

Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Пермский национальный исследовательский политехнический университет»
(ПНИПУ)
Электротехнический факультет
Кафедра «Информационные технологии и автоматизированные системы» (ИТАС)

Методические указания
к лабораторной работе № 6
на тему
«Разработка интерфейса оператора автоматизированной системы
управления в SCADA Trace Mode»
по дисциплине
«Системы реального времени»

г. Пермь, 2018

Разработчик: ст. преп. каф. ИТАС Фёдоров А.Б.

Консультанты: канд. техн. наук, доцент каф. МСА Кычкин А.В.

асс. каф. ИТАС Мехоношин А.С.

Рецензент: д-р экон. наук, профессор каф. ИТАС Файзрахманов Р.А.

Содержание

Лабораторная работа 6. Разработка интерфейса оператора автоматизированной системы управления в SCADA Trace Mode.....	4
Цель и задачи работы	4
Теория	4
Состав SCADA Trace Mode	4
Математическая обработка данных	5
Особенность среды разработки	5
Компоненты данного SCADA-приложения	6
Алгоритм создания проекта SCADA Trace Mode.....	6
Ход работы	8
Задание 1 (Изменение цвета «лампочки» по нажатию кнопки)	8
Задание 2 (Счетчик)	15
Задание 3 (Движение объекта).....	18
Задание 4 (по вариантам)	23
Требования к содержанию отчета	24
Контрольные вопросы	25

Лабораторная работа 6. Разработка интерфейса оператора автоматизированной системы управления в SCADA Trace Mode

Цель и задачи работы

Цель: Освоить методику разработки интерфейса оператора автоматизированной системы управления в SCADA Trace Mode.

Задачи работы:

- 1) Изучить теоретические основы методики разработки интерфейса оператора автоматизированной системы управления в SCADA Trace Mode.
- 2) Выполнить учебные задания.
- 3) Выполнить задание своего варианта.
- 4) Ответить на вопросы.

Теория

SCADA Trace Mode – это распространенная в России SCADA-система, предназначенная для разработки АСУТП широкого назначения. SCADA Trace Mode создана в 1992 году фирмой AdAstrA Research Group, Ltd (Россия) и к настоящему времени имеет свыше 4000 инсталляций. Системы, разработанные на базе SCADA Trace Mode, работают в энергетике, металлургии, нефтяной, газовой, химической и других отраслях промышленности и в коммунальном хозяйстве России. SCADA Trace Mode достаточно универсальная SCADA-система, она может применяться как при автоматизации зданий, так и промышленных объектов.

Состав SCADA Trace Mode

SCADA Trace Mode – это программный комплекс, предназначенный для разработки и запуска в реальном времени распределенных автоматизированных систем управления технологическими процессами (АСУТП) и решения ряда задач управления предприятием (АСУП). Для решения задач АСУП в TRACE MODE 6 интегрирован пакет T-FACTORY. Комплекс программ TRACE MODE 6 можно разделить на 3 части.

Интегрированная среда разработки проекта (ИС) – единая программная оболочка, содержащая все необходимые средства для разработки проекта. Под проектом в TRACE MODE понимается вся совокупность данных и алгоритмов функционирования распределенной АСУ (АСУТП и/или T-FACTORY), заданных средствами TRACE MODE. Итогом разработки проекта в ИС является создание файлов, содержащих необходимую информацию об алгоритмах работы АСУ. Эти файлы затем размещаются на аппаратных средствах (компьютерах и контроллерах) и выполняются под управлением исполнительных модулей TRACE MODE.

Исполнительные модули (мониторы, МРВ) – программные модули различного назначения, под управлением которых в реальном времени выполняются составные части проекта, размещаемые на отдельных компьютерах или в контроллерах.

Составная часть проекта, размещаемая на отдельном компьютере или в контроллере и выполняемая под управлением одного или нескольких исполнительных модулей TRACE MODE, называется узлом проекта.

В общем случае размещение узла на том же аппаратном средстве, на котором он должен исполняться под управлением монитора, не является обязательным – мониторы могут загружать узлы с удаленных аппаратных средств.

Драйверы обмена – драйверы, используемые мониторами TRACE MODE для взаимодействия с устройствами, протоколы обмена с которыми не встроены в мониторы.

Математическая обработка данных

Любая АСУ требует математической обработки данных. Для этого в TRACE MODE 6 предусмотрены следующие средства:

- внутренние алгоритмы числовых каналов;
- программы.

Для разработки программ в ИС встроены языки Техно ST, Техно SFC, Техно FBD, Техно LD и Техно IL, являющиеся модификациями языков ST (Structured Text), SFC (Sequential Function Chart), FBD (Function Block Diagram), LD (Ladder Diagram) и IL (Instruction List) стандарта IEC61131-3. Программы, разрабатываемые в ИС, позволяют использовать функции из внешних библиотек (DLL).

Эти средства обеспечивают возможность математической обработки данных в любом звене информационного потока.

Особенность среды разработки

Состав программных модулей делится на 2 группы ПО:

- 1) для разработки (TRACE MODE IDE)
- 2) для исполнения (MPB TRACE MODE)

Среда разработки является средством управления проектом и средством создания интерфейса оператора. Все функциональные элементы включены в состав среды.

Для создания интерфейса оператора необходимо использовать соединение с внешними устройствами.

Внешними устройствами являются теги или каналы, описывающие взаимодействие с модулями протокола, то есть связаны с компонентом источник/приемник.

Виртуальный канал используется только внутри канала и не используется источником /приемником.

В качестве внешних источников/приемников могут выступать генераторы и модели.

В TRACE MODE можно создать 3 вида проектов:

- 1) простой;
- 2) стандартный;
- 3) компонентный.

Простой вид проекта отображает следующие слои:

- ресурсы;
- система;
- источники/приемники;
- библиотека компонентов.

Стандартный вид проекта отображает все слои простого уровня плюс все слои шаблонов (экраны, программы, связи с БД, документы).

Комплексный вид содержит все слои с учетом слоя База каналов.

TRACE MODE обеспечивает реализацию следующих технологий: soft logic, scada, HMI, EAM, HMR, MES.

Каждая из подсистем обеспечивает процесс автоматизации того или иного уровня от взаимодействия с оборудованием до управления архивной информацией.

Компоненты данного SCADA-приложения

Основным элементом является канал.

Канал – это переменная, обеспечивающая реализацию исполнительных элементов и прием информации с датчиков.

Каналы взаимодействуют с программами, событиями и внешними подсистемами. Включают стандартные каналы плюс каналы вызова, плюс каналы событий.

Шаблоны обеспечивают единое для интерфейса оператора представление страниц пользователей или экранов, программ и документов. (Документы нужны для построения отчетов.) Существуют шаблоны для связи с БД.

Ресурсы – это источники/приемники всей информации, связанной с внешним оборудованием.

Библиотека компонентов содержит модули ActiveX для разработки графики.

Алгоритм создания проекта SCADA Trace Mode

- 1) Запустить TRACE MODE IDE с помощью tmdevenv.exe.
- 2) Создать проект.

Необходимо выбрать сложность проекта. По умолчанию проект стандартный. При создании в окне разработки будет построен шаблон проекта (рисунок 1).

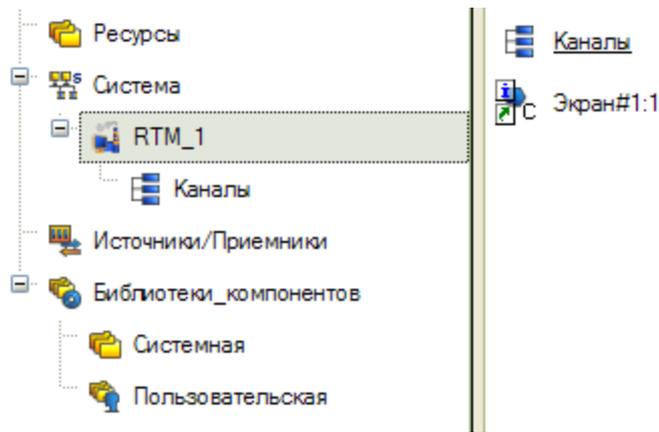


Рисунок 1 – Шаблон простого проекта

- 3) В разделе Система создаем проект запуска или систему управления (модуль RTM). По умолчанию будет создана группа Каналы, которая будет хранить все переменные системы, которые выполняют управление.
Для работы с внешними устройствами каждый канал необходимо связать с источником /приемником.
- 4) В разделе Источники/Приемники создать несколько элементов управления. Таким образом обеспечить связь с реальным протоколом, либо генератором, либо моделью.
- 5) В группе Каналы будем хранить все переменные системы, обеспечивающие процесс функционирование .
- 6) В разделе Шаблоны экранов необходимо создать экран интерфейса оператора. Экран будет общий для всей системы. (В простом проекте модуль RTM создается автоматически вместе с группой каналов и экраном)
- 7) В разделе Каналы пункта RTM необходимо создать канал ввода/вывода в формате предлагаемой группы.
Количество каналов должно соответствовать числу элементов экрана (по возможности). Каждый канал является управляющим звеном для аргументов ГЭ и может быть связан с реальным источником/приемником. Привязка осуществляется через окно свойств. Управление свойствами канала осуществляется через пункт редактирования.
- 8) Размещение ГЭ.
Для отображения действий датчиков и исполнительных элементов создается изображение на экране. Каждое отображение это есть реализация на изменение канала. При выборе свойств объекта появляется окно свойств реакций на действия. Для управления доступны следующие свойства:
 - основные (изменение контура, заливка, текст, цвет текста, подсказка);
 - динамизация (динамическая заливка слоя, контура и т.д.);
 - событие (отправить аргумент, отправить значение в аргумент, показать/скрыть элемент, переместиться на экран, отправить комментарии, послать подсказку, отправить строку и выполнить);
 - динамическая трансформация (перемещение, масштаб, вращение).

Для каждого элемента действия указывается аргумент управления, значение из которого принимается ГЭ, либо значение, в которое сохраняется информация. Каждый элемент управления должен характеризоваться аргументом.

9) Обеспечение привязки аргументов с каналами.

Привязка осуществляется в окне свойств экрана на вкладке Аргументы. Для всех аргументов определены типы: input, output. Внимание: Запись осуществляется во входное значение канала, а чтение из реального значения канала.

10) Сохранение проекта. Построение проекта для МРВ. Запуск в МРВ.

11) Все элементы снабжаются текстовыми комментариями, являющимися статическим текстом.

12) Управление через T-Factory.

Примечания:

- Каждый созданный аргумент должен быть привязан к реальным каналам.
- Тип привязки зависит от вида источника/приемника.
- При использовании программ, аргументы программ привязываются подобным образом как и экраны. При этом аргумент должен быть объявлен в программе как переменная.

Ход работы

Задание 1 (Изменение цвета «лампочки» по нажатию кнопки)

По нажатию на кнопку «лампочка» включается/выключается (меняет цвет на зеленый/красный).

1. Запустите TRACE MODE IDE.

2. Создайте новый проект: *Файл* → *Новый...* Тип проекта выберите *Стандартный*. (Результат выполненных действий смотри на рис.2)

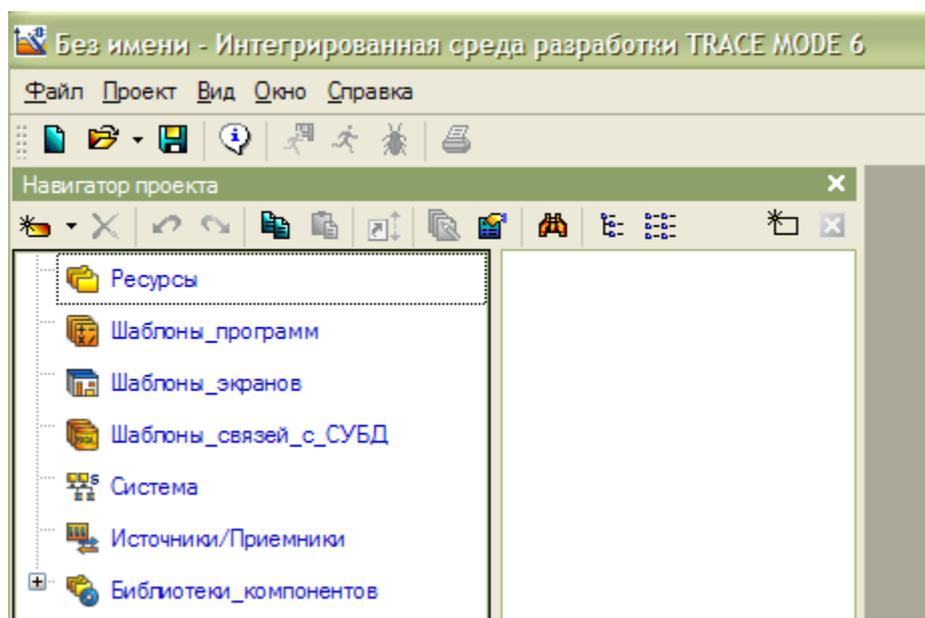


Рисунок 2 – Вид созданного проекта

3. Сделайте необходимые настройки созданного проекта. Например, *Файл* → *Настройки ИС...* → *Редактор аргументов* → *Функции* → *Переименовывать аргументы при привязке* (убрать галочку).

4. В разделе Система создайте узел RTM (см. рис. 3).

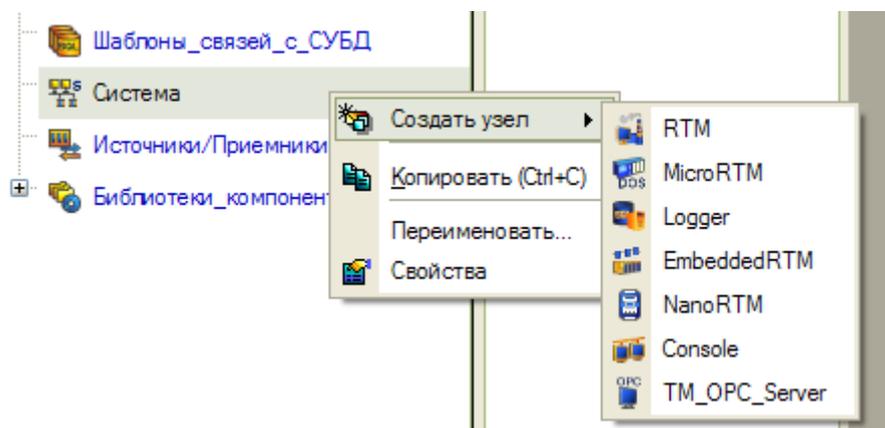


Рисунок 3 – Создание узла RTM

5. В созданном узле RTM в группе Каналы создайте компонент Экран (см. рис. 4).

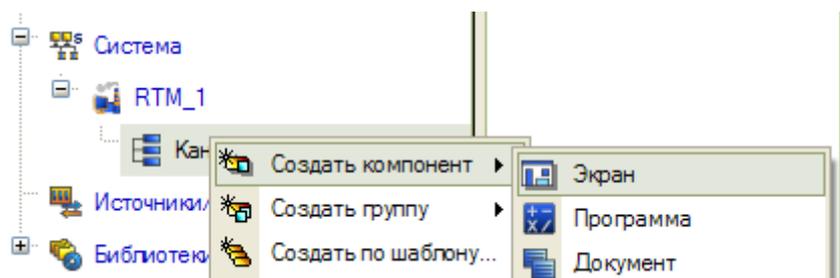


Рисунок 4 – Создание компонента Экран

При этом в разделе Шаблоны_экранов появится созданный экран (см. рис. 5).

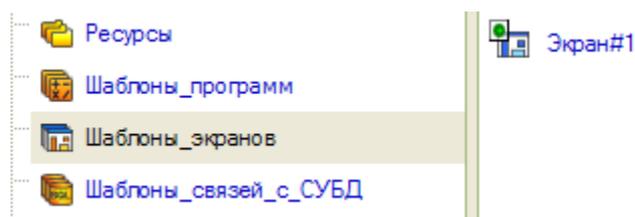


Рисунок 5 – Создание шаблона экрана

6. Перейдите к редактированию экрана. (2ЛК → *Экран#1* (или *Экран#1:1*)) Создайте графические элементы как на рис. 6.

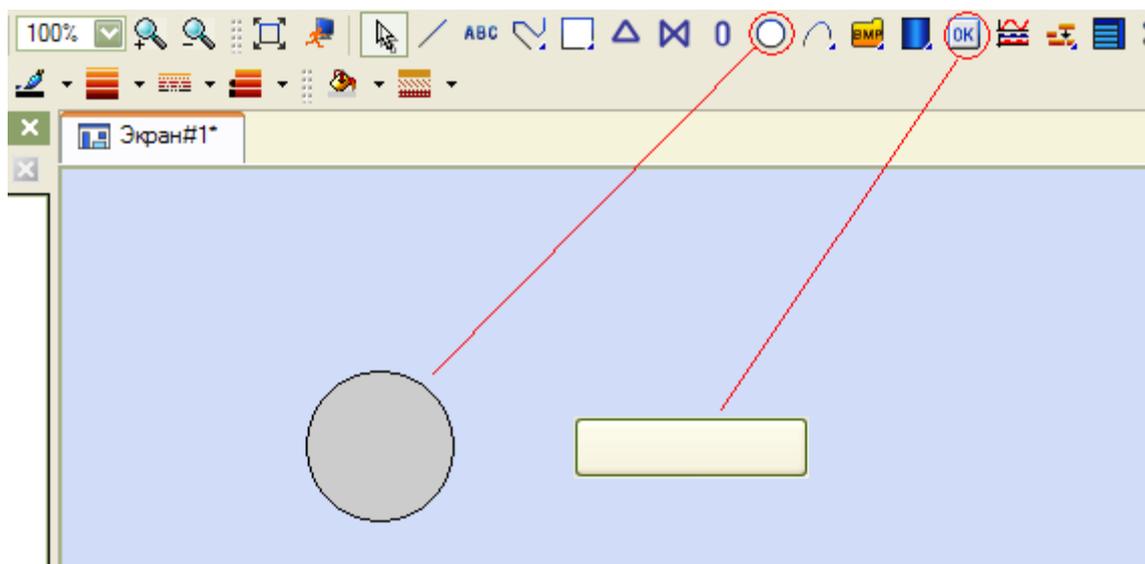


Рисунок 6 – Создание графических элементов

7. В шаблоне экрана создайте аргумент типа `int: Color (IN/OUT)`.
 Экран → ПК → Свойства → Аргументы (см. рис. 7).

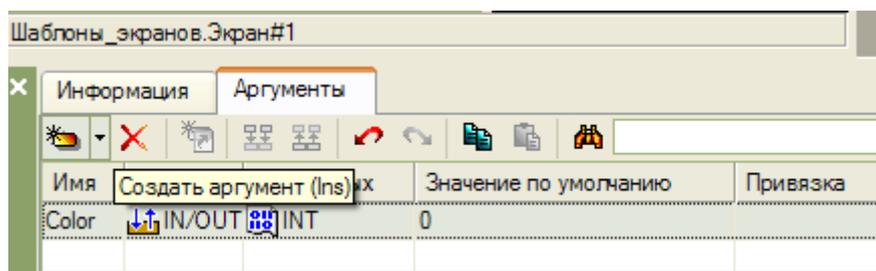


Рисунок 7 – Создание аргументов экрана

8. В свойствах эллипса установите динамизацию заливки. Для этого вид индикации установите `Arg = const` и привяжите к аргументу экрана `Color`. В поле Константа установите 1. (см. рис. 8)

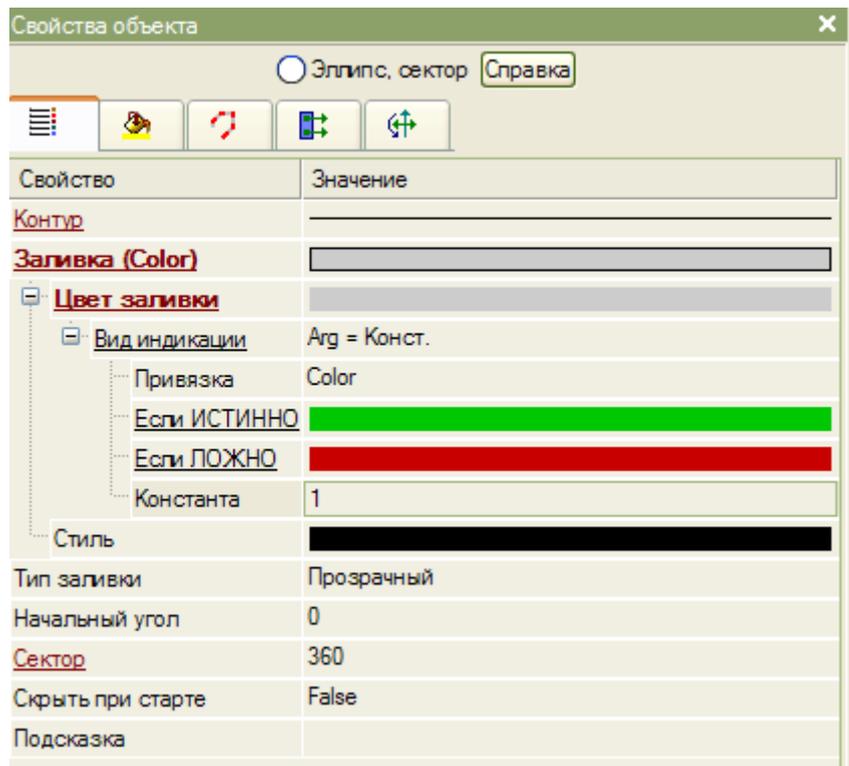


Рисунок 8 – Установка свойств объекта Эллипс

9. Аналогично пункту 5 (создание компонента Экран), создайте компонент Программа. При этом в разделе Шаблоны_программ появится созданная программа. Перейдите к созданию аргументов программы. *Программа* → *ЛК* → *Аргументы* → *ЛК*. Создайте аргументы как на рис 9.

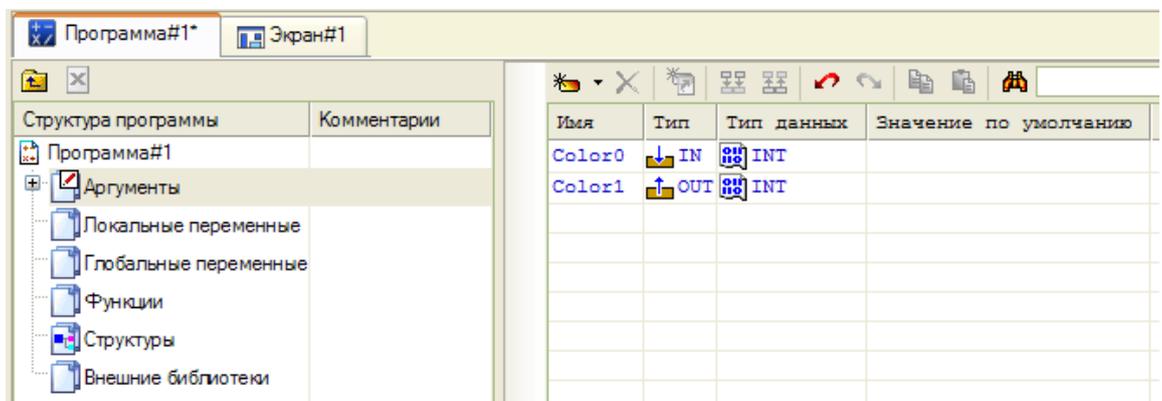


Рисунок 9 – Создание аргументов программы

10. Перейдите к написанию кода программы (*Программа#1* → *ЛК*). При этом появится окно выбора языка программирования (см. рис. 10.1). Выберите язык ST.

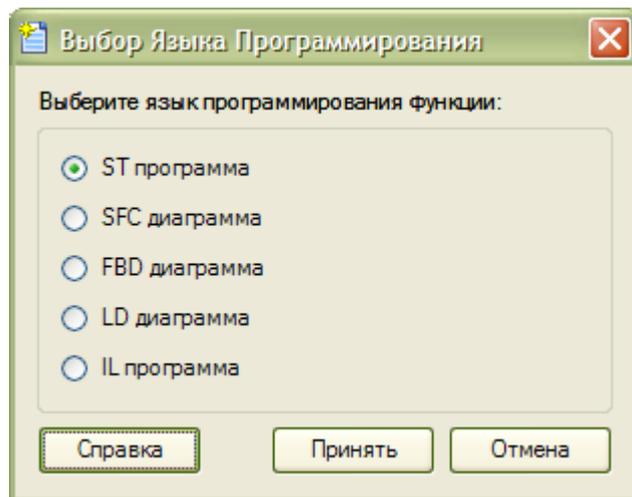


Рисунок 10.1 – Выбор языка программирования

Результат выполненных действий представлен на рис. 10.2.

```
PROGRAM
  VAR_INPUT Color0 : INT; END_VAR
  VAR_OUTPUT Color1 : INT; END_VAR

END_PROGRAM
```

Рисунок 10.2 – Первоначальный код программы

Добавьте код в программу (см. рис 10.3).

```
PROGRAM
  VAR_INPUT Color0 : INT; END_VAR
  VAR_OUTPUT Color1 : INT; END_VAR

  if (Color0 == 0) then  Color1 = 1;
  else Color1 = 0;
  end_if;

END_PROGRAM
```

Рисунок 10.3 – Окончательный вид программы

11. Откомпилируйте программу: *Программа* → *Компиляция* (или F7).
12. Перейдите на экран. В свойствах кнопки установите текст как «Включить/Выключить», затем перейдите на закладку События. На нажатие кнопки добавьте событие Execute – Выполнить. В поле Выполнить укажите созданную вами программу. (см. рис. 12)

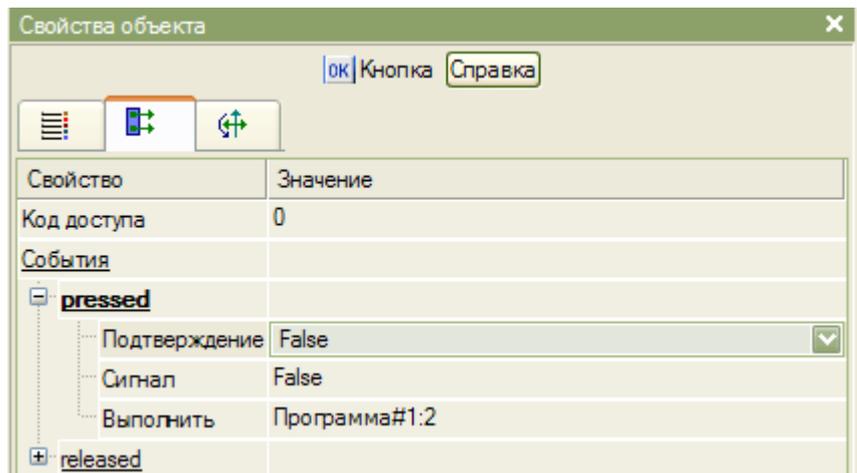


Рисунок 12 – Создание события Выполнить

13. Перейдите в свойства канала вызова экрана узла RTM. В закладке Аргументы привяжите аргумент экрана Color к аргументу программы Color1 (см. рис 13).

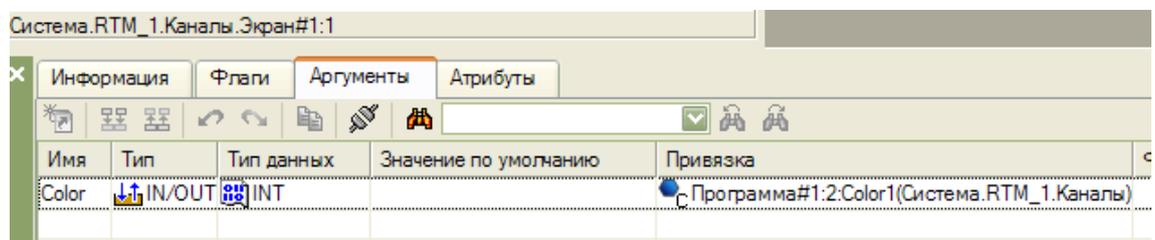


Рисунок 13 – Привязка аргумента канала экрана к аргументу канала программы

14. Перейдите в свойства канала вызова программы узла RTM. В закладке Аргументы привяжите аргумент Color0 к аргументу экрана Color (см. рис 14).

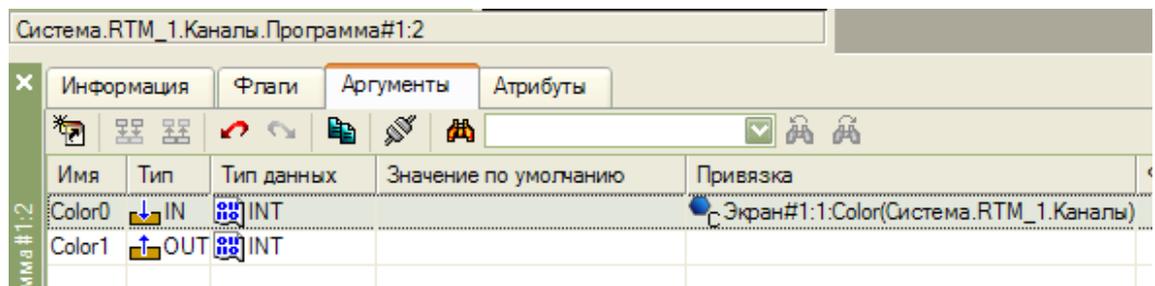


Рисунок 14 – Привязка аргумента канала программы к аргументу канала экрана

15. Перейдите в закладку Флаги, поставьте флаг Выключить (см. рис. 15).

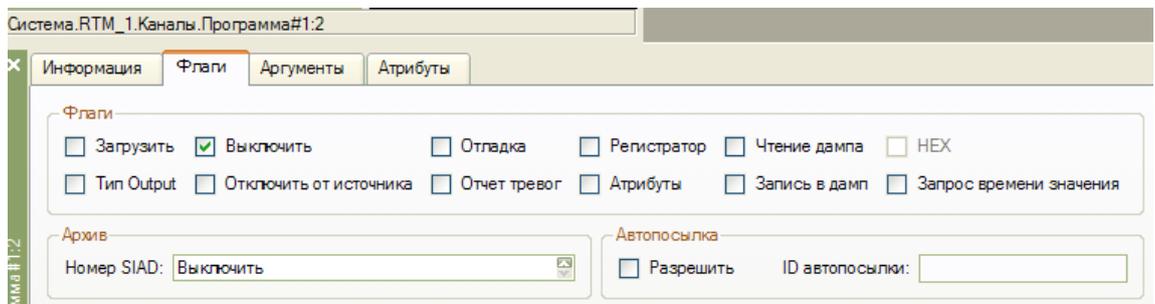


Рисунок 15 – Закладка Флаги

16. Сохраните проект (Файл → Сохранить). Сохраните проект для монитора реального времени (Файл → Сохранить для MPB). Запустите профайлер (Файл → Отладка). Для этого поставьте указатель мышки на узел RTM, при этом станет активна иконка . Результат выполненных действий смотрите на рис. 16.

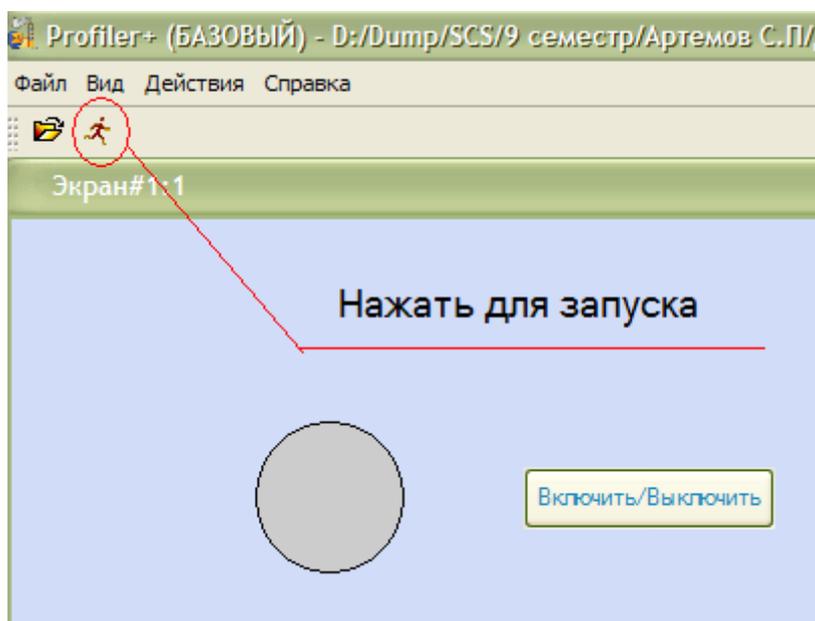


Рисунок 16 – Профайлер

17. Запустите проект. Результат работы приложения см. на рис 17.1 и рис. 17.2.

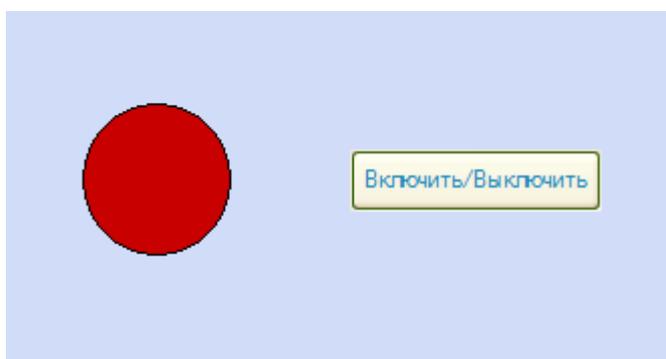


Рисунок 17.1 – Лампа «выключена»

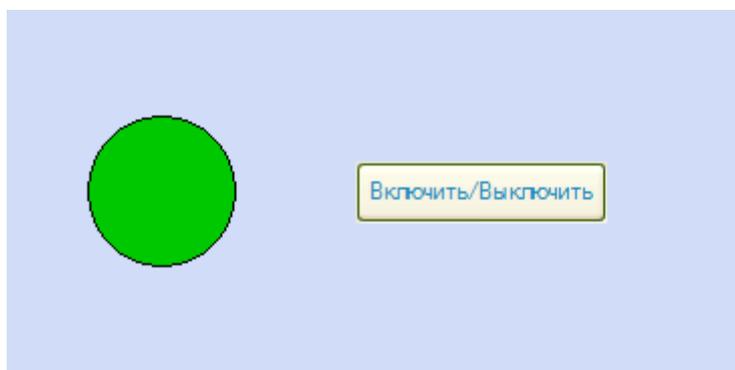


Рисунок 17.1 – Лампа «включена»

18. Остановите выполнение проекта. Закройте профайлер.

Задание 2 (Счетчик)

По нажатию на кнопку «start» счетчик изменяется. По нажатию на кнопку «stop» счетчик останавливается. Затем при нажатии на «start» счет продолжает. Счетчик начинает счет с 0 и считает до 100, затем обнуляется в 0 и снова считает.

1. Выполните пункты с 1 по 5 примера 1.
2. Перейдите к редактированию экрана. Создайте графические элементы как на рис. 18.

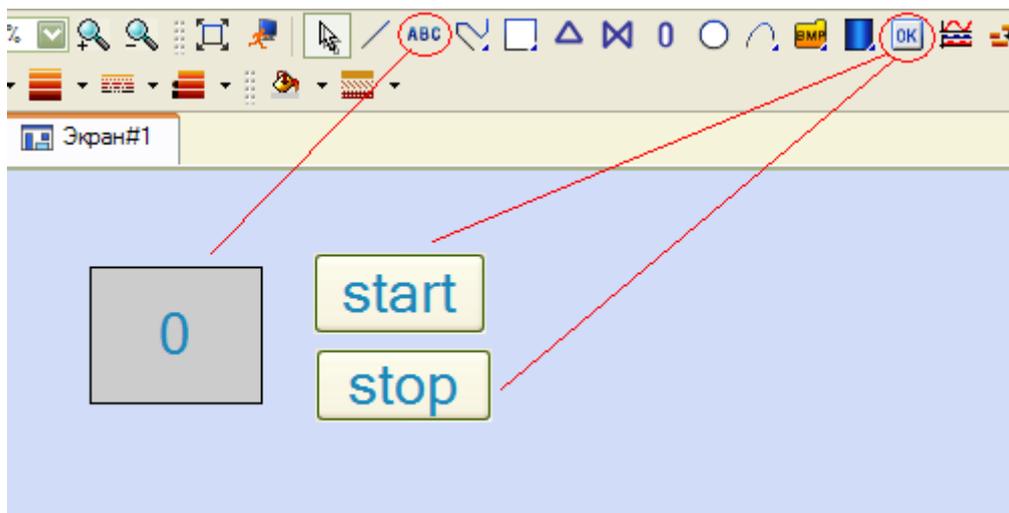


Рисунок 18 – Размещение графических элементов на экране

3. В шаблоне экрана создайте 2 аргумента типа int: Number (IN) и on (IN/OUT).
Экран → ПК → Свойства → Аргументы (см. рис. 19).

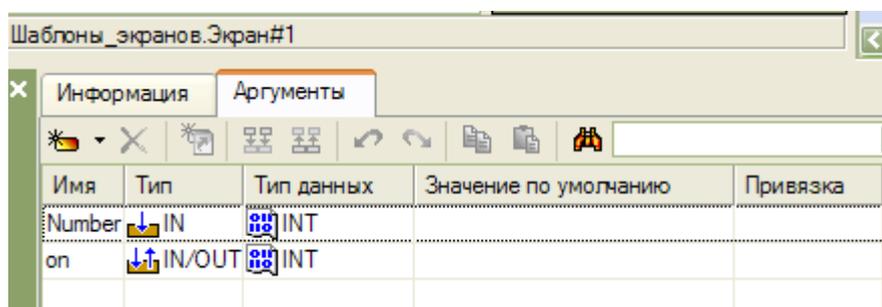


Рисунок 19 – Создание аргументов экрана

4. В свойствах Текста установите привязку к аргументу экрана Number. Вид индикации выберите «Значение». В поле Формат установите «Integer». (см. рис. 20)

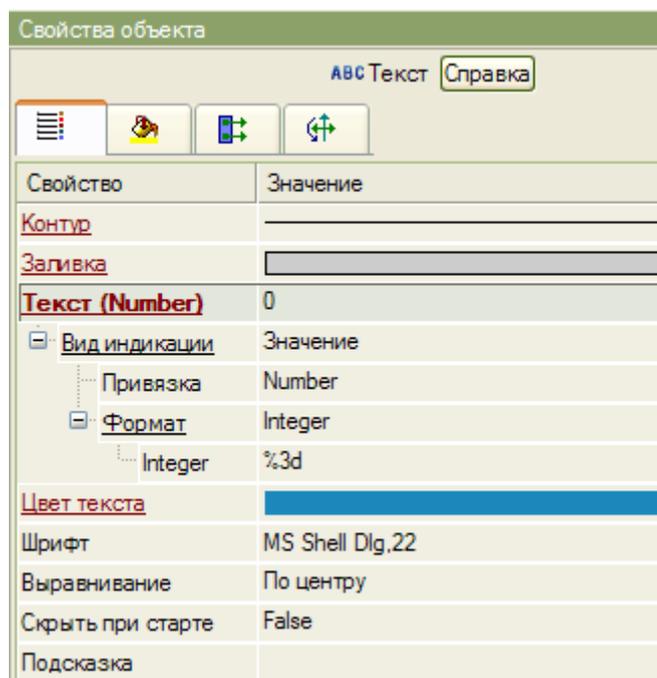


Рисунок 20 – Установка свойств графического элемента Текст

5. В свойствах Кнопки «start» создайте событие Send Value – Передать значение. В поле Тип передачи выберите «Прямая», в поле Значение введите 1, в поле Результат сделайте привязку к аргументу экрана on. (см. рис. 21)

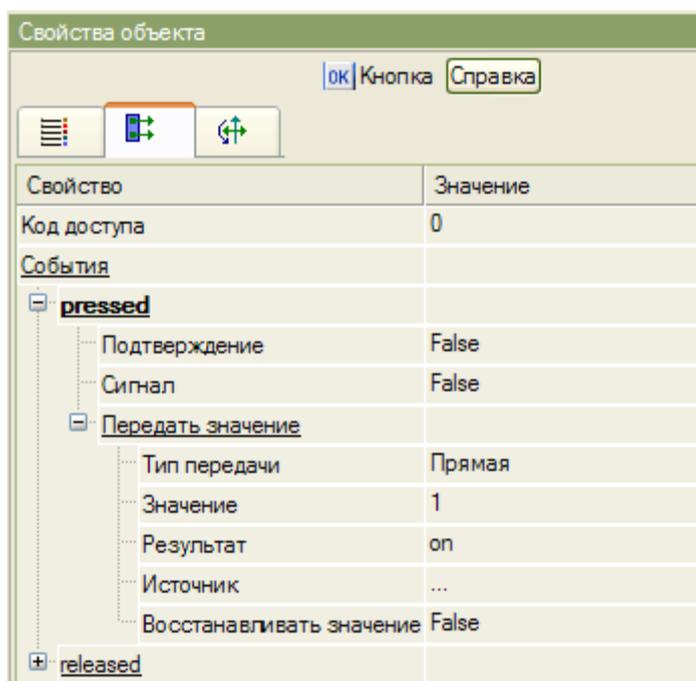


Рисунок 21 – Установка свойств кнопки «start»

6. В свойствах Кнопки «stop» создайте событие Send Value – Передать значение. В поле Тип передачи выберите «Прямая», в поле Значение введите 0, в поле Результат сделайте привязку к аргументу экрана on.

7. Создайте компонент Программа. Перейдите к созданию аргументов программы. Создайте аргументы как на рис 22.

Имя	Тип	Тип данных	Значение по умолчанию
Num	↑ OUT	INT	
on	↓ IN	INT	

Рисунок 22 – Создание аргументов программы

8. Добавьте код в программу, например как на рис. 23.

```

PROGRAM
  VAR_OUTPUT Num : INT; END_VAR
  VAR_INPUT on : INT; END_VAR

  if on == 1 then
    if Num < 100 then Num = Num + 1;
    else Num = 0;
    end_if;
  end_if;

END_PROGRAM

```

Рисунок 23 – Окончательный вид программы

9. Откомпилируйте программу: *Программа* → *Компиляция* (или *F7*).

10. Перейдите в свойства канала вызова экрана узла RTM. В закладке Аргументы привяжите аргумент экрана Number к аргументу программы Num (см. рис 24).

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Number	↓ IN	INT		Программа#1:2:Num(Система.RTM_1.Каналы)
on	↓ IN/OUT	INT		

Рисунок 24 – Привязка аргумента канала экрана к аргументу канала программы

11. Перейдите в свойства канала вызова канала узла RTM. В закладке Аргументы привяжите аргумент канала on к аргументу экрана on (см. рис 25).

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Num	↑ OUT	INT		
on	↓ IN	INT		Экран#1:1:on(Система.RTM_1.Каналы)

Рисунок 25 – Привязка аргумента канала экрана к аргументу канала программы

12. Сохраните проект (*Файл* → *Сохранить*). Сохраните проект для монитора реального времени (*Файл* → *Сохранить для MPB*). Запустите профайлер (*Файл* → *Отладка*). Результат выполненного проекта смотрите на рис. 26.1 и рис 26.2.

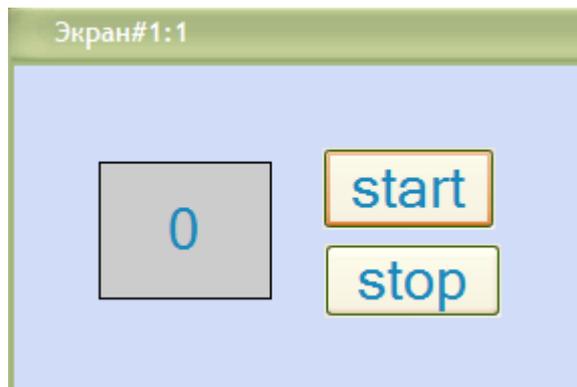


Рисунок 26.1 – Перед началом счета

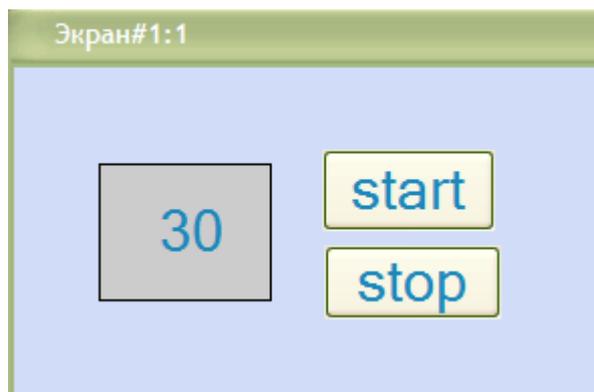


Рисунок 26.2 – После работы счетчика

13. Остановите выполнение проекта. Закройте профайлер.

Задание 3 (Движение объекта)

По нажатию на кнопку «start» объект начинает двигаться. По нажатию на кнопку «stop» объект останавливается. Движение объекта начинается с момента, на котором он был остановлен. После прохождения объектом траектории движения, он начинает движение заново.

1. Выполните пункты с 1 по 5 примера 1.
 2. Перейдите к редактированию экрана. Создайте графические элементы как на рис.
- 27.

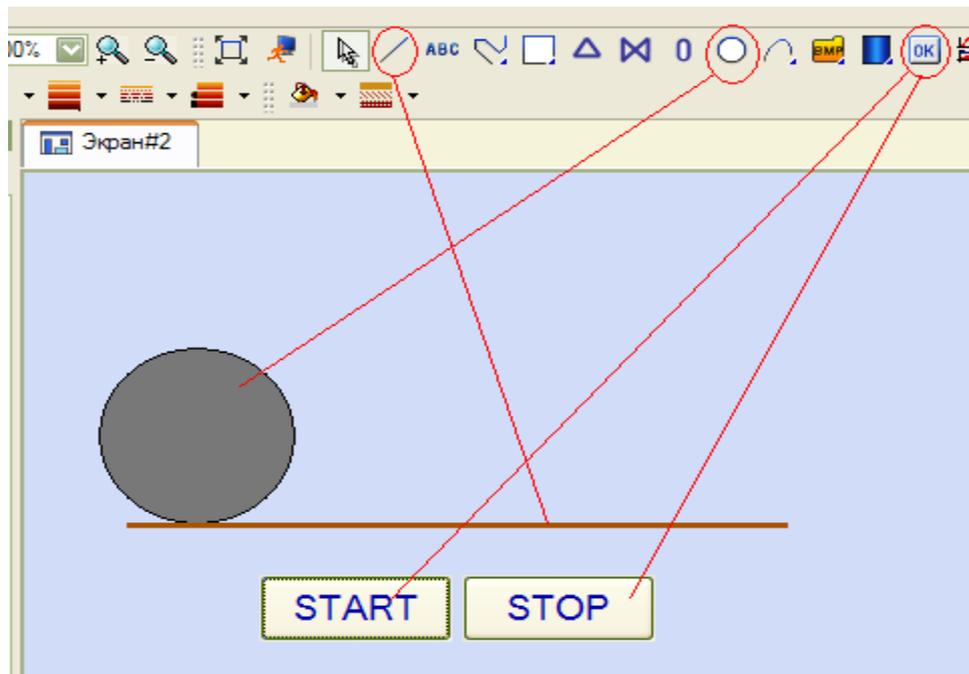


Рисунок 27 – Размещение графических элементов на экране

3. В шаблоне экрана создайте 2 аргумента типа int: Move (IN) и on (IN/OUT).
 Экран → ПК → Свойства → Аргументы (см. рис. 28).

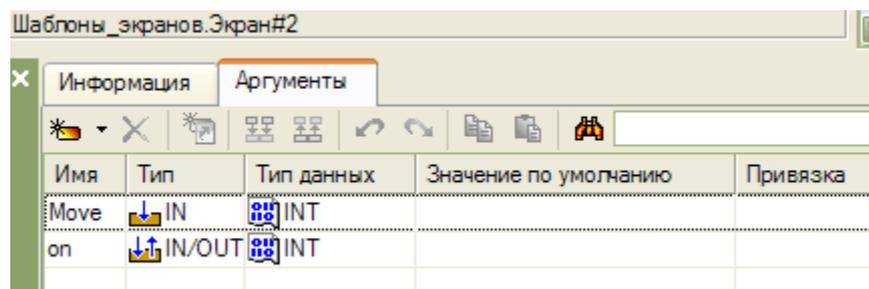


Рисунок 28 – Создание аргументов экрана

4. В свойствах Сферы перейдите на вкладку Динамическая трансформация. Установите флаг «Перемещать». Теперь установите привязку к аргументу экрана Move. (см. рис. 29.1 и рис 29.2)

При работе в реальном времени ГЭ перемещается вдоль траектории, которая задается как ломаная линия (количество узлов ломаной не ограничено). Текущее положение ГЭ зависит от значения привязанного аргумента (числовой аргумент для привязки выбирается в списке Привязка, от значений, заданных для узлов траектории, и флага Перемещать плавно).

Под заданием траектории понимается задание положения ее узлов и задание значений для этих узлов.

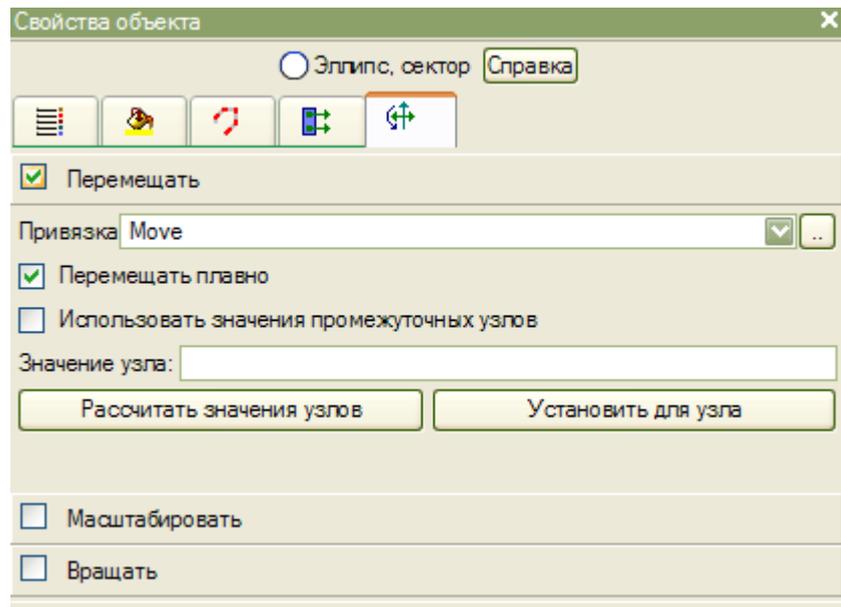


Рисунок 29.1 – Свойства объекта Эллипс

По умолчанию траектория динамического перемещения представляет собой отрезок от точки привязки ГЭ до центра ограничивающего прямоугольника, т.е. имеет 2 узла. Значения для этих узлов устанавливаются равными 0 и 100 соответственно (значение, заданное для узла, отображается в окне Значение узла при наведении курсора на узел).

Узел, первоначально размещенный в точке привязки ГЭ (обозначается красной точкой), остается крайним узлом при любых манипуляциях с траекторией и в дальнейшем называется первым узлом.

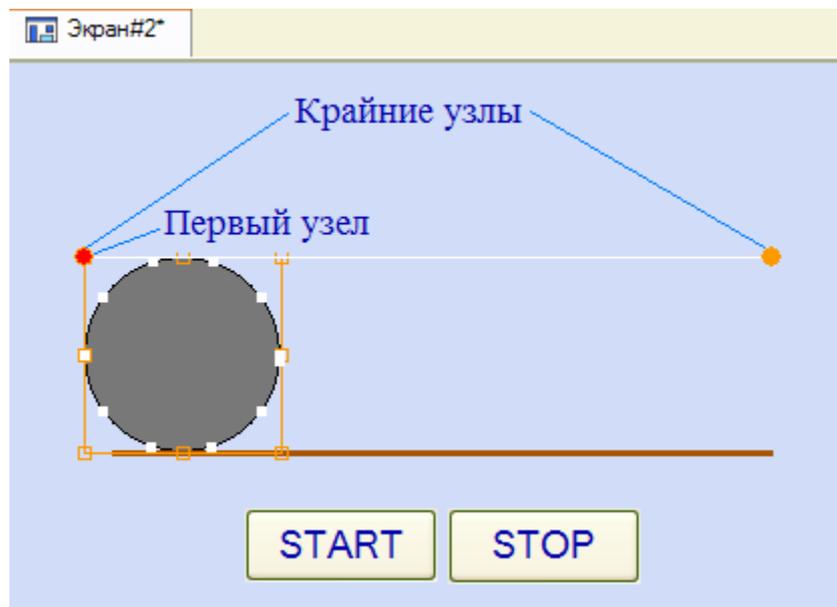


Рисунок 29.2 – Обозначение траектории движения сферы

Значения для крайних узлов траектории задаются вручную. Для этого нужно выделить крайний узел, ввести число в окне Значение узла и нажать кнопку Установить для узла.

5. В свойствах Кнопки «start» создайте событие Send Value – Передать значение. В поле Тип передачи выберите «Прямая», в поле Значение введите 1, в поле Результат сделайте привязку к аргументу экрана on. (см. рис. 30)

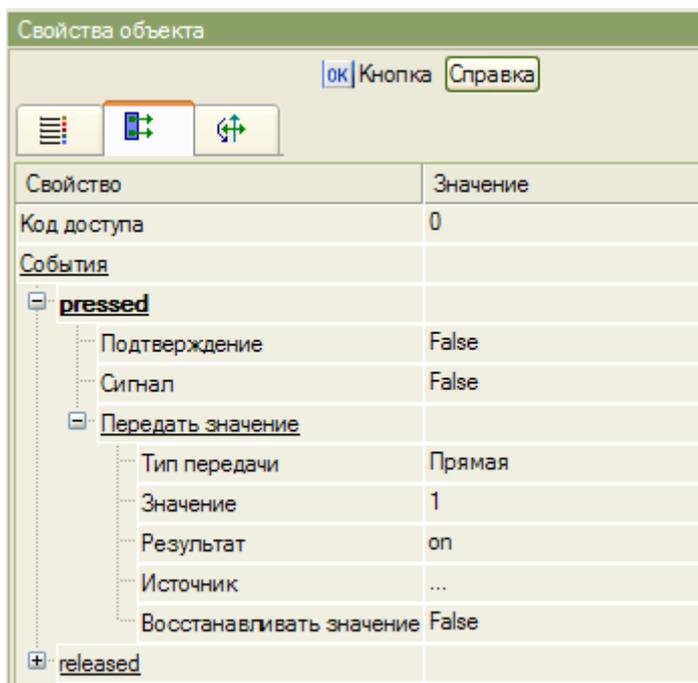


Рисунок 30 – Установка свойств кнопки «start»

6. В свойствах Кнопки «stop» создайте событие Send Value – Передать значение. В поле Тип передачи выберите «Прямая», в поле Значение введите 0, в поле Результат сделайте привязку к аргументу экрана on.

7. Создайте компонент Программа. Перейдите к созданию аргументов программы. Создайте аргументы как на рис 31.

Имя	Тип	Тип данных	Значение по умолчанию
on	IN	INT	
Move	OUT	INT	

Рисунок 31 – Создание аргументов программы

8. Добавьте код в программу (см. рис. 32).

```

PROGRAM
VAR_INPUT on : INT; END_VAR
VAR_OUTPUT Move : INT; END_VAR

if on == 1 then
  if Move < 100 then Move = Move + 1;
  else Move = 0;
  end_if;
end_if;

END_PROGRAM

```

Рисунок 32 – Окончательный вид программы

9. Откомпилируйте программу: *Программа* → *Компиляция* (или F7).

10. Перейдите в свойства канала вызова экрана узла RTM. В закладке Аргументы привяжите аргумент экрана Move к аргументу программы Move (см. рис 33).

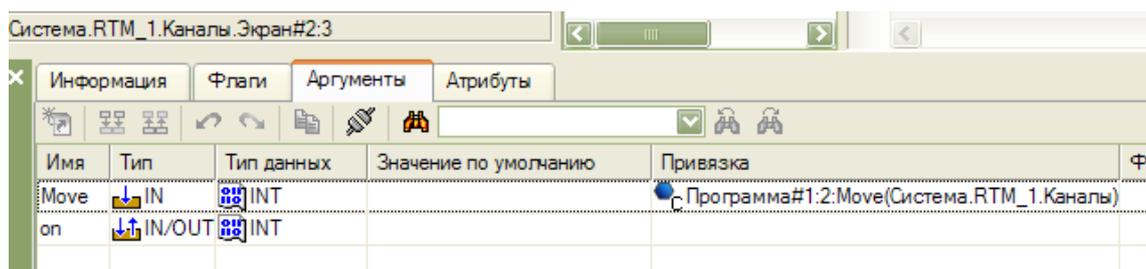


Рисунок 33 – Привязка аргумента канала экрана к аргументу канала программы

11. Перейдите в свойства канала вызова канала узла RTM. В закладке Аргументы привяжите аргумент канала on к аргументу экрана on (см. рис 34).

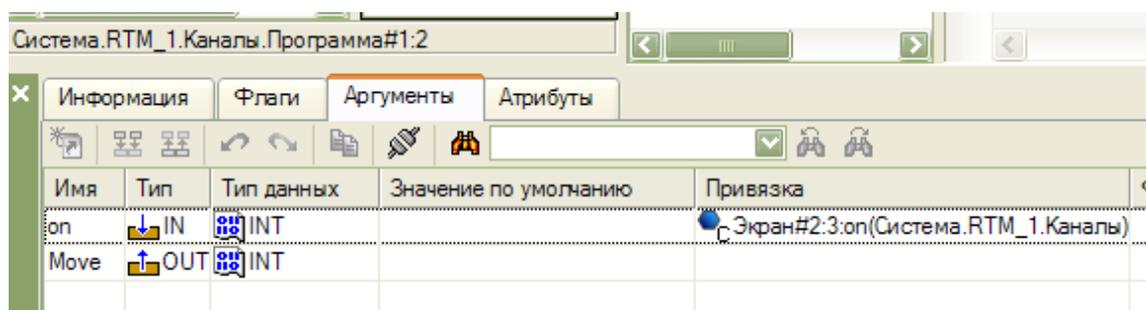


Рисунок 34 – Привязка аргумента канала экрана к аргументу канала программы

12. Сохраните проект (*Файл* → *Сохранить*). Сохраните проект для монитора реального времени (*Файл* → *Сохранить для MPB*). Запустите профайлер (*Файл* → *Отладка*). Результат выполненного проекта смотрите на рис. 35.1 и рис 35.2.

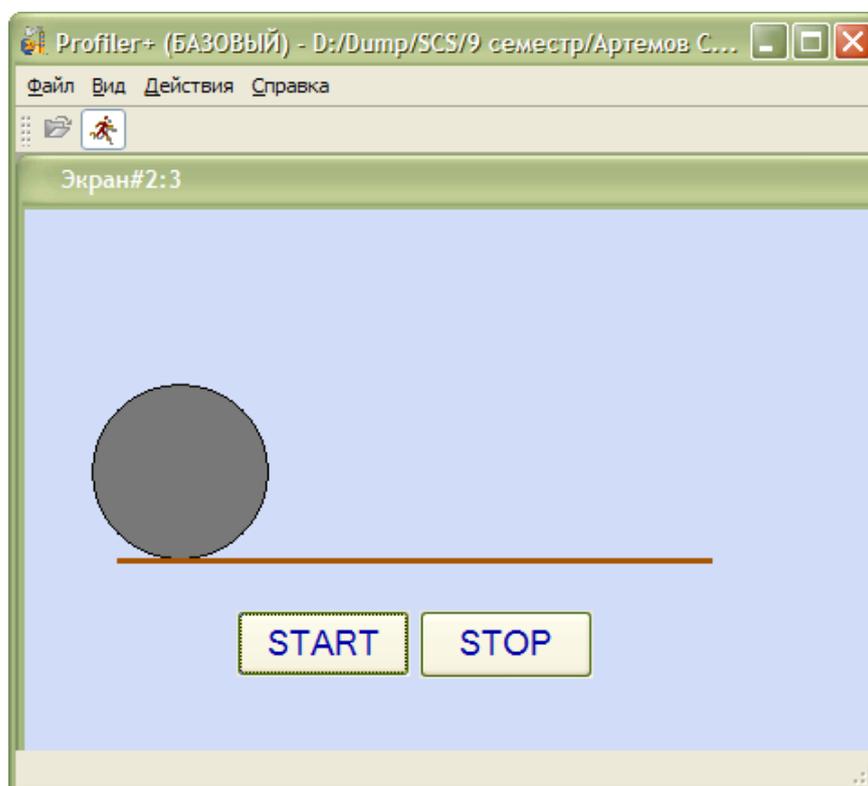


Рисунок 35.1 – Результат выполнения проекта

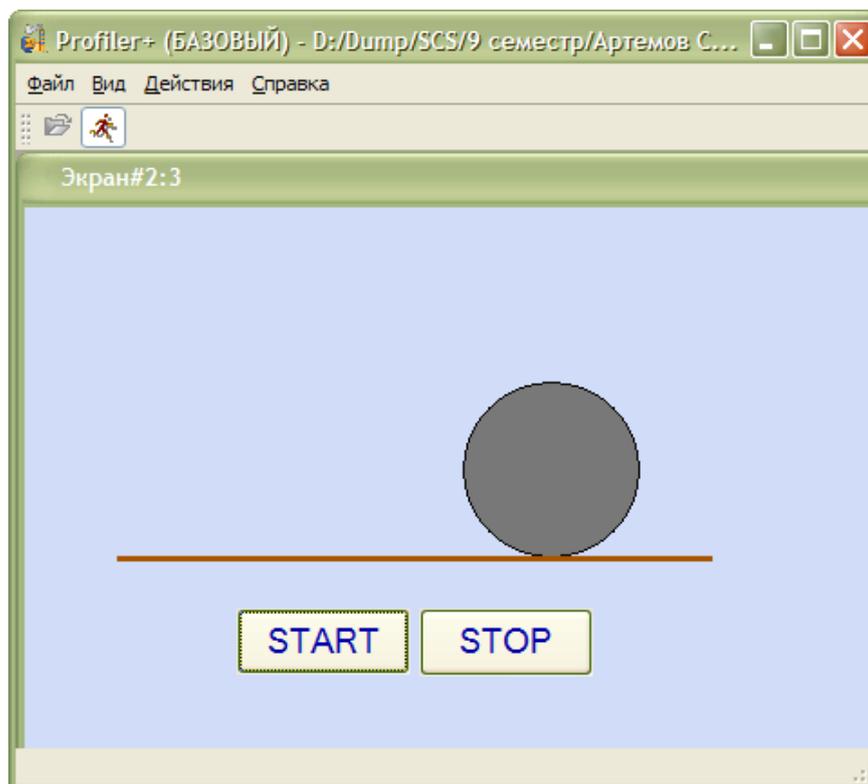


Рисунок 35.2 – Результат выполнения проекта

13. Остановите выполнение проекта. Закройте профайлер.

Задание 4 (по вариантам)

Необходимо разработать интерфейс оператора автоматизированной системы в соответствии с вариантом задания.

- 1) Для автоматизированной системы предусмотреть 4-8 управляющих воздействий на алгоритм функционирования.
- 2) Разработать несколько «графических» исполнительных элементов, которые иллюстрируют действия на систему.
- 3) Создать автономный алгоритм функционирования системы.

Варианты заданий:

- 1) Модель железнодорожной стрелки.
- 2) Модель двух перекрестков: равнозначного и с приоритетом.
- 3) Модель конвейера.
- 4) Модель системы учёта электрической энергии.
- 5) Модель телефонной станции.
- 6) Модель системы контроля освещенности в помещении (жалюзи, уровень освещенности и т.п.).
- 7) Модель автомобильной заправочной станции (АЗС).
- 8) Модель системы контроля парковки по времени.
- 9) Модель системы контроля периметра вокруг здания.
- 10) Модель системы игрового автомата (выигрыш/проигрыш).
- 11) Модель контроля проходной на предприятии.

- 12) Модель системы управления движения скорой помощи по автодорогам (выбор пути, 2-3 пути).
- 13) Модель подачи воды в жилой дом, измерение потока, дублирование при аварии.
- 14) Модель системы управления освещением при входе в помещения (2-3 помещения).
- 15) Модель системы охранной сигнализации на объектах, вывод информации при срабатывании трех типов датчиков.
- 16) Модель системы управления отправкой железнодорожных составов на станции.
- 17) Модель системы управления шлюзованием.
- 18) Модель системы пожарной сигнализации в помещениях здания (5-7 помещения).
- 19) Модель системы управления зернохранилищем (перемещение по контейнерам).
- 20) Модель системы управления вылетом и посадкой самолетов.
- 21) Модель системы управления движением поезда.
- 22) Модель системы управления складом готовой продукции.
- 23) Модель системы управления морским портом.
- 24) Модель системы управления пожарной сигнализации.
- 25) Модель системы управления отоплением в квартире.
- 26) Модель системы управления гидропонной системы (тепличное хозяйство).
- 27) Модель системы управления влажностью в помещении.
- 28) Модель системы управления движением троллейбуса.
- 29) Модель управления елочной гирляндой (переключение режимов, смена цветов).
- 30) Модель системы управления рекламными щитами (изменение реклам, смена подсветки и т.п.)
- 31) Модель системы управления лифтами (грузовой и пассажирский).
- 32) Модель системы кондиционирования в помещениях.
- 33) Модель системы управления подачей воды в бассейн и сменой воды по расписанию.
- 34) Модель системы управления очистки воды на водозаборе.
- 35) Модель системы управления пилорамой.
- 36) Модель системы управления движением автобусом.
- 37) Модель химического производства.
- 38) Модель системы производства руды.
- 39) Модель системы газопровода.
- 40) Модель системы контроля газовой станции.
- 41) Модель системы управления процессом выкачивания нефти.

Требования к содержанию отчета

Отчет по лабораторной работе должен содержать следующие пункты:

- 1) Титульный лист с указанием номера варианта.

- 2) Цель и задачи лабораторной работы.
- 3) Краткие теоретические сведения.
- 4) Ход выполнения заданий 1-3.
- 5) Формулировка задания согласно своему варианту.
- 6) Описание тегов проекта.
- 7) Описание алгоритмов функционирования.
- 8) Графическое представление системы.
- 9) Проверка работы программы.
- 10) Ответы на контрольные вопросы.
- 11) Вывод по лабораторной работе.

Контрольные вопросы

- 1)