


NodeMCU is an open-source [LUA](#) based firmware developed for the ESP8266 wifi chip. (More on this tomorrow)

## 1. Installing Arduino IDE:

<https://www.arduino.cc/en/software>

## Downloads



### Arduino IDE 2.3.3

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

#### DOWNLOAD OPTIONS

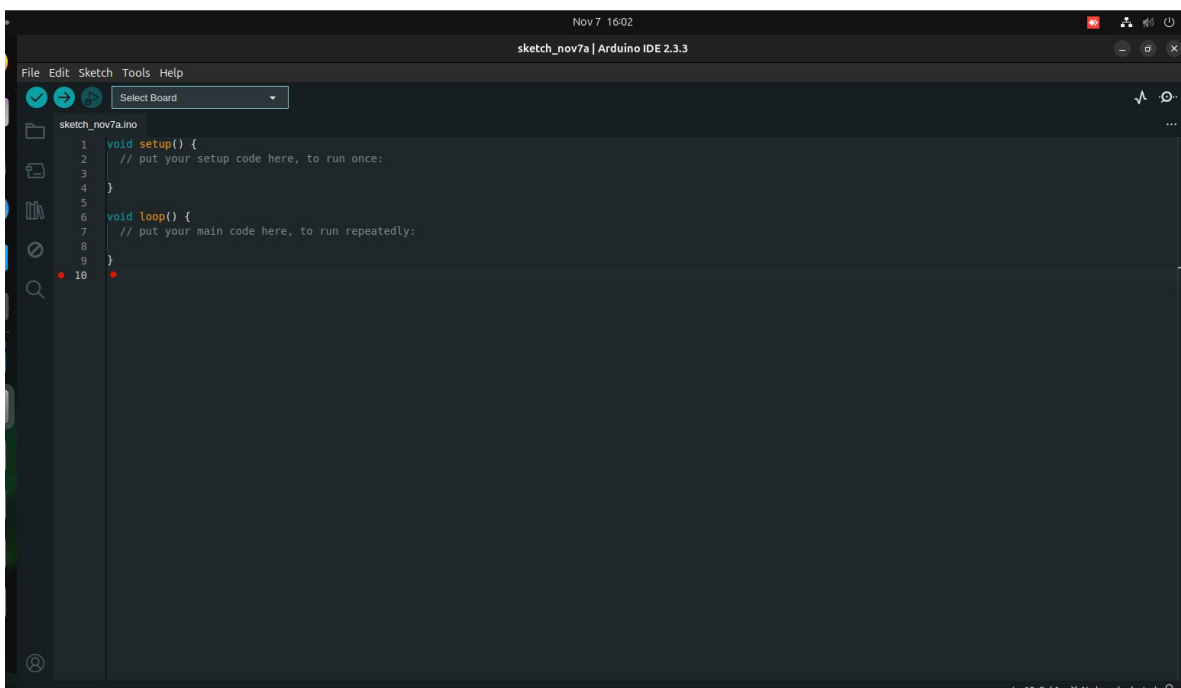
**Windows** Win 10 and newer, 64 bits  
**Windows** MSI installer  
**Windows** ZIP file

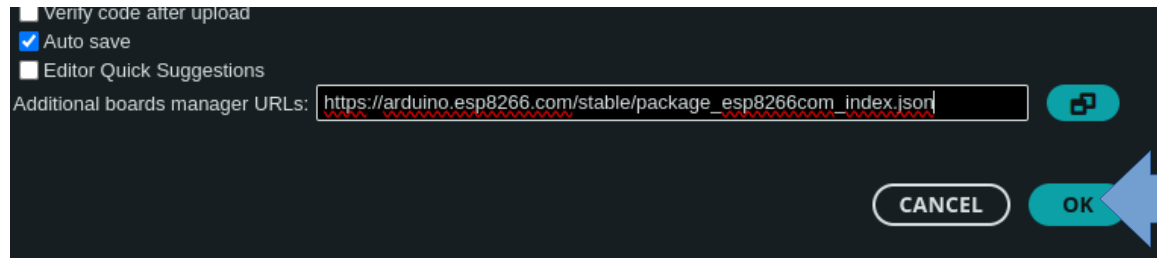
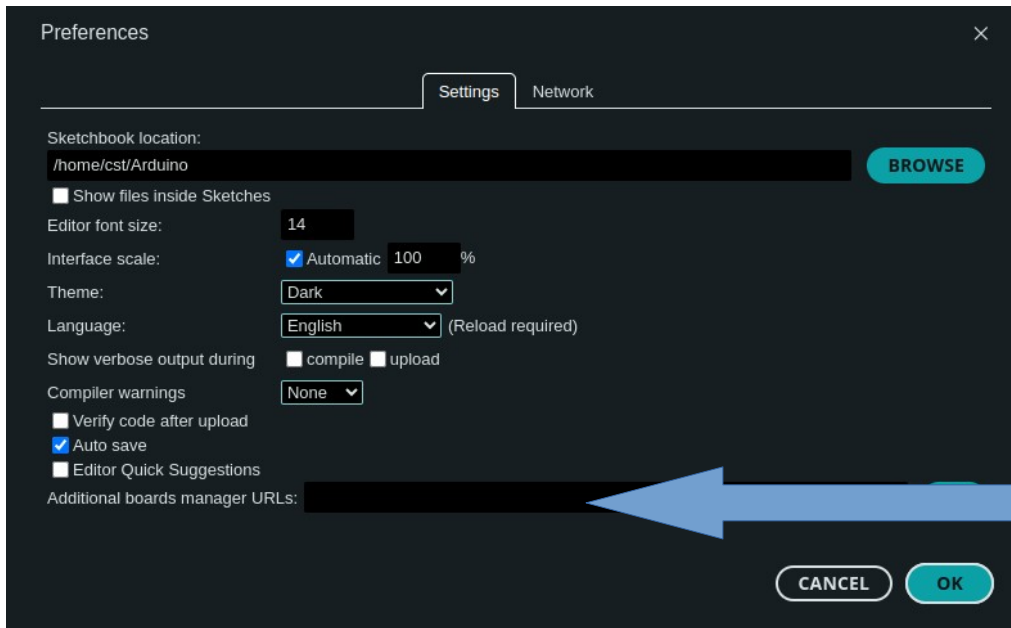
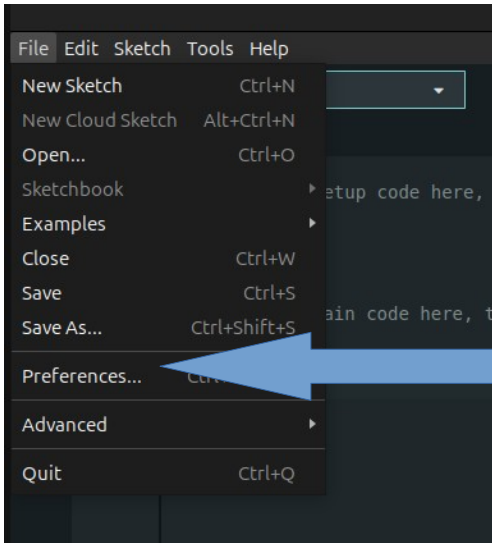
**Linux** AppImage 64 bits (X86-64)  
**Linux** ZIP file 64 bits (X86-64)

**macOS** Intel, 10.15: "Catalina" or newer, 64 bits  
**macOS** Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

## 2. Installing ESP8266 Board Manager:

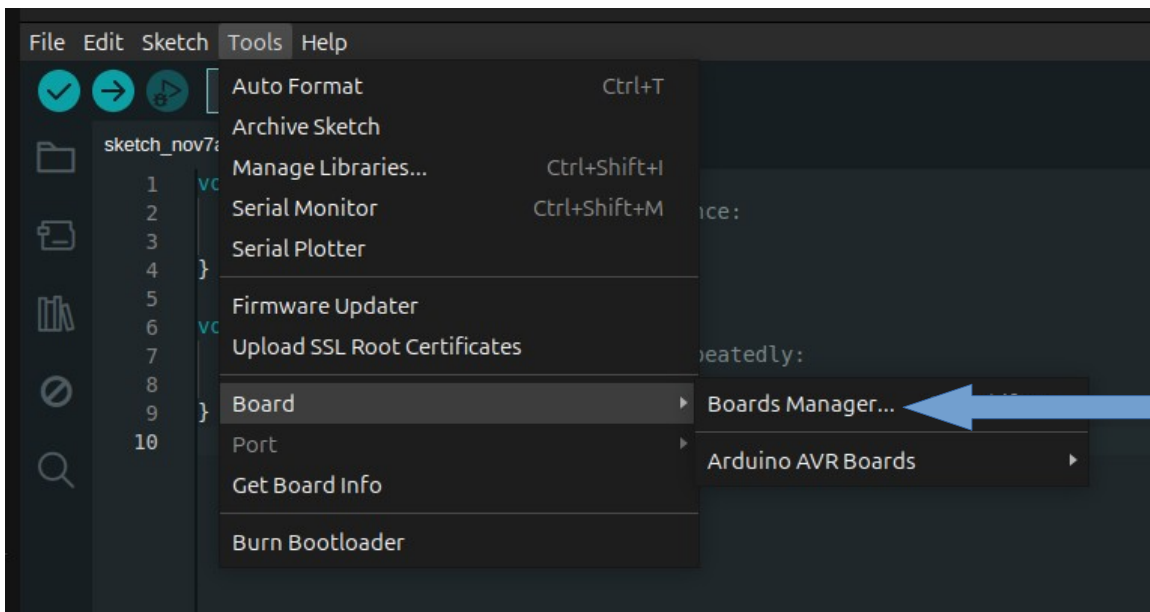
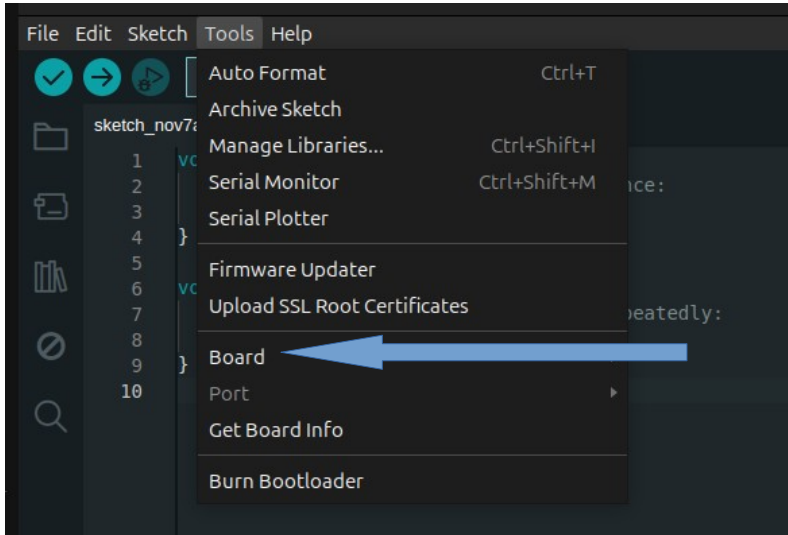


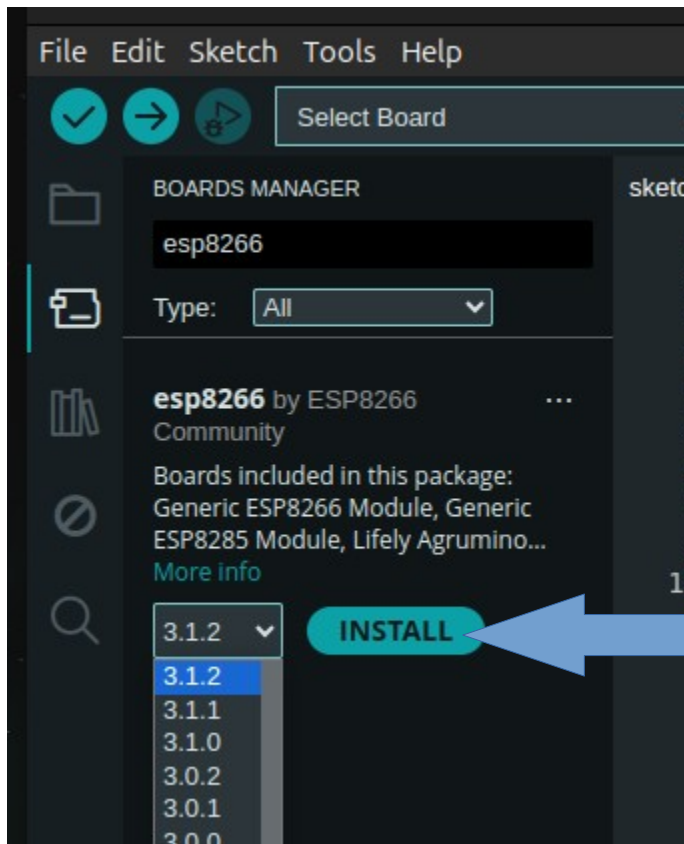
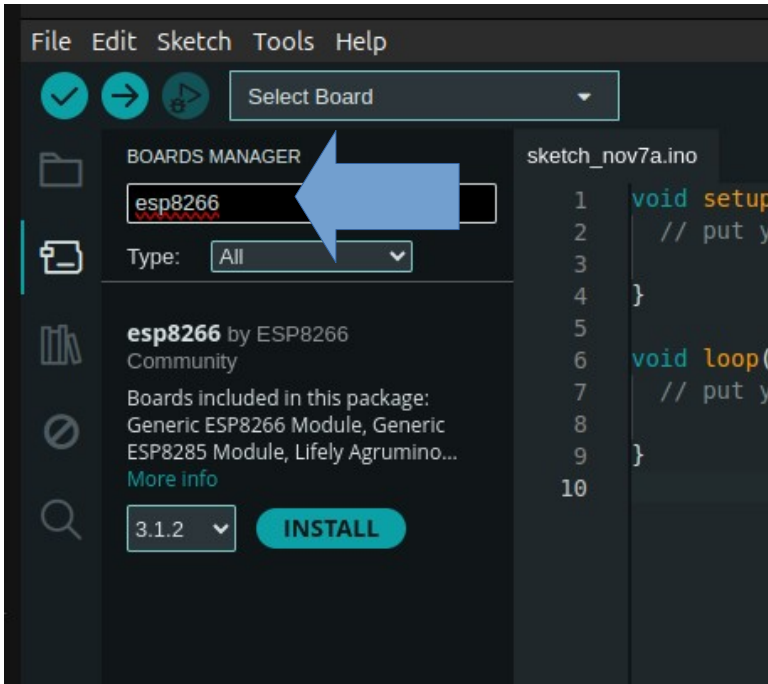


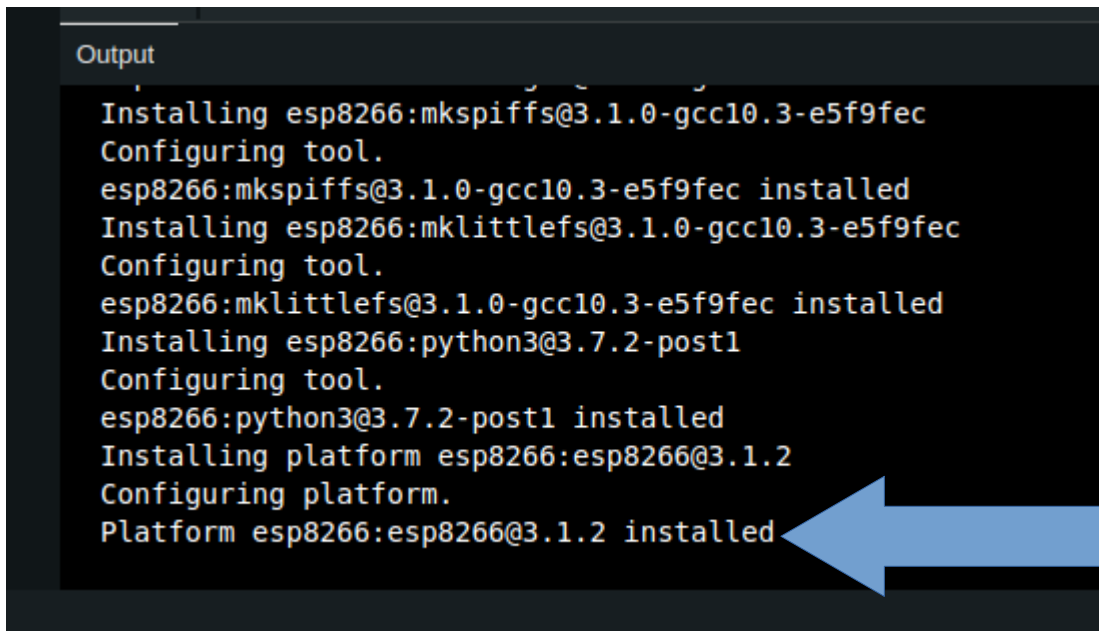
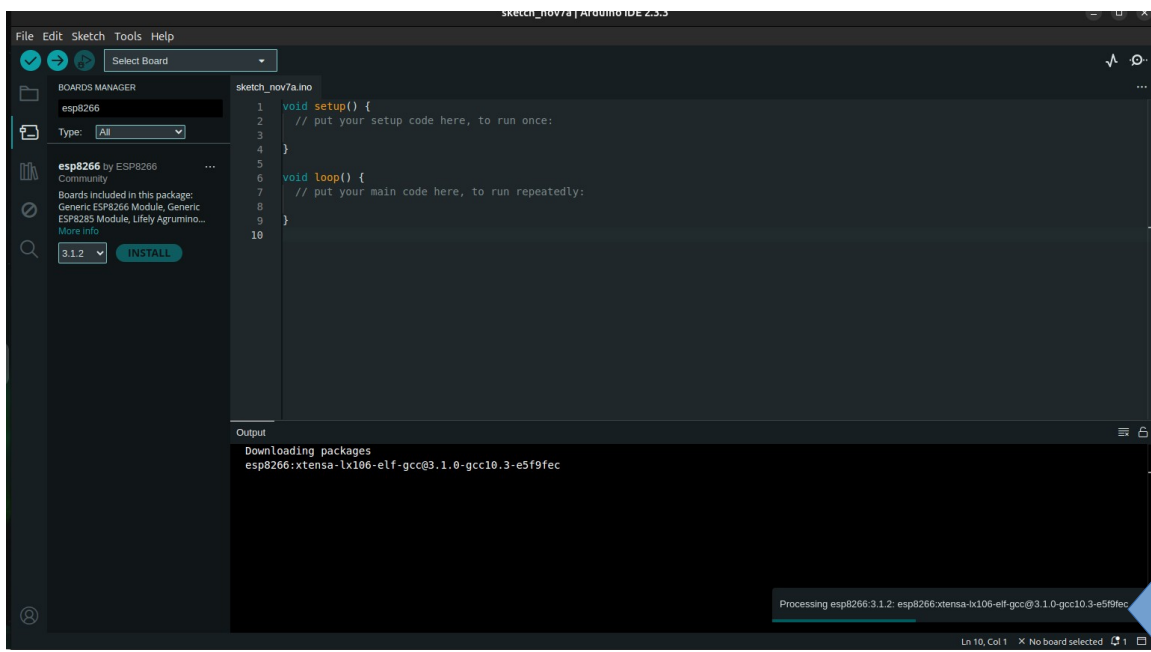
[https://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](https://arduino.esp8266.com/stable/package_esp8266com_index.json)

## Close the Window

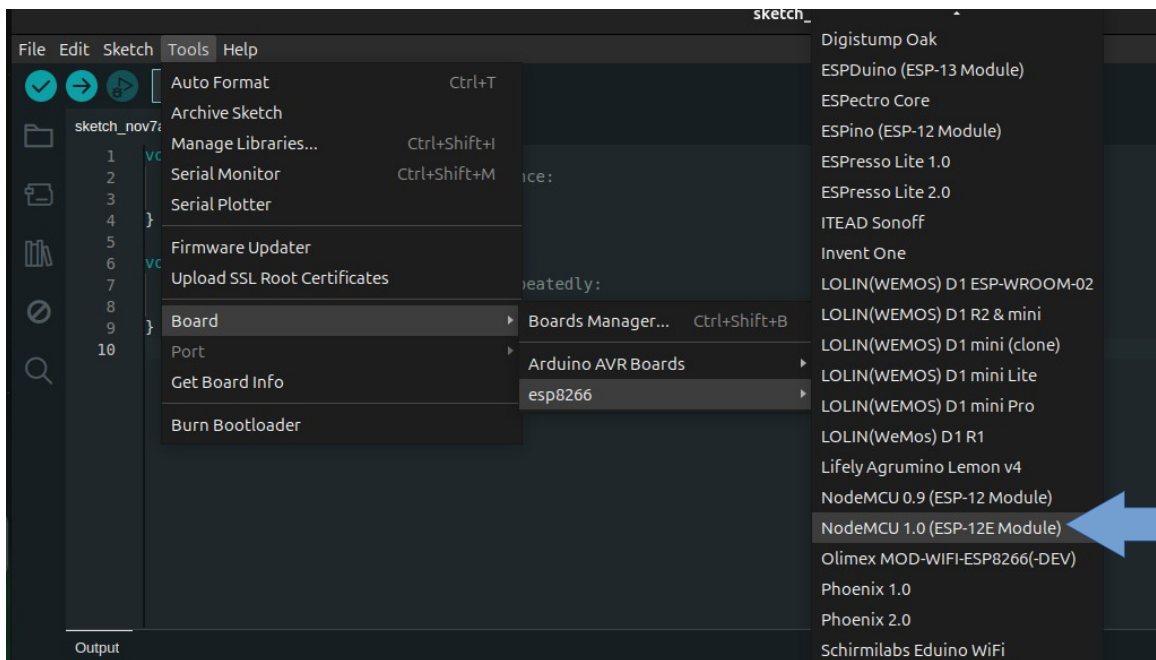
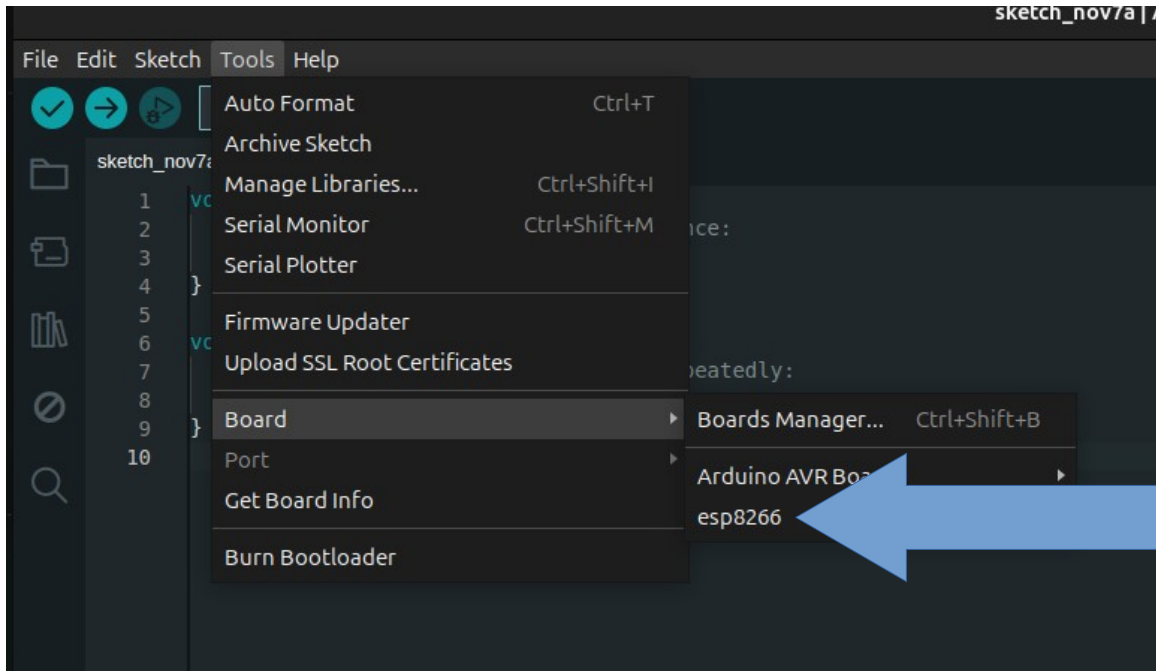
## Open ArduinoIDE once again

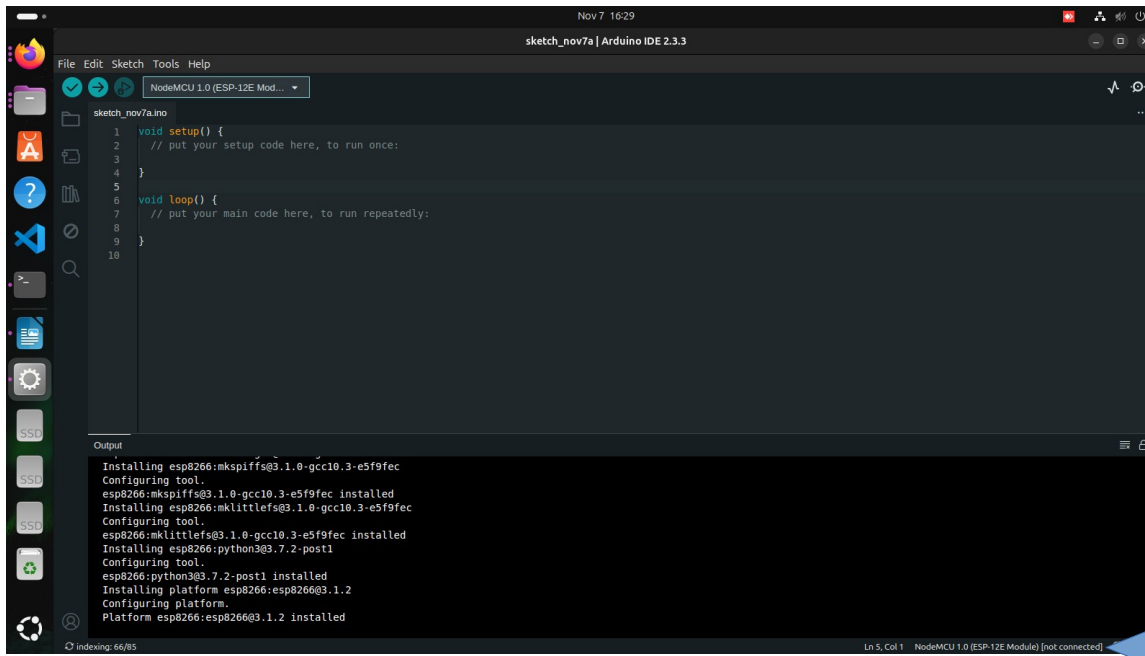
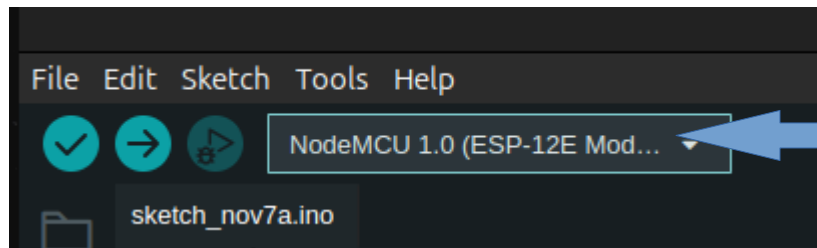






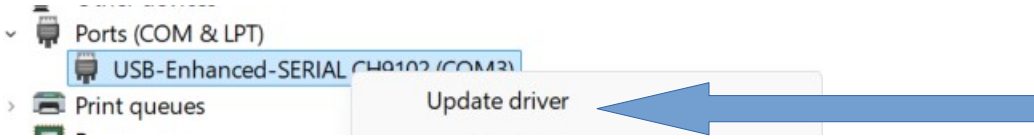
### 3. Selecting NodeMCU 1.0 board:





## 4. First Contact: (Installing Device Drivers)

### Reopen ArduinoIDE after first contact



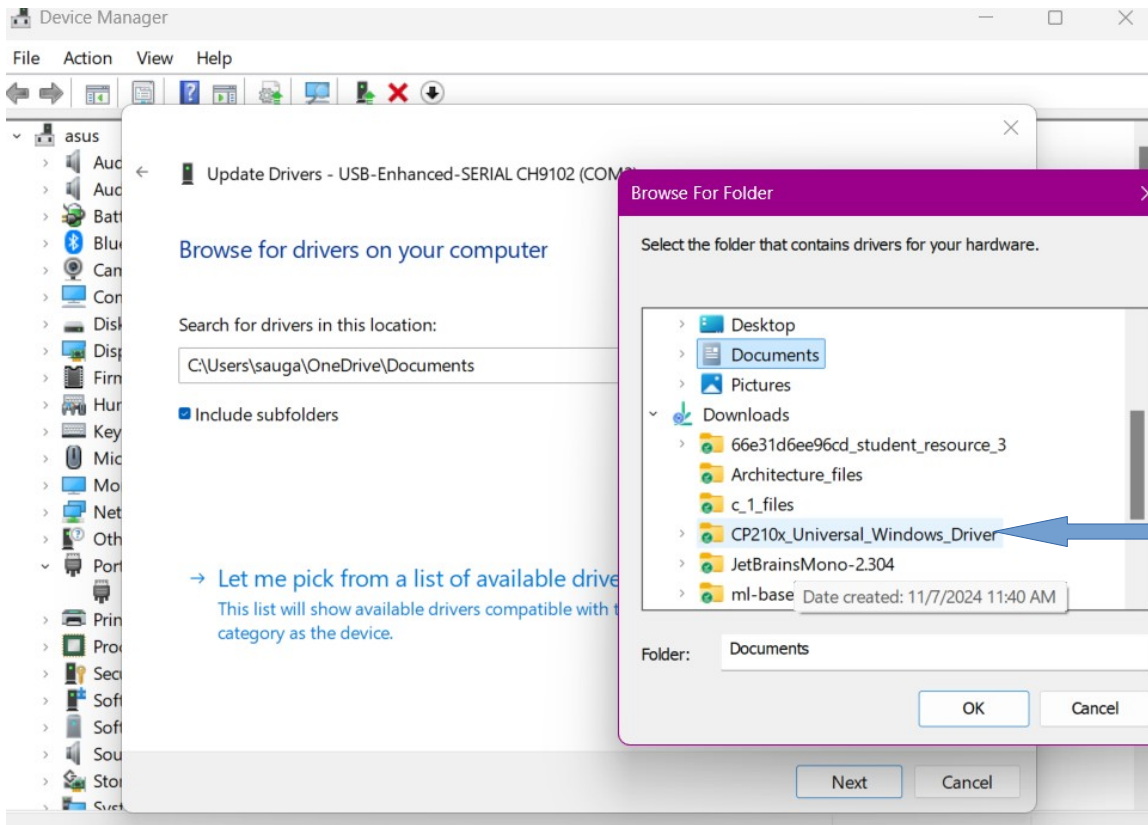
How do you want to search for drivers?

→ Search automatically for drivers

Windows will search your computer for the best available driver and install it on your device.

→ Browse my computer for drivers

Locate and install a driver manually.



## 5. Sample Codes:

### a. Blink an LED

```
// Define the LED pin (D4 on NodeMCU corresponds to GPIO 2)
const int ledPin = D4;

void setup() {
  // Initialize the LED pin as an output
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // Turn the LED on (LOW for active-low LED configuration on NodeMCU)
  digitalWrite(ledPin, LOW);
  delay(500); // Wait for 500 milliseconds (0.5 seconds)

  // Turn the LED off (HIGH for active-low LED configuration)
  digitalWrite(ledPin, HIGH);
  delay(500); // Wait for 500 milliseconds (0.5 seconds)
}
```

### b. Display String in Serial Monitor

```
void setup() {
```

```
// Start serial communication at 115200 baud rate
Serial.begin(115200);

// Wait for the serial monitor to open
while (!Serial) {
  ; // Do nothing until the serial port is ready
}

// Print a message to the serial monitor
Serial.println("Hello, NodeMCU!");
}

void loop() {
  // Continuously display a message
  Serial.println("NodeMCU is running..");
  delay(1000); // Wait for 1 second
}
```