

*МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ*

федеральное государственное бюджетное образовательное учреждение
высшего образования

«Курганский государственный университет»

Кафедра «Экология и безопасность жизнедеятельности»

**СОВРЕМЕННЫЕ МЕТОДЫ
АВТОМАТИЗИРОВАННЫХ ИЗМЕРЕНИЙ
УРОВНЯ БЕЗОПАСНОСТИ**

Методические указания

к комплексу лабораторных работ для студентов направления подготовки
20.04.01 «Техносферная безопасность» направленность
(«Безопасность жизнедеятельности в техносфере»)

Курган 2019

Кафедра: «Экология и безопасность жизнедеятельности».

Дисциплина: «Современные методы автоматизированных измерений уровня безопасности» (направление 20.04.01)

Составил: канд. тех. наук, доц. Е. К. Карпов.

Утверждены на заседании кафедры

«26» июня 2019 г.

Рекомендованы методическим советом университета

«14» марта 2019 г.

ВВЕДЕНИЕ

Целью выполнения лабораторных работ по дисциплине «Современные методы автоматизированных измерений уровня безопасности» является приобретение студентами знаний о современных методах автоматизированных измерений уровня безопасности.

К задачам, выполняемым в рамках прохождения курса лабораторных занятий, относятся: ознакомление с областью применения методов автоматизированных измерений уровня безопасности, классификацией и характеристиками мехатронных устройств, изучение принципов действия элементов исполнительной, управляющей и информационной подсистем мехатронных устройств, обеспечивающих защиту человека в техносфере.

1 ИЗУЧЕНИЕ И ТЕСТИРОВАНИЕ ХАРАКТЕРИСТИК ДАТЧИКОВ

Существует огромное множество видов датчиков, применяемых в автоматизированных измерениях уровня безопасности. В ходе выполнения данной лабораторной работы будут рассмотрены принципы работы некоторых из них, а также способы взаимодействия и обработки информации датчиков при помощи микроконтроллера.

Задания:

- 1) соберите схемы и выполните программы этой главы и проверьте их работу;
- 2) получите несколько датчиков у преподавателя и подключите их к микроконтроллеру. Считайте с них информацию аналогично считыванию информации с фоторезистора и выведите её на компьютер по последовательному порту.
- 3) дополните программу возможностью вывода предупреждений на компьютер при превышении определённых значений, считанных с датчиков.

Для того чтобы написать программу и записать её на микроконтроллер, необходимо выполнить следующую последовательность действий:

- 1) включить на компьютере исполнительный файл `arduino.exe` – это среда предназначена для разработки программного обеспечения и взаимодействия с контроллером посредством последовательного соединения;
- 2) выбрать в выпадающем меню “Файл”->“Новый” или нажать комбинацию клавиш `Ctrl+N` для создания нового проекта;
- 3) написать программный код, который предполагается исполнять на микроконтроллере, одного из заданий или разработанный самостоятельно;
- 4) проверить и скомпилировать программу, нажав на кнопку “Проверить” или использовав комбинацию клавиш `Ctrl+R`. Во время первой проверки программа предложит сохранить файл программы. Если проверка прошла успешно, то можно переходить к следующему пункту действий, в противном случае необходимо проверить синтаксис и правильность написания логики программы ещё раз;
- 5) прежде чем загружать программу в микроконтроллер, необходимо выбрать порт его подключения к компьютеру, тип платы и процессора, если это

необходимо. Все эти операции производятся в подпунктах выпадающего меню “Инструменты” на верхней панели программы (в нашем случае это будет Arduino Uno без выбора типа процессора);

б) если все предыдущие действия были выполнены без ошибок, то можно произвести загрузку программы на контроллер, нажав кнопку “Загрузка” или используя комбинацию клавиш Ctrl+U.

Для установления связи между компьютером и микроконтроллером посредством последовательного соединения необходимо выполнить следующую последовательность действий:

1) включить на компьютере исполнительный файл `arduino.exe` – это среда предназначена для разработки программного обеспечения и взаимодействия с контроллером посредством последовательного соединения;

2) выбрать порт подключения микроконтроллера к компьютеру, тип платы и процессора, если это необходимо. Все эти операции производятся в подпунктах выпадающего меню “Инструменты” на верхней панели программы (в нашем случае это будет Arduino Uno без выбора типа процессора);

3) в выпадающем меню “Инструменты” выбрать пункт “Монитор порта” или использовать комбинацию клавиш Ctrl+Shift+M;

4) в нижнем поле появившегося окна будет выводиться вся информация, передаваемая от микроконтроллера, чтобы передать информацию на него, необходимо ввести её в верхнее поле окна и нажать клавишу Enter или кнопку “Отправить”. Скорость соединения, включение автоматической прокрутки и выбор символа конца строки можно настроить в процессе соединения.

В случае переподключения платы необходимо закрыть окно “Монитор порта” и открыть его заново.

Электрическая схема с кнопкой и диодом

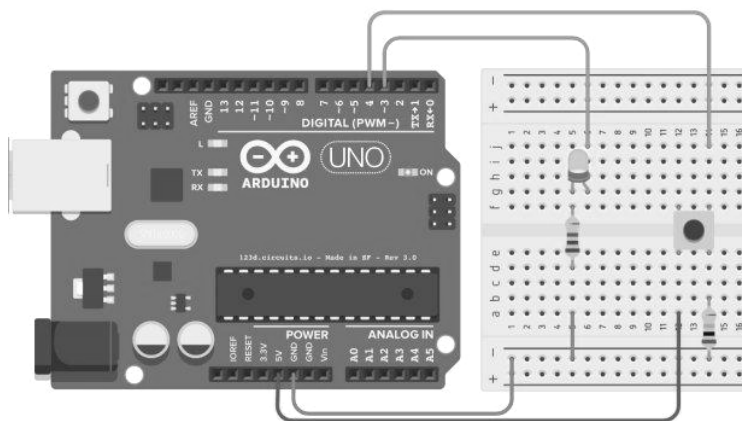


Рисунок 1 – Схема с одной кнопкой и одним диодом

Мигание диодом по нажатию кнопки. Текст программы:

```
void setup() {  
  pinMode(3, OUTPUT); // Третий контакт контроллера – выход (диод)  
  pinMode(4, INPUT); // Четвёртый контакт контроллера – вход (кнопка)  
}
```

```

void loop() {
  int a = digitalRead(4); // Чтение сигнала с кнопки в переменную
  digitalWrite(3,a); // Подача значения переменной на диод
}

```

Электрическая схема с фоторезистором и диодом

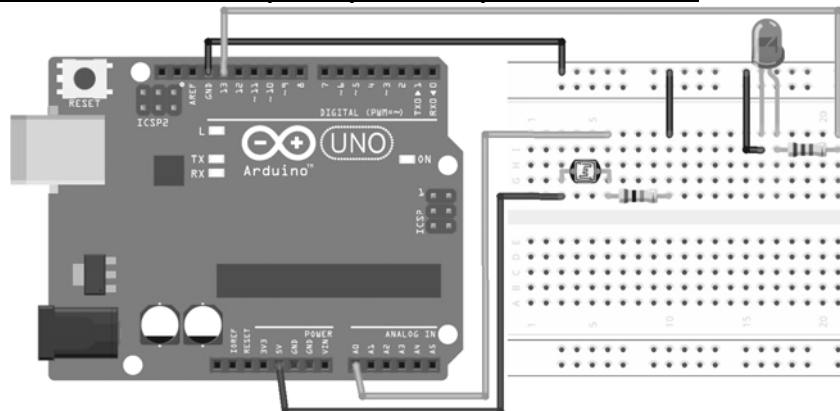


Рисунок 2 – Схема с фоторезистором и одним диодом

Включение диода в темноте и выключение на свету. Текст программы

```

void setup() {
  pinMode(13, OUTPUT); // тринадцатый контакт – выход (диод)
}

void loop() {
  //Если степень освещённости низкая (чтение аналогового порта A0 и
  //сравнение с константой),
  if (analogRead(A0) < 250)
    digitalWrite(13, HIGH); //то включается светодиод

  //Иначе
  else
    digitalWrite(13, LOW); //выключается светодиод
}

```

Обмен информацией между различными устройствами управления в автоматизированной системе является важным фактором для построения распределённых систем и организации их иерархической структуры. В первом случае становится возможна установка отдельных устройств, содержащих датчики или управляющие элементы, на большом расстоянии друг от друга. Во втором случае организуется система с несколькими подчинёнными устройствами, каждое из которых решает локальную задачу, и глобальным устройством управления верхнего уровня, которое задаёт общий вектор управления.

Управление контроллером от компьютера через последовательный порт

```

int val = 0; //создаём переменную для хранения информации с компьютера
int state=LOW; //состояние светодиода (выкл/вкл)

```

```

void setup()
{
  pinMode(13,OUTPUT); //13 ножка - выход(светодиод)
  Serial.begin(9600); //устанавливаем последовательное соединение
}

void loop()
{
  val = Serial.read(); //читаем информацию с компьютера
  if (val == '1') //если считали с компьютера цифру 1, то
  {
    state = !state; //меняем состояние диода на противоположное
                    //(HIGH->LOW или LOW->HIGH)
    Serial.println("Diode switched!"); //выводим сообщение "реверс диода"
  }
  //проверяем статус и включаем или выключаем светодиод
  if (state == HIGH)
    digitalWrite(13,HIGH);
  else
    digitalWrite(13,LOW);
}

```

Подключите к микроконтроллеру датчик температуры и влажности DHT11 и выведите с него информацию на компьютер, используя приведённую программу

Подключение к Ардуино датчика температуры и влажности – DHT11

```

#include "DHT.h" //подключаем библиотеку для датчика
DHT dht(2, DHT11); //определяем, на каком контакте будет датчик
void setup() {
  Serial.begin(9600); //подключаемся к компьютеру
}

void loop() {
  float h = dht.readHumidity(); //считываем температуру
  float t = dht.readTemperature(); //считываем влажность
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.println(" *C");
  delay(2000); //пауза в 2 секунды до нового считывания
}

```

2 УПРАВЛЕНИЕ ПНЕВМАТИЧЕСКИМ И ЭЛЕКТРИЧЕСКИМ ДВИГАТЕЛЯМИ, ПРИМЕНЯЕМЫМИ В СИСТЕМАХ ИЗМЕРЕНИЯ УРОВНЯ БЕЗОПАСНОСТИ

В этой лабораторной работе рассматривается применение промышленных логических контроллеров (ПЛК) для осуществления управления исполнительными элементами с пневматическим приводом. ПЛК и элементы такого типа широко применяются при построении комплексов обеспечения безопасности. В первую очередь они используются в технических системах управления, контроля и обработки информации, получаемой от информационно-измерительных систем.

Пневматические толкатели, оснащённые датчиками положения, применяются в качестве силовых установок для обслуживания различных транспортирующих устройств, а также для автоматизации некоторых производственных процессов.

Лабораторная работа выполняется с целью закрепления теоретических знаний и приобретения навыков программирования и отладки программ для контроллеров OMRON CJ1M.

Лабораторный стенд представляет собой набор пневмооборудования фирмы FESTO, смонтированный на рамной конструкции (рисунок 3). Он включает в себя раму, устройство подготовки воздуха FRC 1/4 D MINI, три пневмоцилиндра DSNU 25 100 P A, три двухпозиционных пневмораспределителя MFH 5 1/4.

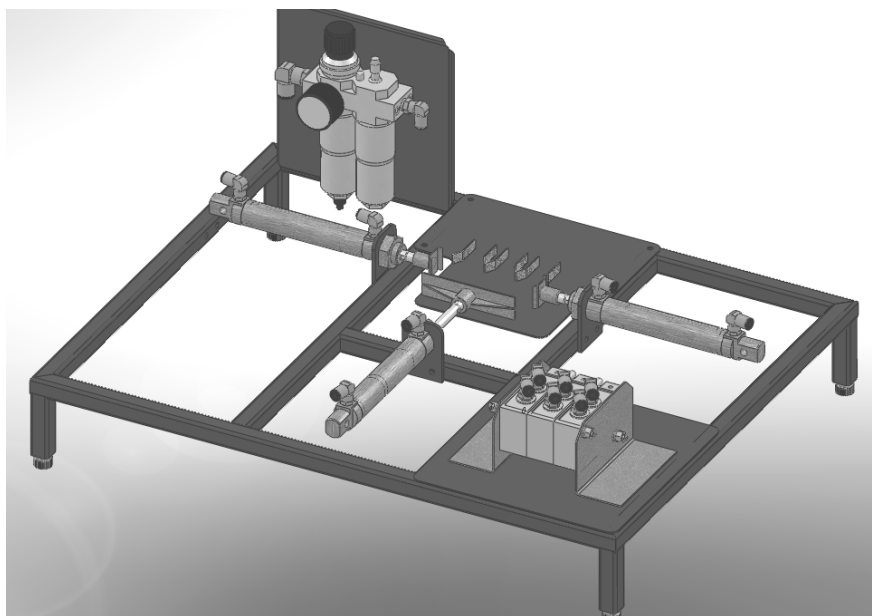


Рисунок 3 – Лабораторный стенд. Общий вид

Данный стенд представляет собой действующую модель установки по сортировке деталей. Рассмотрим подробнее алгоритм его работы:

- 1) подача детали в рабочую зону (передняя кромка стола);
- 2) при появлении сигнала о наличии детали в рабочей области включается пневмораспределитель, отвечающий за работу цилиндра толкателя, в результате чего толкатель начинает поступательное движение, после которого деталь

оказывается в зоне сортировки. По прошествии определенного времени, заданного программно с использованием таймера, пневмораспределитель отключается, и шток с толкателем отходит на исходную позицию;

3) в соответствии с предварительно заданным номером лотка в управляющей программе происходит установка или сброс битов, отвечающих за включение соответствующих выходов контроллера. К этим выходам подключены катушки пневмораспределителей, отвечающих за движение цилиндров позиционирования деталей. С помощью этих цилиндров производится перемещение детали на сортировочном столе, в результате чего она оказывается сориентирована напротив одного из лотков. После этого штоки цилиндров возвращаются в исходное положение;

4) производится полное выдвижение штока цилиндра толкателя, при этом деталь оказывается в заданном лотке. Затем шток возвращается в исходное положение. Далее цикл повторяется.

Средой программирования учебного стенда в рамках данной лабораторной работы является программный пакет CX-Programmer (рисунок 4). Программа CX-Programmer – это инструмент программирования ПЛК, предназначенный для создания, тестирования и обслуживания программ для ПЛК серии CS/CJ, серии CV и серии C производства OMRON. В ней предусмотрено все необходимое для работы с устройствами класса ПЛК и адресной информацией, а также для осуществления связи с ПЛК производства OMRON по сетям поддерживаемых типов.

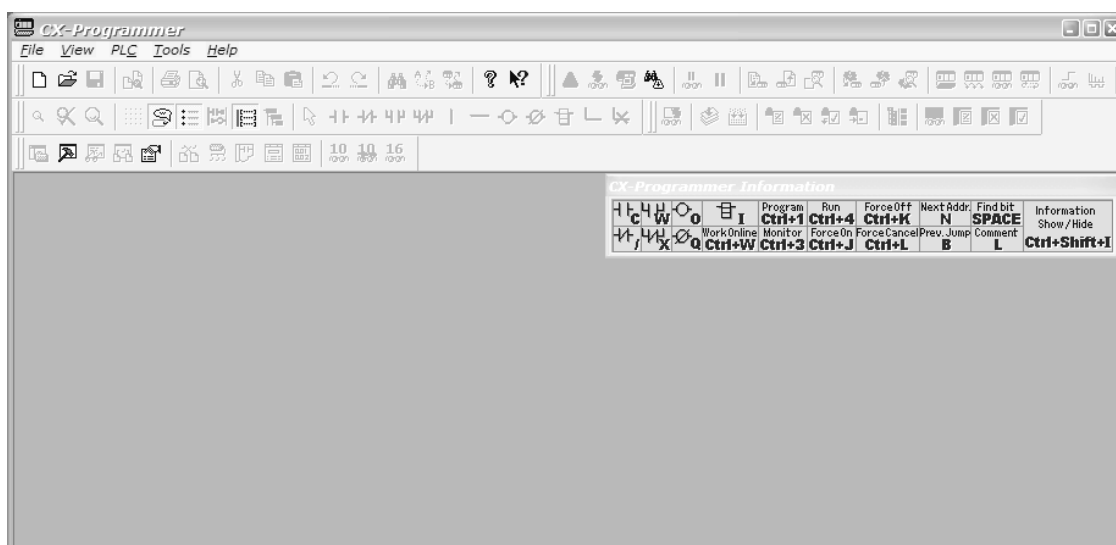


Рисунок 4 – Окно CX-Programmer

Ниже перечислены важнейшие функции данного программного пакета.

- Резервное копирование во флэш-ПЗУ – Поддерживается резервное сохранение данных во флэш-ПЗУ для ПЛК, в которых реализована эта функция.
- Поиск и замена – Расширены границы и возможности функций поиска и замены. Графический интерфейс видоизменен и содержит элементы для выбора диапазона поиска, позволяя распространить поиск на сегмент, глобальный символ и/или локальный символ.

- Считывание/Загрузка – Функции считывания/загрузки обновлены и поддерживают ПЛК, не требуя поиска маркеров сегментов в исходном диалоговом окне Upload (Загрузка).

- Диапазон команды Send/Rec. расширен.

- Функциональность – Добавлены новые уровни функциональности: Junior (Облегченная версия), Demo (Демоверсия) и Trial (Пробная версия).

- Сортировка символов – Сортировка символом в таблице символов для данных числового типа усовершенствована и выполняется отдельно от данных другого типа.

- В набор поддерживаемых файлов прямого импорта включены файлы CPT, SP1 и COD.

- Изменены размеры и положения диалоговых окон редактирования контактов/катушек/команд, поиска и замены и комментирования строк.

- Имитатор online-режима (Auto Online, Work Online) – Расширенная функциональность в online-режиме упрощает подключение к ПЛК и отладку программ.

- Комбинирование и разделение строк – Добавлены функции комбинирования и разделения, упрощающие добавление и разделение строк программы.

- Окно Watch – Окно Watch (Таблица мониторинга) дополнено возможностью непосредственного ввода адресов в таблице мониторинга.

- Назначение клавиш – Информацию о назначении клавиш можно сохранять в файл (*.mac) и загружать в CX-Programmer.

- Менеджер сегментов/строк – Поддерживается функция редактирования структуры программы. В данном диалоговом окне можно редактировать сегменты, строки и комментарии.

- Отображение комментариев к входам/выходам – Предусмотрено окно для редактирования комментариев к адресам.

Далее приведено описание процесса программирования контроллера «ОМРОН» для решения задач транспортно-складского комплекса посредством автоматизирования процессов перемещения заготовок с помощью пневмоцилиндров.

Задания: 1) повторить весь процесс программирования и выполнить программу для размещения заготовок в конечные позиции, начиная с правого края – к левому;

2) дополнить программу фиксацией среднего пневмоцилиндра в промежуточной позиции по сигналу с датчика и вычислить сокращение времени всего процесса;

3) дополнить программу управлением с пульта оператора таким образом, чтобы можно было задавать номер позиции, в которую должна быть перемещена заготовка.

После выбора из меню «File» нового проекта появляется окно настройки контроллера (рисунок 5), в котором необходимо выбрать контроллер, установленный на стенде и аналогичный тому, который представлен на скриншоте экрана программы.

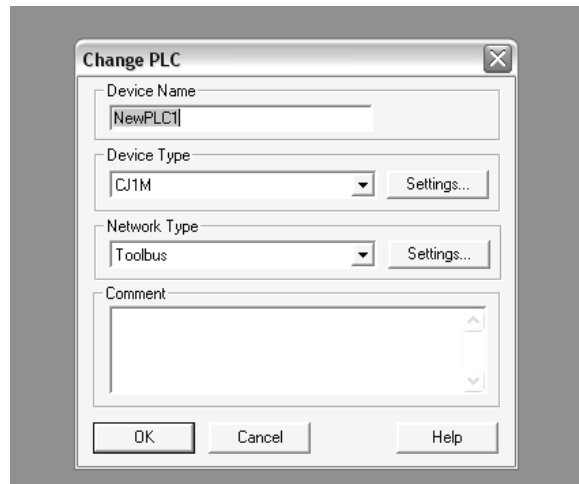


Рисунок 5 – Настройка контроллера при создании нового проекта

Составим простейшую программу для выдвигания одного пневмоцилиндра по нажатию кнопки пульта управления (рисунок 6). Все необходимые для построения элементы обозначены пиктограммами на панели инструментов в верхней части программы. Имена кнопки и пневмоцилиндра записываются t: 0.01 и Q: 1.01 соответственно.

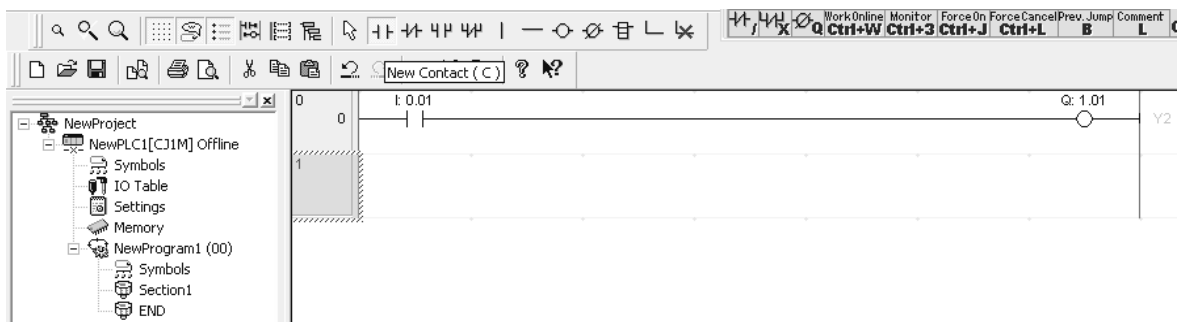


Рисунок 6 – Управление пневмоцилиндром при помощи кнопки

Далее настраиваются модули, установленные на него. Для этого в выпадающем меню, расположенном в программе слева, выбирается пункт IO Table (рисунок 7). В появившемся справа окне «PLC IO Table» с помощью двойного щелчка мыши на слотах оборудования 00 и 01 выбираются дискретные модули ввода и вывода.

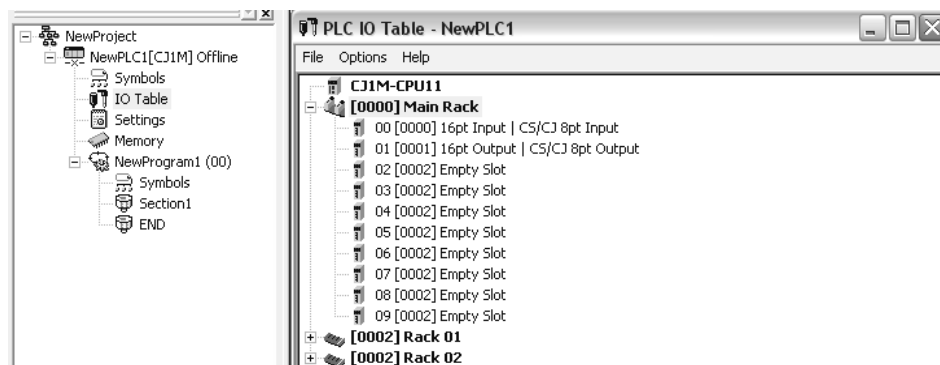


Рисунок 7 – Выбор оборудования контроллера, установленного на стенде

Теперь, если все предыдущие шаги сделаны правильно, можно подключиться к контроллеру и проверить нашу программу. Для этого в выпадающем меню PLC надо выбрать подпункт «Work Online» (рисунок 8). Если контроллер

включён и соединение прошло успешно, то следующим шагом станет загрузка программы на него.

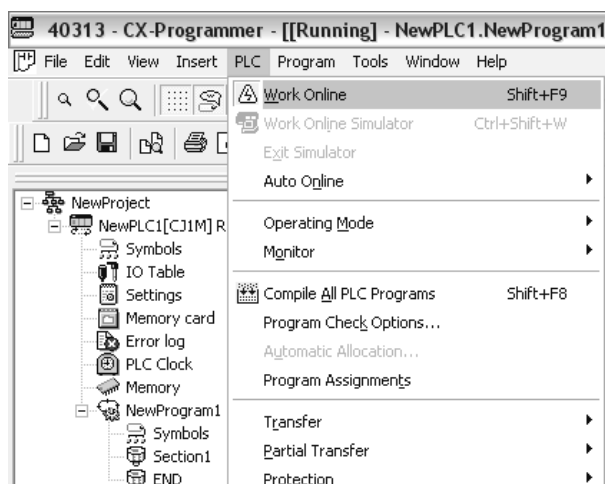


Рисунок 8 – Подключение CX-Programmer к контроллеру

Для этого необходимо выбрать из меню «PLC» -> «Transfer» -> «To PLC...» (рисунок 9), а также установить «галочки» у всех пунктов загрузки появившегося окна «Download Options», как показано на рисунке 10, чтобы перенести на ПЛК не только саму программу, но и все необходимые для неё настройки.

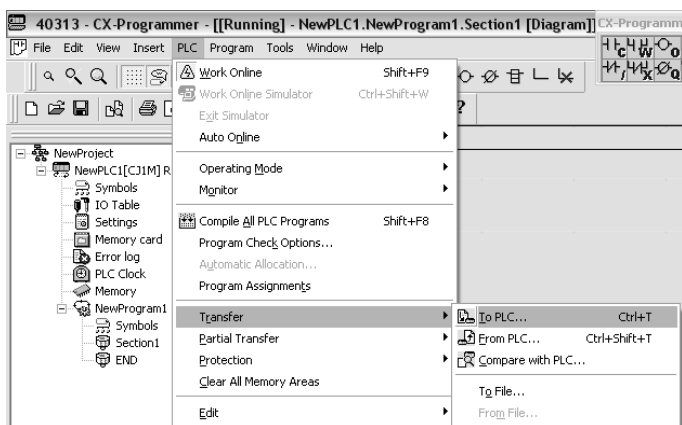


Рисунок 9 – Загрузка программы в контроллер

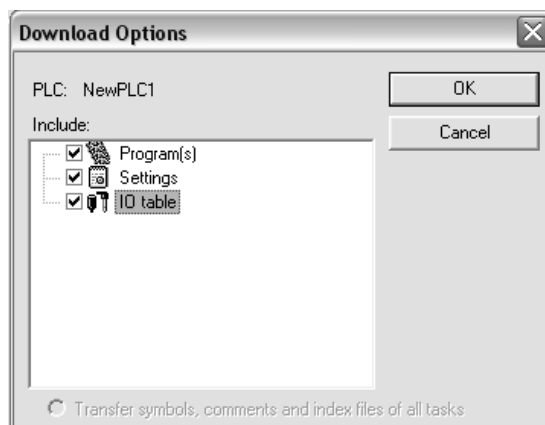


Рисунок 10 – Выбор настроек загрузки

Если всё сделано без ошибок, то, оставаясь в режиме «Онлайн», можно будет отследить в программе в реальном времени внешние управляющие воздействия на стенд, осуществляемые при помощи кнопки, а также моменты, когда пневмоцилиндр приходит в движение. Визуально нажатие кнопки, обозначенной $t: 0.01$, и реакция на это клапана $Q: 1.01$ будут выглядеть как прохождение сигнала – зелёной линии по пока единственной строке программы (рисунок 11). Этим способом можно определить исправность программного кода и оборудования, так как все изменения состояния кнопки будут отображаться в программе. Если при этом пневмоцилиндр не будет работать, то необходимо будет проверить, достаточный ли уровень давления воздуха в пневмосистеме стенда, и увеличить его при помощи насоса при необходимости.

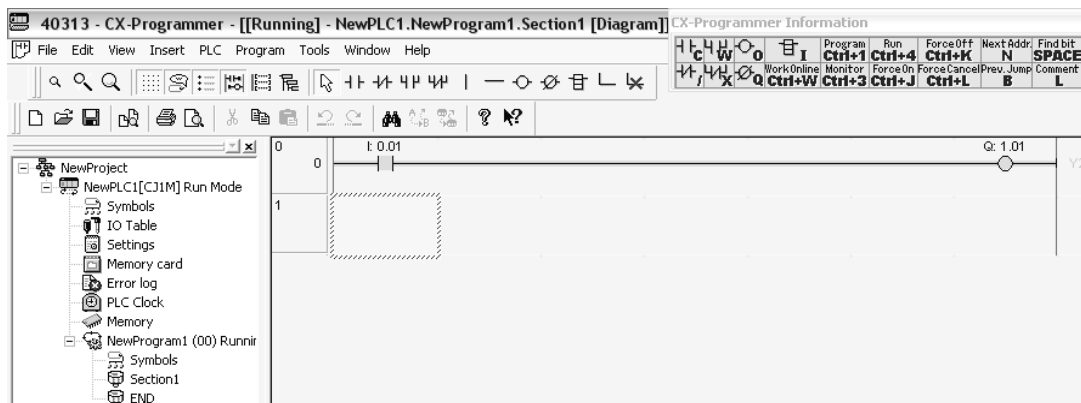


Рисунок 11 – Выполнение программы

Дальнейшим этапом развития программы будет управление всеми тремя цилиндрами при помощи различных кнопок пульта управления. Реализация данной программы представлена на рисунке 12. Отличается она от предыдущей только тем, что содержит не одну, а три строки кодов, в которых соответственно выбраны номера пневмоцилиндров и кнопок: t: 0.02 для Q: 1.02, t: 0.05 для Q: 1.00 и исходная часть – t: 0.01 для Q: 1.01.

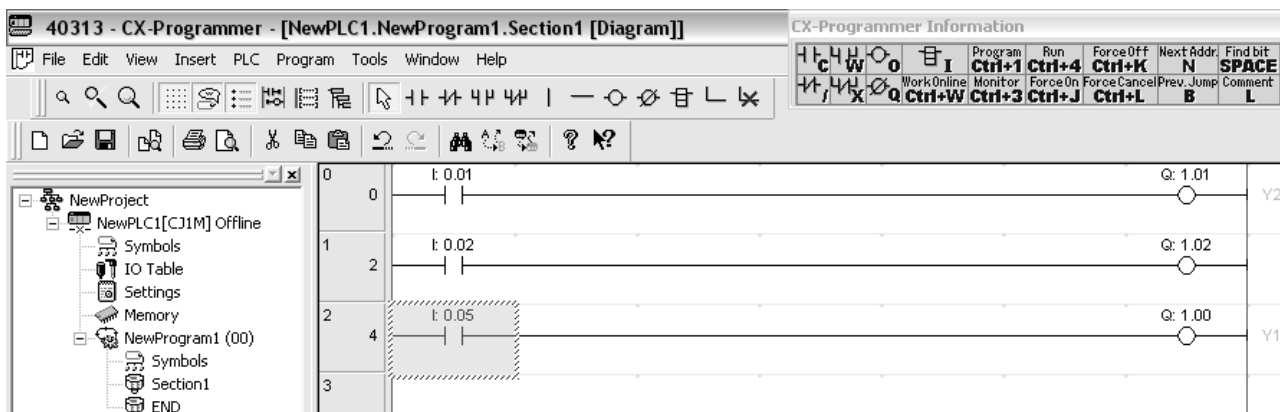


Рисунок 12 – Управление тремя пневмоцилиндрами с пульта

Собрав из элементов представленную на рисунке программу, загрузите её аналогичным образом в контроллер, предварительно проверив правильность настройки таблиц его входов и выходов (IO Table), как в случае с управлением одним клапаном (рисунок 7).

Ход выполнения программы для нажатия на одну из трёх управляющих кнопок представлен на рисунке 13. Здесь можно увидеть, что зелёная линия сигнала, активирующая те или иные исполнительные элементы (в нашем случае пневмоцилиндры) «упирается» в не нажатые кнопки.

Попробуйте: как Вы уже могли заметить, на пульте управления контроллера есть три кнопки и два переключателя. Модифицируйте программу так, чтобы для управления левым цилиндром необходимо было перевести левый переключатель в позицию «1», для управления правым цилиндром – правый переключатель в позицию «1», а для управления центральным цилиндром – оба переключателя в позицию «0».

Попробуйте: помимо пневмоцилиндров учебный стенд оснащён аварийным клапаном сброса давления, которым можно управлять аналогичным способом. Допишите программу таким образом, чтобы он открывался при нажатии

на две из трёх любых кнопок пульта управления. Его номер можно определить экспериментально или воспользовавшись «Методическими указаниями к практическому занятию и самостоятельной работе по курсу "Программное обеспечение компьютерных систем управления"».

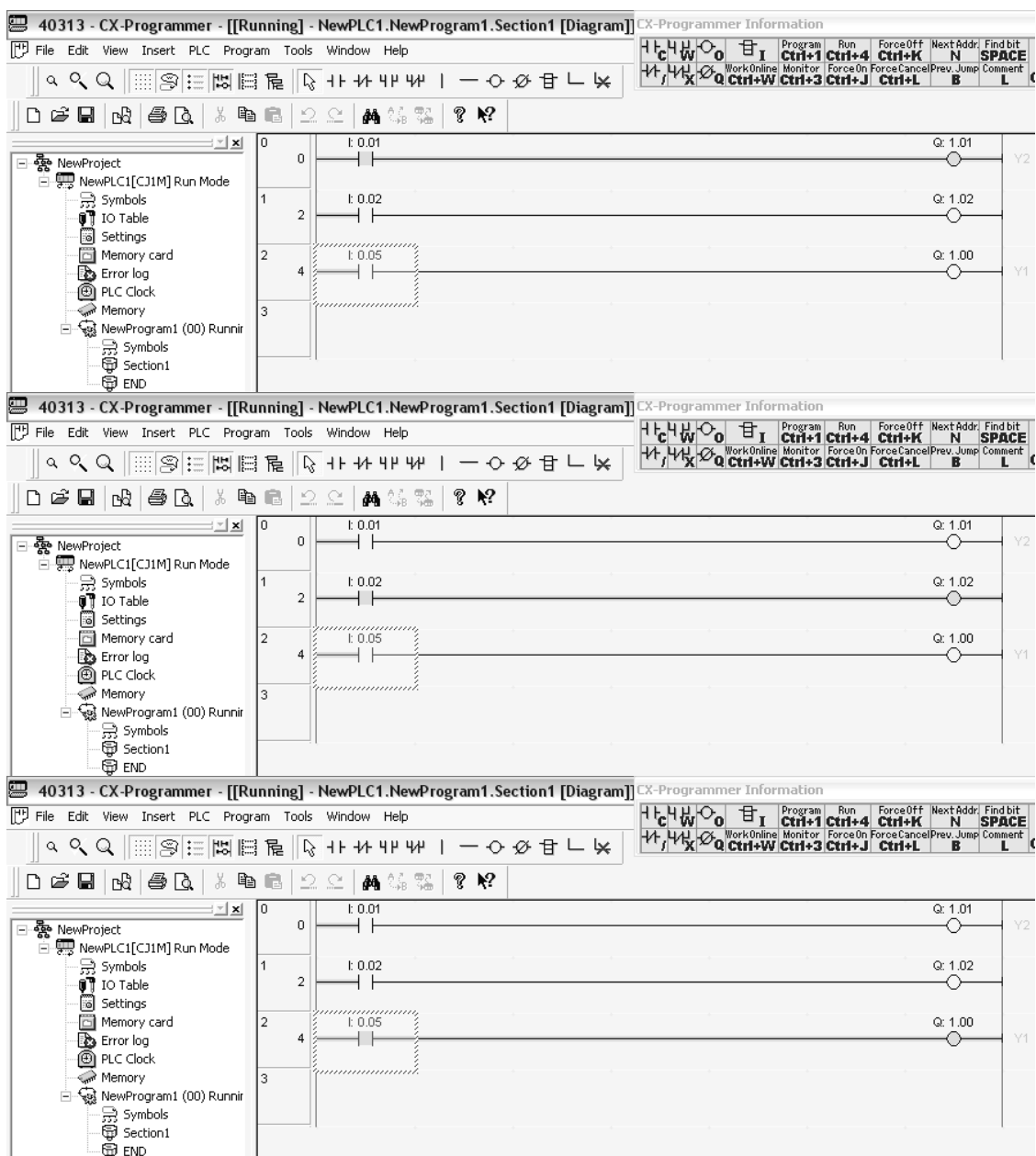


Рисунок 13 – Выполнение программы

Теперь, зная, как организовать управление исполнительными элементами и осуществлять контроль за датчиками-кнопками, можно произвести дальнейшее усовершенствование программы, дополнив её специальными «таймерами» для реализации циклического управления пневмоцилиндрами. Это может быть необходимо в тех случаях, когда надо по нажатию одной клавиши пульта оператора совершить несколько последовательных действий, например, по перемещению заготовки в нескольких плоскостях в процессе сортировки на площадях автоматизированного складского помещения.

Создадим новую программу, предварительно выполнив все настройки контроллера и его модулей аналогично тому, как показано в начале данных ме-

тодических указаний. Добавим кнопку t: 0.01. После этого в меню выберем элемент таймер и в появившемся окне пропишем для него следующие настройки: TIM 0001 #10 так, как показано на рисунке 14.

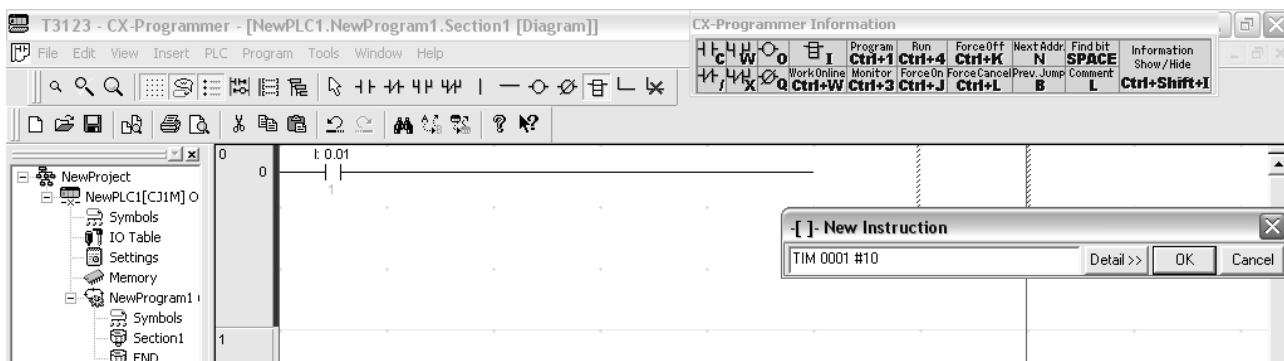


Рисунок 14 – Настройка таймера для программы

После добавления таймера в код программы её внешний вид должен быть идентичен представленному на рисунке 15.

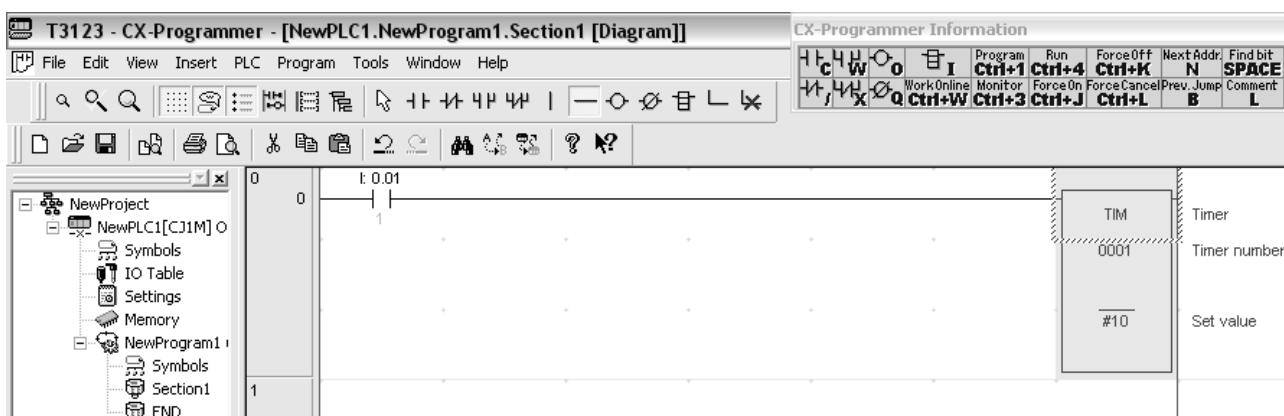


Рисунок 15 – Добавление таймера в программу

Загрузите её в контроллер и проверьте правильность выполнения прежде чем переходить к программе циклического управления пневмоцилиндрами. Реакция стенда на нажатие соответствующей кнопки должна отражаться в программе образом, аналогичным представленному на рисунке 16.

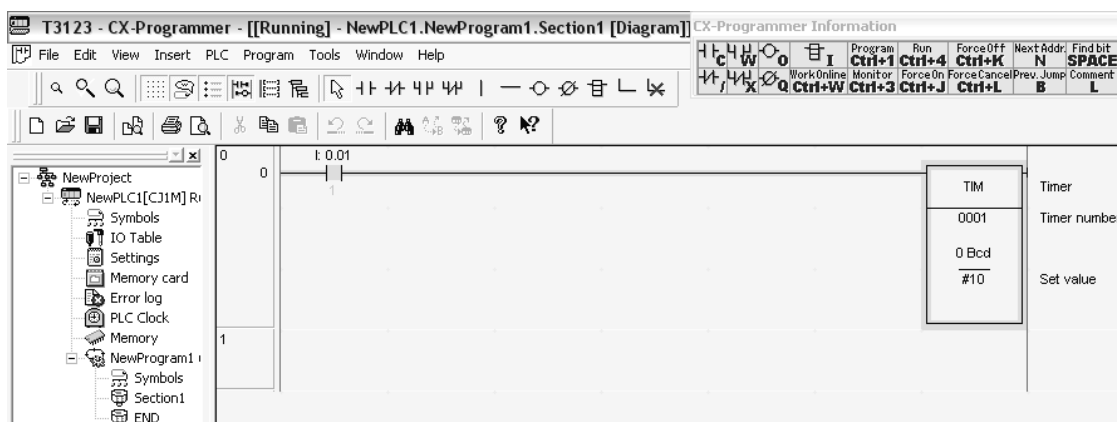


Рисунок 16 – Выполнение таймера на контроллере

Если всё работает правильно, то можно переходить к заключительной части основного задания – написанию и отладке программы последовательного перемещения пневмоцилиндров по жёстко заданному алгоритму. По сравнению

с предыдущими примерами, данная программа довольно объёмна, и её код представлен на двух рисунках – 17 и 18. Как видно из представленных скриншотов CX-Programmer, управление циклом перемещений пневмоцилиндров запускается при помощи нажатия одной из кнопок пульта оператора (в нашем случае – это t: 0.01), а все дальнейшие действия осуществляются по соответствующим значениям запрограммированных таймеров.

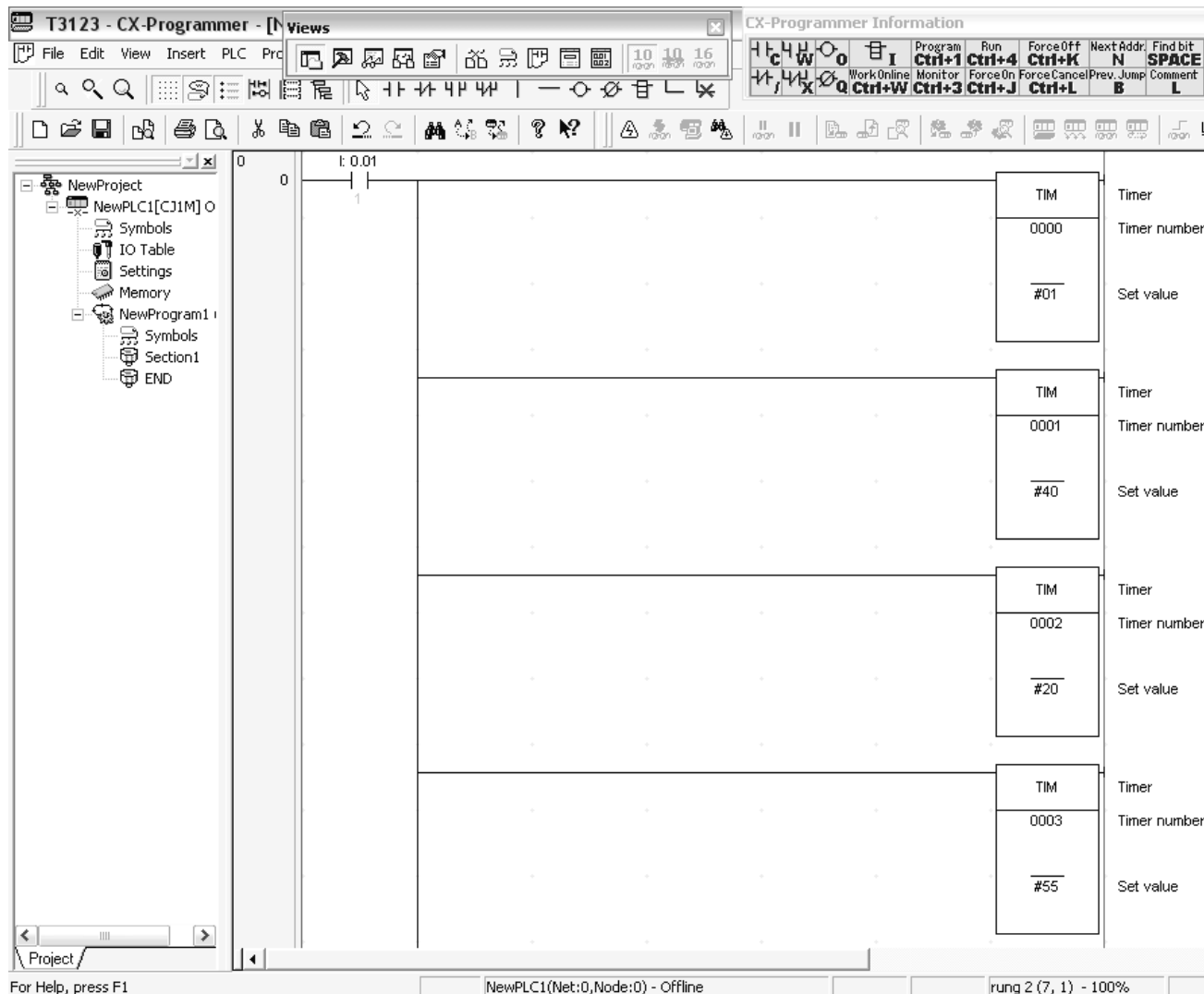


Рисунок 17 – Циклическое управление пневмоцилиндрами

Для проверки правильности выполнения программы и усвоения пройденного материала измените программу таким образом, чтобы активация пневмоцилиндра стенда осуществлялась сначала против часовой стрелки, начиная с правого цилиндра, а затем – по часовой стрелке, начиная с центрального.

В задании 2 необходимо считывать информацию о положении среднего пневмоцилиндра при помощи установленного на него датчика. В программу он добавляется способом, аналогичным способу с кнопками. Его номер можно определить экспериментально или воспользовавшись «Методическими указаниями к практическому занятию и самостоятельной работе по курсу "Программное обеспечение компьютерных систем управления"».

В задании 3 необходимо для каждой из трёх кнопок написать свой алгоритм перемещения пневмоцилиндров для различных номеров позиций заготовок, причём первой позицией надо считать крайний левый «карман» стенда.

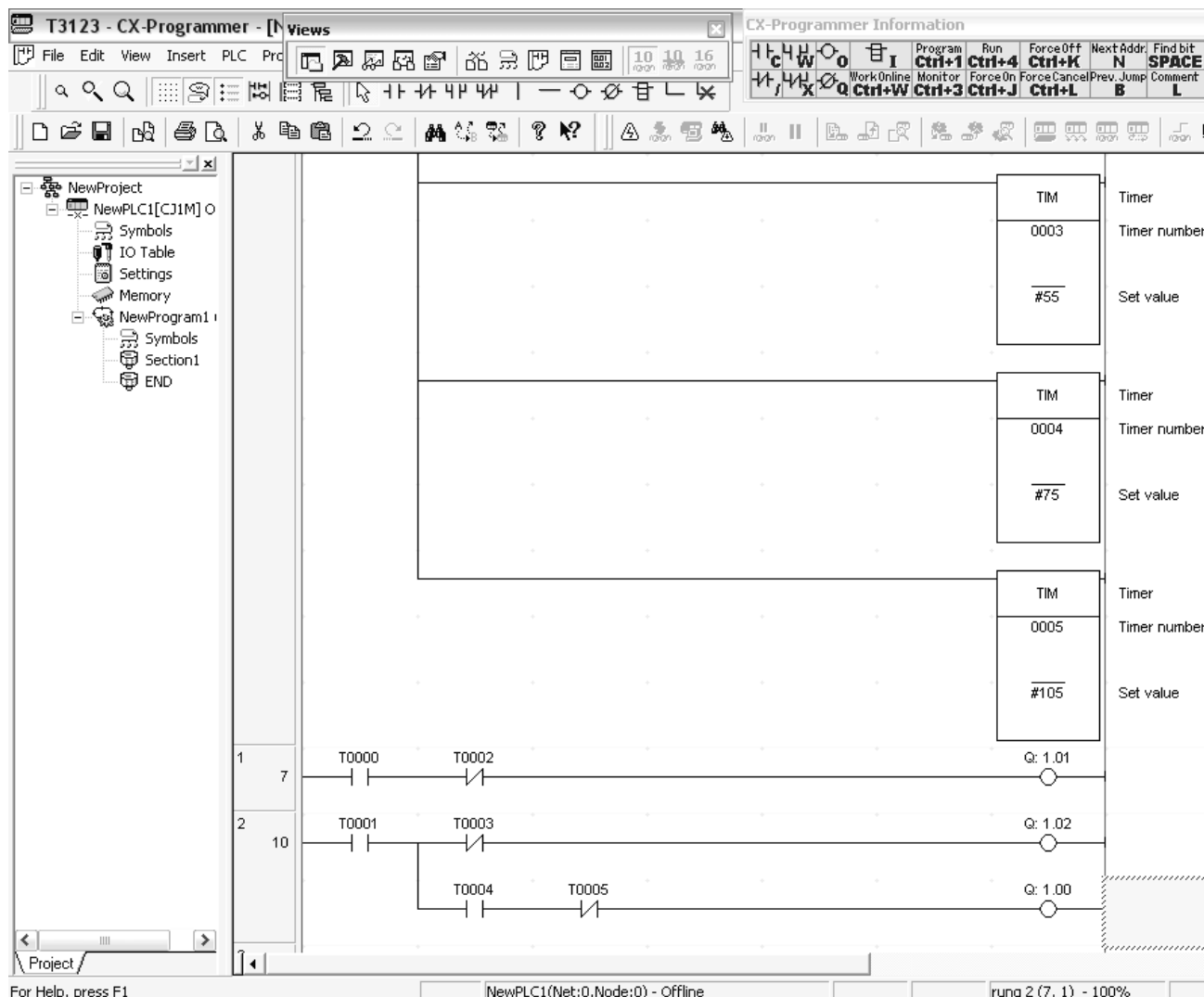


Рисунок 18 – Циклическое управление пневмоцилиндрами (продолжение)

3 ВОПРОСЫ ДЛЯ САМОСТОЯТЕЛЬНОГО ИЗУЧЕНИЯ И ЗАДАНИЯ ПОВЫШЕННОЙ СЛОЖНОСТИ

1 Выполните все задания из обеих лабораторных работ данных методических указаний.

2 В первой лабораторной работе использовались датчики, измеряющие различные факторы безопасности. Приведите примеры их применения и применения комплекса безопасности аналогичной структуры в реальных условиях.

3 Во второй лабораторной работе напишите программу таким образом, чтобы по нажатию на определённую кнопку пульта управления, выполнение всей программы прерывалось и исполнительные элементы возвращались в исходные позиции.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 ИЗУЧЕНИЕ И ТЕСТИРОВАНИЕ ХАРАКТЕРИСТИК ДАТЧИКОВ.....	3
2 УПРАВЛЕНИЕ ПНЕВМАТИЧЕСКИМ И ЭЛЕКТРИЧЕСКИМ ДВИГАТЕ- ЛЯМИ, ПРИМЕНЯЕМЫМИ В СИСТЕМАХ ИЗМЕРЕНИЯ УРОВНЯ БЕЗ- ОПАСНОСТИ.....	7
3 ВОПРОСЫ ДЛЯ САМОСТОЯТЕЛЬНОГО ИЗУЧЕНИЯ И ЗАДАНИЯ ПО- ВЫШЕННОЙ СЛОЖНОСТИ.....	16

Карпов Егор Константинович

СОВРЕМЕННЫЕ МЕТОДЫ АВТОМАТИЗИРОВАННЫХ ИЗМЕРЕНИЙ УРОВНЯ БЕЗОПАСНОСТИ

Методические указания
к комплексу лабораторных работ для студентов направления подготовки
20.04.01 «Техносферная безопасность» направленность
(«Безопасность жизнедеятельности в техносфере»)

Авторская редакция

Подписано в печать 08.10.19	Формат 60x84 1/16	Бумага 65 г/м ²
Печать цифровая	Усл. печ. л. 1,25	Уч.-изд. л. 1.25
Заказ 138	Тираж 25	Не для продажи

БИЦ Курганского государственного университета.
640020, г. Курган, ул. Советская, 63/4.
Курганский государственный университет.