3 Пишем простой пример

3.1 Блок управления светофором

В этой главе мы создадим небольшую программу-пример. Это - простой блок управления движением на перекрестке, имеющем светофоры для двух пересекающихся направлений движения. Понятно, что светофоры должны иметь два противоположных состояния – красный и зеленый. Чтобы избежать несчастных случаев, мы добавим общепринятые переходные стадии: желтый и желто-красный. Последняя стадия должна быть длиннее предыдущей.

В этом примере мы покажем, как управляемые по времени процессы можно представить средствами языков стандарта МЭК 61131-3, как легко можно комбинировать различные языки в CoDeSys и познакомимся со средствами моделирования в CoDeSys.

Создайте POU

Начинать всегда легко: запустите CoDeSys и выберите "File" "New".

В окне диалога определим первый POU. По умолчанию он получает наименование PLC_PRG. Не изменяйте его. Тип этого POU, безусловно, должен быть - программа. Каждый проект должен иметь программу с таким именем. В качестве языка программирования данного POU мы выберем язык Continuous Function Chart (CFC).

Теперь создайте еще три объекта. Воспользуйтесь командой "Project" "Object Add" в системном или в контекстном (нажмите правую кнопку мыши в окне Object Organizer) меню. Создайте: программу на языке Sequential Function Chart (SFC) с именем SEQUENCE, функциональный блок на языке Function Block Diagram (FBD) с именем TRAFFICSIGNAL и еще один аналогичный блок - WAIT, который мы будем описывать на языке Список Инструкции (IL).

Что делает TRAFFICSIGNAL?

В POU TRAFFICSIGNAL мы сопоставим определенные стадии процесса соответствующим цветам. То есть мы удостоверимся, что красный свет зажжен в красной стадии и в желто-красной стадии, желтый свет в желтой и желто-красной стадии и т.д.

Что делает WAIT?

В WAIT мы создадим простой таймер, который на вход получает длину стадии в миллисекундах и на выходе выдает состояние ИСТИНА по истечении заданного периода времени.

Что делает SEQUENCE?

В SEQUENCE все будет объединено так, чтобы нужные огни зажигались в правильное время и на нужный период времени.

Что делает PLC_PRG?

В PLC_PRG вводится входной сигнал включения, разрешающий начало работы светофора и 'цветовые команды' каждой лампы связаны с соответствующими выходами аппаратуры.

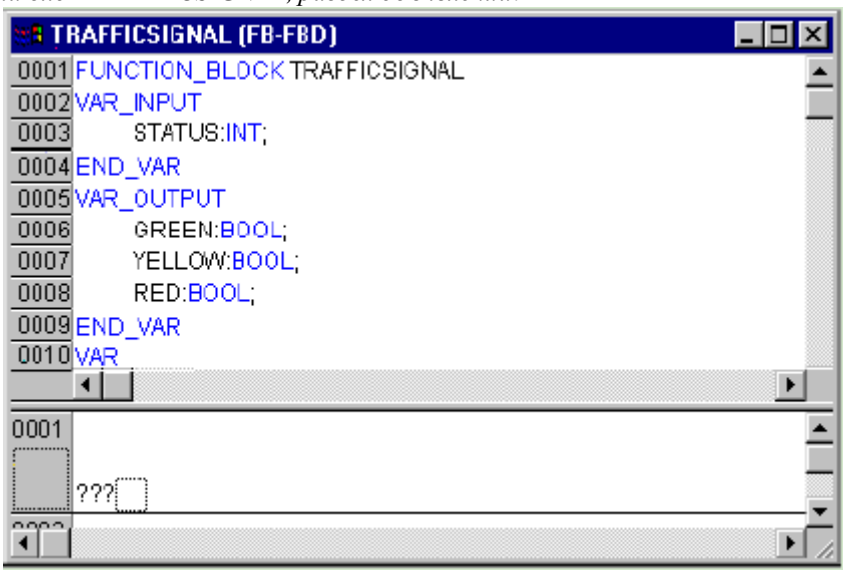
Объявления "TRAFFICSIGNAL"

Вернемся теперь к POU TRAFFICSIGNAL. В редакторе объявлений определите входную переменную (между ключевыми словами VAR_INPUT и END_VAR) по имени STATUS типа INT. STATUS будет

иметь четыре возможных состояния, определяющие соответствующие стадии - зеленая, желтая, желто-красная и красная.

Поскольку наш блок TRAFFICSIGNAL имеет три выхода, нужно определить еще три переменных RED, YELLOW и GREEN. Теперь раздел объявлений блока TRAFFICSIGNAL должен выглядеть так:

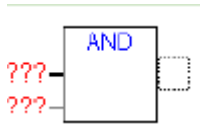
Функциональный блок TRAFFICSIGNAL, раздел объявлений:



Программируем "TRAFFICSIGNAL"

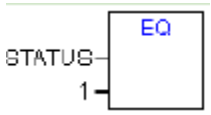
Теперь пора описать, что связывает вход STATUS с выходными переменными. Для этого перейдите в раздел кода POU (body). Щелкните на поле слева от первой цепи (серая область с номером 1). Вы теперь выбрали первую цепь. Теперь дайте команду меню "Insert" "Operator".

В первой цепи будет вставлен прямоугольник с оператором AND и двумя входами:



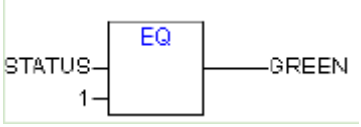
Щелкните мышкой на тексте AND и замените его на EQ.

Три знака вопроса около верхнего из двух входов замените на имя переменной STATUS. Для нижнего входа вместо трех знаков вопроса нужно поставить 1. В результате Вы должны получить следующий элемент:



Щелкните теперь на месте позади прямоугольника EQ. Теперь выбран выход EQ. Выполните команду меню "Insert" "Assignment".

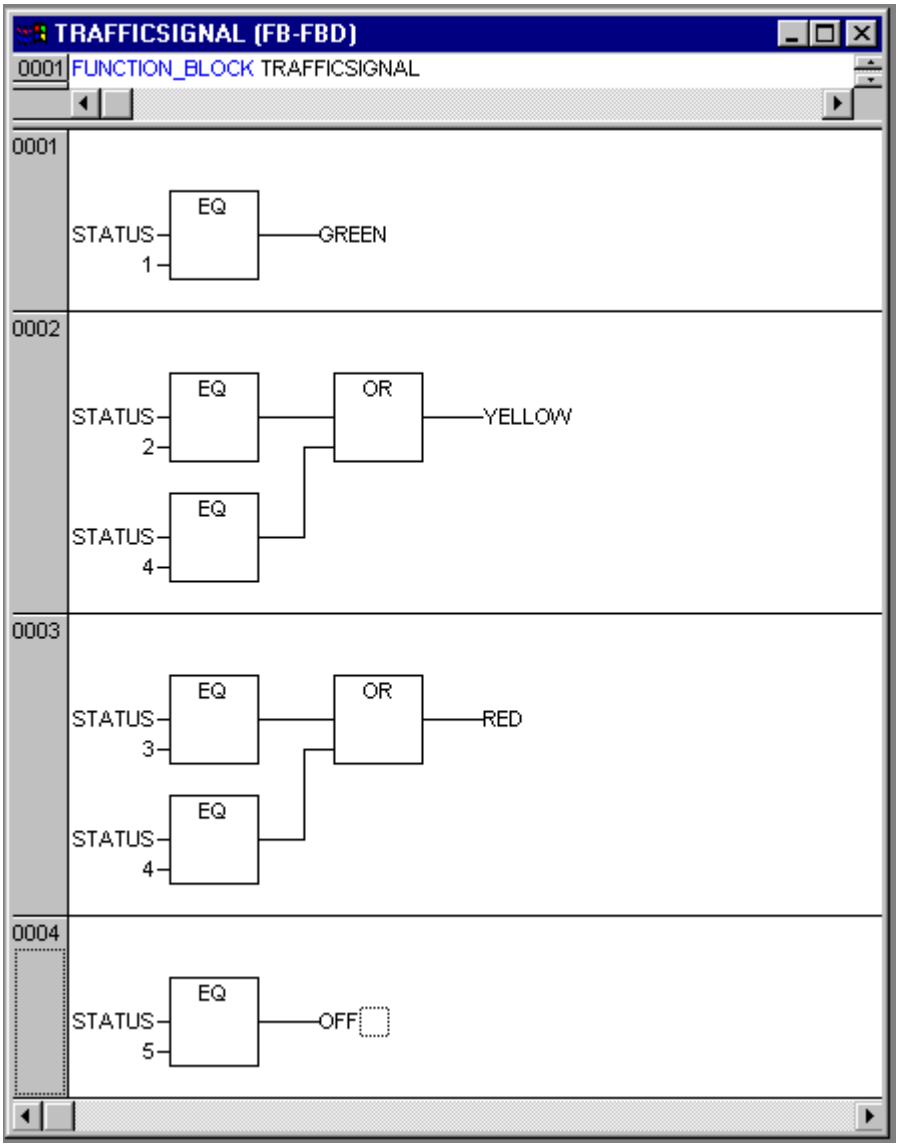
Измените три вопроса ??? на GREEN. Вы теперь создали цепь следующего вида:



STATUS сравнивается с 1, результат присваивается GREEN. Таким образом, GREEN будет включен, когда STATUS равен 1.

Для других цветов TRAFFICSIGNAL нам понадобятся еще две цепи. Создаете их командой "Insert" "Network (after)". Законченный POU должен выглядеть следующим образом:

Функциональный блок TRAFFICSIGNAL:



Чтобы вставить оператор перед входом другого оператора, Вы должны выделить сам вход, а не текст (выделяется прямоугольником). Далее используйте команду "Insert" "Operator".

Теперь наш первый POU закончен. Как и планировалось, TRAFFICSIGNAL будет управлять включением выходов, руководствуясь значением переменной STATUS.

Подключение standard.lib

Для создания таймера в POU WAIT нам понадобится POU из стандартной библиотеки. Итак, откройте менеджер библиотек командами "Window" "Library Manager". Выберете "Insert" "Additional library". Должно открыться диалоговое окно выбора файлов. Выберете standard.lib из списка библиотек.

Объявления "WAIT"

Теперь давайте вернемся к POU WAIT. Как предполагалось, этот POU будет работать таймером, задающим длительность стадий TRAFFICSIGNAL. Наш POU должен иметь входную переменную TIME типа TIME и генерировать на выходе двоичную (Boolean) переменную, которую мы назовем OK. Данная переменная должна принимать значение TRUE, когда желательный период времени закончен.

Предварительно мы устанавливаем эту переменную в FALSE в конце строки объявления (но до точки с запятой, однако) " := FALSE ".

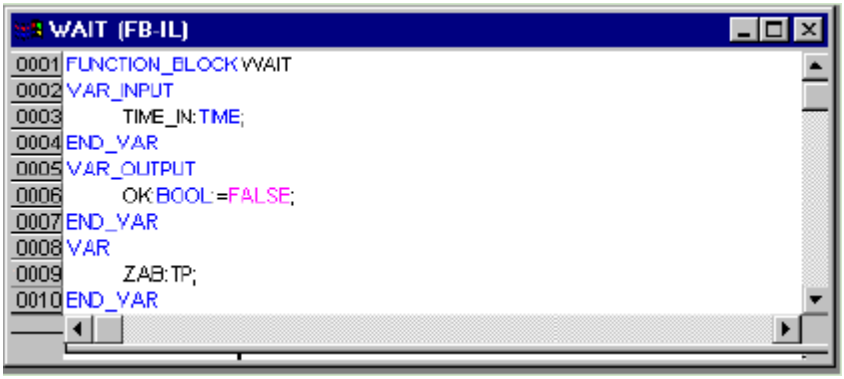
Теперь нам нужен генератор времени POU TP. Он имеет два входа (IN, PT) и два выхода (Q, ET). TP делает следующее:

Пока IN установлен в FALSE, ET будет 0 и Q будет FALSE. Как только IN переключится в TRUE, выход ET начнет отсчитывать время в миллисекундах. Когда ET достигнет значения заданного PT, счет будет остановлен. Тем временем выход Q равен TRUE, пока ET меньше PT. Как только ET достигнет значения PT, выход Q снова переключится в FALSE.

Описание всех POU из стандартной библиотеки приведено в приложении.

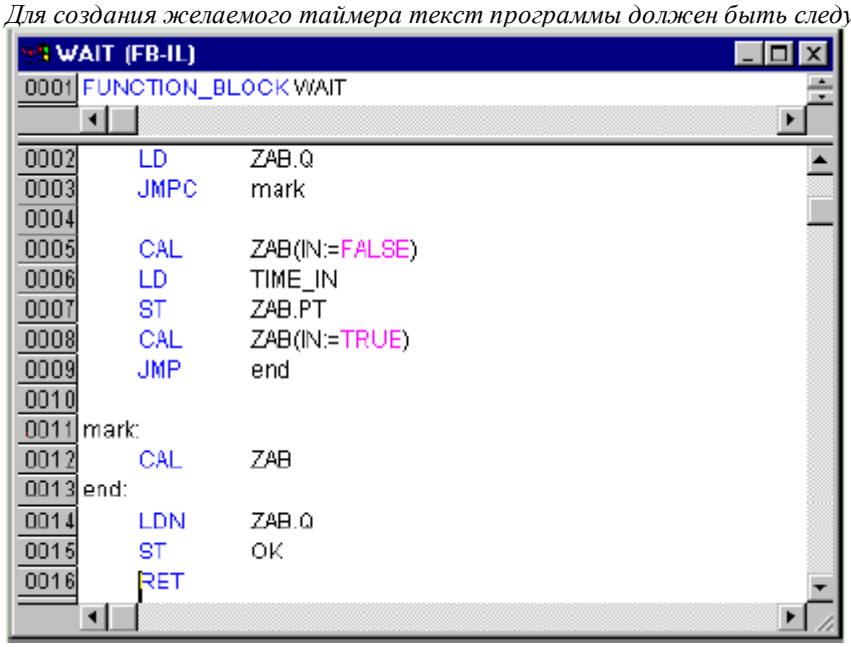
Чтобы использовать TP в POU WAIT, мы должны создать его локальный экземпляр. Для этого мы объявляем локальную переменную ZAB (отсчитанное время) типа TP (между ключевыми словами VAR, END_VAR).

Раздел объявлений WAIT теперь должен выглядеть так:



Программируем "WAIT"

Для создания желаемого таймера текст программы должен быть следующим:



Сначала проверяется, установлен ли Q в TRUE (возможно, отсчет уже запущен), в этом случае мы не трогаем установки ZAB, а вызываем функциональный блок ZAB без входных переменных - чтобы проверить, закончен ли период времени.

Иначе мы устанавливаем переменную IN ZAB в FALSE и одновременно ET в 0 и Q в FALSE. Таким образом, все переменные установлены в начальное состояние. Теперь мы устанавливаем необходимое время TIME переменной PT и вызываем ZAB с IN:=TRUE. Функциональный блок ZAB теперь будет работать, пока не достигает значения TIME и не установит Q в FALSE.

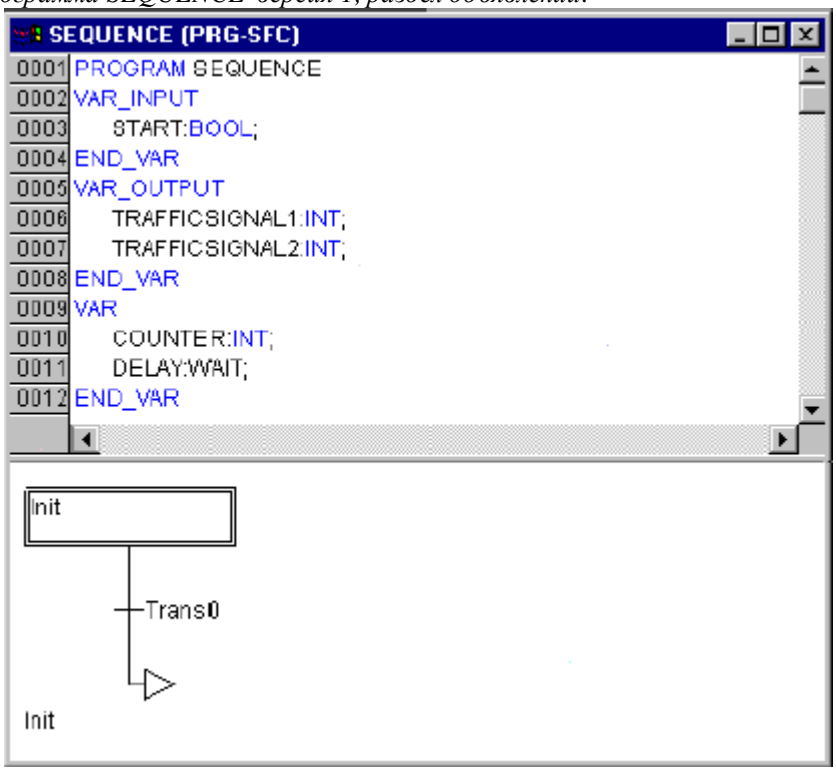
Инвертированное значение Q будет сохраняться в переменной OK после каждого выполнения WAIT. Как только Q станет FALSE, OK примет значение TRUE.

С таймером покончено. Теперь пришло время объединять наши два блока WAIT и TRAFFICSIGNAL в главной программе PLC_PRG.

"SEQUENCE" версия 1

Сначала объявим необходимые переменные. Это входная переменная START типа BOOL, две выходных переменные TRAFFICSIGNAL1 и TRAFFICSIGNAL2 типа INT и одна типа WAIT (DELAY оригинально, не так ли). Программа SEQUENCE теперь выглядит так:

Программа SEQUENCE версия 1, раздел объявлений:



Создаем SFC диаграмму

Первоначально SFC граф всегда состоит из этапа "Init" перехода "Trans0" и возврата назад к Init, естественно, нам придется несколько дополнить его.

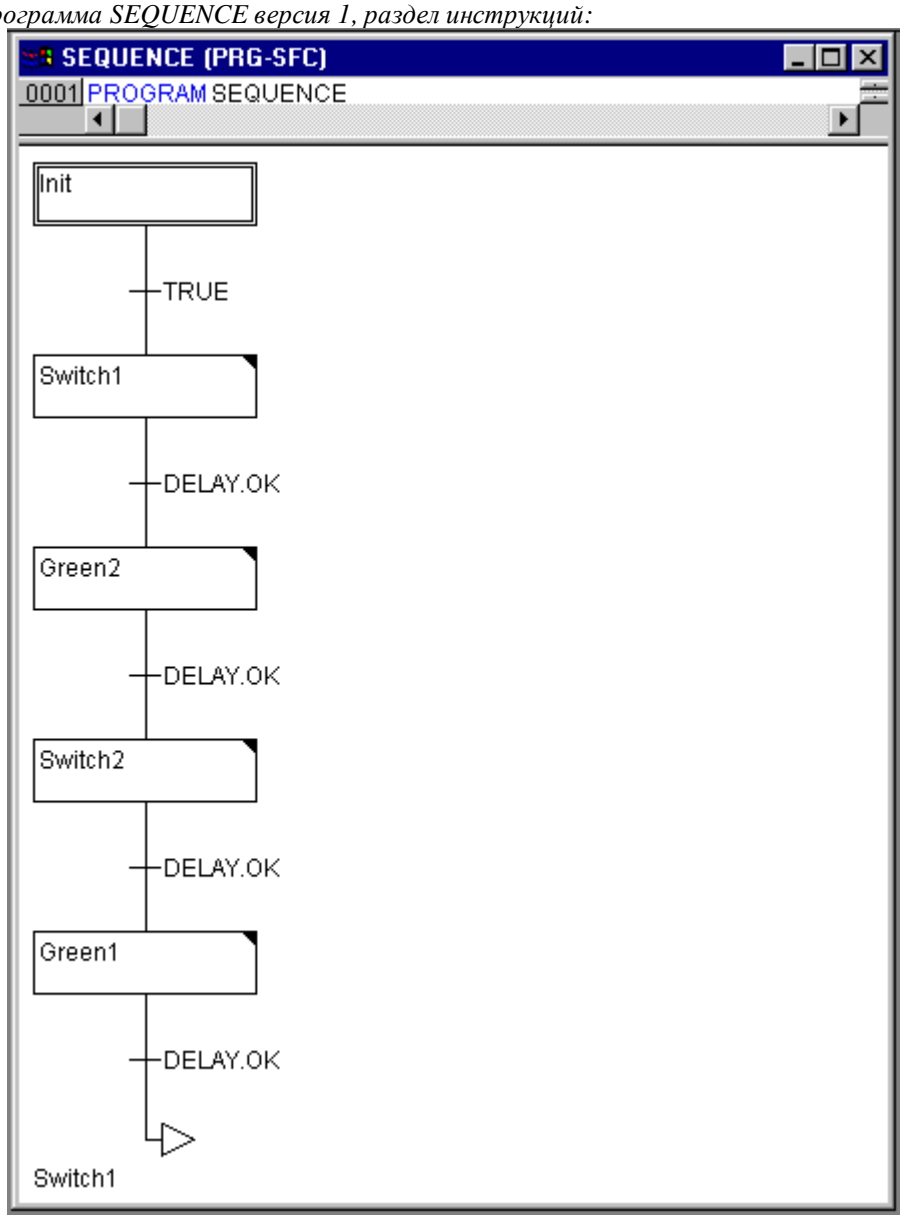
Прежде чем программировать конкретные этапы и переходы, выстроим структуру графа. Сначала нам понадобятся этапы для каждой стадии TRAFFICSIGNAL. Вставьте их, отмечая Trans0 и выбирая команды "Insert" "Step transition (after)". Повторите эту процедуру еще три раза.

Для редактирования названия перехода или этапа нужно просто щелкнуть мышкой на нужном тексте. Назовите первый переход после Init "START", а все прочие переходы "DELAY.OK".

Первый переход разрешается, когда START устанавливается в TRUE, все же прочие - когда DELAY в OK станет TRUE, т.е. когда заданный период закончится.

Этапы (сверху вниз) получают имена Switch1, Green2, Switch2, Green1, ну и Init, конечно, сохранит своё имя. "Switch" должен включать жёлтую фазу, в Green1 TRAFFICSIGNAL1 будет зеленым, в Green2 TRAFFICSIGNAL2 будет зеленым. Наконец, измените адрес возврата Init на Switch1. Если вы всё сделали верно, диаграмма должна выглядеть так:

Программа SEQUENCE версия 1, раздел инструкций:

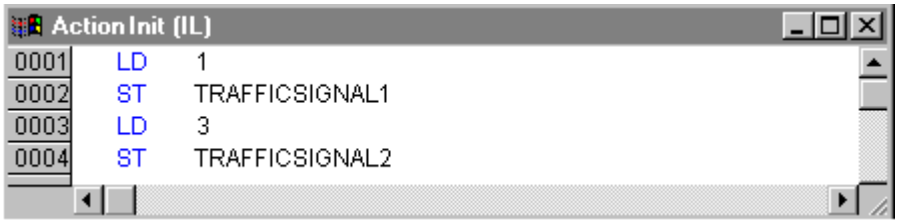


Теперь мы должны запрограммировать этапы. Если вы сделаете двойной щелчок мышью на изображении этапа, то должен открыться диалог определения нового действия. В нашем случае мы будем использовать язык IL (Список Инструкций).

Программирование этапов и переходов

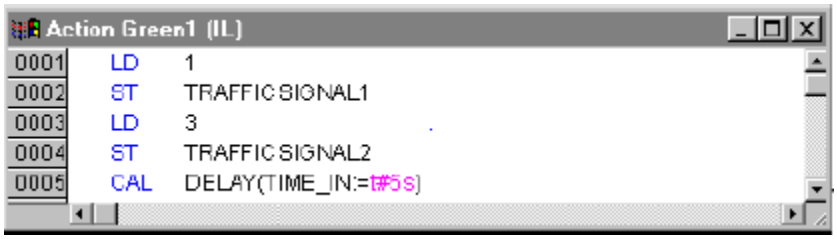
Во время действия этапа Init проверяем, активен сигнал включения START или нет. Если сигнал не активен, то светофор выключается. Этого можно достичь, если записать в переменные TRAFFICSIGNAL1 и TRAFFICSIGNAL2 число 5.

Эman Init:



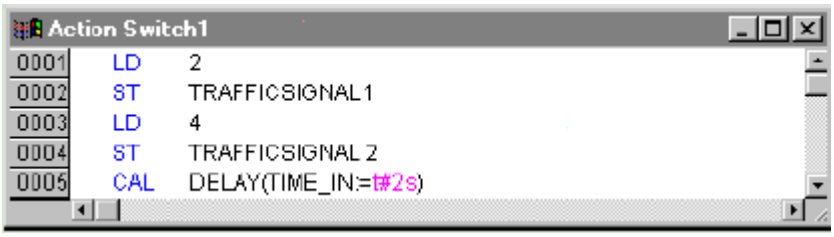
В Green1 TRAFFICSIGNAL1 будет зеленым (STATUS:=1), TRAFFICSIGNAL2 будет красным (STATUS:=3), задержка в 5000 миллисекунд.

Эman Green1:



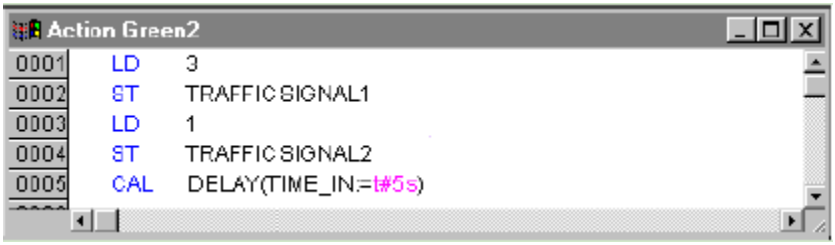
Switch1 изменяет состояние TRAFFICSIGNAL1 на 2 (жёлтое) и, соответственно, TRAFFICSIGNAL2 на 4 (жёлто-красное). Кроме того, теперь устанавливается задержка в 2000 миллисекунд. Это выглядит так:

Эman Switch1:



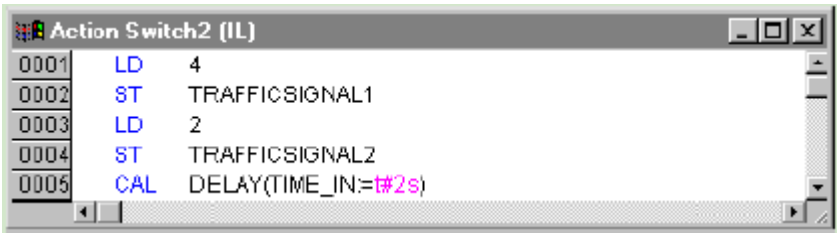
Green2 включает красный в TRAFFICSIGNAL1 (STATUS:=3) и зеленый в TRAFFICSIGNAL2 (STATUS:=1). Задержка устанавливается в 5000 миллисекунд.

Эman Green2:



В Switch2 STATUS в TRAFFICSIGNAL1 изменяется на 4 (жёлто-красный), соответственно, TRAFFICSIGNAL2 будет 2 (жёлтый). Задержка теперь должна быть в 2000 миллисекунд.

Этап Switch2:



Первая версия нашей программы закончена.

Если вы хотите проверить ее работу в режиме эмуляции, сделайте следующее:

Откройте POU PLC_PRG. Каждый проект начинает работу с PLC_PRG. Вставьте в него компонент и замените “AND” на “SEQUENCE”. Входы и выходы оставьте пока свободными.

Теперь вы можете откомпилировать ('Project' 'Build') и её проверить отсутствие ошибок. В окне сообщений вы должны увидеть текст: "0 Errors, 0 Warnings".

Теперь включите флажок 'Online' 'Simulation' и дайте команду 'Online' 'Login'. Запустите программу 'Online' 'Run'.

Откройте программу SEQUENCE. Программа запущена, но не работает, поскольку переменная START должна иметь значение TRUE. Далее это будет делать PLC_PRG, но сейчас вы можете изменить ее вручную. Для этого щелкните дважды мышью по объявлению этой переменной. Ее значение теперь выделено цветом и равно TRUE. Дайте команду записи значений переменных ('Online' 'Write values'). Теперь вы можете понаблюдать за работой программы. Активные шаги диаграммы выделяются голубым цветом.

Для продолжения редактирования программы закройте режим онлайн командой 'Online' 'Logout'.

"SEQUENCE" вторая версия

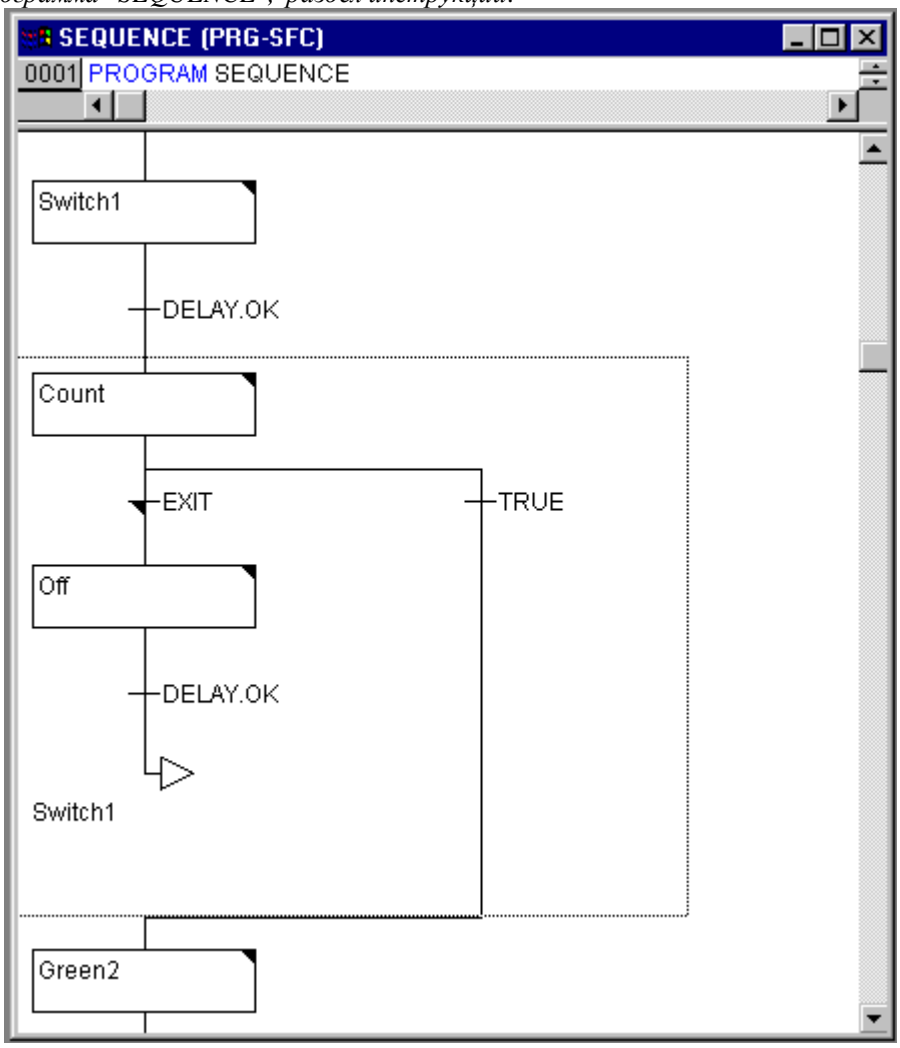
Теперь немного усложним нашу программу. Разумно будет выключать наши светофоры на ночь. Для этого мы создадим в программе счетчик, который после некоторого числа циклов TRAFFICSIGNAL произведет отключение устройства.

Для начала нам нужна новая переменная COUNTER типа INT. Объявите её как обычно в разделе объявлений SEQUENCE.

Теперь выберете переход после Switch1 и вставьте ещё один этап и переход. Выберете результирующий переход и вставьте альтернативную ветвь вправо. После левого перехода вставьте дополнительный этап и переход. После нового результирующего перехода вставьте удаленный переход (jump) на Init.

Назовите новые части так: верхний из двух новых этапов нужно назвать "Count" и нижний "Off". Переходы будут называться (сверху вниз слева на право) EXIT, TRUE и DELAY.OK. Теперь новые части должны выглядеть как фрагмент, выделенный рамкой.

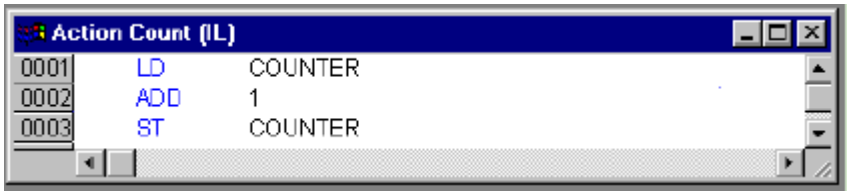
Программа "SEQUENCE", раздел инструкций:



Теперь два новых этапа и перехода необходимо наполнить содержанием.

На этапе Count выполняется только одно действие - COUNTER увеличивается на 1:

Действие Count:

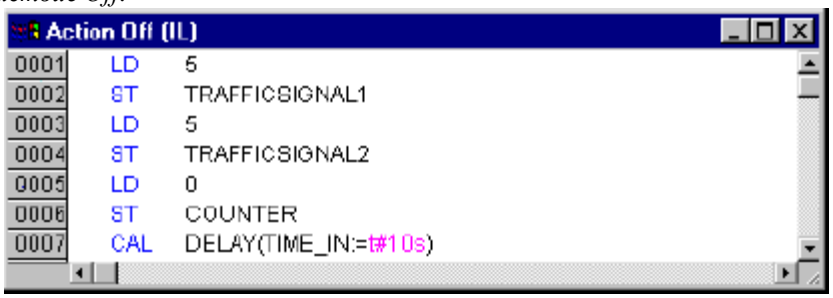


На переходе EXIT1 проверяется достижение счетчиком заданного значения, например 7:

Переход EXIT:

На этапе Off состояние обоих светофоров устанавливается в 5 (светофор выключен), COUNTER сбрасывается в 0 и устанавливается задержка времени в 10 секунд.

Действие Off:



Результат

В нашей гипотетической ситуации ночь наступает после семи циклов TRAFFICSIGNAL. Светофоры полностью выключаются до рассвета, и процесс повторяется снова. При желании вы можете еще раз проверить работу программы в эмуляторе, прежде чем продолжить ее усовершенствование.

PLC_PRG

Мы определили два строго коррелированных во времени светофора в блоке SEQUENCE. Теперь полностью закончим программу. Для этого необходимо распределить входные и выходные переменные в блоке PLC_PRG. Мы хотим дать возможность запустить систему выключателем IN и хотим обеспечить переключение всех шести ламп (2 светофора) путем передачи ”команд переключения” на каждом шаге SEQUENCE. Объявим теперь соответствующие Boolean переменные для всех шести выходов и одного входа, затем создадим программу и сопоставим переменные соответствующим IEC адресам..

Следующий шаг - это объявление переменных LIGHT1 и LIGHT2 типа TRAFFICSIGNAL в редакторе объявлений.

Объявление LIGHT1 и LIGHT2:

Для представления шести ламп светофоров нужно 6 переменных типа Boolean. Однако мы не будем объявлять их в разделе объявлений блока PLC_PRG, вместо этого используем глобальные переменные (Global Variables) из ресурсов (Resources). Двоичная входная переменная IN, необходимая для установки переменной START блока SEQUENCE в TRUE, будет определена таким же образом. Выберете вкладку Resources и откройте список Global Variables.

Объявление глобальных переменных:

```
0001 VAR_GLOBAL
0002     IN:BOOL;
0003     L1_GREEN:BOOL;
0004     L1_YELLOW:BOOL;
0005     L1_RED:BOOL;
0006     L2_GREEN:BOOL;
0007     L2_YELLOW:BOOL;
0008     L2_RED:BOOL;
0009 END_VAR
0010
```

Закончим PLC_PRG. Для этого мы перейдем в окно редактора. Мы выбрали редактор Continuous Function Chart, и, следовательно нам доступна соответствующая панель инструментов.

Щелкните правой клавишей мыши в окне редактора и выберите элемент Box. Щелкните на тексте AND и напишите ”SEQUENCE”. Элемент автоматически преобразуется в SEQUENCE с уже определенными входными и выходными переменными.

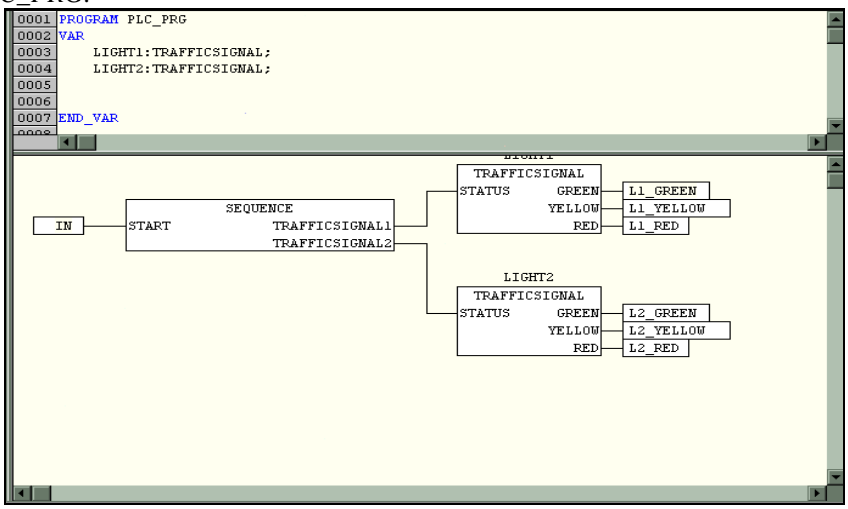
Вставьте далее два элемента и назовите их TRAFFICSIGNAL. TRAFFICSIGNAL - это функциональный блок, и, как обычно, Вы получите три красных знака вопроса, которые нужно заменить уже объявленными локальными переменными LIGHT1 и LIGHT2.

Теперь создайте элемент типа Input, который получит название IN и шесть элементов типа Output, которым нужно дать следующие имена: L1_green, L1_yellow, L1_red, L2_green, L2_yellow, L2_red.

Все элементы программы теперь на месте, и Вы можете соединять входы и выходы. Для этого щелкните мышью на короткой линии входа/выхода и тяните ее (не отпуская клавишу мыши) к входу/выходу нужного элемента.

Наконец Ваша программа должна принять вид, показанный ниже.

PLC_PRG:



Теперь наша программа полностью готова.

TRAFFICSIGNAL эмуляция

Теперь проверьте окончательно вашу программу в режиме эмуляции. Убедитесь в правильности ее работы, контролируя последовательность выполнения и значения переменных в окнах редакторов CoDeSys.

3.2 Визуализация примера

С помощью визуализации можно быстро и легко оживить переменные проекта. Полное описание визуализации Вы найдете в главе 8. Сейчас мы нарисуем два светофора и их выключатель, который позволит нам включать и выключать блок управления светофором.

Создание новой визуализации

Для того чтобы создать визуализацию, выберите вкладку Visualizations в организаторе объектов. Теперь выполните команду 'Project' 'Object Add'.

Диалог для создания новой визуализации:

Введите любое имя для визуализации, например Lights. Когда Вы нажмете кнопку Ok, откроется окно, в котором вы будете создавать визуализацию.

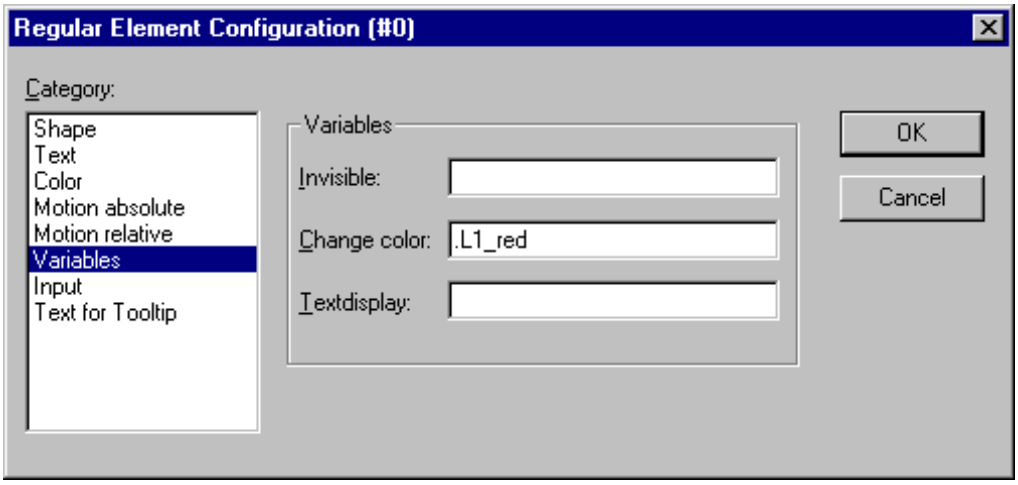
Вставка элемента в визуализацию

Для создания визуализации светофора выполните следующие действия:

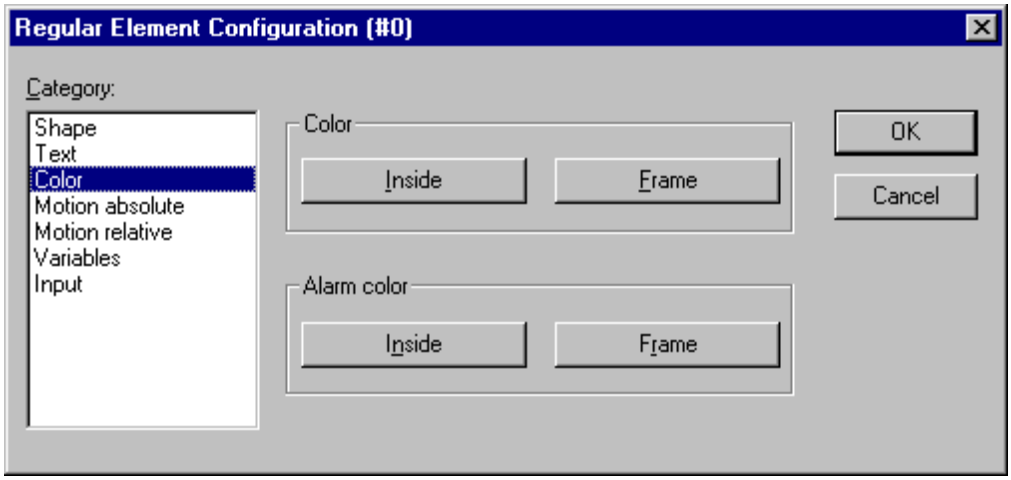
- Выберите команду 'Insert' 'Ellipse' и нарисуйте окружность с диаметром около 2 сантиметров. Для этого щелкните мышью на рабочем поле и, удерживая левую кнопку мыши, растяните появившуюся окружность до требуемого размера.

- Дважды щелкните мышью на окружности. Появится диалоговое окно для настройки элемента визуализации.

- Выберите категорию Variables и в поле Change color введите имя переменной .L1_red. Вводить имя переменной удобно с помощью Input Assistant (клавиша <F2>). Глобальная переменная L1_red будет управлять цветом нарисованной Вами окружности.



- Выберите категорию Color. В области Color нажмите кнопку Inside и в появившемся окне выберите любой нейтральный цвет, например, черный.
- Нажмите кнопку Inside в области Alarm Color и выберите красный цвет.



Полученная окружность будет черной, когда значение переменной ложно, и красной, когда переменная истинна.

Таким образом, мы создали первый фонарь первого светофора.

Остальные цвета светофора.

Теперь вызовите команду копирования 'Edit' 'Copy' (<Ctrl>+<C>) и дважды выполните команду вставки 'Edit' 'Paste'(<Ctrl>+<V>). Вы получите две новых окружности. Перемещать эти окружности можно с помощью мышки. Расположите их так, чтобы они представляли собой вертикальный ряд в левой части окна редактора. Двойной щелчок по окружности приводит к открытию окна для настройки свойств элемента визуализации. В поле Change Color окон настройки свойств соответствующих окружностей введите следующие переменные:

для средней окружности: .L1_yellow
для нижней окружности: .L1_green

В категории Color в области Alarm color установите цвета окружностей (желтый и зеленый).

Корпус светофора.

Теперь вызовите команду “Insert” “Rectangle” и вставьте прямоугольник так, чтобы введенные ранее окружности находились внутри него. Выберите цвет прямоугольника и затем выполните команду “Extras” “ Send to back”, которая переместит его на задний план. После этого окружности снова будут видны.

Активизируйте режим эмуляции, выполнив команду “Online” “Simulation”(режим эмуляции активен, если перед пунктом меню“Online” “Simulation” стоит галочка).

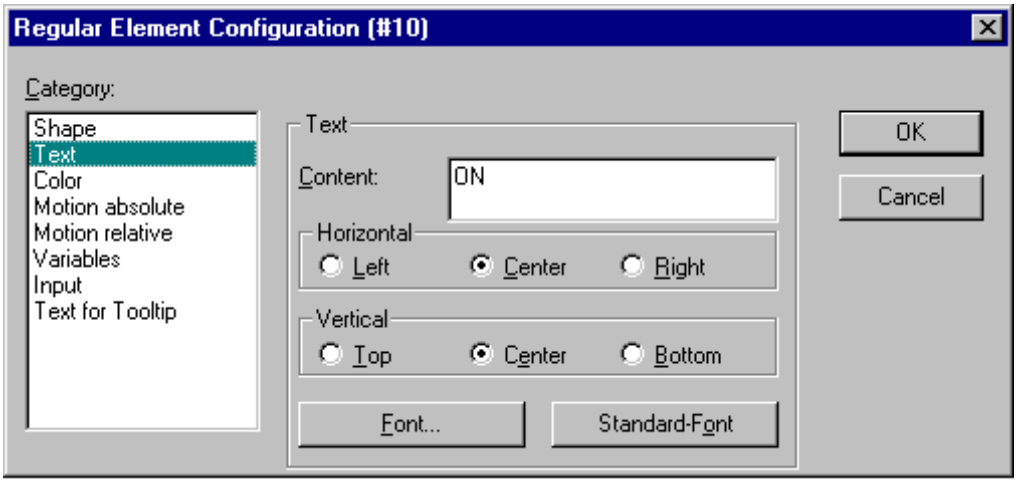
Запустите программу путем выполнения команд “Online” “Login” и “Online” “Run” и вы увидите, как будут меняться цвета светофора.

Второй светофор.

Самый простой способ создать второй светофор – скопировать все элементы первого. Выделите элементы первого светофора и скопируйте их, выполнив команды “Edit” “Copy” и “Edit” “Paste”. Замените имена переменных, управляющих цветами (например, .L1_red на .L2_red), и второй светофор будет готов.

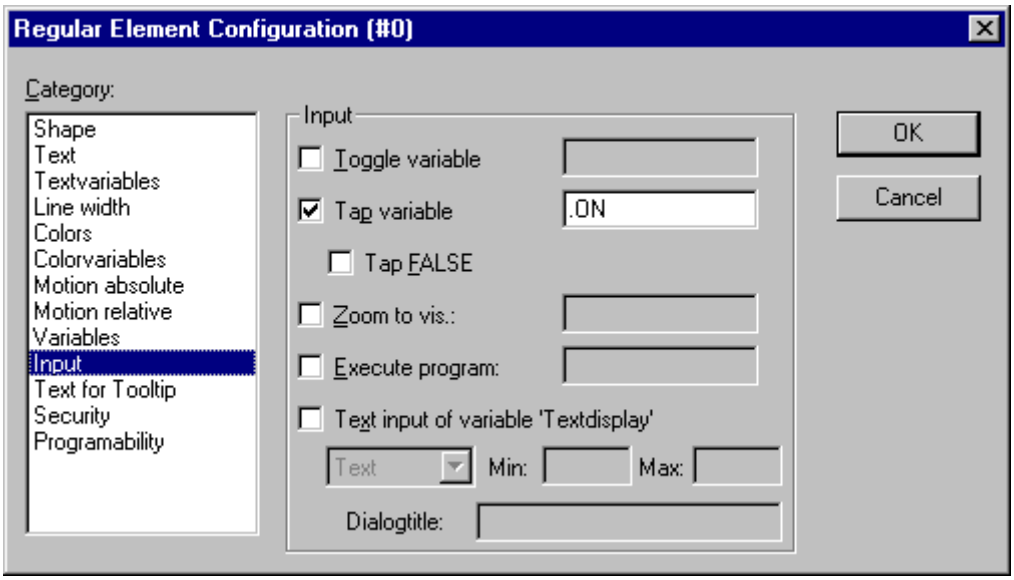
Переключатель ON.

Как описано выше, вставьте прямоугольник, установите его цвет и введите переменную .ON в поле Change Color категории Variables. В поле Content категории Text введите имя “ON”.



Для того чтобы переменная ON переключалась при щелчке мышкой на этом элементе, в поле Toggle variable категории Input введите переменную .ON. Созданный нами переключатель будет включать/выключать светофоры.

Отобразить включенное состояние можно цветом, как и для светофора. Впишите переменную в поле Change Color.



Надписи в визуализации.

Под светофорами вставим два прямоугольника. В свойствах элемента в категории Color цвет границы(frame)прямоугольника задайте белым. В поле Contents(категория Text) введите названия светофоров “Light1” “ Light2”.

Визуализация для проекта Traffic Signal: