

Проект «Инженерные кадры Зауралья»

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Курганский государственный университет»

Кафедра автоматизации производственных процессов

ПРОГРАММИРОВАНИЕ ТАЙМЕРОВ И СЧЕТЧИКОВ В ПРОГРАММНОМ ПАКЕТЕ STEP7

Методические указания
к выполнению лабораторной работы
для студентов направления 220700.62
(очная и заочная форма обучения)

Курган 2015

Кафедра: «Автоматизация производственных процессов»

Дисциплина: «Автоматизация технологических процессов и производств»
(направление 220700.62).

Составил: канд. техн. наук, доцент Н.Б. Сбродов.

Утверждены на заседании кафедры 27 ноября 2014 г.

Рекомендованы методическим советом университета в рамках проекта
«Инженерные кадры Зауралья» 20 декабря 2013 г.

ВВЕДЕНИЕ

Цель лабораторной работы – изучение операций с таймерами и счетчиками, которые являются одними из базовых инструкций в инструментальной системе STEP7, а также - приобретение практических навыков программирования контроллеров SIMATIC S7-300 фирмы Siemens и отладки пользовательских программ [1].

1 ТАЙМЕРНЫЕ ОПЕРАЦИИ

1.1 Общие сведения

Большинство технологических процессов по своей сути разбиты на временные интервалы, чередующиеся события. Для формирования данных временных интервалов и фиксации событий в программах управления ПЛК применяют таймеры.

В SIMATIC Step 7 применяются следующие типы таймеров [1,2]:

- 1) S_PULSE – таймер - формирователь импульса;
- 2) S_PEXT – таймер - формирователь продленного импульса;
- 3) S_ODT – таймер с задержкой включения;
- 4) S_ODTS – таймер с задержкой включения и запоминанием;
- 5) S_OFFDT – таймер с задержкой выключения.

Указанные таймеры в виде функциональных блоков могут входить в состав программы, написанной на языке LAD (язык лестничных диаграмм или контактный план). Имеется второй упрощенный вариант программирования таймеров – в виде битовых инструкций, который рассмотрен ниже.

Набор команд контактного плана поддерживает 256 таймеров для контроллеров SIMATIC S7-300 с процессорным модулем CPU 313C-2DP. Таймеры имеют собственную зарезервированную область памяти в CPU контроллера. Эта область памяти резервирует одно 16-битное слово для каждого адреса таймера. Биты с 0 по 11 в таймерном слове содержат значение времени. Значение времени может быть загружено в двоичном, шестнадцатеричном или двоично-десятичном (BCD) коде. Биты 12 и 13 в таймерном слове содержат базу времени в двоичном коде. База времени определяет интервал, через который значение времени уменьшается на одну единицу.

В данной лабораторной работе используется следующий формат загрузки значения времени:

S5T# aH_bbM_ccS_dddMS.

где a – часы, bb – минуты, cc – секунды, ddd – миллисекунды.

Таймеры имеет следующие параметры:

- 1) no – идентификационный номер таймера;
- 2) S – вход запуска;
- 3) TV – предустановленное значение времени (уставка);

- 4) R – вход сброса;
- 5) Q – состояние таймера (выход);
- 6) BI – оставшееся время (в двоичном коде);
- 7) BCD – оставшееся время (в двоично-десятичном коде).

1.2 Таймер S_PULSE – формирователь импульса

Таймер S_PULSE – формирователь импульса изображен на рисунке 1.

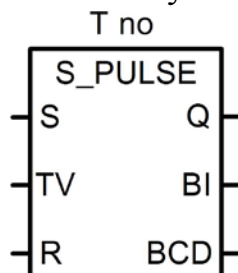


Рисунок 1 – Таймер S_PULSE - формирователь импульса

Командой «Таймер S_PULSE – формирователь импульса» запускается указанный таймер, если имеется положительный фронт (т.е. изменение состояния сигнала из 0 в 1) на входе S (Start [Пуск]). Данное изменение сигнала на входе S всегда необходимо для запуска таймера. Таймер продолжает работать с заданным временем, указанным на входе TV (Time Value [Значение времени]), до тех пор, пока не истечет запрограммированное время, и при условии, что состояние сигнала на входе TV равно 1. Пока таймер работает, состояние сигнала на выходе Q равно 1. Если на входе S происходит изменение с 1 на 0 до истечения заданного времени, таймер останавливается. Тогда состояния сигнала на выходе Q становится равным 0, а значение времени запоминается. При этом при следующем запуске таймера отсчёт будет вестись не от начального значения времени, а от запомненного (если не происходит изменение сигнала с 0 на 1 на входе R). Принцип работы таймера S_PULSE поясняет временная диаграмма, изображенная на рисунке 2.

Изменение с 0 на 1 на входе таймера R (Reset [Сбросить]) во время работы таймера сбрасывает таймер. Это изменение сбрасывает также в ноль время и базу времени. В случае если таймер не работает, а состояние сигнала на входе R становится равно 1, то происходит сброс времени в исходное значение.

1.3 Таймер S_PEXT – формирователь удлиненного импульса

Таймер S_PEXT – формирователь удлиненного импульса изображен на рисунке 3.

Команда «Таймер S_PEXT – формирователь удлиненного импульса» запускает указанный таймер в том случае, если имеется положительный фронт (т.е. изменение состояния сигнала с 0 на 1) на входе S (Start [Пуск]). Таймер продолжает работать с заданным временем, указанным на входе TV (Time

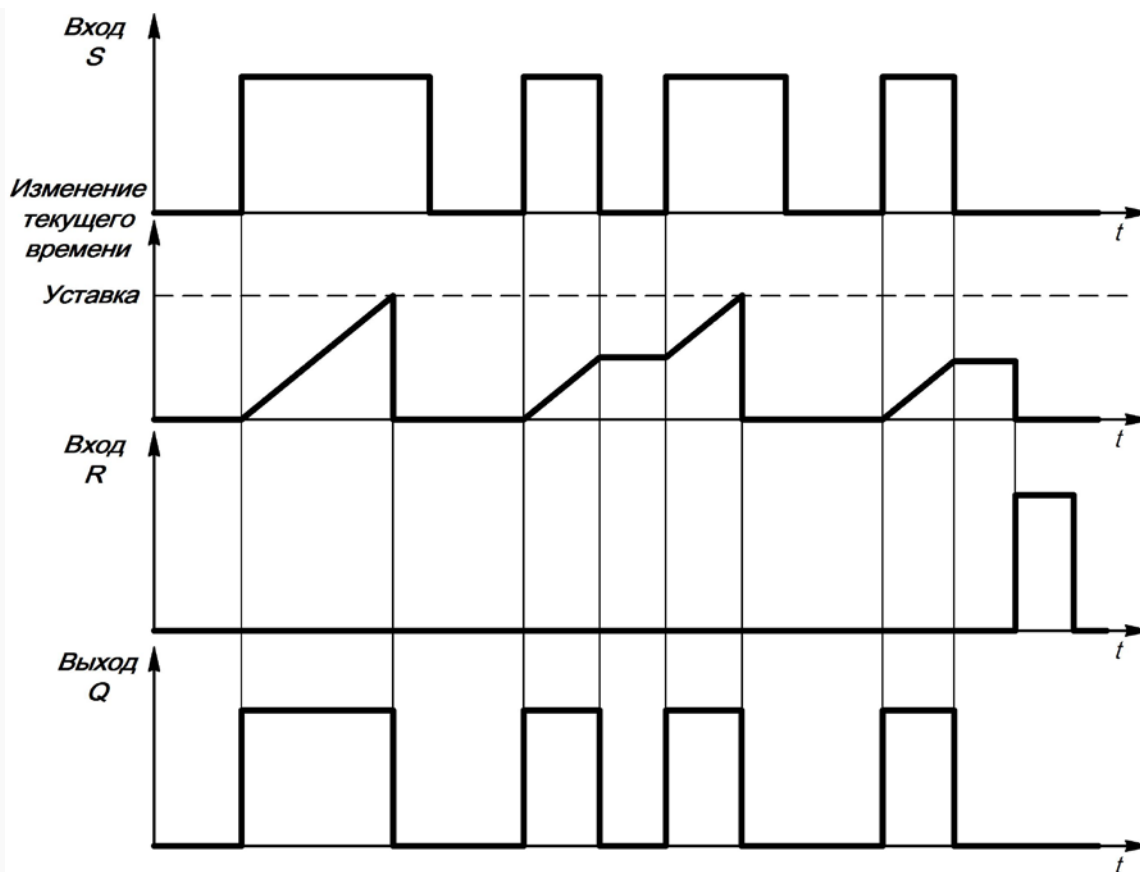


Рисунок 2 – Временная диаграмма работы таймера S_PULSE

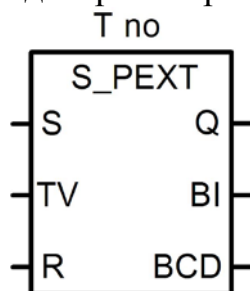


Рисунок 3 – Таймер S_PEXT - формирователь удлиненного импульса

Value [Значение времени]), даже если состояние сигнала на входе S меняется на 0 до истечения времени. Пока таймер работает, состояния сигнала на выходе Q равно 1. Таймер перезапускается с заданным временем, если состояние сигнала на входе S меняется с 0 на 1 во время работы таймера. Изменение с 0 на 1 на входе таймера R (Reset [Сбросить]) во время работы таймера сбрасывает таймер. Это изменение сбрасывает также в ноль время и базу времени. Принцип работы таймера поясняет временная диаграмма, изображенная на рисунке 4.

1.4 Таймер S_ODT – таймер с задержкой включения

Таймер с задержкой включения S_ODT изображен на рисунке 5.

Команда «Таймер с задержкой включения S_ODT» запускает указанный

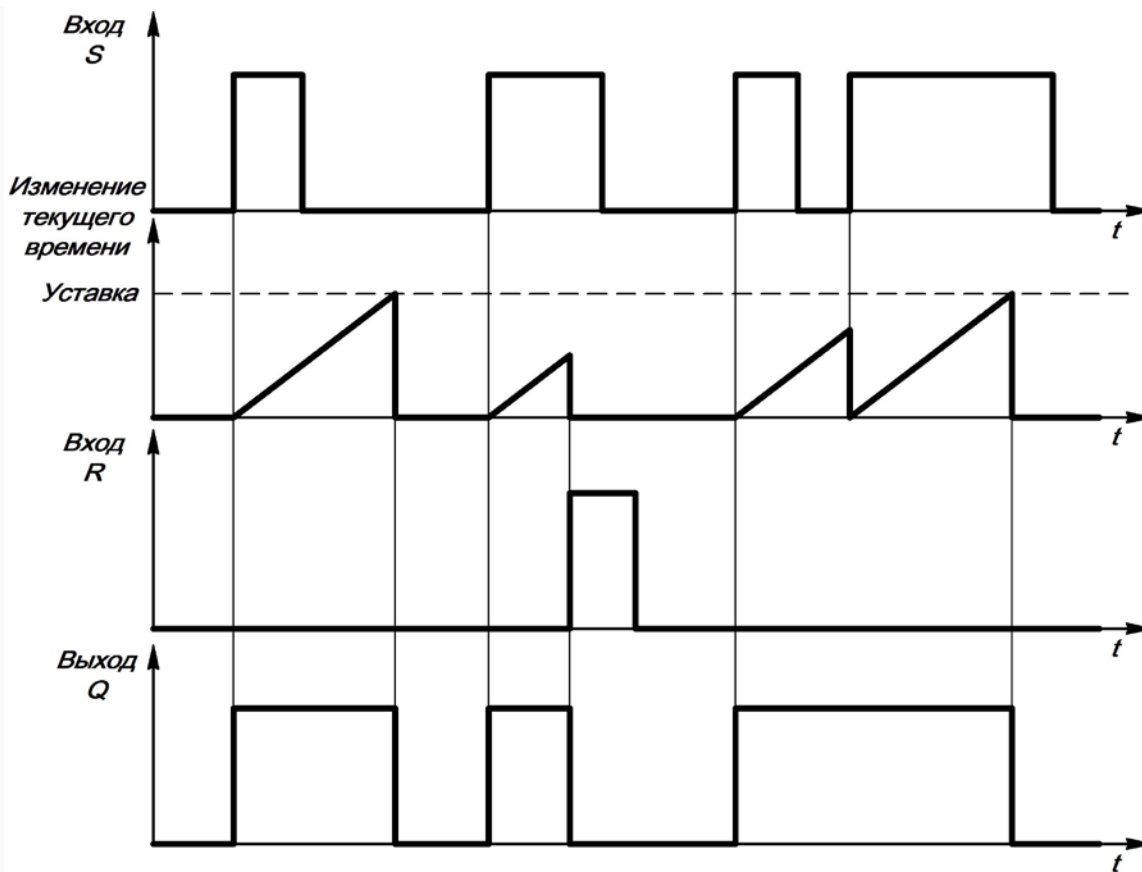


Рисунок 4 – Временная диаграмма работы таймера S_PEXT

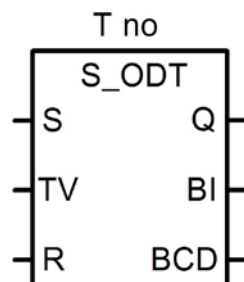


Рисунок 5 – Таймер с задержкой включения S_ODT

таймер, если имеется положительный фронт на входе S. Изменение сигнала всегда необходимо для запуска таймера. Таймер продолжает работать с заданным временем, указанным на входе TV, до тех пор, пока состояние сигнала на входе S равно 1. Состояния сигнала на выходе Q равно 1 тогда, когда время истекло без ошибок, а состояние сигнала на входе S все еще равно 1. Если во время работы таймера состояние сигнала на входе S меняется с 1 на 0, таймер останавливается, и на выходе таймера будет 0. При этом происходит запоминание времени и при следующем запуске таймера, отсчёт будет вестись не от начального значения времени, а от запомненного. Изменение с 0 на 1 на входе таймера R во время работы таймера сбрасывает таймер. Это изменение сбрасывает также в ноль время и базу времени. В случае если таймер не работает, а состояние сигнала на входе R становится равно 1, то происходит

сброс времени в исходное значение.

Принцип работы таймера поясняет временная диаграмма, изображенная на рисунке 6.

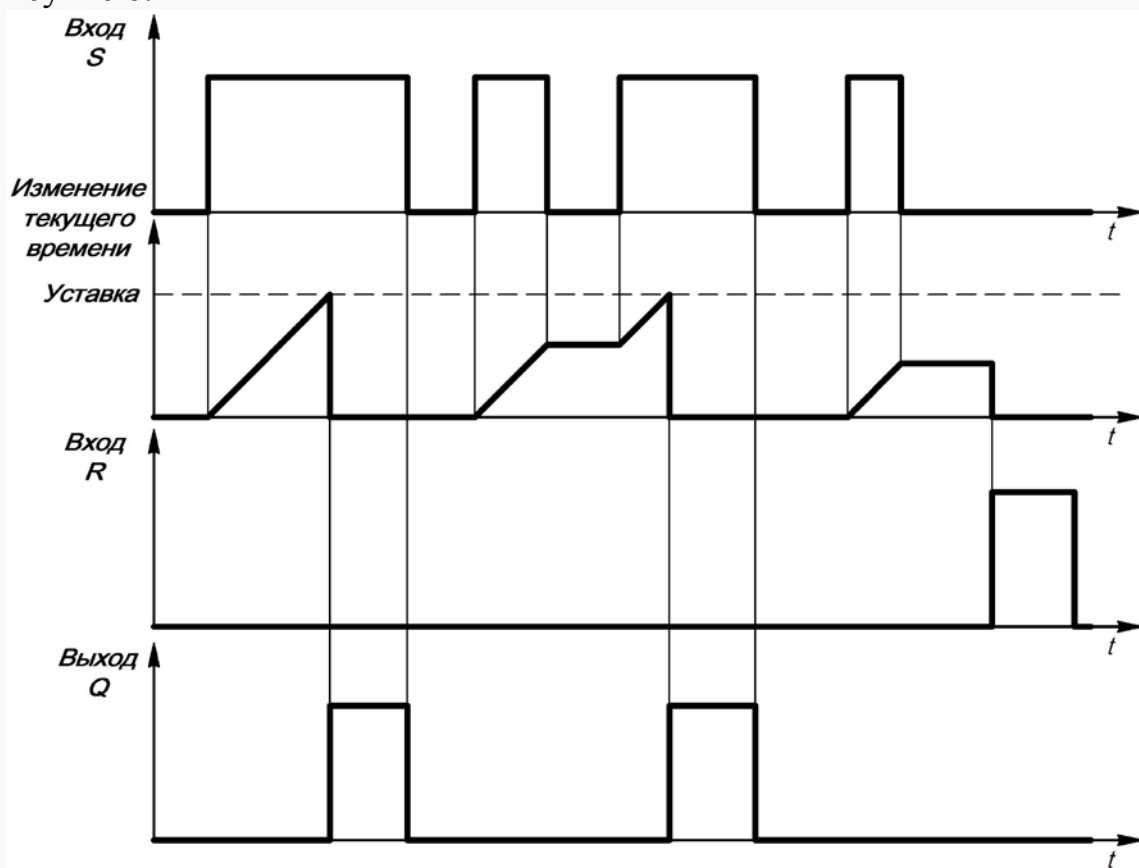


Рисунок 6 – Временная диаграмма работы S_ODT - таймера

1.5 Таймер S_ODTS – таймер с задержкой включения и запоминанием

Таймер S_ODTS – таймер с задержкой включения и запоминанием приведен на рисунке 7.

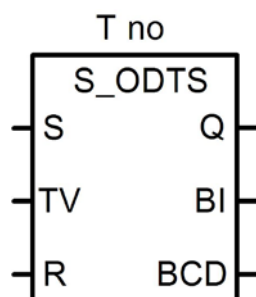


Рисунок 7 – Таймер с задержкой включения и запоминанием S_ODTS

Команда «Таймер S_ODTS – таймер с задержкой включения и запоминанием» запускает указанный таймер, если имеется положительный фронт на входе S. Таймер продолжает работать с заданным временем, указанным на входе TV, даже если состояние сигнала на входе S меняется на 0 до истечения заданного времени. Состояние сигнала на выходе Q равно 1 тогда, когда время истекло независимо от состояния сигнала на входе S, при условии, что вход

сброса (R) остается равным 0. Таймер перезапускается с заданным временем, если состояние сигнала на входе S меняется с 0 на 1 во время работы таймера.

Принцип работы таймера поясняет временная диаграмма, изображенная на рисунке 8.

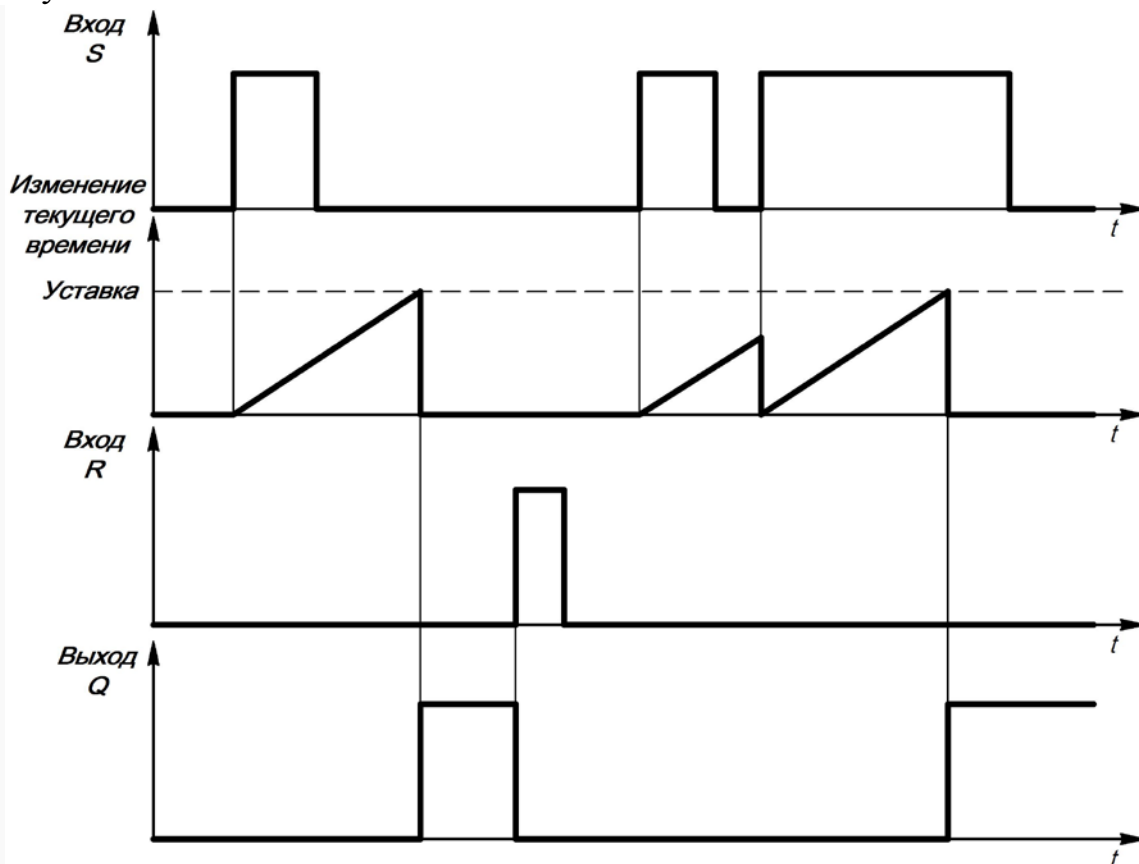


Рисунок 8 – Временная диаграмма работы S_ODTS - таймера

1.6 Таймер S_OFFDT – таймер с задержкой выключения

Таймер с задержкой выключения S_OFFDT изображен на рисунке 9.

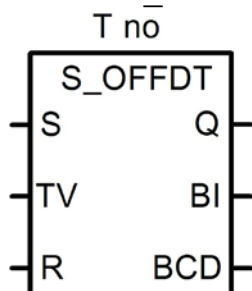


Рисунок 9 – Таймер с задержкой выключения S_OFFDT

Таймер запускается тогда, когда на его вход S поступает отрицательный фронт, то есть изменение сигнала с нуля на единицу. Состояние сигнала на выходе Q равно 1 тогда, когда равно 1 состояние сигнала на входе S или если таймер работает. Если время, заданное на входе TV, истекло, то состояние на выходе Q становится равным 0.

Если состояние сигнала на входе S изменяется с 0 на 1 во время работы

таймера, то значение времени запоминается. Если после этого состояние сигнала на входе S изменится вновь с 1 на 0, то отсчёт времени будет вестись не от начального значения времени, а от запомненного.

Изменение с 0 на 1 на входе таймера R во время работы таймера сбрасывает таймер. Если состояние сигнала на входе R изменяется с 0 на 1 во время, когда таймер не работает, то состояние на выходе Q становится равным 0 и происходит сброс времени.

Принцип работы таймера поясняет временная диаграмма, изображенная на рисунке 10.

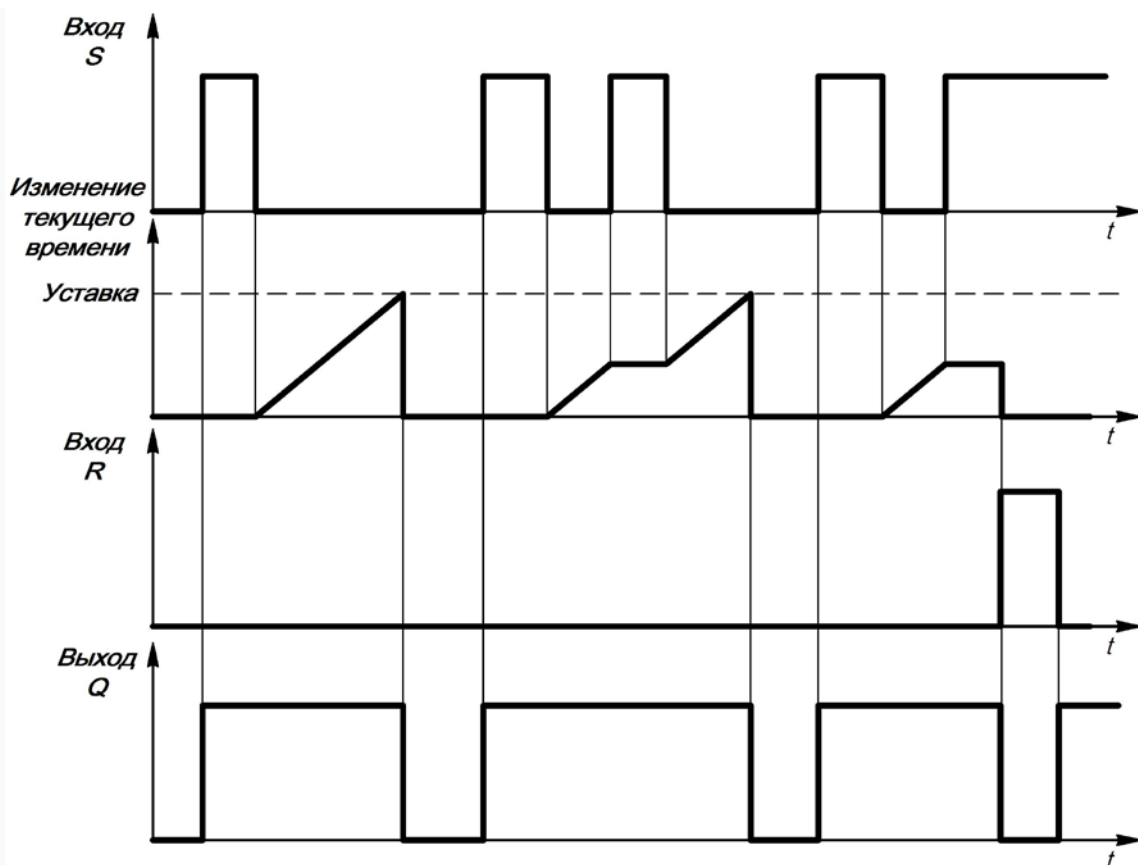


Рисунок 10 – Временная диаграмма работы S_OFFDT – таймера

В качестве примера (рисунок 11) рассмотрим работу таймера с задержкой включения S_ODT.

На вход S таймера поступает сигнал с нормально разомкнутого контакта с адресом I0.1. На вход сброса R поступает сигнал с нормально разомкнутого контакта с адресом I0.2. С выхода таймера сигнал поступает на катушку с адресом Q4.0. После вставки в цепь контактов и катушки необходимо создать таймер и задать его параметры. Для этого в каталоге элементов открываем вкладку «Timers (Таймеры)» и там выбираем: «S_ODT» (таймер с задержкой включения). Чтобы задать время задержки 10с, необходимо на входе TV таймера записать S5T#10S, и подтвердить ввод двойным нажатием клавиши Enter. Если все параметры введены правильно, то все буквы должны быть черного цвета. Неправильно заданные параметры выделяются красным цветом.

Фрагмент LAD программы с таймером «S_ODT» изображен на рисунке 11.

При замыкании контакта I 0.1 на входе S таймер запускается и начинается отсчет времени задержки, которое равно 10с. Спустя 10с состояние выхода Q таймера установится в 1, при условии, что контакт I 0.1 остался замкнутым, а контакт I0.2 не замыкался. Единица на выходе таймера включит катушку Q4.0.

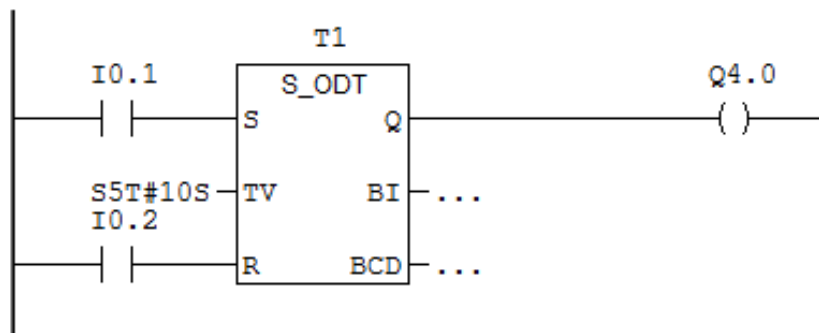


Рисунок 11 – Сегмент LAD программы с таймером S_ODT

1.7 Задание таймеров в виде битовых инструкций

При создании пользовательской программы на языке LAD в программном пакете STEP7 все типы таймеров можно реализовать в виде компактных битовых команд (таблица 1). Запуск и опрос таймеров происходит в разных частях блока. В данном варианте программирования таймеров отсутствуют вход R и выходы BI и BCD.

Таблица 1 – Типы таймеров в виде битовых инструкций

Обозначение таймера	Наименование
---(SP)	Таймер «Формирователь импульса»
---(SE)	Таймер «Формирователь удлиненного импульса»
---(SD)	Таймер с задержкой включения
---(SS)	Таймер с задержкой включения и запоминанием
---(SF)	Таймер с задержкой выключения

В качестве примера (рисунок 12) рассмотрим работу таймера с задержкой включения, заданного в виде битовой инструкции.

В первом сегменте (Network 1) замыкание контакта I0.0 запускает таймер T4 с задержкой включения, равной 5с. При отработке таймером, заданной выдержки времени бит T4, в сегменте Network 2 устанавливается в 1, при условии сохранения замкнутого состояния контакта I0.0. Это обеспечивает включение катушки Q8.0. Замыкание контакта I0.1 сбрасывает таймер T4.

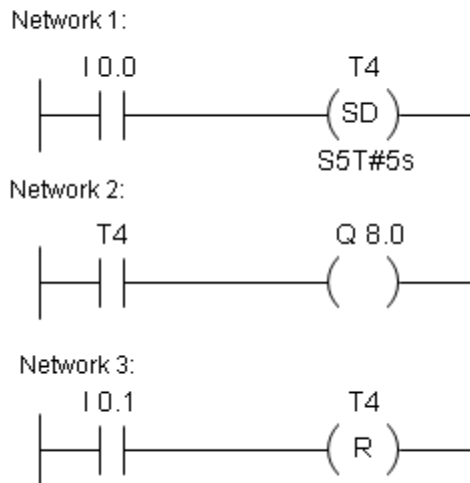


Рисунок 12 – Фрагмент LAD программы с таймером с задержкой включения

2 ОПЕРАЦИИ СО СЧЕТЧИКАМИ

2.1 Общие сведения

При реализации многих технологических процессов необходимо вести подсчет неких событий. Для решения подобных задач в STEP7 применяются различные виды счетчиков: S_CU инкрементный (счетчик прямого счета), S_CD декрементный (счетчик обратного счета) и S_CUD инкрементный/декрементный (счетчик прямого и обратного счета).

Счетчики имеют собственную зарезервированную область памяти в CPU. Эта область памяти резервирует по одному 16-битному слову для каждого счетчика. Набор команд контактного плана поддерживает 256 счетчиков. Операции счета являются единственными функциями, которые имеют доступ к области памяти, зарезервированной для счетчиков. Биты с 0 по 11 в слове счетчика содержат счетное значение в двоичном коде. Диапазон счетных значений лежит между 0 и 999. Для того чтобы запрограммировать счетчик на определенное значение, вводится число от 0 до 999, например, 127, в следующем формате: C#127. Здесь «C#» означает двоично-десятичный формат (формат BCD - каждая группа из 4 битов содержит двоичный код одного десятичного разряда).

Счетчики имеет следующие параметры:

- 1) no – идентификационный номер счётчика;
- 2) CU – вход прямого счёта;
- 3) CV – вход обратного счёта;
- 4) S – вход установки начального значения;
- 5) PV – вход для установки начального значения счётчика;
- 6) R – вход сброса;
- 7) Q – состояние счётчика;
- 8) CV – текущее значение счетчика (двоичный код);
- 9) CV_BCD - текущее значение счетчика (формат BCD).

2.1 Счетчик прямого счета

Счетчик прямого счета (инкрементный счетчик S_CU) изображен на рисунке 13.

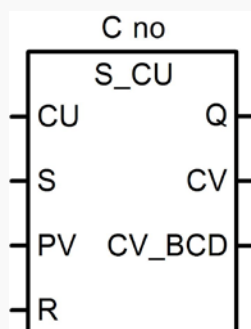


Рисунок 13 – Счетчик прямого счета

Установка счетчика значением на входе PV (Preset Value [Предустановленное значение]) происходит в том случае, если на входе S есть положительный фронт. Положительным фронтом на входе R счетчик сбрасывается. Сброс счетчика делает счетное значение равным 0. При положительном фронте на входе CU значение счетчика увеличивается на 1 при условии, что счетное значение меньше 999. Выход Q равен 1 тогда, когда текущее значение счетчика не равно 0.

2.2 Счетчик обратного счета

Счетчик обратного счета (декрементный счетчик S_CD) изображен на рисунке 14.

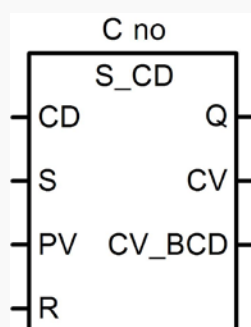


Рисунок 14 – Счетчик обратного счета

Изменение состояния сигнала с 0 на 1 на входе S устанавливает в счетчик значение, записанное на входе PV (Preset Value [Предустановленное значение]). Положительным фронтом на входе R счетчик сбрасывается. Сброс делает счетное значение равным 0. При положительном фронте на входе CD значение счетчика уменьшается на 1 в случае, если текущее значение счетчика больше 0. Сигнал на выходе счетчика равен логической единице тогда, когда значение счетчика не равно нулю. В противном случае сигнал на выходе счетчика будет низкого уровня, т.е. соответствовать логическому нулю.

2.3 Счетчик прямого и обратного счета

Счетчик прямого и обратного счета (инкрементный/декрементный счетчик S_CUD) изображен на рисунке 15.

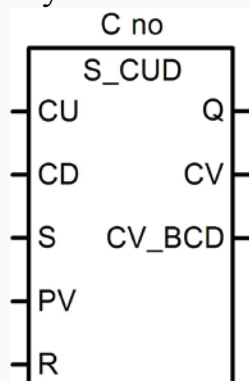


Рисунок 15 – Счетчик прямого и обратного счета

Изменение состояния сигнала с 0 на 1 на входе S устанавливает в счетчик значение, записанное на входе PV (Preset Value [Предустановленное значение]). Положительным фронтом на входе R счетчик сбрасывается. Сброс делает счетное значение равным 0. Счетчик увеличивается на 1, если состояние сигнала на входе CU меняется с 0 на 1 (т.е. имеется положительный фронт), а значение счетчика меньше 999. Счетчик уменьшается на 1, если состояние сигнала на входе CD меняется с 0 на 1 (т.е. имеется положительный фронт), а значение счетчика больше 0. Если положительный фронт имеется на обоих входах, то обе операции выполняются и счетное значение остается тем же самым. Выход Q равен 1 в том случае, если текущее значение счетчик не равно 0.

В качестве примера рассмотрим работу счетчика прямого счета. Для выбора счетчика прямого счета необходимо в каталоге элементов редактора LAD/STL/FBD открыть вкладку «Counter» и там выбрать S_CU. Выбор осуществляется «перетаскиванием» блока на сегмент программы, либо двойным нажатием левой кнопки мыши, но перед выбором счетчика необходимо выделить тот сегмент, куда нужно вставить счетчик, иначе программа создаст еще один сегмент и вставит счетчик туда. К входу прямого счета подключим нормально замкнутый контакт I0.2, а к входу установки начального значения нормально замкнутый контакт I0.3. Это делается нажатием на восклицательные знаки левой кнопкой мыши и последующим выбором необходимого контакта. Установим в счетчик начального значения, равное 7. Для этого нажимаем на вопросительные знаки у входа PV и задаем C#7. После этого подтверждаем ввод двойным нажатием на клавишу Enter. К выходу счетчика подключим катушку Q4.2. Сегмент LAD программы с счетчиком прямого счета изображен на рисунке 16.

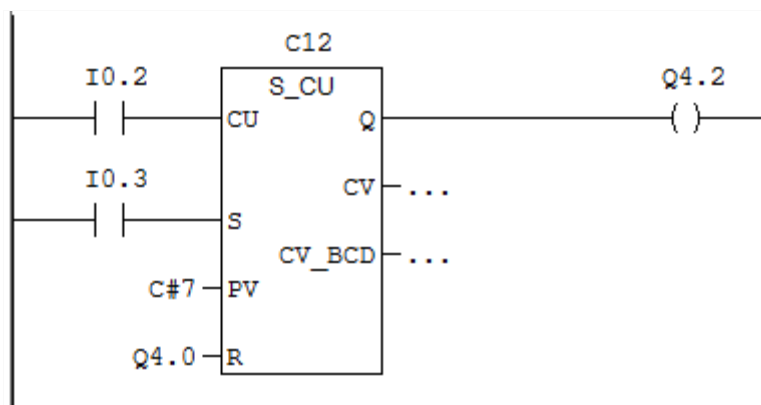


Рисунок 16 – Сегмент LAD программы с S_CU - счетчиком

3 ИНТЕРФЕЙСНЫЙ МОДУЛЬ EasyPort USB И ПУЛЬТ SimuBox

3.1 Общая характеристика

При отладке пользовательских программ наряду с программными средствами STEP7 весьма эффективно применение интерфейсного модуля EasyPort USB в совокупности с пультом SimuBox [3].

Модуль EasyPort (рисунок 17) обеспечивает возможность обмена входными и выходными сигналами между виртуальной средой, созданной на персональном компьютере, и реальным учебным оборудованием.



Рисунок 17 – Интерфейсный модуль EasyPort USB

Через USB-разъём создается соединение EasyPort с ПК. Подключение EasyPort к оборудованию осуществляется при помощи стандартных разъемов

SysLink. Для адаптации EasyPort к различным ситуациям разработана специальная программа EasyVeep, в графическом интерфейсе которой можно создавать различные соединения.

Пульт SimuBox (рисунок 18) служит для индикации выходных дискретных сигналов и имитации входных дискретных сигналов.



Рисунок 18 – Пульт SimuBox

SimuBox с помощью кабеля с разъемами SysLink (рисунок 19) может быть подключен к интерфейсному модулю EasyPort или модулю ввода/вывода любой станции автоматизированной системы MPS.



Рисунок 19 – Кабель с разъемами SysLink

3.2 Программное обеспечение EasyPort

Прежде чем подключать EasyPort к ПК, необходимо убедиться, в том, что нужные драйверы установлены. Для этого нужно открыть программу Festo Didactic EzOPC. Затем во вкладке Overview проверяем столбец Driver. Нужное оборудование должно иметь флаг Installed (рисунок 20).

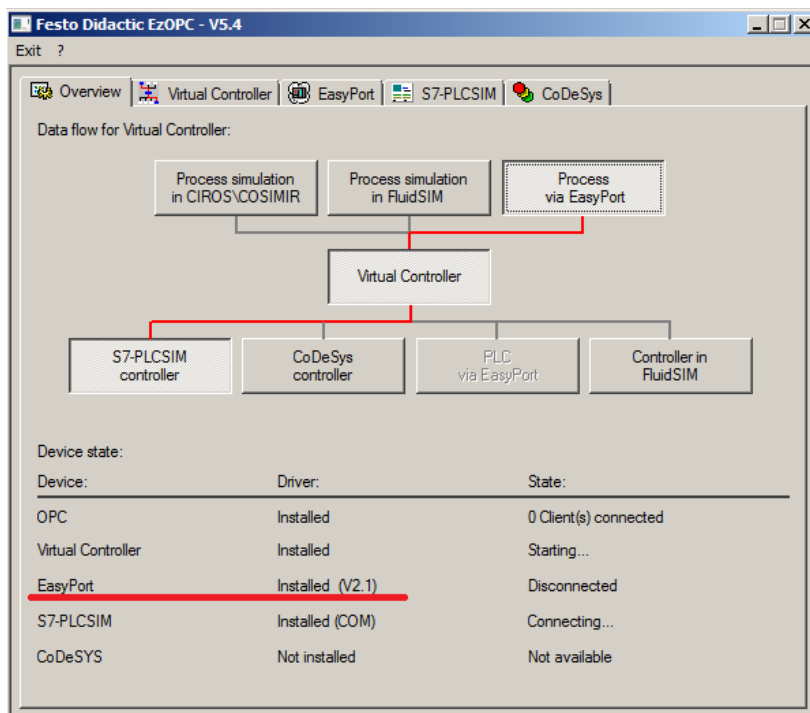


Рисунок 20 – Проверка установки драйверов

Подключаем SimuBox к EasyPort с помощью кабеля с разъемами SysLink. Запускаем программу SIMATIC Manager и открываем в ней проект, который необходимо тестировать (рисунок 21).

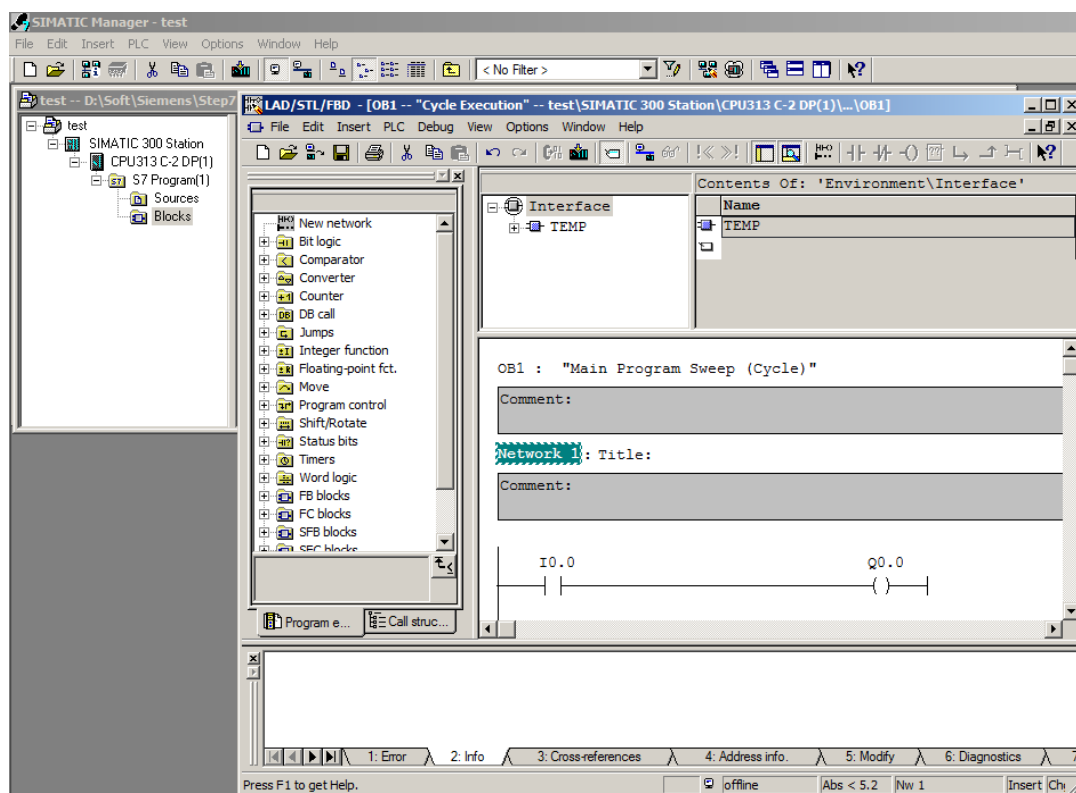


Рисунок 21 – Открытие проекта в SIMATIC Manager

Далее в SIMATIC Manager запускаем виртуальный контроллер, нажимая

кнопку Simulation On и открывая приложение S7-PLCSIM (рисунок 22).

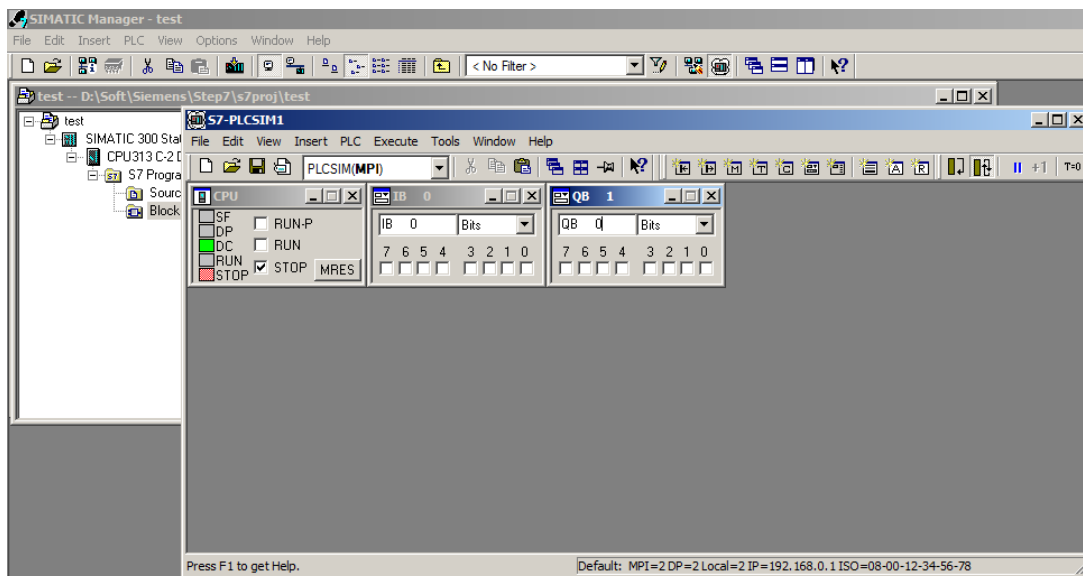


Рисунок 22 – Запуск симулятора S7-PLCSIM

Далее подключаем EasyPort к блоку питания 24В. Синий провод – общий провод, красный провод – плюс 24В. ЖК-дисплей EasyPort должен стать активным и отображать состояние включения.

Следующим шагом будет запуск программы Festo Didactic EzOPC на ПК. В программе выбираем коммутацию Process via EasyPort + Virtual Controller + S7-PLCSIM (рисунок 23). Это позволит с помощью SimuBox управлять входами и выходами виртуального контроллера PLCSIM и наоборот.

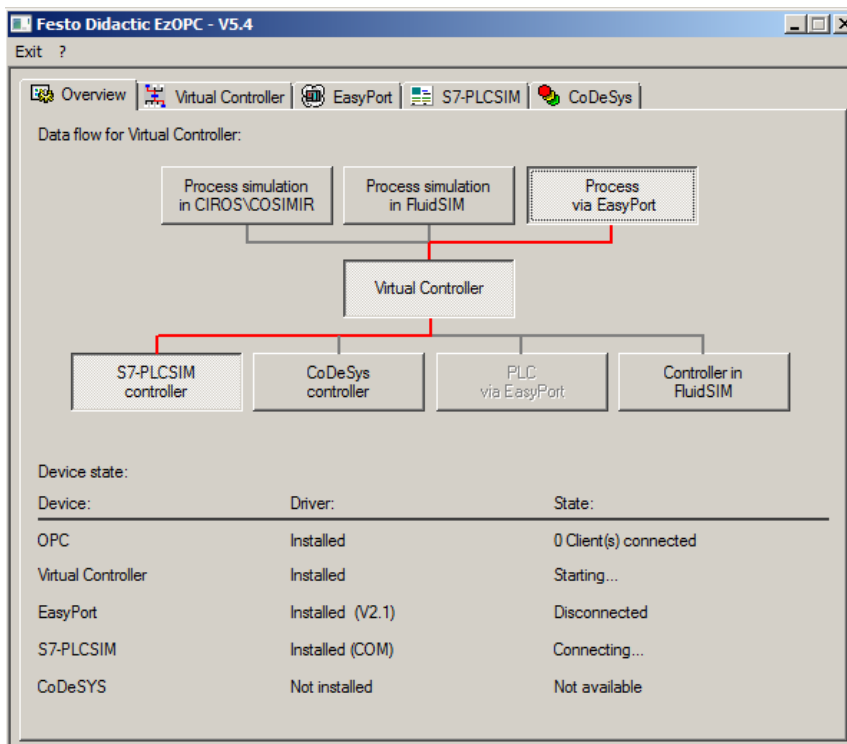


Рисунок 23 – Запуск программы Festo Didactic EzOPC

Подключаем EasyPort к ПК с помощью кабеля USB-RS232. При удачном подключении в программе Festo Didactic EzOPC статус EasyPort должен перейти в состояние Connected. Если этого не произошло, то необходимо обновить состояние программы. Для этого вместо S7-PLCSIM controller устанавливаем CoDeSys controller и обратно на S7-PLCSIM controller. Данное действие обусловлено тем, что программа опрашивает USB-порт лишь в том случае, когда в программу вносятся изменения.

Далее, что бы протестировать программу нужно загрузить ее в виртуальный контроллер. В SIMATIC Manager нажимаем кнопку Download и программа записывается в виртуальный контроллер. После чего в PLCSIM переводим режим контроллера в состояние RUN, установив при этом соответствующую галочку во вкладке CPU (рисунок 24).

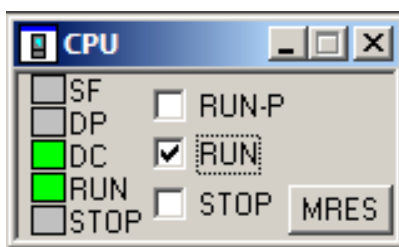


Рисунок 24 – Перевод симулятора PLCSIM в режим RUN

Завершающим шагом будет тестирование программы. В соответствии с заданными переменными и алгоритмом программы, управляем входами в SimuBox и оцениваем результат действий в PLCSIM.

4 ЗАДАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ

В данной лабораторной работе необходимо выполнить исследование работы таймеров и счетчиков в составе LAD программы.

При выполнении лабораторной работы студент должен из таблицы 2 выбрать исходные данные в соответствии с номером варианта, указанным преподавателем.

Таблица 2 – Исходные данные для лабораторной работы

№ варианта	№ временной диаграммы	T1, с	T2, с	T3, с	№ варианта	№ временной диаграммы	T1, с	T2, с	T3, с
1	1	10	10	15	7	1	14	14	13
2	2	12	10	8	8	2	19	10	15
3	1	14	14	11	9	1	17	9	9
4	2	19	17	13	10	2	14	16	12
5	1	10	16	12	11	1	19	11	15
6	2	20	10	20	12	2	17	16	20

Временные диаграммы №1 и №2 представлены на рисунках 25 и 26. На диаграммах показаны изменения входных сигналов SQ1 и BQ1 и временные последовательности формирования выходных сигналов K1 – K6.

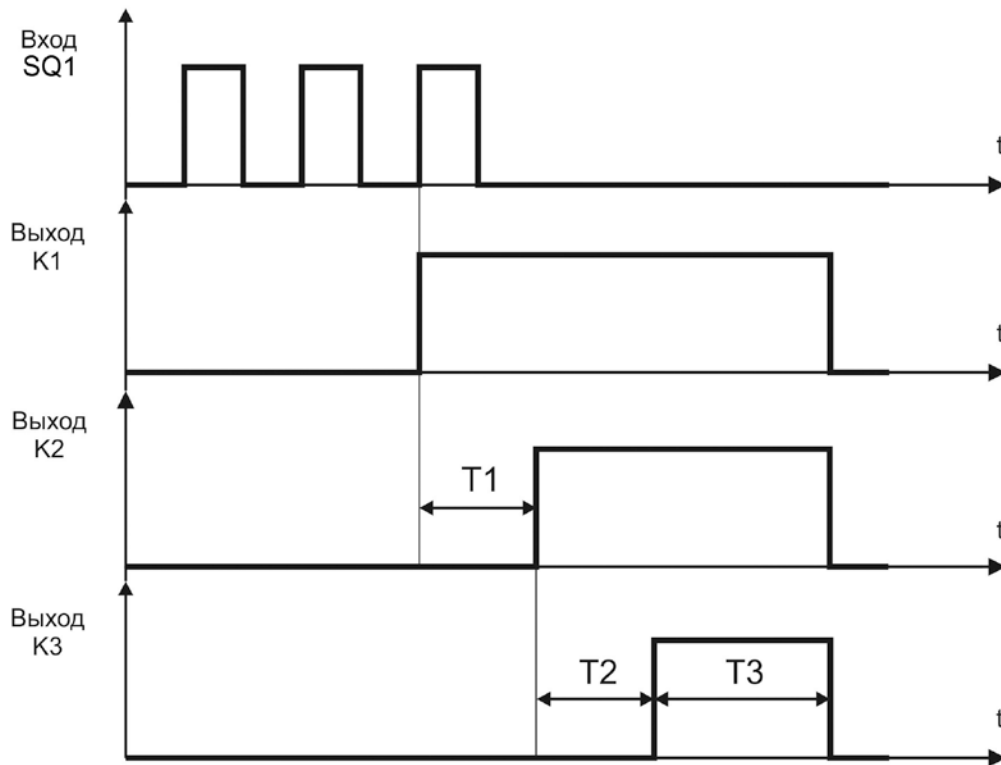


Рисунок 25 – Временная диаграмма №1

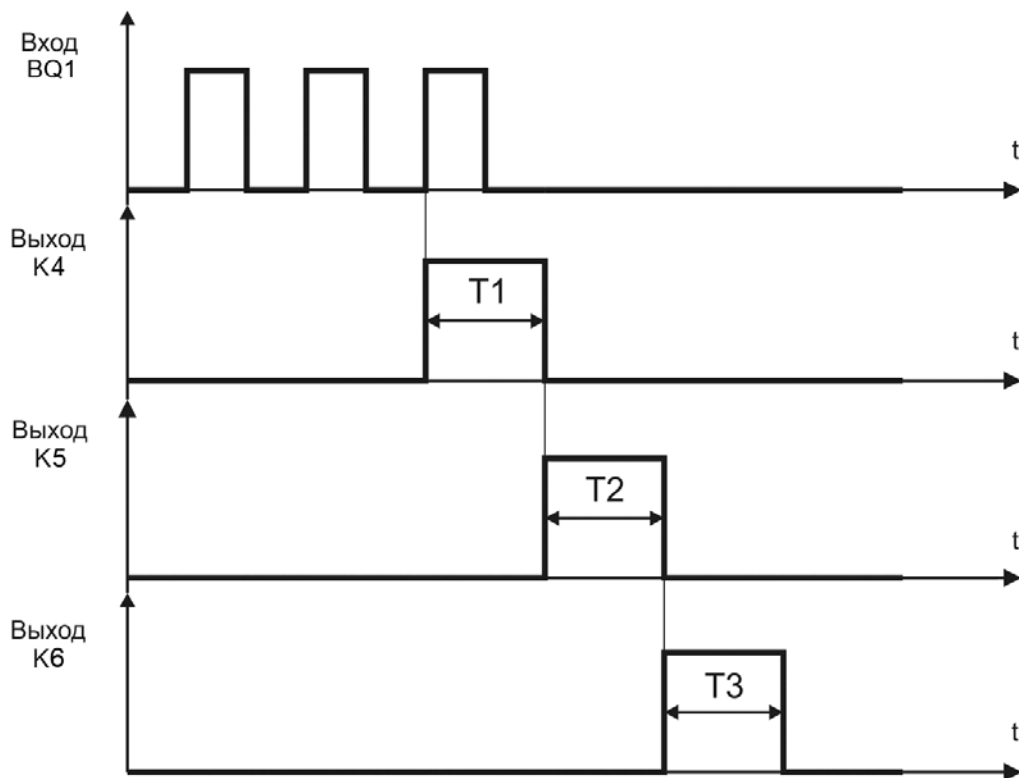


Рисунок 26 – Временная диаграмма №2

5 ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

Лабораторную работу необходимо выполнять в следующей последовательности:

1 Пользуясь данными методическими указаниями, изучить методику задания в STEP7 таймеров и счетчиков (разделы 1, 2).

2 Запустить на компьютере программный пакет STEP7.

3 Отработать в STEP7 все примеры, приведенные в разделах 1 и 2.

4 Получить у преподавателя задание на проектирование пользовательской программы.

5 Используя программный комплекс STEP7, разработать пользовательскую программу на языке LAD.

6 Исследовать разработанную программу, используя симулятор PLCSIM, входящий в состав пакета STEP7.

7 Выполнить подключение интерфейсного модуля EasyPort USB в совокупности с пультом SimuBox к компьютеру и подать напряжение питания на EasyPort.

8 Протестировать разработанную программу, используя аппаратные средства EasyPort и SimuBox. Для этого PLCSIM перевести в состояние RUN. Задавая различные значения входных дискретных сигналов и наблюдая за изменением выходных сигналов, проверить правильность реализации заданного алгоритма.

9 Остановить выполнение программы в STEP7 и отключить питание модуля EasyPort.

10 Оформить отчет по лабораторной работе.

6 СОДЕРЖАНИЕ ОТЧЕТА

В отчете указывается цель лабораторной работы, и приводятся следующие результаты:

1 Задание на разработку пользовательской программы в программном пакете STEP7.

2 Снимок экрана (скриншот) с рабочей областью STEP7, в которой создана LAD программа, реализующая заданный алгоритм.

3 Снимок экрана (скриншот) с рабочей областью симулятора PLCSIM в режиме RUN при тестировании программы.

4 Выводы по результатам работы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 <http://www.siemens.com>

2 <http://www.step7-pro.ru>

3 <http://www.festo.com>

Сбродов Николай Борисович

**ПРОГРАММИРОВАНИЕ ТАЙМЕРОВ И СЧЕТЧИКОВ В
ПРОГРАММНОМ ПАКЕТЕ STEP7**

Методические указания
к выполнению лабораторной работы
для студентов направления 220700.62
(очная и заочная форма обучения)

Авторская редакция

Подписано в печать 26.02.15

Формат 60x84 1/16

Бумага 65 г/м²

Печать цифровая

Усл. печ. л. 1,5

Уч.-изд. л. 1,5

Заказ 40

Тираж 25

Не для продажи

РИЦ Курганского государственного университета.

640000, г. Курган, ул. Советская, 63/4.

Курганский государственный университет.