

Lab1-ESP32: Sending a message to ESP32- OLED display

Objectives:

To program a microcontroller to display a message on OLED display.

Equipment and materials:

S/N	Item	QTY
1.	PC with Arduino software	1
2.	ESP32	1
3.	USB to USB micro cable	1
4.	0.96 ' Arduino OLED display 256 x 64	1
5.	Wire, single core Red	1 feet
6.	Wire, single core Black	1 feet
7.	Wire, single core Blue(can be any colour)	1 feet

Part A: Connection requirements:

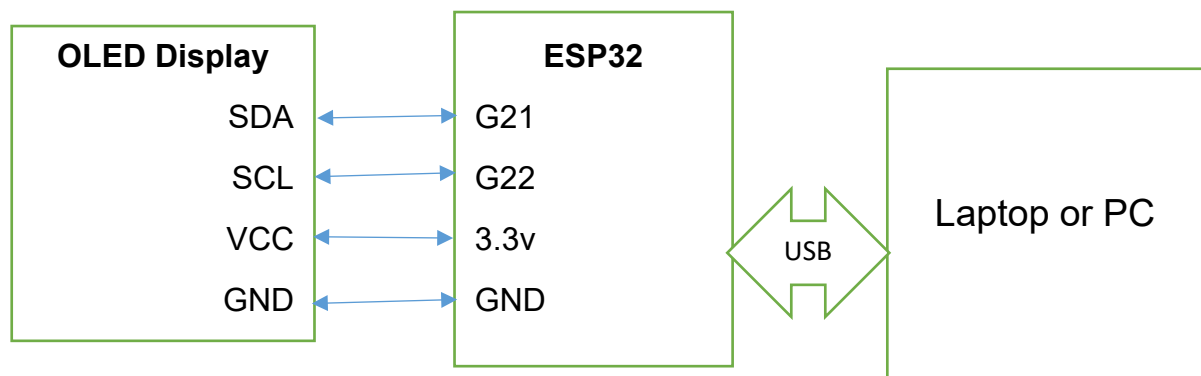


Figure 1: Connections of OLED display on microcontrollers and to PC.

1. The OLED module is connected to the ESP32 microcontroller using the following connections:

ESP32 OLED Display	ESP32	PC
VCC	3.3v	
GND	GND	
SDA	G21	
SCL	G22	
	USB	USB micro

Part B: Installing ESP-32 to Arduino IDE

2. In the Arduino IDE, go to File> Preference

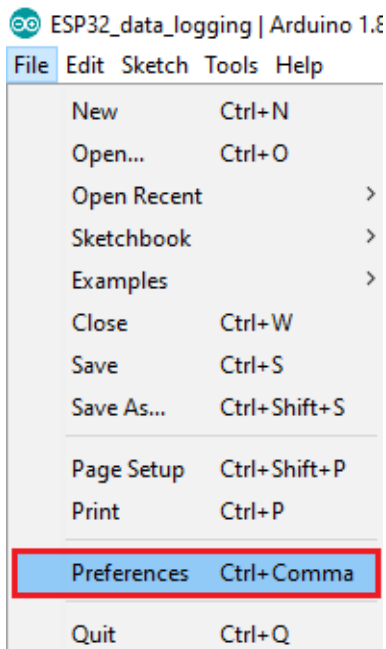


Figure 1- Preferences

3. For the 'Additional Board Manager URLs' enter https://dl.espressif.com/dl/package_esp32_index.json then, click the 'OK' button

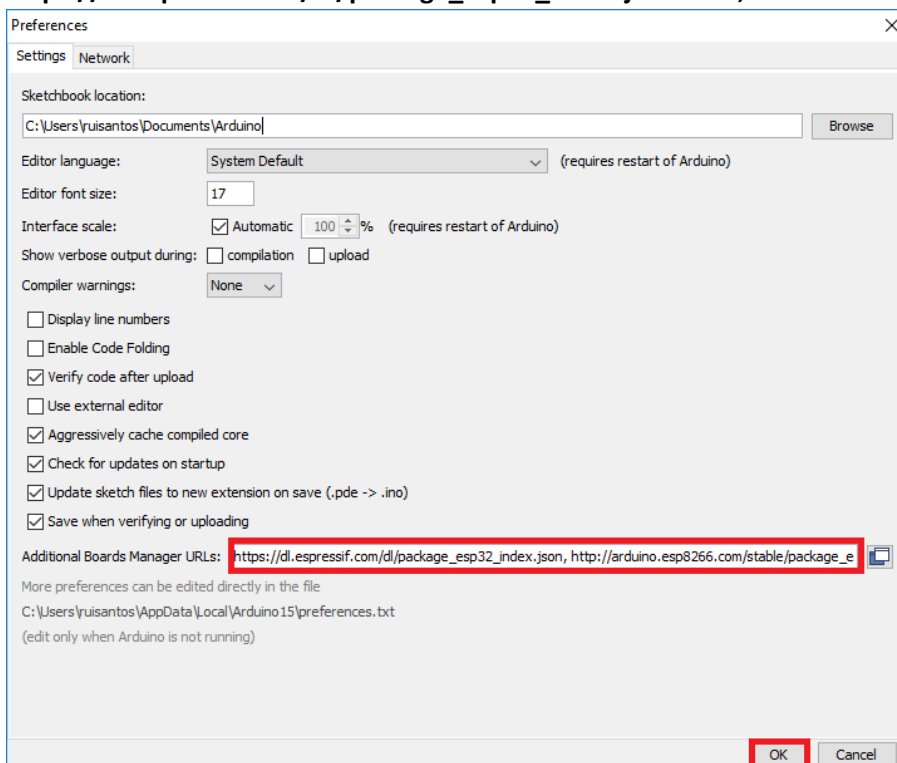


Figure 2: Additional Board Manager URL

4. Open the Boards Manager with Tools > Board > Boards Manager...

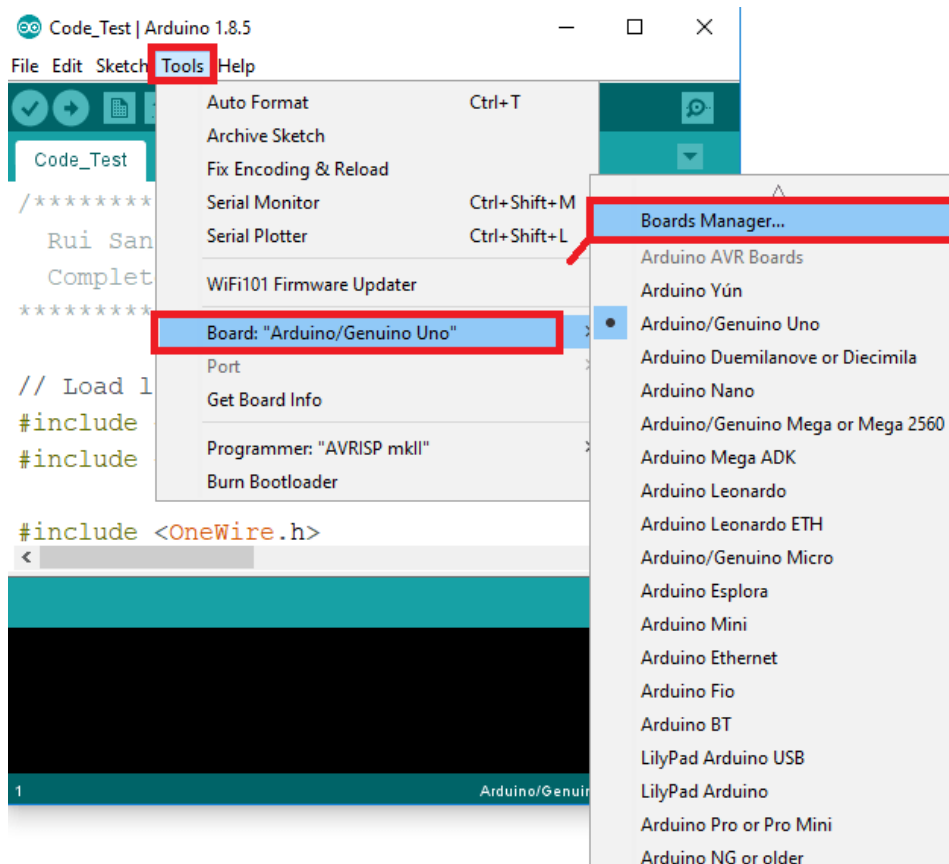


Figure 3: Board Manager

5. Search for ESP32 and press install button for the 'ESP32 by Espressif Systems'.

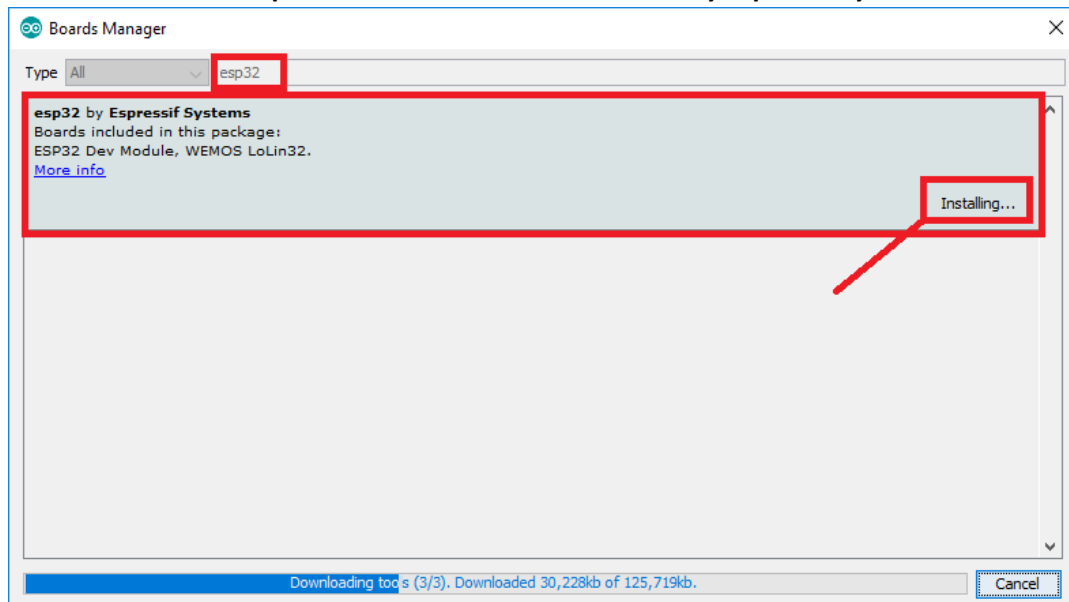


Figure 4: Installing ESP32 Board Manager

6. It should be installed after a few seconds.

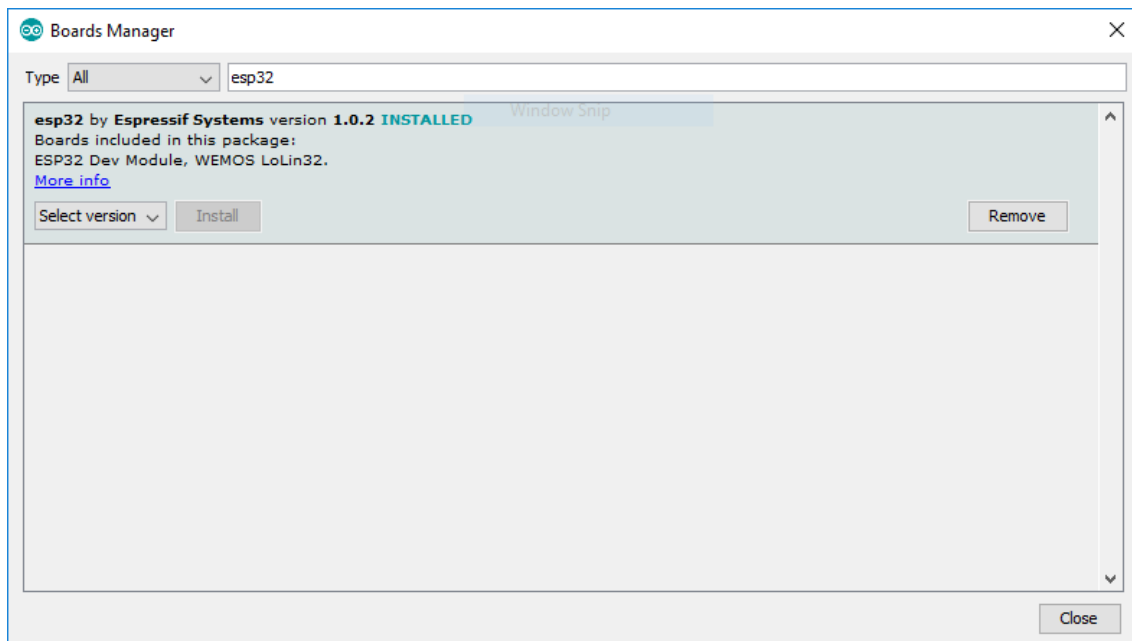


Figure 5: ESP32 Board Manager is installed

7. Click Tools > Board to select 'ESP32 Dev Module'.

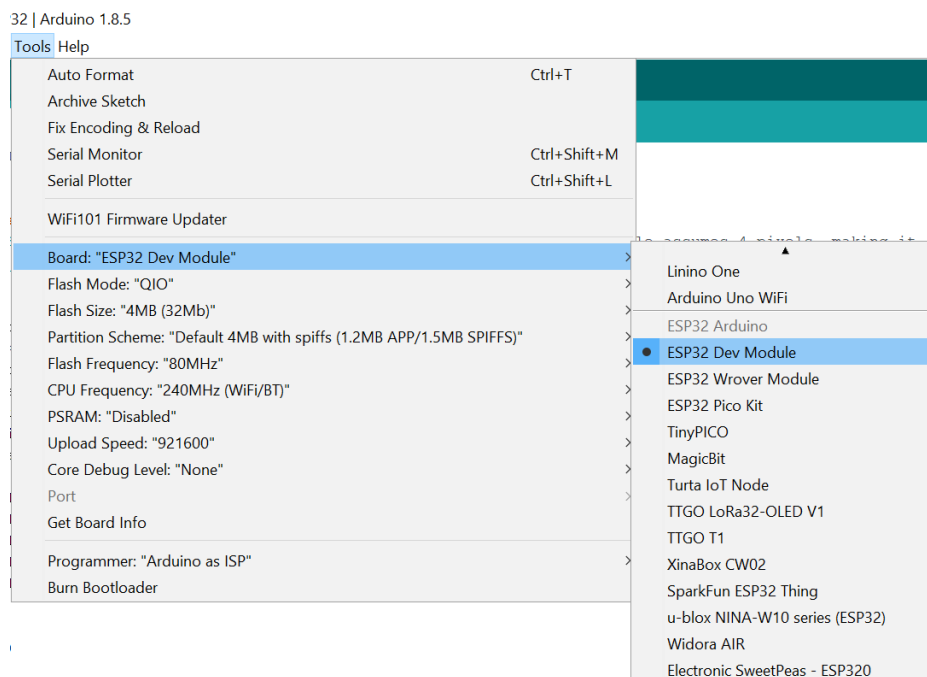


Figure 6: ESP32 Dev Module Selection

8. Connect an ESP32 module to PC

9. Click Tools > Port > to select available hardware port

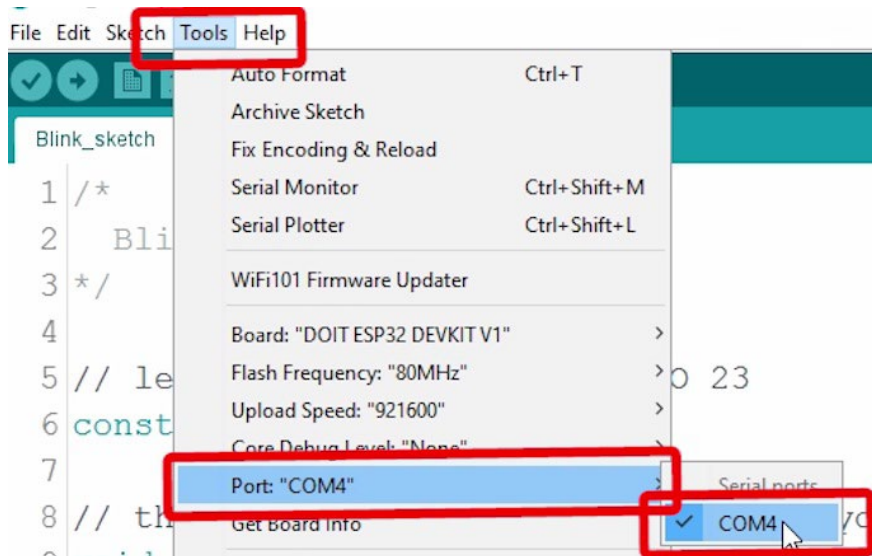
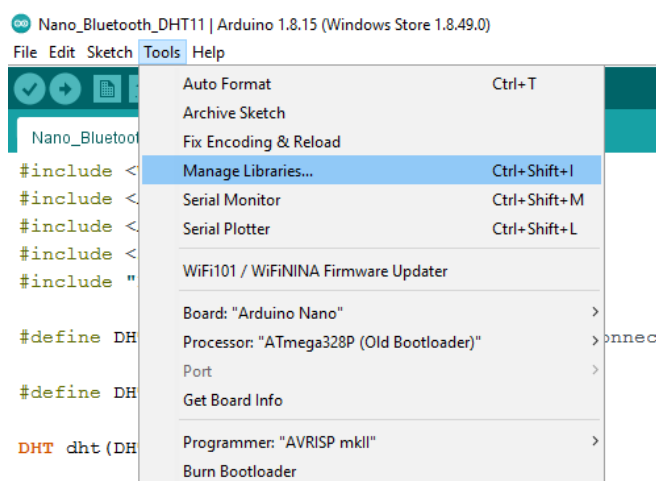


Figure 7: Port Selection

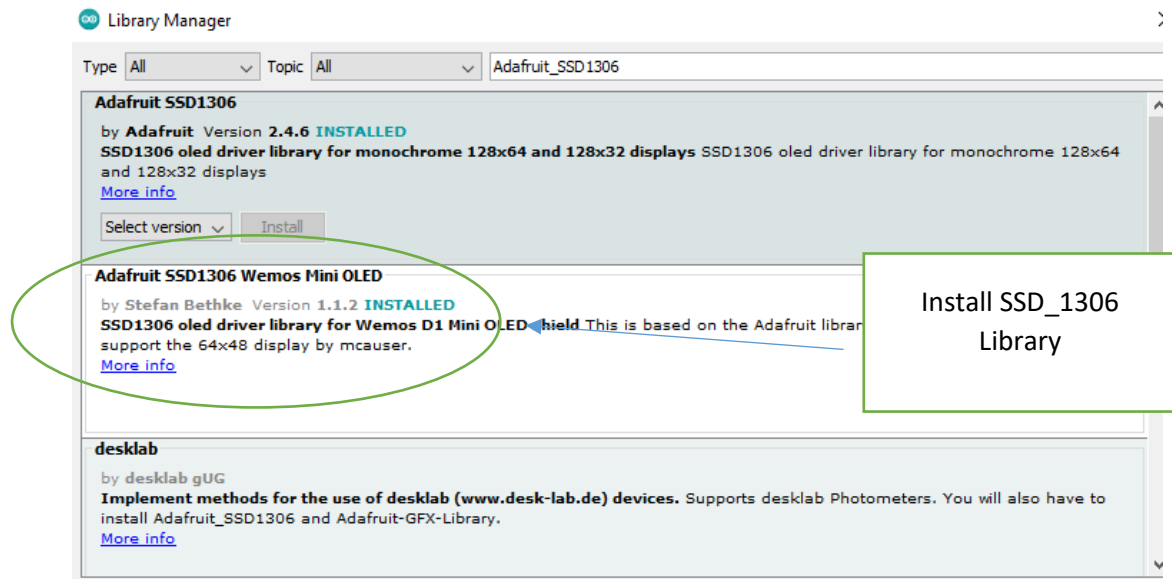
10. If COM Port selection is not available in list, proceed to install **CP210x USB to UART Bridge VCP Drivers**

Part C: Installing Library for OLED Display Module

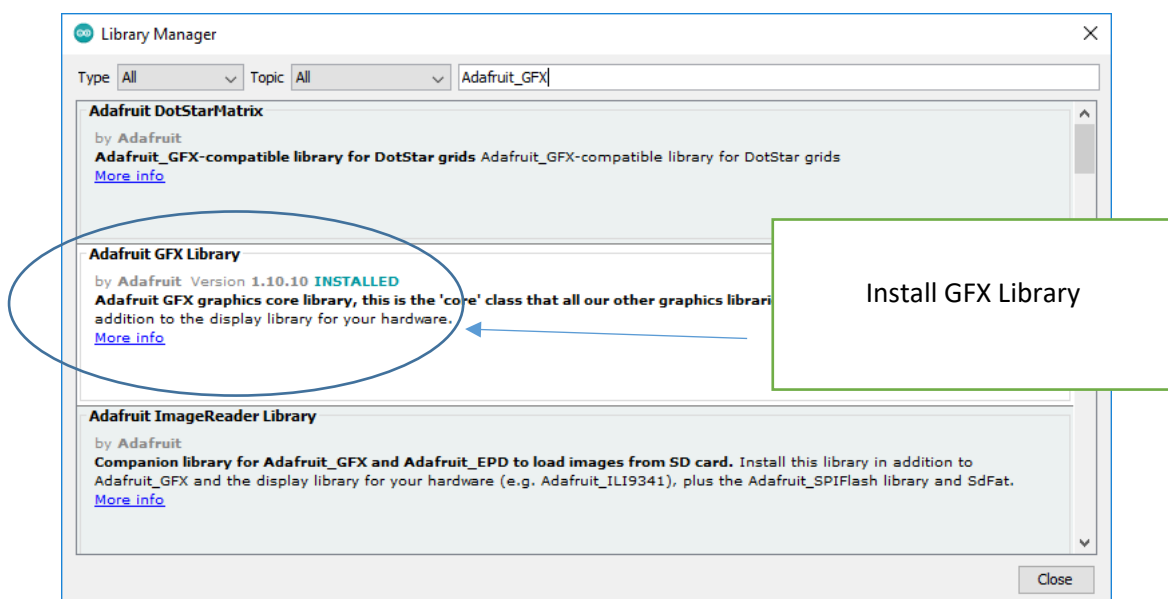
11. The SSD1306 controller of the OLED display has flexible yet complex drivers. Vast knowledge on memory addressing is required in order to use the SSD1306 controller. Fortunately, Adafruit's SSD1306 library was written to hide away the complexities of the SSD1306 controller so that we can issue simple commands to control the display.
12. To install the library navigate to the Tools > Manage Library > Manage Libraries... Wait for Library Manager to download libraries index and update list of installed libraries



13. Filter your search by typing 'adafruit ssd1306'. There should be a couple entries. Look for Adafruit SSD1306 by Adafruit. Click on that entry, and then select Install.

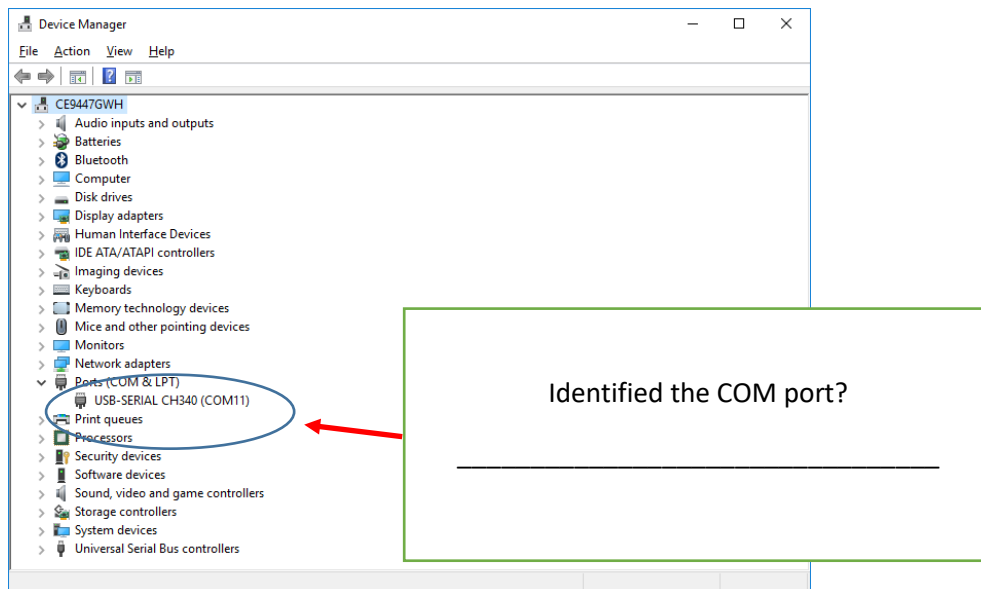


14. This Adafruit SSD1306 library is a hardware-specific library which handles lower-level functions. It needs to be paired with Adafruit GFX Library to display graphics primitives like points, lines, circles, rectangles etc. Install this library as well.



Part D: Programming the Arduino Nano

15. Connect the ESP32 to the PC using the USB to USB micro cable. Make sure the OLED display and ESP32 are inserted to the Bread Board. With the wire properly connected.
16. Check which com port being assign to the ESP32. Go to device manager to identify the COM port.



17. Open up the given ESP32 program OLED-Hello_World_Lab1.ino



18. click on the arrow button to compile and download the program to the microcontroller:
The "Hello World " will display on the OLED display.

Annex 1

Program: OLED-Hello_world_lab1.ino

```
#include <Adafruit_GFX.h>
```

```
#include <Adafruit_SSD1306.h>
```

```
#define SCREEN_WIDTH 128 // OLED display width, in pixels
```

```
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
```

```
#define OLED_RESET 4 // Reset pin # (or -1 if sharing Arduino reset pin)
```

```
#define SCREEN_ADDRESS 0x3c ///< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32
```

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
```

```
  if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS))
```

```
  {
```

```
    Serial.println(F("SSD1306 allocation failed"));
```

```
    for(;;); // Don't proceed, loop forever
```

```
  }
```

```
}
```

```
void loop()
```

```
{
```

```
  display.clearDisplay();
```

```
  display.setTextSize(2); // Draw 2X-scale text
```

```
  display.setTextColor(SSD1306_WHITE);
```

```
  display.setCursor(10, 0);
```

```
  display.println(F("Hello \nWorld"));
```

```
  display.display();
```

```
  return;
```

```
}
```