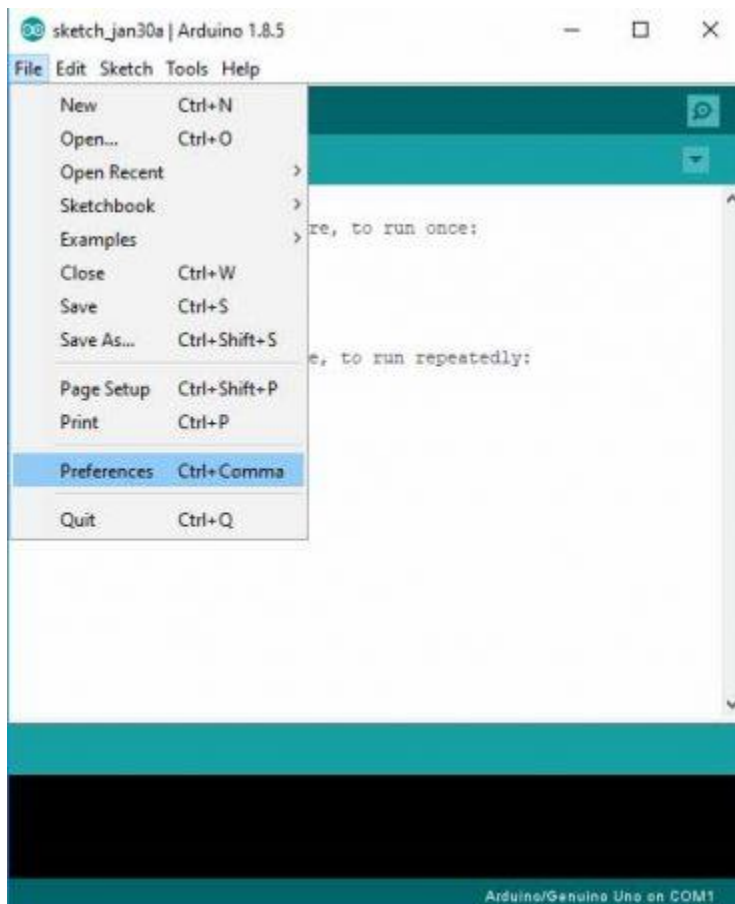


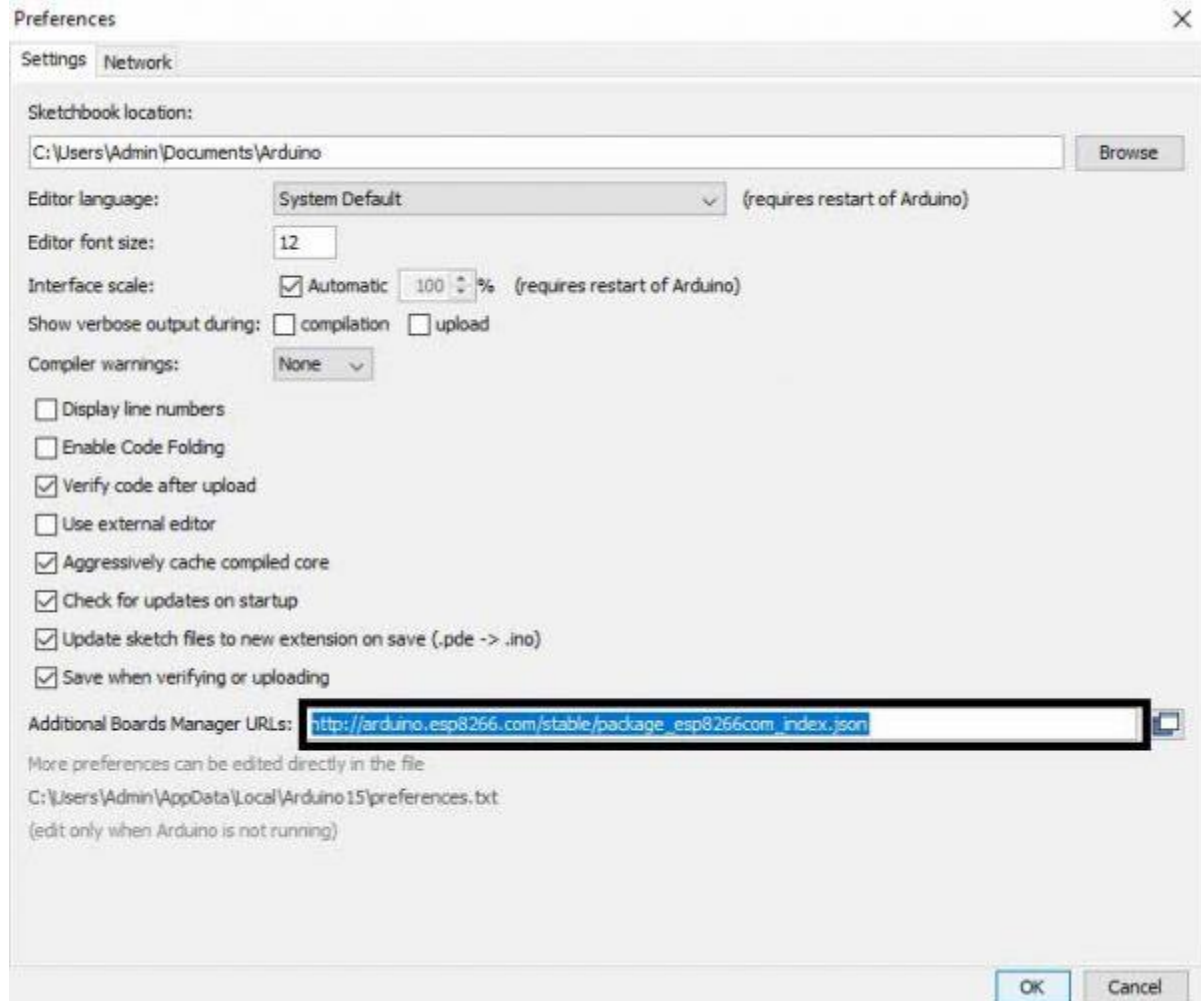
# First Hands on with ESP8266

In this example we will build a simple circuit to use ESP8266 GPIO capabilities to drive a LED bulb on and off periodically.

First setup the repo so that Arduino knows where to find ESP boards



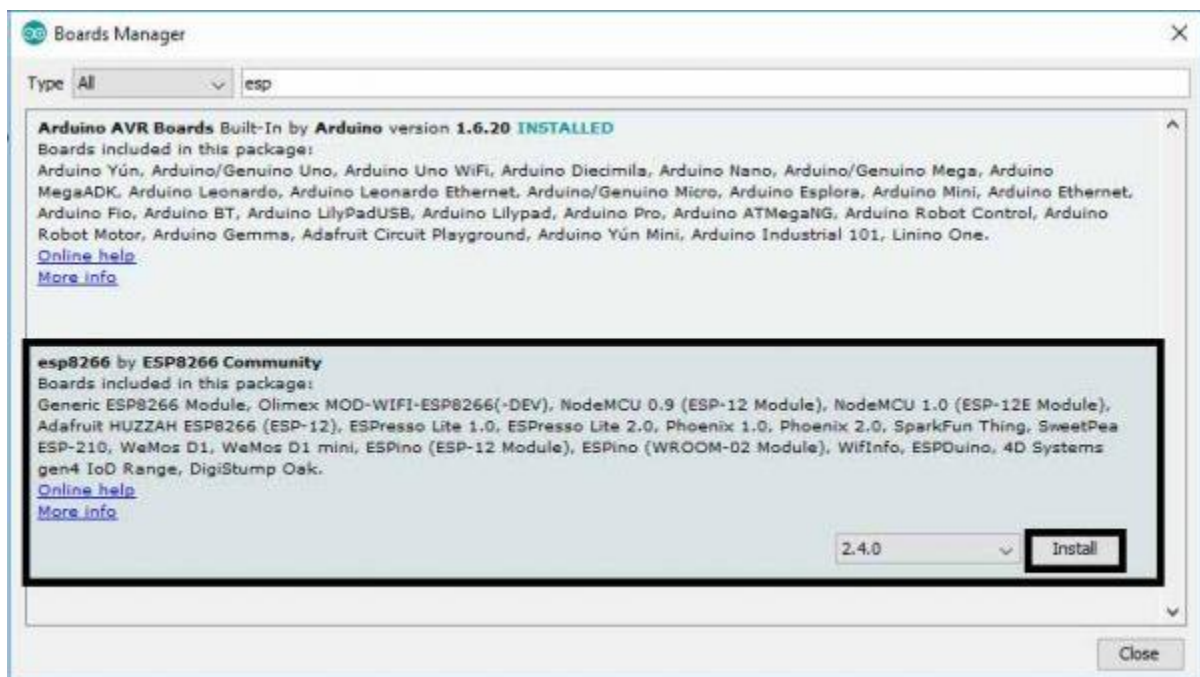
The, enter the URL “[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)” into Additional Board Manager URLs field and click the “OK” button



Now we are ready to add ESP board to the Arduino IDE. Open board manager. Tools -> Board -> Boards Manager

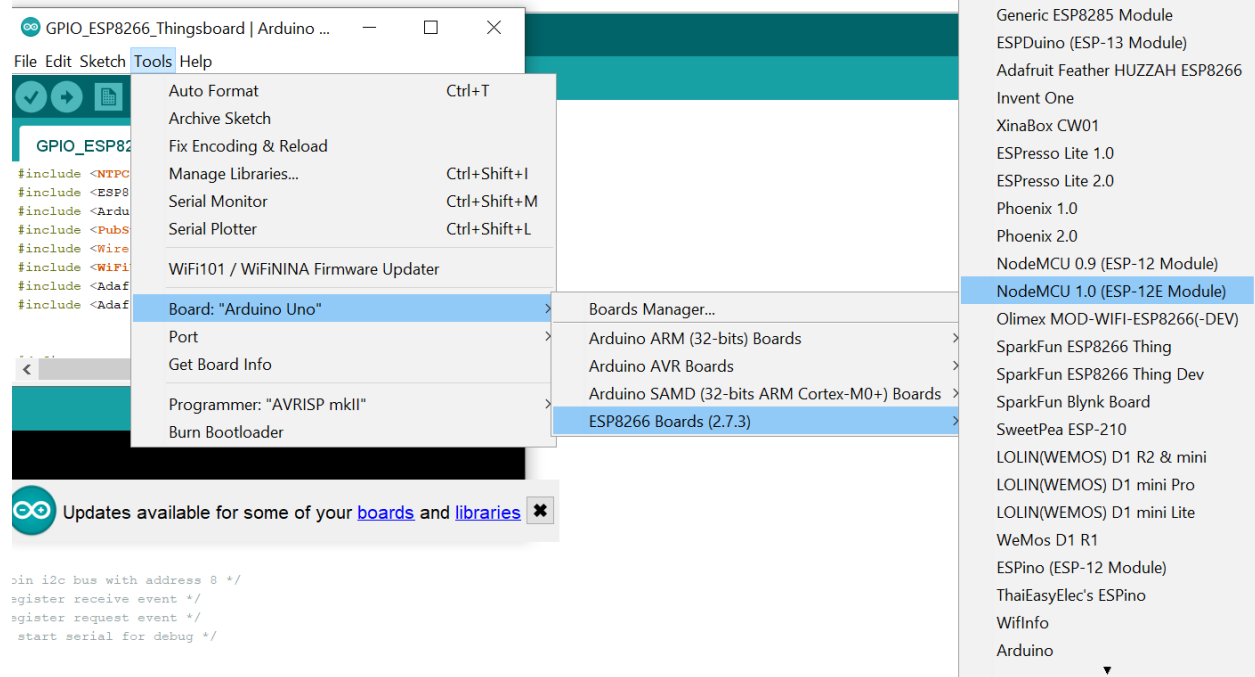


It will open the board manager window. Search esp8266 platform, select latest version, and install it

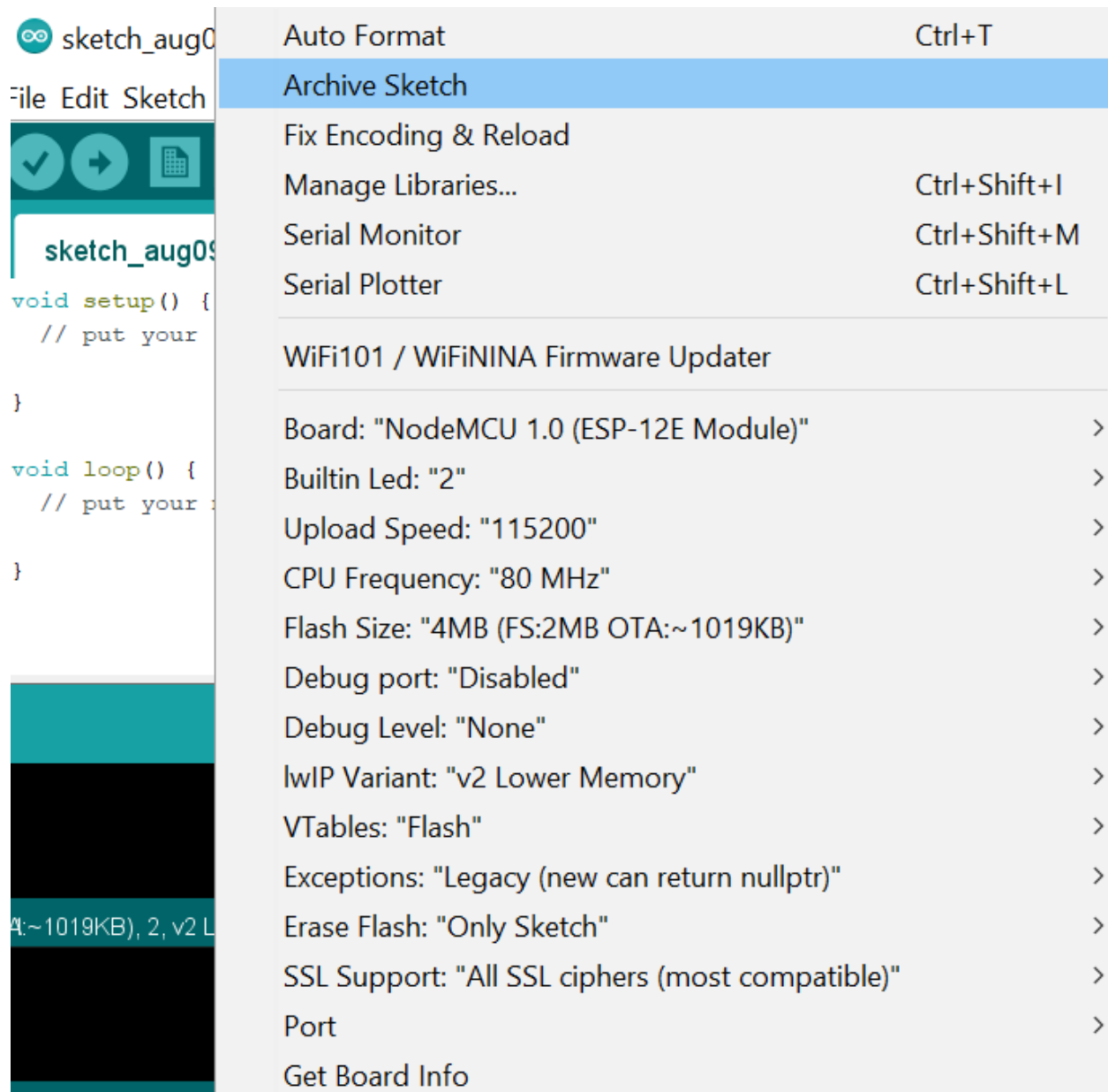


Once installed you are ready to start working with the ESP8266, Select Node MCU 1.0 as your board. This is the development board for ESP8266 that we are using today.

Windows Store 1.8.57.0)



You will see something like this



Now we are ready to program

```

#define MY_BLUE_LED_PIN 2

void setup() {
  pinMode(MY_BLUE_LED_PIN, OUTPUT); // Initialise the LED_BUILTIN pin as an output
}

// the loop function runs over and over again forever
void loop() {

```

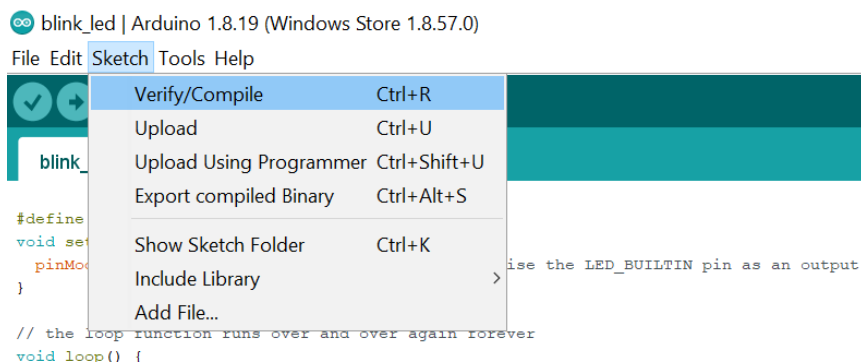
```
digitalWrite(MY_BLUE_LED_PIN, LOW); // Turn the LED on (Note that LOW is the voltage level
// but actually the LED is on; this is because
// it is active low on the ESP-01)

delay(1000);          // Wait for a second

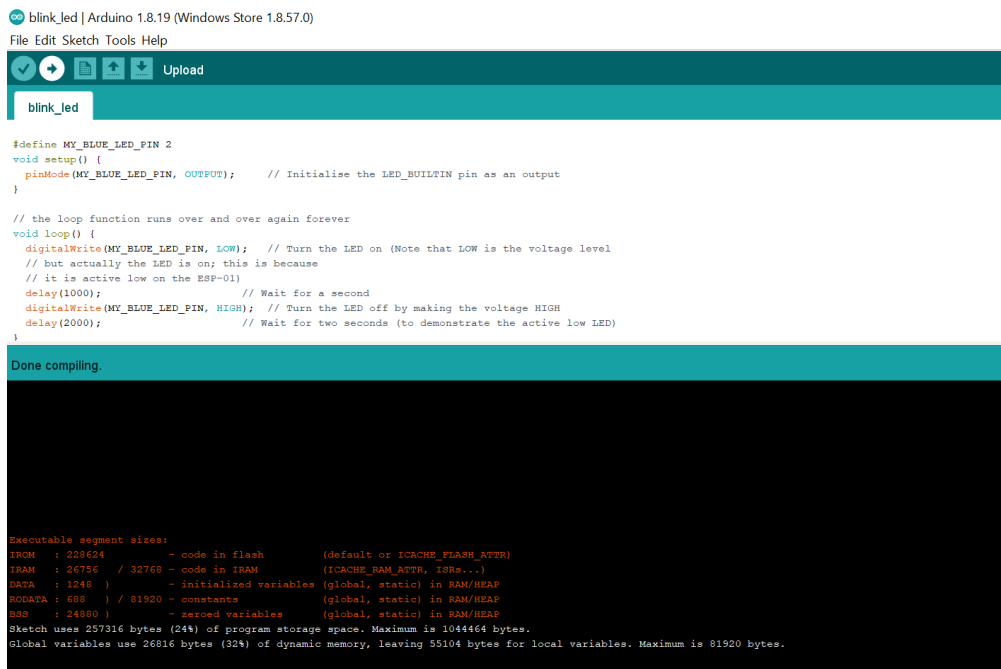
digitalWrite(MY_BLUE_LED_PIN, HIGH); // Turn the LED off by making the voltage HIGH

delay(2000);          // Wait for two seconds (to demonstrate the active low LED)
}
```

## Now compile the program

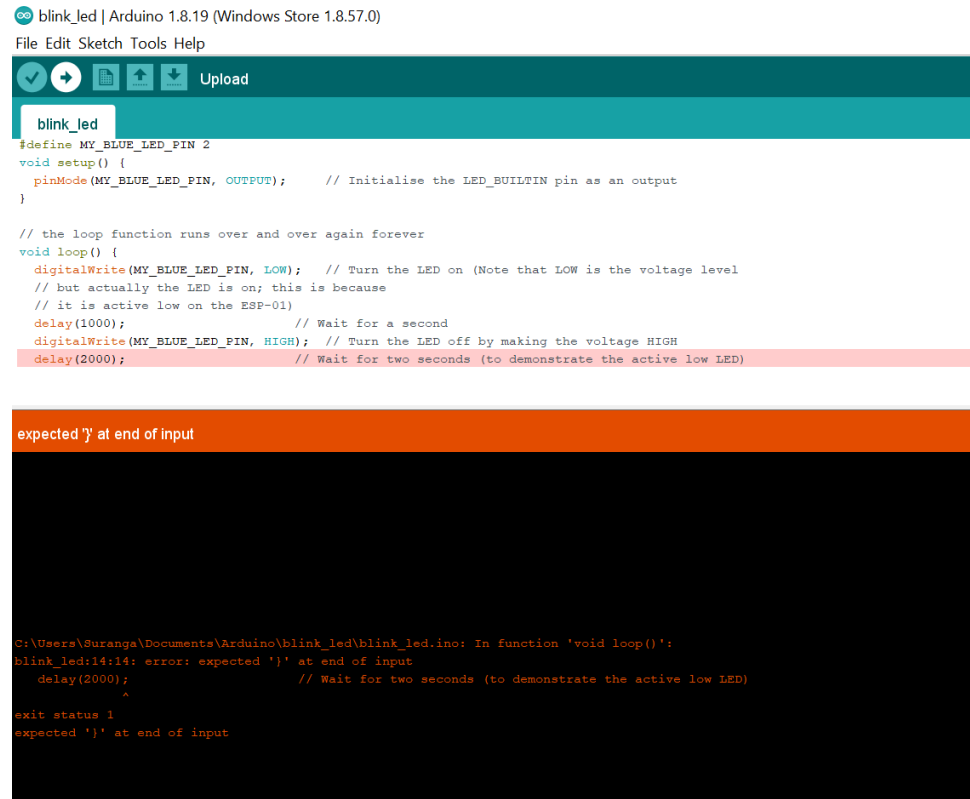


If program compile successfully, you will see following output



If your program is having issues with compiling, look for the error messages.

For example, below the code is missing closing bracket, as you can see, most of the time compiler will point you in to the correct error. However, sometimes, it may not point you to the correct error. You may have to analyze the logic of the code and identify the error.



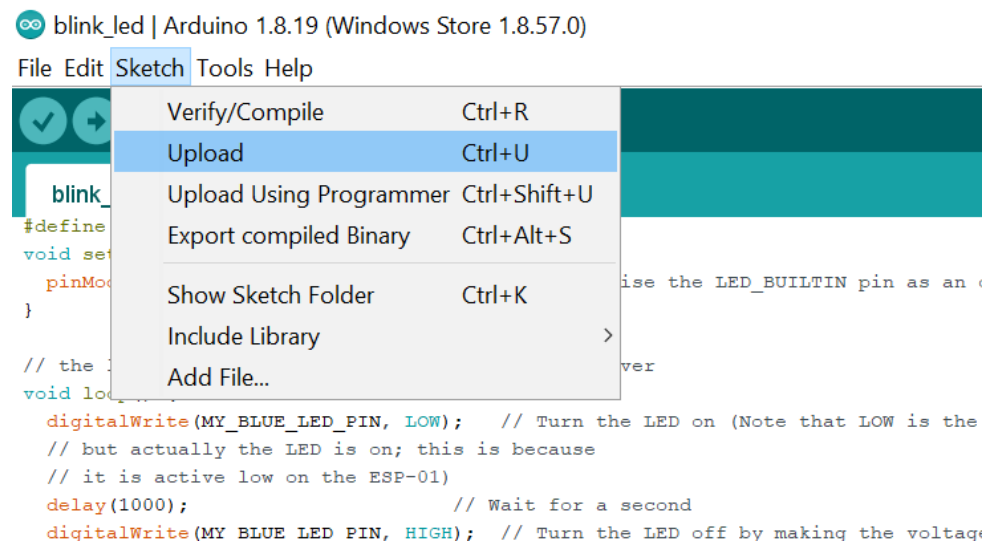
```
blink_led | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
Upload
blink_led
#define MY_BLUE_LED_PIN 2
void setup() {
  pinMode(MY_BLUE_LED_PIN, OUTPUT); // Initialise the LED_BUILTIN pin as an output
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(MY_BLUE_LED_PIN, LOW); // Turn the LED on (Note that LOW is the voltage level
  // but actually the LED is on; this is because
  // it is active low on the ESP-01)
  delay(1000); // Wait for a second
  digitalWrite(MY_BLUE_LED_PIN, HIGH); // Turn the LED off by making the voltage HIGH
  delay(2000); // Wait for two seconds (to demonstrate the active low LED)
}
```

expected \')' at end of input

C:\Users\Suranga\Documents\Arduino\blink\_led\blink\_led.ino: In function 'void loop()':  
blink\_led:14:14: error: expected ')' at end of input  
 delay(2000); // Wait for two seconds (to demonstrate the active low LED)  
 ^  
exit status 1  
expected \')' at end of input

Once code is correctly compiled, we can now transfer the code to the device.



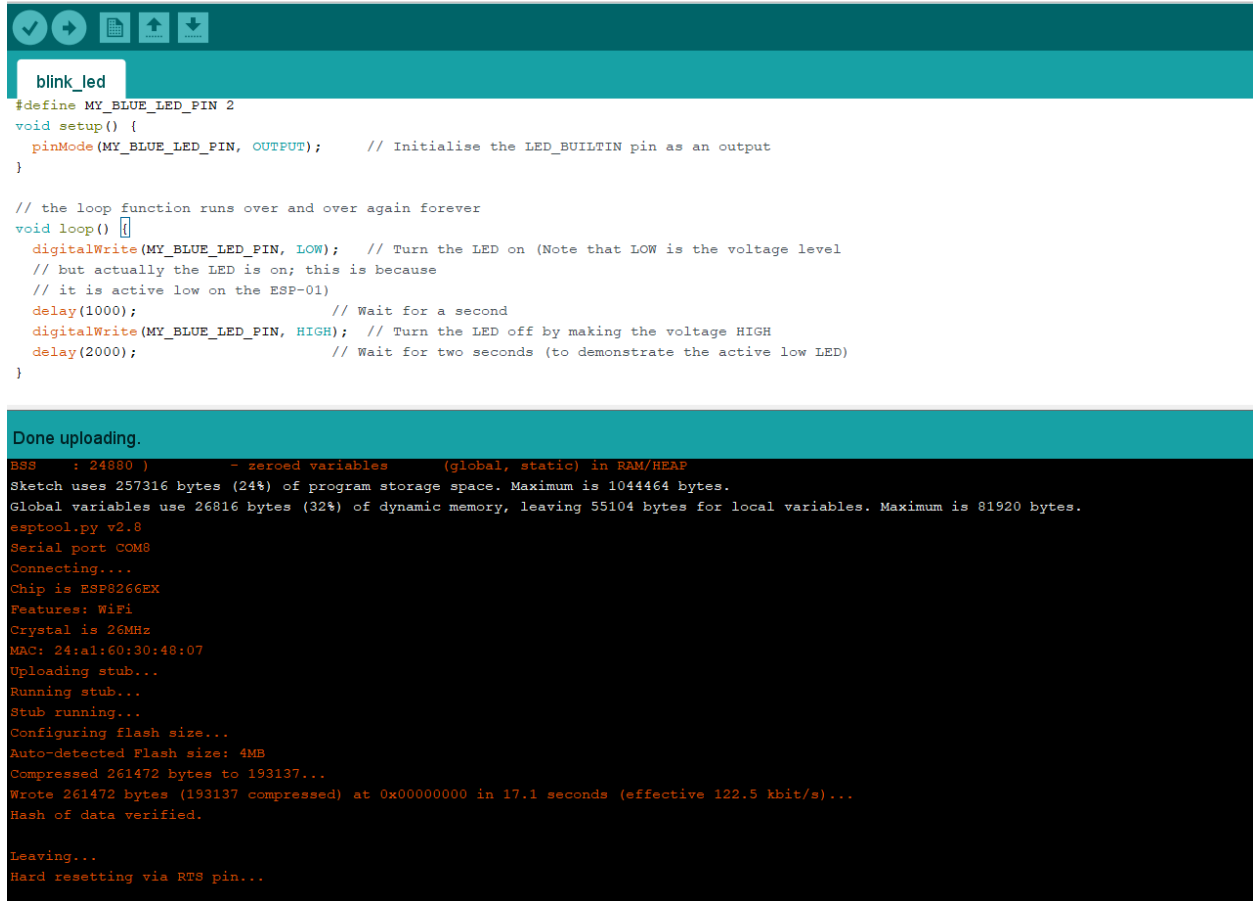
```
blink_led | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
#define MY_BLUE_LED_PIN 2
void setup() {
  pinMode(MY_BLUE_LED_PIN, OUTPUT); // Initialise the LED_BUILTIN pin as an output
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(MY_BLUE_LED_PIN, LOW); // Turn the LED on (Note that LOW is the voltage level
  // but actually the LED is on; this is because
  // it is active low on the ESP-01)
  delay(1000); // Wait for a second
  digitalWrite(MY_BLUE_LED_PIN, HIGH); // Turn the LED off by making the voltage HIGH
  delay(2000); // Wait for two seconds (to demonstrate the active low LED)
}
```

If everything goes fine, you will see a message like this.

🔌 blink\_led | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help



```
blink_led
#define MY_BLUE_LED_PIN 2
void setup() {
  pinMode(MY_BLUE_LED_PIN, OUTPUT);    // Initialise the LED_BUILTIN pin as an output
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(MY_BLUE_LED_PIN, LOW);  // Turn the LED on (Note that LOW is the voltage level
  // but actually the LED is on; this is because
  // it is active low on the ESP-01)
  delay(1000);                          // Wait for a second
  digitalWrite(MY_BLUE_LED_PIN, HIGH);  // Turn the LED off by making the voltage HIGH
  delay(2000);                          // Wait for two seconds (to demonstrate the active low LED)
}
```

```
Done uploading.
BSS      : 24880 )          - zeroed variables      (global, static) in RAM/HEAP
Sketch uses 257316 bytes (24%) of program storage space. Maximum is 1044464 bytes.
Global variables use 26816 bytes (32%) of dynamic memory, leaving 55104 bytes for local variables. Maximum is 81920 bytes.
esptool.py v2.8
Serial port COM8
Connecting....
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 24:a1:60:30:48:07
Uploading stub...
Running stub...
Stub running...
Configuring Flash size...
Auto-detected Flash size: 4MB
Compressed 261472 bytes to 193137...
Wrote 261472 bytes (193137 compressed) at 0x00000000 in 17.1 seconds (effective 122.5 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Once the program uploaded successfully, ESP8266 get a rest set, so that code start running. You will see blinking LED!!

<https://youtube.com/shorts/rRH2XkKwKPA?feature=share>

Congratulations !! you now have successfully completed the first exercise with ESP8266 microcontroller