

Task Week 3

Installation & Configuration



Oleh:

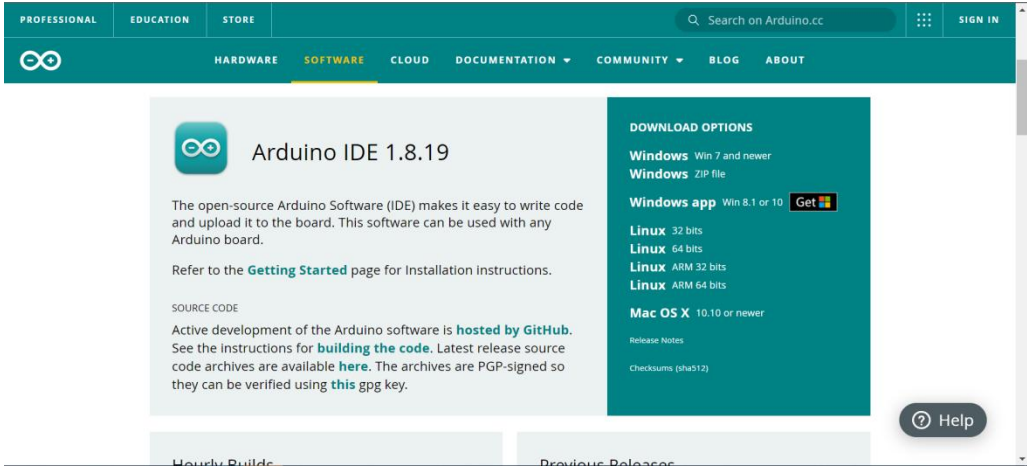
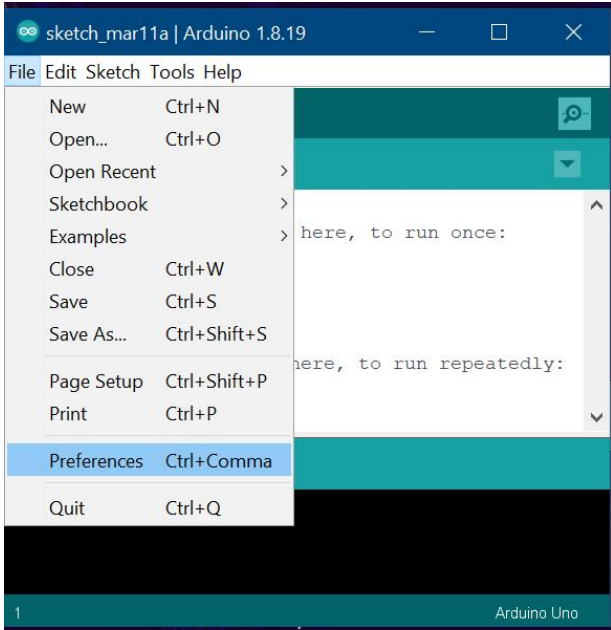
Widiareta Safitri

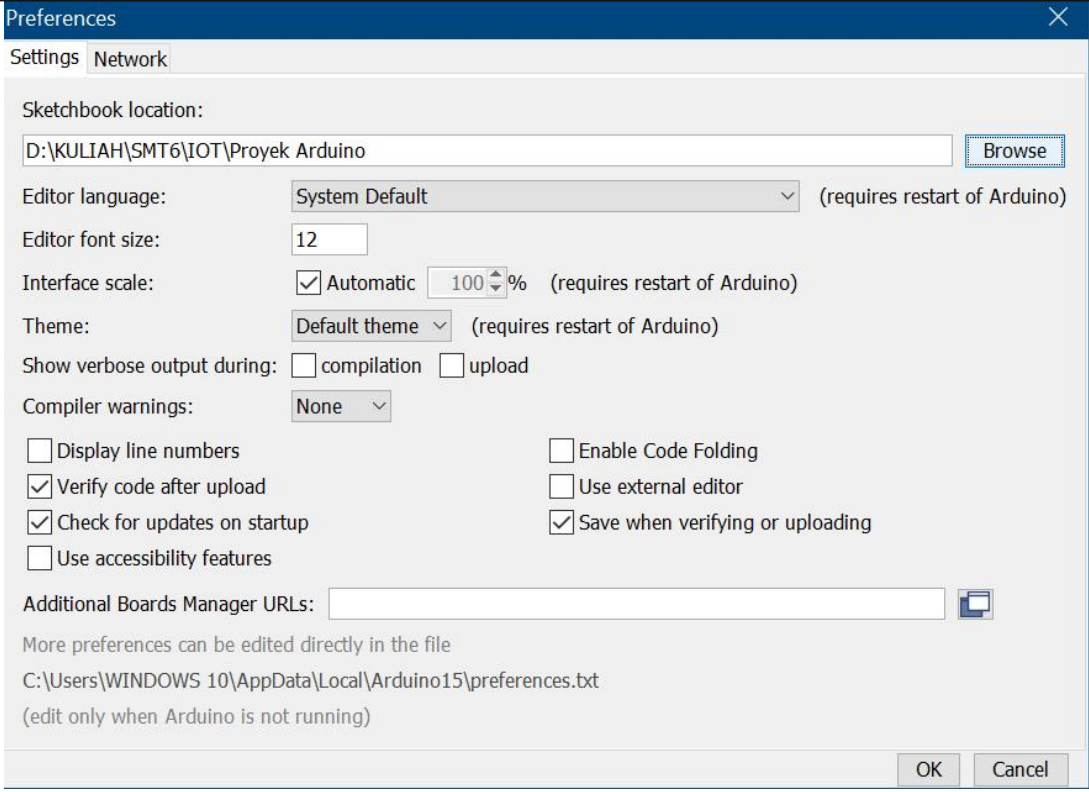
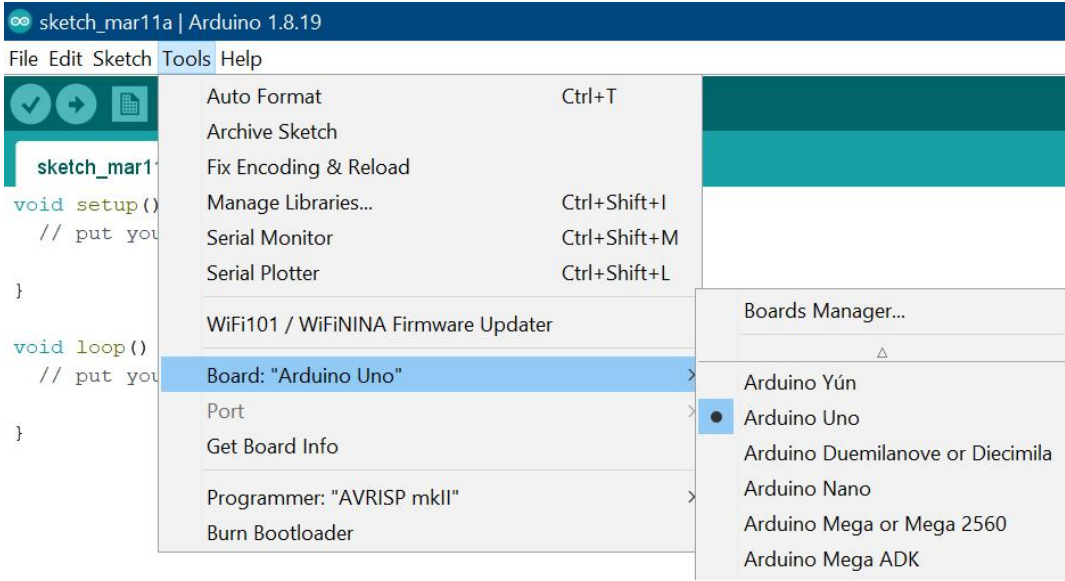
TI-3H / 1941720081 / 26

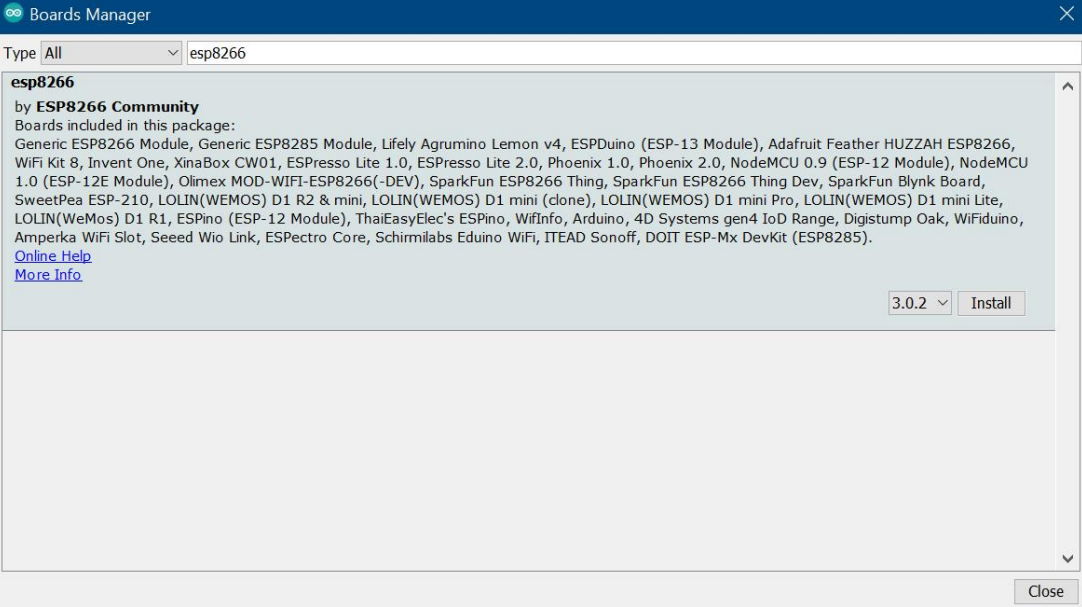
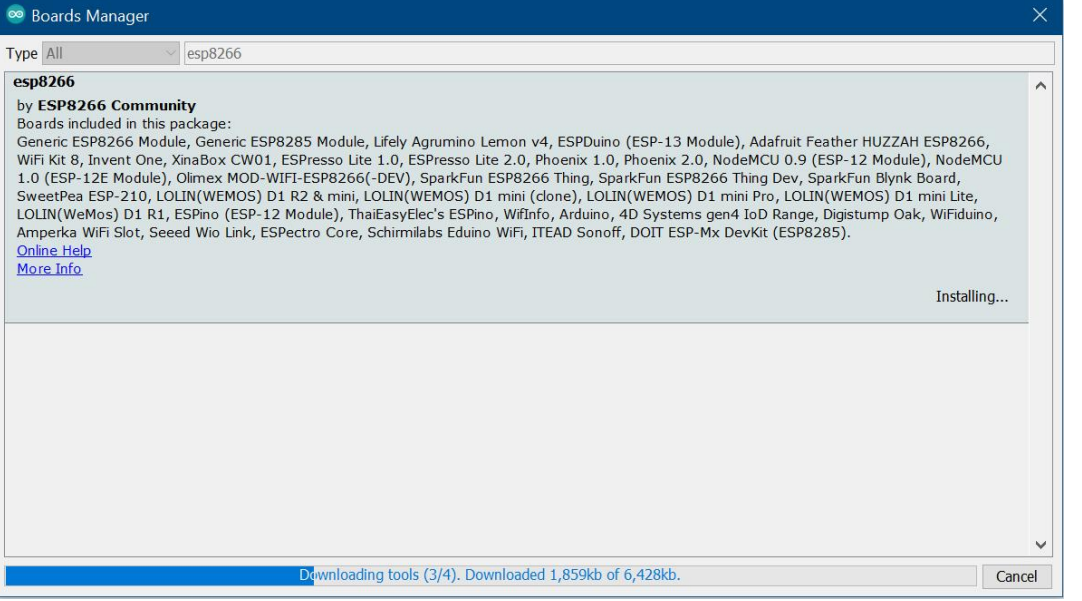
**PROGRAM STUDI D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG**

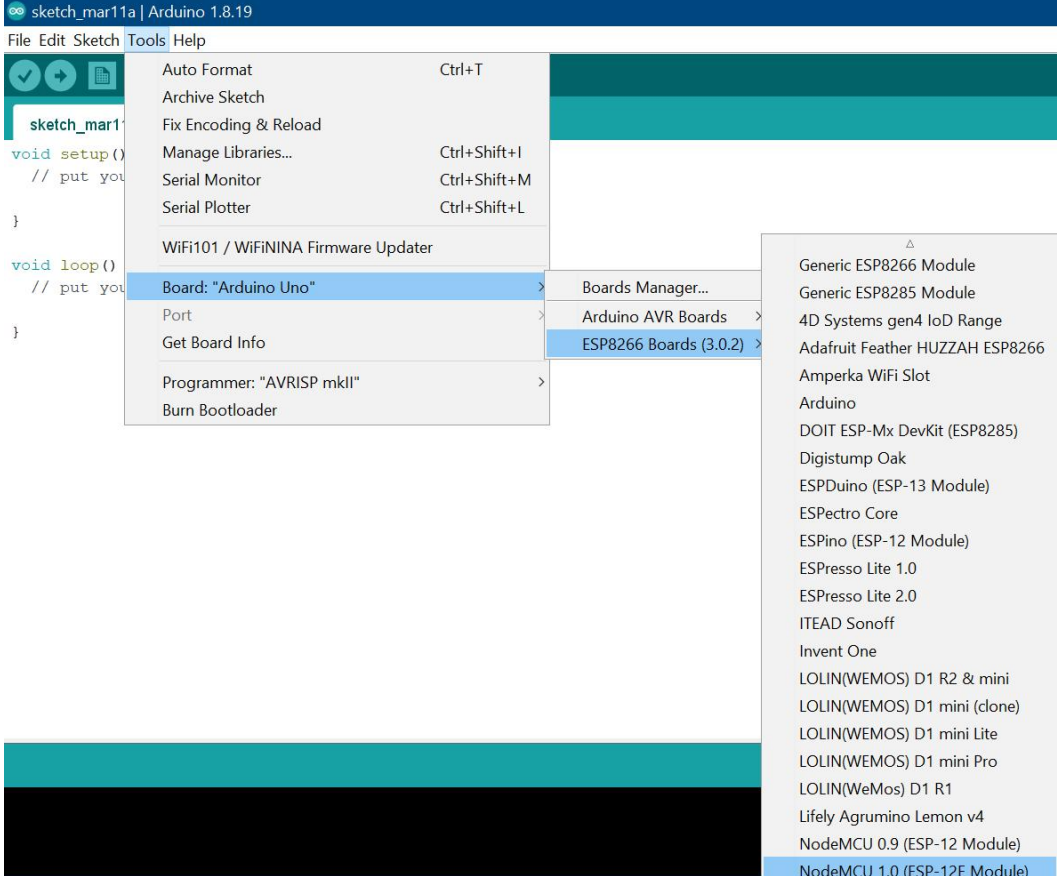
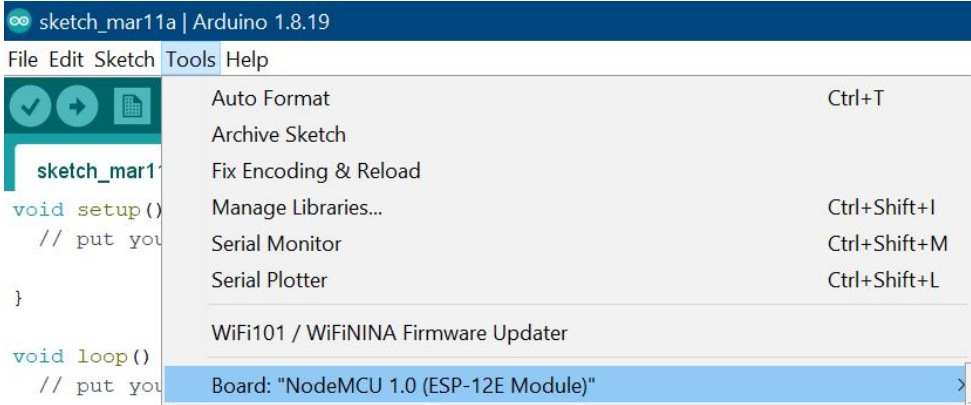
2022

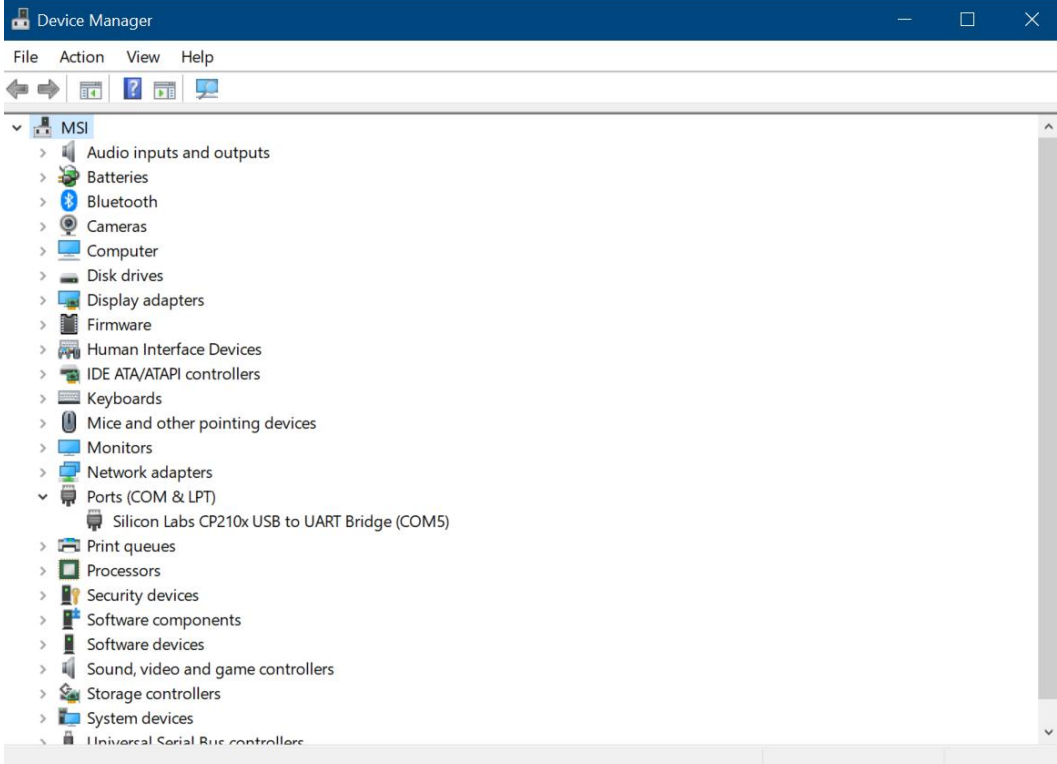
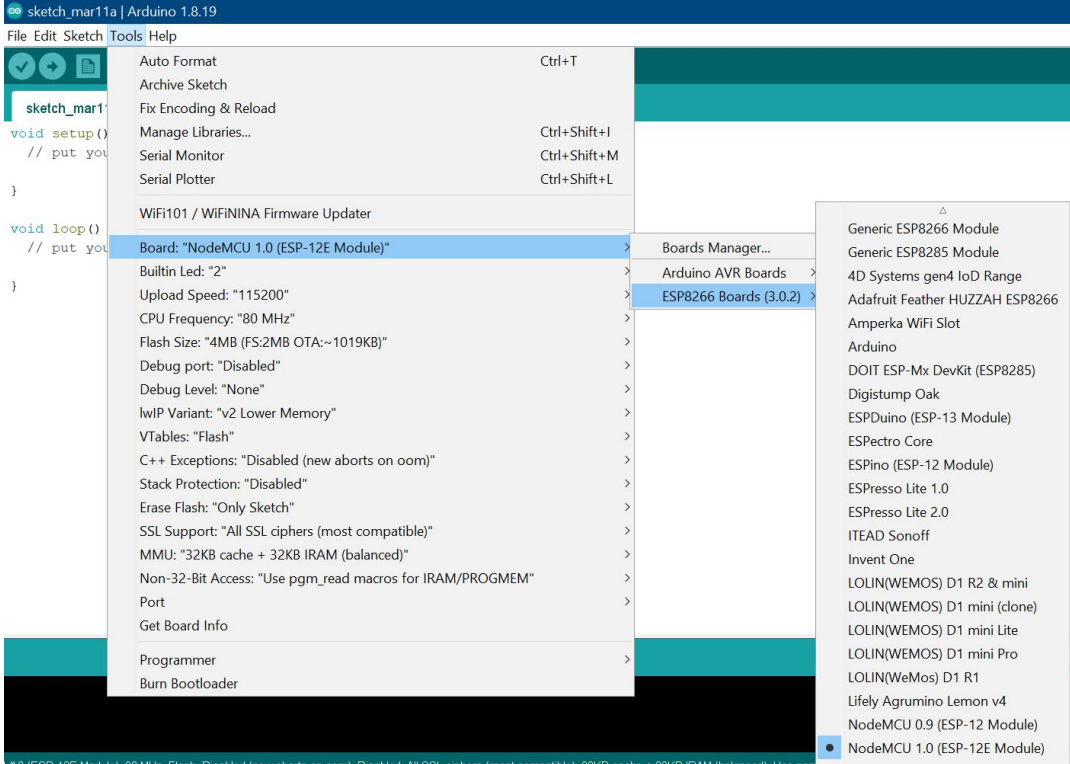
Arduino IDE Installation

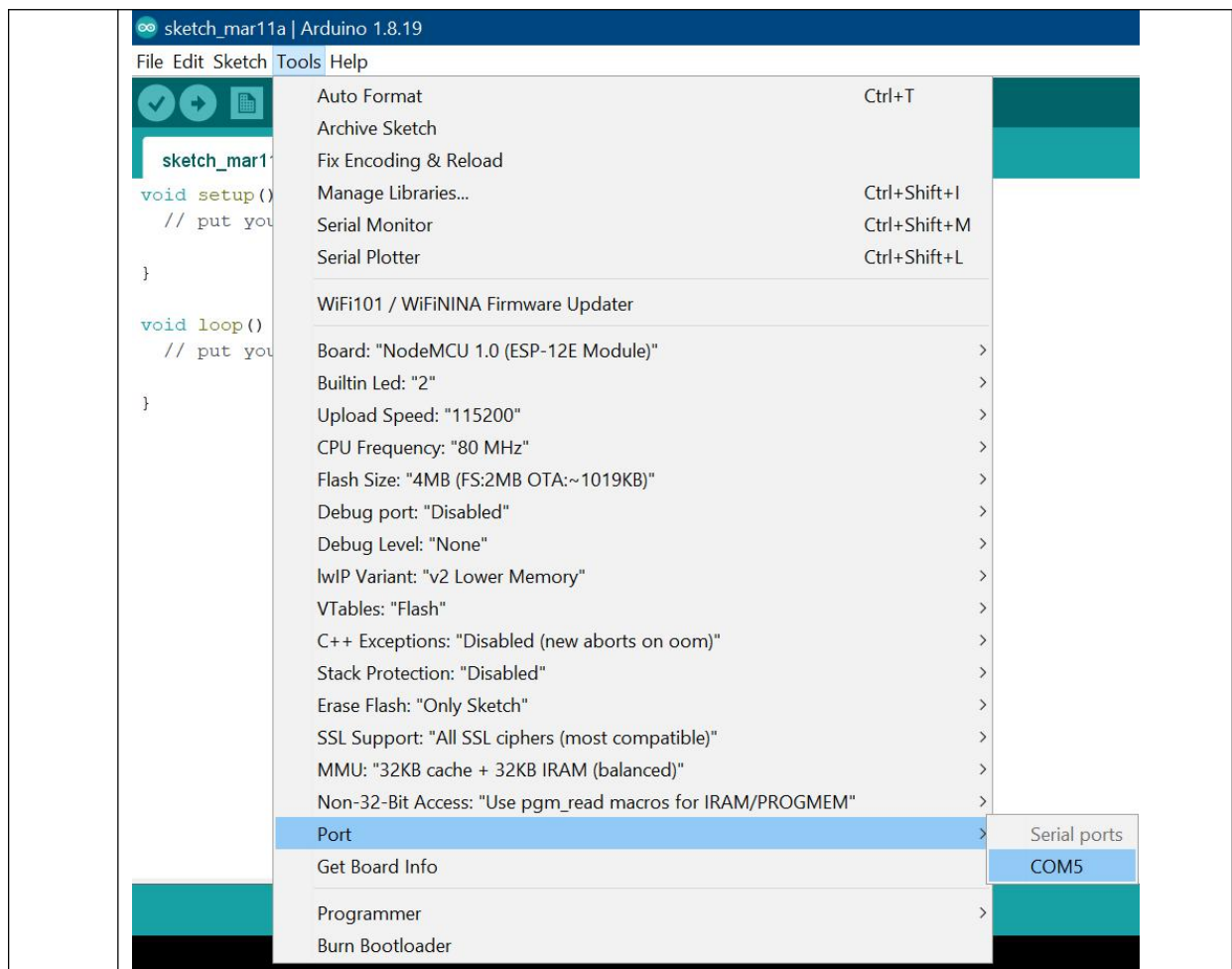
No	Step
1	<p>Install Arduino IDE, download this link https://www.arduino.cc/download_handler.php?f=/arduino-1.8.12-windows.exe.</p> 
2	<p>Open arduino application IDE</p> <p>Open menu File > Preferences</p> 
3	<p>At column "Additional Boards Manager URLs", write copy paste in this address berikut https://arduino.esp8266.com/stable/package_esp8266com_index.json. Next click OK</p>

	
4	<p>Go to Tools > Board > Board Manager menu. Type "esp8266" in the search field. The sentence "esp8266 by ESP8266 Community" will appear in the list below, select it and click the Install button on the bottom right.</p> 

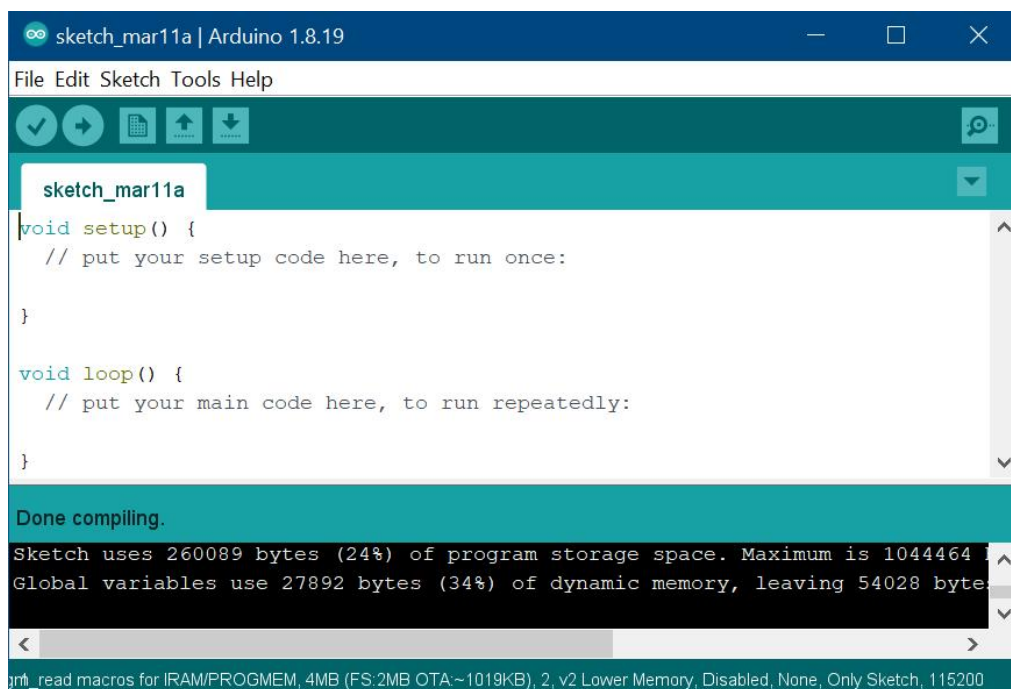
	 
5	Open menu and choose option Tools > Board Manager > NodeMCU 1.0 (ESP-12E Module)

	 <p>sketch_mar11a Arduino 1.8.19</p> <p>File Edit Sketch Tools Help</p> <p>Tools menu options:</p> <ul style="list-style-type: none"> 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" (selected) Port Get Board Info Programmer: "AVRISP mkII" Burn Bootloader <p>Boards Manager options:</p> <ul style="list-style-type: none"> Generic ESP8266 Module Generic ESP8285 Module 4D Systems gen4 IoD Range Adafruit Feather HUZZAH ESP8266 Amperka WiFi Slot Arduino DOIT ESP-Mx DevKit (ESP8285) Digistump Oak ESPDuino (ESP-13 Module) ESPectro Core ESPino (ESP-12 Module) ESPresso Lite 1.0 ESPresso Lite 2.0 ITEAD Sonoff Invent One LOLIN(WEMOS) D1 R2 & mini LOLIN(WEMOS) D1 mini (clone) LOLIN(WEMOS) D1 mini Lite LOLIN(WEMOS) D1 mini Pro LOLIN(WeMos) D1 R1 Lifely Agrumino Lemon v4 NodeMCU 0.9 (ESP-12 Module) NodeMCU 1.0 (ESP-12E Module) (selected)
6	<p>When the installation process is complete and the board has been selected, it should display NodeMCU 1.0 (ESP-12E Module)</p>  <p>sketch_mar11a Arduino 1.8.19</p> <p>File Edit Sketch Tools Help</p> <p>Tools menu options:</p> <ul style="list-style-type: none"> 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: "NodeMCU 1.0 (ESP-12E Module)" (selected)
7	<p>Now insert the USB data cable from the NodeMCU to the Laptop, then first check in Control Panel > Device Manager. The image below shows that the interface between the NodeMCU and the Windows 10 Laptop is connected via a COM3 PORT (the COM address of each person is not always the same, depending on each computer).</p>

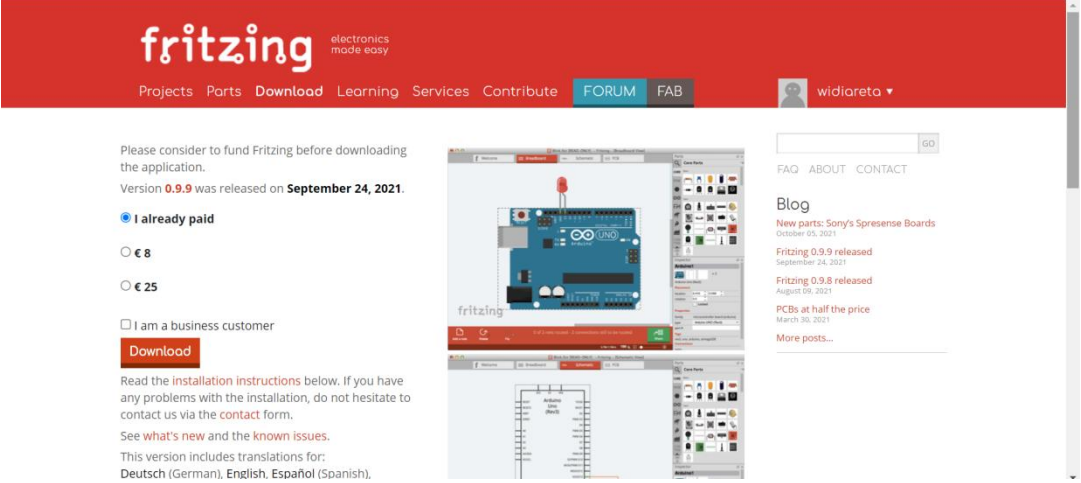
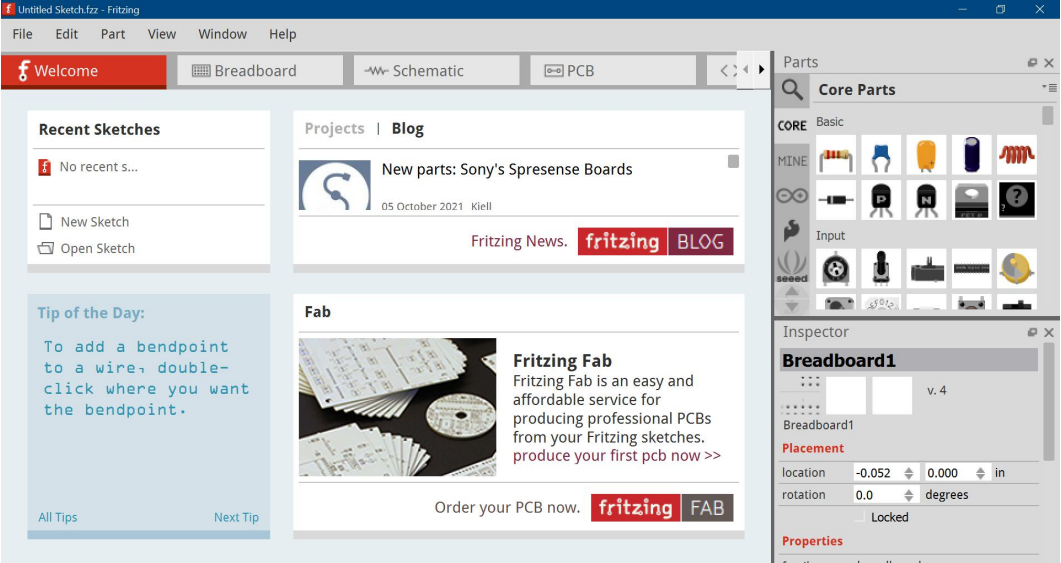
	
8	<p>Open the Arduino IDE application, select the Tools > Board menu, the default may be "Board: Arduino / Genuino Uno" to "NodeMCU 1.0 (ESP-12E Module)"</p> 
9	<p>Open the Arduino IDE application on the Tools > Port > COM3 menu (adjust to the respective port number)</p>

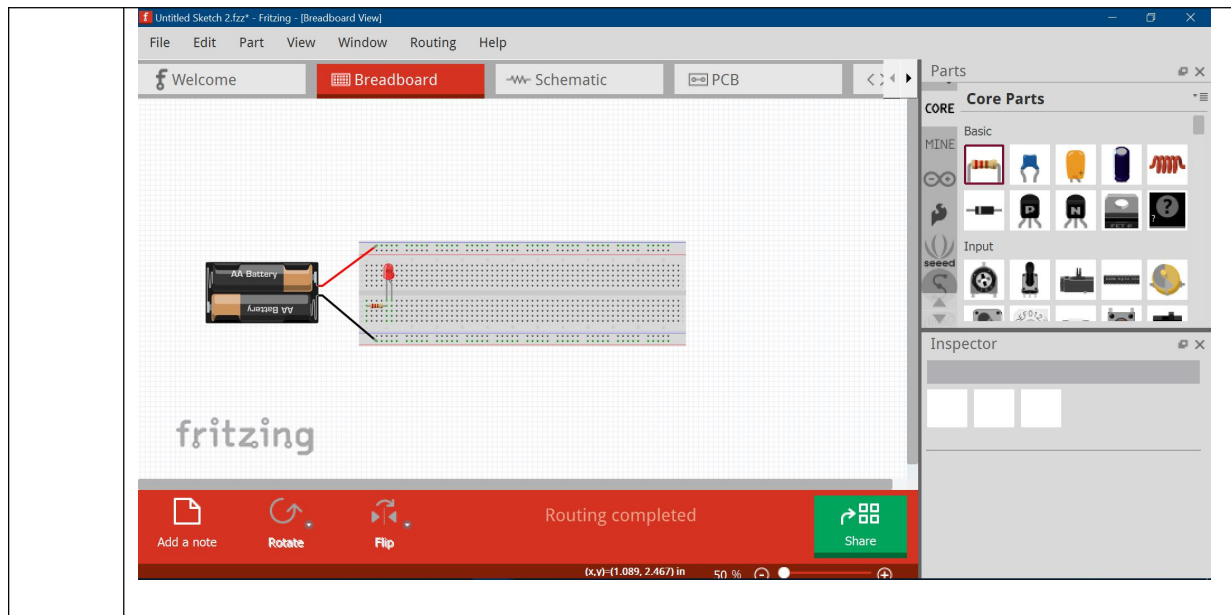


10 Run the standard application as below, then click the check button in the upper toolbar.

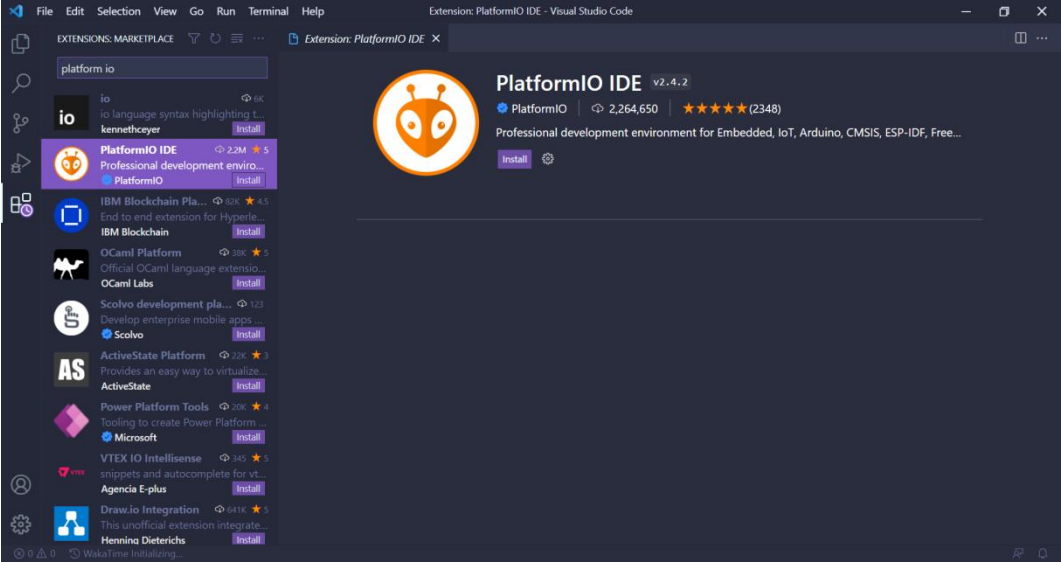
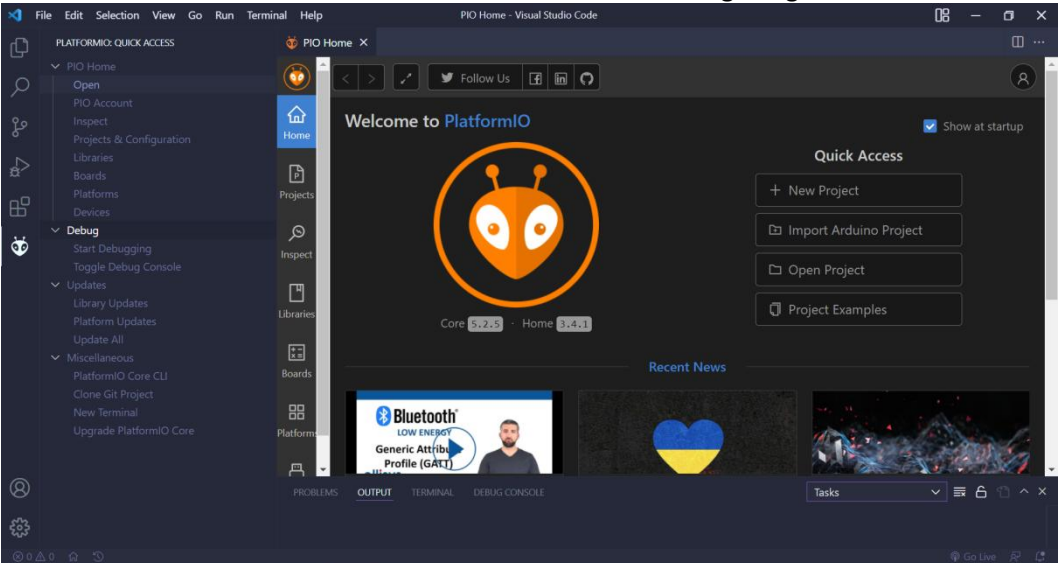


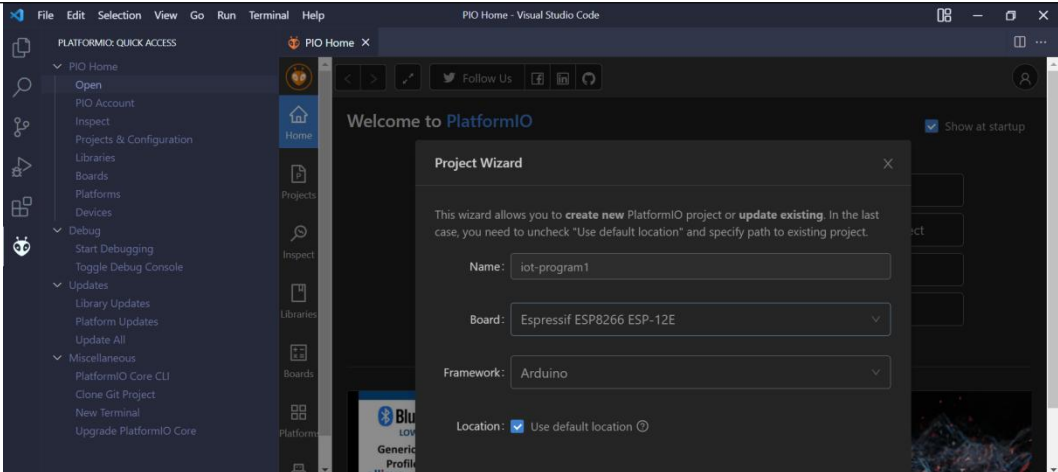
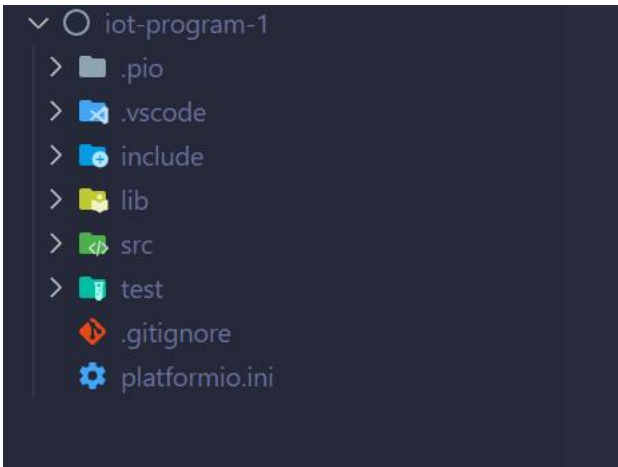
Fritzing Installation

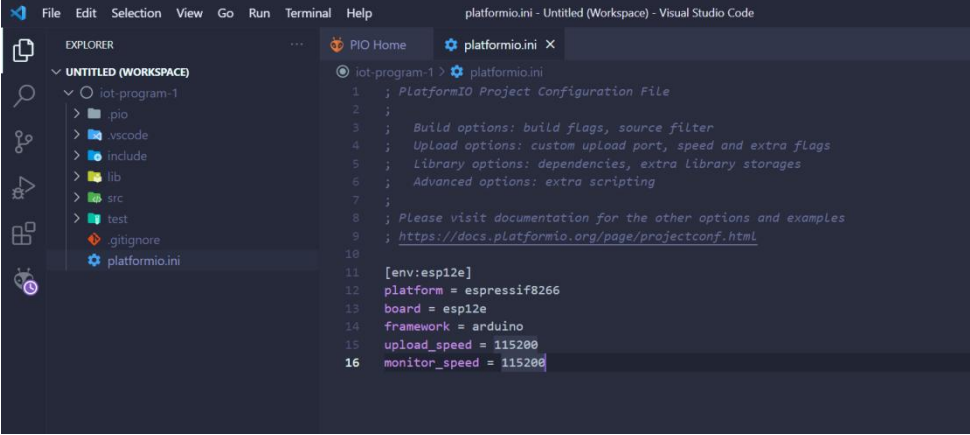
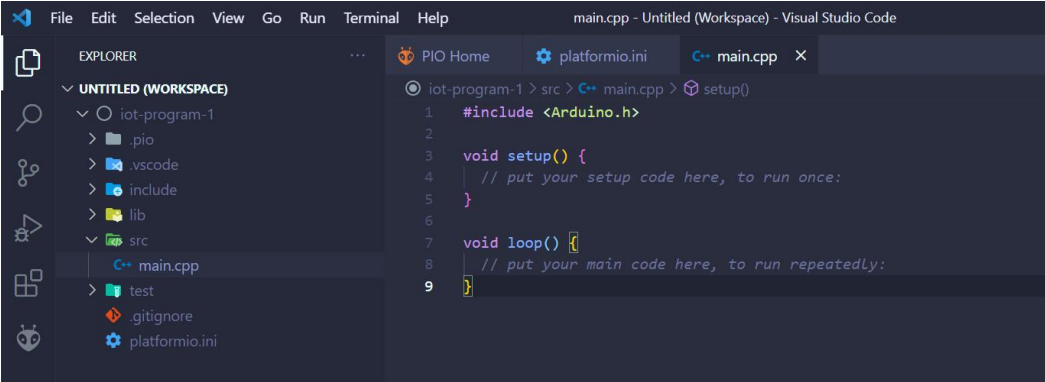
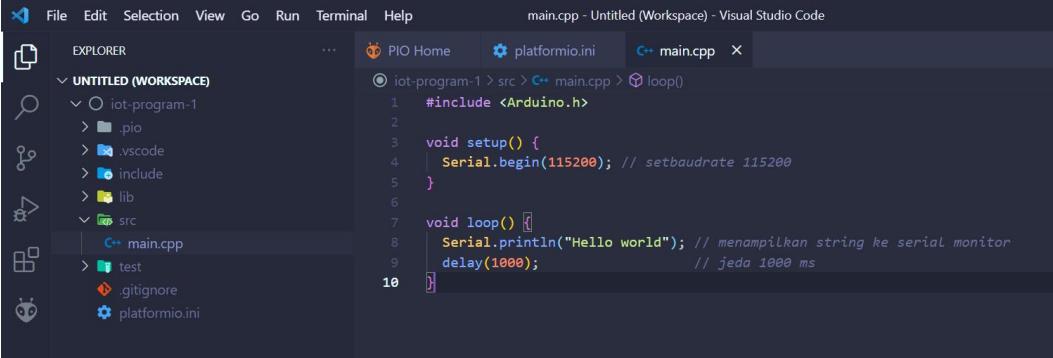
No	Step
1	<p>Download the Fritzing ZIP file at the following website http://fritzing.org/download/. Fritzing file options are tailored to the Windows operating system you have, for example 32bit or 64bit.</p> 
2	<p>For example the Fritzing windows 32bit file is fritzing.0.9.3b.32.pc.zip. Then unpack the ZIP file with the 7z application, Winzip or the like.</p>
3	<p>Run the Fritzing Fritzing.exe Application. If successful, the fritzing display will be as shown in Figure 2.9.</p> 
4	<p>You just need to drag and drop the component images on the right to the design media on the left. If the required components are not found, then you can download additional component libraries that can be obtained on the internet (usually files with the extension .fzpz). Next, import the .fzpz library file to be included in the new component library.</p>

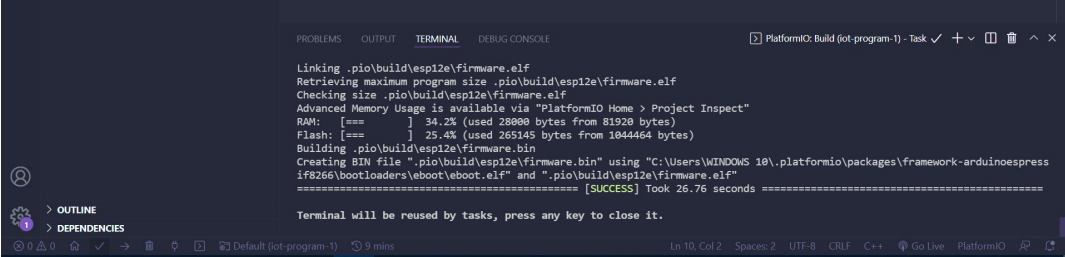
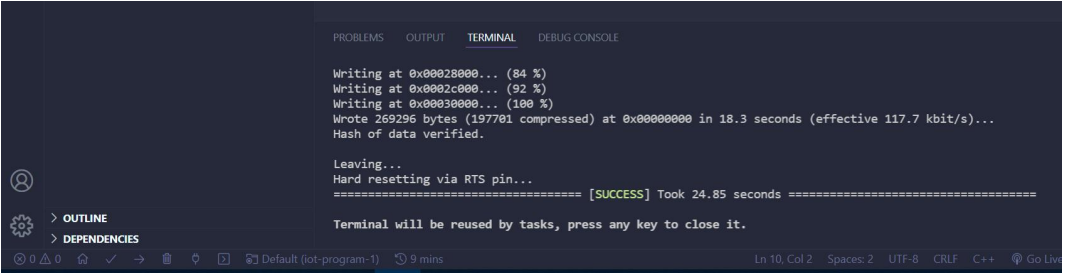
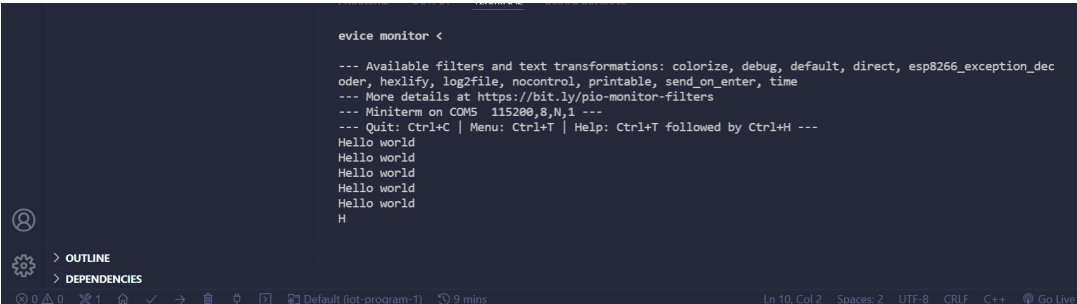


Visual Studio Code (Platform io) Installation

No	Step
1	<p>Please open the Visual Studio Code application, then click the extensions icon and type PlatformIO IDE as shown in the image below</p> 
2	<p>When successful, an alien-shaped icon will be added to the left sidebar, below the extensions icon. For more details can be seen in the following image</p> 
3	<p>Please create a new project by selecting New Project, so the wizard will appear as below. Fill in the Name is vs-program1, in the Board section type ESP8266 and select Espressif ESP8266 EFSP-12E, in the Framework section select Arduino, then the last Location for your project storage location. Please uncheck when you want to change the storage location of another directory.</p>

	
4	After clicking the Finish button, we will create a project template with a folder structure
5	<p>The explanation of the file and directory functions above is as follows</p> <ul style="list-style-type: none"> • .gitignore contains a list of files or folders that are not uploaded or pushed to the repository, git. • .pio is the folder that contains the results of the build of the board we are using, esp12e. • .vscode is a folder for storing Visual Studio Code configuration files. include is the folder where the project dependencies are stored. • lib is a folder to store the libraries we need • platformio.ini is our project's configuration file, for example the baudrate setting, the library used, and others. The contents of the platformio. this file are more or less like the following 
6	<p>Lines 11-14 have been created by the editor when we create a project, but we can change manually or want to add. Suppose we want the setting to equalize the baudare on the serial monitor, which is 115200. Using the property monitor_speed = 115200.</p>

	
7	<p>In general, the main program file structure is like the Arduino IDE, namely there is a setup() method and a loop() method. setup() is usually used to initialize variables and is executed only once when the program is running, while loop() will run the program code in it repeatedly.</p>  <p>Add a line of code to the setup() function as follows</p> <pre>Serial.begin(115200); // setbaudrate 115200</pre> <p>Also add code in loop function as follows</p> <pre>Serial.println("Hello world"); // menampilkan string ke serial monitor delay(1000); // jeda 1000 ms</pre> 
8	<p>Build your project using the build icon until the message on the console shows SUCCESS, to explain more it is shown in the image below</p>

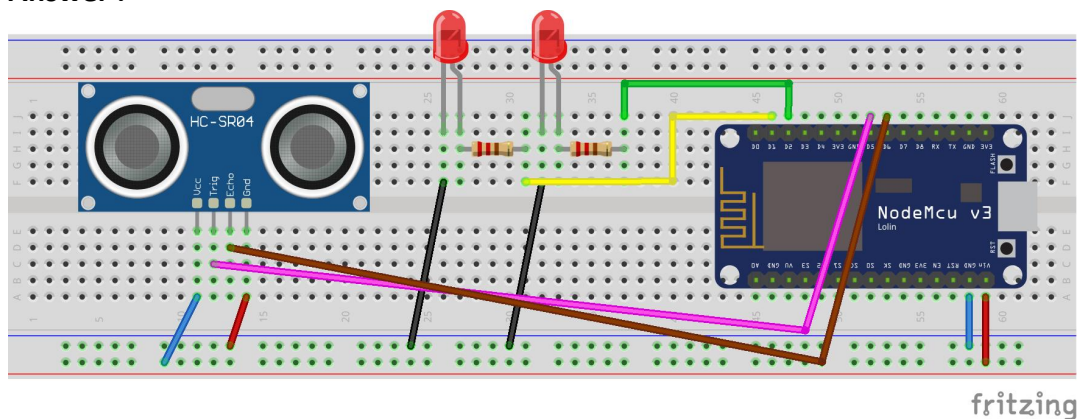
	
9	<p>Upload your project to run the program you created on the controller by clicking the Upload icon, as shown in the image below. If a SUCCESS message appears, it means that the upload process to the controller was successful.</p> 
10	<p>The last step is to see the results of the program that you created on the Serial Monitor, to see it please click the icon such as plug the power in. Please see the image below</p> 

TASK

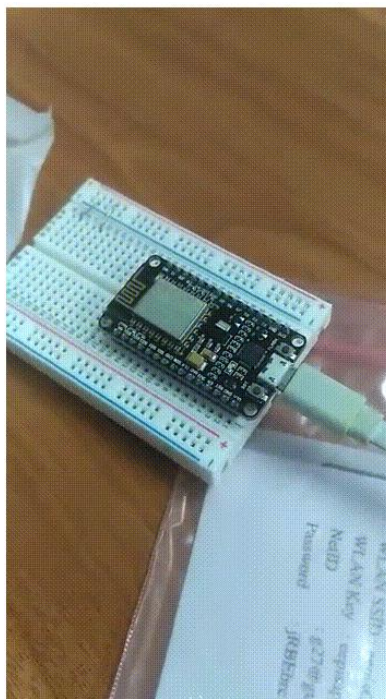
1. From today's activity, what has your group done? Mention if there are obstacles from these activities

Answer : From the activities I did today, I installed and connected esp8266 with the io platform. So far, no problems have been found. When connecting, esp8266 is immediately detected in device manager with COM5 port

2. Make a simple schematic of one of the sensors or actuators your group has purchased
Answer :



3. Make a simple code to turn on the red LED built into the MCU node as shown in the image below



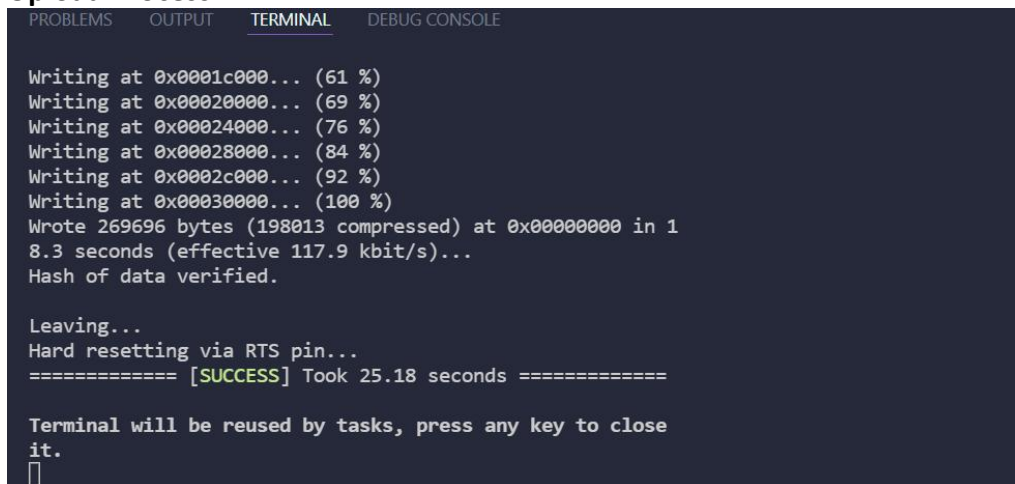
Answer :

Code Program:



```
platformio.ini  main.cpp x  Release Notes: 1.65.2  PIO Home
iot-program-1 > src > main.cpp > setup()
{
  5 void setup() {
  6   Serial.begin(115200);
  7   pinMode(LED_BUILTIN, OUTPUT);
  8 }
  9
 10 void loop() {
 11   digitalWrite(LED_BUILTIN, HIGH); // turn on RED LED
 12   delay(1000);
 13   digitalWrite(LED_BUILTIN, LOW);
 14   delay(1000);
 15 }
```

Upload Process :



```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

Writing at 0x0001c000... (61 %)
Writing at 0x00020000... (69 %)
Writing at 0x00024000... (76 %)
Writing at 0x00028000... (84 %)
Writing at 0x0002c000... (92 %)
Writing at 0x00030000... (100 %)
Wrote 269696 bytes (198013 compressed) at 0x00000000 in 1
8.3 seconds (effective 117.9 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
===== [SUCCESS] Took 25.18 seconds =====

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

Output :

