Руководство пользователя EasyBuilder Pro

Глава 1. Установка EBPro и запуск....9
1.1 Установка EasyBuilder Pro....9
1.2 Этапы установки EasyBuilder Pro:....10
Глава 2. Работа с Менеджером утилит....16
2.1 Пароль, IP-адрес панели....17
2.2 Инструменты редактора....18
2.2.1 Создание загружаемых данных для карты памяти SD или USB-накопителя....18
2.2.1 Пошаговая загрузка проекта в панель с USB-накопителя или карты памяти SD....18
2.3 Перемещение – Transfer....19
2.3.1 Загрузка – Download....19
2.3.2 Выгрузка - Upload....20
2.4 Имитация работы – Simulation....21
2.4.1 Режимы имитации On-line и Off-line....21
2.5 Проходной режим работы - Pass-through....23
Глава 3. Создание проекта EasyBuilder Pro....24
3.1 Создание нового проекта....24
3.2 Сохранение и компиляция проекта....26
3.3 Имитация работы проекта на ПК....27
3.4 Загрузка проекта в панель....28
Глава 4. Настройка оборудования панели....33
Порты входа/выхода панели оператора....33
4.2. Системная настройка панели оператора....34
4.2.1 Перезагрузка системы....34
4.2.2. Панель инструментов системы....35
4.2.3 Информация о системе....36
4.2.4 Настройка системы....36
Глава 5. Системные параметры....40
5.1 Device (Устройство)....41
5.1.1 Управление локальным ПЛК....42
5.1.2 Управление удаленным ПЛК....46
5.1.3 Управление удаленной панелью оператора....48
5.2 Model (Модель панели оператора)....50
5.3 General (Общие параметры)....53
5.4 System Setting (Системные настройки)....56
5.5 Security (Защита)....59
5.5.1 General mode (Основной режим)....59
5.5.1 Enhanced security mode (Улучшенный режим защиты)....61
5.6 Font (Шрифт)....63
5.6 Extended memory (Внешняя память)....65
5.7 Printer/Backup server (Сервер печати)....67
5.9 E- Mail (Почта)....68
5.10 Recipes (Рецепты)....70

Глава 6. Работа с окнами 72
6.1 Типы окон 72
6.1.1 Base Window - Основное окно 72
6.1.2 Fast Selection Window - Окно быстрого вызова 73
6.1.3 Common Window - Общее окно 74
6.1.4 System Message Window - Окно системных сообщений 75
6.2 Создание, удаление и настройка окна 77
6.2.1 Создание и настройка окна 77
6.2.2 Открытие, закрытие и удаление окна 80
Глава 7. Архив событий 81
7.1 Управление регистрацией событий 81
7.1.1 Редактирование в Excel 82
7.2 Создание нового Архива событий 83
7.2.1 Основные настройки Alarm (Event) Log 83
7.2.2 Настройки сообщения Alarm (Event) Log 85
7.2.3 Настройки e- Mail Alarm (Event) Log 87
7.3 Системные регистры для Архива событий 88
Глава 8. Сбор данных 89
8.1 Управление сбором данных 89
8.2 Создание новой выборки данных 90
8.3 Системные регистры для Data Sampling 94
Глава 9. Общие свойства объектов 95
9.1 Выбор соединения с ПЛК 95
9.1.1 Настройка адресов для считывания и записи 95
9.2 Использование библиотек форм и изображений 98
9.2.1 Настройки библиотеки форм 99
9.2.2 Настройки библиотеки изображений 102
9.3 Настройка текстовой надписи 104
9.4 Настройка размеров профиля 108
9.4 Переменные для номера станции 109
9.5 Широковещательный номер станции 110
Глава 10. Защита объектов 111
10.1 Пароли пользователя и классы доступа объекта 111
10.1.1 General Mode - Общий режим 111
10.1.2 Enhanced Security Mode – Режим повышенной безопасности 112
10.2 Режим повышенной безопасности и управляющий адрес 113
10.2.1 Использование Control Address 113
10.2.2 Введение в команды 113
10.2.3 Введение в результирующие выходы 114
10.3 Режим повышенной безопасности и функциональная кнопка 115
10.3.1 Импорт учетных записей пользователей 115
10.3.2 Использование USB Security Key 117
10.4 Режим повышенной безопасности и список Option List Object 119
10.5 Настройки безопасности объектов 120

10.6 Пример настроек защиты ... 121
Глава 11. Индексный регистр ... 124
11.1 Введение ... 124
11.2 Пример индексного регистра ... 125
Глава 12. Проектирование и использование клавиатуры ... 128
12.1 Этапы разработки всплывающей клавиатуры ... 129
12.2 Использование клавиатуры с непосредственным окном ... 131
12.3 Использование закрепленной на экране клавиатуры ... 132
12.4 Использование клавиатуры UNICODE ... 133
Глава 13. Объекты ... 134
13.1 Bit Lamp - битовая лампа ... 134
13.2 Word Lamp - Лампа управляемая словом ... 137
13.3 Set Bit – установить бит ... 142
13.4 Set Word – Изменить слово ... 146
13.5 Function Key – Функциональная кнопка ... 153
13.6 Toggle Switch - Тумблер ... 160
13.7 Multi-State Switch – Переключатель с несколькими состояниями ... 162
13.8 Slider - Ползунок ... 164
13.9 Numeric Input – Цифровой вход и Numeric Display – Цифровой дисплей ... 167
13.10 ASCII Input и ASCII Display – Ввод и отображение символов ... 177
13.11 Объект [Indirect Window] ... 180
13.12 Объект [Direct Window] ... 183
13.13 Moving Shape – Перемещаемая форма ... 185
13.14 Animation - Анимация ... 190
13.15 Bar Graph - Гистограмма ... 194
13.16 Meter Display – Показывающий прибор ... 199
13.17 Trend Display – Регистрирующий прибор ... 206
13.18 History Data Display – Просмотр архива данных ... 215
13.19 Data Block - Отображение блока данных ... 222
13.20 XY Plot - Двумерный график ... 232
13.21 Alarm Bar и Alarm Display – Бегущая строка и дисплей аварийных сообщений ... 242
13.22 Event Display – Отображение событий ... 245
13.23 Data Transfer (Trigger-based) – Перенос данных по событию ... 253
13.24 Backup - Резервное копирование ... 255
13.25 Media Player - Проигрыватель ... 259
13.26 Data Transfer (Time-based) – Перенос данных по времени ... 268
13.27 Объект [PLC Control] ... 271
13.28 Schedule - Расписание ... 277
13.29 Объект [Option List] ... 294
13.30 Timer - Таймер ... 300
13.31 Video In – Ввод видео сигнала ... 304
13.32 System Message – Системное сообщение ... 308
13.33 Recipe View – Просмотр рецептов ... 310
Глава 14. Библиотека форм и Библиотека изображений ... 313
14.1 Создание Библиотеки форм ... 313
14.2 Создание Библиотеки изображений ... 320
Глава 15. Библиотека меток и использование нескольких языков ... 327

15.1 Введение ... 327
15.2 Создание Библиотеки меток ... 328
15.3 Настройка шрифтов Библиотеки меток ... 329
15.4 Использование Библиотеки меток ... 330
15.5 Использование нескольких языков (системный регистр LW9134) ... 331
Глава 16. Библиотека адресов тэгов ... 333
16.1 Создание Библиотеки адресных меток ... 333
16.2 Использование Библиотеки адресов тэгов ... 335
Глава 17. Передача данных рецептов ... 336
17.1 Обновление наборов данных по сети Ethernet или через USB-кабель ... 337
17.2 Обновление данных с использованием CF/SD-карт или USB-накопителей ... 338
17.3 Передача наборов данных ... 339
17.4 Автоматическое сохранение наборов данных ... 339
Глава 18. Макросы ... 340
18.1 Команды Редактора макросов ... 340
18.2 Состав Макроса ... 349
18.3 Синтаксис ... 350
18.3.1 Константы и переменные ... 350
18.3.2 Операторы - Operators ... 352
18.4 Команды - Statement ... 354
18.4.1 Объявление переменных ... 354
18.4.2 Присвоение ... 354
18.4.3 Логические команды ... 354
18.4.4 Команды выбора - Selective Statements ... 356
18.4.5 Циклы - Reiterative Statements ... 358
18.5 Функциональные блоки ... 360
18.6 Встроенные функциональные блоки ... 363
18.6.1 Математические функции ... 363
18.6.2 Преобразование данных ... 368
18.5.3 Обработка данных ... 372
18.5.4 Преобразование битов ... 375
18.5.5 Коммуникации ... 377
18.6.6 Функции работы со строками ... 391
18.6.7 Функции запроса рецепта ... 411
18.6.8 Различные функции запроса ... 413
18.7 Как создать и выполнить макрос ... 419
18.7.1 Как создать макрос ... 419
18.7.2 Выполнение макроса ... 423
18.8 Макросы определяемые пользователем ... 424
18.8.1 Импорт файла библиотеки функций ... 425
18.8.2 Как использовать библиотеку макрофункций ... 426
18.8.3 Интерфейс библиотеки макрофункций ... 428
18.9 Некоторые замечания об использовании макросов ... 434
18.10 Использование открытого протокола для управления устройством ... 435

18.11 Сообщение компилятора об ошибках 440
18.12 Образцы кода макроса 446
18.13 Макрофункция TRACE 450
18.14 Использование функций работы со стрингами 457
18.15 Защита макросов паролем 468
Глава 19. Настройка панели как MODBUS-сервера 469
19.1 Настройка панели, как устройства MODBUS 469
19.2 Изменение номера станции MODBUS-сервера в процессе работы 474
19.3 Типы адресов MODBUS 475
Глава 20. Подключение устройства считывания штрих-кода 476
20.1 Подключение устройства чтения штрих-кода 476
Глава 21. Соединение по Ethernet и соединение нескольких панелей 479
21.1 Соединение двух панелей оператора 480
21.2 Соединение панели оператора с компьютером 481
21.3 Работа ПЛК, соединенного с панелями оператора 482
Глава 22. Системные биты и регистры 483
22.1 Диапазоны Адресов памяти локальной панели 484
22.1.1 Биты 484
22.1.2 Слова 485
22.2 Системное время панели 486
22.3 Пароль и уровень доступа 487
22.4 Опрос данных 488
22.5 Архив событий 489
22.9 Операции с аппаратурой панели 491
22.7 Информация о сети локальной панели 492
22.8 Рецепты и расширенная память 493
22.9 Управление местами хранения данных 494
22.10 Координаты X и Y сенсорного экрана 495
22.11 Переменные номеров станций 496
22.12 Индексный регистр 497
22.13 Информация о файле MTP 498
22.14 Коммуникации сервера MODBUS 499
22.15 Настройки коммуникационных параметров 501
22.16 Состояние связей с ПЛК (COM - порты) 504
22.17 Состояние связей с ПЛК (Ethernet) 506
22.18 Состояние связей с ПЛК (USB) 509
22.19 Состояние связей с ПЛК (CAN Bus) 509
22.20 Состояние связей с удаленными операторскими панелями 510
22.21 Состояние связей с удаленными ПЛК 517
22.1 Сообщения об ошибках связи и номер исполняемой команды 520
22.2 Различные функции 521
22.24 Удаленная печать / Резервный сервер 523
22.25 Легкий доступ (EasyAccess) 524
22.26 Настройки функции сквозного пропуска (Pass-Through) 524
22.27 Отключение окна «PLC No Response» 525
22.28 Панель и ключ защиты 525
22.29 Управление Окном быстрого выбора (FS) 526

22.30 Функции объекта ввода данных 526
22.31 Ограничение локальных / удаленных операций 526
Глава 23. Поддерживаемые принтеры 527
23.1 Типы поддерживаемых принтеров 527
23.2 Как добавить новый принтер и начать печатать 530
23.2.1 Добавьте тип принтера 530
23.2.2 Начало печати 531
Глава 24. Редактор рецептов 532
24.1 Введение 532
24.2 Настройка Редактора Рецептов/Расширенной памяти 532
24.3 Записи рецептов 535
Глава 25. EasyConverter 538
25.1 Экспорт файлов DTL или EVT в Excel 538
25.2 Использование функции масштабирования 540
25.3 Использование преобразования нескольких файлов (Multi-File Conversion) 541
Глава 26. EasyPrinter 542
26.1 Использование EasyPrinter в качестве Сервера печати 543
26.2 Использование EasyPrinter в качестве Резервного сервера 546
26.2.2 Процедура установки в EasyBuilder Pro 547
26.3 Применение EasyPrinter 550
26.3.1 Внешний вид 550
26.3.2 Руководство по применению 551
26.4 Командный файл для конвертирования 556
26.4.1 Convert Batch File (по умолчанию) 556
26.4.2 Специальные критерии 557
26.4.3 Формат Convert Batch File 558
26.4.4 Порядок анализа критериев 558
Глава 27. Имитатор работы 559
27.1 Подготовка файлов 559
27.2 Изменение содержимого «xob_pos.def» 560
Глава 28. Взаимодействие нескольких панелей 561
28.1 Как создать проект ведущей панели (Master HMI) 562
28.2 Как создать проект в ведомой панели (Slave HMI) 563
29.1 Ethernet Mode 567
29.1.1 Как изменить Virtual Serial Port 568
29.1.1 Как использовать режим «Ethernet Mode» 570
29.2 COM Port Mode 572
29.2.1 Настройка режима «COM Port Mode» 572
29.2.2 Режимы работы панели 574
29.3 Использование системных адресов для активации функции Pass-Through 577
Глава 30. Защита проекта 578
30.1 Пароль для файла XOB 579
30.2 Декомпиляция запрещена 580
30.3 Блокировка функции выгрузки из панели [LB-9033] 581
30.4 Ключ проекта (Project Key) 582

30.5 Пароль проекта MTP (Project Password)) ... 583
Глава 31. Протокол Memory Map ... 584
Глава 32. Использование сервера FTP ... 592
32.1 Login FTP Server ... 592
32.2 Резервное копирование архивных данных и обновление рецептов. ... 594
Глава 33. Отладчик - EasyDiagnoser ... 595
33.1 Введение и настройка ... 595
33.2 Настройки EasyDiagnoser ... 598
33.3 Код ошибки - Error Code ... 604
33.4 Сохранить как - Save As ... 605
33.5 Настройка окна - Window Adjustment ... 606
Глава 34. Свободные тэги Rockwell EtherNet/IP ... 607
34.1 Импорт файла CSV User-Defined Tag в EasyBuilder Pro ... 608
34.2 Adding New Data Type ... 610
34.3 Вставка данных ... 612
34.4 Прочие функции ... 614
34.5 Тэги привязанные к модулю - Module-Defined ... 618
Глава 35 Монитор - Easy Watch ... 622
35.1 Введение ... 622
35.1.1 Что такое Easy Watch? ... 622
35.1.2 Зачем нужен Easy Watch? ... 622
35.2 Основные функции ... 623
35.2.1 Основные функции ... 623
35.2.2 Быстрый выбор инструментов ... 624
35.3 Настройки монитора ... 625
35.3.1 Добавить монитор ... 625
35.3.2 Настройки монитора ... 625
35.3.3 Добавить новое устройство (New Device) ... 626
35.4 Настройки Macro ... 629
35.4.1 Добавить Macro ... 629
35.4.2 Настройки Macro ... 629
35.4.3 Добавить новый Macros в список ... 630
35.5 Управление панелью - HMI Manager ... 631
35.5.1 Настройки панели - HMI Settings ... 631
35.5.2 HMI Manager ... 631
35.5.3 Добавить новое устройство ... 631
35.6 Список объектов - Object List ... 633
35.6.1 Настройка страницы ... 633
35.6.2 Колонки списка объектов ... 633
Глава 36 Инструменты Администратора ... 634
36.1 Введение: ... 634
36.2 Учетные записи пользователя - User Accounts ... 635
36.2.1 Введение в User Accounts ... 635
36.2.2 Настройка User Accounts ... 636
36.2.3 Импорт Учетных записей при помощи EasyBuilder Pro ... 638

36.3 USB-ключ защиты – USB Security Key 639
36.3.1 Введение в использование USB-ключа защиты 639
36.3.3 Настройки *USB Security Key* в EasyBuilder Pro 641
36.4 Настройки сервера e-Mail SMTP 642
36.4.1 Введение в настройки e-Mail SMTP 642
36.5 Контакты e-Mail 644
36.5.1 Введение в Контакты e-Mail 644
36.5.2 Настройки Контактов e-Mail 645
36.5.3 Использование EasyBuilder Pro для импорта настроек e-Mail и Контактов 647

Глава 1. Установка EBPro и запуск

1.1 Установка EasyBuilder Pro

Программное обеспечение:

Загрузку программы конфигурации EasyBuilder Pro можно произвести с компакт-диска EasyBuilder Pro или скачать установочный файл с сайта http://www.weintek.com компании Weintek Labs для получения любой доступной языковой версии (включая упрощенной китайской, традиционной китайской, английской, французской, корейской, итальянской, испанской и русской) и последние файлы обновления.

Рекомендуемые системные требования:

Процессор: Intel Pentium II и выше
ОЗУ: 256 Мбайт и больше
Жесткий диск: 2,5 Гбайт и выше (500 Мбайт свободного места – как минимум)
CD-ROM: 4X и выше
Монитор: 256 цветов SVGA с разрешением 1024x768 или выше
Клавиатура и мышь
Сеть Ethernet: для загрузки/выгрузки проекта
Порт USB 2.0 для загрузки/выгрузки проекта
Порт RS-232 COM: один последовательный порт для имитации в режиме он-лайн (минимум)
Принтер

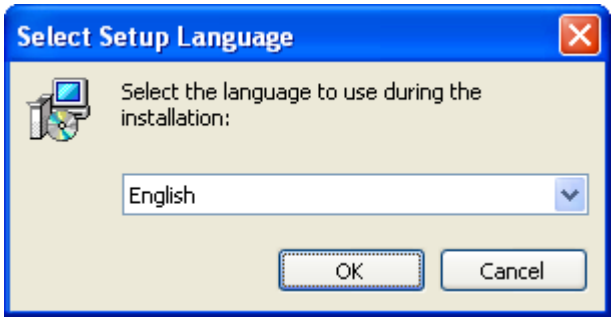
Операционная система:

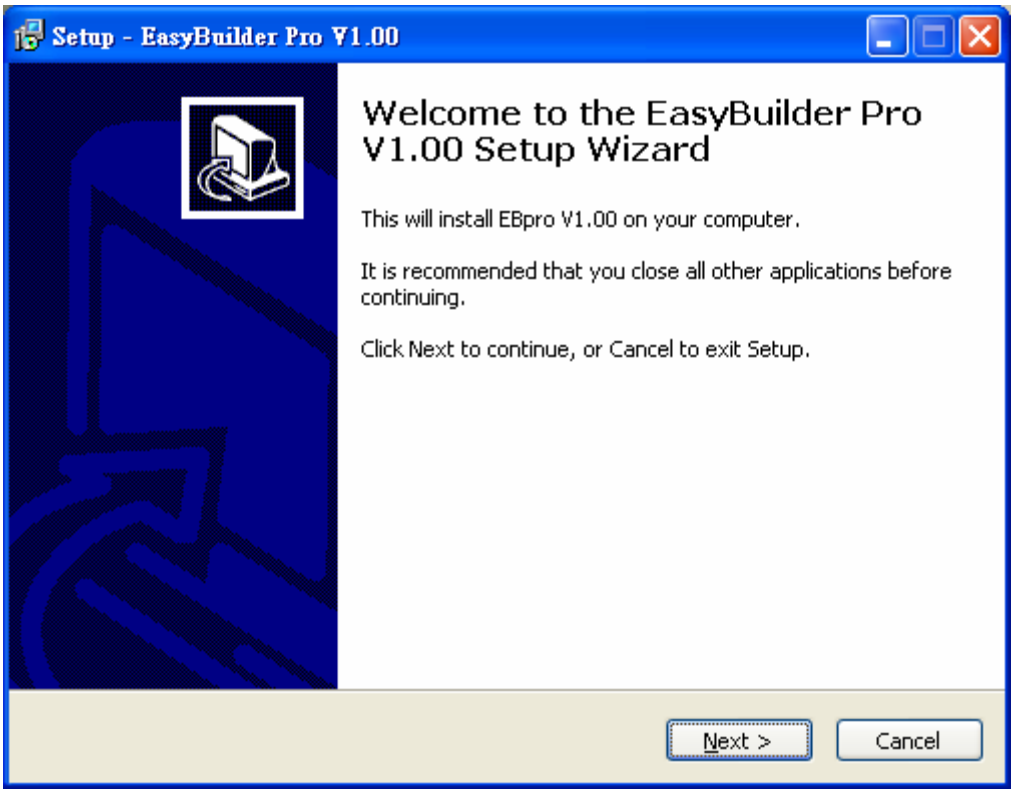
Windows XP/Windows Vista/ Windows 7

1.2 Этапы установки EasyBuilder Pro:

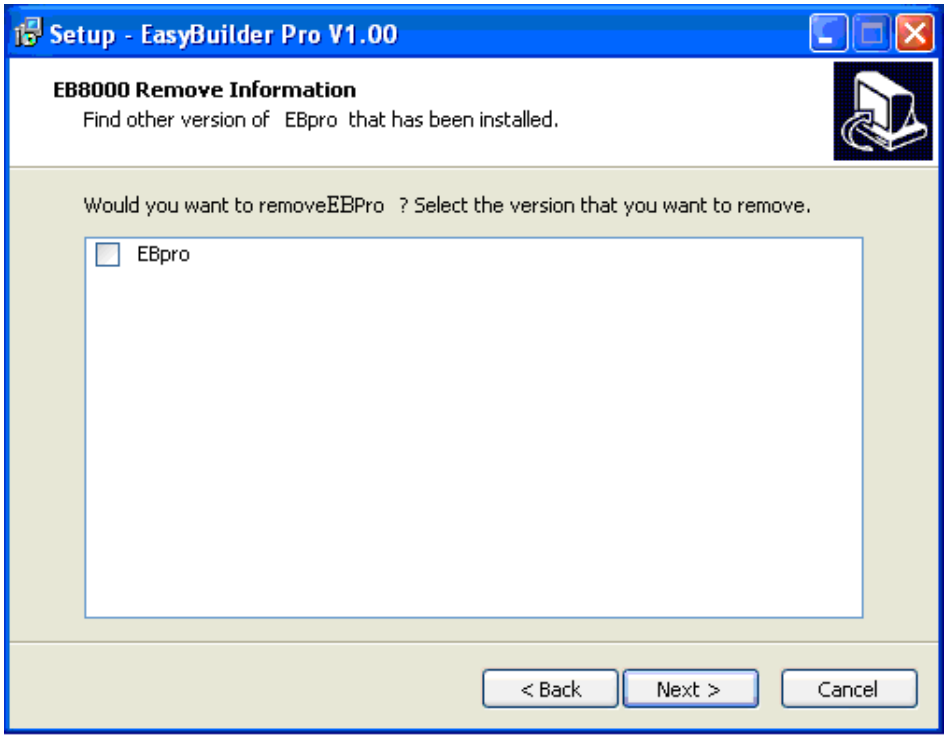
1) Вставьте установочный CD EasyBuilder Pro в CD-привод. После загрузки компакт-диска, автоматически запустится программа установки и появится инсталляционное окно, показанное ниже. Либо можно запустить файл [Autorun.exe] вручную из корневой папки CD-диска.:

2) Щелкните **[Install]** и появится окно, выберите язык используемый при инсталляции нажмите **[Next]** и следуете далее в соответствии с указаниями на экране:

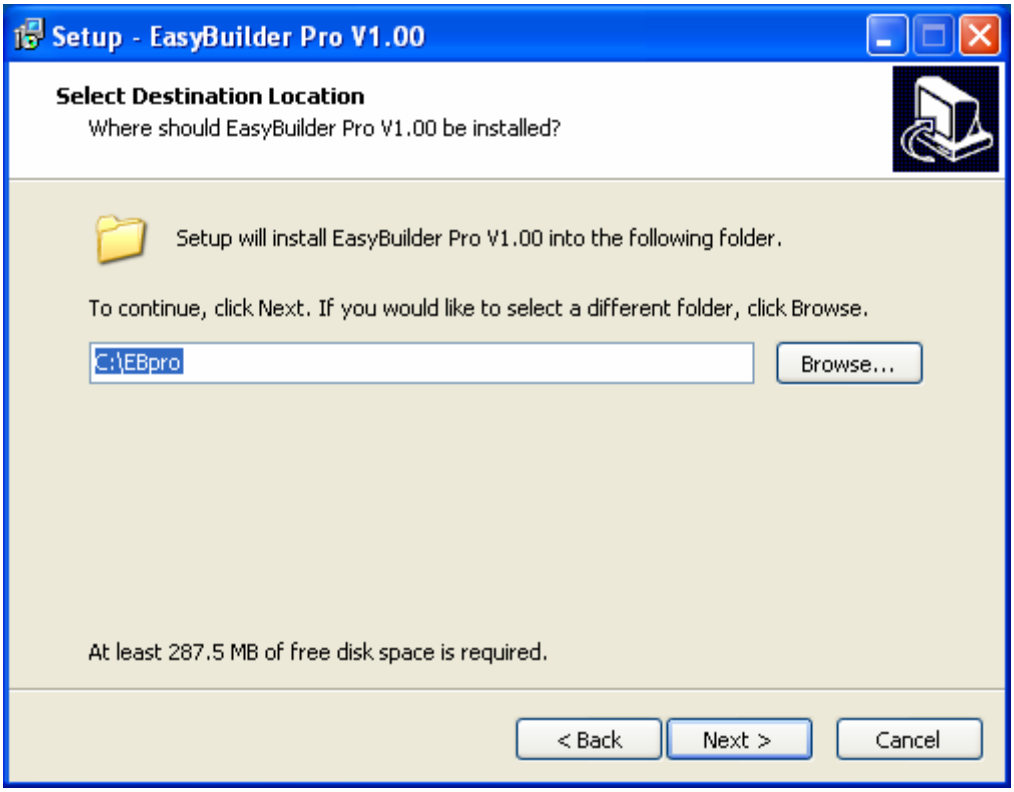




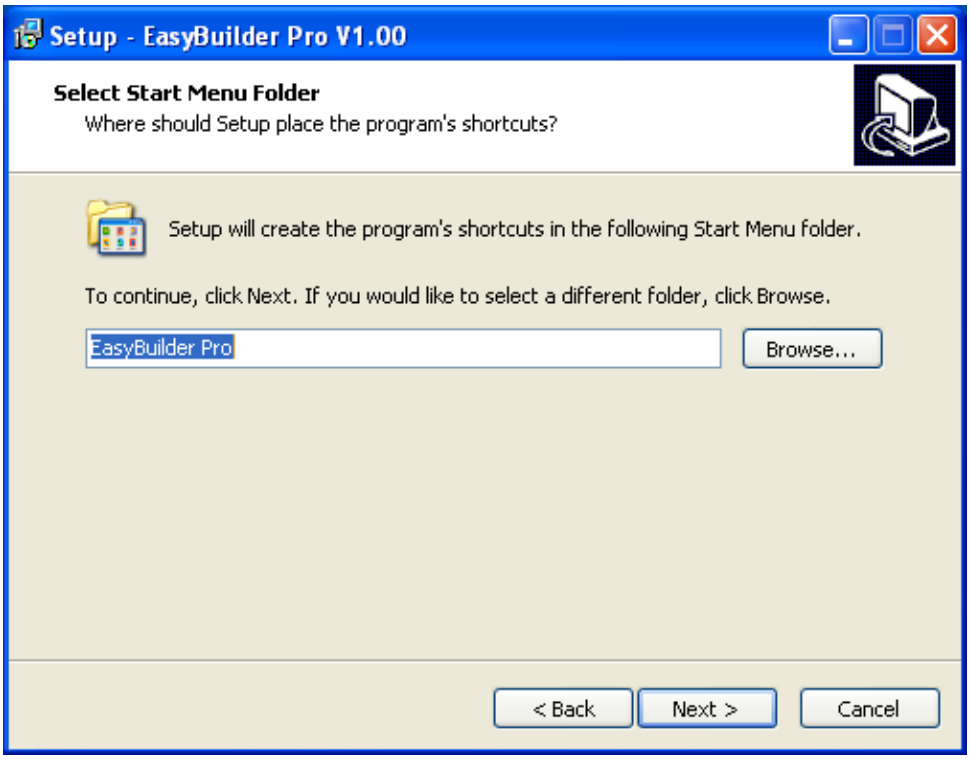
3) Появится окно с вопросом: хотите ли Вы удалить старые версии EasyBuilder, отметьте те версии, которые надо удалить и щелкните **[Next].**



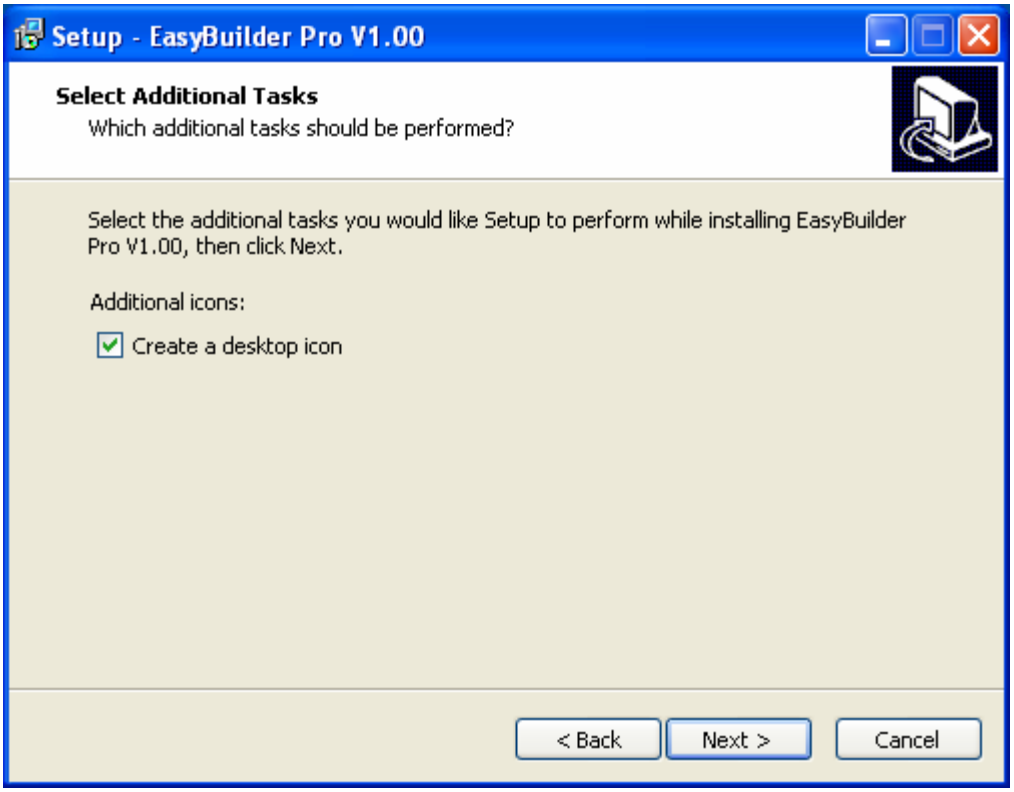
4) Назначьте новую папку для установки EasyBuilder Pro или выберите рекомендованную папку и нажмите **[Next]**.



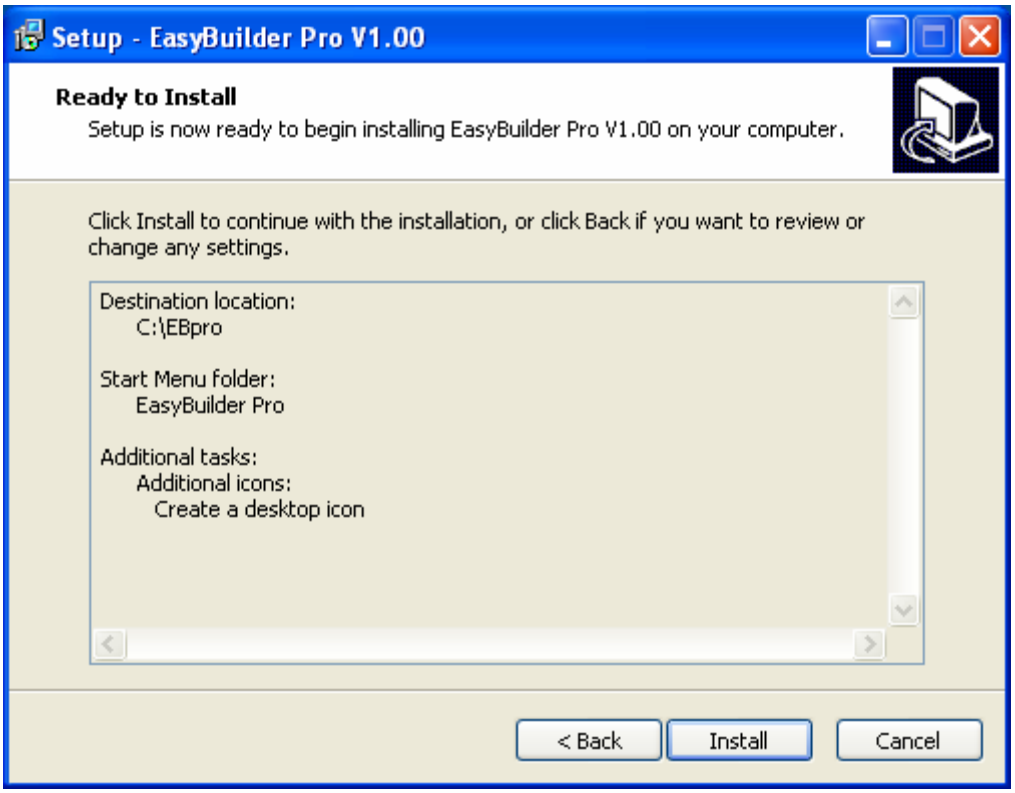
5) Далее выберите папку для Стартового меню, чтобы сохранить быстрый доступ к программе. Нажмите **[Browse],** чтобы выбрать папку или используйте рекомендованную папку, нажмите **[Next]**.



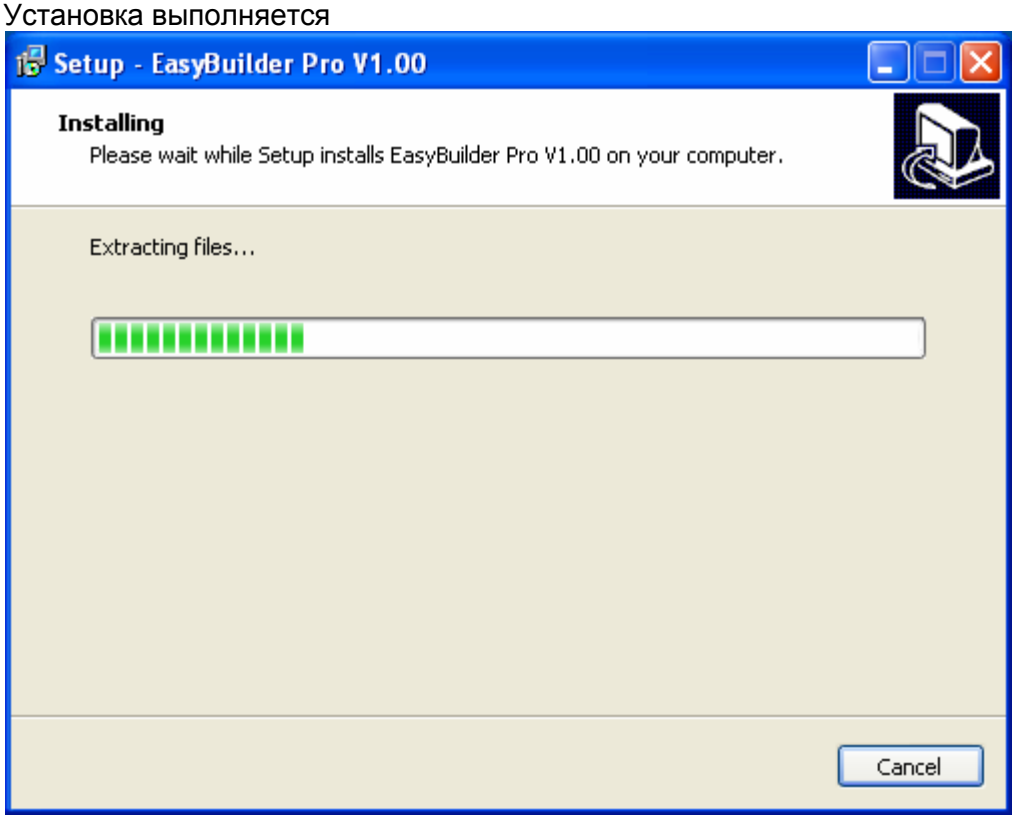
6) Выберите дополнительные задачи, например **[Create a desktop icon]**.- Создать пиктограмму на рабочем столе. Сделайте отметку, если необходимо и нажмите **[Next]**.



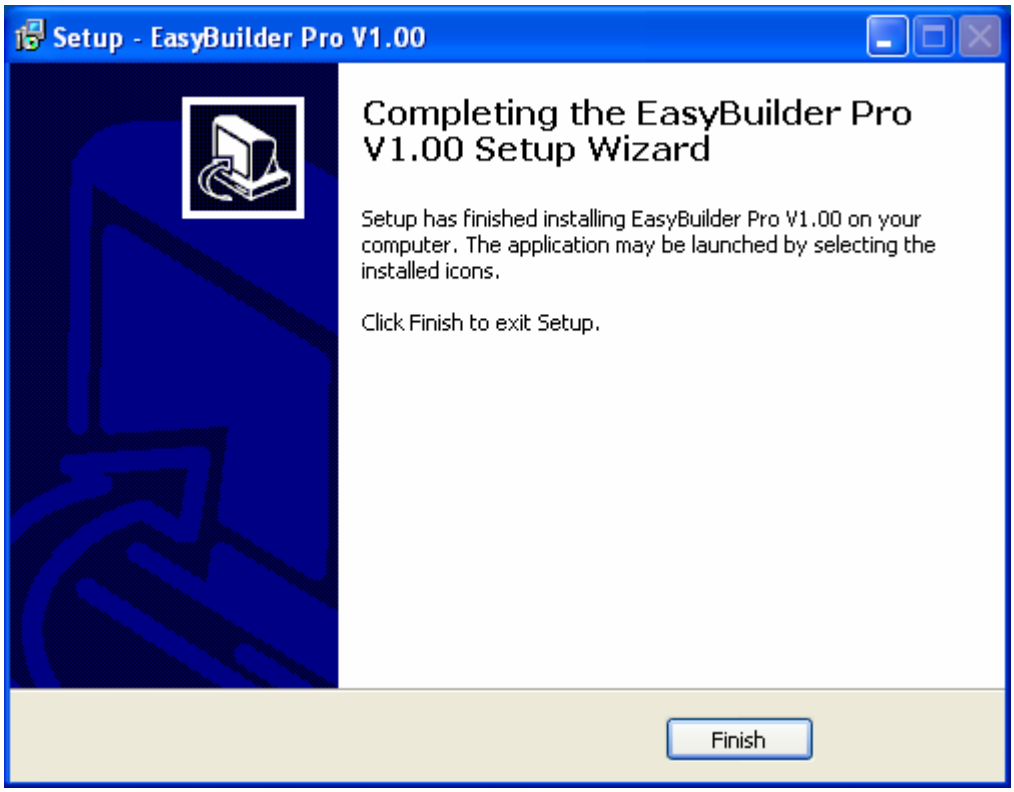
7) К этому моменту все настройки сделаны, проверьте их и для внесения изменений нажмите **[Back]** или нажмите **[Install],** чтобы начать установку**.**



8) Установка выполняется



9) Нажмите **[Finish],** чтобы закончить установку.



10) Для запуска программы используйте меню **[Start]** / **[All Programs]** / **[EasyBuilder Pro]**.

Назначение компонентов меню EasyBuilder Pro:

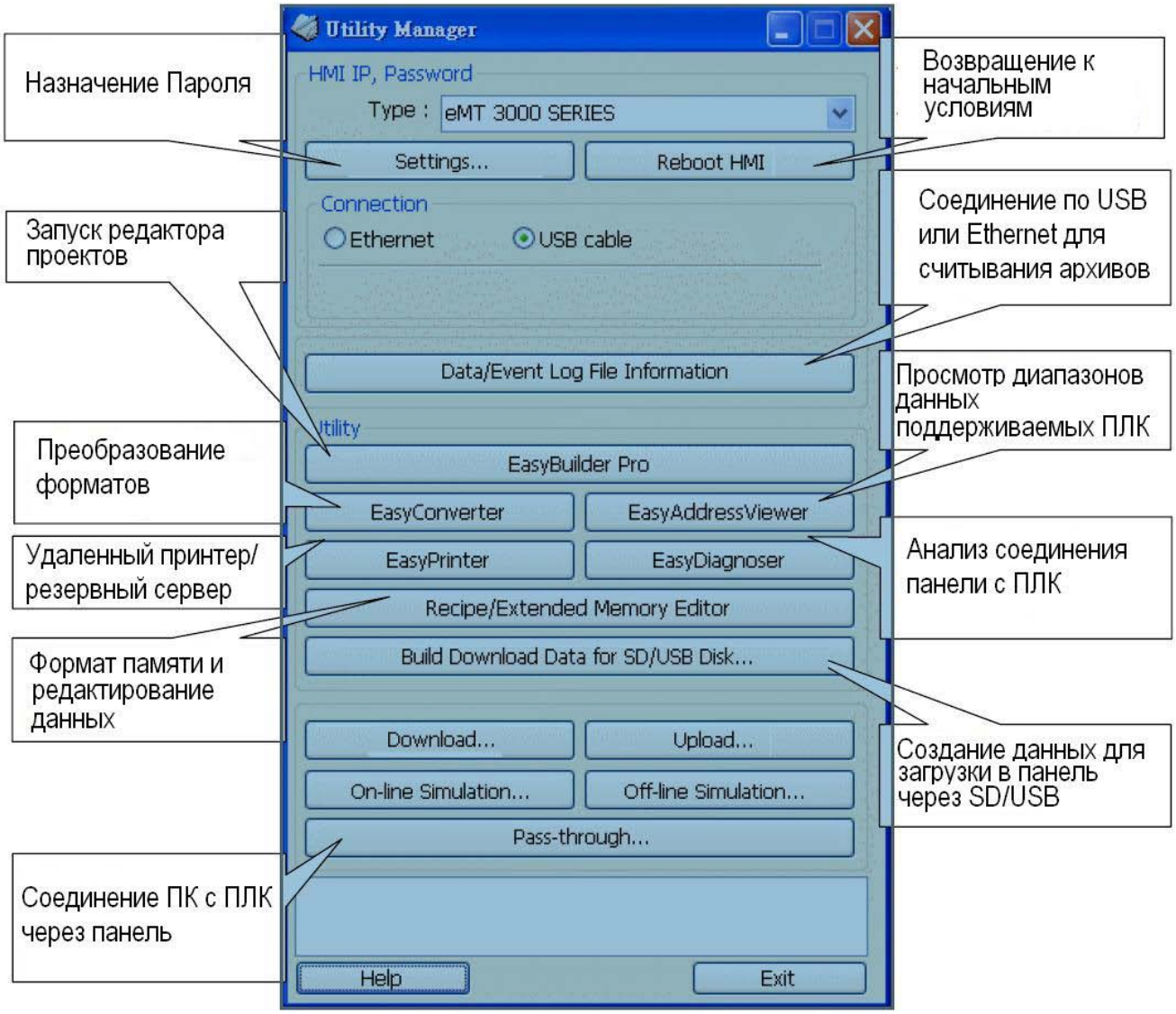
Файл	Описание
Administrator Tool	Сохранение данных *User Accounts, USB Security Key, e-Mail SMTP Server Setting, e-Mail Contacts* на USB и импорт в панель оператора (*HMI*).
EasyBuilder Pro	Редактирование проектов
EasyConverter	Инструмент преобразования форматов данных *Data Sampling* и *Event Log*
EasyDiagnoser	Инструмент для анализа связи ПЛК с операторской панелью
EasyPrinter	Инструмент для печати или сохранения информации.
EasySimulator	Средство имитации выполнения проекта на ПК со соединением или без соединения с ПЛК
EasyWatch	Мониторинг и изменение значений адресов ПЛК
Recipe Editor	Инструмент для настройки формата данных. Пользователь может открыть *Recipe data* или данные в *External Memory*
Release Note	Замечания к версиям программного обеспечения и новейшая информация
Structure Editor	Поддержка механизма *AB TAG* для повышения гибкости при чтении/записи объектов
Utility Manager	Управление проектами EasyBuilder Pro

Refer to

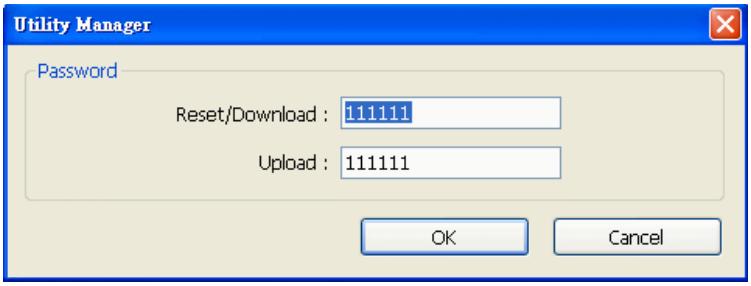
Панели eMT Series поддерживают загрузку/выгрузку проекта по кабелю USB. После установки EasyBuilder Pro, перейдите к [Computer Management] / [Device Manager], чтобы проверить, установлен ли драйвер USB, если не установлен, установите его вручную, используя ЧЗВ на сайте www.weintek.com

Глава 2. Работа с Менеджером утилит

После установки EasyBuilder Pro.сделайте двойной щелчок по **[Utility Manager]**. Менджер утилит (Utlity Manager) это программная оболочка для запуска нескольких утилит. Некоторые функции дублируются в EasyBuilder Pro. Utlity Manager может функционировать как самостоятельная программа.



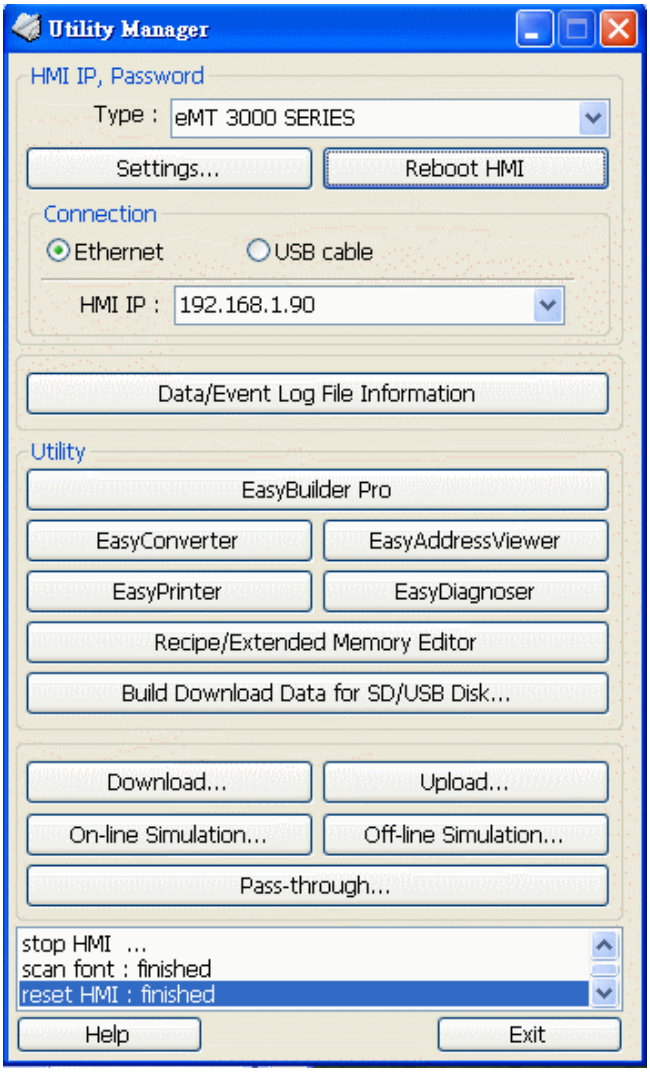
2.1 Пароль, IP-адрес панели



[Settings]
Для работы панели (HMI) по сети Ethernet или USB необходимо назначить пароль..
Пароль обеспечивает защиту от несанкционированного доступа к HMI

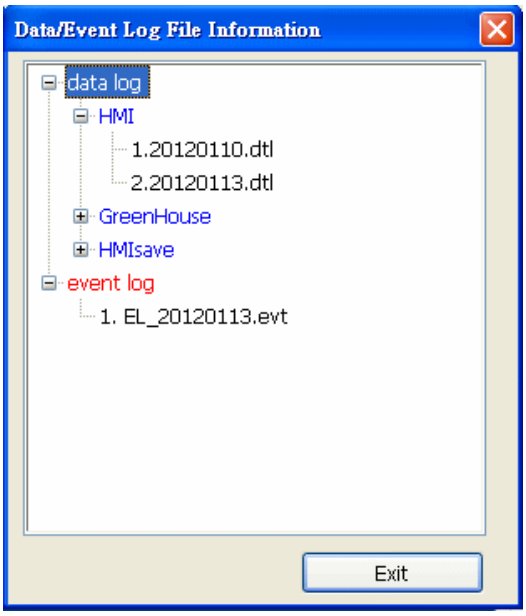
Для функций **[Reset]** и **[Download]** используется одинаковый пароль, а для функции **[Upload]** – другой.

Сохраняйте любые изменения пароля, иначе при изменении пароля на пароль по умолчанию проект и данные в панели будут полностью стерты.



[Reboot HMI]
В некоторых ситуациях необходимо перезагрузить панель. Пользователь не должен отключать питание для перезагрузки. После перезагрузки всё возвращается к условиям начального запуска. Установите корректный IP-адрес при соединении по Ethernet.

[Data/Event Log File Information]
После ввода настроек, соединитесь с панелью и проверьте число архивных файлов (history files) в панели.

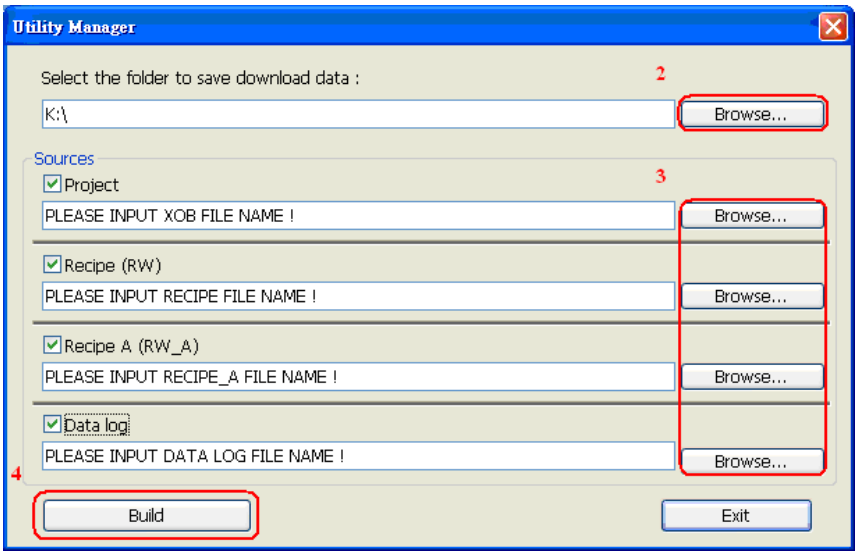


2.2 Инструменты редактора

2.2.1 Создание загружаемых данных для карты памяти SD или USB-накопителя

1. Вставьте SD/USB в компьютер
2. Назначьте путь сохранения данных
3. Создайте данные

Файлы-источники будут сохранены во вставленном устройстве для дальнейшей загрузки в панель. Эта функция предназначена для создания требуемых данных.



2.2.1 Пошаговая загрузка проекта в панель с USB-накопителя или карты памяти SD

Пример загрузки данных из папки “**123**” в USB-накопитель ***(K:\123)***

1. Вставьте USB-накопитель с проектом в панель
2. В диалоговом окне Загрузки/Выгрузки данных выберите [Download]
3. Введите пароль загрузки [Download Password].
4. Проверьте загружаемые файлы проекта и архивные файлы в диалоговом окне [Download Settings] настроек загрузки
5. Нажмите OK
6. В окне [Pick a Directory] выбора директории, укажите путь ***usbdisk/device-0/123***
7. Нажмите OK

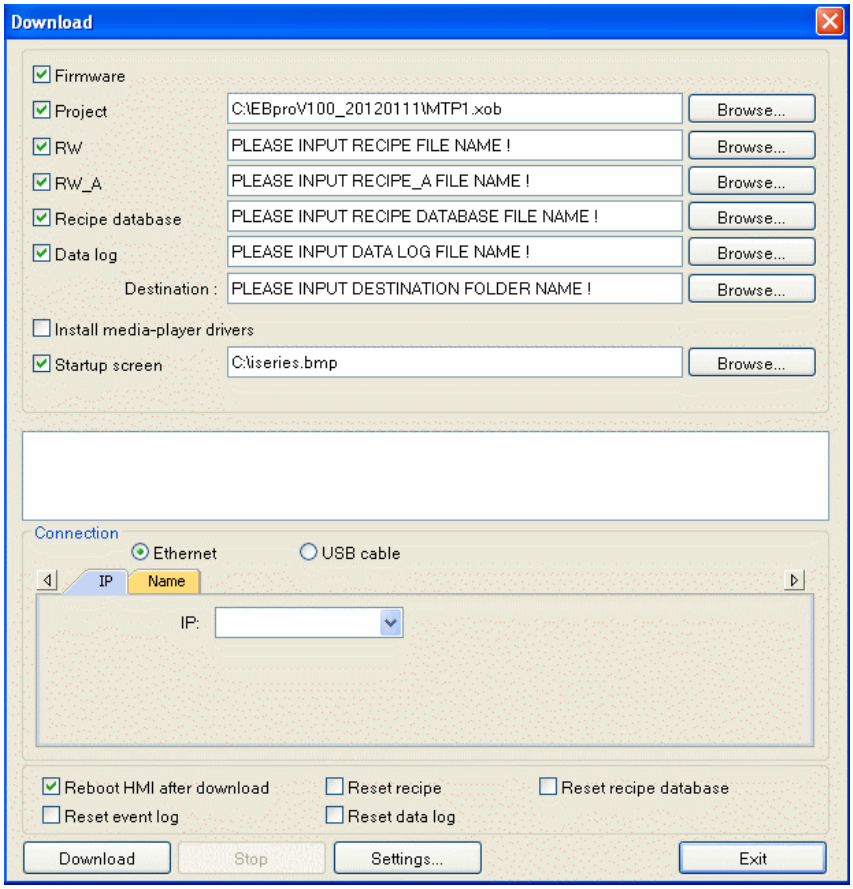
Проект автоматически обновится.

Даже если пользователь загрузил только архивные файлы, все равно необходимо перезагрузить (reboot) панель вручную, чтобы обновить файлы.

2.3 Перемещение – Transfer

2.3.1 Загрузка – Download

Загрузка исходных файлов в панель по Ethernet или USB- кабелю.



Firmware.

Отметьте флаг [Firmware] для обновления базовых программ панели оператора. Это необходимо сделать при первичной загрузке данных в панель.

Project

Выберите файл проекта в формате «.XOB»

Recipe data RW/ RW_A

Выберите файл в формате «.rcp» в папке Recipe

Data log

Выберите файл в формате «.dtl» в папке Datalog

Install X-series media-player drivers

Необходимо отметить при первой загрузке проекта, EBPro в панели X-серии.

Startup Screen

Загрузка файла «.BMP» в панель. В панели он будет появляться после перезагрузки. Пользователь может использовать логотип своей компании.

Reboot HMI after download

Автоматический перезапуск панели после загрузки

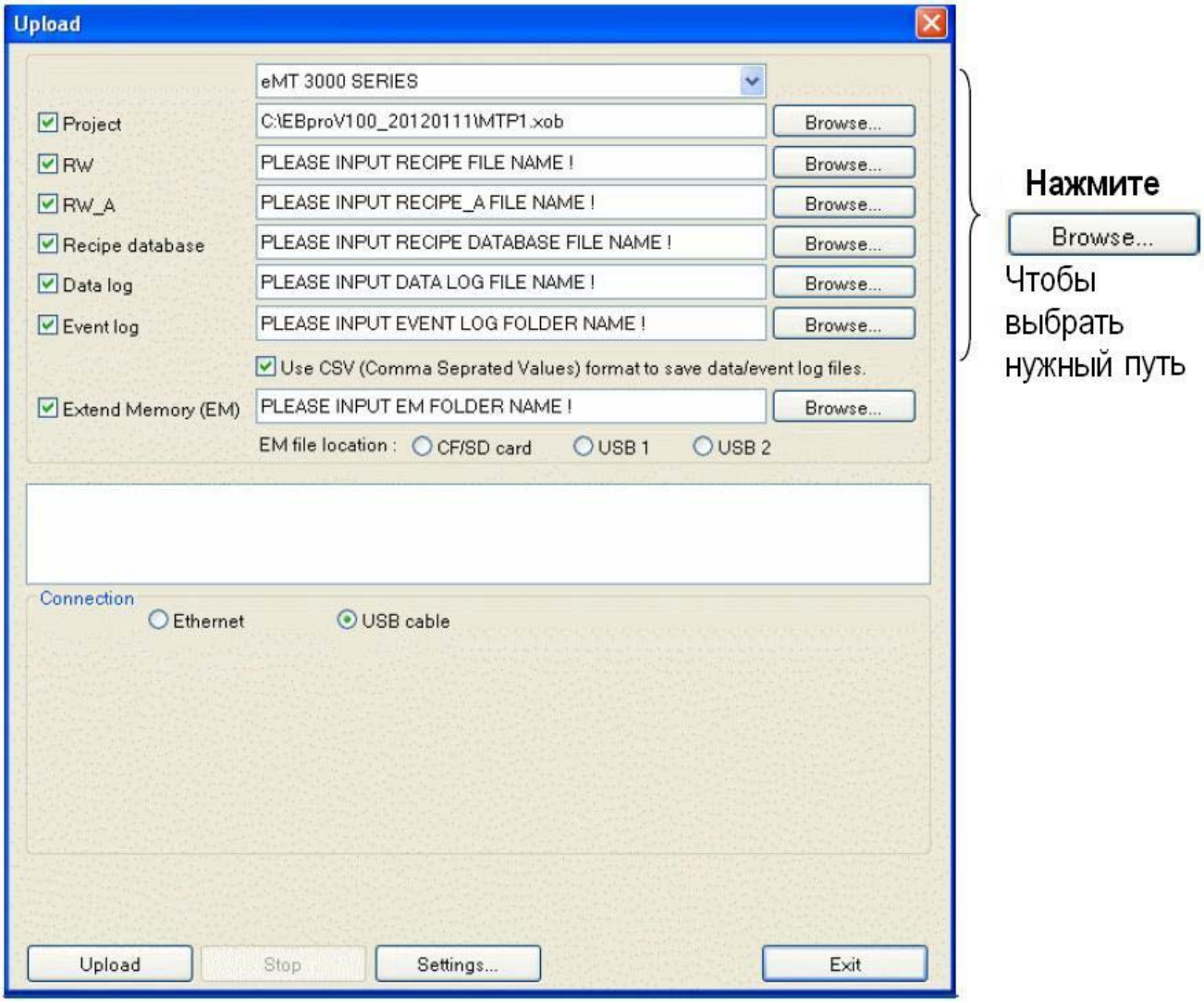
Reset recipe, Reset recipe database, Reset event log, Reset data log

Удалить указанные файлы в панели перед загрузкой.

2.3.2 Выгрузка - Upload

Выгрузка файлов из панели оператора в компьютер по сети Ethernet или USB-кабелю.

Пользователь должен выбрать путь сохранения файла до выгрузки:



О данных типа **Project / Recipe data RW / RW_A / Data log** подробнее в 2.3.1

Event Log
Выгрузка файла «.evt» из панели в компьютер.

Extended Mtmory (EM)
Выгрузка файла «.emi» в карту памяти SD или в USB-накопитель.

Файлы будут выгружены в компьютер в формате «*. XOB». Для редактирования этого файла используя EasyBuilder Pro, декомпилируйте сначала этот файл в файл формата «*.MTP».

2.4 Имитация работы – Simulation

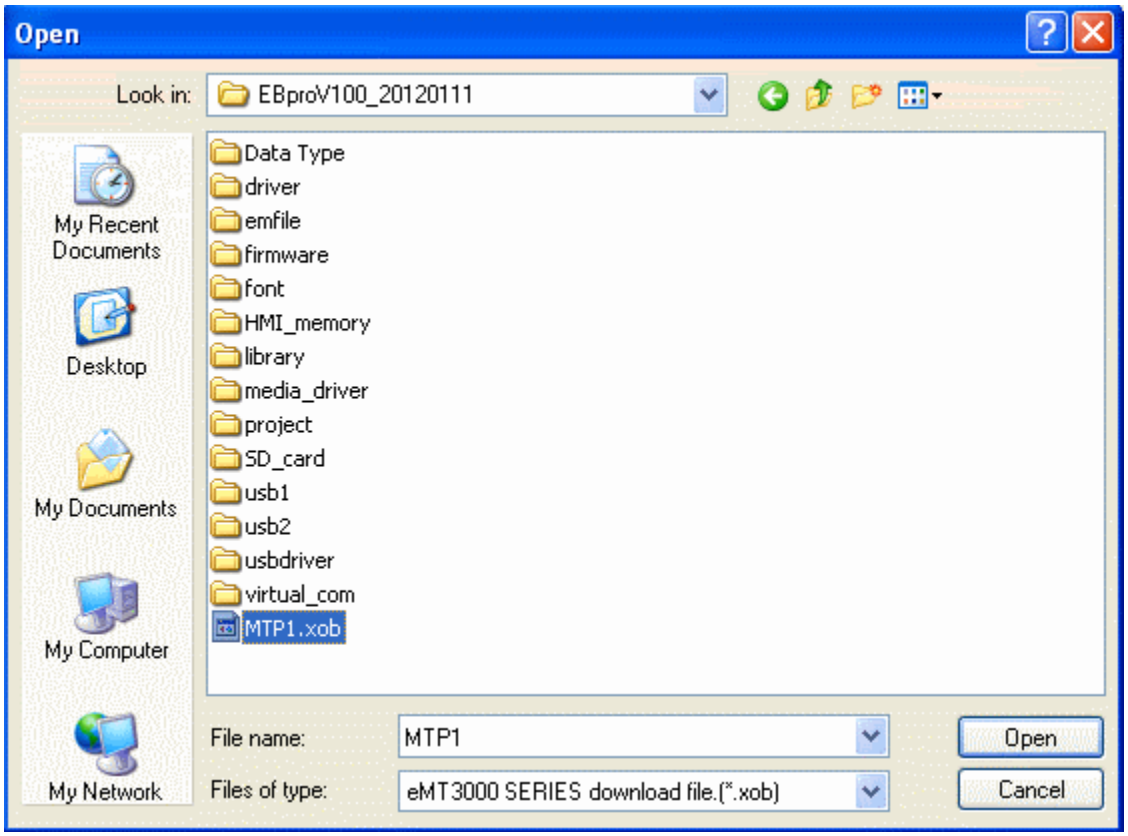
2.4.1 Режимы имитации On-line и Off-line

В режиме «Off-line Simulation» проект исполняется на компьютере

В режиме «On-line Simulation» проект исполняется на компьютере

Note	В режиме «On-line Simulation» на компьютере, если целевым устройством является локальный ПЛК (ПЛК непосредственно подключен к компьютеру), **предел работы в этом режиме составляет 10 минут**.

Перед выполнением Он-лайн/Офф-лайн имитации, выберите файл проекта «*.xob».
.



При выполнении Он-лайн/Офф-лайн имитации, щелчком правой кнопки можно выбрать следующие функции:.

Exit simulation]

Остановить имитацию работы..

[Run EasyDiagnoser]

Для контроля текущего состояния соединения.

[Screenshot]

Захват и сохранение изображения на экране, как файл картинки в папке «screenshot» в установочной директории.

2.5 Проходной режим работы - Pass-through

Эта функция позволяет компьютерным приложениям соединяться с ПЛК через панель оператора. После запуска этой функции, панель оператора работает как преобразователь.

Функция [Pass-through] обеспечивает возможность работы по сети Ethernet или через COM-порт.

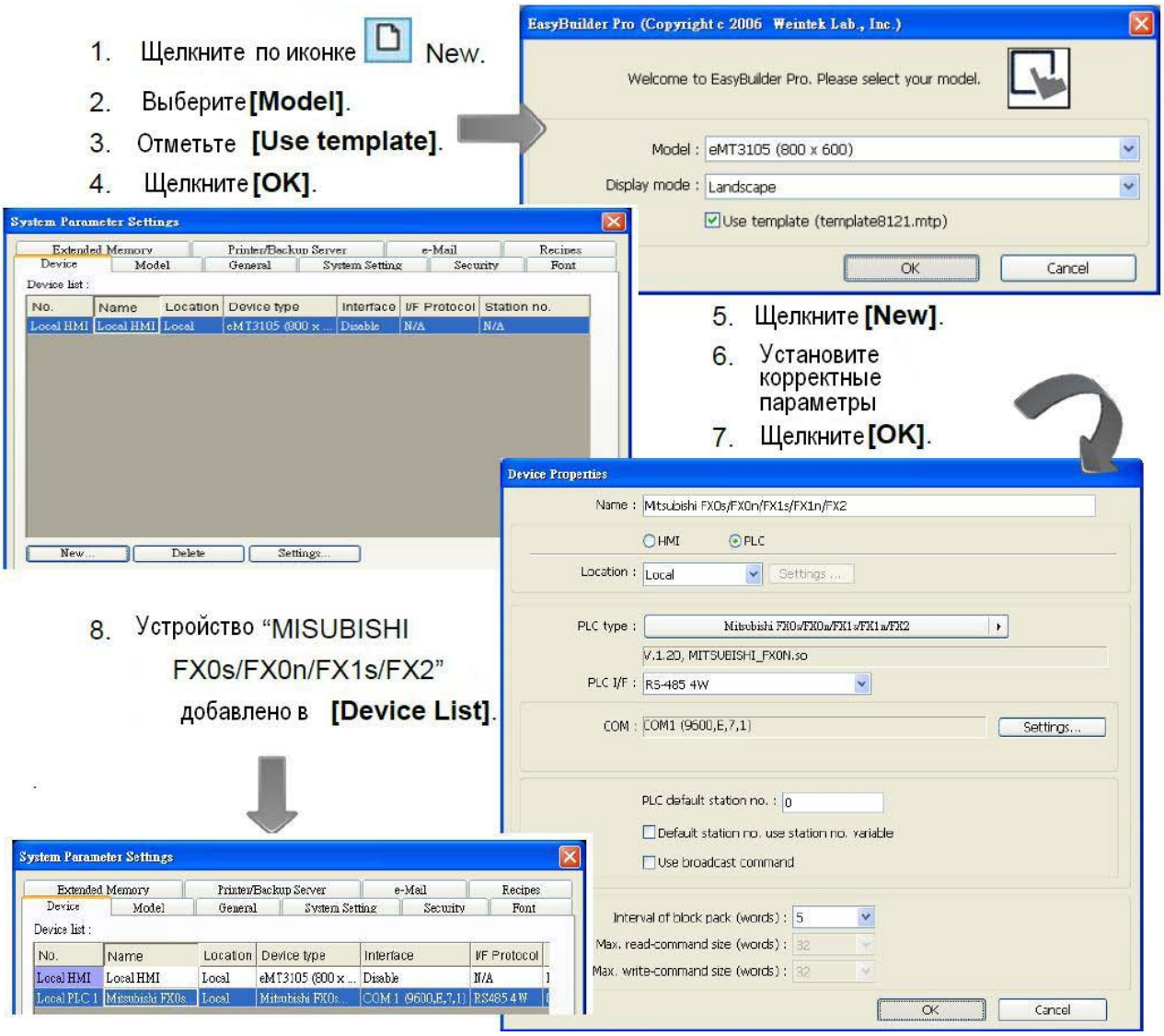
При использовании Ethernet сначала необходимо установить драйвер виртуального последовательного порта

Refer to Подробности в главе 29 « Функция проходного режима».

Глава 3. Создание проекта EasyBuilder Pro

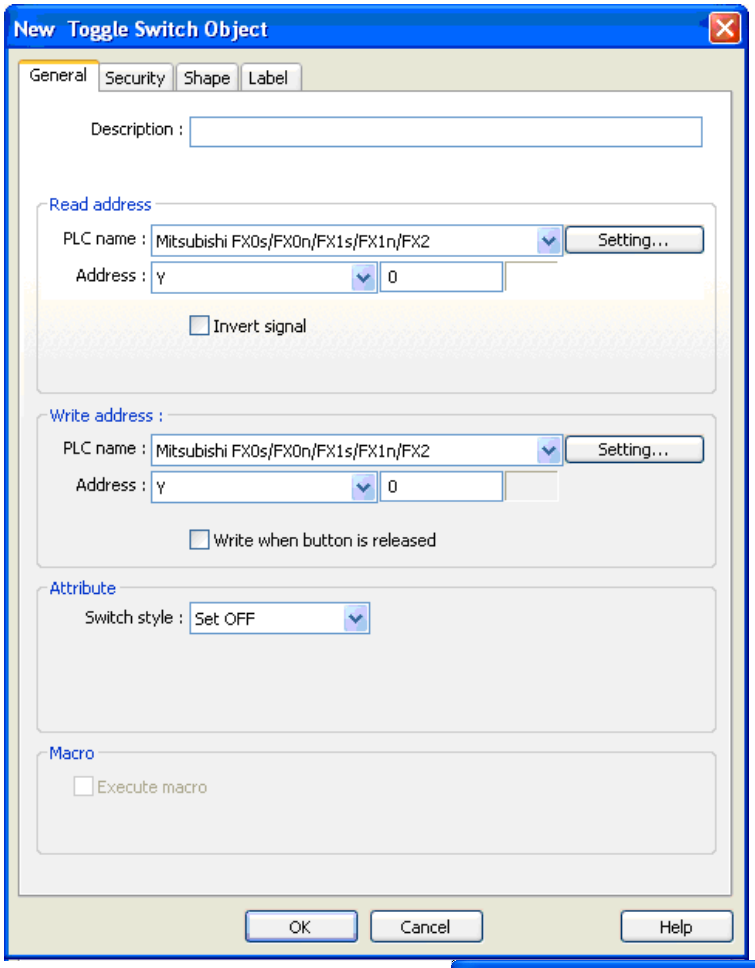
Ниже приведен пример создания простейшего проекта, его компиляции и имитации на ПК и загрузки в панель оператора

3.1 Создание нового проекта

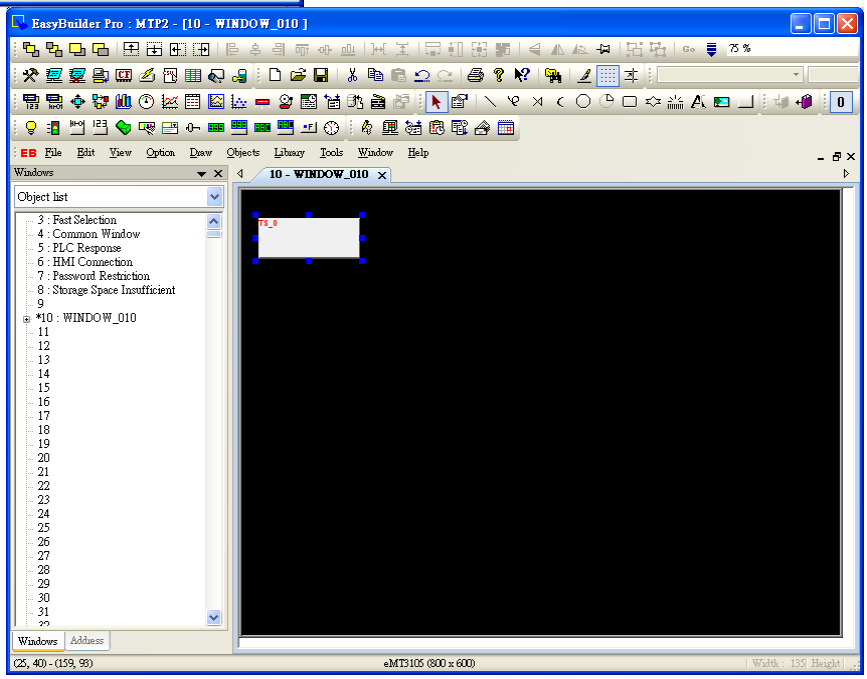


Теперь пользователь может создать простейший проект, содержащий объект «Двухпозиционный переключатель».

1. Щелкните пиктограмму объекта (Toggle Switch)
2. Установите корректные параметры

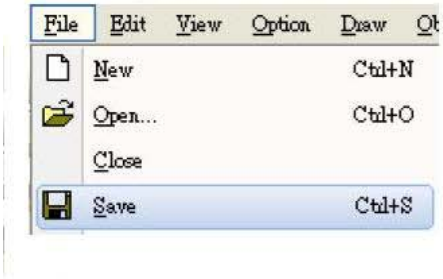


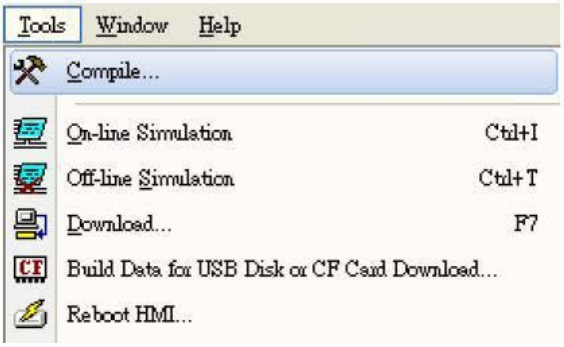
3. Разместите объект в нужном месте окна.
4. Проект с одним объектом создан



3.2 Сохранение и компиляция проекта

Панель инструментов (Tool Bar)
EasyBuilder Pro.
.

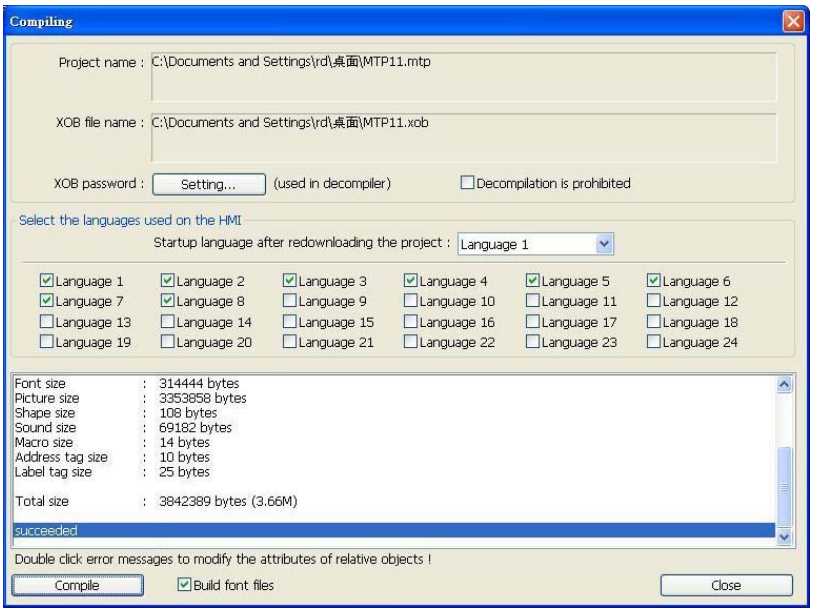




1. Щелкните по [Save], чтобы сохранить файл «*.MTP»

2. После сохранения файла с расширением «*.mtp» выберите команду [Compile] для проверки корректности проекта и получения файла с расширением “.xob” для загрузки в панель.

Пользователь может выбрать язык используемый в проекте, отмечая **[Language 1 to 8]**.



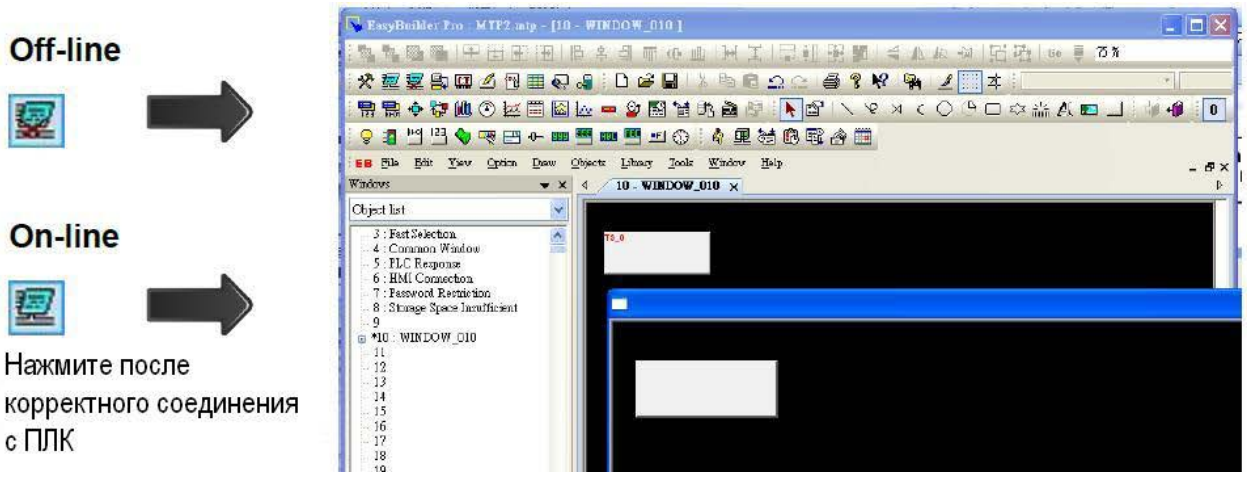
Если файл скомпилирован успешно, появится диалоговое окно

3.3 Имитация работы проекта на ПК

В режиме «Off-line Simulation» проект исполняется на компьютере

В режиме «On-line Simulation» проект исполняется на компьютере

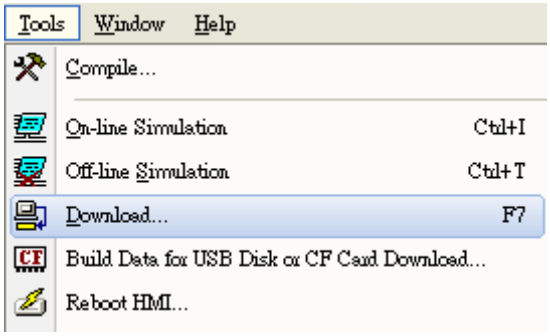
В режиме «On-line Simulation» на компьютере, если целевым устройством является локальный ПЛК (ПЛК непосредственно подключен к компьютеру), **предел работы в этом режиме составляет 10 минут**.



3.4 Загрузка проекта в панель

✹ Способ 1. [Ethernet] / HMI IP:

Перед Загрузкой **[Download]**, убедитесь в корректности всех настроек.



Введите **[Password]**
Укажите **[HMI IP]**

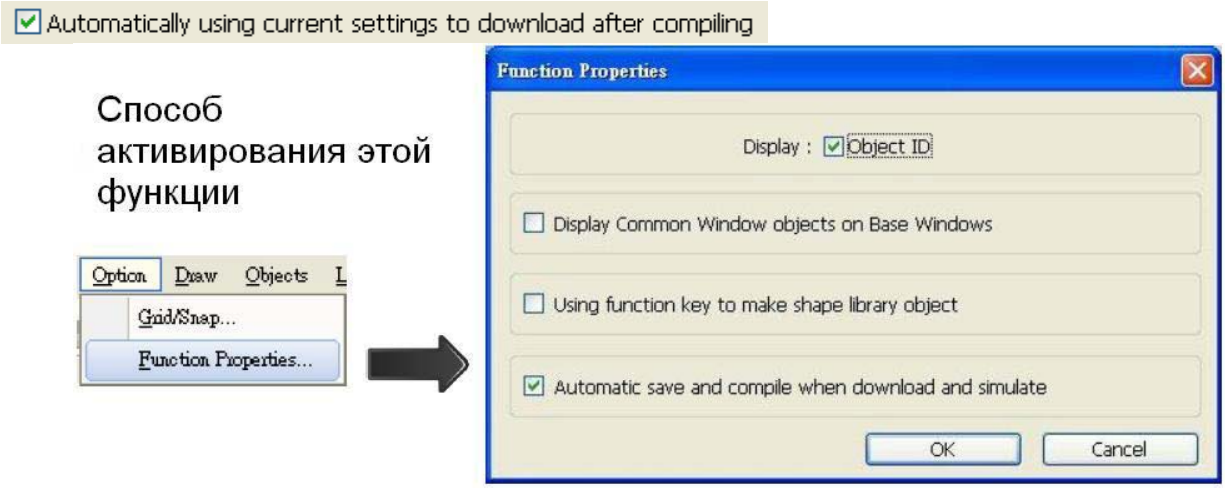
☑ Firmware
Обновление програмного ядра панели.
Обязательно при первой загрузке файлов в панель

☑ Font files
Загрузка шрифтов используемых в проекте

Выбранные файлы будут стерты перед загрузкой

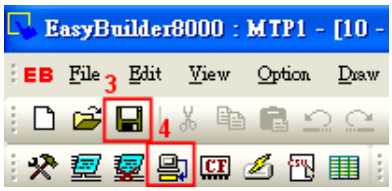
☑ Reboot HMI after download
Панель будет перезапущена после загрузки

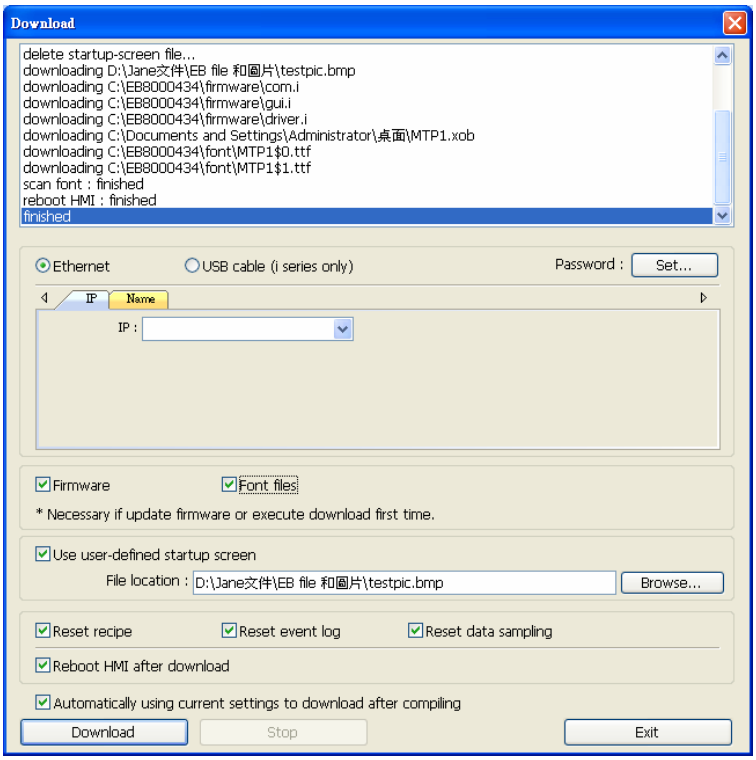
☑ Automatically using current settings to download after compiling
Если сделана эта пометка,система будет загружать проект в соответствии с последними настройками. Смотрите иллюстрации на следующей странице



1. Щелкните по **[Function Properties]**

2. Отметьте **[Automatic save and compile when download and simulate]**





3. Сохраните - **[Save]** проект

4. Щелкните по **[Download] -** Загрузить

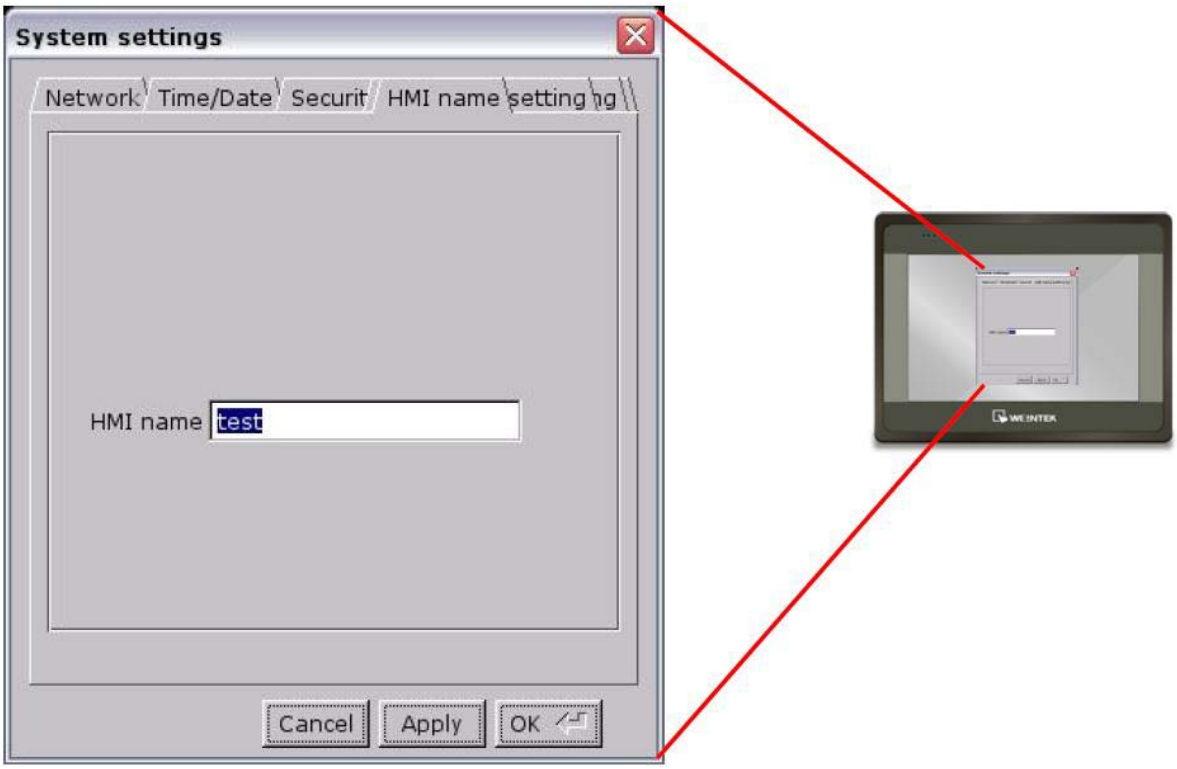
5. В диалоговом окне отметьте **[Automatically using current settings to download after compiling]**

6. Щелкните по **[Download]**

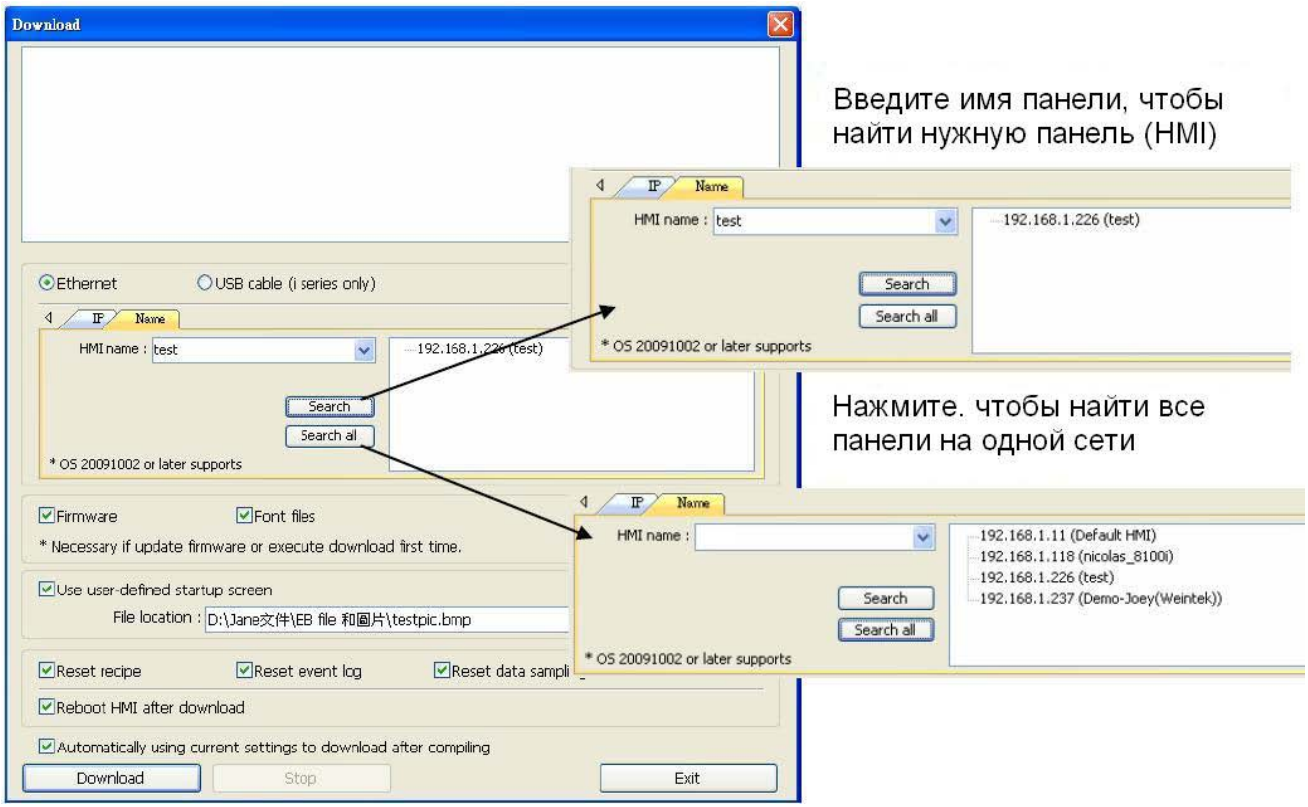
После окончания настроек, при следующем нажатии на [Download], EasyBuilder Pro будет автоматически компилировать и загружать проект в последнюю целевую панель оператора

✹ Способ 2. [Ethernet] / HMI Name:

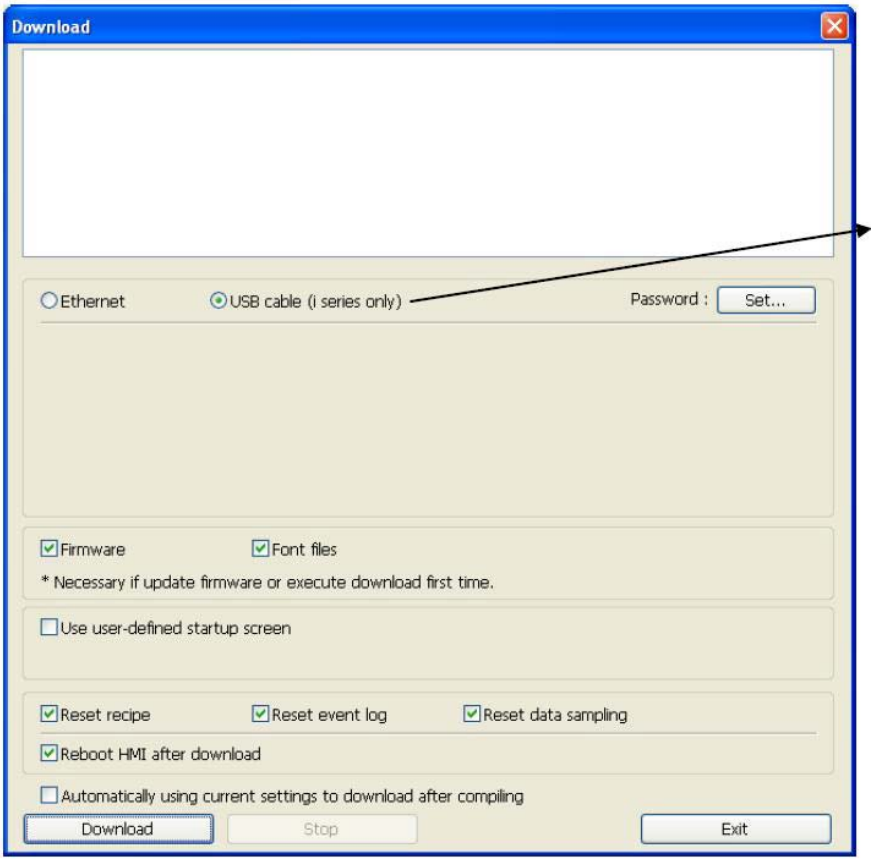
1. Сначала присвойте имя панели (HMI Name).



2. На компьютере выберите установку имени (Name) и начните загрузку



✹ Способ 3. [USB Cable]:

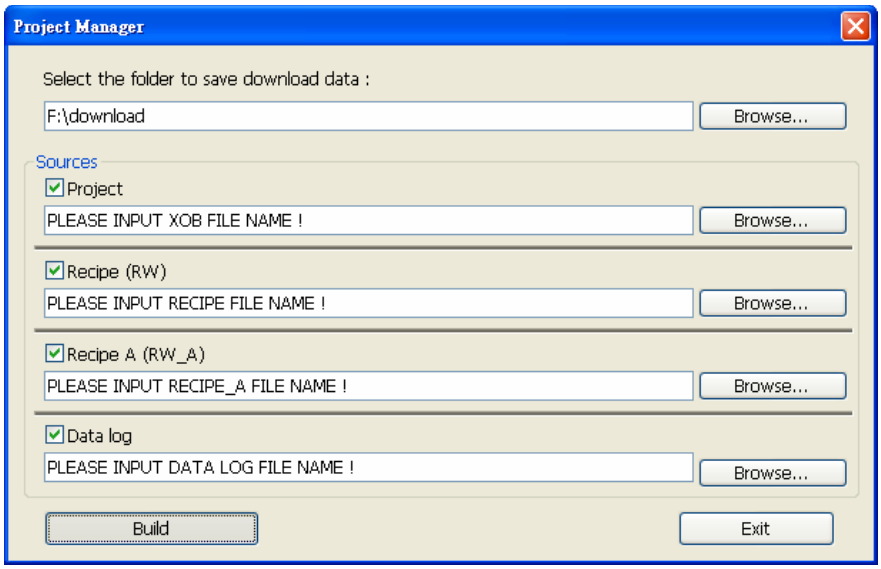


Выберите «USB cable» для загрузки проекта в панель. Способ настройки такой же, как описано в «Способе 1»

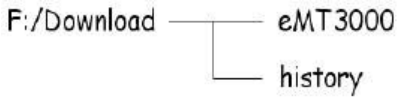
Загрузка проекта по кабелю USB возможно только для панелей серии «i»

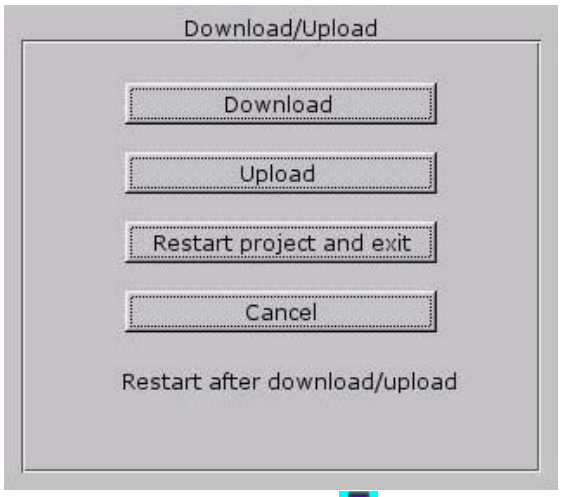
Перед загрузкой проекта по кабелю USB, перейдите к [Computer Management] / [Device Manager], чтобы проверить, установлен ли драйвер USB. Если драйвер не установлен, установите его вручную, используя ЧЗВ на сайте www.weintek.com

✹ Способ 4. [USB Disk / SD Card]:



1. В Utility Manager выберите **[Build Download Data for CF / SD / USB Disk],** чтобы создать данные для загрузки.
   Обычно разделенные на две директории:



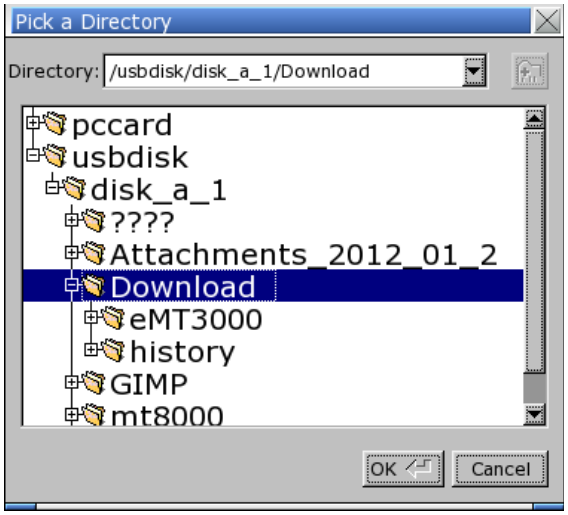


2. Вставьте внешнее устройство в панель

3. Выберите **[Download]** и введите пароль

4. После подтверждения пароля появится директории на внешнем устройстве (pccard: SD/CF Card ; usbdisk: USB Disk)

5. Выберите директорию для хранения проекта и нажмите «OK» для начала загрузки



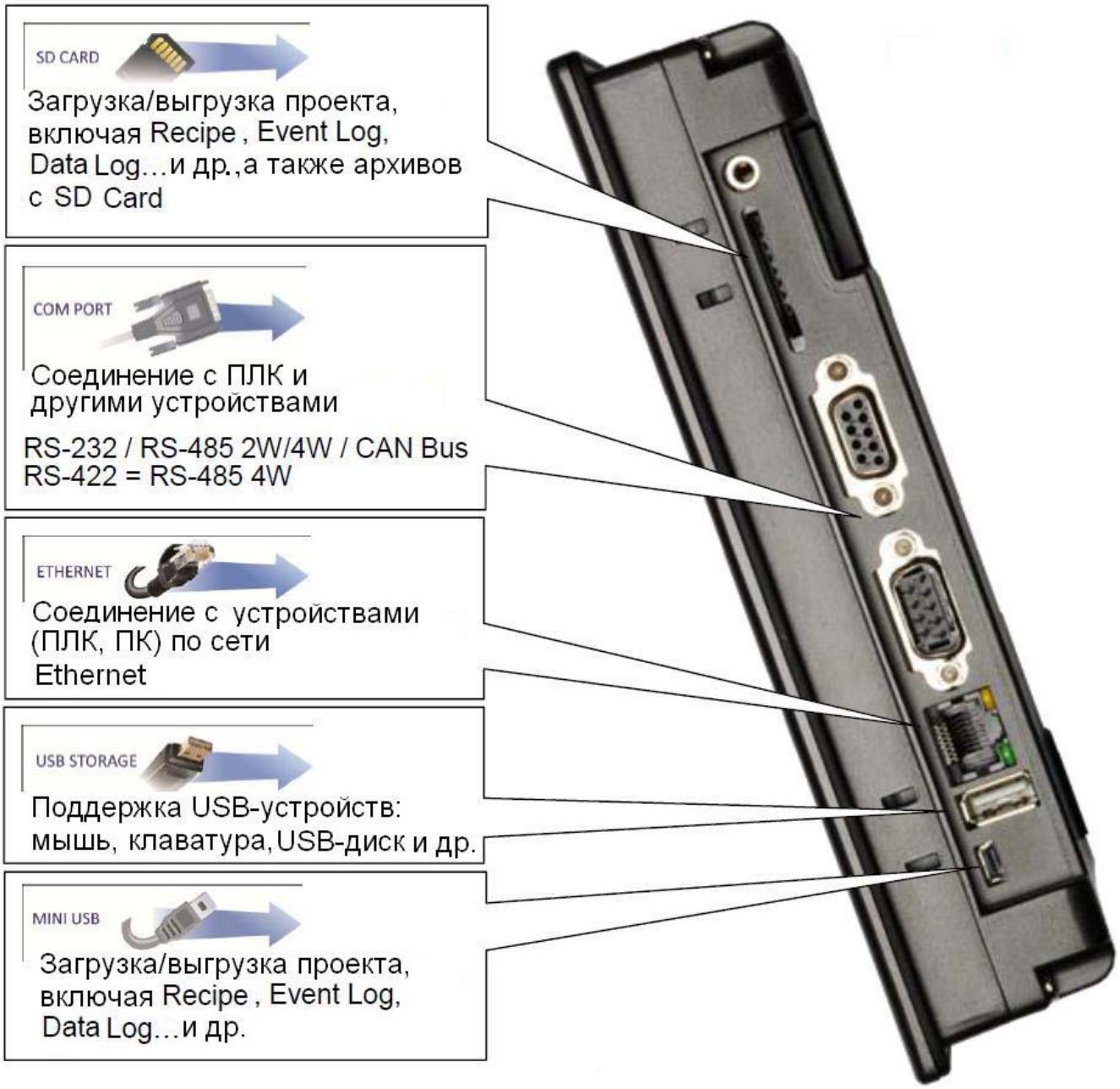
Note

Выберайте ***Верхнюю директорию файла*** при зазгузке.
Для структуры приведенной выше, выберите ***download***, а не ***eMT3000*** или ***history.***

Глава 4. Настройка оборудования панели

Порты входа/выхода панели оператора

Порты Ввода/вывода различны у разных типов панелей.

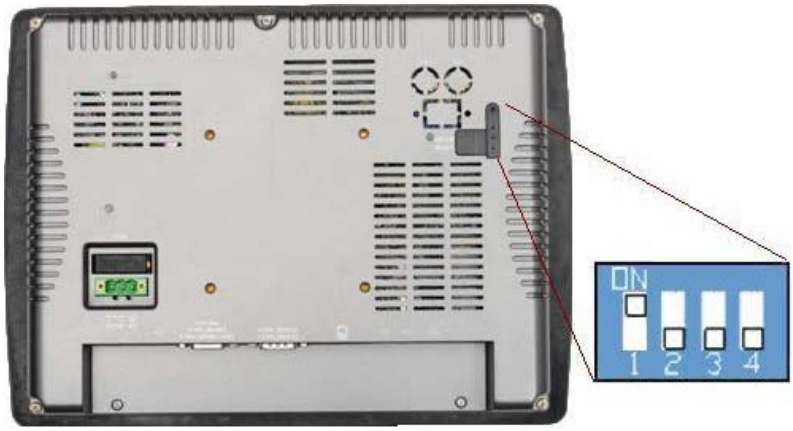


4.2. Системная настройка панели оператора

Перед началом работы с панелью оператора пользователю необходимо сделать системные настройки. После установки панели, создайте пользовательский интерфейс оператора с помощью пакета EasyBuilder Pro.

4.2.1 Перезагрузка системы

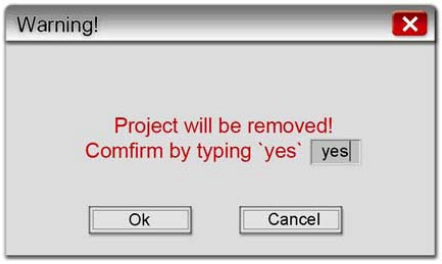
На каждой панели имеется набор DIP-переключателей и кнопка перезагрузки. При использовании DIP-переключателей для изменения режимов работы, задействуются соответствующие функции.



В случае утери пароля пользователь может установить первый DIP-переключатель в положение “ON”, остальные в “OFF” и затем перезапустить панель оператора.
Панель оператора перейдет в режим калибровки сенсорного экрана

Символ “+” появится на экране. Коснитесь центра знака. После выполнения всех 5 касаний, символ “+” исчезнет и параметры сенсорного экрана будут сохранены в системе панели.

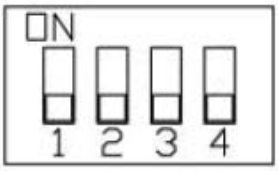
После калибровки появится окно запроса, предлагающее пользователю восстановить пароль по умолчанию, выберите **[Yes].**



Появится другое всплывающее окно (как показано ниже). Пользователю снова нужно подтвердить восстановление пароля по умолчанию вводом **[Yes]** “Да”. Затем нужно щелкнуть **[OK].** Проект и все архивные данные из панели будут удалены. (Пароль по умолчанию: 111111). Тем не менее, другие пароли, включая пароль на загрузку и чтение данных с панели необходимо переустановить).

.

DIP-переключатель

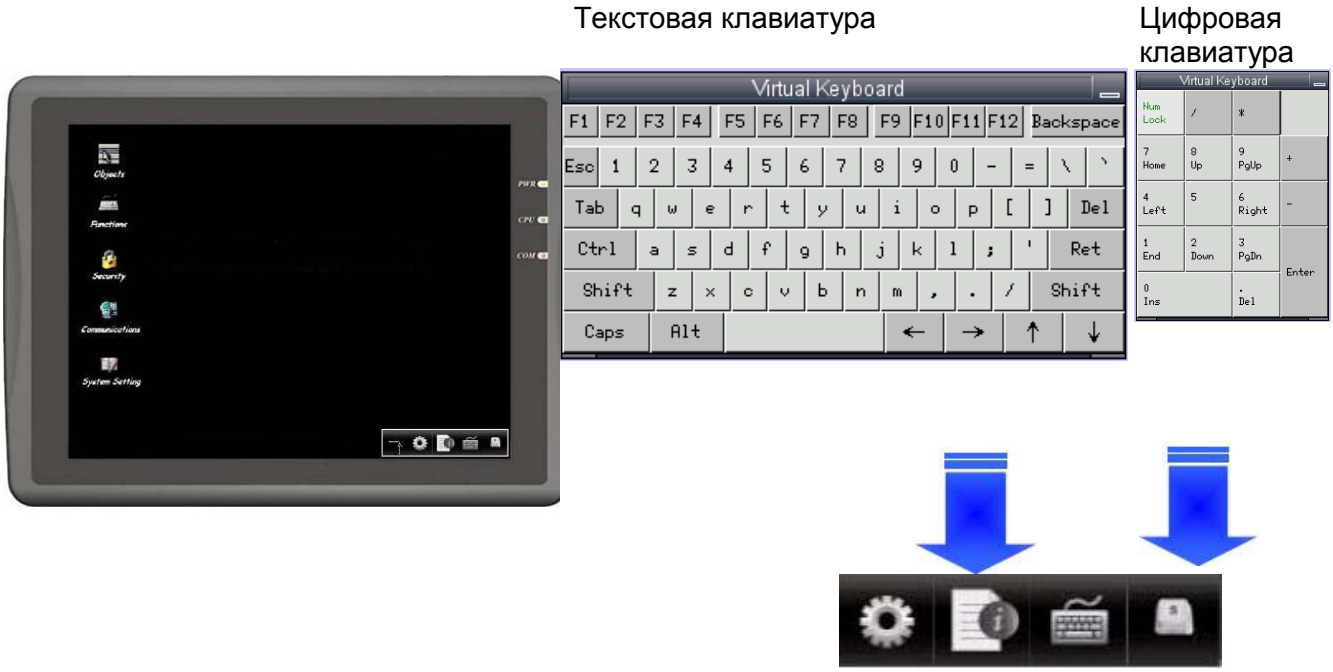


Положения переключателей				Режим
SW1	SW 2	SW 3	SW 4	
ON	OFF	OFF	OFF	Калибровка сенсорного экрана
OFF	ON	OFF	OFF	Спрятать панель инструментов
OFF	OFF	ON	OFF	Загрузка
OFF	OFF	OFF	ON	Резерв
OFF	OFF	OFF	OFF	Нормальная работа

4.2.2. Панель инструментов системы

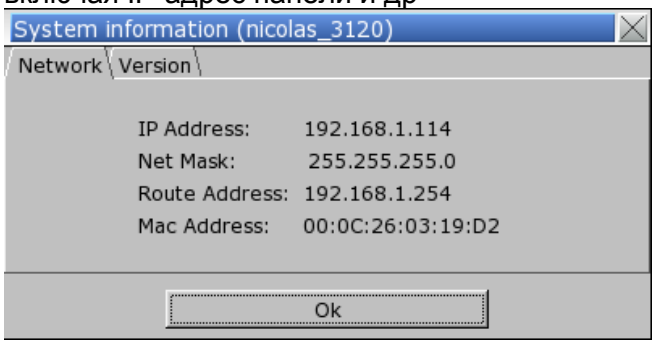
После загрузки панели, пользователь может настроить систему, используя панель инструментов (System Toolbar), расположенную внизу экрана. Для отображения *System Toolbar* на экране нужно нажать точку в правом нижнем углу экрана.

EasyBuilder Pro поддерживает функцию использующую для «отображения/ не отображения» панели системных настроек, системный тэг **[LB-9020]** или положение переключателя**.**
Когда системный тэг **[LB-9020]** в состоянии ON – панель системных настроек отображается на экране. Когда системный тэг **[LB-9020]** в состоянии OFF – панель системных настроек не отображается на экране.
Когда **[DIP Switch 2]** в состоянии ON – панель системных настроек не отображается на экране. Когда **[DIP Switch 2]** в состоянии OFF – панель системных настроек отображается на экране. Чтобы эти настройки сработали, необходимо сделать рестарт панели.
[LB-9020] существует в панелях всех серий. [DIP Switch 2] существует только в панелях серии eMT.

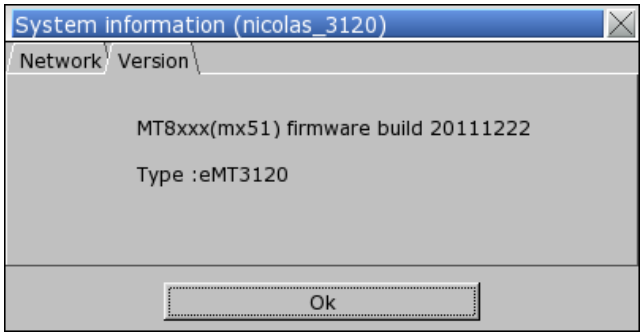


4.2.3 Информация о системе

Закладка **[Network]:** Информация о сети, включая IP-адрес панели и др



Закладка **[Version]:** Информация о версии системы



4.2.4 Настройка системы

Установка или изменение параметров системы. Для начала редактирования необходимо ввести пароль

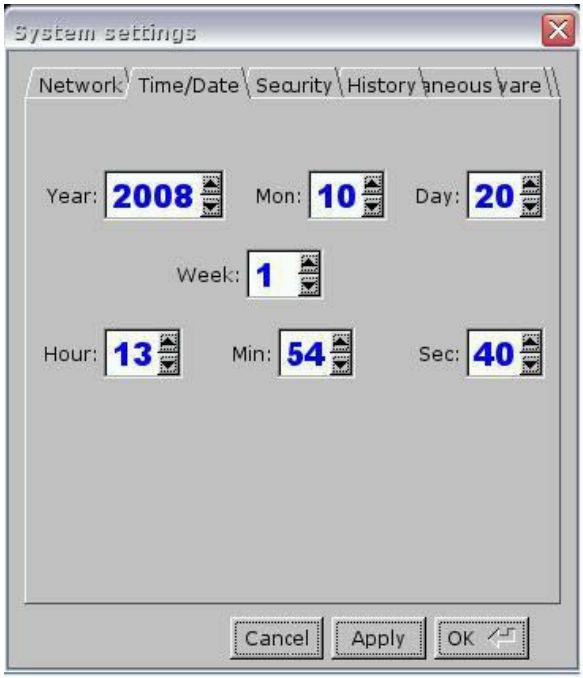
Сеть (Network)
Загрузка проект в панель по сети Ethernet. Необходимо задать IP-адрес панели.
“Auto Get IP Address”. IP-адрес будет назначен автоматически DHCP-сервером.
“IP address get from below”, IP-адрес задают вручную

Защита (Security)
Пароль по умолчанию: 111111.

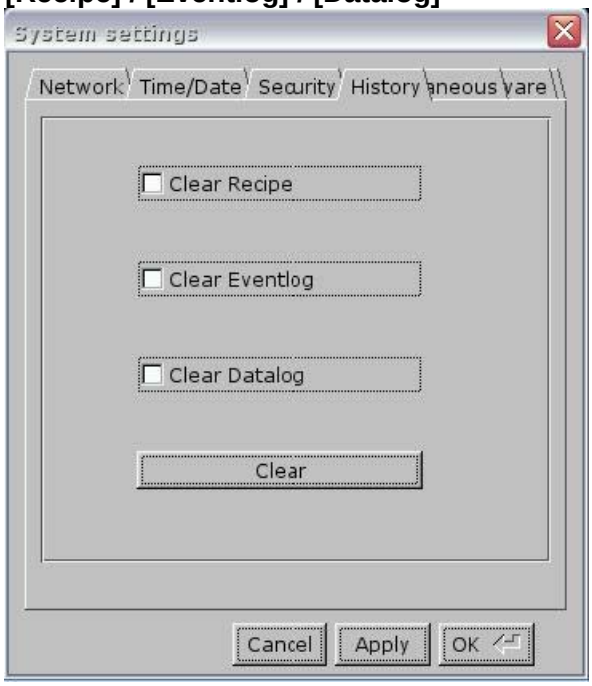
Local Password - Пароль для входа в систему
Upload Password - Пароль для чтения (выгрузки) проекта
Download Password - Пароль для загрузки проекта
Upload (History) Password - Пароль для выгрузки архивных данных

Окно Password confirmation:
(Подтверждение пароля)

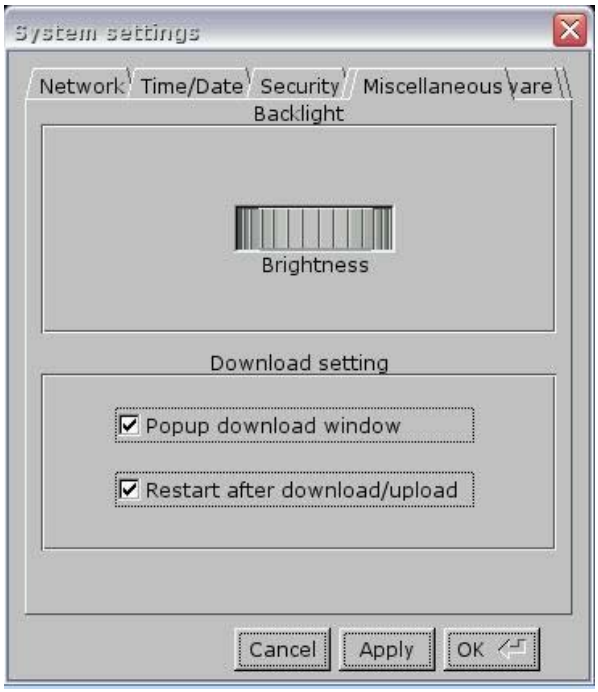
Время/Дата (Time/Date)
Настройка системного времени и даты



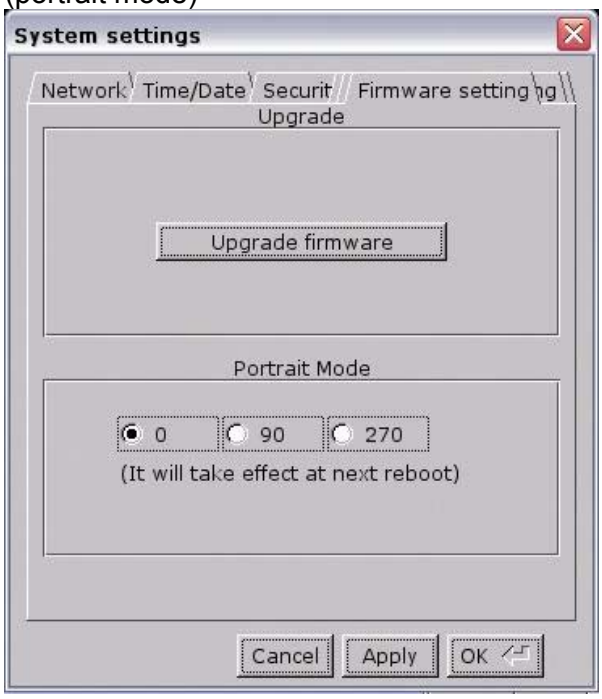
Архив (History)
Инструмент для удаления архивных данных из панели оператора:
[Recipe] / [Eventlog] / [Datalog]



Прочее (Miscellaneous)
Используйте колесико для настройки яркости жидкокристаллического дисплея

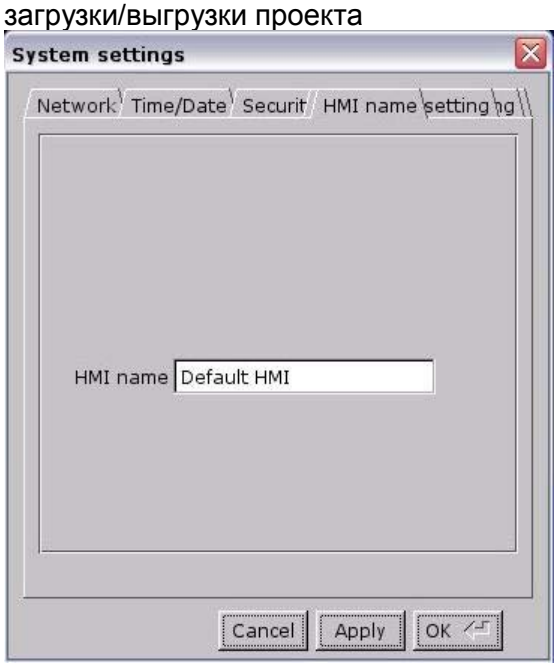


Настройки прошивки (Firmware setting)
Функция обновления прошивки (Upgrade firmware)/ активация режима «Портрет» (portrait mode)

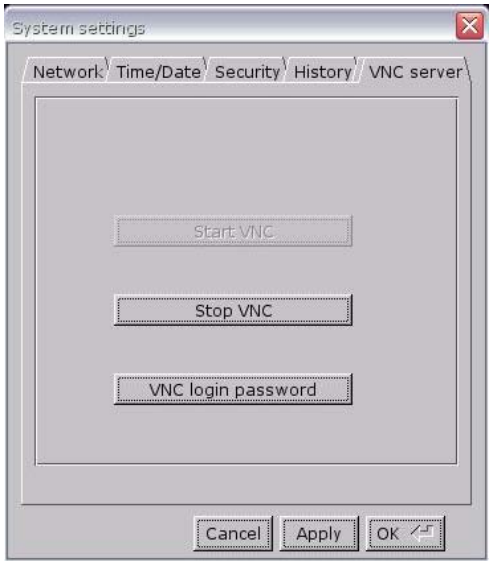


.

Имя панели (HMI name)
Назначьте имя панели для загрузки/выгрузки проекта



Виртуальный сервер (VNC server)



1. Запустите VNC сервер и установите пароль.
2. Установите надстройки Java IE или на компьютере.

3-1. Введите в IE IP-адрес панели оператора: Например,http://192.168.1.28

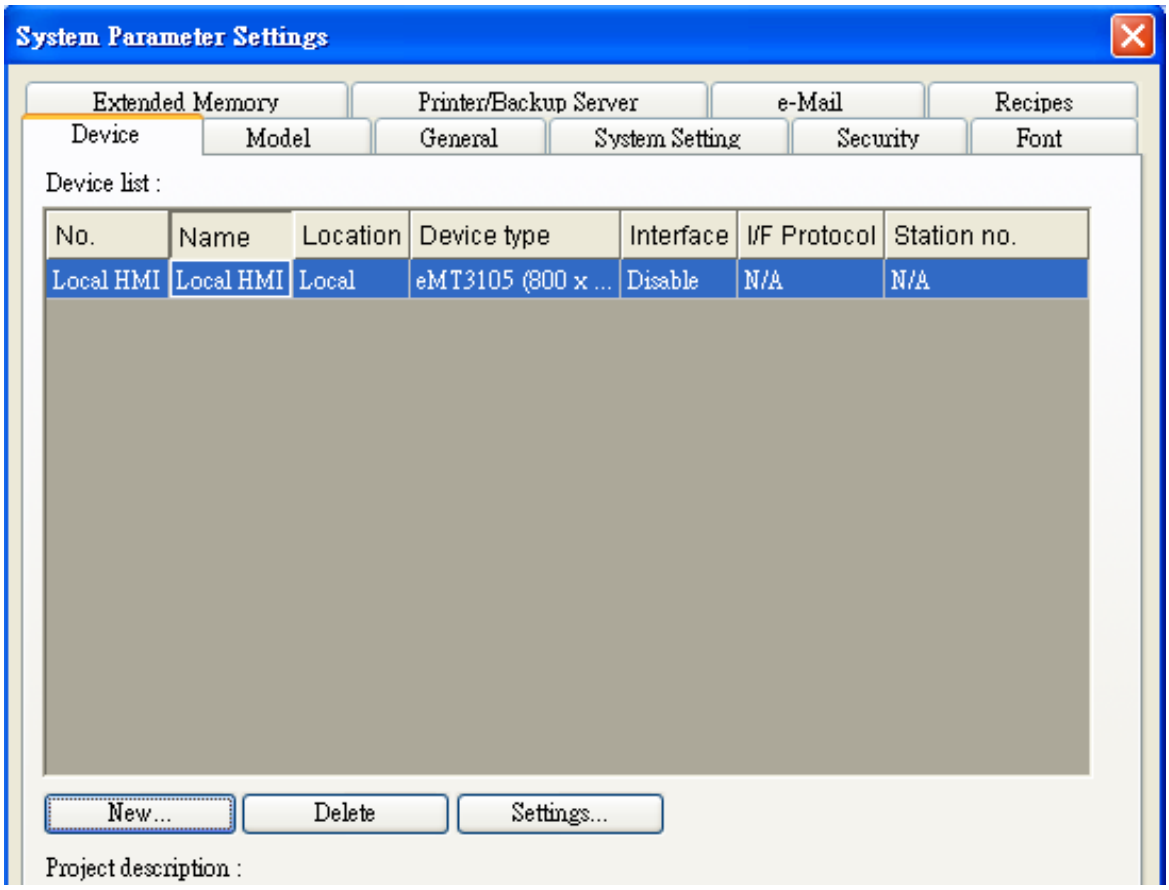
3-2. В VNC viewer введите IP-адрес панели и пароль

✹ В каждый момент времени допустима авторизация только одного пользователя на VNC сервере.

✹ Панель оператора автоматически разорвет соединение с VNC сервером после одного часа простоя

Глава 5. Системные параметры

В EasyBuilder Pro, выберите меню **[Edit] / [System Parameters…]** — появится диалоговое окно **[System Parameter Settings]**:



Системные настройки разделены на несколько групп: **[Device] (Устройство)**, **[Model] (Модель)**, **[General] (Общие)**, **[System Setting] (Системные настройки)**, **[Security] (Учетные записи пользователей)**, **[Font] (Шрифт)**, **[Extend Memory] (Внешняя память) и [Printer/Backup Server] (Сервер печати)** , **[e-Mail] (Почта) и [Recipes] (рецепты) —** которые будут рассмотрены в данной главе.

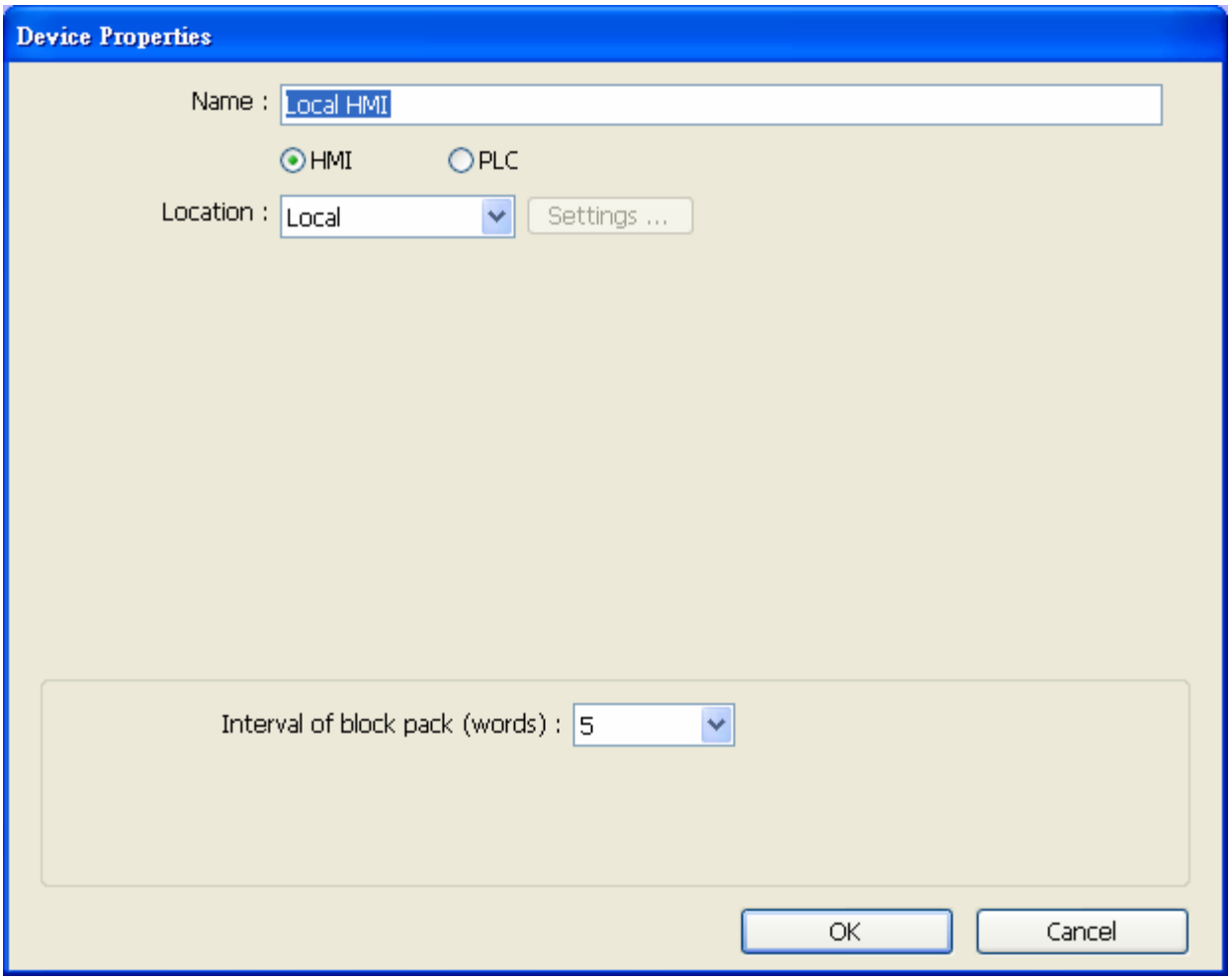
.

5.1 Device (Устройство)

Настройки на закладке **[Device]** определяют все параметры каждого устройства, управляемого панелью оператора. Этими устройствами могут быть: контроллер, другая удаленная панель оператора или компьютер.

После открытия нового файла «*. mtp» в EasyBuilder Pro в списке устройств **[Device List]** по умолчанию находится “Local HMI”. Эта “Local HMI”, используется для определения рабочей панели оператора, что означает, что каждый файл «*. mtp» должен содержать по крайней мере одну “Local HMI” в списке **[Device List]**.

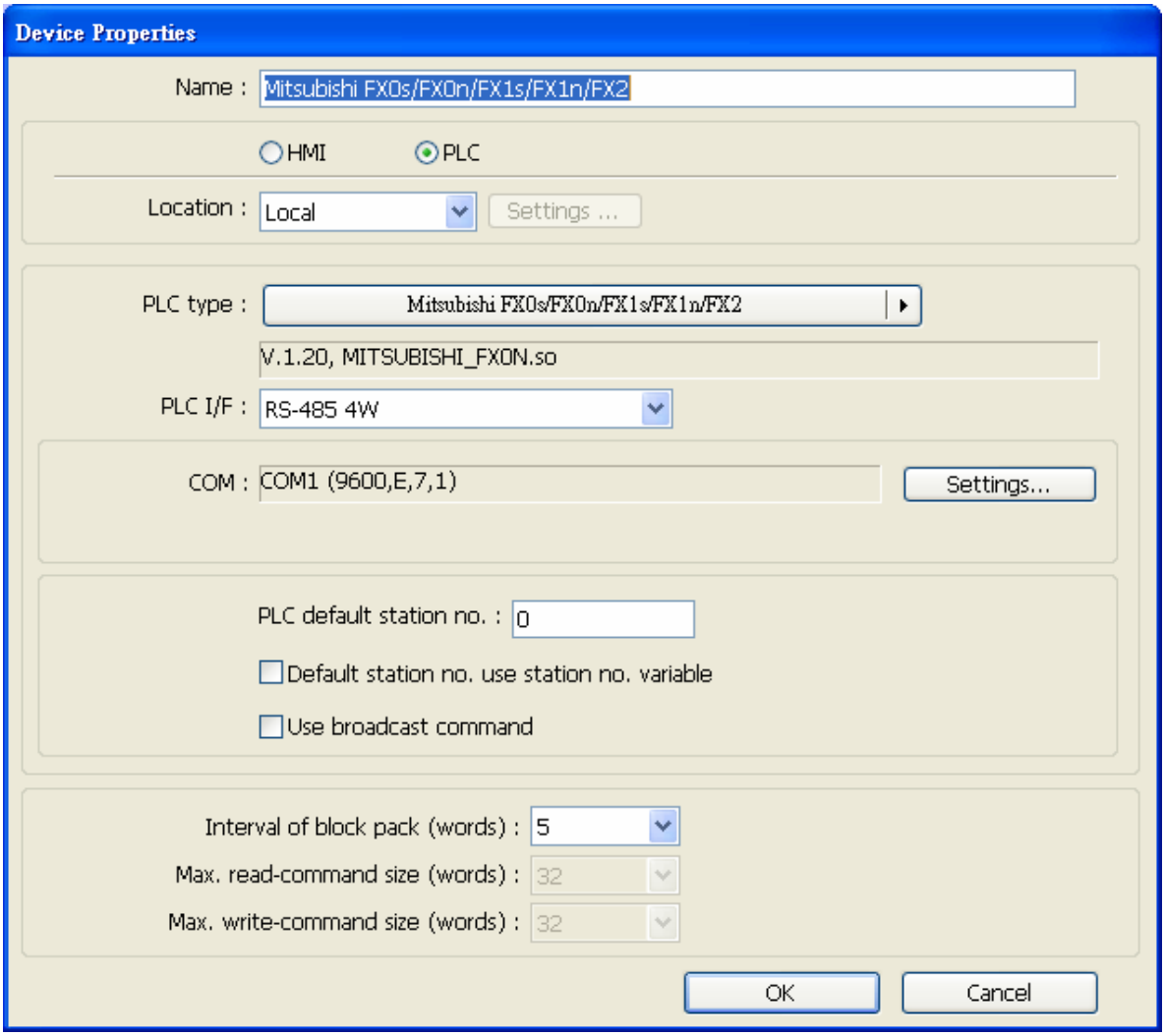
Выберите **[Settings]** в списке устройств, появится окно **[Device Properties]** – свойства устройства, показанное ниже. Из него мы узнаем, что атрибут “Local HMI” это “HMI”- панель и ее расположение “Local” – местное.



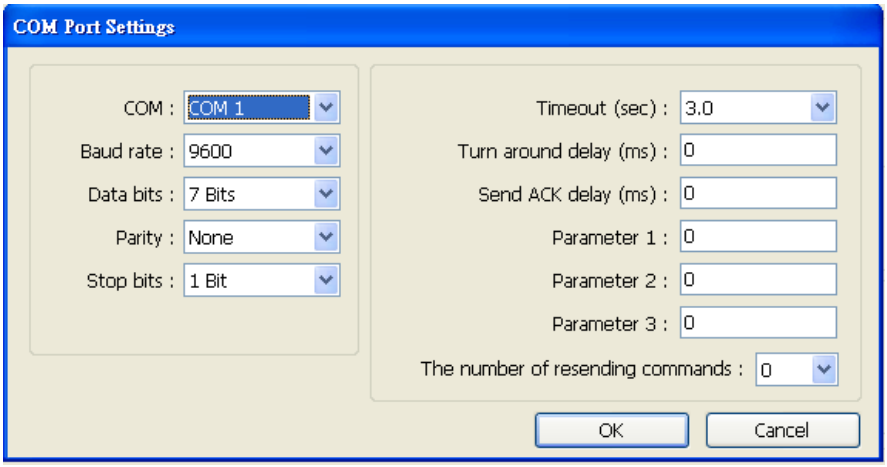
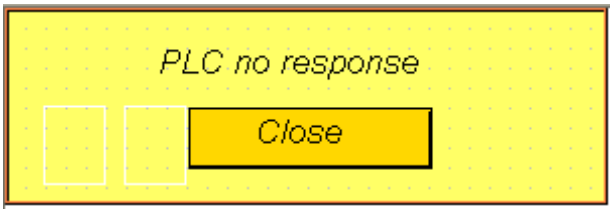
5.1.1 Управление локальным ПЛК

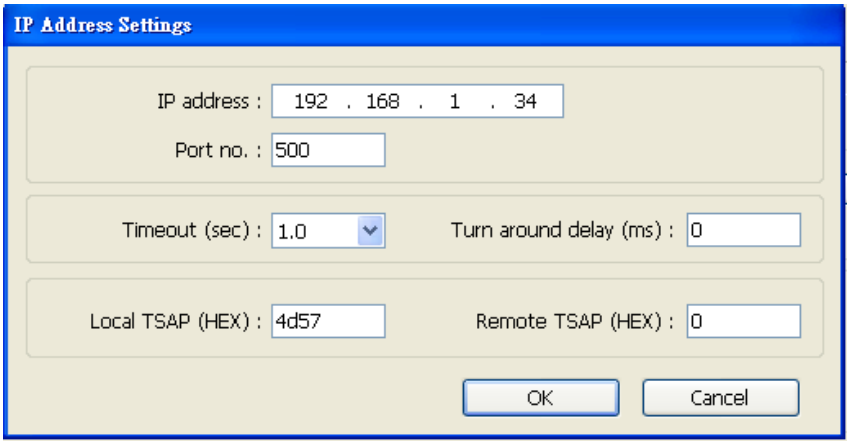
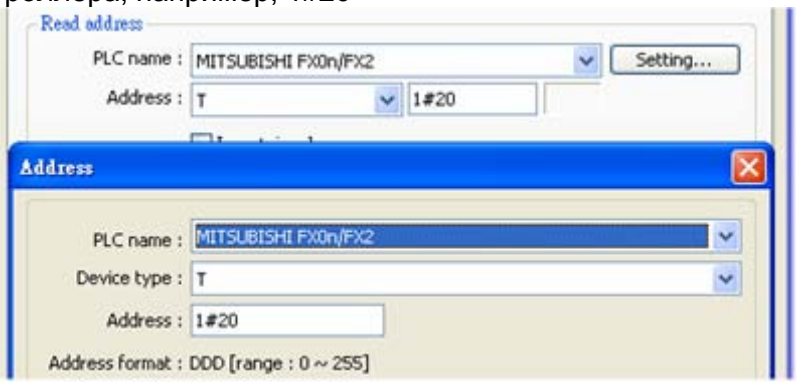
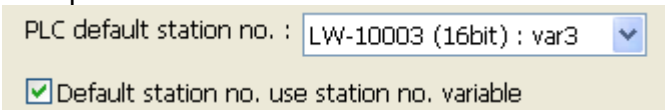
Термин “local PLC” (локальный ПЛК) означает программируемый логический контроллер, подключаемый непосредственно к панели оператора. Для управления локальным ПЛК пользователю необходимо добавить это устройство в список. Щелкните [New…] и появится окно [Device Properties]. Требуется правильно указать все атрибуты.

Ниже приведен пример для локального ПЛК модели MITSUBISHI FX0n/FX2:



Настройка	Описание
Name	Имя устройства
HMI or PLC	Тип устройства. В данном случае нужно выбрать **[PLC].**
Location	Расположение устройства. Выберите **[Local]** в данном случае.
PLC type	Тип контроллера. Выберите MITSUBISHI FX0n/FX2 (для данного примера)

Настройка	Описание
PLC I/F (Интерфейс ПЛК)	Доступны следующие интерфейсы связи: **[RS-232], [RS-485 2W], [RS-485 4W], [Ethernet], [USB].** Если выбран один из интерфейсов: [RS-232], [RS-485 2W], [RS-485 4W] – щелкните **[Settings…]** — появится диалоговое окно **[Com Port Settings]** (Настройки COM-порта). Пользователю необходимо корректно задать параметры COM-порта.  **[Timeout]** Если соединении ПЛК – панель разъединено дольше, чем указанный временной интервал **[Timeout],** на панели всплывает окно № 5 с сообщением “PLC No Response” (ПЛК не отвечает).  **[Turn around delay]** Интервал временной задержки между посылкой следующей команды в ПЛК (после отправки предыдущей) определяется параметром **[Turn around delay]**. При отсутствии специальных требований, значение по умолчанию равно «0». Если ПЛК – это контроллер **SIEMENS S7-200**, этот параметр надо задать **“5”** и **[Parameter 1] “30”.** Если выбран интерфейс **[Ethernet]**, кликните **[Settings…]** — появится окно **[IP Address Settings]**. Пользователю необходимо правильно указать IP-адрес и номер порта контроллера (см. рисунок на следующей странице).

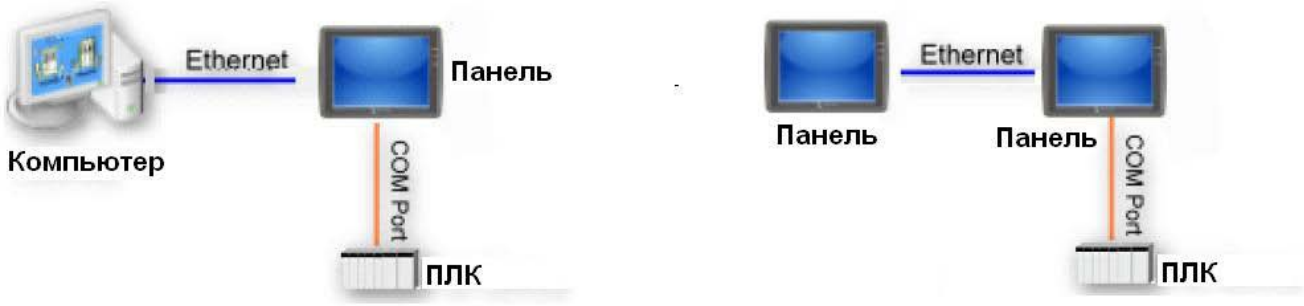
Настройка	Описание
	 Если выбран интерфейс **[USB]**, никакие дополнительные настройки не нужны. Проверьте корректность настроек в **[Device Properties]**
PLC default station no.	Если адрес устройства контроллера не содержит номера станции (**station no.**), то EasyBuilder Pro будет использовать в качестве номера станции параметр, заданный по умолчанию «PLC default setting no. Station». Кроме того, номер станции ПЛК может быть включен непосредственно в адрес контроллера, например, 1#20  «1» означает номер станции ПЛК, и это число должно быть больше 0 и меньше 255. «20» означает адрес ПЛК, символ «#» является разделителем номера станции и адреса устройства.
Default station no. use station no. variable	При настройке номера станции можно выбрать переменную - (variable) и использовать ее в качестве [PLC default station no.]. LW10000~LW10015 могут быть использованы для настройки «station no. Variable» При использовании этой функции, если «station no.» не указан для данного адреса ПЛК, необходимо выбрать «default station no.» из «default station no.» В примере: параметр «var3» использован как «default station no.».  Далее будет показано, как установить Адрес ПЛК и «station no.».

Настройка	Описание
	a. Номер станции ПЛК – «5» PLC name : MODBUS RTU Address : 4x 5#111 **b.** Номер станции ПЛК – переменная «var7». (LW-10007) PLC name : MODBUS RTU Address : 4x var7#111 **c.** Номер станции ПЛК – «111», а номер станции не задан. Номер станции по умолчанию использует переменную «var3». Номер станции ПЛК определен переменной «var3» - (LW-10003). PLC name : MODBUS RTU Address : 4x 111
Use broadcast command	Настройка на использование широковещательных команд этой станцией. Например, установите номер 255. По адресу 255#200 панель оператора отправит команду всем контроллерам, но будет игнорировать ответы ПЛК (это работает только на Modbus) ☑ Use broadcast command Broadcast station no. : 255
Interval of block pack (words)	Если разница адресов, по которым находятся данные, используемые разными командами, не превышает это значение, то такие команды объединяются в одну. Но функция комбинации команд не действует, если заданное значение параметра равно 0. Например, установим значение параметра 5 и предположим, что пользователю нужно считать 1 слово, находящееся по адресу LW3 и 2 слова, начиная с адреса LW6 соответственно. Так как разница между указанными адресами LW3 и LW6– меньше 5, то такие команды считывания можно объединить в одну. Выходными данными этой комбинированной команды будут 5 слов расположенных по 5 адресам, начиная с LW3. (LW3~LW7). **Примечание:** Максимальный объем данных комбинации команды не должен превышать значение **[Max. read-command size]**
Max. read-command size (words)	Максимальный размер данных, считываемых из устройства одновременно. Единица измерения: слово.
Max. write-command size (words)	Максимальный размер данных, записываемых в устройство одновременно. Единица измерения: слово.

После завершения всех настроек новый объект “Local PLC 1” появится в списке устройств.

No.	Name	Location	Device type	Interface	I/F Protocol	Statio
Local HMI	Local HMI	Local	eMT3105 (800 x ...	Disable	N/A	N/A
Local PLC 1	Device 1	Local	Mitsubishi FX0s...	COM 1 (9600,N,7,1)	RS485 4W	0

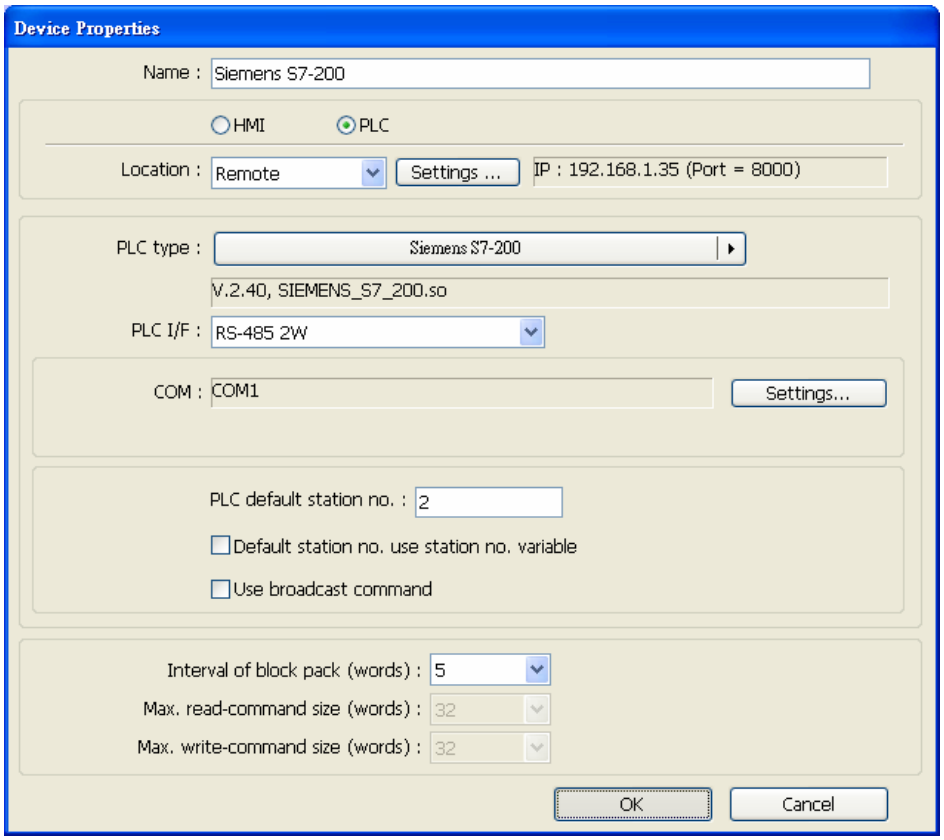
5.1.2 Управление удаленным ПЛК



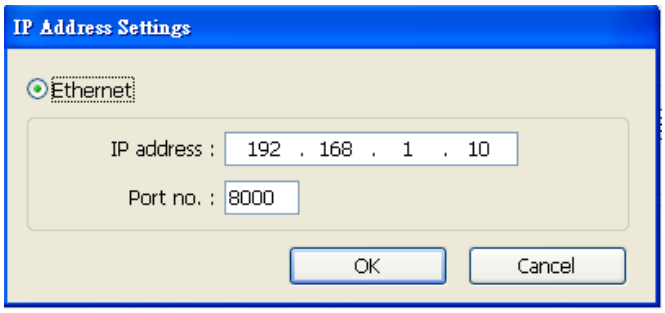
Термин “remote PLC” (удаленный ПЛК) означает контроллер, подключенный к удаленной панели оператора. Для управления удаленным ПЛК пользователю нужно добавить такой тип устройства.

Щелкните **[New…]** в списке **[Device List]** и появится диалоговое окно **[Device Properties]** (Параметры устройства). Требуется указать корректные значения всех параметров.

Далее рассмотрен пример с подключением удаленного ПЛК SIEMENS S7-200



Настройка	Описание
HMI or PLC	Тип устройства. В данном случае нужно выбрать **[PLC].**

Настройка	Описание
Location	Для данного примера выберите вариант **[Remote]** (Удаленный) и укажите IP-адрес удаленной панели оператора, для которой устанавливается соединение с контроллером SIEMENS S-200. Щелкните [Settings…] для задания адреса удаленной панели: 
PLC Type	Тип контроллера. В данном примере это контроллер SIEMENS S7-200.
PLC I/F	Указывает, какой интерфейс используется в удаленном контроллере. При использовании COM порта, надо выбрать из **[RS-232]**, **[RS-485 2W]**, и **[RS485 4W].**
PLC default station no.	Определяет, какой номер станции по умолчанию использует удаленный ПЛК.
COM	Указывает, какой COM-порт используется в удаленном контроллере для соединения с панелью.

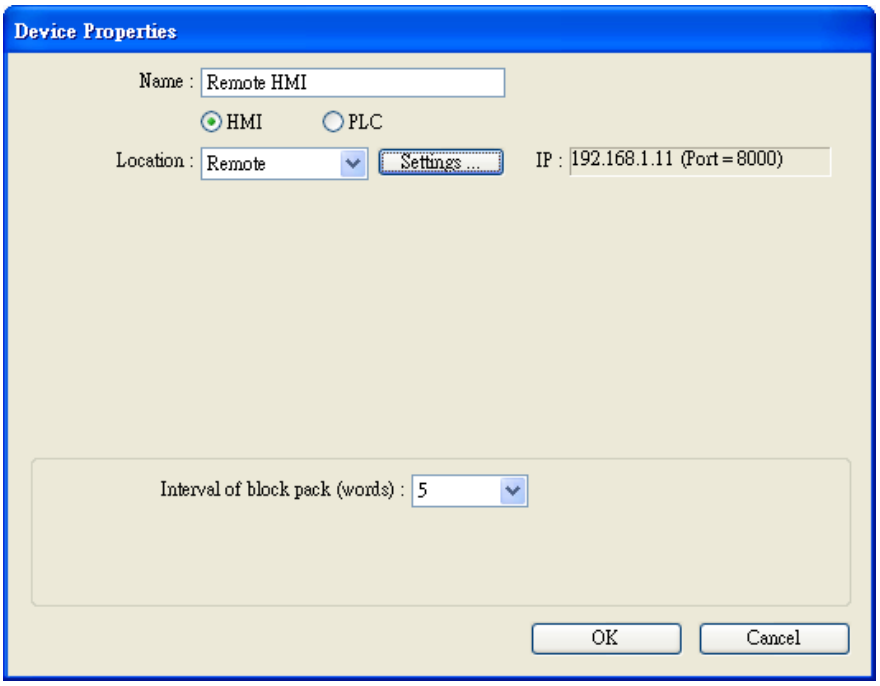
После завершения всех настроек в списке устройств появится новый объект “Remote PLC”.

Device list :

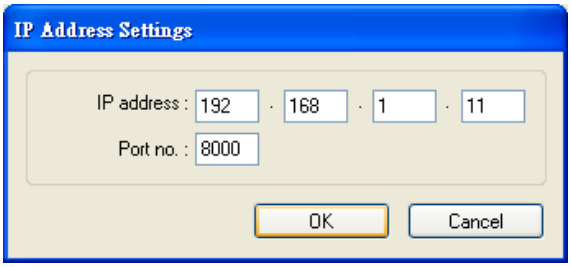
No.	Name	Location	Device type
Local HMI	Local HMI	Local	eMT3105 (800 x ...
Local PLC 1	Device 1	Local	Mitsubishi FX0s...
Remote PLC 1	Siemens S7-200	Remote (IP:192.168.1.10, Port=8000, COM 1)	Siemens S7-200

5.1.3 Управление удаленной панелью оператора

Термин “remote HMI” (удаленная панель оператора) означает, что эта панель оператора, управляет через сеть другая панель или компьютер, выполняющий имитацию проекта в режиме он-лайн. Для управления удаленной панелью оператора пользователю нужно добавить в список устройство данного типа. Щелкните **[New…]** в списке **[Device List]** и откроется диалоговое окно **[Device Properties]** (Параметры устройства). Требуется указать корректные значения всех параметров.



Настройка	Описание
HMI or PLC	Тип устройства. В данном случае **[HMI]** (панель оператора)

Настройка	Описание
Location	В данном случае выбирают **[Remote]** (удаленная), и щелкают **[Settings…]** для задания IP-адреса удаленной панели и номера порта. Номера порта **[Port no.]** удаленной панели можно увидеть в **[Model]** в **[System parameters] сразу** после открытия файла «* .mtp» удаленной панели Номера порта и удаленной и локальной панелей должны совпадать между собой. 

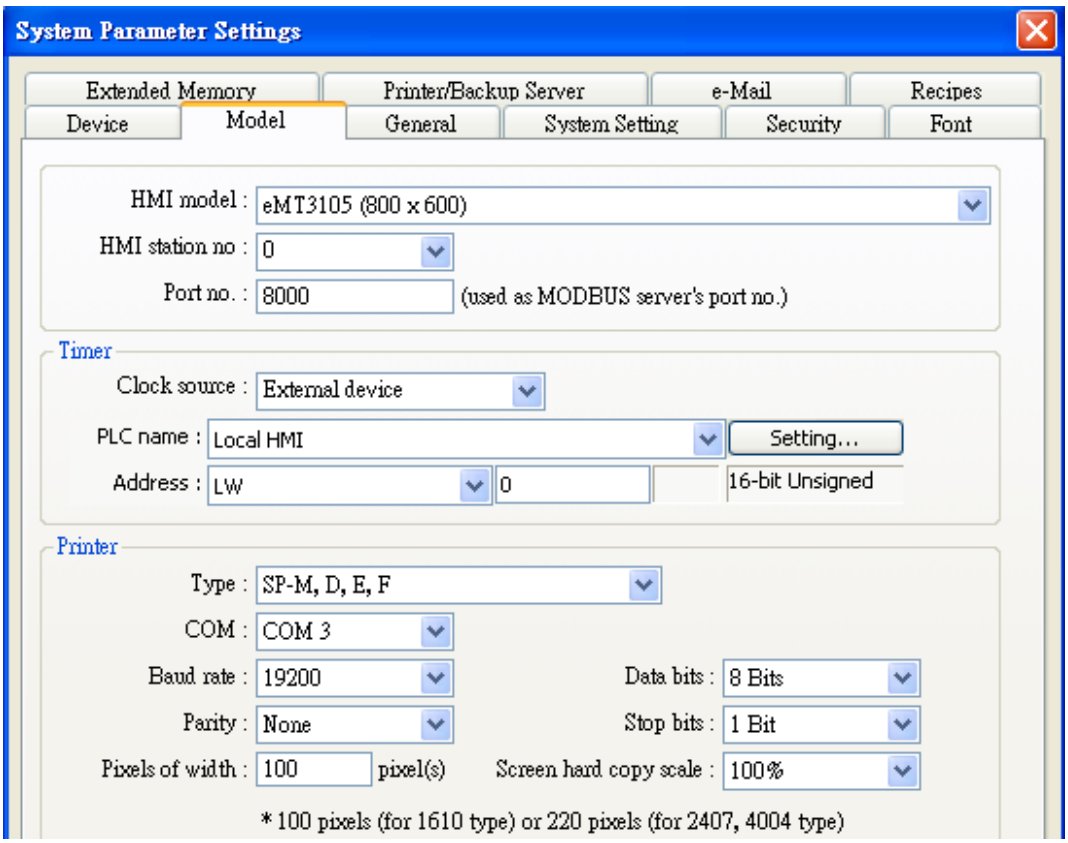
После завершения всех настроек в списке устройств появится новый объект “Remote HMI”.

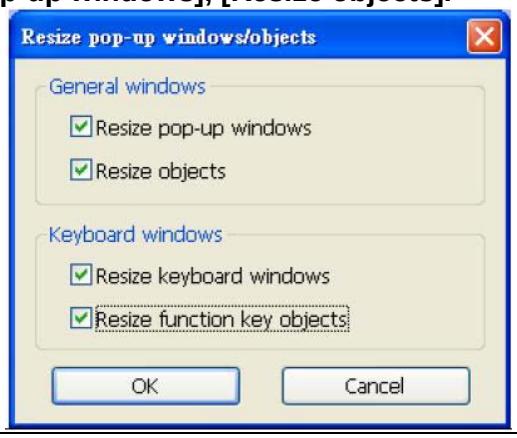
Device list :

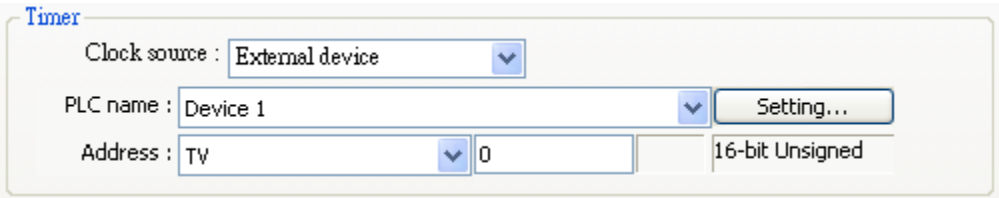
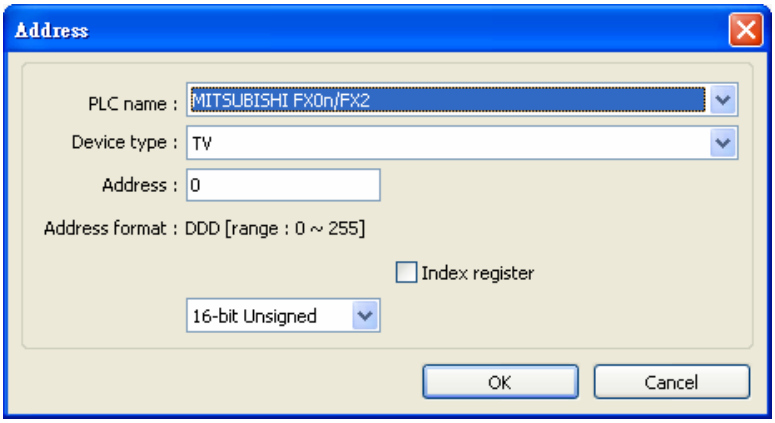
No.	Name	Location	Device type
Local HMI	Local HMI	Local	eMT3105 (800 x ...
Local PLC 1	Device 1	Local	Mitsubishi FX0s...
Remote PLC 1	Siemens S7-200	Remote (IP:192.168.1.10, Port=8000, COM 1)	Siemens S7-200
Remote HMI 1	Device 3	Remote (IP:0.0.0.0, Port=8000)	MT8xxx

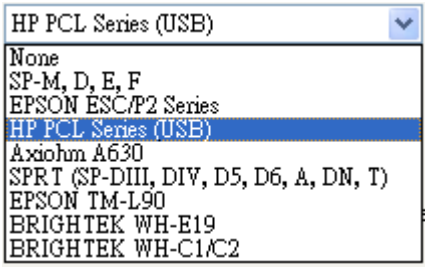
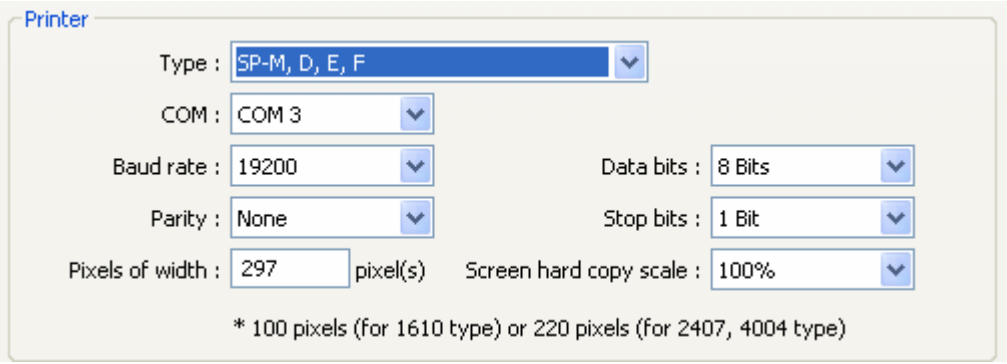
5.2 Model (Модель панели оператора)

Параметры на закладке **[Model]** определяют типы панели оператора, и настройки таймера **[Timer]** и принтера **[Printer]**.



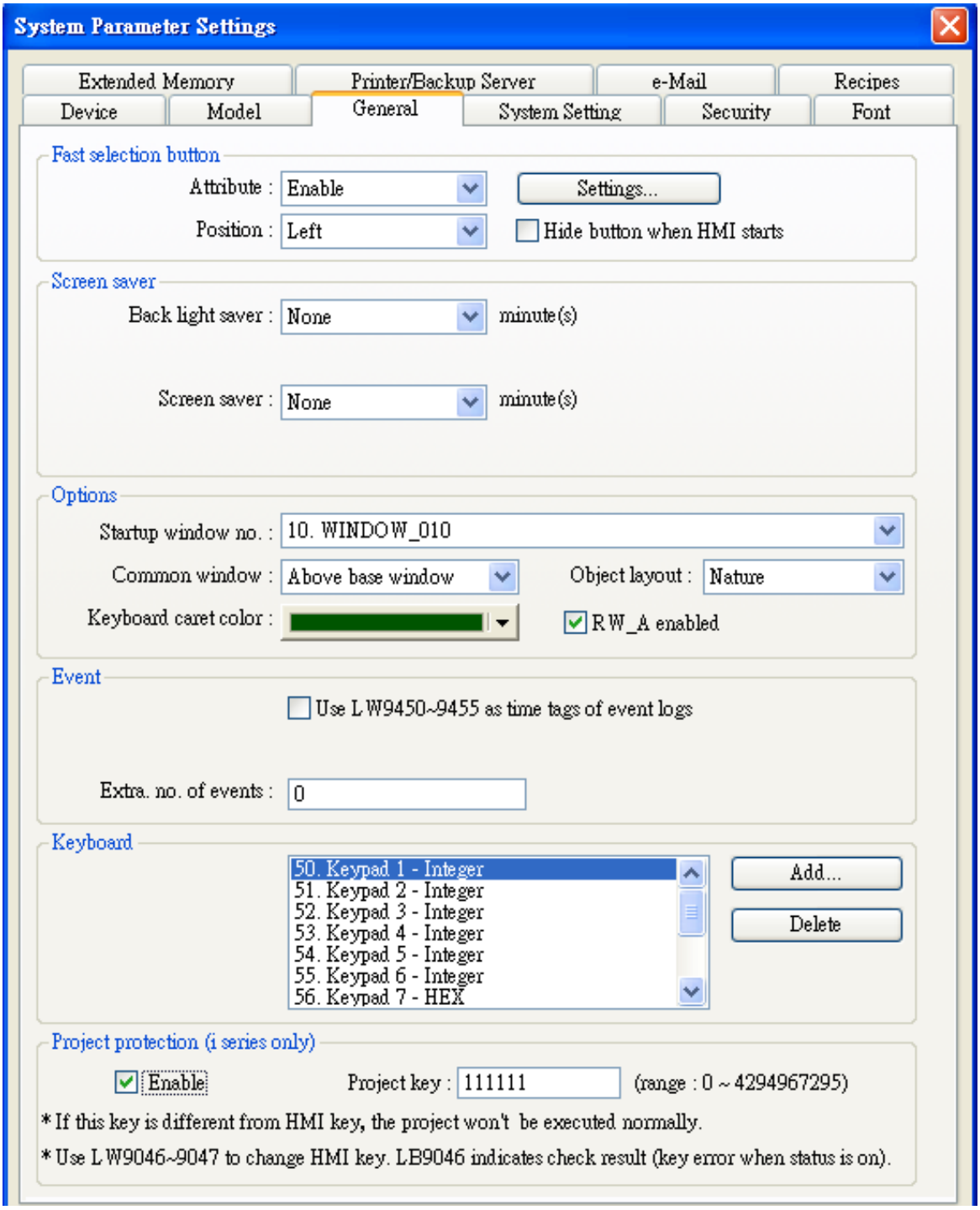
Настройка	Описание
HMI model	Выберите тип рабочей панели оператора из списка: eMT3105 (800 x 600) eMT3105 (800 x 600) eMT3120 (1024 x 768) eMT3070 (800 x 480) После изменения тип панели **[Model],** пользователю доступно изменение размеров всплывающих окно или объектов при настройке параметров проекта **[Resize pop-up windows], [Resize objects].** 

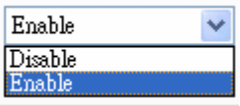
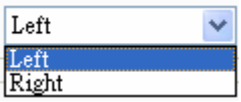
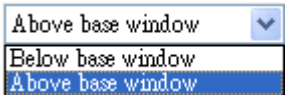
Настройка	Описание
HMI station no.	Установите номер станции панели оператора **[HMI station no.]**. При отсутствии специфических требований, выберите значение по умолчанию.
Port no.	Установите номер порта панели оператора **[Port no.]**. Он используется в качестве номера порта сервера сети MODBUS. При отсутствии специфических задач, выберите значение по умолчанию.
Timer	**[Clock source]** (Источник настройки времени) Задайте устройство, предоставляющее сигнал для объектов использующих текущее время.. Время используется функциями **[Data Sampling]** (Опрос данных), **[Event Log]** (Запись событий) и др. объектами, которым необходима метка времени. a. **[HMI RTC]** – этот вариант означает, что используется сигналы времени от внутренних часов панели. b. **[External device]** – этот вариант означает, что используется сигнал времени от внешнего устройства. Необходимо корректно задать адрес источника времени. Ниже приведен пример: “TV” выдает показания времени с локального ПЛК. Данные, расположены по 6 последовательным адресам, начиная с нулевого означают: TV 0 → Секунды (диапазон: 0 – 59) TV 1 → Минуты (диапазон: 0 – 59) TV 2 → Час (диапазон: 0 – 23) TV 3 → День (диапазон: 0 – 59) TV 4 → Месяц (диапазон: 1 – 12) TV 5 → Год (диапазон: 1970 – 2037)  

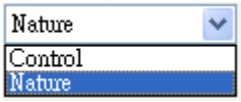
Настройка	Описание
Printer	**[Type]** Показывает поддерживаемые принтеры. Для принтеров моделей HP PCL должен использоваться USB-интерфейс, в то время как для принтеров других фирм необходимо использовать COM-порт. За подробностями обращайтесь к главе 23 (Принтеры поддерживаемые панелями оператора)  При использовании COM-порта параметры соединения должны быть точно заданы. При выборе принтеров типа **SP-M, D, E, F** необходимо точно устанавливать **[pixels of width]** (ширину в пикселях), т.е. задаваемое значение не должно превышать значение по умолчанию для принтера. Иначе результат печати будет неправильный. 

5.3 General (Общие параметры)

Параметры на закладке **[General]** определяют все настройки, связанные с работой экрана.

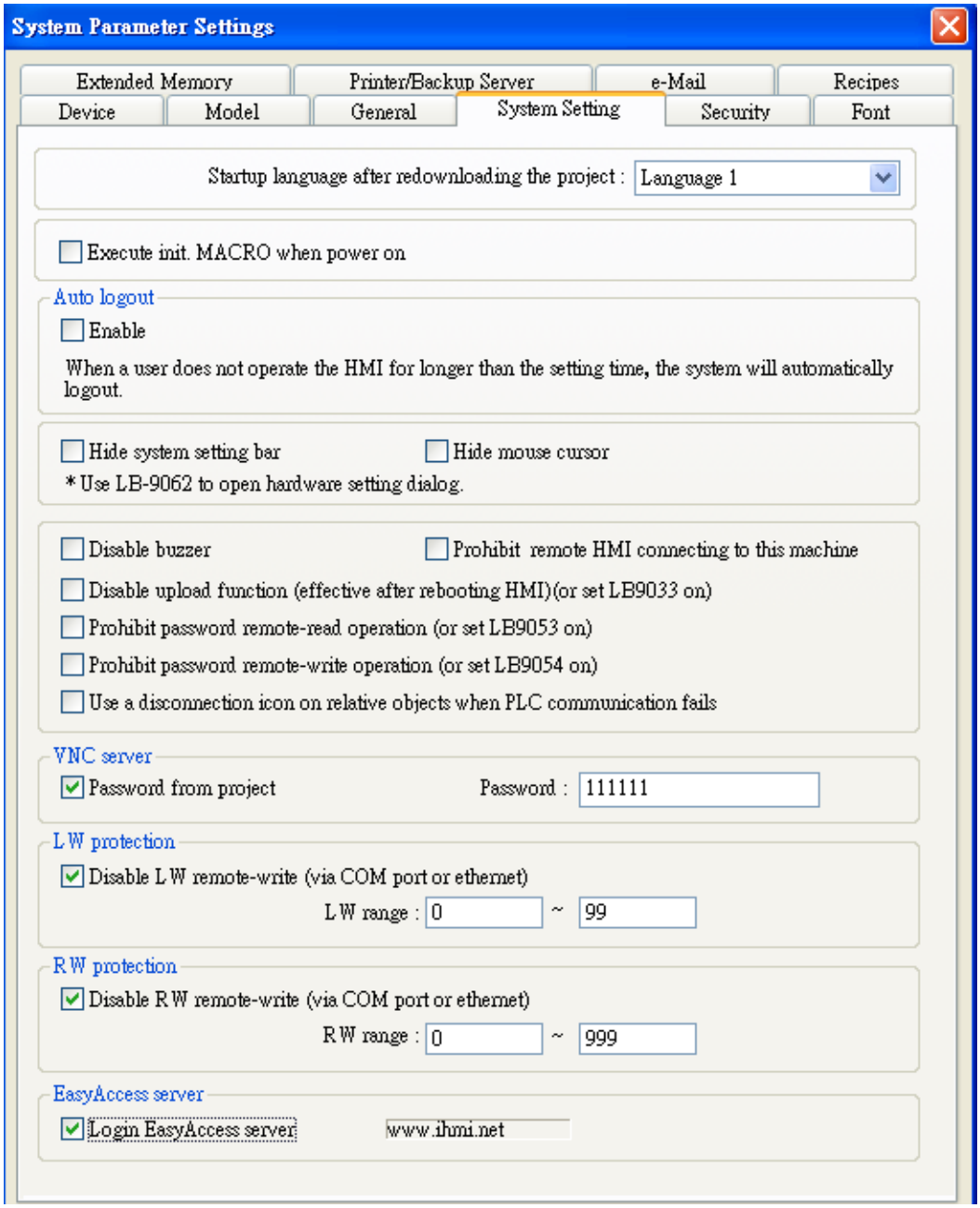


Настройка	Описание
Fast selection button	Определяет параметры всех атрибутов для кнопок быстрого выбора, которые обозначены как окна типа 3. **[Attribute]**  Подключите или отключите функцию окна быстрого выбора. Выберите вариант “Enable” и нажмите [Settings…] для установки свойств кнопки вызова окна, включая цвет и текстовую надпись. **[Position]**  Определите место расположения кнопки быстрого выбора. Если выбран вариант “Left”, то кнопка будет отображаться в левом нижнем углу экрана; если “Right” — то в правом нижнем углу.
Screen saver	**[Back light saver]** Если длительность отсутствия работы с экраном равна этому значению, то подсветка экрана отключается. Единица измерений: минуты. Подсветка экранавключается сразу при возобновлении работы (при прикосновении к экрану). Если выбран вариант **[none]**, то подсветка эрана будет включена всегда. **[Screen saver]** Если длительность отсутствия работы с экраном равна этому значению, то текущая экранная форма автоматически переключается на окно, назначенное в **[Saver window no.].** Единица измерения: минуты. Если выбран вариант **[none]**, то эта функция недоступна. **[Saver window no.]** Назначает окно используемое для заставки **[Screen saver]**
Option	**[Startup window no.]** Назначает окно, появляющееся сразу после запуска панели оператора. **[Common window]**  Объекты общего окна (окна типа 4) будут отображаться в каждом основном окне. Этот выбор позволяет задать слой, на котором будут находиться эти объекты: поверх (above) или под (below) объектами основного окна. **[Keyboard caret color]** Задает цвет курсора ввода.

	[Object layout] – Размещение объектов  Если выбран режим **[Control]**, то во время работы панели, объекты типа анимации **[Animation]** и «подвижная фигура» **[Moving Shape]**будут отображаться поверх объектов других типов независимо от порядка их создания. Если выбран режим “**[Nature]** , то порядок отображения объектов будет соответствовать порядку их создания. **[RW_A enabled]** Активируйте или дезактивируйте использование данных рецептов **RW_A**. Если отметить флаг RW_A, то объект сможет контролировать содержимое RW_A. Размер области RW_A составляет 64 Кбайт.
Event	**[Extra no. of events]** Наибольшее число запоминаемых событий – значение по умолчанию равно 1000. Если пользователю нужно добавить больше записей, то значение параметра может быть увеличено до 10000.
Keyboard	В проекте можно использовать до 32-х различных типов клавиатур для ввода цифр и символов **[Numeric Input]** и **[Word Input]** Если пользователь желает создать новую клавиатуру, ее необходимо сконфигурировать в существующем окне. Нажмите **[Add…]** после создания для добавления окна в список. За деталями обращайтесь к главе 12 (Проектирование и использование клавиатуры)
Project protection **(i series only)**	Проект может быть закреплен и выполняться только на указанной панели. Подробности в главе 30 (Защита проекта).

5.4 System Setting (Системные настройки)

Параметры в закладке **[System Setting]** предназначены для настройки различных вспомогательных (miscellaneous) функций в EasyBuilder Pro.



Некоторые функции дублируются с системными тагами, например: **[Hide system setting bar]** и тэг **(LB-9020), [Hide mouse cursor]** и тэг **(LB-9018), [Disable buzzer]** и тэг **(LB-9019), [Prohibit remote HMI connecting this machine]** и тэг **(LB-9044).**

Это означает, что пользователь может использовать эти функции, применяя системные тэги. Чтобы использовать системный тэг, отметьте **[system tag]** в **[address]** при создании нового объекта. Просмотреть все системные тэги можно в **[Library]** EasyBuilder Pro, выбрав **[Tag]** и затем **[System].**

.

[Startup language after redownloading the project]
Назначить язык, используемый при пуске панели после перезагрузки проекта.

[Execute init. Macro when power on]
Назначить «macro» выполняемым при включении питания панели.

[Auto logout]
Если панель не использовалась дольше указанного здесь времени, то панель отключится автоматически.

.

[Hide System Setting Bar]
Спрятать панель системных настроек (system setting bar) в нижнем правом углу экрана панели..

[Hide Mouse Cursor]
Спрятать курсор па экране панели.

[Disable Buzzer]
Немая панель оператора (Mute HMI).

[Prohibit remote HMI connecting to this machine]
Запрещение связи удаленной панели с этой машиной.

[Disable upload function (effective after rebooting HMI) (или LB9033=ON)]
Отключение функции выгрузки проекта из панели после загрузки. Панель должна быть перезапущена для активации функции «отключение возможности выгрузки проекта»

[Prohibit password remote-read operation (или LB9053=ON)]
Запрещение удаленной панели читать пароль в локальной панели оператора.

[Prohibit password remote-write operation (или LB9054=ON)]
Запрещение удаленной панели писать пароль в локальную панель оператора.

[Use a disconnection icon on relative objects when PLC communication fails]
Решить показывать или нет иконку разъединения (disconnection icon) у соответствующего объекта, при пропадании связи с ПЛК.
При использовании этой функции и пропадании связи с ПЛК, эта иконка будет показана в нижнм правом углу объекта, как показано ниже:

[VNC Server]

Назначение пароля доступа к серверу VNC.

[LW protection], [RW protection]

Если пользователь отметил **[Disable LW/RW remote-write] – (**Отключить удаленное чтение/запись) и установил диапазон защищаемых данных **[LW/RW range]**, значения в защищенном диапазоне не могут быть изменены через удаленную панель (remote HMI).

[Easy Access server]

Используя эту технологию, пользователи могут иметь доступ к любой панели оператора подсоединной к **internet** и взаимодействовать с ней при помощи компьютера, так же как держа сенсорный экран в руках.

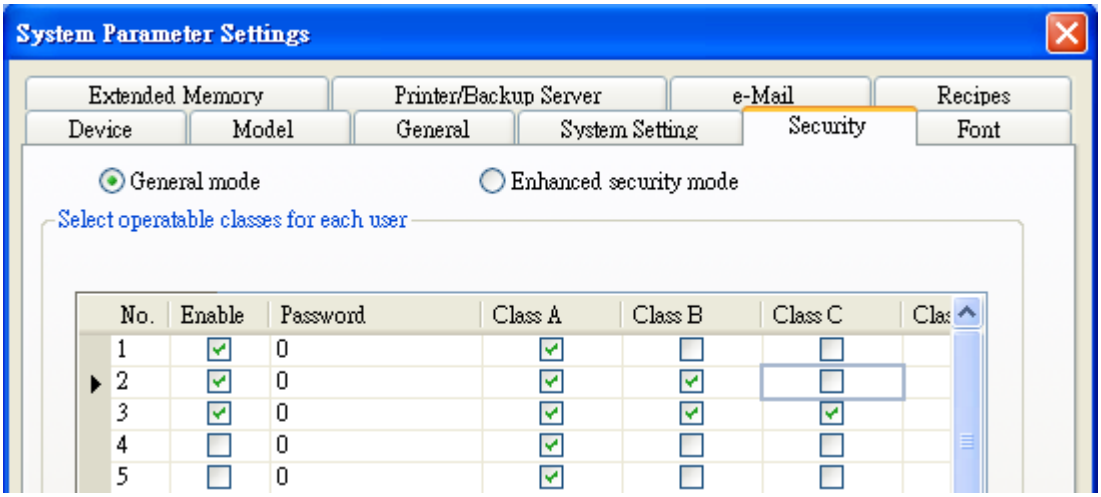
В отличие от большинства серверов, сервер **Easy Access** не нуждается в обновлении графических изображений и передает только данные изменяемые в реальном времени. Это делает обмен быстрым и эффективным.

Дополнительная информация содержится в описании “*EasyAccess*”.

5.5 Security (Защита)

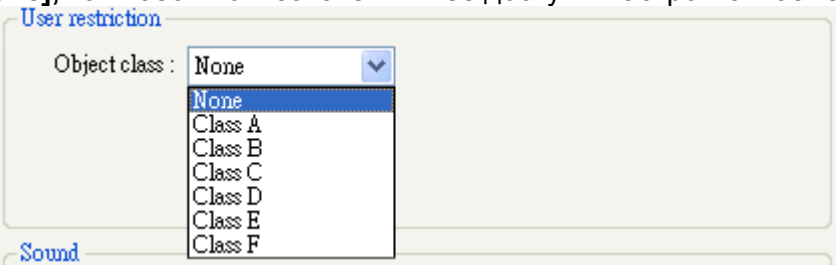
Параметры на закладке [Security] определяют уровни - классы доступа для каждого пользователя и соответствующие пароли. По степени защиты объекты могут иметь классы **[A~F]** и **[None].** До 12 паролей может быть назначено**.**. Допускается использовать только цифровые пароли в диапазоне от 0 до 999999999.

5.5.1 General mode (Основной режим)

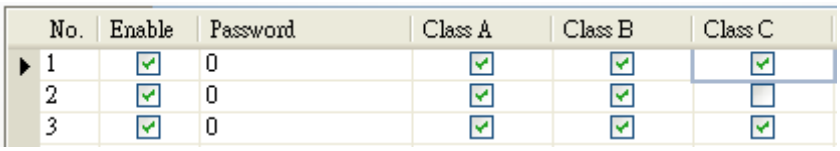


В соответствии с настройками безопасности EasyBuilder Pro управляет классами (уровнями) доступа каждого пользователя для настройки объектов после ввода ими паролей.

В EasyBuilder Pro при создании проекта можно установить классы объектов **[A~F]** и **[None],** как показано ниже**.**
Если выбрано **[None],** то любой пользователь имее доступ к настройкам объекта.



Например, когда уровни доступа пользователя User 1 заданы как показано ниже, только этот пользователь может иметь доступ к объектам классов A, B и C.
Подробнее: см. главу 10 (Защита).



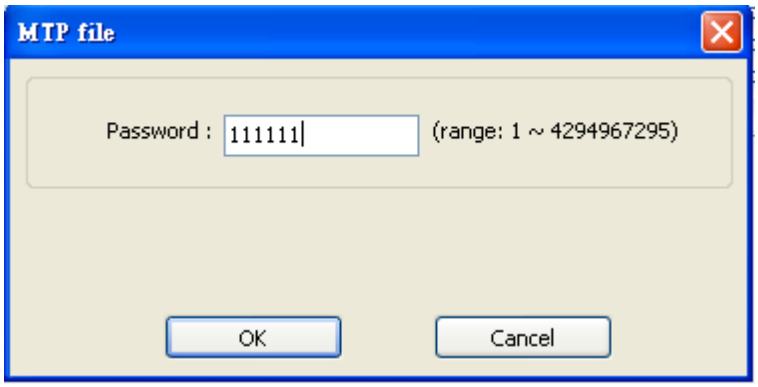
[Project password (MTP file)]

Пользователь может установить пароль, чтобы защитить файл «*.MTP» в меню **[System parameter]** / **[Security tab].**

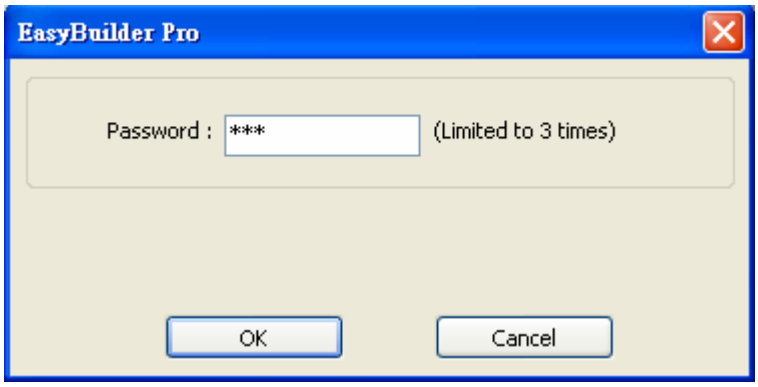
Пользователь должен ввести пароль для защиты редактирования файла «*.MTP».

(Диапазон пароля MTP: 1~4294967295)

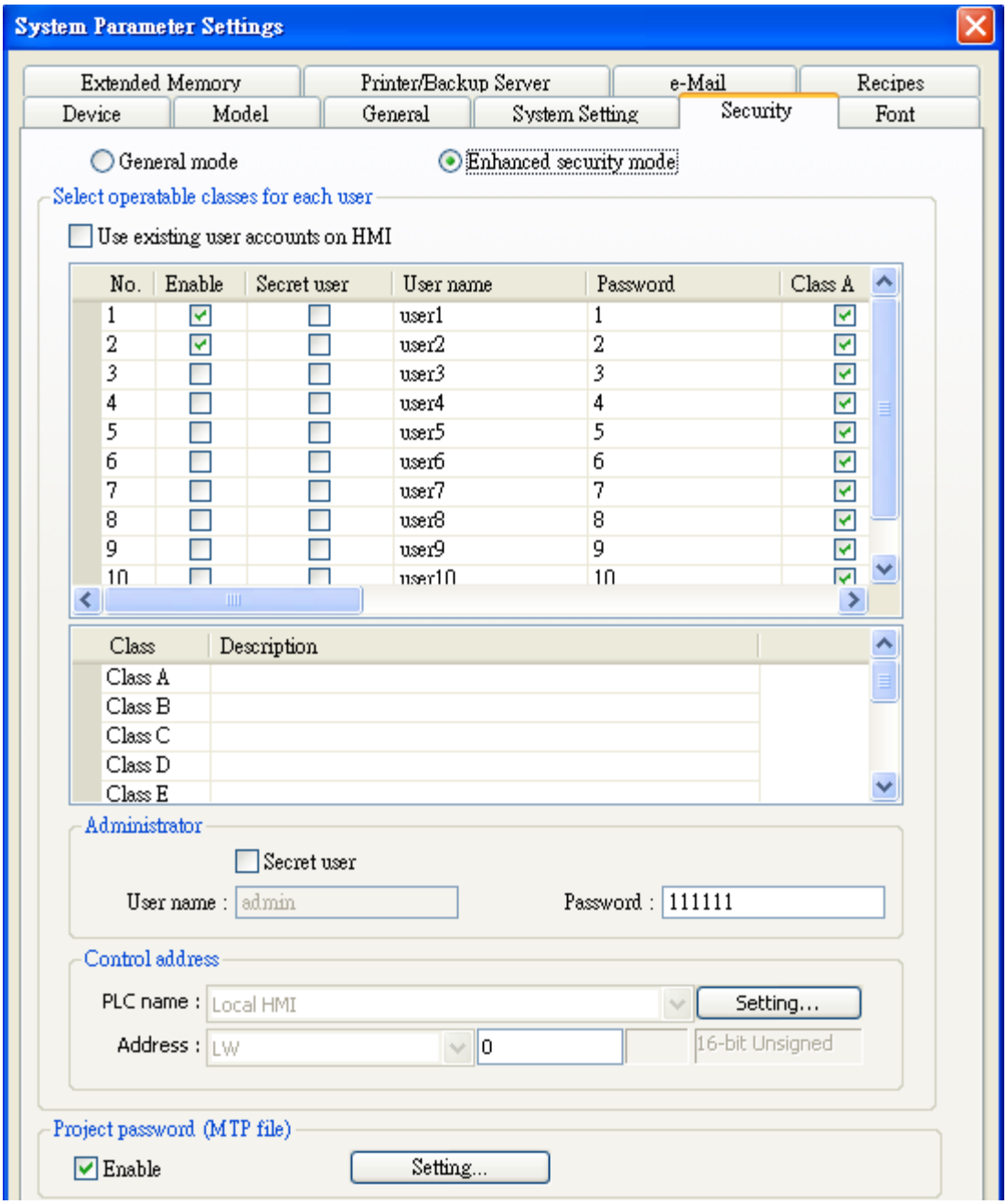
Отметьте **[Enable],** затем щелкните по **[Setting]**, появится окно, показанное ниже.



Перед редактированием проекта всплывет окно с запросом ввода пароля для доступа к проекту.

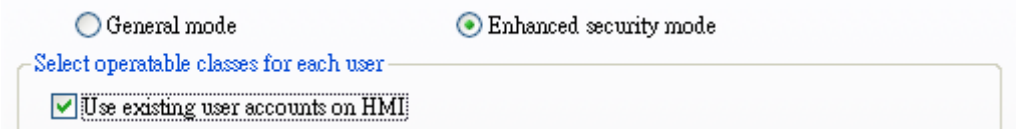


5.5.1 Enhanced security mode (Улучшенный режим защиты)

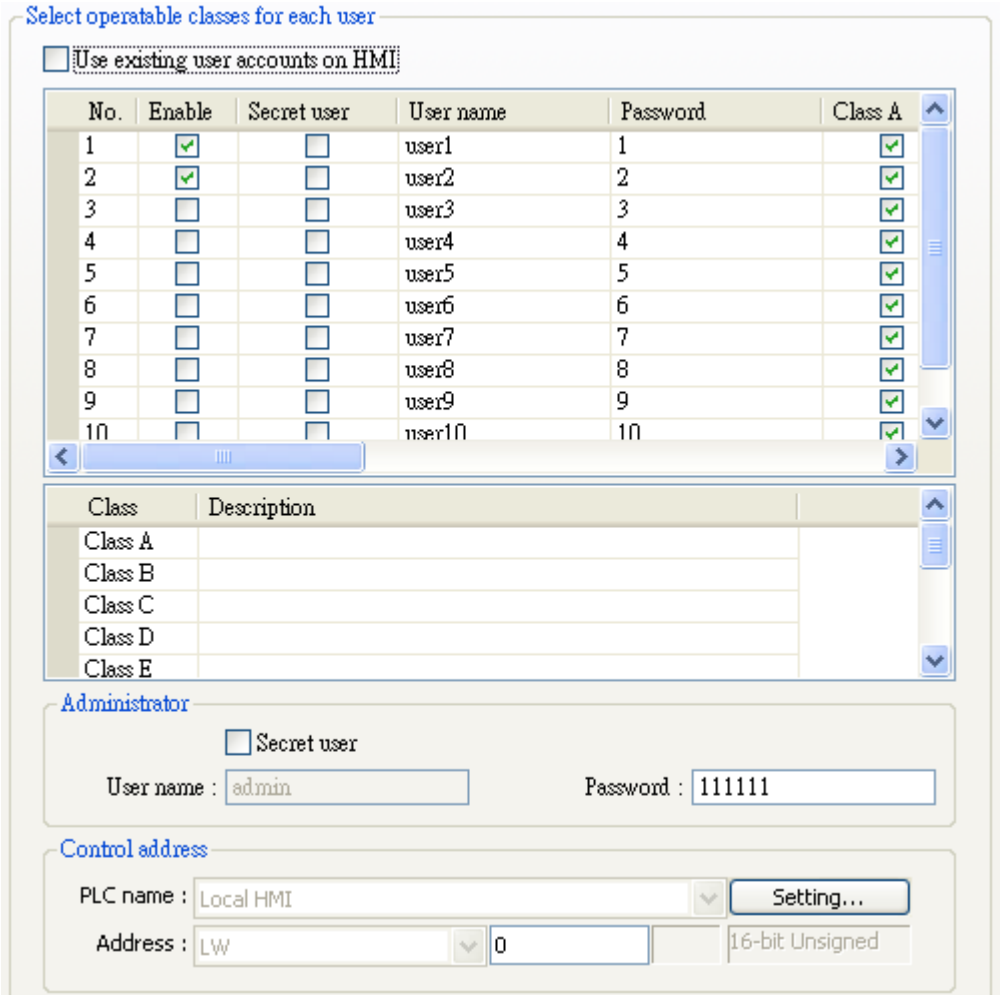


[Select operable classes for each user] – Выбрать действующие классы для каждого пользователя.

Если отметить **[Use existing user accounts on HMI],** действующие объекты для каждого пользователя будут определены в соответствии с настройками панели.



Если не делать эту отметку, настройте учетную запись и пароль, как показано на следующей странице.



[Administrator]

Выберите привилегированных пользователей (secret users).

[Control address]

Назначьте управляющие адреса настройки пароля пользователя.

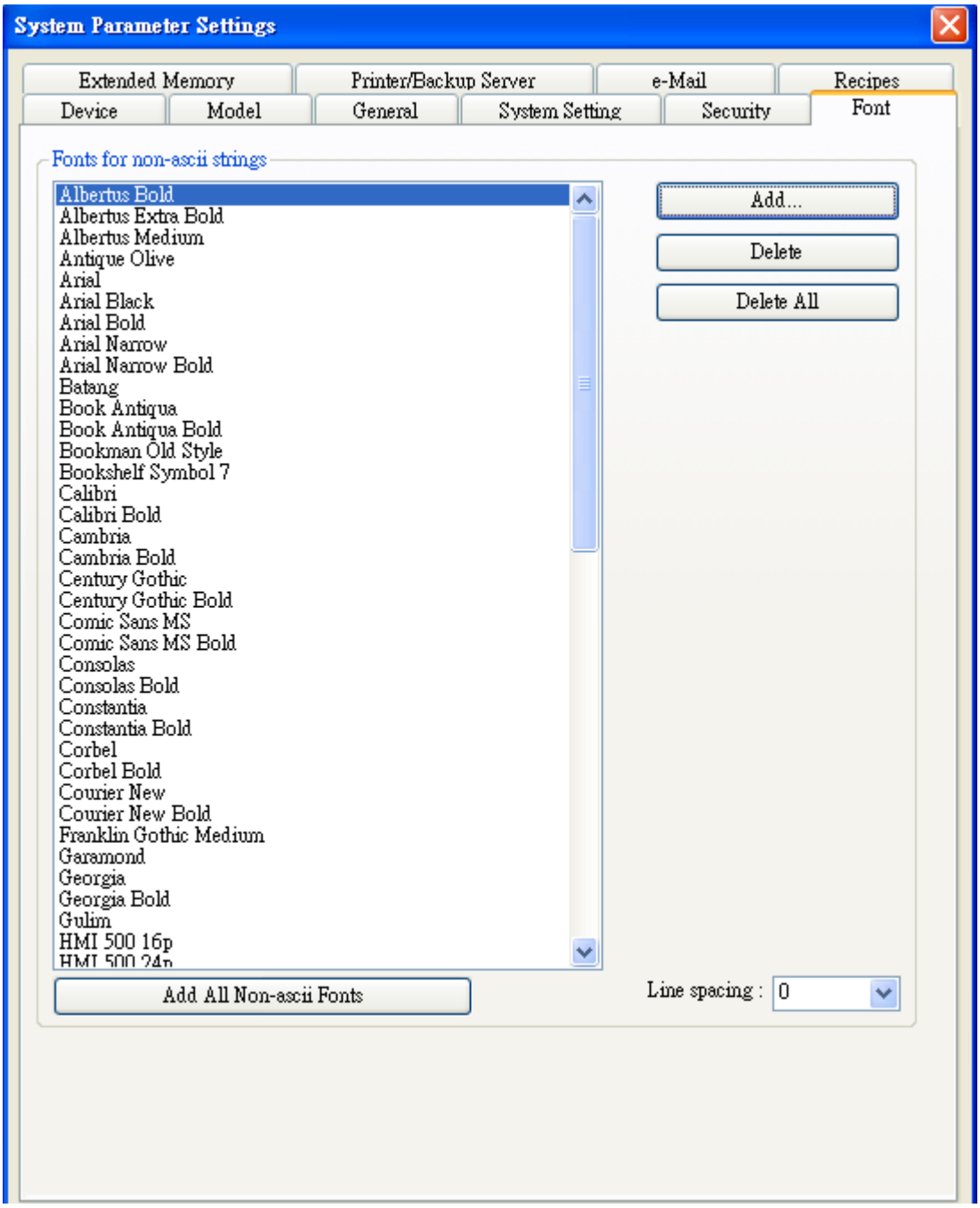
.

[Project password (MTP file)]

Прочтите раздел 5.5.1

5.6 Font (Шрифт)

Параметры на закладке **[Font]** определяют используемые в EasyBuilder Pro шрифты с отличной от ASCII кодировкой.



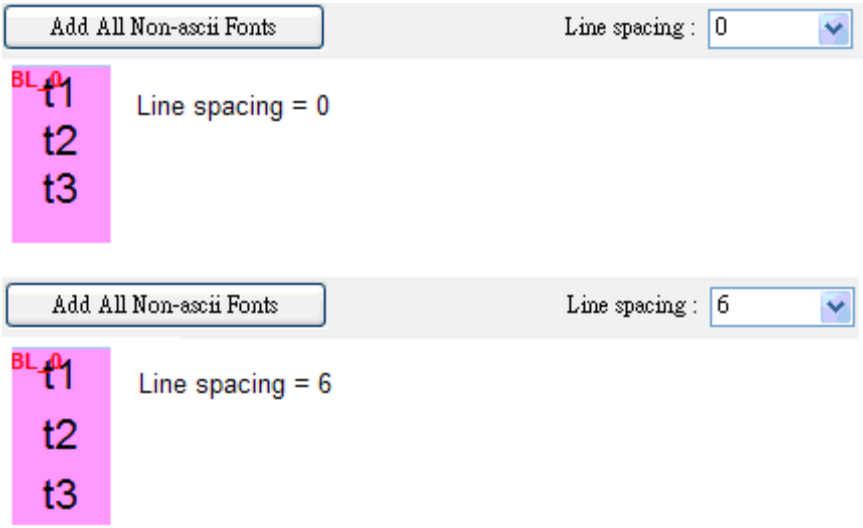
[Fonts for no-ascii strings]

Список шрифтов с кодировкой, отличной от ASCII приведен выше. В случае, если пользователь применяет шрифт с кодировкой, отличной от ASCII и отсутствующий в этом таблице **[Fonts for no-ascii strings]**, EasyBuilder Pro выберет шрифт из списка для автоматической замены.

Пользователь может также проверить, какие шрифты с отличными от ASCII кодировками для Windows могут применяться в EasyBuilder Pro и затем добавить их в таблицу **[Fonts for no-ascii strings].**

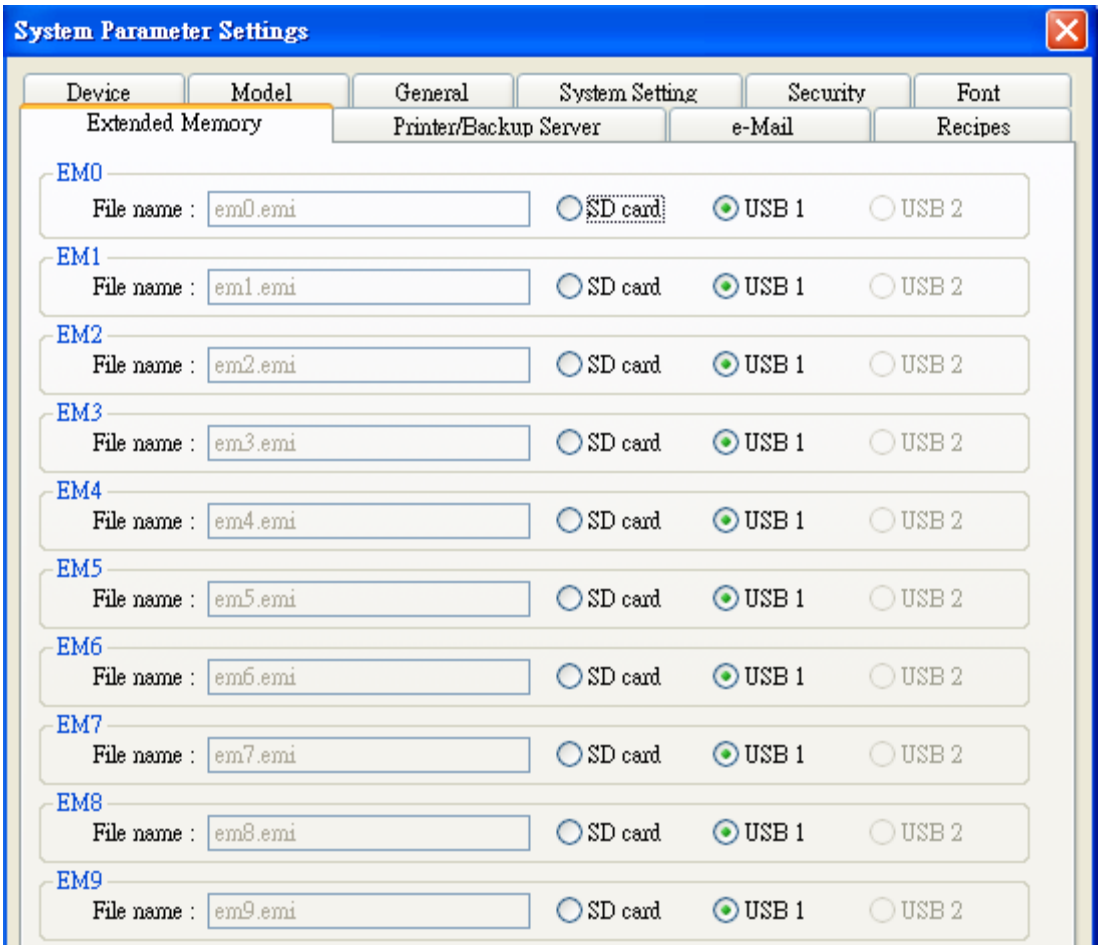
[Line spacing]

Определяет интервал между строками в тексте.

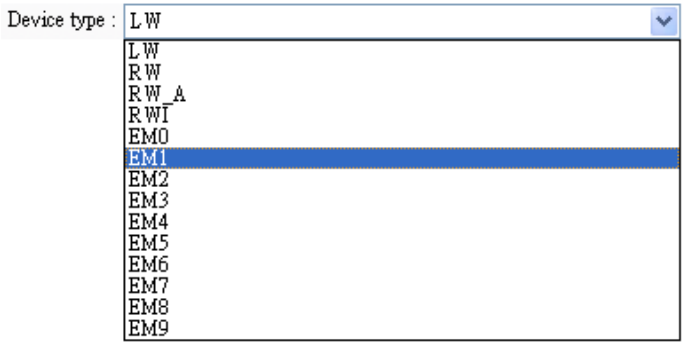


5.6 Extended memory (Внешняя память)

Параметры на закладке **[Extended Memory]** определяют пути хранения файлов на внешних носителях.



Внешняя память нумеруется с EM1 по EM9. Способ использования внешней памяти аналогичен работе с другими устройствами (с адресацией типа LW или RW). Пользователь может просто выбрать из списка **[Device type]** при создании нового объекта. Предельный размер каждой внешней памяти – $2*10^9$ слов.



Данные во внешней памяти хранятся в виде файлов на носителях типа **[SD card], [USB1],** или **[USB2]**. Имя файла выбирается из диапазона **EM0~EM9**, полное имя: **em0.emi~em9.emi**.
Для открытия и редактирования файлов, хранящихся во внешней памяти, можно использовать приложение **RecipeEditor.exe.**

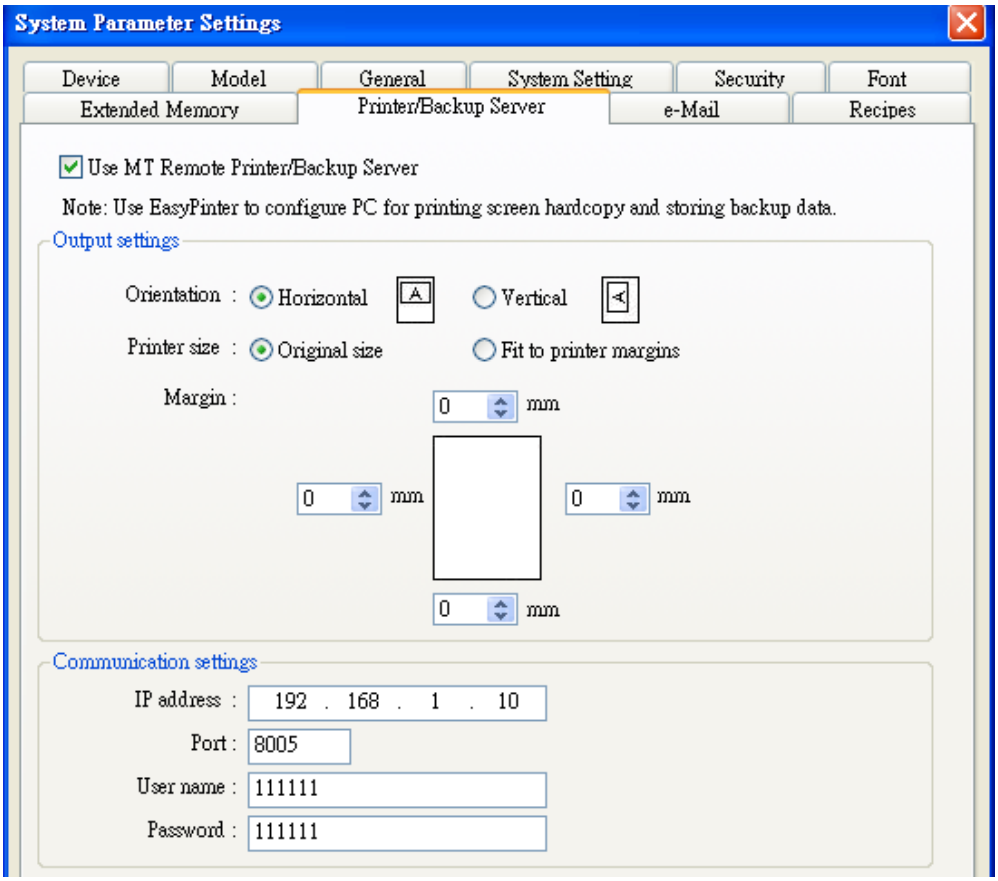
Данные на внешних устройствах памяти не стираются при отключении питания, потому при последующем включении панели данные будут теми же самыми. Это происходит аналогично данным Recipe data (EW, RW_A), отличие лишь в том , что пользователь может выбрать где надо сохранять данные **([SD card], [USB1],** или **[USB2]**).

При попытке чтения данных пользователем с внешнего устройства, которое не подключено, будет показан нулевой объем содержащейся информации. При попытке записи информации на отсутствующий носитель внешней памяти появится сообщение "PLC no response" (“ПЛК не отвечает”) на экране панели.

EasyBuilder Pro поддерживает функцию «горячей» замены карты памяти SD или USB-накопителя. Пользователь может вставлять и вынимать устройства внешней памяти без отключения питания.

5.7 Printer/Backup server (Сервер печати)

Параметры на закладке **[Printer/Backup Server]** нужны для настройки удаленного принтера MT.



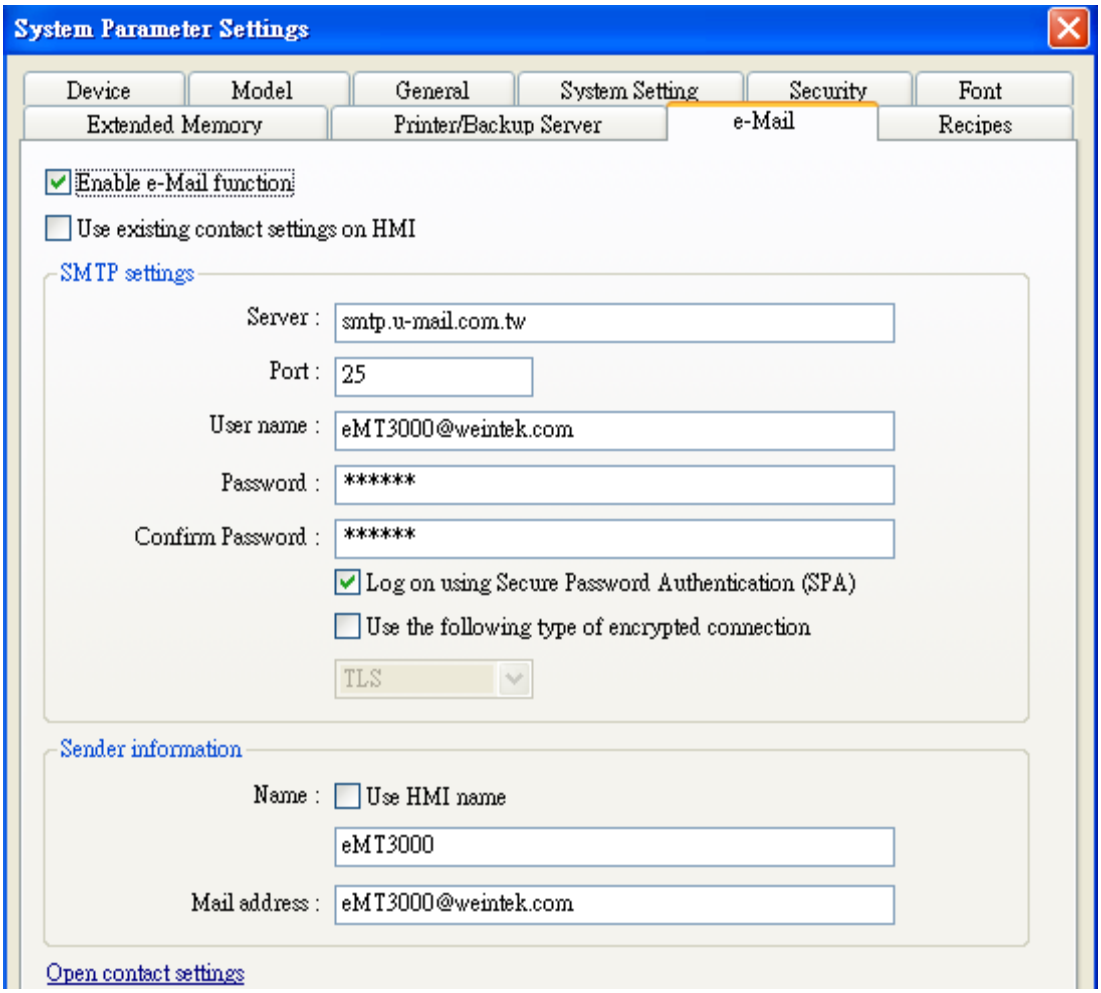
Настройка	Описание
Output settings	**[Orientation] (Ориентация страницы)** Установите ориентацию текста и изображений, выводимых на печать: **Horizontal**: портретная; **Vertical**: Альбомная **[Printer size] (Область печати)** Выберите вариант: печать по действительному формату (**Origainal size**) или масштаб по формату принтера (**Fit to printer margins**). **[Margin] (Отступы от края листа)** Установите границы листа, включая верхнюю, нижнюю, правую и левую
Communication Settings	**[IP address]** Назначить IP-адрес удаленного принтера для работы по сети. **[Port], [User name], [Password] (номер порта, имя пользователя, пароль)** Назначить информацию для доступа. Значение параметра [Port] должно быть в диапазоне от 1 до 65535. Предельная длина имени пользователя: 12 символов.

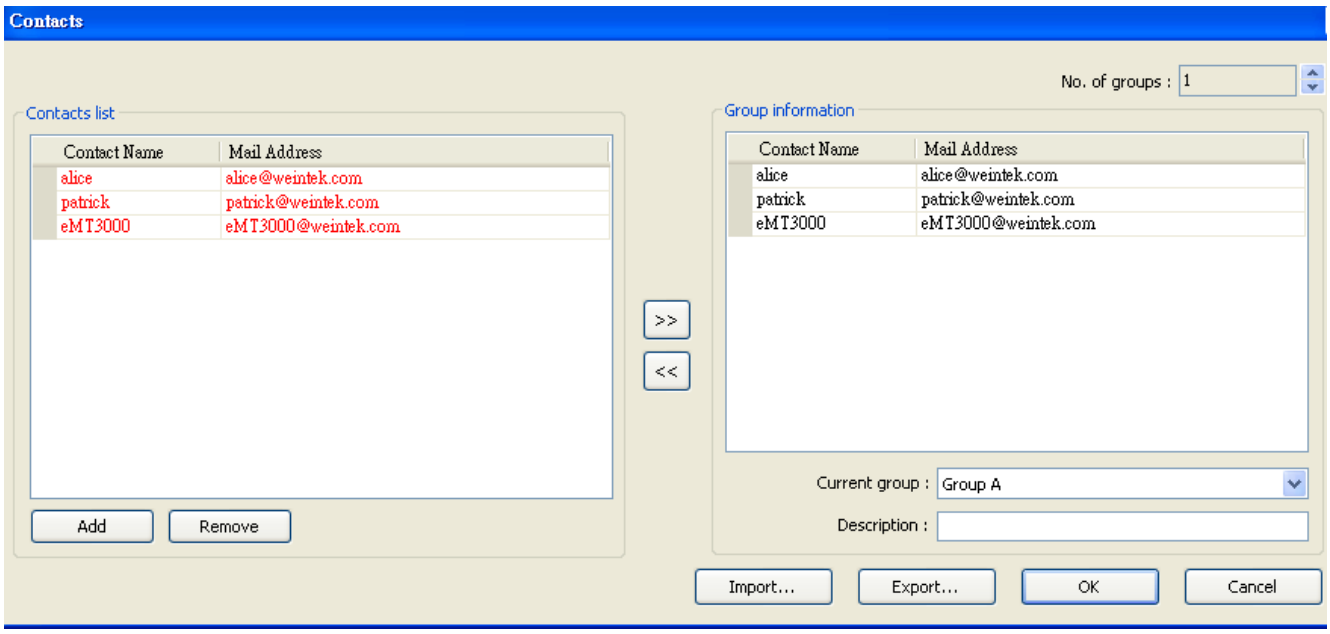
За детальной информацией обращайтесь к главе 26 (EasyPrinter).

5.9 E- Mail (Почта)

Параметры на закладке **[e- Mail]** нужны для настройки почты.

[Enable e- Mail function]

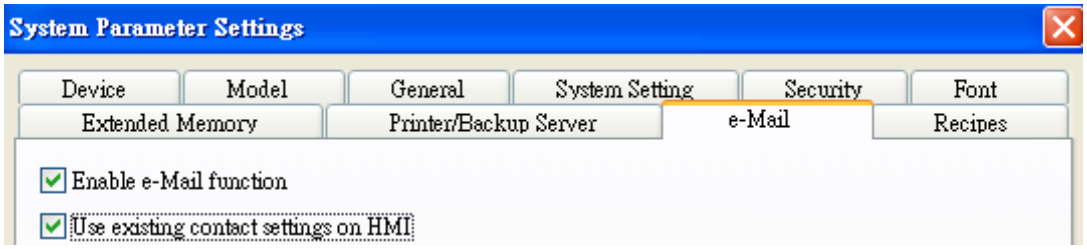




Настройка	Описание
SMTP settings	**[Server]** Настроить сервер SMTP. **[Port]** Настроить communication port. **[User name]** Настроить адрес e-mail. **[Password]** Настроить пароль e-mail. **[Confirm Password]** Подтвердить пароль e-mail. **[Log on using Secure Password Authentication (SPA)]** Необходимо ли использовать SPA при регистрации (login) на e-mail. **[Use the following type of encrypted connection]** Необходимо ли использовать кодированное соединение(TLS, SSL) при посылке e-mail
Sender Information	**[Name]** Указать имя отправителя или использовать имя панели (HMI name). **[Mail Address]** Настроить адрес e-mail
Open Contact settings	**[Contact List]** Добавить или удалить контакт из списка. **[Group Information]** Группа контактов. **[No. of groups]** Настроить номер группы контактов. В соответствии с номером группы будут названы A~P (всего до 16-ти групп может быть назначено). **[Current group]** Текущая группа. **[Description]** Описание группы,

За детальной информацией обращайтесь к главе 24 (Редактор базы данных рецептов).

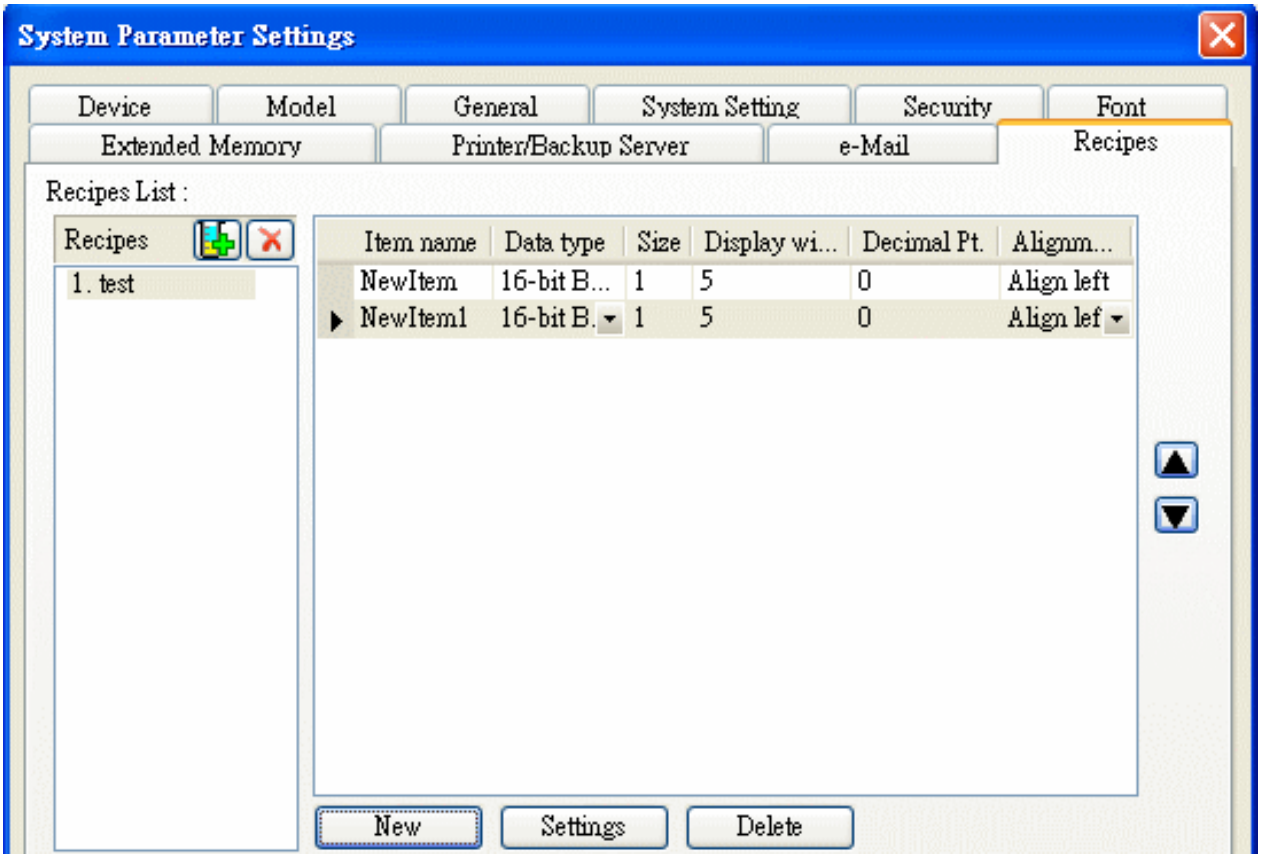
[Use existing contact settings on HMI] – Использовать существующие в панели настройки контактов.



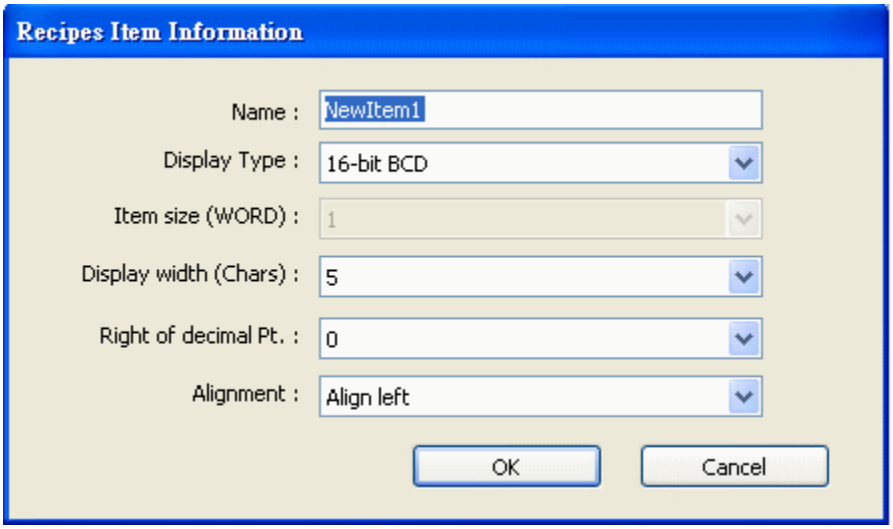
После таких пометок, система будет использовать настройки контактов в панели.

5.10 Recipes (Рецепты)

Параметры на закладке [**Recipes**] нужны для настройки соответствующих данных рецептов.



Настройка	Описание
Recipes List	**[New(Insert)]** Добавить новый рецепт. **[Delete]** Удалить существующий рецепт
New	**[Item name]** Ввести имя элемента рецепта. **[Data type]** Настроить базу данных рецептов. **[Size]** Настроить размер элемента рецепта **[Display width]** Настроить ширину отображения элемента рецепта. **[Decimal Pt.]** Настроить место десятичной точки. **[Alignment]** Настроить выравнивание

Настройка	Описание
Settings	Пользователь может определять и редактировать все пункты настроек. 
Delete	Удаление выделенных настроек

За детальной информацией обращайтесь к главе 24 (Редактор базы данных рецептов).

Глава 6. Работа с окнами

Окно — базовый компонент проекта. С помощью окон пользователь может увидеть на экране панели все виды информации: в виде обЪектов, изображений или текста. До 1997 окон пронумерованных от 3 до 1999 могут быть использованы в EasyBuilder Pro.

6.1 Типы окон

Существуют 4 типа окон, отличающихся по функциональности и способу использования:.
(1) Base Window (Основное окно), (2) Common Window (Общее окно), (3) Fast Selection Window (Окно быстрого выбора), (4) System Message Window (Окно системных сообщений).

6.1.1 Base Window - Основное окно

Этот тип окон используется часто.
Оно применяется как:

a. В качестве фона (background) для других окон
b. Окно с клавиатурой
c. Всплывающее окно для объекта типа «функциональная кнопка» (function key)
d. Всплывающее окно для объектов [Direct window] и [Indirect window].
e. Заставка (screen saver).

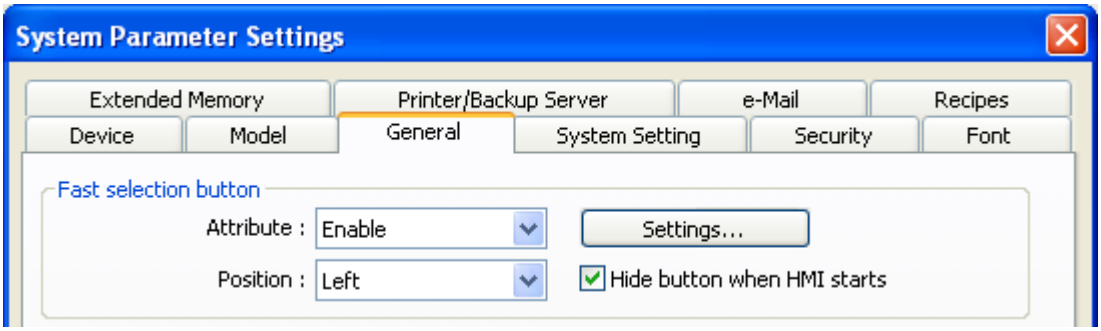
На рисунке показано стартовое окно, в качестве которого использовано окно типа «Base Window».

✹ Окно Base Window должно быть тогоже размера, что и экран панели. Следовательно. разрешение этого оена и панели должны быть одинаковыми.

6.1.2 Fast Selection Window - Окно быстрого вызова

Окно 3 (Window no. 3) определено, как окно быстрого выбора. Это окно может использоваться совместно с основным окном. Как правило, оно содержит часто вызываемые команды, как на рис. ниже:

Кнопка быстрого вызова (Fast Selection Button), используется для активации/дезактивации функции быстрого вызова. Все настройки кнопки быстрого вызова прописаны в Системных настройках: **[System Parameter Settings / General]**



Кроме настройки атрибутов быстрого вызова с помощью кнопки быстрого вызова, в системных регистрах выделено несколько адресов:

[LB9013] Управление окном быстрого вызова (FS) - [Показать (ON) / Спрятать (OFF)]

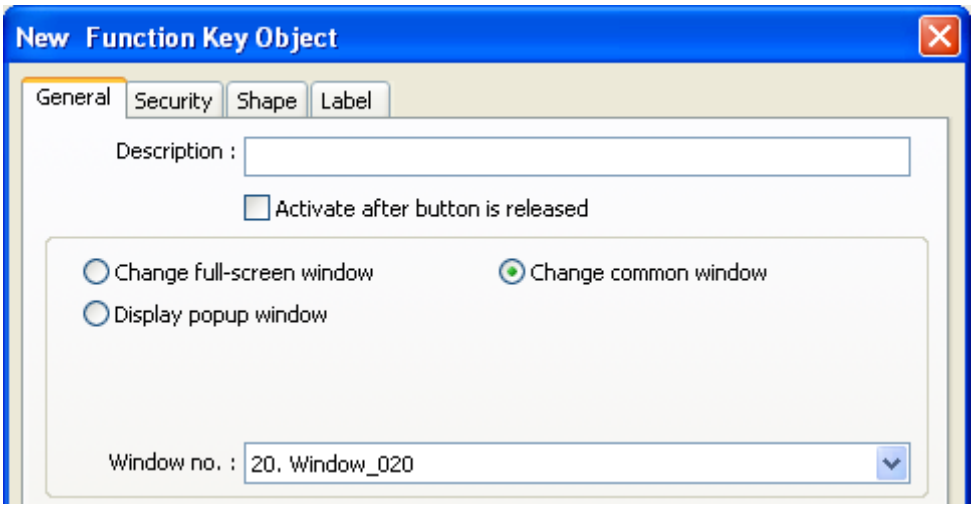
[LB9014] Управление кнопкой быстрого вызова [Спрятать (ON) / Показать (OFF)]

[LB9015] Управление окном / кнопкой быстрого вызова [Спрятать (ON) / Показать (OFF)]

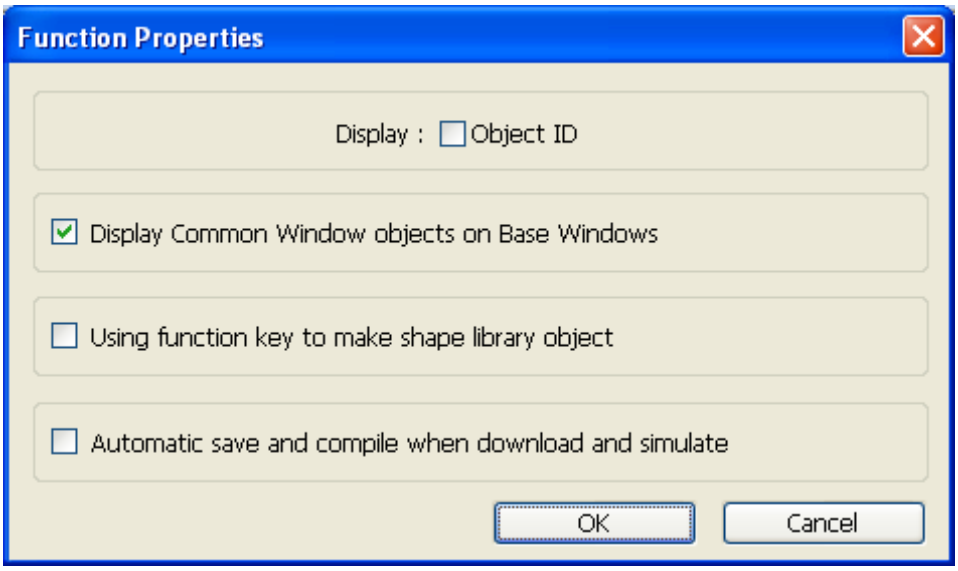
6.1.3 Common Window - Общее окно

Окно 4 (Window no. 4) по умолчанию является общим окном. Объекты этого окна отображаются во всех других окнах, кроме всплывающих окон (popup windows). Поэтому, объекты совместного использования всегда размещают в общем окне. Например: логотип изделия или общие кнопки.

Во время работы системы, выбрав режим **[function key] / [Change common window],** можно изменить источник общего окна. Например, можно сделать общим окно 20 вместо окна 4.



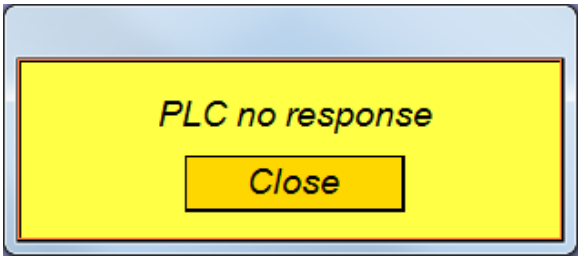
В меню **[Option]/[Function Properties]** выберите **[Display Common Window objects on Base Windows]** при редактировании объекта.
Это поможет избежать накладывания объектов в основном окне на объекты общего окна.



6.1.4 System Message Window - Окно системных сообщений

По умолчанию окнами системных сообщений назначены окна 5, 6, 7 и 8.

[Window No. 5: PLC Response] – “Ответ контроллера”
При прерывании соединения между ПЛК и панелью оператора это окно сообщений автоматически появляется справа на ранее открытом основном окне.



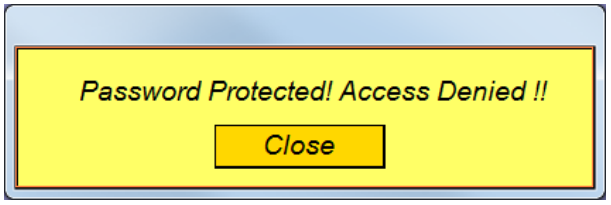
✸ Окно “PLC no response” можно настроить, чтобы оно не появлялось, используя системные регистры. Прочитайте “Глава 22 Системные слова и биты”.

[Window No. 6: HMI Connection]

Окно 6 — окно сообщений соединения с панелью (“**HMI connection**”)
Появляется автоматически при разрыве соединения с удаленной панелью оператора.

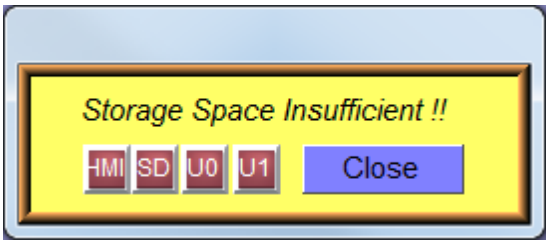
[Window No. 7: Password Restriction]

Окно 7 предназначено для сообщения о необходимости ввода пароля (“**Password Restriction**”) В случае, если у пользователя недостаточно прав доступа к объекту, окно 7 появится автоматически в соответствии с заданными параметрами безопасности.



[Window No.8: Storage Space Insufficient]

Окно 8 предназначено для сообщения о малом объеме оставшейся памяти **(“Free Space Insufficient”**).
Окно появляется автоматически, когда заканчивается свободное пространство внутренней памяти панели, USB-накопителя или флэш-карты.



Пользователь может провести просмотр оставшегося свободного пространства памяти панели, USB-накопителя или SD-карты, используя адреса системных тэгов, которые указаны ниже:

[LW 9072] текущий объем свободной памяти панели (Кбайт)

[LW 9074] текущий объем свободной памяти карты (Кбайт)

[LW 9076] свободное пространство USB-накопителя №1 (Кбайт)

[LW 9078] свободное пространство USB-накопителя №2 (Кбайт)

EASYBUILDER PRO позволяет также устанавливать автоматически оповещения о недостаточном объеме оставшейся памяти:

[LB 9035] оповещение о малом объеме памяти панели (когда состояние ON)

[LB 9036] оповещение о малом объеме памяти карты SD (когда состояние ON)

[LB 9037] оповещение о малом объеме памяти USB-накопителя №1 (когда значение ON)

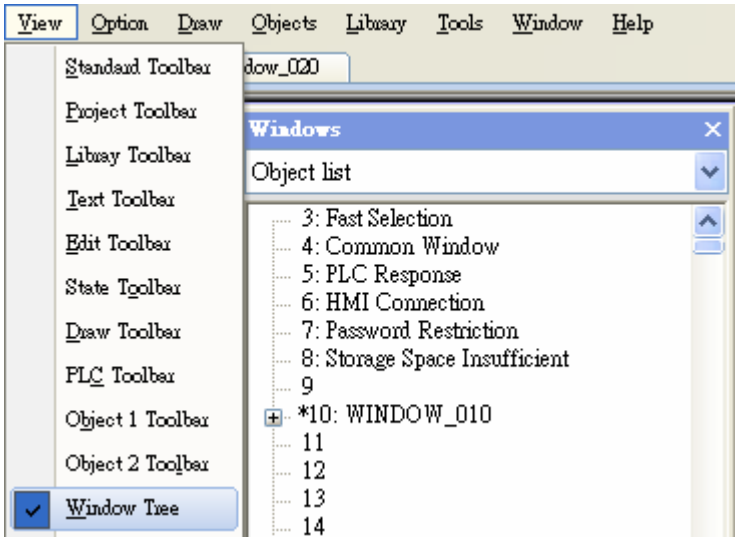
[LB 9038] оповещение о малом объеме памяти USB-накопителя №2 (когда значение ON)

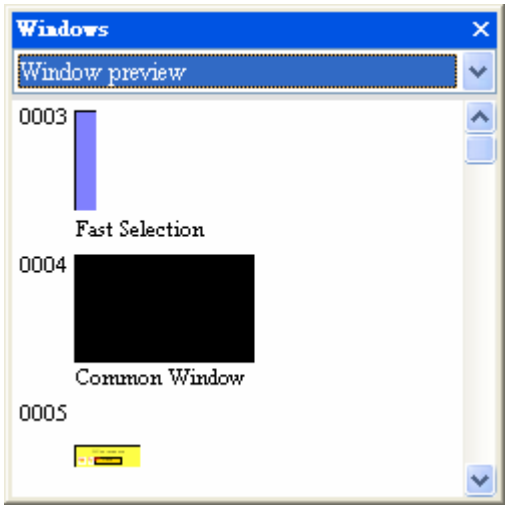
Пользователь может изменять текст показываемый в окнах 5 – 8, для лучшего понимания оператором.

Note

- Одновременно на экране могут отображаться максимум 16 всплывающих окон, включая Окно системных сообщений, объекты типа [Direct window] и [Indirect window].
- Окно может отображаться только однажды в каждый момент времени. Так, пользователь не может открыть два объекта типа [Direct/Indirect window] для отображения одного и того же окна в базовом одновременно.
- Окна с 3 по 9 могут использоваться только системой; окна с 10 по 1999 предназначены для целей пользователя

6.2 Создание, удаление и настройка окна





В меню **EasyBuilder Pro / [View] / [Window Tree]** можно проверить все созданные окна. Список объектов **[Object List]** отображает номера и имена окон. Открытое и редактируемое окно будет помечено The (*) mark, нажмите (+) перед номером окна, чтобы увидеть объекты, ID объекта, адреса и описания объектов, содержащихся в данном окне.

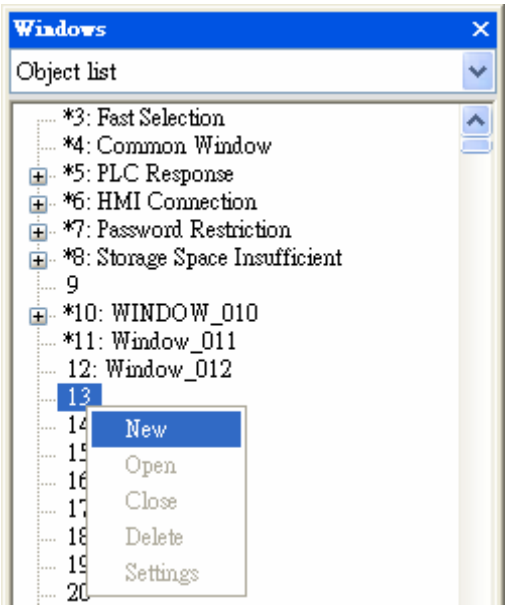
Предварительный просмотр окон **- [Window Preview]** отображает окна в уменьшенном размере**.**.

6.2.1 Создание и настройка окна

Существует два способа создания окна:

В открывшемся контекстном меню нужно выбрать пункт [New] и щелкнуть ОК после завершения всех настроек. Пример ниже иллюстрирует данные действия:

✹ Способ 1
Выберите номер окна в дереве окон, сделайте щелчок правой кнопкой мыши и выберите **[New]**.



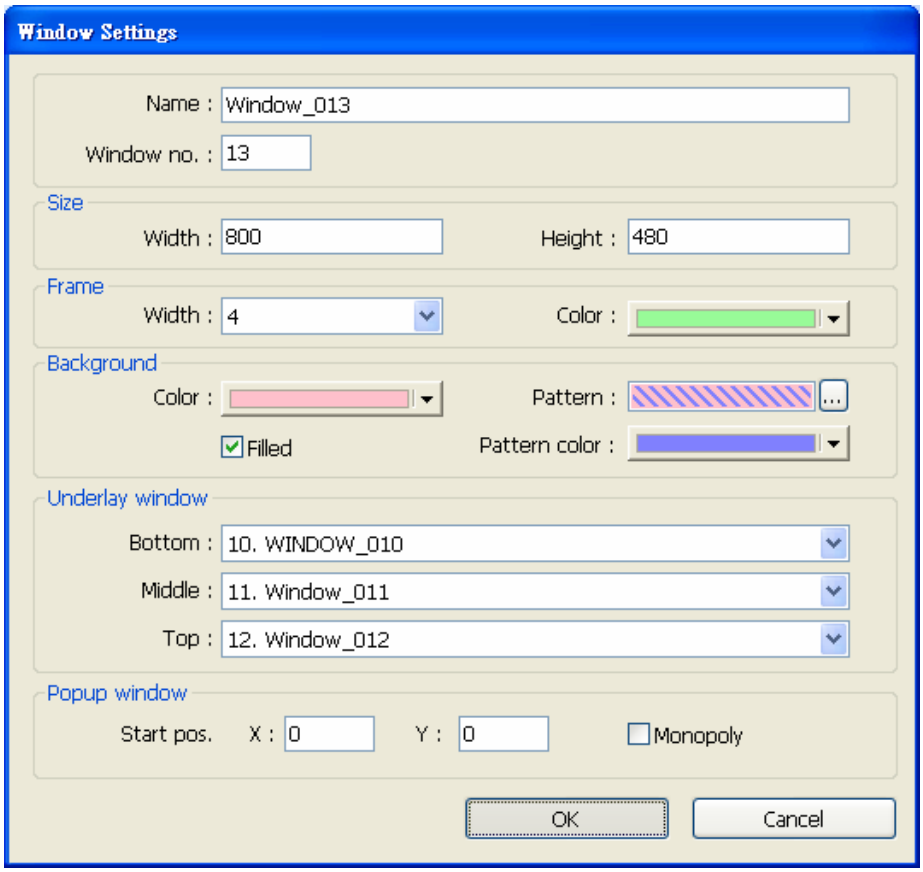
[Name] Название окна, появляющееся в дереве окон и зазоловке окна.

[Window no.]
Номер окна, от 3 до 1999.

[Width]/ [Height] Ширина и высота окна. Настройте основное окно в соответствии с разрешением панели

[Underlay window]
См. описание внизу.

[Popup window]
Основное окно может быть использовано как всплывающее. Задайте координаты **[X]** и **[Y]** для определения местоположения всплывающего окна. Изначально: это координаты верхнего левого угла.



[Monopoly]
Если отмечен данный флаг, то, когда основное окно используется, как всплывающее, пользователю недоступна работа с другими окнами. Если базовое окно используется как окно с клавиатурой, то данный флаг автоматически активируется

[Underlay window] – Подслойное окно**.**

Часто используемые объекты могут быть размещены на разных окнах (но не на всех). *Underlay window* можно рассматривать как дополнительное общее окно. До трех *Underlay window* может быть создано пользователями.

Underlay window это основное окно, которое может отображаться с основным окном, с которого оно было вызвано. До трех основных окон может быть назначено в качестве *Underlay window* для каждого основного окна от **[Bottom]** до **[Top]**. Объекты (но не фоновые) в *Underlay window* отображаются в этом порядке в основном окне.

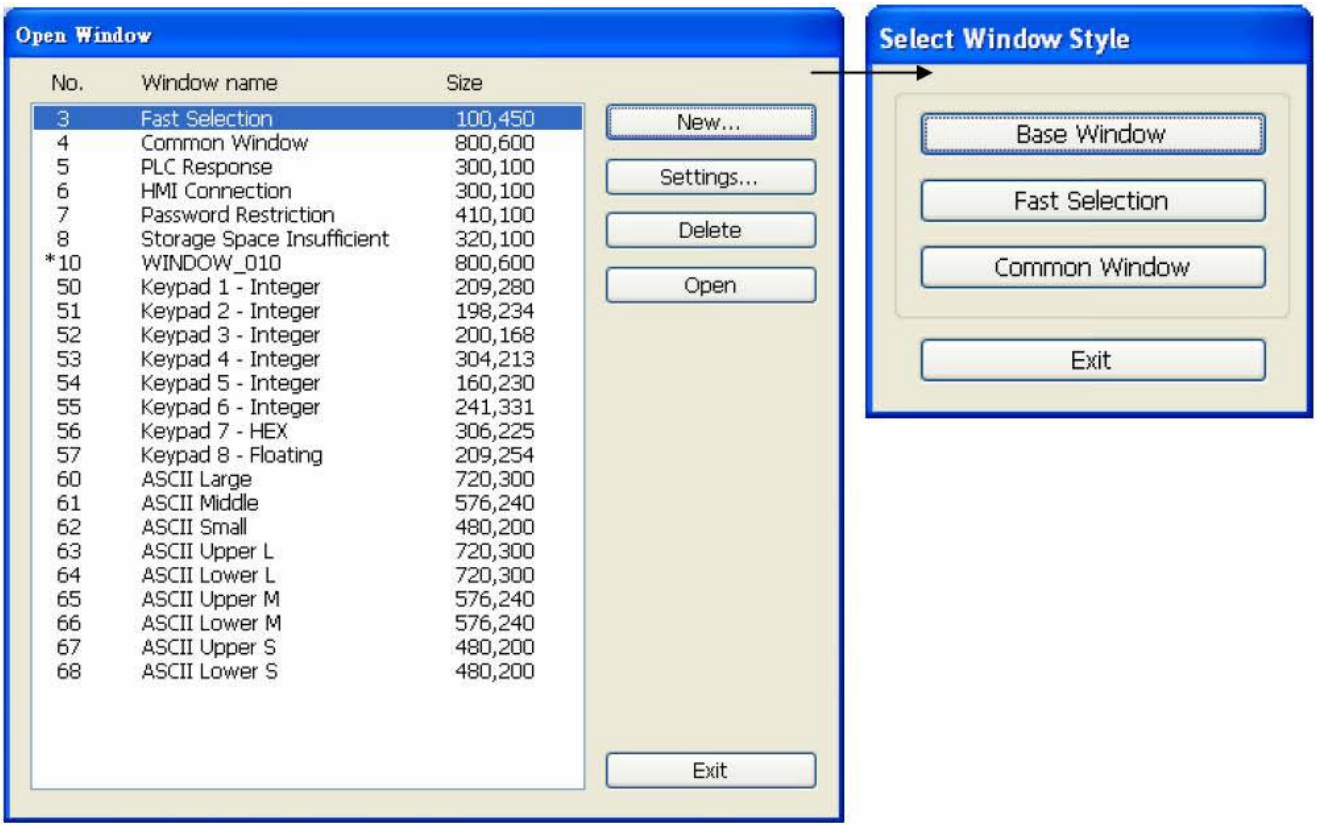
Note

✹ Фоновые (background) объекты нельзя редактировать в основном окне, в котором они отображены. Чтобы редактировать объекты в *Underlay window,* откройте окно, в котором они были созданы, используя EasyBuilder Pro.

✹ Способ 2

EasyBuilder Pro / [Window] / [Open Window]

Нажмите **[New]**, чтобы выбрать стиль окна в появившемся диалоговом окне **[Select Window Style]** и нажмите **[OK]**.



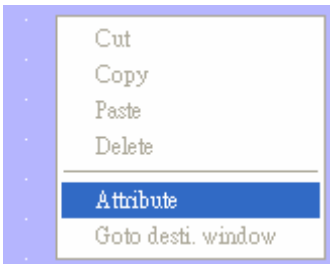
.

В EasyBuilder Pro есть 3 способа вызова окна редактирования свойств окна **[Window Settings]**:

✹ Способ 1
Щелкните правой мыши на номере окне в дереве окон и выберите пункт **[Settings].**

✹ Способ 2
EasyBuilder Pro / [Window] / [Open Window] щелкните по нужному окну и, затем, выберите **[Settings].**

✹ Способ 3
В окне, щелчком правой кнопки по выбранному объекту откройте меню и выберите **[Attribute]**.



6.2.2 Открытие, закрытие и удаление окна

Для открытия существующего окна можно использовать:

✹ Двойной щелчок по номеру окна в дереве окон

✹ В дереве окон выбрать окно и, затем, щелчком правой кнопкой мыши открыть меню и выборать **[Open].**

Аналогично выполняется закрытие или удаление существующего окна.

Обратите внимание, что удаляемое окно должно быть сначала закрыто.

Глава 7. Архив событий

7.1 Управление регистрацией событий

Alarm Bar / Alarm Display / Event Display

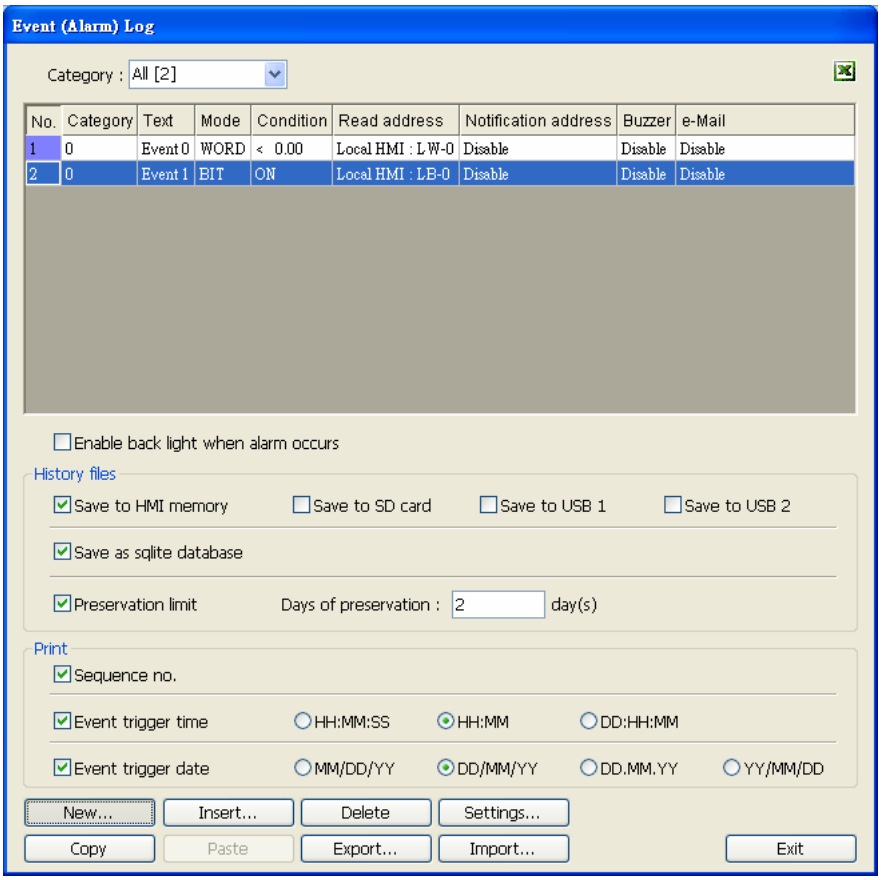
Используя эти объекты - можно проследить весь ход события, начиная от его возникновения, ожидания обработки и удаления аварийного сигнала (аларма). Перед использованием данных объектов, сначала должен быть определен состав события.

Category]

EasyBuilder Pro классифицирует все события по категориям от 0 до 255. Выберите одну категория чтобы добавить или просмотреть *Event Log.* Цифра в скобках **[2],** показывает сколько событий находится в этой категории.

[History files]

Определяет устройство хранения *Event Log*.
Однако во время имитации проекта на ПК, файлы будут сохраняться в одну и ту же папку, находящуюся в установочной директории.



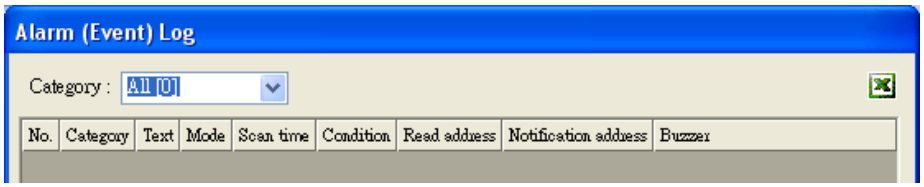
Порядок нумерации USBнакопителей: первому подключенному устройству, присваивается номер 1, следующему номер 2. вне связи с положением USB-порта.

[Preservation limit] – Предельный срок сохранения
Эта настройка определяет предельный срок сохранения данных. Например, если **[Days of preservation]** равен двумя — то это означает, что в памяти панели оператора будут находиться данные о вчерашних и позавчерашних событиях. Данные о событиях, произошедших в другой период времени, будут автоматически удалены для предотвращения переполнения памяти.

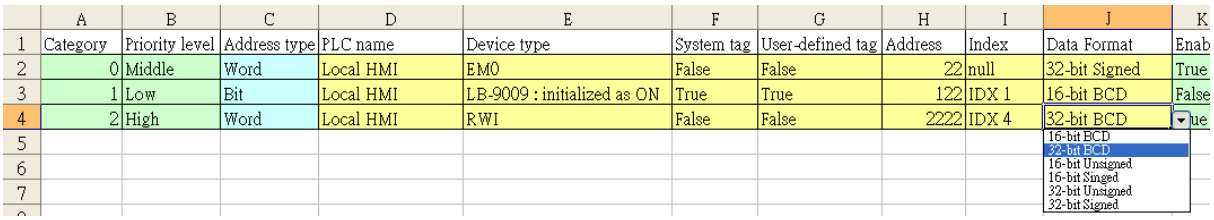
[Print]
Необходимо установить принтер на закладке **[System Parameter Settings] / [Model]** и тогда сообщения о событиях будут выводиться на печать.

7.1.1 Редактирование в Excel

Используйте **Excel** для редактирования *Event Log*



В правом верхнем углу диалогового окна **[Alarm (Event Log)]** находится пиктограмма **Excel**. Щелкните по этой пиктограмме, чтобы открыть шаблон для редактирования их в Excel
Этот шаблон находится в установочной директории - **EventLogExample.xls.**
В шаблоне предусмотрены ниспадающие списки и механизм подтверждения.



	A	B	C	D	E	F	G	H	I	J	K
1	Category	Priority level	Address type	PLC name	Device type	System tag	User-defined tag	Address	Index	Data Format	Enab
2	0	Middle	Word	Local HMI	EM0	False	False	22	null	32-bit Signed	True
3	1	Low	Bit	Local HMI	LB-9009 : initialized as ON	True	True	122	IDX 1	16-bit BCD	False
4	2	High	Word	Local HMI	RWI	False	False	2222	IDX 4	32-bit BCD	ue
5										16-bit BCD	
6										32-bit BCD	
7										16-bit Unsigned	
										16-bit Singed	
										32-bit Unsigned	
										32-bit Signed	

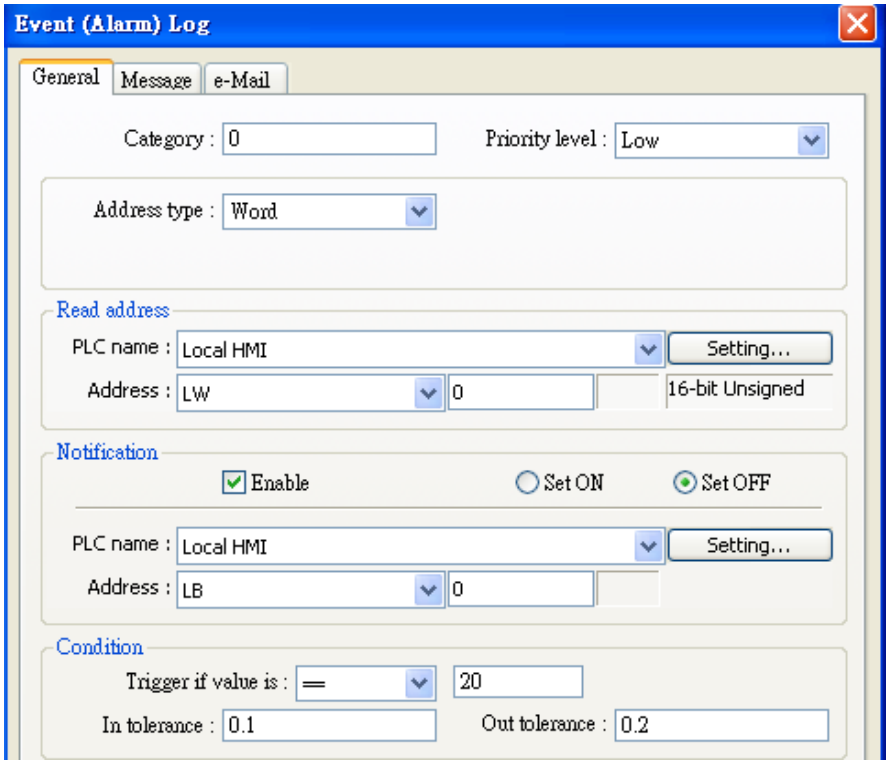
- В полях [System tag] и [User-defined tag] одновременно не может быть установлено значение «true». Если все же в обоих из них установлено значение «true», то значение [User-defined tag] автоматически изменится на «false». Если поле [Device type] должно быть установлено как [User-defined tag] (тэг определяется пользователем), то установите значение «false» в поле [System tag].
- Формат цвета — RGB, где цветовые составляющие (красный, зеленый, голубой) определяются целым числом от 0 до 255
- Когда значение поля [User-defined tag] в Excel равно «true» и адрес тэга, определенный в поле [Device type] не соответствует тэгу определенному пользователем, то в поле [User-defined tag] будет автоматически установлено значение «false»
- Перед импортированием библиотек Label Library / Sound Library, убедитесь, что библиотеки с данными именами есть в системе, иначе система будет использовать имя импортированного файла Excel.

7.2 Создание нового Архива событий

7.2.1 Основные настройки Alarm (Event) Log

Щелкните **[New]** — появится диалоговое окно **[Alarm (Event) Log]** с тремя закладками.

Закладка [General] – основные настройки:



[Category]

Выберите категорию события, 0 - 255.

[Priority level] – Уровень приоритета события

Когда число создаваемых *Event Log* равно максимально возможному, то события с наименьшим приоритетом будут замещаться новыми.

[Read address]

Считывая значение параметра по данному адресу, система проверяет, выполнены ли условия наступления события

[Notification] – Уведомление о событии.

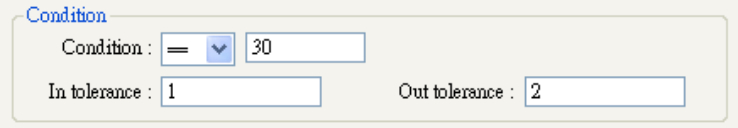
При активации система устанавливает указанный регистр из OFF в ON при возникновении события.

[Condition] - Условие наступления события.

Когда условие имеет тип **[Bit]**, то *Event Log* будет анализировать состояние бита по адресу.

Когда выбрано условие события **[Word]**, то *Event Log* будет анализировать значение в адресе слова: равно ли, больше ли или меньше ли указанного значения.

Пример 1:



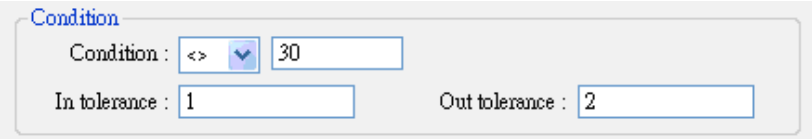
Настройки показанные на рисунке следующие:
Когда значение по адресу в области **[Read address]** больше или равно 29 (=30-1) или меньше или равно 31 (=30+1), то событие инициализируется.
Условие возникновения события (trigger condition):

29 <=значение в [Read address] <=31

После наступления события система будет считаться вернувшейся в нормальное состояние при значении **[Read address]** большем 32 (=30+2) или меньшем 28.

Значение в [Read address] < 28 ИЛИ значение в [Read address] >32

Пример 2:



Настройки показанные на рисунке следующие:
Когда, значении параметра по адресу [Read address] меньше чем 29 (=30-1) или больше чем 31 (=30+1). то событие считается наступившим. Условие возникновения события (trigger condition):

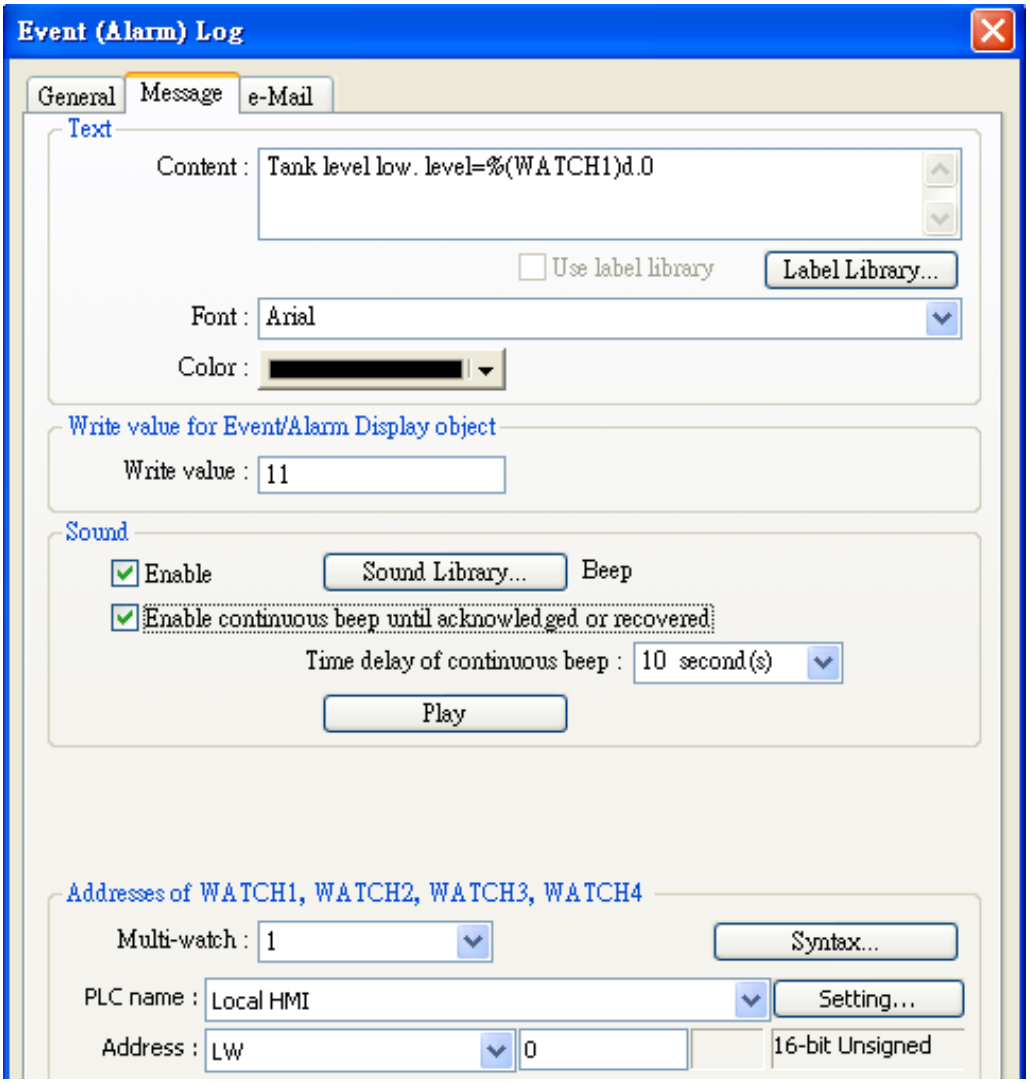
Значение в [Read address] <29 ИЛИ значение в [Read address] >31

После наступления события система будет считаться вернувшейся в нормальное состояние, если значение **[Read address]** больше или равно 28 (=30-2) и меньше или равно 32 (=30+2).

28 <= значение в [Read address] <= 32

7.2.2 Настройки сообщения Alarm (Event) Log

Закладка **[Message] в Alarm (Event) Log:**



Text - [Content]

Текстовое содержимое *Event Log* , отображаемое в объектах типа [alarm bar], [alarm display] и [event display].
Примеры на следующей странице.

Text - [Font][Color]

Шрифт и цвет могут быть назначены различным для каждого события.
Этими настройками определяется шрифт и цвет в объектах [alarm bar], [alarm display] или [event display].

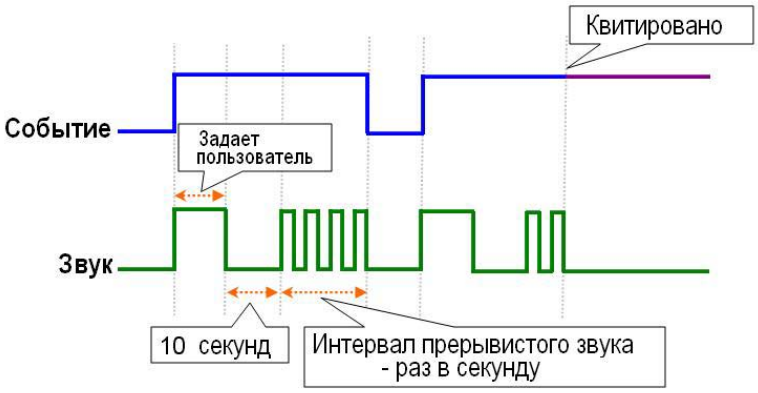
[Write value for Event/Alarm Display object] – Записать значение при событии на экранном объекте.
При квитировании события в объекте типа [event display] или [alarm display] , значение будет записано по указанному адресу.

[Sound] – Звуковой сигнал

Можно использовать звуковой сигнал при возникновении события. Непрерывный звуковой сигнал (beep) можно настроить, чтобы он прекращался только после квитирования или регенерировании.

При использовании непрерывного звукового сигнала для *Event Log,* можно настроить время задержки (delay) между возникновением аварийного события и включением звукового сигнала.

На рисунке показано соотношение события и звукового сигнала.



Пример 1:

Данные по адресу LW возникшего события могут быть вставлены в содержимое (content):
Формат:: **%#d** (где, %: начальный символ, #: адрес, d: конечный символ)

Если LW20 = 13, то событие произошло
Настройка: “High Temperature = %20d”, то на экране будет: “High Temperature = 13”.

Пример 2:

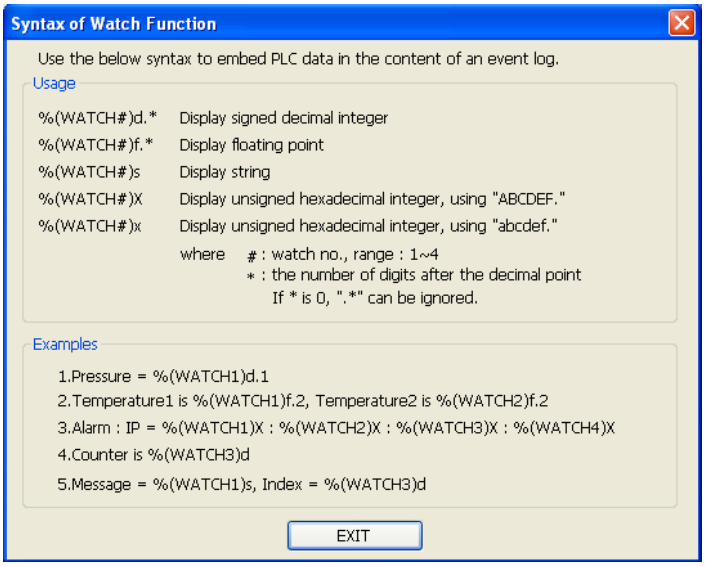
При возникновении события данные из указанного типа устройства могут быть, также, вставлены в содержимое. Тип устройства (device type) должен быть тем же самым что и в **[Read address]** в *Event Log* , например –MW.

Формат: **$#d** (где, $: начальный символ, #: адрес, d: конечный символ)
Если MW-15 = 42, то событие произошло
Настройка: “High Temperature = $15d”, то на экране будет: “High Temperature = 42”.

[Address of Watch] – адрес для наблюдения

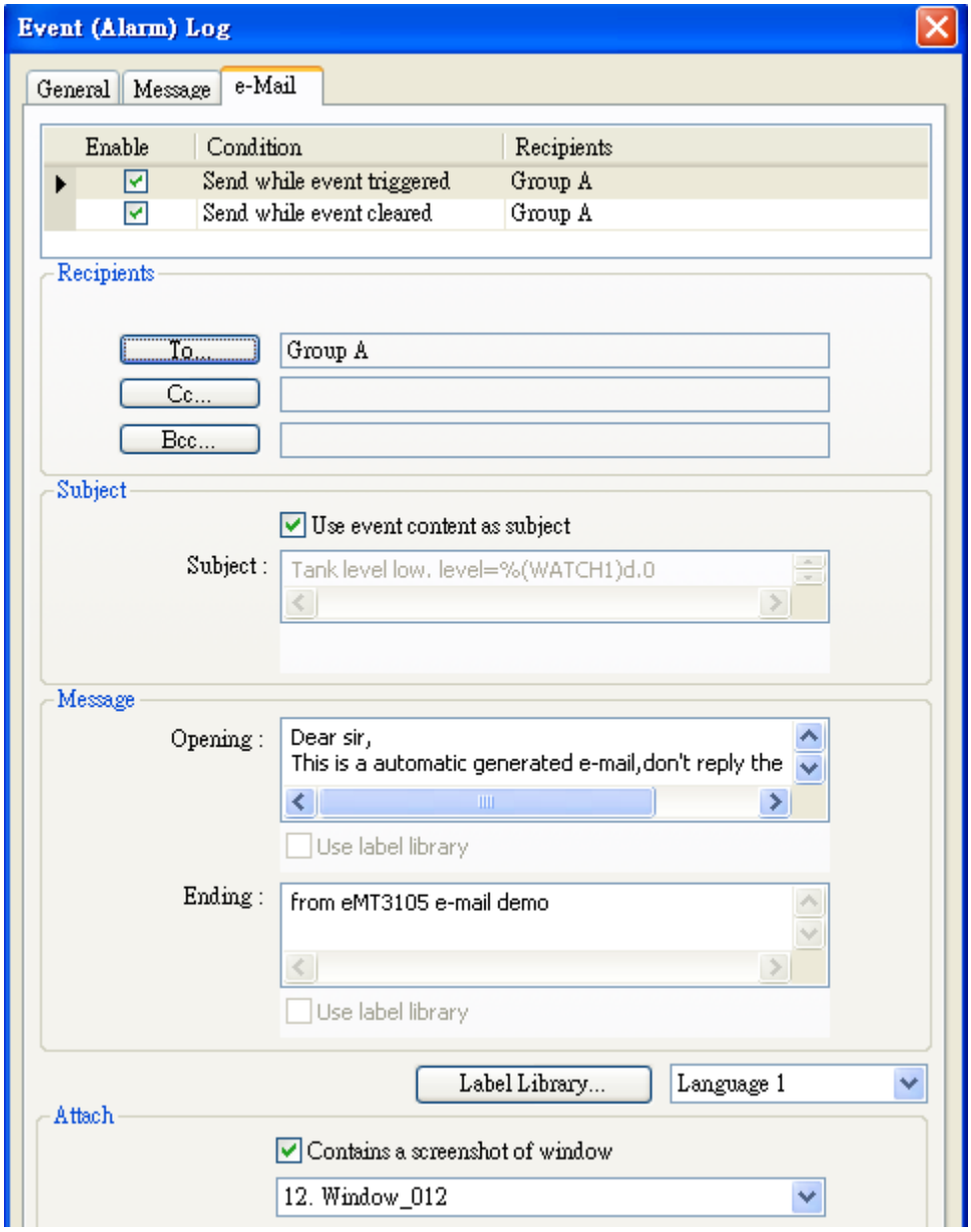
Щелкните по **[Syntax],** чтобы отбразить и редактировать значение «адреса наблюдения» возникновения события.

До четырех «адресов наблюдения» может быть настроено.



7.2.3 Настройки e- Mail Alarm (Event) Log

Закладка **[e- Mail] в Alarm (Event) Log:**



[Recipients] - Получатели
Выберите [To], [Cc], и [Bcc] получателей

[Subject] - Тема
Введите тему e-mail.

[Message] - Сообщение
Enter the content of [Opening] and [Ending] of the e-mail.

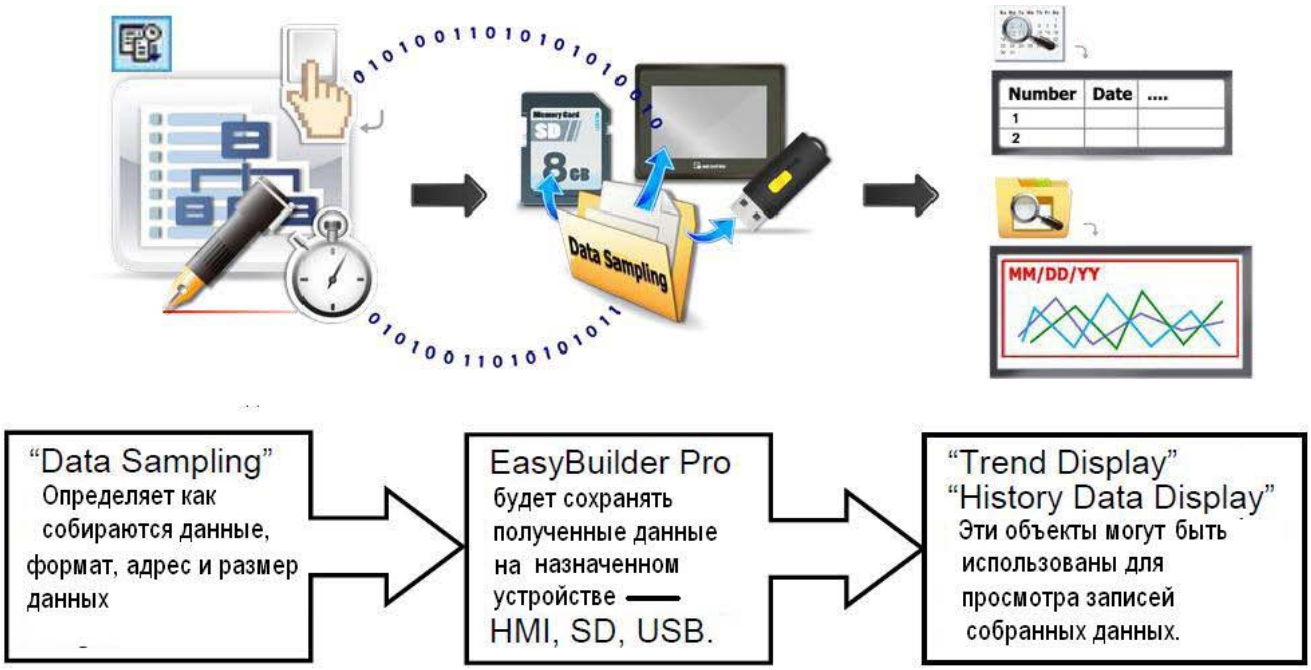
[Attach] - Вложение
Если помечено, выбранное окно будет послано, как вложение - *attachment.*

7.3 Системные регистры для Архива событий

EasyBuilder Pro предоставляет следующие служебных регистров для управления Архивом событий - *event log*:

Адрес	Описание
[LB-9021]	Сброс текущего Архива событий - *event log* (если установлено ON)
[LB-9022]	Удалить самый ранний файл *event log* из памяти панели (если установлено ON)
[LB-9023]	Удалить все файлы *event log* из памяти панели (если установлено ON)
[LB-9024]	Обновить информацию *event log* в памяти панели (если установлено ON)
[LB-9034]	Сохранить опрос данных (*event/data sampling*) в панели, на USB, на карте SD (если установлено ON)
[LB-9042]	Квитировать все аварийные события - *alarm events* (если установлено ON)
[LB-9043]	Существуют не квитированные события (если ON)
[LB-11940]	Удалить самый ранний файл *event log* на карте SD (если установлено ON)
[LB-11941]	Удалить все файлы *event log* из карты SD card (если установлено ON)
[LB-11942]	Обновить информацию *event log* на карте SD (если установлено ON)
[LB-11943]	Удалить самый ранний файл *event log* из USB 1 (если установлено ON)
[LB-11944]	Удалить все файлы *event log* из USB 1 (если установлено ON)
[LB-11945]	Обновить информацию *event log* на USB 1 (если установлено ON)
[LB-11946]	Удалить самый ранний файл *event log* из USB 2 (если установлено ON)
[LB-11947]	Удалить все файлы *event log* из USB 2 (если установлено ON)
[LB-11948]	Обновить информацию *event log* на USB 2 (если установлено ON)
[LW-9060]	(16bit): Число файлов *event log* в памяти панели
[LW-9061]	(32bit): Размер файлов *event log* в памяти панели
[LW-9450]	(16bit): Тэг времени *event log* - секунды
[LW-9451]	(16bit): Тэг времени *event log* – минуты
[LW-9452]	(16bit): Тэг времени *event log* - часы
[LW-9453]	(16bit): Тэг времени *event log* - день
[LW-9454]	(16bit): Тэг времени *event log* - месяц
[LW-9455]	(16bit): Тэг времени *event log* - год
[LW-10480]	(16bit): Число файлов *event log* на карте SD
[LW-10481]	(32bit): Размер файлов *event log* на карте SD card
[LW-10483]	(16bit): Число файлов *event log* на USB 1
[LW-10484]	(32bit): Размер файлов *event log* на USB 1
[LW-10486]	(16bit): Число файлов *event log* на USB 2
[LW-10487]	(32bit): Размер файлов *event log* на USB 2

Глава 8. Сбор данных

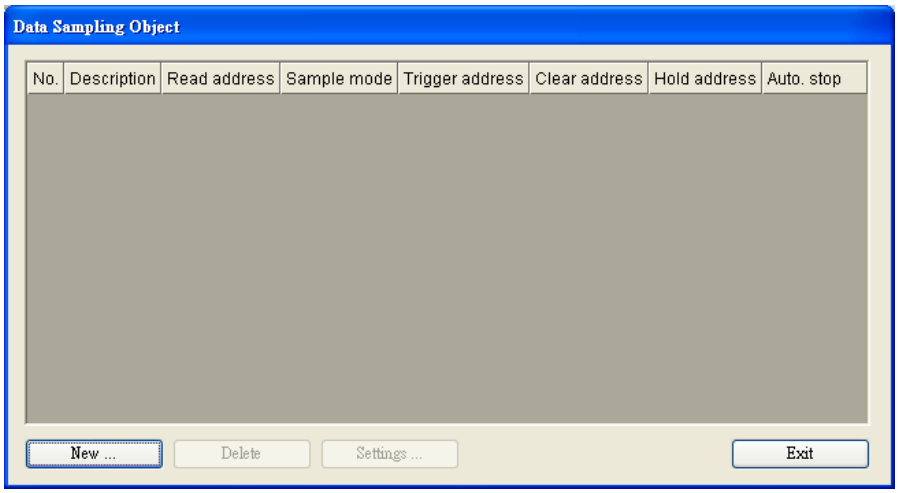


8.1 Управление сбором данных

Перед использованием объектов типа [Trend display] или [History data display]
для просмотра содержимого [Data Sampling], должен быть определен метод сбора данных.

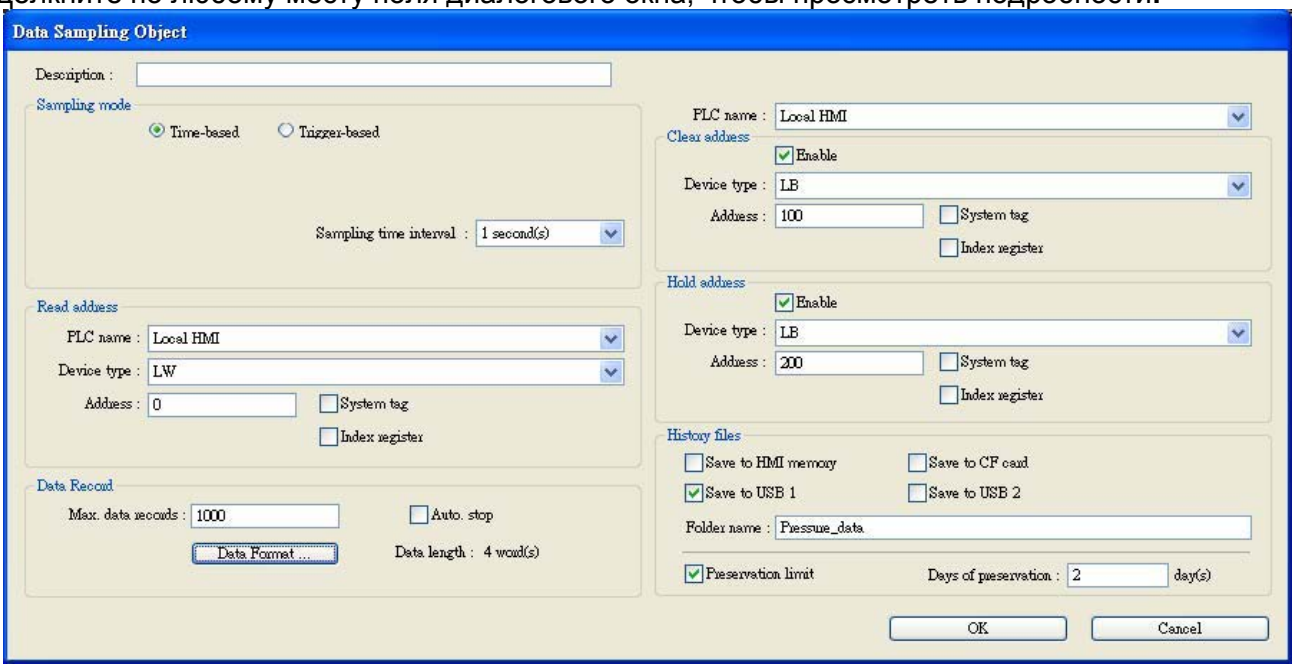
1. Щелкните по пиктограмме объекта — [Data Sampling].

2. Щелкните по **[New]**, чтобы произвести настройки

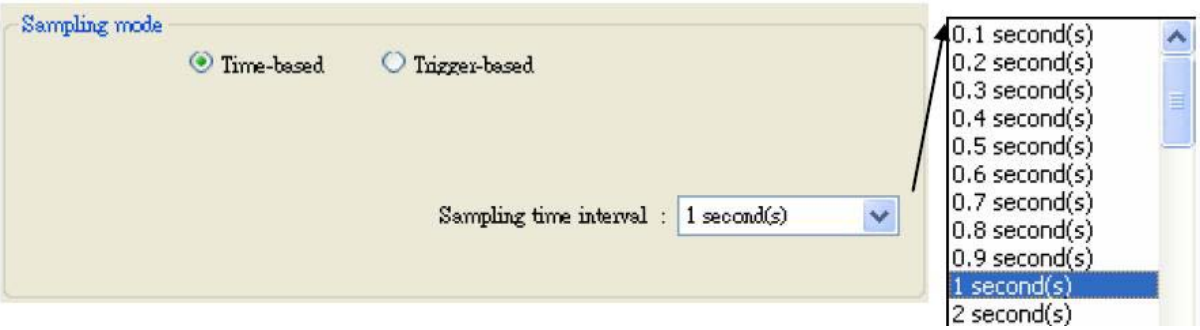


8.2 Создание новой выборки данных

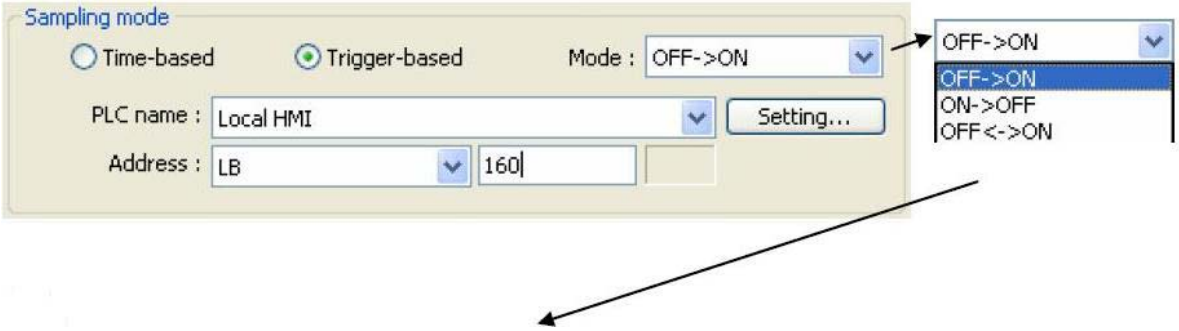
Щелкните по любому месту поля диалогового окна, чтобы просмотреть подробности:



[Sampling mode] - режим опроса.
[Time-based] - режим опроса с фиксированной частотой. Временной интервал опроса **[Sampling time interval]**.может быть установлен в диапазоне от 0.1 секунды до 120 минут.



[Trigger-based] -по условию. Если выбран режим “Trigger-based”, то для инициализации процесса опроса данных используется состояние заданного адреса.



[Mode]
Условия инициализации процесса сбора данных:

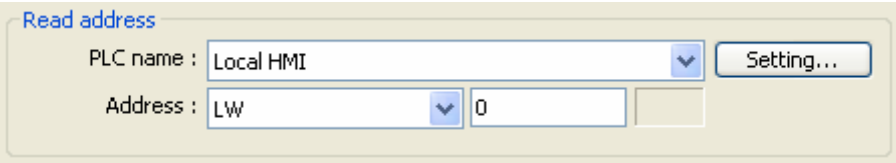
[OFF → ON] Процесс запускается при изменении состояния с [OFF] на [ON]

[ON → OFF] Процесс запускается при изменении состояния с [ON] на [OFF]

[ON←→ OFF] Процесс запускается при любом изменении состояния данного бита

[Read address]

Выберите тип устройства, из которого производится выборка данных.

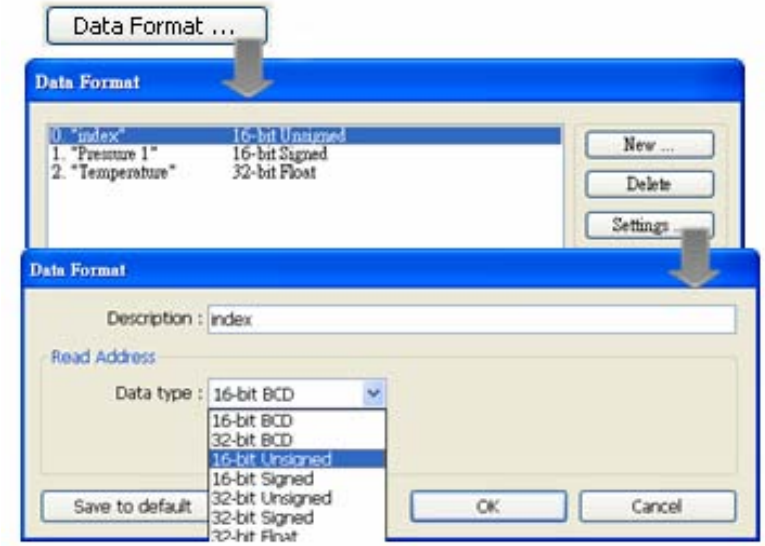


[Data Record]

Максимальное число записей, сохраняемых в выборку одной командой Data sampling.в один день равно 86400. (1 запись в секунду за 24часа). Если **[sampling time interval]** = 0.1 секунде. то максимальное число записей только 86400.

Условие / **Объект**	**[Max. data records]** =10, **[Auto. Stop]** ***не отмечен***	**[Max. data records]** =10, **[Auto. Stop]** ***отмечен***
Trend display – реальное время	На экране будут отображаться 10 самых последних записей	Остановка после отображения 10 значений
Trend display – просмотр архива трендов	Продолжается опрос данных и отображаются все архивные данные.	Остановка после отображения 10 значений
History Data display – просмотр архива данных	Продолжается опрос данных и отображаются все архивные данные.	Остановка после отображения 10 значений
Data sampling – опрос данных	Продолжается запись новых данных.	Остановка опроса после записи 10 значений

[Data Format …]

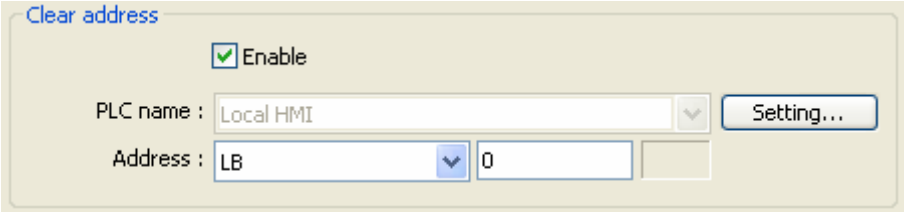


Data sampling может содержать несколько типов записей.
В EasyBuilder Pro *Data Sampling* может сохранять различные типы записей одновременно.
Пользователь может определять содержимое *Data sampling* .
Как показано на рисунке, пользователь определил три типа данных и длину данных в 4 слова.
EasyBuilder Pro будет собирать 4-х словные данные каждый раз из указанного адреса в один *Data sampling.*

✸ Если в режиме имитации Вы захотите изменить формат опрашиваемых данных, сначала удалите предыдущие записи в инсталляционной директории EasyBuilder Pro, чтобы избежать не корректной интрепретации записей данных системой.

[Clear address]

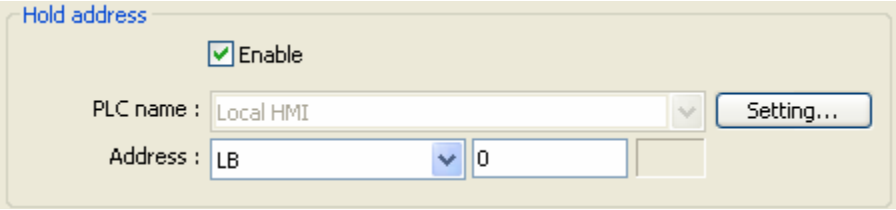
Если состояние назначенного адреса равно «ON», данные полученные объектом **[trend display]** режима реального времени **[real-time]** будут удалены и число данных будет установлено равным 0.



Это не оказывает влияние на данные уже сохраненные в файле.

[Hold address]

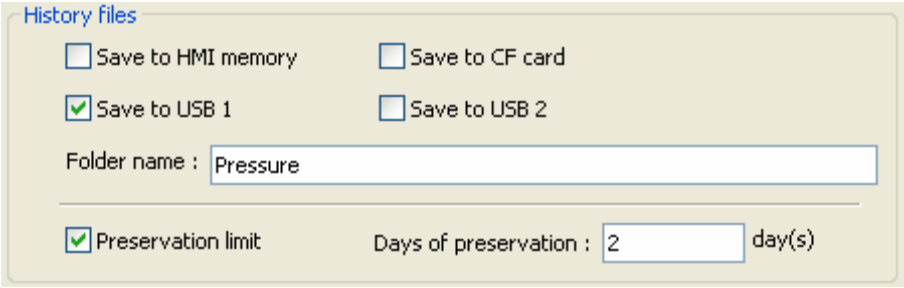
Если состояние указанного адреса будет «ON», то сбор данных приостановиться до тех пор, пока состояние не сменится на «OFF».



[History files]

[Save to HMI memory]

Хранение *Data sampling* во внутренней памяти панели оператора только, если их объем более 4 Кбайт или Используйте бит [LB-9034] для принудительного сохранения.



[Save to SDcard / USB1 / USB2]
Сохранение опрашиваемых данных на указанном устройстве.

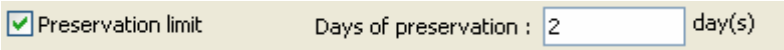
Заметьте, что USB-накопители не назначены, как №1 или №2.
Устройство вставленное первым – это USB1, устройство подключенное следующим будет USB2.

[Folder name]

Определяет имя папки для хранения выборки данных, которое должно состоять из символов **ASCII**.

Имя папки будет записано, как: **[Storage Location] \ [Folder Name] \ yyyymmdd.dtl**

Preservation limit: определяет срок хранения собранных данных.



Если он равен двум дням (как на рисунке вверху), будут сохраняться вчерашние и позавчерашние данные.

Например, если сегодня 1 июля, то будут сохранены данные за 30 и 29 июня и удалены за 28 июня.

. ✹ В режиме имитации на компьютере все опрашиваемые данные, будут сохранены в папке **datalog** в директории **[Storage location]**. .

8.3 Системные регистры для Data Sampling

EasyBuilder Pro предоставляет следующие служебные регистры для управления опросом данных- Data Sampling:

Адрес	Описание
LB-9025	Удалить самый ранний файл *Data Sampling* из памяти панели (если в ON)
LB-9026	Удалить все файлы *Data Sampling* из памяти панели (если в ON)
LB-9027	Обновить информацию *Data Sampling* в памяти панели (если в ON)
LB-9034	Сохранить *event/data sampling* в панели, на USB, на карте SD (если в ON)
LB-11949	Удалить самый ранний файл *Data Sampling* на карте SD (если установлено ON)
LB-11950	Удалить все файлы *Data Sampling* из карты SD card (если установлено ON)
LB-11951	Обновить информацию *Data Sampling* на карте SD (если установлено ON)
LB-11952	Удалить самый ранний файл *Data Sampling* из USB 1 (если установлено ON)
LB-11953	Удалить все файлы *Data Sampling* из USB 1 (если установлено ON)
LB-11954	Обновить информацию *Data Sampling* на USB 1 (если установлено ON)
LB-11955	Удалить самый ранний файл *Data Sampling* из USB 2 (если установлено ON)
LB-11956	Удалить все файлы *Data Sampling* из USB 2 (если установлено ON)
LB-11957	Обновить информацию *Data Sampling* на USB 2 (если установлено ON)
LW-9063	(16bit): Число файлов *Data Sampling* в памяти панели
LW-9064	(32bit): Размер файлов *Data Sampling* в памяти панели
LW-10489	(16bit): Число файлов *Data Sampling* на карте SD
LW-10490	(32bit): Размер файлов *Data Sampling* на карте SD card
LW-10492	(16bit): Число файлов *Data Sampling* на USB 1
LW-10493	(32bit): Размер файлов *Data Sampling* на USB 1
LW-10495	(16bit): Число файлов *Data Sampling* на USB 2
LW-10496	(32bit): Размер файлов *Data Sampling* на USB 2

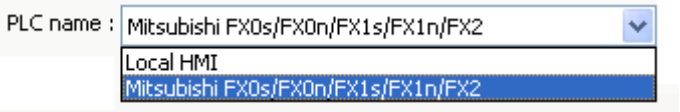
Глава 9. Общие свойства объектов

Настройка общих свойств объектов **[general]** включает:

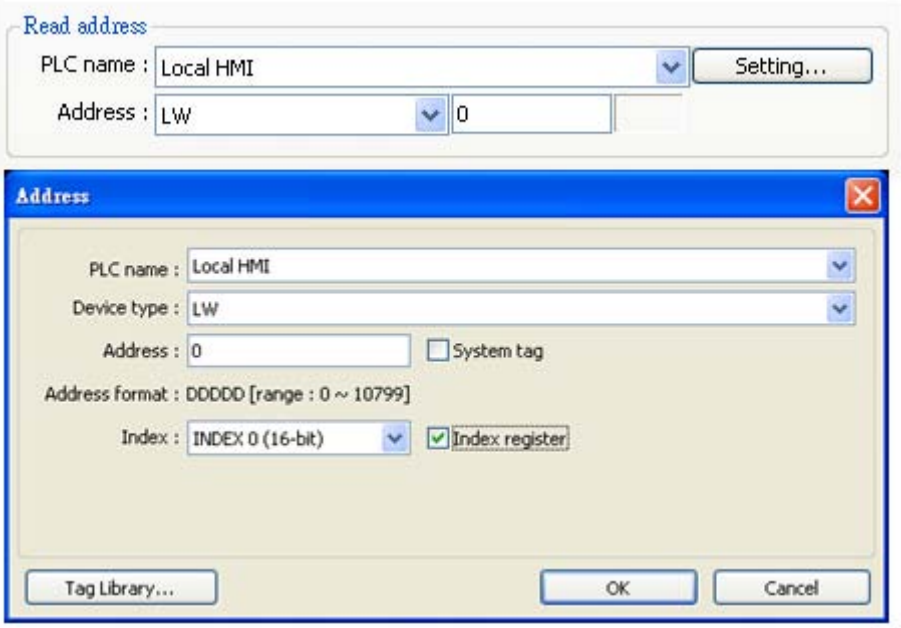
1. Выбор соединения с ПЛК
2. Настройка адресов для чтения и записи
3. Использование библиотеки форм и изображений
4. Настройка текстовой надписи
5. Настройка размеров текста

9.1 Выбор соединения с ПЛК

При использовании некоторых объектов требуется выбор соединения с ПЛК. На рисунке ниже показан список ПЛК **[PLC name]**, с которыми может быть устанавлено соединение. В этом примере доступны два ПЛК: “Local HMI” и “Mitsubishi FX0s/FX0n/FX1s/FX1n/FX2”.
В этот список попадают устройства, заданные в таблице устройств **[Device List]** в окне **[System Parameters Settings]**.



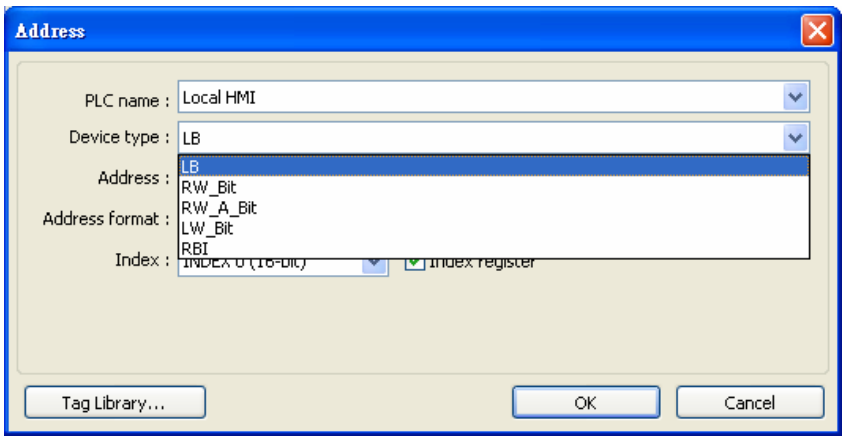
9.1.1 Настройка адресов для считывания и записи



На рисунке выше показаны параметры настройки адресов чтения или записи.

[PLC name]
Это окно предназначено для выбора типа устройства.

У разных ПЛК существуют различные варианты выбора **[Device type]**.



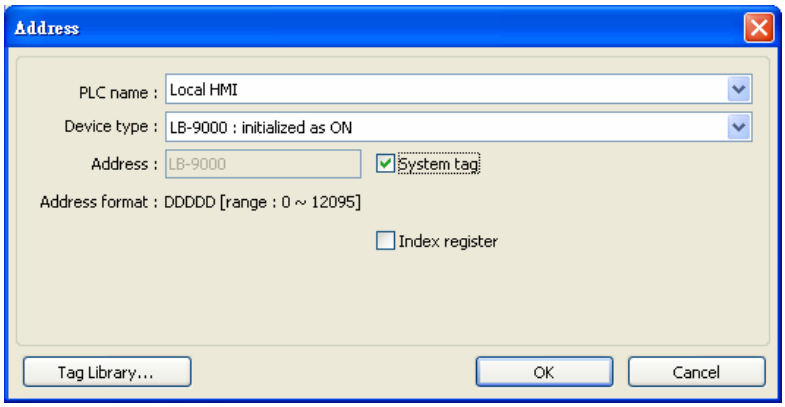
[Address]

Указывает адреса чтения и записи данных.

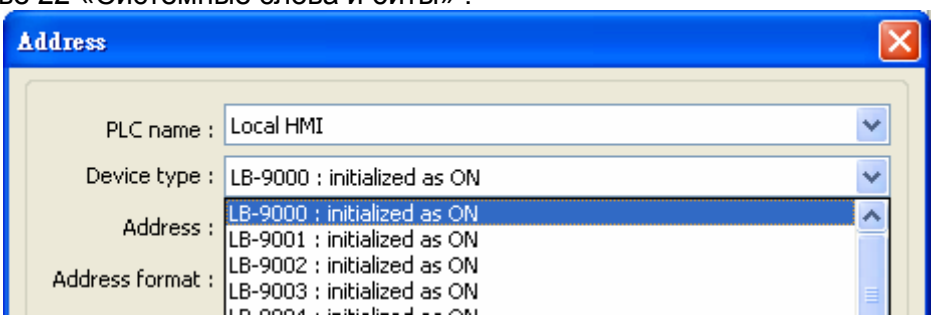
[System tag]

Тэг адреса может быть системным или определяемым пользователем. Щелкните по **[Setting…]** за **[PLC name]** и пометьте **[system tag].** Это позволит применять резервированные системные адреса для конкретных целей. Эти адреса разделены на Биты и Слова (LB или LW).

Если отмечен флаг **[System tag],** то помимо того, что в поле **[Device type]** будет отображаться содержимое системного тэга, в поле **[Address]** будет показан адрес выбранного регистра, как показано ниже.



Ниже показаны некоторые системные тэги. Подробности в Главе 16 «Библиотека адресов тэгов» и в Главе 22 «Системные слова и биты» .

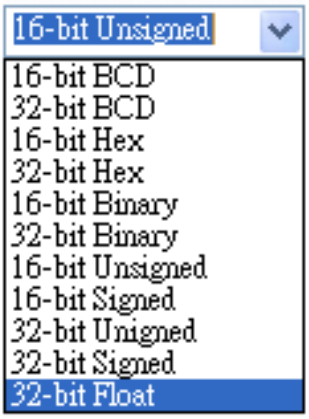


[Index register] – Индексный регистр

За советом о применении или нет Индексного регистра обращайтесь к примерам в Главе 11 «Индексный Регистр».

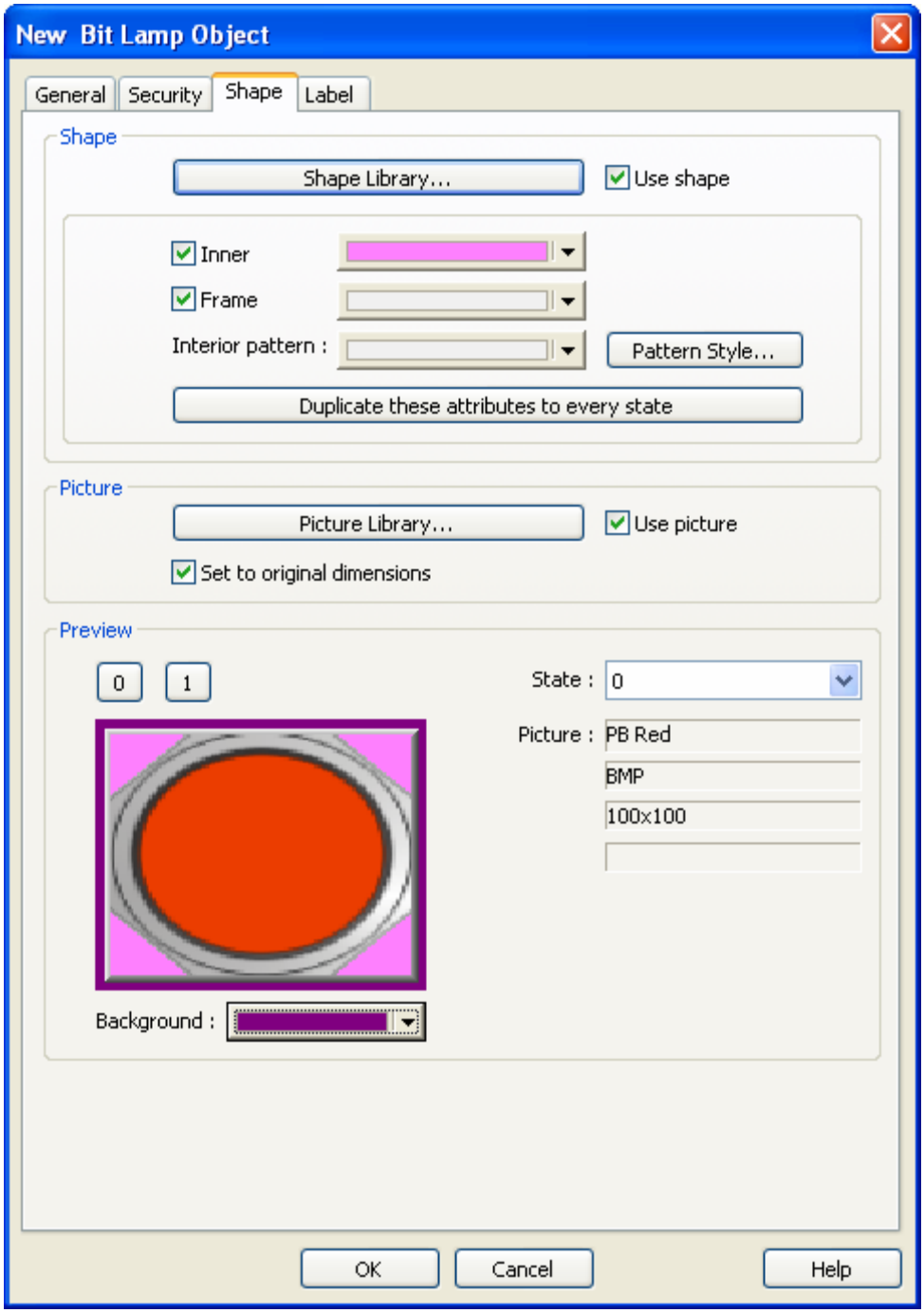
Selecting Data Type – Выбор Типа данных

EasyBuilder Pro поддерживает следующие типы данных. Необходимо выбрать правильный тип данных, особенно при использовании адресных тэгов.



9.2 Использование библиотек форм и изображений

Библиотеки форм **[Shape Library]** и изображений **[Picture Library]** используются в объектах для улучшения визуального эффекта. Для их применения откройте вкладку **[Shape]** в окне добавления нового объекта для настройки Библиотеки форм и Библиотеки изображений.



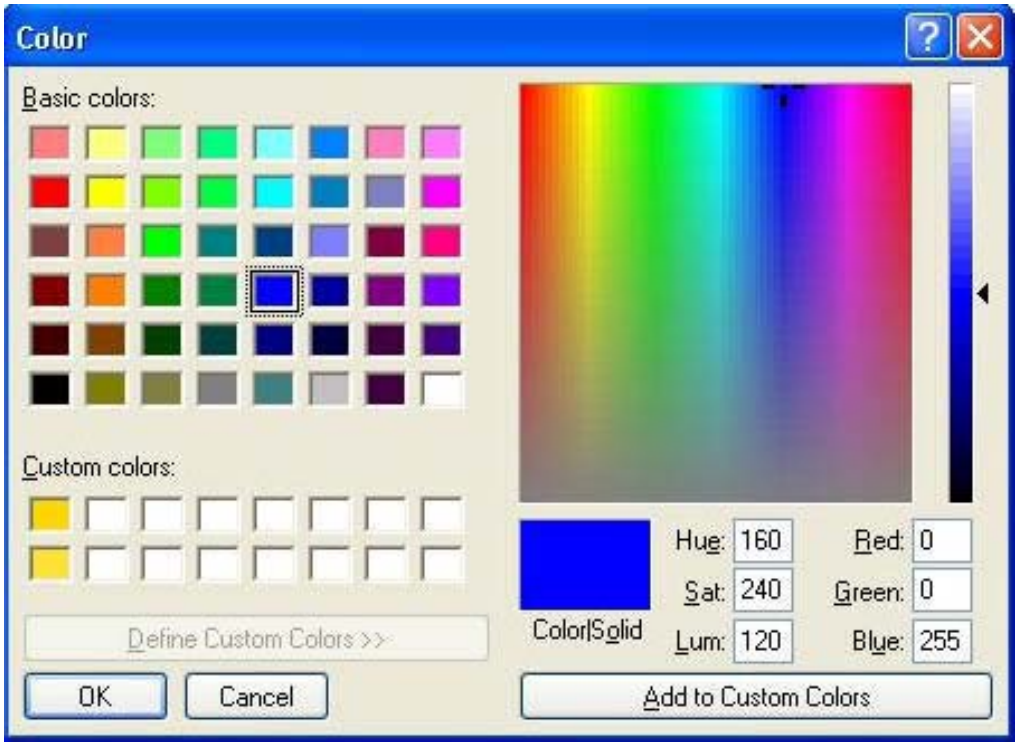
9.2.1 Настройки библиотеки форм

[Shape Library…]

Пометьте **[Use Shape],** чтобы активировать эту настройку и выбрать форму для объекта.

[Inner] – Внутреннее поле объекта

Пометьте **[Inner],** чтобы активировать эту настройку и выбрать цвет для внутренней части объекта. Щелкните по кнопке для открытия диалогового окна **[Color]** и выберите цвет объекта или укажите собственный цвет **[customize],** и затем нажмите **[Add to Custom Colors],** чтобы система сохранила этот цвет.



[Frame] – Рамка

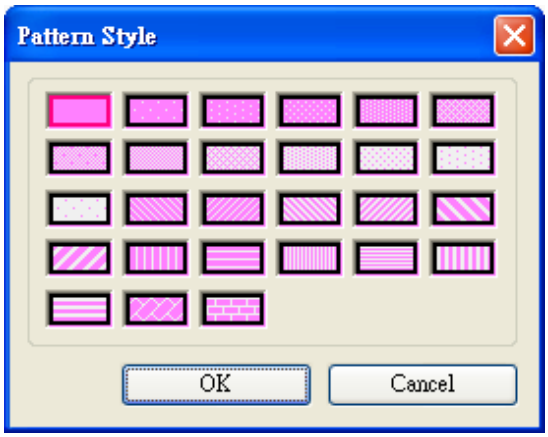
Пометьте **[Frame],** чтобы активировать эту настройку и выбрать цвет [Color] для рамки объекта. Способ настройки такой же, как в предыдущем абзаце.

[Interior Pattern] – Текстура внутреннего поля объекта

Щелкните, чтобы иметь возможность выбрать тип текстуры объекта внутренней области. Можно указать и цвет текстуры.

[Pattern Style] - Тип текстуры внутреннего поля объекта

Нажмите кнопку [Pattern Style] и укажите текстуру в диалоговом окне.

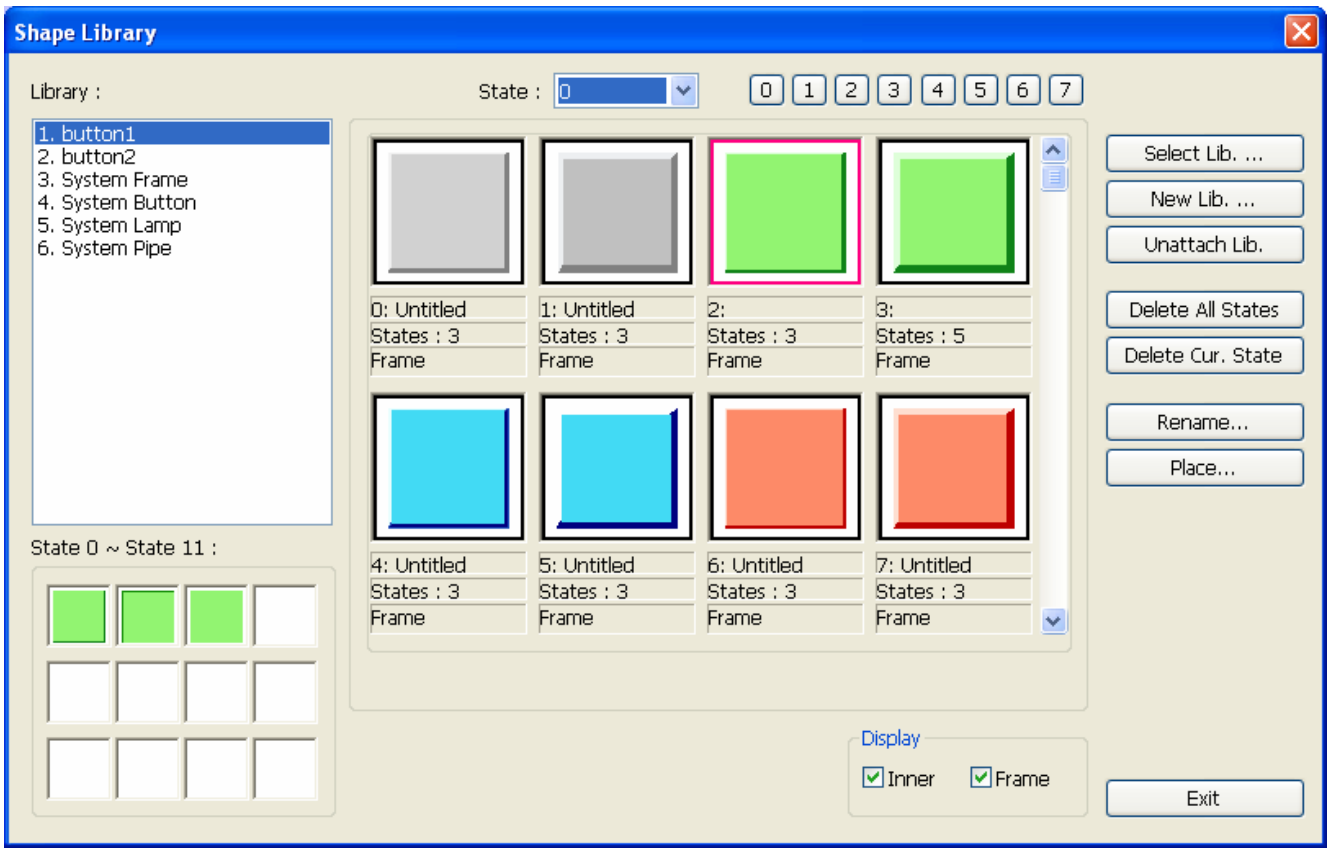


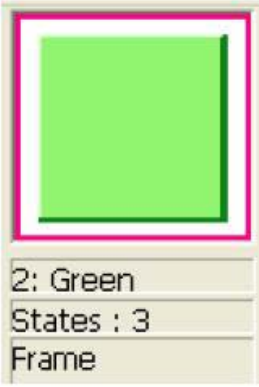
[Duplicate these attributes to every state]

Применить данные настройки текущего состояния для всех других состояний объекта.

Как использовать Библиотеку форм

Щелкните по кнопке [Shape Library…], откроется окно. Выбранная форма будет помечена красной рамкой.

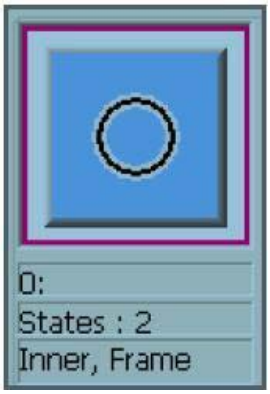




На рисунке вверху показана одна из форм библиотеки со следующими параметрами:

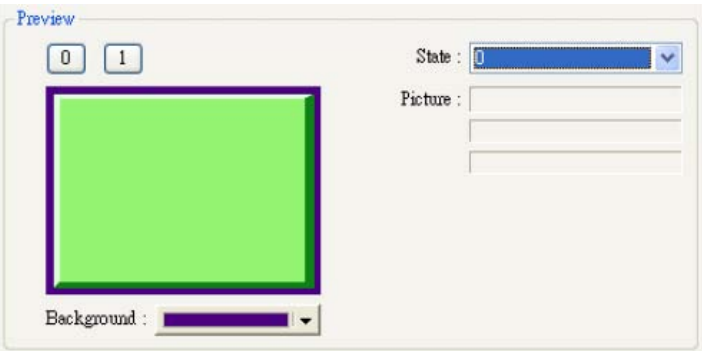
2: Green	Номер и имя формы в библиотеке
States: 3	Число состояний формы. В примере – три состоянния
Frame	Показывает что эта форма использовуется только с рамкой

На рисунке внизу показана одна из форм библиотеки с параметрами “inner” and “frame.”



Примечание: Описание всех настроек Библиотеки форм представлены в иллюстрациях Главы 14 «Библиотека Форм и Библиотека изображений»

Щелкните **[OK]** и просмотрите отредактированную форму после завершения настроек.



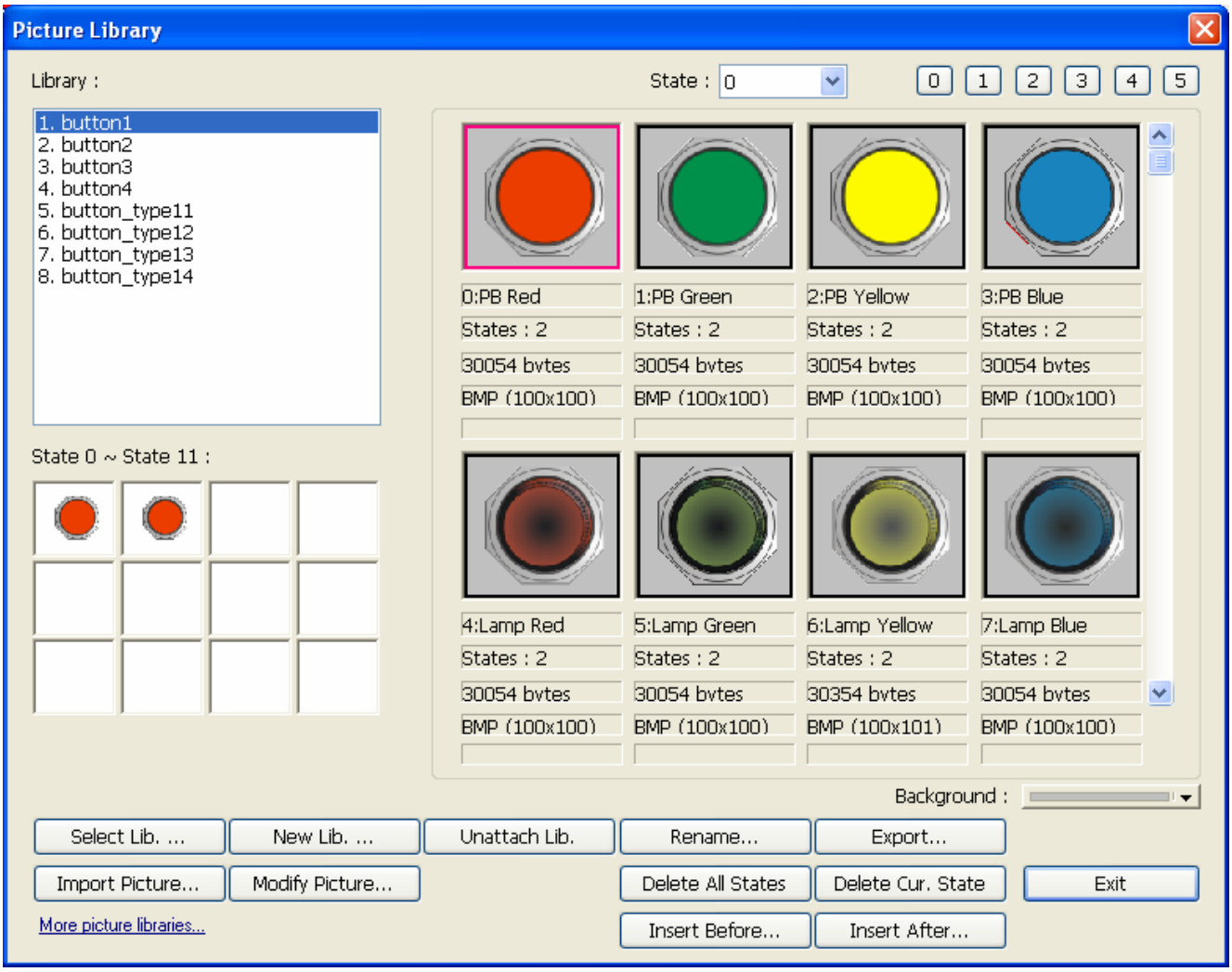
9.2.2 Настройки библиотеки изображений

[Picture Library]

Щелкните по **[Use picture]**, чтобы активировать выбор изображений из библиотеки

Как использовать Библиотеку изображений

Нажмите кнопку **[Picture Library…]** - откроется диалоговое окно, показанное ниже. Выбранное изображение отмечена красной рамкой.



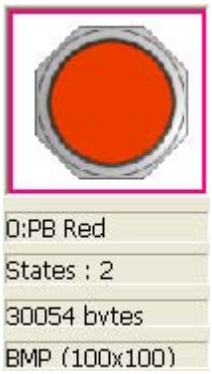


Рисунок выше содержит информацию об одном из изображений в Библиотеке изображений:

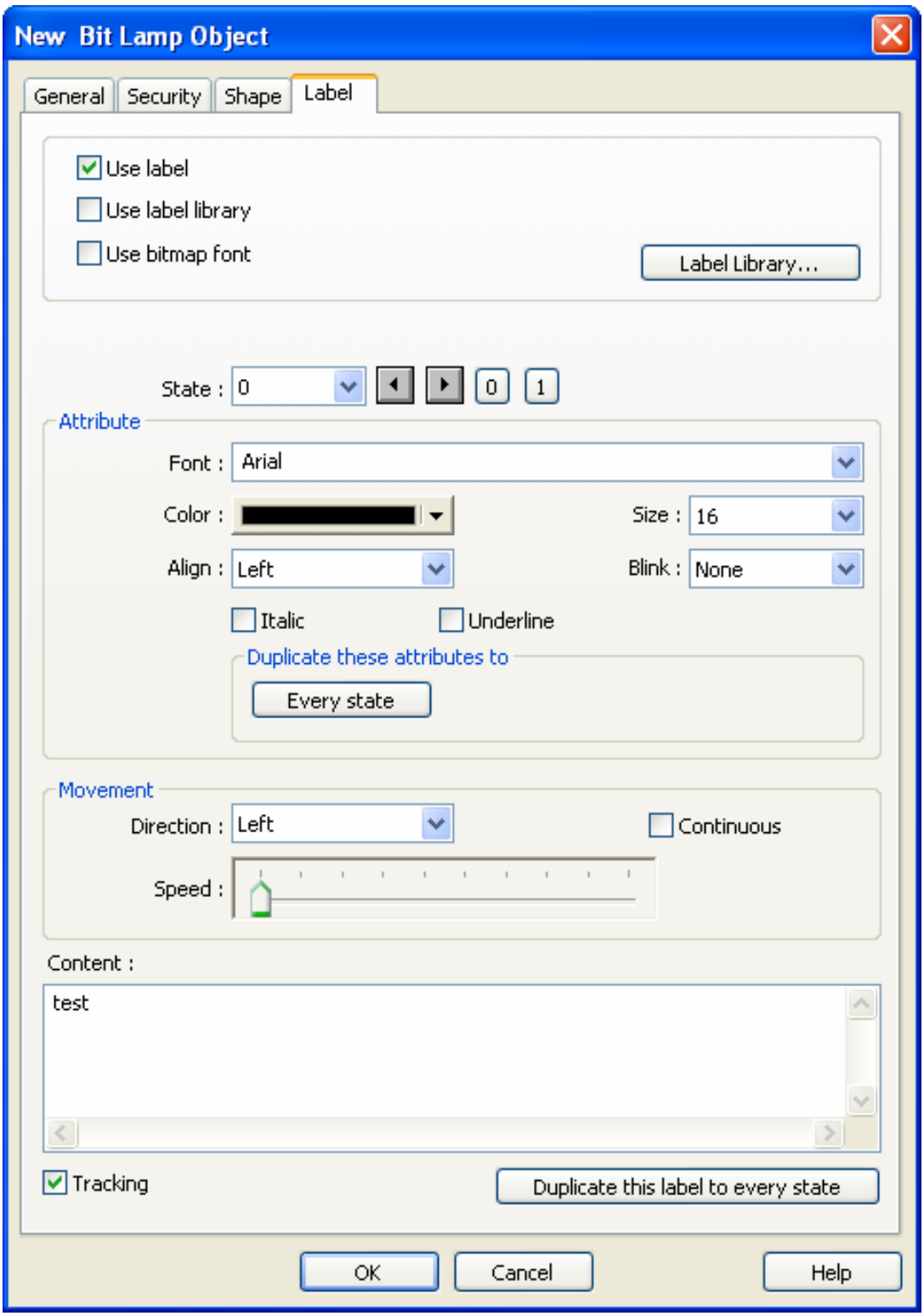
Picture name	0 : PB Red	Номер и название изображения
Total states	2	Количество состояний данного изображения
Image size	30054 bytes	Размер изображения
Image format	BMP (100x100)	Формат изображения и разрешение; могут быть форматы BMP,JPG, PNG, DPD или GIF. Размер:100 пиксел ширина и 100 пиксел высота (в данном случае)

Обращайтесь к рисункам в Главе 14 «Библиотека форм и Библиотека изображений», чтобы подробнее узнать обо всех настройках в диалоговом окне [Picture Library].

Щелкните [OK] и просмотрите [preview] результат после завершения всех настроек.

9.3 Настройка текстовой надписи

Зайдите на вкладку **[Label]** диалогового окна добавленя нового объекта для задания текстового содержимого объекта.

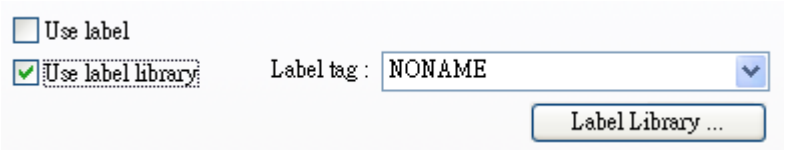


[Use label] – Использовать надпись

Отметьте [Use label] и нажмите кнопку **[Label Library]** для добавления и редактирования текста. EasyBuilder Pro поддерживает шрифты Windows типа «true».

[Use label library] – использовать библиотеку надписей

Отметьте [Use label library], чтобы выбрать существующий текстовый тэг в библиотеке надписей, как показано ниже .



[Label Library…]

См. рисунки в Главе 16 «Библиотека надписей и использование многоязычия» для просмотра всех деталей.

[Font]

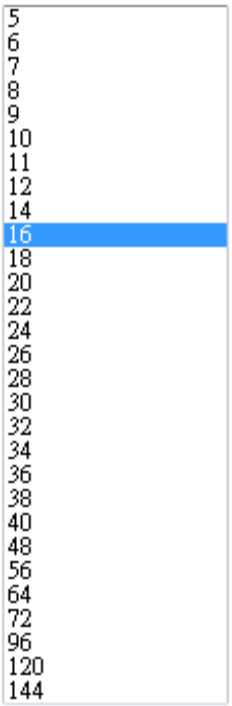
Выберите стиль шрифта из раскрывающегося списка. EasyBuilder Pro поддерживает шрифты Windows типа «true»

[Color]

Выберите текст шрифта текста.

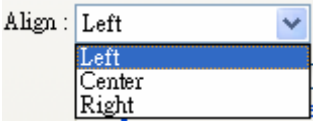
[Size]

Выбор размера шрифта. EasyBuilder Pro поддерживает размеры текста перечисленные справа.



[Align]

Выберите способ выравнивания многострочных надписей.



Текст выровнен по левой стороне - “Left”:

111
222222
333333333

Текст выровнен по центру -“Center”:

111
222222
333333333

Текст выровнен по правой стороне -“Right”:

111
222222
333333333

[Blink]
Задание эффекта мигания текста:
Выберите **[None]**, чтобы отключить эффект или задайте период мигания **[1 second]** или **[0.5 seconds]**.

[Italic]
Использование курсивного начертания.

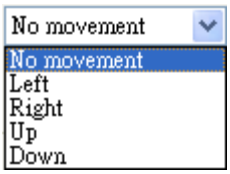
Italic Label

[Underline]
Использование подчеркивания

Underline Label

[Movement] - Настройки перемещения

[Direction] - Направление
Установление направления бегущей строки.



Возможны два варианта отображения бегущей строки:

[Continuous] - непрерывное

Когда флаг [Continuous] не отмечен, следующий фрагмент текстовой надписи появляется только после полного исчезновения предыдущего. См. рисунок ниже:

Когда флаг [Continuous] отмечен, текст будет выводиться непрерывно. См. рисунок ниже:

[Speed]

Настройка скорости перемещения текста.

[Content]

Задание содержания надписи. При использовании Библиотеки надписей **[Label Library]**, содержание будет импортировано из нее.

[Tracking]

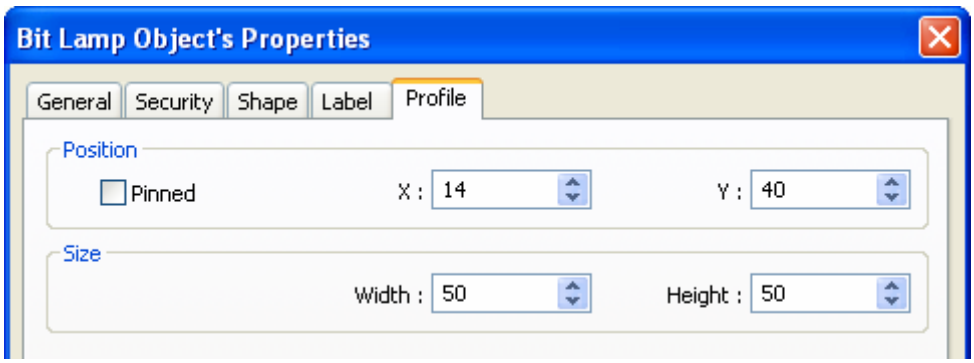
Когда отмечен этот флаг движение текста в одном состоянии будет вызывать движение текстовой надписи в других состояниях.

[Duplicate this label to other states]

Функция распространения текстовой надписи на все состояния.

9.4 Настройка размеров профиля

После создания объекта [Bit Lamp] дважды щелкните по нему. Выберите вкладку [Profile] в окне [Bit Lamp Object's properties] для настройки расположения и размеров объекта.



a. Position - Позиция

[Pinned] — фиксированное положение и размер объекта — когда отмечено, данные параметры объекта недоступны для изменения.

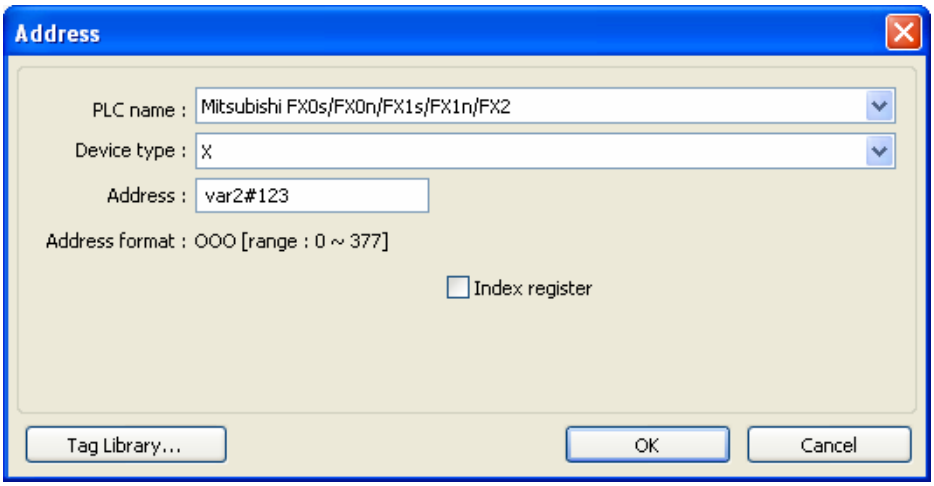
X и **Y** соответствуют координатам левого верхнего угла объекта.

b. Size – Размер

Задание ширины **[width]** и высоты **[height]** объекта.

9.4 Переменные для номера станции

EasyBuilder Pro разрешает пользователю устанавливать переменую для адреса станции ПЛК. Как показано ниже, переменная var2 — одна из 16 доступных переменных номера станции.



Синтаксис номера станции:

varN#address

N — целое из диапазона 0~15;

address — это адрес ПЛК.

Доступны 16 переменных: var0 ~ var15. Данные переменные номера станции считывают значения из адресов LW10000~LW10015.

Ниже приведено соответствие переменных и зарезервированного системой адреса LW:

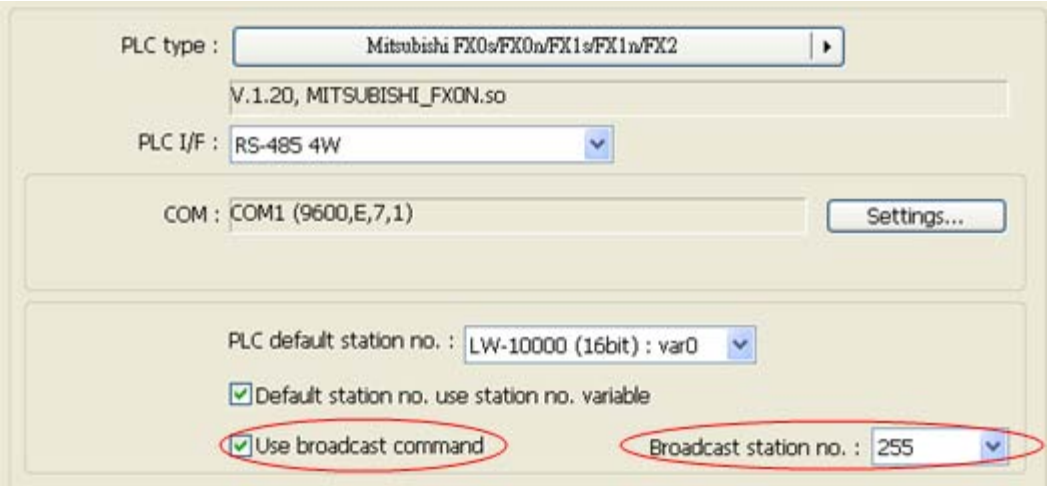
var0	LW10000
var1	LW10001
var2	LW10002
var3	LW10003
var4	LW10004
var5	LW10005
var6	LW10006
var7	LW10007
var8	LW10008
var9	LW10009
var10	LW10010
var11	LW10011
var12	LW10012
var13	LW10013
var14	LW10014
var15	LW10015

Например, «var0» считывает значение из LW10000; если по этому адресу находится число 32, то var0#234 = 32#234 (номер станции — 32); аналогично «var13» использует значение по адресу LW10013; если по нему записано 5, то var13#234 = 5#234.

9.5 Широковещательный номер станции

Панели оператора позволяют использовать широковещательные команды двумя способами.

Первый – напрямую установить параметры в закладке **[Device] в [system parameter settings]**:



Второй — использовать системные биты для включения/отключения широковещательного номера станции или изменения номера станции.

Соответствующие системные биты приведены ниже:

LB9065	Вкл/выкл Широковещательный номер станции порта COM1
LB9066	Вкл/выкл Широковещательный номер станции порта COM2
LB9067	Вкл/выкл Широковещательный номер станции порта COM3
LW9565	Широковещательный номер станции порта COM 1 (broadcast station no.)
LW9566	Широковещательный номер станции порта COM 2 (broadcast station no.)
LW9567	Широковещательный номер станции порта COM 3 (broadcast station no.)

Глава 10. Защита объектов

В EasyBuilder Pro настройка паролей пользователя возможна в двух режимах:

1. General Mode - Общий режим

2. Enhanced Security Mode – Режим повышенной безопасности

Защита объектов включает:
1. Пароль пользователя с доступом к объектам определенных классов.
2. Настройки безопасности для каждого объекта.

10.1 Пароли пользователя и классы доступа объекта

[System Parameter Settings] / [Security]: Механизм безопасности в EasyBuilder Pro включает два режима:

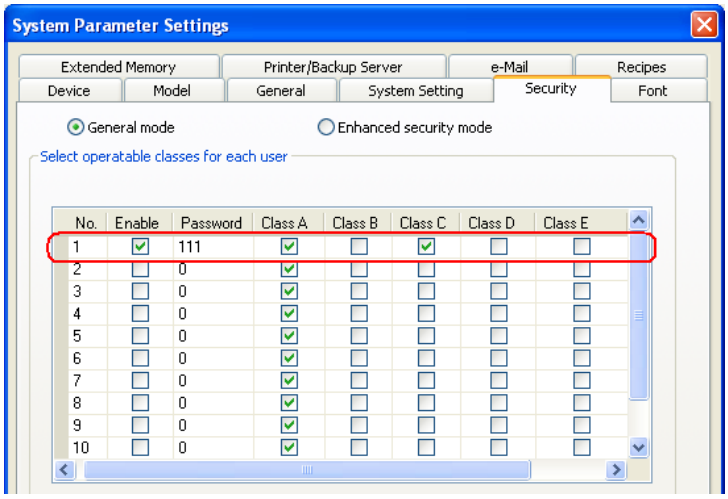
1. General Mode - Общий режим

2. Enhanced Security Mode – Режим повышенной безопасности

10.1.1 General Mode - Общий режим

Пароли должны состоять только из цифр (0-9) и можно задать максимально 12 различных паролей пользователя. Всего существует семь уровней доступа пользователей, включая [none] (отсутствие какой-либо защиты) и уровни с A по F.

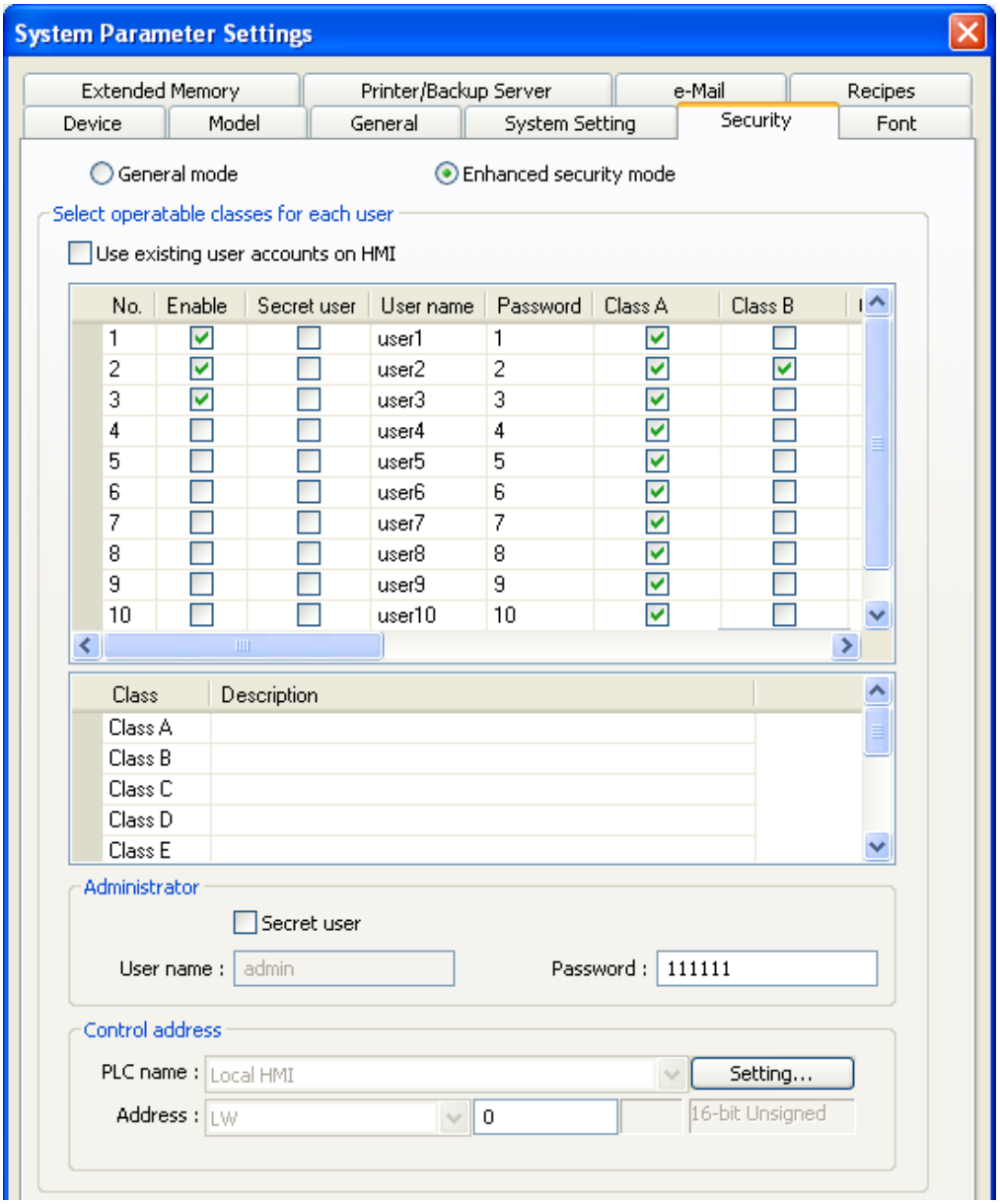
После того, как пароль введен, доступ пользователя к редактированию объектов определяется установками безопасности.
Например, такие установки показаны для пользователя [user 1]. Они означают, что пользователю позволено работать с объектами уровня защиты [none], A, C.



10.1.2 Enhanced Security Mode – Режим повышенной безопасности

В EasyBuilder Pro может быть назначено 11 пользователей, плюс Administrator (default). Пароли пользователя могут быть алфавитно цифровыми и каждый пользователь может оперировать объектами 13-ти классов: от **A** до **L** и **none**.

После того, как пароль введен, доступ пользователя к объектам определяется классами объектов. *Enhanced Security Mode* позволяет использовать управляющие адреса (control address) для манипулирования правами доступа непосредственно с панели.



10.2 Режим повышенной безопасности и управляющий адрес

Управляющий адрес (Control Address) хранится в регистрах «LW» панели. 20 соседних регистров используются для Управления учетными записями пользователей - *User Account Management.*

10.2.1 Использование Control Address

Пример: Когда управляющий адрес задан, как «LW-n».

LW-n (1 word) -> **[Command]**, управление командами, такими как: Login, Logout, Add/Setting/Delete Accounts, и т. д.)

LW-n+1 (1 слово) -> **[Result]**, Отображает результат выполнения команд.

LW-n+2 (1 слово) -> **[Index]**, индекс «accounts» (обычно используется с *Option List Object*).

LW-n+3 (1 слово) -> **[Privilege]**, значения (Level A = bit0, Level B = bit1…и т. д.)

LW-n+4 (8 слов) -> **[Name]**, Имя «account» (буквы, цифры, “-“или “_”, с учетом регистра).

LW-n+12 (8 слов) -> **[Password]**, пароль «account» (буквы, цифры, “-“или “_”, с учетом регистра).

10.2.2 Введение в команды

Ввод разных значений в **[Command]** -> LW-n, вызывает соответствующие функции:

a. Log in using account name (Загрузка имени текущей учетной записи) -> **[value 1]**, с **[Name]** и **[Password]**.

b. Log in using index (Загрузка используемого индекса) -> **[value 2]**, с **[Index]** и **[Password]**.

c. Log out (Завершение работы) -> **[value 3]**.

d. Change the password of current account (Изменить пароль текущей учетной записи)-> **[value 4]**, с **[Name]** и **[Password]**. **[Name]** должен быть спарен с оригинальным паролем, и заполняет новый пароль в **[Password]**.

e. Add an account (Добавить учетную запись) -> **[value 5]**, with **[Name]**, **[Password]** and **[Privilege]**.

f. Add a temporary account (Добавить временную учетную запись) -> **[value 6]**, с **[Name]**, **[Password]**, **[Privilege]**, и **[Index]**. **[Index]** предназначен для указания действительного периода времени в минутах. «0» - обозначает постоянно действительныйпериод.

g. Delete current account (Удалить учетную запись) -> **[value 7]**, с **[Name]**.

h. Delete current account (Удалить учетную запись) -> **[value 8]**, с **[Index]**.

i. Setting the privilege of current account (Настройка привилегий текущей учетной записи) -> **[value 9]**, с **[Name]** и **[Privilege]**.

j. Setting the privilege of current account (Настройка привилегий текущей учетной записи)-> **[value 10]**, with **[Index]** and **[Privilege]**.

k. Setting the password of current account(Настройка привилегий текущей учетной записи) -> **[value 11]**, with **[Name]** and **[Password]**.

l. Setting the password of current account (Настройка пароля текущей учетной записи) -> **[value 12]**, с **[Index]** и **[Password]**.

m. Read the privilege of current account (Читать привилегии текущей учетной записи) -> **[value 13]**, с **[Name],** если прошла, пишите в **[Privilege]**.

n. Read the privilege of current account (Читать привилегии текущей учетной записи) -> **[value 14]**, с **[Index]**, если прошла, пишите в **[Privilege]**.

- **Add a temporary account**: Различие между общей учетной записью и текущей не будут сохранены во Флэш-памяти, следовательно не будут действительны при следующем включении питания.
- **Delete current account:** Текущая работающая учетная запись не может быть удалена.
- **Offline/Online Simulation:** Имитация использования учетных записей. Изменения при имитации не сохранятся для следующей имитации.
- **admin:** Учетная запись администратора по умолчанию. Не может быть удалена. Открывает все классы привилегий и уровни привилегий ее не могут быть изменены.
- **System Register LW-10754:** Отображает текущее имя пользователя

10.2.3 Введение в результирующие выходы

Когда команды выполнены система будет автоматически посылать коды результата на управляющий адрес «LW-n+1». Список кодов результата приведен ниже в шестнадцатиричном формате.

Коды результата: Результирующие сообщения (Result Messages):

(0x001): Command successfully executed – Команда выполнена успешно.

(0x002): Command error – Ошибка команды..

(0x004): Account already exists (при добавлении записи) - Учетная запись уже существует.

(0x008): Account does not exist – Учетная запись не существует.

(0x010): Password error – Ошибка пароля.

(0x020): Current command can’t be executed – Текущая команда не может быть выполнена.

(0x040): Invalid account name – Некорректное имя учетной записи.

(0x080): Invalid account password - Некорректный пароль учетной записи.

(0x100): The imported data is invalid – Импортированные данные некорректные.

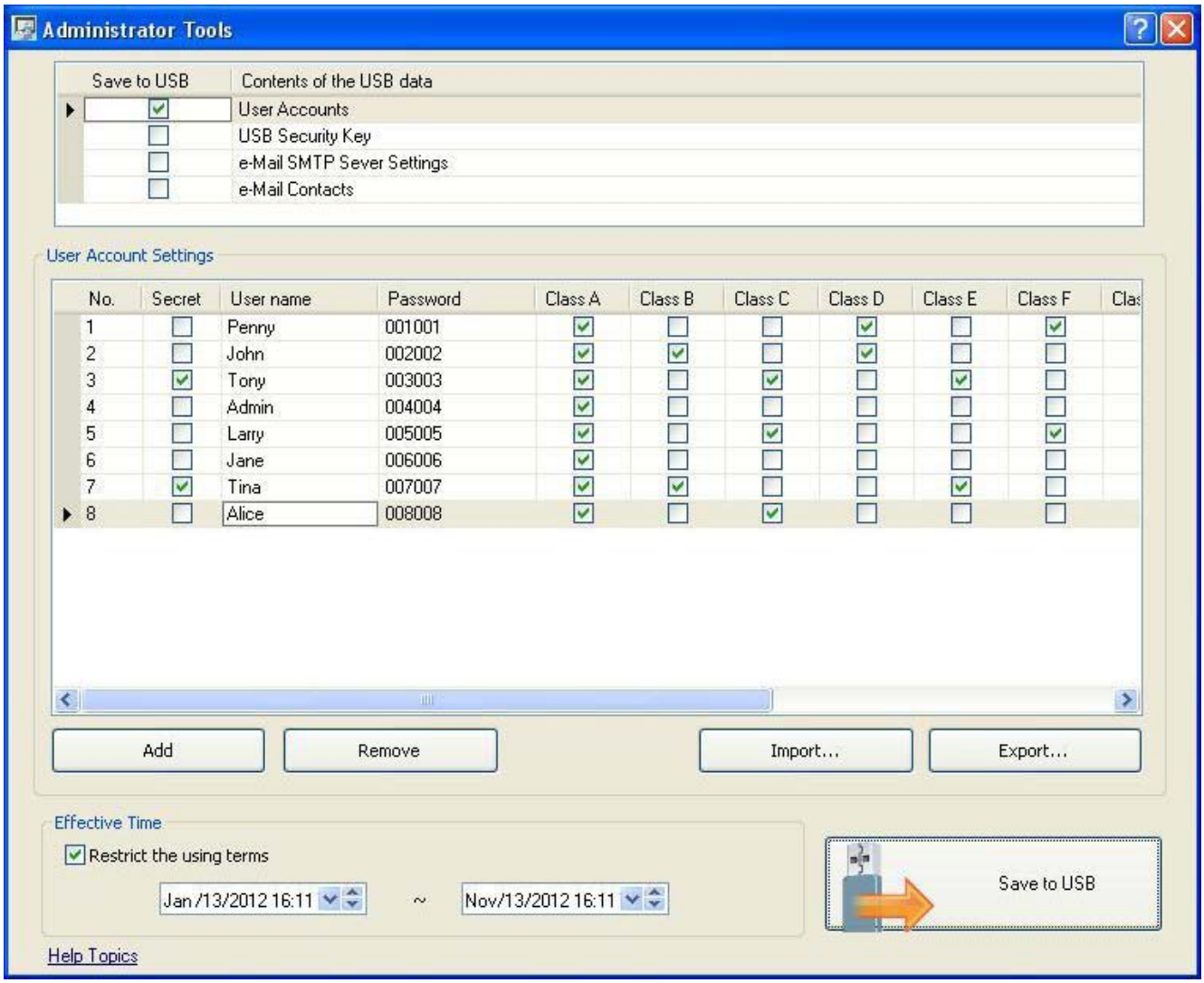
(0x200): Not within the effective time limit – Вне действующего предела времени доступа. (при использовании доступа с USB Security Key)

- Пользователи могут заранее определять результирующие коды *Event Log Object* и, затем, отображать результирующие события на дисплее событий - *Event Display*.

10.3 Режим повышенной безопасности и функциональная кнопка

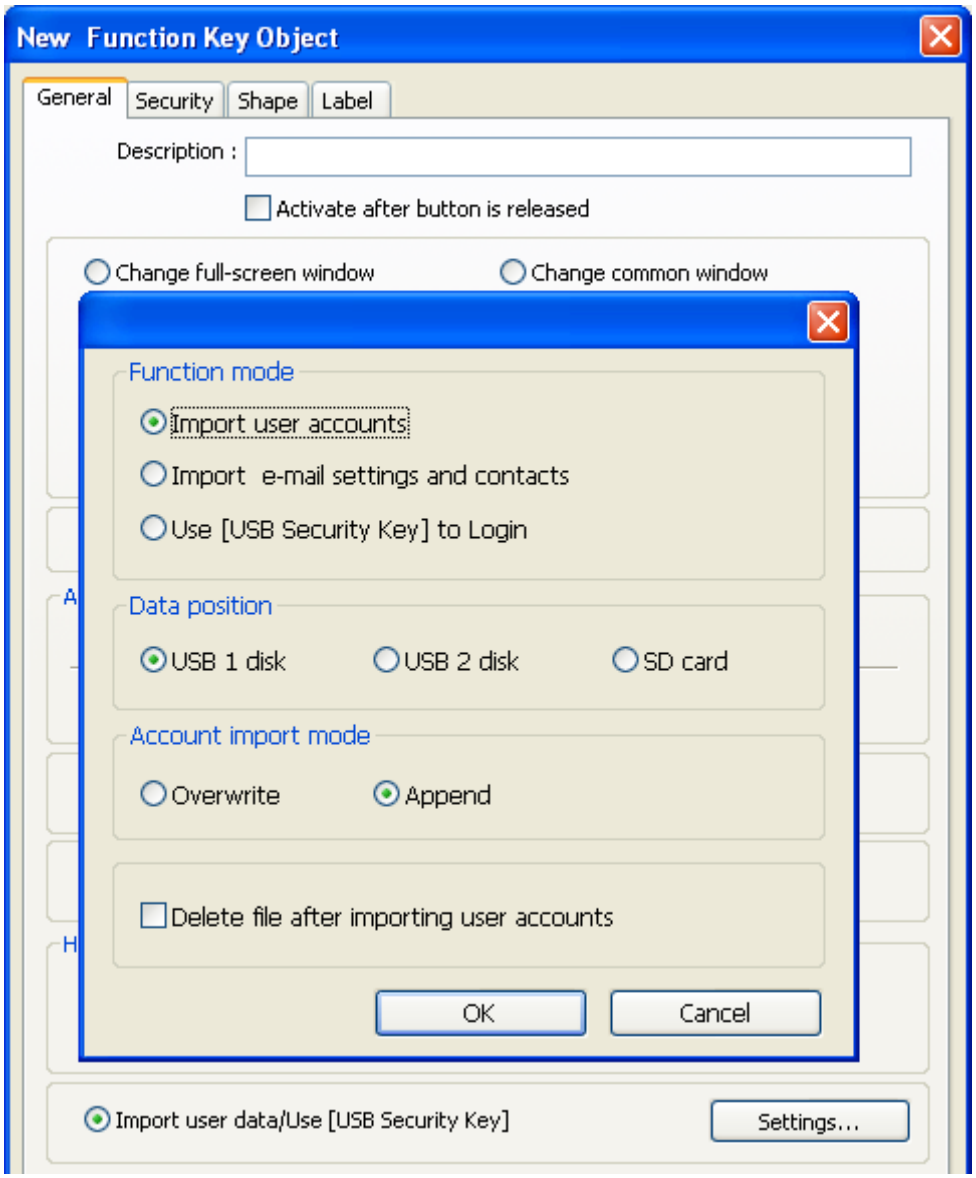
10.3.1 Импорт учетных записей пользователей

Помимо закладки **[System Parameter Settings] / [Security]**, пользователи могут настроить учетные записи пользователей, запустив *Administrator Tools* в установочной директории EasyBuilder Pro и отметив **[User Accounts]**. Максимально 127 учетных записей может быть добавлено, как показано ниже:



Подробности об *Administrator Tools*, приведены в соответствующей главе этого руководства. Добавленные учетные записи могут быть сохранены на USB и SD карте.

Можно создавать функциональную кнопку (Function Key) для импортирования учетных записей, как показано на следующей странице:



По завершении настройки, вставьте внешнее устройство в панель, и используйте *Function Key* для импорта учетных записей.

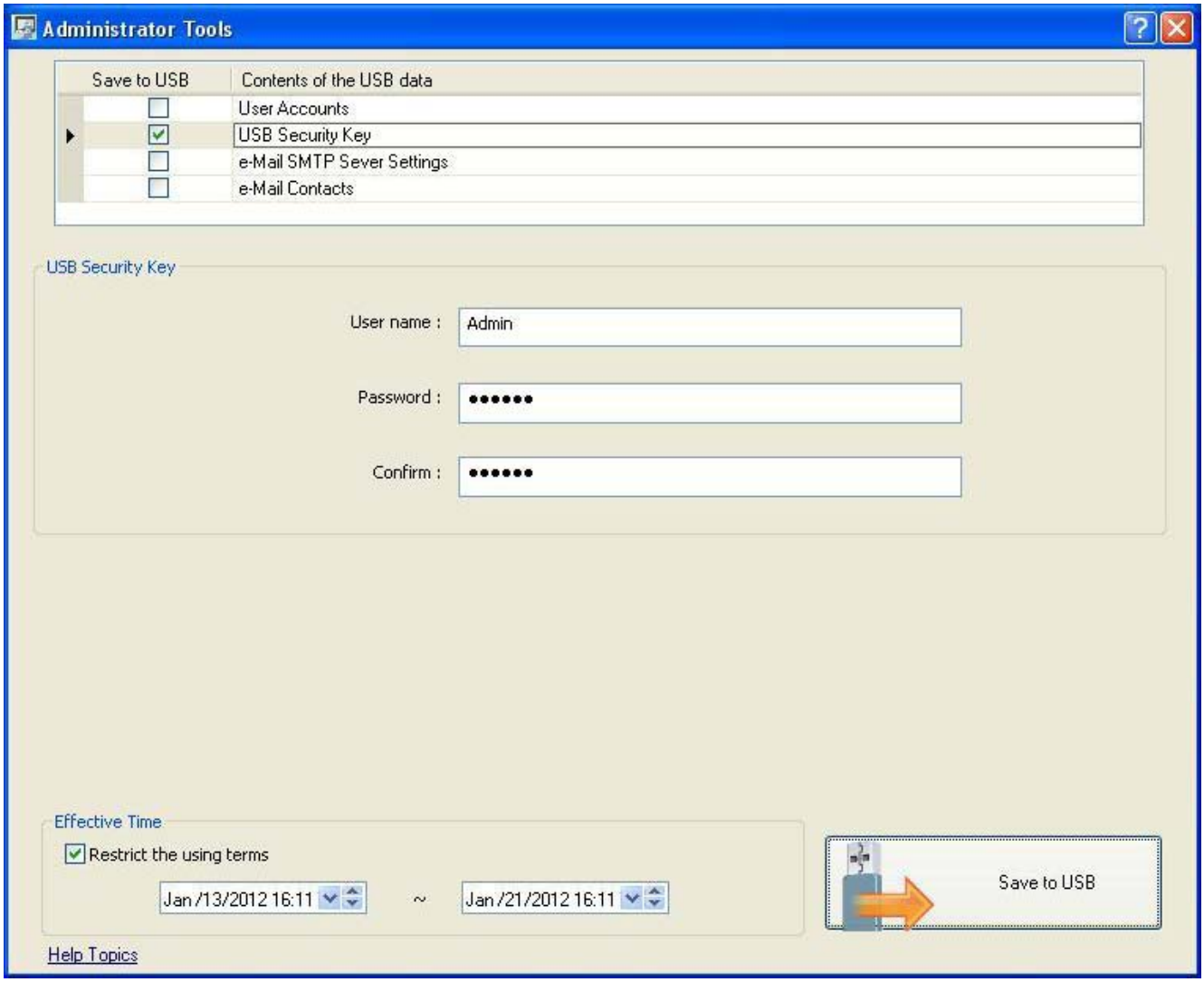
Если выбрано [Overwrite], существующие учетные записи в системе будут удалены перед импортированием и система завершит сеанс после импортирования.

Если отмечено [Delete file after importing user accounts] –Удалить файл после импортирования учетных записей, существующие учетные записи в системе будут удалены и сохранены на внешнем устройстве после импортирования.

Действующий предел времени для импортирования данных может быть настроен только при помощи *Administrator Tools.*.

10.3.2 Использование USB Security Key

Вместо ввода учетной записи и пароля вручную для получения доступа, можно, также, использовать клавишу (key). В директории установки EasyBuilder Pro, запустите *Administrator Tools*, отметьте **[USB Security Key] – USB ключ безопасности** , настройте соответствующую информацию для доступа, *USB Security Key* может быть использован для прямого доступа, как показано ниже:

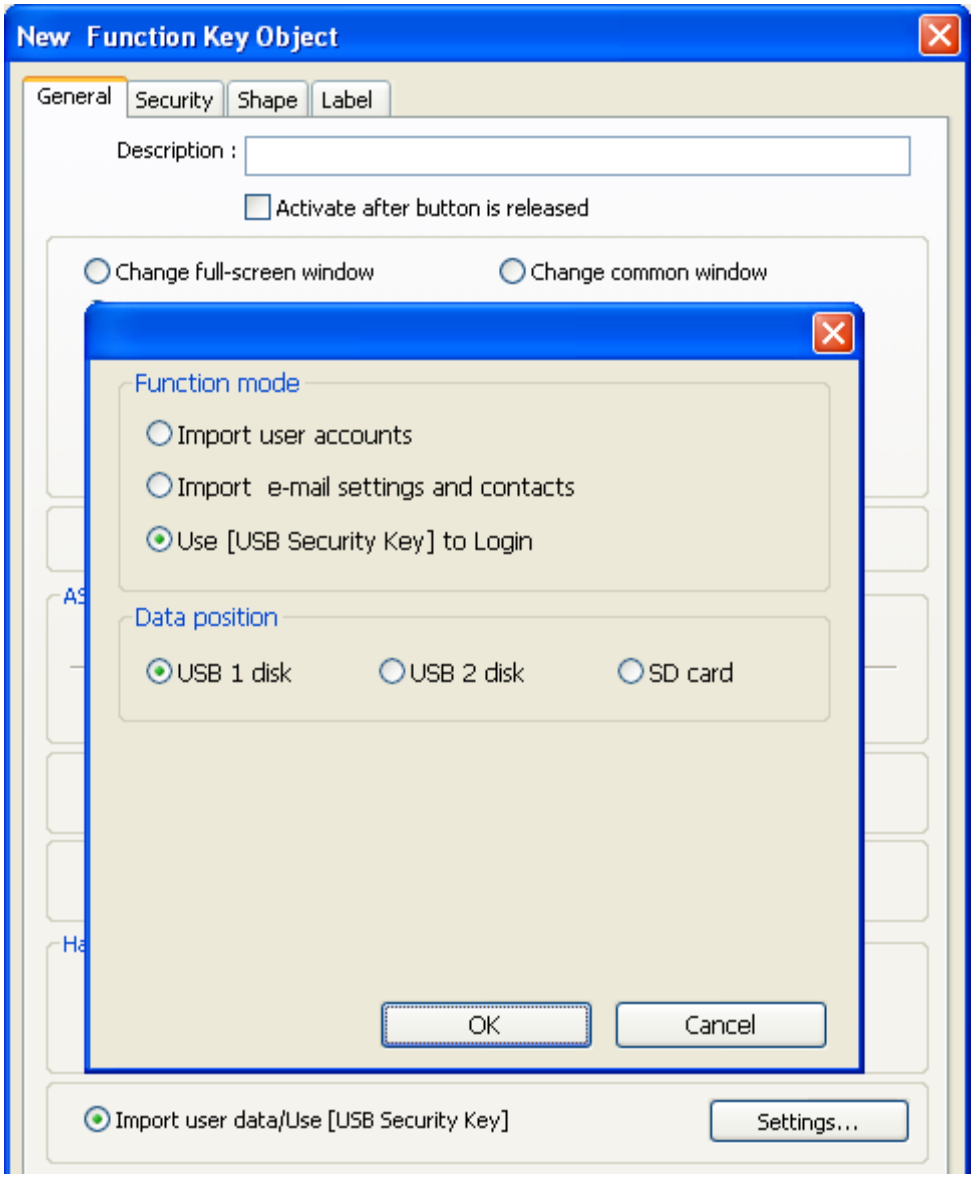


Заметьте, что учетная запись пользователя должна уже существовать в панели до настройки *USB Security Key*.

Подробности об *Administrator Tools*, приведены в соответствующей главе этого руководства.

Настройки *USB Security Key* могут быть сохранены на USB и SD карте.

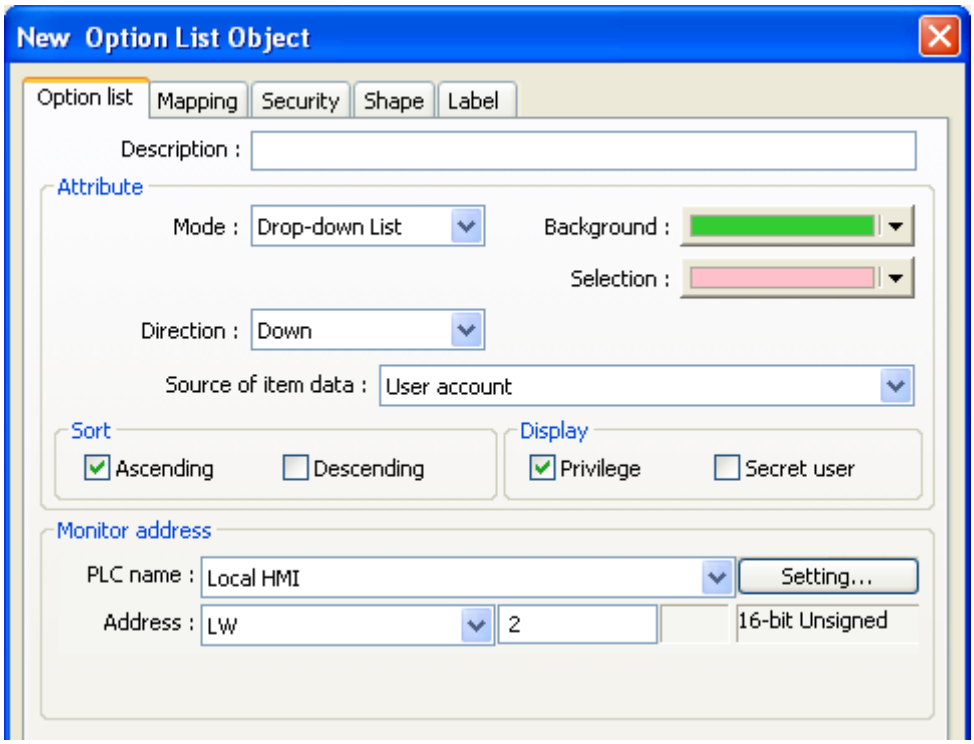
Можно создавать функциональную кнопку (Function Key) для использования доступа с применением *USB Security Key,* как показано на следующей странице:



По завершении настройки, вставьте внешнее устройство в панель, и используйте *Function Key,* чтобы иметь доступ с применением *USB Security Key*. Действующий предел времени для доступа (login) с ключом может быть настроен только при помощи *Administrator Tools.* Система будет отключать возможность доступа автоматически (logout), по прошествии установленного предела времени.

10.4 Режим повышенной безопасности и список Option List Object

Режим повышенной безопасности (Enhanced Security Mode) использует управляющий адрес «LW-n=2» в качестве индекса учетных записей (account index). В списке вариантов свойств объектов (Option List Object), имя записи (account name) и привилегии можно просмотреть. Если управляющий адрес назначен «LW-0», то адрес просмотра будет «LW-2». Пользователь может выбрать: показывать или нет список с привилегиями и секретных пользователей в *Option List*. Секретный пользователь (Secret Users) означает, что в **[System Parameter Settings] / [Security] / [Enhanced Security Mode]**, это имя учетной записи этого пользователя назначено спрятанным (hidden). Другие пользователи не будут в состоянии увидеть относящуюся к нему информацию в *Option List Object*.

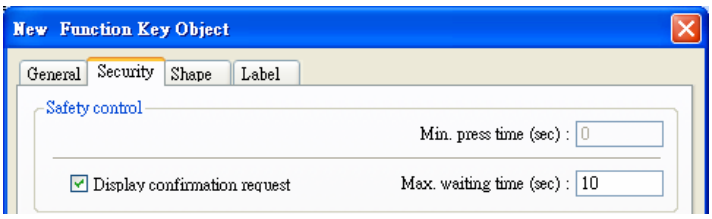


Желаете узнать больше об ***Enhanced Security Mode***?

Соединитесь с Internet и загрузите демо-проект:

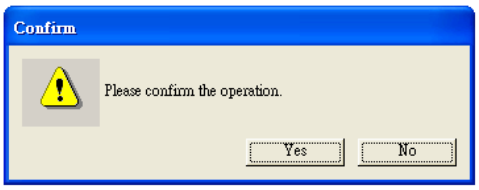
ftp://ftp.weintek.com/eMT3000/Project/2011SPSdemo/Enhanced_Security.zip

10.5 Настройки безопасности объектов



[Safety Control] используется для предотвращения ошибочных действий.

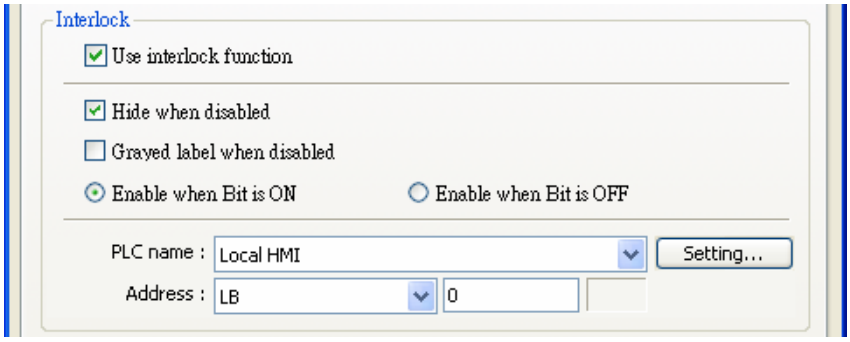
[Min. press time (sec)]
Только при длительном (не меньшем заданного значения) удержании объект может стать доступным.



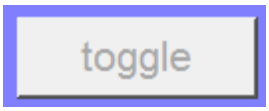
[Display confirmation request]
После изменения объекта появляется окно подтверждения. Если в течение времени, указанного в поле **[Max. waiting time (sec)]** подтверждения не будет, окно закроется с отменой действий.

[Interlock] Когда отмечено, доступность объекта определяется состоянием указанного бита.

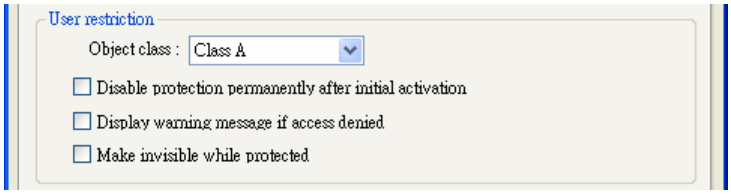
Как показано справа, объект доступен для работы с ним только при состоянии бита «LB-0» ON.



[Hide when disabled]
При выполнении условий блокировки скрывать объект.
[Grayed label when disabled] Когда указанный бит в состоянии OFF, название объекта становится серым



[User restriction]
Только при совпадении разрешенного класса пользователя и класса объекта, пользователь может работать с ним.



[Object class]
[None] означает, что любой пользователь может работать с объектом.
С объектами класса “Administrator” может работать только учетная запись “admin”.

[Disable protection permanently after initial activation]
После первого совпадения класса пользователя и объекта система остановит проверку прав доступа. Даже разные пользователи могут работать свободно.

[Display warning message if access denied]
Когда класс пользователя и объекта не совпадает, появится окно предупреждения (типа 7).
Пользователи могут изменять его содержимое.

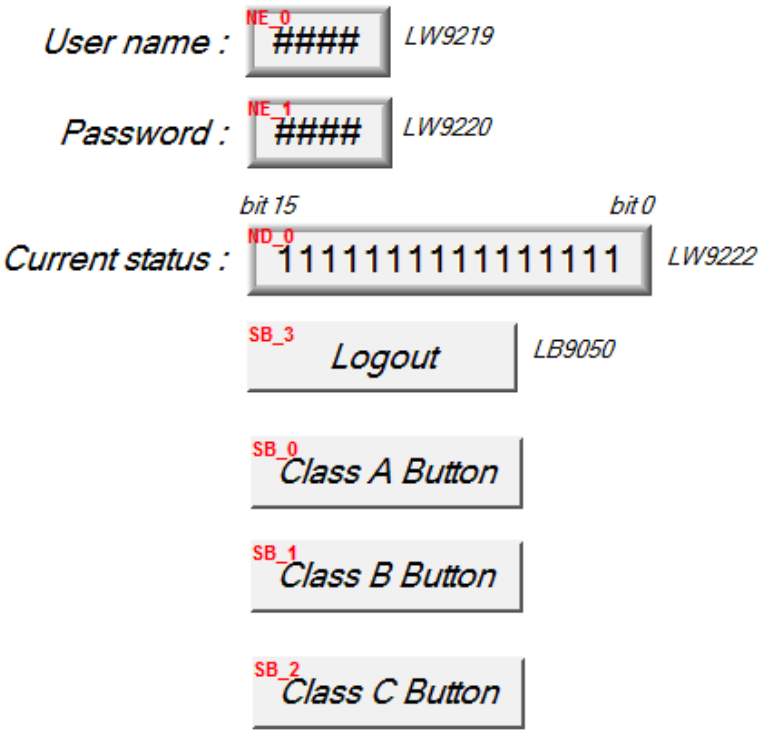
[Make invisible while protected]
Когда класс пользователя и объекта не совпадает, объект будет скрыт.

10.6 Пример настроек защиты

1. Создайте новый проект, перейдите на **[System Parameter Settings] / [Security]** и добавьте трех пользователей:

Пользователь [User 1] - может работать с объектами класса A
Пользователь [User 2] - может работать с объектами классов A и B,
Пользователь [User 3] - может работать с объектами классов классов A, B, C.

2: В рабочем окне [Window_10] EB8000 создайте объекты, показанные ниже:



Цифровой ввод **-[Numeric Input] [LW-9219].** Номер пользователя(1~12) Длина = 1 слово, (16-bit Unsigned)

[LW-9220]. Пароль. Длина = 2 слова, (32-bit Unsigned)

Цифровой дисплей - **[Numeric Display] [LW-9222].** Доступные классы текущего пользователя, (16-bit Binary)

Установить бит **- [Set Bit] [LB-9050].** Выход пользователя.

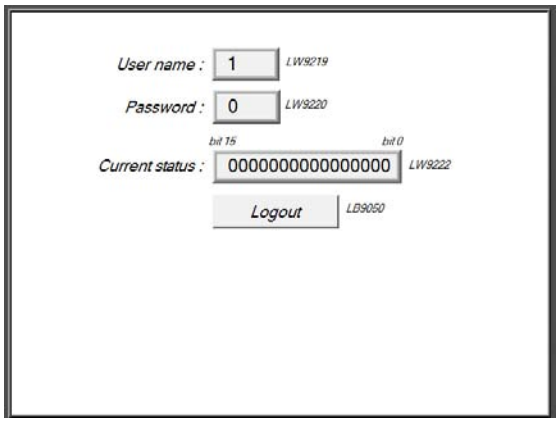
Три объекта **[Set Bit],** каждый из которых связан со своим классом, но все они определяют выбор **[Made invisible while protected] –** Сделать объект невидимым, когда он защищен.

После создания и настройки объектов, сохраните и откомпилируйте проект. Затем. Перейдите в режим имитации работы без соединения с панелью - «off-line simulation».

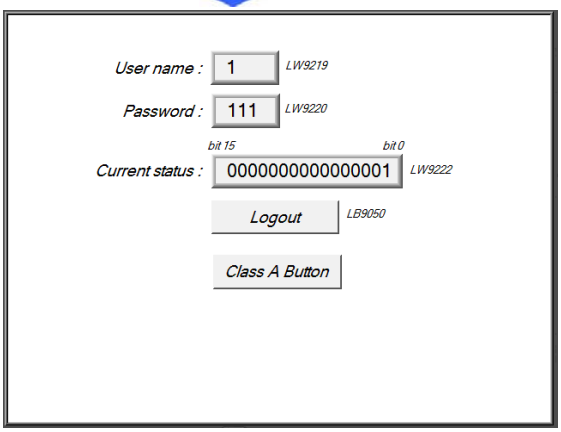
На следующей странице будет показано, как работает проект с этими настройками.

3. На рисунок показан экран имитации, когда пароль еще не введен.

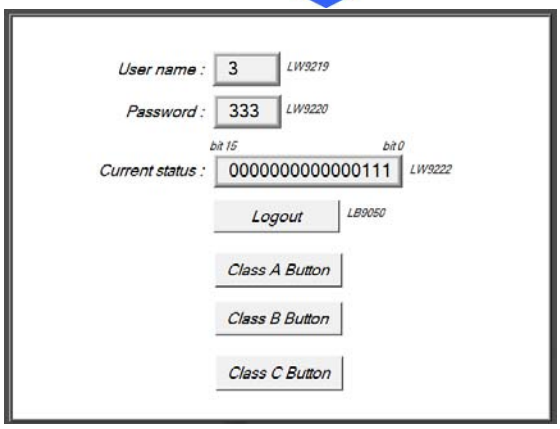
По адресу [LW9222] во всех разрядах находятся 0, это означает, что й пользователь может работать с объектами с уровнем доступа [none] (отсутствие защиты). Объекты [SB_0]~[SB_2] имеют классы защиты A~C, поэтому скрыты в соответствии **[Make invisible while protected].**



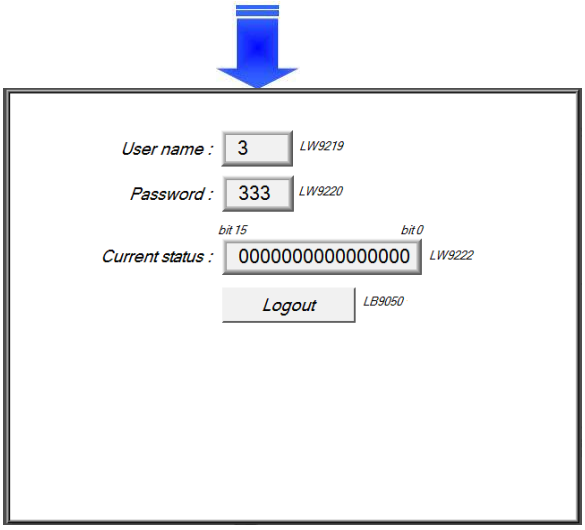
4. После ввода пароля “111” экран приобретает следующий вид:
Пользователь [user 1] может работать только с объектом класса A. Поэтому объект [SB_0] появляется и доступен пользователю.
Бит в нулевом разряде [LW9222] обращается в 1, что означает доступнй пользователю класс - A.



5. При вводе пароля “333” пользователем [user 3] экран становится следующим:
Пользователю [user 3] разрешено работать с объектами классов A, B, C.
Биты «0, 1, 2» в [LW9222] обращаются в 1, подтверждая, что пользователь может работать с объектами классов A, B, C.



6. При нажатии [LB9050] происходит выход текущего пользователя, система возвращается в исходное состояние. Другими словами, текущий пользователь может оперировать только с объектами класса [none].



✹ **Password input – Ввод пароля.**

Если пароль не корректный, бит [LB-9060] будет установлен в состояние ON; Если пароль корректный, бит [LB-9060] вернется в состояние OFF.

✹ **Change password when HMI is in operation – Изменение пароля при работе панели.**

При переходе бита [LB-9061] из состояния OFF в состояние ON, данные в словах [LW-9500] ~ [LW-9522] могут быть использованы для обновления пароля и использования нового пароля в будущем. Доступные пользователю классы объекты не будут изменены при смене пароля.

Глава 11. Индексный регистр

11.1 Введение

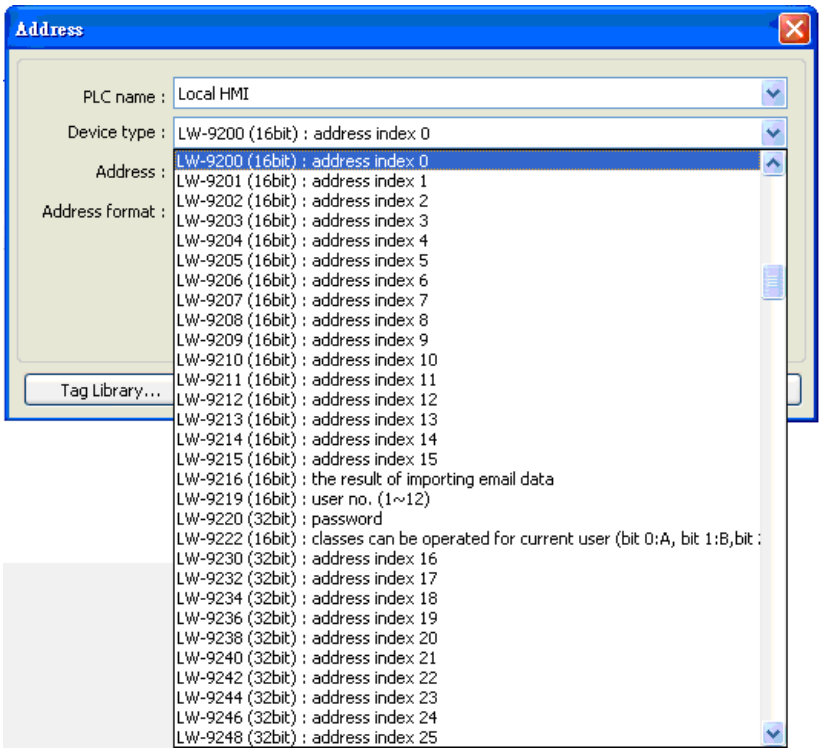
В EasyBuilder Pro предусмотрены 32 индексных регистра (Index Registers) для более гибкого использования адресов. С помощью данных регистров пользователь может обновлять адреса чтения/записи объектов без изменения содержимого объекта во время работы устройства.

Адреса 32 индексных регистров следующие:

16- ти битовые регистры:
INDEX 0 [LW9200] ~
……
……

INDEX 15 [LW9215]
Максимальный диапазон - 65536 слов

32-х битовые регистры:
INDEX 16 [LW9230] ~
……
……
INDEX 31 [LW9260]
Максимальный диапазон - 4294967296 слов



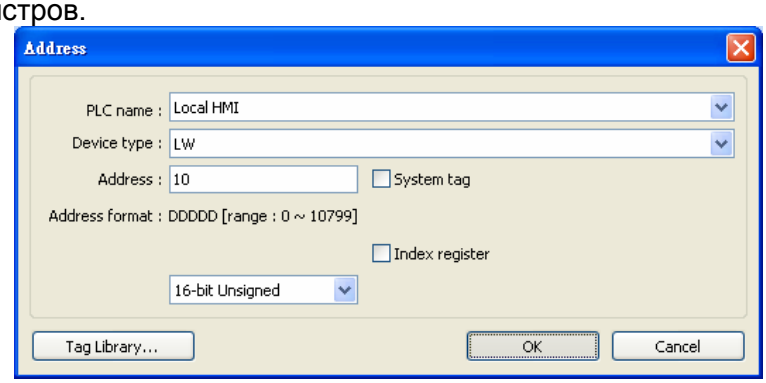
INDEX 0 ~ INDEX 31: Описание индексных регистров (Descriptions)
[LW9260] ~ [LW9260]: Адреса слов индексных регистров.
При использовании **[Index register]**, адрес **[Device type]** будет определяться значением “Константа настройки адреса + значение выбранного индексного регистра”.

✸ Индексный регистр может быть использован во всех **[Device lists]** встроенных в **[System Parameter Settings],** независимо от формата адреса (бит или слово).

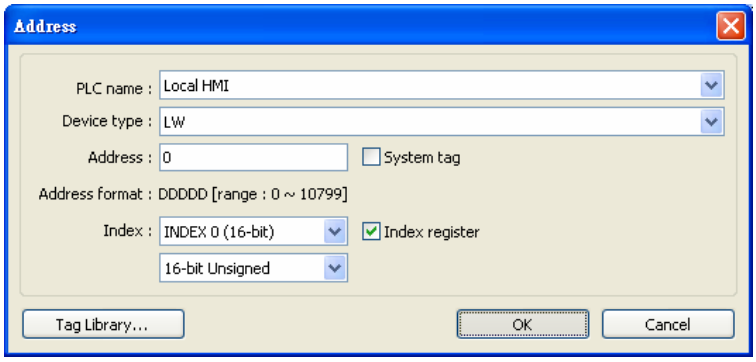
11.2 Пример индексного регистра

Примеры использования индексных регистров.

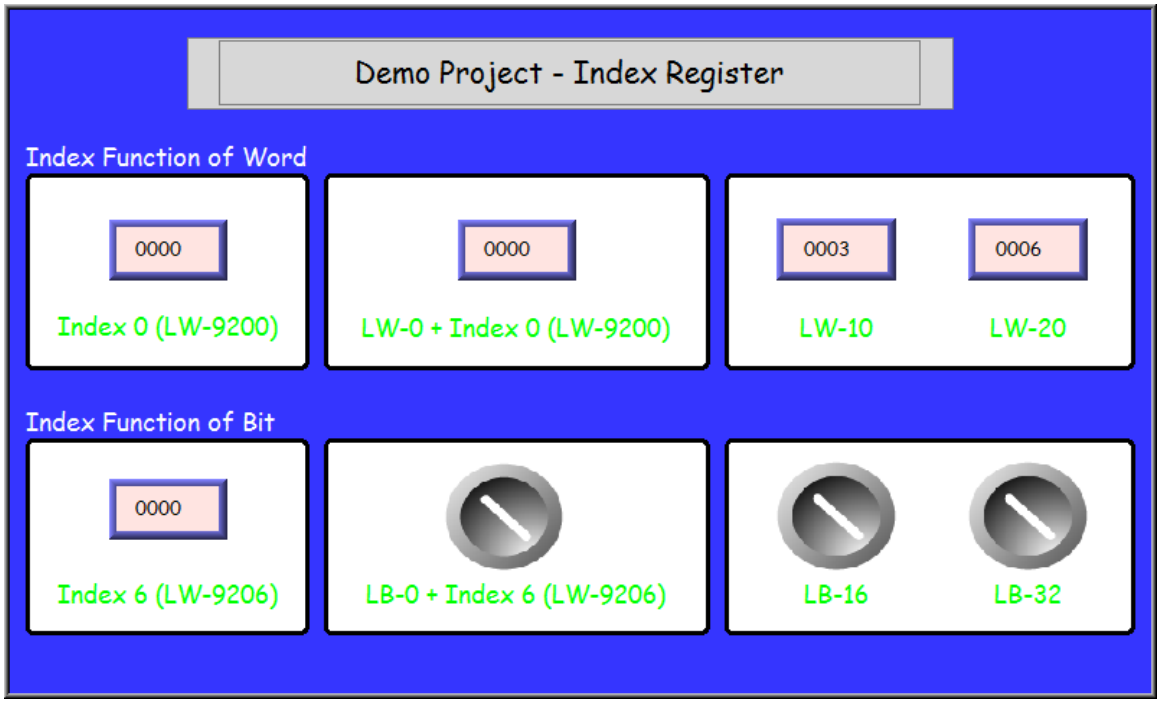
Когда **[Index register]** не отмечен, [Read address] – адрес чтения [LW100] и не будет изменяться при работе проекта.



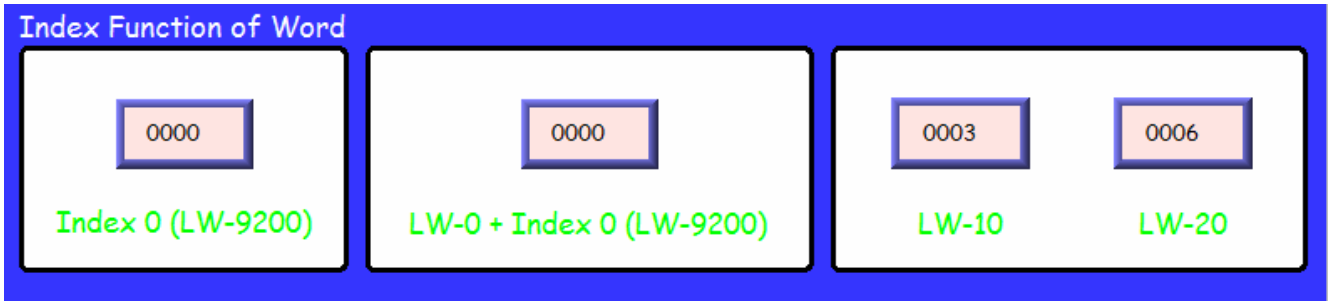
[Index register] отмечен и выбран индексный регистр **[INDEX 0]**:
Адрес чтения настроен на [LW-0 + INDEX 0]
INDEX 0: Index Register 0 или данные адреса [LW-9200].
Если данные адреса [LW-9200] равны “5”, адрес чтения настроен на [LW(0+5)] = [LW-5].



На рисунке ниже показан, в качестве примера, демо-проект:



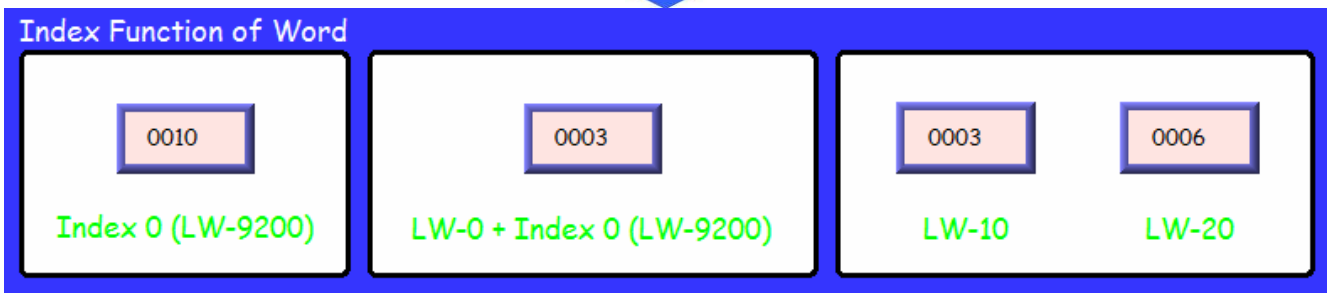
Пример 1: Объект «Index Function of Word» - Функция индексирования слов



Index 0 - “0” = данные адреса [LW-9200] = “0”

Чтение [LW-0 + Index 0] = Чтению содержимого [LW-0]

[LW-10] - ”3” , [LW-20] - ”6”



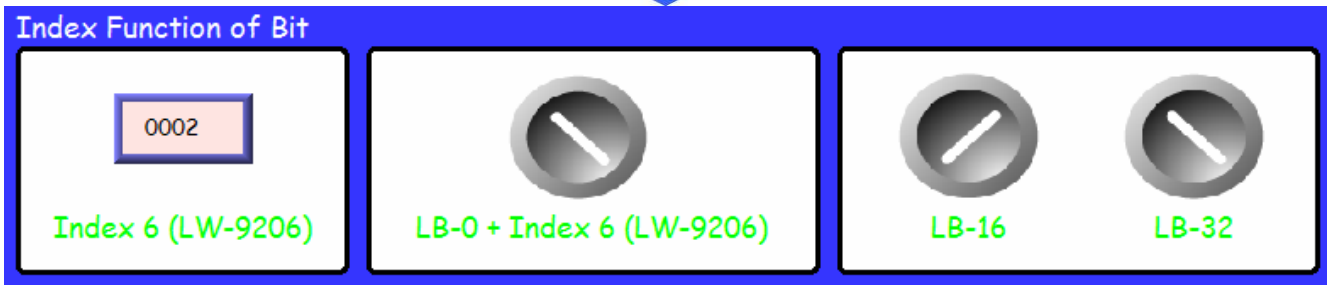
Index 0 [LW-9200] - “10”

Чтение [LW-0 + Index 0] = Чтению [LW-10] = ”3”

Пример 2: Объект «Index Function of Bit» - Функция индексирования битов.
Индексный регистр сдвига может работать с битовыми адресами.
Одно слово (1 word) = 16 бит, изменение значения индексного регистра на одно слово равносильно смещению на 16 бит.

Index 6 [LW-9206] - “1”

Переключатель [LB-0 + Index 6] читает адрес LB-16 = ON установленный в “1”



Index 6 - “2”

Переключатель [LB-0 + Index 6] читает адрес LB-32 = OFF.

✹ **Заключение.**
Индексные регистры используют для того, чтобы изменять адреса. Изменяя данные в индексном регистре, мы можем настроить объект на чтение/запись разичных адресов без изменения собственных адресов устройства. Следовательно мы можем передавать или обмениваться данными по различным адресам.

Соединитесь с Internet и загрузите демо-проект:

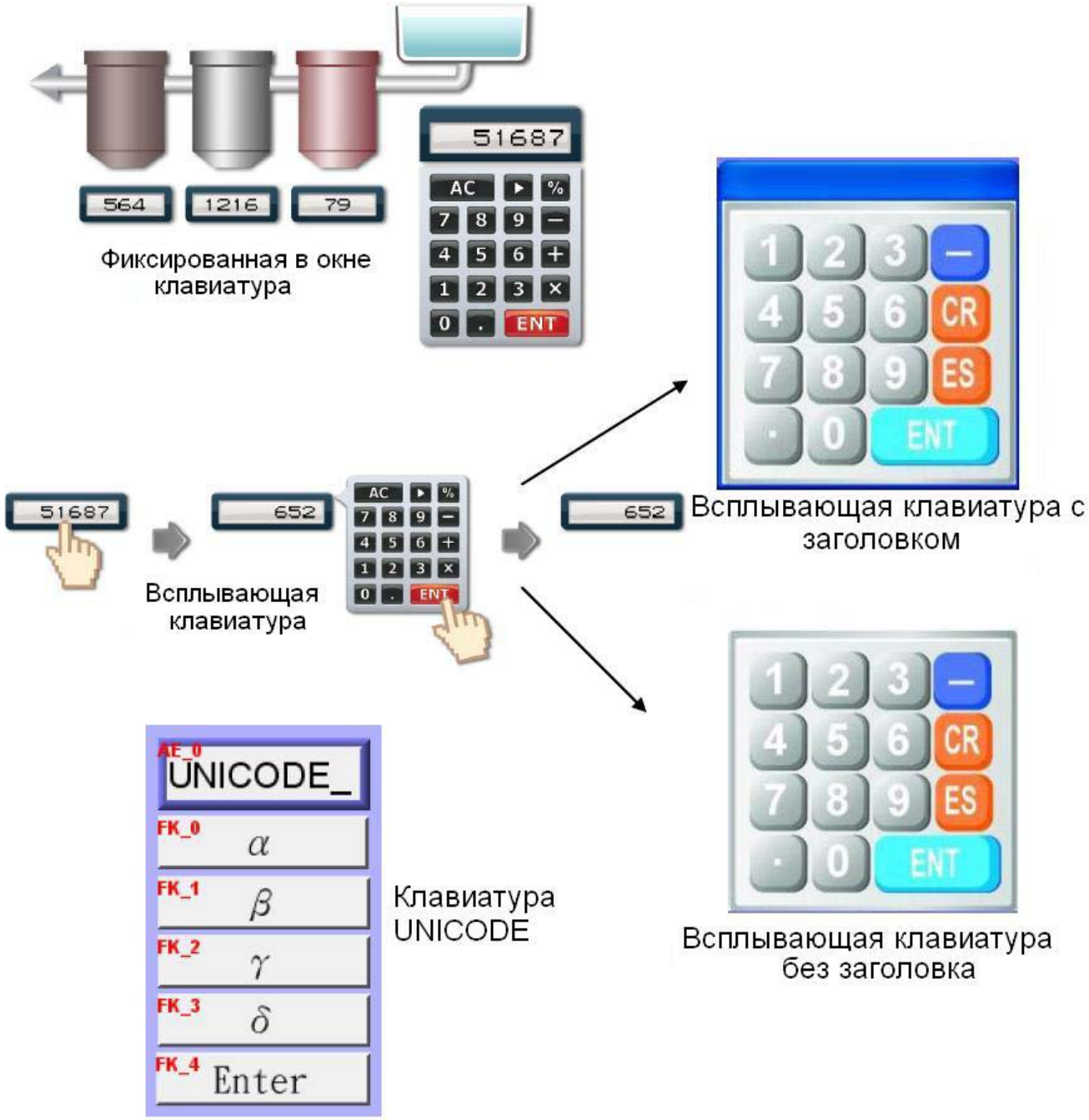
ftp://ftp.weintek.com/MT8000/Project/System_Sample/Index_Function.zip

Глава 12. Проектирование и использование клавиатуры

Для объектов **[Numeric Input]** и **[ASCII Input]** необходима клавиатура.

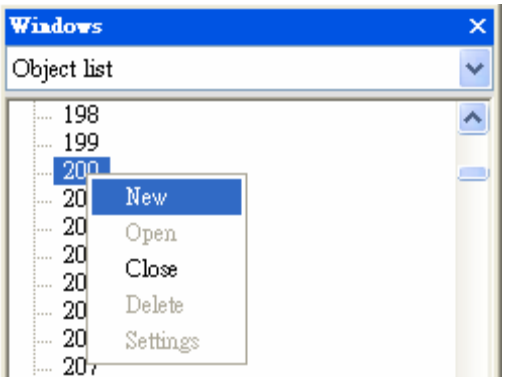
Цифровая и символьная клавиатуры создаются с объектом «Функциональная клавиша» .

Типы клавиатур следующие:



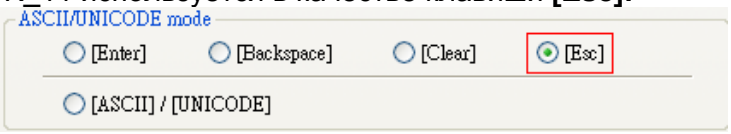
12.1 Этапы разработки всплывающей клавиатуры

Шаг 1
Создайте и откройте окно, в которое будет добавлена клавиатур.
Например, назначьте «*Window 200*» в качестве окна клавиатуры.

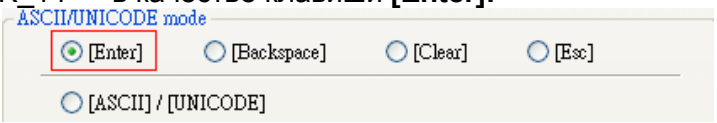


Шаг 2
Настройте высоту и ширину окна «*WINDOW 200*» и создайте несколько объектов [Function Key] в режиме [ASCII/UNICODE mode].

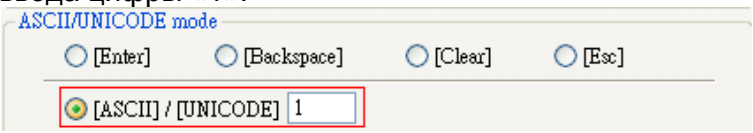
Например:
FK_11 используется в качестве клавиши **[Esc]:**

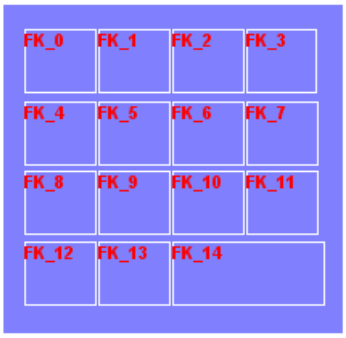


FK_14 — в качестве клавиши **[Enter]:**



Большая часть оставшихся клавиш используется для цифр или текста. Например, FK_0 используется для ввода цифры «1»:



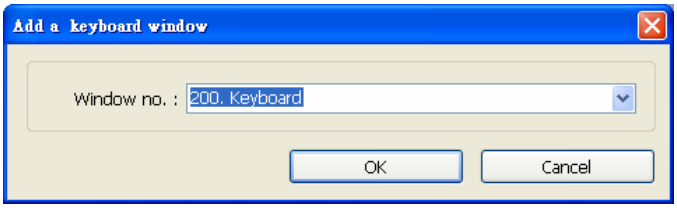


Шаг 3

Выберите соответствующее изображение для каждой кнопки “Function Key”.

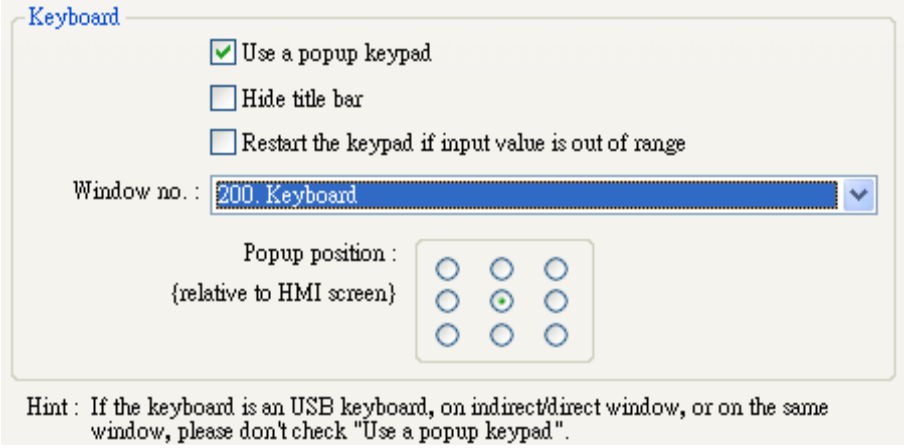
[GP_0] это объект изображения, размещенный на низшем слое в качестве фона.

Шаг 4 Выберите **[System Parameter Settings]** / **[General]** / **[Keyboard]** / **[Add] [Window 200]**. До 32 окон с клавиатурами может быть добавлено.



Шаг 5

После добавления окна клавиатуры, при создании объектов [Numeric Input] или [ASCII Input] окно “200.Keyboard” можно выбрать в **[Data Entry] / [Keyboard] / [Window no.]**.. **[Popup Position]** используется для указания места положения клавиатуры на экране. Система делит экран на 9 позиций.

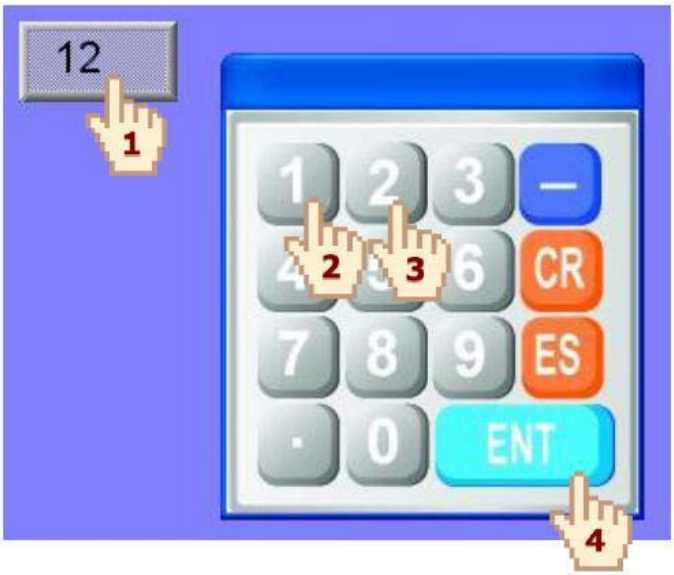


Заметка: если клавиатура – это USB-клавиатура в «indirect/direct» окне или в самом окне, нельзя отмечать «Use a popup keyboard»

Шаг 6

Выберите “200.Keyboard”. При нажатии на объект [Numeric Input] или [ASCII Input] на экране панели будет появляться “WINDOW 200”.

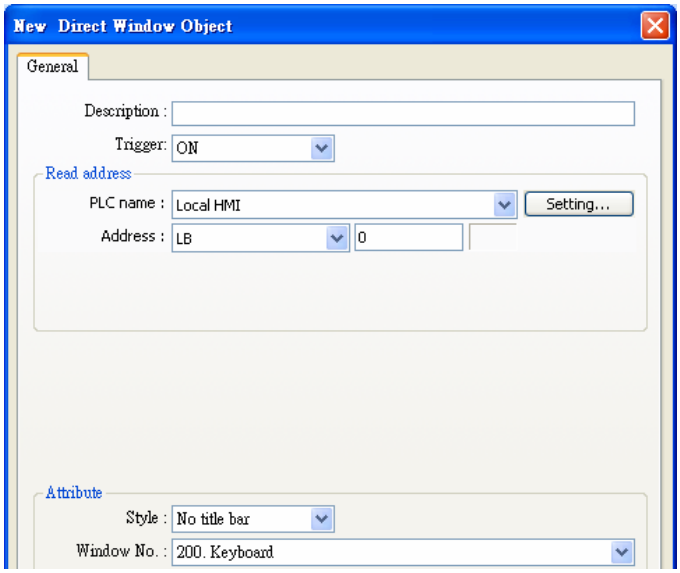
Пользователь может нажимать кнопки для ввода данных.

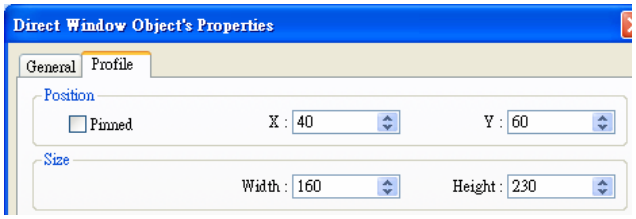


12.2 Использование клавиатуры с непосредственным окном

Шаг 1

Создайте окно типа “[Direct Window” – Прямое или непосредственное окно. Установите адрес активации этого окна. На закладке **[General] / [Attribute]** выберите вариант **[No title bar]** – Без заголовка и укажите **[Window no.].**



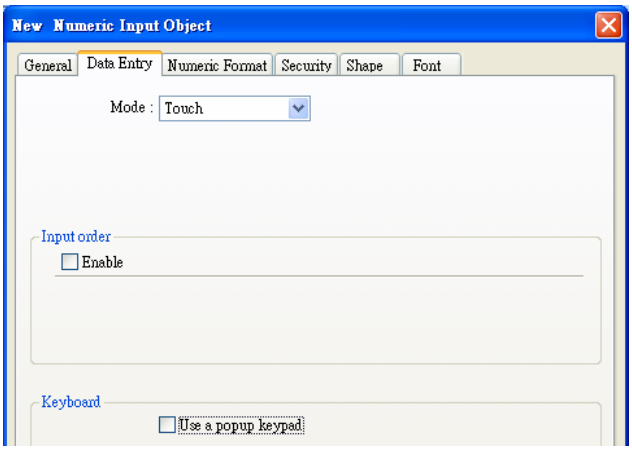


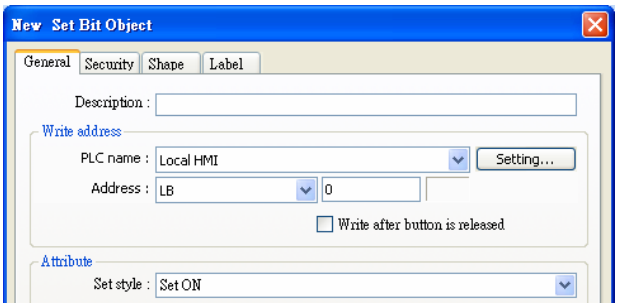
Шаг 2

Установите размеры окна WINDOW 200 такими же, как у клавиатуры.

Шаг 3

Создайте объект типа [Numeric Input], и **<u>не отмечайте</u> [Use a popup keyboard].**





Шаг 4

Добавьте объект “Set Bit”, установите бит [LB0] в состояние ON и поместите его поверх объекта “Numeric Input”.

Шаг 5

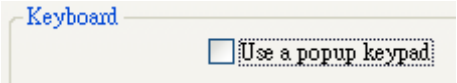
Добавьте объекты “Set Bit” на функциональные кнопки **[Enter]** и **[ESC]** соответственно. [LB0], [Set OFF]. В этом случае, когда пользователи нажимают кнопку [Enter] или [ESC]**,** клавиатура будет закрываться.

12.3 Использование закрепленной на экране клавиатуры

Можно также поместить закрепленную клавиатуру в то же окно, в котором находятся объекты ввода данных, вместо всплывающей.
В этом случае клавиатуру нельзя перемещать или отключать.

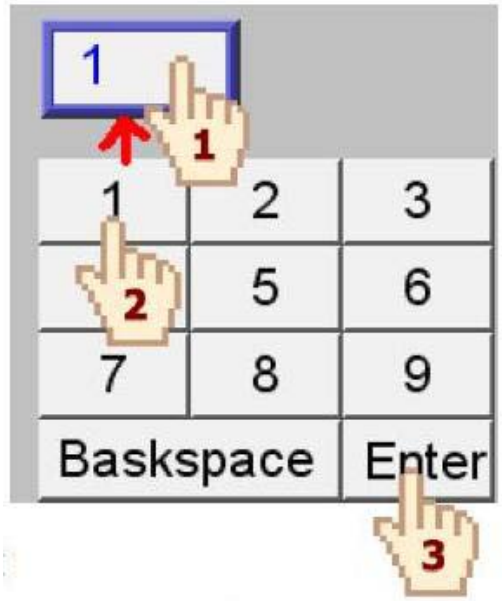
Шаг 1

Создайте объект [Numeric Input] в **[Data Entry] / [Keyboard]** **<u>и не отмечайте</u>** **[Use a popup keypad]**.



Шаг 2

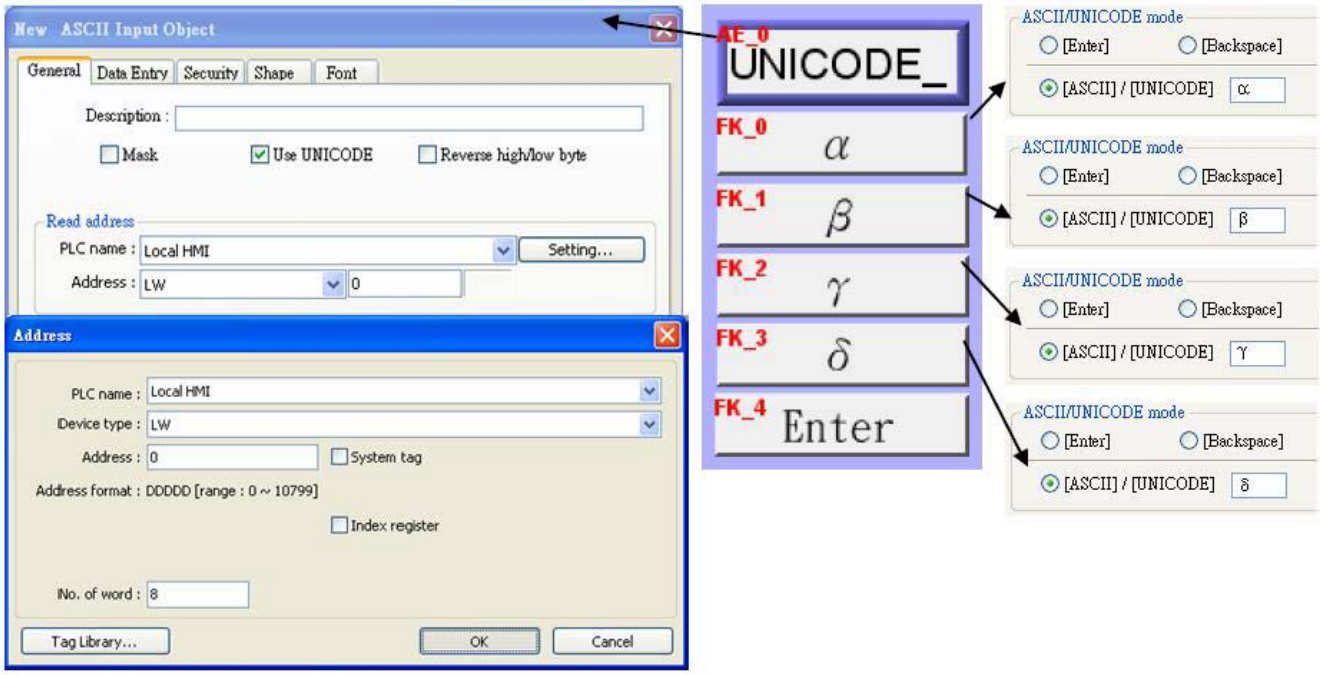
Создайте кнопки клавиатуры [Function Key] и разместите их на экране.



Шаг 3

Нажмите на объект “Numeric Input” .
Пользователь может вводить значения непосредственно с функциональной клавиатуры.

12.4 Использование клавиатуры UNICODE



Разместите объект ”ASCII Input” на экранеи отметьте **[Use UNICODE]**.

Создайте функциональные кнопки ”Function Keys” **[α] [β] [γ] [δ],** как показано выше и клавишу **[Enter].** Простая клавиатура UNICODE готова.

Пользователь может создать группу “Group“ созданной клавиатуры и сохранить в библиотеке - “Save to Group Library” для будущего использования.

Глава 13. Объекты

Данная глава посвящена использования и задания свойств всех типов объектов.
Общие для всех объектов настройки, включая индексный регистр адреса, текстовую надпись, форму и т.п., описаны в главе 9 «Общие свойства объектов».

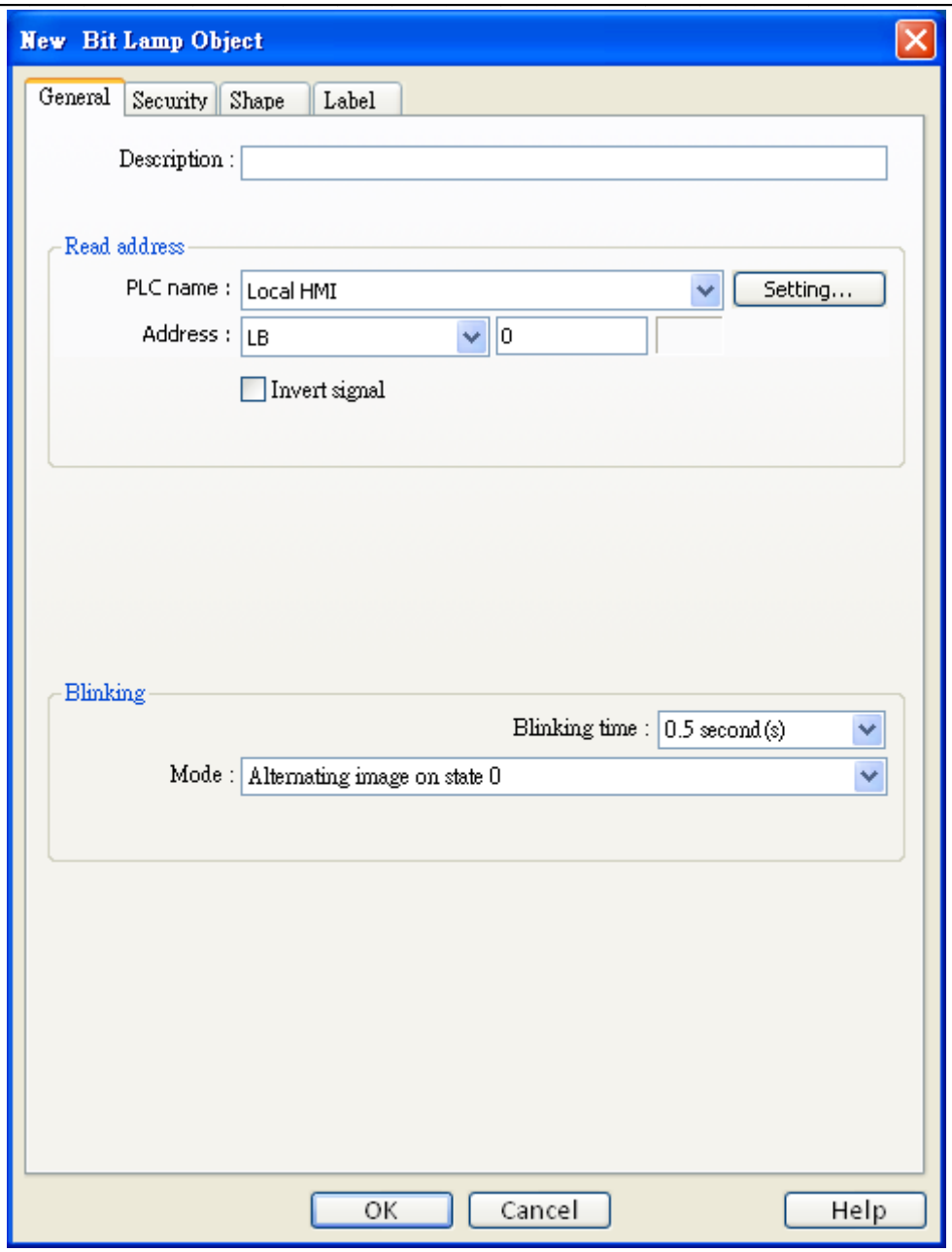
13.1 Bit Lamp - битовая лампа

Краткое описание

Объект [Bit Lamp] показывает состояние битового адреса: 1 или 0 (ON или OFF). Если состояние OFF, то будет выводиться состояние - «State 0». Если ON — будет показано состояние - «State 1»:

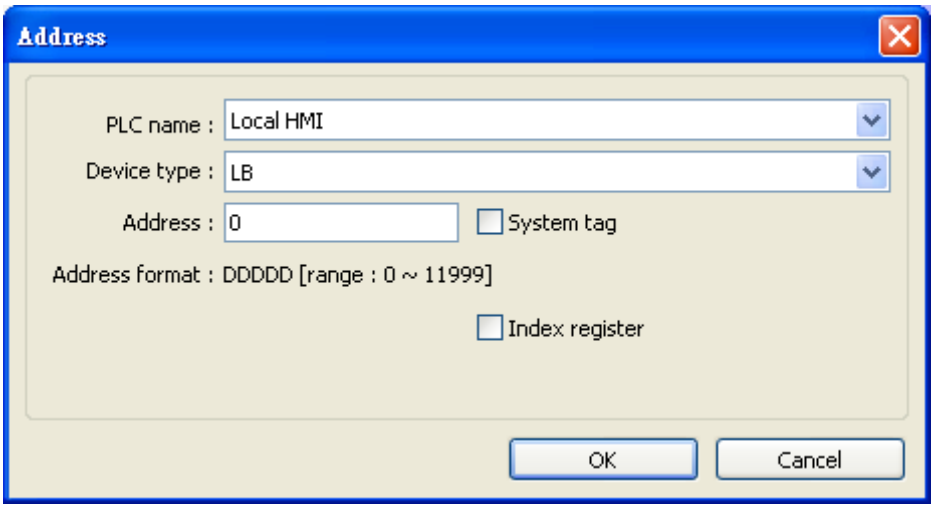
Настройка - Configuration

Щелкните пиктограмму **[Bit Lamp]** напанели инструментов — откроется окно **[Bit Lamp Object's Properties]**; после задания всех полей в этом окне нажмите **[OK]** — будет создан новый объект [bit lamp]. далее



Description Смысловое название (reference name) объекта. Система не использует его, оно нужно только для документации пользователя.

Read address Нажмите **[Setting…] ,** чтобы выбрать бит, управляющий объектом [bit lamp] - **[PLC name], [Address], [Device type], [System tag], [Index register]**. Пользователь может назначить адрес также в закладке **[General]** при добавлении нового объекта.



[Invert signal] Показывает изображение в состоянии, противоположном значению бита; например, если истинное состояние “OFF”, то объект [bit lamp] будет показан в состояни “ON”.

Blinking Устанавливает параметры мерцания объекта

[Blinking mode]

None Отсутствие эффекта мерцания.

Alternating image on state 0 Попеременное отображение «состояния 0/ состояния 1» при значении бита OFF.

Alternating image on state 1 Попеременное отображение «состояния 0 / состояния 1» при значении бита ON.

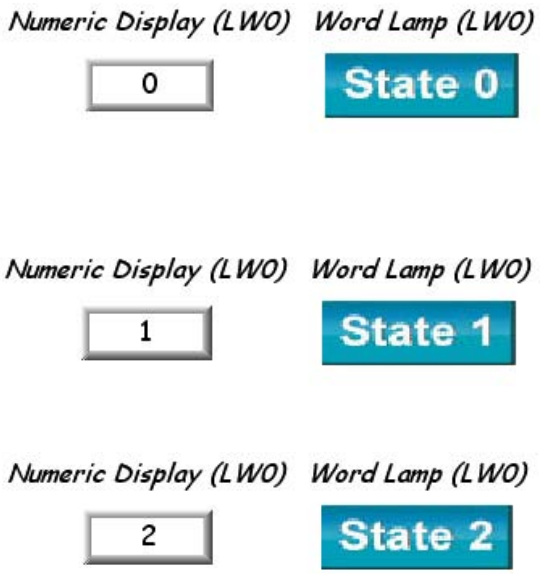
Blinking on state 0 Отображение состояния 0 с мерцанием при значении бита OFF.

Blinking on state 1 Отображение состояния 1 с мерцанием при значении бита ON.

13.2 Word Lamp - Лампа управляемая словом

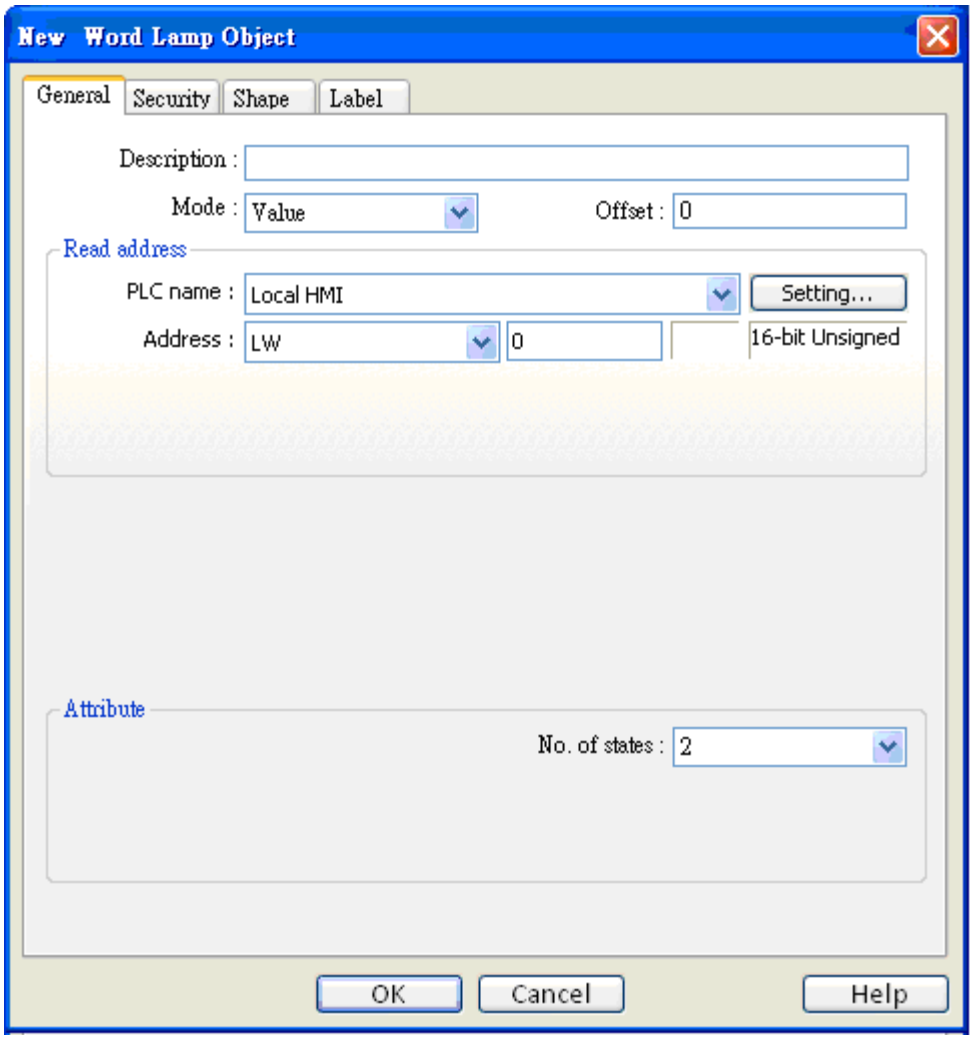
Краткое описание

Объект [Word Lamp] отображает состояние, соответствующее значению, находящемуся по указанному адресу области LW (максимум 256 состояний).



Настройка – Configuration

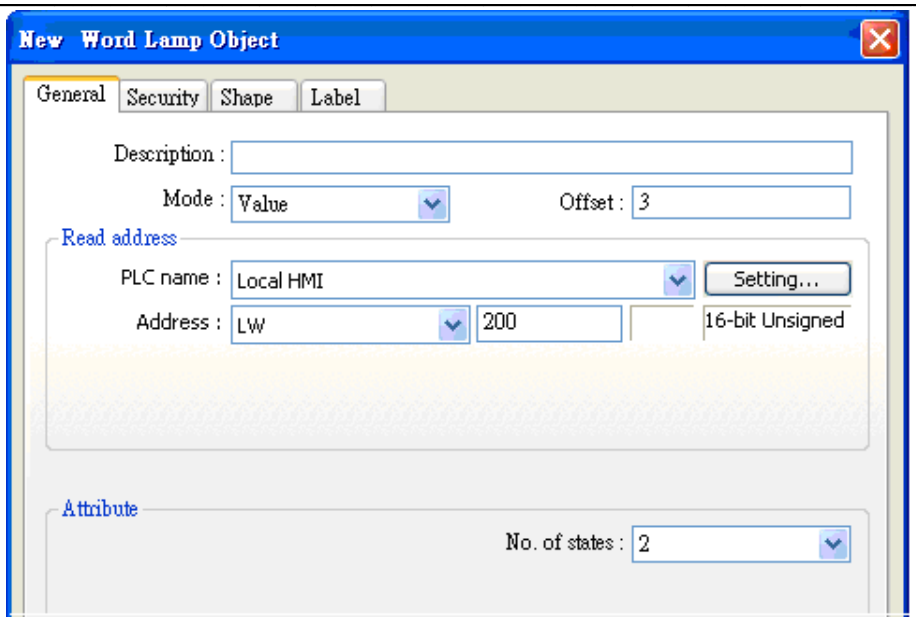
Щелкните пиктограмму **[Word lamp]** на панели инструментов — откроется окно задания свойств **[Word Lamp Object Properties**]. Заполните поля значениями и нажмите **[OK]** — будет создан новый объект [Word lamp].



[Mode] / [Offset] – Режим/Смещение

Объект [Word lamp] имеет три режима:

a. Value В этом режиме вычисляется разность значения и заданного смещения [offset] и отображается соответствующее состояние.



При настройках, показанных выше, если по адресу [LW200] находится значение 5, то будет показано состояние, соответствующее 2.

a

b. LSB Преобразует считанное значение в двоичное, младшие 8 бит, отличные от 0 определяют текущее состояние. См. таблицу ниже:

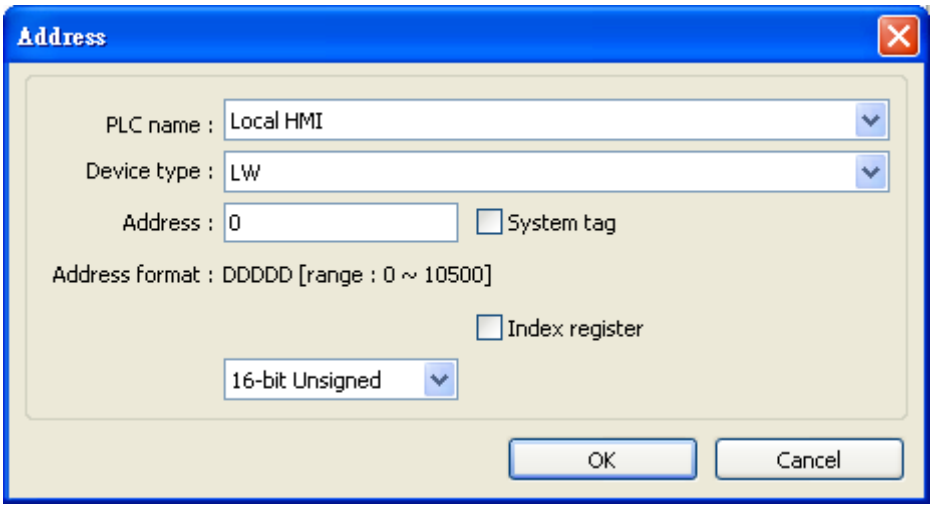
Считанное значение	Двоичное значение	Отображаемое состояние
0	0000	Все биты нулевые, отображается состояние 0.
1	0001	Самый младший бит со значением, отличным от 0 – нулевой, отображается состояние 1.
2	0010	Самый младший бит с ненулевым значением имеет номер 1, отображается состояние 2.
3	0011	Самый младший бит с ненулевым значением имеет номер 0, отображается состояние 1.
4	0100	Самый младший бит с ненулевым значением — второй, отображается состояние 3.
7	0111	Самый младший бит с ненулевым значением – нулевой, отображается состояние 1.
8	1000	Самый младший бит с ненулевым значением имеет номер 3, отображается состояние 4.

c. Change state by time

Состояние объекта не связано с каким-либо значением.
Система отображает различные состояния в соответствии с заданной частотой.

Read address

Нажмите **[Setting…] ,** чтобы выбрать слово (word device), управляющее объектом [word lamp] - **[PLC name], [Address], [Device type], [System tag], [Index register]**. Пользователь может назначить адрес также в закладке **[General]** при добавлении нового объекта.



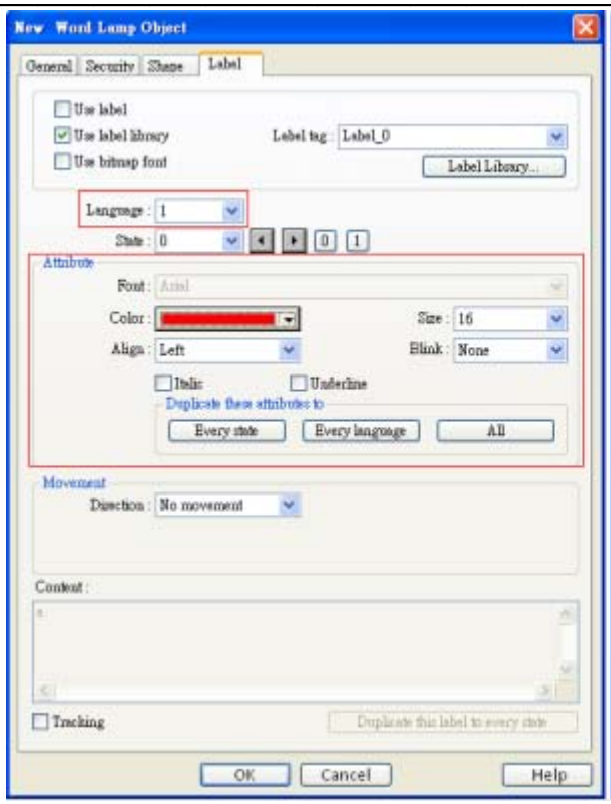
Attribute

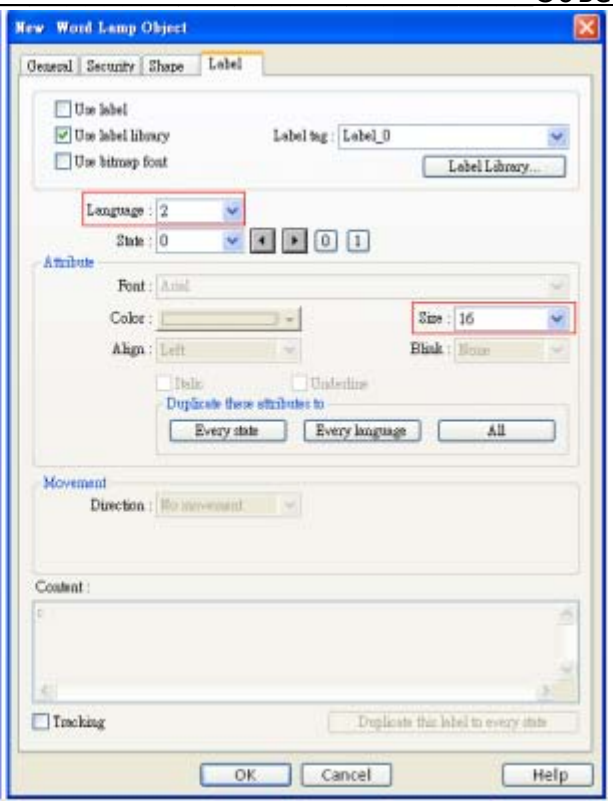
[No. of states]

Количество состояний одного объекта. Нумерация состояния начинается от нулевого. Например, при числе состояний **[No. of states]** равном 8, используемыми будут состояния: 0, 1, 2, …, 7. При превышении значения числа состояний , система будет отображать последнее состояние.

Restrictions - Ограничения

На закладке [Language 1] диалога [label] , можно менять настройки атрибутов, для языков *Language 2~8* можно изменять только размер шрифта, а другие настройки будут определяться настройками *Language 1*.


New Word Lamp Object
General
Security
Shape
Label
Use label
Use label library
Use bitmap font
Label tag : Label_0
Label Library...
Language : 1
State : 0
0
1
Attribute
Font : Arial
Color :
Size : 16
Align : Left
Blink : None
Italic
Underline
Duplicate these attributes to
Every state
Every language
All
Movement
Direction : No movement
Content :
Tracking
Duplicate this label to every state
OK
Cancel
Help

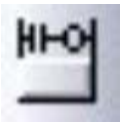

New Word Lamp Object
General
Security
Shape
Label
Use label
Use label library
Use bitmap font
Label tag : Label_0
Label Library...
Language : 2
State : 0
0
1
Attribute
Font : Arial
Color :
Size : 16
Align : Left
Blink : None
Italic
Underline
Duplicate these attributes to
Every state
Every language
All
Movement
Direction : No movement
Content :
Tracking
Duplicate this label to every state
OK
Cancel
Help

13.3 Set Bit – установить бит

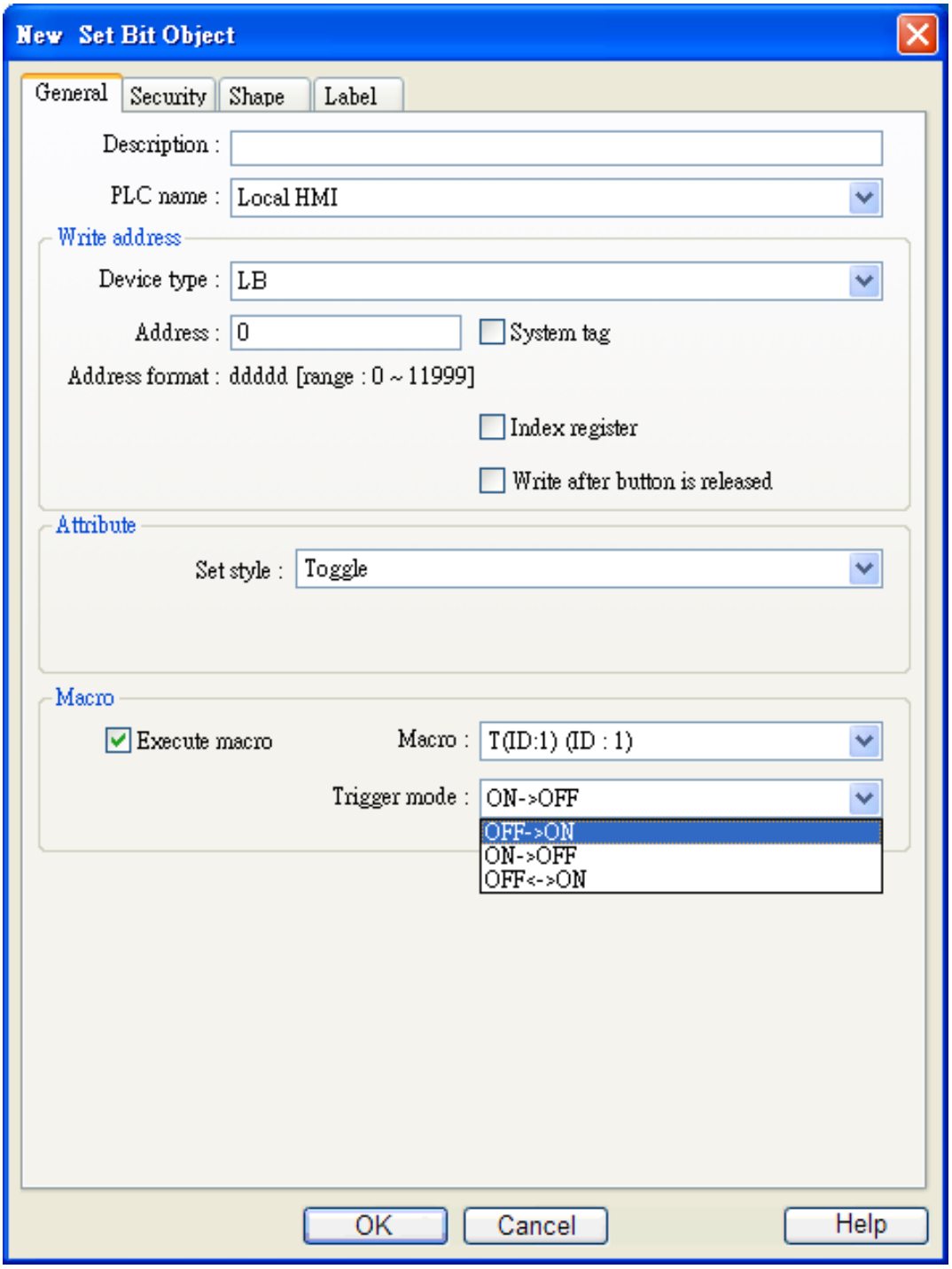
Краткое описание

Объект [Set Bit] имеет два режима работы: в режиме “manual operation” (ручном) задается зона экрана, касанием которой пользователь устанавливает состояние бита ON или OFF. При выборе режима “automatic operation” (автоматического) действие будет выполнено автоматически при выполнении заданных условий, при этом ручное воздействие не влияет ни при каких обстоятельствах.

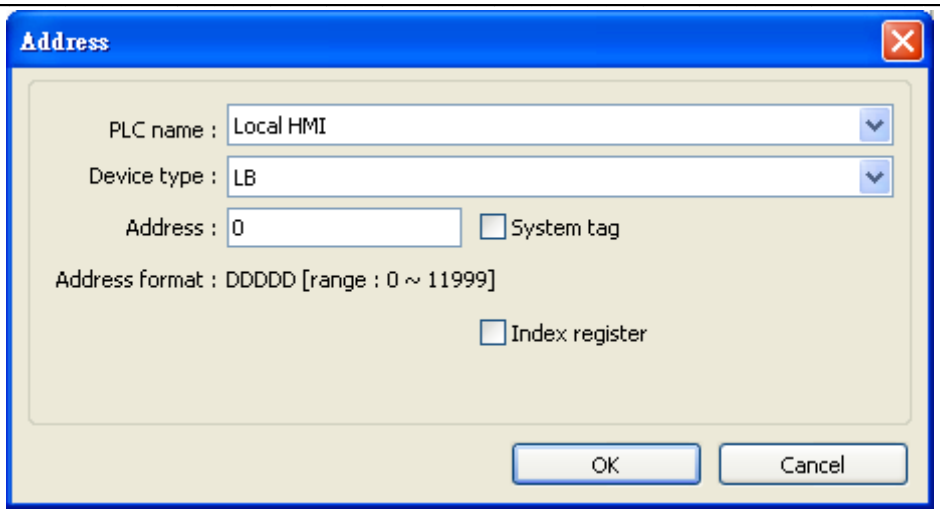
Настройка



Щелкните по пиктограмме **[Set Bit]** на панели инструментов — появится диалоговое окно **[New Set Bit Object];** заполните поля и нажмите **[OK]** — будет создан новый объект [Set Bit].



Write address Нажмите **[Setting…] ,** чтобы выбрать бит, значение которого будет изменять система - **[PLC name], [Address], [Device type], [System tag], [Index register]**. Пользователь может назначить адрес также в закладке **[General]** при добавлении нового объекта.

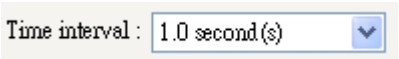


Write after button is release Если выбрана эта функция, то действие производится после отпускания сенсорной кнопки. Если функция не выбрана, действие осуществляется при касании кнопки.

Если выбран режим переключателя “Momentary”, то выбор [Write after button is released] не учитывается.

Attribute

[Set Style] – Выбрать стиль. Ниже описаны различные режимы работы.

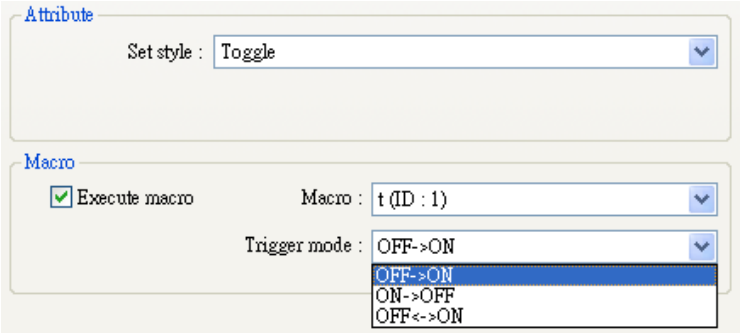
Set Style	**Описание**
Set ON	При активации действия бит (bit device) устанавливается в 1 (ON).
Set OFF	При активации действия бит устройства устанавливается в 0 (OFF).
Toggle	При активации действия состояние бита изменяется на противоположное ON > OFF или OFF > ON
Momentary	При касании и удержании кнопки, бит устанавливается в состояние ON, при прекращении касания — в OFF.
Periodical toggle	Состояние бита будет периодически переключаться между ON и OFF. Период времени выбирается в выпадающем списке. 

Set Style	Описание
Set ON when window open	При открытии окна, содержащего объект [Set Bit] бит автоматически устанавливается в состояние ON.
Set OFF when window open	При открытии окна, содержащего объект [Set Bit] бит автоматически устанавливается в состояние OFF.
Set ON when window close	При закрытии окна, содержащего объект [Set Bit] бит автоматически устанавливается в состояние ON.
Set OFF when window close	При закрытии окна, содержащего объект [Set Bit] бит автоматически устанавливается в состояние OFF.
Set ON when backlight on	При включении подсветки бит автоматически устанавливается в состояние ON.
Set OFF when backlight on	При включении подсветки бит автоматически устанавливается в состояние OFF.
Set ON when backlight off	При отключении подсветки бит автоматически устанавливается в состояние ON.
Set OFF when backlight off	При отключении подсветки бит автоматически устанавливается в состояние OFF.

Macro

Пользователь может использовать объект **[Set Bit]** для запуска макросов. Макрокоманды должны быть заданы до настройки данной функции. Обращайтесь к Главе 18 за информацией по созданию и редактированию макросов.

Set style - Trigger mode



Когда в поле [Set style] выбрано значение [Toggle] доступны три возможных режима запуска макрокоманд: OFF->ON, ON->OFF или ON<->OFF.

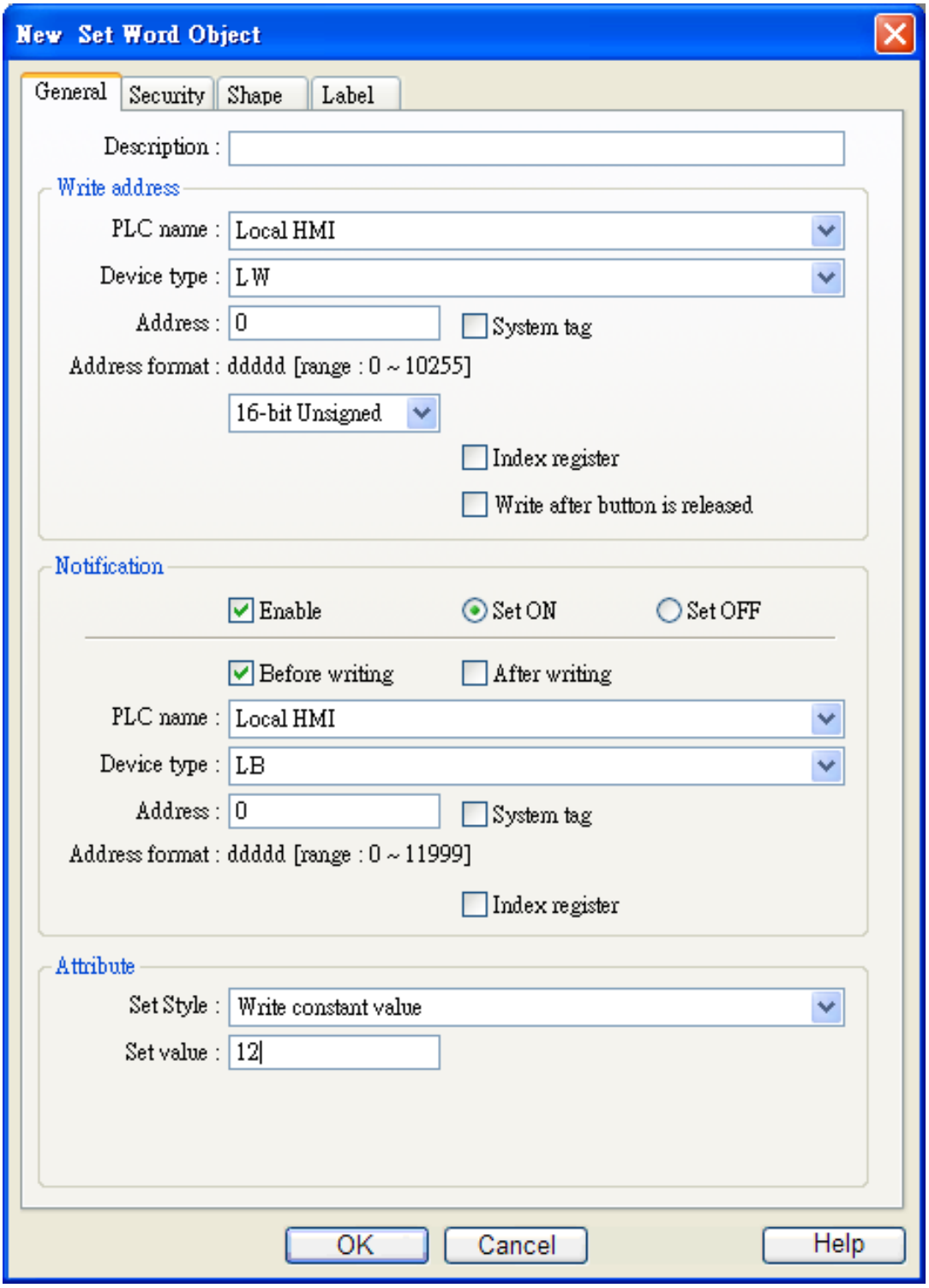
13.4 Set Word – Изменить слово

Краткое описание

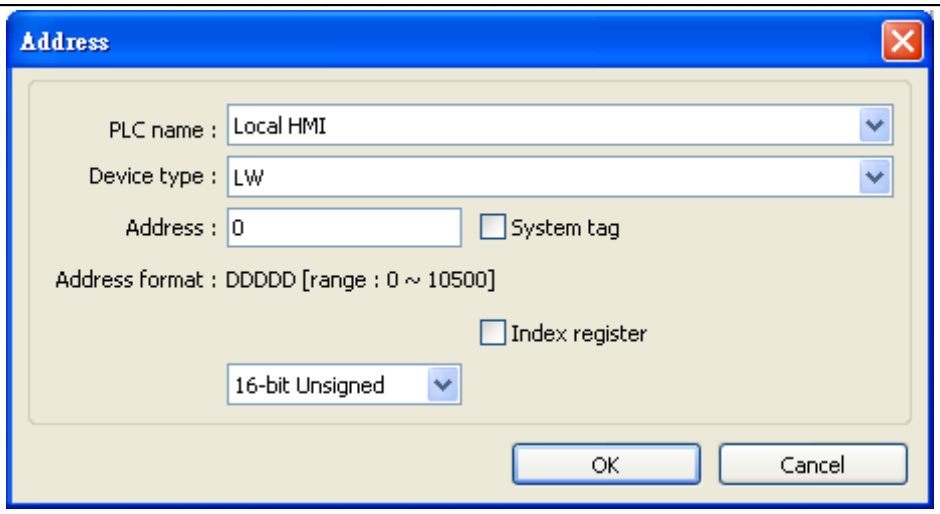
Объект [Set Word] имеет два режима работы: “manual operation” -(ручное управление) и “automatic operation” (автоматическое управление).
В режиме ручного управления задается зона экрана, касанием которой пользователь может активировать изменение значения слова (word device).
При выборе режима автоматического управления, действие будет выполнено автоматически при выполнении заданных условий, при этом зона касания не влияет на работу.

Настройка

Щелкните пиктограмму **[Set Word]** на панели инструментов — откроется диалоговое окно [**New Set Word Object]**, заполните поля, нажмите OK — будет создан новый объект [Set Word].



Write address Нажмите **[Setting…] ,** чтобы выбрать слово, значение которого будет изменять система - **[PLC name], [Address], [Device type], [System tag], [Index register]**. Пользователь может назначить адрес также в закладке **[General]** при добавлении нового объекта.

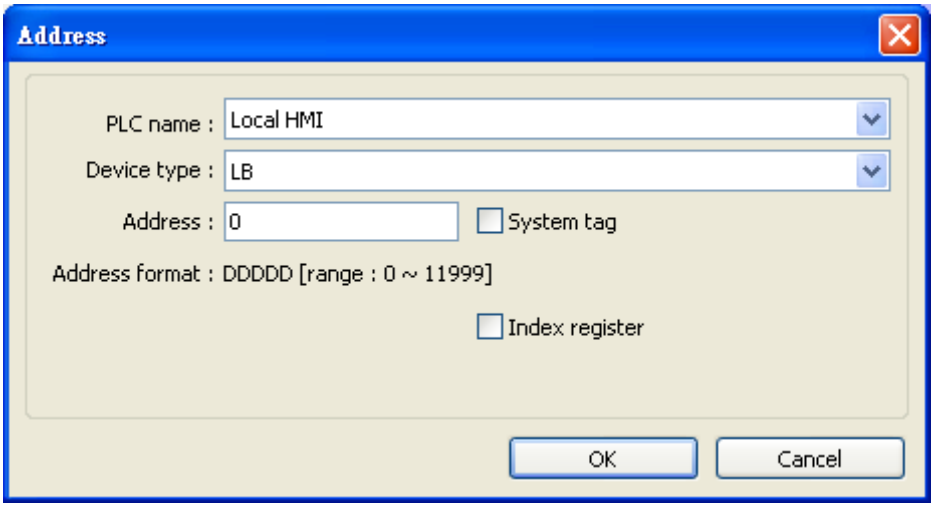


[Write after button is released] Если отмечена данная функция, то действие будет производиться при отпускании клавиши. Если не отмечена — при нажатии.

Notification - Уведомление Когда отмечена данная функция в ручном режиме, состояние бита указанного адреса будет установлено в ON или OFF «после/до» завершения действия.

[Before writing] /[After writing] Установить состояние бита «перед / после» записи значения типа word записью значения типа word.

Нажмите **[Setting…] ,** чтобы выбрать бит *Notification*, значение которого будет изменять система - **[PLC name], [Address], [Device type], [System tag], [Index register]**. Пользователь может назначить адрес также в *Notification area*.



Attribute

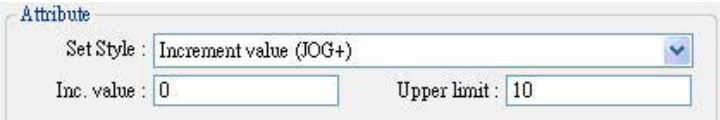
[Set style] Выбор способа выполнения действия. Доступные варианты:

a. Write constant value

Запись константы. Когда выбран данный вариант, по указанном адресу будет записываться значения из **[Set Value].** Формат числа (16-разрядное BCD, 32-разрядное BCD, и т.п.) определяется в **[Write address].**

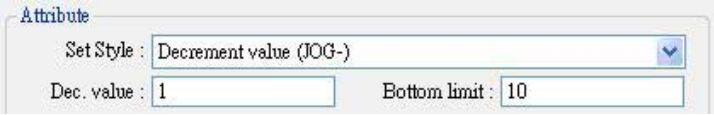
b. Increment value (JOG+)

Функция увеличения. При выборе данного варианта к числу, находящемуся по указанному адресу будет прибавляться значение **[Inc. value]**, результат не должен превышать значения в поле **[Upper limit].**



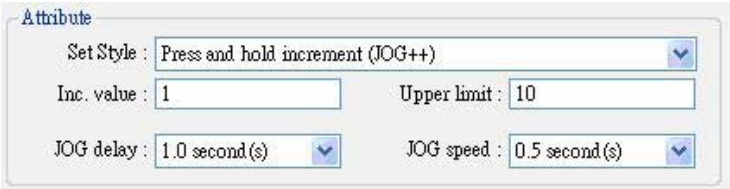
c. Decrement Value (JOG-)

Функция уменьшения значения. При выборе её из значения будет вычитаться число **[Dec. value]** при условии, что результат не меньше нижней границы **[Bottom line].**



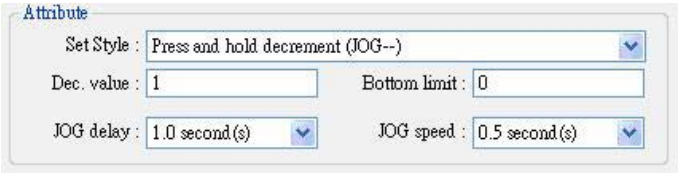
d. Press and hold increment (JOG++)

Функция увеличения при нажатии и удержании. Когда нажатие продолжается дольше установленного в поле **[JOG delay**] интервала времени, к числу будет прибавляться значение из поля **[Inc. value]** со скоростью, указанной в поле **[JOG speed]** при условии, что результат не превышает верхний предел **[Upper limit].**



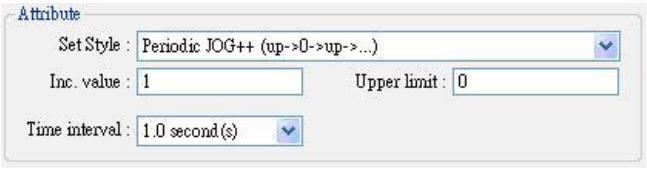
e. Press and hold increment (JOG--)

Функция уменьшения при нажатии и удержании. Когда нажатие продолжается дольше установленного в поле **[JOG delay]** интервала времени, из числа будет вычитаться значение из поля **[Dec. value]** со скоростью, указанной в поле **[JOG speed**] при условии, что результат не меньше нижнего предела **[Bottom limit].**



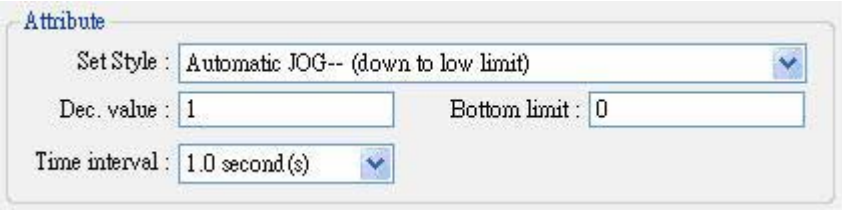
f. Periodical JOG++

Функция периодического увеличения. Параметр **[Time interval]** задает интервал времени, через который происходит автоматическое увеличение числа по указанному адресу на значение **[Inc. value]** при условии, что результат не превосходит верхний предел **[Upper limit]**.



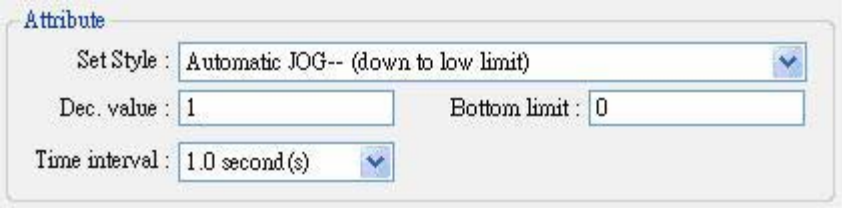
g. Automatic JOG++

Функция периодического увеличения. Параметр **[Time interval]** задает интервал времени, через который происходит автоматическое увеличения числа по указанному адресу на значение **[Dec. value]** при условии, что результат не больше верхнего предела **[Upper limit]**.



h. Automatic JOG--

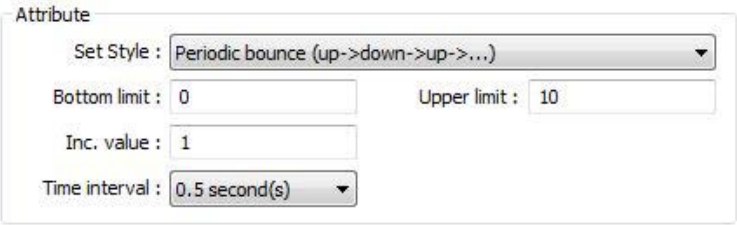
Функция периодического уменьшения. Параметр **[Time interval]** задает интервал времени, через который происходит автоматическое уменьшение числа по указанному адресу на значение **[Dec. value]** при условии, что результат не меньше нижнего предела **[Bottom limit].**



i. Periodical bounce – периодические скачки

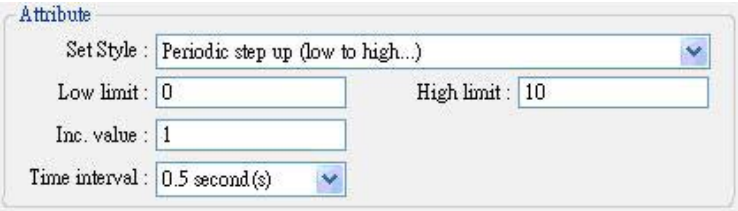
Функция периодического скачкообразного изменения. Объект *Set word* будет прибавлять к числу по указанному адресу значение **[Inc. value]** через интервалы времени, заданные в **[Time interval],** пока результат не достигнет верхнего предела **[Upper limit]**; затем будет производиться вычитание числа **[Inc. value]** в такие же интервалы времени, пока не будет достигнута нижняя граница [**Bottom limit].**

Например: значения по адресу будут изменяться в следующем порядке: 0, 1, 2, …, 9, 10, затем: 9,8, 7, …, 1, 0, 1, 2,….



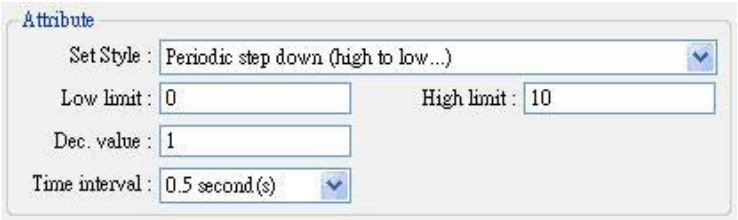
j. Periodical step up

Функция периодического ступенчатого увеличения. К числу по указанному адресу будет прибавляться значение **[Inc. value]** через интервалы времени, заданные в [Time interval] до достижения верхнего предела **[Upper limit],** после чего значение будет сброшено к нижней границе **[Bottom value]** и затем процесс увеличения повторится, чтобы поддерживать значение в активном состоянии. В примере последовательность изменения значения следующая: 0, 1, 2, …, 9, 10, 0, 1, 2,…… .



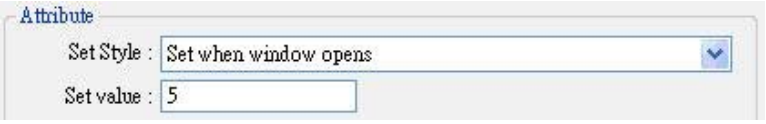
k. Periodical step down

Функция периодического ступенчатого уменьшения. Из числа по указанному адресу будет вычитаться значение **[Dec. Value]** через интервалы времени, заданные в [**Time interval]** до достижения нижнего предела [Bottom value], после чего значение будет возвращено к верхней границе **[Upper limit],** и затем процесс увеличения повторится, чтобы поддерживать значение в активном состоянии. В примере последовательность изменения значения следующая:
10, 9, 8, …, 1, 0, 10, 9, 8, …… .



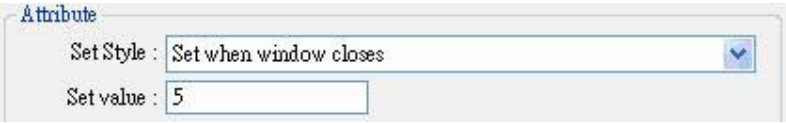
l. Set when window open

При открытии окна, содержащего объект **[Set Word]**, автоматически будет записано значение из поля **[Set value]** по указанному адресу.



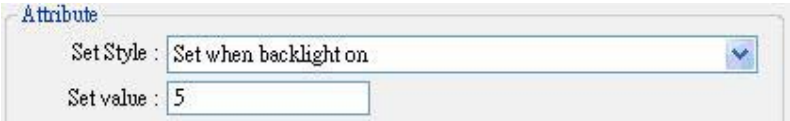
m. Set when window close

При закрытии окна, содержащего объект **[Set Word],** значение из поля **[Set value]** будет автоматически записано в указанный адрес.



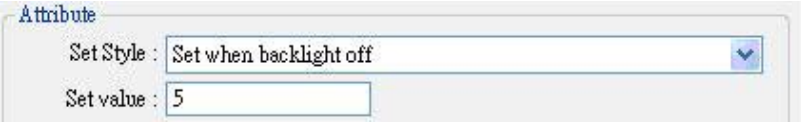
n. Set when backlight on

При включении подсветки значение из поля **[Set value]** будет автоматически записано по указанному адресу.



o. Set when backlight off

При отключении подсветки значение из поля **[Set value]** будет автоматически записано по указанному адресу.



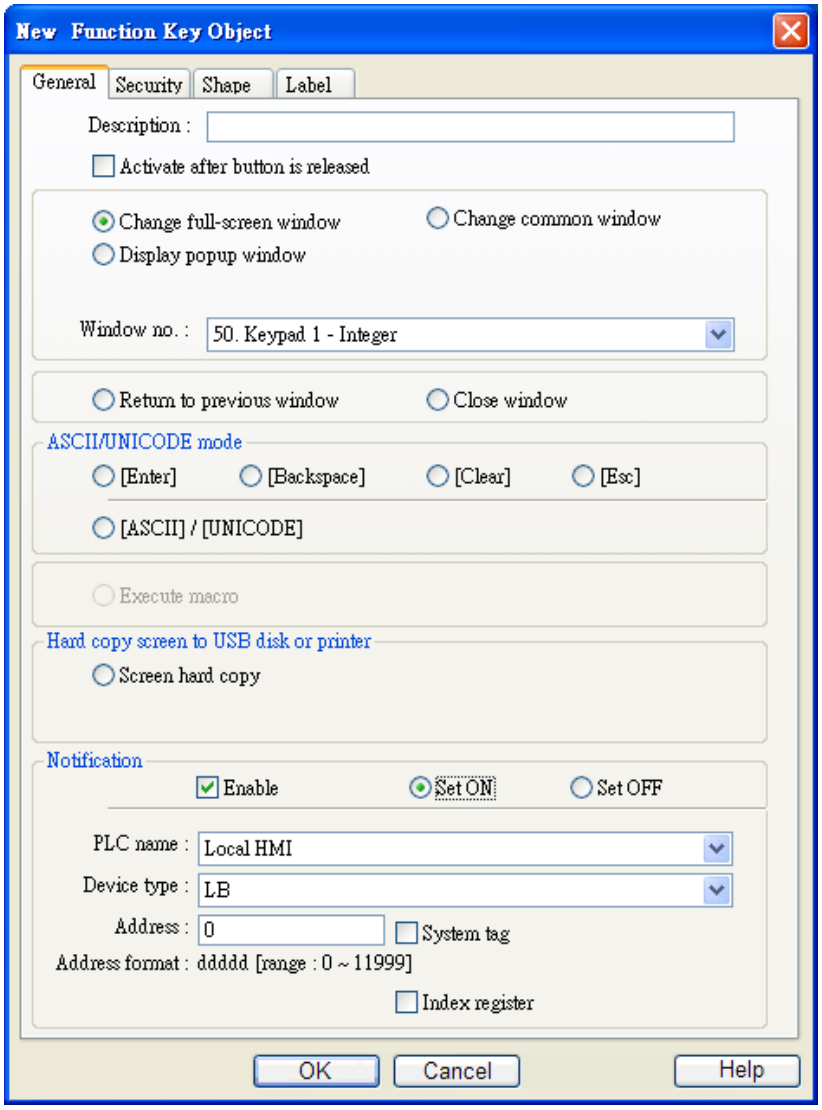
13.5 Function Key – Функциональная кнопка

Краткое описание

Объект *Function Key* используется для смены базового окна, вызова всплывающего окна и закрытия окон. Он может использоваться также при создании кнопок клавиатур.

Настройка

Щелкните пиктограмму **[Function Key]** на инструментальной панели — откроется диалоговое окно **[New Function Key Object]** задания свойств объекта; заполните поля, нажмите OK — будет создан новый объект.



Function Key имеет следующие режимы работы:

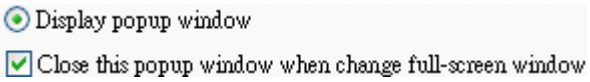
[Active after button is released] Если отмечена данная функция, то действие объекта начинается при отпускании кнопки. Если не отмечена, то действие запускается в момент нажатия.

[Change full-screen Window] Изменить основное окно.

Примечание: Не применяйте данную функцию для вызова окна, открытого объектами типа [direct/indirect window].

[Change Common Window] Изменить общее окно: см. главу «Работа с окнами» для получения соответствующей информации.

[Display Popup Window] Вызоваемое (всплывающее) окно должно быть в верхней части основного окна. Предусмотрен вариант **[Close this popup window when parent window is closed]**, когда всплывающее окно будет закрыто при закрытии исходного окна. Если флаг не отмечен, нужно установить кнопку [Close] в рабочем окне для его закрытия.



[Window no.] Данное поле необходимо для выбора окна при выполнении выбранных выше функций: [change base window], [change common window], [pop up the window].

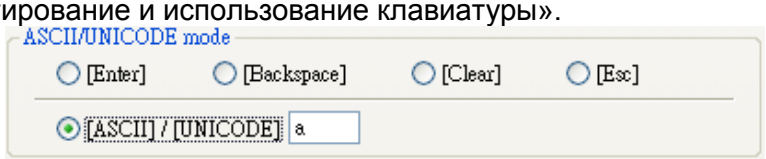
[Return to Previous Window] Возврат в предыдущее основное окно. Например, после смены window 10 на window 20, пользователь может использовать данную функцию для возврата к window 10. Данная функция применима только для базовых окон.

[Close window] Закрытие всплывающего окна вверху основного.

Items in ASCII/UNICODE Mode – Элементы используемые в режиме ASCII/UNICODE

[ASCII/UNICODE mode] используется при создании кнопок клавиатуры, которая нужна для ввода цифровых и буквенных символов в объекты

[Numeric Input] или **[ASCII Input].** Подробнее см. главу «Проектирование и использование клавиатуры».



[Enter]
Эквивалента функции клавиши [Enter] обыкновенной клавиатуры.

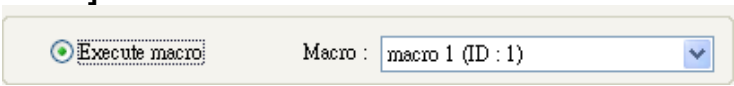
[Backspace]
То же, что и клавиша [Backspace] на обычной клавиатуре.

[Clear]
Для очистки буквенно-цифровых строк, находящихся в буфере обмена.

[Esc]
То же, что и функция [Close window] закрытия окна, данная опция используется для закрытия окна клавиатуры.

[ASCII/UNICODE]
Задание символов, вводимых в объектах типа [Numeric Input] или [ASCII Input]. Можно задавать цифровые символы: 0, 1, 2… или ASCII-символы – например, a, b, c и т.д.

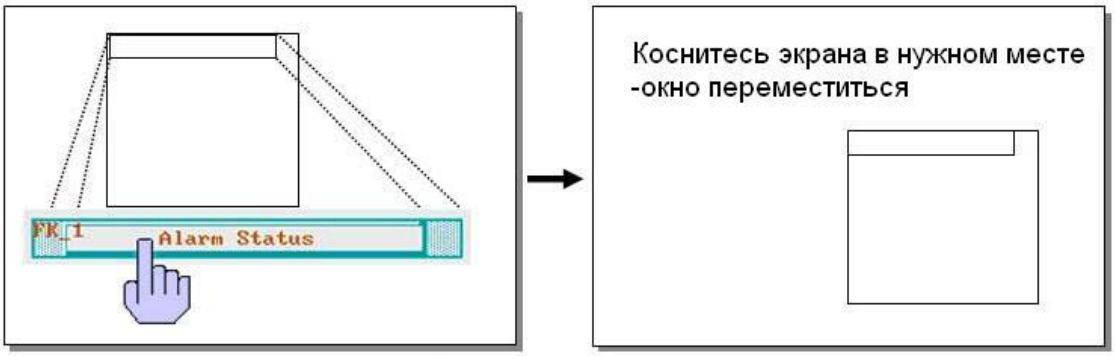
[Execute Macro]



При выборе данной опции выполняются макросы, которые должны быть созданы перед использованием этой функции. См. подробнее соответствующую главу по редактированию макросов.

[Window title bar]
Функциональная кнопка**,** которая определена как *Window Title Bar* - Полоса заголовка окна - может перемещать окно по экрану. Первоначально пользователи могут выбрать всплывающее окно с заголовком, а затем щелкнуть по другому месту экрана для перемещения окна.

Примечание: эта функция доступна для окон типа *indirect/direct* только, когда выбран вариант [no title bar].



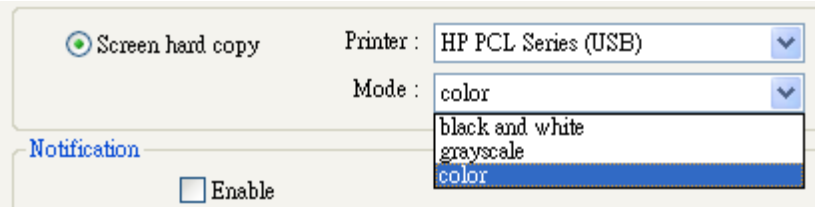
Выберите сначала заголовок окна

[Screen hard copy]

Отправка на печать текущего экрана на принтер, соединенный с панелью. Перед использованием данной функции выберите модель принтера в диалоговом окне **[System Parameter] / [Model] / [Printer].**

Если принтер не поддерживает цветную печать, можно указать **[grayscale]** (печать в градациях серого) для лучшего изображения.

Вариант **[black and white]** применяется для улучшения качества печати текста.



Import user data / Use [USB Security Key]

Пользователь может импортировать контактную информацию с внешнего устройства.
(см. Главу 36 - Administrator Tools).

[**Settings…]**
Function mode: Импорт настроек e-mail и контактов.

[Data Position]
Доступны USB1, USB2, SD карта.

[Account import mode]
Внутренняя память панели будет хранить только учетную информацию импортированную из внешнего устройства, если выбрано *Overwrite* - перезаписать.
Однако, если выбрано *Append*, панель добавит учетную информацию к существующей учетной записи.

[Delete file after importing user accounts]

Удалить учетную информацию на внешнем устройстве после успешного импортирования. Для того чтобы не было утечки информации.

Notification - Уведомление

Когда отмечена данная функция, состояние указанного бита будет установлено в ON или OFF после завершения действия.

Нажмите **[Setting…] ,** чтобы выбрать бит *Notification*, значение которого будет изменять система - **[PLC name], [Address], [Device type], [System tag], [Index register]**.

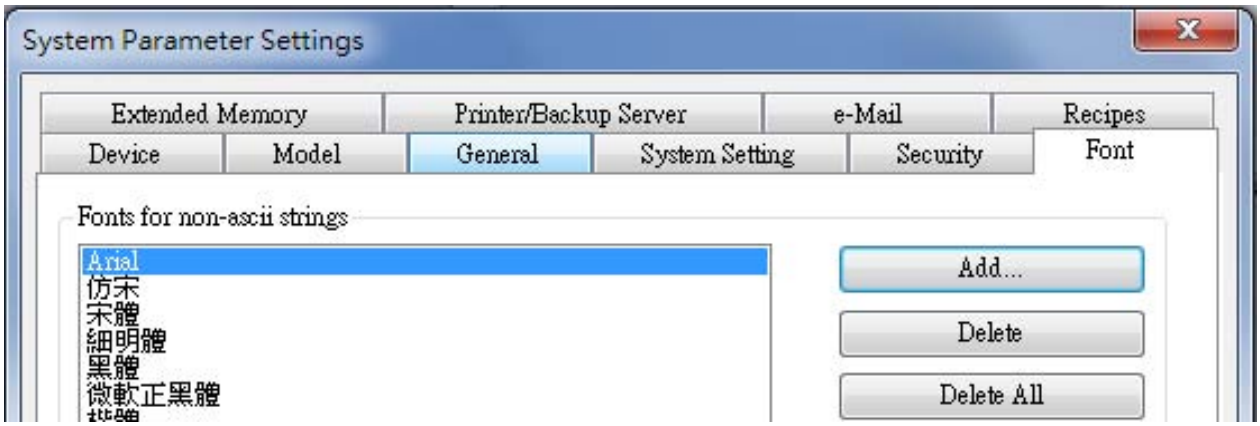
Пользователь может назначить адрес также в *Notification area*.

Клавиатура с кодировкой символов отличной от ASCII

Ниже приведен способ ввода символов не из таблицы ASCII — из таких языков, как традиционный китайский, упрощенный китайский, японский, греческий и т.п.

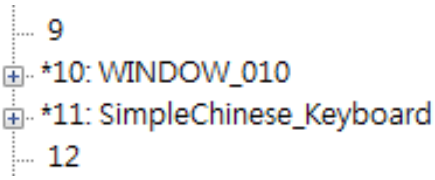
Шаг 1: Установка шрифтов с другой кодировкой

Зайдите на вкладку [Font] диалогового окна [System Parameter Setting] и добавьте нужные шрифты в список “Fonts for non-ascii strings”. Например, для японского языка используйте “AR MinchoL JIS”, “AR MingtiM GB” для упрощенного китайского, “AR MingtiM KSC” для корейского, “Arial” для греческого.

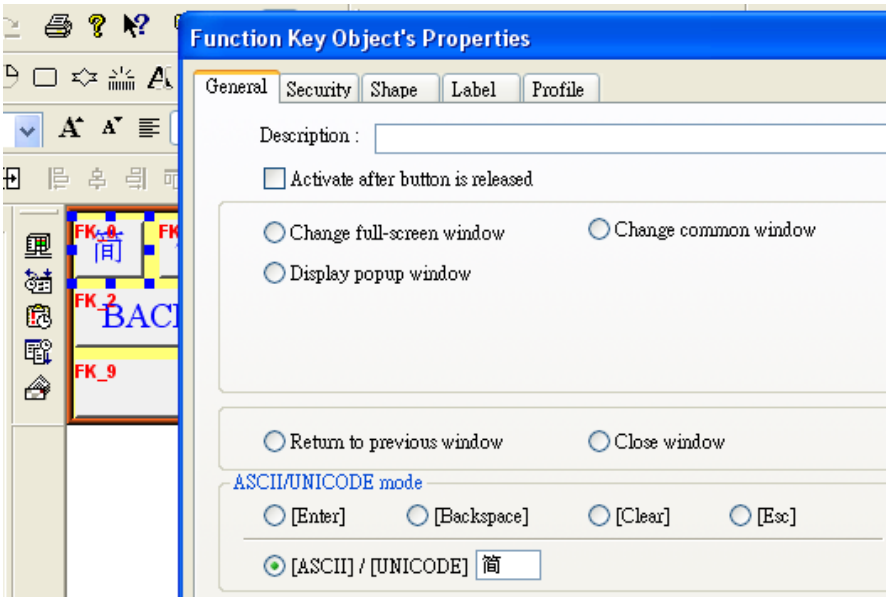


Шаг 2: Создание клавиатуры для ввода символов с кодировкой, отличной от ASCII

Создайте окно “Window 11” клавиатуры для ввода таких символов, ниже показан пример.

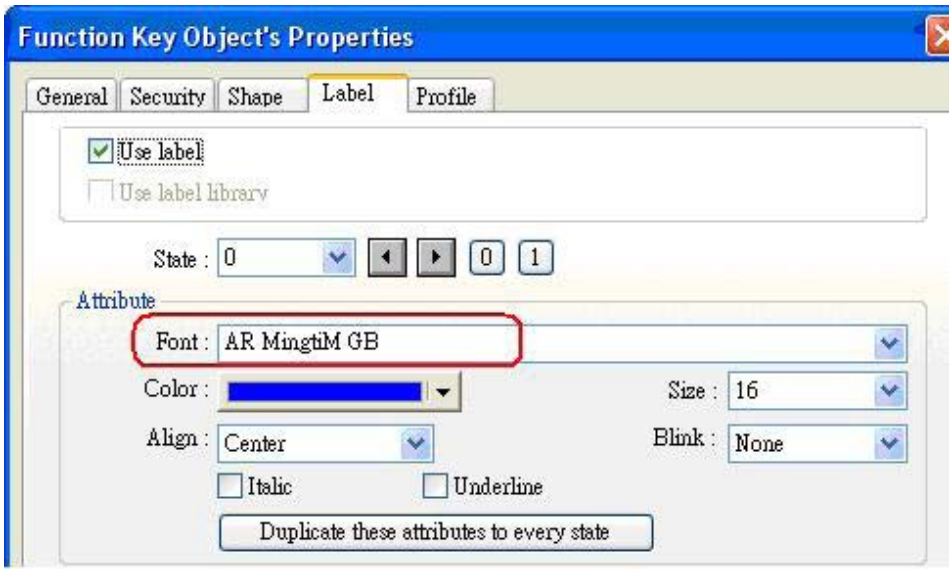


В окне клавиатуры используются объекты [Function Key], предназначенные для ввода изображенного на них символа. Например, чтобы создать клавишу символа ”简”, создайте объект [Function Key], в его свойствах в режиме [ASCII]/[UNICODE] задайте символ ”简” как показано ниже.

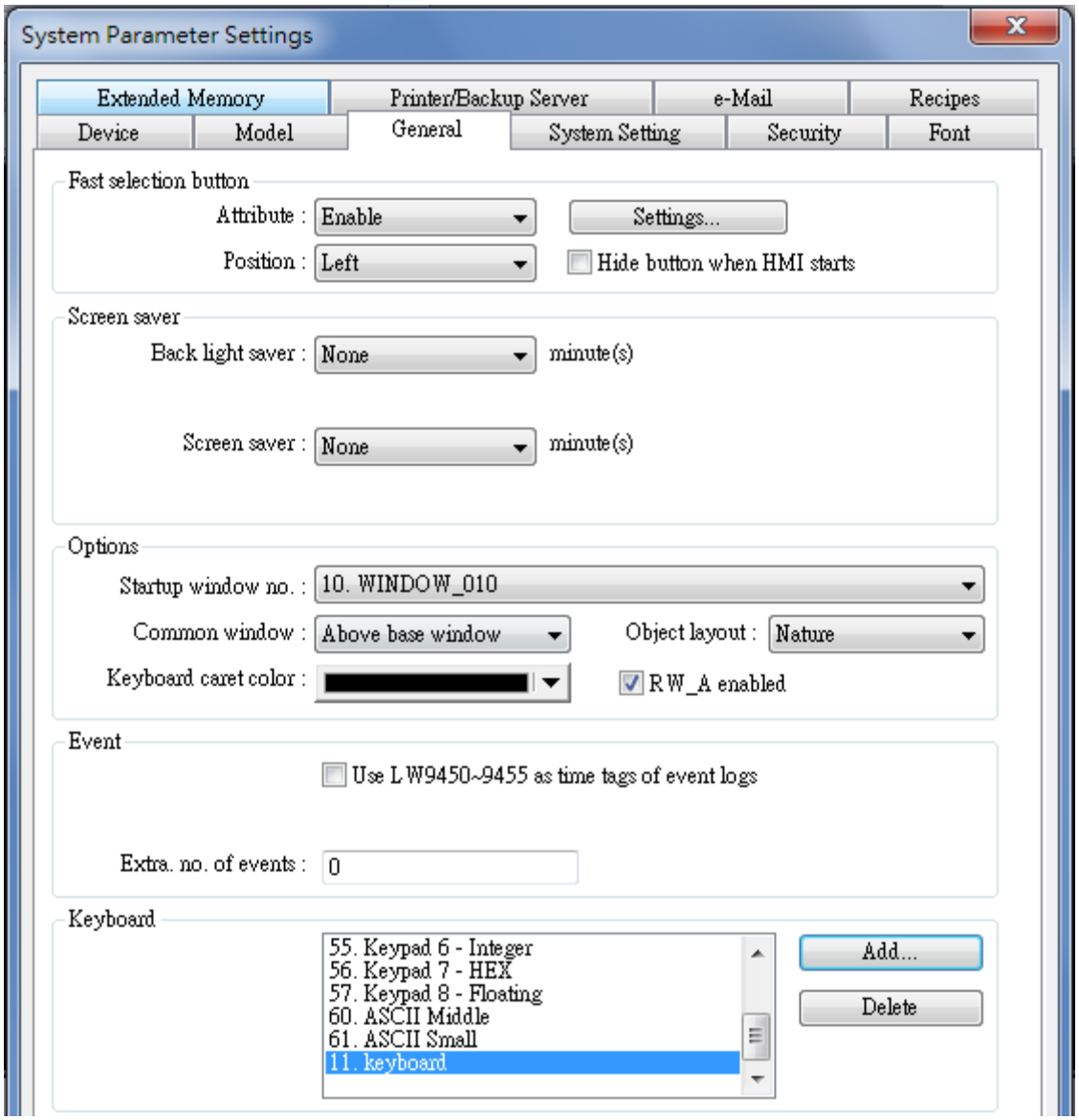


Зайдите на вкладку [Label] окна задания свойств объекта, отметьте [Use label], в поле [Content] укажите символ ”简” и в области окна [Attribute] в качестве шрифта [Font] выберите “AR MingtiM GB”; он должен быть добавлен на шаге 1.

Для метки клавиши ввода символа другой кодировки должен использоваться тот же шрифт. Например, для клавиатуры упрощенного китайского, все шрифты должны быть “AR MingtiM GB”.



После завершения настройки клавиатуры добавьте окно window 11 в список [Keyboard] на закладке [General] окна [System Parameters], как оказано на следующей странице.

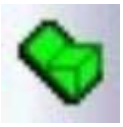

System Parameter Settings
Extended Memory
Printer/Backup Server
e-Mail
Recipes
Device
Model
General
System Setting
Security
Font
Fast selection button
Attribute :
Enable
Settings...
Position :
Left
Hide button when HMI starts
Screen saver
Back light saver :
None
minute(s)
Screen saver :
None
minute(s)
Options
Startup window no. :
10. WINDOW_010
Common window :
Above base window
Object layout :
Nature
Keyboard caret color :
RW_A enabled
Event
Use LW9450~9455 as time tags of event logs
Extra. no. of events :
0
Keyboard
55. Keypad 6 - Integer
56. Keypad 7 - HEX
57. Keypad 8 - Floating
60. ASCII Middle
61. ASCII Small
11. keyboard
Add...
Delete

13.6 Toggle Switch - Тумблер

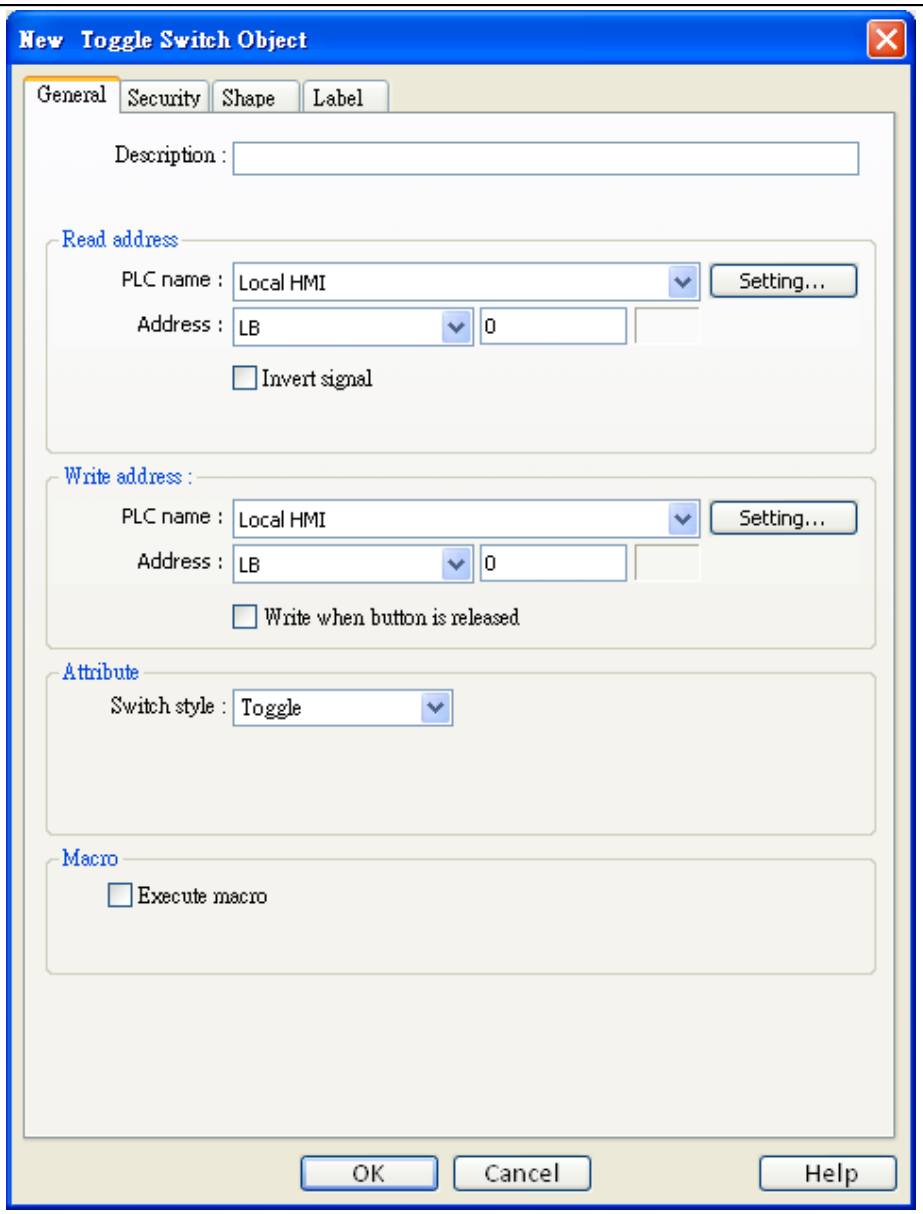
Краткое описание

Объект [Toggle Switch] — это комбинация объектов [Bit Lamp] и [Set bit]. Данный объект может использоваться не только для отображения состояния бита, но также для указания зоны экрана, при касании которой, состояние бита меняется с “ON” на “OFF”.

Настройка



Щелкните пиктограмму **[Toggle Switch]** на панели инструментов — появится окно **[New Toggle Switch Object],** заполните поля, нажмите ОК — будет создан новый объект.



Read address Нажмите **[Setting…],** чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** адрес бита, управляющего состоянием объекта. Пользователь может, также, задать адрес бита в закладке General при добавлении нового объекта

[Invert signal]

Отображать очертания в инверсном состоянии. Например, текущее состояние бита “OFF”, но очертание соответствует состоянию “ON”.

Write address В полях **[PLC name] [device type][address]** введите адрес бита, в который будет записываться значение. Этот адрес может совпадать с [Read address].

[Write when button is released]. Если выбрана эта функция, действие начнется при отпускании. Если выбрана эта функция, действие начнется при нажатии.

Attribute Выбор режима работы. Доступные режимы: “Set ON”, “Set OFF”, ”Toggle”, и ”Momentary”. См. иллюстрации в разделе «Set Bit».

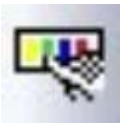
Macro Пользователь может запускать макросы, переключая [toggle switch]. См. Главу 18 - Создание макросов.

13.7 Multi-State Switch – Переключатель с несколькими состояниями

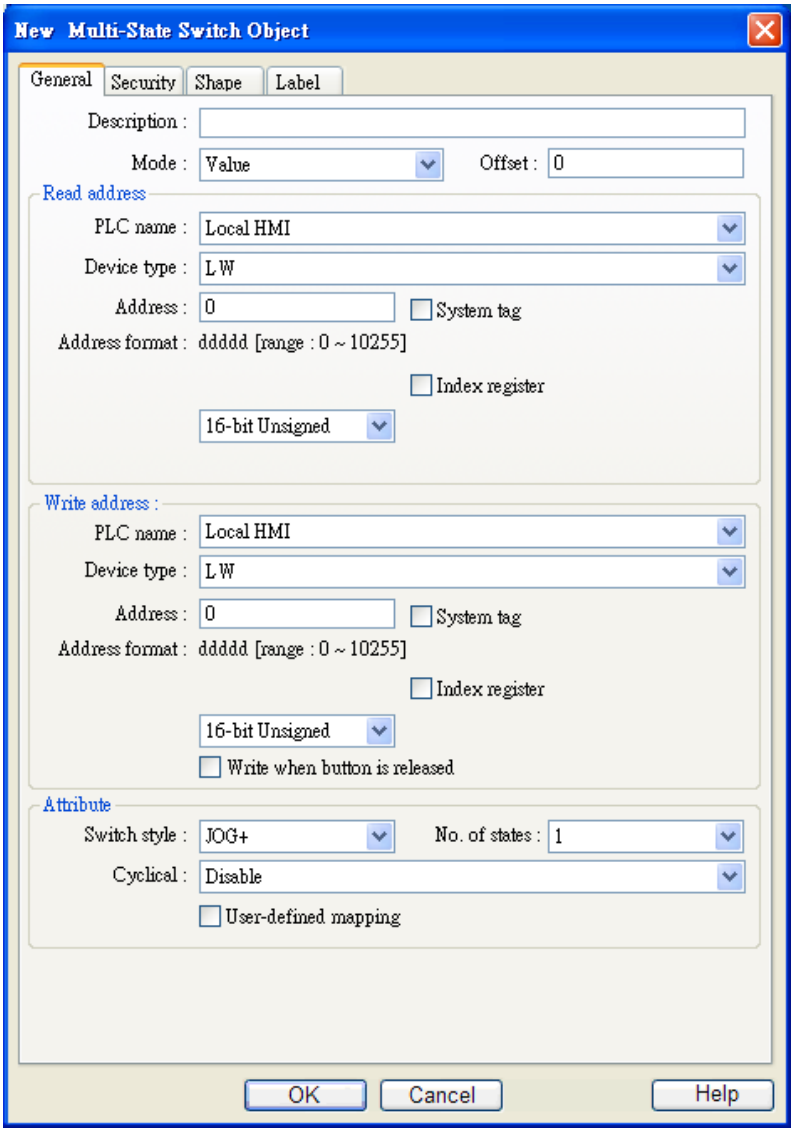
Краткое описание

Объект [Multi-State Switch] — это комбинация объектов [Word Lamp] и [Set word]. Данный объект может использоваться не только для отображения состояния по указанному адресу, но также для установления зоны экрана, при касании которой, может быть установлено значение слова.

Настройка



Щелкните пиктограмму **[Multi-State Switch]** на панели инструментов — появится окно **[New Multi-State Switch Object],** заполните поля, нажмите OK — будет создан новый объект.



[Mode]/ [Offset] Есть два режима отображения: “Value” и “LSB”. См. раздел «Объект [Word Lamp]» для получения информации.

Read address Нажмите **[Setting…],** чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** адрес слова, управляющего состоянием объекта.Пользователь может, также, задать адрес бита в закладке General при добавлении нового объекта

Write address Нажмите **[Setting…],** чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** адрес слова, в которое будет записываться значение. Этот адрес может совпадать с [Read address].
[Write when botton is released]
Если выбрана эта функция, действие начнется при отпускании. Если выбрана эта функция, действие начнется при нажатии.

Attribute Выбор режима работы объекта.

[Switch style]
Есть два варианта: “JOG+” и “JOG-”. Когда [read address] совпадает с [write address], минимальное значение слова равно [Offset] (состояние 0), максимальное значение: [no. of state] -1 + [Offset]. См.рисунок ниже

a. “JOG+”
Когда объект [Multi-State Switch] активен, значение по указанному в [write address] адресу увеличивается на 1. Когда выбран режим отображения [Value], если результат больше или равен значению [No. of States] + [Offset] и выбрано “Enable” в [Cyclic], то значение [write address] будет сброшено на [Offset] и показано состояние 0; в противном случае значение [write address] будет сохраняться равным ([No. Of states] – 1) + [Offset] и будет показано состояние ([No. of states no.] – 1).

Примечание: Как и для объекта [Word lamp], состояние, показываемое объектом [Multi-State Switch], будет равно разности числа по указанному адресу и [Offset].

b. “JOG-”
Когда объект [Multi-State Switch] активен, значение по указанному в [write address] адресу уменьшается на 1. Когда выбран режим отображения [Value], если результат меньше значения и выбрано “Enable” в [Cyclic], то значение [write address] будет возвращено на ([No. Of states] – 1) + [Offset] [Offset] и показано состояние ([No. Of states] – 1; в противном случае значение [write address] будет сохраняться равным [Offset] и будет показано состояние 0.

[User-defined mapping]
Пользователь может модиицировать значение состояния, недопустимый вход (illegal input) и уведомление об ошибке (error notification).
Remain current state (Оставить текущее состояние): если вход недопустимый, переключатель останется в текущем положении.
Jump to error state (Перейти в состояние по ошибке): если вход недопустимый, переключатель перейдет в положение по ошибке (безопасное значение).

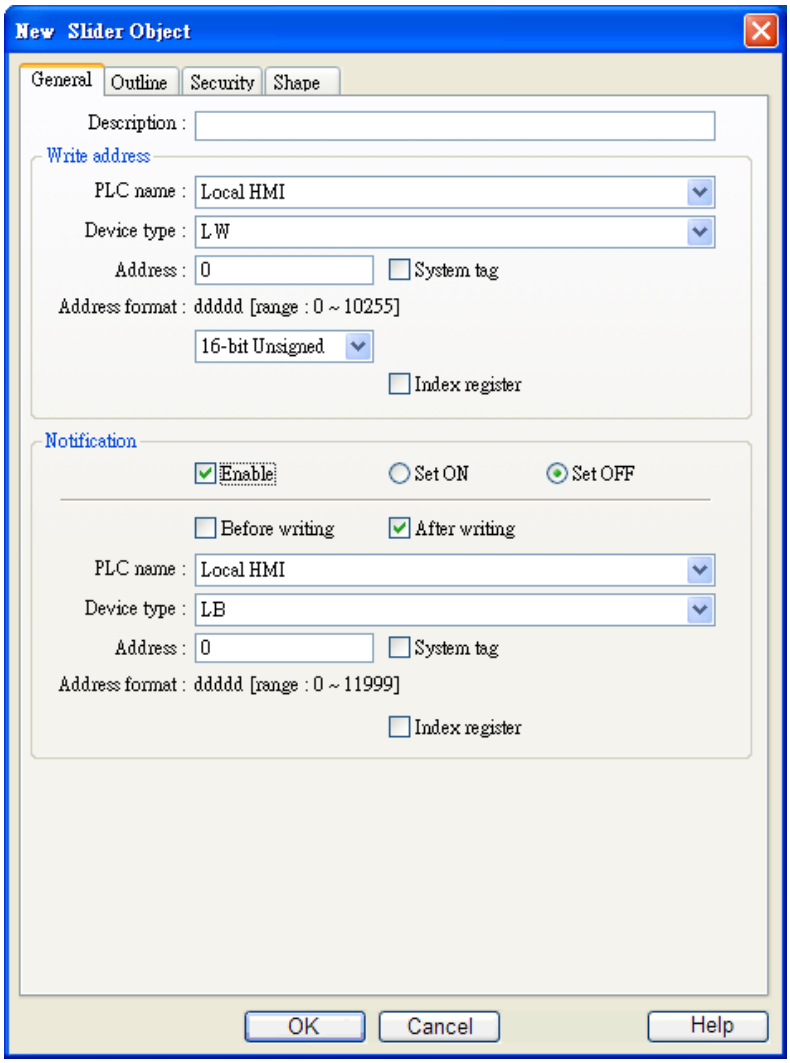
13.8 Slider - Ползунок

Краткое описание

Объект [Slider] может быть использован для создания слота, который изменяет значение слова при перемещении ползунка.

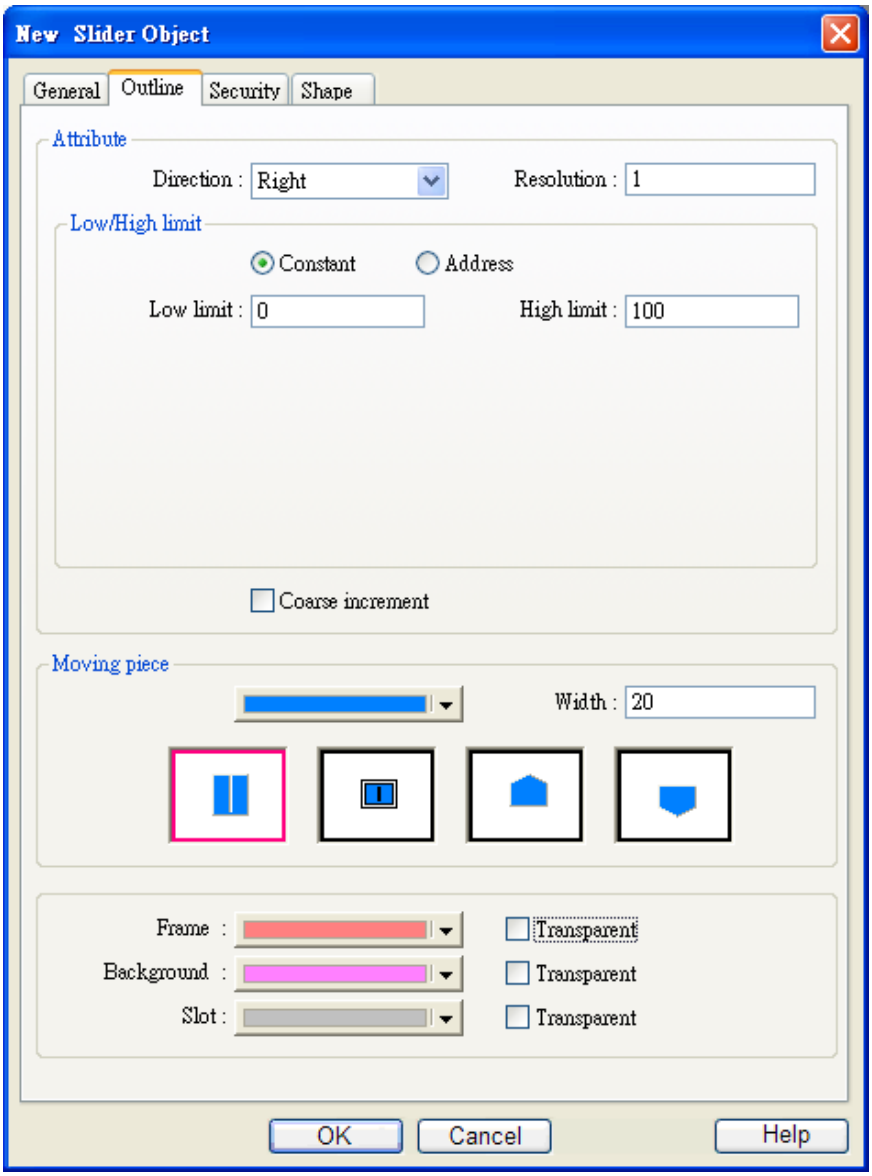
Настройка

Щелкните пиктограмму **[Slider]** на панели инструментов — появится окно **[New Slider Object],** заполните поля, нажмите OK — будет создан новый объект.



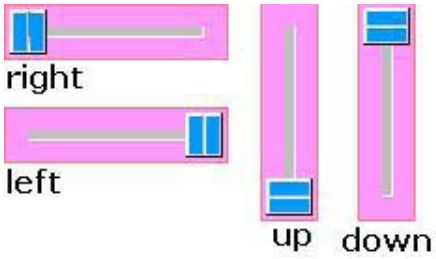
Write address — Нажмите **[Setting…],** чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** адрес слова, по которому система будет записывать значение.Пользователь может, также, задать адрес в закладке General при добавлении нового объекта укажите адрес,

Notification - Уведомление — Нажмите **[Setting…],** чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** адрес бита Notification,который будет взводить система. Пользователь может, также, задать адрес бита в *Notification area.*



Attribute

[Direction:]
Направление перемещения ползунка: left, right, up, down (влево, вправо, вверх и вниз).



[Resolution:]

Ползунок перемещается на каждые [N] шагов, где [N] — разрешение.

Например,

При [N]=10, будет отображаться каждый 10-й шаг перемещения

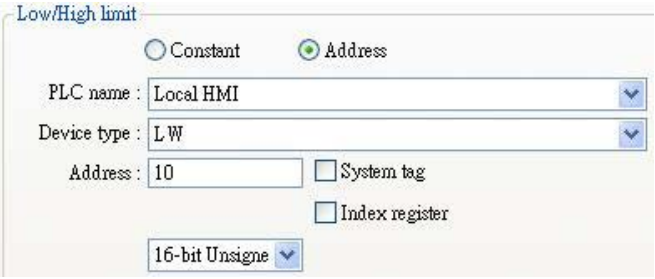
При [N]=5, будет отображаться каждый 5-й шаг перемещения

При [N]=1, будет отображаться каждый шаг перемещения

[Low limit & High limit:]

a. Constant
Нижний и верхний пределы значения по указанному адресу заданы как константы, например: [Input low] и [Input high].

b. Address
Нижний и верхний пределы равны значениям по указанным адресам. Нажмите **[Setting…],** чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** адреса. Пользователь может, также, задать адрес в *Attribute*.



Управляющий адрес	Нижний предел (Low Limit)	Верхний предел (High Limit)
16-битный формат	Address+0	Address+1
32-битный формат	Address+0	Address+2

[Coarse increment:] – Грубое изменение

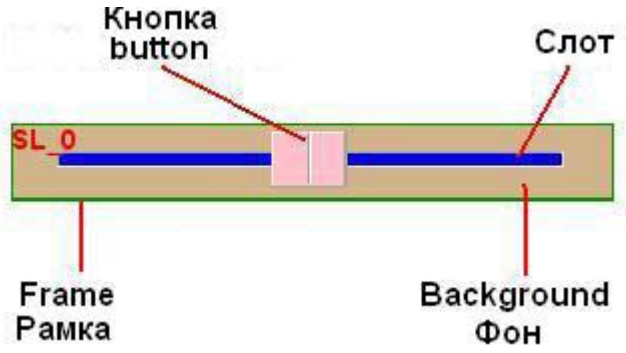
Если выбрана данная опция, значение слова будет увеличивать/уменьшаться на величину [increment] при каждой активации. Если не выбрана, то значение будет установлено в соответствии с установленным положением ползунка.

Slider button type

Можно выбрать один из четырех ползунков(кнопок). Можно также настроить его ширину.

Color

Выбор цвета рамки, фона и паза Frame/Background/Slot, по которому перемещается ползунок.



13.9 Numeric Input – Цифровой вход и Numeric Display – Цифровой дисплей

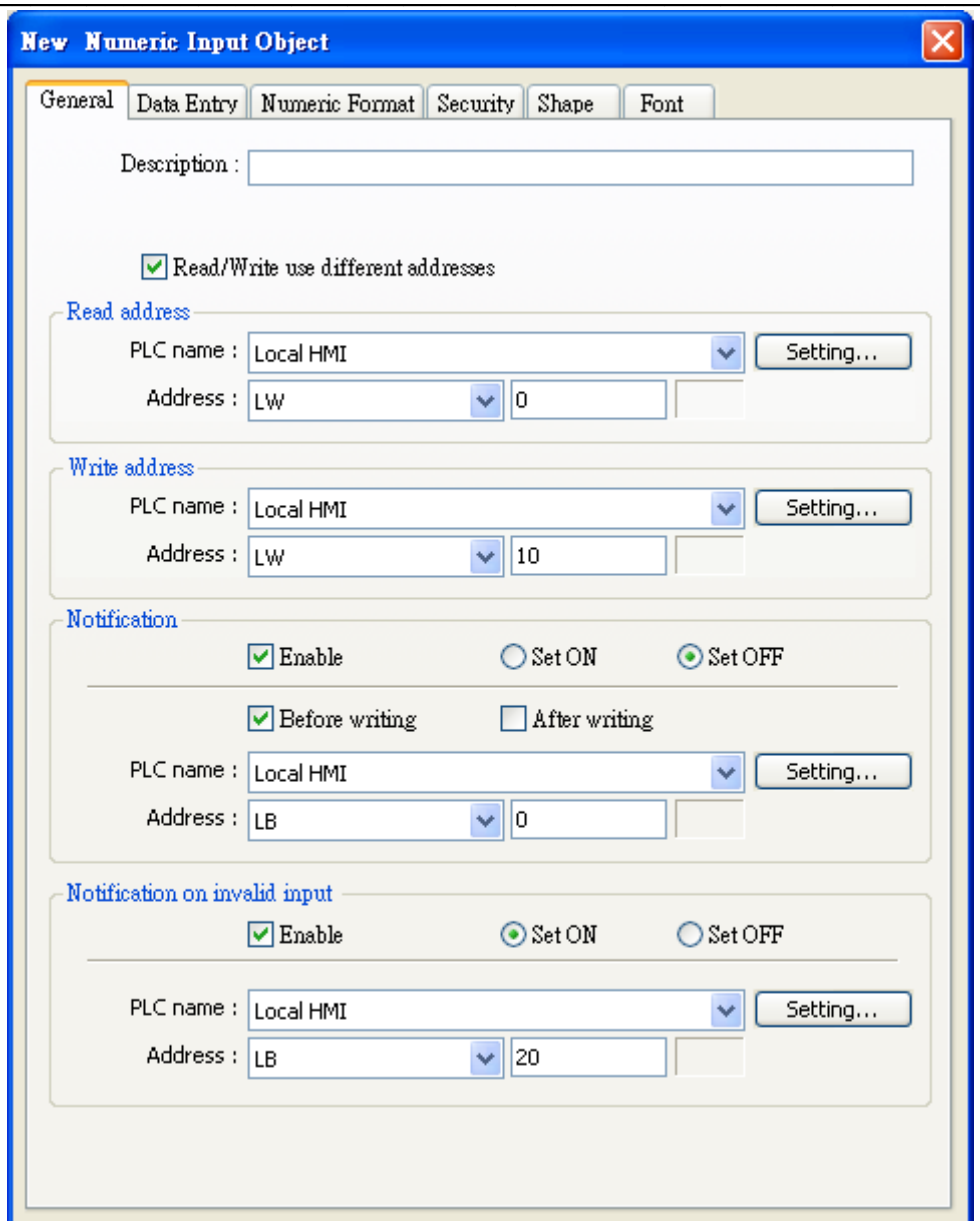
Краткое описание

Оба объекта [Numeric Input] и [Numeric Display] можно использовать для отображения значения слова по адресу. Отличие объектов в том, что [Numeric Input] может быть также использован для ввода значения с клавиатуры в слово по указанному адресу.

Настройка

Щелкните на пиктограмме **[Numeric Input]** или **[Numeric Display]** на панели инструментов и откроется окно **[New Numeric Input Object]** или **[New Numeric Display Object]**, заполните их поля, нажмите ОК — будет создан новый объект.

Отличаются окна настройки данных объектов тем, что в окне [New Numeric Input Object] имеется области настройки [Notification] и ввода с клавиатуры. Далее показан вид вкладки [General] диалогового окна [New Numeric Input Object].

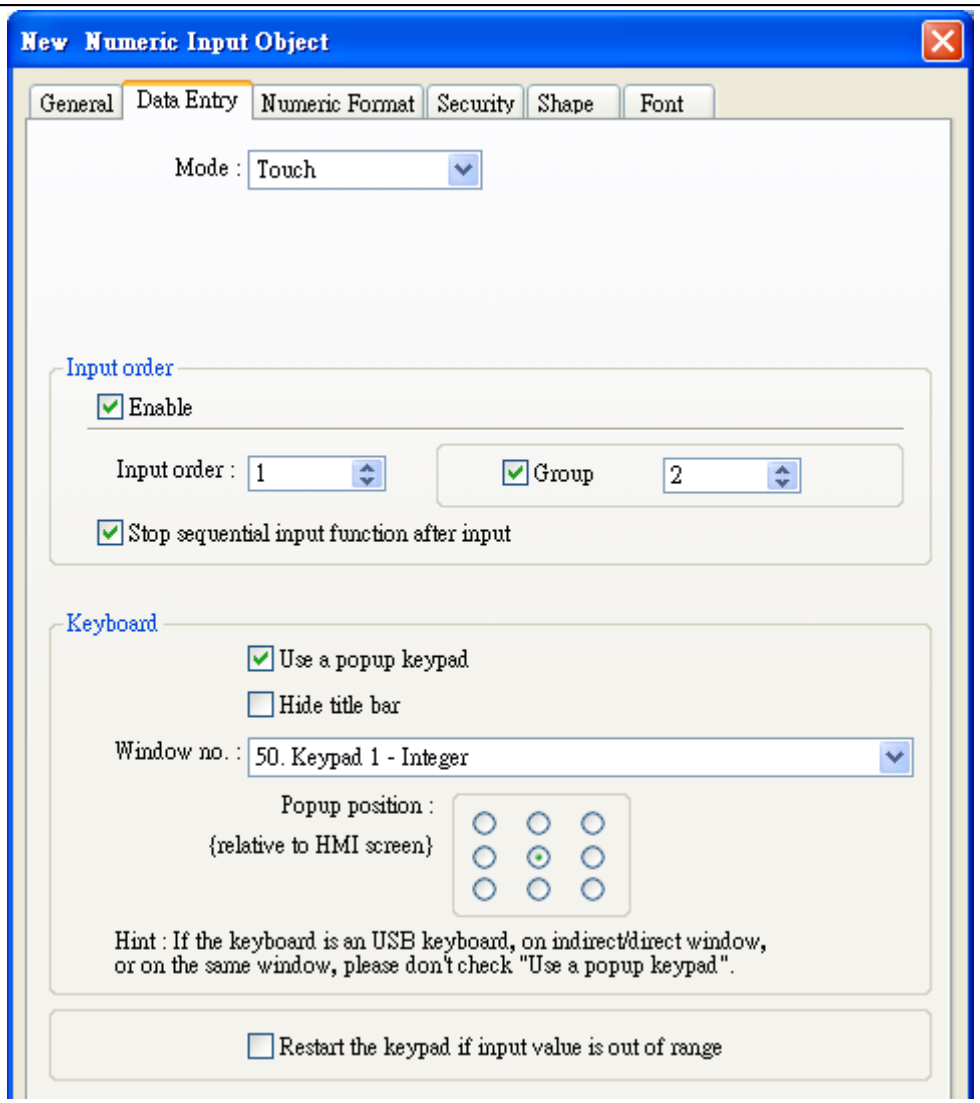


Read/Write use different address — У объекта “Numeric Input” есть выбор [Read/Write use different addresses] – Использовать различные адреса для чтения и записи данных.

Read address — В полях [PLC name] [device type][address] укажите адрес слова, содержимое которого будет отображаться и куда будет записываться новое значение.

Write address — В полях **[PLC name] [device type][address]** укажите адрес слова, куда будет записываться новое значение.

Notification - Уведомление	Когда данная функция выбрана, при успешном изменении значения указанного регистра, состояние назначенного бита будет устанавливаться в [ON] или [OFF]. Нажмите **[Setting…],** чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** адрес бита Notification,который будет взводить система. Пользователь может, также, задать адрес бита в *Notification area.* **[Before writing] / [After writing]** Установка состояния бита перед / после обновления слова.
Notification on invalid input	При вводе недопустимого значения, автоматически будет взводиться бит состояния по указанному адресу.



[Mode]

- **[Touch]**

Объект входит в состояние ввода при касании его пользователем.

- **[Bit control]**

Объект входит в состояние ввода , когда указанный битовый регистр переходит в состояние ON , и выходит из состояния ввода при переходе в OFF. Заметьте, что если есть другой объект ввода уже находится в состоянии ввода, то переход указанного бита в ON не переведет этот объект ввода в состояние разрешения ввода, пока первый не окончит вводданных.

Important

Панель будет дезактивировать всплывающую клавиатуру, когда выбран режим - *Bit Control*. Пользователи должны использовать внешнюю клавиатуру.

Allow input bit address

Нажмите **[Setting…],** чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** адрес битового регистра, который будет управлять состояниями ввода и окончания ввода. Пользователь может, также, задать адрес в закладке Data Entry.

Input Order Настраивая Input Order (Порядок ввода) и Input Order Group (Порядок ввода групп), пользователи могут вводить данные не прерываясь. Система будет автоматически переносить состояние ввода на следующий объект ввода после завершения ввода данных, например, после нажатия клавиши ENT.

● **Enable**

Выберите [**Enable**] для активации данной функции. Далее пользователи могут выбрать [Group] и установить порядок ввода групп [Input Order Group]

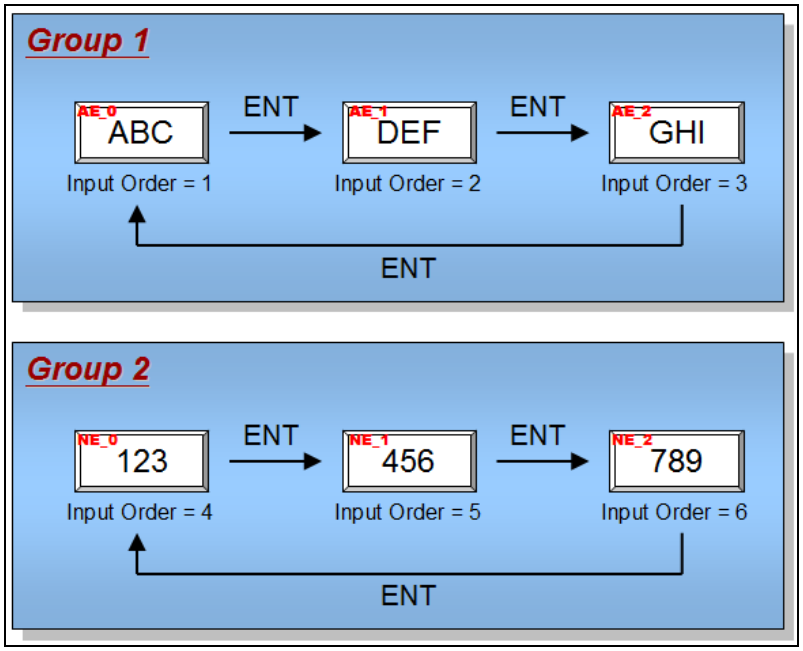
a. Диапазон *Input Order* — от 1 до 511
b. Диапазон *Input Order Group* — от 1 до 15.
c. *Input Order Group*=0, если объект ввода с [Group] не отмечен.

● **Критерии поиска следующего объекта ввода**

a. Система ищет их среди объектов в той же самой *Input Order Group*
b. Система выбирает объект с меньшим *Input Order* для перевода его в состояние ввода, прежде чем другой объект с большим *Input Order*
c. Если два объекта ввода имеют тот же самый *Input Order Group* и *Input Order* , система выберет объект, находящийся на нижнем слое, для перевода его в состояние ввода.

● **When selecting [Touch] as Mode – Когда выбран режим [Touch]**

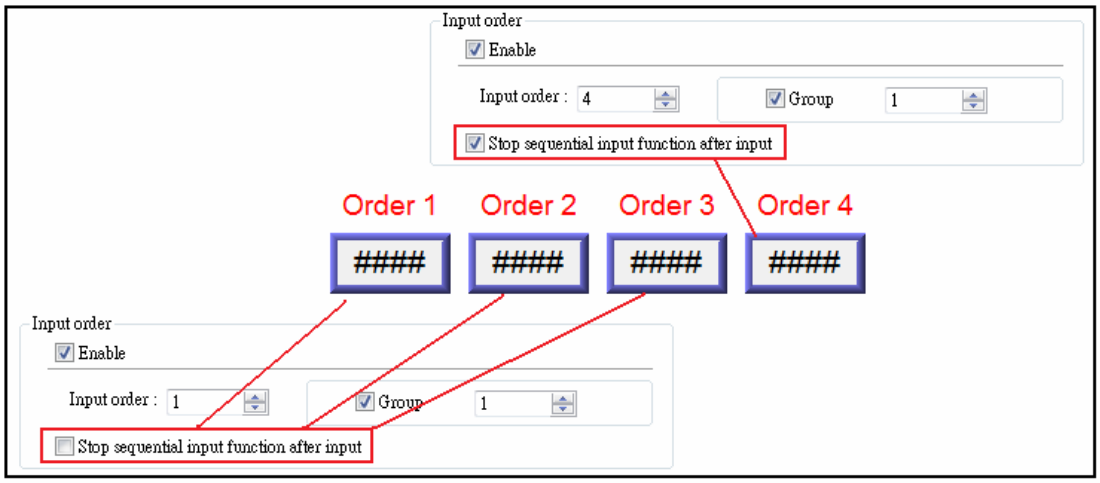
На рисунке показано что, когда пользователи закончат ввод данных на “AE_2”, система перенесет состояние ввода на “AE_0”. Причиной не переноса на “NE_0” в том что группа *Input Order Group “NE_0”* отлична от группы “AE2”.



[Stop sequential input function after input] – Остановить функцию последовательного ввода после ввода

Если объекты одной группы не настроены на эту функцию, порядок ввода будет:
[Order 1] -> [Order 2] -> [Order 3] -> [Order 4] -> [Order 1] -> [Order 2] ->….
И цикл продолжается до нажатия кнопки «ESC».

Если один объект группы настроен на [Stop sequential input function after input] (Объект *Order 4* на рисунке), порядок ввода будет:
[Order 1] -> [Order 2] -> [Order 3] -> [Order 4] ->Конец.
По завершении ввода объекта Order 4 (нажатия ENTER), ввод будет прекращен на этой точке.



● **Если выбран режим [Bit control]**

a. Пользователи должны указать для объекта *Input Order*.

b. Нет необходимости назначать *Input Order Group* , потому что все объекты ввода с режимом [Bit control] принадлежат одной и той же группе *Input Order Group* , которая отличается от группобъектов ввода с режимом [Touch].

Keyboard - Клавиатура

- **Выбрать [Use a popup keypad]**
  Укажите место всплывания окна клавиатуры. Система отображает окно клавиатуры при вводе данных и закрывает поокончании ввода.

- **Отменить [Use a popup keypad]**
  Система не отображает окно клавиатуры автоматически. Пользователь может осуществить ввод:
  a. Создав пользовательскую клавиатуру и поместив ее в окне, где расположен объект ввода.
  b. Использовать внешнюю клавиатуру

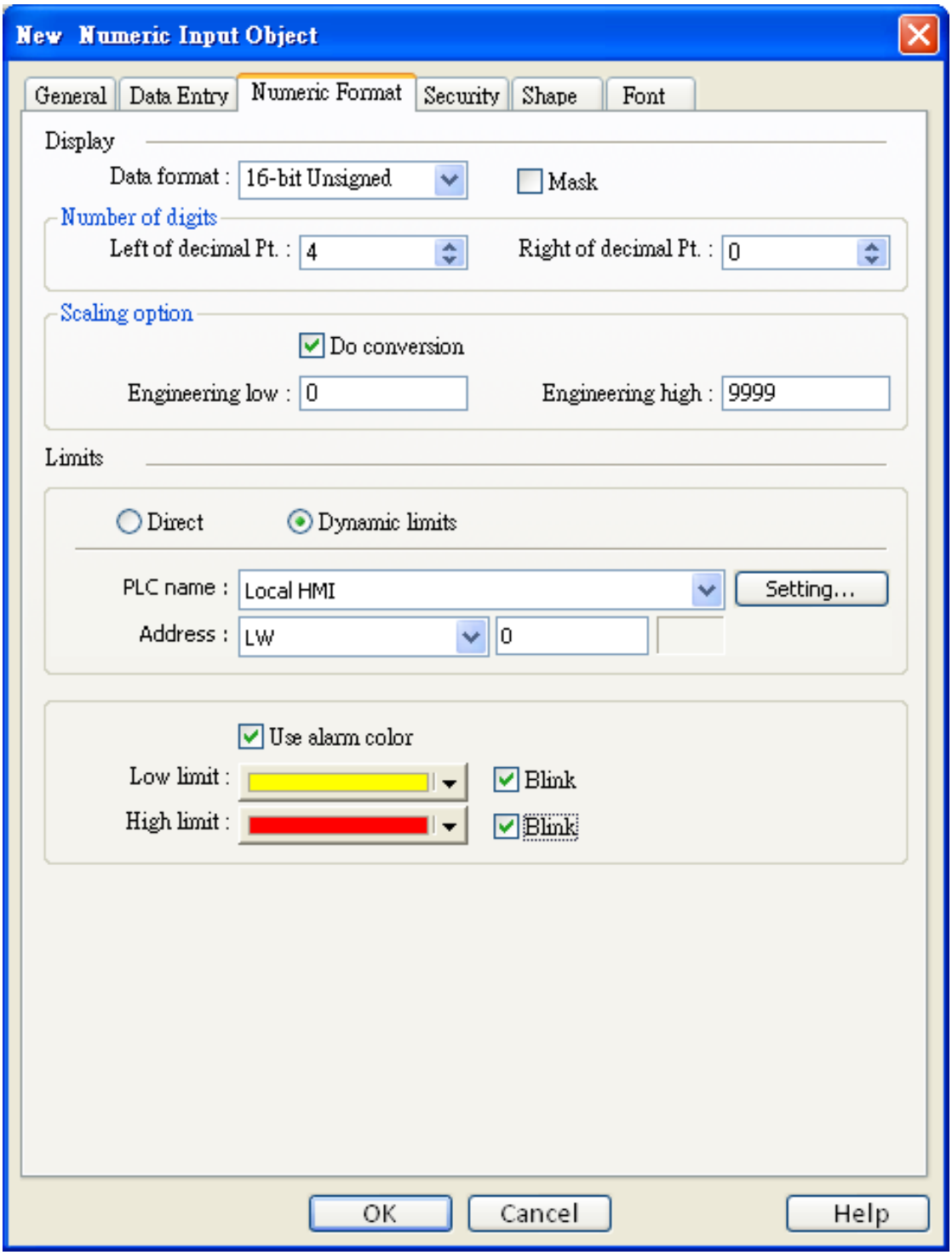
- **Hide title bar – Спрятать полосу заголовка**
  Клавиатура без заголовка может быть выбрана для объектов *Numeric Input / ASCII Input*.

- **Restart the keypad if input value is out of range – Повторение ввода при выходе значения за допустимый диапазон**
  Для объектов типа *Input Value*, запрос на повторение ввода может быть автоматически появляться при возникновении ошибки.

Important

Панель будет дезактивировать всплывающую клавиатуру, когда выбран режим - *Bit Control*.

На рисунке ниже показана Закладка **[Numeric Format],** имеющаяся у объектов: *Numeric Input Object и Numeric Display Object*. Предназначена она для настройки формата отображения.



Display

[Data format]

Выбор формата отображения данных по адресу, назначенному в [Read address]. Список вариантов форматов:

16-bit BCD
32-bit BCD
16-bit Hex
32-bit Hex
16-bit Binary
32-bit Binary
16-bit Unsigned
16-bit Signed
32-bit Unigned
32-bit Signed
32-bit Float

[Mask]

Цифры будут отображаться символом «*», функция цветового предупреждения отключена.

Number of Digits – Число цифр

[Left of decimal Pt]
Число знаков в целой части, до десятичной запятой (точки).

[Right of decimal Pt]
Число знаков после десятичной запятой.

Scaling Option - Масштабирование

[Do conversion] – Делать преобразование

Отображаемые данные – это обработки «сырого» значения по адресу [Read address].

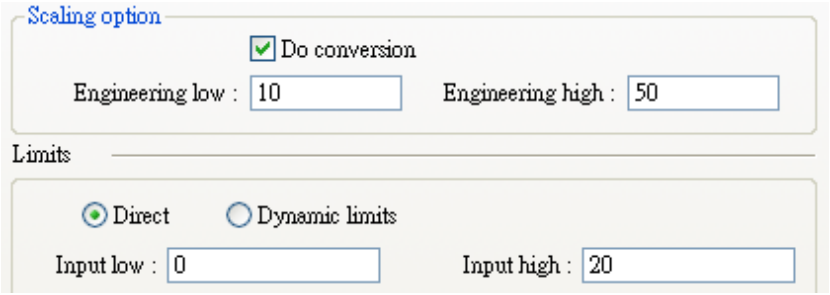
Когда выбрана данная функция, необходимо задать значения [Engineering low], [Engineering high] (нижний и верхний пределы физической величины), [Input low] и [Input high] (нижний и верхний пределы значения по адресу)в области [Limitation].

Предположим, A — «сырое» значение, B — отображаемое значение, тогда формула для преобразования будет:

$$B = [\text{Engineering low}] + (A - [\text{Input low}]) \times \text{ratio}$$

где ratio = ([Engineering high] - [Engineering low]) / ([Input high] - [Input low])

См. пример ниже, «сырое» значение 15, после перевода по формуле 10 + (15 – 0) × (50 – 10) / (20 - 0) = 40, отображается результат 40.

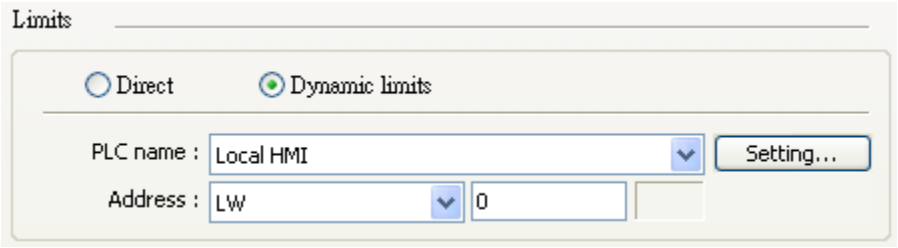


Limits Установка предельных значений входных данных и предупреждающего цвета.

[Direct] – Непосредственные пределы

Нижний и верхний пределы входных данных задаются в полях [Input low] и [Input high] соответственно. Вводимые данные, выходящие за данные пределы, будут проигнорированы.

[Dynamic limits] – Оперативно изменяемые пределы



Эта функция позволяет брать значения нижнего и верхнего пределов входных данных из указанного регистра. Длина данных указанного регистра такая же, как у объекта ввода. В примере (на рисунке вверху) нижний и верхний пределы берутся по адресу [LW100], ниже объяснено, как они размещаются по этому адресу

Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** указанного регистра**.**

Пользователи могут назначать адрес, также, в закладке *Numeric Format*.

Адрес	Нижний предел ввода	Верхний предел ввода
16-битовый формат	LW100	LW101 (Адрес + 1)
32- битовый формат	LW100	LW102 (Адрес + 2)

[Low limit]

Когда значение в указанном регистре ПЛК меньше [Low limit], оно отображается с заранее указанным цветом.

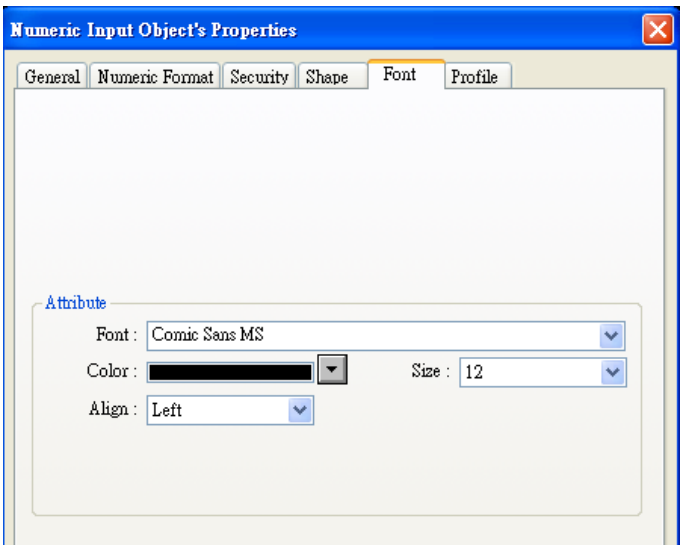
[High limit]

Когда значение в указанном регистре ПЛК [High limit], оно отображается с заранее указанным цветом.

[Blink]

Когда значение в указанном регистре ПЛК меньше [Low limit] или больше [High limit], объект отображается с эффектом мерцания.

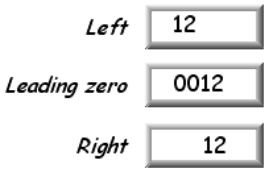
На рисунке ниже показана Закладка [Font], общая для обоих объектов *Numeric input* и *Numeric display* —для установки шрифта, его высоты, цвета и способа выравнивания.



Attribute

[Color]
Когда значение находится в установленных пределах, то оно отображается данным цветом.

[Align]
Есть 3 метода выравнивания: “Left”, “Leading zero”,и “Right”. Рисунок демонстрирует каждый из них.



[Size]
Устанавливает размер шрифта.

13.10 ASCII Input и ASCII Display – Ввод и отображение символов

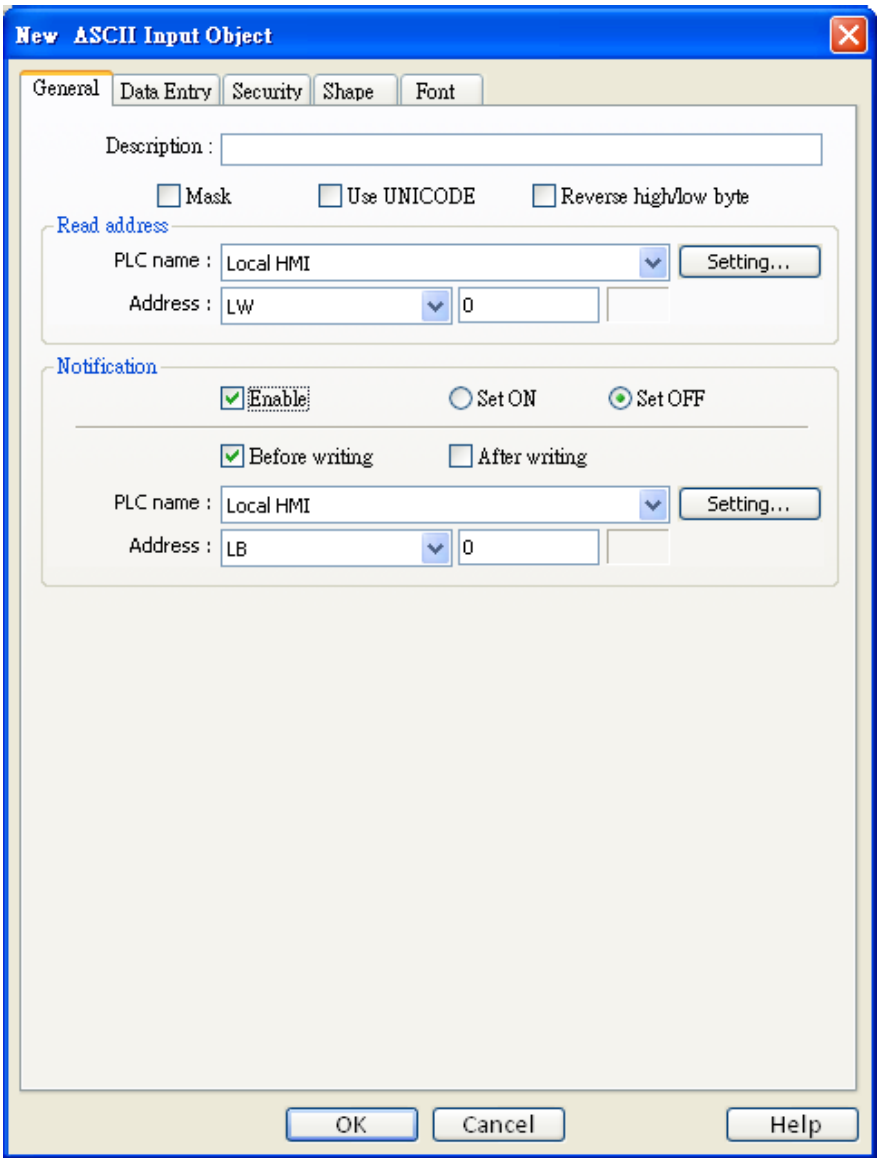
Краткое описание

Оба объекта *ASCII input* и *ASCII display* могут отображать значение указанного адреса в кодировке ASCII. *ASCII input* кроме того, может быть использован для ввода данных с клавиатуры и изменения значения по назначенному адресу.

Настройка свойств

Щелкните на пиктограмме [ASCII Input] или [ASCII Display] на панели инструментов и откроется окно [New ASCII Input Object] или [New ASCII Display Object], заполните их поля, нажмите OK — будет создан новый объект.

Отличаются окна настройки данных объектов тем, что в окне [New ASCII Input Object] имеется области настройки [Notification] и ввода с клавиатуры. Ниже показан вид вкладки [General] диалогового окна [New ASCIIInput Object].



[Mask]

При отображении данных, символ «*» будет использован для замены текста.

[Use UNICODE]

Щелкните [Use UNICODE] для отображения данных в кодировке UNICODE. Иначе данные будут отображаться в кодировке ASCII. Эта функция может использоваться вместе с объектом *Function key* [UNICODE].
Не для всех символов данной кодировки есть соответствующий шрифт, установленный в системе. Шрифт с кодировкой [UNICODE] доступен только для символов UNICODE], вводимых с помощью объекта [Function key].

[Reverse high/low byte]

Обычно ASCII-код отображается в порядке: младший байт – старший байт. Выбор указанного флага меняет порядок отображения на противоположный: старший байт – младший байт.

Read address

Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** слова, значение которого система будет отображать и куда будет записывать новое значение**.**
Пользователи могут назначать адрес в закладке - *General* , при добавлении нового объекта.

[No. of words] – Количество слов
Установка длины данных кодировки ASCII (в словах). Каждый ASCII-символ занимает 1 байт, слово содержит два ASCII-символа.

В примере ниже объект отображает 3 * 2 = 6 символов.

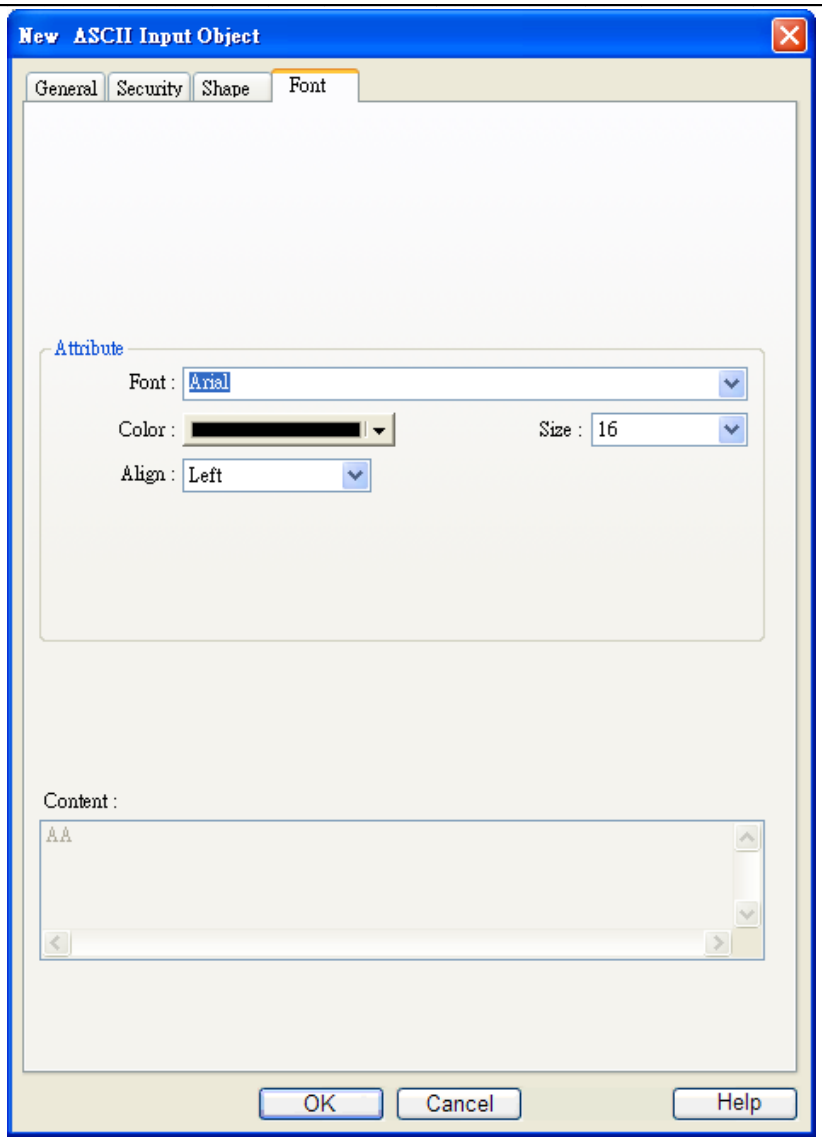
Notification - Уведомление

Когда данная функция выбрана, при успешном изменении значения указанного адреса состояние назначенного бита будет устанавливаться в [ON] или [OFF].

Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** бита уведомления **.**
Пользователи могут назначать адрес в области *Notification.*

[Before writing] / [After writing]

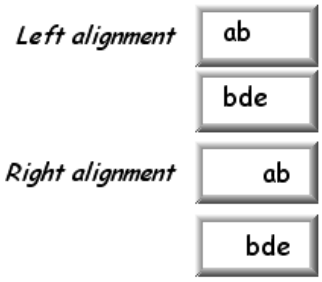
Установка состояния бита «перед / после обновления» содержимого адреса.



Attribute

На рисунке показана Закладка [Font] окна настройки объектов *ASCII input* и *ASCII display*. Здесь можно задать шрифт, его размер, цвет и способ выравнивания.

[Align]
Есть два метода выравнивания: “Left” and “Right”. Рисунок ниже показывает их применение.



[Size]
Установка размера шрифта.

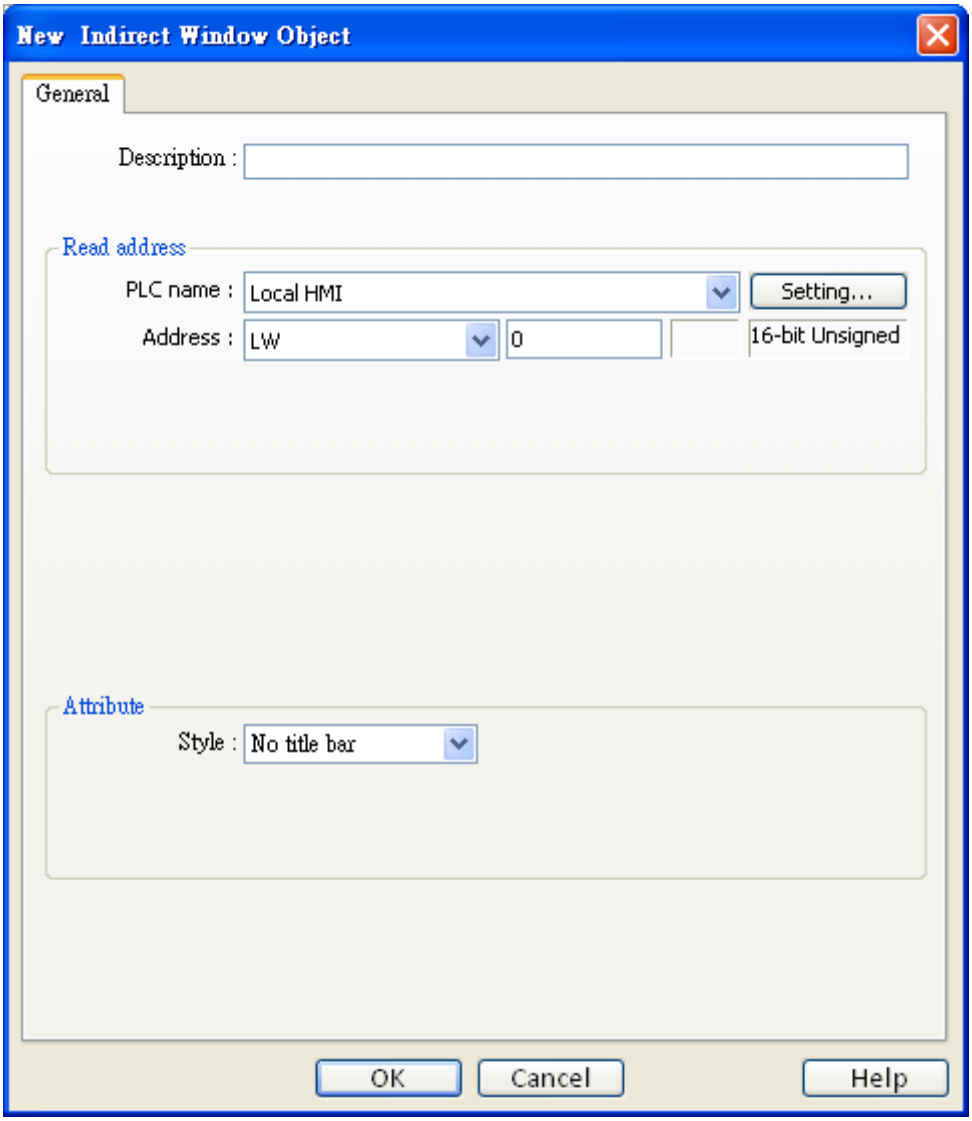
13.11 Объект [Indirect Window]

Краткое описание

Объект [Indirect Window] предназначен для настройки рабочего окна (местоположения, размеров) и адреса. Когда значение по данному адресу содержит допустимый номер окна, окно появится в заданной области. Окно будет закрыто, когда значение по данному адресу станет равно 0. Система активизирует окно только при изменении значения по указанному адресу (0 → допустимый номер окна (ненулевой) → 0, A → B допустимый номер окна).

Настройка свойств

Щелкните пиктограмму [indirect window] на панели инструментов — появится окно [New Indirect Window Object], заполните поля, нажмите OK — будет создан новый объект.



Read address

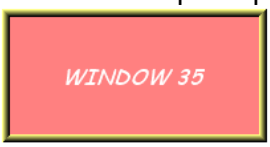
Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** адреса слова, управляющего данным окном.
Пользователи могут назначить адрес в закладке -*General* , при создании нового объекта.

Attribute

[Style]

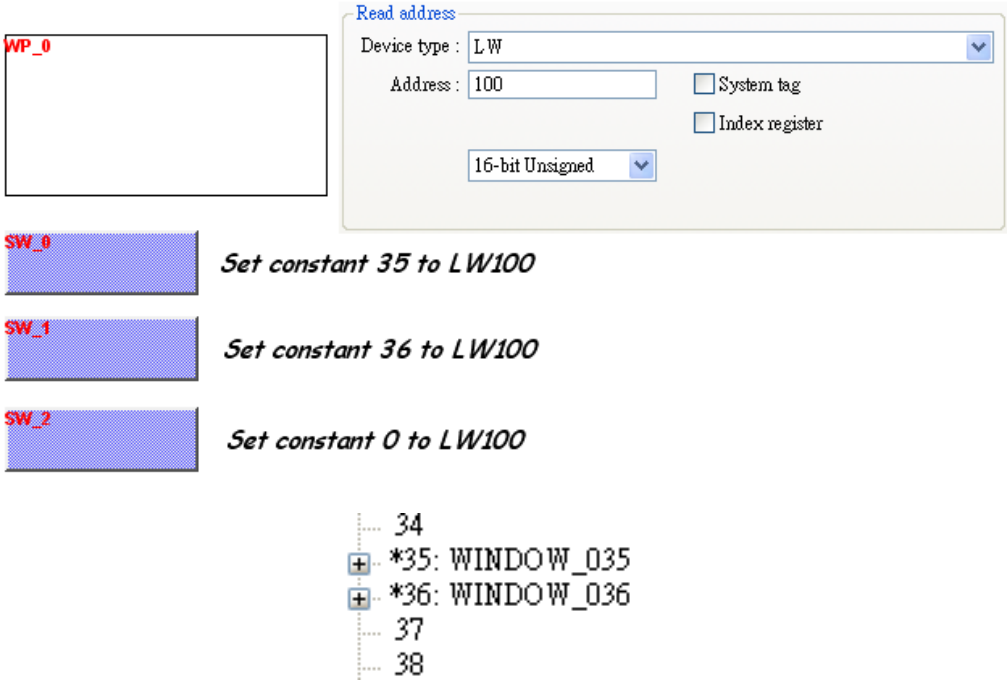
Установка стиля отображения рабочего окна. Есть два варианта: “No title bar” (без полосы заголовка) и (с полосой заголовка) “with title bar”.

a. “No title bar”
Окно не имеет полосы заголовка и его положение фиксированное, заданное в настройках.

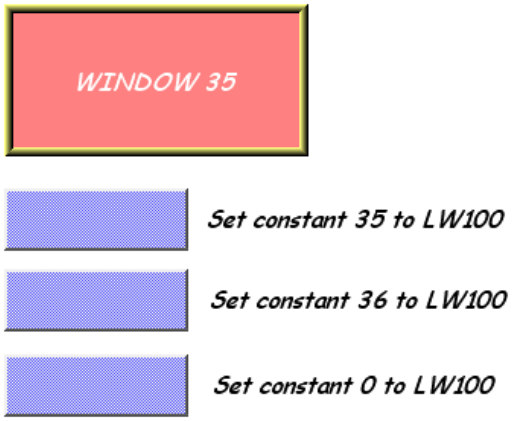


b. “With title bar”
Рабочее окно имеет полосу для перемещения – его положение может быть изменено перетаскиванием.

Пример использования *Indirect Window*
Ниже приведен простой пример, показывающий использование объекта [indirect window]. На рисунках показано, как настраивать этот объект и использовать адрес [LW100] для изменения рабочего окна.



Используя объект [Set Word] SW_0, установите значение 35 по адресу [LW100] — появится рабочее окно 35.



Используя объект [Set Word] SW_1, установите значение 36 по адресу [LW100] — появится рабочее окно 36.

Независимо от того, какое окно открыто, нажмите кнопку SW_2 — будет установлено значение 0 по адресу [LW100] и окно закроется.
Другой способ закрытия рабочего окна: создать объект [function key] с функцией [close window]. При нажатии функциональной кнопки, всплывающее окно закроется.

Примечание: Максимально 16 окон могут быть отображены одновременно в рабочем режиме. Не используйте эту функцию для открытия окна, когда то же самое окно было открыто объектами [function key] или [direct window].

13.12 Объект [Direct Window]

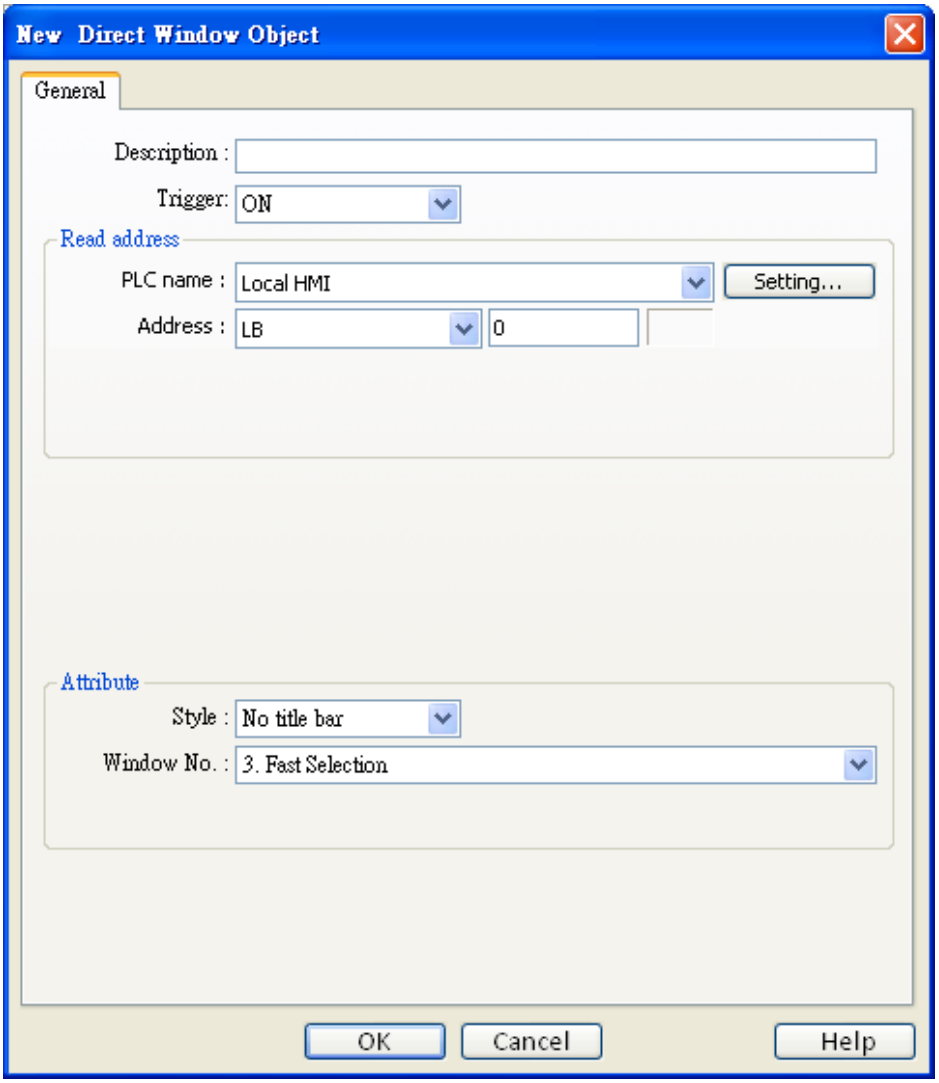
Краткое описание

Объект [Direct window] служит для задания рабочего окна (положение, размеры), бита состояния и номера окна. Когда бит состояния имеет значение ON/OFF, рабочее окно появится в заданной области экрана. Окно закроется при сбросе этого бита. Указанные события происходят только при смене состояния бита (OFF → ON, ON → OFF).

Объект [Direct Window] отличается от [Indirect Window] тем, что в его настройках задается рабочее окно. При работе системы можно использовать состояние указанного регистра для управления открытием/закрытием окна.

Настройка свойств

Щелкните пиктограмму [Direct Window] на панели инструментов — появится окно [New Direct Window Object], заполните его поля, нажмите OK — будет создан новый объект.



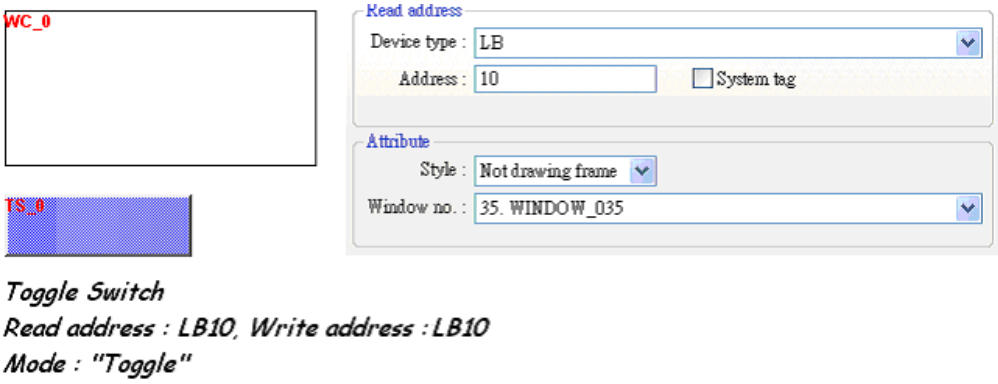
Read address
Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** адреса слова, управляющего данным окном.
Пользователи могут назначить адрес в закладке -*General* , при создании нового объекта.

Attribute

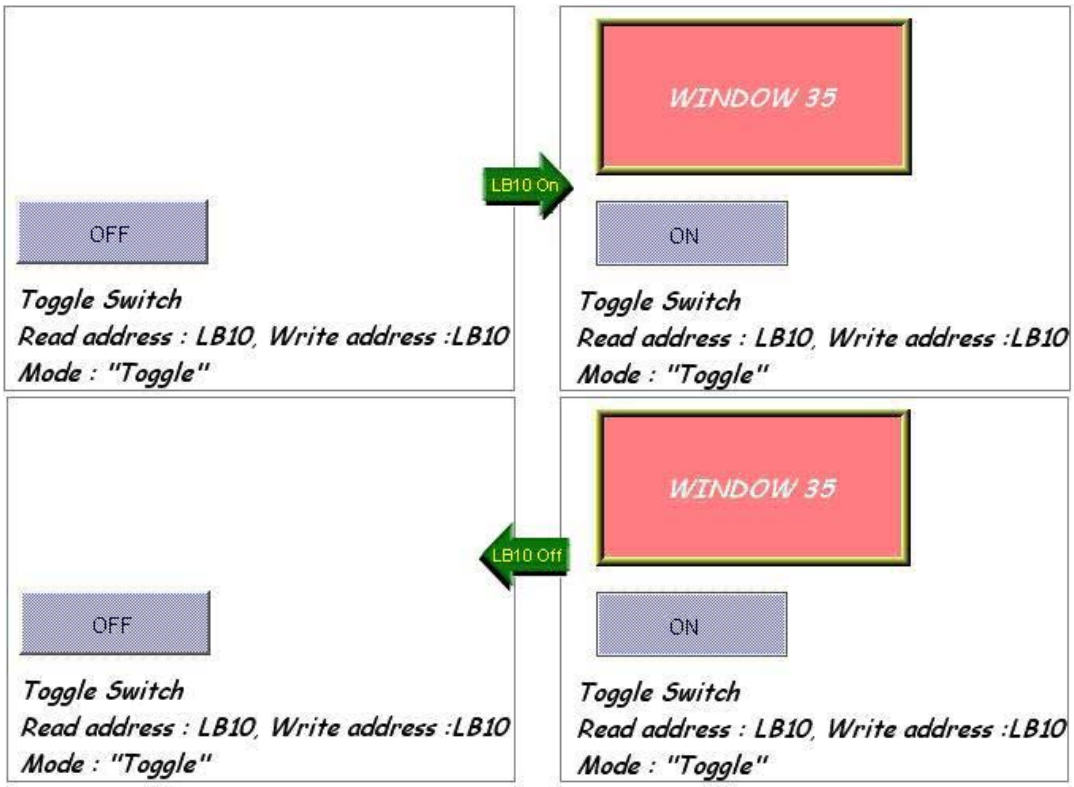
[Style] См. раздел «Indirect Window».

[Window no.]
Назначьте номер всплывающего окна.

Пример использования окна - *direct window*
Ниже приведен пример использования объекта [direct window]. На рисунках показаны настройки объекта. В примере используется бит [LB10] для вызова окна 35.



Когда значение [LB10] равно ON, появляется окно 35; когда OFF — закрывается. См. рисунки ниже.



Примечание: Максимально 16 окон могут быть отображены одновременно в рабочем режиме. Не используйте эту функцию для открытия окна, когда то же самое окно было открыто объектами [function key] или [direct window].

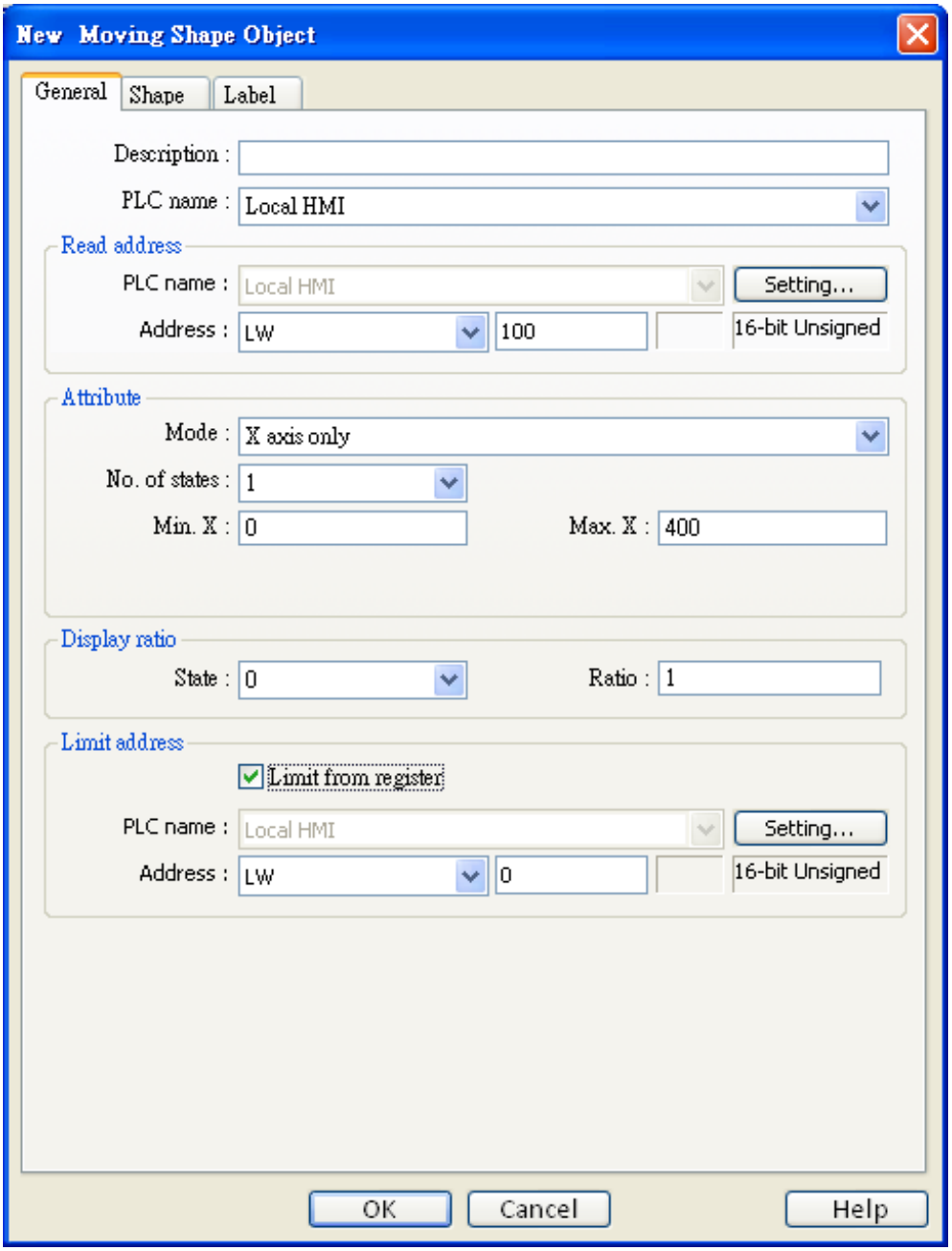
13.13 Moving Shape – Перемещаемая форма

Краткое описание

Объект [Moving Shape] используется для определения состояния объекта и расстояния перемещения. Данный объект нужен для размещения объекта в окне, на месте определенном ПЛК. Состояние и абсолютные координаты объекта в окне зависят от текущих значений трех регистров ПЛК. Обычно первый регистр управляет состоянием объекта, второй – управляет горизонтальным положением объекта (X), третий — вертикальным (Y).

Настройка свойств

Щелкните пиктограмму [Moving Shape] на панели инструментов — откроется окно [New Moving Shape Object], заполните его поля, нажмите OK — будет создан новый объект.



Read address

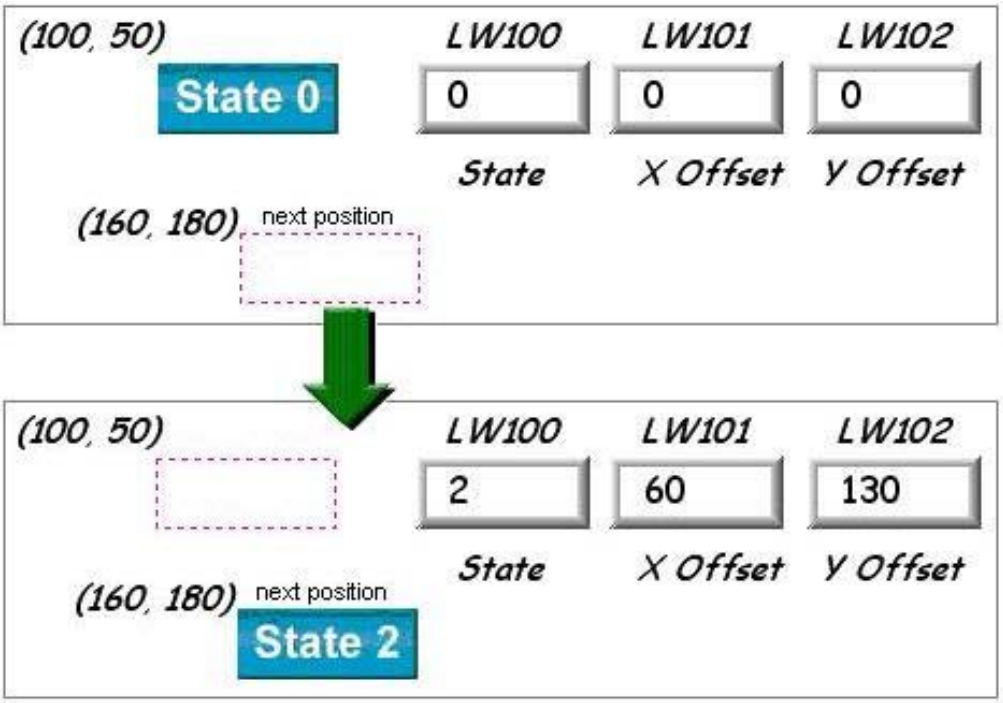
Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** адреса слов, управляющих состоянием объекта и его перемещением.
Пользователи могут назначить адрес в закладке -*General* , при создании нового объекта.

В таблице ниже приведены данные адреса для разных используемых форматов данных.

Формат данных	Адрес управления состоянием объекта	Адрес управления перемещением по оси X	Адрес управления перемещением по оси Y
16-bit	Address	Address + 1	Address + 2
32-bit	Address	Address + 2	Address + 4

Например, если [read address] задан как [LW100] и формат данных выбран “16-bit Unsigned”, то [LW100] используется для управления состоянием объекта, [LW101] для управления расстоянием перемещения вдоль оси X, [LW102] — вдоль оси Y.

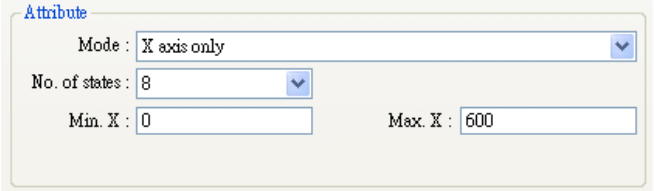
Рисунок ниже демонстрирует, что [read address] задан [LW100], начальная позиция объекта была (100, 50). Предположим, нужно, чтобы объект переместился в точку с координатами (160, 180) и индицировал состояние 2 (state 2). Тогда значение [LW100] нужно сделать равным 2, [LW101]=160-100=60; [LW102]=180-50=130.



Attribute Выбор способа перемещения объекта и диапазона.

a. X axis only – только по оси X

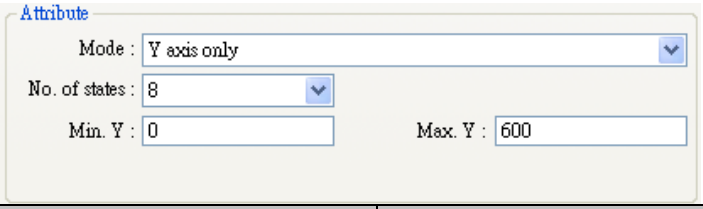
Объект может перемещаться только вдоль оси X. Диапазон перемещения задается границами [Min. X] и [Max. X].



Формат данных	Адрес управления состоянием объекта	Адрес управления перемещенем по оси X
16-bit	Address	Address + 1
32-bit	Address	Address + 2

b. Y axis only– только по оси Y

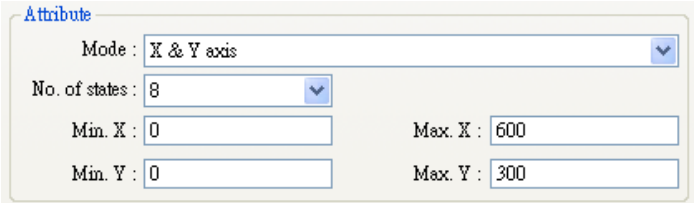
Объект может перемещаться только вдоль оси Y. Диапазон перемещения задается границами [Min. Y] и [Max. Y].



Формат данных	Адрес управления состоянием объекта	Адрес управления перемещением по оси Y
16-bit	Address	Address + 2
32-bit	Address	Address + 4

c. X & Y axis – по осям X и Y

Объект может перемещаться вдоль обеих осей. Расстояние перемещения ограничивается границами [Min. X], [Max. X] и [Min. Y], [Max. Y].

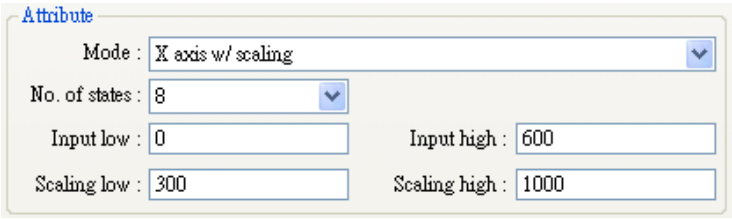


Формат данных	Адрес управления состоянием объекта	Адрес управления перемещением по оси X	Адрес управления перемещением по осиY
16-bit	Address	Address + 1	Address + 2
32-bit	Address	Address + 2	Address + 4

d. X axis w/ scaling

Перемещение вдоль оси X с масштабированием расстояния. Предполагается, что значение указанного регистра равного DATA, будет применяться в следующей формуле для вычисления расстояния перемещения вдоль оси X.

X =(DATA – [Input low]) * ([Scaling high – Scaling low]) / ([Input high] – [inputlow])



Например, объект может перемещаться только между границами 0 и 600, а диапазон значений числа по указанному адресу: от 300 до 1000; нужно установить [Input low] в 0, [Input high] в 600, а [Scaling low] в 300 и [Scaling high] в 1000 и объект будет перемещаться в заданных границах.

Формат данных	Адрес управления состоянием объекта	Адрес управления перемещением по оси X
16-bit	Address	Address + 2
32-bit	Address	Address + 4

e. Y axis w/ scaling

Аналогичная функция для перемещения вдоль оси Y, формула для вычисления расстояния такая же как и в “X axis w/ scaling”.

Формат данных	Адрес управления состоянием объекта	Адрес управления перемещением по оси Y
16-bit	Address	Address + 2
32-bit	Address	Address + 4

f. X axis w/ reverse scaling

Данная функция аналогична функции “X axis w/ scaling”, но направление перемещения противоположное.

g. Y axis w/ reverse scaling

Данная функция аналогична функции “Y axis w/ scaling”, но направление перемещения противоположное.

Display ratio – Отображать пропорционально

Для различных состояний может быть отдельно настроен размер объекта, как показано на рисунке ниже.

Limit Address – адреса ограничений

Диапазон перемещений объекта может быть задан не только границами [Min. X], [Max. X] и [Min. Y] [Max. Y], но также и числами с назначенными адресами. Предположим, диапазон перемещений задан вторым способом регистром с адресом [Address], тогда адреса границ можно определить в соответствии с таблицей.

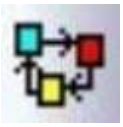
Формат данных	Адрес [Min. X]	Адрес [Max. X]	Адрес [Min. Y]	Адрес [Max. Y]
16-bit BCD	Address	Address + 1	Address + 2	Address + 3
32-bit BCD	Address	Address + 2	Address + 4	Address + 6

13.14 Animation - Анимация

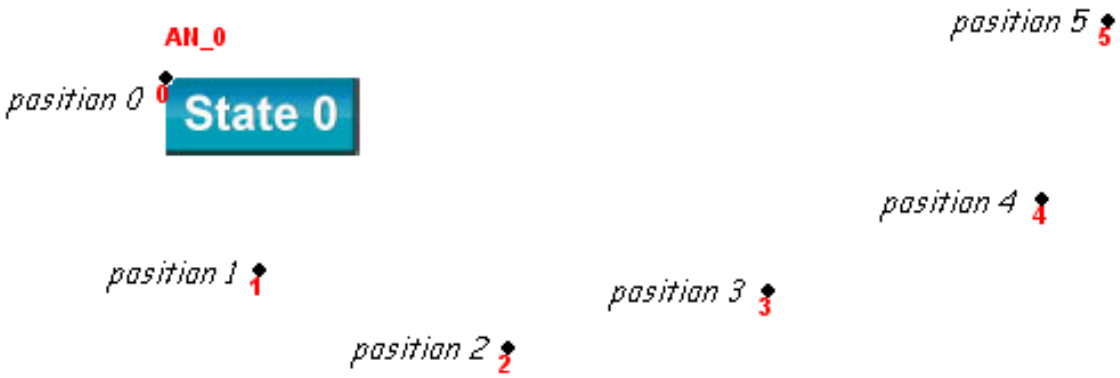
Краткое описание

Объект [Animation] используется для отображения объекта на экране в определенном месте в соответствии с заданной траекторией и параметрами, хранящимися по указанному адресу в ПЛК. Состояние и абсолютные координаты фигуры на экране зависят от текущих значений двух последовательных регистров памяти ПЛК. Обычно первый регистр управляет самим объектом, второй — позицией (положением) вдоль назначенного пути. При изменении значения второго регистра ПЛК, фигура или изображение перемещается в следующую позицию вдоль траектории.

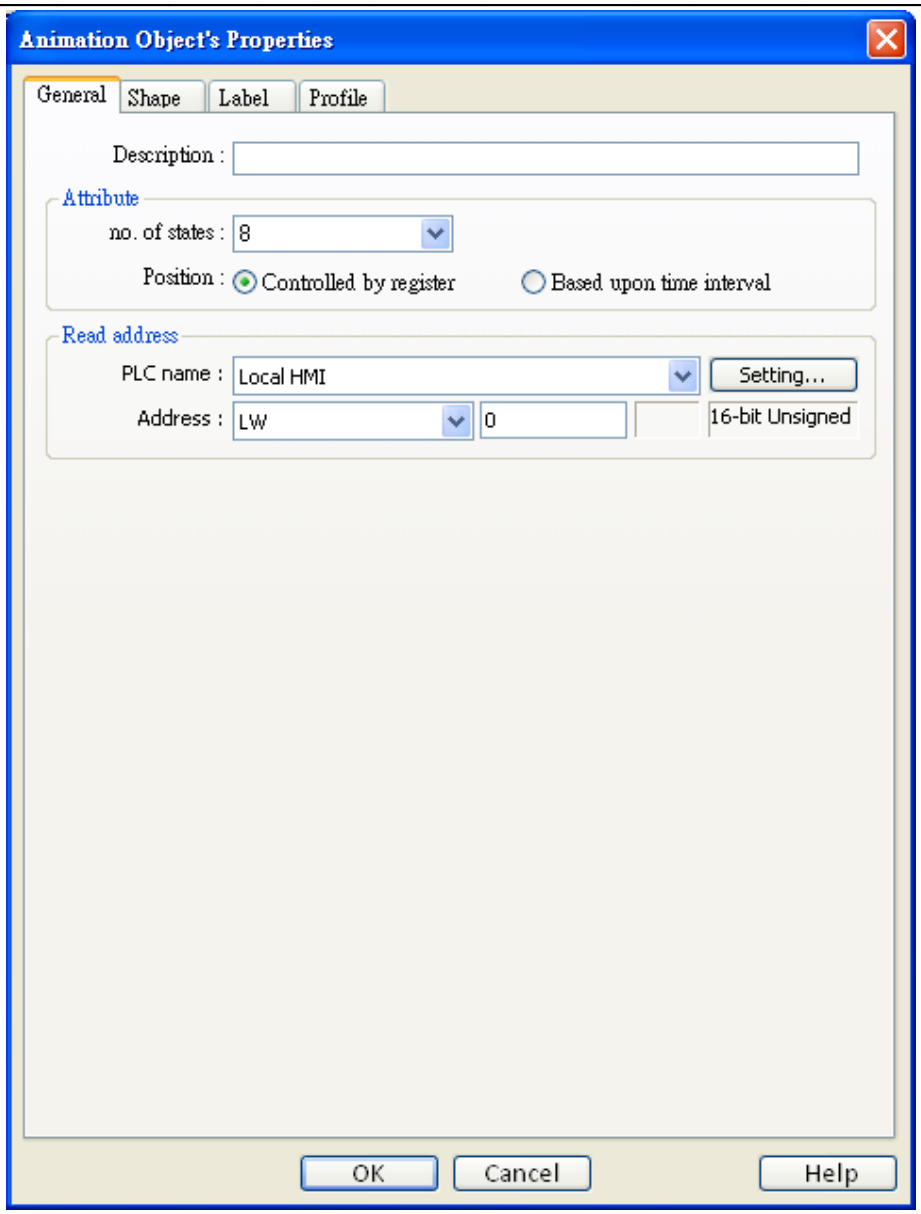
Настройка свойств



Щелкните пиктограмму [Animation] на панели инструментов, перемещая мышь и щелкая левой кнопкой, выберите последовательно все позиции траектории. После определения всех точек траектории щелкните правой кнопкой мыши — будет создан новый объект [Animation]. См. рисунок ниже.



Для изменения свойств объекта дважды щелкните левой кнопкой мыши на объекте — откроется окно [Animation Object’s Properties].



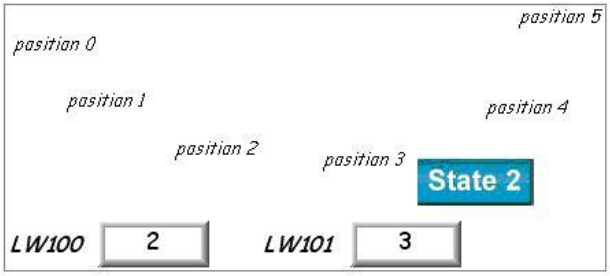
Attribute
[Total no. of States] – Общее число состояний
Задание числа состояний.

[Position] a. Controlled by register
При выборе данного варианта состоянием и положением объекта будет управлять назначенный регистр.

Read address
Если выбран вариант [Controlled by register], то необходимо задать адрес управляющего регистра.
Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** адрес слова.
Пользователи могут назначить адрес в закладке -*General* , при создании нового объекта.
В таблице ниже описано как распределяются адреса, используемые для управления состоянием и позицией в зависимости от формата чисел.

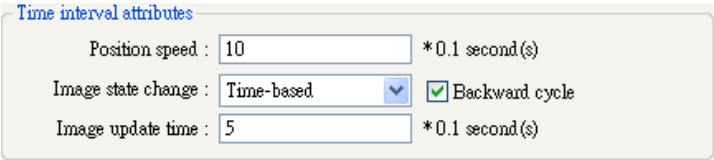
Формат данных	Адрес, управления состоянием	Адрес управления позицией объекта
16-bit BCD	Адрес	Адрес + 1
32-bit BCD	Адрес	Адрес + 2

Например, если назначенный регистр [LW100] и формат данных ”16-bit Unsigned”, то значение адреса [LW100] управляет состоянием объекта, [LW101] — его положением. На рисунке ниже [LW100]=2, [LW101]=3, т.е. состояние объекта — 2, позиция — 3.



[Position] b. Based upon time interval – Основанная на временных интервалах

Если выбран режим “Based upon time interval”, то объект автоматически будет изменять свое состояние и место отображения. Параметр “Time interval attributes” предназначен для задания временного интервала для смены состояний и позиций.



[Position Speed] – Скорость смены позиций

Скорость изменения положения, единица измерения: 0,1 секунды. Предположим, параметр [Speed] установлен равным 10, тогда объект будет изменять положение каждую секунду.

[Backward cycle] – Обратный цикл

Предположим, что объект имеет четыре позиции: position 0, position 1, position 2, и position 3 — и [Backward cycle] не отмечен. В этом случае когда объект достигнет последней точки траектории (position 3), произойдет возврат в исходную точку (position 0) и перемещение вновь повторится. Траектория будет выглядеть следующим образом:

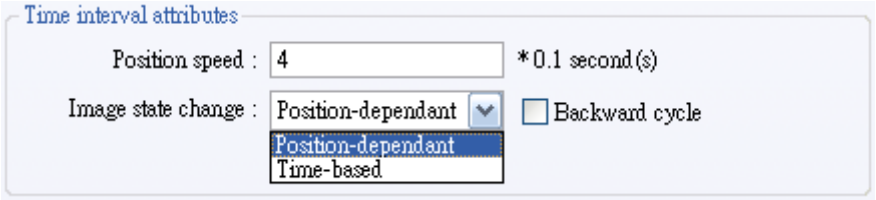
position 0 → position 1 → position 2 → position 3 → position 0 →
position 1 → position 2…

Если флаг [Backward] отмечен, то после того, как объект достигнет последней точки траектории (position 3), он начнет перемещаться обратно в исходную позицию position 0, и данный процесс будет повторяться. Траектория перемещения будет выглядеть следующим образом:

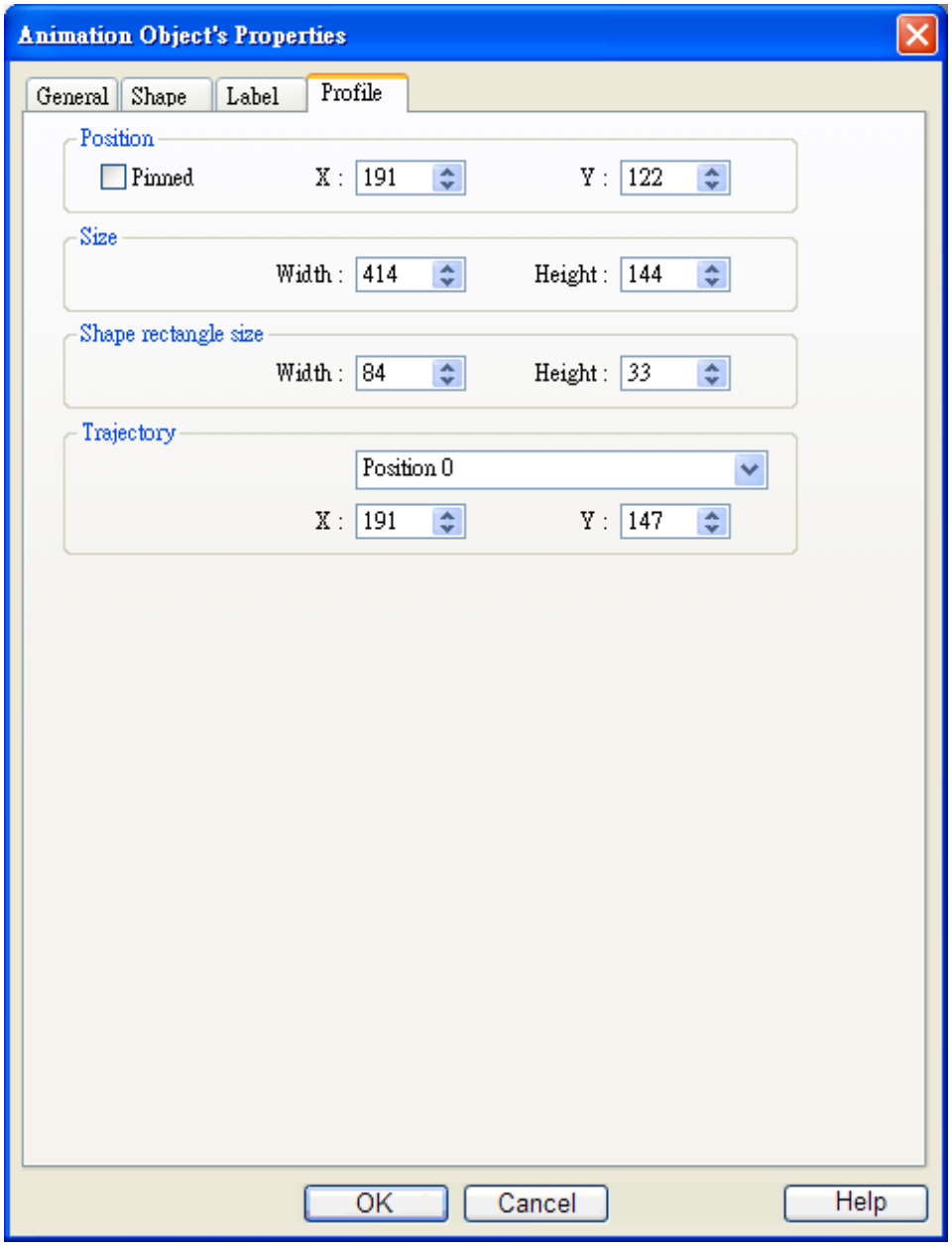
position 0 → position 1 → position 2 → position 3 → position 2 →
position 1 → position 0…

[Image state change]

Режим изменения состояния. Есть две опции: “Position dependant” – зависимый от положения и “Time-based” – зависимый от временных интервалов. Когда выбран режим “Position dependant”, это означает что состояние изменяется в зависимости от положения. Если выбран вариант “Time-based”, то положение будет изменяться со скоростью “Position speed”, а состояние объекта через интервалы времени “Image update time”.



Приведенное ниже диалоговое окно показывает настройку размеров объекта [Animation]. Вызов окна производится двойным щелчком на объекте.



Shape rectangle size

Установить размер фигуры.

Trajectory

Задание положения каждой точки траектории.

13.15 Bar Graph - Гистограмма

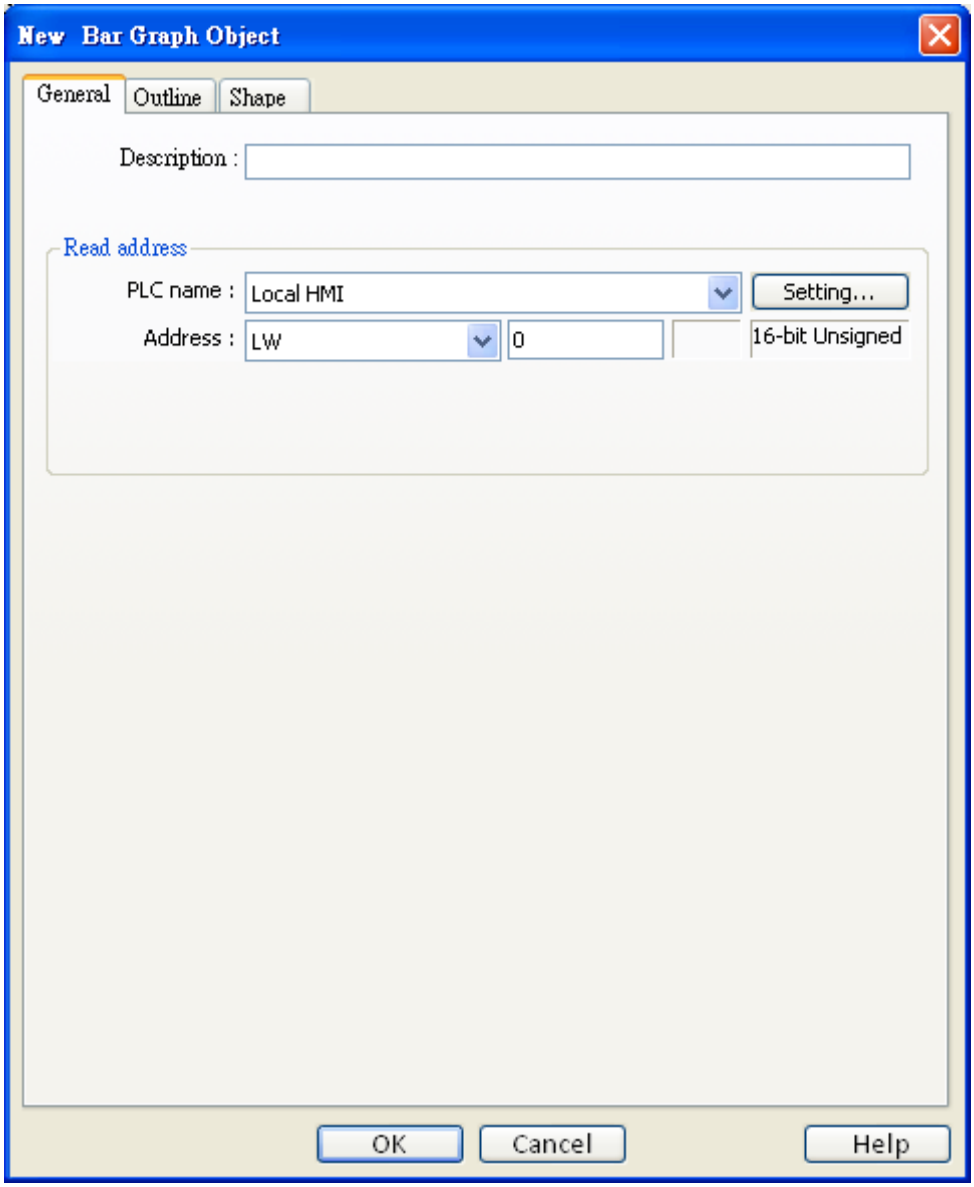
Краткое описание

Объект [Bar Graph] отображает данные регистра в виде столбчатой диаграммы пропорционально значению.

Настройка свойств

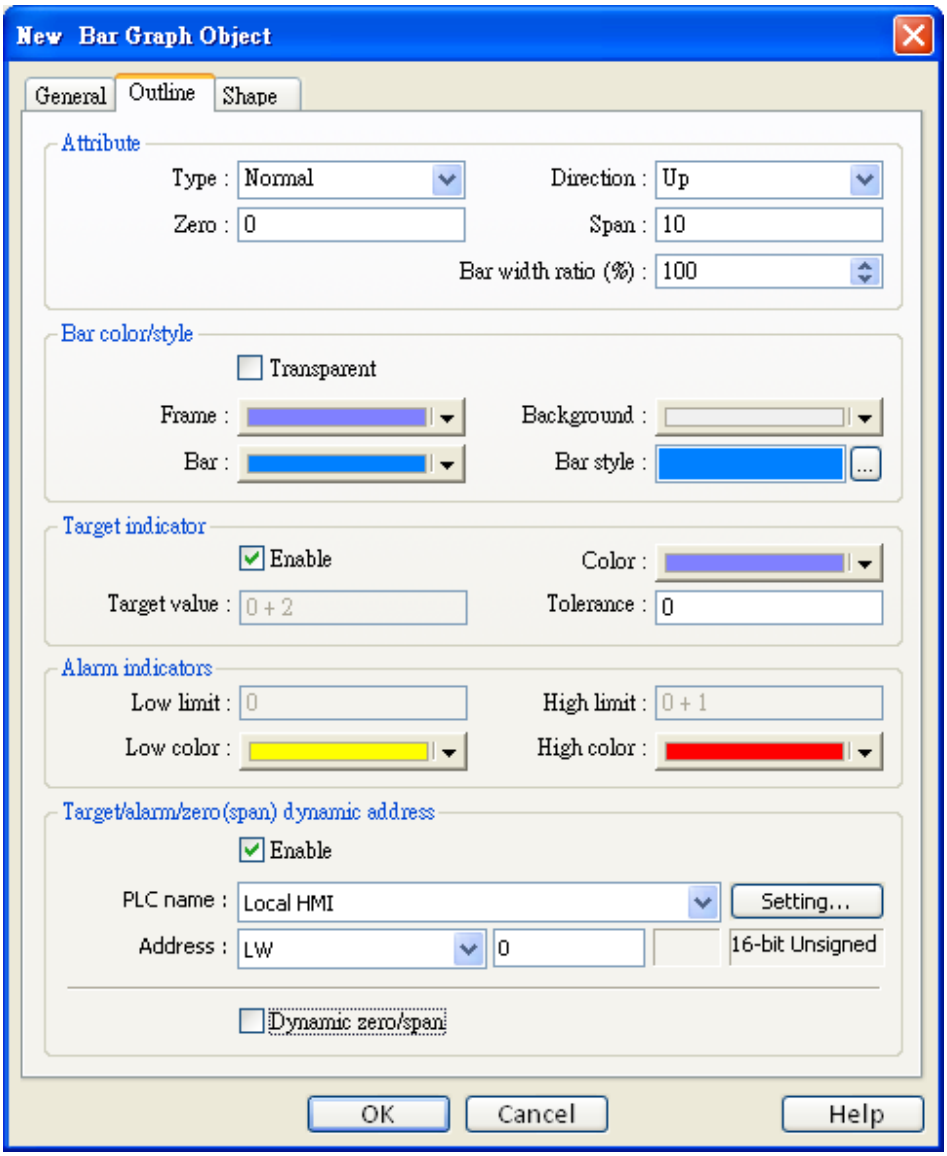
Щелкните пиктограмму [Bar Graph] на панели задач — откроется окно [New Bar Graph Object], заполните все поля, нажмите OK — будет создан новый объект. См. рисунок ниже.

Ниже показана Закладка [General] диалогового окна настройки.

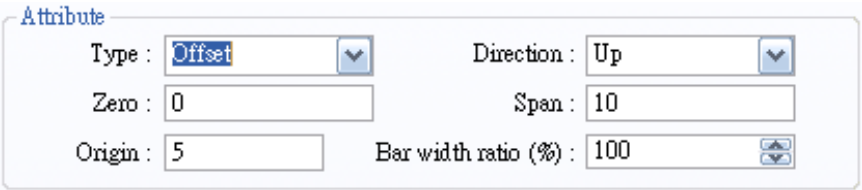


Read address
Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** адрес слова управляющиего барграфом .
Пользователи могут назначить адрес в закладке -*General* , при создании нового объекта.

Ниже показана Закладка [Outline] диалогового окна настройки.



Attribute
[Type] Есть два варианта выбора: “Normal” и “Offset”. При выборе “Offset” необходимо иметь значение, относительно которого идет смещение. См. рисунок ниже.



[Direction] Для выбора направления гистограммы: доступны 4 варианта: вверх, вниз, вправо, влево (”Up”, “Down”, “Right”, “Left”).

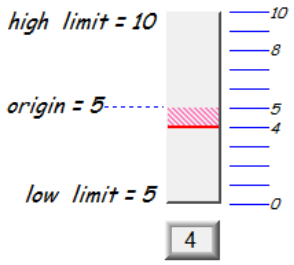
[Zero]、[Span] – Ноль, диапазон

Процент заполнения диаграммы может быть вычислен по следующей формуле:

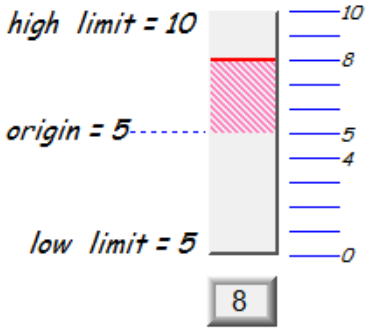
Процент заполнения = (Register value – Zero) / [Span] – [Zero]) * 100%

Когда выбран режим [Offset], и если значение (Register value – Zero) > 0, то прямоугольник диаграммы будет заполняться с ссылочного значения смещения; если же (Register value – Zero) < 0, то секция диаграммы будет заполняться снизу от значения смещения.

Например,
Значения параметров заданы: Origin =5, Span=10, Zero=0 — тогда при различных значениях по адресу [read address], объект будет выглядеть, как показано ниже.
Когда значение [read address] равно 4.

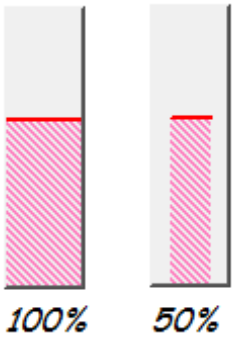


Когда значение [read address] равно 8.



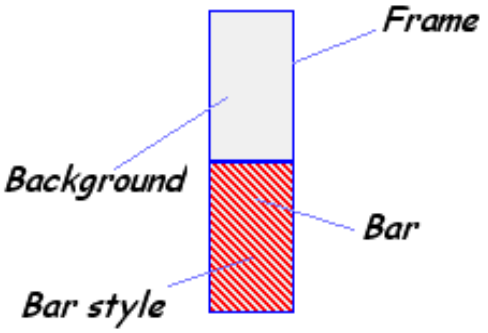
[Bar width ratio(%)] – Соотношение ширины

Для задания соотношения между шириной секции диаграммы и объекта. Например, ниже показан вид при значениях 50% и 100%.



Bar color/style

Установка цвета рамки и фона, стиля и цвета заполнения (Frame, Background color, Bar style, и Bar color) диаграммы. См. рисунок ниже.



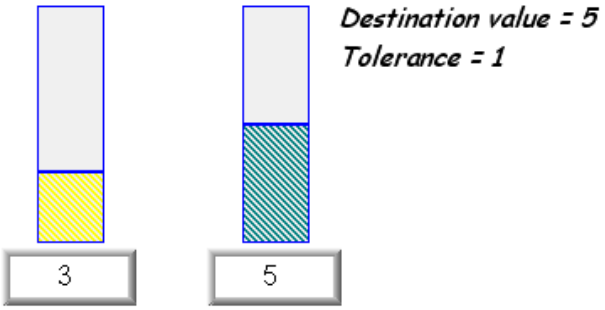
Target Indicator

Когда значение регистра удовлетворяет условию ниже, цвет заполнения изменяется на заданный в поле [Target Color].

Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register**] динамический адрес управляющий цветом. Пользователи могут назначить адрес, также, в закладке - *Outline*.

[Target Value] - [Tolerance] < = Register value < = [Target Value] +[Tolerance]

См. рисунок ниже: в данном случае [Target Value]=5, [Tolerance]=1, если значение регистра больше или равно 5-1=4 или меньше или равно 5+1=6, то цвет заполнения секции диаграммы изменится на заданный.

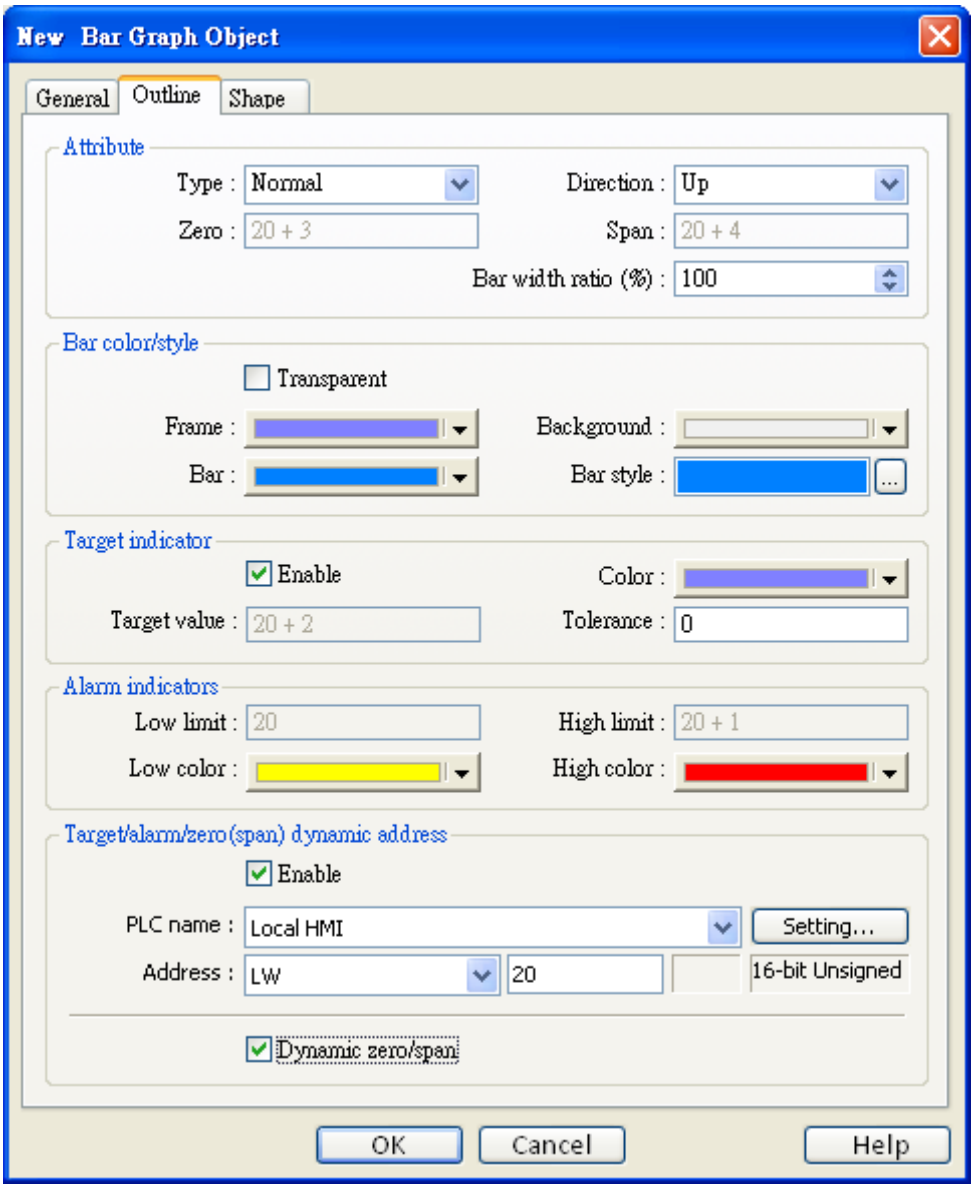


Alarm Indicator

Когда значение по адресу превосходит верхний предел [High limit], цвет заполнения изменится на [High color], когда значение меньше нижнего предела [Low limit], то цвет заполнения изменится на [Low color].

Target/Alarm Dynamic Address

Если отмечен флаг [Enable] такие параметры, как [Low limit] и [High limit] из области [Limitation alarm] и [Target value] из [Target indicator] находятся по заданным адресам. См. рисунок далее.



В таблице ниже показаны адреса указанных параметров. [Address] означает начальный адрес.

Например, если [read address] задан [LW20] и данные являются 16-разрядными, то [Low limit] размещается в [LW20], [High limit] — по адресу [LW21], [Target value] — по адресу [LW22], [Zero] — в [LW23], [Span] — по адресу [LW24].

Формат данных	Адрес [Low limit]	Адрес [High limit]	Адрес [Target value]	Адрес [Zero]	Адрес [Span]
16-bit	Address	Address +1	Address +2	Address +3	Address +4
32-bit	Address	Address +2	Address +4	Address +6	Address +8

13.16 Meter Display – Показывающий прибор

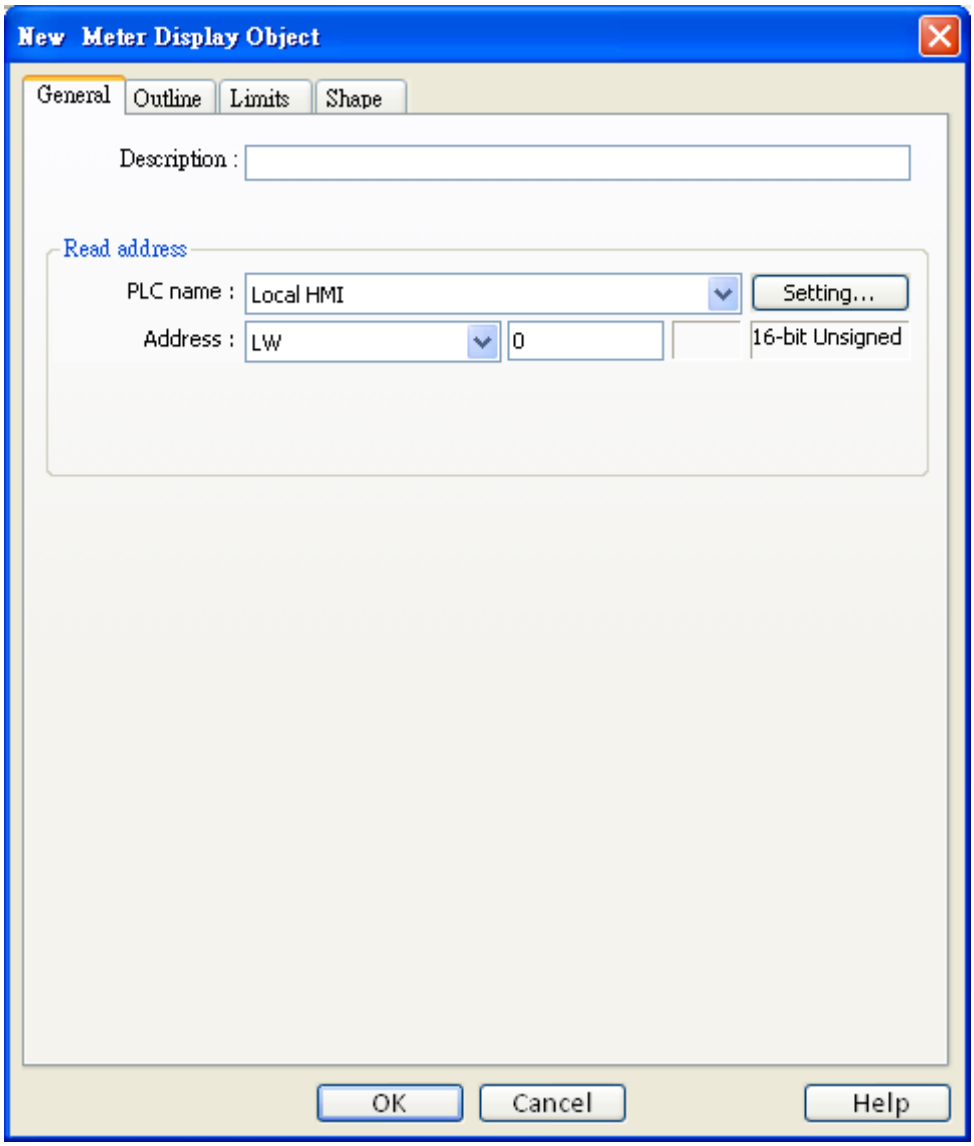
Краткое описание

Объект [Meter Display] может выводить на экран значение в виде индикатора.

Настройка свойств

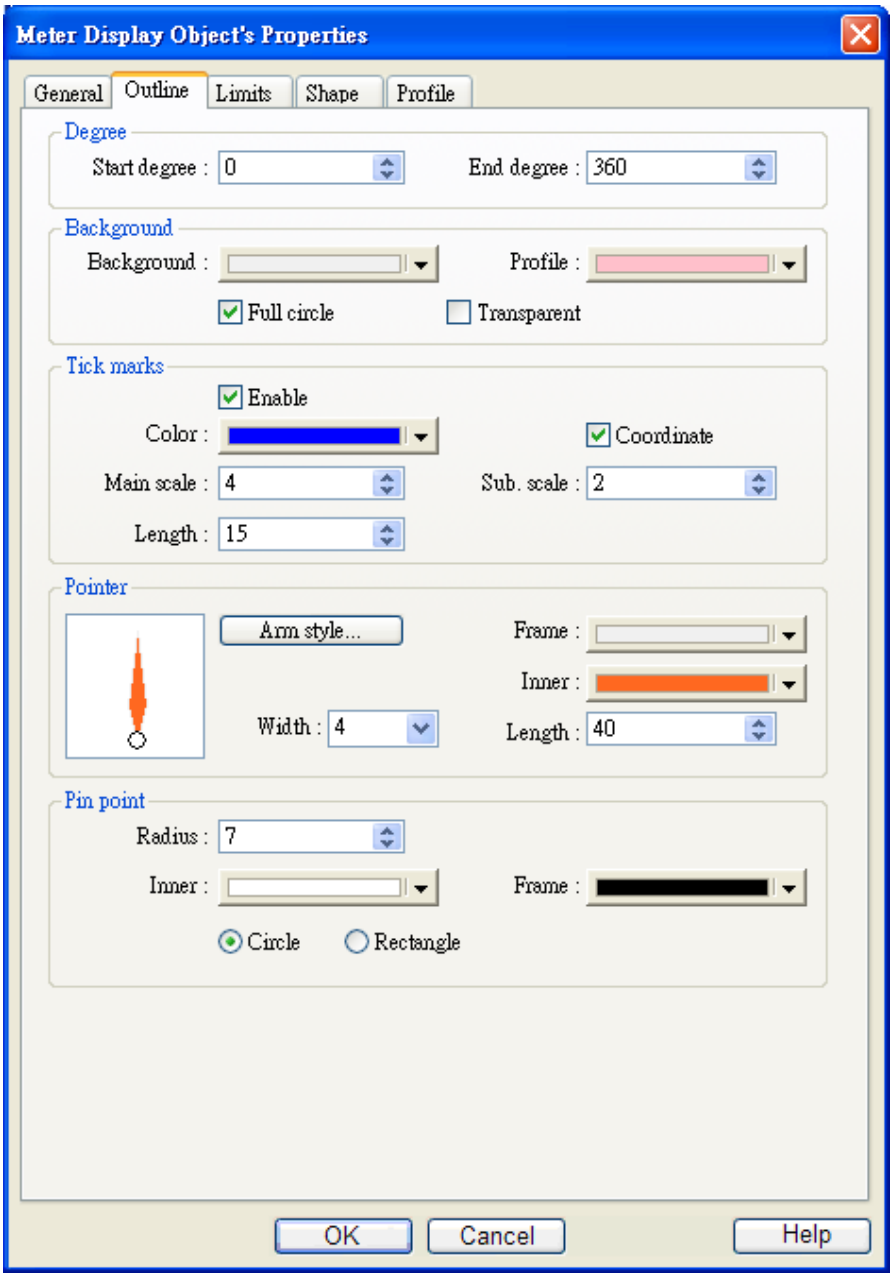
Щелкните пиктограмму [Meter Display] на панели инструментов — откроется окно [Meter Display Object's Properties], заполните все поля, нажмите OK — будет создан новый объект. См. рисунок ниже.

На рисунке ниже показан вид вкладки [General] окна настройки [Meter Display Object's Properties].

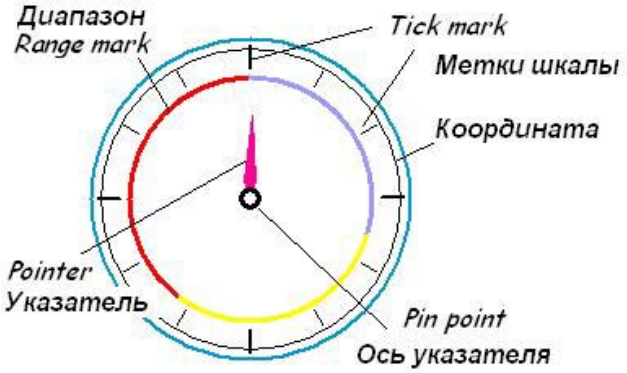


Read address

Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** динамический адрес слова, которое управляет отображаемым значением объекта [Meter Display].
Пользователи могут назначить адрес, также, в закладке – *General* окна “*Meter Display Object’s Properties”* .

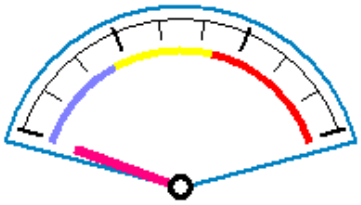


В окне, показанном выше, можно настроить вид объекта. См. рисунок далее для пояснения названия каждой части объекта [Meter Display].

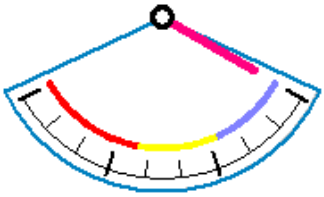


Degree - Градусы

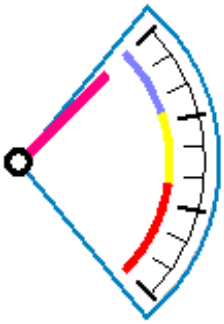
Установите параметры “start degree” (начальный угол) и “end degree” (конечный угол), углы сектора можно задавать из диапазона от 0 до 360°. На рисунке ниже показаны результаты различных настроек.



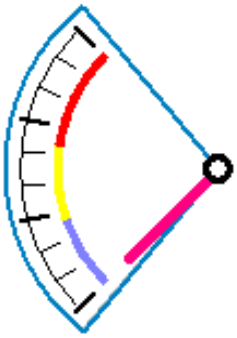
[Start degree] = 290, [End degree] = 70



[Start degree] = 45, [End degree] = 240



[Start degree] = 120, [End degree] = 135



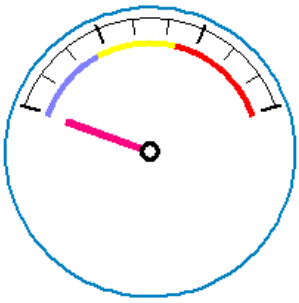
[Start degree] = 225, [End degree] = 315

Background

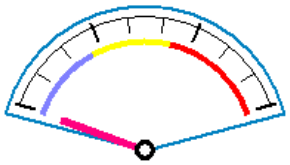
Установка цвета фона и контура объекта.

[Full circle]

Когда отмечен флаг [Full circle] – Полный круг, контур объекта отображается полностью, если не отмечен – то в только в пределах заданных углов сектора. См. рисунки ниже.



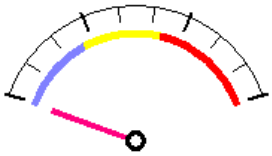
Full circle



non-full circle

[Transparent]

Когда выбран флаг [Transparent], не будет показываться фон и контур объекта. См. рисунок ниже.



Transparent

Tick marks

Настройка числа и цвета меток (засечек) шкалы.

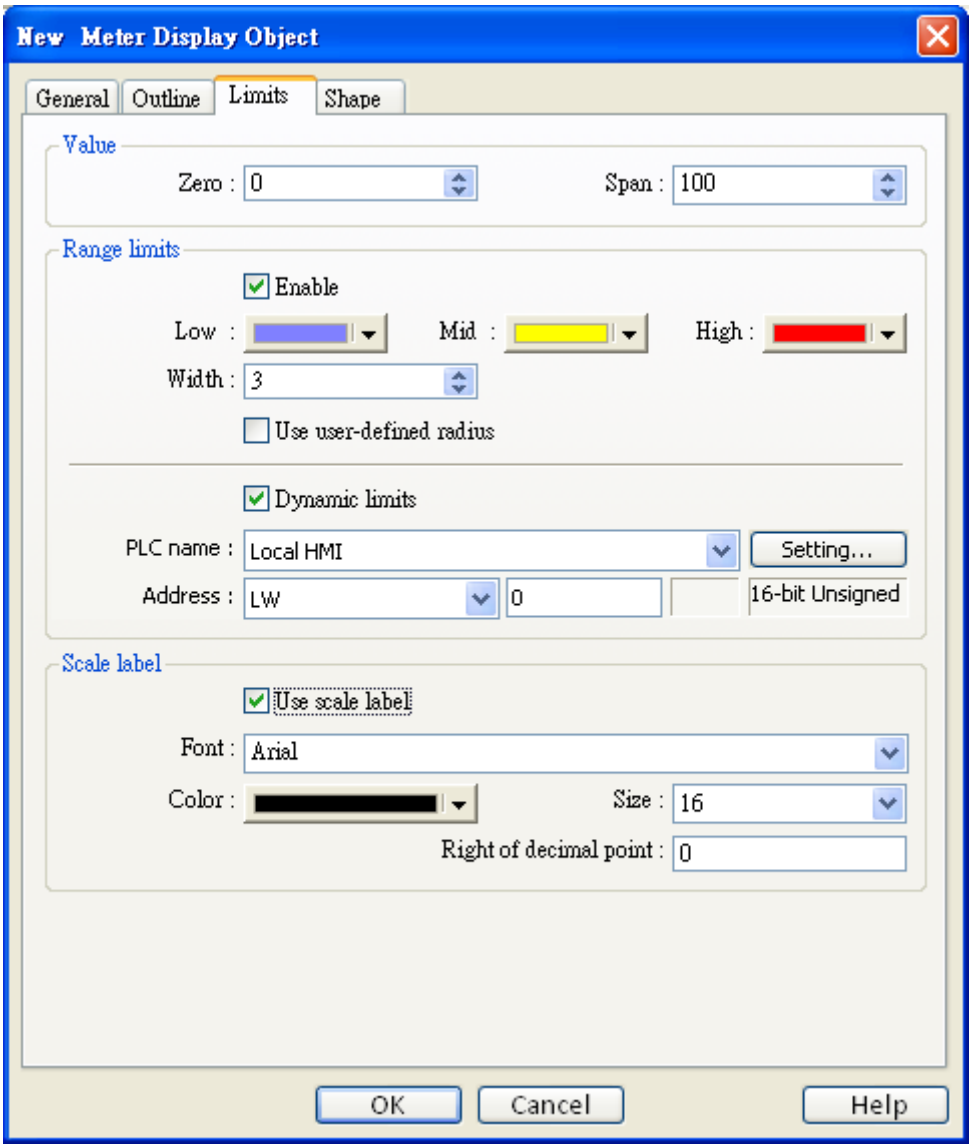
Pointer

Установка стиля, длины, ширины и цвета указателя.

Pin point .

Настройка стиля, радиуса и цвета центральной точки (оси указателя).

На рисунке ниже показана закладка **[Limit]** и значение нижнего и верхнего пределов в закладке [Limit]



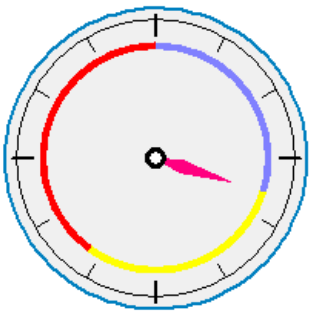
Value - Значение

Задание отображаемого диапазона. Для вычисления позиции указателя будут использоваться значения [Zero] и [Span] и значение по указанному адресу. Например, если [Zero] = 0, [Span] = 100, и по адресу находится значение 30, [Start degree] = 0, [End degree] = 360, то угол указателя будет вычислен следующим образом:

{(30 – [Zero])/([Span] – [Zero])} * ([[End degree] - [Start degree]] =

{(30 – 0) / (100 - 0)} * (360 – 0) = 108

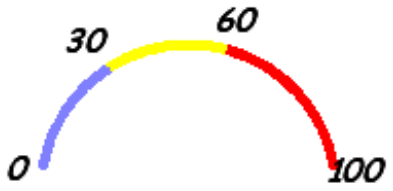
Таким образом, указатель будет находиться в положении 108°. См. рисунок.



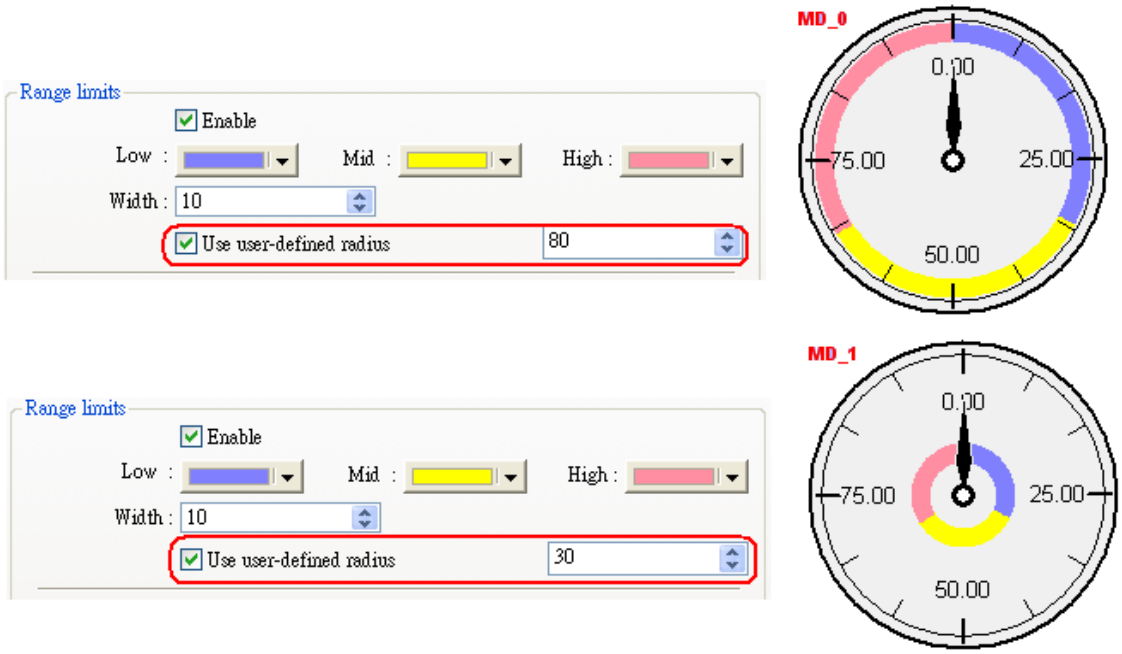
Range limit - Пределы

Установка значений нижнего и верхнего пределов, цвета, ширины отображения диапазонов нижнего, верхнего пределов.

Результат использования настроек в окне выше выглядит так:



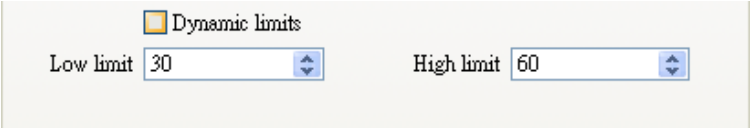
[user-defined radius] – Радиус определяется пользователем



[Dynamic Limits] / uncheck – Динамические пределы не выбраны

Когда флаг [Dynamic Limits] не отмечен, нижний и верхний пределы имеют фиксированные значения, которые берутся непосредственно из настроек объекта.

См. рисунок ниже: заданы значения 30 и 60.



[Dynamic Limits] / check - Динамические пределы выбраны

Когда данный флаг отмечен нижний и верхний пределы определяются указанными регистрами.

Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** адреса слов, которое управляют динамическими пределами.

Пользователи могут назначить адрес, также, в закладке – *Limits,* при создании нового объекта.

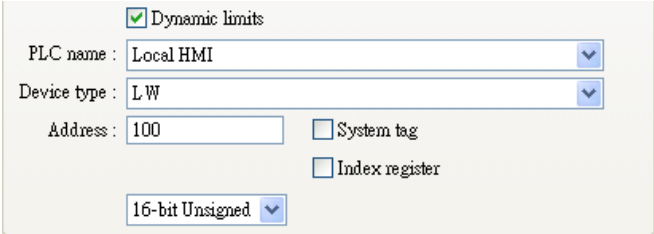
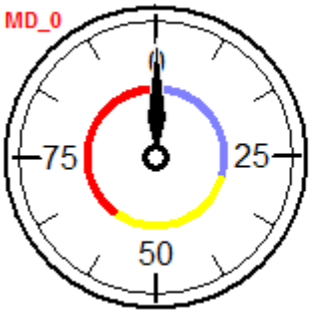


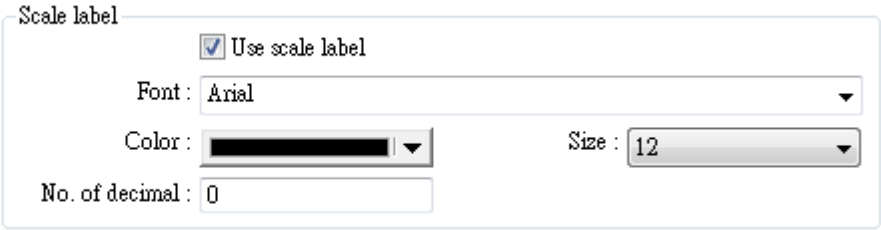
Таблица ниже показывает, как распределяются адреса, содержащие нижний и верхний пределы. Если например, указан регистр [LW100], то “Address” будет равен 100.

Формат данных	Адрес верхнего предела	Адрес нижнего предела
16-bit	Address	Address + 1
32-bit	Address	Address + 2

Scale label

Выбор атрибутов меток шкалы

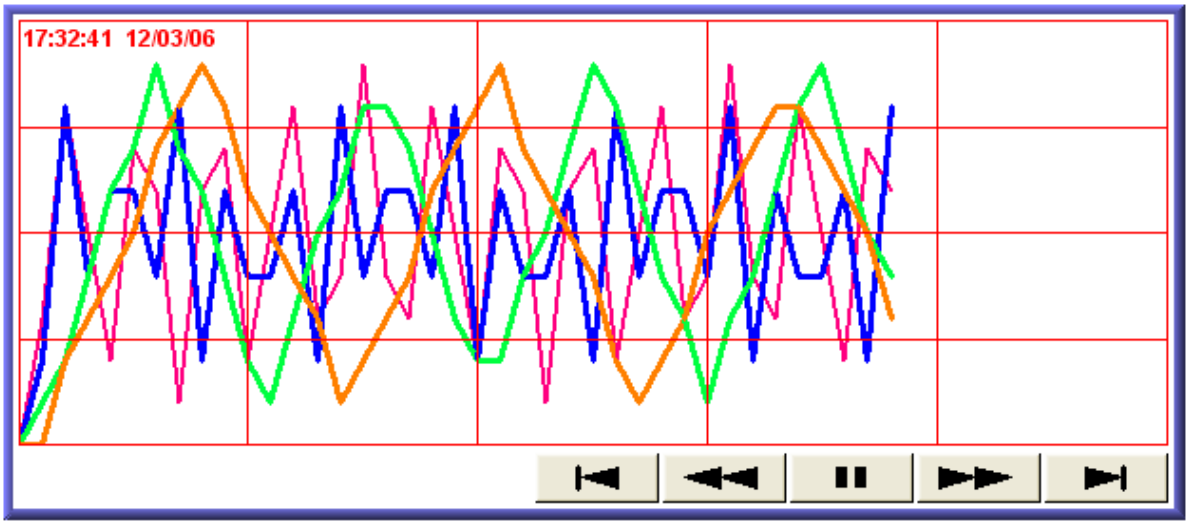




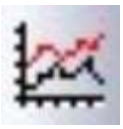
13.17 Trend Display – Регистрирующий прибор

Краткое описание

Объект [Trend Display] используется для построения кривой изменения значения переменной - тренда, показывающей измерения в динамике. Функция сбора данных осуществляется объектами сбора. Объект [Trend Display] показывает результат выборки. На рисунке ниже приведен пример такого объекта.

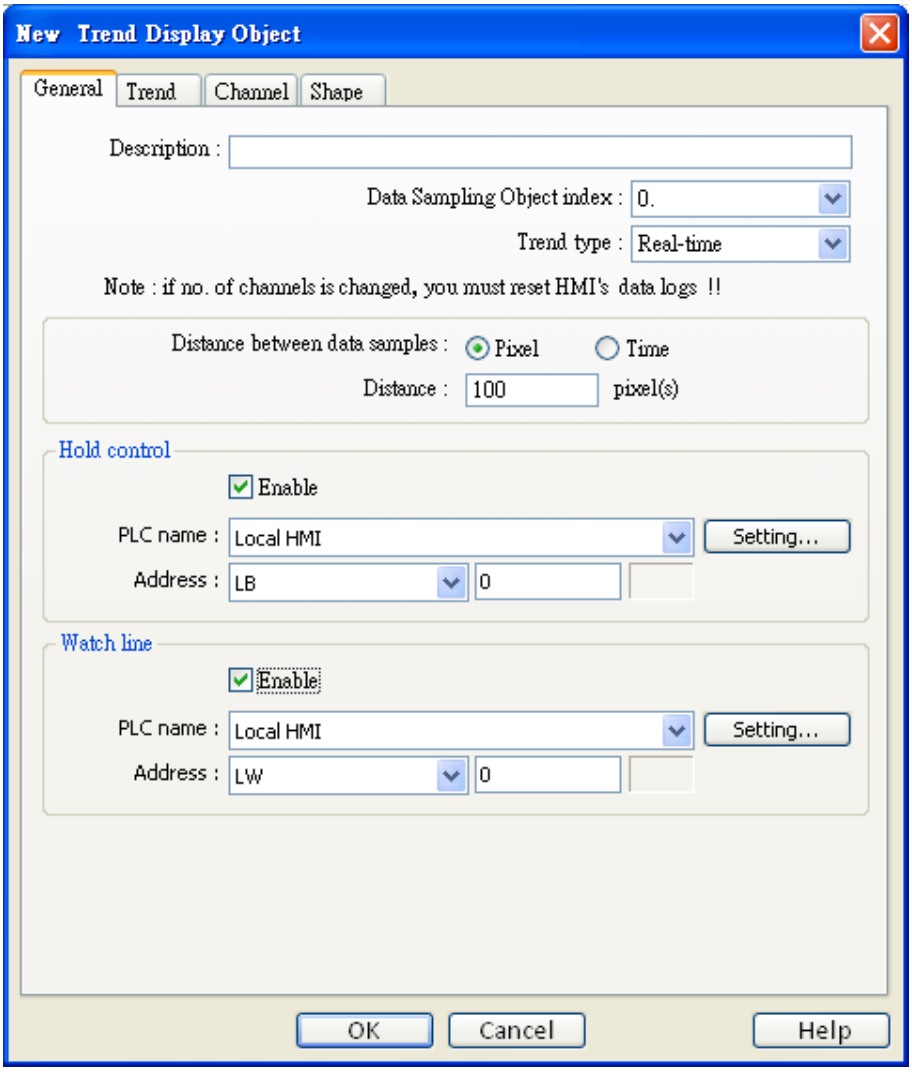


Настройка свойств



Щелкните пиктограмму [Trend Display] на панели инструментов — откроется окно [New Trend Display Object], заполните поля, нажмите OK — будет создан новый объект. См. рисунки далее.

На рисунке ниже показан вид вкладки [General] диалогового окна задания свойств данного объекта.



[Data Sampling Object index]

Для указания выборки данных, используемых в качестве источника информации. См. главу 8 «Сбор данных».

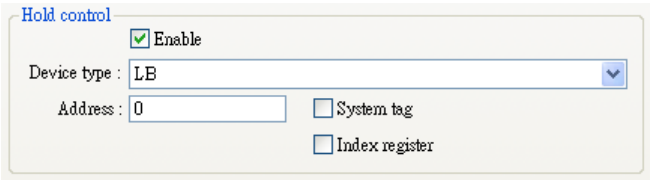
[Trend mode]

Указание источника данных. Есть два варианта: “Real-time” и “History”

a. Real-time

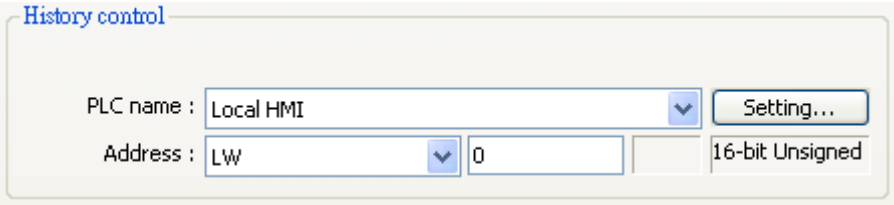
В данном режиме отображаются данные с момента запуска панели и до текущего времени. Если требуются данные более ранних периодов, то нужно выбрать режим “History” для возможности чтения данных, записанных в файлы.

Можно использовать функцию для замедления обновления кривой тренда, однако сбор данных будет продолжаться. На рисунке ниже показаны настройки “Hold control”. Если необходимо замедление обновления установите бит состояния в ON.



b. History

В данном режиме информация берется из архивной записи назначенной в поле [Data sampling object index] выборки данных. Объект сбора данных будет использовать выборку для сортировки данных по датам. Для выборки архивных записей, созданных одним и тем же объектом сбора данных, используется функция “History control”.
На рисунке ниже показаны ее настройки.



Архивные записи сортируются по датам; последний файл выборки — это обычно «record 0» (в стандартной ситуации — это сегодняшняя выборка), второй файл «record 1» и т.д.

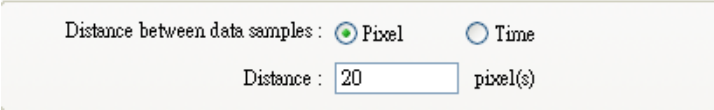
Если значение указанного в области регистра равно «n», то кривая тренда будет отображать данных из записи «record n».

Далее приведен пример, поясняющий использование функции “History control”.
На рисунке указан адрес [LW200], допустим файлы выборки данных называются: pressure_20061120.dtl, pressure_20061123.dtl, pressure_20061127.dtl и pressure_20061203.dtl и сегодня 3.12.2006.
В зависимости от значения по адресу [LW200] будет выбран соответствующий файл для кривой тренда.

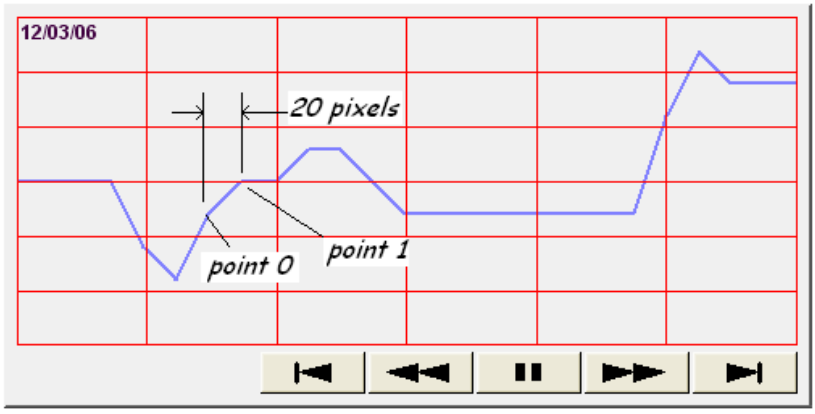
Значение по адресу [LW200]	**Файл выборки**
0	pressure_20061203.dtl
1	pressure_20061127.dtl
2	pressure_20061123.dtl
3	pressure_20061120.dtl

[Distance between data samples:] / Pixel – Расстояние между опросами / В пикселах

При выборе [Pixel]

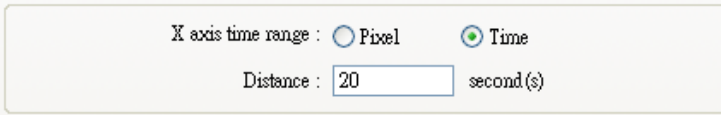


Параметр [Distance] устанавливает расстояние в пикселях между двумя точками измерения. См. рисунок ниже.

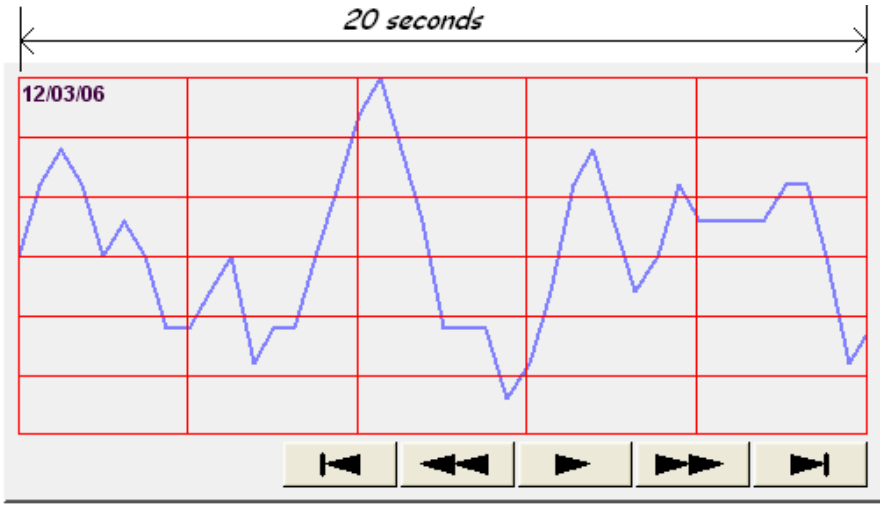


[X axis time range] / Time – Масштаб времени по оси X

При выборе [Time]



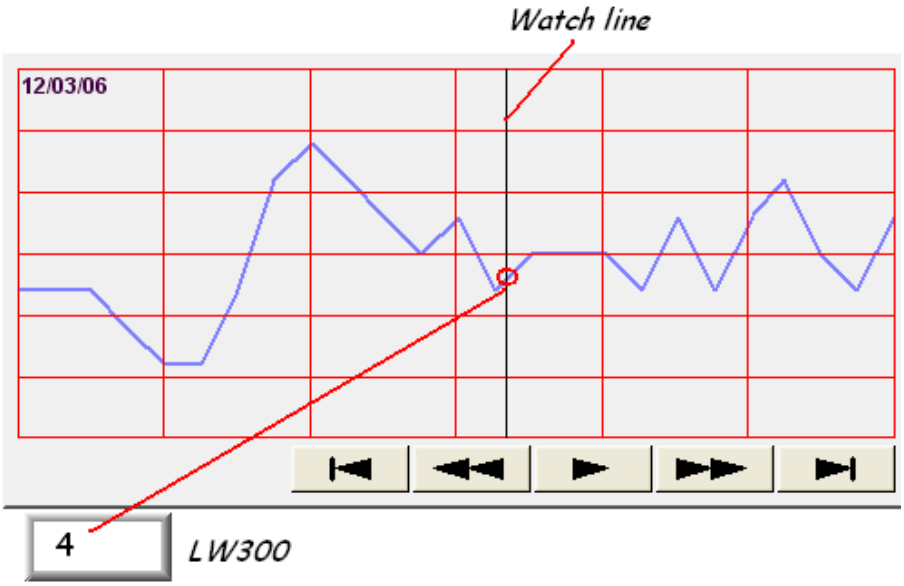
В этом случае параметр [Distance] используется для установления отображаемого диапазона оси X в единицах времени. См. рисунок ниже.



По-другому можно указать параметр [Time] и на закладке [Trend] активировать функцию “Time scale”. Об этой функции см. далее.

Watch line – курсор (линия наблюдения)

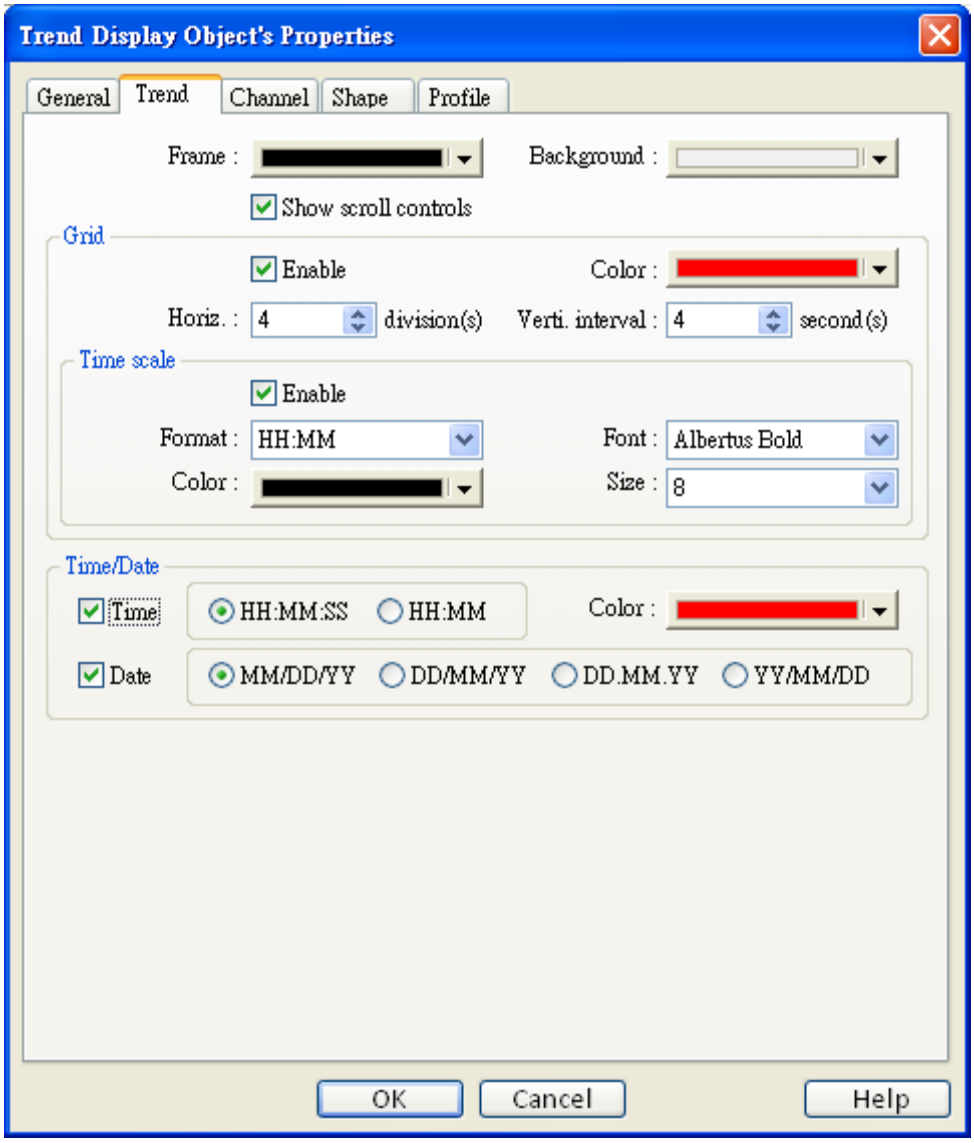
При использовании функции “Watch line”, когда пользователь касается объекта [Trend display], на объекте появляется прямая “watch line” (линия наблюдений) и происходит экспорт данных, находящиеся на пересечении этой линии и кривой тренда по указанному адресу. При этом для отображения значений можно использовать объект [Numeric display] . См. рисунок ниже.



Функция “Watch line” может эскпортировать данных с нескольких кривых трендов. Адрес указанный в поле “watch line” — начальный адрес, начиная с которого будут записываться данные. Могут использоваться данные разных форматов, соответствующие адреса будут последовательными. См. информацию ниже:

[LW300]	Ch. 0 : 16-bit Unsigned	(1 words)
[LW301]	Ch. 1 : 32-bit Unsigned	(2 words)
[LW303]	Ch. 2 : 32-bit Unsigned	(2 words)
[LW305]	Ch. 3 : 16-bit Signed	(1 words)

На рисунке ниже показана Закладка [Trend] окна настройки объект [trend display].

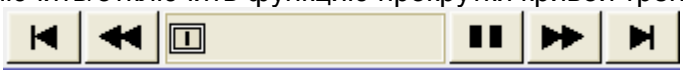


Frame Цвет рамки.

Background Цвет фона.

Show scroll controls

Включить/отключить функцию прокрутки кривой тренда.



Grid

Установка шага и цвета сетки.

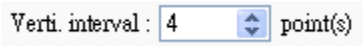
[Horiz.]

Установка числа горизонтальных линий.

[Verti. interval] – Вертикальный интервал

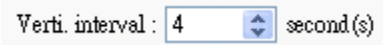
a. Pixel

Когда установлен режим [pixel] интервалов сетки, параметр [Verti. interval] используется для указания количества точек кривой, размещенных между двумя вертикальными линиями сетки. См. рисунок ниже.



b. Time

Когда выбран режим [Time], параметр [Verti. interval] используется для установления временного диапазона между двумя вертикальными линиями сетки. См. рисунок ниже.



При данной установке количество вертикальных линий сеток будет вычисляться автоматически.

[Time Scale]

Подключение функции временного масштаба в нижней части объекта [Trend Display].

[Format]

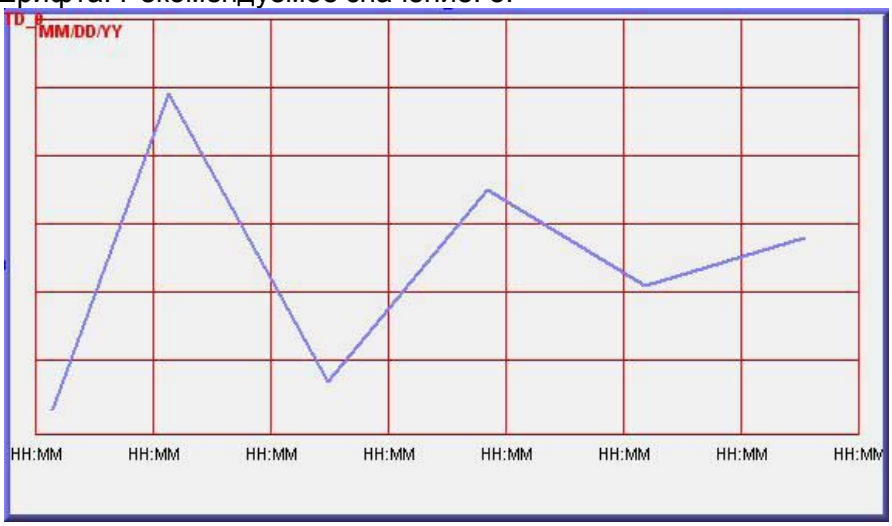
Выбор формата отображения времени: HH:MM или HH:MM:SS (только часы и минуты или часы, минуты и секунды).

[Font]

Установление стиля шрифта.

[Size]

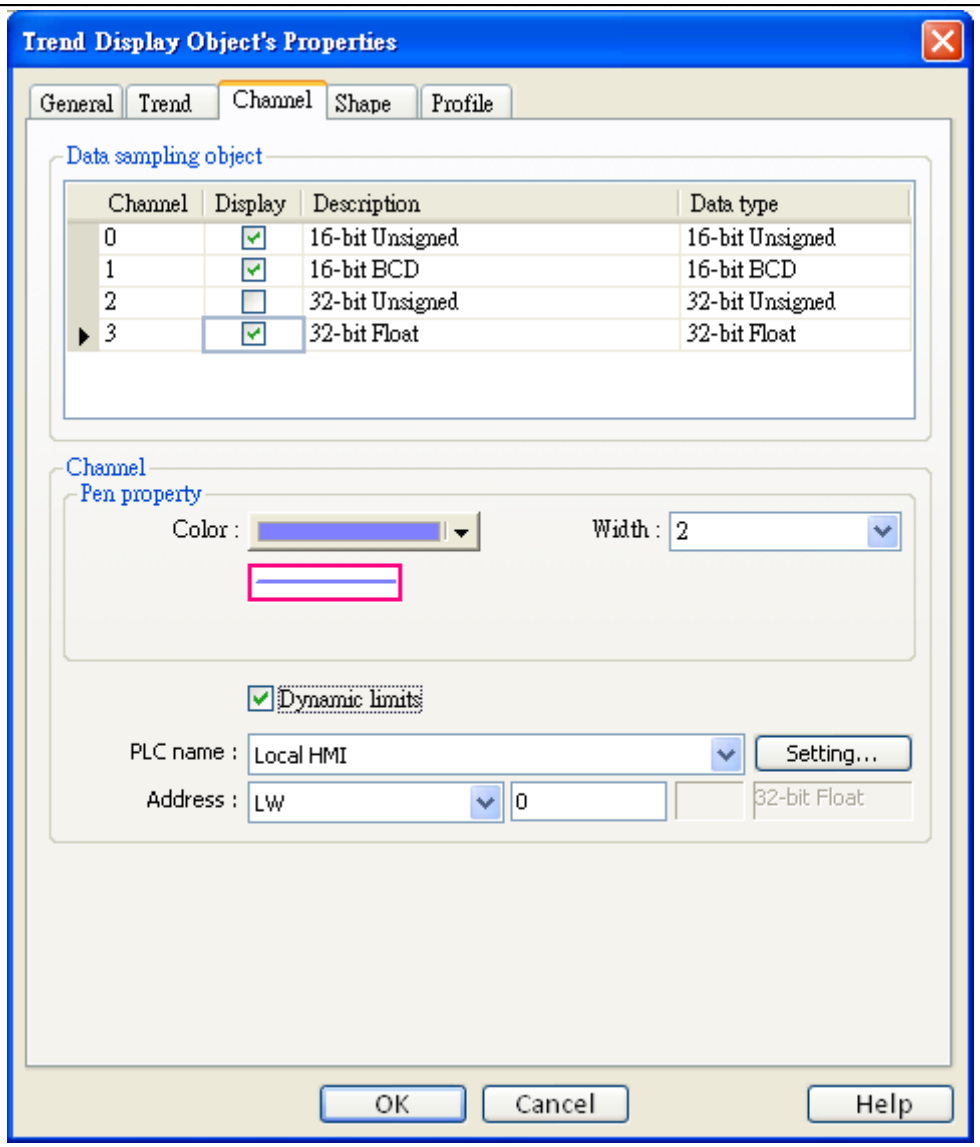
Выбор высоты шрифта. Рекомендуемое значение: 8.



Time / Date

Время самого последнего опроса будет отмечено в верхнем левом углу объекта. Можно настроить формат и цвет шрифта отображения времени.

На рисунке (на следующей странице) показаны атрибуты закладки “Channel” – Канал измерения.



Channel Установка типа линии и цвета кривой тренда и нижнего и верхнего пределов отображаемых данных. Максимальное число каналов -64.

Limit / uncheck “Dynamic limits” – Динамические пределы не отмечены

[Zero]、[Span]

Параметры [Zero] и [Span] используются для установления нижнего и верхнего пределов собираемых данных. Таким образом, если нижний предел [low limit] равен 50, а [high limit] равен 100, то параметры [Zero] и [Span] нужно установить равными 50 и 100, тогда все собираемые данные будут видны в окне [Trend Display].

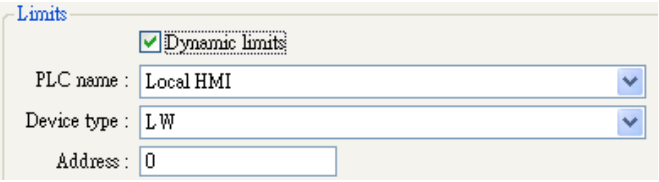
Limit / check “Dynamic limits”– Динамические пределы отмечены

Когда отмечен флаг [Dynamic Limits значения нижнего и верхнего предела находятся по назначенным адресам. Длина данных в этих адресах зависит от формата данных объекта. См. пример:

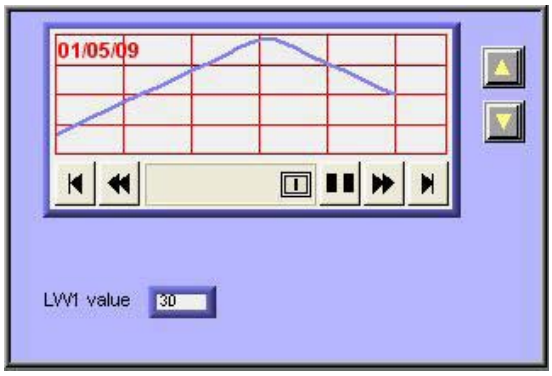
Формат данных	Адрес верхнего предела	Адрес нижнего предела
16-bit	Address	Address + 1
32-bit	Address	Address + 2

Дополнительная функция: «zoom in/out» - увеличение и уменьшение масштаба

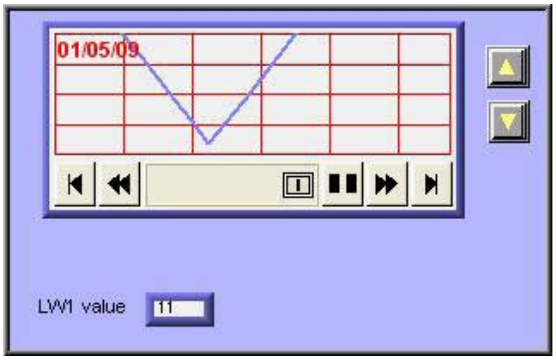
Пример функции «zoom in/out»
Для увеличения/уменьшения кривой тренда нужно отметить флаг [Limits/Dynamic limits].



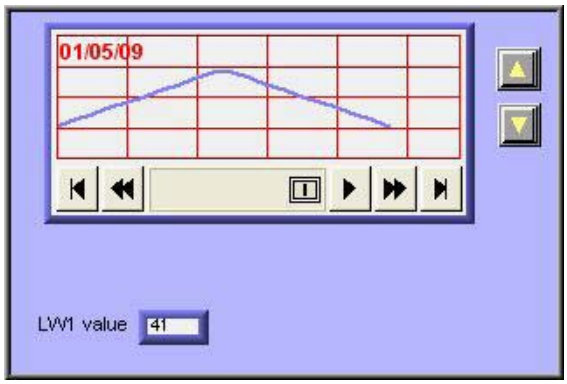
Например, если адреса [LW0] и [LW1] хранят значения нижнего и верхнего пределов, для увеличения/уменьшения графика можно изменять значение по адресу [LW1].
На рисунке ниже показан график с исходными размерам. Диапазон изменения значения тренда: от 0 до 30. Стрелки с правой стороны графика увеличивают/уменьшают по адресу [LW1] для управления масштабом увеличения/уменьшения.



Уменьшим значение по адресу [LW1] для демонстрации функции увеличения:
Значение [LW1] уменьшено до 11.



Увеличим значение по адресу [LW1] для демонстрации функции уменьшения:
Значение [LW1] увеличено до 41.



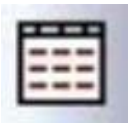
13.18 History Data Display – Просмотр архива данных

Краткое описание

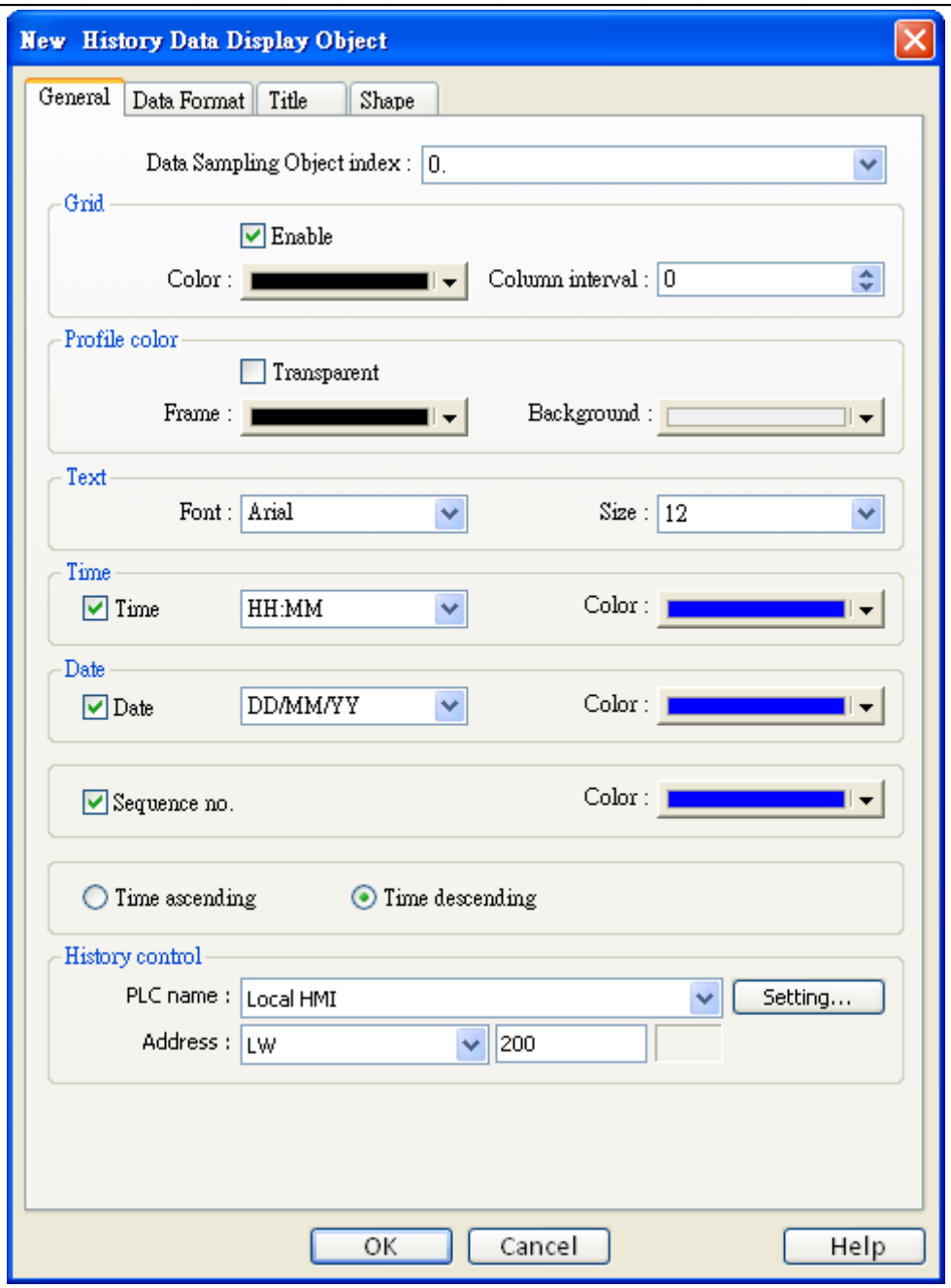
Объект [History Data Display] выводит на экран данные, сохраненные объектом сбора данных. Он отображает архивную информацию в числовой форме. Необходимо помнить, что данный объект не обновляет данные автоматически, он только использует информацию, сохраненную в указанном месте, и отображает ее в рабочем окне. Даже если она была обновлена, объект [History Data Display] этого не покажет.

No.	Time	Date	Ch.0	Ch.1	Ch.2
3577	21:52	16/09/07	0	0	0
3576	21:52	16/09/07	0	0	0
3575	21:52	16/09/07	0	0	0
3574	21:52	16/09/07	0	0	0
3573	21:52	16/09/07	0	0	0
3572	21:52	16/09/07	0	0	0
3571	21:52	16/09/07	0	0	0
3570	21:52	16/09/07	0	0	0
3569	21:52	16/09/07	0	0	0
3568	21:52	16/09/07	0	0	0

Настройка свойств



Щелкните по пиктограмме [History Data Display] на панели инструментов — откроется окно [New History Data Display Object]. Заполните все его поля, нажмите OK — будет создан новый объект.



[Data Sampling object index]

Выберите соответствующую выборку, откуда будут браться архивные данные.

Grid

Включить/отключить сетку.

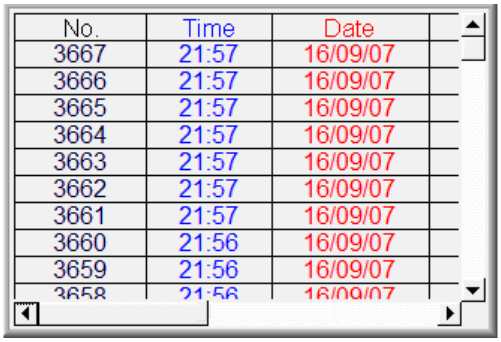
No.	Time	Date	Ch.0	Ch.1	Ch.2
3982	22:02	16/09/07	0	0	0
3981	22:02	16/09/07	0	0	0
3980	22:02	16/09/07	0	0	0
3979	22:02	16/09/07	0	0	0
3978	22:02	16/09/07	0	0	0
3977	22:02	16/09/07	0	0	0
3976	22:02	16/09/07	0	0	0
3975	22:02	16/09/07	0	0	0
3974	22:02	16/09/07	0	0	0
3973	22:02	16/09/07	0	0	0

Grid [Color]

Задание цвета сетки.

[Column interval]
Задание ширины колонки.

No.	Time	Date	Ch.0	Ch.1	Ch.2
3667	21:57	16/09/07	1	0	0
3666	21:57	16/09/07	1	0	0
3665	21:57	16/09/07	1	0	0
3664	21:57	16/09/07	1	0	0
3663	21:57	16/09/07	1	0	0
3662	21:57	16/09/07	1	0	0
3661	21:57	16/09/07	1	0	0
3660	21:56	16/09/07	0	0	0
3659	21:56	16/09/07	0	0	0

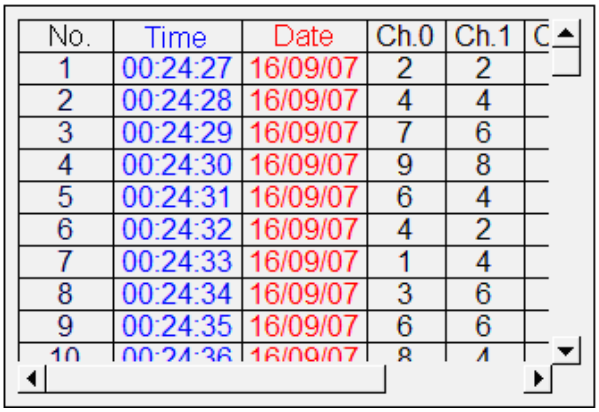


No.	Time	Date
3667	21:57	16/09/07
3666	21:57	16/09/07
3665	21:57	16/09/07
3664	21:57	16/09/07
3663	21:57	16/09/07
3662	21:57	16/09/07
3661	21:57	16/09/07
3660	21:56	16/09/07
3659	21:56	16/09/07

Profile color
Установка цвета рамки и фона. Если установлен режим [transparent] (прозрачный), то параметры рамки и фона учитываться не будут.

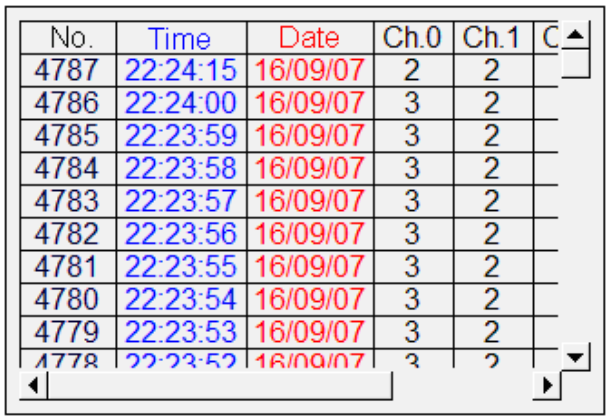
Time and Date
Включение/отключение отображения времени и даты в выборке данных и выбор формата.

[Time ascending]
Сортировка данных по возрастанию времени создания.



No.	Time	Date	Ch.0	Ch.1
1	00:24:27	16/09/07	2	2
2	00:24:28	16/09/07	4	4
3	00:24:29	16/09/07	7	6
4	00:24:30	16/09/07	9	8
5	00:24:31	16/09/07	6	4
6	00:24:32	16/09/07	4	2
7	00:24:33	16/09/07	1	4
8	00:24:34	16/09/07	3	6
9	00:24:35	16/09/07	6	6

[Time descending]
Сортировка данных по убыванию времени создания.



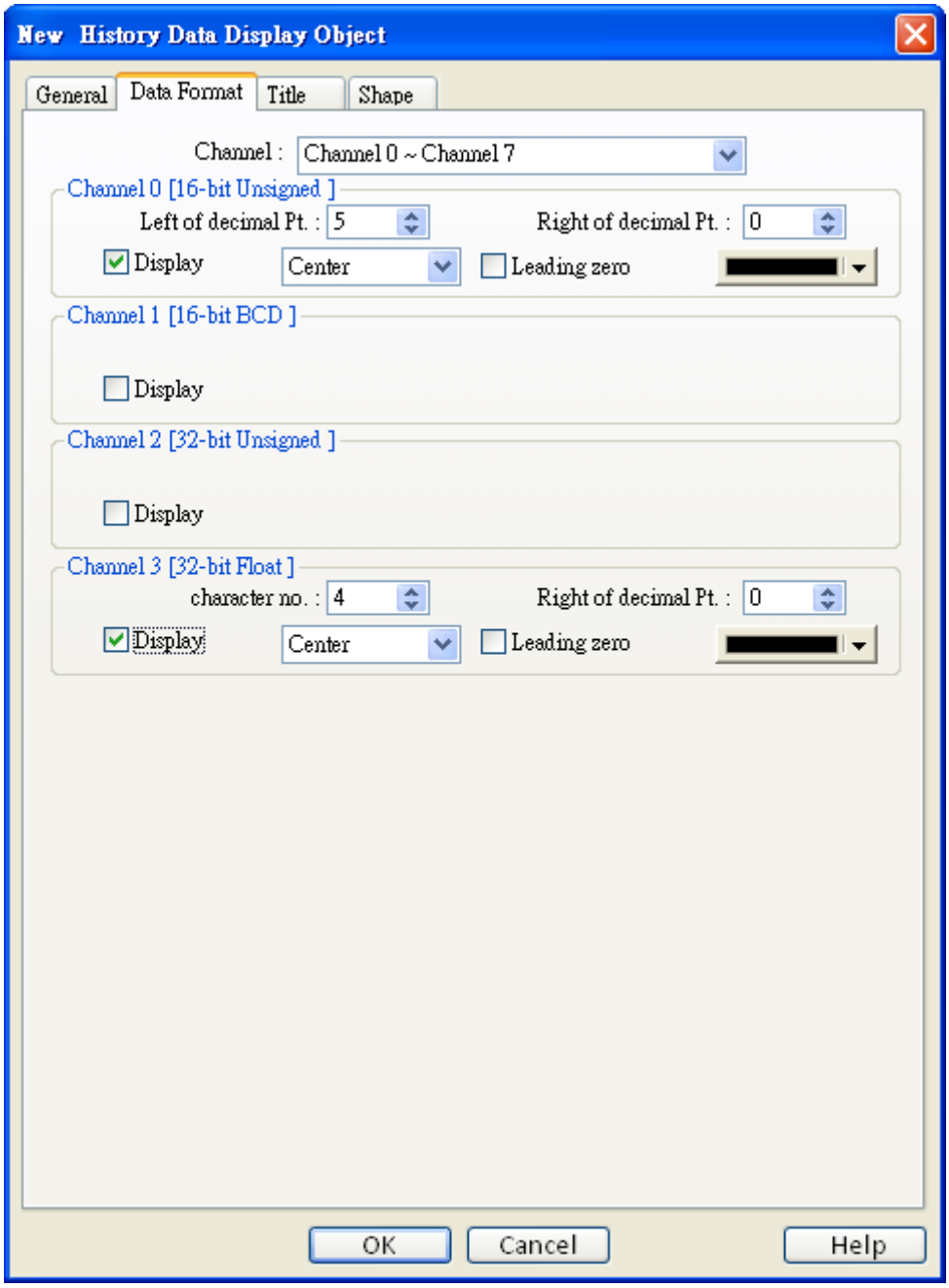
No.	Time	Date	Ch.0	Ch.1
4787	22:24:15	16/09/07	2	2
4786	22:24:00	16/09/07	3	2
4785	22:23:59	16/09/07	3	2
4784	22:23:58	16/09/07	3	2
4783	22:23:57	16/09/07	3	2
4782	22:23:56	16/09/07	3	2
4781	22:23:55	16/09/07	3	2
4780	22:23:54	16/09/07	3	2
4779	22:23:53	16/09/07	3	2

History Control

Файлы архивных данных называются по времени создания. Функция [history control] используется для выбора указанных файлов архивов для вывода на экран. В случае, если значение по указанному адресу равно 0, то отображается последний файл. Если 1, то предпоследний и т.д.

Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** адрес History Control.

Пользователи могут назначить адрес, также, в закладке – *General* , при создании нового объекта.



Каждый объект может отображать значения максимум по 64 каналам сбора информации. Можно выбрать каналы, которые требуется выводить на экран.

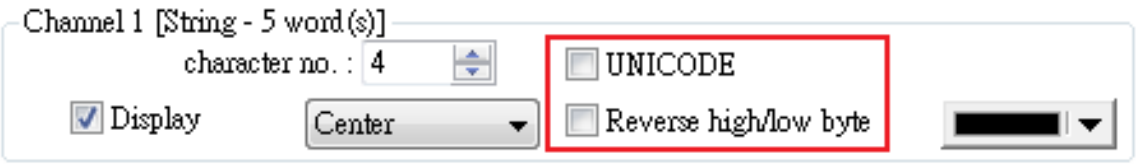
В примере ниже, в объекте сбора данных имеется четыре канала, для вывода на экран выбраны только «ch.0» и «ch.3». Формат данных определяется в объекте сбора информации.

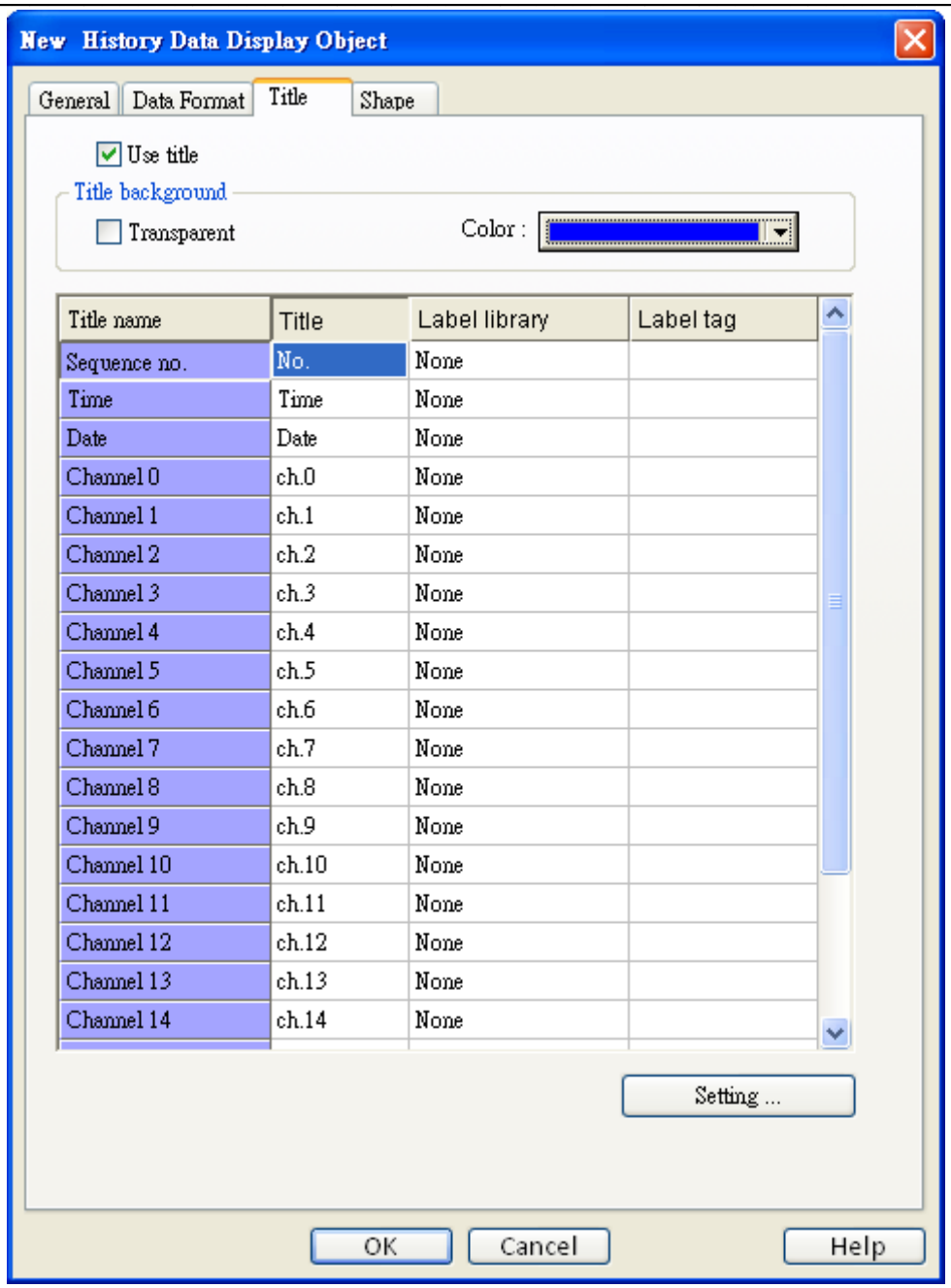
No.	Time	Date	Ch.0	Ch.3
5272	22:43:09	16/09/07	4	1
5271	22:43:08	16/09/07	2	0
5270	22:33:42	16/09/07	0	0
5269	22:33:41	16/09/07	0	0
5268	22:33:40	16/09/07	0	0
5267	22:33:39	16/09/07	0	0
5266	22:33:38	16/09/07	0	0
5265	22:33:37	16/09/07	0	0
5264	22:33:36	16/09/07	0	0
5263	22:33:35	16/09/07	0	0

При отображении формата [String] в объекте «history data display», пользователи могут выбирать:

a. Отображать в режиме [UNICODE]

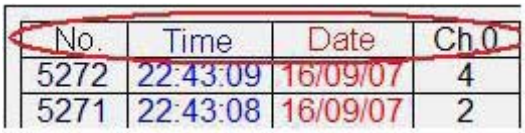
b. Менять местами байты - «high byte» и «low byte» данных при отображении.





[Use title]

Включить/отключить заголовок.



Title background

[Transparent]

Включить/отключить прозрачность.

[Background color]

Цвет фона заголовка.

Setting

Данное диалоговое окно определяет заголовок.

No.	Time	Date	Ch.0
5272	22:43:09	16/09/07	4
5271	22:43:08	16/09/07	2

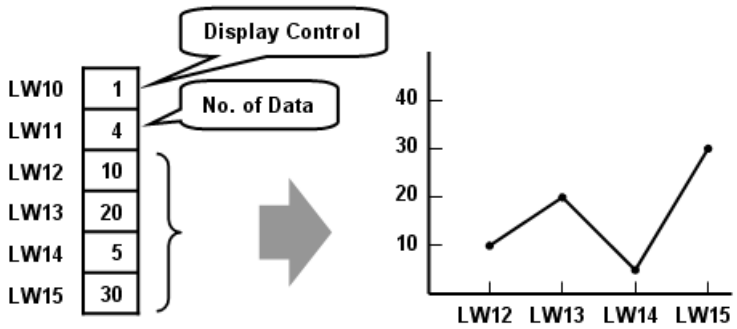
Можно использовать библиотеку надписей для многоязыкового заголовка. Нажмите [Setting…] и выберите одну из возможных меток.

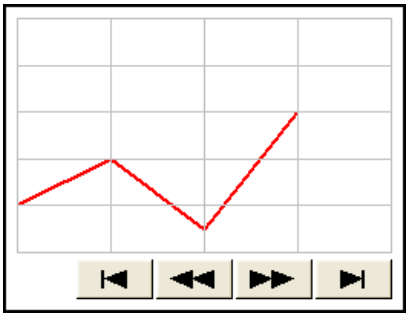
Примечание:
Если вы запускали офф-лайн имитацию, и собранные данные сохранялись в файле, то при необходимости изменения формата данных нужно удалить старые архивные записи в папке **C:\EasyBuilder Pro\HMI_memory\datalog** во избежание неверной интерпретации их системой.

13.19 Data Block - Отображение блока данных

Краткое описание

[Data Block] (блок данных) — это данные, расположенные в последовательных адресах, например, в LW12, LW13, LW14, LW15 и т.п. Можно использовать объект [Data Block Display] для отображения нескольких блоков данных в виде кривой тренда; например, можно отображать два блока данных LW12~LW15 и RW12~RW15 одновременно в виде кривых тренда. Это полезная функция для сравнения различий кривых тренда.



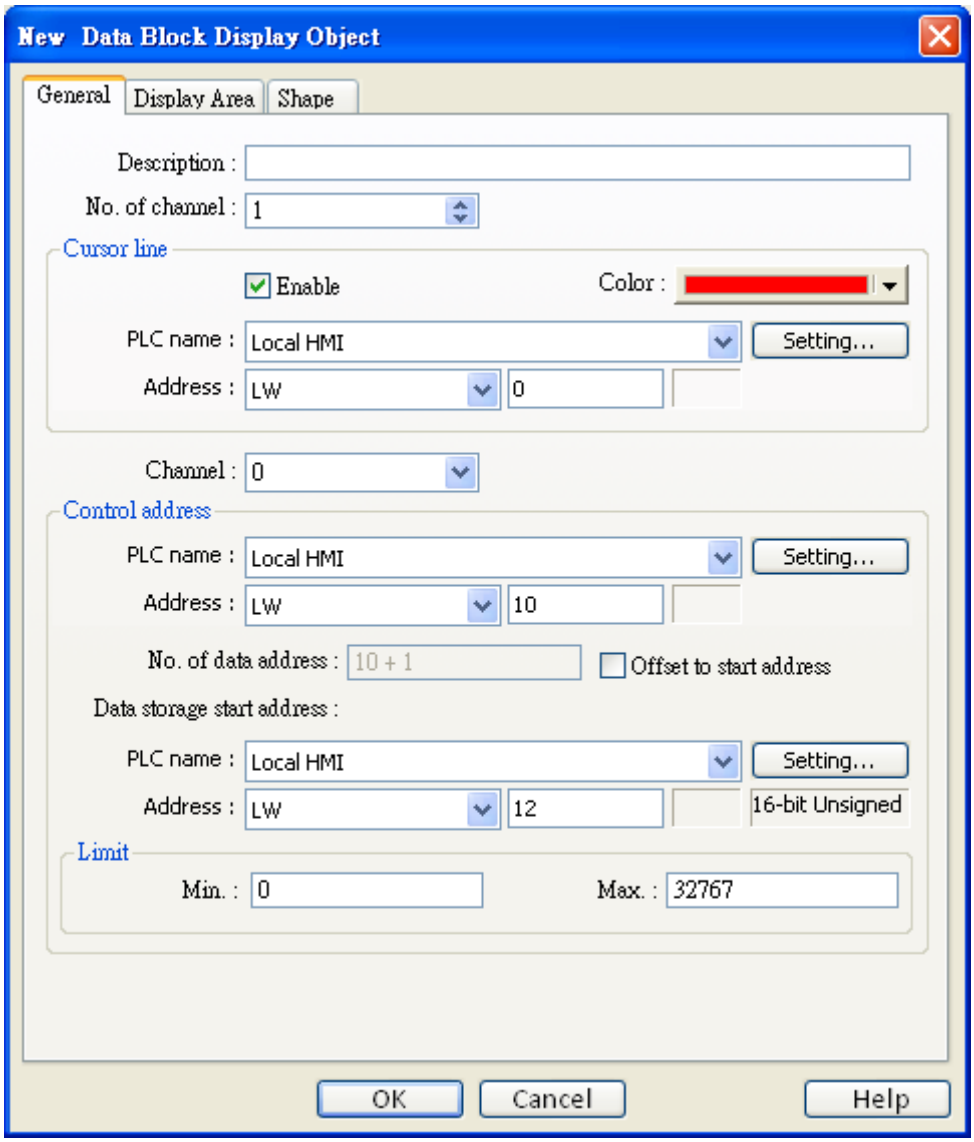


Мгновенная картинка объекта [Data Block Display]

Настройка свойств

[New object]

Щелкните по пиктограмме [Data Block Display] — откроется диалоговое окно [Data Block Display's properties]:



No. of channel
Задайте число каналов измерений для данного объекта. Каждый канал соответствует одному блоку данных. Максимальное число: 12.

Cursor Line
При использовании данной функции, когда пользователь касается объекта [Data Block display], будет отображаться линия наблюдения на объекте, и ее положение, а также значение, соответствующее этому положению, будут записаны в назначенный регистр.

См. также раздел: Работа в режиме он-лайн.

Channel Для каждого канала установите значения аттрибутов.

Control address – Управляющий адрес

[PLC name]
Выберите устройство, в котором находится нужный блок данных.

Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** управляющего адреса.
Пользователи могут назначить адрес в закладке -*General* , при создании нового объекта.

[Device type]
Укажите тип адресного пространства.

[Control word address]

Данное слово используется для управления и очисткой кривой тренда.

0 = Не производить никаких действий (по умолчаниию)

1 = Вывести кривую тренда на печать

2 = Очистить кривую тренда

3 = Обновить кривую тренда

После выполнения указанных действий управляющее слово будет автоматически установлено в 0.

[No. of data address]

По умолчанию данный параметр установлен как “Control word address +1”.

По данному адресу хранится число значений в каждом блоке данных, т.е. количество значений, выводимых на печать на кривой тренда. Максимальное значение: 1024.

[Data storage start address]

Если функция [offset to start address] отключена, то адрес “Data storage start address” по умолчанию задается как “Control word address + 2”.

[Offset value storage address]

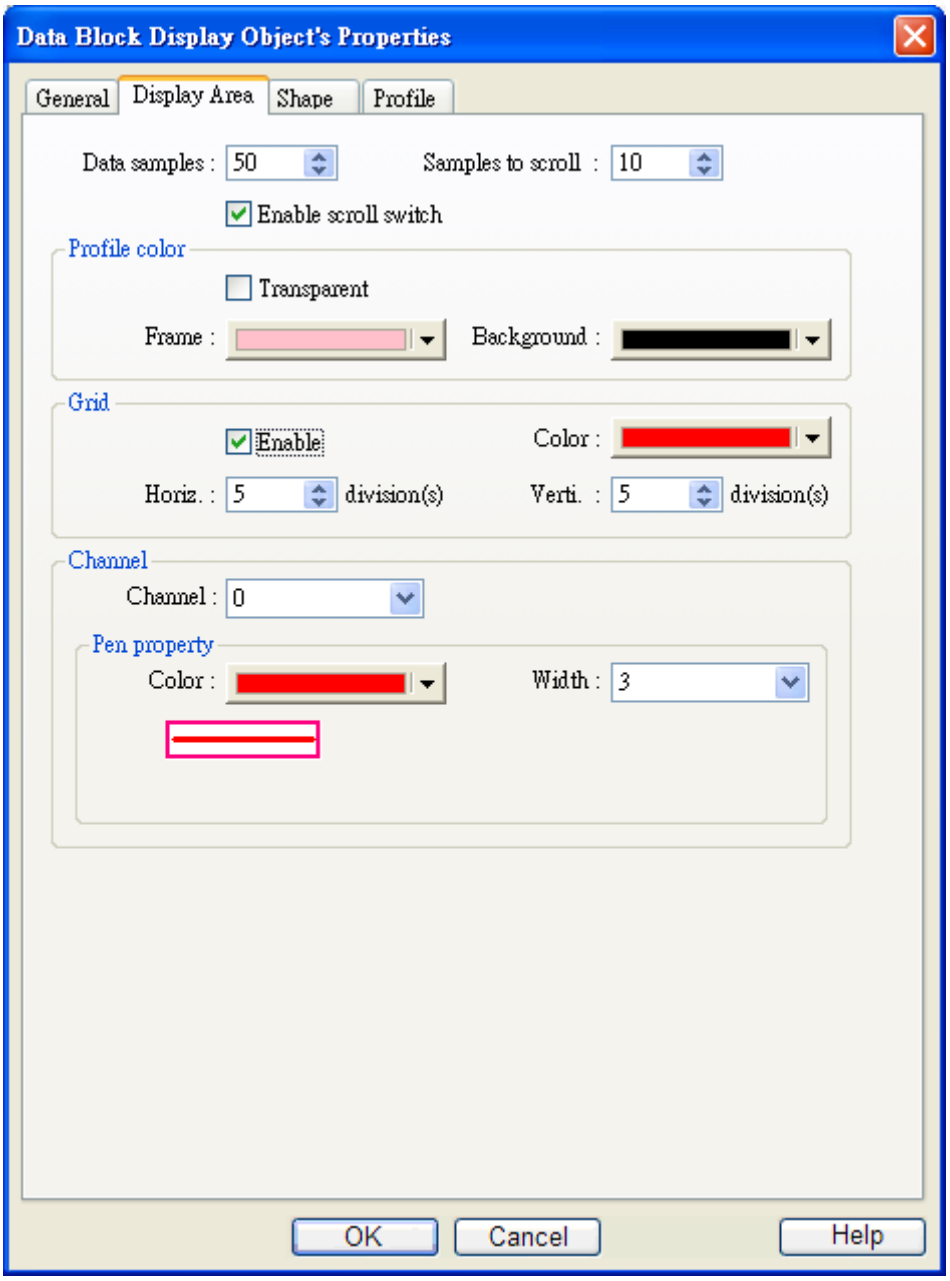
Если функция [offset to start address] включена, то адрес “Data storage start address” вычисляется как “[Offset value storage] + Control word address”, где “Offset value storage address” — есть значение по умолчанию “Control word address” + 2.

[Format]

Если выбран 16-разрядный формат данных, то адреса значений будут последовательными: start address, start address + 1, start address + 2 и т.д.

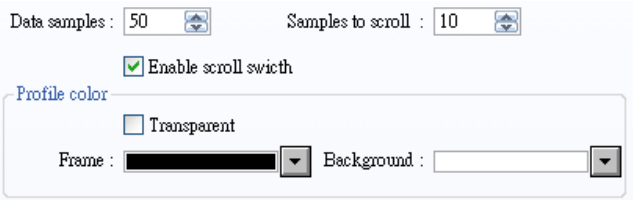
Если выбран 32-разрядный формат данных, то адреса будут: start address, start address + 2, start address + 4 и т.д.

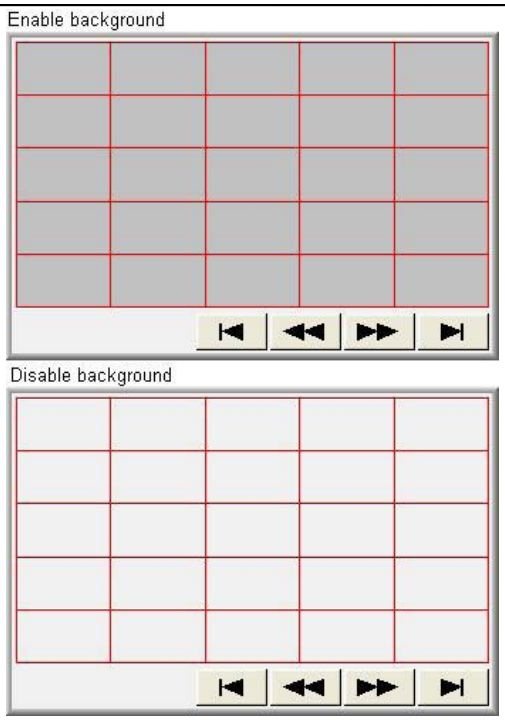
Limit
Установка минимального и максимального пределов кривой тренда, кривая ограничена этими значениями.



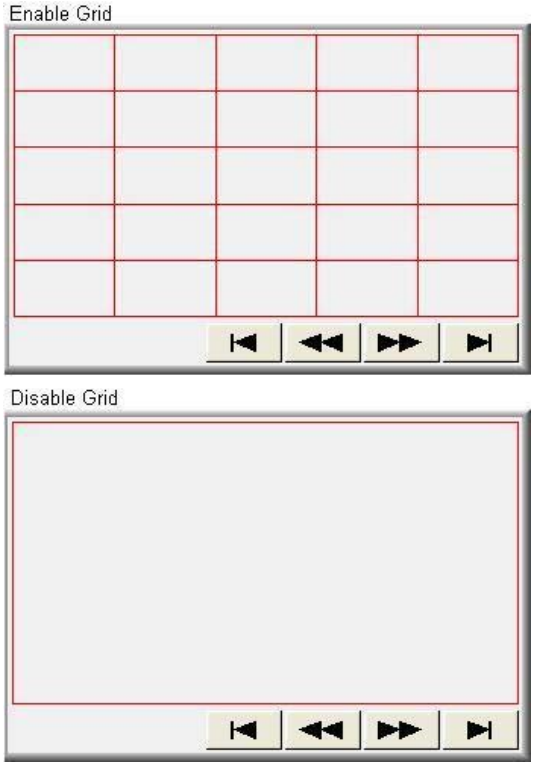
Data Samples – Выборки данных

Установка выборок данных, параметров прокрутки, рамки и цвета фона.





Grid Сетка.



Channel Задание цвета и типа линии кривой тренда для каждого канала.

Работа в режиме он-лайн

Отображение кривой тренда

a. Назначьте количество значений в поле [No. of data address], т.е. по адресу “control word address+1”

b. Подготовьте содержимое блока данных для вывода на экран.

Примечание: блок данных начинается с адреса “control word address + 2”.

c. Запишите «1» в [Control word address], предыдущая кривая тренда останется и новое содержимое блока данных будет выведено на экран.

d. После того как кривая будет выведена в [Control word address], автоматически запишется «0».

Примечание *:* Между выполнением пп. c и d не меняйте содержимое ячеек памяти [Control], [No. Of Data] и [Data], т.к. это может привести к ошибкам в построении кривой тренда.

Очистка кривой тренда

a. Запишите «2» в ячейку [Control word address] — все кривые тренда будут удалены.

b. В ячейку [Control word address] автоматически будет записан 0 после удаления всех кривых.

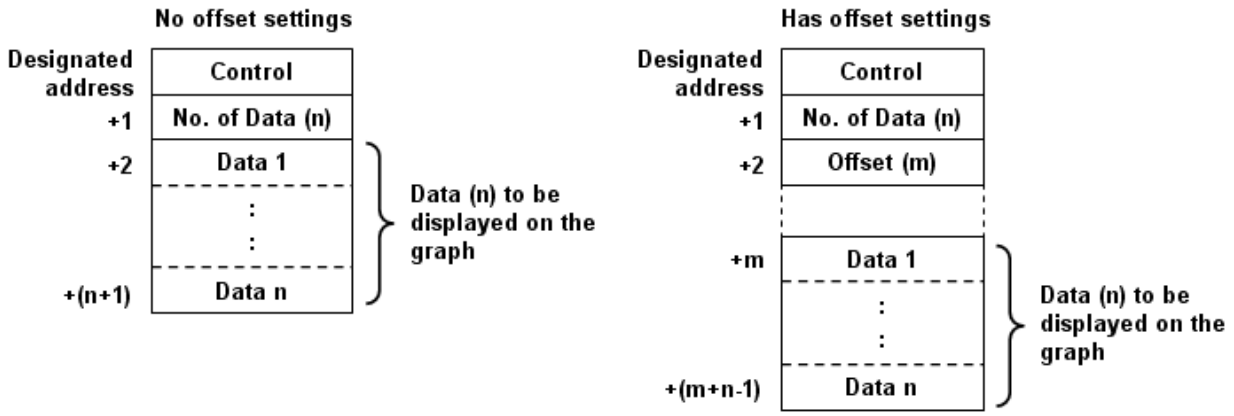
Удаление предыдущей кривой тренда и отображение новой

a. Занести число данных в ячейку [No. of data address], т.е. по адресу “word control address+1”
b. Подготовьте для вывода блок данных.
   Примечание: блок данных начинается с адреса “word control address + 2”.
c. Занесите «3» в ячейку [Control word address], предыдущие кривые тренда будут удалены и новое содержимое блока данных будет выведено на экран.
d. После вывода кривых тренда в ячейку [Control word address] будет автоматически записан 0.

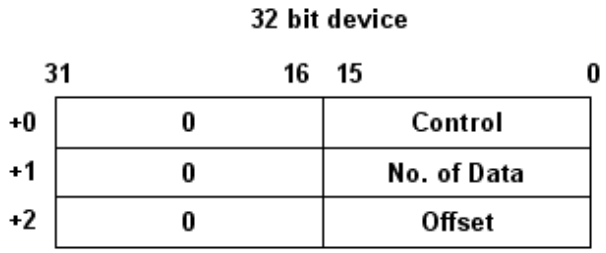
Использование режима смещения адреса [offset]

Если отмечен флаг “offset to start address”, то адрес “Data storage start address” будет вычислен как “control word address + [Offset value storage address]”. “Offset value storage address” — это “control word address +2”.

В примере ниже значение “Offset value storage address” равно **m**, таким образом, блок данных начинается с адреса “control word address + m”.



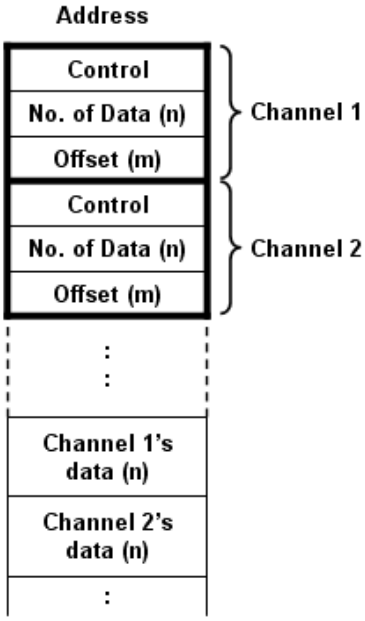
Примечание :
Если управляющий регистр — 32-разрядный, только биты с 0 по 15 будут использоваться в целях управления, биты с 16 по 31 не учитываются (как на рисунке ниже).



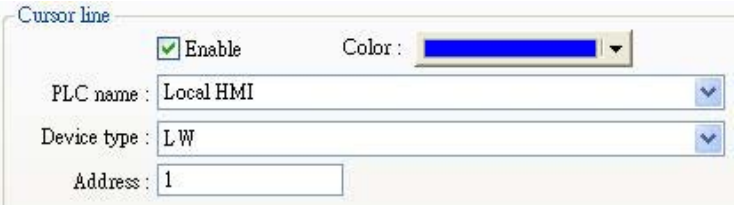
Если не используется функция “offset to start address”, то непрерывно будут читаться адреса [Control] и [No. of Data]. Когда значение ячейки [Control] становится отличным от нуля, автоматически будет читаться идущий следом блок данных. Если используется функция “offset to start address”, то непрерывно будут читаться адреса [Control], [No. of Data] и [Offset].

Рекомендуется использовать функцию “offset to start address” для отображения блока данных с множеством каналов и одинаковым типом адресного пространства. Можно хранить управляющие числа [Control], [No. of Data] и [Offset] каждого канала в непрерывной области. Система автоматически считает их значения за одну команду обращения к регистрам, что увеличит скорость реакции.

На рисунке ниже: управляющие слова канала 1 размещаются, начиная с адреса address 0, управляющее слово канала 2 — с address 3; таким образом, эти управляющие данные расположены непрерывно, и система может их прочитать за одну команду чтения.

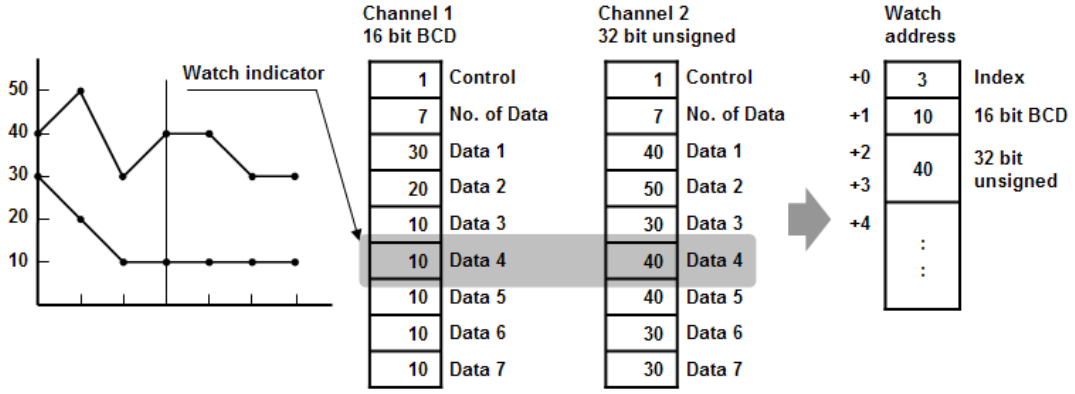


Использование функции просмотра значения точек графика [Watch]



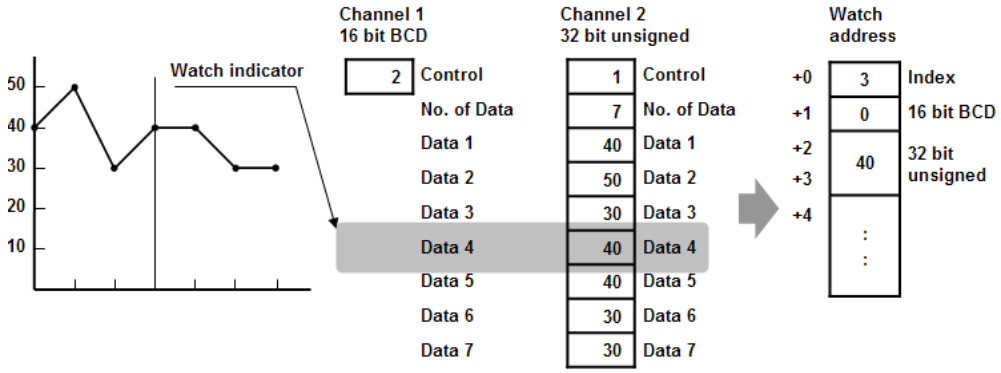
Можно использовать функцию [Watch] для просмотра значения любой точки на кривой тренда. Когда человек-оператор касается объекта [data block], то появляется “Cursor line”, по назначенному адресу будет записано позиция и значение. Это значение можно увидеть с помощью объекта [Numeric value].

В примере объект [data block] включает два блока данных. Формат данных канала 1 — «16 bit BCD», а канала 2 — «32 bit unsigned». Линия трассировки находится в узле 3, который соответствует четвертому значению в блоке данных. В определенный ранее watch-адрес будут записаны: позиция (номер узла), и содержимое блоков данных как показано на рисунке ниже.

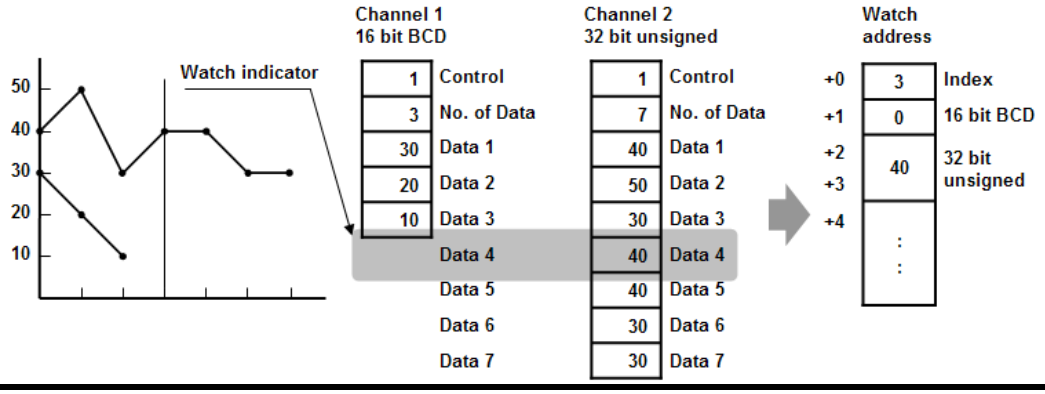


Примечания:

1. [Data Index] – число формата 16 bit unsigned integer; если используется 32-разрядный регистр, то задействованы только биты с 0 по 15.
2. Функция линии трассировки позволяет просматривать только текущие данные из блока. Если на экране присутствуют несколько кривых тренда одного и того же канала, для просмотра будут доступны только последние данные.
3. Если кривая тренда удалена, то при просмотре с помощью линии трассировки, будет выводиться «0» как показано ниже.



4. Если в блоке channel 1 только 3 значения, в то время как линия находится на пересечении с узлом, соответствующим Data 4, то также будет показано значение «0».



Ограничения:

1. **Максимальное число каналов: 12.**
2. **Может выводиться максимум 32 кривых тренда.**
3. **Для каждого канала может быть показано не более 1024 узловых точек.**

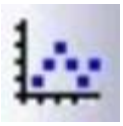
13.20 XY Plot - Двумерный график

Краткое описание

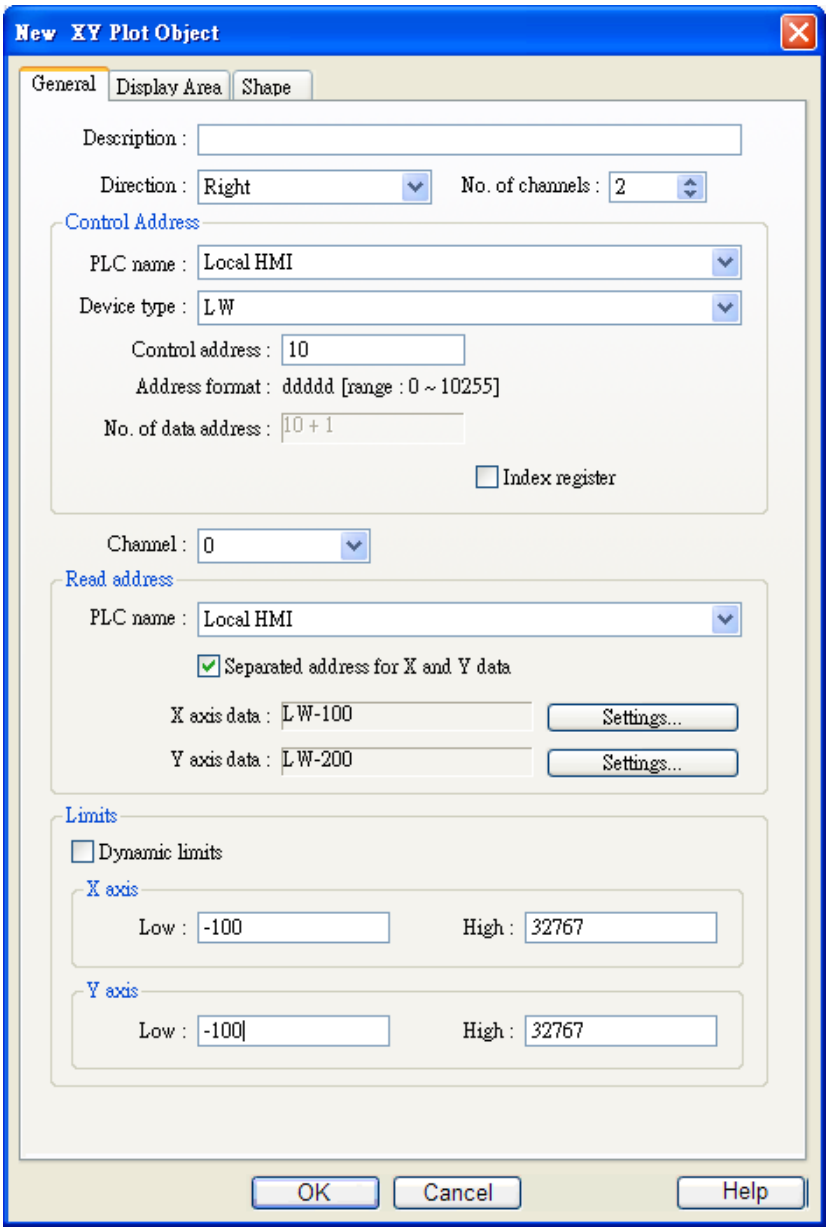
Объект [XY Plot] отображает двумерные данные. Каждая точка графика имеет координаты X и Y и каждая кривая состоит из ряда таких точек. Максимальное число кривых трендов, отображаемых на данном графике равно 16.

Настройка свойств

[New object]

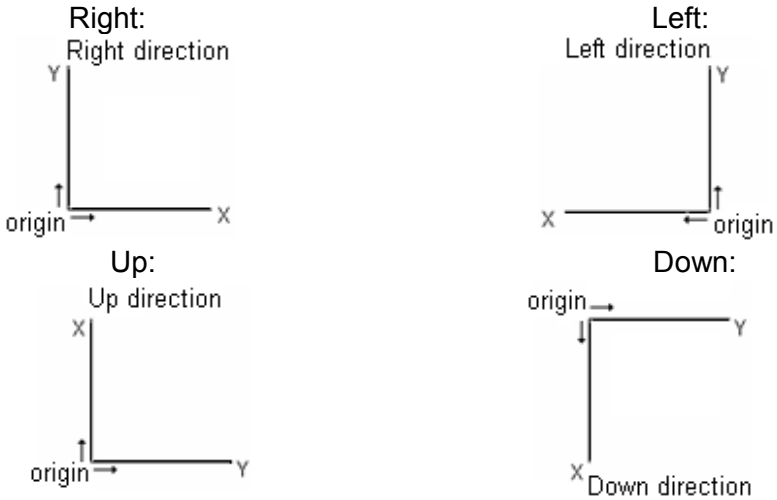


Щелкните по пиктограмме [XY plot] — появится окно [New XY plot object].



General

a. **Direction:** четыре варианта выбора направления системы координат: right, left, up или down.



b. No. of channel.
Установка числа кривых трендов на графике. Каждый канал отдельно может управлять процессом рисования.

Control address

[PLC name]
Выберите ПЛК, откуда придет управляющий адрес.

Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** управляющего адреса.
Пользователи могут назначить адрес в закладке -*General* , при создании нового объекта.

[Device type]
Выберите тип данных управляющего адреса.

[Control address]
“Control address” используется для управления кривой тренда каждым каналом.

1 = *Вывести на экран кривую тренда XY.*
Если «1» записывают для управления выводом XY-графика, в случае наличия предыдущей кривой, она останется. Система сбросит управляющий регистр после завершения операции.

2 = *Удалить кривую тренда XY*
Если «2» записывают в управляющий адрес, будут удалены все предыдущие кривые XY, и после завершения операции будет сброшен управляющий адрес.

3 = *Обновить кривую тренда XY*
Если «3» записывают в управляющий адрес, система удалит предыдущую кривую XY, выведет на экран новую и сбросит управляющий регистр после завершения операции.

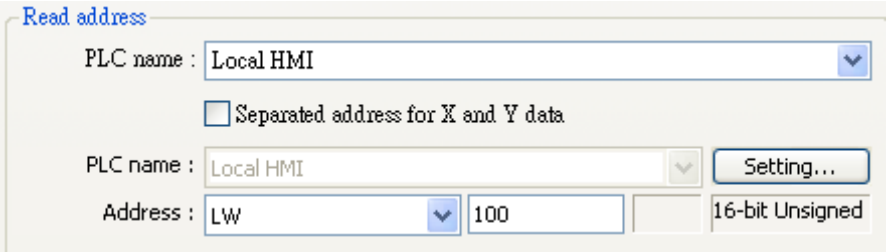
[No. of data address]
Данный адрес хранит число данных кривой XY. Каждый канал может иметь до 1023 данных.

Channel Установка атрибутов для отображения графиков.

Read Address
[PLC name]
Введите название устройства, где размещается управляющий адрес.

Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** управляющего адреса.
Пользователи могут назначить адрес в закладке –*General* , при создании нового объекта.

[PLC address]



Нажмите кнопку [Settings…] чтобы выбрать **[PLC name], [Device type], [Address], , [Index register]** для чтения адреса.

✹ Использование каждого адреса следующее (*Dynamic limits* – **не** выбраны)
Например:
Адрес чтения - Read address - LW100.

X data 0 - Читает значение из адреса -LW100.

X data 1 - Читает значение из адреса -LW101.

X data 2 - Читает значение из адреса -LW102.

X data 3 - Читает значение из адреса -LW103.

X data 4 - Читает значение из адреса -LW104.

X data 5 - Читает значение из адреса -LW105 и так далее…

✹ Использование каждого адреса следующее (*Dynamic limits* – выбраны)

Например:

Адрес чтения - Read address - LW100.

X low limit - Читает значение из адреса -LW100.

X high limit - Читает значение из адреса -LW101.

Y low limit - Читает значение из адреса -LW102.

Y high limit - Читает значение из адреса -LW103.

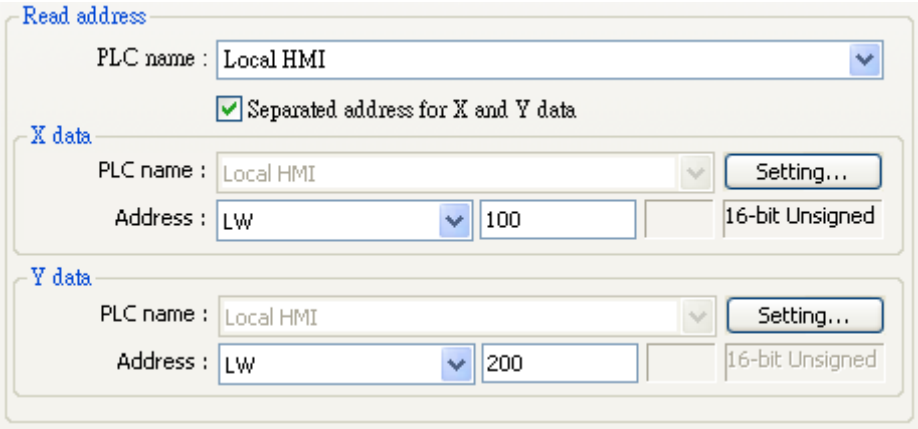
X data 0 - Читает значение из адреса -LW104.

Y data 0 - Читает значение из адреса -LW105.

X data 1 - Читает значение из адреса -LW106.

Y data 1 - Читает значение из адреса -LW107.

Если отмечен флаг “Separated address for X and Y data” , то это позволит назначить разные адреса для осей (X) и (Y) соответственно.



✹ Использование каждого адреса следующее (*Dynamic limits* – **не** выбраны)

Например:
Адреса чтения - Read address - LW100 и LW200.

<u>X data</u>

X low limit - Читает значение из адреса -LW100.

X high limit - Читает значение из адреса -LW101.

X data 0 - Читает значение из адреса -LW102.

X data 1 - Читает значение из адреса -LW103.

X data 2 - Читает значение из адреса -LW104.

X data 3 - Читает значение из адреса -LW105 и так далее…

<u>Ydata</u>

Y low limit - Читает значение из адреса -LW200.

Y high limit - Читает значение из адреса -LW201.

Y data 0 - Читает значение из адреса -LW202.

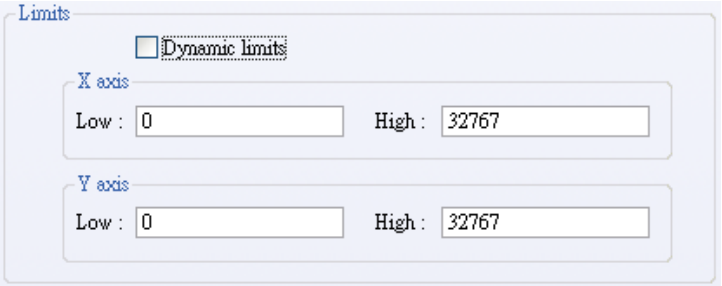
Y data 1 - Читает значение из адреса -LW203.

Y data 2 - Читает значение из адреса -LW204.

Y data 3 - Читает значение из адреса -LW205 и так далее…

Limits

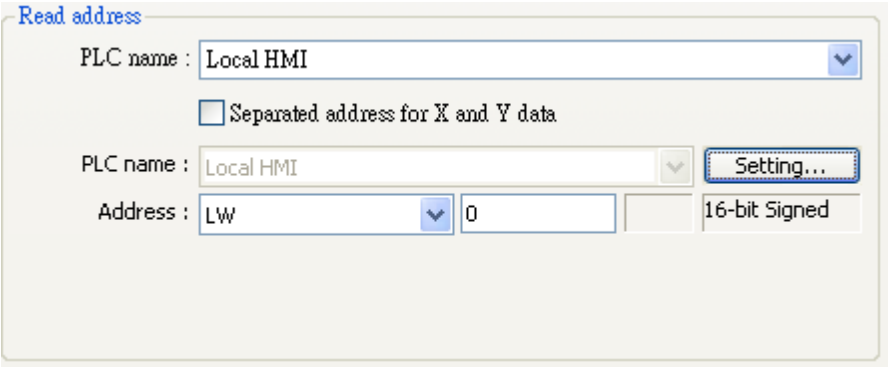
Настройки выше используют динамические пределы,но можно также установить фиксированные верхний и нижний пределы.



Верхний и нижний пределы используются для масштабирования при вычислении координат (в процентах) по X и Y, т.е. X или Y%=(X or Y reading value – low limit) / (high limit – low limit)

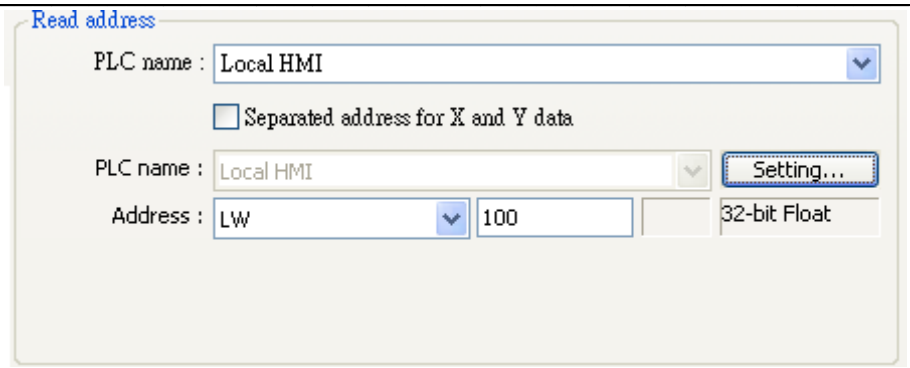
На основе выполненных настроек, распределение памяти для хранения данных XY будет следующим.

Ниже приведены настройки для 16--битового формата чисел со знаком и динамических пределов.



X low limit - Читает значение из адреса - LW0.(n+0)
X high limit - Читает значение из адреса - LW1. (n+1)
Y low limit - Читает значение из адреса - LW2. (n+2)
Y high limit - Читает значение из адреса - LW3. (n+3)
X data 0 - Читает значение из адреса - LW4. (n+4)
Y data 0 - Читает значение из адреса - LW5. (n+5)

Настройки для 32-битового формата с плавающей запятой и динамических пределов.



X low limit - Читает значение из адреса - LW100.(n+0)

X high limit - Читает значение из адреса - LW102. (n+2)

Y low limit - Читает значение из адреса - LW104. (n+4)

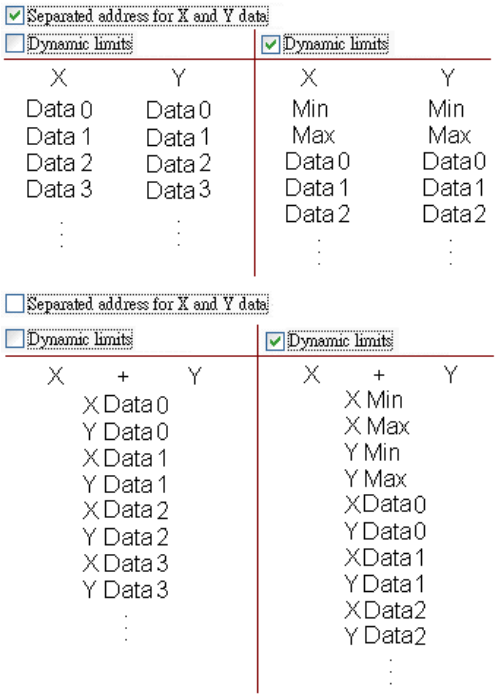
Y high limit - Читает значение из адреса - LW106. (n+6)

X data 0 - Читает значение из адреса - LW108. (n+8)

Y data 0 - Читает значение из адреса - LW110. (n+10)

ПРИМЕЧАНИЕ:

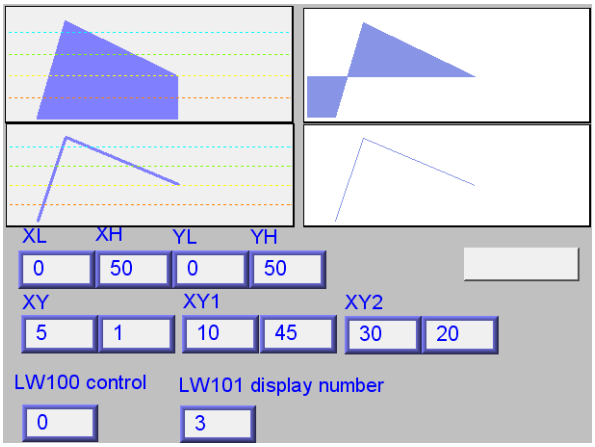
Есть четыре типа выбора назначения памяти для верхнего/нижнего пределов и данных по осям XY. См. различные способы сочетания настроек ниже.



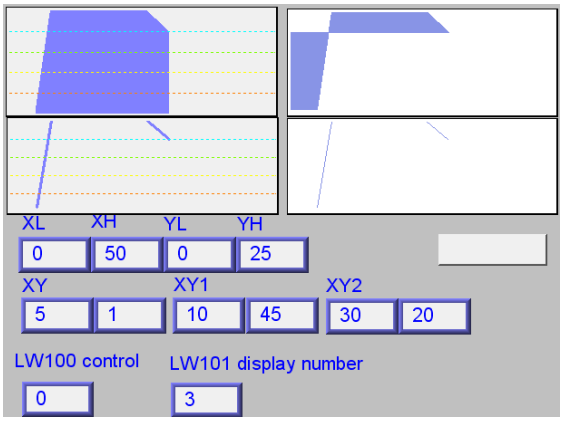
Если включена функция [Dynamic limits], можно изменять значения верхнего и нижнего пределов. (Подробнее о функции [Dynamic limits] в Главе «Trend display».)

В примере далее выбрана функция [Dynamic limits], где XL=X low limit (нижний предел по X), XH= X high limit (верхний предел по X), YL=Y low limit (нижний предел по Y), YH=Y high limit (верхний предел по Y) и три точки графика: XY, XY1, XY2. Затем мы изменяем верхние пределы по X и Y соответственно — наблюдаем эффект увеличения или уменьшения (zoom in

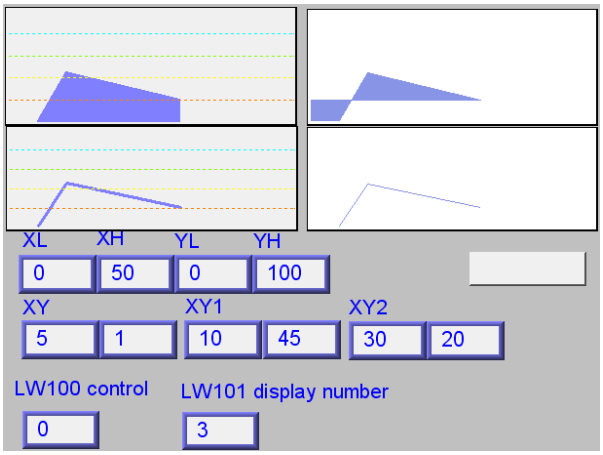
/ zoom out).



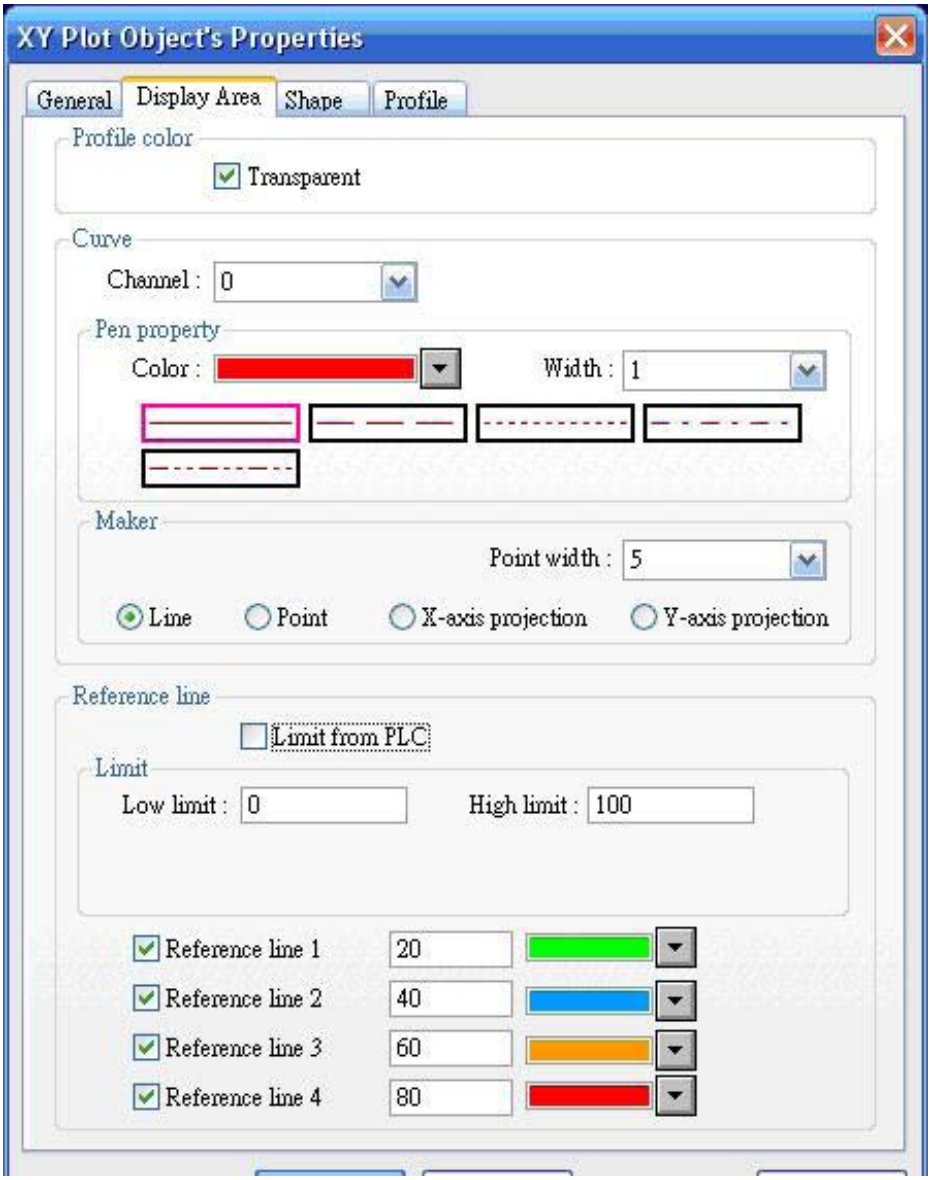
1. Изменим верхний предел по «Y high limit» на 25 для масштаба увеличения.



2. Изменим верхний предел «Y high limit» на 100 для эффекта увеличения.



Закладка [Display Area]



Profile color

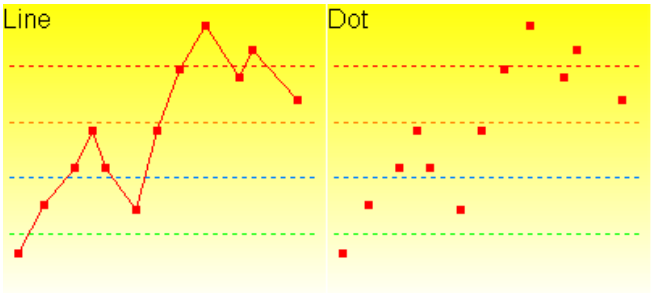
Enable Transparent: Фон не будет отображаться.
Disable Transparent: Фон будет отображаться

Curve Установка свойств каждой кривой XY (цвет и толщину линии).

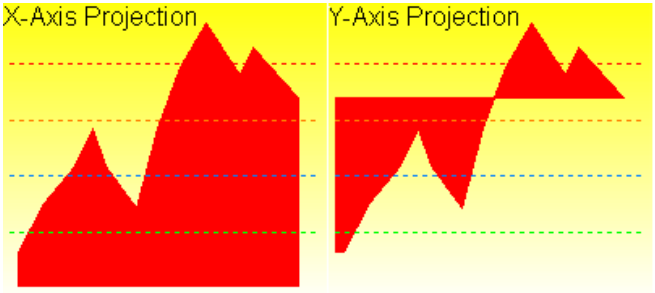
Maker - Построитель

Есть 4 типа линий: Line, Point, X-axis projection и Y-axis projection.
Для типов Line и Point задайте ширину (в пикселях).

Line и Point:



X-axis projection (проекция на ось X) и Y-axis projection (проекция на ось Y):

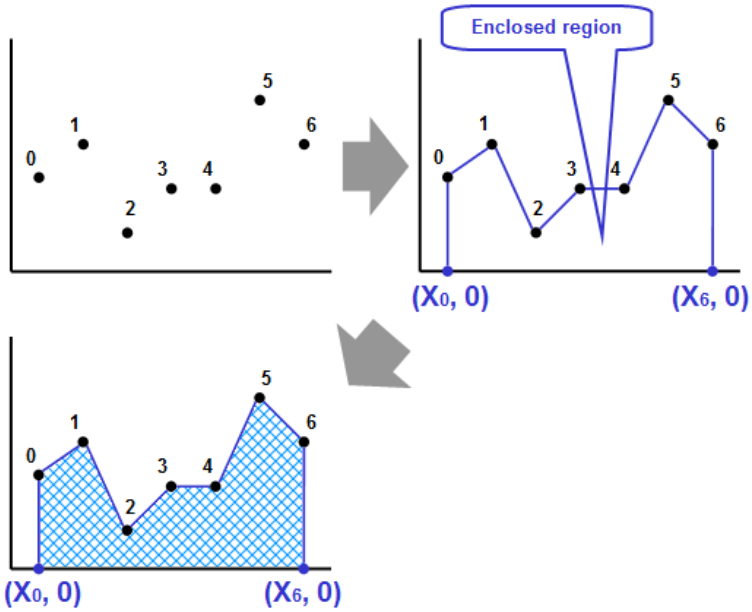


Замечания:

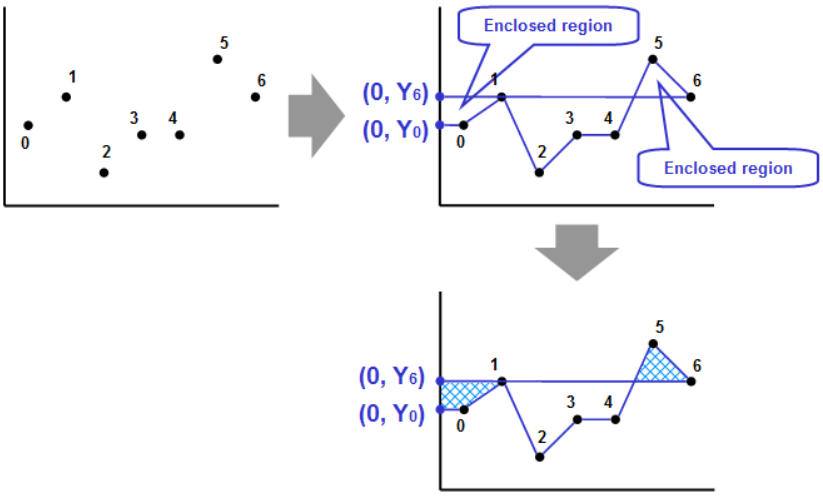
Посмотрите на рисунок внизу - кривая содержит точки P0, P1, …, P6. Система:

1. Автоматически вычисляет две точки проекции на ось X: (X0, 0) и (X6, 0).
2. Соединяет все точки в порядке от (X_0, 0), P0, P1, …, P6, (X_6,0) и происходит возврат в (X_0, 0).
3. Закрашиваются все замкнутые участки под кривой.

X-axis projection :

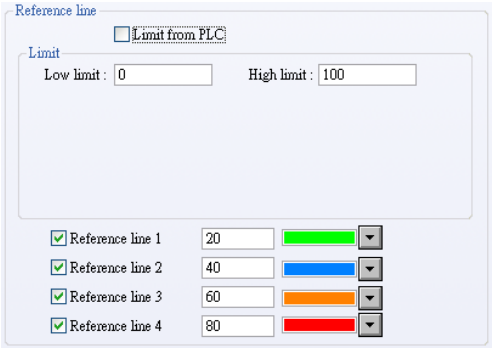


Аналогично для Y-axis projection:

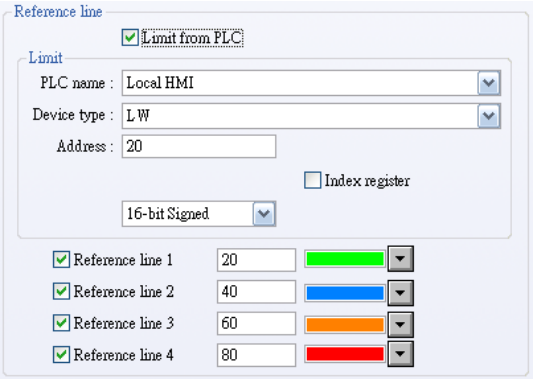


Reference line – Реперная линия

Для того, чтобы график сделать более наглядным, можно настроить до 4-х горизонтальных реперных линий. Задайте верхний и нижний пределы и координату по оси Y для каждой такой линии.



Можно также указать адрес, откуда брать верхний и нижний пределы.



ПРИМЕЧАНИЯ:

- *XY Plot* может быть начерчен повторно 32 раза:

  1 канал ➔ 32 раз

  2 канал ➔ 16 раза

Способ вычисления: «32» разделить на число каналов

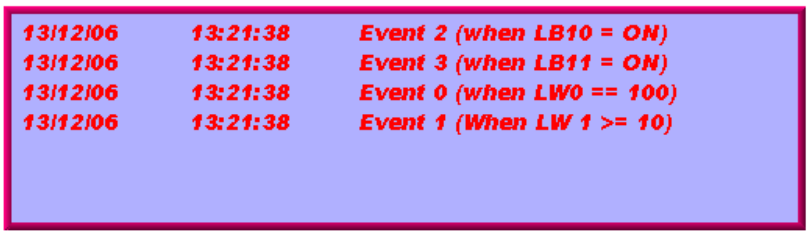
13.21 Alarm Bar и Alarm Display – Бегущая строка и дисплей аварийных сообщений

Краткое описание

Объекты [Alarm Bar] и [Alarm Display] используются для отображения аварийных сообщений (алармов). Аварийные сообщения настроены в [event log] («Регистрация событий») и возникают при выполнении предопределенных условий. Объекты [Alarm Bar] и [Alarm Display] отображают данные сообщения в порядке приоритетов и времени возникновения.

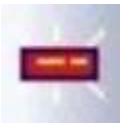
Объект [Alarm Bar] показывает все сообщения-алармы в виде бегущей строки, [Alarm Display] отображает каждое сообщение в отдельной строке. На рисунке ниже показано, как это выглядит. См. также главу «Архив событий - event log».

Alarm bar object

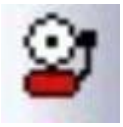


Alarm display object

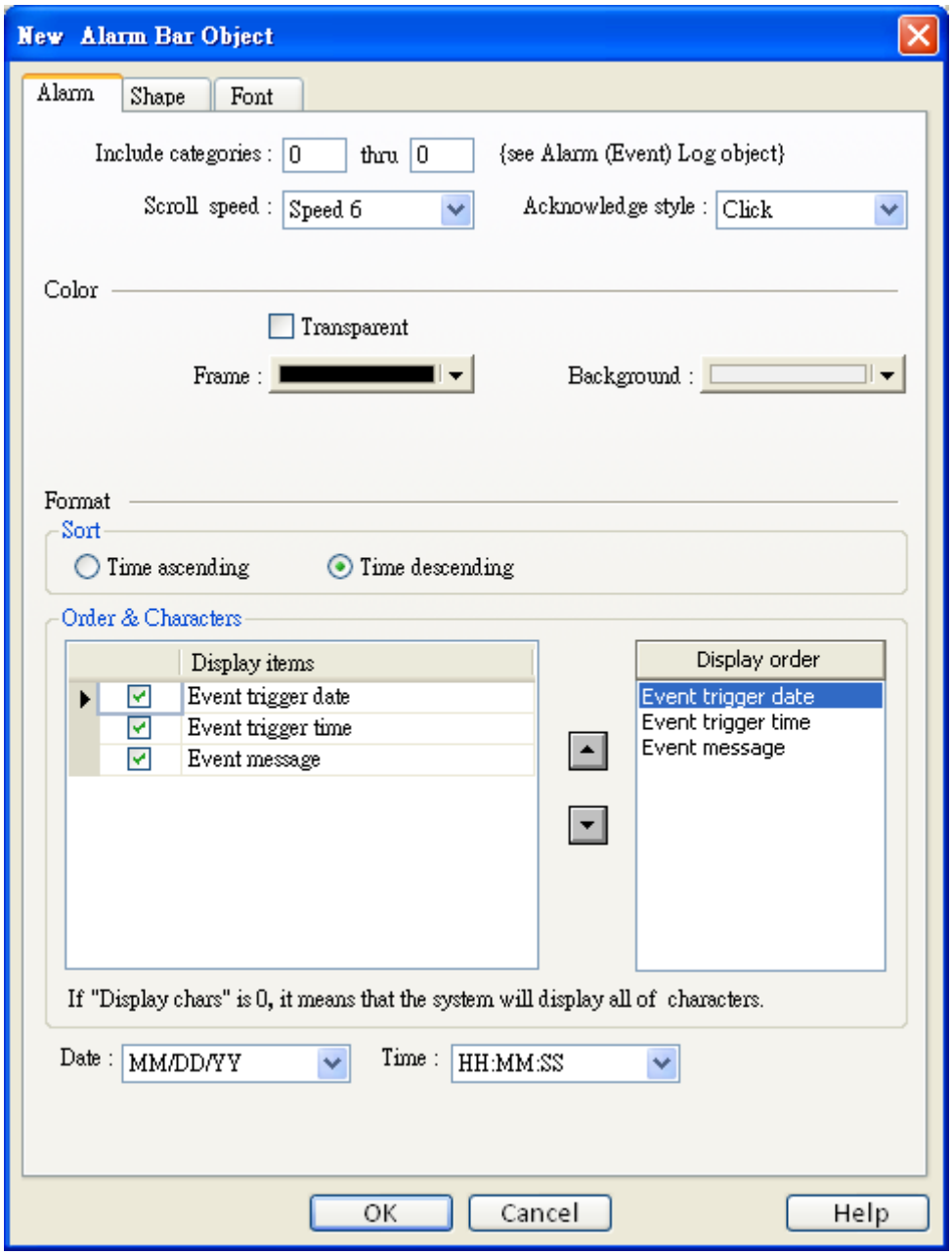
Настройка свойств



Щелкните по пиктограмме [Alarm bar] на панели инструментов — появится диалоговое окно [New Alarm bar Object];



Щелкните по пиктограмме [Alarm display] на панели инструментов, заполните поля все параметры на закладке [General] и нажмите OK — будет создан новый объект. См. рисунок далее.



Include categories Выберите категорию события, которое связано с данным объектом (категорияи событий определяются в таблице [event log]). Например, если категория события для объекта [alarm bar] установлена от 2 до 4, будут отображаться все события с параметром «категория» , равным 2, 3 или 4. См. описание параметра [Category] в главе «Архив событий».

Scroll Speed Задайте скорость перемещения строки объекта [alarm bar].

Color Установите цвет рамки и фона [alarm bar].

Format

a. Sort
Установка порядка отображения сообщений.

Time ascending
Расположить события в порядке возрастания времени возникновения.

Time descending
Расположить по убыванию времени возникновения.

b. Order & Characters

Пользователь может выбрать элементы и установить порядок отображения элементов.

c. Date (Event trigger date)
Отображать метку даты с сообщением. Есть четыре формата метки даты.

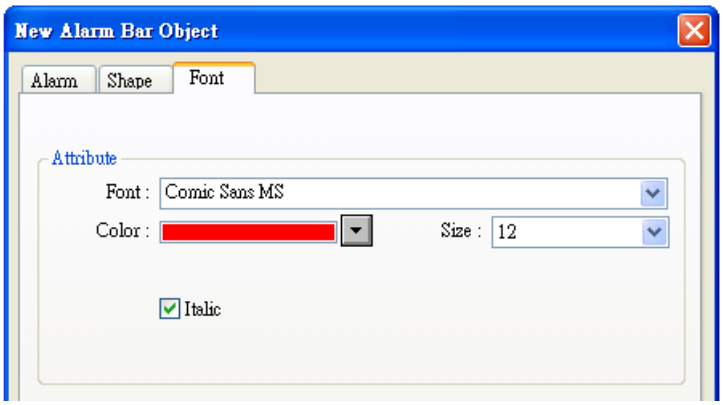
1. MM/DD/YY / 2. DD/MM/YY / 3. DD.MM.YY / 4. YY/MM/DD

d. Time (Event trigger time)

Отображать метку времени вместе с сообщением. Есть три формата временной метки.

1. HH:MM:SS / 2. HH:MM / 3. DD:HH:MM

Установить шрифт и его цвет можно на закладке [Font]. См. рисунок ниже.



13.22 Event Display – Отображение событий

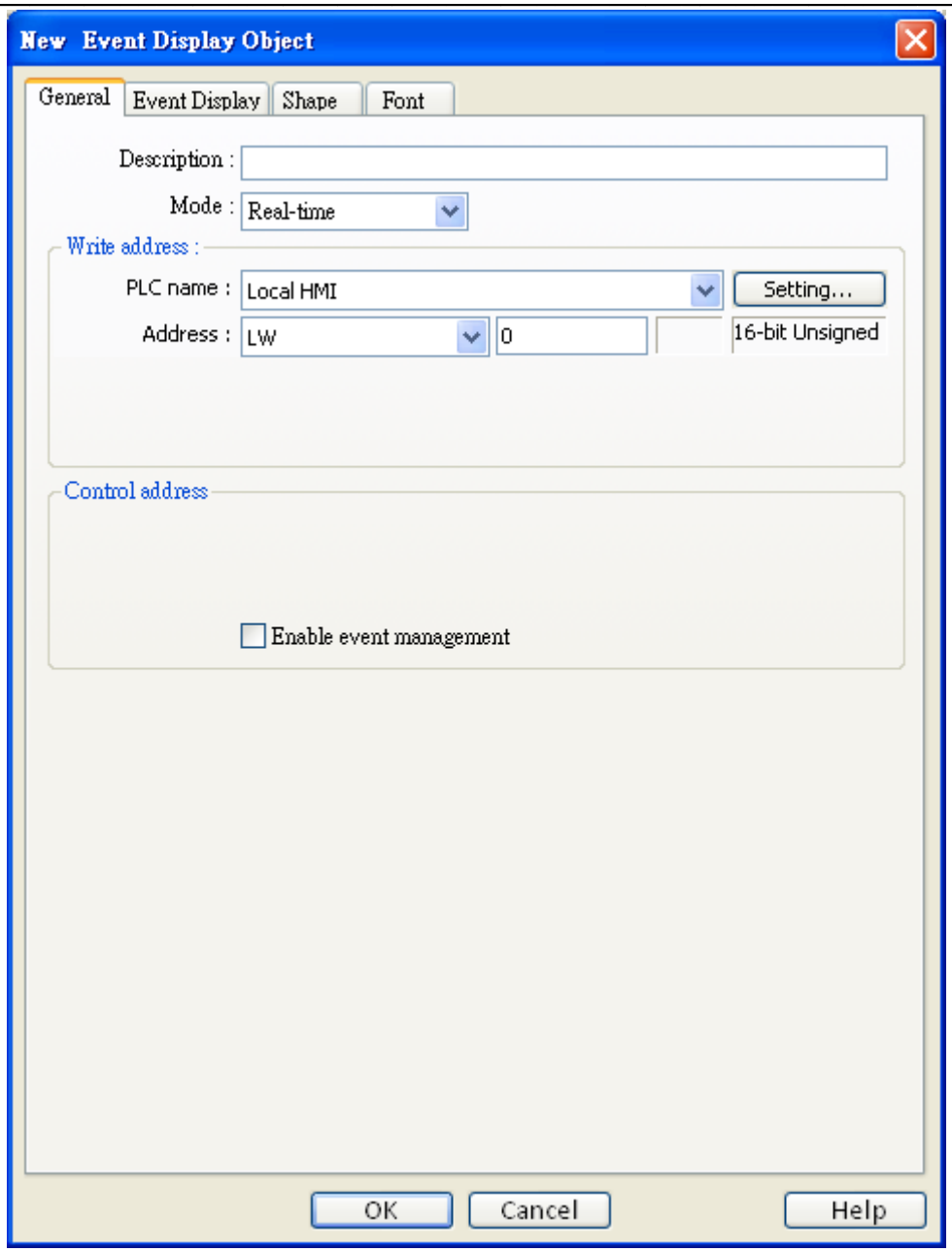
Краткое описание

Объект [Event Display] отображает активные и закончившиеся события. События записываются в объекте [Event log]. Активные сообщения – это те, для которых в данный момент выполняются условия запуска или же запущенные и не квитированные.

Объект [Event Display] отображает эти активные события в порядке времени возникновения. См. рисунок ниже. Данный объект может также отображать время возникновения события, квитирования и восстановления.

Настройка свойств

Щелкните по пиктограмме [Event Display] на панели инструментов — появится окно [New Event Display Object], задайте все параметры на закладке [General], нажмите OK — будет создан новый объект. См. рисунок далее.



[Mode]

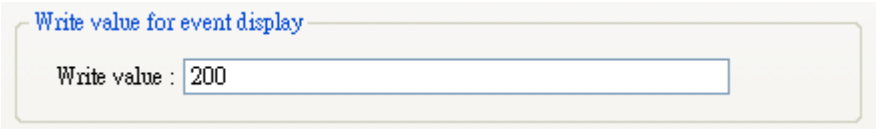
Выберите источник события, есть варианты “Real-time” и “History”.

a. Real-time

Write address

События отображаются от пуска панели до конца существования.

Когда событие квитировано, содержимое [Alarm (Event) Log]/ [Message]/ [Write value for Event Display object] будет экспортировано по адресу [write address] объекта [event display].



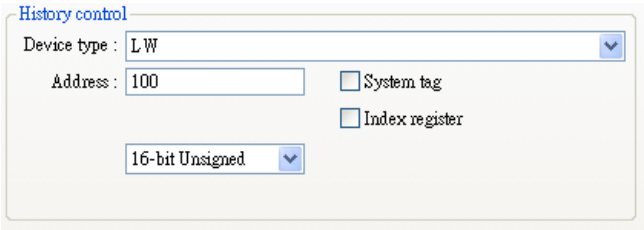
b. History Control

✹ [Enable reading multiple histories] – Разрешить чтение нескольких файлов архивов

- **не выбрано**.

В этом режиме объект будет показывать записи событий из архива. Система хранит историю событий в ежедневной базе, записи событий хранятся в отдельных файлах по дням с меткой даты. “History control” используется для выбора одного архивного файла.

Рисунок ниже показывает настройки “History control”, которые определяют адрес управления архивом.



Система выбирает архивную запись по индексу. Индекс [index 0] относится к самой последней архивной записи (обычно запись текущего дня), [index 1] относится к записи предыдущего дня и т.д.

Текущее значение по адресу “History control” используется в качестве начального адреса для выбора нужной архивной записи.

Ниже приведен пример для пояснения как использовать функцию “History control”. Адрес задан как [LW100], предположим, что записи, сохраненные в системе следующие:

EL_20061120.evt,
EL_20061123.evt,
EL_20061127.evt
EL_20061203.evt,

Где 2006xxxx в названии соответствует дате сохранения данной записи. Данная ниже таблица показывает какие архивы отображает объект [event display] в зависимости от значения по адресу [LW100].

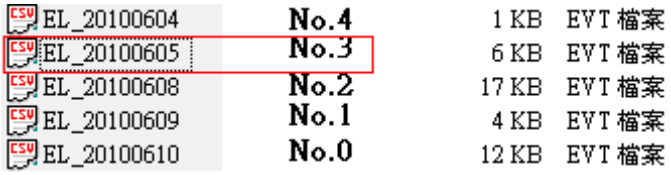
Значение по адресу [LW100]	Соответствующая архивная запись
0	EL_20061203.evt
1	EL_20061127. evt
2	EL_20061123. evt
3	EL_20061120. evt

✹ [Enable reading multiple histories] – Разрешить чтение нескольких файлов архивов
- **выбрано**.

Определение: Отображение списка событий созданных за несколько дней.

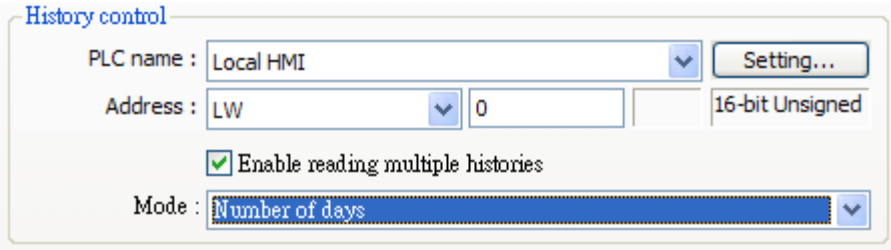
Иллюстрация: Например, «LW0» это **[History Control] [Address],** диапазон отображаемых данных будет формироваться «LW0» и «LW1». При этом «LW0» представляет начальные данные архива.

Пример: Как показано ниже (для упрощения архивные данные пронумерованы: No.0、No.1、No.2…, в соответствии с датой получения), если значение LW0 это “3”, первыми отображаемыми данными будут данные No. 3.



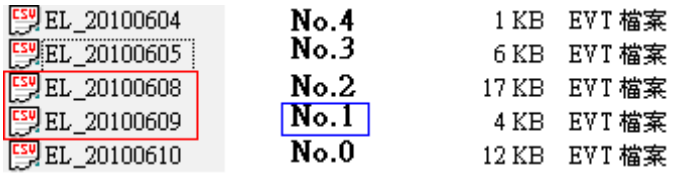
При помощи «LW1», может быть выбрано 2 режима.

a. Number of days – Число дней

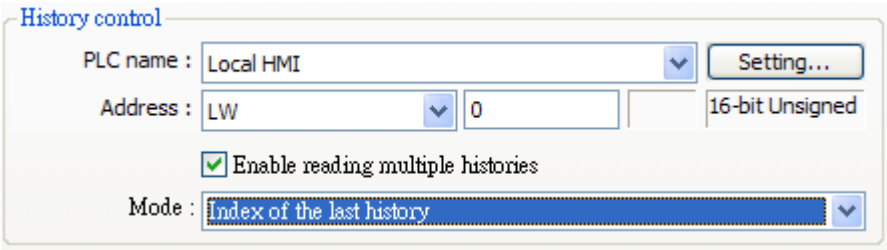


Диапазон «History Data»- Архивных данных для отображения будет начинаться с номера LW0. Значение LW1 представляет сколько дней будет включено, начиная с первого дня и назад.

Пример: Как показано ниже, LW0 это “1”, LW1 это “3”, диапазон данных будет начинаться с 20100609, и включать данные за 2 дня до этого (с учетом 20100609).
Мы можем видеть в этом примере, что данные 20100607 не существуют, отображены данные только 20100609 и 20100608.

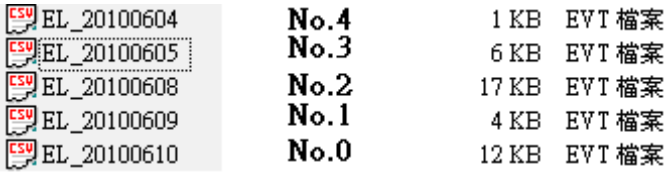


b. Index of the last history – Индекс последнего архивного файла



Диапазон данных для отображения будет начинаться с номера в LW0 и заканчиваться номером в LW1.

Пример: Как показано ниже, LW0 это “1”, LW1 это “3”, диапазон данных будет начинаться с «No. 1», и включать данные за 3 дня (No.1, No.2, No.3).



Максимальный размер данных, который может быть отображен системой – 4 Мбайта, остальные данные будут проигнорированы.

Далее показано, как будут сохранены данные, если размер данных слишком большой.

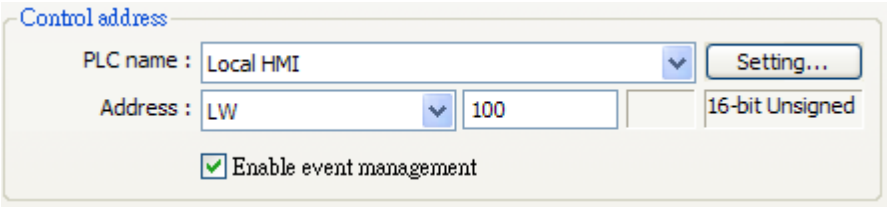
Пример:

a. 5 history data, each with a size of 0.5MB ☐The size of data to be displayed will be 5 x 0.5MB
b. 5 history data, each with a size of 1MB ☐The size of data to be displayed will be 4 x 1MB
c. 5 history data, each with a size of 1.5MB ☐The size of data to be displayed will be 2 x 1.5MB+1 x 1MB (partial)

Определение:

1. To select confirmed or recovered events to be displayed or hidden.
2. In **[Real- time]** mode, select events to be deleted.

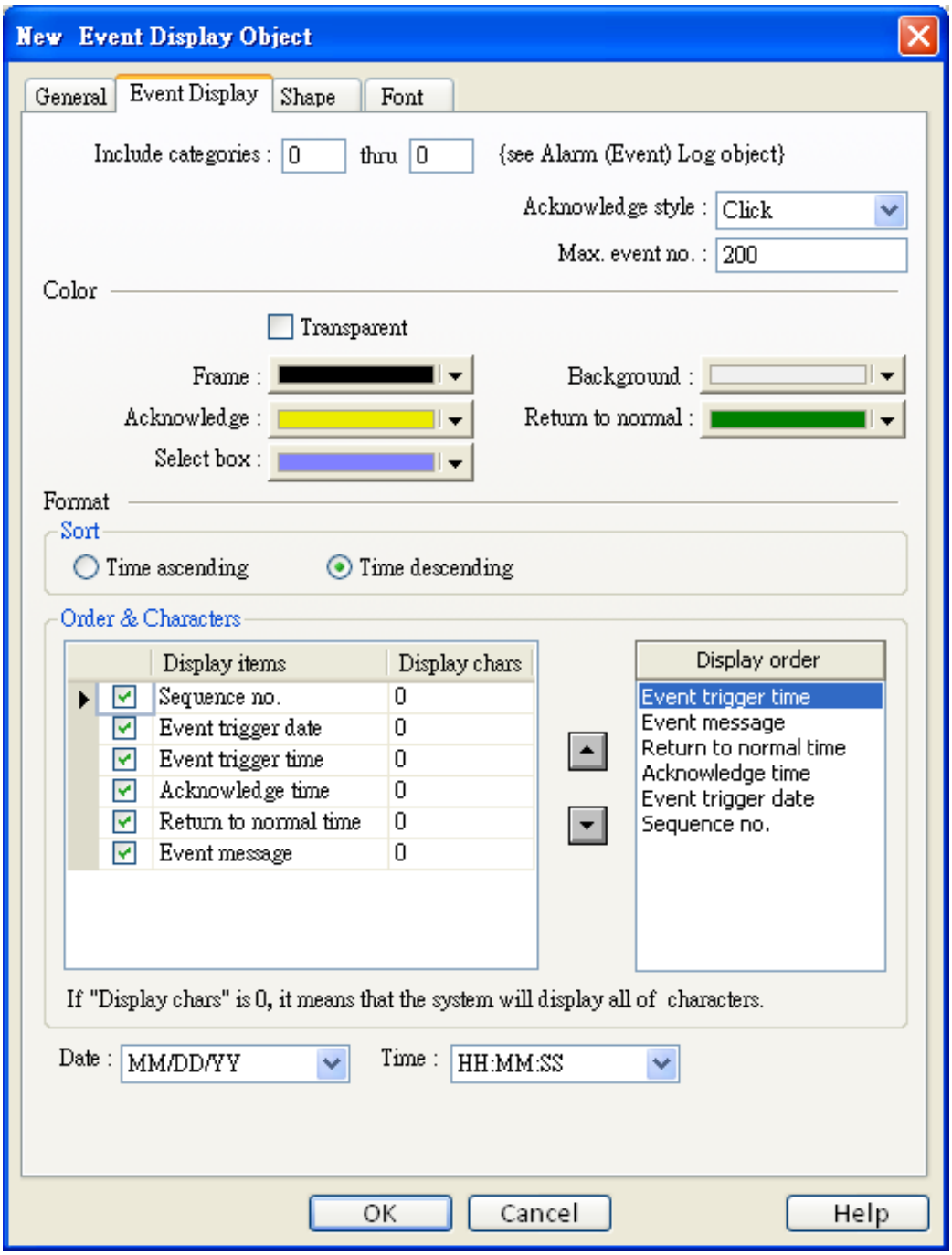
Иллюстрация:



Если в значении «History control» указано LW100:

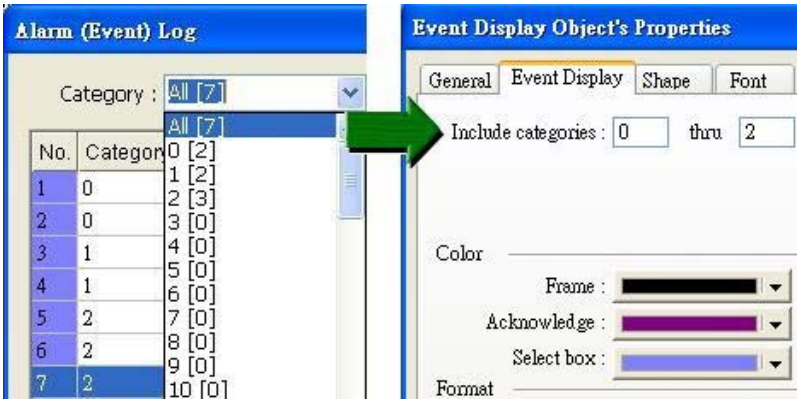
1. Когда значение в [LW100+0] - “0” → Будут отображены все события.
2. Когда значение в [LW100+0] - “1” → Подтвержденные события будут удалены.
3. Когда значение в [LW100+0] - “2” → Восстановленные значения будут спрятаны.
4. Когда значение в [LW100+0] - “3” →Подтвержденные и Восстановленные значения будут спрятаны.

Когда значение в [LW100+1] is “1” → Пользователи удалитьвыбранные события в режиме реального времени [real-time].



[Include categories] - Включить категории

Выберите категории событий, связанных с данным объектом [event display] (категорииия задается в [event log]).
Например, если категории событий, связанных с [event log] заданы в диапазоне от 2 до 4, то будут отображаться сообщения о событиях категории 2, 3 или 4.
См. информацию о [Category] в главе «Архив событий».



[Acknowledge style] – Способ квитирования (подтверждения)

Вы можете выбрать варианты: “Click” (Щелчок) или “Double click” (Двойной щелчок) для квитирования (подтверждения оператором) события. Когда событие произошло, оператор может одинарным или двойным щелчком подтвердить событие, цвет сообщения будет изменен и в назначенный адрес будет занесено значение “write value”.
Используя данную функцию, можно настроить рабочее окно и поместить в него сообщение предупреждения, затем настроить объект [indirect window], и когда аларм будет квитирован, значение “write value” запишется в адрес [read address] объекта [indirect window] и вызовет рабочее окно.

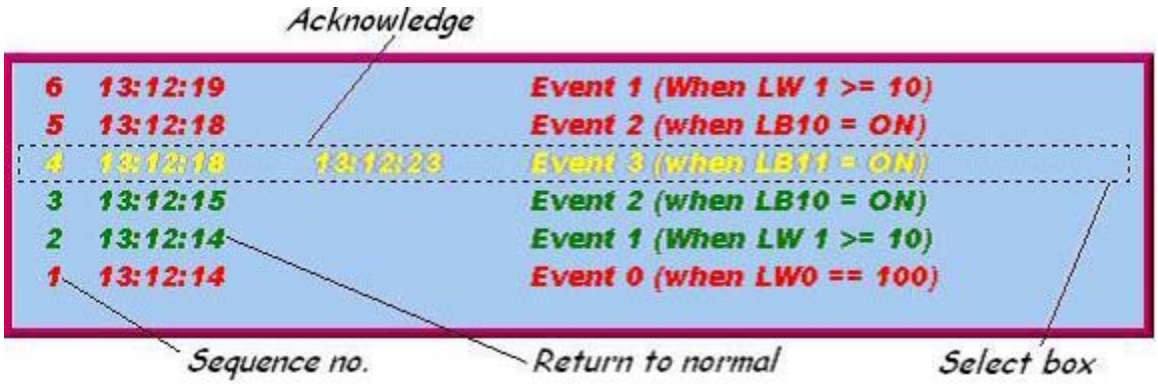
[Max. event no.]

Максимальное число событий, отображаемых в объекте [Event Display]. Когда число событий больше максимального, самое раннее из них будет удалено из [event display].

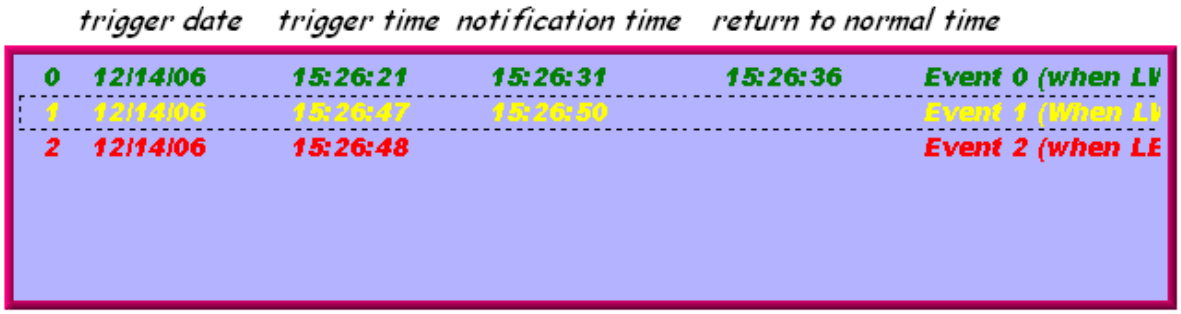
[Color]

Установка цвета строки событий для разных состояний:

a. **Acknowledge (Подтверждение)**
b. **Return to normal (Возврат в штатное состояние)**
c. **Select box —** будет отображаться подсвечиваемый прямоугольник вокруг последнего квитированного события.



Format



Sort
Выберите способ упорядочивания событий.

[Time ascending]
По возрастанию времени возникновения.

[Time descending]
По убыванию времени возникновения.

b. Order & Characters
Пользователь может выбрать элементы и установить порядок отображения элементов.

Date [Event trigger date]

Отображать тэг даты вместе с аварийным сообщением.

Есть четыре формата метки даты.

1. MM/DD/YY / 2. DD/MM/YY / 3. DD.MM.YY / 4. YY/MM/DD

Time [Event trigger time]

Отображать тэг времени вместе с аварийным сообщением.

Есть три формата временной метки.

1. HH:MM:SS / 2. HH:MM / 3. DD:HH:MM

На закладке [Font] задаются высота шрифта и курсивное начертание. Стиль шрифта сообщения устанавливается в объекте [event log].

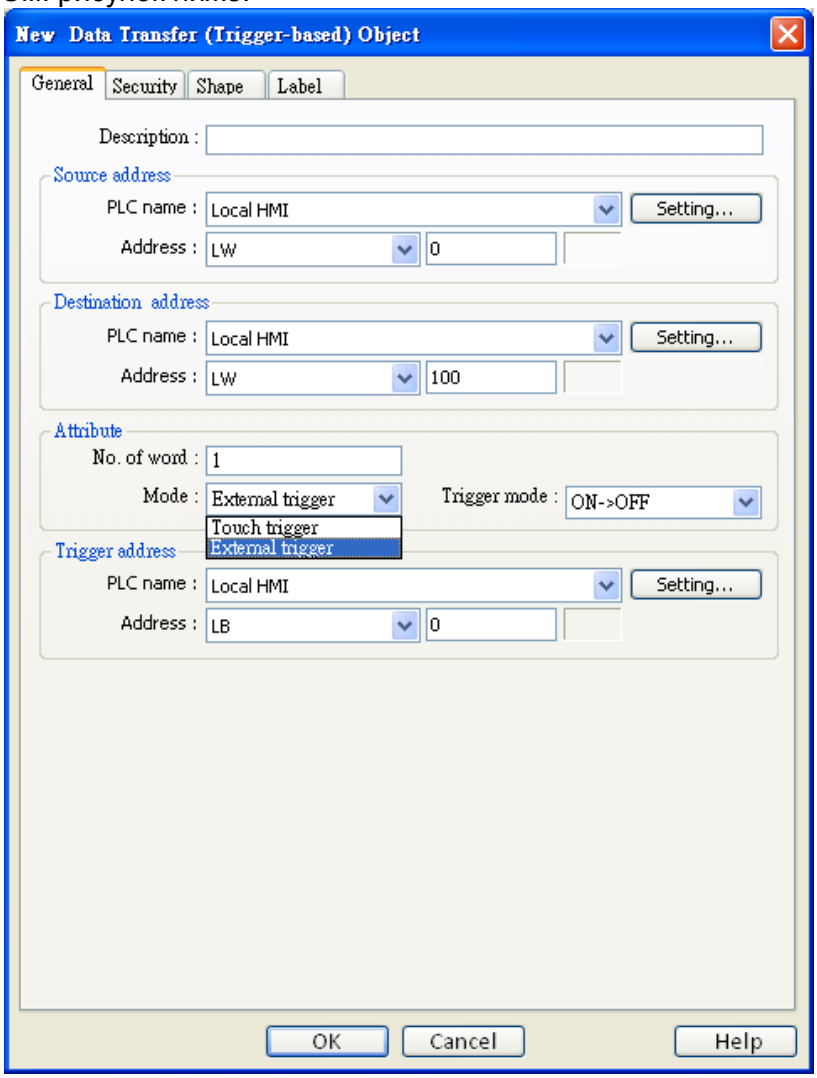
13.23 Data Transfer (Trigger-based) – Перенос данных по событию

Краткое описание

Объект [Data Transfer (Trigger-based)] может перемещать значение из одного адреса в назначенный. Данная операция перемещения может быть активирована нажатием на объекте или установкой бита запуска.

Настройка свойств

Щелкните пиктограмму [Data Transfer (Trigger-based)] на панели инструментов, откроется диалоговое окно [New Data Transfer (Trigger-based) Object], установите значения всех параметров на закладке [General], нажмите OK — будет создан новый объект. См. рисунок ниже.



Source address

Установка адреса-источника данных.

Destination address

Назначенный адрес, куда перемещаются данные.

Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** адреса источника и приемника данных.
Пользователи могут назначить адрес в закладке –*General* , при создании нового объекта.

Attribute

[No. of words]

Число слов, которые должны быть перемещены от источника по назначенному адресу .

[Mode] - Выбор способа (режима) активации процесса перемещения данных

a. Touch trigger mode

Для активации операции перемещения достаточно нажатия на объект.

b. External trigger

Определить бит состояния для запуска операции перемещения.

[ON → **OFF]**

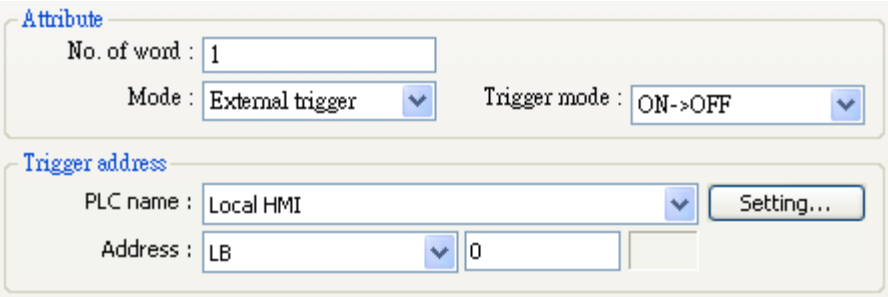
Состояние бита меняется с ON на OFF для активации перемещения.

[OFF → **ON]**

Состояние бита меняется с OFF на ON для активации перемещения.

[ON ←→ **OFF]**

Любое изменение состояния бита активизирует операцию перемещения.



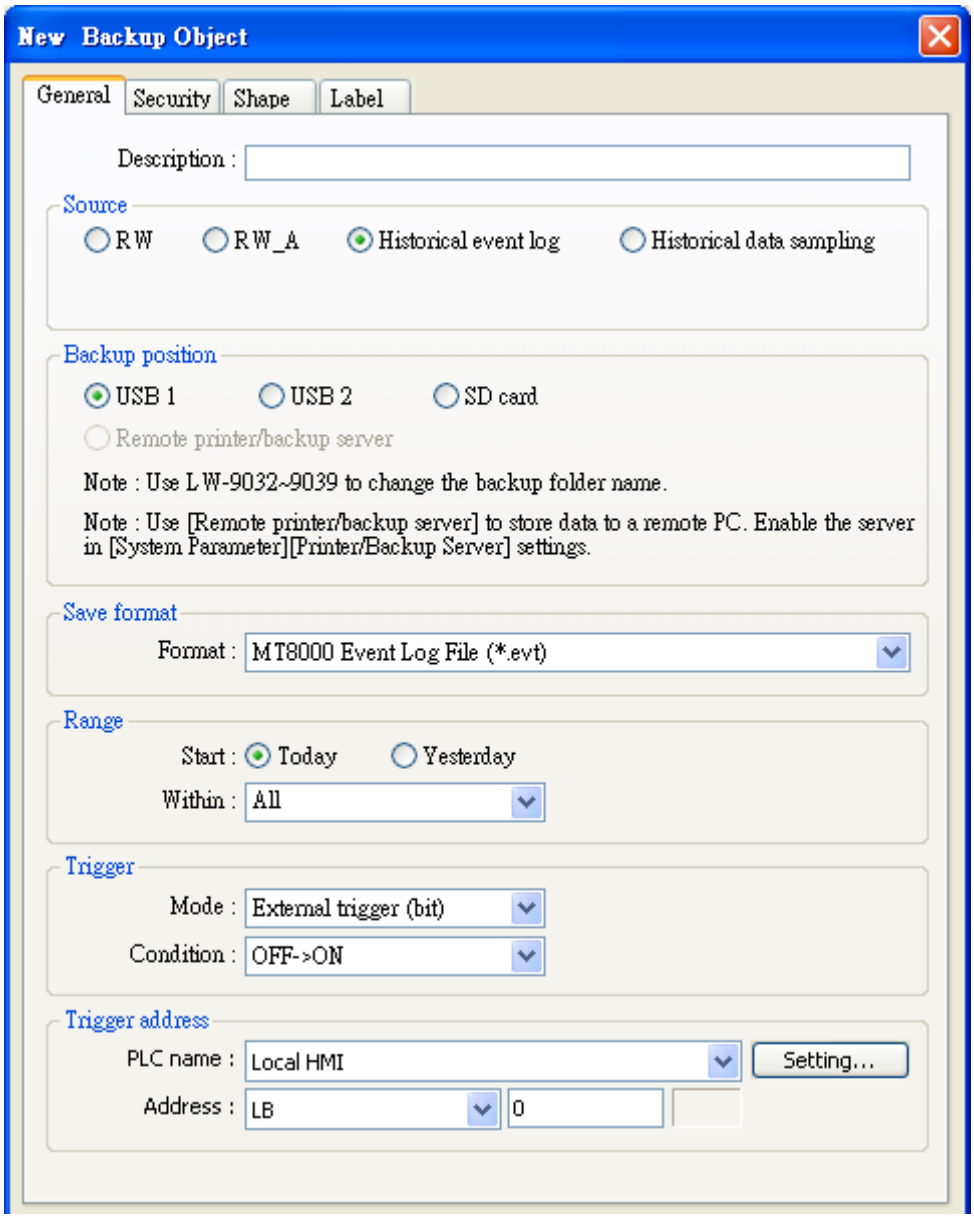
13.24 Backup - Резервное копирование

Краткое описание

Функция резервного копирования позволяет сохранять данные рецептов (RW, RW_A), записи событий и выборки данных на USB-накопителе или удаленном сервере (Remote backup server). Бит [LB-9039] указывает состояние копирования: когда операция в процессе, его значение равно ON.

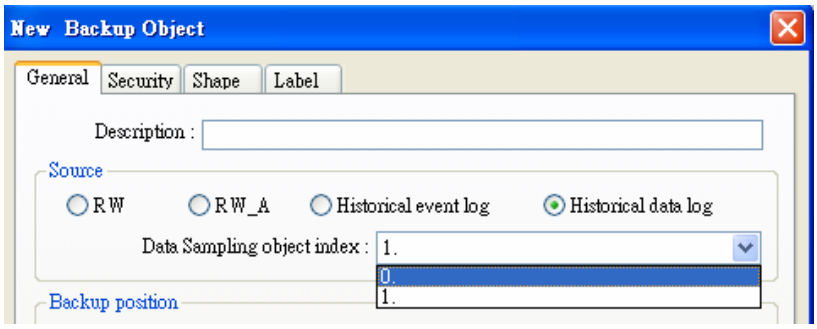
Настройка свойств

Щелкните пиктограмму [Backup Object] на панели инструментов — откроется диалоговое окно [New Backup Object]. См. рисунок ниже.



Source

[RW] [RW_A] [Historical event log], [Historical data log]
Выберите один из указанных выше источников данных. Это могут быть несколько выборок данных, находящихся в одном проекте. Если выбран [Historical data log], используйте раскрывающийся список “**Data Sampling object index**:” — для выбора конкретной записи, как показано ниже.



Backup Position – Позиция копирования

Назначьте запоминающее устройство, куда будут копироваться данные.

a. USB1 или USB2
USB-накопитель, присоединенный к панели .

b. Remote printer/ backup server – удаленный принтер/ сервер резервирования
Для того, чтобы данный вариант был доступен, нужно подключить удаленный сервер печати:
Menu → Edit → System Parameters → Printer/ backup Server

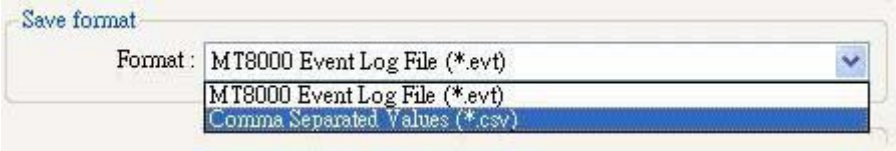
Save format

Пользователь может выбирать формат для резервирования данных.

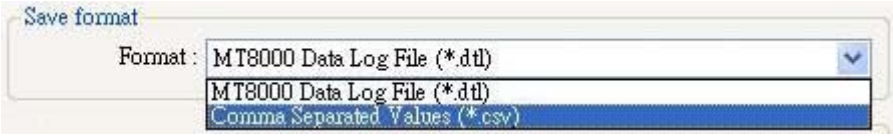
a. HMI Event Log File (*.evt) / HMI Data Log File (*.dtl)

b. Comma Separated Values (*.csv)

➢ **Event Log saved as csv file**

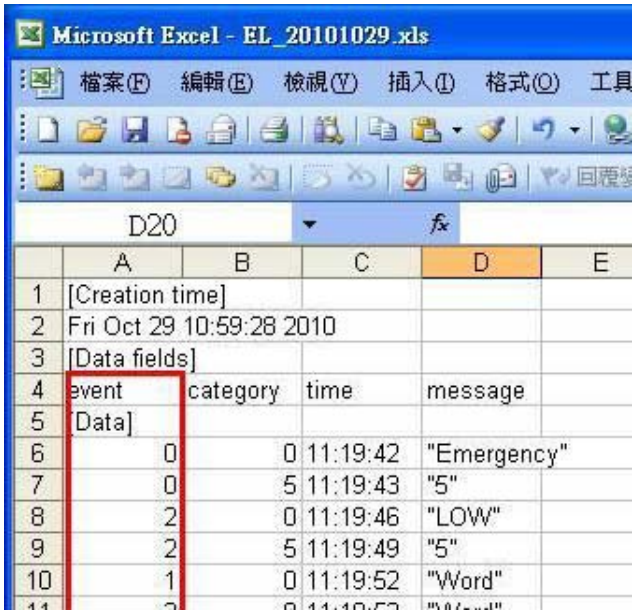


➢ **Data Log saved as csv file**



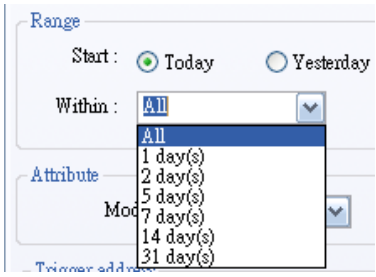
При резервировании архива событий в формате «**csv**», пользователь может просмотреть данные в EXCEL.

0 -> event is triggered – событие произошло
1 -> event is acknowledged – событие квитировано
2 -> event returns to normal – событие вошло в норму



Range - Диапазон

[Start]
Начать с сегодняшнего дня **[Today]** или вчерашнего **[Yesterday]**.



[Within] – В пределах
Выберите, за сколько дней копировать данные. Например, в [Start] указано [Yesterday] и выбрано “2 day(s)”. Это означает, что будут сохраняться файлы вчерашние и позавчерашние. Выберите вариант “All” для сохранения всех имеющихся в системе файлов.

Attribute
Существует два способа активации функции резервного копирования.

a. Touch trigger
Выберите объект для активации операции резервного копирования касанием.

b. External trigger
Использование внешнего бита для активации операции резервного копирования.

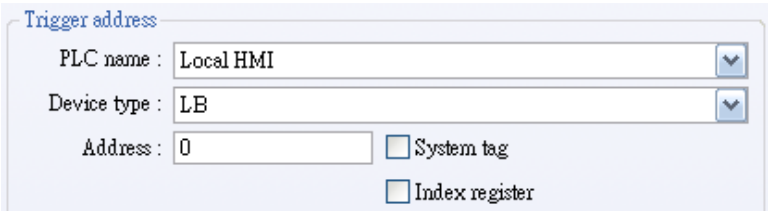
[ON → OFF] изменение состояния бита с ON на OFF

[OFF → ON] изменение состояния бита с OFF на ON

[ON ←→ OFF] любое изменение состояния бита

Trigger address

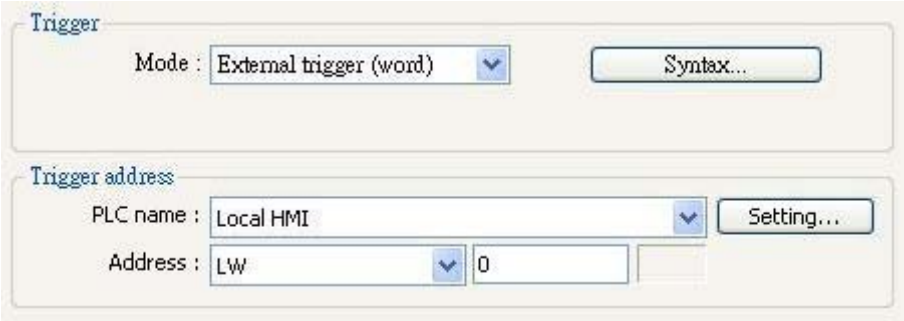
При использовании режима «External trigger» назначьте соответствующий бит, как показано ниже.



b. External trigger (word)

При выборе режима [External trigger (word)], пользователь может указать число дней резервного копирования данных, используя [Trigger address].

Использование внешнего бита для активации операции .



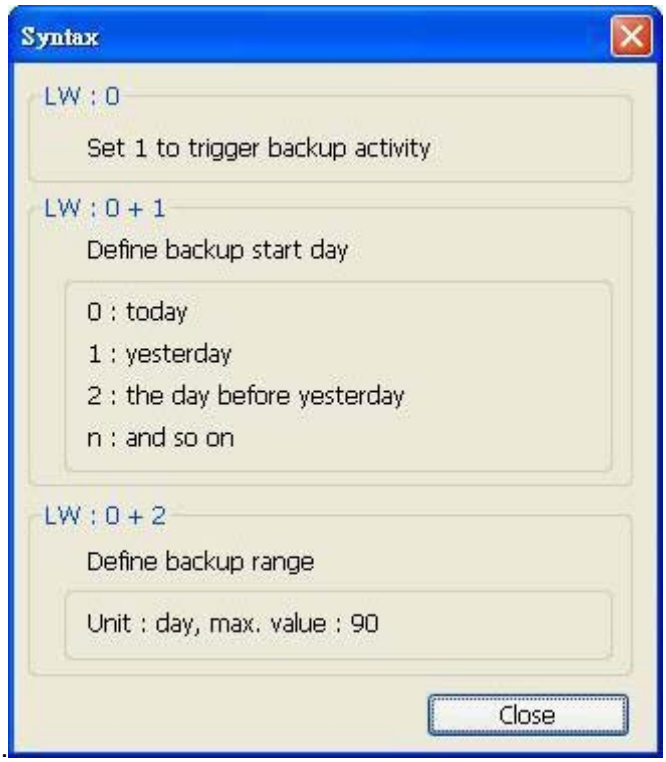
Использование [Trigger address] (предположим, что текущим адресом «Trigger Address» назначен LW-0) :

LW-0: Когда значение этого адреса изменяется с 0 в 1, запускается резервное копирование.

LW-1: Данные в этом адресе указывают начальную дату для резервного копирования.

LW-2: Данные в этом адресе указывают число дней для резервного копирования.

Синтаксис (Syntax) показан ниже:



13.25 Media Player - Проигрыватель

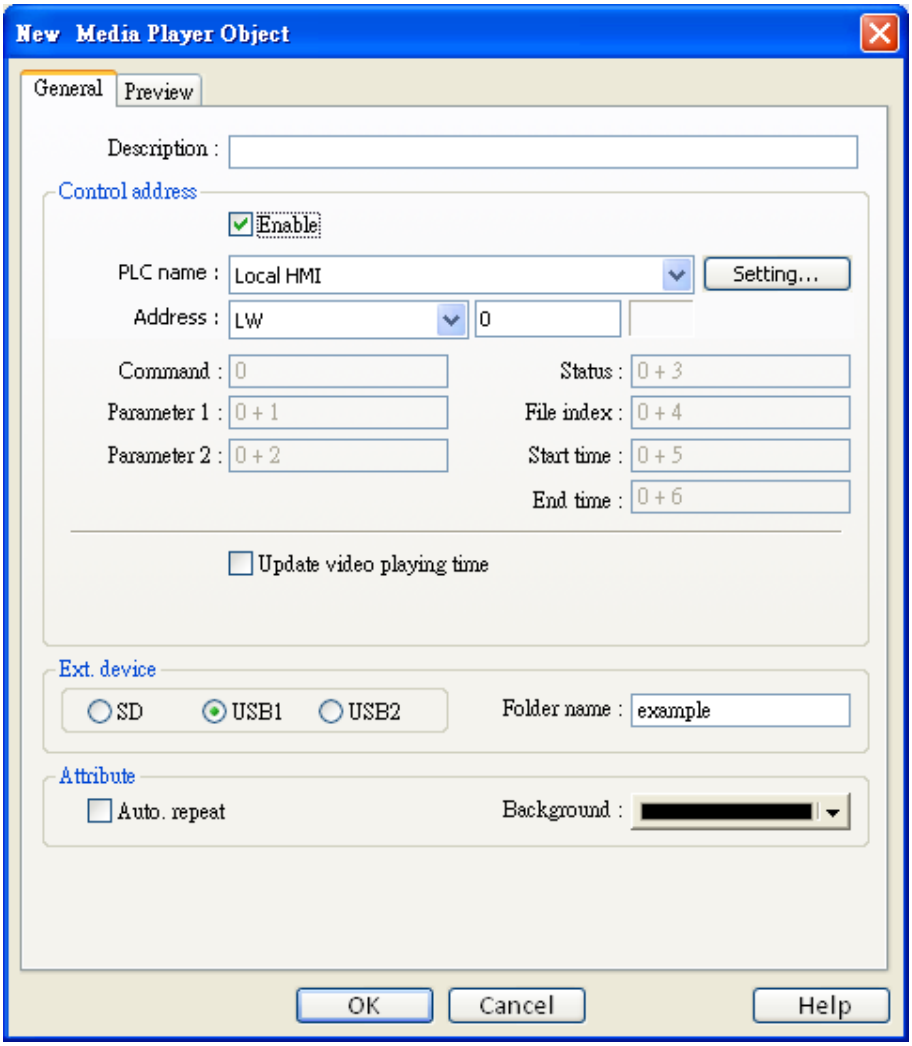
При первом использовании объектов типа [Media Player] нужно загрузить проект в панель оператора по сети Ethernet. Приложение EasyBuilder8000 установит Media Player во время загрузки.

Краткое описание

Media Player используется не только для воспроизведения видео-файлов, но также для поиска, масштабирования, настройки громкости и т.п. С помощью Media Player пользователь может подготовить на видео инструкции по эксплуатации и ремонту или стандартные процедуры, которые позволяют создать условия для эффективного выполнения заданий рабочими в соответствии с понятными, четкими указаниями.

Настройка свойств

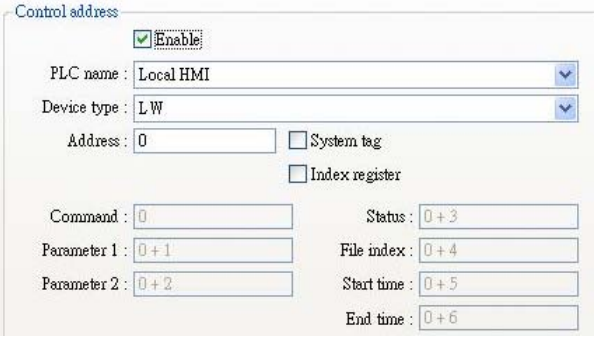
Щелкните пиктограмму [Media Player] на панели инструментов — откроется диалоговое окно [New Media Player Object], настройте параметры на закладке [Genreal], нажмите OK — будет создан новый объект (см. рисунок ниже).
Примечание: содержание этого раздела — пример воспроизводства видео-файла расположенного в папке “/example”.



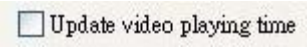
Закладка General :

Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** Управляющий адрес (Control address).
Пользователи могут назначить адрес в закладке –*General* , при создании нового объекта.

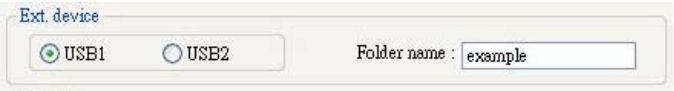
a. В области [Control address], отметьте флаг [Enable] и назначьте адрес для управления работой объекта [Media Player] (например: LW0).



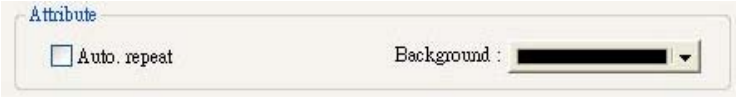
b. В этой же области снимите флажок [Update video playing time]



c. В области [Ext. device] выберите [USB1] и введите название “***example***” в поле [Folder name].



d. В области [Attribute], снимите флаг [Auto.repeat] и выберите черный цвет в качестве фонового.



Закладка Preview :

Можно узнать, поддерживает ли панель данный формат видео с помощью функции предварительного просмотра.

a. Нажмите кнопку [Load…] и выберите файл для проверки. (Пользователю необходимо поместить файл в папку ***/example*** на USB-накопителе.)

b. Если [media player] начнет воспроизведение видео, значит MT8000 поддерживает этот формат видео.Используйте кнопки со стрелками [<<] и [>>] для поиска нужного кадра с шагом 1 минута.
c. Для воспроизведения другого видео, нажмите [Stop] для закрытия файла и повторите шаг “a”.

Подготовка видео-файла:

a. Отсоедините все внешние устройства (SD-карты памяти и USB-флешки), подключенные к панели.

b. Вставьте в USB-накопитель, на котором находится видео-файл.

ПРИМЕЧАНИЕ

Прежде всего нужно убедиться, что USB-устройство было распознано системой как USB1.

Начало/Остановка воспроизведения видео

1. Начало воспроизведения видео

a. Установите [Parameter 1] в 0.
b. Установите [Command] в 1, система откроет видео-файл и начнет воспроизведение.
c. После начала воспроизведения, параметр [Command] будет сброшен в 0.

ПРИМЕЧАНИЕ

Между выполнением шагов b и c не меняйте содержимое адресов [Command], [Parameter 1] и [Parameter 2], это может привести к ошибкам.

2. Остановка воспроизведения видео

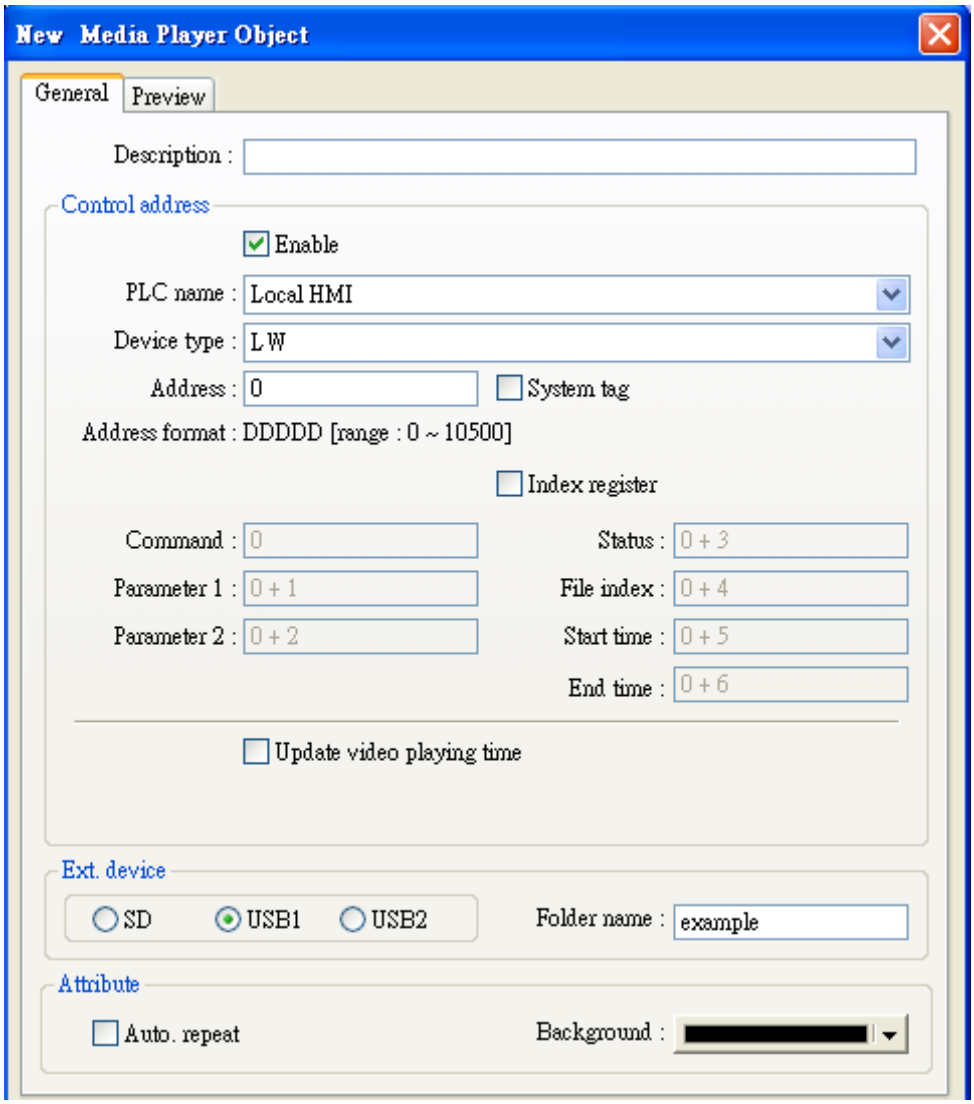
a. Установите значение [Command] равным 5, система остановит воспроизведение видео и закроет файл.
b. После закрытия система сбросит значение [Command] в 0.

ПРИМЕЧАНИЕ

При выполнении шагов a и b не изменяйте значения [Command], [Parameter 1], [Parameter 2] — это может привести к ошибкам.

Руководство по настройке [Media Player]

Закладка General :



Настройка		Описание
Control address	**Enable control address - Разрешение использовать управляющий адрес**	• Enable a. Можно использовать [Control address] для управления работой [Media Player]. b. Укажите адрес для “Control address”. • Disable В этом случае не будет возможности контролировать вручную процесс воспроизведения видео. Система начнет воспроизведение первое видео из назначенной папки, когда открывается рабочее окно.

Control address	**Command**		Пользователь устанавливает данные адреса для управления процессом работы *media player*. ➢ Command (control address + 0)
	Parameter 1		Parameter 1 для управления. ➢ Parameter 1 (control address + 1)
	Parameter 2		Parameter 2 для управления. Parameter 2 (control address + 2)
	Status		Значение бита станет “ON” при изменении состояние или ошибках. ➢ Status (control address + 3)
	File index		Система запишет индекс файла при начале воспроизведении видео. ➢ File index (control address + 4)
	Start time		Система запишет время начала воспроизведения видео (ед. измерения = секунды) ➢ Start time (control address + 5)
	End time		Система запишет время окончания воспроизведения при запуске видео-файла. (ед. измерения = секунды) ➢ End time (control address + 6)
	Video playing time	Update video playing time	• Enable Система будет записывать время, прошедшее с начала воспроизведения, по адресу [playing time] каждый период [update period].
		Update period	Период обновления значения [playing time], находится в диапазоне от 1 до 60 с.
		Playing time	Периодически обновляется прошедшее время воспроизведения (в секундах). ➢ Playing time (control address + 7)
Video file store location	**USB1**		Воспроизведение видео-файлов из устройства USB1.
	USB2		Воспроизведение видео-файлов из устройства USB2.
	Folder name		Название папки, в которой хранятся видео-файлы. Пользователь должен поместить видео в папку (например ***“/example”***), но не в корневую директорию. Примечание 1. [Folder name] не может быть пустым. 2. [Folder name] не может включать символы: **/\:*?”<>|**.
Attribute	**Auto. repeat**		После того, как воспроизведение файла окончено, автоматически начнется воспроизведение следующего. Например, [video 1] → [video 2] → … [video n] → [video 1]
	Background		Выберите цвет фона объекта.

* Обычно формат приведенных выше адресов — 16-разрядный целый. Если в качестве [control address] выбрано 32-разрядное число, используются только биты с 0 по 15-й. Пользователю нужно обнулить биты с 16-го по 31-й.

Control command :

a. Воспроизведение *index file*
[Command] = 1
[Parameter1] = file index
[Parameter2] = ignore (set 0)

ПРИМЕЧАНИЕ

1. Файлы упорядочиваются по имени в порядке возрастания, т.е. сначала идет файл с file index=0 и т.д.
2. Если невозможно прочитать файл, то он бит состояния [status] будет установлен в ON.
3. Если отмечен флаг [Auto. repeat], то автоматически начнется воспроизведение следующего файла после окончания предыдущего.

b. Воспроизведение предыдущего файла
[Command] = 2
[Parameter1] = ignore (set 0)
[Parameter2] = ignore (set 0)

ПРИМЕЧАНИЕ

1. Если file index установлен равным 0, то будет воспроизводиться с начала то же самое видео.
2. Если невозможно прочитать файл, то он бит состояния [status] будет установлен в ON.
3. Если включена функция [Auto. repeat], автоматически начнет воспроизводиться следующий файл после окончания данного.

c. Воспроизведение следующего файла
[Command] = 3
[Parameter1] = ignore (set 0)
[parameter2] = ignore (set 0)

ПРИМЕЧАНИЕ

1. Если нет следующего видео-файла, будет воспроизводиться первый (index 0) файл.
2. Если невозможно найти нужный файл, состояние бита [status] будет изменено на ON.
3. Если включена функция [Auto. repeat], автоматически начнет воспроизводиться следующий файл после окончания данного.

d. Переключатель Пауза/Воспроизведение
[Command] = 4
[Parameter 1] = ignore (set 0)
[Parameter 2] = ignore (set 0)

e. Остановка воспроизведения и закрытие файла

[Command] = 5
[Parameter 1] = ignore (set 0)
[Parameter 2] = ignore (set 0)

f. Начало воспроизведения с назначенного места

[Command] = 6
[Parameter 1] = назначенное место (в секундах)
[Parameter 2] = ignore (set 0)

ПРИМЕЧАНИЕ

Parameter 1 (назначенное место) должно быть меньше общего времени воспроизведения. Если оно превышает это время, будет воспроизведена последняя секунда видео.

g. Перемотка вперед

[Command] = 7
[Parameter 1] = target location (sec)
[Parameter 2] = ignore (set 0)

ПРИМЕЧАНИЕ

1. Сдвиг текущего места воспроизведения на время, задаваемое [Parameter 1] (в секундах). Если система до этого воспроизводила видео, то продолжит воспроизведение. Если была установлена пауза, то она сохранится.
2. Если время смещения превышает длительность видео, будет воспроизведена последняя секунда видео.

h. Перемотка назад

[Command] = 8
[Parameter 1] = target location (The unit is sec)
[Parameter 2] = отменить (set 0)

ПРИМЕЧАНИЕ

1. Сдвиг текущей позиции воспроизведения на время, заданное в [Parameter 1] (в секундах). Если до этого воспроизводилось видео, то продолжится воспроизведение. Если бы установлена пауза, то она сохранится.
2. Если указанное время-результат смещения меньше времени начала видео, оно будет воспроизводиться с самого начала.

i. Настройка громкости

[Command] = 9
[Parameter 1] = volume (0 ~ 128)
[Parameter 2] = ignore (set 0)

ПРИМЕЧАНИЕ Значение по умолчанию равно 128.

j. Установка размеров окна видео

[Command] = 10
[Parameter 1] = display size (0 ~ 16)
[Parameter 2] = ignore (set 0)

ПРИМЕЧАНИЕ

1. [0]: Вписать видео в размеры объекта.
2. [1~16]: Масштабирование: от 25% до 400%. Установите 1 для 25%, 2 для 50%, 3 для 75% и т.д.

k. Статус (control address + 3)

15	09	08	02	01	00	bit
Reserved (all 0)	0	0		0	0	

Бит 00: открытие файла (0: файл закрыт; 1: файл открыт)

Бит 01: воспроизведение файла (0: нет воспроизводимых файлов видео; 1: воспроизведение идет)

Бит 08: бит ошибки команды (0: верная команда; 1: неверная команда или параметры)

Бит 09: бит ошибки файла (0: формат файла поддерживается; 1: неизвестный формат файла или ошибка чтения)

При воспроизведении видео-файла система установит в ON биты [open file bit] и [play file bit]. Если файл не может быть прочитан или неверная команда, то будет установлен в ON (0→1) бит [command error bit].

ПРИМЕЧАНИЕ

1. Если формат файла не поддерживается или возникла ошибка ввода/вывода во время воспроизведения (например, пользователь извлек USB-накопитель), то бит [file error bit] будет установлен в ON (0→1).
2. См. рисунок ниже, значение бита [status] в разных состояниях будет:

   “Stop” [status] = 0

   “Pause” [status] = 1 ([open file bit])

   “Playing” [status] = 3 ([open file bit] + [play file bit])

* Пользователю нужно устанавливать только значения регистров [Command], [Parameter 1] и [Parameter 2], остальные регистры только для чтения.

Ограничения

- Может воспроизводиться только один видео-файл в любой момент времени.
- Если не выбрана функция [Auto. repeat], воспроизведение видео будет остановлено и файл закрыт.
- Если не указан адрес [control address], система найдет первый файл в указанной директории и начнет его воспроизведение.

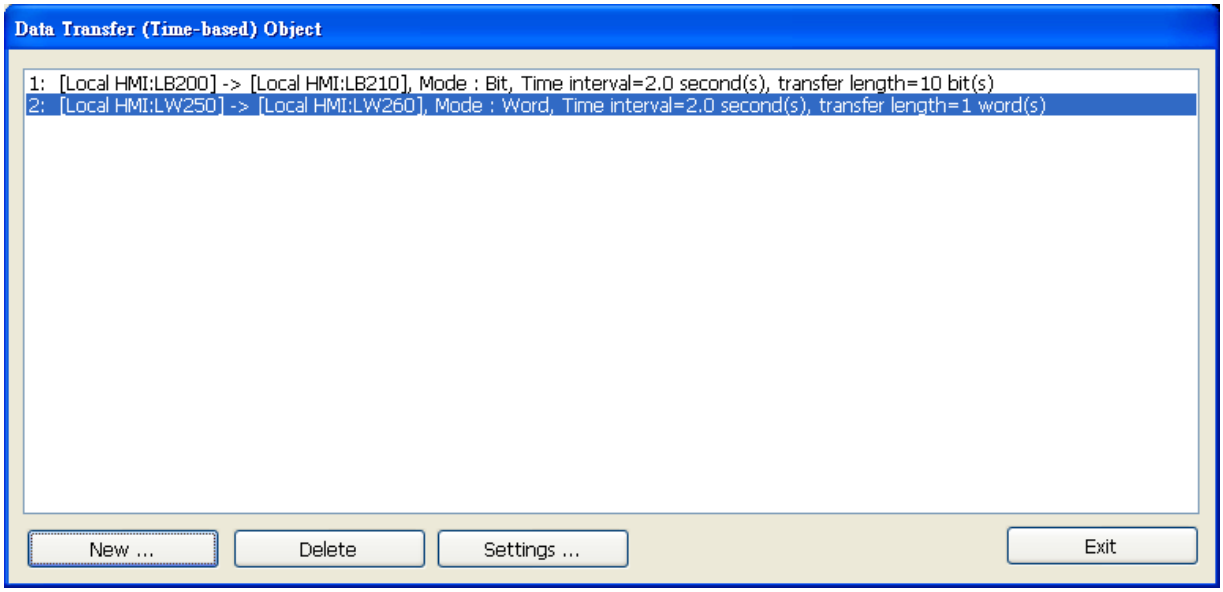
13.26 Data Transfer (Time-based) – Перенос данных по времени

Краткое описание

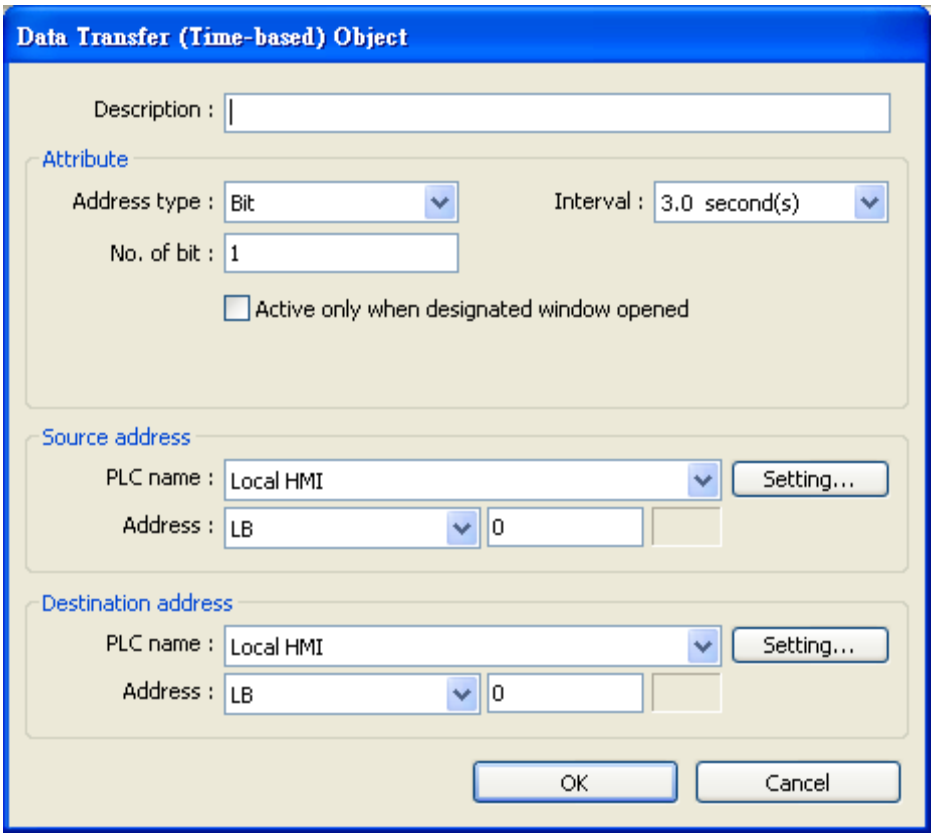
Объект [Data Transfer (Time-based)] — то же, что и объект [data transfer (trigger-based)], т.к. он тоже перемещает данные из одного адреса в назначенный. Разница в способе активации процесса перемещения. [Data Transfer (Time-based)] перемещает данные по времени, также может перемещать переводит данные в группы битов.

Настройка свойств

Щелкните пиктограмму [Data Transfer (Time-based)] на панели инструментов: появится экран со сводкой объектов [Data Transfer (Time-based)].



Нажмите [New…] в таком окне, как показано выше, откроется окно [Data Transfer (Time-based) Object], как показано далее, задайте все параметры, нажмите OK — будет создан новый объект.



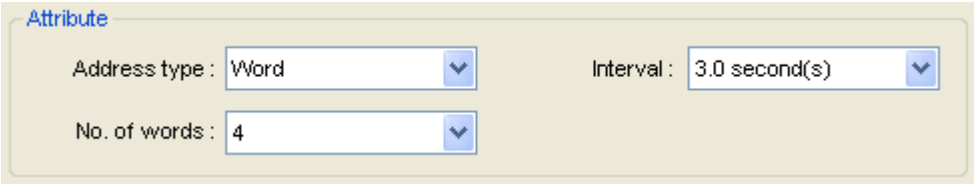
Attribute

Address type

Задайте адрес бита или слова.

[No. of words] or [No. of bits]

При выборе типа “Word type” единицей перемещаемых данных является слово, установите объем перемещаемых данных. См. рисунок ниже.



Если выбран тип данных “Bit type”, то единицей перемещаемых данных является бит, установите число перемещаемых битов.

Interval

Выберите интервал ожидания передачи, например, 3 секунды, система будет передавать данные каждые 3 секунды.

ПРИМЕЧАНИЕ

1. Задание слишком маленьких интервалов времени или большого объема передаваемых данных может вызвать снижение общей производительности из-за потери времени на передачу. Таким образом, следует выбирать большие интервалы и меньшие объемы передаваемых данных.
2. Когда короткие интервалы необходимы, нужно учитывать что они должны быть больше длительности передачи. Например, если процесс передачи занимает 2 секунды, то необходимо задать интервалы больше двух секунд.

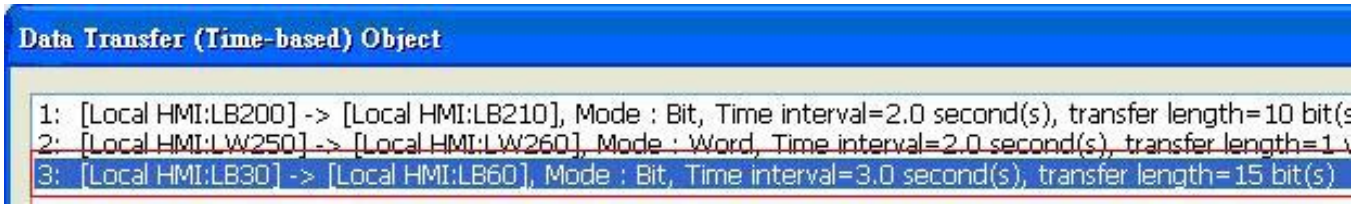
Source address

Укажите адрес-источник.
Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** адрес источника.
Пользователи могут назначить адрес в закладке –*General* , при создании нового объекта.

Destination address

Укажите адрес-приемник данных.
Щелкните [Setting…], чтобы выбрать **[PLC name], [Device type], [Address], [System tag], [Index register]** адрес назначения.
Пользователи могут назначить адрес в закладке –*General* , при создании нового объекта.

После завершения всех настроек и нажатия OK будет создан новый объект [Data Transfer (Time-based)]. В отчете будут кратко показаны все данные об объектах типа [Data transfer].



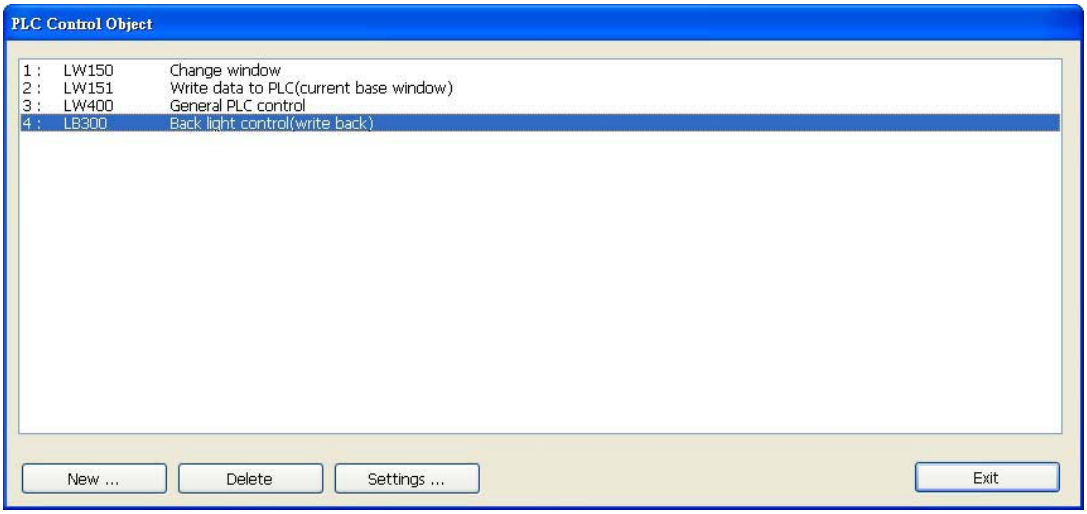
13.27 Объект [PLC Control]

Краткое описание

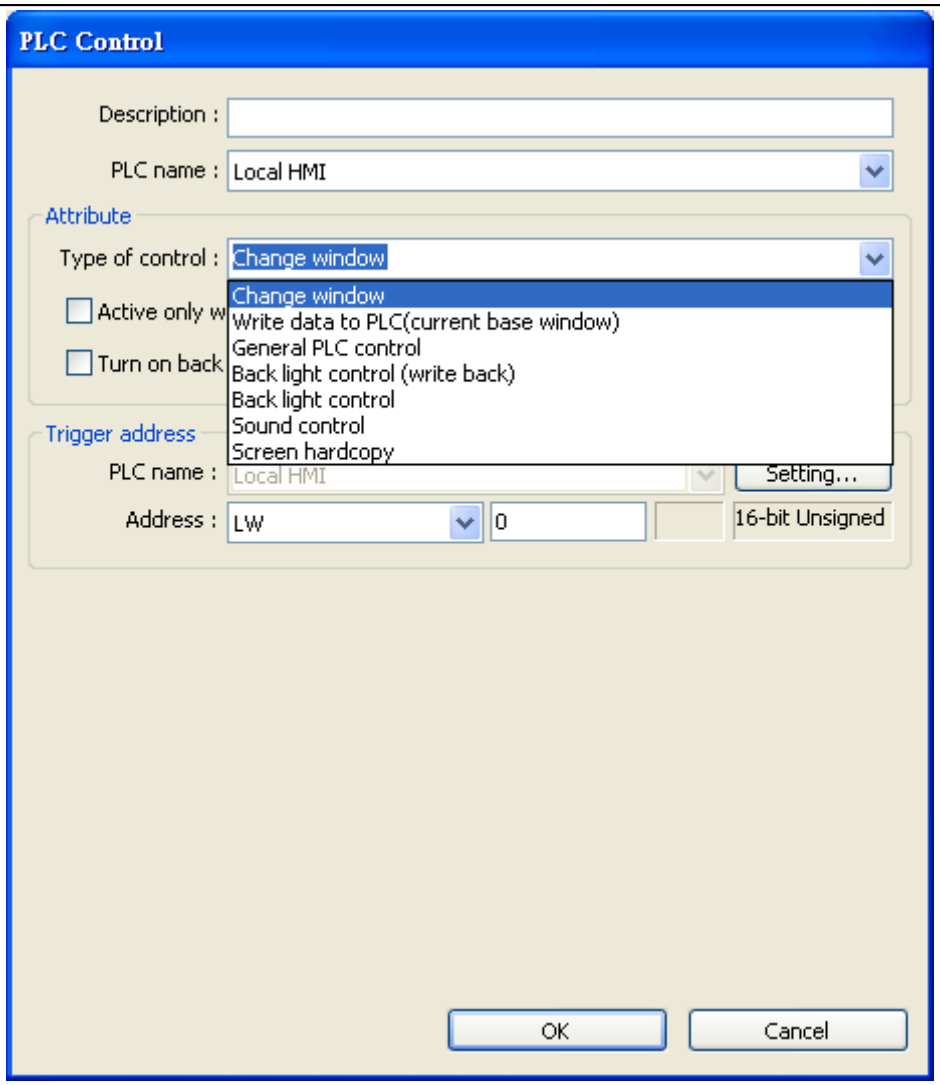
Объект [PLC control] активирует назначенные операции, когда срабатывает соответствующий управляющий адрес.

Настройка свойств

Щелкните пиктограмму [PLC control] — откроется сводка объектов [PLC Control Object] как показано ниже.



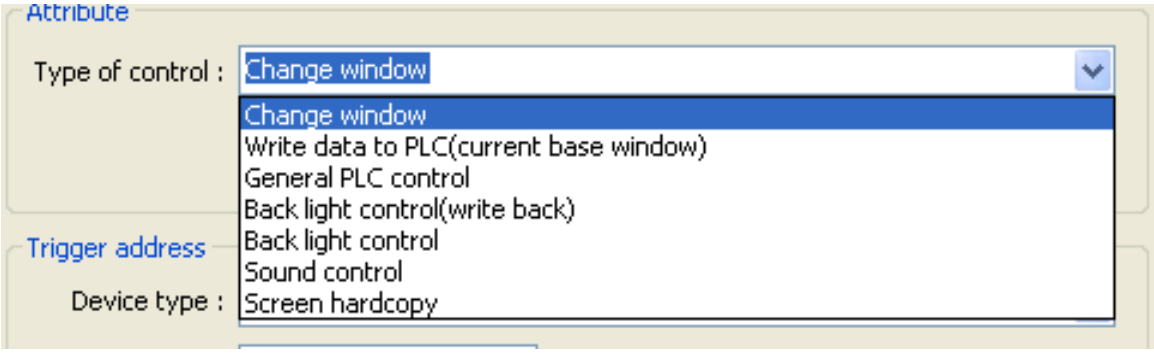
Нажмите кнопку [New…] — откроется диалоговое окно [PLC Control]. Задайте все атрибуты, нажмите ОК — будет создан новый объект.



Attribute & Trigger address

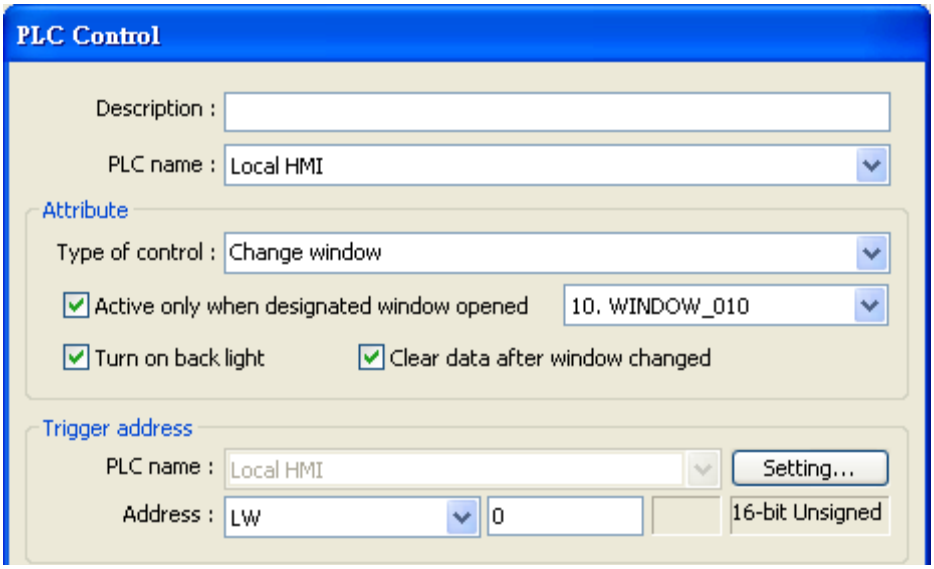
[Type of control]

Установка типа управления. Щёлкните кнопку и вы можете просмотреть выпадающий список всех доступных функций управления ПЛК.



a. “Change window” – Сменить окно

Эта функция используется для смены основного окна. Когда в [trigger address] записан корректный номер окна, система закроет текущее окно и откроет окно, определенное указанным адресом [trigger address]. Новый номер окна будет записан по адресу [trigger address + 1].



Выше приведен пример настроек. Когда по адресу [LW0] записан корректный номер окна — 11, система закроет текущее окно и откроет окно 11, затем запишет номер 11 по адресу LW1 (LW0+1).

Если используется 32-разрядный адрес в качестве [trigger address] и тип адресного пространства – слова, система запишет номер окна по адресу [trigger address +2].
Ниже дан список распределения адресов для различных типов данных.

Формат данных	[Trigger Address]	[Write address]
16-bit BCD	Address	Address + 1
32-bit BCD	Address	Address + 2
16-bit Unsigned	Address	Address + 1
16-bit Signed	Address	Address + 1
32-bit Unsigned	Address	Address + 2
32-bit Signed	Address	Address + 2

ПРИМЕЧАНИЕ : Если [LB9017]=ON, перезапись не выполняется.
Если выбрана функция “clear data after window change” (очистить данные после смены окна), [trigger address] будет сброшен в 0 после открытия нового окна.

b. “Write data to PLC (current base window)” – Записать в ПЛК номер текущего основного окна

Когда система изменяет основное окно, номер нового окна будет записано в [Trigger address].

c.“General PLC Control”

Эта функция выполняет передачу данных между ПЛК и операторской панелью , когда пользователь устанавливает соответствующее значение в [trigger address].

Управляющее число в [trigger address]	Способ переноса адреса
1	PLC register → Панель RW
2	PLC register → Панель LW
3	Панель RW → PLC register
4	Панель LW → PLC register

Для данной функции система использует четыре непрерывных адреса, подробнее см. пример ниже.

Адрес	Назначение	Описание
[Trigger address]	Управляющий код	Корректное значение кода представлено в таблице выше. Когда новое значение записывается в регистр, производится перенос данных.
[Trigger address+1]	Объем данных для переноса (в словах)	
[Trigger address+2]	Смещение для начального адреса регистра ПЛК	Если значение равно “n”, начальный адрес регистра ПЛК равен “Trigger address + 4 + n”.
[Trigger address+3]	Начальный адрес LW или RW	

Например, для переноса данных из регистров ПЛК [DM100, 101, … 105] в регистры [RW10, 11, …, 15] панели оператора нужно выполнять указанную ниже последовательность действий:

1. Установить [Trigger address] в DM10.
2. Установить [DM11]=6 (число слов для переноса).
3. Установить [DM12]=86 (DM10+4+86=DM100)
4. Установить [DM13]=10 (RW10).
5. Установить [DM10]=1, система выполнит перенос данных.

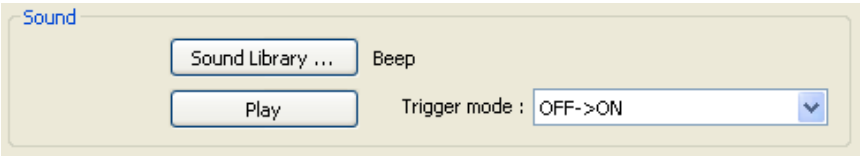
d. “Back light control (write back)”

Установите значение [Trigger address] в 1, система отключит подсветку и сбросит [Trigger address]. Любое прикосновение к экрану включит подсветку.

e. “Back light control”

Эта операция такая же как предыдущая, кроме того, что не сбрасывает значение [Trigger address].

f. “Sound control”



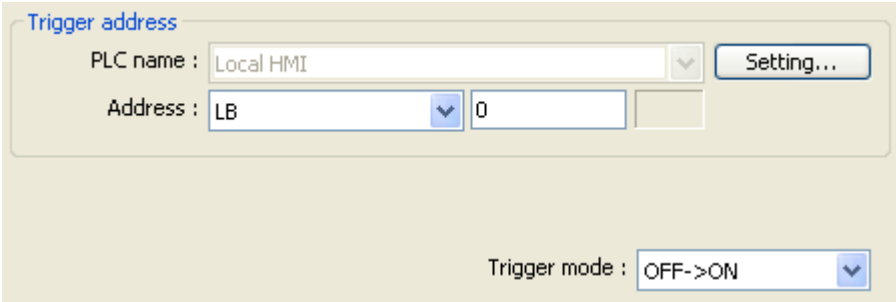
При установке [Trigger address] система воспроизводит звук.
Можно задать различные способы срабатывания [Trigger address]:

(1) OFF → ON

(2) ON → OFF

(3) Любое изменение состояния

g. “Execute macro program”



При установке [Trigger address], будет выполняться макрос.
Можно установить три разных способа активации [Trigger address]:

(1) Изменение состояния OFF → ON

(2) Изменение состояния ON → OFF

(3) Любое изменение состояния

h. “Screen hardcopy”

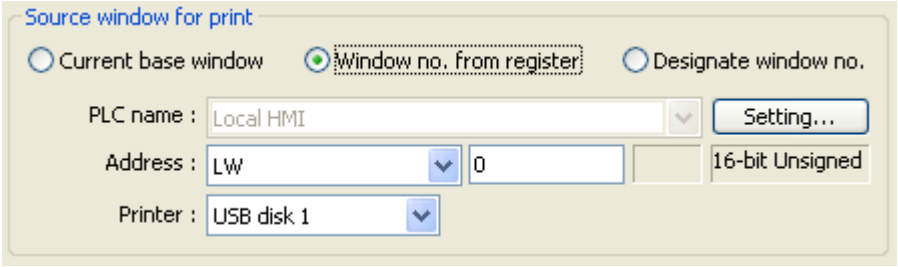
При установке [Trigger address] заданное окно будет выводиться на печать.
Можно установить три разных способа установки [Trigger address]:

(1) Изменение состояния OFF → ON

(2) Изменение состояния ON → OFF

(3) Любое изменение состояния

Назначенное окно может быть одним из трех указанных типов:



[Current base window]

Распечатать текущее основное окно при активации операции.

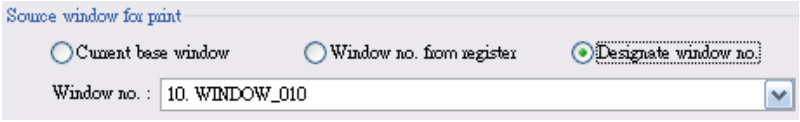
[Window no. from register]

Распечатать окно, назначенное в регистре ПЛК.

Например, если [LW0]=14, то на печать будет выведено окно 14.

[Designate window no.]

Выбрать основное окно для печати.



ПРИМЕЧАНИЕ

1. Система выполняет ***печать в фоновом режиме***, когда выводимое окно не является основным текущим.
2. В окне, назначенном для печати в фоновом режиме, нельзя использовать ни [direct window], ни [indirect window].

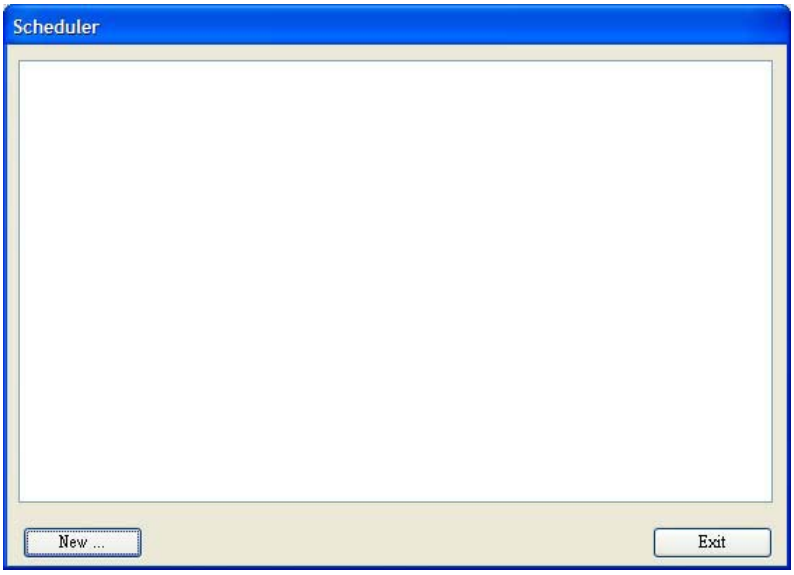
13.28 Schedule - Расписание

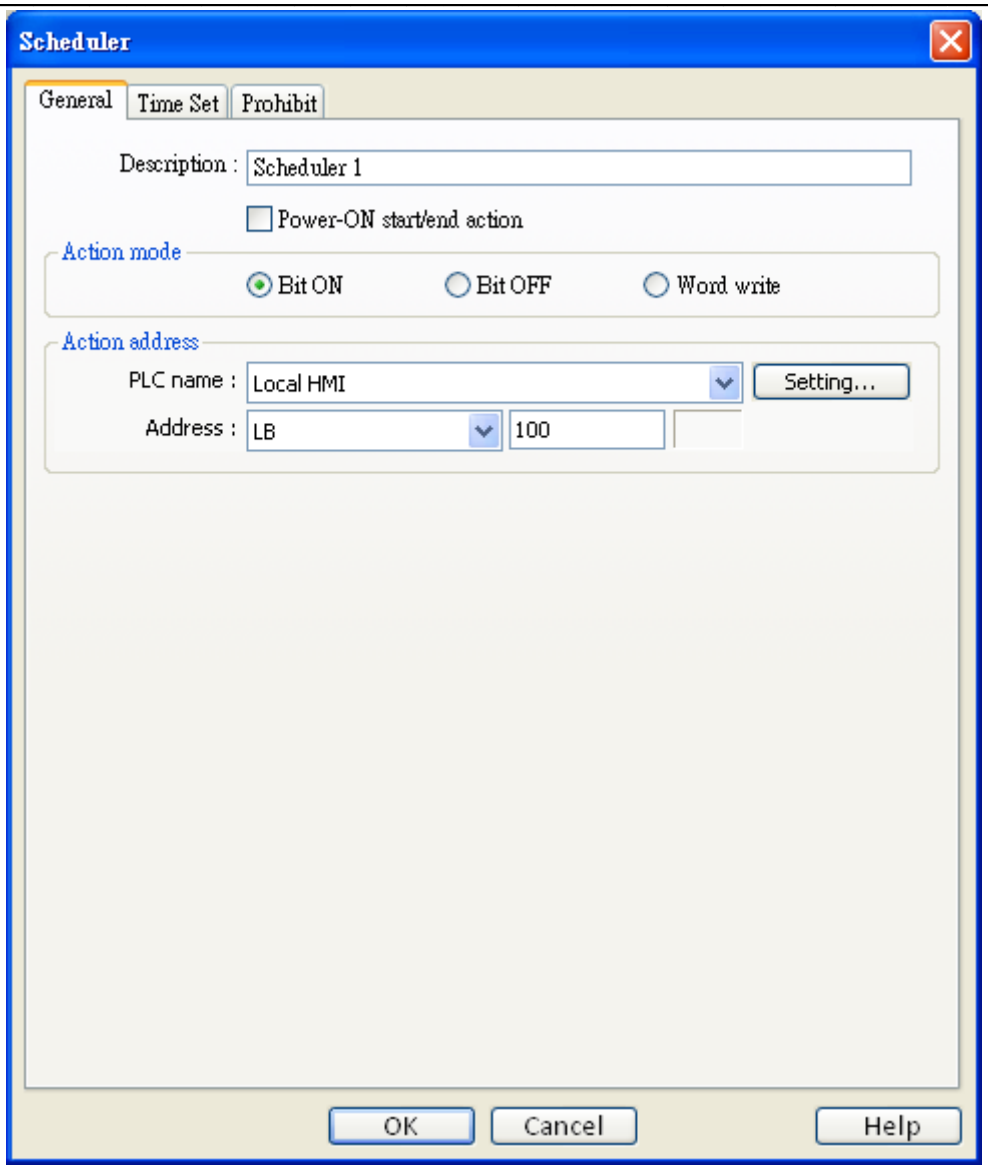
Краткое описание

Объект [Schedule] используется для включения/выключения бита или записи значения слова в назначенное время. Расписание гибкое, оно может быть ежедневным или еженедельным. Для лучшего применения можно использовать таблицу (набор адресов) для задания начального и предельного времени, и далее обновлять таблицу по графику.

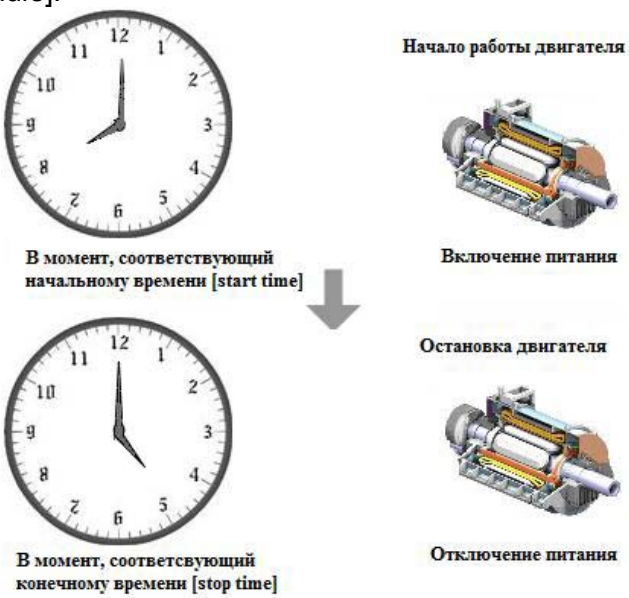
Настройка свойств

Щелкните пиктограмму [Schedule] на панели инструментов — появится окно со списком расписаний [Scheduler], нажмите кнопку [New…], откроется показанное ниже окно:





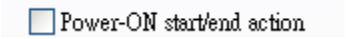
Пример 1:
Двигатель по расписанию включается в 8:00 и его питание отключается в 17:00 с понедельника по пятницу.
В данном случае используется [LB100] для управления двигателем. Нужно следовать шагам по установке объекта [schedule].



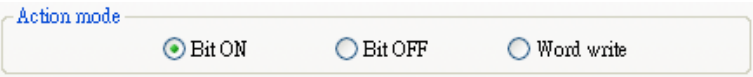
Щелкните [New…] для добавления нового объекта.

Закладка [General]

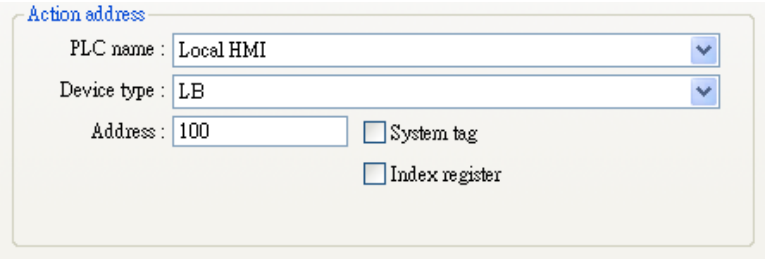
[Power-ON start/end action]
Подробно: см. руководство по настройкам Планировщика (Scheduler) .



1. Отметьте [Bit ON] в области [Action mode],

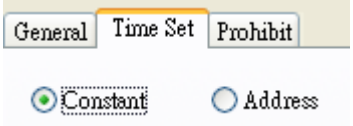


2. Задайте [LB100] в области [Action address]

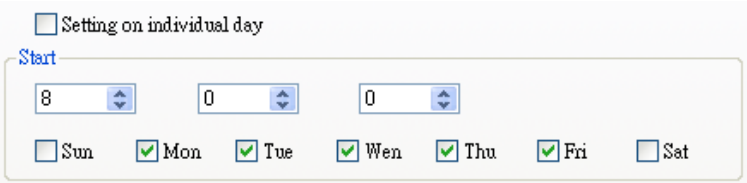


Закладка [Time set]

3. Зайдите на закладку [Time Set], отметьте [Constant]

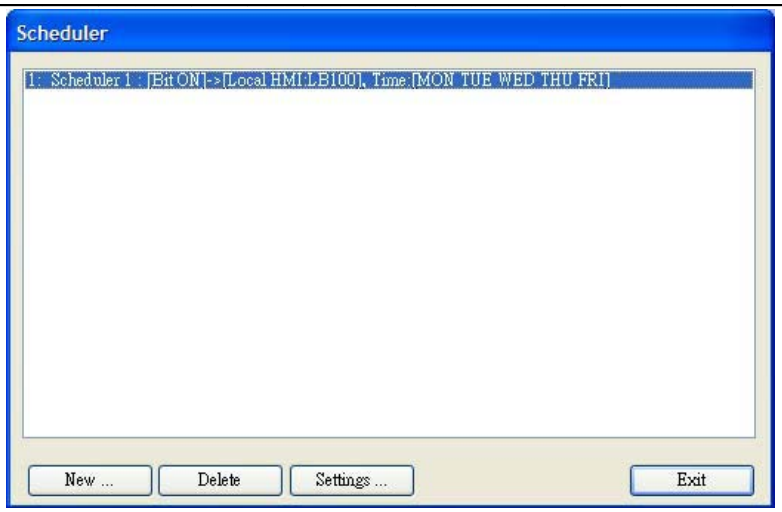


4. Снимите флаг [Setting on individual day] . В области [Start] настройте время как 8:00:00 и выберите дни недели с понедельника по пятницу.



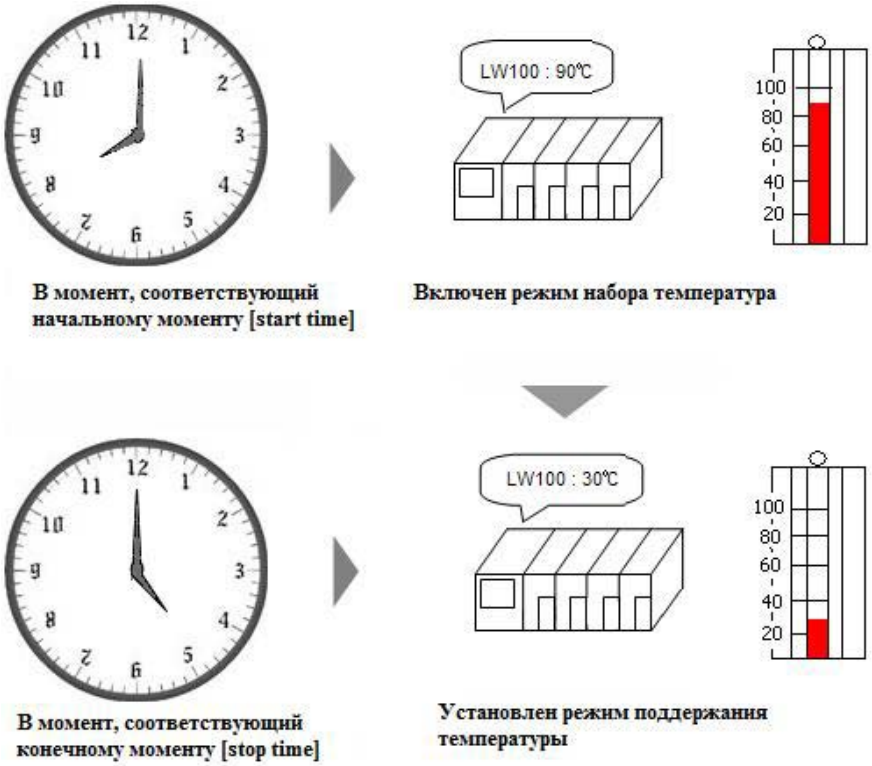
5. В области [End] отметьте флаг [Enable termination action] и настройте время 17:00:00.

6. Нажмите OK, новый объект будет создан и появится в списке [Scheduler].



Пример 2:

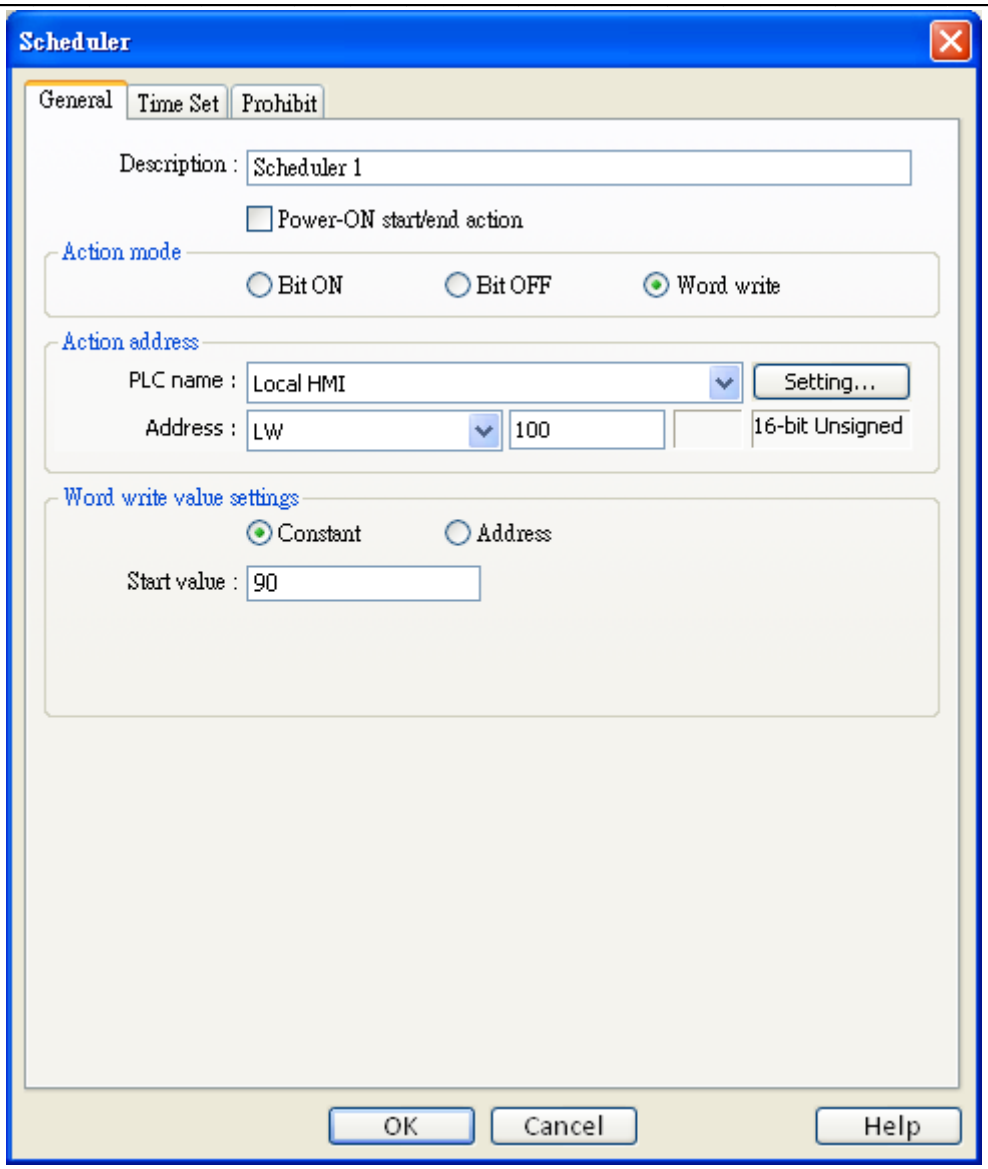
Нужно устанавливать температуру 90F в 8:00 и возвращать к значению к 30F (режим ожидания) в 17:00 с понедельника по пятницу.



Нажмите [New…] для добавления нового объекта [schedule].

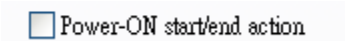
Следуйте указанной ниже последовательности настройки данного объекта.

Адрес [LW100] используется для хранения уставки температуры.



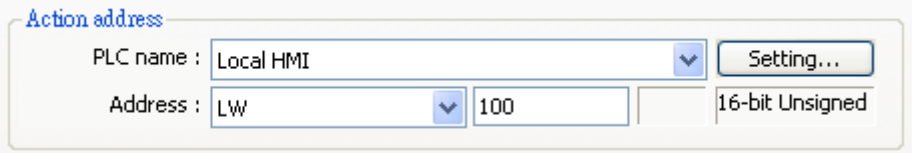
Закладка [General]

1. [Power-ON start/end action]

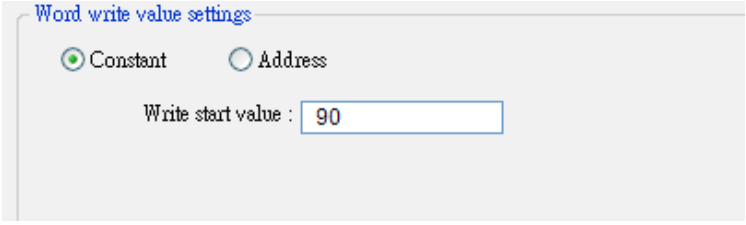


2. Отметьте [Word write] в области [Action mode]

3. Задайте адрес [LW100] в [Action address]

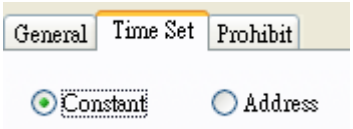


4. Отметьте [Constant] и установите [Write start value] равным 90 в поле [Word write value settings]

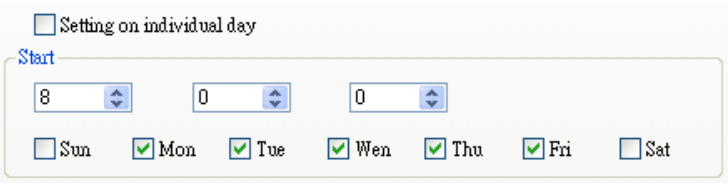


Закладка [Time Set]

5. Зайдите на закладку [Time set], выберите [Constant]



6. Снимите флаг [Setting on individual day]. В области [Start] укажите время 8:00:00 и выберите все дни с понедельника по пятницу (Monday~Friday).



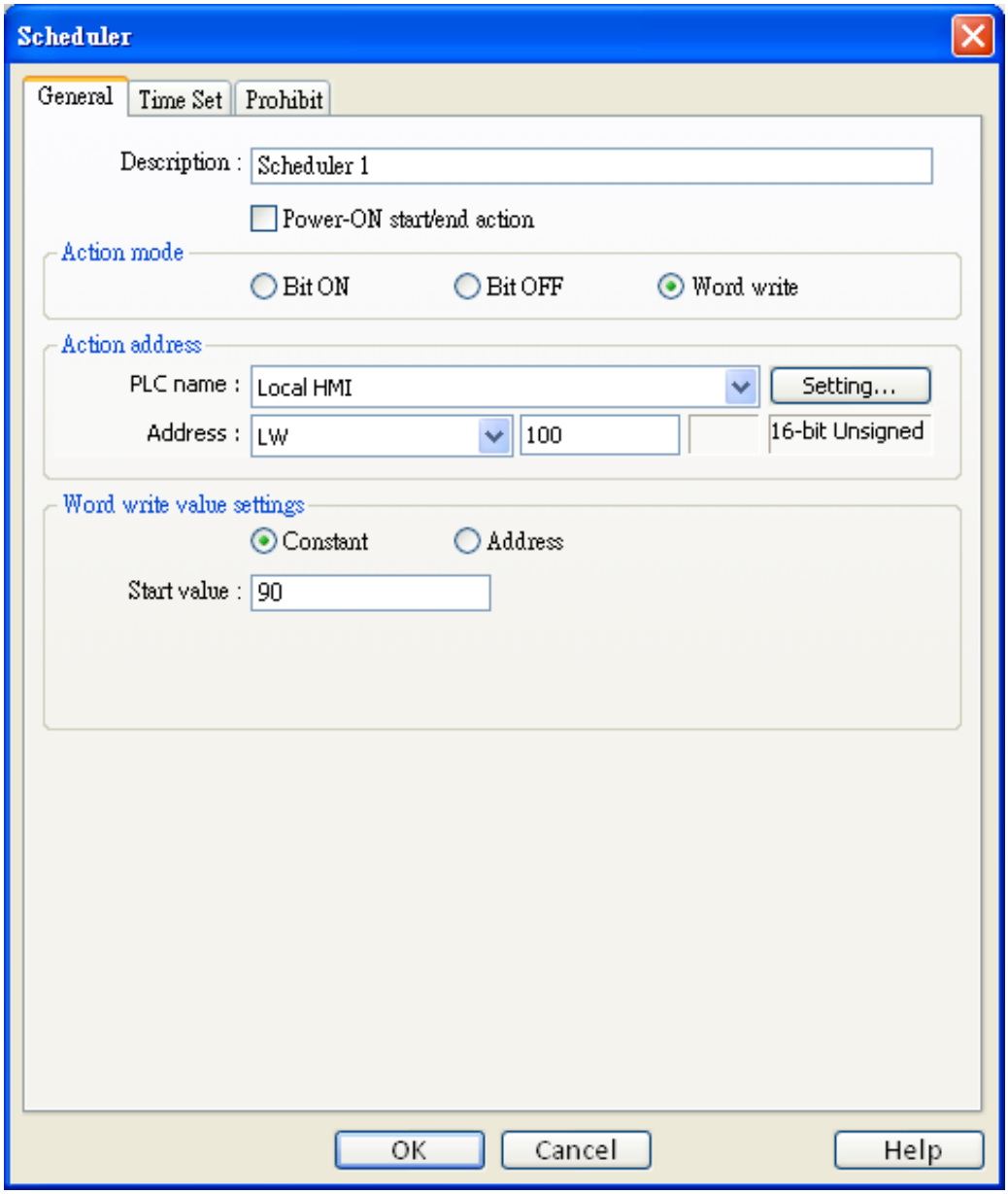
7. В поле [End] отметьте флаг [Enable termination action] укажите время 17:00:00.

8. Зайдите на закладку [General], установите значение [Write start value] равным 90 и [Write end value] равным 30.

9. Нажмите OK, установки появятся в списке [Scheduler].

Руководство по настройкам Планировщика расписаний - [Scheduler]:

Закладка [General]

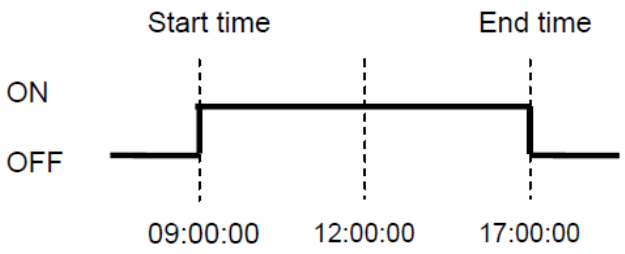


Action Mode Выберите тип операции, выполняемой в указанное время.

[Bit ON]

В начальный момент времени указанному биту присвоить значение ON.
В конечный момент времени, присвоить OFF.

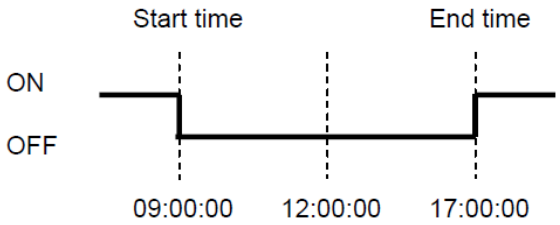
Пример:
Начальный момент = 09:00:00
Конечный момент = 17:00:00



[Bit OFF]

В начальный момент времени установить значение бита в OFF.
В конечный момент времени установить значение ON.

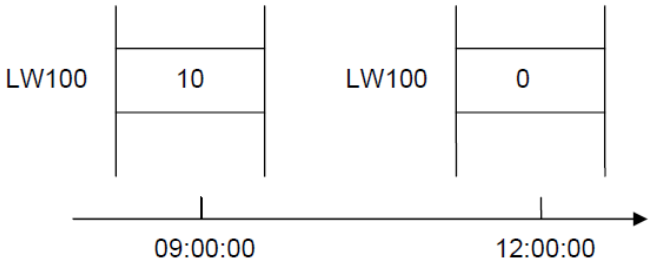
Пример: Начальный момент времени = 09:00:00
Конечный момент времени = 17:00:00



[Word write]

В начальный момент времени указанное значение [Write start value] записывается по адресу.
В конечный момент времени, [Write end value] записывается в тот же адрес.

Пример:
Назначенный адрес = LW100
Начальное время = 09:00:00
Конечное время = 12:00:00
Write start value = 10
Write end value = 0



Action address

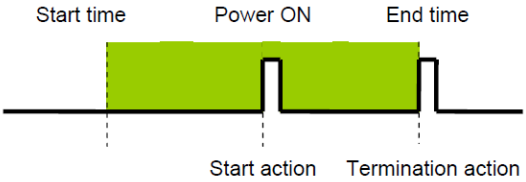
Определение адреса, с которым работает объект [Scheduler].

Power-ON start/end action

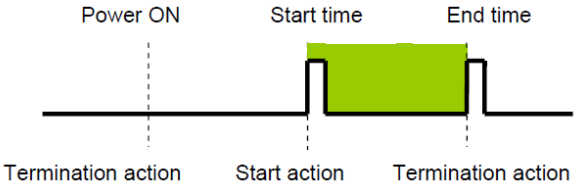
Выберите действие, выполняемое при включении питания.

✹ Если включено (Enable)
Если к панели подано питание внутри интервала времени, определяемого объектом [scheduler], будут выполнено начальное действие. Если питание включено за пределами указанного интервала времени, будет выполнено конечное действие (Termination action) .

Внутри интервала времени:



За пределами временного интервала



✹ Если отключено (Disable)
Когда питание включено, но время больше начального времени, действие не будет выполняться автоматически. Однако, конечное действие будет выполнено автоматически.
Также, если конечное действие не задано, заданный интервал невозможно распознать и действие не выполняется.

Word write value Settings

Эти настройки активны, только если в [Action Mode] выбран [Word Write].

[Write start value]

При выполнении начального действия система запишет данное значение в [action address].
- Для варианта [Constant]

Назначьте значение, записываемое в начальный момент времени.
- Для варианта [Address]

Назначьте адрес, используемый для хранения значения в начальный момент времени.

[Write end value]

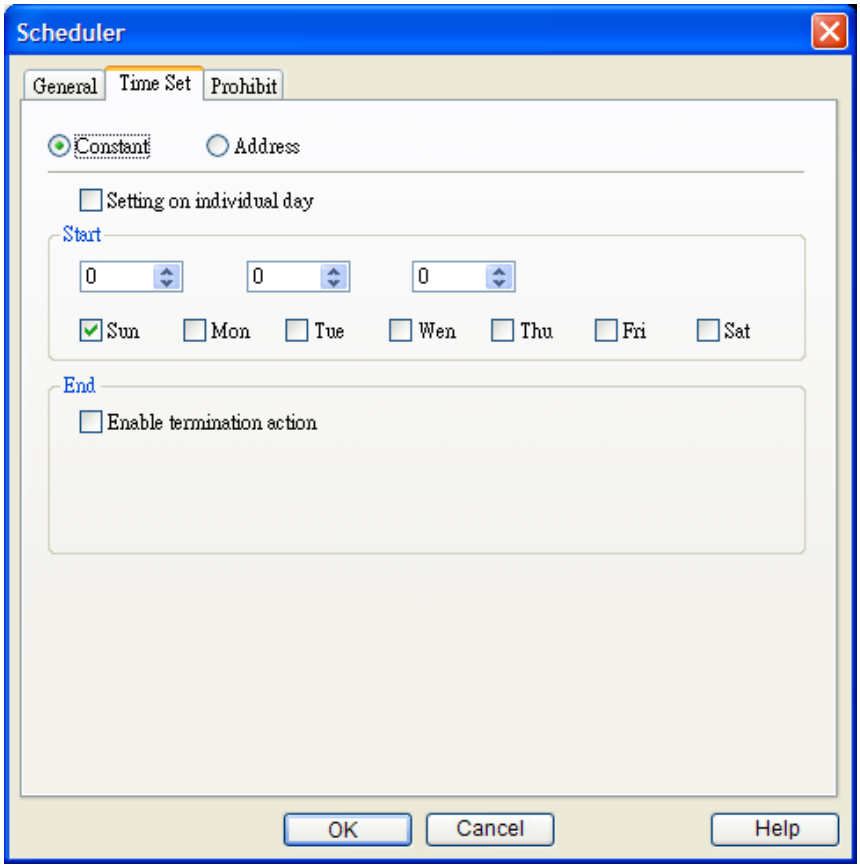
При выполнении конечного действия данное значение будет записано в [action address].
- Для варианта [Constant]

Назначьте значение, записываемое в конечный момент времени.
- Для варианта [Address]

Назначьте адрес, где будет храниться данное значение.

ПРИМЕЧАНИЕ
Данную опцию можно использовать, если на закладке [Time Set] отмечен флаг [Enable termination action].

Закладка [Time Set] - Задание времени, когда выбран вариант [Constant]



Constant/Address

Выберите способ задания начального и конечного времени.

- **Constant**

Определение фиксированного часа и дня.

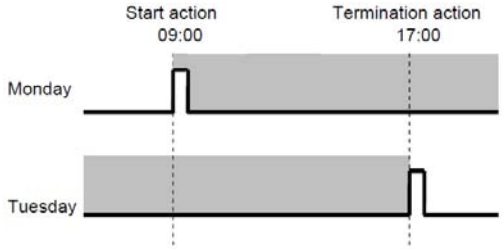
- **Address**

Значение начального/конечного времени берется из указанного адреса при работе в режиме он-лайн.

Setting on individual day

✹ Если выбрано *Enable*

Начальное и конечное время можно установить в различные дни недели. Можно назначить только по одному значению начального и конечного времени в течение недели. В данном режиме нужно указать и начальное и конечное время.



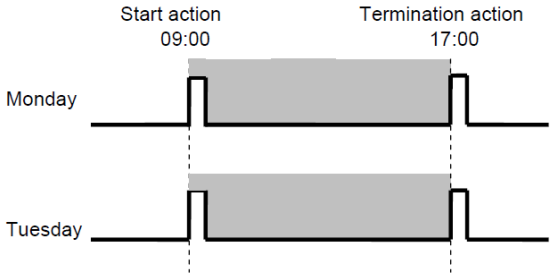
Примечания

1. Необходимо указать начальное и конечное время.
2. Нельзя устанавливать начальное и конечное время одинаковым (один и тот же день и час).

✹ Если выбрано *Disable*

Расписание на 1 день (начальное и конечное время находятся в пределах 24 часов). Можно выбрать несколько начальных и конечных дней. Можно выполнять действия в одно и то же время несколько дней.

Для ввода конечного времени нужно отметить [Enable termination action].



Примечания

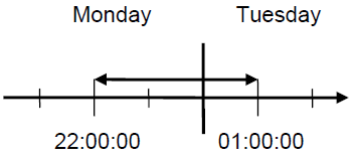
- Нельзя устанавливать начальное и конечно время на один и тот же день и час.
- Расписание рассчитано на один день, таким образом, если для конечного времени задано более раннее значение, чем для начального, то действие, соответствующее [End Time], будет выполнено на следующий день.

Например:

Начальный день: Понедельник (Monday)

Время начала: 22:00:00

Конечное время: 1:00:00

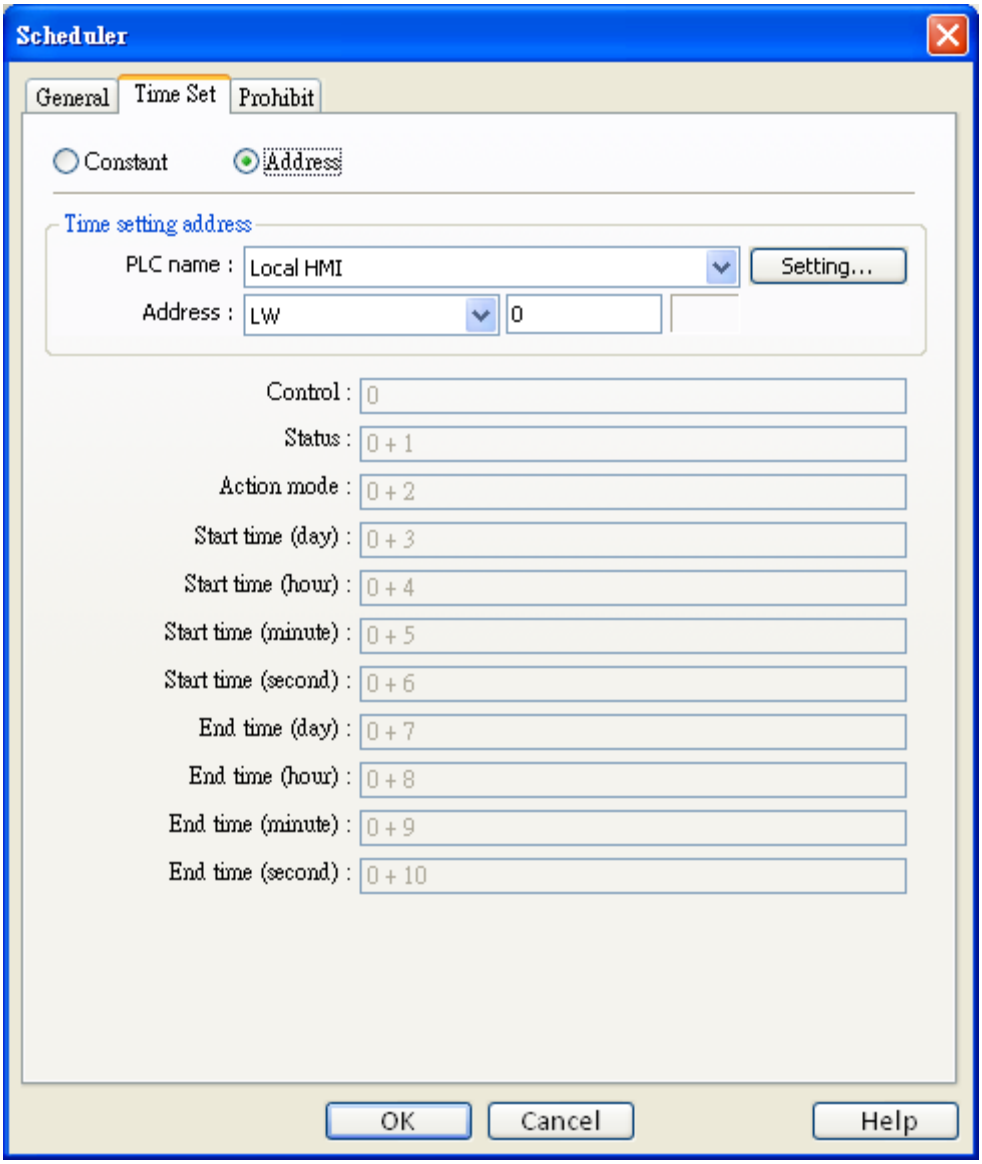


Start Установка начального дня и времени.
Если выбрано [Setting on individual day] - disable , то пользователь может назначать более одного дня

End Установка конечного дня и времени
Оно может быть определено, когда выбран флаг [Enable termination action].
Выбирать дни недели можно выбирать только при отмеченом флаге [Setting on individual day].

Задание времени, когда выбран вариант [Address]

Если выбран режим [address], значения параметров начального/конечного времени находятся по определенным адресам. Таким образом, пользователь может задавать и изменять время в расписании действий.



Пользователь назначает адрес [Time setting address], как начальный адрес, используемый для хранения настроек времени. При этом 11 адресов распределяются автоматически.

Обычно используется формат данных: 16-разрядный целый. Если выбран 32-разрядный формат, используются только биты с 0 по 15-й и пользователю нужно присвоить нулевые значения битам с 16-го по 31-й.

a. Control (Time setting address +0)

Структура управляющего слова (Control word) показана ниже.

Пользователь устанавливает бит [time acquisition request bit] в ON для того, чтобы система считывала значения [Action mode], [Start time], [End time] и использовала их в качестве новых установок расписания.

15	0 Bit
Reserved (0 fixed)	0

Бит 00: запрос определения времени (0: не производит никаких действий; 1: выполнить чтение времени).

Примечание

Система не считывает начальное и конечное время, пока бит [time acquisition request] установлен в ON.

b. Status (Time setting address +1)

Структура данного слова показана ниже.

Когда завершена операция чтения, бит [time acquisition complete] устанавливается в 1 (ON).

Также, если считанные данные некорректны,[error notification bit] устанавливается в 1(ON).

15	02	01	00 Bit
Reserved (0 fixed)	0	0	

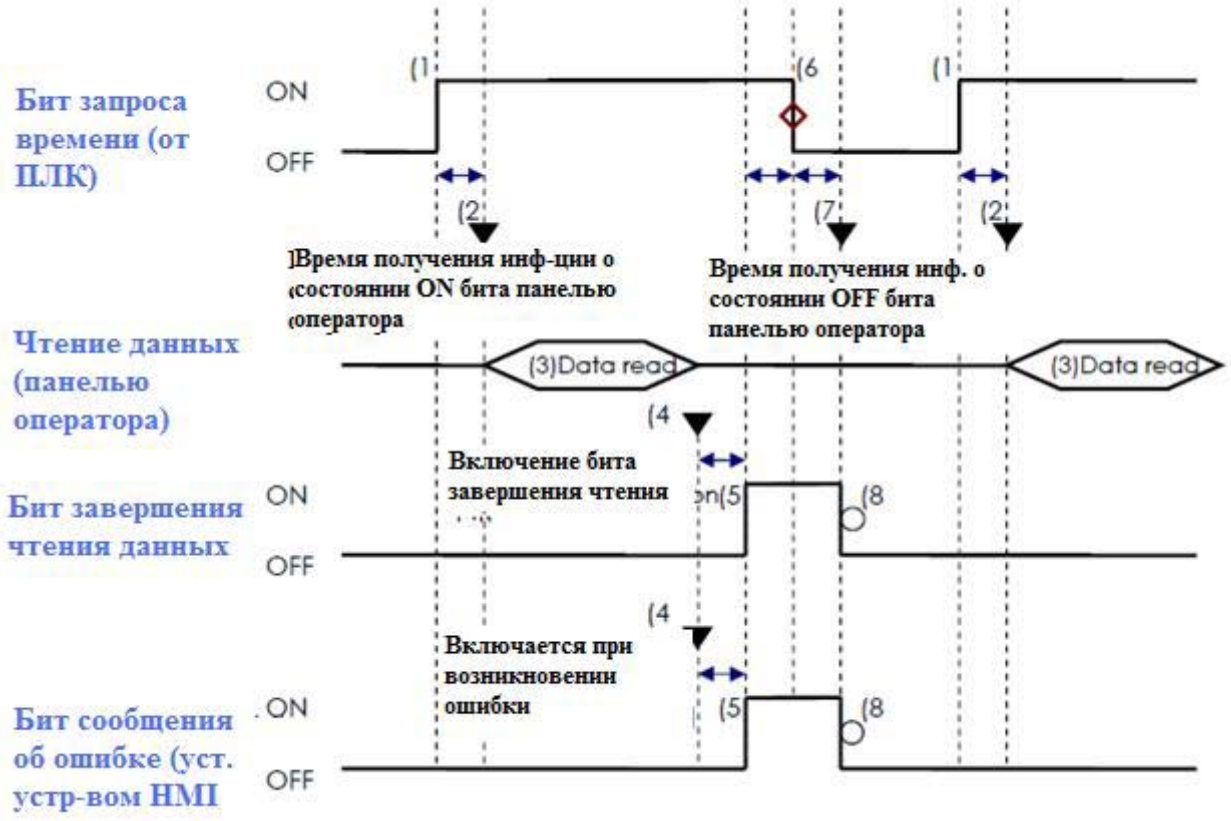
Бит 00: бит завершения чтения (0: null; 1: операция чтения завершена)

Бит 01: бит ошибки (0: нет ошибок, 1: неверный формат начального или конечного времени)

Примечание

После того, как система прочитает время и установит в ON бит [time acquisition complete bit], необходимо убедиться, что нулевой бит [time acquisition request bit] управляющего слова [Control] установлен в OFF.

Как только этот бит будет установлен в OFF, система установит в OFF оба бита: [Status] [time acquisition complete bit] и [error notification bit].



= время задержки связи, ○ = панель устанавливает бит в OFF, ◇ = пользователь устанавливает бит в OFF

c. Action mode (Time setting address +2)

Включение и отключение функций [Termination time action] и [Setting on individual day].

15	02	01	00	Bit
Reserved (0 fixed)		0	0	

Бит 00: Настройка конечного времени [Termination time setting] (0: отключено, 1: включено)
Бит 01: Установка по отдельным дням [Setting on individual day] (0: отключено, 1: включено)

Примечание

1. Если не отмечен флаг [setting on individual day], то будут считаны все 11 адресов, но не учтено конечное время.
2. Если флаг [setting on individual day] отмечен, то нужно убедиться, что введены начальное и конечно время. Если одновременно включены два или более флага конечных/начальных дней, возникает ошибка.

d. Start/End Day (Start Day: Time setting address +3, End Day: Time setting address +7)
Назначение дня запуска начальных/конечных действий.

15 … 07	06	05	04	03	02	01	00	Bit
Reserved (0 fixed)	Sat	Fri	Thu	Wen	Tue	Mon	Sun	

Бит 00: Воскресенье (Sunday) (0: не выбран, 1: выбран)

Бит 01: Понедельник (Monday) (0: не выбран, 1: выбран)

Бит 02: Вторник (Tuesday) (0: не выбран, 1: выбран)

Бит 03: Среда (Wednesday) (0: не выбран, 1: выбран)

Бит 04: Четверг (Thursday) (0: не выбран, 1: выбран)

Бит 05: Friday (0: не выбран, 1: выбран)

Бит 06: Saturday (0: не выбран, 1: выбран)

e. Start/End Time (Start Time: Time setting address +4 to +6, End Time: Time setting address +8 to +10)
Установка значений времени, используемых для начальных/конечных операций в указанном интервале времени.

Час: 0~23

Минуты: 0 -59

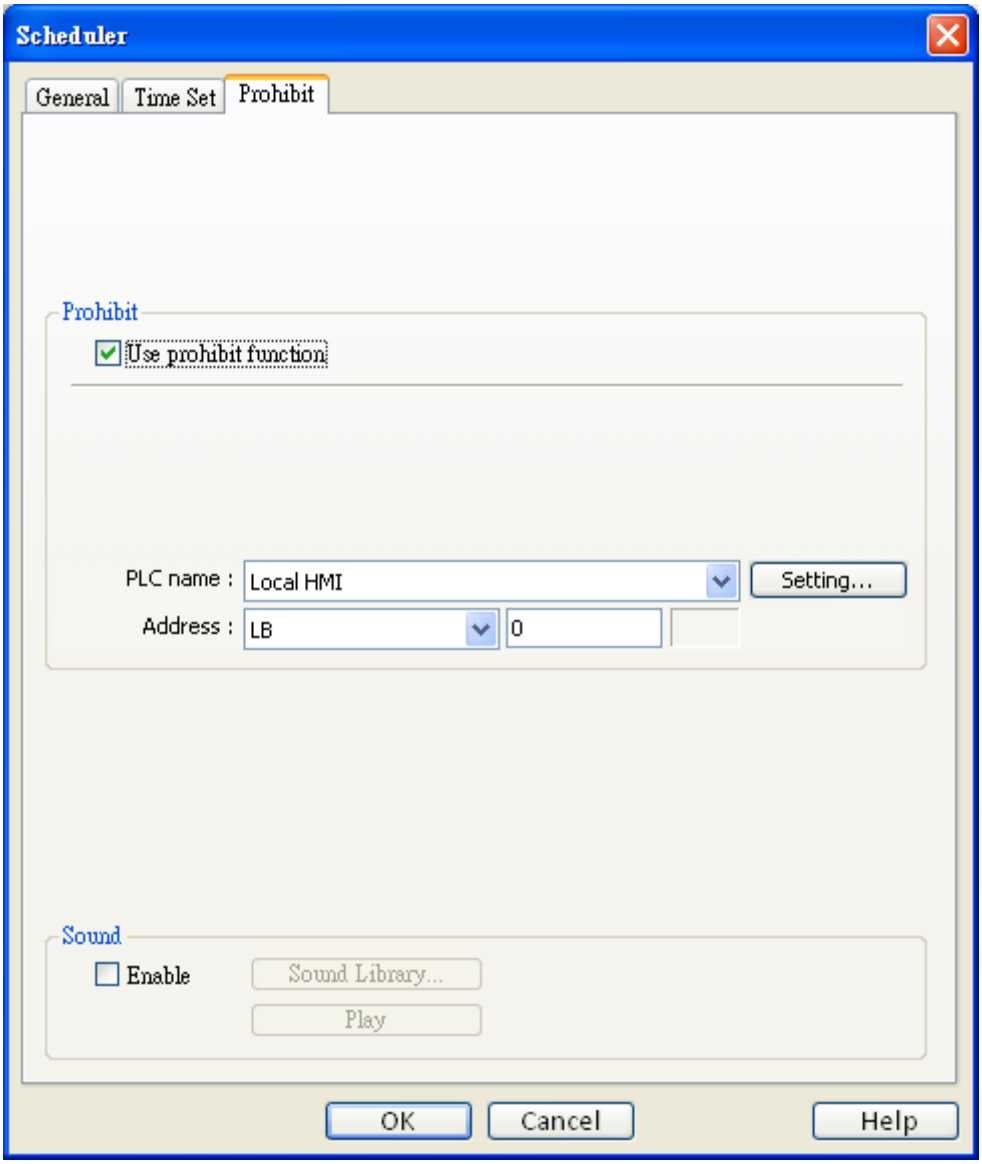
Секунды: 0-59

Если установлены значения за пределами указанных диапазонов, возникнет ошибка.

Примечание

Формат времени должен быть ***16-bit unsigned***, система не поддерживает BCD-формат.

Закладка [Prohibit] –Запрещения



Prohibit - Запретить

- Если включено.

Панель считывает состояние бита перед выполнением начального действия. Если состояние бита ON, то действие по расписанию не выполняется.

Sound - Звук

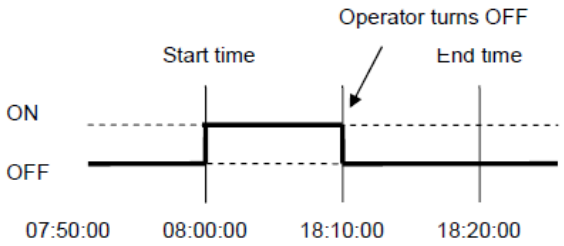
- Если включено

При выполнении начального и конечного действий одновременно будет воспроизводиться заданный звук.

Ограничения

- Пользователь может включить максимум 32 объекта в список [Scheduler].
- Функция включенная в [scheduler] срабатывает только один раз.
  Т.е. при достижении начального [start time] или конечного [end time] времени система один раз записывает значение (действие не повторяется).

Действие: Установка бита
Start time: 08:00:00
End time: 08:20:00



- При выполнении начального действия, система считает значения адресов [Write start address] и [Write end address], после этого, несмотря ни на какие изменения значения в [Write end address], новое значение не будет использоваться.
- При изменении оператором установок системных часов для тех объектов [scheduler], для которых установлены и начальное, и конечное время, система проверяет, не привело ли обновление времени к попаданию в установленный в объекте интервал времени, если это так, то будет выполнено начальное действие.
- Если на одно и то же начальное или конечное время приходится несколько объектов [scheduler], то будут выполняться действия с первого до последнего в порядке возрастания.
- Когда в [Time Set] выбран режим [Address], система будет периодически считывать значение из слова [control].
- Когда в [Time Set] выбран режим [Address] и начальное и конечное время за пределами допустимого диапазона, система не сможет выполнить действие.
- Когда в [Time Set] выбран режим [Address], действие не начнется до успешного обновления информации о времени.

13.29 Объект [Option List]

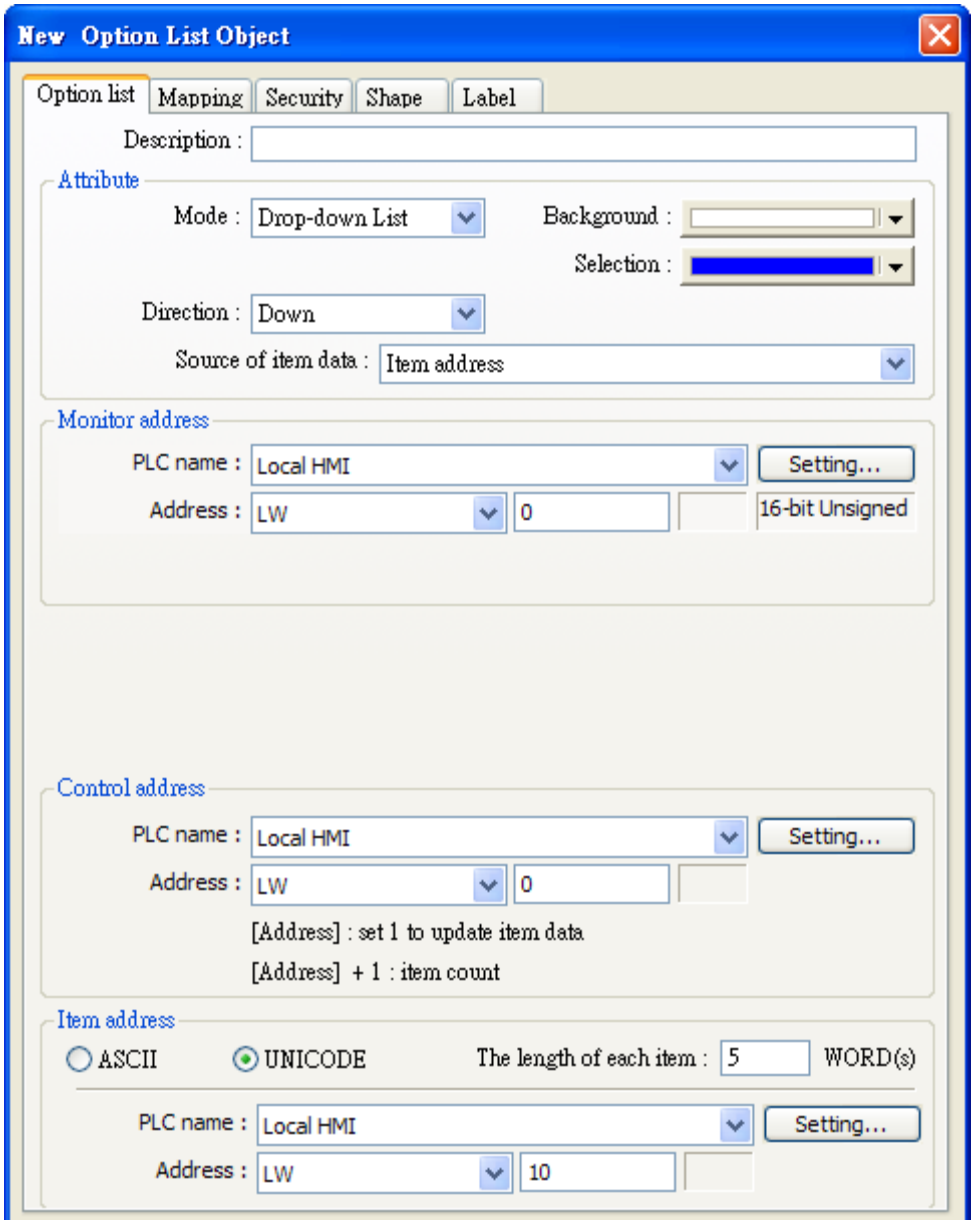
Краткое описание

Объект [Option List] отображает список вариантов, которые пользователь может просмотреть и выбрать. Как только будет выбран конкретный вариант, соответствующее ему значение будет записано по назначенному адресу.
Есть две разновидности этого объекта: Список [Listbox] и Ниспадающий список [Drop-down list]. Первый показывает сразу все варианты и подсвечивает текущий выбор.
Второй показывает только один вариант, остальные выпадают в виде похожего на [Listbox] списка при касании этого объекта.

Настройка свойств

Щелкните по пиктограмме “Option List” — откроется окно “Option List object properties”.

Закладка [Option list]



Attributes

[Mode]
Выбор типа объекта: Listbox или Drop-down list.

[Item No.]
Установка числа состояний (элементов) объекта. Каждый элемент представляется состоянием отображонное в списке и значением, заносимым в [Monitor address].

[Background]
Выбор цвета фона объекта.

[Selection]
Настройка цвета фона для выбранной/подсвечиваемой строки.

[Source of item data]
Можно выбрать источник данных: «Predefine», «Dates of historical data», и «Item address»

Режим «Predefine» - предопределенный

Monitor address

В полях **[PLC name], [Device type], [Address]** укажите адрес, управляющий отображением объекта, и куда система заносит значение состояния.

[Write when button is released] – Записать при отпускании кнопки

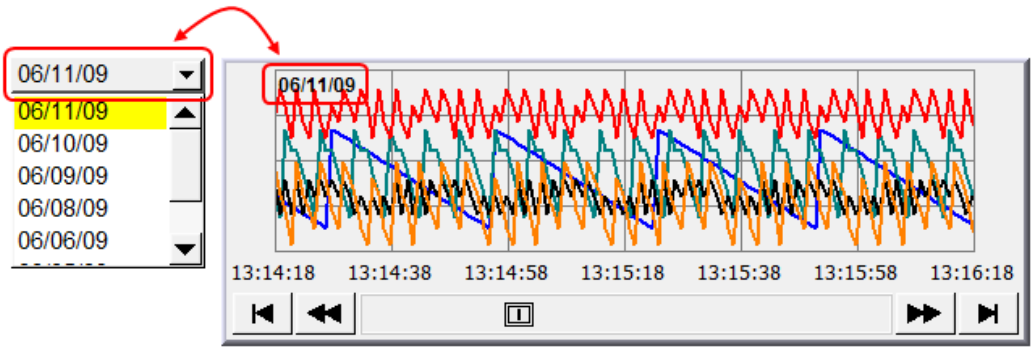
Если функция выбрана, операция будет активирована при прекращении касания.
Если функция не выбрана, операция будет активирована при касании.

Примечание Данная опция доступна только для стиля *Listbox*.

Режим «Dates of historical data » - Данные из архива

Данные состояний из архова (History index mode)

Объект *Option List* может быть использован со следующими дисплеями: *Historical Event-Display, Trend-Display* и *Data-Display* для отображения Файлов архива - History File на *Historical Display.*



[Type]

Архив - *Alarm (Event) log* использован для отображения на *Historical Event-Display*.

Данные опроса - Data sampling использованы для отображения на *Historical Trend-Display* или *Data-Display*.

[Date]
Настройка формата данных.

[Data Sampling object]

Пользователи должны выбрать, какой *Data sampling* - объект запускается при выборе “Data sampling” в качестве типа [Type].

Пользователи должны выбирать тот же самый *Data sampling* – объект, который выбран в *Historical Trend-Display* или *Data-Display*.

Примечания

1. Система будет автоматически отключать функцию *Mapping table* при выборе режима *History Index mode*.

2. Если пользователь выбрал ”*Drop-Down List*” в [Attribute] и активировал режим *History Index mode*, объект *Option List* отобразит “?” в *Error State*.

Режим «Item address mode» - Адрес элемента

Если выбран режим [Item address], пользователь должен корректно указать содержимое [Control address] и [Item address].

Control address

[Address]
Запишите “1” в регистр этого адреса для обновления элементов в *Option List* , используя содержимое регистра [Item address].
После обновления, содержимое этого регистра будет восстановлено в “0”.

[Address] + 1
Адрес следующий за [Control address] предназначен для указания числа элементов.

Item address
Это адрес сохранения содержимого элемента.

[ASCII]
Использование символов ASCII, в качестве содержимого элемента.

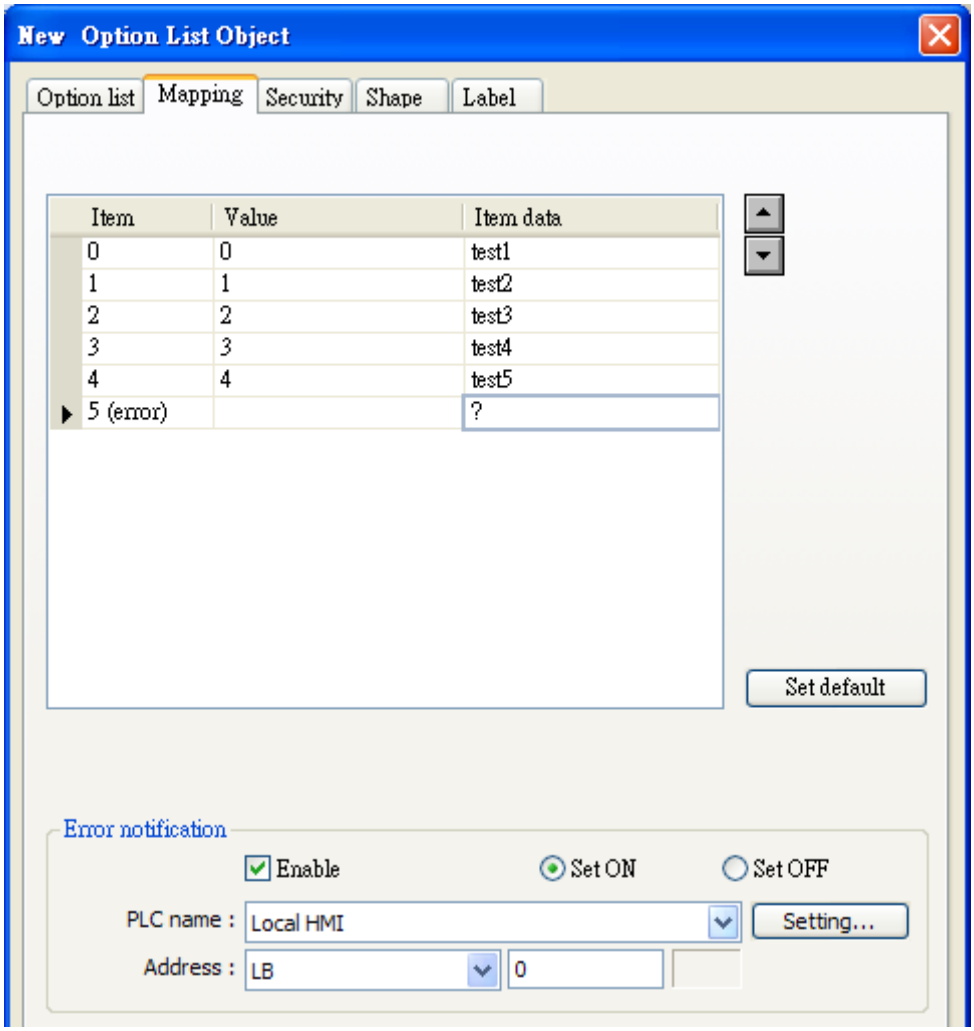
[UNICODE]

Используйте UNICODE в качестве содержимого элемента, для Китайских символов, например. UNICODE может быть использован и в других объектах.
EasyBuilder Pro будет затем компилировать файлы шрифтов, и сохранять их в панели при загрузке, только таким способом UNICODE может отображаться корректно.

[The length of each item]
Длина каждого элемента сейчас ограничена 1024.

Примечание Система будет автоматически отключать функцию *Mapping table* при выборе режима *Item address mode* .

Закладка [Mapping] – Отображение элементов (привязка)



Mapping table

Эта таблица показывает все доступные состояния/элементы (states/items), их названия и соответствующие значения.
Чтобы изменить числов доступных элементов. надо: [Option list tab] → [Attributes] → [Item No.].

[Item]
Система представляет все доступные элементы. Каждый элемент представляет состояние. отображаемое в списке. Поле только для чтения.

[Value]
Пользователь может назначить значение для каждой элемента, опираясь на два условия:

a. [For reading] Если обнаружено любое изменение в [Monitor address], объект сравнивает содержимое адреса с указанным значением и выбирает первый совпадающий элемент. Если ни в одной строке не найдено совпадений, происходит переход в состояние ошибки и записывается сигналы предупреждения (если требуется).
b. [For writing] - Данное значение записывается в [Monitor address], когда выбран данный элемент.

[Item data]

Пользователь может назначить название (data) каждого элемента. Объект будет отображать эти значения, чтобы пользователь мог видеть все элементы и выбирать элемент в списке.

[Error state]

a. Например: элемент *item 8* – это состояние ошибки, когда в параметр [No. of state] задан равным 8. Аналогично, если задать его равным 11, то 11 состояние будет использоваться для ошибки и т.д.

b. При состоянии ошибки в *listbox* будет снята подсветка для того, чтобы показать, что не выбран ни один вариант, *drop-down list* будет отображать состояние ошибки.

c. Элемент ошибочного состояния используется только разновидностью *drop-down list*. Стиль *listbox* не использует этот элемент.

[Set default] .

Установка значений по умолчанию для всех состояний (то есть установка 0 для *item 0*, 1 для *item 1*, и так далее).

Error Notification

Будет установлено значение ON/OFF для назначенного бита при возникновении ошибки. Такой сигнал об ошибке может быть использован для запуска процедуры по ее устранению.

13.30 Timer - Таймер

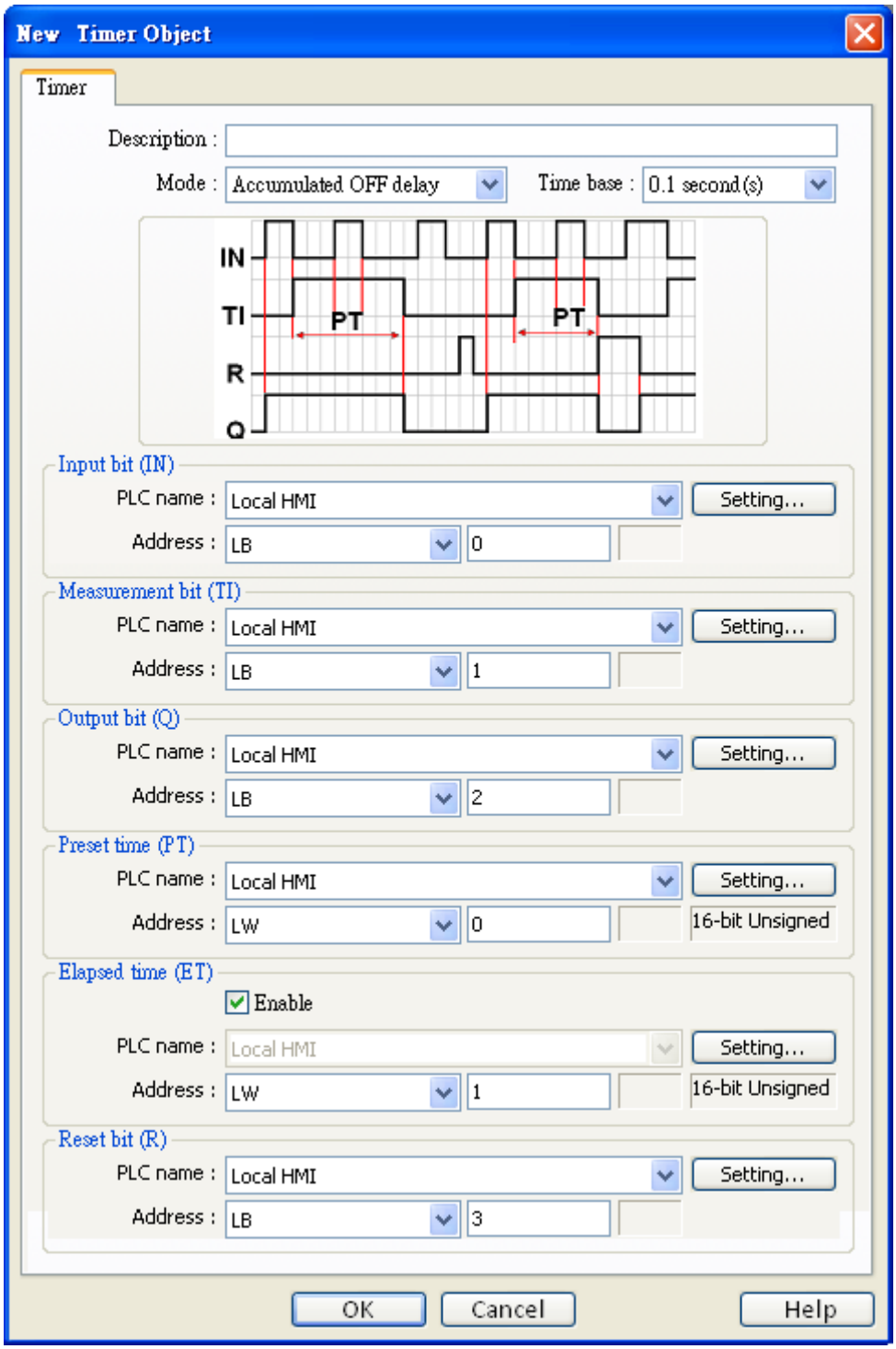
Краткое описание

Используйте переменные объекта *Timer,* для активации команд таймера. Существуют следующие переменные таймера.

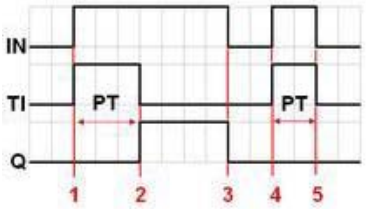
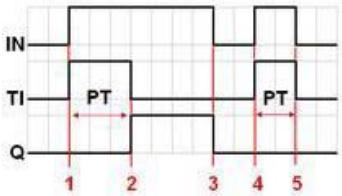
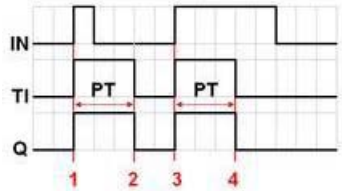
Переменные объекта *Timer*	**Тип переменной**	**Описание**
Input bit (IN)	Bit	Ведущий переключатель таймера
Measurement bit (TI)	Bit	Переходит в «ON» при начале счета
Output bit (Q)	Bit	Переходит в «ON» при окончании счета
Preset time (PT)	Word	Уставка таймера
Elapsed time (ET)	Word	Текущее (прошедшее) значение таймера
Reset bit (R)	Bit	Сброс прошедшего времени в «0»

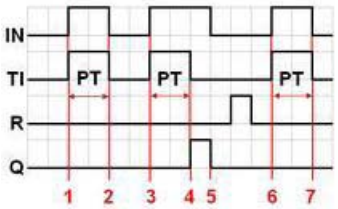
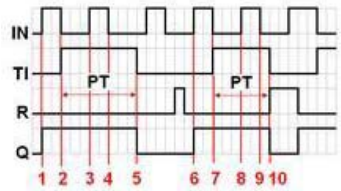
Настройка свойств

Щелкните по пиктограмме “Timer” — откроется окно “Timer object”.



Mode - Режим	Описание
On delay – Задержка включения	**Точка 1**: Когда вход **IN** переходит в состояние ON, **TI** перейдет в ON и прошедшее время *ET* увеличивается. Выход **Q** остается в состоянии OFF. **Точка 2**: когда **ET** равно **PT**, выход **Q** переходит в состояние ON и **TI** перейдет в OFF.

Mode - Режим	Описание
	**Точка 3**: Когда вход **IN** переходит в состояние OFF, the **Q** перейдет в OFF и время **ET** будет сброшено в «0». **Точка 4**: Когда вход **IN** переходит в состояние, **TI** перейдет в ON и прошедшее время **ET** увеличивается. **Точка 5**: При переходе **IN** в состояние OFF до достижения **ET** значения **PT**, **TI** перейдет в OFF, и **ET** сбросится в «0». (**Q** останется в состоянии OFF)
Off delay – Задержка отключения 	**Точка 1**: Когда вход **IN** переходит в состояние ON, **TI** остается в состоянии OFF и **Q** перейдет в ON. **Точка 2**: Когда вход **IN** переходит в состояние OFF, TI перейдет в ON и прошедшее время **ET** увеличивается. (**Q** остается в состоянии ON) **Точка 3**: Когда **ET** равно **PT**, **Q** и **TI** переходят в OFF. **Точка 4**: Когда вход **IN** переходит в состояние ON, **Q** переходит в состояние ON и **ET** сбрасывается в 0. **Точка 5**: Когда вход **IN** переходит в состояние OFF, the TI перейдет в ON и прошедшее время **ET** увеличивается (**Q** остается в состоянии ON) **Точка 6**: При переходе **IN** в состояние OFF до достижения **ET** значения **PT**, **TI** перейдет в OFF, и **ET** сбросится в «0». (**Q** останется в состоянии ON)
Pulse 	**Точка 1**: Когда вход **IN** переходит в состояние ON, **TI** и **Q** переходят в ON, и прошедшее время **ET** увеличивается. **Точка 2**: Когда **ET** равно **PT, TI** и **Q** переходят в OFF. **Точка 3**: Когда вход **IN** переходит в состояние ON, **TI** и **Q** переходят вON, и прошедшее время **ET** увеличивается. **Точка 4**: Когда **ET** равно **PT**, **TI** и **Q** переходят в OFF.

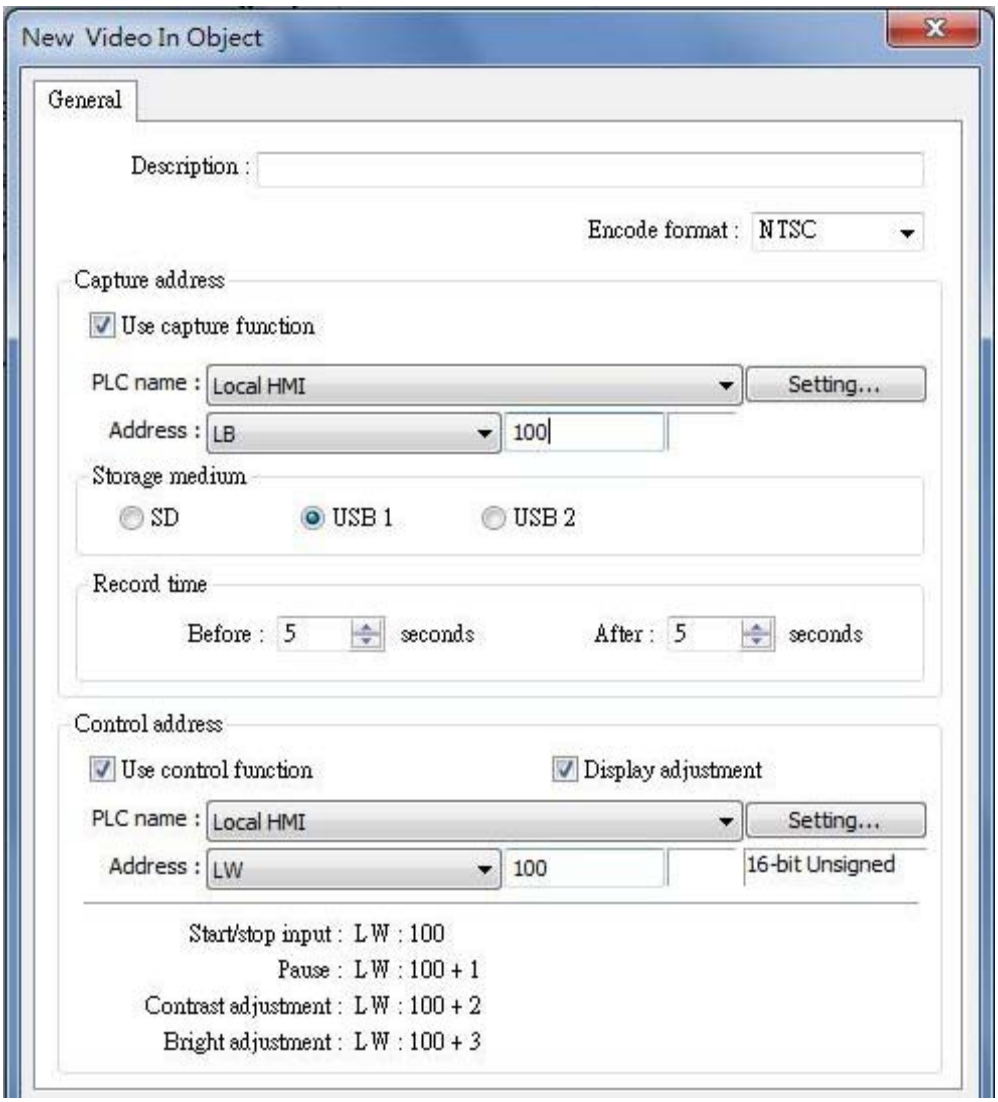
Mode - Режим	Описание
Accumulated On delay – **Аккумулированная задержка на включение** 	**Точка 1**: Когда вход **IN** переходит в состояние ON, **TI** перейдет в состояние ON и прошедшее время **ET** увеличивается (**Q** остается в состоянии ON). **Точка 2**: Когда вход **IN** переходит в состояние OFF, и если **ET** меньше **PT**, **TI** перейдет в состояниеOFF. **ET** в состоянии сохранения значения. **Точка 3**: Когда вход **IN** переходит в состояние ON, **TI** перейдет в состояние ON. Работа таймера начинается опять и **ET** добавляется к сохраненному значению. **Q** остается в состоянии OFF. **Точка 4**: Когда значение **ET** достигнет значения **PT**, **TI** перейдет в OFF и **Q** перейдет в ON. **Точка 5**: Когда вход **IN** переходит в состояние OFF, **Q** перейдет в состояние. (Сброс **ET** в «0», используя бит сброса - Reset bit (**R**))
Accumulated Off delay – **Аккумулированная задержка на выключение** 	**Точка 1**: Когда вход **IN** переходит в состояние ON, **Q** перейдет в ON и **TI** остается в состоянии OFF. **Точка 2**: Когда вход **IN** переходит в состояние OFF, TI перейдет в состояние ON и прошедшее время **ET** увеличивается. (**Q** остается в состоянии ON) **Точка 3**: Когда вход **IN** переходит в состояние ON, таймер переходит в состоянии «пауза». **Точка 4**: Когда вход **IN** переходит в состояние OFF, находившийся в состоянии «пауза» таймер продолжает измерения. **Точка 5**: Когда **ET** сравнивается с **PT**, **TI** и **Q** переходят в OFF. (Сброс **ET** в «0», используя бит сброса - Reset bit (**R**))

13.31 Video In – Ввод видео сигнала

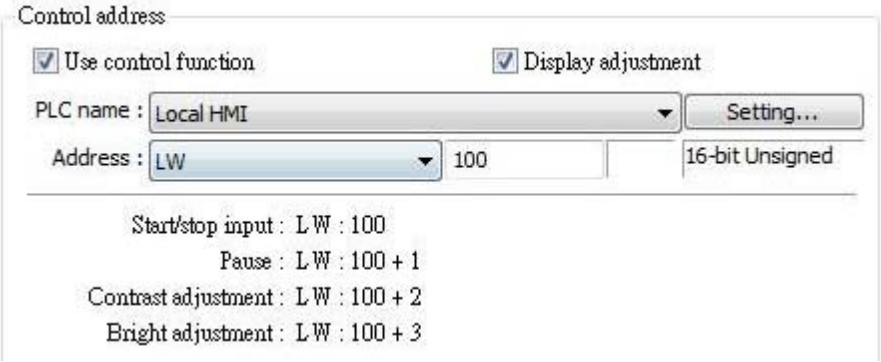
Панель оператора имеет функцию приема видеосигнала - **Video Input**. Пользователи могут установить камеру наблюдения и контроллировать производство в любой момент времени. Видео изображения могут быть сохранены в устройствах и , затем, прсмотрены при помощи Media Player, или использованы для анализа на компьютере.

Эта функция может быть использована нескольких целях. Кроме наблюдения за процессом, она может быть применена, контроля работы приводов и устройств строительной техники, например.

Аппаратно панель поддерживает 2 канала **Video Input**. Пользователи могут свободно переключать каналы для наблюдения, и сохранять изображения без даже, когда просмотр в режиме «Пауза». Сохраненные изображения будут копией текущего изображения с внешнего входа. Поддерживаемые форматы: NTSC и PAL.



Использование функции управления - Control Function
Сделайте отметку в поле **[Use control function]**



Предположим, что **[Control Address]** - “LW100”:

A. Пользователи могут использовать [Control Address+ 0] для активирования / останова (enable/stop) функции *Video Input* .

[LW100] = 0 → Stop Playing – (останов просмотра).

[LW100] = 1 → Изображение ç *Input video* VIP 1 отображается на экране.

[LW100] = 2 → Изображение ç *Input video* VIP 2 отображается на экране.

[LW100] = 3 → Изображение ç *Input video* VIP 1, но не отображается на экране.
Таким способом пользователи могут выполнить *Capture image* – Сохранение изображения

[LW100] = 4 → Изображение ç *Input video* VIP 2, но не отображается на экране.
Таким способом пользователи могут выполнить *Capture image* – Сохранение изображения.

B. Пользователи могут использовать [Control Address +1] для управления отображением видеоизображений:
[LW101] = 1 →Pause/Continue – (Пауза/Продолжение).

C. Если пользователи изменят значение [Control Address + 0], система сохранит новое значение.

D. Если пользователи изменят значение [Control Address + 1], система исполнит сначала соответствующую команду , а затем сотрет новое значение и установит его обратно в “0”.

E. Если не использовать **[Control Function]**, система будет работать с каналами настроенными в соответствии с **[Input channel]** автоматически.

Если отметить **[Display adjustment] – Настройка изображения**

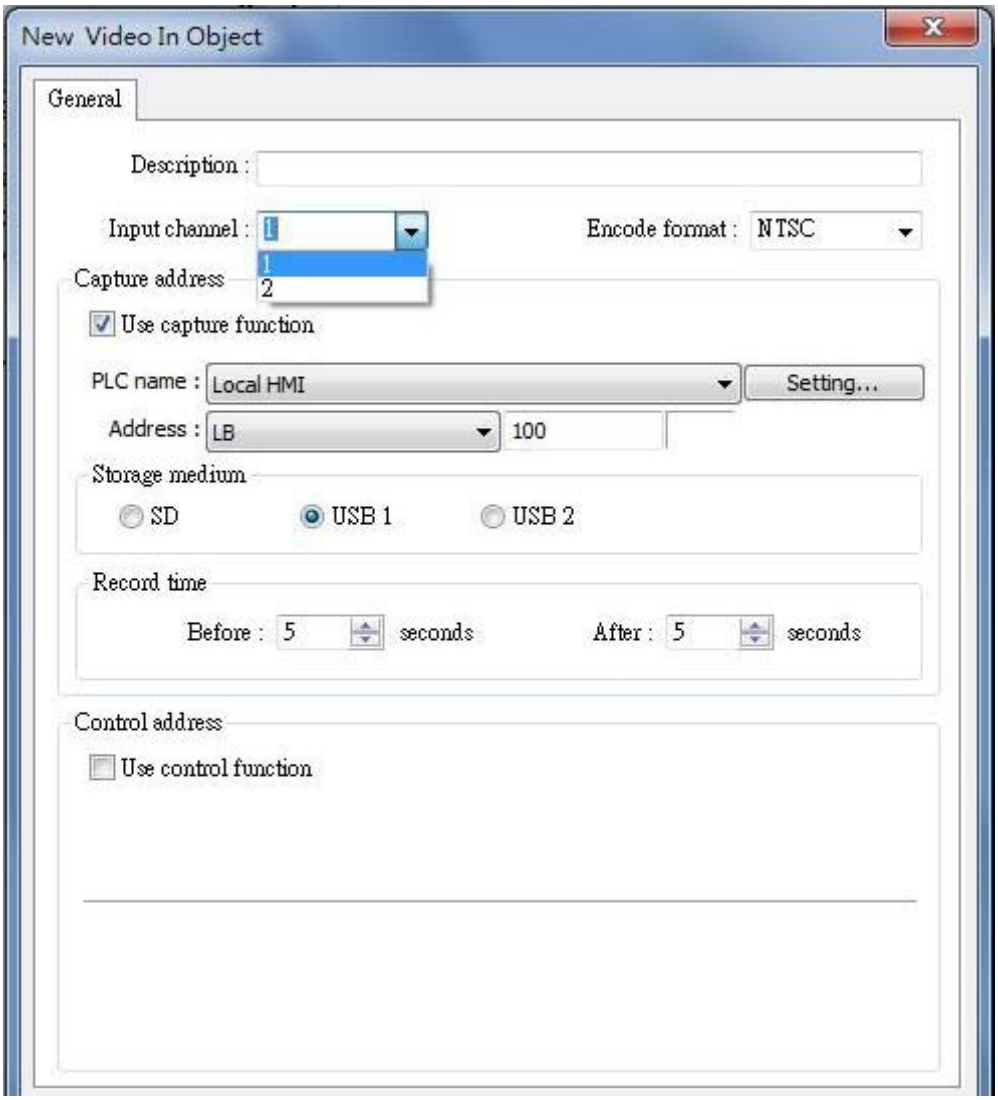
То можно настроить яркость и контрастность изображения.

Если назначить “LW100” в качестве *Control Address* :

A. Настройка соотношения контрастности - Contrast Ratio [Control Address + 2]: LW102,

диапазон: 1~100.

B. Настройка яркость - Brightness [Control Address + 3]: LW103,
диапазон: 1~100.



Использование функции сохранения изображения - Capture Function

Определение: Сохранение изображенис с видео-входа.

Иллюстрация:

A. [Capture address] -*Control Address* который запускает систему насохранение видео-изображения.

B. [Storage medium] – Выбор места сохранения видео-изображения.

Доступные места сохранения: карта SD , USB1 или USB2.

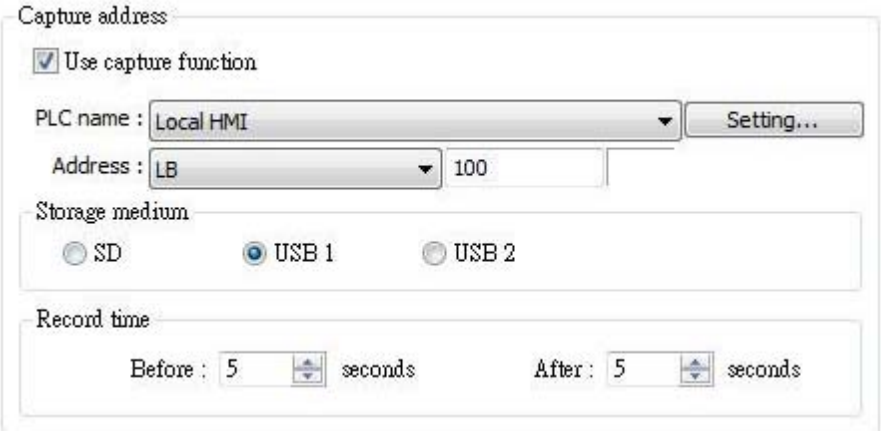
- Изображение канала VIP 1 будет сохранено в файле «VIP 1» в указанном месте, а изображение канала VIP 2 в файле «VIP2».

C. [Record time] – Время записи. Для настройки периода времени сохранения изображения.

- The longest period can be set starts from 10 seconds before triggering **[Capture address]** to 10 seconds after triggering. In this case there will be 21 images captured, including the one captured at the triggering moment.

- Интервал времени для сохранения один раз в секунду.

- Сохраненный файл «.jpg» будет именован следующим образом:
До или после [Capture address] срабатывает: YYYYMMDDhhmmss.jpg
Момент в который [Capture address] срабатывает: YYYYMMDDhhmmss@.jpg



В качестве примера (приведенного выше):

Настройка **[Record time]** “Before - До” и “After - После” равна “5” seconds, когда **[Capture address]** изменит состояние с OFF на ON.

Система будет запускать сохранение изображения, одно изображение в секунду, начиная за 5 секунд до запуска и заканчивая через 5 после времени запуска.

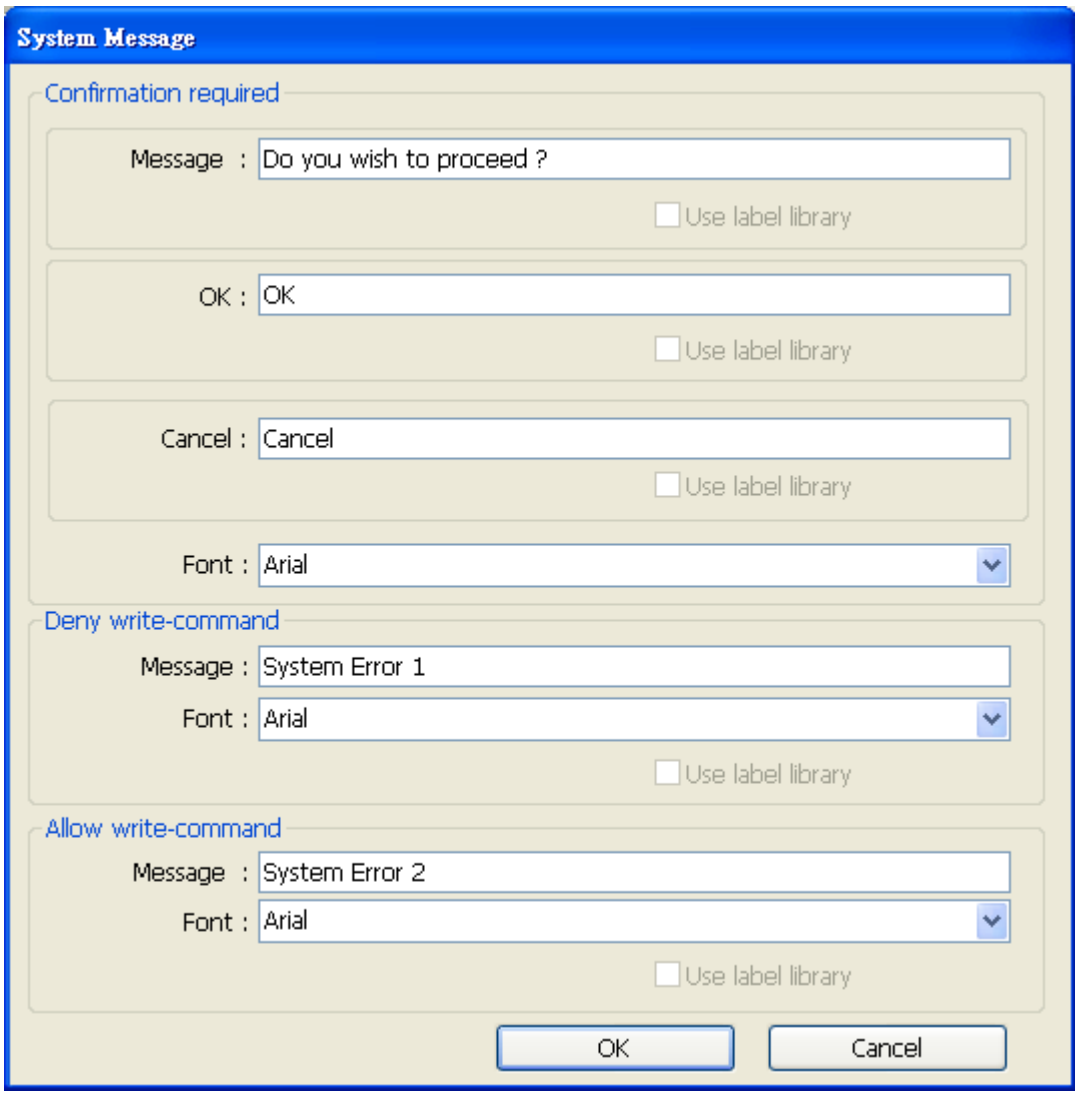
ПРИМЕЧАНИЕ:

1. Объект *Video In* может быть использован только в панелях, которые поддерживают функцию VIP.
2. Только одно видеоизображение в одном канале может быть использовано одновременно в работающей системе.
3. Функция сохранения сообщений не испытывает влияния “Паузы” просмотра.
4. Рекомендованный формат и разрешение:

	1:1	50%
NTSC	720 x 480	360 x 240
PAL	720 x 576	360 x 288

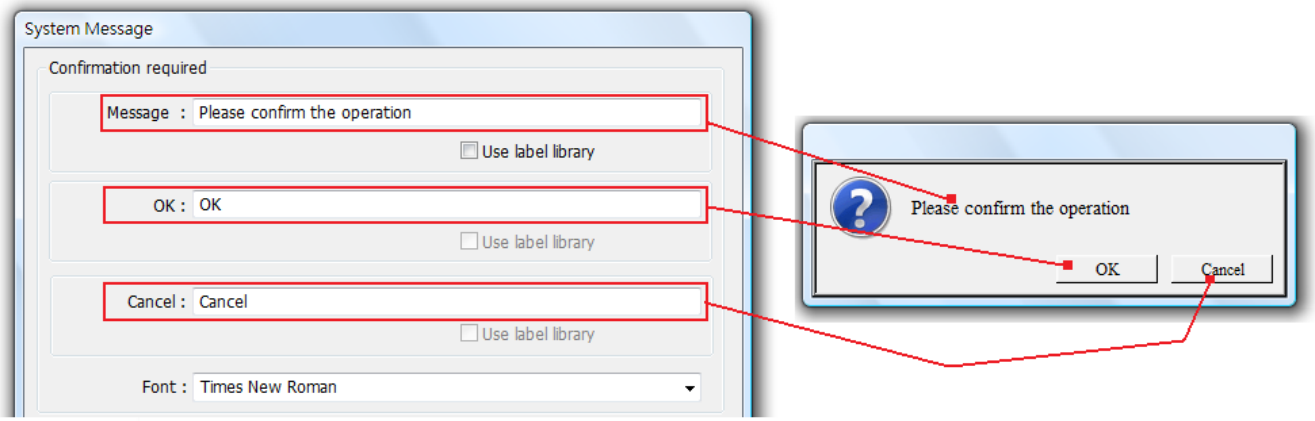
13.32 System Message – Системное сообщение

Эта утилита предназначена для редактирования сообщений, которые отображаются во всплывающих окнах сообщений - *message boxes*.



Confirmation required – Требуется подтверждение

Отображается всякий раз, когда безопасность требует подверждение операций.
Могут быть настроены: Сообщение [Message] появляющееся в диалоге подтверждения, и тектовый заголовок с двумя кнопками [OK] и [Cancel].
Используйте одни и те же шрифты для надписей [Message], [OK] и [Cancel].
Дополнительно, только при выборе [Label Library] для [Message], возможно использование библиотеки *Label Library* для [OK] и [Cancel].



Deny write-command – Запретить команды записи

Отображается, когда системный тэг LB-9196 (локальная панель поддерживает только функции просмотра) переведен в состояние ON.

Allow write-command – Разрешить команды записи

Отображается, когда системный тэг LB-9196 (локальная панель поддерживает только функции просмотра) переведен в состояниеOFF.

13.33 Recipe View – Просмотр рецептов

Краткое описание

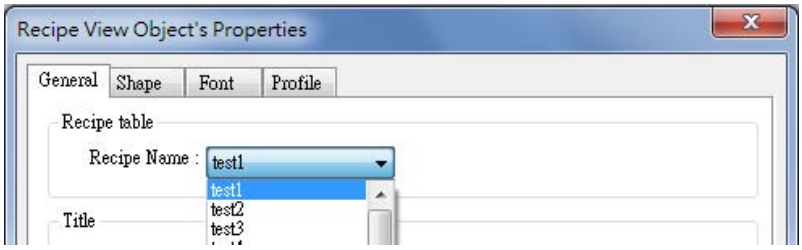
Объект *Recipe View* может быть использован для отображения специфических данных рецептов. Пользователи могут просматривать все позиции и значения рецептов, при помощи этого объекта.

Настройка свойств

Щелкните по пиктограмме “Recipe View” — откроется окно “Recipe View properties”. Заполните все пункты, нажмите и новый объект просмотра рецептов будет создан.

Общий вид.

Recipe table –Таблица рецептов



[Recipe Name] – Имя рецепта
Выберите желаемое имя рецепта или просмотрите другие рецепты из ниспадающего списка.

Title - Заголовок

Каждая позиция имеет заголовок. Заголовок используется в ссылках [System Parameter Setting]

[Transparent] - Прозрачность
Если выбран этот вариант, у заголовка не будет цвета фона (background color). Далее, эта настройка не будет появляться в качестве варианта выбора цвета.

Profile

Здесь можно выбрать рамку (frame) и цвет фона (background color) объекта.

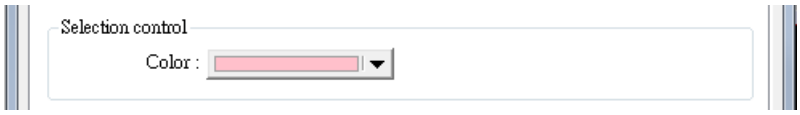
[Transparent]
Если выбран этот вариант, цвет фона (background color) не будет показан. Далее, эта настройка не будет появляться в качестве варианта выбора цвета.

Grid - Сетка

Линии разделяющие каждое отдельное данное.

[Transparent]
Если выбран этот вариант, линии раздела не будут показаны. Далее, эта настройка не будет появляться в качестве варианта выбора цвета.

Selection Control – Контроль выбора



Цвет выбранного ряда при просмотре данных рецепта.

Пользователям необходимо создать данные рецепта до использования объекта Recipe View , просмотрите главу 5 – *System Parameter Settings*. Кроме того, создайте записи рецептов при помощи объекта Recipe Records, просмотрите главу 24 - *Recipe Editor* для получения более подробной информации.

Как просматривать или модифицировать записи рецептов - Recipe Records?

Чтобы Просматривать/добавлять/Удалять показанные записи, регистр может быть настроен на ввод специфического значения.
Создадим, сначала, 4 объекта типа Numeric Input, адреса: Selection, Count, Command, и Result.

[Selection] –Выбор

No	16-BCD	32-BCD	16-Hex	32-Hex	1
0	11.11	66.66	1111	AAAA	0
1	22.22	77.77	2222	BBBB	0
2	33.33	88.88	3333	CCCC	0

Выбор записи,начиная с «0».
При выборе первой записи в поле *Selection* будет показан “0”, и так далее.
Как показано на рисунке, выбрана подсвеченная розовым цветом запись и в поле *Selection* - “1”.

1 Selection

[Count]

No	16-BCD	32-BCD
0	11.11	66.66
1	22.22	77.77
2	33.33	88.88
3	44.44	99.99
4	55.55	12.34

Число записей в текущем рецепте.
Как показано справа, есть 5 записей, и поле *Count* будет “5”.

5 Count

[Command]

No	16-BCD	32-BCD
0	11.11	66.66
1	22.22	77.77
2	33.33	88.88
3	44.44	99.99
4	55.55	12.34
1	22.22	77.77

Ввод определенного значения будет посылать команду для выбранной записи.

Ввод “1”, Добавит новую запись *Recipe Record* к последнему ряду.

Ввод “2”, Обновит указанную запись.

Ввод “3”, Удалит указанную запись.

[Result]

Просмотр результата выполнения команды.

Если показана цифра “1”, Команда успешно выполнена.

Если показана цифра “2”, Выбранная запись не существует.

Если показана цифра “4”, Неизвестная команда.

Если показана цифра “8”, Записи достигли предела (10000 записей), не возможно добавить новые записи.

Глава 14. Библиотека форм и Библиотека изображений

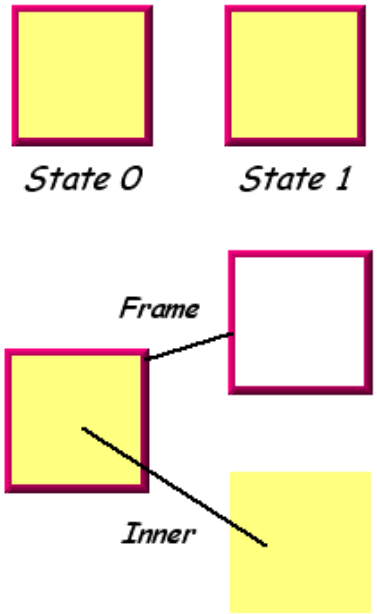
EASYBUILDER PRO позволяет работать с Библиотеками форм и изображений (Shape Library и Picture Library) для увеличения наглядности объектов. Каждая фигура и изображение имеет до 256 состояний. В этой главе объясняется, как создавать библиотеки -Shape Library и Picture Library.

Для получения информации об использовании Библиотек Форм и изображений см. также главу 9 «Общие свойства объектов».

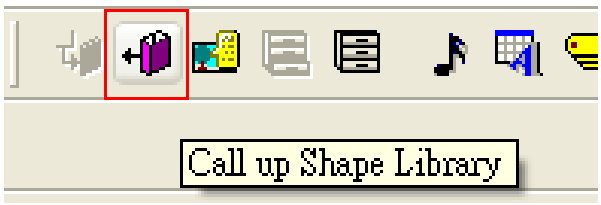
14.1 Создание Библиотеки форм

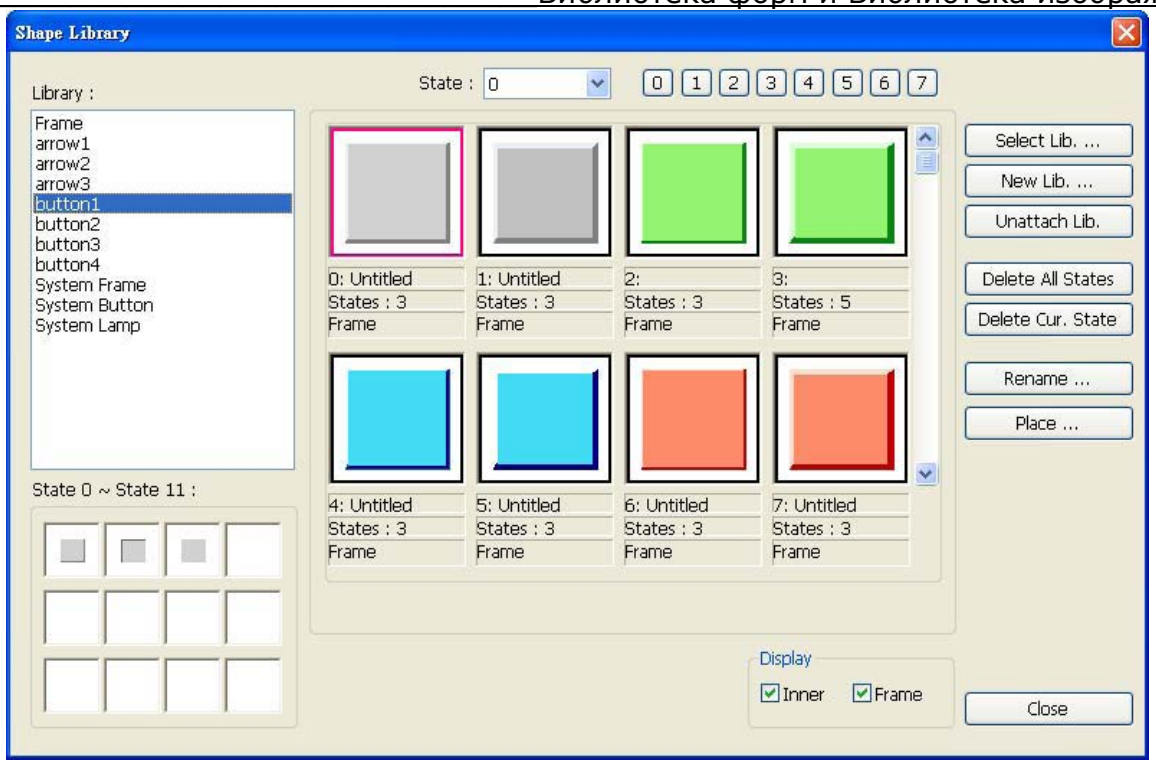
Форма (Shape) — это графический объект, состоящий из линий, прямоугольников и окружностей. Форма может иметь более одного состояния, и каждое из них включает две части: рамку (frame) и внутреннюю область (inner).

См. рисунок ниже.

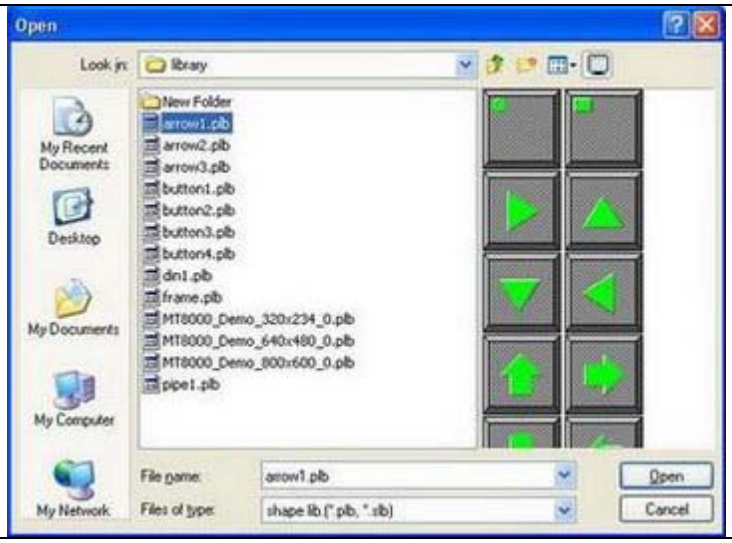
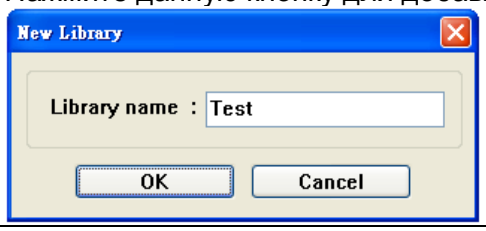
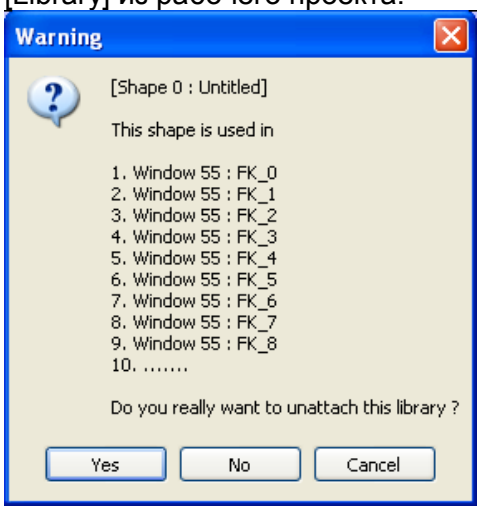
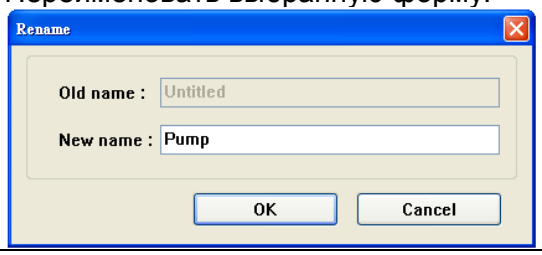


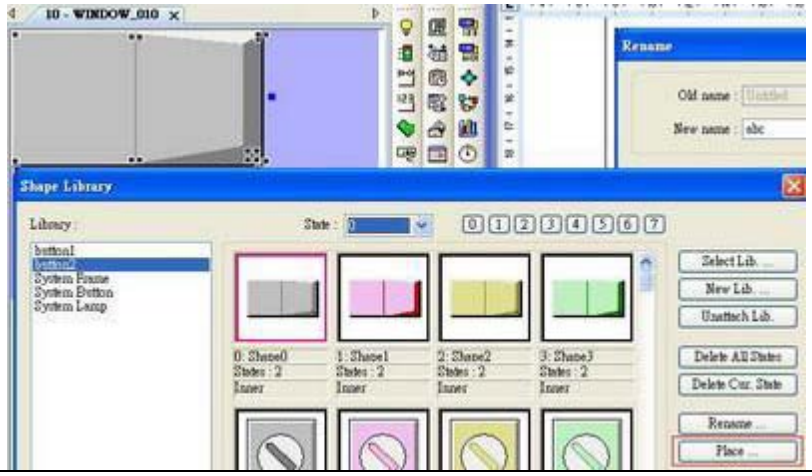
В настройках объекта может быть использована рамка формы, внутренняя область или и то и другое. Щелкните пиктограмму **[Call up Shape Library]** на панели инструментов — появится диалоговое окно **[Shape Library]:**





Настройка	Описание
[Library]	Выберите Библиотеки форм , которые должны быть добавлены в рабочий проект. Выберите источник библиотеки и формы из списка.
[State]	Выбор отображаемого состояния текущей формы. Если выбранная форма не отображается, значит, она не существует или данное состояние для неё не определено.
[Select Lib. …]	Нажмите кнопку **[Select Lib. ...],** и появится диалоговое окно, показанное ниже для определения пути файла добавляемой Библиотеки форм. Просматривая содержимое библиотек в правой части окна, пользователь может выбрать подходящую библиотеку.

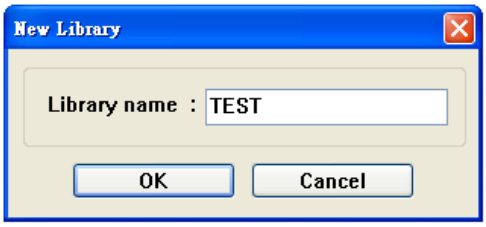
	
[New Lib. ...]	Нажмите данную кнопку для добавления новой библиотеки форм. 
[Unattach Lib.]	Нажмите на данную кнопку для исключения Библиотеки форм в списке [Library] из рабочего проекта. 
[Delete all States]	Удаление всех состояний выбранной формы.
[Delete Cur. State]	Удаление текущего состояния выбранной формы.
[Rename ...]	Переименовать выбранную форму. 

[Place ...]	Экспортировать фигуру для размещения в текущем окне. 

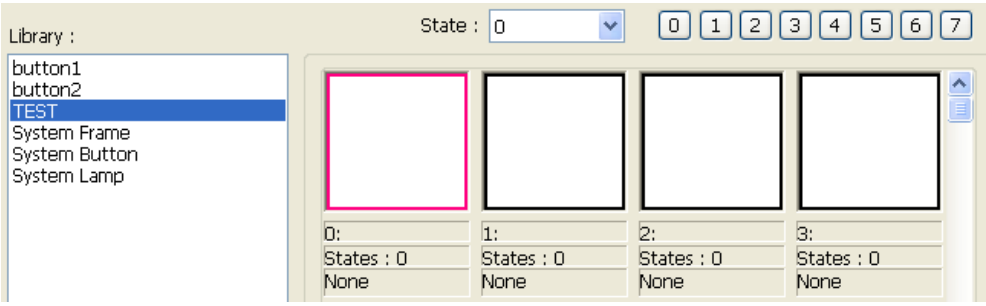
Ниже приведен пример создания Библиотеки форм и добавления в нее формы с двумя состояниями.

Шаг 1

Нажмите **[New Lib. ...]** и введите имя новой Библиотеки форм.

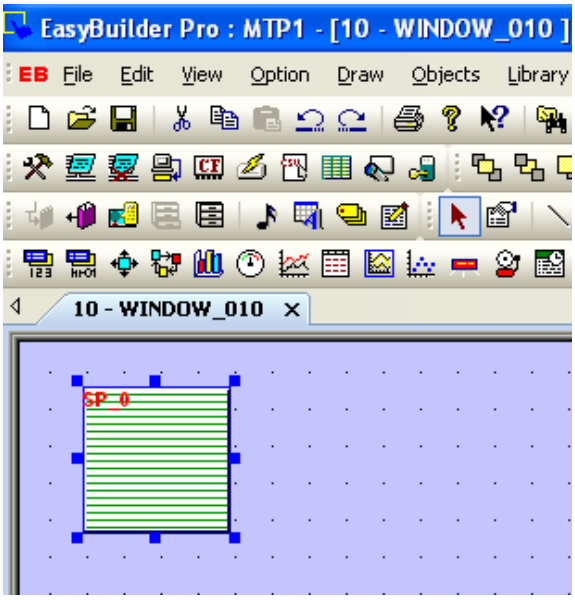


Новая Библиотека с именем “TEST” будет добавлена в диалог [Shape Library]. В настоящий момент в Библиотеке нет ни одной формы.

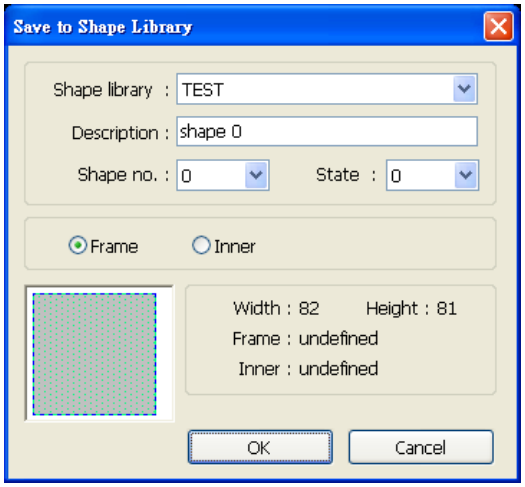


Шаг 2

Добавьте состояние выбранной формы. Для этого используя графические средства, создайте в окне фигуру и выберите ее для добавления в Библиотеку.



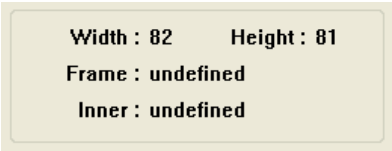
Щелкните кнопку **[Save to Shape Library]** на панели инструментов — появится диалоговое окно:



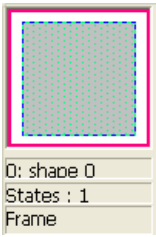
Настройка	Описание
[Shape library]	Выберите библиотеку, в которую будет добавлен данный графический элемент. В данном примере выбрана “TEST”.
[Description]	Название формы.
[Shape no.]	Номер, который будет присвоен данной форме в библиотеке.
[State]	Выберите состояние, которому соответствует данная форма. В данном случае установлено значение состояния 0. Всего в EASYBUILDER PRO доступны 256 состояний для каждой формы.

Настройка	Описание
[Frame]	Если отмечен данный флаг, то выбранный графический объект будет являться рамкой формы.
[Inner]	Если выбран данный флаг, то выбранный графический объект станет внутренней областью формы.

На рисунке же показана информация о форме. Ни рамка, ни внутренняя область не определены в текущем состоянии (state 0).

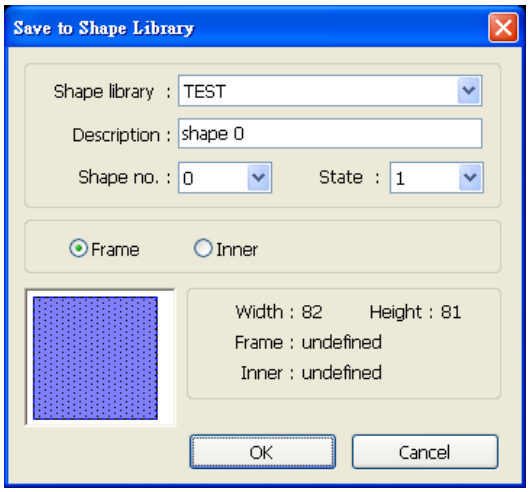


После нажатия OK графический элемент будет добавлен в Библиотеку. Кроме того, будет показано, что фигура Shape No.0 (с нулевым номером) имеет только одно состояние, рамка также будет определена.

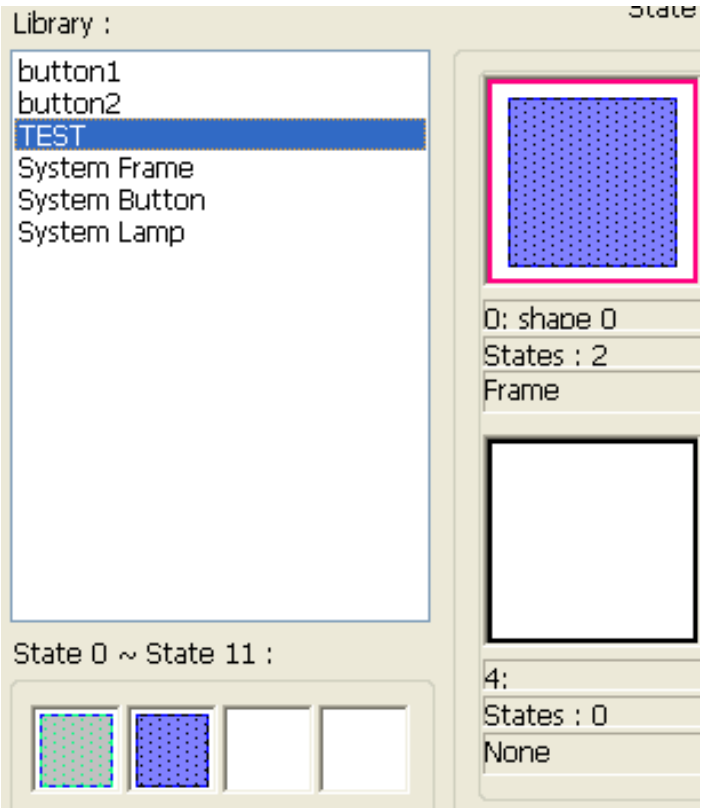


Шаг 3

Аналогичным образом создайте другое состояние, только в данном случае графический элемент нужно определить как состояние 1 **[state 1].**

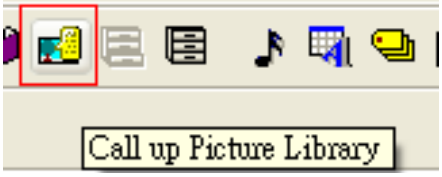


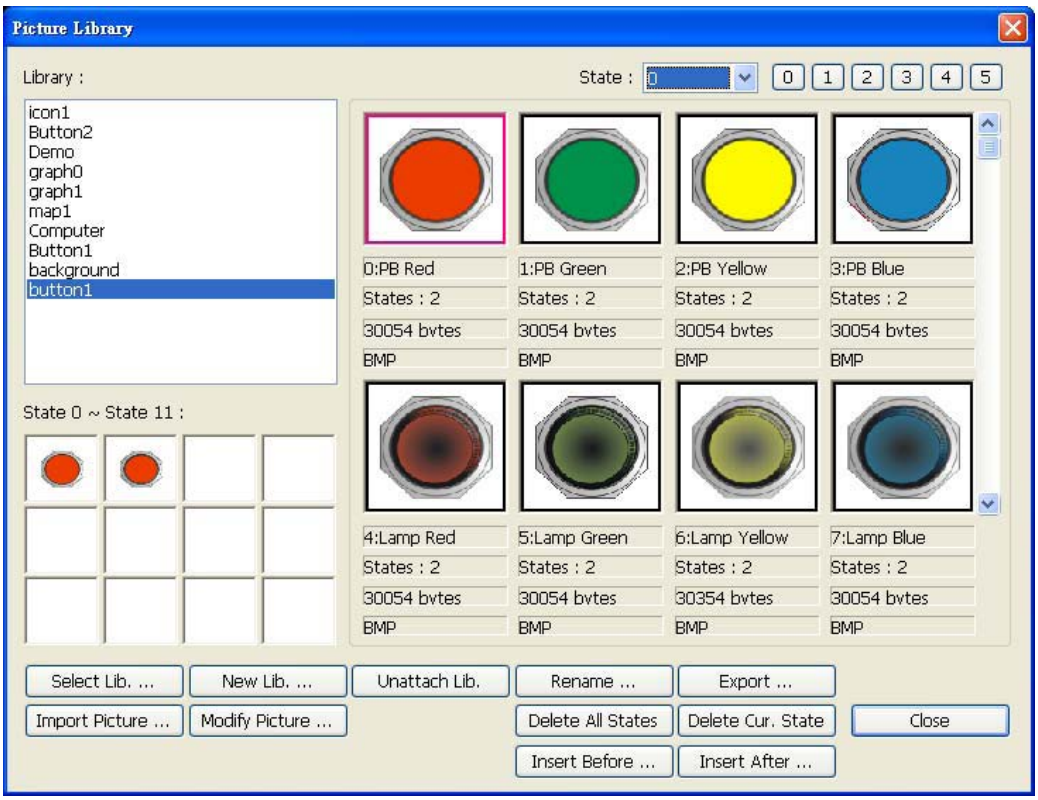
Нажмите OK, и будет создана законченная форма с двумя состояниями. См. рисунок ниже.



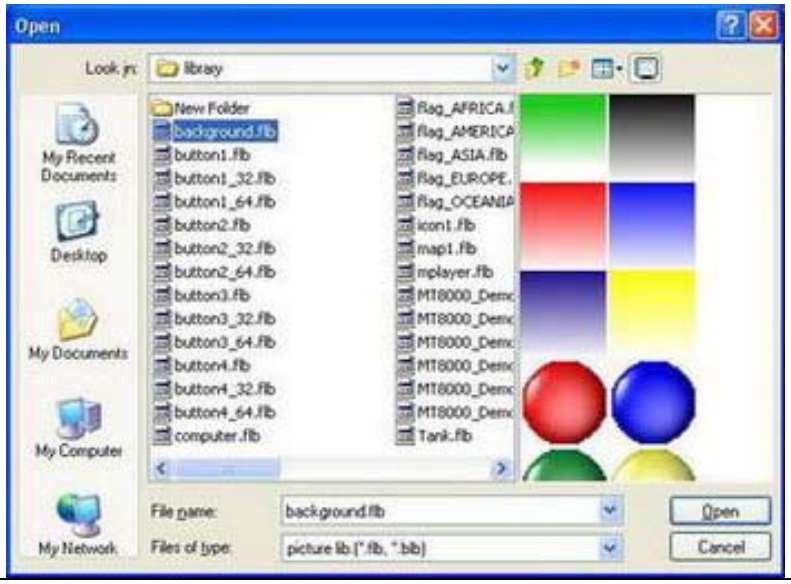
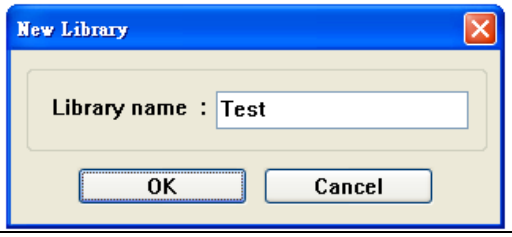
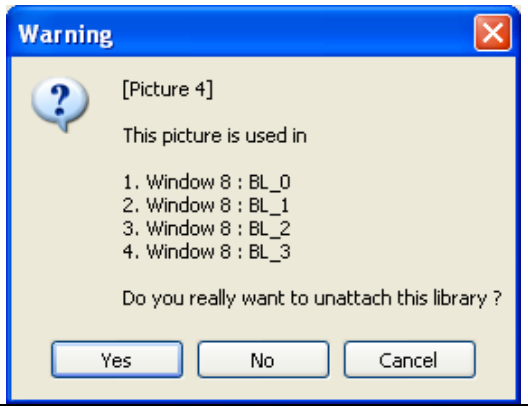
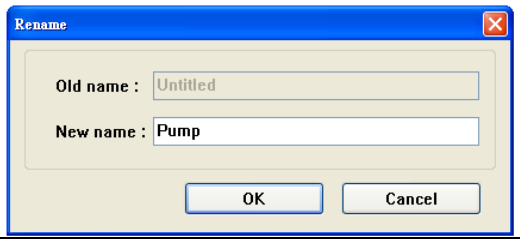
14.2 Создание Библиотеки изображений

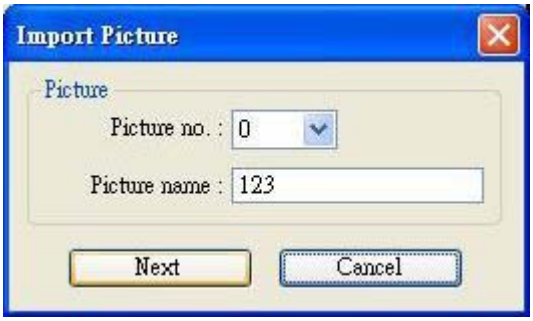
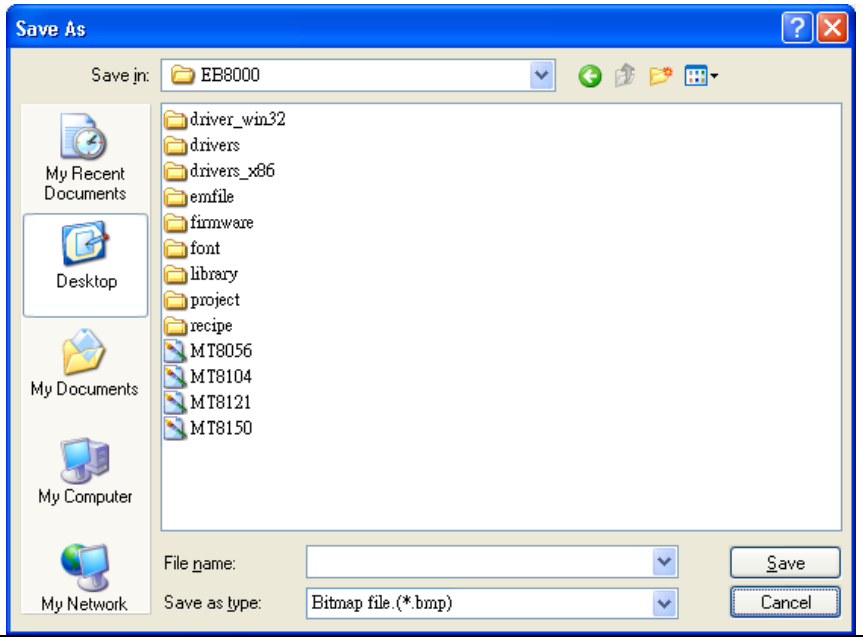
Щелкните пиктограмму **[Call up Picture Library]** на панели инструментов — откроется диалоговое окно **[Picture Library].**





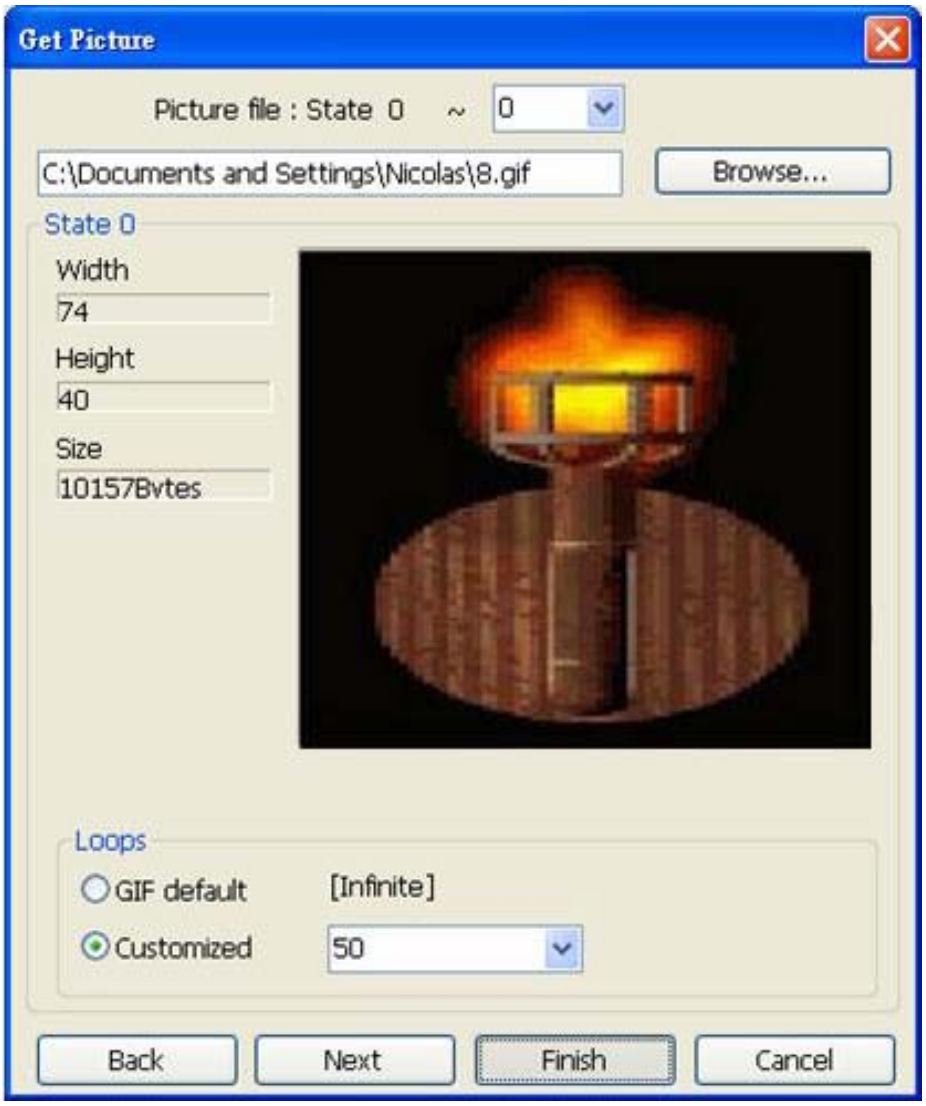
Настройка	Описание
[Library]	Библиотеки изображений, добавленные в рабочий проект. Выберите библиотеку из списка.
[State]	Выберите отображаемое состояние текущего графического объекта. Если выбранное изображение не отображается, значит оно не существует или данное его состояние не определено.
[Select Lib.]	Нажмите кнопку [Select Lib. ...] откроется диалоговое окно для указания пути файла добавляемой библиотеки изображений. Просматривая содержимое библиотек в правой части окна, пользователь может выбрать подходящую.

	
[New Lib.]	Нажмите данную кнопку для добавления новой Библиотеки Изображений. 
[Unattach Lib.]	Щелкните данную кнопку для удаления библиотеки изображений в списке [Library] из текущего проекта. 
[Delete all States]	Удалить все состояния выбранного изображения.
[Delete Cur. State]	Удалить текущее состояние выбранного изображения.
[Rename]	Переименовать выбранное изображение. 
[Insert Before]	Добавить новое состояние перед текущим.
[Insert After…]	Добавить новое состояние после текущего.

[Import Picture]	Добавить новое изображение в *Picture Library* 
[Modify Picture]	Редактировать выбранное изображение
[Export …]	Экспортировать выбранное изображение в указанное место. Как показано ниже, пользователь может работать с исходным изображением. 

ПРИМЕЧАНИЕ: Совместимый формат изображений - *.bmp, *.jpg, *.gif, *.dpd, и *.png.

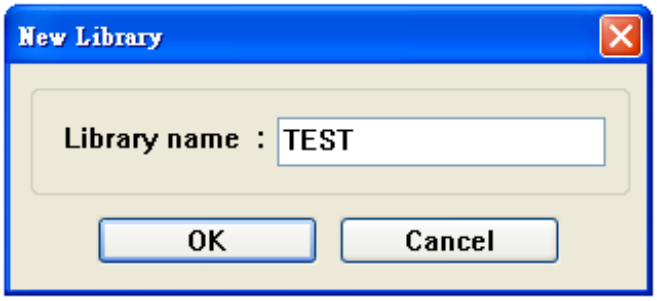
При добавлении изображений GIF в Picture Library, если этот файл изображения анимирован, число раз проигрывания этой анимации может быть настроено пользователем, как показано далее below.



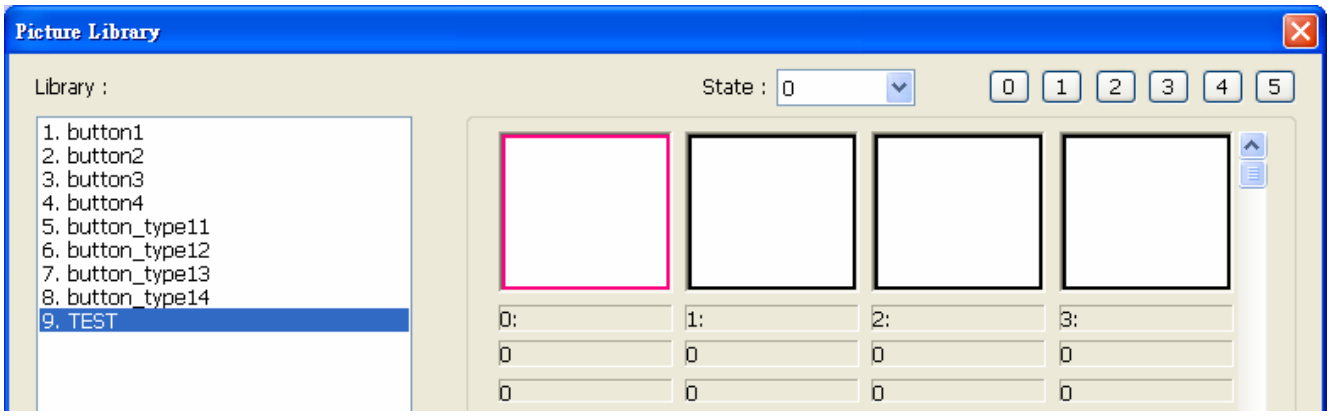
Ниже дан пример создания новой Библиотеки Изображений и добавления в нее изображения с двумя состояниями.

Шаг 1

Нажмите кнопку **[New Lib. ...]** и введите имя новой Библиотеки изображений.

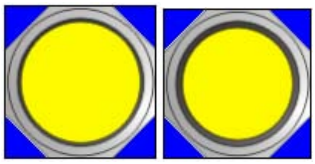


Новая Библиотека “TEST” будет добавлена в список [Library] в диалоговом окне. В данный момент эта Библиотека пуста.

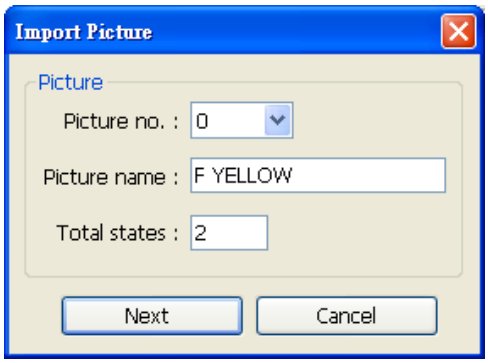


Шаг 2

Подготовьте изображения для добавления в библиотеку; предположим два графических объекта, показанные ниже, используются для представления состояний 0 и 1 соответственно (state 0 и state 1).



Нажмите кнопку **[Import Picture]** — появится окно, показанное ниже. Назначьте **[Picture no.]** и **[Picture name]**, затем щелкните **[Next].**

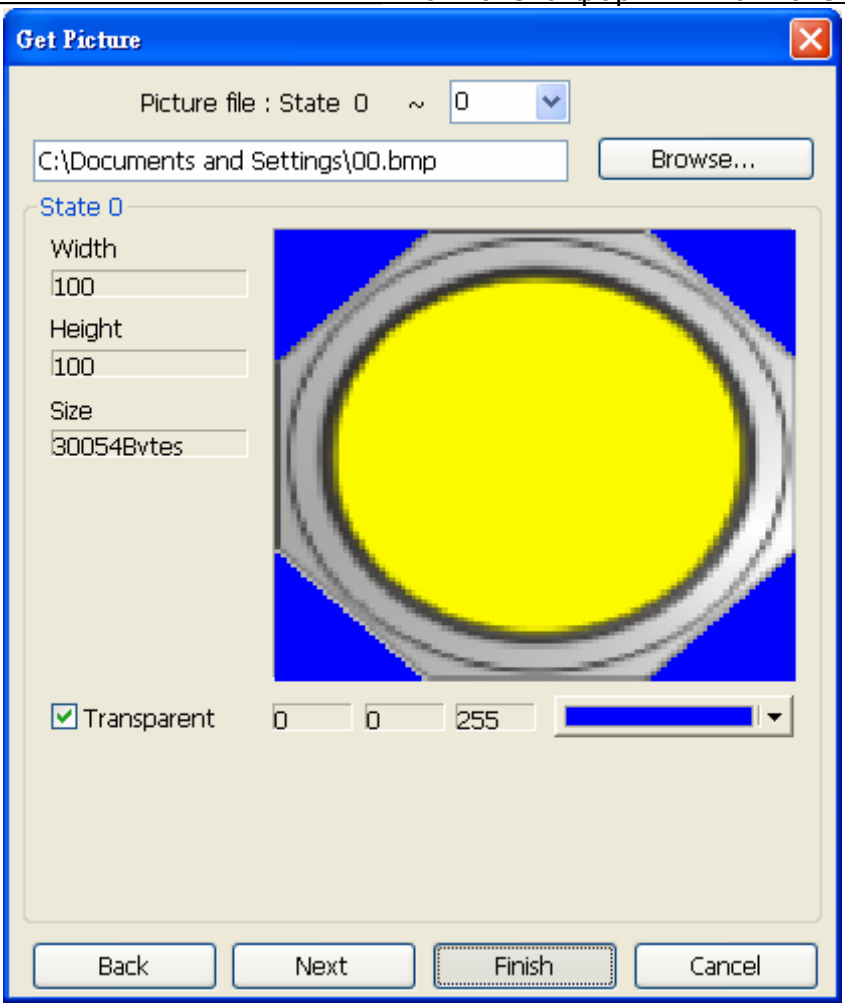


Шаг 3

Когда появится диалоговое окно, показанное далее, выберите библиотеку, состояние state 0, укажите прозрачный (transparent) цвет.

В примере ниже, голубой цвет по системе RGB (0, 0, 255) является прозрачным.

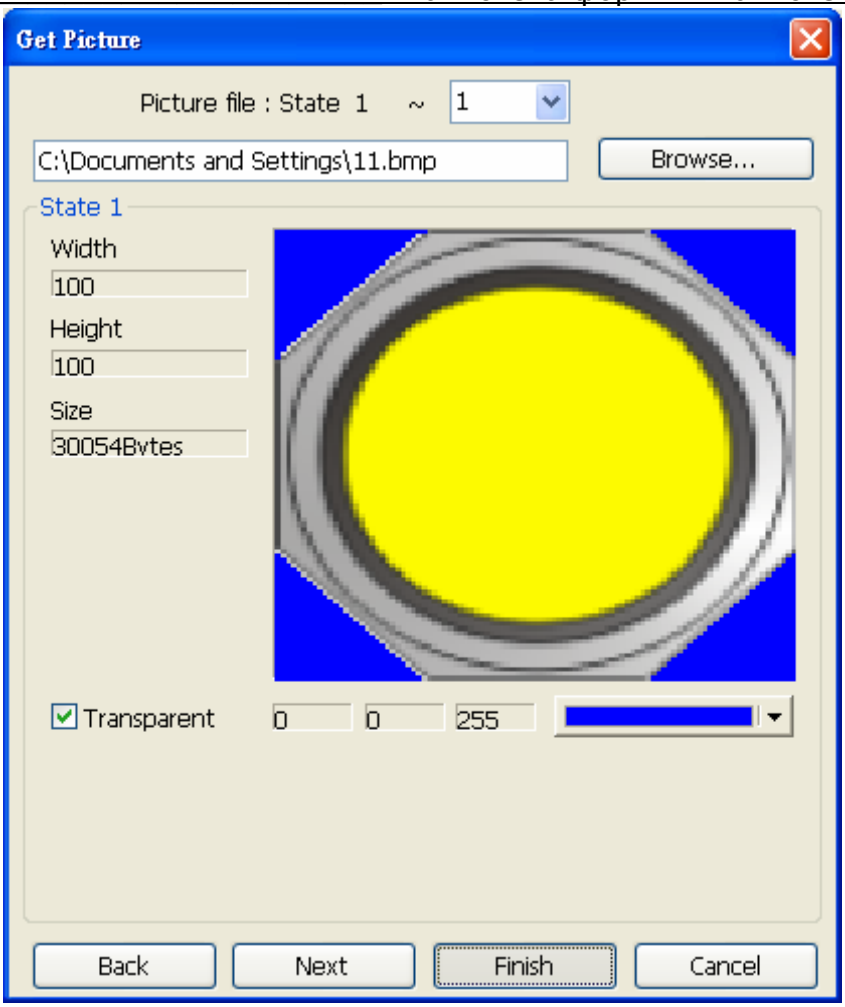
После завершения настроек состояния 0, нажмите [next] для продолжения установок другого состояния.



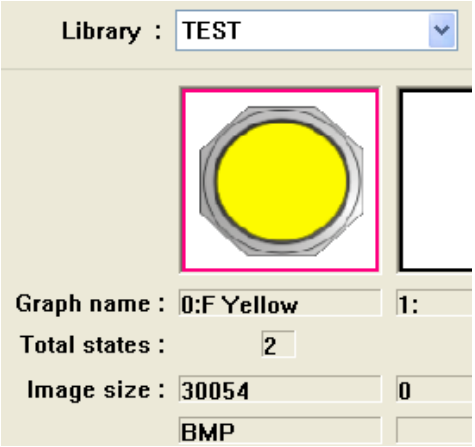
Перед выбором прозрачного цвета, отметьте поле **[Transparent]** и затем щелкните мышью в области данного цвета в окне графического элемента. EASYBUILDER PRO автоматически покажет RGB-код этого цвета. Выше приведен пример, в результате изображение будет выглядеть так:

Шаг 4

Аналогичным образом выберите изображение для состояния 1 и укажите прозрачный цвет для него. После завершения настроек, нажмите кнопку **[Finish].**



Ниже показано создание законченного изображения. Новое изображение “F Yellow” находится в диалоговом окне управления библиотеками изображений [Picture Library]. Из представленной информации видно, что формат данного изображения [bitmap] и оно имеет два состояния.



Глава 15. Библиотека меток и использование нескольких языков

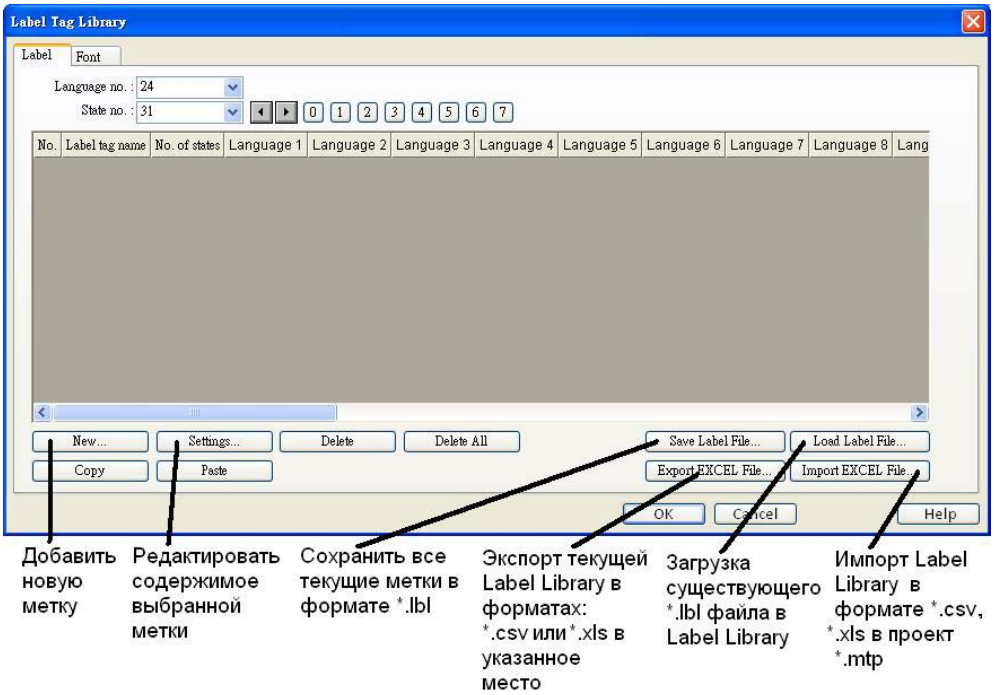
Библиотека меток (надписей) используется в многоязыковой среде. Пользователи могут разработать содержимое Библиотеки меток в соответствии с конкретными требованиями. Затем можно выбрать подходящую метку из Библиотеки, когда требуется ввести текстовую надпись.

15.1 Введение

При работе система будет отображать текст, соответствующий языку, заданному в настройках.

EasyBuilder Pro поддерживает работу с 8 языками одновременно.
Щелкните по пиктограмме **[Label Library Manager]** на панели инструментов и откроется диалоговое окно.

[State no.]
Показывает текущее состояние; каждая метка имеет до 256 состояний (0-255). Число состояний [State no.] зависит от выбранного числа языков **[Language no.]**. Если использовано от 1 до 3-х языков, максимальное значение *state no*. равно 256.
Если использовано более 4-х языков , разделите 768 на число языков, чтобы узнать максимальное число состояний. «max. state number.



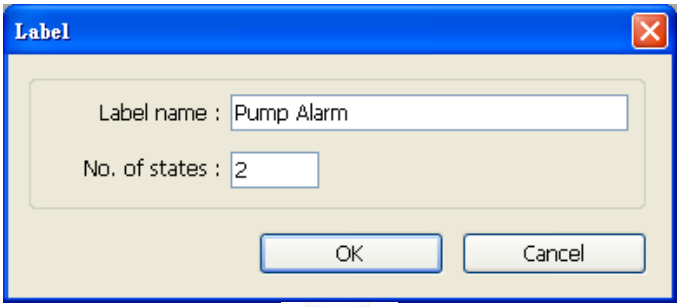
Например: число языков = 24, тогда число состояний = 768/24=32 (states)

При импорте и экспорте в EXCEL, UNICODE не поддерживается

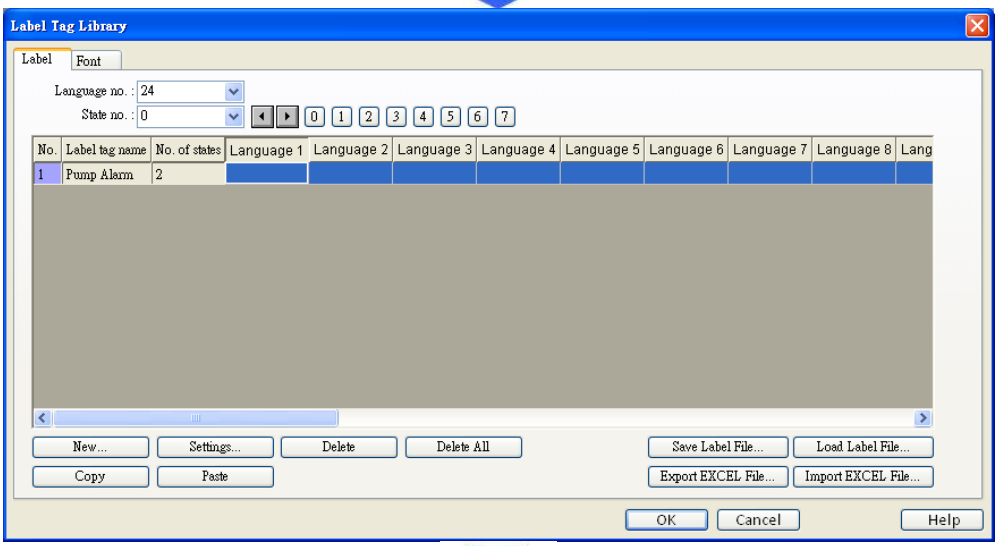
15.2 Создание Библиотеки меток

1. Откройте **Label Tag Library] -> [New]**

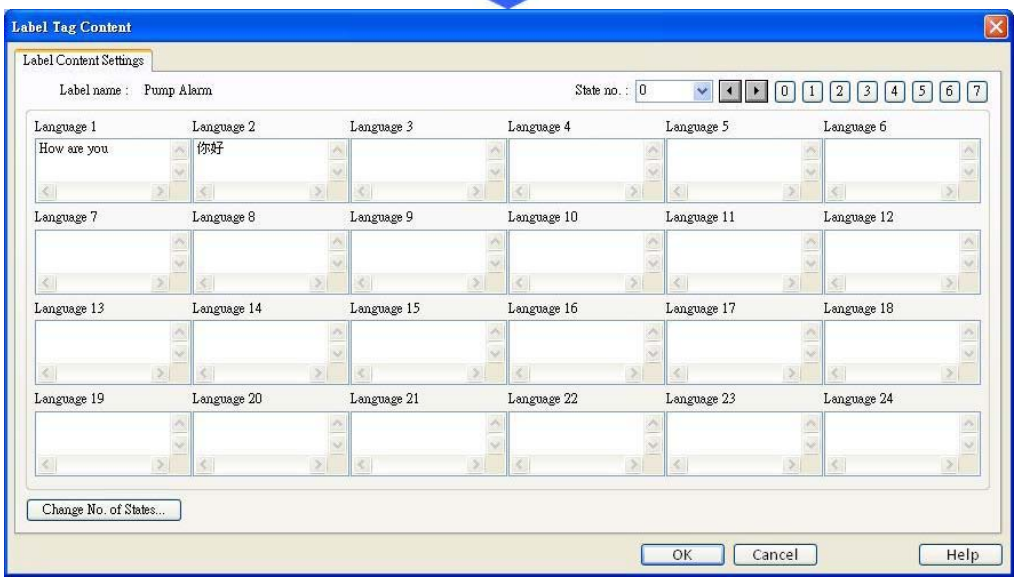
[Label name]
Пользователь может назначить имя *Label*.
[No. of states]
Число состояний для этой *Label.*



2. Щелкните **[OK]** и новая метка “Pump Alarm” с 2-мя состояниями будет добавлена в библиотеку - Label Library,
Выберите её и нажмите **[Settings]**.



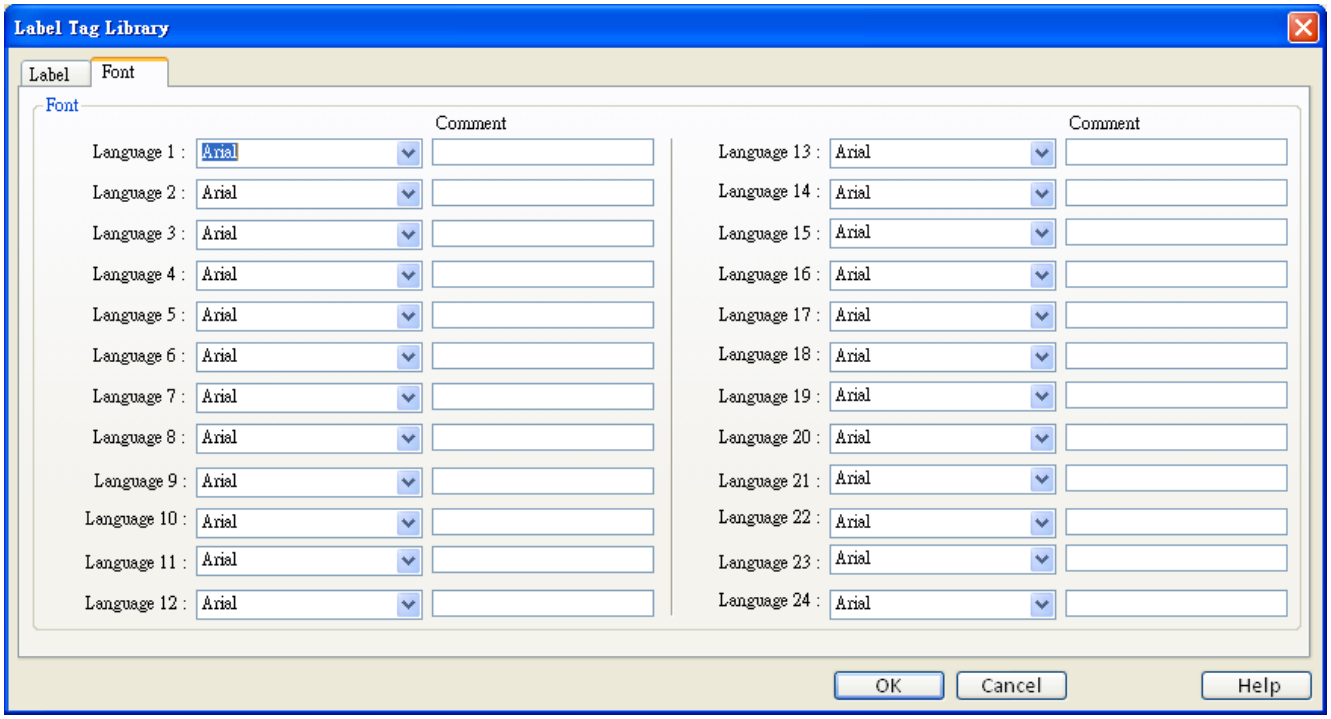
3. Введите соответствующее языковое содержимое.



15.3 Настройка шрифтов Библиотеки меток

Откройте **[Label Tag Library] / [Font] ,** чтобы увидеть языки, используемые в текущей метке.

Для разных языков могут быть выбраны различные шрифты.



[Font]

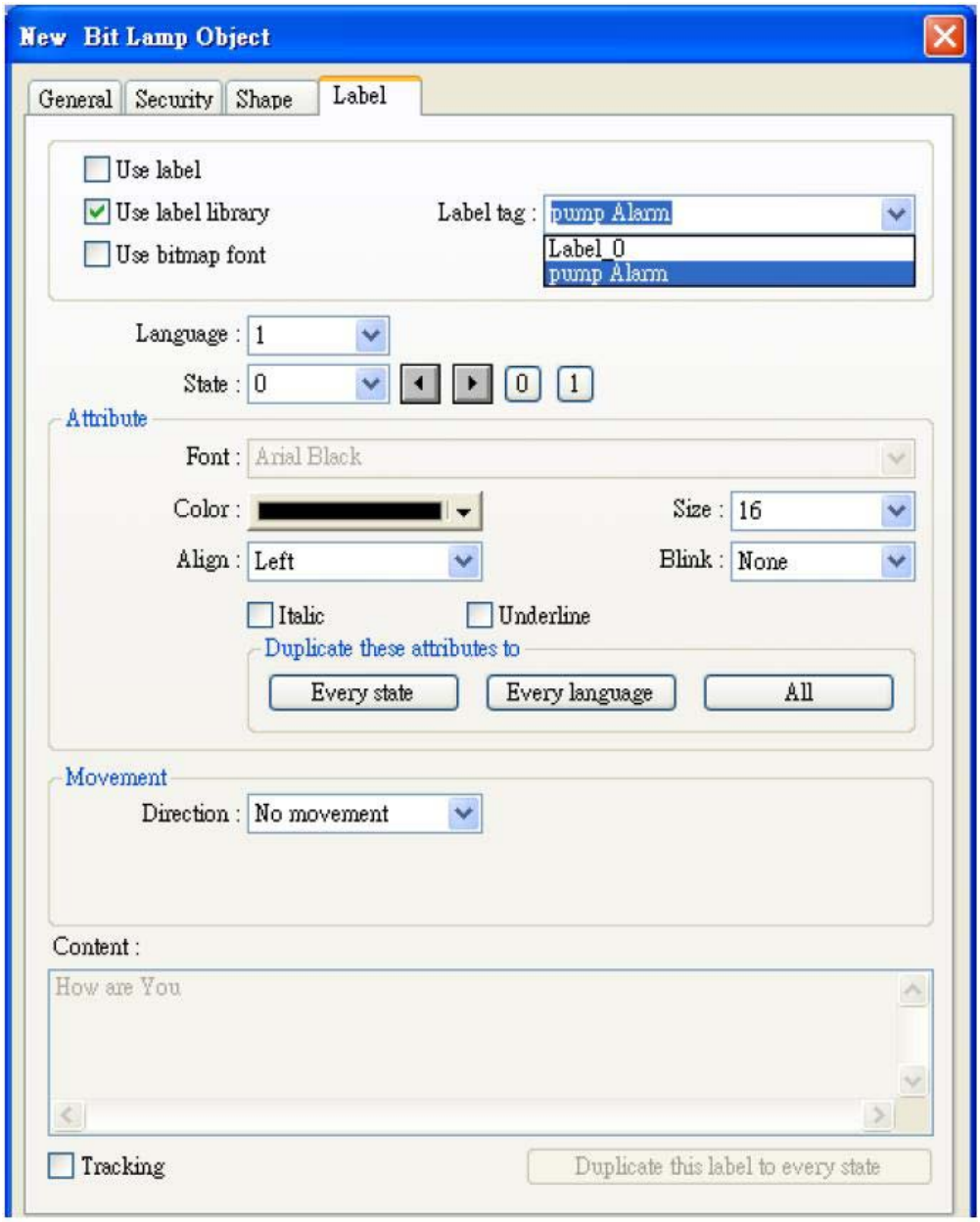
При многоязыковой конфигурации пользователи могут выбирать шрифты для различных языков.

[Comment]

Комментарий (memo) для каждого шрифта.

15.4 Использование Библиотеки меток

Когда в библиотеки уже есть определенные ранее метки, пользователь может видеть их в списке **[Label tag],** отметив прежде флаг **[Use label library]** на закладке **[Label]** окна настройки объекта.



Когда метка выбрана, в области **[Content]** показывается содержимое выбранной метки, и настройки шрифта **[Font]** также хранятся в библиотеке меток.

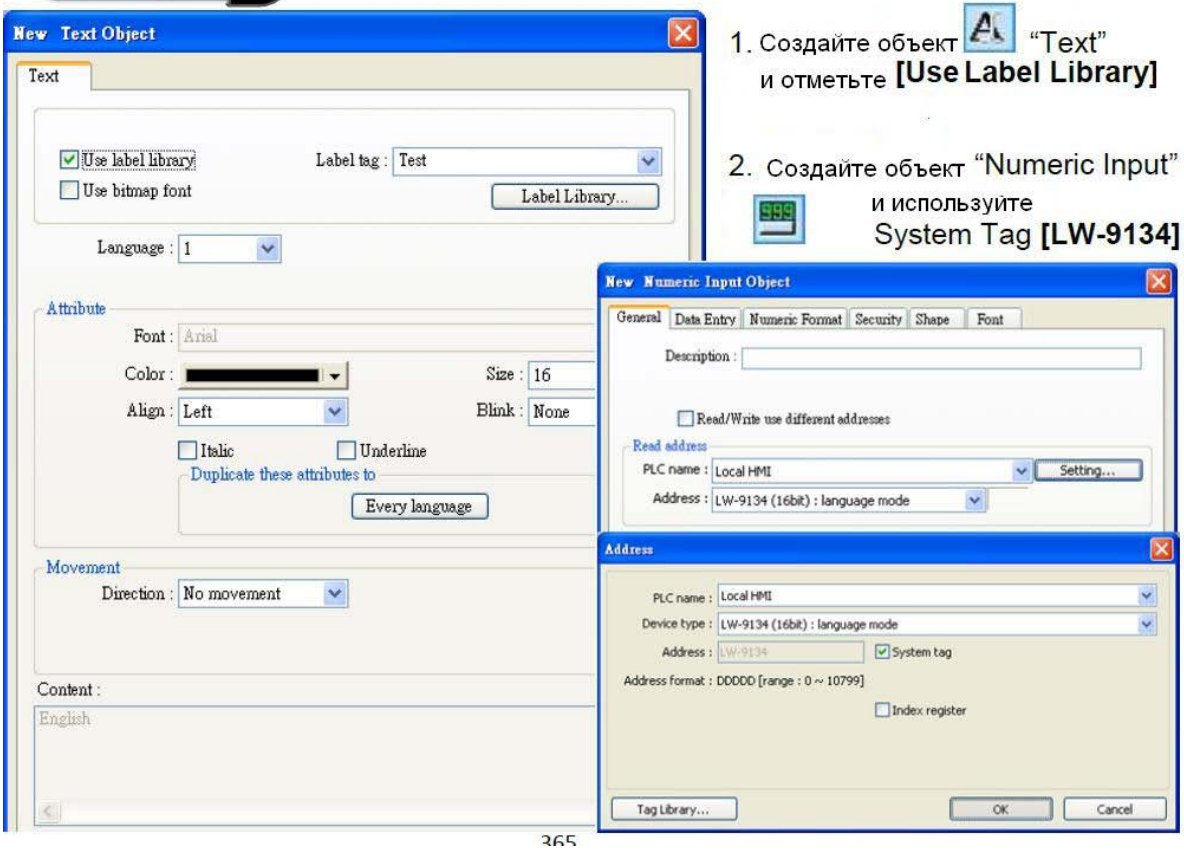
Для языков 2 ~ 24 можно настраивать только размер шрифта - Font **[Size],** все другие настройки (**[Color], [Align], [Blink]** и другие) будут соответствовать настройкам «language 1».

15.5 Использование нескольких языков (системный регистр LW9134)

Когда пользователю необходимо, чтобы текст объекта был многоязыковым, то помимо настроек в библиотеки меток, необходимо использовать также системный регистр [LW9134]. Диапазон его значений: от 0 до 7. Различные значения соответствуют разным языкам. До 24-х языков можно использовать в EasyBuilder Pro и до 8-ми из них могут быть отображены в панели.

Способ использования [LW9134] будет отличаться, если не все языки выбраны при компиляции и загрузке проекта.

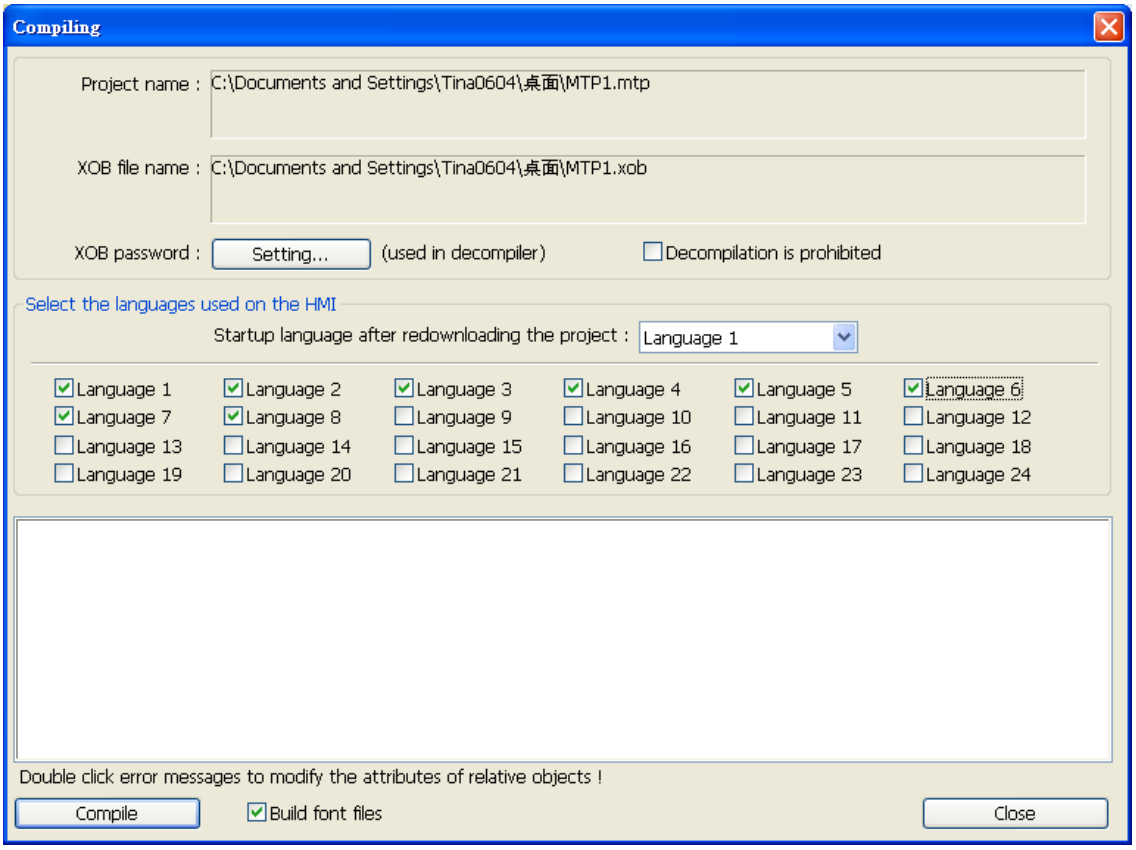
Example 1 **Как использовать многоязычие:**



1. Создайте объект “Text” и отметьте **[Use Label Library]**

2. Создайте объект “Numeric Input” и используйте System Tag **[LW-9134]**

При компиляции отметьте необходимые языки



На рисунках внизу показаны результаты имитации.

При изменении значения по адресу [LW9134] содержание объекта “Text” также изменяется.

Максимально **ВОСЕМ**Ь языков может быть загружено в панель

Соединитесь по internet и загрузите соответствующий demo project:

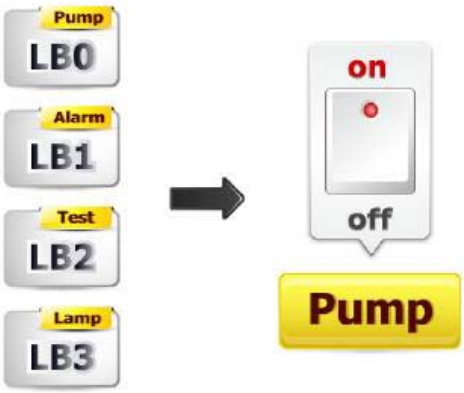
ftp://ftp.weintek.com/MT8000/Project/System_Sample/Multi_Language_with_Option.zip.

Глава 16. Библиотека адресов тэгов

16.1 Создание Библиотеки адресных меток

В начале создания проекта, вообще говоря, рекомендуется определить используемые адреса в Библиотеке адресов тэгов [Address Tag Library]. Это не только позволит избежать повторного ввода одних и тех же адресов, но также более точно использовать адреса в функциях.

Щелкните пиктограмму “Tag” на панели инструментов для вызова окна [Address Tag Library]. См. рисунок ниже.



Отображение тэгов определяемых пользователем

Отображение адресов системных тэгов

Address Tag Library

Customized System

No.	Address tag name	PLC name	Address type	Address	Read/Write	Description
1	Tag_0	Local HMI	Bit	LB-0	Read/Write	test

* Users can import MT500 tag to represent the address.

New... Delete Delete All Settings...

Save Tag File... Load Tag File...

Export CSV... Import CSV... Export EXCEL... Import EXCEL... Exit

Создание новых адресов тэгов показано на следующей странице

Сохранение всех существующих адресов тэгов в файле *.tgl

Загрузка файла *.tgl в Address Tag Library.

Экспорт текущей Address Tag Library в указанное место в формате *.csv

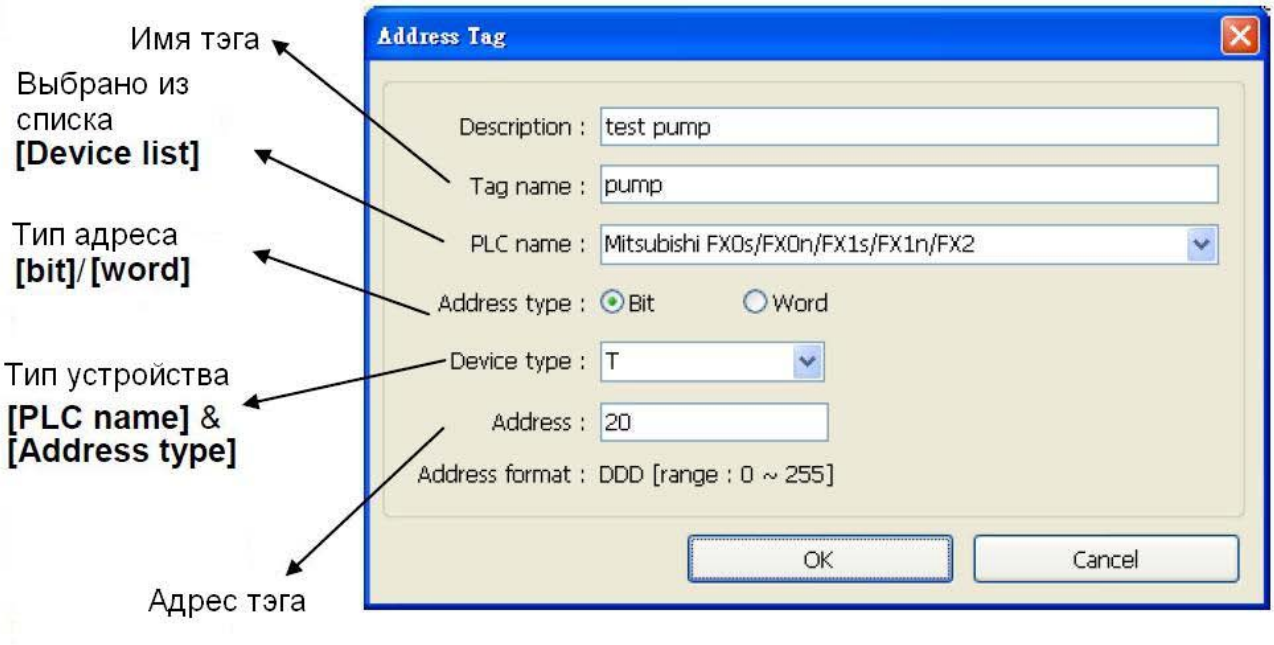
Импорт файла *.csv Address Tag Library в текущий проект

Экспорт текущей Address Tag Library в указанное место в формате *.xls

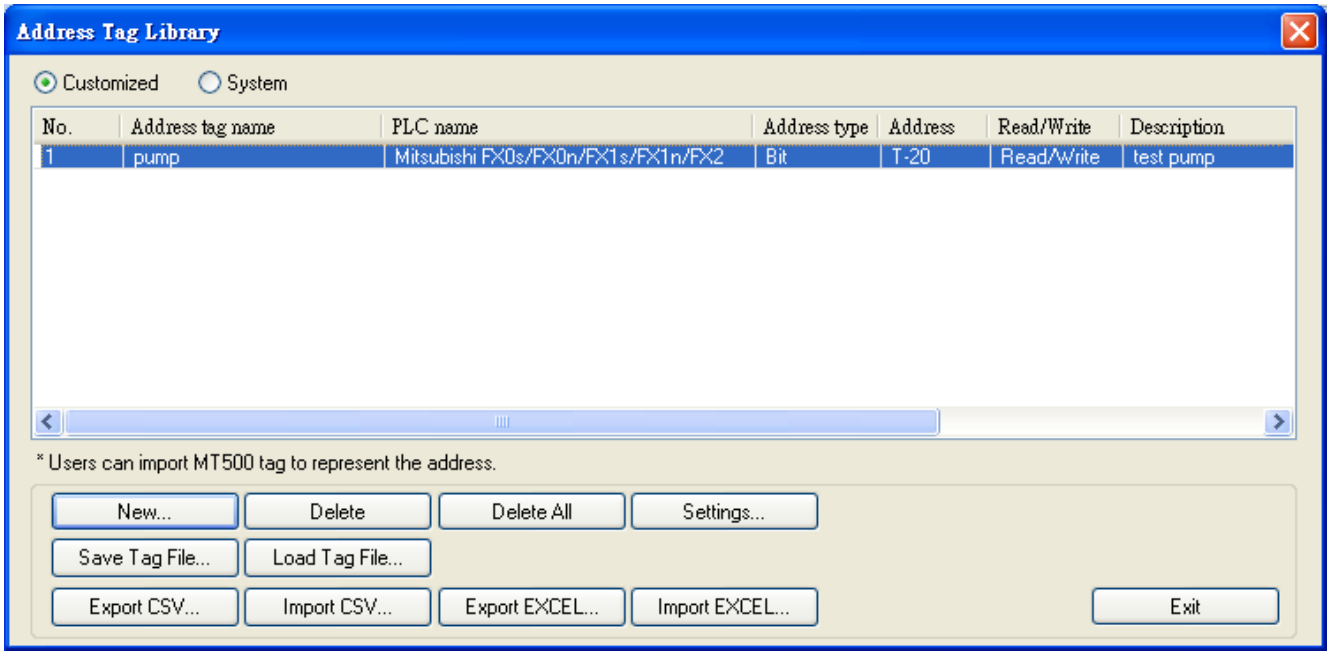
Импорт файла *.xls Address Tag Library в текущий проект

Редактирование выбранного адреса тэга

Нажмите **[New]**

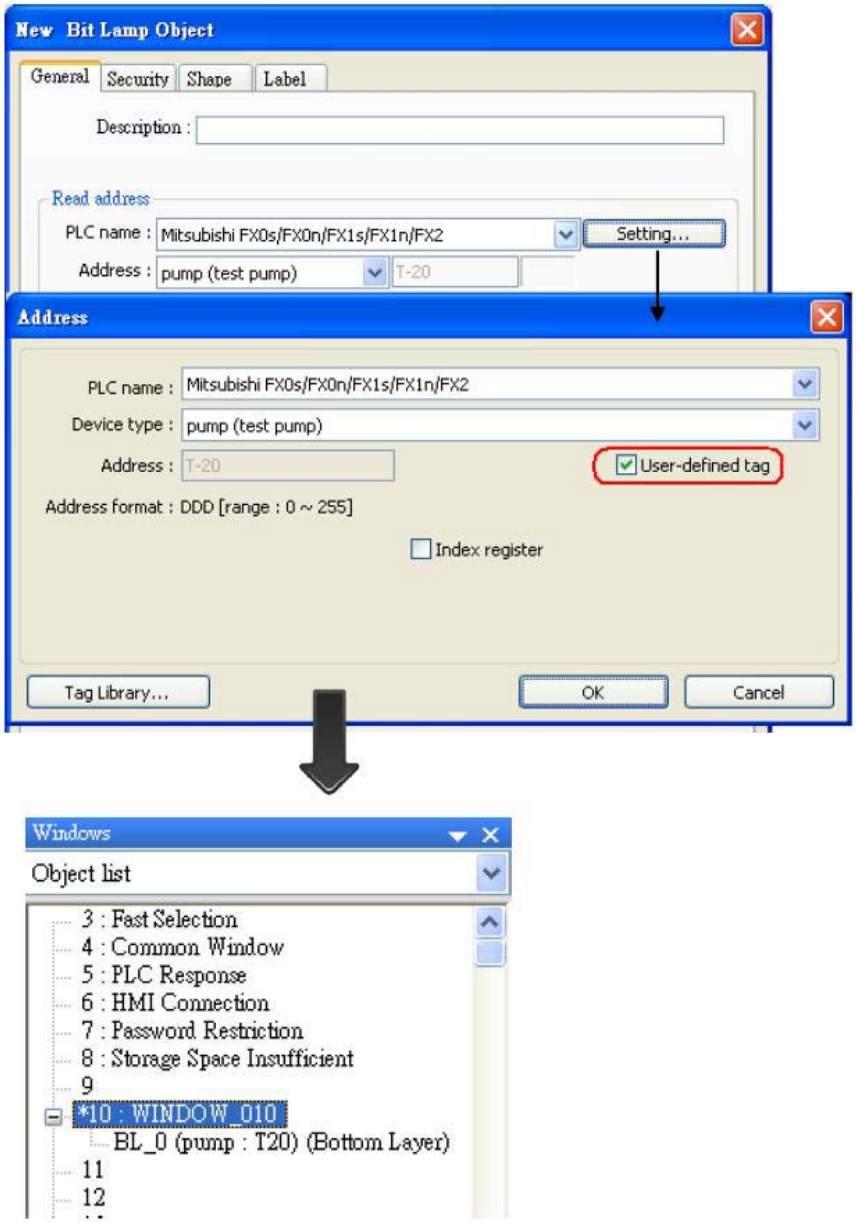


Нажмите ОК после завершения настроек, новый тэг будет помещен в пользовательскую библиотеку **[Customized]**. См. рисунок ниже.



.

16.2 Использование Библиотеки адресов тэгов



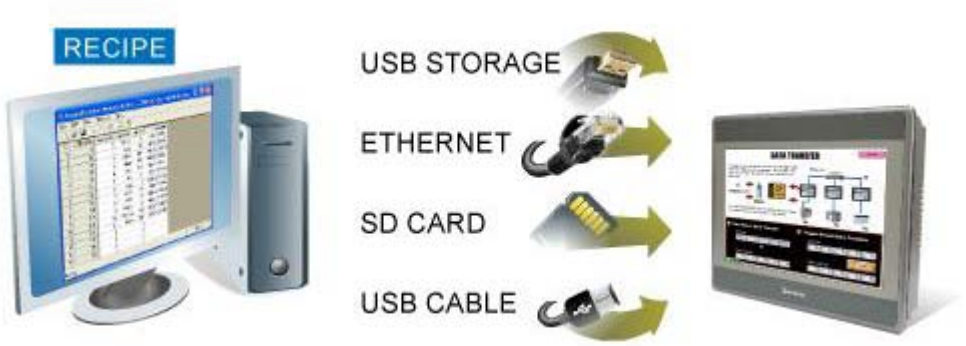
1. Определите библиотеку – Address Tag Library
2. Создайте объект, выберите **[General] / [PLC name]**
3. Нажмите **[Setting]**
4. Отметьте **[User-defined tag]**
   – Тэг определяемый пользователем**.**
5. В **[Device type]** выберите этот тэг.
6. После завершения, в окне появится имя адресуемого тэга использованного в объекте.

.

Глава 17. Передача данных рецептов

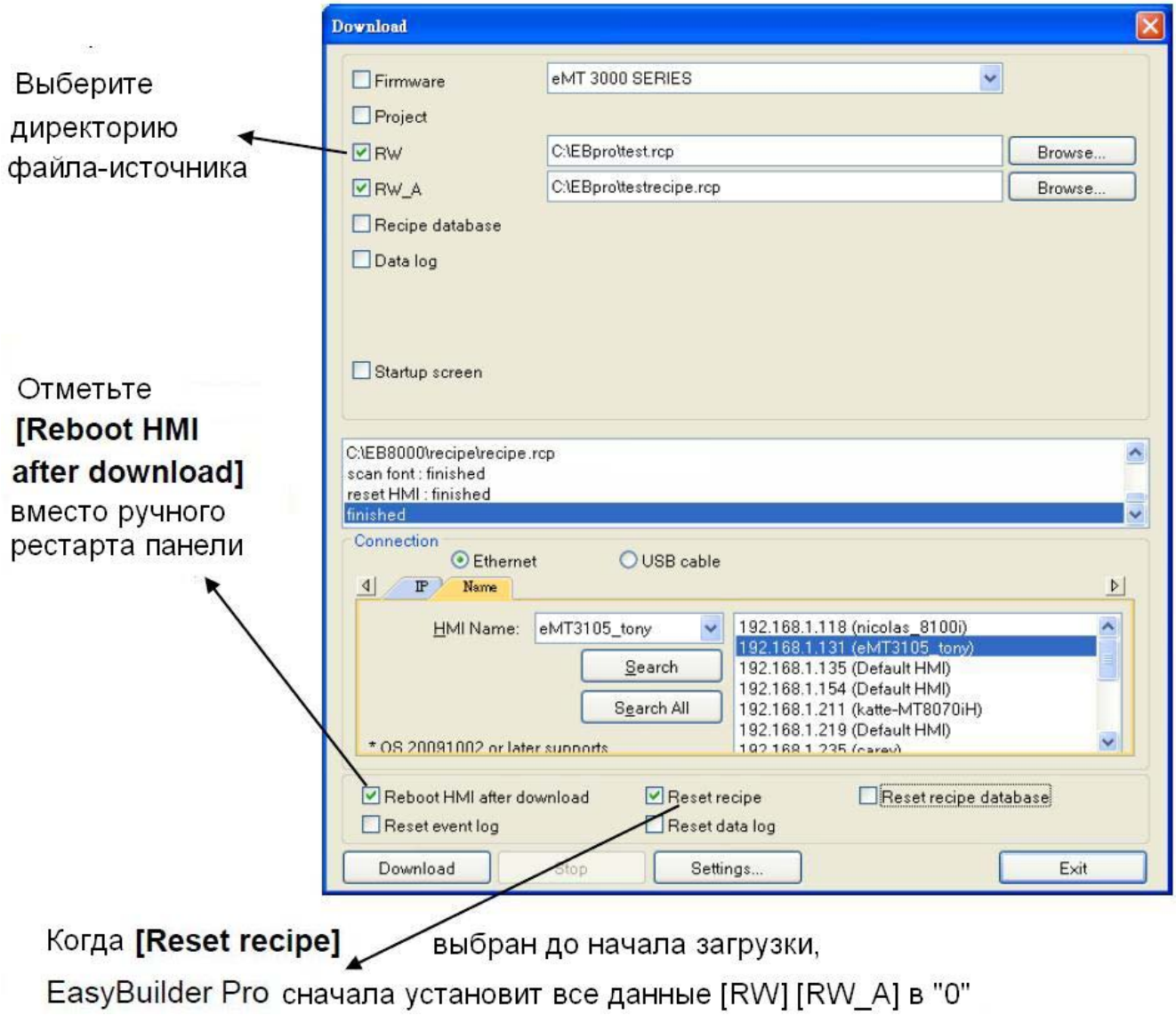
Данные рецептов (Recipe Data) хранятся в адресах типа RW и RW_A. Способ чтения и записи *Recipe Data* такой же, как и для обычных адресов типа [word]. Отличие состоит в том, что *Recipe Data* хранятся во флэш-памяти. При запуске системы последние записи RW и RW_A сохраняются.

Размер наборов данных и в RW составляет 512К-слов, и в RW_A составляет 64К-слов. Пользователь может обновлять *Recipe Data*, используя SD-карту, USB-кабель, по сети Ethernet и использовать для обновления данных в ПЛК.
Возможно загрузить наборы данных в назначенный адрес ПК; далее, можно сохранить данные в памяти рецептовПЛК. Ниже пояснены все способы работы с наборами данных.



17.1 Обновление наборов данных по сети Ethernet или через USB-кабель

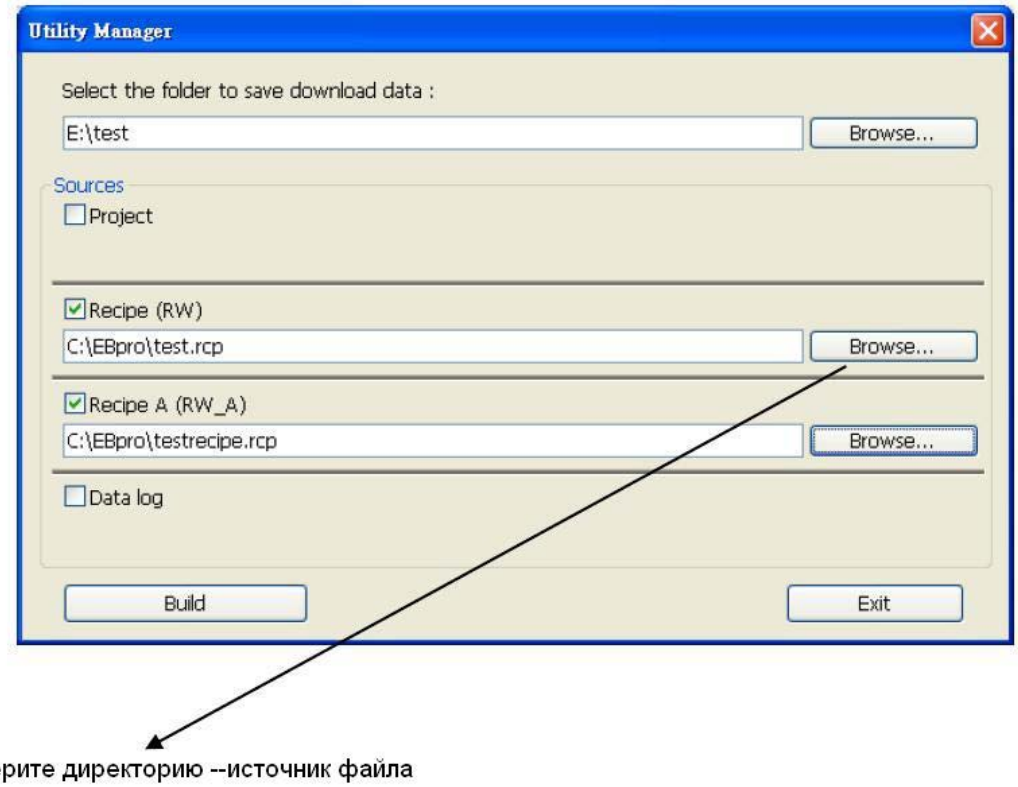
1. Щелкните **[Download]** в Менеджере утилит - **Utility Manager** .
2. Выберите **[RW]** и **[RW_A]** и укажите папку-источник файлов.
3. После завершения загрузки, перезапустите панель оператора и содержание RW и RW_A будет обновлено.



17.2 Обновление данных с использованием CF/SD-карт или USB-накопителей

1. Откройте *Utility Manager* и нажмите **[Build Download Data for CF/SD Card or USB Disk]**.
2. Вставьте SD карту или накопитель USB в компьютер.
3. Нажмите **[Browse] ,** чтобы назначить путь к файлу.
4. Нажмите **[Build]**, EasyBuilder Pro обновит источники в SD карте или в USB-диске.

.

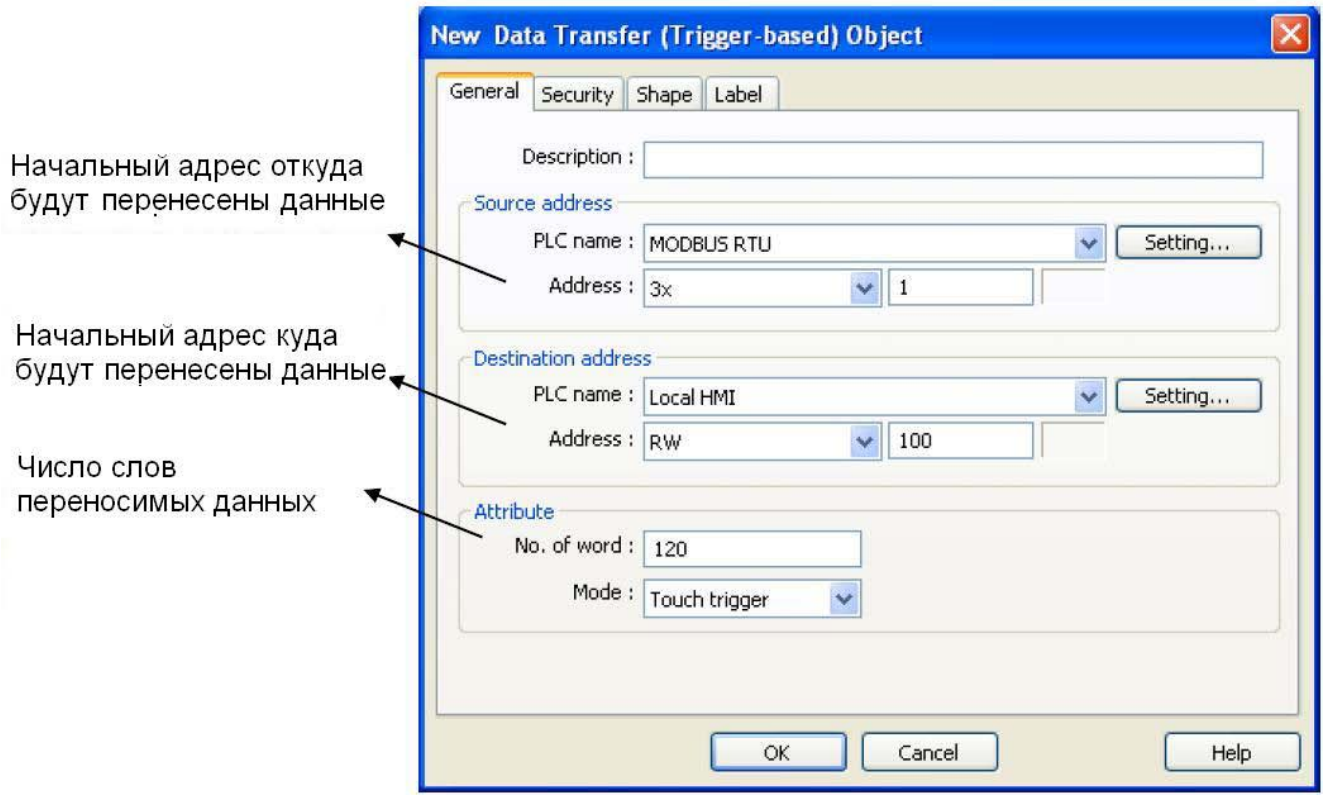


Вытерите директорию --источник файла

Когда данные успешно созданы, две папки могут быть найдены: *history* и emt3000. Файл *emt3000* предназначен для хранения файлов проекта, а файл *history* предназначен для хранения записей Recipe Data и Data Sampling / Event Log

17.3 Передача наборов данных

Для передачи наборов данных в конкретный адрес нужно использовать объект типа **[Data Transfer (Trigger-based)] -** или сохранять данные из указанных адресов в области [RW] и [RW_A].



17.4 Автоматическое сохранение наборов данных

Для увеличения срока службы флэш-памяти панели оператора EasyBuilder Pro **каждую минуту** автоматически сохраняет данные в панели.

Во избежание их потери при отключении панели во время интервала между операциями сохранения, в EasyBuilder Pro есть возможность ручного сохранения наборов данных с помощью системного бита [LB9029: save all recipe data to machine (set ON)].

EasyBuilder Pro сохранит набор данных при установке бита [LB9029] в ON.

Когда пользователь установит в ON бит [LB9028: reset all recipe data (set ON)], EasyBuilder Pro сбросит все наборы данных в 0.

Глава 18. Макросы

Макросы обеспечивают дополнительные функции, которые могут понадобиться для вашего приложения.
Макросы это автоматически выполняемые последовательности команд, исполняемых во время работы. Макросы позволяют решать такие задачи, как сложные вычислительные операции, работа со строками и взаимодействия пользователей с вашими проектами.
В этой главе приведены правила написания, использование и методы программирования макросов.

18.1 Команды Редактора макросов

1. Редактор макросов (Macro editor) обеспечивает следующие новые функции:

a. displaying line number – отображение номера строки

b. Undo / Redo – Отменить/возобновить

c. Cut / Copy / Paste – Вырезать/Копировать/Вставить

d. Select All – Выделить все

e. Toggle Bookmark / Previous Bookmark / Next Bookmark / Clear All Bookmarks – Переключить закладки / Предыдущая закладка / Следующая закладка / Очистить все закладки

f. Toggle All Outlining – Переключить все выделенное

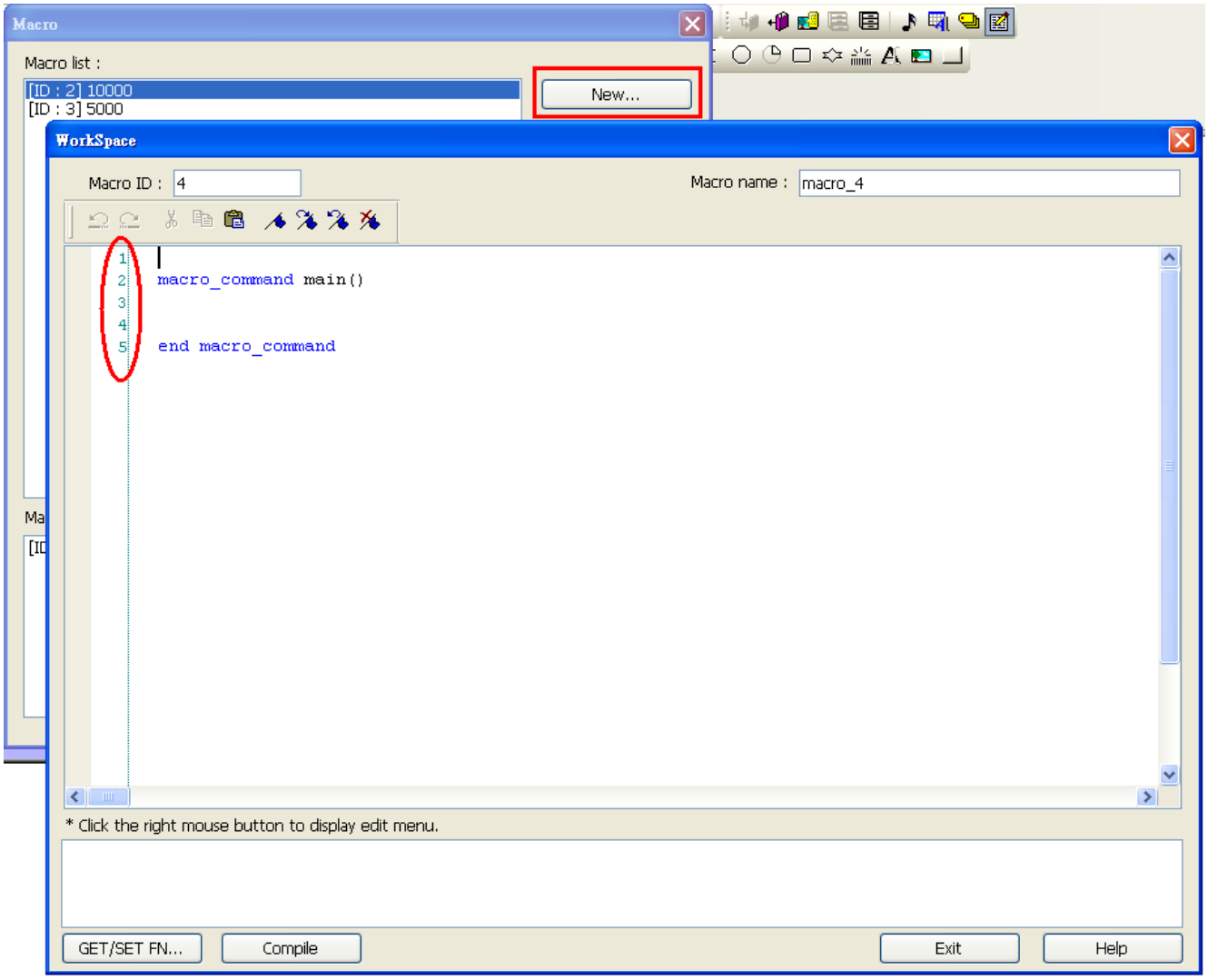
g. Security -> Use execution condition – Использовать условия выполнения

h. Periodical execution – Периодическое выполнение

i. Execute one time when HMI starts – Выполнить один раз при пуске панели оператора

Далее будет показано, как использовать эти инструкции.

2. Открыв *Macro editor* , Вы увидите номера линий (строк) на левой стороне окна редактирования.

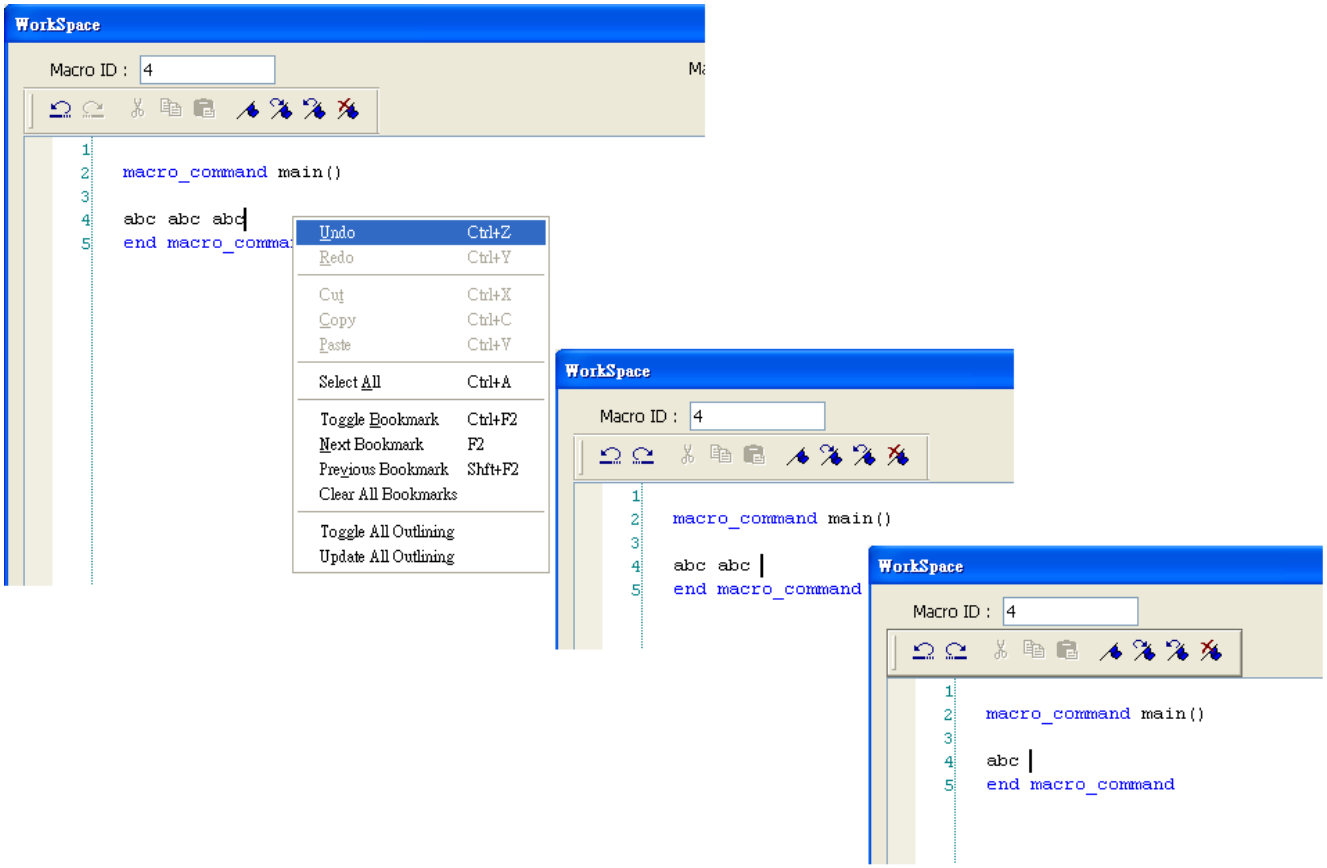


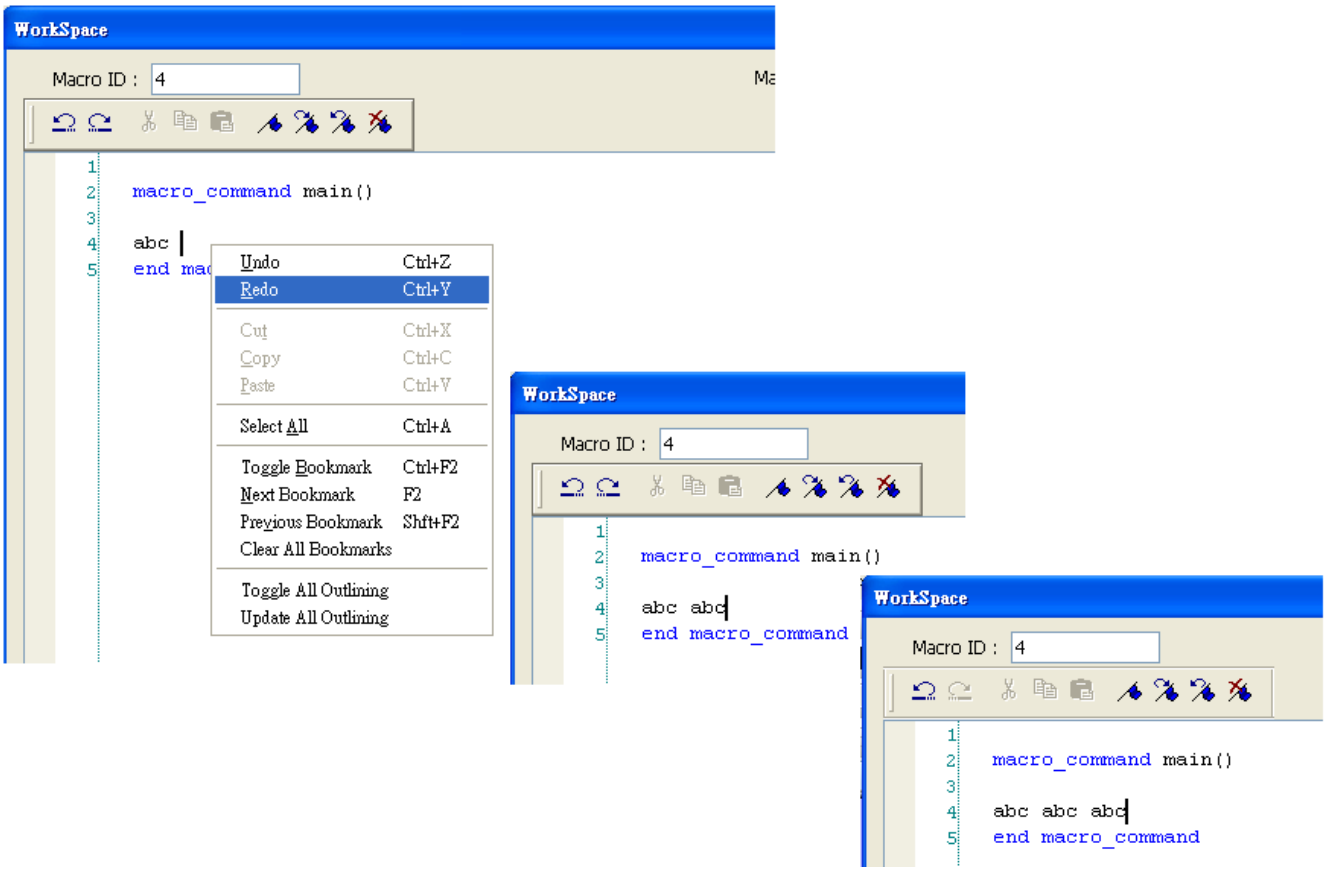
3. Правым щелчком по полю редактированию можно открыть всплывающее меню, показанное ниже:

Не используемые (disabled) пункты окрашены серым цветом, что указывает на то, что их нельзя использовать в этой функции в текущем состоянии редактора.
Например: Вы можете отметить выбранную область для копирования, иначе она будет недоступной (disabled).
«Горячие клавиши» (Accelerators) поддерживаются, как описано в меню.

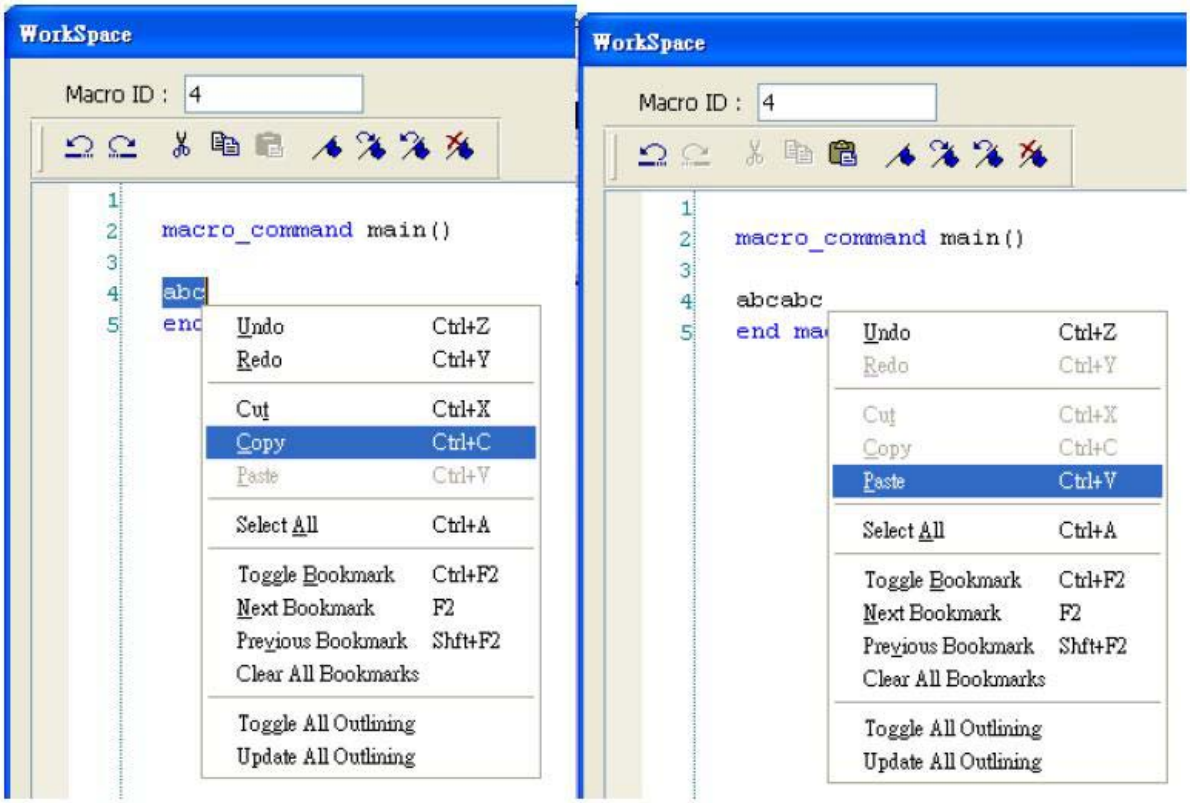
4. Над полем редактирования размещена панель инструментов с кнопками “Undo”, “Redo”, “Cut”, “Copy”, “Paste”, “Toggle Bookmark”, “Next Bookmark”, “Previous Bookmark” и “Clear All Bookmarks” для постоянного использования.

5. Действие, сделанное редактором, можно отменить (undo). Функция *Redo* становится активной после выполнения функции *Undo*. Выполнить функции undo/redo можно, сделав правый щелчок на выбранном элементе или использовать горячие клавиши (Undo: Ctrl+Z, Redo: Ctrl+Y).

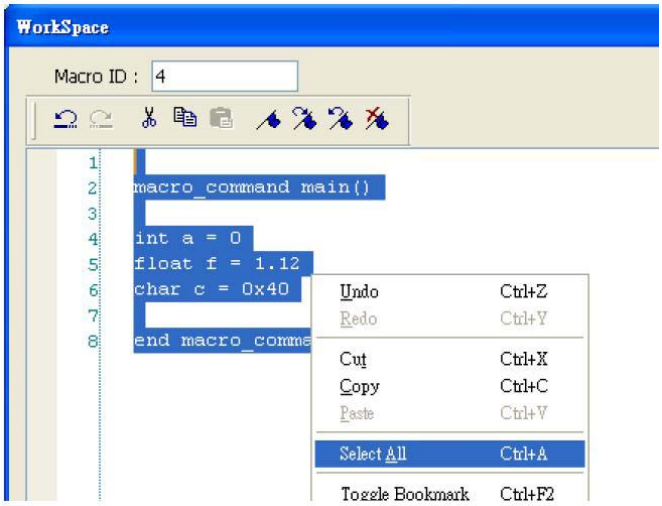




6. Выбор слова для редактирования активирует функции «cut» и «copy». После выполнения функции «cut» или «copy», доступна функция «paste».

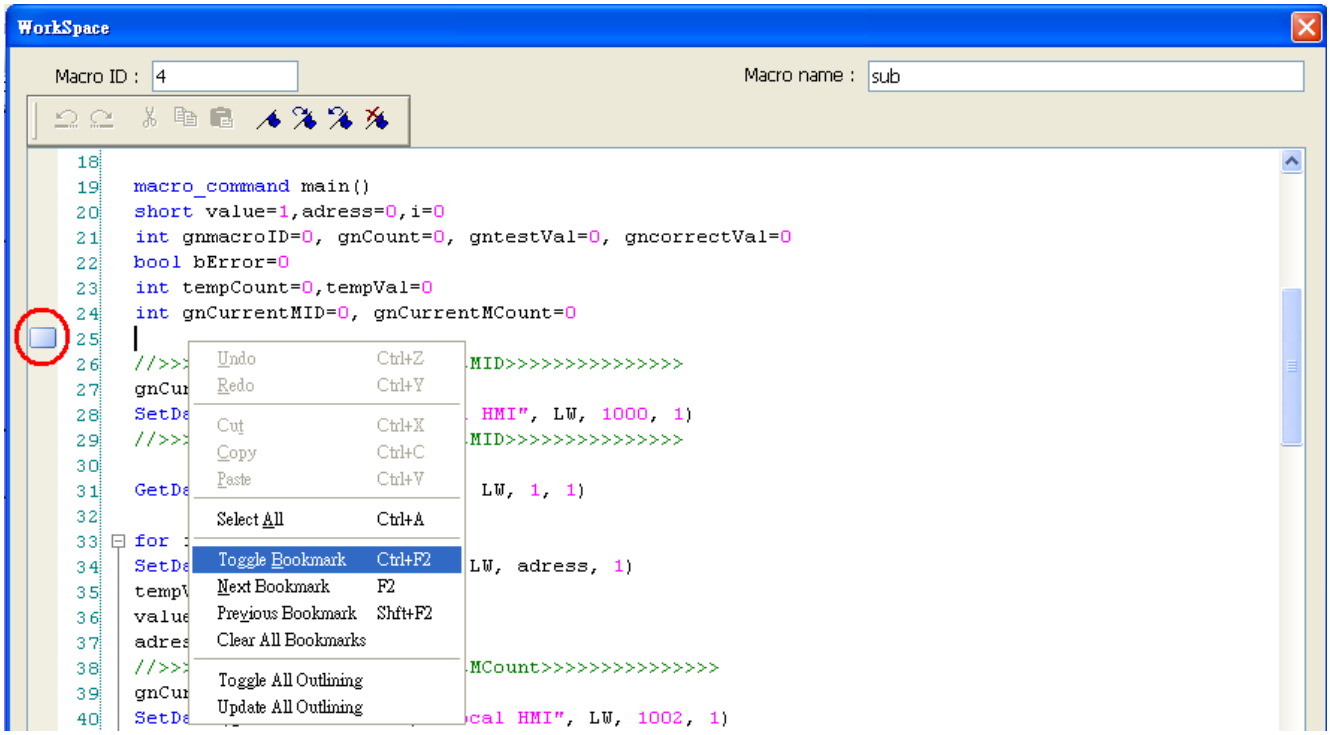


7. Используйте “Select All”, чтобы включить все содержимое редактируемой области.



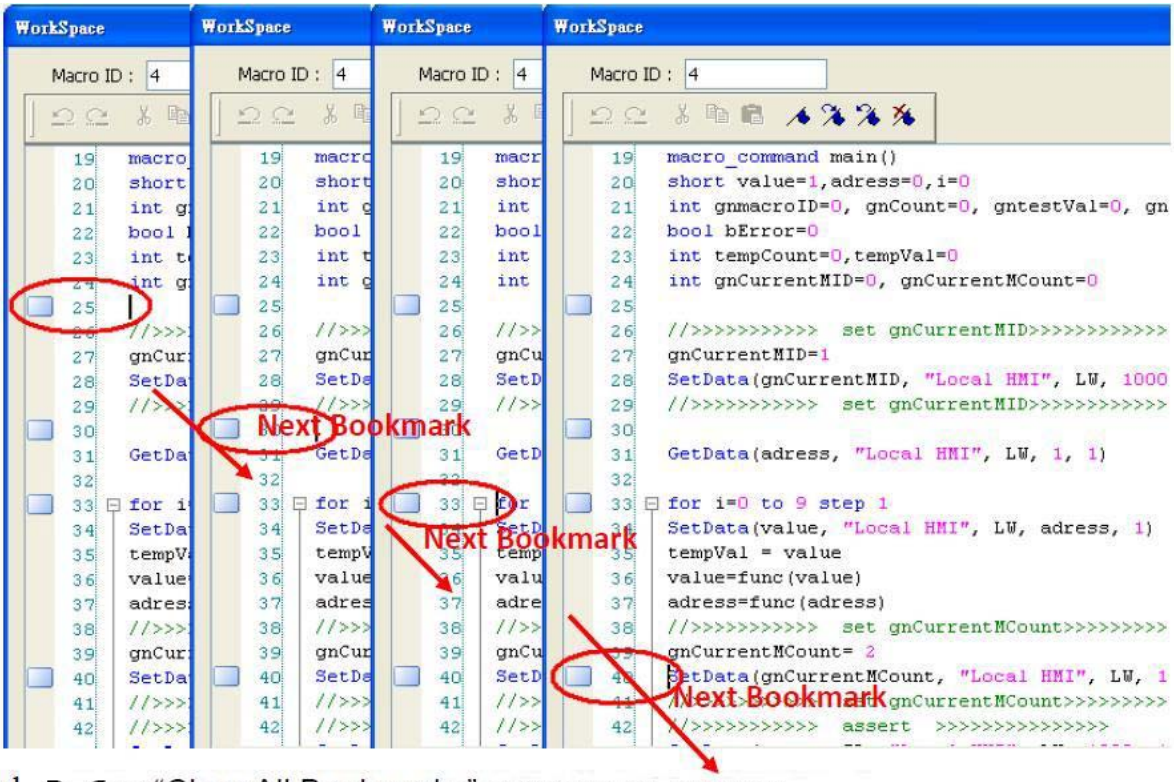
8. Если макрокод становится длинным, для облегчения чтения, можно использовать «закладки» - *bookmarks*. Далее показано, как с ними работать.

a. Переместите Ваш курсор к области редактирования, где нужно вставить *bookmark*. Правым щелчком выберите “Toggle Bookmark”. Появится голубой прямоугольник, который представляет Закладку - bookmark на левой стороне поля редактирования.



b. Если на месте курсора закладка уже существует, выберите “Toggle Bookmark”. Чтобы закрыть её, или откройте её.

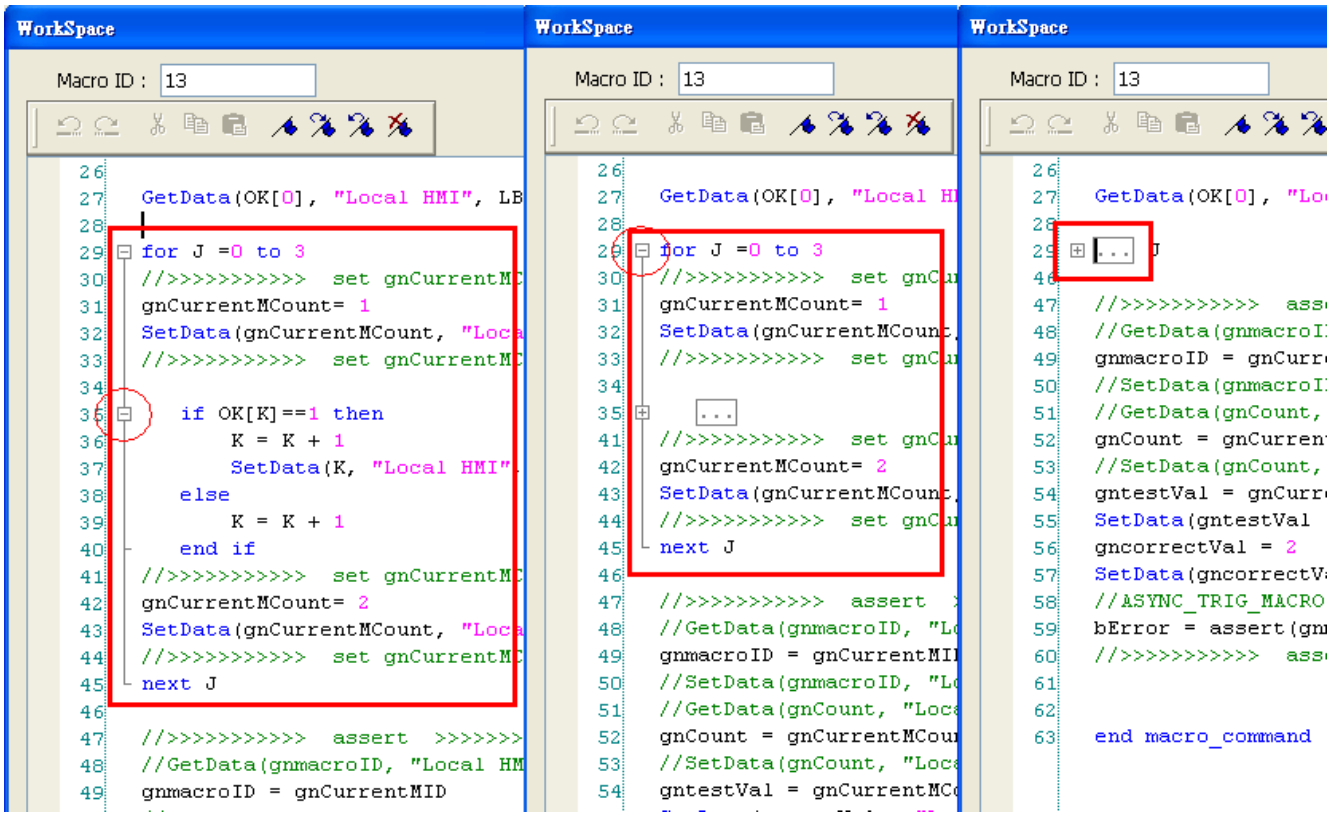
c. Правый щелчок и выбор “Next Bookmark” – курсор переместится к следующей закладке. Выбор” Previous Bookmark” переместит курсор к предыдущей закладке.



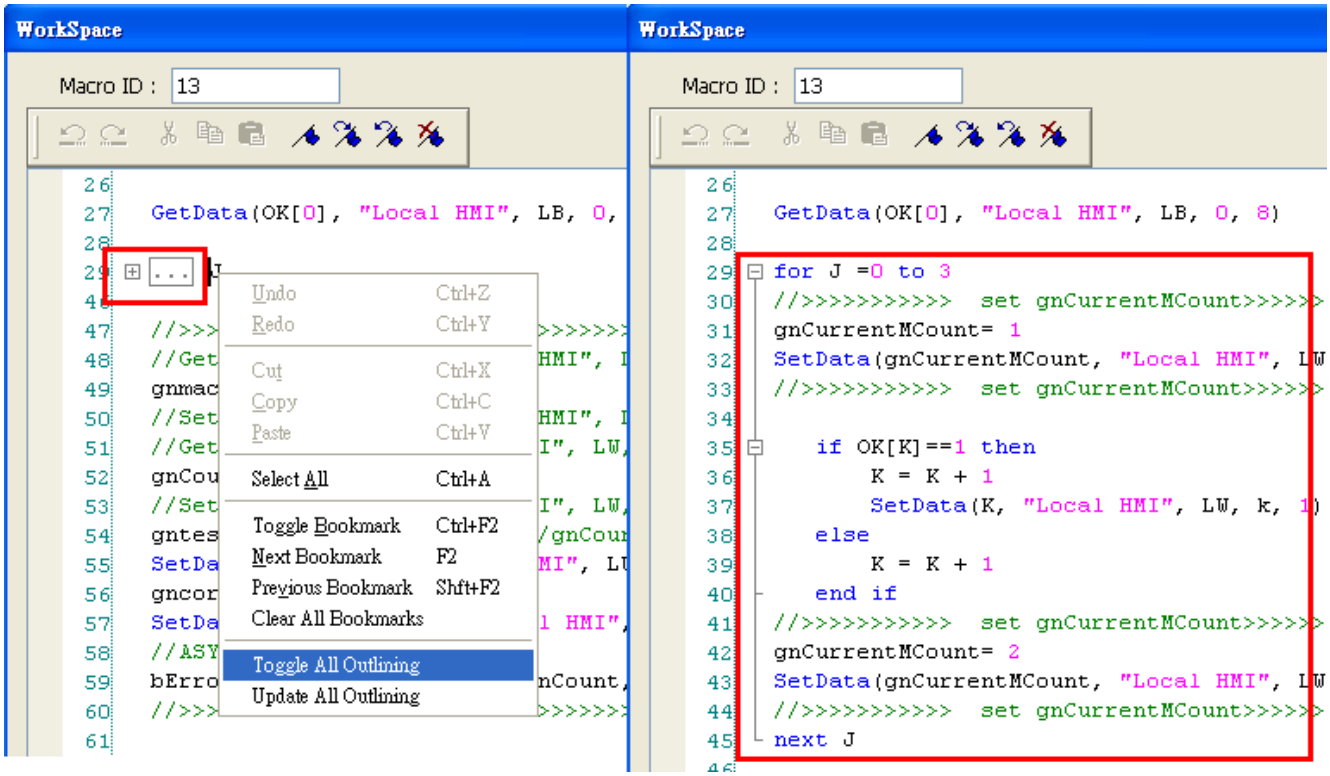
d. Выбор “Clear All Bookmarks” закроет все отметки .

9. У редактора есть функция Оконтуривания – «outlining». Эта функция предназначена для скрывания макрокода принадлежащего тому же самому блоку, и отображению его при помощи пиктограмм . На левой стороне стороне поля редактирования должно быть дерево файлов.

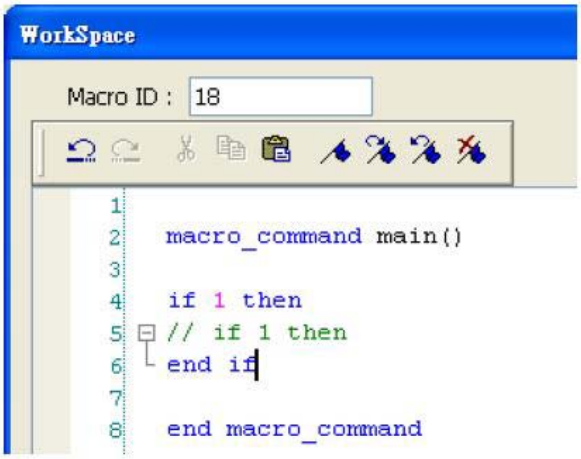
Пользователь может щелкнуть . чтобы спрятать блок или , чтобы открыть блок, как показано на следующей странице:



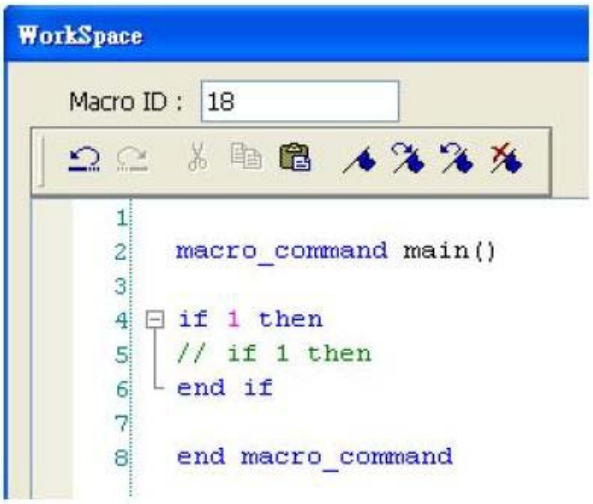
10. Правый щелчок и выбор “Toggle All Outlining”, чтобы открыть весь макрокод блока.



11. Иногда «outlining» может быть не корректным из-за ошибочного трактования ключевых слов «keywords». Например:



Чтобы разрешить эту проблему, правым щелчком выберте “Update All Outlining” чтобы вернуть корректное «outlining».



12. Операторы (statements) окруженные следующими ключевыми словами, называются “блоком” макрокода:

a. Function block:

   sub – end sub

b. Reiterative statements:

   i. for – next

   ii. while – wend

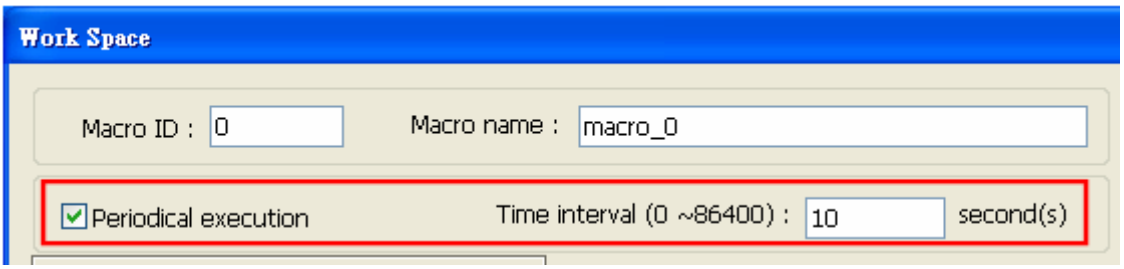
c. Logical statements:

   i. if – end if

d. Selective statements:

   select case – end select

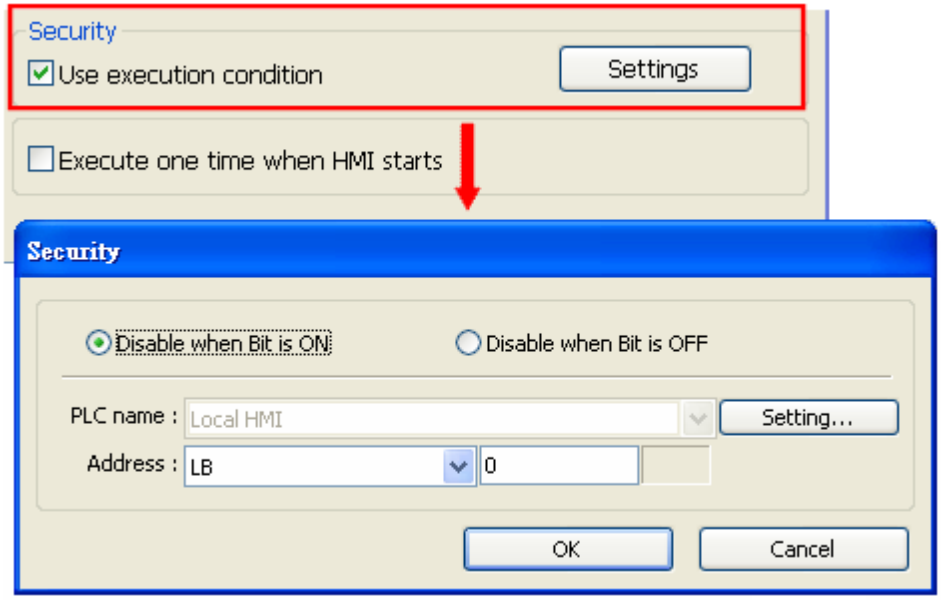
13. Если отметить “Periodical execution”, макрос будет запускаться периодически.



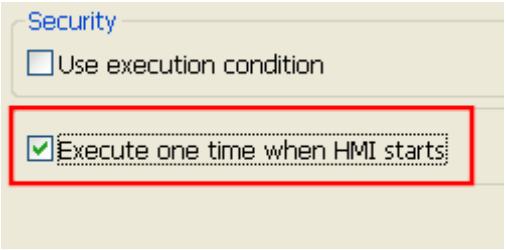
14. Выберите *Security -> Use execution condition -> Settings,* чтобы войти в настройки **Security Settings:**

a. *Disable when Bit is ON*: Когда бит = ON, этот Макрос дезактивирован.

b. *Disable when Bit is OFF*: Когда бит = OFF, этот Макрос дезактивирован.



15. Если выберите “*Execute one time when HMI starts*”, этот Макрос будет выполняться только при запуске панели (HMI).



18.2 Состав Макроса

Макрос состоит из операторов (statements) . Операторы содержат константы, переменные и операции. Они располагаются в определенном порядке для получения требуемого выходного результата.

Макрос составляется в виде:

```
Объявления глобальных переменных      ----------------------------------------Опционально

Подпрограмма. Объявления Функциональных блоков    --------------------Опционально
                Объявления локальных переменных
Конец подпрограммы

macro_command main()                  ----------------------------------------Обязательно
                Объявления локальных переменных
                Операторы
end macro_command()                   ----------------------------------------Обязательно
```

Макрос должен имет одну *main function*, являющуюся начальной точкой исполняемого макроса. Её формат:

macro_command Function_Name()

end macro_command

Локальные переменные используются в теле главной (main) функции макроса или в определенном функциональном блоке. Их значение остается действительным только внутри определенного блока.

Глобальные переменные объявляются перед всеми функциональными блоками и действительны для всех функций макроса. Когда локальные и глобальные переменные имеют одинаковое имя, учитываются только локальные переменные.

Ниже показан образец макроса, включающего объявления переменных и вызов функции.

```
macro_command main()
        short pressure = 10                                  // local variable declaration
        SetData(pressure, "Allen-Bradley DF1", N7, 0, 1)     // function calling
end macro_command
```

18.3 Синтаксис

18.3.1 Константы и переменные

18.3.1.1 Константы

Константы — это фиксированные значения и могут быть использованы сразу в операторе:

Тип константы	Примечание	Пример
Десятичная целая		345, -234, 0, 23456
Шестнадцатеричная	Должна начинаться с 0x	0x3b, 0xffff, 0x237
ASCII (строковая)	Строка должна начинаться заключена в одиночные кавычки	‘a’, ‘data’, ‘name’
Булева		true, false

Пример оператора, использующего константу:

```
macro_command main()
  short A, B     //  A и B переменные
  A = 1234
  B = 0x12      //  1234 и 0x12 константы
end macro_command
```

18.2.1.2 Переменные

Переменные имеют имя, как указатель на их значение. Значение может изменяться, при преобразовании переменной операторами.

Правила назначения имен переменных

1. Имя должно начинаться с буквенного символа.
2. Не допускаются имена переменных длиннее 32 символов.
3. Служебные (Reserved) слова не могут использоваться в качестве имен переменных.

Существует 5 различных типов переменных.

Тип переменной	Описание	Диапазон значений
bool	1 бит (дискретная)	0, 1
Char	8 бит (байт)	+127 ~ -128
short	16 бит (слово)	+32767 ~ -32768
Int	32 бита (двойное слово)	+2147483647 ~ -2147483648
float	32 бита (двойное слово)	
unsigned char	8 бит (байт)	От 0 до 255
unsigned short	16 бит (слово)	От 0 до 65535
unsigned int	32 бита (двойное слово)	От 0 до 4,294,967,295

Объявление переменных

Переменные должны быть объявлены перед использованием. Все объявления переменных должны быть сделаны перед операторами макроса. Для объявления переменной нужно указать тип и имя переменной.

Пример:

```
int       a
short     b, switch
float     pressure
```

Объявление массивов (Arrays)

Макросы поддерживают использование одномерных массивов (zero-based index) . Для объявления массива переменных определите тип, имя переменной и в квадратных скобках “[]” укажите число переменных, входящих в массив. Массивы могут иметь длину от 1 до 4096 переменных.

Пример:

```
Int       a[10]
Short     b[20], switch[30]
float     pressure[15]
```

Минимальная позиция массива – 0, максимальная: (размер массива -1).

Пример:

```
char data [100]          // array size is 100
```

где: минимальный номер элемента 0 и максимальный: 99 (100 – 1)

Инициализация переменных и массивов

Есть два способа инициализации переменных:

1. Перед оператором с помощью присваивания (=)

Пример:

```
int     a
float   b[3]
a = 10
b[0] = 1
```

2. При объявлении (declaration)

```
char a = ‘5’, b = 9
```

Объявление массивов – отдельный случай. Массив целиком может быть инициализирован при объявлении перечислением элементов через запятую внутри фигурных скобок “{}”.

Пример:

```
float data[4] = {11, 22, 33, 44}    // now data[0] is 11, data[1] is 22….
```

18.3.2 Операторы - Operators

Операторы используются для указания того, какие данные и каким образом должны быть преобразованы. В каждой строке операнду слева присваиваются условия справа.

Оператор	Описание	Пример
=	Оператор присвоения	pressure = 10

Арифметические операторы	Описание	Пример
+	Addition - Сумма	A = B + C
-	Subtraction - Разность	A = B – C
*	Multiplication - Умножение	A = B * C
/	Division - Деление	A = B / C
%	Modulo division - Деление по модулю 2 (возвращает остаток от деления)	A = B % 5

Оперраторы сравнения	Описание	Пример
<	Меньше чем	if A < 10 then B = 5
<=	Меньше или равно	if A <= 10 then B = 5
>	Больше чем	if A > 10 then B = 5
>=	Больше или равно	if A >= 10 then B = 5
==	Равно	if A == 10 then B = 5
<>	Не равно	if A <> 10 then B = 5

Логические операции	Описание	Пример
And	Логическое «И»	if A < 10 and B > 5 then C = 10
Or	Логическое «ИЛИ»	if A >= 10 or B > 5 then C = 10
Xor	Исключающее «ИЛИ»	if A xor 256 then B = 5
Not	Логическое отрицание	if not A then B = 5

Сдвиговые и побитовые (Shift и Bitwise) операторы используются для работы с битами переменных типа **char, short,** и **int.**

Приоритет этих операторов внутри записи устанавливается слева направо.

Сдвиговые операторы	Описание	Пример
<<	Сдвиг влево на указанное число бит	A = B << 8
>>	Сдвиг вправо на указанное число бит	A = B >> 8

Побитовые операции	Описание	Пример
&	Побитовое «И»	A = B & 0xf
|	Побитовое«ИЛИ»	A = B | C
^	Побитовое «исключающее ИЛИ»	A = B ^ C
~	Инвертирование - One’s complement	A = ~B

Приоритеты операторов

Общая приоритетность всех операций с высшей по низшую приведена ниже:

Операции в скобках выполняются в первую очередь.

Арифметические операции

Сдвиговые и битовые операции

Операции сравнения

Логические опреации

Присвоение

Служебные слова

Данные ниже слова зарезервированы для использования в Маросе. Их нельзя использовать в качестве имен переменных, массивов или имен функций.

+, -, *, /, %, >=, >, <=, <, <>, ==, and, or, xor, not, <<, >>,=, &, |, ^, ~
exit, macro_command, for, to, down, step, next, return, bool, short, int, char, float, void, if, then, else, break, continue, set, sub, end, while, wend, true, false
SQRT, CUBERT, LOG, LOG10, SIN, COS, TAN, COT, SEC, CSC, ASIN, ACOS, ATAN, BIN2BCD, BCD2BIN, DEC2ASCII, FLOAT2ASCII, HEX2ASCII, ASCII2DEC, ASCII2FLOAT, ASCII2HEX, FILL, RAND, DELAY, SWAPB, SWAPW, LOBYTE, HIBYTE, LOWORD, HIWORD, GETBIT, SETBITON, SETBITOFF, INVBIT, ADDSUM, XORSUM, CRC, INPORT, OUTPORT, POW, GetError, GetData, GetDataEx, SetData, SetDataEx, SetRTS, GetCTS, Beep, SYNC_TRIG_MACRO, ASYNC_TRIG_MACRO, TRACE, FindDataSamplingDate, FindDataSamplingIndex, FindEventLogDate, FindEventLogIndex
StringGet, StringGetEx, StringSet, StringSetEx, StringCopy, StringMid, StringDecAsc2Bin, StringBin2DecAsc, StringDecAsc2Float, StringFloat2DecAsc, StringHexAsc2Bin, StringBin2HexAsc, StringLength, StringCat, StringCompare, StringCompareNoCase, StringFind, StringReverseFind, StringFindOneOf, StringIncluding, StringExcluding, StringToUpper, StringToLower, StringToReverse, StringTrimLeft, StringTrimRight, StringInsert

18.4 Команды - Statement

18.4.1 Объявление переменных

Включает в себя объявление переменных и массивов. Общая структура выглядит следующим образом:

type name который определяет тип имени

Пример:

```
int A    //определение переменной A как целого числа
```

type name[constant] который определяет тип имени массива

Пример:

```
int B[10]    //переменная B задает одномерный массив размерности 10.
```

18.4.2 Присвоение

Команды присвоения используют операторы присвоения для перемещения результата выражения в правой части оператора в адрес переменной слева. Выражение представляет собой сочетание переменных, констант и операторов, дающее выходной результат.

Variable = Expression (Переменная = Выражение)

Пример

A = 2 Где переменной A присвоено значение 2

18.4.3 Логические команды

Логические команды выполняют действия, в зависимости от условий булевого выражения.

Синтаксис команд следующий:

Однолинейный формат

```
if <Condition> then
    [Statements]
else
    [Statements]
end if
```

Пример:

```
if a == 2 then
    b = 1
else
    b = 2
end if
```

Групповой формат - Block Format

```
If <Condition> then
    [Statements]
else if <Condition – n> then
    [Statements]
else
    [Statements]
end if
```

Пример:

```
if a == 2 then
        b = 1
else if a == 3
        b = 2
else
        b = 3
end if
```

Описание синтаксиса:

if	Всегда стоит в начале команды
<Condition>	Обязательная часть. Это управляющее условие. Оно считается FALSE - ложным, когда условие <Condition> равно 0 и считается TRUE - истинным, когда условие не равно 0.
then	Предшествует командам, выполняемым при истинном значении условия.
[Statements]	Это необязательный элемент блок-формате, но необходимый в однолинейном формате без использования *else*. Команда будет выполнена при истинном условии <Condition-n> — TRUE .
else if	Необязательная часть. Команды данной ветви будут выполняться, когда соответствующее условие <Condition-n> — TRUE (истинно).
<Condition-n>	Необязательная часть. См. <Condition>
else	Необязательная часть. Команды данной ветви будет выполняться, когда оба условия: <Condition> и <Condition-n> ложные.
end if	Обязательно стоит в конце условной команды (if-then).

18.4.4 Команды выбора - Selective Statements

Конструкция «select-case» может быть использована для выполнения выбранной группы действий в зависимости от назначенной переменной. Действия при совпадении варианта производятся до чтения команды «break». Синтаксис следующий.

Default case free Format

```
Select Case [variable]
Case [value]
    [Statements]
    break
    end Select
```

Пример:

```
Select Case A
    Case 1
        b=1
    break
end Select
```

Default case Format

```
Select Case [variable]
Case [value]
    [Statements]
    break
    Case else
    break

  end Select
```

Пример:

```
Select Case A
    Case 1
        b=1
    break
    Case else
        b=0
end Select
```

Multiple cases in the same block – Несколько случаев в одном блоке

```
Select Case [variable]
Case [value1]
    [Statements]
Case [value2]
    [Statements]
    break
  end Select
```

Пример:

```
    Select Case A
        Case 1
        Case 2
            b=2
        Case 3
            b=3
        break
     end Select
```

Описание синтаксиса:

Select Case	Должен быть использован в начале команды
[variable]	Обязательный. Значение этой переменной будет сравниваться со значением каждого случая «case».
Case else	Опциональный. Представляет случай по умолчанию. Если ни один из случаев не совпал, будет выполнен случай по умолчанию. Если случай по умолчанию отсутствует, произойдет сброс к концу команд *select-case* statements, если нет совпадающего случая.
break	Опциональный. Команды совпадающего случая будут выполняться до достижеия команды *break*. Если команда *break* отсутствует, происходит просто переход к выполнению следующей команды до достижения команды *end*.
end Select	Показывает конец команд *select-case*

18.4.5 Циклы - Reiterative Statements

Reiterative Statements – Команды циклов нужны для повторяющихся в зависимости от условий задач. Есть два типа циклов.

18.4.5.1 Цикл for-next

Конструкция [for-next] используется для выполнения заданного числа циклов. Переменная используется в качестве счетчика и проверки условия окончания цикла. Используйте данную конструкцию для фиксированного числа циклов. Синтаксис команды:

```
for [Counter] = <StartValue> to <EndValue> [step <StepValue>]
    [Statements]
next [Counter]
```

или

```
for [Counter] = <StartValue> down <EndValue> [step <StepValue>]
    [Statements]
next [Counter]
```

Пример:

```
for a = 0 to 10 step 2
    b = a
next a
```

Описание синтаксиса:

for	Стоит в начале конструкции
[Counter]	Обязательно. Это управляющий элемент. Результат вычисления переменной служит для проверки сравнения.
<StartValue>	Обязательно. Начальное значение переменной [Counter].
to/down	Обязательно. Определяет направление изменения счетчика <Counter> – увеличение или уменьшение. “to” — увеличение счетчика <Counter> на значение <Step Value>. “down” — уменьшение счетчика <Counter> на значение <StepValue>.
<EndValue>	Обязательно. Конечная точка (test point). Если значение счетчика превышает данное значение, цикл заканчивается.
step	Необязательно. Определяет значение <Step Value>, отличное от единицы.
[StepValue]	Необязательно. Шаг увеличения/уменьшения счетчика. Его можно не указывать, если он равняется 1, т.к. в этом случае <StepValue> считается равным 1 по умолчанию.
[Statements]	Необязательно. Команды, выполняемые когда сравнение -TRUE. Можно использовать вложенные циклы.
next	Обязательно.
[Counter]	Необязательно. Используется при вложенных циклах.

18.3.4.2 Цикл [while-wend]

Конструкция [while-wend] используется при неизвестном числе выполнения циклов. Переменная используется для проверки выполнения условия окончания цикла. Когда условие TRUE — истина, выполнение команд повторяется до тех пор, пока условие не станет ложным - FALSE.

Синтаксис следущий:

```
while <Condition>
    [Statements]
wend
```

Пример:

```
while a < 10
        a = a + 10
wend
```

Описание синтаксиса:

while	Используется в начале конструкции.
continue	Обязательно. Это управляющая команда. Когда она становится истиной, цикл начинает выполнение.Когда она становится ложной, цикл заканчивает выполнение.
return [value]	Команды, выполняемые при истинном значении условия.
wend	Показывает конец оператора [while-wend]

18.3.4.3 Другие управляющие команды

break	Используется в конструкциях [for-next] и [while-wend]. Приводит к немедленному выходу из цикла.
continue	Используется в циклах [for-next] и [while-wend]. Вызывает завершение текущей итерации цикла и начало следующей.

18.5 Функциональные блоки

Функциональные блоки применяются для уменьшения повторов одного и того же когда, должны быть определены до использования в программе и могут использовать любые типа переменных и команд. Для вызова функционального блока в тексте главной функции макроса [Main Macro Function] нужно указать его имя и определить параметры в скобках. После того, как функциональный блок отработал, он возвращает значение в главную функцию -Main Function, где используется для операции присвоения или в условии. Возвращаемый тип не обязателен в определении функции, так как не всегда необходимо возвращать значение. Параметры могут также отсутствовать в определении функции, если функция не нуждается в получении параметров от Main Function.Синтаксис следущий:

Определение функции возвращаемого типа:

```
sub type <name> [(parameters)]
    Local variable declarations
    [Statements]
    [return [value]]
end sub
```

Пример:

```
sub int Add(int x, int y)
    int result
    result = x +y
    return result
end sub
macro_command main()
    int a = 10, b = 20, sum
    sum = Add(a, b)
end macro_command
```

или:

```
sub int Add()
    int result, x=10, y=20
    result = x +y
    return result
end sub
macro_command main()
    int sum
    sum = Add()
end macro_command
```

Определение функции не возвращаемого типа:

```
sub <name> [(parameters)]
    Local variable declarations
    [Statements]
end sub
```

Example:

```
sub Add(int x, int y)
    int result
    result = x +y
 end sub

macro_command main()
    int   a = 10, b = 20
    Add(a, b)
end macro_command
```

or:

```
sub Add()
    int result, x=10, y=20
    result = x +y
 end sub
```

Описание синтаксиса:

sub	Должен стоять в начале функционального блока
type	Обязательно. Это тип значения, возвращаемого функцией.
(parameters)	Опционально. Параметры, значения которых передаются в функцию при вызове ее в [main]. Передаваемые параметры должны иметь тип, указанный в поле параметров и назначенные имена. Например: sub int MyFunction(int x, int y). x и y — целые значения, передаваемые из главной программы. Функция вызывается следующим образом: ret = MyFunction(456, pressure). При вызове функции в качестве параметров можно указывать константы или переменные. После выполнения функции переменной “ret“ будет присвоено целое значение, возвращаемой функцией.
Local variable declaration	Переменные, которые используются в функциональном блоке, должны быть сначала объявлены. Это переменные, помимо передаваемых в функцию из основной программы. В примере выше x и y — переменные, которые могут быть использованы в функции. Глобальные переменные также доступны для использования в функциональном блоке.
[Statements]	Исполняемые команды
[return [value]]	Опционально. Используется для возвращения значения при вызове функции. Значение может быть константой или переменной. Этот оператор также является окончанием функционального блока. Функциональному блоку не обязательно возвращать значение, но когда определен возвращаемый тип, команда возвращения обязательна.
end sub	Должен стоять в конце функционального блока.

18.6 Встроенные функциональные блоки

EasyBuilderPro имеет некоторые встроенные функции для считывания и перемещения данных в ПЛК, управления данными и математические функции.

18.6.1 Математические функции

Имя	SQRT
Синтаксис	SQRT(source, result) - исходная переменная, результат
Описание	Вычисляет корень квадратный из исходной переменной, присваивает полученное значение переменной «результат». Исходная переменная может быть константой или переменной, но результат должен быть - переменной. Исходная переменная должна быть неотрицательной величиной.
Пример	macro_command main() float source, result SQRT(15, result) source = 9.0 SQRT(source, result)// result is 3.0 end macro_command

Имя	CUBERT
Синтаксис	CUBERT (source, result) - исходная переменная, результат
Описание	Вычисляет корень кубический из исходной переменной, присваивает полученное значение переменной «результат». Исходная переменная может быть константой или переменной, но результат должен быть - переменной. Исходная переменная должна быть неотрицательной величиной.

Пример	macro_command main()
	float source, result
	
	CUBERT (27, result) // result is 3.0
	
	source = 27.0
	CUBERT(source, result)// result is 3.0
	end macro_command

Имя	POW
Синтаксис	POW (source1, source2, result) - исходные переменные 1 и 2, результат
Описание	Вычисляет «source1» возведенный в степень «source2». Исходные переменные 1 и 2 могут быть константами или переменными, но результат должен быть - переменной. Исходные переменные должны быть неотрицательными величинами.
Пример	macro_command main() float y, result y = 0.5 POW (25, y, result) // result = 5 end macro_command macro_command

Имя	SIN
Синтаксис	SIN (source, result) - исходная переменная, результат
Описание	Вычисляет синус исходной переменной (может быть и константой), присваивает полученное значение переменной «результат».

Пример	macro_command main()
	float source, result
	SIN(90, result)// result is 1
	source = 30
	SIN(source, result)// result is 0.5
	
	end macro_command

Имя	COS
Синтаксис	COS (source, result) - исходная переменная, результат
Описание	Вычисляет косинус исходной переменной (может быть и константой), присваивает полученное значение переменной «результат».

Пример	macro_command main()
	float source, result
	COS(90, result)// result is 0
	source = 60
	GetData(source, "Local HMI", LW, 0, 1)
	COS(source, result)// result is 0.5
	
	end macro_command

Имя	TAN
Синтаксис	TAN (source, result) - исходная переменная, результат
Описание	Вычисляет тангенс исходной переменной (может быть и константой), присваивает полученное значение переменной «результат».

Пример	macro_command main()
	float source, result
	TAN(45, result)// result is 1
	source = 60
	TAN(source, result)// result is 1.732
	end macro_command

Имя	COT
Синтаксис	COT(source, result) - исходная переменная, результат
Описание	Вычисляет котангенс исходной переменной (может быть константой), присваивает полученное значение переменной «результат».

Пример	macro_command main()
	float source, result
	COT(45, result)// result is 1
	source = 60
	COT(source, result)// result is 0.5774
	end macro_command

Имя	SEC
Синтаксис	SEC (source, result) - исходная переменная, результат
Описание	Вычисляет секанс исходной переменной (может быть и константой), присваивает полученное значение переменной «результат».

Пример	macro_command main()
	float source, result
	SEC(45, result)// result is 1.414
	source = 60
	SEC(source, result)// if source is 60, result is 2
	end macro_command

Имя	CSC
Синтаксис	CSC(source, result) -исходная переменная, результат
Описание	Вычисляет косеканс исходной переменной (может быть и константой), присваивает полученное значение переменной «результат».

Пример	macro_command main()
	float source, result
	CSC(45, result)// result is 1.414
	source = 30
	CSC(source, result)// result is 2
	end macro_command

Имя	ASIN
Синтаксис	ASIN (исходная переменная, результат)
Описание	Вычисляет арксинус исходной переменной (может быть константой), присваивает полученное значение переменной «результат».

Пример	macro_command main()
	float source, result
	ASIN(0.8660, result)// result is 60
	source = 0.5
	ASIN(source, result)// result is 30
	end macro_command

Имя	ACOS
Синтаксис	ACOS (source, result) - исходная переменная, результат
Описание	Вычисляет арккосинус исходной переменной (может быть константой), присваивает полученное значение переменной «результат».

Пример	macro_command main()
	float source, result
	ACOS(0.8660, result)// result is 30
	source = 0.5
	ACOS(source, result)// result is 60
	end macro_command

Имя	ATAN
Синтаксис	ATAN (source, result) - исходная переменная, результат
Описание	Вычисляет арктангенс исходной переменной (может быть и константой), присваивает полученное значение переменной «результат».

Пример	macro_command main()
	float source, result
	ATAN(1, result)// result is 45
	source = 1.732
	ATAN(source, result)// result is 60
	end macro_command

Имя	LOG
Синтаксис	LOG (source, result) - исходная переменная, результат
Описание	Вычисляет натуральный логарифм числа. Источник может быть переменной или константой, результат должен быть переменной
Пример	macro_command main() float source = 100, result LOG (source, result)// result is approximately 4.6052 end macro_command

Имя	LOG 10
Синтаксис	LOG 10(source, result) - исходная переменная, результат
Описание	Вычисляет десятичный логарифм числа. Источник может быть переменной или константой, результат должен быть переменной
Пример	macro_command main() float source = 100, result LOG10 (source, result)// result is 2 end macro_command

Имя	RAND
Синтаксис	RAND(result) - результат
Описание	Вычисляет случайное целое число, и сохраняет в переменной «результат».
Пример	macro_command main() short result RAND (result)// result is not a fixed value when executes macro every time end macro_command

18.6.2 Преобразование данных

Имя	BIN2BCD
Синтаксис	BIN2BCD (исходные данные, результат)
Описание	Преобразует исходные данные из формата binary в значение типа BCD. Исходными данными могут быть константы или переменные, результат – всегда переменная.

Пример	macro_command main()
	short source, result
	BIN2BCD(1234, result)// result is 0x1234
	source = 5678
	BIN2BCD(source, result)// result is 0x5678
	end macro_command

Имя	BCD2BIN
Синтаксис	BCD2BIN (исходные данные, результат)
Описание	Преобразует исходные данные из формата BCD в значение типа binary. Исходными данными могут быть константы или переменные, результат – всегда переменная.

Пример	macro_command main()
	short source, result
	BCD2BIN(0x1234, result)// result is 1234
	source = 0x5678
	BCD2BIN(source, result)// result is 5678
	end macro_command

Имя	DEC2ASCII
Синтаксис	DEC2ASCII (исходные данные, result[start], len).
Описание	Преобразует исходные данные десятичного формата в строку символов ASCII, сохраняемую в массив (result). Параметр [len] определяет длину строки, единицы измерения зависят от типа результата, т.е. если тип “char”(размер равен 1 байту), то длина строки равна: byte*len. Если тип результата “short”, то длина строки равна: word*len и т.д. Первый символ помещается в переменную result[start], второй в result[start + 1], последний символ строки помещается в элемент массива result[start + (len -1)]. Параметры [source] и [len] могут быть константами или переменными, но результат – всегда переменная. [start] должен быть константой.

Пример	macro_command main()
	short source
	char result1[4]
	short result2[4]
	source = 5678
	DEC2ASCII(source, result1[0], 4)
	// result1[0] is '5', result1[1] is '6', result1[2] is '7', result1[3] is '8'
	// the length of the string (result1) is 4 bytes(= 1 * 4)
	DEC2ASCII(source, result2[0], 4)
	// result2[0] is '5', result2[1] is '6', result2[2] is '7', result2[3] is '8'
	// the length of the string (result2) is 8 bytes(= 2 * 4)
	end macro_command

Имя	HEX2ASCII
Синтаксис	HEX2ASCII (исходные данные, result[start], len).
Описание	Преобразует исходные данные шестнадцатеричного формата в строку символов ASCII, сохраняемую в массив (result). Параметр [len] определяет длину строки, единицы измерения зависят от типа результата, т.е. если тип “char”(размер равен 1 байту), то длина строки равна: byte*len. Если тип результата “short”, то длина строки равна: word*len и т.д. Первый символ помещается в переменную result[start], второй в result[start + 1], последний символ строки помещается в элемент массива result[start + (len -1)]. Параметры [source] и [len] могут быть константами или переменными, но результат – всегда переменная. [start] должен быть константой.
Пример	macro_command main() short source char result[4] source = 0x5678 DEC2ASCII(source, result[0], 4) // result[0] is '5', result[1] is '6', result[2] is '7', result[3] is '8' end macro_command

Имя	FLOAT2ASCII
Синтаксис	FLOAT2ASCII (исходные данные, result[start], len).
Описание	Преобразует исходные данные формата плавающей запятой в строку символов ASCII, сохраняемую в массив (result). Параметр [len] определяет длину строки, единицы измерения зависят от типа результата, т.е. если тип “char”(размер равен 1 байту), то длина строки равна: byte*len. Если тип результата “short” (размер равен 1 слову), то длина строки равна: word*len и т.д. Первый символ помещается в переменную result[start], второй в result[start + 1], последний символ строки помещается в элемент массива result[start + (len -1)]. Параметры [source] и [len] могут быть константами или переменными, но результат – всегда переменная. [start] должен быть константой.
Пример	macro_command main() float source char result[4] source = 0x5678 FLOAT2ASCII (source, result[0], 4) // result[0] is '5', result[1] is '6', result[2] is '7', result[3] is '8' end macro_command

Имя	ASCII2DEC
Синтаксис	ASCII2DEC (source[start], result, len).
Описание	Преобразует исходные данные-строку в десятичный формат, сохраняет их в массив (result). Параметр [len] определяет длину строки. Первый символ помещается в переменную source[start]. Параметры [source] и [len] могут быть константами или переменными, но результат – всегда переменная. [start] должен быть константой.

Пример	macro_command main()
	char source[4]
	short result
	source[0] = '5'
	source[1] = '6'
	source[2] = '7'
	source[3] = '8'
	ASCII2DEC(source[0], result, 4) // result is 5678
	end macro_command

Имя	ASCII2HEX
Синтаксис	ASCII2HEX (source[start], result, len).
Описание	Преобразует исходные данные-строку в шестнадцатеричный формат, сохраняет их в массив (result). Параметр [len] определяет длину строки. Первый символ помещается в переменную source[start]. Параметры [source] и [len] могут быть константами или переменными, но результат – всегда переменная. [start] должен быть константой.

Пример	macro_command main()
	char source[4]
	short result
	source[0] = '5'
	source[1] = '6'
	source[2] = '7'
	source[3] = '8'
	ASCII2DEC(source[0], result, 4) // result is 0x5678
	end macro_command

Имя	ASCII2FLOAT
Синтаксис	ASCII2FLOAT (source[start], result, len).
Описание	Преобразует исходные данные-строку в формат с плавающей запятой, сохраняет их в массив (result). Параметр [len] определяет длину строки. Первый символ помещается в переменную source[start]. Параметры [source] и [len] могут быть константами или переменными, но результат – всегда переменная. [start] должен быть константой.
Пример	macro_command main() char source[4] float result source[0] = '5' source[1] = '6' source[2] = '.' source[3] = '8' ASCII2FLOAT (source[0], result, 4) // result is 56.78 end macro_command

18.5.3 Обработка данных

Имя	FILL
Синтаксис	FILL (source[start], preset, count).
Описание	Присваивает первым «*count*» элементам массива указанное значение (preset). *Source u start* должны быть переменными, *preset* может константой или переменной.

Пример	macro_command main()
	char result[4]
	char preset
	FILL(result[0], 0x30, 4)
	// result[0] is 0x30, result[1] is 0x30, , result[2] is 0x30, , result[3] is 0x30
	preset = 0x31
	FILL(result[0], preset, 2) // result[0] is 0x31, result[1] is 0x31
	end macro_command

Имя	SWAPB
Синтаксис	SWAPB (source, result).
Описание	Меняет местами старший и младший байты 16-разрядных исходных данных source и сохраняет полученный результат в [result]. Source может быть как константой, так и переменной, но [result] может быть только переменной.

Пример	macro_command main()
	short source, result
	SWAPB(0x5678, result)// result is 0x7856
	source = 0x123
	SWAPB(source, result)// result is 0x2301
	end macro_command

Имя	SWAPW
Синтаксис	SWAPW (source, result).
Описание	Меняет местами старший и младший байты 32-разрядных исходных данных source и сохраняет полученный результат в [result]. Source может быть как константой, так и переменной, но [result] может быть только переменной.

Пример	macro_command main()
	int source, result
	SWAPB(0x12345678, result)// result is 0x78561234
	source = 0x12345
	SWAPB(source, result)// result is 0x23450001

	end macro_command

Имя	LOBYTE
Синтаксис	LOBYTE (source, result).
Описание	Записывает младший байт 16-разрядных входных данных source в [result]. [Source] может быть константой или переменной, но [result] должен быть только переменной.

Пример	macro_command main()
	short source, result
	LOBYTE(0x1234, result)// result is 0x34
	source = 0x123
	LOBYTE(source, result)// result is 0x23
	end macro_command

Имя	HIBYTE
Синтаксис	HIBYTE (source, result).
Описание	Записывает старший байт 16-разрядных входных данных source в [result]. [Source] может быть константой или переменной, но [result] должен быть только переменной.

Пример	macro_command main()
	short source, result
	HIBYTE(0x1234, result)// result is 0x12
	source = 0x123
	HIBYTE(source, result)// result is 0x01
	end macro_command

Имя	LOWORD
Синтаксис	LOWORD (source, result).
Описание	Записывает младшее слово 32-разрядных входных данных source в [result]. [Source] может быть константой или переменной, но [result] должен быть только переменной.

Пример	macro_command main()
	int source, result
	LOWORD(0x12345678, result)// result is 0x5678
	source = 0x12345
	LOWORD(source, result)// result is 0x2345
	end macro_command

Имя	HIWORD
Синтаксис	HIWORD (source, result).
Описание	Записывает старшее слово 32-разрядных входных данных source в [result]. [Source] может быть константой или переменной, но [result] должен быть только переменной.

Пример	macro_command main()
	int source, result
	HIWORD(0x12345678, result)// result is 0x1234
	source = 0x12345
	HIWORD(source, result)// result is 0x0001
	end macro_command

18.5.4 Преобразование битов

Имя	GETBIT
Синтаксис	GETBIT (source, result, bit_pos)
Описание	Возвращает в [result] состояние указанного в [source] бита. Значение [result] может быть 0 или 1. [source] и [bit_pos] могут быть константами или переменными, но [result] должен быть переменной.

Пример	macro_command main()
	int source, result
	short bit_pos
	GETBIT(9, result, 3)// result is 1
	source = 4
	bit_pos = 2
	GETBIT(source, result, bit_pos)// result is 1
	end macro_command

Имя	SETBITON
Синтаксис	SETBITON (source, result, bit_pos)
Описание	Изменяет состояние указанного бита [source] на 1 и помещает полученные данные в [result]. [source] и [bit_pos] могут быть константами или переменными, но [result] должен быть переменной.

Пример	macro_command main()
	int source, result
	short bit_pos
	SETBITON(1, result, 3)// result is 9
	source = 0
	bit_pos = 2
	SETBITON (source, result, bit_pos)// result is 4
	end macro_command

Имя	SETBITOFF
Синтаксис	SETBITOFF (source, result, bit_pos)
Описание	Изменяет состояние указанного бита [source] на 0 и помещает полученные данные в [result]. [source] и [bit_pos] могут быть константами или переменными, но [result] должен быть переменной.

Пример	macro_command main()
	int source, result
	short bit_pos
	SETBITOFF(9, result, 3)// result is 1
	source = 4
	bit_pos = 2
	SETBITFF(source, result, bit_pos)// result is 0
	end macro_command

Имя	INVBIT
Синтаксис	INVBIT (source, result, bit_pos)
Описание	Изменяет состояние указанного бита [source] на противоположное и помещает полученные данные в [result]. [source] и [bit_pos] могут быть константами или переменными, но [result] должен быть переменной.

Пример	macro_command main()
	int source, result
	short bit_pos
	INVBIT(4, result, 1)// result = 6
	source = 6
	bit_pos = 1
	INVBIT(source, result, bit_pos)// result = 4
	end macro_command

18.5.5 Коммуникации

Имя	DELAY
Синтаксис	DELAY (time)
Описание	Приостанавливает выполнение текущего макроса на время, определяемое параметром time. Единицы измерения: миллисекунды. [time] может быть константой или переменной.

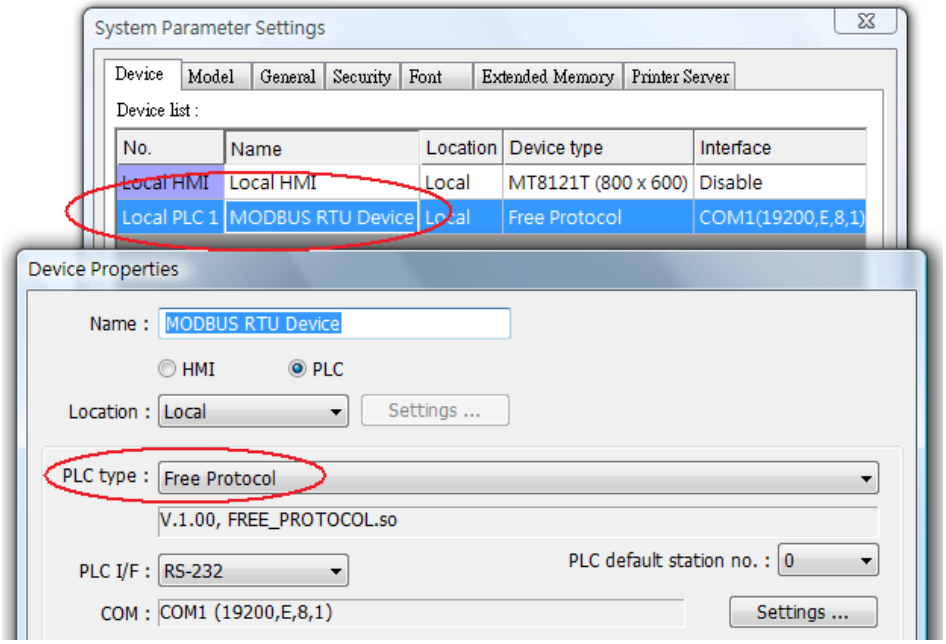
Пример	macro_command main()
	int time == 500
	DELAY(100)// delay 100 ms
	DELAY(time)// delay 500 ms
	end macro_command

Имя	ADDSUM
Синтаксис	ADDSUM (source[start], result, data_count)
Описание	Увеличивает элементы массива source с номера [start] по [start+data_count-1] для получения контрольной суммы. Записывает полученную сумму в [result] (переменная). Data_count — число изменяемых элементов — может быть константой или переменной.
Пример	macro_command main() char data[5] short checksum data[0] = 0x1 data[1] = 0x2 data[2] = 0x3 data[3] = 0x4 data[4] = 0x5 ADDSUM(data[0], checksum, 5)// checksum is 0xf end macro_command

Имя	XORSUM
Синтаксис	XORSUM (source[start], result, data_count)
Описание	Для вычисления контрольной суммы используется метод исключения с номера [start] по [start+data_count-1]. Записывает полученную сумму в [result] (переменная). Data_count — количество вычисляемых элементов массива — может быть константой или переменной.
Пример	macro_command main() char data[5] = {0x1, 0x2, 0x3, 0x4, 0x5} short checksum XORSUM(data[0], checksum, 5)// checksum is 0x1 end macro_command

Имя	CRC
Синтаксис	CRC (source[start], result, data_count)
Описание	Вычисляет 16-разрядную контрольную циклическую сумму для переменных с source[start] по source[start + count -1]. Записывает полученную сумму в [result] (переменная). Data_count — количество вычисляемых элементов массива — может быть константой или переменной.
Пример	macro_command main() char data[5] = {0x1, 0x2, 0x3, 0x4, 0x5} short 16bit_CRC CRC(data[0], 16bit_CRC, 5)// 16bit_CRC is 0xbb2a end macro_command

Имя	OUTPORT
Синтаксис	OUTPORT (source[start], device_name, data_count)
Описание	Посылает элементы массива с source[start] по source[start + count -1] в ПЛК через COM-порт или по Ethernet. device_name — название устройства, определенное в таблице устройств [device table] и устройство должно иметь тип “Free Protocol”. Data_count — количество вычисляемых элементов массива — может быть константой или переменной.
Пример	Для использования функции OUTPORT, сначала необходимо добавить устройство типа “Free Protocol”:



Устройство с именем ”MODBUS RTU Device”. Атрибуты порта зависят от настроек устройства (текущие установки: “19200,E, 8, 1”).

Ниже приведен пример выполнения записи в ПЛК (установление бита в ON) – устройство MODBUS.

```
macro_command main()
char command[32]
short address, checksum
FILL(command[0], 0, 32)          // command initialization
command[0] = 0x1                 // station no
command[1] = 0x5                 // function code : Write Single Coil
address = 0
HIBYTE(address, command[2])
LOBYTE(address, command[3])
command[4] = 0xff// force bit on
command[5] = 0
CRC(command[0], checksum, 6)
LOBYTE(checksum, command[6])
HIBYTE(checksum, command[7])
// send out a “Write Single Coil” command
OUTPORT(command[0], "MODBUS RTU Device", 8)
end macro_command
```

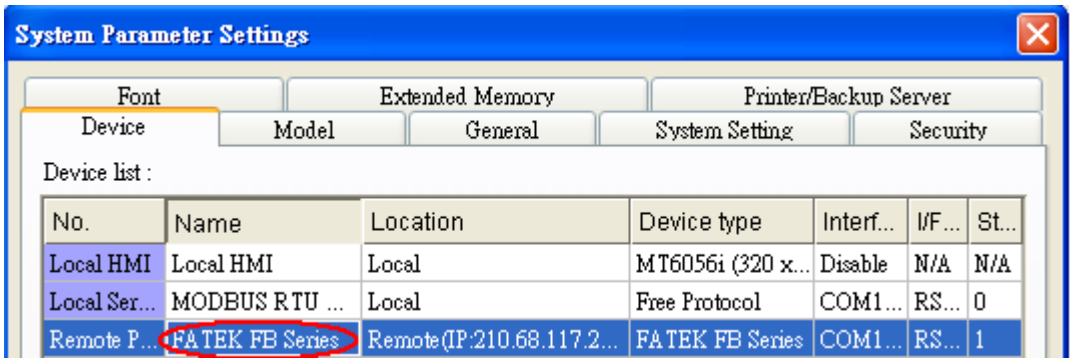
Имя	INPORT
Синтаксис	INPORT(read_data[start], device_name, read_count, return_value)
Описание	Считывает данные через COM-порт или по Ethernet. Затем сохраняет их в read_data[start]~ read_data[start + read_count - 1]. device_name — название устройства, определенное в таблице устройств [device table] и устройство должно иметь тип “Free Protocol”. Read_count — число считываемых элементов — постоянная или переменная. Если функция успешно завершает получение данных, то значение [return_value] устанавливается в 1, в противном случае – в 0.

Пример	Ниже приведен пример выполнения операции чтения из [holding registers] устройства MODBUS.
	
	// Read Holding Registers
	
	macro_command main()
	
	char command[32], response[32]
	
	short address, checksum
	
	short read_no, return_value, read_data[2]
	
	FILL(command[0], 0, 32)// command initialization
	
	FILL(response[0], 0, 32)
	
	command[0] = 0x1// station no
	
	command[1] = 0x3// function code : Read Holding Registers
	
	address = 0
	
	HIBYTE(address, command[2])
	
	LOBYTE(address, command[3])
	
	read_no = 2// read 2 words (4x_1 and 4x_2)
	
	HIBYTE(read_no, command[4])
	
	LOBYTE(read_no, command[5])
	
	CRC(command[0], checksum, 6)
	
	LOBYTE(checksum, command[6])
	
	HIBYTE(checksum, command[7])
	
	// send out a ‘Read Holding Registers” command
	
	OUTPORT(command[0], "MODBUS RTU Device", 8)
	
	// read responses for a ‘Read Holding Registers” command

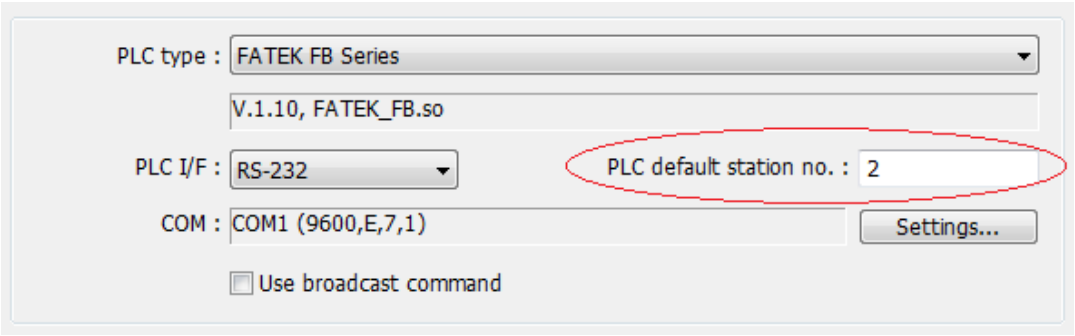
	INPORT(response[0], "MODBUS RTU Device", 9, return_value)
	
	if return_value > 0 then
	
	read_data[0] = response[4] + (response[3] << 8)// data in 4x_1
	
	read_data[1] = response[6] + (response[5] << 8)// data in 4x_2
	
	SetData(read_data[0], "Local HMI", LW, 100, 2)
	
	end if
	
	end macro_command

Имя	INPORT2
Синтаксис	INPORT2(responce[start], device_name, receive_len, wait_time)
Описание	Считывает данные через COM-порт или по Ethernet. Затем сохраняет их в *responce*. Описание *device_name* — должно быть тем же самым, что и в OUTPORT. *receive_len*— число полученных данных — должно быть переменной. *wait_time* (миллисекунд) может быть константой или переменной. После считывания данных, если за время назначенного интервала не было приходящих данных, функция возвращается.

Пример	macro_command main()
	
	short wResponse[6], receive_len, wait_time=20
	
	INPORT2(wResponse[0], "Free Protocol", receive_len, wait_time)
	
	// wait_time unit : millisecond
	
	if receive_len > 0 then
	
	  SetData(wResponse[0], "Local HMI", LW, 0, 6)
	
	  // set responses to LW0
	end if
	
	end macro_command

Имя	GetData
Синтаксис	GetData(read_data[start], device_name, device_type, address_offset, data_count) или GetData(read_data, device_name, device_type, address_offset, 1)
Описание	Принимает данные из ПЛК. Данные сохраняются в read_data[start]~ read_data[start + data_count - 1]. [data_count] — количество принимаемых элементов. В общем случае read_data — массив, но если параметр data_count равен 1, это будет массив из одной переменной. Далее показаны два способа чтения одного слова данных из ПLК macro_command main() short read_data_1[2], read_data_2 GetData(read_data_1[0], “FATEK KB Series”, RT, 5, 1) GetData(read_data_2, “FATEK KB Series”, RT, 5, 1) end macro_command [device_name] — имя ПЛК, заключенное в двойные кавычки (“), это имя должно быть определено в списке устройств окна [system parameters] как показано ниже (см. FATEK KB Series):  [device_type] — тип адреса и метод кодирования (binary или BCD) данных ПЛК. Например, если этот параметр задан LW_BIN — это означает, что используется регистр типа LW и метод кодирования – binary. Если используется метод кодирования BIN, запись “_BIN” может быть проигнорирована. Если device_type определен как LW_BCD, это означает, что используется регистр типа LW и метод кодирования BCD. [address_offset] —адрес смещения в ПЛК. Например, если вызвана функция GetData(read_data_1[0], “FATEK KB Series”, RT, 5, 1), то этот адрес равен 5. Если [address_offset] использует формат “N#AAAAA”, N показывает, что номер станции ПЛК равен N. Например, GetData(read_data_1[0], “FATEK KB Series”,

RT, 2#5, 1) показывает, что номер стойки ПЛК равен 2. Если функция GetData() использует номер станции по умолчанию, определенный в списке устройств, как показано ниже, то не нужно определять address_offset.



Число регистров чтения (последняя колонка табл.) зависит и от типа переменной read_data и от значения параметра data_count.

type of read_data	data_count	actual number of 16-bit register read
char (8-bit)	1	1
char (8-bit)	2	1
bool (8-bit)	1	1
bool (8-bit)	2	1
short (16-bit)	1	1
short (16-bit)	2	2
int (32-bit)	1	2
int (32-bit)	2	4
float (32-bit)	1	2
float (32-bit)	2	4

Когда GetData() использует 32-разрядный тип данных (целых или вещественных), функция автоматически будет преобразовывать данные. Например:

```
macro_command main()
float f
GetData(f, "MODBUS", 6x, 2, 1)    // f will contain a floating point value
end macro_command
```

Пример

```
macro_command main()
bool a
bool b[30]
short c
short d[50]
int e
int f[10]
```

	double g[10]
	// запись состояния бита LB2 в переменную
	GetData(a, “Local HMI”, LB, 2, 1)
	// запись состояний 30 битов: LB0 ~ LB29 в переменные: b[0] ~ b[29]
	GetData(b[0], “Local HMI”, LB, 0, 30)
	// запись одного слова LW2 в переменную c
	GetData(c, “Local HMI”, LW, 2, 1)
	// запись 50 слов: LW0 ~ LW49 — в переменные: d[0] ~ d[49]
	GetData(d[0], “Local HMI”, LW, 0, 50)
	// запись 2-х слов: LW6 ~ LW7 — в переменную e
	// тип переменной e: int
	GetData(e, “Local HMI”, LW, 6, 1)
	// запись 20 слов (10 целых значений): LW0 ~ LW19 — в f[0] ~ f[9]
	// т.к. каждое значение занимает в памяти два слова
	GetData(f[0], “Local HMI”, LW, 0, 10)
	// запись двух слов из LW2 ~ LW3 в переменную f
	GetData(f, “Local HMI”, LW, 2, 1)
	end macro_command

Имя	GetDataEx
Синтаксис	GetDataEx(read_data[start], device_name, device_type, address_offset, data_count) или GetDataEx(read_data, device_name, device_type, address_offset, 1)
Описание	Принимает данные из ПЛК и продолжает выполнение следующей команды, даже если нет ответа от устройства. Описание *read_data, device_name, device_type, address_offset и data_count* – те же самые, что и в GetData.

Пример	macro_command main()
	bool a
	bool b[30]
	short c
	short d[50]
	int e
	int f[10]
	double g[10]
	// запись состояния бита LB2 в переменную
	GetDataEx(a, “Local HMI”, LB, 2, 1)
	// запись состояний 30 битов: LB0 ~ LB29 в переменные: b[0] ~ b[29]
	GetDataEx(b[0], “Local HMI”, LB, 0, 30)

	// запись одного слова LW2 в переменную c
	GetDataEx(c, “Local HMI”, LW, 2, 1)
	// запись 50 слов: LW0 ~ LW49 — в переменные: d[0] ~ d[49]
	GetDataEx(d[0], “Local HMI”, LW, 0, 50)
	// запись 2-х слов: LW6 ~ LW7 — в переменную e
	// тип переменной e: int
	GetDataEx (e, “Local HMI”, LW, 6, 1)
	// запись 20 слов (10 целых значений): LW0 ~ LW19 — в f[0] ~ f[9]
	// т.к. каждое значение занимает в памяти два слова
	GetDataEx (f[0], “Local HMI”, LW, 0, 10)
	// запись двух слов из LW2 ~ LW3 в переменную f
	GetData(f, “Local HMI”, LW, 2, 1)
	end macro_command

Имя	SetData
Синтаксис	SetData(send_data[start], device_name, device_type, address_offset, data_count) или SetData(send_data, device_name, device_type, address_offset, 1)
Описание	Посылает данные в ПЛК. Данные определяются как элементы массива: send_data[start]~ send_data[start + data_count - 1]. [data_count] число отправляемых данных. В общем случае [send_data] — массив, но если data_count=1, то [send_data] состоит только из одной переменной. Ниже даны два метода для пересылки данных объемом в одно слово. macro_command main() short send_data_1[2] = { 5, 6}, send_data_2 = 5 SetData(send_data_1[0], “FATEK KB Series”, RT, 5, 1) SetData(send_data_2, “FATEK KB Series”, RT, 5, 1) end macro_command [device_name] — имя ПЛК, заключенное в двойные кавычки (“), и оно должно быть определено в списке устройств [system parameters]. [device_type] — тип адреса и метода кодирования (binary или BCD) данных ПЛК. Например, если этот параметр задан LW_BIN — это означает, что используется регистр типа LW и метод кодирования – binary. Если используется метод кодирования BIN, запись “_BIN” может быть проигнорирована. [address_offset] —адрес смещения в ПЛК. Например, если вызвана функция GetData(read_data_1[0], “FATEK KB Series”, RT, 5, 1), то этот адрес равен 5. Если [address_offset] использует формат “N#AAAAA”, N показывает, что номер

станции ПЛК равен N. Например, SetData(read_data_1[0], “FATEK KB Series”, RT, 2#5, 1) показывает, что номер стойки ПЛК равен 2. Если функция SetData() использует номер станции по умолчанию, определенный в списке устройств, как показано ниже, то не нужно определять address_offset.

Число регистров действительно посылаемых (последняя колонка табл.) зависит и от типа переменной send_data и от значения параметра data_count.

type of read_data	data_count	actual number of 16-bit register send
char (8-bit)	1	1
char (8-bit)	2	1
bool (8-bit)	1	1
bool (8-bit)	2	1
short (16-bit)	1	1
short (16-bit)	2	2
int (32-bit)	1	2
int (32-bit)	2	4
float (32-bit)	1	2
float (32-bit)	2	4

Когда SetData() использует 32-разрядный тип данных (целых или вещественных), функция автоматически будет преобразовывать данные. Например:

```
macro_command main()
float f = 2.6
SetData(f, "MODBUS", 6x, 2, 1)    // will send a floating point value to the device
end macro_command
```

Пример

```
macro_command main()
int i
bool a = true
bool b[30]
short c = false
short d[50]
int e = 5
int f[10]
for i = 0 to 29
  b[i] = true
next i
for i = 0 to 49
  d[i] = i * 2
```

	next i for i = 0 to 9 f [i] = i * 3 next i // set the state of LB2 SetData(a, “Local HMI”, LB, 2, 1) // set the states of LB0 ~ LB29 SetData(b[0], “Local HMI”, LB, 0, 30) // set the value of LW2 SetData(c, “Local HMI”, LW, 2, 1) // set the values of LW0 ~ LW49 SetData(d[0], “Local HMI”, LW, 0, 50) // set the values of LW6 ~ LW7, note that the type of e is int SetData(e, “Local HMI”, LW, 6, 1) // set the values of LW0 ~ LW19 // 10 integers are equal to 20 words, since each integer value occupies 2 words. SetData(f[0], “Local HMI”, LW, 0, 10) end macro_command

Имя	SetDataEx
Синтаксис	SetDataEx(send_data[start], device_name, device_type, address_offset, data_count) или SetDataEx(send_data, device_name, device_type, address_offset, 1)
Описание	Посылает данные в ПЛК и продолжает выполнение следующей команды, даже если нет ответа от устройства. Описание *read_data, device_name, device_type, address_offset и data_count* – те же самые, что и в SetData.
Пример	macro_command main() int i bool a = true bool b[30] short c = false short d[50] int e = 5 int f[10] for i = 0 to 29 b[i] = true next i for i = 0 to 49

	d[i] = i * 2
	next i
	for i = 0 to 9
	f [i] = i * 3
	next i
	// set the state of LB2
	SetDataEx(a, “Local HMI”, LB, 2, 1)
	// set the states of LB0 ~ LB29
	SetDataEx(b[0], “Local HMI”, LB, 0, 30)
	// set the value of LW2
	SetDataEx(c, “Local HMI”, LW, 2, 1)
	// set the values of LW0 ~ LW49
	SetDataEx(d[0], “Local HMI”, LW, 0, 50)
	// set the values of LW6 ~ LW7, note that the type of e is int
	SetDataEx(e, “Local HMI”, LW, 6, 1)
	// set the values of LW0 ~ LW19
	// 10 integers are equal to 20 words, since each integer value occupies 2 words.
	SetDataEx(f[0], “Local HMI”, LW, 0, 10)
	end macro_command

Имя	GetError
Синтаксис	GetError (err)
Описание	Получает код ошибки.

Пример	macro_command main()
	short err
	char byData[10]
	GetDataEx(byData[0], “MODBUS RTU”, 4x, 1, 10)// read 10 bytes
	// if err is equal to 0, it is successful to execute GetDataEx()
	GetErr(err)// save an error code to err
	end macro_command

Имя	PURGE
Синтаксис	PURGE (com_port)
Описание	com_port – соотносится с последовательными портами: COM1 ~COM 3. Номер порта может быть переменной или константой. Эта функция используется для очистки вхоного и выходного буферов связанных с соответствующим COM портом.
Пример	macro_command main() int com_port=3 PURGE (com_port) PURGE (1) end macro_command

Имя	SetRTS
Синтаксис	SetRTS(com_port, source)
Описание	Set RTS state for RS232. com_port – соотносится с последовательным портом COM1. Номер порта может быть переменной или константой. Source - может быть переменной или константой. Эта команда выставляет сигнал RTS, когда значение *source* больше, чем 0 и снимает сигнал RTS signal while the value of source equals to 0.

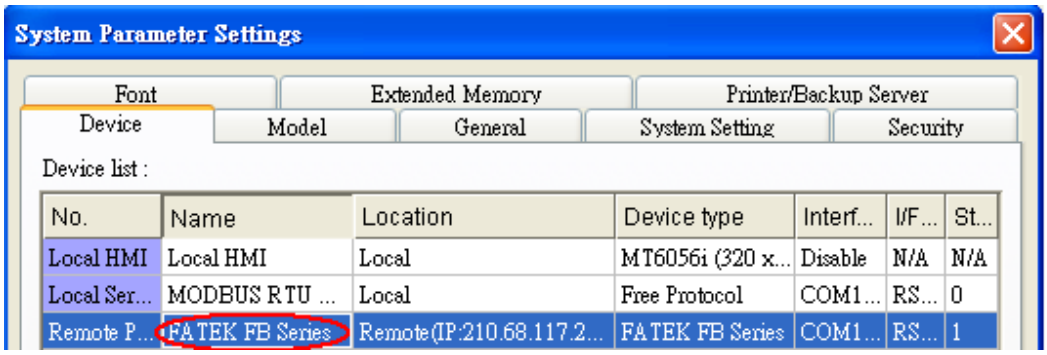
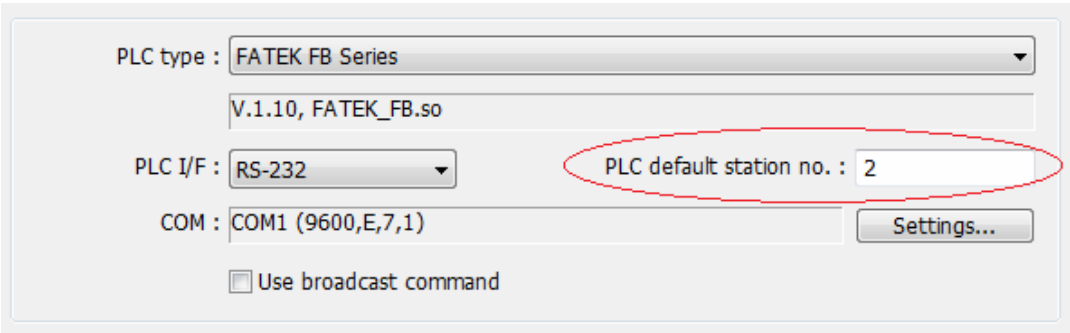
Пример	macro_command main()
	char com_port=1
	char value=1
	SetRTS(com_port, value) // raise RTS signal of COM1 while value>0
	SetRTS(1, 0) // lower RTS signal of COM1
	end macro_command

Имя	GetCTS
Синтаксис	GetCTS(com_port, result)
Описание	Получает состояние CTS сигнала порта RS232. com_port – соотносится с последовательным портом COM1. Номер порта может быть переменной или константой. *Result* используется для получания сигнала CTS, он должен быть переменной . Эта команда получает сигнал CTS и сохраняет в переменной *result*. Когда сигнал CTS на высоком уровне, в *result* записывается 1, иначе записывается 0.

Пример	macro_command main()
	char com_port=1
	char result
	GetCTS(com_port, result) // get CTS signal of COM1
	GetCTS (1, result) // get CTS signal of COM1
	end macro_command

Имя	Beep
Синтаксис	Beep ()
Описание	Воспроизводит звуковой сигнал. Эта команда воспроизводит звуковой сигнал 800 Герц и продолжительностью 30 миллисекунд.
Пример	macro_command main() Beep() end macro_command

18.6.6 Функции работы со строками

Имя	StringGet
Синтаксис	StringGet(read_data[start], device_name, device_type, address_offset, data_count)
Описание	Принимает данные из ПЛК. Данные сохраняются в **read_data[start]~ read_data[start + data_count - 1]**. *read_data* должен быть символьным массивом первого порядка. *Data_count* - количество принимаемых символов (characters), может быть константой или переменной. *Device_name* - имя ПЛК, заключенное в двойные кавычки (“), это имя должно быть определено в списке устройств окна [system parameters] как показано ниже (см. FATEK KB Series):  *Device_type* - тип адреса и метод кодирования (binary или BCD) данных ПЛК. Например, если этот параметр задан LW_BIN — это означает, что используется регистр типа LW и метод кодирования – binary. Если используется метод кодирования BIN, запись “_BIN” может быть проигнорирована. Если device_type определен как LW_BCD, это означает, что используется регистр типа LW и метод кодирования BCD. *Address_offset* — смещение адресов в ПЛК. Например, StringGet(read_data_1[0], “FATEK KB Series”, RT, 5, 1) представляет смещение адреса равно 5. Если *address_offset* использует формат – “N#AAAAA”, N показывает, что номер станции ПЛК равен N. AAAAA представляет смещение адреса . Этот формат используют, когда несколько ПЛК присоединены к одному порту. Например, StringGet(read_data_1[0], “FATEK KB Series”, RT, 2#5, 1) показывает, что номер станции ПЛК равен 2. Если StringGet() использует номер станции по умолчанию в списке устройств, как показано ниже, не является необходимым определять номер станции в *address_offset*. 

<table>
<tr><td></td><td>Число регистров чтения зависит от значения параметра data_count, которые read_data размещает в символьном массиве.

<table>
<tr><th>Тип read_data</th><th>data_count</th><th>actual number of 16-bit register read</th></tr>
<tr><td>char (8-bit)</td><td>1</td><td>1</td></tr>
<tr><td>char (8-bit)</td><td>2</td><td>1</td></tr>
</table>

Регистр длиной в 1 WORD (16-bit) равен по размеру 2-м символам ASCII. В соответствии с таблицей, чтение 2-х символов ASCII соответствует актуальному чтению одного 16-ти битового регистра.</td></tr>
<tr><td>Пример</td><td>macro_command main()
char str1[20]
// read 10 words from LW0~LW9 to the variables str1[0] to str1[19]
// since that 1 word can store 2 ASCII characters, reading 20 ASCII
// characters is actually reading 10 words of register
StringGet(str1[0], “Local HMI”, LW, 0, 20)
end macro_command</td></tr>
</table>

Имя	StringGetEx
Синтаксис	StringGetEx (read_data[start], device_name, device_type, address_offset, data_count)
Описание	Принимает данные из ПЛК и продолжает выполнение следующей команды, даже если нет ответа от устройства. Описание *read_data, device_name, device_type, address_offset и data_count* – те же самые, что и в GetData

Пример	macro_command main()
	char str1[20]
	short test=0
	// macro will continue executing test = 1 even if the MODBUS device is
	// not responding
	StringGetEx(str1[0], "MODBUS RTU", 4x, 0, 20)
	test = 1
	// macro will not continue executing test = 2 until MODBUS device responds
	StringGetEx(str1[0], "MODBUS RTU", 4x, 0, 20)
	test = 2
	end macro_command

Имя	StringSet
Синтаксис	StringSet(send_data[start], device_name, device_type, address_offset, data_count)
Описание	Посылает данные в ПЛК. Данные определяются как элементы одномерного массива символов: send_data[start]~ send_data[start + data_count - 1]. *data_count* - число отправляемых символов, оно может быть константой или переменной. *device_name* — имя ПЛК, заключенное в двойные кавычки (“), и оно должно быть определено в списке устройств [system parameters]. *device_type* — тип адреса и метода кодирования (binary или BCD) данных ПЛК. Например, если этот параметр задан LW_BIN — это означает, что используется регистр типа LW и метод кодирования – binary. Если используется метод кодирования BIN, запись “_BIN” может быть проигнорирована. Если адрес типа LW_BCD, это означает что регистр - LW и метод кодирования - BCD. *address_offset* — смещение адресв ПЛК. Например, StringSet(read_data_1[0], “FATEK KB Series”, RT, 5, 1), представляет смещение адреса равное 5. Если *address_offset* использует формат “N#AAAAA”, N показывает, что номер станции ПЛК равен N. «AAAAA» представляет смещение адреса. Например, StringSet(read_data_1[0], “FATEK KB Series”, RT, 2#5, 1) показывает, что номер стойки ПЛК равен 2. Если функция StringSet() использует номер станции по умолчанию, определенный в списке устройств, как показано ниже, то не нужно определять address_offset. Число регистров действительно посылаемых зависит от значения параметра *data_count*, которые *send_data* размещает в символьном массиве. **Тип *read_data*** | ***data_count*** | **actual number of 16-bit register send** char (8-bit) | 1 | 1 char (8-bit) | 2 | 1 Регистр длиной в 1 WORD (16-bit) равен по размеру 2-м символам ASCII. В соответствии с таблицей, чтение 2-х символов ASCII соответствует актуальному чтению одного 16-тибитового регистра. Символы ASCII хранятся в регистрах типа WORD от младшего байта к старшему. При использовании объекта «ASCII display» для отображения строковых переменных хранящихся в регистрах, значение *data_count* должно быть умножено на 2, чтобы отображать полное содержание строки (string). Например:

```
macro_command main()
char src1[10]="abcde"
StringSet(src1[0], "Local HMI", LW, 0, 5)
end macro_command
```

Объект ASCII display покажет :

Когда *data_count* четное число, которое больше или равно длине стринга, содержимое стринга может быть показано полностью:

```
macro_command main()
char src1[10]="abcde"
StringSet(src1[0], "Local HMI", LW, 0, 6)
end macro_command
```

Пример

```
macro_command main()
char str1[10]=”abcde”
// Send 3 words to LW0~LW2
// Data are being sent until the end of string is reached.
// Even though the value of data_count is larger than the length of string
// , the function will automatically stop.
StringSet(str1[0], "Local HMI", LW, 0, 10)
end macro_command
```

Имя	StringSetEx
Синтаксис	StringSetEx(send_data[start], device_name, device_type, address_offset, data_count)
Описание	Посылает данные в ПЛК и продолжает выполнение следующей команды, даже если нет ответа от устройства. Описание *read_data, device_name, device_type, address_offset и data_count* – те же самые, что и в StringSet.

Пример	macro_command main()
	char str1[20]=”abcde”
	short test=0
	// macro will continue executing test = 1 even if the MODBUS device is
	// not responding
	StringSetEx(str1[0], "MODBUS RTU", 4x, 0, 20)

	test = 1 // macro will not continue executing test = 2 until MODBUS device responds StringSet(str1[0], "MODBUS RTU", 4x, 0, 20) test = 2 end macro_command

Имя	StringCopy
Синтаксис	success = StringCopy (“source”, destination[start]) или success = StringCopy (source[start], destination[start])
Описание	Копирует один стринг в другой. Эта функция копирует статический стринг (который заключен в кавычки) или стринг, который хранится в массиве в буфер назначения. Параметр *source* принимает статические строки (в форме “source”) и символьный массив (в форме source[start]). *destination*[start] – должен быть одномерным символьным массивом. Эта функция возвращает булевую переменную, покзывающую, что процесс успешно завершен (true) или нет (false). Если длина строки-источника превышает максимальный размер буфера назначения, возвращается *false* и содержимое буфера назначения остается без изменений. Поле *success* не обязательное.

Пример	macro_command main()
	char src1[5]="abcde"
	char dest1[5]
	bool success1
	success1 = StringCopy(src1[0], dest1[0])
	// success1=true, dest1=”abcde”
	char dest2[5]
	bool success2
	success2 = StringCopy("12345", dest2[0])
	// success2=true, dest2=”12345”
	char src3[10]="abcdefghij"
	char dest3[5]
	bool success3
	success3 = StringCopy(src3[0], dest3[0])
	// success3=false, dest3 remains the same.
	char src4[10]="abcdefghij"
	char dest4[5]
	bool success4

	success4 = StringCopy(src4[5], dest4[0]) // success4=true, dest4=”fghij” end macro_command

Имя	StringDecAsc2Bin
Синтаксис	success = StringDecAsc2Bin(source[start], destination) или success = StringDecAsc2Bin(“source”, destination)
Описание	Эта функция преобразует децимальный стринг в целочисленный. Она преобразует децимальный стринг из параметра *source* в целочисленный и сохраняет в переменной назначения. Параметр *source* принимает статические строки (в форме “source”) и символьный массив (в форме source[start]). *destination* – должен быть переменной; хранит результат преобразования. Эта функция возвращает булевую переменную, покзывающую, что процесс успешно завершен (true) или нет (false). Если строка-источник содержит символы отличающиеся от 0 до 9, возвращается *false.* Поле *success* не обязательное.

Пример	macro_command main()
	char src1[5]="12345"
	int result1
	bool success1
	success1 = StringDecAsc2Bin(src1[0], result1)
	// success1=true, result1 is 12345
	char result2
	bool success2
	success2 = StringDecAsc2Bin("32768", result2)
	// success2=true, but the result exceeds the data range of result2
	char src3[2]="4b"
	char result3
	bool success3
	success3 = StringDecAsc2Bin (src3[0], result3)
	// success3=false, because src3 contains characters other than ‘0’ to ‘9’
	end macro_command

Имя	StringBin2DecAsc
Синтаксис	success = StringBin2DecAsc(source[start], destination)
Описание	Эта функция преобразует целочисленный стринг в децимальный. Она преобразует целочисленный стринг из параметра *source* в децимальный и сохраняет в буфере назначения. *source* – должен быть константой или переменной. *destination*[start] – должен быть одномерным символьным массивом чтобы хранить результат преобразования. *destination* ; храненит результат преобразования. Эта функция возвращает булевую переменную, покзывающую, что процесс успешно завершен (true) или нет (false). Если длина строки-источника превышает максимальный размер буфера назначения, возвращается *false.* Поле *success* не обязательное.

Пример	macro_command main()
	int src1 = 2147483647
	char dest1[20]
	bool success1
	success1 = StringBin2DecAsc(src1, dest1[0])
	// success1=true, dest1=”2147483647”
	short src2 = 0x3c
	char dest2[20]
	bool success2
	success2 = StringBin2DecAsc(src2, dest2[0])
	// success2=true, dest2=”60”
	int src3 = 2147483647
	char dest3[5]
	bool success3
	success3 = StringBin2DecAsc(src3, dest3[0])
	// success3=false, dest3 remains the same.
	end macro_command

Имя	StringDecAsc2Float
Синтаксис	success = StringDecAsc2Float(source[start], destination) или success = StringDecAsc2Float(“source”, destination)
Описание	Эта функция преобразует децимальный стринг в вещественный. Она преобразует децимальный стринг из параметра *source* в вещественный и сохраняет в переменной назначения. Параметр *source* принимает статические строки (в форме “source”) и символьный массив (в форме source[start]). *destination* – должен быть переменной; хранит результат преобразования. Эта функция возвращает булевую переменную, покзывающую, что процесс успешно завершен (true) или нет (false).

	Если строка-источник содержит символы отличающиеся от 0 до 9, возвращается *false.* Поле *success* не обязательное.

Пример	macro_command main()
	char src1[10]="12.345"
	float result1
	bool success1
	success1 = StringDecAsc2Float(src1[0], result1)
	// success1=true, result1 is 12.345
	float result2
	bool success2
	success2 = StringDecAsc2Float("1.234567890", result2)
	// success2=true, but the result exceeds the data range of result2, which
	// might result in loss of precision
	char src3[2]="4b"
	float result3
	bool success3
	success3 = StringDecAsc2Float(src3[0], result3)
	// success3=false, because src3 contains characters other than ‘0’ to ‘9’ or // ‘.’
	end macro_command

Имя	StringFloat2DecAsc
Синтаксис	success = StringFloat2DecAsc(source[start], destination)
Описание	Эта функция преобразует вещественный стринг в децимальный. Она преобразует вещественный стринг из параметра *source* в децимальный и сохраняет в буфере назначения. *Source* - может быть константой или переменной. *destination* – должен быть одномерным массивом символов; хранит результат преобразования. Эта функция возвращает булевую переменную, покзывающую, что процесс успешно завершен (true) или нет (false). Если длина строки-источника превышает максимальный размер буфера назначения, возвращается *false.* Поле *success* не обязательное.
Пример	macro_command main() float src1 = 1.2345 char dest1[20] bool success1 success1 = StringFloat2DecAsc(src1, dest1[0]) // success1=true, dest1=” 1.2345” float src2 = 1.23456789 char dest2 [20]

	bool success2
	success2 = StringFloat2DecAsc(src2, dest2 [0])
	// success2=true, but it might lose precision
	float src3 = 1.2345
	char dest3[5]
	bool success3
	success3 = StringFloat2DecAsc(src3, dest3 [0])
	// success3=false, dest3 remains the same.
	end macro_command

Имя	StringBin2HexAsc
Синтаксис	success = StringBin2HexAsc(source[start], destination)
Описание	Эта функция преобразует двоичные данные в шестнадцатиричный стринг. Она преобразует двоичные данные из параметра *source* в шестнадцатиричный стринг и сохраняет в буфере назначения. *Source* - может быть константой или переменной. *destination* – должен быть одномерным массивом символов; хранит результат преобразования. Эта функция возвращает булевую переменную, покзывающую, что процесс успешно завершен (true) или нет (false). Если длина шестнадцатиричной строки после преобразования превышает максимальный размер буфера назначения, возвращается *false.* Поле *success* не обязательное.

Пример	macro_command main()
	int src1 = 20
	char dest1[20]
	bool success1
	success1 = StringBin2HexAsc(src1, dest1[0])
	// success1=true, dest1=”14”
	short src2 = 0x3c
	char dest2[20]
	bool success2
	success2 = StringBin2HexAsc(src2, dest2[0])
	// success2=true, dest2=”3c”
	int src3 = 0x1a2b3c4d
	char dest3[6]
	bool success3
	success3 = StringBin2HexAsc(src3, dest3[0])
	// success3=false, dest3 remains the same.
	end macro_command

Имя	StringMid
Синтаксис	success = StringMid(source[start], count, destination) или success = StringMid(“string”, start, count, destination)
Описание	Эта функция извлекает последовательность символов с указанным смещением из строки *source* и сохраняет её в буфере назначения. Параметр *source* принимает статические строки (в форме “source”) и символьный массив (в форме source[start]). Для source[start] начальное смещение подстроки указывается индексным значением. Для статического источника строк (“source”), второй параметр (start) указывает начальное смещение подстроки. Параметр *count* – указывает длину извлекаемой подстроки (substring). *destination* – должен быть одномерным символьным массивом, чтобы хранить выделенную подстроку (substring). Эта функция возвращает булевую переменную, покзывающую, что процесс успешно завершен (true) или нет (false). Если длина выделенной подстроки превышает максимальный размер буфера назначения, возвращается *false* и содержимое буфера назначения остается без изменений. Поле *success* не обязательное.

Пример	macro_command main()
	char src1[20]="abcdefghijklmnopqrst"
	char dest1[20]
	bool success1
	success1 = StringMid(src1[5], 6, dest1[0])
	// success1=true, dest1=”fghijk”
	char src2[20]="abcdefghijklmnopqrst"
	char dest2[5]
	bool success2
	success2 = StringMid(src2[5], 6, dest2[0])
	// success2=false, dest2 remains the same.
	char dest3[20]="12345678901234567890"
	bool success3
	success3 = StringMid("abcdefghijklmnopqrst", 5, 5, dest3[15])
	// success3= true, dest3=” 123456789012345fghij”
	end macro_command

Имя	StringLength
Синтаксис	success = StringLength(source[start]) или success = StringLength(“source”)
Описание	Эта функция получает длину строки. Она возвращает длину строки *source* и сохраняет её в поле *Length* на левой стороне оператора ‘=’. Параметр *source* принимает статические строки (в форме “source”) и символьный массив (в форме source[start]). Возвращаемое значение функции отображает длину строки – источника.

Пример	macro_command main()
	char src1[20]="abcde"
	int length1
	length1= StringLength(src1[0])
	// length1=5
	char src2[20]={'a', 'b', 'c', 'd', 'e'}
	int length2
	length2= StringLength(src2[0])
	// length2=20
	char src3[20]="abcdefghij"
	int length3
	length3= StringLength(src3 [2])
	// length3=8
	end macro_command

Имя	StringCat
Синтаксис	success = StringCat(source[start], count, destination) или success = StringCat(“source”, destination[start])
Описание	Эта функция присоединяет строку-источник к строке назначения. Она добавляет содержимое строки *source* к концу содержимого строки *destination.* Параметр *source* принимает статические строки (в форме “source”) и символьный массив (в форме source[start]). *destination* – должен быть одномерным символьным массивом. Эта функция возвращает булевую переменную, покзывающую, что процесс успешно завершен (true) или нет (false). Если длина строки-результата после конкатенации превышает максимальный размер буфера назначения, возвращается *false*. Поле *success* не обязательное.
Пример	macro_command main() char src1[20]="abcdefghij" char dest1[20]="1234567890" bool success1 success1= StringCat(src1[0], dest1[0])

	// success1=true, dest1=”123456790abcdefghij”
	char dest2 [10]="1234567890"
	bool success2
	success2= StringCat("abcde", dest2 [0])
	// success2=false, dest2 remains the same.
	char src3[20]="abcdefghij"
	char dest3[20]
	bool success3
	success3= StringCat(src3[0], dest3[15])
	// success3=false, dest3 remains the same.
	end macro_command

Имя	StringCompare
Синтаксис	ret = StringCompare (str1[start], str2[start]) ret = StringCompare (“string1“, str2[start]) ret = StringCompare (str1[start], “string2”) ret = StringCompare (“string1”, “string2”)
Описание	Эта функция делает сранение двух строку с учетом регистра символов. Оба параметра принимают статические строки (в форме “string1”) и символьный массив (в форме str1[start]). Эта функция возвращает булевую переменную, покзывающую, что процесс успешно завершен (true) или нет (false). Поле *ret* не обязательное.

Пример	macro_command main()
	char a1[20]="abcde"
	char b1[20]="ABCDE"
	bool ret1
	ret1= StringCompare(a1[0], b1[0])
	// ret1=false
	char a2[20]="abcde"
	char b2[20]="abcde"
	bool ret2
	ret2= StringCompare(a2[0], b2[0])
	// ret2=true
	char a3 [20]="abcde"
	char b3[20]="abcdefg"
	bool ret3
	ret3= StringCompare(a3[0], b3[0])
	// ret3=false
	end macro_command

Имя	StringCompareNoCase
Синтаксис	ret = StringCompareNoCase (str1[start], str2[start]) ret = StringCompareNoCase (“string1“, str2[start]) ret = StringCompareNoCase (str1[start], “string2”) ret = StringCompareNoCase (“string1”, “string2”)
Описание	Эта функция делает сранение двух строку без учета регистра символов. Оба параметра принимают статические строки (в форме “string1”) и символьный массив (в форме str1[start]). Эта функция возвращает булевую переменную, покзывающую, что процесс успешно завершен (true) или нет (false). Поле *ret* не обязательное.

Пример	macro_command main()
	char a1[20]="abcde"
	char b1[20]="ABCDE"
	bool ret1
	ret1= StringCompareNoCase(a1[0], b1[0])
	// ret1=true
	char a2[20]="abcde"
	char b2[20]="abcde"
	bool ret2
	ret2= StringCompareNoCase(a2[0], b2[0])
	// ret2=true
	char a3 [20]="abcde"
	char b3[20]="abcdefg"
	bool ret3
	ret3= StringCompareNoCase(a3[0], b3[0])
	// ret3=false
	end macro_command

Имя	StringFind
Синтаксис	position = StringFind (source[start], target[start]) position = StringFind (“source“, target[start]) position = StringFind (source[start], “target”) position = StringFind (“source“, “target“)
Описание	Эта функция возвращает позицию первого обнаружения целевой строки (target) в строке-источнике. Оба параметра принимают статические строки (в форме “source”) и символьный массив (в форме source[start]). Эта функция возвращает индекс начинающийся с «0» первого символа подстроки в строке-источнике, котора совпадает с целевой строкой. Если совпадения нет, функция возвращает «-1».
Пример	macro_command main() char src1[20]="abcde" char target1[20]="cd"

	bool pos1
	pos1= StringFind(src1[0], target1[0])
	// pos1=2
	char target2[20]="ce"
	bool pos2
	pos2= StringFind("abcde", target2[0])
	// pos2=-1
	char src3[20]="abcde"
	bool pos3
	pos3= StringFind(src3[3], "cd")
	// pos3=-1
	end macro_command

Имя	StringReverseFind
Синтаксис	position = StringReverseFind (source[start], target[start]) position = StringReverseFind (“source“, target[start]) position = StringReverseFind (source[start], “target”) position = StringReverseFind (“source“, “target“)
Описание	Эта функция возвращает позицию последнего обнаружения целевой строки (target) в строке-источнике. Оба параметра принимают статические строки (в форме “source”) и символьный массив (в форме source[start]). Эта функция возвращает индекс начинающийся с «0» первого символа подстроки в строке-источнике, которая совпадает с целевой строкой. Заметьте, что полная последовательность символов должна совпадать. Если есть несколько совпадающих подстрок, функция вернет позицию последней совпадающей подстроки. Если совпадения нет, функция возвращает «-1».

Пример	macro_command main()
	char src1[20]="abcdeabcde"
	char target1[20]="cd"
	bool pos1
	pos1= StringReverseFind(src1[0], target1[0])
	// pos1=7
	char target2[20]="ce"
	bool pos2
	pos2= StringReverseFind("abcdeabcde", target2[0])
	// pos2=-1
	char src3[20]="abcdeabcde"
	bool pos3
	pos3= StringReverseFind(src3[6], "ab")
	// pos3=-1
	end macro_command

Имя	StringFindOneOf
Синтаксис	position = StringFindOneOf (source[start], target[start]) position = StringFindOneOf (“source“, target[start]) position = StringFindOneOf (source[start], “target”) position = StringFindOneOf(“source“, “target“)
Описание	Эта функция возвращает позицию первого символа в строке-источнике, которй совпадает с любым символом, содержащимся в целевой строке (target). Оба параметра принимают статические строки (в форме “source”) и символьный массив (в форме source[start]). Эта функция возвращает индекс начинающийся с «0» первого символа в строке-источнике, который также присутствует в целевой строке. Если совпадения нет, функция возвращает «-1».

Пример	macro_command main()
	char src1[20]="abcdeabcde"
	char target1[20]="sdf"
	bool pos1
	pos1= StringFindOneOf(src1[0], target1[0])
	// pos1=3
	char src2[20]="abcdeabcde"
	bool pos2
	pos2= StringFindOneOf(src2[1], "agi")
	// pos2=4
	char target3 [20]="bus"
	bool pos3
	pos3= StringFindOneOf("abcdeabcde", target3[1])
	// pos3=-1
	end macro_command

Имя	StringIncluding
Синтаксис	success = StringIncluding (source[start], set[start], destination[start]) success = StringIncluding (“source“, set[start], destination[start]) success = StringIncluding (source[start], “set“, destination[start]) success = StringIncluding (“source“, “set“, destination[start])
Описание	Эта функция извлекает подстроку (substring) из строки *source,* которая содержит символы в наборе строк (set), начиная с первого символа строке-источнике и кончая, когда найден символ в строке-источнике, который отсутствует в целевой строке. Параметры *source и set* принимают статические строки (в форме “source”) и символьный массив (в форме source[start]). Эта функция возвращает булевую переменную, покзывающую, что процесс успешно завершен (true) или нет (false). Если длина извлеченной подстроки превышает максимальный размер буфера назначения, возвращается *false*..
Пример	macro_command main() char src1[20]="cabbageabc"

	char set1[20]="abc"
	char dest1[20]
	bool success1
	success1 = StringIncluding(src1[0], set1[0], dest1[0])
	// success1=true, dest1="cabba"
	char src2[20]="gecabba"
	char dest2[20]
	bool success2
	success2 = StringIncluding(src2[0], "abc", dest2[0])
	// success2=true, dest2=""
	char set3[20]="abc"
	char dest3[4]
	bool success3
	success3 = StringIncluding("cabbage", set3[0], dest3[0])
	// success3=false, dest3 remains the same.
	end macro_command

Имя	StringExcluding
Синтаксис	success = StringExcluding (source[start], set[start], destination[start]) success = StringExcluding (“source“, set[start], destination[start]) success = StringExcluding (source[start], “set“, destination[start]) success = StringExcluding (“source“, “set“, destination[start])
Описание	Эта функция извлекает подстроку (substring) из строки *source,* которая содержит символы отсутствующих в наборе строк (set), начиная с первого символа строке-источнике и кончая, когда найден символ в строке-источнике, который есть в целевой строке. Параметры *source и set* принимают статические строки (в форме “source”) и символьный массив (в форме source[start]). Эта функция возвращает булевую переменную, покзывающую, что процесс успешно завершен (true) или нет (false). Если длина извлеченной подстроки превышает максимальный размер буфера назначения, возвращается *false*..
Пример	macro_command main() char src1[20]="cabbageabc" char set1[20]="ge" char dest1[20] bool success1 success1 = StringExcluding(src1[0], set1[0], dest1[0]) // success1=true, dest1="cabba" char src2[20]="cabbage" char dest2[20] bool success2 success2 = StringExcluding(src2[0], "abc", dest2[0])

	// success2=true, dest2=""
	char set3[20]="ge"
	char dest3[4]
	bool success3
	success3 = StringExcluding("cabbage", set3[0], dest3[0])
	// success3=false, dest3 remains the same.
	end macro_command

Имя	StringToUpper
Синтаксис	success = StringToUpper (source[start], set[start], destination[start]) success = StringToUpper (“source“, set[start], destination[start])
Описание	Преобразует все символы строки *source* в символы верхнего регистра (uppercase) и сохраняет результат в буфере назначения *destination.* Параметр *source* принимает статические строки (в форме “source”) и символьный массив (в форме source[start]). Эта функция возвращает булевую переменную, покзывающую, что процесс успешно завершен (true) или нет (false). Если длина строки после преобразования превышает максимальный размер буфера назначения, возвращается *false*..

Пример	macro_command main()
	char src1[20]="aBcDe"
	char dest1[20]
	bool success1
	success1 = StringToUpper(src1[0], dest1[0])
	// success1=true, dest1="ABCDE"
	char dest2[4]
	bool success2
	success2 = StringToUpper("aBcDe", dest2[0])
	// success2=false, dest2 remains the same.
	end macro_command

Имя	StringToLower
Синтаксис	success = StringToLower (source[start], set[start], destination[start]) success = StringToLower (“source“, set[start], destination[start])
Описание	Преобразует все символы строки *source* в символы нижнего регистра (Lowercase) и сохраняет результат в буфере назначения *destination.* Параметр *source* принимает статические строки (в форме “source”) и символьный массив (в форме source[start]). Эта функция возвращает булевую переменную, покзывающую, что процесс успешно завершен (true) или нет (false). Если длина строки после преобразования превышает максимальный размер буфера назначения, возвращается *false*..
Пример	macro_command main() char src1[20]="aBcDe"

	char dest1[20]
	bool success1
	success1 = StringToUpper(src1[0], dest1[0])
	// success1=true, dest1="abcde"
	char dest2[4]
	bool success2
	success2 = StringToUpper("aBcDe", dest2[0])
	// success2=false, dest2 remains the same.
	end macro_command

Имя	StringToReverse
Синтаксис	success = StringToReverse (source[start], set[start], destination[start]) success = StringToReverse (“source“, set[start], destination[start])
Описание	Реверсирует все символы строки *source* и сохраняет результат в буфере назначения *destination.* Параметр *source* принимает статические строки (в форме “source”) и символьный массив (в форме source[start]). Эта функция возвращает булевую переменную, покзывающую, что процесс успешно завершен (true) или нет (false). Если длина строки после реверсирования превышает максимальный размер буфера назначения, возвращается *false*..

Пример	macro_command main()
	char src1[20]="abcde"
	char dest1[20]
	bool success1
	success1 = StringToUpper(src1[0], dest1[0])
	// success1=true, dest1="edcba"
	char dest2[4]
	bool success2
	success2 = StringToUpper("abcde", dest2[0])
	// success2=false, dest2 remains the same.
	end macro_command

Имя	StringTrimLeft
Синтаксис	success = StringTrimLeft (source[start], set[start], destination[start]) success = StringTrimLeft (“source“, set[start], destination[start]) success = StringTrimLeft (source[start], “set“, destination[start]) success = StringTrimLeft (“source“, “set“, destination[start])
Описание	Обрезает ведущие символы в наборе буфера из строки *source.* Параметр *source* принимает статические строки (в форме “source”) и символьный массив (в форме source[start]). Эта функция возвращает булевую переменную, покзывающую, что процесс успешно завершен (true) или нет (false). Если длина обрезанной строки превышает максимальный размер буфера назначения, возвращается *false*..
Пример	macro_command main()

	char src1[20]= "# *a*#bc"
	char set1[20]="# *"
	char dest1[20]
	bool success1
	success1 = StringTrimLeft (src1[0], set1[0], dest1[0])
	// success1=true, dest1="a*#bc"
	char set2[20]={'#', ' ', '*'}
	char dest2[4]
	success2 = StringTrimLeft ("# *a*#bc", set2[0], dest2[0])
	// success2=false, dest2 remains the same.
	char src3[20]="abc *#"
	char dest3[20]
	bool success3
	success3 = StringTrimLeft (src3[0], "# *", dest3[0])
	// success3=true, dest3="abc *#"
	end macro_command

Имя	StringTrimRight
Синтаксис	success = StringTrimRight (source[start], set[start], destination[start]) success = StringTrimRight (“source“, set[start], destination[start])
Описание	Обрезает замыкающие символы в наборе буфера из строки *source.* Параметр *source* принимает статические строки (в форме “source”) и символьный массив (в форме source[start]). Эта функция возвращает булевую переменную, покзывающую, что процесс успешно завершен (true) или нет (false). Если длина обрезанной строки превышает максимальный размер буфера назначения, возвращается *false*..

Пример	macro_command main()
	char src1[20]= "# *a*#bc# * "
	char set1[20]="# *"
	char dest1[20]
	bool success1
	success1 = StringTrimRight(src1[0], set1[0], dest1[0])
	// success1=true, dest1="# *a*#bc"
	char set2[20]={'#', ' ', '*'}
	char dest2[20]
	success2 = StringTrimRight("# *a*#bc", set2[0], dest2[0])
	// success2=true, dest2="# *a*#bc"
	char src3[20]="ab**c *#"
	char dest3[4]
	bool success3
	success3 = StringTrimRight(src3[0], "# *", dest3[0])

	// success3=false, dest3 remains the same. end macro_command

Имя	StringInsert
Синтаксис	success = StringInsert (pos, insert[start], destination[start]) success = StringInsert (pos, “insert“, destination[start]) success = StringInsert (pos, insert[start], length, destination[start]) success = StringInsert (pos, “insert“, length, destination[start])
Описание	Вставляет строку в указанное место в содержимом строки назначения – *destination.* Место вставки указывается параметром *pos* - позиция. Параметр *insert* принимает статические строки (в форме “source”) и символьный массив (в форме source[start]). Число символов для вставки может быть задано параметром *length* (длина). Эта функция возвращает булевую переменную, покзывающую, что процесс успешно завершен (true) или нет (false). Если длина строки после вставки превышает максимальный размер буфера назначения, возвращается *false*..

Пример	macro_command main()
	char str1[20]="but the question is"
	char str2[10]=", that is"
	char dest[40]="to be or not to be"
	bool success
	success = StringInsert(18, str1[3], 13, dest[0])
	// success=true, dest="to be or not to be the question"
	success = StringInsert(18, str2[0], dest[0])
	// success=true, dest="to be or not to be, that is the question"
	success = StringInsert(0, "Hamlet:", dest[0])
	// success=false, dest remains the same.
	
	end macro_command

18.6.7 Функции запроса рецепта

Имя	RecipeGetData
Синтаксис	RecipeGetData (destination, recipe_address, record_ID)
Описание	Получает данные рецепта (Recipe Data). Полученные данные будут сохранены в *destination,* и должны быть переменными. Адрес рецепта состоит из имени рецепта и имени элемента “recipe_name.item_name”. Параметр *record_ID* указывает ID номер записи в рецепте, из которого извлекаются данные.

Пример	macro_command main()
	int data=0
	char str[20]
	int recordID
	bool result
	recordID = 0
	result = RecipeGetData(data, "TypeA.item_weight", recordID)
	// From recipe "TypeA" get the data of the item “item_weight” in record 0.
	recordID = 1
	result = RecipeGetData(str[0], "TypeB.item_name", recordID)
	// From recipe "TypeB" get the data of the item “item_name” in record 1.
	end macro_command

Имя	RecipeQuery
Синтаксис	RecipeQuery (SQL command, destination)
Описание	Использует команду SQL для запроса данных рецепта. Число записей полученных в результате запроса будут сохранены в *destination.* Это должны быть переменные. Команда SQL может использовать статические стринги или массивы символов. Например: RecipeQuery(“SELECT * FROM TypeA”, destination) или RecipeQuery(sql[0], destination) Команда SQL должна стартовать с ”SELECT * FROM” и последующего *recipe name* и *query condition* (условия запроса)

Пример	macro_command main()
	int total_row=0
	char sql[100]="SELECT * FROM TypeB"
	bool result
	result = RecipeQuery("SELECT * FROM TypeA", total_row)
	// Query Recipe "TypeA". Store the number of records of query result in total_row.
	result = RecipeQuery(sql[0], total_row) // Query Recipe "TypeB". Store the number of records of query result in total_row.
	end macro_command

Имя	RecipeQueryGetData
Синтаксис	RecipeQueryGetData (destination, recipe_address, result_row_no)
Описание	Получает данные в результате запроса полученного RecipeQuery. Эта функция должна вызываться после вызова RecipeQuery и указывать то же самое имя рецепта в адресе рецепта, что и в RecipeQuery. Параметр *result_row_no* указывает последовательный номер ряда в результате запроса.

Пример	macro_command main()
	int data=0
	int total_row=0
	int row_number=0
	bool result_query
	bool result_data
	result_query = RecipeQuery("SELECT * FROM TypeA", total_row)
	// Query Recipe "TypeA". Store the number of records of query result in total_row.
	if (result_query) then
	  for row_number=0 to total_row-1
	  result_data = RecipeQueryGetData(data, "TypeA.item_weight",
	  row_number)
	  next row_number end if
	  end macro_command

Имя	RecipeQueryGetRecordID
Синтаксис	RecipeQueryGetRecordID (destination, result_row_no)
Описание	Получает номера записей (Record ID) в результате запроса полученного чьи записи определены в RecipeQuery. Эта функция должна вызываться после вызова RecipeQuery. Параметр *result_row_no* указывает последовательный номер ряда в результате запроса и пишет полученные записи в *destination* .

Пример	macro_command main()
	int recordID=0
	int total_row=0
	int row_number=0
	bool result_query
	bool result_id
	result_query = RecipeQuery("SELECT * FROM TypeA", total_row)
	// Query Recipe "TypeA". Store the number of records of query result in total_row.
	if (result_query) then
	  for row_number=0 to total_row-1
	  result_id = RecipeQueryGetRecordID(recordID, row_number)
	  next row_number

	end if end macro_command

18.6.8 Различные функции запроса

Имя	SYNC_TRIG_MACRO
Синтаксис	SYNC_TRIG_MACRO(macro_id)
Описание	Запускает (Trigger) выполнение макроса синхронно (используйте *macro_id* , чтобы обозначить этот макрос) в работающем макросе. Текущий макрос будет переведен в «Паузу» до конца выполнения этого вызванного макроса. *macro_id* может быть константой или переменной.

Пример	macro_command main()
	char ON = 1, OFF = 0
	SetData(ON, “Local HMI”, LB, 0, 1)
	SYNC_TRIG_MACRO(5)// call a macro (its ID is 5)
	SetData(OFF, “Local HMI”, LB, 0, 1)
	end macro_command

Имя	ASYNC_TRIG_MACRO
Синтаксис	ASYNC_TRIG_MACRO(macro_id)
Описание	Запускает (Trigger) выполнение макроса асинхронно (используйте *macro_id* , чтобы обозначить этот макрос) в работающем макросе. Текущий макрос будет продолжать выполнение следующих команд после запуска указанного макроса. Другими словами, два макроса будут активны одновременно. *macro_id* может быть константой или переменной.

Пример	macro_command main()
	char ON = 1, OFF = 0
	SetData(ON, “Local HMI”, LB, 0, 1)
	ASYNC_TRIG_MACRO(5)// call a macro (its ID is 5)
	SetData(OFF, “Local HMI”, LB, 0, 1)
	end macro_command

Имя	TRACE
Синтаксис	TRACE(format, argument)
Описание	Эту функцию используют, чтобы послать указанный стринг в утилиту - *EasyDiagnoser*. Пользователи могут распечатать текущие значения переменных при работе макроса в режиме работы для отладки. Когда TRACE встречает первый формат спецификации (если их несколько), происходит преобразование значения певого аргумента после формата и и выводит их соответственно. *format* – указывает формат управляющий выходящим стрингом.

	Спецификация формата, которая состоит из опциональных (в скобках [ ]) и требуемых полей (выделеных жирным шрифтом), имеет следующую форму: **%**[flags] [width] [.precision] **type** Каждое поле описано ниже: *flags* (не обязательное): - + *width* (не обязательное): Не отрицательное десятичное целое число, которое управляет минимальным числом печатаемых символов. *precision* (не обязательное): Не отрицательное десятичное целое число, которое указывает точность и число символов для печати. *type:* C или c : указывает однобайтный символ. d : десятичное целое число со знаком. i : десятичное целое число со знаком. o : октальное целое число без знака. u : десятичное целое число без знака. X или x : шестнадцатиричное целое число без знака. E или e : Знаковое значение имеющее форму [–]*d***.***dddd* **e** [*sign*]*ddd,* где *d* –одна десятичная цифра, *dddd* –одна или больше десятичных цифр, *ddd* –точно три десятичных цифры, и *sign* - «+» или «–». f : Знаковое значение имеющее форму[–]*dddd***.***dddd*, где *dddd* –одна или больше десятичных цифр. Длина стринга на выходе ограничена 256 символами. Лишние символы будут проигнорированы. Часть *argument* part - не обязательная. Один формат спецификации преобразует точно один аргумент.

Пример	macro_command main()
	char c1 = ’a’
	short s1 = 32767
	float f1 = 1.234567
	TRACE(“The results are”) // output: The results are
	TRACE(“c1 = %c, s1 = %d, f1 = %f”, c1, s1, f1)
	// output: c1 = a, s1 = 32767, f1 = 1.234567
	end macro_command

Имя	FindDataSamplingDate
Синтаксис	return_value = FindDataSamplingDate (data_log_number, index, year, month, day) или FindDataSamplingDate (data_log_number, index, year, month, day)
Описание	Это функция запроса на поиск данных в указанном файле опроса данных (data sampling file) в соответствии с номером «data sampling no.» и индексом файла. Дата хранится в “year”, “month” и “day” соответственно в формате YYYY, MM и DD. Data Sampling Object No. | Description | Read address | Sample mode | Trigger address | Clear address | Hold address | Auto. stop 1 | Local HMI : LW0 | Periodical | Disable | Disable | Disable | Disable 2 | Local HMI : LW0 | Periodical | Disable | Disable | Disable | Disable «data sampling no.» Директория сохраняемых данных: [Storage location]\[filename]\yyyymmdd.dtl. Файлы «data sampling» в одной директории отсортированы в соответствии с именем файла и проиндексированы, начиная с 0. Самый последний сохраненный файл имеет наименьший индексный номер. Например, если есть четыре файла опросов: 20101210.dtl 20101230.dtl 20110110.dtl 20110111.dtl Индексы файлов: 20101210.dtl -> индекс - 3 20101230.dtl -> индекс - 2 20110110.dtl -> индекс - 1 20110111.dtl -> индекс - 0 “*return_value*” равно 1, если искомый файл данных успешно найден, иначе оно равно 0. “*data_log_number*” и “*index*” могут быть константами и переменными. “*year*”, “*month*”, “*day*” и “*return_value*” должны быть переменными. Поле “return_value” не обязательное.

Пример	macro_command main()
	short data_log_number = 1, index = 2, year, month, day
	short success
	// if there exists a data sampling file named 20101230.dtl, with data sampling //
	number 1 and file index 2.
	// the result after execution: success == 1, year == 2010, month == 12 and//day
	== 30
	success = FindDataSamplingDate(data_log_number, index, year, month, day)
	end macro_command

Имя	FindDataSamplingIndex
Синтаксис	return_value = FindDataSamplingIndex (data_log_number, year, month, day, index) или FindDataSamplingIndex (data_log_number, year, month, day, index)
Описание	Это функция запроса на поиск индекса файла в указанном файле опроса данных (data sampling file) в соответствии с номером «data sampling no.» и Датой. Индекс файла хранится в “index”, “year”, “month” и “day” соответственно в формате YYYY, MM и DD. Data Sampling Object No. | Description | Read address | Sample mode | Trigger address | Clear address | Hold address | Auto. stop 1 | | Local HMI : LW0 | Periodical | Disable | Disable | Disable | Disable 2 | | Local HMI : LW0 | Periodical | Disable | Disable | Disable | Disable Директория сохраняемых данных: [Storage location]\[filename]\yyyymmdd.dtl. Файлы «data sampling» в одной директории отсортированы в соответствии с именем файла и проиндексированы, начиная с 0. Самый последний сохраненный файл имеет наименьший индексный номер. Например, если есть четыре файла опросов: 20101210.dtl 20101230.dtl 20110110.dtl 20110111.dtl Индексы файлов будут: 20101210.dtl -> индекс - 3 20101230.dtl -> индекс - 2 20110110.dtl -> индекс - 1 20110111.dtl -> индекс - 0 “*return_value*” равно 1, если искомый файл данных успешно найден, иначе оно равно 0. “*data_log_number*” и “*index*” могут быть константами и переменными. “*year*”, “*month*”, “*day*” и “*return_value*” должны быть переменными. Поле “return_value” не обязательное.

Пример	macro_command main()
	short data_log_number = 1, year = 2010, month = 12, day = 10, index
	short success
	// if there exists a data sampling file named 20101210.dtl, with data sampling //
	number 1 and file index 2.
	// the result after execution: success == 1 and index == 2
	success = FindDataSamplingIndex (data_log_number, year, month, day, index)
	end macro_command

Имя	FindEventLogDate
Синтаксис	return_value = FindEventLogDate (index, year, month, day) или FindEventLogDate (index, year, month, day)
Описание	Это функция запроса на поиск даннвх в указанном файле архива событий (event log file) в соответствии с индексом файла. Индекс файла хранится в “index”, “year”, “month” и “day” соответственно в формате YYYY, MM и DD. Файлы архивов событий хранящиеся в одном месте (в памяти панели или на внешнем устройстве) отсортированы в соответствии с именем файла и проиндексированы, начиная с 0. Самый последний сохраненный файл имеет наименьший индексный номер. Например, если есть четыре файла архивов событий: EL_20101210.evt EL_20101230.evt EL_20110110.evt EL_20110111.evt Индексы файлов будут: EL_20101210.evt -> индекс - 3 EL_20101230.evt -> индекс - 2 EL_20110110.evt -> индекс - 1 EL_20110111.evt -> индекс - 0 “*return_value*” равно 1, если искомый файл данных успешно найден, иначе оно равно 0. “*data_log_number*” и “*index*” могут быть константами и переменными. “*year*”, “*month*”, “*day*” и “*return_value*” должны быть переменными. Поле “return_value” не обязательное.

Пример	macro_command main()
	short index = 1, year, month, day
	short success
	// if there exists an event log file named EL_20101230.evt , with index 1
	// the result after execution: success == 1, year == 2010, month == 12, day //== 30
	success = FindEventLogDate (index, year, month, day)
	end macro_command

Имя	FindEventLogIndex
Синтаксис	return_value = FindEventLogIndex (year, month, day, index) или FindEventLogIndex (year, month, day, index)
Описание	Это функция запроса на поиск индекса файла в указанном файле архива событий данных (event log file) в соответствии с датой. Индекс файла хранится в “index”, “year”, “month” и “day” соответственно в формате YYYY, MM и DD. Файлы архивов событий хранящиеся в одном месте (в памяти панели или на внешнем устройстве) отсортированы в соответствии с именем файла и проиндексированы, начиная с 0. Самый последний сохраненный файл имеет наименьший индексный номер. Например, если есть четыре файла архивов событий: EL_20101210.evt EL_20101230.evt EL_20110110.evt EL_20110111.evt Индексы файлов будут: EL_20101210.evt -> индекс - 3 EL_20101230.evt -> индекс - 2 EL_20110110.evt -> индекс - 1 EL_20110111.evt -> индекс - 0 “*return_value*” равно 1, если искомый файл успешно найден, иначе оно равно 0. “*data_log_number*” и “*index*” могут быть константами и переменными. “*year*”, “*month*”, “*day*” и “*return_value*” должны быть переменными. Поле “return_value” не обязательное.

Пример	macro_command main()
	short year = 2010, month = 12, day = 10, index
	short success
	// if there exists an event log file named EL_20101210.evt, with index 2
	// the result after execution: success == 1, index == 2
	success = FindEventLogIndex (year, month, day, index)
	end macro_command

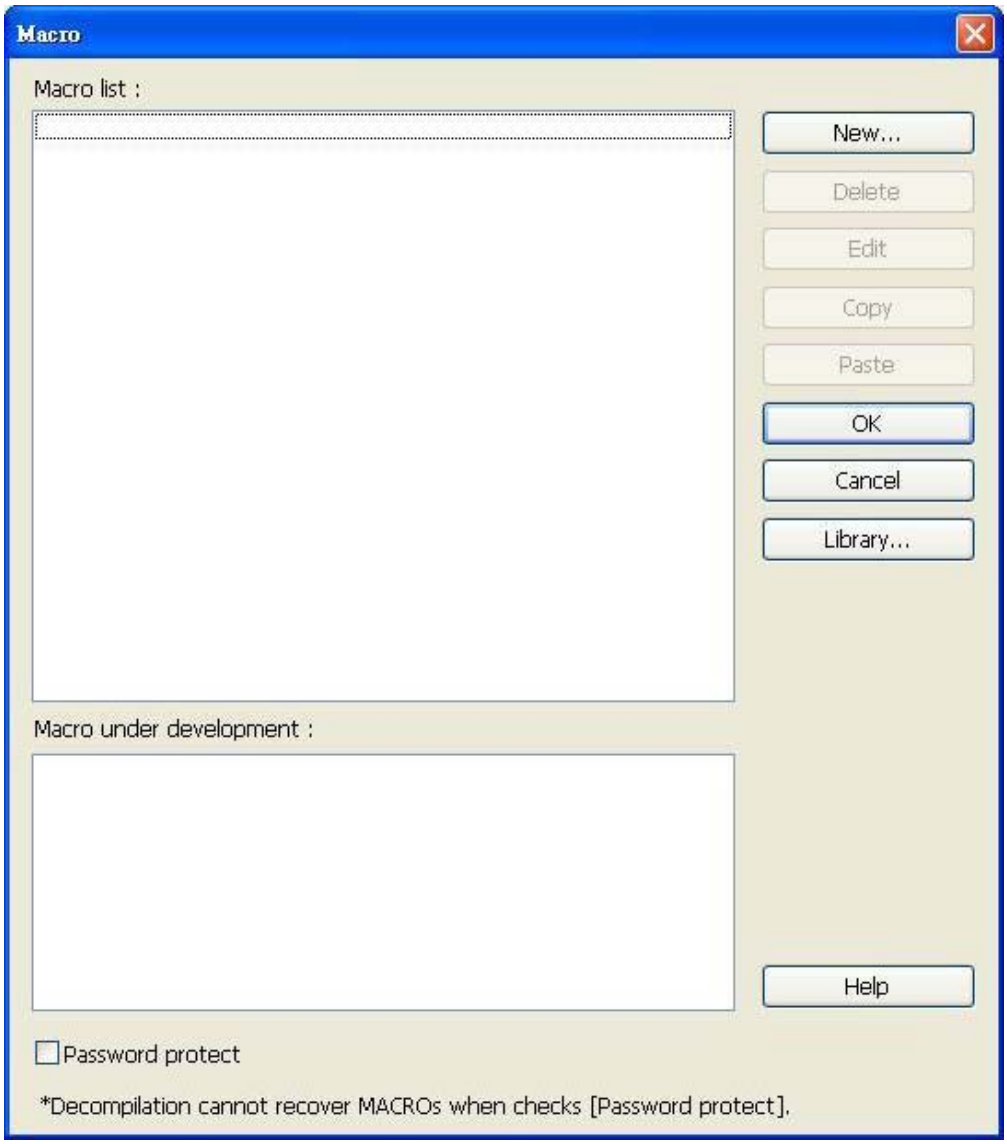
18.7 Как создать и выполнить макрос

18.7.1 Как создать макрос

Программирование макроса может быть разделено на следующие этапы:

Шаг 1:

Щелкните на пиктограмме “Macro Manager” на панели инструментов EasyBuilderPro для открытия Macro Manager, как показано ниже.



В окне [Macro Manager] все успешно откомпилированные макросы отображаются в списке “Macro list”, а все редактируемые макросы — в списке ‘Macro under development”. Ниже приведено назначение различных кнопок.

[New]
Открывает редактор “WorkSpace” создания нового макроса.

[Delete]
Удаляет выделенные макросы.

[Edit]
Открывает редактор “WorkSpace” и загружает выделенный макрос.

[Copy]
Копирование выбранного макроса в буфер обмена.

[Paste]
Вставка макроса из буфера обмена в список, назначение нового имени макросу.

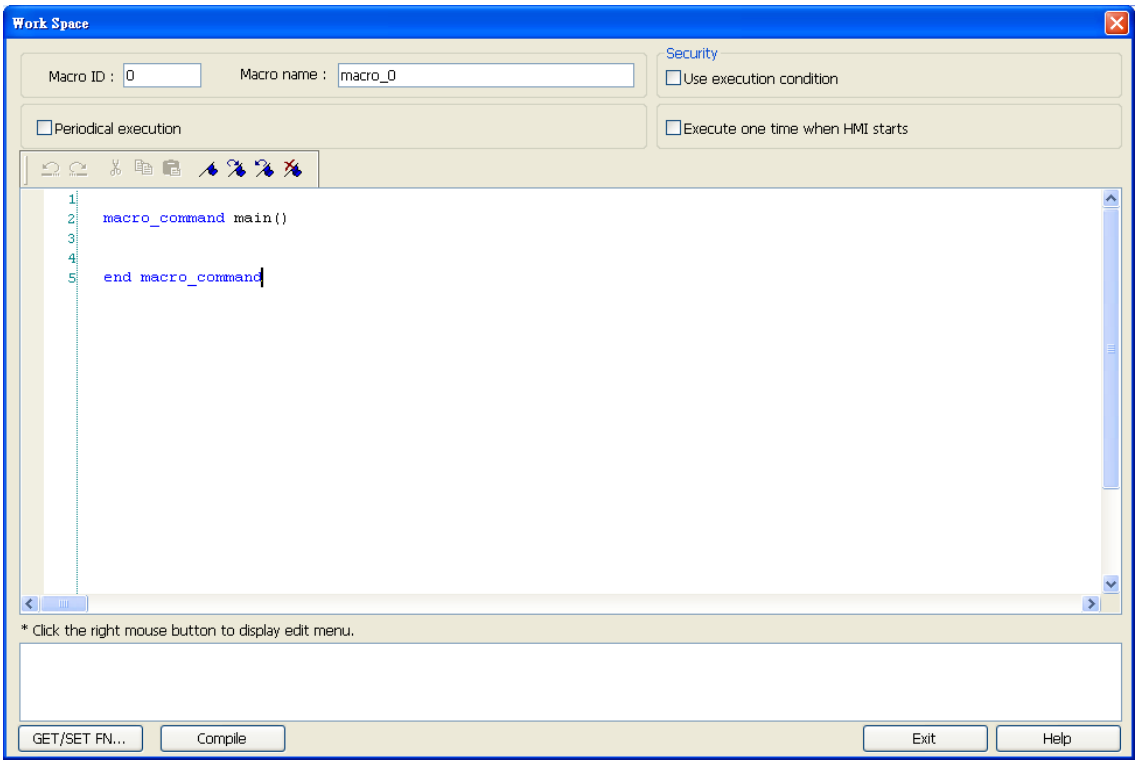
[OK]
Подтверждение всех редактированных макросов и сохранение нового содержимого перед выходом из диалога.

[Cancel]
Отказ и выход из диалога редактора *Macro*.

[Library…]
Открывает диалог работы с *Macro Funtion Library* .

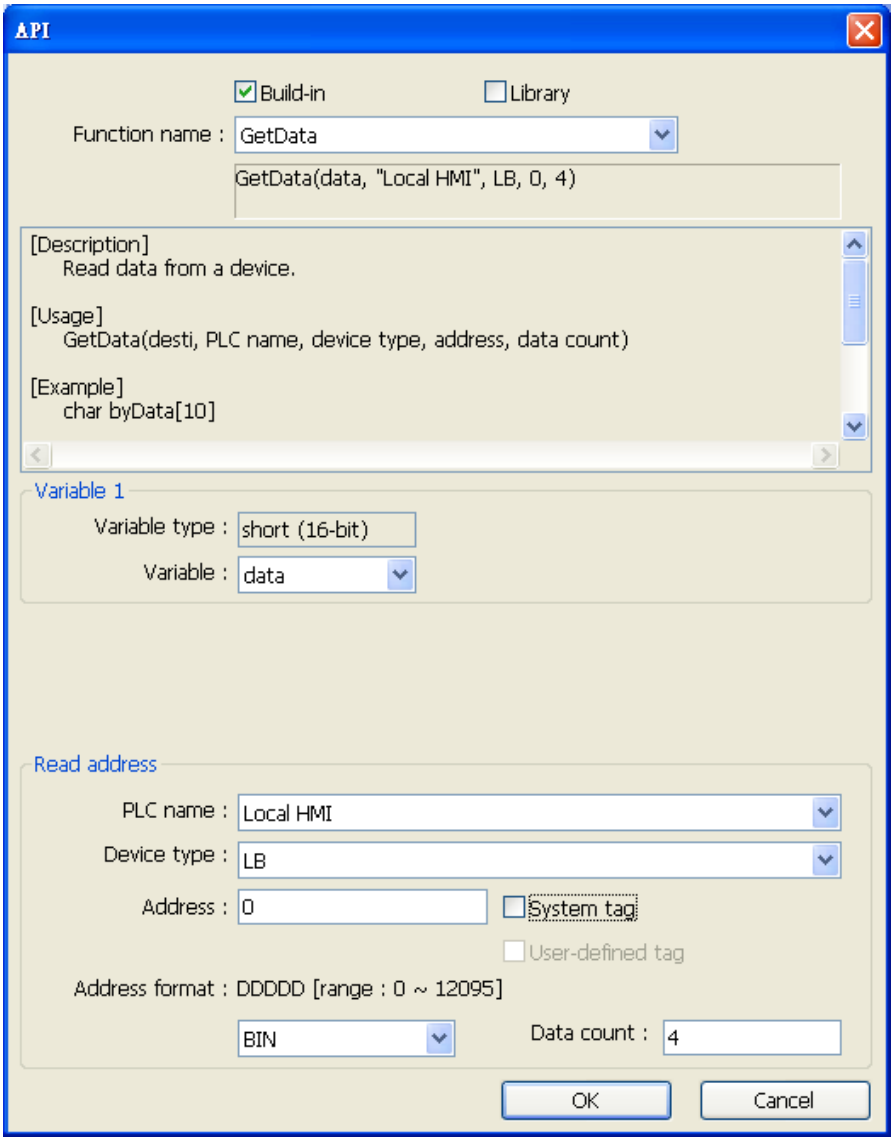
Шаг 2:

Нажмите кнопку “New” для открытия редактора “WorkSpace”. Каждый макрос имеет уникальный номер, определенный в поле “Macro ID”, имя макроса также должно быть задано, в противном случае при компиляции возникнет ошибка.



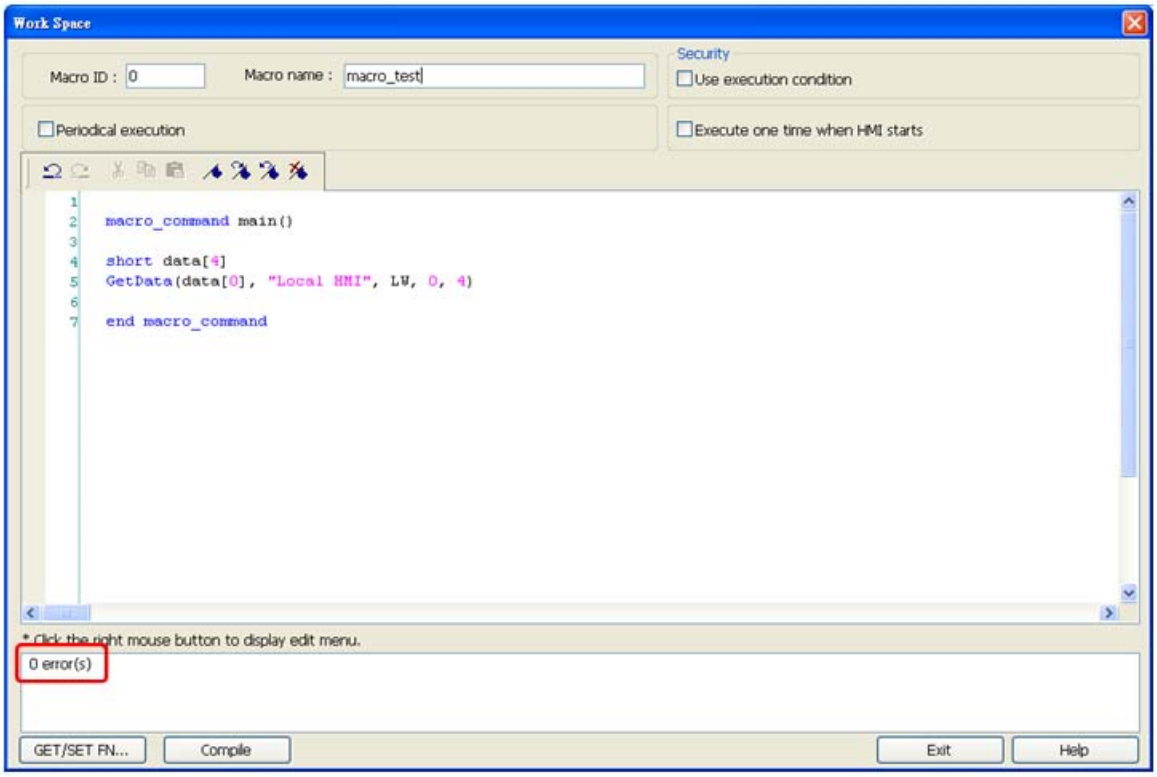
Шаг 3:

Создайте свой макрос. При необходимости используйте встроенные функции (такие как SetData() или Getdata()), нажмите кнопку ‘Get/Set FN…” для открытия диалогового окна API и выберите функцию и задайте необходимые параметры.

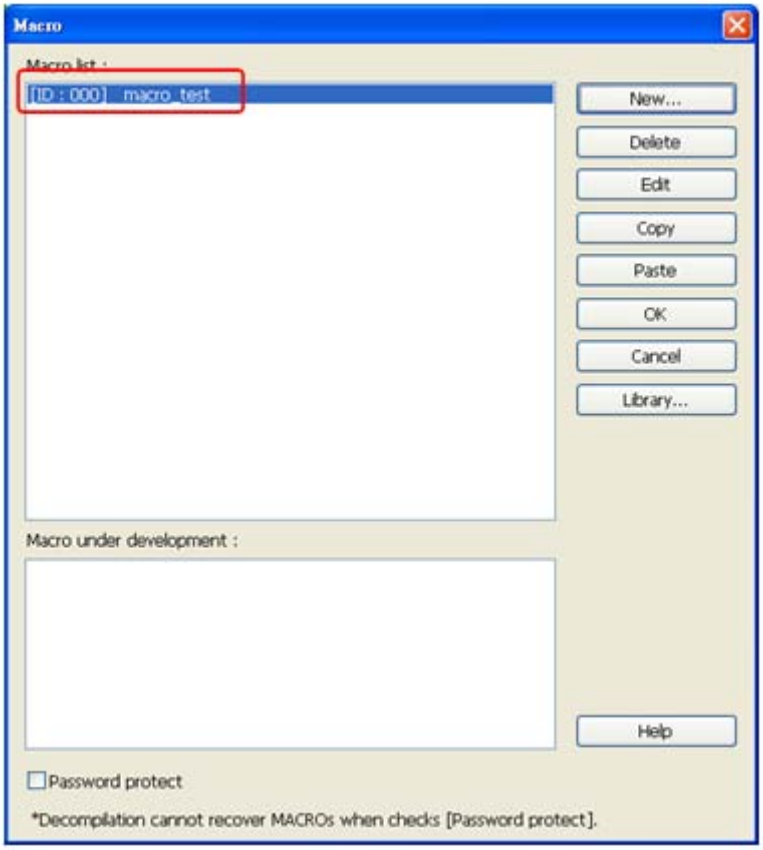


Шаг 4:

После создания нового макроса, нажмите кнопку ‘Compile” для компиляции макроса.



Если нет ошибок, нажмите кнопку “exit”, новый макрос “macro_test” появится в списке “Macro list”.



18.7.2 Выполнение макроса

Есть несколько способов выполнения макроса.

a. С использование объекта [PLC control]

1. Откройте объект [PLC control] и установите атрибут “Execute macro program”.
2. Выберите имя макроса. Выберите бит и укажите условие запуска макроса. Макрос будет запускаться до тех пор, пока выполняется условие. Для того, чтобы макрос запустился только один раз, нужно заблокировать бит и затем переустановить условие запуска внутри макроса.
3. Используйте объекты [Set Bit] или [Toggle Switch] для установки бита.

b. С использованием объектом [Set Bit] или [Toggle Switch]

1. На вкладке [General] диалогового окна [Set Bit] или [Toggle Switch] выберите опцию Execute Macro.
2. Выберите макрос. Макрос будет выполнен один раз когда указанный объект активирован.

c. С использованием объекта [Function Key]

1. На вкладке [General] диалогового окна [Function Key] выберите Execute Macro.
2. Выберите макрос. Макрос будет выполнен один раз, когда кнопка активная.

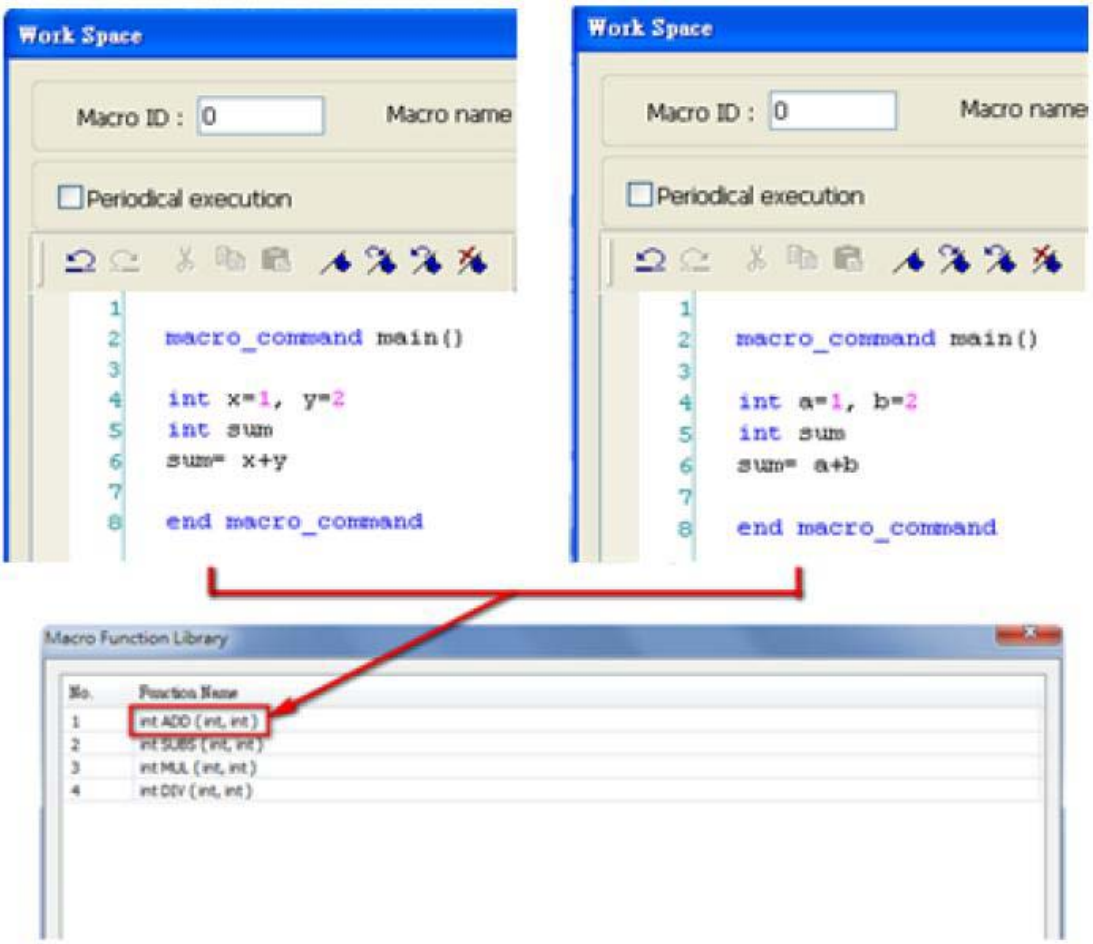
Условия выполнения макроса - Macro Execution Conditions:

1. Периодическое выполнение: макрос будет запускаться периодически.

2. Выполнение один раз при пуске панели (HMI starts): макрос будет выполнен только однажды при пуске панели.

18.8 Макросы определяемые пользователем

Чтобы сократить время редактирования Macro, пользователь может использовать нужные функции из встроенной Библиотеки макрофункций (Macro Function Library). Однако, некоторые функции, хотя и часто используемые, могут там отсутствовать. В этом случае, пользователь может определить нужную функцию и сохранить ее для использования в будущем.
Когда эта функция потребуется снова, сохраненная функция может быть вызвана из Библиотеки макрофункций для облегчения редактирования.
Дополнительно, Библиотеки макрофункций значительно улучшает переносимость функций определяемых пользователем. Перед созданием функции проверьте, может быть существует встроенная функция в библиотеке.



18.8.1 Импорт файла библиотеки функций

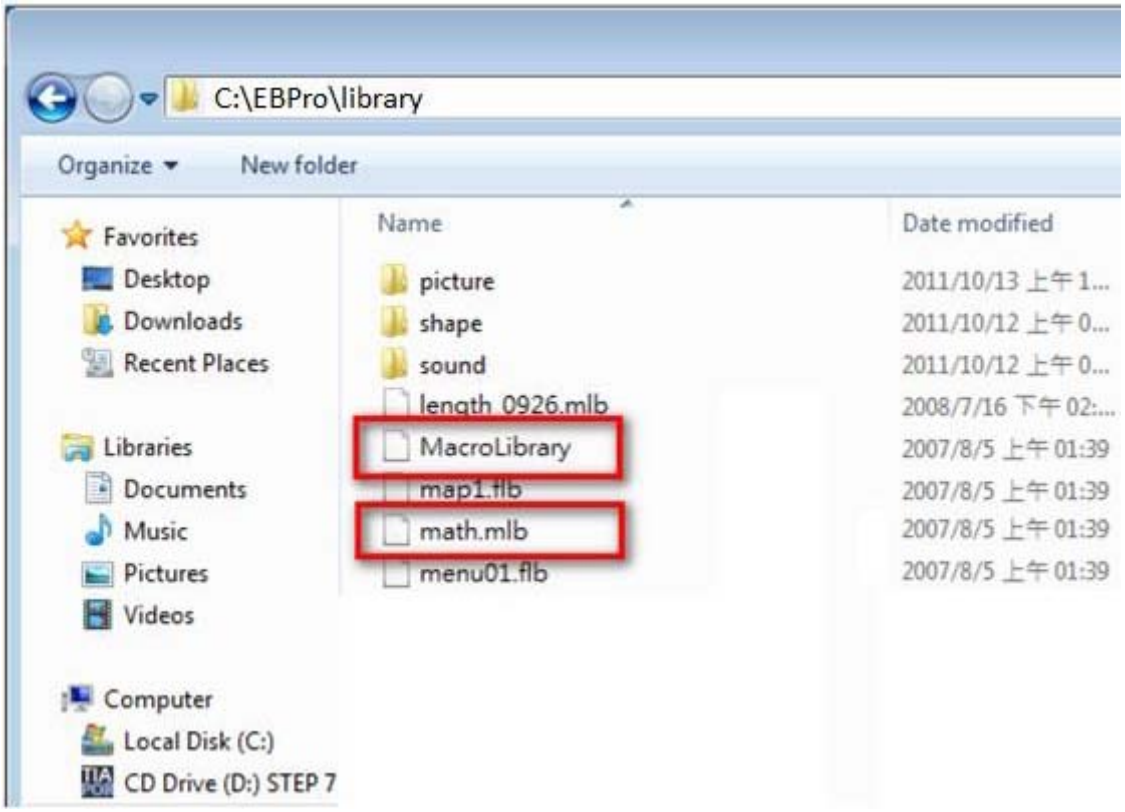
Откройте EasyBuilder Pro, встроенный файл библиотеки (default Function Library File) будет считан автоматически и информация о функциях будет загружена. В этот момент, если вызвана функция определенная пользователем, подходящий файл «*.mlb» должен бы импортирован первым.

1. Имя Библиотеки функций по умолчанию: MacroLibrary (без расширения)

2. Директория Библиотеки функций : Инсталляционная директория EasyBuilder Pro\library (папка)

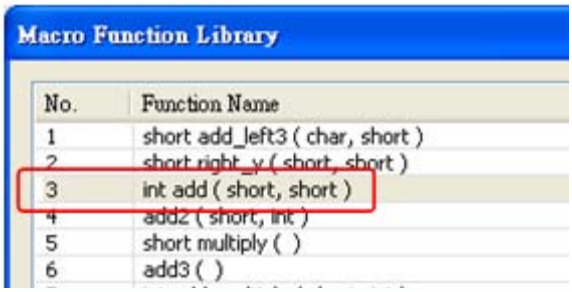
3. Папка « \library» содержит два типа файлов библиотеки функций:
– Без расширения имени файла: MacroLibrary - Default Function Library, которая считывается при запуске EasyBuilder Pro .
– С расширением имени файла (*.mlb): Например: ”math.mlb”. Эти файлы можно читать/писать при помощи импорта/экспорта. Эти файлы переносимые и могут быть вызваны из папки при необходимости.

4. При открытии EasyBuilder Pro, только функции *Default Function Library* будут загружены, чтобы использовать функции в файлах «*.mlb», необходимо их импортировать сначала.

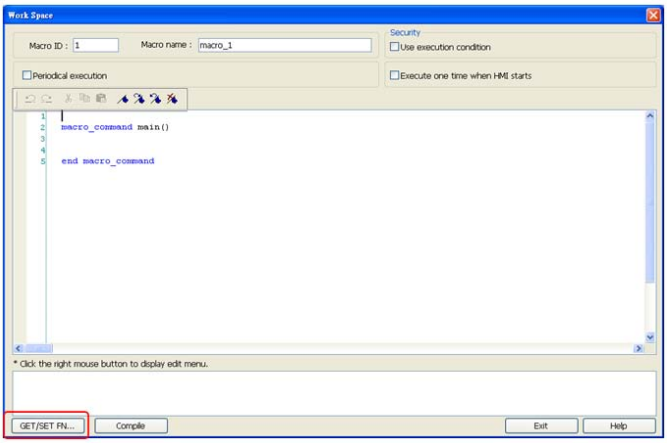


18.8.2 Как использовать библиотеку макрофункций

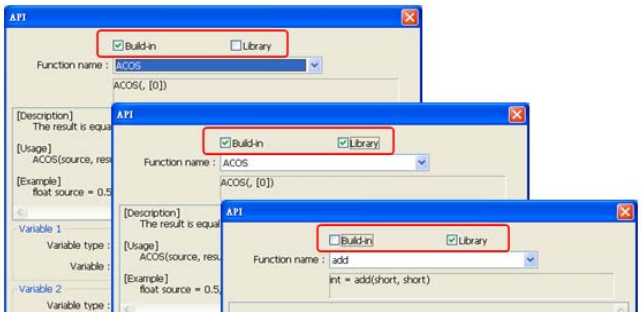
1. Выберите функцию прямо из Macro Function Library.



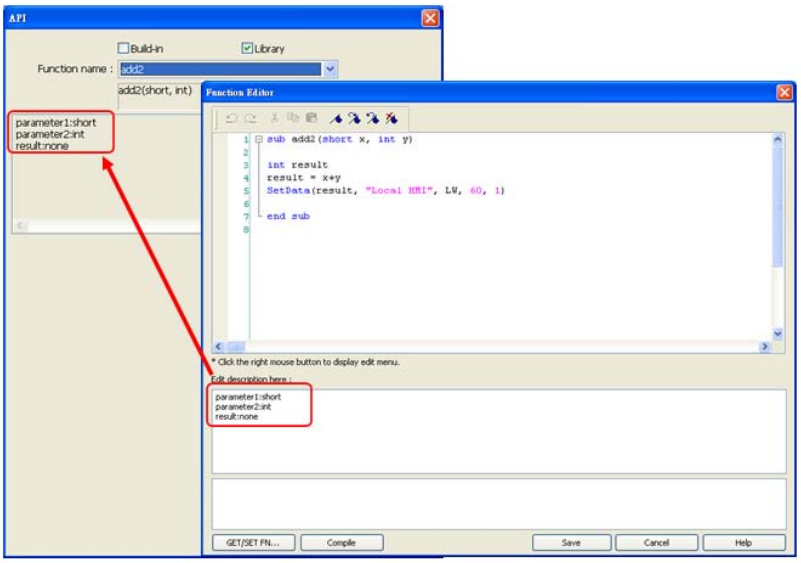
2. В поле редактора щелкните [GET/SET FN…] , чтобы открыть окно диалога API.



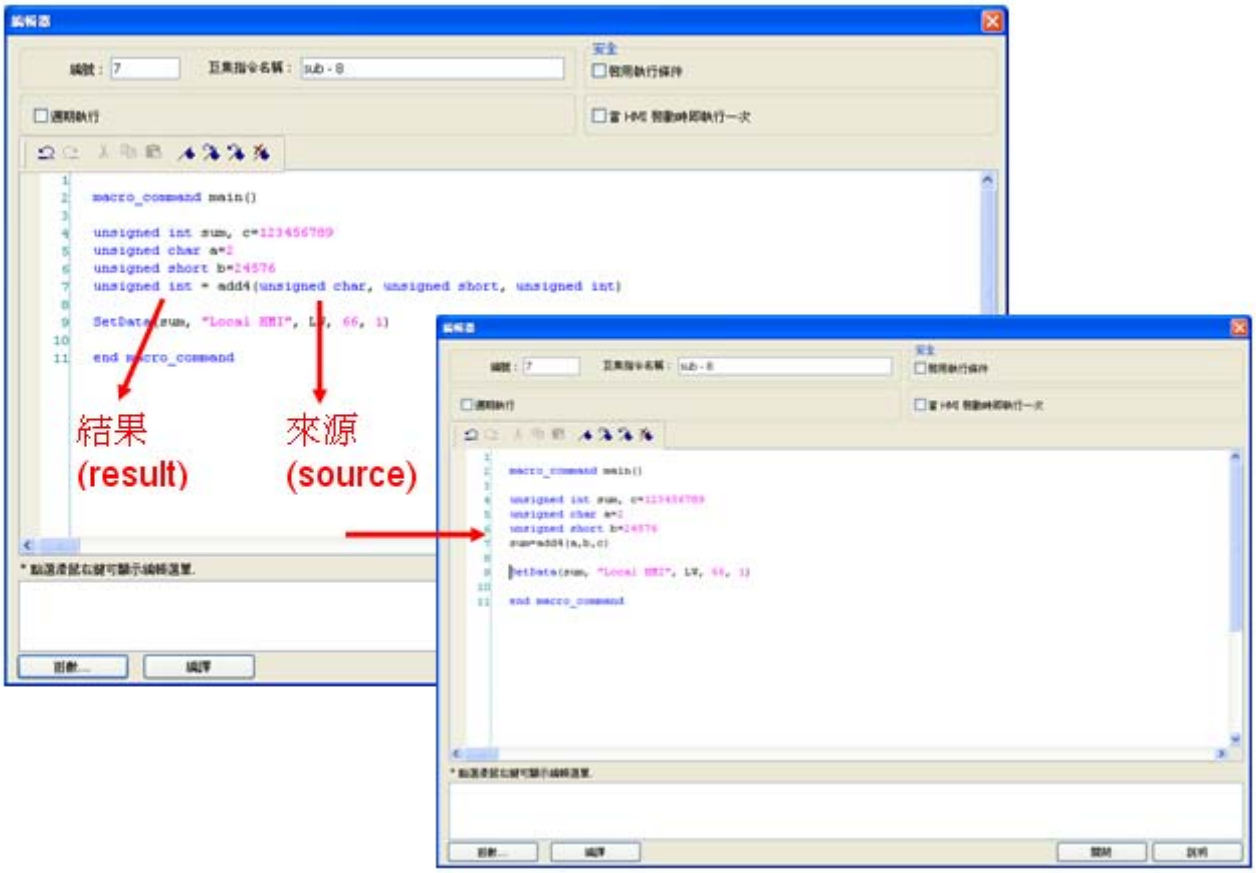
3. Наконец, отметьте [Library] или [Build-in] и выберите функцию для использования.



4. Описание, появляющееся в окне диалога API, то же самое, что написано в редакторе функций (Function Editor).



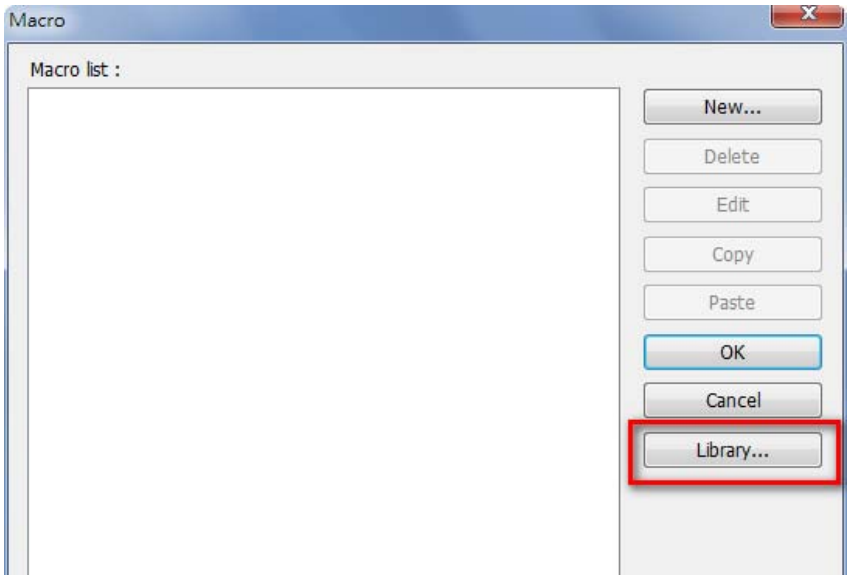
5. Выберите функцию для использования, введите соответствующие переменные в соответствии с типом данных.



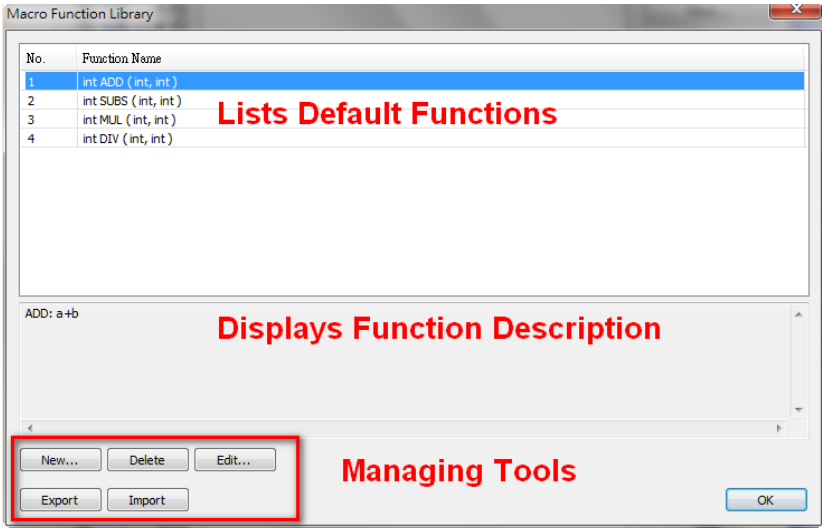
После завершения всех шагов, функция определенаая функция может быть свободно использована без определения той же самой функции повторно.

18.8.3 Интерфейс библиотеки макрофункций

1 Откройте диалог *Macro* , щелкните по [Library], чтобы войти в интерфейс Библиотеки Макрофункций.



2. Появится список функций, при открытии проекта, EasyBuilder Pro загрузит все функции в Библиотеку функций по умолчанию (Default Function Library).



3. Формат каждой строки в списке функций:

return_type function_name (*parameter_type1*, …, *parameter_typeN*

return_type – показывает тип возвращаемого значения. Если это значение не существует, эта колонка будет пропущена.

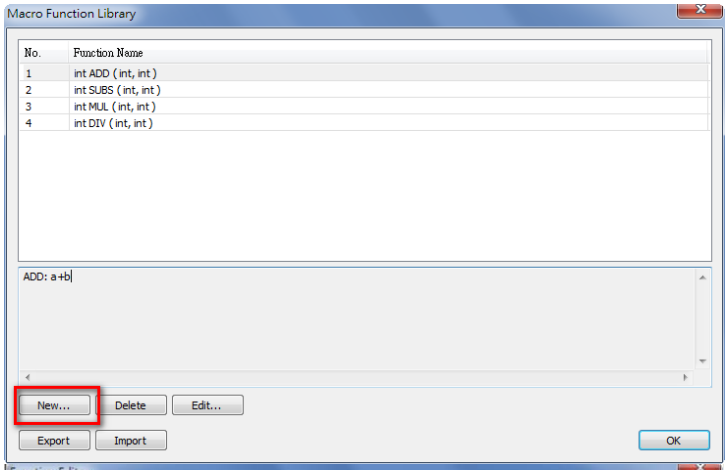
function_name - показывает имя функции

“N” в *parameter_typeN* – предназначен для номера типа параметра. Если эта функция не принимает никаких параметров, эта колонка будет пропущена.

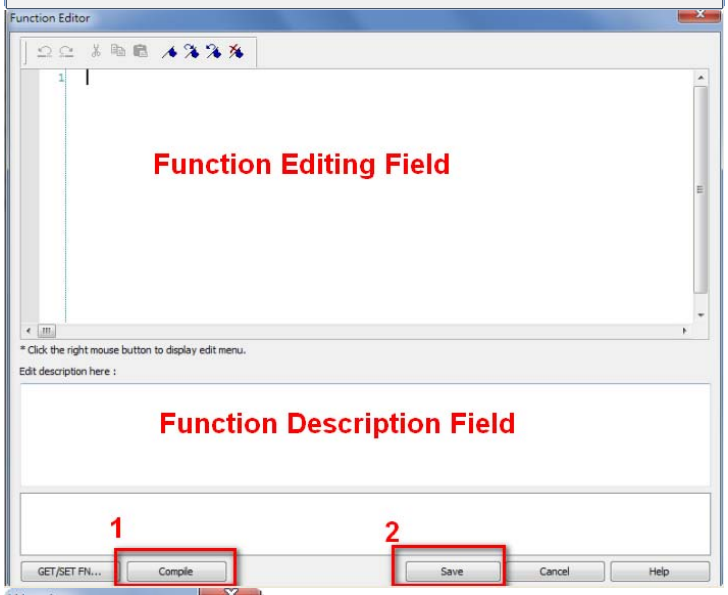
```
1  sub int ADD(int a, int b)
2    int ret
3    ret = a+b
4    return ret
5  end sub
6
```

18.8.3.1 Создание функции

1. Щелкните по [New], чтобы войти в Редактор функций (Function Editor).

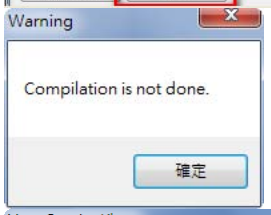


2. Редактируйте функцию в поле редактора (Function Editing Field).

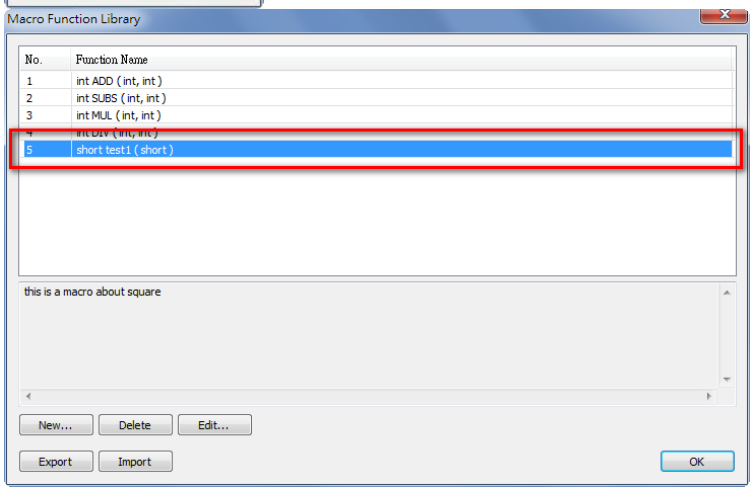


3. Это поле для описания функции: спецификациях, использовании, команд редактора и т. п..

4. После редактирования, щелкните по [Compile] и [Save], чтобы сохранить эту функцию в Библиотеке. Если она не откомпилирована, появится предупреждение.



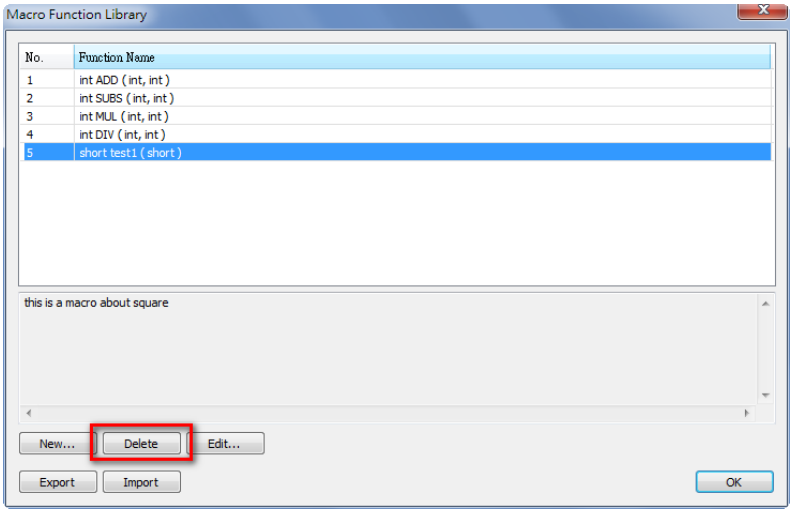
5. Функция успешно добавлена в Macro Function Library.



1. Общий размер типов данных для декларирования в функции - 4096 байт.
2. Имя функции должно содержать только алфавитно-цифровые символы и не может начинаться с номера.

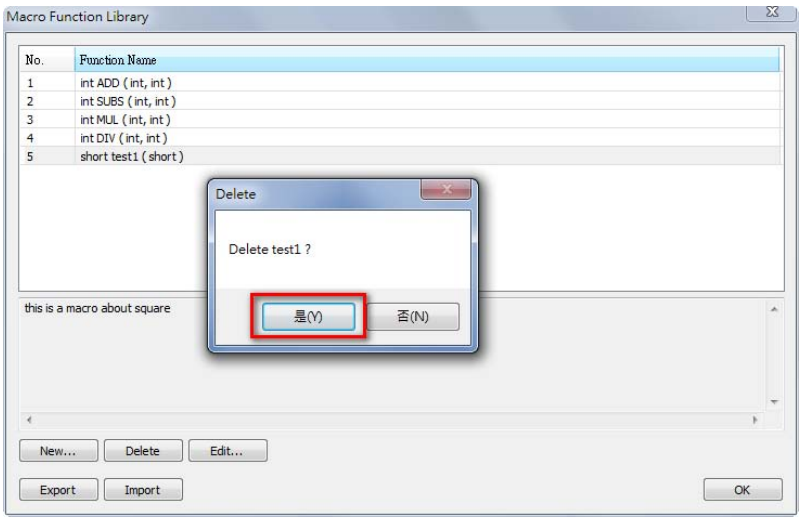
18.8.3.2 Удаление функции

1. В списке функций выберите нужную функцию и нажмите [Delete].



2. Нажмите [Yes] для подтверждения или [No] для отмены удаления.

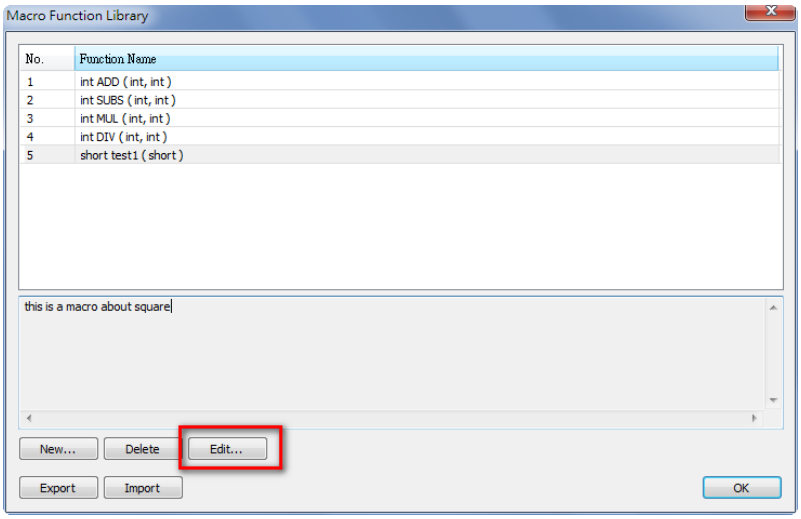
3. Нажмите [Yes] для удаления функции MAX_SHORT.



18.8.3.3 Модификация функции

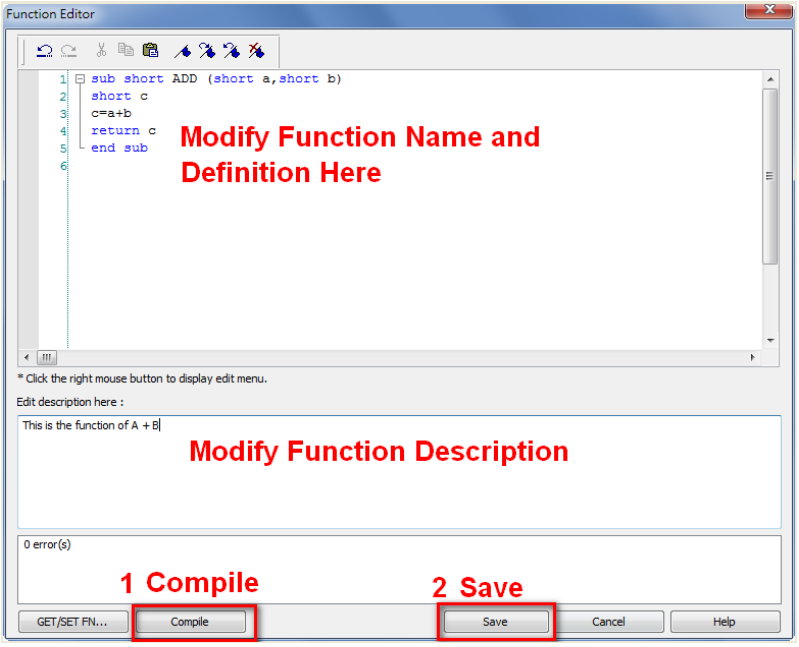
1. Пользователь может модифицировать функцию существующую в Библиотеке.

2. Выберите функцию и щелкните по [Edit], чтобы войти в Редактор функций.



3. Двойной щелчок по функции, также, откроет Редактор функций.

4. После модифицирования, [Compile] и [Save] перед выходом из редактора.

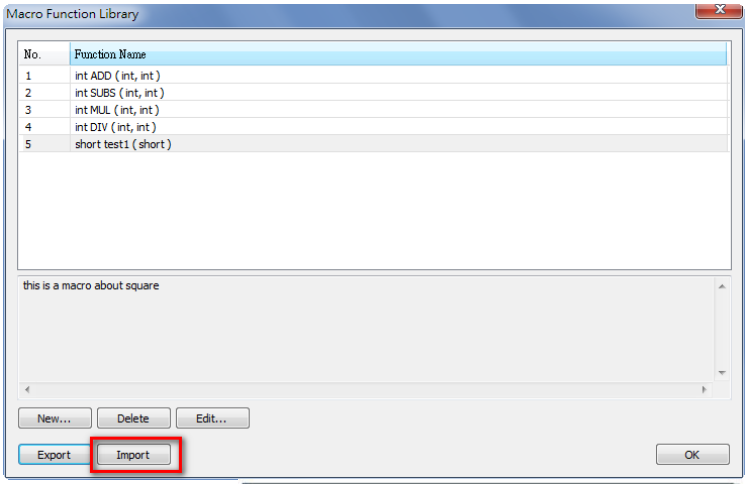


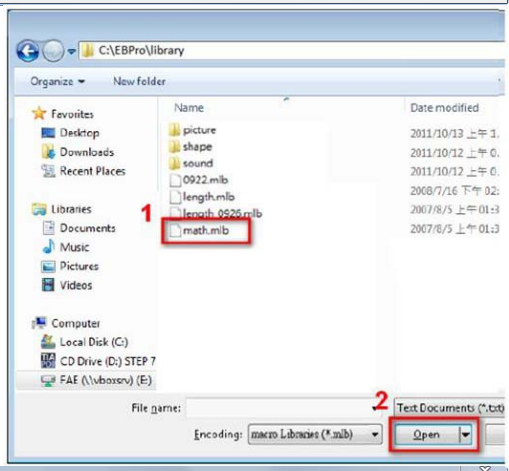
18.8.3.4 Import a Function

1. Функции могут быть импортированы с использованием внешнег файла «*.mlb».

2. Пример: Чтобы импортировать библиотеку функций“math.mlb”, которая содержит функцию “test1”.

3. Щелкните по [Open].





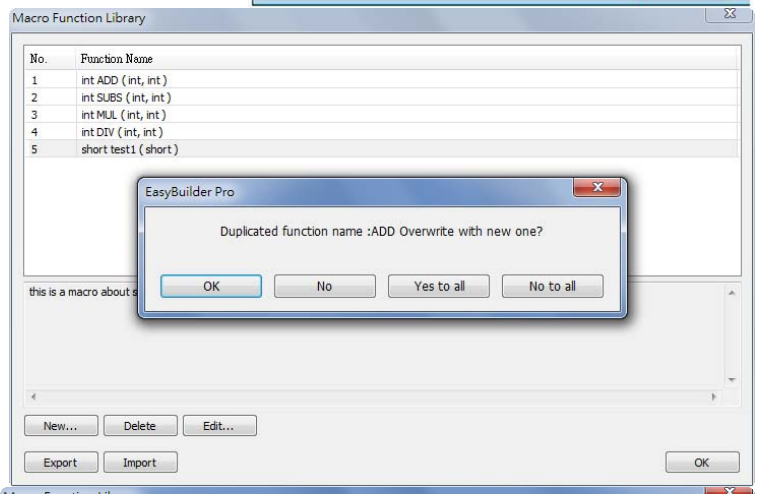
4. При импортировании функции имя которой уже существует в библиотеке, всплывет сообщение:

– OK: Импортируемая функция перепишет существующую.

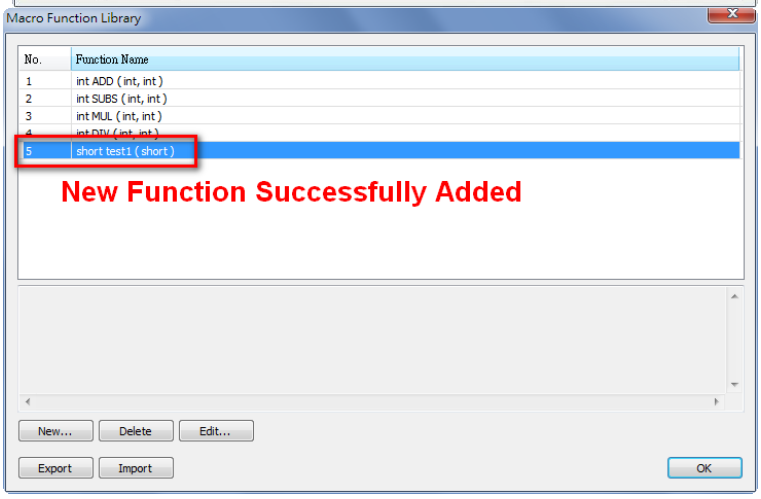
– NO: Отказ от импортирования функции с тем же самым именем.

– Yes to all: Переписать, используя все импортируемые функции с теми же самыми именами.

– No to all: Отказ от импортирования всех функций с теми же самыми именами.



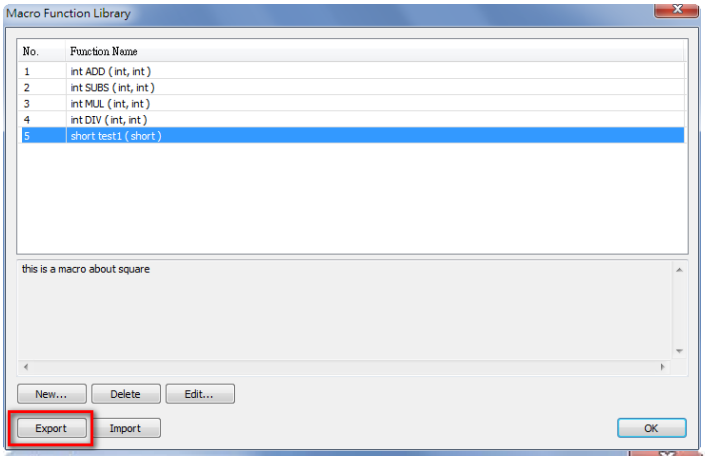
5. Импортированные функции будут сохранены в Default Function Library, поэтому, если файл “math.mlb” удалить, “test1” будет существовать в Библиотеке, даже после рестарта пакета программирования.



18.8.3.5 Export a Function

1. Экспорт функции из Function Library и сохранение ее в файле «*.mlb».

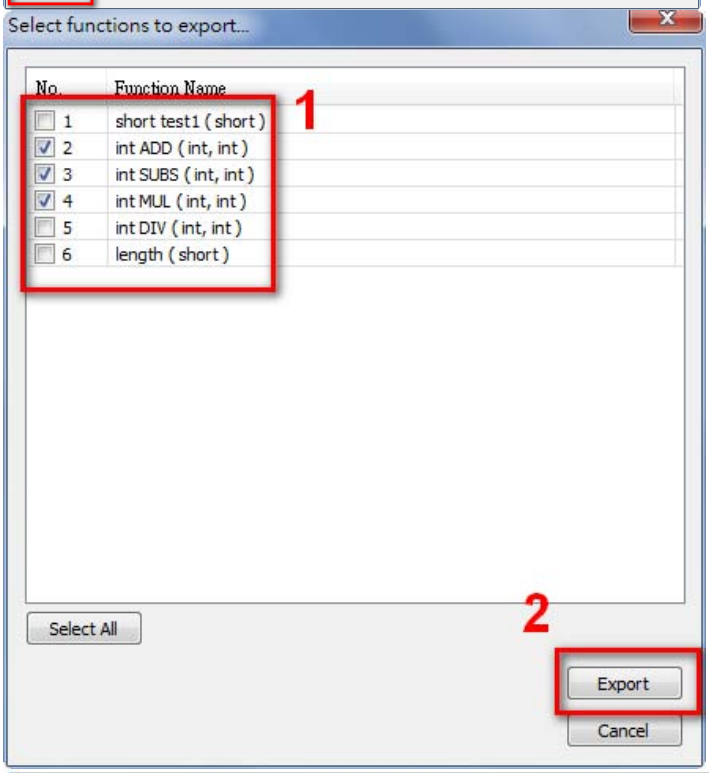
2. Нажмите [Export].



3. Выберите функцию и нажмите [Export].

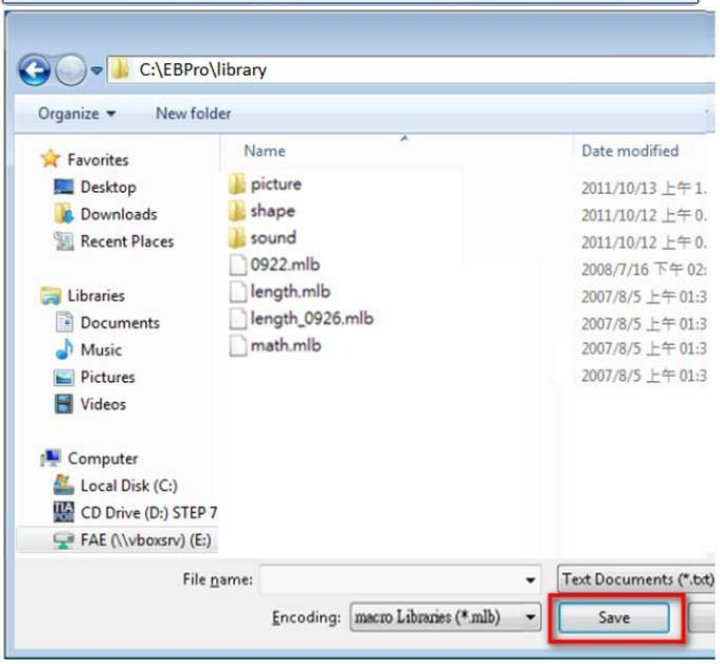
4. Файл “math.mlb” может быть найден в директории «export»

Этот файл содержит 4 функции: ADD, SUBS, MUL, и DIV.



5. Экспортируемый файл *.mlb может быть импортирован другим компьютером.

Откройте EasyBuilder Pro, импортируйте,
затем функции этого файла можно использовать.



18.9 Некоторые замечания об использовании макросов

1. Максимальный объем памяти для хранения локальных переменных составляет 4 килобайта. Таким образом, максимальные размеры массивов различных типов переменных следующие:

   char a[4096]
   bool b[4096]
   short c[2048]
   int d[1024]
   float e[1024]

2. В проекте EasyBuilderPro может быть максимум 256 макросов.

3. Макрос может привести к «зависанию» панели оператора. Возможные причины:
   - Макрос содержит бесконечный цикл без связи с ПЛК.
   - Переполнение памяти, отводенной для массива данных макроса .

4. Время, затрачиваемое на связь с ПЛК, может приводить к замедлению выполнения макроса. Слишком большое число макросов замедляет обмен с ПЛК.

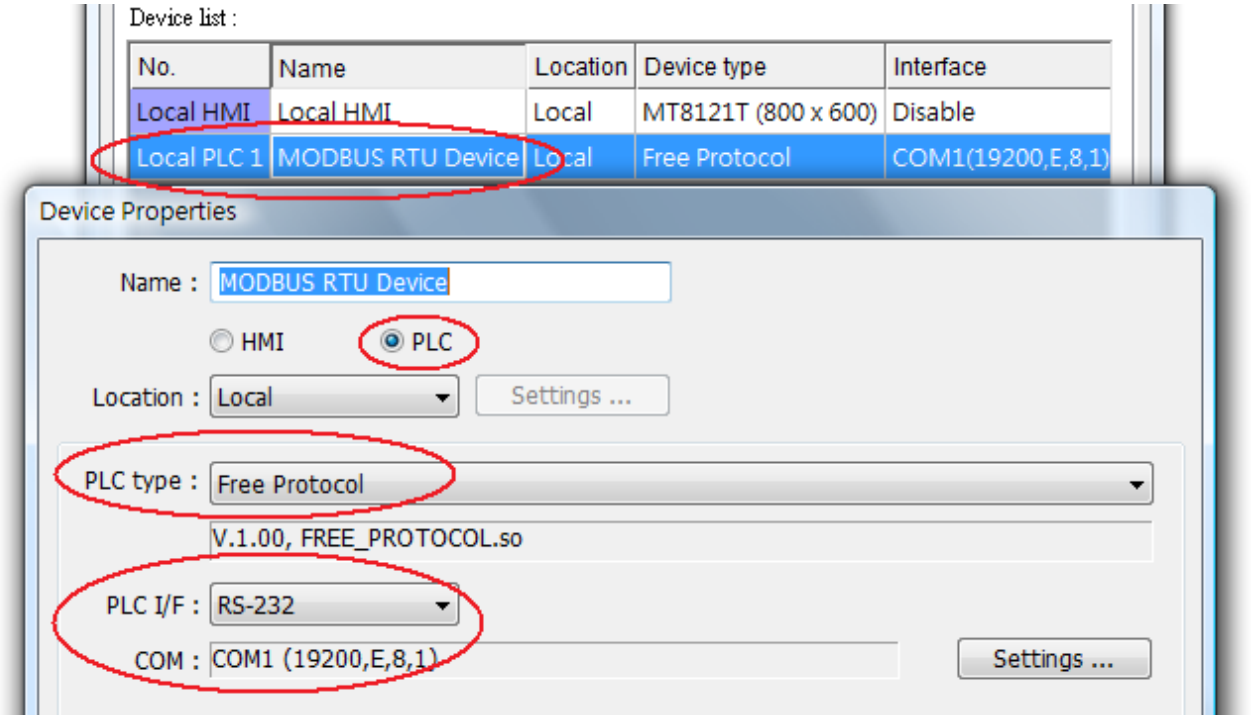
18.10 Использование открытого протокола для управления устройством

Если в EasyBuilderPro нет драйвера для связи с конкретным устройством, пользователь может использовать функции OUTPORT и INPORT для управления устройством.

Данные посылаемые функциями OUTPORT и INPORT должны соответствовать коммуникационному протоколу этого устройства.

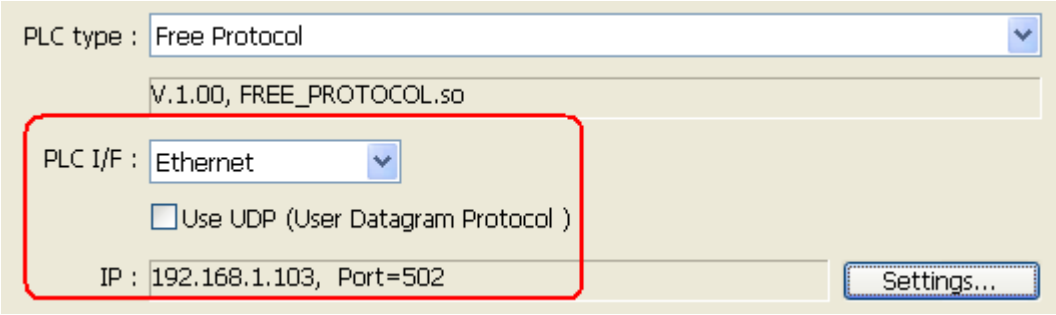
В примере будет показано, как использовать эти функции для управления устройством по протоколу MODBUS RTU.

Сначала создайте новое устройство в *device table*. Тип нового устройства установите "Free Protocol" – Произвольный протокол и назовите "MODBUS RTU device":



Используемый интерфейс соединения с устройством (PLC I/F) - "RS-232" .

Если нужно соединиться с устройством, используя MODBUS TCP/IP, нужно выбрать интерфейс 'Ethernet". Затем, надо настроить IP-адрес и номер порта:



Предположим, что панель будет считывать данные устройства с адресами: 4x_1 и 4x_2.

Сначала, используйте OUTPORT, чтобы послать в устройство запрос на чтение.

Образец OUTPORT:

OUTPORT(command[start], device_name, cmd_count

Запрос на чтение по протоколу MODBUS RTU на чтение Регистров общего назначения - ”Reading Holding Registers (0x03)” . На рисунках показано содержание команды. (Номер станции (байт 0) и последние два байта (CRC) не показаны).

Request

Function code	1 Byte	**0x03**
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	1 to 125 (0x7D)

Response

Function code	1 Byte	**0x03**
Byte count	1 Byte	2 x **N***
Register value	**N*** x 2 Bytes	

***N** = Quantity of Registers

Error

Error code	1 Byte	**0x83**
Exception code	1 Byte	01 or 02 or 03 or 04

Содержание команды чтения - следующее (Всего 8 байт):

command[0] : Номер станции (BYTE 0)

command[1] : Код функции (BYTE 1)

command[2] : Старший байт начального адреса (BYTE 2)

command[3] : Младший байт начального адреса (BYTE 3)

command[4] : Старший байт числа регистров (BYTE 4)

command[5] : Младший байт числа регистров (BYTE 5)

command[6] : Младший байт 16-bit CRC (BYTE 6)

command[7] : Старший байт 16-bit CRC (BYTE 7)

Создаваемый запрос на чтение - следующий :

```
char command[32]
short address, checksum
FILL(command[0], 0, 32) // initialize command[0]~command[31] to 0
command[0] = 0x1 // station number
command[1] = 0x3 // read holding registers (function code is 0x3)
address = 0// starting address (4x_1) is 0
HIBYTE(address, command[2])
LOBYTE(address, command[3])
read_no = 2// общее число считываемых слов - 2 words
HIBYTE(read_no, command[4])
LOBYTE(read_no, command[5])
```

CRC(command[0], checksum, 6)// calculate 16-bit CRC

LOBYTE(checksum, command[6])

HIBYTE(checksum, command[7])

Далее, используем OUPORT, чтобы послать запрос чтения в PLC

OUTPORT(command[0], "MODBUS RTU Device", 8)// send read request

После посылки запроса, используйте INPORT, чтобы получить ответ ПЛК.

Содержимое ответа (response) - следующее (Всего 9 байт):

command[0] : Номер станции (BYTE 0)

command[1] : Код функции (BYTE 1)

command[2] : Число байт (BYTE 2)

command[3] : Старший байт 4x_1 (BYTE 3)

command[4] : Младший байт числа 4x_1 (BYTE 4)

command[5] : Старший байт числа 4x_2 (BYTE 5)

command[6] : Младший байт числа 4x_2 (BYTE 6)

command[7] : Младший байт 16-bit CRC (BYTE 7)

command[8] : Старший байт 16-bit CRC (BYTE 8)

Использование INPORT описано ниже:

INPORT(response[0], "MODBUS RTU Device", 9, return_value)// read response

Действительное число считанных данных записывается в переменную *return_value* (в байтах).

Если *return_value* = 0, это означает, что произошла ошибка исполнения INPORT

response[1] должен быть равен 0x3, если ответ корректный.

После получения корректного ответа, считываются данные из регистров 4x_1 / 4x_2 и put записываются в регисты панели: LW100 и LW101 .

```
if (return_value >0 and response[1] == 0x3) then
read_data[0] = response[4] + (response[3] << 8)// 4x_1
read_data[1] = response[6] + (response[5] << 8)// 4x_2
SetData(read_data[0], "Local HMI", LW, 100, 2)
end if
```

Законченный макрос:

```
//   Read Holding Registers
macro_command main()

char command[32], response[32]
short address, checksum
short read_no, return_value, read_data[2], i

FILL(command[0], 0, 32)//   initialize command[0]~command[31] to 0
FILL(response[0], 0, 32)

command[0] = 0x1//   station number
command[1] = 0x3//   read holding registers (function code is 0x3)

address = 0
address =   0//   starting address (4x_1) is 0
HIBYTE(address, command[2])
LOBYTE(address, command[3])

read_no = 2/   the total words of reading is 2 words
HIBYTE(read_no, command[4])
LOBYTE(read_no, command[5])

CRC(command[0], checksum, 6)//   calculate 16-bit CRC

LOBYTE(checksum, command[6])
HIBYTE(checksum, command[7])

OUTPORT(command[0], "MODBUS RTU Device", 8 )//   send request
INPORT(response[0], "MODBUS RTU Device", 9, return_value)//   read response

if (return_value > 0 and response[1] == 0x3) then
   read_data[0] = response[4] + (response[3] << 8)//   4x_1
   read_data[1] = response[6] + (response[5] << 8)//   4x_2

   SetData(read_data[0], "Local HMI", LW, 100, 2)
end if

end macro_command
```

В следующем примере показано, как сформировать запрос на запись в регистр 0x_1.

В запросе использована команда ”Write Single Coil(0x5)” – Запись в один регистр реле.

Request

Function code	1 Byte	**0x05**
Output Address	2 Bytes	0x0000 to 0xFFFF
Output Value	2 Bytes	0x0000 or 0xFF00

Response

Function code	1 Byte	**0x05**
Output Address	2 Bytes	0x0000 to 0xFFFF
Output Value	2 Bytes	0x0000 or 0xFF00

Error

Error code	1 Byte	**0x85**
Exception code	1 Byte	01 or 02 or 03 or 04

Законченный макрос:

```
//   Write Single Coil (ON)
macro_command main()

char command[32], response[32]
short address, checksum
short i, return_value

FILL(command[0], 0, 32)//   initialize command[0]~ command[31] to 0
FILL(response[0], 0, 32)

command[0] = 0x1//   station number
command[1] = 0x5//   function code : write single coil

address = 0
HIBYTE(address, command[2])
LOBYTE(address, command[3])

command[4] = 0xff//   force 0x_1 on
command[5] = 0

CRC(command[0], checksum, 6)

LOBYTE(checksum, command[6])
HIBYTE(checksum, command[7])

OUTPORT(command[0], "MODBUS RTU Device", 8)//   send request
INPORT(response[0], "MODBUS RTU Device", 8, return_value)//   read response

end macro_command
```

18.11 Сообщение компилятора об ошибках

1. **Формат сообщения об ошибке:**

error c# : error description (номер ошибки # и описание)

Пример: error C37 : undeclared identifier : i

Когда возникают ошибки компиляции, описание ошибки можно найти по ее номеру в сообщении.

2. Описание ошибки

(C1) syntax error : 'identifier'

Есть несколько причин, приводящих к подобным ошибкам.

Например:

```
macro_command main()
char i, 123xyz //  -  это недопустимое имя переменной
end macro_command
```

(C2) 'identifier' used without having been initialized

В макросе должны быть определены размеры массива во время декларации.

Например:

```
macro_command main()
char i
int g[i] // i должно быть числовой константой
end macro_command
```

(C3) redefinition error : 'identifier'

Имена переменной и функции должны быть уникальными

Например:

```
macro_command main()
int g[10] , g // error
end macro_command
```

(C4) function name error : 'identifier'

Служебные слова и константы не могут быть именем функции.

Например:

```
sub int if() // error
```

(C5) parentheses have not come in pairs

Пропущена одна из скобок: “(“ или “)”.

Например:

```
macro_command main ) // missing “(“
```

(C6) illegal expression without matching ‘if’

Пропущено выражение в операторе “if”.

(C7) illegal expression (no ‘then’) without matching ‘if’

Пропущено “then” в операторе “if”.

(C8) illegal expression (no ‘end if’)

Пропущено служебное слово “end if”

(C9 illegal ‘end if’ without matching ‘if’

Незавершенный оператор “If’ перед “End If”

(C10) illegal ‘else’ - (Неверное употребление ‘else’)

Формат оператора “if”:

```
if [logic expression] then
[ else [if [logic expression] then ] ]
end if
```

Любой другой формат данного оператора вызывает ошибки компиляции.

(C17) illegal expression (no 'for') without matching ‘next’

Ошибка в конструкции “for”: пропущено слово “for” перед “next”.

(C18) illegal variable type (not integer or char)

Переменная должна быть типа integer или char.

(C19) variable type error (Ошибка типа переменной)

Переменная не определена

(C20) must be keyword ‘to’ or ‘down’

Пропущено служебное слово “to” или “down”

(C21) illegal expression (no 'next')

Формат оператора “for”:

```
for [variable] = [initial value] to [end value] [step]
next [variable]
```

Любой другой формат приведет к ошибкам компиляции.

(C22) ‘wend’ statement contains no ‘while’

Ошибка в конструкции “While”: пропущено слово “while” перед “Wend”

(C23) illegal expression without matching ‘wend’

Формат цикла “while”:

```
while [logic expression]
wend
```

Любой другой формат приведет к ошибкам компиляции.

(C24) syntax error : ‘break’

Оператор “break” может использоваться только в циклах “for”, “while”

(C25) syntax error : ‘continue’

Оператор “continue” может использоваться только в циклах “for”, “while”

(C26) syntax error

Выражение неверно.

(C27) syntax error

Ошибка в операторе привела к ошибке компиляции.

Например:

```
macro_command main( )
int a, b
for a = 0 to 2
        b = 4 + xyz // illegal : xyz is undefined
next a
end macro_command
```

(C28) must be ‘macro_command’

Должна быть ’macro_command’

(C29) must be key word ‘sub’

Должно присутствовать служебное слово “sub”. Формат объявления функции:

```
sub [data type] function_name(…)
……..
end sub
```

Например:

```
sub int pow(int exp)
…….
end sub
```

Любой другой формат приведет к ошибке компиляции.

(C30) number of parameters is incorrect

Несоответствие числа параметров

(C31) parameter type is incorrect

Несоответствие типа параметра

(C32) variable is incorrect

Параметры функции должны быть эквиваленты аргументам функции для отсутствия ошибок компиляции.

(C33) function name : undeclared function (неопределенная функция)

(C34) expected constant expression (должна быть константа)

(C35) invalid array declaration (неправильное объявление массива)

(C36) array index error (ошибка в индексе массива)

(C37) undeclared identifier : i ‘identifier’

Любая переменная должна быть объявлена до использования в программе.

(C38) un-supported PLC data address

Параметры в функциях GetData(…) , SetData(…) должны быть определены как верные адреса в ПЛК.

(C39) ‘idenifier’ must be integer, char or constant

Формат массива:

Объявление: array_name[constant] (constant – это размер массива)
Использование массива: array_name[integer, character or constant]

Любой другой формат может привести к ошибкам компиляции.

(C40) execution syntax should not exist before variable declaration or constant definition

Например:

```
macro_command main( )
int a, b
for a = 0 To 2
    b = 4 + a
int h , k// illegal – определение должно быть сделано до любой команды или выражения
next a
end macro_command
```

(C41) float variables cannot be contained in shift calculation (переменные вещественного типа не могут использоваться в операторах смещения)

(C42) function must return a value (функция должна возвращать значение)

(C43) function should not return a value (функция не возвращает значение)

(C44) float variables cannot be contained in calculation (вещественные переменные не могут использоваться в вычислениях)

(C45) PLC address error (ошибка в адресе ПЛК)

(C46) array size overflow (max. 4k) (превышен максимальный размер массива – 4 кб)

(C47) macro command entry function is not only one (несколько главных функций макроса)

(C48) macro command entry function must be only one

Должна быть только одна главная функция макроса:

```
macro_command function_name( )
end macro_command
```

(C49) an extended addressee's station number must be between 0 and 255

Например:

```
SetData(bits[0] , "PLC 1", LB , 300#123, 100)
// illegal : 300#123 означает, что номер станции 300, в то время как максимально
допустимое значение: 255
```

(C50) an invalid PLC name

Данное имя ПЛК не определено в списке устройств [system parameters].

(C51) macro command do not control a remote device

Макрос может управлять только локальным устройством.

Например:

SetData(bits[0] , “PLC 1”, LB , 300#123, 100)

“PLC 1“ - связан с удаленной панелью, поэтому он не может работать.

18.12 Образцы кода макроса

1. Конструкция ”for” и другие выражения (арифметические, смещение влево, логические и сравнения)

```
macro_command main()
int a[10], b[10], i

b[0]= (400 + 400 << 2) / 401
b[1]= 22 *2 - 30 % 7
b[2]= 111 >> 2
b[3]= 403 > 9 + 3 >= 9 + 3 < 4 + 3 <= 8 + 8 == 8
b[4]= not 8 + 1 and 2 + 1 or 0 + 1 xor 2
b[5]= 405 and 3 and not 0
b[6]= 8 & 4 + 4 & 4 + 8 | 4 + 8 ^ 4
b[7]= 6 – (~4)
b[8]= 0x11
b[9]= 409

for i = 0 to 4 step 1
    if (a[0] == 400) then
        GetData(a[0],”Device 1”, 4x, 0,9)
        GetData(b[0],”Device 1”, 4x, 11,10)
end If
next i
end macro_command
```

2. Конструкции: “while”, “if” и “break”

```
macro_command main()
int b[10], i
i = 5
while i == 5 - 20 % 3
    GetData(b[1], ”Device 1”, 4x, 11, 1)
    if b[1] == 100 then
        break
    end if
wend
end macro_command
```

3. Глобальные переменные и вызов функции

```
char g
sub int fun(int j, int k)
int y

SetData(j, "Local HMI", LB, 14, 1)
GetData(y, "Local HMI", LB, 15, 1)
g = y

return y
end Sub

macro_command main()
int a, b, i

a = 2
b = 3
i = fun(a, b)
SetData(i, "Local HMI", LB, 16, 1)
end macro_command
```

4. Оператор"if"

```
macro_command main()
int k[10], j

for j = 0 to 10
    k[j] = j
next j

if k[0] == 0 then
    SetData(k[1], "Device 1", 4x, 0, 1)
end if

if k[0] == 0 then
```

```
        SetData(k[1], “Device 1”, 4x, 0, 1)
else
        SetData(k[2], “Device 1”, 4x, 0, 1)
end if

if k[0] == 0 then
        SetData(k[1], “Device 1”, 4x, 1, 1)
else if k[2] == 1 then
        SetData(k[3], “Device 1”, 4x, 2, 1)
end If

if k[0] == 0 then
        SetData(k[1], “Device 1”, 4x, 3, 1)
else if k[2] == 2 then
        SetData(k[3], “Device 1”, 4x, 4, 1)
else
        SetData(k[4], “Device 1”, 4x, 5, 1)
end If
end macro_command
```

5. Конструкция “while” и “wend”

```
macro_command main()
char i = 0
int a[13], b[14], c = 4848

b[0] = 13

while b[0]
        a[i] = 20 + i * 10

        if a[i] == 120 then
                c =200
                break
        end if

        i = i + 1
wend
SetData(c, “Device 1”, 4x, 2, 1)
end macro_command
```

6. Операторы “break” and “continue”

```
macro_command main()
char i = 0
int a[13], b[14], c = 4848
b[0] = 13
while b[0]
        a[i] = 20 + i * 10
        if a[i] == 120 then
                c =200
                i = i + 1
                continue
        end if
        i = i + 1
        if c == 200 then
                SetData(c, “Device 1”, 4x, 2, 1)
        break
        end if
wend
end macro_command
```

7. Array

```
macro_command main()
int a[25], b[25], i
b[0] = 13
for i = 0 to b[0] step 1
            a[i] = 20 + i * 10
next i
SetData(a[0], “Device 1”, 4x, 0, 13)
end macro_command
```

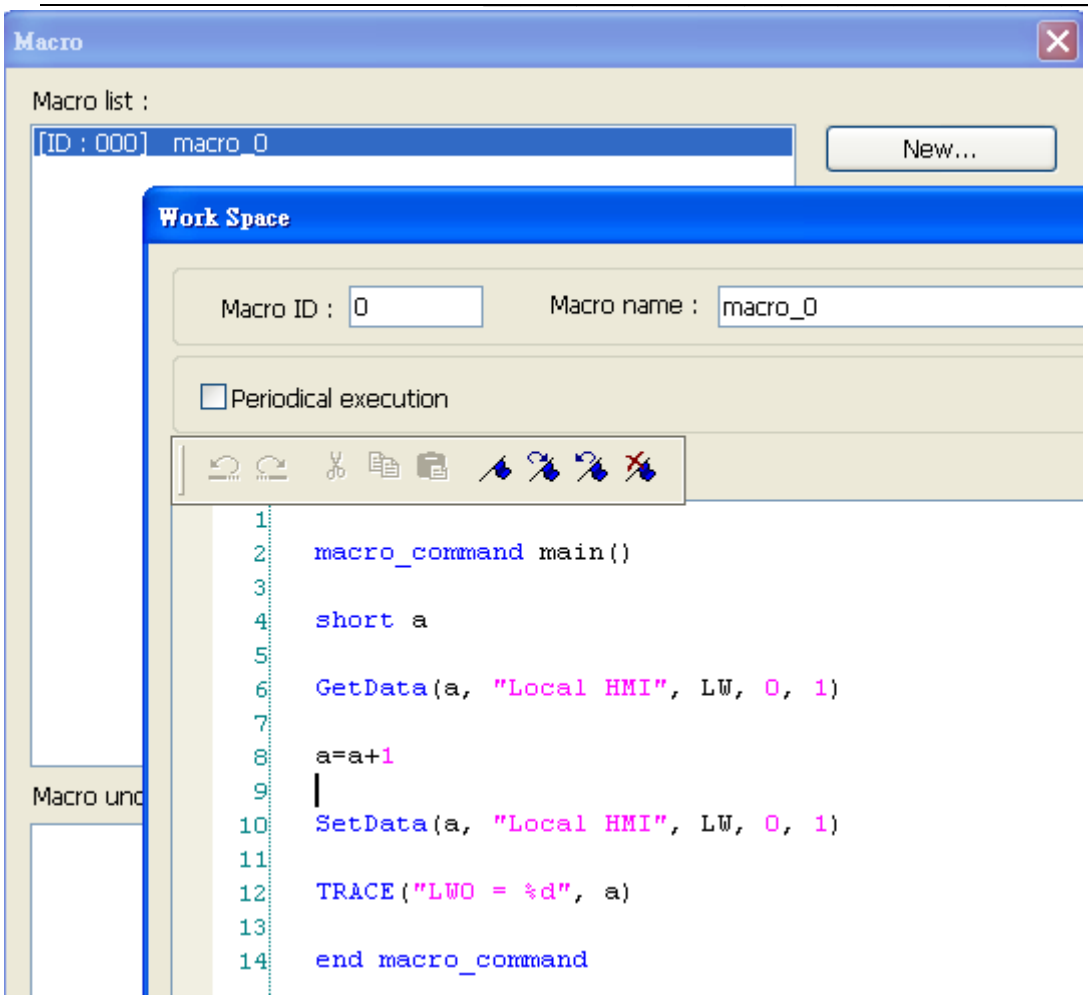
18.13 Макрофункция TRACE

1. Функция TRACE может быть добавлена к макросу и может быть использована совместно с **EasyDiagnoser**, для просмотра текущего состояния используемых переменных.
Далее показано, как использовать функцию TRACE в MACRO.

Сначала, добавьте *macro_1* в проект, а в *macro_1* добавьте TRACE (“LW = %d”, a).
“%d” – означает индикация текущего значения LW в десятичном формате.
Содержание *macro_1* следующее:

```
macro_command main()

short a

GetData(a, "Local HMI", LW, 0, 1)

a= a + 1

SetData(a, "Local HMI", LW, 0, 1)

TRACE ("LW0 = %d", a)

end macro_command
```

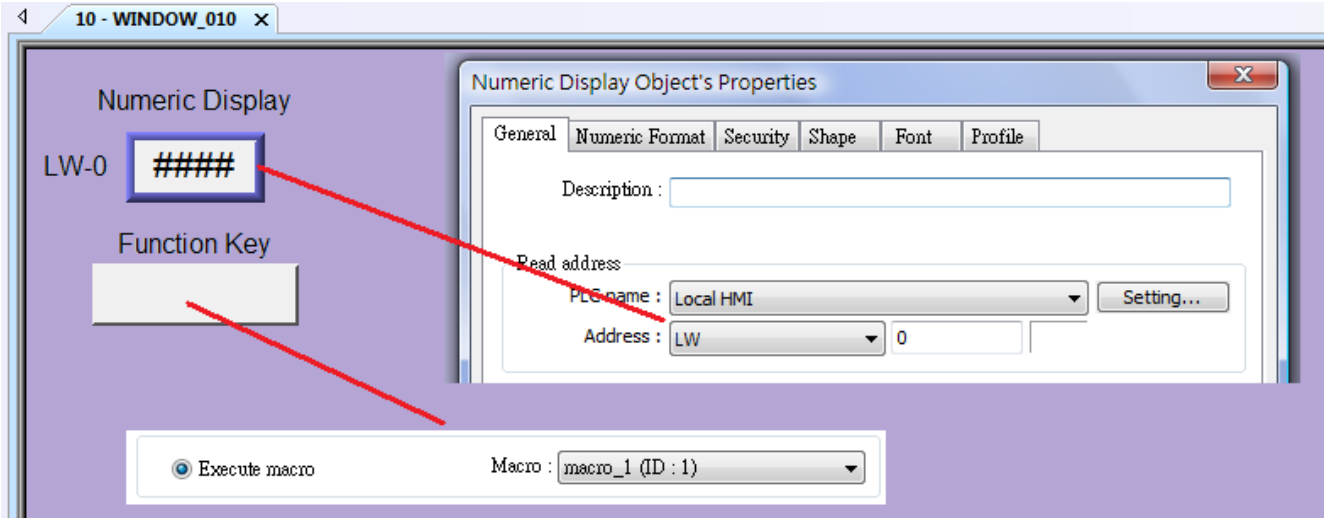
Подробнее об использовании функции TRACE можно узнать по иллюстрациям в следующем разделе.



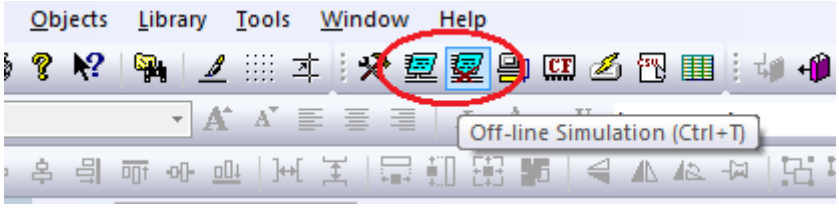
Далее, добавьте объекты *Numeric Display* и *Function Key* в окно 10 проекта.

Настройки этх объектов показаны далее.

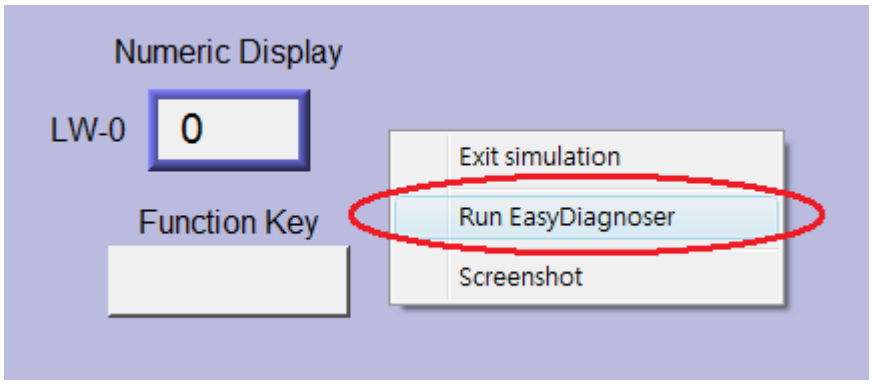
Объект Function Key использован для выполнения ***macro_1.***



Наконец, откомпилируйте проект и выполните имитацию работы проекта *Off-line* или *On-line*.



При выполнении имитации на компьютере, сделайте щелчок правой кнопкой и выберите “Run EasyDiagnoser” во всплывающем меню.

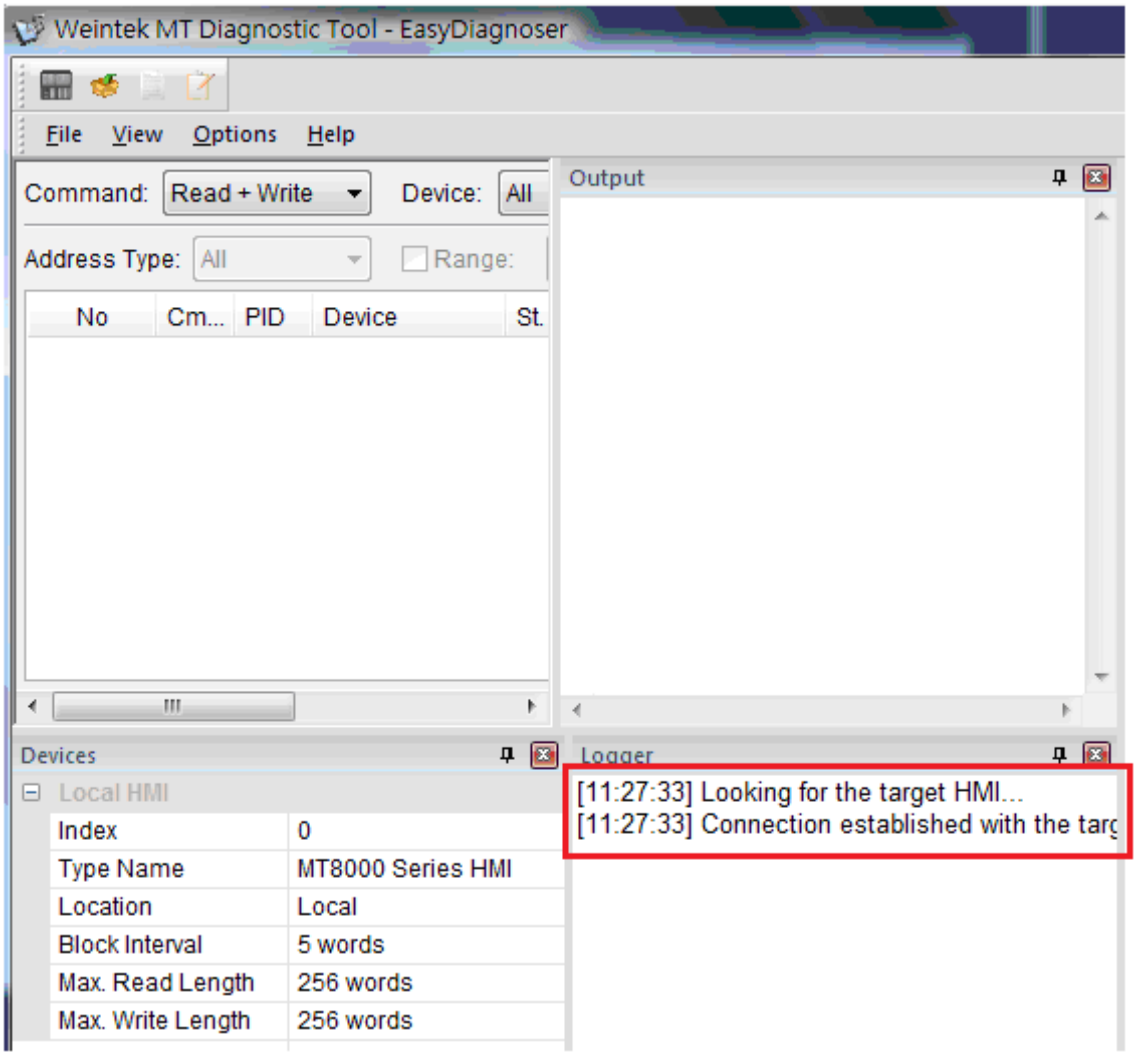


Потом, запустите **EasyDiagnoser**.

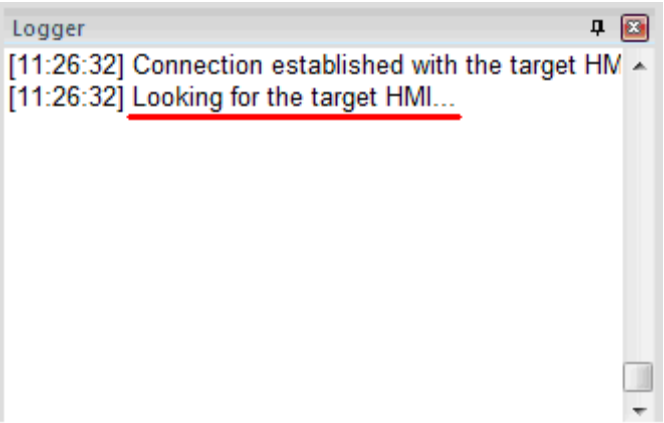
Окно [Logger] покажет - может ли EasyDiagnoser соединиться с панелью для наблюдения за ее работой или нет.

Окно [Output] отобразит выход функции TRACE.

Рисунок на следующей странице показывает, что EasyDiagnoser успешно соединился с панелью.

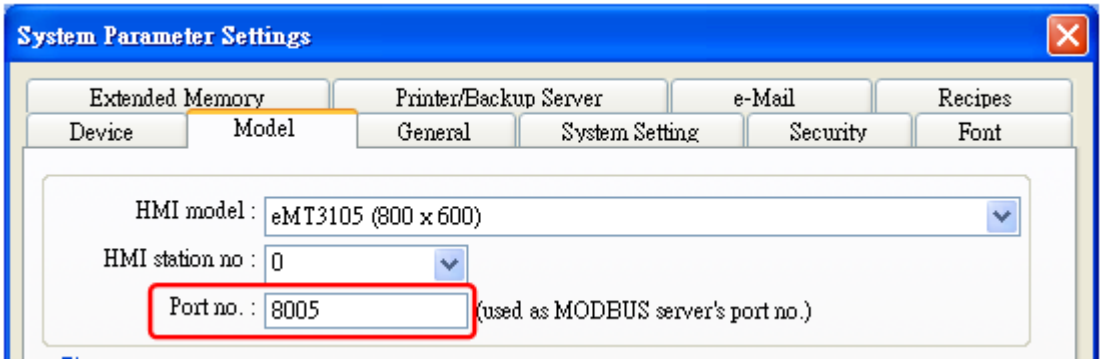


Когда EasyDiagnoser не в состоянии соединиться с панелью, окно [Logger] отобразит следующее содержание:

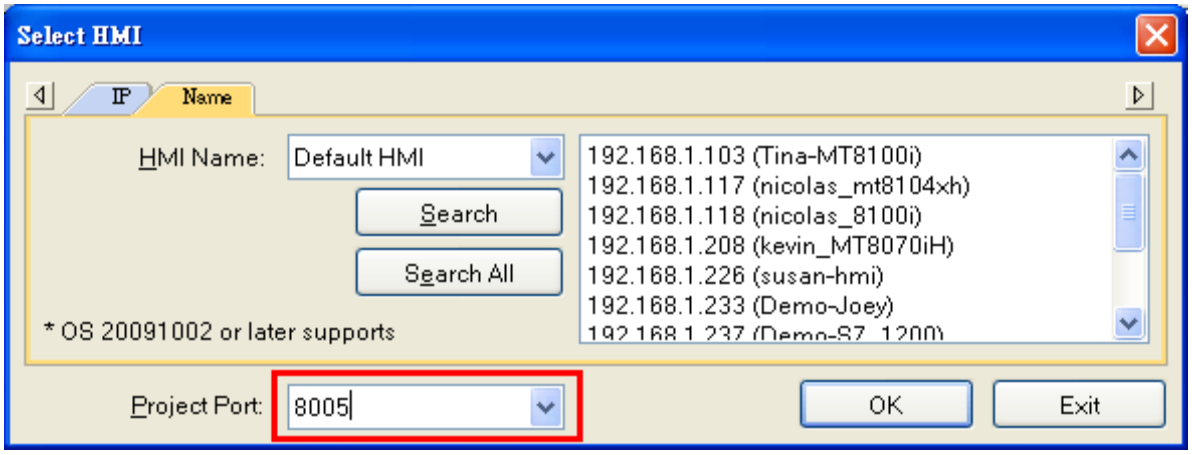


Looking for target HMI – Поиск целевой панели.

Возможной причиной отказа создания соединения с панелью может быть ошибка выполнения режима имитации (simulation) на компьютере. Другой причиной может быть некорректный е номер порта «Port No.» использованного в проекте для симуляции на компьютере (или этот порт занят системой). Измените «Port No», как показано, откомпилируйте проект и запустите режим имитации снова.



При открытии EasyDiagnoser, «Port No.» должен быть тем же самым, что и в проекте. Только так можно успешно соединиться.



Три последовательных номера порта «port no.» резервируются в проекте are для соединения с панелью.

Для примера рассмотрим следующие настройки: Если Port No. настроен на 8005, то порты 8005, 8006 и 8007 будут зарезервированы. В этом случае, при выполнении симуляции на компьютере, убедитесь, что эти порты не заняты другими программами.

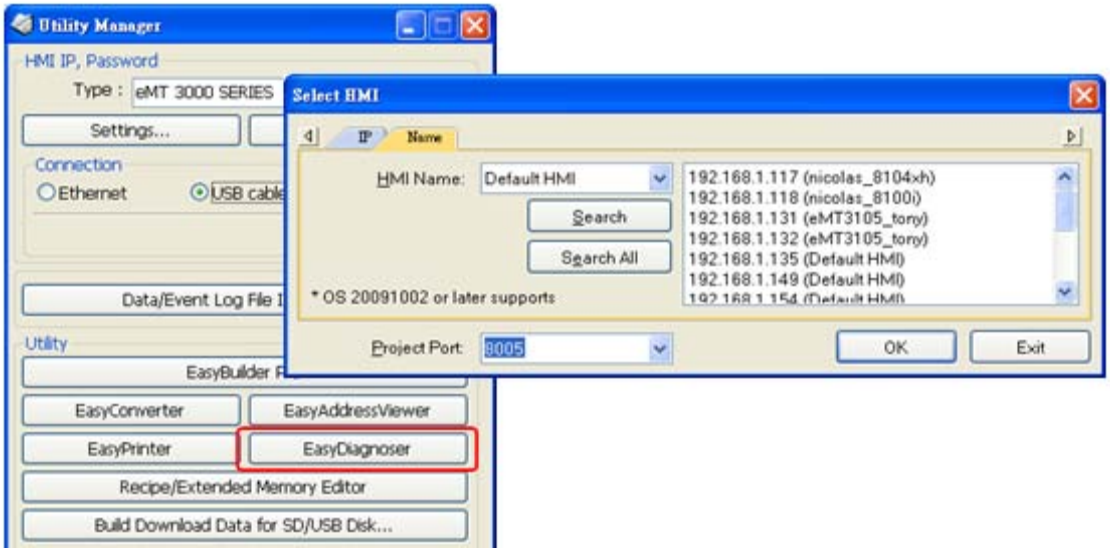
2. Синтаксис функции TRACE:

Имя	TRACE
Синтаксис	TRACE(format, argument)
Описание	Эту функции используют, чтобы послать указанную строку в E*asyDiagnoser*. Пользователь может напечатать текущие значения переменных в реальном времени для отладки. Когда TRACE встречает первую спецификацию формата (если их несколько), она преобразует значение первого аргумента после *format* и выводит его соответственно. *format* – относится к формату строк на выходе функции. Спецификация формата состоит из не обязательных полей (в [ ]) и обязательных полей (жирный шрифт), и имеет следующую форму: **%**[flags] [width] [.precision] **type** Каждое поле формата описано ниже: *flags* (не обязательный): - + *width* (не обязательный): Неотрицательное целое десятичное число – указывает минимальное число символов для печати. *precision* (не обязательный): Неотрицательное целое десятичное число – указывает точность и число символов для печати. *type:* C или c: однобайтовый символ. d: десятичное целое число со знаком. i: десятичное целое число со знаком. o: октальное целое число без знака. u: десятичное целое число без знака. X или x: шестнадцатиричное целое число без знака. E или e: значение со знаком, имеющее следующую форму. [–]*d***.***dddd* **e** [*sign*]*ddd,* где *d* –одна десятичная цифра, *dddd* –одна или больше десятичных цифр, *ddd* –точно три десятичных цифры, и *sign* - «+» или «–». f : Знаковое значение имеющее форму [–]*dddd***.***dddd*, где *dddd* –одна или больше десятичных цифр. Длина стринга на выходе ограничена 256 символами. Лишние символы будут проигнорированы. Часть *argument* - не обязательная.

Пример	macro_command main()
	char c1 = 'a'
	short s1 = 32767
	float f1 = 1.234567
	
	TRACE("The results are") // output: The results are
	TRACE("c1 = %c, s1 = %d, f1 = %f", c1, s1, f1)
	// output: c1 = a, s1 = 32767, f1 = 1.234567
	
	end macro_command

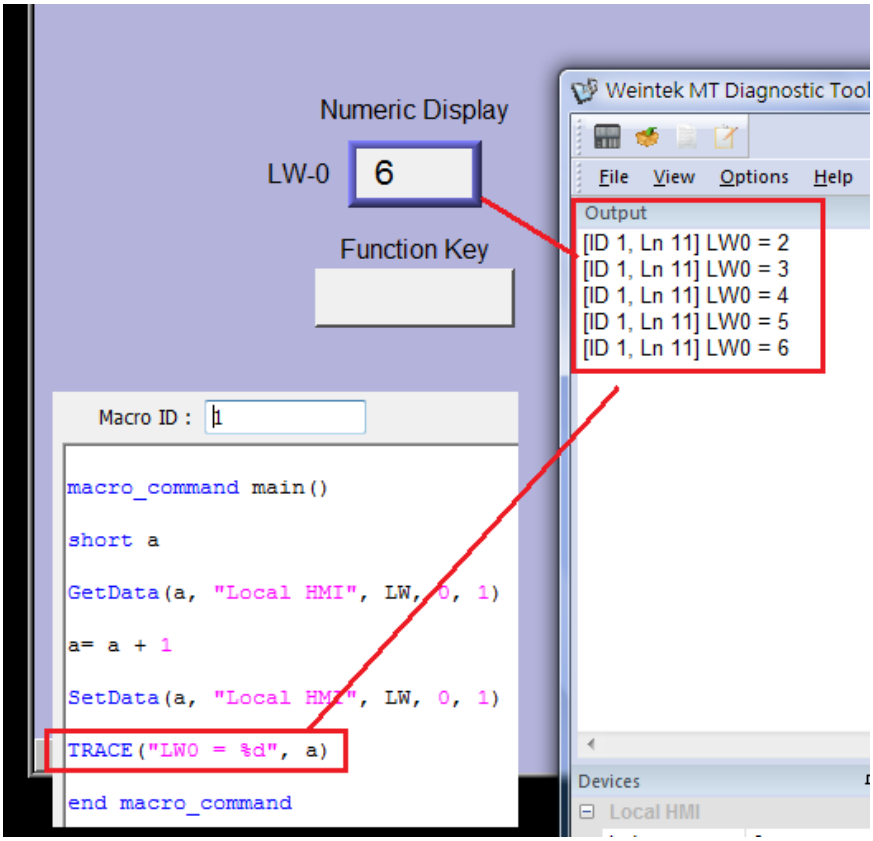
3. Недавно добавленная переменная LB9059 – отключает макрофункцию TRACE function (когда находится в состоянии ON). При этом сообщение выхода TRACE не будет посылаться в *EasyDiagnoser*.

4. Пользователь может запустить *EasyDiagnoser*.exe прямо из *Utility Manager*. В *Utility Manager*, текущая присоединенная панель будет в списке; пользователь иожет выбрать панель и наблюдать за работой.



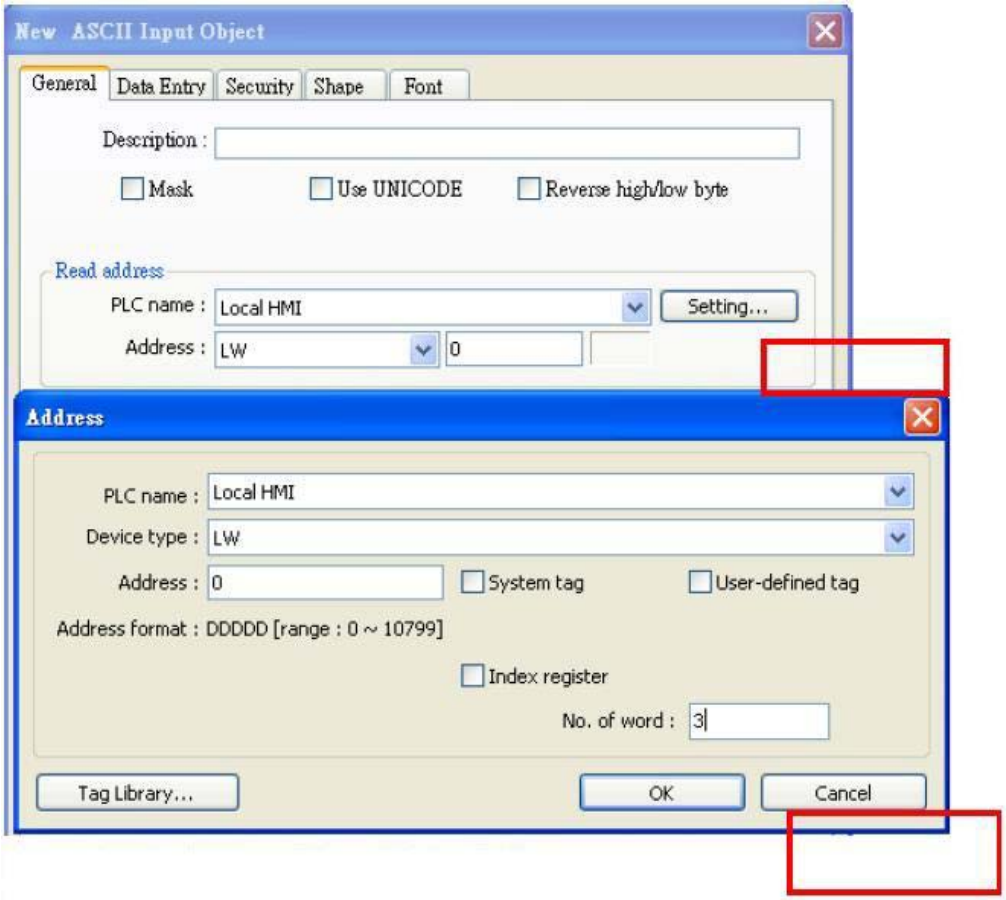
5. Загрузите проект в панель. Если EasyDiagnoser не в состоянии соединиться с панелью, то возможно, панель не включена, или *Port No*. не корректный. Это может вызвать непрерывное включение/отключение EasyDiagnoser от панели. Проверьте соответствие *Port No.* в настройках и в проекте, они должны совпадать. Исправьте *Port No.* как было показано ранее.

6. После соединения EasyDiagnoser с панелью, просто выполните *macro_1*, В окне [Output] появится выход функции TRACE.



18.14 Использование функций работы со стрингами

Функций работы со стрингами добавлены в макросы для удобства работы со стрингами. Термин “string” (строка) означает последовательность символов ASCII , которые занимают 1 байт. Последовательность символов может быть сохранена в 16-ти битовых регистрах с наименее значимым байтом первым. Напрмер, создайте «*ASCII input object*» и настройте его так:



Запустите режим имитации и введите “abcdef”:

Строка “abcdef” хранится в LW0~LW2 (LB - low byte (младший байт) и HB - high byte (старший байт)):

	HB	LB
LW0	'B'	'A'
LW1	'D'	'C'
LW2	'F'	'E'
LW3		
LW4		
LW5		

Объект «ASCII input» читает 1 слово (2 байта) одновременно.

Если объект «ASCII input» настроен на чтение 3 слов (как показано на примере вверху), он може читать, самое большее 6 символов ASCII, так как один символ ASCII занимает 1 байт.

Функции всех операций со строками описаны в таблице на следующей странице:

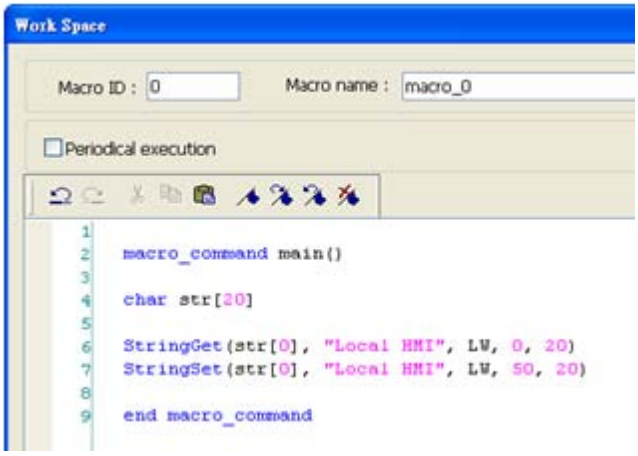
Имя функции	Описание
StringGet	Чтение строковых данных из устройства.
StringGetEx	Чтение строковых данных из устройства и выполнение следующей команды, даже если нет ответа от устройства.
StringSet	Запись строковых данных в устройство.
StringSetEx	Запись строковых данных в устройство и выполнение следующей команды, даже если нет ответа от устройства.
StringCopy	Копирование одной строки в другую.
StringMid	Извлечение подстроки (substring).
StringDecAsc2Bin	Преобразование десятичной строки в целочисленную.
StringBin2DecAsc	Преобразование целочисленной строки в десятичную.
StringDecAsc2Float	Преобразование десятичной строки в формат с плавающей запятой.
StringFloat2DecAsc	Преобразование строки формата с плавающей запятой в десятичную.
StringHexAsc2Bin	Преобразование шестнадцатиричной строки в двоичные данные.
StringBin2HexAsc	Преобразование двоичных данных в шестнадцатиричную строку.
StringLength	Получение длины строки.
StringCat	Дабавление строки-источника к строке назначения.
StringCompare	Сравнение двух строк с учетом регистра (case-sensitive).
StringCompareNoCase	Сравнение двух строк без учета регистра.
StringFind	Поиск подстроки в большой строке.
StringReverseFind	Поиск подстроки в большой строке, начиная с конца.
StringFindOneOf	Поиск первого совпадающего символ в наборе.
StringIncluding	Извлечение подстроки, которая содержит только символы из набора.
StringExcluding	Извлечение подстроки, которая содержит только символы отсутствующих в наборе.
StringToUpper	Преобразование символов строки в верхний регистр (uppercase).
StringToLower	Преобразование символов строки в нижний регистр (lowercase).
StringToReverse	Реверсирование символов в строке.
StringTrimLeft	Удаление указанного набора ведущих символов из строки-источника.
StringTrimRight	Удаление указанного набора замыкающих символов из строки-источника.
StringInsert	Вставка строки в указанное место в другой строке.

Более подробная информация по работе со строками прведена в разделе “Build-In Function Block” –Встроенные функциональные блоки.
Для того, чтобы продемонстрировать возможности функций работы со строками, далее приведены несколько примеров создания файлов проекта и их использования с применением новых макрофункций.

1. Чтение (или запись) строки из устройства.

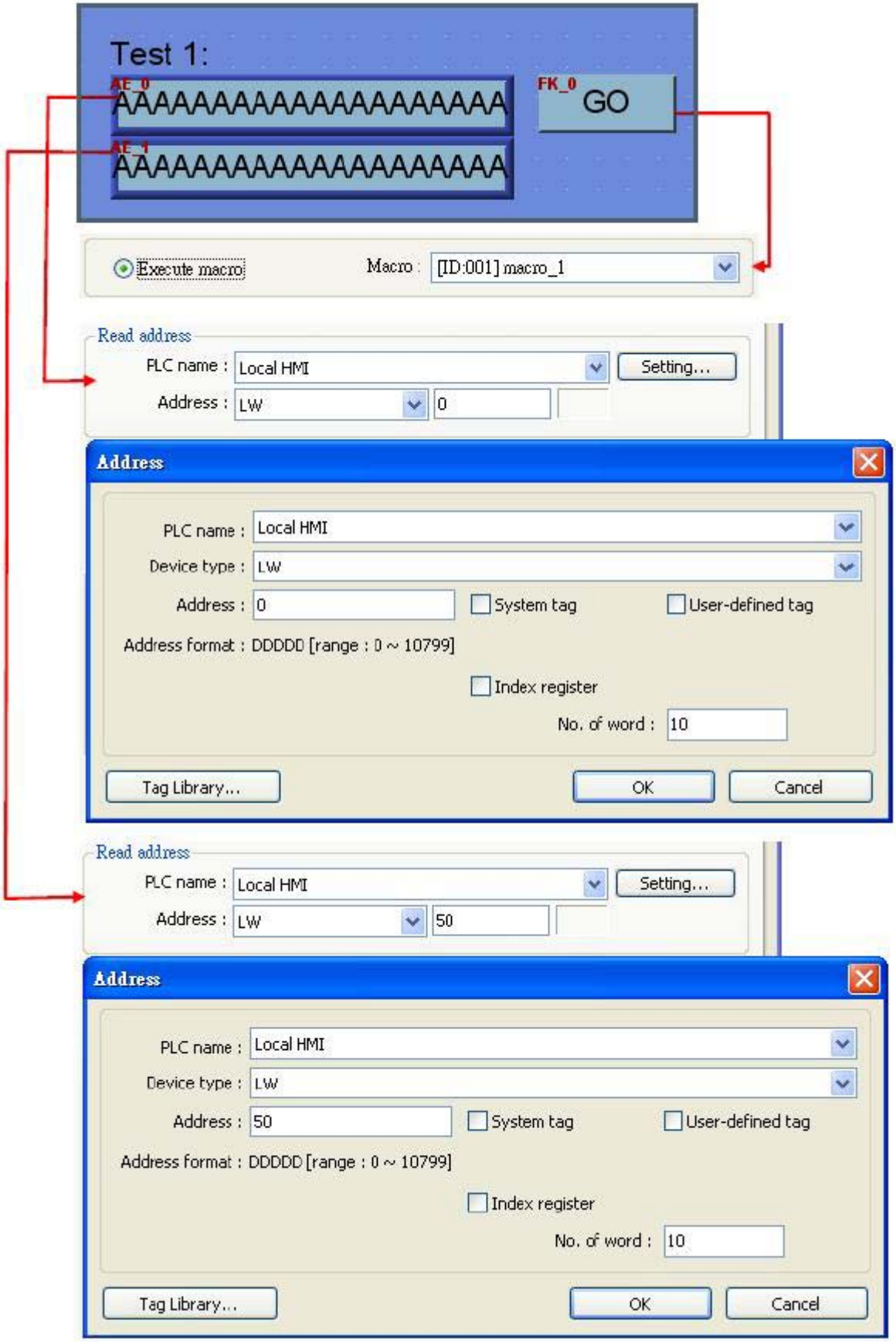
Создайте новый *macro*:

Отредактируйти содержание:



Первая функция “StringGet” использована для чтения строк LW0~LW19, и сохранения их в массиве *str* . Вторая функция “StringSet” использована для вывода содержимого массива *str* .

Добавьте объекты «ASCII Input » и «Function Key » в окне 10 проекта. Настройки этих объектов показаны далее. Объект «Function Key» использован для запуска *macro_1.*


Test 1:
AE_0
AAAAAAAAAAAAAAAAAAAAA
AE_1
AAAAAAAAAAAAAAAAAAAAA
FK_0
GO
Execute macro
Macro :
[ID:001] macro_1
Read address
PLC name :
Local HMI
Setting...
Address :
LW
0
Address
PLC name :
Local HMI
Device type :
LW
Address :
0
System tag
User-defined tag
Address format : DDDDD [range : 0 ~ 10799]
Index register
No. of word :
10
Tag Library...
OK
Cancel
Read address
PLC name :
Local HMI
Setting...
Address :
LW
50
Address
PLC name :
Local HMI
Device type :
LW
Address :
50
System tag
User-defined tag
Address format : DDDDD [range : 0 ~ 10799]
Index register
No. of word :
10
Tag Library...
OK
Cancel

Затем, скомпилируйте проект и включите режим *Off-line* или *On-line* имитацию.

Выполните следующие шаги для работы с исполняемым проектом:

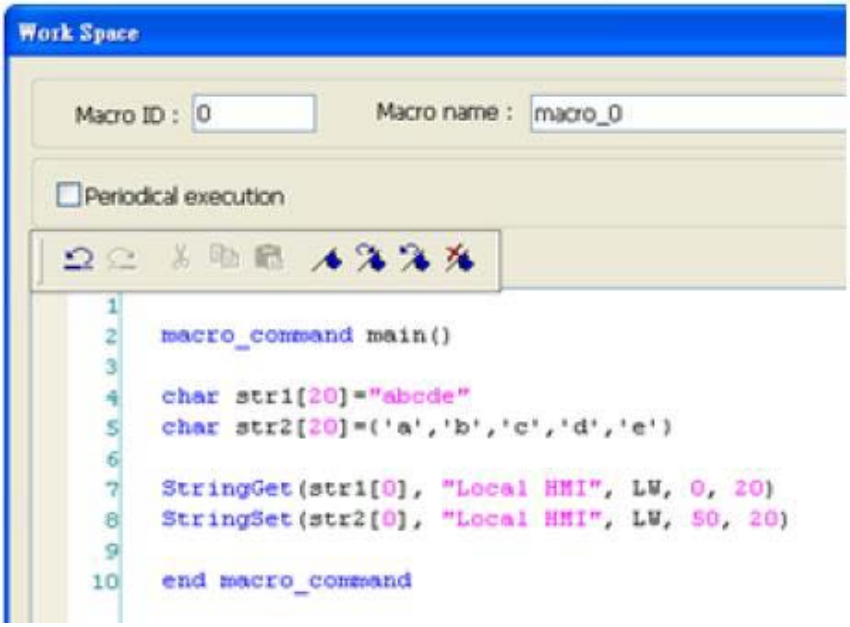
ШАГ 1: Введите строку

ШАГ 2: Нажмите кнопку «GO» - Выполнить.

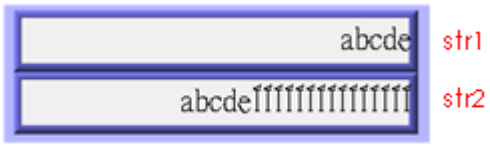
ШАГ 3: Просмотрите строку выхода

2. Инициализация строки.

Создайте новый макрос и отредактируйте содержание:

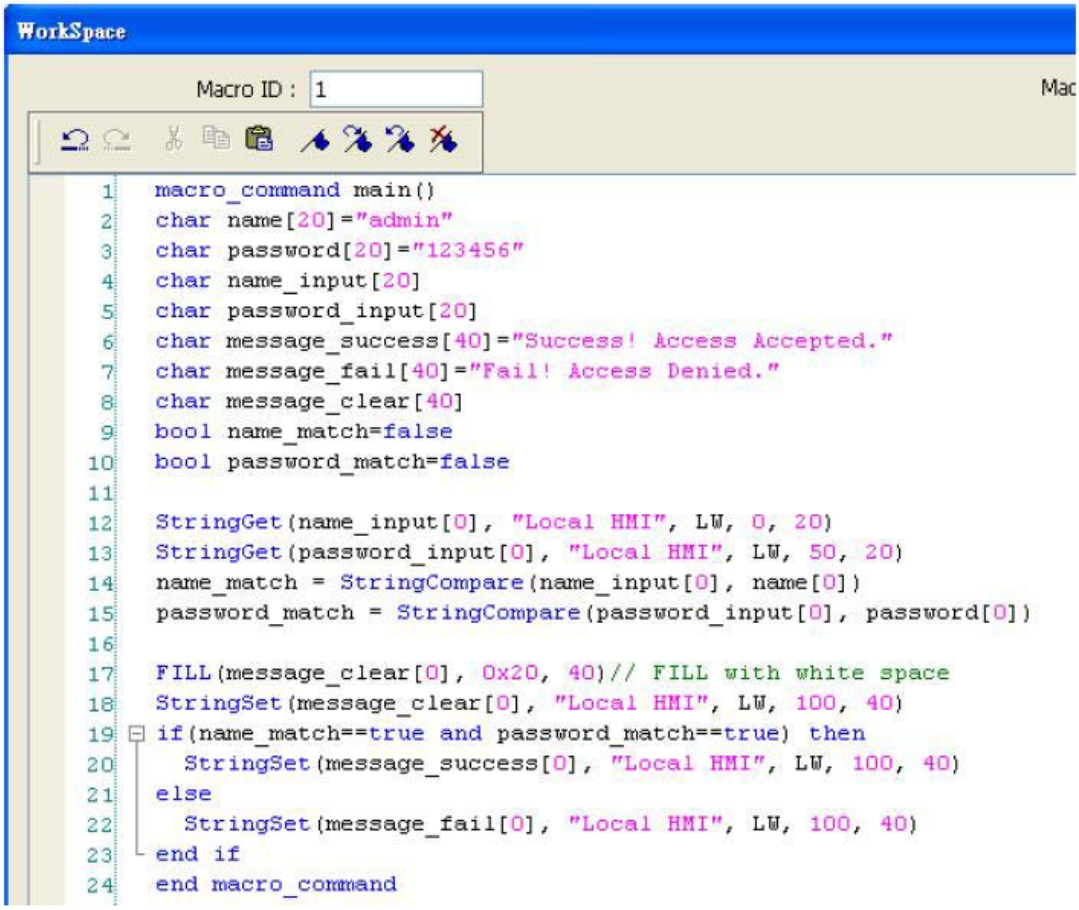


Данные заключенные в двойные кавычки (“”) просматриваются, как строки. *str1* инициализироана, как строка, а *str2* инициализирована, как массив символов (char array).
На снимке экрана имитации показано различие между *str1* и *str2,* использующих два объекта «ASCII input»:



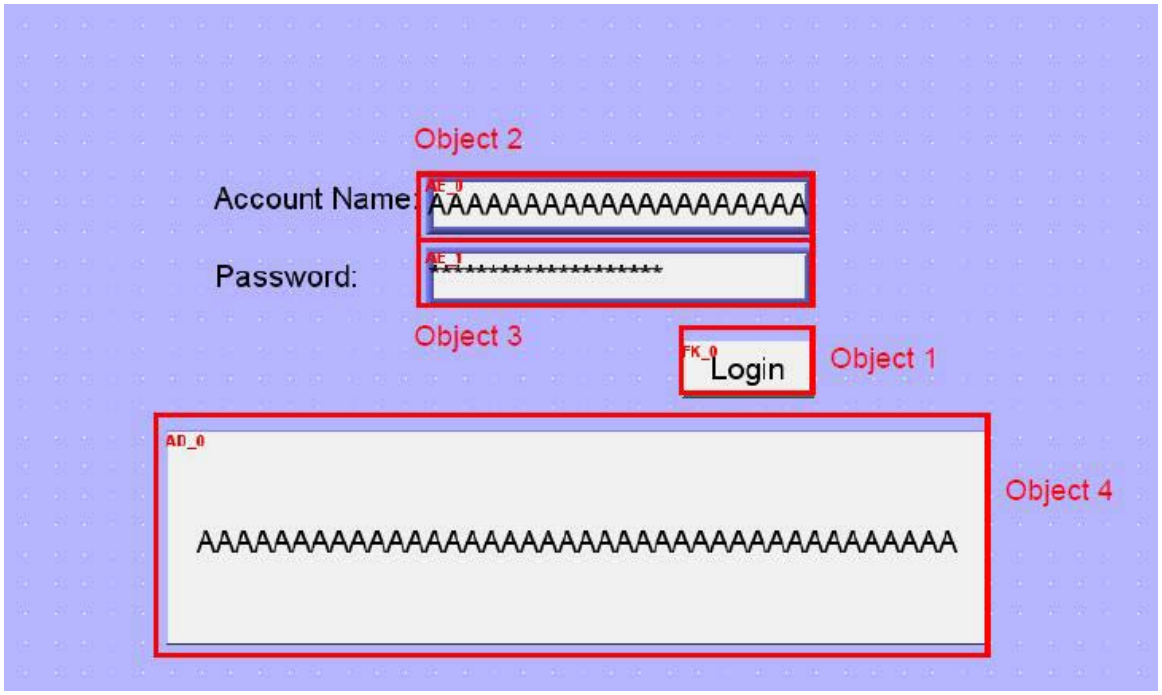
Макро компилятор будет добавлять символы Нуля (‘\0’) в конец строки.
Функция “StringSet” будет посылать каждый символ из *str1* в регистры до появления символа Нуля. Лишние символы, следующие за Нулем будут проигнорированы, даже если число данных настроено большим, чем длина строки.
Наоборот, макро компилятор не будет добавлятьсимволы Нуля (‘\0’) в конец массива символов. Действительное число символов в *str2,* посланных в регистры, зависит от значения числа данных (value of data count), которые прошли функцию “StringSet”.

3. Простая страница регистрации - login page.
Создайте новый макрос и отредактируйте содержание:



Первые две функции “StringGet” будут считывать строки, вводимые пользователем и хранить их в массивах, названных *name_input* и *password_input* отдельно. Используйте функцию “StringCompare” , чтобы проверять совпадение вводимого «*account name*» и пароля. Если *account name* совпадает, *name_match* становится истинным; если пароль совпадает, *password_match* становится истинным. Если *name*_совпадает и *password_match* истинно, появляется строка на выходе: “Success! Access Accepted.”. Иначе строка на выходе будет “Fail! Access Denied.” – Отказ. В доступе отказано .

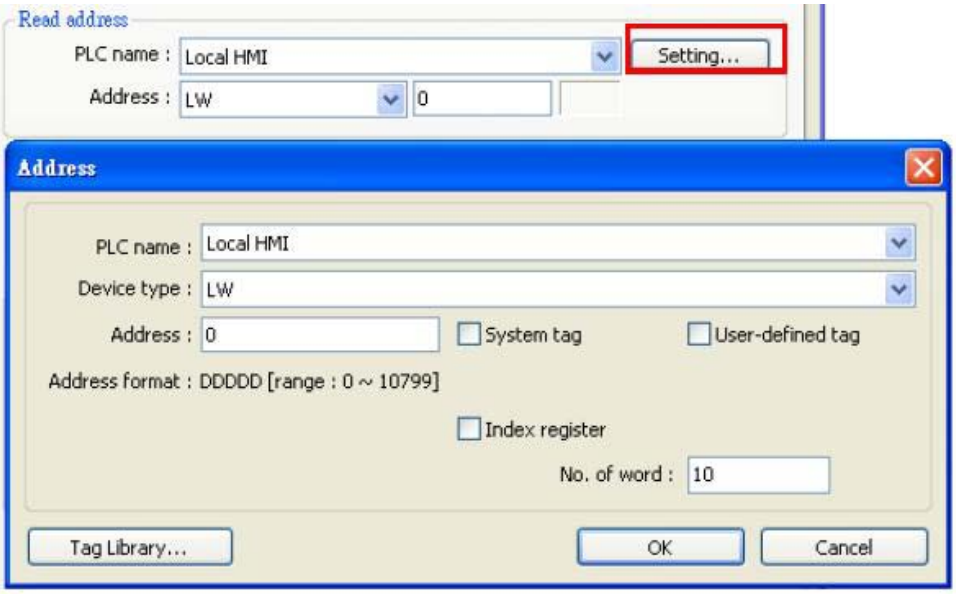
Добавьте объекты «ASCII Input » и «Function Key » в окно10 проекта. Настройки этих объектов приведены далее. Объект «Function Key» предназначен для запуска *macro_1*.



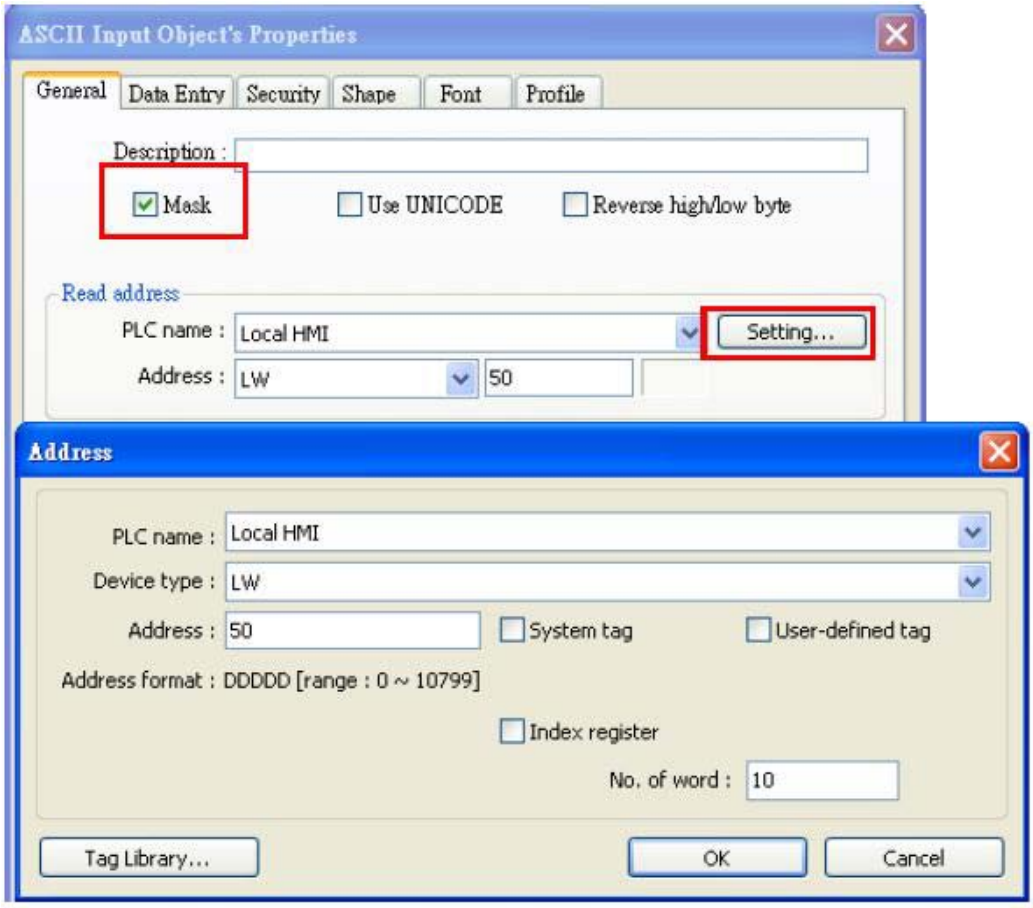
Настройки объектов:

Объект 1: Function Key

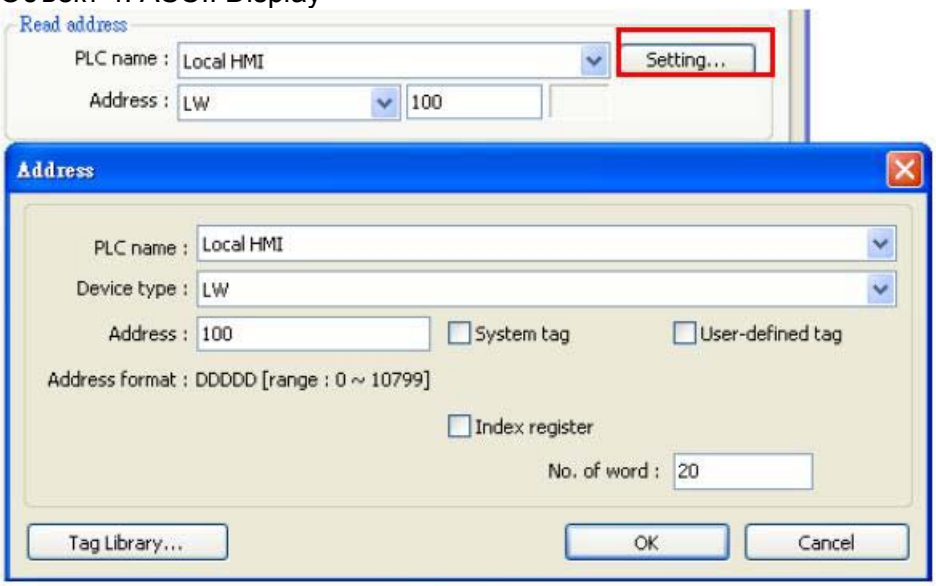
Объект 2: ASCII Input



Объект 3: ASCII Input

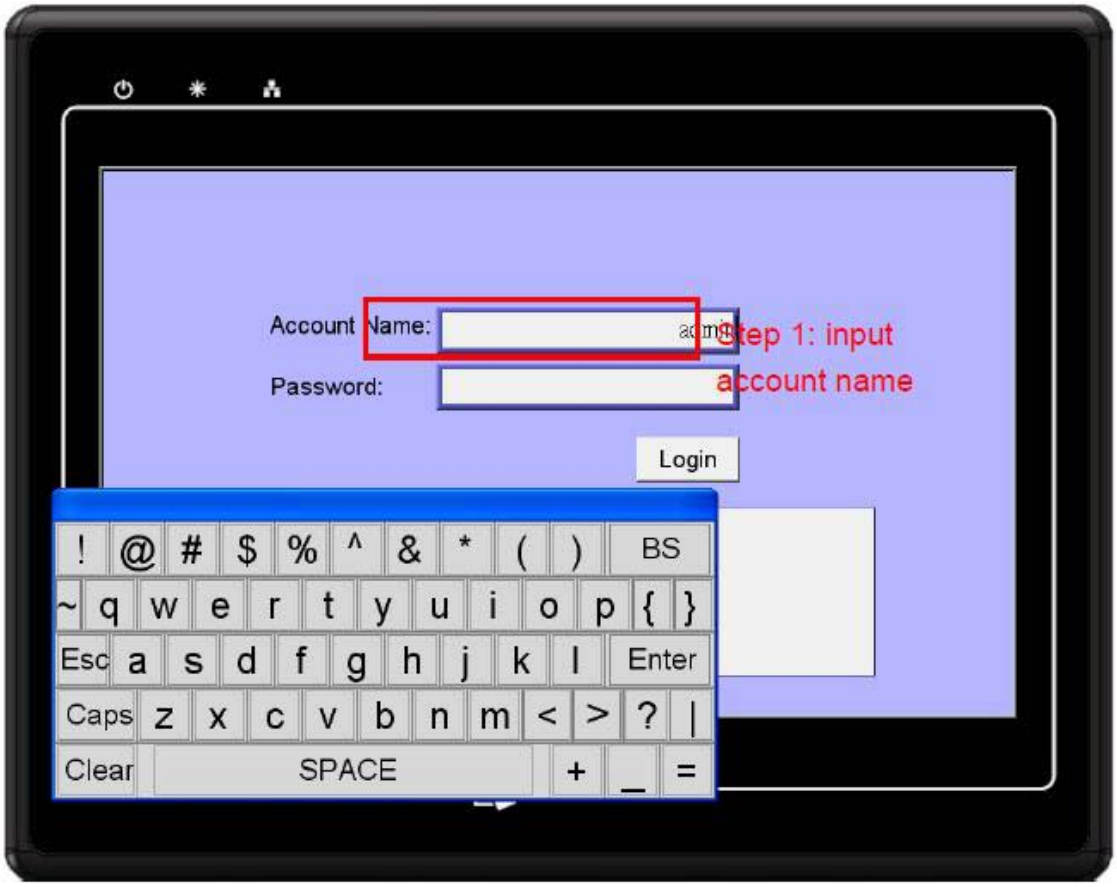


Объект 4: ASCII Display

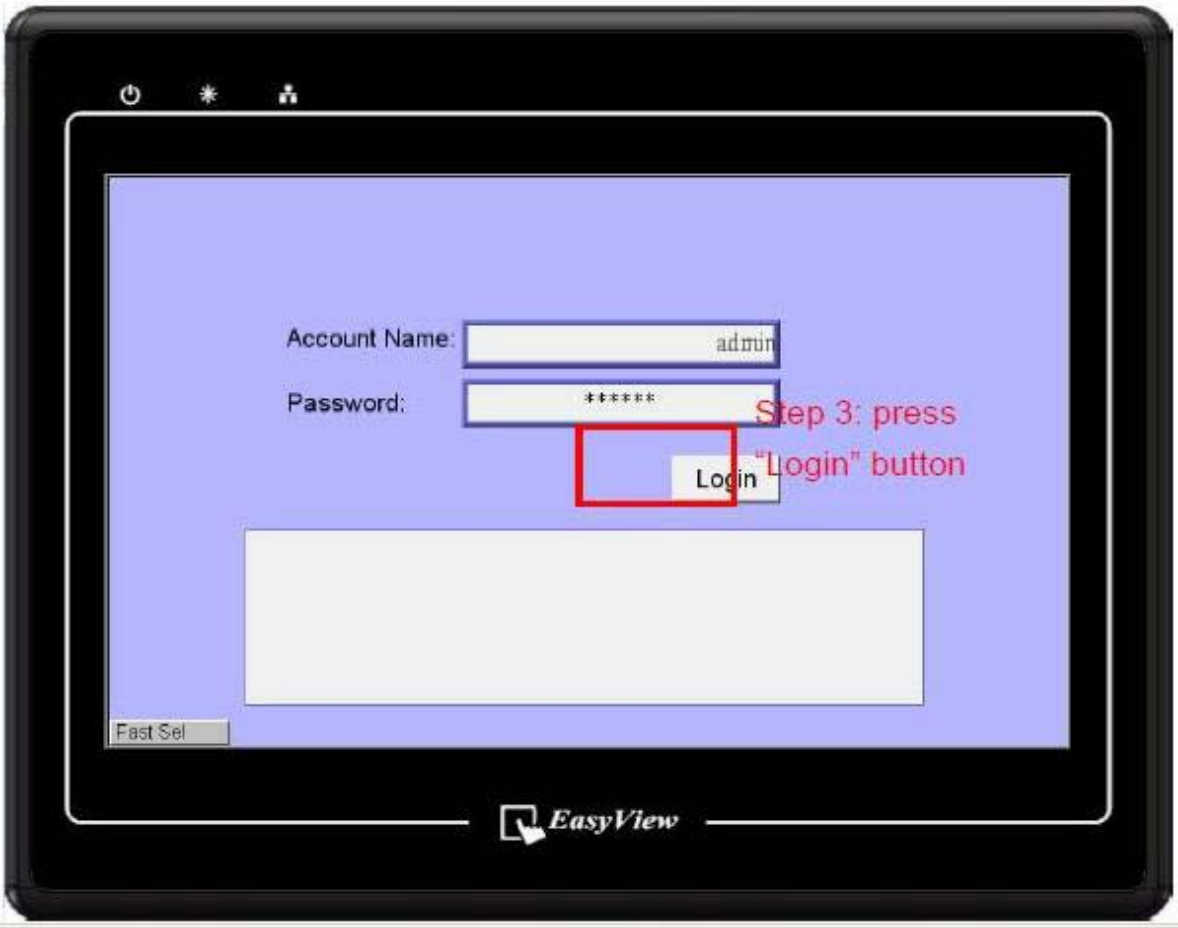


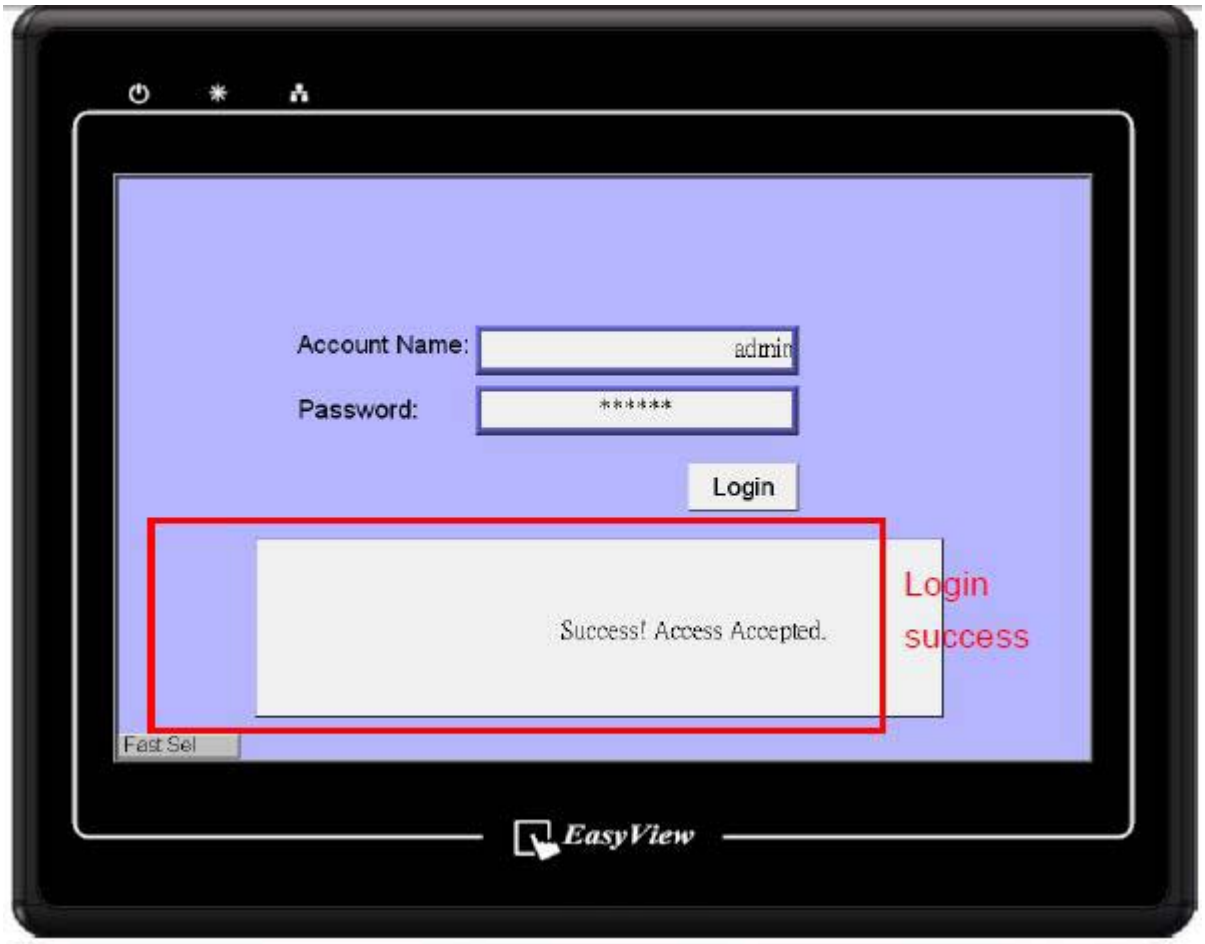
Затем, откомпилируйте проект и выполните Off-line или On-line имитацию.

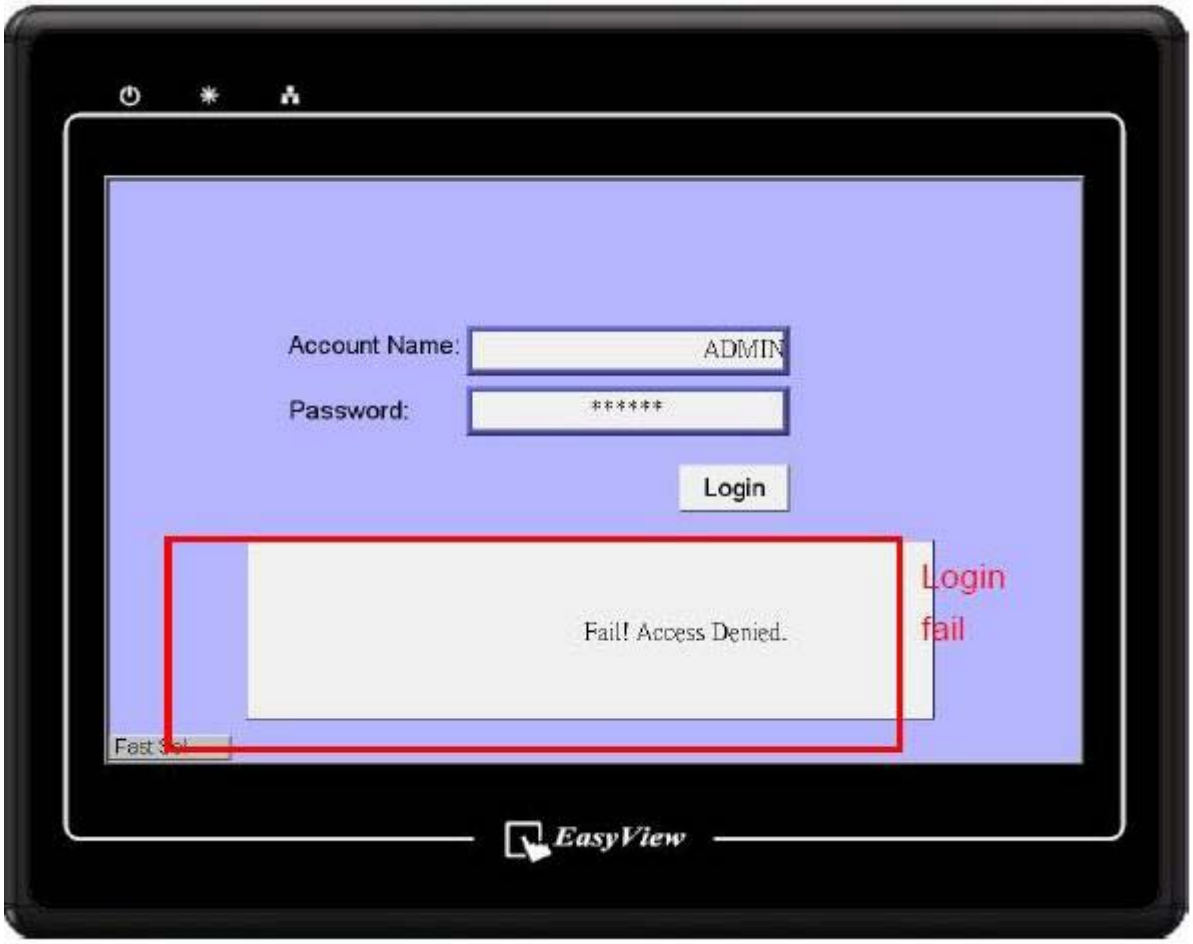
Выполните следующие шаги взаимодействия с проектом:



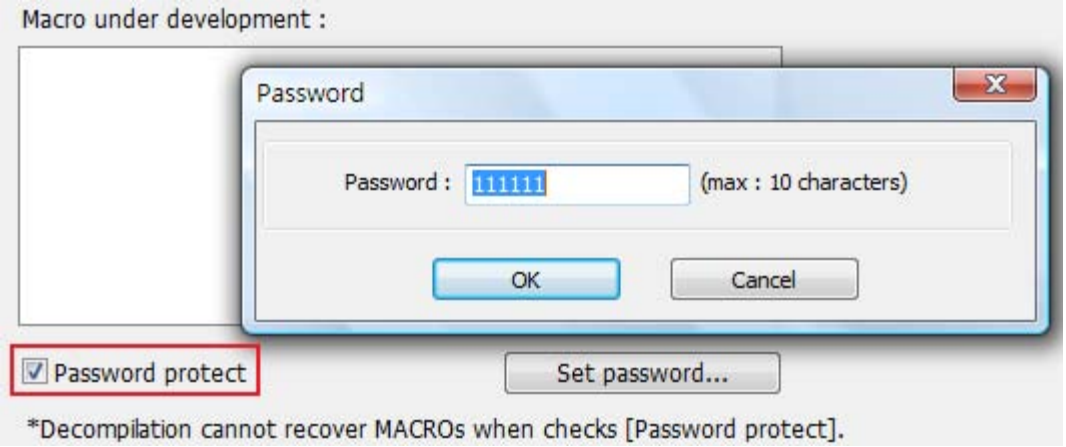
Account Name:
admin
Password:
******
Step 2: input
password
Login
1 2 3 4 5 6 7 8 9 0 BS
` Q W E R T Y U I O P [ ]
Esc A S D F G H J K L Enter
Caps Z X C V B N M , . / \
Clear SPACE + - *


Account Name:
admin
Password:
******
Step 3: press
"Login" button
Login
Fast Sel
EasyView


Account Name:
admin
Password:
******
Login
Success! Access Accepted.
Login
success
Fast Sel
EasyView

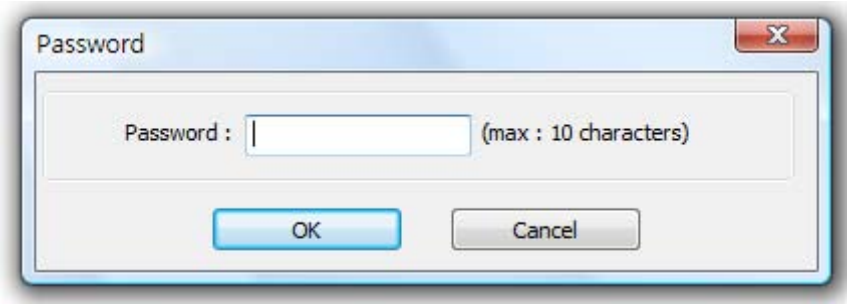

Account Name:
ADMIN
Password:
******
Login
Fail! Access Denied.
Login
fail
EasyView

18.15 Защита макросов паролем

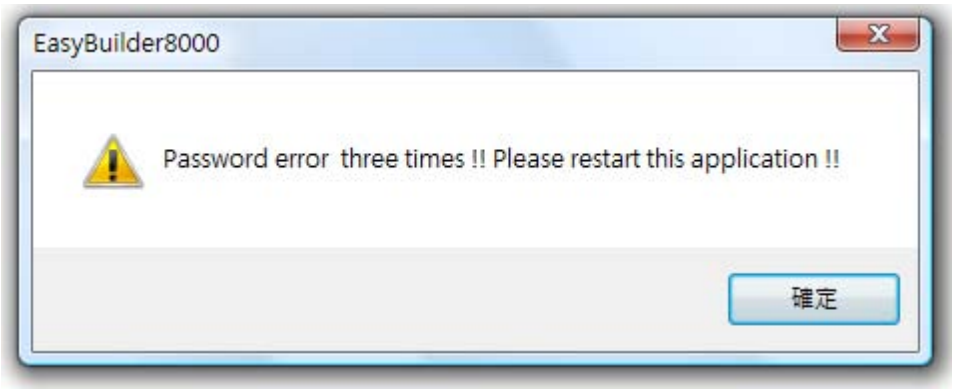


В окне редактора MACRO есть поле выбора [Password protect] , пометьте его и нажмите [Set password…], чтобы установить пароль состоящий только из 10 или менее символов ASCII , например. “a$#*hFds”.

После установки MACRO пароля, пользователь должен вводить корректый парольдля открытия окна редактирования MACRO.



EasyBuilder Pro будет перезагружен для ввода пароля снова, после 3-х неудачных попыток ввода пароля.

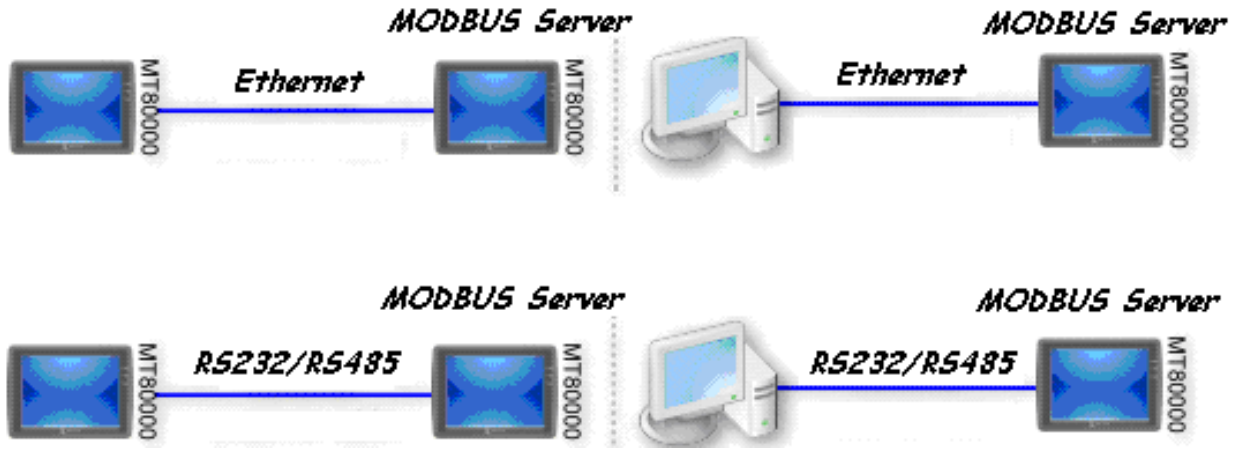


[Внимание] Когда макрос защищен паролем, декомпиляция файла «XOB» не восстановит содержимое макроса.

Глава 19. Настройка панели как MODBUS-сервера

19.1 Настройка панели, как устройства MODBUS

После настройки MODBUS-сервера данные с панели можно считывать или записать по протоколу MODBUS.



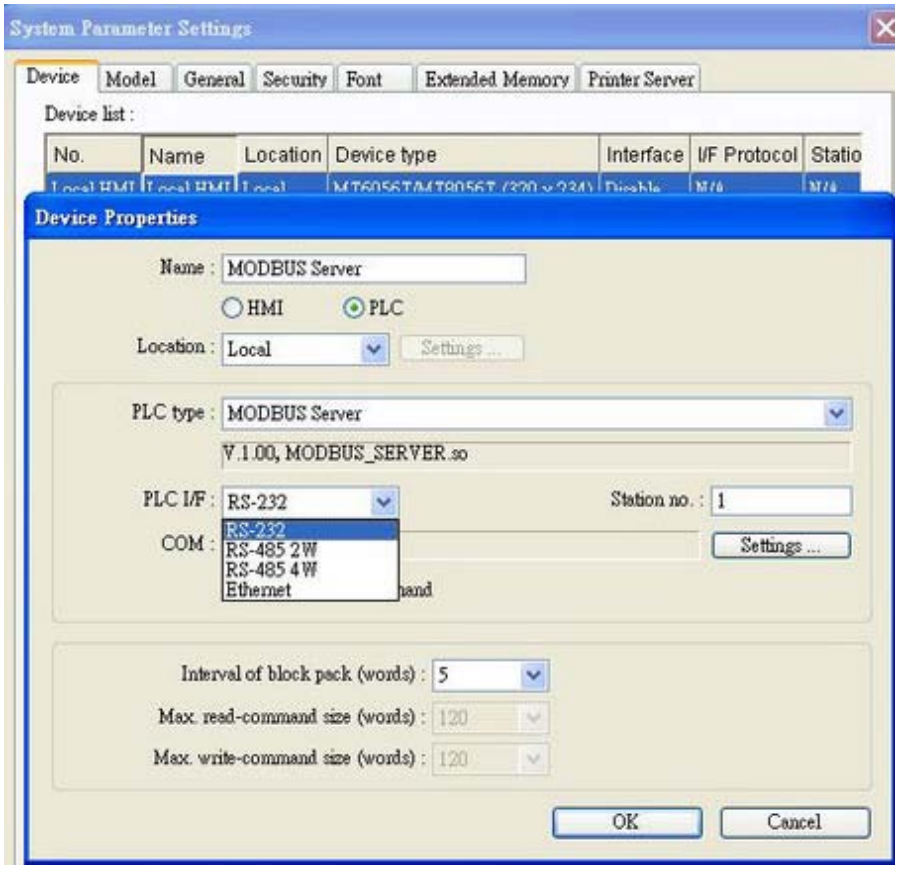
На рисунке выше показано, что панель настроена в качестве MODBUS-сервера.

Панель оператора или компьютер могут считывать или записывать данные в панель, используя протокол MODBUS, по сети Ethernet или интерфейсу RS232/485.

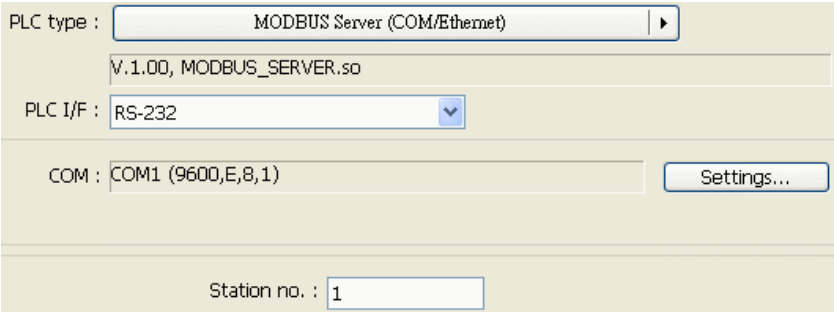
См. описанные ниже шаги.

19.1.1 Создание MODBUS-сервера

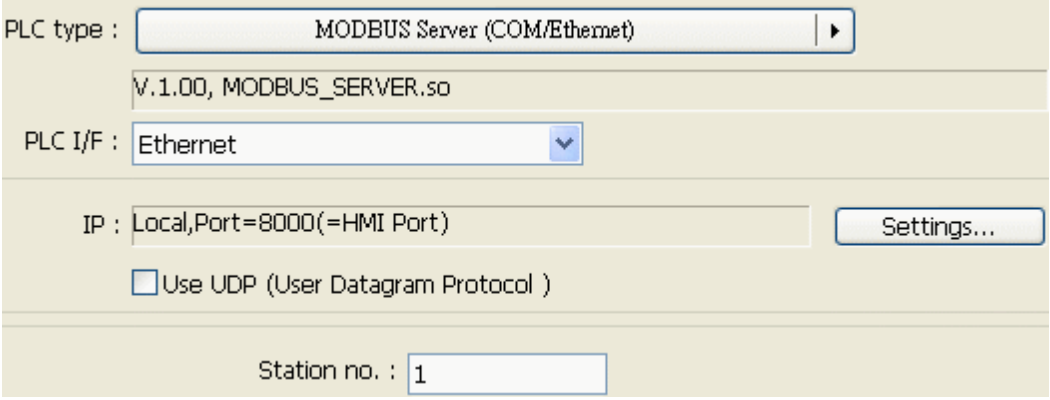
Сначала необходимо добавить новое устройство “MODBUS Server” в списке устройств **[Device]** окна **[System Parameter Setting],** для параметра **[PLC I/F] – Интерфейс ПЛК** может выбрано любое значение: RS232, RS485 2W (2-х проводный), RS485 4W (4-х проводный), Ethernet.



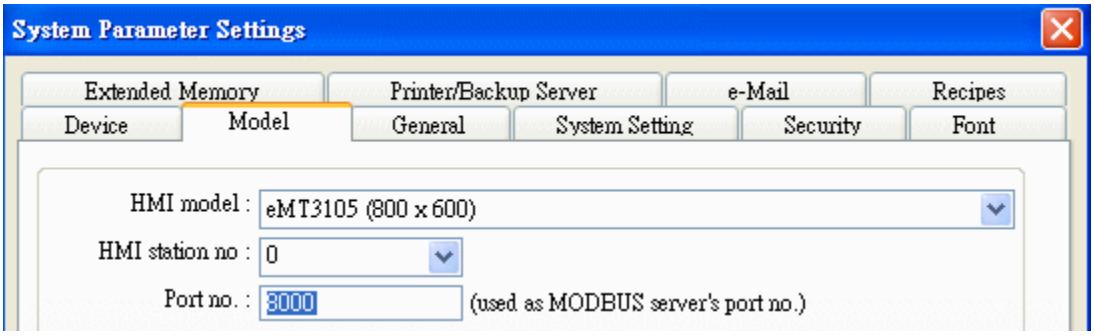
Если выбраны интерфейсы **RS232** или **RS485**, необходимо задать корректные настройки COM-порта **[COM Port Settings]** (COM 1~COM3), как показано ниже. Номер станции MODBUS Server –установить «1».



Если выбран интерфейс **Ethernet,** настройте **[IP address] ,** как показано ниже.

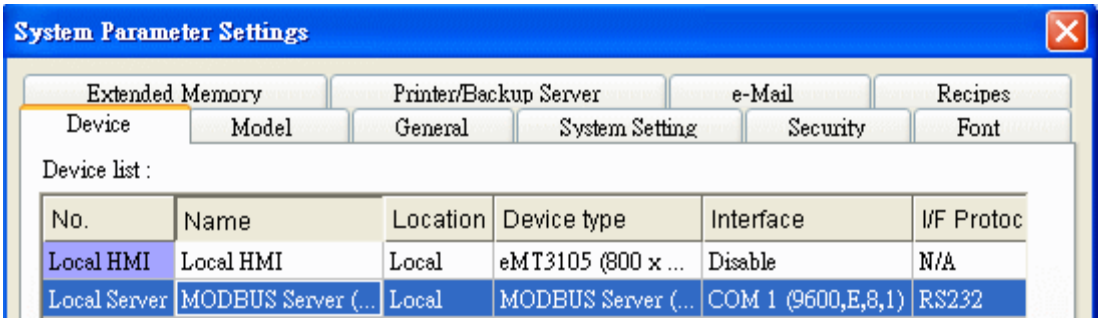


Для MODBUS-сервера и панели оператора(HMI) используется один и тот же номер порта. Изменить номер порта MODBUS-сервера можно на вкладке [Model] окна [System Parameter Settings].



После завершения всех настроек MODBUS-сервер появится в списке устройств на вкладке **[Device].**

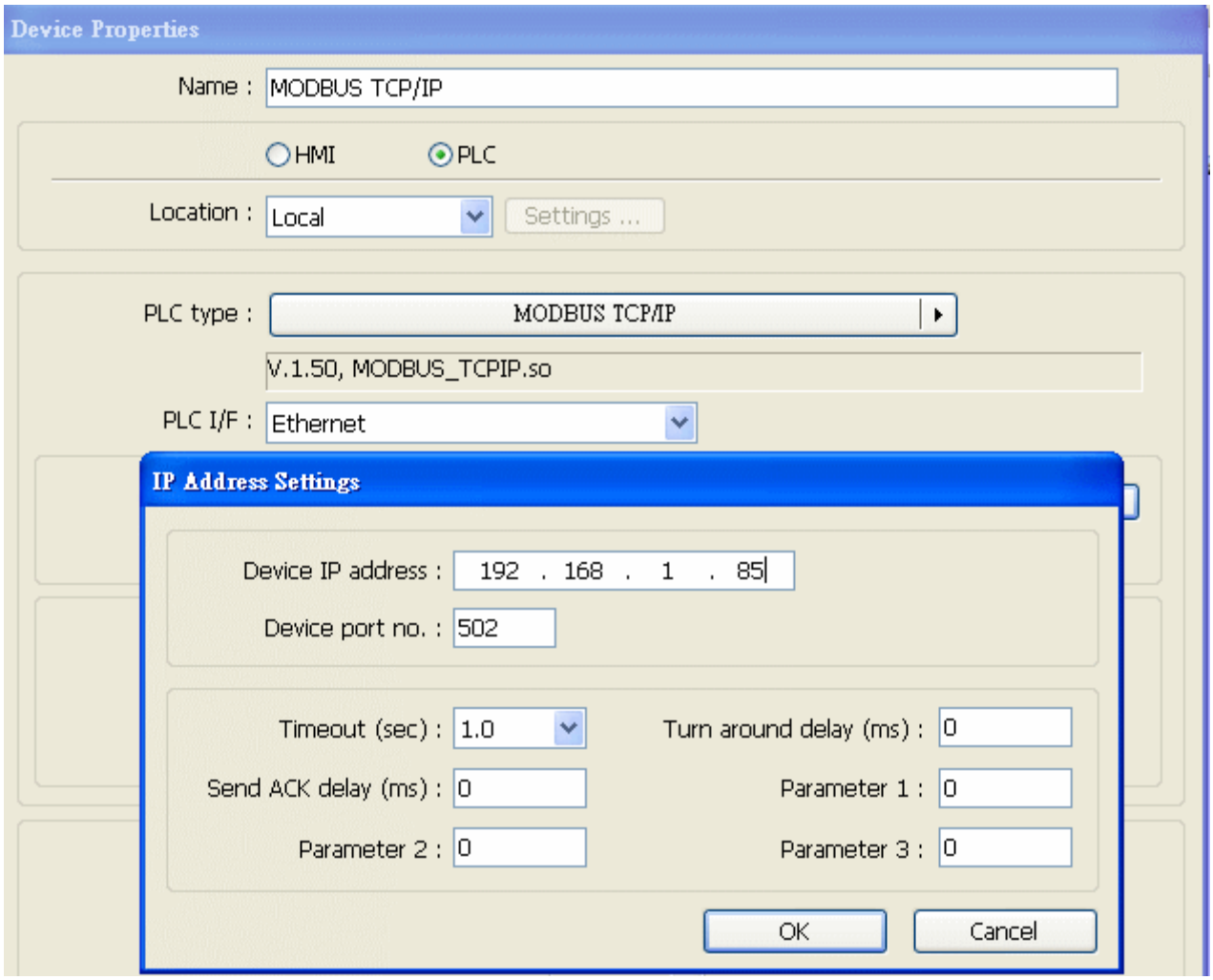
Вы можете послать команду протокола MODBUS для чтения или записи информации в MODBUS-сервере после загрузки файла «.xob» в панель оператора.



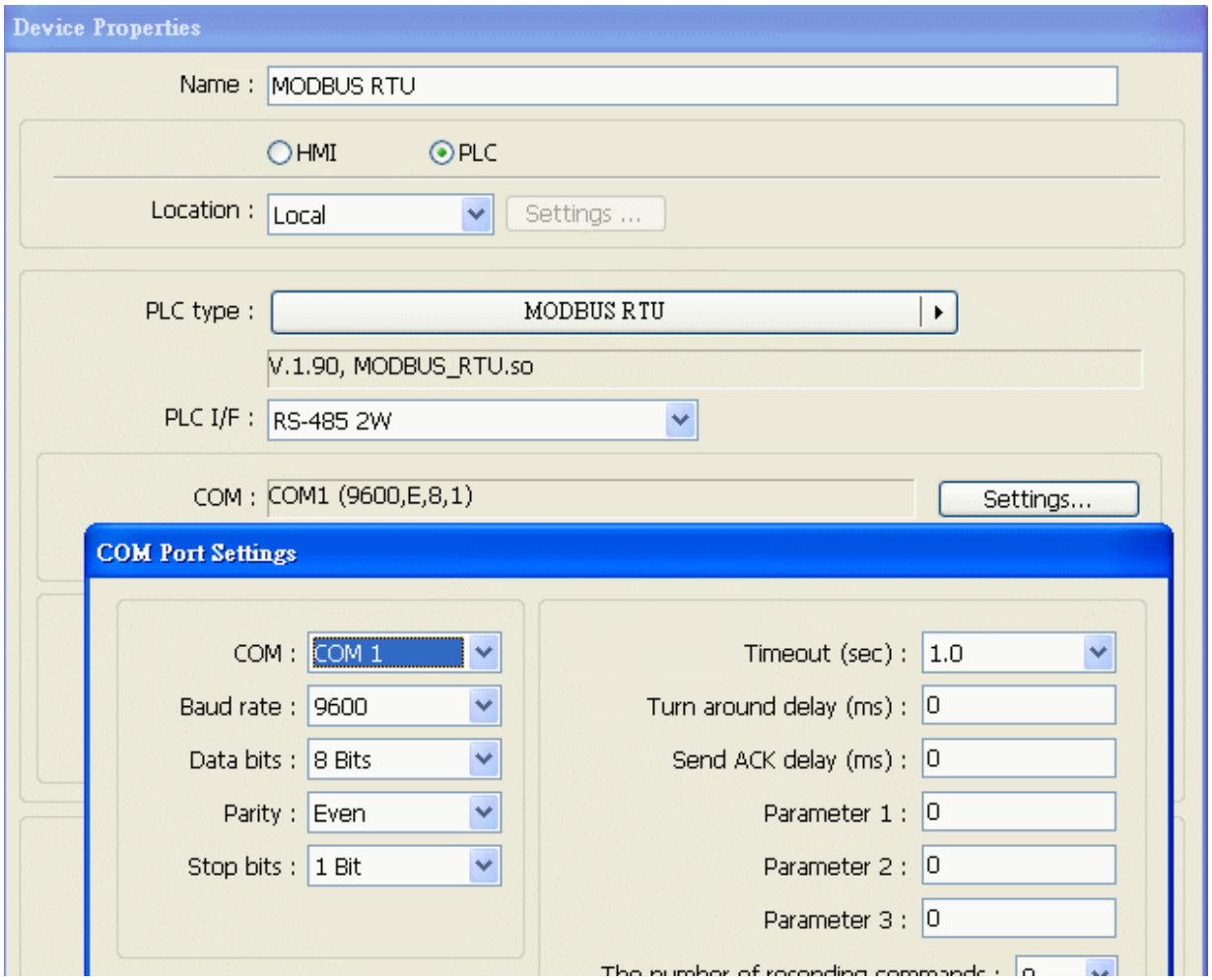
19.1.2 Чтение/запись данных в MODBUS-сервер

Панель (панель является клиентом) может считать/записать данные в другой панели (сервере) по протоколу MODBUS.

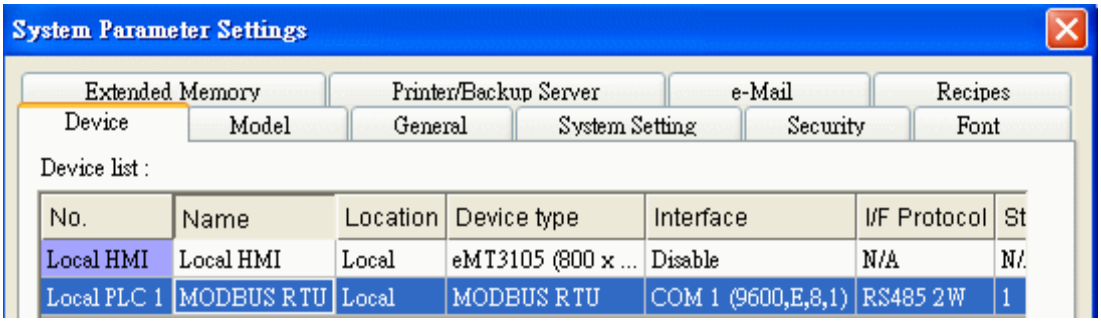
Добавьте новое устройство - клиента. Если устанавливается соединение по **Ethernet**, выберите вариант ”MODBUS RTU TCP/IP” в качестве **[PLC type]** и введите корректный IP-адрес и номер порта **[Port no.]**.



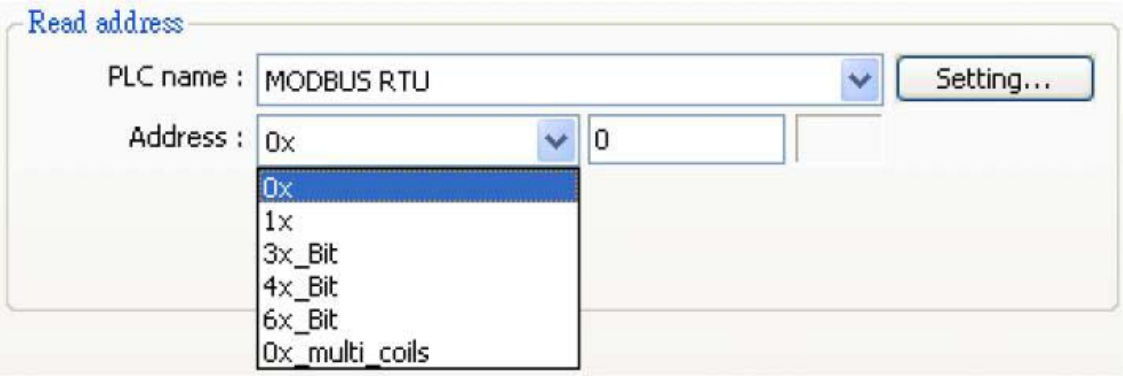
Если же используется соединение по интерфейсу **RS232/485**, необходимо установить **[PLC type]** - ”MODBUS RTU”. Убедитесь, что параметры соединения установлены правильно.



После установок нажмите OK, новое устройства ”MODBUS RTU” появится в списке устройств на вкладке [Device].



На странице настроек каждого объекта в поле **[PLC name]** есть вариант “MODBUS RTU”, можно выбрать соответствующий тип адреса (device type) и адрес.



Так как сервер это панель, соответствие адресов чтения/записи следующее:

чтение/запись 0x/1x(1~9999)	для чтения/записи LB(0~9998)
чтение/запись 3x/4x/5x(1~9999)	для чтения/записи LW(0~9998)
чтение/запись 3x/4x/5x(10000~75533)	для чтения/записи RW(0~65533)

19.2 Изменение номера станции MODBUS-сервера в процессе работы

Измените содержание соответствующих служебных регистров для изменения номера станции MODBUS-сервера (панели).

[LW9541] Номер станции MODBUS-сервера (COM 1)
[LW9542] Номер станции MODBUS-сервера (COM 2)
[LW9543] Номер станции MODBUS-сервера (COM 3)
[LW-9544] Номер станции MODBUS-сервера (Ethernet)

19.3 Типы адресов MODBUS

Типы адресов протокола MODBUS в EasyBuilder Pro следующие:
0x, 1x, 3x, 4x, 5x, 6x, 3x_bit и 4x_bit.

Функциональные коды Modbus RTU :

0x	0x01 Read coil	0x05 запись одного реле (single coil)
0x_multi_coils	0x01 Read coil	0x0f запись нескольких реле
1x	0x02 Read discrete input	нет - для операции записи
3x	0x04 Read input register	нет - для операции записи
4x	0x03 Read holding register	0x10 запись нескольких регистров
5x	0x03 Read holding register	0x10
6x	0x03 Read holding register	0x06 запись одного регистра
3x_bit	0x04 Read input register	нет - для операции записи
4X_bit	0x03 Read holding register	0x10 запись нескольких регистров

ПРИМЕЧАНИЯ:

1. Адреса типа “5x” отображаются в «Holding Register». Коммуникационный протокол “5x” почти тот же самый, что “4x”, за исключением того, что “5x” делает замену мест двойных слов (double word swap).

Если регистр 4x содержит следующую информацию:

Address: 1 2 3 4 5 6 ...

Data in word: 0x1 0x2 0x3 0x4 0x5 0x6

Data: 0x20001 0x40003 0x60005

Для регистра 5x, это становится:

Address: 1 2 3 4 5 6 ...

Data in word: 0x2 0x1 0x4 0x3 0x6 0x5

Data: 0x10002 0x30004 0x50006

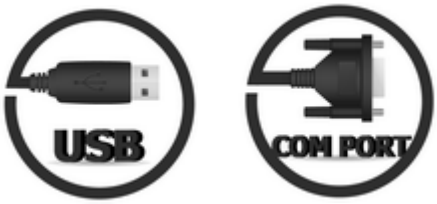
2. Тип адреса “6x” ограничен использованием только одного слова данных.

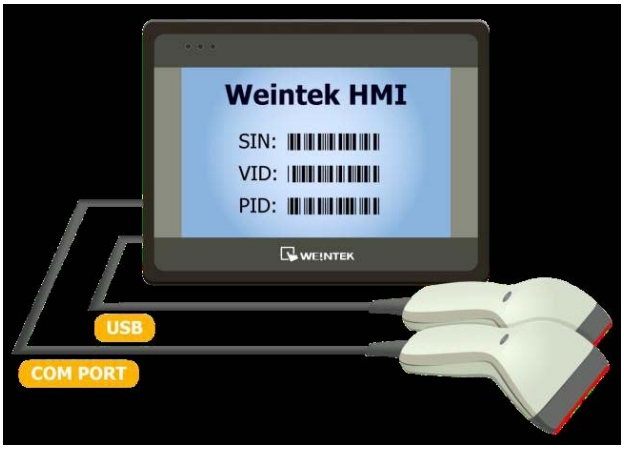
-

3. Коммуникационный протокол 3x_bit и 4x_bit тот же самый, что и 3x и 4x.

Различие состоит в том, что протоколы 3x_bit и 4x_bit читают единственный бит из целого слова.

Глава 20. Подключение устройства считывания штрих-кода



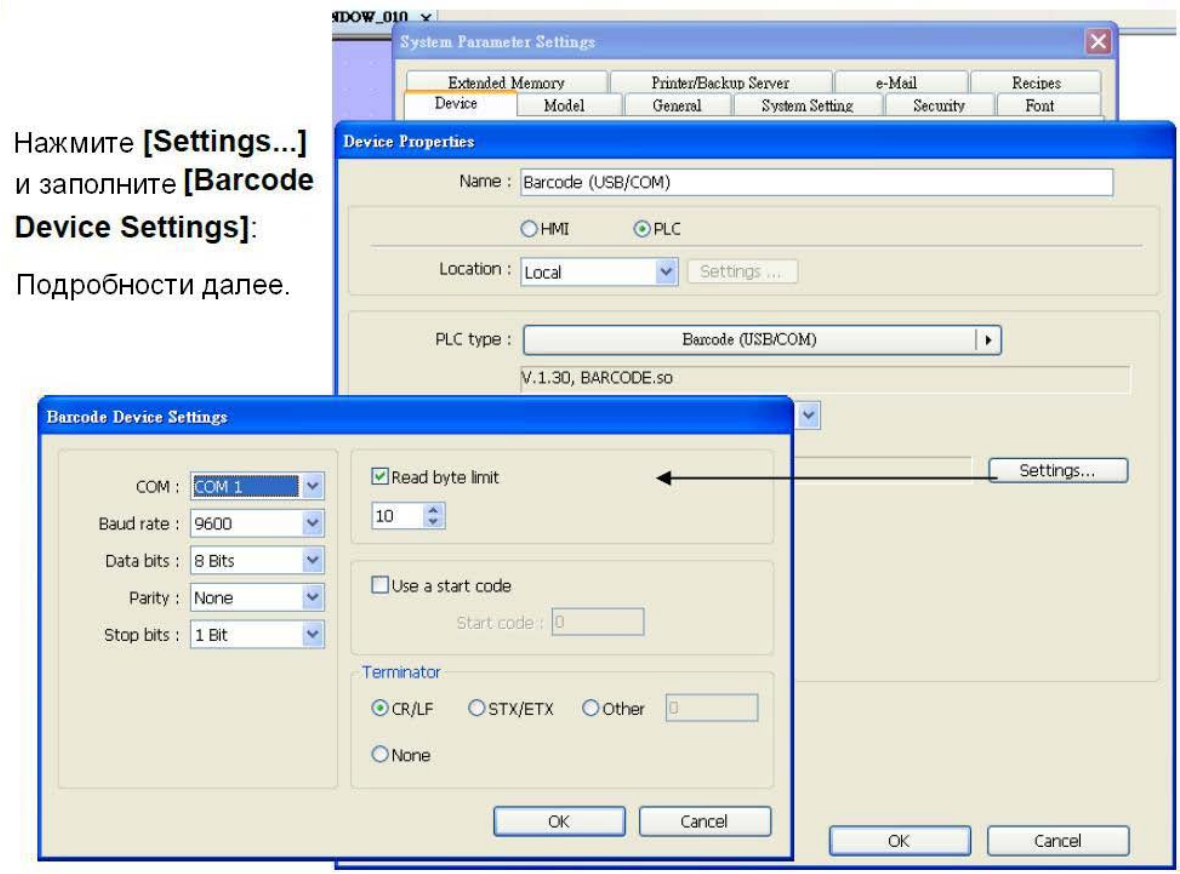


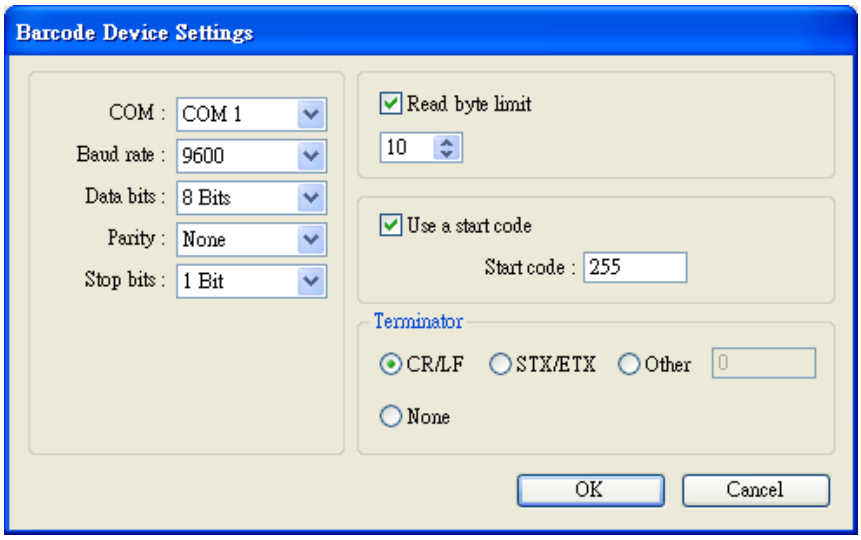
Интерфейс устройства чтения штрих-кода(Barcode Device)

20.1 Подключение устройства чтения штрих-кода

Панели Weintek поддерживают соединене с баркодерами по USB/COM.

Сначала необходимо добавить новое устройство чтения штрих-кода в список устройств: **[Edit]/ [System Parameter Settings]/ [Device list]** , как показано ниже.





[COM]、[Baud rate]、[Data bits]、[Parity]、[Stop bits]

Устройство считывания может быть подключено через порты COM1~COM3 или USB. При использовании COM-портов установите настройки соответствующие устройству. При соединении по USB никакие настройки не нужны.

[Read byte limit]

Эта функция ограничивает число байт для чтения. Диапазон значений: от 10 до 512.

Например:

Если **[Read byte limit]** = 10, а устройство чтения сгенерирует данные: “0x34 0x39 0x31 0x32 0x30 0x30 0x34 0x37 0x30 0x38 0x33 0x38”(12 байт) .

Но прочитаны будут, в этом случае, только первые 10 байт: “0x34 0x39 0x31 0x32 0x30 0x30 0x34 0x37 0x30 0x38”.

[Use a start code]

С помощью этой функции панель оператора распознает начальные данные полученные устройством считывания штрих-кода с легальным стартовым кодом . Иначе, данные будут уигнорироваться. Вся информация после начального кода будет сохранена по указанному адресу. Введите децимальное значение символа ASCII.

Например: если начальный код 255 (0xff), исходные данные “0xff 0x34 0x39 0x31 0x32 0x30 0x30 0x34 0x37”, то по указанному адресу будет сохранена следующая информация: “0x34 0x39 0x31 0x32 0x30 0x30 0x34 0x37”.

[Terminator]

Это конечный ограничитель, который сигнализирует о конце пакета данных, когда он распознан, это означает конец потока данных.

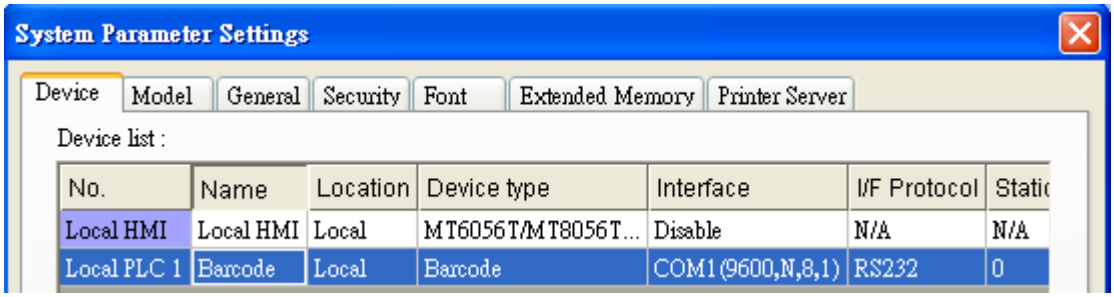
[CR/LF] 0x0a или 0x0d означает конец данных.

[STX/ETX] 0x02 или 0x03 означает конец данных.

[Other] Можно назначить конечный ограничитель вручную.

[None] Панель сохранит все данные в назначенный адрес баркодера.

После завершения настроек в списке устройств **[Device list]**. появится новое устройство *Barcode*.



Теперь устройство считывания штрих-кода может быть выбрано в **[PLC type]** в диалоговом окне настроек параметров. Устройство [Barcode] имеет два типа адресов (Flag и Barcode).

Тип адреса	Имя адреса	Описание
Bit	FLAG	FLAG 0 показывает состояние чтения данных. Когда чтение данных завершено, состояние FLAG 0 изменится из OFF в ON.
Word	BARCODE	BARCODE 0 Число байт считываемых данных BARCODE 1~n определяют адрес хранения данных с дешифратора штрих-кода.

Ниже показана настройка считывателя штрих-кода. Данные считанные баркодером: ”9421007480830”.

BARCODE 0 - это адрес объекта “Numeric Display”(bytes)

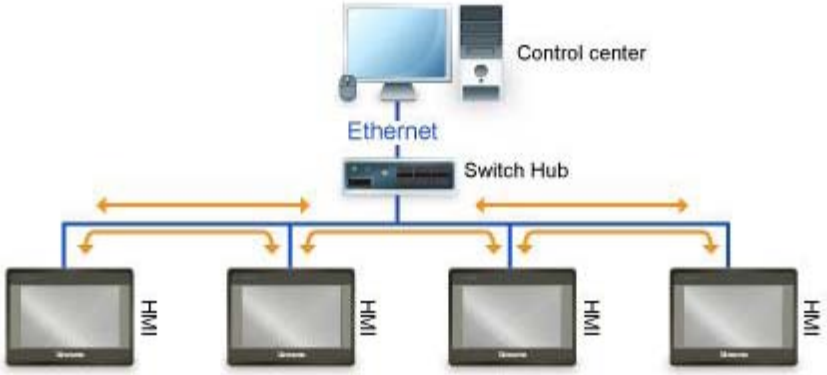
BARCODE 1~n - это адрес объекта “ASCII Display” (barcode).

Объекты показывают число принимаемых байт информации и сами данные.

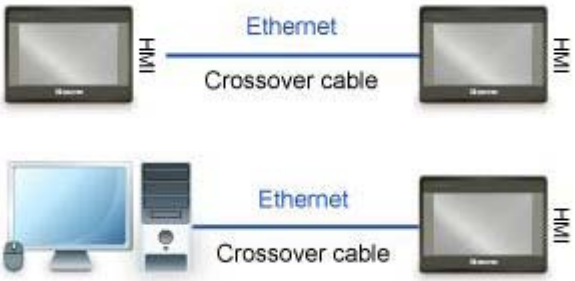
Глава 21. Соединение по Ethernet и соединение нескольких панелей

Существует два способа соединения по интерфейсу Ethernet:

1. Используя прямой кабель с разъемом RJ45 и концентратор (Hub),



2. Используя перекрестный кабель (crossover)с разъемом RJ45 и без концентратора. Этот способ предназначен для соединений типа «точка-точка» (т.е. например, панель оператора с другой панелью или ПК с панелью). Далее описано, как настраивать и осуществлять соединение по Ethernet в каждом случае.



Используя соединение по сети Ethernet, EasyBuilder Pro обеспечивает следующие способы передачи данных:

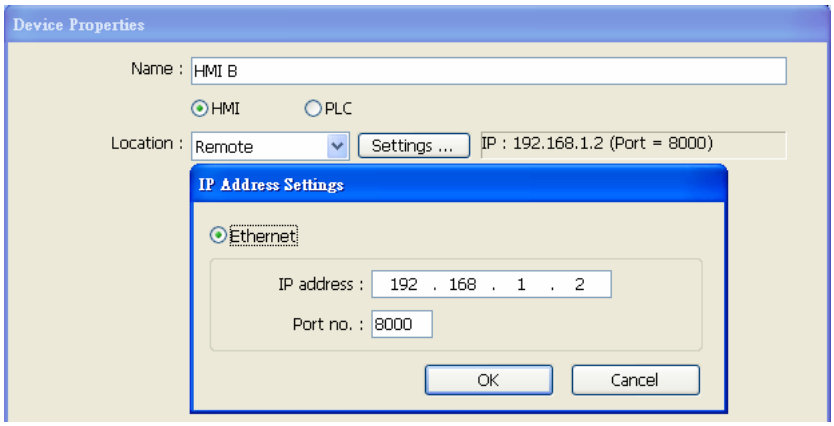
1. Соединение двух операторских панелей.
2. Соединение панели оператора с персональным компьютером.
3. Управление ПЛК соединным с другой панелью оператора.

21.1 Соединение двух панелей оператора

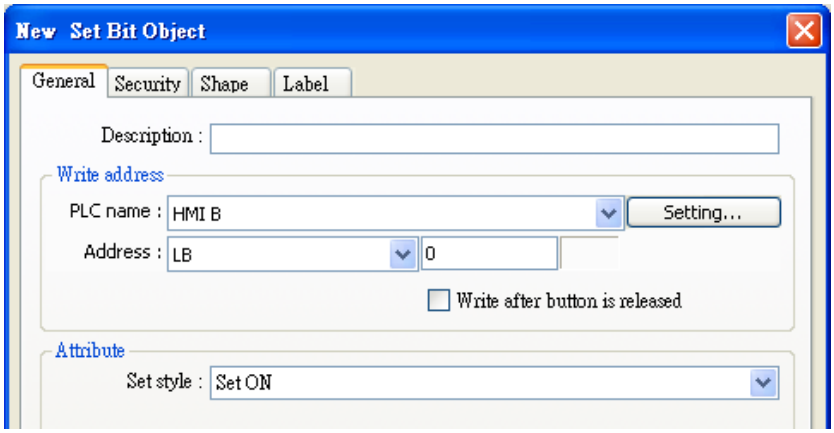
Ниже дан пример соединения двух панелей оператора, в котором панель HMI A использует объект [Set Bit] для управления битом [LB0] панели HMI B.

1. Установите IP-адреса обеих панелей. Например, HMI A = 192.168.1.1 и HMI B = 192.168.1.2.

2. Проект панели HMI A / **[System Parameter Settings] / [Device List].** Добавьте удаленную панель HMI IP192.168.1.2

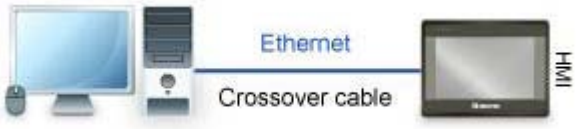


3. В окне [Set Bit Object] в поле [PLC name] введите “ HMI B”. Панель HMI A может работать с содержимым адреса LB0 удаленной панели HMI B



Note	Одна панель может обрабатывать одновременно запросы максимум от 64-х других панелей.

21.2 Соединение панели оператора с компьютером



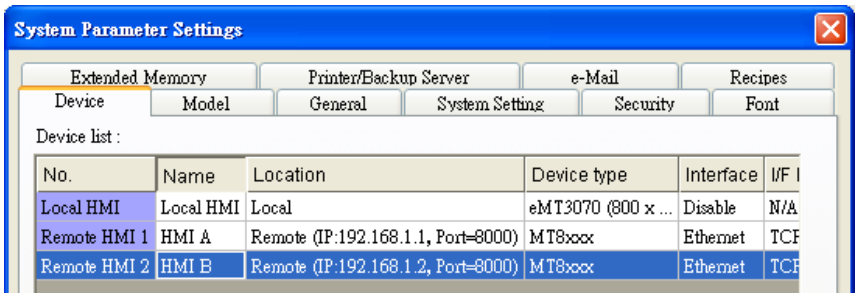
Используя функцию имитации (On-line Simulation), компьютер может собирать данные с панели оператора по сети Ethernet и сохранять их в файлах.

ПК может управлять панелью при помощи ее системных регистров (LB или LW).
Панель оператора, также, может управлять работой компьютера, например, делать запросы на сохранение данных из панели или ПЛК.

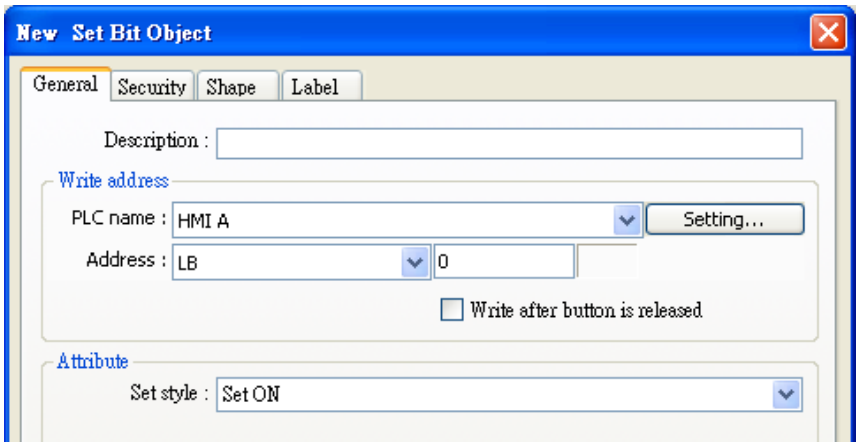
Предположим, что компьютер соединен с двумя панелями (HMI A и HMI B).
Последовательность настроек в проекте компьютер следующая:

1. Установите IP-адреса обеих панелей. Например, HMI A = 192.168.1.1 и HMI B = 192.168.1.2.

2. Проект на компьютере / **[System Parameter Settings] / [Device List].** Добавьте удаленные панели HMI A и HMI B

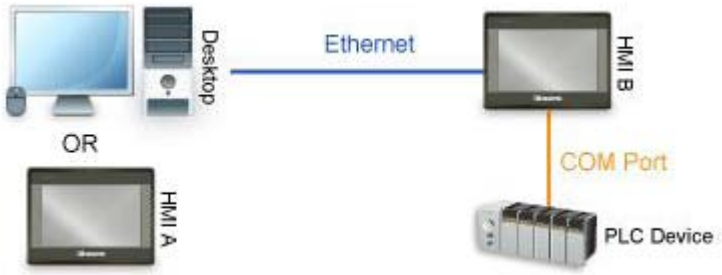


3. В окне [Set Bit Object] в поле [PLC name] выберите HMI A [LB0].



Note	● Число управляемых компьютером панелей не ограничено. ● Панель может управлять данными в компьютере, относящимся к другой панели оператора. Добавьте ПК, как новое удаленную панель (Remote HMI) в проект «.MTP» панели и задайте IP-адрес ПК.

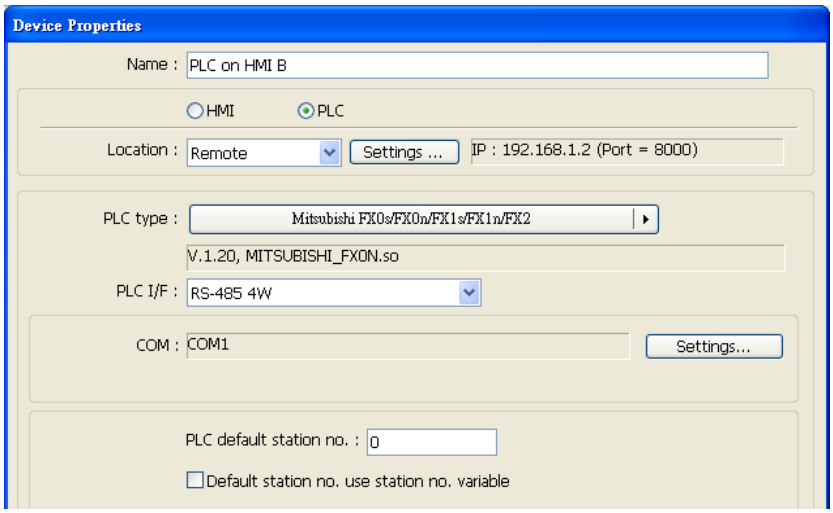
21.3 Работа ПЛК, соединенного с панелями оператора



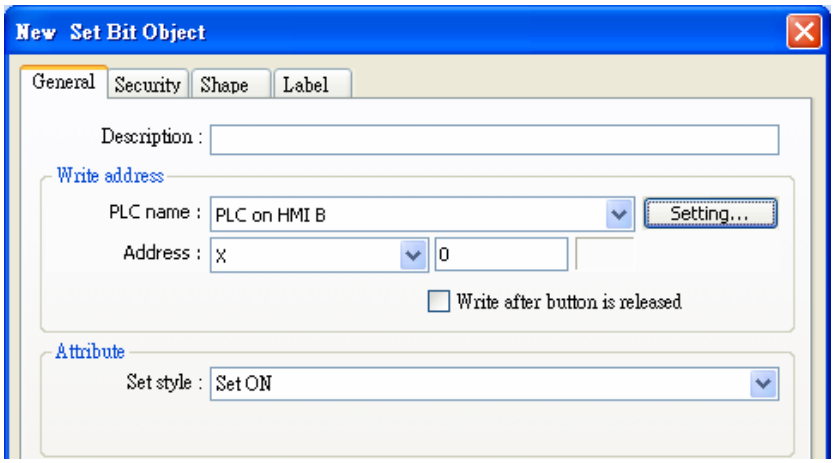
По сети Ethernet компьютер или панель могут также работать с контроллером (ПЛК), соединенным с другой панелью оператора.
Например, с ПЛК Mitsubishi соединеным через COM 1 с панелью HMI B.
Если нужно обеспечить возможность работы ПК или панели HMI A с данным ПЛК, последовательность настроек в проектах ПК или HMI A следующая:

1. Установите IP-адрес панели HMI B = 192.168.1.2 (например).

2. Проект на компьютере или HMI A/ **[System Parameter Settings] / [Device List].** Добавьте удаленный ПЛК и настройте параметры связи. Так как ПЛК присоединен к удаленной панели HMI B, установите тот же самый IP-адрес - 192.168.1.2, что и у панели HMI B.

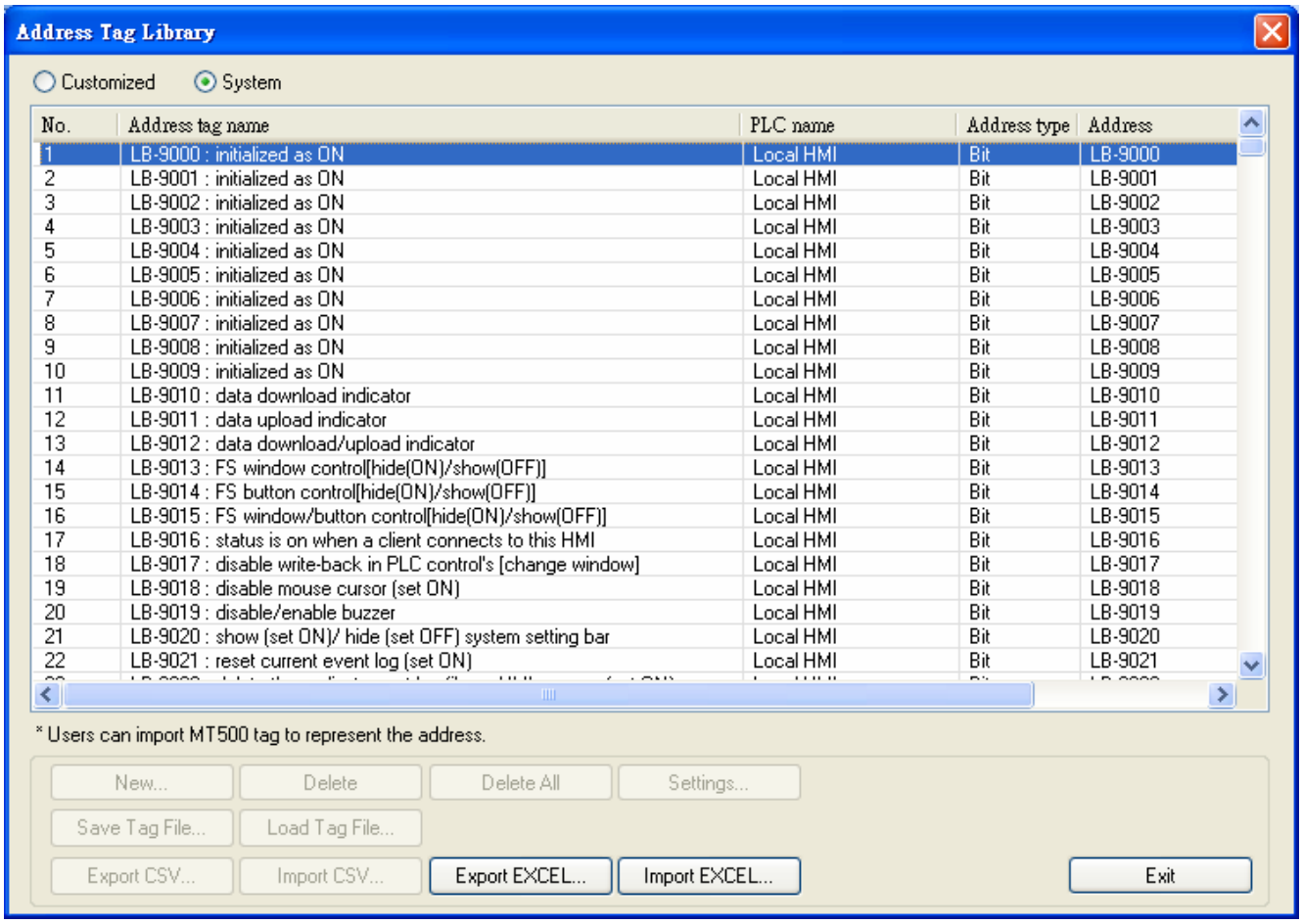


3. В окне [Set Bit Object] в поле [PLC name] выберите “PLC on HMI B” .



Глава 22. Системные биты и регистры

Некоторые локальные слова и биты зарезервированы для использования системой.



No.	Address tag name	PLC name	Address type	Address
1	LB-9000 : initialized as ON	Local HMI	Bit	LB-9000
2	LB-9001 : initialized as ON	Local HMI	Bit	LB-9001
3	LB-9002 : initialized as ON	Local HMI	Bit	LB-9002
4	LB-9003 : initialized as ON	Local HMI	Bit	LB-9003
5	LB-9004 : initialized as ON	Local HMI	Bit	LB-9004
6	LB-9005 : initialized as ON	Local HMI	Bit	LB-9005
7	LB-9006 : initialized as ON	Local HMI	Bit	LB-9006
8	LB-9007 : initialized as ON	Local HMI	Bit	LB-9007
9	LB-9008 : initialized as ON	Local HMI	Bit	LB-9008
10	LB-9009 : initialized as ON	Local HMI	Bit	LB-9009
11	LB-9010 : data download indicator	Local HMI	Bit	LB-9010
12	LB-9011 : data upload indicator	Local HMI	Bit	LB-9011
13	LB-9012 : data download/upload indicator	Local HMI	Bit	LB-9012
14	LB-9013 : FS window control[hide(ON)/show(OFF)]	Local HMI	Bit	LB-9013
15	LB-9014 : FS button control[hide(ON)/show(OFF)]	Local HMI	Bit	LB-9014
16	LB-9015 : FS window/button control[hide(ON)/show(OFF)]	Local HMI	Bit	LB-9015
17	LB-9016 : status is on when a client connects to this HMI	Local HMI	Bit	LB-9016
18	LB-9017 : disable write-back in PLC control's [change window]	Local HMI	Bit	LB-9017
19	LB-9018 : disable mouse cursor (set ON)	Local HMI	Bit	LB-9018
20	LB-9019 : disable/enable buzzer	Local HMI	Bit	LB-9019
21	LB-9020 : show (set ON)/ hide (set OFF) system setting bar	Local HMI	Bit	LB-9020
22	LB-9021 : reset current event log (set ON)	Local HMI	Bit	LB-9021

22.1 Диапазоны Адресов памяти локальной панели

22.1.1 Биты

Память	Тип (Device Type)	Диапазон адресов	Формат адреса
Биты локальной памяти	LB	0 ~ 12095	DDDDD
Слова локальной памяти	LW_BIT	0 ~ 1079915	DDDDDdd DDDDD: address dd: bit no. (00 ~ 15)
Индекс бита Сохраняемой памяти (Retentive Memory)	RBI	0 ~ 65535f	DDDDDh DDDDD: address h: bit no. (0 ~ f) Example: 567a RW_Bit address = 567 + [LW-9000] bit offset = a
Биты слов Сохраняемой памяти (Retentive Memory)	RW_Bit	0 ~ 524287f	DDDDDh DDDDD: address h: bit no. (0 ~ f)
Биты слов Сохраняемой памяти A (Retentive Memory)	RW_A_Bit	0 ~ 65535f	DDDDh DDDDD: address h: bit no. (0 ~ f)

22.1.2 Слова

Память	Тип (Device Type)	Диапазон адресов	Формат адреса
Слова локальной памяти	LW	0 ~ 10799	DDDDD
Слова Сохраняемой памяти (Retentive Memory)	RW	0 ~ 524287	DDDDDD
Индекс слов Сохраняемой памяти (Retentive Memory)	RWI	0 ~ 65535	DDDDD Example: 567 RW address = 567 + [LW-9000]
Слова Сохраняемой памяти A (Retentive Memory)	RW_A	0 ~ 65535	DDDDD
Слова расширенной памяти (Extended Memory)	EM0~EM9	0 ~ 1073741823	DDDDDDDDDD Ограничивается устройством – максимум 2 Гигабайта

22.2 Системное время панели

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-9010	(BCD-16 бит) Местное время (секунды)	R/W	R/Y	R/Y
LW-9011	(BCD-16 бит) Местное время (минуты)	R/W	R/Y	R/Y
LW-9012	(BCD-16 бит) Местное время (час)	R/W	R/Y	R/Y
LW-9013	(BCD-16 бит) Местное время (день)	R/W	R/Y	R/Y
LW-9014	(BCD-16 бит) Местное время (месяц)	R/W	R/Y	R/Y
LW-9015	(BCD-16 бит) Местное время (год)	R/W	R/Y	R/Y
LW-9016	(BCD-16 бит) Местное время (неделя)	R	R	R
LW-9017	(16 бит) Местное время (секунды)	R/W	R/Y	R/Y
LW-9018	(16 бит) Местное время (минуты)	R/W	R/Y	R/Y
LW-9019	(16 бит) Местное время (час)	R/W	R/Y	R/Y
LW-9020	(16 бит) Местное время (день)	R/W	R/Y	R/Y
LW-9021	(16 бит) Местное время (месяц)	R/W	R/Y	R/Y
LW-9022	(16 бит) Местное время (год)	R/W	R/Y	R/Y
LW-9023	(16 бит) Местное время (неделя)	R	R	R
LW-9030	(32 бита) Системное время (ед. изм.: 0,1 секунды)	R	R	R
LW-9048	(16 бит) Время (0: AM; 1: PM)	R/W	R/Y	R/Y
LW-9049	(16 бит) Местное время (час – 12-ти часовой формат)	R/W	R/Y	R/Y

22.3 Пароль и уровень доступа

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-9050	Выход из учетной записи пользователя (logout)	W	Y	Y
LB-9060	Ошибка в пароле	R	R	R
LB-9061	Обновление пароля (Когда установлен в ON)	W	Y	Y
LW-9219	(16-bit) Номер пользователя (1 ~ 12)	R/W	R/Y	R/Y
LW-9220	(32-bit) Пароль	R/W	R/Y	R/Y
LW-9222	(16-bit) Уровни доступа (*Classes)* текущего покьзователя (bit 0:A, bit 1:B,bit 2:C, ...)	R	R	R
LW-9500	(32-bit) Пароль пользователя *user* 1	R/W	R/Y	R/Y
LW-9502~ LW-9503	(32-bit) Пароль пользователя *user* 2	R/W	R/Y	R/Y
LW-9504	(32-bit) Пароль пользователя *user* 3	R/W	R/Y	R/Y
LW-9506	(32-bit) Пароль пользователя *user* 4	R/W	R/Y	R/Y
LW-9508	(32-bit) Пароль пользователя *user* 5	R/W	R/Y	R/Y
LW-9510	(32-bit) Пароль пользователя *user* 6	R/W	R/Y	R/Y
LW-9512	(32-bit) Пароль пользователя *user* 7	R/W	R/Y	R/Y
LW-9514	(32-bit) Пароль пользователя *user* 8	R/W	R/Y	R/Y
LW-9516	(32-bit) Пароль пользователя *user* 9	R/W	R/Y	R/Y
LW-9518	(32-bit) Пароль пользователя *user* 10	R/W	R/Y	R/Y
LW-9520	(32-bit) Пароль пользователя *user* 11	R/W	R/Y	R/Y
LW-9522	(32-bit) Пароль пользователя *user* 12	R/W	R/Y	R/Y
LW-10754	(8 слов): Имя текущего пользователя (см.примечание)	R/W	R/Y	R/Y

Note

Только для расширенного режима безопасности (Enhanced security) .

Download the Demo

В демо-проекте приведен пример использования пароля:

ftp://ftp.weintek.com/MT8000/Project/Part_Sample/Security.zip

22.4 Опрос данных

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-9025	Удаление файлов самых ранних выборок данных из памяти панели. (При установке в ON)	W	Y	Y
LB-9026	Удаление всех файлов выборок данных из памяти панели (ON)	W	Y	Y
LB-9027	Обновление информации о файлах выборок данных в памяти панели (ON)	W	Y	Y
LB-9034	Сохранить *event/data* на HMI, USB , SD (ON)	W	Y	Y
LB-11949	Удаление файлов самых ранних данных на SD (ON)	W	Y	Y
LB-11950	Удаление всех файлов выборок данных на SD (ON)	W	Y	Y
LB-11951	Обновление информации о файлах выборок данных на SD (ON)	W	Y	Y
LB-11952	Удаление файлов самых ранних данных на USB 1 (ON)	W	Y	Y
LB-11953	Удаление всех файлов выборок данных на USB 1 (ON)	W	Y	Y
LB-11954	Обновление информации о файлах выборок данных на USB 1 (ON)	W	Y	Y
LB-11955	Удаление файлов самых ранних данных на USB 2 (ON)	W	Y	Y
LB-11956	Удаление всех файлов выборок данных на USB 2 (ON)	W	Y	Y
LB-11957	Обновление информации о файлах выборок данных на USB 2 (ON)	W	Y	Y
LW-9063	Число записей выборок данных в MT8000.	R	R	R
LW-9064	Размеры файлов всех выборок данных в MT8000 (32-bit Unsigned)	R	R	R
LW-10489	(16bit): Число файлов данных на SD	R	R	R
LW-10490	(32bit): Размер файлов данных на SD	R	R	R
LW-10492	(16bit) : Число файлов данных на USB 1	R	R	R
LW-10493	(32bit): Размер файлов данных на USB 1	R	R	R
LW-10495	(16bit) : Число файлов данных на USB 2	R	R	R
LW-10496	(32bit): Размер файлов данных на USB 2	R	R	R

22.5 Архив событий

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-9021	Очистика текущегой *event log* (При установке в ON).	W	Y	Y
LB-9022	Удаление самого раннего файла *event log* из памяти панели (ON).	W	Y	Y
LB-9023	Удаление всех файлов *event log* из памяти панели (ON).	W	Y	Y
LB-9024	Обновление информации о *event log* в памяти панели (ON).	W	Y	Y
LB-9034	Сохранить *event/data* на HMI, USB, SD (ON)	W	Y	Y
LB-9042	Квитированных всех аварийных событий (ON) .	W	Y	Y
LB-9043	Состояние ON - имеются неквитированные события	R	R	R
LW-9060	Число существующих записей событий	R	R	R
LW-9061	Размеры файлов всех записей событий (32-bit Unsigned)	R	R	R
LB-11940	Удаление самого раннего файла *event log* из SD (ON).	W	Y	Y
LB-11941	Удаление всех файлов *event log* из SD(ON).	W	Y	Y
LB-11942	Обновление информации о *event log* в SD(ON).	W	Y	Y
LB-11943	Удаление самого раннего файла *event log* из USB 1 (ON).	W	Y	Y
LB-11944	Удаление всех файлов *event log* из USB 1 (ON).	W	Y	Y
LB-11945	Обновление информации о *event log* в USB 1(ON).	W	Y	Y
LB-11946	Удаление самого раннего файла *event log* из USB 2 (ON).	W	Y	Y
LB-11947	Удаление всех файлов *event log* из USB 2 (ON).	W	Y	Y
LB-11948	Обновление информации о *event log* в USB 2(ON).	W	Y	Y
LW-9060	(16bit): Число файлов *event log* в памяти панели	R	R	R
LW-9061	(32bit): Размер файлов *event log* в памяти панели	R	R	R

Архив событий - продолжение

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-9450	(16bit): Тэг времени *event log* – секунды *Note	R/W	R/Y	R/Y
LW-9451	(16bit) : Тэг времени *event log* – минуты *Note	R/W	R/Y	R/Y
LW-9452	(16bit): Тэг времени *event log* – часы *Note	R/W	R/Y	R/Y
LW-9453	(16bit): Тэг времени *event log* – день *Note	R/W	R/Y	R/Y
LW-9454	(16bit): Тэг времени *event log* – месяц *Note	R/W	R/Y	R/Y
LW-9455	(16bit): Тэг времени *event log* – год *Note	R/W	R/Y	R/Y
LW-10480	(16bit): Число файлов *event log* на SD	R	R	R
LW-10481	(32bit): Размер файлов *event log* на SD	R	R	R
LW-10483	(16bit): Число файлов *event log* на USB 1	R	R	R
LW-10484	(32bit): Размер файлов *event log* на USB 1	R	R	R
LW-10486	(16bit): Число файлов *event log* на USB 2	R	R	R
LW-10487	(32bit): Размер файлов *event log* на USB 2	R	R	R

Note	Если LW-9450 ~ LW-9455 использованы в качестве тэгов источника времени Event Log, настройте корректно [system parameters] / [General] .

22.9 Операции с аппаратурой панели

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	Макросы R/Y	Удаленная панель R/Y
LB-9018	Отключение курсора мышки (При установке в ON).	R/W	R/Y	R/Y
LB-9019	Отключение звуковых сигналов (ON).	R/W	R/Y	R/Y
LB-9020	Показать - ON, спрятать - OFF полосу системных настроек	R/W	R/Y	R/Y
LB-9033	Запретить - ON, разрешить - OFF функции выгрузки из панели *Note 1	R/W	R/Y	R
LB-9040	Увеличение яркости фоновой подсветки (ON). *Note 2	W	Y	Y
LB-9041	Уменьшение яркости фоновой подсветки (ON). *Note 2	R/W	R/Y	R/Y
LB-9047	Перезагрузка панели (При установке в ON, когда LB9048 -ON)	W	Y	Y
LB-9048	Защита перезагрузки панели	R/W	R/Y	R/Y
LB-9062	Открыть диалог аппаратных настроек (ON)	W	Y	Y
LB-9063	Запретить - ON, разрешить - OFF появление диалога при обнаружении USB	R/W	R/Y	R/Y
LW-9008	(32bit-float): Напряжение батарейки*Note3	R	R	R
LW-9025	(16bit): Загрузка ЦПУ (x 100%)	R	R	R
LW-9026	(16bit): Версия ОС (год)	R	R	R
LW-9027	(16bit): Версия ОС (месяц)	R	R	R
LW-9028	(16bit): Версия ОС (день)	R	R	R
LW-9040	(16bit): Индекс подсветки *Note2	R	R	R
LW-9080	(16bit): Время хранителя подсветки (в минутах)	R/W	R/Y	R/Y
LW-9081	(16bit): Время хранителя экрана (в минутах)	R/W	R/Y	R/Y

Note

1. После изменения настроек, перезапустите панель для обновления.

2. LW-9040 используемая вместе с LB-9040 ~ LB-9041, позволяет настроить уровни яркости подсветки - 0 ~ 31.

3.Когда напряжение батарейки меньше 2.89 В (LW-9008), рекомендуется заменить батарейку.

.

22.7 Информация о сети локальной панели

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-9125	(16bit): Ethernet шлюз 0 (только для использования машиной)	R/W	R/Y	R/Y
LW-9126	(16bit): Ethernet шлюз 1 (только для машины)	R/W	R/Y	R/Y
LW-9127	(16bit): Ethernet шлюз 2 (только для машины)	R/W	R/Y	R/Y
LW-9128	(16bit): Ethernet шлюз 3 (только для машины)	R/W	R/Y	R/Y
LW-9129	(16bit): Ethernet панели IP 0 (только для машины)	R/W	R/Y	R/Y
LW-9130	(16bit): Ethernet панели IP 1 (только для машины)	R/W	R/Y	R/Y
LW-9131	(16bit): Ethernet панели IP 2 (только для машины)	R/W	R/Y	R/Y
LW-9132	(16bit): Ethernet панели IP 3 (только для машины)	R/W	R/Y	R/Y
LW-9133	Номер порта Ethernet	R	R	R
LW-9135	(16bit): Управление доступом к среде (MAC) адрес 0	R	R	R
LW-9136	(16bit): Управление доступом к среде (MAC) адрес 1	R	R	R
LW-9137	(16bit): Управление доступом к среде (MAC) адрес 2	R	R	R
LW-9138	(16bit): Управление доступом к среде (MAC) адрес 3	R	R	R
LW-9139	(16bit): Управление доступом к среде (MAC) адрес 4	R	R	R
LW-9140	(16bit): Управление доступом к среде (MAC) адрес 5	R	R	R
LW-10750	(16bit): HMI ethernet Mask 0 (machine used only)	R/W	R/Y	R/Y
LW-10751	(16bit): Ethernet Маска 0 (только для машины)	R/W	R/Y	R/Y
LW-10752	(16bit): Ethernet Маска 0 (только для машины)	R/W	R/Y	R/Y
LW-10753	(16bit): Ethernet Маска 0 (только для машины)	R/W	R/Y	R/Y

22.8 Рецепты и расширенная память

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-9028	Очистить все данные рецептов (При установке в ON)	W	Y	Y
LB-9029	Очистить все данные рецептов в машине (ON)	W	Y	Y
LB-9460	Накопитель EM0 (SD card) не существует (когда ON)	R	R	R
LB-9461	Накопитель EM1 (SD) не существует	R	R	R
LB-9462	Накопитель EM2 (SD) не существует	R	R	R
LB-9463	Накопитель EM3 (SD) не существует	R	R	R
LB-9464	Накопитель EM4 (SD) не существует	R	R	R
LB-9465	Накопитель EM5 (SD) не существует	R	R	R
LB-9466	Накопитель EM6 (SD) не существует	R	R	R
LB-9467	Накопитель EM7 (SD) не существует	R	R	R
LB-9468	Накопитель EM8 (SD) не существует	R	R	R
LB-9469	Накопитель EM9 (SD) не существует	R	R	R
LB-9470	Накопитель EM0 (USB1) не существует	R	R	R
LB-9471	Накопитель EM1 (USB1) не существует	R	R	R
LB-9472	Накопитель EM2 (USB1) не существует	R	R	R
LB-9473	Накопитель EM3 (USB1) не существует	R	R	R
LB-9474	Накопитель EM4 (USB1) не существует	R	R	R
LB-9475	Накопитель EM5 (USB1) не существует	R	R	R
LB-9476	Накопитель EM6 (USB1) не существует	R	R	R
LB-9467	Накопитель EM7 (USB1) не существует	R	R	R
LB-9478	Накопитель EM8 (USB1) не существует	R	R	R
LB-9479	Накопитель EM9 (USB1) не существует	R	R	R
LB-9480	Накопитель EM0 (USB2) не существует	R	R	R
LB-9481	Накопитель EM1 (USB2) не существует	R	R	R
LB-9482	Накопитель EM2 (USB2) не существует	R	R	R
LB-9483	Накопитель EM3 (USB2) не существует	R	R	R
LB-9484	Накопитель EM4 (USB2) не существует	R	R	R
LB-9485	Накопитель EM5 (USB2) не существует	R	R	R
LB-9486	Накопитель EM6 (USB2) не существует	R	R	R
LB-9487	Накопитель EM7 (USB2) не существует	R	R	R
LB-9488	Накопитель EM8 (USB2) не существует	R	R	R
LB-9489	Накопитель EM9 (USB2) не существует	R	R	R

22.9 Управление местами хранения данных

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-9035	Нехватка памяти панели оператора (когда ON)	R	R	R
LB-9036	Недостаточно памяти карты памяти SD (когда ON)	R	R	R
LB- 9037	Недостаточно места в USB1-накопителе (когда ON)	R	R	R
LB-9038	Недостаточно места в USB2-накопителе (когда ON)	R	R	R
LW-9070	(16bit): Предупреждение о недостаточности свободной памяти (M bytes)	R	R	R
LW-9071	(16bit): Объем резервированной памяти (K bytes)	R	R	R
LW-9072	(32bit): Свободный объем памяти панели (K bytes)	R	R	R
LW-9074	(32bit): Свободный объем карты памяти SD (K bytes)	R	R	R
LW-9076	(32bit): Свободный объем накопителя USB1 (K bytes)	R	R	R
LW-9078	(32bit): Свободный объем накопителя USB2 (K bytes)	R	R	R

Вы можете загрузить демо-проект, в котором используются LW-9072 ~ LW-9078 совместно с объектом «Backup »

ftp://ftp.weintek.com/MT8000/Project/Part_Sample/Backup_Control.zip

22.10 Координаты X и Y сенсорного экрана

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-9041	(16bit): Слово состояния сенсорного экрана (если bit 0 в состоянии ON – пользователь коснулся экрана)	R	R	R
LW-9042	(16bit): Координата X позиции касания	R	R	R
LW-9043	(16bit): Координата Y позиции касания	R	R	R
LW-9044	(16bit): Координата X позиции отрыва касания	R	R	R
LW-9045	(16bit): Координата Y позиции отрыва касания	R	R	R

Вы можете загрузить демо-проект, в котором реализовано переключение окон скользящим касанием.

ftp://ftp.weintek.com/MT8000/Project/Macro_Sample/Demo_Changing_Windows_with_Finger_Slide.zip

22.11 Переменные номеров станций

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	Макросы R/Y	Удаленная панель R/Y
LW-10000	(16bit) : var0 – Переменная номера станции (использование: var0#address)	R/W	R/Y	R/Y
LW-10001	(16bit) : var1 - Переменная номера станции (использование : var1#address)	R/W	R/Y	R/Y
LW-10002	(16bit) : var2 - Переменная номера станции (использование : var2#address)	R/W	R/Y	R/Y
LW-10003	(16bit) : var3 - Переменная номера станции (использование : var3#address)	R/W	R/Y	R/Y
LW-10004	(16bit) : var4 - Переменная номера станции (использование : var4#address)	R/W	R/Y	R/Y
LW-10005	(16bit) : var5 - Переменная номера станции (использование : var5#address)	R/W	R/Y	R/Y
LW-10006	(16bit) : var6 - Переменная номера станции (использование : var6#address)	R/W	R/Y	R/Y
LW-10007	(16bit) : var7 - Переменная номера станции (использование : var7#address)	R/W	R/Y	R/Y
LW-10008	(16bit) : var8 - Переменная номера станции (использование : var8#address)	R/W	R/Y	R/Y
LW-10009	(16bit) : var9 - Переменная номера станции (использование : var9#address)	R/W	R/Y	R/Y
LW-10010	(16bit) : var10 - Переменная номера станции (использование : var10#address)	R/W	R/Y	R/Y
LW-10011	(16bit) : var11 - Переменная номера станции (использование : var11#address)	R/W	R/Y	R/Y
LW-10012	(16bit) : var12 - Переменная номера станции (использование : var12#address)	R/W	R/Y	R/Y
LW-10013	(16bit) : var13 - Переменная номера станции (использование : var13#address)	R/W	R/Y	R/Y
LW-10014	(16bit) : var14 - Переменная номера станции (использование : var14#address)	R/W	R/Y	R/Y
LW-10015	(16bit) : var15 - Переменная номера станции (использование : var15#address)	R/W	R/Y	R/Y

Download the Demo Вы можете загрузить соответствующий демо-проект

ftp://ftp.weintek.com/MT8000/Project/System_Sample/Station_Number_Variable.zip

22.12 Индексный регистр

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	Макросы R/Y	Удаленная панель R/Y
LW-9200	(16bit) : Индекс адреса 0	R/W	R/Y	R/Y
LW-9201	(16bit) : Индекс адреса 1	R/W	R/Y	R/Y
LW-9202	(16bit) : Индекс адреса 2	R/W	R/Y	R/Y
LW-9203	(16bit) : Индекс адреса 3	R/W	R/Y	R/Y
LW-9204	(16bit) : Индекс адреса 4	R/W	R/Y	R/Y
LW-9205	(16bit) : Индекс адреса 5	R/W	R/Y	R/Y
LW-9206	(16bit) : Индекс адреса 6	R/W	R/Y	R/Y
LW-9207	(16bit) : Индекс адреса 7	R/W	R/Y	R/Y
LW-9208	(16bit) : Индекс адреса 8	R/W	R/Y	R/Y
LW-9209	(16bit) : Индекс адреса 9	R/W	R/Y	R/Y
LW-9210	(16bit) : Индекс адреса 10	R/W	R/Y	R/Y
LW-9211	(16bit) : Индекс адреса 11	R/W	R/Y	R/Y
LW-9212	(16bit) : Индекс адреса 12	R/W	R/Y	R/Y
LW-9213	(16bit) : Индекс адреса 13	R/W	R/Y	R/Y
LW-9214	(16bit) : Индекс адреса 14	R/W	R/Y	R/Y
LW-9215	(16bit) : Индекс адреса 15	R/W	R/Y	R/Y
LW-9230	(32bit) : Индекс адреса 16	R/W	R/Y	R/Y
LW-9232	(32bit) : Индекс адреса 17	R/W	R/Y	R/Y
LW-9234	(32bit) : Индекс адреса 18	R/W	R/Y	R/Y
LW-9236	(32bit) : Индекс адреса 19	R/W	R/Y	R/Y
LW-9238	(32bit) : Индекс адреса 20	R/W	R/Y	R/Y
LW-9240	(32bit) : Индекс адреса 21	R/W	R/Y	R/Y
LW-9242	(32bit) : Индекс адреса 22	R/W	R/Y	R/Y
LW-9244	(32bit) : Индекс адреса 23	R/W	R/Y	R/Y
LW-9246	(32bit) : Индекс адреса 24	R/W	R/Y	R/Y
LW-9248	(32bit) : Индекс адреса 25	R/W	R/Y	R/Y
LW-9250	(32bit) : Индекс адреса 26	R/W	R/Y	R/Y
LW-9252	(32bit) : Индекс адреса 27	R/W	R/Y	R/Y
LW-9254	(32bit) : Индекс адреса 28	R/W	R/Y	R/Y
LW-9256	(32bit) : Индекс адреса 29	R/W	R/Y	R/Y
LW-9258	(32bit) : Индекс адреса 30	R/W	R/Y	R/Y
LW-9260	(32bit) : Индекс адреса 31	R/W	R/Y	R/Y

Вы можете загрузить соответствующий демо-проект.

ftp://ftp.weintek.com/MT8000/Project/System_Sample/Index_Function.zip

22.13 Информация о файле MTP

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-9100	(16bit): Имя проекта (16 слов)	R	R	R
LW-9116	(32bit): Размер проекта (в байтах)	R	R	R
LW-9118	(32bit): Размер проекта (в Кбайтах)	R	R	R
LW-9120	(32bit): Версия компилятора	R	R	R
LW-9122	(16bit): Время компиляции проекта (год)	R	R	R
LW-9123	(16bit): Время компиляции проекта (месяц)	R	R	R
LW-9124	(16bit): Время компиляции проекта (день)	R	R	R

22.14 Коммуникации сервера MODBUS

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-9055	MODBUS сервер (COM 1) получает запрос (когда ON)	R	R	R
LB-9056	MODBUS сервер (COM 2) получает запрос (когда ON)	R	R	R
LB-9057	MODBUS сервер (COM 3) получает запрос (когда ON)	R	R	R
LB-9058	MODBUS сервер (ethernet) получает запрос (when ON)	R	R	R
LW-9270	(16bit): Функциональный код запроса- MODBUS сервер (COM 1)	R	R	R
LW-9271	(16bit): Начальный адрес запроса - MODBUS сервер (COM 1)	R	R	R
LW-9272	(16bit) : Количество регистров в запросе - MODBUS сервер (COM 1)	R	R	R
LW-9275	(16bit): Функциональный код запроса - MODBUS сервер (COM 2)	R	R	R
LW-9276	(16bit) : Начальный адрес запроса - MODBUS сервер (COM 2)	R	R	R
LW-9277	(16bit): Количество регистров в запросе - MODBUS сервер (COM 2)	R	R	R
LW-9280	(16bit) : Функциональный код запроса- MODBUS сервер (COM 3)	R	R	R
LW-9281	(16bit): Начальный адрес запроса - MODBUS сервер (COM 3)	R	R	R
LW-9282	(16bit): Количество регистров в запросе - MODBUS сервер (COM 3)	R	R	R
LW-9285	(16bit): Функциональный код запроса- MODBUS сервер (ethernet)	R	R	R
LW-9286	(16bit): Начальный адрес запроса - MODBUS сервер (ethernet)	R	R	R
LW-9287	(16bit): Количество регистров в запросе - MODBUS сервер (ethernet)	R	R	R
LW-9541	(16bit): Номер станции MODBUS/ASCII сервера (COM 1)	R/W	R/Y	R/Y
LW-9542	(16bit): Номер станции MODBUS/ASCII сервера (COM 2)	R/W	R/Y	R/Y
LW-9543	(16bit): Номер станции MODBUS/ASCII сервера (COM 3)	R/W	R/Y	R/Y
LW-9544	(16bit): Номер станции MODBUS/ASCII сервера (ethernet)	R/W	R/Y	R/Y

Коммуникации сервера MODBUS - продолжение

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-9570	(32bit) : Число полученных данных (bytes) (COM 1 MODBUS сервер)	R	R	R
LW-9572	(32bit): Число полученных данных (bytes) (COM 2 MODBUS сервер)	R	R	R
LW-9574	(32bit): Число полученных данных (bytes) (COM 3 MODBUS сервер)	R	R	R
LW-9576	(32bit : Число полученных данных (bytes) (Ethernet MODBUS сервер)	R	R	R

22.15 Настройки коммуникационных параметров

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-9030	Обновление коммуникационных параметров COM1 (При установке в ON)	R/W	R/Y	R/Y
LB-9031	Обновление коммуникационных параметров COM 2 (ON)	R/W	R/Y	R/Y
LB-9032	Обновление коммуникационных параметров COM 3 (ON)	R/W	R/Y	R/Y
LB-9065	Запретить/Разрешить широковещательный номер станции COM1	R/W	R/Y	R/Y
LB-9066	Запретить/Разрешить широковещательный номер станции COM2	R/W	R/Y	R/Y
LB-9067	Запретить/Разрешить широковещательный номер станции COM3	R/W	R/Y	R/Y
LW-9550	(16bit): Режим COM 1 (0:RS232,1:RS485 2W,2:RS485 4W)	R/W	R/Y	R/Y
LW-9551	(16bit): Скорость COM 1 (7:1200,8:2400,0:4800,1:9600,2:1920 0,3: 38400,4:57600,..)	R/W	R/Y	R/Y
LW-9552	(16bit) : Биты данных COM 1 (7 : 7 bits, 8 : 8 bits)	R/W	R/Y	R/Y
LW-9553	(16bit) : Четность COM 1 (0:none, 1:even, 2:odd, 3:mark, 4:space)	R/W	R/Y	R/Y
LW-9554	(16bit) : Стоповые биты COM 1 (1 : 1 bit, 2 : 2 bits)	R/W	R/Y	R/Y
LW-9555	(16bit): Режим COM 2 (0:RS232,1:RS485 2W,2:RS485 4W)	R/W	R/Y	R/Y
LW-9556	(16bit): Скорость COM 2 (7:1200,8:2400,0:4800,1:9600,2:1920 0,3: 38400,4:57600,..)	R/W	R/Y	R/Y
LW-9557	(16bit) :Биты данных COM 2 (7 : 7 bits, 8 : 8 bits)	R/W	R/Y	R/Y
LW-9558	(16bit) : Четность COM 2 (0:none, 1:even, 2:odd, 3:mark, 4:space)	R/W	R/Y	R/Y
LW-9559	(16bit) : Стоповые биты COM 2 (1 : 1 bit, 2 : 2 bits)	R/W	R/Y	R/Y
LW-9560	(16bit): Режим COM 3 (0:RS232,1:RS485 2W)	R/W	R/Y	R/Y
LW-9561	(16bit): Скорость COM 3 (7:1200,8:2400,0:4800,1:9600,2:1920 0,3: 38400,4:57600,..)	R/W	R/Y	R/Y
LW-9562	(16bit) : Биты данных COM 3 (7 : 7 bits, 8 : 8 bits)	R/W	R/Y	R/Y

Настройки коммуникационных параметров – продолжение

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-9563	(16bit) : Четность COM 3 Четность (0:none, 1:even, 2:odd, 3:mark, 4:space)	R/W	R/Y	R/Y
LW-9564	(16bit) : Стоповые биты COM 3 (1 : 1 bit, 2 : 2 bits)	R/W	R/Y	R/Y
LW-9565	(16bit) : COM 1 широковещательный номер станции	R/W	R/Y	R/Y
LW-9566	(16bit) : COM 2 широковещательный номер станции .	R/W	R/Y	R/Y
LW-9567	(16bit) : COM 3 широковещательный номер станции .	R/W	R/Y	R/Y
LW-10500	(16bit) : PLC 1 Таймаут (ед.изм.: 100ms)	R/W	R/Y	R/Y
LW-10501	(16bit) : PLC 1 Задержка возвращения «turn around» (ед.изм.: ms)	R/W	R/Y	R/Y
LW-10502	(16bit) : PLC 1 Задержка посылки ACK (ед.изм.: ms)	R/W	R/Y	R/Y
LW-10503	(16bit) : PLC 1 параметр. 1	R/W	R/Y	R/Y
LW-10504	(16bit) : PLC 1 параметр. 2	R/W	R/Y	R/Y
LW-10505	(16bit) : PLC 2 Таймаут (ед.изм.: 100ms)	R/W	R/Y	R/Y
LW-10506	(16bit) : PLC 2 Задержка возвращения (ед.изм. : ms)	R/W	R/Y	R/Y
LW-10507	(16bit) : PLC 2 Задержка посылки ACK (ед.изм.: ms)	R/W	R/Y	R/Y
LW-10508	(16bit) : PLC 2 параметр. 1	R/W	R/Y	R/Y
LW-10509	(16bit) : PLC 2 параметр. 2	R/W	R/Y	R/Y
LW-10510	(16bit) : PLC 3 Таймаут (ед.изм.: 100ms)	R/W	R/Y	R/Y
LW-10511	(16bit) : PLC 3 Задержка возвращения (ед.изм. : ms)	R/W	R/Y	R/Y
LW-10512	(16bit) : PLC 3 Задержка посылки ACK (ед.изм. : ms)	R/W	R/Y	R/Y
LW-10513	(16bit) : PLC 3 параметр. 1	R/W	R/Y	R/Y
LW-10514	(16bit) : PLC 3 параметр. 2	R/W	R/Y	R/Y
LW-10515	(16bit) : PLC 4 Таймаут (ед.изм.: 100ms)	R/W	R/Y	R/Y
LW-10516	(16bit) : PLC 4 Задержка возвращения (ед.изм. : ms)	R/W	R/Y	R/Y
LW-10517	(16bit) : PLC 4 Задержка посылки ACK (ед.изм. : ms) (SIEMENS S7/400 Link type)	R/W	R/Y	R/Y
LW-10518	(16bit) : PLC 4 параметр. 1 (SIEMENS S7/400 rack)	R/W	R/Y	R/Y

Настройки коммуникационных параметров – продолжение

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-10519	(16bit) : PLC 4 параметр. 2 (SIEMENS S7/400 CPU slot)	R/W	R/Y	R/Y
LW-10520	(16bit) : PLC 5 Таймаут (ед.изм.: 100ms)	R/W	R/Y	R/Y
LW-10521	(16bit) : PLC 5 Задержка возвращения (ед.изм. : ms)	R/W	R/Y	R/Y
LW-10522	(16bit) : PLC 5 Задержка посылки ACK (ед.изм. : ms) (SIEMENS S7/400 Link type)	R/W	R/Y	R/Y
LW-10523	(16bit) : PLC 5 параметр. 1 (SIEMENS S7/400 rack)	R/W	R/Y	R/Y
LW-10524	(16bit) : PLC 5 параметр. 2 (SIEMENS S7/400 CPU slot)	R/W	R/Y	R/Y
LW-10525	(16bit) : PLC 6 Таймаут (ед.изм.: 100ms)	R/W	R/Y	R/Y
LW-10526	(16bit) : PLC 6 Задержка возвращения (ед.изм. : ms)	R/W	R/Y	R/Y
LW-10527	(16bit) : PLC 6 Задержка посылки ACK (ед.изм. : ms) (SIEMENS S7/400 Link type)	R/W	R/Y	R/Y
LW-10528	(16bit) : PLC 6 параметр. 1 (SIEMENS S7/400 rack)	R/W	R/Y	R/Y
LW-10529	(16bit) : PLC 6 параметр. 2 (SIEMENS S7/400 CPU slot)	R/W	R/Y	R/Y
LW-10530	(16bit) : PLC 7 Таймаут (ед.изм.: 100ms)	R/W	R/Y	R/Y
LW-10531	(16bit) : PLC 7 Задержка возвращения (ед.изм. : ms)	R/W	R/Y	R/Y
LW-10532	(16bit) : PLC 7 Задержка посылкиACK (ед.изм.: ms) (SIEMENS S7/400 Link type)	R/W	R/Y	R/Y
LW-10533	(16bit) : PLC 7 параметр. 1 (SIEMENS S7/400 rack)	R/W	R/Y	R/Y
LW-10534	(16bit) : PLC 7 параметр. 2 (SIEMENS S7/400 CPU slot)	R/W	R/Y	R/Y
LW-10535	(16bit) : PLC 8 Таймаут (ед.изм.: 100ms)	R/W	R/Y	R/Y
LW-10536	(16bit) : PLC 8 Задержка возвращения (ед.изм. : ms)	R/W	R/Y	R/Y
LW-10537	(16bit) : PLC 8 Задержка посылкиACK (ед.изм.: ms) (SIEMENS S7/400 Link type)	R/W	R/Y	R/Y
LW-10538	(16bit) : PLC 8 параметр. 1 (SIEMENS S7/400 rack)	R/W	R/Y	R/Y
LW-10539	(16bit) : PLC 8 параметр. 2 (SIEMENS S7/400 CPU slot)	R/W	R/Y	R/Y

22.16 Состояние связей с ПЛК (COM - порты)

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-9150	Автоматическое соединение с PLC 1 (COM1) (при установке в ON)	R/W	R/Y	R/Y
LB-9151	Автоматическое соединение с PLC 2 (COM2) (при установке в ON)	R/W	R/Y	R/Y
LB-9152	Автоматическое соединение с PLC 3 (COM3) (при установке в ON)	R/W	R/Y	R/Y
LB-9200	Состояние PLC 1 (SN0, COM1), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9201	Состояние PLC 1 (SN1, COM1), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9202	Состояние PLC 1 (SN2, COM1), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9203	Состояние PLC 1 (SN3, COM1), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9204	Состояние PLC 1 (SN4, COM1), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9205	Состояние PLC 1 (SN5, COM1), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9206	Состояние PLC 1 (SN6, COM1), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9207	Состояние PLC 1 (SN7, COM1), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9500	Состояние PLC 2 (SN0, COM2), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9501	Состояние PLC 2 (SN1, COM2), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9502	Состояние PLC 2 (SN2, COM2), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9503	Состояние PLC 2 (SN3, COM2), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9504	Состояние PLC 2 (SN4, COM2), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9505	Состояние PLC 2 (SN5, COM2), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9506	Состояние PLC 2 (SN6, COM2), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9507	Состояние PLC 2 (SN7, COM2), повторная попытка соединения (ON)	R/W	R/Y	R/Y

Состояние связей с ПЛК (COM - порты) - продолжение

Адрес	Описание	Read (R) / Write (W) / Control (Y)	Адрес	Описание
LB-9800	Состояние PLC 3 (SN0, COM3), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9801	Состояние PLC 3 (SN1, COM3), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9802	Состояние PLC 3 (SN2, COM3), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9803	Состояние PLC 3 (SN3, COM3), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9804	Состояние PLC 3 (SN4, COM3), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9805	Состояние PLC 3 (SN5, COM3), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9806	Состояние PLC 3 (SN6, COM3), повторная попытка соединения (ON)	R/W	R/Y	R/Y
LB-9807	Состояние PLC 3 (SN7, COM3), повторная попытка соединения (ON)	R/W	R/Y	R/Y

22.17 Состояние связей с ПЛК (Ethernet)

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-9153	Автоматическое соединение с PLC 4 (ethernet) (когда в ON)	R/W	R/Y	R/Y
LB-9154	Автоматическое соединение с PLC 5 (ethernet) (когда в ON)	R/W	R/Y	R/Y
LB-9155	Автоматическое соединение с PLC 6 (ethernet) (когда в ON)	R/W	R/Y	R/Y
LB-9156	Автоматическое соединение с PLC 7 (ethernet) (когда в ON)	R/W	R/Y	R/Y
LB-9157	Автоматическое соединение с PLC 8 (ethernet) (когда в ON)	R/W	R/Y	R/Y
LB-9158	Автоматическое соединение с PLC 9 (ethernet) (когда в ON)	R/W	R/Y	R/Y
LB-10070	Принудительное соединение с PLC 4 (ethernet), когда IP или системные параметры изменены «on-line» (в ON)	R/W	R/Y	R/Y
LB-10071	Принудительное соединение с PLC 5 (ethernet), когда IP или системные параметры изменены «on-line» (в ON)	R/W	R/Y	R/Y
LB-10072	Принудительное соединение с PLC 6 (ethernet), когда IP или системные параметры изменены «on-line» (в ON)	R/W	R/Y	R/Y
LB-10073	Принудительное соединение с PLC 7 (ethernet), когда IP или системные параметры изменены «on-line» (в ON)	R/W	R/Y	R/Y
LB-10074	Принудительное соединение с PLC 8 (ethernet), когда IP или системные параметры изменены «on-line» (в ON)	R/W	R/Y	R/Y
LB-10075	Принудительное соединение с PLC 9 (ethernet), когда IP или системные параметры изменены «on-line» (в ON)	R/W	R/Y	R/Y
LB-10100	Состояние PLC 4 (ethernet),попытка повторного соединения (уст. в ON)	R/W	R/Y	R/Y
LB-10400	Состояние PLC 5 (ethernet),попытка повторного соединения (уст. в ON)	R/W	R/Y	R/Y
LB-10700	Состояние PLC 6 (ethernet),попытка повторного соединения (уст. в ON)	R/W	R/Y	R/Y
LB-11000	Состояние PLC 7 (ethernet),попытка повторного соединения (уст. в ON)	R/W	R/Y	R/Y
LB-11300	Состояние PLC 8 (ethernet),попытка повторного соединения (уст. в ON)	R/W	R/Y	R/Y
LB-11600	Состояние PLC 9(ethernet),попытка повторного соединения (уст. в ON)	R/W	R/Y	R/Y
LB-11900	Состояние PLC 10 (ethernet),попытка повторного соединения (уст. в ON)	R/W	R/Y	R/Y
LB-11901	Состояние PLC 11 (ethernet),попытка повторного соединения (уст. в ON)	R/W	R/Y	R/Y

Состояние связей с ПЛК (Ethernet) - продолжение

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-11902	Состояние PLC 12 (ethernet),попытка повторного соединения (уст. в ON)	R/W	R/Y	R/Y
LB-11903	Состояние PLC 13 (ethernet),попытка повторного соединения (уст. в ON)	R/W	R/Y	R/Y
LB-11904	Состояние PLC 14 (ethernet),попытка повторного соединения (уст. в ON)	R/W	R/Y	R/Y
LB-11905	Состояние PLC 15 (ethernet),попытка повторного соединения (уст. в ON)	R/W	R/Y	R/Y
LB-11906	Состояние PLC 16 (ethernet),попытка повторного соединения (уст. в ON)	R/W	R/Y	R/Y
LW-9600	(16bit) : PLC 4 адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9601	(16bit) : PLC 4 адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9602	(16bit) : PLC 4 адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9603	(16bit) : PLC 4 адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9604	(16bit) : PLC 4 – Номер порта	R/W	R/Y	R/Y
LW-9605	(16bit) : PLC 5 адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9606	(16bit) : PLC 5 адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9607	(16bit) : PLC 5 адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9608	(16bit) : PLC 5 адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9609	(16bit) : PLC 5 – Номер порта	R/W	R/Y	R/Y
LW-9610	(16bit) : PLC 6 адресIP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9611	(16bit) : PLC 6 адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9612	(16bit) : PLC 6 адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9613	(16bit) : PLC 6 адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9614	(16bit) : PLC 6 – Номер порта	R/W	R/Y	R/Y
LW-9615	(16bit) : PLC 7 адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9616	(16bit) : PLC 7 адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9617	(16bit) : PLC 7 адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9618	(16bit) : PLC 7 адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y

Состояние связей с ПЛК (Ethernet) – продолжение

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-9619	(16bit) : PLC 7 – Номер порта	R/W	R/Y	R/Y
LW-9620	(16bit) : PLC 8 адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9621	(16bit) : PLC 8 адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9622	(16bit) : PLC 8 адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9623	(16bit) : PLC 8 адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9624	(16bit) : PLC 8 – Номер порта	R/W	R/Y	R/Y
LW-9625	(16bit) : PLC 9 адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9626	(16bit) : PLC 9 адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9627	(16bit) : PLC 9 адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9628	(16bit) : PLC 9 адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9629	(16bit) : PLC 9 – Номер порта	R/W	R/Y	R/Y

22.18 Состояние связей с ПЛК (USB)

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-9190	Автоматическое соединение с PLC (USB 1) (когда в ON)	R/W	R/Y	R/Y
LB-9191	Состояние PLC (USB 1),попытка повторного соединения (уст. в ON)	R/W	R/Y	R/Y
LB-9193	Автоматическое соединение с PLC (USB 2) (когда в ON)	R/W	R/Y	R/Y
LB-9194	Состояние PLC (USB 2),попытка повторного соединения (уст. в ON)	R/W	R/Y	R/Y

22.19 Состояние связей с ПЛК (CAN Bus)

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-12080	Автоматическое соединение с PLC (CAN Bus) (когда в ON)	R/W	R/Y	R/Y
LB-12081	Состояние PLC (CAN Bus),попытка повторного соединения (уст. в ON)	R/W	R/Y	R/Y

22.20 Состояние связей с удаленными операторскими панелями

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-9068	Автоматическое соединение с удаленной панелью HMI 1 (когда ON)	R/W	R/Y	R/Y
LB-9069	Автоматическое соединение с удаленной панелью HMI 2 (когда ON)	R/W	R/Y	R/Y
LB-9070	Автоматическое соединение с удаленной панелью HMI 3 (когда ON)	R/W	R/Y	R/Y
LB-9071	Автоматическое соединение с удаленной панелью HMI 4 (когда ON)	R/W	R/Y	R/Y
LB-9072	Автоматическое соединение с удаленной панелью HMI 5 (когда ON)	R/W	R/Y	R/Y
LB-9073	Автоматическое соединение с удаленной панелью HMI 6 (когда ON)	R/W	R/Y	R/Y
LB-9074	Автоматическое соединение с удаленной панелью HMI 7 (когда ON)	R/W	R/Y	R/Y
LB-9075	Автоматическое соединение с удаленной панелью HMI 8 (когда ON)	R/W	R/Y	R/Y
LB-9100	Состояние удаленной панели HMI 1, попытка повторного соединения (ON)	R/W	R/Y	R/Y
LB-9101	Состояние удаленной панели HMI 2, попытка повторного соединения (ON)	R/W	R/Y	R/Y
LB-9102	Состояние удаленной панели HMI 3, попытка повторного соединения (ON)	R/W	R/Y	R/Y
LB-9103	Состояние удаленной панели HMI 4, попытка повторного соединения (ON)	R/W	R/Y	R/Y
LB-9104	Состояние удаленной панели HMI 5, попытка повторного соединения (ON)	R/W	R/Y	R/Y
LB-9105	Состояние удаленной панели HMI 6, попытка повторного соединения (ON)	R/W	R/Y	R/Y
LB-9106	Состояние удаленной панели HMI 7, попытка повторного соединения (ON)	R/W	R/Y	R/Y
LB-9107	Состояние удаленной панели HMI 8, попытка повторного соединения (ON)	R/W	R/Y	R/Y
LB-9149	Принудительное пересоединение с удаленной панелью remote HMI, когда IP изменен «on-line» (Установить в ON)	R/W	R/Y	R/Y
LW-9800	(16bit) : Удаленная панель HMI 1, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9801	(16bit) : Удаленная панель HMI 2, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9802	(16bit) : Удаленная панель HMI 1, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y

Состояние связей с удаленными операторскими панелями – продолжение

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-9803	(16bit) : Удаленная панель HMI 1, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9804	(16bit) : Номер порта уд. панели HMI 1	R/W	R/Y	R/Y
LW-9805	(16bit) : Удаленная панель HMI 2, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9806	(16bit) : Удаленная панель HMI 2, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9807	(16bit) : Удаленная панель HMI 2, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9808	(16bit) : Удаленная панель HMI 2, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9809	(16bit) : Номер порта уд. панели HMI 2	R/W	R/Y	R/Y
LW-9810	(16bit) : Удаленная панель HMI 3, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9811	(16bit) : Удаленная панель HMI 3, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9812	(16bit) : Удаленная панель HMI 3, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9813	(16bit) : Удаленная панель HMI 3, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9814	(16bit) : Номер порта уд. панели HMI 3	R/W	R/Y	R/Y
LW-9815	(16bit) : Удаленная панель HMI 4, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9816	(16bit) : Удаленная панель HMI 4, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9817	(16bit) : Удаленная панель HMI 4, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9818	(16bit) : Удаленная панель HMI 4, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9819	(16bit) : Номер порта уд. панели HMI 4	R/W	R/Y	R/Y

Состояние связей с удаленными операторскими панелями - продолжение

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-9820	(16bit) : Удаленная панель HMI 5, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9821	(16bit) : Удаленная панель HMI 5, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9822	(16bit) : Удаленная панель HMI 5, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9823	(16bit) : Удаленная панель HMI 5, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9824	(16bit) : Номер порта уд. панели HMI 5	R/W	R/Y	R/Y
LW-9825	(16bit) : Удаленная панель HMI 6, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9826	(16bit) : Удаленная панель HMI 6, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9827	(16bit) : Удаленная панель HMI 6, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9828	(16bit) : Удаленная панель HMI 6, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9829	(16bit) : Номер порта уд. панели HMI 6	R/W	R/Y	R/Y
LW-9830	(16bit) : Удаленная панель HMI 7, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9831	(16bit) : Удаленная панель HMI 7, адрес (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9832	(16bit) : Удаленная панель HMI 7, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9833	(16bit) : Удаленная панель HMI 7, адрес (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9834	(16bit) : Номер порта уд. панели HMI 7	R/W	R/Y	R/Y
LW-9835	(16bit) : Удаленная панель HMI 8, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9836	(16bit) : Удаленная панель HMI 8, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y

Состояние связей с удаленными операторскими панелями - продолжение

Адрес	Описание	Read ® / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-9837	(16bit) : Удаленная панель HMI 8, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9838	(16bit) : Удаленная панель HMI 8, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9839	(16bit) : Номер порта уд. панели HMI 8	R/W	R/Y	R/Y
LW-9905	(16bit) : Удаленная панель HMI 21, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9906	(16bit) : Удаленная панель HMI 21, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9907	(16bit) : Удаленная панель HMI 21, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9908	(16bit) : Удаленная панель HMI 21, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9909	(16bit) : Номер порта уд. панели HMI 21	R/W	R/Y	R/Y
LW-9910	(16bit) : Удаленная панель HMI 22, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9911	(16bit) : Удаленная панель HMI 22, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9912	(16bit) : Удаленная панель HMI 22, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9913	(16bit) : Удаленная панель HMI 22, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9914	(16bit) : Номер порта уд. панели HMI 22	R/W	R/Y	R/Y
LW-9915	(16bit) : Удаленная панель HMI 23, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9916	(16bit) : Удаленная панель HMI 23, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9917	(16bit) : Удаленная панель HMI 23, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y

Состояние связей с удаленными операторскими панелями - продолжение

Адрес	Описание	Read ® / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-9918	(16bit) : Удаленная панель HMI 23, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9919	(16bit) : Номер порта уд.панели HMI 23	R/W	R/Y	R/Y
LW-9920	(16bit) : Удаленная панель HMI 24, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9921	(16bit) : Удаленная панель HMI 24, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9922	(16bit) : Удаленная панель HMI 24, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9923	(16bit) : Удаленная панель HMI 24, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9924	(16bit) : Номер порта уд.панели HMI 24	R/W	R/Y	R/Y
LW-9925	(16bit) : Удаленная панель HMI 25, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9926	(16bit) : Удаленная панель HMI 25, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9927	(16bit) : Удаленная панель HMI 25, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9928	(16bit) : Удаленная панель HMI 25, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9929	(16bit) : Номер порта уд.панели HMI 25	R/W	R/Y	R/Y
LW-9930	(16bit) : Удаленная панель HMI 26, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9931	(16bit) : Удаленная панель HMI 26, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9932	(16bit) : Удаленная панель HMI 26, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9933	(16bit) Удаленная панель HMI 26, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9934	(16bit) : Номер порта уд.панели HMI 26	R/W	R/Y	R/Y
LW-9935	(16bit) : Удаленная панель HMI 27, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9936	(16bit) : Удаленная панель HMI 27, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9937	(16bit) : Удаленная панель HMI 27, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9938	(16bit) : Удаленная панель HMI 27, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9939	(16bit) : Номер порта уд.панели HMI 27	R/W	R/Y	R/Y
LW-9940	(16bit) : Удаленная панель HMI 28, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y

Состояние связей с удаленными операторскими панелями - продолжение

Адрес	Описание	Read ® / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-9941	(16bit) : Удаленная панель HMI 28, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9942	(16bit) : Удаленная панель HMI 28, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9943	(16bit) : Удаленная панель HMI 28, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9944	(16bit) : Номер порта уд. панели HMI 28	R/W	R/Y	R/Y
LW-9945	(16bit) : Удаленная панель HMI 29, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9946	(16bit) : Удаленная панель HMI 29, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9947	(16bit) : Удаленная панель HMI 29, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9948	(16bit) : Удаленная панель HMI 29, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9949	(16bit) : Номер порта уд. панели HMI 29	R/W	R/Y	R/Y
LW-9950	(16bit) : Удаленная панель HMI 30, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9951	(16bit) : Удаленная панель HMI 30, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9952	(16bit) : Удаленная панель HMI 30, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9953	(16bit) : Удаленная панель HMI 30, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9954	(16bit) : Номер порта уд. панели HMI 30	R/W	R/Y	R/Y
LW-9955	(16bit) : Удаленная панель HMI 31, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9956	(16bit) : Удаленная панель HMI 31, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y

Состояние связей с удаленными операторскими панелями - продолжение

Адрес	Описание	Read ® / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-9957	(16bit) : Удаленная панель HMI 31, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9958	(16bit) : Удаленная панель HMI 31, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9959	(16bit) : Номер порта уд. панели HMI 31	R/W	R/Y	R/Y
LW-9960	(16bit) : Удаленная панель HMI 32, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9961	(16bit) : Удаленная панель HMI 32, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9962	(16bit) : Удаленная панель HMI 32, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9963	(16bit) : Удаленная панель HMI 32, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9964	(16bit) : Номер порта уд. панели HMI 32	R/W	R/Y	R/Y

22.21 Состояние связей с удаленными ПЛК

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-10050	(16bit) : IP0 панели HMI соединенной с удаленным PLC 1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10051	(16bit) : IP1 панели HMI соединенной с удаленным PLC 1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10052	(16bit) : IP2 панели HMI соединенной с удаленнымPLC 1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10053	(16bit) : IP3 панели HMI соединенной с удаленным PLC 1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10054	(16bit) : Номер порта панели HMI соединенной с удаленным PLC 1	R/W	R/Y	R/Y
LW-10055	(16bit) : IP0 панели HMI соединенной с удаленным PLC 2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10056	(16bit) : IP1 панели HMI соединенной с удаленным PLC 2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10057	(16bit) : IP2 панели HMI соединенной с удаленным PLC 2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10058	(16bit) : IP3 панели HMI соединенной с удаленным PLC 2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10059	(16bit) : Номер порта панели HMI соединенной с удаленным PLC 2	R/W	R/Y	R/Y
LW-10060	(16bit) : IP0 панели HMI соединенной с удаленным PLC 3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10061	(16bit) : IP1 панели HMI соединенной с удаленным PLC 3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10062	(16bit) : IP2 панели HMI соединенной с удаленным PLC 3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10063	(16bit) : IP3 панели HMI соединенной с удаленным PLC 3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10064	(16bit) : Номер порта панели HMI соединенной с удаленным PLC 3	R/W	R/Y	R/Y
LW-10065	(16bit) : IP0 панели HMI соединенной с удаленным PLC 4 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y

Состояние связей с удаленными ПЛК – продолжение

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	Макросы R/Y	Удаленная панель R/Y
LW-10066	(16bit) : IP1 панели HMI соединенной с удаленным PLC 4 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10067	(16bit) : IP2 панели HMI соединенной с удаленнымPLC 4 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10068	(16bit) : IP3 панели HMI соединенной с удаленным PLC 4 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10069	(16bit) : Номер порта панели HMI соединенной с удаленным PLC 4	R/W	R/Y	R/Y
LW-10300	(16bit) : Удаленный PLC 1, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10301	(16bit) : Удаленный PLC 1, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10302	(16bit) : Удаленный PLC 1, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10303	(16bit) : Удаленный PLC 1, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10304	(16bit) : Номер порта удаленного PLC 4	R/W	R/Y	R/Y
LW-10305	(16bit) : Удаленный PLC 2, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10306	(16bit) : Удаленный PLC 2, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10307	(16bit) : Удаленный PLC 2, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10308	(16bit) : Удаленный PLC 2, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10309	(16bit) : Номер порта удаленного PLC 2 .	R/W	R/Y	R/Y
LW-10310	(16bit) : Удаленный PLC 3, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10311	(16bit) : Удаленный PLC 3, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10312	(16bit) : Удаленный PLC 3, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10313	(16bit) : Удаленный PLC 3, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10314	(16bit) : Номер порта удаленного PLC 3	R/W	R/Y	R/Y

Состояние связей с удаленными ПЛК – продолжение

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-10315	(16bit) : Удаленный PLC 4, адрес IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10316	(16bit) : Удаленный PLC 4, адрес IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10317	(16bit) Удаленный PLC 4, адрес IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10318	(16bit) : Удаленный PLC 4, адрес IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10319	(16bit) : Номер порта удаленного PLC 4	R/W	R/Y	R/Y

22.1 Сообщения об ошибках связи и номер исполняемой команды

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-9350	(16bit) : номер исполняемой команды в панели HMI	R	R	R
LW-9351	(16bit) : номер исполняемой команды в PLC 1 (COM 1)	R	R	R
LW-9352	(16bit) : номер исполняемой команды в PLC 2 (COM 2)	R	R	R
LW-9353	(16bit) : номер исполняемой команды в PLC 3 (COM 3)	R	R	R
LW-9354	(16bit) : номер исполняемой команды в PLC 4 (ethernet)	R	R	R
LW-9355	(16bit) : номер исполняемой команды в PLC 5 (ethernet)	R	R	R
LW-9356	(16bit) : номер исполняемой команды в PLC 6 (ethernet)	R	R	R
LW-9357	(16bit) : номер исполняемой команды в PLC 7 (ethernet)	R	R	R
LW-9390	(16bit) : номер исполняемой команды в PLC (USB)	R	R	R
LW-9392	(16bit) : номер исполняемой команды вPLC (CAN Bus)	R	R	R
LW-9400	(16bit) : код ошибки для PLC 1	R	R	R
LW-9401	(16bit) : код ошибки для PLC 2	R	R	R
LW-9402	(16bit) : код ошибки для PLC 3	R	R	R
LW-9403	(16bit) : код ошибки для PLC 4	R	R	R
LW-9404	(16bit) : код ошибки для PLC 5	R	R	R
LW-9405	(16bit) : код ошибки для PLC 6	R	R	R
LW-9406	(16bit) : код ошибки для PLC 7	R	R	R
LW-9407	(16bit) : код ошибки для PLC 8	R	R	R
LW-9490	(16bit) : код ошибки для USB PLC	R	R	R

22.2 Различные функции

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-9000 ~ LB-9009	Инициализированы в состоянии ON	R/W	R/Y	R/Y
LB-9010	Индикатор загрузки данных	R	R	R
LB-9011	Индикатор выгрузки данных	R	R	R
LB-9012	Индикатор загрузки/выгрузки	R	R	R
LB-9016	Клиент соединен с панелью (ON)	R	R	R
LB-9017	Отключение обратной записи в управление ПЛК [change window]	R/W	R/Y	R/Y
LB-9039	Состояние резервного файла (Резервирование идет, когда в ON)	R	R	R
LB-9045	Отказ связи «memory-map» (ON)	R	R	R
LB-9049	Разрешить (ON) / отключить (OFF) сторожевой таймер *Note1	R/W	R/Y	R/Y
LB-9059	Отключить функцию MACRO TRACE (когда в ON) *Note2	R/W	R/Y	R/Y
LB-9064	Активировать устройство USB-barcode (отключение клавишами) (когда в ON) *Note3	R/W	R/Y	R
LW-9006	(16bit) : номер подсоединенного клиента.	R	R	R
LW-9024	(16bit): системный регистр «memory link»	R/W	R/Y	R/Y
LW-9032	(8 words) : Имя резервного файла истории на SD, USB	R/W	R/Y	R/Y
LW-9050	(16bit) : «ID» текущего основного окна	R	R	R
LW-9134	(16bit) : Языковой режим *Note4	R/W	R/Y	R/Y
LW-9141	(16bit) : Номер станции панели HMI	R/W	R/Y	R/Y
LW-9216	(16bit) : the result of importing email data	R	R	R
LW-9300	(16bit) : «ID» драйвера локального PLC 1	R	R	R
LW-9301	(16bit) : «ID» драйвера локального PLC 2	R	R	R
LW-9302	(16bit) : «ID» драйвера локального PLC 3	R	R	R
LW-9303	(16bit) : «ID» драйвера локального PLC 4	R	R	R
LW-9530	(8 words) : Пароль сервера «VNC»	R/W	R/Y	R/Y

1. Когда функция сторожевого таймера LB-9049 активирована, если происходит отказ связи с панелью, система будет перезагружена через 10 секунд.

2. Демонстрационный проект с использованием LB-9059
ftp://ftp.weintek.com/MT8000/Project/Macro_Sample/Macro_Trace.zip

3. Демонстрационный проект с использованием LB-9064
ftp://ftp.weintek.com/MT8000/Project/PLC_Sample/Barcode_Scanner/Demo_Barcode_Scanner.zip

4. Если пользователь хочет чтобы текстовый объект был многоязычным (исключая использование Label Library), необходимо использовать системный регистр [LW-9134: language mode – языковой режим]. Значение LW-9134 может быть установлено в диапазоне от 0 до 7. Различные числа LW-9134 соотносятся с различными языками. Способ использования LW-9134 будет отличаться, если не все языки будут выбраны при компиляции файла проекта.

Например: Если пользователь определил 5 языков в *Label Library:* Language 1 (Traditional Chinese), Language 2 (Simplified Chinese), Language 3 (English), Language 4 (French) и Language 5 (Japanese), а использовал при загрузке только Language 1, 3, 5 , the соответствующие языкам числа в LW-9134 будут:

0 -> Language 1 (Traditional Chinese),

1 -> Language 3 (English),

2 -> Language 5 (Japanese).

Демонстрационный проект переключения языков с использованием объекта «Option List» и LW-9134?
ftp://ftp.weintek.com/MT8000/Project/System_Sample/Multi_Language_with_Option.zip

22.24 Удаленная печать / Резервный сервер

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-10069	forced to reconnect remote printer/backup server when IP changed on-line (set ON)	R/W	R/Y	R/Y
LW-9770	(16bit) : IP0 удаленного принтера/резервного сервера (IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9771	(16bit) : IP1 удаленного принтера/резервного сервера (IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9772	(16bit) : IP2 удаленного принтера/резервного сервера (IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9773	(16bit) : IP3 удаленного принтера/резервного сервера (IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9774	(6 words) : Имя пользователя удаленного принтера/резервного сервера *Note1	R/W	R/Y	R/Y
LW-9780	(6 words) : Пароль удаленного принтера/резервного сервера *Note1	R/W	R/Y	R/Y

Note 1. После изменения LW-9774 и LW9780, необходимо перезагрузить панель.

Download the Demo Загрузите соответствующий демо-проект:
ftp://ftp.weintek.com/MT8000/Project/System_Sample/Printer_Backup_Server.zip

22.25 Легкий доступ (EasyAccess)

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-9051	Отсоединиться - (OFF) / соединиться (ON) с сервером EasyAccess	R/W	R/Y	R/Y
LB-9052	Состояние соединения с сервером EasyAccess	R	R	R
LB-9196	Локальная панель поддерживает только функции наблюдения (когда ON)	R/W	R/Y	R/Y
LB-9197	Поддержка только функций наблюдения для удаленных панелей (когда ON)	R/W	R/Y	R/Y

Дополнительная информация об EasyAccess:
http://www.ihmi.net/

22.26 Настройки функции сквозного пропуска (Pass-Through)

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-9900	(16bit) : Режим работы панели (0 : normal, 1~3 : test (COM 1~COM 3)	R/W	R/Y	R/Y
LW-9901	(16bit) : Сквозной пропуск, COM-порт источник (1~3 : COM 1~COM 3)	R/W	R/Y	R/Y
LW-9902	(16bit) : Сквозной пропуск, COM-порт назначения port (1~3 : COM 1~COM 3)	R/W	R/Y	R/Y

22.27 Отключение окна «PLC No Response»

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-9192	Отключить окно "PLC No Response" для USB 1 PLC (когда ON)	R/W	R/Y	R/Y
LB-9195	Отключить окно "PLC No Response" для USB 2 PLC (когда ON)	R/W	R/Y	R/Y
LB-11960	Отключить окно "PLC No Response" для PLC 1 (когда ON)	R/W	R/Y	R/Y
LB-11961	Отключить окно "PLC No Response" для PLC 2 (когда ON)	R/W	R/Y	R/Y
LB-11962	Отключить окно "PLC No Response" для PLC 3 (когда ON)	R/W	R/Y	R/Y
LB-11963	Отключить окно "PLC No Response" для PLC 4 (когда ON)	R/W	R/Y	R/Y
LB-11964	Отключить окно "PLC No Response" для PLC 5 (когда ON)	R/W	R/Y	R/Y
LB-11965	Отключить окно "PLC No Response" для PLC 6 (когда ON)	R/W	R/Y	R/Y
LB-11966	Отключить окно "PLC No Response" для PLC 7 (когда ON)	R/W	R/Y	R/Y
LB-11967	Отключить окно "PLC No Response" для PLC 8 (когда ON)	R/W	R/Y	R/Y
LB-12082	Отключить окно "PLC No Response" для устройства CAN Bus (когда ON)	R/W	R/Y	R/Y

22.28 Панель и ключ защиты

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-9046	Ключ проекта отличается от ключа панели (когда ON)	R	R	R
LW-9046	(32bit) : Ключ панели (HMI key) *Note1	R/W	R/Y	R

1. При изменении ключа панели (HMI key) с использованием LW-9046, перезагрузите панель, чтобы активировать новые настройки/.

22.29 Управление Окном быстрого выбора (FS)

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-9013	Управление окном Fast Selection (FS) [спрятать –ON / показать - OFF]	R/W	R/Y	R/Y
LB-9014	Кнопка управления окном FS [спрятать –ON / показать - OFF]	R/W	R/Y	R/Y
LB-9015	Окно FS / кнопка управления [спрятать –ON / показать - OFF]	R/W	R/Y	R/Y

22.30 Функции объекта ввода данных

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LW-9002	(32bit-float) : Верхний предел ввода	R	R	R
LW-9004	(32bit-float) : Нижний предел ввода	R	R	R
LW-9052	(32bit-float) : Предыдущее значение ввода объекта «numeric input»	R	R	R
LW-9150	(32 words) : Данные введенные с клавиатуры (ASCII)	R	R	R
LW-9540	(16bit) : Резервировано для клавиши «caps lock»	R	R	R

22.31 Ограничение локальных / удаленных операций

Адрес	Описание	Read (R) / Write (W) / Control (Y)		
		Локальная панель	**Макросы R/Y**	**Удаленная панель R/Y**
LB-9044	Отключить удаленное управление (когда ON)	R/W	R/Y	R/Y
LB-9053	Запретить операции удаленного чтения с паролем (когда ON)	R/W	R/Y	R/Y
LB-9054	Запретить операции удаленной записи с паролем (когда ON)	R/W	R/Y	R/Y
LB-9196	Локальная панель поддерживает только функции наблюдения (когда ON)	R/W	R/Y	R/Y
LB-9197	Поддержка функций наблюдения только для удаленных панелей (когда ON)	R/W	R/Y	R/Y
LB-9198	Отключить возможность локальной панели запускать MACRO (когда ON)	R/W	R/Y	R/Y
LB-9199	Отключить возможность удаленной панели запускать MACRO (когда ON)	R/W	R/Y	R/Y

Глава 23. Поддерживаемые принтеры

23.1 Типы поддерживаемых принтеров

Панели совместимы со следующими принтерами:

EPSON ESC/P2 Series

Совместимые с EPSON последовательные принтеры. Настройте коммуникационные параметры совпадающие с настройками принтера.
Используется протокол EPSON ESC/P2.

Контактные принтеры:
LQ-300, LQ-300+, LQ-300K+ (RS232)
LQ-300+II (RS232

Струйный принтер:

Stylus Photo 750

Лазерный принтер:

EPL-5800

.

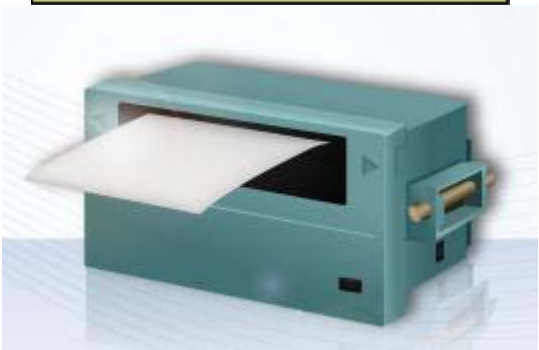
HP PCL Series (USB)

Совместимые с HP USB-принтеры, которые поддерживают протокол HP PCL5 level 3 .
● PCL 5 был обновлен в модели HP LaserJet III в марте 1990: добавлено масштабирование шрифтов Intellifont (разработанное Compugraphic, теперь принадлежит Agfa), контурные шрифты и векторная графика HP-GL/2.

● PCL 5e (PCL 5 enhanced) был обновлен в модели HP LaserJet 4 в октябре 1992: добавлена дуплексная связь между принтером и ПК и шрифты Windows

Перед присоединением к панели поддерживает ли HP-принтер PCL5 , иначе может наблюдаться черный экран панели.

SP-M, D, E, F



Последовательные принтеры. Настройте коммуникационные параметры совпадающие с настройками принтера.

Pixels of Width (Ширина в пикселях) должна быть корректно настроена и не может превышать настройки принтера по умолчанию:

100 pixels для 1610

220 pixels для 2407, 4004

EPSON ESC Protocol Serial Micro Printer:

SIUPO (Beijing)

http://www.siupo.com

SP-M, D, E, F Series

SP-E1610SK (бумага шириной 45мм), SP-E400-4S (бумага шириной 57.5мм)

Тип принтера, рекомендуемый для потребителей за пределами Китая

Axiohm A630

Микро-принтер из Франции, соединяется через последовательный порт.

Настройте коммуникационные параметры совпадающие с настройками принтера..

EPSON TM-L90

Последовательные принтеры. Настройте коммуникационные параметры совпадающие с настройками принтера.

Pixels of Width (Ширина в пикселях) должна быть корректно настроена и не может превышать настройки принтера по умолчанию“576”:

SPRT

Последовательные принтеры. Настройте коммуникационные параметры совпадающие с настройками принтера.

Pixels of Width (Ширина в пикселях) должна быть корректно настроена и не может превышать настройки принтера по умолчанию “100”:

SP-DN40SH Dot Matrix Printer

SP-RMDIII40SH Thermal

Remote Printer Server

Используйте «EasyPrinter», чтобы начать печатать на принтере соединенным с компьютером по Ethernet.
«EasyPrinter» работает под MS Windows, поэтому поддерживается большинство принтеров имеющихся на рынке.

BRIGHTEK WH-C1/C2

Последовательные принтеры. Настройте коммуникационные параметры совпадающие с настройками принтера.
Может быть выбран режим отрезки бумаги: **[No cut], [Half cut] и [Full cut]**.

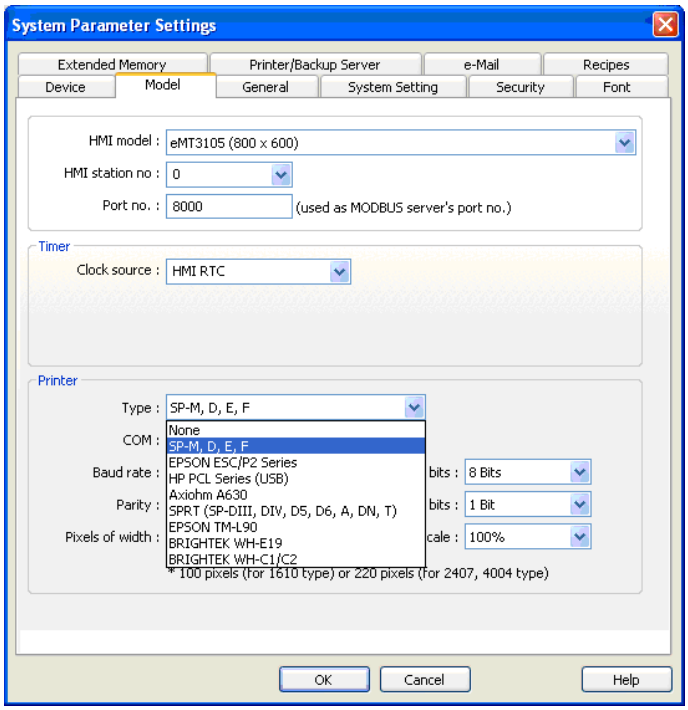
BRIGHTEK WH-E19

Последовательные принтеры. Настройте коммуникационные параметры совпадающие с настройками принтера.

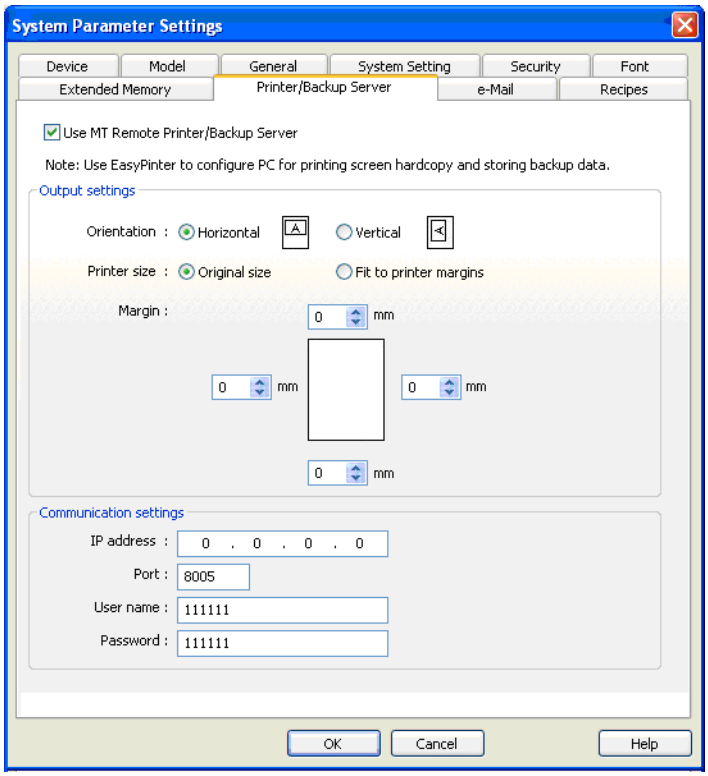
23.2 Как добавить новый принтер и начать печатать

23.2.1 Добавьте тип принтера

В **[System Parameter Settings] / [Model]** выберите тип принтера и установите соответствующие параметры:

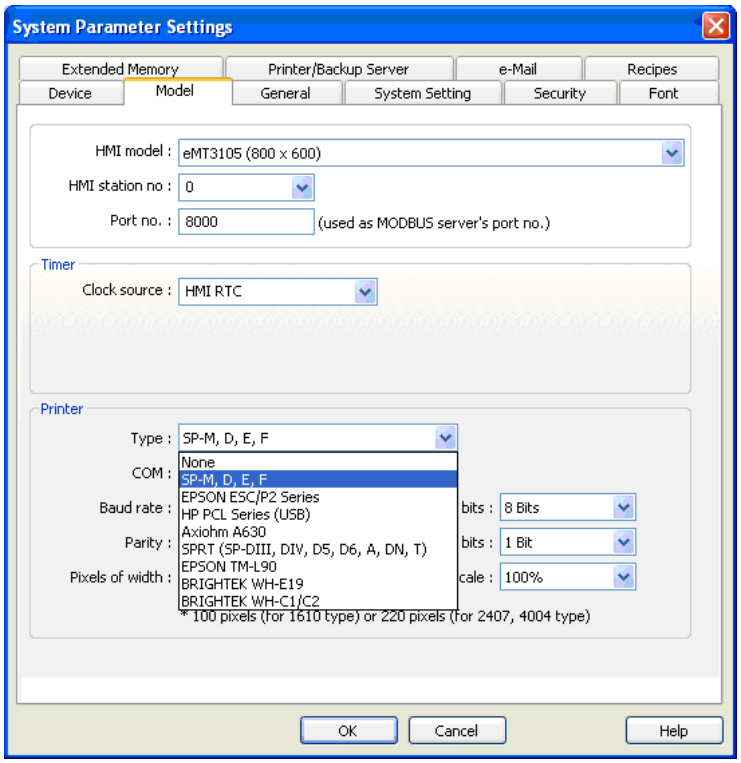


Чтобы соединится с Удаленным сервером печати (Remote Printer Server), перейдите **[System Parameter Settings] / [Printer/Backup Server]**, и настройте параметры корректно.

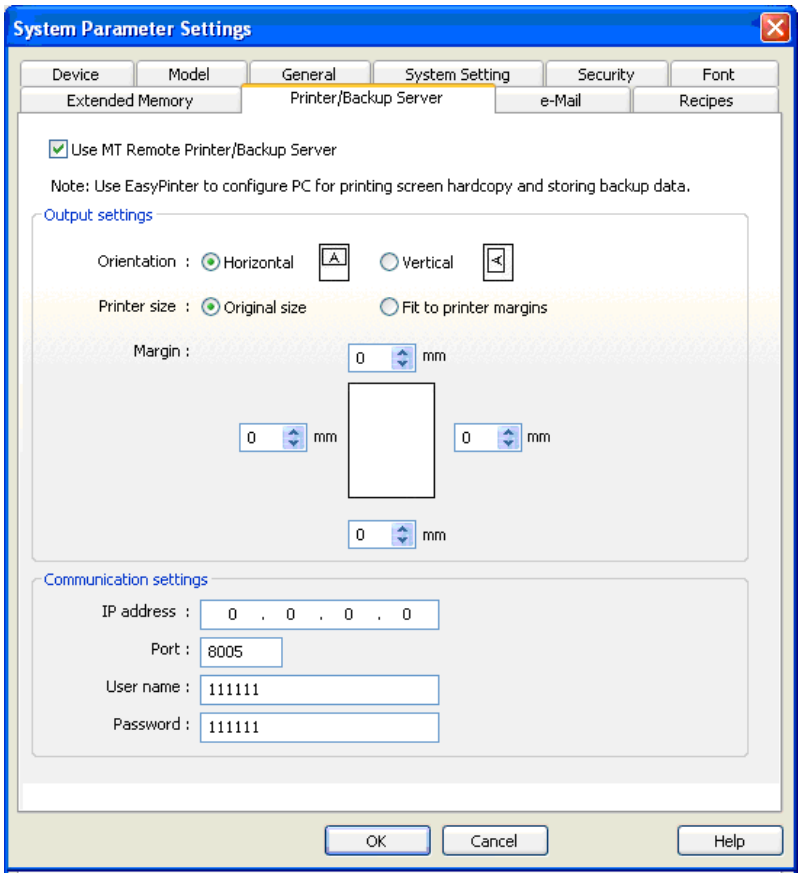


23.2.2 Начало печати

Начните печать при помощи функциональной кнопки :



Или, используйте заранее определенный битовый регистр управления «PLC Control » **[Screen hardcopy],** чтобы начать печать.



Глава 24. Редактор рецептов

24.1 Введение

Редактор рецептов предназначен для создания, просмотра и редактирования рецептов - Recipe (*.rcp) и файлв EMI (*.emi) для панелей.

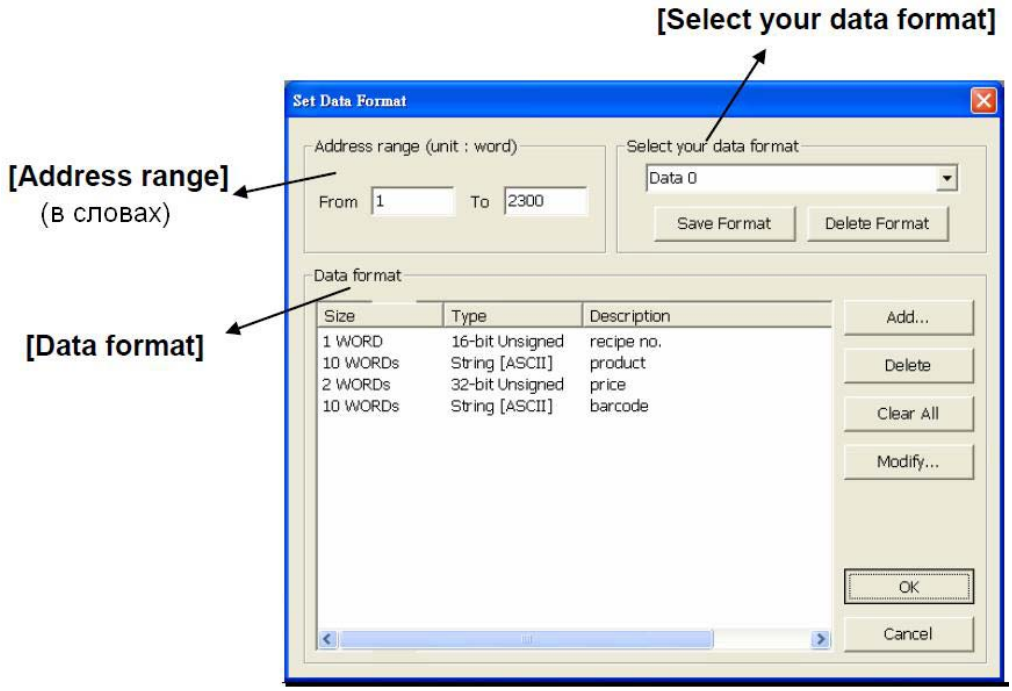
Откройте *Utility Manager* и выберите - **[Recipe/Extended Memory Editor]**.

В EasyBuilder Pro есть еще один инструмент редактирования рецептов: Recipe Records. Этот инструмент можно найти в *EasyBuilder Pro > System Paramter Settings > закладка Recipe*. Он может бытьиспользован вместе с объектом *Recipe View*. Далее следует будет описаны способы использование этих двух редакторов.

24.2 Настройка Редактора Рецептов/Расширенной памяти

Добавление файлов *.rcp / *.emi files

Set [Address Range] -> Select [Data Format] - Выберите диапазон адресов и Формат данных

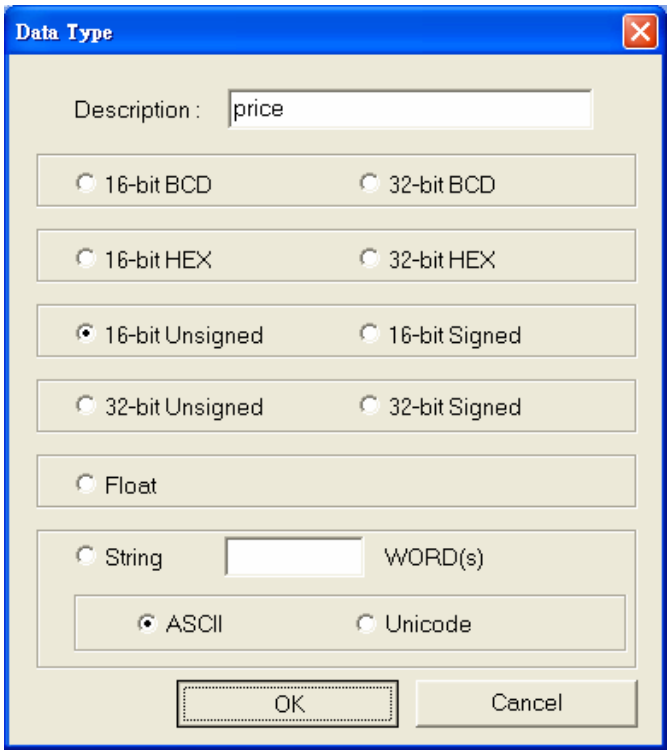


[Select your data format]

Сохраните указанный формат данных для загрузки в следующий раз. Имя сохраняемого файла: “dataEX.fmt” в инсталляционной директории EasyBuilder Pro.

Example 1

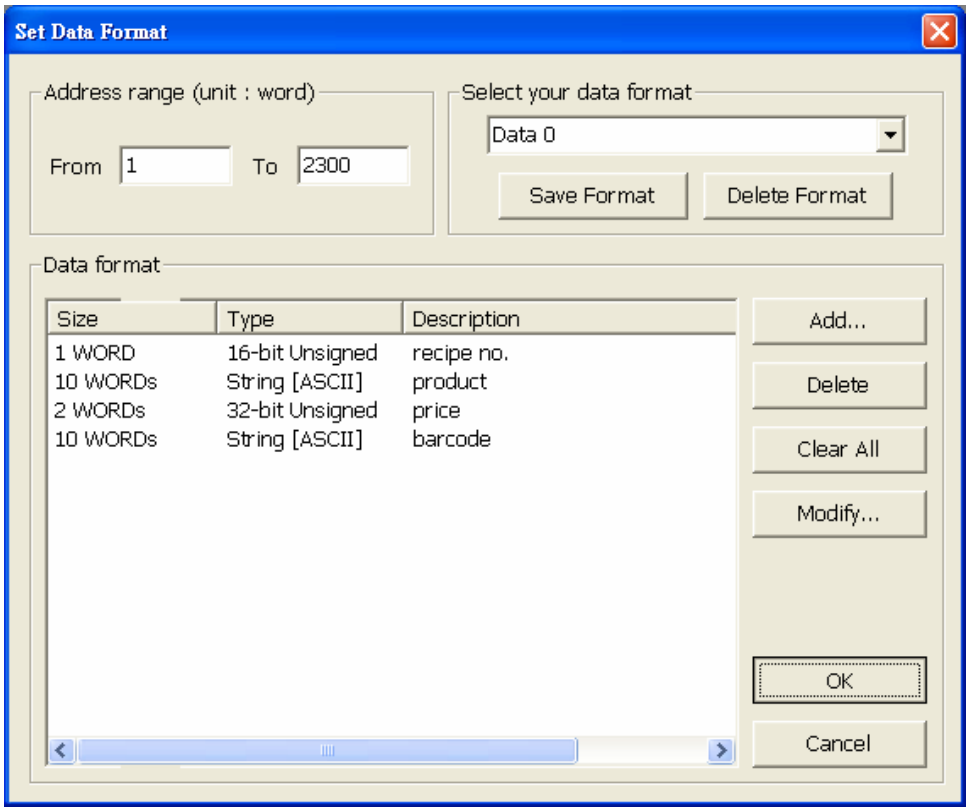
1. Щелкните по **[Add]**.

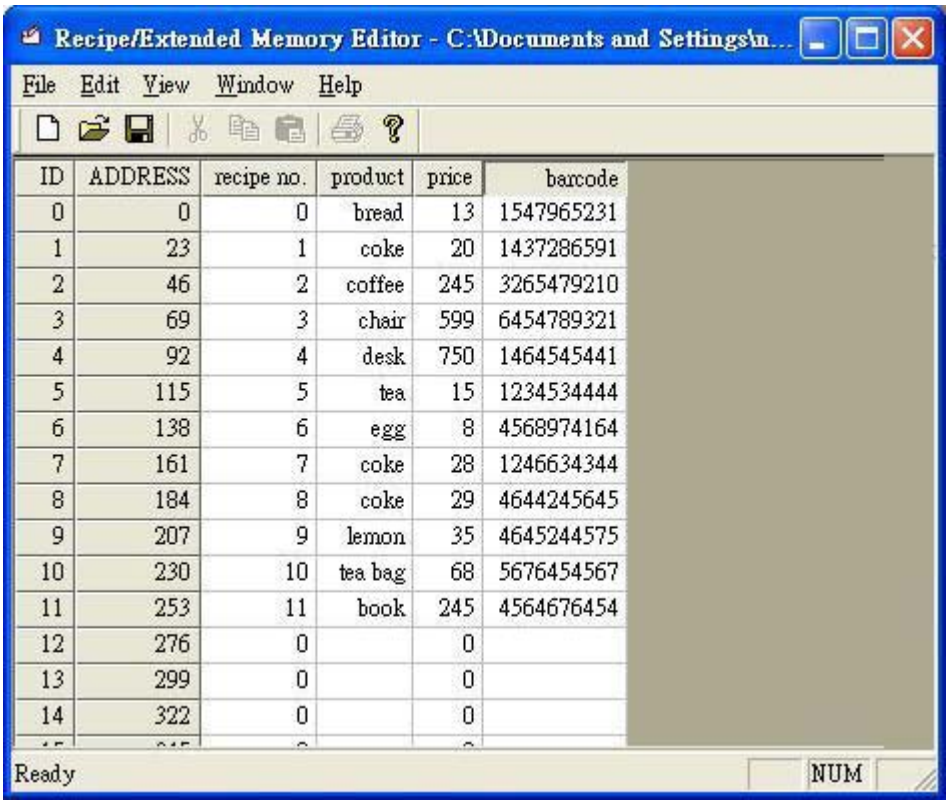


[Description]
Введите имя и тип данных.

[Data format] Выберите формат данных.
Если выбран **[String]**, введите длину (words) и кодировку «ASCII/Unicode».

2. После настройки, нажмите **[OK],** чтобы начать редактировать рецепт





ID	ADDRESS	recipe no.	product	price	barcode
0	0	0	bread	13	1547965231
1	23	1	coke	20	1437286591
2	46	2	coffee	245	3265479210
3	69	3	chair	599	6454789321
4	92	4	desk	750	1464545441
5	115	5	tea	15	1234534444
6	138	6	egg	8	4568974164
7	161	7	coke	28	1246634344
8	184	8	coke	29	4644245645
9	207	9	lemon	35	4645244575
10	230	10	tea bag	68	5676454567
11	253	11	book	245	4564676454
12	276	0		0	
13	299	0		0	
14	322	0		0	

3. В этом примере, общая длина формата данных - 23 слова и будет представлена, как один набор данных рецепта (set of recipe data).

Первый набор: “recipe no.” = address 0, ”product” = address 1 ~ 10, ”price” = address 11 ~ 12, "barcode” = address 13 ~ 22;

Второй набор: “recipe no.” = address 23, ”product” = address 24 ~ 33, "price” = address 34 ~ 35, "barcode” = address 36 ~ 45…и так далее.

■ После редактирования данных рецепта, он может быть сохранен, как файл с расширениями *.rcp, *.emi, или *.csv.

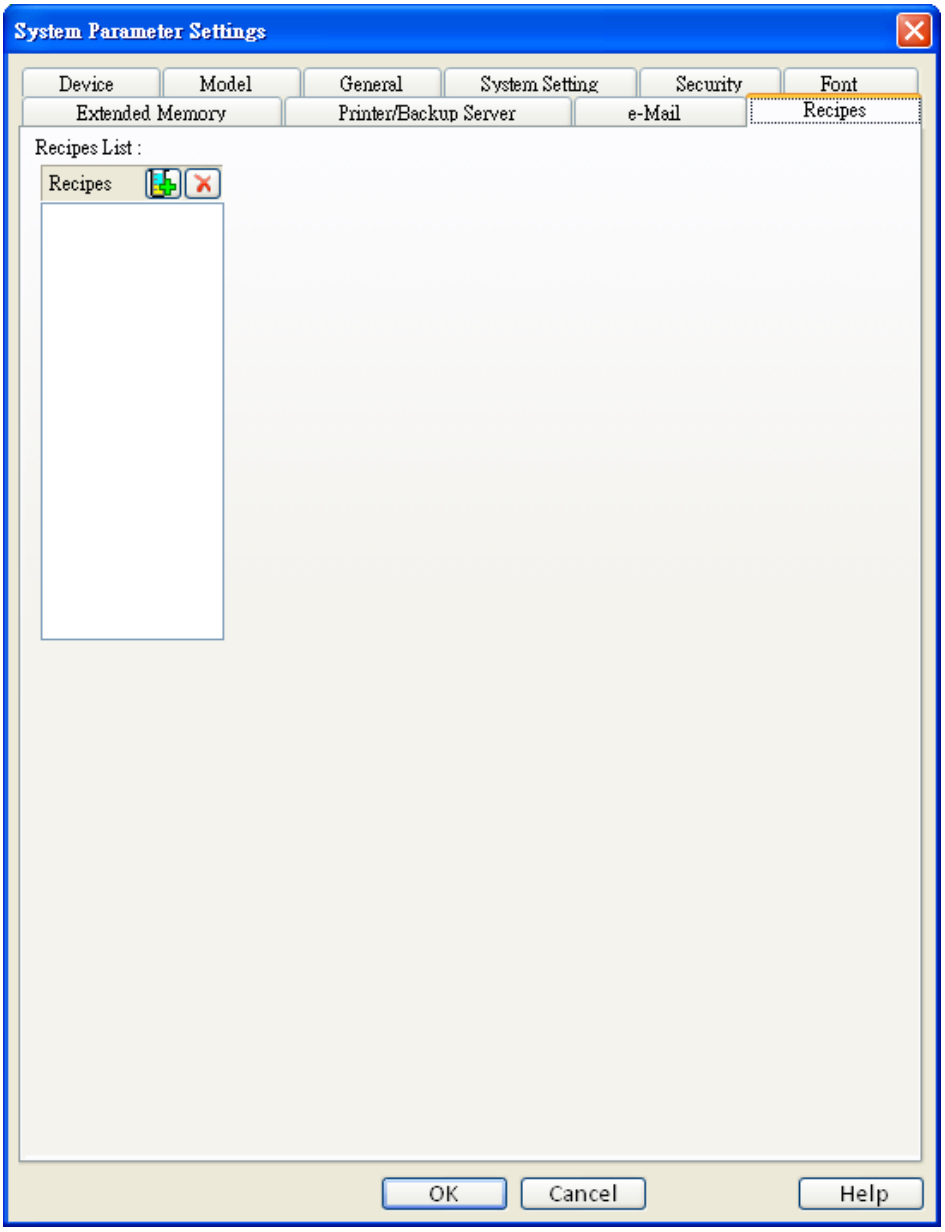
Файл *.rcp может быть загружен в панель при помощи *Utility Manager* или во внешнее устройство (USB или SD).

Файл *.emi может быть прямо сохранен на внешнем устройстве и устройство можно вставить в панель для чтения (EM register).

24.3 Записи рецептов

Перед использованием «Recipe Records», закончите настройки в **EasyBuildr Pro / System Parameter Settings / Recipe.**

Детали см. “Глава 5 Настройка системных параметров”.



После настройки системных параметров, окно «Recipe Records» может быть открыто щелчком по пиктограмме в основном меню EasyBuilder Pro . В примере на следующей странице, *Recipe1* и *Recipe2* содержат три позиции, показанных на правой стороне. Имя рецептов получено из настроек системных параметров.

Далее показано, как вставить записи в рецепт в соответствии с форматом данных позиции (item).

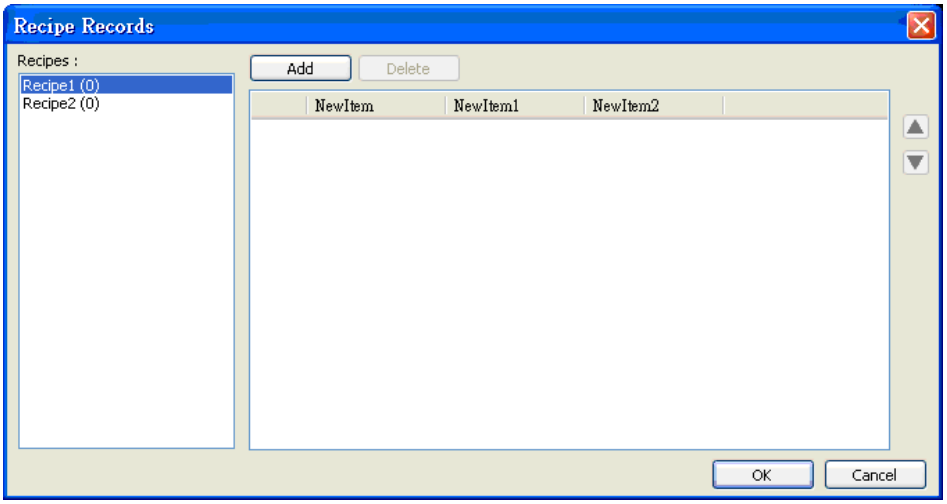
Add] / [Delete]

Нажмите [Add], чтобы вставить записи в рецепт в соответствии с форматом элемента.

Нажмите [Delete], чтобы удалить редактируемое содержимое.

Recipe list.

Список рецептов показывает рецепты созданные в «System Parameter Settings». Число в скобках показывает общее число записей в соответствующем рецепте.

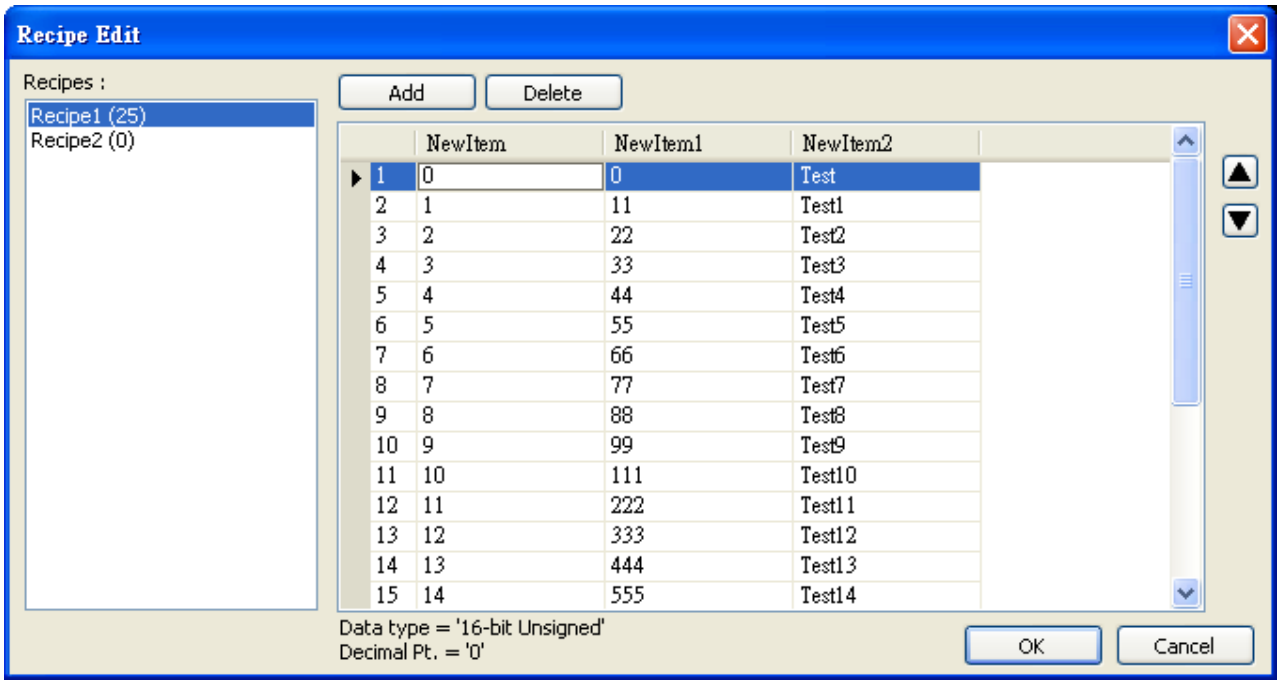


Click the up and down arrows to select the record to be edited.

Example 1 Нажмите кнопку **[Add],** чтобы добавить новую запись и начать редактировать каждый элемент**.**

В нижней части окна, появляется формат элемента, если щелкнуть по элементу. Это помогает пользователю заполнить каждый элемент допустимыми значениями.

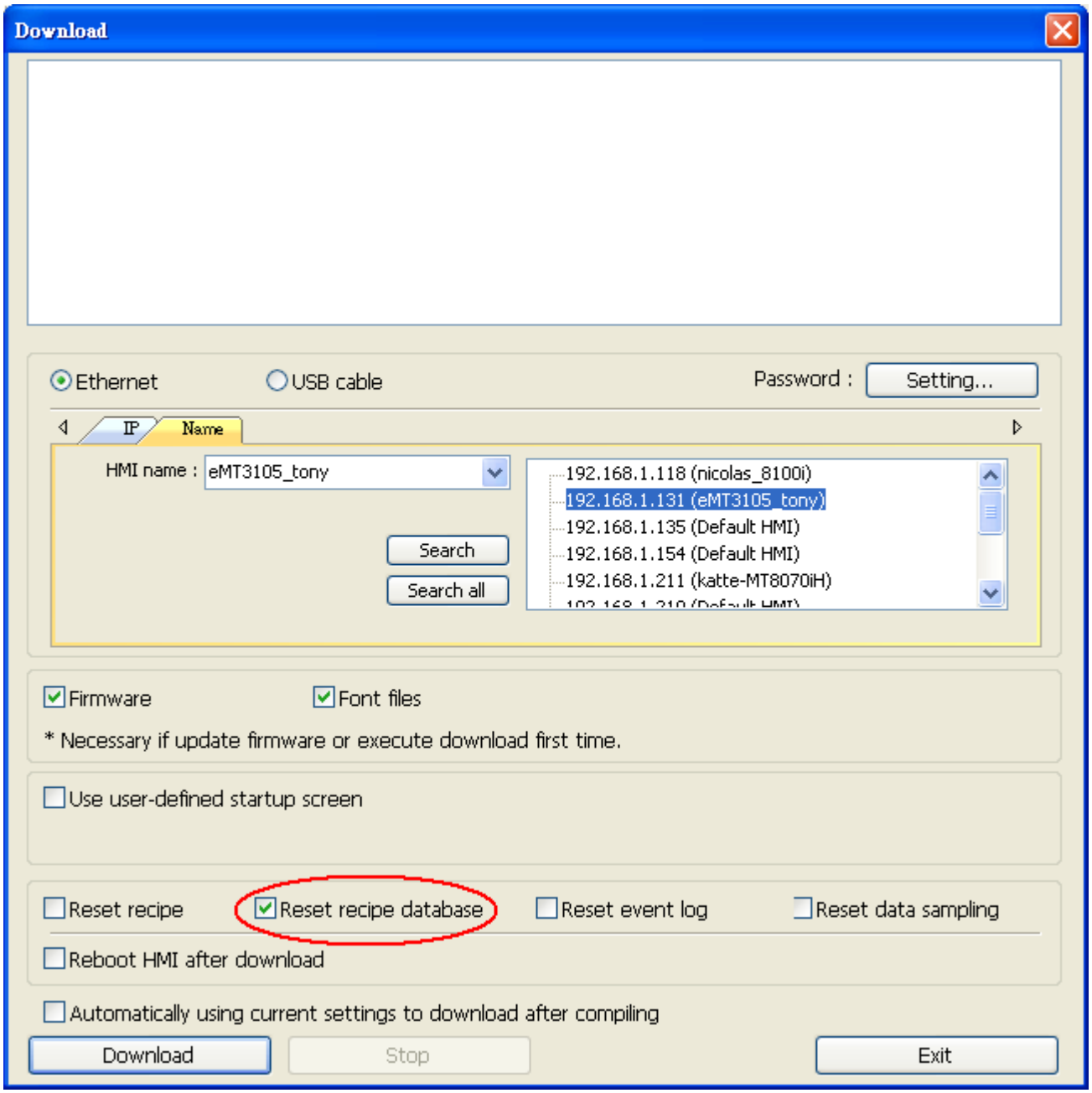
Нажмите **[OK],** для подтверждения и сохранения записей.



■ Если рецептов много, каждый рецепт может содержать максимально 10000 записей.

■ Записи рецептов будут сохранены в файле *.xob после компиляции и будут загружены в панель. Эти рецепты нельзя объединить с другими файлами проекта.

Если пользователю необходимо модифицировать рецепт, используя «Recipe Records» и загрузить в проект, сделайте отметку в окошке **[Reset recipe database]** . Если это не сделать, то база данных рецептов в панели не будет обновлена.



Глава 25. EasyConverter

Это программное приложение используется, когда архивная запись выборки данных (dtl) или архив событий (evt) из панели необходимо сохранить в Excel.

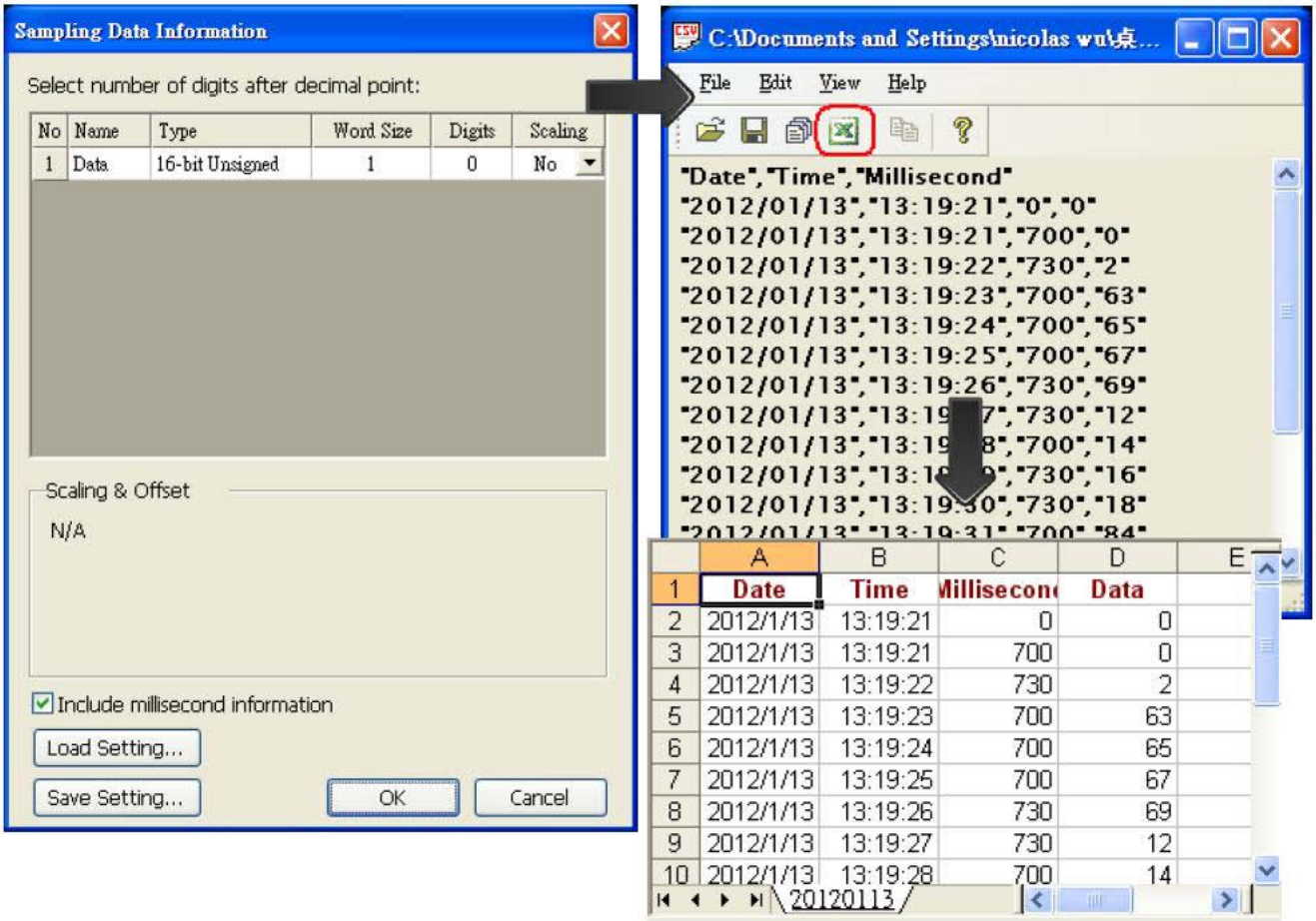
Запустить “EasyConverter” можно:

■ Из **Utility Manager**, щелкнув **EasyConverter**

■ Из меню **EasyBuilder Pro**, щелкнув **Tool / Data/Event Log Converter**

25.1 Экспорт файлов DTL или EVT в Excel

1. **[EasyConverter] / [Open] / [OK]**
2. Щелкните **[Export to Microsoft Excel]**

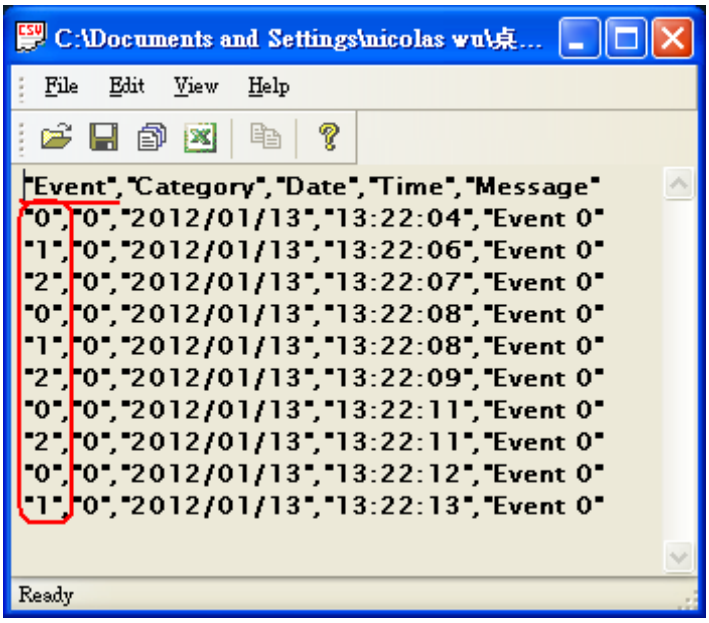


Открыв «event log», поле **[Event]** может быть найденов *EasyConverter*, как показано ниже.

0 -> Event triggered – Событие произошло;

1 -> Event acknowledged – Событие квитпровано;

2 -> Event returns to normal – Событие вошло в норму



25.2 Использование функции масштабирования

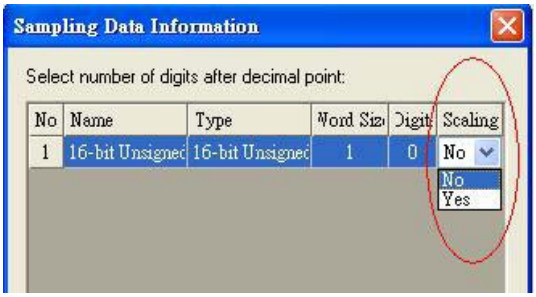
Функция ***scaling*** используется для коррекции данных.

Новое значение = [(value + A) x B] + C, пользователь может задавать значения A, B и C.

Где: A: нижний предел *value* ;

B: [(верхний предел шкалы) - (нижний предел шкалы) / (верхний предел) - (нижний предел)];

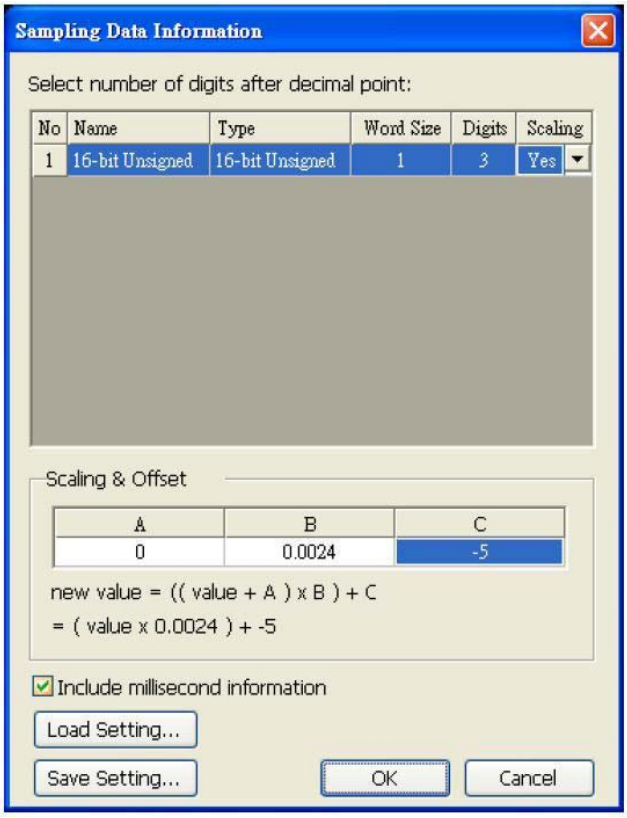
C: нижний предел шкалы

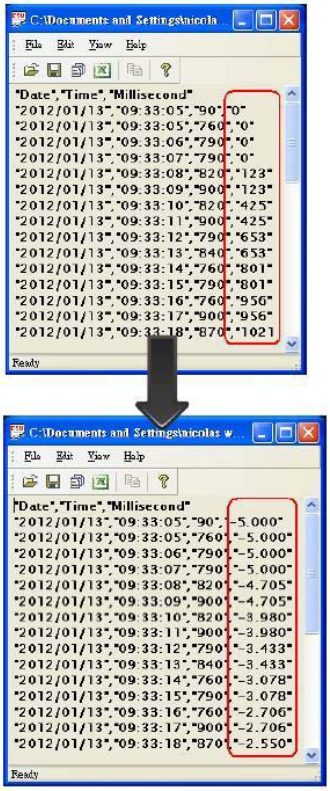


Example 1 Например, напряжение, представлено в формате 16-bit unsigned (16-разрядное беззнаковое) (диапазон значений: 0~4096).

Пользователю необходимо значение в вольтах в диапазоне от -5 В до +5 В.

Новое значение = [(value + 0) x 0.0024] + (-5):





Note

■ Настройки данных можно сохранить, как образец и загрузить в любой момент времени. Имя файла примера: *.LGS.

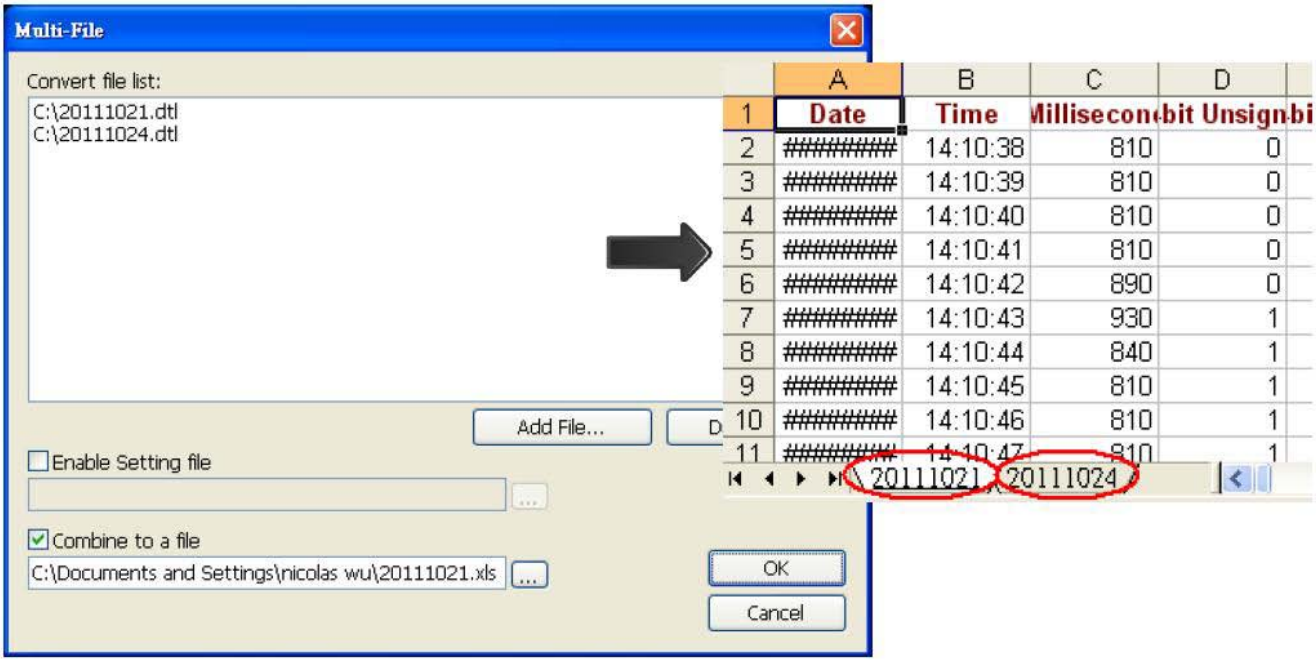
■ После настройки параметров масштабирования, нажмите **[Save Setting]** и в новом примере, щелкните **[Load Setting] ,** чтобы использовать выше приведенный пример.

25.3 Использование преобразования нескольких файлов (Multi-File Conversion)

Example 1

1: Щелкните **[File] / [Multi-File]** / **[Add File],** чтобы скомбинировать несколько файлов в один файл Excel .

2: Щелкните **[Combine to a file]**, файлы будут разделены на листы в одном файле EXCEL (*.XLS) с именем даты создания. Если пользователь не сделает эту отметку, файлы будут экспортированы в Excel индивидуально.

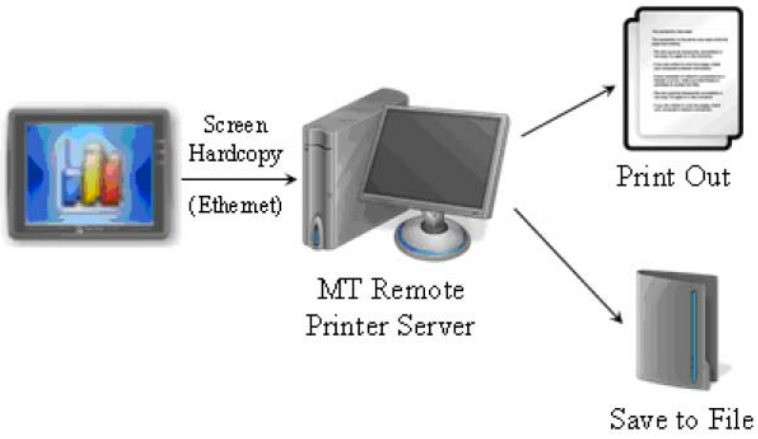


Сохраненные настройки файлов могут быть загружены для комбинирования:

Отметьте окошки **[Enable Setting file]** и **[Combine to a file]** и выберите файлы для комбинирования, затем нажмите **[OK]**.

Глава 26. EasyPrinter

EasyPrinter – это программное приложение Win32 и может работать только под MS Windows 2000 / XP / Vista / 7. Оно позволяет выводить печатные копии экранов панелей на удаленный компьютер по сети Ethernet. Посмотрите на рисунок внизу.



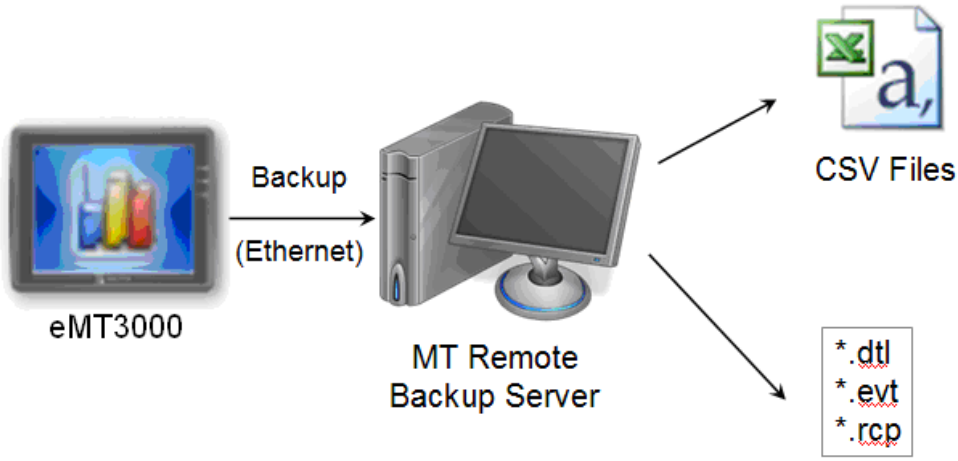
Использование EasyPrinter дает следующие приимущества:

EasyPrinter имеет два режима печати: Print-Out (Вывод на печать) и Save-to-File (Сохранение в файл).

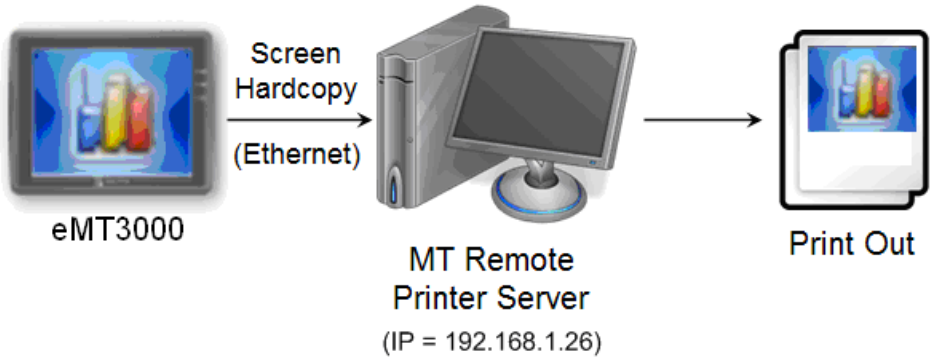
Так как EasyPrinter работает под MS Windows, он поддерживает большинство принтеров доступных на рынке.

Несколько панелей имогут использовать один принтер при помощи EasyPrinter. Пользователю не надо подготавливать принтер для каждой панели.

Дополнительно, EasyPrinter может быть резервный сервер (backup server). Пользователи могут использовать объекты резервирования (backup objects) в панели, чтобы копировать архивные файлы, такие, как Data-Sampling или Event-Log в удаленный компьютер по сети Ethernet. Посмотрите на рисунок внизу.



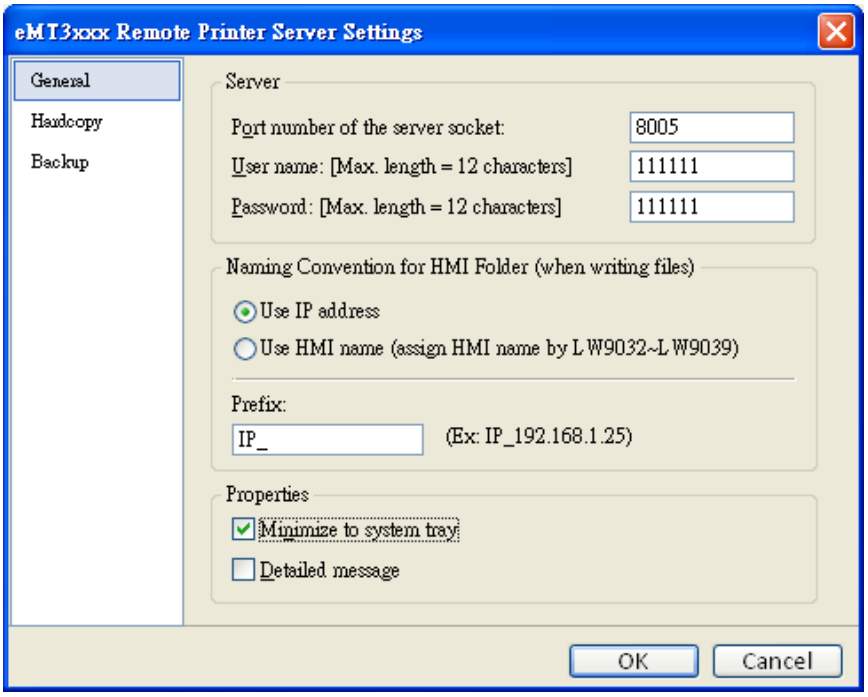
26.1 Использование EasyPrinter в качестве Сервера печати



Пользователь может производить вывод на печать при помощи объекта **[Function Key]**. Печатные копии будут переданы в MT Remote Printer Server по сети Ethernet и , затем, выведены на печать.

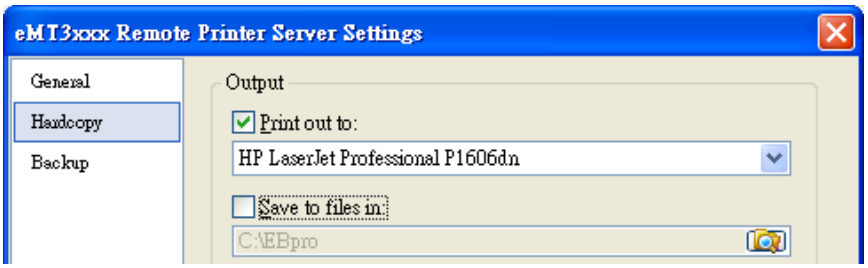
26.1.1 Установка EasyPrinter

В **[Menu]** ➔ **[Options]**, выберите **[Settings…],** появится следующее окно:



1. В **[Server]**, присвойте **[Port number of the server socket]** - “8005”, **[User name]** - “111111” и **[Password]** - “111111”. (Это значения по умолчанию.)
2. В **[Naming Convention for HMI Folder]**, выберите **[Use IP address]** и присвойте “IP_”, как префикс **[Prefix]**.
3. В **[Properties]**, выберите **[Minimize to system tray]**.

Выберите закладку **[Hardcopy]** на левой стороне окна, как показано:



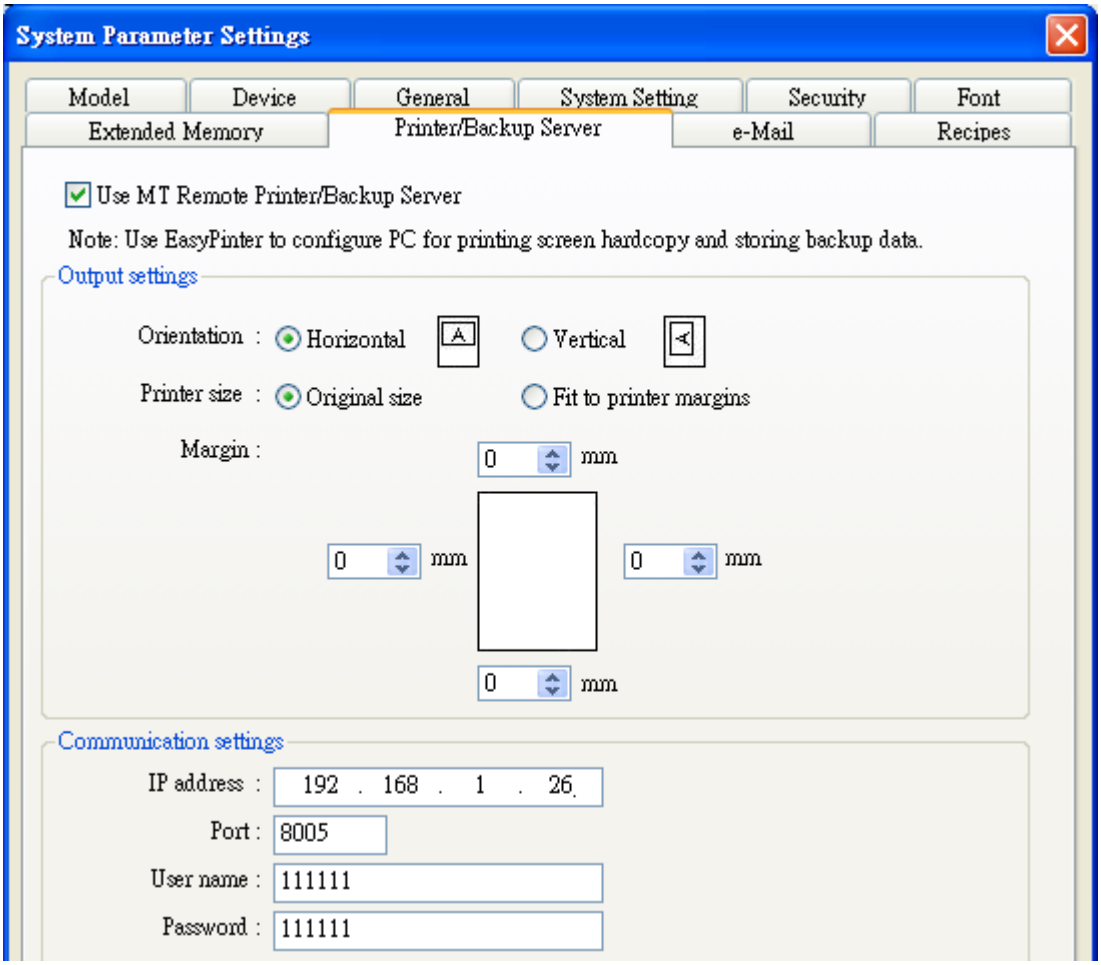
4. В **[Output]**, выберитет **[Print out to]** и выберите принтер, как устройство вывода копий экрана. Примечание: Пользователи могут выбирать только из принтеров доступный системе, поэтому возможно, что “hp LaserJet 3380 PCL 5” не будет в вашем списке.)

5. Нажмите **[OK],** чтобы применить настройки.

6. В **[Menu]** ➔ **[File]**, выберите **[Enable Output],** чтобы разрешить EasyPrinter вывод любого запрос на печать, например, копии экрана.

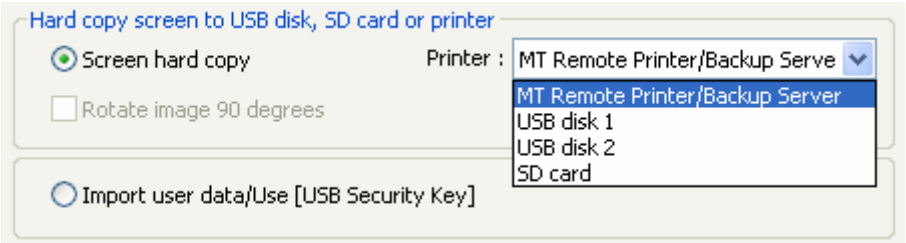
26.1.2 Процедура настройки в EasyBuilder Pro

В EasyBuilder Pro **[Menu]** ➔ **[Edit]** ➔ **[System Parameters]**, откройте закладку **[Printer Server] и** выберите **[Use MT Remote Printer/Backup Server]**, откоется следующее окно:



7. В **[Output settings]**, установите соответствующие размеры полей страницы.
8. В **[Communication settings]**, заполните **[IP address]** сервера печати (как в п.1), присвойте **[port number]** - “8005”, **[User name]** - “111111” и **[Password]** - “111111”.

В EasyBuilder Pro **[Menu]** → **[Objects]** → **[Buttons]**, выберите **[Function Key]** и присвойте **[Screen hardcopy]** серверу **[MT Remote Printer/Backup Server]**.

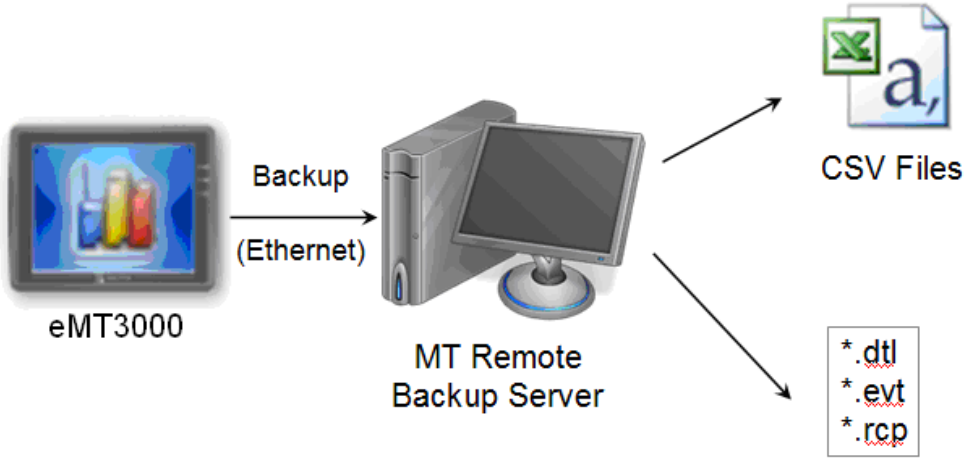


9. Поместите объект **[Function Key]** на общее окно (window no. 4), и пользователи смогут делать копии экрана в любое время.
10. **[Compile]** и **[download]** проект в панель. Нажмите объект **[Function Key]** (созданный в п.9), чтобы сделать копию экрана.

NOTE

5. Пользователи могут использовать, также, объект **[PLCControl],** чтобы делать копии экрана.
6. Пользователи НЕ могут печатать аварийные сообщения через EasyPrinter.
7. EasyPrinter может соединяться с панелью, только по Ethernet, поэтому выбирайте панели с поддержкой Ethernet

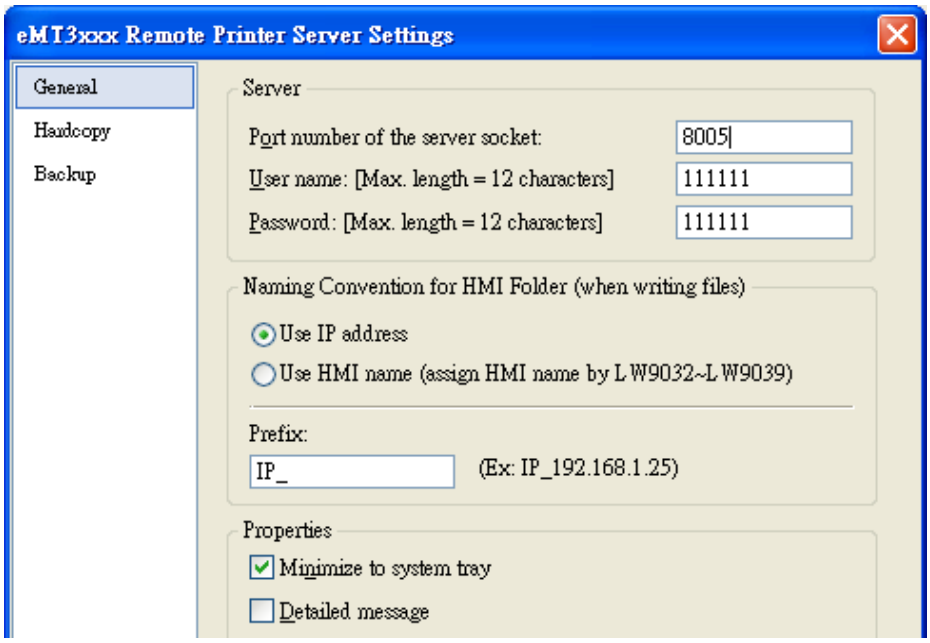
26.2 Использование EasyPrinter в качестве Резервного сервера



Пользователи могут выгружать архивную информацию (файлы Data-Sampling и Event-Log) в резервный сервер *MT remote backup server* при помощи объектов **[Backup]** .

26.1.1 Процедура установки в EasyPrinter

В **[Menu]** ➔ **[Options]**, выберите **[Settings…],** появится следующее окно:

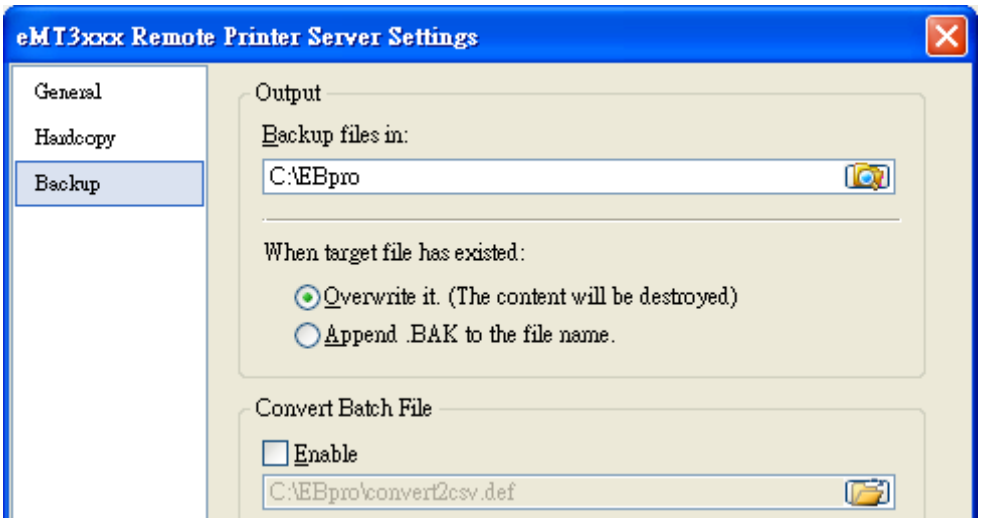


1. В **[Server]**, присвойте **[Port number of the server socket]** - “8005”, **[User name]** - “111111” и **[Password]** - “111111”. (Это настройки по умолчанию.)

2. В **[Naming Convention for HMI Folder]**, выберите **[Use IP address]** и присвойте “IP_” , как **[Prefix]**.

3. В **[Properties]**, выберитет **[Minimize to system tray]**.

   Нажмите закладку **[Backup]** на левой стороне окна, появится следующее окно:



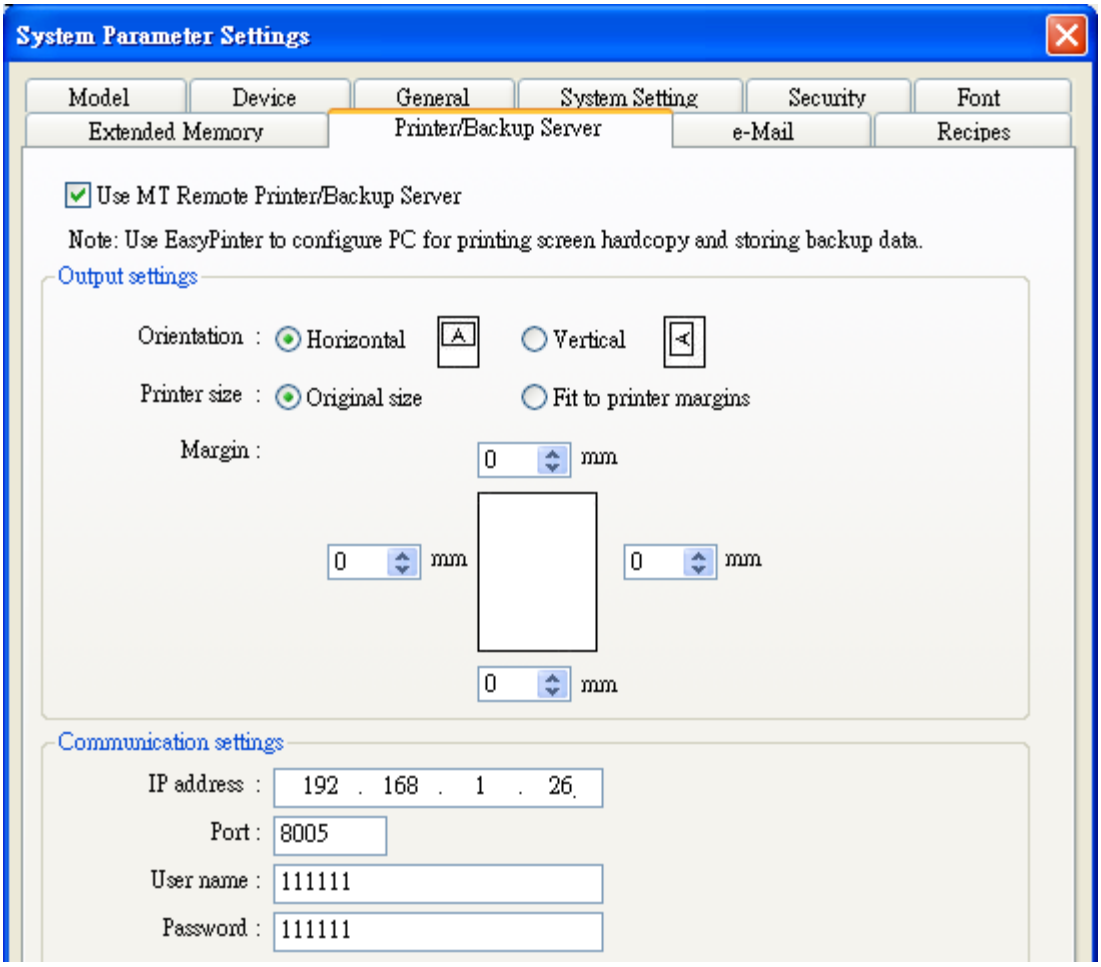
4. В **[Output]**, нажмите кнопку «browse» и укажите путь для хранения входящих архивных файлов.

5. Нажмите **[OK],** чтобы применить настройки.

6. В **[Menu]** ➔ **[File]**, выберите **[Enable Output],** чтобы разрешить EasyPrinter сохранять любой запрос на резервирование в месте, указаппом в п.4.

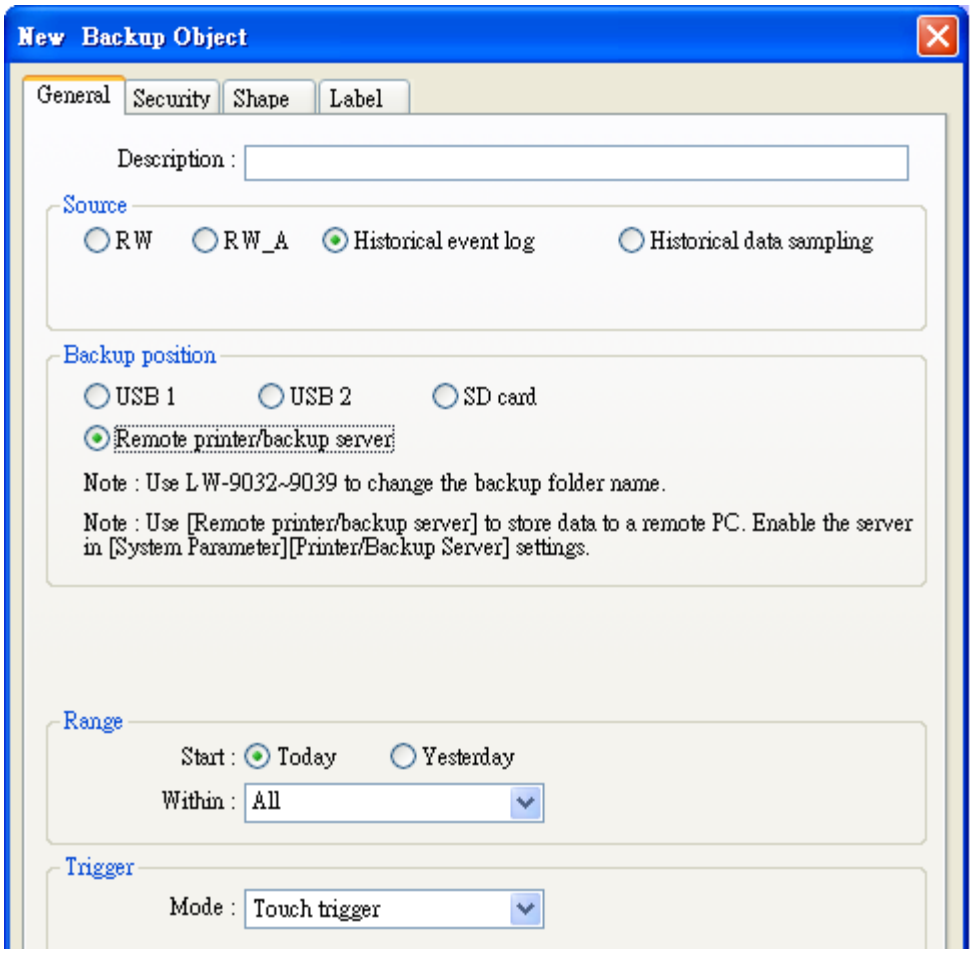
26.2.2 Процедура установки в EasyBuilder Pro

В EasyBuilder Pro **[Menu]** ➔ **[Edit]** ➔ **[System Parameters]**, откройте закладку **[Printer Server]** и выберите**[Use MT Remote Printer/Backup Server]**, откроется следующее окно:



7. В **[Communication settings]**, заполните **[IP address]** сервера печати (как в п.1), присвойте **[port number]** - “8005”, **[User name]** - “111111” и **[Password]** - “111111”.

В EasyBuilder Pro **[Menu] ➔[Objects]**, выберите **[Backup],** откроется следующее окно:



8. В [Source], выберите **[Historical event log]**.
9. В [Backup position], выберите **[Remote printer/backup server]**.
10. В [Range], выберите **[Today]** and **[All]**.
11. В [Trigger], выберите **[Touch trigger]**.
12. Рзместите объект **[Backup]** в общем окне (window no. 4), и пользователи смогут делать резервные копии в любой момент времени.
13. **[Compile]** и **[download]** проект в панель. Нажмите на объект **[Backup]** (созданный в п.12), чтобы сделать резервную копию архива Event-Log.

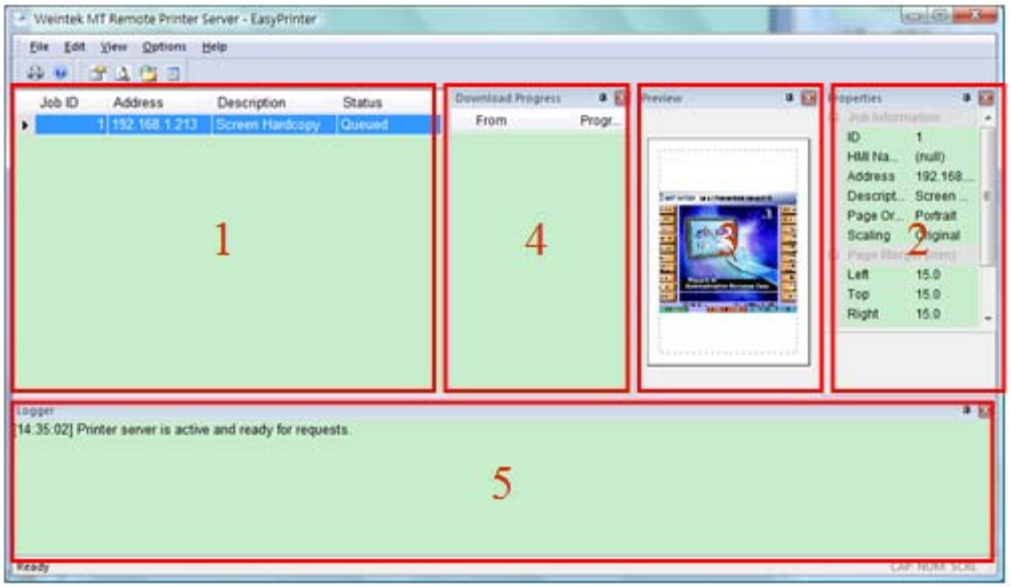
NOTE

8. Объект **[Backup]** может срабатывать от битового сигнала.

9. Пользователь может настроить объект **[Scheduler]**, который переведет бит в состояние ON в конце недели, чтобы сработал объект **[Backup]** для автоматического создания резервных копий всех архивных файлов.

26.3 Применение EasyPrinter

26.3.1 Внешний вид



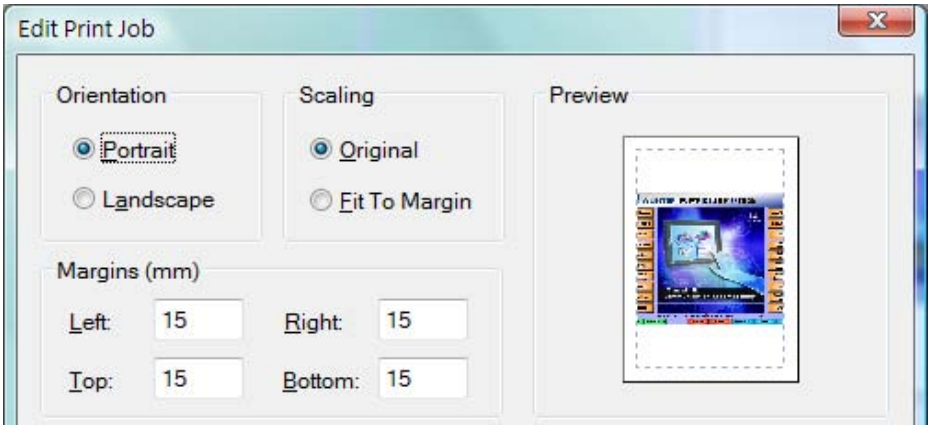
Участок	Имя	Описание
1	Job List – Рабочий список	В этом окне показан список всех входящих задач, например, запросов на копии экрана или на резерное копирование.
2	Property Window – Окно свойств	В этом окне показана информация обо всех задачах, выбранных из списка “Job List.”
3	Preview Window – Окно просмотра	В этом окне показано изображение копии экрана задачи выбранной из “Job List.”
4	Download Progress Window –Окно индикации загрузки	В этом окне показано состояние процесса загрузки водящих запросов.
5	Message Window – Окно сообщений	В этом окне показано время и сообщения о событиях, таких, как входящий запрос, неверный пароли, и т. д.

26.3.2 Руководство по применению

Menu →File	**Описание**
Enable Output	● Selected - Выбрано EasyPrinter обрабатывает задачи последовательно. ● Unselected – Не выбрано EasyPrinter сохраняет входящие задания в памяти.

NOTE

10. EasyPrinter может зарезервировать до 128 MB памяти для данных в задачах. Если память заполнена, любой приходящий запрос в дальнейшем будет отвергнут и пользователь должен управлять **[Enable Output]** или удалить некоторые задачи, освободив место для новых задач.

Menu →File	**Описание**
Enable Output – Вывод разрешен	Для редактирования задачи на создание копии экрана для печати.  Здесь пользователь может изменять: **[Orientation]**, **[Scaling] и [Margins]** – Ориентацию, масштаб и поля.
Delete	Для удаления выбранных задач постоянно.
Select All	Для выбора всех задач “Job List.”

NOTE

11. Задачи резервирования не редактируемы.

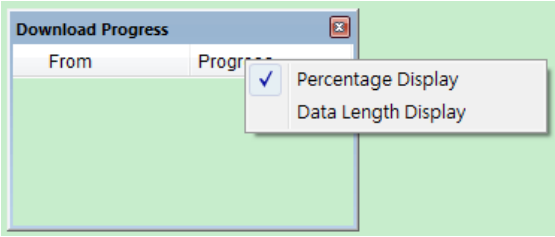
12. Функция **[Edit]** доступна только, когда задача выбрана.

13. Функция **[Delete]** доступна только тогда, когда по крайней мере одна задача выбрана.

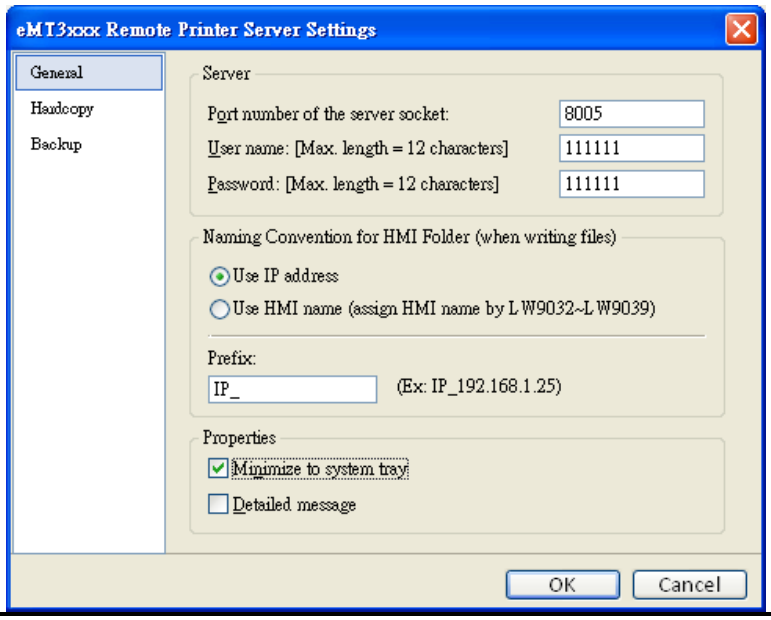
Menu →File	Описание
Properties Bar	Для открытия или скрывания окна *Property Window*.
Preview Bar	Для открытия или скрывания окна *Preview Window*.
Download Bar	Для открытия или скрывания окна *Download Progress Window*.
Logger Bar	Для открытия или скрывания окна *Message Window*.

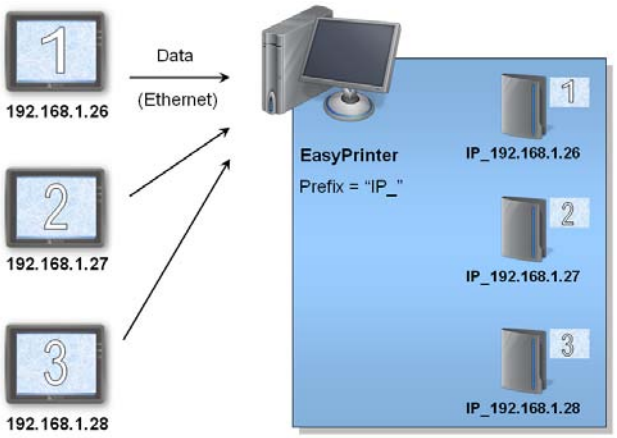
NOTE

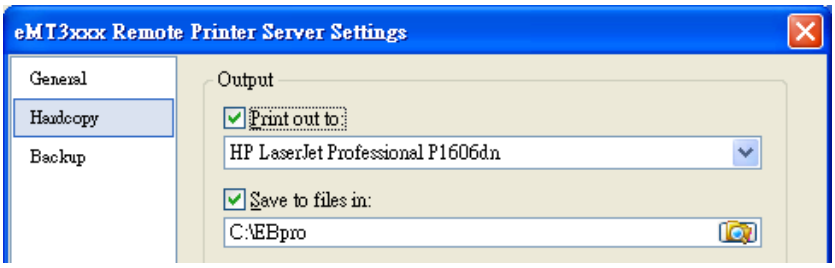
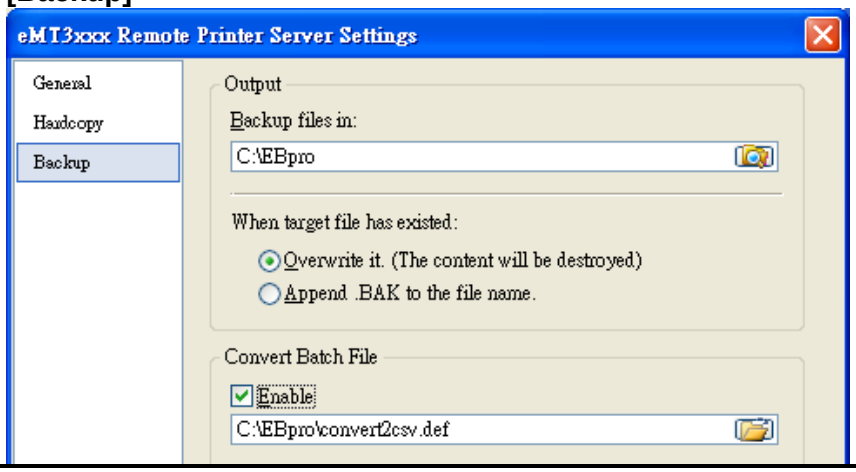
14. В окне **[Download Progress]**, пользователь может выбрать способ показа процесса загрузки, щелкнув по заголовку колонки **[progress]**:



15. EasyPrinter может обслужить до 10,000 сообщений в окне *Message Window*. Если новое сообщение придет, самое старое сообщение будет удалено.

Menu →File	Описание
Toolbars	Для открытия или скрывания панелей инструментов - *toolbars*.
Status Bar	Для открытия или скрывания панели состояния - *status bar*.
Settings	Настройки EasyPrinter. Посмотрите на следующие рисунки: **[General]** 

Menu →File	Описание
Settings	**[Server] →[Port number of the server socket]** Установите номер *Ethernet socket* для связи с панелью. Диапазон от 1 до 65535; значение по умолчанию - 8005. ● **[Server] →[User name] & [Password]** Укажите *user name* и *password* , чтобы разрешить посылать запросы в EasyPrinter только авторизованным панелям. ● **[Naming Convention for HMI Folder]** EasyPrinter создает различные папки для хранения файлов (например. BMP-файлов копий экранов, резервных файлов) from different HMI. Существует два пути наименования папок: **a. Используя IP адрес** EasyPrinter называет папку по IP- адресу панели, посылающей запрос. (например, [Prefix] + [IP address]), см. рисунок:  **b. Используя имя панели** EasyPrinter называет папку Имени панели, посылающей запрос. (например,. [Prefix] + [HMI name]) ● **[Properties] →[Minimize to system tray]** Выберите этот вариант для отображения пиктограммы EasyPrinter в *system tray,* вместо панели задач. Двойным щелчком по этой пиктограмме, пользователь может развернуть окно EasyPrinter. ● **[Properties] →[Detailed message]** Выберите этот вариант для отбражения более детальных сообщений о событиях в окне - *message window*.

Menu →File	**Описание**
Settings	**[Hardcopy]**  ● **[Output]** EasyPrinter обеспечивает два режима вывода результатов на печать: *Print-Out* и *Save-to-File*. **a. Print-Out** Выбор этого варианта информирует EasyPrinter, что печатную копию надо вывести на указанный принтер. **b. Save-to-File** Выбор этого варианта информирует EasyPrinter что печатную копию надо преобразовать в файл «bitmap» и сохранить его в указанной директории. Пользователь сможет найти файлы *bitmap* : [Specified Path] → [HMI Folder] → yymmdd_hhmm.bmp Например, по запросу на печатную копию от 17:35:00 12/Jan/2009, файл *bitmap* будет назван “090112_1735.bmp”. Если есть другой файл *bitmap* созданный в ту же минуту, он будет назван “090112_1735_01.bmp” и так далее. **[Backup]** 

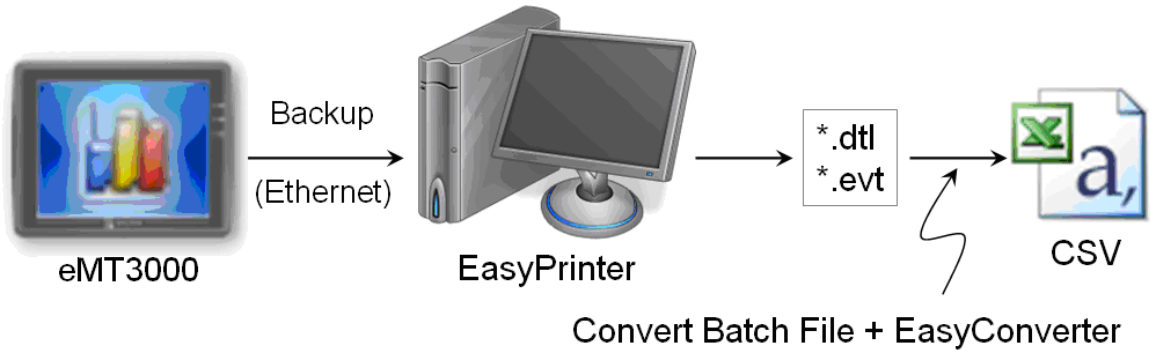
Menu →File	**Описание**
Settings	● **[Output]** EasyPrinter сохраняет резервные файлы по указанным путям. Архивные файлы *Event-Log* : [Указанный путь] → [HMI Folder] → [eventlog] → EL_yyyymmdd.evt Архивные файлы *Data-Sampling* : [Указанный путь] → [HMI Folder] → [datalog] → [Имя папки объекта Data-Sampling]→ yyyymmdd.dtl Файлы рецептов: [Указанный путь] → [HMI Folder] → [recipe] → recipe.rcp или recipe_a.rcp ● **[Convert Batch File]** Выберите **[Enable]** и назначте *Convert Batch File* для автоматического конвертирования выгружаемых архивных файлов в файлы формата CSV или MS Excel . В следующем разделе приведены подробности использования *Convert Batch File*.

NOTE

16. Пользователи могут назначить имена панелей от LW9032 до LW9039.
17. EasyPrinter именует папки по IP-адресу, если имя панели не введено.

26.4 Командный файл для конвертирования

EasyPrinter снабжен механизмом для автоматического конвертирования выгружаемых архивных файлов *Data-Sampling* и *Event-Log* history (которые храняться в двоичном коде) в файлы CSV. Для использования этой функции пользователь должен приготовить *Convert Batch File* (Командный файл для конвертирования), чтобы снабдить EasyPrinter информацией о конвертировании архивных файлов.



Как показано на рисунке, преобразование происходит в EasyConverter. EasyPrinter просто следует указаниям (criteria) Convert Batch File и активирует EasyConverter с соответствующими аргументами преобразования.

NOTE

18. EasyConverter –это приложение Win32 для преобразования архивных файло в файлы CSV или MS Excel (*.xls). EasyConverter находится в инсталляционной директории EasyBuilder Pro.
19. Для использования этой функции EasyPrinter и EasyConverter должны быть размещены в одной и той же директории.

26.4.1 Convert Batch File (по умолчанию)

В пакет EasyBuilder Pro включен по умолчанию следующий Convert Batch File:

Default Convert Batch File (convert2csv.def)

1: "dtl", "EasyConverter /c $(PathName)"

2: "evt", "EasyConverter /c $(PathName)"

В файле две строки текста. В каждой строке по два аргумента разделенных запятой, которые формируют критерий обработки указанного типа файла (“dtl” для архивных файлов *Data-Sampling* , “evt” для архивных файлов *Event-Log*). Первый аргумент указывает имя расширения обрабатываемого файла, а второй указывает точную команду для выполнения в консольном режиме. “$(PathName)” - это ключевое слово, указывающее EasyPrinter заменить его реальным именем резервируемого файла для преобразования.

Например, если архивный файл Data-Sampling названный «20090112.dtl» выгружен и сохранен, EasyPrinter пошлет следующую команду в окно консоли:

EasyConverter /c 20090112.dtl

И , затем, файл CSV с именем 20090112.csv будет создан.

Следовательно, критерий Convert Batch File такой:

1. Преобразовать все архивные файлы Data-Sampling (*.dtl) в файлы CSV.

2. Преобразовать все архивные файлы Event-Log (*.evt) в файлы CSV.

NOTE

20. В действительности, “$(PathName)” во втором аргументе это место для полного пути к файлу. В первом случае, EasyPrinter заменит его на:
[Указанный путь] \[HMI Folder] \[datalog] \
[Папка объекта *Data-Sampling*] \20090112.dtl
21. EasyPrinter интрепретирует Convert Batch File построчно, т.е. каждая строка формирует критерий.
22. Любые два аргумента должны быть разделены запятой.
23. Каждый аргумент должен быть в двойных кавычках.
24. Внутри аргумента не должно быть никаких запятых.
25. Дополнительная информация в Главе 25. Easy Converter.

26.4.2 Специальные критерии

Иногда необходимо использовать для обработки выгружаемые файлы из конкретной панели. Например:

Специальный критерий для панели с адресом IP = 192.168.1.26

3: "dtl", "EasyConverter /c $(PathName)", "192.168.1.26"

Специальный критерий для панели с именем = Weintek_01

4: "dtl", "EasyConverter /c $(PathName)", "Weintek_01"

Или необходимо обработать определенные архивные файлы *Data-Sampling* .

Специальный критерий для папки объектов с именем = Voltage

5: "dtl", "EasyConverter /s Voltage.lgs $(PathName)", "*", "Voltage"

Пятый критерий может быть представлен только для файлов выгружаемых из папки объектов **[Data Sampling]** с именем “Voltage”. Третий аргумент (“*”) показывает, что этот критерий принимает квалифицированные файлы *Data-Sampling* из любой панели. Пользовательможет изменить третий аргумент на “192.168.1.26”, “192.168.1.*”, имя панели и т.п., etc. для указания целевой панели.

26.4.3 Формат Convert Batch File

В таблице объяснены все аргументы в критерии.

№	Аргумент	Описание
1	File Type	Этот аргумент указывает имя расширения загружаемого файла – цели этого критерия (например, “dtl” для архивных файлов *Data-Sampling* , “evt” для архивных файлов *Event-Log*)
2	Command Line	Точная команда EasyPrinter посылаемая в окно консоли, если загружаемый файл квалифицирован.
3	a. HMI IP address b. HMI name	Этот аргумент указывает панель (HMI) - цель критерия.
4	Condition 1	● Если тип файла - “dtl” Этот аргумент указывает имя папки объектов **[DataSampling].** Это цели критерия. ● Другие Не используется .
5	Condition 2	Не используется (резерв на будущее)

26.4.4 Порядок анализа критериев

EasyPrinter анализирует критерии по возрастанию, каждый раз при загрузке файла. После того как файл был квалифицирован по критерию, анализ прекращается и начинается обработка следующего файла.

Следовательно, пользователь должен разместить критерий более значимый в верху Convert Batch File, а менее значимые ниже.

Возьмем 5 критериев (из п.26.4.3), для примера, правильный порядок будет такой:

Правильный порядок для предыдущего критерия

"dtl", "EasyConverter /s Voltage.lgs $(PathName)", "*", “Voltage"

"dtl", "EasyConverter /c $(PathName)", "EasyView"

"dtl", "EasyConverter /c $(PathName)", "192.168.1.26"

"dtl", "EasyConverter /c $(PathName)"

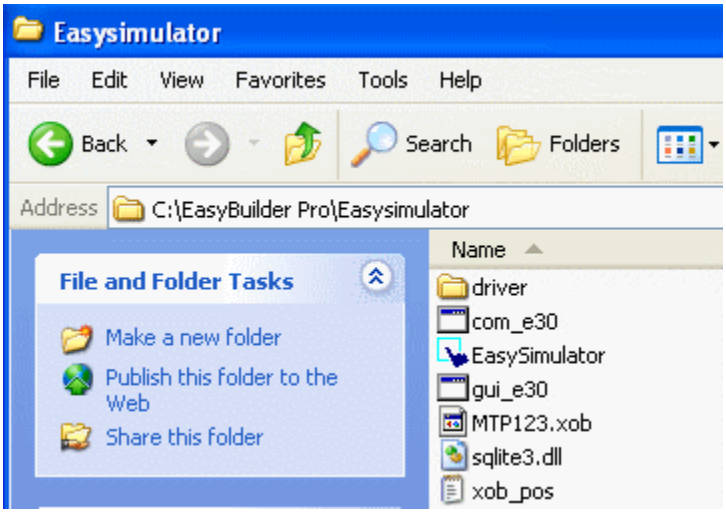
"evt", "EasyConverter /c $(PathName)"

Глава 27. Имитатор работы

Имитатор работы - *EasySimulator* позволяет пользователям выполнять имитацию в режиме «ON Line/OFF Line» без необходимости установки программного обеспечения EayBuilder Pro. Для этого пользователю нужно собрать некоторые файлы в одну папку.

27.1 Подготовка файлов

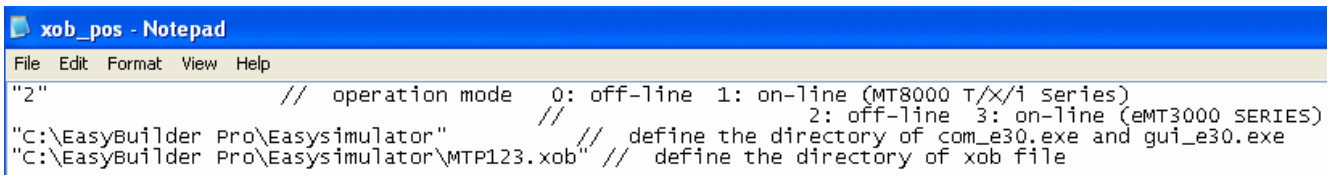
1. [driver] → [win32]
2. com_e30.exe
3. EasySimulator.exe
4. gui_e30.exe
5. sqlite3.dll
6. xob_pos.def



Пользователь может установить все перечисленные выше файлы в папку установки EasyBuilder Pro, это предполагает установку приложения EasyBuilder Pro на компьютер и, затем, копирование файлов на нужный компьютер.

27.2 Изменение содержимого «xob_pos.def»

Шаг 1. Откройте файл xob_pos.def, используя текстовый редактор (например, блокнот).



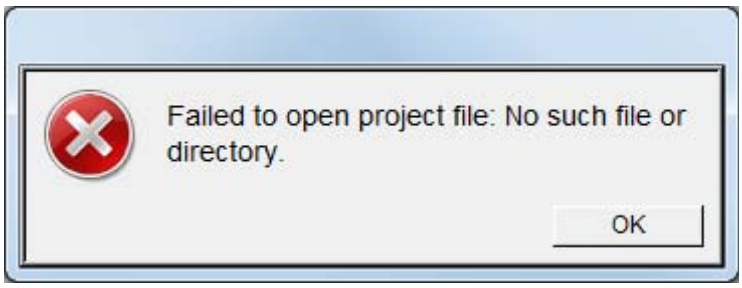
Номер строки	Описание
1	**[“2”]** Выполнить офф-лайн имитацию **[“3”]** Выполнить он-лайн имитацию
2	Укажите полный путь к местоположению файлов (com_e30.exe, gui_e30.exe, EasySimulator.exe, и т.д.)
3	Укажите полный путь к файлу проекта (*.xob)

Шаг 2.
Двойным щелчком запустите EasySimulator.exe, чтобы начать имитацию

Шаг 3.
Он-лайн/офф-лайн имитация отобразится на экране.

■ Если EasySimulator.exe не запускается, проверьте корректность указания необходимых директорий.

■ Окно показанное ниже означает, что есть ошибка в директории файла *.xob, скорректируйте ошибку.



Глава 28. Взаимодействие нескольких панелей

Взаимодействие нескольких панелей (Multi-HMI intercommunication) означает, что панель использует COM для связи с удаленной панелью, и может читать/писать данные из/в ПЛК соединенный с удаленной панелью, как показано ниже:

На рисунке показан ПЛК (PLC) соединенный с панелью «HMI 1», и панель «HMI 1» соединенная с панелью «HMI 2» через COM порт, поэтому HMI 2 может управлять PLC через HMI 1.

Далее приведен пример, как использовать EasyBuilder, чтобы создать проекты используемые в панелях HMI 1(Master) и HMI 2 (Slave).

28.1 Как создать проект ведущей панели (Master HMI)

Далее показано содержимое проекта панели «HMI 1» в **[System Parameter Settings] / [Device].**

Device list :

No.	Name	Location	Device type	Interface	I/F Protocol
Local HMI	Local HMI	Local	MT8xxx	Disable	N/A
Local PLC 1	FATEK FB Series	Local	FATEK FB Series	COM 1 (9600,E,7,1)	RS232
Local Server	Master-Slave Se...	Local	Master-Slave Se...	COM 3 (115200,E,8,1)	RS232

1. Так как панель HMI 1 соединена через COM 1 с PLC, список устройств должен включать **[Local PLC 1],** в нашем случае “FATEK FB Series”. Коммуникационные параметры должны быть настроены корректно.

2. Так как панель HMI 1 использует COM 3 для получения команд HMI 2; а должно быть добавлено новое устройство – **[Master-Slave Server]** для настройки коммуникационных свойств порта COM 3.

На рисунке вверху показаны параметры COM 3 - “115200, E, 8, 1”, и использование RS232. Эти параметры не обязаны совпадать с параметрами в PLC, но **[Data bits]** должны быть настроены на **8**. В общем, рекомендуется устанавливать более высокую скорость передачи COM 3 для более эффективного обмена данными с ПЛК.

28.2 Как создать проект в ведомой панели (Slave HMI)

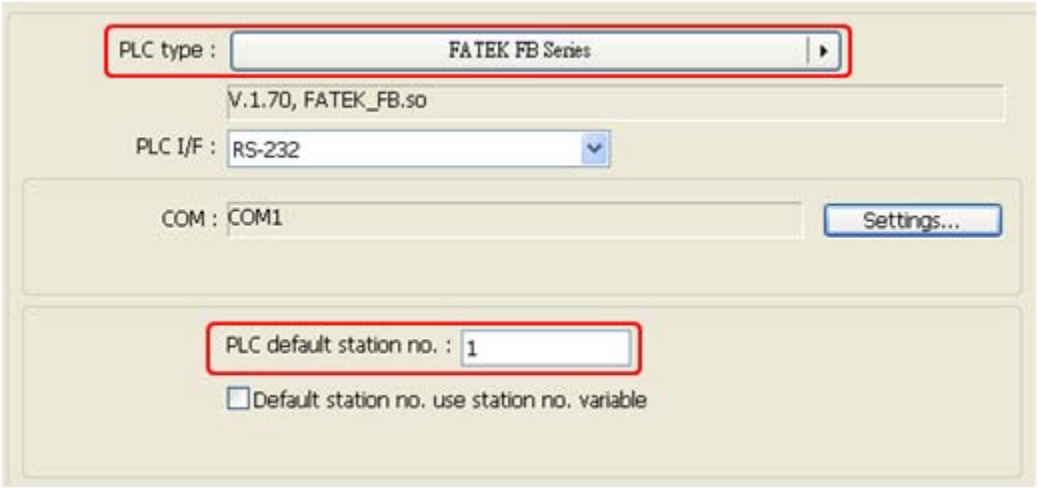
Содержимое проекта панели «HMI 2» в **[System Parameter Settings] / [Device].**

Device list :

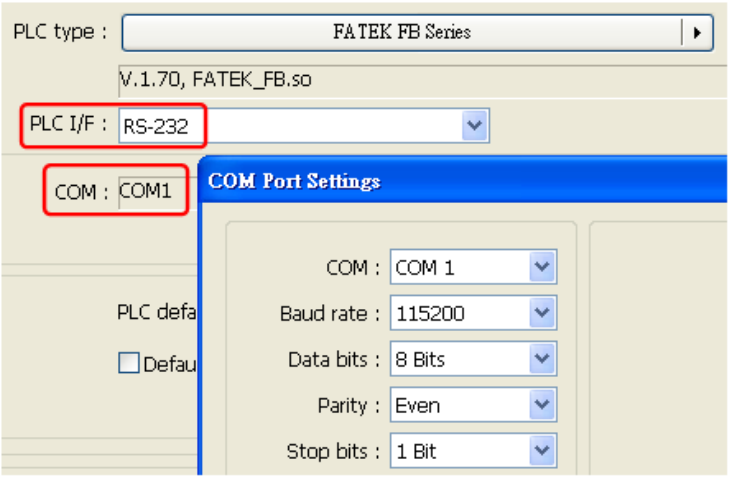
No.	Name	Location	Device type	Interface
Local HMI	Local HMI	Local	MT8xxx	Disable
*Remote PLC 1	FATEK FB Series	COM 1 (master-slave mode)	FATEK FB Series	COM 1 (115200,E

Панель HMI 2 читает из контроллера соединенного с панелью HMI 1, таким образом, что HMI 2 видит ПЛК, как удаленное устройство. Следовательно, необходимото добавить **[*Remote PLC 1]** в список устройств, в нашем случае -“FATEK FB Series”. Как создать [*Remote PLC 1] описано далее:

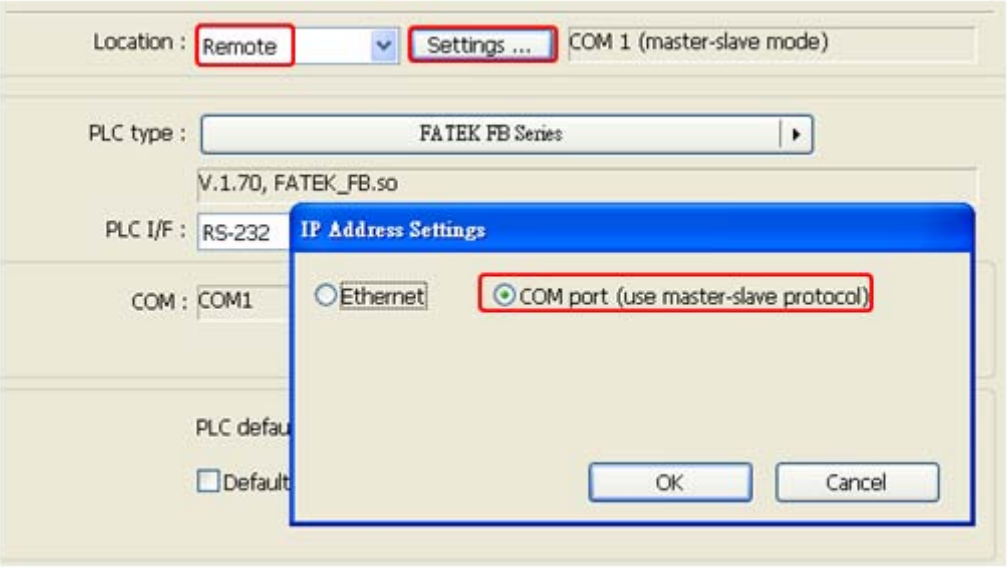
1. Создайте новое устройство ”FATEK FB Series”. **[PLC default station no.]** должен быть тем же самым, что и у присоединенного ПЛК.

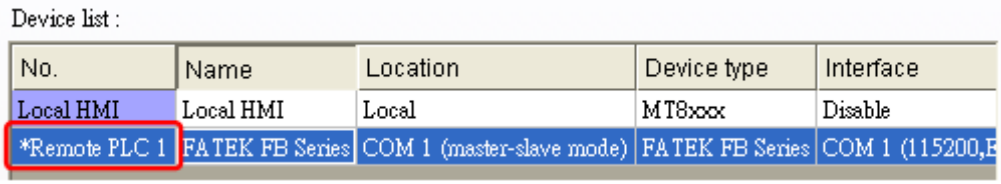


2. Корректно настройте параметры. COM 1 панели «HMI 2» соединен с COM 3 панели «HMI 1», поэтому их коммуникационные параметры и интерфейс должны быть одинаковыми, независимо от параметров ПЛК.
В примере,
Используем RS232,
Параметры - [115200, E, 8, 1]



3. Так как HMI 2 видит PLC, как удаленное устройство, здесь мы изменим **[Location]** на **[Remote]**, и выберем **[COM port]** для связи с удаленной панелью (HMI 1).





4. После завершения всех настроек, в списке **[Device List]** появится новое устройство с именем **[*Remote PLC 1].** Название устройства содержит символ “*” , который означает, что даже несмотря на наличие “Remote” в имени, в действительности устройство посылает команды и получает ответы через локальный COM порт, и следовательно соединение с PLC может быть просмотрено в локальных системных регистрах, таких как, [*Remote PLC 1], [*Remote PLC 2], [*Remote PLC 3] и [Local PLC 1], [Local PLC 2], [Local PLC 3] использующих одни и те же системные регистры из списка, приведенного на следующей странице:

Тэг	Описание
LB-9150	Когда ON, автосоединение с PLC (COM 1) при разрыве связи. Когда OFF, игнорирует разрыв соединения с PLC.
LB-9151	Когда ON, автосоединение с PLC (COM 2) при разрыве связи. Когда OFF, игнорирует разрыв соединения с PLC.
LB-9152	Когда ON, автосоединение cPLC (COM 3) при разрыве связи. Когда OFF, игнорирует разрыв соединения с PLC.
LB-9200~ LB-9455	Эти локальные регистры сигнализируют состояние соединения с PLC (через COM1). LB9200 показывает состояние соединения с PLC (station no. 0), LB9201 показывает состояние соединения с PLC (station no. 1) и так далее. Когда ON, показывает, что состояние соединения нормальное . Когда OFF, показывает разрыв соединения с PLC. При установке в ON снова, система будет пытаться соединиться с PLC.
LB-9500~ LB-9755	Эти локальные регистры сигнализируют состояние соединения с PLC (через COM2). LB9500 показывает состояние соединения с (station no. 0), LB9501 показывает состояние соединения с PLC (station no. 1) и так далее. Когда ON, показывает, что состояние соединения нормальное . Когда OFF, показывает разрыв соединения с PLC. При установке в ON снова, система будет пытаться соединиться с PLC.
LB-9800~ LB-10055	Эти локальные регистры сигнализируют состояние соединения с PLC (через COM3). LB9800 показывает состояние соединения с PLC (station no. 0), LB9801 показывает состояние соединения с PLC (station no. 1) и так далее. Когда ON, показывает, что состояние соединения нормальное . Когда OFF, показывает разрыв соединения с PLC. При установке в ON снова, система будет пытаться соединиться с PLC.

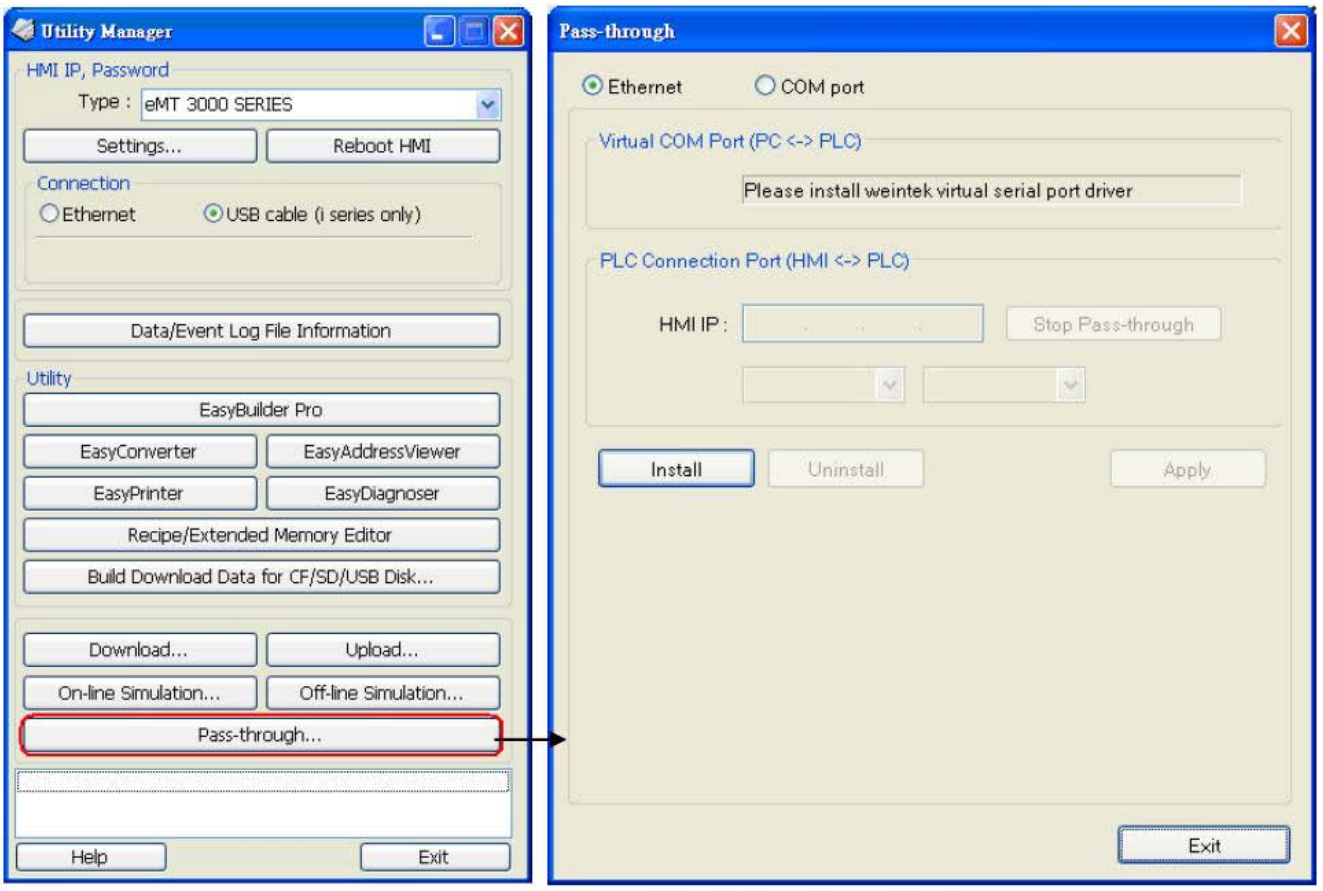
Глава 29. Функция сквозной передачи

Функция сквозной передачи (pass-through function) позволяет приложению на компьютере управлять ПЛК через панель оператора. В этом случае панель работает, как конвертер.

Функция сквозной передачи может быть реализована в двух режимах:

[Ethernet] и **[COM port]**.

Щелчок по **[Pass-through]** в **[Utility Manager]** откроет окно настроек.

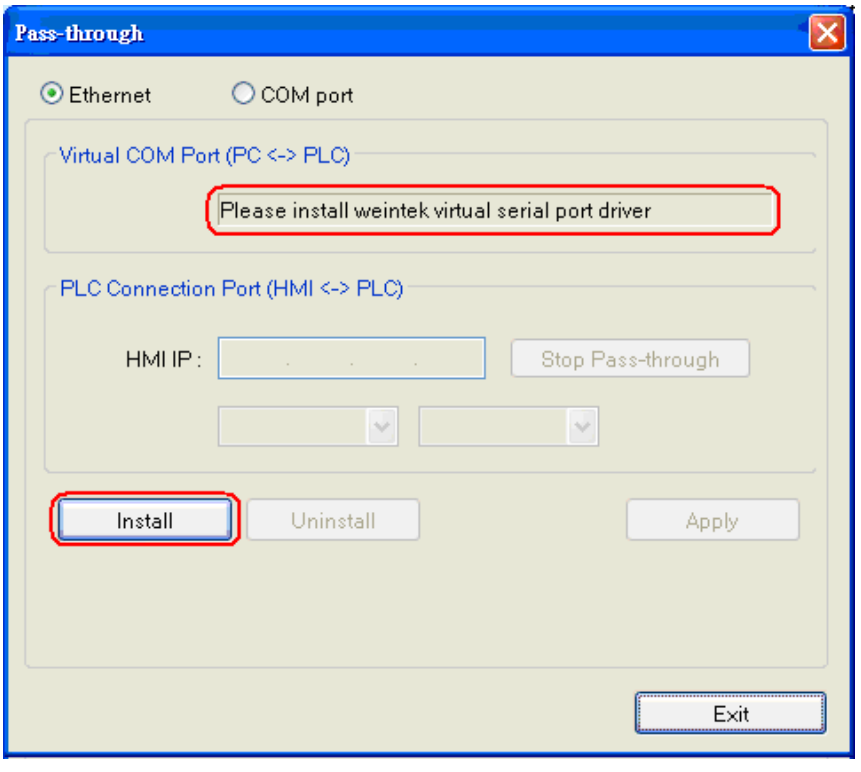


29.1 Ethernet Mode

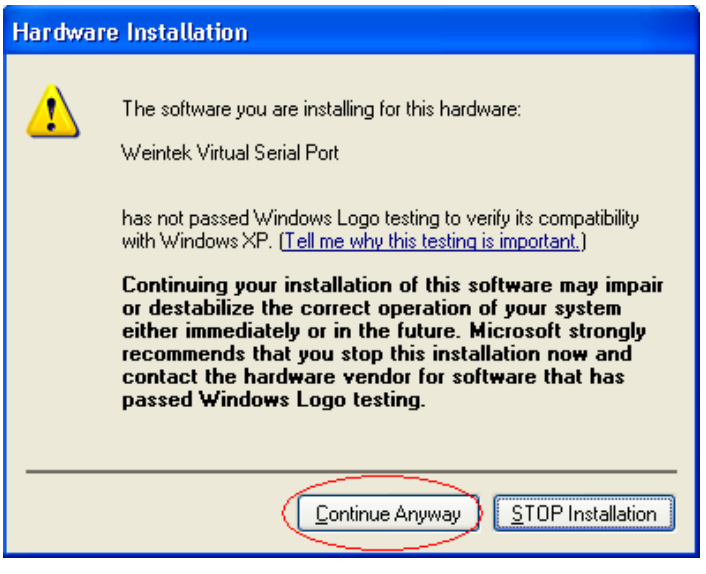
[How to install virtual serial port driver] – Как установить драйвер виртуального последовательного порта.

Перед использованием режима [Ethernet], убедитесь в том, что драйвер виртуального последовательного порта (Weintek virtual serial port driver) установлен, как показано ниже:

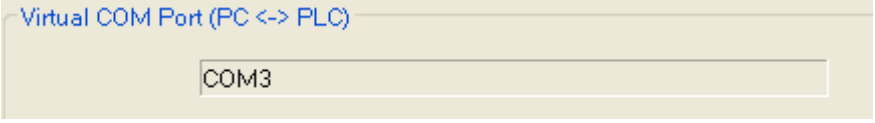
Если **[Virtual COM port (PC<->PLC)]** показывает **[Please install weintek virtual serial port driver]**, щелкните по **[Install].**



Если это окно всплывет во время установки, нажмите **[Continue Anyway] –** Все равно продолжить.

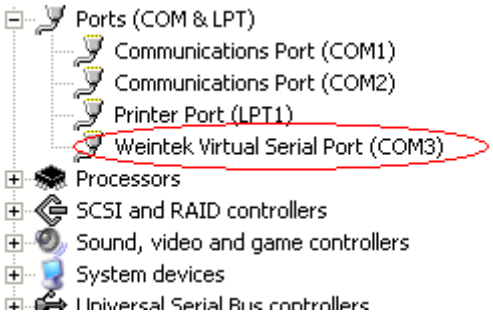


После завершения процесса, виртуальный COM порт будет отображен, как показано ниже.

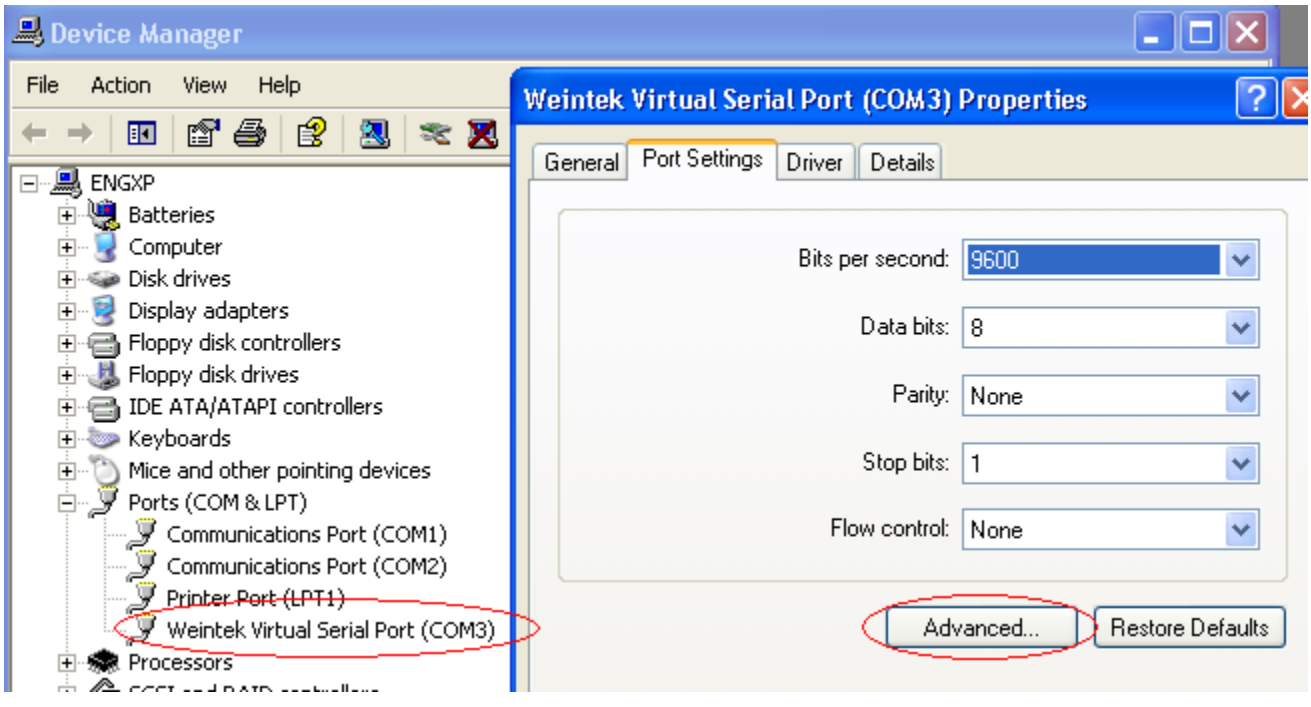


29.1.1 Как изменить Virtual Serial Port

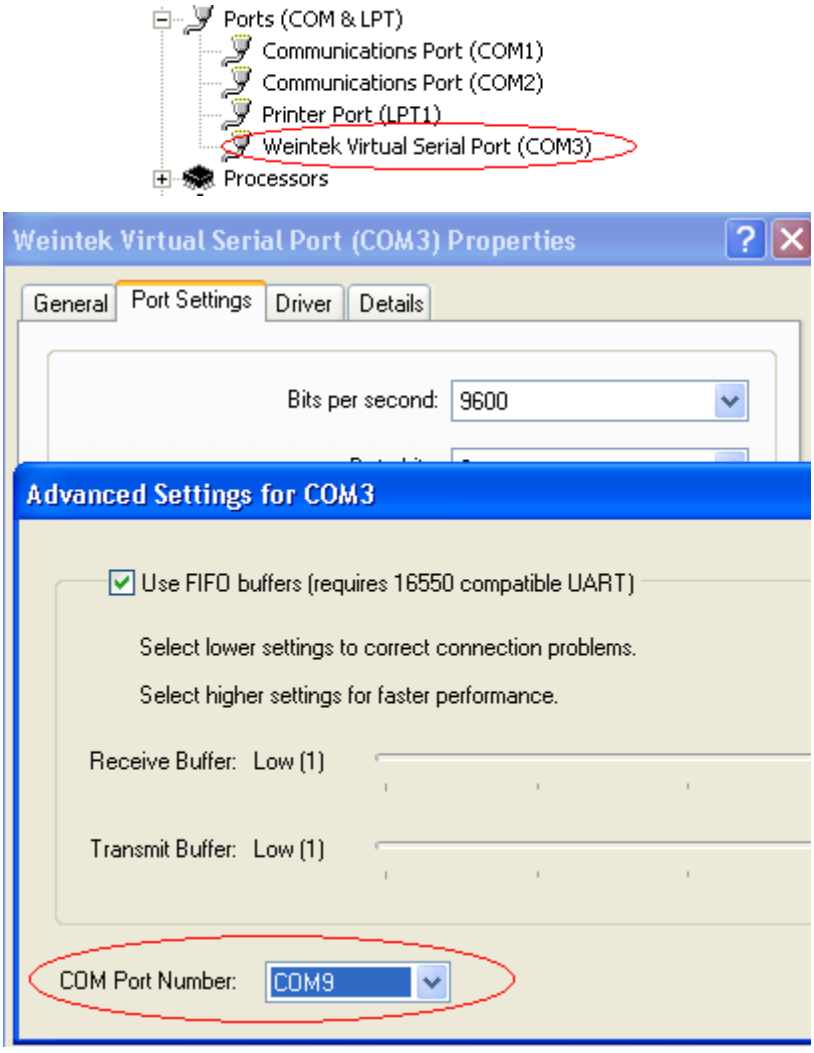
Откройте **[System Properties] -> [Device Manager],** чтобы проверить успешность установки виртуального последовательного порта.



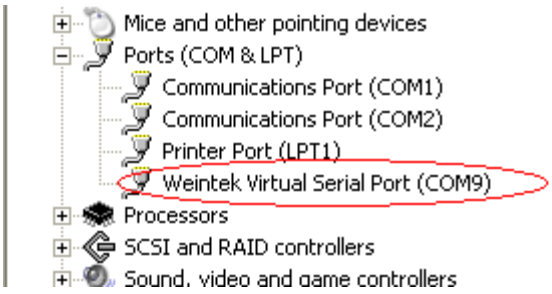
Если Вы хотите изменить номер виртуального последовательного порта, щелкните по **[Weintek Virtual Serial Port],** чтобы открыть **[Port Settings] / [Advanced…]** :



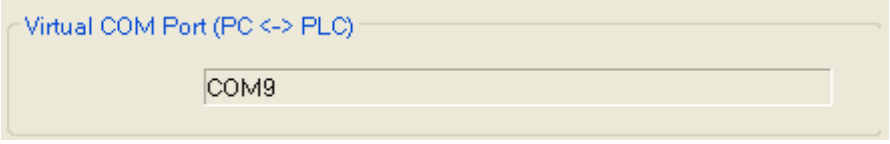
Например , изменить номер виртуального последовательного порта с COM 3 на COM 9.



Выберите COM 9 и нажмите **[OK]**, номер виртуального последовательного порта стал COM 9.



Это можно увидеть и в **[Utility Manager]**.



29.1.1 Как использовать режим «Ethernet Mode»

После установки драйвера виртуального последовательного порта, необходимо сделать четыре шага, чтобы использовать режим «Ethernet mode» сквозной передачи.

Шаг 1

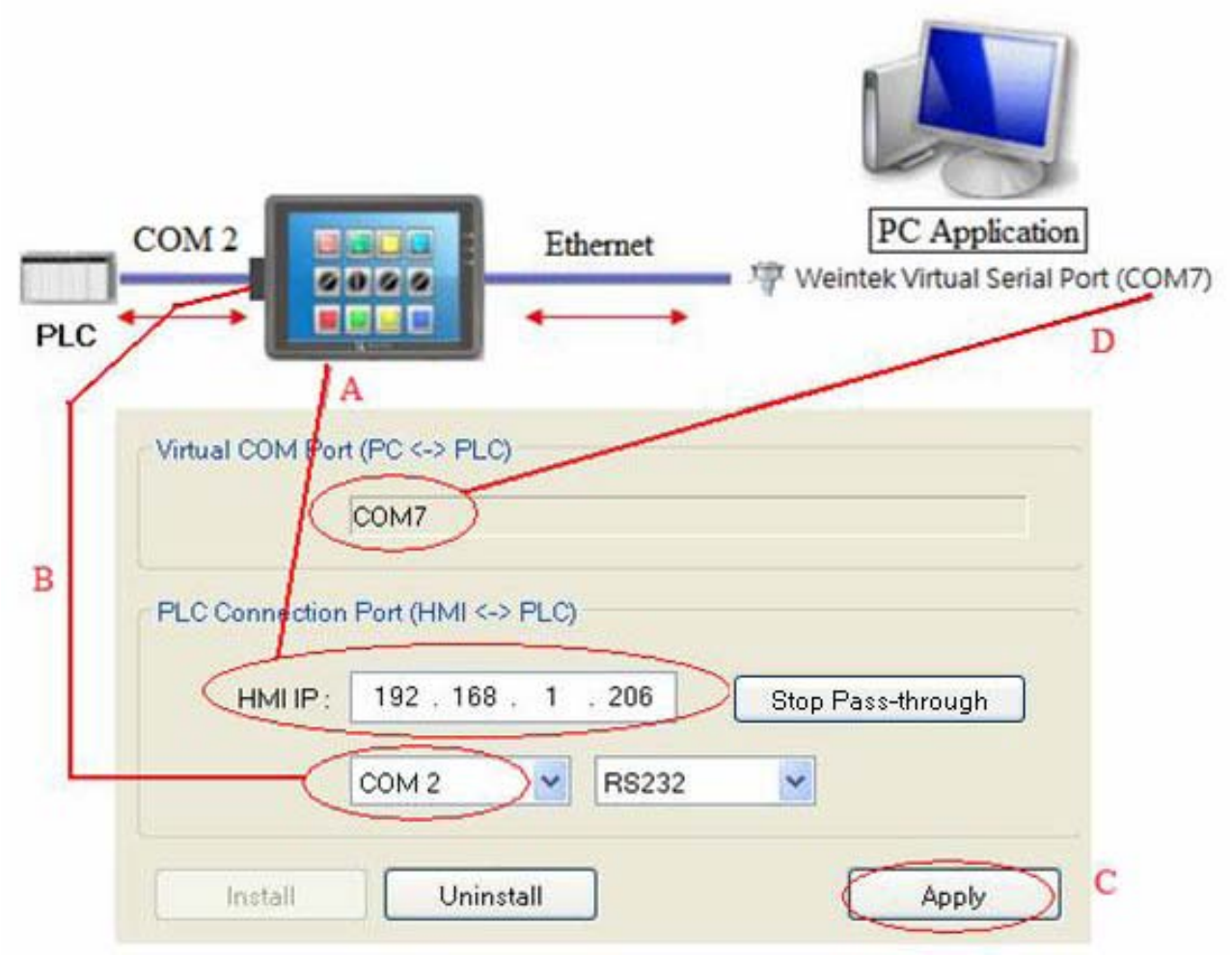
Установите IP-адрес панели соединенной с ПЛК. Например, IP 192.168.1.206

Шаг 2

Назначьте последовательный порт соединяющий панель с ПЛК. Например, COM2 (RS232).

Шаг 3

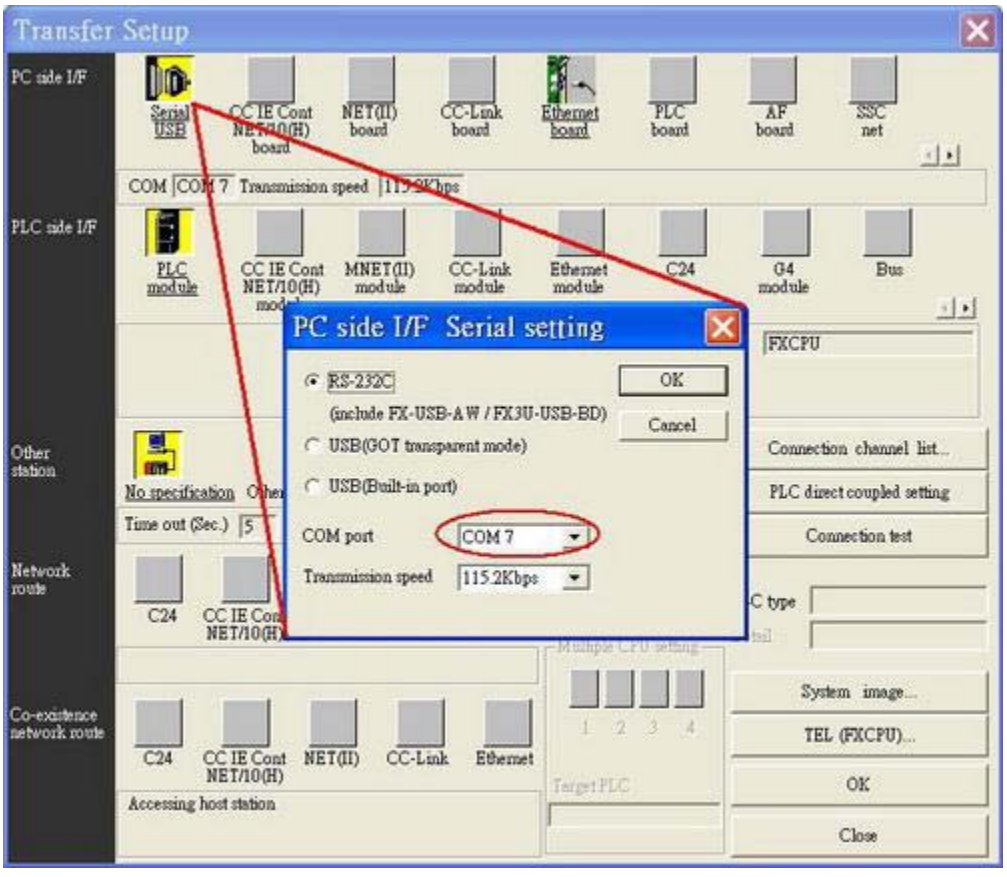
Нажмите **[Apply] - Применить**, и эти настройки будут обновлены.



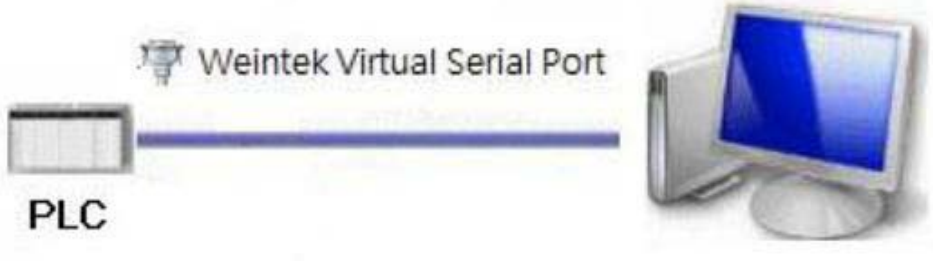
Шаг 4

В приложении на компьютере, номер последовательного порта должен быть тем же самым, что и виртуального порта.

Например, используя приложение Mitsubishi (если виртуальный последовательный порта - COM 7), откройте **[PC side I/F Serial setting] / [COM port] ,** чтобы выбрать COM 7, как показано:

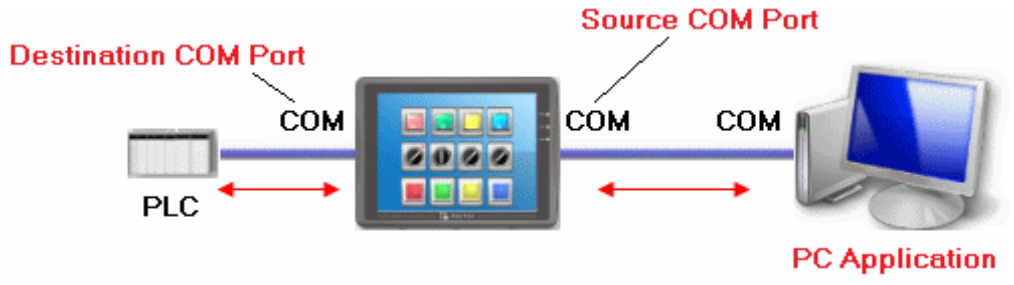


После выполнения всех настроек, при выполнении приложения панель будет автоматически переключена в режим сквозной передачи (соединение между панелью и ПЛК будет приостановлено в этот момент и будет продолжено при закрытии приложения), как показано ниже:



С этого момента приложение управляет ПЛК напрямую через виртуальный последовательный порт.

29.2 COM Port Mode



Source COM Port – Последовательный порт - Источник

Порт использован для соединения панели с компьютером.

Destination COM Port - Последовательный порт назначения

Порт использован для соединения панели с ПЛК.

При использовании режима « **[COM port]** mode» сквозной передачи, необходимо корректно настроить свойства последовательных портов: порта источника и порта назначения.

29.2.1 Настройка режима «COM Port Mode»

Существует два способа настройки функции сквозной передачи в режиме **[COM port]** .

(1) Используя *Utility Manager*

(2) Используя системные регистры LW-9901 и LW-9902

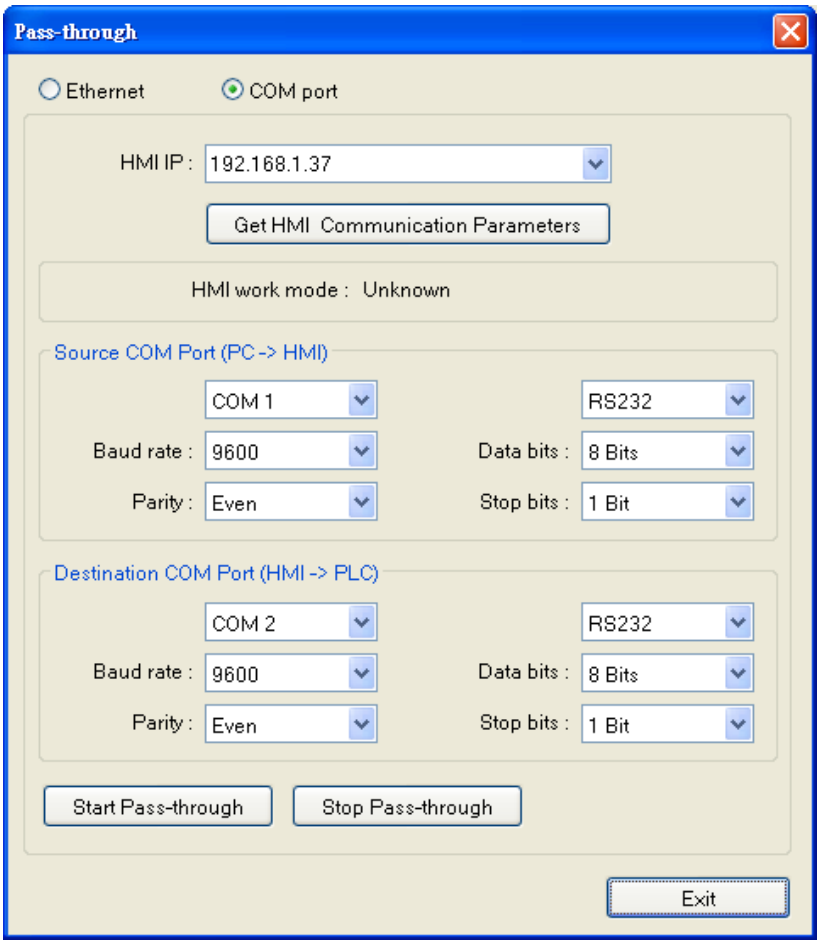
LW-9901: COM порт источник сквозной передачи (1~3: COM1~COM3)

LW-9902: COM порт назначения сквозной передачи (1~3: COM1~COM3)

ПРИМЕЧАНИЕ: После окончания использования функции *Pass Through*, пользователь должен щелкнуть по **[Stop Pass-through],** чтобы дезактивировать эту функцию и панель смогла бы начать обмен данными с ПЛК

Начало настройки функции сквозной передачи при помощи *Utility Manager*.

Нажмите кнопку **[Pass-through]** в Utility Manager, чтобы установить коммуникационные параметры.



[HMI IP]

Назначьте IP-адрес панели.

Нажмите **[Get HMI Communication Parameters],** для получения настроек портов источника и назначения. Параметры придут из зарезервированных адресов:

COM порт источник (Source) и COM порт назначения (Destination)

LW-9901 (Source COM port)	1 : COM 1	3 : COM 3
LW-9902 (Destination COM port)	1 : COM 1	3 : COM 3

Настройки COM 1

LW-9550 (PLC I/F)	0 : RS232	1 : RS485/2W		2 : RS485/4W
LW-9551 (baud rate)	0 : 4800 4 : 57600	1 : 9600 5 : 115200	2 : 19200	3 : 38400
LW-9552 (data bits)	7 : 7 bits	8 : 8 bits		
LW-9553 (parity)	0 : none	1 : even 2 : odd		
LW-9554 (stop bits)	1 : 1 bit	2 : 2 bits		

Настройки COM 3

LW-9560 (PLC I/F)	0 : RS232 1 : RS485/2W
LW-9561 (baud rate)	0 : 4800 1 : 9600 2 : 19200 3 : 38400 4 : 57600 5 : 115200
LW-9562 (data bits)	7 : 7 bits 8 : 8 bits
LW-9563 (parity)	0 : none 1 : even 2 : odd
LW-9564 (stop bits)	1 : 1 bit 2 : 2 bits

Нажмите **[Get HMI Communication Parameters],** чтобы обновить текущие значения коммуникационных параметров панели.

29.2.2 Режимы работы панели

Существует три режима работы функции «pass-through»,

Режим	Описание
Unknown – Неизвестный	До получения настроек из панели, «HMI work mode» показывает режим “Unknown”.
Normal - Нормальный	После получения настроек из панели, если «HMI work mode» показывает режим “Normal”, то компьютер не может управлять ПЛК через панель.
Pass-through – Сквозная передача	Панель работает в режиме сквозной передачи; в это время приложение на компьютере может управлять ПЛК через порт-источник.

[Source COM Port]、[Destination COM Port]

Коммуникационные араметры портов источнка и назначения отображаются в двух местах. Настройки будут использованы, когда выбран **[Start pass-through]**.

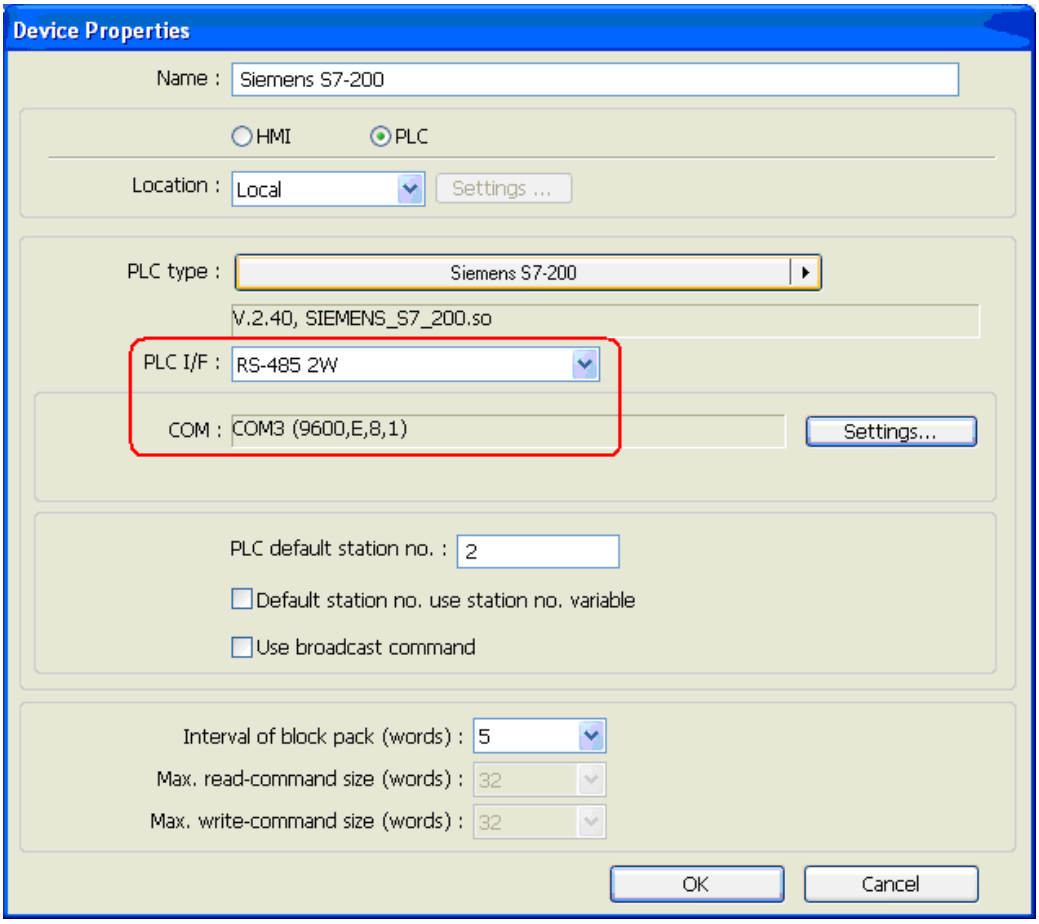
The “Baud rate”, “Data bits”, “Parity”, “Stop bits” портов [Source COM Port] и [Destination COM Port] должныбыть одинаковы.

[Source COM Port] соединен с компьютером, поэтому выбираем RS232;

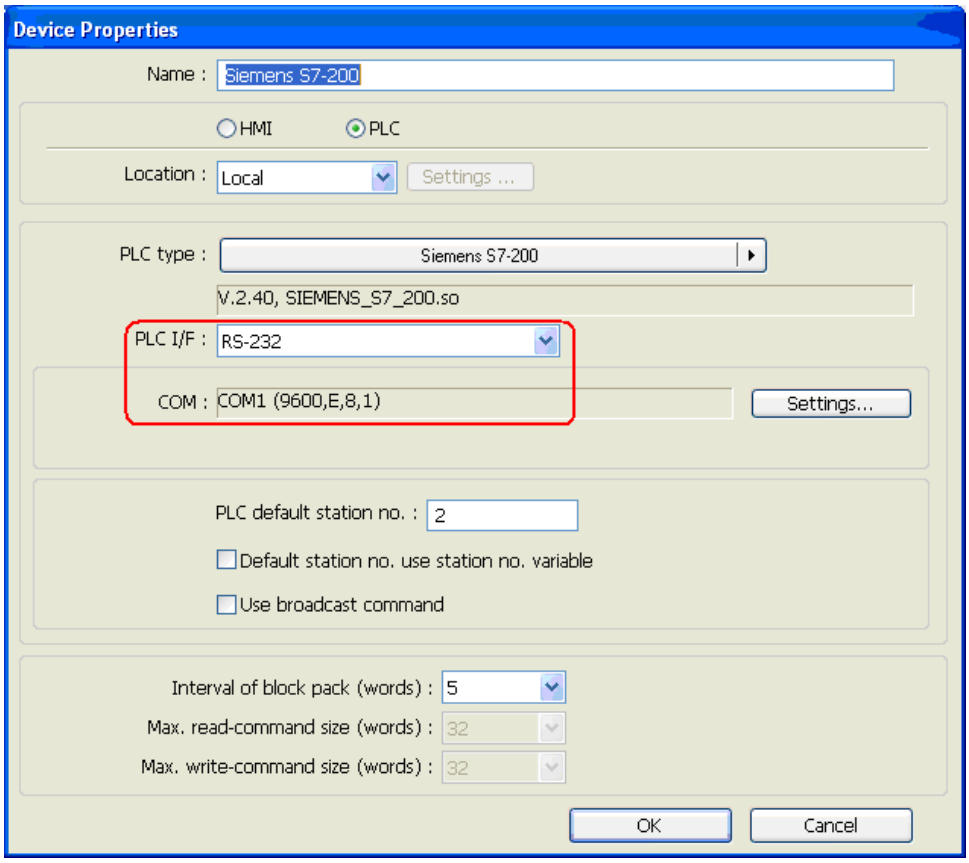
[Destination COM Port] соединен с ПЛК, поэтому настройки его зависят от настроек порта ПЛК.

На рисунке на следующей странице показаны настройки панели соединенной с ПЛК SIEMENS S7/200.

Порт панели COM 1 (RS232) соединен с компьютером, COM 3 (RS485 2W) соединен с ПЛК. Параметры связи ПЛК -”9600, E, 8, 1”. До начала работы функции «pass-through», пользователь должен настроить параметры проекта MTP и загрузить проект в панель.

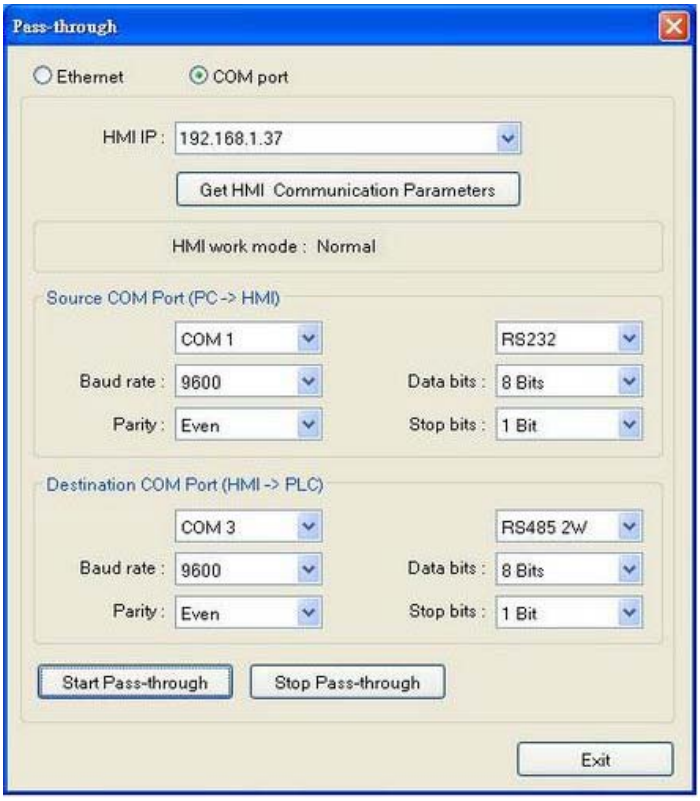


После загрузки проекта в панель, откройте проект и измените «PLC I/F» и COM порт в COM 1 RS232 (компьютер использует COM 1 для связи с панелью), как показано далее:



После этого, нажмите **[Pass-through],** чтобы присвоить панели IP -адрес; например, 192.168.1.37.

Наконец, нажмите **[Get HMI Communication Parameters]**, как показано:



Нажмите **[Start Pass-through]** режим работы панели «HMI work mode» переключится в “Pass-through”. Пользователь может проводить имитацию работы - *on-line simulation*.
Теперь приложение на компьютере может управлять ПЛК через панель, а панель работает в это время, как конвертер.

ПРИМЕЧАНИЕ: Обмен между панелью и ПЛК будет находится в паузе во время работы функции *Pass Through.*
Если пользователь хочет прекратить обмен между панелью и компьютером, необходимо щелкнуть по **[Stop Pass-through],** чтобы дезактивировать эту функцию и панель смогла бы начать обмен данными с ПЛК

29.3 Использование системных адресов для активации функции Pass-Through

Другой способ активировать функцию «pass-through» - это использование LW-9901/LW-9902, для настройки порта – источника и порта - назначения напрямую.
Когда значения LW-9901 и LW-9902 соответствуют условиям, приведенным ниже, панель начнет работать в режиме «pass-through» автоматически:
a. Значения LW-9901 и LW-9902 должны быть 1 или 3
(1: COM 1, 3: COM 3).
b. Значения LW-9901 и LW-9902 не должны быть одинаковыми.

Если необходимо изменить коммуникационные параметры, просто измените значение в соответствующем системном адресе и установте в ON : LB-9030, LB-9031 и LB-9032.
Панель будет принуждена принять новые настройки.

Тэг	Описание
LB-9030	Обновить настройки COM1 (при установке в ON)
LB-9032	Обновить настройки COM3 (при установке в ON)

ПРИМЕЧАНИЕ: Если пользователь хочет остановить «pass-through», то необходимо изменить значения в LW-9901 и LW-9902 на значения которые не являются: 1, 2, 3 (EX: 0).

Глава 30. Защита проекта

Права собственности на разработанную программу должны быть защищены.
EasyBuilder Pro поддерживает функции защиты файлов проекта, чтобы защитить результаты работы разработчика.

Эти функции защиты не могут быть расшифрованы изготовителем, так как они закодированы пользователем, поэтому, ЗАПОМИНАЙТЕ ВАШ ПАРОЛЬ

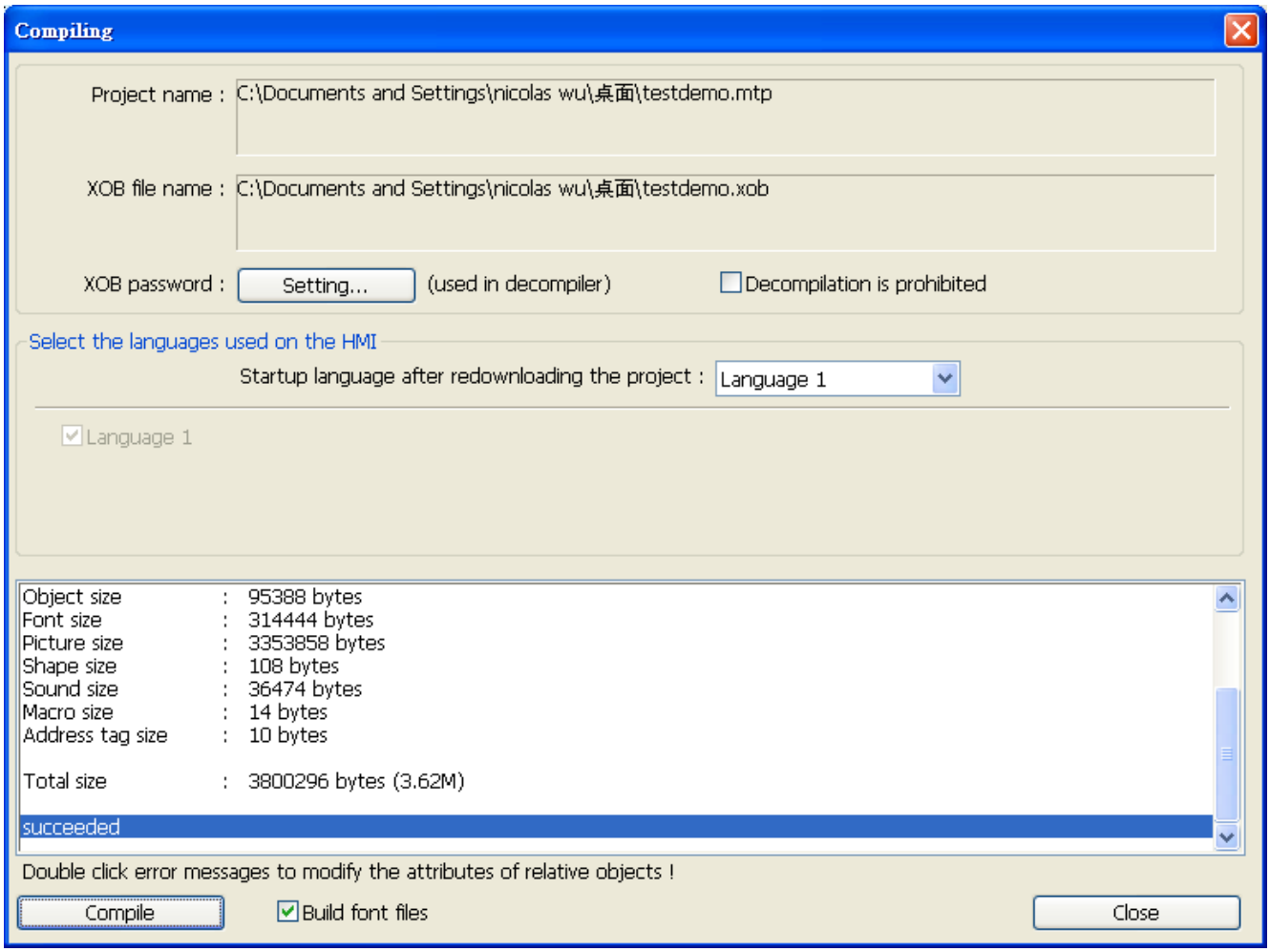
30.1 Пароль для файла XOB

После компиляции файла проекта (MTP), пользователь может скомпилировать файл в формат XOB, который может быть загружен в панель.

Пароль (Password) может быть установлен для защиты файла XOB в окне **[Compiling]**.

Паороль будет затребован, при попытке де-компилировать файл XOB в файл MTP.

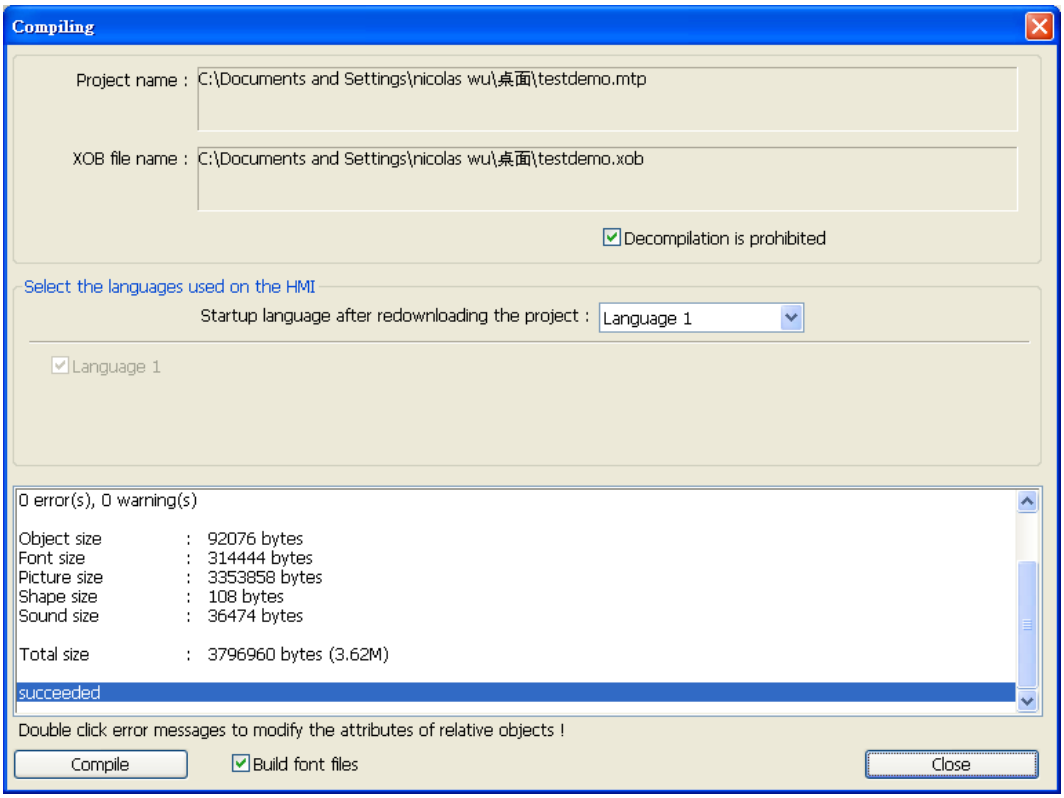
(диапазон XOB пароля: 0 ~ 4294967295)



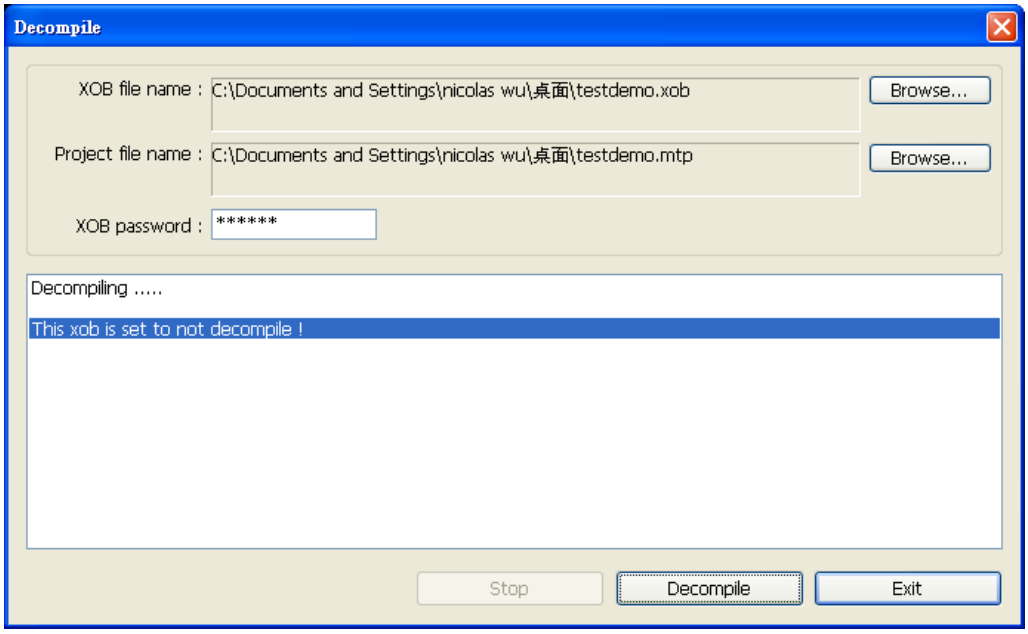
Если неправильный пароль был введентри раза при декомпиляции, необходипо перезапустить декомпилятор.

30.2 Декомпиляция запрещена

Если сделана отметка в окошке **[Decompilation is prohibited]**, система будет автоматически отвергать **[XOB password]**. Следовательно, файл XOB не может быть декомпилирован в файл MTP.

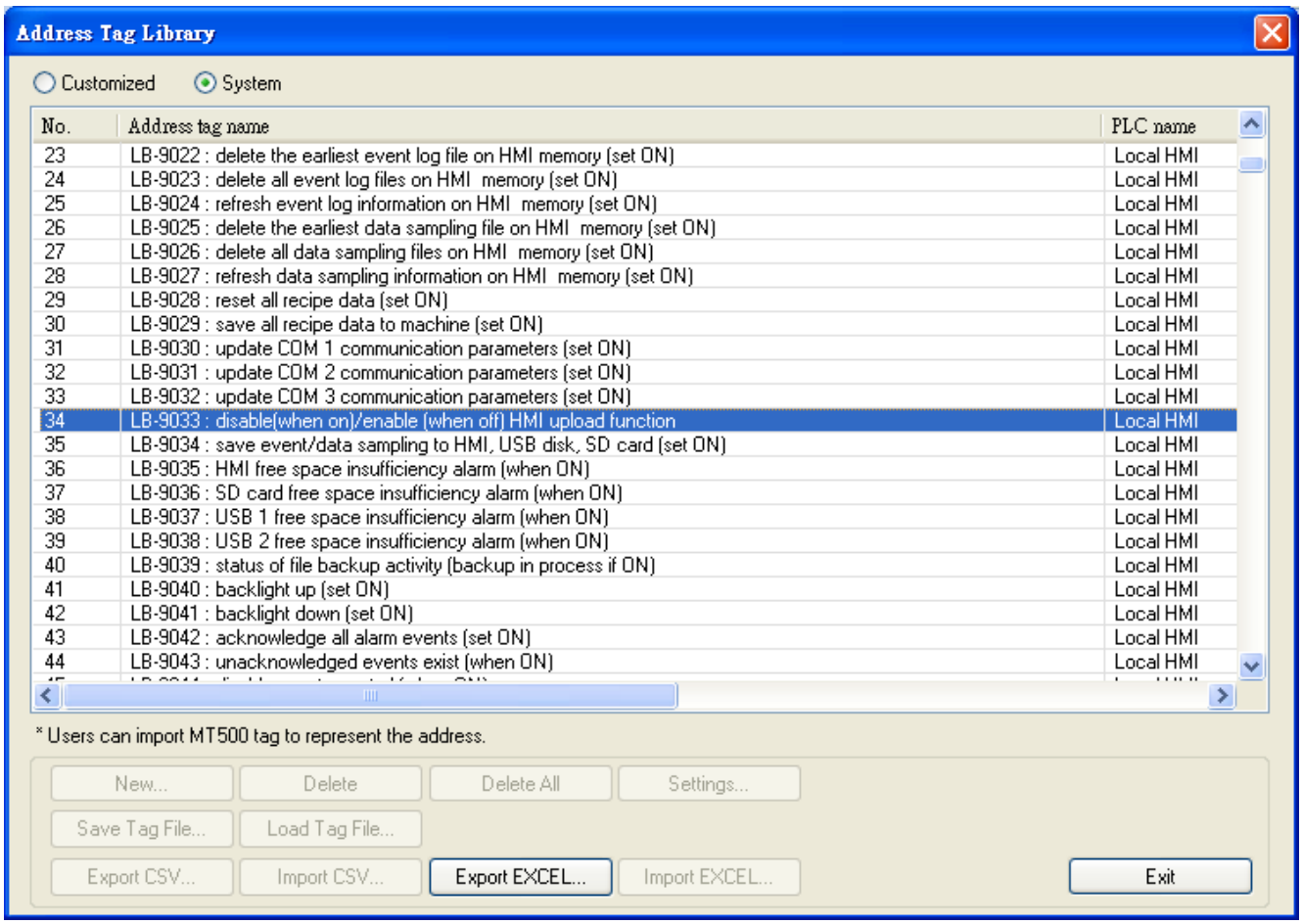


При попытке декомпилировать файл XOB с настройкой **[Decompilation is prohibited]**, появится сообщение об ошибке **“This xob is set to not decompile!!”,** как показано ниже.



30.3 Блокировка функции выгрузки из панели [LB-9033]

В EasyBuilder Pro есть системный регистр [LB-9033]. Когда значение бита этого адреса переходит в «ON», панель отключает функцию выгрузки файла XOB. Чтобы активировать бит [LB-9033], необходимо перезагрузить панель.

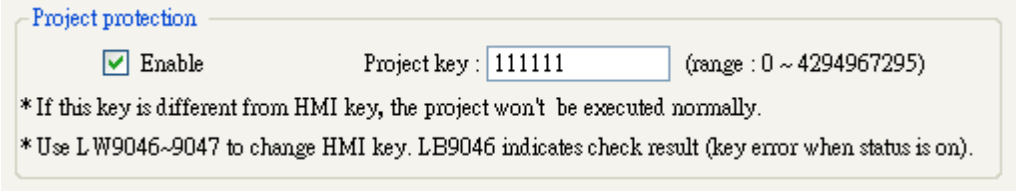


При попытке выгрузить файл XOB (с активированной этой функцией), запрашиваемый файл XOB после выгрузки будет иметь размер 0 байт, и не может быть декомпилирован.

30.4 Ключ проекта (Project Key)

Проект пользователя может быть ограничен исполнением только на указанной панели (применимо толькодля панелей серии «I»).

Перейдите **[System Parameters Settings] / [General] / [Project protection]**.



LW-9046 ~ LW-9047 (32-bit) могут быть использованы в качестве ключа панели - **[HMI key]**. Значения этих двух регистров не может быть прочитано или записано с удаленной панели. Используя эту функцию, установите пароль или ключ проекта (**[Project key]** в диапазоне: 0 ~ 4294967295), и файл XOB будет ограничен исполнением только в панели, у которой [HMI key] и [Project key] идентичны. Если они различаются, система переведет бит LB-9046 в состояние «ON». Панель необходимо перезагружать, каждый раз при коррекции ключа панели - [HMI key].

Important

■ Когда ключи [HMI key] и [Project key] различаются, панель и ПЛК не в состоянии соединяться между собой.

Download the Demo

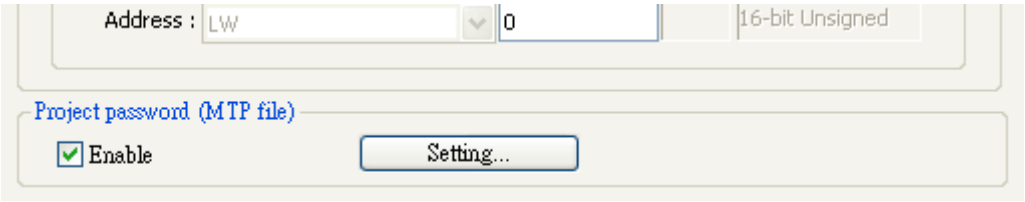
Вы можете загрузить соответствующий демо-проект:

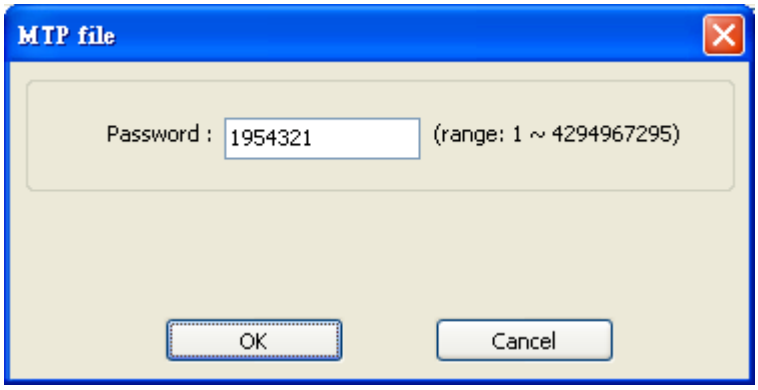
ftp://ftp.weintek.com/MT8000/Project/System_Sample/Project_Key.zip

30.5 Пароль проекта MTP (Project Password))

Пароль может быть установлен для защиты файла MTP в закладке **[System parameter] / [Security]** .

После активирования этой функции, при попытке редактирования файла MTP будет запрашиваться пароль (диапазон пароля MTP : 1 ~ 4294967295).





После настройки, при открытии проекта, всплывет окно с запросом на ввод пароля, как показано на рисунке.

■ При использовании функции “Window Copy”, если файл-источник защищен паролем MTP, необходимо ввести пароль, чтобы EasyBuilder Pro выполнил копирование окна.

Глава 31. Протокол Memory Map

Коммуникационный протокол MemoryMap подобен протоколу IBM 3764R, он используется когда изменения данных в памяти небольшие. (Большие изменения могут привести к перегрузке протокола MemoryMap.) Протокол MemoryMap используют для связи между двумя устройствами. При настройке MemoryMap, одно должно быть назначено Ведущим - Master, адругое Ведомым - Slave. В нормальном состоянии, Master и Slave не обмениваются данными между собой, исключая случаи с изменением данных в памяти одного из них.

Как только данные становятся одинаковыми, обмен прекращается.

Поэтому этот протокол используют для поддержания согласованность указанных зон памяти двух устройств (Master and Slave).

Соответствующие зоны памяти должны иметь одинаковые свойства - регистры MW(MB) в Master и Slave. (1000 слов MW(MB) зарезервированы для коммуникации панели по протоколу MemoryMap).

Свойства памяти: MB соотносятся с MW, в соответствии со следующим списком, MB0~MBf и MW0, MB10~MB1f и MW1…, все они указывают одни и те же регистры.

Имя устройства (Device name)	**Формат**	**Диапазон**
MB	dddd(h)	dddd:0~4095 h:0~f(hex)
MW	dddd	dddd:0~9999

При использовании коммуникационного протокола MemoryMap , «master» и «slave» должны использовать одни и те же коммуникационные настройки.

Схемы соединений следующие:

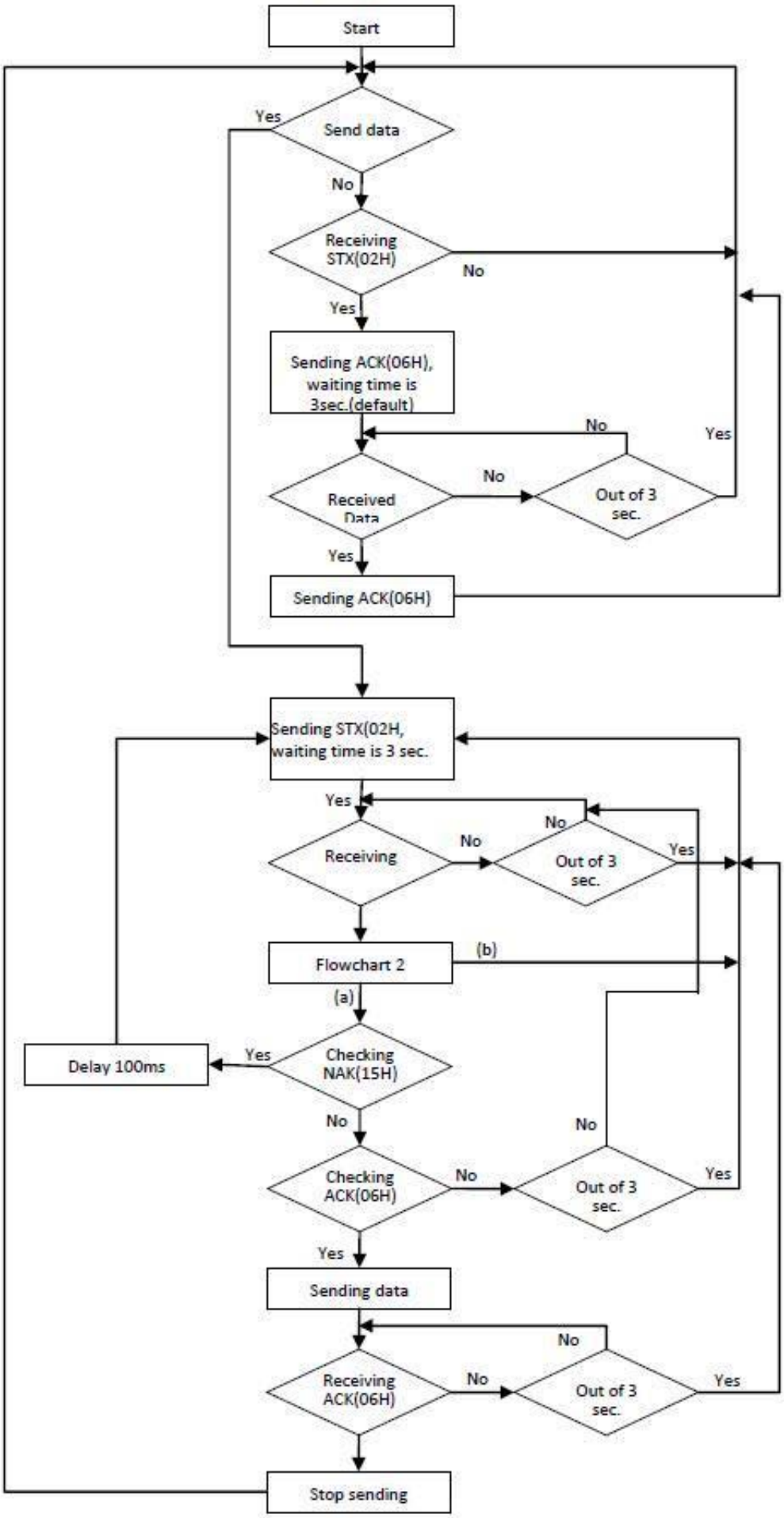
RS232	
Master	Slave
TX(#)	RX(#)
RX(#)	TX(#)
GND(#)	GND(#)

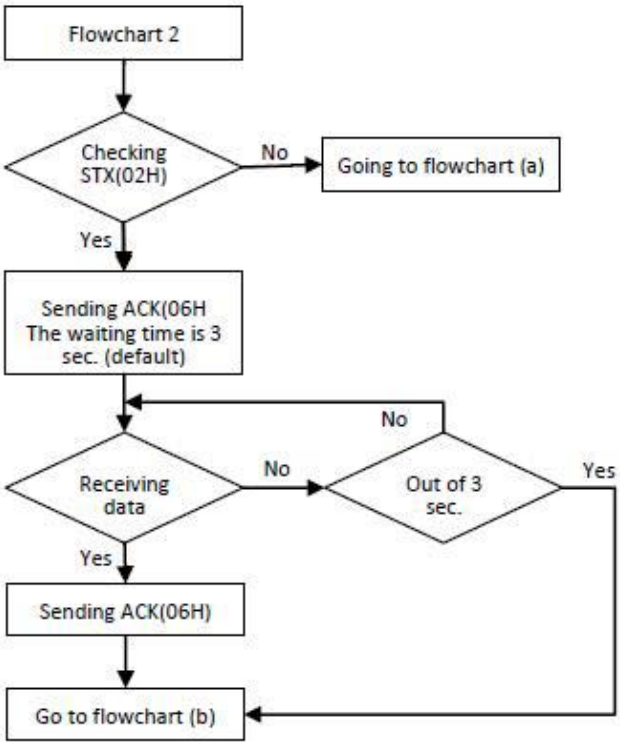
RS485 (4W)	
Master	Slave
TX+(#)	RX+(#)
TX-(#)	RX-(#)
RX+(#)	TX+(#)
RX-(#)	TX-(#)
GND(#)	GND(#)

ПРИМЕЧАНИЕ: # - означает, что конкретное число определяются устрой ствами.

Блок-схема (Flowchart 1)процесса обмена приведена на следующей странице:

Flowchart 1


Start
Send data
Yes
No
Receiving STX(02H)
No
Yes
Sending ACK(06H), waiting time is 3sec.(default)
No
Yes
Received Data
No
Out of 3 sec.
Yes
Sending ACK(06H)
Sending STX(02H, waiting time is 3 sec.
Yes
Receiving
No
No
Out of 3 sec.
Yes
Flowchart 2
(b)
(a)
Delay 100ms
Yes
Checking NAK(15H)
No
No
Checking ACK(06H)
No
Out of 3 sec.
Yes
Yes
Sending data
No
Receiving ACK(06H)
No
Out of 3 sec.
Yes
Stop sending



ПРИМЕЧАНИЕ: Блок-схема 2 (*Flowchart 2*) применима для «slave», но не для «master»,

STX – сигнал запроса на обмен,

ACK – сигнал обратной связи (feedback),

NAK - сигнал занятости.

Существует два формата данных, один для MB и другой для MW:

Для команд MB		
Смещение (байт)	**Формат**	**Описание**
0	0x02	Сигнал работы к MB
1 2	0x## 0x##	Адрес (Младший байт - Low byte) Адрес Бита (Старший байт - High byte) Например: MB12=>1*16+2=18, это 0x12 и 0x00
3	0x00(or 0x01)	Данные адреса MB . (Это биты, и должны быть: 0 или 1)
4 , 5	0x10 , 0x03	Стоповые биты
6	0x##	Контрольная сумма, «xor» от 0 байта к пятому байту

Для команд MW		
Смещение (байт)	**Формат**	**Описание**
0	0x01	Сигнал работы к MW
1 2	0x## 0x##	Адрес (Младший байт - Low byte) Адрес Бита (Старший байт - High byte) Если есть 0x10 включенный в адрес, и вставить 0x10 после него, байт будет перемещен на следующую позицию. Например: 0x10, 0x04 станет 0x10,0x10,0x04
3	0x##	Посылаемый байт (Байт должен быть четным (even), т.к. оперирует со словами). Если байт 0x10 затем вставлен 0x10 после него, байт будет перемещен на следующую позицию.
4~4+n-1	0x##(L) 0x##(H) 0x##(L)…	Данные начального адреса для соотносящихся адресов для 1,2 байтов, «n» – байты данных, если данные включают 0x10 и затем вставлен 0x10, номер посылаемого байта останется тем же самым, затем: n=n+1, и так далее…
4+n , 4+n+1	0x10 , 0x03	Знак конца
4+n+2	0x##	Контрольная сумма, проверка «xor» и байты впереди

Ниже приведен пример наблюдения за процессом обмена данными. Если у *Master* в MW3 находится 0x0a , в соответствии с этим протоколом, *master* немедленно будет обмениваться со *slave* , и *slave* положит 0x0a в соответствующий MW3, процедура следующая:

Master посылает STX(0x02h).

Slave получает STX(0x02h) от *master*, и посылает ACK(0x06h) *master*.

Master получает ACK(0x06h) от *slave*.

Смещение (байт)	**Формат**	**Описание**
0	0x01	Сигнал работы к MW
1 2	0x03 0x00	Адрес (Младший байт - Low byte) Адрес Бита (Старший байт - High byte)
3	0x02	Посылаемый байт (Байт должен быть четным (even), т.к. MW3 -двухбаайтовый)
4 , 5	0x0a , 0x00	Содержимое MW3: 0x0a , 0x00
6 , 7	0x10 , 0x03	Знак конца
8	0x19	Контрольная сумма, 0x01^0x03^0x00^0x02^0x0a^0x00^0x10^0x03=0x19

Slave получает данные от *master* и затем посылает ACK(0x06h).

Master получает ACK(0x06h) от *Slave*.

При завершении обмена, *Master* посылает ревизированные данные из MW в *Slave*, и *Slave* изменяет содержимое MW, которое соотносится с содержимым у *Master*. В этот момент, *Master* и *Slave* сохраняют одинаковые данные в одинаковых адресах.

Другой пример внизу, адрес и данные содержат 0x10; заметьте изменения в формате данных. Теперь, если у нас есть 0x10 в MW16 (*Slave*), в соответствии с этим протоколом, *Slave* свяжется с *Master* немедленно, и *Master* положит 0x10 в данные соответствующего регистра MW16, процедура следующая:

Slave посылает STX(0x02h)

Master получает STX(0x02h) от *Slave*, and посылает ACK(0x06h) *Slave*.

Slave получает ACK(0x06h) от *Master*

Slave посылает данные 0x01,0x10,0x10,0x00,0x02,0x10,0x10,0x00,0x10,0x03,0x10, как показано ниже:

Смещение (байт)	**Формат**	**Описание**
0	0x01	Сигнал работы к MW
1 2 3	0x10 0x10 0x00	Адрес (Младший байт - Low byte) Insert 0x10 Адрес Бита (Старший байт - High byte)
4	0x02	Sending byte (MW10 is two bytes)
5 6 7	0x10 0x10 0x00	0x10 – это младший байт MW10 Вставка 0x10 0x00 в старшем байте
8 , 9	0x10 , 0x03	Знак конца
10	0x10	Контрольная сумма, 0x01^0x10^0x10^0x00^0x02^0x10^0x10^0x00^0x10 ^0x03=0x10

Master получает данные от *slave* и посылает ACK(0x06h) *slave*.

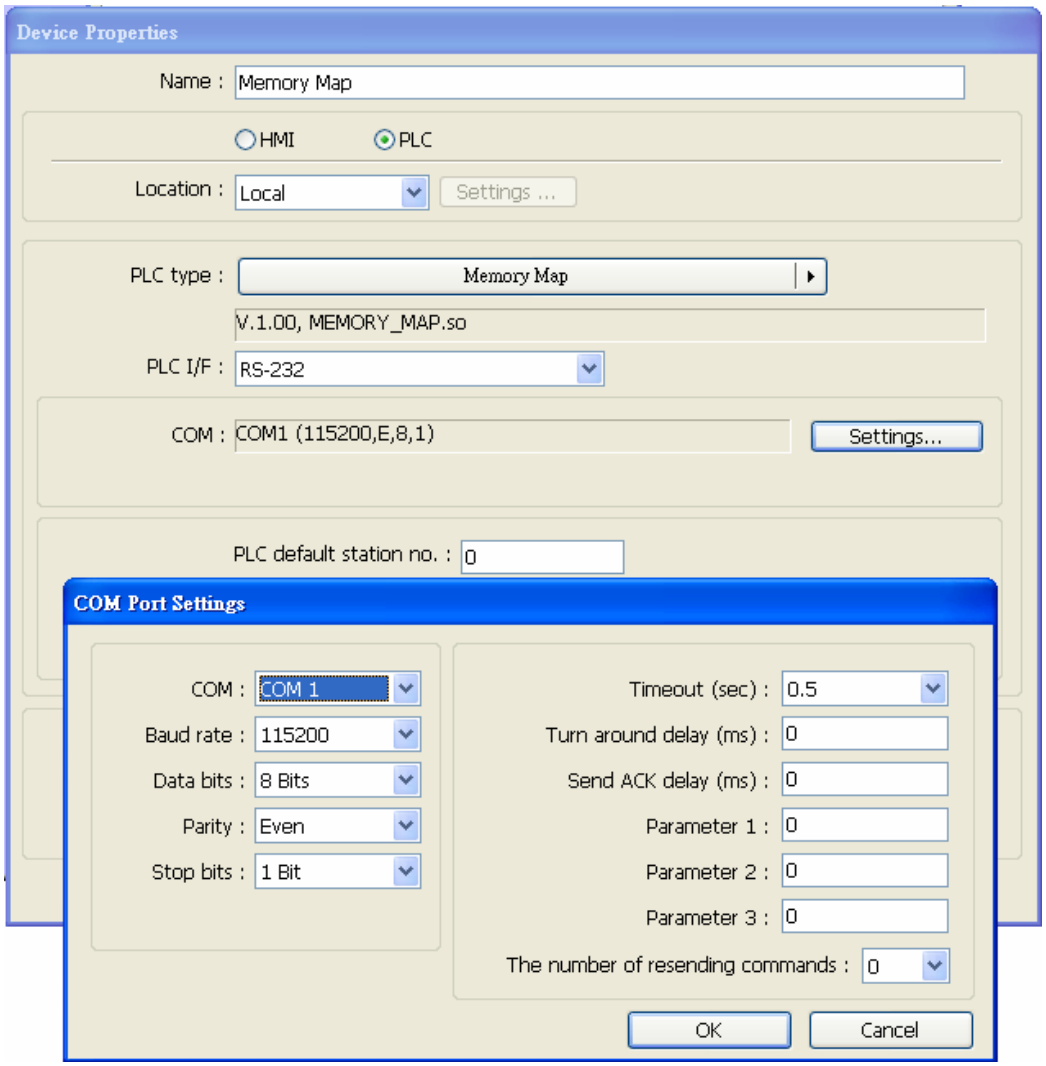
Slave получает ACK(0x06h) от *master*.

При завершении обмена, «Slave» посылает адрес и содержимое MW ведущему -«Master», Master изменяет данные регистра MW соотносящегося с регистром Slave, так что master и slave сохраняют одинаковые данные в одних и тех же адресах.

Далее приведен пример взаимодействия между двумя панелями HMI при помощи MemoryMap.

Сначала создайте новый проект в EasyBuilder Pro.

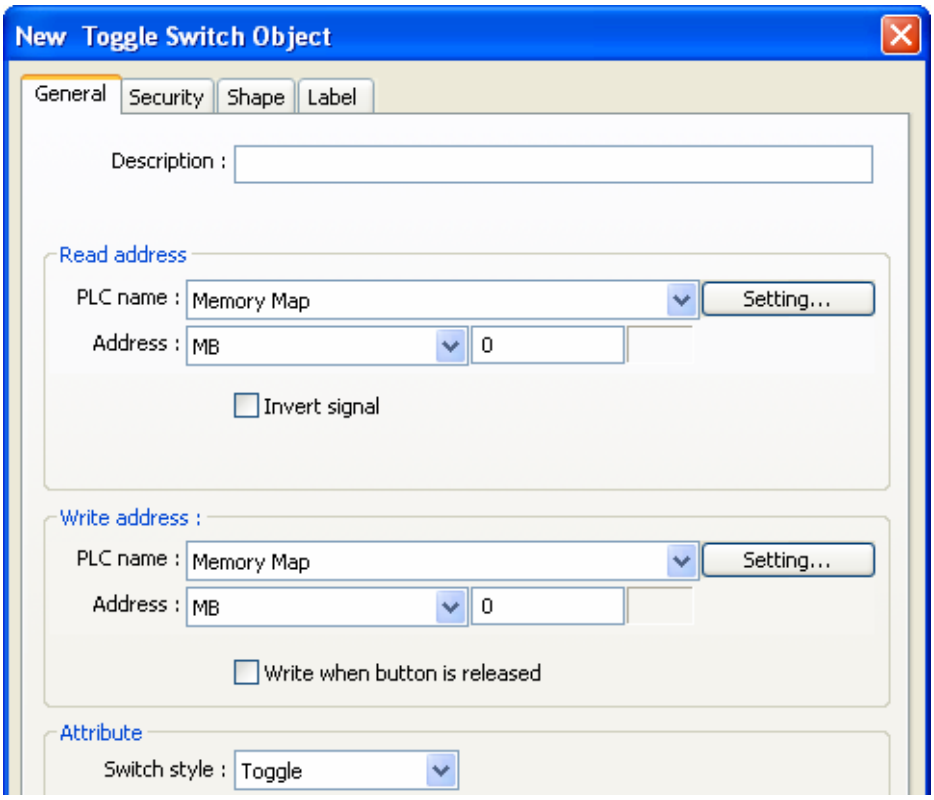
Отредактируйте : Edit/System Parameter Setting/PLC



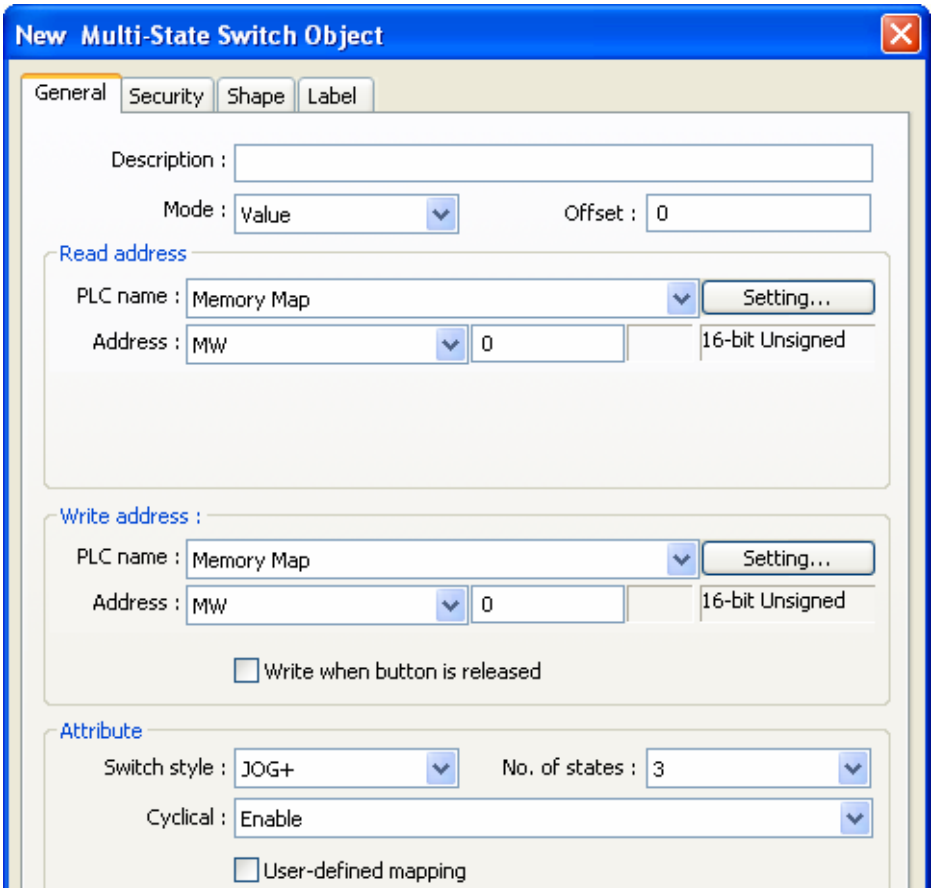
ПРИМЕЧАНИЕ:

1. Панель eMT3000 отличается от панели MT500, в которой память разделена на Memory Map_Master, MemoryMap, Slaver, поэтому разрешен простой выбор Memory Map.
2. [Data bit] должны иметь 8 бит.
3. Остальные настройки должны быть идентичными в обеих панелях.

Добавьте два объекта в «window10», настройки переключателя «toggle switch» показаны на рисунке на следующей странице:



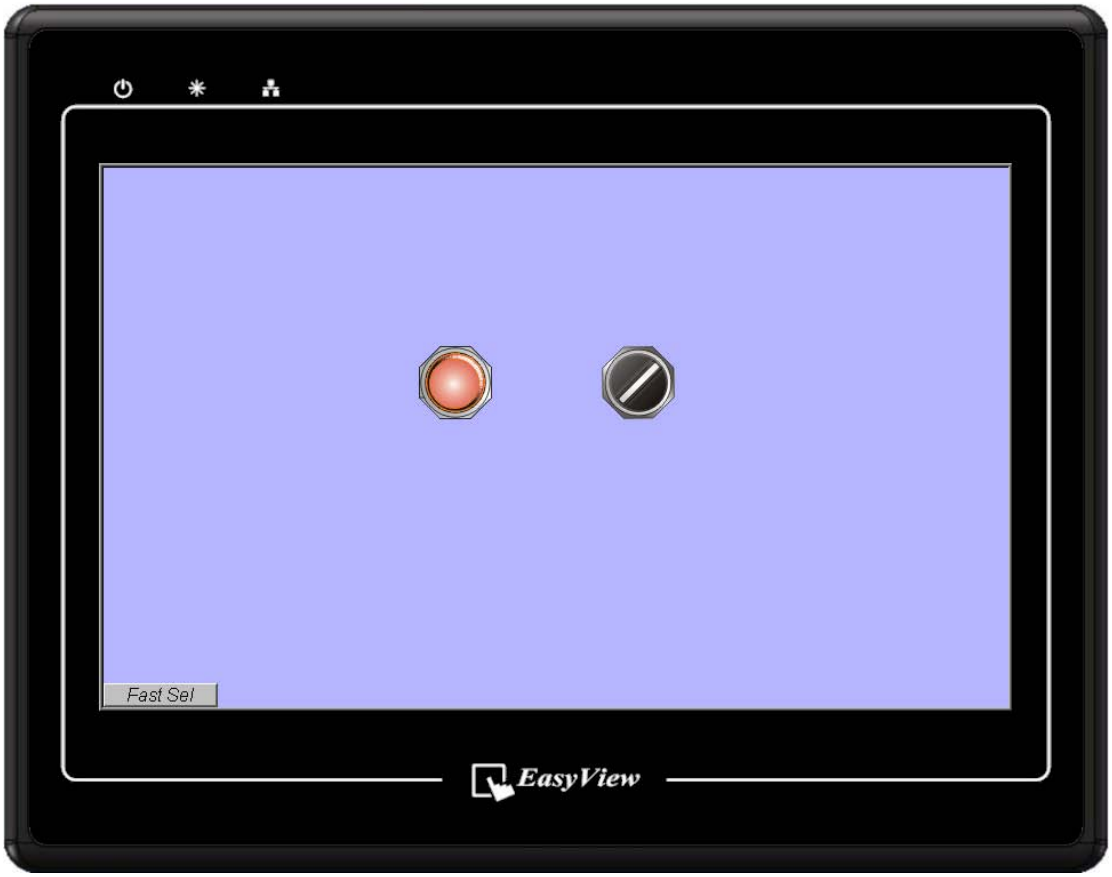
Объект Многопозиционный переключатель (multi-state switch) настройте так:



[Save],[Compile],[Download]

Измените параметры [System Parameter Setting]/[PLC] и загрузите их в другую панель.

На экране панели будет следующее:



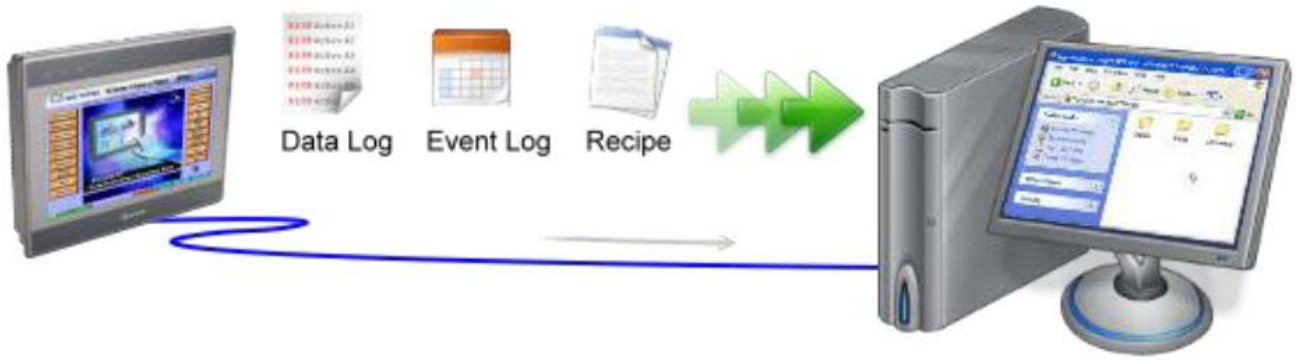
Пользователь может использовать сенсорный экран для проверки взаимодействия; другая панель будет действовать так же как первая.

Срособ обмена – как описано ранее.

Точки содержат одинаковые данные в одинаковых регистрах.

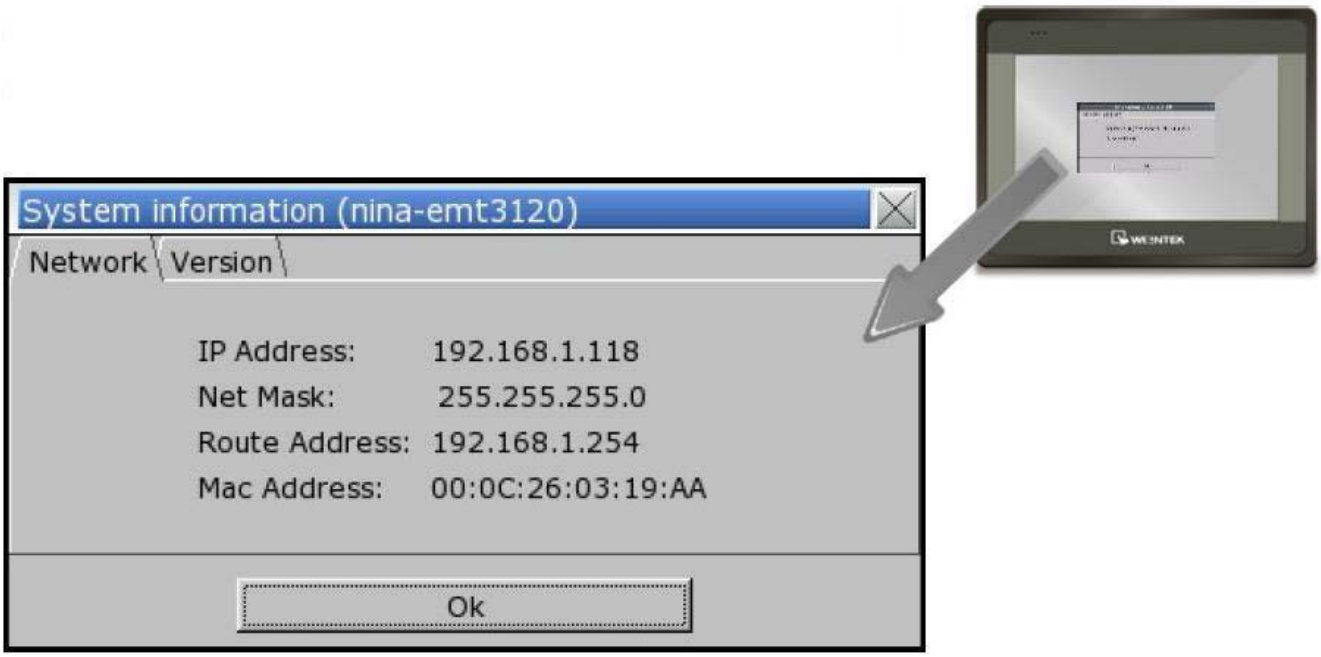
Глава 32. Использование сервера FTP

В дополнение к сохранению архивных данных из панели в компьюере при помощи SD-карт, USB-накопителей или функции *EasyPrinter*, для этого можно использовать сервер FTP. После загрузки проекта в панель, сервер FTP может быть использован для резервирования архивных данных и данных рецептов, а также для обновления данных рецептов. Файлы в сервере FTP не могут быть удалены.

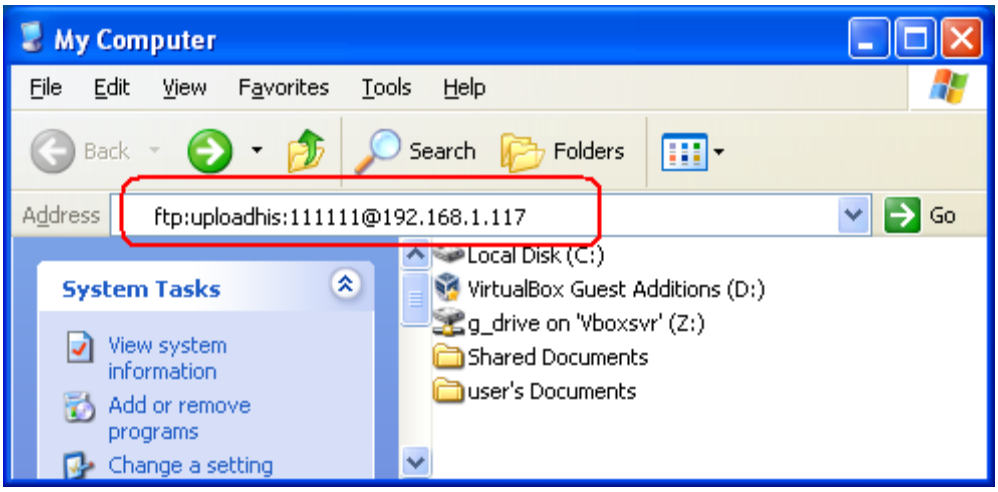


32.1 Login FTP Server

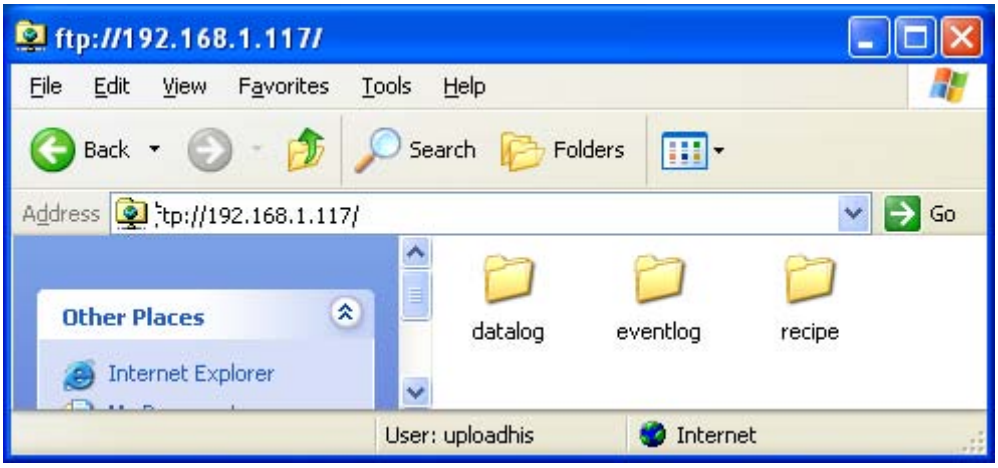
Шаг 1. Перед входом на сервер FTP Server, проверьте IP адрес панели.



Шаг 2. Введите IP адрес панели: ftp://192.168.1.117/ (пример), регистрационное имя пользователя (login user name): *uploadhis* и пароль на выгрузку архивов панели (если не изменяли, по умолчанию он - 111111). или прямо введите ftp://uploadhis:111111@192.168.1.117/



Шаг 3. После ввода IP, ftp://192.168.1.117/
как показано, можно увидеть папки “datalog”, “eventlog”, и “recipe”.



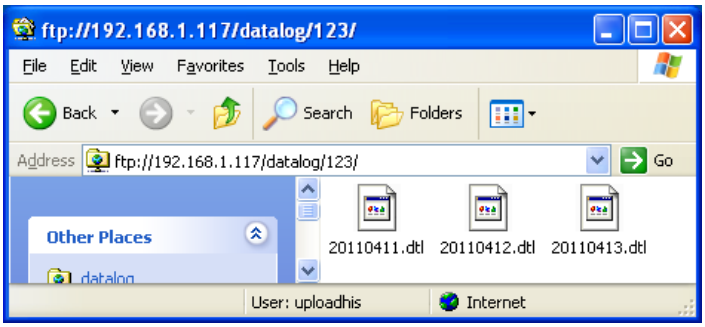
32.2 Резервное копирование архивных данных и обновление рецептов.

◆ **Чтобы копировать записи “Data Sampling”**

1. Выберите папку “datalog”, для уточнения имени файла созданного EasyBuilder Pro.

2. Откройте файл и просмотрите его.

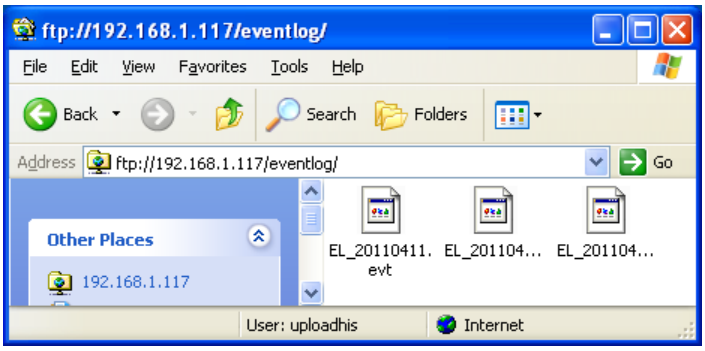
3. Сохраните файлы на компьютере при помощи «копирования-вставки»



◆ **Чтобы копировать записи “Event (Alarm) Log”**

1. Выберите папку “eventlog” для проверки файлов.

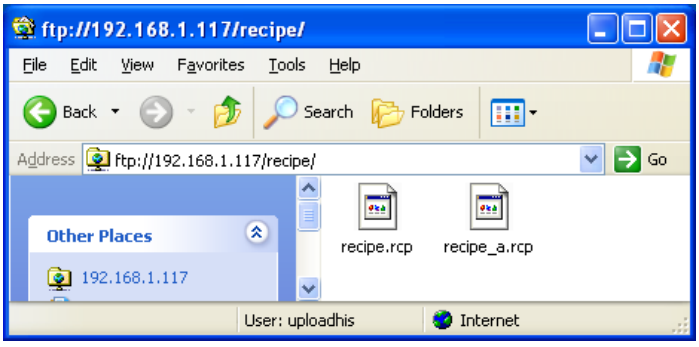
2. Сохраните файлы на компьютере при помощи «копирования-вставки»



◆ **To backup and update “Recipe” records**

1. Выберите папку “recipe” для проверки файлов.

2. Чтобы обновить данные “recipe” в панели, перепишите “recipe.rcp” новыми данными и перезагрузите панель в течении одной минуты.



Important ■ Так как данные рецептов автоматически сохраняютс каждую минуту, после обновления “recipe.rcp” или “recipe_a.rcp”, панель должна быть перезапущена в течении одной минуты,иначе новые обновленные данные будут перезаписаны прежними данными.

Для ре-старта панели, можно, также, использовать [LB-9047] и [LB-9048] . Установите [LB-9048] в ON и, затем, установите [LB-9047] в ON для ре-старта панели.

Системные регистры:

[LB-9047] - перезагружает панель (установить в ON, когда LB9048 - ON)

[LB-9048] - защита от перезагрузки панели

Глава 33. Отладчик - EasyDiagnoser

33.1 Введение и настройка

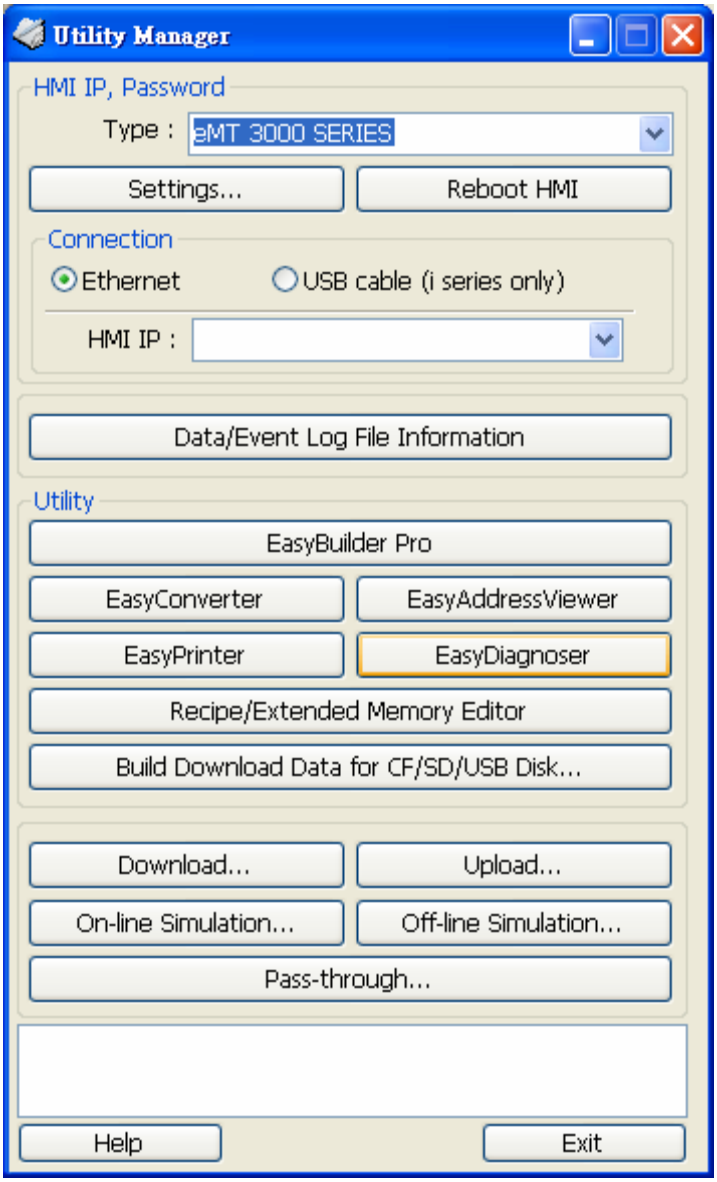
Введение

EasyDiagnoser – это инструмент для обнаружения ошибок при коммуникации панели с ПЛК.

Настройка

Шаг 1.

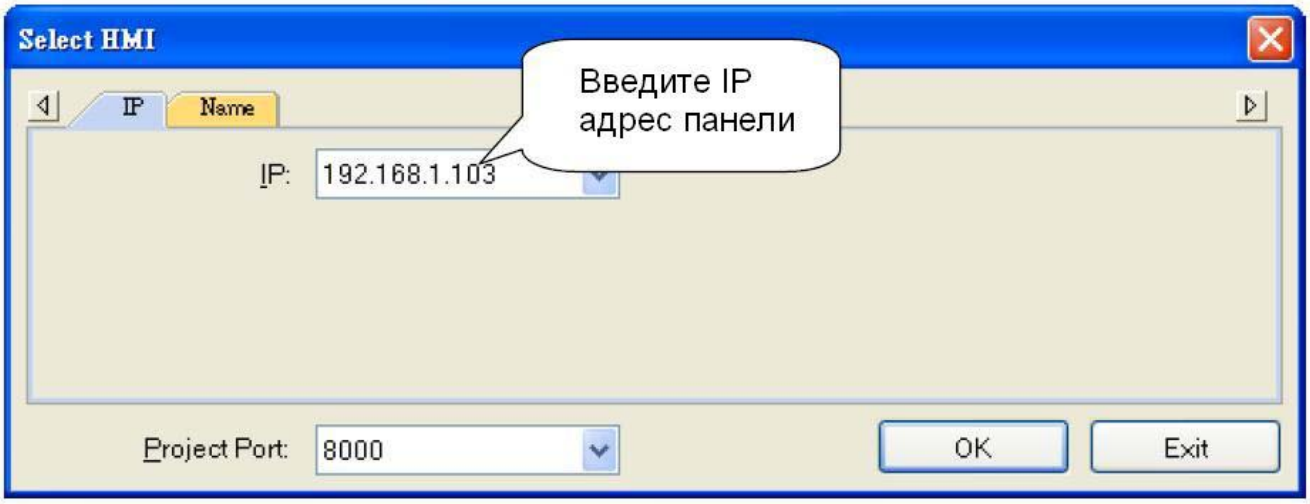
Откройте *Utility Manager* и щелкните по *EasyDiagnoser*.

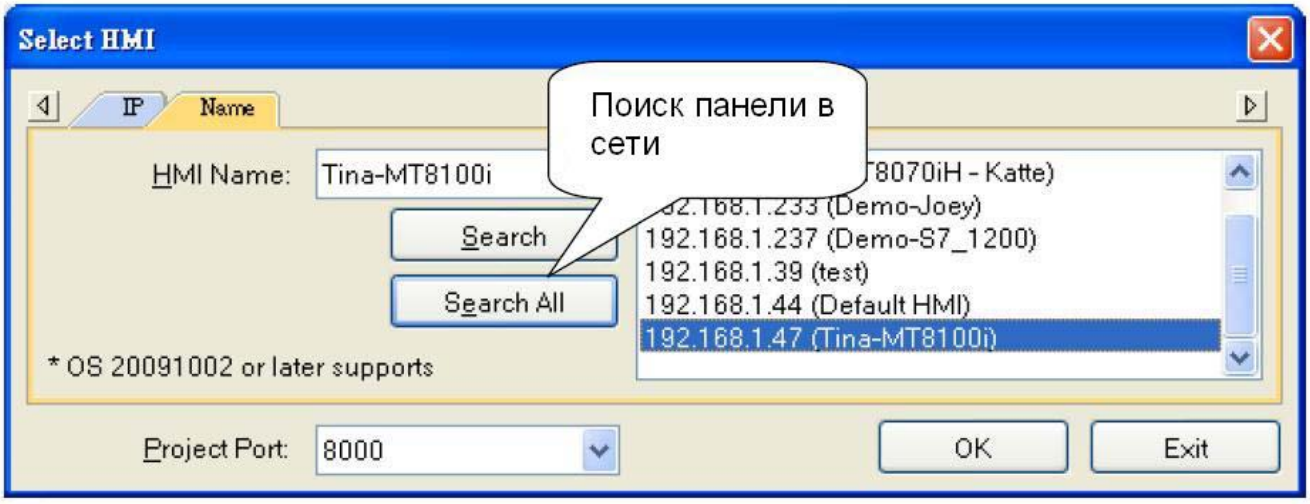


Шаг 2.

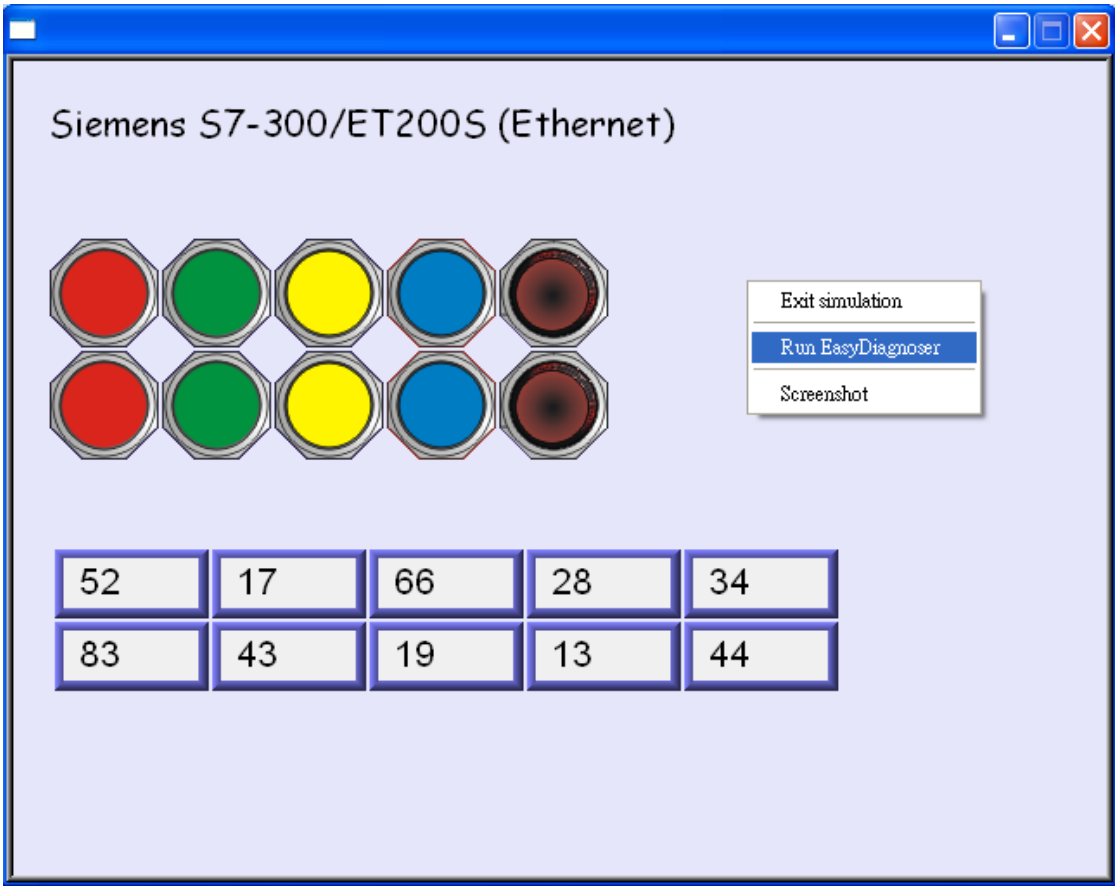
Установите IP-адрес панели, с которой хотите соединиться.

Можно ввести IP-адрес вручную или выбрав [Search All]. Введите *Project Port*, как показано далее.

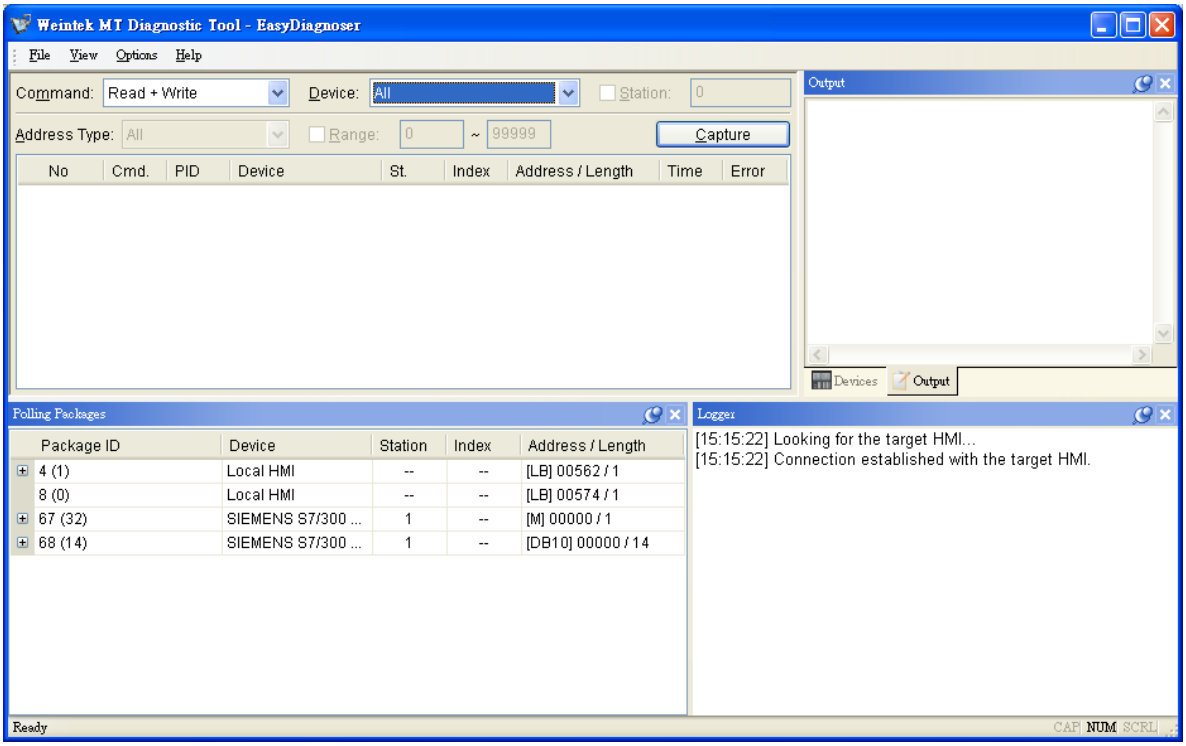

Select HMI
IP
Name
Введите IP адрес панели
IP:
192.168.1.103
Project Port:
8000
OK
Exit


Select HMI
IP
Name
Поиск панели в сети
HMI Name:
Tina-MT8100i
Search
Search All
192.168.1.237 (Demo-S7_1200)
192.168.1.39 (test)
192.168.1.44 (Default HMI)
192.168.1.47 (Tina-MT8100i)
* OS 20091002 or later supports
Project Port:
8000
OK
Exit

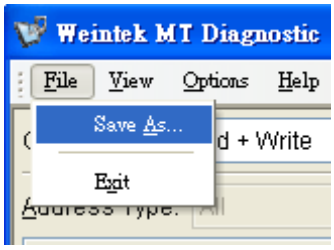
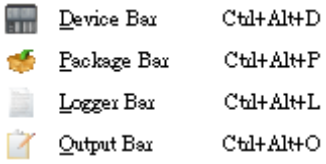
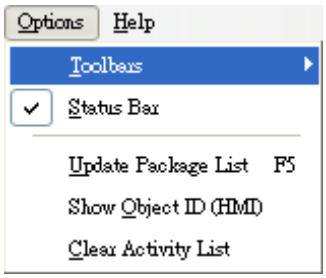
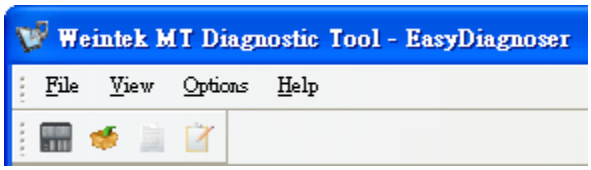
Можно, также, сделать щелчок правой кнопкой мышки и выбрать “Run EasyDiagnoser” для входа в окно настроек, если Вы находитесья в режиме *On-Line Simulation* EasyBuilder Pro.

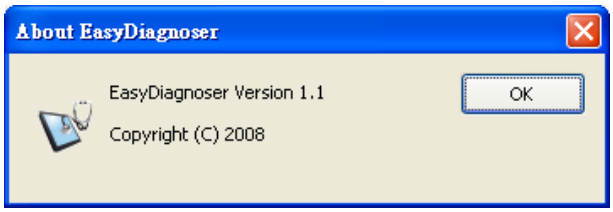


После завершения, нажмите OK, откроется рабочее окно EasyDiagnoser:



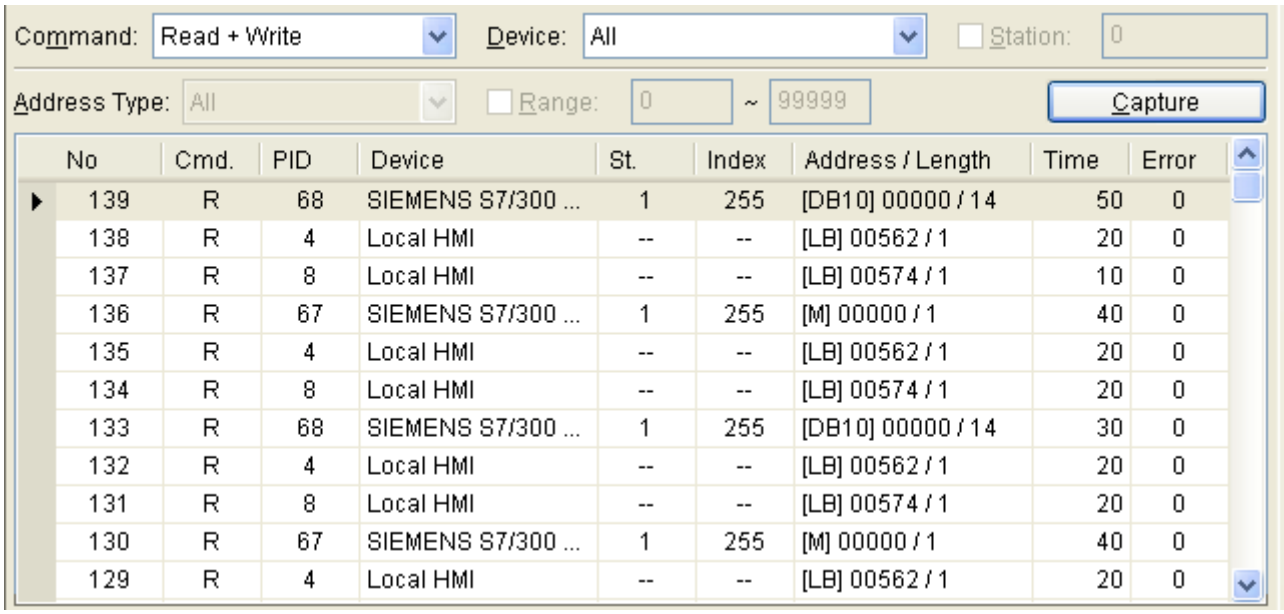
33.2 Настройки EasyDiagnoser

Элемент	Описание
File	**Save As** Захваченная информация Easy Diagnoser может быть сохранена, как файл *.xls , который можно прочитать в Excel. 
	Exit Выход из текущего файла.
View 	Щелкните по [Device Bar], чтобы открыть окно *Device.* Щелкните по [Package Bar], чтобы открыть окно *Package* . Щелкните по [Logger Bar], чтобы открыть окно *Logger*. Щелкните по [Output Bar], чтобы открыть окно *Output*.
Options 	**Toolbars - Панель инструментов** Отображает пиктограммы инструментов [Device Bar] [Package Bar] [Logger Bar] [Output Bar]. 
	Show Status Bar – Показать строку состояния В нижней части окна EasyDiagnoser window, отображена информация о CAP, NUM, и SCRL.
	Update Package List – Обновить список Опрашиваемого пакета Когда пользователь меняет окно на панели, обновите информацию в пакете опрашиваемых данных (Polling Package) текущего окна в этом списке.
	Show Object ID (HMI) – Показать идентификатор объекта Показать идентификаторы (ID) объектов панели, как показано на следующей странице.

	Clear Activity List Очистить всю информацию в активной области.
Help	Показывает версию EasyDiagnoser version . 

- **Активная область - Activity area**

В активной области пользователь может наблюдать за обменом данными между панелью и ПЛК.



No	Cmd.	PID	Device	St.	Index	Address / Length	Time	Error
139	R	68	SIEMENS S7/300 ...	1	255	[DB10] 00000 / 14	50	0
138	R	4	Local HMI	--	--	[LB] 00562 / 1	20	0
137	R	8	Local HMI	--	--	[LB] 00574 / 1	10	0
136	R	67	SIEMENS S7/300 ...	1	255	[M] 00000 / 1	40	0
135	R	4	Local HMI	--	--	[LB] 00562 / 1	20	0
134	R	8	Local HMI	--	--	[LB] 00574 / 1	20	0
133	R	68	SIEMENS S7/300 ...	1	255	[DB10] 00000 / 14	30	0
132	R	4	Local HMI	--	--	[LB] 00562 / 1	20	0
131	R	8	Local HMI	--	--	[LB] 00574 / 1	20	0
130	R	67	SIEMENS S7/300 ...	1	255	[M] 00000 / 1	40	0
129	R	4	Local HMI	--	--	[LB] 00562 / 1	20	0

Элемент	Описание
Command	**a. Read + Write** Отображать команды *Read / Write* в активной области.
	b. Read Отображать только команду *Read* в активной области.
	c. Write Отображать только команду *Write* в активной области.

Элемент	Описание
Device	**a. All** Отображение информации о локальной панели и ПЛК. Оно зависит от настроек команды: • Если команда настроена на **Read + Write**, Read и Write информация локальной панели и ПЛК будет отображаться в активной области. • Если команда настроена на **Read**, Read (считываемая) информация локальной панели и ПЛК будет отображаться в активной области. • Если команда настроена на **Write**, Write (записываемая) информация локальной панели и ПЛК будет отображаться в активной области.
	b. Local HMI Отображение информации о локальной панели. Оно зависит от настроек команды: • Если команда настроена на **Read + Write**, Read и Write информация локальной панели будет отображаться в активной области. • Если команда настроена на **Read**, Read информация локальной панели будет отображаться в активной области. • Если команда настроена на **Write**, Write информация локальной панели будет отображаться в активной области.
	c. PLC Отображение информации о ПЛК. Оно зависит от настроек команды:. • Если команда настроена на **Read + Write**, Read и Write информация ПЛК будет отображаться в активной области. • Если команда настроена на **Read**, Read информация ПЛК будет отображаться в активной области. • Если команда настроена на **Write**, Write информация ПЛК будет отображаться в активной области.
Station	Выбор конкретной *Station* для отображения на экране. (Эта функция будет отключена при выборе [All] в *Device*).
Address Type	Пользователь может выбирать все или часть типов адресов для отображения на экране. (Эта функция будет отключена при выборе [All] в *Device*).
Range	Установка диапазона адресов отображаемых на экране. (Эта функция будет отключена при выборе [All] в *Device*).
Capture	Щелкнуть по «start/stop» для захвата коммуникационного сообщения.
Error	Ошибки описаны далее.

- **Polling Packages – Пакеты опрашиваемых данных**

Polling Packages

Package ID	Device	Station	Index	Address / Length
4 (1)	Local HMI	--	--	[LB] 00562 / 1
8 (0)	Local HMI	--	--	[LB] 00574 / 1
67 (32)	SIEMENS S7/300 Ethernet	1	--	[M] 00000 / 1
68 (3)	SIEMENS S7/300 Ethernet	1	10	[DB10] 00000 / 3
69 (3)	SIEMENS S7/300 Ethernet	1	11	[DB10] 00003 / 3
70 (3)	SIEMENS S7/300 Ethernet	1	12	[DB10] 00006 / 3
71 (5)	SIEMENS S7/300 Ethernet	1	--	[DB10] 00009 / 5

Элемент	Описание
Package ID	Используйте информацию об идентификаторе пакета, чтобы проверять PID в активной области для поиска проблем.
Device	Отображает тип панели и ПЛК.
Station	Отображает Номер станции ПЛК (PLC station number).
Index	Отображает число используемых объектом индексных регистров.
Address/ Length	Address - отображает тип данных (device type). Length – сколько слов в Пакете,

Polling Packages

Object		Screen	ID	Address
4 (1)	Local HMI	--	--	[LB] 00562 / 1
8 (0)	Local HMI	--	--	[LB] 00574 / 1
67 (32)	SIEMENS S7/300 Ethernet	1	--	[M] 00000 / 1
Toggle S...		10	30	[M] 00000
Toggle S...		10	30	[M] 00000
Toggle S...		10	29	[M] 00000
Toggle S...		10	29	[M] 00000
Toggle S...		10	28	[M] 00000
Toggle S...		10	28	[M] 00000
Toggle S...		10	27	[M] 00000

Элемент	Описание
Object	Идентификатор Пакета (Package ID), где размещен объект.
Screen	Окно в проекте, где размещен объект.
ID	Идентификатор объекта.
Address	Адрес объекта.

ПРИМЕЧАНИЯ:

a. Щелкните по **[Package ID]**, номер станции устройства будет показан в третьей колонке.

Polling Packages

Package ID	Device	Station	Index	Address / Length
4 (1)	Local HMI	--	--	[LB] 00562 / 1
8 (0)	Local HMI	--	--	[LB] 00574 / 1
67 (32)	SIEMENS S7/300 Ethernet	1	--	[M] 00000 / 1
68 (3)	SIEMENS S7/300 Ethernet	1	10	[DB10] 00000 / 3

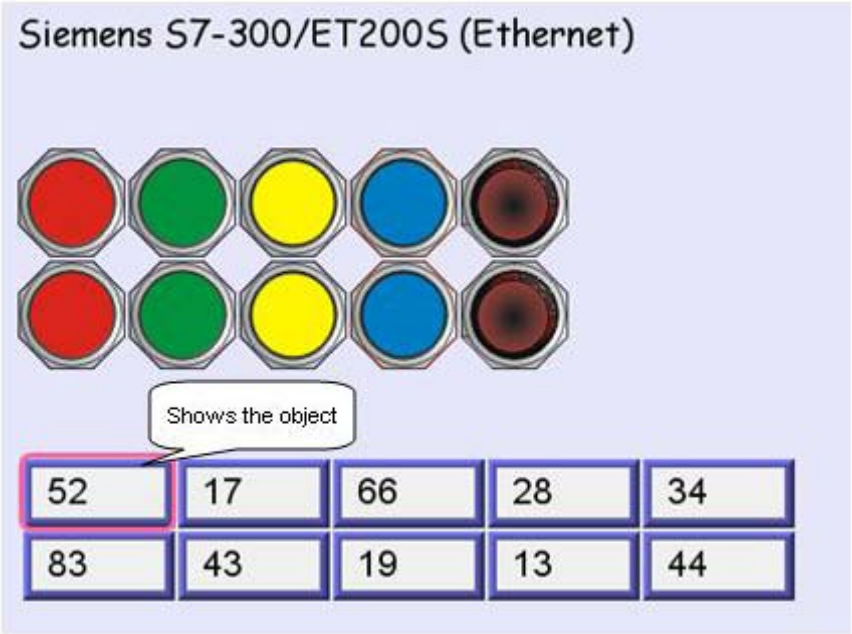
b. Сделайте двойной щелчок по **[Package ID],** затем выберите **[object]**, 1-ая колонка укажет позицию объекта.

Например, выберите [Numeric Input] и появится Номер экрана -10.

Это показывает, что этот объект находится в окне номер 10 в проекте и будет отмечен розовой рамкой на экране панели, как показано ниже.

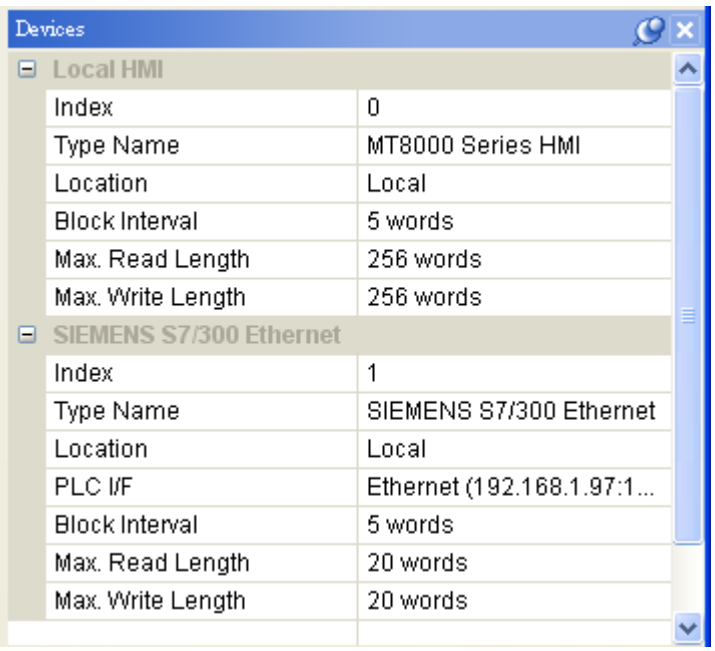
Polling Packages

Object		Screen	ID	Address
4 (1)	Local HMI	--	--	[LB] 00562 / 1
8 (0)	Local HMI	--	--	[LB] 00574 / 1
67 (32)	SIEMENS S7/300 Ethernet	1	--	[M] 00000 / 1
68 (3)	SIEMENS S7/300 Ethernet	1	10	[DB10] 00000 / 3
Numeric I...		10	2	[DB10] 00000
Numeric I...		10	3	[DB10] 00001
Numeric I...		10	4	[DB10] 00002



- **Devices – Устройства**

Окно *Devices* содержит информацию о панели (HMI) и ПЛК.



- **Output (Macro debug) – Выход (Отладчик Макросов)**

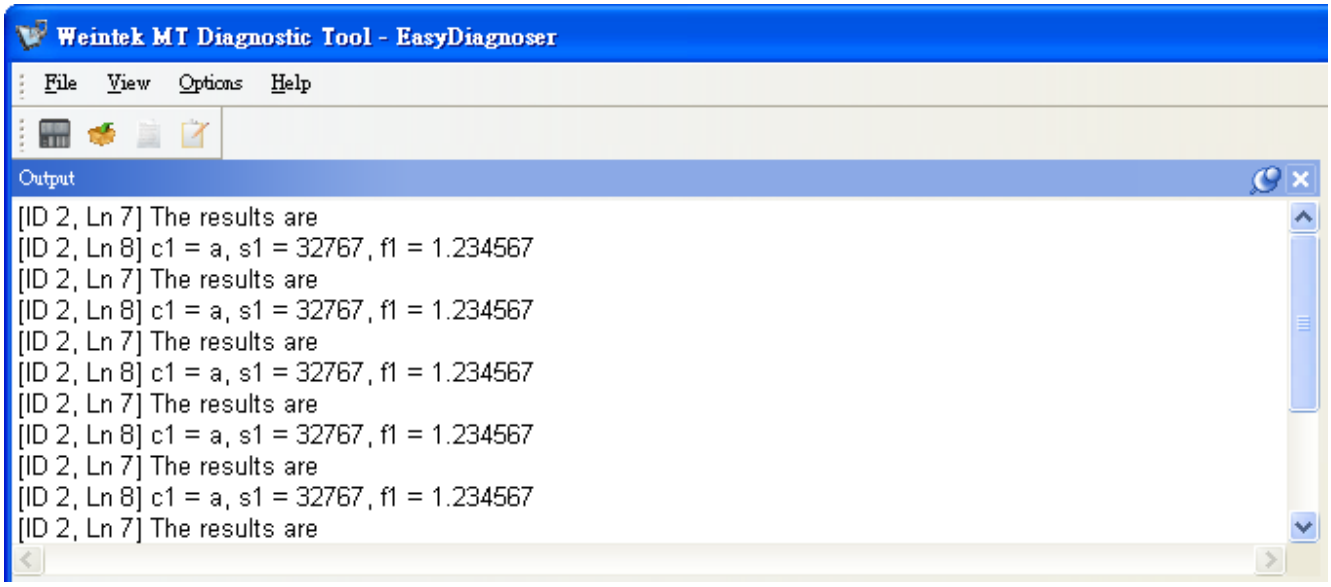
С функцией *Trace* используемой с Macro, можно увидеть состояние исполнения Macro.

Подробности в *“Главе 18. MACRO”* .

На рисунке внизу, для [ID 2, Ln 7] и [ID 2, Ln 8]

ID 2 – представляет имя Макроса.

Ln 7 и Ln 8 представляют 7-ую и 8-ую строки Макроса.



33.3 Код ошибки - Error Code

В активной области, можно найти причину ошибки при помощи списка кодов ошибок:

0: Normal - Нормально

1: Time out – Превышено время ожидания

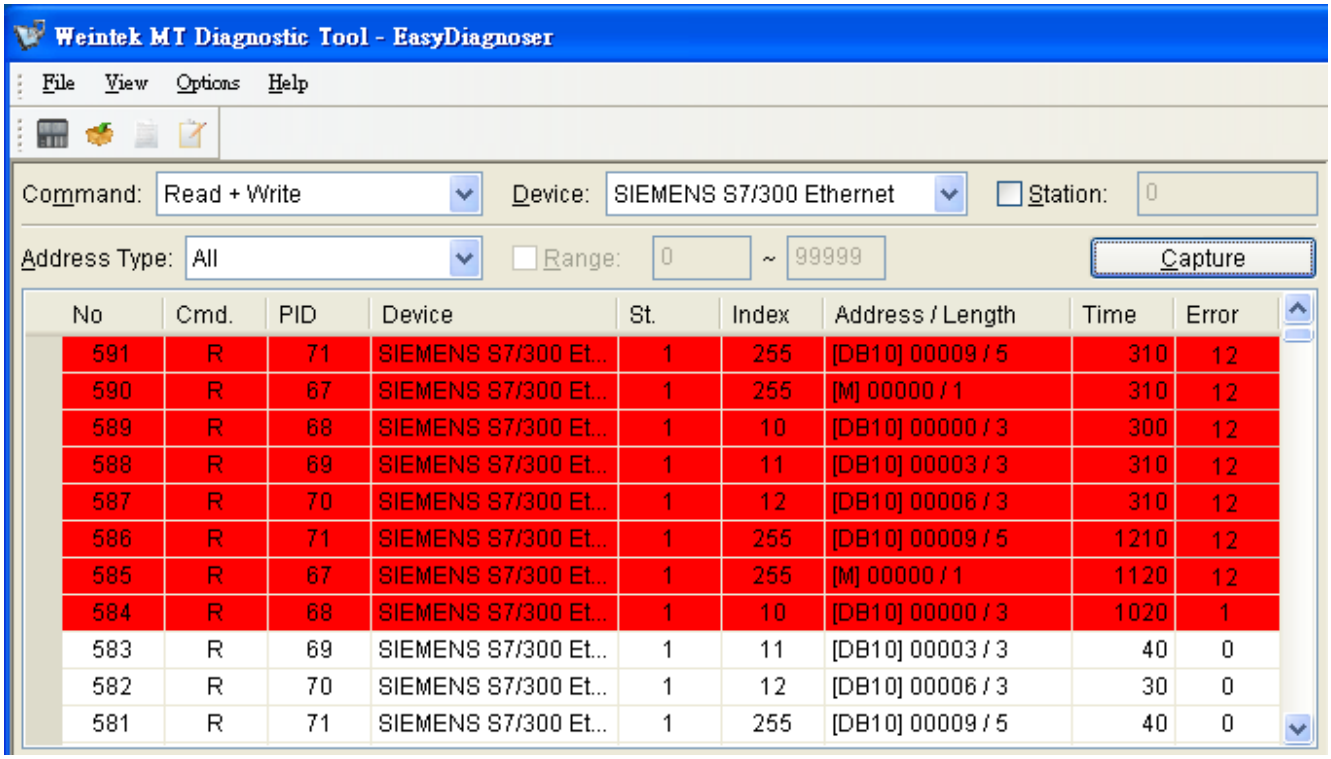
2: Fail Error – Ошибка исполнения

12: Ignore - Игнорировать

При возникновении ошибки, сообщение об ошибке выделяется красным, как показано ниже.

Код ошибки 1 - ПЛК отсоединен от панели.

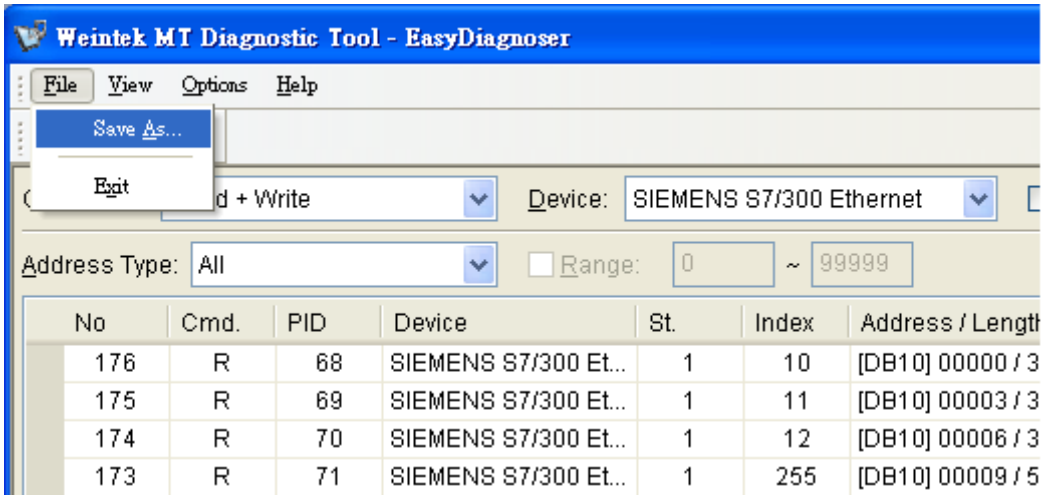
Код ошибки 12 – Окно с сообщением “PLC No Response” показано.



No	Cmd.	PID	Device	St.	Index	Address / Length	Time	Error
591	R	71	SIEMENS S7/300 Et...	1	255	[DB10] 00009 / 5	310	12
590	R	67	SIEMENS S7/300 Et...	1	255	[M] 00000 / 1	310	12
589	R	68	SIEMENS S7/300 Et...	1	10	[DB10] 00000 / 3	300	12
588	R	69	SIEMENS S7/300 Et...	1	11	[DB10] 00003 / 3	310	12
587	R	70	SIEMENS S7/300 Et...	1	12	[DB10] 00006 / 3	310	12
586	R	71	SIEMENS S7/300 Et...	1	255	[DB10] 00009 / 5	1210	12
585	R	67	SIEMENS S7/300 Et...	1	255	[M] 00000 / 1	1120	12
584	R	68	SIEMENS S7/300 Et...	1	10	[DB10] 00000 / 3	1020	1
583	R	69	SIEMENS S7/300 Et...	1	11	[DB10] 00003 / 3	40	0
582	R	70	SIEMENS S7/300 Et...	1	12	[DB10] 00006 / 3	30	0
581	R	71	SIEMENS S7/300 Et...	1	255	[DB10] 00009 / 5	40	0

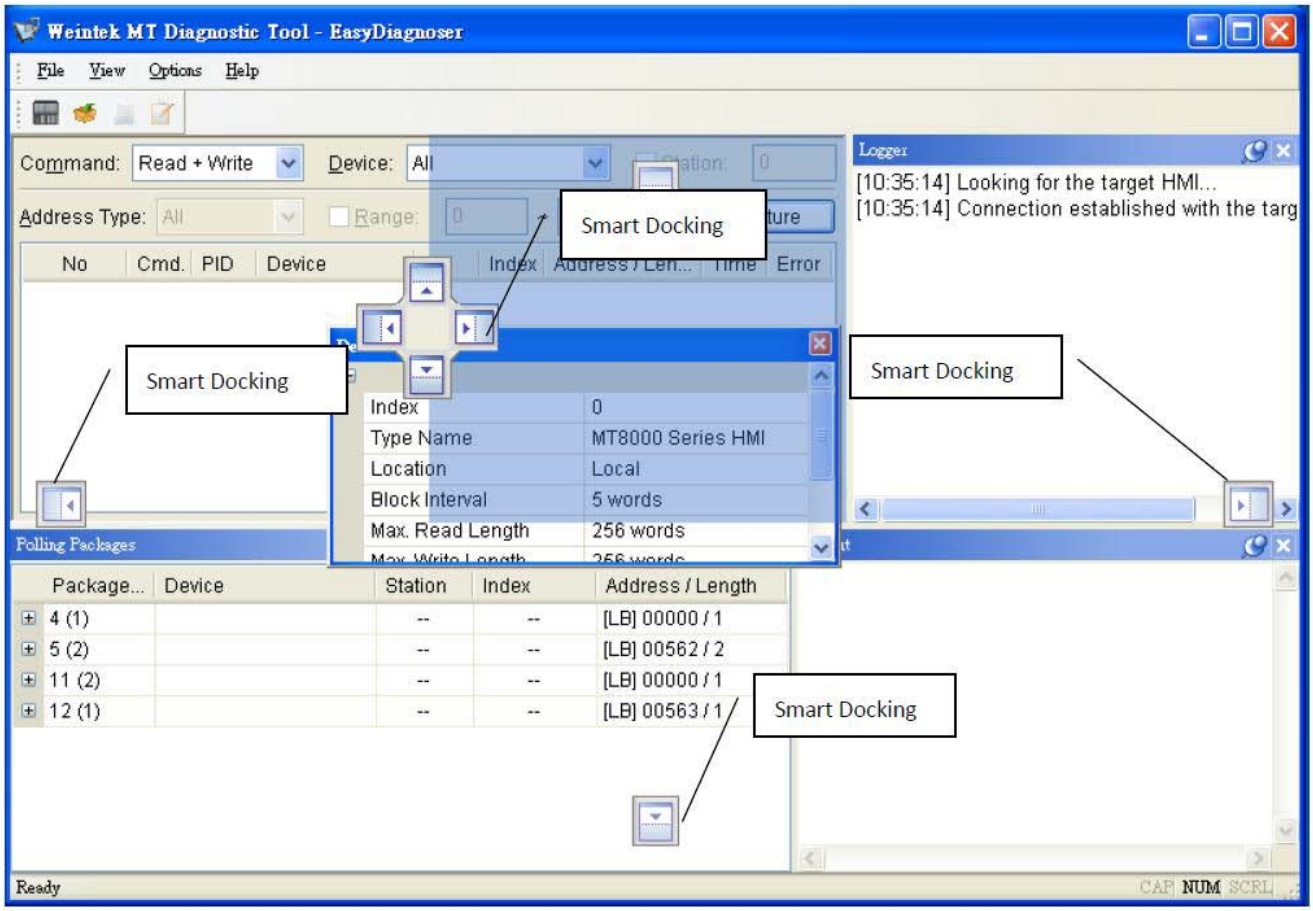
33.4 Сохранить как - Save As

Захваченная Easy Diagnoser информация может быть сохранена, как файл *.xls , который может быть открыт в Excel.



33.5 Настройка окна - Window Adjustment

Пользователь может перетаскивать окно или использовать пиктограммы прикрепления - «smart docking» в окне редатора, чтобы разместить окна в желаемом положении.



ПРИМЕЧАНИЕ:

EasyDiagnoser не поддерживает Siemens S7/1200 (Ethernet) и Allen-Bradley Ethernet/IP (CompactLogix/ControlLogix) – Free Tag Names , так как оба этих ПЛК используют тэги.

Глава 34. Свободные тэги Rockwell EtherNet/IP

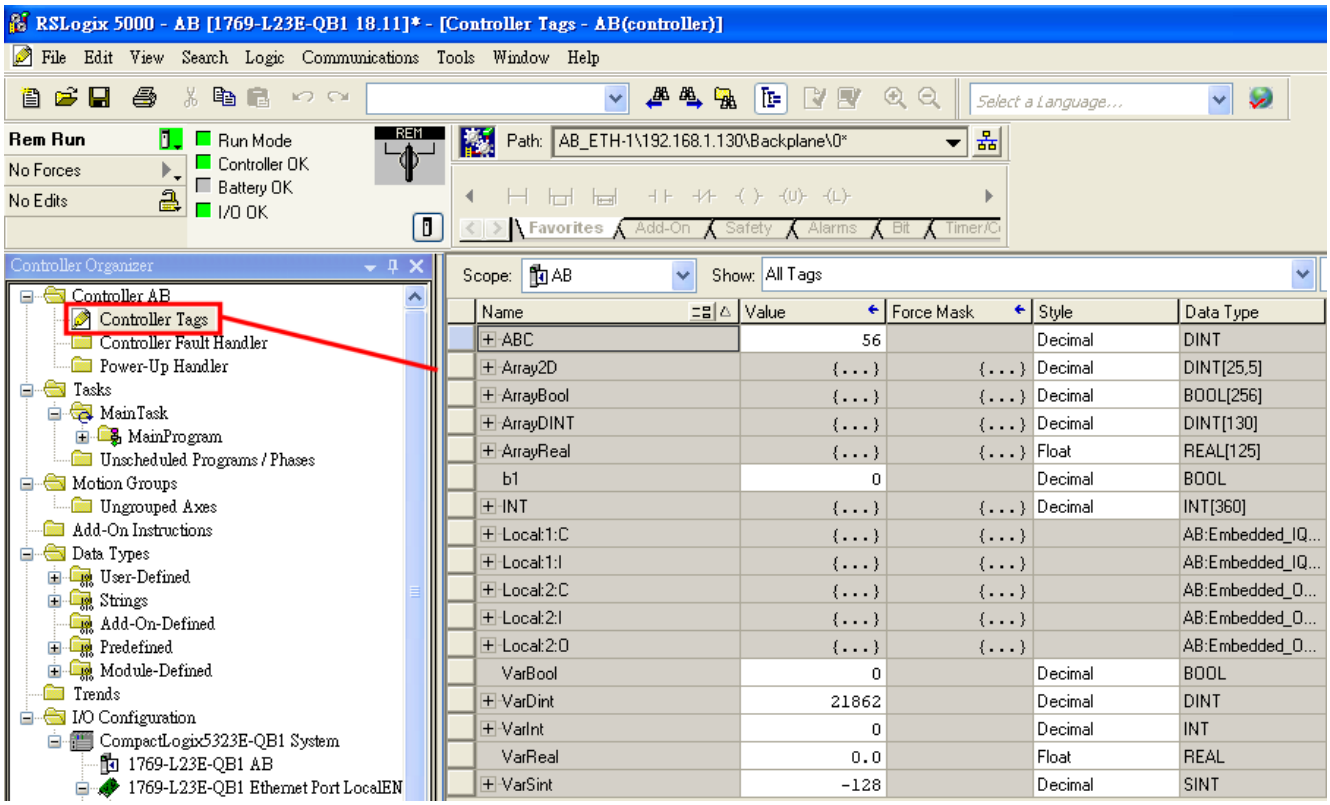
При использовании драйвера Rockwell EtherNet/IP-Tag (CompactLogix/ ControlLogix) в EasyBuilder Pro, пользователь может импортировать *User-Defined Tag* из файла «CSV» RSLogix5000. Однако, данные типа *User-Defined*, *Predefined* и *Module-Defined Structure* не будут импортированы.

	A	B	C	D	E	F	
7	TYPE	SCOPE	NAME	DESCRIPT	DATATYPE	SPECIFIER	ATTRIBUTES
8	TAG		Local:1:C		AB:Embedded_IQ16F:C:0		
9	TAG		Local:1:I		AB:Embedded_IQ16F:I:0		
10	TAG		Local:2:C		AB:Embedded_OB16:C:0		
11	TAG		Local:2:I		AB:Embedded_OB16:I:0		
12	TAG		Local:2:O		AB:Embedded_OB16:O:0		
13	TAG		Array2D		DINT[25,5]		(RADIX := Decimal, Cons
14	TAG		ArrayBool		BOOL[256]		(RADIX := Decimal, Cons
15	TAG		ArrayDINT		DINT[130]		(RADIX := Decimal, Cons
16	TAG		ArrayReal		REAL[125]		(RADIX := Float, Constant
17	TAG		B001		INT[15]		(RADIX := Decimal, PLCM
18	TAG		b003		INT[255]		(RADIX := Decimal, PLCM
19	TAG		b1		BOOL		(RADIX := Decimal, Cons

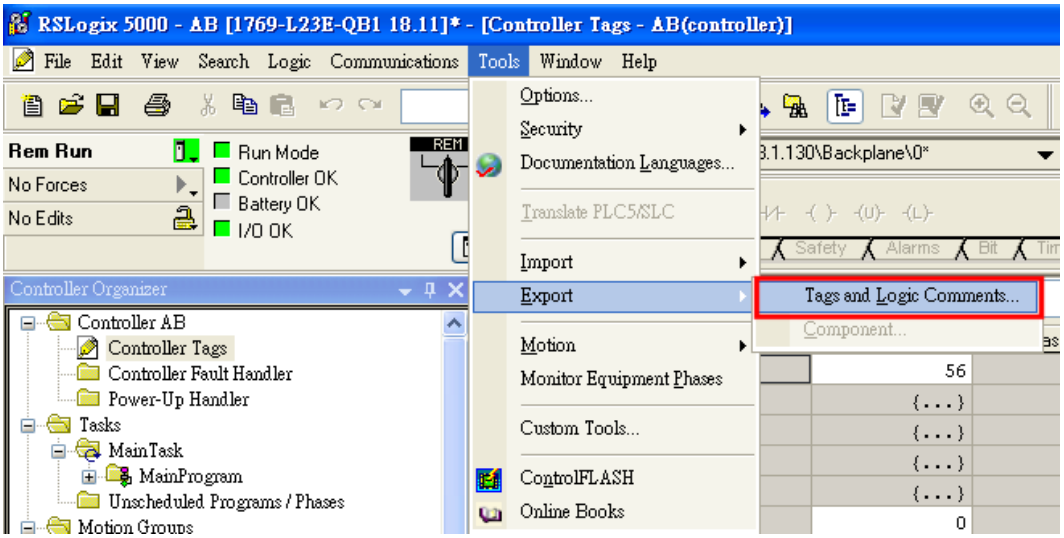
Поэтому, необходимо использовать *Structure Editor* в EasyBuilder Pro для импорта и редактирования структур *User-Defined*, *Predefined* и *Module-Defined* .

34.1 Импорт файла CSV User-Defined Tag в EasyBuilder Pro

Шаг 1. Создайте тэги в RSLogix5000.

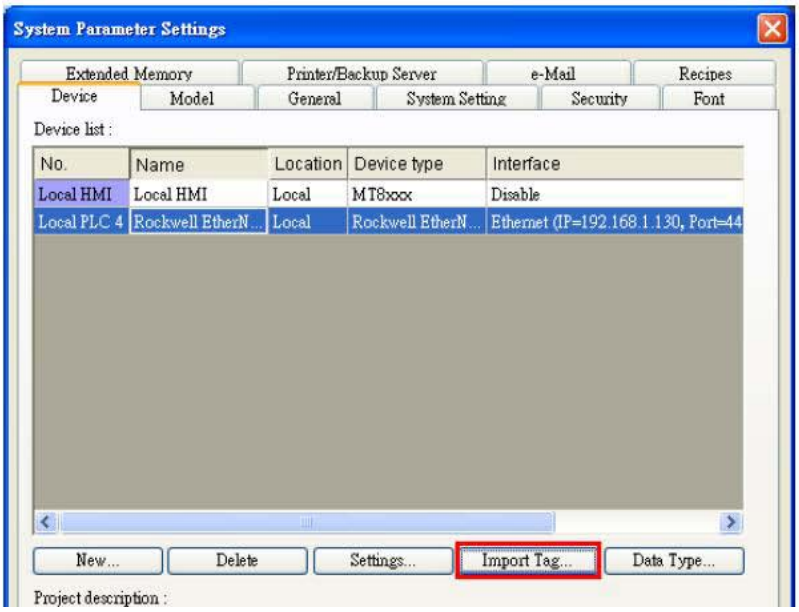


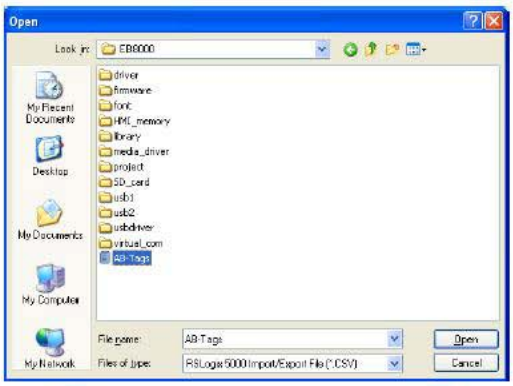
Шаг 2. Экспортируйте данные тэгов в файл CSV.



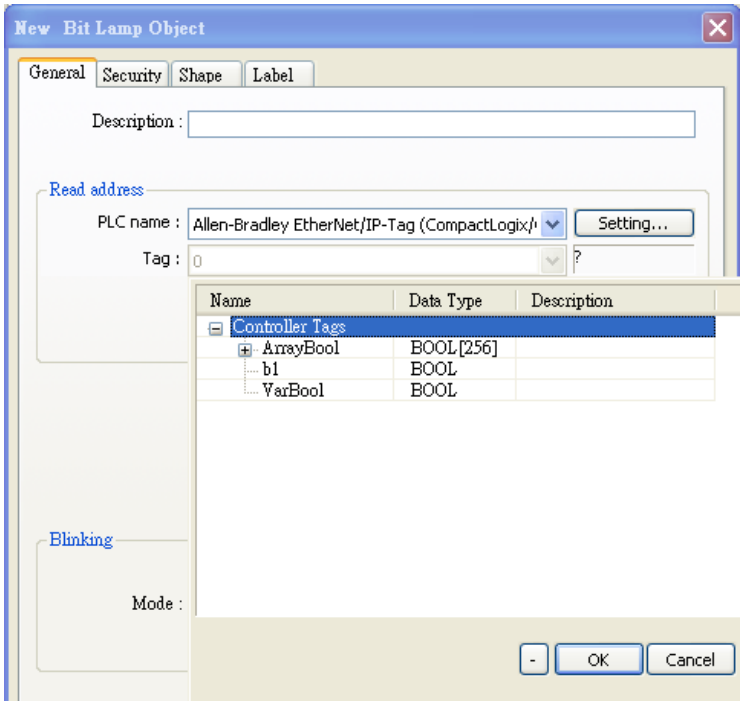
Шаг 3. В EasyBuilder Pro, создайте драйвер Rockwell EtherNet/IP-Tag (CompactLogix/ControlLogix) .

Введите IP-адрес ПЛК. В окне *System Parameter Settings* нажмите кнопку [Import Tag…]



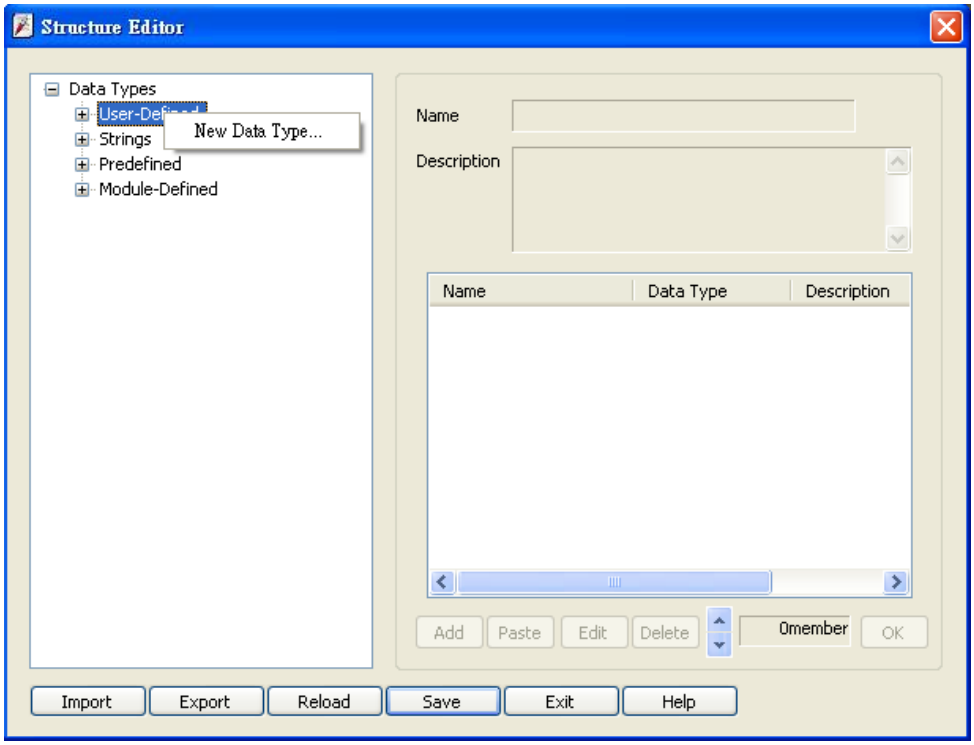


Шаг 4. В окне объекта, выберите ПЛК, щелкните по *Controller Tags* и выберите нужный тэг.



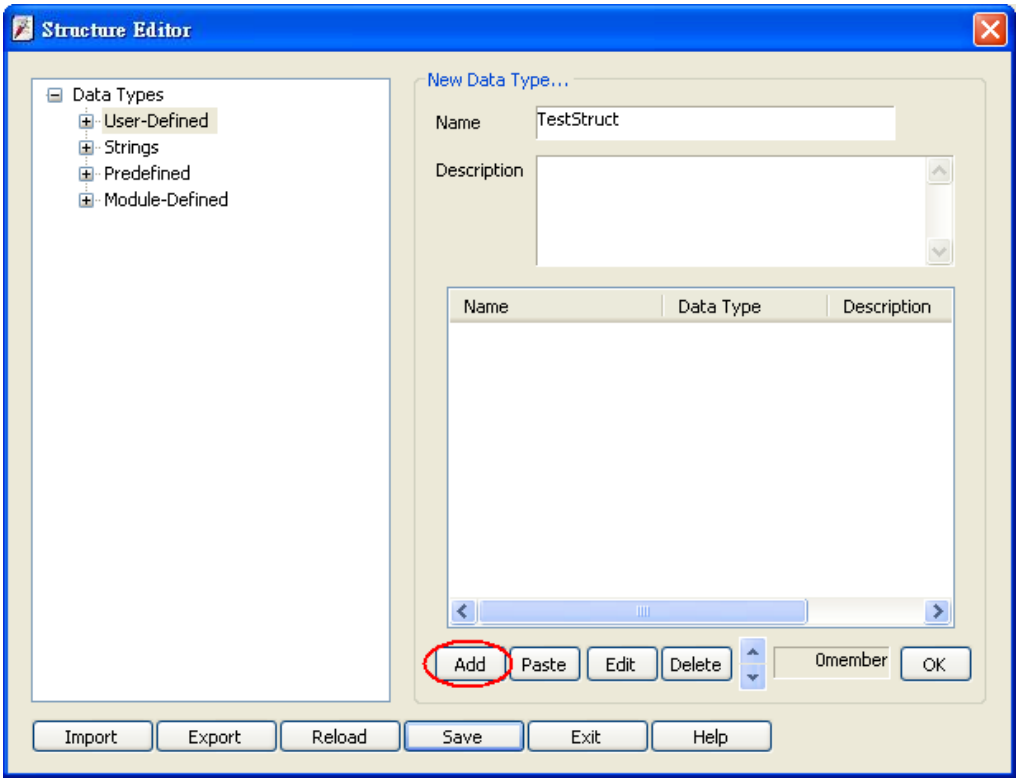
34.2 Adding New Data Type

Шаг 1. Щелчок правой кнопкой по типу данных (обычно отмеченного, как [User-Defined]), затем щелкните по [New Data Type], чтобы начать редактирование.

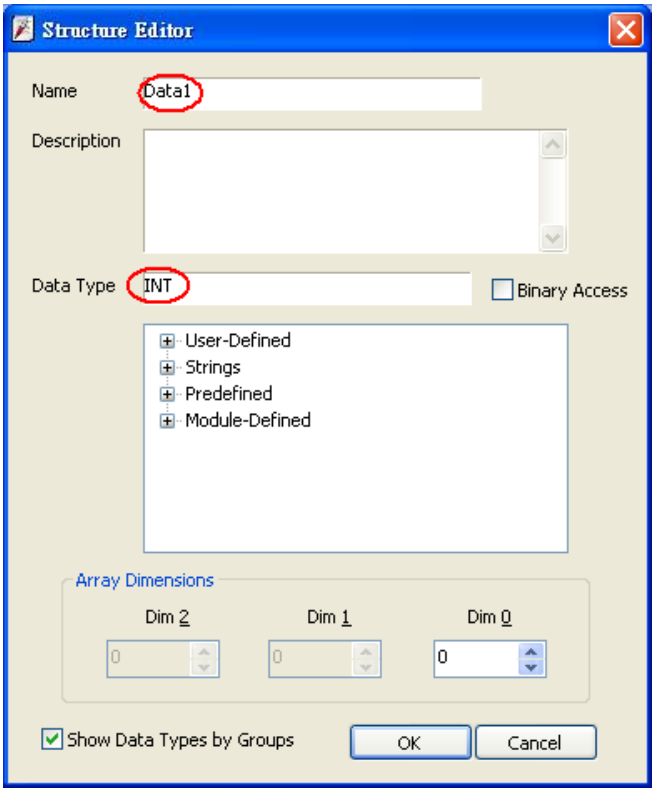


Шаг 2. Введите [Name] и тип данных. [Description] можно опустить.

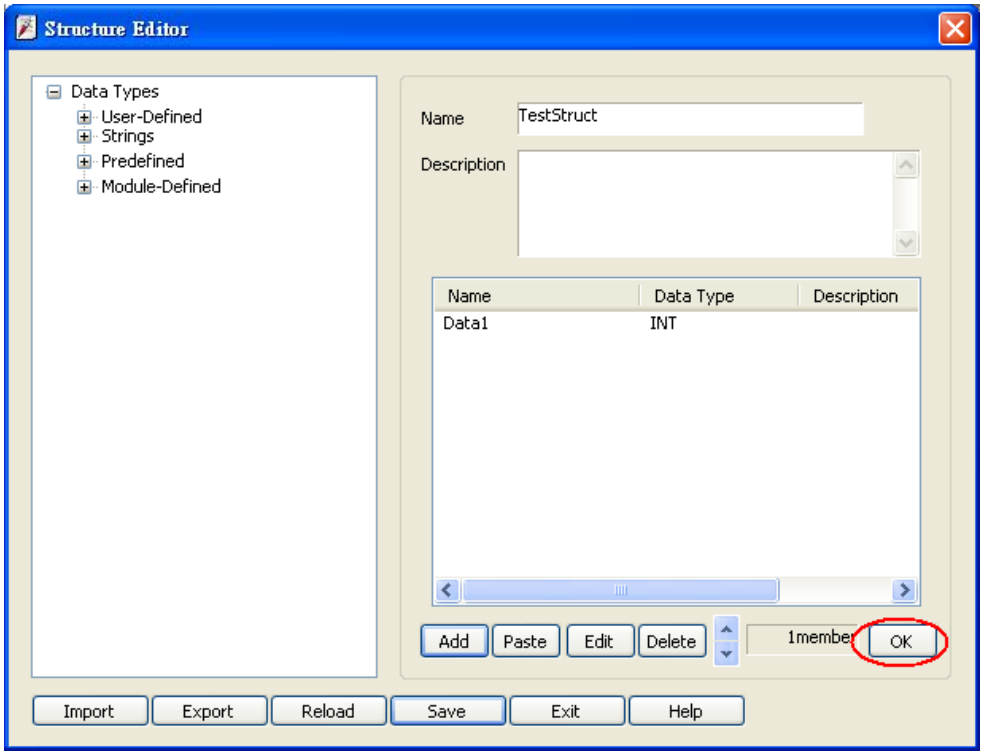
Для добавления элемента данных (data member), click [Add].



Шаг 3. Введите [Name] и [Data Type], затем, нажмите [OK] для выхода.



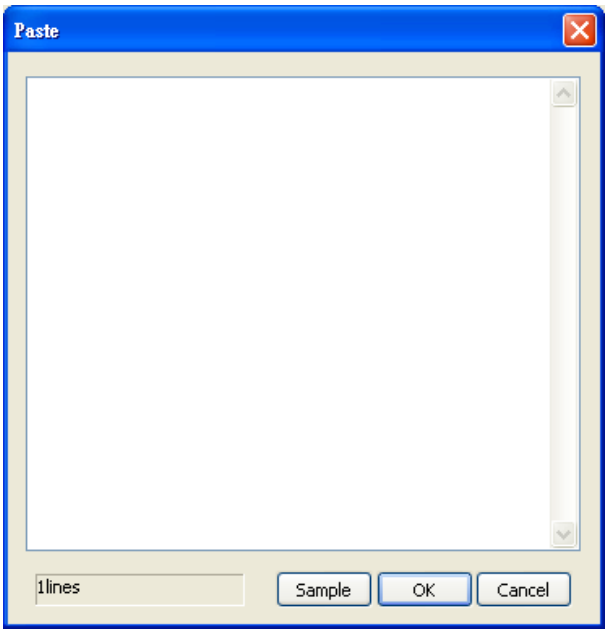
Шаг 4. После добавления всех элементов данных, нажмите [OK]. Созданные типы данных появятся в спике на левой стороне.



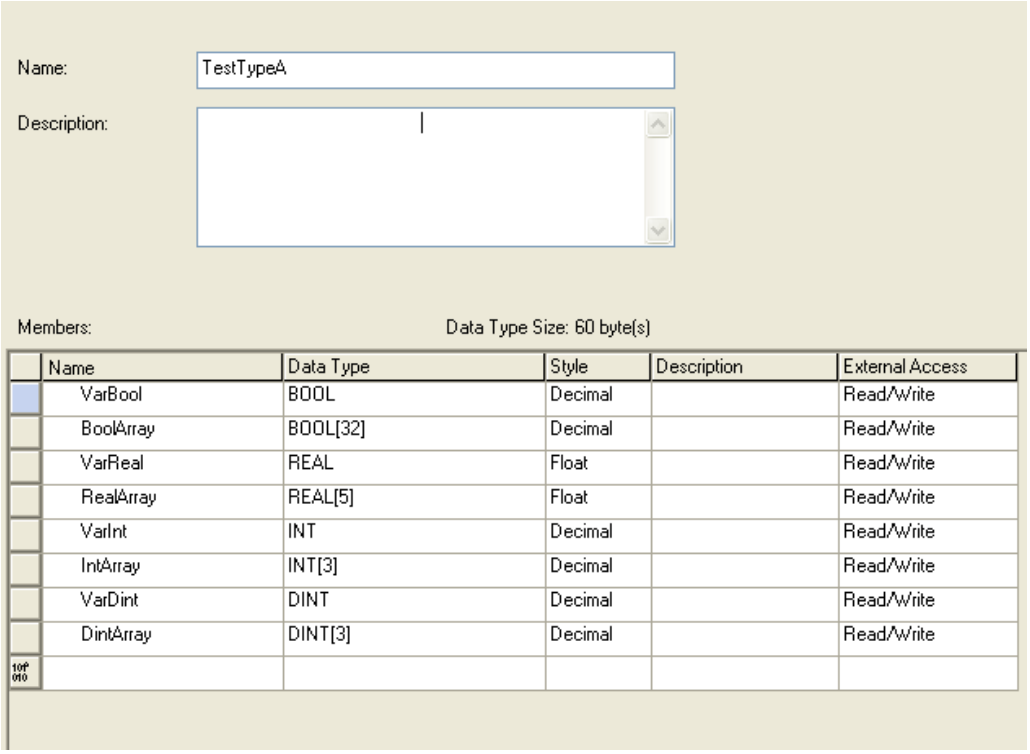
ПРИМЕЧАНИЕ: После изменения [Name] или [Description] типа данных, необходимо нажать [OK], чтобы активировать изменения.

34.3 Вставка данных

Шаг 1. При добавлении новых элементов данных (data members), эта функция позволяет добавлять несколько данных одновременно. Щелкните по [Paste] в основном окне.

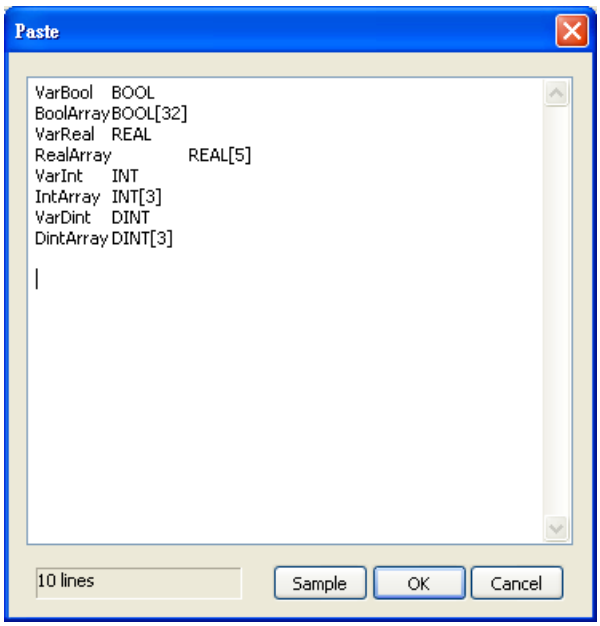


Шаг 2. Для редактирования, сначала, введите имя данных в каждой строке, затем используйте клавиши «space» или «tab», чтобы оставить пробелы в каждой строке. И, затем, введите тип данных или щелкните по [Sample], чтобы увидеть некоторые ссылки. Рекомендуется делать прямое копирование из RSLogix5000 во избежание ошибок.

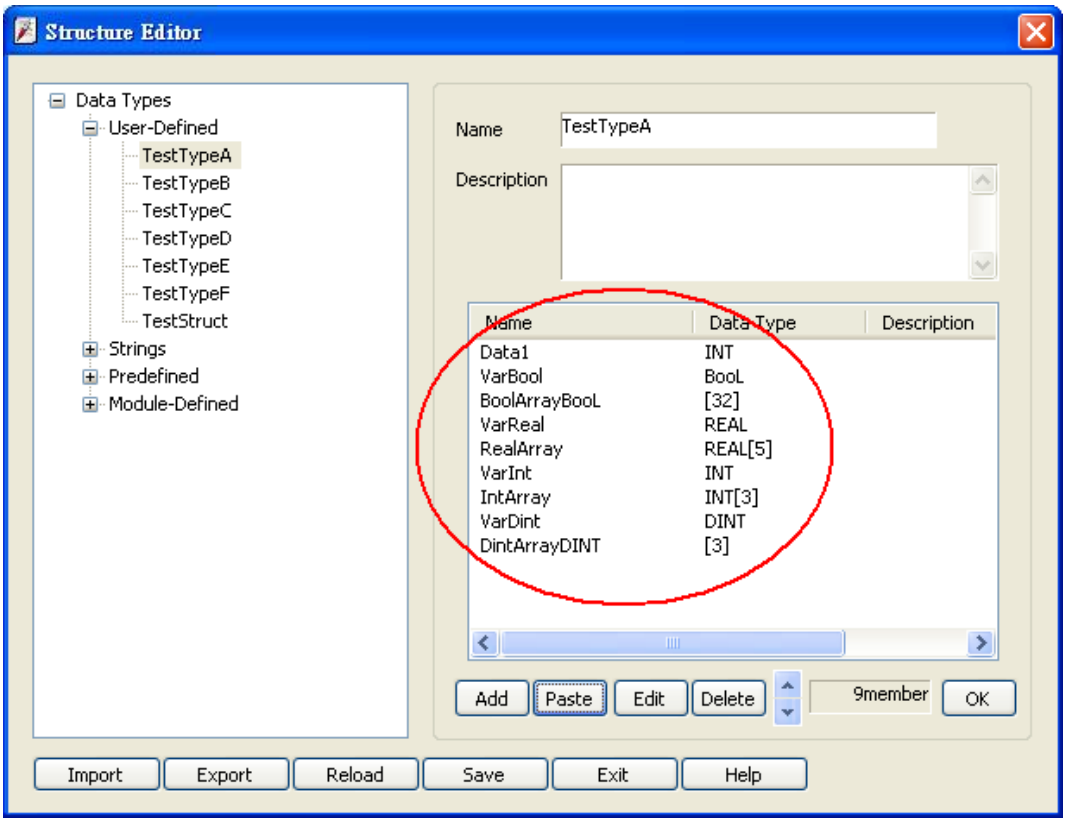


Name	Data Type	Style	Description	External Access
VarBool	BOOL	Decimal		Read/Write
BoolArray	BOOL[32]	Decimal		Read/Write
VarReal	REAL	Float		Read/Write
RealArray	REAL[5]	Float		Read/Write
VarInt	INT	Decimal		Read/Write
IntArray	INT[3]	Decimal		Read/Write
VarDint	DINT	Decimal		Read/Write
DintArray	DINT[3]	Decimal		Read/Write

Шаг 3. В таблице расположенной выше показаны типы данных определенные в RSLogix. Выберите [Name] и [Data Type] мышкой. Это можно сделать нажатием и удерживанием на первом элементе и перемещении вниз с прокруткой до конца и затем прекращения удержания. Все элементы будут выбраны. Нажмите «ctrl+v», чтобы скопировать и затем вставить выбранные в окно редактирования.



Шаг 4. Потом нажмите [OK], для завершения операцию и вернуться в основное окно, чтобы увидеть добавленные данные.



34.4 Прочие функции

● Коррекция элемента данных - *Revising member data*:

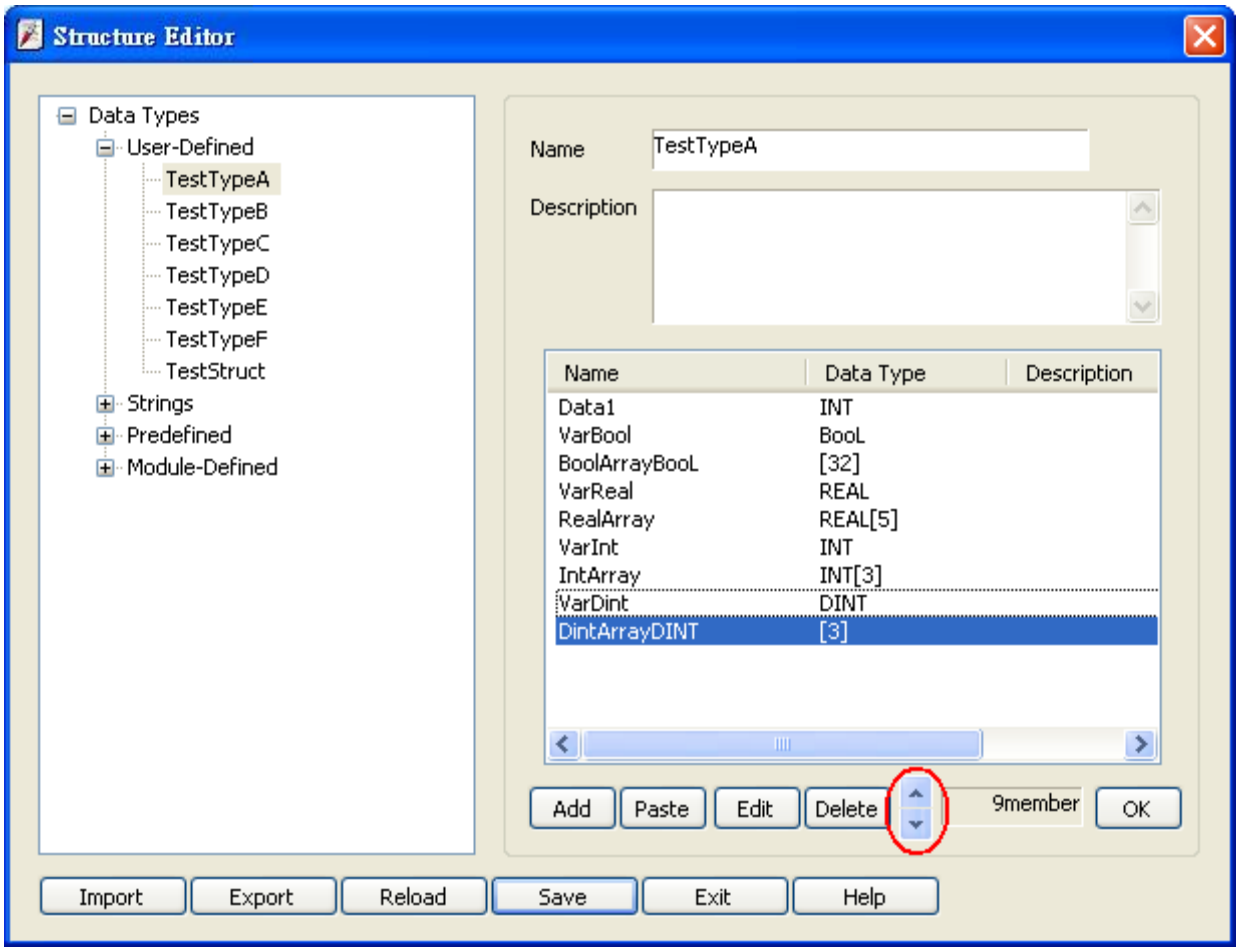
Сделайте двойной щелчок непосредственно по элементу данных, чтобы корректировать его в основном окне, или щелкните по элементу данных и нажмите [Edit].

● Удаление элемента данных (*data member*):

Выберите данные для удаления и нажмите [Delete]. Дл удаления всех элементов данных, нажмите и удерживайте клавишу [Delete] на клавиатуре и нажмите кнопку [Delete] в основном окне редактора.

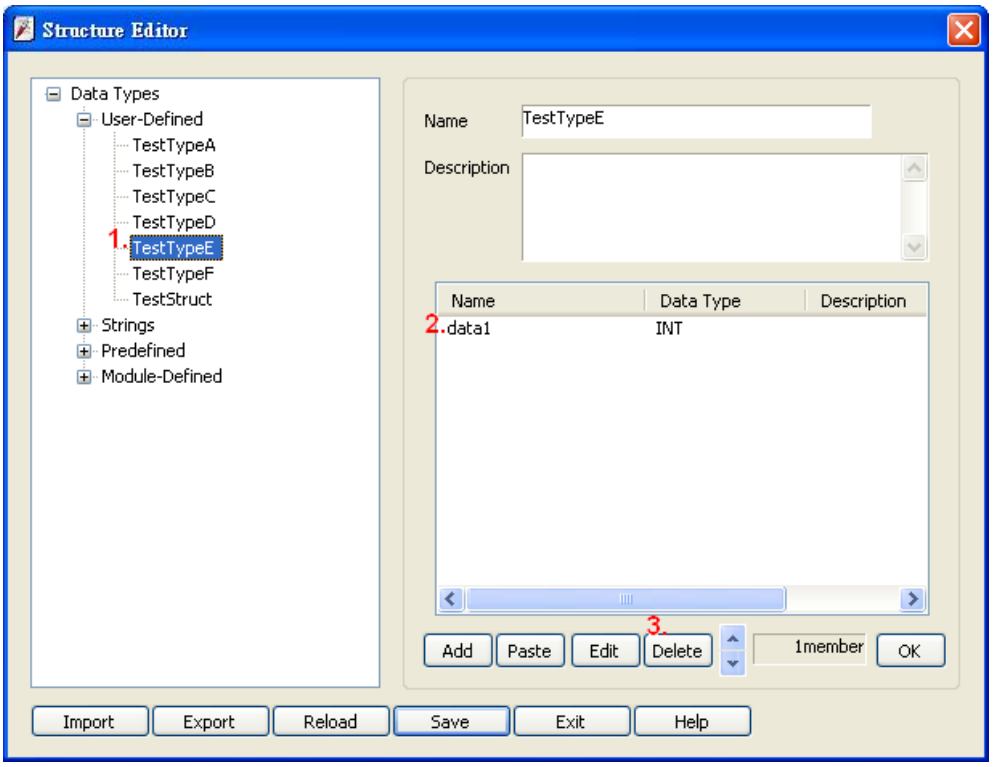
● Настройка порядка элементов данных - *Adjusting the order of data members*:

После выбора оного элемента данных, используйте кнопки перемещения «up» и «down» в основном окне для изменения порядка размещения элементов. Это позволит удобнее работать с выделенными данными в EasyBuilder Pro.



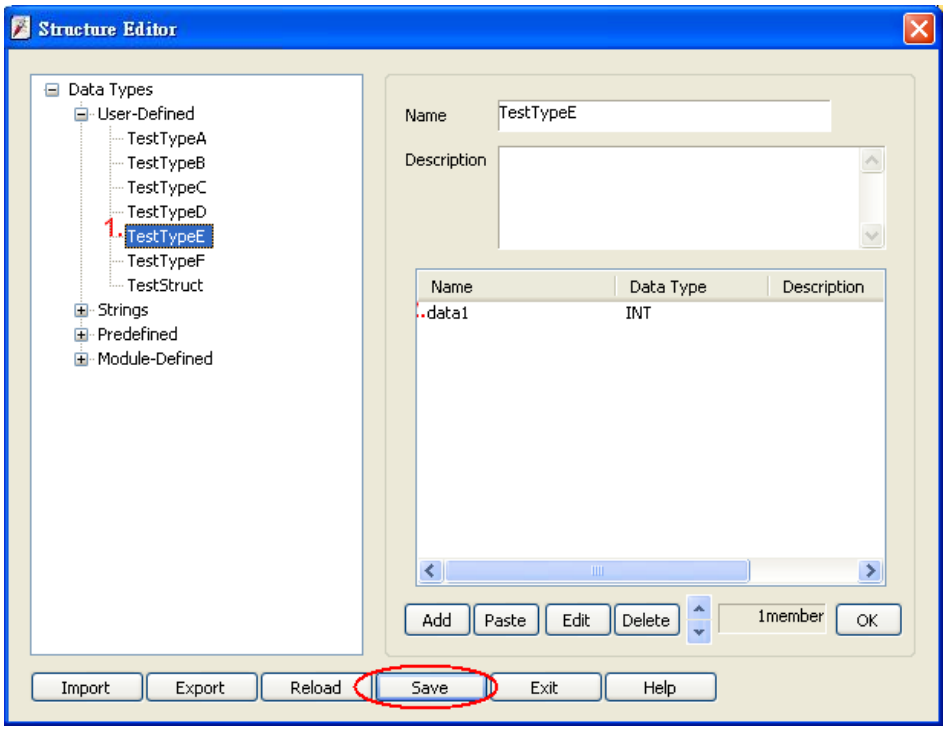
- Удаление типа данных:

Выберите из списка на левой стороне основного окна, затем выберите тип данных для удаления на правой стороне и нажмите [Delete] на клавиатуре. Тип данных будет удален.



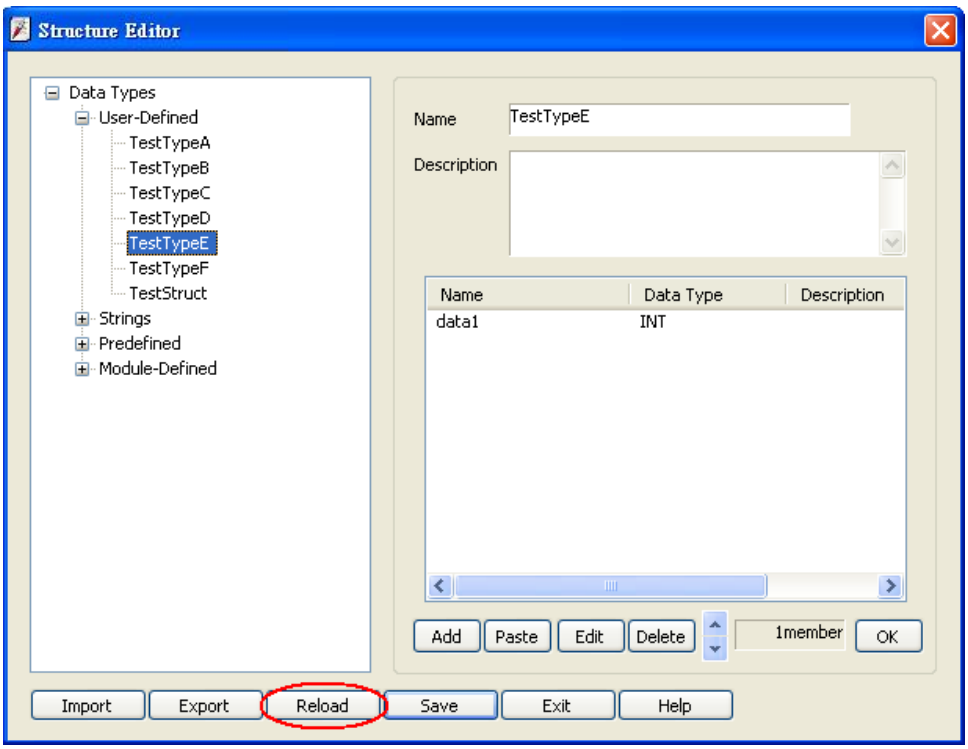
- Сохранение результатов исправлений:

После коррекции, нажмите кнопку [Save] на основном окне. Рестартуйте EasyBuilder Pro и увидите результаты изменений.



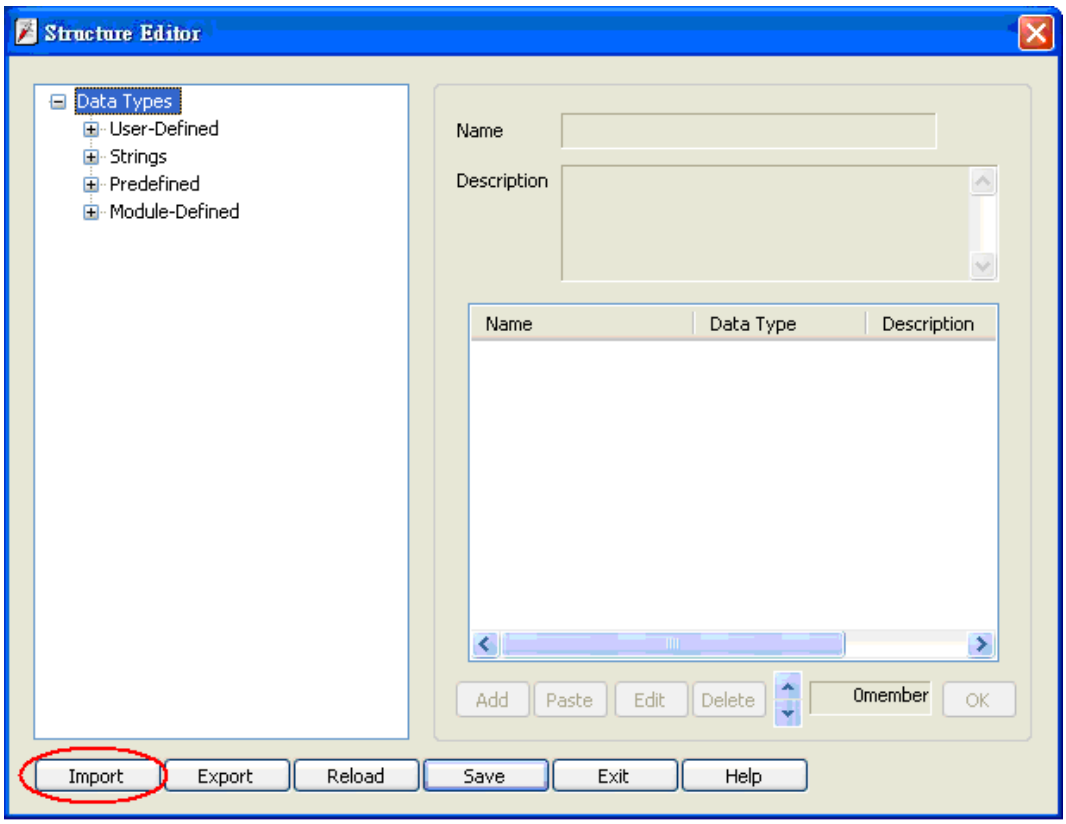
- **Re-edit:**

Для отказа от всех изменений и повторного редактирования (re-edit), нажмите кнопку [Reload] в основном окне.



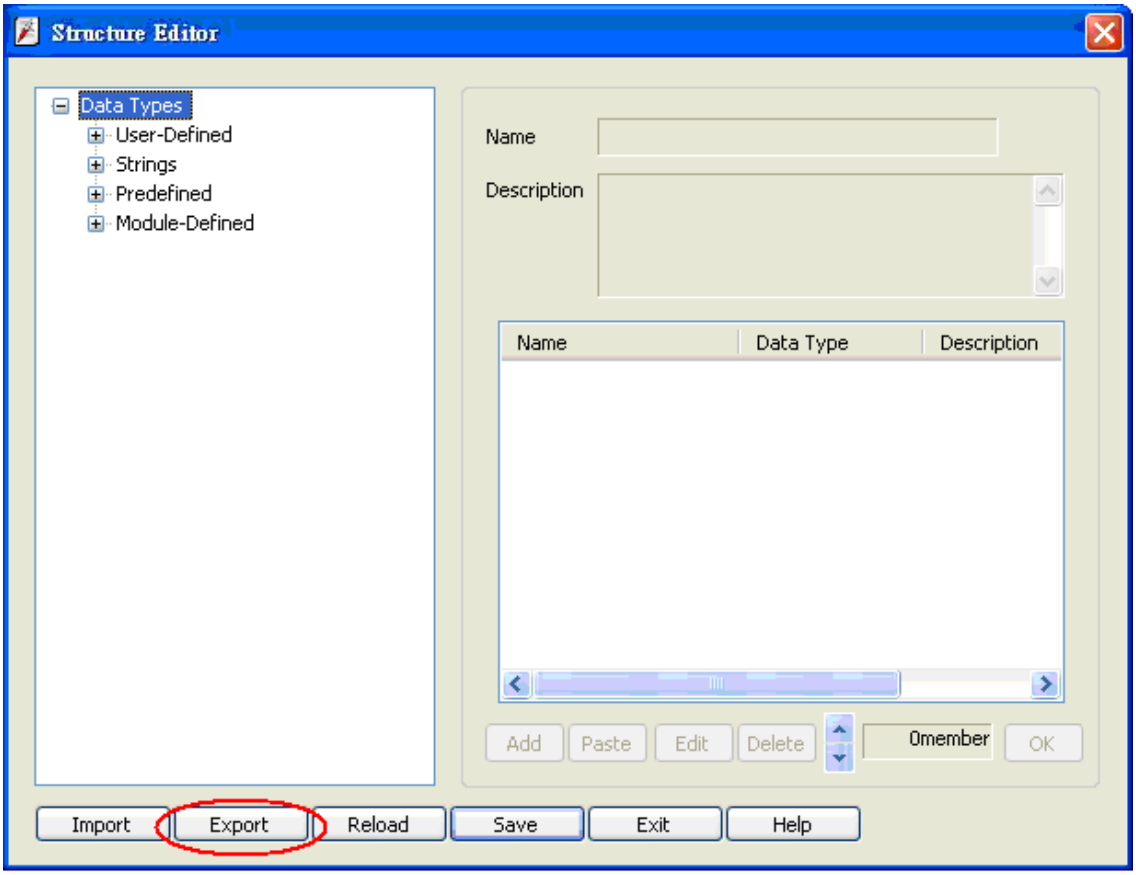
- **Import:**

Импорт предназначен для открытия файлов «TDF».



- **Export:**

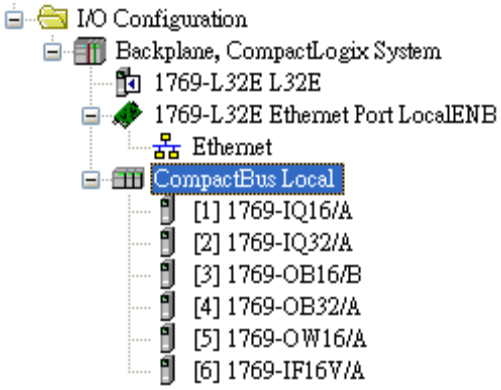
Экспорт редактированных данных в файл «XXX. TDF». Экспортированный файл TDF может быть использован в компьютере или для резервирования.



34.5 Тэги привязанные к модулю - Module-Defined

Далее приведен пример определения структуры модуля по умолчанию для.

В **I/O Configuration** пакета RSLogix5000 содержатся настройки модулей Ввода/вывода:

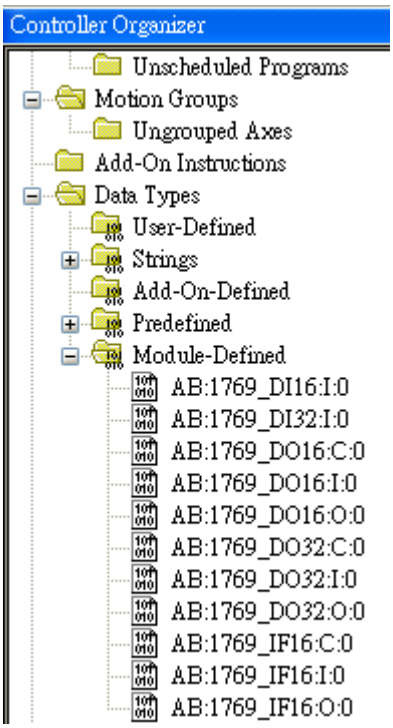


Тэгов этих моделей не будет в списке структур при экспортировании файла «CSV». Поэтому, пользователь должен создать их сначала.

	A	B	C	D	E	F	G	H
7	TYPE	SCOPE	NAME	DESCRIPT	DATATYPE	SPECIFIER	ATTRIBUTES	
8	TAG		Local:1:I		AB:1769_DI16:I:0			
9	TAG		Local:2:I		AB:1769_DI32:I:0			
10	TAG		Local:3:C		AB:1769_DO16:C:0			
11	TAG		Local:3:I		AB:1769_DO16:I:0			
12	TAG		Local:3:O		AB:1769_DO16:O:0			
13	TAG		Local:4:C		AB:1769_DO32:C:0			
14	TAG		Local:4:I		AB:1769_DO32:I:0			
15	TAG		Local:4:O		AB:1769_DO32:O:0			
16	TAG		Local:5:C		AB:1769_DO16:C:0			
17	TAG		Local:5:I		AB:1769_DO16:I:0			
18	TAG		Local:5:O		AB:1769_DO16:O:0			
19	TAG		Local:6:C		AB:1769_IF16:C:0			
20	TAG		Local:6:I		AB:1769_IF16:I:0			
21	TAG		Local:6:O		AB:1769_IF16:O:0			

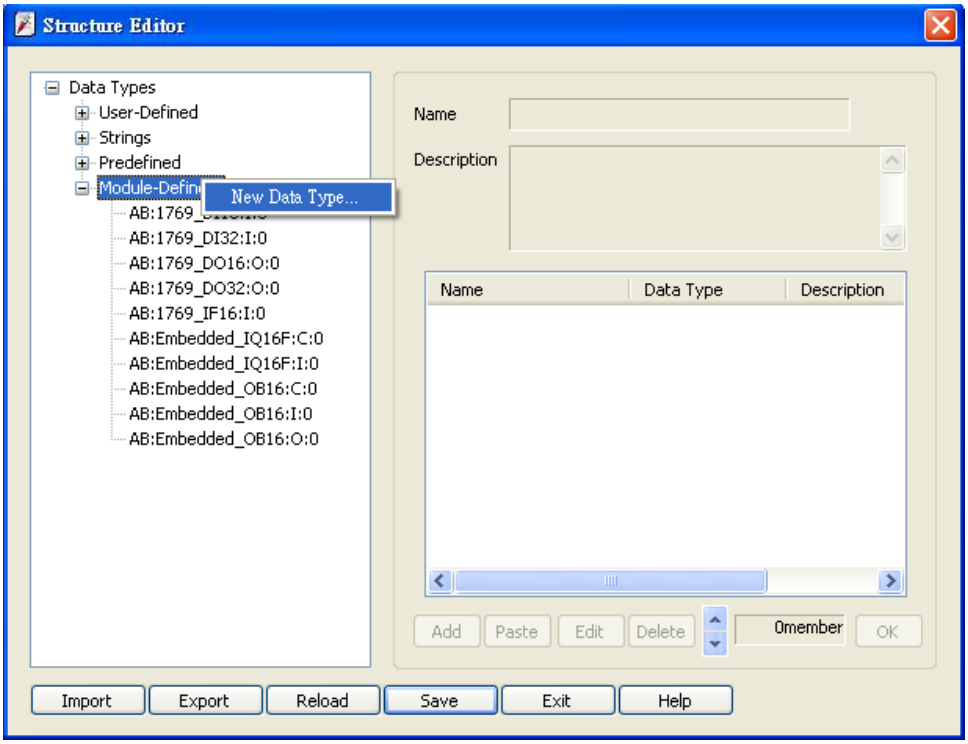
①

В [Controller Organizer/Data Types/Module-Defined] RSLogix5000, сделайте двойной щелчок по *Data Type* модуля. Элементы данных (*Data members*) этого типа модуля появятся в списке во всплывающем окне. Копируйте [Name] и [Data Type] Элементов (*Members*).



②

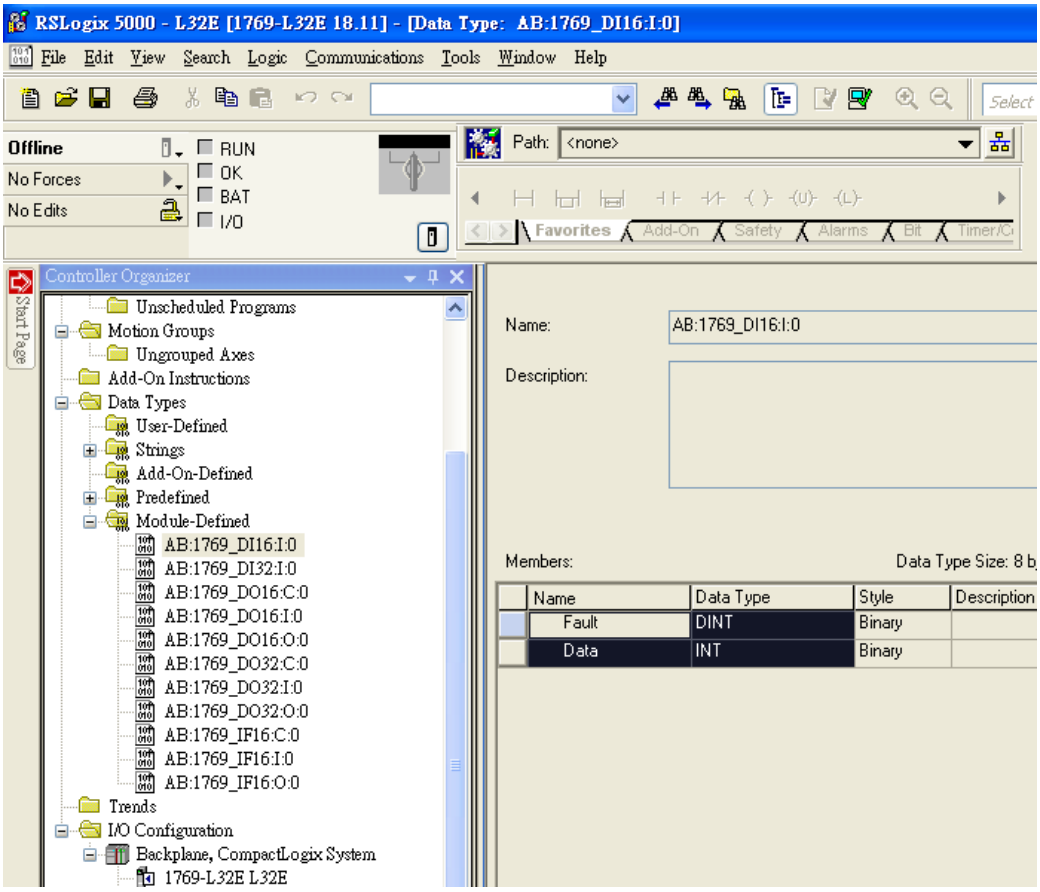
В редакторе (Structure Editor. exe) EasyBuilder Pro, щелкните правой кнопкой по [Module-Defined], и, затем, щелкните по [New Data Type…].

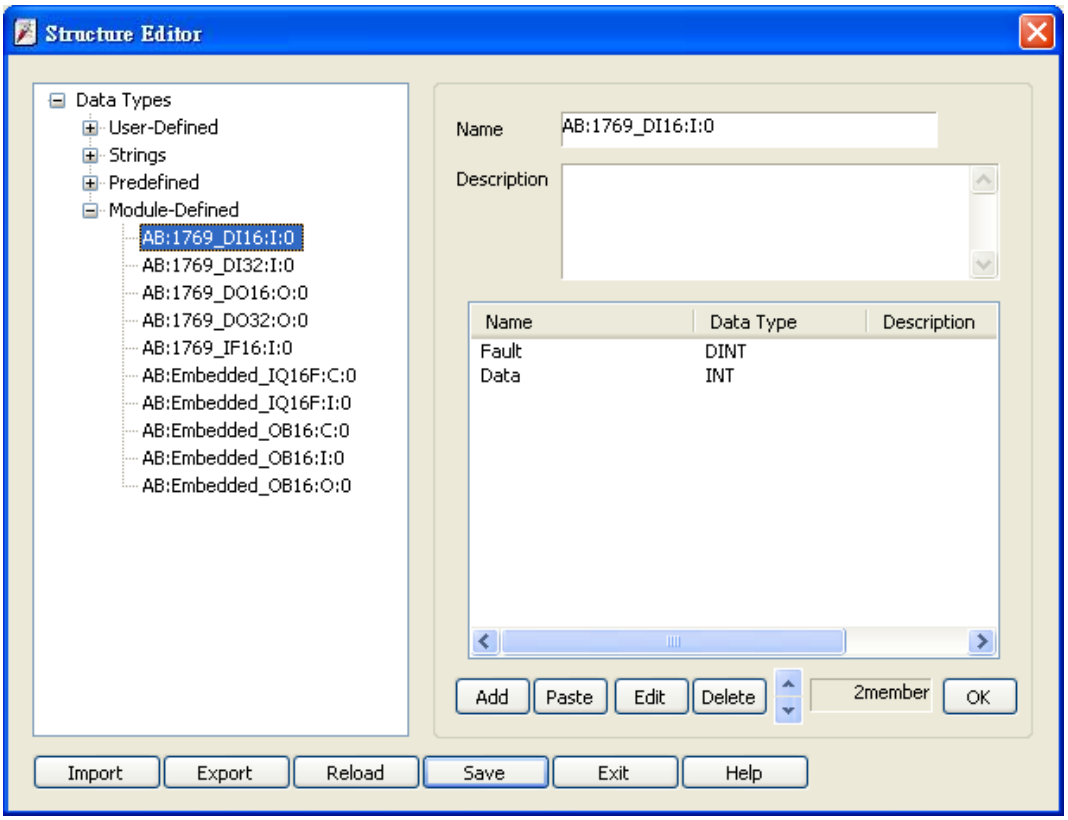


В колонке [Name] типа данных [New Data Type], введите *Module-Defined Name*.

③

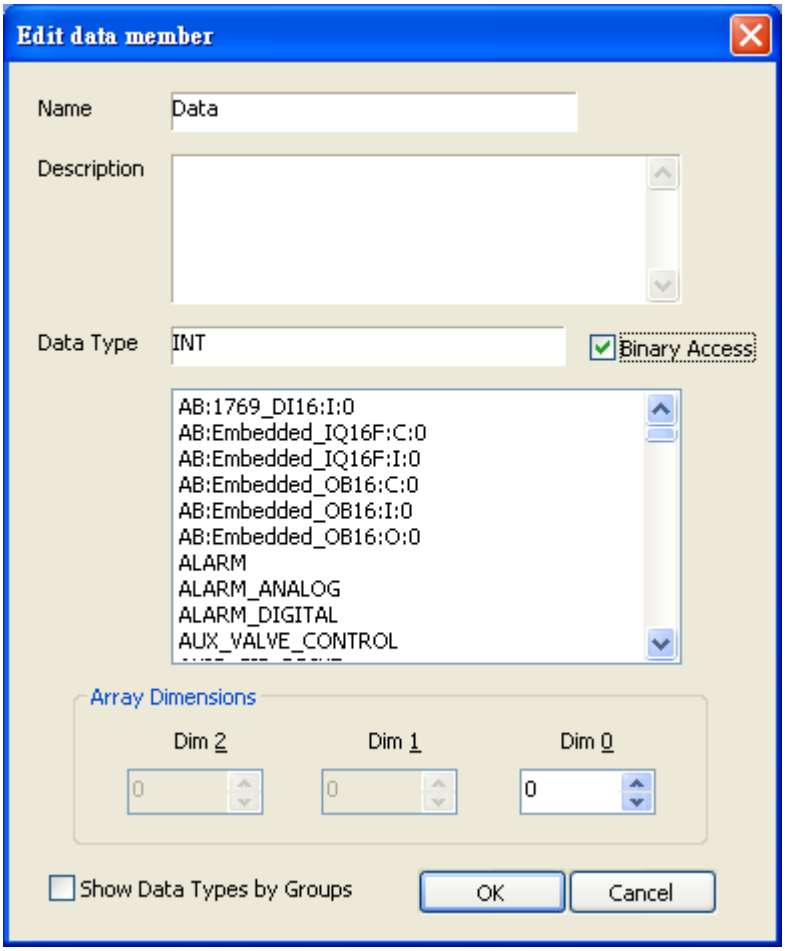
Щелкните [Paste], в диалоговом окне нажмите «Ctrl+V», чтобы вставить *Name* и *Data Type*.





④

Выберите данные и нажмите [Edit], так как данные модуля могут управляться битами, должен быть выбран [Binary Access] , затем, нажать [OK], чтобы вернуться в *Structure Editor*.



Нажмите [OK] для завершения настройки.

Глава 35 Монитор - Easy Watch

35.1 Введение

35.1.1 Что такое Easy Watch?

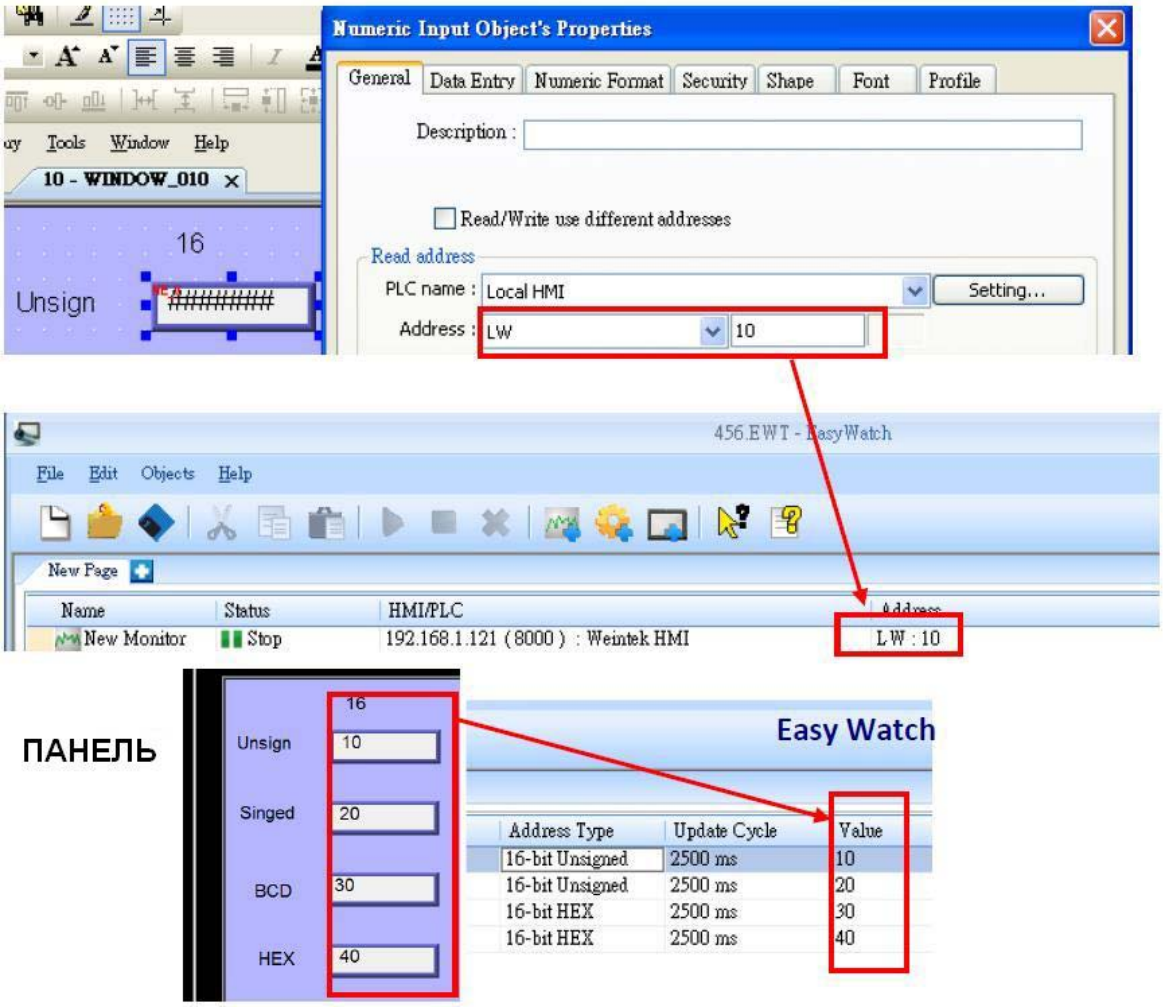
Easy Watch позволяет наблюдать и задавать значения адресов панели или ПЛК, и в тоже самое время вызывать *Macro* для облегчения отладки, удаленного наблюдения и управления. В этой главе описаны основные операции монитора, настройки монитора, настройки макросов и управление панелью, чтобы ознакомить пользователя с функциями Easy Watch.

35.1.2 Зачем нужен Easy Watch?

При создании нового проекта в EasyBuilder Pro, можно проверять точность заданных значений и текущих данных при помощи Easy Watch.

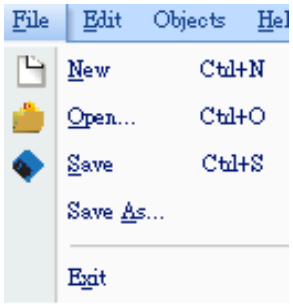
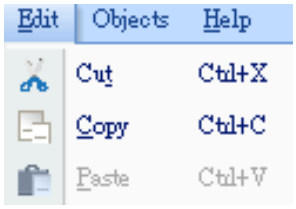
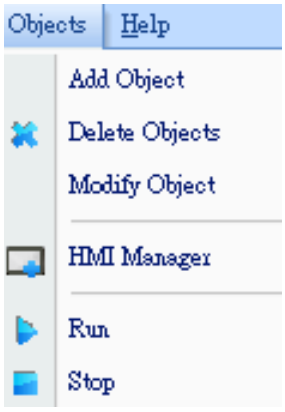
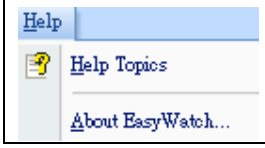
В EasyBuilder Pro добавить объект *Numeric Input*, адрес: LW10 и, затем, сделать то же самое в Easy Watch.

Если [Status] показывает соединение, и значение [Value] корректное, значит соединение работает и позволяет осуществлять мониторинг. Easy Watch будет отражать те же самые значения, что и панель, если настройки корректные.



35.2 Основные функции

35.2.1 Основные функции

Функция	Описание
File 	**New** Открыть новый файл Easy Watch
	Open Открыть существующий файл Easy Watch
	Save Сохранить настройки файла Easy Watch
	Save As Сохранить настройки файла Easy Watch в формате EWT
	Exit Выйти из Easy Watch
Edit 	**Cut** Вырезать выделенные объекты в буфер (clipboard)
	Copy Копировать выделенные объекты из буфера (clipboard)
	Paste Вставить содержимое буфера в выделенное место
Objects 	**Add Object** Добавить новый объект *Monitor* или *Macro*
	Delete Objects Выделить объекты для удаления, всплывет окно, нажмите “Yes”, чтобы удалить.
	Modify Object Изменение настроек выделенного объекта.
	HMI Manager Добавить, изменить, или переместить настройки панели (HMI).
	Run Выполнить действие выделенного объекта
	Stop Остановить выполнение действия выделенного объекта.
Help 	**Help Topics** Объяснение действий основных функций
	About Easy Watch Версия Easy Watch

35.2.2 Быстрый выбор инструментов

New: Открыть новый файл Easy Watch .

Open: Открыть существующий файл Easy Watch .

Save: Сохранить настройки файла Easy Watch .

Cut: Вырезать выделенные объекты в буфер (clipboard) .

Copy: Копировать выделенные объекты из буфера (clipboard).

Paste: Вставить содержимое буфера в выделенное место.

Run: Выполнить действие выделенного объекта .

Stop: Остановить выполнение действия выделенного объекта.

Delete Objects: Выделить объекты для удаления.

Monitor: Добавить новый объект *Monitor.*

Macro: Добавить новый объект *Macro*.

HMI Manager: Добавить, изменить, или переместить настройки панели (HMI). .

Help: Ссылки на описания основных функций.

Help Topics: Объяснение действий основных функций

35.3 Настройки монитора

35.3.1 Добавить монитор

Существует два сособа добавления объекта Monitor:

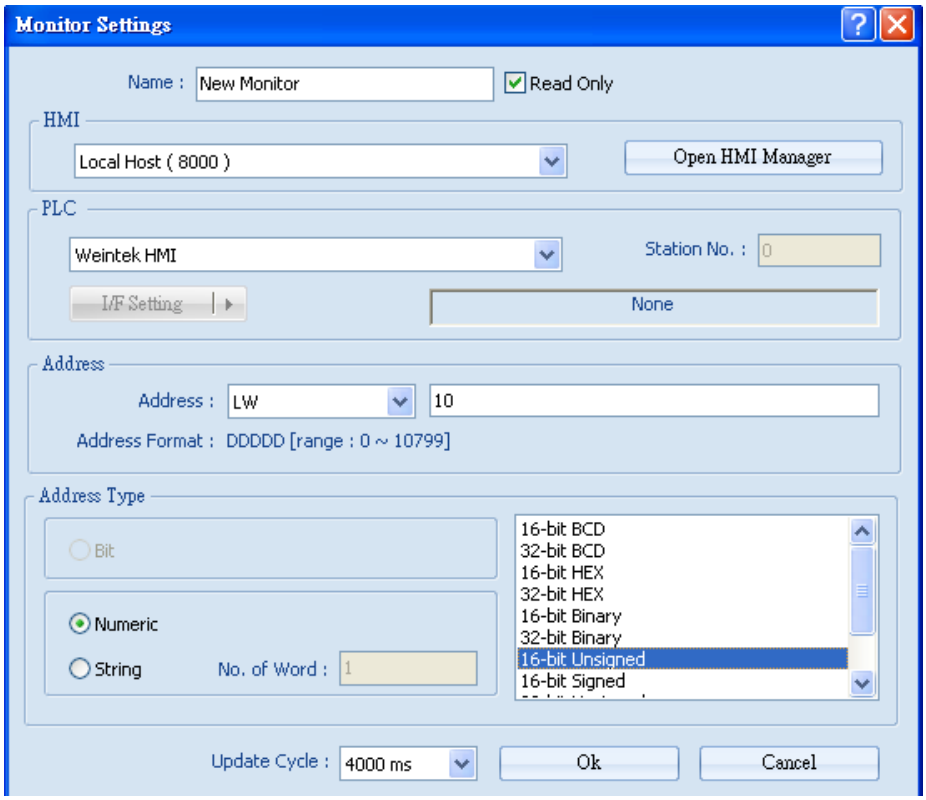
a. Выбрать из основной панели инструментов:

Objects->Add Object->Add Monitor

b. Выбрать на панели быстрого выбора инструментов:

Add Monitor

35.3.2 Настройки монитора



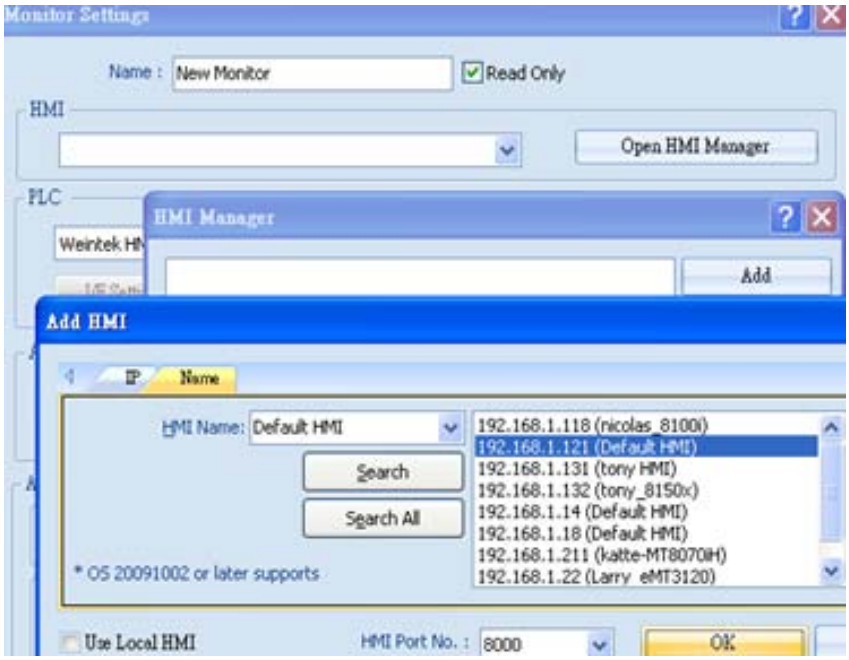
1. ***Object Name:*** Имя объекта, и оно не должно повторяться and the name can't repeat
2. ***Read Only:*** Если отметить это, значение адреса не может быть настроено.
3. ***Target HMI:*** Панель с адресами для наблюдения.
4. ***PLC Settings:*** Настройки типа, номера станции, пути соединения с ПЛК, чьи адреса должны контролироваться.
5. ***Address:*** Настройка адреса.
6. ***Address Type:*** Когда адрес выбран, будут показаны допустимые типы данных.
7. ***Update Cycle:*** Временной интервал обновления значений адресов. Если обрабатывается много объектов одновременно, могут возникать задержки и ошибки.

35.3.3 Добавить новое устройство (New Device)

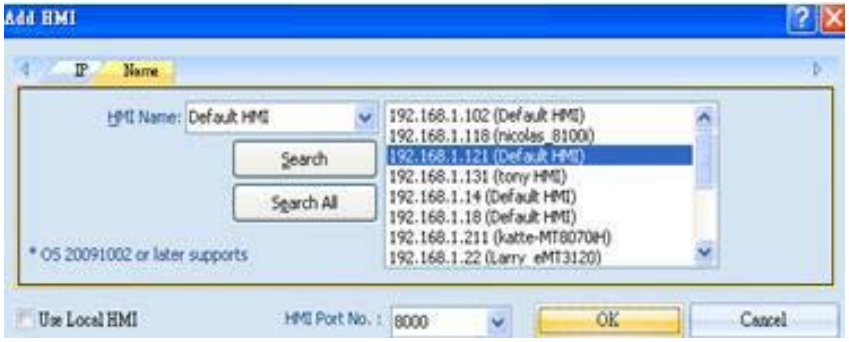
1. Откройте окно *Monitor Settings*, если панель не обозначена там, она может быть добавлена:

1-1 Щелкните по [Open HMI Manager]

1-2 Щелкните по [Add], чтобы найти все панели в сети.



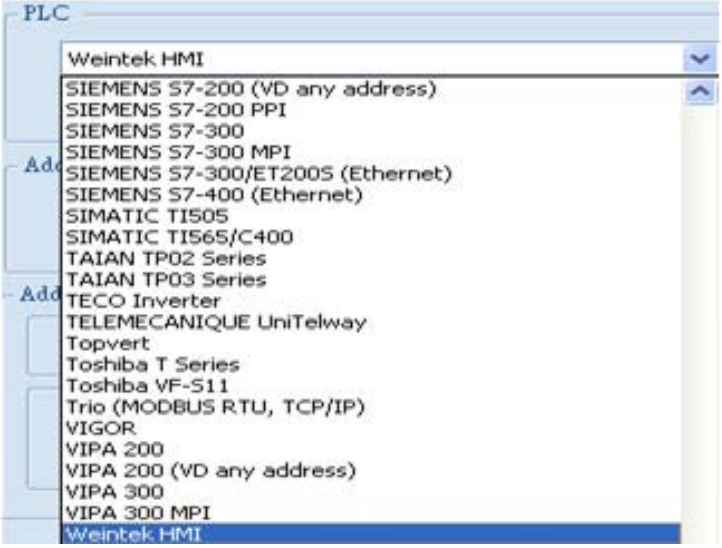
1-3 Выберите панель (HMI) и нажмите [OK], чтобы закончить добавление.



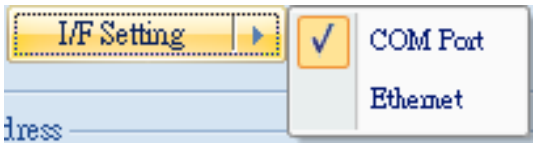
1-4 В режиме «off-line simulation» панель может быть добавлена, если отметить [Use Local HMI] – Использовать локальную панель.

2. В настройках контроллера «PLC» выберите тип ПЛК или целевую панель (HMI).

2-1 Выберите “Weintek HMI”, чтобы оперировать данными локальной панели (local HMI).



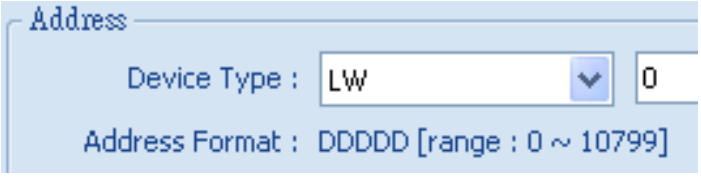
3. Чтобы наблюдать за ПЛК, необходимо выбрать интерфейс. Настройки интерфейса - «I/F Setting» могут быть настроены на [COM Port] или[Ethernet].



3-1 Отметьте [COM Port], щелкните по [I/F Setting], чтобы выбрать COM порт.

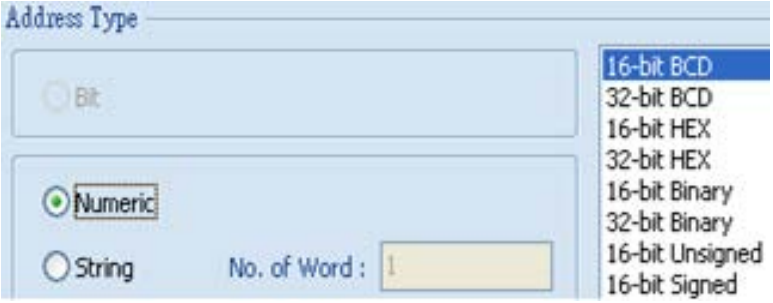
3-2 Отметьте [Ethernet], щелкните по [I/F Setting], чтобы назначить IP-адрес.

4. Настройка адресов данных ПЛК.



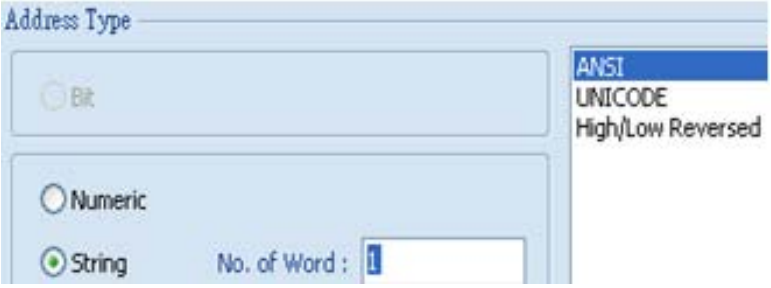
5. *Address Type* может быть задан [Numeric] или [String].

5-1 *Numeric:* выберите формат данных адресов для чтения.

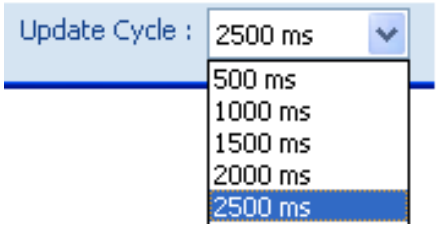


5-2 *String:* выберите формат данных [ANSI] или [UNICODE], и [High/Low Reversed] .

Назначьте число слов [No. of Word] для чтения.



6. Настройте интервал обновления - Update Cycle



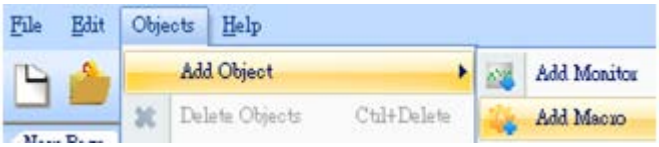
35.4 Настройки Macro

35.4.1 Добавить Macro

Существует два способа добавления объекта *Macro*.

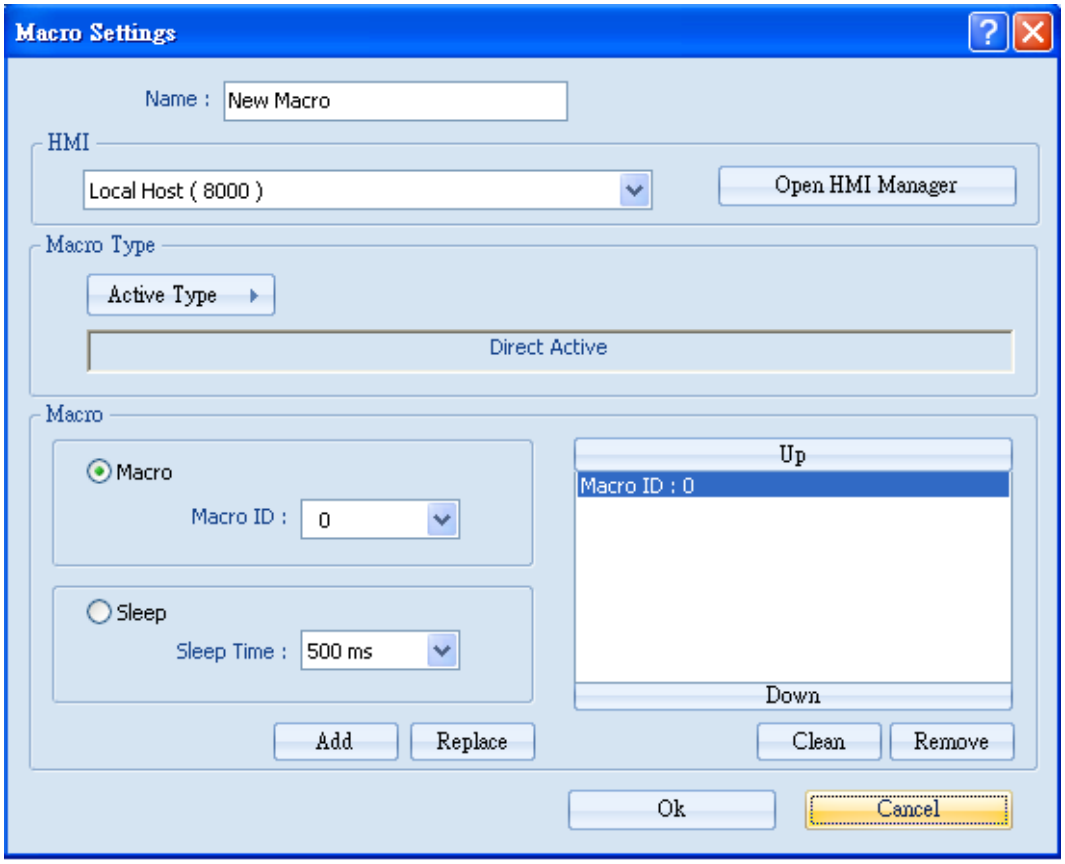
a. Выбрать на основной панели инструментов:

Objects->Add Object->Add Macro



b. Выбрать на панели быстрого выбора инструментов: *Add Macro*

35.4.2 Настройки Macro



1. ***Object Name:*** имя объекта и оно не должно повторяться.

2. **Target HMI:** Панель, в которой использован этот *Macro*.

3. ***Macro Active Type:*** тип активности *Direct Active* или *Cycle Active*

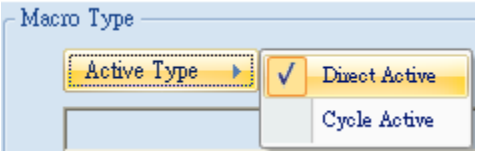
4. ***MACRO List Editing:*** Каждый объект *Macro* может исполнять несколько макросов. Может быть настроен интервал между двумя макросами.

35.4.3 Добавить новый Macros в список

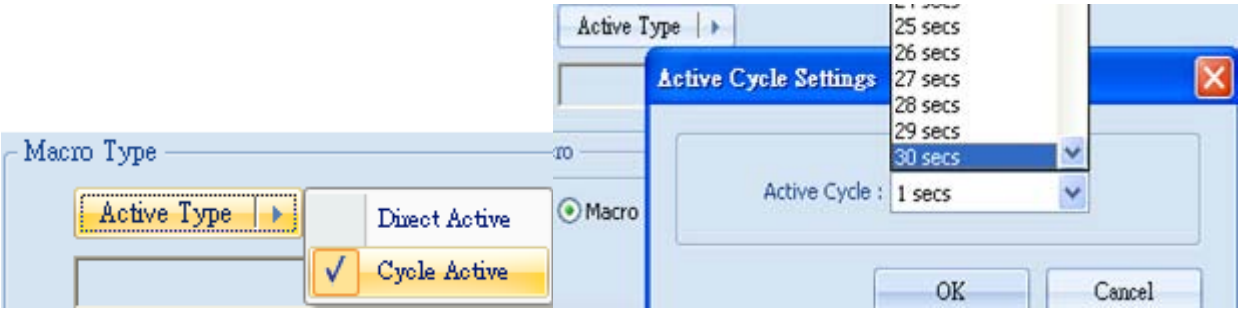
1. Как добавить новую панель, см. раздел “35.3.3 Добавить новое устройство”.

2. *Macro Active Type* может быть выбран [Direct Active] или [Cycle Active].

2-1 ***Direct Active:*** Macro исполняется однажды после нажатия кнопки [Active] в списке объекта.

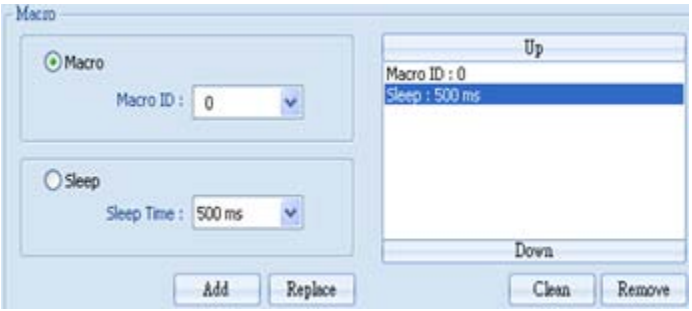


2-2 ***Cycle Active:*** Настройте интервал выполнения *Macros.* Если [Active Cycle] равно “5 Secs”, после выполнения всех макросов, они будут выполняться следующий раз через 5 секунд.

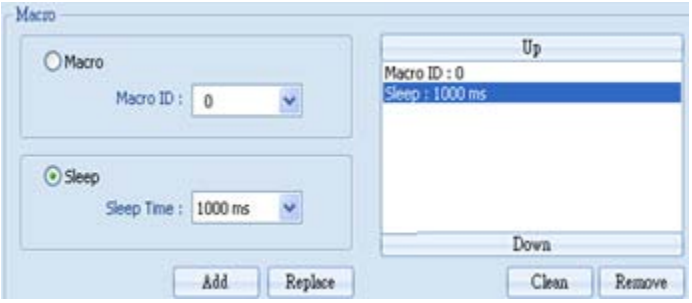


3. Настройки *Macro* включают [Macro ID] и [Sleep Time] – (Время бездействия). Назначьте «ID» *Macro* и временной интервал между каждым *Macro*. Нажмите [Add] или [Replace], чтобы добавить или заменить *Macros* в списке.

3-1 Назначьте *Macro ID*, нажмите [Add], чтобы добавить его в список.



3-2 Установка *Sleep Time:* выберите *Sleep* в списке и нажмите [Replace], чтобы заменить выбранное *Sleep time*.



35.5 Управление панелью - HMI Manager

35.5.1 Настройки панели - HMI Settings

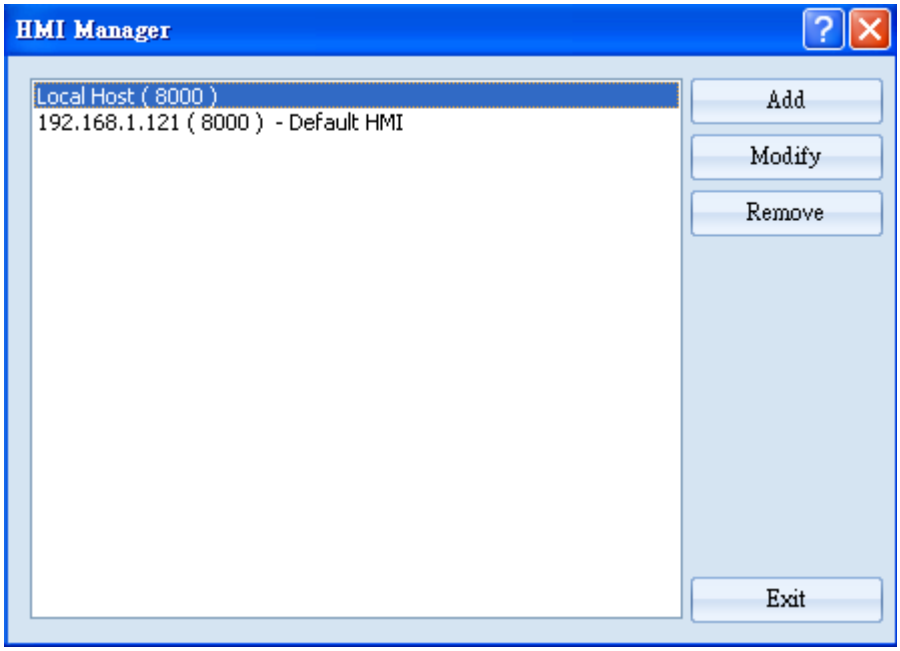
Существует два способа открытия окна *«HMI Settings»:*

a. Выбрать на основной панели инструментов:

Objects->HMI Manager

b. Выбрать на панели быстроговыбора:
HMI Manager

35.5.2 HMI Manager



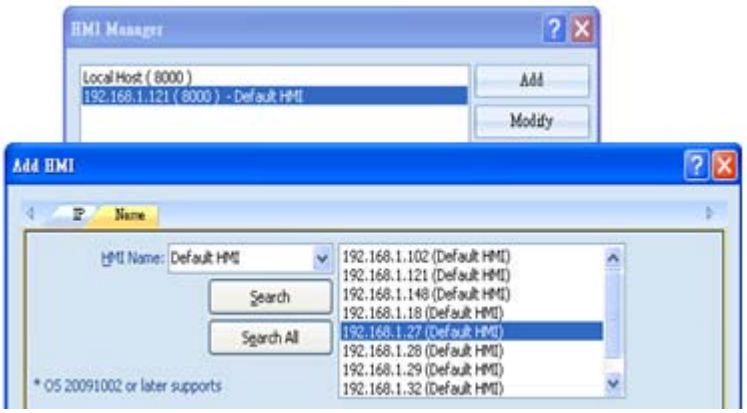
EasyWatch позволят наблюдать адреса нескольких панелей.

35.5.3 Добавить новое устройство

1. HMI Manager может [Add], [Modify] или [Remove] панель (HMI).

1-1 *Add:* Как добавить новую панель см. раздел 35.3.3.

1-2 *Modify:* Выбрать панель для редактирования.

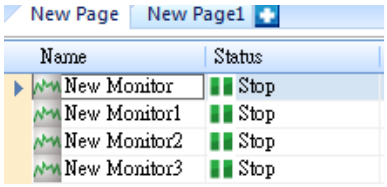


1-3 *Remove:* Выбрать панель для перемещения и подтверждение удаления.

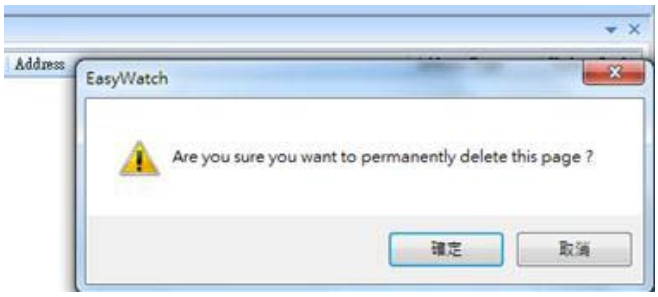
35.6 Список объектов - Object List

35.6.1 Настройка страницы

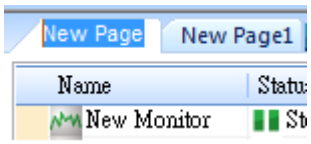
1-1 Добавление новой страницы: Щелкните по пиктограмме “+” .



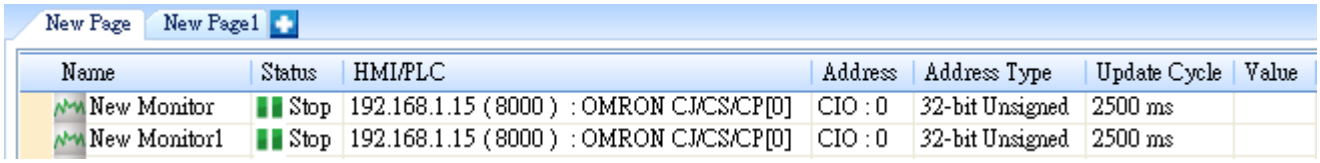
1-2 Удаление страницы: Щелкните по пиктограмме “X” и подтвердите удаление.



1-3 Переименование страницы: Сделайте двойной щелчок по Имени страницы и напечатайте новое имя



35.6.2 Колонки списка объектов



Name	Status	HMI/PLC	Address	Address Type	Update Cycle	Value
New Monitor	Stop	192.168.1.15 (8000) : OMRON CJ/CS/CP[0]	CIO : 0	32-bit Unsigned	2500 ms	
New Monitor1	Stop	192.168.1.15 (8000) : OMRON CJ/CS/CP[0]	CIO : 0	32-bit Unsigned	2500 ms	

1. ***Name:*** Отображает имена объектов, небольшая пиктограмма рядом с именем показывает тип объекта.

2. ***Status:*** Отображает состояние объектов: Connecting, Connected, или Stop (Соединяется, Соединен, Остановлен). Если панель не соединена или номер порта не корректный появляется сообщение “HMI Not Found” – Панель не найдена. Для объектов *Monitor* , если адрес не корректный, появляется сообщение “Address Error”.

3. ***HMI/PLC:*** Отображает информацию о «HMI/PLC», которые сейчас работают с объектами.

4. ***Address / Address Type:*** Для объектов *Monitor* , показаваются допустимые настройки адресов.

5. ***Update Cycle:*** Временной интервал обновления значений адресов.

6. ***Value:*** Для объектов *Monitor* , если состояние “Connected”, будут отображаться значения адресов панели. Если этот объект *Monitor* не определен, как «read only», редактируя, можно изменять наблюдаемые адреса. Для объектов *Macro* , если выбрано «Direct Active», в этой колонке будет находится кнопка [Active] для выполнения *Macro*.

7. Перемещайте заголовки колонок в желаемые положения – «Drag and drop».

Глава 36 Инструменты Администратора

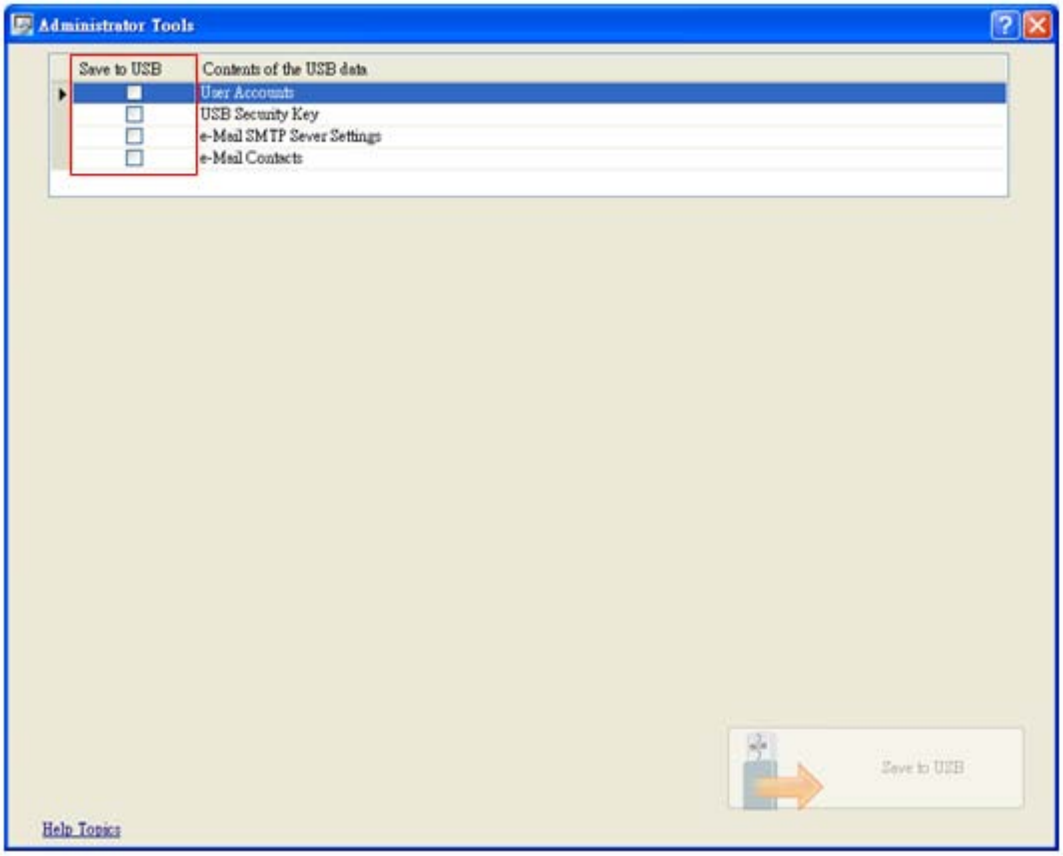
36.1 Введение:

Инструменты Администратора (Administrator Tools) позволяют сохранять Учетные записи пользователя (User Accounts), Ключи (USB Security Key), Настройки почтового сервера (E-mail SMTP Server) и Контакты (E-mail Contacts) на USB.

EasyBuilder Pro *User accounts,* функции *e-Mail* и созданные данные могут быть импортированы в панель, с использованием объекта **Function Key / Import user data / Use [USB Security Key].** Это позволяет удобно осуществлять переносим.

Совет по использованию:

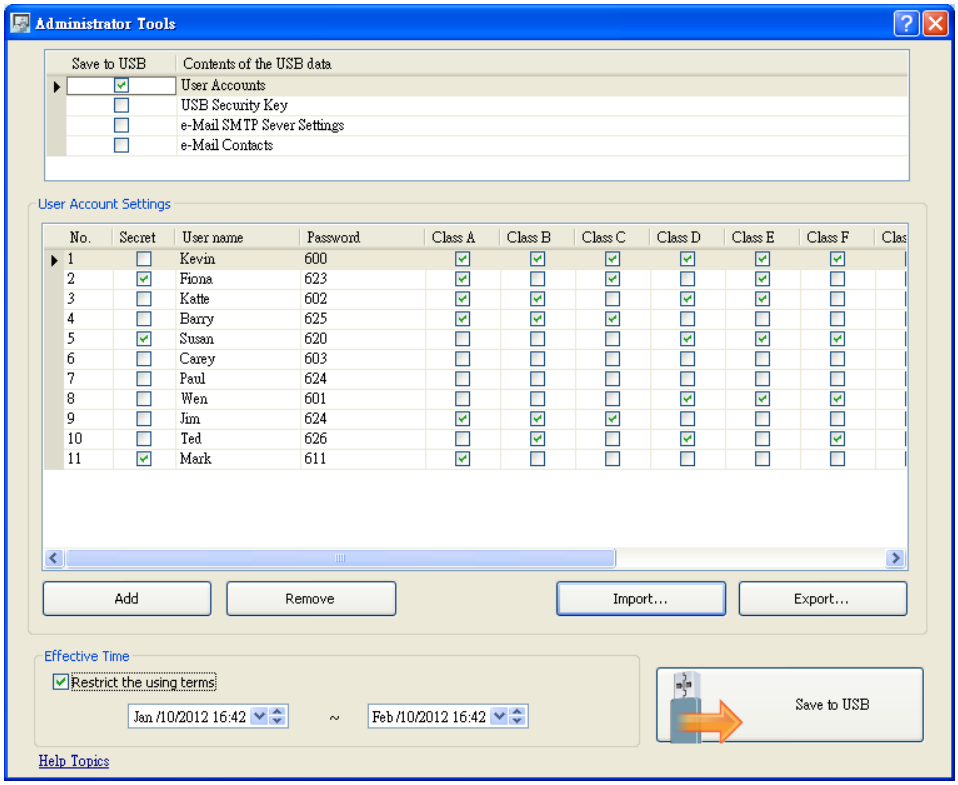
Запустите Administrator Tools, сделайте отметки в окошках [Save to USB], чтобы разрешить настройку выбранных функций.



36.2 Учетные записи пользователя - User Accounts

36.2.1 Введение в User Accounts

Сделайте отметку в окошке *User Accounts* , чтобы заполнить необходимые настройки, как показано ниже:



Настройки	Описания
Secret	Отметьте, чтобы создать секретные учетные записи пользователя
User Name	Ввод имени пользование *Примечание 1
Password	Ввод пароля *Примечание 1
Class A~L	Классы привилегий пользователя
Add	Добавление новой учетной записи *Примечание 2
Remove	Удалить существующую учетную запись пользователя
Import	Импорт данных учетной записи пользователя
Export	Экспорт данных учетной записи пользователя
Effective Time	Импорт данных в панель в течение указанного периода времени, импортируемые данные эффекивны постоянно. Если не указывать *Effective Time*, данные могут быть импортированы а любое время.
Save to USB	Save data to USB

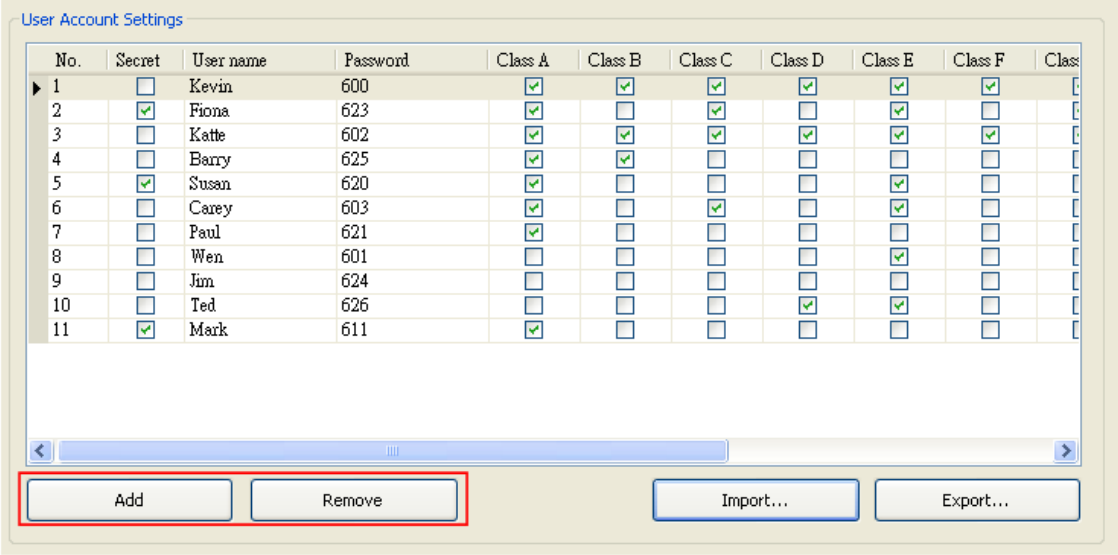
Note

Примечание 1: Может быть составлен из букв, цифр, ”-“, ”_”. Регистр учитывается.

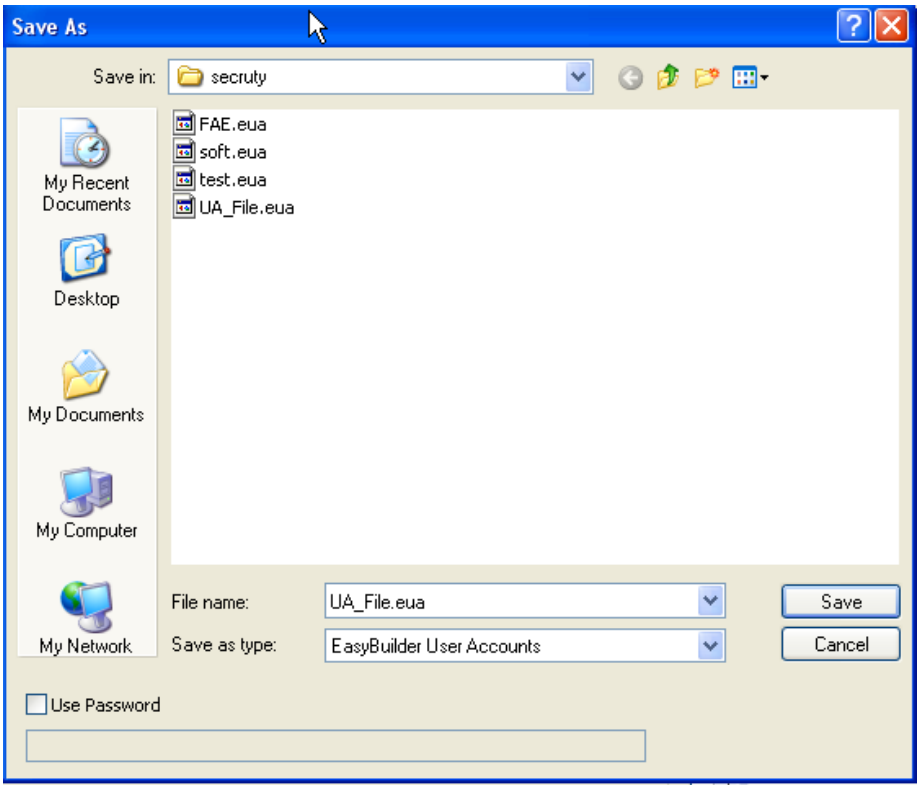
Примечание 2: Максимально 127 учетных записей пользователя может быть добавлено.

36.2.2 Настройка User Accounts

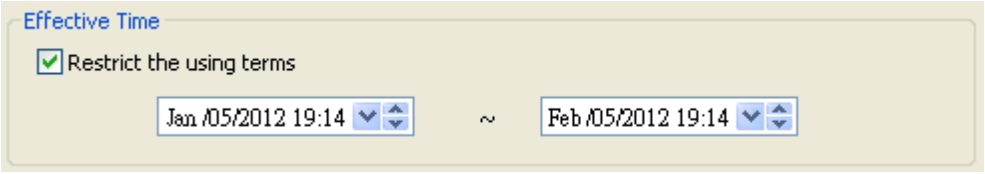
a、 Щелкните по **[Add]** чтобы создать новую Учетную запись. Щелкните по **[Remove]** , чтобы удалить выделенную Учетную запись. Щелкните по **[Secret],** чтобы определить *account* , как секрет пользователя. Напечатайте **[User name]** и **[Password]** и отметьте классы привилегий **[Class A] ~ [Class L]**.



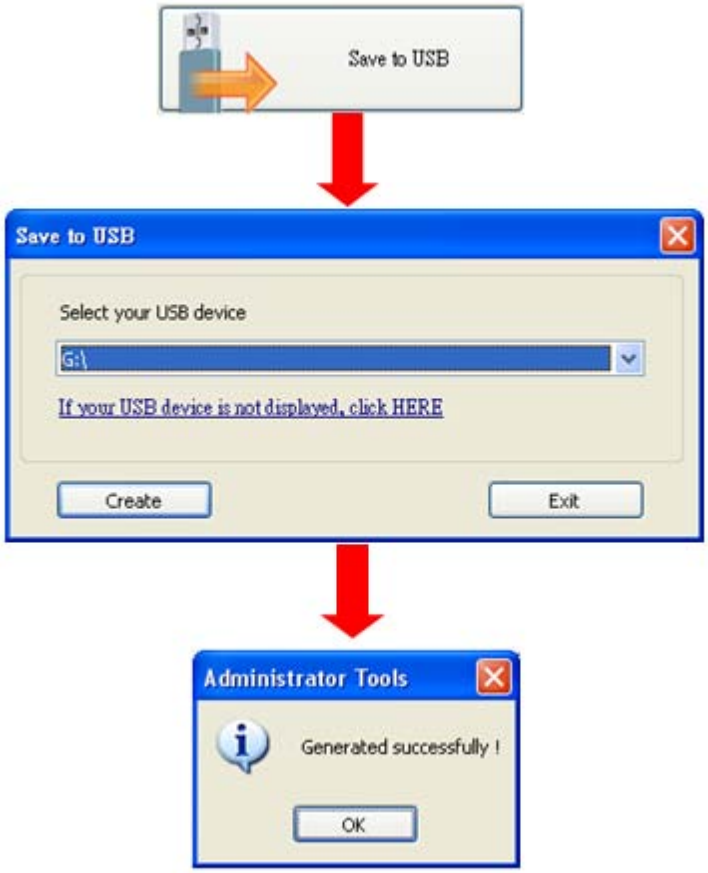
b、 После создания Учетной записи, щелкните по **[Export],** чтобы сделать резервную копию данных (backup data) . Для перестройки и изменения, щелкните по **[Import],** чтобы импортировать резервные данные.



c、 Если отмечено **[Effective Time]** -> **[Restrict the using terms]**, то только в течении указанного периода времени пользователь может импортировать *account data* в панель через USB. Если не отмечено, пользователь может импортировать *account data* в панель в любое время.



d、 После завершения настроек, щелкните по **[Save to USB]**, выберите USB и, затем щелкните по **[Create] - Создать**. Сообщение ”Generated successfully!” появится, нажмите **[OK]**.

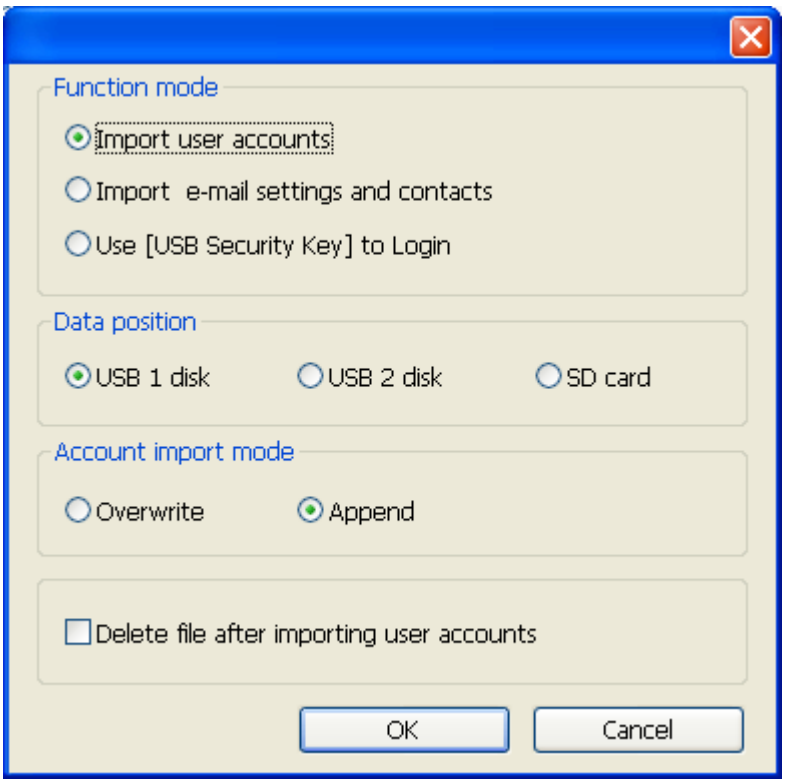


36.2.3 Импорт Учетных записей при помощи EasyBuilder Pro

Создайте объект *Function Key* в EasyBuilder Pro. При касании этого объекта на экране панели, будет выполняться импорт. Далее пример создания этого объекта *Function Key*.

a、 При создании объекта Function Key в EasyBuilder Pro, выберите “Import user data/Use [USB Security Key]”и, затем, щелкните по [Settings].

b、 В [Function mode] выберите [Import user accounts]. Укажите позицию, где находятся импортируемые данные в [Data position]. Выберите [Overwrite] в [Account import mode]; Панель будет хранить только импортированные данные учетных записей. Выберите [Append], Панель сохранит импортированные данные учетных записей и те, которые уже существуют. Отметьте [Delete file after importing user accounts], чтобы удалить файлы-источники, после импортирования. Нажмите [OK], для завершения настройки.



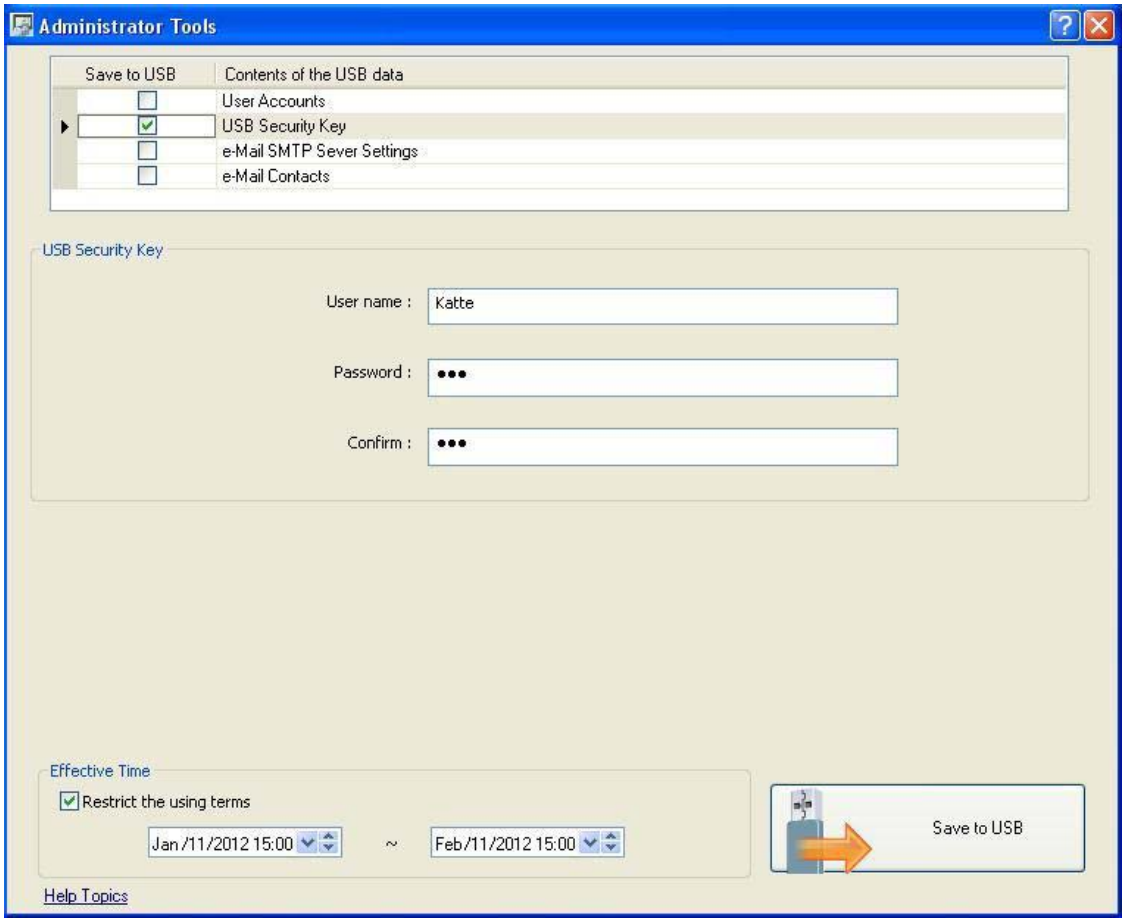
Хотите узнать, как импортировать «user accounts» при помощи *Function Key*?

Download the Demo Загрузите демо-проект:
ftp://ftp.weintek.com/eMT3000/Project/2011SPSdemo/Enhanced_Security.zip

36.3 USB-ключ защиты – USB Security Key

36.3.1 Введение в использование USB-ключа защиты

Сделайте отметку в «USB Security Key», чтобы выполнить необходимые настройки. С предопределенной информацией о пользователе (*user login information*), *USB Security Key* может быть использован для непосредственого доступа . Пример настроек ниже:

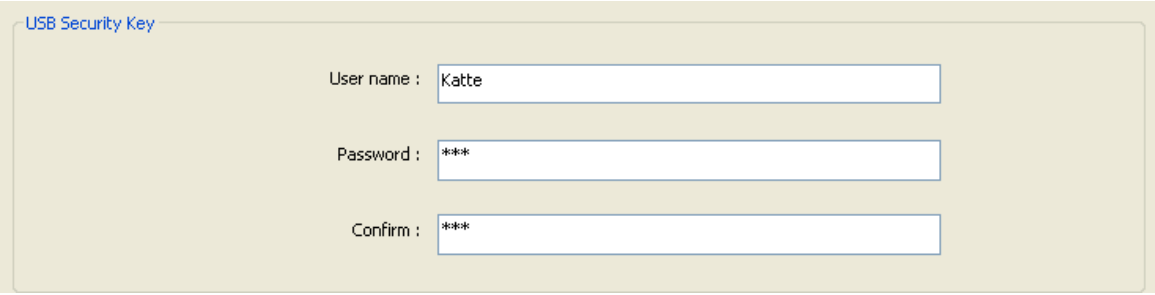


Настройки	Описание
User Name	Ввод имени пользование *Примечание 1
Password	Ввод пароля *Примечание 1
Confirm	Подтверждение пароля
Effective Time	Доступ с использованием *USB Security Key* в течение указанного периода времени. Если не указывать *Effective Time*, доступ возможен в любое время.
Save to USB	Сохранить данные в USB

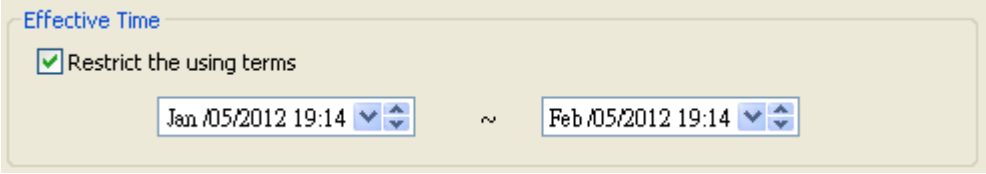
<Примечание 1> Может быть составлен из букв, цифр, ”-“, ”_”. Регистр учитывается.

36.3.2 Setting USB Security Key

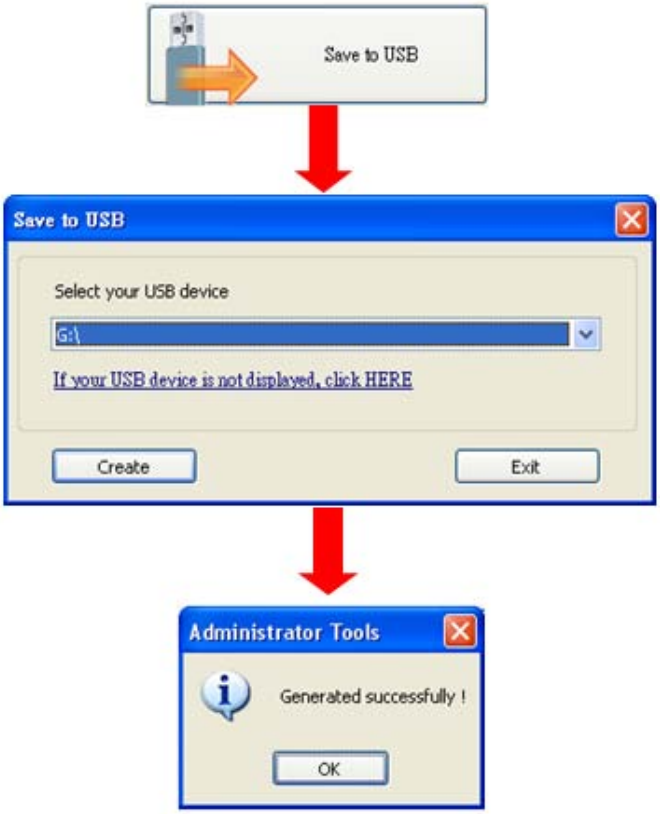
a、 Напечатайте имя и пароль в полях **[User name]** и **[Password]** . Подтвердите пароль в поле **[Confirm]**.



b、 Если отмечено **[Effective Time]** -> **[Restrict the using terms]**, доступ с использованием *USB Security Key* будет возможен только в течение указанного периода времени. Если не указывать *Effective Time*, доступ возможен в любое время.



c、 После завершения настроек, щелкните по **[Save to USB]**, выберите USB и, затем щелкните по **[Create] - Создать**. Сообщение ”Generated successfully!” появится, нажмите **[OK]**.

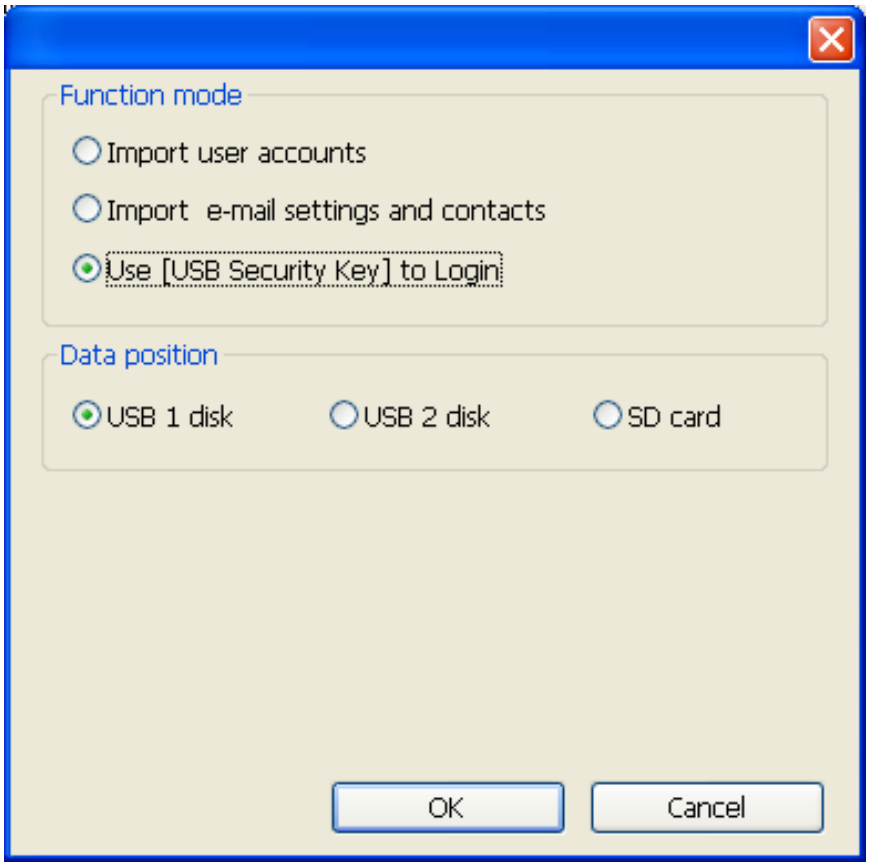


36.3.3 Настройки *USB Security Key* в EasyBuilder Pro

Создайте объект *Function Key* в EasyBuilder Pro, при касании этого объекта на экране панели, *USB Security Key* будет активирован для доступа. Далее приведен пример создания объекта *Function Key*.

a、При создании объекта *Function Key* в EasyBuilder Pro, выберите "**Import user data/Use [USB Security Key]**"и. затем, нажмите **[Settings]**.

b、 В **[Function mode]** выберите **[Use USB Security Key to Login]**. Укажите позицию, где хранятся данные «*security key*» в **[Data position]** и нажмите **[OK]** для завершения настройки.



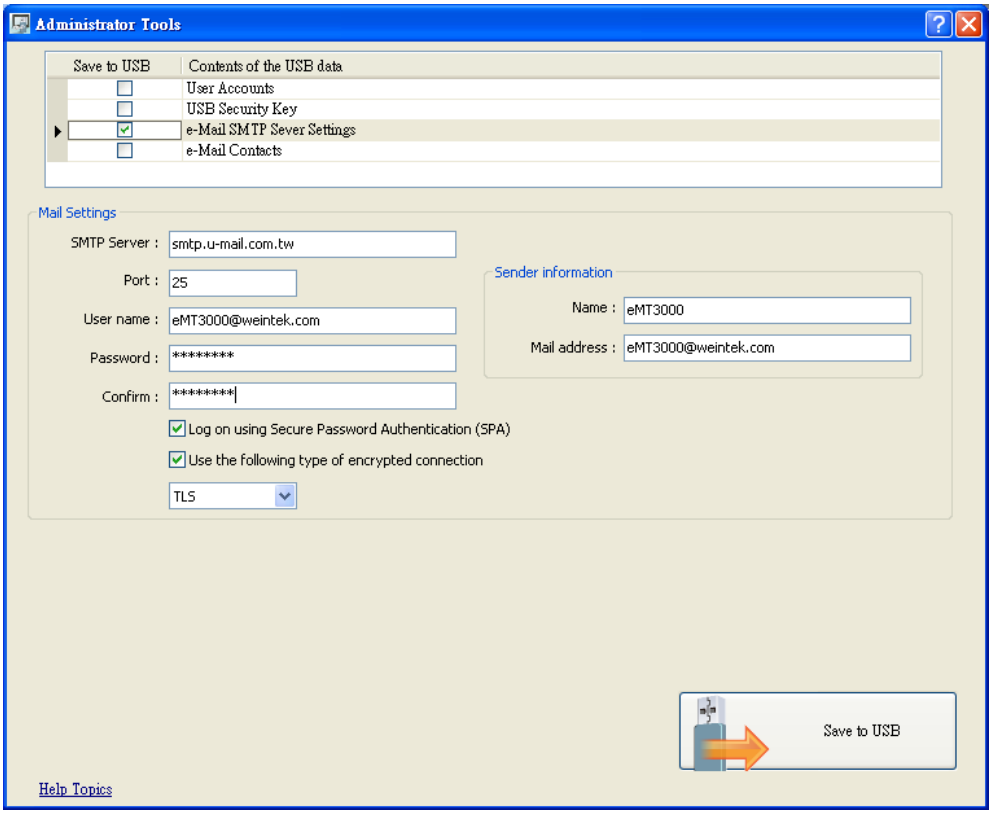
Хотите узнать, как войти в систему используя *USB Security Key* и *Function Key*?
Загрузите демо-проект:

ftp://ftp.weintek.com/eMT3000/Project/2011SPSdemo/Enhanced_Security.zip

36.4 Настройки сервера e-Mail SMTP

36.4.1 Введение в настройки e-Mail SMTP

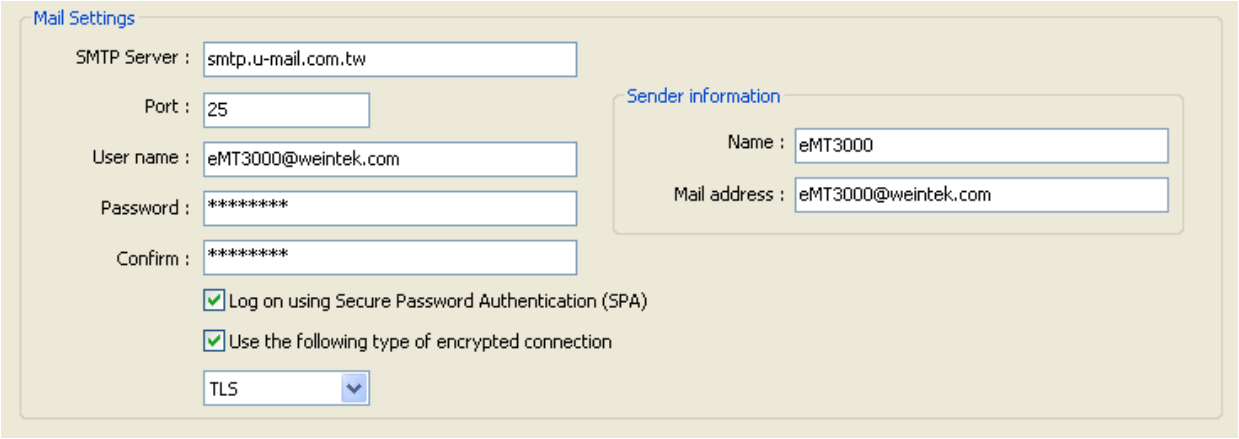
Сделайте отметку в окошке «e-Mail SMTP Server Settings», чтобы заполнить необходимые настройки, как показано ниже:



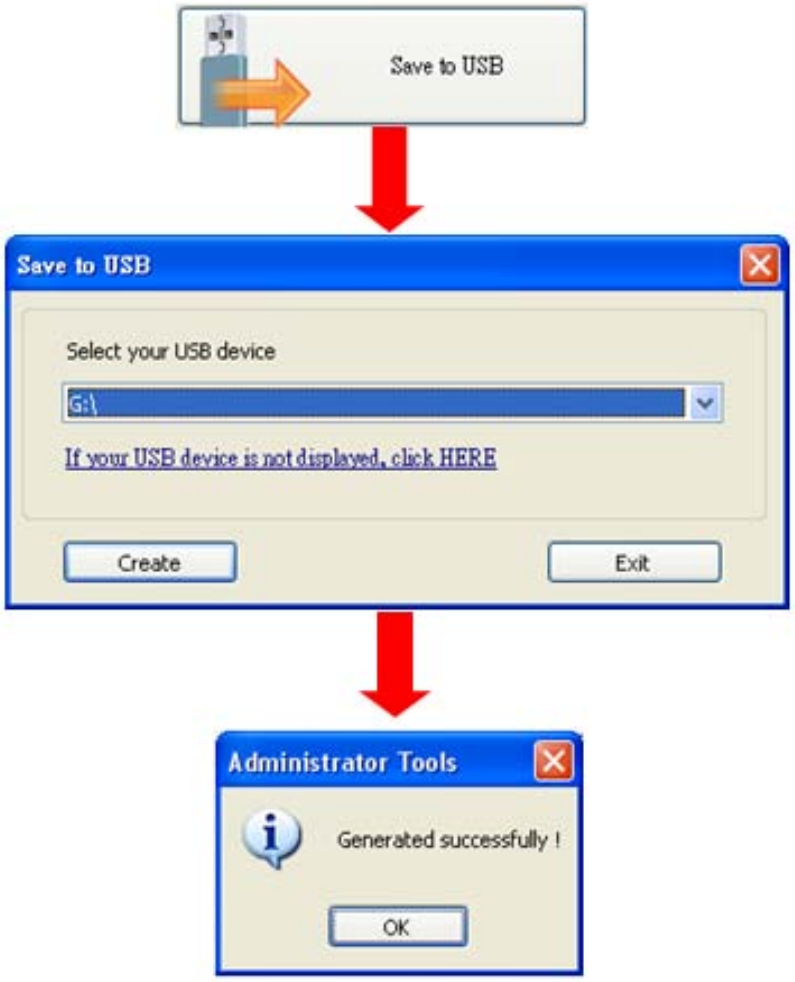
Настройки почты	Описание
SMTP Server	Укажите сервер SMTP
Port	Номер учетной записи сервера (SMTP Server account number)
User Name	Ввод имени пользователя (User e-mail account name)
Password	Ввод пароля (User e-mail account password)
Confirm	Подтверждение пароля
Отправитель	**Описание**
Name	Ввод имени отправителя, отображаемого при получении почты
Mail address	Адрес отправителя, отображаемый при получении почты
Save to USB	Сохранить на USB

36.4.2 Настройки сервера SMTP e-Mail

a、 Далее приведен пример настроек e-mail SMTP:



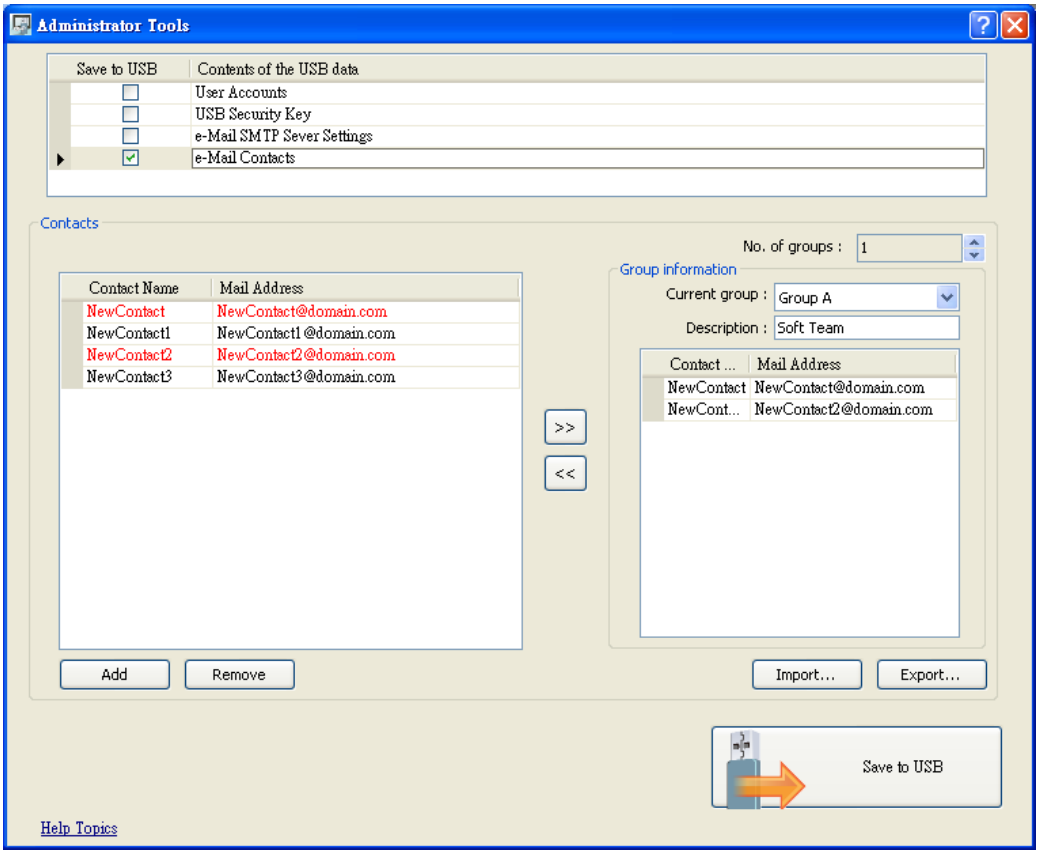
b、 После завершения настроек, щелкните по **[Save to USB]**, выберите USB и, затем щелкните по **[Create] - Создать**. Появится сообщение ”*Generated successfully*!” , нажмите **[OK]**..



36.5 Контакты e-Mail

36.5.1 Введение в Контакты e-Mail

Сделайте отметку в окошке «e-Mail Contacts», чтобы заполнить необходимые настройки, как показано ниже:



Настройки	Описание
Add	Добавить новый контакт *Примечание 1
Remove	Удалить контакт
No. of groups	Число групп *Примечание2
Current group	Имя текущей группы *Примечание3
Description	Описание группы
Import	Информация по Импорту группы
Export	Информация по Экспорту группы
Save to USB	Сохранить данные на USB

<Примечание 1> Максимум 256 контактов может быть добавлено.

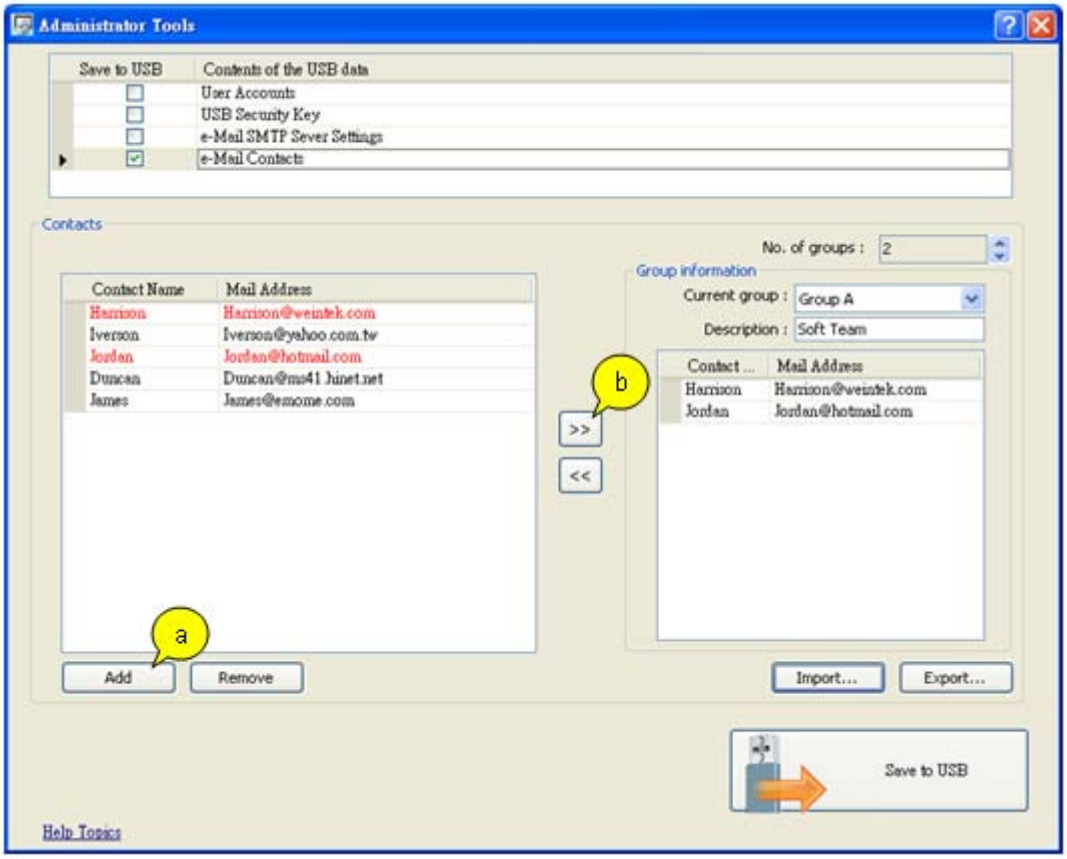
<Примечание 2> Максимум 16 групп может быть добавлено **(Group A ~ Group P).**

<Примечание 3> Group A ~ P, когда число групп “1”, только Group A будет существовать, Если “2”, то Group A и Group B будут существовать.

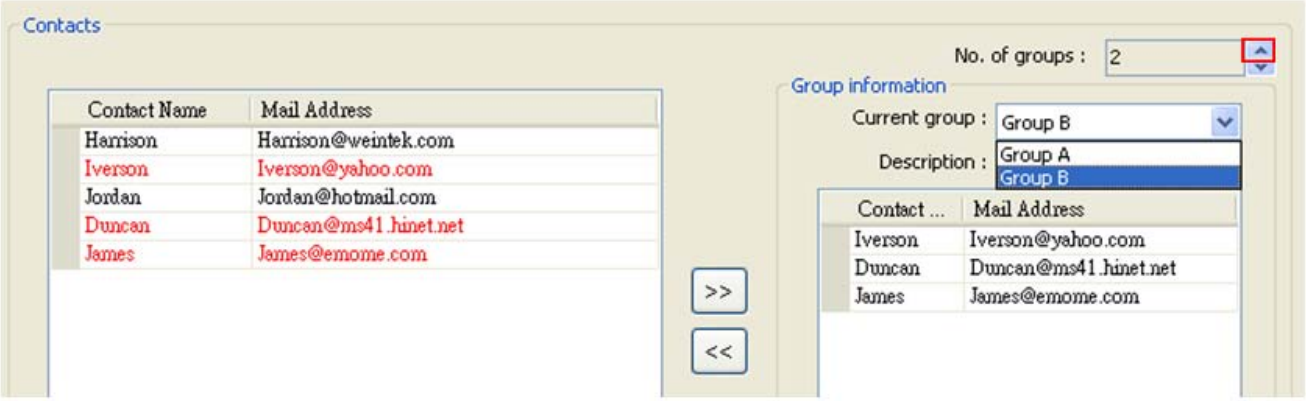
36.5.2 Настройки Контактов e-Mail

a、 Щелкните по **[Add],** чтобы добавить все контакты.

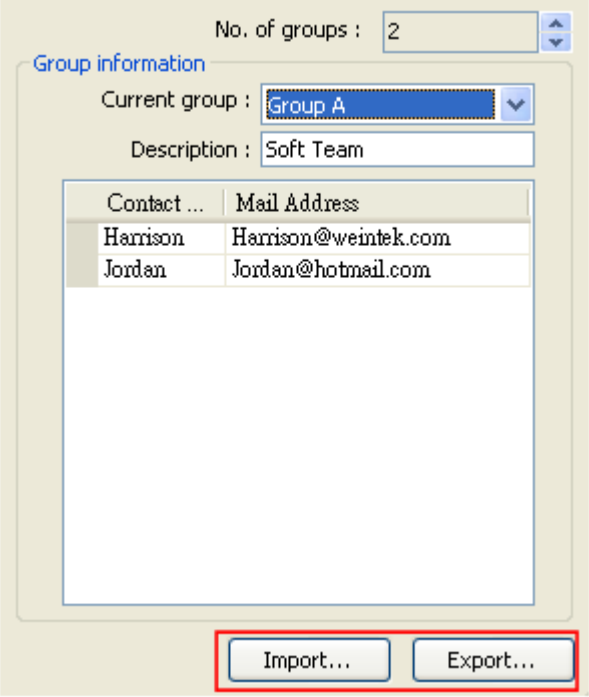
b、 Добавьте контакты в *Group A*, добавленные контакты будут выделены красным шрифтом.



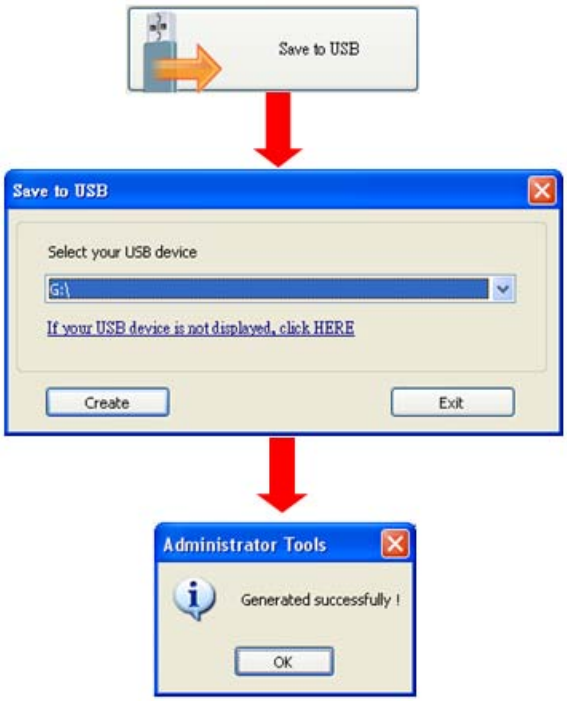
c、 В **[No. of groups]** нажмите «▲», чтобы добавить новую группу, *Group B* можно будет увидеть. Повторите шаги «a» и «b», чтобы добавить контакты в группы.



d、 После добавления всех контактов e-mail , щелкните по **[Export],** чтобы сделать резервную копию данных (backup data) . Для перестройки и изменения, щелкните по **[Import],** чтобы импортировать резервные данные.



e、 После завершения настроек, щелкните по **[Save to USB]**, выберите USB и, затем щелкните по **[Create] - Создать**. Сообщение ”Generated successfully!” появится, нажмите **[OK]**.

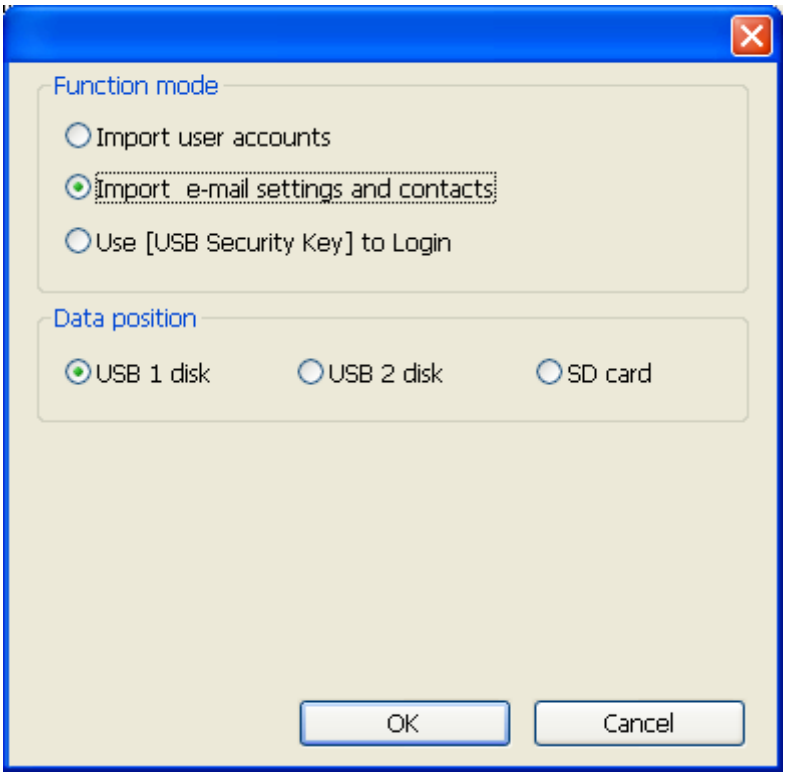


36.5.3 Использование EasyBuilder Pro для импорта настроек e-Mail и Контактов

Создайте объект *Function Key* в EasyBuilder Pro, при касании этого объекта на экране панели, будет выполняться импорт. Далее пример создания этого объекта *Function Key*.

a、 При создании объекта *Function Key* в EasyBuilder Pro, выберите "**Import user data/Use [USB Security Key]**", затем выберите **[Settings]**.

b、 В **[Function mode]** выберите **[Import e-mail settings and contacts]**. Укажите позицию, где храняться данные в **[Data position],** нажмите **[OK],** чтобы закончить настройку.



Хотите узнать, как импортировать настройки e-mail и контакты, используя *Function Key*?
Загрузите демо-проект:
ftp://ftp.weintek.com/eMT3000/Project/2011SPSdemo/e_Mail.zip