



# UNI-PRO 3

**СРЕДА РАЗРАБОТКИ ДЛЯ  
ПРОГРАММИРУЕМЫХ КОНТРОЛЛЕРОВ**



**Руководство по программному обеспечению**

**КОД 114UP3E464**

**Важная информация**

Перед использованием внимательно прочтите данное руководство и изучите все предупреждения, используйте руководство при работе для дальнейших консультаций.

# Содержание

1	ОСНОВНЫЕ ПОЛОЖЕНИЯ .....	9
1.1	Информация о UNI-PRO 3 .....	9
1.2	Системные требования .....	10
1.2.1	Требования к оборудованию .....	10
1.2.2	Требования к программному обеспечению .....	10
1.3	Разработка проектов с использованием UNI-PRO .....	11
1.4	Ваша первая UNI-PRO программа .....	13
2	ОСНОВНЫЕ ОПЕРАЦИИ .....	18
2.1	Создание нового проекта .....	18
2.2	Сохранение и загрузка существующего проекта .....	18
2.3	Сущности: установка и использование .....	19
2.3.1	Имя сущности .....	19
2.3.2	Введение сущностей .....	20
2.3.3	Перемещение сущностей .....	20
2.3.4	Удаление сущностей .....	20
2.3.5	Вырезание, копирование и вставка сущностей .....	20
2.4	Использование листов управления .....	21
2.5	Соединение сущностей управления .....	22
2.5.1	Имена входных и выходных меток сущностей .....	23
2.5.2	Имена внешних связей .....	24
2.6	Использование объектов-массивов .....	25
2.6.1	Совместимость и связи .....	26
2.7	Тип SJ_CHAR для управления строками .....	27
2.8	Использование CodeEditor .....	28
2.8.1	Алгоритм с использованием входов/выходов типа массив .....	32
2.8.2	Входы и выход типа массив .....	32
2.8.3	Изменение порядка входных терминалов алгоритмов .....	33
2.8.4	Использование глобальных алгоритмов .....	33
2.9	Симулятор алгоритмов .....	35
2.9.1	Абсолютная точка останова .....	40
2.9.2	Условная точка останова .....	42
2.9.3	Установочное значение .....	44
2.9.4	Использование во время отладки .....	46
2.10	Создание библиотек .....	47
2.11	Создание EIML-страниц .....	48
2.12	Симуляция EIML-страниц .....	49
2.13	Выбор оборудования .....	51
2.13.1	Сеть ModbusMaster .....	53
2.13.2	Конфигурация VASnet MS/TP .....	53
2.13.3	Конфигурация протоколов поверх Ethernet .....	54
2.14	Соединение с физическими терминалами .....	58
2.14.1	Цифровые входы .....	58
2.14.2	Цифровые выходы .....	59
2.14.3	Аналоговые входы .....	60
2.14.4	Аналоговые выходы .....	61
2.14.5	Часы .....	62
2.14.6	Индикаторы .....	62
2.14.7	Кнопки .....	63

2.14.8	Вход команды .....	64
2.14.9	Выход команды.....	65
2.14.10	Звуковые сигналы .....	66
2.15	Компиляция программы .....	67
2.15.1	Использование кеша .....	69
2.15.2	Компиляция с дисплеем EPJ-Color .....	69
2.16	Загрузка программы .....	70
2.16.1	Загрузка и программирование через USB кабель .....	70
2.16.2	Загрузка с дисплеем EPJ-Color .....	71
2.16.3	Загрузка и программирование через USB накопитель .....	71
2.17	Использование отладчика .....	73
2.17.1	Текущее значение при отладке .....	79
2.18	Web-сервер .....	80
2.18.1	Симулятор Web-сервера .....	82
2.18.2	Экспортируемые объекты .....	83
2.18.3	Работа .....	85
2.18.4	Чтение объектов .....	85
2.18.5	Запись объектов .....	86
2.18.6	Защита от записи .....	86
2.18.7	Защита Web-страниц .....	87
2.18.8	Удаленное управление журналом данных .....	87
2.18.9	Информация о конфигурации сети .....	88
2.18.10	Автоматическая генерация Web-сайта .....	89
3	ОСНОВНЫЕ ЭЛЕМЕНТЫ .....	92
3.1	Типы данных .....	92
3.1.1	Типы простых данных .....	92
	CJ_VOID .....	93
	CJ_LED и CJ_BUZZ .....	93
	CJ_DATE .....	93
	CJ_TIME .....	93
	CJ_DATETIME .....	93
3.1.2	Структурные типы данных .....	94
	CJ_ANALOG.....	94
	CJ_CMD .....	94
	CJ_BTN .....	95
	CJ_DATE_STRUCT .....	95
	CJ_TIME_STRUCT .....	95
	CJ_DATE_TIME_STRUCT .....	96
3.2	Таблица совместимости типов .....	97
3.3	Библиотечные функции встроенные в ПО и используемые в алгоритмах .....	98
3.3.1	CJ_DATETIME StructToDate(CJ_DATETIME_STRUCT rtc) .....	98
3.3.2	CJ_DATE StructToDate (CJ_DATE_STRUCT date) .....	98
3.3.3	CJ_TIME StructToTime (CJ_TIME_STRUCT time) .....	98
3.3.4	CJ_DATETIME_STRUCT DateTimeToStruct(CJ_DATETIME Value) .....	98
3.3.5	CJ_DATE_STRUCT DateToStruct (CJ_DATE Value) .....	98
3.3.6	CJ_TIME_STRUCT TimeToStruct (CJ_TIME Value) .....	98
3.3.7	CJ_BYTE CJ_GetSeconds(CJ_DATETIME dt) .....	98
3.3.8	CJ_BYTE CJ_GetMinutes(CJ_DATETIME dt) .....	98
3.3.9	CJ_BYTE CJ_GetHours(CJ_DATETIME dt) .....	98
3.3.10	CJ_BYTE CJ_GetDay(CJ_DATETIME dt) .....	98
3.3.11	CJ_BYTE CJ_GetWeekDay(CJ_DATETIME dt).....	98

3.3.12	CJ_BYTE CJ_GetMonth(CJ_DATETIME dt) .....	98
3.3.13	CJ_BYTE CJ_GetYear(CJ_DATETIME dt) .....	99
3.3.14	CJ_TIME CJ_GetTime(CJ_DATETIME dt) .....	99
3.3.15	CJ_DATE CJ_GetDate(CJ_DATETIME dt) .....	99
3.3.16	CJ_BIT CJ_GetSecondTic (void) e CJ_BIT CJ_GetMinuteTic (void) .....	99
3.3.17	CJ_BIT CJ_IsFirstMain(void) .....	99
3.3.18	CJ_WORD CJ_SemaphoreRead(CJ_WORD i).....	99
3.3.19	CJ_BYTE CJ_SemaphoreWrite(CJ_WORD i, CJ_WORD value) .....	99
3.3.20	CJ_BIT CJ_FlagWrite (CJ_WORD i, CJ_BIT val) - CJ_BIT CJ_FlagRead (CJ_WORD i) .....	99
3.3.21	CJ_SHORT CJ_WriteVarExpo(CJ_WORD add, CJ_LONG value) .....	100
3.3.22	CJ_LONG CJ_ReadVarExpo(CJ_WORD add) .....	100
3.3.23	CJ_BYTE CJ_ModbusAskQueue(void) .....	100
3.3.24	CJ_BYTE CJ_SendCommand (CJ_BYTE channel, CJ_BYTE node, CJ_BYTE command, CJ_SHORT par1) .....	100
3.3.25	CJ_WORD CJ_MaxMainTime(void) .....	100
3.3.26	CJ_WORD CJ_MinMainTime(void) .....	101
3.3.27	CJ_WORD CJ_RunMainTime(void) .....	101
3.3.28	CJ_BYTE CJ_MaxInterruptTime(void) .....	101
3.3.29	CJ_BYTE CJ_MinInterruptTime(void) .....	101
3.3.30	CJ_BYTE CJ_RunInterruptTime(void) .....	101
3.3.31	CJ_BIT CJ_Stack_Error_Read(void) .....	101
3.3.32	CJ_BIT CJ_Math_Error_Read(void) .....	101
3.3.33	CJ_BIT CJ_DivByZero_Error_Read(void) .....	101
3.3.34	CJ_BIT CJ_Overflow_Error_Read(void) .....	101
3.3.35	CJ_BIT CJ_Underflow_Error_Read(void) .....	101
3.3.36	CJ_BIT CJ_NaN_Error_Read(void) .....	101
3.3.37	void CJ_DivByZero_Error_Write(void) .....	101
3.3.38	void CJ_Overflow_Error_Write(void) .....	101
3.3.39	void CJ_Underflow_Error_Write(void) .....	101
3.3.40	void CJ_NaN_Error_Write(void) .....	101
3.3.41	void CJ_Math_Error_Reset(void) .....	101
3.3.42	void CJ_DivByZero_Error_Reset(void) .....	101
3.3.43	void CJ_Overflow_Error_Reset(void) .....	102
3.3.44	void CJ_Underflow_Error_Reset(void) .....	102
3.3.45	void CJ_NaN_Error_Reset(void) .....	102
3.3.46	CJ_DWORD CJ_Read_DI_PulseCnt (CJ_BYTE in) .....	102
3.3.47	void CJ_Clear_DI_PulseCnt (CJ_BYTE in).....	102
3.3.48	CJ_WORD CJ_Read_DI_Frequency (CJ_BYTE in) .....	102
3.3.49	void CJ_ModBus_SetProfile(CJ_WORD numProfile, CJ_BIT currentStatus) .....	102
3.3.50	CJ_DATETIME RTCLocalRead(void) .....	102
3.3.51	CJ_SHORT RTCLocalWrite(CJ_DATETIME val) .....	102
3.3.52	CJ_WORD CJ_GetTimer1ms(void) .....	102
3.3.53	CJ_BIT EIMLinEdit(void) .....	102
3.3.54	CJ_BYTE USB_PenDriveIsConnect(void) .....	102
3.3.55	void CJ_AIxFilterWrite(CJ_BYTE value) .....	102
3.3.56	void CJ_AIxEnaFilterWrite(CJ_BIT value) .....	102
3.4	Компоненты управления .....	103
3.4.1	Программное обеспечение .....	103
	Алгоритм .....	103
	Глобальный алгоритм .....	104

Алгоритмы в одном экземпляре .....	104
Подсхема .....	104
Fix .....	105
Par .....	106
Pers .....	107
Var .....	108
Таймер .....	109
Команда входящая .....	111
Команда исходящая .....	111
Пометка .....	112
3.4.2 Оборудование .....	114
Цифровой вход .....	114
Цифровой выход .....	115
Аналоговый вход .....	116
Аналоговый выход .....	117
Кнопка .....	118
Индикатор .....	118
Звуковой сигнал .....	119
Часы .....	120
3.4.3 Библиотеки .....	122
3.4.4 Документация по используемым библиотекам .....	122
3.5 Элементы на EIML страницах .....	123
3.5.1 Панель инструментов .....	123
3.5.2 Внесение элементов на EIML страницу .....	123
3.5.3 Свойства страницы .....	123
3.5.4 Мультиязычные страницы .....	125
3.5.5 Мультиязычные текстовые элементы (с языковым ID) .....	126
3.5.6 Страницы с 256 цветами (EPJ Color) .....	126
3.5.7 Страницы с 16 цветами (EPJ Graph) .....	127
3.5.8 Свойства элемента .....	128
Текст .....	129
Переменная .....	130
Строки .....	133
Значки .....	134
Combo .....	136
Таблица .....	138
Динамический список .....	141
График .....	144
Линия .....	146
Прямоугольник .....	147
3.5.9 Множественный выбор из дерева страниц .....	147
4 РАСШИРЕННЫЕ ДЕЙСТВИЯ .....	149
4.1 Команды .....	149
4.2 Перемещение по страницам .....	152
4.3 Защита информации паролем .....	153
4.4 Установка задач выполнения .....	154
4.5 Задание последовательности выполнения программы .....	154
4.5.1 Список вызовов .....	156
4.5.2 Порядок выполнения при активном списке вызовов .....	158
4.6 Экспорт внутренних переменных .....	159
4.6.1 Документация Modbus и CANopen .....	162

4.7	Условная видимость .....	163
4.7.1	Функции встроенного ПО void CJ_SetCondVisBit(word idx, bool value) .....	163
4.7.2	Пример .....	164
4.8	Семафоры .....	166
4.9	Упорядочивание EIML элементов .....	167
4.10	Настройки проекта .....	168
4.10.1	Информация о версии .....	168
4.10.2	Проект .....	168
4.10.3	История .....	170
4.10.4	Лог данных .....	171
4.10.5	Определения проекта .....	172
4.10.6	Структуры .....	172
4.10.7	Опции .....	173
4.10.8	Лог изменений в проекте .....	174
4.11	Лог данных .....	176
4.12	Шрифт, настраиваемый для EIML .....	178
4.12.1	Сохранение выбранных шрифтов .....	178
4.13	Фоны для EIML .....	180
4.14	Битовые изображения для EIML .....	181
4.15	Конфигурация среды .....	182
4.15.1	Опция «Загрузить все схемы» .....	184
4.16	Экспорт / Импорт .....	191
4.17	Экспорт внутренних переменных библиотек/макроблоков .....	191
4.18	Резервная копия проекта .....	192
4.19	Инструмент программирования продукта (Download Manager) .....	193
4.20	VTP Maker .....	195
4.20.1	Свойства RICS .....	197
4.21	Parameters Manager и VTPEditor .....	198
4.21.1	Создание и изменение файла VTP драйвера .....	198
4.22	Расширения и конфигурация удаленных терминалов .....	200
4.22.1	Intrabus Master .....	200
4.22.2	CAN master.....	201
4.23	Ссылки на объекты Modbus Master .....	202
4.24	ВАСnet.....	203
4.24.1	Конфигурация ВАСnet MS/TP .....	203
4.24.2	Конфигурация ВАСnet IP .....	204
4.24.3	Файл профиля ВАСnet .....	205
4.24.4	Конфигуратор ВАСnet .....	205
4.24.5	Свойства объекта устройства .....	207
4.24.6	Свойства объекта Аналоговый Вход .....	208
4.24.7	Свойства объекта Аналоговый Выход .....	209
4.24.8	Свойства объекта Аналоговое Значение .....	211
4.24.9	Свойства объекта Цифровой Вход .....	212
4.24.10	Свойства объекта Цифровой Выход .....	214
4.24.11	Свойства объекта Цифровое Значение .....	215
4.24.12	Свойства объекта с несколькими значениями состояния .....	217
4.24.13	Свойства объекта Класс Уведомлений .....	218
4.25	Функция “Удаление и Сохранение ссылок” и LinkFixer .....	220
4.25.1	Функция “Правка ссылок” .....	221
4.26	Конфигуратор CAN .....	222
4.27	Таблица текстов .....	225

4.28	Переводчик EIML .....	227
4.29	Редактор сущностей .....	228
4.30	Профиль CANOpen для EIML .....	229
4.30.1	Список сущностей CANOpen .....	230
4.30.2	Сохранение профиля CANOpen .....	230
4.31	Контроль совместимости для дисплея EPJ-Color .....	231
4.32	Меры предосторожности при разработке проекта .....	231
4.33	Переменные “Run-time” .....	231
4.34	Регистрация лицензии .....	234
4.35	Палитра 256 цветов в UNI-PRO .....	235



## 1 ОСНОВНЫЕ ПОЛОЖЕНИЯ

### 1.1 Информация о UNI-PRO 3

*UNI-PRO 3* (далее просто *UNI-PRO*) для Windows® является интегрированной средой разработки для программируемых контроллеров C-Pro 3, которая позволяет настраивать и осуществлять комплексное системное управление проектами.

Она позволяет легко и удобно для пользователя установить цепь модулей оборудования, которую можно использовать для программирования контроллера, а также для задания всех функциональных особенностей и особенностей управления системой.

Среда программирования состоит из наборов связанных и взаимодействующих друг с другом команд, которые позволяют выполнять следующие операции:

- Создание и управление объектами и алгоритмами управления;
- Создание и управление пользовательскими интерфейсами;
- Настройка сетей контроллера;
- Выбор оборудования, компиляция, проверка и скачивание кода;
- Наладка работы контроллера;
- Вспомогательное управление пользовательскими параметрами контроллера;
- Создание и изменение библиотек объектных модулей и управление соответствующими версиями;
- Автоматическое создание документов проекта;

## **1.2 Системные требования**

### **1.2.1 Требования к оборудованию**

Ниже перечислены минимальные системные требования для работы UNI-PRO:

Микропроцессор @ 2.0 GHz (или мощнее)

2 GB RAM (или больше)

н° 2 USB-порта

500 MB свободного места на жестком диске

SVGA видео карта, 1024x768 с 512 MB памяти (или больше)

### **1.2.2 Требования к программному обеспечению**

Одна из ниже перечисленных операционных систем должна быть установлена на компьютере для работы UNI-PRO:

Microsoft® Windows XP

Microsoft® Windows VISTA

Microsoft® Windows 7

Microsoft® Windows 8/8.1

Microsoft® Windows 10

Важное примечание: Для установки, запуска и удаления программного обеспечения UNI-Pro требуются права администратора.

Необходима программа просмотра PDF, например Acrobat Reader версии 6.0 или выше, для правильного просмотра страниц справки и документации по оборудованию/программному обеспечению.

**ВАЖНО.** Настройка экрана по умолчанию, используемая UNI-PRO, составляет 96 DPI (точка на дюйм). Изменение этой настройки приводит к изменению размера положения и размеров элементов пользовательского интерфейса. Поэтому некоторые окна могут отображаться некорректно. Рекомендуется использовать UNI-PRO с настройкой 96 точек на дюйм.

### 1.3 Разработка проектов с использованием UNI-PRO

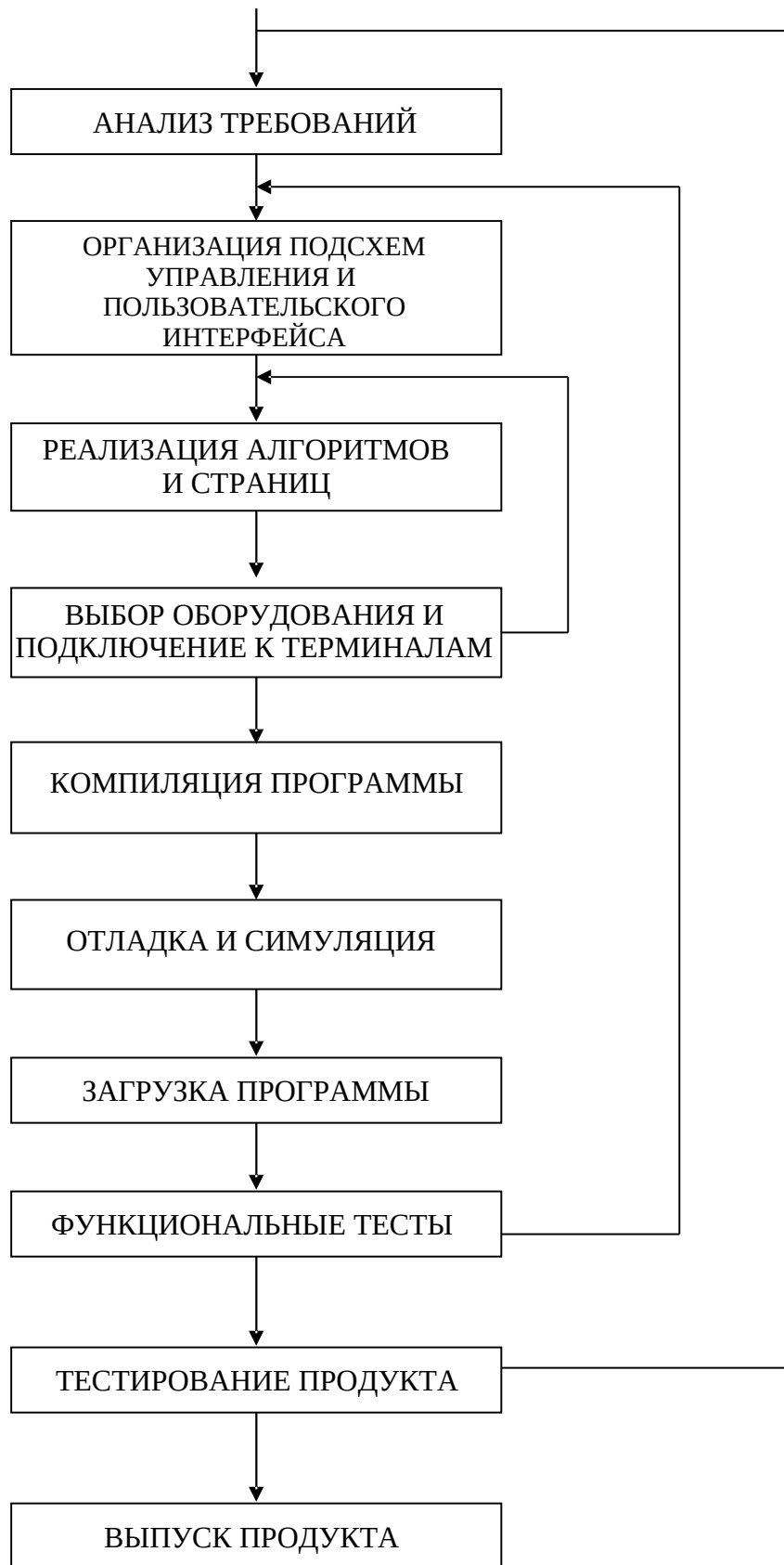
С помощью среды программирования UNI-PRO Вы можете создавать и осуществлять распределенное управление проектами независимо от их размера.

UNI-PRO является не только средством разработки кода контроллера, а также полезным средством проектирования для создания и реализации Вашего собственного ноу-хау. Фактически Вы можете использовать его на стадии анализа, чтобы выбрать наилучший вариант управления определенной системой, или для рассмотрения вариантов управления, созданных в соответствии с требованиями заказчика. Создание библиотек объектных модулей позволяет создать собственный набор объектов, которые могут быть повторно использованы в различных проектах, защищая, таким образом, внутреннюю информацию компании и минимизируя время программирования.

Во многих случаях метод разработки проекта является фундаментальным. С помощью UNI-PRO вы можете выполнять разработку как методом “сверху-вниз” (т.е. сначала определение структуры системы, а затем добавление деталей) так и методом “снизу-вверх” (т.е. начиная с основы компьютера, элементов нижнего уровня и, затем, добавляя все более и более сложные блоки, один модуль за раз и так до конца).

Цикл разработки проекта состоит из нескольких стадий. На первой стадии графические объекты, представляющие основные компоненты управления, вносятся в рабочий лист. После настройки свойств этих объектов они легко используются для создания логической структуры контроллера. Во время этой стадии также будут разработаны страницы, формирующие пользовательский интерфейс программы. Вторая стадия включает соединение с “физической” частью выбранного контроллера, т.е. с входами, выходами и внутренними ресурсами оборудования. На следующей стадии составляется графическая часть программы, которая включает соответствующие алгоритмы и страницы пользовательского интерфейса. В завершающей стадии загружается код и запускается программа для проверки ее работоспособности и внесения необходимых изменений.

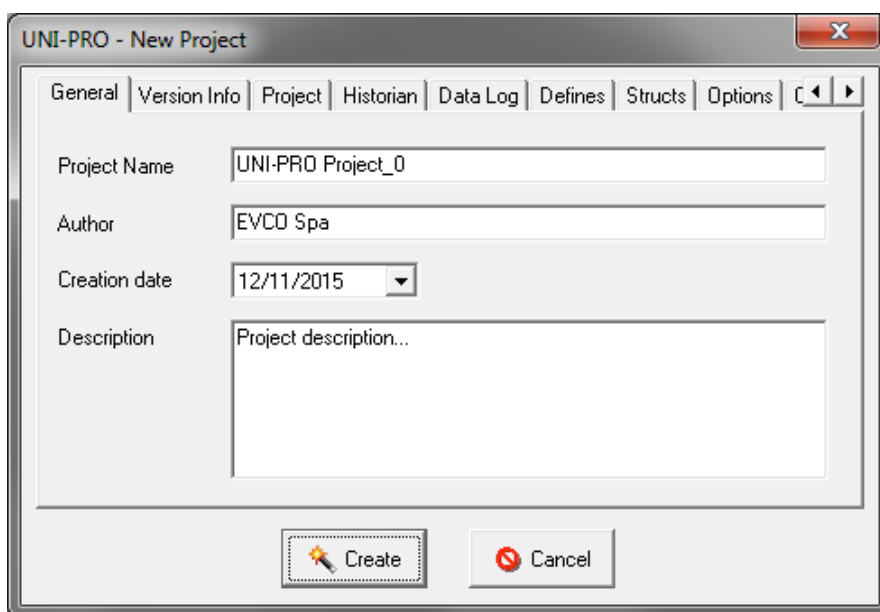
Чтобы использовать UNI-PRO для разработки проекта, необходимо следовать циклической и поэтапной методике, соответствующей данной диаграмме:



## 1.4 Ваша первая UNI-PRO программа

Ниже приведен пример программы, чтобы помочь Вам ознакомиться со средой программирования UNI-PRO. Программа использует *Digital In* и *Digital Out*, которые будут связаны друг с другом. Желаемый эффект заключается в том, чтобы при переключении цифрового входа связанный с ним цифровой выход тоже переключался. В дополнении будет приведен пример создания EIML-страницы, в которой цифровой выход присоединен к объекту Combo с двумя текстами таким образом, чтобы различные сообщения выводились в соответствии с состоянием выхода.

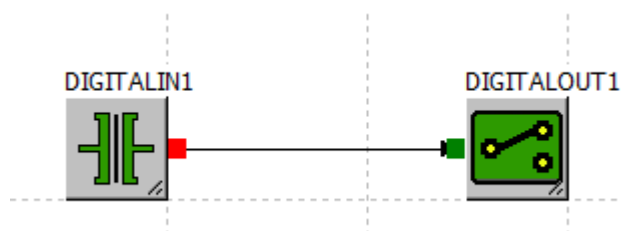
Чтобы создать новый проект, мы можем выбрать *Файл/Новый проект* в меню или нажать значок *Создать новый проект* в панели инструментов. Появится окно с названием “UNI-PRO – новый проект”. Вводим название “Пример1” в поле «Название проекта» и нажимаем «Создать».



В первую очередь необходимо использовать цифровой тип входа и выхода, который соответствуют реле. На панели оборудования выбираем *Добавить цифровой вход* и нажимаем кнопку мыши в любом месте рабочего листа: появится значок цифрового входа со стандартным названием DIGITALIN1. Можно использовать мышку для перемещения его в нужную позицию. Таким же способом располагаем *Цифровой выход*.

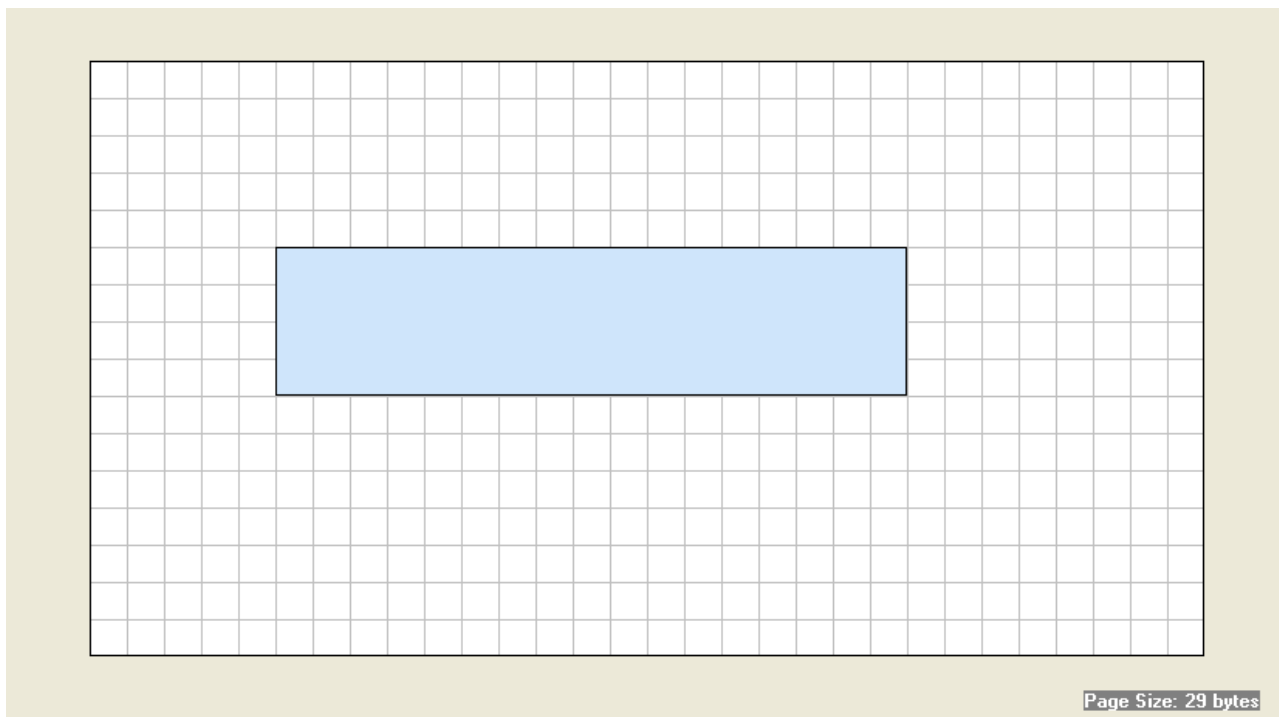
Как только оба объекта будут добавлены на главный рабочий лист, необходимо их соединить. Для этого подводим курсор мыши к выходу значка цифрового входа и, как только он поменяет размер, нажимаем на левую кнопку мыши. Мы увидим появившуюся стрелку у курсора мыши, выходящую из **выходного контакта** значка входа. Подводим курсор мыши к входному контакту значка цифрового выхода и, как только он изменит размер, нажимаем на левую кнопку мыши.

Если мы выполнили процедуру правильно, оба объекта будут соединены стрелкой.



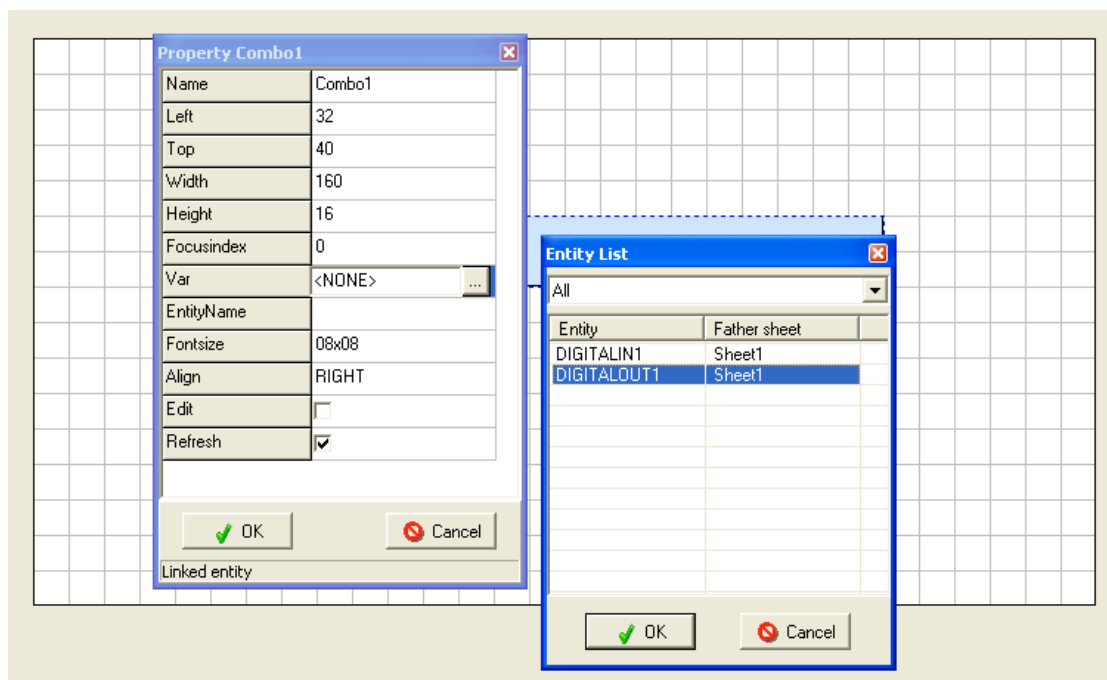
Чтобы сохранить проект, выбираем пункт меню *Файл/Сохранить проект* или нажимаем на соответствующий значок на панели инструментов.

Чтобы создать графический интерфейс: в главном меню выбираем *Файл/Новый/Новая EIML стр. (128x64)*. Появится пустая страница, соответствующая графическому ЖК-дисплею с разрешением 128x64. На панели страницы выбираем значок *Combo*, затем нажимаем на пустую страницу и перемещаем ее вниз, удерживая левую кнопку мыши, до тех пор, пока не образуется прямоугольник:



Как только прямоугольник будет создан, можно изменить его положение. Если необходимо поменять размер, открываем окно свойств (для этого наводим курсор на прямоугольник и нажимаем правую кнопку мыши, выбираем «Свойства») и устанавливаем требуемые значения *X*, *Y*, *Ширина* и *Высота*. Также размер объекта можно изменить графически, расположив курсор мыши в углах элемента.

В дополнение в окне свойств, свяжем свойство *Переменная* с переменной DigitalOut, как показано на рисунке:



Таким образом, мы создали соединение переменной DigitalOut с графическим элементом Combo1, который является ее представлением.

В завершение, присвоим свойству *Выравнивание* значение CENTER, *Размер шрифта* - 8x16 и поставим галочку на свойстве *Обновлять*.

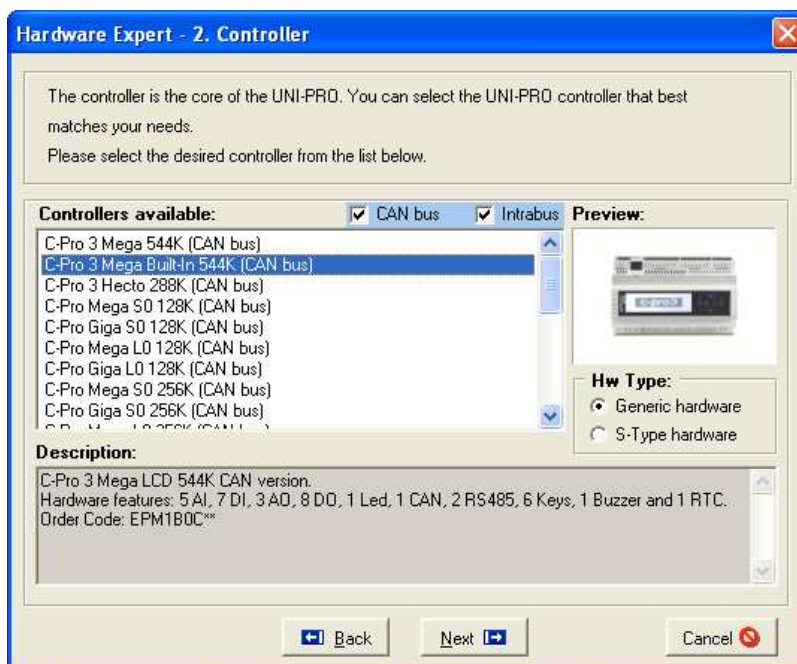
Теперь щелкнем правой клавишей мыши на прямоугольнике, чтобы выбрать *Мастер Combo*. Сохраним значение Text в свойстве *Тип*, а в поле *Имя файла* введем “Пожалуйста, включите”, затем нажмем «Добавить». Таким образом, данный текст останется связанным со значением 0 переменной Digitalout (контакт разомкнут). Повторим последнюю операцию, но в этот раз присвоим текст “Hello World!” значению 1 (контакт замкнут).

Теперь попробуем запустить компиляцию, выбрав в меню *Проект/Компилировать* или нажав соответствующий значок на панели инструментов.

Если мы еще не выбрали оборудование, с использованием которого будет работать проект, автоматически запустится программа **Hardware Expert**.



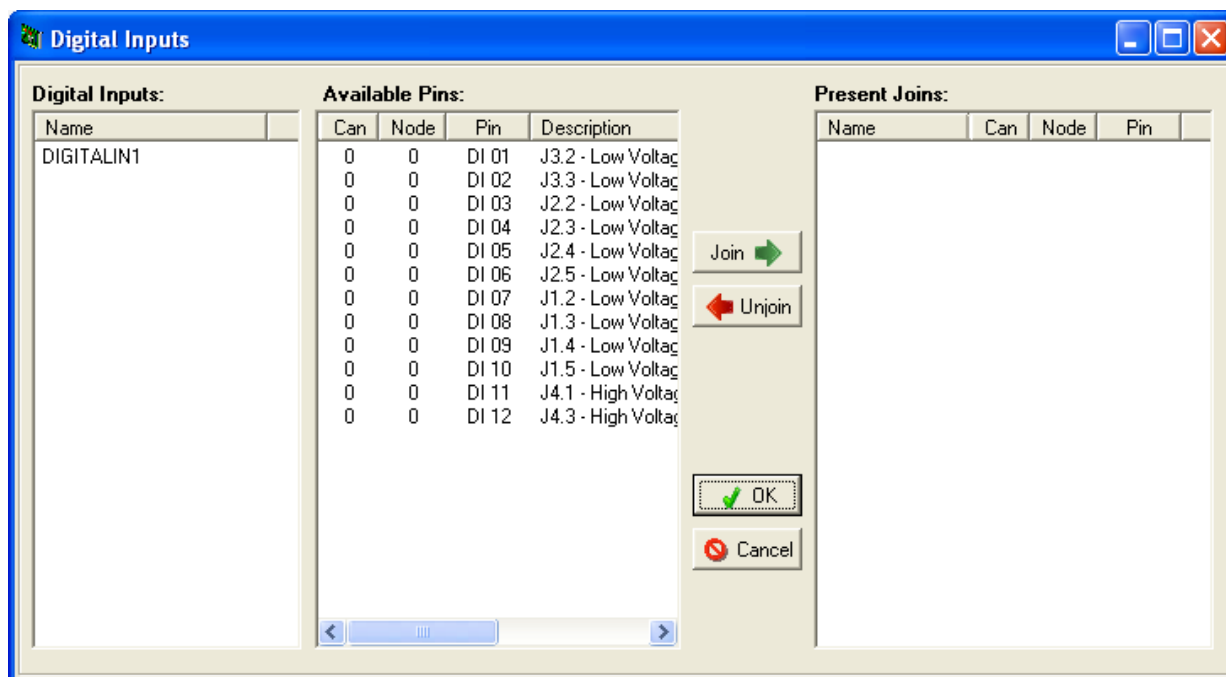
После ознакомления с введением нажмем *Далее*, чтобы перейти к следующему окну, где с левой стороны будет показан список имеющихся контроллеров, в центре - краткое описание оборудования, а с правой стороны изображение контроллера.



Выберем подходящий контроллер и нажмем *Далее*. В третьем окне будут показаны конфигурации последовательного порта (оставим без изменений). В четвертом окне показаны имеющиеся расширения с кратким описанием с правой стороны, а также предлагается присоединить их к выбранной шине CAN. Не будем добавлять никаких расширений и нажмем *Далее* (для более подробного описания, пожалуйста, обратитесь к разделу **Выбор оборудования**). В пятом окне можно выбрать тип дисплея. В последнем окне **Hardware Expert** будет показана выбранная конфигурация. При нажатии *Завершить* программа закроется, и мы автоматически перейдем к окну, где можно проверить соединения между элементами и физическими терминалами.

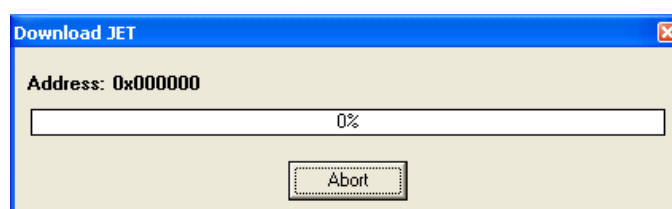
Для нашего случая необходимо связать DigitalIn и DigitalOut в нашем проекте с выбранным цифровым входом и цифровым выходом соответственно.





Выбираем DIGITALIN из списка слева и физический терминал, с которым мы хотим связать его, из списка в центре, например, с первым (для более подробной информации о физических терминалах обратитесь к документации оборудования). После нажатия *Соединить* выбранные объекты будут удалены из списков, в правом списке появится новая строка с описанием созданного соединения. При закрытии данного окна система автоматически запросит связать DIGITALOUT, выполните это тем же способом. После того, как все объекты в секции оборудования будут связаны, запустится компиляция проекта (результат компиляции будет показан в выведенном окне).

После этого можно загрузить программу в контроллер: для этого активируем в меню *Проект/Загрузка* (или нажимаем соответствующий значок на панели инструментов). Появится следующее окно:



Если соединение успешно установлено, через несколько секунд индикатор выполнения процесса начнет увеличиваться и, как только загрузка закончится, появится сообщение об успешном завершении операции. В противном случае появится сообщение об ошибке; в этом случае повторим попытку снова после проверки некоторых условий:

Последовательный порт, выбранный в опциях среды (который может быть активирован в меню через *Инструменты/Настройки*), должен являться портом, через который будет выполняться загрузка.


Кабель должен быть подключен к контроллеру и компьютеру.

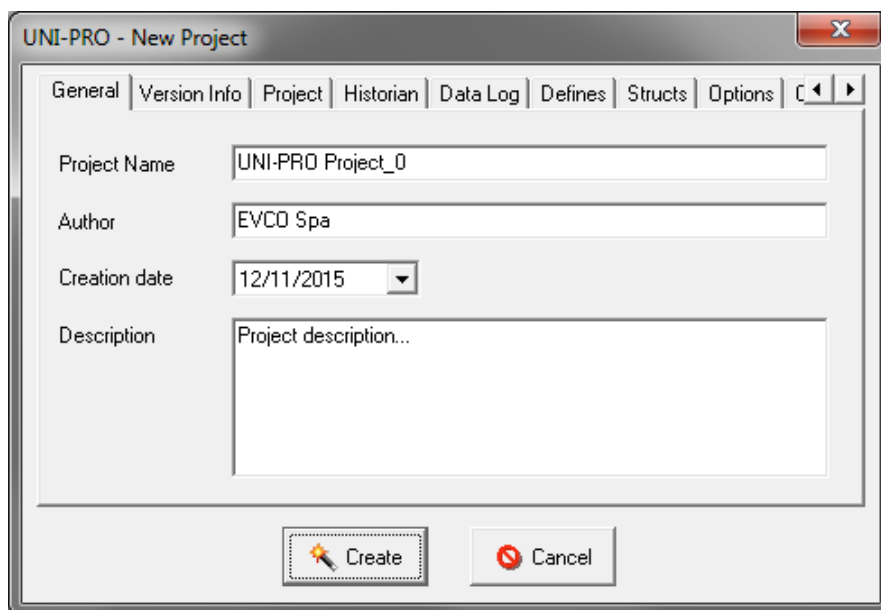
После завершения операции загрузки Ваша первая программа должна быть выгружена в ПЛК, если Вы замкнете контакт, с которым связан DigitalIn, реле, соответствующее DigitalOut, замкнется и графическая программа просмотра покажет сообщение "Hello World!".

## 2 ОСНОВНЫЕ ОПЕРАЦИИ


В данной главе будут описаны основные операции, позволяющие разработать проект с использованием среды программирования UNI-PRO. Пользователю будут показаны все операции, необходимые для разработки проекта (создание, разработка алгоритмов, компиляция и загрузка в контроллер).

### 2.1 Создание нового проекта

Первое, что нужно сделать для начала работы с UNI-PRO, - это создать новый проект: для этого выберите в меню *Файл/Новый проект* или нажмите значок *Создать новый проект*  на панели инструментов. Будет выведено окно, содержащее список всех свойств проекта.




При создании нового проекта рекомендуется всегда вводить его название, уникальный номер проекта, имя разработчика, дату создания (текущая дата появляется по умолчанию) и краткое описание проекта. Каждый проект может быть идентифицирован по номеру проекта, версии и модификации. Чтобы создать проект, нажмите кнопку *Создать* или кнопку *Отмена* для отмены.

Все опции проекта могут быть изменены в дальнейшем, для этого выберите пункт меню *Проект/Свойства* или нажмите соответствующую кнопку на панели инструментов .

### 2.2 Сохранение и загрузка существующего проекта

Сохраняя проект на диске и перезагружая его, вы можете разработать проект в несколько этапов и сохранять различные разработанные проекты.


Чтобы сохранить проект, сделайте одну из перечисленных процедур:

- Выберите пункт меню «Файл/Сохранить проект»;
- Нажмите значок  на панели инструментов;
- Нажмите Ctrl-S

Сохраняя проект в первый раз, Вам необходимо ввести название директории и имя целевого файла, впоследствии проект всегда будет сохраняться в том же файле.

Если Вы хотите сохранить проект под другим именем, выберите пункт меню *Файл/Сохранить как...*; таким образом, Вы сможете задать новое имя файла.

Чтобы загрузить проект, сохраненный ранее на диске, проделайте одну из перечисленных процедур:

- Выберите пункт меню *Файл/Открыть проект*;
- Нажмите значок  на панели инструментов;
- Нажмите **Ctrl-O**

Появится окно со списком файлов, из которого Вы сможете выбрать тот файл, в котором был сохранен проект.

Самый быстрый способ нахождения и открытия только что сохраненного проекта - это использование списка *Недавно открытые проекты*, в котором перечислены последние открытые проекты, сортированные по дате. Чтобы использовать данную функцию, выберите пункт меню *Файл/Недавно открытые проекты* и щелкните по нужному проекту.

## 2.3 Сущности: установка и использование

Сущности являются базовыми элементами метода разработки UNI-PRO, эти объекты графически представлены значком на листе разработки. Они имеют общие свойства:

- Переменное число входов, характеризующихся типом (см. параграф 3.1);
- Переменное число выходов, характеризующихся типом (см. параграф 3.1);
- Свойства *X* и *Y* (координаты левого верхнего угла), позволяющие определять свое графическое положение в пределах рабочего листа;
- Свойства *Ширина* и *Высота*, которые позволяют определять размер каждого объекта;
- Уникальное имя, которое определяет модуль в проекте;
- Краткое описание;

В дополнение каждая из сущностей имеет специальные свойства, которые характеризуют ее работу (они будут подробно описаны далее в главе 3).

Примерами сущностей являются переменные, цифровые входы (DI), цифровые выходы (DO), алгоритмы, библиотеки и т.д.

### 2.3.1 Имя сущности

Каждая введенная сущность должна обладать уникальным именем, которое характеризует ее в проекте; желательно, чтобы присваиваемое сущности имя удовлетворяло следующим критериям:

- Имена сущностей, алгоритмов, категорий, входов, выходов и т.д. не должны содержать более 80 символов;
- Компилятор проекта не чувствителен к регистру, поэтому имена двух различных сущностей не могут различаться только регистрами символов. Например, две сущности с именами *requestFans* и *RequestFANS* вызовут ошибку при компиляции;
- Имена сущностей должны соответствовать языку C, т.е. не должны содержать особые символы, такие как: пробел, запятая, точка, апостроф, штрих, кавычки и математические знаки (+, -, \*, /);
- Имена сущностей чувствительны к регистру.

Данные правила рекомендуется использовать при разработке во избежание ошибок во время фазы компиляции.

### 2.3.2 Введение сущностей

Чтобы поместить сущность на рабочий лист, необходимо подвести к панели, где он находится, курсор и нажать левую кнопку мыши, затем нажать на то место рабочего листа, где Вы хотите его расположить: сущность будет иметь размер, установленный по умолчанию, который Вы сможете впоследствии изменить.

### 2.3.3 Перемещение сущностей

Чтобы передвинуть сущность на другое место рабочего листа, сделайте одно из следующих действий:

- Щелкните левой кнопкой мыши по сущности и перетащите ее в нужное место (метод перетаскивания);
- Выделите одну или несколько сущностей и нажимайте кнопки со стрелками (чтобы ускорить передвижение, одновременно нажмите и удерживайте кнопку Ctrl).

### 2.3.4 Удаление сущностей

Чтобы удалить сущность с рабочего листа, сделайте одно из следующих действий: выделите ее и нажмите кнопку *Del*, выберите пункт меню *Правка/Удалить* или значок *Удалить* во всплывающем меню. Появится окно с запросом подтвердить удаление сущности. Эту операцию можно выполнять с несколькими выделенными сущностями.

Если Вы удаляете папку, все входящие в нее сущности также будут удалены.

В дополнение, когда сущность удалена, все соединения с другими сущностями и любые найденные ссылки на EIML-страницах также будут удалены.

### 2.3.5 Вырезание, копирование и вставка сущностей

Вы можете выполнять любые операции изменения (вырезать, копировать или вставить) с выбранными сущностями как с помощью комбинации кнопок (*Ctrl-x*, *Ctrl-c*, *Ctrl-v* соответственно), так и используя меню *Правка* или соответствующий значок (*Вырезать*, *Копировать*, *Вставить*) во всплывающем меню. Вырезание сущности схоже с операцией ее удаления. Кроме того, когда сущность вырезается, все соединения с другими сущностями и любые ссылки, расположенные на EIML-страницах, также будут удалены.



## 2.4 Использование листов управления

Лист управления является специальной сущностью, способной содержать другие сущности (включая другие листы). Благодаря использованию листов и дочерних листов Вы можете создать проект с древовидной структурой, применяя подход как Сверху-Вниз, так и Снизу-Вверх или комбинацию обоих в соответствии с Вашими предпочтениями и требованиями к рассматриваемому проекту.

Кроме того, лист управления позволяет группировать определенные сущности проекта в логическую макро-сущность. Если сохранить данный лист в качестве библиотеки, то Вы сможете использовать его впоследствии, извлекая из соответствующей панели.

Каждый проект состоит хотя бы из одного главного листа, в который Вы можете добавить как дочерние листы, так и сущности.

Существуют два различных подхода добавления папки в проект:

- Вы можете добавить пустую папку, выбрав значок  на панели программного обеспечения и затем поместить на него различные сущности;
- Или Вы можете автоматически передвинуть определенные сущности из активного листа в папку, сделав групповое выделение необходимых сущностей и выбрав пункт меню *Библиотека/Создать под-схему* или нажав значок  на панели инструментов

Чтобы соединить сущности, находящиеся в папке, с внешними сущностями, необходимо выполнить экспорт контактов входа и выхода задействованных сущностей (см. параграф 2.5).


## 2.5 Соединение сущностей управления

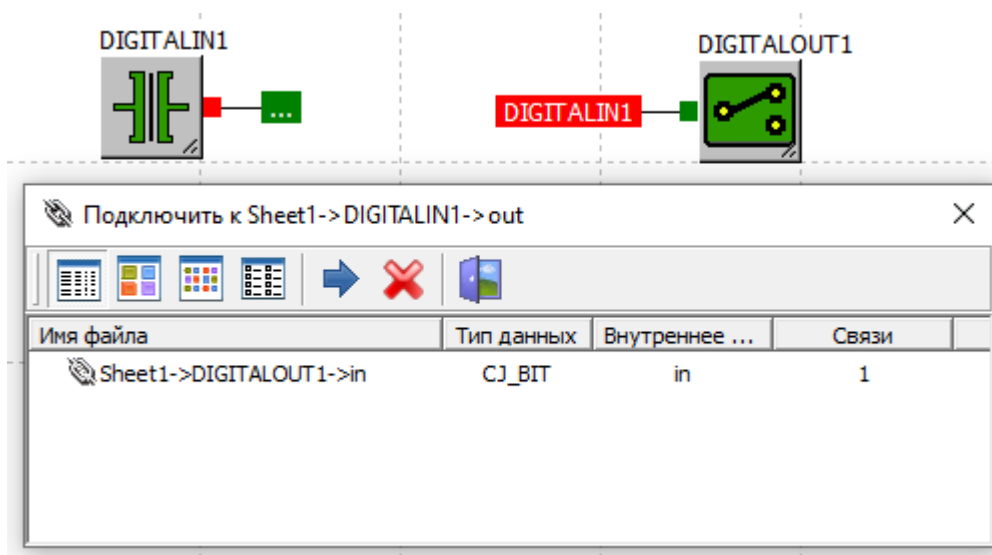
Чтобы подготовить проект к работе, необходимо поместить сущности на листы и соединить их вместе в соответствии с логической структурой, которая определяет работу контроллера.

Соединение сущностей означает связать каждый выход с соответствующими ему входами других сущностей, исходя из определенных правил ниже:

- Максимальное количество контактов одного выхода может быть подключено к контакту каждого входа;
- Тип данных, связанный с контактом входа, должен быть совместим с типом данных, связанным с контактом выхода, к которому вы намерены его присоединить.

**Чтобы соединить две сущности, сделайте одну из перечисленных ниже операций:**

- Если сущности находятся на одном листе, Вы можете соединить их с помощью стрелки: подведите курсор к контакту выхода сущности, который хотите соединить, и удерживайте до тех пор, пока он не подсветится; нажмите левую кнопку мыши и передвигайте курсор – между выделенным выходом и курсором появится стрелка. Если Вы хотите нарисовать соединение, состоящее из нескольких ломаных линий, просто щелкните левой кнопкой мыши на предполагаемом месте излома стрелки и продолжайте двигать мышью в другом направлении, в сторону контакта входа. Когда Вы подведете курсор мыши к контакту входа, он подсветится: щелкните левой кнопкой мыши, чтобы завершить соединение. Теперь, если вход не был соединен ранее и если типы входа и выхода совместимы, будет нарисована стрелка в подтверждение того, что сущности соединены. Если Вы хотите отменить эту операцию, щелкните в любое время правой кнопкой мыши.
- Если сущности расположены на различных листах, Вы можете **экспортировать** входы/выходы. Каждая папка позволяет экспортировать определенные контакты входов или выходов входящих в нее сущностей. Чтобы это осуществить, подведите курсор к контакту, который Вы хотите экспортировать, щелкните правой кнопкой мыши и выберите из всплывающего меню *Экспорт I/O*. Экспортируемый контакт меняет цвет (станет синим по умолчанию) и появится за пределами папки. Предположим, Вы хотите соединить цифровой вход с переменной, находящейся в папке: сначала следует экспортировать контакт входа переменной. После этого Вы можете связать контакт выхода цифрового входа с контактом, экспортированным из папки.
- Также возможно связывать входы и выходы (даже если они находятся на разных листах) без использования соединений стрелками, а при помощи внешних связей. Чтобы установить внешнюю связь, необходимо выполнить следующий порядок действий: нажмите кнопку  на панели программного обеспечения, выберите выход, который Вы хотите соединить, и затем выберите вход. Если во время соединения удерживать нажатой клавишу CTRL, выходной зажим, из которого исходит соединение, сохраняется в памяти; это облегчает соединение с другим входом, поскольку исходный выход не теряется и его поиск не требуется. Как только Вы закончите соединять сущности, можно вывести на дисплей только что созданные соединения, для этого выберите **Показать связи** во всплывающем меню входа или выхода...

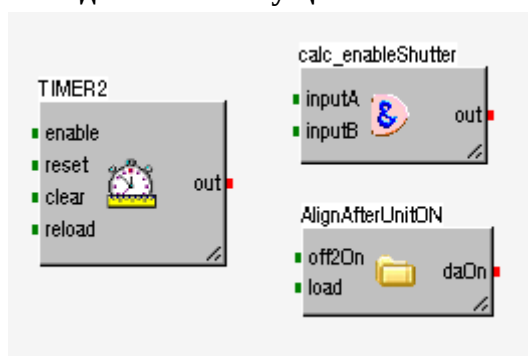


Эта функция весьма удобна, если связи довольно запутанны или их много, но делает проект менее удобным для чтения.

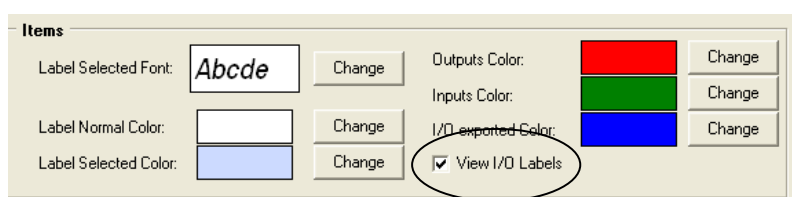
Чтобы разорвать связь между сущностями, можно действовать двумя различными способами в зависимости от типа связи:

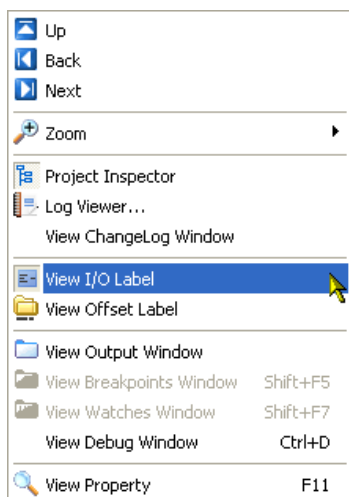
- Если сущности соединены стрелкой, выделите стрелку и нажмите кнопку Del на клавиатуре или выберите пункт меню *Правка/Удалить*.
- Если сущности были соединены с использованием внешней связи, необходимо открыть окно **Показать связи...**, затем выделить соединение и удалить его нажатием соответствующей кнопки.

### 2.5.1 Имена входных и выходных меток сущностей



Данную функцию можно активировать в пункте *Настройки*:

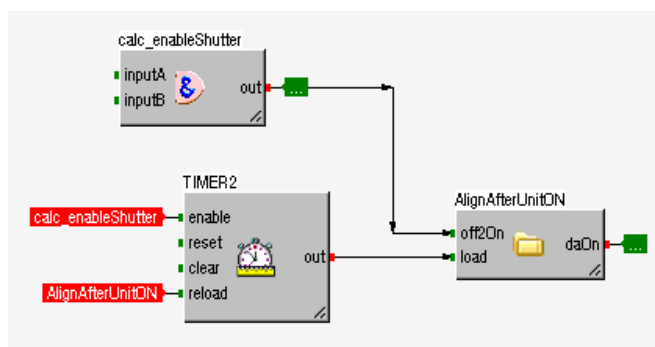





или в пункте меню Вид->Показать метки I/O:

**Примечание:** имена применимы к сущностям ТАЙМЕР, АЛГОРИТМ, БИБЛИОТЕКА и ЛИСТ в общем.

### 2.5.2 Имена внешних связей



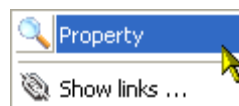
#### Выходное соединение

Выходные метки изображаются следующим образом:  Двойной щелчок на метке мгновенно откроет окно *Связи*.

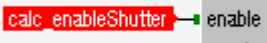
Выпадающее меню

Свойства: свойства связи

Показать связи: окно с имеющимися соединениями



#### Входное соединение

Входные зажимы изображаются следующим образом:  , в цветном прямоугольнике показано имя сущности, с которой они соединены; в этом случае "calc\_enableShutter". Двойной щелчок по зажиму вернет выделение соединенной сущности.

Выпадающее меню:

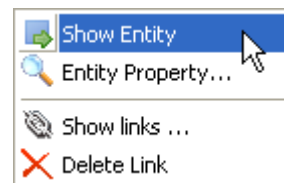
Показать сущность: выбирает соединенную сущность

Свойства сущности: открывает свойства соединенной сущности

Показать связи: открывает окно связей

Изъять связь: удаляет выбранную связь





(удаление также может быть выполнено с помощью SHIFT + Left кнопка мыши на вводе метки)



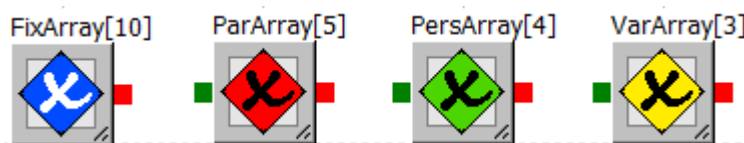
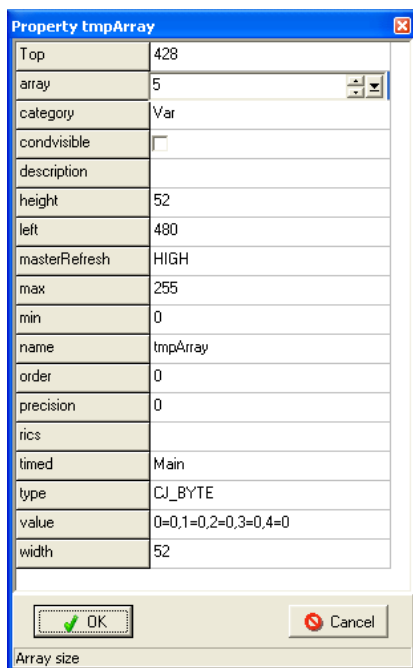


## 2.6 Использование сущностей-массивов

«Массив» - структура данных, содержащая N элементов числовой информации одного типа. Допустимы N массивов размерности N. Свойство «array» («массив») доступно только для следующих типов данных: CJ\_VOID, CJ\_BIT, CJ\_BYTE, CJ\_S\_BYTE, CJ\_SHORT, CJ\_WORD, CJ\_DWORD, CJ\_LONG. Массив является изменяемым свойством только для следующих сущностей:

-  PAR - Parameters
-  FIX - Constants
-  PERS - Persistent
-  VAR - Variables
- Входные и выходные данные алгоритмов.

Если тип данных отличен от перечисленных выше, свойство *массив* не подлежит модификации, а является фиксированным (только для чтения) со значением 1. Для всех остальных сущностей, не относящихся к PAR – FIX – PERS – VAR, свойство *массив* недоступно.




Показанное имя массива можно изменить (если свойство *массив* больше 1) путём добавления внутрь квадратной скобки размерности массива, как показано ниже.

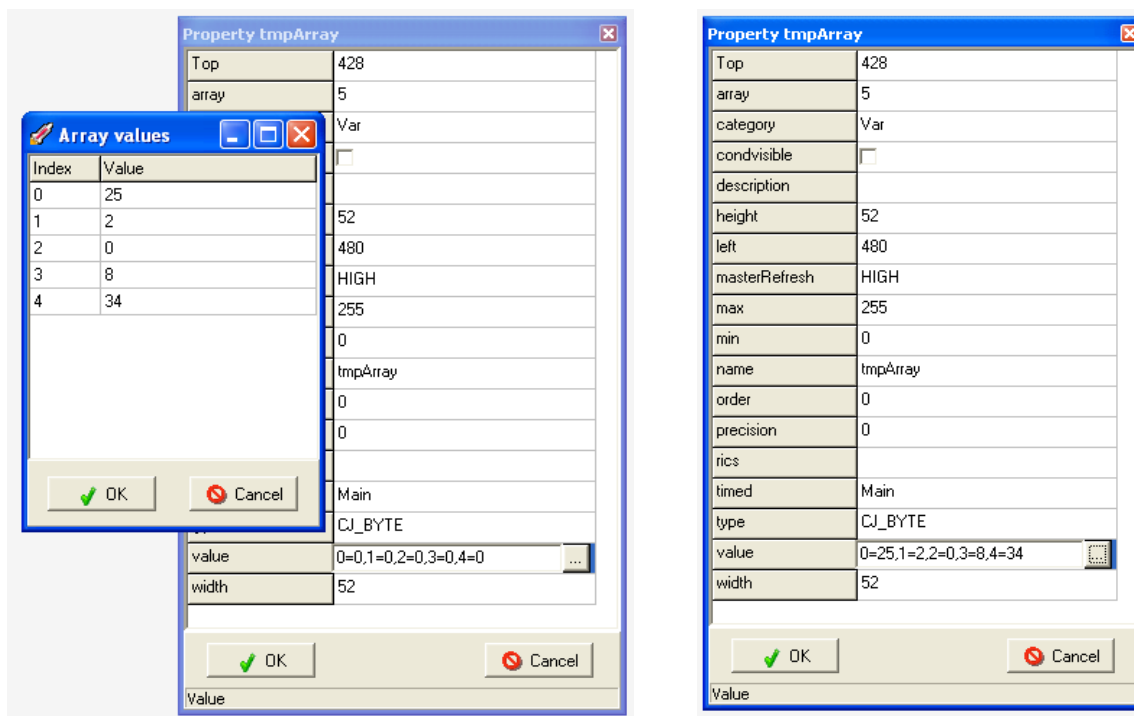
Массив является числовым типом с максимальным значением 100 и минимальным значением 1 (одномерная сущность): если число элементов массива превосходит единицу, то свойство *value* (значение) принимает вид строки, сопоставляющей индекс массива и соответствующее ему значение). Синтаксис выглядит следующим образом:

**0=value[0], ..., array-1=value[array-1]**

Также при открытии окна свойства с помощью данной строки можно быстро определить положение каждого элемента массива.

Все свойства значения, например, *min*, *max*, *precision* (минимум, максимум, точность) определяют пределы каждого числового значения массива: если в свойствах значения указано *min*=1, *max*=10 и *precision*=1, все значения будут не меньше 1 и не больше 10 и представлены с точностью в 1 десятичный знак.

Для «сущностей-массивов», как и для одномерных сущностей, можно задать значения по умолчанию. После щелчка по кнопке  рядом со свойством «value» («значение») будет открыто следующее окно для редактирования значений по умолчанию:



После нажатия на кнопку **ОК** изменённые значения будут приняты и автоматически внесены в свойство «value», как показано на картинке справа.

После нажатия на кнопку **Отмена** изменённые значения будут отклонены и строка «value» примет предыдущий вид.

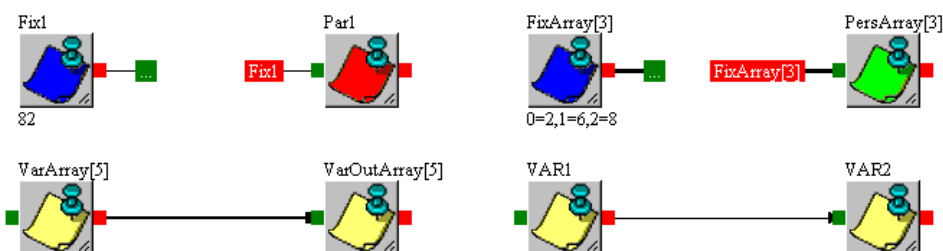
### 2.6.1 Совместимость и связи

Две сущности, у которых свойство *массив* больше 1, совместимы между собой при выполнении следующих двух условий:

1. если у обеих сущностей одна и та же размерность массива (свойства *массив* совпадают);
2. если тип данных обеих сущностей совпадает за исключением:
  - а) когда у начальной сущности тип данных CJ\_BIT, а у присоединенной - CJ\_S\_BYTE;
  - б) когда у начальной сущности тип данных CJ\_BIT, а у присоединенной - CJ\_BYTE.

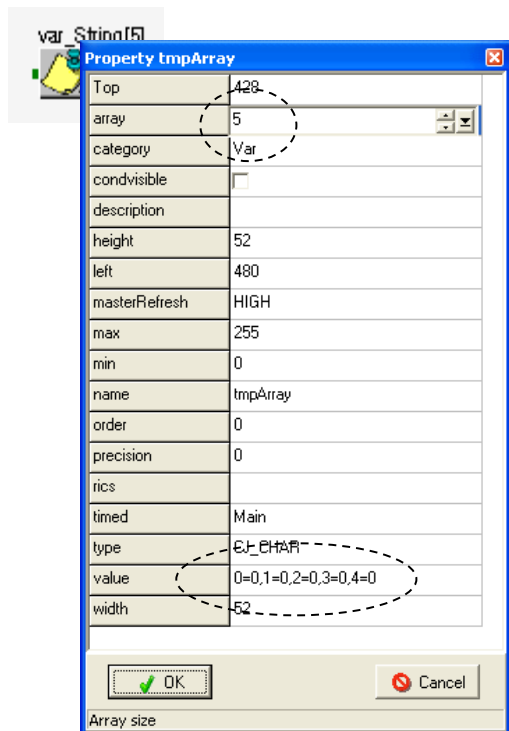
Во всех остальных случаях сущности несовместимы. При попытке соединить две несовместимые массивные сущности появится окно с сообщением об ошибке, связь будет разорвана.

Отличить связь между одномерными сущностями от связи между сущностями-массивами можно, изменив свойство *Array Segment Thickness* (*Толщина сегмента массива*) в окне *Settings/Настройка* (значение по умолчанию *Array Segment Thickness* = 2): в этом случае толщина линий, соединяющих сущности, будет разной.



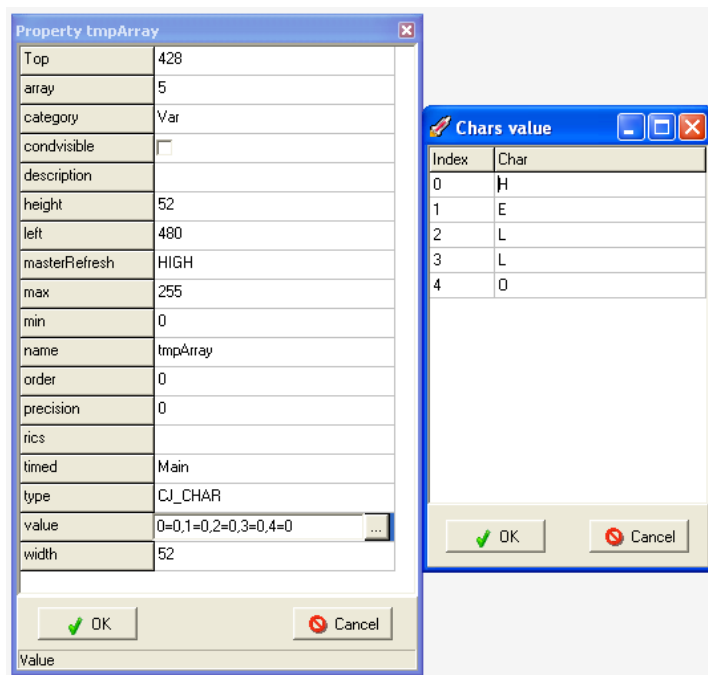
## 2.7 Тип CJ\_CHAR для управления строками

С помощью типа CJ\_CHAR и свойства *массив* можно управлять строкой длиной до 100 символов. Для использования этой функции необходимо установить тип сущности CJ\_CHAR, а в поле свойства *массив* – число символов строки.



Пример: данные настройки идентифицируют строку в 5 символов.

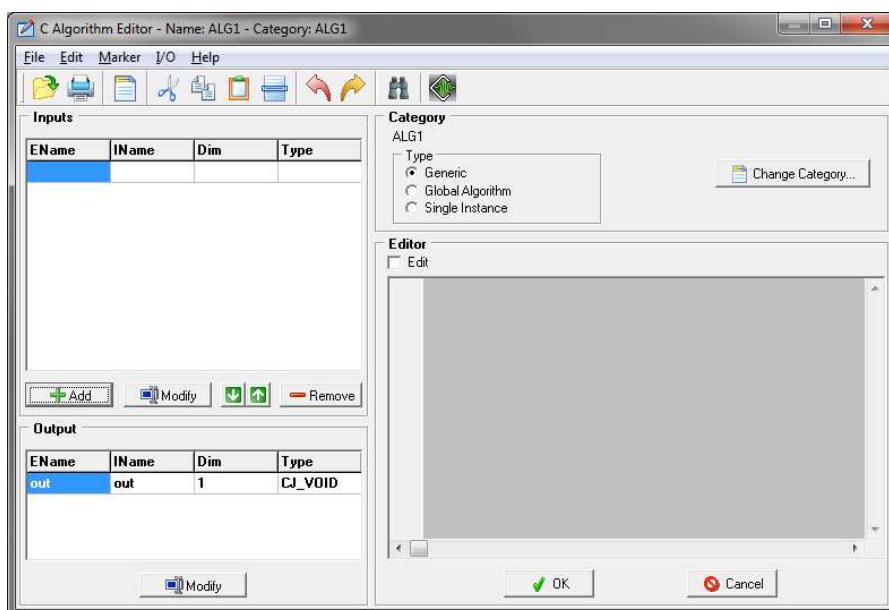
Для задания значений по умолчанию одиночных символов нажмите на кнопку, появляющуюся при выборе свойства *значение*: при этом на экране отобразится окно, идентичное окну управления значениями массива.



Как только поля будут заполнены, нажмите ОК; свойство *значение* будет обновлено.

## 2.8 Использование CodeEditor

CodeEditor является графической программой, которая определяет работу каждого алгоритма. Чтобы открыть программу, два раза щелкните левой кнопкой мыши по алгоритму или во всплывающем меню выберите *Show (Показать)*.



Алгоритм характеризуется серией входов, одним выходом и кодом, описывающим его работу. *CodeEditor* состоит из нескольких частей, благодаря которым Вы легко можете изменять все характеристики: в левой части окна *CodeEditor* представлены две таблицы, в которые сведены свойства входов и выхода; в верхней части окна представлена категория алгоритма, в центре – встроенный редактор.

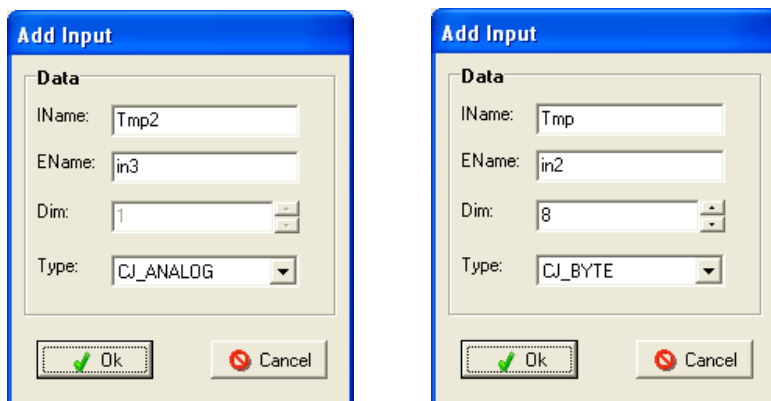
Весь проект также можно сохранить из CodeEdit (меню File), за исключением случаев изменения ввода-вывода.

### Редактировать Ввод/Вывод

В таблице ввода указаны их тип и количество; для каждого ввода задаются три свойства:

- **EName**, или внешнее имя, которое позволяет идентифицировать вход внутри листа;
- **IName**, внутреннее имя, т.е. имя, используемое в коде; оно может быть легко введено двойным нажатием левой кнопки мыши на строку;
- **Dim** определяет размерность входа или выхода (минимум 1, максимум 100). Если вход/выход является одномерной сущностью, то значение свойства равно 1. Если вход/выход является сущностью-массивом, то значение свойства *Dim* представляет собой число элементов массива. **Dim** является изменяемым свойством только для входов/выходов следующих типов: CJ\_VOID, CJ\_BIT, CJ\_BYTE, CJ\_S\_BYTE, CJ\_SHORT, CJ\_WORD, CJ\_DWORD, CJ\_LONG; для каждого другого типа данных свойству *Dim* присвоено значение 1.
- **Type**, т.е. тип обрабатываемых данных.

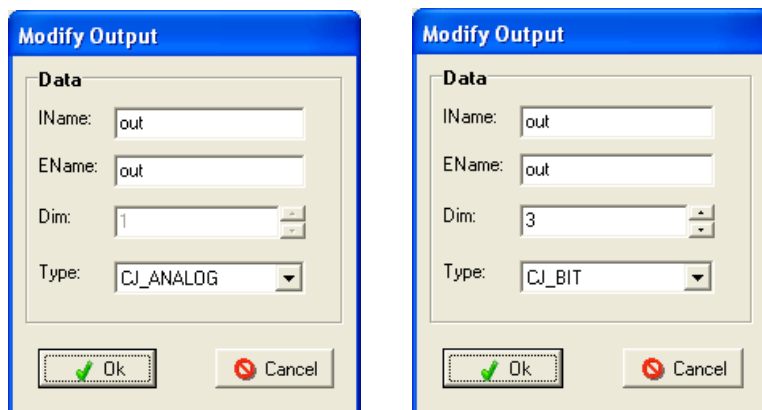
Если Вы хотите добавить новый вход, не закрывая при этом программу *CodeEditor*, нажмите кнопку *Add*; откроется окно:



в котором, Вы сможете ввести имя нового входа, его размерность и тип.

Если выделить вход и нажать кнопку *Modify* (Изменить), откроется похожее окно, в котором Вы сможете изменить имя и тип выбранного входа; нажимая кнопку *Remove*, можно удалить выбранный вход из алгоритма.

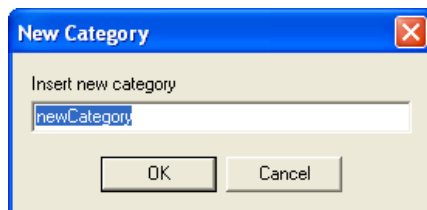
Таблица “*Modify output*” (Изменить выход) позволяет отображать и изменять выход алгоритма. Закономерно, выход алгоритма должен быть единственным. Для всех выходов действуют те же правила, что и для входов: выход также обладает свойством *Dim*. Размерность выходных данных - *Dim* – может меняться в пределах от 1 до 100 для следующих типов данных: CJ\_VOID, CJ\_BIT, CJ\_BYTE, CJ\_S\_BYTE, CJ\_SHORT, CJ\_WORD, CJ\_DWORD, CJ\_LONG.



В верхней части окна располагается **категория** алгоритма. Концепция категории схожа с концепцией класса в объектном программировании и позволяет использовать тот же исполняемый код для всех алгоритмов, входящих в одну и ту же категорию, тем самым, оптимизируя ресурсы. Два алгоритма одной категории должны обязательно иметь одинаковое количество и тип входов, выход и тот же код (см. раздел **Оптимизация**).

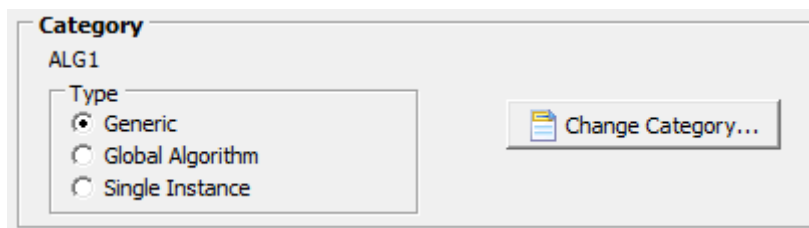
### Редактировать категорию

Нажмите *Change category...* (Изменить категорию), откроется следующее окно:



в котором Вы можете задать новую категорию.

Допускается изменение типа алгоритма.



- Базовый: универсальный алгоритм.
- Глобальный алгоритм: они определяют функции, которые могут вызываться несколько раз в коде других алгоритмов (см. 2.8.4).
- Один экземпляр: алгоритм, который можно использовать только один раз в проекте (полезно, когда он содержит некоторые уточнения, которые ни в коем случае нельзя повторять).

### Изменение кода

В центре окна находится часть редактора, в которой вы можете установить код алгоритма. Чтобы изменить или добавить код, отметьте ячейку *Edit*: когда вы захотите изменить алгоритм, система проверит наличие других алгоритмов в этой категории; если они присутствуют, система попросит изменить категорию или же внести те же изменения во все алгоритмы данной категории.

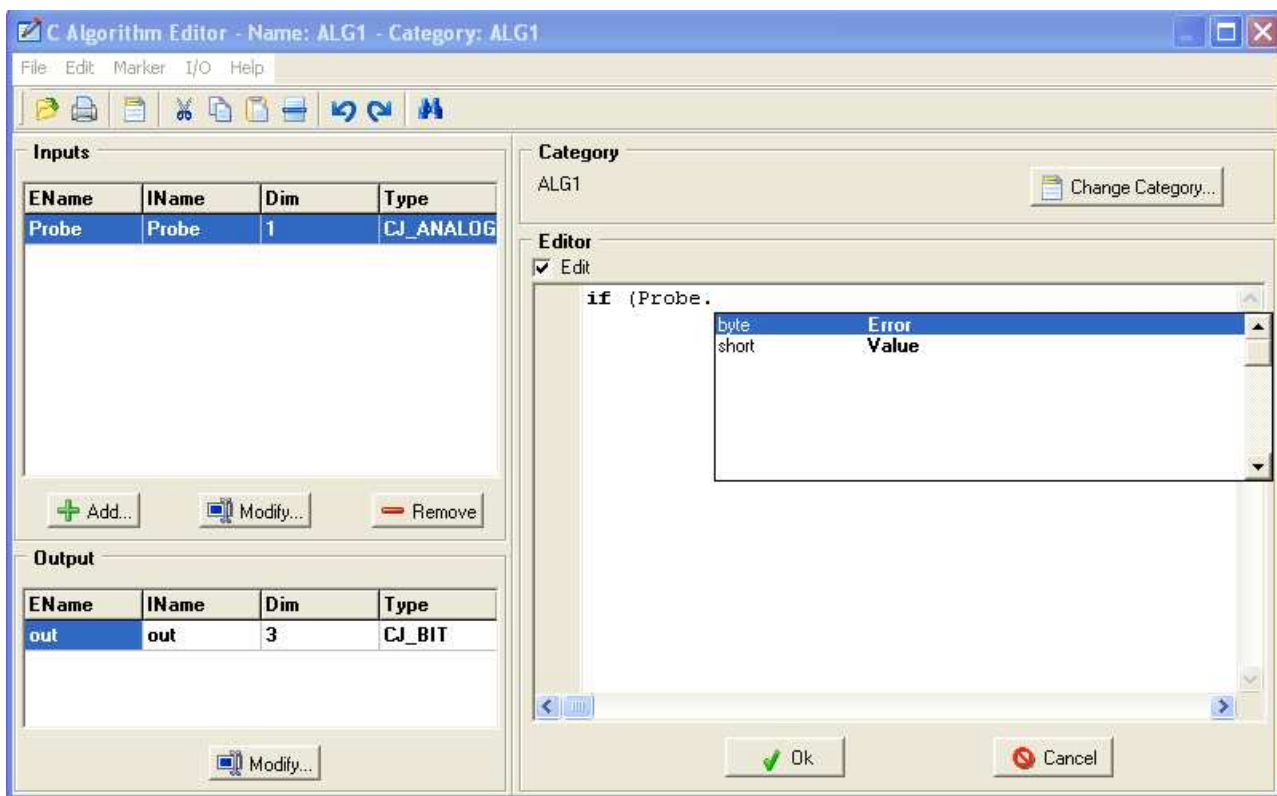
Редактор способен распознавать синтаксическую структуру ANSI C и обеспечивать некоторыми функциями для управления кодом алгоритма:

- Автоформирование кода;
- Считывание кода, подготовленного для использования различных редакторов файла;
- Внесение кода в файл;
- Распечатка кода;
- Поиск с применением Найти&Заменить;
- Вырезание, копирование и вставка частей кода;
- Использование закладок.

Функция автоформирования кода является хорошим помощником для разработчиков, так как позволяет сформировать синтаксис предопределённых структур или функций без постоянного обращения к документации.

В случае со структурами окно автоформирования будет появляться сразу, как будет поставлена точка перед именем поля структуры.

Например, если вы задаёте аналоговый вход как *Probe* (тип данных: *CJ\_ANALOG*) и хотите использовать поле значений внутри алгоритма, но не помните правильный синтаксис, Вы можете напечатать “*Probe.*” в коде и подождать пока всплывет окно автоформирования, как показано на картинке ниже:



При одновременном нажатии кнопок *Ctrl* и *пробел* откроется новое окно, показывающее необходимую информацию по редактированию кода, такую как:

- Функции и постоянные коэффициенты различных драйверов, которые могут быть использованы в алгоритмах;
- Глобальный алгоритм, определённый в проекте (прототип);
- Зарезервированные слова языка C;
- Типы данных, которые могут быть использованы;
- Имена входов в алгоритме;
- Определения проекта.

Нажимая одновременно *Ctrl-Shift-пробел*, все параметры функции будут показаны.

При введении кода настоятельно рекомендуется следовать определенным правилам:

- Для возвращения предыдущего значения всегда используйте синтаксическое зарезервированное слово языка C **return** (или программа-компилятор покажет предупреждение);
- Вы можете использовать статические переменные (т.е. присвоить атрибут **static** переменной);
- Рекомендуется использовать типы, входящие в систему UNI-PRO, как описано в параграфе 3.1;
- Обратитесь к структурным типам, если, например, у вас есть входы структурного типа (такие как CJ\_ANALOG);
- Будьте аккуратны при использовании циклов и избегайте большой вычислительной нагрузки или, что еще хуже, бесконечных циклов!

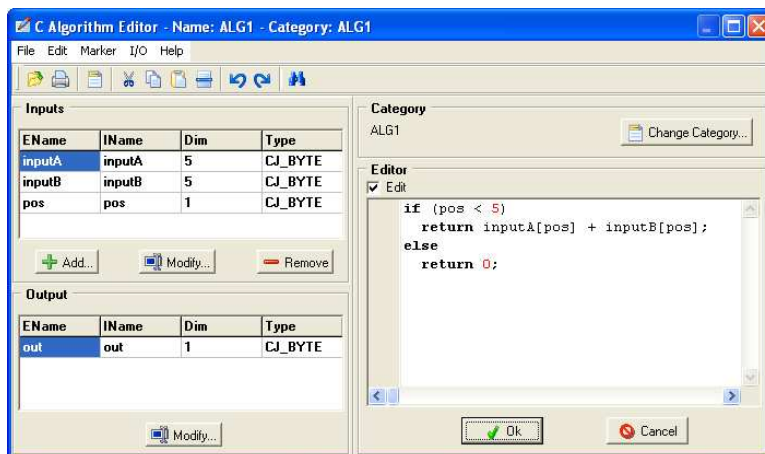
Код алгоритма может быть определен в любое время при условии, что это происходит до компиляции.

### 2.8.1 Алгоритм с использованием входов/выходов типа массив

Здесь представлены два простых примера корректного написания алгоритмов для входов и выхода типа массив на языке C с использованием UNI-PRO Code Editor.

#### Входы типа массив

Имея 2 входа-массива типа CJ\_BYTE, оба превышающие значение 1, мы возвращаемся к алгоритму суммирования значений, содержащихся в позиции *pos* обоих входов-массивов.



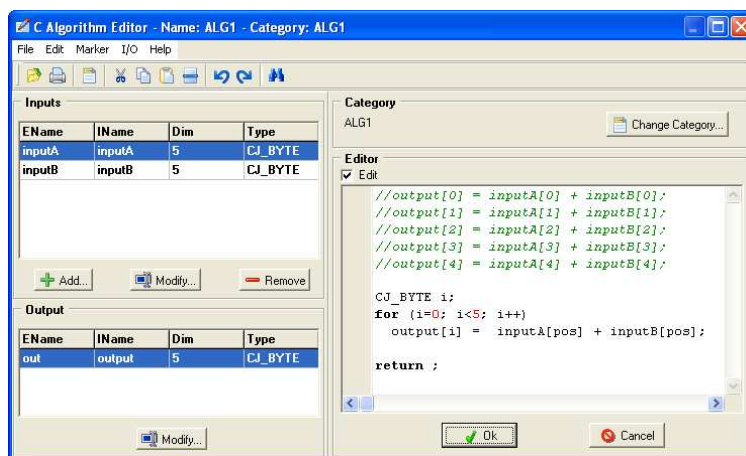
В данном примере два входа имеют одинаковую размерность массивов, но размерность может быть и разной. В каждом из массивов содержится 5 элементов числовой информации. Для доступа к каждой ячейке массива используйте следующий синтаксис на C:

***nameVar [ index ]***

*index* является индексом доступа к массиву, всегда начинается с 0; таким образом, **первому элементу массива соответствует *index 0*, а последнему элементу массива - *index Dim-1*.**

### 2.8.2 Входы и выход типа массив

Здесь приведён более сложный вариант предыдущего примера. Сейчас мы хотим сложить каждые 2 элемента массивов-входов с одинаковым индексом и вывести результирующее значение в массив-выход.



Для оптимизации кода используйте «цикл **for**» для доступа ко всем индексам массива без указания для каждого индекса необходимой операции.



Переменная выхода Алгоритма с именем *output*, в отличие от алгоритмов с одно-размерным выходом, напрямую используется в коде. Выход-массив используется, как если бы он являлся входным параметром, передаваемым в качестве *ссылки* (с учётом правил компьютерного программирования); изменение выхода и доступ к нему производятся непосредственно с самим выходом, а не копией в виде временной переменной.

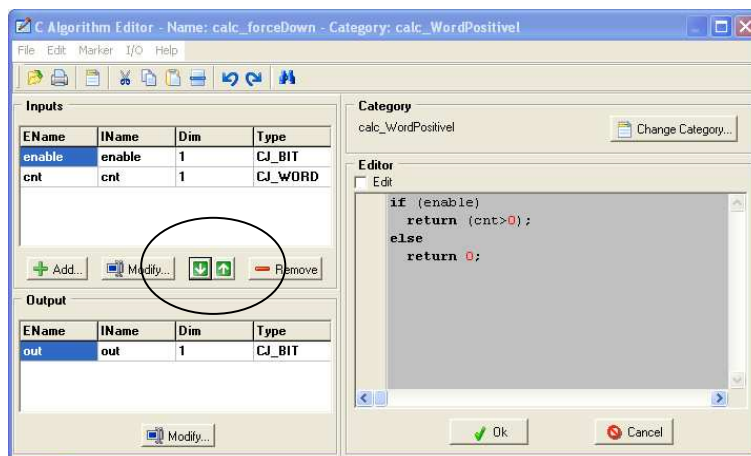
Обратите внимание: ключевое слово **return** следует писать следующим образом: “**return;** “. Как сказано ранее, оно не вовлечено в обработку, но должно быть там упомянуто: в противном случае компилятор выдаст *Предупреждение (Warning)* в ходе компиляции проекта.

Когда выходом алгоритма является сущность-массив, запомните следующие правила:

- выход должен напрямую использоваться при редактировании кода на C;
- имя используемого выхода должно совпадать с именем, установленным в свойстве **IName**;
- для доступа к индексу *i* массива используйте синтаксис *nameVar [ i ]*;
- допустимый индекс массива лежит в пределах **0** и **Dim -1**;
- используйте **return;** инструкция в конце алгоритма для избежания *Предупреждения (Warning)* в ходе компиляции проекта.

### 2.8.3 Изменение порядка входных терминалов алгоритмов

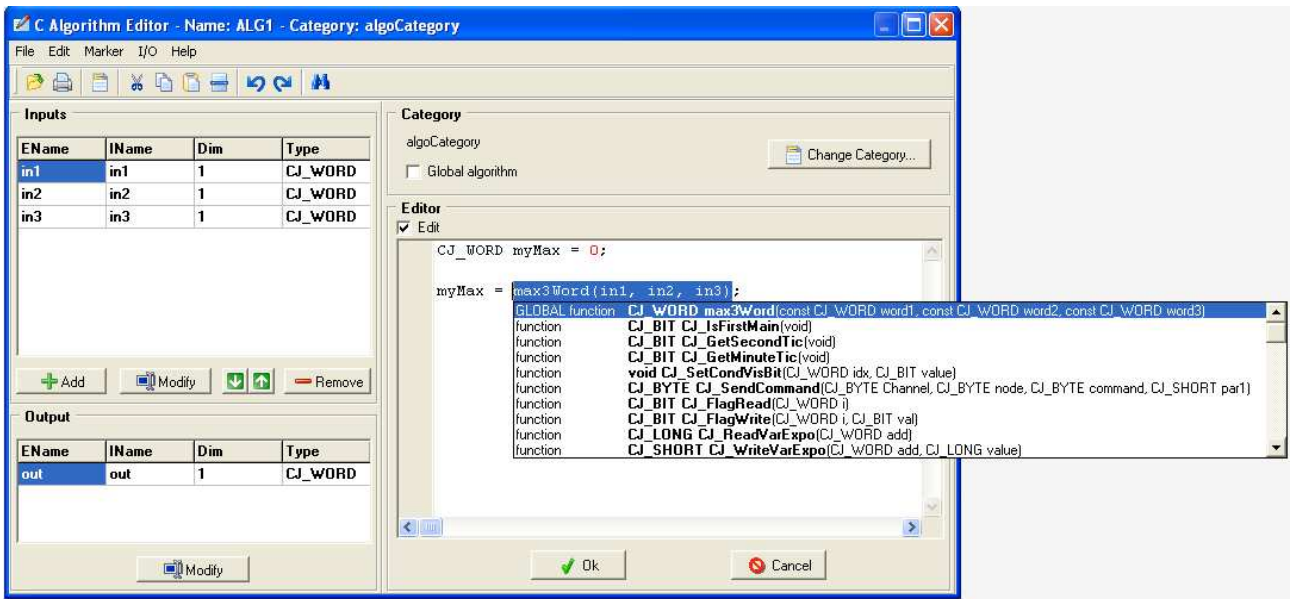
Для изменения порядка входных терминалов алгоритма без удаления связей в «Code Editor» предусмотрены специальные кнопки-стрелки.



**Примечание:** в настоящий момент данную функцию нельзя использовать при наличии экспортируемых входов алгоритма.

### 2.8.4 Использование глобальных алгоритмов

Глобальный алгоритм может быть использован в других алгоритмах при соединении с именем категории и передаче ей списка параметров (с помощью соответствующего порядка и типа). На рисунке приведён пример глобального алгоритма с категорией “*max3Word*”, дающего на выходе максимум из 3 вводных данных. Если Вы хотите использовать его в других алгоритмах, то Вам следует выполнить с соединением с категорией “*max3Word*” (не с именем сущности!) и передать ей 3 параметра типа *word* в соответствующем порядке.



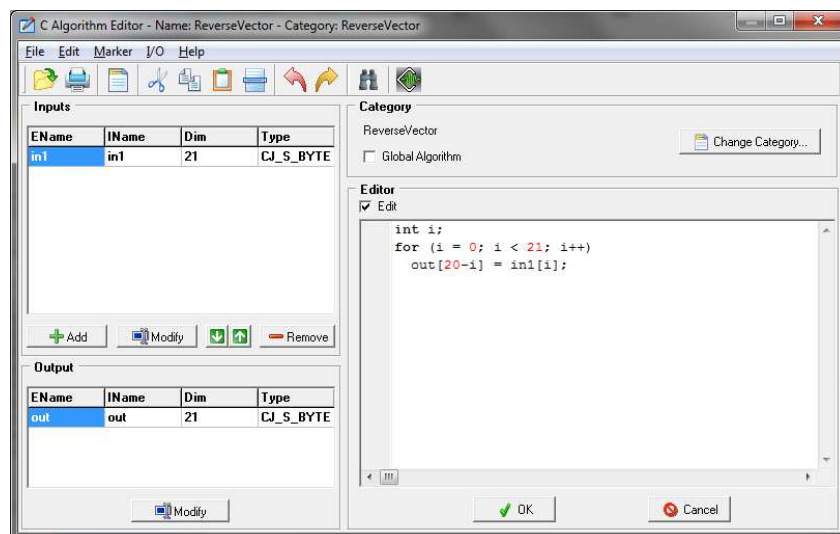
Для корректного вызова глобального алгоритма Вам следует найти его в проекте либо – более простым путём – в списке функций раздела *GLOBAL function* с помощью клавиш Ctrl+Space.

## 2.9 Симулятор алгоритмов

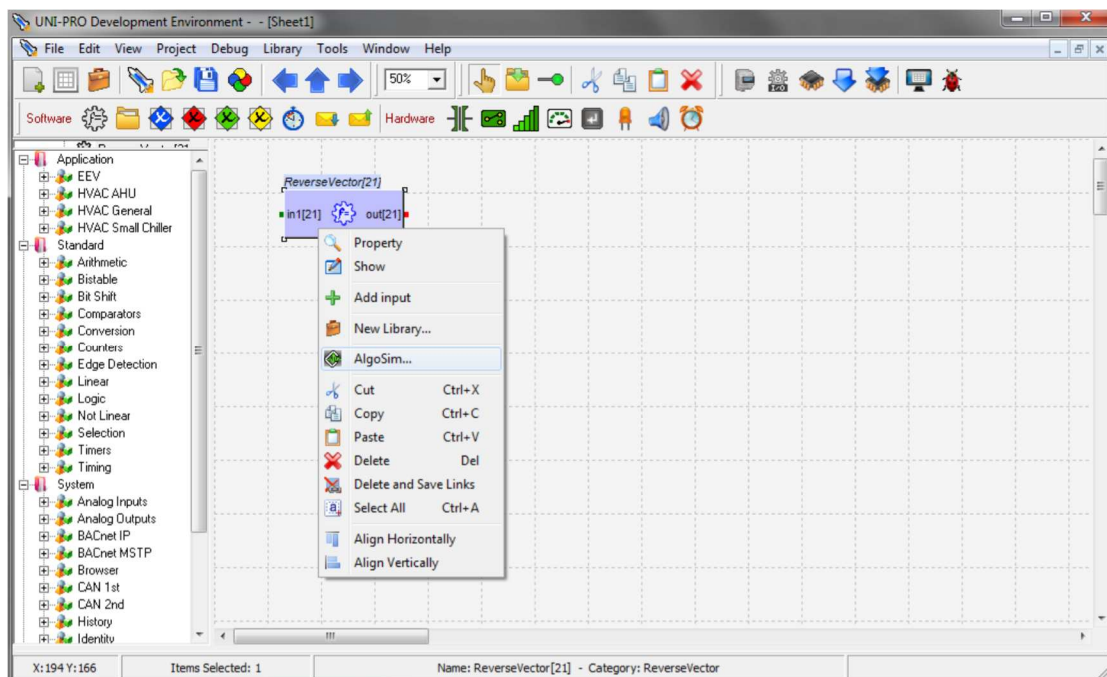
Сложность алгоритма зависит от длины его кода, используемых математических выражений, количества входных переменных. Выяснить производительность и конечный результат может быть сложно. *AlgoSim* — это инструмент, который помогает в этой задаче, инструмент, который позволяет моделировать процесс через аппаратное обеспечение с произвольными значениями входных переменных.


Самый практичный способ изучить работу на примере.

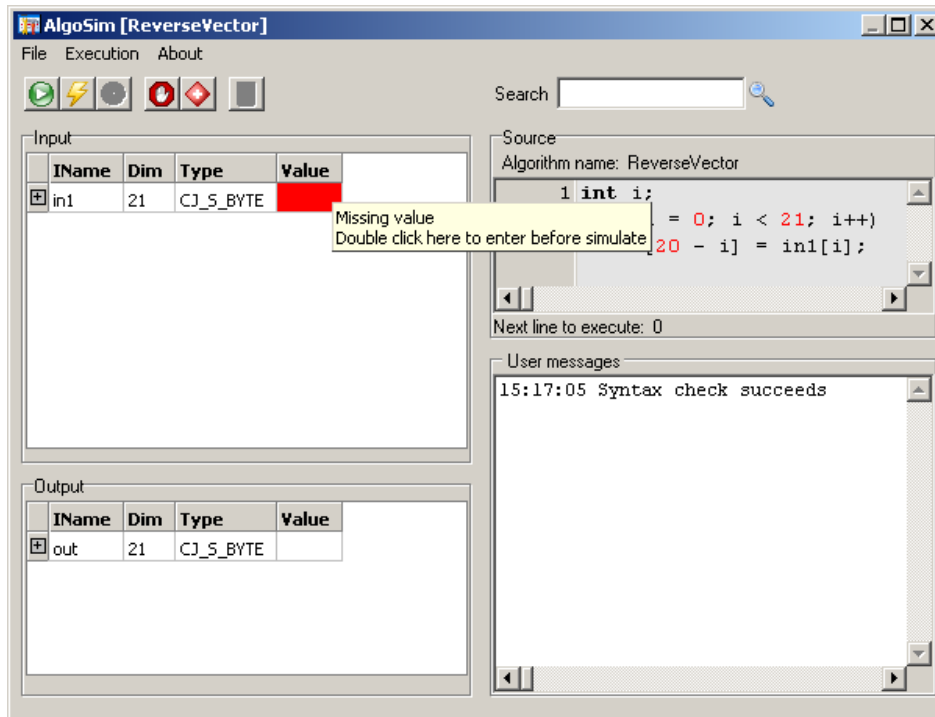
Предлагается выполнить алгоритм, названный *ReverseVector*, который принимает на вход массив *in1*, меняет положение значений с начала в конец массива, сохраняет результат в обратной переменной того же типа и размера.



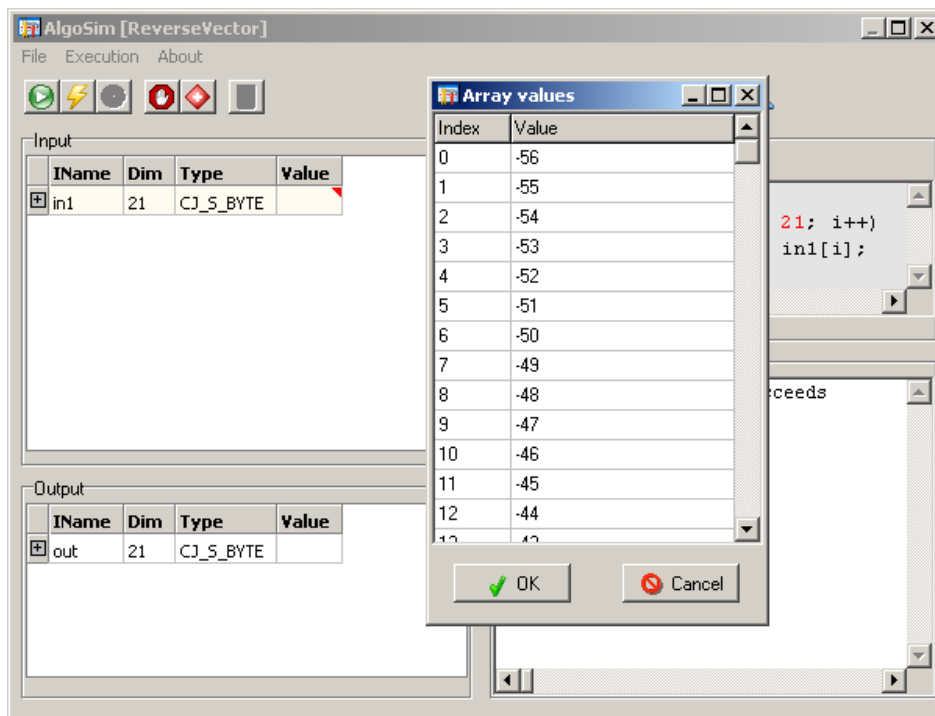
После создания и подтверждения щелкните правой кнопкой мыши на символ алгоритма и выберите *AlgoSim* во всплывающем меню.




Окно симулятора похоже на окно CodeEditor: входные данные, возвращаемое значение и код. При нажатии кнопки Выполнить  поле, соответствующее значению in1, мигает красным, а заголовок указывает, что оно пусто.




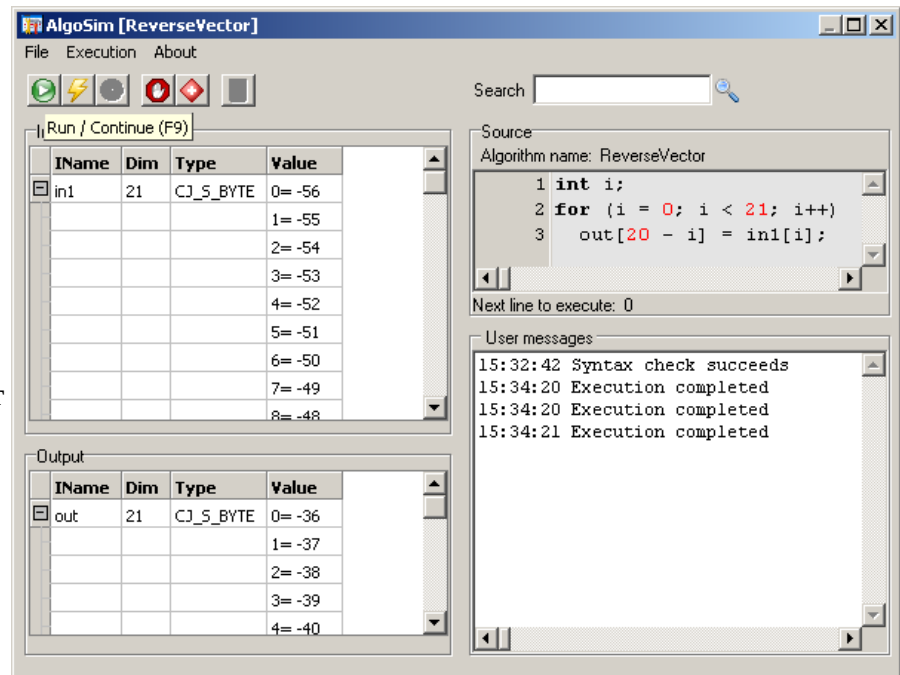
Дважды щелкните поле и заполните таблицу нужными значениями.



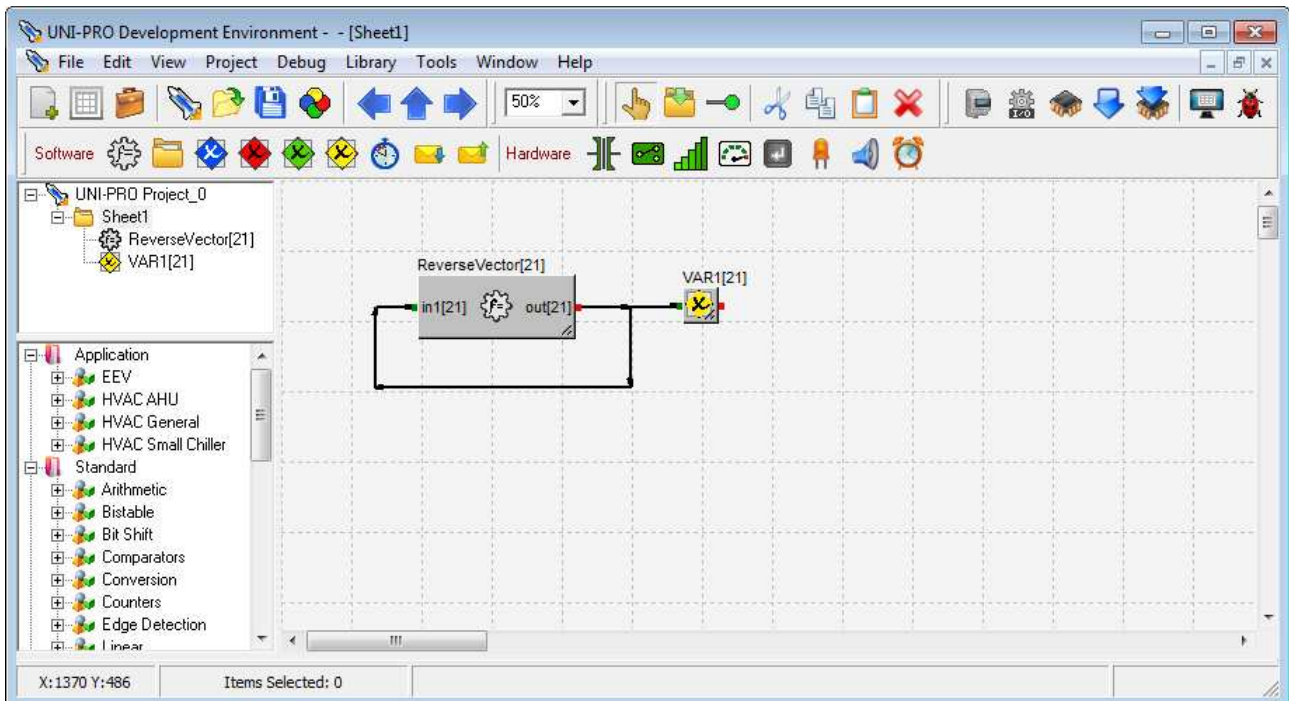
Для проведения симуляции важно, чтобы все входные данные имели значение.


Нажмите на кнопку Запустить , чтобы симулятор обработал все инструкции сразу. Результат появится на панели вывода.

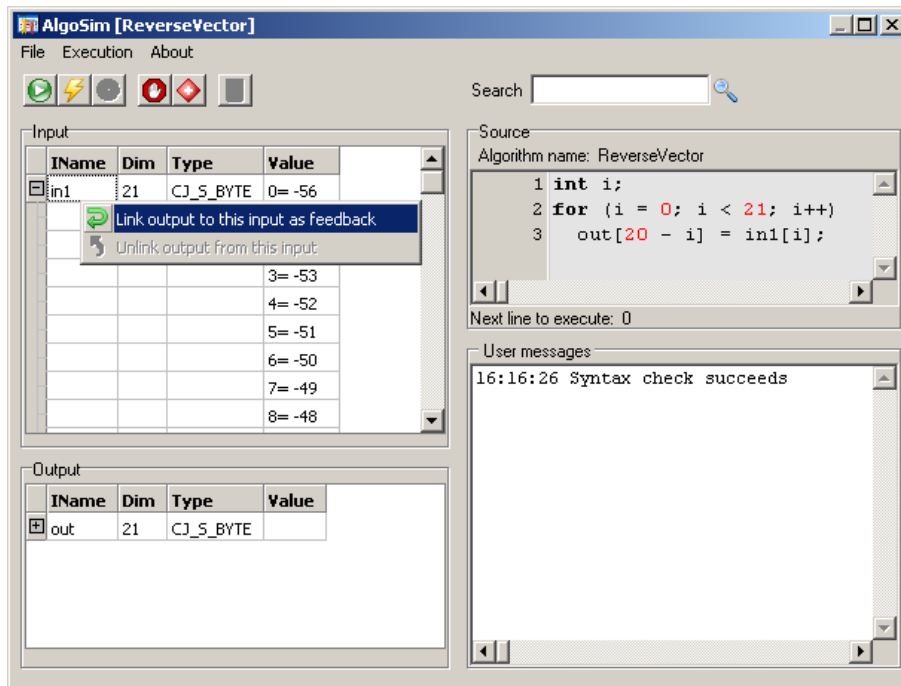
Если несколько раз нажать Run , результат не изменится, поскольку вектор in1 всегда один и тот же.



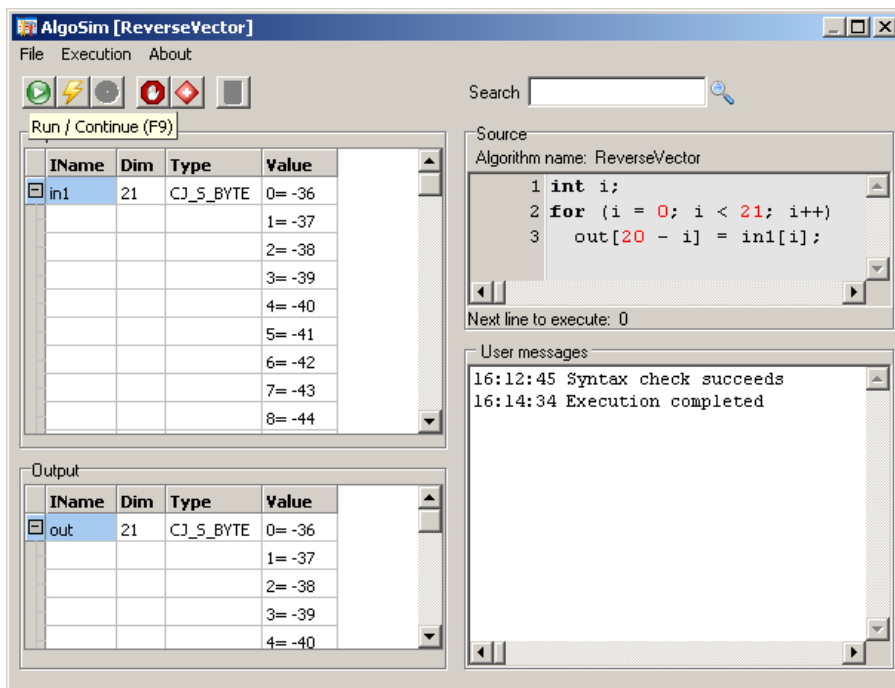
Предположим, мы хотим переложить массив сначала в одну сторону, а затем в другую. Если один из входов и выход имеют одинаковый тип и размер, вы можете связать их вместе, чтобы выход был доступен для входа до следующего цикла. Мы говорим об обратной связи.



Щелкните правой кнопкой мыши входной параметр. Действие Связать выход с этим входом в качестве обратной связи  включено, потому что вход удовлетворяет двум установленным требованиям: типу и размеру.



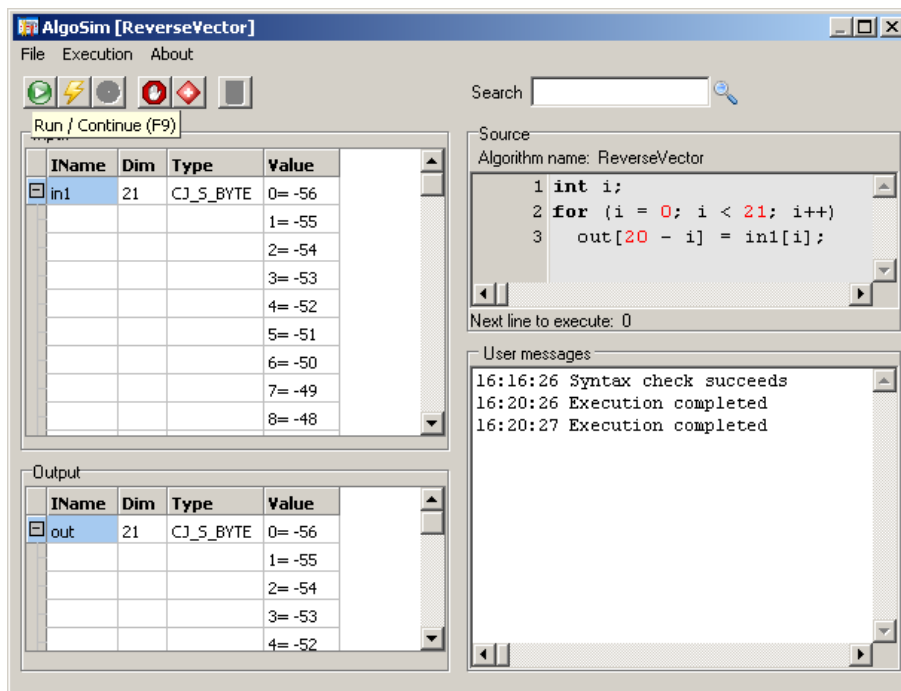
Вход и выход теперь связаны, выделены синим цветом.



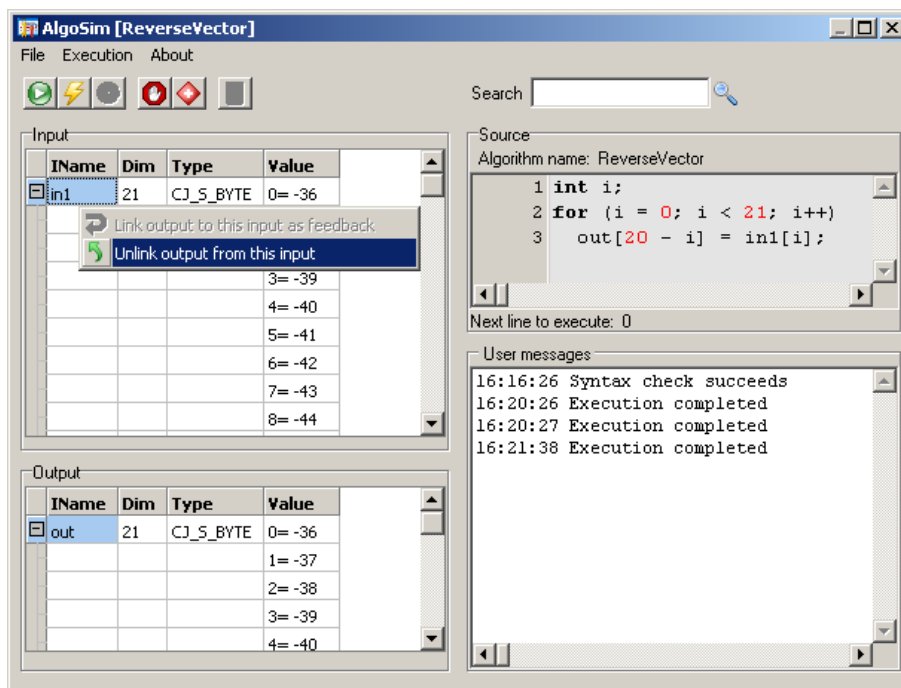
Симулятор загружает указанные значения одно за другим, выполняет код C, сохраняет результат в переменной *out* и, наконец, копирует его во вход, подключенный к выходу для следующего запуска.

Сначала вектор *in1* содержит число -56 -55 -54 и т.д., алгоритм переворачивает его в ряду -36 -37 -38, копируя его в *out*. Окончательно, вектор *in1* перезаписывается вектором *out*. Все делается всего в один клик.

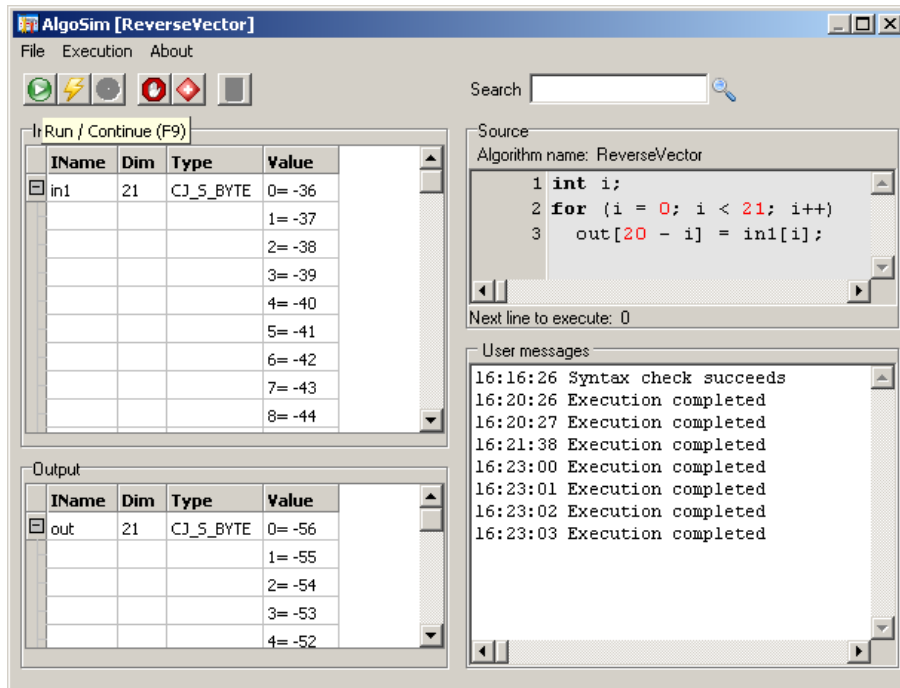
При повторном запуске как in1, так и out содержат исходный ряд.



Чтобы остановить обратную связь in1, просто щелкните правой кнопкой мыши параметр in1 и выберите Unlink output from the input. 🖱️

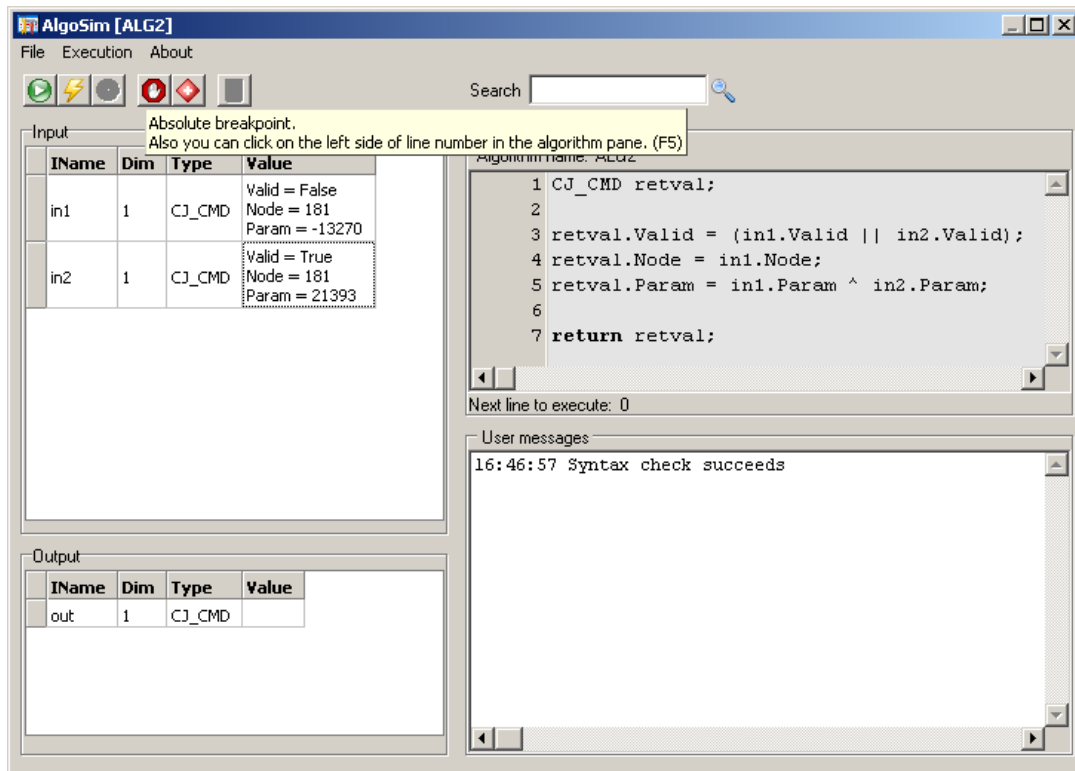


Без обратной связи при многократном выполнении алгоритма результат не меняется, потому что вход неизменен.



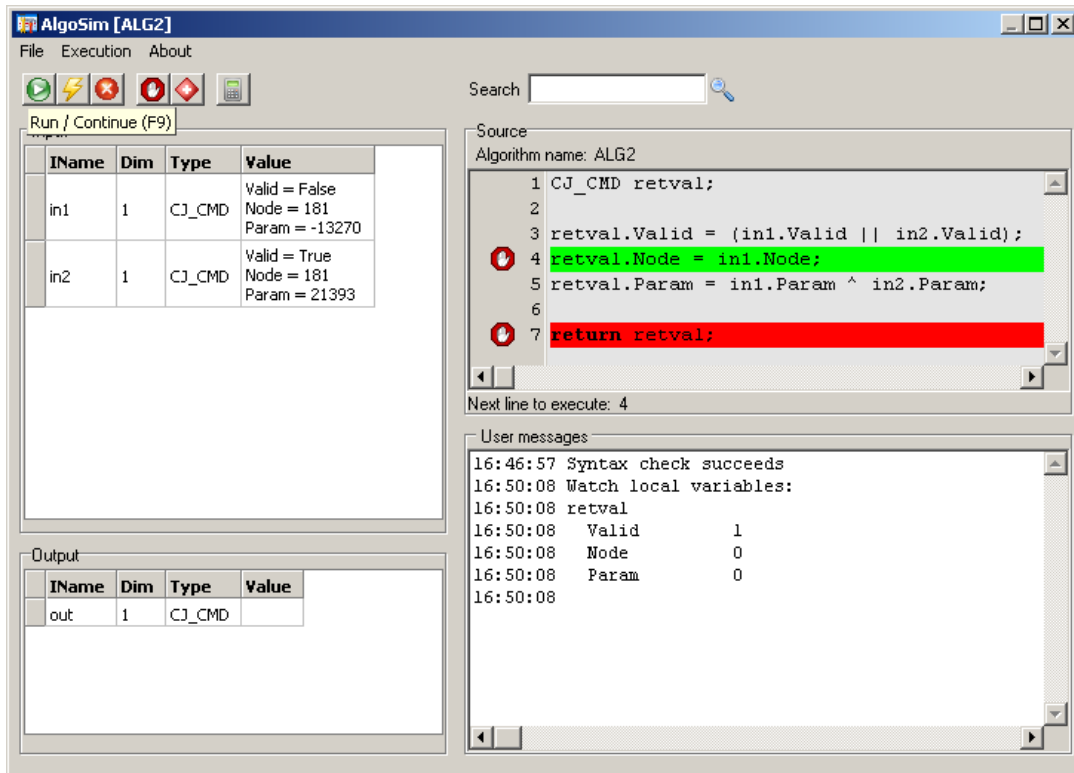
### 2.9.1 Абсолютная точка останова


Указывает строку кода, на которой выполнение должно быть остановлено независимо от каких-либо факторов.

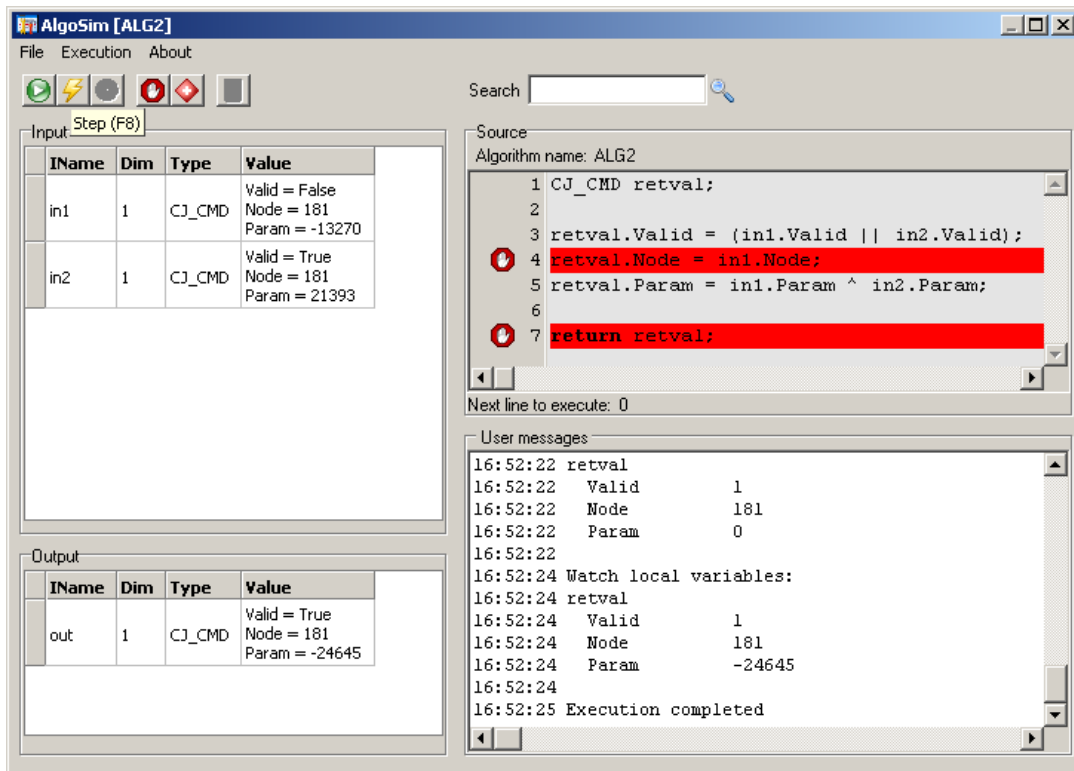




Самый быстрый способ установить абсолютную точку останова — щелкнуть по левому краю панели, содержащей код, или слева от номеров строк.




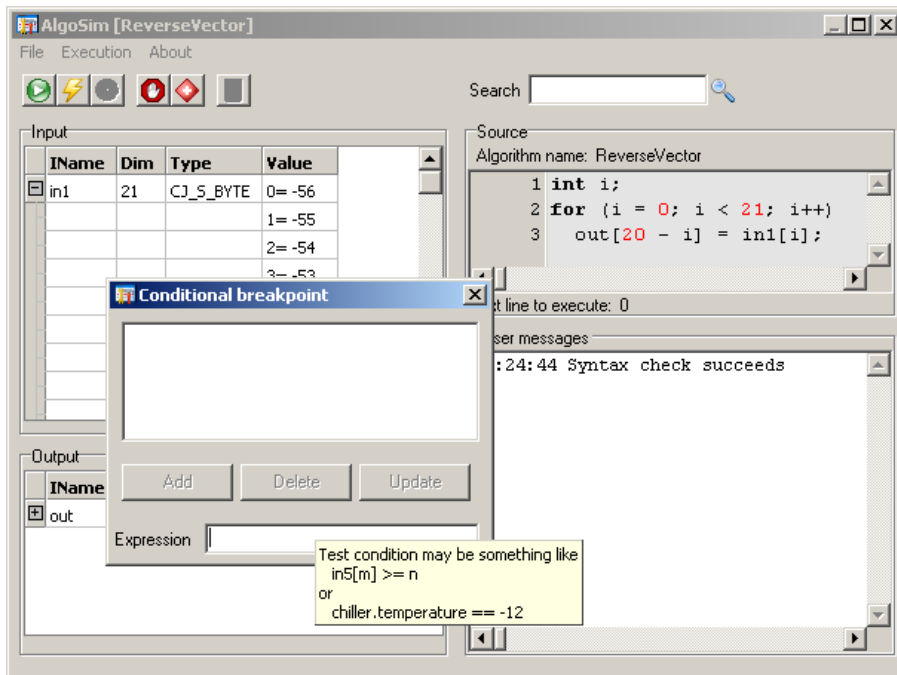
Либо щелкните нужную строку или переместите мигающий курсор внутрь поля, а затем нажмите кнопку  на панели инструментов.



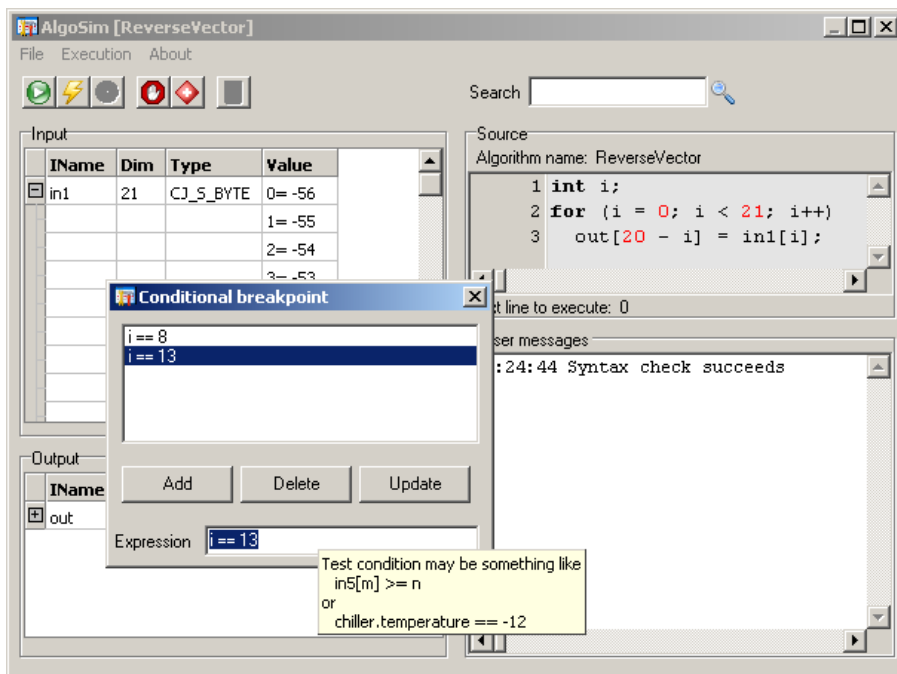
### 2.9.2 Условная точка останова

Предполагается прекратить работу, когда выполняется определенное условие.

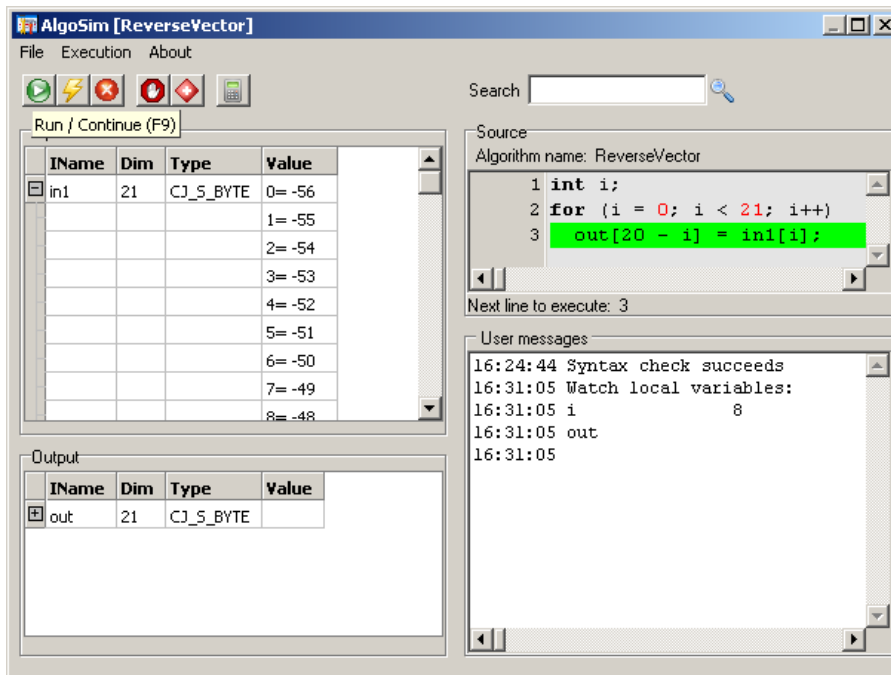
При нажатии на кнопку  появляется всплывающее окно, с помощью которого мы можем управлять всеми условиями, которые могут возникнуть во время выполнения алгоритма.




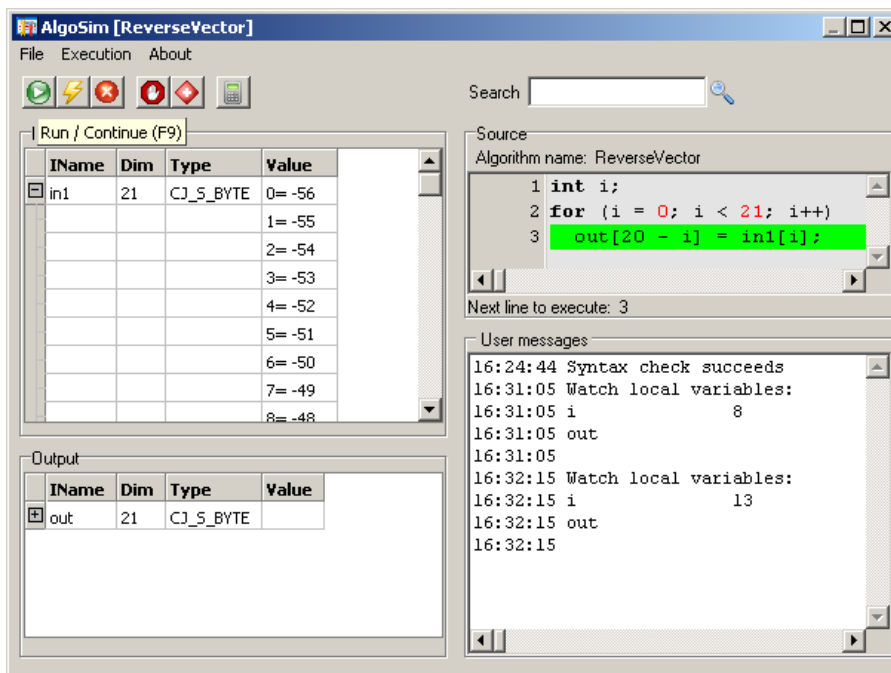
Как следует из заголовка, условие выражается на языке C как классический оператор if-then-else.



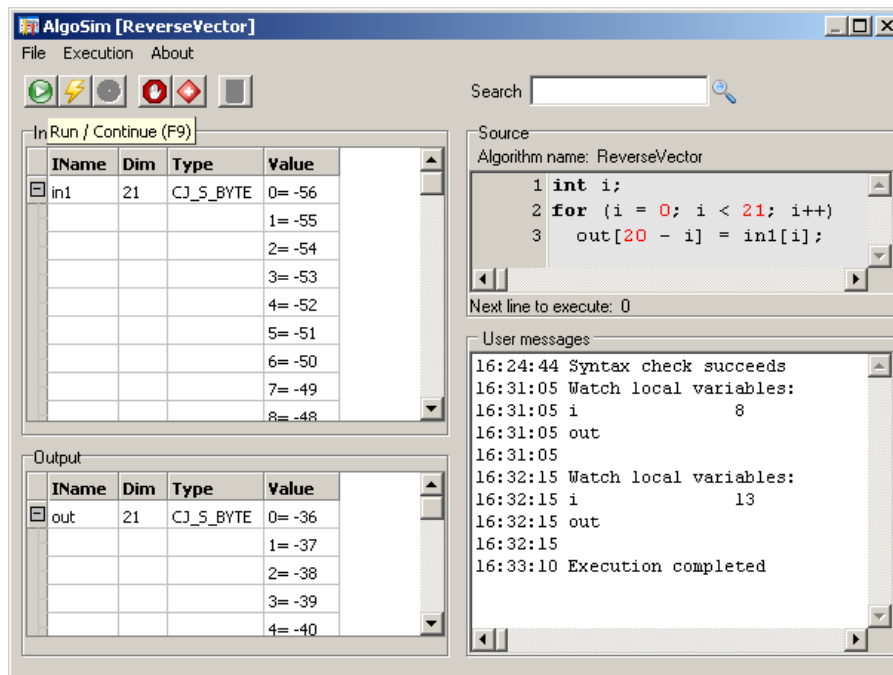
В этом примере при нажатии на кнопку *Run*  программа выполняет 8 итераций перед остановкой и ожидает действий от пользователя.




Нажав еще раз *Run* , программа завершает еще 5 итераций, от 8 до 13, прежде чем остановиться и дожидаться действий от пользователя.



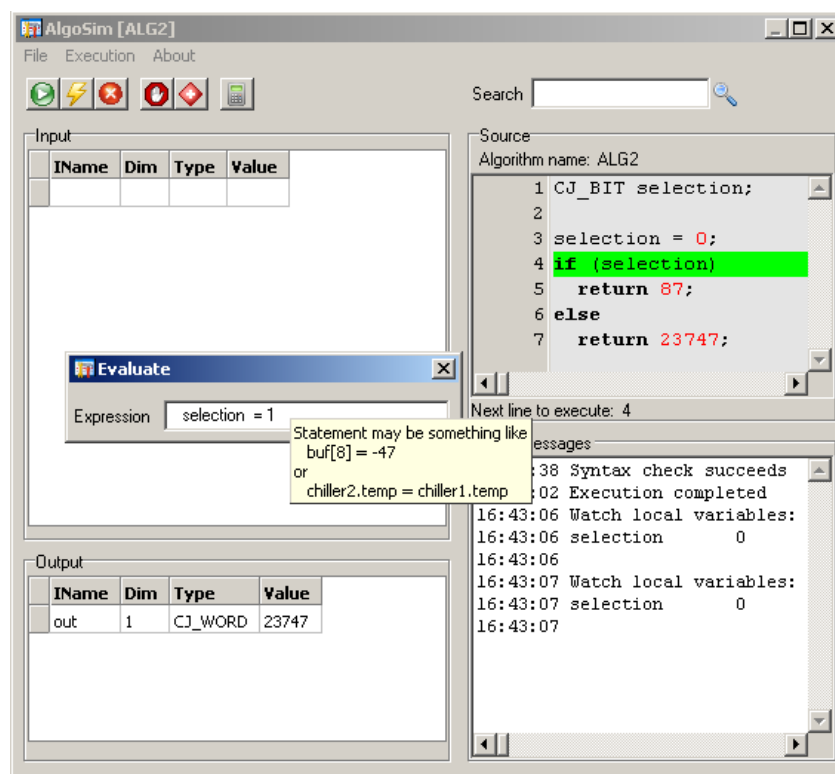
Нажав кнопку *Run*  в третий раз, программа завершит оставшиеся итерации, показав результат.




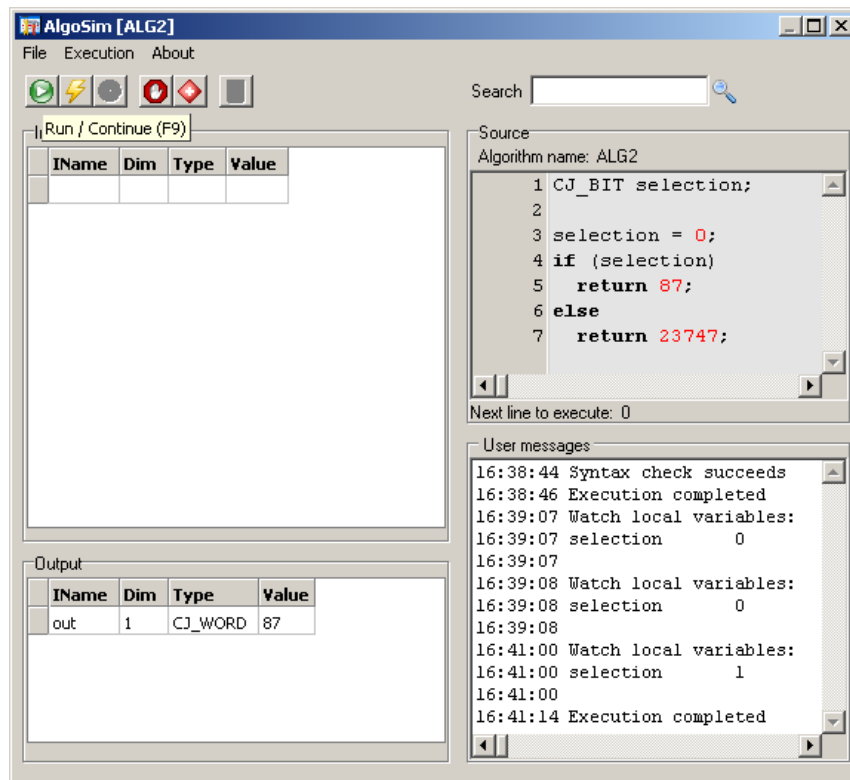
### 2.9.3 Установочное значение


Предполагается выполнять одну инструкцию за раз с помощью кнопки *Step*  на панели инструментов. Предположим также, что необходимо изменить на лету значение локальной переменной, чтобы попробовать другое направление кода.

В приведенном ниже примере алгоритм всегда возвращает 23747.



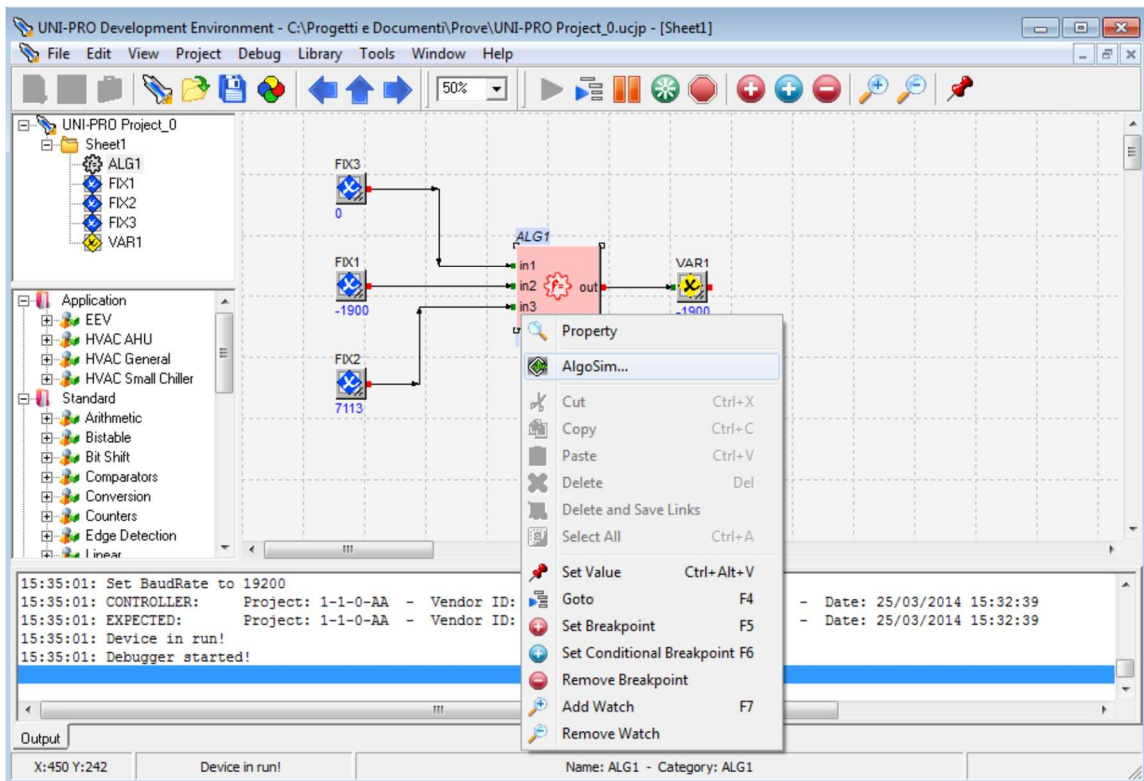
После того, как программа присвоит 0 выбору переменной, нажмите кнопку  и измените значение с помощью всплывающего окна. Назначение производится на языке C без знака ; в конце.



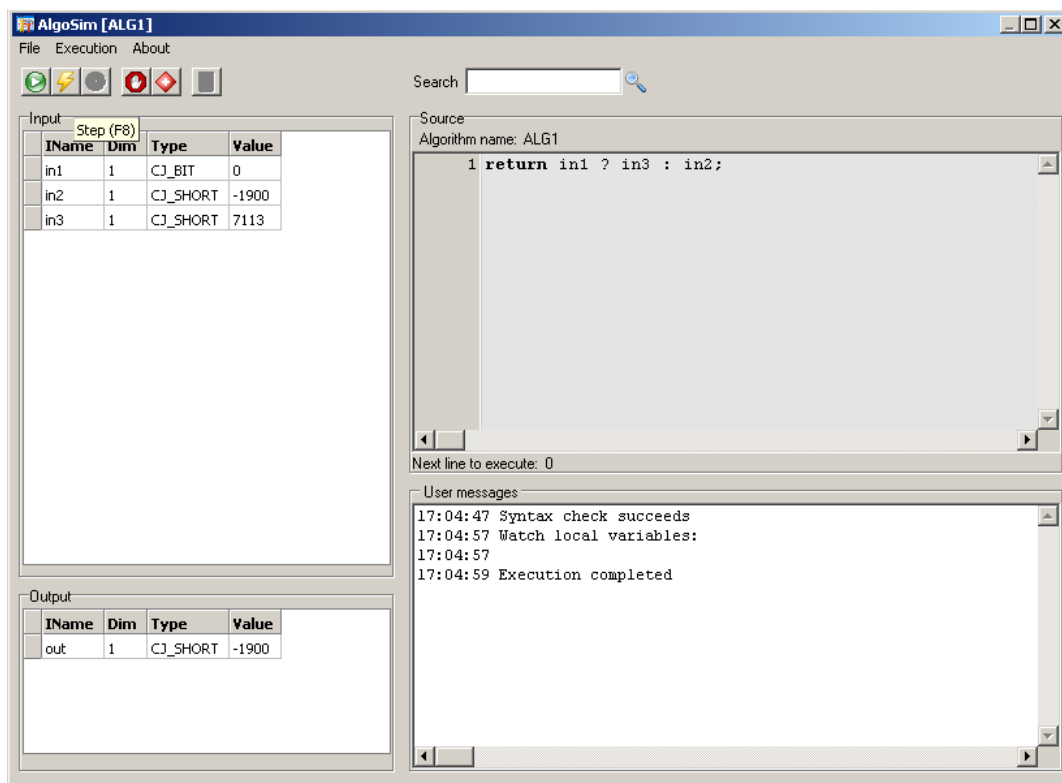
Нажмите на *Step* , симулятор использует исполняемую ветвь, отличную от предыдущей попытки, таким образом можно быстро и подробно опробовать поведение алгоритма в отношении нескольких ситуаций исполнения.

## 2.9.4 Использование во время отладки

Когда проект выполняется на оборудовании, подключенном к вашему ПК, благодаря отладчику вы можете наблюдать за промежуточными состояниями и переходом от выходных данных к изменениям входных данных.




Щелкнув правой кнопкой мыши по любому алгоритму и выбрав AlgoSim, он открывает симулятор с уже предустановленными входными данными со значениями, принятыми в данный конкретный момент.



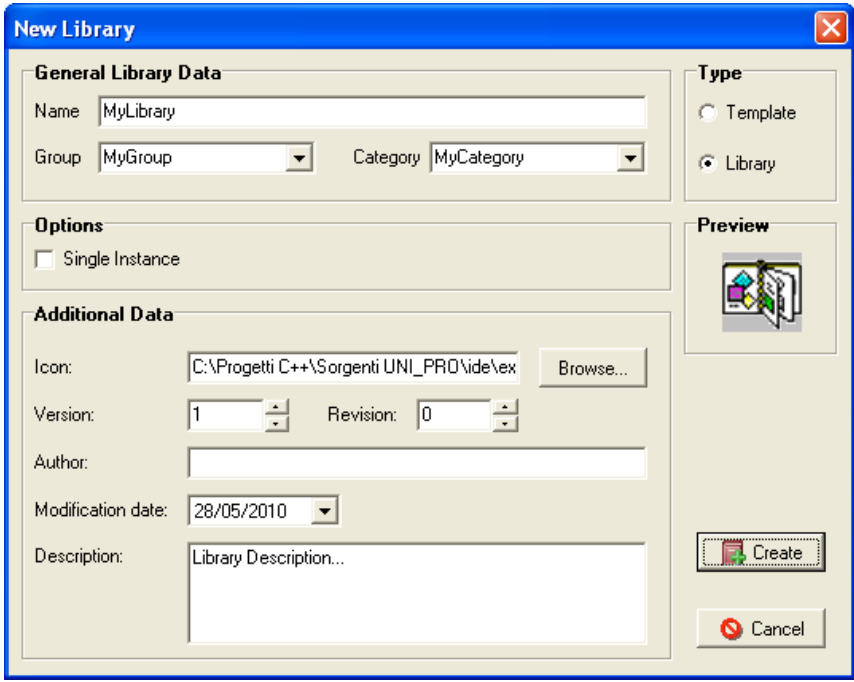
## 2.10 Создание библиотек

Возможность хранения серий объектов в формате библиотеки позволяет создать свою собственную коллекцию объектов, которую Вы можете повторно использовать в других проектах. Это дает возможность значительно увеличить экономию времени и, в то же самое время, убедиться, что ноу-хау останется в пределах компании.

Чтобы создать библиотеку, выделите выбранный элемент – алгоритм или папку – и затем выполните одно из перечисленных действий:

- Нажмите правой кнопкой мыши и выберите *New library* во всплывающем меню;
- Выберите в меню Library/New library;
- Нажмите значок  на панели инструментов.

Появится окно:



Чтобы создать новую библиотеку, необходимо задать определенные характеристики, которые могут быть поделены на четыре секции: *General Data* (Основные данные), *Options* (Опции), *Additional Data* (Дополнительные данные) и *Type* (Тип).

В секции *General Data*, Вы можете задать имя библиотеки, *Group* – группу, к которой она относится, - и дочернюю папку *Category* библиотеки. Без *Group* или *Category* библиотеки создавать нельзя. Свойство *Single Instance* (если выбрано) добавляет к библиотеке особенность быть использованной единожды в проекте, в которой она будет добавлена. Данная функция полезна при создании библиотек, поскольку множественное присутствие в проекте может создать проблемы в работе программы; например, если в библиотеке содержатся алгоритмы со статическими переменными или функциями, вызывающими драйверы встроенного ПО приложения. В секции *Additional Data* Вы можете задать опции, которые позволят настроить каждую конкретную библиотеку; фактически Вы можете соединить ее со значком и ввести номер версии, имя разработчика, дату создания/модификации и краткое описание. В секции *Type* Вы можете выбрать тип библиотеки: *Template* (Шаблон) является группой многократно используемых сущностей, которые могут быть просмотрены, т.е. после того как Вы добавили ее в проект, можно двойным нажатием левой кнопки мыши на ее значок просмотреть или изменить ее содержание; с другой стороны, *Library* является группой сущностей, которые нельзя открыть или изменить, и, соответственно, после создания этого типа библиотеки Вы не сможете просмотреть ее содержание.


После ввода всех данных нажмите *Create* для подтверждения. Если библиотека с таким именем до сих пор не существует, то новая библиотека будет сразу добавлена в древовидное меню библиотек с выбранным значком, иначе система запросит подтверждение перед перезаписыванием.

При подведении курсора к значку новой библиотеки на панели инструментов будет показана определенная информация о ней, такая как имя, версия, дата последней модификации, описание и тип.

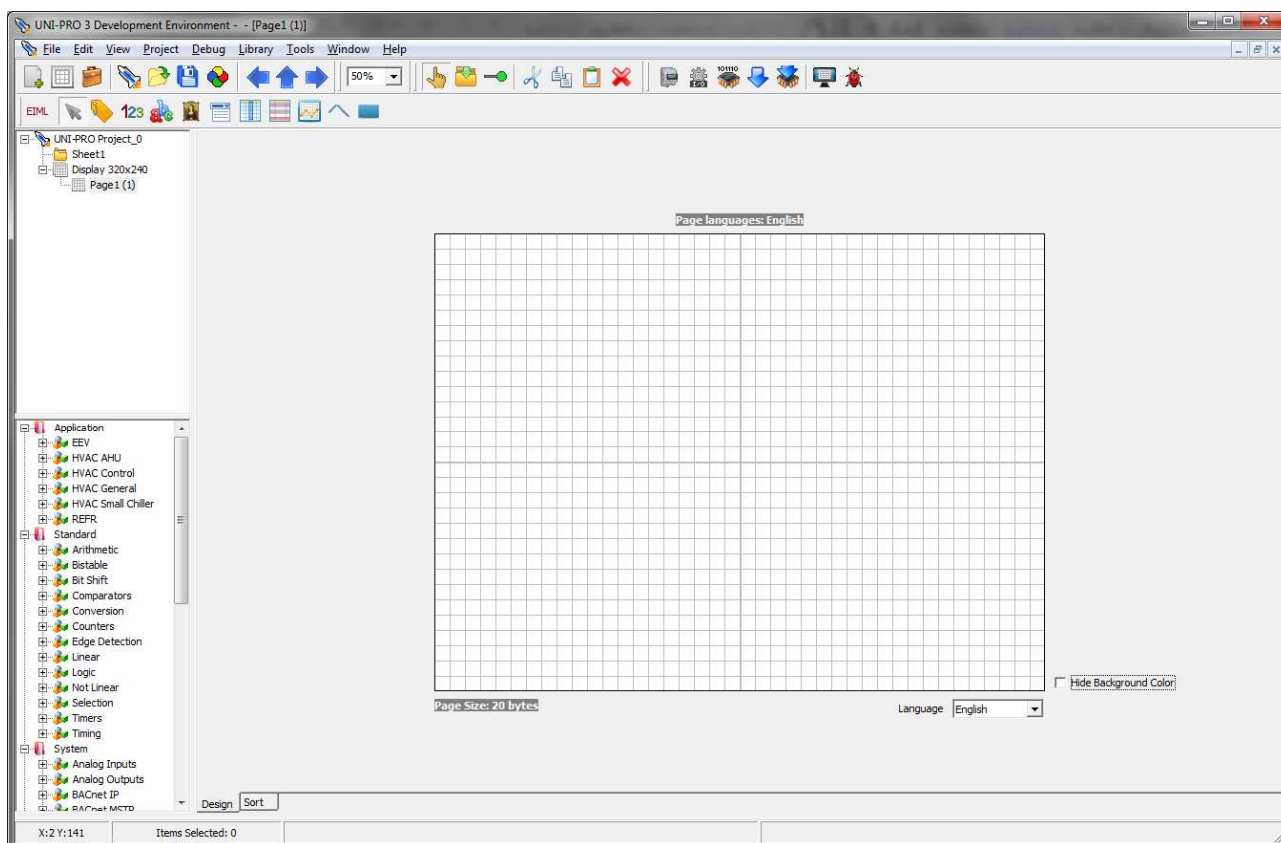
Чтобы добавить библиотеку в новый проект, просто выделите ее и перетащите внутрь листа.

## 2.11 Создание EIML-страниц

EIML-страницы позволяют описывать графический интерфейс контроллера в графическом формате. Проект может содержать несколько взаимосвязанных страниц для выведения на экран текстов, значков и переменных, представляющих внутренние состояния контроллера.

Чтобы добавить страницу в проект, выберите в меню *File/New/New EIML Page* или нажмите значок  на панели инструментов. Как только Вы выберете тип страницы, она станет готова к созданию, на дереве проекта появится узел, представляющий только что добавленную страницу, а визуальный образ страницы появится в центре среды разработки.

EIML-страницы связаны с определенным типом дисплея; например, рисунок снизу показывает чистую страницу LCD графического дисплея с разрешением 320x240 пикселей.




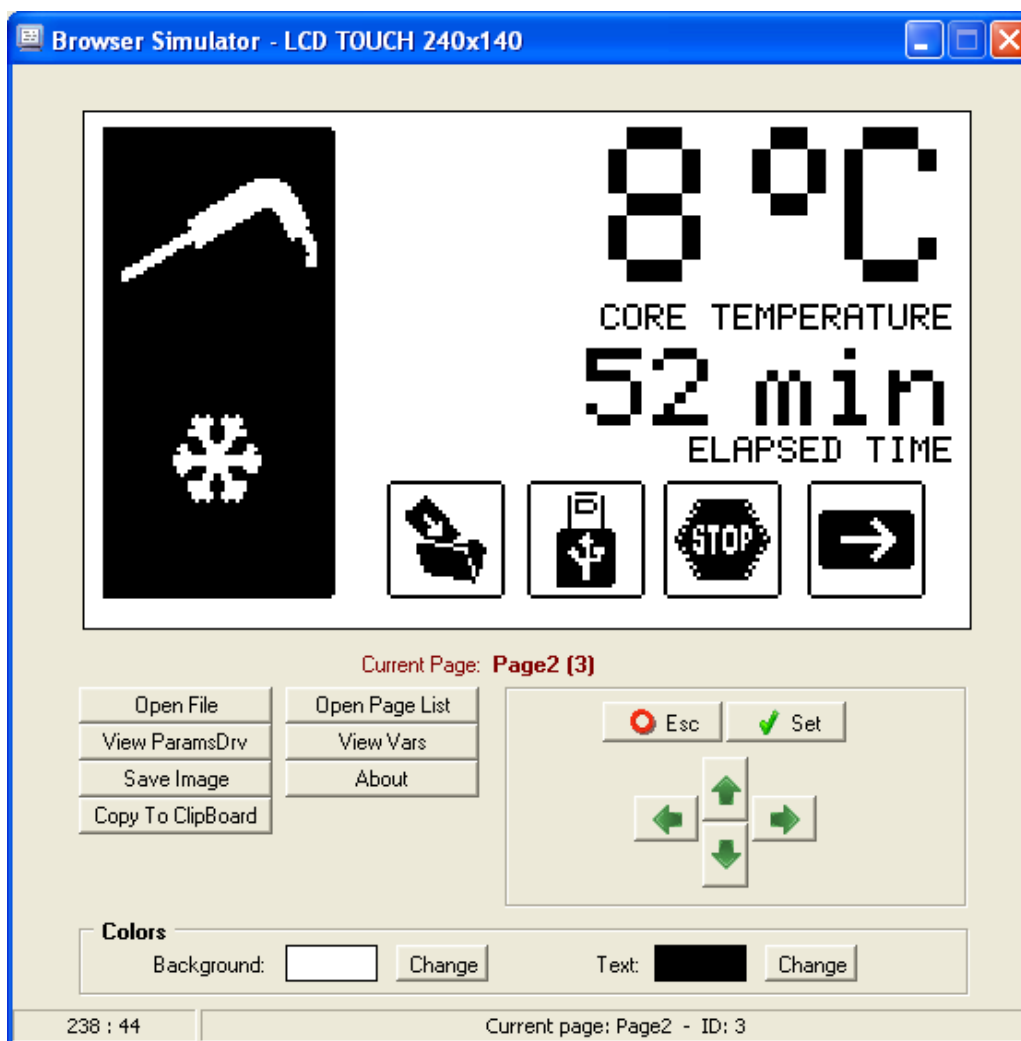
EIML-страницы могут содержать тексты, переменные, значки, поля со списком, таблицы, линии, прямоугольники, все, что будет описано в параграфе 3.4.

Созданные страницы следуют методу WYSIWYG («что видишь, то и получишь»). Это означает, что занятые позиции, размеры шрифта и статические свойства, такие как расположение или текст, выводятся на дисплей в том же самом виде, в котором они и были установлены. Чтобы отобразить свойства динамического типа, такие как мигание или движение курсора, Вам необходимо перейти к моделированию EIML-страниц.



## 2.12 Симуляция EIML-страниц

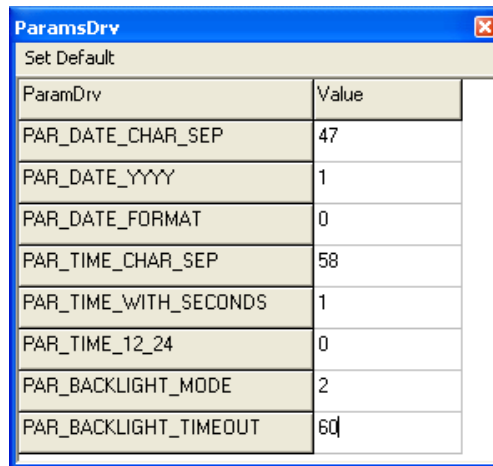
При разработке графического интерфейса было бы целесообразно смоделировать реальное поведение разрабатываемых страниц, чтобы проверить, были ли они корректно внедрены. UNI-PRO имеет встроенную программу, которая позволяет вывести на экран выход страницы графического интерфейса таким образом, как если бы он был выполнен в виде дисплея. Чтобы активировать данную программу, перейдите на страницу, которую вы хотите смоделировать, и выберите пункт меню *Project/EIML Simulator* или нажмите значок  на панели инструментов. Ниже представлен пример моделирования 240x128 пиксельной графической страницы:



Моделирующая программа позволяет проверить правильность вывода на дисплей всех элементов внутри страницы (текстов, переменных, значков, таблиц и т.д.), правильность выполнения всех действий (перемещение через поля, обработка и отправка команд), а также могут ли страницы просматриваться корректно.

В левой части окна моделирующей программы имеется кнопка *Open file*: нажатие на нее загружает EIML-страницу, сохраненную в двоичном формате. Вместо этого, при нажатии кнопки *Open Page List*, Вы можете загрузить группу страниц, сохраненных в качестве файлов, чтобы проверить, просматриваются ли они корректно. Когда программа моделирования запускается из UNI-PRO, загружается список страниц, входящих в текущий проект, и предлагается загрузить в качестве первой страницы ту, что была обработана.

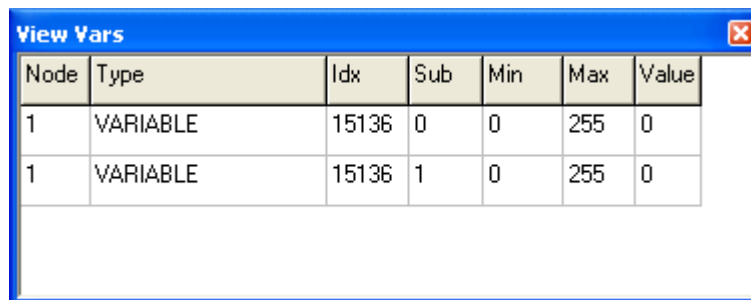
Нажимая *View ParamsDrv*, Вы можете отобразить и внести изменения в определенные параметры моделирования, также включая и встроенное программное обеспечение, что повлияет на способ вывода страниц на дисплей.



ParamDrv	Value
PAR_DATE_CHAR_SEP	47
PAR_DATE_YYYY	1
PAR_DATE_FORMAT	0
PAR_TIME_CHAR_SEP	58
PAR_TIME_WITH_SECONDS	1
PAR_TIME_12_24	0
PAR_BACKLIGHT_MODE	2
PAR_BACKLIGHT_TIMEOUT	60

Например, Вы можете выбрать, какие символы будут использоваться в качестве разделительных знаков времени и даты, как должны выводиться на дисплей значения года, в качестве 2-х или 4-х значных чисел, а также установить формат данных.

При нажатии кнопки *View Var* будет показан список переменных, используемых на странице, и, если необходимо, Вы можете изменять их значения во время моделирования.

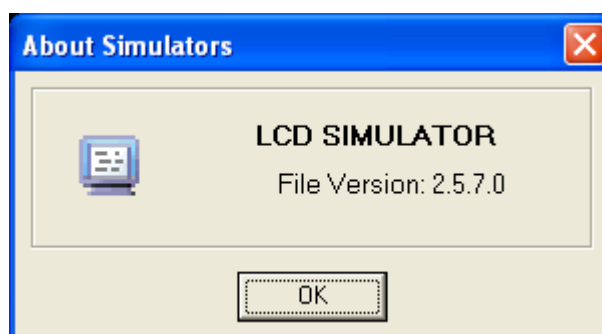


Node	Type	Idx	Sub	Min	Max	Value
1	VARIABLE	15136	0	0	255	0
1	VARIABLE	15136	1	0	255	0

Для каждой переменной, используемой страницей, добавляется строка, где сведены основные свойства, и в которой Вы можете изменять их значения.

Кнопки *Save Image* и *Copy to clipboard* используются для сохранения текущего отображения в качестве файла точечной графики или копирования его в буфер обмена системы, чтобы Вы могли вставить его (*Paste*) в другую программу.

При нажатии кнопки *About* появится окно с версией программы моделирования.



Оставшиеся кнопки, ESC, ENTER, UP, DOWN, LEFT и RIGHT, моделируют работу соответствующих кнопок, находящихся в пользовательском интерфейсе, что позволяет выполнять такие действия, как передвижение курсора (UP и DOWN), обработка значения (ENTER, UP, DOWN, ENTER или ESC) или просмотр страниц (ENTER для элемента, с которым связана загрузка страницы, ESC для возврата к предыдущей странице).

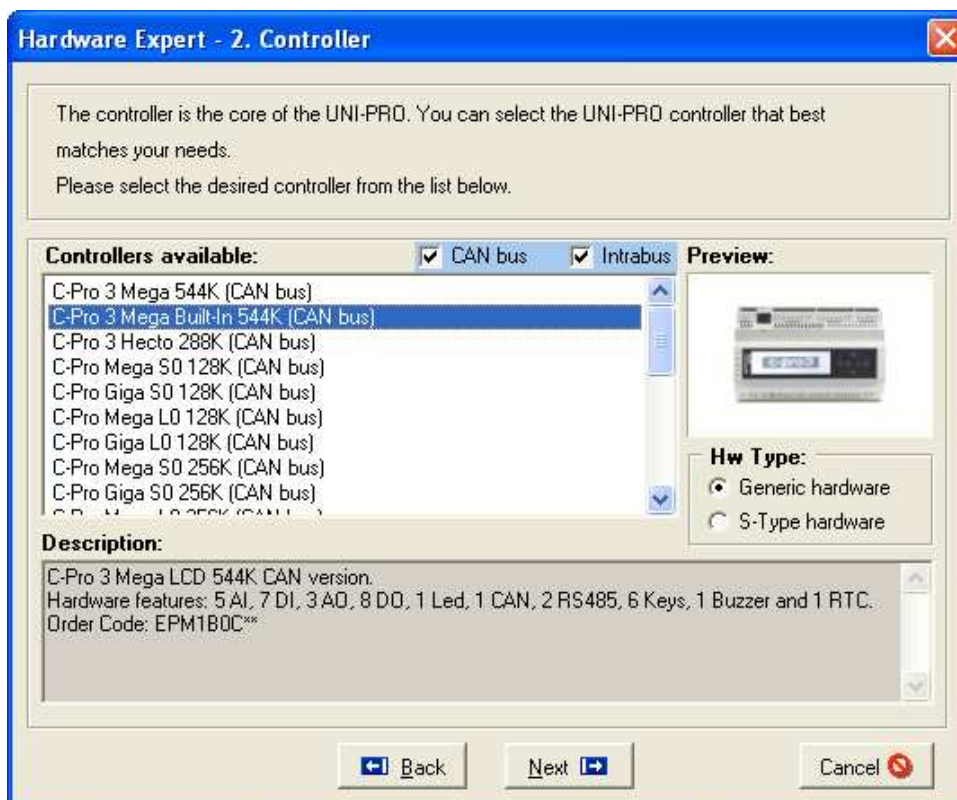
## 2.13 Выбор оборудования

Разработка проекта с использованием UNI-PRO может быть применима для группы контроллеров, совместимых с этой средой программирования. Поэтому Вы должны определить, для каких контроллеров будет применяться проект. Для облегчения работы разработчиков контроллер может быть выбран как на начальной фазе разработки, для создания проекта в зависимости от контроллера, так и на стадии компиляции, в зависимости от используемых ресурсов.

Чтобы выбрать оборудование, запустите программу *Hardware Expert*, выберите в меню *Project/Hardware Expert* или нажмите на значок . Появится окно презентации программы.

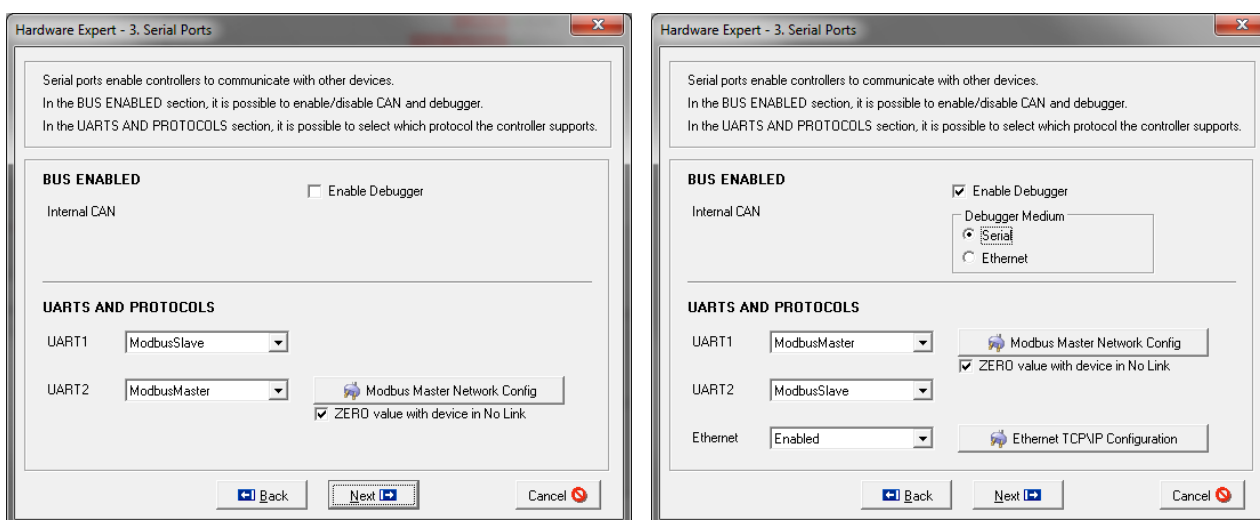


Нажмите *Next*, чтобы перейти к следующему окну, где будут перечислены все типы контроллеров.



При выделении контроллера в центре окна появляется кратное описание его ресурсов, а с правой стороны – изображение контроллера. Контроллеры могут различаться в зависимости от локальной шины, которая может представлять собой IntraBus либо CAN (Controlled Area Network). В разделе *Hw type* требуется выбрать используемый тип аппаратного обеспечения, универсальный или особый (S-Type).

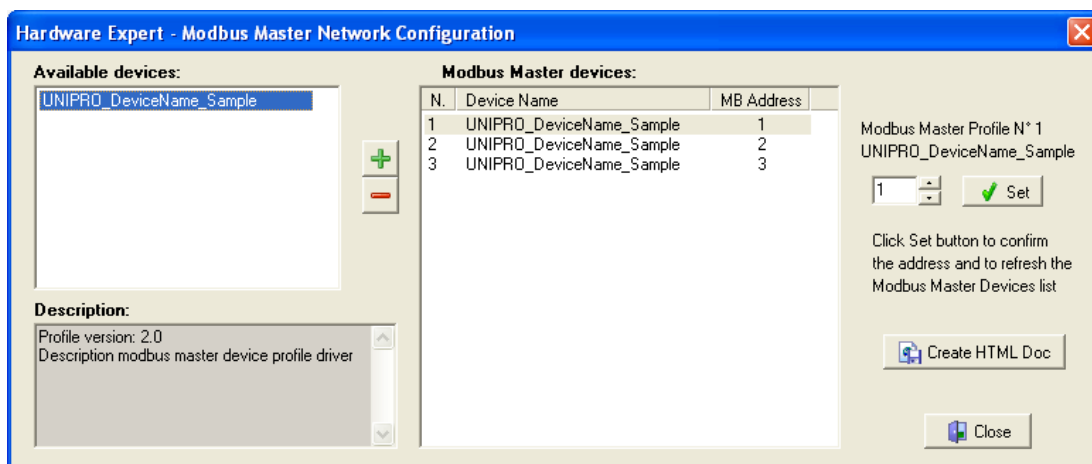
Выберите наиболее подходящий контроллер и нажмите *Next*.



Третье окно составлено из двух секций: первая позволяет подключать каналы CAN, имеющиеся в контроллере, чтобы настроить их для сети; вторая позволяет выбрать протокол (например, None, ModbusSlave, VACnet, Ethernet и т.д.) для каждого имеющегося UART (универсальный асинхронный интерфейс).

### 2.13.1 Сеть ModbusMaster

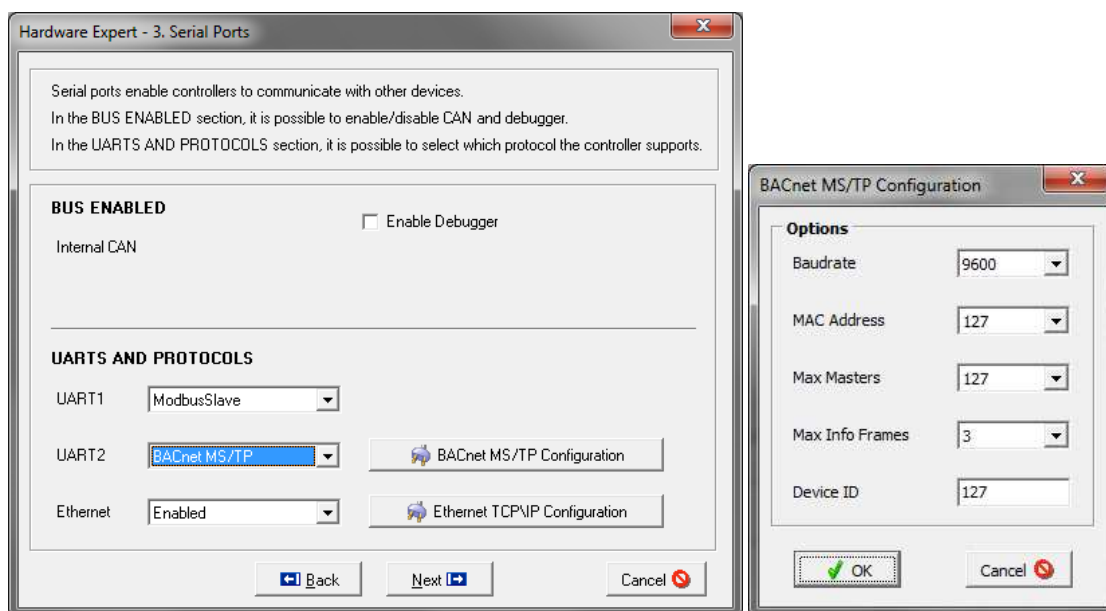
Если в данном разделе выбирается протокол ModbusMaster, то справа появляется кнопка конфигурации. Нажмите её, чтобы открыть программу конфигурации сети ModbusMaster.



Слева находится список всех доступных профилей ModbusMaster; чтобы добавить другие профили, требуется создать драйвер конфигурации. С помощью кнопок **+** и **-** можно добавлять или удалять профили для реализации в сети. Когда сеть сформирована, следует настроить все адреса modbus. Чтобы продолжить, нажмите Close.

### 2.13.2 Конфигурация BACnet MS/TP

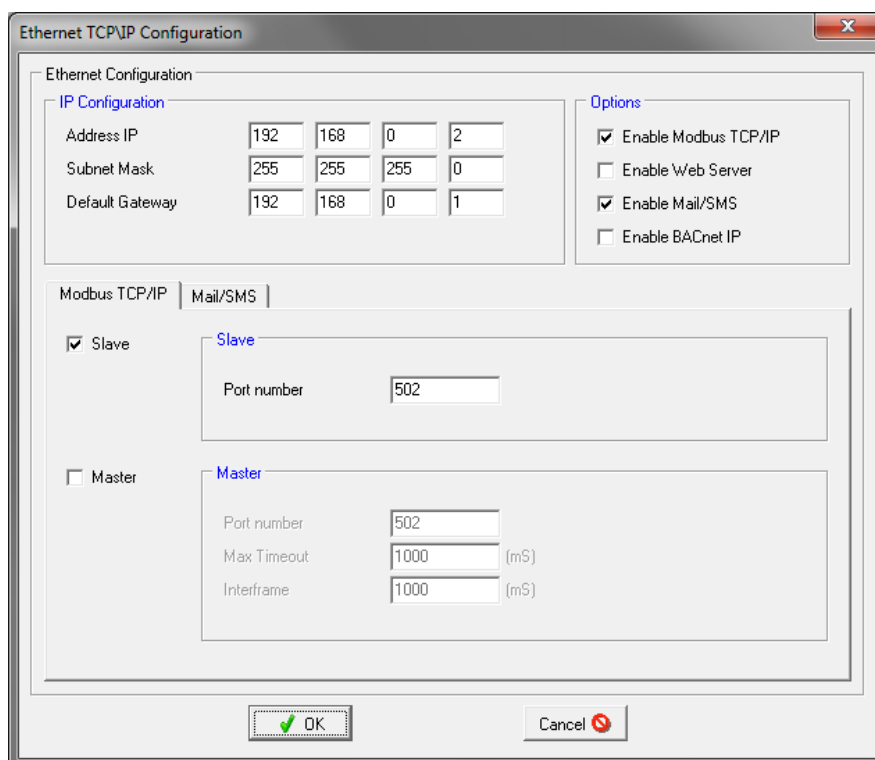
Выбрав BACnet MS/TP на UART, вы можете настроить свойства связи.



Протокол BACnet MS/TP является альтернативой протоколу BACnet IP и наоборот.  
**ПРИМЕЧАНИЕ.** Проверьте в Руководстве по оборудованию, совместим ли контроллер с протоколом BACnet.

### 2.13.3 Конфигурация протоколов поверх Ethernet

Если вы решите использовать порт Ethernet, можно установить основные параметры для связи выбранных протоколов по IP.



Протокол BACnet IP является альтернативой протоколу BACnet MS/TP и наоборот. IP-протокол BACnet является альтернативой протоколу веб-сервера и наоборот.

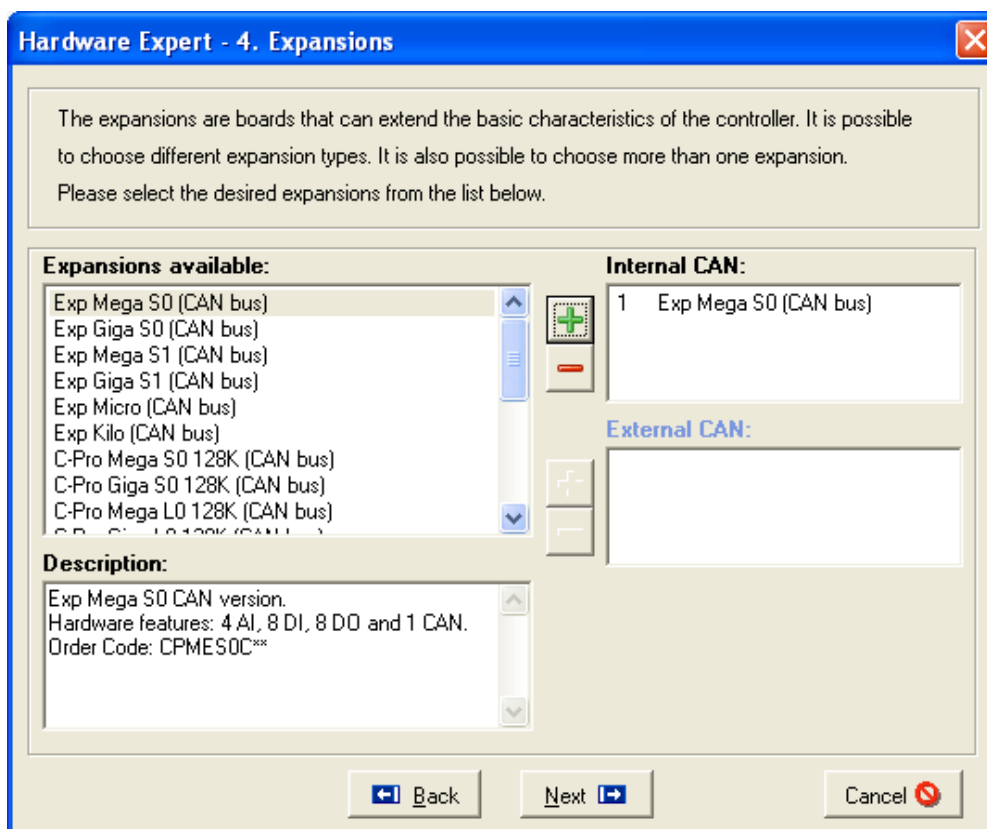
**ПРИМЕЧАНИЕ.** Проверьте в Руководстве по оборудованию, совместим ли контроллер с протоколом BACnet.

### Отправка почты и SMS

Если вы хотите настроить контроллер для отправки электронной почты и SMS, вы должны сначала включить порт Ethernet в эксперте по оборудованию, а затем выбрать Mail/SMS в параметрах, доступных для этого порта.


Затем управление передается пользовательскому приложению с использованием соответствующих стандартных библиотек (см. *Руководство по стандартной библиотеке*).

После этого Эксперт по оборудованию позволяет получить доступ к разделу «Расширения».

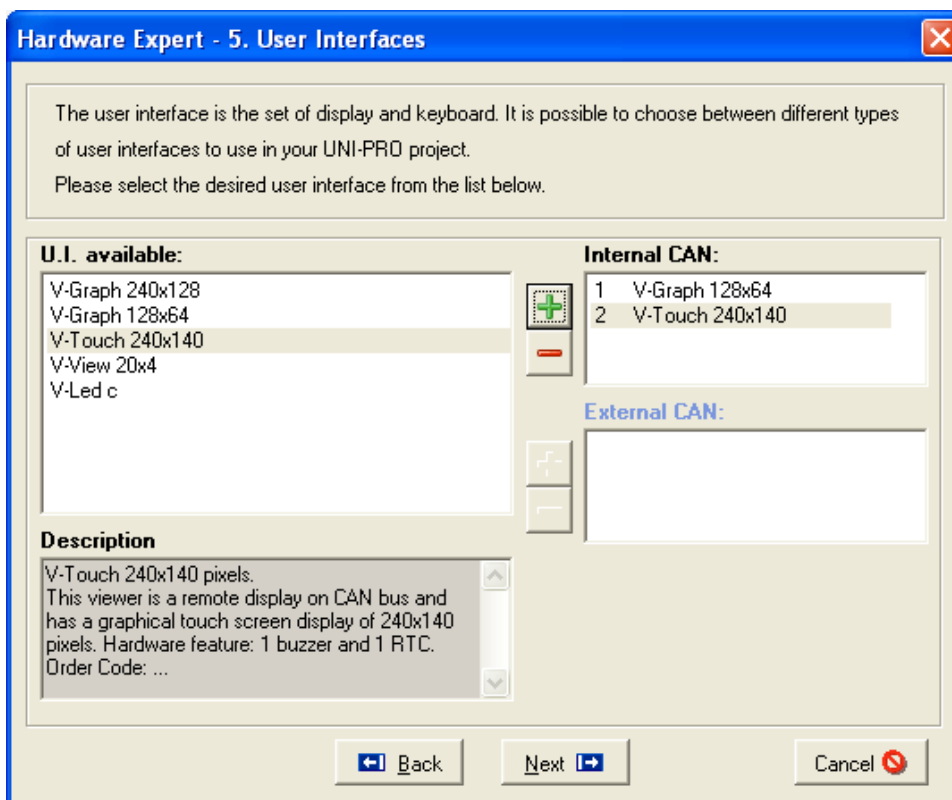


С левой стороны находится список всех доступных расширений, которые, в зависимости от контроллера, могут быть подключены к внутренней или внешней шине CAN.

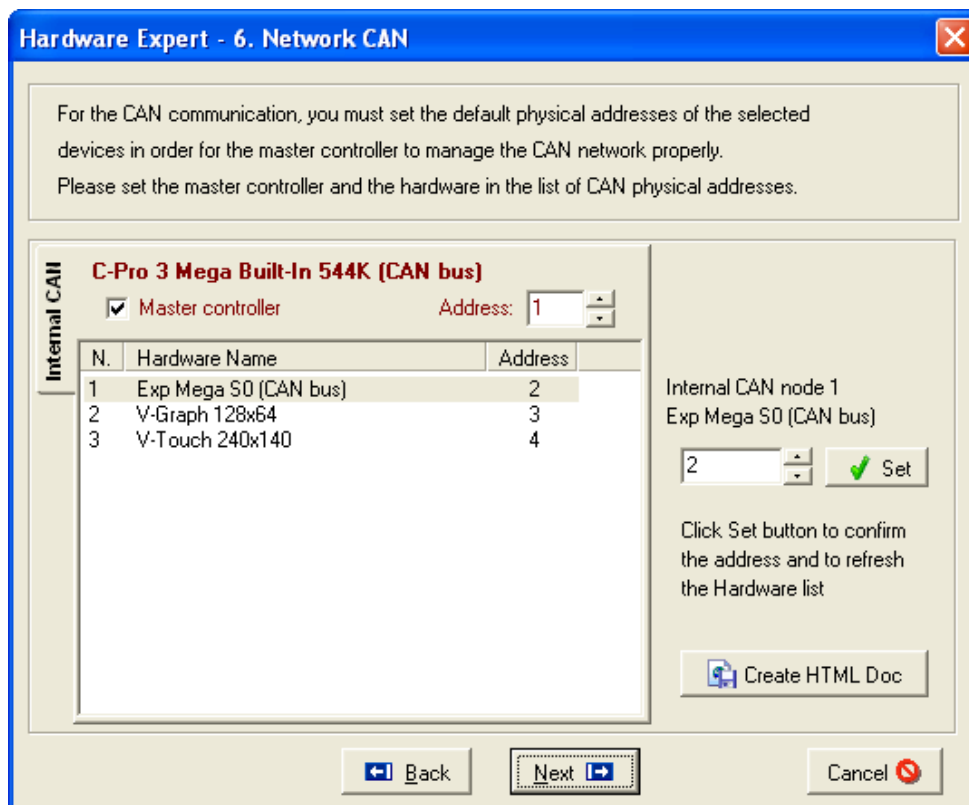
Рамка и кнопки для шины CAN будут доступны, если контроллер имеет эту функцию и если они были активированы на предыдущем шаге настройки.

Если вы хотите добавить расширение, просто выберите его и нажмите кнопку  коммуникационной шины CAN, с помощью которой вы хотите ее связать: появится новая строка.

После того, как вы выбрали все дополнения для добавления, нажмите «Далее», чтобы продолжить.



Теперь можно установить параметры, которые определяют внутреннюю или внешнюю сеть CAN, такие как физический адрес, условие, что контроллер управляет сетью CAN как мастер, а так же все физические адреса других элементов сети:

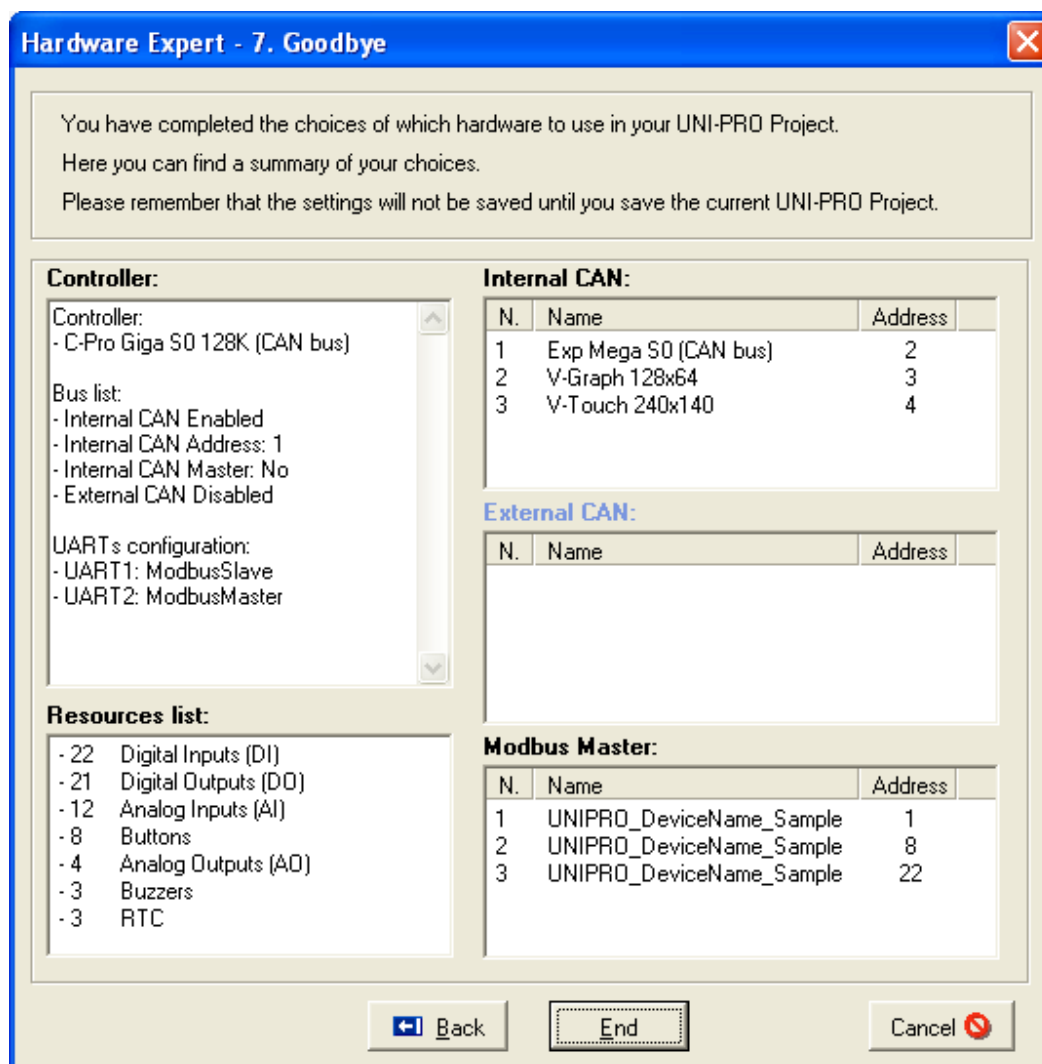




**Примечание.** Если контроллер должен управлять расширениями, важно отметить поле мастера контроллера.

Четвертый шаг предоставляет список всех возможных пользовательских интерфейсов с кратким описанием каждого из них справа. Чтобы добавить пользовательский интерфейс к выбранной шине, действуйте так же, как если бы вы добавили расширение.

На последнем шаге мастер суммирует выбранную конфигурацию, включая тип контроллера, конфигурации последовательного порта, расширения и интерфейсы, связанные с каждой соответствующей коммуникационной шиной.



Чтобы завершить работу мастера, нажмите *Завершить*. Чтобы отменить все внесенные изменения, нажмите кнопку *Отмена*.

## 2.14 Соединение с физическими терминалами

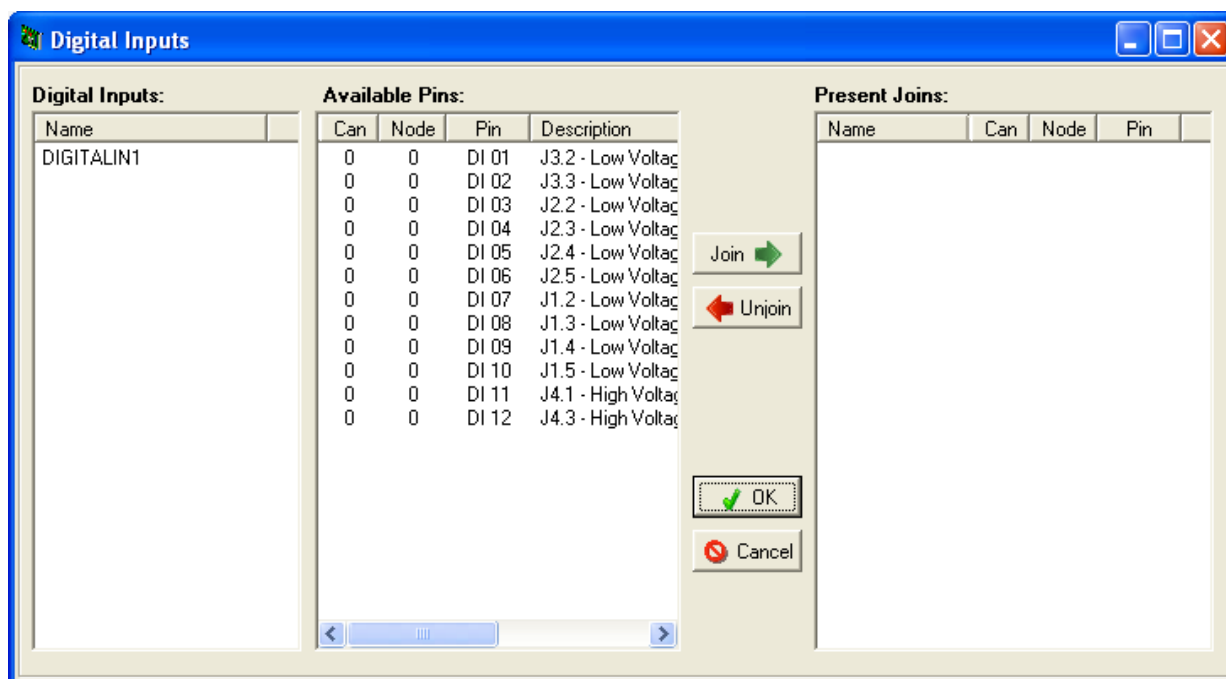
Так как сущности, используемые в UNI-PRO, являются логическим представлением, не зависимым от типа контроллера, то перед компиляцией необходимо установить связь между логическими ресурсами, используемыми в проекте, и физическими ресурсами, имеющимися в выбранном контроллере.

Поэтому все типы сущностей, представляющие Ввод/Вывод контроллера (цифровые входы, цифровые выходы, аналоговые входы, аналоговые выходы, часы, индикаторы, кнопки, вход команды, выход команды и звуковой сигнал), должны быть соединены с локальным или выносным физическим Вводом/Выводом с помощью *Join Tools (Инструменты соединения)*.

Прежде чем рассмотреть каждый конкретный Join Tool, скажем пару слов о функции *Check Join*: она автоматически проверяет все соединения и, если нужно, открывает необходимые окна, чтобы создать соединения для тех сущностей, для которых они еще не были установлены.

### 2.14.1 Цифровые входы

Мастер соединения цифровых входов автоматически запускается во время работы функции Check Join, если какие-либо цифровые входы остались неподсоединенными; также ее можно запустить вручную из меню, для этого выберите *Project/Join Tools/Digital Inputs*.



Окно разделено на три части: *Digital Inputs*, *Available Pins* и *Present Joins*.

Список первой части содержит сущности проекта, которые еще не были связаны с ресурсами оборудования.

Список второй части содержит имеющиеся физические цифровые входы со следующими характеристиками:

- Node : номер узла, где располагается ресурс
- Can : шина, к которой присоединен контроллер
- Pin : логический указатель терминала
- Description : краткое описание из документации по руководству к использованию выбранного контроллера
- HW name : имя контроллера, в котором располагается ресурс

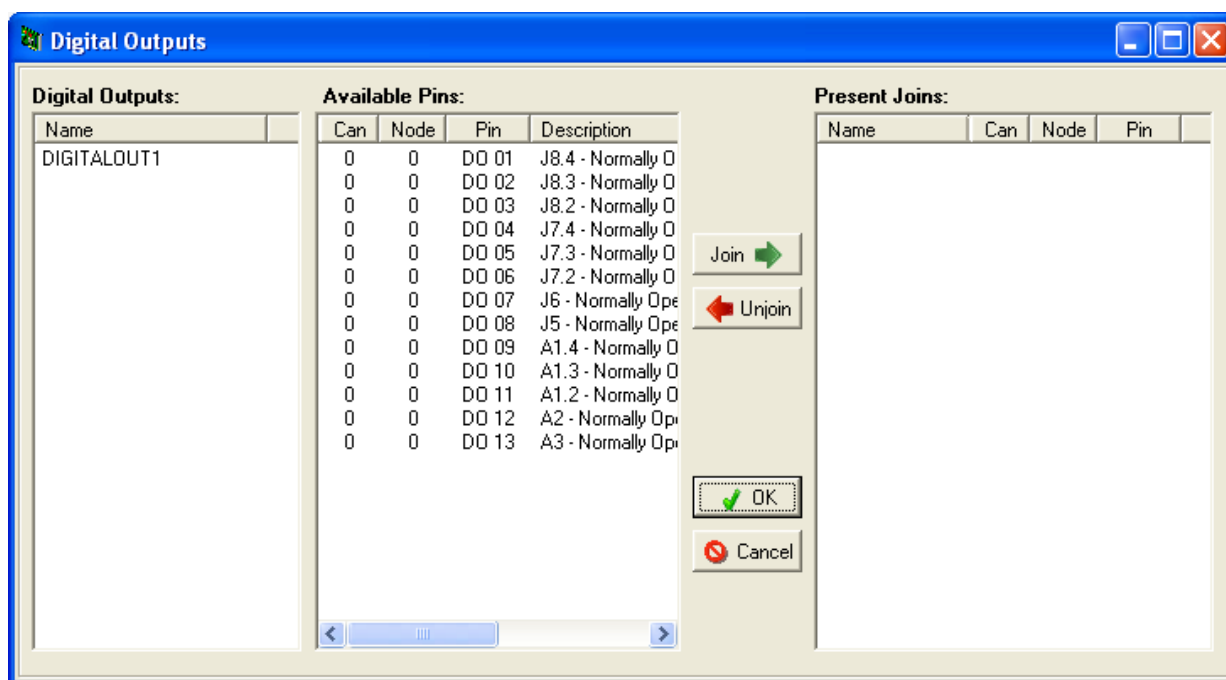
Список третьей части содержит уже созданные соединения между цифровыми входами и физическими ресурсами.

Чтобы связать сущность с ресурсом, выделите вход в таблице Digital Inputs и физический ресурс из списка Available Pins, затем нажмите кнопку *Join*. Сущность и терминал оборудования исчезнут из соответствующих списков, а в списке Present Joins появится новая строка.

С другой стороны, чтобы удалить соединение, выделите строку в списке Present Joins и нажмите кнопку *Unjoin*: строка будет удалена, логический терминал будет восстановлен в списке первой части, а физический ресурс – во второй. Для подтверждения связи нажмите *OK*, для отмены проделанных операций нажмите *Cancel*.

### 2.14.2 Цифровые выходы

Программа соединения цифровых выходов автоматически запускается во время работы функции Check Join, если какие-либо цифровые выходы остались неподсоединенными; также ее можно запустить вручную из меню, для этого выберите *Project/Join Tools/Digital Outputs*.



Окно разделено на три части: *Digital Outputs*, *Available Pins* и *Present Joins*.

Список первой части содержит сущности проекта, которые еще не были связаны с ресурсами оборудования.

Список второй части содержит имеющиеся физические цифровые выходы, со следующими характеристиками:

- Node : номер узла, где располагается ресурс
- Can : шина, к которой присоединен контроллер
- Pin : логический указатель терминала
- Description : краткое описание из документации по руководству к использованию выбранного контроллера
- Hardware name : имя контроллера, в котором располагается ресурс

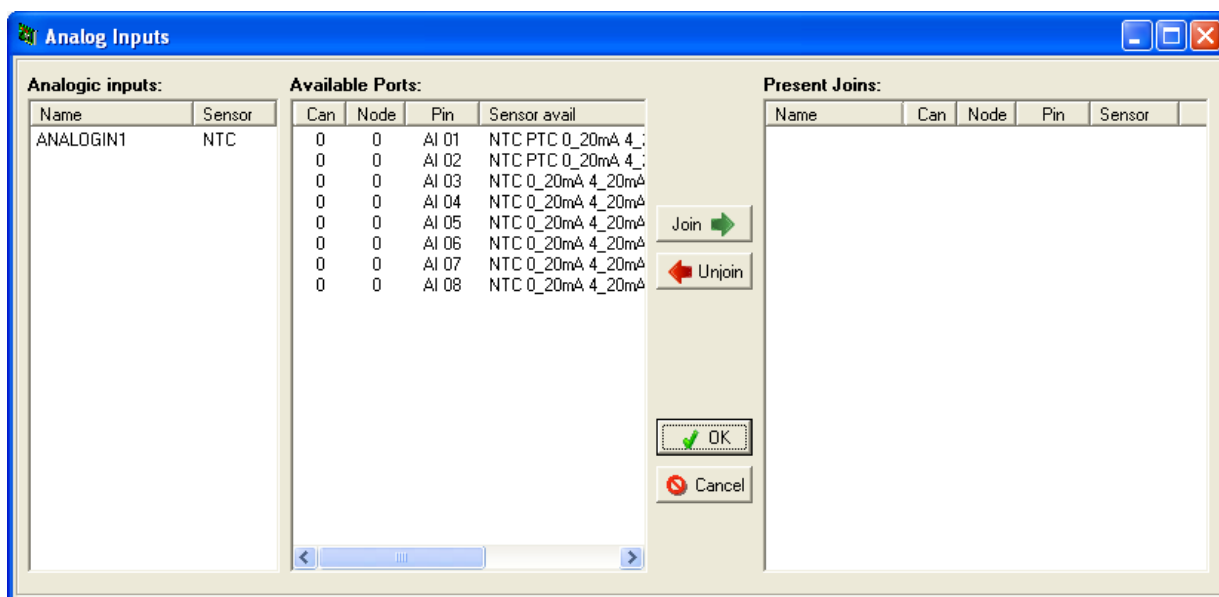
Список третьей части содержит уже созданные соединения между цифровыми выходами и физическими ресурсами.

Чтобы связать сущность с ресурсом, выделите выход в списке Digital Outputs и физический ресурс из списка Available Pins, затем нажмите кнопку *Join*. Сущность и терминал оборудования исчезнут из соответствующих списков, а в списке «Present Joins» появится новая строка.

С другой стороны, чтобы удалить соединение, выделите строку в списке «Present Joins» и нажмите кнопку *Unjoin*: строка будет удалена, логический терминал будет восстановлен в списке первой части, а физический ресурс – во второй. Для подтверждения связи нажмите *OK*, для отмены проделанных операций нажмите *Cancel*.

### 2.14.3 Аналоговые входы

Программа соединения аналоговых входов автоматически запускается во время работы функции Check Join, если какие-либо цифровые входы остались неподсоединенными; также ее можно запустить вручную из меню, для этого выберите *Project/Join Tools/Analog Inputs*.



Окно разделено на три части: *Analog Inputs*, *Available Pins* и *Present Joins*.

Список первой части содержит сущности проекта, которые еще не были связаны с ресурсами оборудования.

Список второй части содержит имеющиеся физические аналоговые входы, со следующими характеристиками:

- Node : номер узла, где располагается ресурс
- Can : шина, к которой присоединен контроллер
- Pin : логический указатель терминала
- Sensor avail : список типов датчиков, которые могут быть соединены с физическим ресурсом
- Hw Name : имя контроллера, в котором располагается ресурс

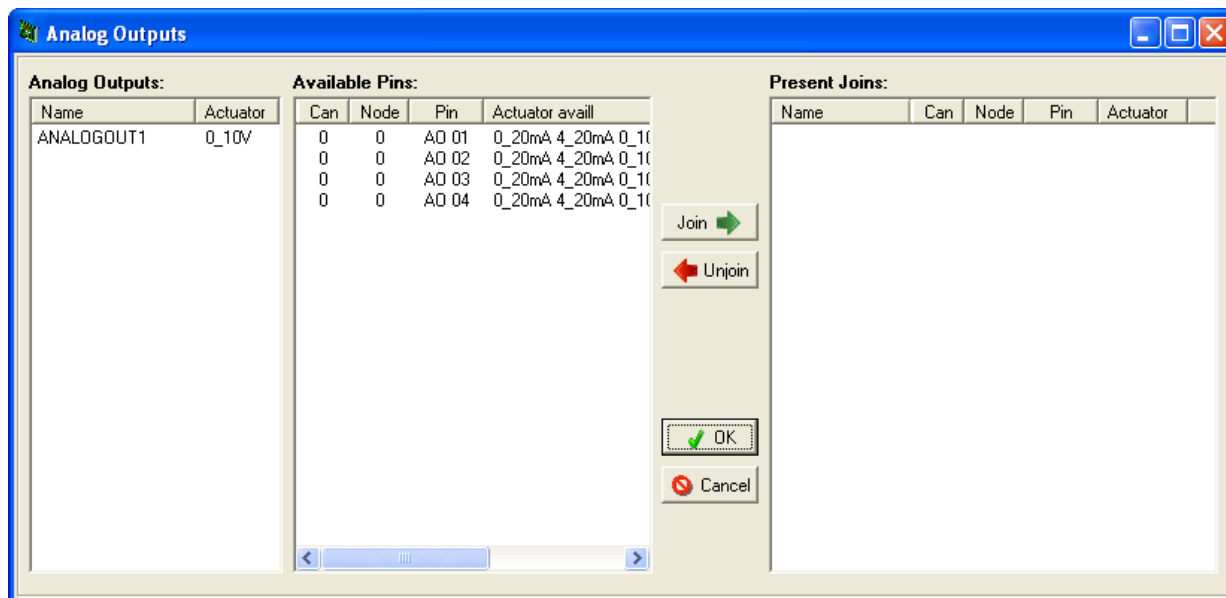
Список третьей части содержит уже созданные соединения между аналоговыми входами и физическими ресурсами.

Чтобы связать сущность с ресурсом, выделите вход в списке Analog Inputs и физический ресурс с совместимым типом датчика в списке Available Pins, затем нажмите кнопку *Join*. Сущность и терминал оборудования исчезнут из соответствующих списков, а в списке *Present Joins* появится новая строка.

С другой стороны, чтобы удалить соединение, выделите строку в списке *Present Joins* и нажмите кнопку *Unjoin*: строка будет удалена, логический терминал будет восстановлен в списке первой части, а физический ресурс – во второй. Для подтверждения связи нажмите *OK*, для отмены проделанных операций нажмите *Cancel*.

#### 2.14.4 Аналоговые выходы

Программа соединения аналоговых выходов автоматически запускается во время работы функции Check Join, если какие-либо цифровые выходы остались неподсоединенными; также ее можно запустить вручную из меню, для этого выберите *Project/Join Tools/Analog Outputs*.



Окно разделено на три части: *Analog Outputs*, *Available Pins* и *Present Joins*.

Список первой части содержит сущности проекта, которые еще не были связаны с ресурсами оборудования.

Список второй части содержит доступные физические аналоговые выходы со следующими характеристиками:

- Node : номер узла, где располагается ресурс
- Can : шина, к которой присоединен контроллер
- Pin : логический указатель терминала
- Actuator avail: список типов аналоговых выходов
- Hw name : имя контроллера, в котором располагается ресурс

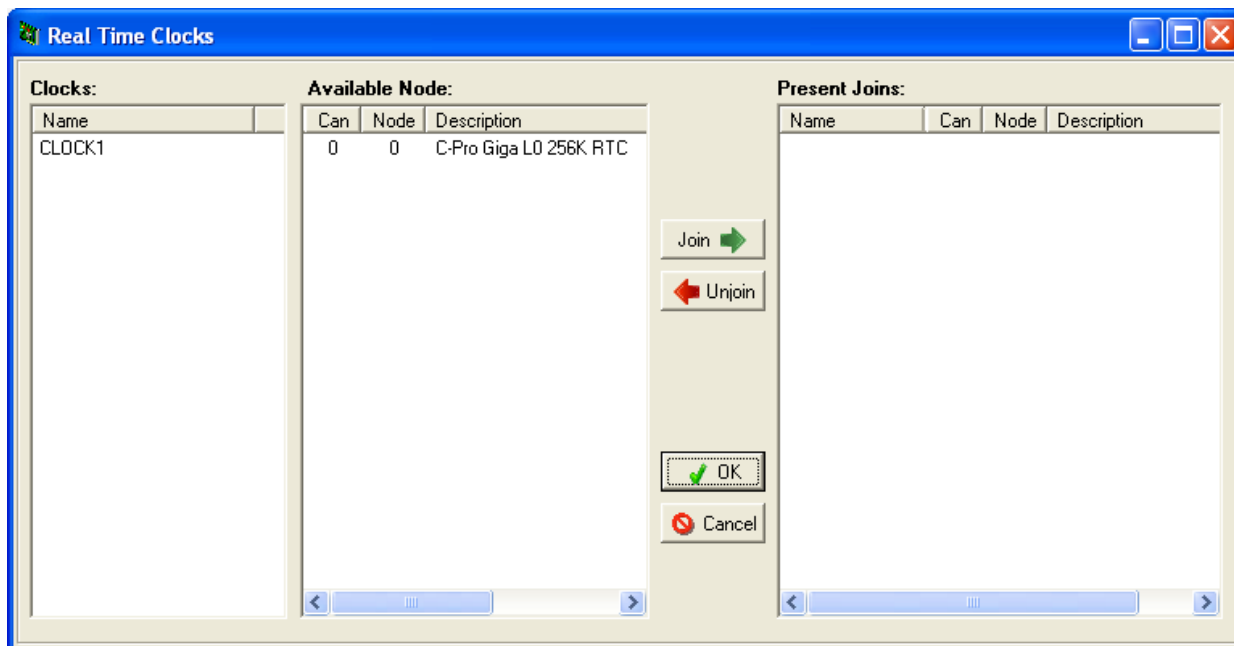
Список третьей части содержит уже созданные соединения между аналоговыми выходами и физическими ресурсами.

Чтобы связать сущность с ресурсом, выделите выход в списке *Analog Outputs* и физический ресурс из списка *Available Pins*, затем нажмите кнопку *Join*. Сущность и терминал оборудования исчезнут из соответствующих списков, а в списке *Present Joins* появится новая строка.

С другой стороны, чтобы удалить соединение, выделите строку в списке *Present Joins* и нажмите кнопку *Unjoin*: строка будет удалена, логический терминал будет восстановлен в списке первой части, а физический ресурс – во второй. Для подтверждения связи нажмите *OK*, для отмены проделанных операций нажмите *Cancel*.

### 2.14.5 Часы

Программа соединения часов автоматически запускается во время работы функции Check Join, если какие-либо часы остались неподсоединенными; также ее можно запустить вручную из меню, для этого выберите *Project/Join Tools/Clocks*.



Окно разделено на три части: *Clocks*, *Available Node* и *Present Joins*.

Список первой части содержит сущности проекта, которые еще не были связаны с ресурсами оборудования.

Список второй части содержит имеющиеся физические часы со следующими характеристиками:

- Node : номер узла, где располагается ресурс
- Can : шина, к которой присоединен контроллер
- Description : название типа контроллера, где находятся часы

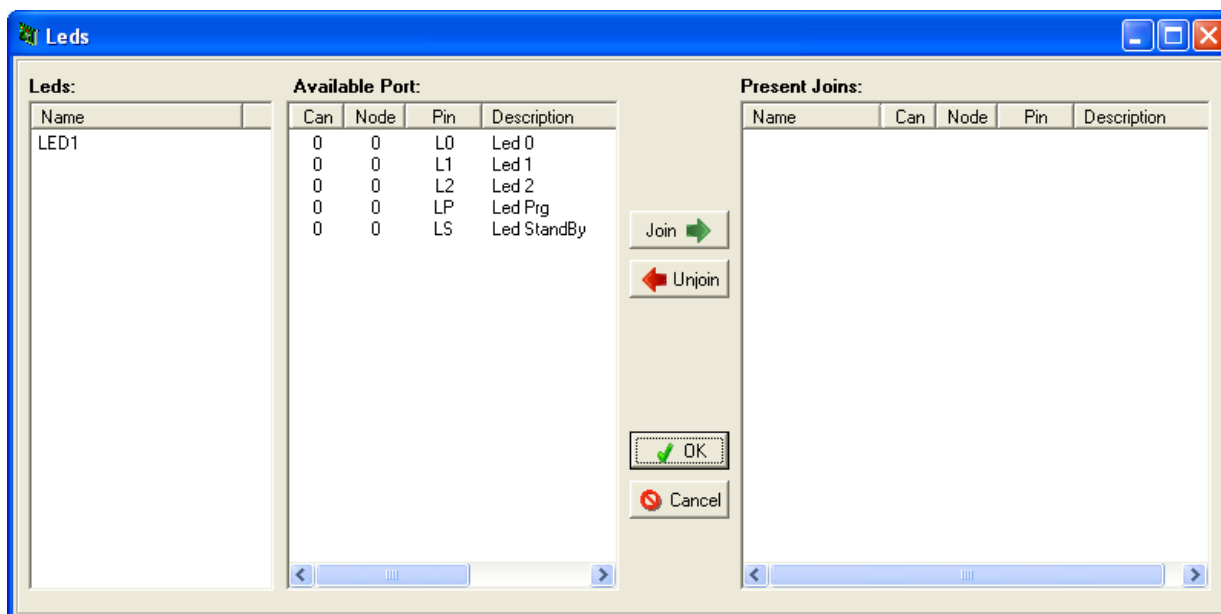
Список третьей части содержит уже созданные соединения между часами и физическими ресурсами.

Чтобы связать сущность с ресурсом, выделите RTC в списке *Clocks* и физический ресурс из списка *Available Node*, затем нажмите кнопку *Join*. Сущность и терминал оборудования исчезнут из соответствующих списков, а в списке *Present Joins* появится новая строка.

С другой стороны, чтобы удалить соединение, выделите строку в списке *Present Joins* и нажмите кнопку *Unjoin*: строка будет удалена, логический терминал будет восстановлен в списке первой части, а физический ресурс – во второй. Для подтверждения связи нажмите *OK*, для отмены проделанных операций нажмите *Cancel*.

### 2.14.6 Индикаторы

Программа соединения индикаторов автоматически запускается во время работы функции Check Join, если хотя бы один индикатор остался неподключенным; также ее можно запустить вручную из меню, для этого выберите *Project/Join Tools/LEDs*.



Окно разделено на три части: *LEDs*, *Available Port* и *Present Joins*.

Список первой части содержит сущности проекта, которые еще не были связаны с ресурсами оборудования.

Список второй части содержит имеющиеся физические индикаторы со следующими характеристиками:

- Node : номер узла, где располагается ресурс
- Can : шина, к которой присоединен контроллер
- Pin : логический указатель терминала
- Description : краткое описание из документации по руководству к использованию контроллера
- Hw name : имя контроллера, в котором располагается ресурс

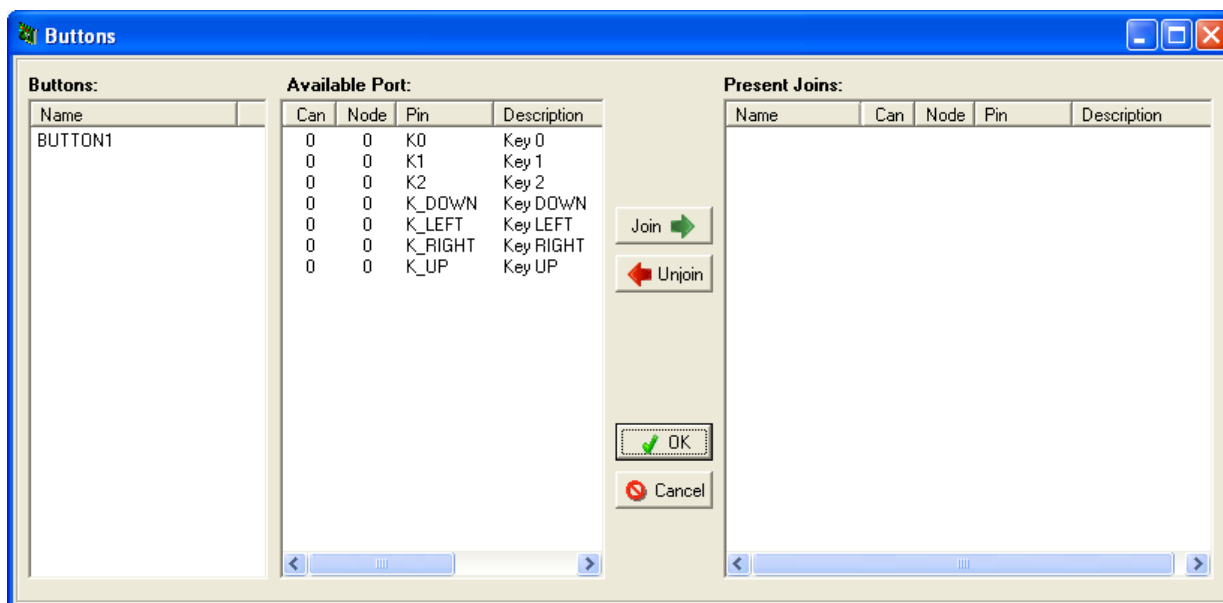
Список третьей части содержит уже созданные соединения между индикаторами и физическими ресурсами.

Чтобы связать сущность с ресурсом, выделите индикатор в списке первой части и физический ресурс из списка *Available Port*, затем нажмите кнопку *Join*. Сущность и терминал оборудования исчезнут из соответствующих списков, а в списке *Present Joins* появится новая строка.

С другой стороны, чтобы удалить соединение, выделите строку в списке *Present Joins* и нажмите кнопку *Unjoin*: строка будет удалена, логический терминал будет восстановлен в списке первой части, а физический ресурс – во второй. Для подтверждения связи нажмите *OK*, для отмены проделанных операций нажмите *Cancel*.

### 2.14.7 Кнопки

Программа соединения кнопок автоматически запускается во время работы функции *Check Join*, если хотя бы одна кнопка осталась неподключенной; также ее можно запустить вручную из меню, для этого выберите *Project/Join Tools/Buttons*.



Окно разделено на три части: *Buttons*, *Available Port* и *Present Joins*.

Список первой части содержит сущности проекта, которые еще не были связаны с ресурсами оборудования.

Список второй части содержит имеющиеся кнопки со следующими характеристиками:

- Node : номер узла, где располагается ресурс
- Can : шина, к которой присоединен контроллер
- Pin : логический указатель терминала
- Description : краткое описание из документации по руководству к использованию контроллера
- Hw name : name of the controller where the resource is found

Список третьей части содержит уже созданные соединения между кнопками и физическими ресурсами.

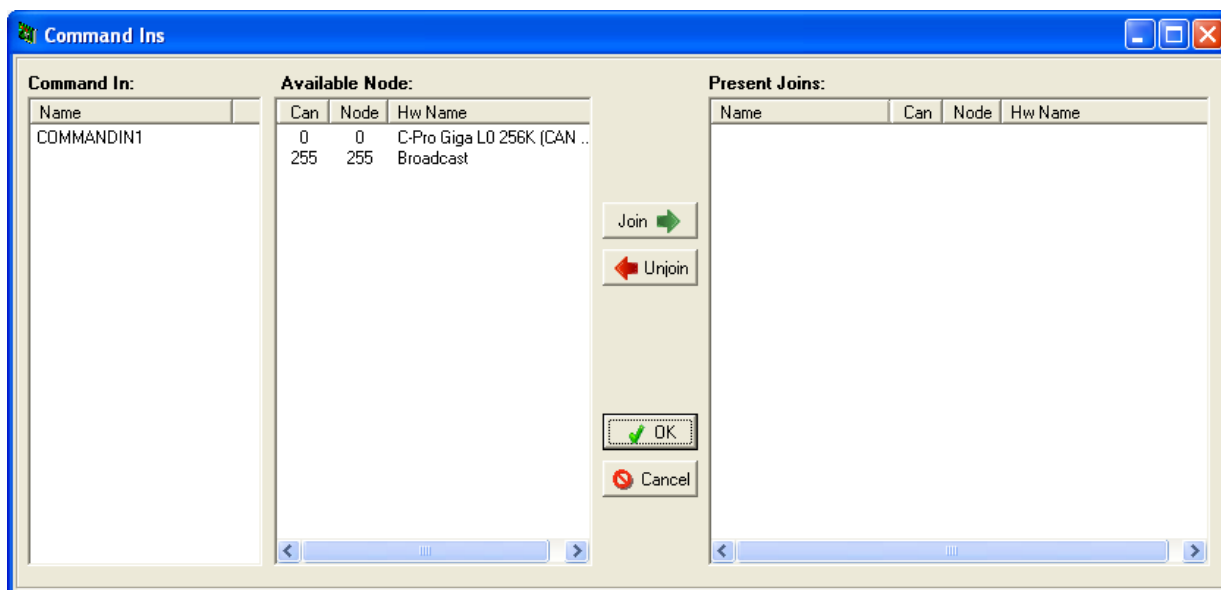
Чтобы связать сущность с ресурсом, выделите кнопку в списке первой части и физический ресурс в списке *Available Port*, затем нажмите кнопку *Join*. Сущность и терминал оборудования исчезнут из списков, а в списке «*Present Joins*» появится новая строка.

С другой стороны, чтобы удалить соединение, выделите строку в списке *Present Joins* и нажмите кнопку *Unjoin*: строка будет удалена, логический терминал будет восстановлен в списке первой части, а физический ресурс – во второй. Для подтверждения связи нажмите *OK*, для отмены проделанных операций нажмите *Cancel*.

#### 2.14.8 Входная команда

Программа соединения входной команды автоматически запускается во время работы функции *Check Join*, если хотя бы одна входная команда осталась не подключенной; также ее можно запустить вручную из меню, для этого выберите *Project/Join Tools/Command In*.





Окно разделено на три части: *Command In*, *Available Node* и *Present Joins*.

Список первой части содержит сущности проекта, которые еще не были связаны с узлом, от которого осуществляется получение команды.

Список второй части содержит все имеющиеся контроллеры (включая расширения), которые были добавлены в проект с использованием программы *Hardware Expert*, от которых Вы можете получать установленную команду. Каждый элемент этого списка имеет следующие характеристики:

- Node : номер узла, от которого принимается команда
- Can : шина, к которой присоединен контроллер
- Pin : описание узла контроллера, от которого принимается команда  
 Может принимать два типа значений:
  - Значение *Broadcast*, это означает, что выбранный входная команда может приниматься от любого контроллера сети
  - Имя типа контроллера связано с выводимым на дисплей *Can*-узлом
- Description : краткое описание из документации по руководству к использованию контроллера
- Hw name : имя контроллера, в котором располагается ресурс

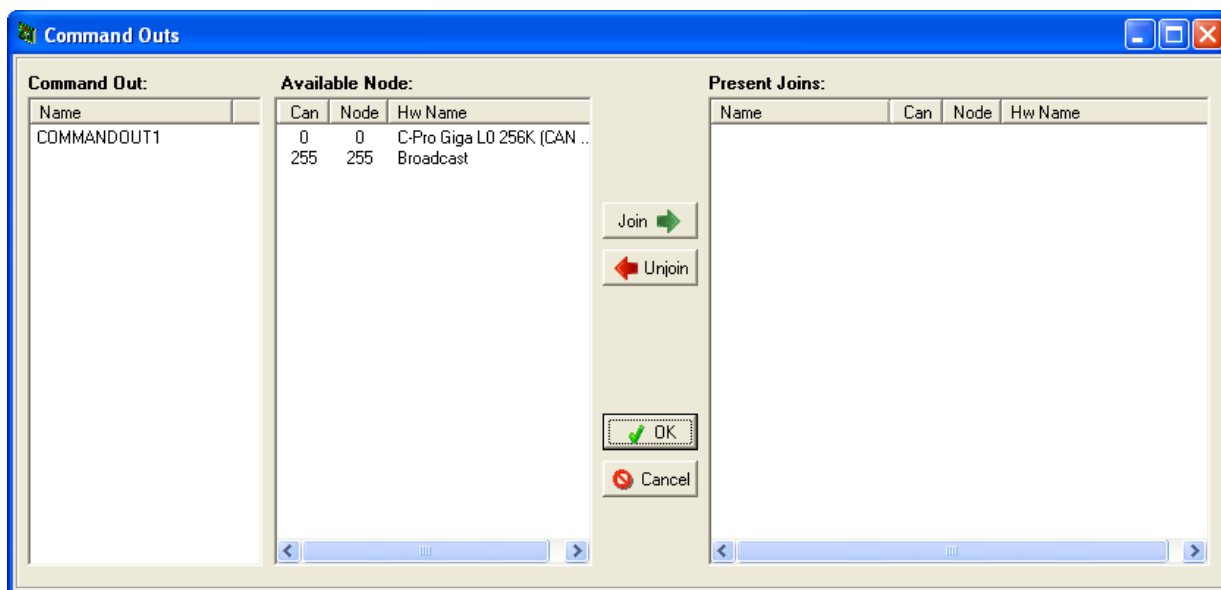
Список третьей части содержит уже созданные соединения входных команд и контроллеров.

Чтобы создать соединение, выделите входную команду в списке первой части, затем из списка *Available Node* выберите узел, от которого Вы хотите получать команду, и нажмите кнопку *Join*. Выделенный команда исчезнет из списка сущностей, а в списке *Present Joins* появится новая строка.

С другой стороны, чтобы удалить соединение, выделите строку в списке *Present Joins* и нажмите кнопку *Unjoin*: строка будет удалена и вход команды будет восстановлен в списке первой части. Для подтверждения связи нажмите *OK*, для отмены проделанных операций нажмите *Cancel*.

#### 2.14.9 Выходная команда

Программа соединения выходной команды автоматически запускается во время работы функции *Check Join*, если хотя бы одна выходная команда осталась не подключенной; также ее можно запустить вручную из меню, для этого выберите *Project/Join Tools/Command Out*.



Окно разделено на три части: *Command Out*, *Available Node* и *Present Joins*.

Список первой части содержит сущности проекта, которые еще не были связаны с узлом, на который передается команда.

Список второй части содержит все имеющиеся контроллеры (включая расширения), которые были добавлены в проект с использованием программы Hardware Expert и на которые Вы по выбору можете передавать установленную команду. Каждый элемент этого списка имеет следующие характеристики:

Node : номер узла, на который передается команда

Can : шина, к которой присоединен контроллер

Pin : описание узла контроллера, на который будет передаваться команда  
 Может принимать два типа значений:

- Значение Broadcast, это означает, что выбранная выходная команда может передавать команду на любой контроллер сети;
- Имя типа контроллера связано с выводимым на дисплей Can-узлом

Description : краткое описание из документации по руководству к использованию контроллера

Hw name : имя контроллера, в котором располагается ресурс

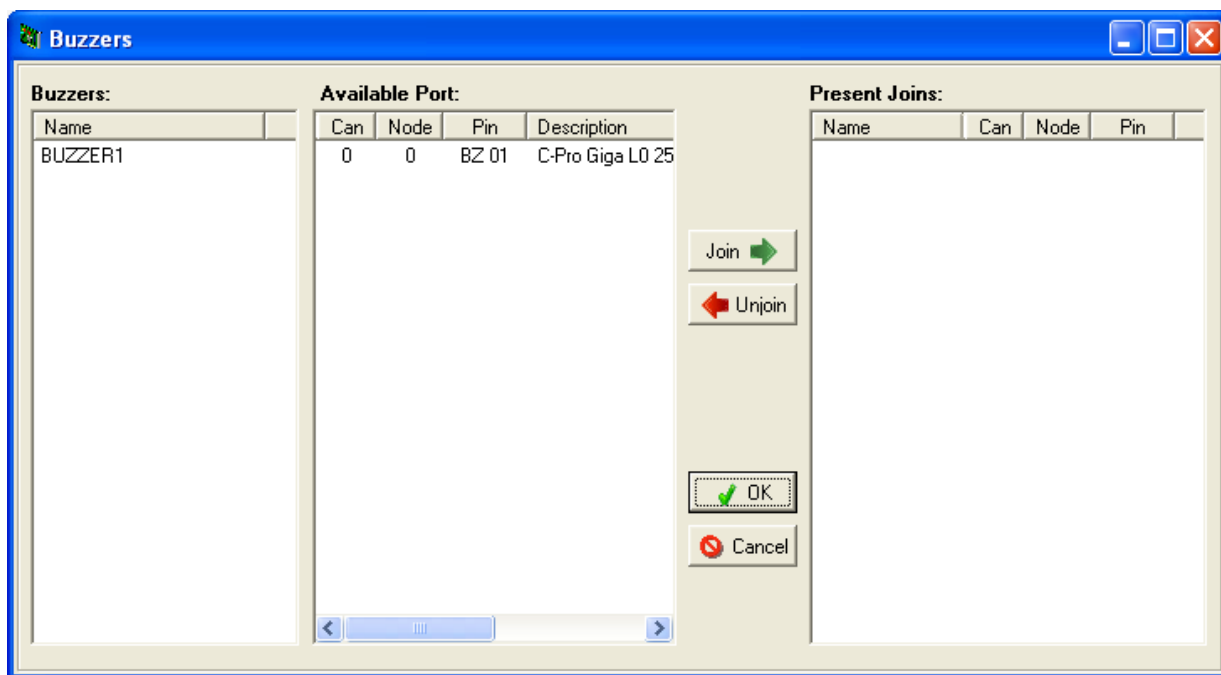
Список третьей части содержит уже созданные соединения выходов команды и контроллеров.

Чтобы создать соединение, выделите выход команды в списке первой части, затем в списке *Available Node* выделите узел, на который Вы хотите передавать команду, и нажмите кнопку *Join*. Выделенный выход команды исчезнет из списка сущностей, а в списке *Present Joins* появится новая строка.

С другой стороны, чтобы удалить соединение, выделите строку в списке *Present Joins* и нажмите кнопку *Unjoin*: строка будет удалена и выход команды будет восстановлен в списке первой части. Для подтверждения связи нажмите *OK*, для отмены проделанных операций нажмите *Cancel*.

#### 2.14.10 Звуковые сигналы

Программа соединения звукового сигнала автоматически запускается во время работы функции Check Join, если хотя бы один звуковой сигнал остался неподключенным; также ее можно запустить вручную из меню, для этого выберите *Project/Join Tools/Buzzers*.



Окно разделено на три части: *Buzzers*, *Available Port* и *Present Joins*.

Список первой части содержит сущности проекта, которые еще не были связаны с ресурсами оборудования.

Список второй части содержит имеющиеся звуковые сигналы со следующими характеристиками:


- Node : номер узла, где располагается ресурс
- Can : шина, к которой присоединен контроллер
- Pin : логический указатель терминала
- Description : краткое описание из документации по руководству к использованию контроллера
- Hw name : имя контроллера, в котором располагается ресурс

Список третьей части содержит уже созданные соединения между звуковыми сигналами и физическими ресурсами.

Чтобы связать сущность с ресурсом, выделите звуковой сигнал в списке первой части и физический ресурс из списка *Available Port*, затем нажмите кнопку *Join*. Сущность и терминал оборудования исчезнут из соответствующих списков, а в списке «*Present Joins*» появится новая строка.

С другой стороны, чтобы удалить соединение, выделите строку в списке *Present Joins* и нажмите кнопку *Unjoin*: строка будет удалена, логический терминал будет восстановлен в списке первой части, а физический ресурс – во второй. Для подтверждения связи нажмите *OK*, для отмены проделанных операций нажмите *Cancel*.

## 2.15 Компиляция программы

После того, как Вы сохранили проект, выбрали оборудование для его выполнения и успешно запустили программу *Check Join*, можно переходить к фазе компиляции; для этого выберите пункт меню *Project/Compile* или нажмите значок  на панели инструментов.

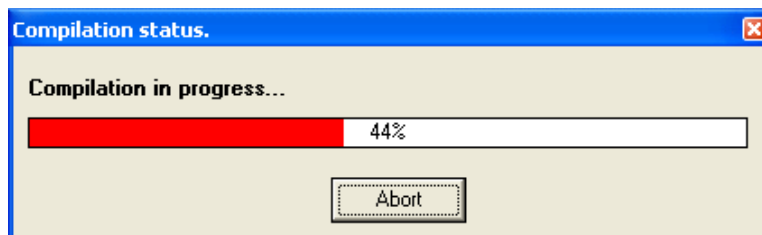
Фаза компиляции может быть разделена на пять стадий:

- Автосохранение проекта
- Создание файла-источника
- Компиляция

- Соединение
- Оценка доступных ресурсов

Если проект еще не был сохранен, появится запрос на введение имени проекта для его сохранения. Затем проект анализируется, создаются файлы-источники, которые будут скомпилированы и соединены для создания исполняемого файла, который, в свою очередь, будет загружен в контроллер с использованием функции загрузки. В заключение, оцениваются доступные ресурсы: неиспользованное количество памяти FLASH и RAM.

Во время компиляции файлов-источников на переднем плане появляется окно с индикатором выполнения, отображающим завершенность процесса в процентах, а также с кнопкой Abort для прерывания процесса.



В дополнение, в нижней части среды появится окно, в котором последовательно будет показан результат компиляции. Если процесс компиляции успешно завершен, то будут соединены объектные файлы, создан последовательно загруженный исполняемый файл, показан отчет обо всех доступных FLASH и RAM ресурсах.

```

Output
14.36.17: Checking hardware ... ok!
14.36.17: Checking joins ..... ok!
14.36.17: Checking UI ..... ok!
14.36.18: Saving projects ..... ok!
14.36.18: Generating code ..... ok!
14.36.18: Compiling ..... ok!
14.36.19: Linking ..... ok!
14.36.20:
14.36.20: Compilation end at 14.36.20 ( 0.00.01 ).
14.36.20: Errors: 0.
14.36.20:
14.36.20: ***** FREE *****
14.36.20: RAM   : 4286 bytes
14.36.20: FLASH : 168326 bytes
14.36.20: *****

```

Если процесс компиляции не был успешно завершен, то каждая обнаруженная ошибка вызовет появление

```

Output
14.38.30: Checking hardware ... ok!
14.38.30: Checking joins ..... ok!
14.38.30: Checking UI ..... ok!
14.38.31: Saving projects ..... ok!
14.38.31: Generating code ..... ok!
14.38.32: Compiling .....
*** RientroManuale (2) - Error 4062C : syntax error near `/'
14.38.33: Compilation end at 14.38.33 ( 0.00.00 ).
14.38.33: Errors: 1.

```

Каждая строка ошибки состоит из нескольких частей информации, которые могут быть интерпретированы



Category	: категория алгоритма, где была обнаружена ошибка
Line	: строка, в которой появилась ошибка
Error Type	: Информация, Предупреждение, Ошибка, Фатальная Ошибка
Error Code	: код, присвоенный ошибке ( <b>Compilation Errors/Ошибки компиляции</b> )
Error Description	: краткое описание ошибки

Щелкните два раза левой кнопкой мыши по строке ошибки, которую Вы хотите исправить; автоматически откроется CodeEditor алгоритма, в котором была обнаружена ошибка, и курсор будет расположен в соответствующей строке.

### 2.15.1 Использование кеша

В главном меню *Инструменты* и подменю *Настройки* можно задать использование кеша для компиляции файлов проекта. Для каждого проекта сохраняется кэш файла для компиляции и при компиляции будут перекомпилированы только те файлы, которые имели модификации. Этот вариант ускоряет процесс. Чтобы включить кеш, установите флаг *Включить кеш* в *Настройках*. Если желательно удалить весь кеш скомпилированных проектов, сделайте *Очистить кеш* в *Настройках* в меню *Инструменты*. Если требуется очистить только кеш открытого проекта, сделайте *Очистить кеш* из раздела *Опции* в свойствах проекта.

В главном меню *Проект* пункт *Очистить* позволяет удалить кеш открытого проекта и все временные файлы компиляции.

В главном меню *Проект* пункт *Компилировать все* позволяет полностью скомпилировать проект, предварительно запустив операцию очистки.

### 2.15.2 Компиляция с дисплеем EPJ-Color

Если предполагается дисплей EPJ-Color, то для полной компиляции проекта необходимо будет сделать две операции компиляции. Первая компиляция предназначена для контроллера, а вторая — для дисплея, поскольку цветные страницы EIML и другие проекты должны быть скомпилированы и загружены непосредственно в дисплее.

Процедура компиляции начинает компилировать файлы для контроллера; если эта фаза успешно завершена, через несколько секунд автоматически запускается компиляция для дисплея. После завершения компиляции создаются объектные файлы приложения. Объектные файлы (ucjb и ucje) дисплея отличаются от файлов контроллера из-за имени, которое заканчивается именем проекта «\_Browser».

Например, если проект TestProject состоит из контроллера C-Pro 3 Micro Basic и дисплея EPJ-Color 320x240, то результатом компиляции будет следующее:

Объектные файлы для контроллера:


- TestProject.ucjb
- TestProject.ucje

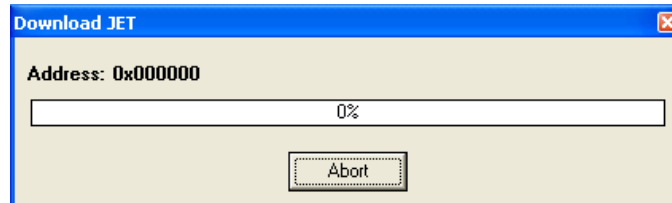
Объектные файлы для дисплея:

- TestProject\_Browser.ucjb
- TestProject\_Browser.ucje

## 2.16 Загрузка программы

### Для контроллеров не имеющих порта USB.

После завершения компиляции код выполнения может быть загружен в контроллер. Перед началом загрузки проверьте правильность соединения оборудования к последовательному порту и его готовность к выполнению программы (см. описание прибора). Чтобы начать программу загрузки, выберите в меню *Project/Download* или значок , расположенный на панели инструментов. Появится окно:



Если соединение с платой было успешно установлено, то через несколько секунд появится окно индикатора выполнения со степенью завершенности процесса в процентах; как только загрузка завершится, появится сообщение об успешном завершении операции.

В противном случае, появится сообщение об ошибке; тогда повторите попытку после проверки следующих условий:

- Последовательный порт, выбранный в опциях среды (может быть активирован из меню выбором *Tools/Settings*), должен быть тем, через который Вы хотите осуществлять загрузку;
- Серийный кабель должен быть подключен как к компьютеру, так и к контроллеру.

Если индикатор выполнения процесса остается на одном месте и появляется сообщение об ошибке, проверьте соединения, попробуйте перезагрузить контроллер и начать загрузку заново.

Чтобы поменять опции загрузки, обратитесь к параграфу **Конфигурация среды**.

### 2.16.1 Загрузка и программирование через USB

#### Для контроллеров с портом USB.

The procedure related here is valid only for the controllers c-pro 3 micro/kilo; these controllers can be connected to the Personal computer via USB and the binary generated when compiling is made of two files instead one.

1. *Project name.ucjb*: binary file of the application
2. *Project name.ucje*: binary file of the EIML pages

Since there are two files, the way to program changes; anyway the user will be asked for just one operation. The steps are:

- binary download of EIML pages (*Project name.ucje*)
- binary download of the application (*Project name.ucjb*)
- restart-up of the controller
- microcontroller memory programming, using the binary that have been found
- file generation with the programino report.

For these controllers it is not necessary to set any COM port (for programming/debugging); it is automatically recognized by UNI-PRO. It is very important to verify there is just one controller connected via USB because, otherwise, the wrong one could be programmed.

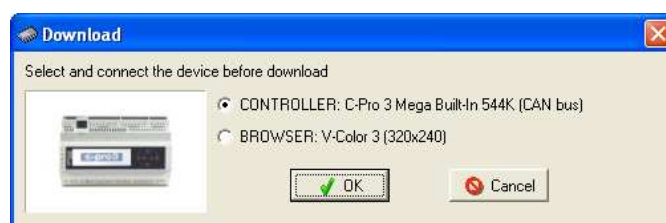
Once the procedure has finished, a text file is displayed; it contains a series of information on the programming procedure or a message relating the reason why it has not been possible to program the controller.

The visibility of this file depends on the status of the property "Show Report After Download" in the windows *Settings* of UNI-PRO.

COM-порт должен быть распознан автоматически, потому что после перезапуска контроллера он работает как накопитель данных (т.е. это не виртуальный COM). Именно среда разработки с помощью команд конвертирует его из ЗУ в виртуальный COM, получает значение и использует его в различных функциях, требующих последовательный порт.

### 2.16.2 Загрузка с дисплеем EPJ-Color

При наличии в сети CAN дисплея EPJ-Color компиляция создаст, помимо бинарных файлов для контроллера, еще и бинарные файлы для дисплея; это означает, что также должен быть запрограммирован и дисплей EPJ-Color. После выбора загрузки появится окно для выбора оборудования, используемого для загрузки.



До конца загрузки можно повторить процедуру для следующих устройств.

### 2.16.3 Загрузка и программирование через USB накопитель

Для всех контроллеров с USB-портом можно загрузить программу с помощью стандартного USB-накопителя (флешки), следуя приведенной ниже процедуре.

- Переименуйте бинарные файлы контроллера, сгенерированные во время компиляции:
  - o TestProject.ucjb
  - o TestProject.ucje
- Соответственно в:
  - o work.ucjb
  - o work.ucje
- Скопируйте их в корень флешки.
- Вставьте флешку в USB-порт с включенным питанием контроллера.
- Дождитесь автоматического копирования файлов во внутреннюю память и затем перепрограммирования контроллера.
- Эта процедура завершается, когда контроллер мигает во время автотеста сброса.

Флешка (флешка) должна быть отформатирована как FAT32.

#### ПРИМЕЧАНИЕ .

Во время процедуры программирования контроллер выполняет следующие тесты:

- Если файлы work.ucjb и work.ucje совпадают с теми, что находятся в памяти контроллера, они не копируются повторно и, естественно, контроллер не перепрограммируется.
- Контроллер проверяет соответствие двоичного файла после копирования.
- Контроллер проверяет соответствие выбранного оборудования двоичного файла и контроллера, чтобы избежать необратимых загрузок.

В случае компиляции с использованием дисплея EPJ-Color можно выполнить ту же процедуру программирования с использованием флэш-накопителя USB и для дисплея EPJ-Color .

- Переименуйте бинарные файлы дисплея, сгенерированные при компиляции:
  - TestProject\_Browser.ucjb
  - TestProject\_Browser.ucje

Соответственно в:

- work.ucjb
  - work.ucje
- Скопируйте их в корень флешки.
  - Вставьте флешку в USB-порт с включенным питанием дисплея.
  - Дождитесь автоматического копирования файлов во внутреннюю память и затем перепрограммирования дисплея.
  - Процедура завершается, когда на дисплее отображается страница инициализации во время автотеста сброса.

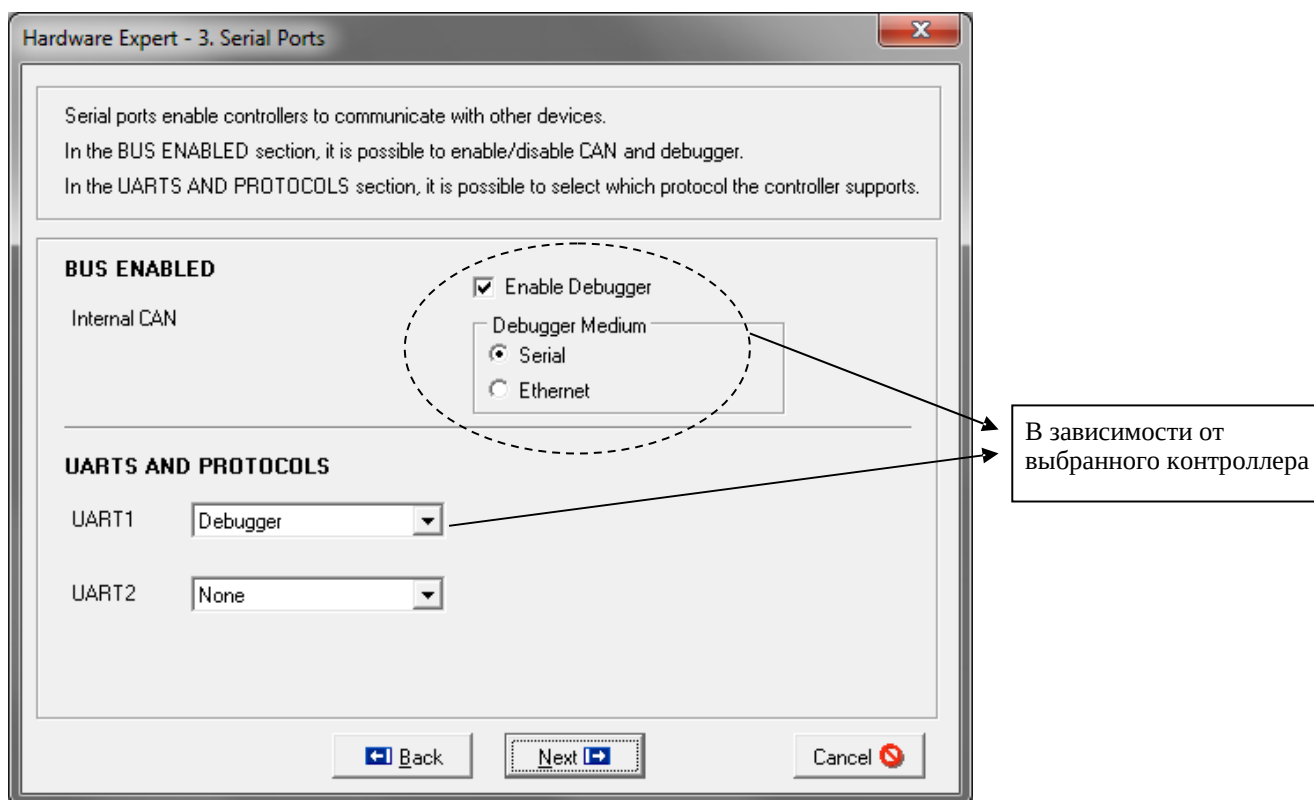


## 2.17 Использование отладчика

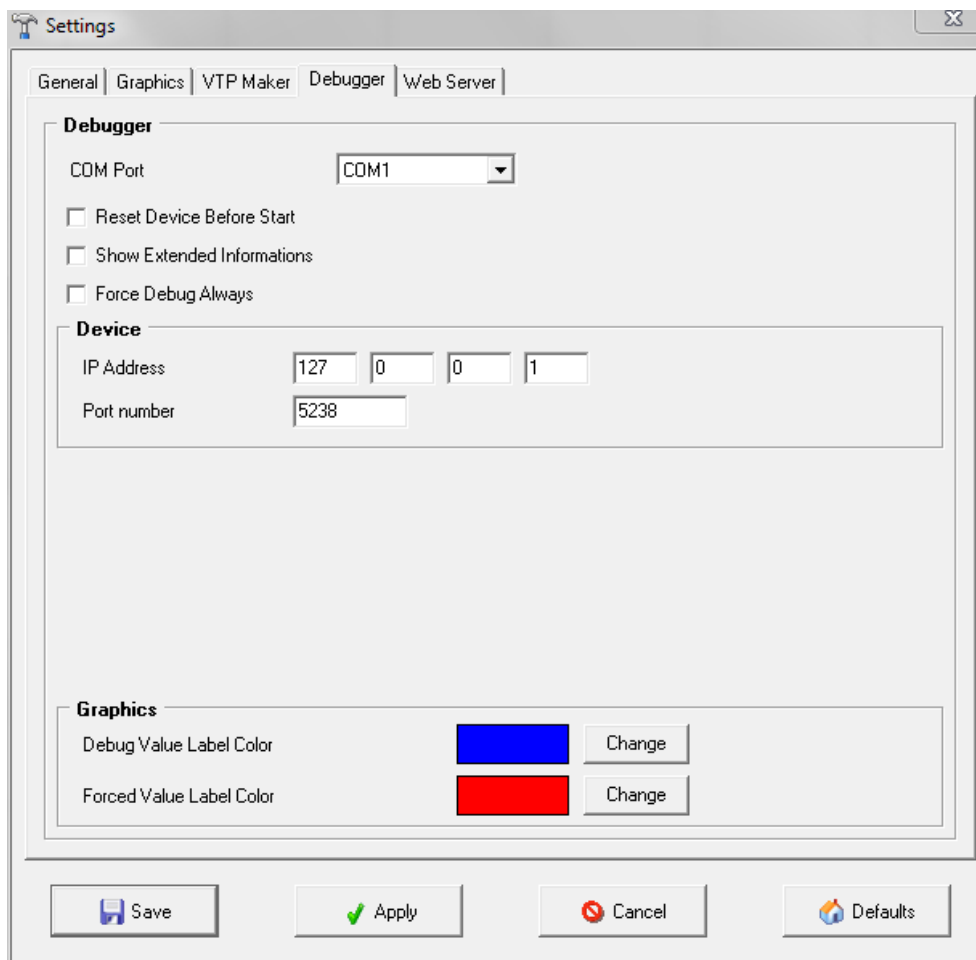
Отладчик позволяет проверить функционирование программы при её выполнении контроллером. Активация отладчика может проходить в 2 режимах: с помощью «перезапуска» (например, сразу после загрузки программы), либо «в режиме реального времени» (без остановки приложения). Последний вариант особенно полезен во время тестирования и тонкой настройки машинного управления.


При использовании отладчика состояние всех внутренних (переменные, таймеры, параметры) и внешних (входы и выходы) сущностей может отслеживаться прямо через графическую сущность, используемую при разработке. Значения параметров и статусы можно изменять, так же как и моделировать аналоговые и цифровые входы, прерывая ход программы с помощью условных остановов, делая скачок на выполнение операций для особых вычислений или начиная выполнение с самого начала. Целые сеансы отладки могут проводиться при изучении неисправностей функционирования или просто для ускорения фазы функциональных тестов при внесении поправок.

Для активации режима отладчика программа должна быть скомпилирована только после того, как протокол Отладчика будет выбран на одном из свободных последовательных портов контроллера. Другой способ – через активацию опции *Enable DEBUGGER* в разделе *BUS ENABLED*: данная операция выполняется в Эксперте оборудования в течение фазы конфигурации протокола последовательного порта (см. рисунок ниже).

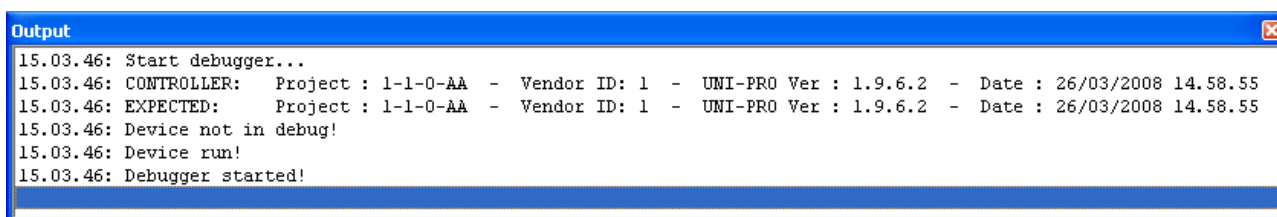


Также необходимо настроить последовательный COM-порт ПК, который будет использоваться для связи с контроллером. В случае отладчика через Ethernet необходимо также установить IP-адрес и TCP-порт для связи, используя следующую страницу, доступ к которой можно получить из Tools/Settings/Debugger меню:



Наконец, когда порт ПК будет соединен с портом контроллера с помощью соединения RS232 или RS485 или Ethernet, всё будет готово для начала сеанса отладки. Для его активации выберите в меню *Debug/Start Debugger* или значок  на панели инструментов.

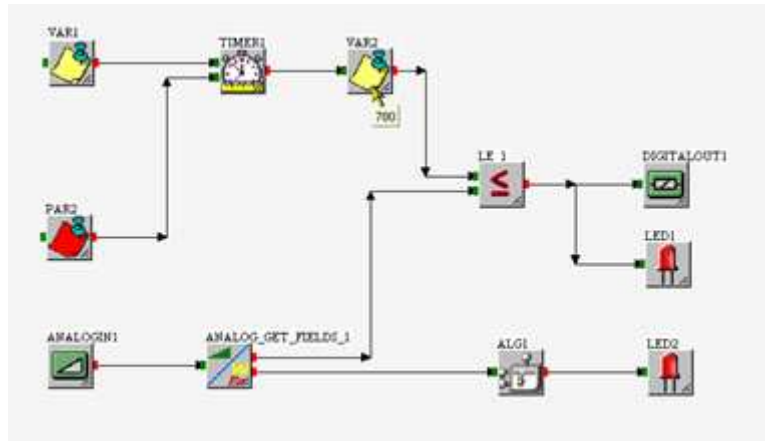
Программа выполнит соединение с контроллером и произведет спроектированное управление. Если проверка даёт положительные результаты, в выходном окне появится сообщение *Debugger started!*.




На командной панели появятся следующие значки:




В данный момент значения сущностей программы можно просмотреть с помощью наведения курсора мыши на переменные (следует дождаться появления жёлтого окна ввода).



Окно будет активно в течение нескольких секунд, для его повторного просмотра следует пошевелить мышью. Если Вы хотите посмотреть значение определённой сущности, Вам следует зайти в окно Просмотра с помощью специального меню (вызов правой кнопкой мыши) и выбрать Add Watch, либо щёлкнуть значок  на панели инструментов. Окно просмотра появится над окном состояния:

Entity	Value
VAR2	351

Output Watches


В окно Просмотра можно добавить до 4 переменных включительно. Удаление из окна происходит последовательно по одной с помощью специального меню (вызов правой кнопкой мыши) и выбора Remove Watch, либо с помощью значка  на панели инструментов. Все переменные окна просмотра могут быть удалены одновременно выбором *Debug/Remove All Watch* в меню.

При регулировании согласованности между проектом на ПК и проектом на контроллере сравнивается следующая информация:

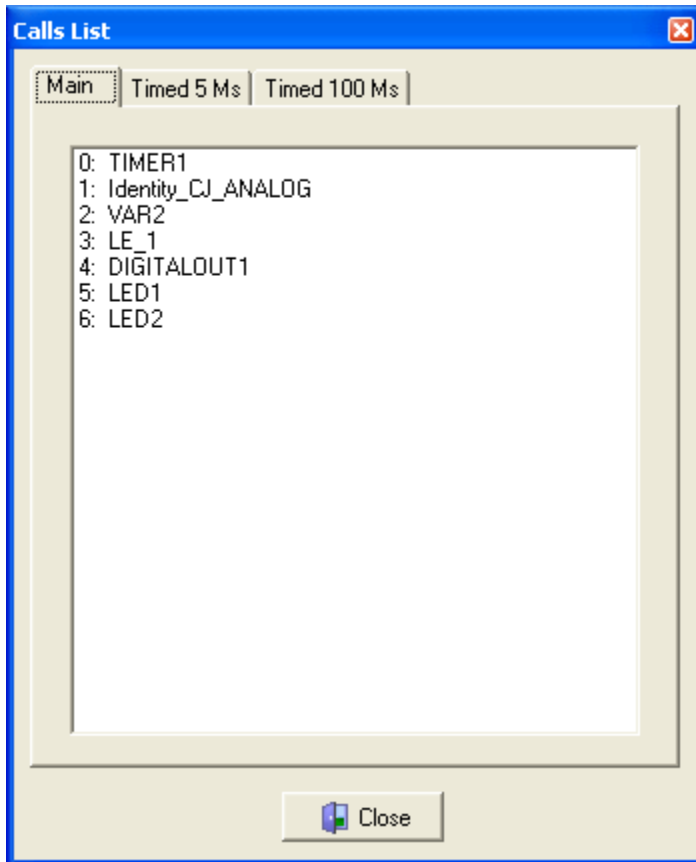
- Номер проекта
- Версия проекта
- ID поставщика
- Версия ПО UNI-PRO
- Дата компиляции
- Количество сущностей в проекте
- Количество задач в проекте

При этом 1, 6 и 7 пункты вышестоящего списка должны совпадать, иначе отладчик не будет запущен. В случае несовпадения остальных пунктов появится предупреждающее сообщение, но сеанс отладки будет активирован.

Для завершения фазы отладки выберите в меню *Debug/Stop Debugger* или нажмите значок  на панели инструментов.


Для прекращения выполнения программы выберите в меню *Debug/Break* или нажмите значок  на панели инструментов.


Программа остановится при выполнении сущности, индекс этой сущности будет указан на панели состояния: **Device break (1:4)**. Это относится к задаче, требуемой для вычисления, из Списка вызовов (Call List):





В приведённом примере выполнение программы было прервано во время вычисления сущности DIGITALOUT1, соответствующей задаче с номером 4.

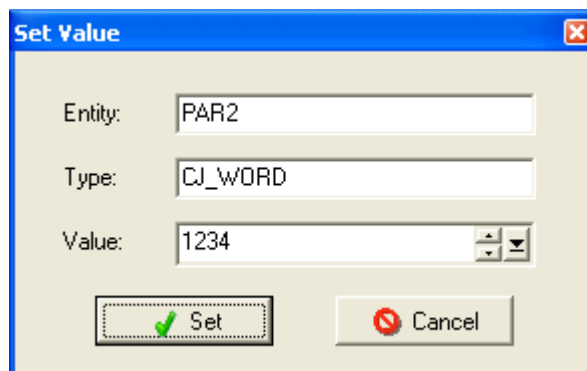
При остановке программы «замораживаются» все внутренние счётчики и таймеры, исключая входные данные и часы реального времени. Также возможно исследовать значение сущности или просматривать несколько сущностей в окне просмотра.

Существуют два способа перезапуска программы. Первый подразумевает выполнение куска программы до начала вычисления другой сущности с помощью команды *Debug/GoTo* или значка  на панели инструментов. В приведённом примере, если Вы хотите выполнять программу до вычисления сущности LED2, то будут произведены вычисления задач 5 и 6, после чего программа вновь будет прервана.

В качестве альтернативы, программу можно перезапустить продолжительным методом с помощью команды *Debug/Run* или значка  на панели инструментов. В этом случае программа будет выполняться с момента её прерывания.


Если Вы хотите начать выполнение программы с самого начала, необходимо произвести перезапуск контроллера с помощью команды *Debug/Reset* или значка  на панели инструментов. После этого перезапуск можно осуществить через команду *Run*. Операция перезапуска обнуляет все внутренние переменные, таймеры / счётчики точно так же, как после сбоя в электропитании.


Можно задавать значение выбранной сущности независимо от выполнения программы с помощью команды *Debug/Set Value* или значка  на панели инструментов. При этом появляется окно, в котором можно задать новое значение и подтвердить в письменной документации.

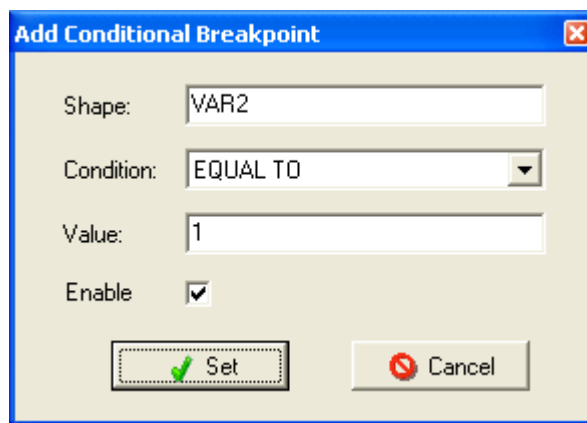


Значения аналоговых и цифровых входных значений также можно устанавливать. В этом случае вход, однажды установленный, будет перемещён в режим «Моделирование входа» и не будет больше принимать никакие другие значения, посылаемые от драйвера переноса данных контроллера. Такая ситуация будет продолжаться, пока Вы не выйдете из режима отладки.

Для выполнения сеанса отладки, в котором требуется исследовать причины возможной неисправности, очень важно иметь возможность добавлять точки прерывания в различных местах программы. В этом случае ПО сможет контролировать появление состояний прерывания и мгновенно останавливать программу контроллера, если эти состояния возникнут.

Точку прерывания можно задавать и без условий, чтобы программа могла остановиться точно в месте выполнения, в котором находится данная точка. Для добавления безусловной точки прерывания для определённой сущности следует выбрать эту сущность, выполнить команду *Debug/Set Breakpoint* или нажать значок  на панели инструментов. Этот метод полезен, если Вы хотите проверить поведение программы в пределах одного цикла или между основных циклов, останавливая программу на каждой точке прерывания.

Однако часто более полезным бывает исследование, что произойдёт при возникновении определённых условий. В этих случаях требуются условные точки прерывания, активирующиеся с помощью команды *Conditional Breakpoint* либо значка  на панели инструментов.




Условия можно выбрать из следующего списка:

<b>EXECUTE (ВЫПОЛНИТЬ)</b>	Условий нет (равно безусловной точке прерывания).
<b>EQUAL TO (РАВЕН ...)</b>	Программа остановится, когда сущность примет значение, заданное в поле Value.
<b>DIFFERENT FROM (ОТЛИЧАЕТСЯ ОТ ...)</b>	Программа остановится, когда сущность примет значение, отличное от значения, заданного в поле Value.
<b>LESS THAN (МЕНЬШЕ, ЧЕМ ...)</b>	Программа остановится, когда сущность примет значение, меньшее, чем заданное в поле Value.
<b>LESS OR EQUAL (МЕНЬШЕ ЛИБО РАВНО ...)</b>	Программа остановится, когда сущность примет значение, меньшее либо равное, заданному в поле Value.
<b>MORE THAN (БОЛЬШЕ, ЧЕМ ...)</b>	Программа остановится, когда сущность примет значение, большее, чем заданное в поле Value.
<b>MORE OR EQUAL (БОЛЬШЕ ЛИБО РАВНО ...)</b>	Программа остановится, если сущность примет значение, большее либо равное значению, заданному в поле Value.

При задании точки прерывания в верхней части окна состояния появится следующее окно:

Index	Entity Name	Condition	Value	Enable	BreakIdx	Idx	SubIdx
1	VAR2	EQUAL TO	1	Enable	0	15136	0

Output Breakpoints

Можно задать до 4 условий точки прерывания, которые будут отображены в этом окне. В нём также можно изменять свойства *Condition*, *Value* и *Enable*. Удаление точек прерывания производится по отдельности через специальное меню (вызов правой кнопкой мыши) или выбором *Remove Breakpoint*, или с помощью значка  на панели инструментов. Все точки прерывания могут быть одновременно удалены выбором пункта меню *Debug/Remove All Breakpoints*.

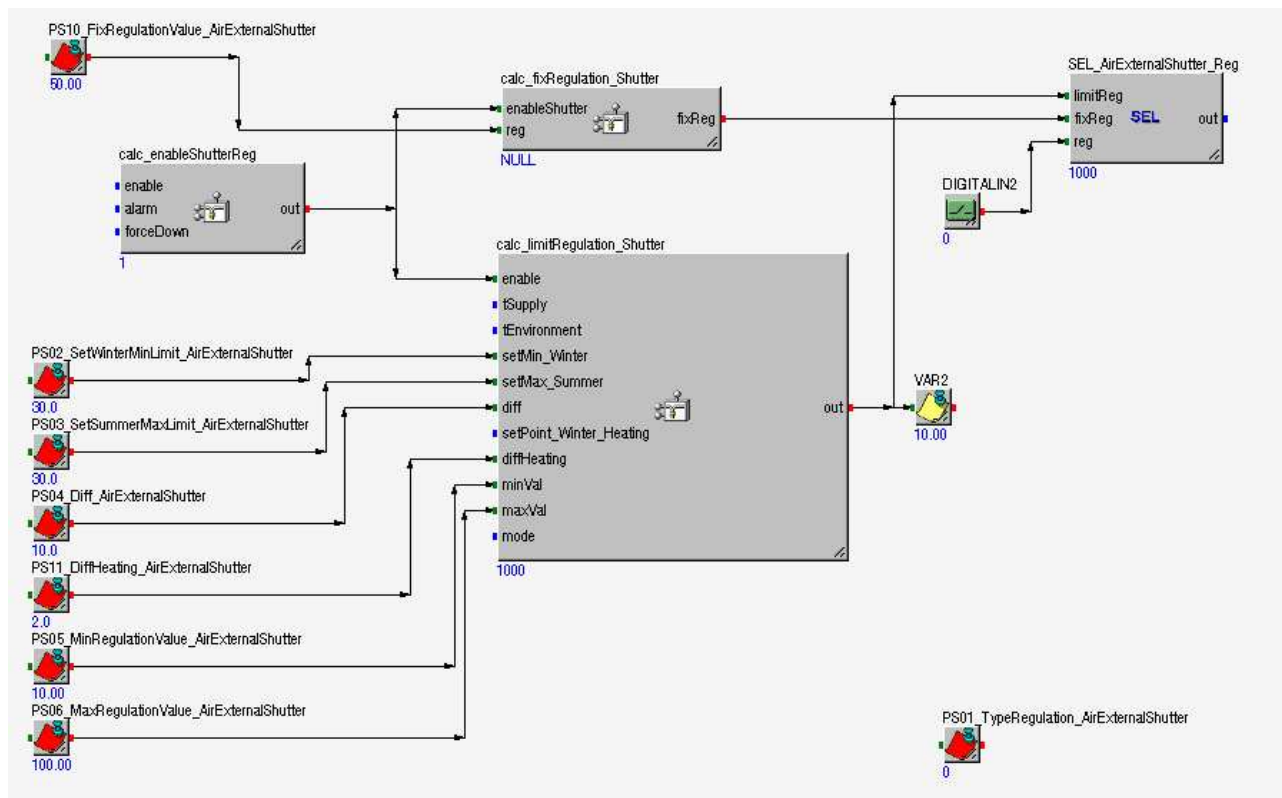
Условная точка разрыва также может быть задана и для возникновения особого события, такого как нажатие кнопки или поступление команды. В этом случае поле *Value* может принимать следующие значения:

- Value = 0      События нет
- Value = 1      Нажатие кнопки или поступление команды

**ПРИМЕЧАНИЕ:** в случае, когда выход алгоритма имеет более одной ветви, как, например, в предыдущем примере библиотеки *LE\_1*, система разработки выполнит промежуточные вычисления для данного выхода. Следовательно, во время фазы отладки можно установить для него точку прерывания; его значение также можно просмотреть с помощью мыши или в окне просмотра, как в случае неявных переменных. Тем не менее, для такого выхода невозможно применить *Set Value*.

### 2.17.1 Текущее значение при отладке

Во время отладки проекта для каждой вычисляемой сущности существует текущее значение – значение, которое принимает сущность во время цикла выполнения. Это предполагает целую серию значений, которые изменяются по ходу программы и которые можно просмотреть в любое время.



Если цифровому или аналоговому выходу присвоено определённое значение во время отладки, поэтому отключение от физической передачи данных приводит к подсвечиванию другим цветом.

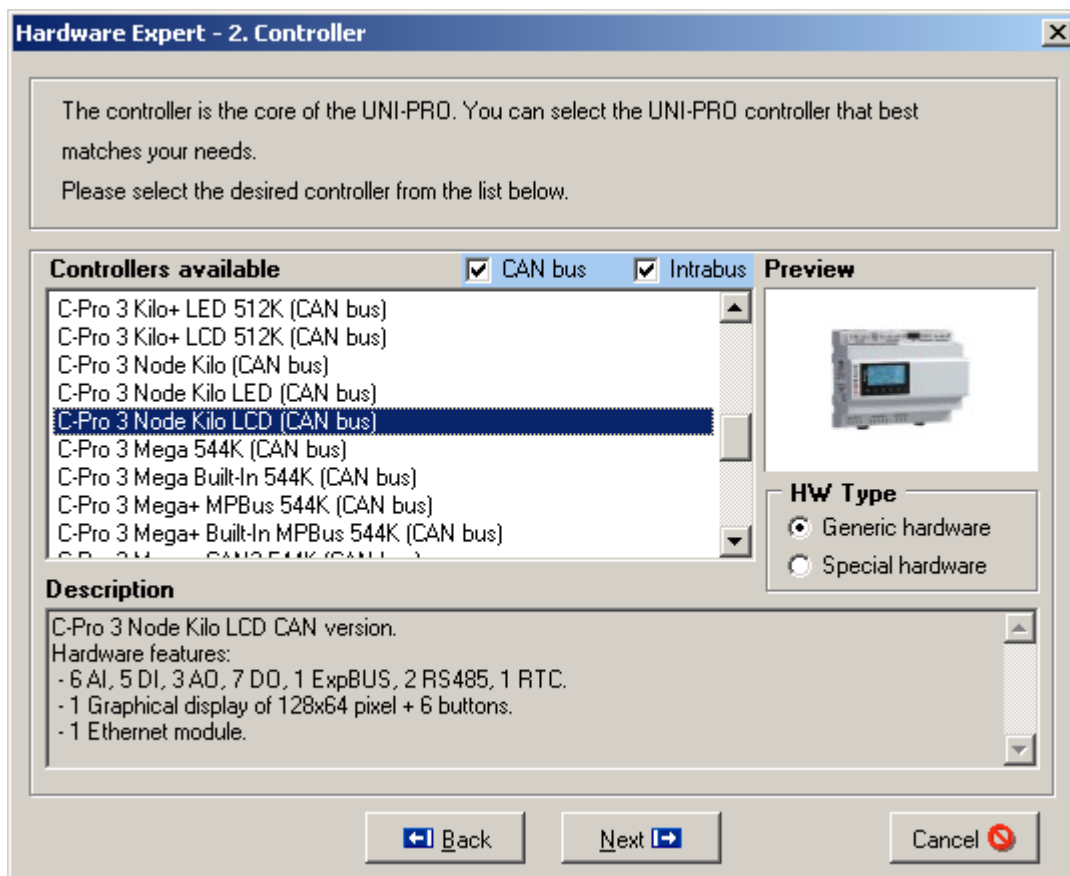
Цвет метки может быть изменён с помощью значения DEBUG в окне *Settings*, вкладка *Debugger*.

## 2.18 Web-сервер

Некоторые контроллеры с интерфейсом Ethernet могут быть доступны из браузера, который отображает находящиеся внутри контроллера веб-страницы. Устройство реализует минимальный web-сервер, который отвечает HTTP-клиентам, подключающимся из сети TCP/IP. Сервер публикует файлы, хранящиеся во флэш-памяти, посредством среды разработки, с помощью меню Project/Download WebSite.

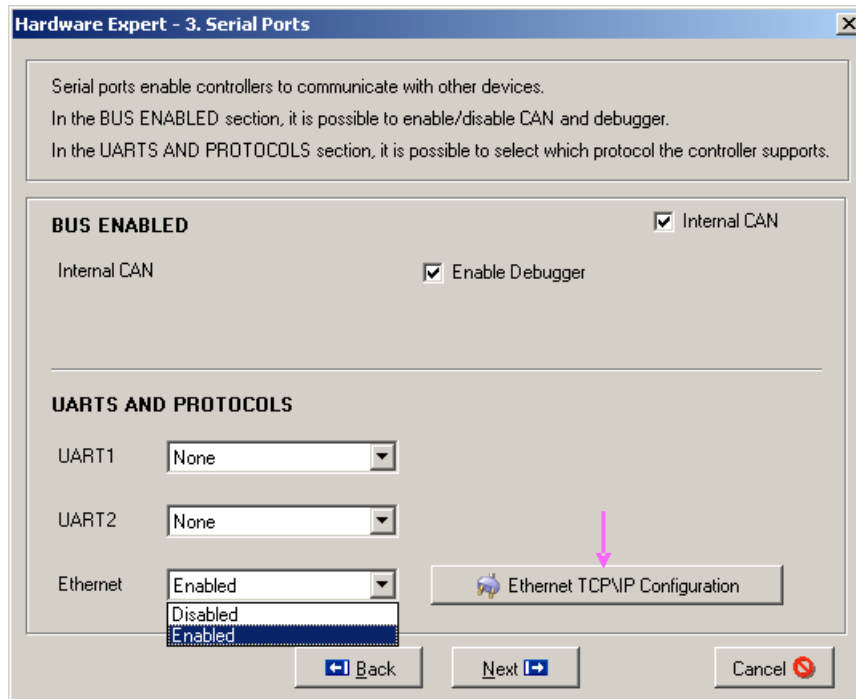
Благодаря таким инструментам, как AJAX и CGI, можно создавать интерактивные веб-приложения, с помощью которых можно просматривать информацию и изменять значения. Браузер становится расширением интерфейса, где пользователь выполняет некоторые операции. Предупреждение: браузер никогда не заменит внутренний интерфейс, особенно когда речь идет о правилах безопасности. Например, вы можете наблюдать за температурой в помещении и изменять задание скорости инвертора, управляющего вытяжкой воздуха. Контроллер теоретически доступен с любого терминала, подключенного к Интернету, даже с планшета, пока вы удобно сидите дома.

Открыв мастер *Hardware Expert*, выберите контроллер с интерфейсом Ethernet.

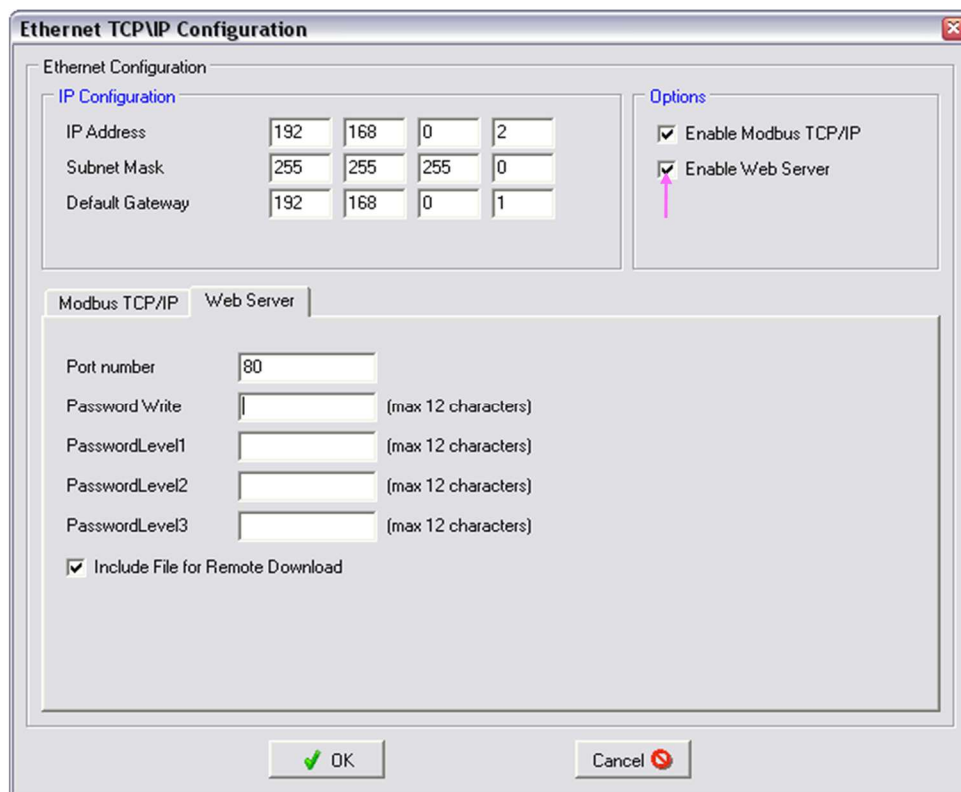




На следующем шаге щелкните поле со списком и включите *Ethernet*.  
Это делает видимой кнопку сразу справа.



При нажатии на кнопку *Конфигурация Ethernet TCP/IP* открывается следующее окно



Убедитесь, что установлен флажок *Включить web-сервер*.  
Номер прослушиваемого порта по умолчанию — 80; в большинстве случаев остается так.

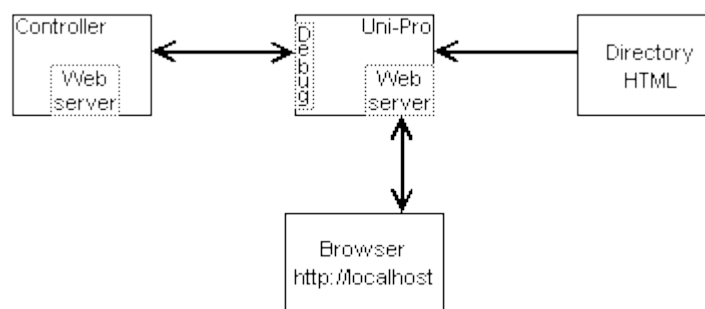
Поле Пароль содержит секретное слово для защиты от несанкционированной записи параметров. Поля «Уровень пароля 1», «Уровень пароля 2» и «Уровень пароля 3» позволяют предотвратить несанкционированный доступ к страницам и функциям с использованием трех уровней аутентификации. Если пусто, защита отключена.

Опция *Включить файл для удаленной загрузки* активирует или нет возможность включения в генерацию (и загрузку на контроллер) файла web-сервера для удаленной загрузки прошивки.

Html-страницы написаны специальными инструментами, требующими, как это обычно бывает, множества мелких и постепенных изменений, результат которых нужно каждый раз проверять. Для того, чтобы работа контроллера не замедлялась, нужно быстро публиковать страницы. Передача файлов на контроллер может быть неправильной как из-за медленной связи, так и из-за того, что она использует флэш-память, которая, несмотря на высокую номинальную производительность, имеет ограниченное количество циклов записи. Чтобы избежать этих проблем, был реализован симулятор веб-сервера.

### 2.18.1 Симулятор Web-сервера

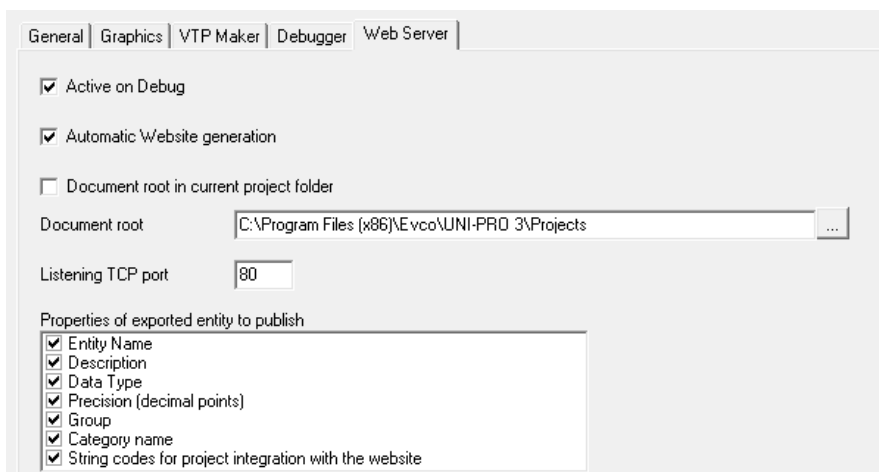
Uni-Pro встраивает web-сервер, работающий на компьютере, который имитирует поведение сервера, работающего в контроллере.



Автор сохраняет страницы в папку на жестком диске, модифицируя их по своему усмотрению. Uni-Pro рисует здесь в режиме реального времени, обновляя браузер. Имея возможность публиковать страницы локально, их написание становится легким.

Только когда требуется текущий тест или когда работа закончена, вы публикуете страницы в контроллере. Как только это будет выполнено, браузер взаимодействует через отладчик.

В меню *Инструменты/Настройки* выберите вкладку *web-сервер*.



Параметр *Активен при отладке* означает, что сервер запускается при запуске отладчика, то есть когда контроллер обменивается данными.

В поле *Источник документа* указана исходная папка по умолчанию, из которой разворачивается дерево веб-сайта. Параметр «Источник документа в текущей папке проекта» вместо этого создает корень веб-сайта в текущем пути проекта. Предположим, что он установлен в C:\Jobs\Uni-Pro и есть проект C:\Jobs\Uni-Pro\chiller\_alfa.ucjp. Если отсутствует, при запуске отладчика создается папка C:\Jobs\Uni-Pro\www\chiller\_alfa\. Другими словами, для проекта XYZ будет создана папка www\XYZ\, прикрепленная к указанной в общих настройках. Здесь должны быть сохранены файлы html, JavaScript, изображения и все остальное, что будет передано в контроллер.

### 2.18.2 Экспортируемые сущности

Сущности, которые можно читать и записывать через CGI, сначала экспортируются, щелкнув меню «*Инструменты/Экспорт объектов*» и отметив «X» в столбце *BMS*.

Используя автоматическое создание таблицы для Интернета, из меню *Проект/Создание Web таблицы элементов* в папке сайта создается файл с именем XYZ (экспортированный регистр var).txt в формате json, содержащий отмеченные элементы списка, показанного в предыдущей картинке. Этот файл можно было бы переименовать в XYZ.js, и это имя будет вызываться внутри HTML-страницы:

```
<script src='XYZ.js'></script>
```

Кроме того, необходимо объявить как переменную созданный таким образом массив записей, например:

```
var MyData =
{
  "tab_prop": [
    {
      "cat": "Par",
      "webServer": "",
      "group": "PARAM",
      "descr": "",
      "type": "CJ_LONG",
      "prec": "0",
      "name": "PAR1",
      "addr": "10"
    },
    ...
  ]
}
```

Файл не перезаписывается, если он существует, поэтому любые внесенные вами изменения сохраняются.

Мы выделяем свойства под названием *Group* и *WebServer*. В отличие от других, *Group* не существует как статическое свойство в Uni-Pro, а создается динамически.

Может оказаться полезным группировать объекты. Если не указано иное, имя предварительно заполняется в соответствии со следующим фиксированным правилом

Категория сущности: Var, Pers, Command In, Clock, Timer, Led, Buzzer, Button  
Контрольная группа: STATUS

Категория сущности: Digital input  
Контрольная группа: DIGITAL\_IN

Категория сущности: Digital output  
Контрольная группа: DIGITAL\_OUT

Категория сущности: Analog input  
Контрольная группа: ANALOG\_IN

Категория сущности: Analog output  
Контрольная группа: ANALOG\_OUT

Категория сущности: Par  
Контрольная группа: PARAM

*WebServer* — это свойство, предназначенное для обмена любыми данными между проектом и веб-приложением, поэтому его можно свободно использовать. Это также позволяет перезаписывать текущие значения других опубликованных свойств.

Предположим, есть сущность VAR1\_Long. Если свойство WebServer пусто, может быть создана следующая запись:

```
{
  "cat": "Var",
  "webServer": "",
  "group": "STATUS",
  "descr": "",
  "type": "CJ_LONG",
  "prec": "0",
  "name": "VAR1_Long",
  "addr": "23"
}
```

Установка указанного выше свойства в “Sierra; GrOUp=PUMP;06501; nAMe =Pump Hours; rtFoo;preC=1;” мы бы нашли в созданном файле

```
{
  "cat": "Var",
  "webServer": "Sierra;06501;rtFoo;",
  "group": "PUMP",
  "descr": "",
  "type": "CJ_LONG",
  "prec": "1",
  "name": "Pums Hours",
  "addr": "23"
}
```

Если *WebServer* содержит шаблон *key=value*; а *key* представляет собой одно из опубликованных свойств, тогда *value* заменяет его текущие данные.

### 2.18.3 Эксплуатация

Изменение общих настроек применяется при повторном запуске отладчика.

Локальный веб-сервер подключается либо к localhost, либо к 127.0.0.1.

При вводе `http://localhost` в адресной строке браузера сервер автоматически ищет `index.html` в качестве домашней страницы по умолчанию. Если URL-адрес содержит несуществующий путь, вы увидите код ошибки 404.

Для того, чтобы перенести сайт, нажмите в меню *Проект/Загрузить сайт*.

Следующие предварительные проверки выполняются и должны быть выполнены.

- 1) Папка не должна превышать максимальный размер, равный емкости контроллера (например, 2 МБ). В противном случае браузер отображает код ошибки 412, для устранения которого необходимо очистить и оптимизировать содержимое.
- 2) Имена файлов не могут быть длиннее 24 символов.
- 3) Сайт должен иметь один и только один уровень, т. е. вложенные папки не допускаются.

Веб-страница взаимодействует с контроллером через встроенную программу CGI.

Вызов предусматривает некоторые параметры, которые мы обнаруживаем прямо сейчас.

Внимание: синтаксис чувствителен к регистру. В то время как имя CGI пишется прописными буквами, то, что следует за знаком вопроса, пишется строчными буквами.

### 2.18.4 Чтение сущностей

Чтобы получить фактический статус всех экспортированных сущностей, введите

```
http://localhost/ODJET_CGI?pread=all
```

получение в качестве ответа массива в формате json

```
{
  "tab_param": [
    {"addr": "100", "value": "698", "min": "0", "max": "65535", "def": "0"},
    {"addr": "384", "value": "0", "min": "0", "max": "65535", "def": "0"},
    {"addr": "525", "value": "-32767", "min": "0", "max": "0", "def": "0"},
    {"addr": "1280", "value": "0", "min": "0", "max": "0", "def": "0"}
  ]
}
```

`addr` — это адрес регистра Modbus, а `def` — значение по умолчанию. Если вы хотите прочитать разрозненный набор сущностей, введите URL-адрес, например

```
http://localhost/ODJET_CGI?pread=6,8-10,12
```

получение в качестве ответа массива

```
{
  "tab_param": [
    {"min": "-32768", "max": "32767", "def": "89", "value": "89", "addr": "6"},
    {"min": "0", "max": "255", "def": "121", "value": "121", "addr": "8"},
    {"min": "-32768", "max": "32767", "def": "1354", "value": "1354", "addr": "9"},
    {"min": "-32768", "max": "32767", "def": "30059", "value": "30059", "addr": "10"},
    {"min": "-32768", "max": "32767", "def": "-30060", "value": "-30060", "addr": "12"}
  ]
}
```

в котором появляются адреса 6, 8, 9, 10 и, наконец, 12.

Параметр заранее принимает элементы, разделенные запятыми, каждый из которых может быть либо одним адресом, либо диапазоном, определяемым пределами (`lim`) `limA-limB`. Последний представляет собой непрерывную последовательность целых чисел от `limA` до `limB`.

Если экспортируемых сущностей нет или адрес неизвестен, ответ будет

```
{ "tab_param": [] }
```

### 2.18.5 Запись сущности

Предположим, вы пишете заглавную букву А в переменной типа CJ\_CHAR, экспортируемой в Modbus-адрес 8. Инструкция для запуска:

```
http://localhost/ODJET_CGI?pwrite=8&value=65
```

где параметр value может принимать исключительно числовые значения.

Браузер получает массив в формате json, содержащий рассматриваемую сущность. Ответ можно использовать для понимания, были ли данные успешно записаны или возникло непредвиденная ситуация.

```
{
  "tab_param": [
    { "min": "0", "max": "255", "def": "121", "value": "65", "addr": "8" }
  ]
}
```

### 2.18.6 Защита на запись

Чтобы узнать состояние защиты контента, введите URL

```
http://localhost/ODJET_CGI?ppsw=status
```

Если защита включена, ответ

```
{
  "tab_system": [
    { "ppsw": "lock" }
  ]
}
```

И наоборот, если защита выключена, ответ будет

```
{
  "tab_system": [
    { "ppsw": "unlock" }
  ]
}
```

Для активации защиты введите

```
http://localhost/ODJET_CGI?ppsw=lock&value=secret_word
```

где *secret\_word* — это пароль, установленный для проекта в Hardware Expert.

Чтобы отключить защиту, введите

```
http://localhost/ODJET_CGI?ppsw=unlock&value=secret_word
```

При блокировке и разблокировке ответ один и тот же, как при команде состояния.

Допустим защита включена и пароль 0\$0Zt6Q44.

Предположим, что при попытке снять блокировку мы отправляем контроллеру:

```
http://localhost/ODJET_CGI?ppsw=unlock&value=SierraTango
```

Получаем ответ

```
{
  "tab_system": [
    { "ppsw": "lock" }
  ]
}
```

потому что SierraTango не соответствует 0\$0Zt6Q44, блокировка сохраняется.

### 2.18.7 Защита Web-страниц

Есть еще две команды CGI для входа в систему и для проверки активного уровня пользователя.

Команда для чтения активного уровня пользователя:

```
/ODJET_CGI?stslev=1
```

Ответ

```
{
  "tab_loguser": [
    {"usrlog": "3", "result": "1", "err": "0"}
  ]
}
```

Где usrlog указывает уровень активного пользователя ( 1 = уровень 1, 2 = уровень 2, 3 = уровень 3). Значение 0 указывает, что ни один пользователь не вошел в систему. Результат поля, равный 1, указывает на отсутствие ошибок.

Команда для входа:

```
/ODJET_CGI?levpsw=Password&lvl= Level_Number
```

Где Password должен быть заменен цифровым паролем (с учетом регистра), а Level\_Number равен 1,2 или 3 в зависимости от запрошенного уровня аутентификации.

Вопрос

```
{
  "tab_loguser": [
    {"usrlog": "3", "result": "1", "err": "0"}
  ]
}
```

Где поля означают то же что и раньше.

В случае ошибок возвращаемое значение имеет следующий смысл:

- 1: запрошенный уровень не поддерживается
- 2 – 3 : количество символов больше максимального значения
- 4: ошибка введен неверный пароль
- 5: запрошенный уровень не активен (при попытке входа на неактивный уровень)

После правильной процедуры входа в систему навигация между веб-страницами этого уровня разрешена до фиксированного тайм-аута в 5 минут. По истечении этого времени запрос новой страницы отклоняется с кодом ошибки 401. Этот тайм-аут перезагружает каждую запрошенную страницу, но не во время обмена данными CGI.

Таким образом, можно оставаться на одной и той же защищенной странице в течение времени, превышающего тайм-аут.

### 2.18.8 Удаленное управление журналом данных

Две команды CGI можно использовать для чтения информации о состоянии журнала данных и для загрузки файла CSV удаленно.

Команда для чтения состояния журнала данных:

```
/ODJET_CGI?pdatalogf=1
```

Ответ web-сервера:

```
{
  "status_data_log": [
    { "num_rec": "23", "status": "ok_run" },
  ]
}
```

Описание полей:

“status”: uses the following return strings:

```
“no_link”
“ok_stop”
“ok_run”
“err_compatible_mem”
“err_rtc”
“err_corrupted_mem”
“err_insufficient_mem”
“busy” <-- in progress a USB data extraction
```

```
“num_rec”: != 0 <- number of records at the moment filled
0 <- no records
```

Команда для загрузки CSV-файла:

```
/ODJET_CGI?pdata_logf=csv
```

Отправьте команду в браузер с запросом на загрузку файла.

С этой командой нет ответов JSON.

**ПРИМЕЧАНИЕ.** Перед отправкой команды на загрузку файла важно убедиться в отсутствии ошибок в журнале данных и количестве записей больше нуля, в противном случае обратная связь в браузере может быть неправильно интерпретирована.

**ПРИМЕЧАНИЕ.** Рекомендуется выполнять операцию загрузки CSV в безопасном режиме (например, когда устройство остановлено), чтобы приложение не работало медленно во время подготовки файла CSV перед его передачей.

### 2.18.9 Информация о конфигурации сети

Можно запросить у web-сервера информацию: MAC-адрес и IP-адрес контроллера, а также сетевые конфигурации. Команда для использования следующая:

```
/ODJET_CGI?pnet=all
```

Ответ:

```
{
  "MAC_ADDRESS": "00:AC:DD:BB:CC:DD",
  "IP": "192.168.0.1",
```



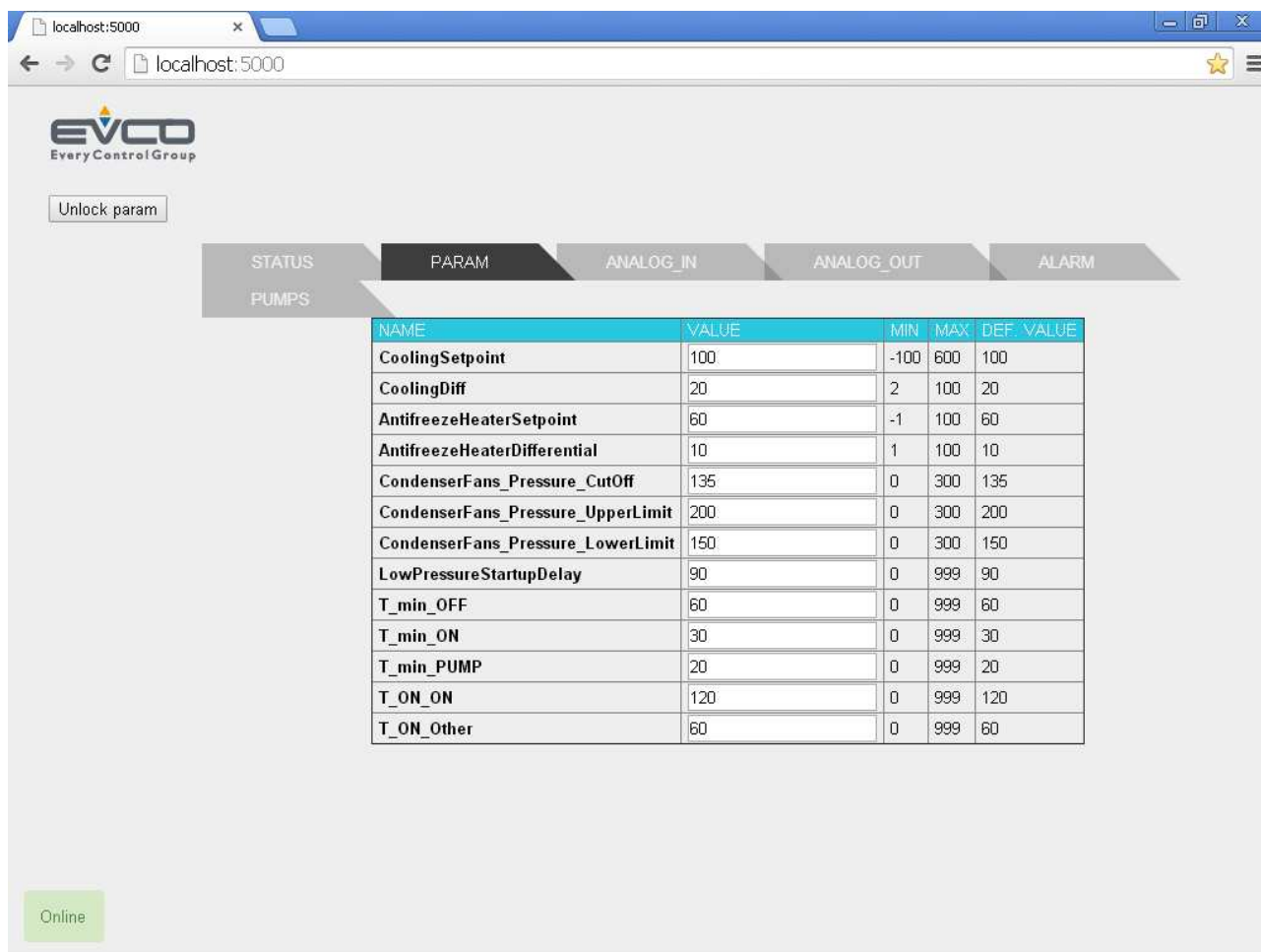
```
"SUBNET_MASK": "255.255.255.0",
"GATEWAY": "192.168.0.1"
}
```

Возможные параметры:

- all -> вся информация возвращена
- 0 -> вернуть только MAC
- 1 -> только IP адрес
- 2 -> IP , маска, шлюз вернуть

### 2.18.10 Автоматическая генерация Web-сайта

Включив опцию *Автоматическая генерация web-сайта*, можно создать web-страницу с таблицей, в которой можно сгруппировать всю информацию об экспортируемом объекте в столбцах, упорядоченных по типу принадлежности:



The screenshot shows a web browser window at localhost:5000 displaying the EVCO (Every Control Group) interface. The 'PARAM' tab is selected, showing a table of parameters. The table has columns for NAME, VALUE, MIN, MAX, and DEF. VALUE. The parameters listed are:

NAME	VALUE	MIN	MAX	DEF. VALUE
CoolingSetpoint	100	-100	600	100
CoolingDiff	20	2	100	20
AntifreezeHeaterSetpoint	60	-1	100	60
AntifreezeHeaterDifferential	10	1	100	10
CondenserFans_Pressure_CutOff	135	0	300	135
CondenserFans_Pressure_UpperLimit	200	0	300	200
CondenserFans_Pressure_LowerLimit	150	0	300	150
LowPressureStartupDelay	90	0	999	90
T_min_OFF	60	0	999	60
T_min_ON	30	0	999	30
T_min_PUMP	20	0	999	20
T_ON_ON	120	0	999	120
T_ON_Other	60	0	999	60

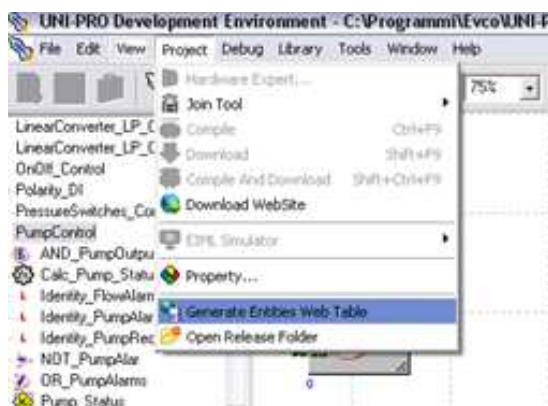
The interface also includes a navigation menu with tabs for STATUS, PARAM, ANALOG\_IN, ANALOG\_OUT, and ALARM, and a sub-menu for PUMPS. An 'Unlock param' button is visible, and an 'Online' indicator is shown at the bottom left.

Эта генерация происходит автоматически, вам нужно только выполнить необходимые шаги, без каких-либо знаний о языках или инструментах веб-программирования. Конечно, возможность персонализировать страницу ограничена выбором логотипа, названием и количеством групп, которые будут отображаться на вкладке, и именами объектов, как показано в таблице.

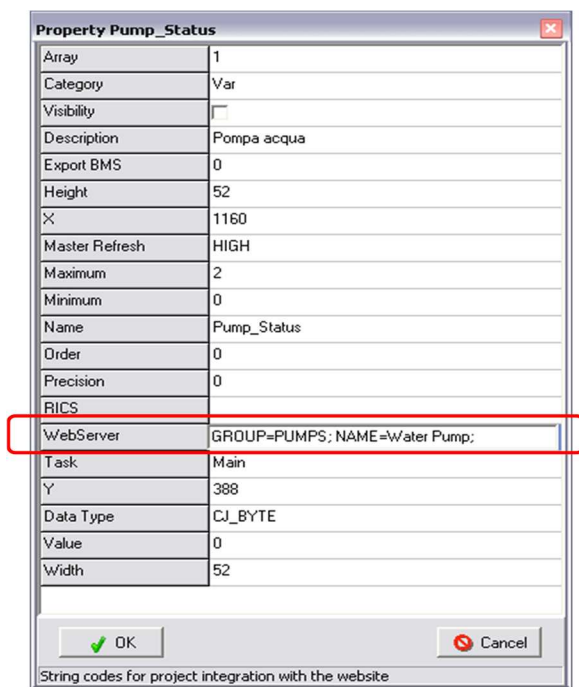
Можно будет читать значения, которые постоянно обновляются, и, защищенные паролем, их также можно будет записывать.

Объекты, которые могут отображаться в веб-таблице, являются единственными объектами, экспортируемыми в списке регистров с включенным параметром BMS.

Включив опцию *WebServer*, вы можете скомпилировать и загрузить программу. Затем вам нужно сгенерировать таблицу web-сущностей непосредственно из меню *Проект/Создать web-таблицу элементов*:



С помощью этой процедуры вы можете генерировать информацию, необходимую для создания таблицы и управления связью с переменными. При этом все файлы, необходимые для управления веб-сайтом, копируются из папки *Uni-Pro3\Config\Webserver files* в целевой каталог *www*. Конфигурация проекта возможна за счет использования специальных ключевых слов в свойстве *WebServer*:



Эти ключевые слова назначают представление таблицы, отличное от таблицы по умолчанию. Например, предыдущая сущность *Var* должна отображаться в столбце *STATUS* с именем *Pump\_Status*. Назначив *GROUP=PUMPS; НАЗВАНИЕ=Water Pump*; вместо этого он будет отображаться в столбце *PUMPS* с желаемым названием *Water Pump*.

NAME	VALUE	MIN	MAX	DEF. VALUE
PumpAlarm	0	0	1	
PumpThermalOverloadDelay	30	0	999	30
Water Pump	0	0	2	0

GROUP=PUMPS

Name=Water Pump

Ключевые слова, которые следует использовать, следующие:

- GROUP=XXXX; изменяет назначение группы или создает новую группу XXXX. GROUP поле не должно содержать пробелов.
- NAME=YYYY; изменяет имя по умолчанию (*Name*) на желаемое имя.
- DESC=YYYY; изменяет описание по умолчанию (*Description*) на желаемое описание

## 3 ОСНОВНЫЕ ЭЛЕМЕНТЫ

### 3.1 Типы данных

Типы данных, допустимые в среде программирования, могут быть разделены на две категории:

- Простые
- Структурные

Простые данные состоят из отдельных значений, которые могут быть напрямую использованы в алгоритмах с целью обработки, в то время как структурные данные образованы группой полей, содержащих несколько компонентов информации. Например, тип данных CJ\_ANALOG содержит два компонента информации: значение датчика и код ошибки датчика.

Недавно созданный тип данных CJ\_VOID, представленный системой программирования UNI-PRO, относится к категории простых данных.

#### 3.1.1 Типы простых данных

Простые данные, в свою очередь, могут быть разделены на две логические категории: первая категория включает все типы данных языка C, а вторая категория включает новые не структурные типы данных, созданные средой программирования UNI-PRO.

В приведенной ниже таблице сведены все типы данных, которые относятся к первой категории, и представлена информация по каждому типу:

Minus sign	: так или иначе может представлять отрицательные числа;
Repr.	: число бит, фактически используемых объектом этого типа;
Min	: минимальное принимаемое значение;
Max	: максимальное принимаемое значение;
Corr. ANSI C	: соответствующий ANSI C тип данных.

Тип данных	Minus sign	Представление	Min.	Max.	Соотв. ANSI C
CJ_BIT	NO	1 bit	0	1	
CJ_S_BYTE	YES	8 bits	-128	127	Signed char
CJ_BYTE CJ_CHAR	NO	8 bits	0	255	Unsigned char
CJ_SHORT	YES	16 bits	-32768	32767	Signed short
CJ_WORD	NO	16 bits	0	65535	Unsigned short
CJ_LONG	YES	32 bits	-2147483648	2147483647	Signed long
CJ_DWORD	NO	32 bits	0	4294967295	Unsigned long

Все операции, допустимые языком ANSI C, могут быть выполнены с представленными выше типами данных.

Новые типы данных (CJ\_VOID, CJ\_LED, CJ\_BUZZ, CJ\_DATE, CJ\_TIME, CJ\_DATETIME), созданные средой UNI-PRO, будут рассмотрены индивидуально.

### **CJ\_VOID**

Тип данных CJ\_VOID является прогрессивным решением, созданным средой программирования UNI-PRO, которое позволяет резко сократить затраты времени на программирование, а также обеспечивает высокий уровень гибкости.

Благодаря новому решению, Вы можете установить тип данных многофункционального объекта (такого, как переменная или входы алгоритма), просто соединив его с объектом, тип данных которого был задан заранее.

Например, если Вы добавляете переменную в проект, тип ее данных по умолчанию будет установлен как CJ\_VOID. Если Вы соединяете эту переменную с цифровым входом (установленным по умолчанию как CJ\_BIT), то тип данных переменной автоматически будет установлен как CJ\_BIT.

### **CJ\_LED and CJ\_BUZZ**

Типы данных CJ\_LED и CJ\_BUZZ очень схожи, именно поэтому они и рассматриваются вместе. Они представляют возможные значения, которые могут принимать индикаторы и звуковой сигнал соответственно.

Эти типы данных могут принимать значения в диапазоне от 0 до 3, что соответствует следующим состояниям:

- 0 : Выключен
- 1 : Непрерывно включен;
- 2 : Медленно мигает/издает звуковые сигналы;
- 3 : Быстро мигает/издает звуковые сигналы.

### **CJ\_DATE**

Тип данных CJ\_DATE был введен в работу для выполнения операций по датам; он показывает количество секунд, прошедших с полуночи 1 Января 2000, и может показывать даты до 2068 года. Использование этого типа данных может быть удобно, если, например, Вы хотите управлять операциями, основанных на определенных фиксированных датах.

Если Вы решили использовать этот тип данных в алгоритме, то проще всего будет использовать структуру CJ\_DATE\_STRUCT.

### **CJ\_TIME**

Тип данных CJ\_TIME был введен в работу для выполнения операций по времени; например, использование этого типа данных может быть удобным для управления диапазонами времени различных контроллеров и для множества других случаев. Он показывает количество секунд, прошедших с начала дня (00:00), и может быть легко преобразован в структуру CJ\_TIME\_STRUCT с помощью специальной функции преобразования.

### **CJ\_DATETIME**

Тип данных CJ\_DATETIME был введен в работу для случаев, когда необходимо выполнение операций как по датам, так и по времени одновременно; он показывает количество секунд, прошедших с полуночи 1 Января 2000 и может показывать даты до 2068 года.

Этот тип данных может быть использован в алгоритмах напрямую; кроме того, его можно преобразовать в структуру CJ\_DATETIME\_STRUCT с использованием функций библиотеки (см. раздел CJ\_DATETIME\_STRUCT), что позволяет облегчить Вашу работу.

### 3.1.2 Структурные типы данных

Структурные типы данных могут содержать множество компонентов информации, внедренных в среду программирования UNI-PRO. Они являются структурами C, состоящими из *n* полей, к которым можно получить доступ с использованием обычного C синтаксиса:

***structure.fieldname***

Ниже будут рассмотрены структурные типы данных и их значения.

#### CJ\_ANALOG

Тип данных CJ\_ANALOG описывает состояние аналогового входа. Структура состоит из двух полей:

**Error** – байтовый тип данных, описывающий код ошибки. Если это поле имеет значение ноль, значит, неисправность датчика отсутствует; иначе, оно примет следующие значения:

- 1 : короткое замыкание датчика;
- 2 : датчик не подключен или отсутствует.

**Value**, тип данных short, представляющий значения, считываемые датчиком.

Некоторые определения проекта могут ассоциироваться с полем значения (с использованием алгоритмов), чтобы управлять типом данных CJ\_ANALOG:

```
-32768: CJ_AI_ERR_GEN
-32767: CJ_AI_NOT_AVAIL
-32766: CJ_AI_ERR_CC_MIN
-32765: CJ_AI_ERR_OPEN_MIN
-32764: CJ_AI_DISABLED
-32763: CJ_AI_000001
-32762: CJ_AI_ERR_REF_MIN
-32761: CJ_AI_MIN_RESERVED
32767: CJ_AI_OK_OVERFLOW
32766: CJ_AI_ERR_CC_MAX
32765: CJ_AI_ERR_OPEN_MAX
32762: CJ_AI_ERR_REF_MAX
32761: CJ_AI_MAX_RESERVED
```

CJ\_AI\_MIN\_RESERVED и CJ\_AI\_MAX\_RESERVED: соответственно, минимальное и максимальное значение, за пределами которых значение поля датчика фиксирует ошибку. Если значение поля датчика не лежит в указанных пределах, датчику присваивается состояние ошибки (статус поля датчика положителен). Обладая данной информацией, можно сбросить поле ошибки с помощью поля значения, не прибегая к структуре CJ\_ANALOG. Эти значения также полезны, когда необходимо поменять значение датчиков в системе коммуникации CAN.

CJ\_AI\_DISABLED: значение, определяющее состояние отключения датчика. Если полю Value присвоено это значение, датчик будет отключен, а отображение EIML-страниц будет представлено многоточием: "...". Использовать это значение удобно, если Вы хотите задать зависимость между активностью датчика и возможным параметром активации.

#### CJ\_CMD

Тип данных CJ\_CMD является структурой, связанной с поступлением команды. Он состоит из следующих полей:

**Valid** – булевский тип данных, описывающий регистрацию команды. Если этот параметр TRUE, значит, команда была получена и могут быть выполнены необходимые действия; иначе, команда не была получена;

**Node** – байтовый тип, определяющий логический узел контроллера, от которого получена команда;

**Param** – тип short, описывающий параметр команды.

## CJ\_BTN

Тип данных CJ\_BTN является структурой, связанной с действиями, производимыми с кнопками: нажатие, удержание, отпускание.

Он состоит из следующих полей:

**Valid** - булевский тип данных, описывающий действие, произведенное с кнопкой (т.е., была ли она нажата, отпущена или удерживалась). Если он принимает значение TRUE, значит, действие, обнаруженное объектом Btn, было зарегистрировано; в противном случае команда не была получена;

**Node** - байтовый тип, определяющий логический узел, где было установлено воздействие на кнопку;

**Param** - тип short, описывающий количество секунд, в течение которых удерживалась кнопка.

## CJ\_DATE\_STRUCT

Тип данных CJ\_DATA\_STRUCT является очень удобным для выполнения операций по датам. Начиная с неструктурного типа данных CJ\_DATE, Вы можете разработать структуру CJ\_DATA\_STRUCT с помощью функции преобразования.

Он состоит из следующих полей:

**Day:** байтовый тип данных, определяющий день [1...31];

**Month:** байтовый тип данных, определяющий месяц [1=Январь, 2=Февраль,...12=Декабрь];

**Year:** байтовый тип данных, определяющий последние две цифры года, начиная с 2000 года. Например, если поле принимает значение 12, это означает «год 2012».

Эта структура обычно заполняется функцией преобразования DateToStruct(), которая имеет следующий синтаксис на языке C:

***CJ\_DATE\_STRUCT DateToStruct(CJ\_DATE Value);***

где параметр *Value* является датой, закодированной в секундах, начиная с полуночи 2000 года. Чтобы преобразовать эту структуру обратно в тип CJ\_DATE, используйте функцию StructToDate(), которая имеет следующий синтаксис:

***CJ\_DATE StructToDate(CJ\_DATE\_STRUCT date);***

## CJ\_TIME\_STRUCT

Тип данных CJ\_TIME\_STRUCT является очень удобным для выполнения операций по времени: например, для управления различными временными диапазонами. Начиная с неструктурного типа данных CJ\_TIME, Вы можете разработать структуру CJ\_TIME\_STRUCT с помощью функции преобразования.

Он состоит из следующих полей:

**Sec** - байтовый тип данных, определяющий секунды [0...59];

**Min:** байтовый тип данных, определяющий минуты [0...59];

**Hour:** байтовый тип данных, определяющий часы [0...23].

Эта структура обычно заполняется функцией преобразования TimeToStruct(), которая имеет следующий синтаксис на языке C:

***CJ\_TIME\_STRUCT TimeToStruct(CJ\_TIME Value);***

где параметр *Value* является временем, закодированным в секундах, начиная с полуночи того же дня. Чтобы преобразовать эту структуру обратно в тип CJ\_TIME, используйте функцию StructToTime(), которая имеет следующий синтаксис:

***CJ\_TIME StructToTime(CJ\_TIME\_STRUCT time);***

### **CJ\_DATE\_TIME\_STRUCT**

Тип данных CJ\_DATETIME\_STRUCT используется для преобразования типа данных CJ\_DATETIME (который представляет собой дату/время, закодированные в секундах) в более удобный формат для пользователя.

Эта структура обычно заполняется функцией преобразования DateTimeToStruct(), которая имеет следующий синтаксис на языке C:

***CJ\_DATETIME\_STRUCT DateTimeToStruct(CJ\_DATETIME Value);***

где параметр *Value* является датой-временем, закодированными в секундах, начиная с полуночи 2000 года.

Представляем краткое описание его полей:

**Sec:** байтовый тип данных, определяющий секунды [0...59];

**Min:** байтовый тип данных, определяющий минуты [0...59];

**Hour:** байтовый тип данных, определяющий часы [0...23];

**Day:** байтовый тип данных, определяющий день месяца [1...31];

**WeekDay:** байтовый тип данных, определяющий день недели [0=Воскресенье, 1=Понедельник, ... 6=Суббота];

**Month:** байтовый тип данных, определяющий месяц [1=Январь, 2=Февраль, ... 12=Декабрь];

**Year:** байтовый тип данных, определяющий последние две цифры года, начиная с 2000. Например, если поле принимает значение 12, это означает «год 2012».

Чтобы преобразовать структуру обратно в тип CJ\_DATETIME, используйте функцию StructToDateTime(), которая имеет следующий синтаксис:

***CJ\_DATETIME StructToDateTime(CJ\_DATETIME\_STRUCT rtc);***



### 3.2 Таблица совместимости типов

Когда Вы пытаетесь соединить две сущности, система проверяет, совместимы ли типы данных соединяемых сущностей. Таблица совместимости ниже используется, чтобы убедиться, является ли тип данных выхода совместимым с типом данных входа. Чтобы воспользоваться таблицей, найдите тип данных выхода, от которого вы начали проводить соединение, в левой части (текст выделен красным), и тип данных входа, к которому вы хотите подвести соединение, в верхней части (текст выделен зеленым). Если на пересечении строки и столбца в ячейке стоит X, это означает, что два типа данных совместимы и могут быть успешно соединены, иначе появится сообщение об ошибке и сущности соединены не будут.

Вышесказанное не относится к типу данных CJ\_VOID: как видно из таблицы, этот тип совместим со всеми остальными типами. При образовании соединения, в котором вход или выход имеет тип данных CJ\_VOID, среда программирования автоматически преобразует его в тип данных, соответствующий типу, с которым происходит соединение.

	CJ_VOID	CJ_BIT	CJ_BYTE CJ_CHAR	CJ_S_BYTE	CJ_LED	CJ_BUZZ	CJ_WORD	CJ_SHORT	CJ_ANALOG	CJ_DWORD	CJ_LONG	CJ_DATE	CJ_TIME	CJ_DATETIME	CJ_CMD	CJ_BTN
CJ_VOID	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
CJ_BIT	X	X	X	X	X	X	X	X		X	X					
CJ_BYTE CJ_CHAR	X		X				X	X		X	X					
CJ_S_BYTE	X			X			X	X		X	X					
CJ_LED	X		X	X	X		X	X		X	X					
CJ_BUZZ	X					X										
CJ_WORD	X						X			X	X					
CJ_SHORT	X							X		X	X					
CJ_ANALOG	X								X							
CJ_DWORD	X									X						
CJ_LONG	X										X	X	X	X		
CJ_DATE	X										X	X	X	X		
CJ_TIME	X										X	X	X	X		
CJ_DATETIME	X										X	X	X	X		
CJ_CMD	X														X	
CJ_BTN	X															X

Эта проверка совместимости типа данных выполняется только при создании соединения. Поэтому если Вы внесете изменения в тип данных входов или выходов после проверки, несовместимость может быть не обнаружена, что может привести к несоответствиям при выполнении программы.

Две сущности со свойством *array*, превосходящим 1, совместимы друг с другом только при выполнении следующих двух условий

1. если обе сущности имеют одинаковую размерность массива (значения свойства *array* совпадают);
2. если сущности имеют одинаковый тип данных за исключением:
  - a. если тип данных исходной сущности - CJ\_BIT, а тип данных соединяемой сущности - CJ\_S\_BYTE;
  - b. если тип данных исходной сущности - CJ\_BIT, а тип данных соединяемой сущности - CJ\_BYTE.

Во всех остальных случаях сущности несовместимы.

### 3.3 Функции встроенной библиотеки, используемые в алгоритмах

#### 3.3.1 CJ\_DATETIME StructToDateTime(CJ\_DATETIME\_STRUCT rtc)

Эта функция преобразует структуру данных CJ\_DATETIME\_STRUCT rtc в число, которое закодировано в секундах, начиная с полуночи 2000 года, что является типом CJ\_DATETIME.

#### 3.3.2 CJ\_DATE StructToDate (CJ\_DATE\_STRUCT date)

Эта функция преобразует структуру данных CJ\_DATE\_STRUCT date в число, которое закодировано в секундах, начиная с полуночи 2000 года, что является типом CJ\_DATE.

#### 3.3.3 CJ\_TIME StructToTime (CJ\_TIME\_STRUCT time)

Эта функция преобразует структуру данных CJ\_TIME\_STRUCT time в число, которое закодировано в секундах, начиная с полуночи 2000 года, что является типом CJ\_TIME.

#### 3.3.4 CJ\_DATETIME\_STRUCT DateTimeToStruct(CJ\_DATETIME Value)

Эта функция выдает структуру данных CJ\_DATETIME\_STRUCT, созданную из значения *Value* параметров типа CJ\_DATETIME, используемых в качестве входных данных.

#### 3.3.5 CJ\_DATE\_STRUCT DateToStruct (CJ\_DATE Value)

Эта функция выдает структуру данных CJ\_DATE\_STRUCT, созданную из значения *Value* параметров типа CJ\_DATE, используемых в качестве входных данных.

#### 3.3.6 CJ\_TIME\_STRUCT TimeToStruct (CJ\_TIME Value)

Эта функция выдает структуру данных CJ\_TIME\_STRUCT, созданную из значения *Value* параметров типа CJ\_TIME, используемых в качестве входных данных.

#### 3.3.7 CJ\_BYTE CJ\_GetSeconds(CJ\_DATETIME dt)

Эта функция выдает количество секунд [0...59], содержащихся в параметре *dt* типа CJ\_DATETIME.

#### 3.3.8 CJ\_BYTE CJ\_GetMinutes(CJ\_DATETIME dt)

Эта функция выдает количество минут [0...59], содержащихся в параметре *dt* типа CJ\_DATETIME.

#### 3.3.9 CJ\_BYTE CJ\_GetHours(CJ\_DATETIME dt)

Эта функция выдает количество часов [0...23], содержащихся в параметре *dt* типа CJ\_DATETIME.

#### 3.3.10 CJ\_BYTE CJ\_GetDay(CJ\_DATETIME dt)

Эта функция выдает значение дня [1...31], содержащееся в параметре *dt* типа CJ\_DATETIME.

#### 3.3.11 CJ\_BYTE CJ\_GetWeekDay(CJ\_DATETIME dt)

Эта функция выдает день недели [0=Воскресенье, 1..., 6=Суббота], содержащийся в параметре *dt* типа CJ\_DATETIME.

#### 3.3.12 CJ\_BYTE CJ\_GetMonth(CJ\_DATETIME dt)

Эта функция выдает значение месяца [1...12], содержащееся в параметре *dt* типа CJ\_DATETIME.

**3.3.13 CJ\_BYTE CJ\_GetYear(CJ\_DATETIME dt)**

Эта функция выдает значение года [00...68], содержащееся в параметре *dt* типа CJ\_DATETIME.

**3.3.14 CJ\_TIME CJ\_GetTime(CJ\_DATETIME dt)**

Эта функция выдает время CJ\_TIME, полученное из параметров *dt*, которое является количеством секунд, начиная с полуночи того же дня.

**3.3.15 CJ\_DATE CJ\_GetDate(CJ\_DATETIME dt)**

Эта функция выдает дату CJ\_DATE, полученную из параметров *dt*, которая является количеством секунд, начиная с полуночи 2000 года до полуночи того же дня.

**3.3.16 CJ\_BIT CJ\_GetSecondTic (void) e CJ\_BIT CJ\_GetMinuteTic (void)**

Эти две функции управляются средствами системного прерывания. Функции имеют те же критерии сущности-корреспондента TIMER, но они не используют память контроллера, так как управляются на уровне встроенных программ и, кроме того, могут быть напрямую использованы в алгоритме.

Функция *CJ\_BIT CJ\_GetSecondTic (void)* выдает логическое значение “1” для каждой прошедшей секунды.

Функция *CJ\_BIT CJ\_GetMinuteTic (void)* выдает логическое значение “1” для каждой прошедшей минуты.

Рекомендуется использовать эти функции, если проекту требуется большое количество сущностей TIMER, чтобы не перегружать функциональные возможности программы.

**3.3.17 CJ\_BIT CJ\_IsFirstMain(void)**

Возвращает 1 при полном первом прогоне главного цикла приложения.

**3.3.18 CJ\_WORD CJ\_SemaphoreRead(CJ\_WORD i)**

Считывает и возвращает значения *n*-ого семафора. Доступные семафоры в пределах 0 – 9.

**3.3.19 CJ\_BYTE CJ\_SemaphoreWrite(CJ\_WORD i, CJ\_WORD value)**

Записывает в *n*-ый семафор значение *value*. Запись семафора разрешена в том случае, если выполнено одно из следующих условий:

1. Семафор свободен либо имеет значение 0.
2. Значение для записи *value* равно 0.

Возвращает 0 = код ошибки (индекс за допустимыми пределами), Ok (запись выполнена), 2 = семафор занят (запись не разрешена).

**3.3.20 CJ\_BIT CJ\_FlagWrite (CJ\_WORD i, CJ\_BIT val) - CJ\_BIT CJ\_FlagRead (CJ\_WORD i)**

Эта функция позволяет управлять семафорами (*Semaphores*) в алгоритме. Допускается управление совместно разделенным между несколькими сущностями ресурсом, при этом каждая сущность должна знать состояние остальных сущностей для корректного использования разделенного ресурса. Что касается UNI-PRO, это свойство включает использование нескольких связей между алгоритмами или наличие сведений у каждого алгоритма о состоянии других алгоритмов, использующих одни и те же разделенные ресурсы в качестве входных данных. Данное решение имеет недостатки, так как использует большое количество памяти контроллера.

Для решения этой проблемы предлагается стандартное решение с использованием параллельного программирования: а именно, нескольких сущностей, называемых семафорами – *Semaphores*. *Semaphore* является структурой, способной организовать доступ к некоторым разделенным ресурсам для управления и назначения их соответствующим образом. Функциями для выполнения управления этими данными являются:

CJ\_BIT CJ\_FlagWrite (word i, byte val)

CJ\_BIT CJ\_FlagRead (word i)

Функция *CJ\_BIT CJ\_FlagWrite (word i, byte val)* позволяет установить состояние *i*-го семафора. Если *val*=0, семафор ‘свободен’, в связи с чем можно установить его состояние как ‘занят’ и затем использовать ресурсы.

Если  $val=1$ , семафор ‘занят’, в этом случае ресурсы использовать нельзя и необходимо подождать.

Например, для следующего вызова функции:

### **CJ\_FlagWrite (10, 1)**

состояние десятого семафора установлено как ‘занят’, поэтому ресурсы, управляемые этим семафором, будут доступны только для сущности, установившего данное состояние.

*Переключение состояния семафора с «занят» на «свободен» может осуществляться той же сущностью, что и установила это состояние.*

Вызов функции:

### **CJ\_FlagWrite (10, 0)**

состояние десятого семафора установлено как ‘свободен’, следовательно, открыт доступ к использованию ресурсов.

Состояние семафора управляется средствами следующей функции:

### **CJ\_BIT CJ\_FlagRead (CJ\_WORD i)**

это позволяет запрашивать состояние  $i$ -го семафора. Она выдает ‘1’, если семафор находится в состоянии ‘занят’, иначе выдает ‘0’. Если семафор находится в состоянии ‘занят’, следовательно, нельзя использовать контролируемые ресурсы до тех пор, пока они не освободятся.

#### **3.3.21 CJ\_SHORT CJ\_WriteVarExpo(CJ\_WORD add, CJ\_LONG value)**

Эта функция позволяет записывать напрямую из алгоритма значение “*value*” экспортируемой переменной в “*add*” адрес протокола Modbus. Таким образом, переменная модифицируется для всего алгоритма проекта. Данная особенность является аналогом концепции глобальной переменной – *global variable*, используемой в программировании.

Для корректного использования переменной необходимо, чтобы ее внесение в протокол Modbus осуществлялось с использованием функции *Export Entities*, активированной в меню *Tools/Export Entities* среды программирования.

#### **3.3.22 CJ\_LONG CJ\_ReadVarExpo(CJ\_WORD add)**

Эта функция позволяет считывать значение экспортируемой переменной в “*add*” адресе протокола Modbus. Функция является симметричной для *CJ\_WriteVarExpo (word add, long value)*. Для корректного использования переменной необходимо, чтобы ее внесение в протокол Modbus осуществлялось с использованием функции *Export Entities*, активированной в меню *Tools/Export Entities* среды программирования.

#### **3.3.23 CJ\_BYTE CJ\_ModbusAskQueue(void)**

Возвращает число свободных элементов очереди Modbus.

#### **3.3.24 CJ\_BYTE CJ\_SendCommand (CJ\_BYTE channel, CJ\_BYTE node, CJ\_BYTE command, CJ\_SHORT par1)**

Позволяет посылать команду изнутри алгоритма;

Возвращает 0 = команда отправлена; 1 = полная очередь;

Канал: 0 = CAN локальный, 1 = CAN широкий (при наличии);

Узел: логический узел канала CAN;

Команда: индекс команды;

Par1: 16-битный параметр, ассоциированный с командой

#### **3.3.25 CJ\_WORD CJ\_MaxMainTime(void)**

Возвращает максимальное время цикла аппликативной подачи питания в миллисекундах.

**3.3.26 CJ\_WORD CJ\_MinMainTime(void)**

Возвращает минимальное время цикла аппликативной подачи питания в миллисекундах.

**3.3.27 CJ\_WORD CJ\_RunMainTime(void)**

Возвращает текущее время цикла аппликативной подачи питания в миллисекундах.

**3.3.28 CJ\_BYTE CJ\_MaxInterruptTime(void)**

Возвращает максимальное время цикла аппликативного прерывания в миллисекундах.

**3.3.29 CJ\_BYTE CJ\_MinInterruptTime(void)**

Возвращает минимальное время цикла аппликативного прерывания в миллисекундах.

**3.3.30 CJ\_BYTE CJ\_RunInterruptTime(void)**

Возвращает текущее время цикла аппликативного прерывания в миллисекундах.

**3.3.31 CJ\_BIT CJ\_Stack\_Error\_Read(void)**

Возвращает 1 при переполнении стека в ходе программы.

**3.3.32 CJ\_BIT CJ\_Math\_Error\_Read(void)**

Возвращает 1 при возникновении ошибок DivByZero, Overflow, Underflow или NaN.

**3.3.33 CJ\_BIT CJ\_DivByZero\_Error\_Read(void)**

Возвращает 1 в случае деления на ноль.

**3.3.34 CJ\_BIT CJ\_Overflow\_Error\_Read(void)**

Возвращает 1 в случае переполнения разрядов.

**3.3.35 CJ\_BIT CJ\_Underflow\_Error\_Read(void)**

Возвращает 1 в случае исчезновения разрядов.

**3.3.36 CJ\_BIT CJ\_NaN\_Error\_Read(void)**

Возвращает 1 в случае возникновения общей ошибки NaN («не число»).

**3.3.37 void CJ\_DivByZero\_Error\_Write(void)**

Позволяет задать математическую ошибку, вызванную делением на ноль.

**3.3.38 void CJ\_Overflow\_Error\_Write(void)**

Позволяет задать математическую ошибку, вызванную переполнением разрядов.

**3.3.39 void CJ\_Underflow\_Error\_Write(void)**

Позволяет задать математическую ошибку, вызванную исчезновением разрядов.

**3.3.40 void CJ\_NaN\_Error\_Write(void)**

Позволяет задать математическую ошибку, вызванную возникновением общей арифметической ошибки; NaN означает Not a Number («не число»). Например, извлечение квадратного корня из отрицательного числа.

**3.3.41 void CJ\_Math\_Error\_Reset(void)**

Сброс математической ошибки, вызванной ошибочными математическими функциями.

**3.3.42 void CJ\_DivByZero\_Error\_Reset(void)**

Сброс математической ошибки, вызванной делением на ноль.

**3.3.43 void CJ\_Overflow\_Error\_Reset(void)**

Сброс математической ошибки, вызванной переполнением разрядов.

**3.3.44 void CJ\_Underflow\_Error\_Reset(void)**

Сброс математической ошибки, вызванной исчезновением разрядов.

**3.3.45 void CJ\_NaN\_Error\_Reset(void)**

Сброс математической ошибки, вызванной исключением, полученным вследствие ошибки NaN («не число»).

**3.3.46 CJ\_DWORD CJ\_Read\_DI\_PulseCnt (CJ\_BYTE in)**

Возвращает количество импульсов, измеренных цифровым входом (примечание: in=0, DI=1).

**3.3.47 void CJ\_Clear\_DI\_PulseCnt (CJ\_BYTE in)**

Очистить количество импульсов, измеренных цифровым входом (примечание: in=0, DI=1).

**3.3.48 CJ\_WORD CJ\_Read\_DI\_Frequency (CJ\_BYTE in)**

Возвращает частоту [Герц] счетчика импульсов, подключенного к цифровому входу (примечание: in=0, DI=1).

**3.3.49 void CJ\_ModBus\_SetProfile(CJ\_WORD numProfile, CJ\_BIT currentStatus)**

Позволяет включать/отключать вызовы удаленного устройства Modbus, настроенного с помощью профиля Modbus Master, добавленного в список ModBus Master Network Config в мастере UniPro Hardware Expert.

- *numProfile* соответствует номеру профиля, который вы хотите включить/отключить, как указано в списке.

- *currentStatus*, чтобы установить значение 1 для включения или 0 для отключения.

**3.3.50 CJ\_DATETIME RTCLocalRead(void)**

Это позволяет считывать значение, выраженное в секундах, локального RTC.

**3.3.51 CJ\_SHORT RTCLocalWrite(CJ\_DATETIME val)**

Используется для записи значения, выраженного в секундах, локального RTC.

Возвращает 0, если успешна запись, иначе ошибка.

**3.3.52 CJ\_WORD CJ\_GetTimer1ms(void)**

Возвращает значение в миллисекундах счетчика свободного хода, который сбрасывается при 10000.

**3.3.53 CJ\_BIT EIMLinEdit(void)**

Возвращает 1, если страница встроенного дисплея в настоящее время редактируется.

**3.3.54 CJ\_BYTE USB\_PenDrivesConnect(void)**

Возвращает 1, когда USB-накопитель распознается как вставленный.

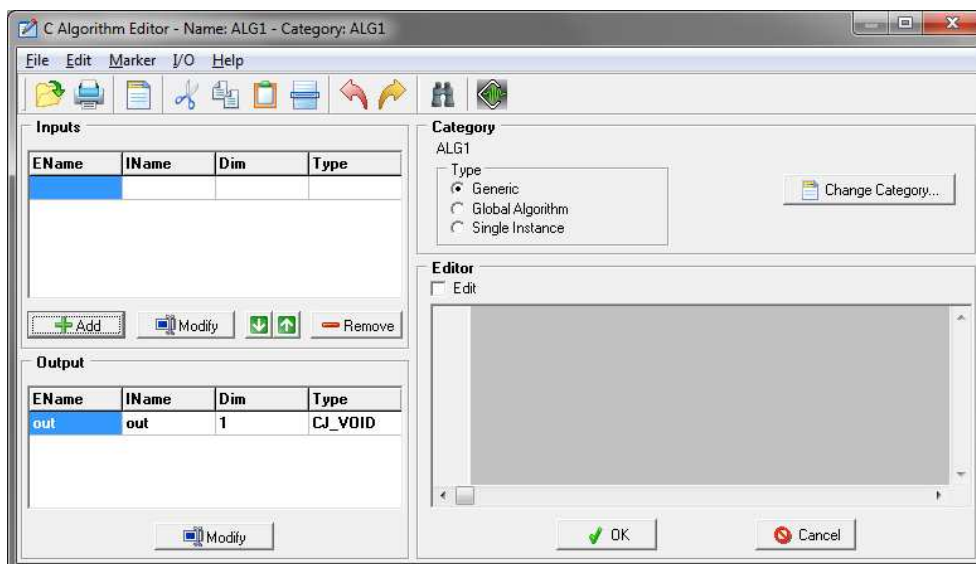
**3.3.55 void CJ\_AlxFilterWrite(CJ\_BYTE value)**

Позволяет изменить фильтр аналогового входа «х» относительно значения по умолчанию (обычно значение = 10). Применяется только к локальным входам, для расширений используйте инструмент CAN Master.

**3.3.56 void CJ\_AlxEnaFilterWrite(CJ\_BIT value)**

Позволяет включать/отключать фильтр аналогового входа «х».  
значение=1 включить (по умолчанию),  
значение=0 отключить





(для более подробной информации см. главу **Использование CodeEditor**).

### Глобальный алгоритм

Глобальный алгоритм – это алгоритм, который можно использовать в коде других алгоритмов; он подобен проектной функции, которую можно присоединять к другим частям кода. Преимущество использования глобальных алгоритмов заключается в том факте, что их можно вызывать из любого другого алгоритма проекта; это означает, что их можно использовать множество раз без повторной записи кода либо добавления новых алгоритмов. Для добавления глобального алгоритма следует выбрать флажок “Global Algorithm” в CodeEditor либо выполнить те же операции при создании нового алгоритма.

Глобальный алгоритм должен отвечать нескольким правилам:

- имя категории должно быть в формате языка “С”;
- глобальный алгоритм не должен являться алгоритмом категории; другими словами, невозможно использовать глобальный алгоритм, категория которого уже используется;
- терминалы глобальных алгоритмов не должны быть соединены;
- глобальный алгоритм может быть трансформирован в универсальный;
- универсальный алгоритм может быть трансформирован в глобальный, если он соответствует перечисленным выше правилам.

Все глобальные алгоритмы, используемые в проекте, можно найти с помощью функции “Find” или в CodeEditor в списке функций (*Ctrl*+пробел).

### Алгоритмы одного экземпляра

Алгоритмы такого типа — это алгоритмы, которые можно использовать в проекте только один раз, например, алгоритмы, содержащие статические операторы или уточнения, которые должны использоваться исключительно.

### Подсхема

Объекты подсхемы позволяют логически сгруппировать серии сущностей, таким образом разделяя проект на функциональные блоки.

Подсхемы могут содержать все типы сущностей, включая даже сами подсхемы, что позволяет организовать Ваш проект в виде многоуровневой структуры. Чтобы соединить сущности, находящиеся в одной подсхеме, применяются стандартные процедуры. Чтобы соединить внутренние и внешние сущности, необходимо экспортировать соответствующие входы/выходы. Экспортированные входы/выходы будут отображены другим цветом (черным, по умолчанию).

Вы можете добавить пустую подсхему в проект или сначала выделить сущности, которые необходимо сгруппировать, а затем использовать команду *Create Subsheet*.

Подсхемы являются основой для создания библиотек и шаблонов (см. главу **Создание Библиотек**).



**Fix (Привязка)**

В большинстве проектов необходимо задать константы; для этого можно использовать привязки. Они представляют область флэш-памяти, в которой Вы можете задать значение константы (таким образом, не влияя на размер RAM памяти, занимаемой проектом), которое может быть использовано в проекте.

Объекты-привязки имеют следующие свойства:

<b>Top</b>	Устанавливает положение, куда будет перемещен элемент, относительно верхнего края страницы.
<b>Array</b>	Размерность массива сущности. Если значение превосходит 1, то сущность является массивом, в противном случае представляет собой обычную сущность.
<b>Category</b>	Является свойством только для чтения; устанавливается для Fix.
<b>Show Out LIB</b>	Если он активен, он указывает этот объект как «общедоступный», когда он вставлен в библиотеку.
<b>Description</b>	Поле для текстового описания, куда Вы можете вносить собственные записи.
<b>Height</b>	Размер значка, представляющего сущность, по вертикали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.
<b>Left</b>	Устанавливает положение, куда будет перемещен элемент, относительно левого края страницы.
<b>MasterRefresh</b>	Время обновления сущности на главном сетевом трафике.
<b>Max</b>	Максимальное значение, которое можно присвоить константе. Зависит от типа.
<b>Min</b>	Минимальное значение, которое можно присвоить константе. Зависит от типа.
<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>Precision</b>	Определяет число десятичных знаков, используемое для расшифровки внутреннего значения эле
<b>Mb Access</b>	Указывает, активна ли защита чтения/доступа через Modbus.
<b>Mb Level</b>	Указывает уровень защиты от записи через Modbus (если level=0, защиты нет).
<b>WebServer</b>	Поле для вставки полезной информации для представления объекта на сайте контроллера.
<b>Timed</b>	Позволяет определить, в какой задаче вычисляется этот элемент и другие элементы, связанные с его входами. (Более подробную информацию см. в разделе «Определение задач выполнения»).
<b>Type</b>	Задает тип данных константы. Принимает значение по умолчанию CJ_VOID. Его можно изменить как автоматически, с помощью присоединения входа с типом данных, отличным от CJ_VOID, к выходу сущности, так и вручную, устанавливая этот параметр в окне свойств. В соответствии с выбранным типом данных константа будет иметь различные диапазоны и занимать различное количество флэш-памяти.
<b>Value</b>	Значение, присвоенное константе. Всегда проверяйте, совместимо ли оно с выбранным типом данных, т.е. находится ли оно в диапазоне, установленном критериями Min и Max.
<b>Width</b>	Размер значка, представляющего сущность, по горизонтали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.

**Значение под привязкой (константы)**

Под сущностью типа «привязка» (синего цвета) было добавлено значение, которое принимает форму константы.

FIX\_Konstant\_45



45

**Par (Параметры)**

Объекты Par представляют параметры прикладного программного обеспечения, которые позволяют изменять поведение устройства во время его работы. Параметры хранятся в памяти E2, поэтому все установленные значения сохраняются, даже если устройство будет выключено и включено заново.

Для каждого проекта максимально доступное число объектов PAR составляет 2047.

Они имеют следующие свойства:

<b>Top</b>	Устанавливает положение, куда будет перемещен элемент, относительно верхнего края страницы.
<b>Array</b>	Размер массива сущностей. Если значение больше 1 объекта, это массив, иначе это обычный объект.
<b>Category</b>	Является свойством только для чтения, устанавливается для Par.
<b>Condisible</b>	Если выбрано, разрешить использование параметра с условной видимостью.
<b>Show Out LIB</b>	Если он активен, он указывает этот объект как «общедоступный», когда он вставлен в библиотеку.
<b>Description</b>	Поле описания текстового типа, в которое вы можете вводить свои собственные записи.
<b>Height</b>	Размер значка, представляющего сущность, по вертикали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.
<b>Left</b>	Устанавливает положение, куда будет перемещен элемент, относительно левого края страницы.
<b>MasterRefresh</b>	Время обновления сущности на главном сетевом трафике.
<b>Max</b>	Максимальное значение, которое можно присвоить параметру. Зависит от типа.
<b>Min</b>	Минимальное значение, которое можно присвоить параметру. Зависит от типа.
<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>Parmax</b>	Ограничивает максимальное значение текущих параметров до значения другой сущности, имеющейся в приложении. Если установить как <NONE>, то максимальным пределом будет являться значение, установленное в свойстве Max.
<b>Parmin</b>	Ограничивает минимальное значение текущих параметров до значения другой сущности, имеющейся в приложении. Если установить как <NONE>, то минимальным пределом будет являться значение, установленное в свойствах Min.
<b>Order</b>	Порядок выполнения, применяемый для расчета этого элемента (см. параграф “Установка порядка выполнения программы”).
<b>Precision</b>	Определяет число десятичных знаков, используемое для расшифровки внутреннего значения элемента и для отображения значения в EIML-страницах.
<b>RICS</b>	Поле, куда вводится необходимая информация для создания драйвера VTP, требуемого для соединения с EVCO системой мониторинга R.I.C.S. и EVCO параметрами управляющего программного обеспечения <i>Params Manager</i> .
<b>Mb Access</b>	Указывает, активна ли защита чтения/доступа через Modbus.
<b>Mb Level</b>	Указывает уровень защиты от записи через Modbus (если level=0, защиты нет).
<b>WebServer</b>	Поле для вставки полезной информации для представления объекта на сайте контроллера.
<b>Timed</b>	Позволяет определить, в рамках какой задачи будет вычисляться данный элемент и другие элементы, связанные с его входными данными. (Для более подробной информации смотрите опцию <b>Определение задач выполнения</b> ).
<b>Type</b>	Задает тип данных параметра. Принимает значение по умолчанию CJ_VOID. Его можно изменить как автоматически, с помощью присоединения элемента с типом данных,

	отличным от CJ_VOID, к входу/выходу сущности, так и вручную, устанавливая этот параметр в окне свойств. В соответствии с выбранным типом данных параметр будет иметь различные диапазоны и занимать различное количество Е2 памяти.
<b>Value</b>	Значение, принимаемое параметром. Всегда проверяйте, совместимо ли оно с выбранным типом данных, т.е. находится ли оно в диапазоне, установленном критериями Min и Max (для более детальной информации, пожалуйста, обратитесь к опции <b>Принудительная загрузка параметров</b> ).
<b>Width</b>	Размер значка, представляющего сущность, по горизонтали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.

Чтобы изменить значение параметра, проделайте одну из следующих операций:

- присоедините другую сущность (например, алгоритм) к контакту входа;
- присоедините параметр к страницам EIML и включите его для редактирования
- используйте один из протоколов последовательной связи для запуска запроса на запись параметров

Примечание: если значение сущности, связанное с максимальным/минимальным пределом параметра, сдерживается динамикой страницы, то новый предел не будет установлен, до тех пор, пока страница не будет перезагружена в приложении.

### Pers (Постоянные)

Pers-объекты представляют все устойчивые состояния, т.е. все те значения, которые Вы хотите сохранить даже после перезагрузки устройства (например, время работы компрессора или состояние освещения).

Для каждого проекта максимально доступное число объектов PERS составляет 1535.

Pers-объекты имеют следующие свойства:

<b>Top</b>	Устанавливает положение, куда будет перемещен элемент, относительно верхнего края страницы.
<b>Array</b>	Размер массива сущностей. Если значение больше 1 объекта, это массив, иначе это обычный объект.
<b>Category</b>	Является свойством только для чтения; устанавливается для Pers.
<b>Condisible</b>	Если отмечено, то параметр может использоваться с обусловленной видимостью.
<b>Show Out LIB</b>	Если он активен, он указывает этот объект как «общедоступный», когда он вставлен в библиотеку.
<b>Description</b>	Поле для текстового описания, куда Вы можете вносить собственные записи.
<b>Height</b>	Размер значка, представляющего сущность, по вертикали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.
<b>Left</b>	Устанавливает положение, куда будет перемещен элемент, относительно левого края страницы.
<b>MasterRefresh</b>	Время обновления сущности на главном сетевом трафике.
<b>Max</b>	Максимальное значение, которое может принимать состояние. Зависит от типа.
<b>Min</b>	Минимальное значение, которое может принимать состояние. Зависит от типа.
<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>Order</b>	Порядок выполнения, применяемый для расчета этого элемента (см. параграф “Установка порядка выполнения программы”).
<b>Precision</b>	Определяет число десятичных знаков, используемое для расшифровки внутреннего значения элемента и для отображения значения в EIML-страницах.

<b>RICS</b>	Поле, куда вводится необходимая информация для создания драйвера VTP, требуемого для соединения с EVCO системой мониторинга R.I.C.S. и EVCO параметрами управляющего программного обеспечения <i>Params Manager</i> .
<b>Mb Access</b>	Указывает, активна ли защита чтения/доступа через Modbus.
<b>Mb Level</b>	Указывает уровень защиты от записи через Modbus (если level=0, защиты нет).
<b>WebServer</b>	Поле для вставки полезной информации для представления объекта на сайте контроллера.
<b>Timed</b>	Позволяет определить, в рамках какой задачи будет вычисляться данный элемент и другие элементы, связанные с его входными данными. (Для более подробной информации смотрите опцию <b>Определение задач выполнения</b> ).
<b>Type</b>	Задаёт тип данных параметра. Принимает значение по умолчанию CJ_VOID. Его можно изменить как автоматически, с помощью присоединения элемента с типом данных, отличным от CJ_VOID, к входу/выходу сущности, так и вручную, устанавливая этот параметр в окне свойств. В соответствии с выбранным типом данных параметр будет иметь различные диапазоны и занимать различное количество памяти E2.
<b>Value</b>	Значение, принимаемое состоянием. Всегда проверяйте, совместимо ли оно с выбранным типом данных, т.е. находится ли оно в диапазоне, установленном критериями Min и Max (для более детальной информации, пожалуйста, обратитесь к опции <b>Принудительная загрузка параметров</b> ).
<b>Width</b>	Размер значка, представляющего сущность, по горизонтали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.

Чтобы изменить состояние, сделайте одну из следующих операций:

- Присоедините другую сущность (например, алгоритм) к контакту входа;
- Присоедините параметр к EIML-страницам и сделайте его доступным для изменения;
- Используйте один из серийных протоколов связи, чтобы активировать запрос на запись параметра.

### Var (Переменная)

Переменные представляют распределение памяти RAM, которая может быть использована для обработки данных; так как они хранятся в энергозависимой памяти, то будут принимать свои стандартные значения каждый раз, когда Вы перегружаете устройство.

Для каждого проекта максимально доступное число объектов VAR составляет 2047.

Переменные имеют следующие свойства:

<b>Top</b>	Устанавливает положение, куда будет перемещен элемент, относительно верхнего края страницы.
<b>Array</b>	Размер массива сущностей. Если значение больше 1 объекта, это массив, в противном случае это обычный объект.
<b>Category</b>	Является свойством только для чтения; устанавливается для Var.
<b>Convisible</b>	Если отмечено, то параметр может использоваться с обусловленной видимостью.
<b>Show Out LIB</b>	Если отмечено, то параметр «общедоступный», когда он вставлен в библиотеку.
<b>Description</b>	Поле для текстового описания, куда Вы можете вносить собственные записи.
<b>Height</b>	Размер значка, представляющего сущность, по вертикали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.
<b>Left</b>	Устанавливает положение, куда будет перемещен элемент, относительно левого края страницы.
<b>MasterRefresh</b>	Время обновления сущности на главном сетевом трафике.
<b>Max</b>	Максимальное значение, которое может принимать переменная. Зависит от типа.

<b>Min</b>	Минимальное значение, которое может принимать переменная. Зависит от типа.
<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>Order</b>	Порядок выполнения, применяемый для расчета этого элемента (см. параграф “Установка порядка”).
<b>Precision</b>	Определяет число десятичных знаков, используемое для расшифровки внутреннего значения элемента и для отображения значения в EIML-страницах.
<b>RICS</b>	Поле, куда вводится необходимая информация для создания драйвера VTP, требуемого для соединения с EVCO системой мониторинга R.I.C.S. и EVCO параметрами управляющего программного обеспечения <i>Params Manager</i> .
<b>Mb Access</b>	Указывает, активна ли защита чтения/доступа через Modbus.
<b>Mb Level</b>	Указывает уровень защиты от записи через Modbus (если level=0, защиты нет).
<b>WebServer</b>	Поле для вставки полезной информации для представления объекта на сайте контроллера.
<b>Timed</b>	Позволяет определить, в рамках какой задачи будет вычисляться данный элемент и другие элементы, связанные с его входными данными. (Для более подробной информации смотрите опцию <b>Определение задач выполнения</b> ).
<b>Type</b>	Задаст тип данных переменной. Принимает значение по умолчанию CJ_VOID. Его можно изменить как автоматически, с помощью присоединения элемента с типом данных, отличным от CJ_VOID, к входу/выходу сущности, так и вручную, устанавливая этот параметр в окне свойств. В соответствии с выбранным типом данных, параметр будет иметь различные диапазоны и занимать различное количество памяти RAM.
<b>Value</b>	Значение по умолчанию при включении электропитания контроллера. Всегда проверяйте, совместимо ли оно с выбранным типом данных, т. е. находится ли оно в диапазоне, установленном пределами Min и Max.
<b>Width</b>	Размер значка, представляющего сущность, по горизонтали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.

## Timer (Таймер)

Таймеры являются счетчиками, которые автоматически увеличиваются/уменьшаются системой по оси времени. Они имеют входы, которые Вы можете использовать для сброса и загрузки значения или активации/остановки отсчета времени, а также один выход, который выдает значение счетчика. Единственный вход, который всегда активен, - вход сброса, в то время как входы активации, очистки и перезагрузки могут быть активированы из контекстного меню:



Если вход **Enable** принимает значение 0, отсчет времени будет остановлен, если значение равно 1, отсчет времени будет активирован. Вход **Reset** обнаруживает переход от 0 к 1, чтобы начать отсчет времени. Вход **Clear** воспринимает переход от 0 к 1 как сигнал к удалению текущего отсчета. Соединение параметра с входом **reload** означает загрузку счетчика таймера со значением параметра вместо значения по умолчанию (см. свойство Max).

Вы можете легко установить свойства таймера, чтобы определить, как часто счетчик должен переключаться и в каком направлении будет вестись отсчет: в сторону увеличения или уменьшения.

Используя эти объекты, Вы можете обеспечить защиту короткого цикла компрессора или регулируемое по времени освещение.

Для каждого проекта максимально доступное число объектов TIMER составляет 255.

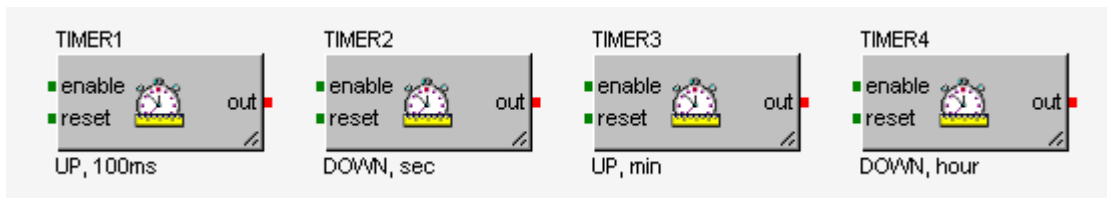
Таймеры имеют следующие свойства:

<b>Top</b>	Устанавливает положение, куда будет перемещен элемент, относительно верхнего края страницы.
<b>Category</b>	Является свойством только для чтения; устанавливается для Timer.
<b>Description</b>	Поле для текстового описания, куда Вы можете вносить собственные записи.
<b>Direction</b>	Направление, в котором изменяется значение во время активации входа <b>Enable</b> . Если этот критерий установлен как UP, значение счетчика будет увеличиваться по заранее установленной оси времени, начиная с нуля (когда происходит сброс времени) и до значения, установленного критерием Max. И наоборот, если критерий установлен как DOWN, отсчет будет проходить в обратном направлении, начиная от значения, установленного Max и до нуля.
<b>Height</b>	Размер значка, представляющего сущность, по вертикали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.
<b>Left</b>	Устанавливает положение, куда будет перемещен элемент, относительно левого края страницы.
<b>Max</b>	Максимальное значение, которое может принимать таймер. Данный критерий зависит от значения, установленного в свойстве Direction: если направление задано как UP, то данный критерий устанавливает значение, которое необходимо достигнуть таймеру; иначе устанавливает начальное значение после сброса. Если активен вход перезагрузки, то именно это значение будет загружено в качестве начального либо конечного предела, в то время как Max будет представлять основной контрольный предел.
<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>Order</b>	Порядок выполнения, применяемый для расчета этого элемента (см. параграф “Установка порядка выполнения программы”).
<b>RICS</b>	Поле, куда вводится необходимая информация для создания драйвера VTP, требуемого для соединения с EVCO системой мониторинга R.I.C.S. и EVCO параметрами управляющего программного обеспечения <i>Params Manager</i> .
<b>Mb Access</b>	Указывает, активна ли защита чтения/доступа через Modbus.
<b>Mb Level</b>	Указывает уровень защиты от записи через Modbus (если level=0, защиты нет).
<b>WebServer</b>	Поле для вставки полезной информации для представления объекта на сайте контроллера.
<b>Timed</b>	Позволяет определить, в рамках какой задачи будет вычисляться данный элемент и другие элементы, связанные с его входными данными. (Для более подробной информации смотрите опцию <b>Определение задач выполнения</b> ).
<b>Type</b>	Установленный тип данных CJ_WORD, который не может быть изменен. В связи с этим таймер может принимать максимальное значение, равное предельному для этого типа данных (см. параграф <b>Типы данных</b> ).
<b>Udm</b>	Единица времени, которая вызывает изменение значения таймера. Данный критерий может принимать следующие значения: 100 мс с мин час В зависимости от установленного здесь значения, таймер будет увеличивать или уменьшать свое значение каждые 100 миллисекунд, одну секунду, одну минуту или один час соответственно.
<b>Width</b>	Размер значка, представляющего сущность, по горизонтали, в пикселях.

Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.

### Отображение свойств Udm и Direction под сущностью таймера

Под сущностью TIMER будет добавлена строка со значениями свойств *Direction* и *Udm*.



### Command In (Входная команда)

Среда UNI-PRO позволяет устанавливать групповые, определяемые пользователем команды, которые активируют определенные действия. Объекты Command In указывают на получение этих команд и активируют любое действие, которое может быть с ними связано. Все объекты этого типа имеют выход, который выдает значение параметра полученной команды и узел, с которого эта команда была отправлена.

Для каждого проекта максимально доступное число объектов COMMAND IN составляет 255.

Объекты Command In имеют следующие свойства:

<b>Top</b>	Устанавливает положение, куда будет перемещен элемент, относительно верхнего края страницы.
<b>Category</b>	Является свойством только для чтения; устанавливается для Command In.
<b>Cmd</b>	Указывает имя, определяющее команду, получение которой Вы хотите записать (см. параграф <b>Команды</b> ).
<b>Description</b>	Поле для текстового описания, куда Вы можете вносить собственные записи.
<b>Height</b>	Размер значка, представляющего сущность, по вертикали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.
<b>Left</b>	Устанавливает положение, куда будет перемещен элемент, относительно левого края страницы.
<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>Type</b>	Тип данных установлен как CJ_CMD и изменен быть не может.
<b>Width</b>	Размер значка, представляющего сущность, по горизонтали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.

### Command Out (Выходная команда)

Выходы команд используются для отправления команд, которые будут интерпретированы входом команды в том же самом или другом контроллере. Этот тип объекта имеет два входа: **param** и **trigger**. Первый вход используется для считывания параметра при отправлении команды; второй является входом триггера, т.е., он воспринимает изменение сигнала, соединенного с сущностью, которая активирует передачу команды, когда значение увеличивается от 0 до 1. Для каждого проекта максимально доступное число объектов COMMAND OUT составляет 255.

Объекты имеют следующие свойства:

<b>Top</b>	Устанавливает положение, куда будет перемещен элемент, относительно верхнего края страницы.
<b>Category</b>	Является свойством только для чтения; устанавливается для Command Out.
<b>Cmd</b>	Указывает имя, определяющее команду, получение которой Вы хотите записать (см. параграф <b>Команды</b> ).
<b>Description</b>	Поле для текстового описания, куда Вы можете вносить собственные записи.
<b>Height</b>	Размер значка, представляющего сущность, по вертикали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.
<b>Left</b>	Устанавливает положение, куда будет перемещен элемент, относительно левого края страницы.
<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>Order</b>	Порядок выполнения, применяемый для вычисления этого элемента (см. раздел «Установка порядка выполнения программы»).
<b>Timed</b>	Позволяет определить, в рамках какой задачи будет вычисляться данный элемент и другие элементы, связанные с его входными данными. (Для более подробной информации смотрите опцию <b>Определение задач выполнения</b> ).
<b>Width</b>	Размер значка, представляющего сущность, по горизонтали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.

### Note (Примечание)

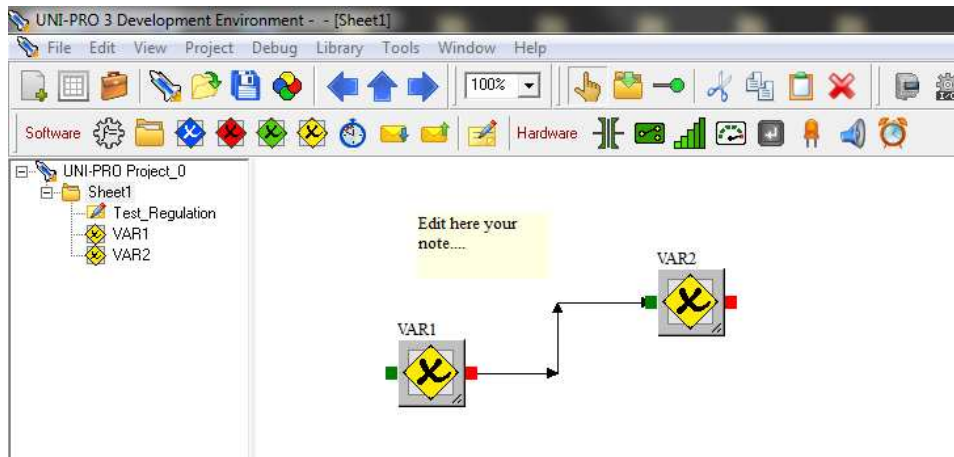
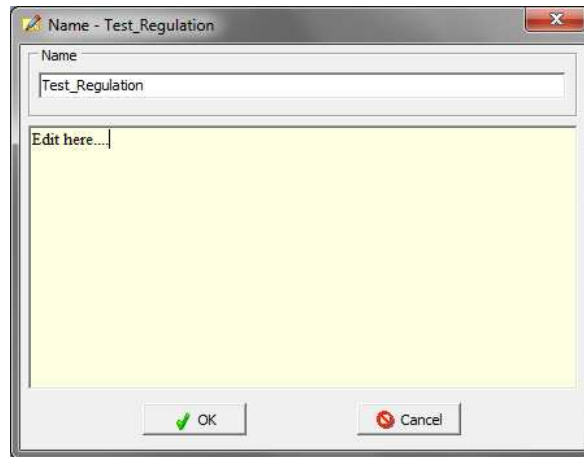
Возможно добавление в схемы проекта текстового примечания для добавления комментариев, полезных при составлении заявки.

Сущность Примечание имеет следующие свойства:

<b>Top</b>	Устанавливает положение, в котором элемент будет нарисован относительно верхнего края листа.
<b>Category</b>	Это свойство доступно только для чтения и имеет значение Note.
<b>Description</b>	Поле текстового типа, где вы можете ввести заметки.
<b>Height</b>	Размер по вертикали в пикселях значка, представляющего объект. Чтобы изменить его, вы можете либо использовать таблицу свойств, либо графически изменить размер объекта с помощью указателя мыши.
<b>Left</b>	Устанавливает положение, в котором элемент будет отрисовываться относительно левого края листа.
<b>Name</b>	Уникальное имя, идентифицирующее элемент внутри проекта.
<b>Width</b>	Горизонтальный размер в пикселях значка, представляющего объект. Чтобы изменить его, вы можете либо использовать таблицу свойств, либо графически изменить размер объекта с помощью указателя мыши.

Двойной щелчок по объекту открывает редактор для ввода текста примечания.



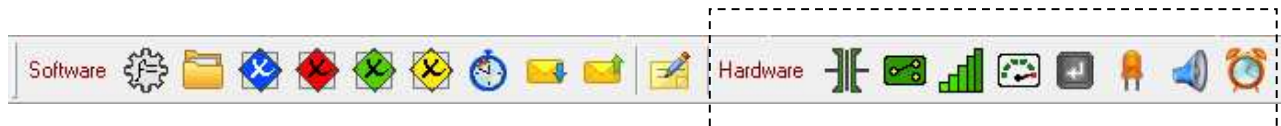


Объекты Notes являются только визуальными и не влияют на компиляцию проекта.

Цвет фона и шрифт Заметок можно изменить в окне настроек среды UNI-PRO.

### 3.4.2 Оборудование

Секция оборудования включает все сущности, которые представляют физический Вход/Выход. Количество и тип сущностей оборудования, необходимых в проекте, зависит от типа контроллера и количества присоединенных расширений. Следовательно, Вы можете действовать двумя способами: добавить все необходимые сущности в проект и затем выбрать все подходящее оборудование, или сначала выбрать оборудование, на основе которого Вы решили разработать проект, а затем добавить необходимые сущности.



Эта категория включает:

- Цифровые входы;
- Цифровые выходы;
- Аналоговые входы;
- Аналоговые выходы;
- Кнопки;
- Индикаторы;
- Звуковые сигналы;
- Часы.

Каждая сущность в этой секции должна быть присоединена к физическому ресурсу (Вход/Выход, чрв, Индикатор и т.д.) с помощью **Join Tools**.

Для каждой из этих категорий максимально доступное число сущностей составляет 255.

#### Digital Input (Цифровой вход)

Объекты Digital Input представляют цифровые входы контроллера или расширений, если таковые присутствуют, и имеют следующие свойства:

<b>Top</b>	Устанавливает положение, куда будет перемещен элемент, относительно верхнего края страницы.
<b>Category</b>	Устанавливается для DigitalIn и не может быть изменена.
<b>Description</b>	Поле для текстового описания, куда Вы можете вносить собственные записи.
<b>Height</b>	Размер значка, представляющего сущность, по вертикали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.
<b>Left</b>	Устанавливает положение, куда будет перемещен элемент, относительно левого края страницы.
<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>RICS</b>	Поле, куда вводится необходимая информация для создания драйвера VTP, требуемого для соединения с EVCO системой мониторинга R.I.C.S. и EVCO параметрами управляющего программного обеспечения <i>Params Manager</i> .
<b>Mb Access</b>	Указывает, активна ли защита чтения/доступа через Modbus.
<b>Mb Level</b>	Указывает уровень защиты от записи через Modbus (если level=0, защиты нет).
<b>WebServer</b>	Поле для вставки полезной информации для представления объекта на сайте контроллера.
<b>Type</b>	Устанавливает тип данных, который для цифровых входов определен как CJ_BIT и не может быть

<b>Width</b>	Горизонтальный размер в пикселях значка, представляющего объект. Чтобы изменить его, вы можете либо использовать таблицу свойств, либо графически изменить размер объекта с помощью указателя мыши.
--------------	---

Чтобы добавить Digital In в проект, выберите панель Hardware и нажмите значок Digital In; затем щелчком мыши укажите место на листе, куда вы хотите добавить сущность. Чтобы просмотреть ее свойства, щелкните два раза по Digital In (или нажмите правую кнопку мыши и выберите **Properties** в меню).

Digital Inputs позволяют отследить значение, принимаемое физическим цифровым входом, и использовать его в качестве входа для алгоритма, чтобы выполнять более сложные операции.

Чтобы узнать, как присоединить Digital In к соответствующему физическому цифровому входу, пожалуйста, обратитесь к главе **Join Wizard**.

### Digital Output (Цифровой выход)

Объекты Digital Out представляют цифровые выходы контроллера или расширений, если присутствуют, и имеют следующие свойства:

<b>Top</b>	Устанавливает положение, куда будет перемещен элемент, относительно верхнего края страницы.
<b>Category</b>	Устанавливается для DigitalOut и не может быть изменена.
<b>Description</b>	Поле для текстового описания, куда Вы можете вносить собственные записи.
<b>Height</b>	Размер значка, представляющего сущность, по вертикали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.
<b>Left</b>	Устанавливает положение, куда будет перемещен элемент, относительно левого края страницы.
<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>RICS</b>	Поле, куда вводится необходимая информация для создания драйвера VTP, требуемого для соединения с EVCO системой мониторинга R.I.C.S. и EVCO параметрами управляющего программного обеспечения <i>Params Manager</i> .
<b>Mb Access</b>	Указывает, активна ли защита чтения/доступа через Modbus.
<b>Mb Level</b>	Указывает уровень защиты от записи через Modbus (если level=0, защиты нет).
<b>WebServer</b>	Поле для вставки полезной информации для представления объекта на сайте контроллера.
<b>Order</b>	Порядок выполнения, применяемый для расчета этого элемента (см. параграф “Установка порядка выполнения программы”).
<b>Timed</b>	Позволяет определить, в рамках какой задачи будет вычисляться данный элемент и другие элементы, связанные с его входными данными. (Для более подробной информации смотрите опцию <b>Определение задач выполнения</b> ).
<b>Type</b>	Устанавливает тип данных, который для цифровых выходов определен как CJ_BIT и не может быть изменен.
<b>Width</b>	Размер значка, представляющего сущность, по горизонтали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.

Чтобы добавить Digital Out на рабочий лист, выполните те же действия, что и при добавлении Digital Input.

**Analog Input (Аналоговый вход)**

Объекты Analog Input являются абстрактным представлением любого типа аналогового терминала. Фактически они могут представлять физический вход следующих типов: NTC, PTC, 0-5 В, 0-10 В, 0-20 мА, 4-20 мА, RT1000, ТЕМПЕРАТУРА, ВЛАЖНОСТЬ. Это означает, что этот тип элемента может быть использован для считывания показаний датчика или основной температуры, давления или положения чувствительного элемента.

Analog Input характеризуется контактом выхода CJ\_ANALOG и следующими свойствами:

<b>Top</b>	Устанавливает положение, куда будет перемещен элемент, относительно верхнего края страницы.
<b>Category</b>	Устанавливается для AnalogIn и не может быть изменена.
<b>Description</b>	Поле для текстового описания, куда Вы можете вносить собственные записи.
<b>Height</b>	Размер значка, представляющего сущность, по вертикали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.
<b>Left</b>	Устанавливает положение, куда будет перемещен элемент, относительно левого края страницы.
<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>Precision</b>	Определяет число десятичных знаков, используемое для расшифровки внутреннего значения элемента и для отображения значения в EIML-страницах.
<b>RICS</b>	Поле, куда вводится необходимая информация для создания драйвера VTP, требуемого для соединения с EVCO системой мониторинга R.I.C.S. и EVCO параметрами управляющего программного обеспечения <i>Params Manager</i> .
<b>Mb Access</b>	Указывает, активна ли защита чтения/доступа через Modbus.
<b>Mb Level</b>	Указывает уровень защиты от записи через Modbus (если level=0, защиты нет).
<b>WebServer</b>	Поле для вставки полезной информации для представления объекта на сайте контроллера.
<b>Sensor</b>	Для каждого аналогового входа, устанавливаемого в проекте, необходимо определить тип чувствительного элемента, значение которого вы хотите считывать. Возможны следующие варианты: 0-5 В 0-10 В 0-20 мА 4-20 мА NTC PTC RT1000 TEMPERATURE (ТЕМПЕРАТУРА) HUMIDITY (ВЛАЖНОСТЬ ) NTC 10K-2 NTC 10K-3 RESISTANCE (СОПРОТИВЛЕНИЕ)  Выбранный тип чувствительного элемента должен поддерживаться оборудованием.
<b>Type</b>	Устанавливает тип данных, который для аналоговых входов определен как CJ_ANALOG и не может быть изменен.
<b>Width</b>	Размер значка, представляющего сущность, по горизонтали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.

Чтобы считывать состояние аналогового входа, добавленного в проект, соедините контакт его выхода с входом Algo или Var. Как отмечалось ранее, CJ\_ANALOG является структурным типом данных, состоящим из нескольких компонентов информации: поле Error, которое указывает на возможную ошибку, и поле Value, которое показывает значение.

**Примечание:** датчики типа 0-10 В и 0-5 В не определяют наличие неисправного либо неподключенного датчика относительно действительно измеренного значения 0.

### Analog Output (Аналоговый выход)

Аналоговые выходы являются абстрактным представлением аналогового выхода тока (0-20 мА) или аналогового выхода напряжения (0-10 В). Эти типы объектов имеют контакт входа CJ\_WORD, который выражает принимаемое значение в процентах, от 0 до 100, и имеют следующие свойства:

<b>Top</b>	Устанавливает положение, куда будет перемещен элемент, относительно верхнего края страницы.
<b>Actuator</b>	Для каждого аналогового выхода, используемого в проекте, необходимо указать тип привода, соединенного с выходом, из списка ниже:  FAN (ВЕНТИЛЯТОР) 0-10 В 0-20 мА 4-20 мА PWM BELIMO EEV_U
<b>Category</b>	Тип датчика должен поддерживаться аппаратным обеспечением. Устанавливается для AnalogOut и не может быть изменена.
<b>Description</b>	Поле для текстового описания, куда Вы можете вносить собственные записи.
<b>Frequency</b>	Значение по умолчанию (Гц) частоты цикла для привода PWM. [Мин. 10 Гц – макс. 2000 Гц]
<b>Height</b>	Размер значка, представляющего сущность, по вертикали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.
<b>Left</b>	Устанавливает положение, куда будет перемещен элемент, относительно левого края страницы.
<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>Order</b>	Порядок выполнения, применяемый для вычисления рассматриваемого элемента (см. раздел «Установка порядка выполнения программы»).
<b>Precision</b>	Определяет число десятичных знаков, используемое для расшифровки внутреннего значения элемента и для отображения значения в EIML-страницах.
<b>RICS</b>	Поле, куда вводится необходимая информация для создания драйвера VTP, требуемого для соединения с EVCO системой мониторинга R.I.C.S. и EVCO параметрами управляющего программного обеспечения <i>Params Manager</i> .
<b>Mb Access</b>	Указывает, активна ли защита чтения/доступа через Modbus.
<b>Mb Level</b>	Указывает уровень защиты от записи через Modbus (если level=0, защиты нет).

<b>WebServer</b>	Поле для вставки полезной информации для представления объекта на сайте контроллера.
<b>Timed</b>	Позволяет определить, в рамках какой задачи будет вычисляться данный элемент и другие элементы, связанные с его входными данными. (Для более подробной информации смотрите опцию <b>Определение задач выполнения</b> ).
<b>Type</b>	Устанавливает тип данных, который для аналоговых выходов определен как CJ_WORD и не может быть изменен, т.к. чтобы установить аналоговый выход, необходимо присвоить ему значение.
<b>Width</b>	Размер значка, представляющего сущность, по горизонтали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.

### Button (Кнопка)

Элементы Button представляют действия, производимые с кнопками, действия могут быть следующими:

- Press (Нажатие)
- Hold down (Удерживание)
- Release (Отпускание)

Именно поэтому к одной кнопке могут относиться три различных объекта Button, каждый из которых производит соответствующие действия.

Чтобы выделить действие, которое Вы хотите проанализировать, установите свойство Cmd.

<b>Top</b>	Устанавливает положение, куда будет перемещен элемент, относительно верхнего края страницы.
<b>Category</b>	Устанавливается для Button и не может быть изменена.
<b>Cmd</b>	Свойство Cmd идентифицирует тип действия, которое Вы хотите прервать. Он может принимать следующие значения: ON_PRESSED ON_CONTINUE ON_RELEASE
<b>Description</b>	Поле для текстового описания, куда Вы можете вносить собственные записи.
<b>Height</b>	Размер значка, представляющего сущность, по вертикали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.
<b>Left</b>	Устанавливает положение, куда будет перемещен элемент, относительно левого края страницы.
<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>Type</b>	Устанавливает тип данных, который для кнопок определен как CJ_BUZZER и не может быть изменен.
<b>Width</b>	Размер значка, представляющего сущность, по горизонтали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.

### Led (Индикатор)

Объекты LED являются логическим представлением индикаторов, расположенных в программируемом Вами контроллере, или в присоединенных к нему расширениях. Этот тип объекта имеет вход контакта CJ\_LED, и диапазон его значений составляет 0-3.

Для удобства установки его значения Вы можете изменять режим работы выбранного индикатора следующим образом:

- 0 : off (выключен)
- 1 : on (включен)
- 2 : slow frequency flash (медленно мигает)
- 3 : fast frequency flash (быстро мигает)

(См. параграф **Типы данных**).

Индикаторы могут быть использованы для передачи пользователю информации, которая всегда должна быть доступна, например, состояние компрессора или состояние открытия двери. Эти объекты имеют следующие свойства:

<b>Top</b>	Устанавливает положение, куда будет перемещен элемент, относительно верхнего края страницы.
<b>Category</b>	Устанавливается для LED и не может быть изменена.
<b>Description</b>	Поле для текстового описания, куда Вы можете вносить собственные записи.
<b>Height</b>	Размер значка, представляющего сущность, по вертикали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.
<b>Left</b>	Устанавливает положение, куда будет перемещен элемент, относительно левого края страницы.
<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>Order</b>	Порядок выполнения, применяемый для расчета этого элемента (см. параграф “Установка порядка выполнения программы”).
<b>RICS</b>	Поле, куда вводится необходимая информация для создания драйвера VTP, требуемого для соединения с EVCO системой мониторинга R.I.C.S. и EVCO параметрами управляющего программного обеспечения <i>Params Manager</i> .
<b>Mb Access</b>	Указывает, активна ли защита чтения/доступа через Modbus.
<b>Mb Level</b>	Указывает уровень защиты от записи через Modbus (если level=0, защиты нет).
<b>WebServer</b>	Поле для вставки полезной информации для представления объекта на сайте контроллера.
<b>Timed</b>	Позволяет определить, в рамках какой задачи будет вычисляться данный элемент и другие элементы, связанные с его входными данными. (Для более подробной информации смотрите опцию <b>Определение задач выполнения</b> ).
<b>Type</b>	Устанавливает тип данных, который для индикаторов определен как CJ_LED и не может быть изменен.
<b>Width</b>	Размер значка, представляющего сущность, по горизонтали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.

### **Buzzer (Звуковой сигнал)**

Объекты Buzzer являются крайне полезными в тех ситуациях, где необходимо звучание сигнала тревоги. Например, удобно использовать звуковой сигнал как предупреждение о потенциально опасном состоянии или в критической ситуации.

Чтобы подключить звуковой сигнал, просто добавьте объект Buzzer в проект и соедините его с алгоритмом, имеющим выход CJ\_BUZZ.

В зависимости от значения входа звукового сигнала, его состояние будет установлено в следующих режимах:

0 : off (выключен)

1 : on (включен)

2 : slow frequency beep (медленно издает звуковые сигналы)

3 : fast frequency beep (быстро издает звуковые сигналы)

(См. параграф **Типы данных**).

<b>Top</b>	Устанавливает положение, куда будет перемещен элемент, относительно верхнего края страницы.
<b>Category</b>	Устанавливается для Buzzer и не может быть изменена.
<b>Description</b>	Поле для текстового описания, куда Вы можете вносить собственные записи.
<b>Height</b>	Размер значка, представляющего сущность, по вертикали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.
<b>Left</b>	Устанавливает положение, куда будет перемещен элемент, относительно левого края страницы.
<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>RICS</b>	Поле для ввода полезной информации в генерацию драйвера VTP, необходимой для связи с программным обеспечением контроля EVCO и программным обеспечением для управления параметрами EVCO Params Manager.
<b>Mb Access</b>	Указывает, активна ли защита чтения/доступа через Modbus.
<b>Mb Level</b>	Указывает уровень защиты от записи через Modbus (если level=0, защиты нет).
<b>WebServer</b>	Поле для вставки полезной информации для представления объекта на сайте контроллера.
<b>Order</b>	Порядок выполнения, применяемый для расчета этого элемента (см. параграф “Установка порядка выполнения программы”).
<b>Timed</b>	Позволяет определить, в рамках какой задачи будет вычисляться данный элемент и другие элементы, связанные с его входными данными. (Для более подробной информации смотрите опцию <b>Определение задач выполнения</b> ).
<b>Type</b>	Указывает тип данных, который для светодиодов явно установлен на CJ_LED и не может быть изменен.
<b>Width</b>	Размер значка, представляющего сущность, по горизонтали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.

## Clock (Часы)

Объекты Clock являются логическим представлением часов реального времени, встроенных в контроллеры и расширения.

С помощью этих объектов Вы можете установить текущую дату и время, таким образом, разрешая все операции, которые каким-либо образом связаны с часами. Например, Вы можете запрограммировать режим работы другого контроллера в зависимости от диапазона времени, в котором он работает.

Объекты Clock выдают значение CJ\_DATETIME, которое Вы легко можете преобразовать в структуру CJ\_DATETIME\_STRUCT, чтобы выполнять более сложные операции (пожалуйста, обратитесь к параграфу Типы данных – CJ\_DATETIME\_STRUCT).

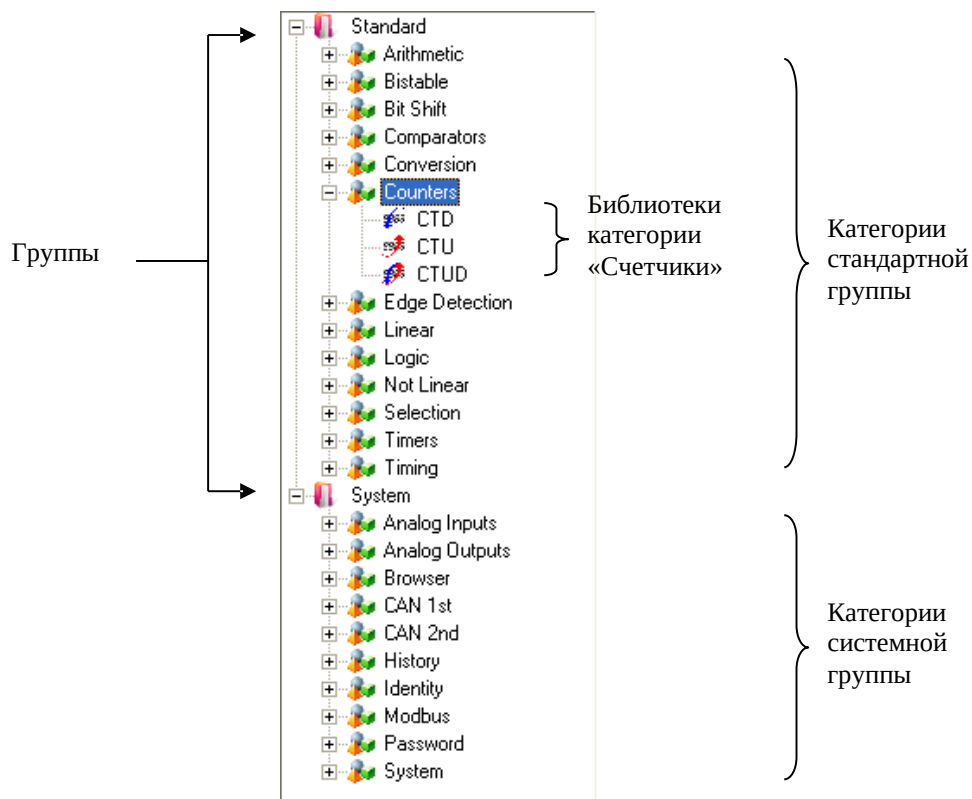
<b>Top</b>	Устанавливает положение, куда будет перемещен элемент, относительно верхнего края страницы.
------------	---



<b>Category</b>	Устанавливается для Clock и не может быть изменена.
<b>Description</b>	Поле для текстового описания, куда Вы можете вносить собственные записи.
<b>Height</b>	Размер значка, представляющего сущность, по вертикали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.
<b>Left</b>	Устанавливает положение, куда будет перемещен элемент, относительно левого края страницы.
<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>RICS</b>	Поле, куда вводится необходимая информация для создания драйвера VTP, требуемого для соединения с EVCO системой мониторинга R.I.C.S. и EVCO параметрами управляющего программного обеспечения <i>Params Manager</i> .
<b>Mb Access</b>	Указывает, активна ли защита чтения/доступа через Modbus.
<b>Mb Level</b>	Указывает уровень защиты от записи через Modbus (если level=0, защиты нет).
<b>WebServer</b>	Поле для вставки полезной информации для представления объекта на сайте контроллера.
<b>Type</b>	Устанавливает тип данных, который для часов определен как CJ_DATETIME и не может быть изменен.
<b>Width</b>	Размер значка, представляющего сущность, по горизонтали, в пикселях. Чтобы его изменить, используйте свойства таблицы, или графически измените размер сущности с помощью курсора мыши.

### 3.4.3 Библиотеки

Древовидное меню библиотек представляет собой собрание пользовательских библиотек и библиотек среды в виде двухуровневой структуры. Первый уровень – это библиотека группы, а второй – категория; в категориях перечислены библиотеки.



Группы *Standard* (стандартная) и *System* (системная) – 2 группы по умолчанию библиотек UNI-PRO; Вы можете добавлять другие группы и расширять их новыми категориями, создавая новые библиотеки.

### 3.4.4 Используемая документация по библиотекам

С помощью пункта меню *Project->Info Libraries Used* можно создать документ формата HTML, содержащий информацию обо всех используемых библиотеках/шаблонах (тип библиотек, имя сущности, версия и т.д.).


### 3.5 Элементы в EIML-страницах

EIML-страницы используются для создания графического интерфейса контроллера. Они позволяют отображать тексты, значки, переменные, параметры, внутренние состояния, а также изменять значения, подключать команды и т.д.

С помощью EIML (встроенный язык написания пользовательского интерфейса) Вы можете создавать окна пользовательского интерфейса, соответствующие особым требованиям, и физически хранить их в памяти контроллера, чтобы они могли загружаться на экране в процессе работы.

Легко соединяя объекты и устанавливая их свойства, Вы можете создать удобную для пользователя навигацию по страницам.

Существует три различных типа страниц, в зависимости от дисплея, для которого они разрабатываются: графический (240x128 или 128x64 пикселей), буквенно-цифровой (20x4 или 16x4 символов) и 7-сегментный.

Если выбрать в меню *File/New/New EIML Page* или нажать значок  на панели инструментов, то будет показан список страниц, которые могут быть созданы. Как только Вы выберете одну из них, в рабочей области появится окно, в котором Вы сможете сформировать страницу и установить необходимые свойства дисплея.

Размер каждой страницы в байтах показан в нижнем правом углу. Эта информация необходима для определения способности терминала отображать данную страницу.

Для каждого проекта максимально допустимое число EIML-страниц составляет 767.

#### 3.5.1 Панель инструментов

EIML панель инструментов содержит элементы, которые Вы можете добавить на страницу: Тексты, Переменные, Значки, Combo, Таблицы, Линии, Прямоугольники.



Первым элементом в панели инструментов идет значок со стрелкой, который позволяет выделять, изменять размер и воздействовать на элементы, находящиеся в странице.

Чтобы добавить новый элемент (текст, переменную, значок и т.д.), выберите необходимый элемент в панели инструментов и перетащите его в ту область, где Вы хотите его расположить.

#### 3.5.2 Внесение элемента на EIML-страницу

Чтобы внести элемент на страницу, выделите его на панели инструментов и передвигайте мышью, удерживая нажатой левую кнопку мыши, чтобы создать прямоугольник нужного размера. Как только прямоугольник был создан, его можно передвинуть в нужное положение.

#### 3.5.3 Свойства страницы

EIML станицы являются вместилищами для других EIML-объектов и имеют следующие свойства:

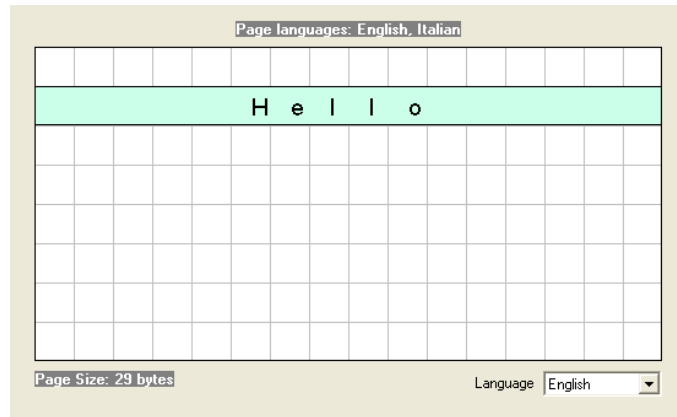
<b>Name</b>	Уникальное имя, которое идентифицирует страницу внутри проекта.
<b>Id</b>	Параметр Id является цифровым индексом в диапазоне от 1 до 240, который позволяет вам идентифицировать страницу внутри проекта.
<b>PreviousPage</b>	Если не установлено <NONE>, указывает на предыдущую страницу, которая будет загружена при нажатии кнопки LEFT.
<b>NextPage</b>	Если не установлено <NONE>, указывает на следующую страницу, которая будет загружена при нажатии кнопки RIGHT.
<b>TimeoutPage</b>	Если не установлено <NONE>, указывает на страницу, которая будет загружена, когда время ожидания, установленное в свойствах Timeout, истечет.

	Эта страница загружается также и при нажатии кнопки ESC. Если значение установлено как <NONE>, будет загружена, по умолчанию, страница пользовательского интерфейса.
<b>Timeout</b>	Если отлично от нуля, то определяет количество секунд, которое должны пройти с момента последнего нажатия кнопки перед автоматическим возвратом к предыдущей странице.
<b>Language</b>	Указывает язык, используемый для записи страницы. Контроллер обладает системной переменной, которая обозначает текущий язык: каждый раз, когда запрашивается страница с определенным Id, в первую очередь проводится поиск страницы на текущем языке. Если страница многоязычна, данный параметр позволяет отображать страницу на выбранном языке.
<b>Level</b>	Указывает уровень защиты страницы: дисплей, который запрашивает страницу, должен иметь уровень доступа, равный или превосходящий уровень страницы. В противном случае появится запрос на введение пароля для более высокого уровня доступа.
<b>Encoding</b>	Указывает на кодировку страницы. Если на странице используются переменные, то при выборе данного свойства flash-размер страницы будет уменьшен, что восстанавливает объем памяти. Закодированный размер страницы перекрывает собой исходный размер "Page Size".
<b>Description</b>	Поле для текстового описания, куда Вы можете вносить собственные записи.
<b>Color (only color pages)</b>	Цвет, используемый для фона страницы.
<b>Background (only color pages)</b>	Название растрового фона страницы. Если <NONE>, фон страницы будет иметь цвет, установленный в свойстве «Цвет».
<b>Show background (only color pages)</b>	Если активен, то показывает фон страницы, время оформления (возможно задерживает показ).
<b>Navigation bar (only color pages)</b>	Устанавливает отображение панели навигации внизу страницы со следующими значениями: <ul style="list-style-type: none"> <li>- NO: Никогда не виден</li> <li>- ALWAYS: Всегда виден</li> <li>- ON TOUCH: Отключающийся</li> </ul>
<b>Circular Focus</b>	Если выбрано, то позволяет круговое перемещение элементов со свойством Focusindex. Поэтому, когда курсор передвинулся к последнему элементу страницы, следующий элемент, активированный курсором, будет первым на той же странице.
<b>Memo Focus Index</b>	Если выбрано, то позволяет сохранять свойство Focusindex во время навигации по страницам.
<b>Visibility</b>	Если выбрано, позволяет использовать Условную видимость на EIML-страницах.
<b>Multilanguage (only for page with multilanguage as 7-Segments)</b>	Если выбрано, активирует использование многоязычности на странице. Это означает возможность размещения текстов на разных языках.

### 3.5.4 Многоязычные страницы

Если для EIML-страницы выбрано свойство *Multilanguage*, то данную страницу можно создать на нескольких языках; благодаря этой особенности для каждого нового языка не надо создавать новую страницу (с тем же *Id*).

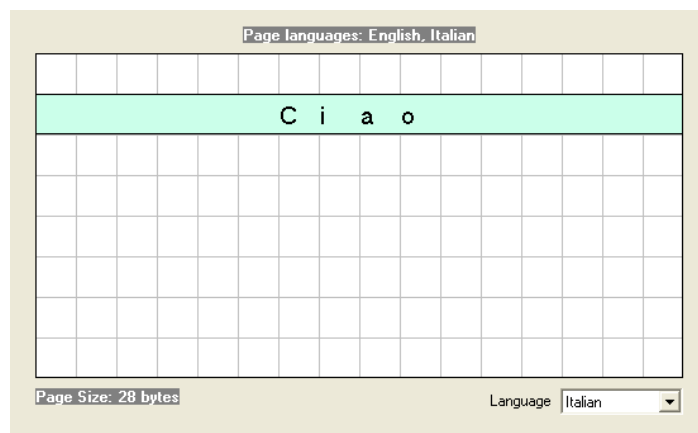
Данное свойство также ускоряет создание и управление проектами с пользовательскими интерфейсами на нескольких языках. Пример многоязычной странице ниже:



Сверху имеется список языков, для которых (как минимум) был введен текст; внизу, в дополнение к информации о занятости страницы, можно задать просмотр страницы на выбранном языке.

Если Вы смените язык, можно проверить смену текста на тот случай, если какой-либо текст не был вставлен (либо занимает место, превышающее допустимый размер).

Например: если для текста “*Hello*” был введен перевод на итальянский “*Ciao*”, то при смене языка просмотра страница будет выглядеть следующим образом:

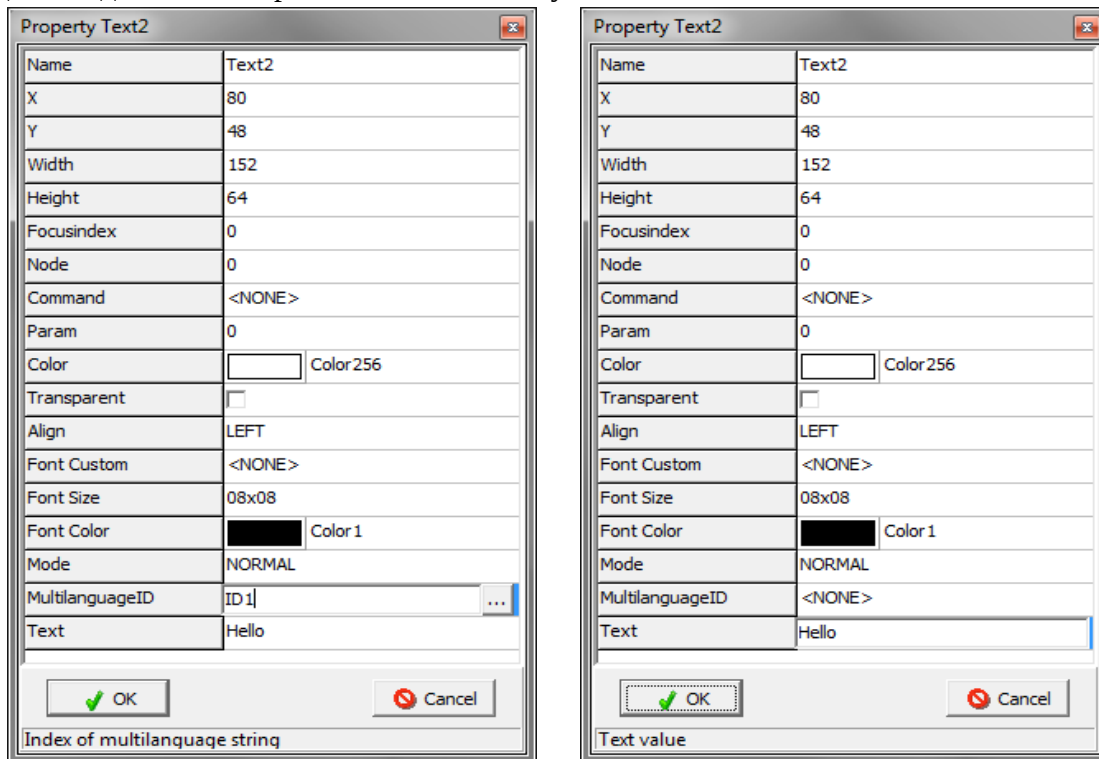


Отсутствующий текст, возможно, является результатом пропущенного перевода либо ошибки.

**Примечание:** для текстов в элементах COMBO и CHARTS просмотр может быть запущен только при моделировании страницы с помощью EIML-симулятора и со сменой языков с использованием параметров симулятора.

### 3.5.5 Многоязычные текстовые элементы (с языковым идентификатором)

Каждый текстовый элемент может быть многоязычным или иметь фиксированный текст для каждого языка. Для настройки многоязычного текста необходимо связать идентификатор из лингвистической базы данных. Если для свойства MultilanguageID задано значение <NONE>, текст для каждого языка принимает значение, указанное в свойстве Text.



На первом изображении показаны свойства многоязычного текста со связанным идентификатором ID1, а на втором — текст с фиксированным текстом «Hello».

Каждый идентификатор может использоваться в нескольких текстовых элементах.

Отображение и предварительный просмотр страниц идентичны предыдущему случаю (многоязычные страницы).

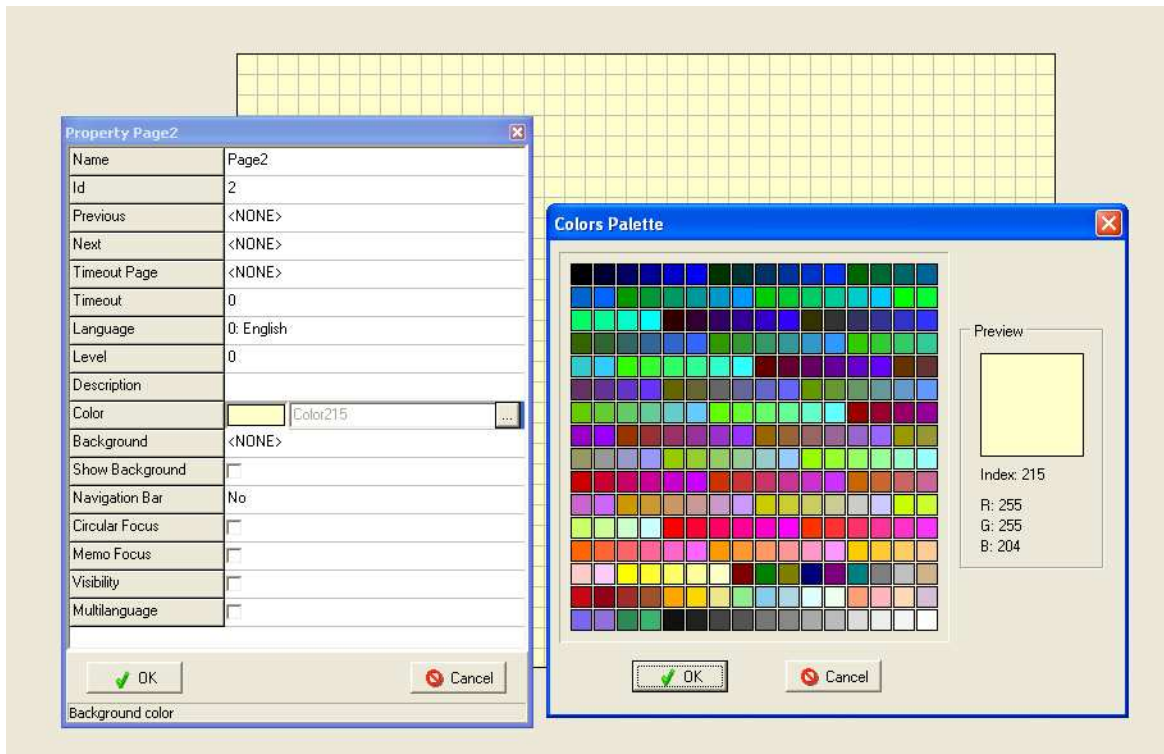
База данных для управления лингвистическими идентификаторами находится в Tools->StringsTable, однако она доступна с каждой страницы, которая использует тексты с идентификатором, в этом случае база данных будет отображать только представление идентификаторов, используемых на этой странице.

Эта функция улучшает старую многоязычную страницу (3.5.4), поскольку позволяет избежать дублирования страниц в памяти, страница всегда уникальна, и только тексты на странице отображаются в соответствии с активным языком.

### 3.5.6 Страницы с 256 цветами (EPJ Color)

Цветные страницы могут управлять элементами с цветовыми свойствами, фоном и растровыми изображениями, состоящими из 256 цветов. Фон страницы может быть одноцветным или растровым, выбираемым из максимум 10 фонов (см. инструмент Фоны EIML).

Для задания цветовых свойств элементов EIML необходимо использовать рамку, имеющую все 256 цветов палитры.

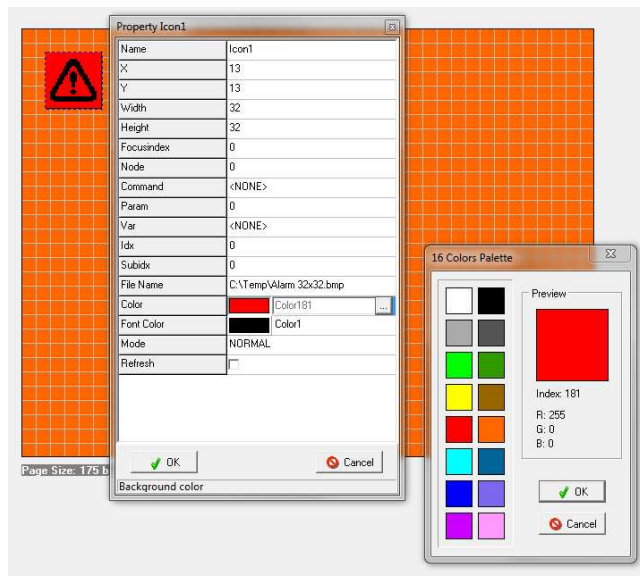


Палитра цветов UNI-PRO 3 состоит из 256 неизменяемых цветов.

### 3.5.7 Страницы с 16 цветами (EPJ Graph)

16-цветные страницы (320x240 16C) могут управлять элементами с цветовыми свойствами и растровыми изображениями, состоящими из двух цветов. Фон страницы может быть только одноцветным.

Для задания цветовых свойств элементов EIML необходимо использовать рамку, имеющую все 16 цветов палитры.

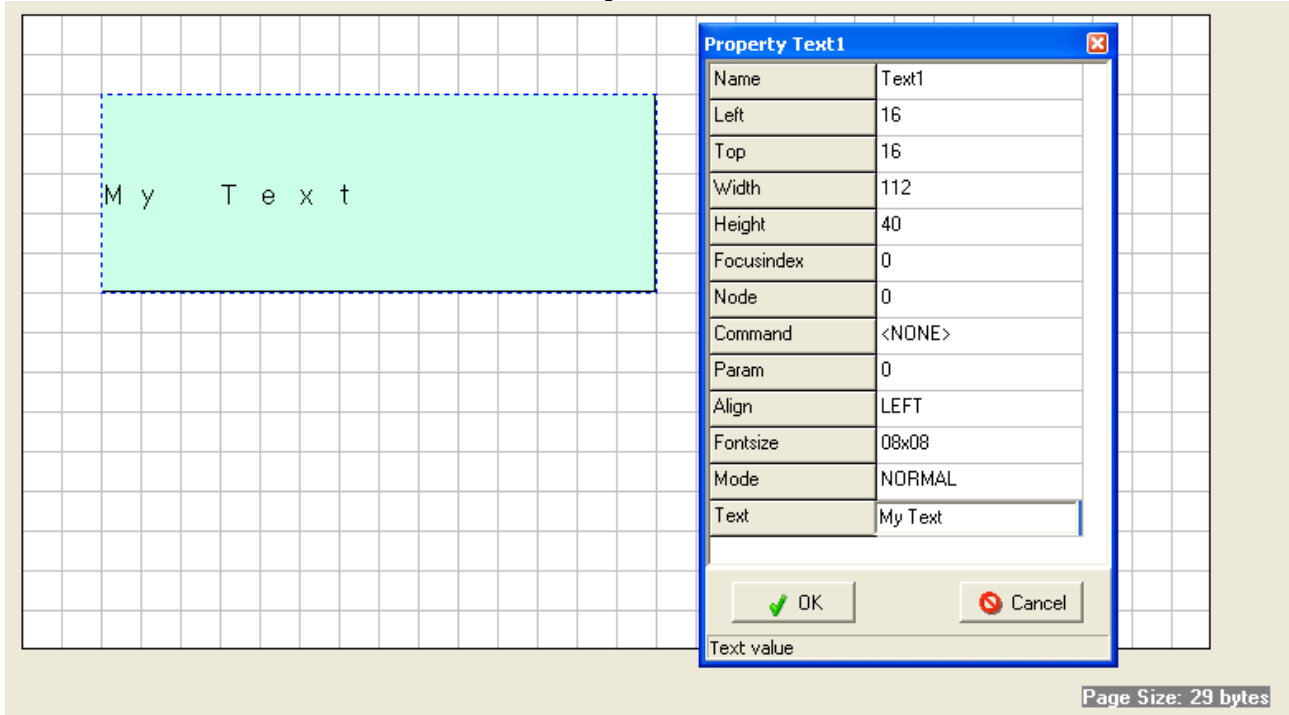


Пользовательские шрифты, которые можно использовать на этих страницах, являются подмножеством всех шрифтов, присутствующих в Uni-Pro, и их нельзя редактировать, поскольку они предварительно загружены в программы просмотра (EPJ Graph).


### 3.5.8 Свойства элемента

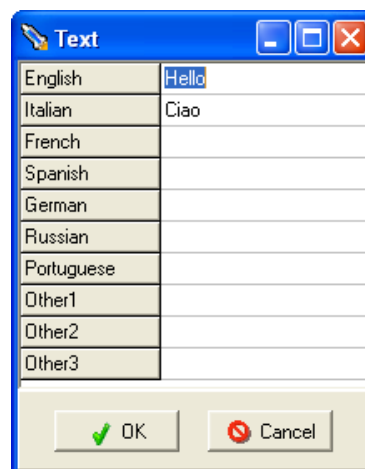
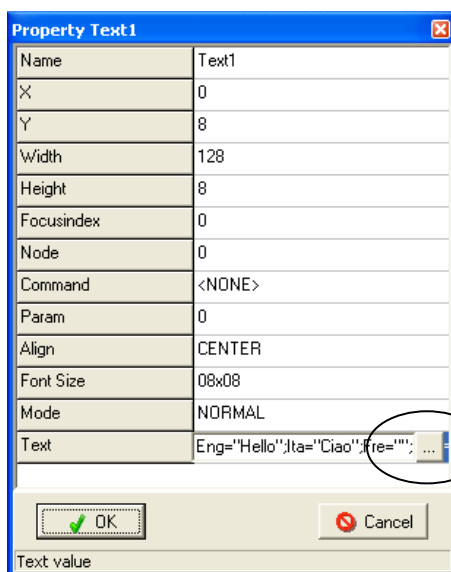
Каждый EIML-элемент имеет определенные свойства, которые позволяют установить формат его отображения и режим работы. Чтобы отобразить и изменить свойства элемента, выделите его, щелкните правой кнопкой мыши и выберите во всплывающем меню "*Properties*" или нажмите двойным щелчком на сам элемент.

Появится окно со списком всех свойств выбранного элемента.



Если страница является многоязычной, то свойство *Text* элементов Text **A**, текст в COMBO и текст в таблицах будет отображаться по-другому.

Чтобы установить отображение текста на желаемом языке, щелкните кнопку  в окне элемента; после этого появится следующее окно (для записи строк).



Если элементы являются многоязычными, связанными с идентификатором, тексты должны быть отредактированы с помощью специального инструмента в *Сервис->Таблица строк*.



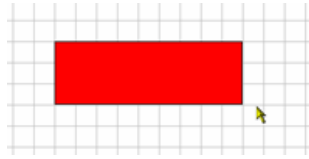
**Text (Текст)**

Элемент Text позволяет добавлять описания, активировать команды и подключать просмотр страницы.

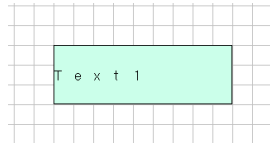
Как было описано в главе *Основные операции*, текстовый элемент имеет определенные свойства, которые позволяют графически располагать его внутри страницы (Left, Top, Width и Height), в то время как оставшиеся свойства описывают его работу.

<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>Left</b>	Смещение относительно левого края страницы, в пикселях.
<b>Top</b>	Смещение относительно верхнего края страницы, в пикселях.
<b>Width</b>	Ширина элемента, в пикселях.
<b>Height</b>	Высота элемента, в пикселях.
<b>Focusindex</b>	Свойство Focusindex устанавливает последовательность, в которой произойдет фокусирование на элементе при движении курсора. Допустимый диапазон – от 0 до 255. Если установлен 0, элемент не будет выделяться курсором. Если установлено другое значение, то оно показывает, через какое число движений курсора элемент будет выделен (для тех, кто знаком с программированием Windows, эта та же концепция, что и при управлении TabIndex).
<b>Command and Param</b>	Эти два свойства позволяют тексту отправлять команду контроллеру. Действие может быть приписано тексту с помощью установки команды, отличной от <NONE>. Когда курсор будет подведен к этому элементу и будет нажата кнопка “Enter”, появится запрос к контроллеру выполнить команду с параметром, установленным в свойстве param.
<b>Color (only color pages)</b>	Цвет, используемый для фона текста.
<b>Trasparent (only color pages)</b>	Включите прозрачность элемента, рисование текста выполняется без цвета фона.
<b>Align</b>	Допустимые значения LEFT, CENTER и RIGHT, которые устанавливают соответствующее выравнивание по левому краю, по центру и правому краю.
<b>Font Custom (only color pages)</b>	Указывает, какой пользовательский шрифт проекта используется для вывода текста. Если выбрано значение <NONE>, используется стандартный шрифт (свойство Fontsize).
<b>Fontsize</b>	Характеризует размер шрифта, используемого для введения текста. Например, если Вы используете размер шрифта 8x8, размер символа будет 8 пикселей в высоту и 8 пикселей в ширину.
<b>Font color (only color pages)</b>	Цвет шрифта текста.
<b>Mode</b>	Текст может быть отображен в четырех различных режимах: NORMAL NEGATIVE (only black and white pages) NORMAL-BLINKING NEGATIVE-BLINKING (only black and white pages) Если состояние NORMAL, текст будет отображен черным цветом на белом фоне; в случае NEGATIVE текст, наоборот, будет отображен белым цветом на черном фоне. Если Вы выберете “BLINKING” режимы, текст также будет мигать в соответствующем режиме.
<b>Node</b>	Физический узел для ассоциации с каждой командой.
<b>MultilanguageID</b>	ID в языковой базе. Если установлено значение <NONE>, текст будет тем, что указан в свойстве Text.
<b>Text</b>	Представляет текст, который Вы хотите отобразить.

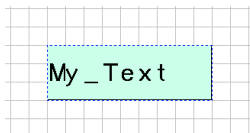
Чтобы добавить текст на страницу, выберите значок **Text** на панели инструментов EIML и обозначьте область, которую необходимо использовать для введения текста на странице; для этого нажмите и удерживайте левую кнопку мыши при передвижении курсора.



Когда Вы отпустите кнопку, будет нарисован текстовый элемент с установленным по умолчанию значением.



Теперь, чтобы ввести текст, который Вы хотите отобразить, откройте окно свойств (например, двойным щелчком по элементу текста) и измените свойство *Text* на необходимое. После этого Вы можете продолжить установку оставшихся свойств (*Fontsize*, *Align* и т.д.), а элемент будет отображен в измененном виде.

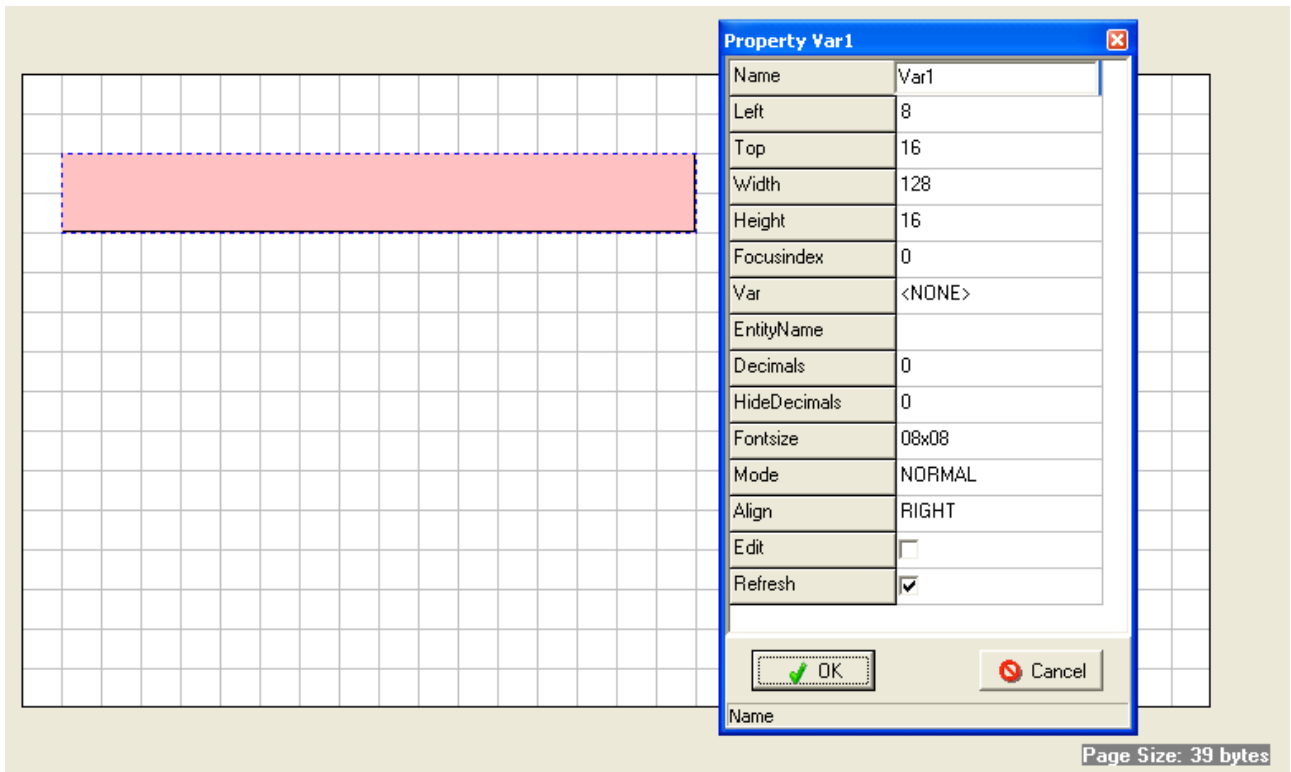


Если страница является многоязычной, то для изменения свойства *Text* необходимо использовать окно редактора многоязычных строк.

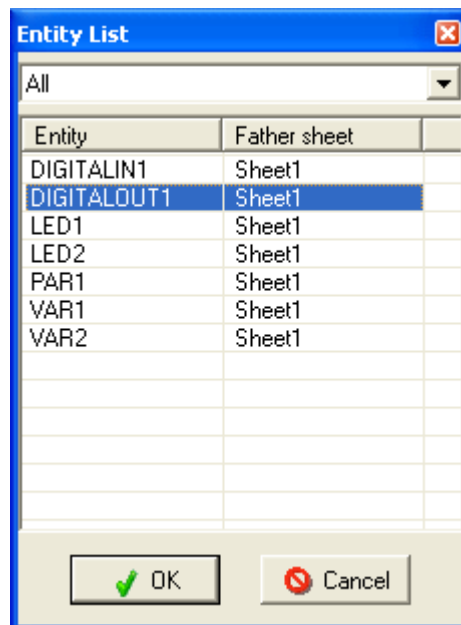
### **Variable (Переменная)**

Объекты *Variable* позволяют отображать и, если необходимо, устанавливать значения внутреннего состояния, входов, выходов и т.п.

Чтобы добавить переменную на страницу, проведите те же действия, что и при добавлении текста.



Элемент переменной может быть присоединен к переменной проекта: для этого выберите свойство *Var* (по умолчанию установлено как <NONE>) и нажмите появившуюся кнопку. Появится окно, в котором можете выбрать одну из сущностей проекта, значение которой Вы хотите отобразить.



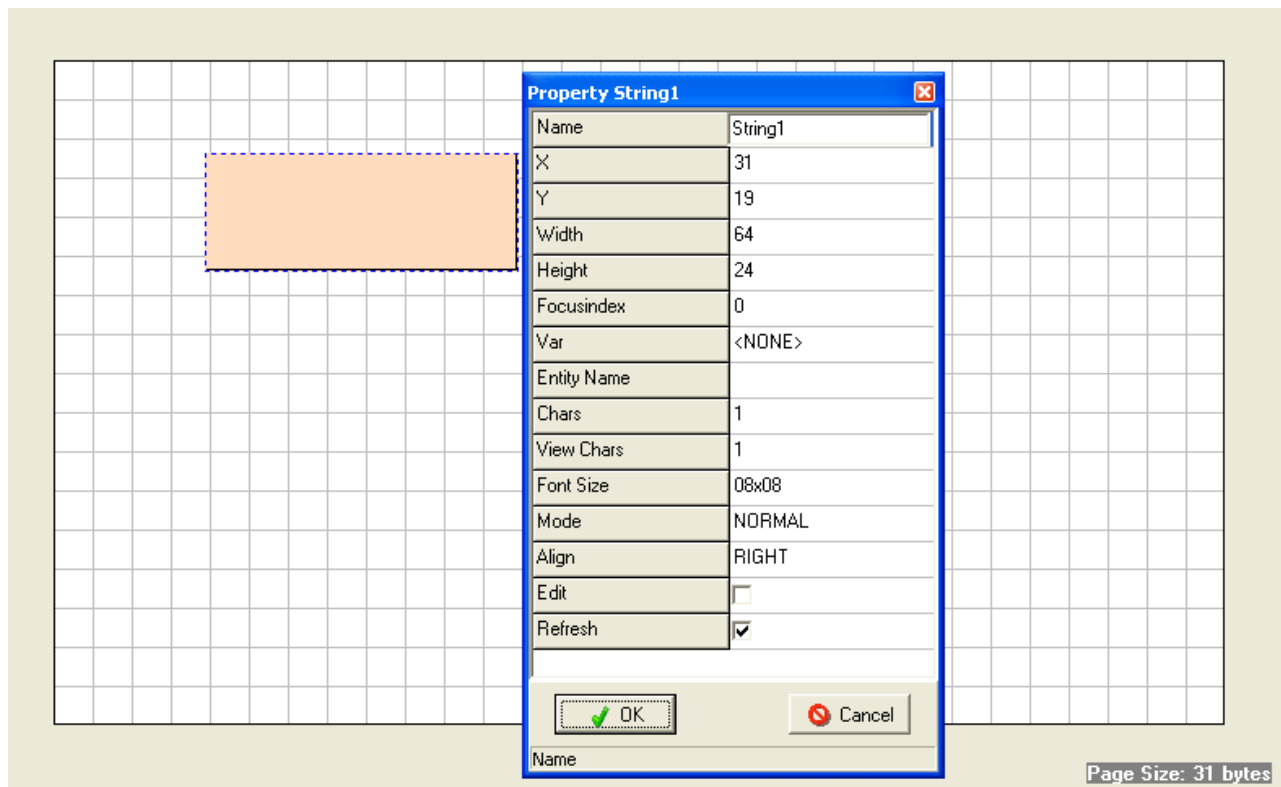
Далее будут описаны свойства переменных.

<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>Left</b>	Смещение относительно левого края страницы, в пикселях.
<b>Top</b>	Смещение относительно верхнего края страницы, в пикселях.

<b>Width</b>	Ширина элемента, в пикселях.
<b>Height</b>	Высота элемента, в пикселях.
<b>Focusindex</b>	Свойство Focusindex устанавливает последовательность, в которой произойдет фокусирование на элементе при движении курсора. Допустимый диапазон – от 0 до 255. Если установлен 0, элемент не будет выделяться курсором. Если установлено другое значение, то оно показывает, через какое число движений курсора элемент будет выделен (для тех, кто знаком с программированием Windows, эта та же концепция, что и при управлении TabIndex).
<b>Var</b>	Используя данное свойство, Вы можете присоединить сущность проекта (такую как Var, Par, Pers, DI, DO и т.д.). Выберите это свойство и нажмите на появившуюся кнопку: откроется окно, в котором можно выбрать одну из сущностей проекта, значение которой Вы хотите отобразить.
<b>CANOpen Var</b>	Это свойство позволяет подключать сущность из профилей CANOpen. Выбрав свойство и нажав на появившуюся кнопку, откроется окно списка объектов CANOpen, позволяющее выбрать объект для подключения.
<b>Decimals</b>	Число десятичных знаков, используемое для отображения значения [0...3]. Это свойство доступно только для чтения и устанавливается автоматически при присоединении сущности, которую Вы хотите отобразить (см. <i>Precision</i> в свойствах сущности).
<b>HideDecimals</b>	Позволяет изменить количество десятичных знаков, которое Вы хотите скрыть. Это свойство может принимать любое значение от 0 до значения свойства Decimals, и предназначено, чтобы скрыть несколько десятичных цифр.
<b>Color (only color pages)</b>	Цвет, используемый для фона переменной.
<b>Trasparent (only color pages)</b>	Включить прозрачность элемента, рисование переменных выполняется без цвета фона.
<b>Align</b>	Позволяет выровнять переменную по левому краю (LEFT), по центру (CENTER) или по правому краю (RIGHT).
<b>Font Custom (only color pages)</b>	Он указывает, какой пользовательский шрифт проекта используется для рисования текста. Если установлено значение <NONE>, используется стандартный шрифт (свойство Fontsize).
<b>Fontsize</b>	Характеризует размер шрифта, используемого для введения текста. Например, если Вы используете размер шрифта 8x8, размер символа будет 8 пикселей в высоту и 8 пикселей в ширину.
<b>Font Color (only color pages)</b>	Цвет, используемый для шрифта текста.
<b>Mode</b>	<p>Переменная может быть отображена в четырех различных режимах:</p> <p>NORMAL</p> <p>NEGATIVE (только черно-белые страницы)</p> <p>NORMAL-BLINKING</p> <p>NEGATIVE-BLINKING (только черно-белые страницы)</p> <p>Если состояние NORMAL, текст будет отображен черным цветом на белом фоне; в случае NEGATIVE текст, наоборот, будет отображен белым цветом на черном фоне.</p> <p>Если Вы выберете “BLINKING” режимы, текст также будет мигать в соответствующем режиме.</p>
<b>Edit</b>	Активируйте это свойство, чтобы включить режим редактирования переменных и изменить связанное значение (для редактирования переменной необходимо, чтобы для свойства focusindex было установлено значение, отличное от нуля).
<b>Refresh</b>	Если свойство Refresh включено, переменная будет постоянно запрашиваться; в противном случае он будет обновляться медленно, чтобы избежать перегрузки связи между контроллерами.

**Strings (Строки)**

Объекты *String* позволяют отображать и устанавливать строковые значения или сущности CJ\_CHAR.



Ассоциирование элементов String в EIML-страницах с сущностями CJ\_CHAR возможно так же, как и для переменных, с использованием инструмента Entity List/Список сущностей (см. пункт **Переменные**). С помощью инструмента Entity list можно задать первые символы строки, с которых начнется отображение строки; для этого достаточно выбрать из списка желаемую позицию массива элементов CJ\_CHAR. Свойство focusIndex элемента String всегда ссылается на положение первых изменяемых символов; остальные можно выбрать с помощью кнопок пользовательского интерфейса.

Далее представлено детальное описание всех свойств значка:

<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>Left</b>	Смещение относительно левого края страницы, в пикселях.
<b>Top</b>	Смещение относительно верхнего края страницы, в пикселях.
<b>Width</b>	Ширина элемента, в пикселях.
<b>Height</b>	Высота элемента, в пикселях.
<b>Focusindex</b>	Свойство Focusindex устанавливает последовательность, в которой произойдет фокусирование на элементе при движении курсора. Допустимый диапазон – от 0 до 255. Если установлен 0, элемент не будет выделяться курсором. Если установлено другое значение, то оно показывает, через какое число движений курсора элемент будет выделен (для тех, кто знаком с программированием Windows, эта та же концепция, что и при управлении TabIndex).
<b>Var</b>	Используя данное свойство, Вы можете присоединить сущность проекта (такую как Var, Par, Pers, DI, DO и т.д.). Выберите это свойство и нажмите на появившуюся кнопку: откроется окно, в котором можно выбрать одну из сущностей проекта, значение которой Вы хотите отобразить.

<b>CANOpen Var</b>	Это свойство позволяет подключать сущность из профилей CANOpen. Выбрав свойство и нажав на появившуюся кнопку, откроется окно списка объектов CANOpen, позволяющее выбрать объект для подключения.
<b>Chars</b>	Представляет количество символов, из которых состоит строка.
<b>ViewChars</b>	Устанавливает число символов, которые будут отображены, начиная с первого символа строки.
<b>Color (only color pages)</b>	Цвет, используемый для фона строки.
<b>Trasparent (only color pages)</b>	Включите прозрачность элемента, рисование строки выполняется без цвета фона.
<b>Align</b>	Позволяет выравнивать строку по левому краю (LEFT), по центру (CENTER) или по правому краю (RIGHT).
<b>Font Custom (only color pages)</b>	Он указывает, какой пользовательский шрифт проекта используется для рисования текста. Если установлено значение <NONE>, используется стандартный шрифт (свойство Fontsize).
<b>Fontsize</b>	Размер шрифта устанавливает размер шрифта, которым будет написан текст. Например, при использовании 8x8 символ будет иметь высоту восемь пикселей и ширину восемь пикселей.
<b>Font color (only color pages)</b>	Цвет шрифта текста.
<b>Mode</b>	Переменная может быть отображена в четырех различных режимах: NORMAL NEGATIVE (только черно-белые страницы) NORMAL-BLINKING NEGATIVE-BLINKING (только черно-белые страницы) Если состояние NORMAL, текст будет отображен черным цветом на белом фоне; в случае NEGATIVE текст, наоборот, будет отображен белым цветом на черном фоне. Если Вы выберете "BLINKING" режимы, текст также будет мигать в соответствующем режиме.
<b>Edit</b>	Активация этого свойства включает режим редактирования переменной и изменение соответствующего значения (редактирование переменной требует, чтобы значение свойства focusindex было установлено как отличное от нуля).
<b>Refresh</b>	Если свойство Refresh включено, переменная будет запрашиваться постоянно; в противном случае, она будет обновляться не так часто, чтобы избежать перегрузки соединения между контроллерами.

### Icons (Значки)

Значки делают EIML-страницы более красочными и позволяют отображать состояние или активировать команду.

Далее представлено детальное описание всех свойств значка:

<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>Left</b>	Смещение относительно левого края страницы, в пикселях.
<b>Top</b>	Смещение относительно верхнего края страницы, в пикселях.

<b>Width</b>	Ширина элемента, в пикселях.
<b>Height</b>	Высота элемента, в пикселях.
<b>Focusindex</b>	Свойство focusindex определяет последовательность, в которой элемент будет получать фокус при использовании курсора. Допустимый диапазон от 0 до 255. Если установлено значение 0, элемент никогда не будет выбран курсором. Если установлено другое значение, определяет, через сколько перемещений курсора он будет выбран.
<b>Command and Param</b>	Эти два свойства позволяют тексту отправлять команду контроллеру. Действие может быть приписано тексту с помощью установки команды, отличной от <NONE>. Когда курсор будет подведен к этому элементу и будет нажата кнопка “Enter”, появится запрос к контроллеру выполнить команду с параметром, установленным в свойстве param.
<b>Var(*)</b>	Свойство Var позволяет присоединять сущность к значку таким же образом, как и переменные. Однако, в отличие от переменных, здесь Вы можете присоединить только сущности типов CJ_BIT и CJ_BYTE. В зависимости от значения, считываемого с сущности, значок будет отображен в следующих режимах: 0 = NORMAL 1 = NEGATIVE 2 = NORMAL-BLINK 3 = NEGATIVE-BLINK
<b>CANOpen Var</b>	Это свойство позволяет подключать сущность из профилей CANOpen. Выбрав свойство и нажав на появившуюся кнопку, откроется окно списка объектов CANOpen, позволяющее выбрать объект для подключения.
<b>Filename</b>	Свойство Filename указывает исходное изображение, из которого точечный рисунок был загружен. Используя это свойство, Вы можете изменить значок после того, как он был добавлен в проект. Вы можете выбирать значки только в графическом формате точечного рисунка Windows, который будет загружен и преобразован в черно-белый.
<b>Bitmap</b>	Ссылка на значок, загруженный в инструмент растровых изображений EIML (альтернатива свойству Filename, только для страниц с 256 цветами).
<b>Color (only 16 colors pages)</b>	Цвет значка 1 (только страницы с 16 цветами).
<b>Font color (only 16 colors pages)</b>	Цвет значка 2 (только страницы с 16 цветами).
<b>Mode(*)</b>	Значок может быть отображен в четырех различных режимах: NORMAL NEGATIVE (только черно-белые страницы) NORMAL-BLINKING NEGATIVE-BLINKING (только черно-белые страницы) Если состояние NORMAL, текст будет отображен черным цветом на белом фоне; в случае NEGATIVE текст, наоборот, будет отображен белым цветом на черном фоне. Если Вы выберете “BLINKING” режимы, текст также будет мигать в соответствующем режиме.
<b>Refresh</b>	Свойство Refresh позволяет установить, как часто будет запрашиваться значение сущности, соединенной со значком. Если значок не присоединен к какой-либо сущности, свойство будет пропущено.

На цветных страницах значки показаны с использованием цветовой палитры UNI-PRO, затем, если растровое изображение для загрузки не состоит из 256 цветов палитры, необходимо будет подождать несколько секунд, пока изображение преобразуется. Показанный значок может отличаться от оригинала.

**Примечание:** если свойство **Var** установлено, то Режим (Mode) напрямую ему соответствует независимо от самого значения свойства **Mode**.

## Combo

Объекты Combo являются инновационным решением для предоставления информации, содержащейся в сущности. С помощью этих объектов Вы можете присоединять текст или значок к каждому значению, принимаемому сущностью, таким образом обеспечивая гибкое и удобное отображение содержания сущности.

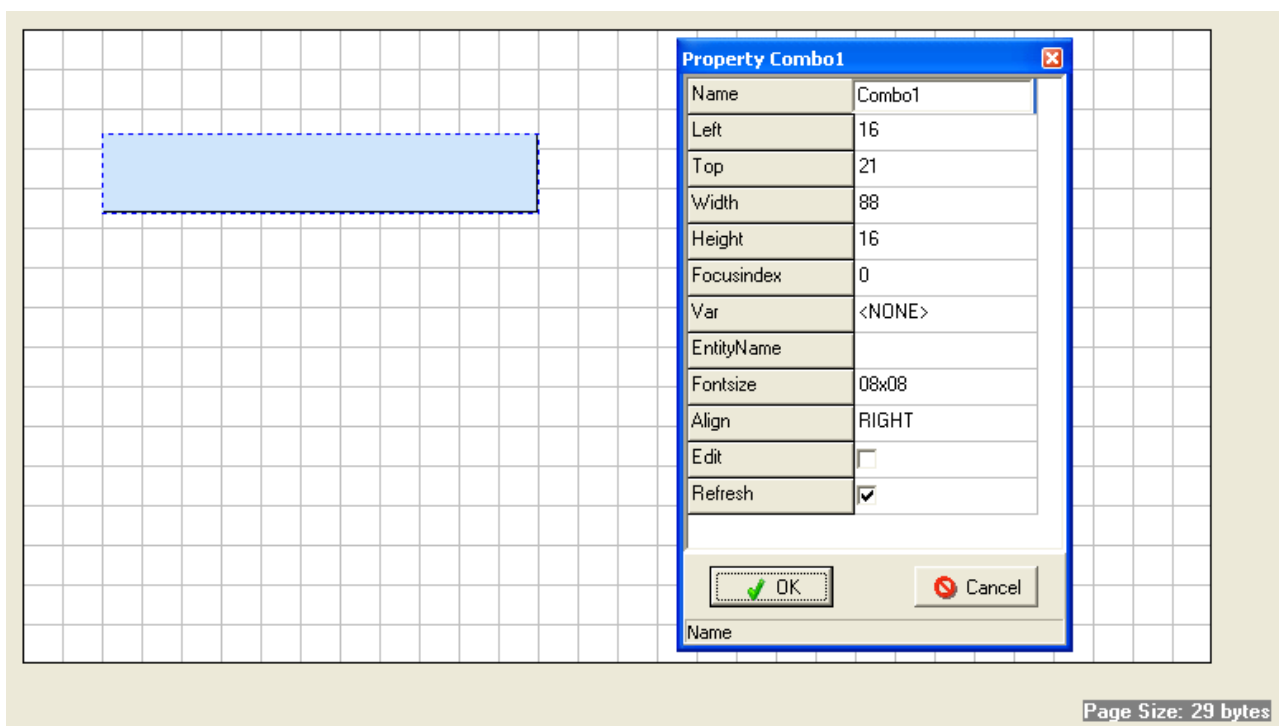
В отличие от переменных, объекты Combo не включают следующие свойства: **Mode**, **Decimals** и **HideDecimals**. В дополнение, к ним могут быть присоединены только сущности типов CJ\_BIT или CJ\_BYTE.

<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>Left</b>	Смещение относительно левого края страницы, в пикселях.
<b>Top</b>	Смещение относительно верхнего края страницы, в пикселях.
<b>Width</b>	Ширина элемента, в пикселях.
<b>Height</b>	Высота элемента, в пикселях.
<b>Focusindex</b>	Свойство Focusindex устанавливает последовательность, в которой произойдет фокусирование на элементе при движении курсора. Допустимый диапазон – от 0 до 255. Если установлен 0, элемент не будет выделяться курсором. Если установлено другое значение, то оно показывает, через какое число движений курсора элемент будет выделен (для тех, кто знаком с программированием Windows, эта та же концепция, что и при управлении TabIndex).
<b>Var</b>	Используя данное свойство, Вы можете присоединить сущность проекта (такую как Var, Par, Pers, DI, DO и т.д.). Выберите это свойство и нажмите на появившуюся кнопку: откроется окно, в котором можно выбрать одну из сущностей проекта, значение которой Вы хотите отобразить.
<b>CANOpen Var</b>	Это свойство позволяет подключать сущность из профилей CANOpen. Выбрав свойство и нажав на появившуюся кнопку, откроется окно списка объектов CANOpen, позволяющее выбрать объект для подключения.
<b>Font Custom (only color pages)</b>	Он указывает, какой пользовательский шрифт проекта используется для рисования элементов комбинированного текста. Если установлено значение <NONE>, используется стандартный шрифт (свойство Fontsize).
<b>Fontsize</b>	Описывает размер шрифта, используемый для написания текста. Например, если вы используете размер шрифта 8x8, символ будет иметь высоту 8 пикселей и ширину 8 пикселей.
<b>Align</b>	Позволяет выравнивать переменную по левому краю (LEFT), по центру (CENTER) и правому краю (RIGHT).
<b>Edit</b>	Активация этого свойства включает режим редактирования переменной и изменение соответствующего значения (редактирование переменной требует, чтобы значение свойства focusindex было установлено как отличное от нуля).
<b>Refresh</b>	Если свойство Refresh включено, переменная будет запрашиваться постоянно; в противном случае, она будет обновляться не так часто, чтобы избежать перегрузки соединения между контроллерами.

Отличительной особенностью объектов Combo является программа Combo Wizard, позволяющая отображать и редактировать элементы, которые Вы хотите соединить со значениями сущностей.

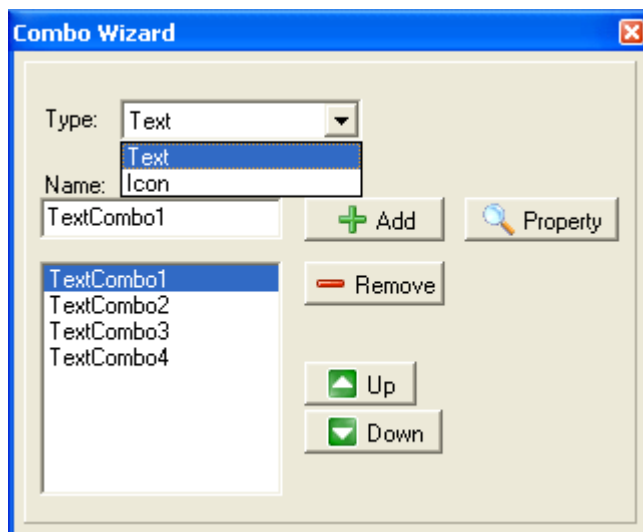


Теперь мы добавим объект Combo на нашу страницу и откроем окно свойств:



Чтобы соединить Combo с сущностью, выберите свойство *Var* (установленное, по умолчанию, как <NONE>), нажмите на появившуюся справа кнопку и выберите необходимую переменную из списка сущностей.

Нажмите правой кнопкой мыши на элемент Combo и во всплывающем меню выберите Combo Wizard. Появится следующее окно:



Чтобы связать тексты или значки со значениями присоединяемой сущности, выделите тип объекта, который Вы хотите добавить из выпадающего меню (Text/Icon) и присвойте ему имя, затем нажмите **Add**. Элементы будут добавлены в список снизу в том порядке, в котором он были введены. Первый элемент будет отображен, когда присоединенная сущность примет значение «ноль», второй – когда присоединенная сущность примет значение «единица» и т.д.

Чтобы изменить последовательность, выделите элемент, который Вы хотите переместить, и нажмите соответственно кнопку **Move Up** или **Move Down**.

Как только все элементы были добавлены, Вы можете приступить к их *конфигурации*: чтобы отобразить свойства элемента, выделите его в списке и нажмите **Properties**. Чтобы определить значение, обратитесь к свойствам текстов и значков.

Если страница является многоязычной, то для изменения свойства *Text* необходимо использовать окно редактора многоязычных строк. В этом случае просмотр доступен только с использованием EIML-симулятора.

Чтобы понять, как использовать объекты Combo, обратитесь к модели Icon, находящейся в папке Samples.

### Table (Таблица)

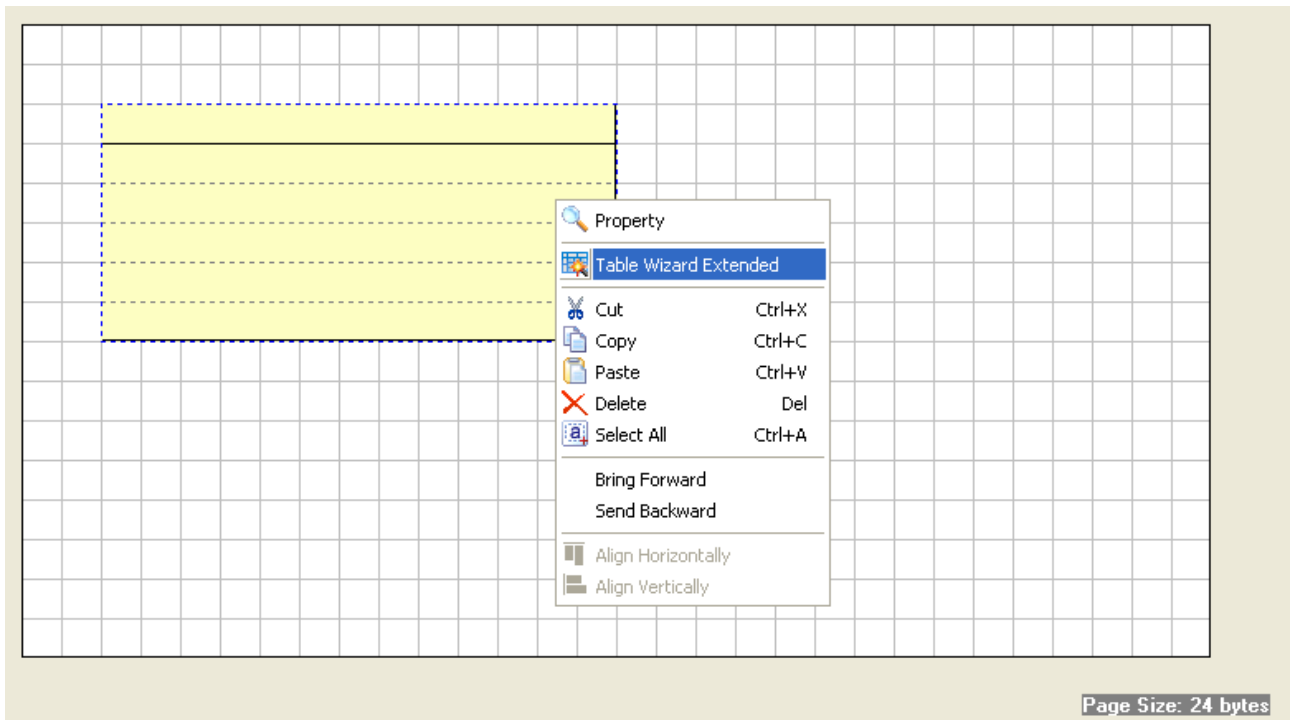
Таблицы являются дополнительным инновационным объектным типом, созданным для помощи при разработке графических интерфейсов и характеризующимся возможностью **прокручивания**.

Таким образом, используя таблицы, Вы сможете отображать большее количество данных без необходимости загрузки больше чем одной страницы.

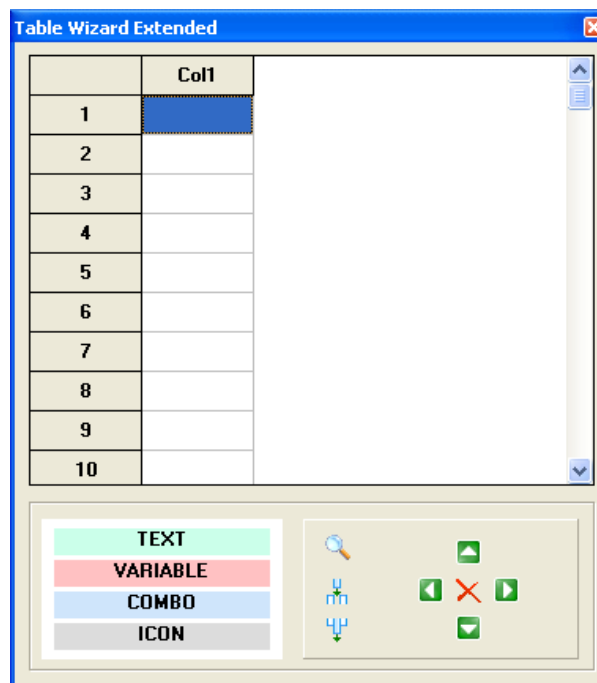
Если добавить таблицу в страницу, можно будет отобразить ее свойства:

<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>Left</b>	Смещение относительно левого края страницы, в пикселях.
<b>Top</b>	Смещение относительно верхнего края страницы, в пикселях.
<b>Width</b>	Ширина элемента, в пикселях.
<b>Height</b>	Высота элемента, в пикселях.
<b>Nrow</b>	Указывает число видимых строк таблицы. Это свойство определяется автоматически, исходя из высоты таблицы, высоты строки, а также того, имеет ли таблица заголовки.
<b>RowHeight</b>	Определяет высоту строки в пикселях. Это логически связано с высотой таблицы, отображением заголовков, и общим числом строк таблицы.
<b>Color (only color pages)</b>	Цвет, используемый в качестве фона таблицы.
<b>Font Custom (only color pages)</b>	Указывает, какой пользовательский шрифт проекта используется для рисования текстовых элементов таблицы. Если установлено значение <NONE>, используется стандартный шрифт (свойство Fontsize).
<b>Fontsize</b>	Это свойство определяет размер шрифта всех элементов текста, отображаемых в таблице.
<b>Header Color (only color pages)</b>	Цвет фона заголовка.
<b>Header Font Color (only color pages)</b>	Цвет шрифта заголовка.
<b>Header</b>	Установите это свойство, чтобы отображать заголовки столбцов или нет.
<b>Borders</b>	Если включено, будут показаны границы таблицы; иначе, границы показаны не будут.
<b>Borders Colors (only color pages)</b>	Цвета линий краев таблицы.

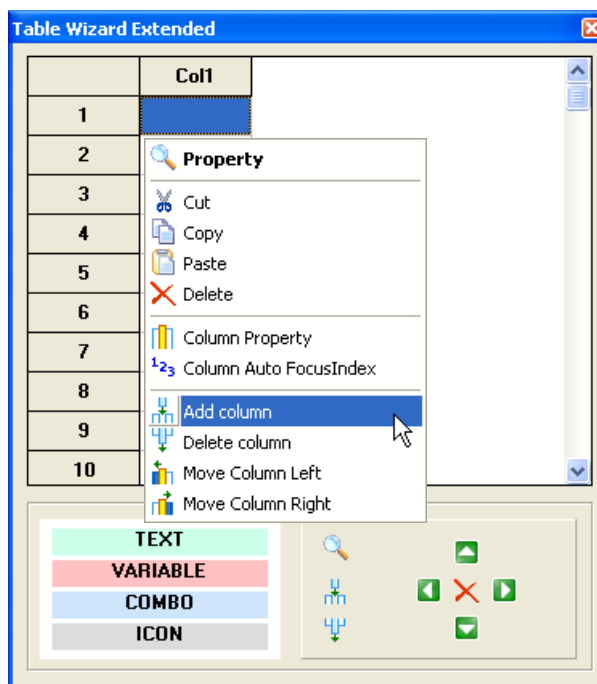
Чтобы добавить объект в таблицу, используйте программу **Table Wizard Extended** (нажмите правой кнопкой на элемент и затем выберите Table Wizard Extended).



Появится следующее окно:



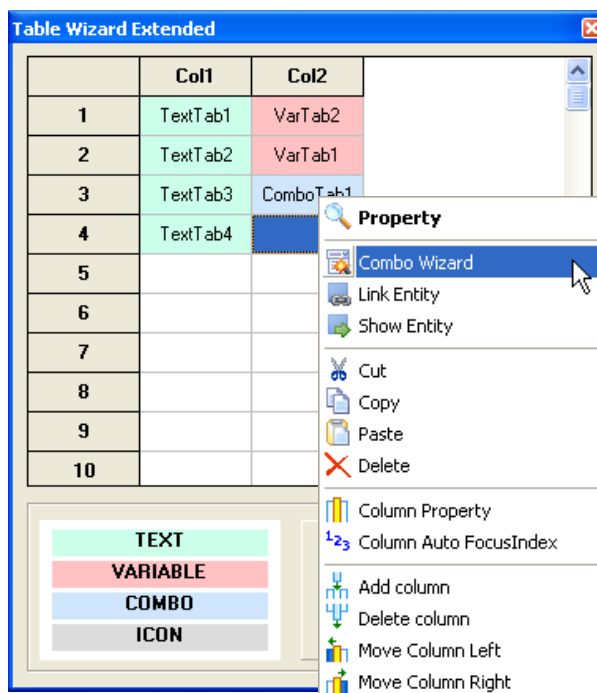
Первое, что нужно сделать для использования таблицы, - это создать ее столбцы. Объект Table позволяет добавлять от одного до четырех столбцов. По умолчанию таблица имеет один активный столбец. Можно создавать новые столбцы с использованием контекстного меню (правый щелчок по столбцу и выбор *Add Column*):



Для каждого столбца Вы можете установить его название, ширину, заголовок и выравнивание выбором *Column Property*, а также изменить его положение с помощью *Move Column Left* и *Move Column Right*.

Если страница является многоязычной, то для изменения свойства *Text* необходимо использовать окно редактора многоязычных строк. В этом случае просмотр доступен только с использованием EIML-симулятора.

Чтобы вставить необходимый элемент в каждый столбец, выберите его из списка в нижней части окна и с помощью мыши перетащите в требуемое положение в столбце:



Используя кнопки позиционирования, расположенные справа в нижней части окна, можно перемещать выбранный элемент в таблице: для перемещения элемента между строками и столбцами нажимайте соответствующие стрелки, а чтобы удалить элемент, нажмите на значок X, расположенный в центре.

Чтобы изменить свойства элемента, выделите его и выберите пункт **Property** из контекстного меню или дважды щёлкните по выбранному элементу.

Из контекстного меню (нажмите правой кнопкой мыши на выделенный элемент) Вы можете скопировать или просмотреть свойства выбранного элемента: например, Вы можете присоединить переменную или combo к сущности проекта. Если Вы уже добавили Combo в список, то после настройки его свойств Вам понадобится отобразить **Combo Wizard** для выбора присоединяемых элементов.

Некоторые операции могут быть осуществлены с использованием комбинаций кнопок. В таблице ниже приведены действия, ассоциированные с этими комбинациями.

Кнопка	Действие
<b>Ctrl+Up, Ctrl+Down, Ctrl+Left, Ctrl+Right</b>	Перемещение выделенного элемента в таблице.
<b>Ctrl+C, Ctrl+Ins</b>	Копирование выбранного элемента.
<b>Ctrl+X</b>	Вырезание выделенного элемента.
<b>Ctrl+V, Shift+Ins</b>	Вставка выделенного элемента.
<b>Canc</b>	Удаление выделенного элемента.
<b>Enter, F11</b>	Просмотр свойств выделенного элемента.
<b>Ctrl+1, Ctrl+2, Ctrl+3, Ctrl+4, Ctrl+5</b>	Вставка в столбец текста, переменной, combo или текста соответственно.

### Dynamic List (Динамический список)

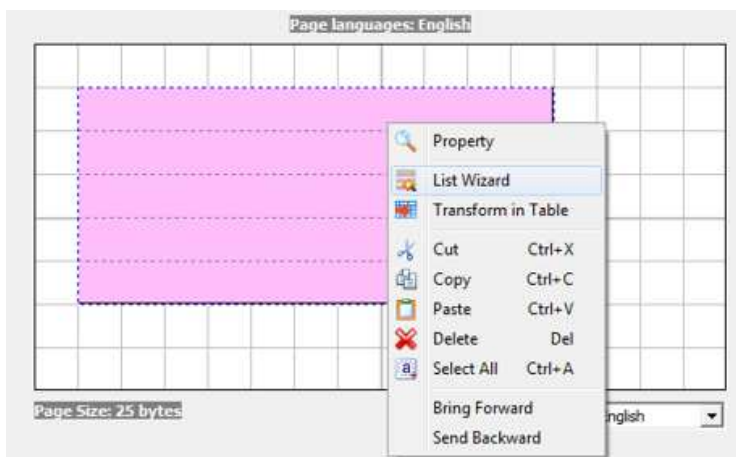
Объекты списки — это элементы, очень похожие на таблицы, плюс возможность динамического скрытия или отображения строк списка. Видимость строки может быть связана с сущностью типа CJ\_BIT. Отображение списка в зависимости от статуса видимости каждой строки, взрывает или сжимает строки автоматически и динамически.

Ниже приведены свойства элемента Список:

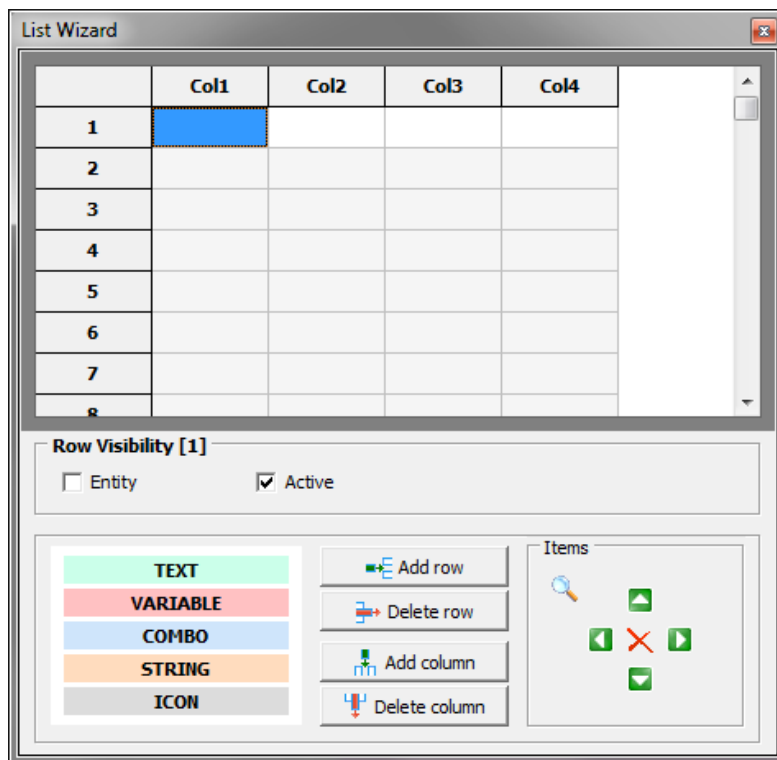
<b>Name</b>	Уникальное имя, идентифицирующее элемент внутри проекта.
<b>Left</b>	Смещение от левого края страницы в пикселях.
<b>Top</b>	Смещение от верхнего края страницы в пикселях.
<b>Width</b>	Ширина элемента в пикселях.
<b>Height</b>	Высота элемента в пикселях.
<b>Nrow</b>	Указывает количество видимых строк в списке. Это свойство вычисляется автоматически на основе высоты списка, высоты строки и наличия заголовков в списке.
<b>RowHeight</b>	Определяет высоту строки в пикселях. Он логически связан с высотой списка, отображением заголовка и общим количеством строк списка.
<b>Color (only color pages)</b>	Цвет, используемый в качестве фона списка.
<b>Font Custom (only color pages)</b>	Он указывает, какой пользовательский шрифт проекта используется для рисования текстовых элементов списка. Если установлено значение <NONE>, используется стандартный шрифт (свойство Fontsize).
<b>Fontsize</b>	Это свойство определяет размер шрифта всех текстовых элементов, отображаемых в списке.
<b>Header Color (only color pages)</b>	Цвет фона заголовка.
<b>Header Font Color (only color pages)</b>	Цвет шрифта заголовка.

<b>Header</b>	Установите это свойство, чтобы выбрать, отображать ли заголовки столбцов.
<b>Borders</b>	Если включено, границы таблицы будут отображаться, иначе не будут отображаться.
<b>Borders Colors (only color pages)</b>	Цвета линий краев списка.

Чтобы добавить объекты в список, используйте **List Wizard** (щелкните элемент правой кнопкой мыши и выберите пункт меню **List Wizard**).



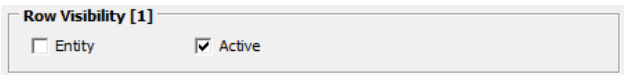
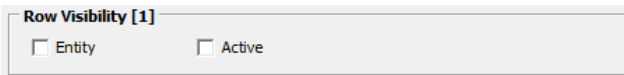
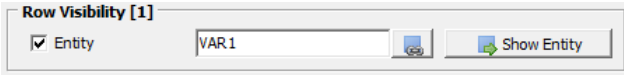
Появится следующее окно конфигурации:



Использование очень похоже на Table Wizard Extended, плюс у него есть возможность настроить видимость каждой строки с сущностями проекта.

Для определения видимости строки необходимо установить два свойства Entity и Active в секции Видимость строки [x] (значение, содержащееся в скобках, идентифицирует отображаемый номер строки).

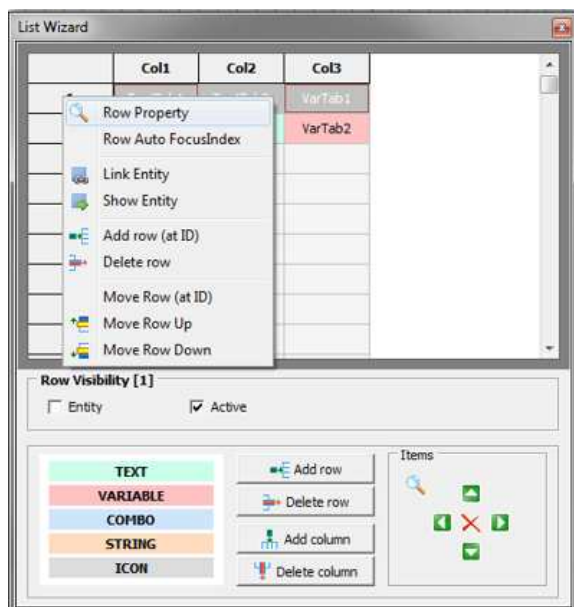
В зависимости от значения двух свойств видимость строки может быть трех типов:

-  Всегда видима.
-  Всегда невидима. Строка, которая всегда невидима, полезна, если, например, вы используете один и тот же список на нескольких страницах, где вам нужно скрыть некоторую информацию, но вы хотите сохранить одну и ту же структуру данных.
-  Отображается сущностью. Видимость строки обусловлена значением связанного объекта. Сущность должна быть типа CJ\_VIT, при =1 строка видна, при =0 строка скрыта. Для подключения сущности необходимо активировать свойство Сущность и выбрать ресурс в появившемся окне Список сущностей. Чтобы изменить связанный объект, нажмите кнопку для ссылки на объект, Показать сущность отобразит связанный объект на рабочем листе.

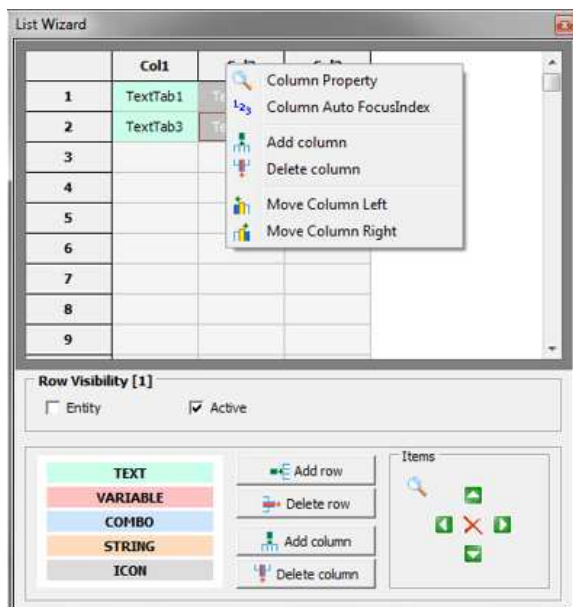
Видимость строки также может быть связана с сущностью CANOrep.

Прежде чем вы сможете вставлять элементы, вам нужно создать строки (а также столбцы). Каждый новый список по умолчанию имеет одну строку и один столбец. Добавление новой строки осуществляется с помощью кнопки *Добавить строку* и путем ввода индекса строки (по умолчанию он вставляется внизу), если вставлен уже занятый индекс, новая строка будет вставлена в эту позицию, а все последующие строки будут перемещены вниз.

При нажатии правой кнопки на индексы строк появится контекстное меню для выполнения действий со строками, при нажатии на заголовки столбцов появится контекстное меню для выполнения действий со столбцами.



Меню строки

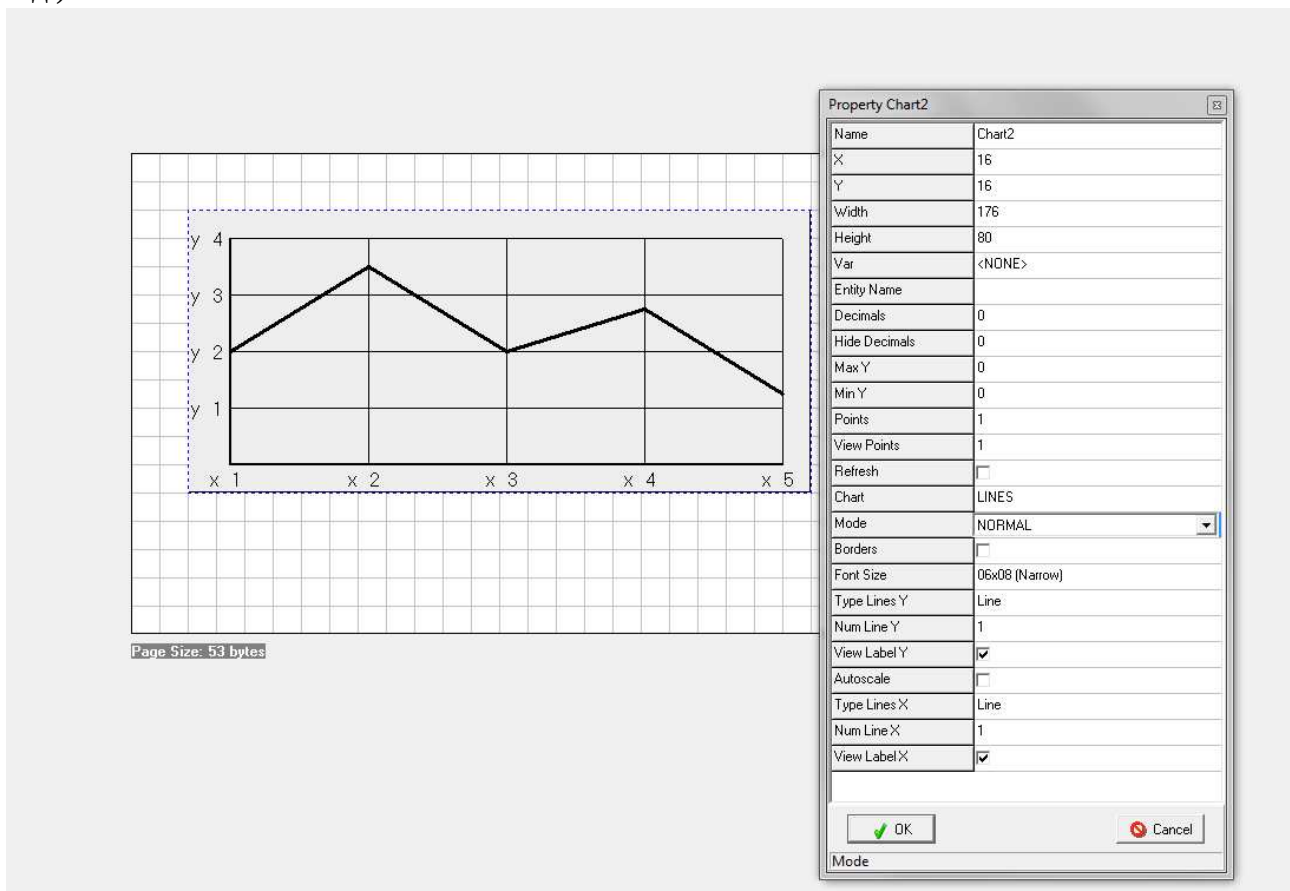


Меню столбца

С помощью кнопок позиционирования в правой нижней части можно переместить выбранный элемент в список, чтобы переместить его между строками и столбцами, вы должны нажать на соответствующие стрелки, чтобы удалить его, вы должны нажать на X в середине. Чтобы изменить свойства элемента, выделите его и выберите в контекстном меню Свойства, или дважды щелкните элемент в фокусе. Из контекстного меню (щелчок правой кнопкой мыши по выбранному элементу) можно скопировать его или просмотреть его свойства, например, можно связать переменную или комбо с сущностью проекта. Если вы добавили комбо в список, после настройки его свойств вам нужно будет показать мастер комбо, чтобы выбрать, какие элементы вы хотите связать с ним.

## Chart (График)

Объекты График позволяют построить простой график, подключив переменную как объект массива, можно представить диапазон до 100 точек. Существует ряд других свойств, которые позволяют настраивать график в соответствии с потребностями (тип, оси, цвет, значения и т. д.).



Связь графика с сущностью проекта возможна так же, как и с переменными, с помощью инструмента *Список сущностей* (см. Переменные). С помощью инструмента также можно задать первую точку, с которой начинать отрисовку графика, для этого достаточно просто выбрать из списка нужную позицию в массиве. Чтобы начать проектирование с первой точки, выберите расположение массива *Переменная[0]*. Для страниц можно нарисовать до 3 разных графиков по одной сущности, однако они будут заполняться произвольно, заранее определяется, какими будут индексы первой, второй и третьей линии, первая линия всегда будет эталонной.

После этого свойства элементов диаграммы:



<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>X</b>	Смещение в пикселях от левой стороны страницы.
<b>Y</b>	Смещение в пикселях от верха страницы.
<b>Width</b>	Ширина элемента, в пикселях.
<b>Height</b>	Высота элемента, в пикселях.
<b>Var</b>	Используя данное свойство, Вы можете присоединить сущность проекта (такую как Var, Par, Pers, DI, DO и т.д.). Выберите это свойство и нажмите на появившуюся кнопку: откроется окно, в котором можно выбрать одну из сущностей проекта, значение которой Вы хотите отобразить.
<b>CANOpen Var</b>	Это свойство позволяет подключать сущность из профилей CANOpen. При выборе свойства и нажатии на появившуюся кнопку открывается окно списка объектов CANOpen, позволяющее выбрать объект для подключения.
<b>Decimals</b>	Количество десятичных знаков, используемых для представления значения [0..3] по оси Y. Это свойство доступно только для чтения и устанавливается автоматически, когда объект, который вы хотите отобразить, объединяется (см. раздел «Точность» в разделе «Свойства объекта»).
<b>HideDecimals</b>	Позволяет вам изменить количество десятичных знаков, которые вы хотите скрыть. Это свойство может принимать любое значение от 0 до значения свойства Decimals и предназначено для сокрытия некоторых десятичных цифр.
<b>Max Y</b>	Максимальное значение Y. Точки выше этого значения будут исключены из линии графика.
<b>Min Y</b>	Минимальное значение Y. Точки ниже этого значения будут исключены из линии графика.
<b>View Points</b>	Устанавливает количество точек, которые будут нарисованы от начальной точки графика.
<b>Idx S2 (only color pages)</b>	Индекс массива, указывающий на начало линии 2. Если = 0, линия 2 не рисуется.
<b>Idx S3 (only color pages)</b>	Индекс массива, указывающий на начало линии 3. Если = 0, линия 3 не рисуется.
<b>Refresh</b>	Если свойство Refresh включено, переменная будет запрашиваться постоянно; в противном случае, она будет обновляться не так часто, чтобы избежать перегрузки соединения между контроллерами.
<b>Chart</b>	Это позволяет выбрать тип графика: POINTS (точки). LINES (линии). HISTOGRAM (столбчатая диаграмма).
<b>Color (only color pages)</b>	Цвет, используемый для фона графика.
<b>Axis Color (only color pages)</b>	Цвет, используемый для оси графика
<b>Lines Color (only color pages)</b>	Цвет, используемый для линий/галочек на графике
<b>Mode (only pages in black &amp; white)</b>	Режим отображения графика, он может быть двух видов: NORMAL NEGATIVE (только для черно-белой страницы)

	Статус NORMAL позволяет просматривать сегменты графика и текста черным цветом на белом фоне; наоборот, НЕГАТИВ устанавливает оси и белый текст на черном фоне.
<b>Borders</b> (only pages in black & white)	Если включено, границы области диаграммы будут отображаться, в противном случае они не будут отображаться.
<b>Fontsize</b>	Описывает размер шрифта, используемый для написания метки рядом с осями. Например, если вы используете размер шрифта 8x8, символ будет иметь высоту 8 пикселей и ширину 8 пикселей.
<b>Font Color</b> (only color pages)	Цвет, используемый для шрифта метки.
<b>Type Lines Y</b>	Это позволяет отображать горизонтальную сетку под графиком: NO: нет линий LINE: горизонтальные линии TICK: отметки на оси Y
<b>Num Lines Y</b>	Количество линий/галочек по оси Y [1..4].
<b>View Label Y</b>	Это позволяет отображать значение рядом с осью Y.
<b>Autoscale</b>	Если эта функция активирована, график будет автоматически масштабироваться между максимальной и минимальной точкой линии 1.
<b>Type Lines X</b>	Позволяет отображать вертикальную сетку под графиком: NO: нет линий LINE: вертикальные линии TICK: отметки на оси X
<b>Num Lines X</b>	Количество линий/галочек по оси X [1..Видимые точки].
<b>View Label X</b>	Это позволяет отображать значение рядом с осью x.
<b>S1 Color</b> (only color pages)	Цвет, используемый для Линии 1.
<b>S2 Color</b> (only color pages)	Цвет, используемый для Линии 2.
<b>S3 Color</b> (only color pages)	Цвет, используемый для Линии 3.

**Примечание.** Для каждой страницы EIML разрешено использовать только один объект График.

### Line (Линия)

Линия имеет следующие свойства:

<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>X1</b>	X-координата первой указанной точки.
<b>Y1</b>	Y-координата первой указанной точки.
<b>X2</b>	X-координата второй указанной точки.

<b>Y2</b>	Y-координата второй указанной точки.
<b>Color</b>	Цвет, используемый для рисования элемента, это может быть черный или белый цвет на черно-белых страницах или один из 256 цветов на цветных страницах.

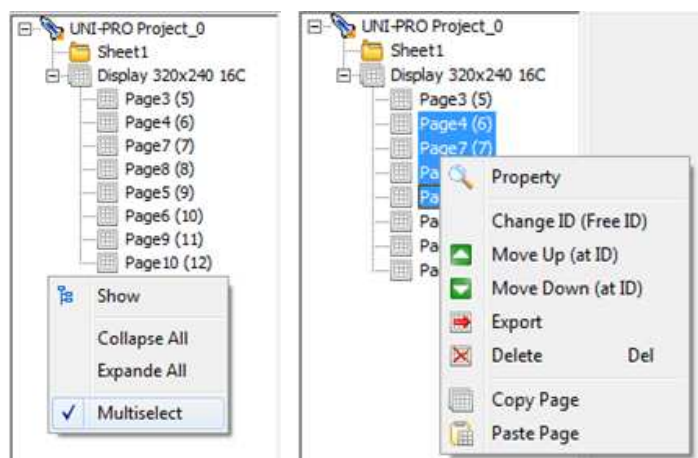
### Rectangle (Прямоугольник)

Прямоугольник обладает следующими свойствами:

<b>Name</b>	Уникальное имя, которое идентифицирует элемент внутри проекта.
<b>Left</b>	Смещение относительно левого края страницы, в пикселях.
<b>Top</b>	Смещение относительно верхнего края страницы, в пикселях.
<b>Right</b>	Смещение относительно правого края страницы, в пикселях.
<b>Bottom</b>	Смещение относительно нижнего края страницы, в пикселях.
<b>Color</b>	Цвет, используемый для рисования элемента: это может быть черный или белый цвет на черно-белых страницах или один из 256 цветов на цветных страницах.
<b>Filled</b>	Указывает, должен ли прямоугольник быть заполненным каким-либо определенным цветом.

### 3.5.9 Множественный выбор из дерева страниц

Из дерева страниц, нажав правую кнопку, вы можете активировать режим Multiselect из появившегося выпадающего меню.



Этот режим позволяет вам выбирать несколько страниц (с помощью клавиш Shift и Ctrl) и выполнять некоторые специальные операции, в том числе:

**Переместить вверх/вниз (по идентификатору):** перемещает выбранные страницы вверх/вниз и последовательно размещает их на указанном идентификаторе, если указанный идентификатор не существует, перемещение не происходит.

**Change ID (Free):** изменяет ID выбранных страниц, начиная с указанного, занимает только свободные ID, при перекрытии ID перемещение не может быть осуществлено.

**Экспорт:** экспорт выбранных страниц.

**Удалить:** удалить выбранные страницы.



## 4 РАСШИРЕННЫЕ ДЕЙСТВИЯ

### 4.1 Команды

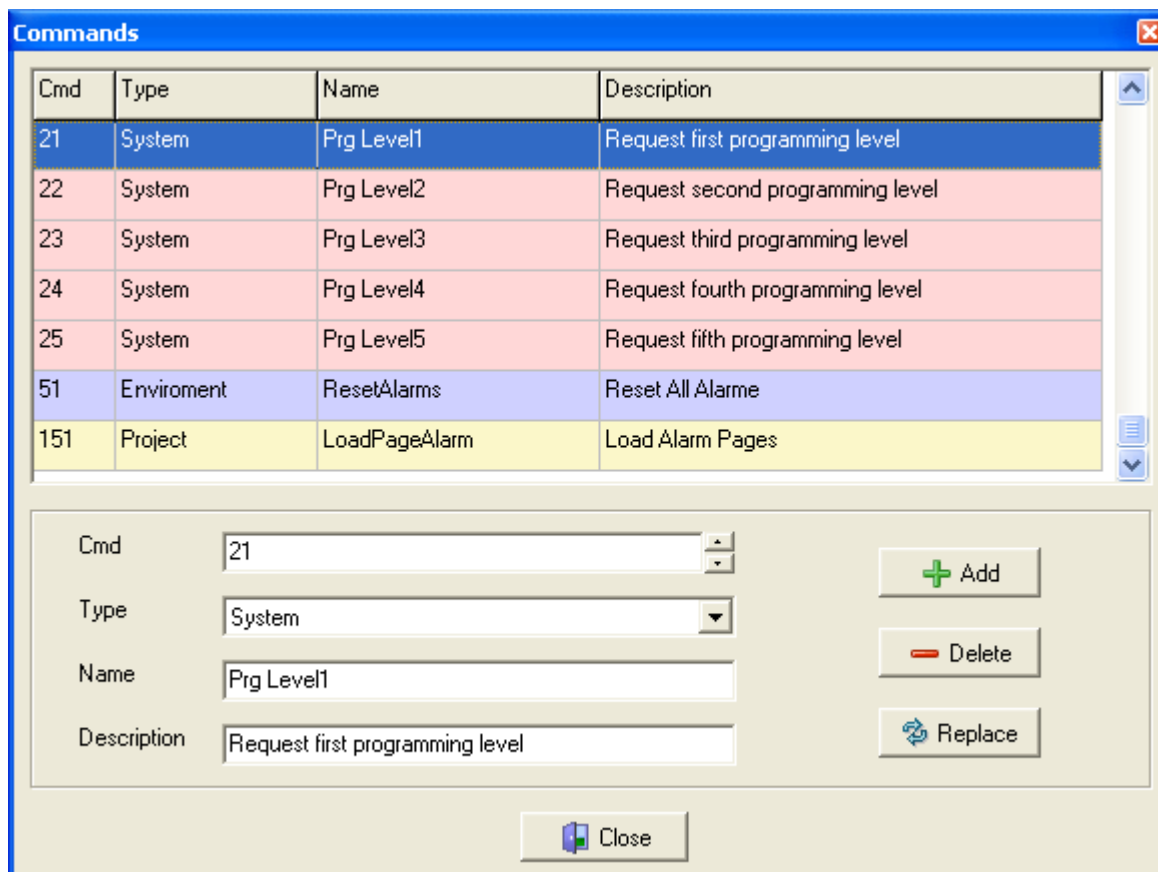
Команды являются эффективным средством разработки для создания приложений, способных управлять такими процессами, как запрашивание новой страницы, перенастройка архива событий, установка значений по умолчанию для параметров и информирование другого устройства сети об определенном состоянии.

Управление командами в папке осуществляется с помощью сущностей **CommandIn** и **CommandOut** или с помощью присваивания команде отличного от *NONE* в свойстве *Command* значения определенных элементов EMIL страницы (Текстов, Значков).

Команды разделены на три категории: команды **системы, среды и проекта**.

Системные команды имеют уникальное значение, определяемое системой, и используются для всех программ (предупреждение: некоторые программы могут отвечать не совсем корректно на некоторые предопределённые команды, потому что не имеют соответствующей функции; например, команда *Erase Historian* не будет работать для устройств, не имеющих функцию регистрации событий). В случае команд среды и проекта их значения могут быть присвоены на усмотрение разработчика. Разница между этими командами заключается в том, что команды среды могут быть использованы во всех проектах, а команды проекта задаются проектом.

Выберите *Tools/Commands...* в меню, чтобы открыть окно ниже, которое показывает доступные команды системы и позволяет добавить или изменить команды среды и проекта.



Условно приняты следующие значения: от 1 до 50 для системных команд (розовые), от 51 до 150 для команд среды (фиолетовые) и от 151 до 250 для команд проекта (желтые).

Структура команды состоит из трех полей:

**COMMAND:** показывает числовое значение команды. Имя связано с каждой командой, которая является начальной установкой для системных команд, действительной для команд среды во всех проектах (от 51 до 150) и для команд проекта, заданных проектом (от 151 до 250).

**PARAMETER:** 16-битное значение, связанное с командой. В зависимости от команды может просматриваться как одно поле или как сочетание нескольких полей.

**NODE:** значение элемента сети, отправляющего (или принимающего) команду. Это значение присваивается с использованием соответствующего окна соединения.

Чтобы отправить команду из EIML-страницы, направьте курсор на элемент (текст или значок), который имеет отличное от нуля значение свойства *Command*, и нажмите *ENTER*.

Встроенная в папку команда активируется, когда вход триггера (*trigger input*) переключится с нуля на единицу.

Далее представлен список всех доступных команд системы и их значения:

**Load Page:** Команда запроса/отправки страницы. Соответствующий *параметр* указывает индекс запрошенной страницы.

**Mute Buzzer:** Команда отключения звукового сигнала платы, установленного *узлом*. Если поле узла установлено для пересылки сигналов, эта команда отключит все элементы в сети.

**Sync Clock:** Команда синхронизирует часы узла назначения с часами узла источника.

**Save Par DRV:** Команда для создания резервной копии всех параметров системы (предупреждение: данная операция может занять определенное время).

**Save Par App:** Команда для создания резервной копии всех параметров приложения (предупреждение: данная операция может занять определенное время).

**Restore Par DRV:** Команда восстановления всех параметров системы из резервной копии (предупреждение: данная операция может занять определенное время).

**Restore Par App:** Команда восстановления всех параметров приложения из резервной копии (предупреждение: данная операция может занять определенное время).

**Load Default Par:** Команда загрузки стандартных значений всех параметров, заданных во время фазы разработки (предупреждение: данная операция может занять определенное время).

**Erase Historian:** Команда, которая стирает из памяти архив событий (предупреждение: все данные будут удалены).

**Send Page:** Команда для отправки из алгоритма конкретной страницы на дисплей контроллера, принадлежащего сети CAN.

**Toggle ON/OFF:** Команда ВКЛ./ВЫКЛ. прибора. Управление этой командой должно осуществляться проектом.

**Prg Level 1-5:** Данная команда запрашивает отправку страницы с наименьшим индексом относительно уровня безопасности 1-5 (подключение этого свойства осуществляется параметром *EnablePrgLevel*).

**Request Alarm:** команда запроса первого сигнала тревоги.

**Request Historian:** команда запроса первого элемента архива событий.

**Key Save Par Drv:** сохраняет параметры конфигурации оборудования на периферийном USB-устройстве.

**Key Save Par App:** сохраняет параметры конфигурации приложения на периферийном USB-устройстве.

**Key Restore Par Drv:** загружает параметры конфигурации оборудования на периферийное USB-устройство.

**Key Restore Par App:** загружает параметры конфигурации приложения на периферийное USB-устройство.

**Send EVCM Commands:** отправляет команды на EVCM (\*).

**Send EVCM Manual Pos:** отправляет заданную вручную контрольную точку положения EVCM (\*).

**Receive EVCM Current Pos:** получает текущую позицию EVCM (\*).

**Receive EVCM Status:** получает статус EVCM (\*).

**Receive EVCM Alarms:** получает статус сигналов тревоги EVCM (\*).

(\*). см. руководство по EVDRIVE.

## 4.2 Перемещение по страницам

Важным аспектом для рассмотрения является вопрос выбора страницы из другой страницы и взаимосвязь страниц. Ранее было отмечено, что страницы хранятся в контроллере и их создание совпадает с разработкой алгоритмов управления. В древе проекта страницы отображаются после папок, в одной или нескольких группах, в зависимости от количества дисплеев, которое имеет контроллер.

Эта особенность много-дисплейного управления позволяет отображать страницы, созданные для определенных пользовательских интерфейсов с более мощным браузером (например, Вы можете отображать страницы, написанные для буквенно-цифрового дисплея 20x4 на графическом дисплее 240x128!).

Приведем пример относительно указанного выше древа страниц: страницы с ID=3 и ID=4 могут являться буквенно-цифровыми страницами, также разработанными для отображения на графическом дисплее, что значительно упрощает код. Когда дисплей 240x128 запрашивает страницу с ID=3 и если первый поиск в списке графических страниц результата не дает, то будет осуществлен поиск в списке буквенно-цифровых страниц, и найденная в нем страница будет отображена с соответственным изменением размера.

В основном, перемещение между страницами возможно благодаря механизму, который использует команду **Load Page** (см. параграф *Команды*). Например, если Вы установили команду и свойства параметра текста следующим образом: *Command=Load Page, Param= 3*, - то будет запрошена страница с ID=3. В особых случаях, если Вам нужно отправить запрос на открытие страницы сигнала тревоги нажатием определенной кнопки, можно отправить страницу на панель пользовательского интерфейса, используя выходы управления под влиянием действия “нажатие кнопки”. В этом случае, в дополнение к ID страницы, необходимо определить требуемый тип браузера для входа параметра в соответствии со следующей таблицей:

Display 120x32	7-сегментный 4-цифровой дисплей	= 1
Page5 (1)	4x20 буквенно-цифровой дисплей	= 3
Display 128x64	128x64 графический дисплей	= 4
Page4 (1)	240x140 графический дисплей	= 6
Display 20x4	120x32 графический дисплей	= 7
Page6 (1)	7-сегментный 2x4-цифровой дисплей	= 8
Display 240x140	320x240 цветной графический дисплей	= 9
Page3 (1)	320x240 16-ти цветный графический дисплей	= 10
Display 2x4 Digits		
Page7 (1)		
Display 320x240		
Page1 (1)		
Display 320x240 16C		
Page2 (1)		
Display 4 Digits		
Page8 (1)		

Задавая свойства *PreviousPage*, *NextPage* и *TimeoutPage*, можно установить перемещение между страницами. Присваивая им значения, отличные от <NONE>, можно установить, какие страницы загружаются при соответствующем нажатии кнопок LEFT, RIGHT или ESC.

В дополнение, задавая *Timeout* и *TimeoutPage*, Вы можете установить период бездействия до загрузки страницы.

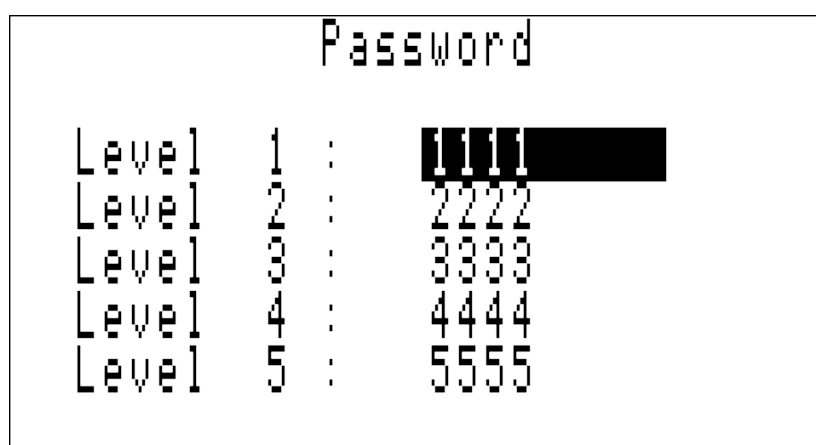


### 4.3 Информация, защищенная паролем

Во многих приложениях доступ посторонних пользователей к определенным данным ограничивается, чтобы обеспечить высокую безопасность уровня. Неверное изменение установки или рабочих параметров контроллера может привести к неисправности или прекращению работы целой системы.

Утверждение системы защиты базируется на 5 уровнях, которые могут быть присвоены каждой EIML-странице с помощью установки свойства *Level*. Каждый уровень связан с различным паролем в контроллере. Когда дисплей запрашивает страницу, характеристика *Level* которой задана отличным от нуля значением, контроллер проверяет, является ли запрос, сопровождаемый авторизацией уровня, равным или выше уровня страницы. В противном случае пользователю придется ввести пароль. Если введенный пароль относится к одному из 5 имеющихся уровней контроллера, уровень авторизации для дисплея будет установлен как уровень пароля.

Например, предположим, что контроллер имеет следующую страницу установки:



Если дисплей, начинающийся с уровня авторизации 0, запрашивает страницу с *Level*=2, контроллер попросит пользователя ввести пароль.



Если будет введен пароль 1111, контроллер запросит пользователя ввести пароль заново, т.к. новый уровень (1) все ещё ниже, чем требуемый (2). Если будет введен пароль 3333, запрашиваемая страница будет отправлена пользователю, который будет иметь доступ ко всем страницам, имеющим уровень, равный или ниже третьего.

Если ни одна из кнопок на кнопочной панели не была нажата за период времени начальной установки (стандартное значение системного параметра – 60 сек.), уровень авторизации будет сброшен.

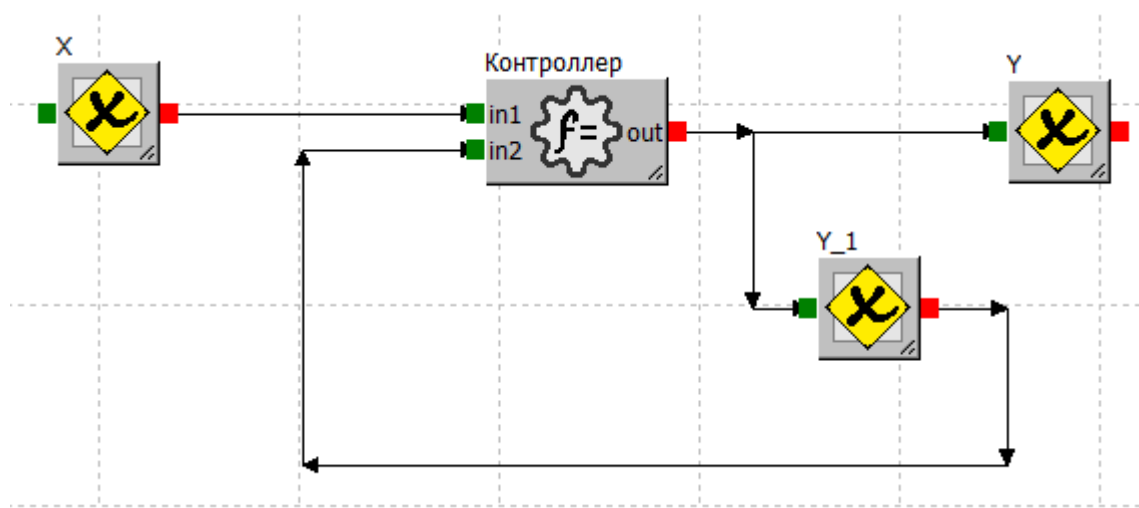
#### 4.4 Установка задач выполнения

Все операции расчета, установленные с помощью соединений сущностей, могут быть выполнены в трех различных задачах. Как правило, большая часть вычислений выполняется в Главной Задаче. Если необходимо вычислить некоторые критические операции, то они могут быть выполнены в одной из первоочередных задач Timed100ms, которые выполняются с прерыванием каждые 100 мс.

Замечаем, что последние две задачи должны быть использованы только в случае крайней необходимости, и их перегрузка может привести к ошибкам выполнения.

#### 4.5 Задание последовательности выполнения программы

Программа, графический интерфейс которой создан с помощью соединения сущностей в папках, следует определенным правилам предварительной установки для определения последовательности выполнения запросов различных алгоритмов. Такая последовательность зависит, главным образом, от маршрута от выходов к входам и порядка, в котором они были добавлены в папки, который автоматически присваивается генератором кода. Довольно часто, однако, выполнение определенных частей программы требует принудительного перехода к предварительно установленной последовательности. Соединения обратной связи, как те, что проиллюстрированы ниже, являются хорошим примером.



В данном случае мы хотим, чтобы выход  $Y$ , вычисляемый алгоритмом, использовал вход  $Y_1$  (который представляет предыдущий выход) до того, как вход был изменен! Чтобы установить определенный порядок выполнения этой части программы, необходимо установить правильную последовательность значений свойства *Order* для всех участвующих переменных. В приведенном выше примере, правильное действие будет обеспечено, если свойство *Order* для  $Y$  выше (или равно)  $X$ , но ниже  $Y_1$ . В связи с этим возможно следующее сочетание:

Order  $X$  = 0

Order  $Y$  = 1

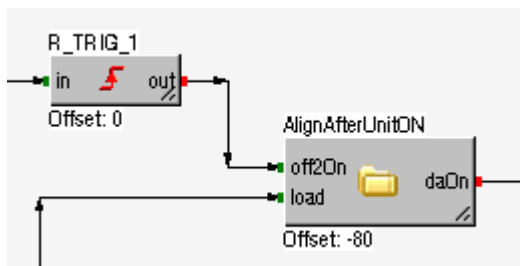
Order  $Y_1$  = 2

Это означает, что при выполнении программы переменная  $X$  будет вычислена в первую очередь, затем  $Y$  и  $Y_1$ .

Воздействие на порядок выполнения программы имеют все сущности со свойством *Order*, т.е.: Alg., Var, Par, Pers, DigitalOut, AnalogOut, LED, Buzzer и CommandOut.

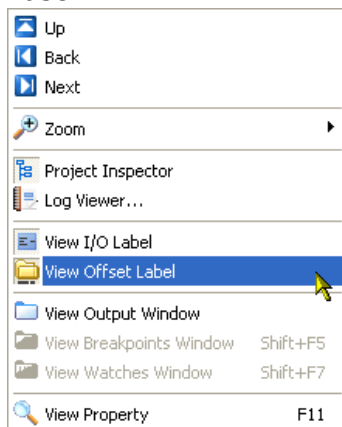
При работе со свойством дочерней папки *Offset* можно изменить последовательность выполнения для всех содержащихся в ней сущностей.

Свойства OFFSET для дочерних папок и библиотек можно просмотреть.



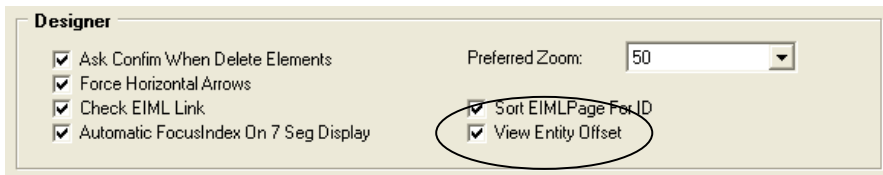
Для активации этой функции перейдите:

1) Menu View->View Offset Label



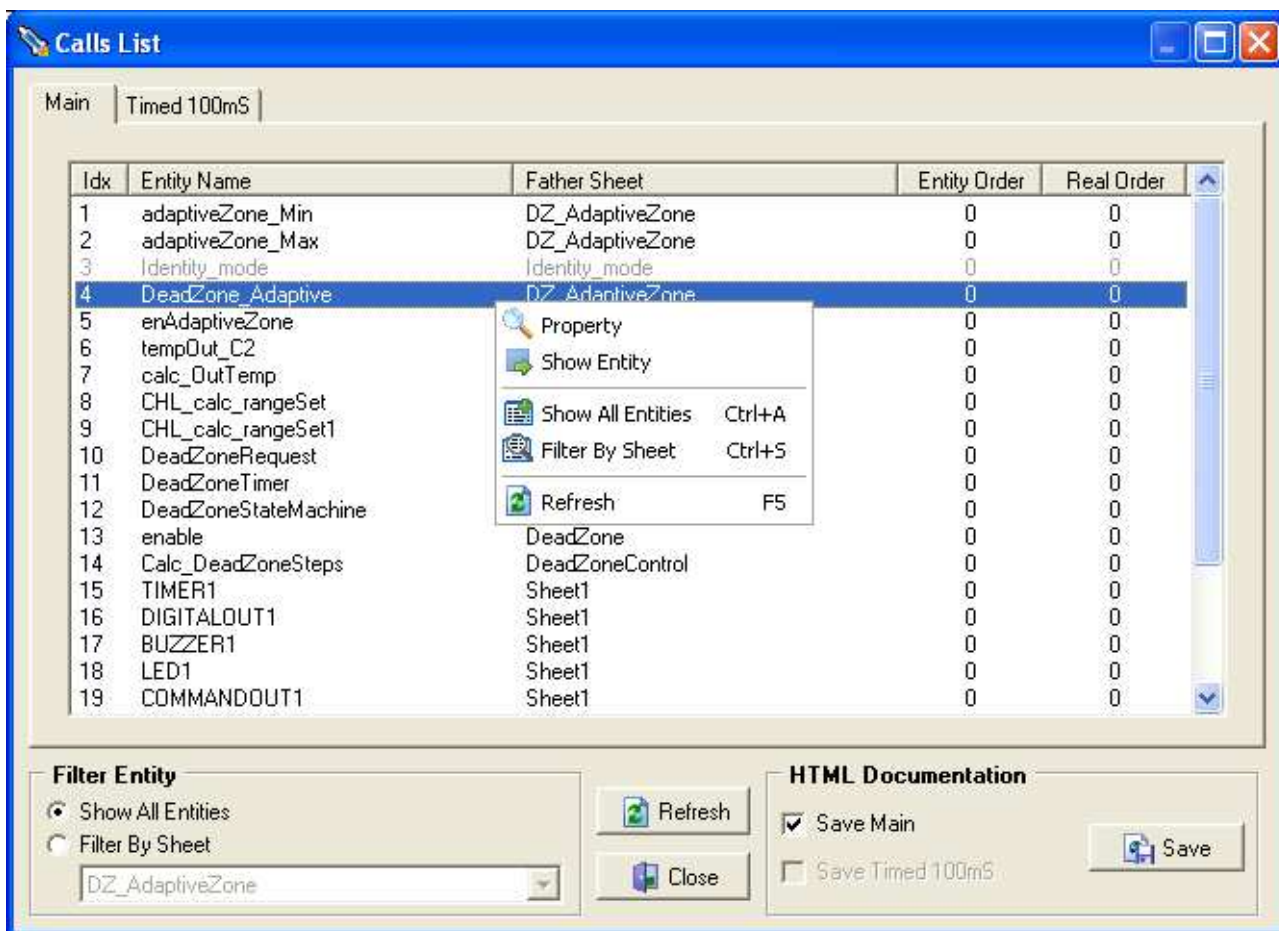
Или

2) Из Settings



### 4.5.1 Список запросов

Чтобы узнать, какая часть программы выполняется, можно обратиться к соответствующему упорядоченному списку через пункт меню *Tools / Calls List*. Пример – на картинке ниже:

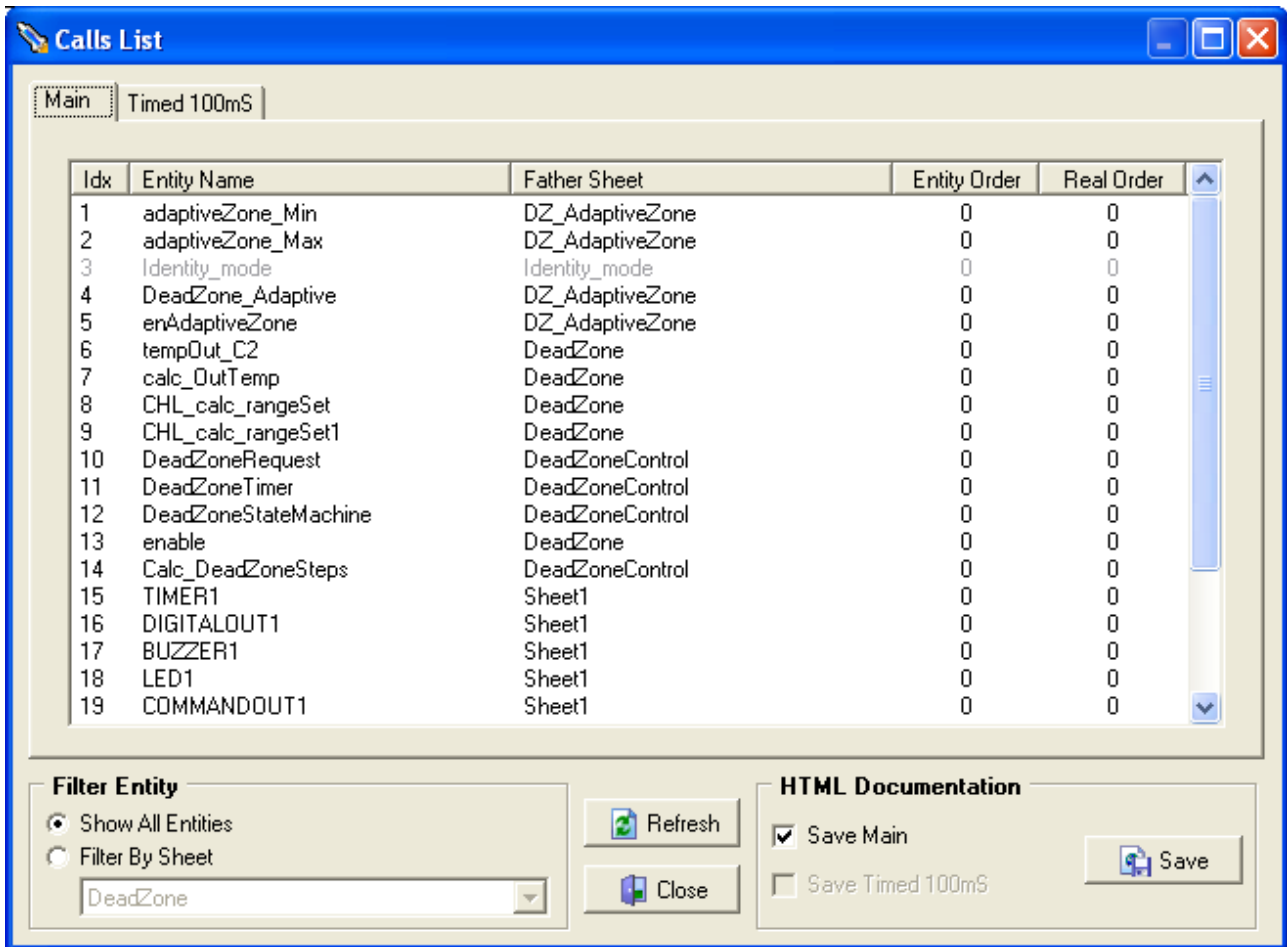


Запросы подразделены на 3 секции (одна для каждой задачи) и отображены в последовательности, в которой они будут выполняться. При двойном нажатии на любую из этих строк активируется поиск сущности, которая и будет отображена впоследствии.

С помощью контекстного меню (правый щелчок по выбранному элементу) можно просмотреть и изменить выделенные свойства сущности. Изменяя свойство *Order*, можно менять порядок выполнения задач и пересчитывать «на лету» новый порядок выполнения с помощью пункта *Refresh* контекстного меню.

Выбором пункта *Filter By Sheet* и соответствующей папки можно просмотреть сущности конкретной папки для более точного отображения элементов, которыми Вы хотите управлять.

Выбора нужной папки можно осуществить с помощью элемента *combo*, щелкнув по соответствующей строке таблицы.

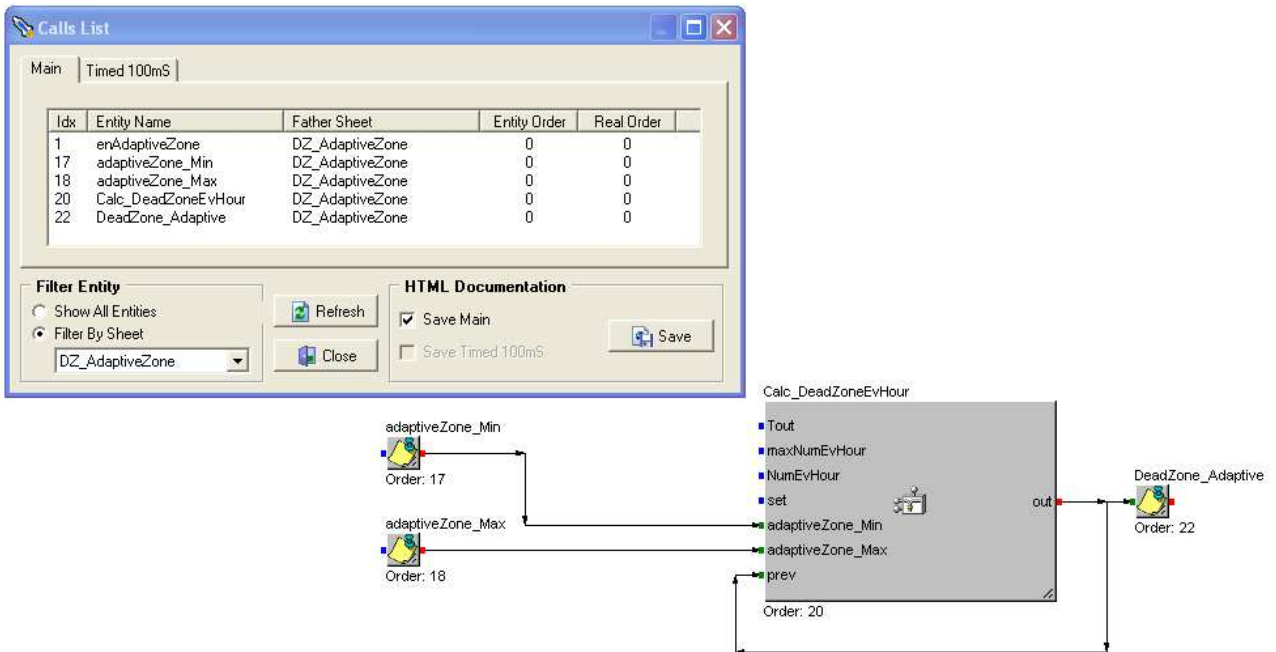


Также возможен возврат к обычному виду; для этого выберите *Show All Entities* и нажмите кнопку *Refresh* или выберите пункт *Show All Entities* контекстного меню.

С помощью кнопки *Save* можно сохранить выбранную таблицу как документ HTML.

#### 4.5.2 Порядок выполнения при активном списке запросов

При активном списке запросов можно просмотреть порядок выполнения под вычисляемой сущностью; такое возможно, только если и когда он используется. Полезно правильно понимать главный цикл программы.

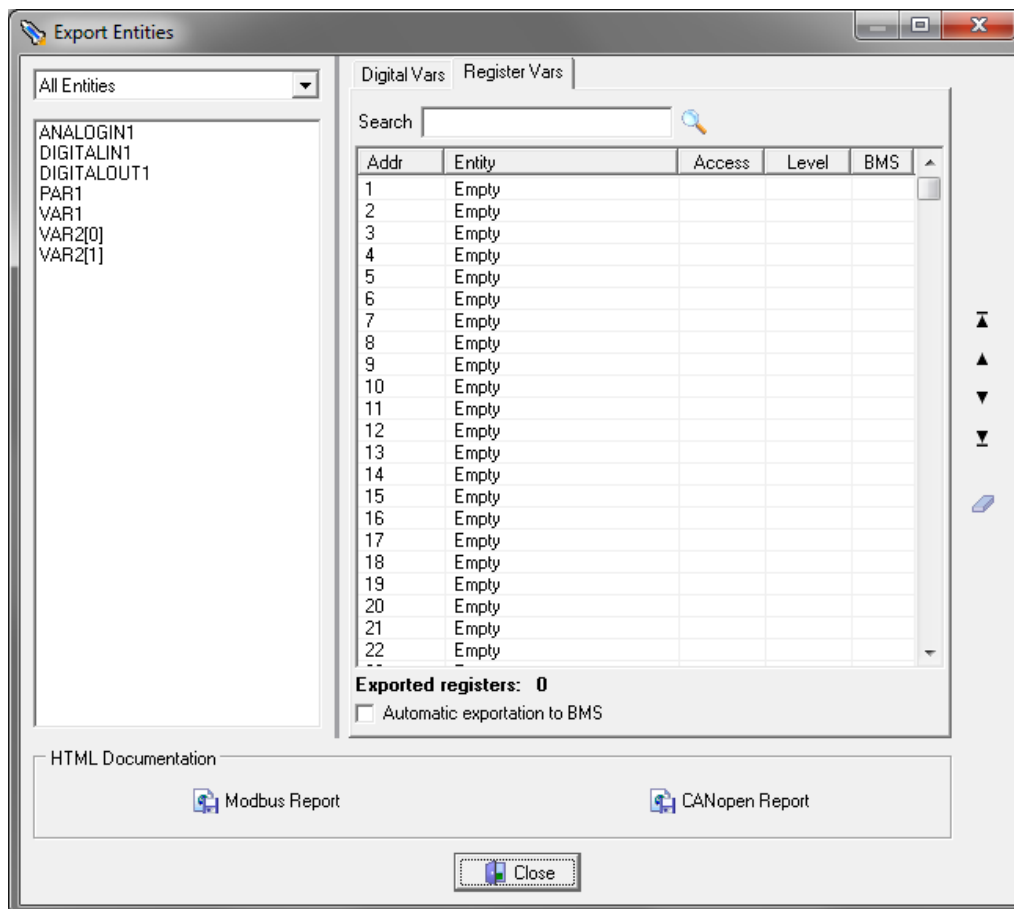


В данном случае понятно, что переменная вычисляется сразу после алгоритма.

## 4.6 Экспорт внутренних переменных

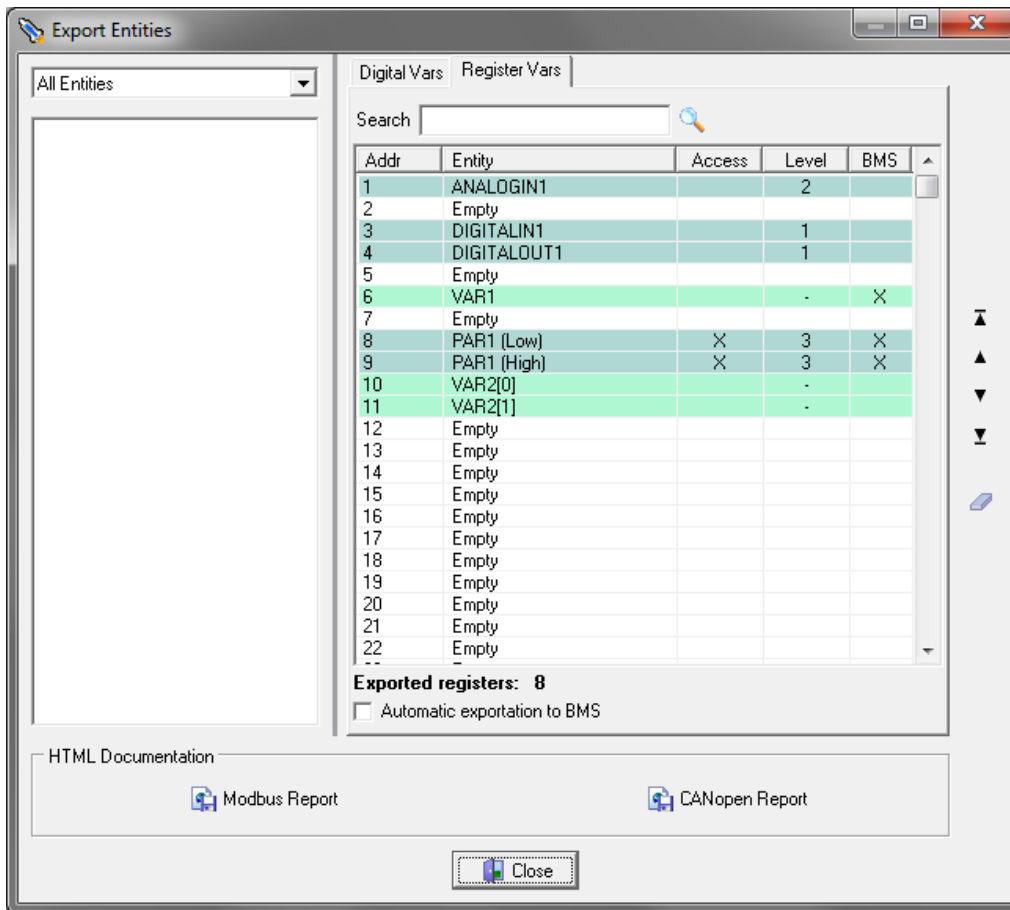
Чтобы получить доступ к внутренним переменным контроллера через один из доступных протоколов связи, вам необходимо определить, какие переменные вы хотите «экспортировать» (т. е. сделать видимыми снаружи) и отсортировать их по критерию, способному поддерживать тот же порядок сортировки даже в том случае, если впоследствии в программу будут внесены изменения.

Чтобы решить, какие переменные экспортировать, и определить порядок их сортировки, вы можете использовать следующее окно (выберите *Инструменты/Экспорт переменных...* из меню, чтобы открыть его):



Слева находится список всех переменных проекта, а справа два дополнительных списка, которые можно выбрать, щелкнув соответствующие вкладки: *Битовые Vars* и *Регистровые Vars*. Чтобы упростить поиск, используйте комбинацию в верхнем левом углу окна, чтобы отфильтровать список по типу.

Чтобы добавить переменную в список, выберите ее в списке слева и с помощью мыши перетащите ее в нужное место в списке справа.

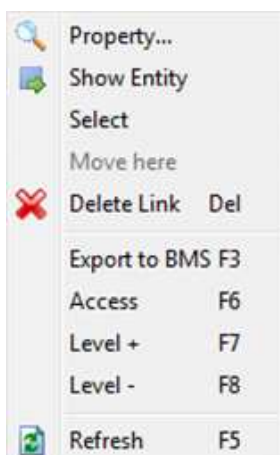


В результате этой операции имя переменной исчезнет из списка слева и появится в списке справа, на позиции, указанной в столбце ID. Используя этот метод перетаскивания, вы можете переместить элементы в список экспортируемых переменных или вернуть их в список переменных, которые еще не были экспортированы.

Столбец *BMS* указывает, используется ли экспортируемый объект (X) или нет в протоколе BMS (WebServer). С помощью клавиши F3 вы можете выбрать/отменить выбор этого свойства. Столбец *Access* указывает, защищен ли экспортируемый объект при чтении (X) или нет в протоколе Modbus.

С помощью клавиши F6 можно выбрать/снять выбор этого свойства. В столбце *Уровень* указан уровень защиты от записи объекта, экспортируемого по Modbus, при отсутствии защиты (уровень 0) отображается «-». Через клавиши F6 и F7 можно повышать/понижать уровень защиты.

В качестве альтернативы можно использовать контекстное меню.



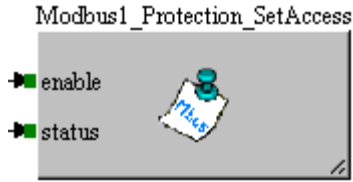
С помощью контекстного меню также можно выделить и переместить блок сущностей с одной позиции на другую, не производя повторный экспорт. После того, как объекты выбраны, выберите «Выбрать» в контекстном меню, чтобы открыть его снова с новой позиции, затем «Переместить сюда».

Таким образом, сущность будет помещена в новое положение.



Для управления в проекте защитой R&W Modbus возможно использование некоторых системных библиотек или вызовов других системных функций изнутри алгоритмов:

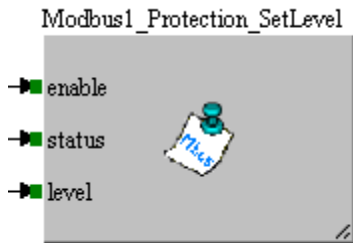
- Чтобы включить доступ к объекту, защищенному на чтение (X), можно использовать библиотеку Modbus1\_Protection\_SetAccess со входом статуса = 1 и входом разрешения = 1.



- Или вызов функции **CJ\_Modbus\_SetAccess(1);**

При установке в 1 входа разрешения со статусом = 0 или вызове функции **CJ\_Modbus\_SetAccess(1)** защита чтения восстанавливается.

- Чтобы разрешить доступ объекту, защищенному при написании, например, уровнем 2, можно использовать библиотеку Modbus1\_Protection\_SetLevel со входом статуса = 1, со входом уровня = 2 и установкой 1 на вход разрешения



- Или вызов функции **CJ\_Modbus\_SetProtections (1, 2);**

При установке 1 на вход разрешения, со статусом = 0 и уровнем = 2 или вызовом функции **CJ\_Modbus\_SetAccess(0,2)** восстанавливается прописанная защита для этого уровня.

- Защита Modbus, как при чтении, так и при записи, может быть автоматически восстановлена после истечения тайм-аута после истечения последнего полученного сообщения Modbus, используя соответствующую библиотеку, чтобы установить этот тайм-аут:



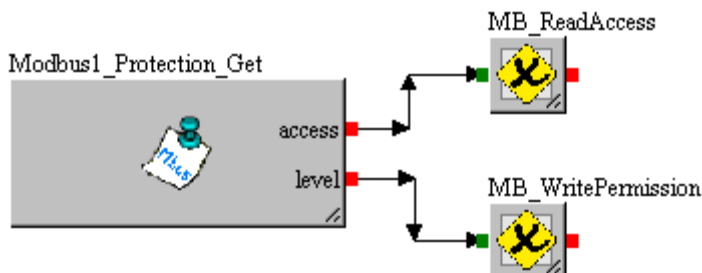
- Или вызов функции **CJ\_Modbus\_SetTimeOutProtections(value);**

Значение, равное 0 на входе библиотеки, или вызов функции

**CJ\_Modbus\_SetTimeOutProtections(0)** вызывает неограниченную разблокировку.

Для включения защиты необходимо использовать эту библиотеку в проекте, введя в качестве входных данных значение, отличное от 0.

- Чтобы в любое время узнать состояние защиты как для чтения, так и для защиты от записи, можно использовать библиотеку



-

- Или вызвав функцию **CJ\_BYTE CJ\_Modbus\_GetStatusProtections()**, в младшей части байта можно прочитать текущий уровень разрешений на запись, а в бите 4 статус доступа на чтение, например:  
`level = CJ_Modbus_GetStatusProtections() & 0x0F;`  
`access = CJ_Modbus_GetStatusProtections() & 0x10 > 0;`

В зависимости от списка можно экспортировать только определенные типы переменных, как показано в таблицах ниже:

Битовые переменные	
Сущность	Тип данных
Digital In	
Digital Out	
Par	только CJ_BIT
Pers	только CJ_BIT
Var	только CJ_BIT
Fixs	только CJ_BIT

Регистровые переменные	
Сущность	Тип данных
Analog In	CJ_ANALOG: только значение
Analog Out	
Clock	
Timer	
Par	Все
Pers	Все
Var	Все
Fixs	Все

ПРИМЕЧАНИЕ. 32-битные типы данных (CJ\_LONG, CJ\_DWORD, CJ\_DATETIME, CJ\_DATE и CJ\_TIME) занимают две строки в списке переменных регистров: младшие 16 бит находятся в первой строке, а старшие 16 бит — во второй.

Если в Настройках выбран параметр «Экспорт упакованного CJ\_CHAR», типы CJ\_CHAR экспортируются в Modbus группами (состоящими из двух элементов) для каждого регистра.

#### 4.6.1 Документация Modbus и CANopen

Существует возможность отобразить экспорт в HTML-документе. Этот документ будет содержать всю полезную информацию об экспортированных сущностях как для протоколов Modbus, так и для CANopen.

Чтобы подключить данное свойство, просто нажмите на кнопки *Create Modbus Documentation* или *Create CANopen Documentation*, расположенные в нижней части окна *Export Entities*.

Документ CANopen содержит информацию для доступа к переменным проекта с помощью индекса и субиндекса *Object Dictionary*.

Документ Modbus содержит информацию для доступа к переменным проекта с помощью Modbus-команд считывания/записи *coils* и *registers*.

## 4.7 Условная видимость

Условная видимость позволяет скрыть параметры конфигурации и статус в соответствии с определёнными настройками. Некоторые сущности могут быть видны только при определённой конфигурации, в этом случае количество параметров и состояний, отображаемых в пользовательском интерфейсе, будет меньше, чем их общее количество. Другими словами, все не относящиеся к делу параметры не будут показаны в пользовательском интерфейсе. Для пользователя становится проще просматривать пользовательский интерфейс, ускоряется поиск и установка указанных сущностей. Польза заключается ещё и в том, что условная видимость может скрыть некоторые параметры и состояния, которые «опасны» для приложения в том смысле, что неверные настройки от постороннего пользователя могут привести к ошибкам выполнения.

Условная видимость может использоваться для переменных, параметров и постоянных ссылок на EIML-страницы в виде переменных, комбо и таблиц. Для использования данной функциональности выполните следующие шаги:

1. Выберите свойство *condvisible* для всех требуемых сущностей.
2. Выполните экспорт этих сущностей по протоколу Modbus с помощью инструмента *Export Entities*.
3. Задайте алгоритмы, управляющие отображением нужных сущностей. Используйте функцию встроенного ПО **void CJ\_SetCondVisBit(word idx, bool value)**.
4. Выберите свойство *CondVis* на каждой странице, где присутствуют переменные с условной видимостью.

Все этапы должны быть корректно выполнены, в противном случае условная видимость сущностей не будет работать.

На буквенно-цифровом дисплее 4x20 и графическом дисплее 240x128 все сущности, скрытые вследствие условной видимости, будут заменены многоточиями "...", и их нельзя будет изменить.

На 7-сегментном дисплее скрытые сущности не будут отображаться.

### 4.7.1 Функция встроенного ПО void CJ\_SetCondVisBit(word idx, bool value)

Данная функция позволяет управлять напрямую из алгоритма видимостью экспортируемой переменной с адресом "*idx*" в протоколе Modbus. Функция использует два параметра:

- *word idx*: Modbus-адрес сущности, экспортируемой по протоколу Modbus (инструмент *Export Entities*);
- *bool value*: видимость сущности. Данный параметр принимает булевы значения:  
Если *value* = 1 (или условие выполнено), то сущность будет скрыта.  
Если *value* = 0 (или условие не выполнено), то сущность будет показана.

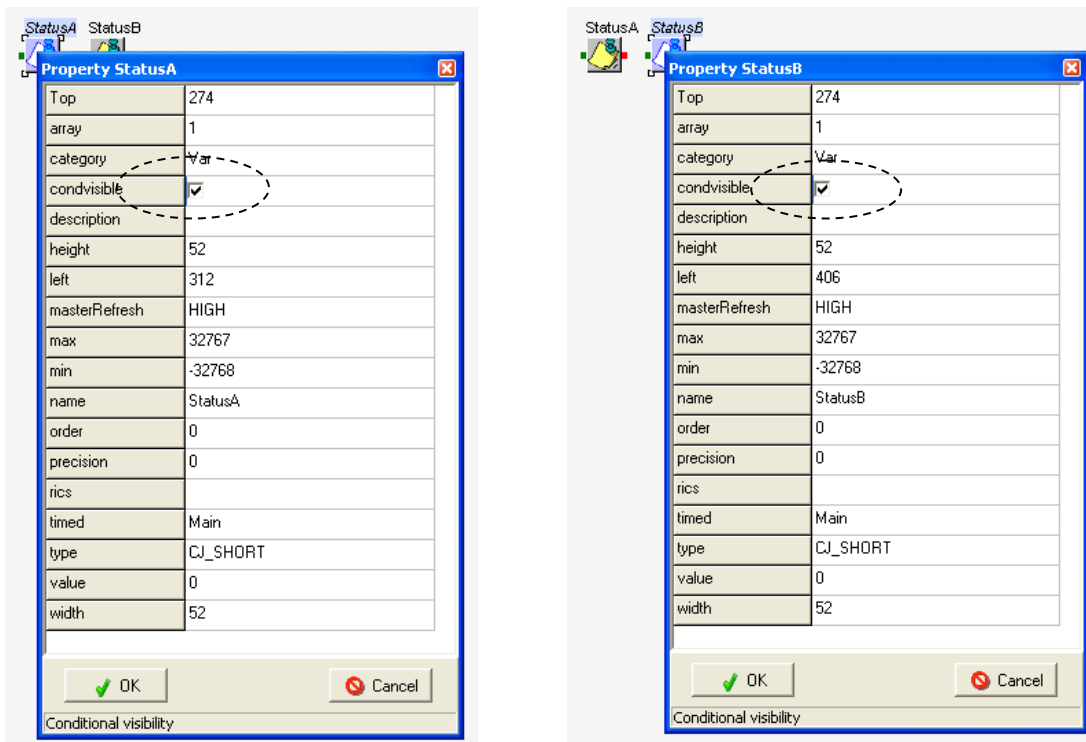
Вызов функции в виде **CJ\_SetCondVisBit(idx, 1)** означает, что сущность с адресом *idx* всегда будет **скрыта**.

Вызов функции в виде **CJ\_SetCondVisBit(idx, 0)** означает, что сущность с адресом *idx* всегда будет **видима**.

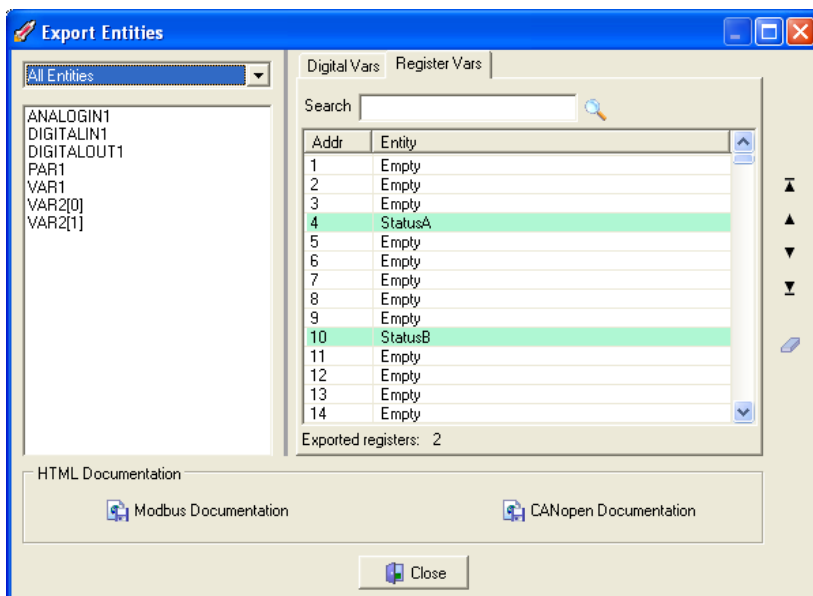
### 4.7.2 Пример

С помощью простого примера продемонстрируем этапы корректной установки условной видимости. Предположим, мы хотим задать условие видимости двух переменных *StatusA* и *StatusB* с помощью параметра P001.

На первом этапе требуется выбрать свойство *condvisible* обеих сущностей.



Как сказано выше, обе переменные должны быть экспортированы в таблицу Modbus.

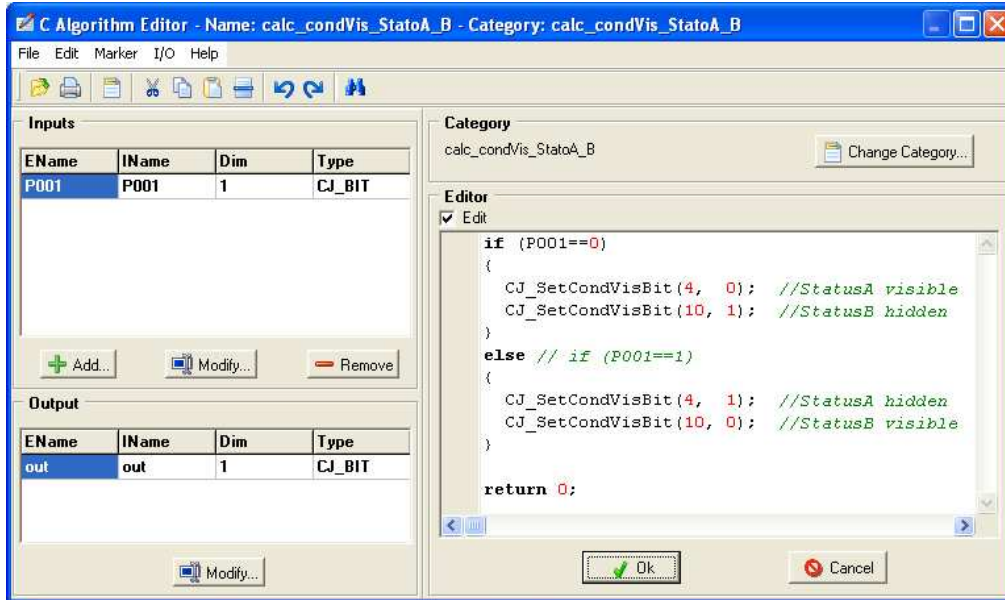


С помощью инструмента *Export Entities* выполните экспорт переменной *StatusA* на адрес 4 и *StatusB* на адрес 10.

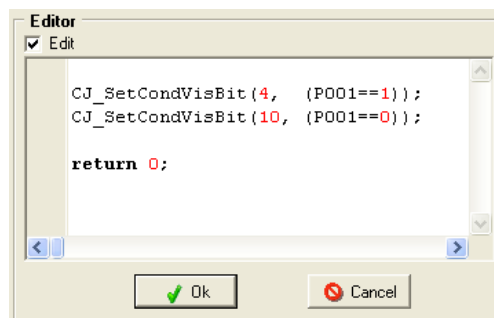
Теперь можно записать алгоритм, управляющий отображением переменных. По примеру мы хотим, чтобы:

- $P001=0$ : *StatusA* видима (со значением ON), а *StatusB* скрыта;
- $P001=1$ : *StatusA* скрыта, а *StatusB* видима (со значением ON).

Алгоритм, управляющий условной видимостью, выглядит следующим образом:



Тот же результат мы получим, если перепишем алгоритм как условие для параметра *value* функции:



Эффект на пользовательский интерфейс дисплея 4x20 (V-View) или дисплея 240x128 (V-Graph) выглядит следующим образом:

**P001=0:**

```
P001=0  StatusA  ON
        StatusB  ...
```

**P001=1:**

```
P001=1  StatusA  ...
        StatusB  ON
```

Эффект на пользовательский интерфейс 7-сегментного дисплея будет следующим: строка с переменной *StatusA* или *StatusB* не будет отображена в таблице:

**Без условной видимости:**

StatusX	1
StatusA	ON
StatusB	ON

**P001=0:**

StatusX	1
StatusA	ON

**P001=1:**

StatusX	1
StatusB	ON

## 4.8 Семафоры

С помощью функций *CJ\_WORD CJ\_SemaphoreRead(CJ\_WORD i)* и *CJ\_BYTE CJ\_SemaphoreWrite(CJ\_WORD i, CJ\_WORD value)* можно использовать 10 семафоров для регулирования доступа к ресурсу в ходе выполнения программы.

Функция *CJ\_WORD CJ\_SemaphoreRead(CJ\_WORD i)* позволяет считывать значение, установленное n-ым семафором.

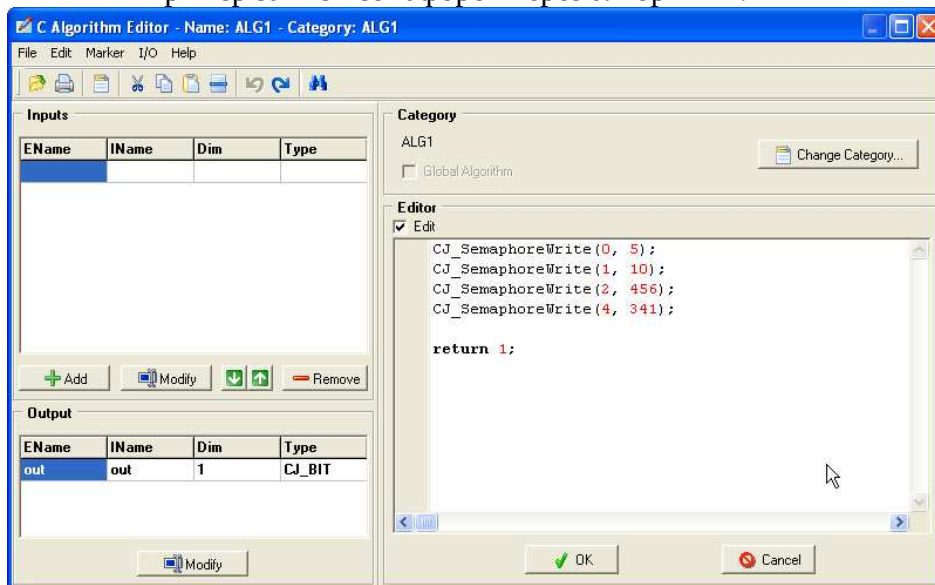
Функция *CJ\_WORD CJ\_SemaphoreWrite(CJ\_WORD i, CJ\_WORD value)* позволяет записывать значение *value* в n-ый семафор. Запись семафора разрешена при выполнении одного из условий ниже:

1. Семафор принимает значение «истина» или содержит значение 0.
2. Значение *value* для записи – значение 0.

Семафор может быть записан повторно до тех пор, пока остаётся свободен, либо для записи 0.

Другое использование семафоров – при отладке, либо для контролирования значения конкретной переменной в алгоритмах; во время отладки приложения можно отображать список 10 семафоров с помощью пункта меню *View/reView Semaphores Window*.

Пример записи семафоров через алгоритмы.

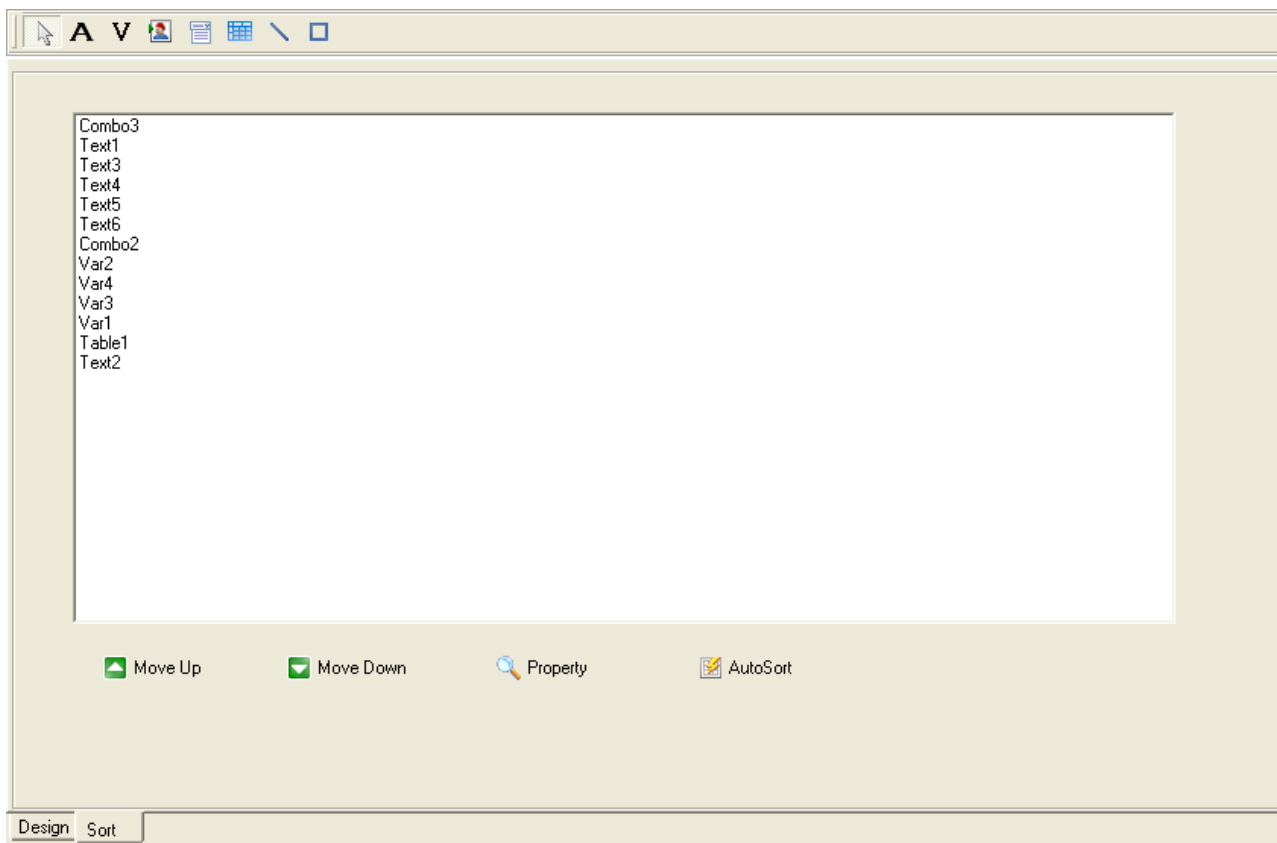


Окно со списком семафоров в течение отладки.

Semaphore	Value
Semaphore_0	5
Semaphore_1	10
Semaphore_2	456
Semaphore_3	0
Semaphore_4	341
Semaphore_5	0
Semaphore_6	0
Semaphore_7	0
Semaphore_8	0
Semaphore_9	0

## 4.9 Упорядочивание EIML элементов


Элементы, добавленные на EIML-страницу (тексты, переменные, значки, combo, таблицы, линии и прямоугольники), кодируются внутри страницы в той последовательности, в которой они были добавлены разработчиком при создании страницы. Когда страница загружается с помощью панели пользовательского интерфейса, элементы страницы будут появляться в той последовательности, в которой они были введены, что может не производить желаемого оптического эффекта. Чтобы настроить упорядочивание различных графических элементов и, следовательно, последовательность, в которой они будут появляться на странице, нажмите вкладку *Sort* и откройте следующее окно:



В этом окне перечислены имена различных графических объектов, встроенных в страницу, в порядке их воспроизведения, когда страница отображается в пользовательском интерфейсе. Выделите один или более объектов и перетащите их с помощью мыши, чтобы изменить последовательность, в которой они будут появляться. С другой стороны, Вы можете использовать кнопки *Move Up* и *Move Down* для перемещения элемента вверх или вниз по списку.

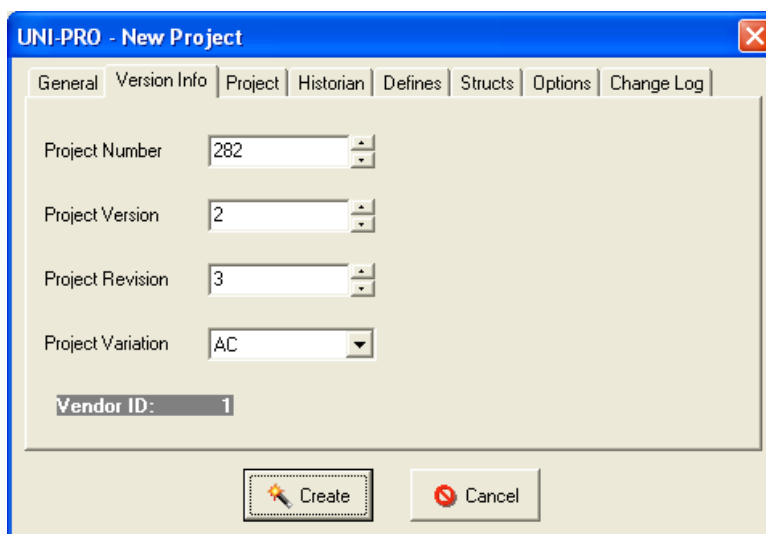
Очень полезной вспомогательной функцией является функция *AutoSort*, которая автоматически упорядочивает элементы в зависимости от их положения на странице. Они будут упорядочены, начиная с элемента, расположенного в верхней левой части страницы, и заканчивая элементом в нижней правой части.

## 4.10 Настройки проекта

В дополнении к основным данным, таким как имя, разработчик, дата создания, проект и номер версии, каждый проект содержит определенное количество установок, характеризующихся по типу их действия, такие как серийные протоколы, наличие архива событий, и “определения“, используемые в алгоритмах. Чтобы получить доступ к этим настройкам, выберите в меню *Project/Property...* или нажмите  и откройте окно свойств проекта.

### 4.10.1 Информация о версии

В окне «Информация о версии» можно задать номер проекта, версию, редакцию и вариацию пользовательского проекта.



В окне есть значение *Vendor ID*, доступное только для чтения; изменяемую информацию представляют следующие пункты:

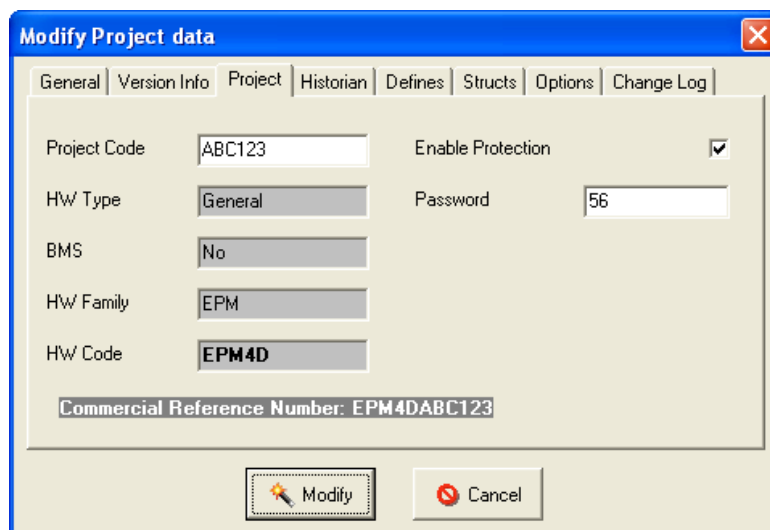
<b>Project Number:</b>	Номер проекта.
<b>Project Version:</b>	Версия проекта.
<b>Project Revision:</b>	Редакция проекта.
<b>Project Variation:</b>	Вариация проекта. Если два проекта одинаковы, но различаются языком EIML-страниц либо оборудованием, не имеет смысла присваивать им различные номера либо версии. Достаточно идентифицировать их только по вариации так, чтобы целевые данные оставались одинаковыми.

Данную информацию всегда можно изменить.

### 4.10.2 Проект

В данном окне можно задать Код Проекта и защиту паролем для загрузки проекта контроллера.





Все значения, отображенные только для чтения, получены от выбранного оборудования и нужны для его идентификации во время загрузки программы.

<b>Project Code:</b>	Код проекта.
<b>HW Type:</b>	Тип аппаратного обеспечения.
<b>BMS:</b>	Указывает на наличие протокола BMS в контроллере.
<b>HW Family:</b>	Семейство оборудования, к которому принадлежит контроллер.
<b>HW Code:</b>	ID аппаратного обеспечения контроллера.
<b>Enable Protection:</b>	Включает защиту паролем.
<b>Password:</b>	Защитный пароль при загрузке программы.

Защита паролем необходима для защиты приложения с целью его загрузки только определенным оборудованием, запрограммированным тем же паролем.

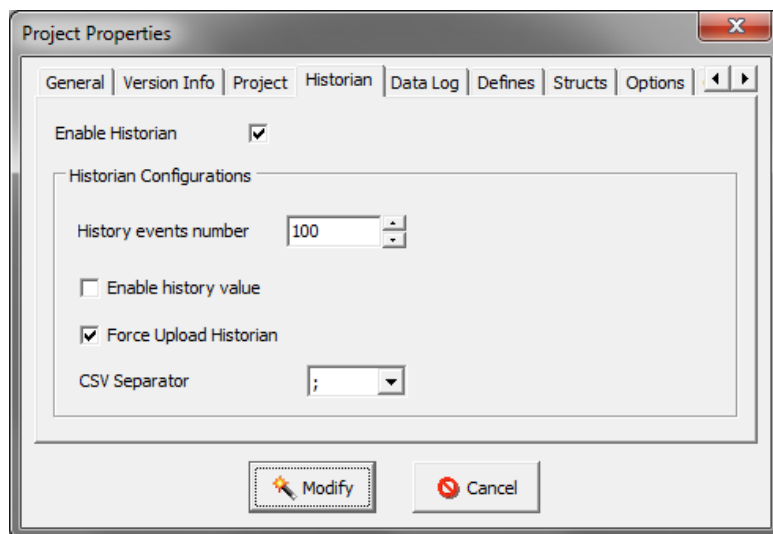
Чтобы запрограммировать один или несколько приборов для получения ПО только с конкретным паролем, необходимо выполнить как минимум 2 этапа, а именно:

- **Первый этап.** Установите *Enable Protection=NO* и задайте (цифрами) нужный пароль *Password*. Выполните компиляцию и загрузку в контроллер.
- **Второй и последующие этапы.** Установите *Enable Protection=YES* и задайте (цифрами) нужный пароль *Password*. Выполните компиляцию и загрузку в контроллер. С этого момента контроллер будет принимать только приложения, содержащие этот пароль (см. пример 56).

Чтобы освободить контроллер от защиты и разрешить загрузку при любых условиях, оставьте *Enable Protection=NO*.

### 4.10.3 История

В окне Historian Вы можете подключать и конфигурировать функции учета событий и сигналов тревоги, имеющиеся в определенных UNI-PRO контроллерах.



Подключение этой функции вызовет соединение библиотеки архива с функциями контроллера. Если Вы не хотите подключать функцию регистрации событий, не ставьте галочку в ячейке *Enable Historian*.

Если Вы поставили галочку, то в поле *Events Number* потребуется установить максимальное число событий, которые будут записаны в память. Очередность Event Historian организована по методу FIFO (в порядке поступления). Например, если Вы ввели 30, самые последние 30 событий будут храниться в памяти контроллера, как только это количество будет превышено, самые старые события будут перезаписаны.

Включение свойства *Enable History Value* также позволяет сохранить значение, связанное с событием истории.

При отключении свойства *Force Upload Historian* история будет сохраняться в памяти даже после новой загрузки программы, иначе всегда удаляется.

Свойство Разделитель CSV позволяет настроить символ-разделитель в файле CSV, сгенерированном контроллером (это свойство является уникальным и общим со свойством журнала данных).

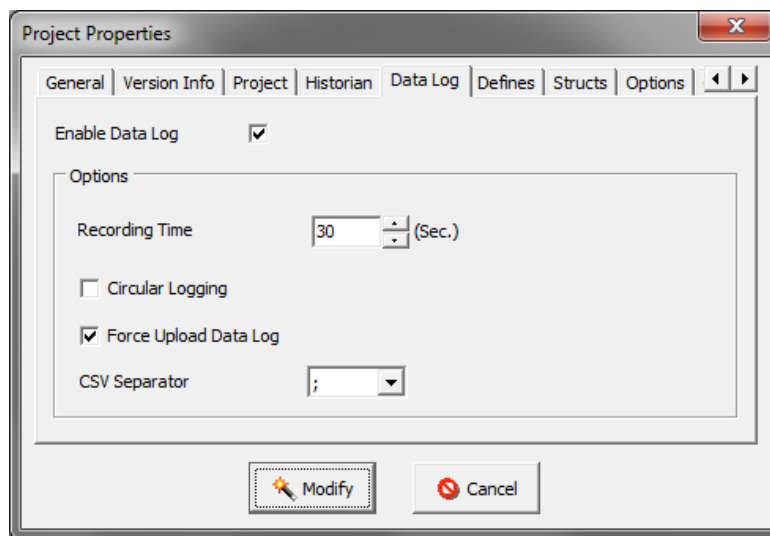
В архиве событий хранится следующая информация:

Данные	Тип	Описание
Дата	CJ_DATETIME	Дата и время события.
Код	CJ_WORD	Код идентификации события.
Очередность	CJ_LONG	Номер очередного события.
Значение	CJ_SHORT	Возможное значение, ассоциируемое с событием.

Чтобы добавить событие в архив, используйте системную библиотеку *HistoryWriteEvent*; чтобы считать событие из архива, используйте библиотеку *HistoryReadEvent* (обратитесь к документу *UNI-PRO: Стандартные библиотеки*).

#### 4.10.4 Data Log (Журнал данных)

В окне *Data Log* вы можете включить и настроить функциональность для регистрации данных проекта, доступную в определенных программируемых контроллерах UNI-PRO.



Включение этой функции приведет к тому, что библиотека журнала данных будет связана с функциями контроллера. Если вы не хотите реализовывать функцию регистрации данных, оставьте флажок *Включить Data Log* снятым.

Изменяя значение *Частота регистрации*, вы устанавливаете время записи (в секундах) и, соответственно, определяет максимальное количество записываемых событий в памяти.

Включение свойства *Циклическое ведение журнала* включает циклическое ведение журнала в памяти, или при превышении выделенной памяти запись перезаписывает самые старые данные. Если свойство отключено, запись идет линейно и поэтому при заполнении памяти больше ничего не записывается.

При отключении свойства *Выполнить загрузку Data Log* записанные данные будут сохраняться в памяти даже после новой загрузки программы, в противном случае они всегда удаляются.

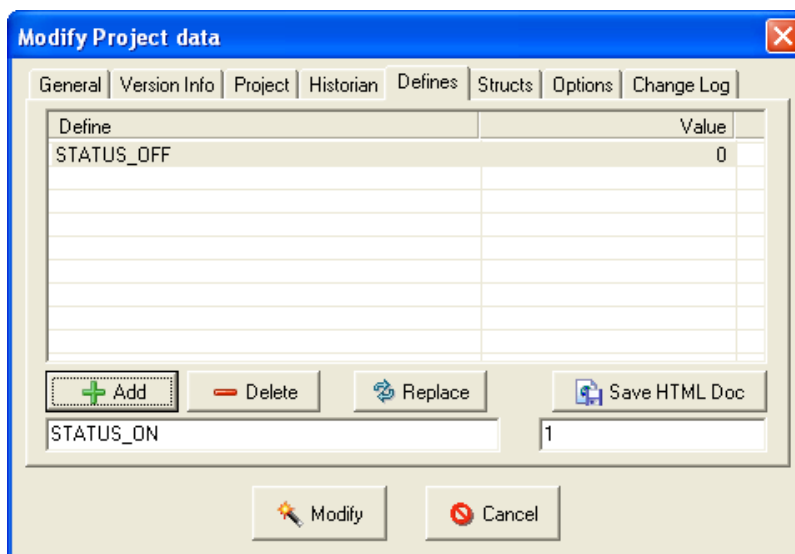
Свойство *Сепаратор CSV* позволяет настроить символ-разделитель в файле CSV, сгенерированном контроллером (это свойство является уникальным и общим со свойством из Истории).

Чтобы определить, какие объекты следует записывать, используйте инструмент *Data Log...* в меню *Инструменты*.

Активацию, деактивацию, отмену и другие функции журнала данных можно настроить с помощью системных библиотек (Библиотеки\Системные\Журнал данных) в стандартных библиотеках UNI-PRO (см. документ UNI-PRO: Librerie Standard).

#### 4.10.5 Определения проекта

Окно определений проекта позволяет задать константы и присвоить им значения, таким образом, Вы сможете использовать их несколько раз в алгоритмах, которые составляют Ваш проект.



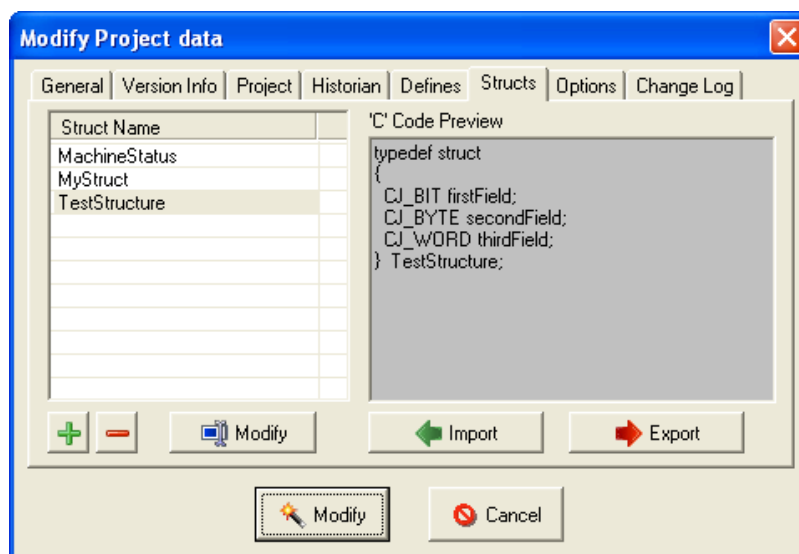
Чтобы добавить определение, введите уникальное имя в левом поле под списком и нажмите *Add*. Это приведет к добавлению определения в список. Если Вы хотите присвоить ему определенное значение, введите его в правом поле под списком.

Чтобы изменить определение, выделите его в списке, измените необходимым образом, а затем нажмите *Replace* для подтверждения.

Для удаления нажмите *Delete*.

#### 4.10.6 Структура

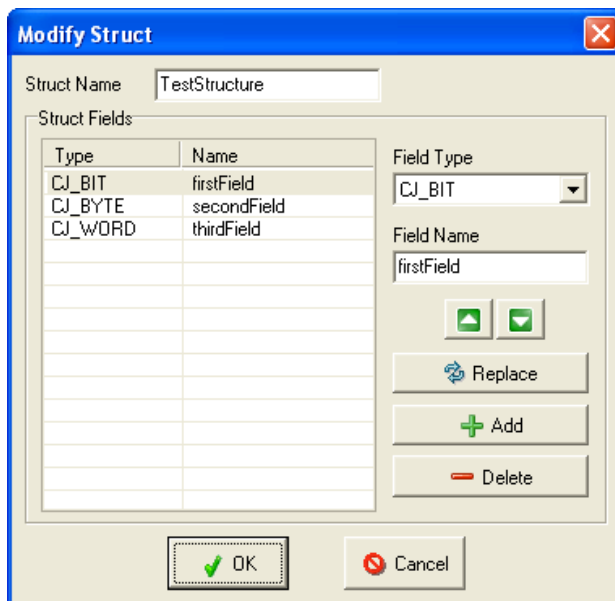
В окне проектных структур можно задать структуры, чтобы многократно использовать их в алгоритмах проекта. Каждый элемент *Struct* должен иметь однозначно определённое имя.



В поле справа приведена версия структуры на языке “С”.

Действие “Export” позволяет создать файл со всеми структурами, имеющимися в проекте. Действие “Import” позволяет загрузить файл, содержащий файл с экспортированными структурами (с помощью кнопки “Export”) из другого проекта. Возможные структуры с уже имеющимся именем загружаться не будут.

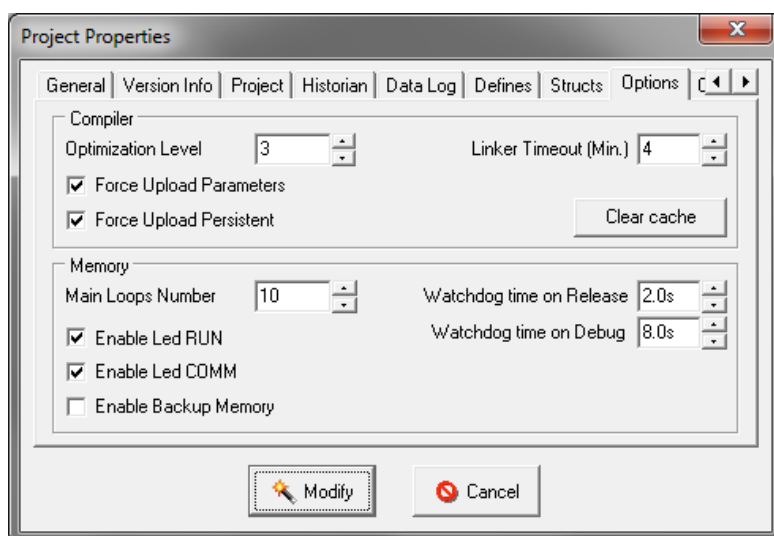
Действия “+” и *Изменить* позволяют добавлять и изменять структуры соответственно с использованием следующего окна:



В окне можно изменить имя элемента *Struct*, заменить, добавить или убрать поля структуры. Необходимо, чтобы каждое поле имело уникальное имя в структуре.

#### 4.10.7 Опции

Используя окно оптимизации, расположенное в опциях проекта, можно установить некоторые параметры, которые оптимизируют выполнение программы. Изменение данных опций рекомендуется только для опытных пользователей.



**Optimizations Level:** Уровень оптимизации компилятора.

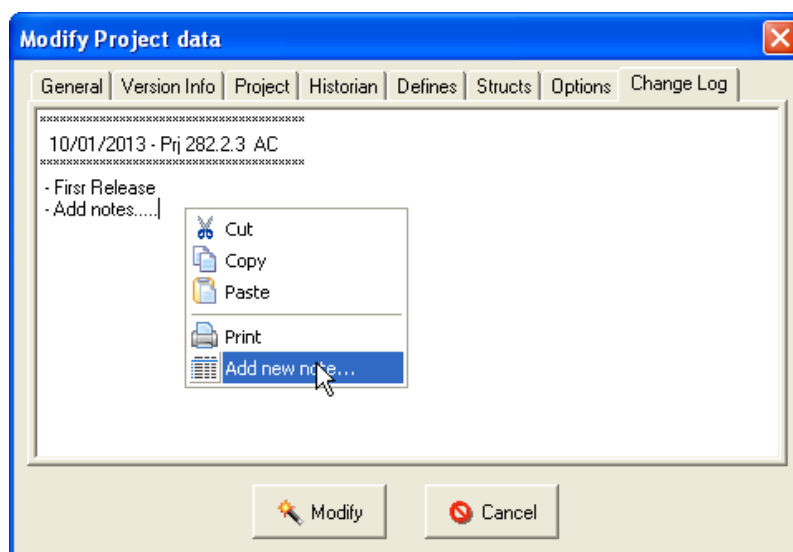
**Force upload parameters :** Принудительная перезапись параметров в E2 во время первого выполнения после загрузки.

<b>Force upload persistent :</b>	Принудительно перезаписывает PERS в E2 во время первоначального выполнения после загрузки.
<b>Linker Timeout:</b>	Время ожидания (минуты) завершения процесса компоновки после компиляции. Для очень больших проектов может потребоваться увеличить это время.
<b>Clear Cache:</b>	Отменяет кеш компиляции проектов.
<b>Main Loops Number:</b>	Используя эту конфигурацию, можно указать количество действий, рассчитанных в группе. Для очень больших проектов, увеличив это число, можно ускорить работу основной программы.
<b>Watchdog time on Release:</b>	Время сторожевого таймера (*) для проектов, скомпилированных во время релиза (без проток
<b>Watchdog time on Debug:</b>	Время сторожевого таймера (*) для проектов, скомпилированных во время отладки (с проток
<b>Enable Led RUN:</b>	Включить/выключить управление светодиодом RUN контроллера (если он отключен, он всегда остается выключенным).
<b>Enable Led COMM:</b>	Включить/отключить управление светодиодом CAN контроллера (если он отключен, он всегда остается выключенным).
<b>Enable Backup Memory:</b>	Он позволяет разрешить или запретить копирование конфигурации и параметров приложения в память E2 с помощью системных команд.

(\*) Время сторожевого таймера – это время, в течение которого алгоритмы завершают исполнение кода; по истечении данного времени контролер выполняет перезагрузку оборудования. Это своеобразная система защиты от бесконечных циклов или слишком длинных процессов, которые подвергают риску корректное функционирование контроллера.

#### 4.10.8 Перечень изменений проекта

Чтобы помочь пользователю отслеживать изменения, сделанные в различных проектах, среда программирования предусматривает наличие истории проектов (Change Log). В этом окне можно осуществить запись выполненных изменений и любых других замечаний для каждой версии.



Щелчком правой кнопкой мыши по окну можно просмотреть соответствующее меню, которое в дополнении к стандартным действиям Вырезания/Копирования/Вставки позволяет:

- распечатывать содержимое перечня изменений;

- добавлять в текущее положение курсора новую запись (также добавляются несколько строк с заголовками, содержащими дату и текущую версию проекта).

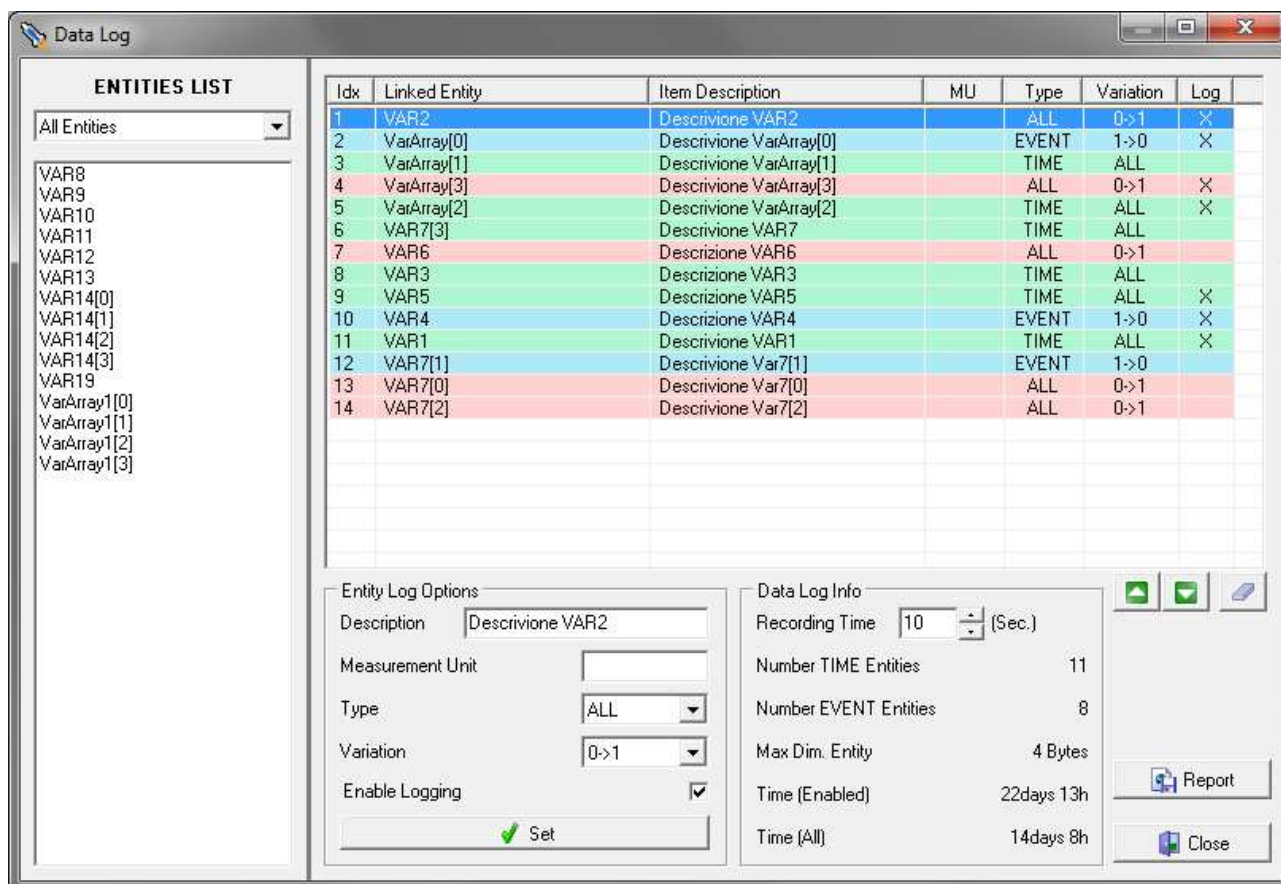
## 4.11 Data Log (Журнал данных)

Из меню *Инструменты->Data Log* можно получить доступ к фрейму, позволяющему выбирать, какие объекты экспортировать и настраивать для журнала проекта.

Для каждого экспортируемого объекта вы можете настроить следующие свойства:

- **Описание.** Максимум 24 символа.
- **Единица измерения.** Максимум 6 символов.
- **Тип.** Тип регистрируемой сущности.
  - *TIME.* Регистрация выполняется по времени с использованием параметров времени регистрации.
  - *EVENT.* Запись выполняется при изменении значения.
  - *ALL.* Включает в себя как функциональность *TIME*, так и *EVENT*.
- **Variation (Проекта).** Определяет, когда регистрировать сущность типа *EVENT*.
  - *ALL.* Все варианты.
  - *0->1.* Вариация от 0 до 1 (больше 0)
  - *1->0.* Вариация от 1 до 0 (меньше 1)
- **Включить регистрацию.** Определяет, разрешено ли экспортируемому объекту ведение журнала по умолчанию или нет.

Вы можете настроить максимум 32 объекта *TIME* и 32 объекта *EVENT*. Если сущность экспортируется и относится к типу *ВСЕ* (*TIME* и *EVENT*), она занимает место в обоих списках.



Чтобы добавить объекты в экспортируемый список, просто перетащите их в правый список, удаление может происходить путем перетаскивания в противоположном направлении или с помощью функции *Удалить*, связанной с клавишей *Del*.



Чтобы изменить свойства экспортируемого объекта, выберите его и отредактируйте новые свойства в области *Опции Сущность* и подтвердите выбор, нажав кнопку *Задать*, или используйте раскрывающееся меню, которое появляется при нажатии на выбор правой кнопкой.

В разделе *Data Log Инфо* представлена некоторая информация, включая время, в течение которого у вас закончится свободное место для записи. Вы также можете изменить время записи здесь, чтобы увеличить или сократить общее время журнала.

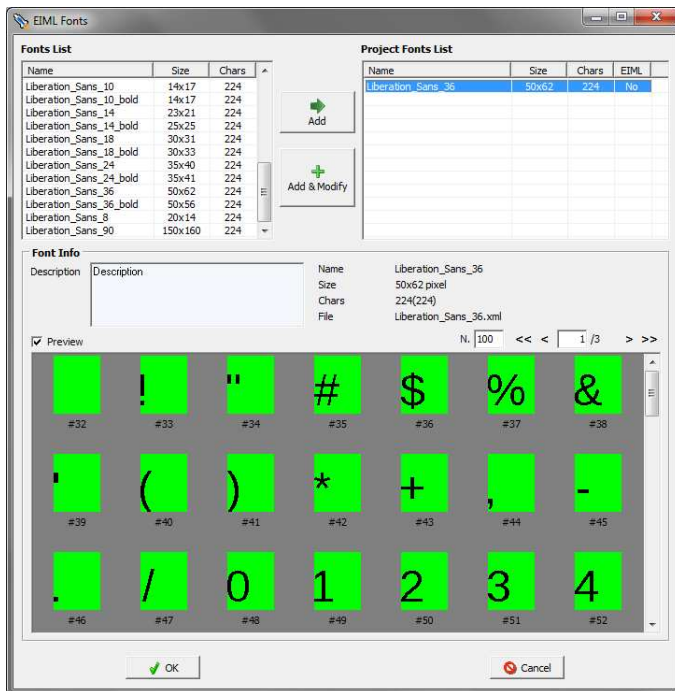
Через *Отчет* генерируется документация в формате html, обобщающая сделанные конфигурации.

Для управления журналом данных из приложения вы можете обратиться к библиотекам в *System/Data Log*, с помощью которых вы можете:

- включить/отключить регистрацию данных.
- включить/выключить запись отдельных сконфигурированных объектов.
- настроить время записи.
- извлекать данные из отдельных записей для печати на дисплее.
- создайте файл .CSV на USB-порту со всеми записями.
- проверить состояние памяти, выделенной для журнала.

## 4.12 Пользовательский шрифт в соответствии с EIML

В цветных страницах EIML вы можете использовать пользовательские шрифты вместо стандартного шрифта страницы. Все шрифты UNI-PRO загружаются при запуске программы, и через конфигуратор, доступный в меню Инструменты->Шрифт EIML, вы можете выбрать и отредактировать, какие из них использовать в проекте.

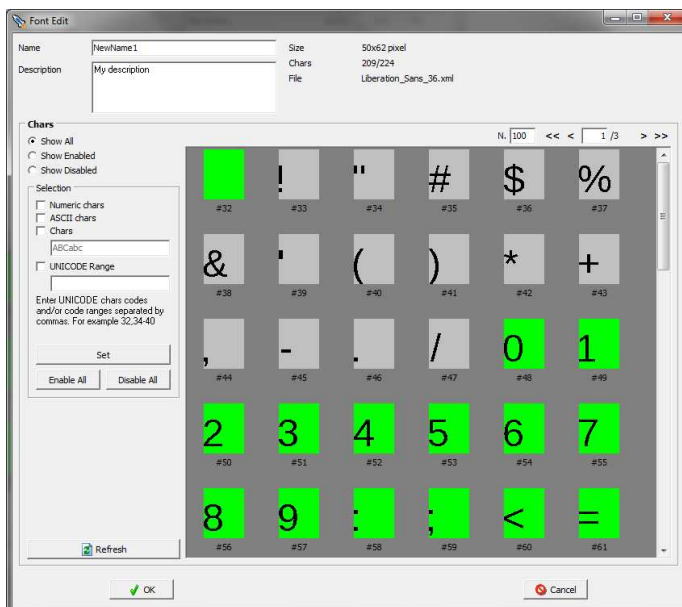


Вы можете решить использовать весь шрифт как есть, добавив его в проект с помощью кнопки *Добавить*.

Кнопка *Добавить* и *Изменить* позволяет добавить новый шрифт, изменив его напрямую, выбрав всего несколько символов.

Изменение шрифта проекта можно выполнить позже с помощью команды *Изменить* из раскрывающегося меню.

Выбор подмножества символов может понадобиться, если у вас проблемы с нехваткой памяти при компиляции вашего проекта, это делается через это окно (из команды *Изменить*):



Есть несколько опций, облегчающих выбор символа, в противном случае вы можете выбрать их вручную, включив/отключив их клавишей F3.

Включенный символ имеет зеленый фон, а отключенный персонаж - серый фон.

Отключенные символы не будут доступны на страницах EIML.

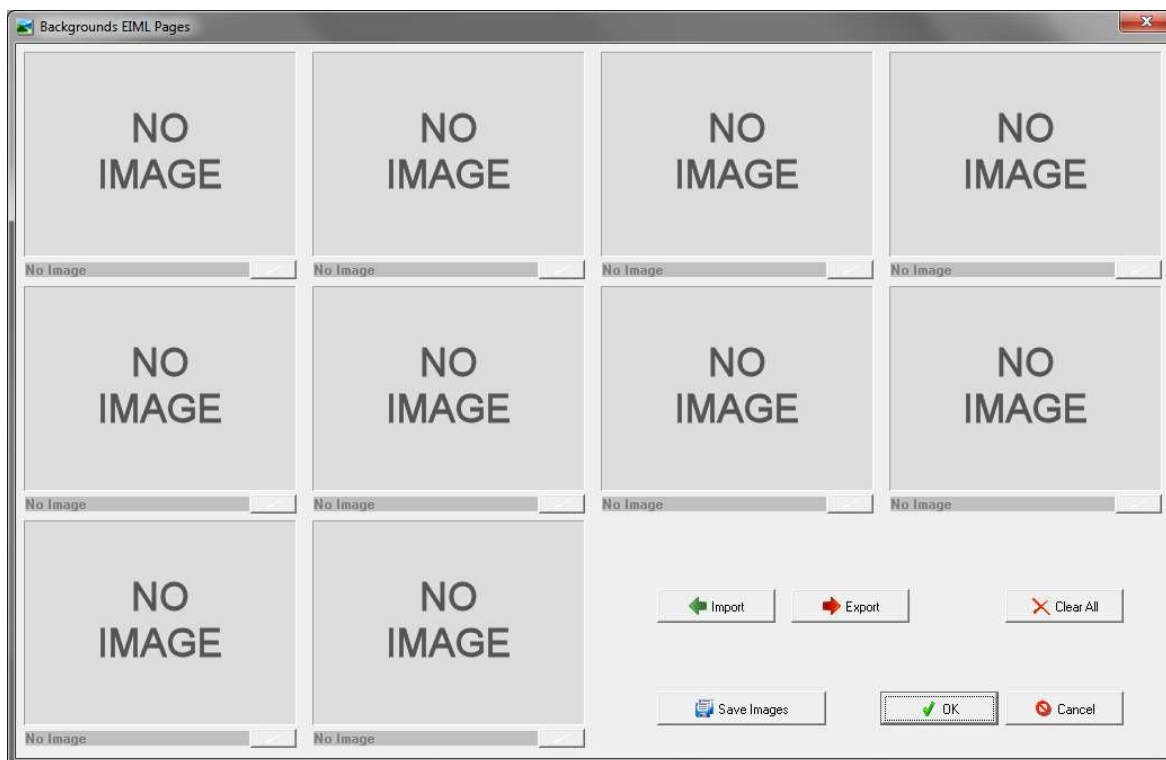
### 4.12.1 Сохранение выбранных шрифтов

Используемые шрифты проекта, оригинальные или модифицированные, будут сохранены в виде файлов драйвера XML вместе с проектом, они будут сохраняться каждый раз, когда вы сохраняете проект, и будут храниться на том же уровне вложения файла проекта ucjr (например, «ProjectName\_X.ucjr») в специально созданной папке (например, "\\Fonts\_ProjectName\_X\\*.xml").

Имя файла драйвера XML, а также имя папки не следует изменять. Если вы переименовываете файл проекта .isjr, также необходимо переименовать папку таким же образом (не меняйте имена файлов .xml). При открытии проекта UNI-PRO загрузит шрифт из своего файла драйвера, если файл отсутствует, сообщается об ошибке. Во избежание неконтролируемой потери данных, если во время загрузки драйвера возникает ошибка, закройте проект без сохранения и переместите XML-файл драйвера в указанную папку поиска.

### 4.13 Фоны EIML

Из меню *Инструменты*->*Фон EIML* можно получить доступ к окну, позволяющему выбрать до 10 фонов для цветных страниц EIML проекта.



Чтобы вставить/изменить фон, дважды щелкните изображение и выберите растровое изображение размером 320x240 пикселей. После загрузки изображения будет показан предварительный просмотр, и оно получит имя, аналогичное имени загруженного растрового изображения, чтобы его было легче распознать.

Если загружаемое растровое изображение не состоит из 256 цветов палитры, нужно будет подождать несколько секунд из-за конвертирования (изображение может немного отличаться от оригинала).

Кнопка *Импорт* и *Экспорт* позволяет импортировать и экспортировать фоны в файл XML, который можно использовать в других проектах UNI-PRO.

Чтобы стереть фон, нажмите кнопку X в правом нижнем углу изображения. С помощью кнопки *Очистить все* можно очистить все фоны сразу.

Каждое изображение можно сохранить, щелкнув правой кнопкой мыши по изображению и выбрав *Сохранить изображения*. Кнопка *Сохранить изображения* полезна для сохранения всех изображений в папку проекта.

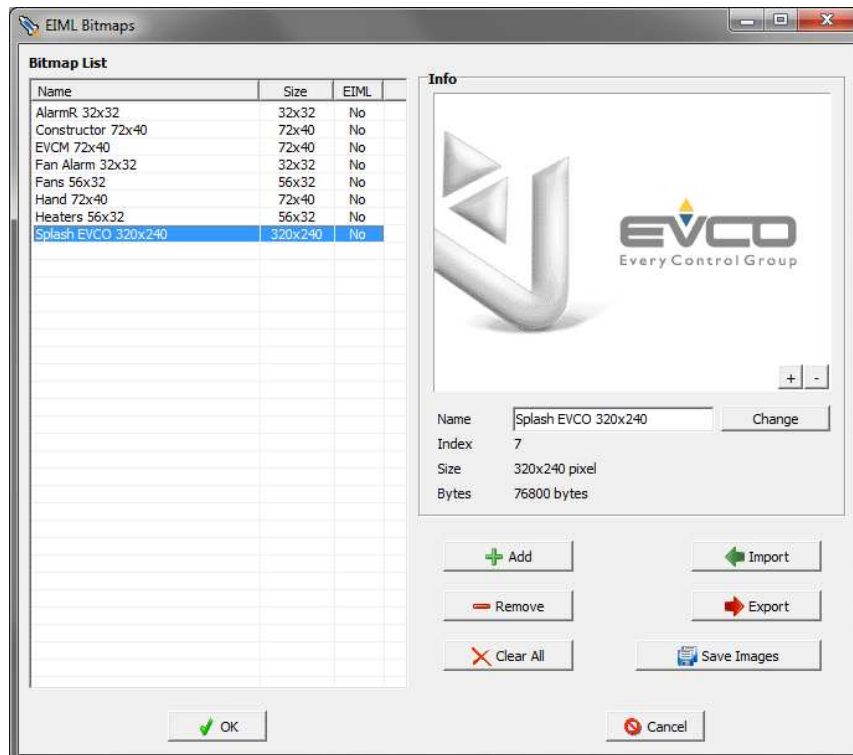
Начиная с версии 3.19.0.0 UNI-Pro фоны, загружаемые с помощью этого инструмента, будут сохраняться в папке на том же уровне, что и файл проекта ucjr (пример «ProjectName\_X.ucjr», папка сохранения «\ Background\_ProjectName\_X»). Имя папки, содержащей изображения, и название изображений не должны изменяться.

Если есть необходимость переименовать файл проекта ucjr, папка также должна быть переименована таким же образом. При открытии проекта UNI-PRO загрузит файлы, если один из файлов отсутствует, будет сообщено об ошибке.

## 4.14 EIML растровые изображения

Из меню *Инструменты* -> *Рисунок EIML* вы получаете доступ к окну, которое позволяет вам выбирать значки, которые можно использовать в проекте, все растровые изображения, включенные в этот список, могут использоваться, на страницах EIML, поддерживающих эту функцию.

Использование значков из этого списка сохраняет некоторый код во флэш-памяти контроллера; по факту, страница больше не будет выделять место для значка, а только ссылку на него, задачей контроллера будет распознать эту ссылку и нарисовать связанное растровое изображение. Поэтому, если данная иконка будет использоваться в проекте несколько раз, нет необходимости вставлять ее несколько раз на страницы, но достаточно будет вставить его в список выше, а затем указать в свойстве "Bitmap" элементов типа значка на страницах EIML.



Если загружаемое растровое изображение не состоит из 256 цветов палитры, необходимо будет подождать некоторое время, пока оно преобразуется. Преобразованное изображение может немного отличаться от оригинала.

После выбора изображения можно изменить его собственное имя с помощью кнопки *Изменить*. С помощью кнопки *Очистить все* можно очистить все растровые изображения сразу. Кнопки *Импорт* и *Экспорт* позволяет импортировать и экспортировать растровое изображение в виде XML-файла, переносимого для других проектов UNI-PRO.

Каждое изображение можно сохранить, щелкнув правой кнопкой мыши по изображению и выбрав *Сохранить изображения*. Кнопка *Сохранить изображения* полезна для сохранения всех изображений в папку проекта.

Начиная с версии 3.19.0.0 UNI-Pro растровые изображения, загружаемые с помощью этого инструмента, будут сохраняться в папке на том же уровне, что и файл проекта ucjr (пример «ProjectName\_X.ucjr», папка сохранения «\ Background\_ProjectName\_X»). Имя папки, содержащей изображения, и название изображений не должны изменяться.

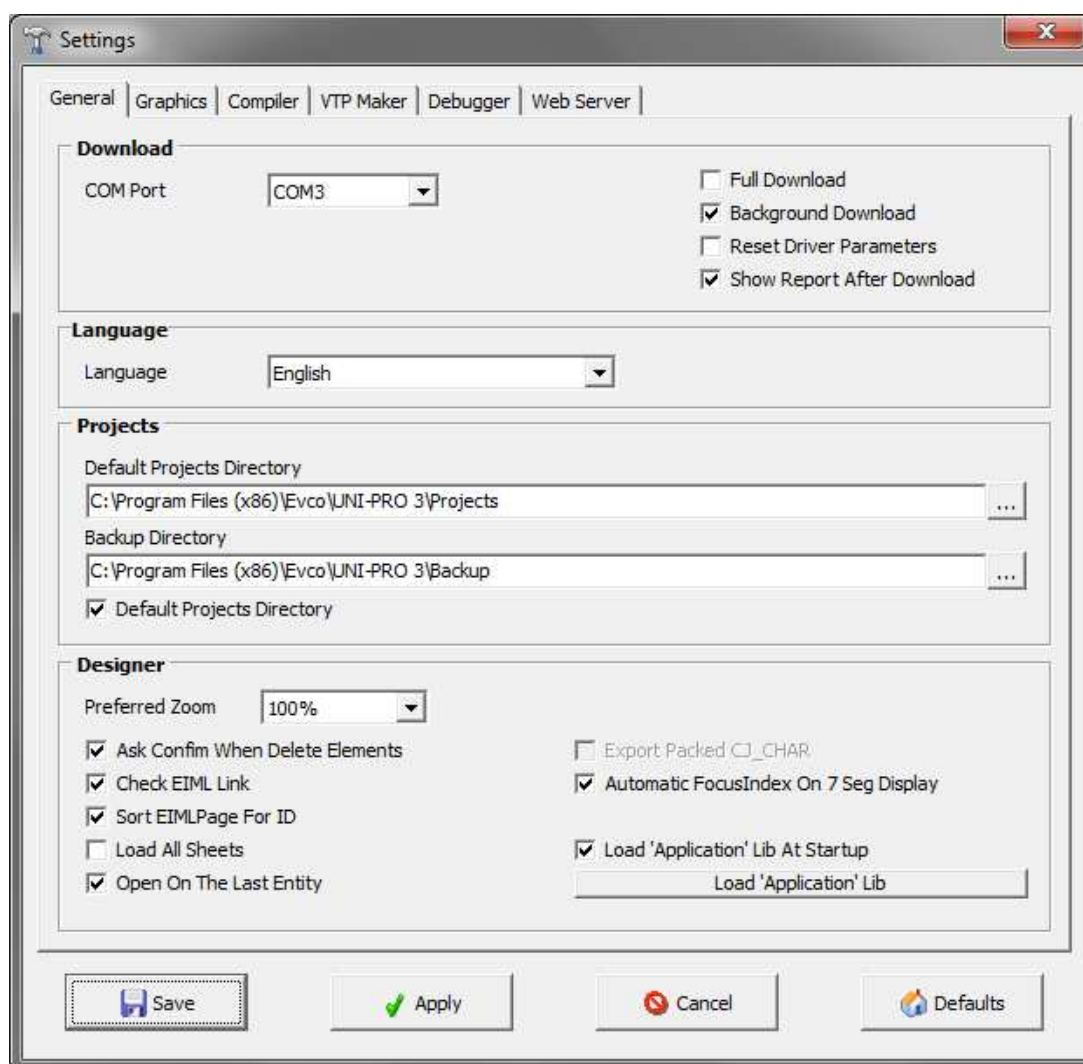
Если есть необходимость переименовать файл проекта ucjr, папка также должна быть переименована таким же образом. При открытии проекта UNI-PRO загрузит файлы, если один из файлов отсутствует, будет сообщено об ошибке.

## 4.15 Конфигурация среды

Среда программирования UNI-PRO является очень гибкой, так как легко может быть настроена по желанию пользователя. Активируя пункт меню *Инструменты/Настройки*, можно получить доступ к следующим окнам:

- General/Общие
- Graphics/Графика
- Compiler/Компилятор
- VTP Maker/Создание VTP
- Debugger/Отладчик
- Web Server/Web-сервер

Экран *Общие* содержит разделы *Загрузка*, *Язык*, *Проекты* и *Дизайнер*. Все варианты которых будут разобраны здесь ниже:



### Default COM port :

Последовательный порт, используемый для операций загрузки. Если подключены контроллеры с-pro 3 микро/кило, COM-порт распознается автоматически, а установленное вами значение игнорируется.

### Reset Driver Parameters:

Вызывает сброс параметров драйвера контроллера после загрузки ПО.

**Full download :** Принуждает к полной загрузке всего исполняемого файла. Чтобы ускорить загрузку параметров, можно отменить выбор этой опции и после корректного удаления имеющейся в микроконтроллере флэш-памяти продолжить загрузку только используемой флэш-памяти.

**Background download:** Позволяет работать над проектом на этапе загрузки.

**Show Report After Download:** Если выбрано, после каждой загрузки в контроллеры s-pro 3 micro/kilo будет показан файл отчета.

**Language:** Изменяет язык среды разработки. Для применения модификации необходимо снова запустить UNI-PRO.

**Default projects directory :** Каталог по умолчанию, предложенный для сохранения файлов проекта.

**Backup directory :** Каталог, используемый для резервного копирования файлов проекта (см. параграф «Резервное копирование проекта»).

**Auto increment revision when backup project's files :** Активировав эту опцию, каждый раз, когда над файлами проекта выполняются операции резервного копирования, номер ревизии будет автоматически увеличиваться, в противном случае пользователю придется выполнять эту операцию.

**Ask confirm when delete elements :** Активирует или деактивирует запрос на подтверждение удаления объектов и компонентов EIML.

**Preferred zoom :** Позволяет применить свойство масштабирования при загрузке проекта. После этого уровень масштабирования может быть легко изменен с помощью контекстного меню, расположенного на панели инструментов.

**Check EIML link :** Проверяет, соединены ли должным образом все элементы, связанные с сущностями, представленными в секции графического интерфейса. Выполните эту проверку во время фазы моделирования графического интерфейса.

**Sort EIML Page for ID:** При подключении приводит к упорядочиванию EIML-страниц в соответствии со свойствами *Id*; в противном случае они будут отображены в алфавитном порядке.

**Load All Sheet:** При включении проект будет открыт полностью. При отключении проект будет открыт частично, загружаются только страницы EIML и основные листы.

**Open On The Last Entity:** Если этот параметр включен, повторное открытие недавнего проекта будет произведено на объекте, на котором проект был закрыт в предыдущий раз.

<b>Export Packed CJ_CHAR:</b>	Если подключено, то экспорт по протоколу Modbus для строковых типов (CJ_CHAR) выполняется по группам (составленных из 2 байт для каждого регистра).
<b>Automatic FocusIndex On7 Seg Display:</b>	Если включено, включается автоматическое управление фокус-индексом для страниц с 7-сегментным дисплеем.
<b>Load 'Application' Libs at Startup</b>	Если включено, библиотеки Application libraries загружаются при запуске UNI-PRO. С помощью кнопки <i>Загрузить Application lib</i> вы можете загрузить библиотеки приложений в любое время, не изменяя значение свойства.

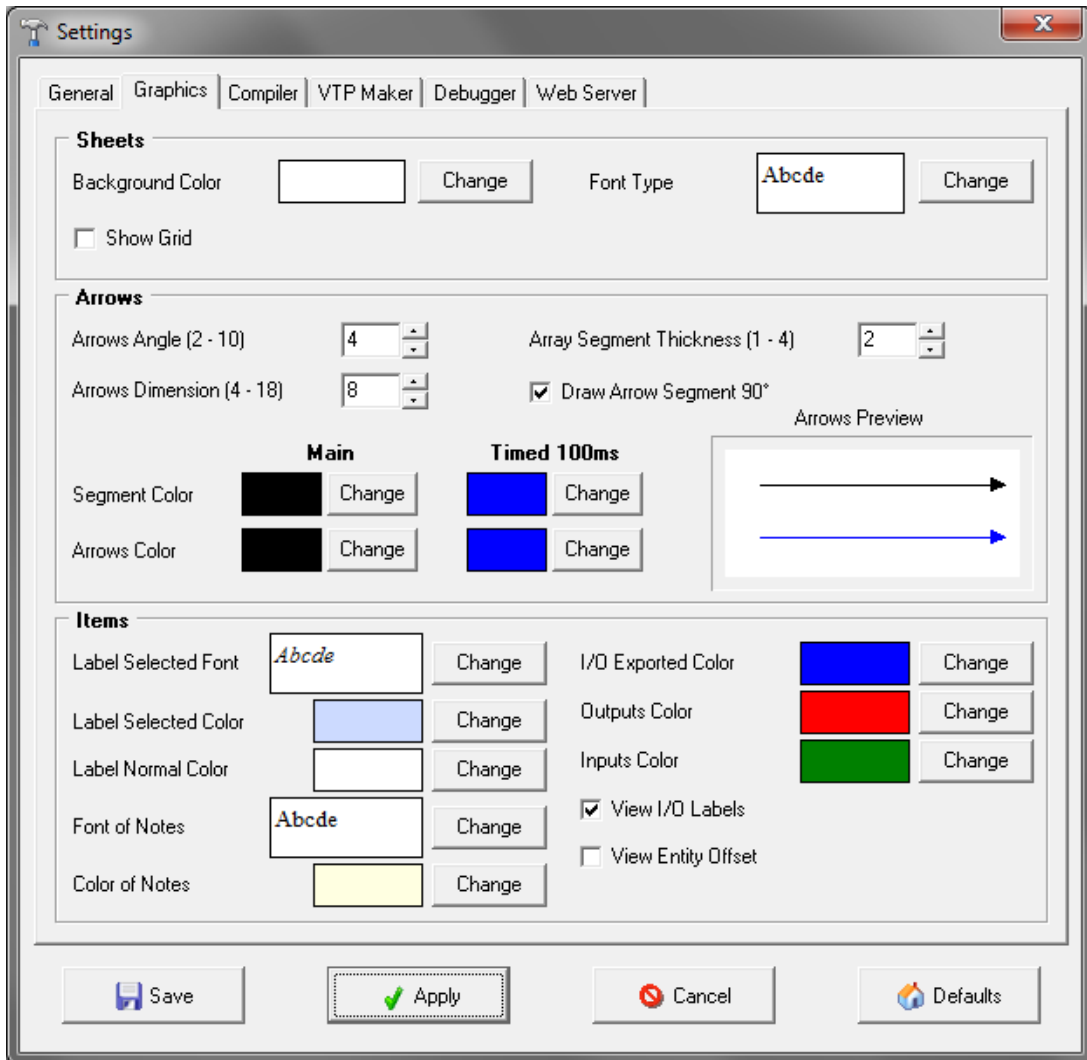
#### 4.15.1 Опция «Загрузить все листы»

Когда эта опция включена, при обычном использовании UNI-PRO считается, что проект полностью открыт. Однако в этом режиме память используется очень интенсивно, и открытие проекта, в зависимости от размера, может занимать несколько минут. Для облегчения аботы среды рекомендуется держать эту опцию выключенной, так меньше используется памяти, а также открытие проекта происходит намного быстрее, будут загружены только страницы EIML и основные схемы. Доступ к еще не загруженной сущности графически может осуществляться как вручную, просматривая и открывая нужные схемы, так и автоматически, когда вы пытаетесь получить доступ к объекту способом, предусмотренным UNI-PRO. Открытие пошагово схем занимает время, но намного меньшее, чем которое потребовалось бы для открытия всего проекта, после открытия листа он остается открытым на протяжении всего сеанса проекта.

Используя опцию *Открыть с последней схемы* (когда открытие завершено), вы можете открыть схему, с которой вы работали в последний раз, когда был закрыт текущий проект.



Окно *Graphics* состоит из секций *Sheets*, *Arrows* и *Items*. Опции каждой из них будут проанализированы ниже:



**Sheets background color :** Устанавливает цвет фона папок проекта.

**Sheets Default font :** Устанавливает шрифт для всех существующих компонентов внутри папок проекта.

**Show Grid :** Отображает сетку на рабочих листах; если свойство подключено, можно также задать ее цвет.

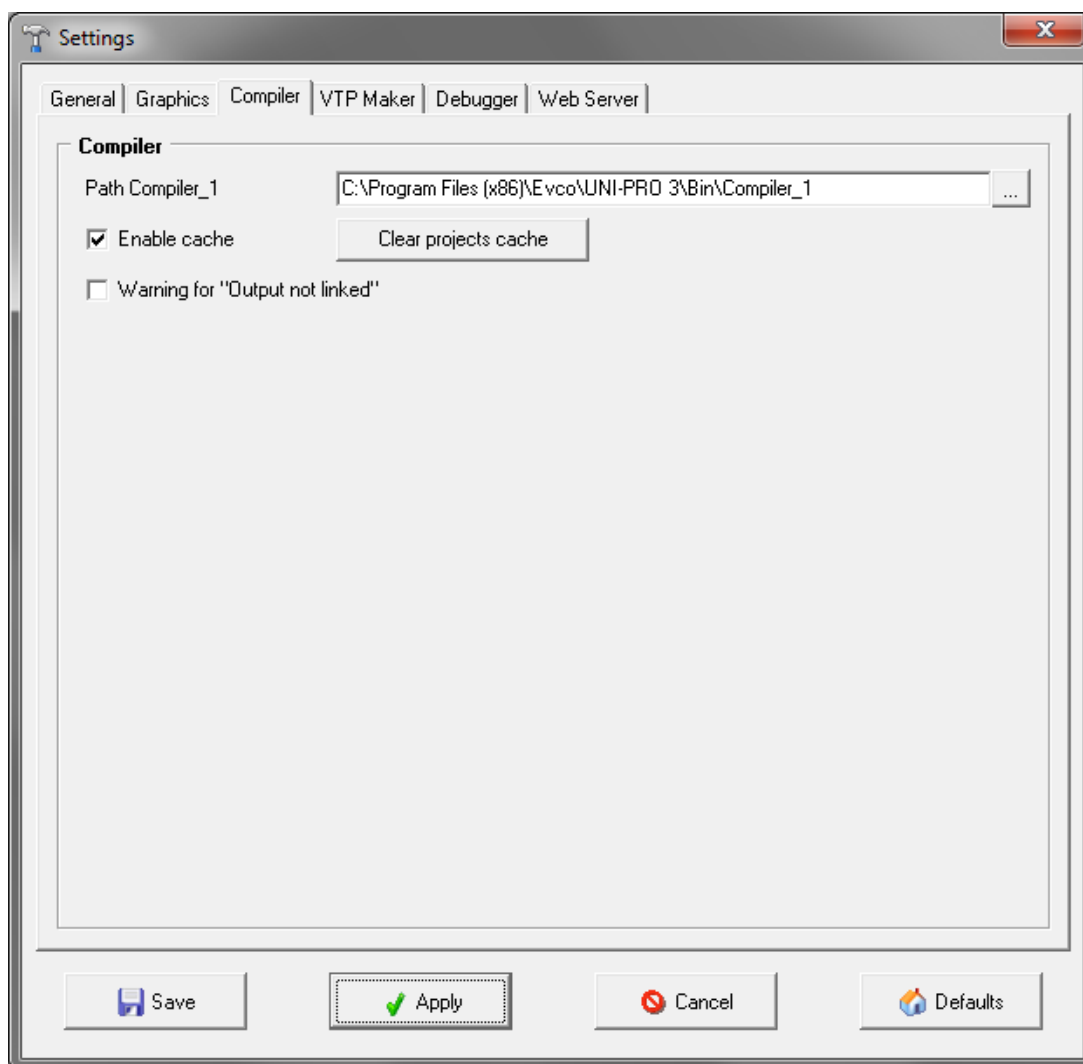
**Arrows angle :** Изменяет угол раствора кончика стрелки, соединенного с сущностью.

**Arrows dimension :** Изменяет величину кончика стрелки, соединенного с сущностью.

**Array Segment thickness :** Изменяет толщину линий, соединенных с сущностью-массивом (сущности со свойством array, превышающим единицу).

<b>Draw Arrow Segment 90°:</b>	Если включено, то при рисовании стрелок используются квадратные сегменты.
<b>Segment color and Arrows color :</b>	Применяя это свойство, можно изменять цвет стрелок. Для обеих опций необходимо указать используемый цвет, если маршрут вычисляется во время основной задачи каждые 100мс (см. параграф <b>Определение задач выполнения</b> ).
<b>Preview :</b>	Окно предварительного просмотра, в котором отображаются примеры установленных параметров стрелок.
<b>Label selected font :</b>	Устанавливает шрифт, применяемый для выбранных объектов.
<b>Label selected color :</b>	Устанавливает цвет фона текста для выбранных сущностей.
<b>Label normal color :</b>	Устанавливает цвет фона текста для не выделенных сущностей.
<b>Font of Notes :</b>	Шрифт, который будет применяться к объектам типа Note.
<b>Color of Notes :</b>	Цвет фона текста для объектов типа Note.
<b>I/O exported color :</b>	Устанавливает цвет, присваиваемый входам/выходам сущностей, вносимых в дочерние папки, которые в них содержатся.
<b>Outputs color :</b>	Устанавливает цвет, присваиваемый выходам сущностей.
<b>Inputs color :</b>	Устанавливает цвет, присваиваемый входам сущностей.
<b>View I/O Labels:</b>	Показывает/скрывает метку с именем зажима для папки, алгоритма, библиотеки...
<b>View Entity Offset:</b>	Если включено, разрешить отображать свойство смещения под объектами.

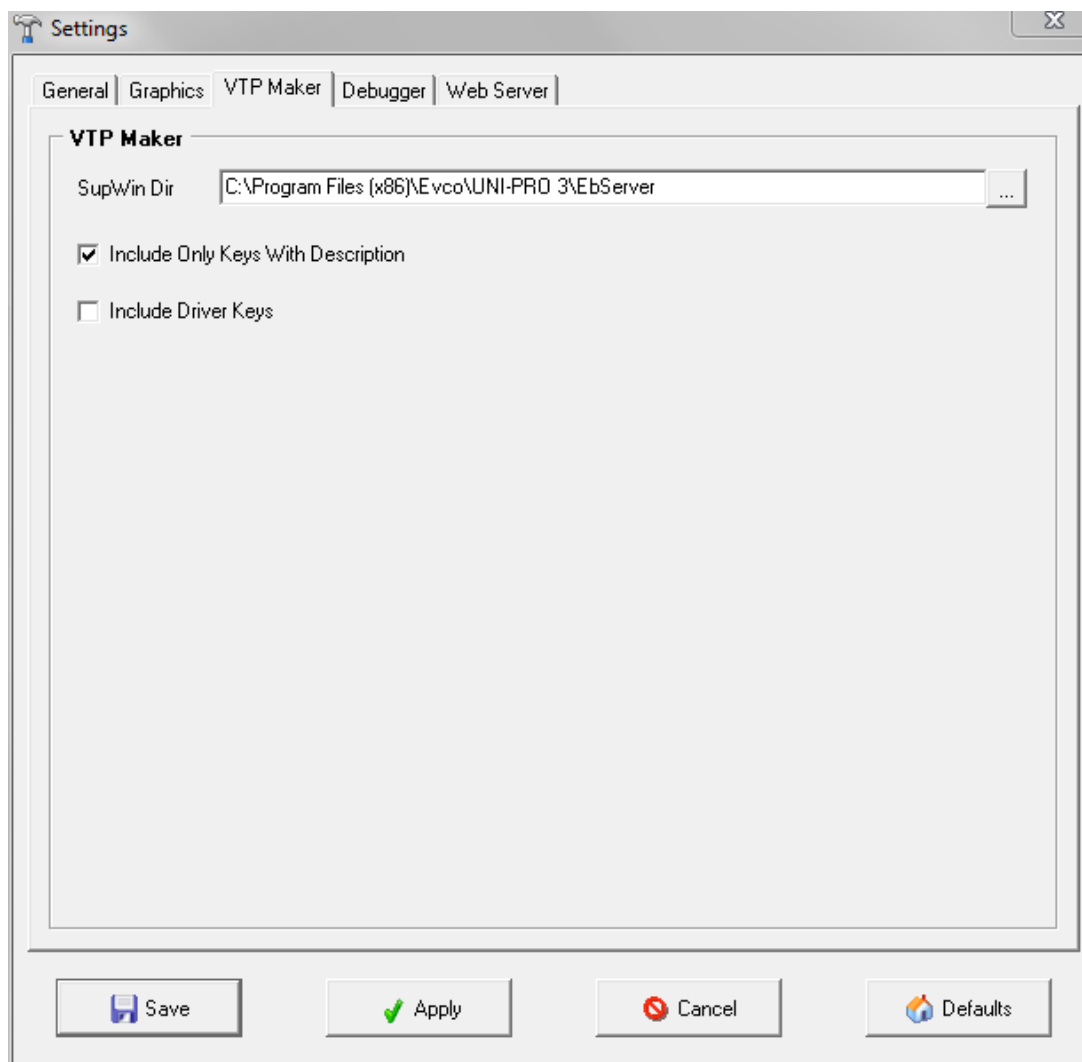
Экран *Compiler* содержит конфигурацию компиляторов C.



Варианты следующие:

<b>Path Compiler_1:</b>	Укажите полный путь, где находятся исходные коды Compiler_1 для компиляции проектов.
<b>Enable cache:</b>	Это позволяет использовать кэш при компиляции.
<b>Clear projects cache:</b>	Очищает весь кэш проектов.
<b>Warning "Output not linked":</b>	Включить / отключить предупреждение о компиляции, когда есть не активные алгоритмы (поскольку их вывод ни с чем не связан)

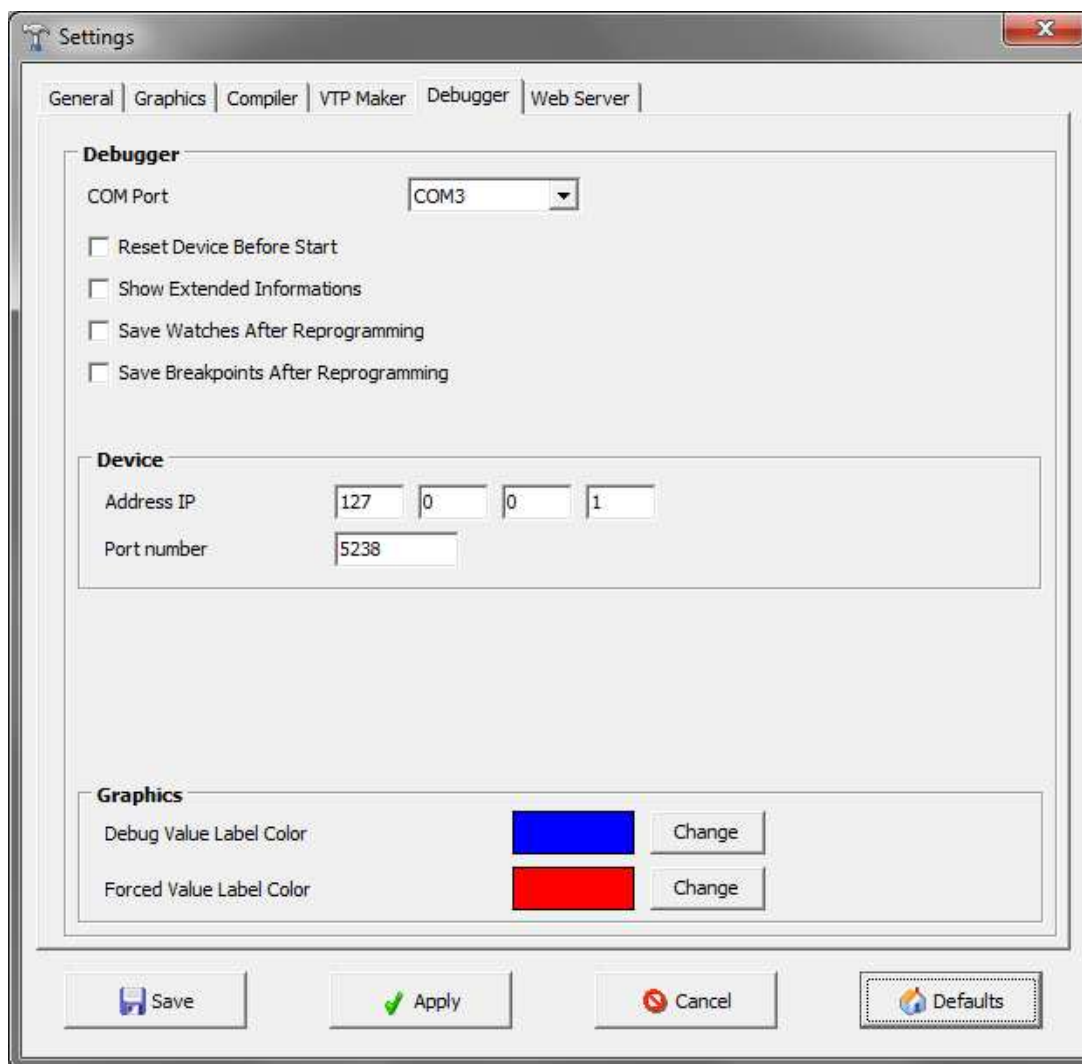
Окно *VTP Maker* содержит опции конфигурации, используемые инструментом VTP Maker для создания VTP файла драйвера, необходимого для связи с программным обеспечением EVCO.



Приведем описание опций:

<b>SupWin Dir :</b>	Определяет каталог назначения для сгенерированных файлов VTP Maker. По умолчанию: <code>..\UNI-PRO 3\EbServer</code>
<b>Include Only Keys With Description :</b>	Используется для специализированного протокола Evcobus. Формирует драйвер кнопок VTP только для сущностей с установленным свойством description.
<b>Include Driver Keys :</b>	Если опция выбрана, то в драйвер файлов VTP включается используемый контроллером драйвер оборудования.

Окно *Debugger* содержит опции конфигурации, используемые для отладки инструмента, например, для создания связи.

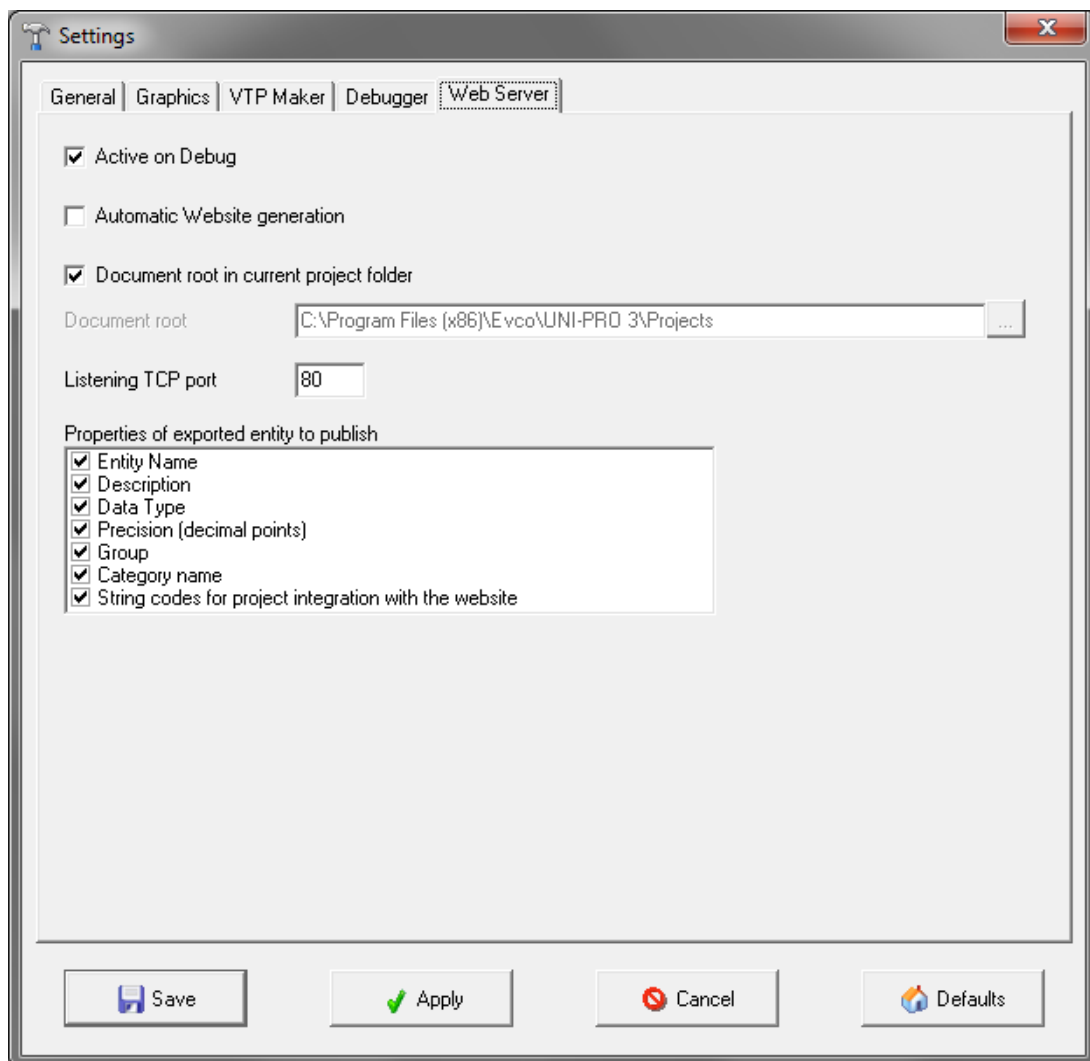


Приведем описание опций:

<b>COM Port :</b>	Последовательный порт, используемый для связи с контроллером во время отладки. Если у контроллера есть только порт программирования и нет порта отладки, то данное свойство должно идентифицироваться с COM-портом, выбранным для загрузки. Если подключены контроллеры s-pro 3 micro/kilo, то COM-порт автоматически распознается, а заданное Вами значение игнорируется.
<b>Reset device before start:</b>	Перезагружает контроллер в начале каждого сеанса отладки.
<b>Show Extended Informations :</b>	Если опция выбрана, то на панели состояния будет показана более подробная информация, как, например, индекс задачи, во время которой происходит прерывание программы в соответствии с установленной точкой.
<b>IP Address:</b>	IP-адрес подключенного контроллера для использования отладчика через Ethernet.

<b>Port number:</b>	ТСР-порт, выделенный для отладчика через Ethernet.
<b>Show watches after reprogramming:</b>	Сохранение и восстановление часов после перепрограммирования контроллера.
<b>Show breakpoints after reprogramming:</b>	Сохранение и восстановление точек останова после перепрограммирования контроллера.
<b>Debug Value Label Color:</b>	Цвет значений отладчика, расположенных под сущностью.
<b>Forced Value Label Color:</b>	Цвет вынужденных значений отладчика (для аналогового входа и цифрового входа).

Вкладка *Web-сервера* содержит параметры для имитации операций Web-сервера при активной отладке.



Варианты следующие:

<b>Active On Debug :</b>	Он определяет, использовать ли симулятор для Web-сервера при запуске отладчика. Если вы отмените выбор, симулятор Web-сервера не будет активен.
--------------------------	---

<b>Automatic Website Generation:</b>	При активации генерирует и автоматически копирует все файлы на тестовый стандартный сайт.
<b>Documet root in current project folder:</b>	При активации создает корневую папку веб-сайта в текущей папке проекта (на том же уровне, что и папка «\exe»).
<b>Documet root:</b>	Указывает директорию, из которой разворачивается дерево сайта. Включается, только если свойство <i>Корень документа в текущей папке проекта</i> отключено.
<b>Listening TCP port:</b>	Прослушивание порта TCP для симулятора Web-сервера.
<b>Properties of exported entity to publish:</b>	Он определяет, какие свойства объектов для экспорта на Web-сервер будут создаваться и управляться на Web-сайте.

#### 4.16 Экспорт / Импорт

Используя меню экспорта и импорта, Вы можете копировать EIML-страницу из одного проекта в другой. С помощью пункта меню *File->Export->Export EIML Pages* можно выбрать (среди страниц проекта) страницы на экспорт, при этом будет создан файл, который будет моментально скопирован в проект UNI-PRO. С помощью пункта меню *Edit->Remove EIML Pages* можно отобразить то же поле с такими же фильтрами для удаления страниц (через их выбор).

С помощью пунктов меню *File->Import->Import EIML Languages* и *File->Export->Export EIML Languages* можно осуществить импорт и экспорт многоязычного файла CSV, содержащего все многоязычные строки EIML-страниц. Сохранённые файлы могут быть обработаны инструментом *EIML Translator*.

#### 4.17 Экспорт внутренних переменных библиотек/макроблоков

Все сущности типа VAR, PAR, PERS и FIX, закрытые в библиотеках и макроблоках, можно сделать общедоступными, даже если они обычно недоступны, для экспорта этих сущностей необходимо активировать свойство *Экспорт LIB*.

Чтобы использовать эту функцию, перед созданием библиотеки выберите объекты, которые вы хотите «экспортировать», и установите для каждого свойство *Экспорт LIB*, а затем создайте новую библиотеку. Отныне, когда будет использоваться новая библиотека, «общедоступные» объекты будут видны в списке объектов инструментов и на страницах EIML, поэтому можно будет изменить некоторые свойства (имя, описание, значение по умолчанию, ...) и связать их на страницах EIML и в инструментах сетевых протоколов (журнал данных, modbus, bacnet, ...).

Если открыть контекстное меню на библиотеке с экспортированными сущностями и если щелкнуть пункт *Показать объекты*, откроется список всех общедоступных сущностей библиотеки, по двойному щелчку по элементам откроется окно соответствующих свойств открыт. Нередатируемые свойства представлены как доступные только для чтения.

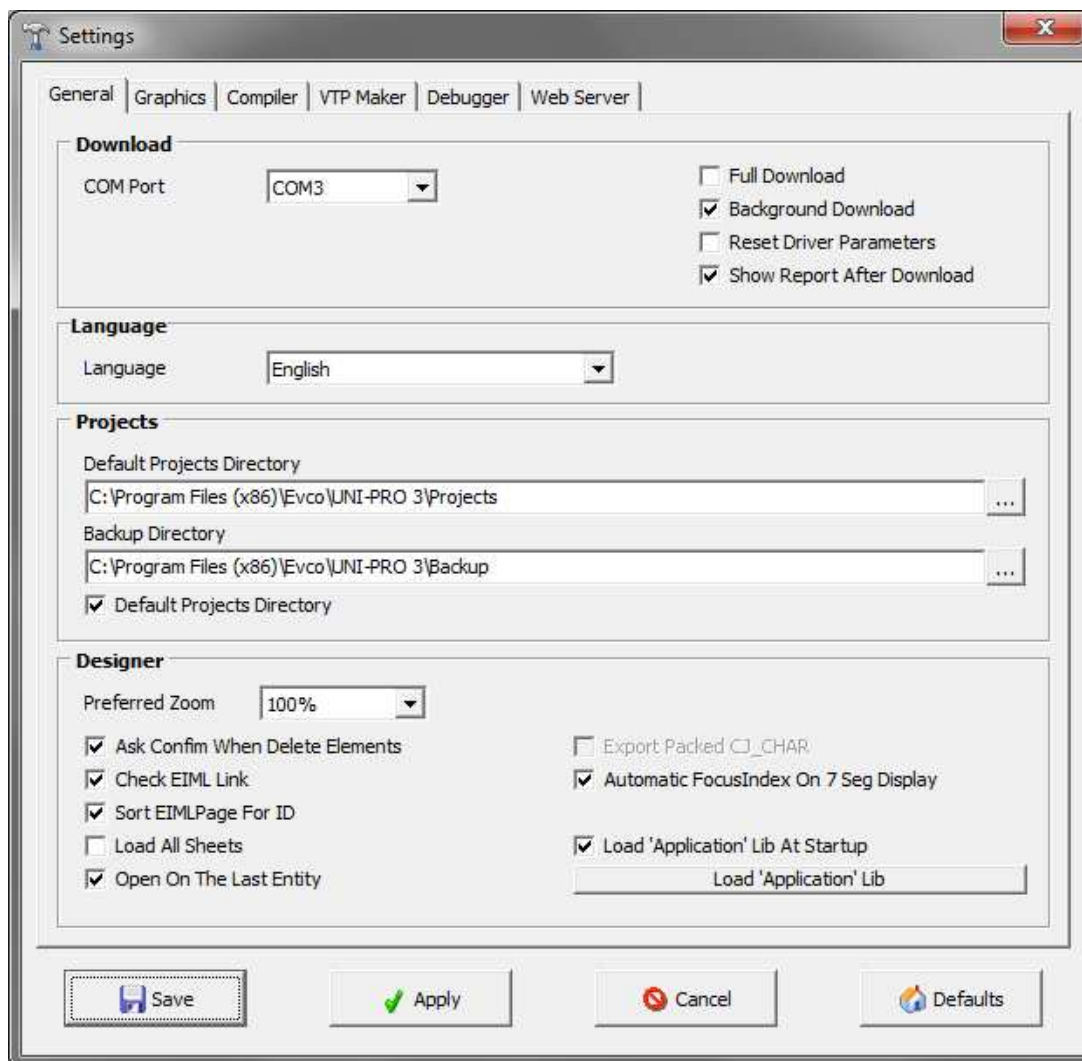
## 4.18 Резервная копия проекта

Среда программирования UNI-PRO имеет автоматическую функцию создания резервной копии, которая позволяет копировать файлы проекта в специальную директорию, указанную в опциях среды.

Задавая некоторые опции, находящиеся в окне *Settings* (пункт меню *Tools/Settings*), можно создать резервную копию вышеупомянутой директории и выбрать, увеличивать ли автоматически значение редакции после выполнения этой операции.

Директория, в которой создается резервная копия:

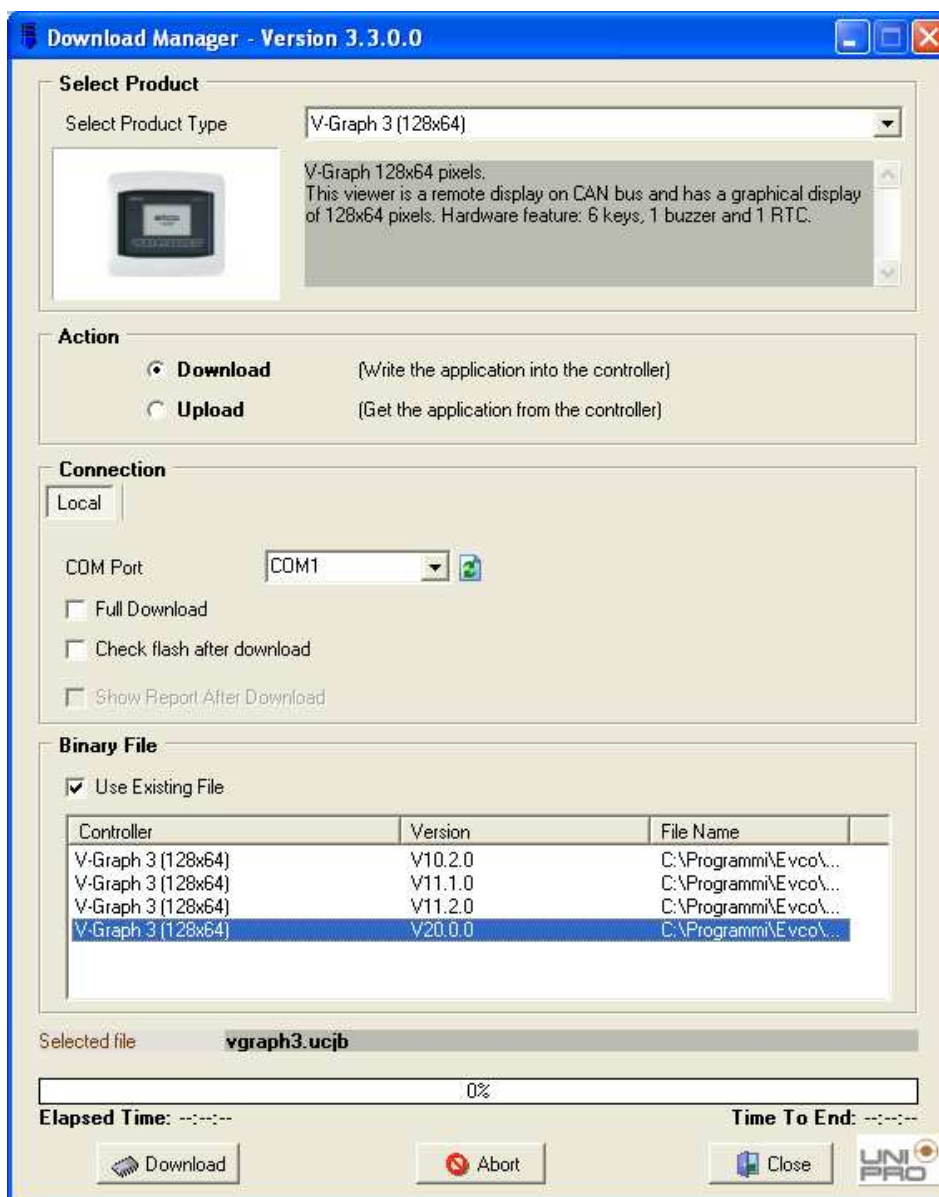
<Default directory>\Prj\_<project number >\Ver.\_<project version>.<project revision>





### 4.19 Программные средства продукта (*Download Manager*)

Среда программирования имеет средство программирования и обновления всех устройств (контроллеров и устройств визуализации). Чтобы просмотреть его, выберите в меню *Tools/Download products*.



*Download Manager* позволяет как программирование контроллеров (загрузку), так и запись и извлечение встроенного ПО в двоичном формате в присоединённый контроллер (выгрузка).

Программа состоит из одного окна, разделенного на четыре секции:

- **Select product**: выбор контроллера для программирования или контроллера, из которого следует извлечь встроенное ПО (которое должно сочетаться с присоединённым контроллером);
- **Action**: выберите требуемое действие: загрузку (для программирования устройства) или выгрузку (для извлечения встроенного ПО);
- **Connection**: выбор последовательного порта, с помощью которого подключено устройство; если подключен контроллер s-pro 3 micro/kilo, то COM-порт автоматически распознается, а заданное Вами значение игнорируется.

- Binary File: в случае загрузки можно выбрать, использовать ли существующее встроенное ПО либо указать файл для загрузки; в случае выгрузки можно определить имя файла, в который сохранить считанное с устройства встроенное ПО.

До начала фазы загрузки рекомендуется убедиться, что программируемая система правильно подключена к последовательному порту, выбранному в секции *Connections*.

В случае контроллеров с-pro 3 micro/kilo файлов, загруженных в контроллер или выгруженных из контроллера, всегда два: двоичный файл приложения (.ucjb) и двоичный файл EIML-страницы (.ucje). В любом случае необходимо указать в программе только двоичный файл приложения (.ucjb); Download Manager также выполнит загрузку другого файла (в случае загрузки) или его извлечение (в случае выгрузки). Файл EIML-страниц будет всегда иметь то же имя, что и файл приложения (за исключением расширения).

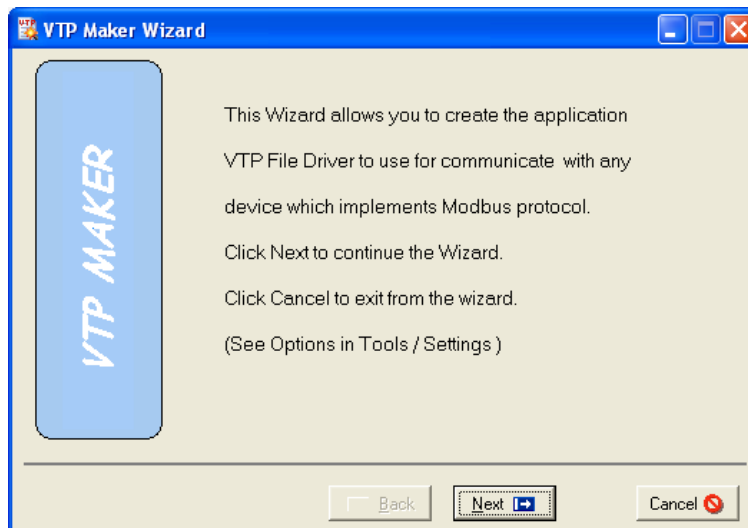
Наконец, чтобы начать загрузку, нажмите кнопку *Download*; если выбранное действие – запись встроенного ПО, нажмите кнопку *Upload*. В нижней части окна будет показано состояние процесса, а именно затраченное время (Elapsed Time), оставшееся время (Time To End) и загрузка в процентах (на индикаторе процесса).

Если загрузка/выгрузка не были завершены нужным образом, проверьте соединение с устройством, используемые конфигурации и повторите операцию.

## 4.20 VTP Maker

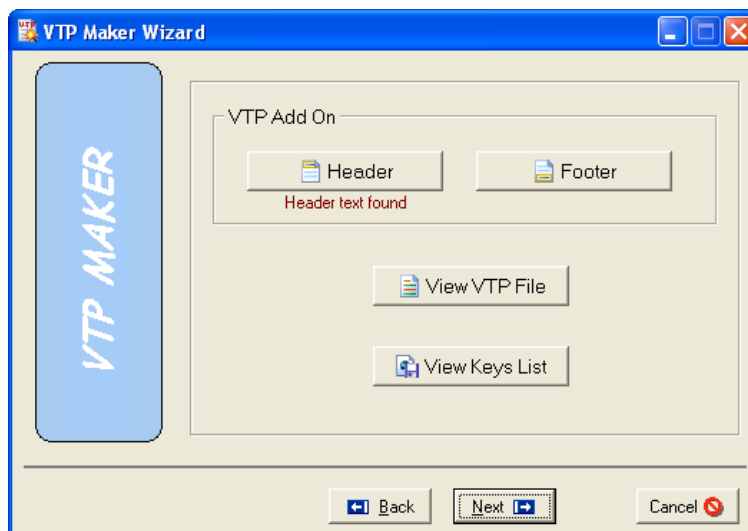
Среда программирования имеет средство, которое автоматически создает файл драйверов для связи с программным обеспечением мониторинга *R.I.C.S.* и параметрами управления программным обеспечением *Params Manager*. Необходимо, чтобы все нужные сущности были экспортированы по протоколу Modbus с помощью пункта *Export Entities* из меню *Tools/Export Entities*.

Выберите в меню *Tools/VTP Maker*.



До начала создания драйвера VTP-файла убедитесь, что опции для автоматического создателя установлены корректно в меню *Settings/VTP Maker*.

В следующем окне Вы можете выбрать ещё три функциональности:



### Header:

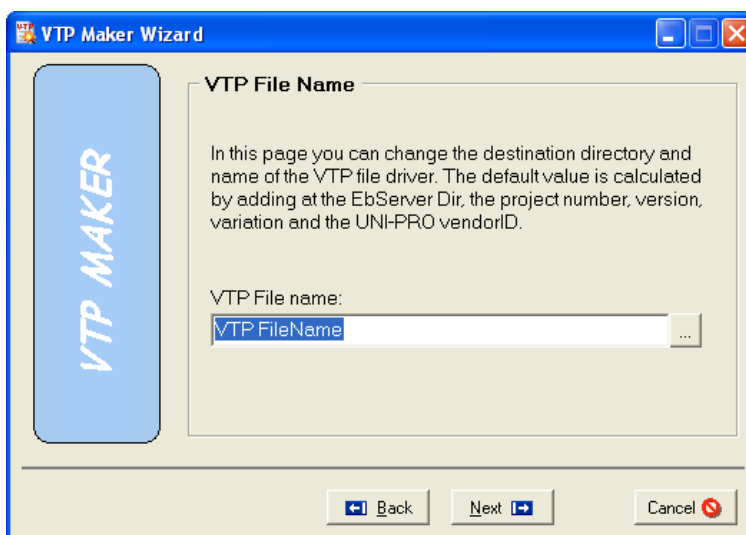
Откройте текстовый редактор для добавления комментария в начало VTP-файла. Если текст верхнего колонтитула имеется, то внизу кнопки появится сообщение “Header text found” (“Найден текст верхнего колонтитула”).

### Footer :

Откройте текстовый редактор для добавления комментария в конец VTP-файла. Если текст нижнего колонтитула имеется, то внизу кнопки появится сообщение “Header text found” (“Найден текст нижнего колонтитула”).

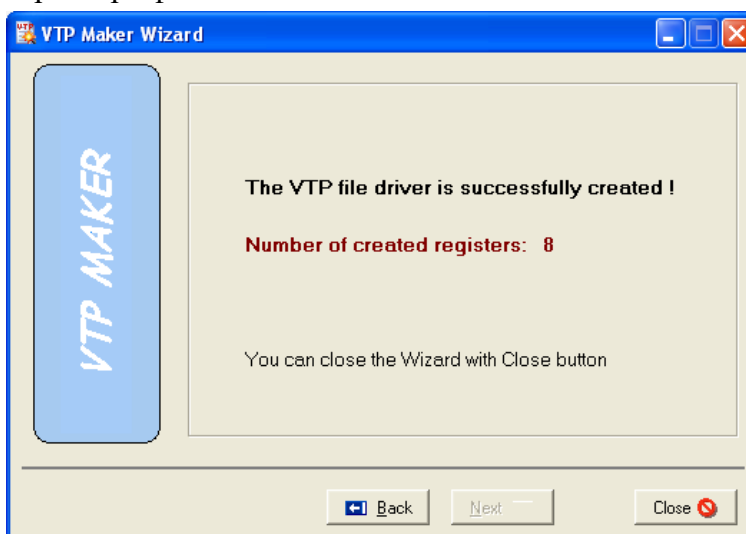
<b>View VTP File :</b>	Представляет простой предварительный просмотр VTP-файла; изменение вручную невозможно.
<b>View Keys List :</b>	Создает и предлагает сохранить HTML-файл со всеми кодами драйвера и некоторой текущей информацией по проекту.

Нажмите кнопку *Next>*, чтобы перейти к окну выбора пути и имени сохранения драйвера.



Если установлен *Parameters Manager*, то созданный VTP-файл также копируется в директорию с драйверами *Parameters Manager*. Если файл драйвера уже имеется, то пользователю будет предложено сделать копию или ручное сравнение нового и существующего файлов, чтобы возможные изменения драйвера не оказались перезаписанными.

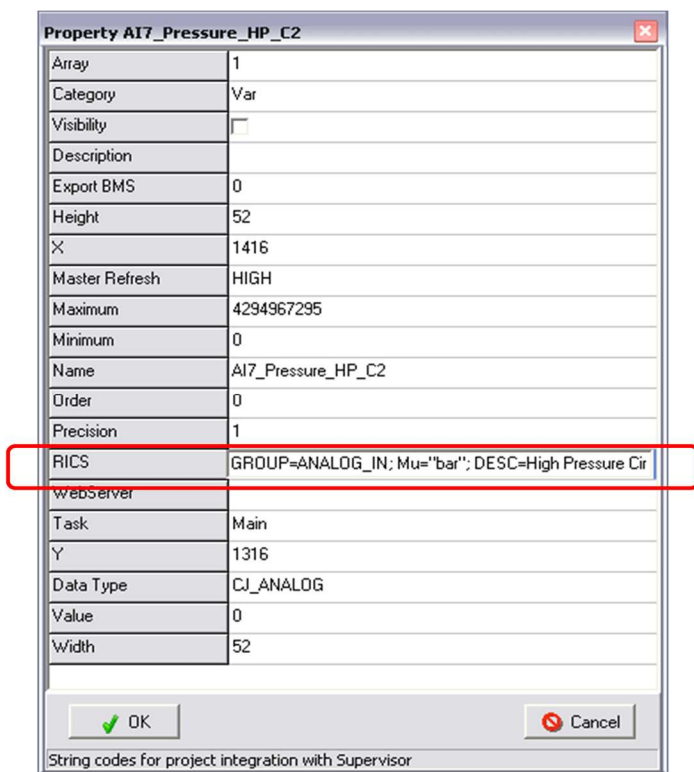
Последнее окно, при условии отсутствия ошибок при создании файла, свидетельствует об успешном создании драйвера файла VTP.



В любое время Вы можете нажать кнопку *<Back*, чтобы изменить Ваш выбор. Нажмите кнопку *Close*, чтобы завершить программу VTP Maker.

#### 4.20.1 Свойство RICS

Перед запуском инструмента VTP Maker можно выполнить некоторые настройки, чтобы настроить свойства драйвера во время автоматической генерации VTP. Эта настройка проекта возможна с использованием некоторых ключевых слов для записи в свойстве RICS:



Эти ключевые слова назначают представление таблицы, отличное от поведения по умолчанию. Например, предыдущий объект, то есть Var, должен быть помещен в столбец STATUS с именем AI7\_Pressure\_HP\_C2. После назначения *GROUP=ANALOG\_IN*; *DESC=Контур 2 высокого давления*; и *Mu="бар"*, ParametersManager отобразит его в столбце I/O с именем Контур 2 высокого давления и единицей измерения бар.

```
R_key=kyAI7_Pressure_HP_C2;
Group=ANALOG_IN;
R_var=MODBUS_HLDREG; R_addr=15;
R_for=SIGN; R_dec=1; R_do=RWW;
R_inf=-32768; R_sup=32767;
Desc="Контур 2 высокого давления";
Mu="бар";
<<<
```

Наиболее важные ключевые слова:

- **GROUP=XXXX**; Изменяет ассоциацию группы по умолчанию.
- **DESC=YYYY**; Изменяет описание (по умолчанию *Имя свойства*) на желаемое описание.
- **Mu="ZZZ"**; Установите единицу измерения. Например Mu="°C".
- **DEC=X**; Изменяет положение десятичной точки.
- **INF=XXX, SUP=YYY**; Измените пределы inf и sup (представление с фиксированной точкой)
- **EDIT=RRR**; Защищает объект от записи (только для чтения)

- **R\_flag="OFF", "ON";** Разрешает текстовое представление OFF и ON вместо числового представления 0 и 1 соответственно. Как правило, это позволяет связывать тексты и числовые значения.
- **hidden=1;** Пропускает ту запись, которая не будет отображаться в Parameters Manager.

## 4.21 Parameters Manager & VTPEditor

С помощью среды программирования UNI-PRO можно создавать и впоследствии изменять (с помощью соответствующего редактора *VTPEditor*) VTP-драйверы, используемые программой *Parameters Manager*, чтобы поддерживать связь с контроллером по протоколу связи Modbus.

Чтобы система UNI-PRO корректно работала с *Parameters Manager*, нужно, чтобы UNI-PRO обнаружила установленную программу; пожалуйста, обратите внимание на то, что несмотря на наличие установочного файла *Parameters Manager* на диске UNI-PRO, программа *Parameters Manager* не устанавливается автоматически при установке UNI-PRO и требует отдельной установки.

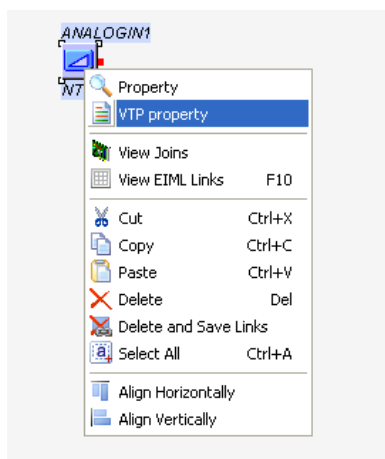
### 4.21.1 Создание и изменение файла VTP-драйвера

Для создания VTP-драйвера (VTP-драйвера, который ещё не создавали прежде) следует использовать инструмент *VTPMaker (Tools/VTPMaker Wizard)*; см. соответствующий раздел в руководстве.

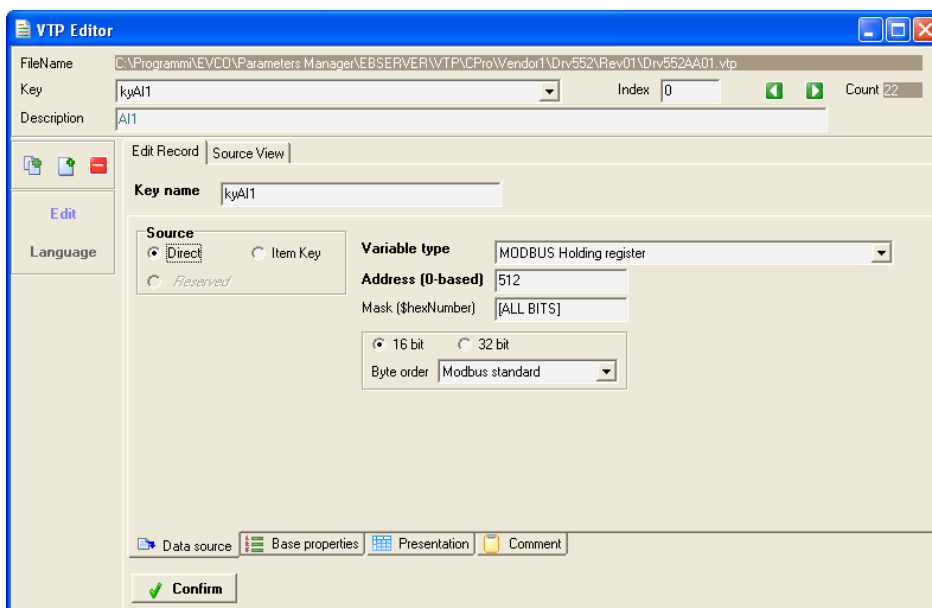
Если *Parameters Manager* установлен в системе, то сгенерированный драйвер будет также скопирован в папку с драйверами, используемыми программой. Изменения, которые через *VTPEditor* применяются к свойствам переменных драйвера, сохраняются в управляющем драйвере *Parameters Manager*.

Как только VTP-драйвер будет создан и окажется в папке драйверов *Parameters Manager*, во всплывающем меню сущности появится новый пункт (“VTP Property”).

При выборе данного пункта откроется новое окно; в этом окне можно редактировать свойства драйвера и соответствующей сущности.



Выбором *VTP Property* можно получить доступ к инструменту *VTP Editor*.

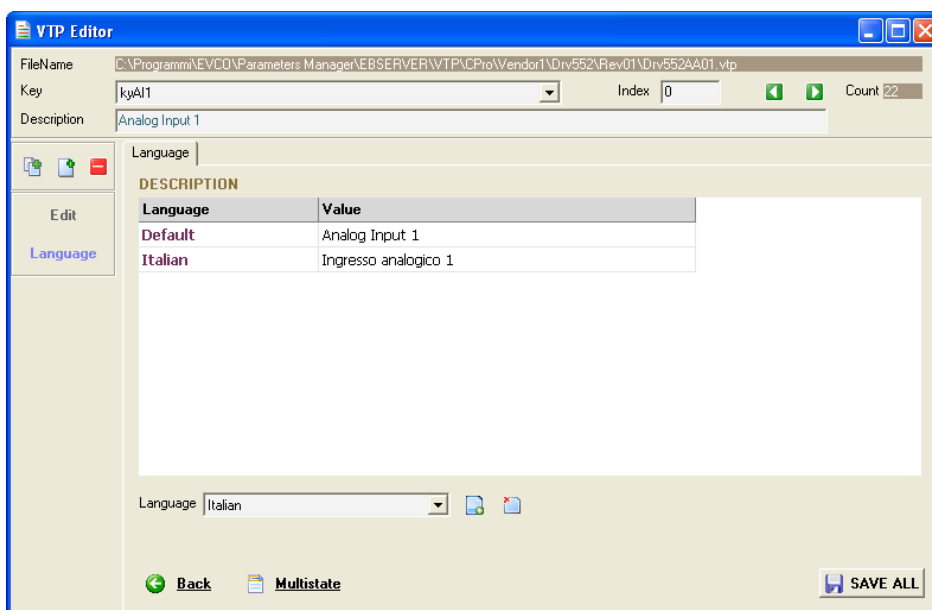


Окно редактора свойств сущности в VTP разделено на две секции: *Edit/Редактировать* и *Language/Язык*.

Секция *Edit* в свою очередь разделена ещё на четыре секции:

- *Data Source*: указывает источники сущности (адрес Modbus, номер бита для экспорта, порядок экспорта в Modbus);
- *Base properties*: указывает свойства, идентифицирующие сущность (тип переменной, минимальные и максимальные значения, единицы измерения; если переменная представляет состояние, то также значения, которые оно может принимать, и его описание);
- *Presentation*: указывает представление сущности в Parameters Manager (принадлежность к группе, стиль, уровень видимости и т.д.);
- *Comment*: комментарии, которые пользователь может добавлять к сущности.

В разделе *Language* можно вставить описание сущности (значение по умолчанию либо описание на разных языках).

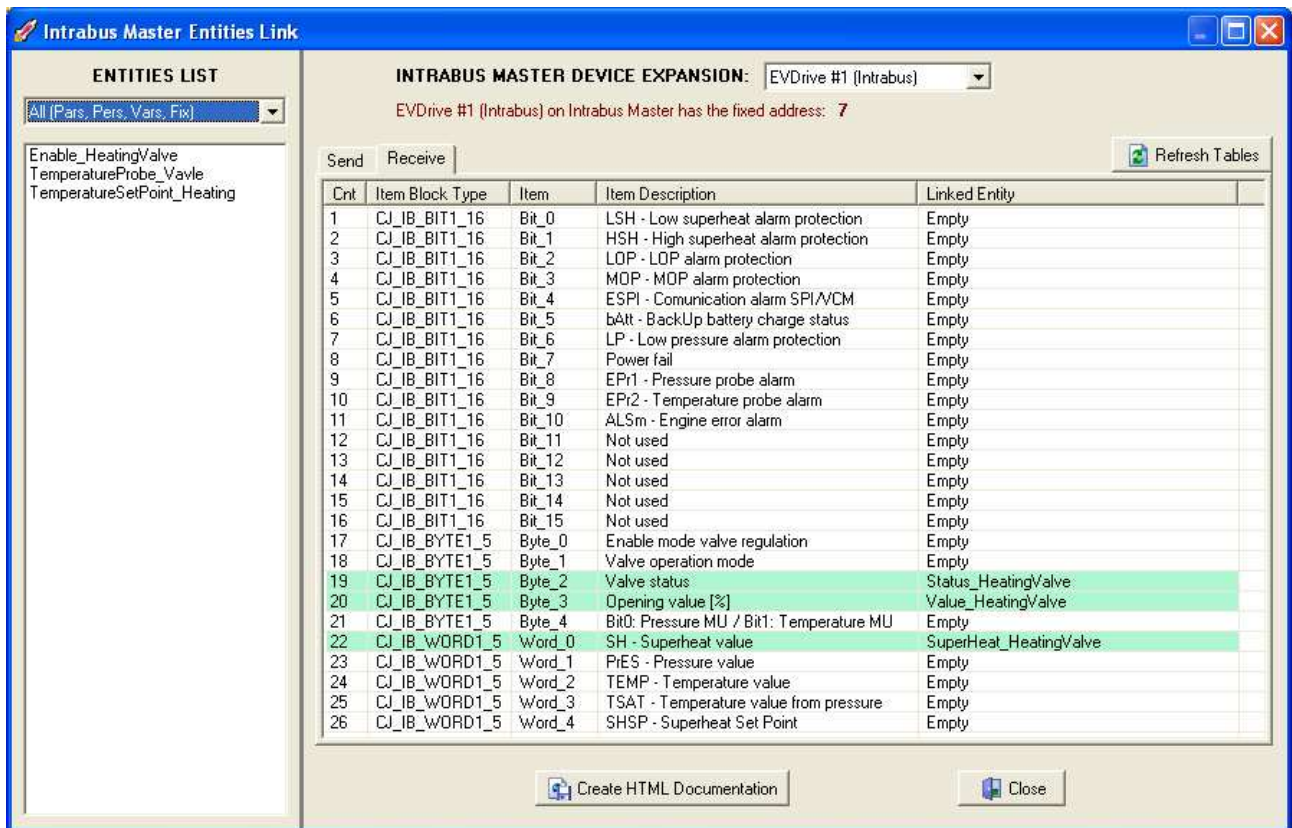


## 4.22 Расширения и конфигурация удаленных терминалов

Для реализации программного обеспечения приложения можно использовать расширения для увеличения ввода/вывода, особые драйверы для регулирующих клапанов и удалённые терминалы. Все эти элементы могут управляться главным контроллером, определяющим в проекте связи с параметрами конфигурации или внутренним состоянием с копиями в прикладном проекте с помощью специальных инструментов. В зависимости от выбранной локальной шины (Intrabus или CAN) можно вызвать соответствующий инструмент конфигурирования.

### 4.22.1 Intrabus Master

Если локальной шиной является Intrabus, то при стандартной конфигурации функции мастера выполняет контроллер, который может управлять расширениями ввода/вывода и драйверами для регулирующих клапанов (до двух штук в обоих случаях). Выбором пункта меню *Tools/Intrabus master...* можно активировать следующее окно конфигурации:



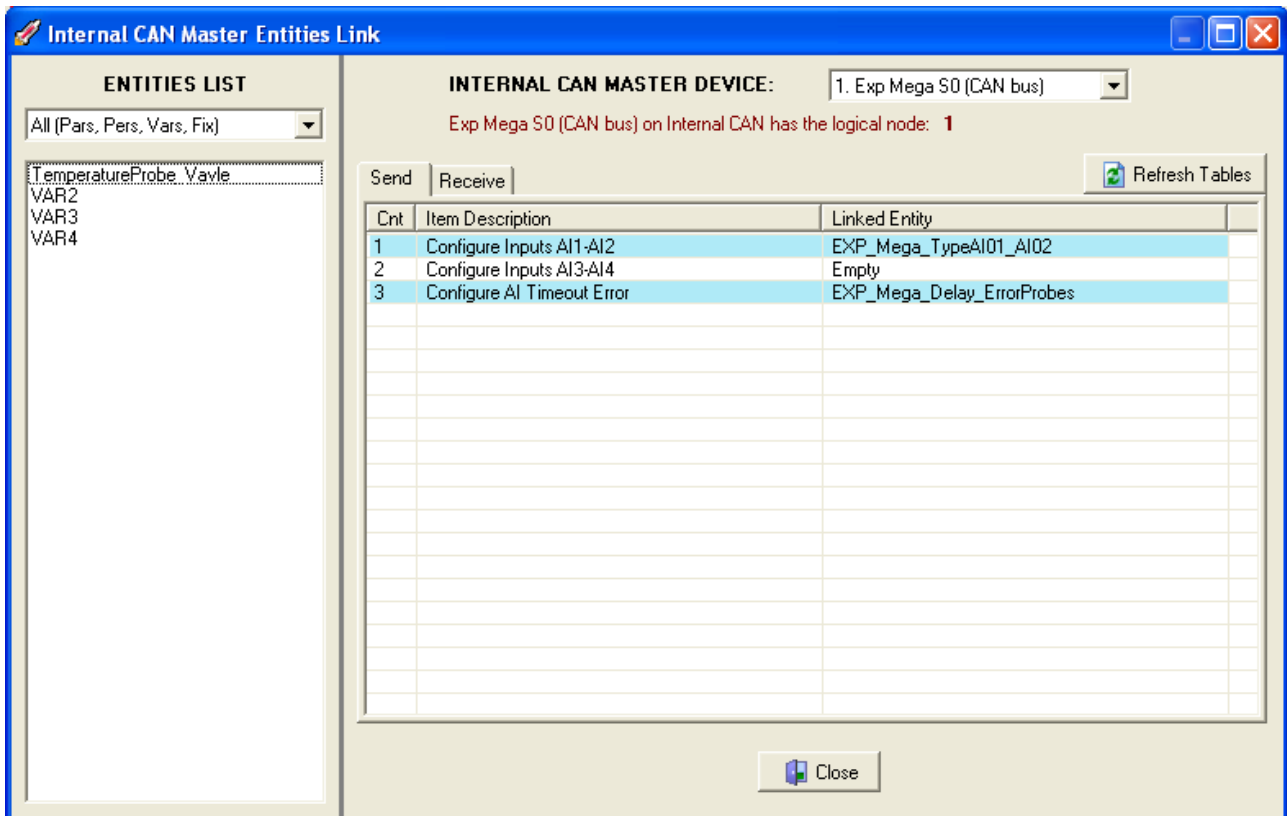
В левой части окна находятся проектные переменные, а в правой – для каждого конфигурируемого элемента – два списка (каждый на соответствующей вкладке): список конфигураций для *отправки* и список конфигурация для *получения*. Для более удобного поиска можно воспользоваться фильтром, расположенным в combo в левом верхнем углу.

Для присоединения параметра или проектной переменной к параметру или переменной сети необходимо выбрать их в левом списке и перетащить мышью на желаемую позицию в правом списке.



#### 4.22.2 CAN master

В случае, когда локальной шиной является CAN, контроллер является мастером только для этой шины, если активирован соответствующий параметр конфигурации (см. Мастер оборудования - 6 - Сеть CAN). Некоторые контроллеры могут управлять каналами CAN (внутренними и внешними, до двух штук включительно). Выбором пункта меню *Tools/CAN master...* и соответствующего канала можно активировать следующее окно конфигурации:



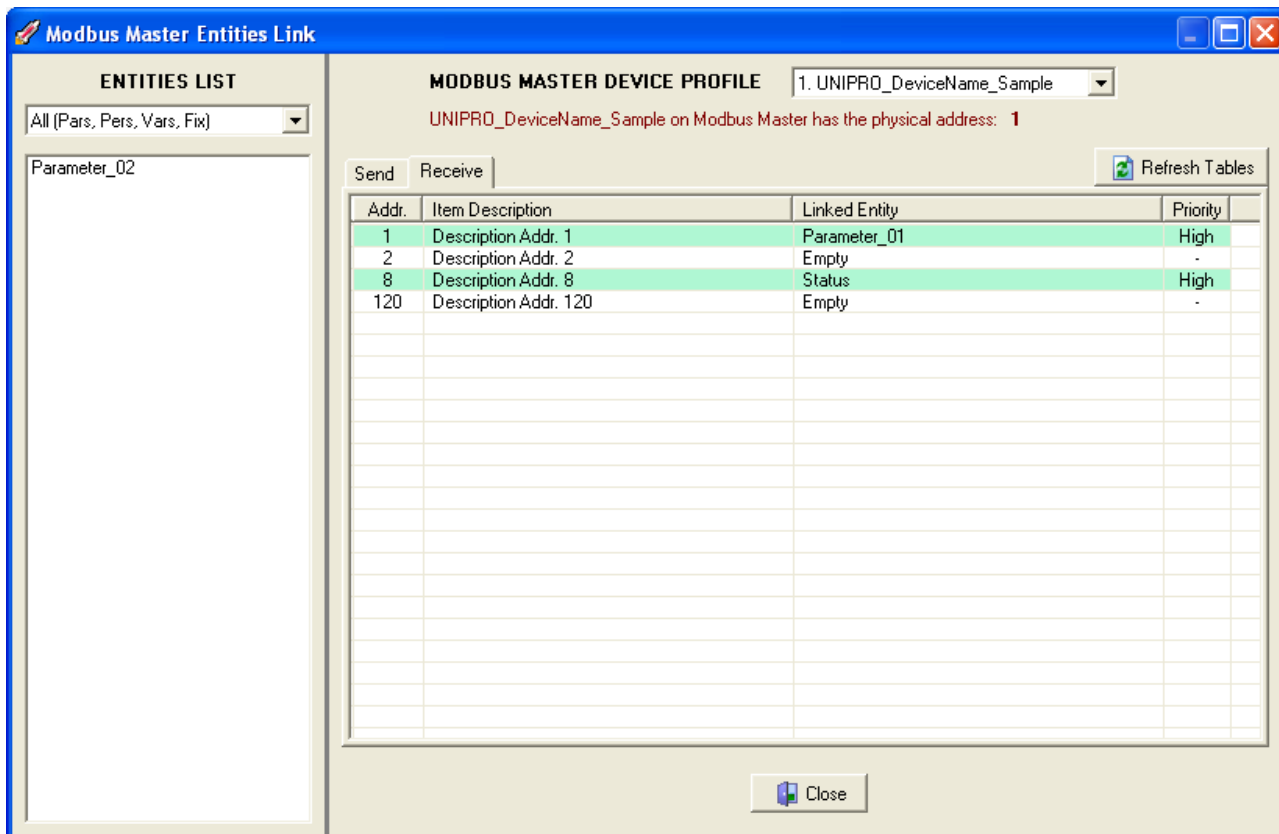
В левой части окна находятся проектные переменные, а в правой – для каждого конфигурируемого элемента – два списка (каждый на соответствующей вкладке): список конфигураций для *отправки* и список конфигурация для *получения*. Для более удобного поиска можно воспользоваться фильтром, расположенным в combo в левом верхнем углу.

Для присоединения параметра или проектной переменной к параметру или переменной сети необходимо выбрать их в левом списке и перетащить мышью на желаемую позицию в правом списке.

### 4.23 Соединение сущностей в Modbus Master

Проектные сущности, состояния и управляющие элементы для сети Modbus могут быть отправлены или получены с помощью этого инструмента, что позволяет управляющему контроллеру корректно сообщаться с присоединенными устройствами.

Выбором пункта меню *Tools/Modbus Master...* можно активировать следующее окно конфигурации:



В левой части окна находятся проектные переменные, а в правой – для каждого конфигурируемого элемента – два списка (каждый на соответствующей вкладке): список конфигураций для *отправки* и список конфигураций для *получения*. Для более удобного поиска можно воспользоваться фильтром, расположенным в combo в левом верхнем углу.

Для присоединения параметра или проектной переменной к параметру или переменной сети необходимо выбрать их в левом списке и перетащить мышью на желаемую позицию в правом списке.

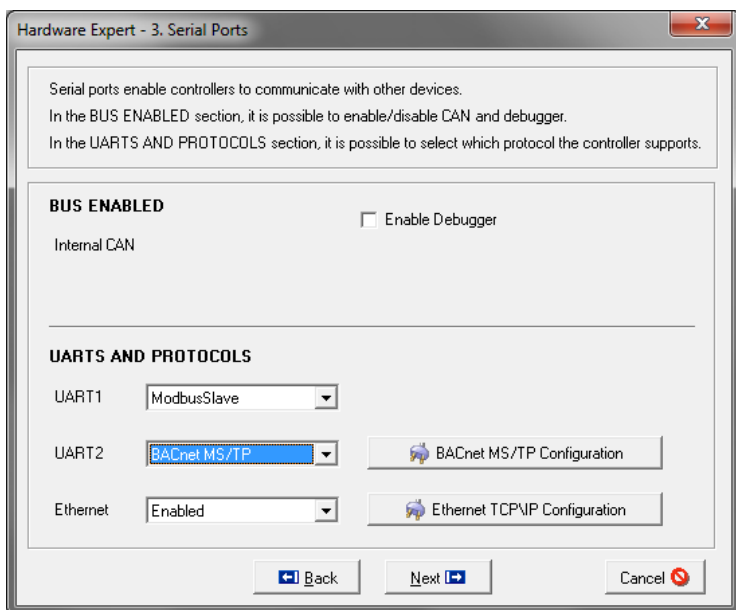
Столбец *Priority* соответствует свойству *masterRefresh* присоединенных сущностей.

## 4.24 BACnet

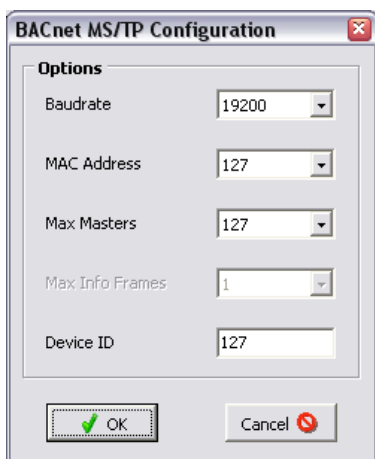
В проектах, где выбран протокол BACnet, вы можете определить объекты, для которых вы хотите выполнять мониторинг в сети BACnet MS/TP или IP-сети, в зависимости от выбора в эксперте по оборудованию. Профиль будет создан с помощью конфигуратора в UNI-PRO и сохранен во внешний файл проекта.

### 4.24.1 BACnet MS/TP конфигурация

Если вы хотите сконфигурировать контроллер с BACnet MS/TP, вы должны выбрать этот протокол в эксперте по оборудованию из выпадающего списка UART.



Нажав кнопку *BACnet MS/TP Configuration*, можно настроить основные коммуникационные свойства протокола:



Свойства:

- **BaudRate** установите скорость связи, используя выпадающее меню. Поддерживаемая скорость передачи для BACnet MS/TP 9600, 19200, 38400 и 76800. По умолчанию = 19200.
- **MAC Address** определяет физический адрес (1..127) в локальной сети BACnet. По умолчанию = 127.
- **Max Master** указывает максимально допустимый адрес для главных узлов в сети, чтобы оптимизировать процедуру передачи токена. По умолчанию = 127.

- **Max Info Frames** указывает максимальное количество кадров для каждого сообщения сессии. По умолчанию = 1.
- **Device ID** введите значение экземпляра устройства (1..4194303). Это свойство является числовым кодом, который используется для идентификации объекта «Устройство»: идентификатор объекта должен быть уникальным для всей сети. По умолчанию = 127.

#### 4.24.2 BACnet IP конфигурация

Если вы хотите настроить контроллер с использованием BACnet IP, вам сначала необходимо включить порт Ethernet в мастере аппаратного обеспечения, а затем выбрать этот протокол в параметрах, доступных для этого порта:

На вкладке BACnet IP можно настроить основные коммуникационные свойства протокола:

- **Port Number** указывает порт, который устройство будет прослушивать. Порт BACnet по умолчанию 47808 (или BAC0 в шестнадцатеричном формате)
- **Device ID** введите значение экземпляра устройства (1..4194303). Это свойство представляет собой числовой код, который используется для идентификации объекта «Устройство»: идентификатор объекта должен быть уникальным для всей сети. По умолчанию = 127.
- **BBMD Address IP** указывает IP-адрес BACnet Broadcast Management Device (BBMD). Как локальные, так и внешние устройства BACnet используют этот адрес для регистрации в BBMD, чтобы иметь возможность отправлять и получать широковещательные сообщения BACnet.
- **BBMD Port Number** указывает порт, через который устройство будет транслировать. По умолчанию BACnet порт 47808 (или BAC0 в шестнадцатеричном формате).
- **BBMD Time To Live** определяет продолжительность времени в секундах, в течение которого BBMD пересылает широковещательные сообщения. По умолчанию = 300 сек.

#### 4.24.3 BACnet файл профиля

Для проектов, использующих протокол BACnet, будут сохранены файлы с расширением \*.upbr, содержащие всю информацию профиля для созданных объектов. Файл будет сохраняться каждый раз, когда вы сохраняете проект, и будет храниться на том же уровне, что и файл проекта ucjr (например, «ProjectName\_X.ucjr») в свободном файле.

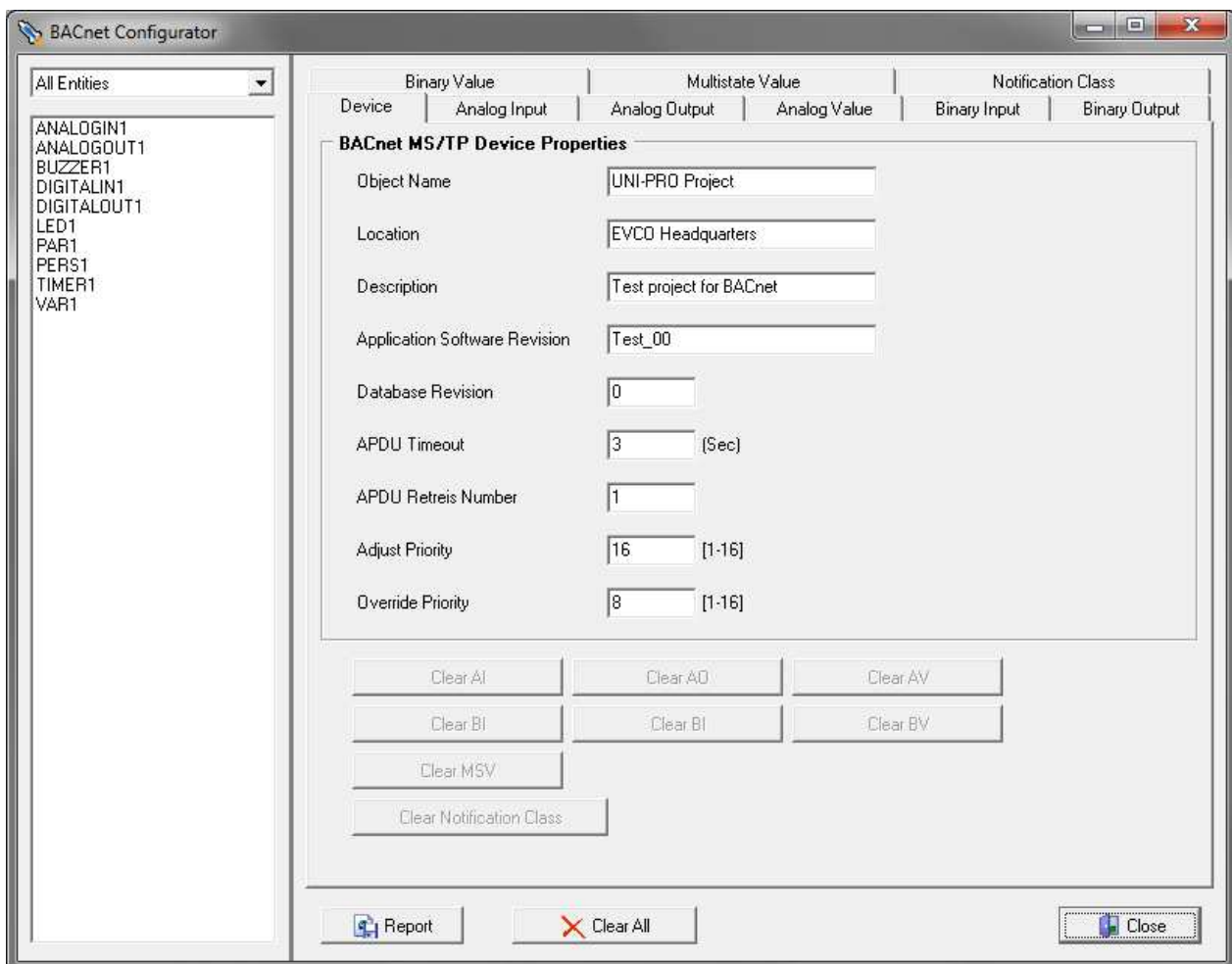
Создается папка (например, "\BACnet\_ProjectName\_X\ProjectName\_X\_Profile.upbr").

Имя профиля приложения, а также имя файлов папки не следует изменять. Если вы переименовываете файл проекта ucjr, также необходимо переименовать папку и файл профиля таким же образом.

При открытии проекта UNI-PRO загрузит информацию из файла профиля BACnet, если файл отсутствует, будет сообщено об ошибке. Во избежание неконтролируемой потери данных, если во время загрузки профиля возникает ошибка, закройте проект без сохранения и переместите файл профиля BACnet в указанный путь поиска.

#### 4.24.4 BACnet конфигуратор

Через меню *Tools*, выбрав пункт *BACnet Configurator*, мы можем открыть конфигуратор объектов BACnet; каждый тип объекта представлен в своем разделе.



В разделе Устройство вы можете установить идентифицирующие свойства контроллера в сети.

Базовые объекты, такие как Аналоговый вход, Аналоговый выход, Аналоговое значение, Цифровой вход, Цифровой выход, Бинарное значение и Значение с несколькими состояниями, создаются путем объединения объекта UNI-PRO с использованием перетаскивания объекта в соответствующем списке объектов. Ассоциация <entities, объект BACnet> разрешен только в том случае, если объект «представлен» относительным объектом BACnet в соответствии со следующими правилами:

**Analog Input (AI)**, объекты:

- Аналоговый вход
- Объекты с типом CJ\_ANALOG (исключая Fix).

**Analog Output (AO)**, объекты:

- Аналоговый выход
- Таймер
- Объекты с типом CJ\_WORD (исключая Fix).

**Analog Value (AV)**, объекты:

- Таймер
- Объекты с типом CJ\_ANALOG, CJ\_LONG, CJ\_DWORD, CJ\_SHORT, CJ\_WORD, CJ\_CHAR, CJ\_S\_BYTE и CJ\_BYTE (исключая Fix).

**Binary Input (BI)**, объекты:

- Цифровой вход
- Объекты с типом CJ\_BIT (исключая Fix).

**Binary Output (BO)**, объекты:

- Цифровой выход
- Объекты с типом CJ\_BIT (исключая Fix).

**Binary Value (BV)**, объекты:

- Объекты с типом CJ\_BIT (исключая Fix).

**Multi State Value (MSV)**, для этих объектов максимальное количество состояний 16. Объекты:

- Звуковой сигнал
- Индикатор
- Объекты с типом CJ\_BUZZER, CJ\_LED, CJ\_BIT (исключая Fix).
- Объекты с типом CJ\_BYTE с максимальными свойствами  $\leq 15$ .

После того, как вы добавили элементы, вы можете изменить их свойства; чтобы применить изменения, вы должны подтвердить, нажав кнопку *Задать свойства*. Чтобы удалить элемент, просто удалите ссылку обратным перетаскиванием или действием удаления (кнопкой или меню).

Раздел **Notification Class** определяет объекты для уведомления о событии, эти объекты не связаны с какой-либо сущностью, поэтому создание и удаление связано с явным действием добавления и удаления с помощью кнопки (или меню). Создав объекты, вы можете изменить их свойства, чтобы применить изменения, вы должны подтвердить, нажав кнопку *Задать свойства*.

Следуйте далее после задания свойств для каждого описанного объекта VASnet.

## 4.24.5 Свойства объекта устройства

**Object Name** - определяет имя объекта устройства BACnet. Это свойство представляет собой строку печатаемых символов, которая должна представлять имя объекта, уникальное для всей сети.

**Location** - это свойство представляет собой строку печатных символов, указывающую физическое расположение устройства BACnet.

**Description** - определяет описание объекта устройства BACnet. Это свойство представляет собой строку печатных символов, которая может использоваться для описания приложения, выполняемого устройством BACnet, или другой описательной информации, требуемой локально.

**Application Software Revision** - определяет версию прикладного программного обеспечения, установленного на машине. Содержимое этой строки является локальным вопросом, но это может быть отметка даты и времени, имя программиста, номер версии хост-файла и т. д.

**Database Revision** - это свойство типа Unsigned является логическим номером версии базы данных устройства.

**APDU Timeout** - этот параметр представляет, сколько времени клиент ожидает ответа от Устройства. По умолчанию 3 с (3000 мс).

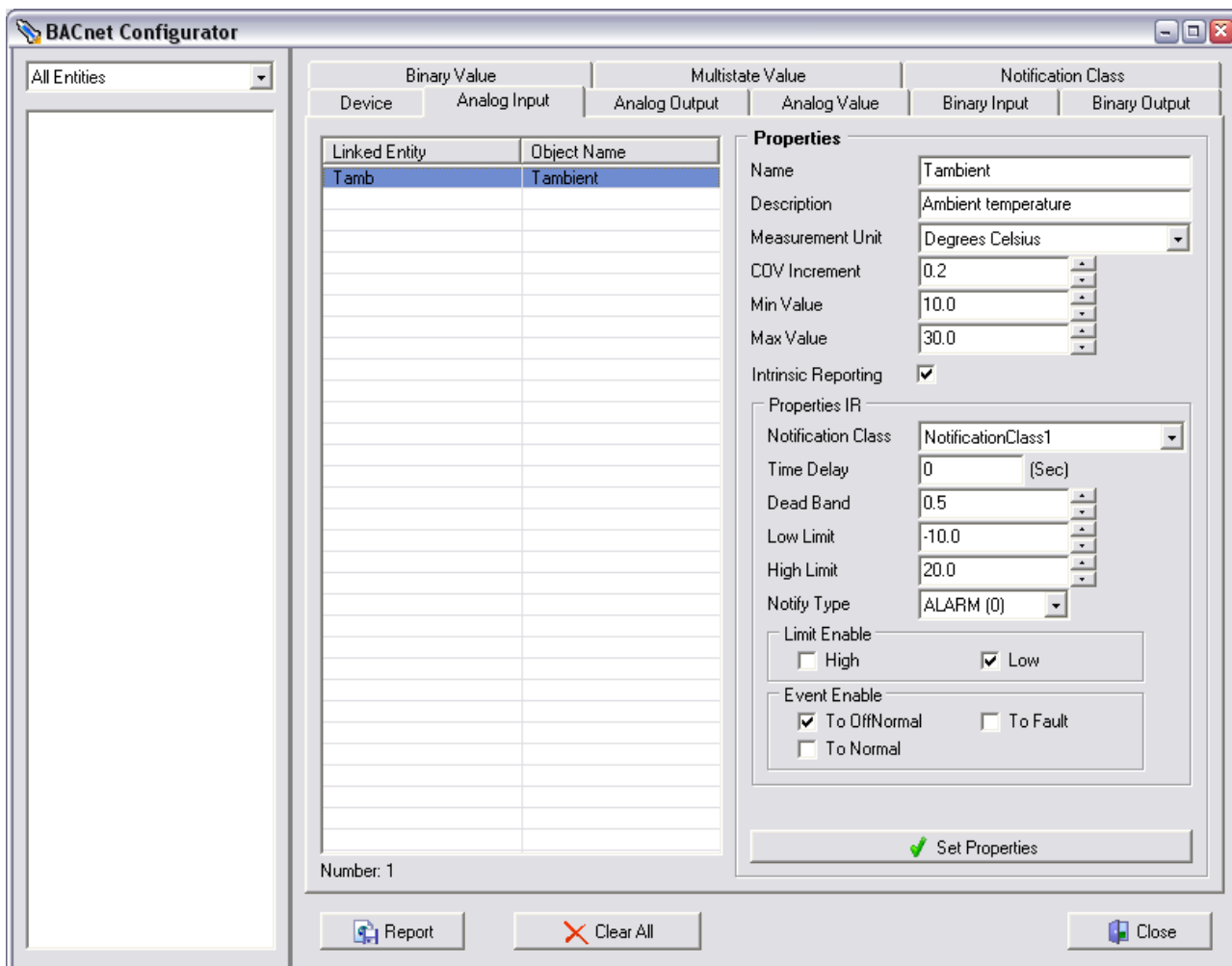
**APDU Retries Number** - этот параметр представляет значение свойства Число APDU попыток объекта Устройства от узла.

**Reboot Password** Служба Сброса Устройства используется пользователем-клиентом BACnet, чтобы дать указание удаленному устройству перезагрузить себя (холодный запуск), сбросить себя до некоторого предопределенного начального состояния (горячий запуск). Из-за конфиденциального характера этой службы от клиента BACnet-пользователя перед выполнением услуги требуется пароль. Этот пароль должен совпадать с паролем перезагрузки в контроллере.

**Adjust Priority** - BACnet Priority Array — это особый тип свойств, описанный стандартом как один из механизмов приоритезации команд. Параметр Adjust Priority определяет, с каким приоритетом контроллер будет задавать значения AV / BV / MV. По умолчанию = 16.

**Override Priority** BACnet Priority Array — это особый тип свойств, описанный стандартом как один из механизмов приоритезации команд. Параметр Override priority определяет, с каким приоритетом контроллер будет задавать значения AO/VO/МО. По умолчанию = 8.

#### 4.24.6 Свойства объекта Аналогового Входа



**Object Name** Он определяет имя объекта BACnet. Это свойство представляет собой строку печатных символов, которая должна представлять имя объекта, уникальное в пределах поддерживающего его устройства BACnet.

**Description** Он определяет описание объекта BACnet. Это свойство представляет собой строку печатных символов, содержимое которой не ограничено.

**Measurement Unit** Это свойство указывает единицы измерения этого объекта. Вы можете выбрать его из раскрывающегося списка технических единиц, определенных настоящим стандартом.

**COV Increment** Это свойство должно указывать минимальное изменение в Текущем Значении, которое приведет к выдаче COVNotification клиентам-подписчикам COV.

**Min Value** Это свойство указывает наименьшее число в единицах измерения, которое можно надежно получить для свойства Текущее Значение этого объекта.



**Max Value** Это свойство указывает максимальное число в единицах измерения, которое может быть получено для свойства Текущее Значение этого объекта.

**Intrinsic Reporting** Это свойство указывает, поддерживает ли объект встроенную отчетность, будет (ИСТИНА) или нет (ЛОЖЬ) объект учитываться службами событий и аварийных сигналов.

**Notification Class (\*)** Это свойство должно указывать экземпляр объекта Notification Class для использования рассылки уведомлений о событиях.

**Time Delay (\*)** Это свойство является временной задержкой в секундах для события объекта.

**Dead Band (\*)** Это свойство является значением мертвой зоны для событий объекта.

**Low Limit (\*)** Это свойство является нижним предельным значением для событий объекта.

**High Limit (\*)** Это свойство является верхним предельным значением для событий объекта.

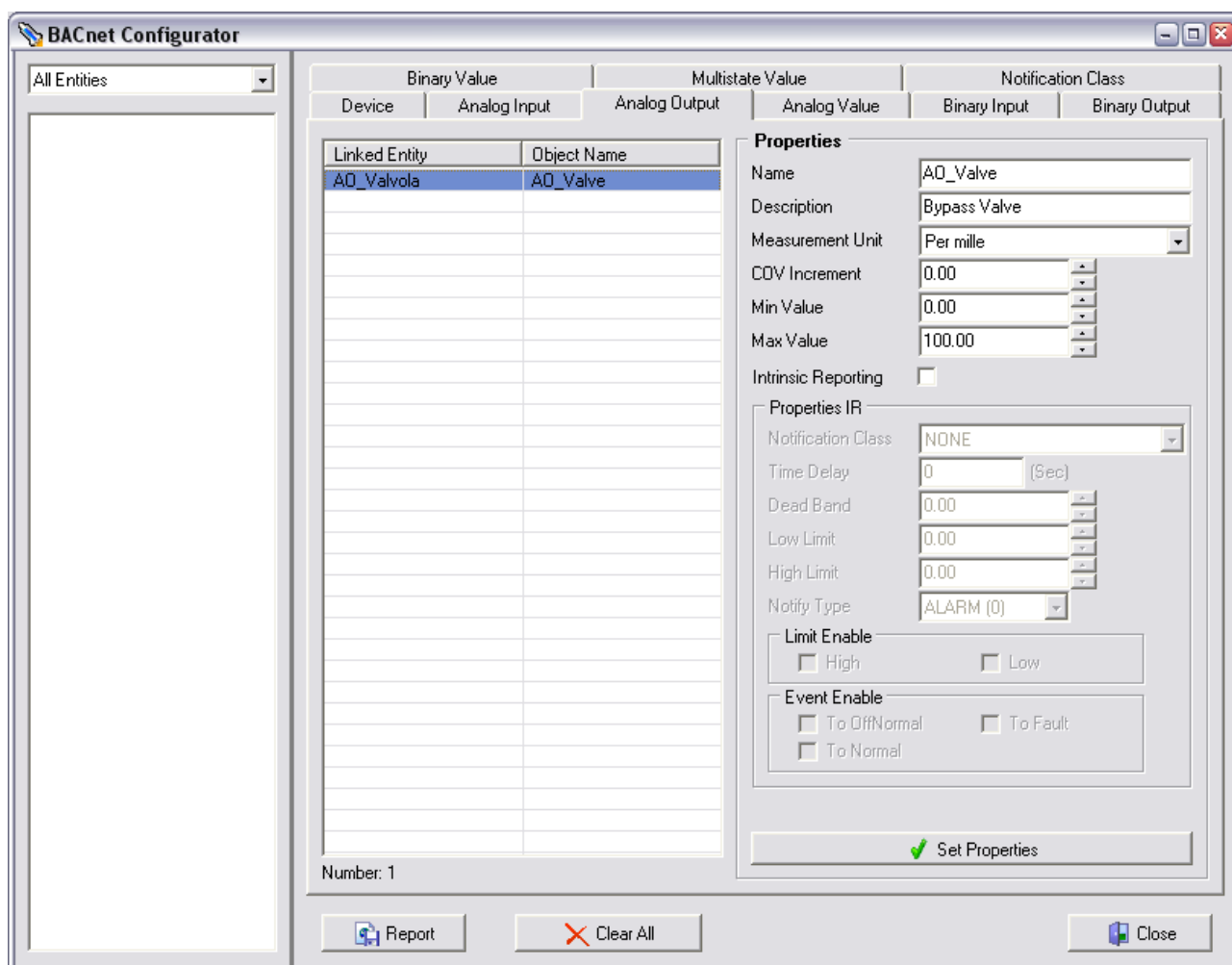
**Notify Type (\*)** Это свойство должно указывать, должны ли уведомления, генерируемые объектом, быть событиями или тревогами. Значение свойства ALARM(0) или EVENT(1) используется в качестве значения служебного параметра Notify Type в уведомлениях о событиях.

**Limit Enable (\*)** Это свойство указывает, какие ограничения учитываются для событий объекта.

**Event Enable (\*)** Это свойство должно передавать три флага, которые отдельно включают и отключают рассылку уведомлений TO\_OFFNORMAL, TO\_FAULT и TO\_NORMAL.

(\*) Свойство активно только в том случае, если для **Intrinsic Reporting** установлено значение TRUE.

#### 4.24.7 Свойства объекта Аналогового Выхода



**Object Name** Он определяет имя объекта ВАСnet. Это свойство представляет собой строку печатных символов, которая должна представлять имя объекта, уникальное в пределах поддерживающего его устройства ВАСnet.

**Description** Он определяет описание объекта ВАСnet. Это свойство представляет собой строку печатных символов, содержимое которой не ограничено.

**Measurement Unit** Это свойство указывает единицы измерения этого объекта. Вы можете выбрать его из раскрывающегося списка технических единиц, определенных настоящим стандартом.

**COV Increment** Это свойство должно указывать минимальное изменение в Текущем Значении, которое приведет к выдаче COVNotification клиентам-подписчикам COV.

**Min Value** Это свойство указывает наименьшее число в единицах измерения, которое можно надежно получить для свойства Текущее Значение этого объекта.

**Max Value** Это свойство указывает максимальное число в единицах измерения, которое может быть получено для свойства Текущее Значение этого объекта.

**Intrinsic Reporting** Это свойство указывает, поддерживает ли объект встроенную отчетность, будет (ИСТИНА) или нет (ЛОЖЬ) объект учитываться службами событий и аварийных сигналов.

**Notification Class (\*)** Это свойство должно указывать экземпляр объекта Notification Class для использования рассылки уведомлений о событиях.

**Time Delay (\*)** Это свойство является временной задержкой в секундах для события объекта.

**Dead Band (\*)** Это свойство является значением мертвой зоны для событий объекта.

**Low Limit (\*)** Это свойство является нижним предельным значением для событий объекта.

**High Limit (\*)** Это свойство является верхним предельным значением для событий объекта.

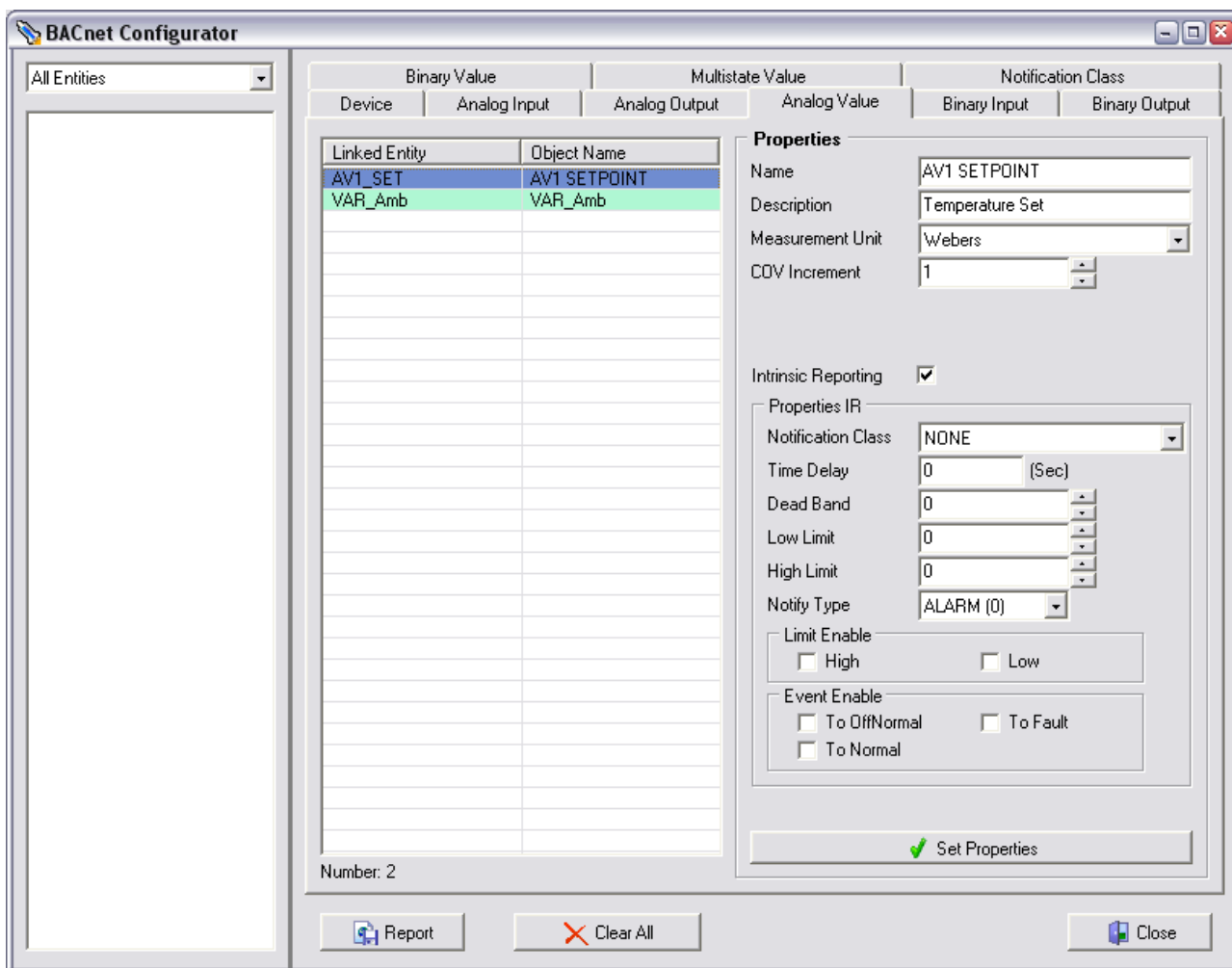
**Notify Type (\*)** Это свойство должно указывать, должны ли уведомления, генерируемые объектом, быть событиями или тревогами. Значение свойства ALARM(0) или EVENT(1) используется в качестве значения служебного параметра Notify Type в уведомлениях о событиях.

**Limit Enable (\*)** Это свойство указывает, какие ограничения учитываются для событий объекта.

**Event Enable (\*)** Это свойство должно передавать три флага, которые отдельно включают и отключают рассылку уведомлений TO\_OFFNORMAL, TO\_FAULT и TO\_NORMAL.

(\*) Свойство активно только в том случае, если для **Intrinsic Reporting** установлено значение TRUE.

## 4.24.8 Свойство объекта Аналоговое Значение



**Object Name** Он определяет имя объекта BACnet. Это свойство представляет собой строку печатных символов, которая должна представлять имя объекта, уникальное в пределах поддерживающего его устройства BACnet.

**Description** Он определяет описание объекта BACnet. Это свойство представляет собой строку печатных символов, содержимое которой не ограничено.

**Measurement Unit** Это свойство указывает единицы измерения этого объекта. Вы можете выбрать его из раскрывающегося списка технических единиц, определенных настоящим стандартом.

**COV Increment** Это свойство должно указывать минимальное изменение в Текущем Значении, которое приведет к выдаче COVNotification клиентам-подписчикам COV.

**Intrinsic Reporting** Это свойство указывает, поддерживает ли объект встроенную отчетность, будет (ИСТИНА) или нет (ЛОЖЬ) объект учитываться службами событий и аварийных сигналов.

**Notification Class (\*)** Это свойство должно указывать экземпляр объекта Notification Class для использования рассылки уведомлений о событиях.

**Time Delay (\*)** Это свойство является временной задержкой в секундах для события объекта.

**Dead Band (\*)** Это свойство является значением мертвой зоны для событий объекта.

**Low Limit (\*)** Это свойство является нижним предельным значением для событий объекта.

**High Limit (\*)** Это свойство является верхним предельным значением для событий объекта.

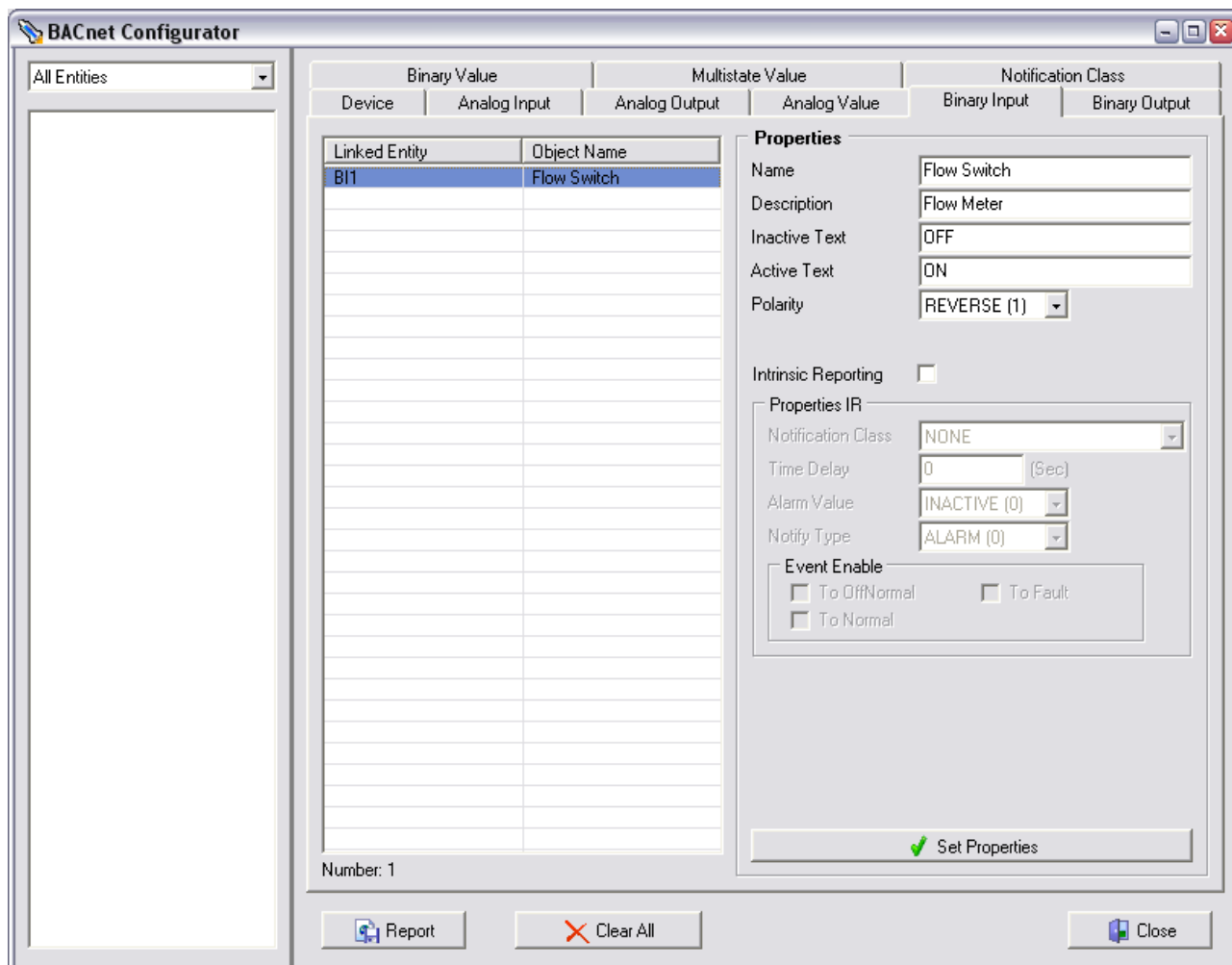
**Notify Type (\*)** Это свойство должно указывать, должны ли уведомления, генерируемые объектом, быть событиями или тревогами. Значение свойства ALARM(0) или EVENT(1) используется в качестве значения служебного параметра Notify Type в уведомлениях о событиях.

**Limit Enable (\*)** Это свойство указывает, какие ограничения учитываются для событий объекта.

**Event Enable (\*)** Это свойство должно передавать три флага, которые отдельно включают и отключают рассылку уведомлений TO\_OFFNORMAL, TO\_FAULT и TO\_NORMAL.

(\*) Свойство активно только в том случае, если для **Intrinsic Reporting** установлено значение TRUE.

#### 4.24.9 Свойства объекта Цифровой Вход



**Object Name** Он определяет имя объекта BACnet. Это свойство представляет собой строку печатных символов, которая должна представлять имя объекта, уникальное в пределах поддерживающего его устройства BACnet.

**Description** Он определяет описание объекта BACnet. Это свойство представляет собой строку печатных символов, содержимое которой не ограничено.

**Inactive Text** Это свойство описывает эффект состояния НЕАКТИВНО для Текущего Значения видимое оператору. Содержимое этой строки является локальным обозначением и предназначено для представления удобочитаемого описания состояния НЕАКТИВНО. Если присутствует свойство Inactive\_Text, либо свойство Active\_Text, то они оба должны быть активны.

**Active Text** Это свойство описывает эффект состояния АКТИВНО для Текущего Значения видимое оператору. Содержимое этой строки является локальным обозначением и предназначено для представления удобочитаемого описания состояния АКТИВНО. Если присутствует свойство Inactive\_Text, либо свойство Active\_Text, то они оба должны быть активны.

**Polarity** Это свойство указывает на взаимосвязь между физическим состоянием входа и логическим состоянием, представленным свойством Текущее Значение. Если свойство Polarity имеет значение NORMAL, то состояние ACTIVE свойства Текущее Значение является состоянием ON физического входа. Если свойство Polarity имеет значение REVERSE, тогда состояние ACTIVE свойства Текущее Значение является состоянием OFF физического входа.

**Intrinsic Reporting** Это свойство указывает, поддерживает ли объект встроенную отчетность, будет (ИСТИНА) или нет (ЛОЖЬ) объект учитываться службами событий и аварийных сигналов.

**Notification Class (\*)** Это свойство должно указывать экземпляр объекта Notification Class для использования рассылки уведомлений о событиях.

**Time Delay (\*)** Это свойство является временной задержкой в секундах для события объекта.

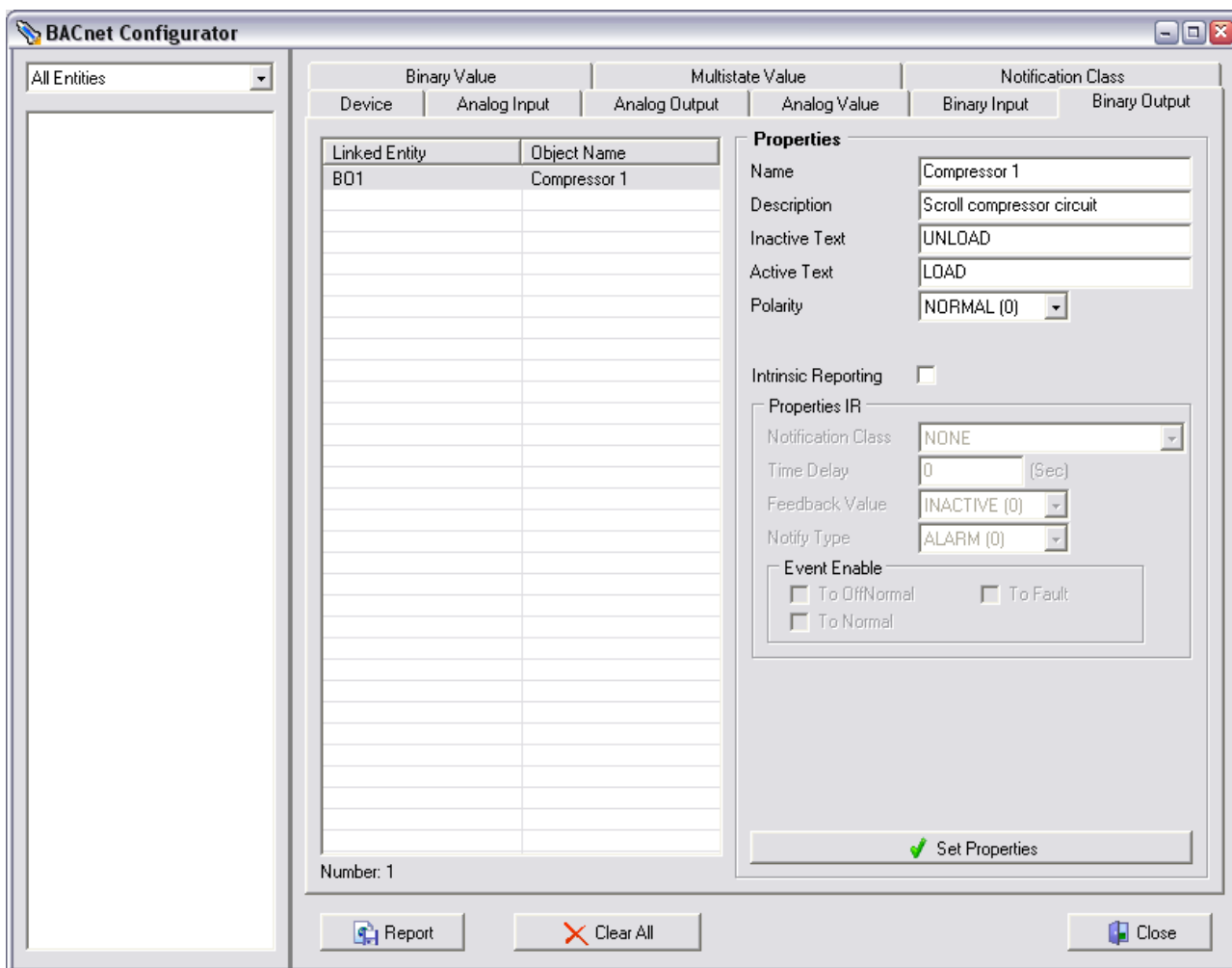
**Alarm Value (\*)** Это свойство принимает значения (ACTIVE или INACTIVE) для анализа событий объекта.

**Notify Type (\*)** Это свойство должно указывать, должны ли уведомления, генерируемые объектом, быть событиями или тревогами. Значение свойства ALARM(0) или EVENT(1) используется в качестве значения служебного параметра Notify Type в уведомлениях о событиях.

**Event Enable (\*)** Это свойство должно передавать три флага, которые отдельно включают и отключают рассылку уведомлений TO\_OFFNORMAL, TO\_FAULT и TO\_NORMAL.

(\*) Свойство активно только в том случае, если для **Intrinsic Reporting** установлено значение TRUE.

## 4.24.10 Свойства объекта Цифровой Выход



**Object Name** Он определяет имя объекта ВАСнет. Это свойство представляет собой строку печатных символов, которая должна представлять имя объекта, уникальное в пределах поддерживающего его устройства ВАСнет.

**Description** Он определяет описание объекта ВАСнет. Это свойство представляет собой строку печатных символов, содержимое которой не ограничено.

**Inactive Text** Это свойство описывает эффект состояния НЕАКТИВНО для Текущего Значения видимое оператору. Содержимое этой строки является локальным обозначением и предназначено для представления удобочитаемого описания состояния НЕАКТИВНО. Если присутствует свойство Inactive\_Text, либо свойство Active\_Text, то они оба должны быть активны.

**Active Text** Это свойство описывает эффект состояния АКТИВНО для Текущего Значения видимое оператору. Содержимое этой строки является локальным обозначением и предназначено для представления удобочитаемого описания состояния АКТИВНО. Если присутствует свойство Inactive\_Text, либо свойство Active\_Text, то они оба должны быть активны.

**Polarity** Это свойство указывает на взаимосвязь между физическим состоянием входа и логическим состоянием, представленным свойством Текущее Значение. Если свойство Polarity имеет значение NORMAL, то состояние АКТИВНО свойства Текущее Значение является состоянием ON физического входа. Если свойство Polarity имеет значение REVERSE, тогда состояние АКТИВНО свойства Текущее Значение является состоянием OFF физического входа.

**Intrinsic Reporting** Это свойство указывает, поддерживает ли объект встроенную отчетность, будет (ИСТИНА) или нет (ЛОЖЬ) объект учитываться службами событий и аварийных сигналов.

**Notification Class (\*)** Это свойство должно указывать экземпляр объекта Notification Class для использования рассылки уведомлений о событиях.

**Time Delay (\*)** Это свойство является временной задержкой в секундах для события объекта.

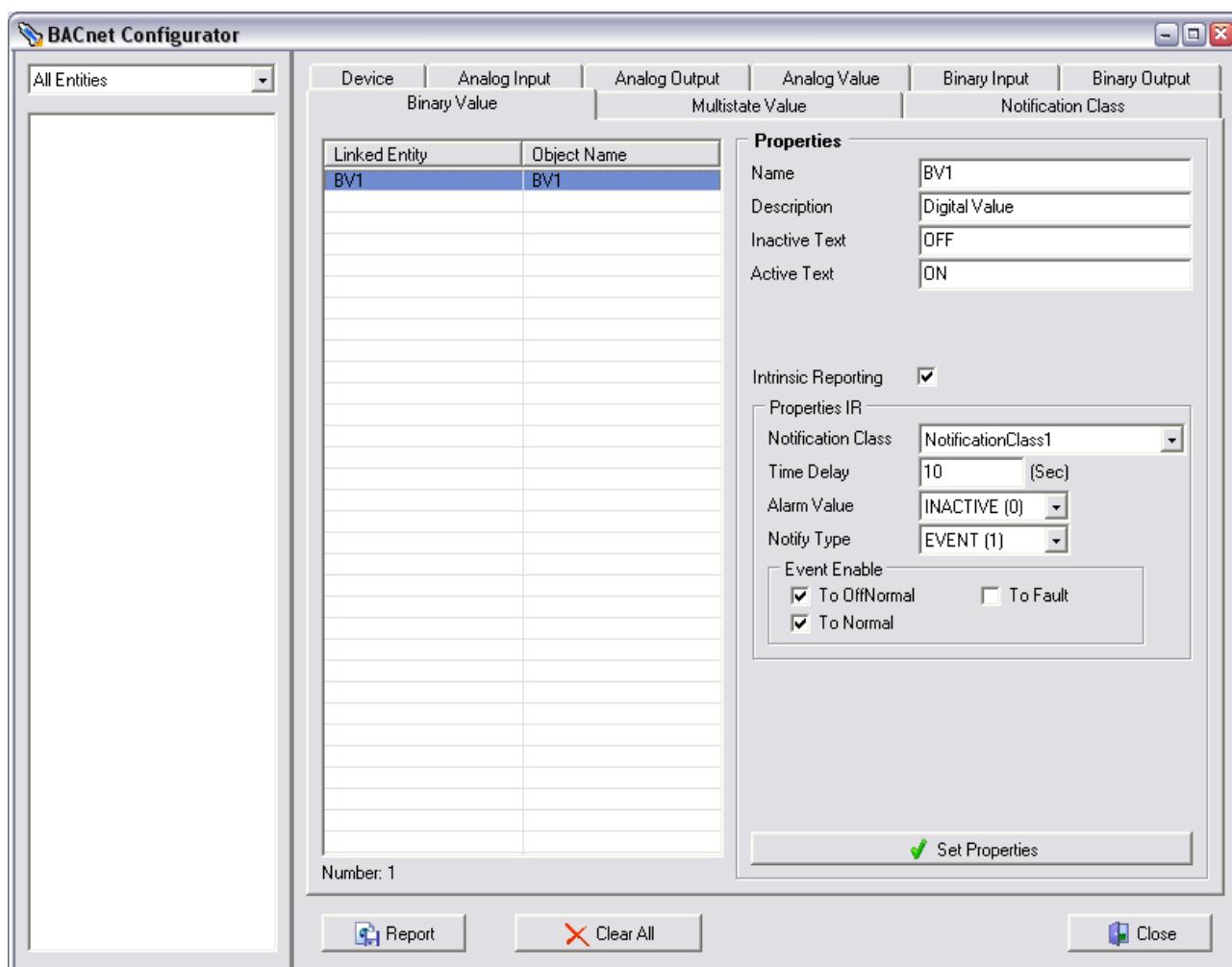
**Feedback Value (\*)** Это свойство указывает фактическое значение объекта, используемого для анализа событий объекта, если объект поддерживает отчеты о событиях.

**Notify Type (\*)** Это свойство должно указывать, должны ли уведомления, генерируемые объектом, быть событиями или тревогами. Значение свойства ALARM(0) или EVENT(1) используется в качестве значения служебного параметра Notify Type в уведомлениях о событиях.

**Event Enable (\*)** Это свойство должно передавать три флага, которые отдельно включают и отключают рассылку уведомлений TO\_OFFNORMAL, TO\_FAULT и TO\_NORMAL.

(\*) Свойство активно только в том случае, если для **Intrinsic Reporting** установлено значение TRUE.

#### 4.24.11 Свойства объекта Цифровое Значение



**Object Name** Он определяет имя объекта BACnet. Это свойство представляет собой строку печатных символов, которая должна представлять имя объекта, уникальное в пределах поддерживающего его устройства BACnet.

**Description** Он определяет описание объекта BACnet. Это свойство представляет собой строку печатных символов, содержимое которой не ограничено.

**Inactive Text** Это свойство описывает эффект состояния НЕАКТИВНО для Текущего Значения видимое оператору. Содержимое этой строки является локальным обозначением и предназначено для представления удобочитаемого описания состояния НЕАКТИВНО. Если присутствует свойство Inactive\_Text, либо свойство Active\_Text, то они оба должны быть активны.

**Active Text** Это свойство описывает эффект состояния АКТИВНО для Текущего Значения видимое оператору. Содержимое этой строки является локальным обозначением и предназначено для представления удобочитаемого описания состояния АКТИВНО. Если присутствует свойство Inactive\_Text, либо свойство Active\_Text, то они оба должны быть активны.

**Intrinsic Reporting** Это свойство указывает, поддерживает ли объект встроенную отчетность, будет (ИСТИНА) или нет (ЛОЖЬ) объект учитываться службами событий и аварийных сигналов.

**Notification Class (\*)** Это свойство должно указывать экземпляр объекта Notification Class для использования рассылки уведомлений о событиях.

**Time Delay (\*)** Это свойство является временной задержкой в секундах для события объекта.

**Alarm Value (\*)** Это свойство принимает значения (ACTIVE или INACTIVE) для анализа событий объекта.

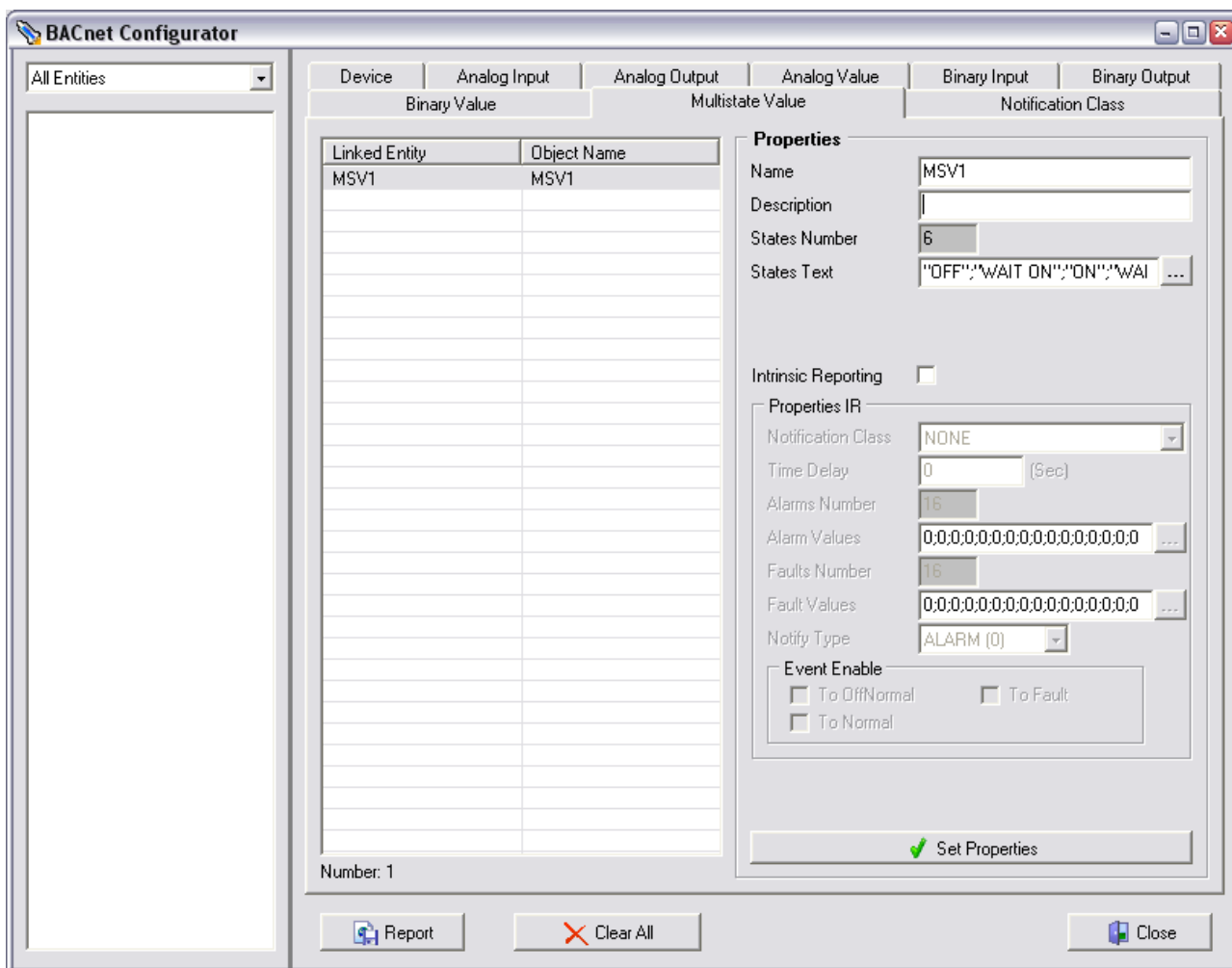
**Notify Type (\*)** Это свойство должно указывать, должны ли уведомления, генерируемые объектом, быть событиями или тревогами. Значение свойства ALARM(0) или EVENT(1) используется в качестве значения служебного параметра Notify Type в уведомлениях о событиях.

**Event Enable (\*)** Это свойство должно передавать три флага, которые отдельно включают и отключают рассылку уведомлений TO\_OFFNORMAL, TO\_FAULT и TO\_NORMAL.

(\*) Свойство активно только в том случае, если для **Intrinsic Reporting** установлено значение TRUE.



## 4.24.12 Свойства объекта Мультистатусное Значение



**Object Name** Он определяет имя объекта BACnet. Это свойство представляет собой строку печатных символов, которая должна представлять имя объекта, уникальное в пределах поддерживающего его устройства BACnet.

**Description** Он определяет описание объекта BACnet. Это свойство представляет собой строку печатных символов, содержимое которой не ограничено.

**States Number** Это свойство определяет количество состояний, которые может иметь Текущее Значение. Свойство всегда должно иметь значение больше нуля.

**States Text** Это свойство представляет собой список строк печатных символов, представляющих описания всех возможных состояний Текущего Значения.

**Intrinsic Reporting** Это свойство указывает, поддерживает ли объект встроенную отчетность, будет (ИСТИНА) или нет (ЛОЖЬ) объект учитываться службами событий и аварийных сигналов.

**Notification Class (\*)** Это свойство должно указывать экземпляр объекта Notification Class для использования рассылки уведомлений о событиях.

**Time Delay (\*)** Это свойство является временной задержкой в секундах для события объекта.

**Alarm Values (\*)** Это свойство принимает значения (ACTIVE или INACTIVE) для анализа событий объекта.

**Alarm Number** Это значение только для чтения выдается из списка значений сигналов тревоги.

**Fault Values (\*)** Это свойство представляет собой индексный список возможных состояний, представляющих состояние аварии для определения событий объекта.

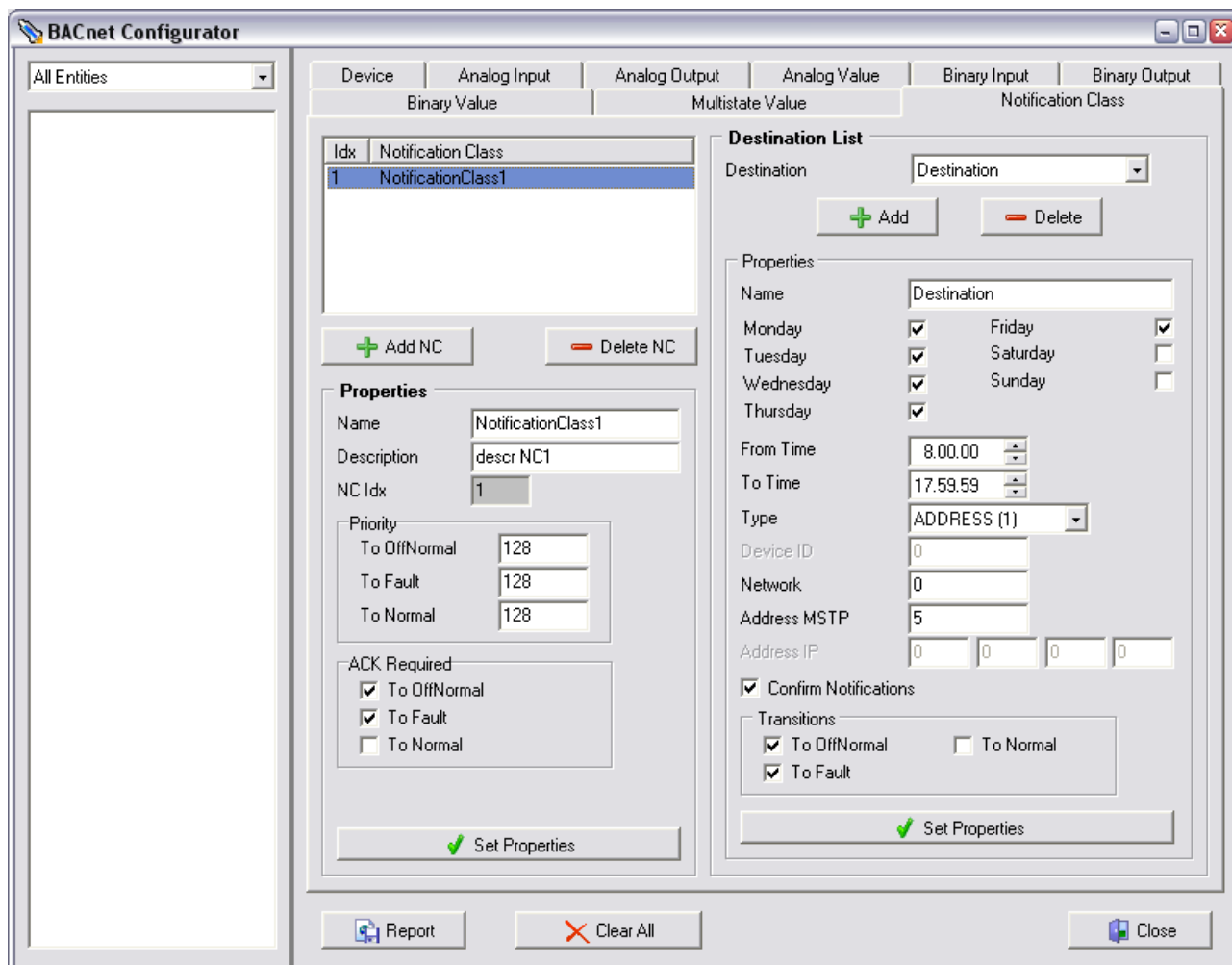
**Fault Number** Это значение только для чтения автоматически выдается из списка Fault Values.

**Notify Type (\*)** Это свойство должно указывать, должны ли уведомления, генерируемые объектом, быть событиями или тревогами. Значение свойства ALARM(0) или EVENT(1) используется в качестве значения служебного параметра Notify Type в уведомлениях о событиях.

**Event Enable (\*)** Это свойство должно передавать три флага, которые отдельно включают и отключают рассылку уведомлений TO\_OFFNORMAL, TO\_FAULT и TO\_NORMAL.

(\*) Свойство активно только в том случае, если для **Intrinsic Reporting** установлено значение TRUE.

#### 4.24.13 Свойства объекта Класса Уведомлений



Используя эту форму, можно настроить до 5 Классов Уведомлений, каждый со списком получателей, ограниченным максимум 5 пунктами назначения.

**Object Name** Он определяет имя объекта BACnet. Это свойство представляет собой строку печатных символов, которая должна представлять имя объекта, уникальное в пределах поддерживающего его устройства BACnet.

**Description** Он определяет описание объекта BACnet. Это свойство представляет собой строку печатных символов, содержимое которой не ограничено.

**NC Idx** Он представляет собой прогрессивный индекс объекта класса уведомлений.

**Priority** Это свойство, значение без знака для каждого из трех возможных событий, должно передавать приоритет, который будет использоваться для уведомлений о событиях TO\_OFFNORMAL, TO\_FAULT и TO\_NORMAL. Приоритеты должны находиться в диапазоне от 0 до 255 включительно. Меньшее число указывает на более высокий приоритет.

**Ack required** Это свойство - одно логическое условие для каждого из трех возможных событий, должно передавать три отдельных флага, которые указывают, требуется ли подтверждение в уведомлениях, генерируемых для переходов событий TO\_OFFNORMAL, TO\_FAULT и TO\_NORMAL, соответственно.

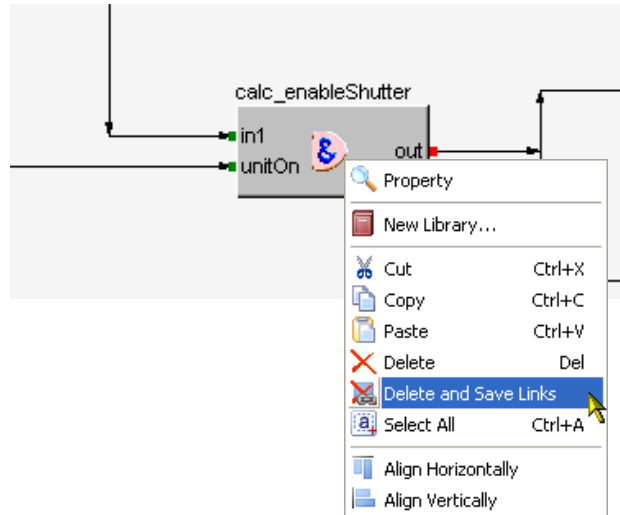
**Destination List** Это свойство определяет список из нуля или более получателей, которым должны быть отправлены уведомления, когда инициирующие событие объекты, использующие этот класс, обнаруживают возникновение события. Эти пункты назначения получателей должны быть относительно постоянными, не имеют срока действия и должны поддерживаться при сбое питания или перезагрузке устройства.

Для каждого элемента списка можно настроить:

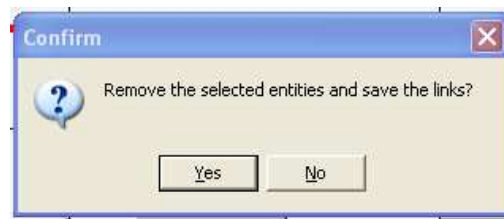
- Имя
- Задействованные дни
- Время, в течение которого адресат доступен в дни недели, указанные как допустимые дни.
- Тип получателя NETWORK (0) или ADDRESS(1)
- Device ID в случае NETWORK (0)
- ID сети и MAC адрес (BACnet MS/TP) или IP адрес (BACnet IP) в пункте ADDRESS(1)
- Флаг подтверждения уведомления: (TRUE) если должны быть отправлены подтвержденные уведомления и (FALSE), если должны быть отправлены неподтвержденные уведомления
- Переходы: набор из трех флагов, указывающих на переходы {TO\_OFFNORMAL, TO\_FAULT, TO\_NORMAL}, для которых подходит этот получатель

## 4.25 Функция удаления и сохранения связей и *LinkFixer*

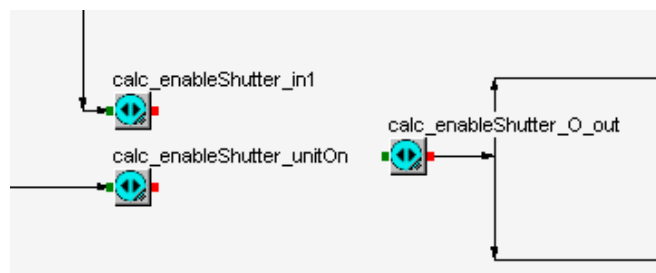
Данная функция может быть использована для удаления выбранных сущностей и для сохранения связей, при этом появляются переменные, которые «хранят» соединения (удаленной сущности) с другими сущностями проекта.



Нажмите...

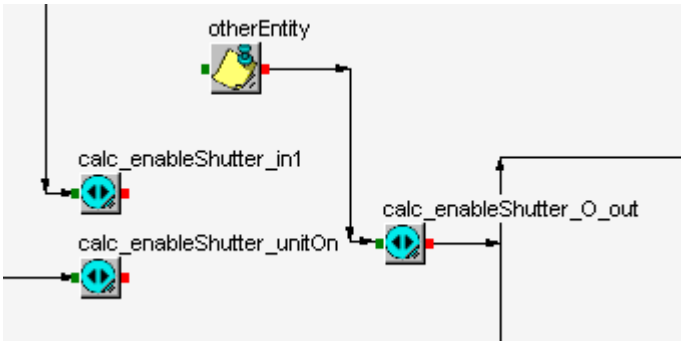


Нажмите Да ....

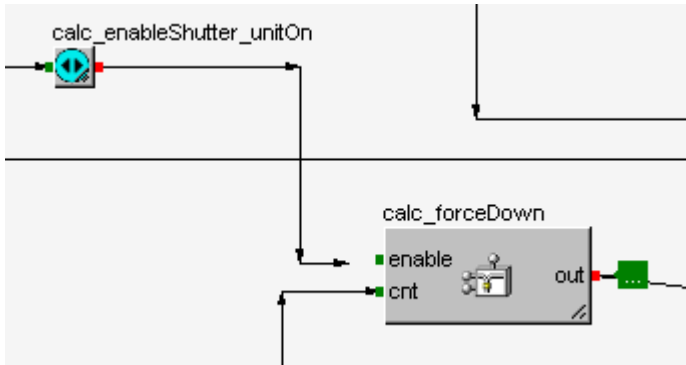


Сущность *calc\_enableShutter* была удалена, при этом были созданы три переменных (по одной для каждого входа и выхода), которые соединены с сущностью, с которой была связана *calc\_enableShutter*. Таким образом, входы/выходы были «сохранены» и могут быть обработаны и использованы повторно.

Для налаживания входов/выходов удаленных сущностей создаются специальные сущности “Link Fixer”. Эти объекты позволяют запоминать связи, и при этом они упрощены; если присоединить свободный вход или выход к другой сущности, то объект Link Fixer будет автоматически удалён, а связь автоматически присоединена к новой сущности.

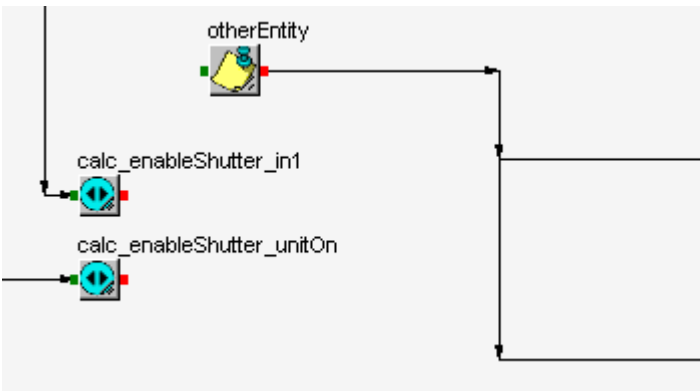


A)  
Соединение *calc\_enableShutter\_O\_out* с новой сущностью “otherEntity”.

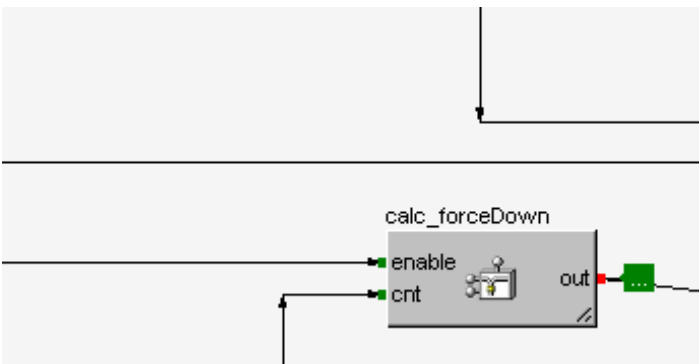


B)  
Соединение *calc\_enableShutter\_unitON* с входом “enable” алгоритма “calc\_forceDown”

Разрешение Link Fixer



A)  
Упрощение Link Fixer *calc\_enableShutter\_O\_out*; сущность “otherEntity” будет автоматически соединена с зажимами, соединенными с Link fixer, которая будет удалена.



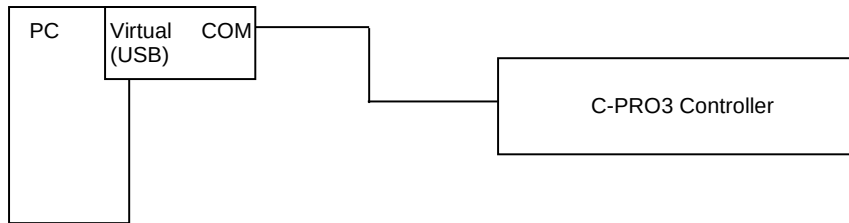
B)  
Упрощение Link Fixer *calc\_enableShutter\_unitOn*; вход “enable” алгоритма будет автоматически соединен с сущностью, соединенной с Link fixer, которая будет удалена.

4.25.1 Функция «Сохранить связь»

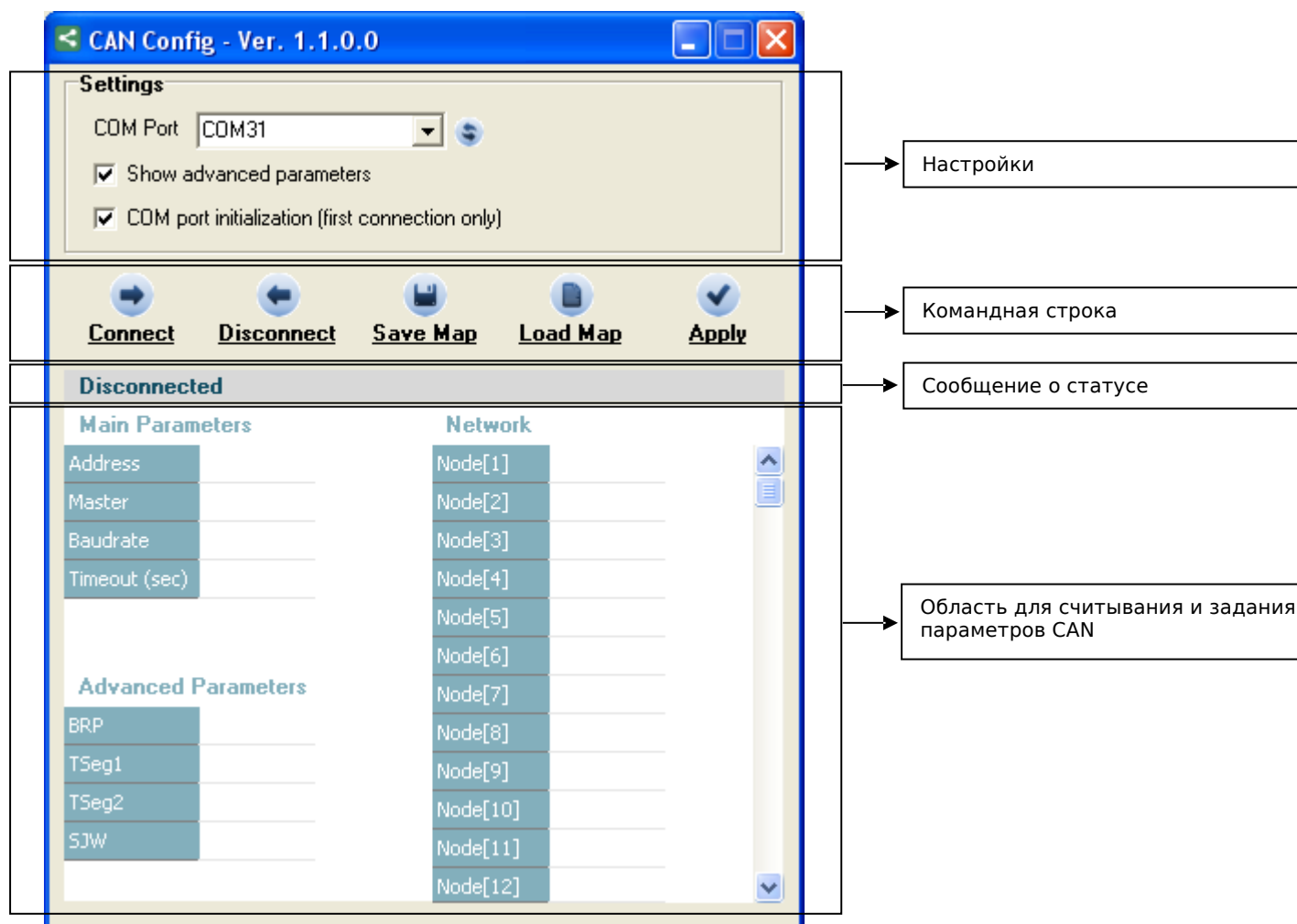
Эта функция позволяет вам «отсоединять и присоединять» связи для отдельных терминалов ввода-вывода объектов. Аналогично предыдущей функции, выбрав функцию «Сохранить связь» в контекстном меню на терминале, создается средство исправления связей, и связи удаляются из зажима и переносятся в новый только что созданный объект средства исправления связей.

## 4.26 CAN Config

CAN Config – это инструмент, позволяющий программировать параметры CANbus контроллеров, расширений и пользовательских интерфейсов, принадлежащих с-про 3. Для использования CAN Config не требуется особенное оборудование.



После выбора пункта меню *Инструменты/Конфигуратор CAN* появляется следующее окно:



### Настройки

**COM Port:** COM-порт автоматически распознается

**Show advanced parameters:** если выбрано, то параметры *побитовой* синхронизации отображаются и могут быть настроены (показаны в таблице как *Advanced Parameters*).

**COM port initialization:** если выбрано, то будет выполнена инициализация последовательного порта (при первом подключении программы к контроллеру с использованием *COM-порта*, чтобы убедиться, что соединение с контроллером не относится к неопределенному состоянию при первом подключении). Если контроллер останется подключенным к серийному порту, будет выполнена перезагрузка.

### Командная строка

**Connect:** начинает подключение к контроллеру. С этого момента можно отображать и изменять параметры коммуникации CANbus.

**Disconnect:** прекращает связь с контроллером.

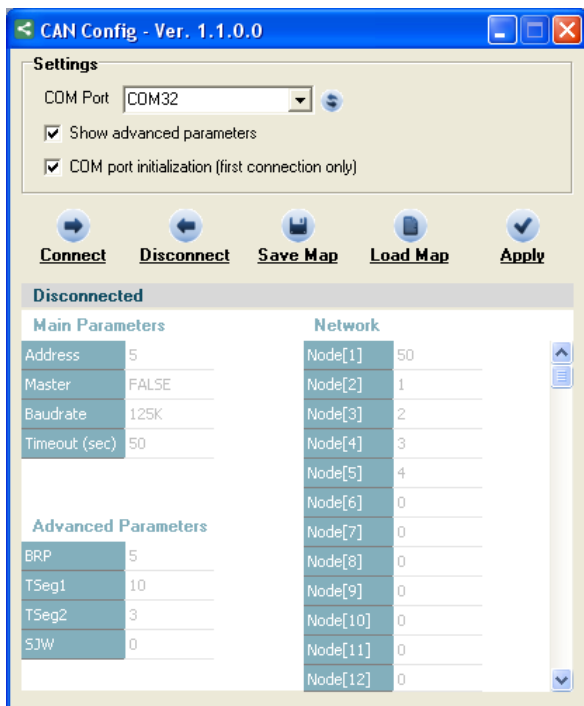
**Save Map:** сохраняет значения, находящиеся в схемах файла (схема; Основные параметры; Сеть; Расширенные параметры).

**Load Map:** загружает «схему» файла и записывает загруженные значения в контроллер.

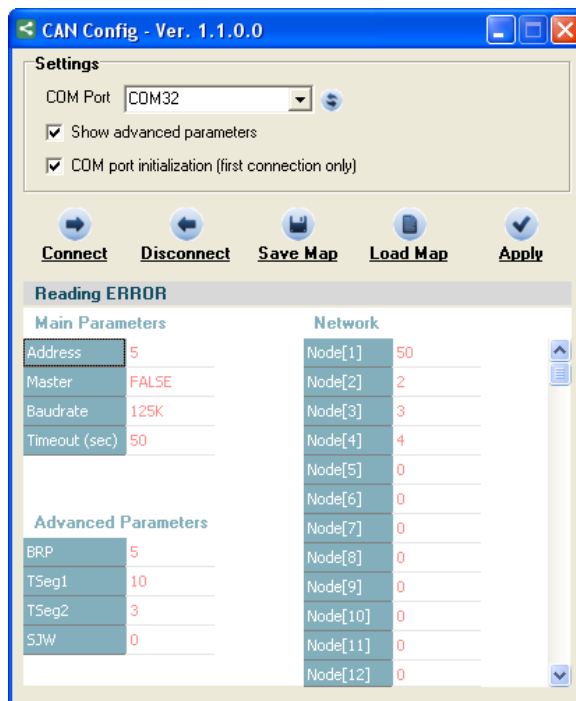
**Apply:** отправляет контроллеру команду утвердить параметры, которые были изменены.

Чтобы изменить параметры при активном соединении, достаточно записать значения в сетку. Программа обновляет значения каждую секунду и обеспечивает немедленную обратную связь (при условии, что запись была успешно завершена). Чтобы сделать только что записанные значения действительными и чтобы контроллер снова начал коммуникацию CAN с использованием новых параметров, необходимо нажать кнопку ПРИМЕНИТЬ.

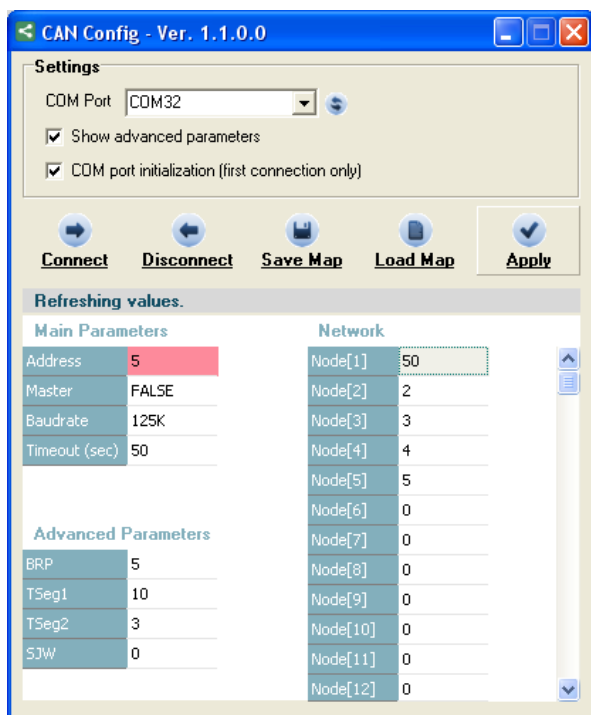
Если после проверки параметров некоторые из них перезаписываются приложением (например, с использованием библиотек, принадлежащих группе System/CAN 1st), несоответствие подчеркивается красным. Для сохранения и загрузки карт необходимо, чтобы связь была активной; никакие изменения не разрешены в автономном режиме.



Значения обозначены серым цветом при отсутствии соединения.



Значения обозначены красным цветом в случае ошибки соединения.

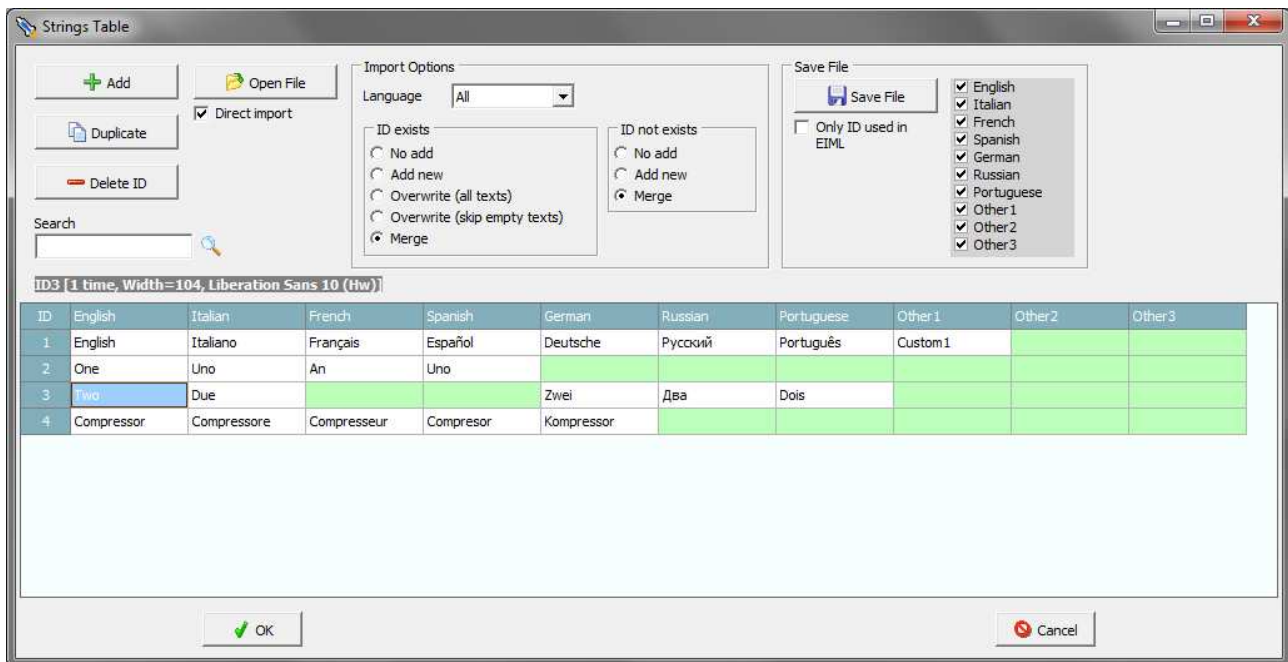


Значения, выделенные красным, отличаются от считанных значений. Управление осуществляется после подтверждения изменений с помощью команды APPLY.



## 4.27 Таблица строк

В меню *Инструменты* → *Таблица строк (Текстов)* используется инструмент, который позволяет вам создавать и манипулировать строками языковой базы данных, которые будут использоваться на страницах EIML.



Над таблицей отображается строка с полезной информацией об идентификаторе:

- Количество экземпляров идентификатора на всех страницах EIML.
- Максимальное количество отображаемых символов, когда текст связан только с элементами EIML, в которых не используются настраиваемые шрифты, или максимальный размер в пикселях, когда идентификатор связан хотя бы с одним элементом, использующим настраиваемый шрифт.
- Пользовательский шрифт большей ширины, с которым связан идентификатор.

Если текст превышает максимальное количество символов, которое может отображаться элементом EIML на странице, он выделяется красным цветом. Если элемент имеет пустой текст, он выделяется зеленым цветом, в противном случае он имеет белый фон.

**Добавить.** Добавить новый ID.

**Дублировать.** Дублировать выбранный ID.

**Удалить ID.** Удалить выбранный ID.

**Открыть файл.** Это позволяет вам открыть ранее сохраненный языковой файл и иметь возможность изменять его, в этом режиме появляется текст **РЕЖИМ ОТКРЫТИЯ ФАЙЛА**, чтобы указать, что вы работаете с внешним файлом, а не с реальным языковым содержимым проекта, будет возможность импортировать этот файл в файл проекта.

В разделе **Параметры импорта** указаны параметры импорта:

- **Язык:** импортирует только выбранный язык
- **Существующие ID:** действия, если импортируемый идентификатор, уже есть в проекте
  - Не добавлять ID.
  - Добавить новый ID (первый свободный)

- Заменяет все тексты текстами импортируемого идентификатора.
- Заменяет тексты непустыми текстами импортируемого идентификатора, оставляет исходные тексты без изменений, если текст нового идентификатора равен нулю.
- Слияние. Добавляются новые идентификаторы, совмещаясь с существующими, для каждого нового идентификатора проверяется, есть ли совместимый, если есть, новые переводы дополняются, иначе новый идентификатор добавляется как есть.
  - ID не существует: действие, которое необходимо выполнить, если идентификатор, который необходимо импортировать, отсутствует в проекте.
- Не добавляются ID.
- Добавляется новый ID.
- Слияние. Добавляются новые идентификаторы, совмещаясь с существующими, для каждого нового идентификатора проверяется, есть ли совместимый, если есть, новые переводы дополняются, иначе новый идентификатор добавляется как есть.

**Прямой импорт.** Это позволяет вам открыть файл с помощью команды *Открыть файл* и импортировать его непосредственно в соответствии с параметрами импорта, не просматривая файл, который нужно импортировать (вы не проходите через РЕЖИМ ОТКРЫТИЯ ФАЙЛА).

**Сохранить файл.** Это позволяет сохранить файл в формате CSV со всеми выбранными идентификаторами и языками (возможно, только идентификаторами, используемыми в EIML, отметив соответствующую опцию). Впоследствии файл можно импортировать в другой проект.

Через выпадающее меню правой кнопкой мыши можно выполнить такие операции:

- **Изменить ID:** изменить номер идентификатора, выбранного в таблице.
- **Удалить язык:** удалить все строки выбранного языка.
- **Удалить неиспользуемый ID:** удалите ID, не используемые на страницах EIML.
- **Удалить все:** удалить все IDs языков.
- **Заменить ID в EIML:** Заменяет на страницах EIML, использующих идентификатор, выбранный в таблице, другим идентификатором в таблице (эта операция является окончательной, после ее выполнения вернуться назад невозможно).
- **Удалить ID с EIML:** Безвозвратно удаляет связь выбранного идентификатора со всех страниц EIML, которые его используют (эта операция является окончательной, после ее выполнения вернуться назад невозможно).
- **Удалить все ID с EIML:** Безвозвратно удаляет связь всех идентификаторов со всех страниц EIML, которые их используют (эта операция является окончательной, после ее выполнения вернуться назад невозможно).

Также можно запустить инструмент со страницы EIML (из выпадающего меню на странице), таким образом будут перечислены только используемые на странице языковые идентификаторы. Действия в этом режиме ограничены.

## 4.28 Переводчик EIML

Выбором пункта меню Инструменты → Переводчик EIML можно получить доступ к инструменту, позволяющему легко справиться с многоязычными строками на EIML-страницах. На рисунке показан список всех внесенных текстов; данный инструмент также позволяет добавлять соответствующие переводы на нужных языках.

IdName	Max	English	Italian	French	Spanish	German	Russian	Portugur	Other1	Other2	Other3
Text116	9	Unit OFF	Unità OFF								
Keyboard	21	by Key	da Tasto								
OFF_D2	21	by Digital Input	da Ingresso Digitale								
OFF_S1	21	by Supervisor	da Supervisore								
OFF_Sc	21	by Scheduler	da Scheduler								
OFF_A1	21	by Alarm	da Allarme								
ONB6	21										
Text160	8	Environment	Ambiente								
*C35	2	°C	°C								
*F35	2	°F	°F								
Text152	8	Return	Ripresa								
Text171	1	%	%								
Text156	8	Supply	Mandata								
Text173	1	%	%								

Первый столбец указывает имя EIML-элементов, второй столбец показывает максимальный размер отображения элемента на странице, далее же следуют переводы на языках, поддерживаемых в EIML.

Если элементу присвоен пустой текст, он будет выделен зеленым цветом.

Если текст длиннее, чем максимальное число символов, доступное для отображения элемента на EIML-странице, то он выделен красным цветом.

<b>Open File</b>	Открывает многоязычный файл CSV, который до этого был сохранен при экспорте языков проекта (также возможно открытие файлов других проектов).
<b>Save File</b>	Сохраняет многоязычный файл CSV со всеми элементами в многоязычных таблицах.
<b>Import in Project</b>	Пробует импортировать элементы в текущий проект; в случае возникновения ошибки во время импорта показывается предупреждающее сообщение, которое затем сохраняется в файле журнала (в директории проекта).
<b>Language</b>	При импорте строк в проект импортируются только элементы выбранного языка (или ВСЕ).
<b>Skip Blank Cell</b>	Если этот флажок установлен, при импорте в проект пустые ячейки пропускаются, сохраняется предыдущая строка.

## 4.29 Редактор сущностей

Из меню *Инструменты* → *Редактор сущностей* вы можете получить доступ к инструменту для просмотра и изменения некоторых свойств объектов проекта.

The screenshot shows the 'Entities Editor' window with a search bar and a table of object properties. The table has columns for CAT, Data Type, Name, Value, Min, Max, Par Min, Par Max, Precision, Description, VTP, Web Server, and Visibility. The 'EV23\_Frequency' row is highlighted in blue.

CAT	Data Type	Name	Value	Min	Max	Par Min	Par Max	Precision	Description	VTP	Web Server	Visibility
◆	CJ_WORD	EV16_StartUpDelay_EV1	30	0	255	[NONE]	[NONE]	0	sec			<input type="checkbox"/>
◆	CJ_WORD	EV17_StartUpPosition_EV	50.00	0.00	100.00	[NONE]	[NONE]	2	%			<input type="checkbox"/>
◆	CJ_BIT	EV18_AI14probeType_EV	1	0	1	[NONE]	[NONE]	0	0=4...20mA, 1=0-5V			<input type="checkbox"/>
◆	CJ_SHORT	EV19_AI14_ymin_EV1	0.00	-1.00	300.00	[NONE]	[NONE]	2				<input type="checkbox"/>
◆	CJ_BIT	EV20_DI1	0	0	1	-	-	0				<input type="checkbox"/>
◆	CJ_BIT	EV20_DI2	0	0	1	-	-	0				<input type="checkbox"/>
◆	CJ_BIT	EV20_DI3	0	0	1	-	-	0				<input type="checkbox"/>
◆	CJ_BIT	EV23_Frequency	0	0	1	[NONE]	[NONE]	0				<input type="checkbox"/>
◆	CJ_SHORT	EV20_AI14_ymax_EV1	17.30	-1.00	300.00	[NONE]	[NONE]	2				<input type="checkbox"/>
◆	CJ_SHORT	EV21_TemperatureOffs	0.0	-72.5	72.5	Lim_memoSbar	Lim_Sbar	1	°K			<input type="checkbox"/>
◆	CJ_BIT	EV22_AI12probeType_EV	1	0	1	[NONE]	[NONE]	0	0=4...20mA, 1=0-5V			<input type="checkbox"/>
◆	CJ_SHORT	EV23_AI12_ymin_EV1	0.00	-1.00	300.00	[NONE]	[NONE]	2				<input type="checkbox"/>
◆	CJ_SHORT	EV24_AI12_ymax_EV1	45.00	-1.00	300.00	[NONE]	[NONE]	2				<input type="checkbox"/>
◆	CJ_SHORT	EV25_OffsetHP_EV1	0.0	-72.5	72.5	Lim_memoSbar	Lim_Sbar	1	Bar/psi			<input type="checkbox"/>
◆	CJ_SHORT	EV26_TsTemperatureOffs	0.0	-18.0	18.0	Delta_Gr_Memo10	Delta_Gr_10	1	°K			<input type="checkbox"/>
◆	CJ_SHORT	EV27_TaraturaTsCarico_E	0.0	-18.0	18.0	Delta_Gr_Memo10	Delta_Gr_10	1	°C			<input type="checkbox"/>
◆	CJ_BYTE	EV29_ValveType_EV2	4	1	28	[NONE]	[NONE]	0				<input type="checkbox"/>
◆	CJ_BIT	EV3_FunctioningMode_EV	0	0	1	[NONE]	[NONE]	0	0=algo, 1=manual			<input type="checkbox"/>
◆	CJ_BIT	EV30_Frequency_EV2	0	0	1	[NONE]	[NONE]	0				<input type="checkbox"/>
◆	CJ_BIT	EV31_FunctioningMode_E	0	0	1	[NONE]	[NONE]	0	%			<input type="checkbox"/>
◆	CJ_WORD	EV32_ManualValvePositor	0.00	0.00	100.00	[NONE]	[NONE]	2	%			<input type="checkbox"/>
◆	CJ_BIT	EV33_resyncro_EV2	0	0	1	-	-	0				<input type="checkbox"/>
◆	CJ_SHORT	EV34_Shsetpoint_EV2	6.0	3.0	25.0	[NONE]	[NONE]	1	°K			<input type="checkbox"/>
◆	CJ_SHORT	EV35_LoShsetpoint_EV2	3.0	1.0	3.0	[NONE]	[NONE]	1	°K			<input type="checkbox"/>

Это список в формате таблицы всех объектов, с помощью которой можно просматривать и изменять некоторые свойства без необходимости поиска объекта в проекте.

Это полезно, прежде всего, на этапе проверки при проверке правильности значений, присвоенных во время разработки, и/или для изменения конфигураций приложения по умолчанию.

Свойства можно редактировать индивидуально, изменяя ячейку, некоторые необходимо модифицировать с помощью специальных окон, нажав кнопку «...» Кнопка, которая появляется при редактировании ячейки.

Серые ячейки указывают на то, что свойство не может быть активировано для конкретного объекта.

Применение изменений к проекту происходит только после подтверждения закрытия инструмента кнопкой **ОК**. Нажатие кнопки **Отмена** отменит все внесенные изменения.

Управляемые свойства следующие:

- **Имя** указывает имя сущности. Любое изменение этого свойства приводит к обновлению таблицы, поскольку она должна быть уникальна.
- **Значение, Min, Max** соответственно представляют собой значение по умолчанию, минимальный предел и максимальный предел сущности.
- **Пар Min** и **Пар Max** указывают, ограничен ли объект (только Par) максимумом и минимумом другими объектами; NONE означает отсутствие ограничений.
- **VTP** содержит строку для создания дополнительных свойств в файле драйвера VTP.
- **Web-сервер** содержит строку для корректной генерации дополнительной информации для отображения объекта на Web-сервере контроллера.
- **Видимость** указывает, подлежит ли объект условной видимости на страницах EIML.

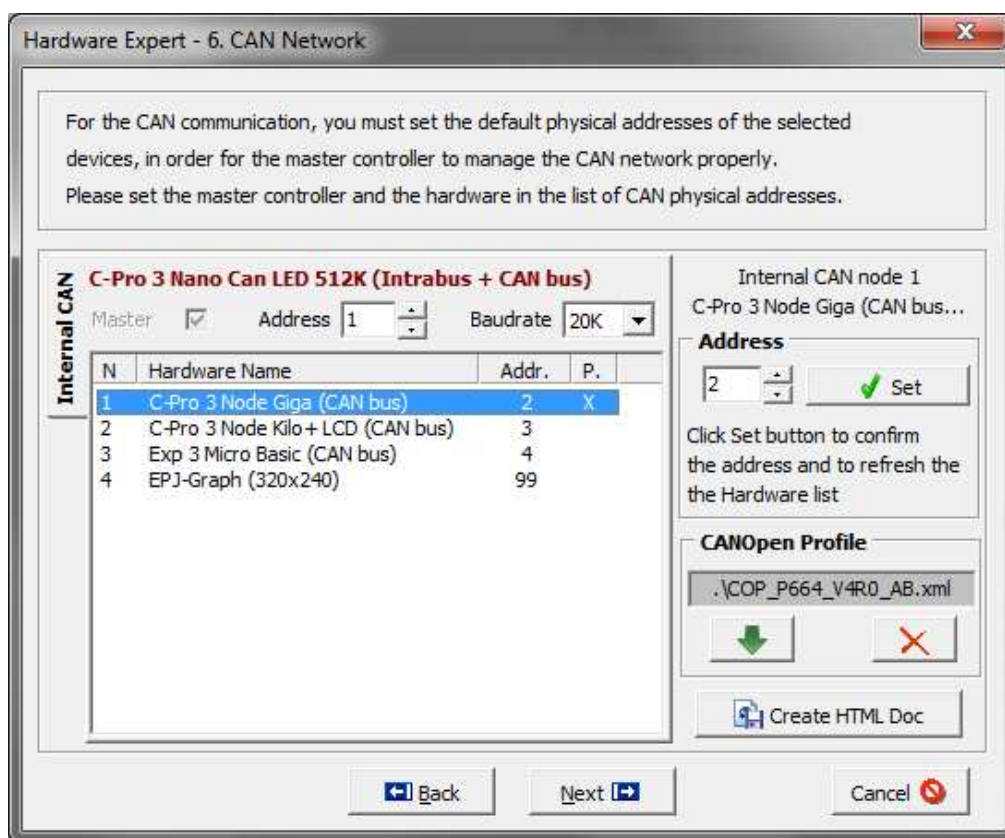
### 4.30 Профиль CANOpen для EIML

Через профили CANOpen можно использовать на страницах EIML объекты, присутствующие в других проектах UNI-PRO, вставленных в ту же сеть CAN разрабатываемого проекта.


Файлы профилей представляют собой файлы драйверов «.xml», генерация профилей CANOpen отдельных проектов происходит путем выбора пункта в меню *Проект/Создать профиль CANOpen*. Профиль содержит все объекты, которые ранее были экспортированы в регистры Modbus с помощью инструмента *Экспорт объектов*.


Чтобы получить сводную документацию по сгенерированному профилю, можно использовать документацию CANOpen, сгенерированную инструментом *Экспорт объектов*.

Профили CANOpen должны быть загружены с помощью HwExpert. В окне сети CAN вы можете загрузить профили для тех узлов, которыми вы хотите управлять. Профили могут быть связаны только с узлами - контроллерами, то есть только с узлами, которые могут содержать проект. Расширения ввода-вывода и средства просмотра не могут быть связаны с профилем.



Имя связанного файла профиля отображается в разделе *Профиль CANOpen*, в противном случае отображается «Нет профиля». Раздел отключен, если выбран узел в сети, который не может загрузить профиль.

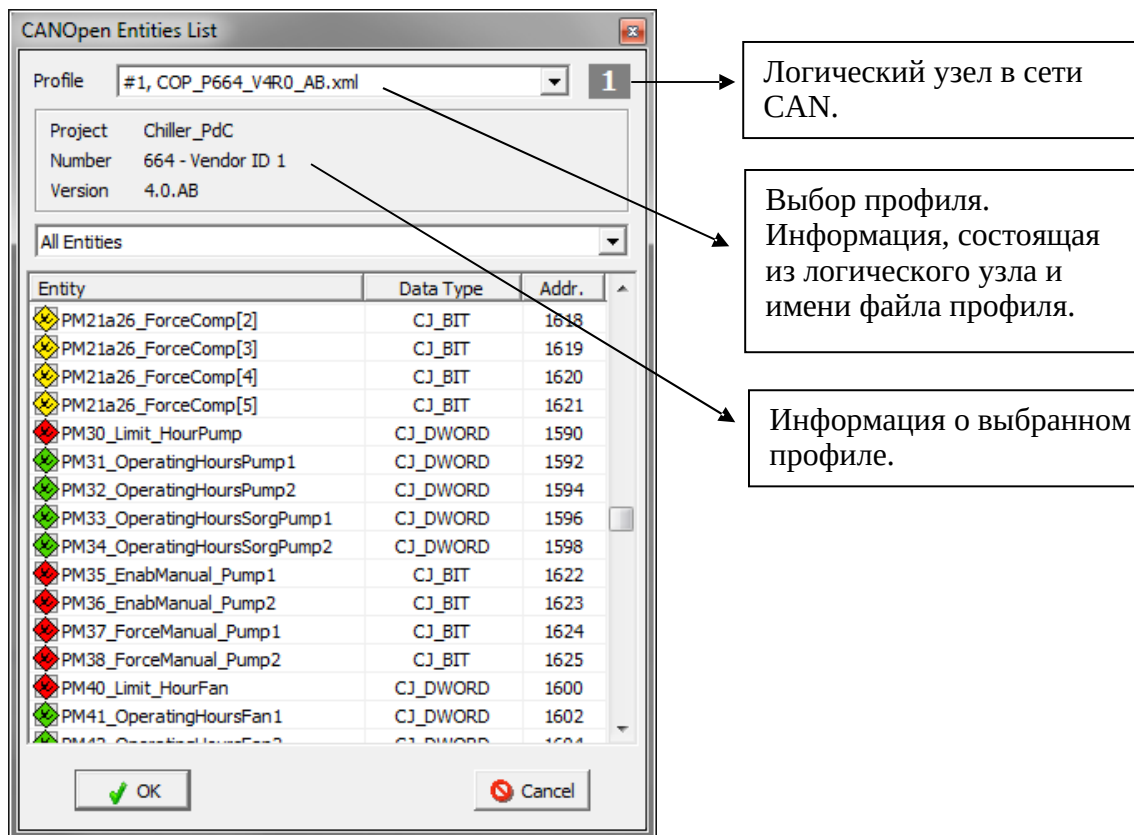
Чтобы связать профиль с выбранным оборудованием, нажмите кнопку 

Чтобы удалить профиль, нажмите кнопку 

### 4.30.1 Список объектов CANOpen

Через свойство *CANOpen Var* элементов EIML можно подключить внешний объект, выбрав его из присутствующих в профилях CANOpen, загруженных в проект.

Откроется инструмент списка объектов CANOpen.



Чтобы выбрать объект, выберите его из списка и нажмите кнопку ОК или дважды щелкните выбранный объект.

### 4.30.2 Сохранение профиля CANOpen

При сохранении проекта все профили, загруженные в каталог того же уровня, что и файл проекта *ucjr*, также сохраняются на диск (пример «ProjectName\_X.ucjr», папка сохранения «\COPProfile\_ProjectName\_X»). Имя папки, содержащей профили, и имя профилей менять нельзя. Если есть необходимость переименовать файл проекта *ucjr*, папка также должна быть переименована таким же образом.

При открытии проекта UNI-PRO загрузит файлы профиля, если один из файлов отсутствует, будет сообщено об ошибке. Во избежание нежелательной потери данных, в случае ошибки загрузки профилей закройте проект без сохранения и восстановления исходной ситуации.

### 4.31 Контроль совместимости для дисплея EPJ-Color

При наличии в сети CAN дисплея EPJ-Color необходимо скомпилировать и загрузить объектный файл (prjName\_Browser.ucjb и prjName\_Browser.ucje) также для интерфейса.

Цветные страницы EIML, фоны и некоторая информация о проекте останутся в EPJ-Color, поэтому необходимо также обновить это оборудование.

Когда запускается связь CAN между контроллером и дисплеем, выполняется контроль совместимости. Элемент управления необходим, чтобы гарантировать обновление данных в контроллере и в дисплее. Результат связывания можно проверить через страницу «Проверка» дисплея в начале соединения. Тест на совместимость может дать три результата:

- **No Compatibility:** Совместимости нет, количество страниц EIML или количество загруженного проекта и в браузере разное. В этом состоянии браузер не может связаться с контроллером.
- **Partial Compatibility:** Существует частичная совместимость, страницы EIML могут быть другими или некоторые объекты могут быть изменены; причиной является модификация данных о версии/ревизии проекта или перекомпиляция и обновление одной из частей проекта. Дисплей может связываться с контроллером, пользователь знает, что связь все еще действительна.
- **Compatibility OK:** Полная совместимость.

По окончании соединения интерфейс начнет связь или останется в состоянии «Проверка» в зависимости от результата проверки.

### 4.32 Меры предосторожности при разработке проекта

#### 1) Ресурсы могут быть ограничены

Контроллеры имеют ограниченные ресурсы оперативной памяти и памяти STACK. Они различаются по типу управления. Использование массива потребляет эти ресурсы: например, алгоритм с двумя входами массива из 100 CJ\_BYTE и выходом из 100 CJ\_BYTE использует 300 байт в стеке, в некоторых случаях может быть достаточно очистить стек с последующим сбросом контроллера! То же самое верно, если эти массивы объявлены локально внутри алгоритма.

Пример:

Порядок размера стека для используемых контроллеров Nano, Micro и Kilo - 300 байт.

#### 2) Использование математики с фиксированной и плавающей запятой

Использование арифметики (сложение, умножение, деление...) внутри алгоритмов на языке Си является обязанностью разработчика. Поэтому будьте внимательны при вводе кодов алгоритмов и заранее проверяйте результат операций на диапазон допустимых значений. Возможности автоматически контролировать, чтобы результат любой операции не выходил за допустимые пределы (переполнение) или чтобы делитель не принимал нулевое значение (деление на ноль), не существует.

Кроме того, 16-битная архитектура контроллеров обеспечивает, если не указано иное в операциях «приведения», что временный результат промежуточной операции сохраняется в 16-битном регистре с возможным усечением значения.

### 4.33 Переменные “Времени выполнения”

Во время или после разработки приложения полезно проверить системные переменные контроллера, которые позволяют оценить производительность и критические точки программы во время ее работы.

Эти системные переменные позволяют измерить время цикла основной части программы, части, находящейся в режиме прерывания, и узнать, сколько байтов в стеке еще свободно. Возможен доступ к этой информации из меню конфигурации каждого контроллера: из локального пользовательского интерфейса (встроенного) или из удаленного (дисплея), необходимо нажать UP+DOWN в течение 3 секунд, чтобы получить доступ к этому меню, затем выберите пункт «отладка». Можно использовать функции внутри алгоритмов для чтения этих переменных, чтобы обеспечить пользовательскую диагностику. Ниже подробно описаны эти переменные.

### **Время основного цикла**

Представляет время в миллисекундах, которое требуется контроллеру для выполнения основного цикла, с момента получения и блокировки всех входов до момента разблокировки и обновления выходов. Представляет время, необходимое для выполнения всех вызовов из списка вызовов основной части. Переменную можно прочитать в меню «отладка» на первых страницах конфигурации, представляет собой максимальное значение, рассчитанное при последнем сбросе контроллера. Можно изменить это время на более низкие значения, чтобы определить среднее время. В алгоритмах можно использовать следующие функции для расчета максимального значения, минимума и мгновенного значения основного времени:

```
CJ_WORD CJ_MaxMainTime(void)
CJ_WORD CJ_MinMainTime(void)
CJ_WORD CJ_RunMainTime(void)
```

Время основного цикла зависит от многих факторов, таких как используемый процессор контроллера, размер программы, использование протоколов последовательной связи, наличие расширений и удаленных пользовательских интерфейсов, расчеты с фиксированной и плавающей запятой, ... измерение этого времени может дать представление о производительности программы относительно величин контроллера. В любом случае время, превышающее 1000 мс (одну секунду), может быть избыточным для выполнения некоторых частей программы. Рекомендуется поддерживать среднее значение ниже 500 мс.

### **Таймер (или прерывание) Времени цикла**

Представляет время в миллисекундах, которое требуется контроллеру для запуска синхронизированного цикла (при прерывании 100 мс), от получения и блокировки их синхронизированных входов до момента, когда он разблокирует и обновит их синхронизированные выходы. Он представляет собой время, необходимое для выполнения всех вызовов списка относительно части 5 мс или одной двадцатой части 100 мс.

Переменная, которую можно просмотреть в меню «отладка» на первых страницах конфигурации, представляет собой максимальное значение, рассчитанное с момента последнего сброса контроллера. Можно изменить это время на более низкие значения, чтобы назначить среднее время. В алгоритмах могут быть использованы следующие функции для вычисления максимума, минимальное и мгновенное значение отсчета времени:

```
CJ_BYTE CJ_MaxInterruptTime(void)
CJ_BYTE CJ_MinInterruptTime(void)
CJ_BYTE CJ_RunInterruptTime(void)
```

Время цикла прерывания зависит от многих факторов, таких как используемый процессор контроллера, размер программы, использование протоколов последовательной связи, наличие расширений и удаленных пользовательских интерфейсов, вычисления с фиксированной и плавающей запятой, ... измерение этого времени может дать представление о производительности программы относительно величин контроллера. В любом случае время, превышающее 5 мс, может оказаться избыточным для выполнения некоторых частей программы. Рекомендуется поддерживать среднее значение менее 3 мс.

### **Размер свободного стека**

Стек — это память, которую контроллер использует динамически для размещения переменных внутри алгоритмов и вызовов функций.



Каждый контроллер имеет размера стека пропорционально общему объему ОЗУ, поэтому самые мощные контроллеры могут изначально использовать 2000 байт, а младшие — только 300 байт. Во время функционирования контроллера эта область будет использоваться для выполнения драйверов и алгоритмов и будет постоянно храниться максимальное значение, достигнутое этим занятием.

Переменная, которую вы можете прочесть в меню «отладка» на страницах конфигурации, представляет собой минимальное значение свободного стека, рассчитанное с момента последнего сброса контроллера. В алгоритмах можно использовать следующие функции для чтения минимального значения свободного стека:

```
CJ_WORD CJ_GetStackFree(void)
```

Например, если это значение равно 0, это означает, что произошло переполнение памяти с катастрофическими последствиями для стабильности программы. Рекомендуется при разработке проверять это значение и оставлять свободную зону безопасности (20-30 байт), чтобы избежать возможных неудобств.

#### 4.34 Регистрация лицензии

В меню *Помощь* выберите действие *Регистрация...*, и появится окно (из которого вы получите данные для регистрации лицензии):

Сначала вам нужно сгенерировать код активации (кнопка *Get*), затем вы должны предоставить его EVCO (через веб-сайт или по электронной почте). EVCO вернет вам соответствующий серийный код.

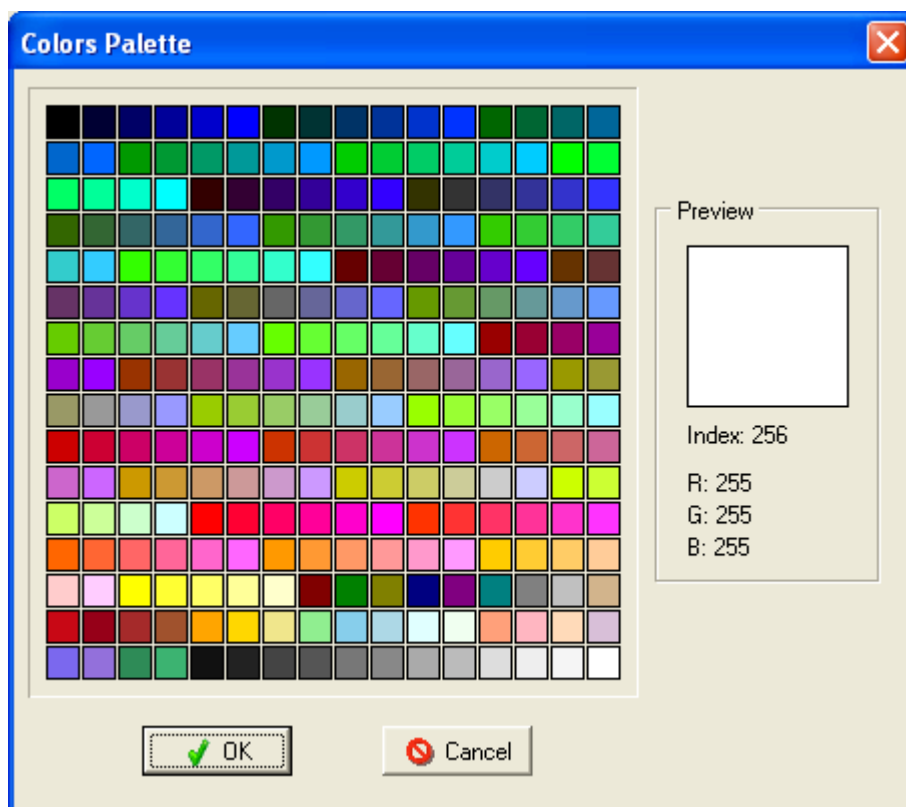
Для активации лицензии введите сгенерированный код активации, полученный серийный код и нажмите *Регистрация*. В зависимости от достоверности данных будет отображен результат операции.

Если все верно, лицензия будет немедленно активирована, иначе вы получите найденную ошибку.

Чтобы проверить статус лицензии, в меню *Помощь* выберите «О лицензии».

### 4.35 Палитра из 256 цветов UNI-PRO

Здесь и далее полный список 256 цветов, используемых UNI-PRO.



Color 1: RGB ( 0, 0, 0)	Color 31: RGB ( 0, 255, 0)	Color 61: RGB ( 51, 204, 0)
Color 2: RGB ( 0, 0, 51)	Color 32: RGB ( 0, 255, 51)	Color 62: RGB ( 51, 204, 51)
Color 3: RGB ( 0, 0, 102)	Color 33: RGB ( 0, 255, 102)	Color 63: RGB ( 51, 204, 102)
Color 4: RGB ( 0, 0, 153)	Color 34: RGB ( 0, 255, 153)	Color 64: RGB ( 51, 204, 153)
Color 5: RGB ( 0, 0, 204)	Color 35: RGB ( 0, 255, 204)	Color 65: RGB ( 51, 204, 204)
Color 6: RGB ( 0, 0, 255)	Color 36: RGB ( 0, 255, 255)	Color 66: RGB ( 51, 204, 255)
Color 7: RGB ( 0, 51, 0)	Color 37: RGB ( 51, 0, 0)	Color 67: RGB ( 51, 255, 0)
Color 8: RGB ( 0, 51, 51)	Color 38: RGB ( 51, 0, 51)	Color 68: RGB ( 51, 255, 51)
Color 9: RGB ( 0, 51, 102)	Color 39: RGB ( 51, 0, 102)	Color 69: RGB ( 51, 255, 102)
Color 10: RGB ( 0, 51, 153)	Color 40: RGB ( 51, 0, 153)	Color 70: RGB ( 51, 255, 153)
Color 11: RGB ( 0, 51, 204)	Color 41: RGB ( 51, 0, 204)	Color 71: RGB ( 51, 255, 204)
Color 12: RGB ( 0, 51, 255)	Color 42: RGB ( 51, 0, 255)	Color 72: RGB ( 51, 255, 255)
Color 13: RGB ( 0, 102, 0)	Color 43: RGB ( 51, 51, 0)	Color 73: RGB ( 102, 0, 0)
Color 14: RGB ( 0, 102, 51)	Color 44: RGB ( 51, 51, 51)	Color 74: RGB ( 102, 0, 51)
Color 15: RGB ( 0, 102, 102)	Color 45: RGB ( 51, 51, 102)	Color 75: RGB ( 102, 0, 102)
Color 16: RGB ( 0, 102, 153)	Color 46: RGB ( 51, 51, 153)	Color 76: RGB ( 102, 0, 153)
Color 17: RGB ( 0, 102, 204)	Color 47: RGB ( 51, 51, 204)	Color 77: RGB ( 102, 0, 204)
Color 18: RGB ( 0, 102, 255)	Color 48: RGB ( 51, 51, 255)	Color 78: RGB ( 102, 0, 255)
Color 19: RGB ( 0, 153, 0)	Color 49: RGB ( 51, 102, 0)	Color 79: RGB ( 102, 51, 0)
Color 20: RGB ( 0, 153, 51)	Color 50: RGB ( 51, 102, 51)	Color 80: RGB ( 102, 51, 51)
Color 21: RGB ( 0, 153, 102)	Color 51: RGB ( 51, 102, 102)	Color 81: RGB ( 102, 51, 102)
Color 22: RGB ( 0, 153, 153)	Color 52: RGB ( 51, 102, 153)	Color 82: RGB ( 102, 51, 153)
Color 23: RGB ( 0, 153, 204)	Color 53: RGB ( 51, 102, 204)	Color 83: RGB ( 102, 51, 204)
Color 24: RGB ( 0, 153, 255)	Color 54: RGB ( 51, 102, 255)	Color 84: RGB ( 102, 51, 255)
Color 25: RGB ( 0, 204, 0)	Color 55: RGB ( 51, 153, 0)	Color 85: RGB ( 102, 102, 0)
Color 26: RGB ( 0, 204, 51)	Color 56: RGB ( 51, 153, 51)	Color 86: RGB ( 102, 102, 51)
Color 27: RGB ( 0, 204, 102)	Color 57: RGB ( 51, 153, 102)	Color 87: RGB ( 102, 102, 102)
Color 28: RGB ( 0, 204, 153)	Color 58: RGB ( 51, 153, 153)	Color 88: RGB ( 102, 102, 153)
Color 29: RGB ( 0, 204, 204)	Color 59: RGB ( 51, 153, 204)	Color 89: RGB ( 102, 102, 204)
Color 30: RGB ( 0, 204, 255)	Color 60: RGB ( 51, 153, 255)	Color 90: RGB ( 102, 102, 255)

## UNI-PRO 3 РУКОВОДСТВО ПО ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Color 91: RGB (102, 153, 0)	Color152: RGB (204, 51, 51)	Color213: RGB (255, 255, 102)
Color 92: RGB (102, 153, 51)	Color153: RGB (204, 51, 102)	Color214: RGB (255, 255, 153)
Color 93: RGB (102, 153, 102)	Color154: RGB (204, 51, 153)	Color215: RGB (255, 255, 204)
Color 94: RGB (102, 153, 153)	Color155: RGB (204, 51, 204)	Color216: RGB (128, 0, 0)
Color 95: RGB (102, 153, 204)	Color156: RGB (204, 51, 255)	Color217: RGB ( 0, 128, 0)
Color 96: RGB (102, 153, 255)	Color157: RGB (204, 102, 0)	Color218: RGB (128, 128, 0)
Color 97: RGB (102, 204, 0)	Color158: RGB (204, 102, 51)	Color219: RGB (0, 0, 128)
Color 98: RGB (102, 204, 51)	Color159: RGB (204, 102, 102)	Color220: RGB (128, 0, 128)
Color 99: RGB (102, 204, 102)	Color160: RGB (204, 102, 153)	Color221: RGB ( 0, 128, 128)
Color100: RGB (102, 204, 153)	Color161: RGB (204, 102, 204)	Color222: RGB (128, 128, 128)
Color101: RGB (102, 204, 204)	Color162: RGB (204, 102, 255)	Color223: RGB (192, 192, 192)
Color102: RGB (102, 204, 255)	Color163: RGB (204, 153, 0)	Color224: RGB (210, 180, 140)
Color103: RGB (102, 255, 0)	Color164: RGB (204, 153, 51)	Color225: RGB (200, 8, 21)
Color104: RGB (102, 255, 51)	Color165: RGB (204, 153, 102)	Color226: RGB (150, 0, 24)
Color104: RGB (102, 255, 102)	Color166: RGB (204, 153, 153)	Color227: RGB (165, 42, 42)
Color105: RGB (102, 255, 153)	Color167: RGB (204, 153, 204)	Color228: RGB (160, 82, 45)
Color106: RGB (102, 255, 204)	Color168: RGB (204, 153, 255)	Color229: RGB (255, 165, 0)
Color107: RGB (102, 255, 255)	Color169: RGB (204, 204, 0)	Color230: RGB (255, 215, 0)
Color108: RGB (153, 0, 0)	Color170: RGB (204, 204, 51)	Color231: RGB (240, 230, 140)
Color110: RGB (153, 0, 51)	Color171: RGB (204, 204, 102)	Color232: RGB (144, 238, 144)
Color111: RGB (153, 0, 102)	Color172: RGB (204, 204, 153)	Color233: RGB (135, 206, 235)
Color112: RGB (153, 0, 153)	Color173: RGB (204, 204, 204)	Color234: RGB (173, 216, 230)
Color113: RGB (153, 0, 204)	Color174: RGB (204, 204, 255)	Color235: RGB (224, 255, 255)
Color114: RGB (153, 0, 255)	Color175: RGB (204, 255, 0)	Color236: RGB (240, 255, 240)
Color115: RGB (153, 51, 0)	Color176: RGB (204, 255, 51)	Color237: RGB (255, 160, 122)
Color116: RGB (153, 51, 51)	Color177: RGB (204, 255, 102)	Color238: RGB (255, 182, 193)
Color117: RGB (153, 51, 102)	Color178: RGB (204, 255, 153)	Color239: RGB (255, 218, 185)
Color118: RGB (153, 51, 153)	Color179: RGB (204, 255, 204)	Color240: RGB (216, 191, 216)
Color119: RGB (153, 51, 204)	Color180: RGB (204, 255, 255)	Color241: RGB (123, 104, 238)
Color120: RGB (153, 51, 255)	Color181: RGB (255, 0, 0)	Color242: RGB (147, 112, 219)
Color121: RGB (153, 102, 0)	Color182: RGB (255, 0, 51)	Color243: RGB ( 46, 139, 87)
Color122: RGB (153, 102, 51)	Color183: RGB (255, 0, 102)	Color244: RGB ( 60, 179, 113)
Color123: RGB (153, 102, 102)	Color184: RGB (255, 0, 153)	Color245: RGB ( 17, 17, 17)
Color124: RGB (153, 102, 153)	Color185: RGB (255, 0, 204)	Color246: RGB ( 34, 34, 34)
Color125: RGB (153, 102, 204)	Color186: RGB (255, 0, 255)	Color247: RGB ( 68, 68, 68)
Color126: RGB (153, 102, 255)	Color187: RGB (255, 51, 0)	Color248: RGB ( 85, 85, 85)
Color127: RGB (153, 153, 0)	Color188: RGB (255, 51, 51)	Color249: RGB (119, 119, 119)
Color128: RGB (153, 153, 51)	Color189: RGB (255, 51, 102)	Color250: RGB (136, 136, 136)
Color129: RGB (153, 153, 102)	Color190: RGB (255, 51, 153)	Color251: RGB (170, 170, 170)
Color130: RGB (153, 153, 153)	Color191: RGB (255, 51, 204)	Color252: RGB (187, 187, 187)
Color131: RGB (153, 153, 204)	Color192: RGB (255, 51, 255)	Color253: RGB (224, 224, 224)
Color132: RGB (153, 153, 255)	Color193: RGB (255, 102, 0)	Color254: RGB (238, 238, 238)
Color133: RGB (153, 204, 0)	Color194: RGB (255, 102, 51)	Color255: RGB (245, 245, 245)
Color134: RGB (153, 204, 51)	Color195: RGB (255, 102, 102)	Color256: RGB (255, 255, 255)
Color135: RGB (153, 204, 102)	Color196: RGB (255, 102, 153)	
Color136: RGB (153, 204, 153)	Color197: RGB (255, 102, 204)	
Color137: RGB (153, 204, 204)	Color198: RGB (255, 102, 255)	
Color138: RGB (153, 204, 255)	Color199: RGB (255, 153, 0)	
Color139: RGB (153, 255, 0)	Color200: RGB (255, 153, 51)	
Color140: RGB (153, 255, 51)	Color201: RGB (255, 153, 102)	
Color141: RGB (153, 255, 102)	Color202: RGB (255, 153, 153)	
Color142: RGB (153, 255, 153)	Color203: RGB (255, 153, 204)	
Color143: RGB (153, 255, 204)	Color204: RGB (255, 153, 255)	
Color144: RGB (153, 255, 255)	Color205: RGB (255, 204, 0)	
Color145: RGB (204, 0, 0)	Color206: RGB (255, 204, 51)	
Color146: RGB (204, 0, 51)	Color207: RGB (255, 204, 102)	
Color147: RGB (204, 0, 102)	Color208: RGB (255, 204, 153)	
Color148: RGB (204, 0, 153)	Color209: RGB (255, 204, 204)	
Color149: RGB (204, 0, 204)	Color210: RGB (255, 204, 255)	
Color150: RGB (204, 0, 255)	Color211: RGB (255, 255, 0)	
Color151: RGB (204, 51, 0)	Color212: RGB (255, 255, 51)	

## **5 Что нового в последней версии**

### **5.1 *Новые особенности***



## UNI-PRO 3 РУКОВОДСТВО ПО ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

UNI-PRO 3 - Руководство по программному обеспечению.

Версия 4.6 – Декабрь 2021.

Код 114UP3E464.

Файл 114UP3E464.pdf.

Данная публикация является исключительной собственностью Evco. Копирование и воспроизведение материалов в любой форме без предварительного разрешения Evco запрещено. Evco не несёт ответственности за характеристики, техническую информацию и другие ошибки, представленные в данной публикации, а также за последствия их использования. Evco не несёт ответственности за нарушения, вызванные несоблюдением мер предосторожности. Компания сохраняет за собой право вносить изменения в ходе технических разработок в любое время без предварительного уведомления при отсутствии значительных изменений, касающихся функционала или безопасности.



**ГЛАВНЫЙ ОФИС**

**Evco**

Via Mezzaterra 6, 32036 Sedico Belluno ITALY  
Tel. +39 0437-852468  
Fax +39 0437-83648  
info@evco.it  
www.evco.it

**ПРЕДСТАВИТЕЛЬСТВА КОМПАНИИ**

**Control France**

155 Rue Roger Salengro, 92370 Chaville Paris FRANCE  
Tel. 0033-1-41159740  
Fax 0033-1-41159739  
control.france@wanadoo.fr

**Evco Latina**

Larrea, 390 San Isidoro, 1609 Buenos Aires ARGENTINA  
Tel. 0054-11-47351031  
Fax 0054-11-47351031  
evcolatina@anykasrl.com.ar

**Evco Pacific**

59 Premier Drive Campbellfield, 3061, Victoria Melbourne, AUSTRALIA  
Tel. 0061-3-9357-0788  
Fax 0061-3-9357-7638  
everycontrol@pacific.com.au

**Evco Russia**

111141 Russia Moscow 2-oy Proezd Perova Polya 9  
Tel. 007-495-3055884  
Fax 007-495-3055884  
info@evco.ru

**Every Control do Brasil**

Rua Marino Félix 256, 02515-030 Casa Verde São Paulo SÃO PAULO BRAZIL  
Tel. 0055-11-38588732  
Fax 0055-11-39659890  
info@everycontrol.com.br

**Every Control Norden**

Cementvägen 8, 136 50 Haninge SWEDEN  
Tel. 0046-8-940470  
Fax 0046-8-6053148  
mail2@unilec.se

**Every Control Shangai**

B 302, Yin Hai Building, 250 Cao Xi Road, 200235 Shangai CHINA  
Tel. 0086-21-64824650  
Fax 0086-21-64824649  
evcosh@online.sh.cn

**Every Control United Kingdom**

Unit 19, Monument Business Park, OX44 7RW Chalgrove, Oxford, UNITED KINGDOM  
Tel. 0044-1865-400514  
Fax 0044-1865-400419  
info@everycontrol.co.uk