//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

}