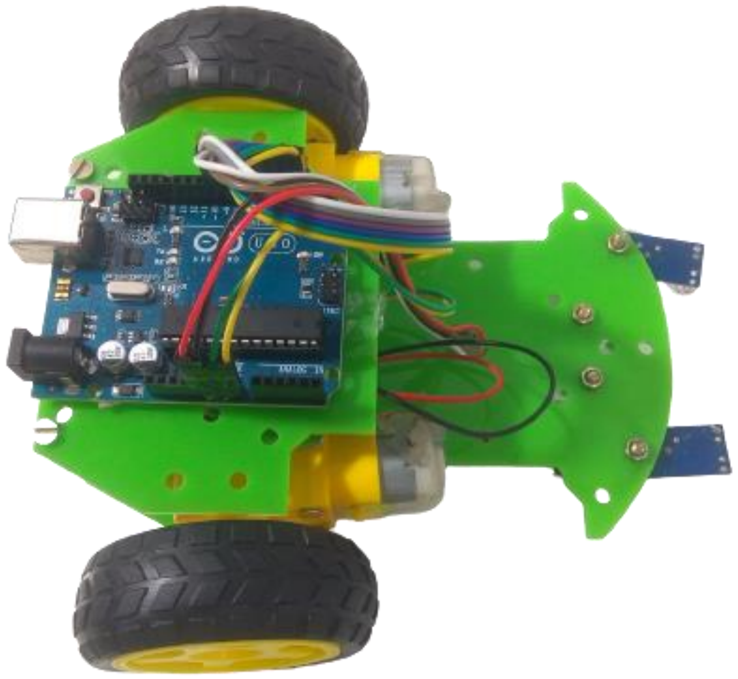
Line following manual

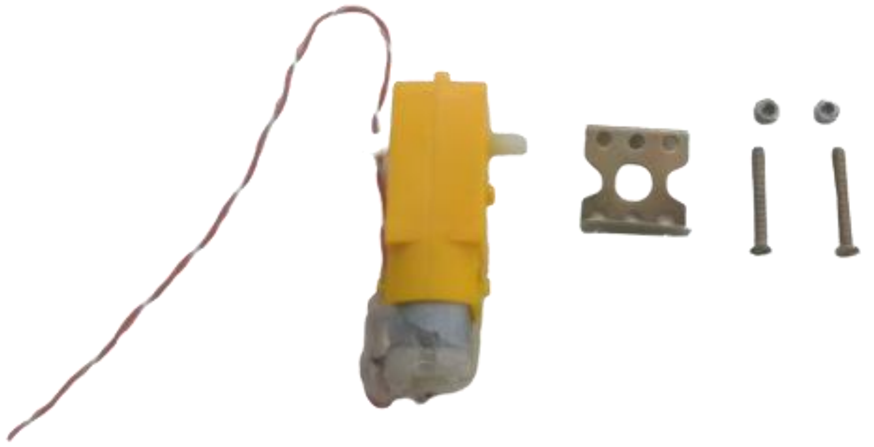


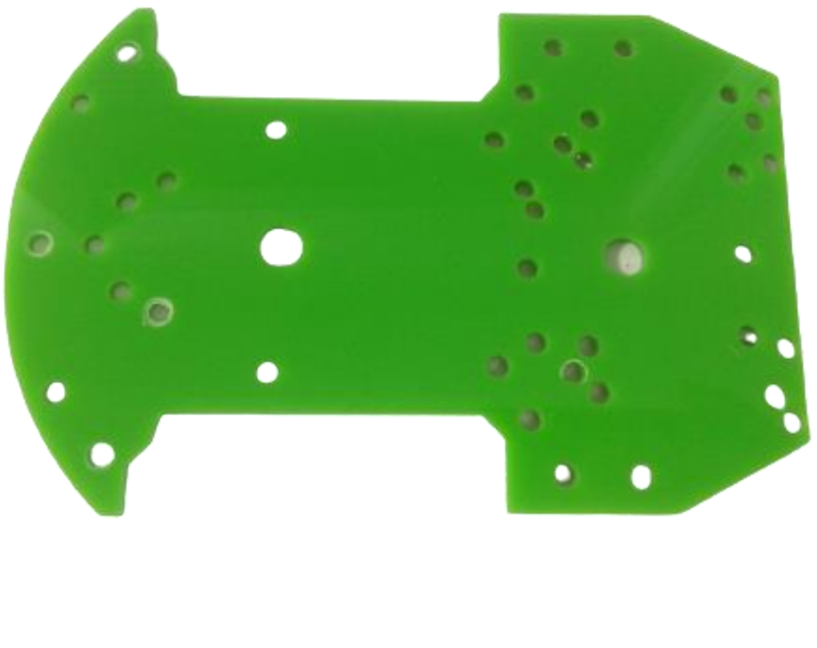
Assembly wiring and programming of Line Following.

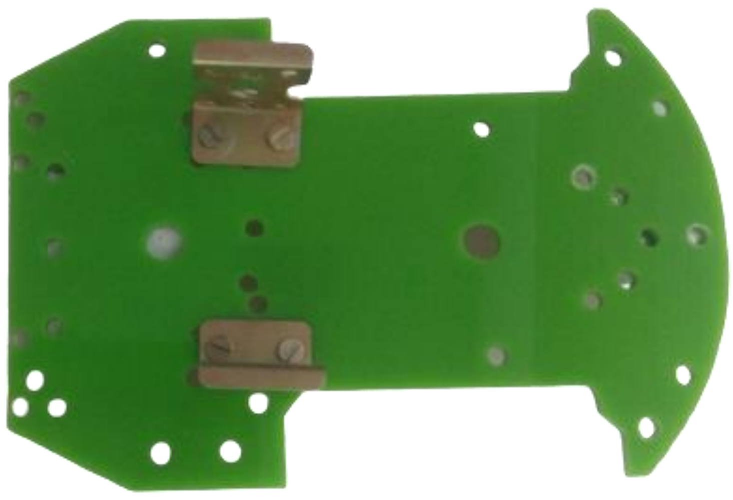
Step 1.

Take the base frame of line following robot.

Assemble the BO Motor mounting clamp on the base of the line following robot.

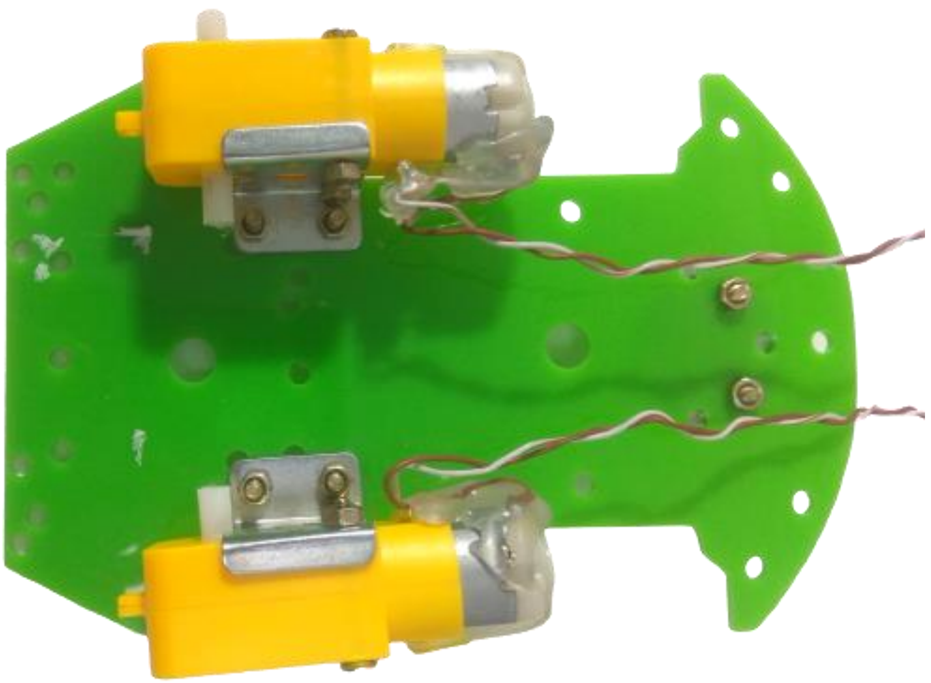






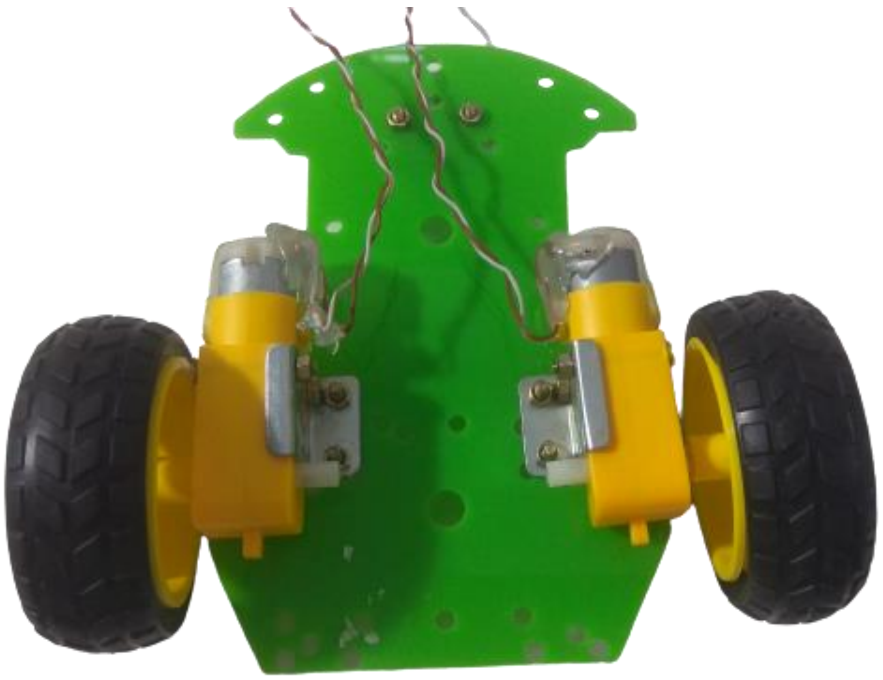
Step 2.

Take the BO Motor and assemble it with BO Motor mounting clamp.



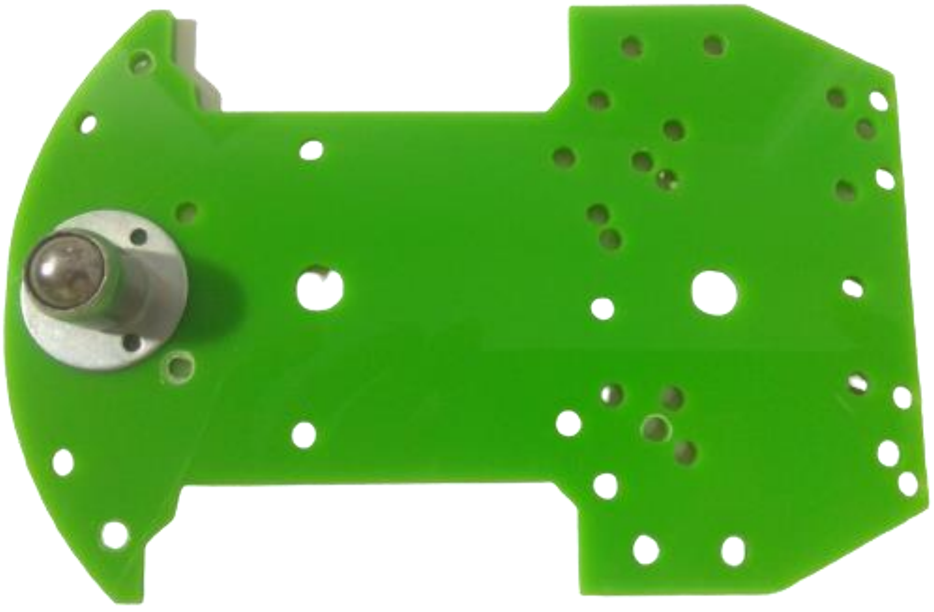
Step 3.

Fix the BO wheel with the BO Motor.



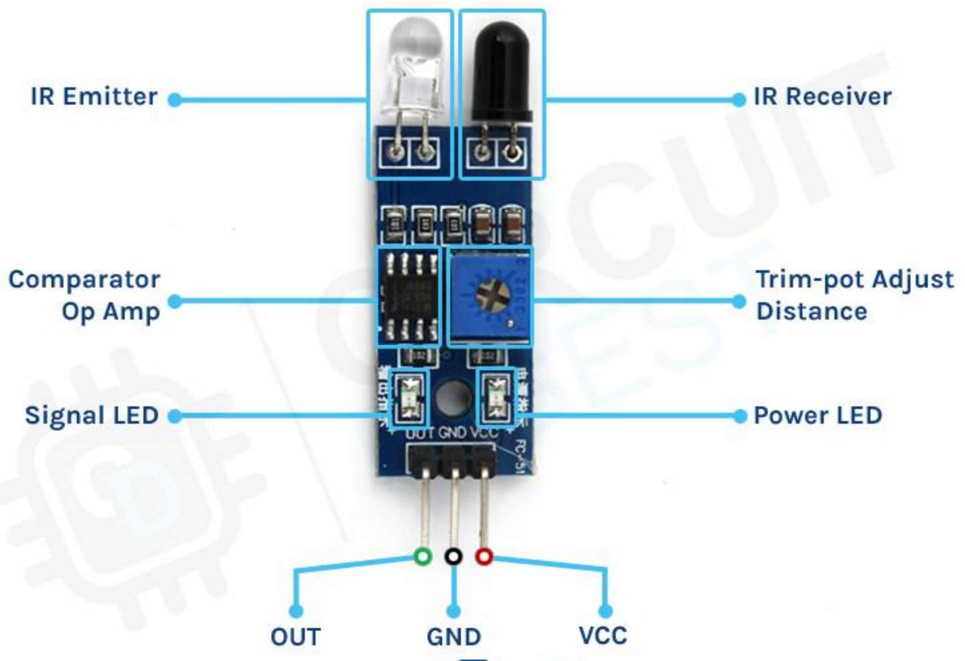
Step 4.

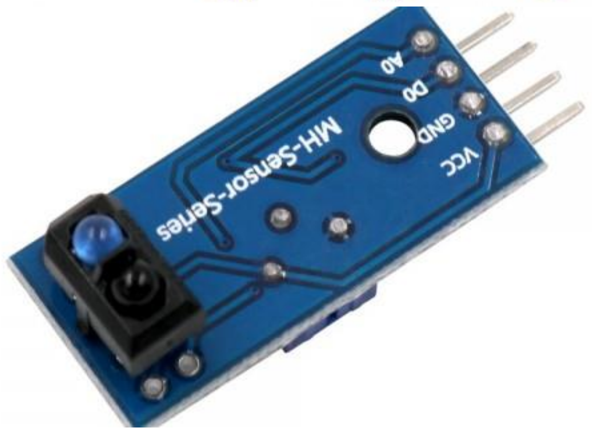
Now assemble the castor wheel on the front side of the robot from bottom side

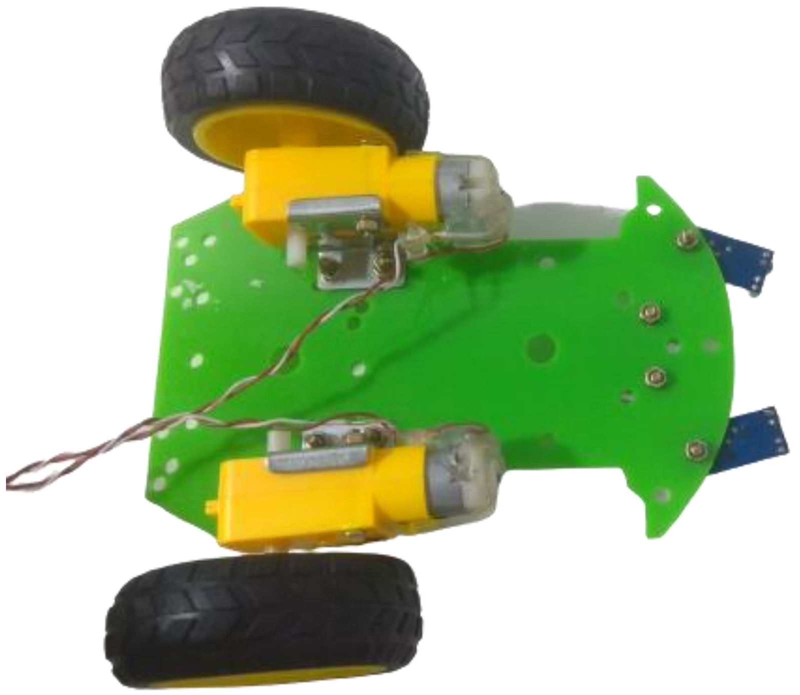


Step 6.

Not take 2 IR sensor module and fix it on both front side of line following robot. Keep the IR sensor led towards the surface.

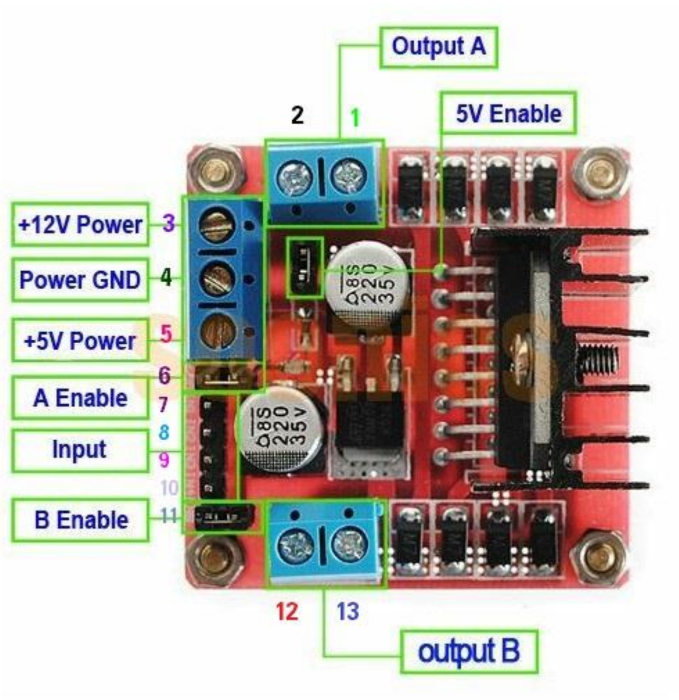

IR Emitter
IR Receiver
Comparator Op Amp
Trim-pot Adjust Distance
Signal LED
Power LED
OUT GND VCC
OUT
GND
VCC


A0
D0
GND
VCC
MH-Sensor-Series



Step 7.

Now take the L298N motor driver and place it and stick it with dual sided tape.



Pin 1 - Green Wire (A+) Stepper Motor

Pin 2 - Black Wire (A-) Stepper Motor

Pin 3 - 12 V Adaptor (+ve lead)

Pin 4 - Adaptor Ground)

Pin 5 - (Nothing)

Pin 6 - Jumper Connected

Pin 7 - To Arduino Uno Pin 8

Pin 8 - To Arduino Uno Pin 9

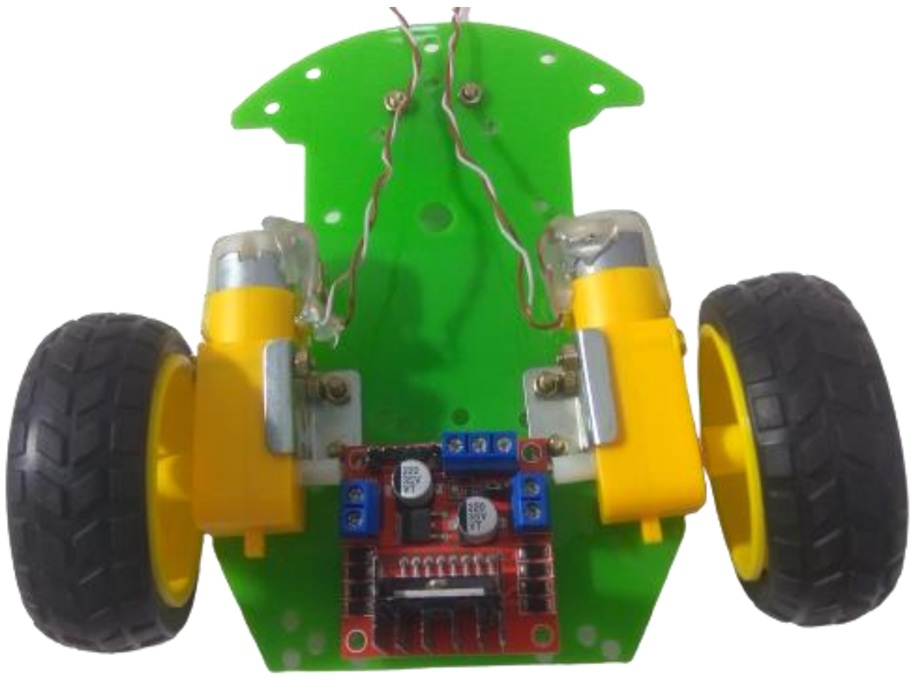
Pin 9 - To Arduino Uno Pin 10

Pin 10 - To Arduino Uno Pin 11

Pin 11 - Jumper Connected

Pin 12 - Red Wire (B+) Stepper Motor

Pin 13 - Blue Wire (B-) Stepper Motor



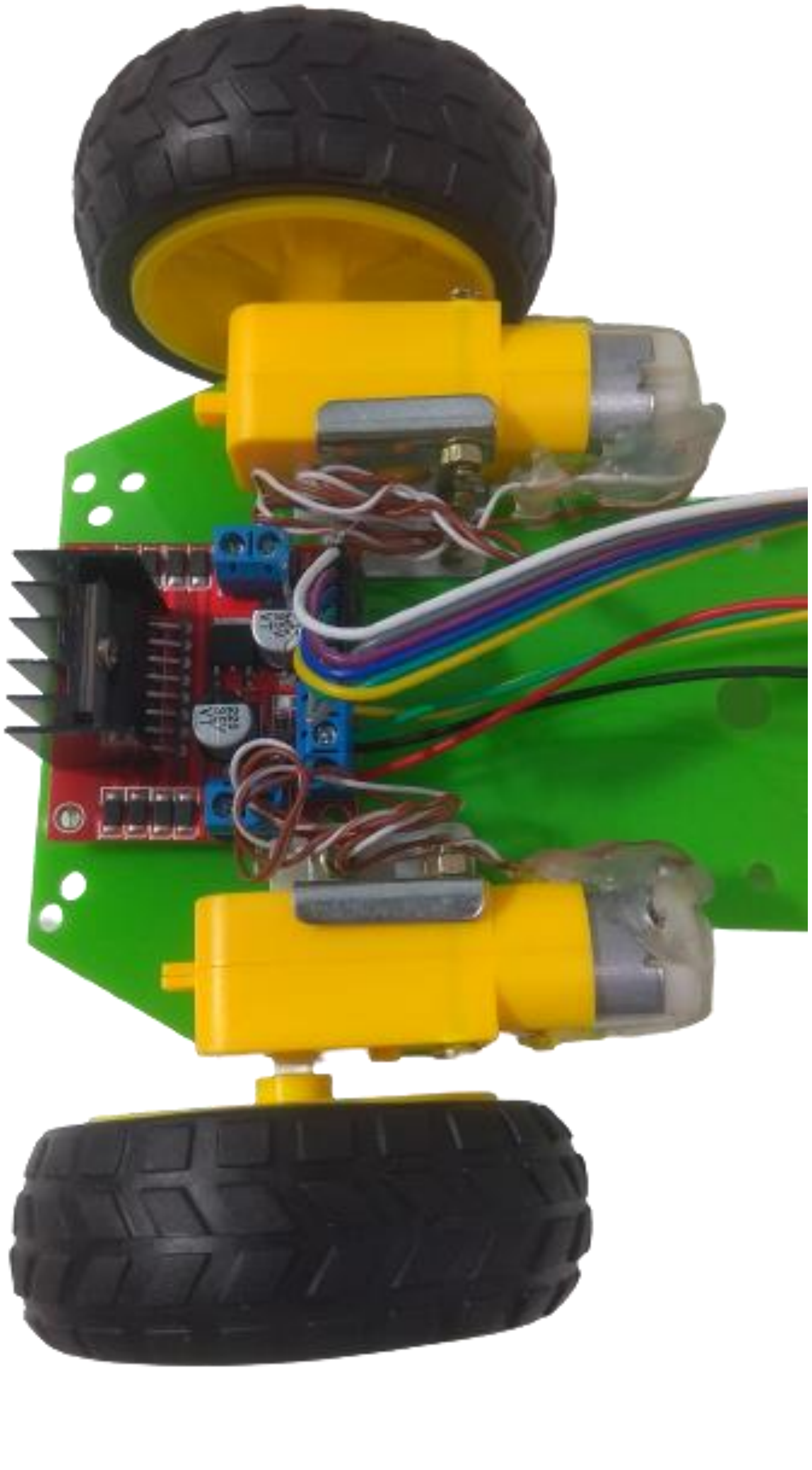
Step 8.

Now connect the male to female jumper wire with motor driver.

And also connect the motor with the L298n Motor Driver.

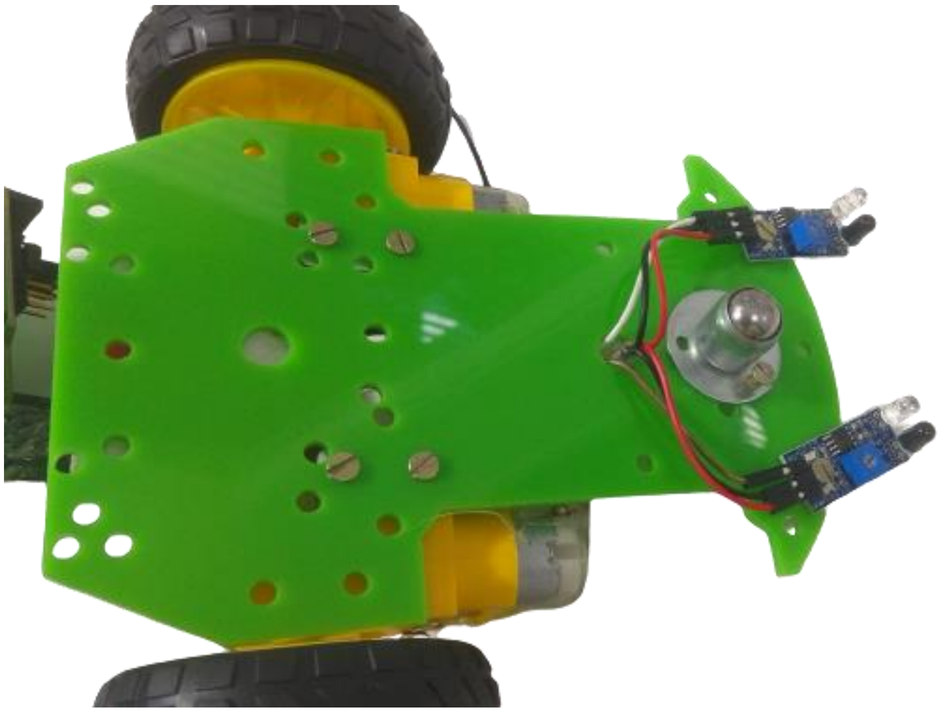
Connect left motor in out1 and out2

And right motor to out3 and out4 of the l298N Motor driver.



Step 9.

Now connect the male female wire with both IR Sensor module.

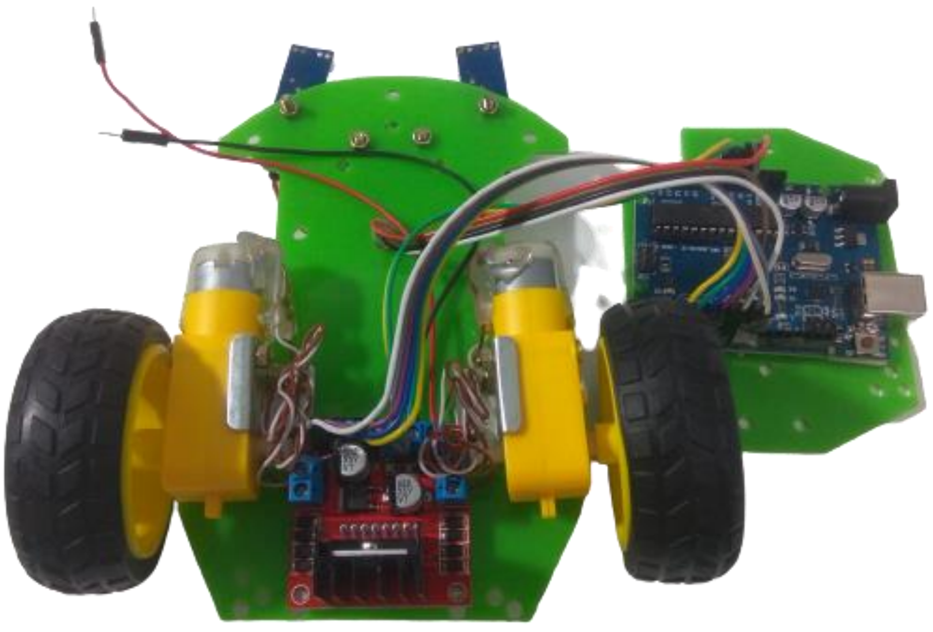


Step 10.

Now connect all these jumper wire with the Arduino Uno board.

You can also use bread board. If required.

Here we have just connected all positive with positive wire or jumper then these must be connected with 5v of Arduino Uno pin and all negative with Negative wires and these negative must be connected with GND of Arduino Uno board.

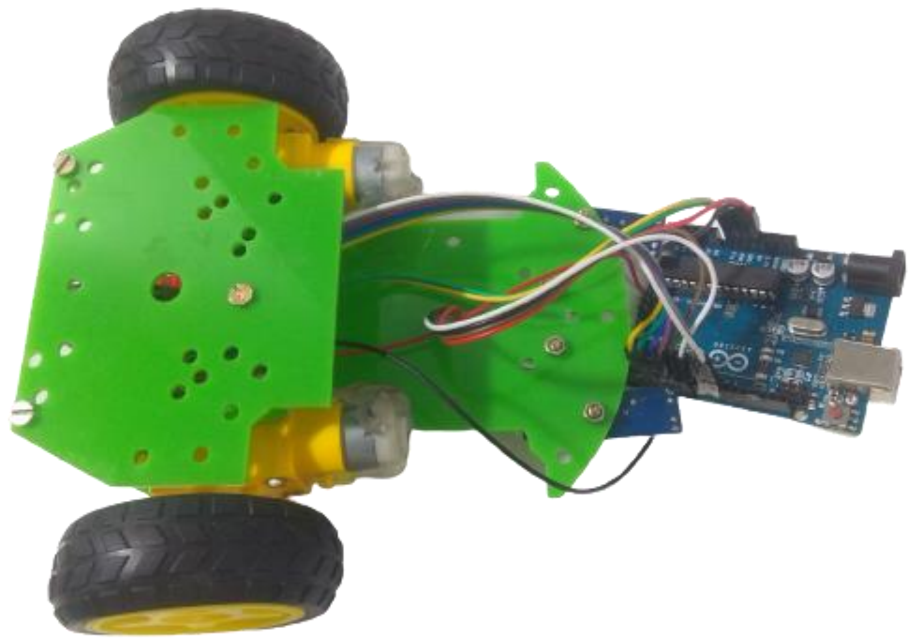


Step 11.

Now place the top layer of the line following robot.

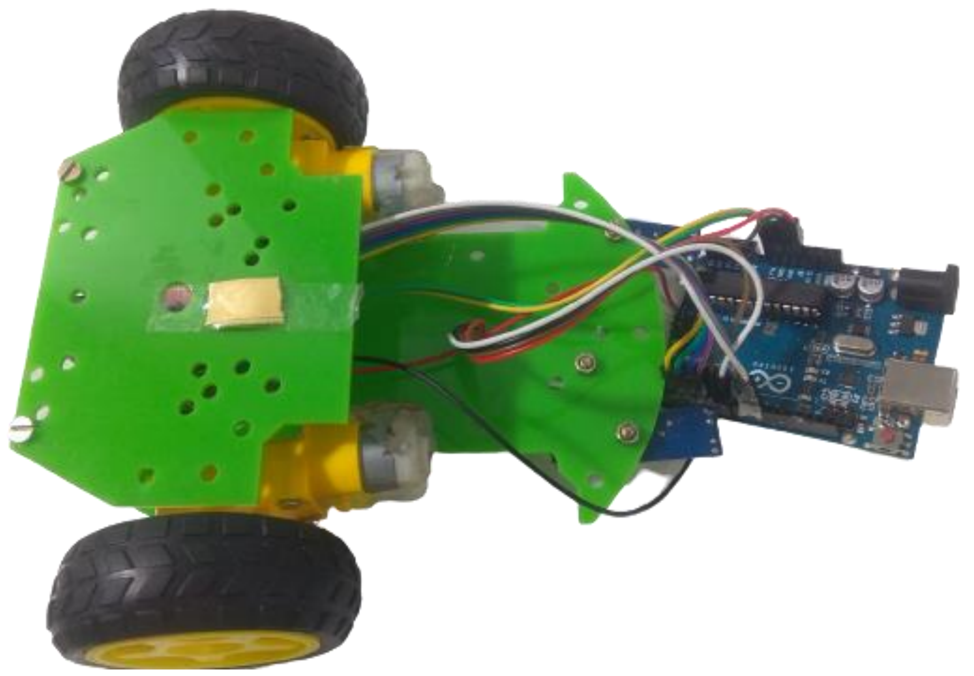
Place the spacer between the top layer and base of the robot.

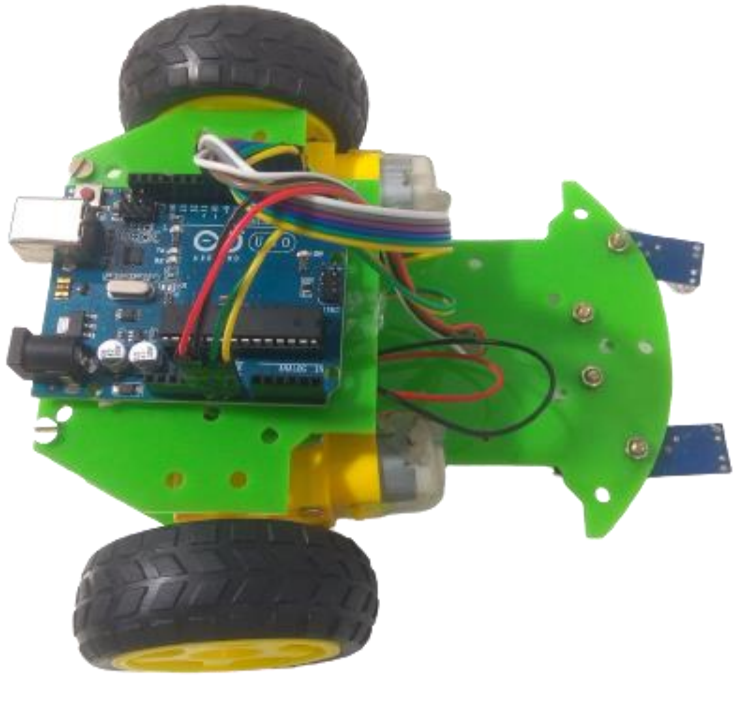
And the fix it using nut and bolt.



Step 12.

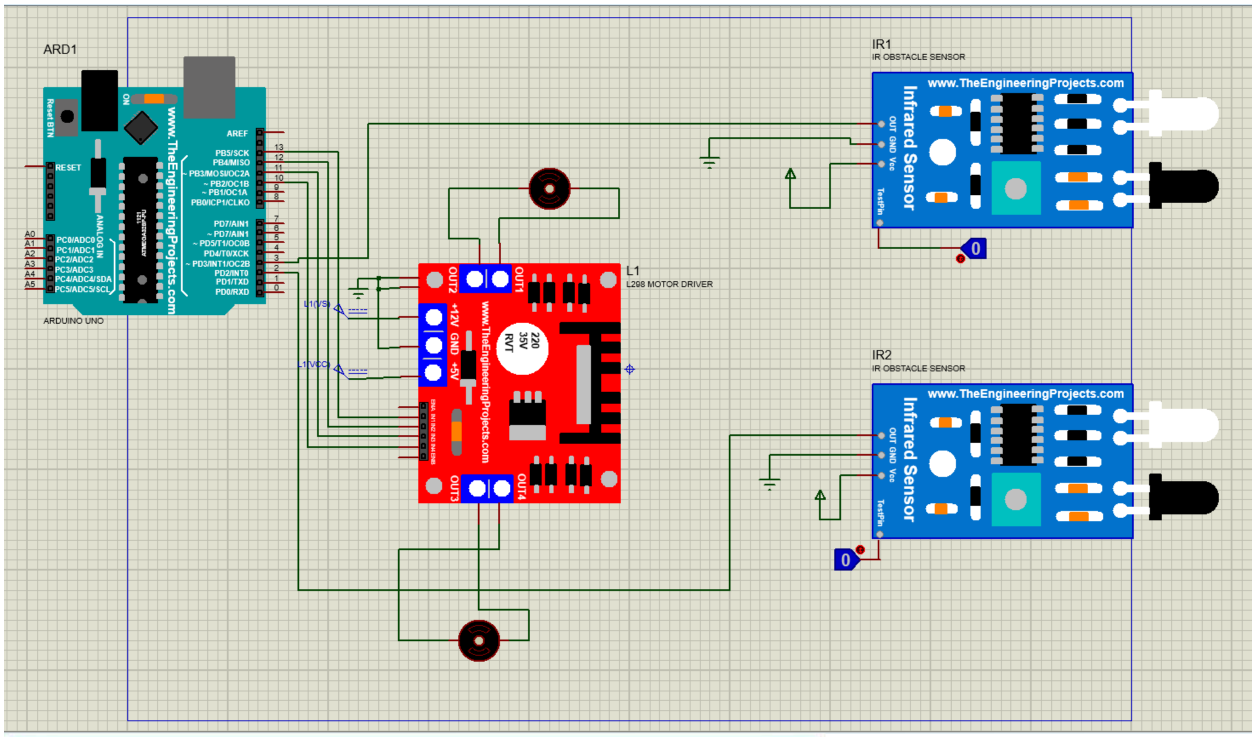
Now stick the Arduino Uno on the top of the line following robot and cover the nuts or bolt where you have to put Arduino Uno or motor driver shield to protect electronic components with short circuit.

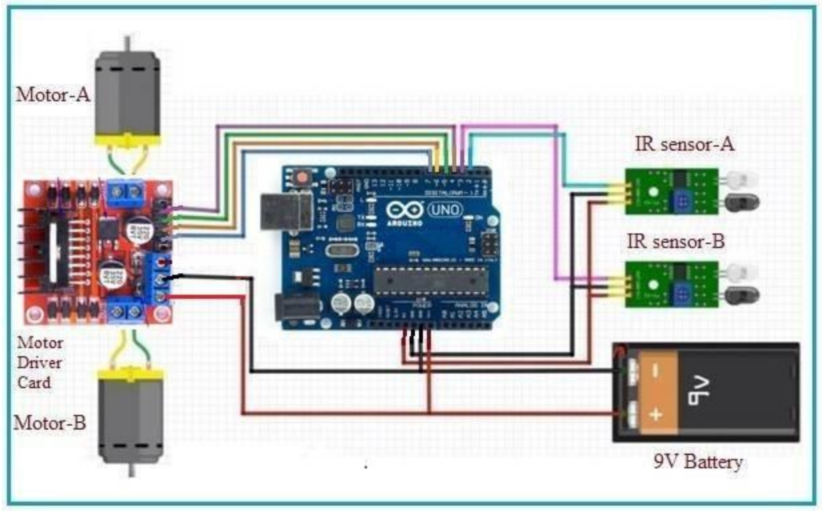




Wiring connection.

This image is only for reference.





L298N motor driver and Arduino Uno connection.

ENA of motor driver to pin no 9 of Arduino Uno.

IN1 of motor driver to pin no 2 of Arduino Uno.

IN2 of motor driver to pin no 3 of Arduino Uno.

IN3 of motor driver to pin no 4 of Arduino Uno.

IN4 of motor driver to pin no 5 of Arduino Uno.

ENB of motor driver to pin no 10 of Arduino Uno.

IR Sensor module with Arduino Uno

Left IR module

Vcc to 5v of Arduin Uno.

Gnd to GND of Arduin Uno.

And Signal to Pin no 6 of arduino.

Right IR module

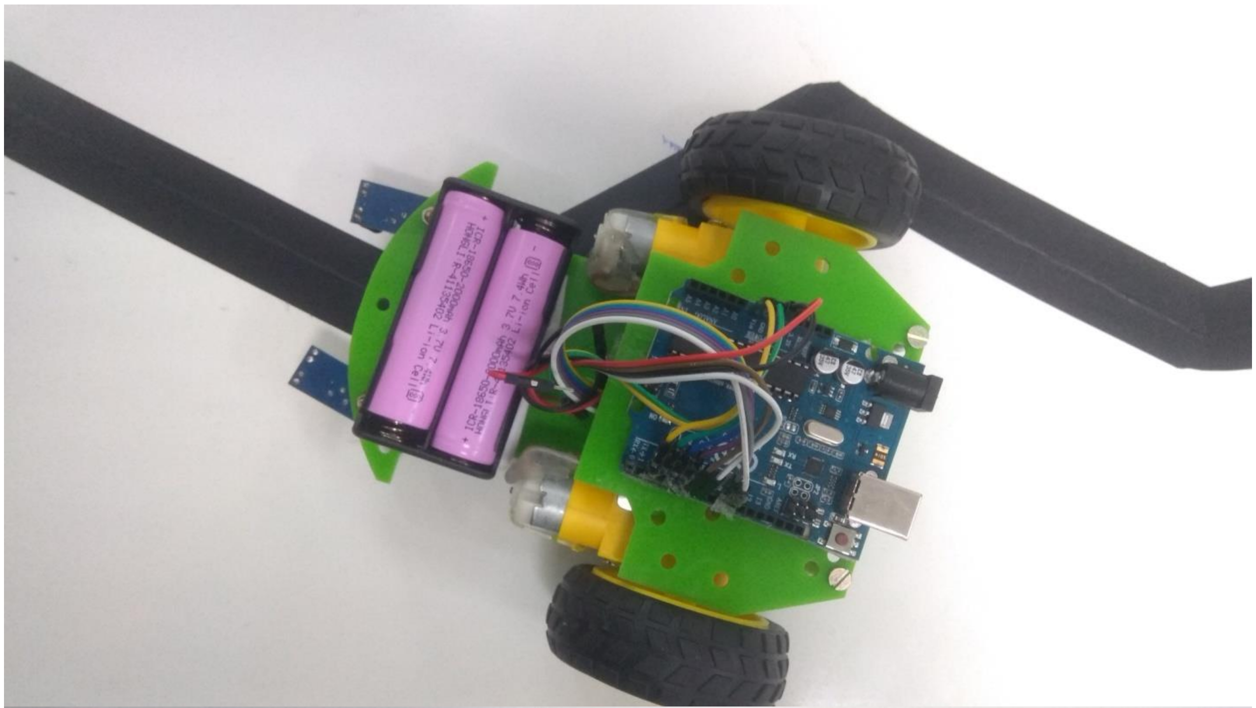
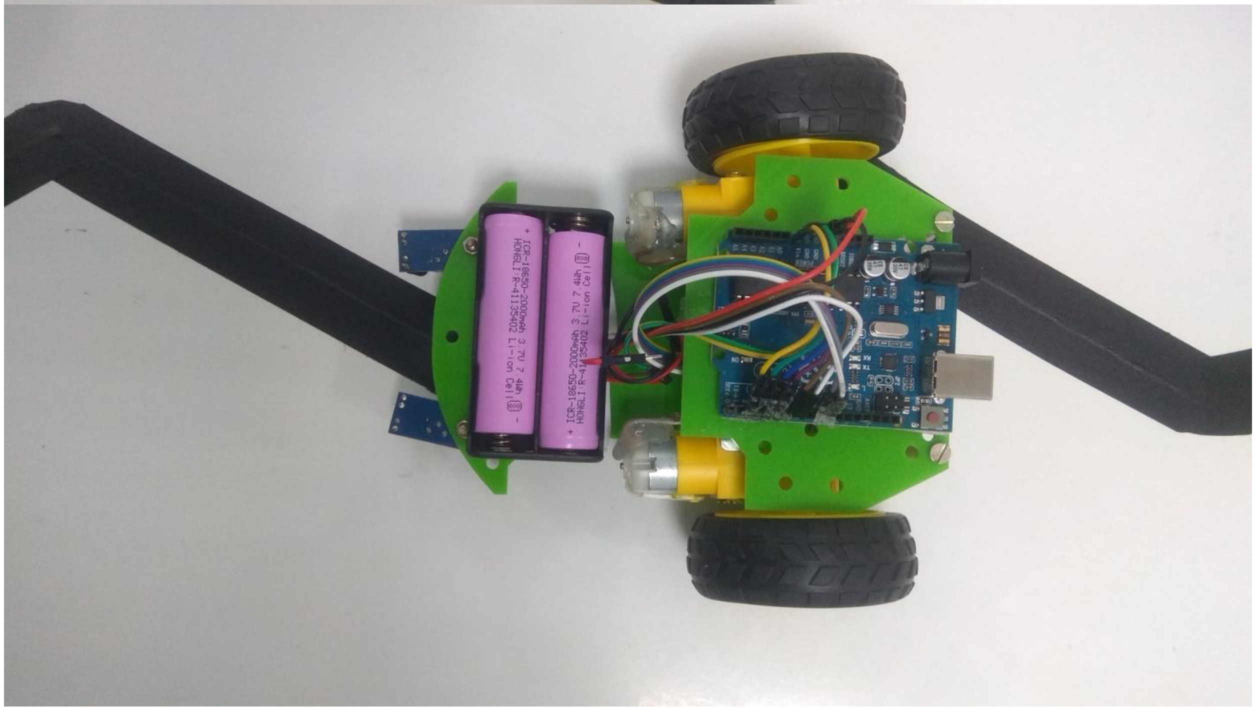
Vcc to 5v of Arduin Uno.

Gnd to GND of Arduin Uno.

And Signal to Pin no 7 of arduino.

Step 13.

Finally our line following robot is ready.

Micro controller which is used and which can be used.

About the microcontroller.

For this robot here we used Arduino Uno R3 microcontroller. You can also use Abira board SV1.0

Some features of these board are.

Arduino Uno R3 Board

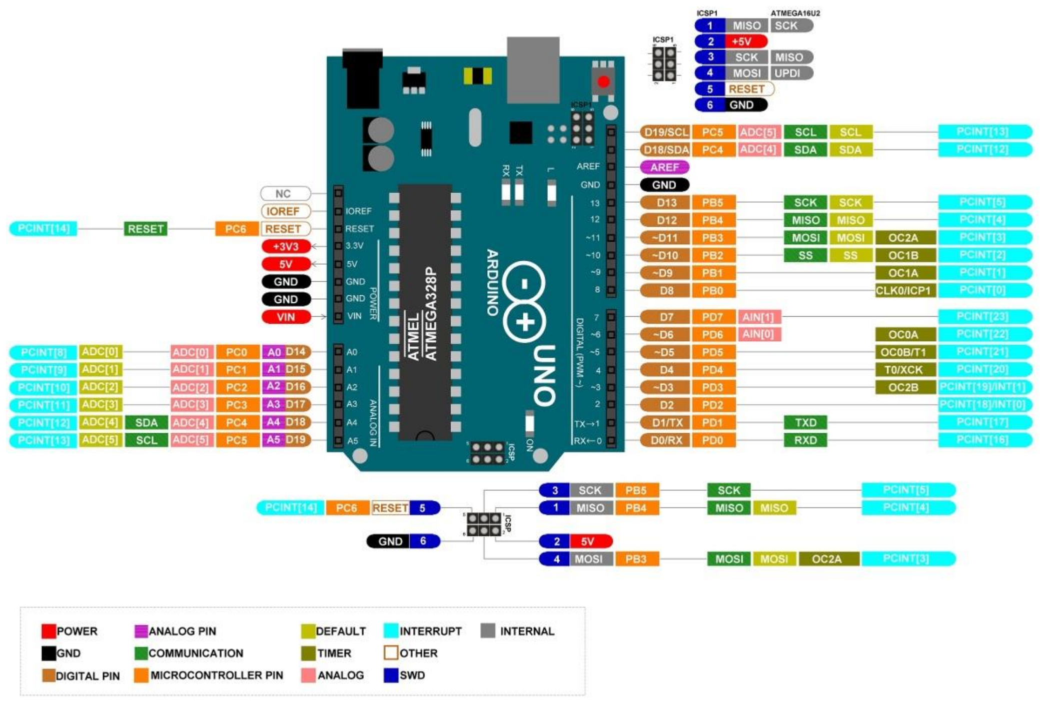
Arduino UNO is a microcontroller board based on the **ATmega328P**. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

Features:

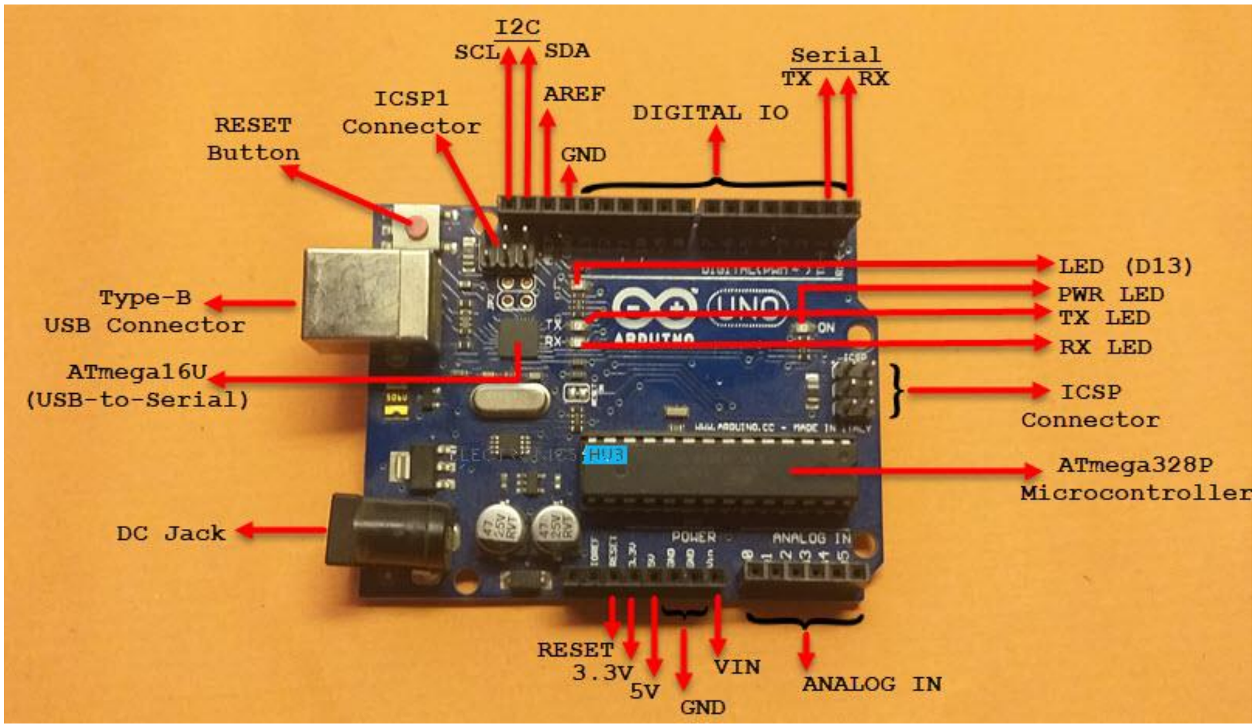
Model Type	Arduino Uno R3
Microcontroller Chip	ATmega328
Operating Voltage (VDC)	5
Input Voltage(Recommended)	7-12V
Input Voltage (limit)	6-20V
Analog I/O Pins	6
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
DC Current for 3.3V Pin	50

(mA)	
DC Current per I/O Pin (mA)	40
Clock Speed	16 MHz
SRAM	2
EEPROM	1 KB (ATmega328)
Flash Memory	32 KB
On Board LEDs	On/Off, L (PIN 13), TX, RX
Dimensions in mm (LxWxH)	75 x 54 x 12
Weight (gm)	26

Arduino Uno pin details.

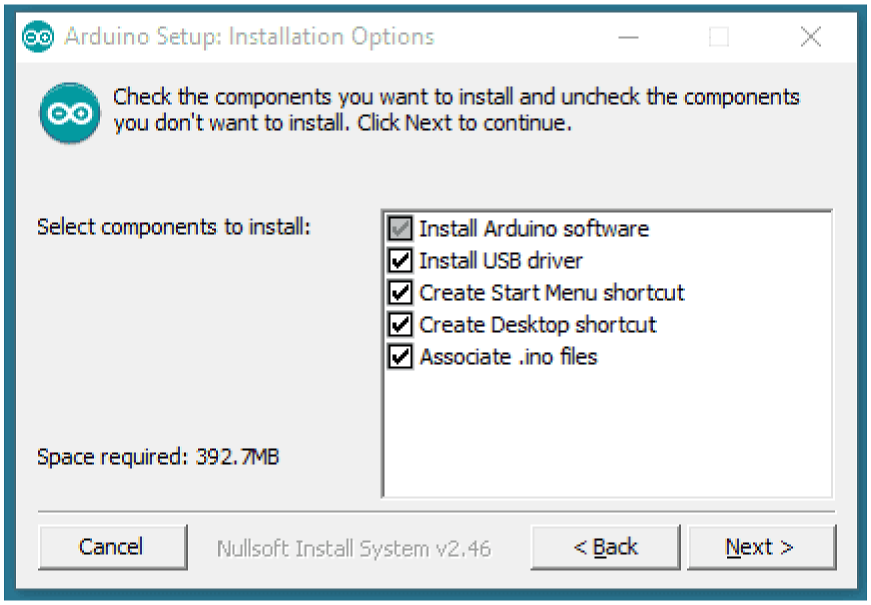


ICSP1
ATMEGA16U2
1 MISO SCK
2 +5V
3 SCK MISO
4 MOSI UPDI
5 RESET
6 GND
D19/SCL PC5 ADC[5] SCL SCL PCINT[13]
D18/SDA PC4 ADC[4] SDA SDA PCINT[12]
AREF
GND
D13 PB5 SCK SCK PCINT[5]
D12 PB4 MISO MISO PCINT[4]
~D11 PB3 MOSI MOSI OC2A PCINT[3]
~D10 PB2 SS SS OC1B PCINT[2]
~D9 PB1 OC1A PCINT[1]
D8 PB0 CLK0/ICP1 PCINT[0]
D7 PD7 AIN[1] PCINT[23]
~D6 PD6 AIN[0] OC0A PCINT[22]
~D5 PD5 OC0B/T1 PCINT[21]
D4 PD4 T0/XCK PCINT[20]
~D3 PD3 OC2B PCINT[19]/INT[1]
D2 PD2 PCINT[18]/INT[0]
D1/TX PD1 TXD PCINT[17]
D0/RX PD0 RXD PCINT[16]
NC
IOREF
PCINT[14] RESET PC6 RESET
+3V3
5V
GND
GND
VIN
PCINT[8] ADC[0] ADC[0] PC0 A0 D14
PCINT[9] ADC[1] ADC[1] PC1 A1 D15
PCINT[10] ADC[2] ADC[2] PC2 A2 D16
PCINT[11] ADC[3] ADC[3] PC3 A3 D17
PCINT[12] ADC[4] SDA ADC[4] PC4 A4 D18
PCINT[13] ADC[5] SCL ADC[5] PC5 A5 D19
ATMEL ATMEGA328P
ARDUINO
UNO
DIGITAL (PWM ~)
ANALOG IN
POWER
ICSP
3 SCK PB5 SCK PCINT[5]
1 MISO PB4 MISO MISO PCINT[4]
PCINT[14] PC6 RESET 5
GND 6
2 5V
4 MOSI PB3 MOSI MOSI OC2A PCINT[3]
POWER
ANALOG PIN
DEFAULT
INTERRUPT
INTERNAL
GND
COMMUNICATION
TIMER
OTHER
DIGITAL PIN
MICROCONTROLLER PIN
ANALOG
SWD



How to install the Arduino Software (IDE)

This document explains how to install the Arduino Software (IDE) on Windows machines.
Download the Arduino Software (IDE)
Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. The Zip file is also useful if you want to create a portable installation.
When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.



Arduino Setup: Installation Options

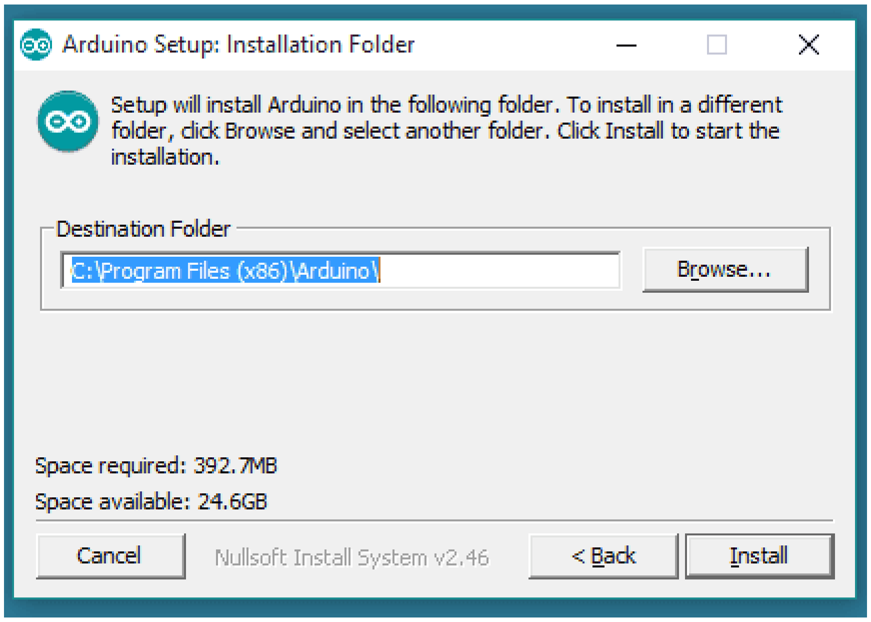
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

- [x] Install Arduino software
- [x] Install USB driver
- [x] Create Start Menu shortcut
- [x] Create Desktop shortcut
- [x] Associate .ino files

Space required: 392.7MB

Choose the components to install.



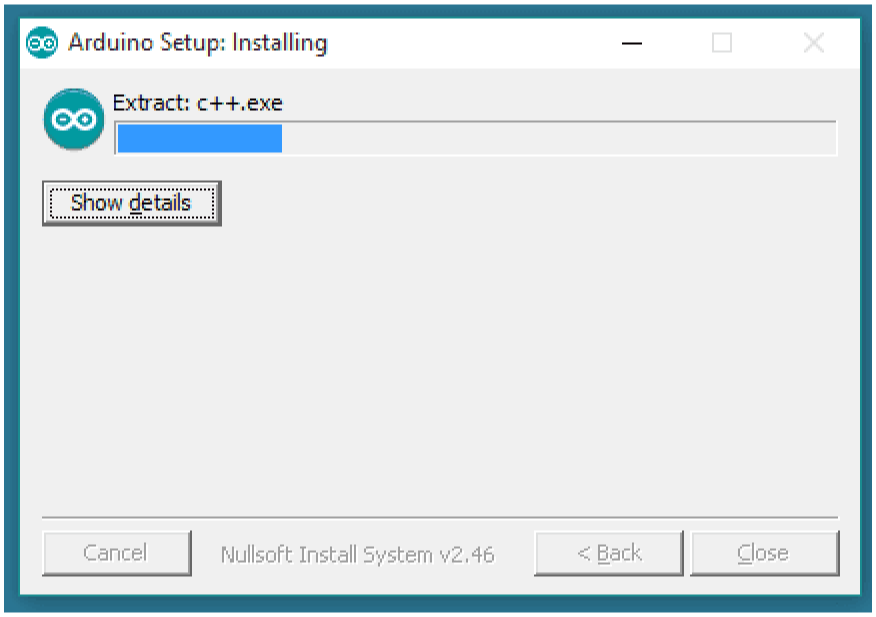
Arduino Setup: Installation Folder

Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: C:\Program Files (x86)\Arduino\

Space required: 392.7MB
Space available: 24.6GB

Choose the installation directory.



Installation in progress.

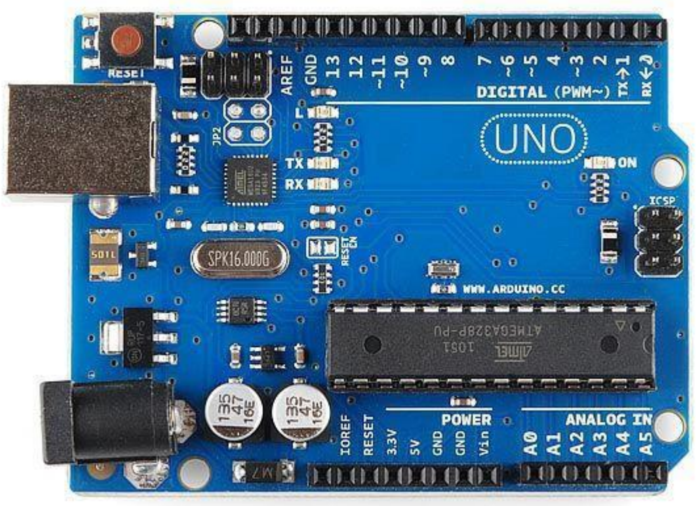
The process will extract and install all the required files to execute properly the Arduino Software (IDE)
The text of the Arduino getting started guide is licensed under a Creative Commons Attribution-ShareAlike 3.0 License. Code samples in the guide are released into the public domain.

How to connect Arduino Uno R3.

Steps to upload a program to Arduino UNO board through a Windows PC:- In order to upload a program to an Arduino UNO board using a Windows PC involves

- Connecting UNO with PC using a USB cable.
- Selection of the correct board and port in the Tools menu.
- Uploading the program using the upload button in Arduino IDE after successful compilation.

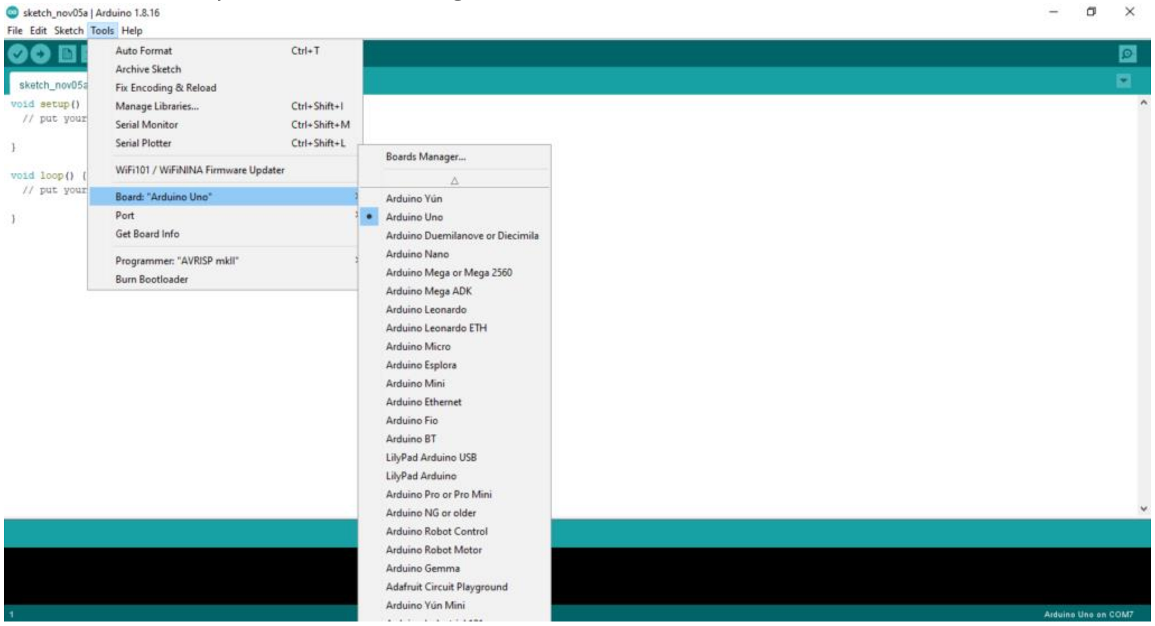
Arduino UNO board(SMD/DIP)



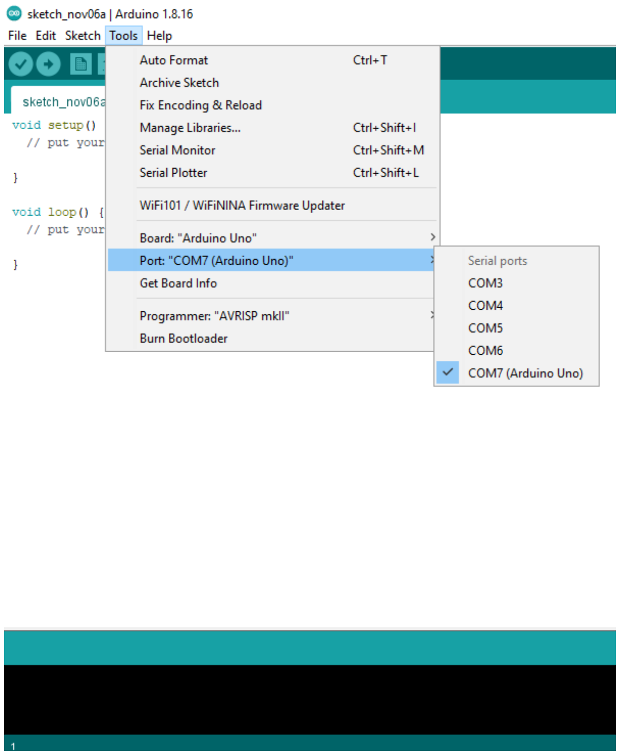
Arduino Uno R3 DIP

First, we will discuss the problems related to the DIP board.
Problem with uploading programs to Arduino board:-
There are multiple pieces involved to upload a program onto an Arduino board and if any of them aren’t right, the upload can fail.
Following are some possible solutions: –
1. Arduino Software

- One possible reason could be that the right Arduino board is not selected in Arduino IDE. To check this, in your Arduino IDE, go to Tools > Board menu and select Arduino UNO.



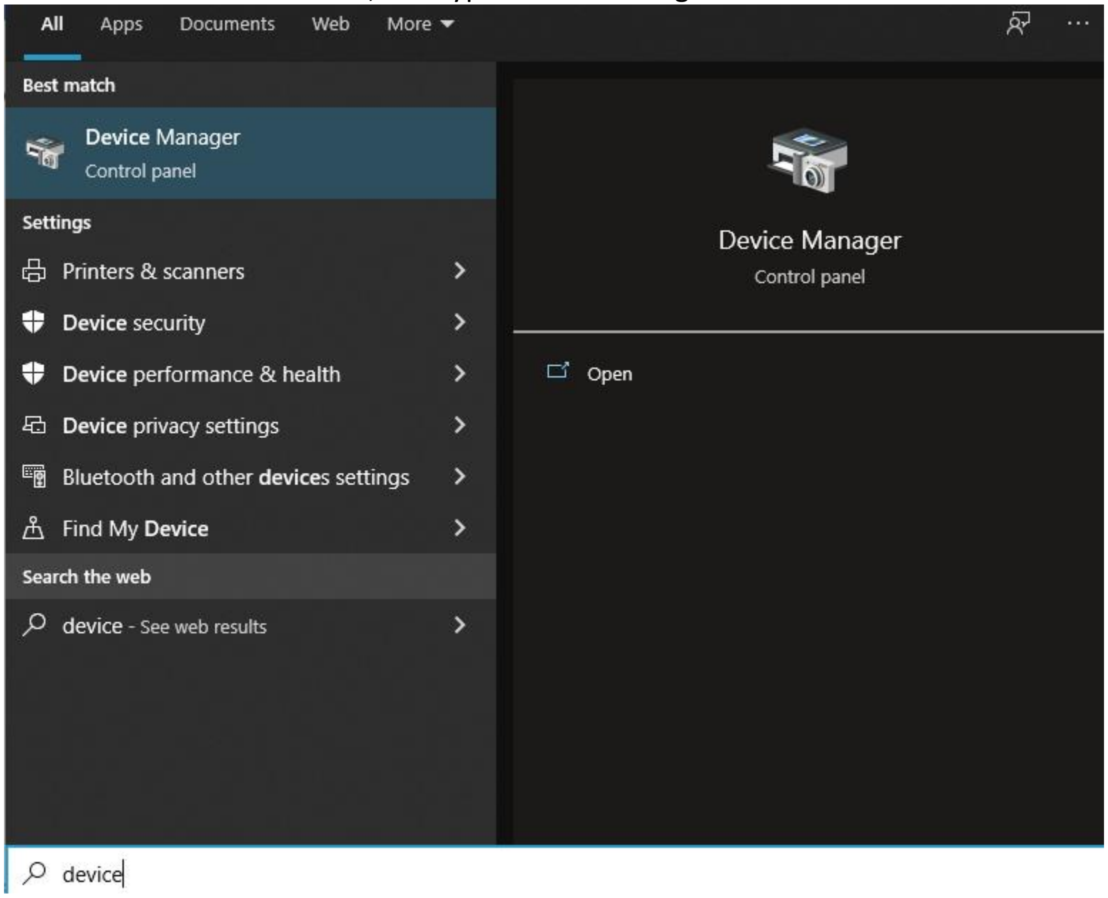
- Check if the correct port is selected by navigating to Tools > Port menu (if your port doesn't appear, try restarting the IDE with the board connected to the computer), then select the port which has Arduino UNO written with it. If you don't seem to have a serial port for your Arduino board, see the following information about drivers.


sketch_nov06a | Arduino 1.8.16
File Edit Sketch Tools Help
sketch_nov06a
void setup() {
// put your
}
void loop() {
// put your
}
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port: "COM7 (Arduino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM3
COM4
COM5
COM6
COM7 (Arduino Uno)

2. Drivers – Drivers provide a way for software on your computer (i.e. the Arduino software) to talk to the hardware you connect to your computer (the Arduino board). In the case of Arduino, the drivers work by providing a virtual serial port (or virtual COM port). If you are operating a

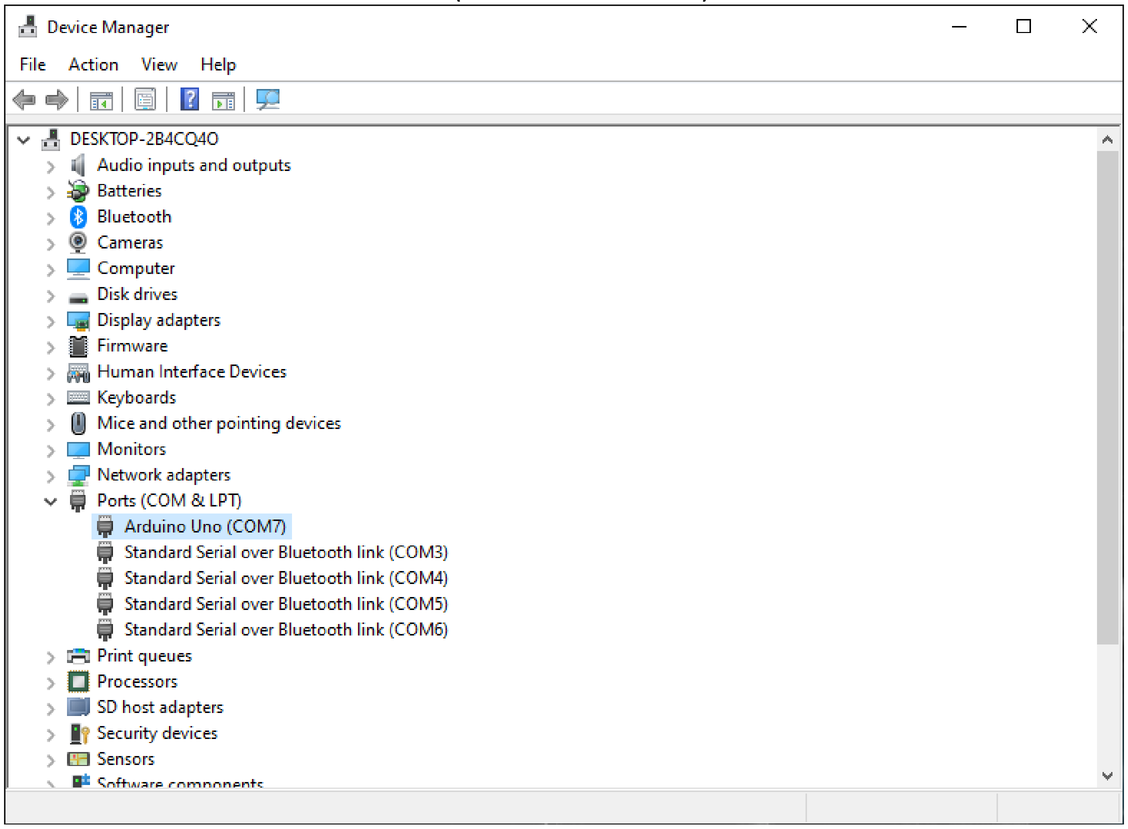
Windows system, and your Arduino drivers are not up to date you will usually get Arduino ports not showing up. If this is the case, update using these steps below.

- Go to Start Menu, and type Device Manager on the search bar.



- Click the Device Manager icon to open a new window.

- Scroll down to Ports (COM & LPT) and click on to expand.

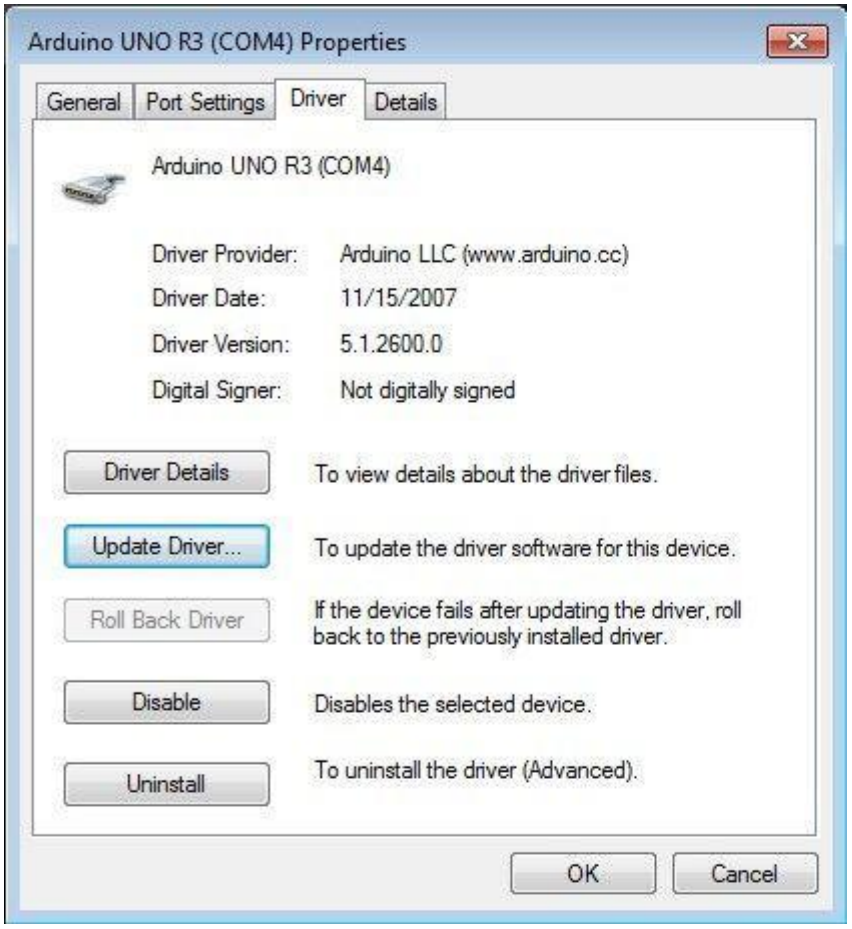


Device Manager

File Action View Help

- DESKTOP-2B4CQ4O
  - Audio inputs and outputs
  - Batteries
  - Bluetooth
  - Cameras
  - Computer
  - Disk drives
  - Display adapters
  - Firmware
  - Human Interface Devices
  - Keyboards
  - Mice and other pointing devices
  - Monitors
  - Network adapters
  - Ports (COM & LPT)
    - Arduino Uno (COM7)
    - Standard Serial over Bluetooth link (COM3)
    - Standard Serial over Bluetooth link (COM4)
    - Standard Serial over Bluetooth link (COM5)
    - Standard Serial over Bluetooth link (COM6)
  - Print queues
  - Processors
  - SD host adapters
  - Security devices
  - Sensors
  - Software components

- Double-click the Arduino Uno device for the properties window to display.
- Select the Driver tab, and click Update Driver


Arduino UNO R3 (COM4) Properties

General | Port Settings | Driver | Details

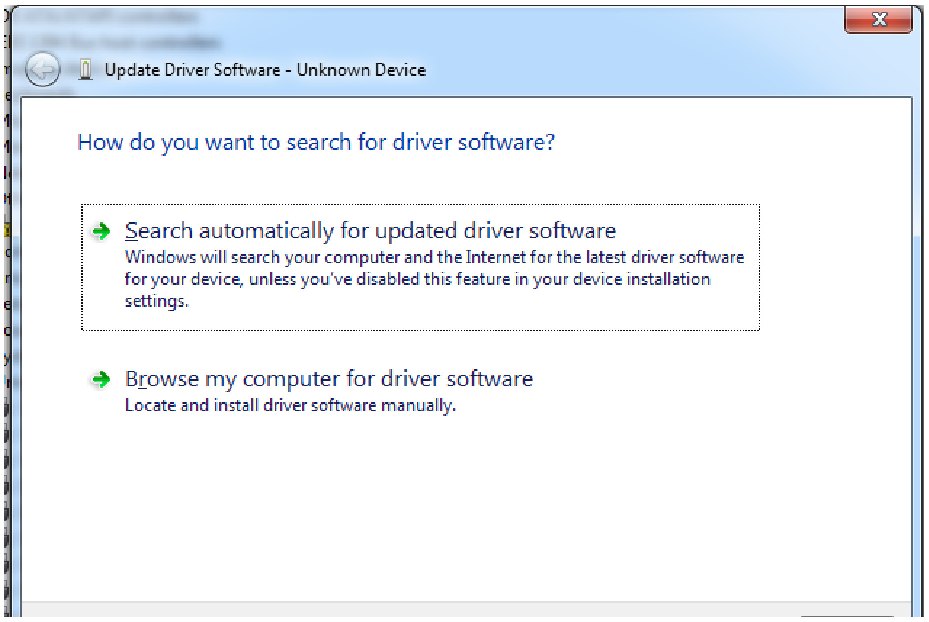
Arduino UNO R3 (COM4)

Driver Provider:	Arduino LLC (www.arduino.cc)
Driver Date:	11/15/2007
Driver Version:	5.1.2600.0
Digital Signer:	Not digitally signed

Driver Details	To view details about the driver files.
Update Driver...	To update the driver software for this device.
Roll Back Driver	If the device fails after updating the driver, roll back to the previously installed driver.
Disable	Disables the selected device.
Uninstall	To uninstall the driver (Advanced).

OK Cancel

- Choose the search automatically for the updated driver software option.


Update Driver Software - Unknown Device

How do you want to search for driver software?

Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.

Browse my computer for driver software
Locate and install driver software manually.

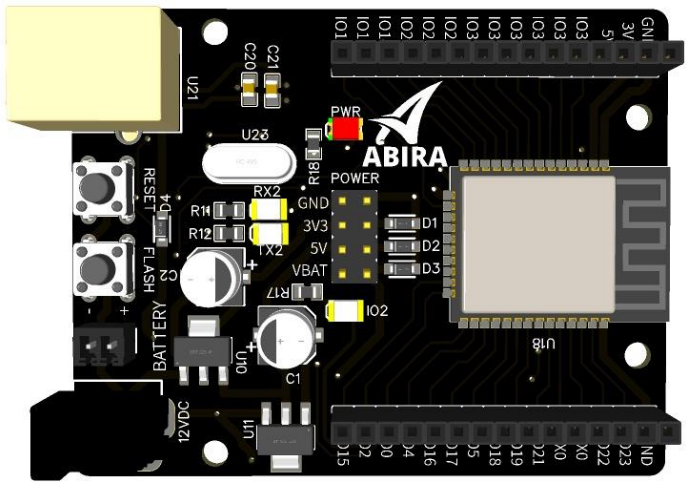
- Windows will begin to update.
- After the update is done, “Windows has successfully updated your driver software” will be displayed on your screen.
- Run a final check to be sure the driver was updated correctly.

3. Physical Connection

- First, make sure your board is on (the green LED is on) and connected to the computer.
- The Arduino Uno may have trouble connecting to Windows through a USB hub. If nothing appears in your “Tools > Serial Port” menu, try plugging the board directly into your computer and restarting the Arduino IDE.
- Disconnect digital pins 0 and 1 while uploading as they are shared with serial communication with the computer (they can be connected and used after the code has been uploaded).
- Try uploading with nothing connected to the board (apart from the USB cable).
- Make sure the board isn’t touching anything metallic or conductive.
- Try a different USB cable; sometimes they don’t work.
- If you have a board that doesn’t support auto-reset, be sure that you are resetting the board a couple of seconds before uploading using the reset button present in the Arduino UNO board.
- If you get this error: [VP 1] Device is not responding correctly. try uploading again (i.e., reset the board and press the download button a second time).

4. Bootloader – Make sure there’s a bootloader burned on your Arduino board. To check, reset the board. The built-in L LED (which is connected to pin 13) should blink. If it doesn’t, there may not be a bootloader on your board.

Abira Board SV1.0



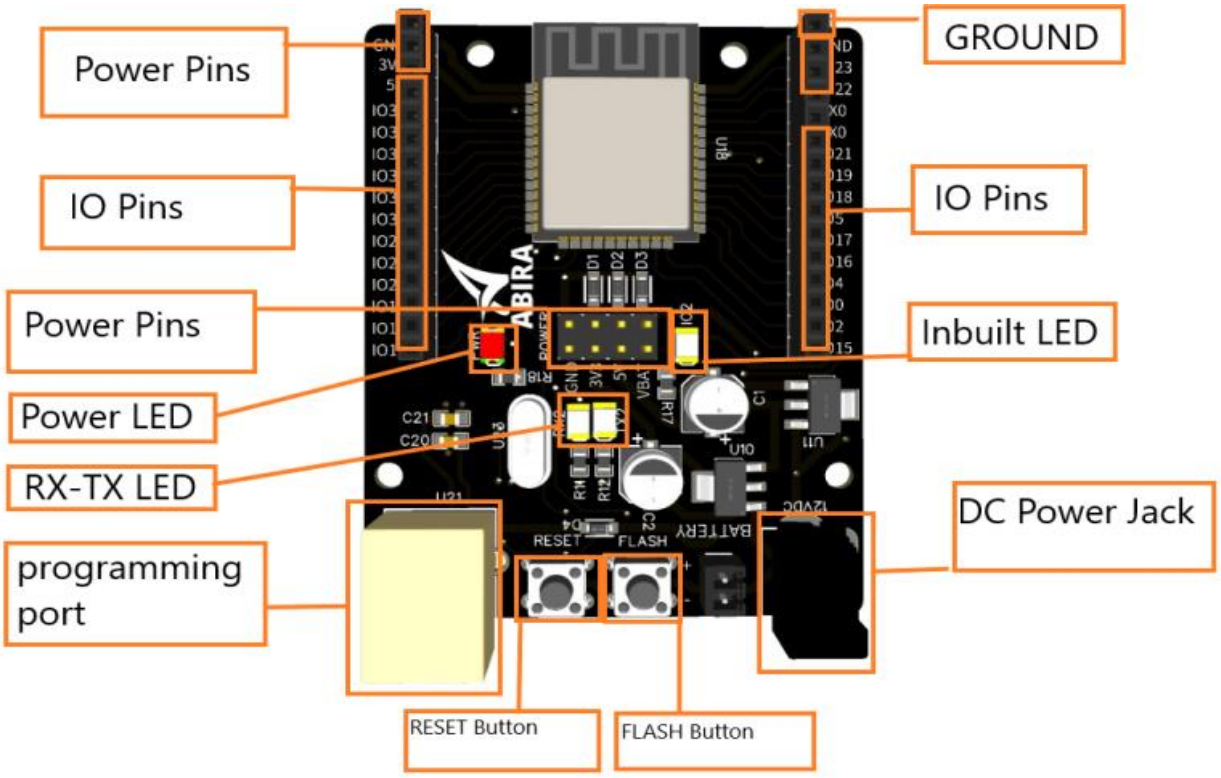
Abira board SV1.0 is best development board for getting into the field of Electronics and learning with hands on Experience using such a powerful hardware. Including the features which comes with this board makes learning with it so much fun and the possibilities is endless.

Specification:
AbiraSV1.0 comes with ESP32 Microcontroller on Board. Powered by 40 nm technology, ESP32 provides a robust, highly integrated platform, which helpsmeet the continuous demands for efficient power usage, compact design, security, high performance, and reliability.

Main Features
It Comes with 2.4GHz Wi-Fi band
Bluetooth
Dual high performance Xtensa 32-bit LX6 CPU cores
Ultra Low Power co-processor
Multiple peripherals
4Mb of Internal flash memory
320Kb SRAM
4Kb EEPROM
Peripheral
34 Programmable GPIOs
2 8bit Digital to analogue converter
4 SPI communication interfaces
2 I2C Interfaces

Programming of Arduino Uno R3.
First we need to install Arduino IDE from Arduino official website

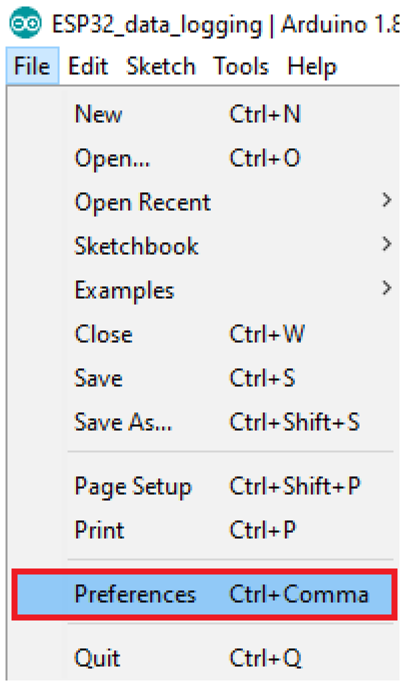
https://docs.Arduino.cc/software/ide-v1/tutorials/Windows
Then install it or extract it. After that you will get Arduino exe or Arduino application.



Installing ESP32 Add-on in Arduino IDE

To install the ESP32 board in your Arduino IDE, follow these next instructions:

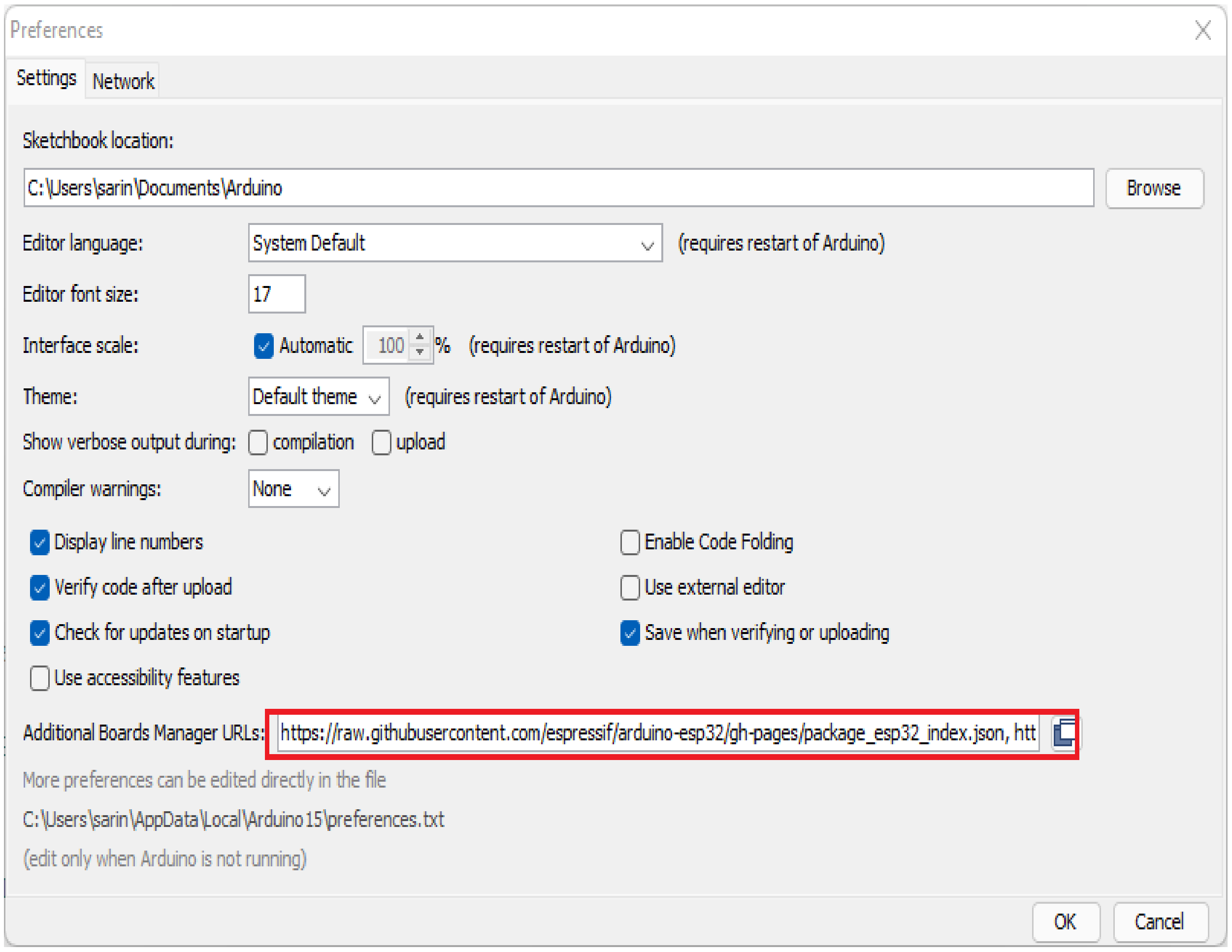
1. In your Arduino IDE, go to **File**> **Preferences**



2. Enter the following into the “Additional Board Manager URLs” field:

```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-
pages/package_esp32_index.json
```

Then, click the “OK” button:



Preferences

Settings | Network

Sketchbook location:
C:\Users\sarin\Documents\Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 17

Interface scale: ✓ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: ☐ compilation ☐ upload

Compiler warnings: None

✓ Display line numbers
✓ Verify code after upload
✓ Check for updates on startup
☐ Use accessibility features
☐ Enable Code Folding
☐ Use external editor
✓ Save when verifying or uploading

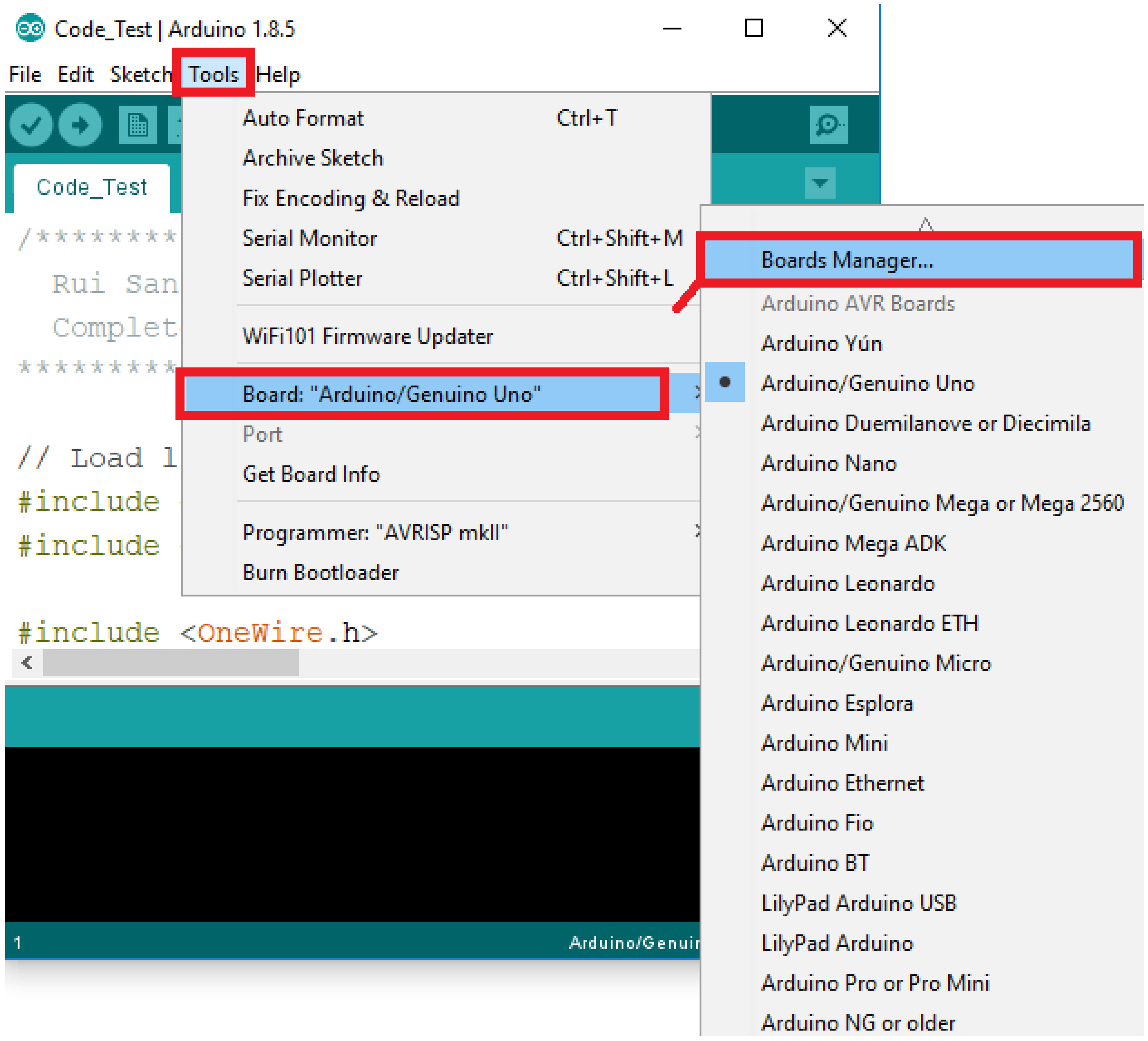
Additional Boards Manager URLs: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json, htt

More preferences can be edited directly in the file
C:\Users\sarin\AppData\Local\Arduino15\preferences.txt
(edit only when Arduino is not running)

Note: if you already have the ESP8266 boards URL, you can separate the URLs with a comma as follows:

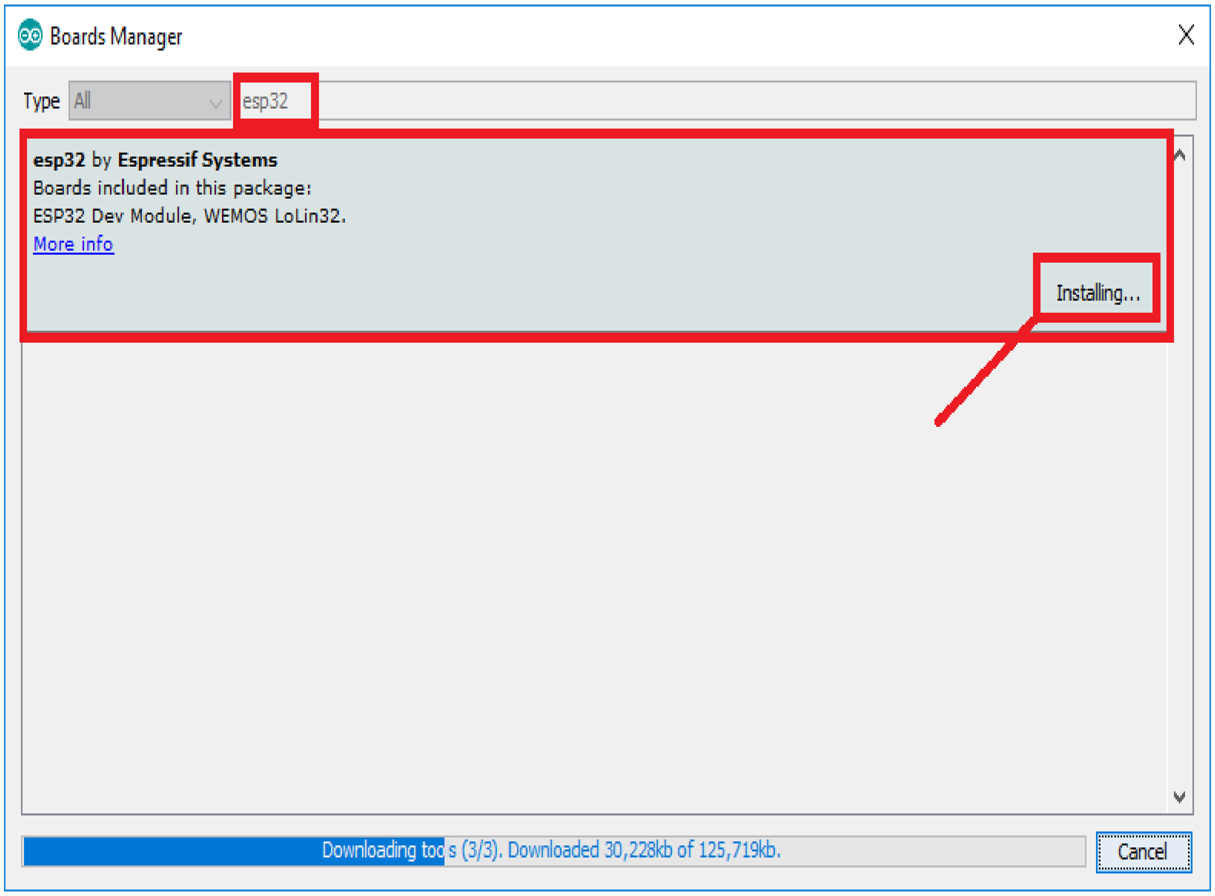
```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-
pages/package_esp32_index.json,
http://arduino.esp8266.com/stable/package_esp8266com_index.json
```

3. Open the Boards Manager. Go to **Tools** > **Board** > **Boards Manager…**



```
/*********
  Rui San
  Complet
*********

// Load l
#include
#include

#include <OneWire.h>
```

4. Search for **ESP32** and press install button for the “**ESP32 by Espressif Systems**“:


Boards Manager

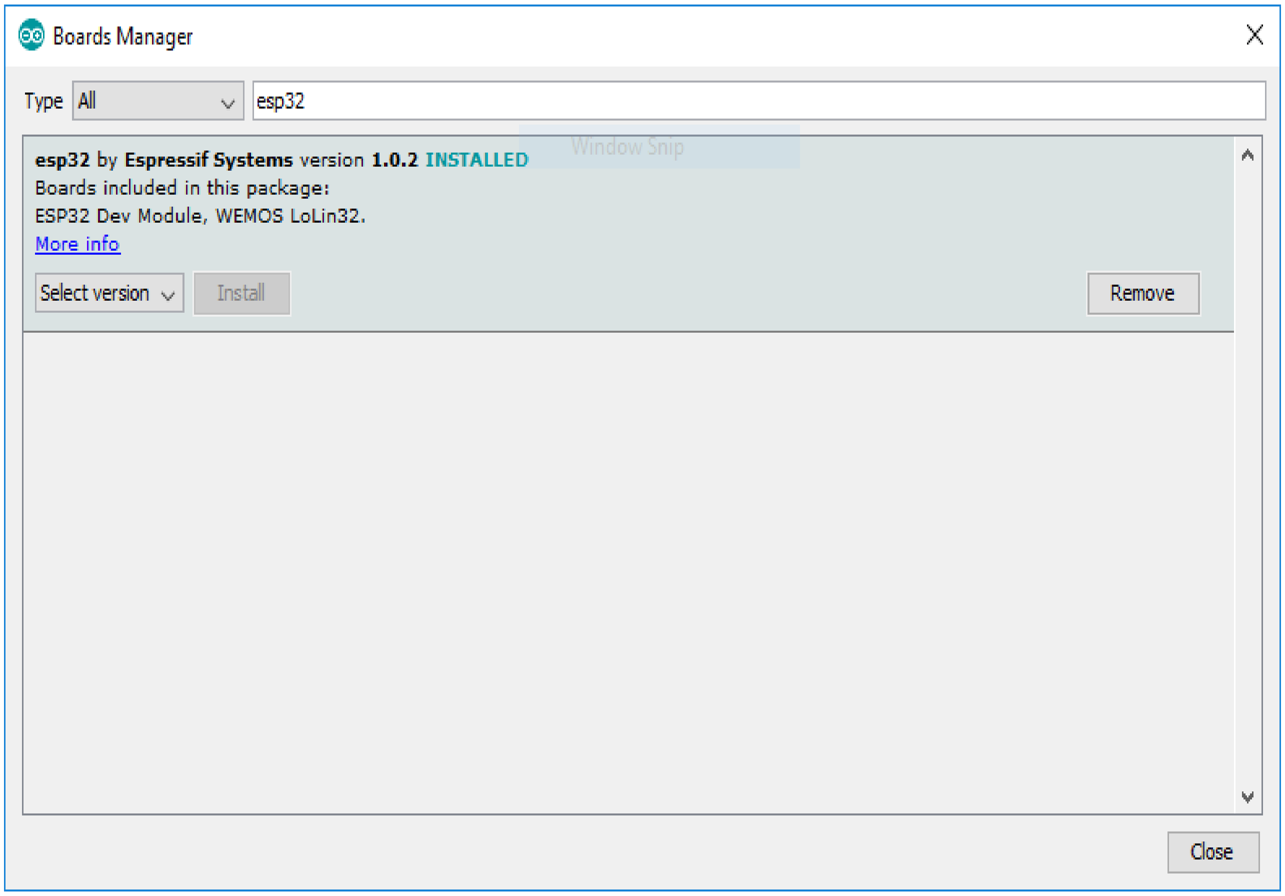
Type All esp32

esp32 by **Espressif Systems**
Boards included in this package:
ESP32 Dev Module, WEMOS LoLin32.
More info

Installing...

Downloading tools (3/3). Downloaded 30,228kb of 125,719kb.

5. That's it. It should be installed after a few seconds.

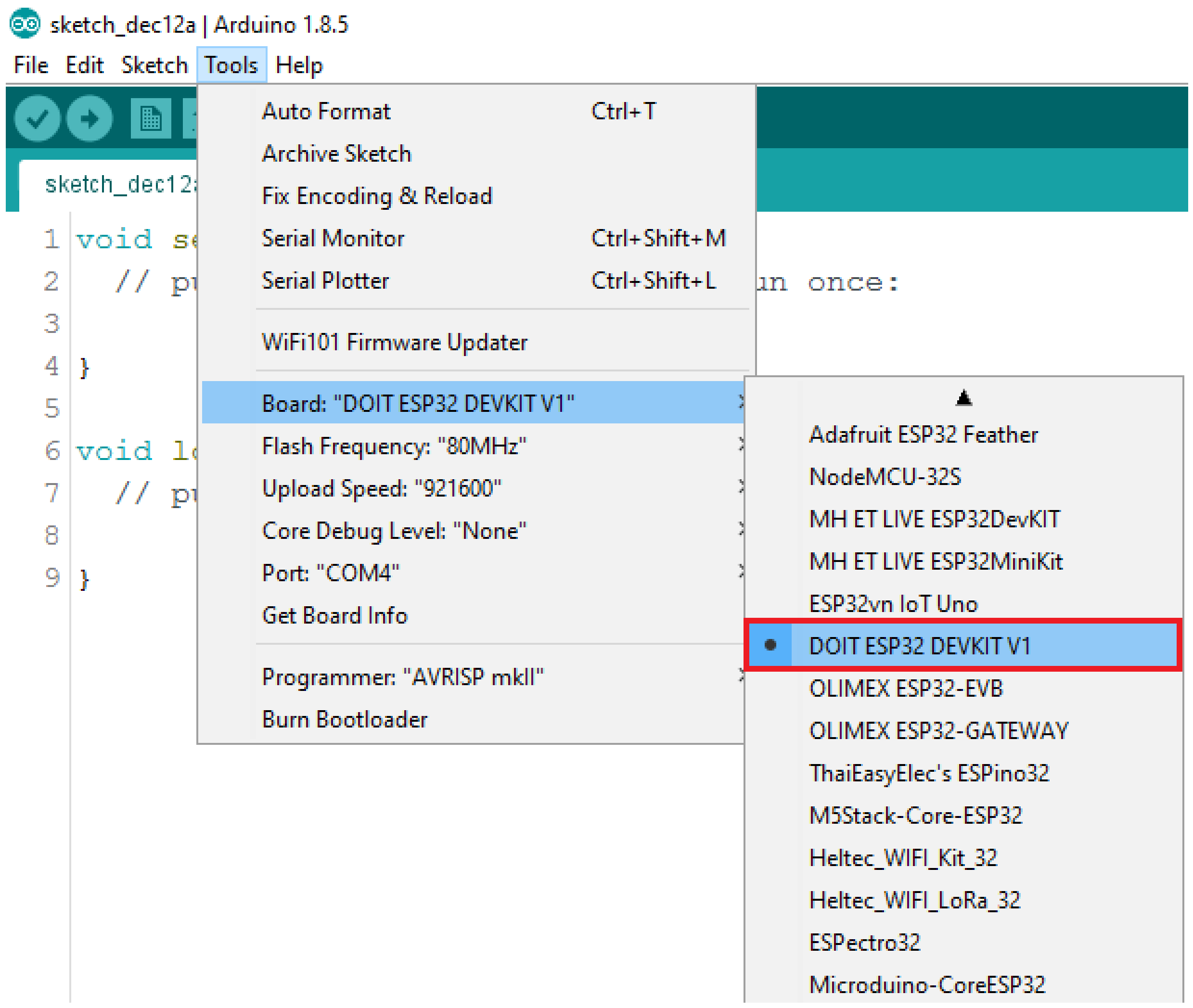


esp32 by **Espressif Systems** version **1.0.2** **INSTALLED**
Boards included in this package:
ESP32 Dev Module, WEMOS LoLin32.

TESTING

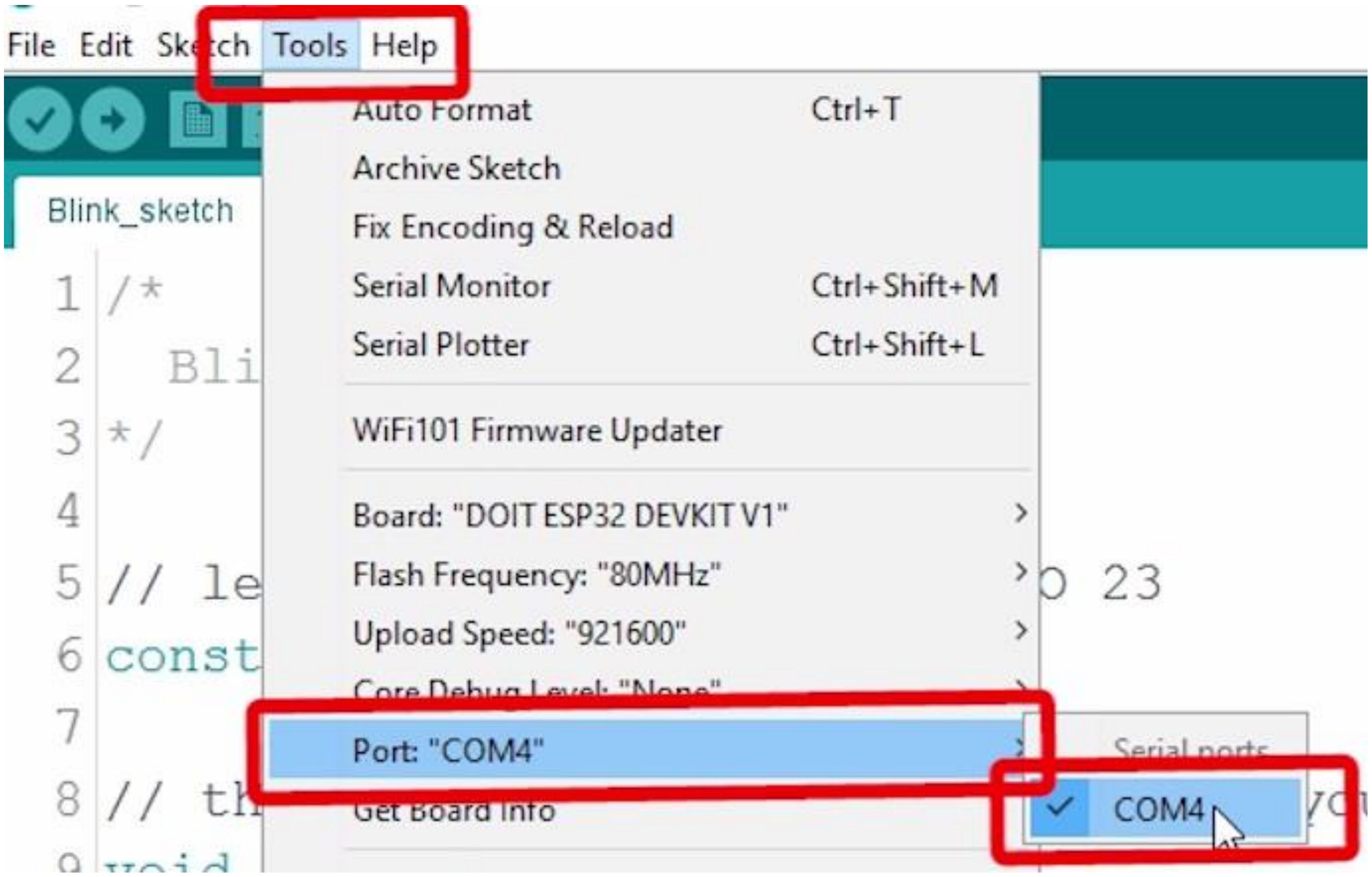
Plug the ESP32 board to your computer. With your Arduino IDE open, follow these steps:

1. Select your Board in **Tools** > **Board** menu (in my case it's the **DOIT ESP32 DEVKIT V1**)

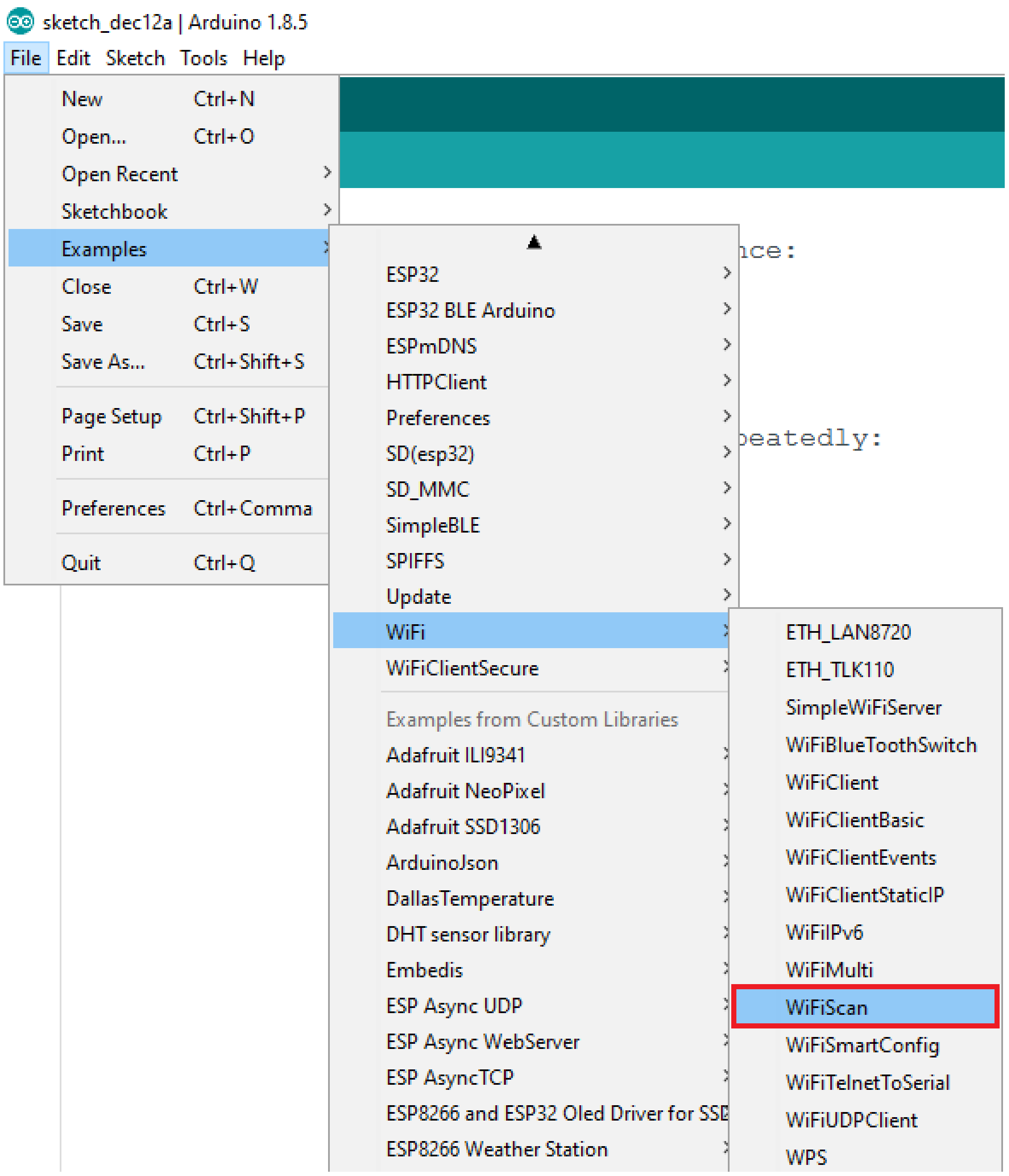


```
void se
  // pu                    un once:

}

void lo
  // pu

}
```

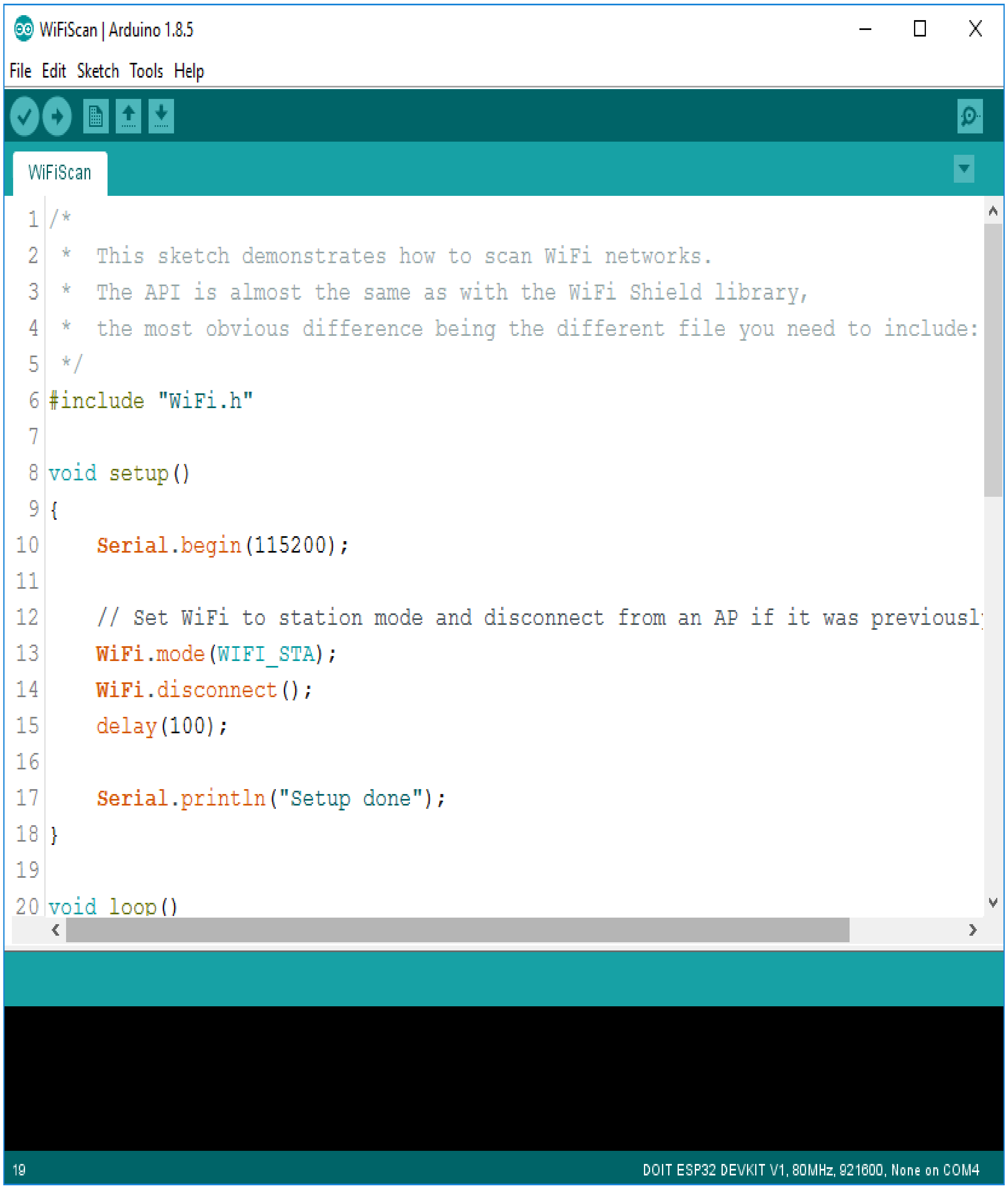
2. Select the Port (if you don’t see the COM Port in your Arduino IDE, you need to install the CP210x USB to UART Bridge VCP Drivers):



3. Open the following example under **File** > **Examples** > **WiFi (ESP32)** > **WiFiScan**



4. A new sketch opens in your Arduino IDE:


WiFiScan | Arduino 1.8.5

File Edit Sketch Tools Help

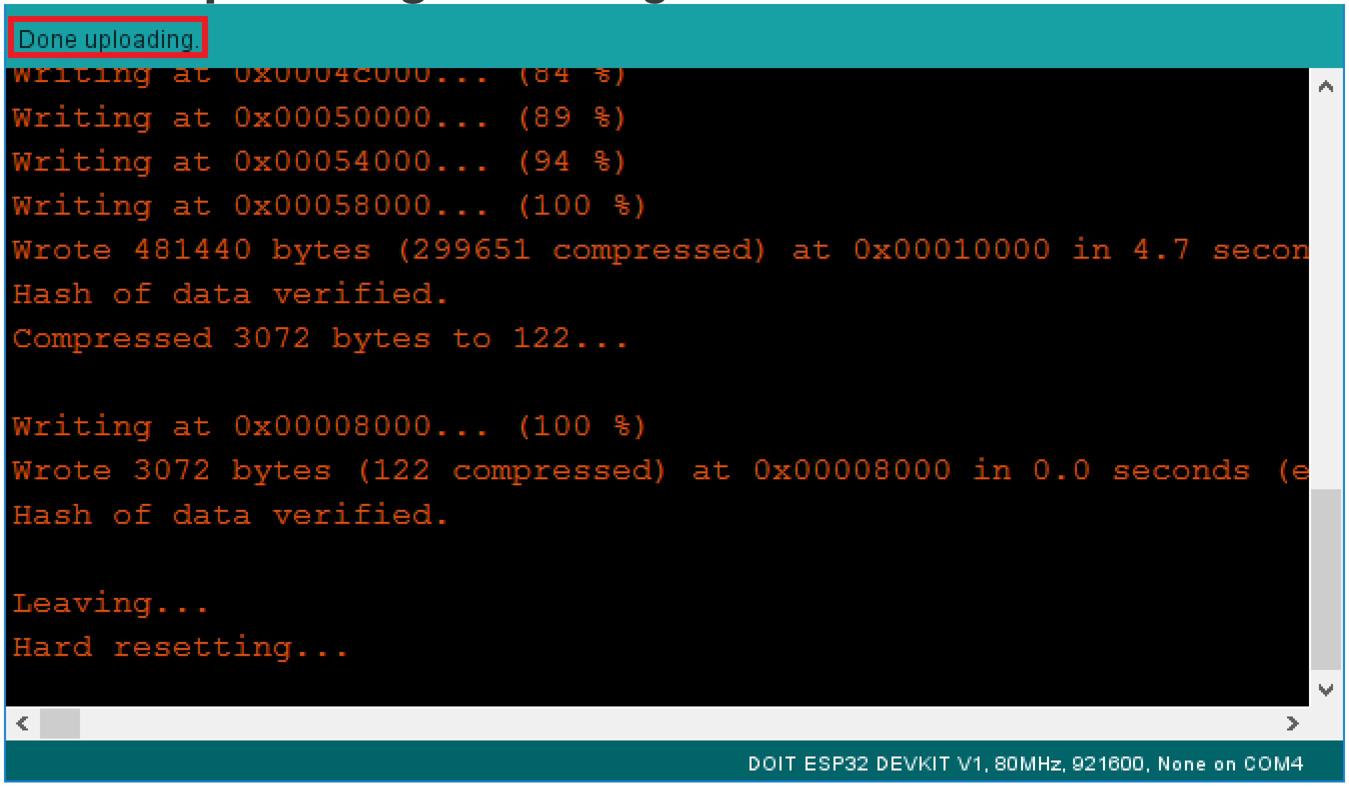
WiFiScan

```
/*
 *  This sketch demonstrates how to scan WiFi networks.
 *  The API is almost the same as with the WiFi Shield library,
 *  the most obvious difference being the different file you need to include:
 */
#include "WiFi.h"

void setup()
{
    Serial.begin(115200);

    // Set WiFi to station mode and disconnect from an AP if it was previousl
    WiFi.mode(WIFI_STA);
    WiFi.disconnect();
    delay(100);

    Serial.println("Setup done");
}

void loop()
```

19 DOIT ESP32 DEVKIT V1, 80MHz, 921600, None on COM4

5. Press the **Upload** button in the Arduino IDE. Wait a few seconds while the code compiles and uploads to your board.

6. If everything went as expected, you should see a "**Done uploading.**" message.



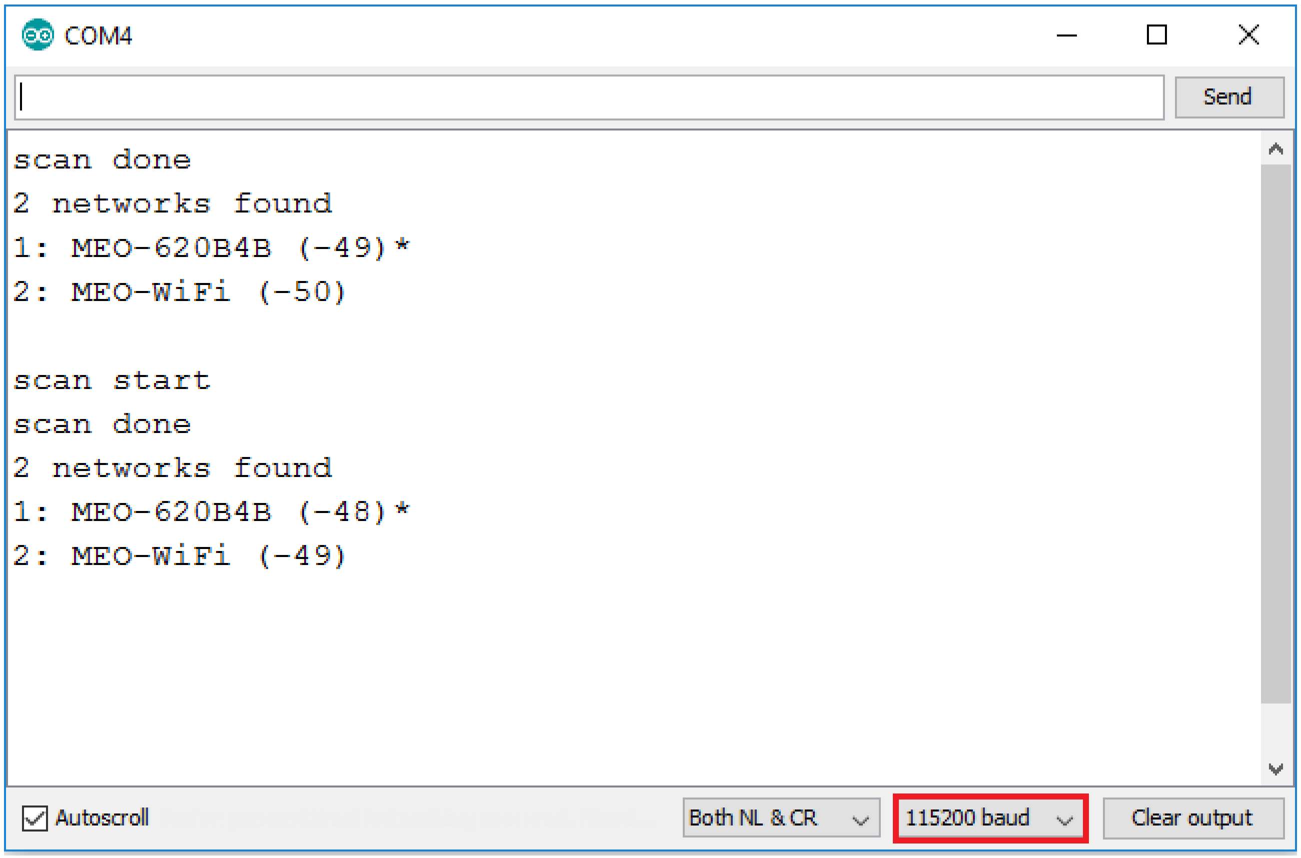
Done uploading.

```
Writing at 0x0004c000... (84 %)
Writing at 0x00050000... (89 %)
Writing at 0x00054000... (94 %)
Writing at 0x00058000... (100 %)
Wrote 481440 bytes (299651 compressed) at 0x00010000 in 4.7 secon
Hash of data verified.
Compressed 3072 bytes to 122...

Writing at 0x00008000... (100 %)
Wrote 3072 bytes (122 compressed) at 0x00008000 in 0.0 seconds (e
Hash of data verified.

Leaving...
Hard resetting...
```

DOIT ESP32 DEVKIT V1, 80MHz, 921600, None on COM4

7. Open the Arduino IDE Serial Monitor at a baud rate of 115200:

8. Press the ESP32 on-board **Enable** button and you should see the networks available near your ESP32:



```
scan done
2 networks found
1: MEO-620B4B (-49)*
2: MEO-WiFi (-50)

scan start
scan done
2 networks found
1: MEO-620B4B (-48)*
2: MEO-WiFi (-49)
```

Test :Blinking Program

```
void setup() {
pinMode(2, OUTPUT);
}
void loop() {
digitalWrite(2, HIGH);
delay(1000);
digitalWrite(2, LOW);
delay(1000);
}
```

Now Upload this Code.

```
int lmf = 2;

int lmb = 3;

int lme = 9;



int rmf = 4;

int rmb = 5;

int rme = 10;


int leftsensor = 6;

int rightsensor = 7;


int leftsensorvalue=0;

int rightsensorvalue=0;
```

```
void setup()

{

//   pinMode(9, OUTPUT);  // this pin will pull the HC-05 pin 34 (key pin) HIGH to switch
module to AT mode

  //digitalWrite(9, HIGH);

Serial.begin(9600);

  pinMode(lmf, OUTPUT);

  pinMode(lmb, OUTPUT);

  pinMode(rmf, OUTPUT);

  pinMode(rmb, OUTPUT);

  pinMode(lme, OUTPUT);

  pinMode(rme, OUTPUT);

  //digitalWrite(led, HIGH);


}


void loop()

{


  // Keep reading from HC-05 and send to Arduino Serial Monitor

 /// if (BTSerial.available())

    //Serial.write(BTSerial.read());
}


  // Keep reading from Arduino Serial Monitor and send to HC-05
  //if (Serial.available())
    //BTSerial.write(Serial.read());


//delay(500);
if ((digitalRead(6) == LOW) && (digitalRead(7) == LOW) )
 {
  forward();
Serial.println("forward");
  }


if ((digitalRead(6) == HIGH) && (digitalRead(7) == LOW) )
{
  left();
Serial.println("left");
  }


if ((digitalRead(6) == LOW) && (digitalRead(7) == HIGH) )
{
  right();
  Serial.println("right");
  }
```

```
if ((digitalRead(6) == HIGH) && (digitalRead(7) == HIGH) )

{

  stop1();

  Serial.println("stop1");

  }



/*

  leftsensor=digitalRead(6);

  rightsensor=digitalRead(7);

  Serial.println(leftsensor);

  Serial.println(rightsensor);

*/

}








void left()


{

  analogWrite(lme, 100);
  digitalWrite(lmf, HIGH);

  digitalWrite(lmb, LOW);

  analogWrite(rme, 100);

  digitalWrite(rmf, LOW);

  digitalWrite(rmb, HIGH);

  //for testing

  }


void right()


{

  analogWrite(lme, 100);

  digitalWrite(lmf, LOW);

  digitalWrite(lmb, HIGH);

  analogWrite(rme, 100);

  digitalWrite(rmf, HIGH);

  digitalWrite(rmb, LOW);

  //for testing

  }




void backward()
```

```
{
 analogWrite(lme, 100);
 digitalWrite(lmf,  LOW);
 digitalWrite(lmb, HIGH);
 analogWrite(rme, 100);
 digitalWrite(rmf, LOW);
 digitalWrite(rmb, HIGH);
 //for testing
 }

 void forward()

 {
 analogWrite(lme, 100);
 digitalWrite(lmf, HIGH);
 digitalWrite(lmb, LOW);
 analogWrite(rme, 100);
 digitalWrite(rmf, HIGH);
 digitalWrite(rmb, LOW);
 //for testing

  }
```

```
void stop1()


{
analogWrite(lme, 100);
digitalWrite(lmf, LOW);
digitalWrite(lmb, LOW);
analogWrite(rme, 100);
digitalWrite(rmf, LOW);
digitalWrite(rmb, LOW);
//for testing


  }
```

How to use this robot.

Give the 12v power supply to this robot

And place it on black path or line where background will be white.

Now it will start follow the black line.