

Федеральное государственное бюджетное образовательное  
учреждение высшего профессионального образования  
«Чувашский государственный университет имени И.Н. Ульянова»

**ЗАПУСК НА ВЫПОЛНЕНИЕ ГОТОВЫХ ПРОЕКТОВ.**

---

Назарова Ольга Васильевна

После разработки Проекта необходимо сформировать привычный для оператора механизм запуска нового программного приложения в рамках операционной системы. Для этого используется ярлык, который посредством специальных ключей позволит запустить на выполнение МРВ, открыть файл Проекта, перевести МРВ в полноэкранный режим. Рассмотрим ключи, которые выполняют данные функции.

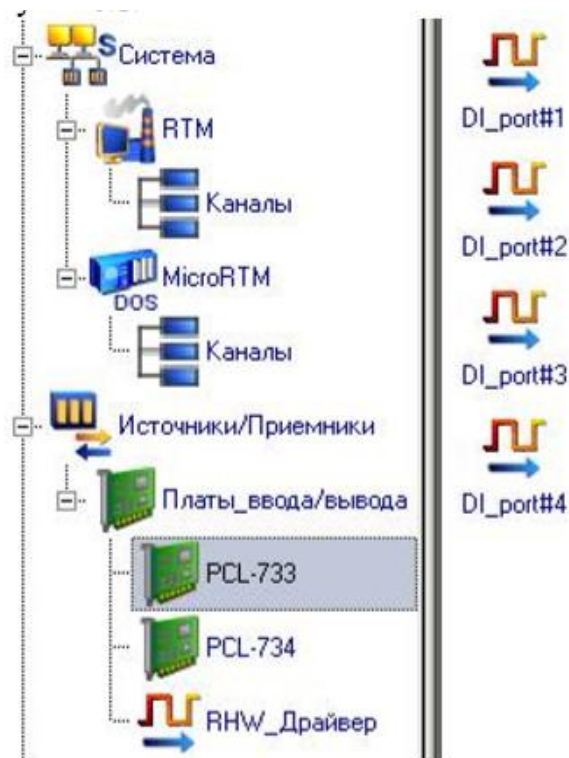
- ❑ «/run» (Ctrl + R) – ключ используется для автоматического запуска на выполнение МРВ.
- ❑ «/fullscreen» (Ctrl + F) – ключ используется для перевода МРВ в полноэкранный режим.
- ❑ «/hidemenu» (Ctrl + T) – ключ используется для того, чтобы скрыть консоль кнопок с пиктограммами.

Таким образом, полная строка запуска на выполнение МРВ будет выглядеть так:

«[Path\_RTM/rtc.exe] [Path\_name.dbb/name.dbb] /run /fullscreen /hidemenu»,  
где rtc.exe – запускаемый файл МРВ, name.dbb – основной файл Проекта.

## Принципы разработки и структура Проекта создаваемого в TRACE MODE 6.06.2 для микропроцессорного контроллера с платами УСО

Принципиальным отличием Проектов предназначенных для контроллеров с УСО от других, является наличие дополнительных компонентов настраиваемых в слое «Источники/Приёмники».



Рассмотрим структуру Проекта с использованием драйверов УСО на примере рисунков 2, 3.

Пусть имеется плата ввода дискретных сигналов в составе некоторого микропроцессорного контроллера. Плата имеет 32 входа: DI-0 ... DI-31. В составе Проекта инициализируется соответствующий типу и названию платы драйвер. Автоматически в составе слоя с названием драйвера создаются информационные порты в количестве 4-х штук. В составе каждого порта имеется восемь регистров группирующих на выходе порта последовательность из восьми бит данных. Состояние каждого регистра характеризует состояние соответствующего входа платы УСО. Для того чтобы в составе Проекта можно было использовать значения, получаемые на выходе драйвера УСО, необходимо настроить порты драйвера (данная операция будет рассмотрена далее) и выполнить их автопривязку (методом «Drag and Drop») ко входам каналов в составе слоя «Система» (для плат ввода сигналов) или входов портов драйвера к выходам каналов в слое «Система» (для плат вывода сигналов из ЭВМ).

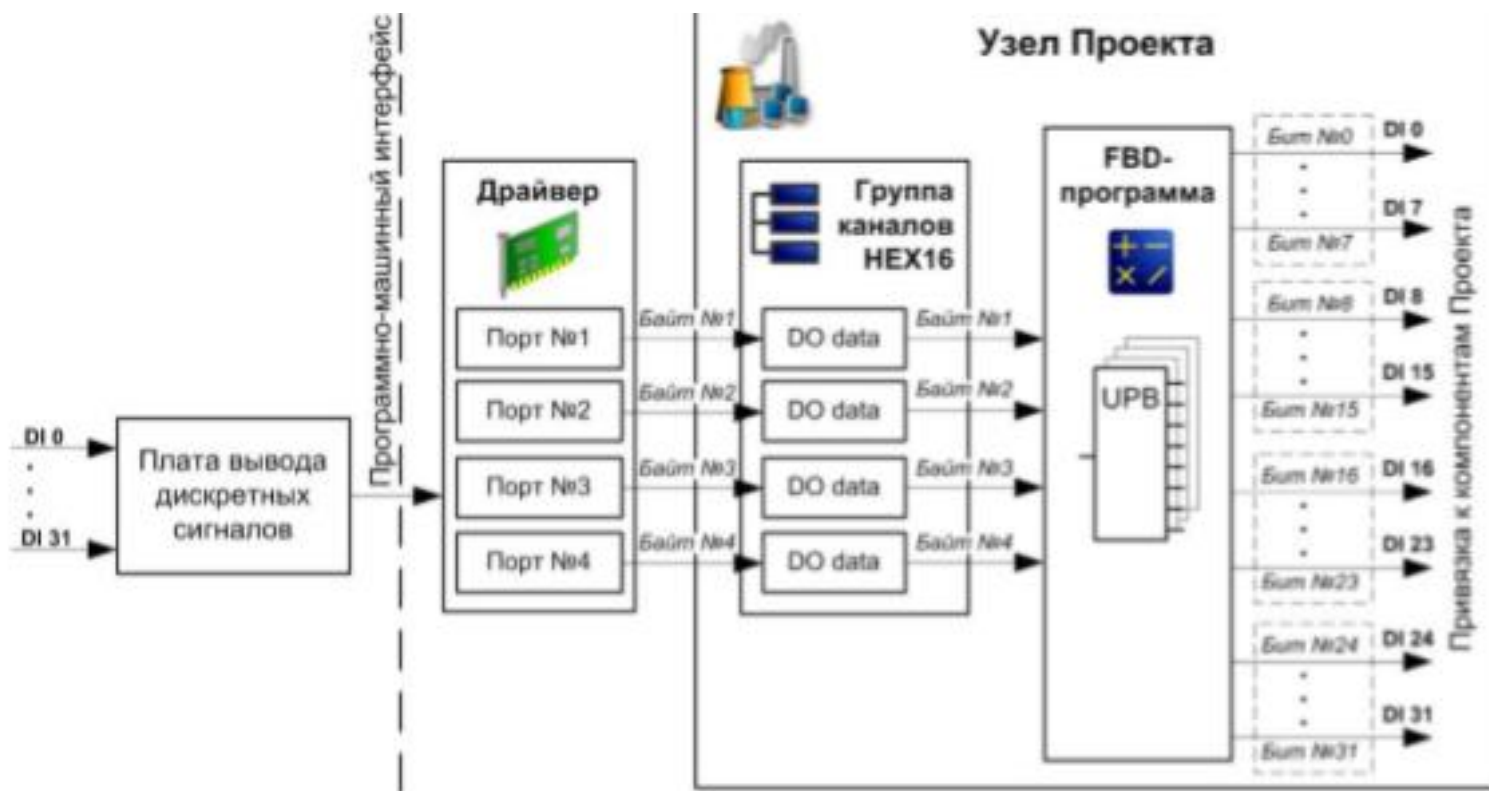


Рис. 2 Структура Проекта с использованием драйвера платы ввода дискретных сигналов

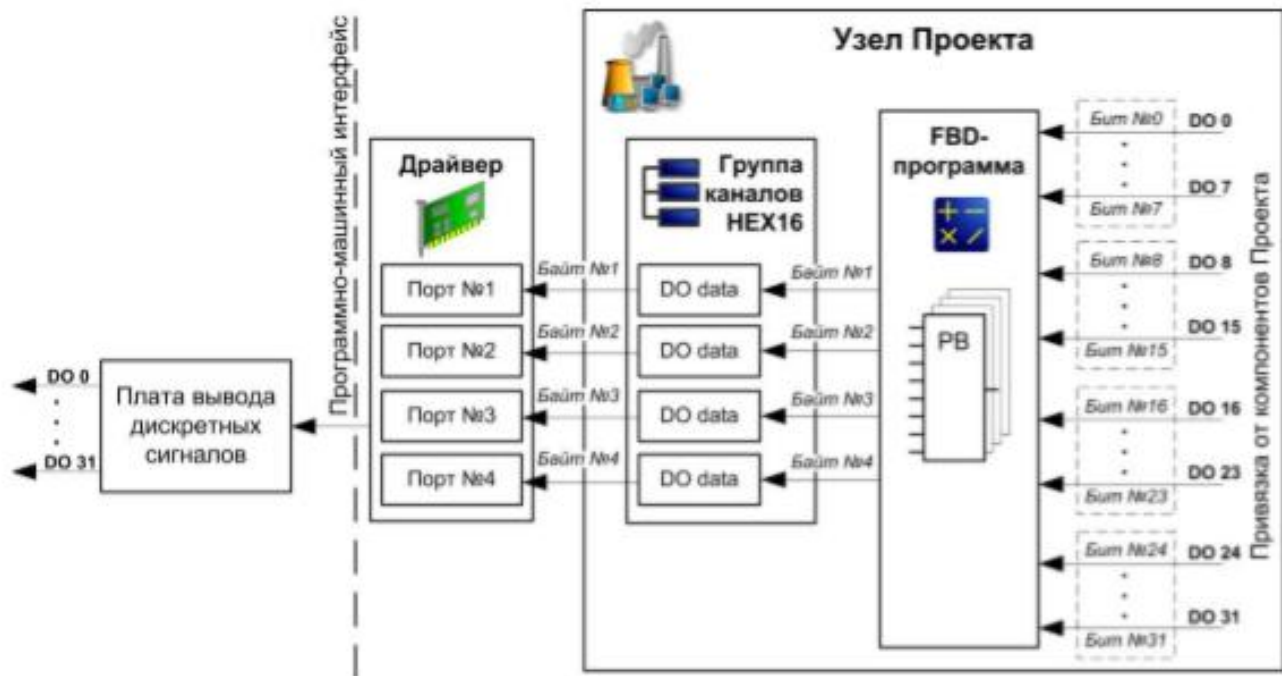


Рис. 3 Структура Проекта с использованием драйвера платы вывода дискретных сигналов

## Настройка драйвера в составе Проекта

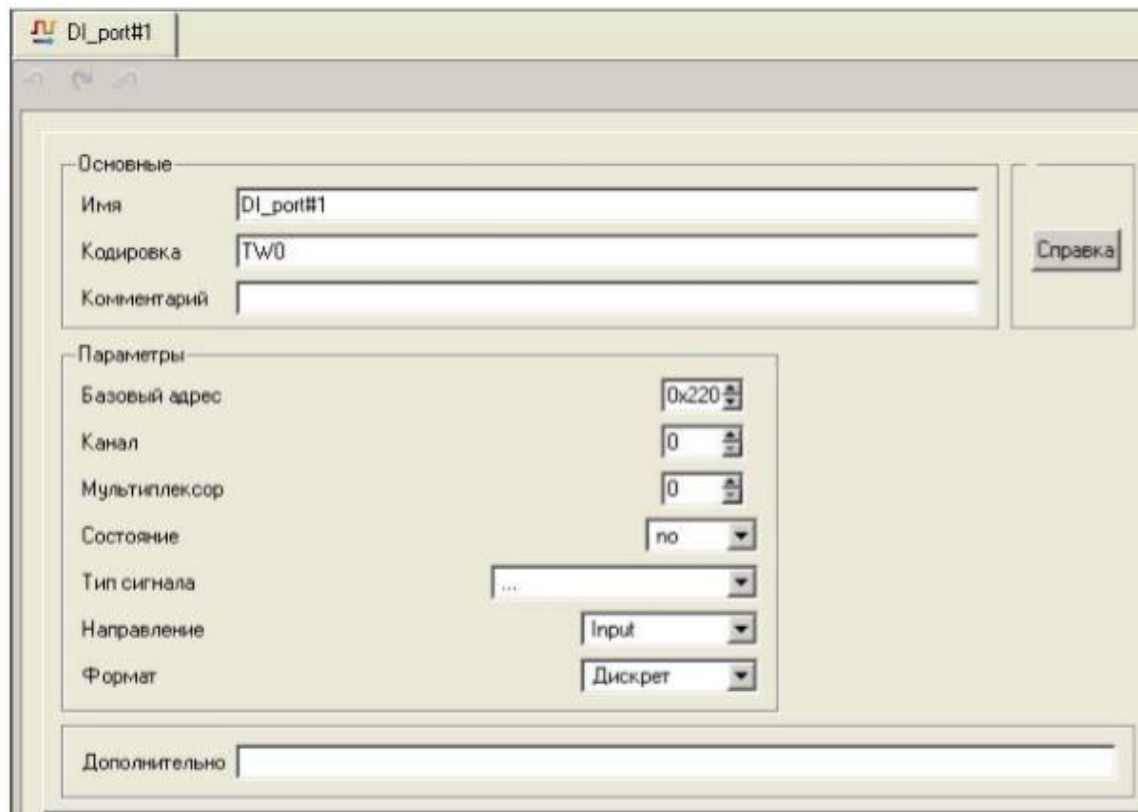


Рис. 4. Вид окна редактора параметров настройки драйвера

Состав полей редактора может быть различным в зависимости от назначения УСО. В этот состав могут входить поля настройки следующих параметров:

1. имя – название порта или канала платы УСО;
2. кодировка – уникальная кодировка компонента Проекта (задаётся автоматически);
3. комментарий – поясняющий текст, в произвольной форме, предназначенный для текущего элемента Проекта;
4. базовый адрес – адрес на локальной шине, начиная с которого УСО выделяется рабочее адресное пространство;
5. канал – смещение от базового адреса для каждого порта драйвера (настраивается автоматически при инициализации связи с драйвером);
6. мультиплексор – номер канала мультиплексора (если используется);
7. тип сигнала – диапазон изменения сигнала в физических единицах измерения (например, 4-20 мА, 1-5 В);
8. направление – направление передачи данных (в/из ЭВМ);
9. формат – формат передаваемых данных (дискретный или аналоговый).



# Драйвер RWH

При необходимости использовать УСО, драйвер для которого не существует в стандартной библиотеке TRACE MODE, можно разработать собственный драйвер на языке C++, оформить его в виде библиотеки RWH и добавить в состав Проекта. Методы использования RWHдрайвера не отличаются от методов использования встроенных драйверов. В составе слоя RWH-драйвер необходимо создать достаточное количество портов ввода или вывода сигналов, а затем настроить их в соответствии со структурой данного драйвера.

В составе среды разработки TRACE MODE 6.06.2 также поддерживаются драйверы типа t13.dll, которые использовались в более ранних версиях TRACE MODE. Однако разработчики не рекомендуют использовать данную технологию в новых проектах ПО.

# Драйвер терминала в DOS

Для проектирования средств отображения информации в DOS в составе TRACE MODE 6.06.2 используется драйвер терминала. Компонент «Терминал» в TRACE MODE инициализируется в слое «Источники/Приёмники» в подслое «Терминалы». Данный компонент должен быть вызван каналом класса CALL типа Input с типом вызова «Mneto» из Узла MicroRTM.

## Настройка компонента «Терминал»

В настройках данного компонента необходимо задать значения параметров «Параметр» и «Размер строки». «Параметр» определяет количество аргументов в канале класса CALL устанавливающих параметры отображения мнемознака (например, 4), «Размер строки» определяет максимальное количество мнемознаков в строке на экране (например, 80).

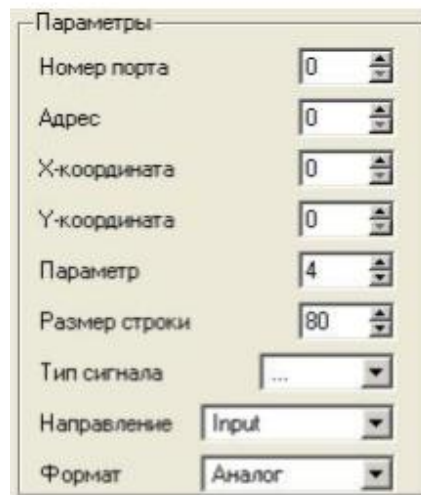


Рис. 5. Вид окна программы-настройщика параметров компонента «Терминал»

# Настройка параметров отображения данных

Структура аргументов следующая:

Arg000 – задаёт позицию начальной строки отображения данных (тип данных int);

Arg001 – задаёт количество строк на экране (тип данных int);

Arg002 – зарезервирован, не используется.

Далее аргументы формируются группами для каждого отображаемого параметра в следующем порядке:

Arg003 – номер позиции знакоместа отображаемых данных по оси X (тип данных UDINT);

Arg004 – номер позиции знакоместа отображаемых данных по оси Y (тип данных UDINT);

Arg005 – задаёт цвет отображаемого параметра в соответствии с таблицей кодов (таблица 1) (тип данных INT);

Arg006 – задаёт название отображаемого параметра как название канала, к которому данный аргумент привязан.

Внутреннее значение Канала используется как значение отображаемого параметра (тип данных REAL).

## Настройка параметров отображения данных

Далее блоки аргументов (по 4 шт. – определяется параметром «Параметр» компонента «Терминал») повторяются в зависимости от количества отображаемых параметров на экране. Значения аргументов Arg003 – Arg006 в канале класса CALL, задаются в шестнадцатеричном формате («16#001» или «16#00A»). Цвет отображаемого параметра задаётся в формате «16#ху», где х – цвет фона отображаемого параметра, у – цвет текста отображаемого параметра. Таблица кодирования цвета представлена ниже.

Таблица кодов цвета отображаемого на терминале аргумента

Число	Цвет	Число	Цвет
0	Black (черный)	8	Gray (серый)
1	Blue (синий)	9	Light Blue (светло-синий)
2	Green (зеленый)	A	Light Green (светло-зеленый)
3	Cyan (голубой)	B	Light Cyan (светло-голубой)
4	Red (красный)	C	Light Red (светло-красный)
5	Magenta (фиолетовый)	D	Light Magenta (светло-фиолетовый)
6	Yellow (желтый)	E	Light Yellow (светло-желтый)
7	White (белый)	F	Bright White (ярко-белый)

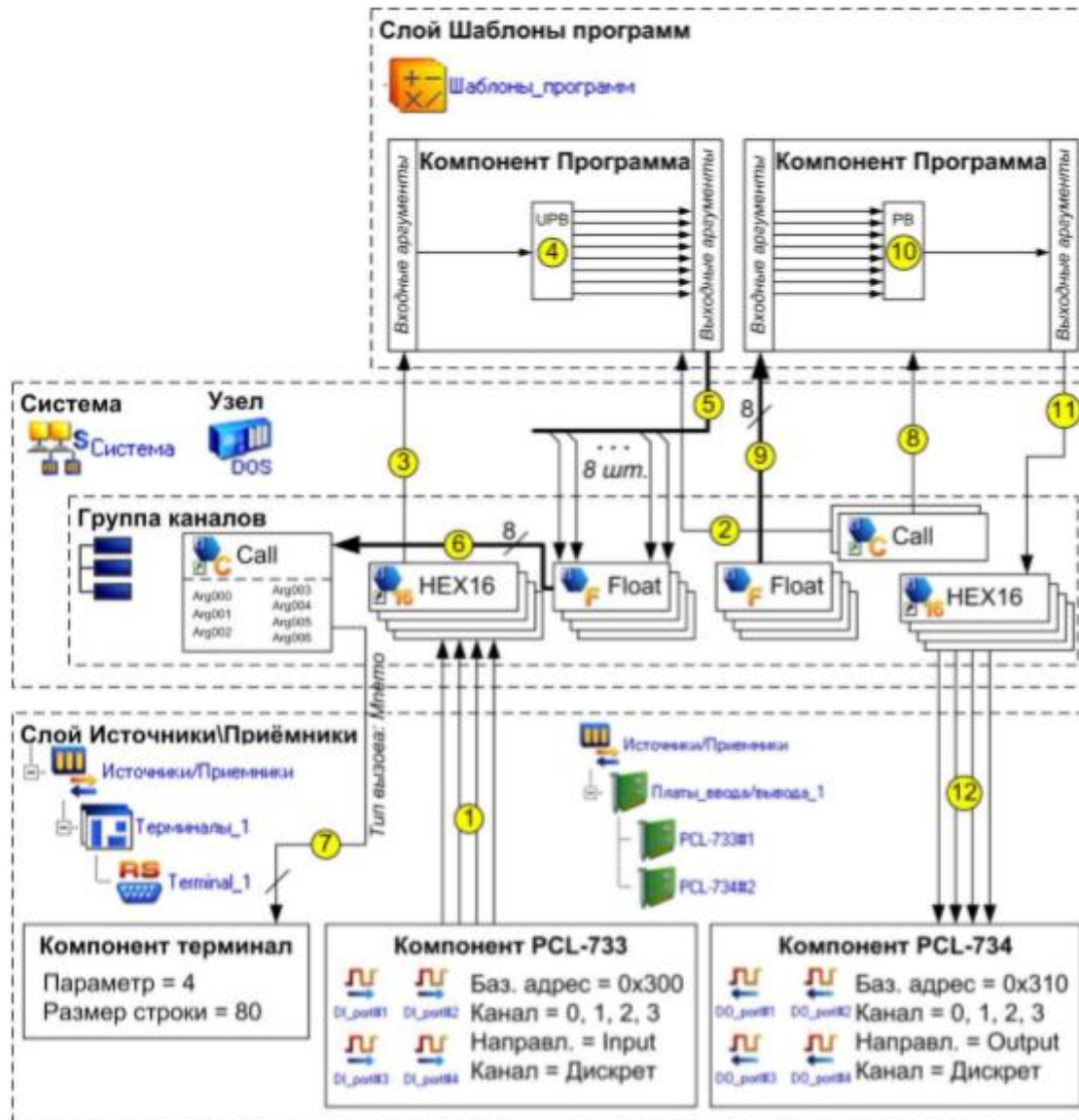


Рисунок 6. Структура Проекта, использующего компонент «Терминал». Цифрами указана последовательность операции обработки передачи данных

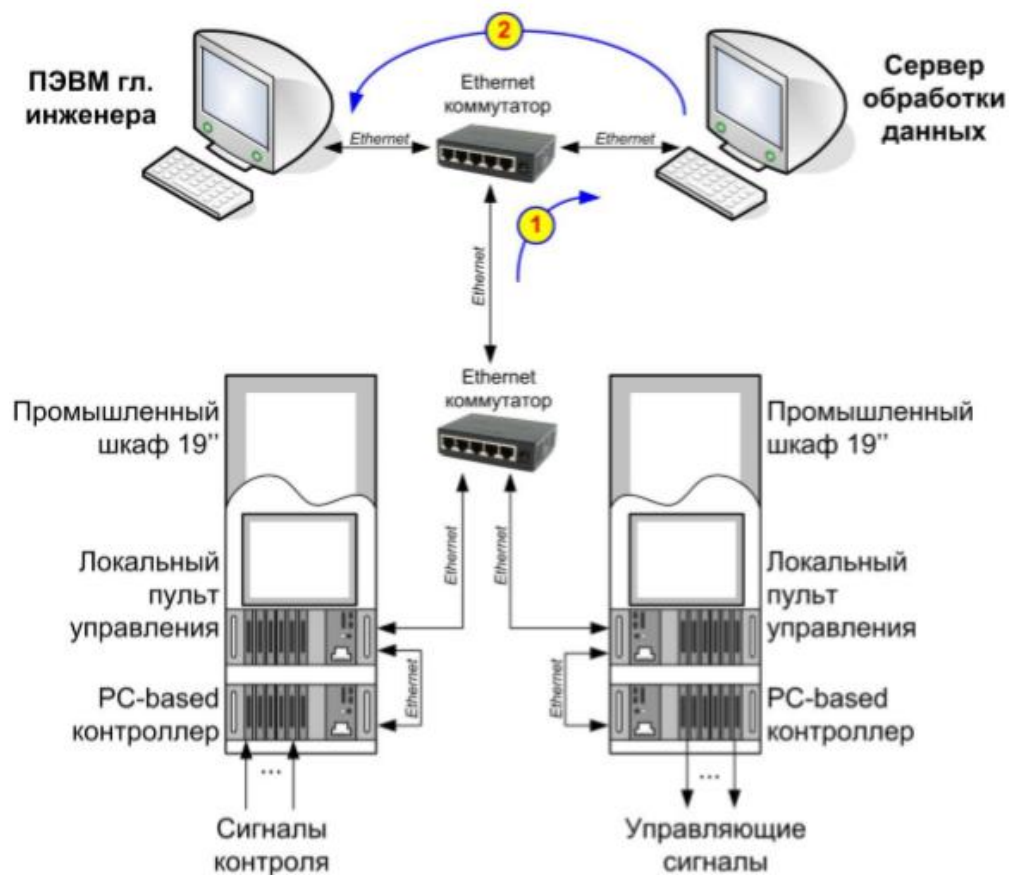
ADVANTECH PCL-733

DI_1 0	DI_9 0	DI_17 0	DI_25 0
DI_2 0	DI_10 0	DI_18 0	DI_26 0
DI_3 1	DI_11 0	DI_19 0	DI_27 0
DI_4 0	DI_12 0	DI_20 1	DI_28 0
DI_5 1	DI_13 0	DI_21 0	DI_29 0
DI_6 0	DI_14 1	DI_22 0	DI_30 0
DI_7 0	DI_15 0	DI_23 0	DI_31 0
DI_8 0	DI_16 0	DI_24 0	DI_32 0

Рисунок 7. Снимок экрана, сделанный при запуске на выполнение Проекта в микропроцессорном контроллере, работающем под управлением DOS

# УСО в распределённых АСУ

АСУ могут быть значительно распределёнными и содержать множество различных узлов, каждый из которых выполняет сбор данных. Часто обработку данных выполняют в рабочих станциях и микропроцессорных контроллерах верхнего уровня (рисунок 8).





## Настройка сетевого обмена данными

На рисунке 8 представлена группа узлов АСУ, каждый из которых содержит один или несколько микропроцессорных модулей. Пусть для каждого из узлов АСУ создаётся Узел Проекта, для которого должен быть настроен сетевой интерфейс.

При настройке сетевого взаимодействия в TRACE MODE 6 по интерфейсу Ethernet необходимо выполнить следующие действия:

1. настроить IP-адрес для каждого Узла (рисунок 9);
2. выбрать сетевой контроллер, используемый для обмена с другими Узлами Проекта;
3. настроить индивидуальный и групповой адрес Узла в Проекте;
4. выполнить привязку каналов между Узлами Проекта.

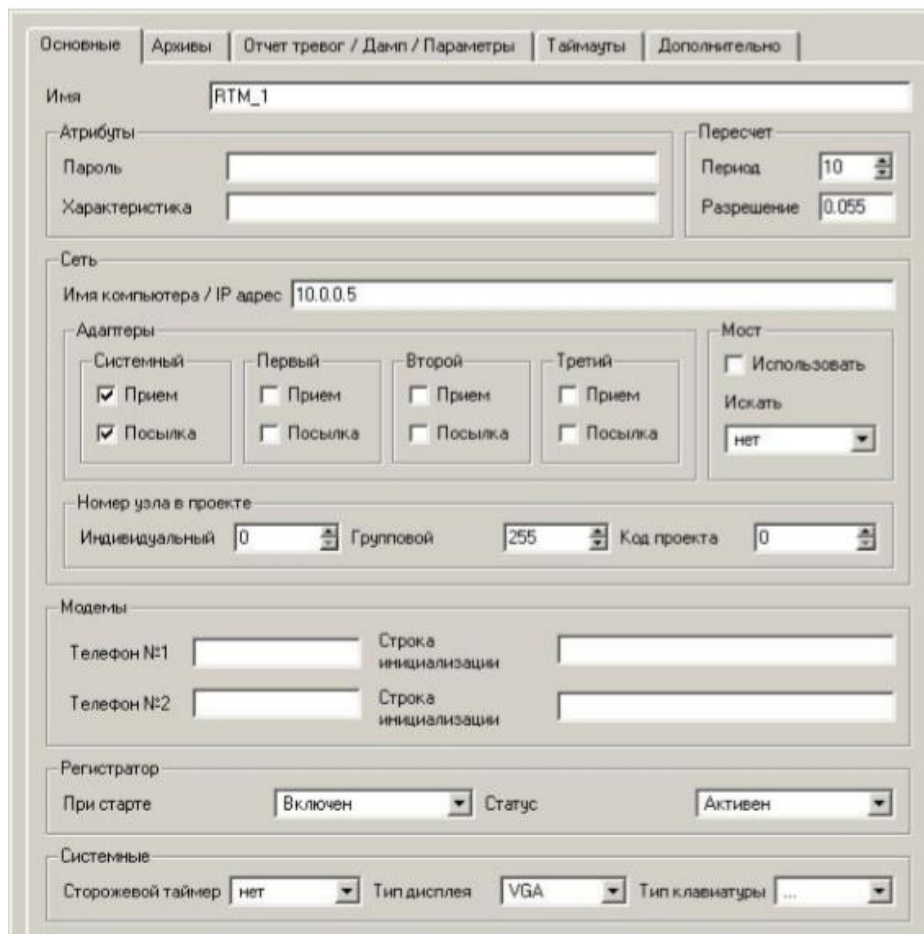


Рисунок 9. Вид окна программы «Редактор параметров узла»

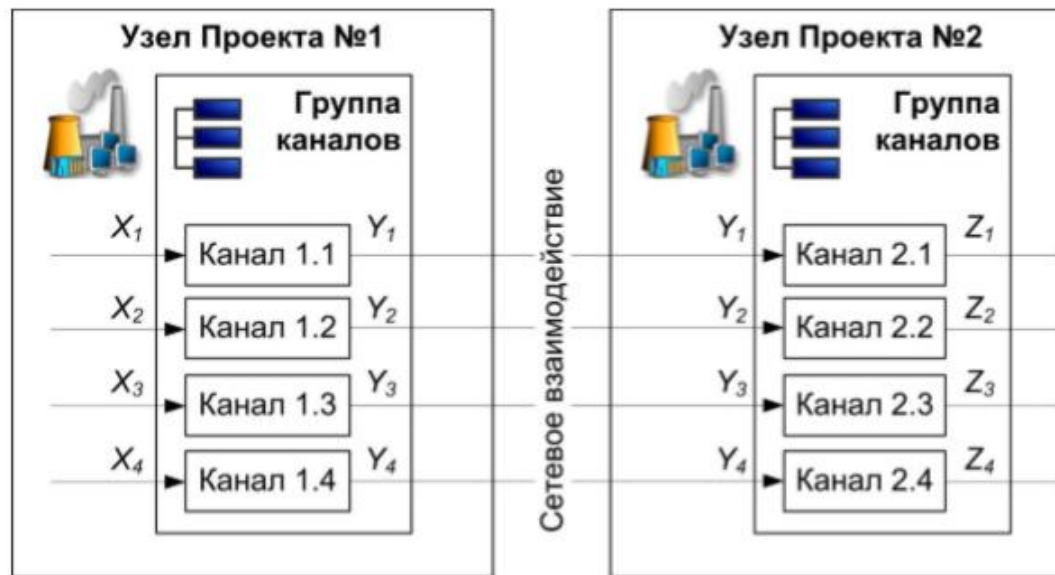


Рисунок 10. Схема сетевого взаимодействия Узлов Проекта

# Элементы индикации и управления, используемые в составе интерфейса пользователя при работе с УСО

Индикация состояний дискретных сигналов

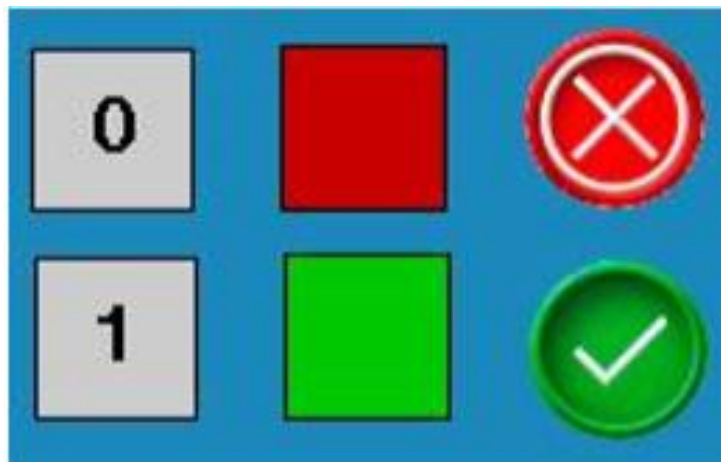


Рисунок 11. Способы индикации состояния входов устройства ввода дискретных сигналов

## Индикация значений аналоговых сигналов

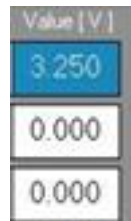


Рисунок 13. Изображение текстового индикатора значения аналогового сигнала с цветовой сигнализацией того, что значение отлично от нуля

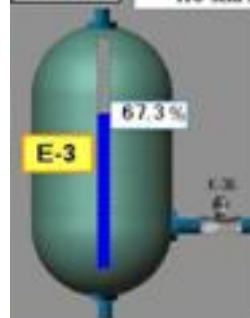


Рисунок 14. Изображение текстового индикатора значения аналогового сигнала с функцией перемещения по вертикали в зависимости от уровня жидкости в баке



Рисунок 15. Изображение графического элемента «Задатчик уставки» типа «Ползунок» (слева) и изображения графических индикаторов значения аналогового сигнала справа и по центру

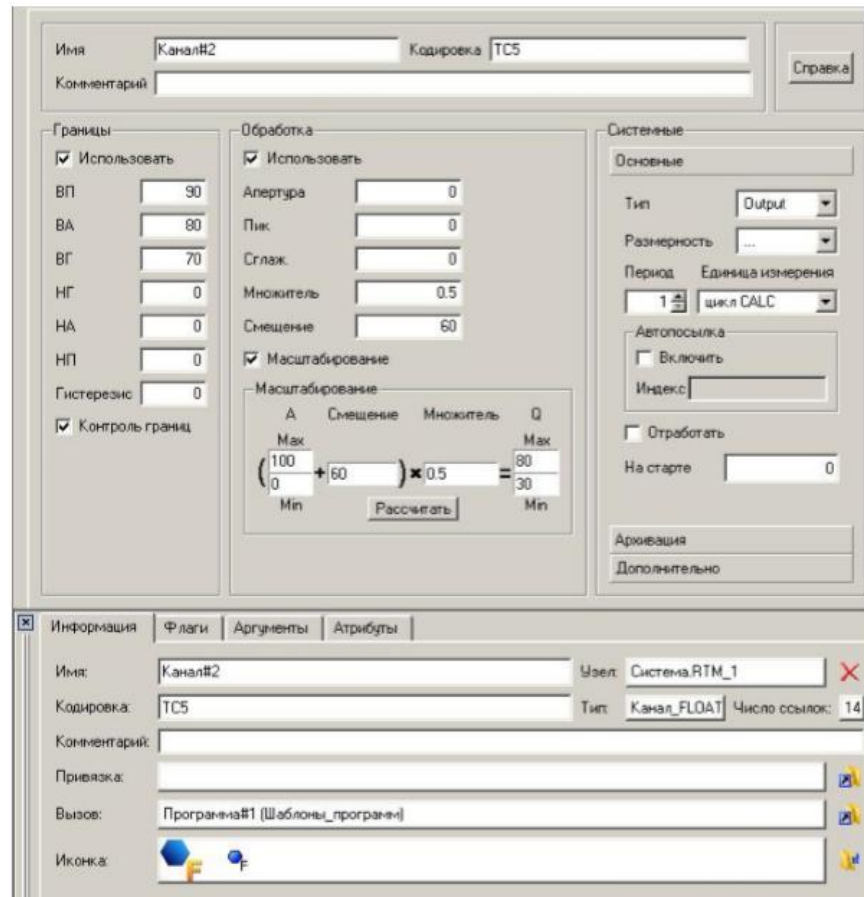


Рисунок 16. Вид окна редактора параметров канала класса FLOAT в Проекте-примере

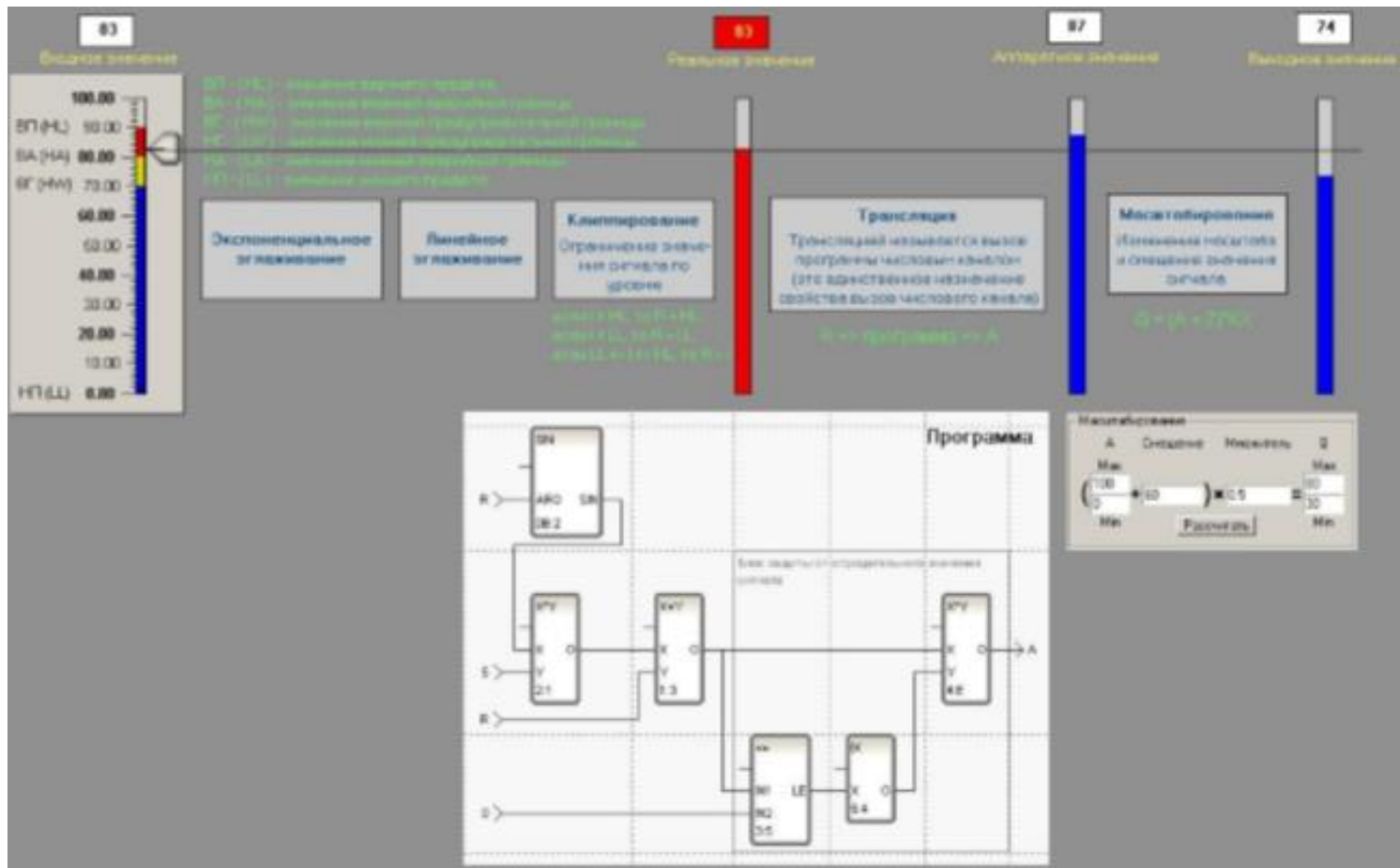


Рисунок 17. Вид окна программы-примера