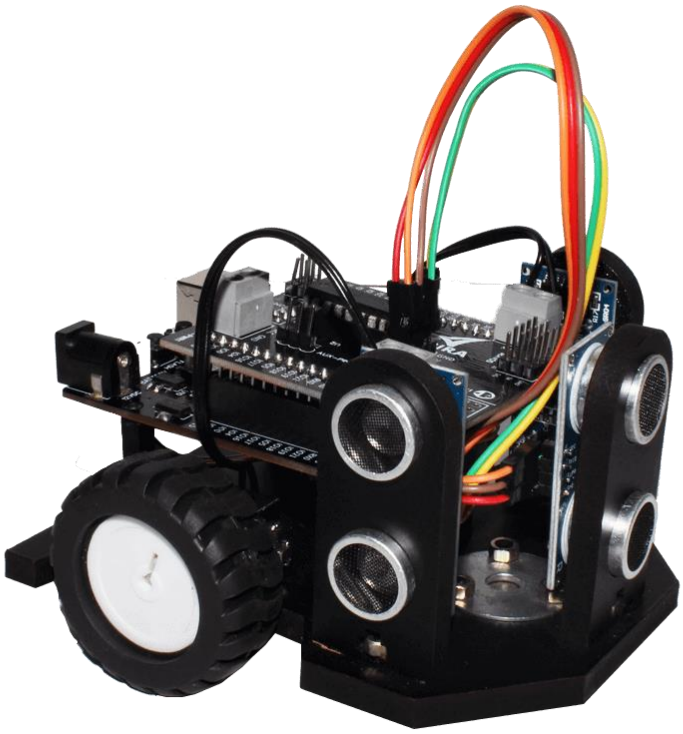
Micro Mouse Manual



Assembly And Wiring of Micro mouse.

These are some parts of the robot which have to be attached with robot to make a complete Micro mouse.

Step 1.

Attach the castor wheel in the front side of the robot. Insert the castor wheel from the top of the robot frame to bottom side.

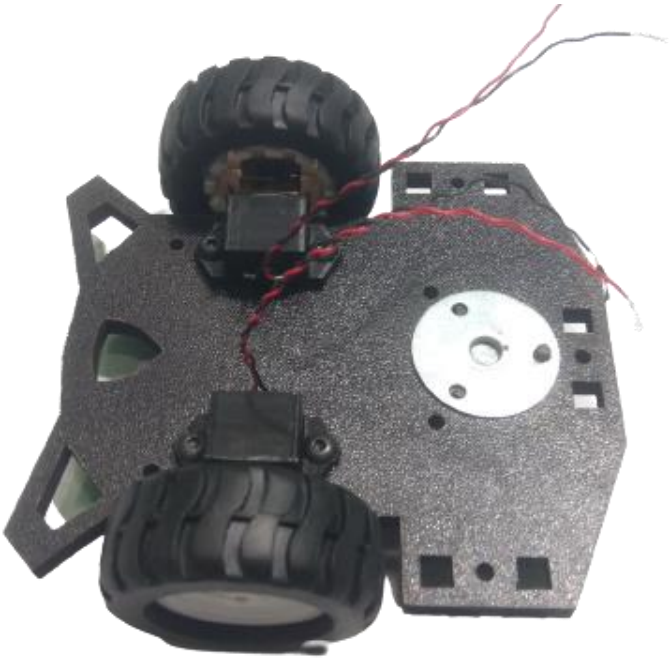
And the tight it with allen bolt and nut using allen key.

Step 2.

Attach the N20 wheel with the N20 motor. Please see find the symmetric cuts in wheel acoord to the shaft of the motor then attach the wheel with the motor.

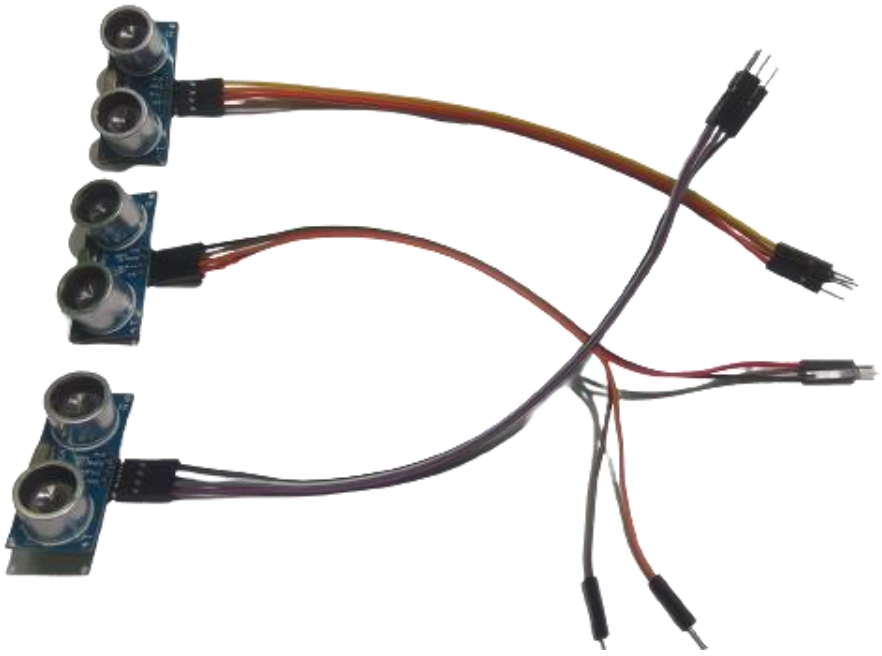
Step 3.

Attach the N20 dc gear motor on the robot chassis use N20 motor clamp to attach the motor with the bottom chassis of the robot.



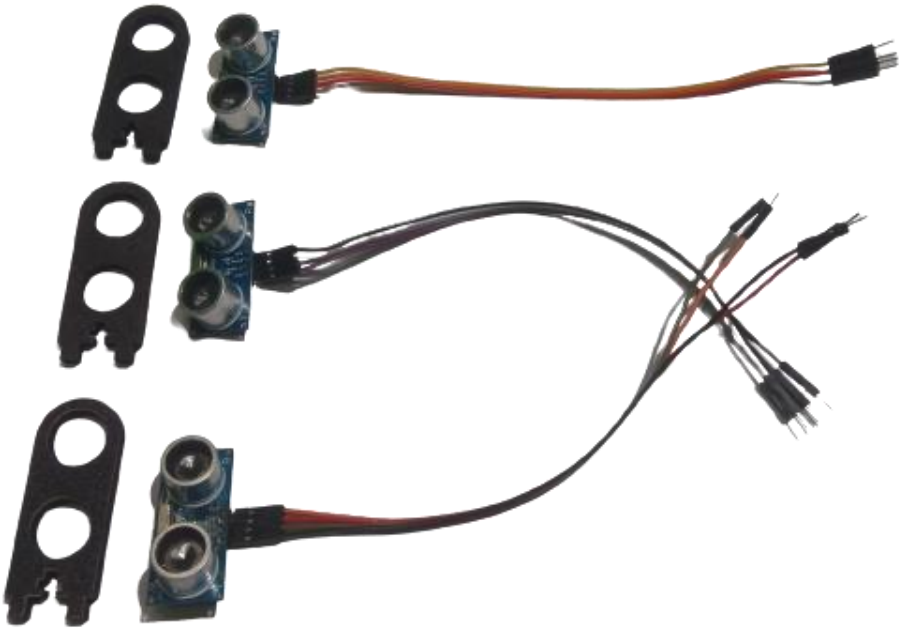
Step 4.

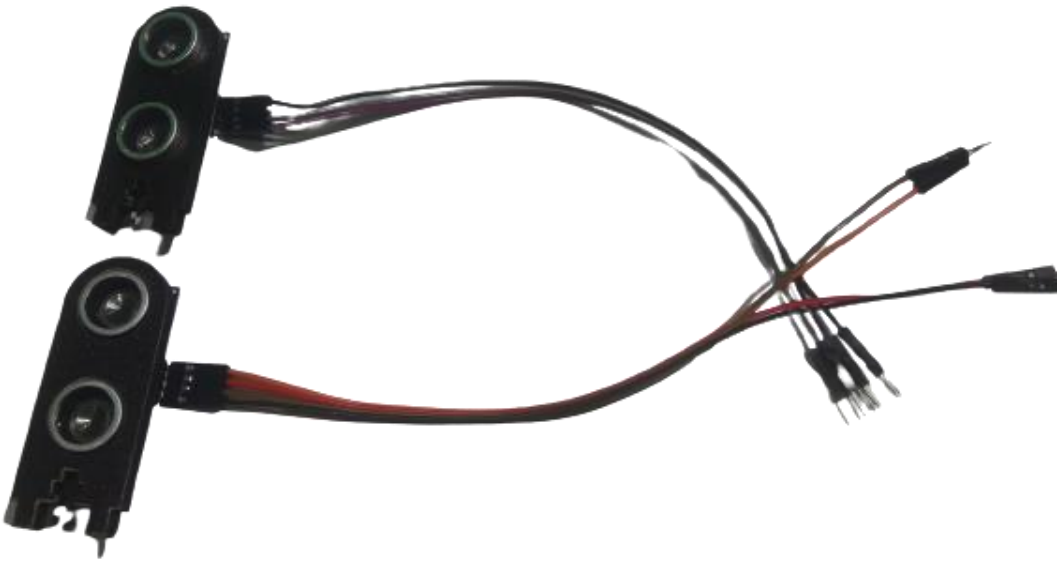
Connect all the Ultrasonic Sensor with female to male jumper wire as shown in figure.

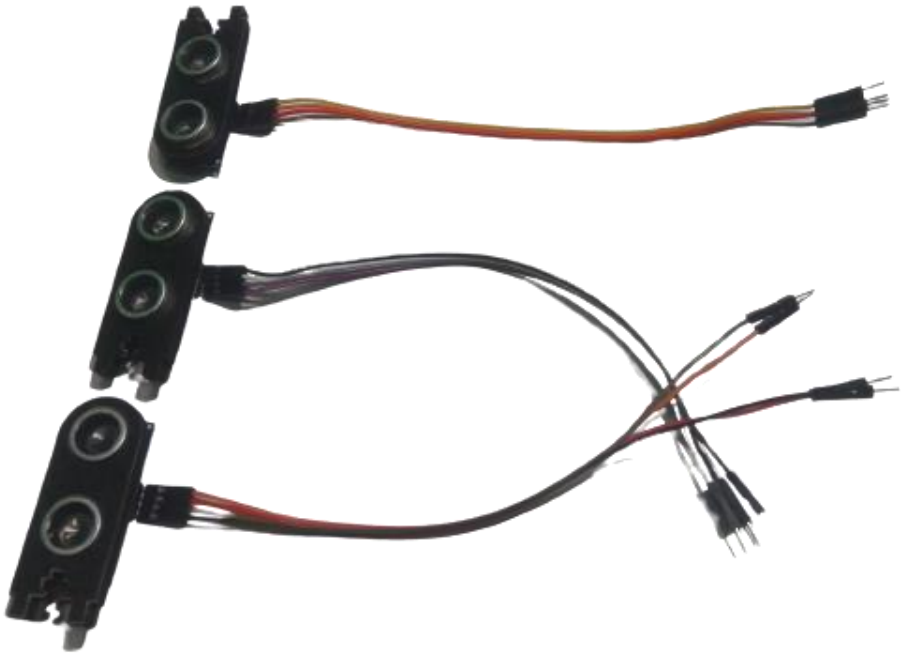


Step 5.

Insert the ultrasonic sensor with the ultrasonic sensor holder frame part of the robot. As shown in figure.



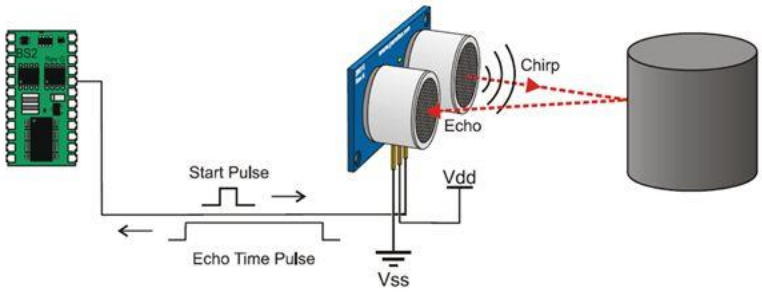


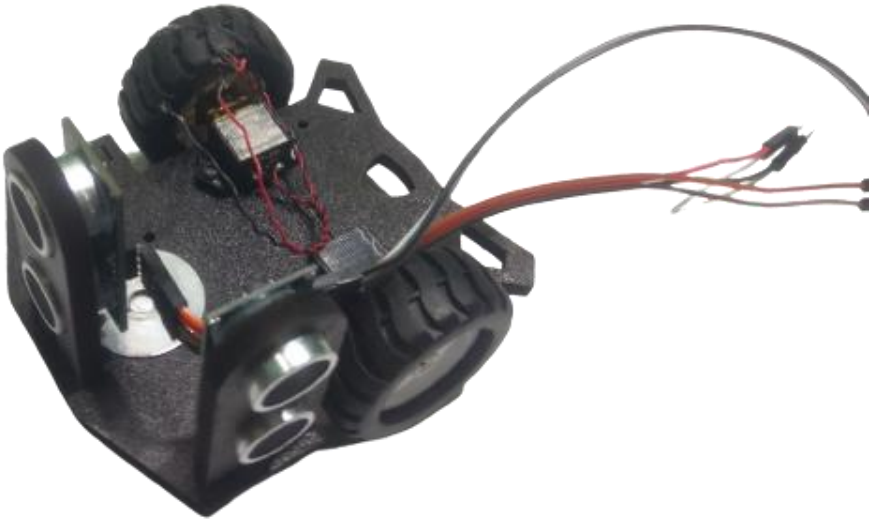


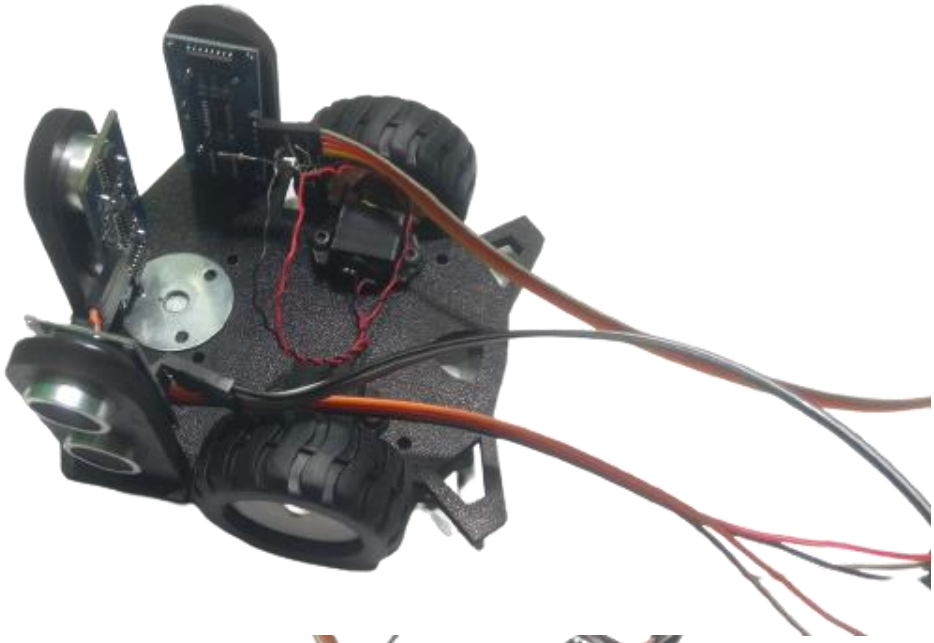
Step 6.

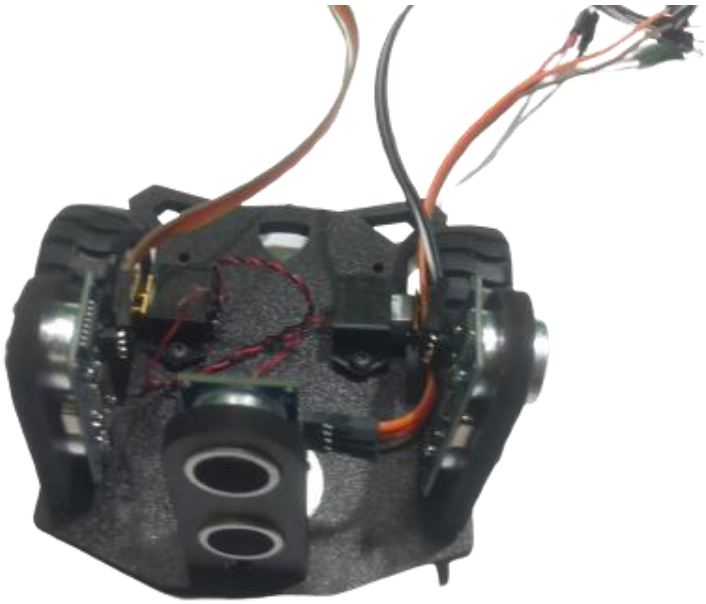
Then insert these ultrasonic holder in the chassis of robot.

Keep the ultrasonic sensor towards outside o the robot.









Step 7.

Not attach the Arduino Uno R3 micro controller on the center of the robot from the top side.

Keep the programming port toward back side of the robot, so it’s easy to reprogramme it.

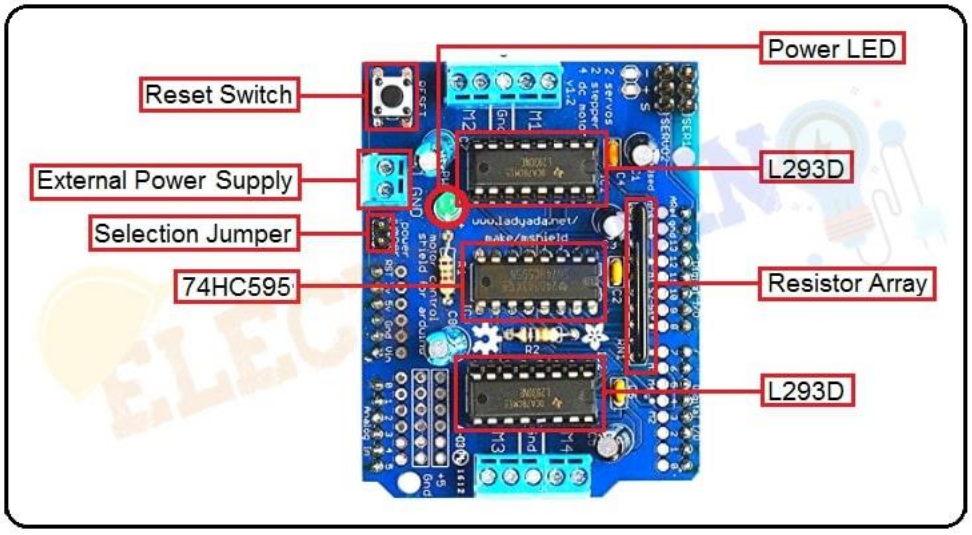
And then finally attach the nut and bolts. Pass it through the Arduino Uno R3 holes to robot chassis holes then tight the nut from the bottom side of the robot.

Step 8.

Solder the male header pins on the L293d motor diver shield.

Brake 3 header pins with 6 pins in each length.

Now solder the header pins in the way by which you can attach ultrasonic female jumper to the soldered header pins of motor driver shield.



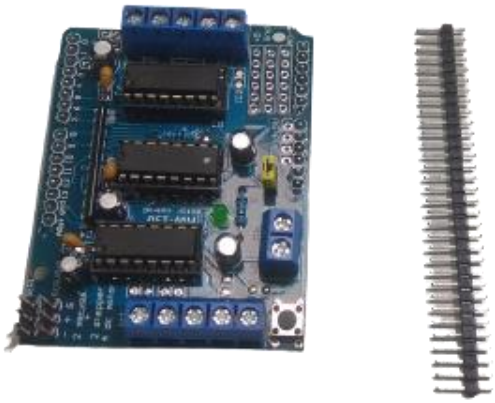


Fig. In this figure you have to solder this header pins.

Fig. In this figure header pins are soldered.

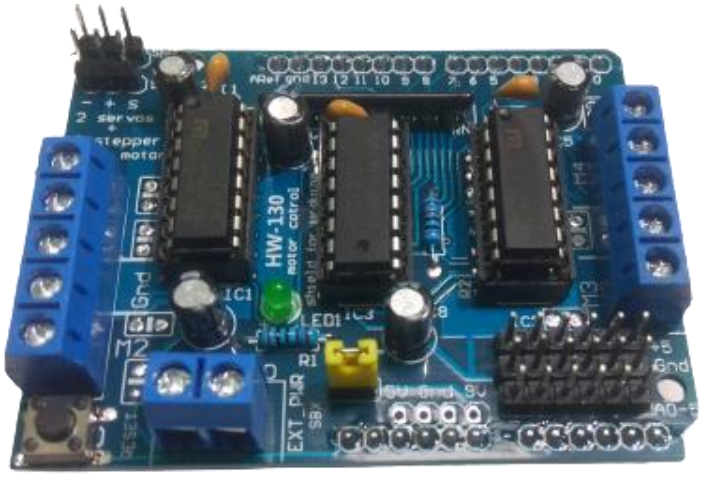
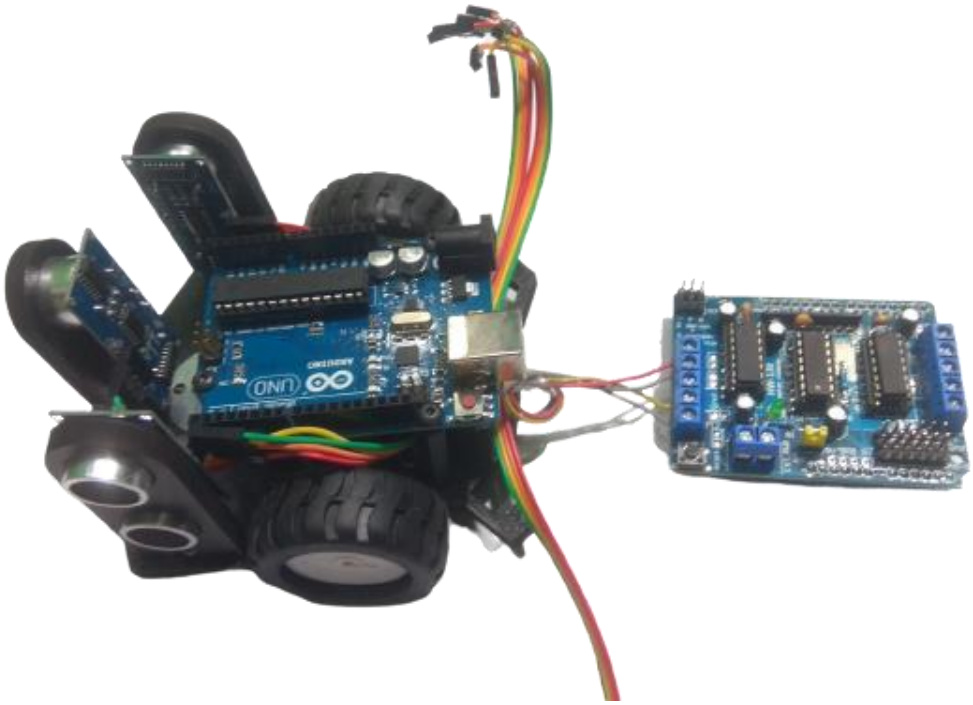


Fig. In this figure header pins are soldered.

Step 9.

Now attach the wire of both N20 motor to the motor driver shield.

There are two wire in one N20 motor connet it to Motor 1 Socket and Motor 2 Socket on the L293d motor driver shield.

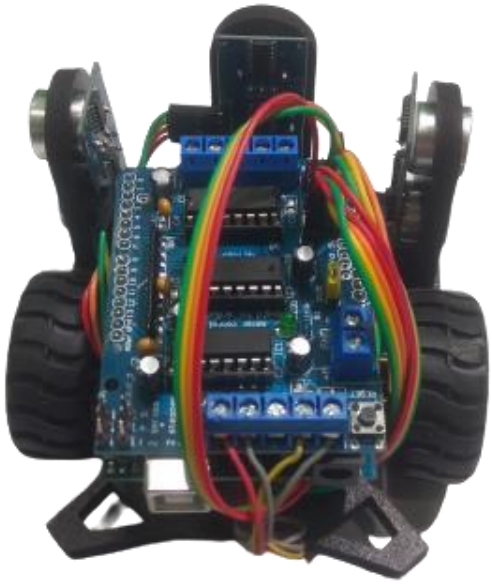


Step 10.

Then attach the motor driver shield on the top of the Arduino Uno R3.

Match all the male header pins of L293d motor driver shield with the females pin of Arduino Uno R3 .

Also connect all the wires of all three ultrasonic sensors on L293d motor driver shield.



And now the assembly part is completed.

Connections.

Connect all positive 5v of ultrasonic sensor to 5v of L293d motor driver shield.

And all negative or ground of ultrasonic sensor to negative or ground of motor driver shield.

Echo pin of left ultrasonic sensor to A0 pin on L293d motor driver shield.

Trigger pin of left ultrasonic sensor to A1 pin on L293d motor driver shield.

Echo pin of center ultrasonic sensor to A2 pin on L293d motor driver shield.

Trigger pin of center ultrasonic sensor to A3 pin on L293d motor driver shield.

Echo pin of right ultrasonic sensor to A4 pin on L293d motor driver shield.

Trigger pin of right ultrasonic sensor to A5 pin on L293d motor driver shield.

And give to 12v supply to the L293d motor driver shield.

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

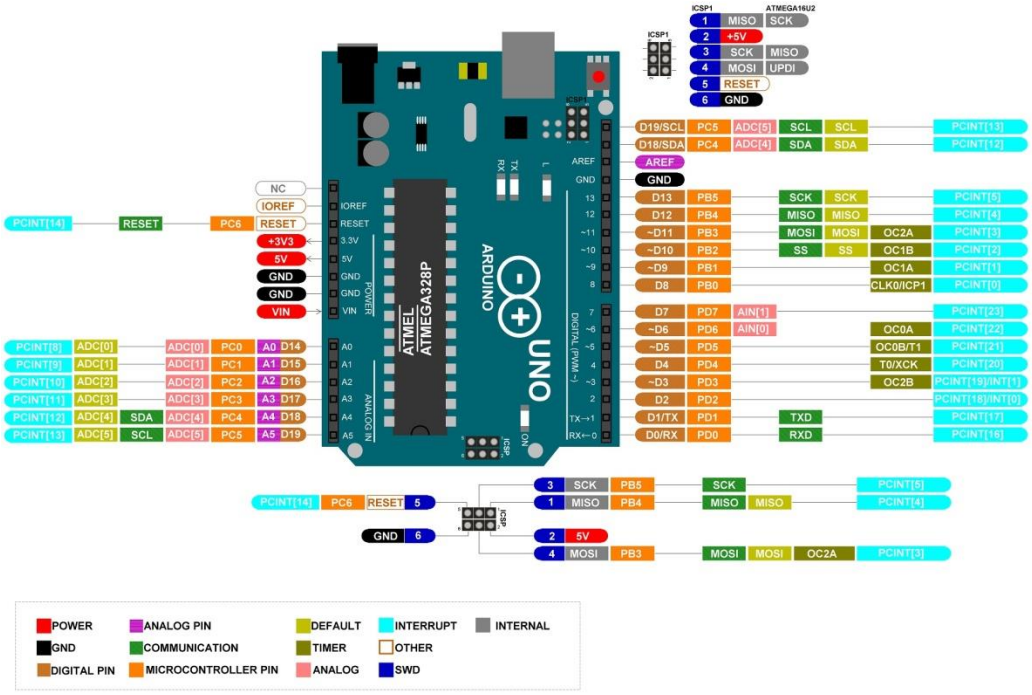
Arduino UNO is a microcontroller board based on the **ATmega328P**. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the 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.

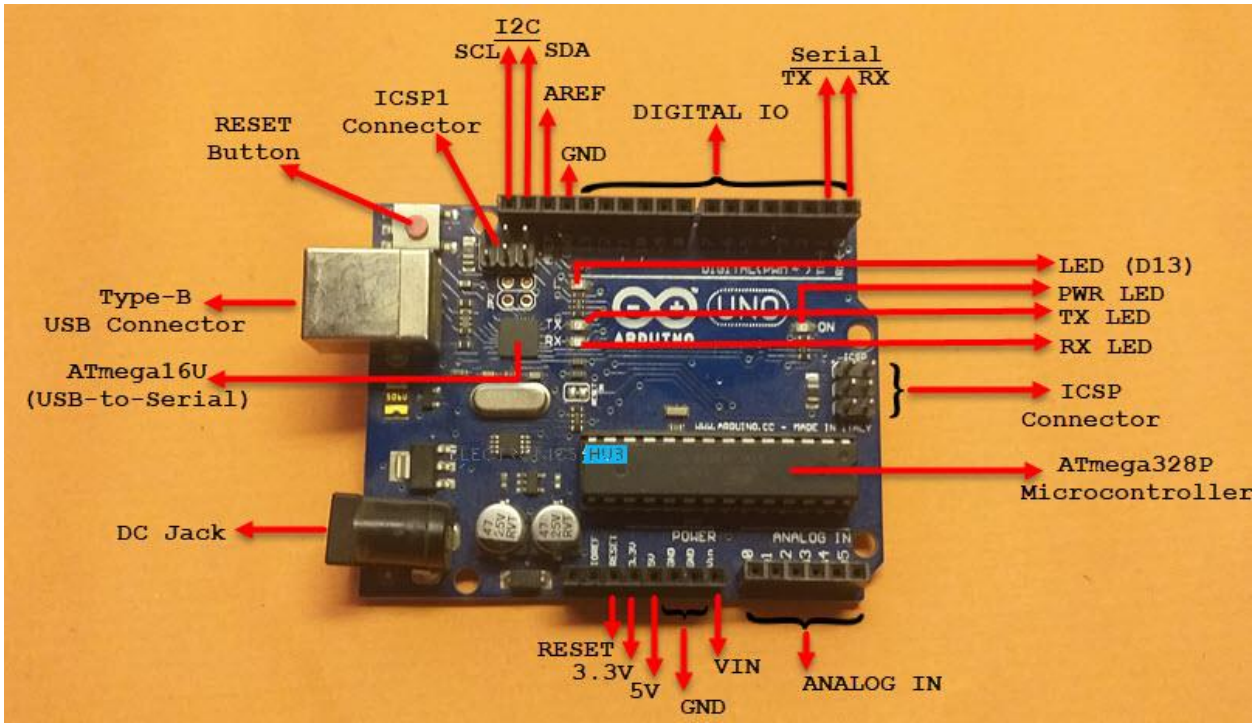
Features:

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 (mA)	50

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

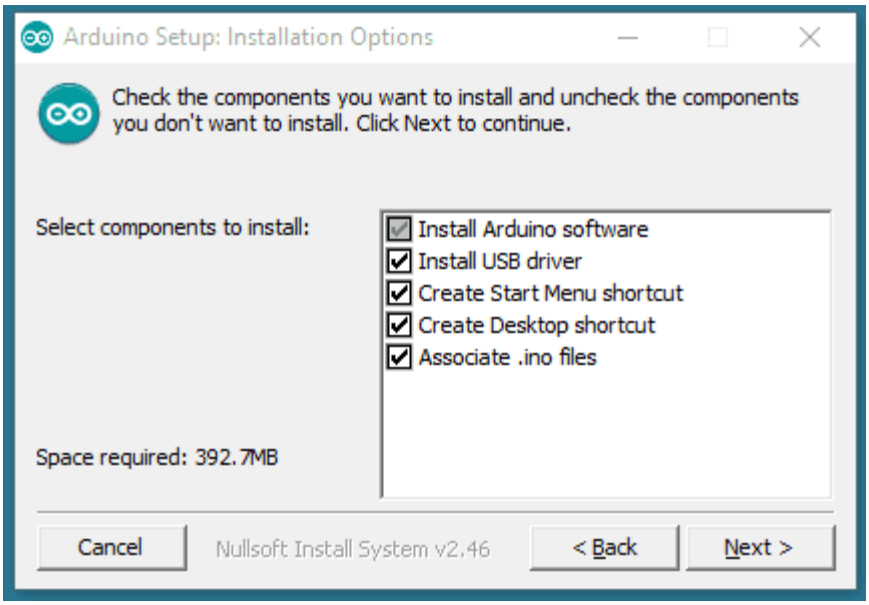
Arduino Uno pin details.



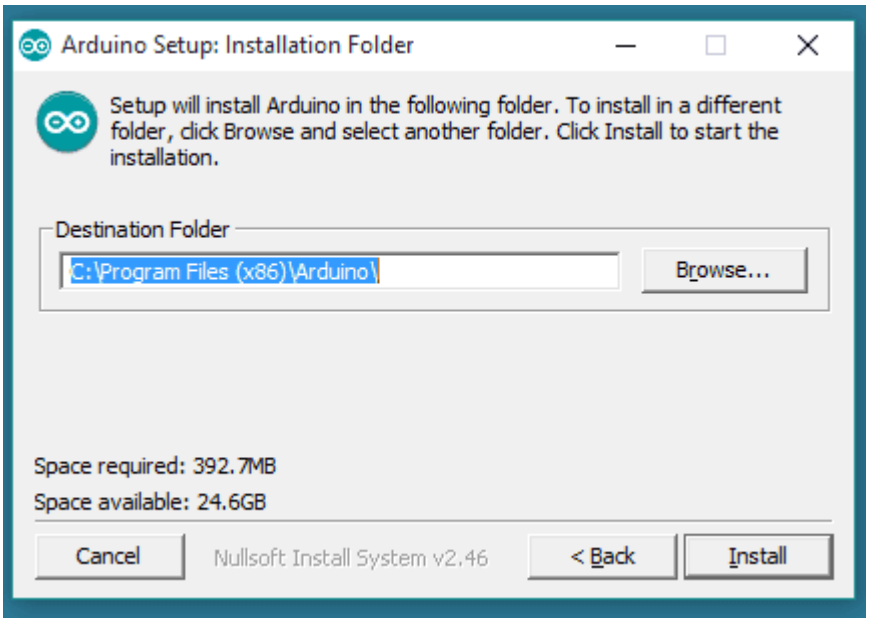


How to install the Arduino Software (IDE)

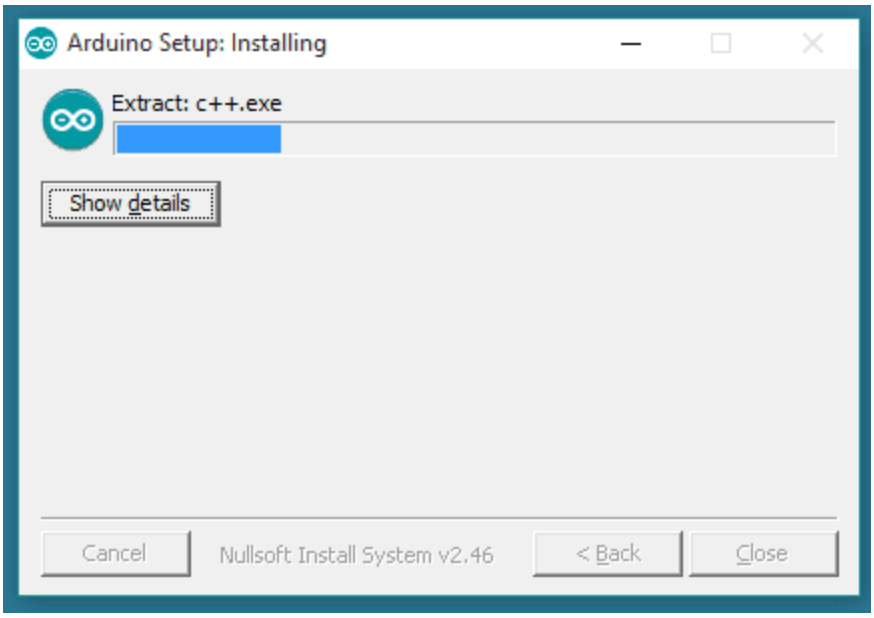
This document explains how to install the Arduino Software (IDE) on Windows machines.
Download the Arduino Software (IDE)
Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything 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.



Choose the components to install.



Choose the installation directory.



Installation in progress.

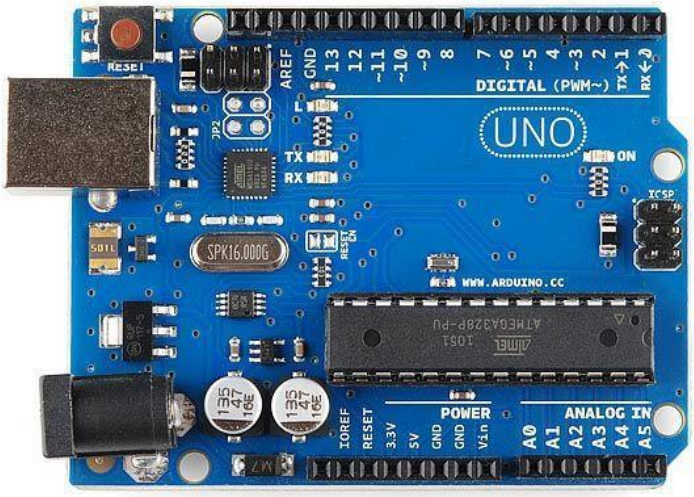
The process will extract and install all the required files to execute properly the Arduino Software (IDE)

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

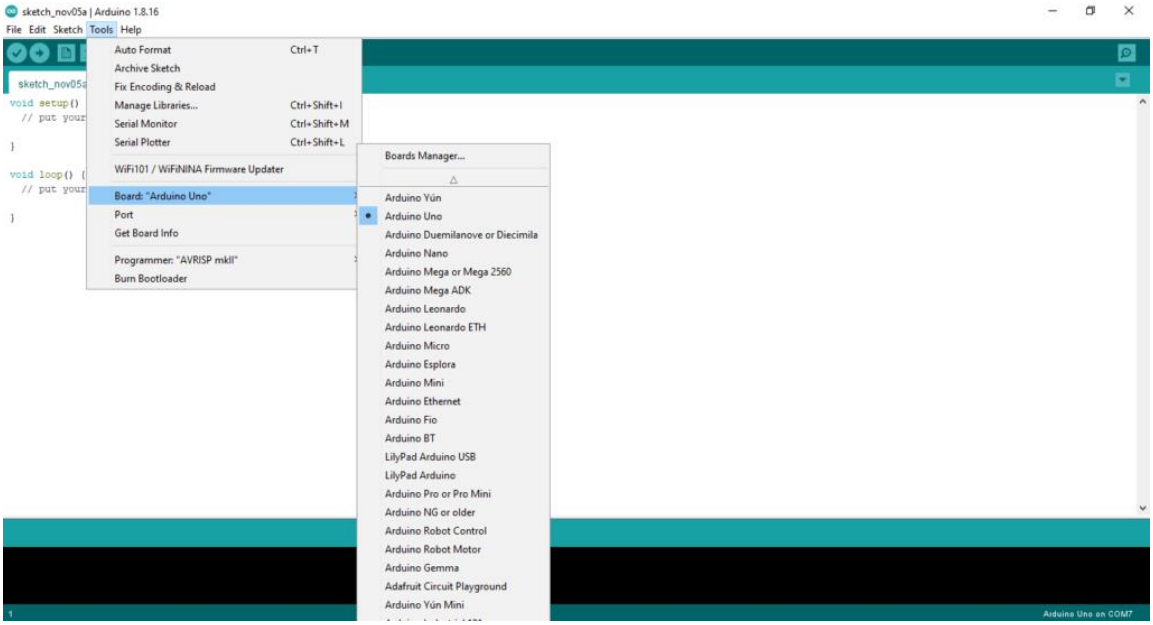
First, we will discuss the problems related to the DIP board.

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.

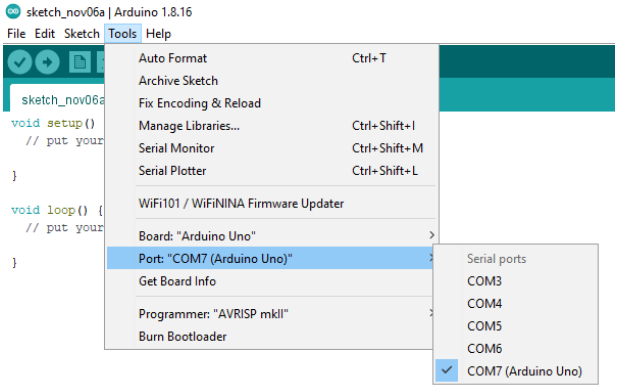
Following are some possible solutions: –

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.



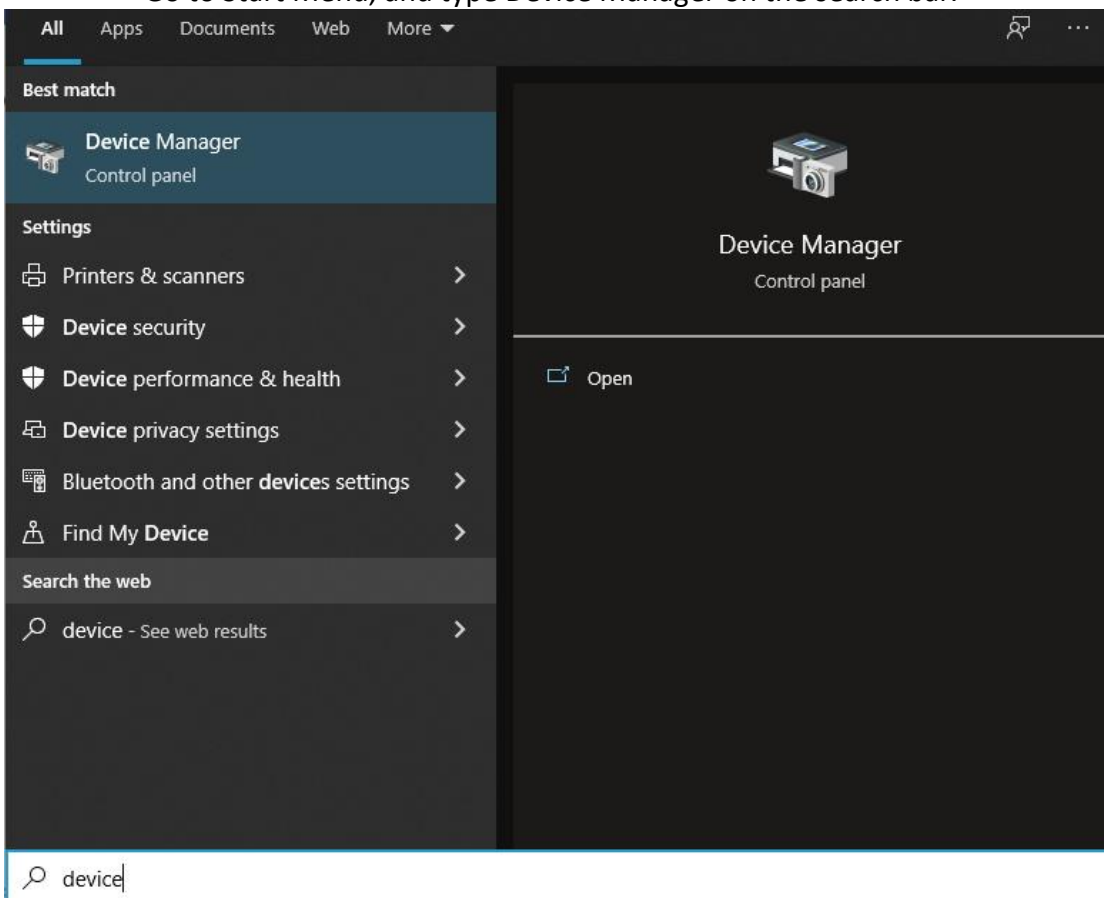
- Check if the correct port is selected by navigating to Tools > Port menu (if your port doesn't appear, try restarting the IDE with the board connected to the computer), then select the port which has Arduino UNO written with it. If you don't seem to have a serial port for your Arduino board, see the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



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

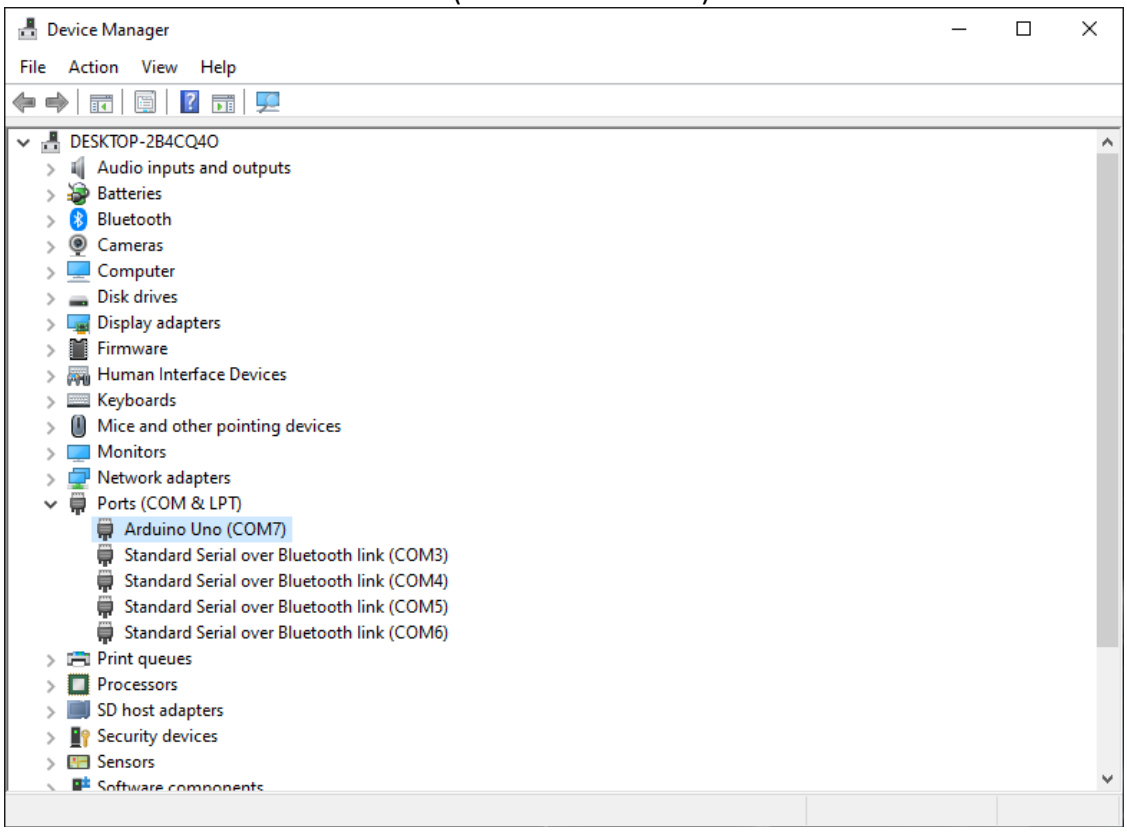
Windows system, and your Arduino drivers are not up to date you will usually get Arduino ports not showing up. If this is the case, update using these steps below.

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

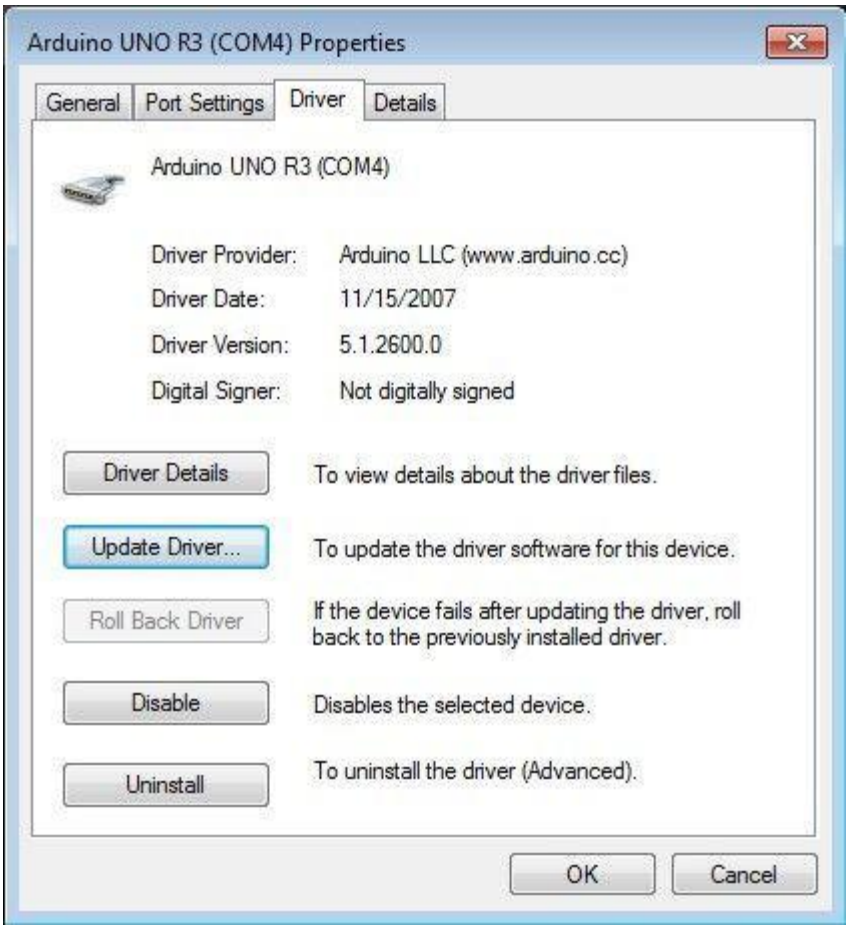


- Click the Device Manager icon to open a new window.

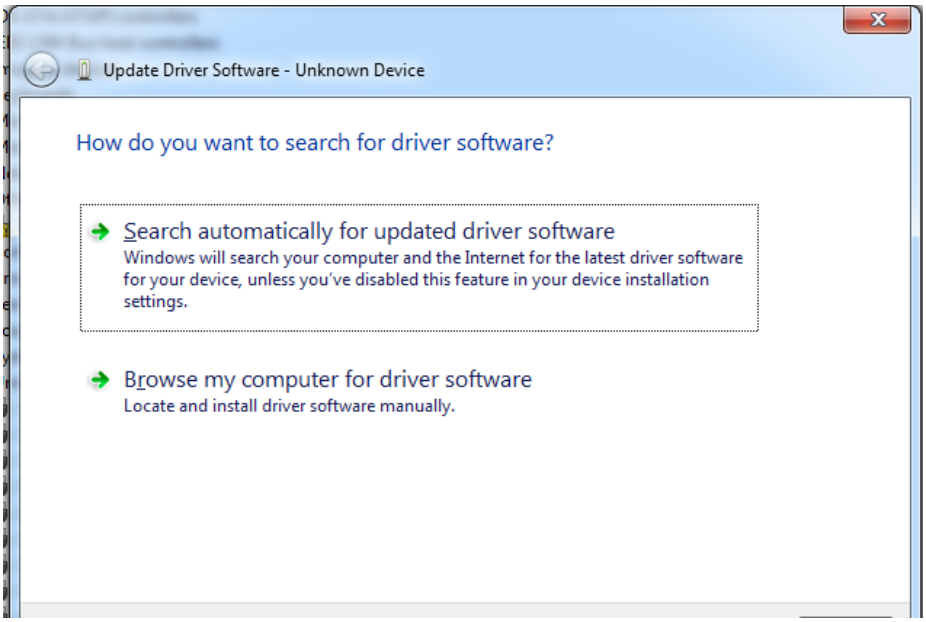
- Scroll down to Ports (COM & LPT) and click on to expand.



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



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



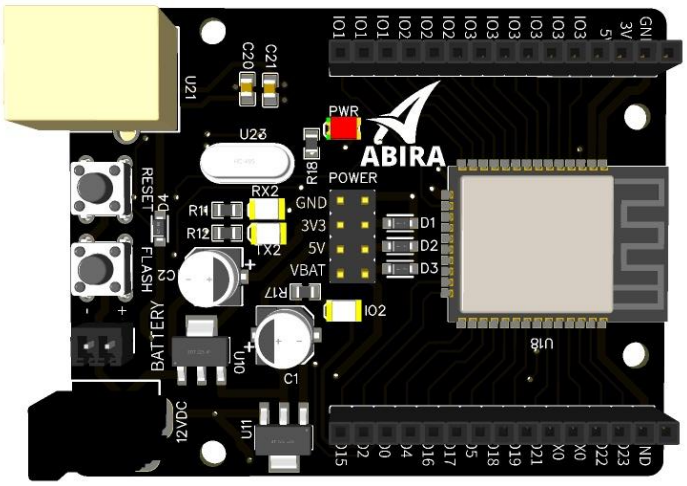
- Windows will begin to update.
- After the update is done, “Windows has successfully updated your driver software” will be displayed on your screen.
- Run a final check to be sure the driver was updated correctly.

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



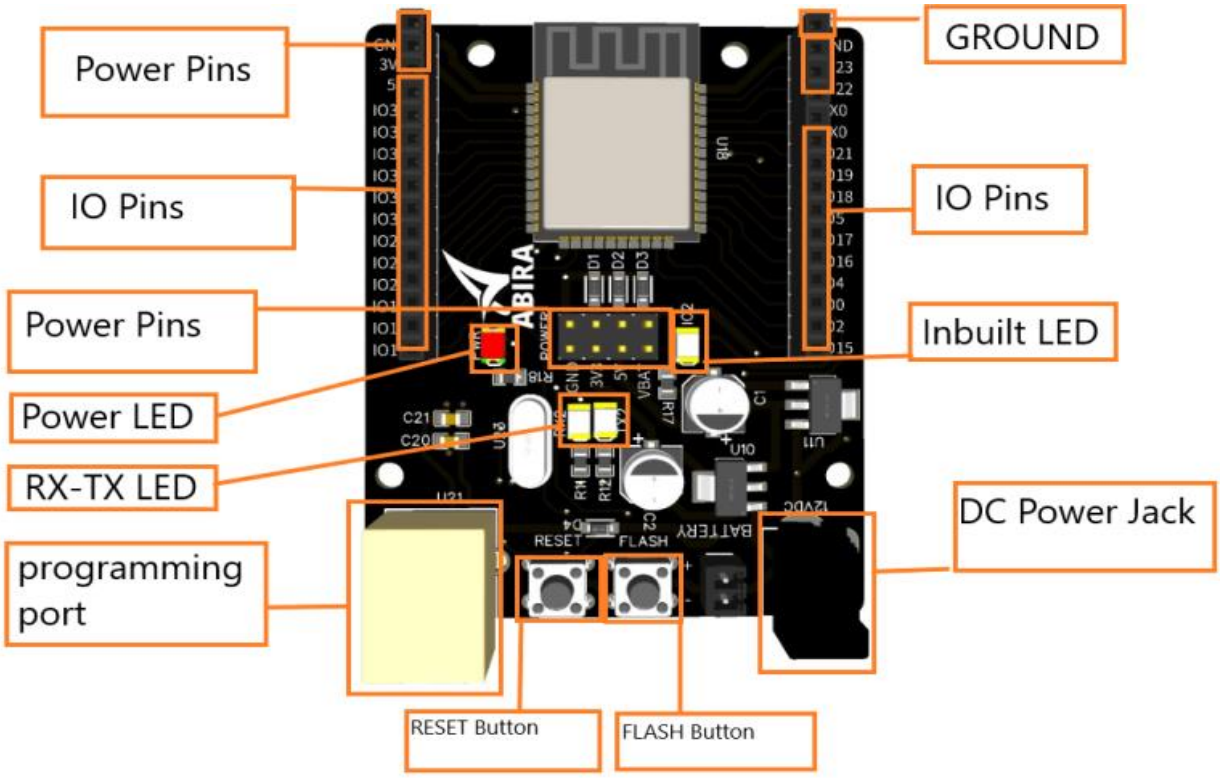
Abira board SV1.0 is best development board for getting into the field of Electronics and learning with hands on Experience using such a powerful hardware. Including the features which comes with this board makes learning with it so much fun and the possibilities is endless.

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

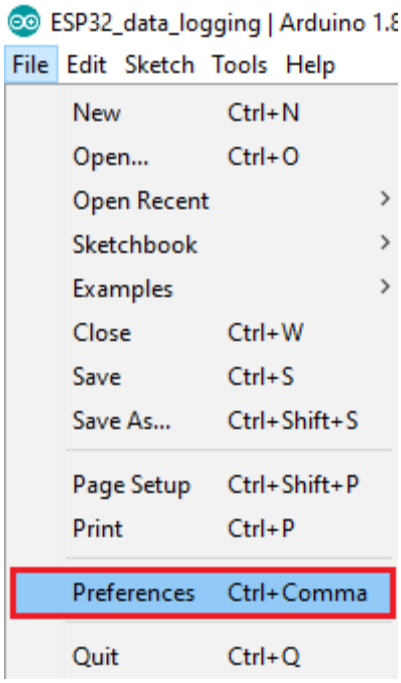
https://docs.Arduino.cc/software/ide-v1/tutorials/Windows
Then install it or extract it. After that you will get Arduino exe or Arduino application.



Installing ESP32 Add-on in Arduino IDE

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

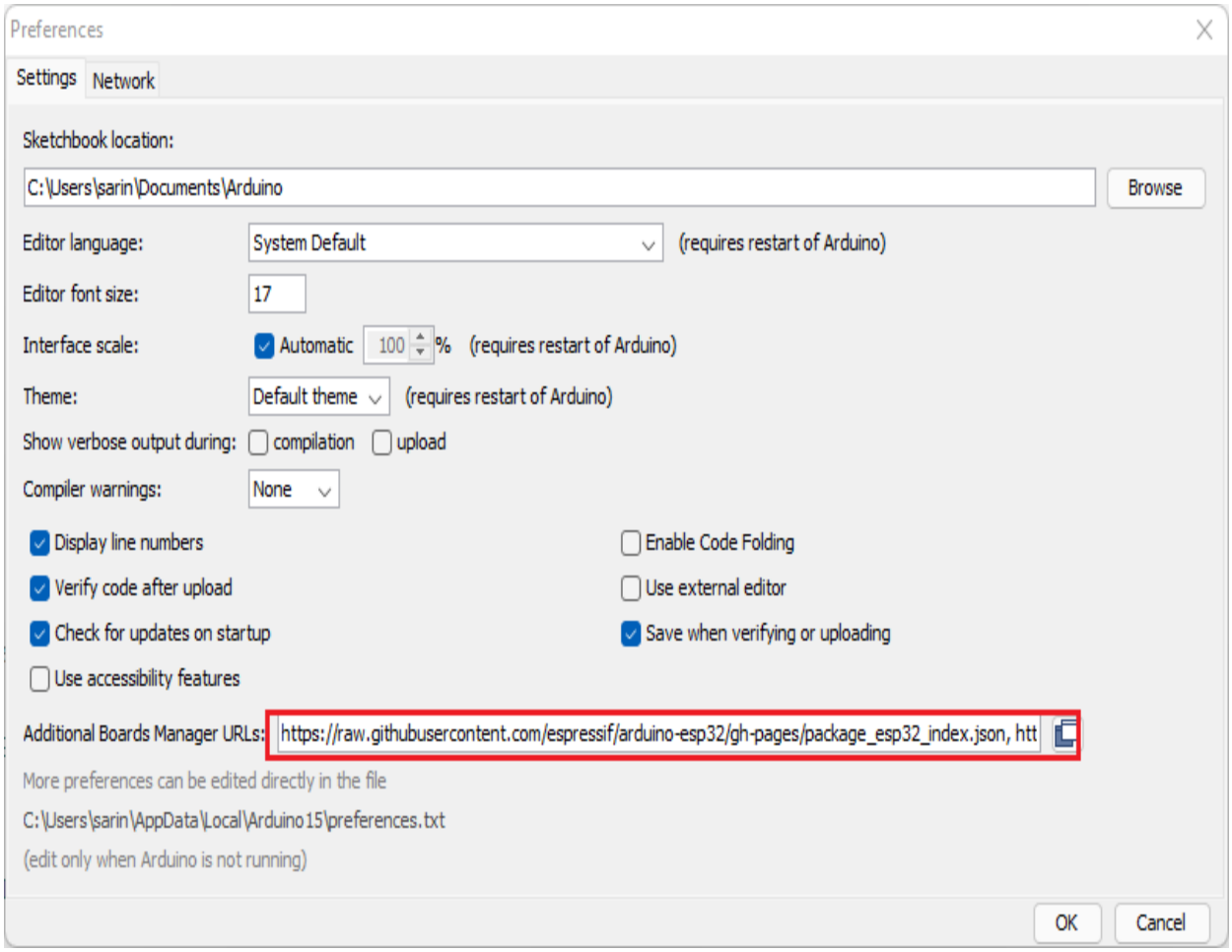
1. In your Arduino IDE, go to **File**> **Preferences**



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

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

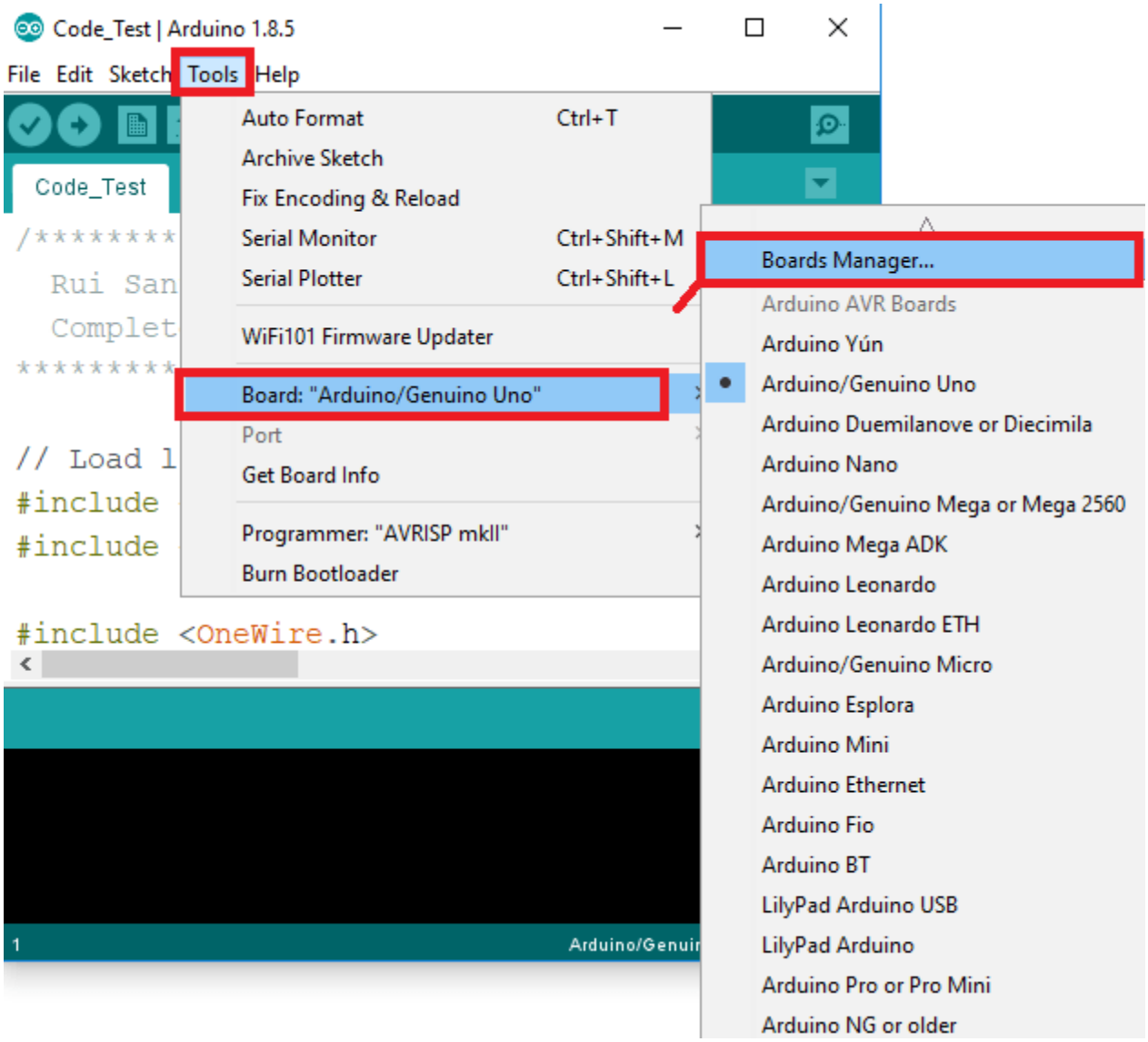
   Then, click the “OK” button:



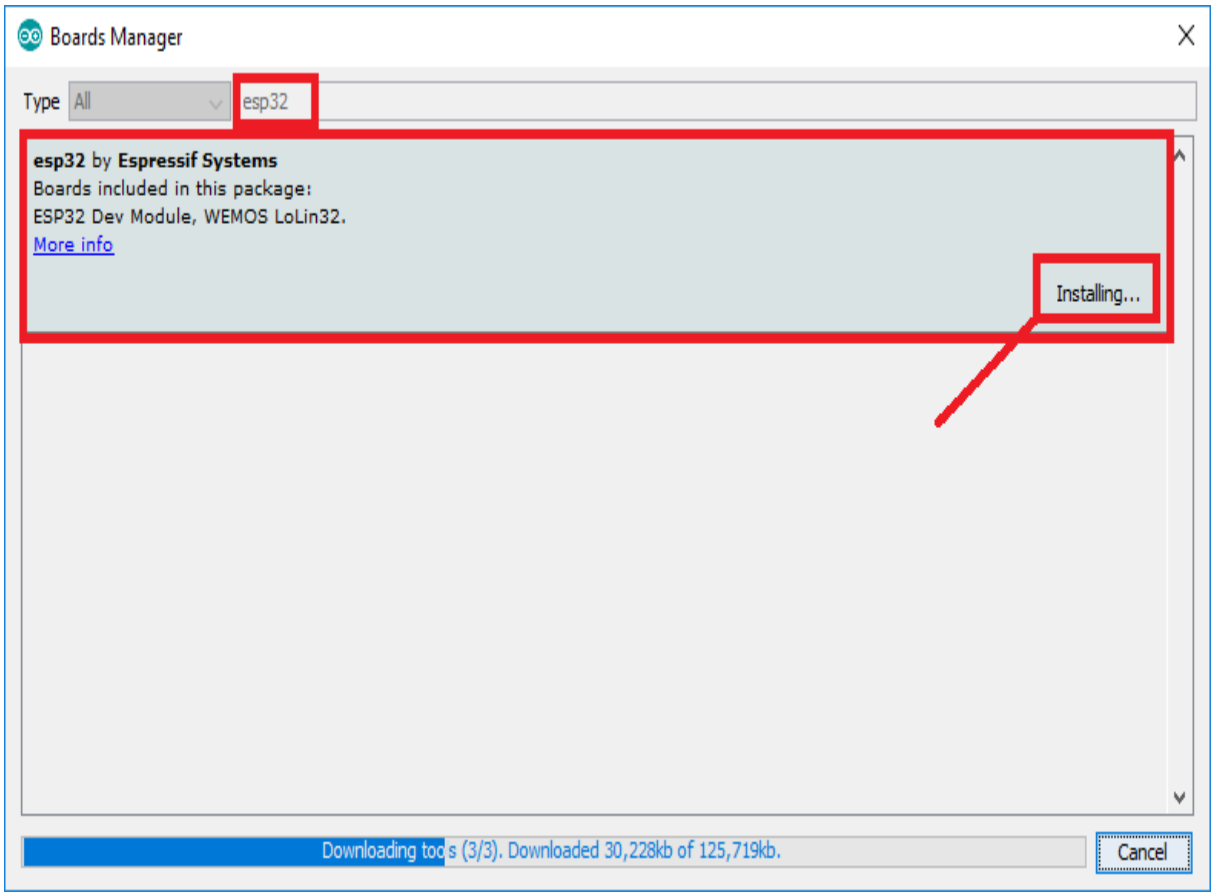
Note: if you already have the ESP8266 boards URL, you can separate the URLs with a comma as follows:

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

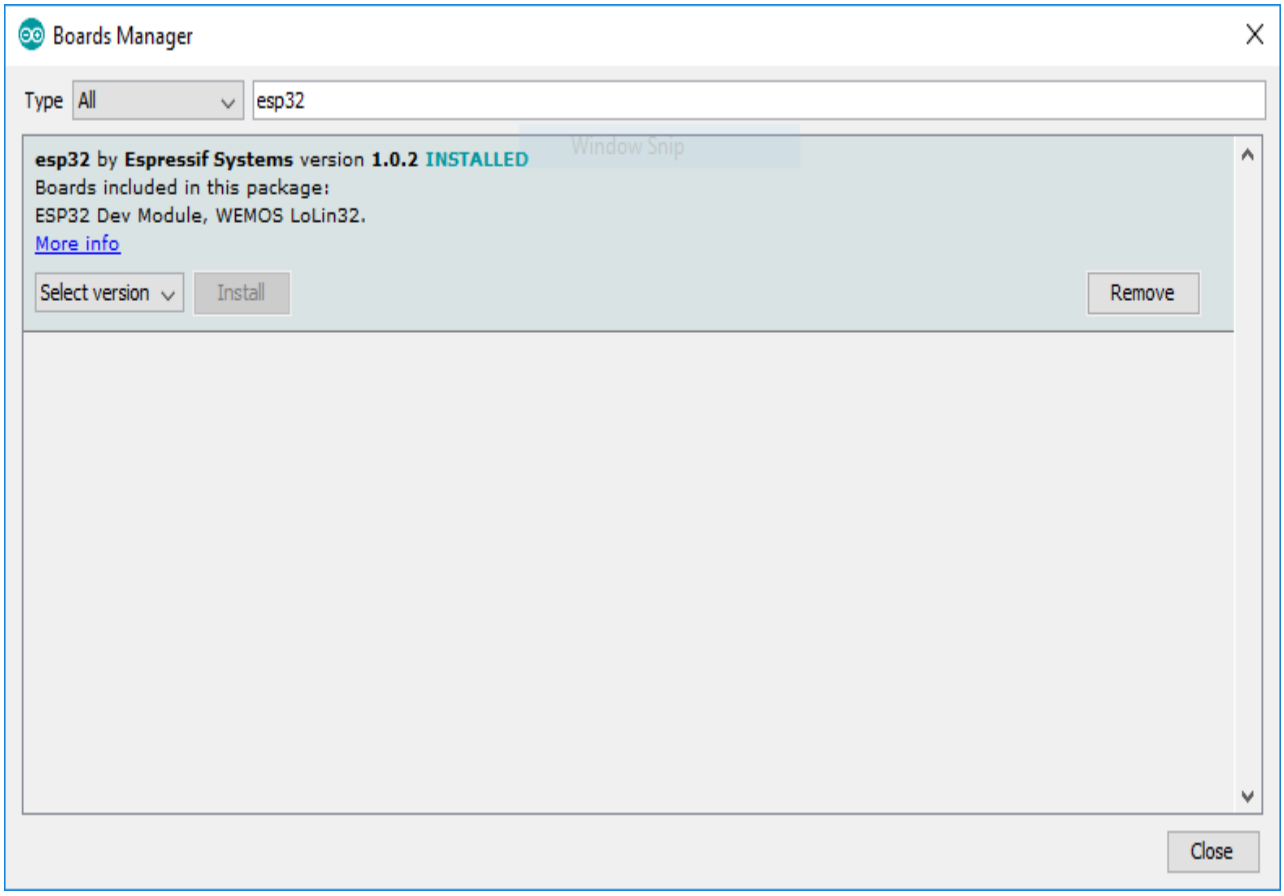
3. Open the Boards Manager. Go to **Tools** > **Board** > **Boards Manager…**



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



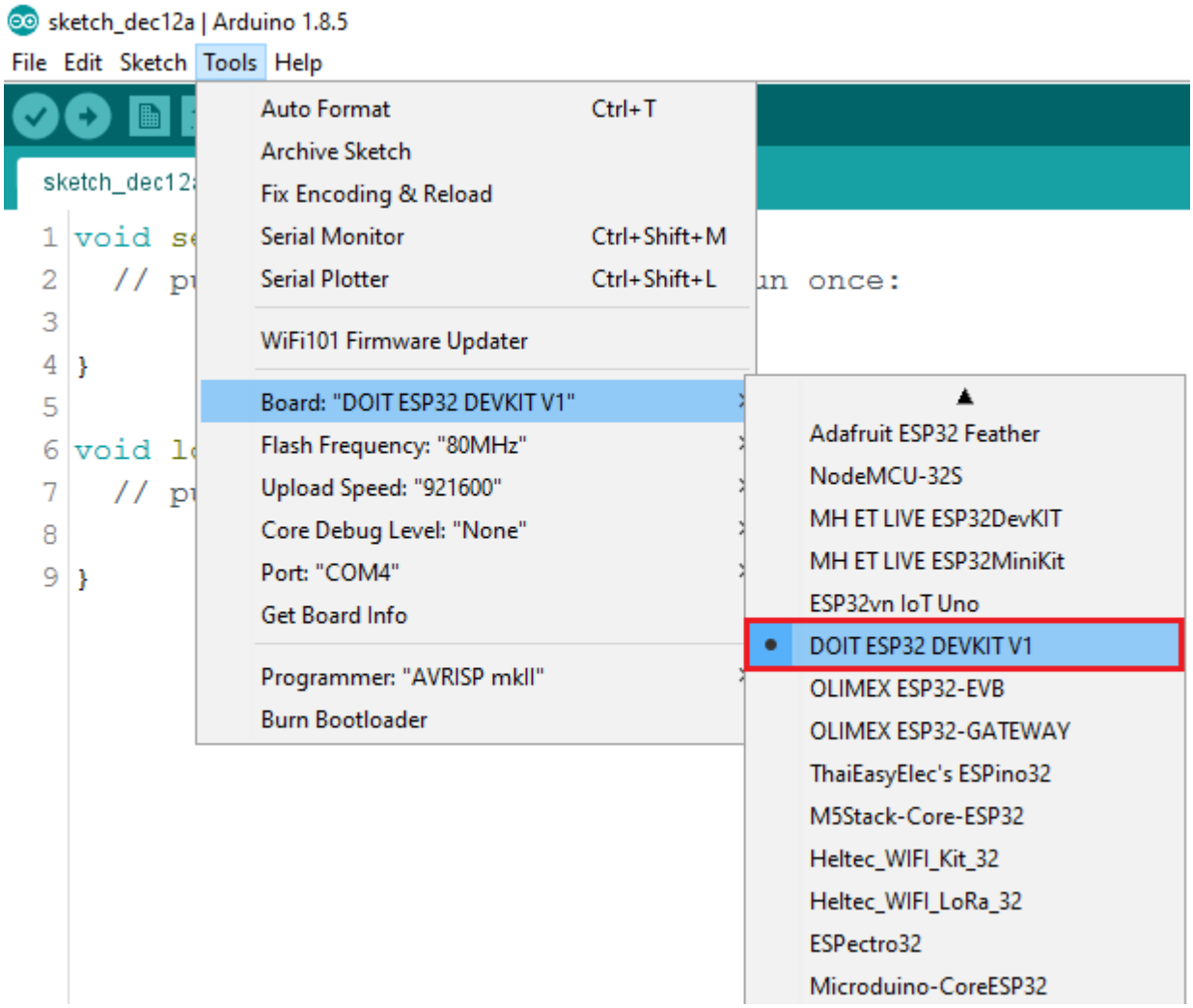
5. That's it. It should be installed after a few seconds.



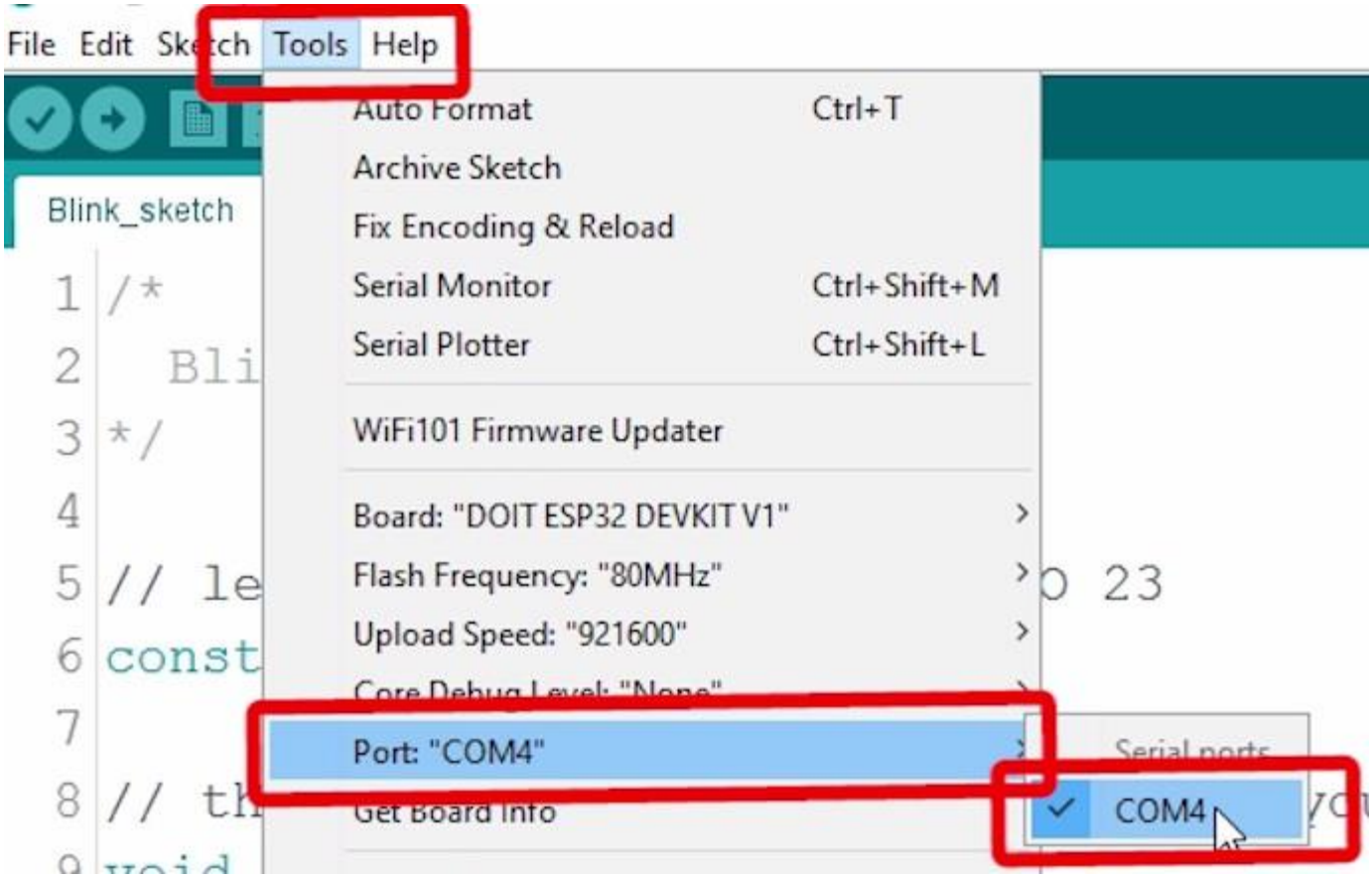
TESTING

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

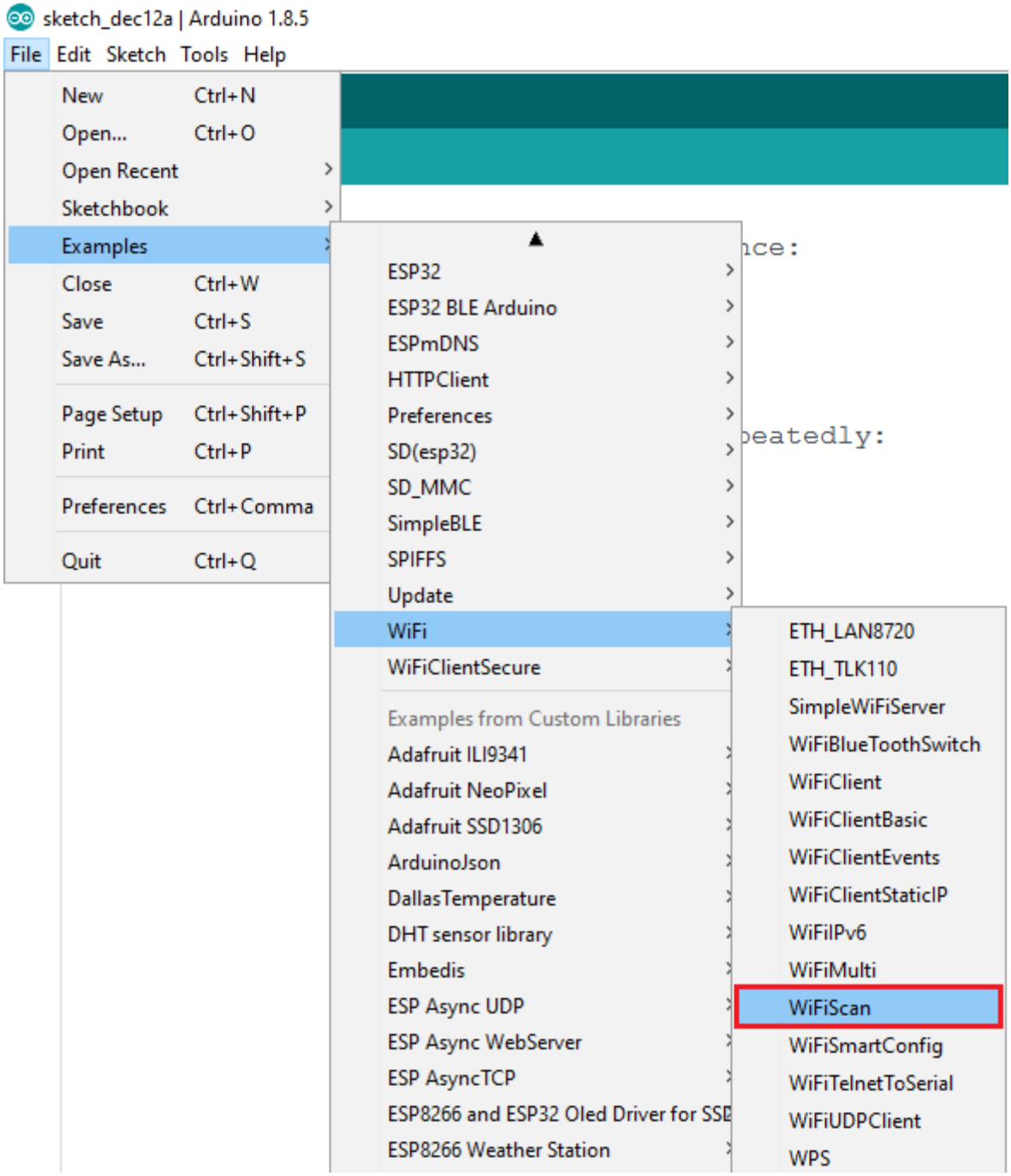
1. Select your Board in **Tools** > **Board** menu (in my case it's the **DOIT ESP32 DEVKIT V1**)



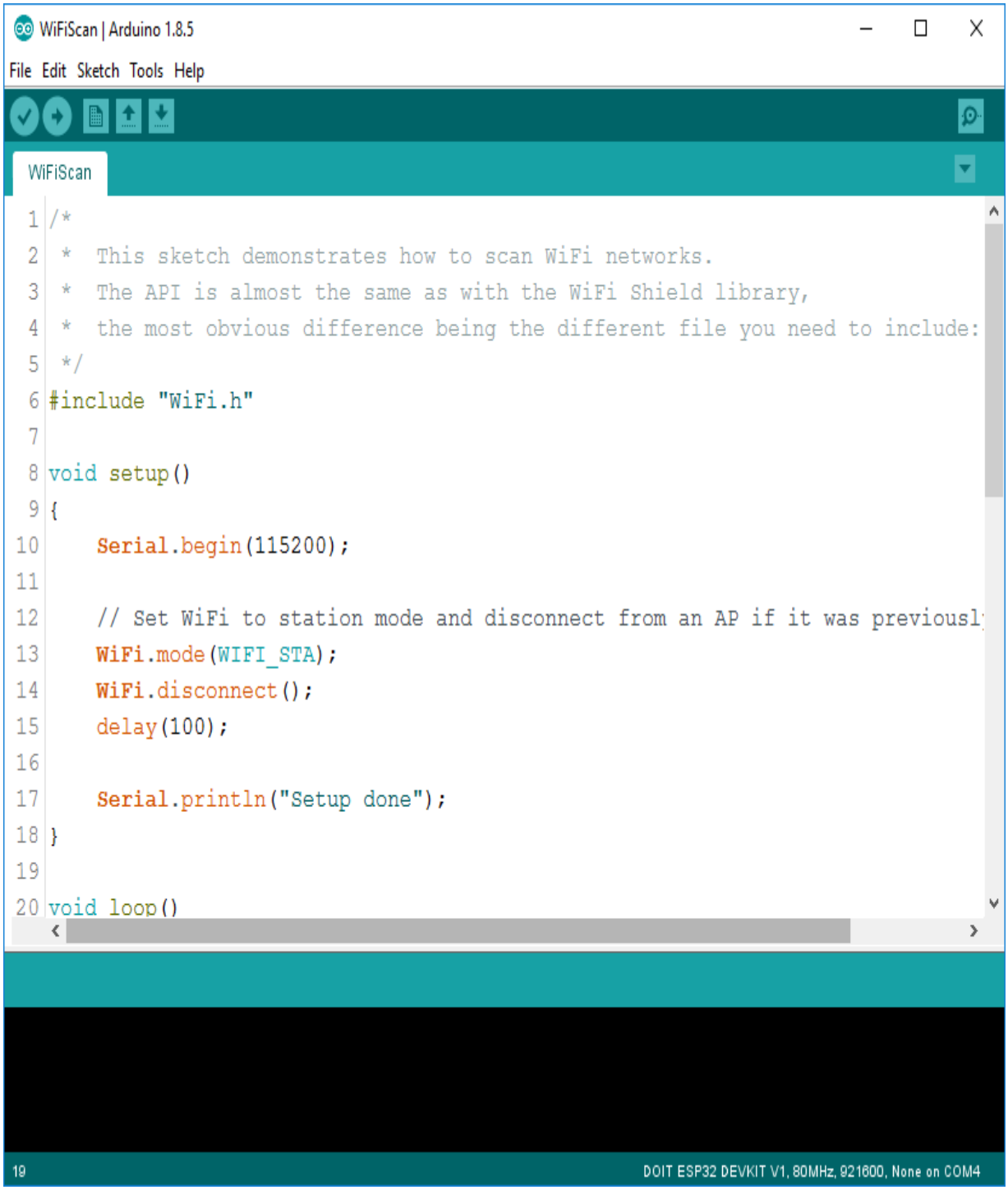
2. Select the Port (if you don’t see the COM Port in your Arduino IDE, you need to install the CP210x USB to UART Bridge VCP Drivers):



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

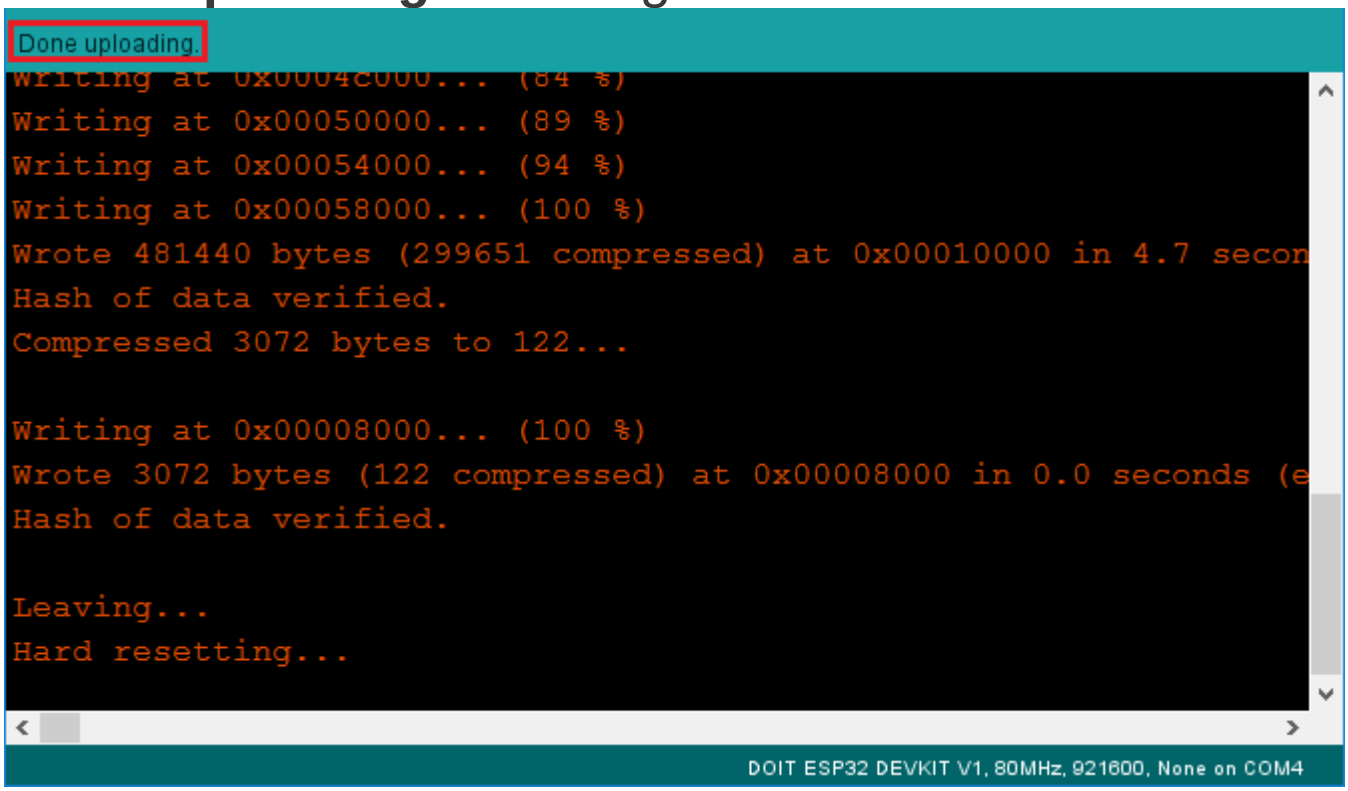


4. A new sketch opens in your Arduino IDE:


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

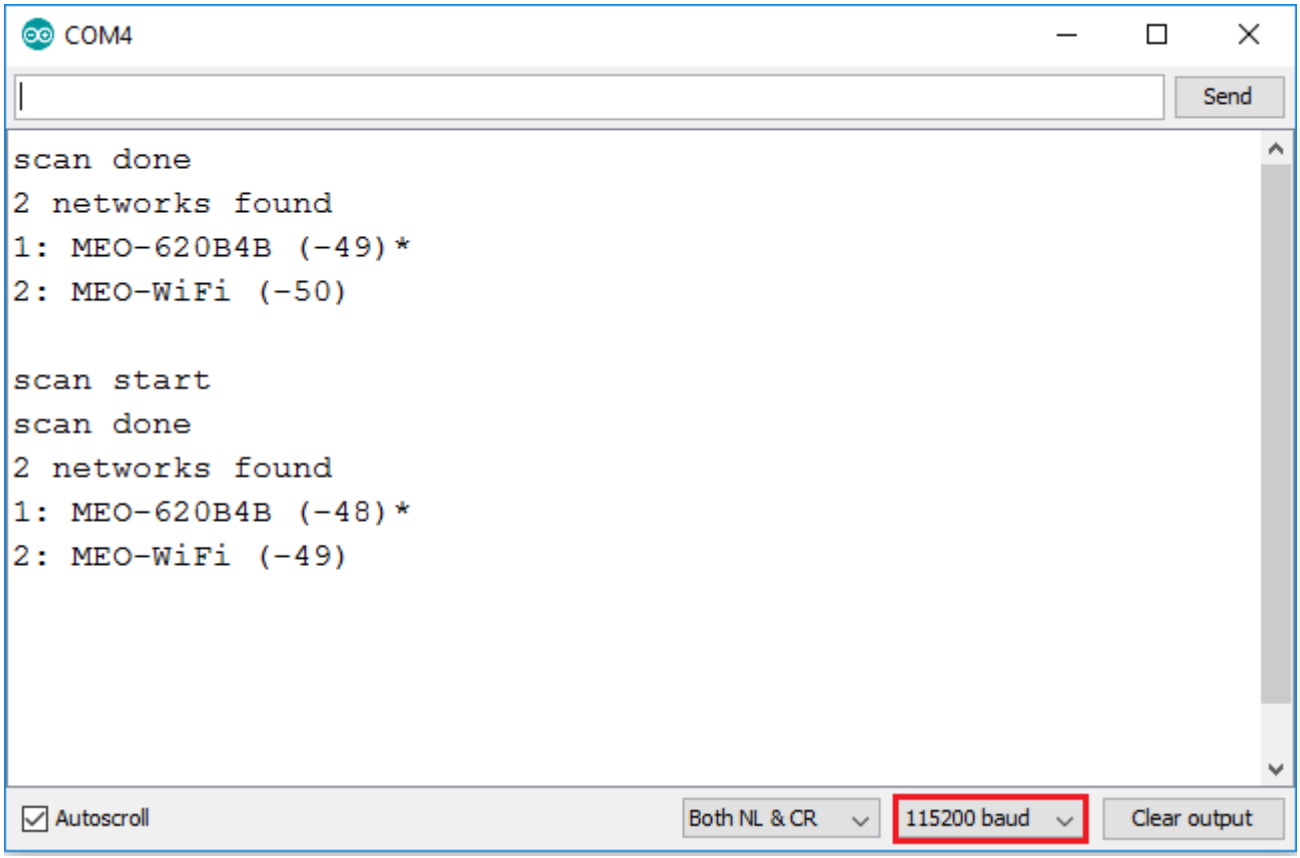
5. Press the **Upload** button in the Arduino IDE. Wait a few seconds while the code compiles and uploads to your board.

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



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:



Test :Blinking Program

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

Now Upload this Code.

```
#include <AFMotor.h>

#define echopinl  A0 // echo pin
#define trigpinl A1 // Trigger pin
#define echopinc  A2 // echo pin
#define trigpinc A3 // Trigger pin
#define echopinr  A4 // echo pin
#define trigpinr A5 // Trigger pin

long durationl, distancel, durationc, distancec, durationr, distancer ;

AF_DCMotor motorl(1);
AF_DCMotor motorr(2);
void setup()
{

    Serial.begin(9600);
```

```
  pinMode (trigpinl, OUTPUT);
  pinMode (echopinl, INPUT );
  pinMode (trigpinc, OUTPUT);
  pinMode (echopinc, INPUT );
  pinMode (trigpinr, OUTPUT);
  pinMode (echopinr, INPUT );



  //Set initial speed of the motor & stop
  motorl.setSpeed(250);
  //motorl.run(RELEASE);
  motorr.setSpeed(250);
 // motorr.run(RELEASE);
}
void loop()
{

    digitalWrite(trigpinl,LOW);
    delayMicroseconds(2);

    digitalWrite(trigpinl,HIGH);
    delayMicroseconds(10);
```

```
durationl=pulseIn (echopinl,HIGH);

distancel= durationl/58.2;
delay (50);
Serial.println(distancel);




digitalWrite(trigpinc,LOW);
delayMicroseconds(2);

digitalWrite(trigpinc,HIGH);
delayMicroseconds(10);

durationc=pulseIn (echopinc,HIGH);

distancec= durationc/58.2;
delay (50);
Serial.println(distancec);
```

```
   digitalWrite(trigpinr,LOW);
   delayMicroseconds(2);

   digitalWrite(trigpinr,HIGH);
   delayMicroseconds(10);

   durationr=pulseIn (echopinr,HIGH);

   distancer= durationr/58.2;
   delay (50);
   Serial.println(distancer);



if (distancec >15)
{
if (distancel > distancer)
{
  leftside();
}
if (distancer > distancel)
```

```
{
  rightside();
}
}

if (distancec <15)
{
if (distancel > distancer)
{
  left();
}
if (distancer > distancel)
{
  right();
}

}


  /*
  //uint8_t i;
  // Turn on motor
  motorl.setSpeed(150);
```

```
  motorr.setSpeed(150);
  motorl.run(FORWARD);
  motorr.run(FORWARD);
  delay(1000);
  motorl.setSpeed(150);
  motorr.setSpeed(150);
  motorl.run(BACKWARD);
  motorr.run(BACKWARD);
  // Accelerate from zero to maximum speed

  delay(1000);
*/
      }

void forward()

{
  motorl.setSpeed(250);
  motorr.setSpeed(250);
  motorl.run(FORWARD);
  motorr.run(FORWARD);
  }
```

```
void left()

{
  motorl.setSpeed(150);
  motorr.setSpeed(150);
  motorl.run(BACKWARD);
  motorr.run(FORWARD);
  }


void right()

{
  motorl.setSpeed(150);
  motorr.setSpeed(150);
  motorl.run(FORWARD);
  motorr.run(BACKWARD);
  }


void leftside()
```

```
{
  motorl.setSpeed(150);
  motorr.setSpeed(250);
  motorl.run(FORWARD);
  motorr.run(FORWARD);
  }


void rightside()

{
  motorl.setSpeed(250);
  motorr.setSpeed(150);
  motorl.run(FORWARD);
  motorr.run(FORWARD);
  }
```

How to use this robot.

This robot is fully automatic. So for running or using this robot just siply connect the battery and give the 6-12v power to L293d motor driver shield.
And keep the robot in the MAZE Solver arena or near any wall.
It run itself.