

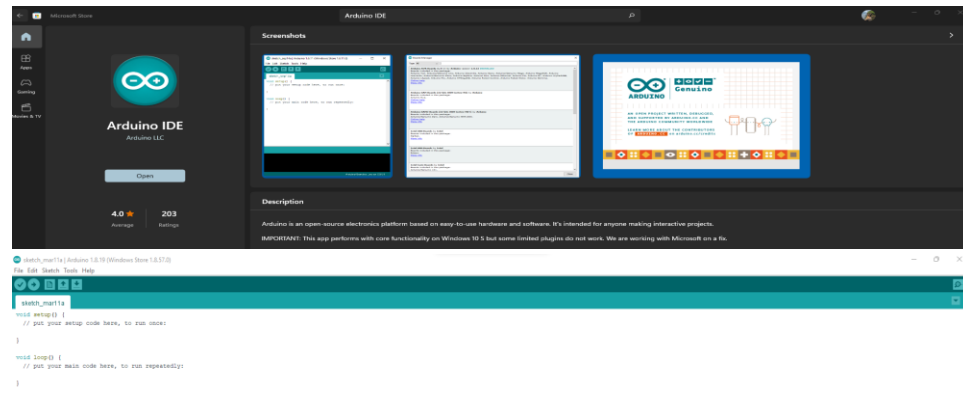
# JOBSHEET 1

Omar Al-Maktary  
TI-3H 1941720237

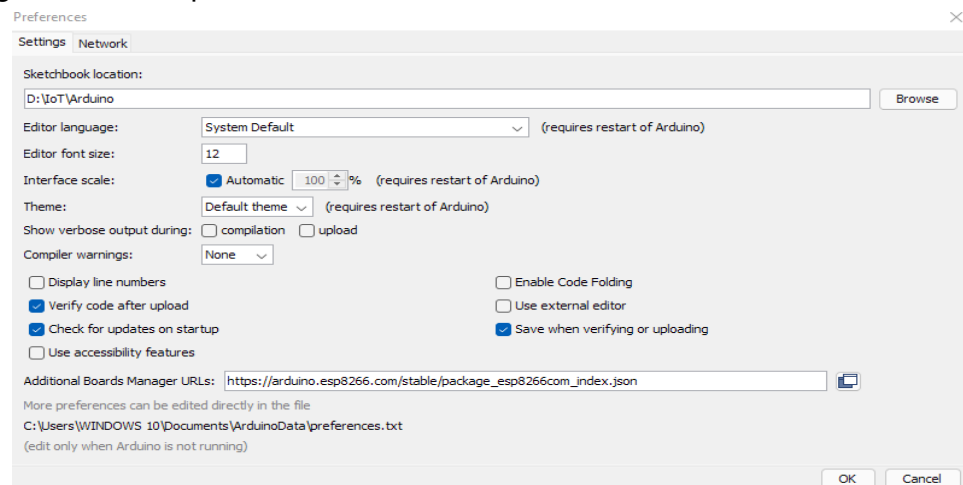
## Practical Steps:

### 1. Download and install Arduino IDE:

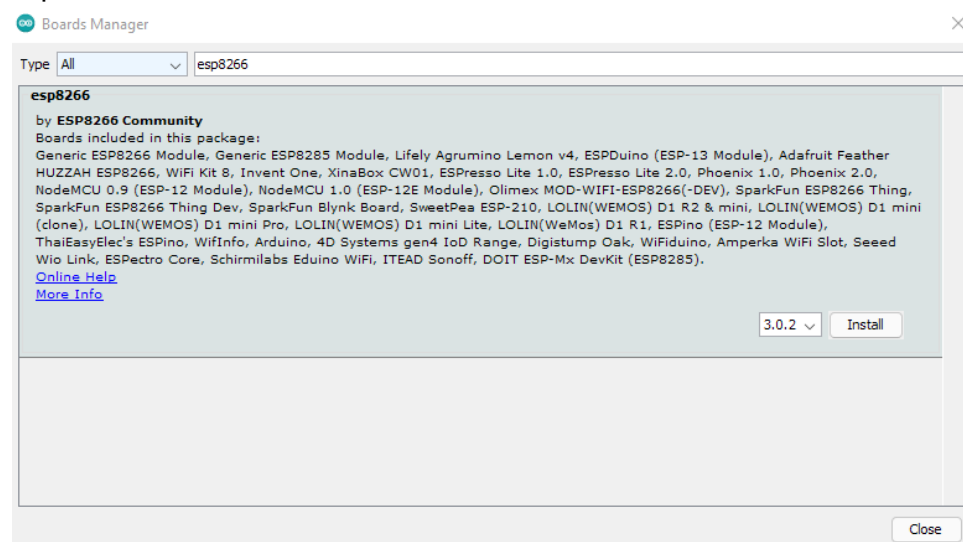
We can download Arduino from Microsoft Store:



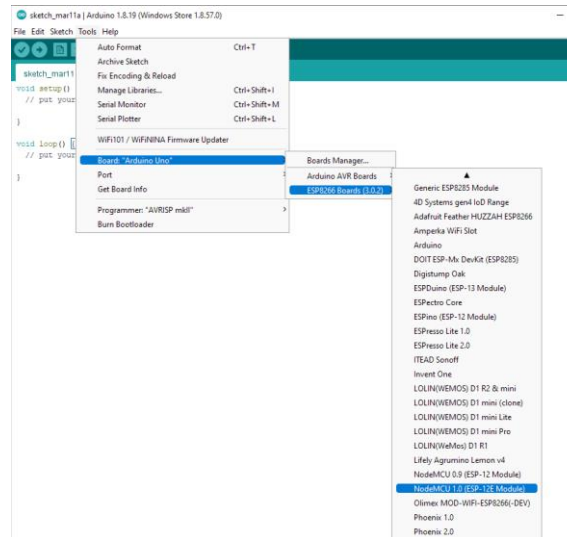
### 2. Configure Arduino preference:



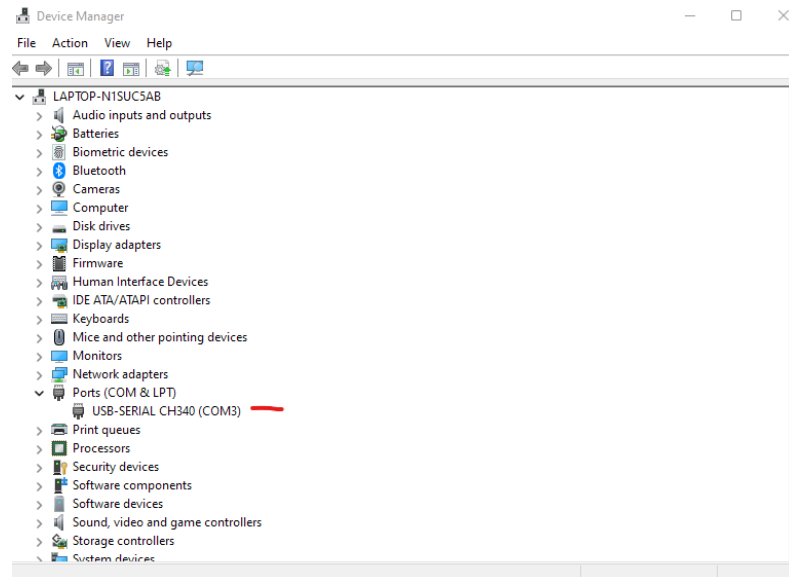
### 3. Install Esp8266:



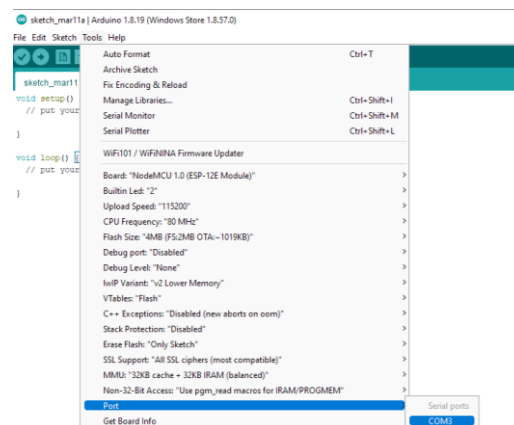
#### 4. Choose the board NodeMCU 1.0 (ESP-12E Module):



#### 5. Check the board connection:



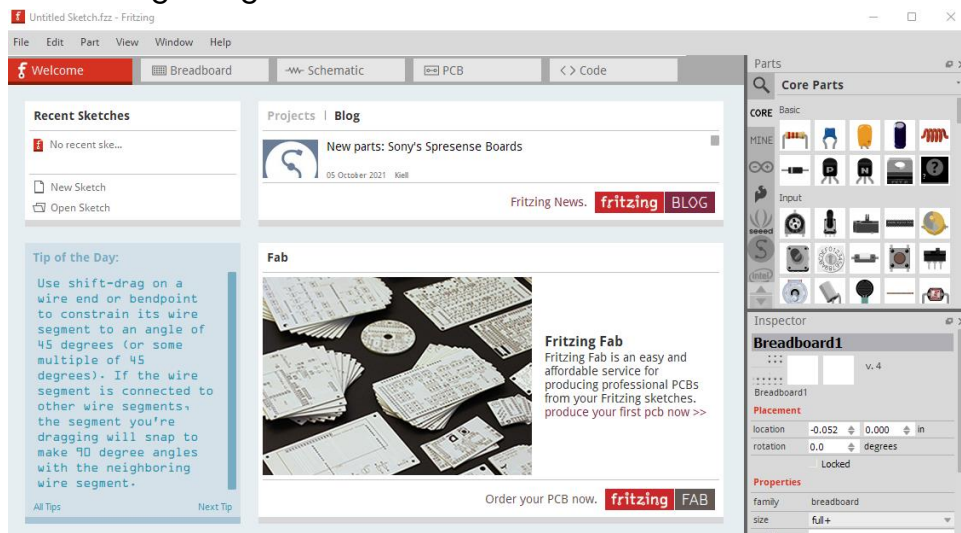
#### 6. Select COM3:



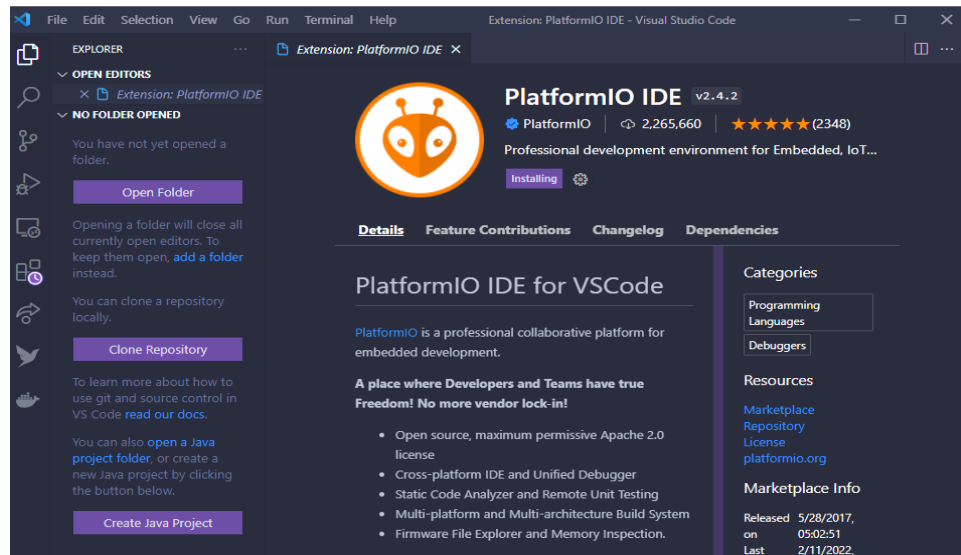
## 7. Compiling to the ESP8266 board:



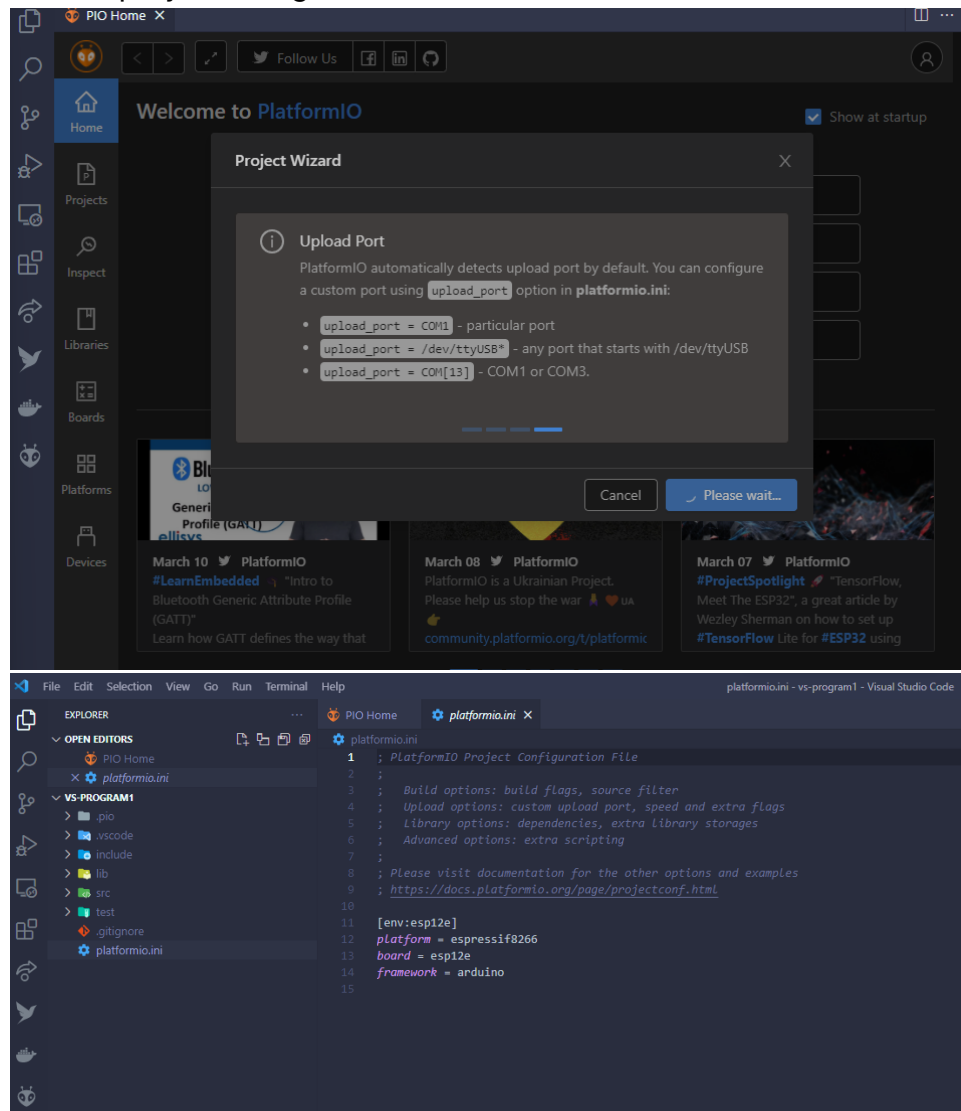
## 8. Download Fritzing Designer:



## 9. Download PlatformIO IDE extension to Visual Studio:



## 10. Create a new project using PlatformIO IDE:



## 11. Configure the "Hello World" code:

```
src > C++ main.cpp > loop()
1 #include <Arduino.h>
2
3 void setup()
4 {
5     // put your setup code here, to run once:
6     Serial.begin(115200); // setbaudrate 115200
7 }
8
9 void loop()
10 {
11     // put your main code here, to run repeatedly:
12     Serial.println("Hello world"); // show the following string to the serial monitor
13     delay(1000); // delay 1000 ms
14 }
```

## 12. Build, upload, and show the serial monitor the project

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Compiling .pio\build\esp12e\framework-arduinoespressif\src\main.cpp.o
Compiling .pio\build\esp12e\framework-arduinoespressif\src\main.cpp.o
Archiving .pio\build\esp12e\framework-arduinoespressif\src\main.cpp.o
Linking .pio\build\esp12e\firmware.elf
Retrieving maximum program size .pio\build\esp12e\firmware.elf
Checking size .pio\build\esp12e\firmware.elf
Advanced memory usage is available via "PlatformIO Home > Project Inspector"
RAM: [---] 34.2% (used 28800 bytes from 83520 bytes)
Flash: [---] 25.4% (used 26144 bytes from 104448 bytes)
Building .pio\build\esp12e\firmware.bin
Creating BIN file ".pio\build\esp12e\firmware.bin" using "C:\Users\WIDOKAS\AppData\Local\PlatformIO\packages\framework-arduinoespressif\tools\bootloaders\bootloader.py" and ".pio\build\esp12e\firmware.elf"
===== [SUCCESS] Took 10.14 seconds =====

Terminal will be reused by tasks, press any key to close it.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
writing at 0x00010000... (61 %
writing at 0x00020000... (69 %
writing at 0x00030000... (76 %
writing at 0x00040000... (84 %
writing at 0x00050000... (92 %
writing at 0x00060000... (100 %)
Wrote 262096 bytes (197700 compressed) at 0x00000000 in 17.5 seconds (effective 123.2 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
===== [SUCCESS] Took 20.79 seconds =====

Terminal will be reused by tasks, press any key to close it.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
> Executing task: C:\Users\WIDOKAS\AppData\Local\PlatformIO\Scripts\platformio.exe device monitor --environment esp12e <
--- Available filters and text transformations: colorize, debug, default, direct, esp206_exception_decoder, hexlify, log2file, nocontrol, printable, send_on_enter, time
--- More details at https://bit.ly/pio-monitor-filters
--- Minimize on CPU: 9600,8,4,1 ---
--- QUIT: Ctrl+C | MENU: Ctrl+M | HELP: Ctrl+H followed by Ctrl+H ---
1000000000
```

## 13. Make the LED run:

```
src > C++ main.cpp > setup()
1 #include <Arduino.h>
2
3 void setup()
4 {
5     // put your setup code here, to run once:
6
7     //practicum 1
8     // Serial.begin(115200); // setbaudrate 115200
9
10    //practicum 2
11    pinMode(D4, OUTPUT);
12 }
13
14
15 void loop()
16 {
17     // put your main code here, to run repeatedly:
18
19     //practicum 1
20     // Serial.println("Hello World"); // show the following string to the serial monitor
21     // delay(1000); // delay 1000 ms
22
23     //practicum 2
24     digitalWrite(D4, HIGH);
25     delay(1000);
26     digitalWrite(D4, LOW);
27     delay(1000);
28 }
29
```

Results:

