

Controller

Team Knightmares

September 8, 2022

Abstract

Making a Bot Controller using ESP32

1 Introduction

We made this controller for XLR8 (IIT Bombay's Self Bot racing event) We first tried controlling our bot using MIT app inventor and putting URL in every button but it was too tedious plus we couldn't interchange between two buttons fast. Whole circuitry and Flow of the code is explained here. You can get the code from our github repository and there is link to all the components required to make this bot.

[help library](#), or head to our plans page to [choose your plan](#).

2 Hardware

2.1 Components

Components used:

1. ESP32 DEVKIT V1 [Buy](#)
2. Analog Joystick (HW-504) [Buy](#)
3. Potentiometer (Any one which works of 3 pins) [Buy](#)
4. Perforated Board and soldering stuff [Buy](#)

2.2 Connections

1. Power for esp: We took power directly from MicroUSB port and attached a portable charger on the controller
2. Power for joystick: Take 3.3 V from ESP32
3. Connect Ground to ESP32 GND
4. Connect X-Axis to Pin 34
5. Connect Y-Axis to Pin 35
6. Potentiometer : Connect 2 extreme (GND and Power) to ESP's Ground and Power respectively. Connect variable potential pin to ESP pin 32

3 Flow of Code

Read this for basic understanding so you can alter the code and customize it. Code has a lot of comment to help.

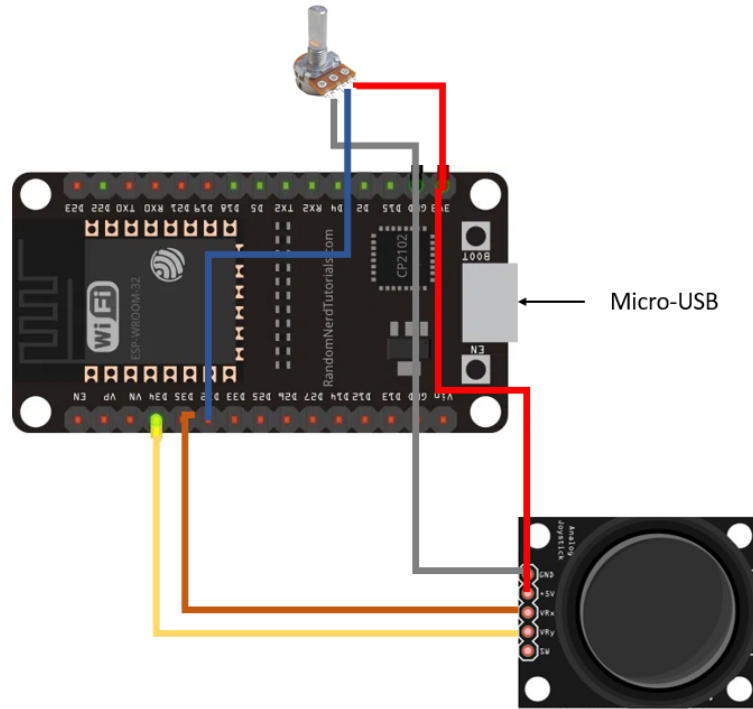


Figure 1: Connections

3.1 Interfacing of Analog Joysticks and Potentiometer

Analog Joystick is a potentiometer which gives variable potential at different positions. Note that ESP and joystick should have same positive and ground voltage to function properly. Similarly for Potentiometer We take this analog input through pin 34 and 35 (x,y). ESP has a built in ADC which gives 0 to 255 (8 bit value) corresponding to 0 to 3.3V voltage of analog pin. Hence we use `analogRead(jpin no.)` for taking in x and y values and then map it accordingly.

Used as global variables

```
int JoyStick_X = 35; // Analog Pin  X
int JoyStick_Y = 34; // // Analog Pin  Y
int SpeedPin = 32 ; // Potentiometer variable voltage pin
```

For defining the pins as input, defined in void setup

```
pinMode(JoyStick_X,INPUT);
pinMode(JoyStick_Y,INPUT);
pinMode(SpeedPin,INPUT);
```

For taking in inputs in every loop.

```
int x = analogRead(JoyStick_X); // X
int y = analogRead(JoyStick_Y); // Y
int speedraw = analogRead(SpeedPin);
```

3.2 Data to be sent

ESP-Now is a WiFi communication between 2 or more ESP's which has one Transmitter and Other Receivers. For our model we use one transmitter and one reciever which is simpler to understand. You create a structure or an object which contains different variables which you want to transmit using ESP-now. In our project we had 4 variables

1. int x coordinate which gives analog position of our joystick in X direction
2. int y coordinate which gives analog position of our joystick in Y direction
3. int cmd this is like an operation mode where each integer value from 0 to 8 has different function and make the bot move in 8 different directions.
4. int spid gives the amount of PWM to give for our moterdrivers (Scaled from 0 to 5 to 0 to 255 in the bot's esp)

Structure defining and declaration

```
typedef struct struct_message {

    int x,y,cmd;
    int spid;

} struct_message;

// Create a structured object
struct_message myData;
```

You have to define the same structure in receiver's code too.
In every loop we have to update the structure variables for it to send

```
// Format structured data
myData.x = x;
myData.y = y;
myData.cmd = cmd();
myData.spid=map(speedraw,0,4095,0,5);;
```

3.3 Funtion cmd()

Due to clipping we needed to give direction to the bot using joysticks clipped output so we had to create an function which takes in x and y values and returns corresponding 0 to 8 integer value.

```
int cmd()
{
    int valX=digi(analogRead(34));
    int valY = digi(analogRead(35));
    // put your main code here, to run repeatedly:
    Serial.print("X: ");
    Serial.print(valX);
    Serial.print(" Y: ");
    Serial.print(valY);
    Serial.println();
    if ((valX==1)and (valY==1)){
        return 0;
    }
    if ((valX==1)and (valY==2)){
        return 1;
    }
    if ((valX==2)and (valY==2)){
        return 2;
    }
    if ((valX==2)and (valY==1)){
```

```

        return 3;
    }
    if ((valX==2)and (valY==0)){
        return 4;
    }
    if ((valX==1)and (valY==0)){
        return 5;
    }
    if ((valX==0)and (valY==0)){
        return 6;
    }
    if ((valX==0)and (valY==1)){
        return 7;
    }
    if ((valX==0)and (valY==2)){
        return 8;
    }
}

int digi(int analval){
    if ( analval < 200) return 0;
    if((analval >1500)and (analval<2000)) return 1;
    if (analval >3800) return 2;
    else return 3;
}

```

3.4 ESP-Now

[Use this to get a basic understanding of ESP-Now](#) Once you went through this page you can see how we customised the code for transmitting it our data.