

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 № 5
на тему
«Разработка интерфейса оператора автоматизированной системы
управления в CitectSCADA»
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 5. Разработка интерфейса оператора автоматизированной системы управления в CitectSCADA	4
Цель и задачи работы	4
Теория	4
Общие сведения о CitectSCADA	5
Алгоритм создания проекта	6
Ход работы	7
Задание 1 (Управление лампочкой)	7
Задание 2 (Изменение счетчика)	13
Задание 3 (Движение объекта).....	16
Задание 4 (по вариантам)	18
Требования к содержанию отчета	21
Контрольные вопросы	21

Лабораторная работа 5. Разработка интерфейса оператора автоматизированной системы управления в CitectSCADA

Цель и задачи работы

Цель: Освоить методику разработку интерфейса оператора автоматизированной системы управления в CitectSCADA.

Задачи работы:

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

Теория

SCADA (Supervisory Control And Data Acquisition) – централизованный контроль и сбор данных. SCADA позволяет осуществлять взаимодействие с применением вычислительной техники. В конечном итоге SCADA – это средство позволяющее контролировать объект управления (ОУ) и формировать для него управляющее воздействие.

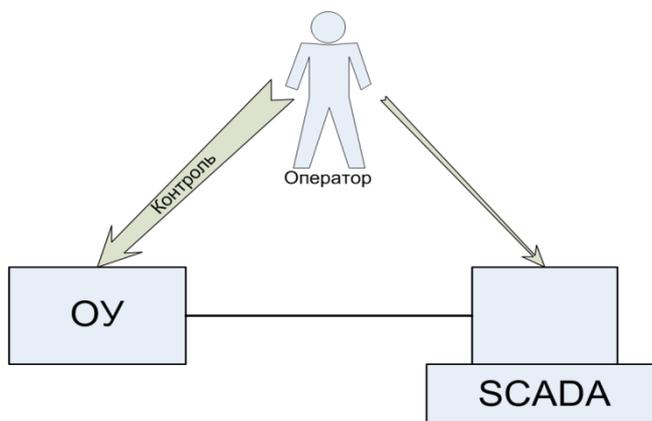


Рисунок 1 – Место SCADA в АСУ

SCADA получает только те данные, которые регламентированы протоколом связи. Если элемент SCADA имеет взаимодействие с физическим ОУ, то связь является физической (ФС), если элемент SCADA имеет взаимодействие с внутренним ОУ то соединение виртуальное (ВС) и ОУ виртуальный. Виртуальный ОУ может быть связан с физическим ОУ с помощью других вычислительных средств. Физический объект описывается протоколом, а виртуальный – с помощью OPC-средств или Generic.

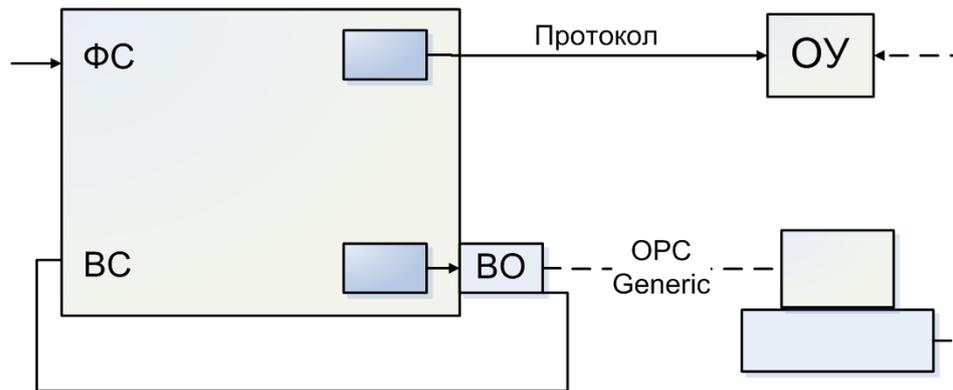


Рисунок 2 – Взаимодействие SCADA с физическим и виртуальным объектом управления

Любая SCADA состоит минимум из двух основных групп модулей:

- 1) среда разработки (*.prj);
- 2) среда исполнения (*.exe).

Задача среды разработки – это разработка интерфейса оператора и коммуникация его с объектом. Среда исполнения необходима для запуска проектов, разработанных в среде разработки. Деление необходимо, чтобы разграничить процесс создания от конечных запускаемых модулей.

Компоненты SCADA:

- 1) Устройства физические и виртуальные – это физическое описание адреса устройства в распределенной системе с декларированием по протоколам. Устройство – это физический контроллер, имеющий уникальный адрес.
- 2) Теги, переменные или каналы – это физические датчики и исполнительные элементы, подключенные к устройству ввода/вывода и декларированные протоколом. Теги создаются для каждого элемента протокола с уникальным адресом в рамках одной группы, но для разных групп, адреса тегов могут совпадать.
- 3) Сервер ввода/вывода – это вычислительный узел со SCADA, имеющий физическое подключение с объектом управления. Служит для построения распределенных вычислительных систем.
- 4) Системные компоненты – служат для управления временем и узлами вычислительного компонента (объявление таймеров, прерываний).
- 5) Компоненты для формирования и хранения alarm-ов – сообщения об аварийных ситуациях.
- 6) Средства для работы с трендами – графики параметров в текущий момент времени. Тренд – это множество параметров выведенных на одном графике.

Общие сведения о CitectSCADA

CitectSCADA – это SCADA-приложение, разработанное Citect Technologies (Австралия).

Среда разработки:

Citect Explorer – это средство управления проектом. Позволяет создавать страницы проекта и устанавливать параметры опроса и параметры графических элементов. Обеспечивает выбор компонентов системы.

Citect Builder – служит для просмотра и создания всех компонентов системы. Обеспечивает проверку ошибок компиляции. Для создания графических элементов используется модификация Citect GraphBuilder.

Среда исполнения:

Citect Runtime – это система запуска приложения, разработанного в SCADA, и проверки его функционирования в режиме реального времени и в режиме эмуляции.

В Citect при использовании не тривиальных функций управления осуществляется программирование проекта на встроенном языке CiCode. Циклы не используются, так как сам модуль запускается циклически.

Алгоритм создания проекта

- 1) Запуск Citect Explorer.
- 2) Создание проекта или ссылки на проект.

При создании проекта в окне Citect Explorer формируется шаблон проекта, включающий средства создания страниц интерфейса оператора и всех компонент ввода/вывода.

- 3) Создание сервера ввода/вывода.

Необходимо указать имя сервера.

- 4) Создание устройства ввода/вывода, с которым будет взаимодействовать SCADA.

Задается адрес устройства (от 1 и выше), тип устройства (Genetic – общее устройство) и тип памяти, с которой данные будут приниматься (если устройство виртуальное, то оперативная – Memory или на накопителях – Disk).

- 5) Создание тегов для всех датчиков и исполнительных устройств (в разделе Variable Tag).

Необходимо указать уникальное имя, тип передаваемой переменной и адрес. Тип является наименованием группы (int, decimal).

- 6) Создание страниц интерфейса оператора.

Существует два вида страниц: стартовая (всегда одна) и вызываемая (ни одной или несколько).

Стартовая страница запускается при запуске Citect Runtime.

- 7) Обеспечение переключения между страницами с применением кнопок управления.

На стартовой странице необходимо реализовать кнопку выхода.

8) Размещение на страницах всех датчиков и исполнительных элементов в виде графических и управляющих элементов через панель инструментов.

9) Для всех исполнительных элементов в окне свойств необходимо указать действие при работе с графическим элементом.

С этой целью необходимо ввести текст в окна вкладок Access (обеспечивает реализацию строки при доступе к объекту) и Input (позволяет ввести значение).

10) Для всех датчиков необходимо в окне свойств необходимо указать обрабатываемое воздействие для отображения на экране.

Для этого существуют следующие вкладки:

Appearance – появление объекта;

Movement – движение объекта;

Fill – закрашка объекта;

Scaling – изменение масштаба объекта;

Slider – обеспечение анимации объекта.

В окне ввода текста указывается условие, при котором обрабатывается действие.

Все свойства могут работать с множеством значений.

11) Каждый элемент может быть снабжен текстовым описанием.

Примечания:

а) При необходимости в проект могут быть включены элементы из внешних графических редакторов (bmp, jpg).

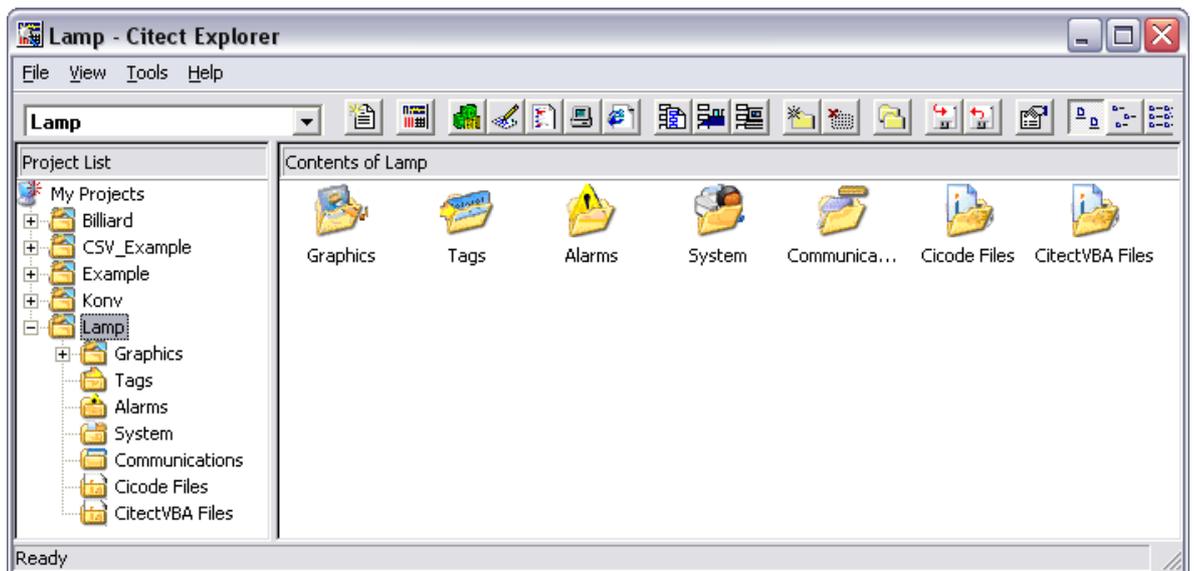
б) При создании тэгов необходимо указывать устройство ввода/вывода, через которое тэг работает.

Ход работы

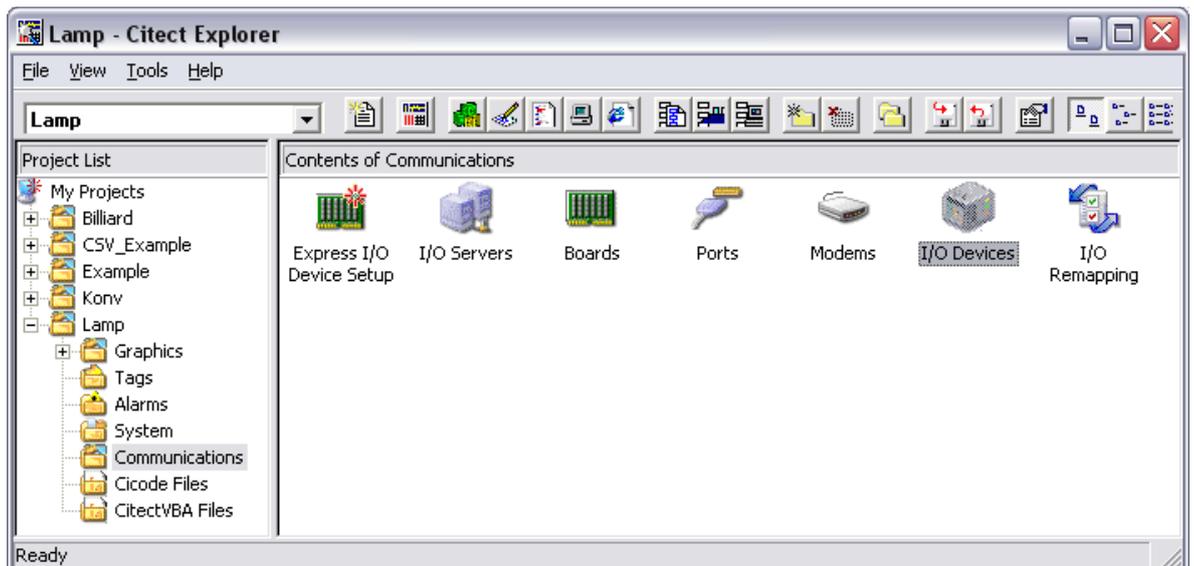
Задание 1 (Управление лампочкой)

По нажатию на кнопку лампочка включается/выключается.

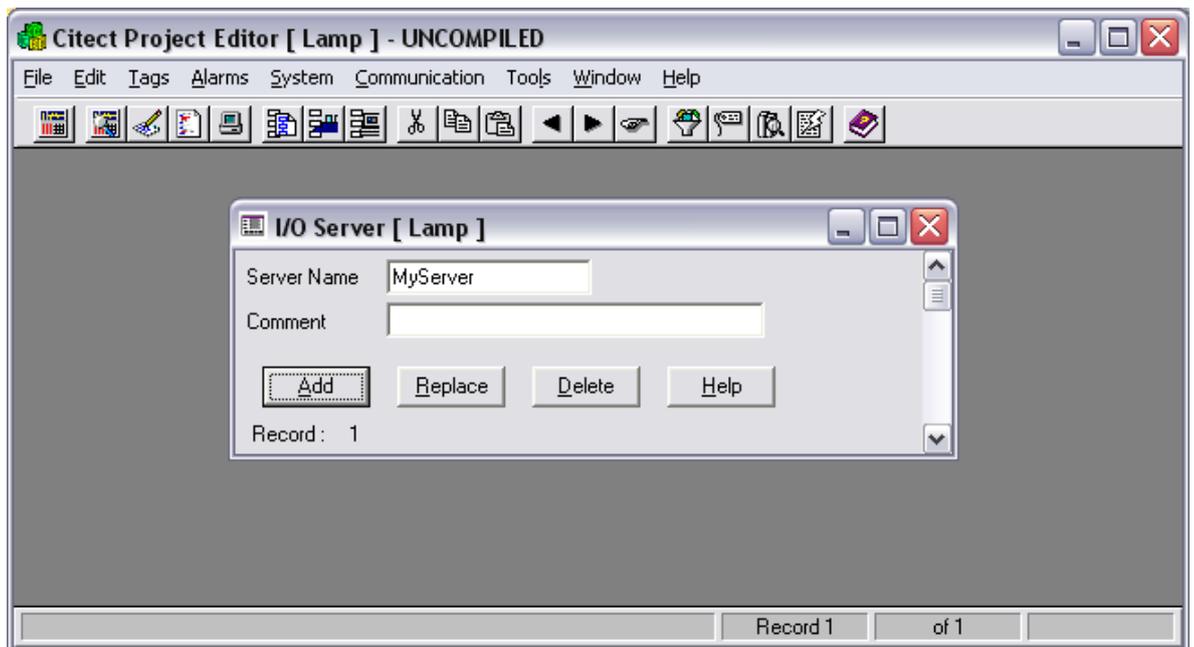
1. Запустите *Citect Explorer*.
2. Создайте проект *File* → *New Project*. Задайте имя проекта (напр., *Lamp*).



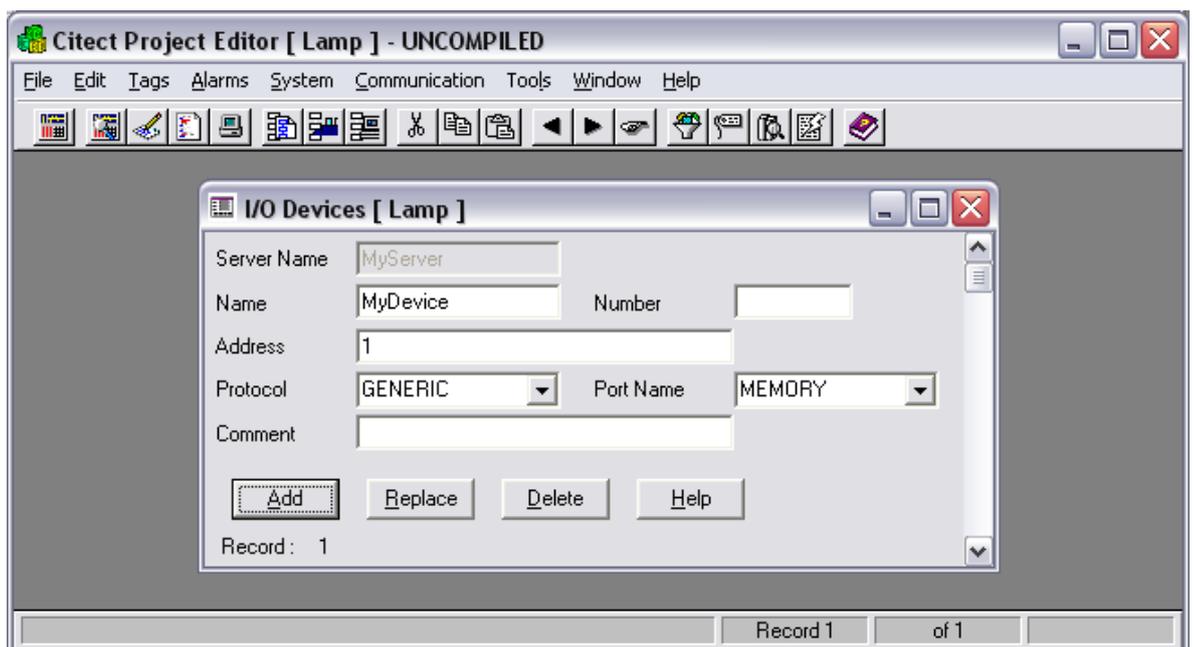
3. Создадим сервер ввода/вывода. Для этого на панели *Project List* или *Contents of Lamp* выберите *Communications* (двойным щелчком левой кнопки мыши), затем *I/O Servers*.



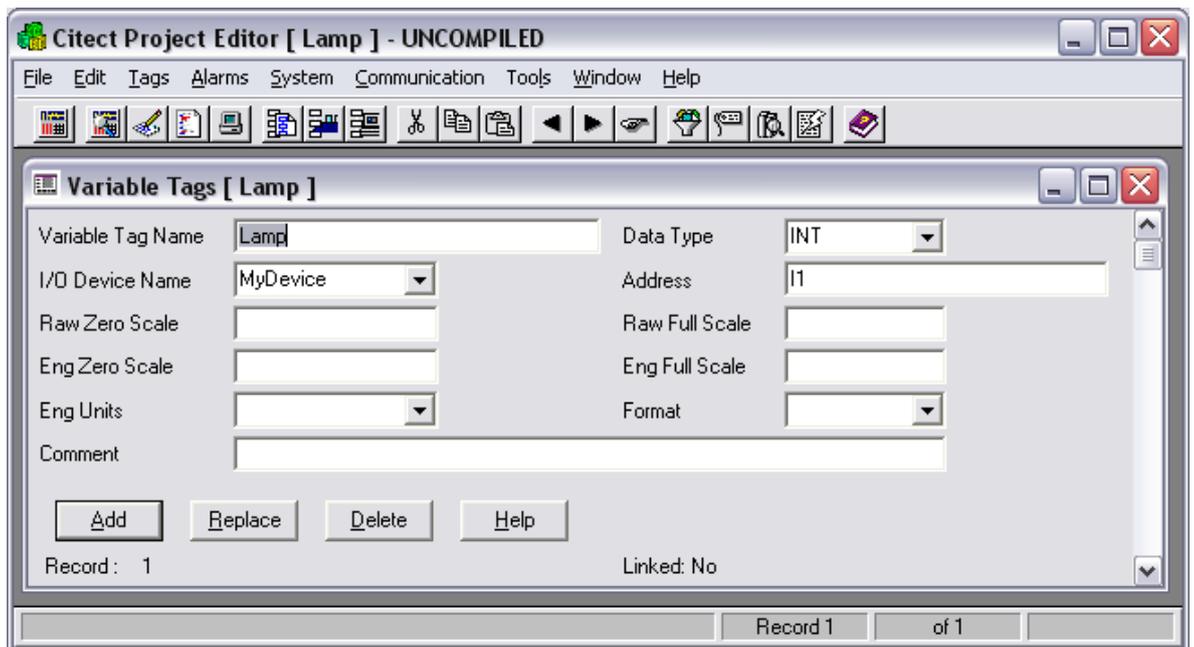
В *Citect Project Editor* в окне *I/O Server* создадим сервер: задайте имя сервера *MyServer* и нажмите *Add*.



4. Создадим устройство ввода/вывода. Для этого перейдите снова в *Citect Explorer*. Выберите *Communications*, затем *I/O Devices*. В *Citect Project Editor* в окне *I/O Devices* задайте имя устройства *MyDevice*, адрес устройства от 1 и выше, тип устройства *Generic*, тип памяти *Memory* и нажмите *Add*.



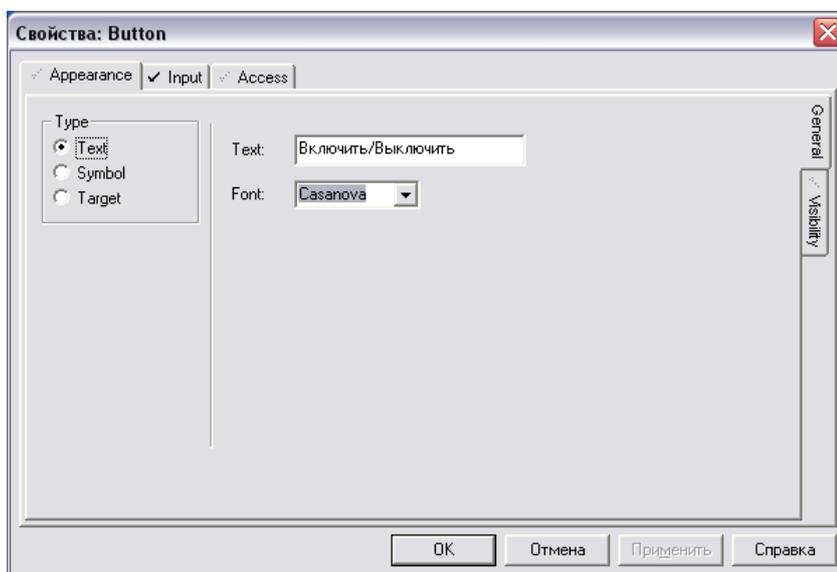
5. Создадим тег, отвечающий за состояние лампочки. Для этого перейдем в *Citect Explorer*. На панели *Project List* или *Contents of Lamp* выберите *Tags*, затем *Variable Tags*. В *Citect Project Editor* в окне *Variable Tags* задайте имя тега *Lamp*, выберите устройства ввода/вывода *MyDevice*, тип передаваемой переменной и адрес.



6. Создадим стартовую страницу интерфейса оператора. Для этого перейдем в *Citect Explorer*. На панели *Project List* или *Contents of Lamp* выберите *Graphics*, затем *Pages* и *Create a new page*. В *Citect Graphics Builder* в окне *Use Template* выберите шаблон страницы (*Normal*), стиль (*Xp_style*) и нажмите *OK*.

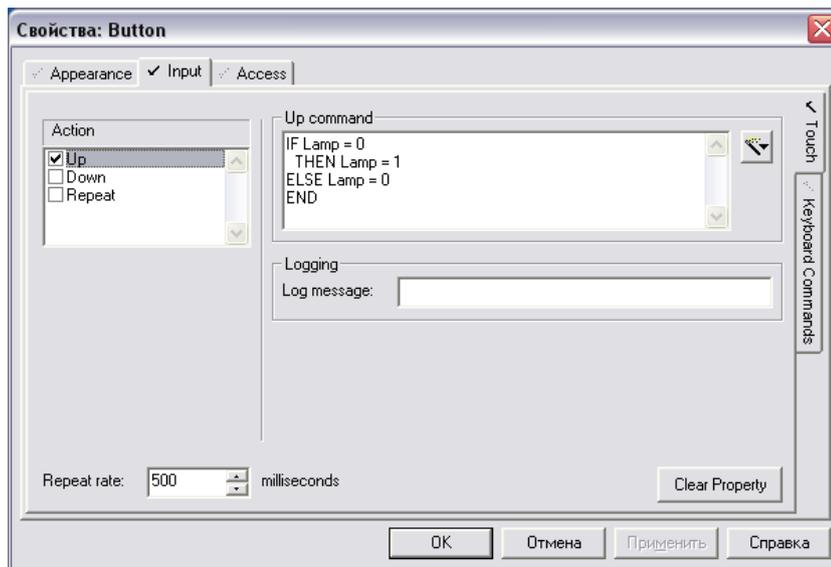
С помощью панели инструментов *Tools* создайте на странице две кнопки (одна – для управления лампочкой, другая – для выхода) и круг.

В свойстве *Appearance* кнопки управления лампочкой выберите тип *Text*, введите текст, который будет отображаться на кнопке (напр., «Включить/Выключить»).



В свойстве *Input* введите код, обрабатывающий нажатие кнопки, в текстовое окно *Up command*. Код может быть следующий:

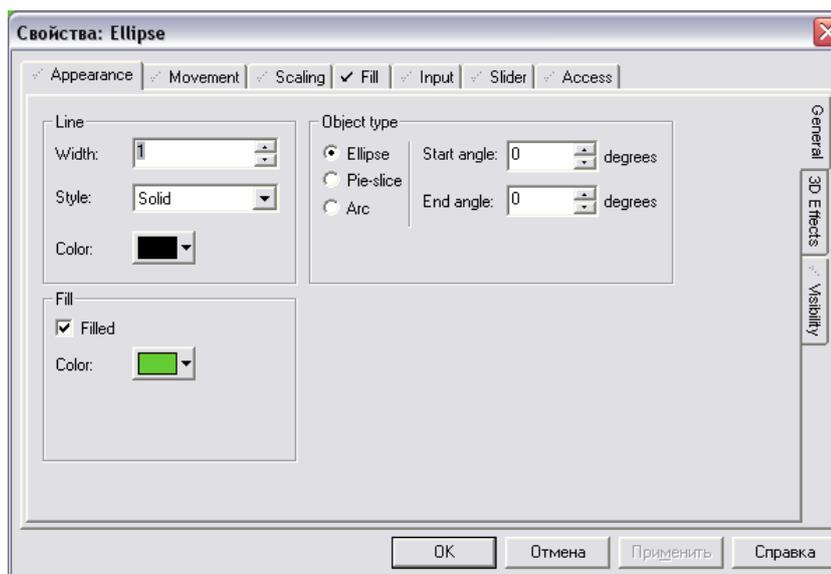
```
IF Lamp = 0
    THEN Lamp = 1
ELSE Lamp = 0
END
```



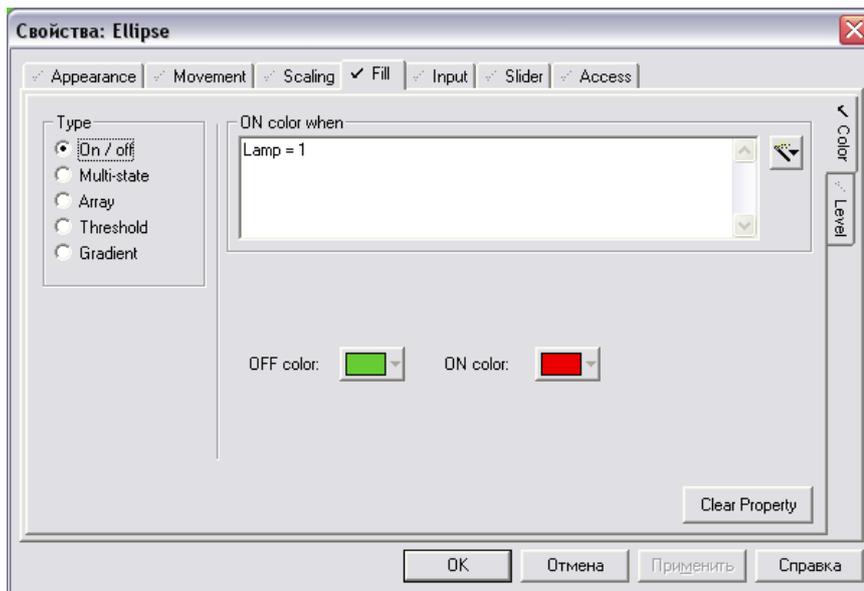
Нажмите *OK*.

Аналогично в свойстве *Appearance* кнопки выхода выберите тип *Text*, введите текст, который будет отображаться на кнопке (напр., «Выход»).

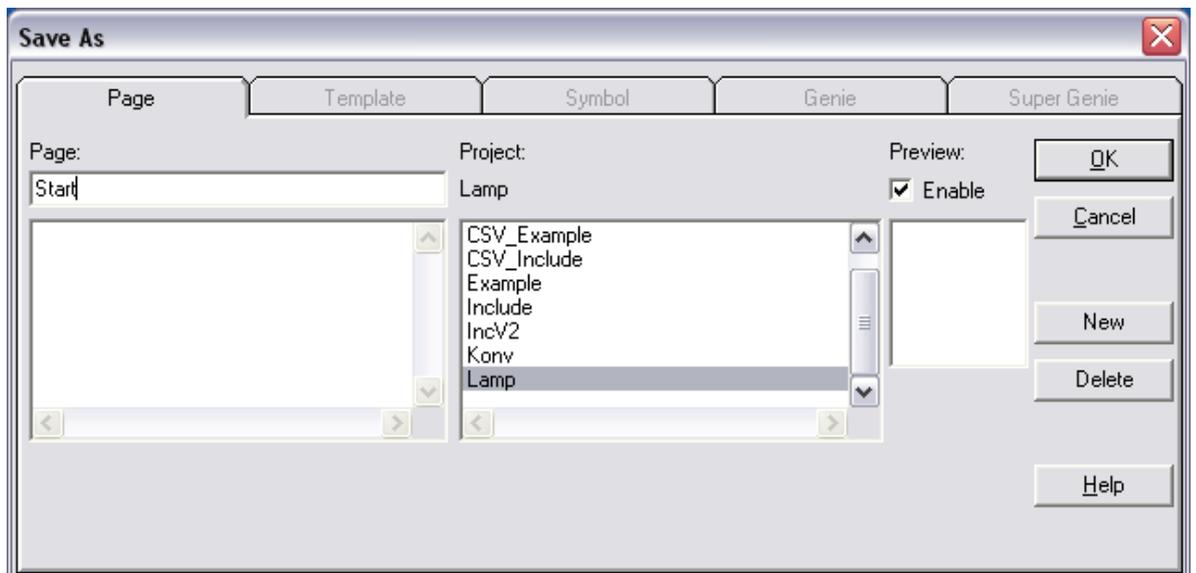
В свойстве круга *Appearance* в разделе *Fill* выберите цвет лампочки (цвет заливки круга) в выключенном состоянии.



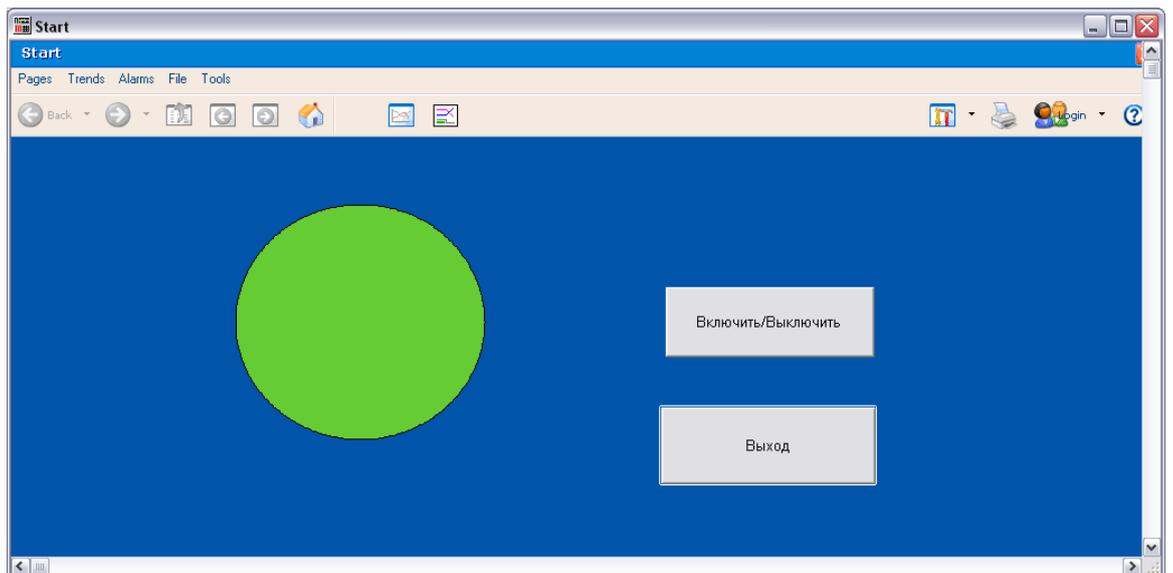
В свойстве *Fill* выберите цвет лампочки (цвет заливки круга) во включенном состоянии и задайте условие включения лампочки в текстовом окне *ON color when*, напр., *Lamp = 1*.



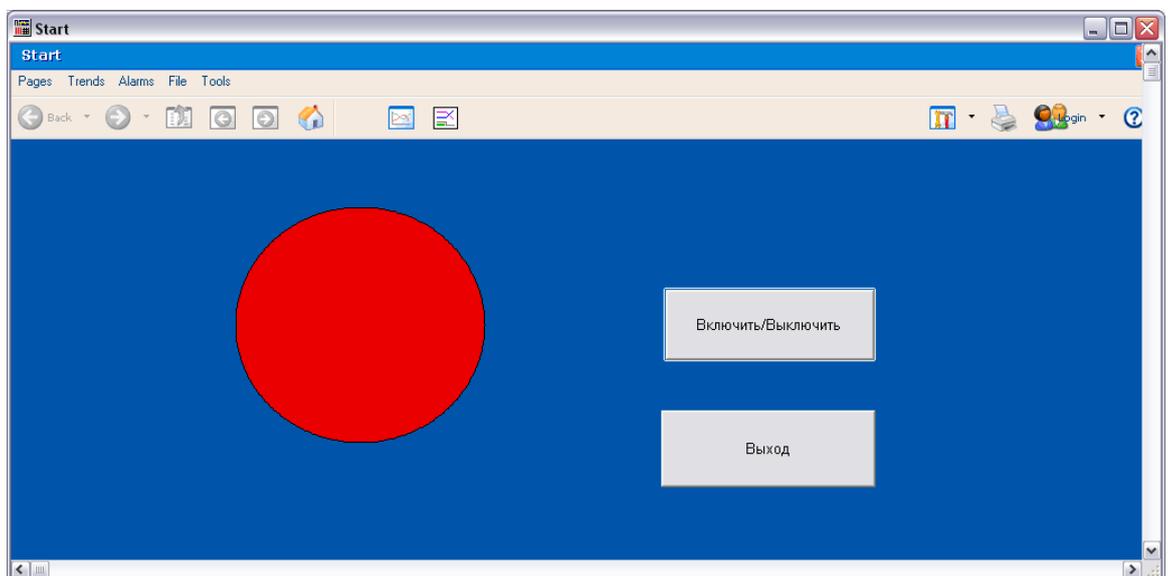
Сохраните страницу *File* – *Save As*. В появившемся диалоге задайте имя страницы и нажмите *OK*.



7. Запустите приложение *Run Project*. Результат выполнения:



После нажатия на кнопку Включить/Выключить лампочка меняет цвет:



После нажатия на кнопку Выход приложение завершает работу.

Задание 2 (Изменение счетчика)

По нажатию на кнопку изменяется значение счетчика.

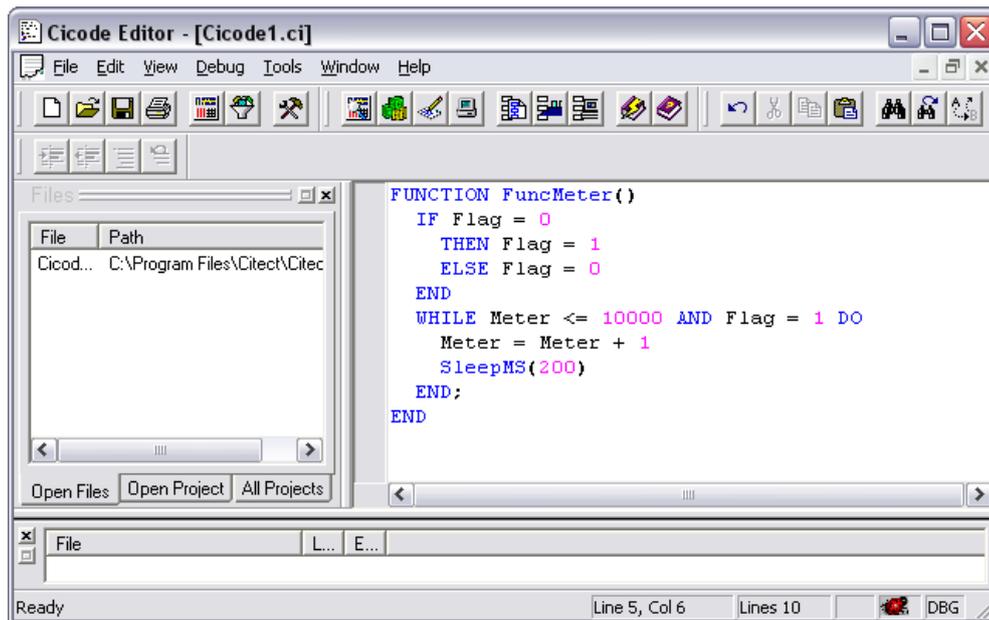
1. Выполните пункты с 1 по 4 примера 1.
2. Аналогично примеру 1, создайте тег, отвечающий за включение/выключение счетчика (напр., *Flag*) и тег, являющийся самим счетчиком (напр., *Meter*).
3. Создадим свою функцию *FuncMeter()* для управления включением/выключением счетчика. Для этого перейдем в *Citect Explorer*. На панели *Project List* или *Contents of Lamp* выберите *Cicode Files*, затем *Create a new Cicode File*. В окне *Cicode Editor* создайте функцию *File* – *> New*. В появившемся диалоге выберите тип документа *Cicode* и нажмите *OK*. Напишите ее код, например :

```
FUNCTION FuncMeter ()
    IF Flag = 0
```

```

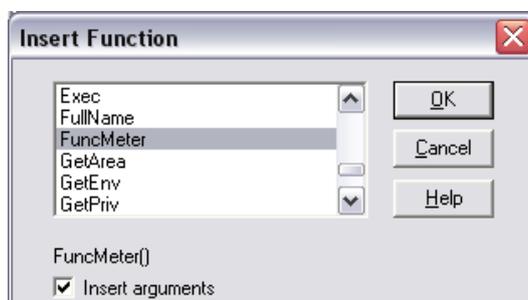
    THEN Flag = 1
    ELSE Flag = 0
END
WHILE Meter <= 10000 AND Flag = 1 DO
    Meter = Meter + 1
    SleepMS(200)
END;
END

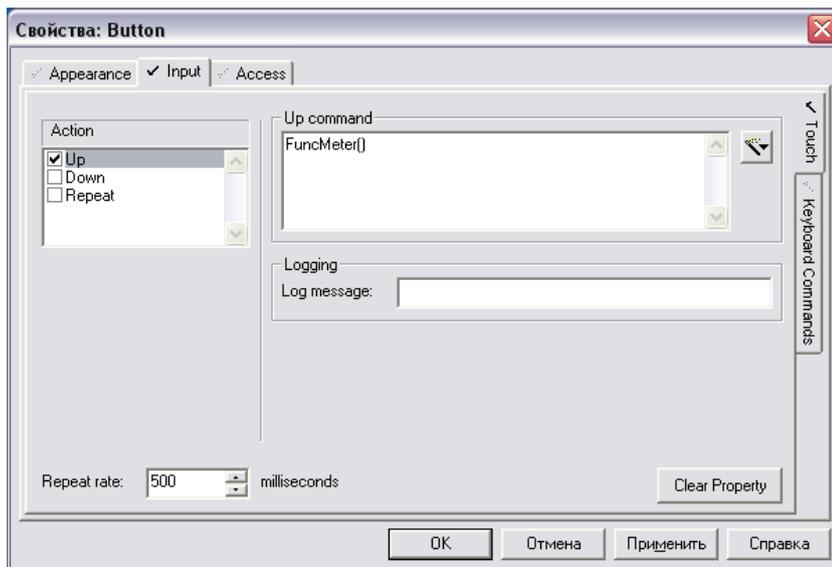
```



4. Аналогично примеру 1, создайте стартовую страницу, выберите для нее шаблон, стиль и создайте кнопку выхода.

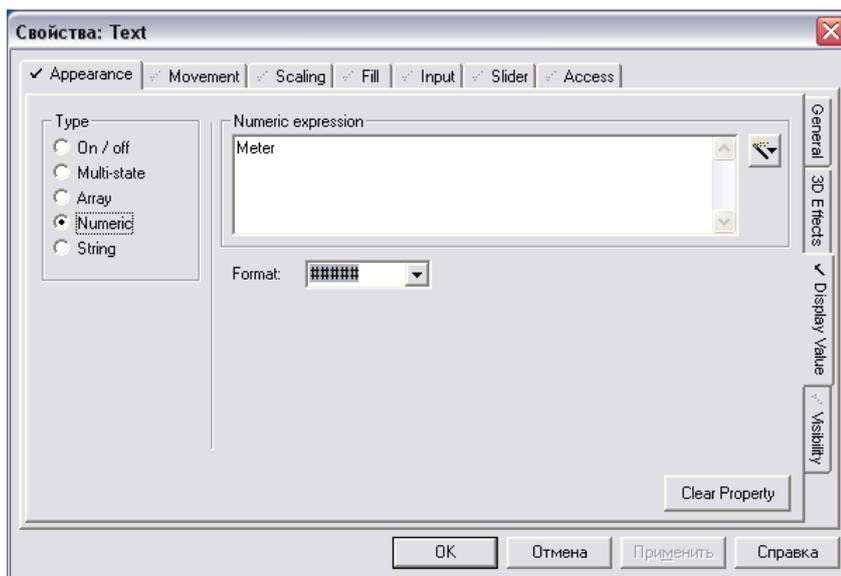
Создайте еще одну кнопку, которая будет управлять включением/выключением счетчика. В ее свойстве *Input* с помощью помощника, находящегося справа от текстового окна *Up command*, выберите *Insert function*. В появившемся диалоге выберите нашу функцию *FuncMeter* и нажмите *OK*.





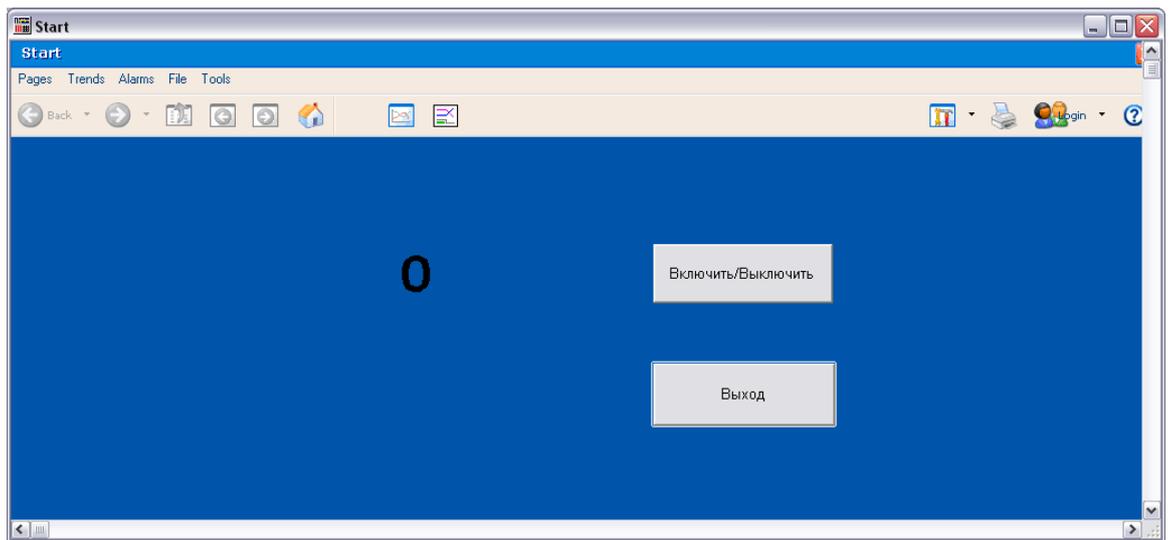
Нажмите *OK*.

Создайте на странице объект *Number* (поле ввода/вывода) с помощью панели инструментов *Tools* . В его свойстве *Appearance* задайте тип *Numeric*, выберите формат, свяжите его с тегом *Meter* в текстовом окне *Numeric expression*.

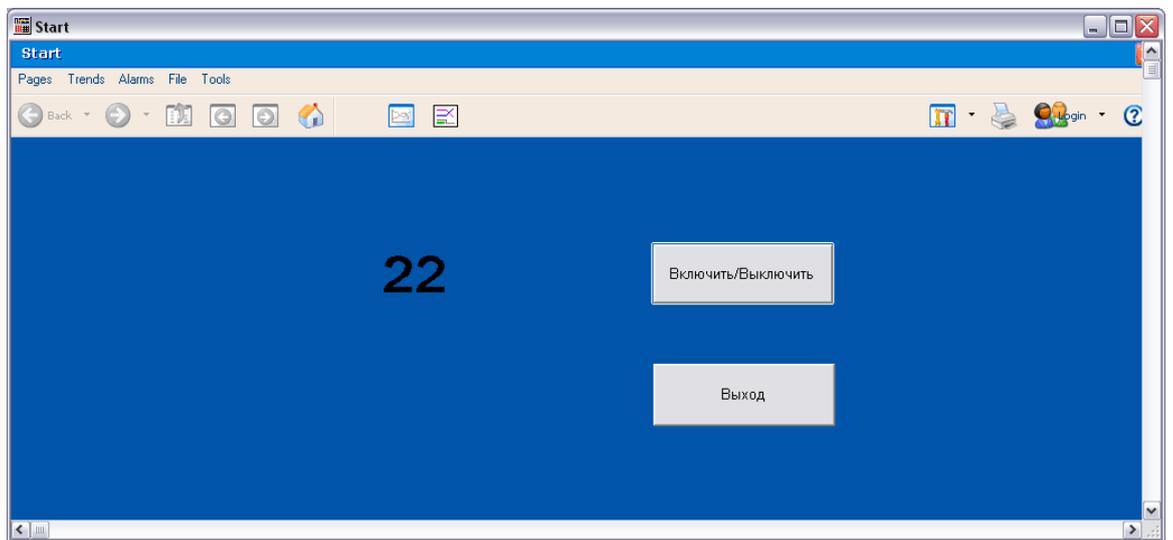


Сохраните страницу задав ее имя.

5. Запустите приложение *Run Project*. Результат выполнения:



После нажатия на кнопку Включить/Выключить включается счетчик (если счетчик был включен, то выключается):



После нажатия на кнопку Выход приложение завершает работу.

Задание 3 (Движение объекта)

По нажатию на кнопку объект двигается.

1. Выполните пункты с 1 по 4 примера 1.
2. Аналогично примеру 1, создайте тег, отвечающий за включение/выключение движения (напр., *Flag*) и тег перемещения (напр., *Move*).

3. Аналогично примеру 2, создадим свою функцию *FuncMove()* для управления включением/выключением движения. Код функции следующий:

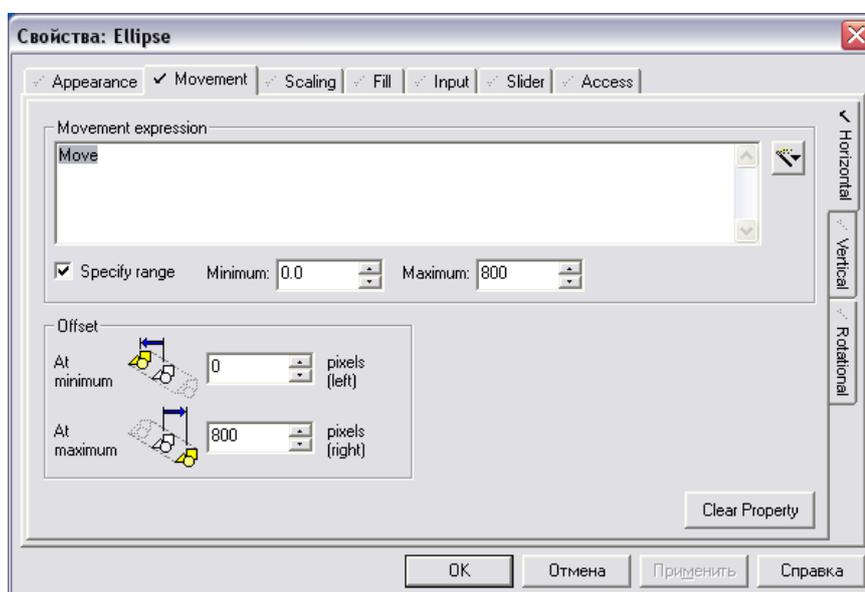
```
FUNCTION FuncMove ()
  IF Flag = 0
    THEN Flag = 1
    ELSE Flag = 0
  END
  WHILE Move <= 800 AND Flag = 1 DO
```

```
Move = Move + 1
SleepMS (100)
END;
END
```

4. Аналогично примеру 1, создайте стартовую страницу, выберите для нее шаблон, стиль и создайте кнопку выхода.

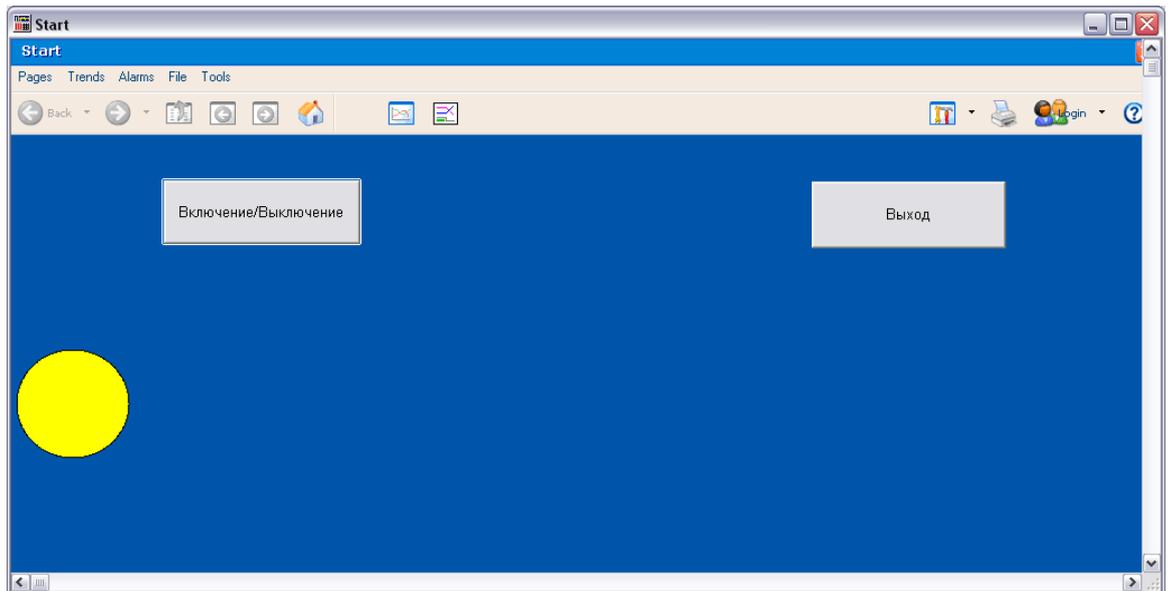
Создайте еще одну кнопку, которая будет управлять включением/выключением движения и, аналогично примеру 2, свяжите ее с нашей функцией *FuncMove*.

Создайте графический объект, например круг. В его свойствах *Appearance* выберите цвет заливки. В свойствах *Movement* на вкладке *Horizontal* свяжите объект с его перемещением *Move* в текстовом окне *Movement expression*, поставьте галочку в *Specify range* и задайте максимальное местонахождение объекта относительно стартовой точки (*Maximum*) 800. аналогично задайте максимальное отклонение в поле *Offset At maximum*. Нажмите *OK*.

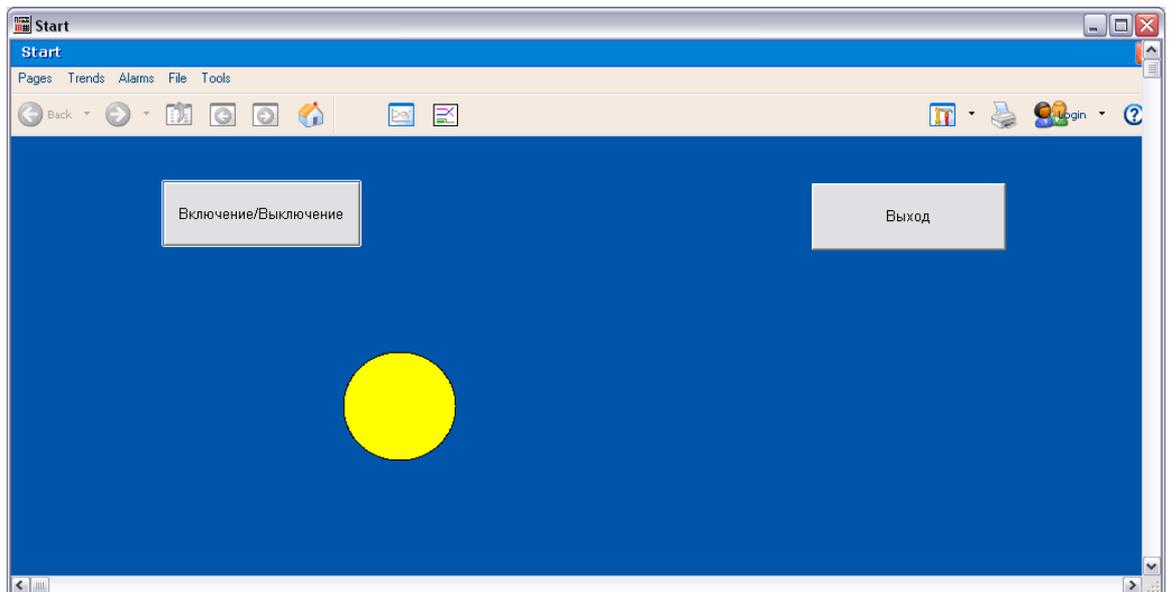


Сохраните страницу задав ее имя.

5. Запустите приложение *Run Project*. Результат выполнения:



После нажатия на кнопку Включить/Выключить объект начинает движение (если он двигался, то останавливается):



После нажатия на кнопку Выход приложение завершает работу.

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.

Ход работы:

Для создания интерфейса оператора необходимо использовать виртуальное внешнее устройство (Generic, либо OPC).

Для создания проекта в Citect используются три программных компонента:

- Citect Explorer – создание страниц проекта, выбор компонент системы (устройства, переменные, (теги), сервера, платы ввода вывода). Основное средство управления проектом.
- Citect Builder – для просмотра и создания всех элементов системы, а также ошибок компиляции.
- Citect Runtime – система запуска приложения, разработанного в SCADA и ее проверки функционирования в режиме реального времени и режиме эмуляции.

При использовании нетривиальных функций управления, осуществляется программирование проекта. Программирование функций выполняется на встроенном языке – Cicode, вызов редактора осуществляется из Citect Explorer.

Алгоритм создания проекта:

- 1) Создание проекта, либо создания ссылки на проект. При создании в окне Citect Explorer для заданного проекта формируется шаблон, включающий средство создания страниц, а также всех компонент системы. Компоненты системы – теги переменных, устройства, сервера ввода вывода, системные компоненты.
- 2) Создается сервер ввода вывода. При выборе данной функции необходимо указать имя.
- 3) Создается устройство ввода вывода, с которым будет взаимодействовать ПК. Для устройства необходимо указать уникальный идентификатор, адрес (от 1 и выше), тип устройства (Generic), тип памяти, из которой будут приниматься данные (MEMORY).
- 4) Для всех датчиков (устройства передающие сигналы и управляемые пользователем) создаются теги (переменные). При создании Variable Tag важно указать его имя и тип формируемых данных. Имя – уникальный в системе идентификатор, тип – *INT*, *DECIMAL* и пр. Также необходимо для всех элементов указать адрес устройства. При указании адреса следует именовать их начиная с буквы, именующий тип. При типе *DECIMAL* адрес должен быть *D1...D100*, при *INT* – *I1...I100* и так далее. Адрес состоит из двух частей – буквенное обозначение (первая буква типа данных), число обозначающее адрес.
- 5) Для исполнительных устройств также могут быть созданы теги, если вывод информации осуществляется на внешние устройства, либо производится опрос. Если используется виртуальный объект теги указать необязательно.

- 6) В проекте создаются страницы интерфейса оператора. В проекте может быть одна стартовая страница и несколько вызываемых страниц в процессе функционирования. (Требуется создать 1 – 3 страницы). Создание осуществляется через Citect Explorer.
- 7) Переключение между страницами реализуется через кнопки управления, выбираемые в панели управления. Необходимо реализовать кнопку «Выход».
- 8) Для интерфейса оператора на страницах необходимо разместить все датчики и исполнительные элементы. Добавление осуществляется через панель инструментов.
- 9) Для всех датчиков (управляются пользователем) в окне свойств необходимо перейти к вкладке Access или Input и указать в окне ввода текста реакцию на действие датчика (например, установить значения тега в заданное значение).
- 10) Для исполнительных элементов реакция указывается на одном из окон (Appearance, Movement, Scaling, Fill, Slider). Реакцией может быть изменение масштаба объекта, изменение цвета заливки, изменение кадра, движение по заданным величинам, задается степень видимости объекта. В окне ввода текста указывается условие, при котором заданное действие наступает.
- 11) Каждый элемент также может иметь текстовое описание при выводе на экран.

Примечания:

- 1) При необходимости в проект могут быть включены элементы из внешних графических редакторов.
- 2) Для каждого тега необходимо указывать устройство ввода вывода.

Требования к содержанию отчета

Отчет по лабораторной работе должен содержать следующие пункты:

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

Контрольные вопросы

- 1)