

```
#include <WiFi.h>
#include <Arduino.h>
#include <stdint.h>
#include "SCMD.h"
#include "SCMD_config.h" //Contains #defines for common SCMD register names and values.
#include "Wire.h"
#include <ESP32Servo.h>
```

```
Servo myServo;
```

```
SCMD myMotorDriver; //This creates the main object of one motor driver and connected peripherals.
```

```
const char* ssid   = "RedmiN";
const char* password = "12345678";
```

```
WiFiServer server(80); // Port 80
```

```
const int redled = 6;
const int blueled = 8;
const int greenled = 10;
const int Ylow = 4;
const int Yhigh = 37;
const int Xlow = 35;
const int Xhigh = 13;
const int servo = 34;
const int hallSensorPin = 11;
const int pushbutton = 33;
int lmt1read=1;
int lmt2read=1;
int lmt3read=1;
int lmt4read=1;
int pen_up=0;
```

```

int pos=0;

int emerg=0;

// bool notfound = true;

String state = "Pen plotter is ready";

String coordinates;

String x, y;

int x_coord,y_coord;

unsigned long start=millis();

// int pxtt=500; int ntxt=500; int pytt=525; int nytt=550; int nett=580; int nwtt=320; int sett=570; int
swtt=320;

int wait30 = 30000; // time to reconnect when connection is lost.


// void pos_x();
// void neg_x();
// void pos_y();
// void neg_y();
// void ne();//north east
// void nw();
// void se();
// void sw();
// void stp();
// void home();
// void origin();


void setup() {
  Serial.begin(9600);

  pinMode(Ylow, INPUT_PULLUP);
  pinMode(Yhigh, INPUT_PULLUP);
  pinMode(Xlow, INPUT_PULLUP);
  pinMode(Xhigh, INPUT_PULLUP);

```

```
myServo.attach(servo);

pinMode(hallSensorPin, INPUT);

pinMode(blueled, OUTPUT);

pinMode(redled, OUTPUT);

pinMode(greenled, OUTPUT);

pinMode(pushbutton, INPUT_PULLUP);


randomSeed(analogRead(A0));


// Connect WiFi net.

Serial.println();

Serial.print("Connecting with ");

Serial.println(ssid);

WiFi.begin(ssid, password);


while (WiFi.status() != WL_CONNECTED)
{
    delay(1000);
    Serial.print(".");
}

Serial.println("Connected with WiFi.");


// Start Web Server.

server.begin();

Serial.println("Web Server started.");


Serial.print("This is IP to connect to the WebServer: ");

Serial.print("http://");

Serial.println(WiFi.localIP());
```

```
myMotorDriver.settings.commInterface = I2C_MODE;
```

```
// set address if I2C configuration selected with the config jumpers
```

```
myMotorDriver.settings.I2CAddress = 0x5D; //config pattern is "1000" (default) on board for address 0x5D
```

```
// set chip select if SPI selected with the config jumpers
```

```
myMotorDriver.settings.chipSelectPin = 10;
```

```
Serial.println(myMotorDriver.begin());
```

```
// **initialize the driver get wait for idle**//
```

```
while ( myMotorDriver.begin() != 0xA9 ) //Wait until a valid ID word is returned
```

```
{
```

```
    Serial.println( "ID mismatch, trying again" );
```

```
    delay(1000);
```

```
}
```

```
Serial.println( "ID matches 0xA9" );
```

```
Serial.println(myMotorDriver.ready());
```

```
// Check to make sure the driver is done looking for peripherals before beginning
```

```
Serial.println("Waiting for enumeration...");
```

```
while ( myMotorDriver.ready() == false );
```

```
Serial.println("Done.");
```

```
Serial.println();
```

```
while ( myMotorDriver.busy() );
```

```
myMotorDriver.enable(); //Enables the output driver hardware
```

```
Serial.println( "not busy" );
```

```
#define LEFT_MOTOR 0
```

```
#define RIGHT_MOTOR 1
```

```
}
```

```
void blueledfastblink()
```

```
{
```

```
  digitalWrite(blueled, HIGH);
```

```
  delay(50);
```

```
  digitalWrite(blueled, LOW);
```

```
  delay(50);
```

```
}
```

```
void blueledblink()
```

```
{
```

```
  digitalWrite(blueled, HIGH);
```

```
  delay(500);
```

```
  digitalWrite(blueled, LOW);
```

```
  delay(500);
```

```
}
```

```
void Pen_Up()
```

```
{
```

```
  for (int pos = 0; pos <= 90; pos += 1)
```

```
  {
```

```
    blueledfastblink();
```

```
    myServo.write(pos);
```

```
    delay(15);
```

```
  }
```

```
  Serial.println("pen up");
```

```
  // penUp = true;
```

```
}
```

```
void Pen_Down()
```

```
{
```

```
  for (int pos = 90; pos >= 0; pos -= 1)
```

```
  {
```

```
    blueledfastblink();
```

```
    myServo.write(pos);
```

```
    delay(15);
```

```
  }
```

```
  Serial.println("pen down");
```

```
  // pendown = true;
```

```
}
```

```
void blueledglow()
```

```
{
```

```
  digitalWrite(blueled, HIGH);
```

```
}
```

```
void nw() {
```

```
  // Perform the motor actions
```

```
  myMotorDriver.setDrive( LEFT_MOTOR, 0, 207); //nw dir
```

```
  myMotorDriver.setDrive( RIGHT_MOTOR, 0, 200);
```

```
}
```

```
void pos_y(){
```

```
  myMotorDriver.setDrive( RIGHT_MOTOR, 0, 200);
```

```
}
```

```
void neg_y(){
```

```
  myMotorDriver.setDrive( RIGHT_MOTOR, 1, 200);
```

```
}
```

```
void se(){
```

```
    myMotorDriver.setDrive( LEFT_MOTOR, 1, 200); //se dir
```

```
    myMotorDriver.setDrive( RIGHT_MOTOR, 1, 210);
```

```
}
```

```
void sw(){
```

```
    myMotorDriver.setDrive(LEFT_MOTOR, 0, 235); // sw dir
```

```
    myMotorDriver.setDrive(RIGHT_MOTOR, 1, 200);
```

```
}
```

```
void neg_x(){
```

```
    myMotorDriver.setDrive( LEFT_MOTOR, 0, 200); // -x dir
```

```
}
```

```
void pos_x(){
```

```
    myMotorDriver.setDrive( LEFT_MOTOR, 1, 200); // +x direction
```

```
}
```

```
void stp(){
```

```
    myMotorDriver.setDrive( LEFT_MOTOR, 0, 0); //Stop motor
```

```
    myMotorDriver.setDrive( RIGHT_MOTOR, 0, 0);
```

```
}
```

```
void ne(){
```

```
    myMotorDriver.setDrive( LEFT_MOTOR, 1, 190); //ne dir
```

```
myMotorDriver.setDrive( RIGHT_MOTOR, 0, 200);
```

```
}
```

```
void origin(){
```

```
  if(pen_up != 1){
```

```
    Pen_Up();
```

```
    pen_up=1;}
```

```
  int lmt1state = digitalRead(Xlow);
```

```
  int lmt2state = digitalRead(Ylow);
```

```
  while(lmt1state==0){
```

```
    neg_x();
```

```
    lmt1state = digitalRead(Xlow);
```

```
    delay(5);
```

```
  }
```

```
  Serial.println( "left" );
```

```
  stp();delay(600);
```

```
  while(lmt2state==0){
```

```
    neg_y();
```

```
    lmt2state = digitalRead(Ylow);
```

```
    delay(5);}
```

```
  Serial.println( "bottom" );
```

```
  stp();delay(600);}
```

```
void move_to_home()
```

```
{ for(int h=0;h<1;h++){
```

```
  // if(pen_up != 1){
```

```
  // Pen_Up();
```

```
  // pen_up=1;}
```



```

// delay(1000);

while
(digitalRead(Xlow)*digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRead(pushbutton)
==1)
{
    myMotorDriver.setDrive(LEFT_MOTOR, 0, 150);

}

if (digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRead(pushbutton)==0)
{
    break;
}

myMotorDriver.setDrive(LEFT_MOTOR, 1, 0);
delay(1000);

while (digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRead(pushbutton)==1)
{
    myMotorDriver.setDrive(RIGHT_MOTOR, 1, 150);

}

if (digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRead(pushbutton)==0)
{
    break;
}

myMotorDriver.setDrive(RIGHT_MOTOR, 1, 0);
delay(1000);

while (digitalRead(Ylow) == LOW || digitalRead(Xlow) == LOW &&
digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRead(pushbutton)==1)
{
    int sensorValue = digitalRead(hallSensorPin);

    if (sensorValue == LOW)

```

```

{
    while (sensorValue != HIGH &&
digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRead(pushbutton)==1)
    {
        blueledblink();
        myMotorDriver.setDrive(LEFT_MOTOR, 1, 60);
        Serial.println("Magnetic field not detected. moving to home position");
        sensorValue = digitalRead(hallSensorPin);
    }

    if (digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRead(pushbutton)==0)
    {
        break;
    }

    myMotorDriver.setDrive(LEFT_MOTOR, 1, 0);
    Serial.println("Magnetic field detected. At home position");
    unsigned long startTime = millis();
    Serial.print("start TIME = ");
    Serial.print(startTime);
    // elapsedTimeMoveToHome = startTime;
}

else if (sensorValue == HIGH)
{
    start= millis(); while((millis()-start<150) && digitalRead(pushbutton)==1 &&
digitalRead(Yhigh)==1)
    {myMotorDriver.setDrive(RIGHT_MOTOR, 0, 200);
        Serial.println("moving to start nicholas house");
        delay(500);
    }

    if (digitalRead(Yhigh)*digitalRead(pushbutton)==0)

```

```

    {
        break;
    }

    myMotorDriver.setDrive(RIGHT_MOTOR, 0, 0);
    delay(500);
}
}

// elapsedTimeMoveToHome = millis() - elapsedTimeMoveToHome + 1000;
// Serial.print("ELAPSED TIME = ");
// Serial.print(elapsedTimeMoveToHome);
}

myMotorDriver.setDrive(LEFT_MOTOR, 1, 0);
myMotorDriver.setDrive(RIGHT_MOTOR, 1, 0);
delay(100);
}

void loop() {
    if ((WiFi.status() != WL_CONNECTED) && (millis() > wait30)) {
        Serial.println("Trying to reconnect WiFi...");
        WiFi.disconnect();
        WiFi.begin(ssid, password);
        wait30 = millis() + 30000;
    }

    // Check if a client has connected..
    WiFiClient client = server.available();
    if (!client) {
        return;
    }
}

```

```
Serial.print("New client: ");
```

```
Serial.println(client.remoteIP());
```

```
// while(!client.available()){ delay(1); }
```

```
// Read the information sent by the client.
```

```
String req = client.readStringUntil('\r');
```

```
Serial.println(req);
```

```
// for manual movement.
```

```
if (req.indexOf("mmne") != -1)
```

```
{if(pen_up != 1){
```

```
Pen_Up();
```

```
pen_up=1;}
```

```
start= millis(); while((millis()-start<200) && (digitalRead(Ylow)==1) &&  
(digitalRead(Ylow)==1)){ne();}
```

```
stp();
```

```
}
```

```
if (req.indexOf("mmnw") != -1)
```

```
{if(pen_up != 1){
```

```
Pen_Up();
```

```
pen_up=1;}
```

```
start= millis(); while((millis()-start<200) && (digitalRead(Ylow)==1)&&  
(digitalRead(Ylow)==1)){nw();}
```

```
stp();
```

```
}
```

```
// if (req.indexOf("mmn") != -1)
```

```
// {start= millis(); while((millis()-start<1) && (digitalRead(lmt3)==0)){pos_y();}  
//  stp();  
// }
```

```
if (req.indexOf("mmnn") != -1)  
{  
  // if(pen_up != 1){  
  //  Pen_Up();  
  //  pen_up=1;}  
  start= millis(); while((millis()-start<200) && (digitalRead(Yhigh)==1)){pos_y();}  
  stp();  
  // Pen_Up();  
}
```

```
if (req.indexOf("mmsw") != -1)  
{  
  // if(pen_up != 1){  
  //  Pen_Up();  
  //  pen_up=1;}  
  start= millis(); while((millis()-start<200) && (digitalRead(Ylow)==1) &&  
(digitalRead(Ylow)==1)){sw();}  
  stp();  
}
```

```
if (req.indexOf("mmse") != -1)  
{  
  // if(pen_up != 1){  
  //  Pen_Up();  
  //  pen_up=1;}  
  start= millis(); while((millis()-start<200) && (digitalRead(Ylow)==1) &&  
(digitalRead(Ylow)==1)){se();}
```

```
stp();
```

```
}
```

```
// if (req.indexOf("mms") != -1)
```

```
// {start= millis(); while((millis()-start<1) && (digitalRead(lmt4)==0)){neg_y();}
```

```
// stp();
```

```
// }
```

```
if (req.indexOf("mmss") != -1)
```

```
{
```

```
// if(pen_up != 1){
```

```
// Pen_Up();
```

```
// pen_up=1;}
```

```
start= millis(); while((millis()-start<200) && (digitalRead(Ylow)==1)){neg_y();}
```

```
stp();
```

```
// Pen_Down();
```

```
}
```

```
if (req.indexOf("mme") != -1)
```

```
{
```

```
// if(pen_up != 1){
```

```
// Pen_Up();
```

```
// pen_up=1;}
```

```
start= millis(); while((millis()-start<200) && (digitalRead(Xhigh)==1)){pos_x();}
```

```
stp();
```

```
}
```

```
if (req.indexOf("mmw") != -1)
```

```
{
```

```
// if(pen_up != 1){
```

```
// Pen_Up();
```

```

// pen_up=1;}

start= millis(); while((millis()-start<200) && (digitalRead(Xlow)==1)){neg_x();}

stp();

}

```

```

// for home position

if (req.indexOf("home") != -1)
{
    move_to_home();
    Serial.println("HOME POSITION");
    state="home position";
}

```

```

// drawing Nikolaus haus

if (req.indexOf("draw") != -1 )
{
    // if(pen_up == 1){
    // Pen_Down();
    // pen_up=0;}

    for(int plot=0;plot<1;plot++){
        Serial.println("executing pos_x");

        // home();pos_x();delay(500);stp();delay(600);pos_y();delay(400);stp();delay(600);

        start= millis(); while((millis()-start<500) &&
(digitalRead(Xlow)*digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRead(pushbutton)
)==1){pos_x();if(digitalRead(Xlow)*digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digital
Read(pushbutton)==0){break;}stp(); delay(1500);

        start= millis(); while((millis()-start<525) &&
(digitalRead(Xlow)*digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRead(pushbutton)
)==1){pos_y();if(digitalRead(Xlow)*digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digital
Read(pushbutton)==0){break;}stp(); delay(1500);

```

```

    start= millis(); while((millis()-start<500) &&
(digitalRead(Xlow)*digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRead(pushbutton
)==1)){neg_x();}if(digitalRead(Xlow)*digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digital
Read(pushbutton)==0){break;}stp(); delay(1500);

```

```

    start= millis(); while((millis()-start<520) &&
(digitalRead(Xlow)*digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRead(pushbutton
)==1)){neg_y();}if(digitalRead(Xlow)*digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digital
Read(pushbutton)==0){break;}stp(); delay(1500);

```

```

    start= millis(); while((millis()-start<650) &&
(digitalRead(Xlow)*digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRead(pushbutton
)==1)){ne();}if(digitalRead(Xlow)*digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRea
d(pushbutton)==0){break;}stp(); delay(1500);

```

```

    start= millis(); while((millis()-start<320) &&
(digitalRead(Xlow)*digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRead(pushbutton
)==1)){nw();}if(digitalRead(Xlow)*digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRea
d(pushbutton)==0){break;}stp(); delay(1500);

```

```

    start= millis(); while((millis()-start<320) &&
(digitalRead(Xlow)*digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRead(pushbutton
)==1)){sw();}if(digitalRead(Xlow)*digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRea
d(pushbutton)==0){break;}stp(); delay(1500);

```

```

    start= millis(); while((millis()-start<580) &&
(digitalRead(Xlow)*digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRead(pushbutton
)==1)){se();}if(digitalRead(Xlow)*digitalRead(Ylow)*digitalRead(Xhigh)*digitalRead(Yhigh)*digitalRea
d(pushbutton)==0){break;}stp(); delay(1500);

```

```

    state = "Nikolaus Haus completed";}

```

```

    Serial.println("out of loop");

```

```

    stp(); delay(100);

```

```

    // if(pen_up != 1){

```

```

    // Pen_Up();

```

```

    // pen_up=1;}

```

```

    }

```

```

    // // set coordinates

```

```

    // if (req.indexOf("+") != -1 ){

```

```

    // req.replace("+", " ");

```



```

// req.replace(" HTTP/1.1", "");
// req.replace("GET /", "");
// coordinates=req;
// Serial.println(coordinates);
// int spaceIndex = coordinates.indexOf(' ');

// if (spaceIndex != -1) {
//   home();
//   x = coordinates.substring(0, spaceIndex);
//   y = coordinates.substring(spaceIndex + 1);
//   x_coord = x.toInt();
//   y_coord = y.toInt();
//   Serial.print("x_coord: ");
//   Serial.println(x_coord);
//   Serial.print("y_coord: ");
//   Serial.println(y_coord);
//   int edelay=x_coord*(200/1.5);
//   int ndelay=y_coord*(200/1.5);
//   pos_x();delay(edelay);stp();delay(500);
//   pos_y();delay(ndelay);stp();delay(500);

// }

// }

// if (req.indexOf("defspeed") != -1 ){
//   speeds();timings();
//   state="Default speed set";
// }

// if (req.indexOf("speed") != -1 ){

```

```

// req.replace("speed", "");
// req.replace(" HTTP/1.1", "");
// req.replace("GET /", "");
// speed=req.toInt();
// Serial.print("Speed is set to ");
// Serial.println(speed);
// px=speed; nx=speed; py=speed; ny=speed; nes=speed; nws=speed; ses=speed; sws=speed;
// // pxt=(300/225)*speed; nxt=(300/225)*speed; pyt=(300/225)*speed; nyt=(300/225)*speed;
// // net=(300/225)*speed; nwt=(300/225)*speed; set=(300/225)*speed; swt=(300/225)*speed;
// state=req;
// }

```

```

// if (req.indexOf("pxt") != -1 ){
// req.replace("pxt", "");
// req.replace(" HTTP/1.1", "");
// req.replace("GET /", "");
// pxtt=req.toInt();
// Serial.print("pxt is: ");
// Serial.println(pxtt);
// state=req;
// }

```

```

// if (req.indexOf("nxt") != -1 ){
// req.replace("nxt", "");
// req.replace(" HTTP/1.1", "");
// req.replace("GET /", "");
// nxtt=req.toInt();
// Serial.print("nxt is: ");
// Serial.println(nxtt);
// state=req;
// }

```

```
// if (req.indexOf("pyt") != -1 ){  
// req.replace("pyt", "");  
// req.replace(" HTTP/1.1", "");  
// req.replace("GET /", "");  
// pytt=req.toInt();  
// Serial.print("pyt is: ");  
// Serial.println(pytt);  
// state=req;  
// }
```

```
// if (req.indexOf("nyt") != -1 ){  
// req.replace("nyt", "");  
// req.replace(" HTTP/1.1", "");  
// req.replace("GET /", "");  
// nytt=req.toInt();  
// Serial.print("nyt is: ");  
// Serial.println(nytt);  
// state=req;  
// }
```

```
// if (req.indexOf("net") != -1 ){  
// req.replace("net", "");  
// req.replace(" HTTP/1.1", "");  
// req.replace("GET /", "");  
// nett=req.toInt();  
// Serial.print("net is: ");  
// Serial.println(nett);  
// state=req;  
// }
```

```
// if (req.indexOf("nwt") != -1 ){  
// req.replace("nwt", "");  
// req.replace(" HTTP/1.1", "");  
// req.replace("GET /", "");  
// nwtt=req.toInt();  
// Serial.print("nwt is: ");  
// Serial.println(nwtt);  
// state=req;  
// }
```

```
// if (req.indexOf("swt") != -1 ){  
// req.replace("swtt", "");  
// req.replace(" HTTP/1.1", "");  
// req.replace("GET /", "");  
// swtt=req.toInt();  
// Serial.print("swt is: ");  
// Serial.println(swtt);  
// state=req;  
// }
```

```
// if (req.indexOf("swr") != -1 ){  
// req.replace("swtt", "");  
// req.replace(" HTTP/1.1", "");  
// req.replace("GET /", "");  
// swtt=req.toInt();  
// Serial.print("swt is: ");  
// Serial.println(swtt);  
// state=req;  
// }
```

```
// if (req.indexOf("ipset") != -1 ){
```

```
// Serial.println("Connected to HMI");
// state="Pen plotter is ready";
// }

// WEB PAGE. //////////////////////////////////////

client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("");
client.println(state);

Serial.print("Client disconnected: ");
Serial.println(client.remoteIP());
client.flush();
client.stop();
if ((state == "Nikolaus Haus completed") || (state == "home position")){
    state="Pen plotter is ready";
}
}
```