

|  |  |
| --- | --- |
| Дата | Вид деятельности |
| 06.09.2022 | Получение задания на практику |
| 07.09.2022 – 25.11.2022 | Выполнение задания |
| 25.11.2022 – 27.11.2022 | Подготовить и оформить отчет |

Введение

Производственная практика является неотъемлемой и важной частью учебного процесса. В ходе неё не только закрепляются теоретические знания, полученные во время обучения, но и отрабатываются необходимые практические умения для эффективной работы в профессиональной сфере, осваивается инструментарий для работы и взаимодействия в команде, оттачиваются коммуникативные навыки для общения с коллегами.

Данный документ представляет собой отчет по производственной практике, проходившей в период с 6 сентября по 27 ноября ПАО «Кировский завод «Маяк», г. Киров.

1. Общие сведения о предприятии

ПАО «Кировский завод «Маяк» был создан 30 июля 1941 года.

Физический адрес: Российская Федерация, Кировская область, город Киров, улица Молодой Гвардии, 67

Публичное акционерное общество (ПАО) — это общество, чьи акции и ценные бумаги, которые конвертируются в акции, публично размещаются или публично обращаются. То есть их можно свободно приобрести на рынке ценных бумаг.

Сегодня это крупное машиностроительное предприятие, специализирующееся на выпуске товаров народного потребления, а также специальной военной продукции.

Отдел главного конструктора, в котором проходила практика, занимается разработкой и сопровождением конструкторской документации нестандартного оборудования. Нестандартное оборудование служит для проверки электрических параметров изделий, выпускаемых ПАО «КЗМ  «Маяк», а также покупных изделий, проверяемых на входном контроле.

1. Описание выполненной работы

В соответствии с заданием от организации во время прохождения практики необходимо разработать устройство автоматизированной проверки блоков релейной логики.

Разработка состоит из следующих этапов:

- подключение светодиодов через регистр к плате Arduino

- подключение dip переключателей через регистр к плате Arduino

- создание каскада регистров

* 1. Подключение светодиодов через регистр к плате Arduino
     1. Подключение светодиода ко входу

Для того, чтобы подключить светодиод необходимо:

- установить диод на макетной плате

- катод соединить с «землей» (0 В)

- анод через регистр подключить ко входу

Сопротивление регистра вычисляется по формуле:

**R = (Vps — Vdf) / If**

Где Vps — напряжение источника питания (5В)

Vdf — падение напряжения на элементе

If — номинальный ток элемента

Подключение светодиода представлено на рисунке 1

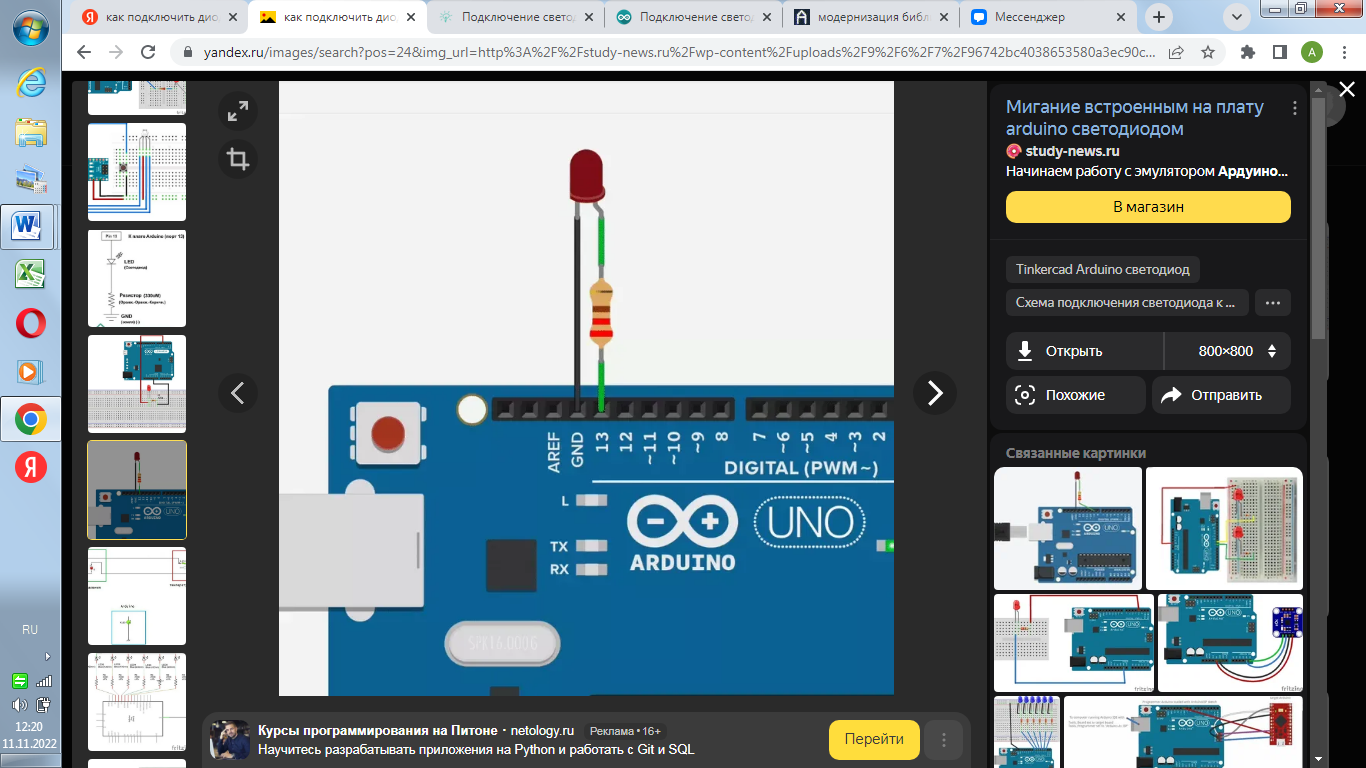


Рисунок 1 – Подключение светодиода к плате

* + 1. Подключение регистра 74hc595 к плате

Подключение регистра осуществляется соединением 11, 12 и 14 пинов регистра c пинам Arduino. Распиновка регистра представлена на рисунке 2

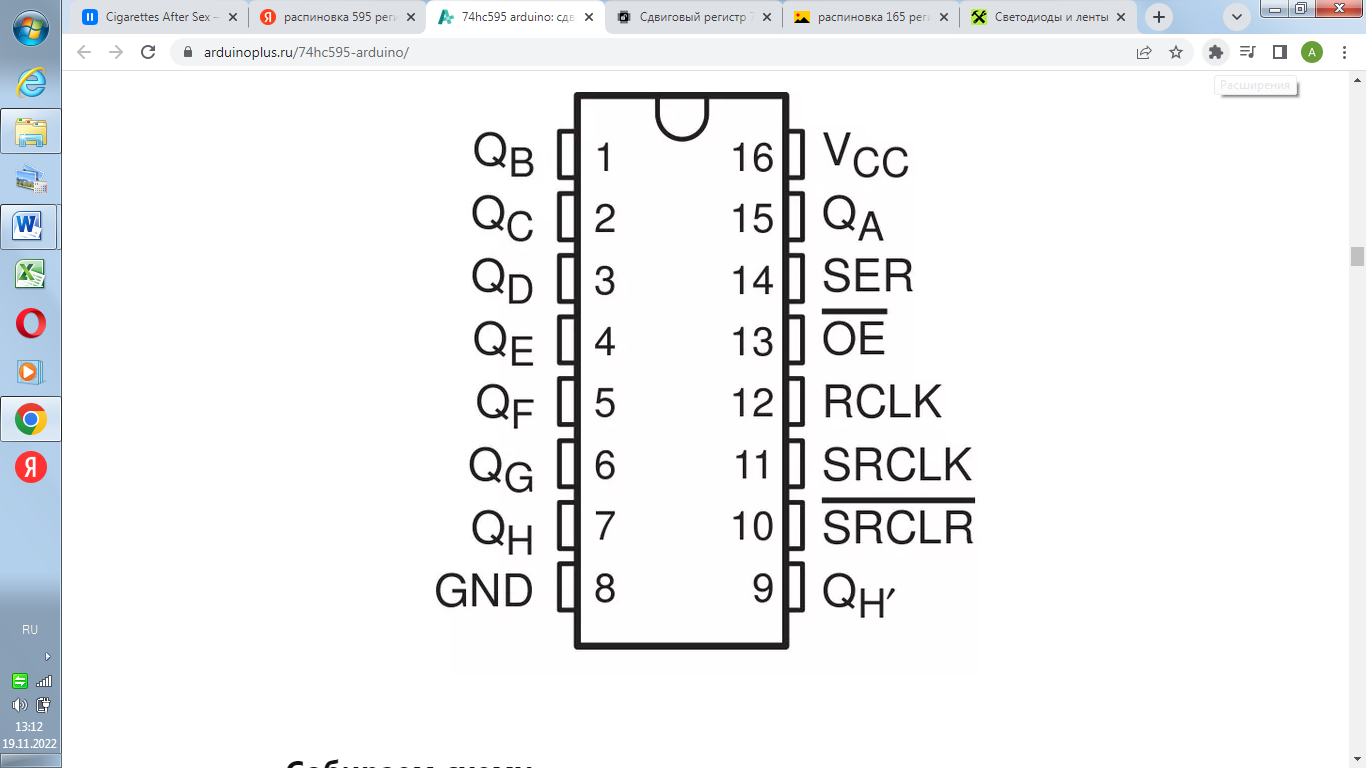


Рисунок 2 – Распиновка регистра

Pin 11 (SRCLK) сдвигового регистра 74HC595 на пин 13 на Arduino - это будет называться «синхронизирующим пином»,

Pin 12 (RCLK) сдвигового регистра на пин 8 на Arduino - это будет обозначаться как «пин защелка»,

Pin 14 (SER) сдвигового регистра на пин 11 на Arduino - это будет называться «пином данных»

* + 1. Подключение светодиодов через регистр к Arduino

Подключение светодиодов представлено на рисунке 3

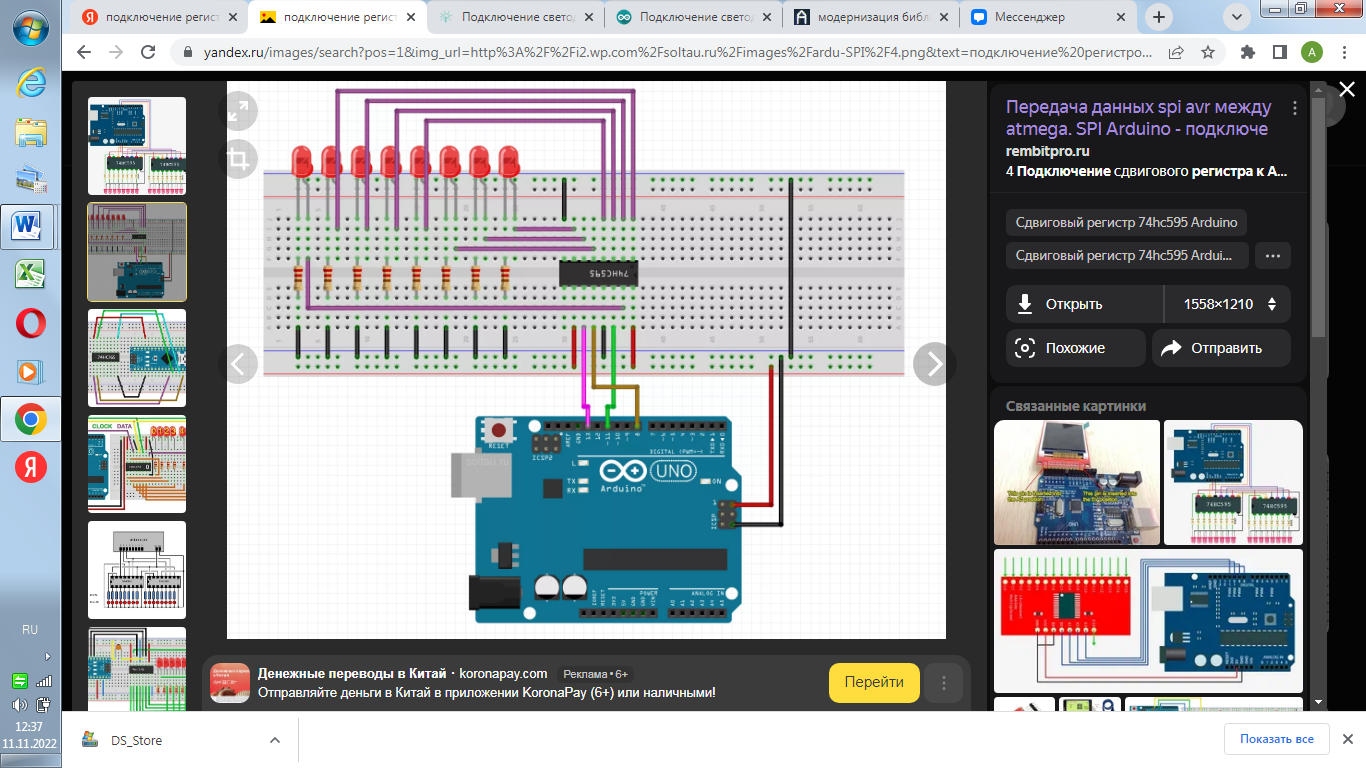


Рисунок 3 – Подключение светодиодов

Для того, что управлять светодиодами через регистр необходимо написать программу. Пример программы приведен на рисунке 4.

const int latchPin=11;

const int dataPin=8;

const int clockPin= 13;

void setup()

{

pinMode(latchPin, OUTPUT);

pinMode(clockPin, OUTPUT);

pinMode(dataPin, OUTPUT);

}

void loop()

{

digitalWrite(latchPin, LOW);

shiftOut(dataPin, clockPin, LSBFIRST, 255);

digitalWrite(latchPin, HIGH);

delay(200);

}

Рисунок 4 – Управление светодиодами через регистр продолжение

* 1. Подключение dip переключателя через регистр к плате Arduino
     1. Подключение dip переключателя ко входам

Для подключения dip переключателя ко входам необходимо

- установить переключатель на макетную плату;

- одну строну соединить с «землей» (0 В);

- противоположную соединить со входами.

Подключение dip переключателя представлено на рисунке 5.

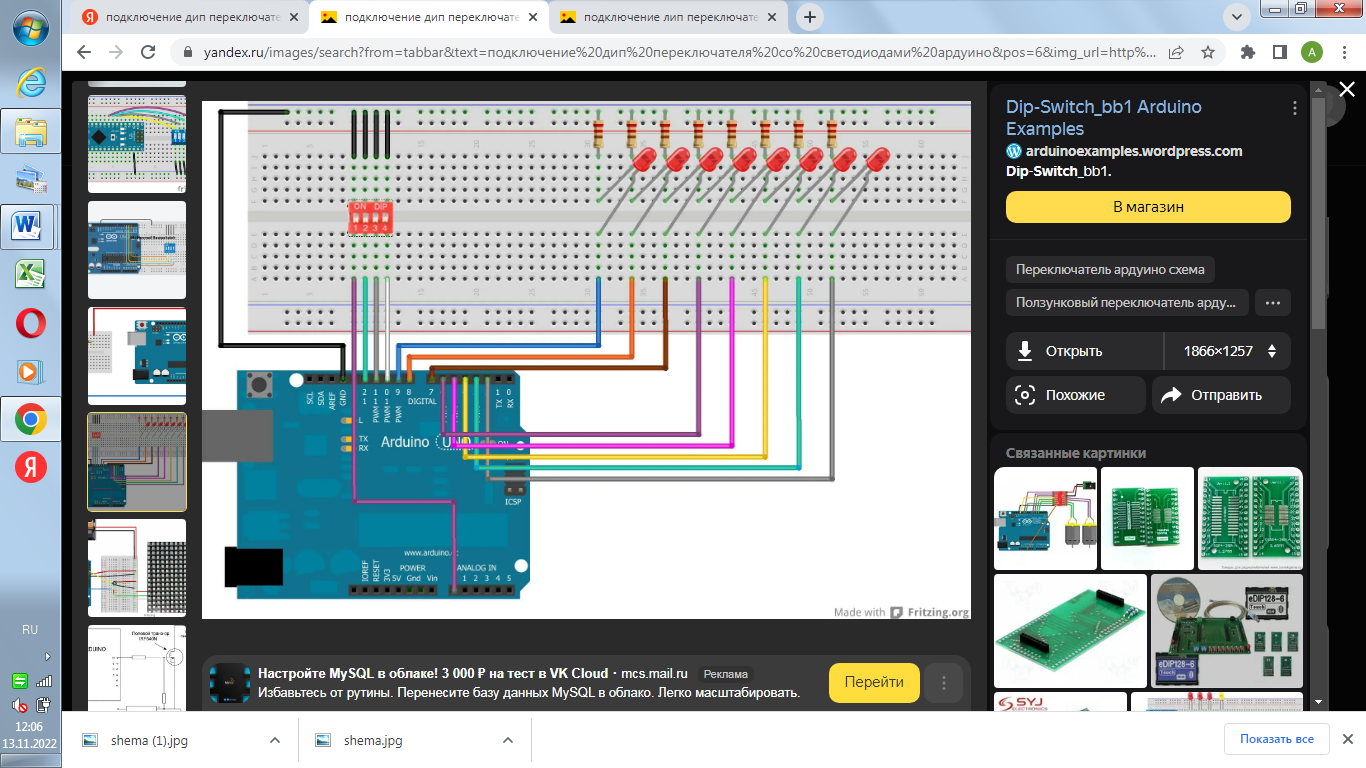


Рисунок 5 – Подключение dip переключателя

* + 1. Подключение регистра 74hc165 к плате

Подключение регистра осуществляется соединением 1, 2, 9 и 15 пинов регистра c пинам Arduino. Распиновка регистра представлена на рисунке 6.

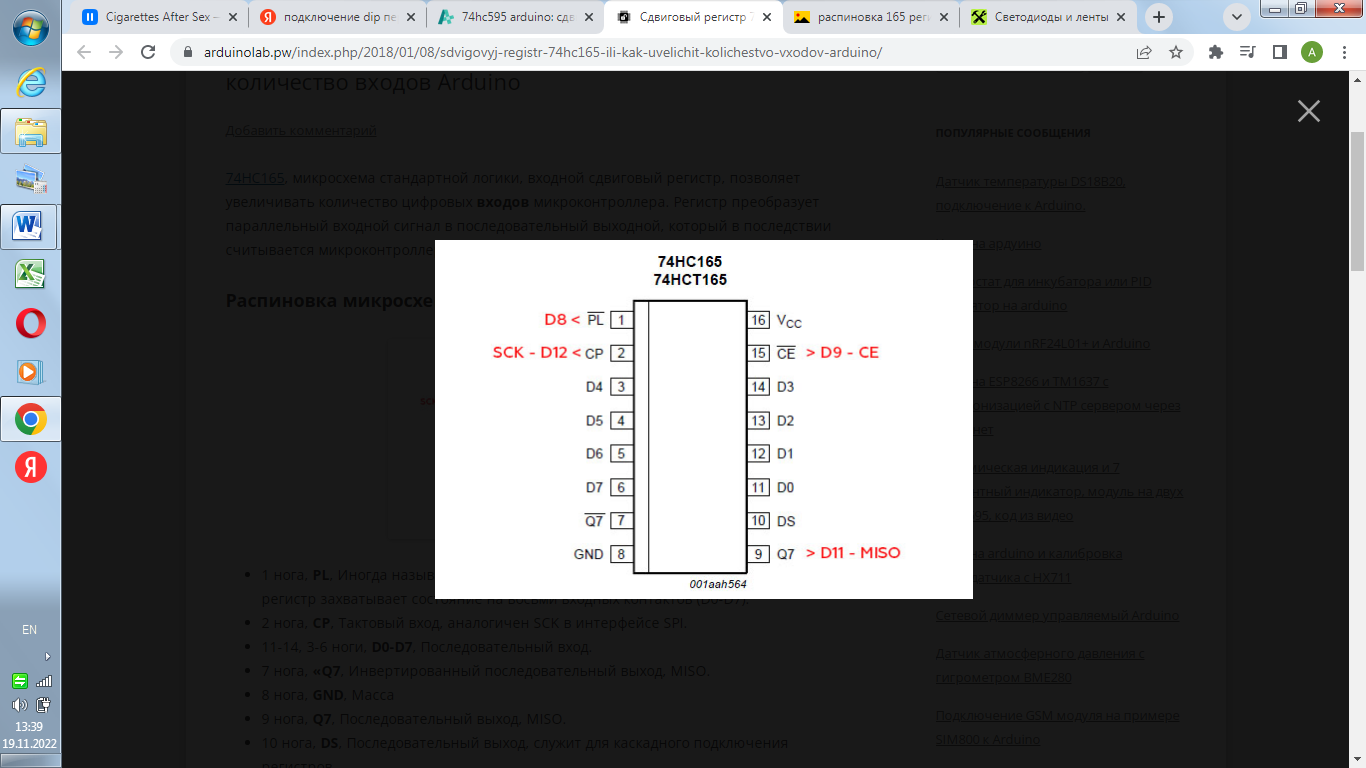


Рисунок 6 – Распиновка регистра

Pin 1 (PL) сдвигового регистра 74HC595 на пин 13 на Arduino - это будет называться «пин защелка»,

Pin 2 (CP) сдвигового регистра на пин 8 на Arduino - это будет обозначаться как «синхронизирующий сигнал на пин»,

Pin 9 (Q7) сдвигового регистра на пин 11 на Arduino - это будет называться «пином данных»

Pin 15 (CE) сдвигового регистра на пин 11 на Arduino - это будет называться «пином разрешения на запись»

* + 1. Подключение dip переключателя через регистр к Arduino

Подключение dip переключателя через регистр представлено на рисунке 7

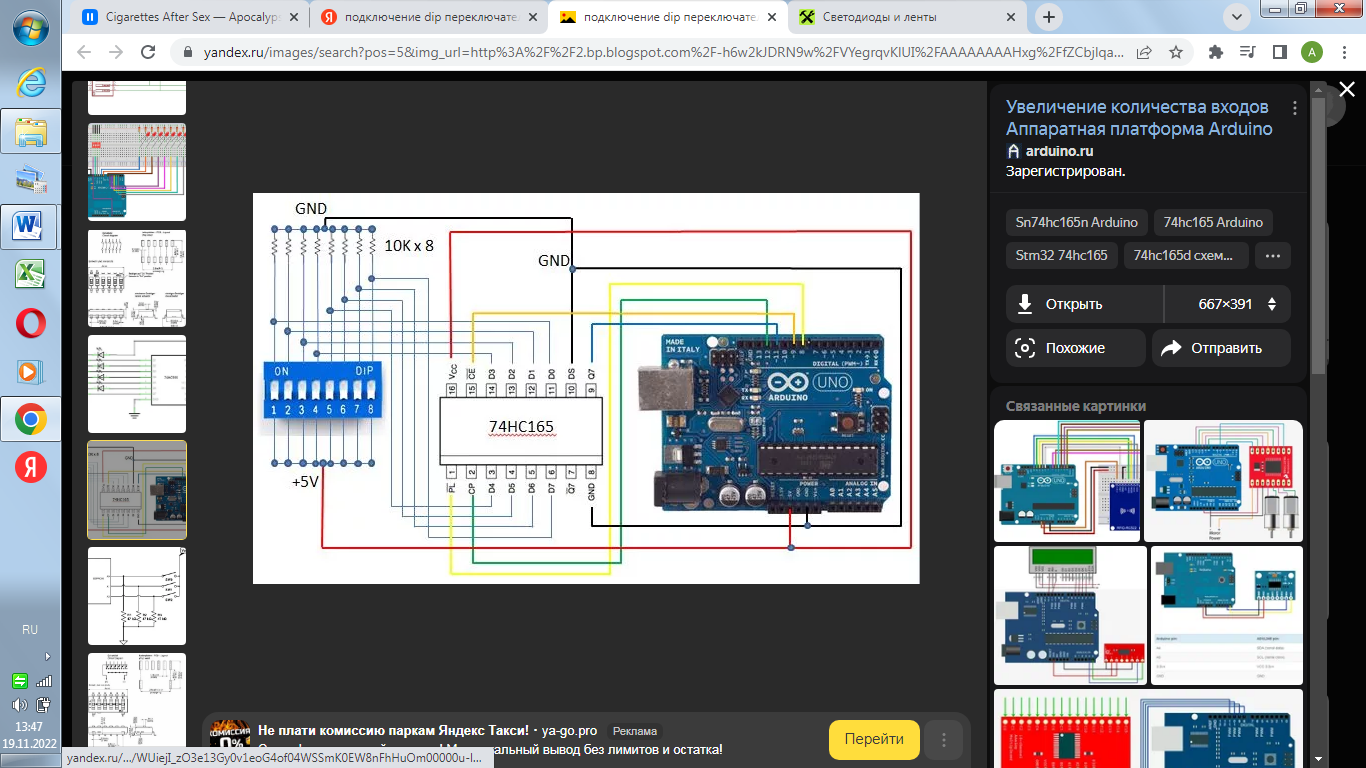


Рисунок 7 – подключение dip переключателя через регистр

Для того, что читать состояний переключателей через регистр необходимо написать программу. Пример программы приведен на рисунках  8-9.

int load = 8;

int clockEnablePin = 9;

int dataIn = 11;

int clockIn = 12;

void setup()

{

Serial.begin(9600);

// Setup 74HC165 connections

pinMode(load, OUTPUT);

pinMode(clockEnablePin, OUTPUT);

pinMode(clockIn, OUTPUT);

pinMode(dataIn, INPUT);

}

Рисунок 8 – Чтение состояний переключателей

void loop()

{

digitalWrite(load, LOW);

delayMicroseconds(5);

digitalWrite(load, HIGH);

delayMicroseconds(5);

digitalWrite(clockIn, HIGH);

digitalWrite(clockEnablePin, LOW);

byte incoming = shiftIn(dataIn, clockIn, LSBFIRST);

digitalWrite(clockEnablePin, HIGH);

Serial.print("Pin States:\r\n");

Serial.println(incoming, DEC);

}

Рисунок 9 – Чтение состояний переключателей продолжение

* 1. Создание каскадов регистров
     1. Каскад регистров 74hc595

Для создания каскада необходимо

- соединить 11 и 12 пины регистров

- соединить 9 пин с 14 пином следующего регистра

Каскад из 2х регистров представлен на рисунке 10

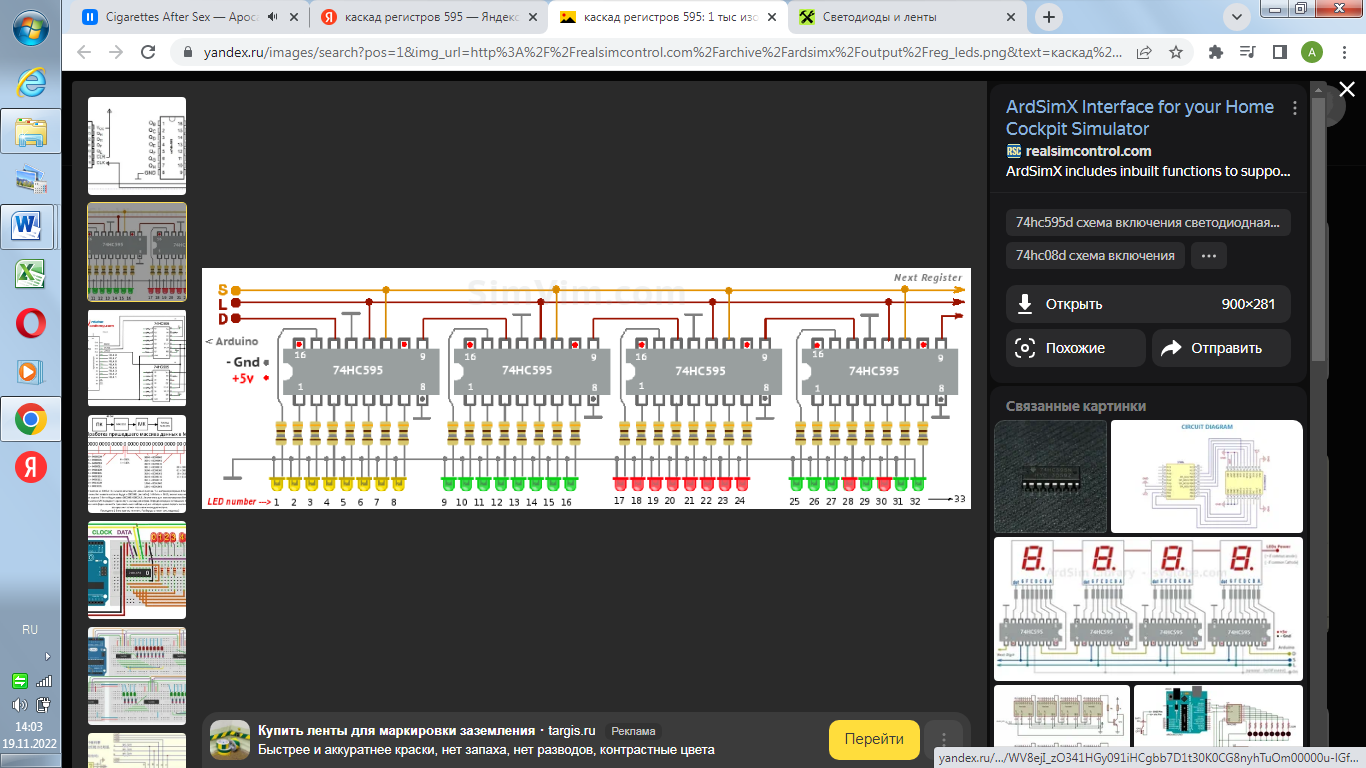


Рисунок 10 – Каскад из 2х регистров

* + 1. Каскад регистров 74hc165

Для создания каскада необходимо:

- соединить 1, 2 и 15 пины регистра

- соединить 10 пин с 9 пином следующего регистра

Каскад из 2х регистров представлен на рисунке 11

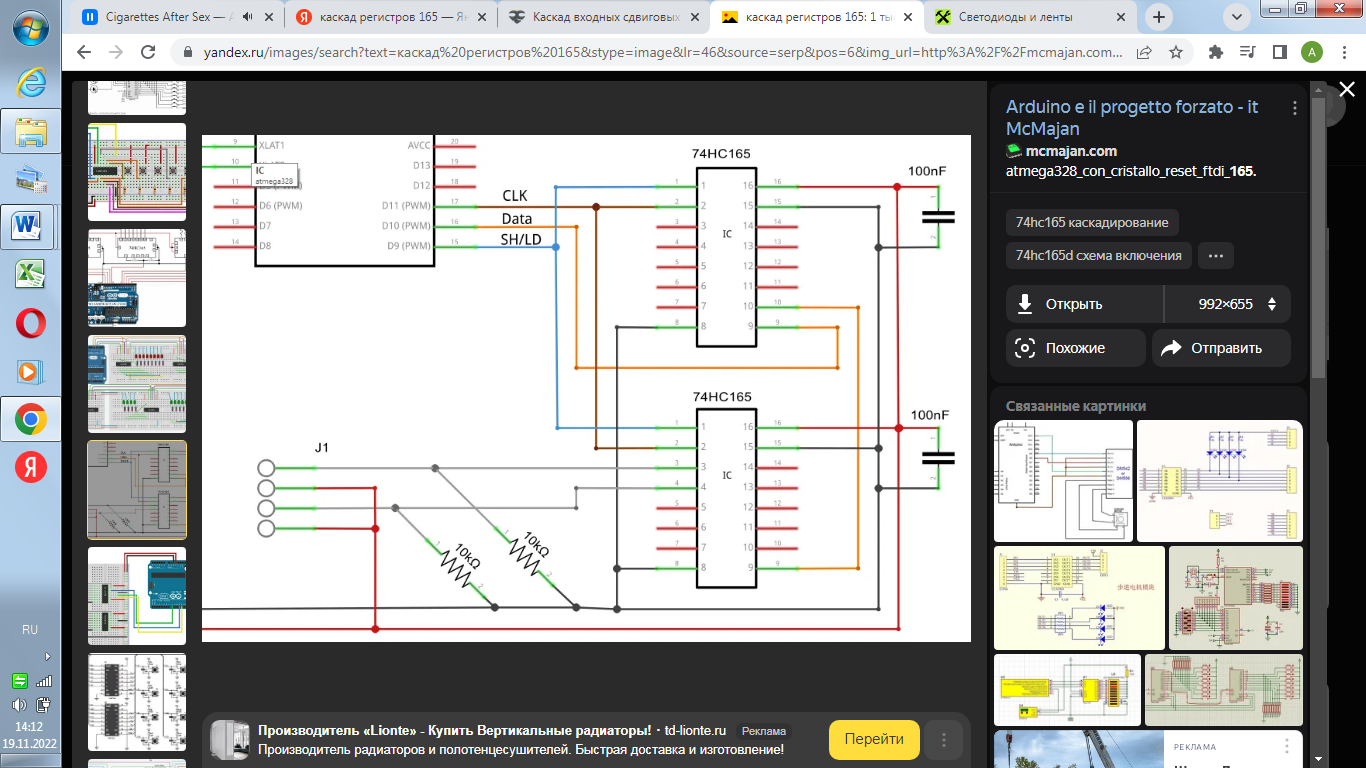


Рисунок 11 – Каскад из 2х регистров

В результате выполнения всех этапов разработки было создано устройство автоматизированной проверки блоков релейной логики, схема которого представлена на рисунке 12 и в приложении А.

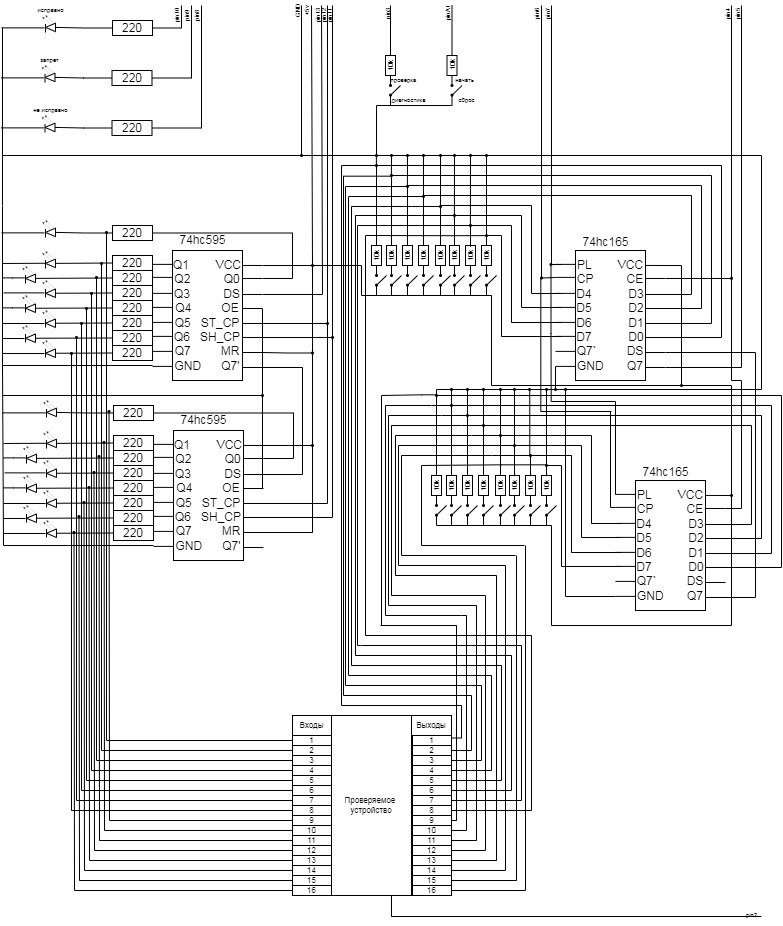


Рисунок 12 – схема устройства

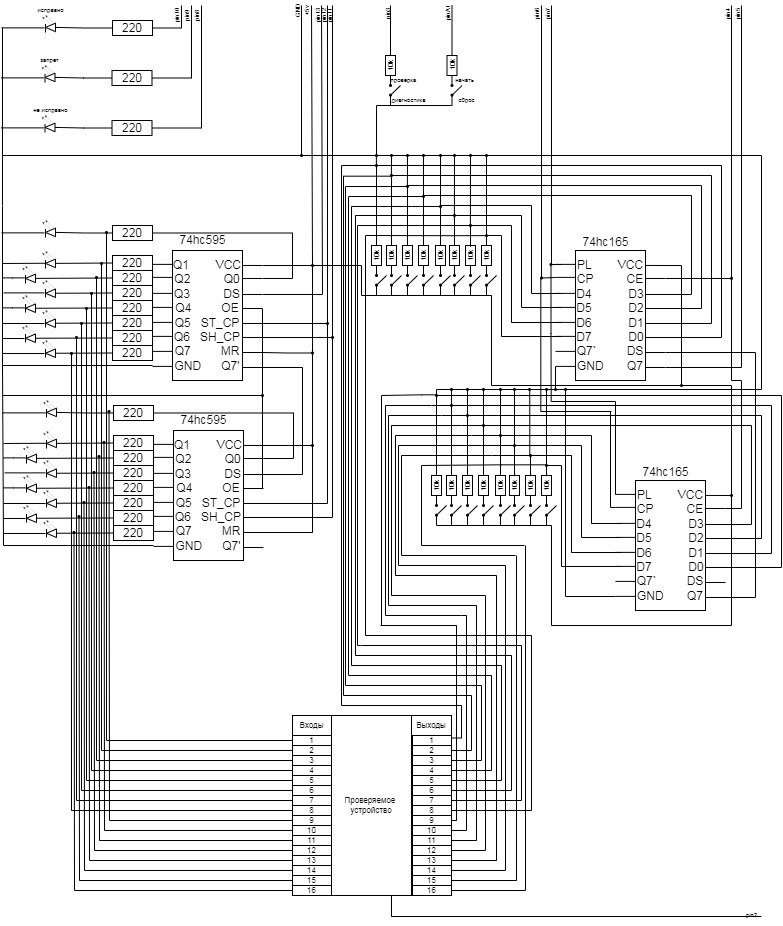
Для управления работой этого устройства написана программа, листинг которой приведен в приложении Б.

Заключение

В ходе выполнения производственной практики были получены навыки работы с конструктором Arduino для изготовления макетов устройств. Кроме того, были получены знания о применении регистров для увеличения числа входов. Все полученные навыки и знания были применены для разработки устройства автоматизированной проверки. В результате было создано устройство для автоматизированной проверки блоков релейной логики, выполняющее требования, которые предъявлялись к нему.

Приложение А

Схема устройства



Приложение Б

Листинг кода

//установка соответствия между названием и номером пина

//595 регистр

int latchPin=12;

int clockPin=11;

int dataPin=13;

//165 регистр

uint8\_t latchIn=7;

uint8\_t clockIn=6;

uint8\_t dataIn=5;

uint8\_t clockEnableIn=4;

//настройки

int regim =3;

int number=1;

int ledRed=8;

int ledYellow=9;

int ledGreen=10;

int vcc=2; //проверка питания

int for\_check=A1;

//инициализация

void setup() {

Serial.begin(9600); //подключение монитора порта

//установка пинов на вход или на выход

//led

pinMode(ledGreen, OUTPUT);

pinMode(ledRed, OUTPUT);

pinMode(ledYellow, OUTPUT);

//165

pinMode(latchIn, OUTPUT);

pinMode(clockIn, OUTPUT);

pinMode(dataIn, INPUT);

pinMode(clockEnableIn, OUTPUT);

//595

pinMode(latchPin, OUTPUT);

pinMode(dataPin, OUTPUT);

pinMode(clockPin, OUTPUT);

//настройки

pinMode(regim, INPUT);

pinMode(vcc, INPUT);

pinMode(for\_check, INPUT);

//начальная установка

digitalWrite(latchPin, LOW);//установить защелку, начать передавать бит

shiftOut(dataPin, clockPin, LSBFIRST, 0);

shiftOut(dataPin, clockPin, LSBFIRST, 0);

digitalWrite(latchPin, HIGH); // сигнализировать, что больше не надо считывать

}

//функция диагностики

void diagnostic()

{

//записывание

digitalWrite(latchPin, LOW);//установить защелку, начать передавать бит

//сдвинуть байт данных по одному биту за раз, записывая на вывод данных

shiftOut(dataPin, clockPin, LSBFIRST, 0);

//так как 16 разрядов - 2 байта, поэтому 2 функции

shiftOut(dataPin, clockPin, LSBFIRST, 0);

digitalWrite(latchPin, HIGH); // сигнализировать, что больше не надо считывать

//устанавить защелку, чтобы захватить состояния выходов

digitalWrite(latchIn, LOW);

delayMicroseconds(5);

digitalWrite(latchIn, HIGH);

delayMicroseconds(5);

//считывание

digitalWrite(clockIn, HIGH); //без этого не считывает последний бит

digitalWrite(clockEnableIn, LOW); //установить запись, начать считывать бит

//сдвигаем байт данных по одному за раз, записывая на ввод данных

uint8\_t in1=shiftIn(dataIn, clockIn, LSBFIRST);

uint8\_t in2=shiftIn(dataIn, clockIn, LSBFIRST);

digitalWrite(clockEnableIn, HIGH); //сигнализировать, что больше не надо считывать

//вывод в монитор порта

Serial.println(in1, DEC);

Serial.println(in2,DEC);

Serial.print('\n');

//записывание

digitalWrite(latchPin, LOW);//установить защелку, начать передавать бит

shiftOut(dataPin, clockPin, LSBFIRST, in2);

shiftOut(dataPin, clockPin, LSBFIRST, in1);

digitalWrite(latchPin, HIGH); // сигнализировать, что больше не надо считывать

delay(10);

}

//функция проверки устройства

void check(byte data, uint8\_t \*in1, uint8\_t \*in2)

{

//записывание

digitalWrite(latchPin, LOW);//установить защелку, начать передавать бит

shiftOut(dataPin, clockPin, LSBFIRST, data);

shiftOut(dataPin, clockPin, LSBFIRST, data);

digitalWrite(latchPin, HIGH); // сигнализировать, что больше не надо считывать

//устанавить защелку, чтобы захватить состояния выходов

digitalWrite(latchIn, LOW);

delayMicroseconds(5);

digitalWrite(latchIn, HIGH);

delayMicroseconds(5);

//считывание

digitalWrite(clockIn, HIGH); //без этого не считывает последний бит

digitalWrite(clockEnableIn, LOW); //устанавить запись, начать считывать бит

\*in1=shiftIn(dataIn, clockIn, LSBFIRST);

\*in2=shiftIn(dataIn, clockIn, LSBFIRST);

digitalWrite(clockEnableIn, HIGH); //сигнализировать, что больше не надо считывать

}

//функция проверки состояний переключателя в режиме проверки

void check\_begining(uint8\_t \*in1, uint8\_t \*in2)

{

//устанавливаем защелку, чтобы захватить состояния выходов

digitalWrite(latchIn, LOW);

delayMicroseconds(5);

digitalWrite(latchIn, HIGH);

delayMicroseconds(5);

//считывание

digitalWrite(clockIn, HIGH); //без этого не считывает последний бит

digitalWrite(clockEnableIn, LOW); //установить запись, начать считывать бит

\*in1=shiftIn(dataIn, clockIn, LSBFIRST);

\*in2=shiftIn(dataIn, clockIn, LSBFIRST);

digitalWrite(clockEnableIn, HIGH); //сигнализировать, что больше не надо считывать

}

//основная программа

void loop() {

uint8\_t in1;

uint8\_t in2;

int s1=digitalRead(regim); //считывание состояние переключателя режима

int s2=digitalRead(vcc); // считывание состояние питания

int s3=digitalRead(for\_check); //считываем состояние переключателя проверки

if (s1==0) //режим диагностики

{

if (s2==0) //не подключено проверяемое устройство

{

digitalWrite(ledRed, LOW);// все диоды потушить

digitalWrite(ledYellow, LOW);

digitalWrite(ledGreen, LOW);

diagnostic(); //диагностика

}

else //проверяемое устройство подключено

{

digitalWrite(ledRed, LOW);

digitalWrite(ledYellow, HIGH); //зажечь желтый диод - диагностика не возможна

digitalWrite(ledGreen, LOW);

}

number=1; //номер проверки =1

}

else //режим проверки

{

//подать нули на входы

digitalWrite(latchPin, LOW);//установить защелку, начать передавать бит

shiftOut(dataPin, clockPin, LSBFIRST, 0);

shiftOut(dataPin, clockPin, LSBFIRST, 0);

digitalWrite(latchPin, HIGH); // сигнализировать, что больше не надо считывать

check\_begining(&in1, &in2); //проверить все ли переключатели, задающие выходы в режиме диагностики, выключены

Serial.println(in1, DEC);

Serial.println(in2,DEC);

Serial.print('\n');

delay(1000);

if ((in1!=0 ||in2!=0) && number==1 || s2==0 ) //переключатели не выключены в момент первой проверки или проверяемое устройство не подключено

{

digitalWrite(ledYellow, HIGH); //зажечь желтый диод - проверка не возможна

}

else //всё норм

{

if (s3==1)//начать проверку

{

switch(number) //номер проверки

{

case 1:

{

digitalWrite(ledGreen, LOW);

digitalWrite(ledYellow, LOW);

digitalWrite(ledGreen, LOW);

check(0, &in1, &in2);//проверка

//вывод в монитор порта

Serial.println(in1, DEC);

Serial.println(in2,DEC);

Serial.print('\n');

if (in1==0 && in2==0)//сравнить полученные значения с эталонными

{

digitalWrite(ledGreen, HIGH); //проверка прошла

number++; //следующая проверка

} //if

else

{

digitalWrite(ledRed, HIGH);//не исправно

number=5;//заканчиваем проверку

}//else

delay(1000);

break;

}//1

case 2:

{

digitalWrite(ledGreen, LOW);

digitalWrite(ledYellow, LOW);

digitalWrite(ledGreen, LOW);

delay(1000);

check(51, &in1, &in2);//проверка

//вывод в монитор порта

Serial.println(in1, DEC);

Serial.println(in2,DEC);

Serial.print('\n');

if (in1==51 && in2==51 )

{

digitalWrite(ledGreen, HIGH);

number++;

} //if

else

{

digitalWrite(ledRed, HIGH);

number=5;

}//else

delay(1000);

break;

}//2

case 3:

{

digitalWrite(ledGreen, LOW);

digitalWrite(ledYellow, LOW);

digitalWrite(ledGreen, LOW);

delay(1000);

check(204, &in1, &in2);//проверка

//вывод в монитор порта

Serial.println(in1, DEC);

Serial.println(in2,DEC);

Serial.print('\n');

if (in1==136 && in2==136 )

{

digitalWrite(ledGreen, HIGH);

number++;

} //if

else

{

digitalWrite(ledRed, HIGH);

number=5;

}//else

delay(1000);

break;

}//3

case 4:

{

digitalWrite(ledGreen, LOW);

digitalWrite(ledYellow, LOW);

digitalWrite(ledGreen, LOW);

delay(1000);

check(255, &in1, &in2);//проверка

//вывод в монитор порта

Serial.println(in1, DEC);

Serial.println(in2,DEC);

Serial.print('\n');

if (in1==221 && in2==221 )

{

digitalWrite(ledGreen, HIGH);

number++;

} //if

else

{

digitalWrite(ledRed, HIGH);

number=5;

}//else

delay(1000);

break;

}

default:

{

//записывание

digitalWrite(latchPin, LOW);//установить защелку, начинать передавать бит

shiftOut(dataPin, clockPin, LSBFIRST, 0); //сброс входов

shiftOut(dataPin, clockPin, LSBFIRST, 0);

digitalWrite(latchPin, HIGH); // сигнализировать, что больше не надо считывать

break;

}

}//switch

} // готовы к проверке

else //сброс

{

number=1;

digitalWrite(ledGreen, LOW);

digitalWrite(ledYellow, LOW);

digitalWrite(ledRed, LOW);

}

}//питание

} // regim = check

}