

**Министерство науки и высшего образования РФ**  
**Федеральное государственное автономное образовательное учреждение**  
**высшего образования**  
**«Казанский (Приволжский) Федеральный университет»**

**КАЗАНСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**  
**ИНСТИТУТ ФИЗИКИ**  
*Кафедра радиоэлектроники*

**М.Н.ОВЧИННИКОВ, А.Г.ГАВРИЛОВ, А.И.ДЕРКАЧ,**  
**В.А.МАЦЕНКО, А.О.ХРАМОВ**

**МЕТОДИЧЕСКОЕ ПОСОБИЕ ПО ИЗУЧЕНИЮ ЯЗЫКА**  
**МАКРОСОВ ДЛЯ ГРАФИЧЕСКОЙ ПАНЕЛИ «WT3010»**

**Учебно-методическое пособие**

**Казань – 2020**

*Принято на заседании кафедры радиоэлектроники  
Протокол № 5 от «09» декабря 2020 г*

**Рецензент:**

канд. физ.-мат. наук, зав. лаб. **Я.В. Фаттахов**

**Овчинников М.Н.**

**Методическое пособие по изучению языка макросов для графической панели «WT3010»** Учебно – методическое пособие / М.Н. Овчинников, А.Г. Гаврилов, А.И. Деркач, В.А. Маценко, А.О. Храмов– Казань: Казанский (Приволжский) федеральный университет, 2020. – 55с.

Настоящее учебно-методическое пособие адресовано студентам – бакалаврам, магистрантам и аспирантам, обучающимся по профилям «Радиофизика», «Радиофизические измерения», «Разработка и эксплуатация нефтяных месторождений». Учебное пособие будет интересно также широкому кругу читателей, занимающихся проблемами проектирования, разработки и отладки автоматизированных систем контроля (АСК) в различных областях применения: в нефтяной и газовой промышленности, на транспорте, в энергетике, машиностроении, коммунальном хозяйстве.

Знания и практические навыки, полученные в Лаборатории подземной гидродинамики, где находится учебный лабораторный комплекс, в области проектирования и создания автоматизированных систем управления производствами на базе современных комплектующих мировых ведущих производителей и стандартизованных, принятых в мировой практике языков программирования МЭК61131–13, будут востребованы в различных областях науки и производства. В процессе выполнения рассмотренных в учебном пособии лабораторных работ студенты знакомятся с азами программирования в среде разработки Weintek EasyBuilder 8000, учатся создавать и редактировать простейшие программные продукты (приложения) для графической панели WT3010.

© **Овчинников М.Н., Гаврилов А.Г.,  
Деркач А.И., Маценко В.А., Храмов А.О., 2020**  
© **Казанский университет, 2020**

## ОГЛАВЛЕНИЕ

Введение.	3
Глава 1. Изучение принципов взаимодействия объектов в среде разработки Weintek EasyBuilder 8000.	4
1.1 Теоретические сведения.	
1.2 Практическое занятие: ввод и отображение чисел, оформление объектов, переключение между окнами, использование функциональных клавиш.	
1.3 Задание для самостоятельной работы.	
Глава 2. Изучение перевода данных из одного формата в другой в среде Weintek EasyBuilder 8000.	26
2.1 Теоретические сведения.	
2.2 Практическое занятие: создание проекта «Ввод/вывод чисел с разными типами данных».	
2.3 Задание для самостоятельной работы.	
Глава 3. Знакомство с макросами в среде Weintek EasyBuilder 8000.	33
3.1 Теоретические сведения.	
3.2 Практическое занятие.	
3.3 Задание для самостоятельной работы.	
Глава 4. Выполнение арифметических операций внутри макросов.	39
4.1 Теоретические сведения.	
4.2 Практическое занятие: создание проекта с применением макроса.	
4.3 Задание для самостоятельной работы.	
Глава 5. Выполнение арифметических операций в макросе над числами, вводимыми с «ползунков».	42
5.1 Теоретические сведения.	
5.2 Практическое занятие: создание проекта с применением чисел, вводимых с помощью «ползунка».	
5.3 Задание для самостоятельной работы.	
Список литературы.	55

## ВВЕДЕНИЕ

Человеко-машинный интерфейс (ЧМИ) - набор технических средств, обеспечивающих взаимодействие между человеком и машинным оборудованием и позволяющих человеку управлять работой оборудования.

Панели оператора Weintek являются устройствами ЧМИ и представляют собой компактные устройства, предназначенные для реализации операторского интерфейса. В процессе выполнения рассмотренных в учебном пособии лабораторных работ студенты знакомятся с азами программирования в среде разработки Weintek EasyBuilder 8000, учатся создавать и редактировать простейшие программные продукты (приложения) для графической панели WT3010.

Актуальность данной тематики связана с развитием автоматизации технологических процессов и ростом потребности в устройствах, с помощью которых человек может контролировать функционирование автоматизированного оборудования.

Предлагаемое учебное пособие состоит из пяти глав, посвященных ознакомлению обучающихся с интерфейсом среды Weintek EasyBuilder 8000 в рамках разработки приложений для графической панели WT3010.

## **Глава 1. Изучение принципов взаимодействия объектов в среде разработки Weintek EasyBuilder 8000.**

### **Назначение:**

- Ознакомление со средой Weintek EasyBuilder 8000 для разработки приложений графической панели WT3010.
- Изучение принципов взаимодействия объектов в среде Weintek EasyBuilder 8000.

### **Цель работы:**

Освоение практических навыков:

- Создания и оформления объектов,
- Ввода/вывода чисел,
- Работы с окнами в среде Weintek EasyBuilder 8000.

### **Используемое инструментальное ПО:**

- Среда разработки Weintek EasyBuilder 8000.

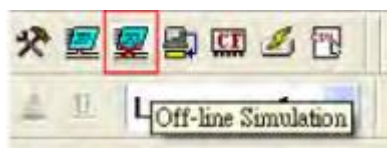
### **Теоретические сведения:**

#### **• Имитация проекта на ПК**

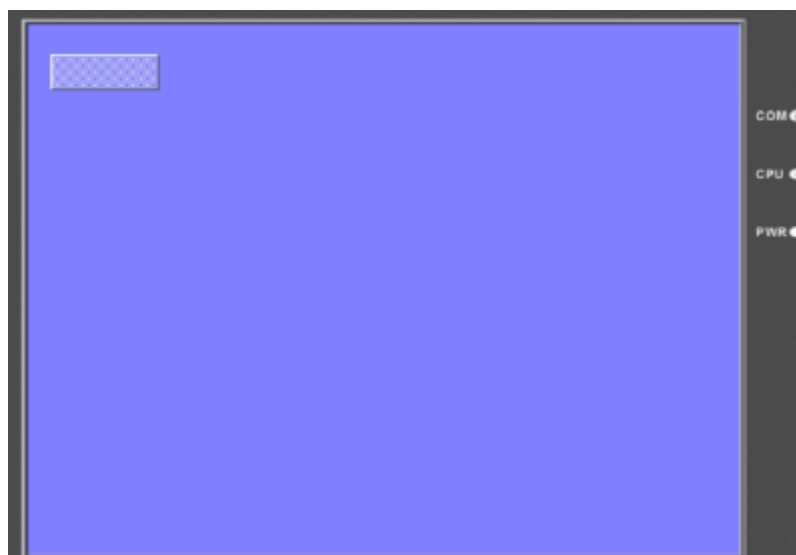
Существует два режима имитации: Off-line simulation и On-line simulation. В первом режиме ПК имитирует работу ПЛК без соединения с ним. Во втором режиме наоборот, имитация выполняется при наличии соединения с ПЛК и точно настроенных параметрах соединения.

Если при имитации на компьютере происходит управление локальным ПЛК (т.е. непосредственно подключенным к компьютеру), накладывается 10-минутное ограничение.

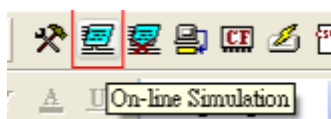
Щелкните пиктограмму для выполнения имитации в режиме [Off-line Simulation].



После выполнения имитации экран выглядит следующим образом:

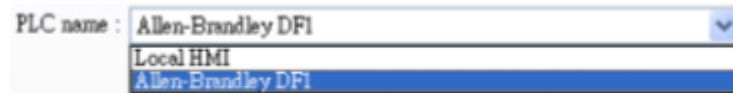


Для выполнения имитации в режиме [On-line Simulation] кликните соответствующую пиктограмму для выполнения процесса после корректного подключения устройства.

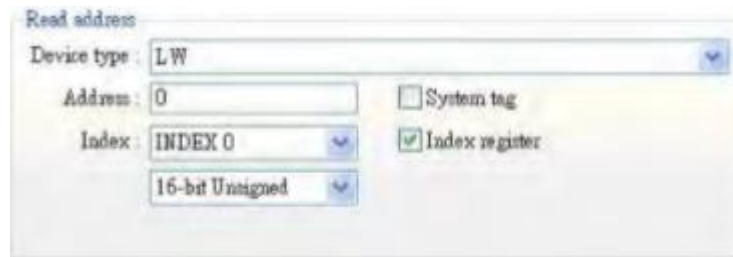


- **Выбор соединения с ПЛК**

При использовании некоторых объектов требуется выбор соединения с ПЛК. На рисунке ниже показан раскрывающийся список [PLC name] для указания имени устройства, с которым устанавливается соединение: для выбора доступны два устройства: “Local HMI” и “AllenBrandley DF1”. В этот список попадают устройства, заданные в таблице устройств “device table” в окне “System Parameters Settings”.



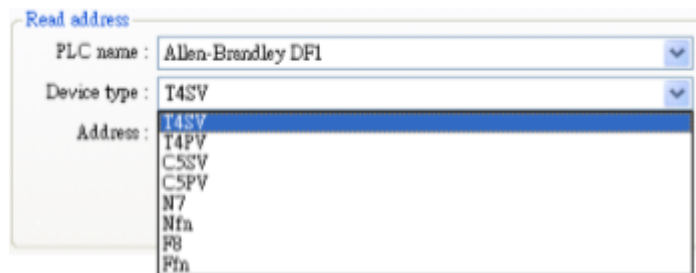
### Установка адресов для считывания и записи



На рисунке выше показано, что настройка адресов чтения и записи включает задание следующих параметров:

#### [Device type]

В списке [Device type] для разных устройств будут различные варианты выбора.



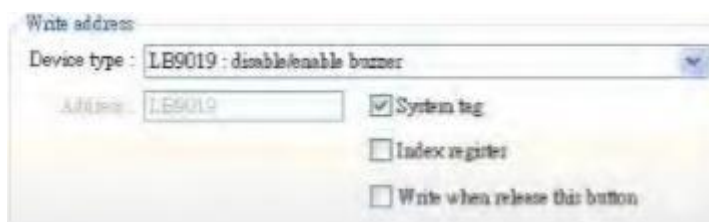
#### [Address]

Установка адресов для чтения и записи данных.

#### [System tag]

Метка адреса может быть системной или определяемой пользователем. Системная метка [System tag], включая метку адреса бита и слова, предназначена для резервирования адресов с определенными целями. Если отмечен флаг [System tag], то помимо того, что в поле [Device type] будет

отображаться содержимое системной метки, в поле [Address] показан адрес выделенной метки (см. рис. ниже).

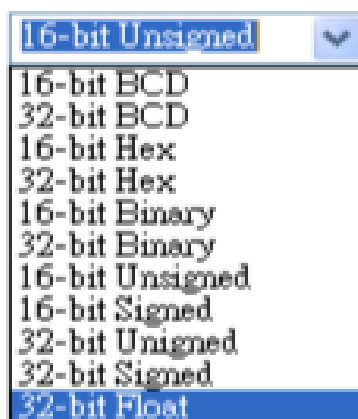


Ниже показано содержание системной метки для адреса бита и для адреса машинного слова соответственно. Подробнее смотрите примеры в параграфе «Библиотека меток».

### [Index register]

Обращайтесь к примерам в параграфе «Регистр приращения адреса» за сведениями, когда нужно отмечать данный флаг, когда — нет.

EV 8000 поддерживает следующие типы чисел. Необходимо выбрать правильный тип числа, особенно при использовании адресных меток.



- **Numeric Input – Цифровой вход и Numeric Display – Цифровой дисплей**

### Краткое описание

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

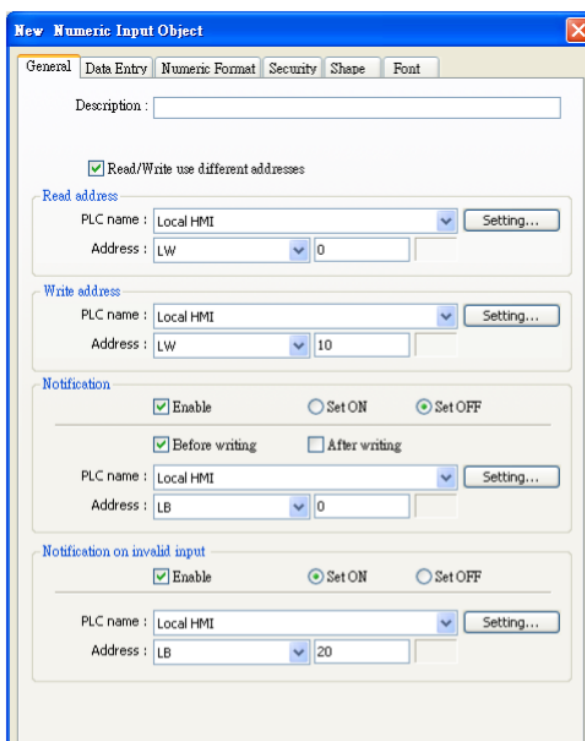


## Настройка



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

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



## Read/Write use different address

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

### Read address

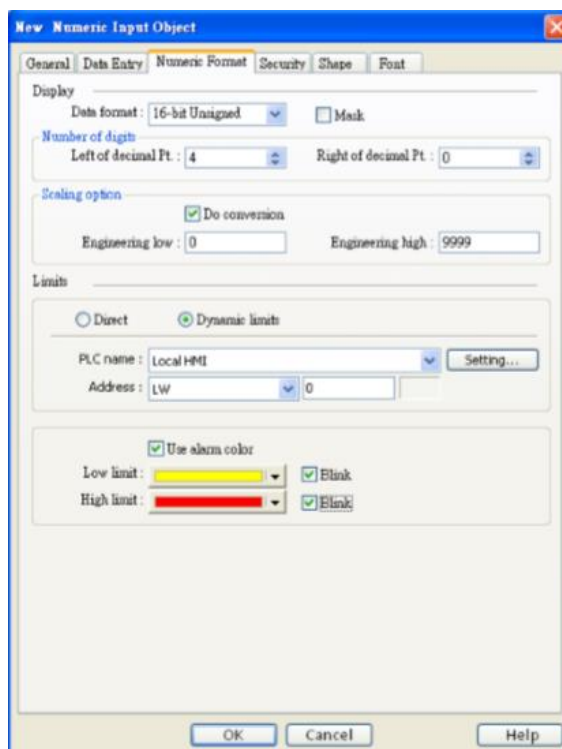
В полях [PLC name] [device type][address] укажите адрес слова, содержимое которого будет отображаться и куда будет записываться новое значение.

### Write address

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

### [Numeric Format]

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



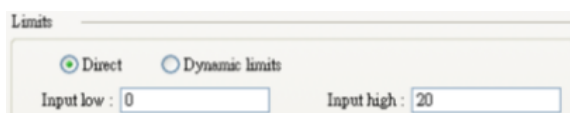
### Number of Digits – Число цифр

[Left of decimal Pt]

Число знаков в целой части, до десятичной запятой (точки).

[Right of decimal Pt]

Число знаков после десятичной запятой.



### [Direct limits] – Непосредственные пределы

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

#### • **Работа с окнами**

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

### Создание окна

Существует два способа создания окна:

Первый заключается в выборе номера в дереве окон и щелчок правой кнопкой мыши. В открывшемся контекстном меню нужно выбрать пункт [New] и щелкнуть ОК после завершения всех настроек.

Другой способ создания окна: выбрать пункт [Open Window] в меню [Window] — откроется диалоговое окно [Open Window].

### Настройки окна

В EB8000 есть 2 способа редактирования свойств окна:

- a. Щелкните правой кнопкой мыши на нужном окне в дереве окон и выберите пункта [Settings] для изменения свойств окна.
- b. Выберите пункт [Open Window] в основном меню — появится диалоговое окно [Open Window]. Нажмите [Settings] для изменения свойств окна.

### Открытие, закрытие и удаление окна

Для открытия существующего окна можно использовать: двойной щелчок на нужной строчке в дереве окон или контекстное меню (открывается щелчком правой кнопкой мыши) и выбором пункта [Open].

Аналогично выполняется закрытие или удаление существующего окна.

Обратите внимание, что удаляемое окно должно быть закрыто!

- **Объект [Function Key]**

### Обзор

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

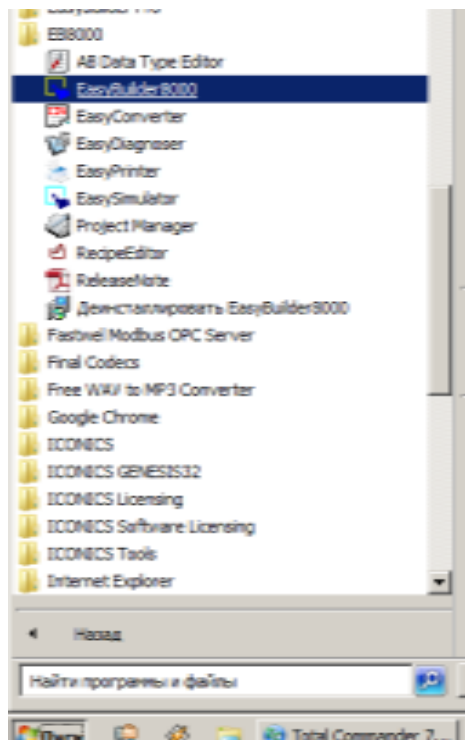
### Задание свойств

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



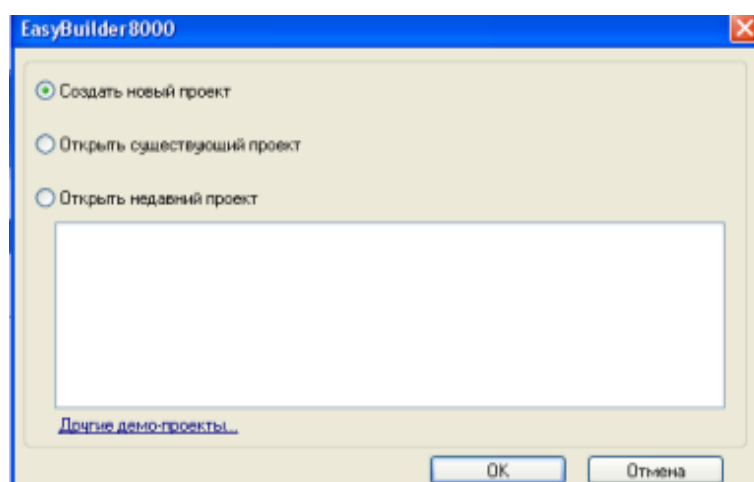
## Практическое занятие 1 Ввод и отображение чисел

1.1. Средствами операционной системы Windows 7 загрузить на исполнение редактор проектов EasyBuilder8000 (рис.1) : **Пуск – Все программы – EB8000 – EasyBuilder8000**



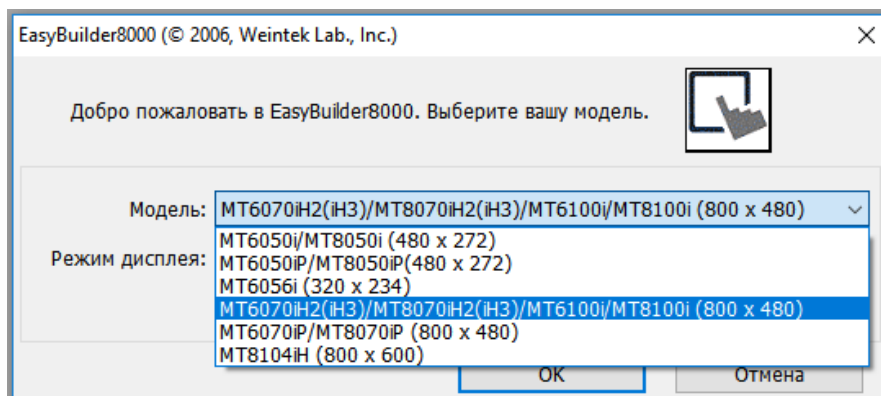
**Рис. 1. Загрузка редактора проектов EasyBuilder8000**

1.2. В появившейся диалоговой панели (рис. 2) выбрать пункт «Создать новый проект». Нажать кнопку **ОК**.



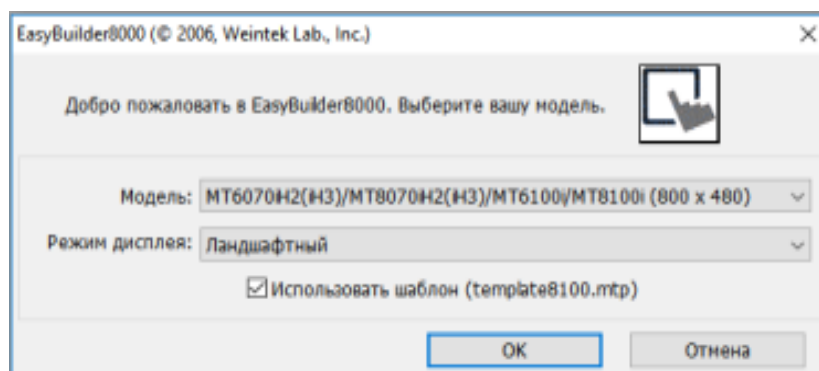
**Рис. 2. Создание нового проекта**

- 1.3. В появившемся диалоговом окне раскрыть спускающийся список моделей графических панелей Weitek. Выбрать строку, в которой присутствует панель WT3010, как показано на рис.3.



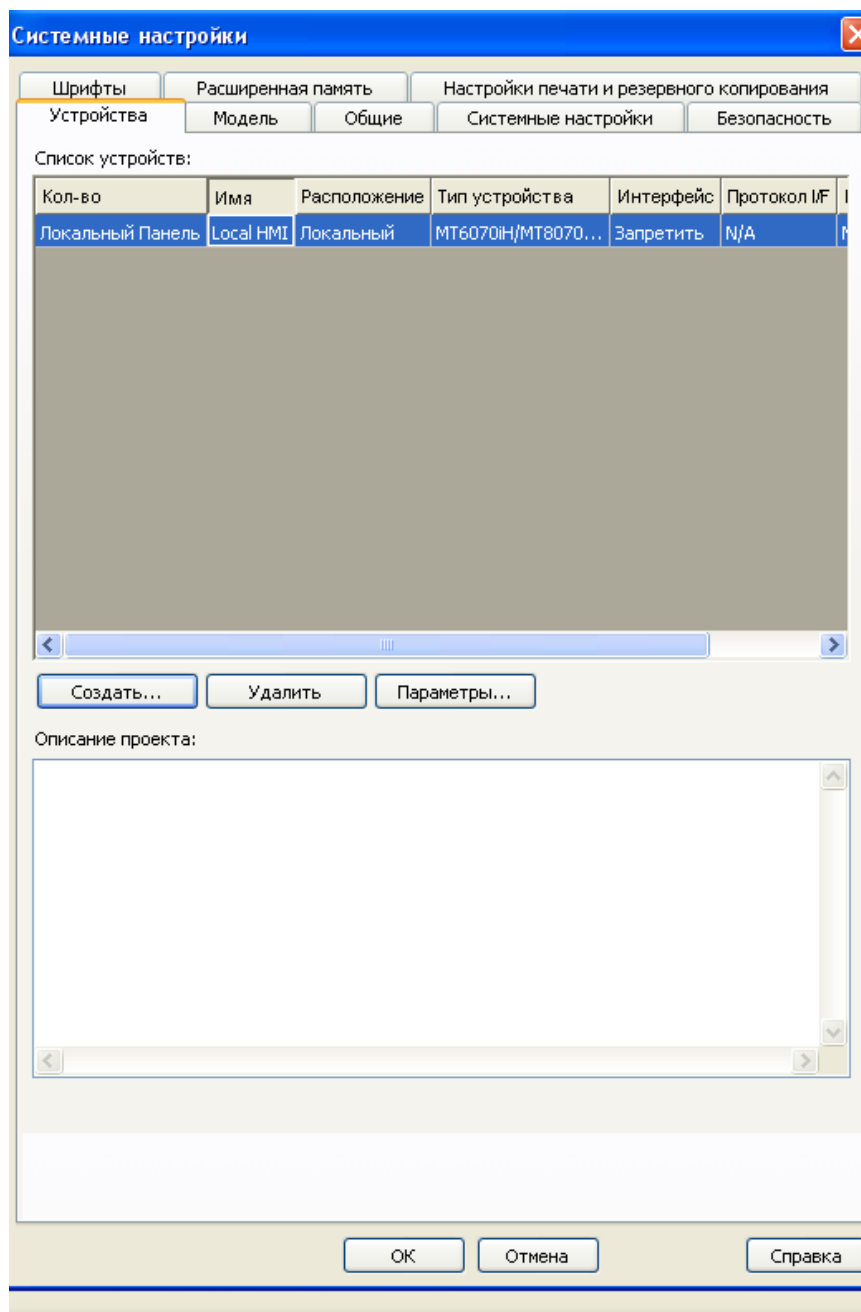
**Рис. 3. Выбор модели графической панели**

- 1.4. В том же окне выбрать режим дисплея – «Ландшафтный» (горизонтальная ориентация экрана), как показано на рис. 4. Нажать кнопку **ОК**.



**Рис. 4. Выбор режима дисплея**

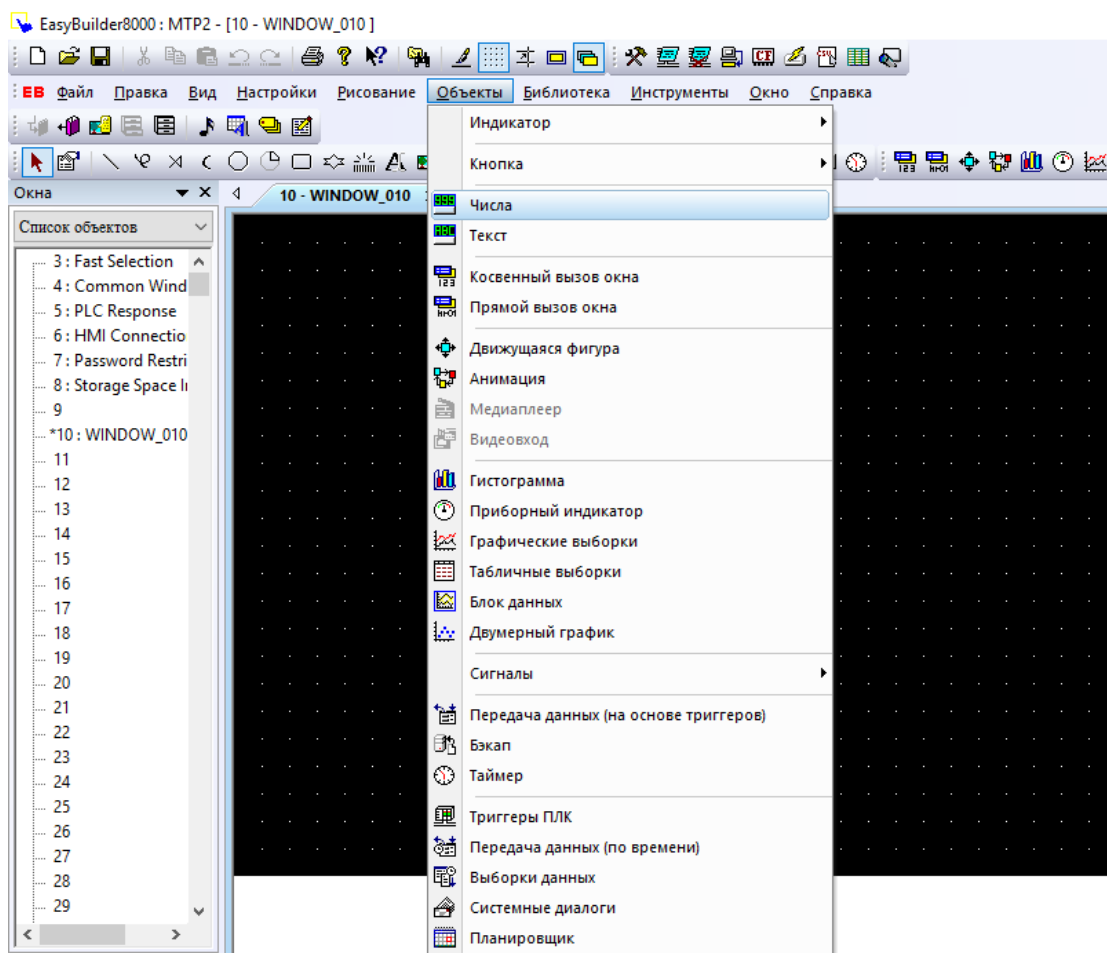
- 1.5. Появляется окно редактирования системных настроек (рис.5). Нажать кнопку **ОК**.



**Рис. 5. Окно редактирования системных настроек**

1.6. Сохранить созданный проект под именем **Lab\_1.pro** на локальном диске в папке C:\Лаборатория гидродинамики \Labs\. При необходимости можно в название файла внести фамилию студента и номер учебной группы, например **Lab\_1\_Ivanov\_5101.pro**

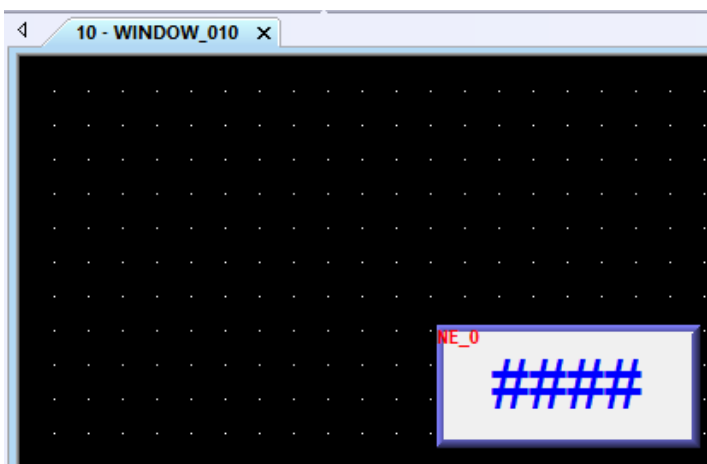
1.7. В пункте меню объекты выбрать пункт **Числа** (Рис. 6)



**Рис. 6. Пункт меню Объекты**

1.8. Откроется окно атрибутов объекта Числа. В этом окне нажать кнопку ОК.

1.9. Расположить созданный элемент в рабочей зоне (Рис. 7). Двойным щелчком левой кнопки мыши по объекту открыть его атрибуты.





## Рис. 7. Объект Числа

1.10. На вкладке **Общие** (Рис. 8) необходимо установить адрес ячейки памяти, куда будет записываться значение числа, и с какого адреса оно будет считываться. Local HMI - это локальная память панели, LW 0, 1, 2,..., - адреса ячеек этой памяти. Возьмем адрес LW 0. Установить галочку **Разрешить ввод**.

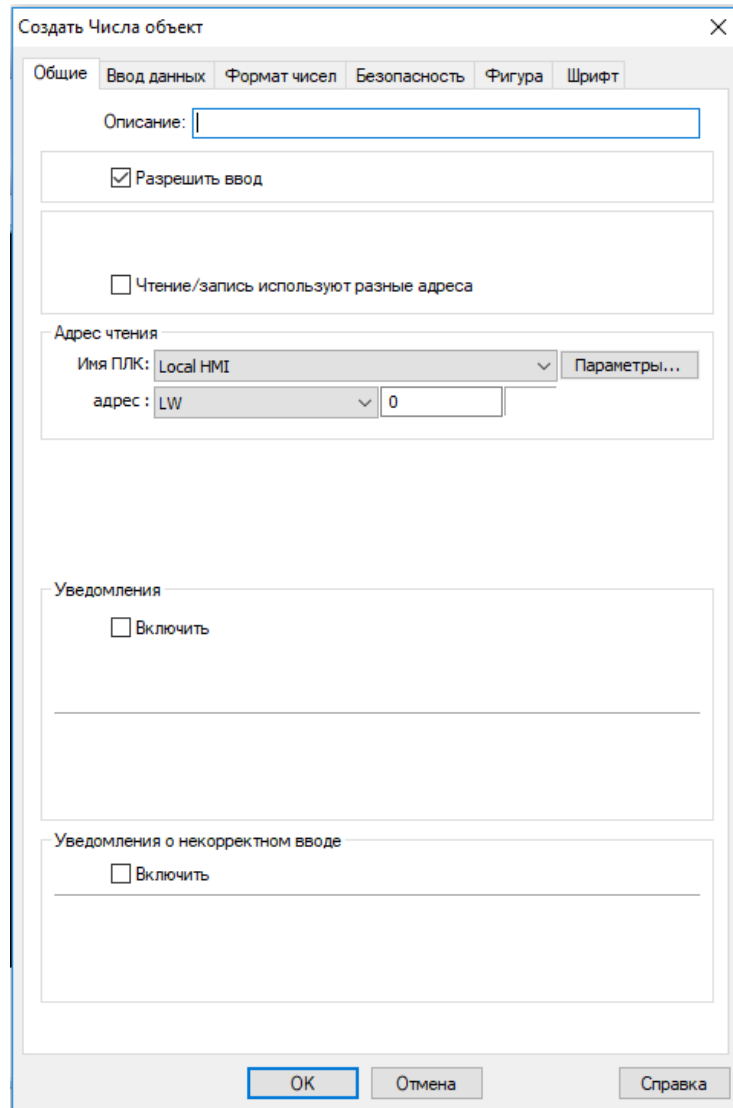
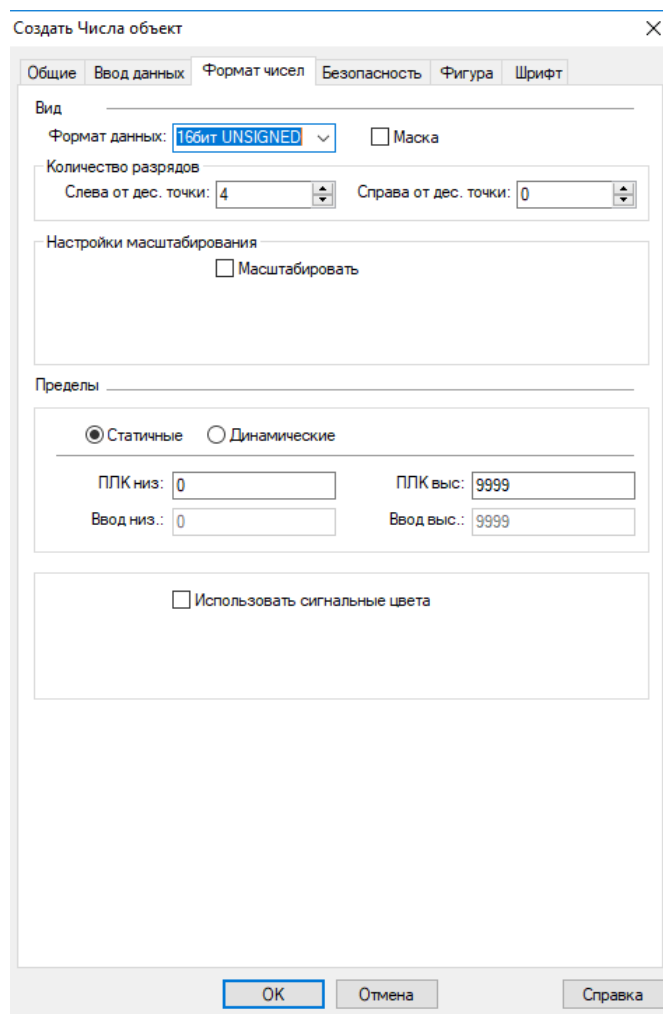


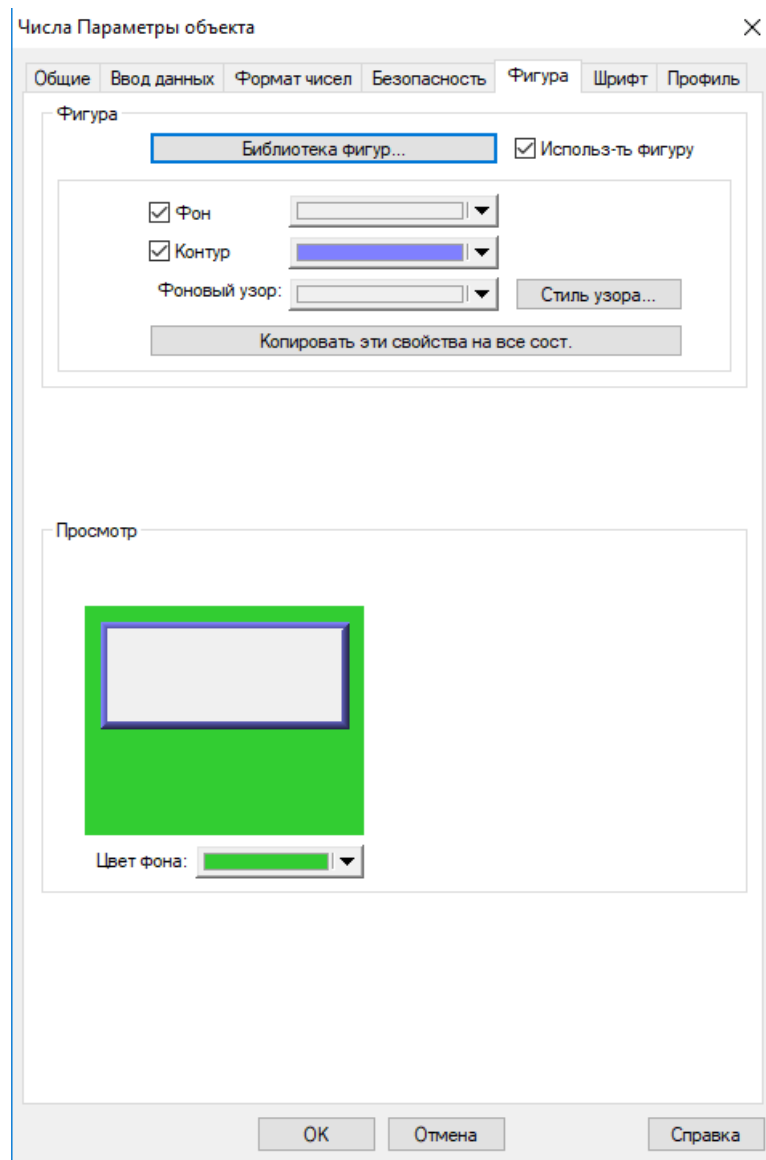
Рис. 8. Окно атрибутов объекта Числа. Вкладка Общие

1.11. Перейти на вкладку **Формат чисел** (Рис. 9). Здесь мы можем выбрать формат данных (см. Теоретические сведения), выберем 16бит UNSIGNED. Выбираем количество разрядов слева (возьмем 4) и справа (возьмем 0) от десятичной точки.



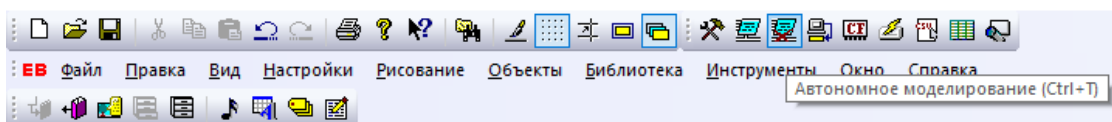
**Рис. 9. Вкладка Формат чисел**

1.12. В этом же окне перейти на вкладку Фигура (Рис. 10). Здесь мы можем выбрать, как будет выглядеть наш числовой дисплей. В библиотеке фигур можно выбрать готовые шаблоны, в пунктах Фон, Контур можно выбрать соответствующие цветовые характеристики дисплея или убрать их (для этого нужно просто убрать галочки напротив этих пунктов). Нажать кнопку ОК.



**Рис. 10. Вкладка Фигура**

1.13. Щелкнуть по пиктограмме [Автономное моделирование] (Рис. 11).



**Рис. 11. Выбор режима Автономного моделирования**

1.14. Откроется окно (Рис. 12), в котором будет видно наше числовое поле. Щелкнув, по нему можно открыть клавиатуру, с которой вводятся значения.

Ввод числа осуществляется нажатием кнопки ENT, закрытие клавиатуры - ES.



Рис. 12. Окно автономного моделирования

## Практическое занятие 2 Оформление объектов

2.1. В пункте меню Рисование выбрать пункт Прямоугольник (Рис. 13).

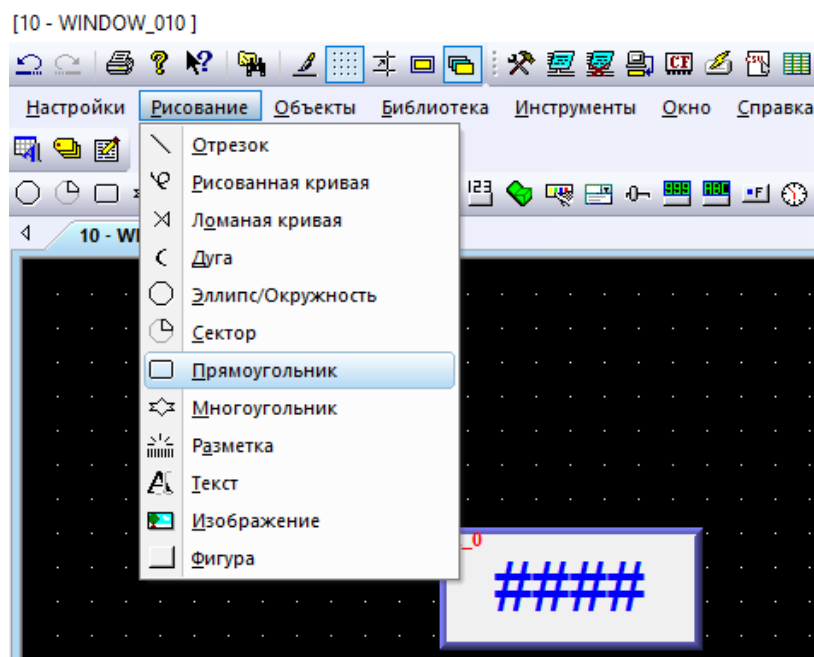
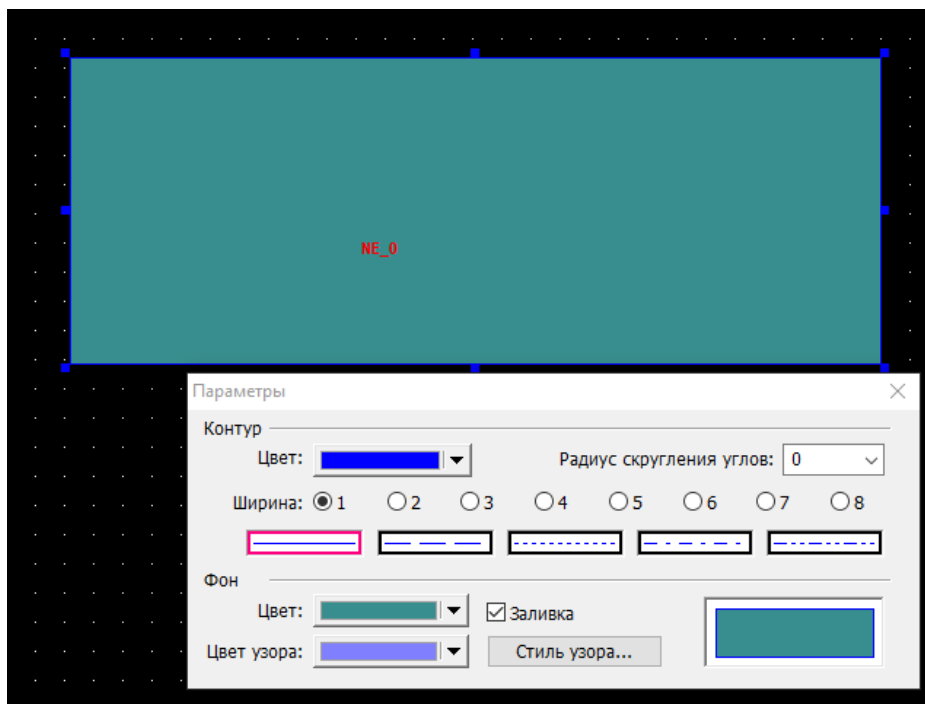


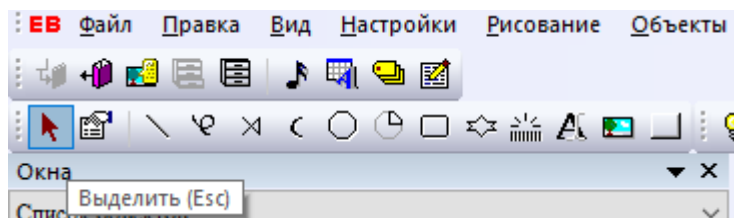
Рис. 13. Пункт меню рисование

2.2. Появится окно параметров прямоугольника, в котором можно задавать его цвет, контур и параметры углов (Рис. 14). Щелкнуть левой кнопкой мыши в рабочей зоне. Начнется рисование прямоугольника. Растянуть его до нужных размеров и снова щелкнуть левой кнопкой мыши.



**Рис. 14. Выбор параметров прямоугольника**

2.3. Щелкнуть по пиктограмме Выделить (Рис. 15). Навести курсор мыши на получившийся прямоугольник и, нажав и удерживая левую кнопку мыши, перетащить его на середину рабочей зоны.



**Рис. 15. Выбор пиктограммы Выделить в верхнем меню окна EasyBuilder 8000**

2.4. В пункте меню Правка выбрать пункт Порядок. В развернувшемся списке выбрать пункт На задний план (Рис. 16). После этого нарисованный прямоугольник будет находиться за числовым полем.

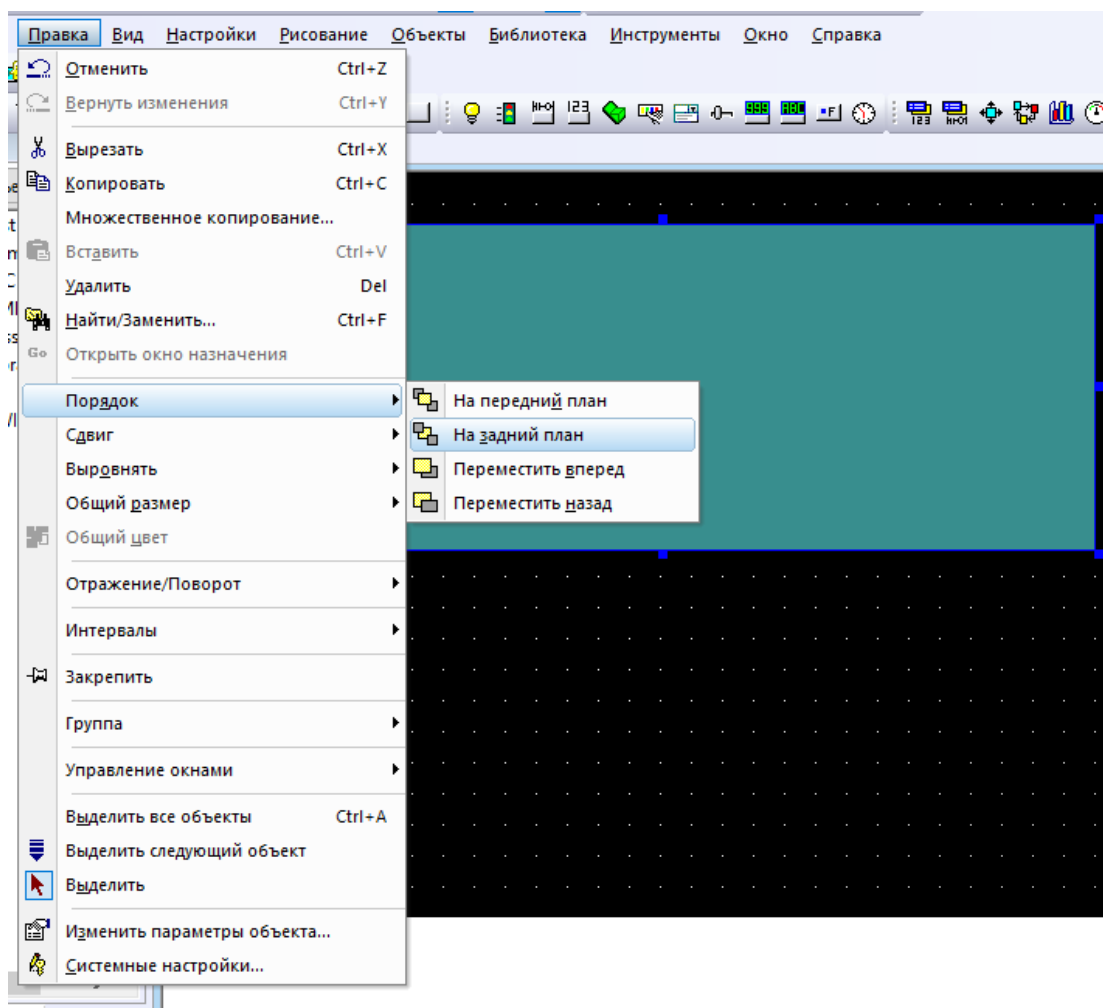


Рис. 16. Пункт меню правка

2.5. Щелкнуть по пиктограмме Текст (Рис. 17).

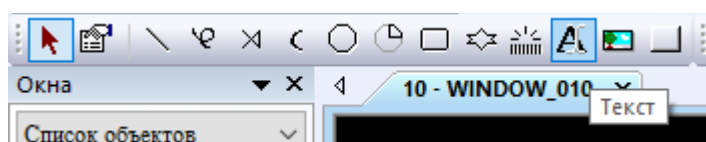
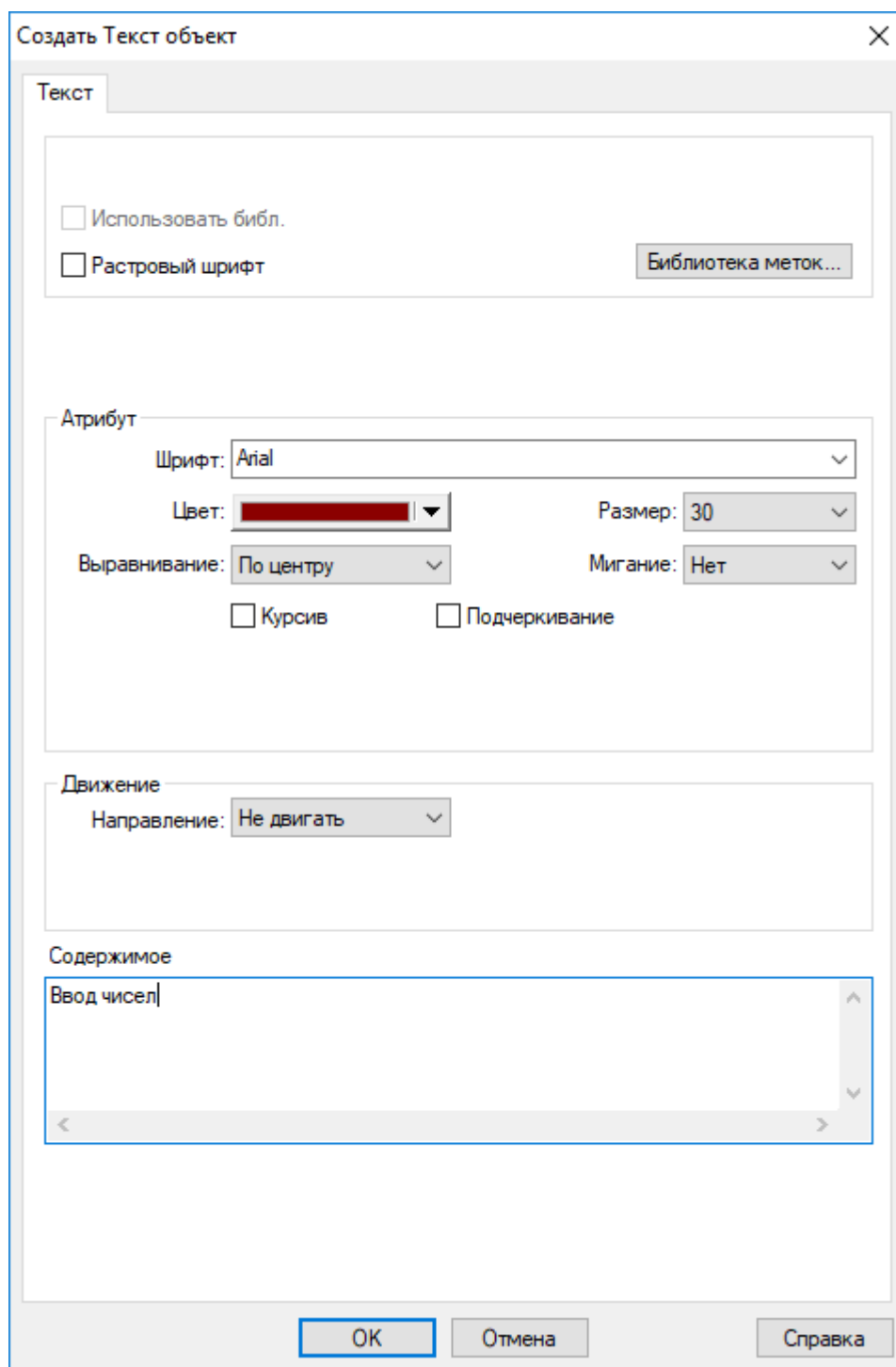


Рис. 17. Выбор пиктограммы Текст в верхнем меню окна программы

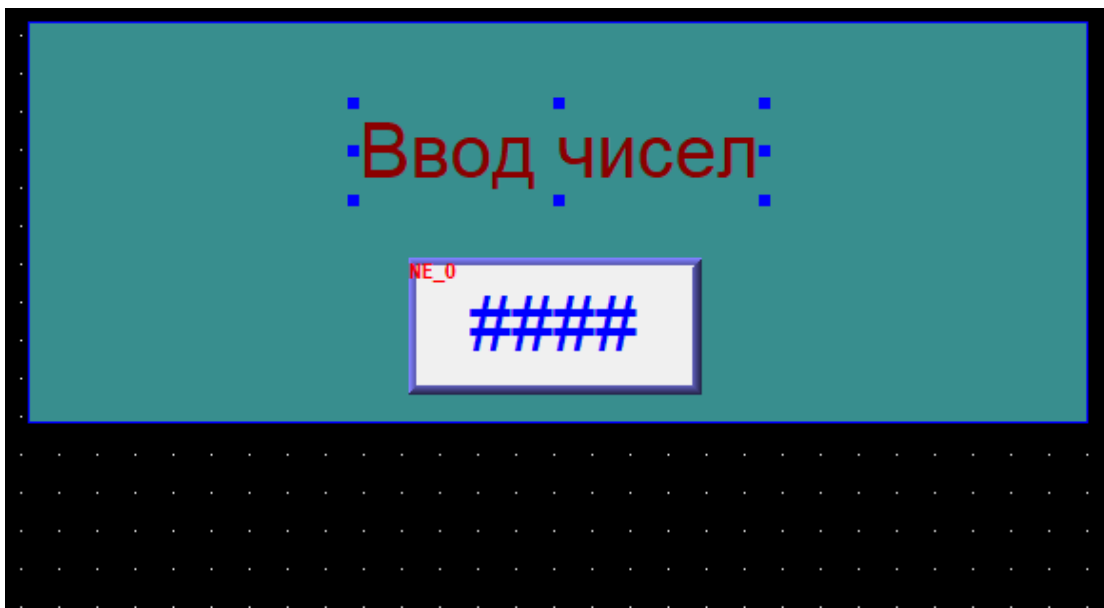
2.6. Откроется окно атрибутов текста, в котором мы можем выбрать шрифт, цвет и размер текста, его выравнивание и другие параметры (Рис. 18).

В нижней части окна в поле Содержимое ввести текст. Выбрать его параметры. Нажать ОК.



**Рис. 18. Окно атрибутов текста**

2.7. Курсором мыши перетащить получившуюся область текста на центр нарисованного прямоугольника, после чего щелкнуть левой кнопкой мыши (Рис. 19).



**Рис. 19. Объекты Числа, Текст и Прямоугольник в рабочей зоне**

2.8. Запустить проект, щелкнув по пиктограмме Автономное моделирование. В открывшемся окне пронаблюдать нарисованную рамку, введенный текст, проверить ввод числа (Рис. 20). Закрыть окно, нажав крестик.



**Рис. 20. Окно автономного моделирования**



## Практическое занятие 3 Переключение между окнами. Использование функциональных клавиш

3.1. В дереве объектов в левой части экрана создать окно. Для этого выбрать любое из пустых окон в списке (возьмем Окно 11), щелкнуть по нему правой кнопкой мыши и в открывшемся меню выбрать пункт Создать (Рис. 21).

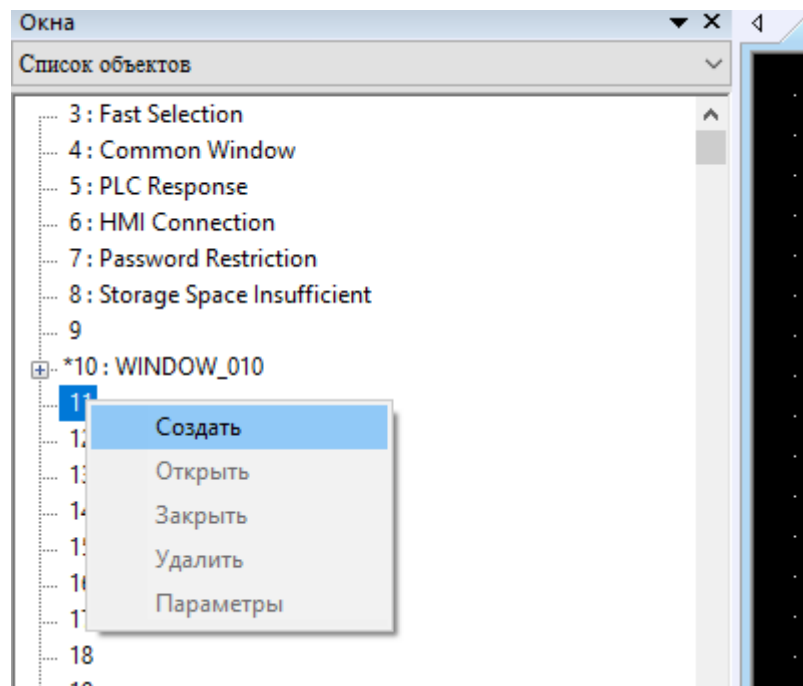
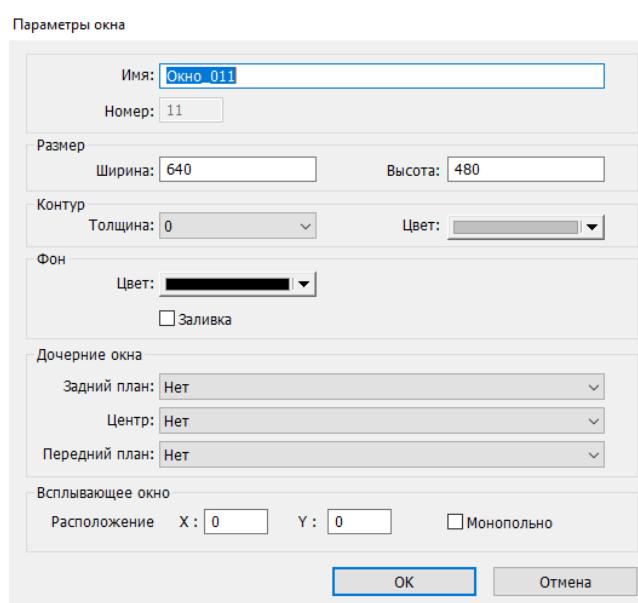


Рис. 21. Дерево объектов

3.2. Откроются Параметры окна (Рис. 22). Здесь можно выбрать размеры окна, параметры контура, его цвет. Нажать ОК.

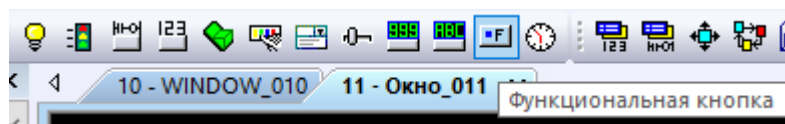


## Рис. 22. Параметры окна

3.3. Перейти в Окно 11, дважды щелкнув левой кнопкой мыши по его названию в дереве объектов.

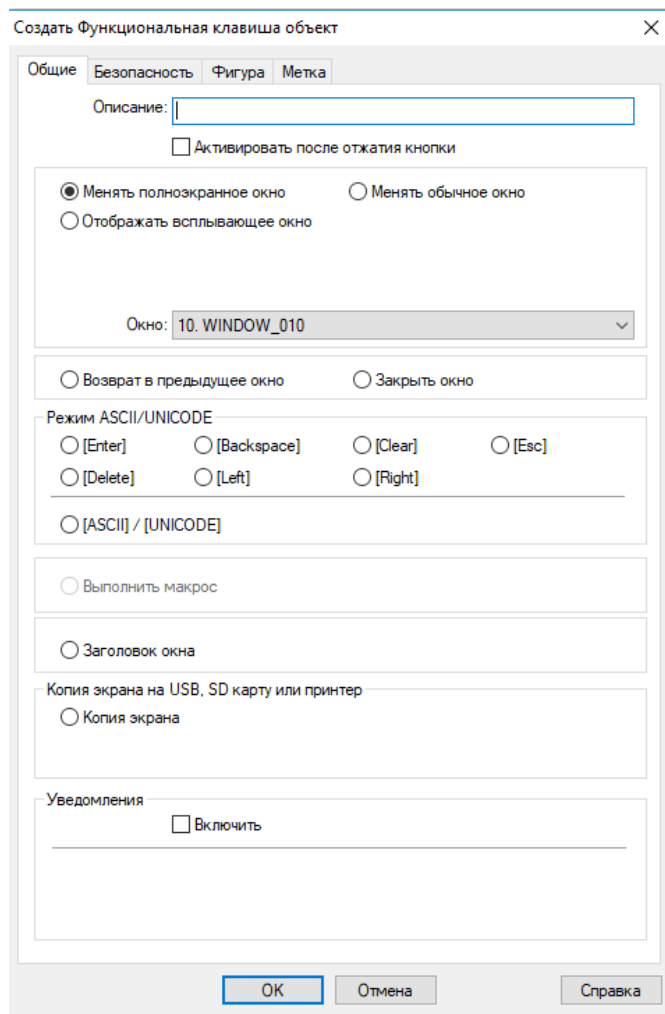
3.4. Создадим кнопки для переключения между окнами. Для этого используется элемент Функциональная кнопка.

Щелкнуть по пиктограмме Функциональная кнопка (Рис. 23).



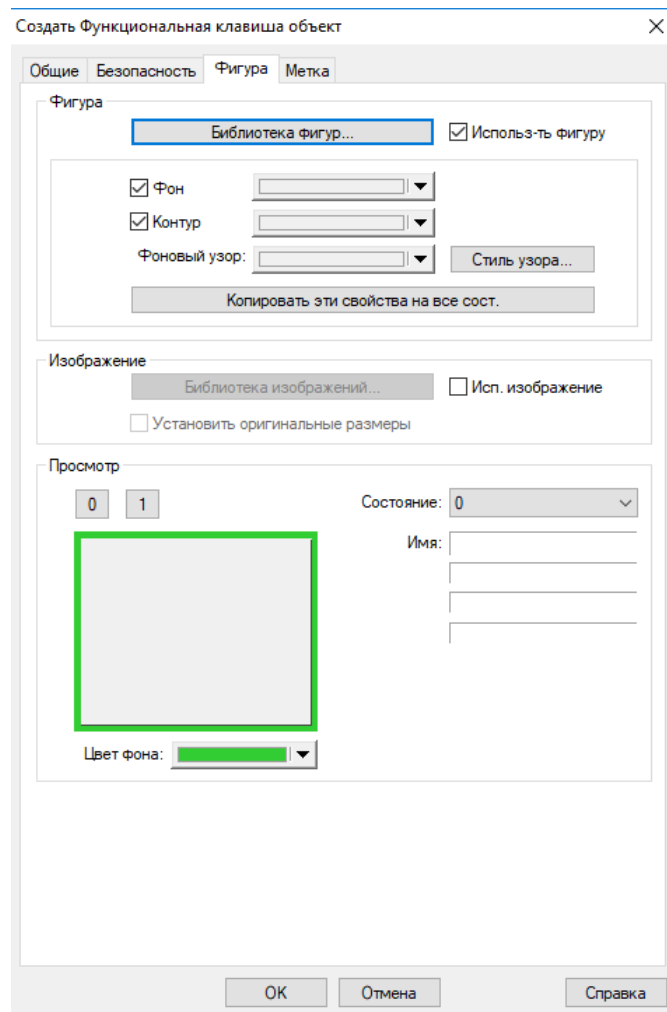
**Рис. 23. Выбор пиктограммы Функциональная кнопка в верхнем меню окна программы**

3.5. Откроется окно атрибутов функциональной клавиши (Рис. 24). Выбрать окно, в которое с помощью данной клавиши будет совершаться переход (Окно 10).



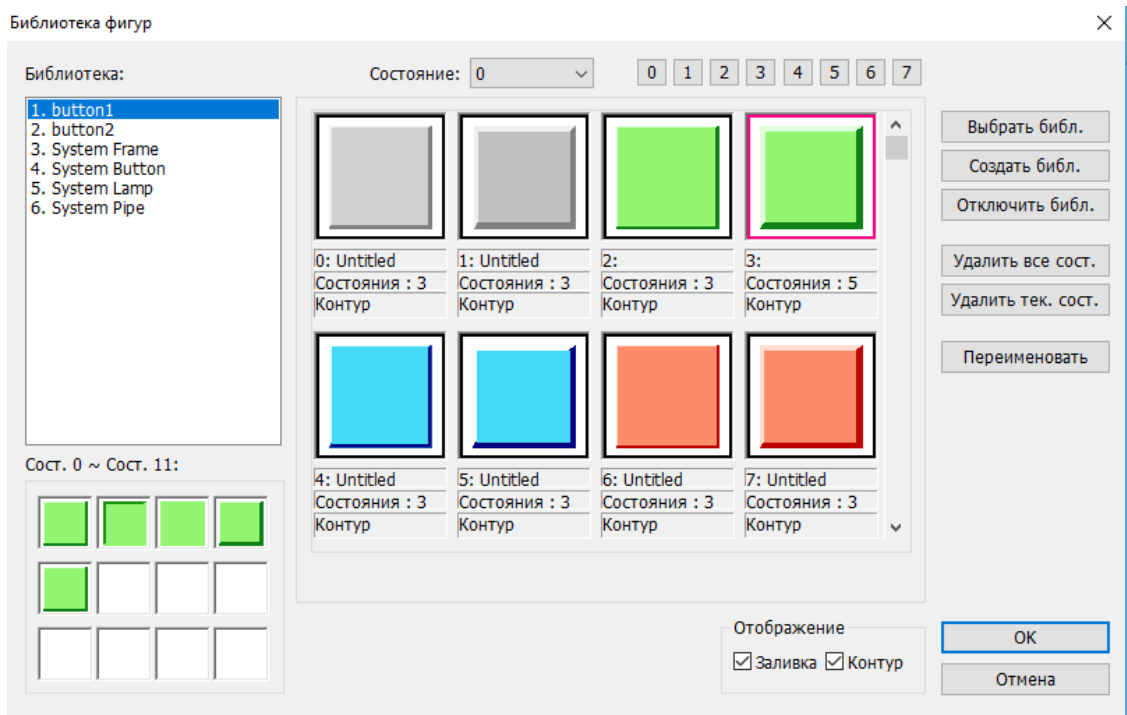
**Рис. 24. Окно атрибутов функциональной клавиши**

3.6. Перейти на вкладку Фигура (Рис. 25). Нажать кнопку Библиотека фигур.



**Рис. 25. Вкладка Фигура**

3.7. В библиотеке фигур можно выбрать, как будет выглядеть наша функциональная кнопка (Рис. 26). Вернуться на вкладку Фигура, нажав ОК. Выбрать цветовые характеристики кнопки. Нажать ОК.



**Рис. 26. Библиотека фигур**

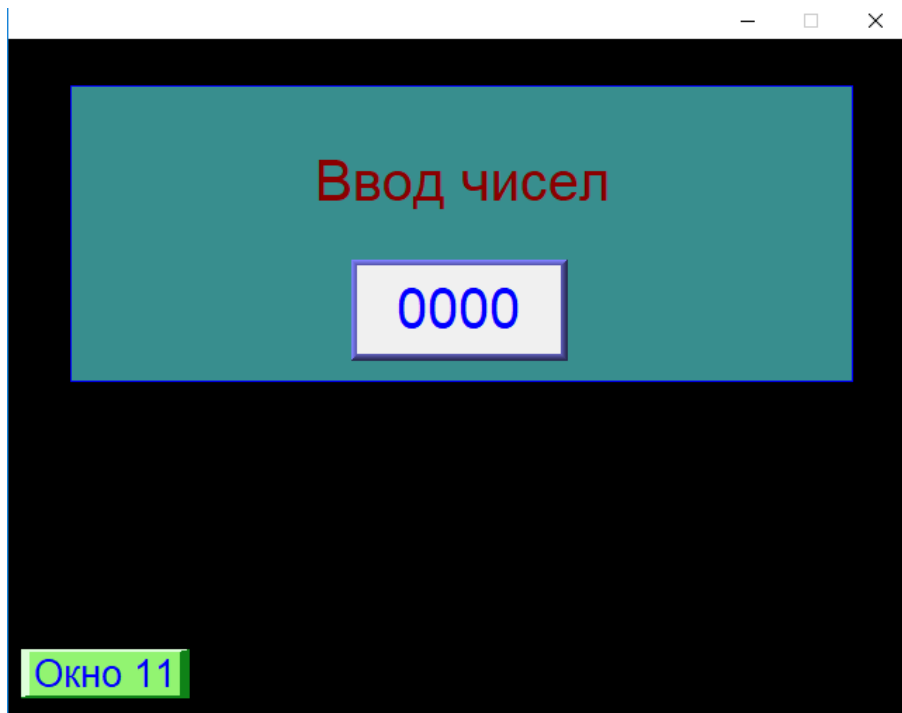
3.8. Перетащить кнопку в нужное место рабочей зоны. Создать надпись на кнопке, добавив текст и перетащив его на кнопку (Рис. 27)



**Рис. 27. Объект Кнопка в рабочей зоне**

3.9. Провести аналогичные операции для создания кнопки для перехода из Окна 10 в Окно 11.

3.10. Запустить проект и убедиться, что переключение между окнами работает (Рис. 28).



**Рис. 28. Запуск готового проекта в режиме автономного моделирования**

**Задание для самостоятельной работы:** Создать в новом окне числовое поле, дублирующее значение, введенное в окне 10.

**Примечание:** Для этого необходимо в атрибутах нового числового поля установить тот же адрес, что и у числового поля, созданного в Окне 10; и убрать галочку: Разрешить ввод.

## Глава 2. Изучение перевода данных из одного формата в другой в среде Weintek EasyBuilder 8000.

### Назначение:

- Изучение перевода данных из одного формата в другой в среде Weintek EasyBuilder 8000.

### Цель работы:

Освоение практических навыков:

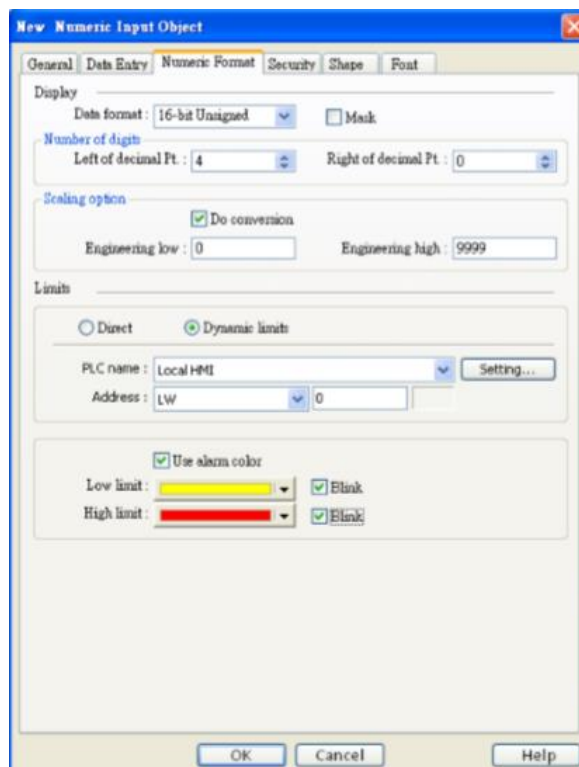
- Ввода/вывода чисел с различными типами данных
- Множественного копирования объектов в среде Weintek EasyBuilder 8000.

### Используемое инструментальное ПО:

- Среда разработки Weintek EasyBuilder 8000

### Теоретические сведения:

- Типы данных числового поля



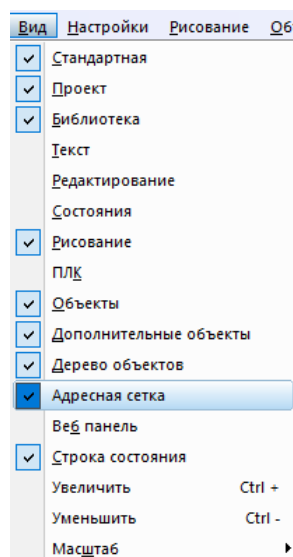
## Закладка [Numeric Format]

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

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

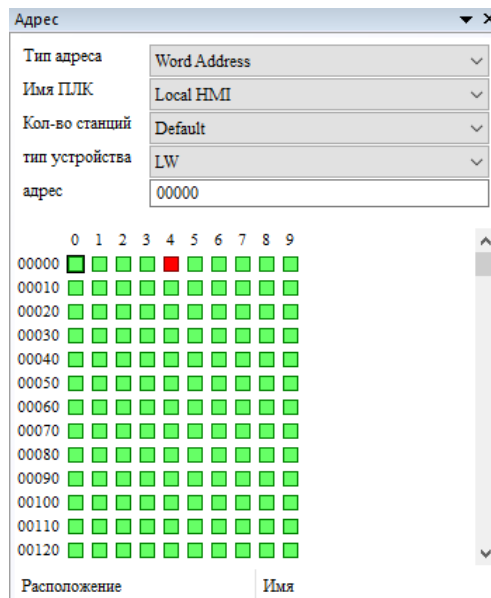
- **Занятость адресов**

Оперативная память ПЛК (обозначается LW) состоит из 11201 ячейки, каждая из которых может хранить 16 бит информации. У каждой из ячеек есть свой адрес (LW 1, LW 2, ..., LW 11201). Посмотреть занятость ячеек можно, установив галочку напротив пункта **Адресная сетка** в меню **Вид**.



В левой части окна редактирования проекта появится окно Адрес.

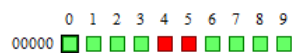




В этом окне представлен массив ячеек. Зеленым цветом обозначаются свободные ячейки. Если ячейка обозначена красным, то этот адрес используется (на рисунке выше адрес LW 4 занят).

При использовании 16-битного формата данных используется только одна ячейка памяти. Если использовать 32-битный формат, то автоматически будет занят следующий адрес. Это необходимо учитывать при использовании форматов 32-bit BCD, 32-bit Hex, 32-bit Binary, 32-bit Unsigned, 32-bit Signed, 32-bit Float.

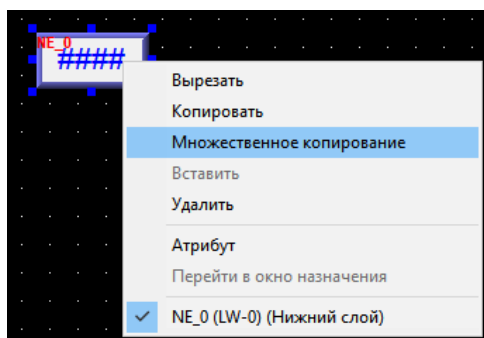
Если мы, например, установим адрес числового поля LW 4 и формат данных 32-bit Unsigned, то автоматически будет занят и адрес LW 5.



В этом случае для работы с другим объектом адрес LW 5 использовать нельзя, иначе данные для обоих объектов будут некорректны.

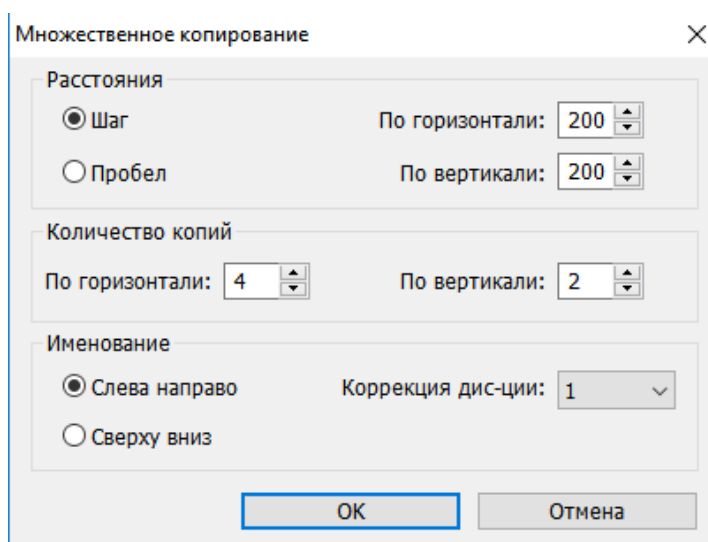
**Задание:**

1. Средствами операционной системы Windows 7 загрузить на исполнение редактор проектов EasyBuilder8000, создать новый проект и сохранить его (См. п. 1.1–1.3).
2. Создать объект числовое поле, щелкнуть по нему правой кнопкой мыши и в открывшемся списке выбрать пункт Множественное копирование (Рис. 1).



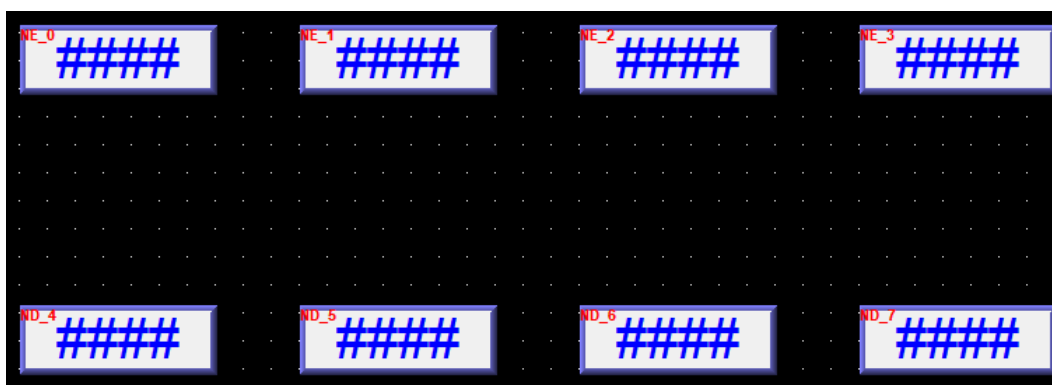
**Рис. 1. Выбор пункта Множественное копирование**

3. В открывшемся окне установить значения расстояний по горизонтали и по вертикали, количество копий по горизонтали и по вертикали (Рис. 2). Нажать ОК.



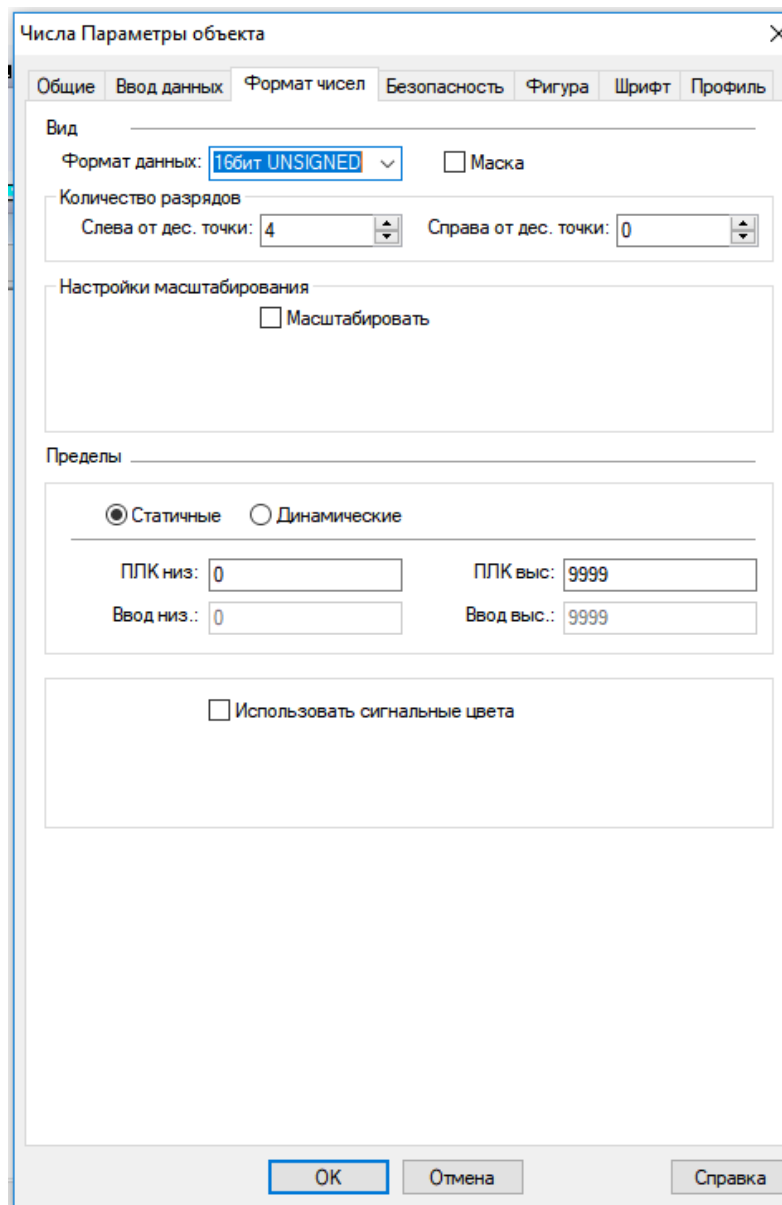
**Рис. 2. Окно настройки множественного копирования**

4. В рабочей области появится два горизонтальных ряда, каждый из которых состоит из четырех числовых полей, пронумерованных построчно числами от 0 до 7 (Рис. 3).



**Рис. 3. Результат множественного копирования объекта числовое поле**

5. Выделить нулевое поле (левое поле в верхнем ряду), щелкнув по нему левой кнопкой мыши. Щелкнуть по нему же правой кнопкой мыши и выбрать в открывшемся списке пункт Атрибут. Перейти на вкладку Формат чисел, установить Формат данных 16бит UNSIGNED, количество разрядов слева от десятичной точки - 4, справа - 0. Нажать ОК. (Рис. 4)



**Рис. 4. Окно атрибутов числового поля**

6. Выполнить действия, аналогичные п. 5, с оставшимися тремя числовыми полями из верхнего ряда, установив для поля 1 Формат данных 32бит FLOAT и количество разрядов слева от десятичной точки - 3, справа - 3.

Для поля 2 - 32бит UNSIGNED, количество разрядов слева от десятичной точки - 4, справа - 0. Для поля 3 - 32бит BIN (двоичное число) и количество разрядов слева от десятичной точки - 13.

Также необходимо учитывать занятость адресов (См. Теоретические сведения), поэтому номера адресов для полей следует выбирать чередующимися через 2 (Например, для поля 0 - LW 0, для поля 1 - LW 2, для поля 2 - LW 4, для поля 3 - LW 6).

7. Для полей из нижнего ряда должны быть установлены адреса соответствующих полей из верхнего ряда. Т.е. у четвертого поля должен быть адрес нулевого поля, у пятого - адрес первого, у шестого - адрес второго, у седьмого - адрес третьего. Кроме того, в атрибутах полей из нижнего ряда на вкладке Общие следует убрать галочку Разрешить ввод (Т. к. они будут использоваться только для отображения чисел).

Установить для поля 4 Формат данных 16бит UNSIGNED и количество разрядов слева от десятичной точки - 4, справа - 0.

Для поля 5 - 32бит FLOAT, количество разрядов слева от десятичной точки - 3, справа - 3.

Для поля 6 - 32бит BIN (двоичное число) и количество разрядов слева от десятичной точки - 14.

Для поля 7 - 32бит UNSIGNED, количество разрядов слева от десятичной точки - 4, справа - 0.

8. Запустить проект и проверить ввод/вывод данных. В верхние поля вводятся числа, а в нижних полях происходит их отображение. При этом в первом столбце вводится десятичное число и происходит его отображение, во втором - вводится и отображается десятичное число с плавающей точкой. В третьем и четвертом столбцах совершается перевод числа из десятичной системы счисления в двоичную и наоборот соответственно (Рис. 5).

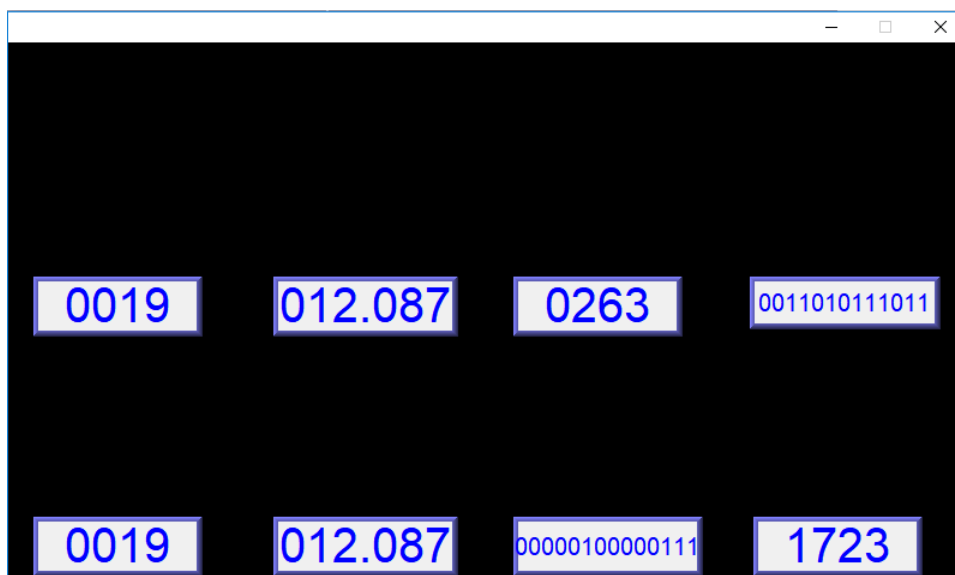


Рис. 5. Запуск проекта в режиме автономного моделирования

9. Оформить рабочую область с числовыми полями, создав рамки и надписи (Рис. 6; см. Урок1). Запустить и проверить готовый проект.



Рис. 6. Оформленный проект

**Задание для самостоятельной работы:** Провести ввод и отображение чисел с какими-либо другими форматами данных (Например, 16бит HEX - шестнадцатиричное число). Для числа с плавающей точкой изменить количество разрядов слева и справа от точки.

## Глава 3. Знакомство с макросами в среде Weintek EasyBuilder 8000.

### Назначение:

- Ознакомление с макросами в среде Weintek EasyBuilder 8000.

### Цель работы:

Освоение практических навыков:

- Работы с простейшими макросами в среде Weintek EasyBuilder 8000.

### Используемое инструментальное ПО:

- Среда разработки Weintek EasyBuilder 8000

### Теоретические сведения:

Макросы обеспечивают дополнительные функции, которые могут понадобиться для вашего приложения.

Макросы - это автоматически выполняемые последовательности команд, исполняемых во время работы. Макросы позволяют решать такие задачи, как сложные вычислительные операции, работа со строками и взаимодействия пользователей с вашими проектами.

#### ● Состав Макроса

Макрос состоит из операторов (statements) . Операторы содержат константы, переменные и операции. Они располагаются в определенном порядке для получения требуемого выходного результата.

Макрос составляется в виде:

Объявление глобальных переменных-----Опционально  
Подпрограмма. Объявления Функциональных блоков-----Опционально  
Объявление локальных переменных  
Конец подпрограммы

macro\_command                    main()-----Обязательно  
Объявления локальных переменных

## Операторы

`end macro_command()`-----Обязательно

Макрос должен иметь одну `main function`, являющуюся начальной точкой исполняемого макроса.

Её формат:

**macro\_command** Function\_Name()

**end macro\_command**

Локальные переменные используются в теле главной (`main`) функции макроса или в определенном функциональном блоке. Их значение остается действительным только внутри определенного блока.

Глобальные переменные объявляются перед всеми функциональными блоками и действительны для всех функций макроса. Когда локальные и глобальные переменные имеют одинаковое имя, учитываются только локальные переменные.

- **Константы и переменные**

### Константы

Константы — это фиксированные значения и могут быть использованы сразу в операторе:

Тип константы	Примечание	Пример
Десятичная целая		345, -234, 0, 23456
Шестнадцатеричная	Должна начинаться с 0x	0x3b, 0xffff, 0x237
ASCII (строковая)	Строка должна начинаться	'a', 'data', 'name'

	заклучена В одионочные кавычки	
Булева		true, false

### Переменные

Переменные имеют имя, как указатель на их значение. Значение может изменяться, при преобразовании переменной операторами.

### Объявление переменных

Переменные должны быть объявлены перед использованием. Все объявления переменных должны быть сделаны перед операторами макроса. Для объявления переменной нужно указать тип и имя переменной.

Пример:

```
int a
short b, switch
float pressure
```

### Инициализация переменных и массивов

Есть два способа инициализации переменных:

1. Перед оператором с помощью присваивания (=)

Пример:

```
int a
float b[3]
a = 10
b[0] = 1
```

2. При объявлении (declaration)

```
char a = '5', b = 9
```

### • **Встроенные функциональные блоки**

EasyBuilderPro имеет некоторые встроенные функции для считывания и перемещения данных в ПЛК, управления данными.

### Коммуникации

Функция SetData посылает данные в ПЛК.



SetData(send\_data[start], device\_name, device\_type, address\_offset, data\_count) или SetData(send\_data, device\_name, device\_type, address\_offset, 1)

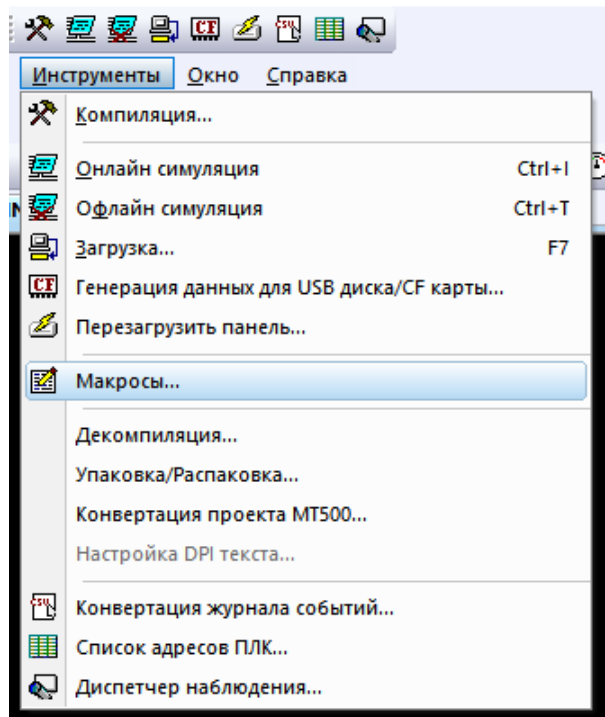
[data\_count] число отправляемых данных. В общем случае [send\_data] — массив, но если data\_count=1, то [send\_data] состоит только из одной переменной.

[device\_name] — имя ПЛК, заключенное в двойные кавычки (“”), и оно должно быть определено в списке устройств [system parameters]. [device\_type] — тип адреса и метода кодирования (binary или BCD) данных ПЛК. Например, если этот параметр задан LW\_BIN — это означает, что используется регистр типа LW и метод кодирования — binary. Если используется метод кодирования BIN, запись “\_BIN” может быть проигнорирована.

[address\_offset] — адрес смещения в ПЛК.

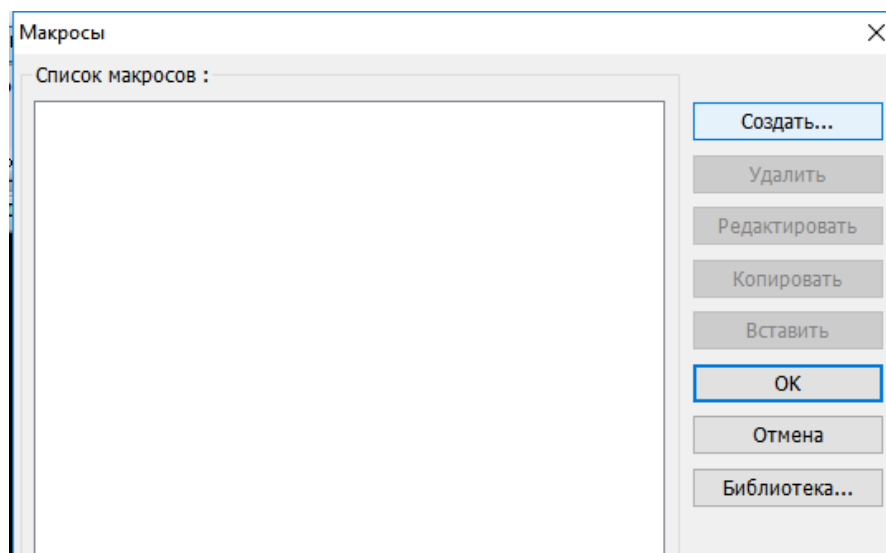
### **Задание:**

1. Средствами операционной системы Windows 7 загрузить на исполнение редактор проектов EasyBuilder8000, создать новый проект и сохранить его.
2. Создать числовое поле, убрав в его атрибутах галочку Разрешить ввод и установить формат данных 16бит UNSIGNED. Нажать ОК.
3. Далее в меню Инструменты выбрать пункт Макросы (или же щелкнуть по соответствующей пиктограмме на панели инструментов в верхней части окна; Рис. 1).



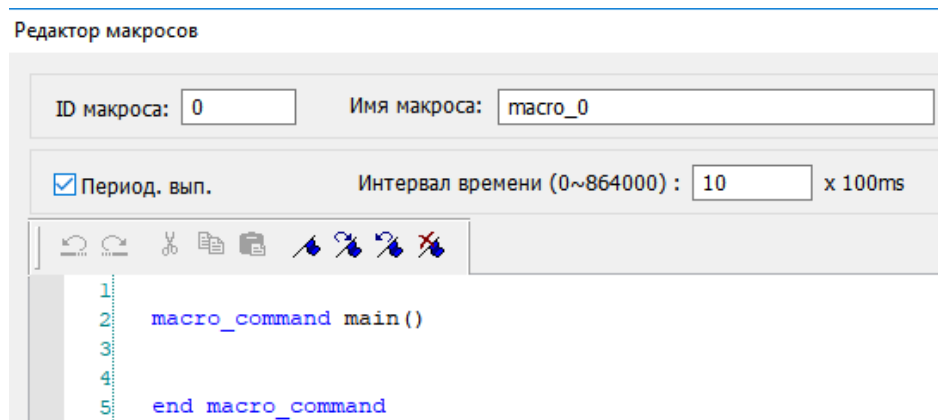
**Рис. 1. Выбор пункта Макросы меню Инструменты**

4. В открывшемся окне выбрать пункт Создать (Рис. 2).



**Рис. 2. Окно макросов**

5. Откроется окно редактирования макроса (Рис. 3). В верхней части этого окна установить галочку Период. вып. В верхней части окна можно также изменить имя макроса.



**Рис. 3. Окно редактирования макроса**

6. Ввести в рабочее поле текст макроса:

```
macro_command main()
int x=5
SetData(x, "Local HMI", LW, 0, 1)
end macro_command
```

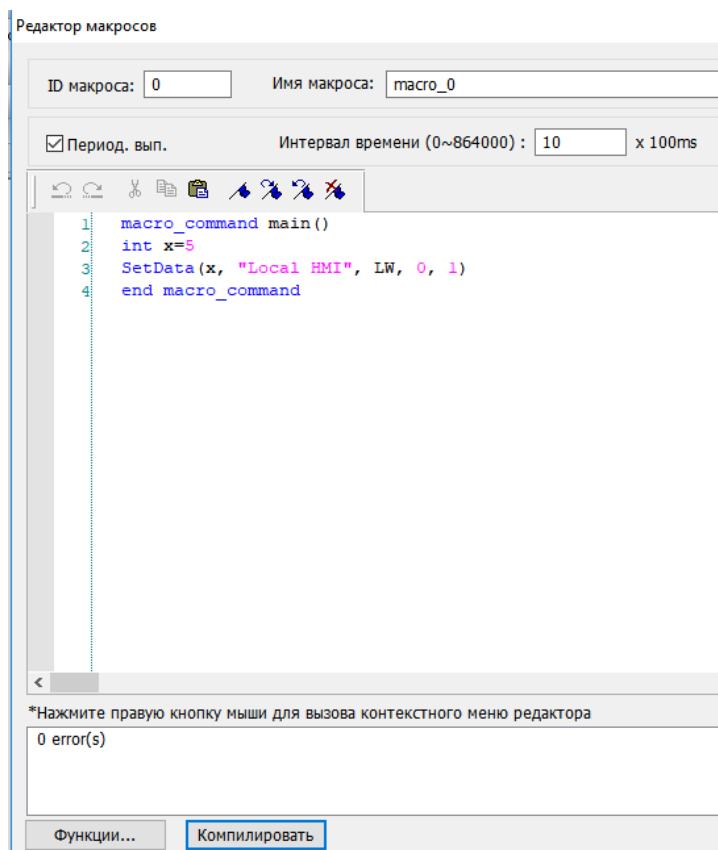
Первая строка программы объявляет ее начало.

Во второй строке программы объявляется константа типа int и устанавливается ее значение.

В третьей строке используется команда SetData, которая запишет нашу константу в числовое поле при выполнении макроса. В скобках на первом месте идет имя самой константы, затем идет имя устройства ("Local HMI" - локальная память панели). Далее указывается адрес ячейки этой памяти, состоящий из типа устройства (LW - это оперативная память) и номера ячейки памяти. На последнем месте идет количество принимаемых элементов.

Четвертая строка объявляет завершение программы.

7. В нижней части окна нажать кнопку Компилировать. Если поле над этой кнопкой после ее нажатия выдаст 0 error(s), то макрос составлен правильно. Если же в этом поле будут выведены сведения об ошибках, то необходимо найти и исправить эти ошибки, после чего компилировать еще раз (Рис. 4).



**Рис. 4. Компиляция макроса в окне редактирования макроса. Макрос составлен без ошибок**

8. После того, как редактор макросов выдаст 0 error(s), можно сохранить и протестировать получившийся макрос. Для этого нажать кнопку Выход в нижней части окна редактора. Затем в окне Макросы нажать кнопку ОК.

9. Запустить проект. Числовое поле должно выдавать константу, записанную в макросе (в нашем случае 5; Рис. 5).



**Рис. 5. Готовый проект, запущенный в автономном режиме**

**Задание для самостоятельной работы:** Оформить получившееся числовое поле и окно панели. Создать еще одно такое же числовое поле с другим адресом. Добавить в макрос еще одну константу другого типа данных (например, тип “float” - число с плавающей точкой) и строку, записывающую ее во второе числовое поле.

## Глава 4. Выполнение арифметических операций внутри макросов

### Назначение:

- Ознакомление с математическими операциями внутри макросов в среде Weintek EasyBuilder 8000 для разработки приложений графической панели WT3010.

### Цель работы:

Освоение практических навыков:

- Написания макросов, выполняющих арифметические действия над переменными
- Ввода чисел на вход макроса в среде Weintek EasyBuilder 8000.

### Используемое инструментальное ПО:

- Среда разработки Weintek EasyBuilder 8000

### Теоретические сведения:

#### • Операторы в макросах - Operators

Операторы используются для указания того, какие данные и каким образом должны быть преобразованы. В каждой строке операнду слева присваиваются условия справа.

Арифметический оператор	Описание	Пример

=	Оператор присвоения	pressure = 10
+	Addition - Сумма	A = B + C
-	Subtraction - Разность	A = B - C
*	Multiplication Умножение	A = B * C
/	Division - Деление	A = B / C

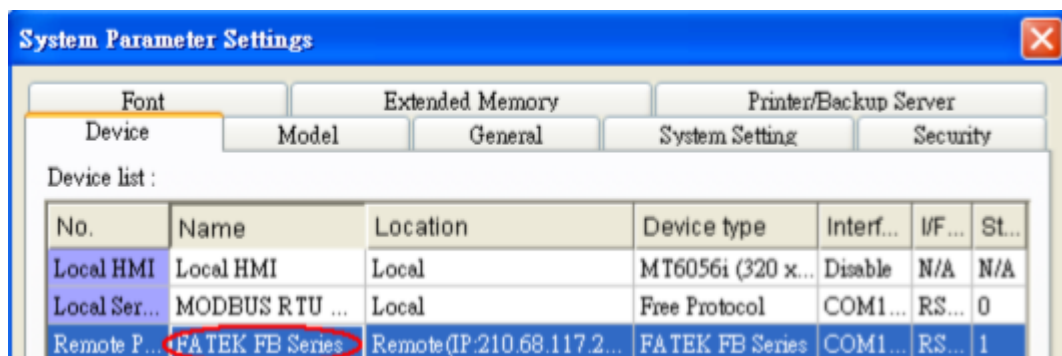
- **Встроенные функциональные блоки**

Функция GetData принимает данные из ПЛК.

GetData(read\_data[start], device\_name, device\_type, address\_offset, data\_count) или GetData(read\_data, device\_name, device\_type, address\_offset, 1)

[data\_count] — количество принимаемых элементов. В общем случае read\_data — массив, но если параметр data\_count равен 1, это будет массив из одной переменной.

[device\_name] — имя ПЛК, заключенное в двойные кавычки (“”), это имя должно быть определено в списке устройств окна [system parameters] как показано ниже (см. FATEK KB Series):



[device\_type] — тип адреса и метод кодирования (binary или BCD) данных ПЛК. Например, если этот параметр задан LW\_BIN — это означает, что используется регистр типа LW и метод кодирования – binary. Если используется метод кодирования BIN, запись “\_BIN” может быть проигнорирована.

[address\_offset] — адрес смещения в ПЛК.

### **Задание:**

6. Средствами операционной системы Windows 7 загрузить на исполнение редактор проектов EasyBuilder8000, создать новый проект и сохранить его.

7. Создать три числовых поля с разными адресами (возьмем LW 0, 2, 4). В атрибутах третьего поля убрать галочку Разрешить ввод. В атрибутах всех трех полей установить формат данных 16бит UNSIGNED.

8. Создать макрос (См. Урок 3)

9. В верхней части этого окна редактирования макроса установить галочку Период. вып.

10. Ввести в рабочее поле текст макроса:

```
macro_command main()
int int_x1, int_x2, int_x3
GetData(int_x1, "Local HMI", LW, 0, 1)
GetData(int_x2, "Local HMI", LW, 2, 1)
int_x3=int_x1+int_x2
SetData(int_x3, "Local HMI", LW, 4, 1)
end macro_command
```

Во второй строке программы объявляются переменные типа int.

В третьей и четвертой строках производится ввод данных с первых двух числовых полей в макрос с помощью команды GetData. В скобках на первом месте указывается имя переменной, принимающей значение, введенное в числовое поле. Далее указывается имя устройства "Local HMI", адрес соответствующего числового поля (LW 0, LW 2) и количество принимаемых элементов - 1.

Пятая строка производит суммирование значений из первых двух числовых полей.

Шестая строка выводит получившуюся сумму на третье числовое поле.

11. Проверить ошибки с помощью кнопки Компилировать и исправить их, если они есть. Выйти из окна редактирования макроса и в окне со списком макросов нажать ОК.

12. Запустить проект. Третье числовое поле должно выдавать сумму значений первых двух числовых полей (Рис. 1).



**Рис. 1. Запуск готового проекта**

**Задание для самостоятельной работы:** Оформить числовые поля и окно панели. Изменить получившийся макрос так, чтобы третье поле выдавало произведение значений двух первых полей.



## **Глава 5. Выполнение арифметических операций в макросе над числами, вводимыми с ползунков**

### **Назначение:**

- Изучение принципов взаимодействия объектов с макросами в среде Weintek EasyBuilder 8000.

### **Цель работы:**

Освоение практических навыков:

- Работы с объектом Ползунок
- Написания макроса для выполнения арифметических действий над числами, вводимыми с помощью Ползунка в среде Weintek EasyBuilder 8000.

### **Используемое инструментальное ПО:**

- Среда разработки Weintek EasyBuilder 8000

### **Теоретические сведения:**

#### **Slider - Ползунок**

- **Краткое описание**

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

- **Настройка**

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

- **Attribute**

[Direction:]

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

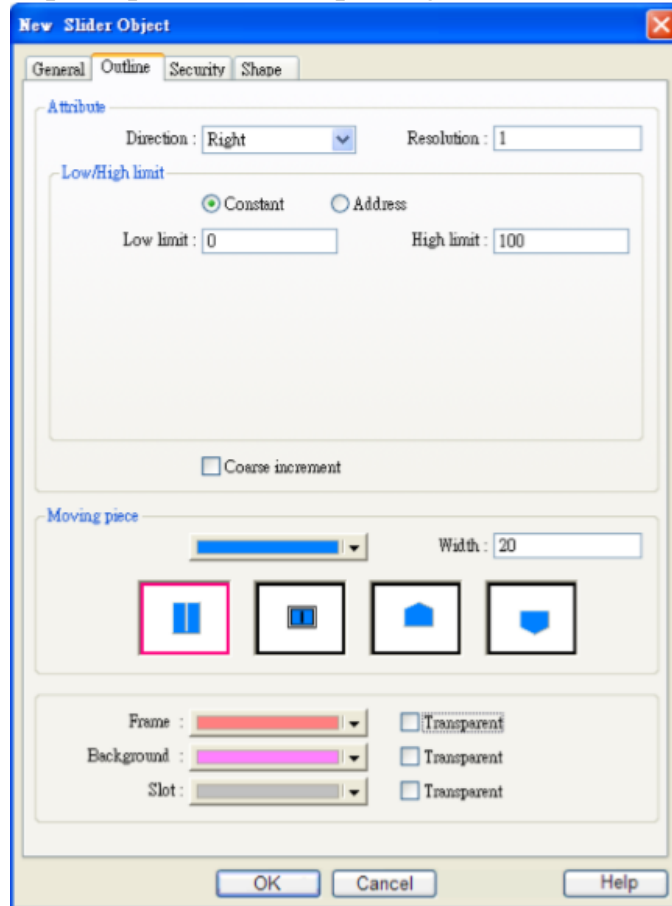
- **[Resolution:]**

Ползунок перемещается на каждые [N] шагов, где [N] — разрешение. Например, При [N]=10, будет отображаться каждый 10-й шаг

перемещения При  $[N]=5$ , будет отображаться каждый 5-й шаг перемещения При  $[N]=1$ , будет отображаться каждый шаг перемещения.

- **Constant Low limit & High limit**

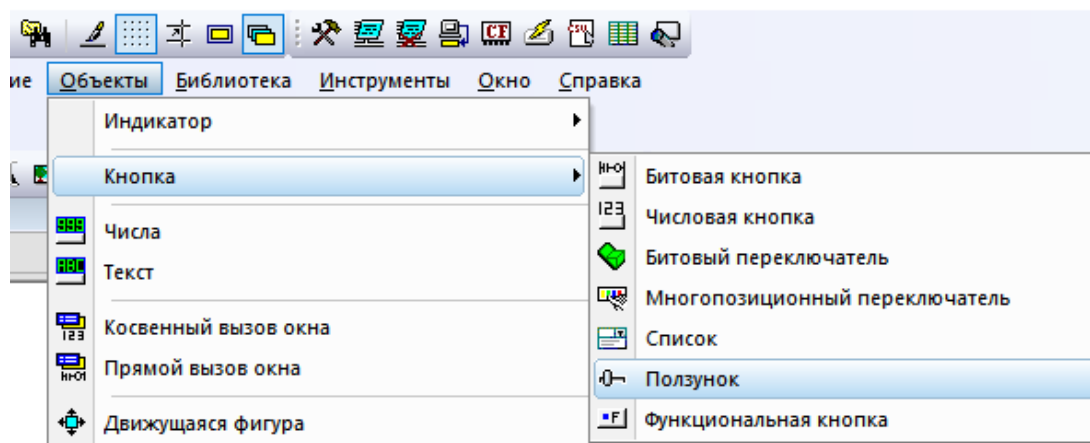
Нижний и верхний пределы значения по указанному адресу заданы как константы, например: [Input low] и [Input high].



**Задание:**

13. Средствами операционной системы Windows 7 загрузить на исполнение редактор проектов EasyBuilder8000, создать новый проект и сохранить его.

14. В меню Объекты в пункте Кнопка выбрать пункт Ползунок (или же щелкнуть по соответствующей пиктограмме на панели инструментов в верхней части окна; Рис. 1).



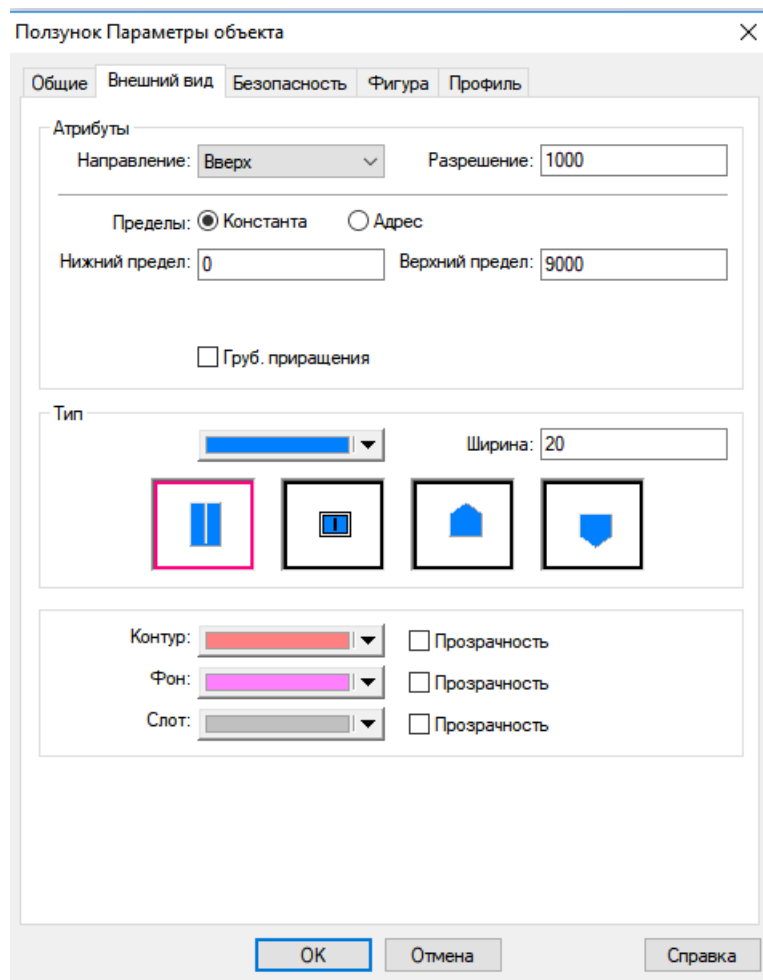
**Рис. 1. Выбор объекта Ползунок в меню Объекты**

15. В открывшемся окне на вкладке Внешний вид в пункте Направление выбрать Вверх. На этой же вкладке для первого ползунка установить Разрешение - 1000, Нижний предел - 0, Верхний предел - 9000 (Рис. 2).

Для второго ползунка: Разрешение - 100, Нижний предел - 0,  
Верхний предел - 900;

Для третьего ползунка: Разрешение - 10, Нижний предел - 0,  
Верхний предел - 90;

Для четвертого ползунка: Разрешение - 1, Нижний предел - 0,  
Верхний предел - 9.



**Рис. 2. Окно параметров объекта Ползунок**

16. На вкладке Фигура можно также выбрать как будет выглядеть наш ползунок. Нажать ОК.

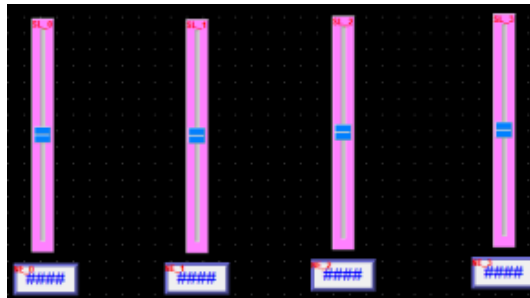
17. Созданный ползунок необходимо растянуть, чтобы в дальнейшем с ним было удобно работать (Рис. 3).



**Рис. 3. Объект ползунок**

18. Создать еще три копии этого ползунка. В атрибутах каждого из получившихся четырех ползунков должны быть установлены разные адреса (возьмем LW 0, 2, 4, 6).

19. Для каждого ползунка создать числовое поле с форматом данных 16бит UNSIGNED и соответствующим адресом (Рис. 4).



**Рис. 4. Ползунки и соответствующие им числовые поля в рабочей зоне**

20. Создать еще одно числовое поле с адресом, отличным от адреса каждого из ползунков (возьмем LW 8).

21. Создать макрос и ввести в него программный код:

```
macro_command main()
int int_x1, int_x2, int_x3, int_x4, int_x5
GetData(int_x1, "Local HMI", LW, 0, 1)
GetData(int_x2, "Local HMI", LW, 2, 1)
GetData(int_x3, "Local HMI", LW, 4, 1)
GetData(int_x4, "Local HMI", LW, 6, 1)
int_x5=int_x1+int_x2+int_x3+int_x4
SetData(int_x5, "Local HMI", LW, 8, 1)
end macro_command
```

Во второй строке макроса объявляется 5 переменных типа int.

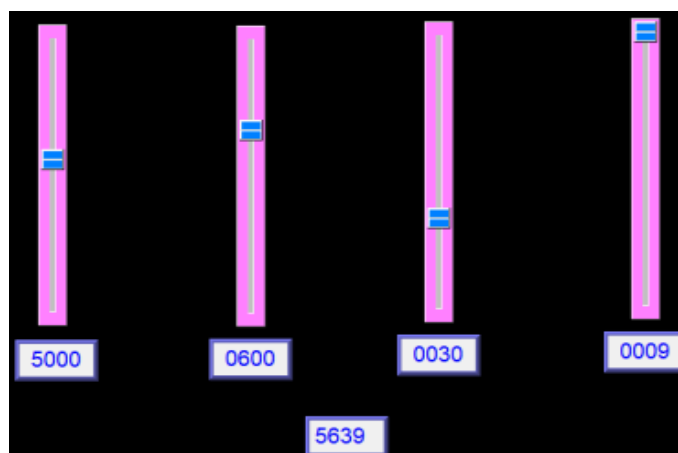
В строках с третьей по шестую происходит чтение значений с ползунков и запись их в переменные int\_x1, int\_x2, int\_x3, int\_x4.

В седьмой строке производится суммирование этих значений и запись их в переменную int\_x5.

В восьмой строке значение переменной int\_x5 выводится на числовое поле с адресом LW 8.

22. Проверить макрос на наличие ошибок и сохранить его.

23. Запустить проект (Рис. 5).



**Рис. 5. Готовый проект, запущенный в режиме автономного моделирования**

При передвижении ползунков на соответствующие имя поля должны выводиться числа разной разрядности (для первого ползунка - тысячи, для второго - сотни, для третьего - десятки, для четвертого - единицы). Числовое поле снизу будет выдавать их сумму. Таким образом, с помощью этих ползунков на нижнем числовом поле можно установить любое целое число от 0 до 9999.

**Задание для самостоятельной работы:** Оформить проект. Поменять разрешение, верхние и нижние пределы ползунков.

## Список литературы

1. Руководство пользователя EB8000
2. Руководство пользователя EasyBuilder Pro
3. Учебно-методический комплекс по применению автоматизированных систем для исследования явлений переноса в пористых средах. - Методическое пособие. Лабораторная работа №4. - «Основные принципы разработки экранных форм графической панели WT3010. Организация взаимодействия панели WT3010 и контроллера СРМ902 по протоколу Modbus» / Казанский (Приволжский) Федеральный университет. Физический факультет. Кафедра Радиоэлектроники. Казань, 2012.
4. Александр Деркач, Раис Камалиев, Вячеслав Маценко. Примеры решений на базе сенсорных панелей Weintek для АСУ ТП и встраиваемых систем. / Разработки. - Контрольно-измерительные системы. Казань, СТА3/2014. // СТА-пресс: [www.cta.ru](http://www.cta.ru)
5. Виктор Денисенко. Протоколы и сети Modbus и Modbus TCP / В записную книжку инженера / СТА4/2010. // СТА-пресс: [www.cta.ru](http://www.cta.ru)
6. Сенсорные операторские панели Weintek // [www.weintek.ru](http://www.weintek.ru)
7. ADAM-3600 - Многофункциональный промышленный контроллер // [m.habr.com/ru/company/advantech/blog/446210/](http://m.habr.com/ru/company/advantech/blog/446210/)
8. Просто о протоколе Modbus RTU // [m.ipc2u.ru/articles/prostye-resheniya/modbus-rtu/](http://m.ipc2u.ru/articles/prostye-resheniya/modbus-rtu/)