Working with Node_MCU v1.0

Step 1: Please download and install the latest version of Arduino IDE using this <u>link.</u>

Step 2: Start Arduino software and follow these options from menu bar: **File>>Preferences**.

Step 3: In **Preferences** window go to **settings** tab as shown in Figure 1. Copy the below URL in the Additional boards Manager URLs box and press "OK".

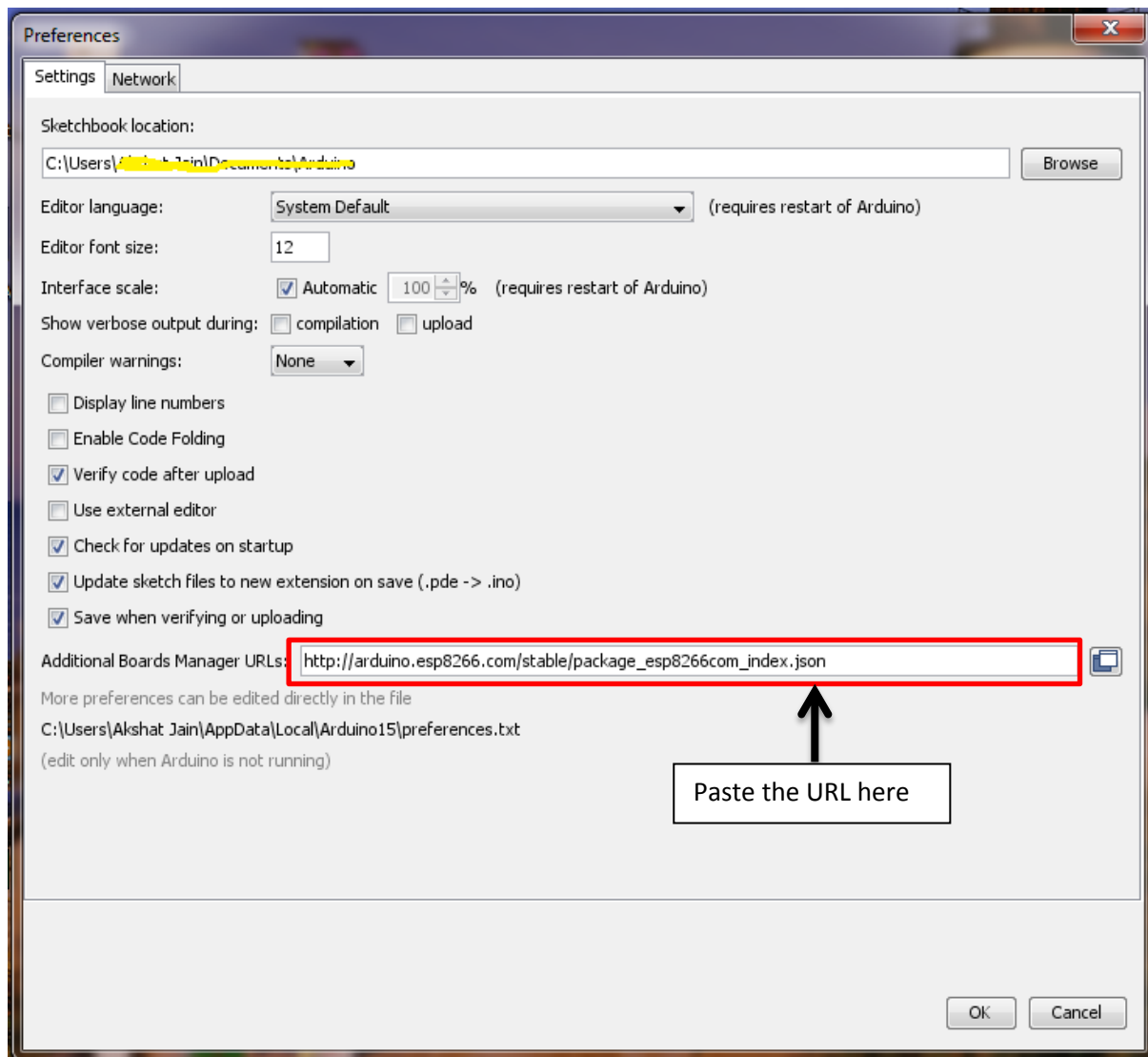**URL: http://arduino.esp8266.com/stable/package_esp8266com_index.json**

Figure 1: Preferences window

Step 4: Now, follow these options: **Tools>> Board>> Boards Manager.**

NOTE: Please make sure to have a working internet connection as you need to download files via Boards Manager. Don't use a proxy based internet connection.

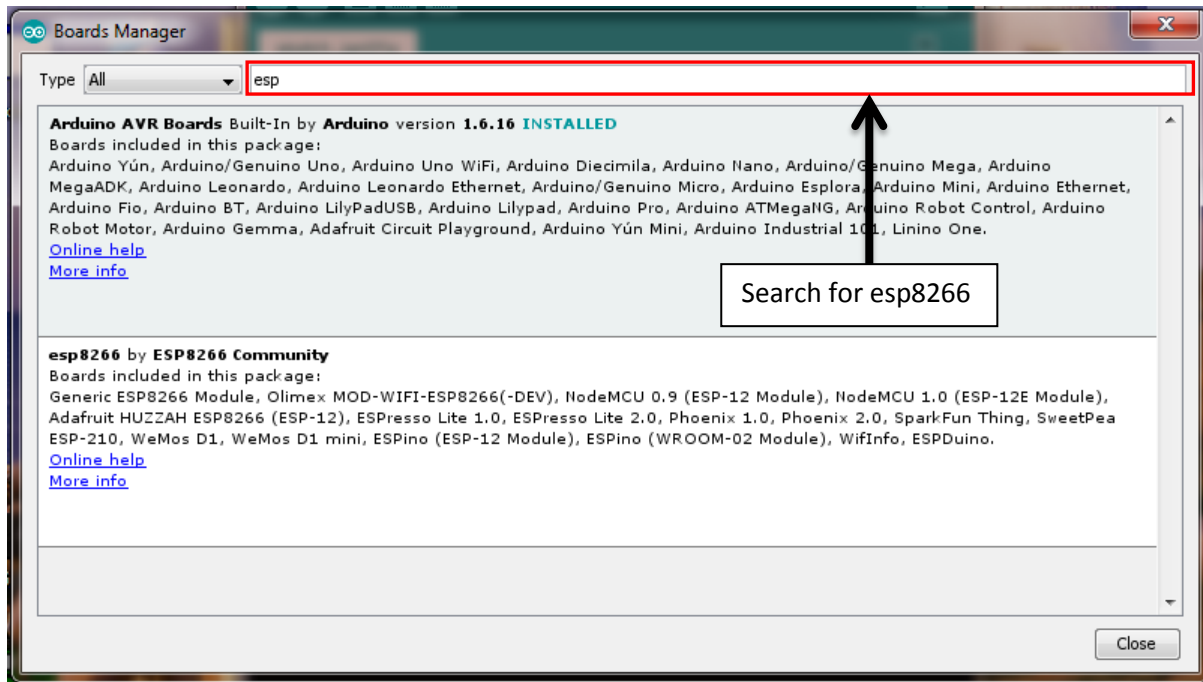**Step 5:** In **Boards Manager** window, search: **esp8266** as shown in Figure 2.

Figure 2: Boards Manager

Step 6: Click on "**esp8266 by ESP8266 Community**" and install as shown in Figure 3.

Figure 3: Install "esp8266 by ESP8266 Community"

Step 7: Wait until installation is finished by Boards Manager and click on "Close".

Step 8: Restart the Arduino software. Follow these options: **Tools>>Board>>NodeMCU 1.0(ESP 12E-Module)** as shown in Figure 4.
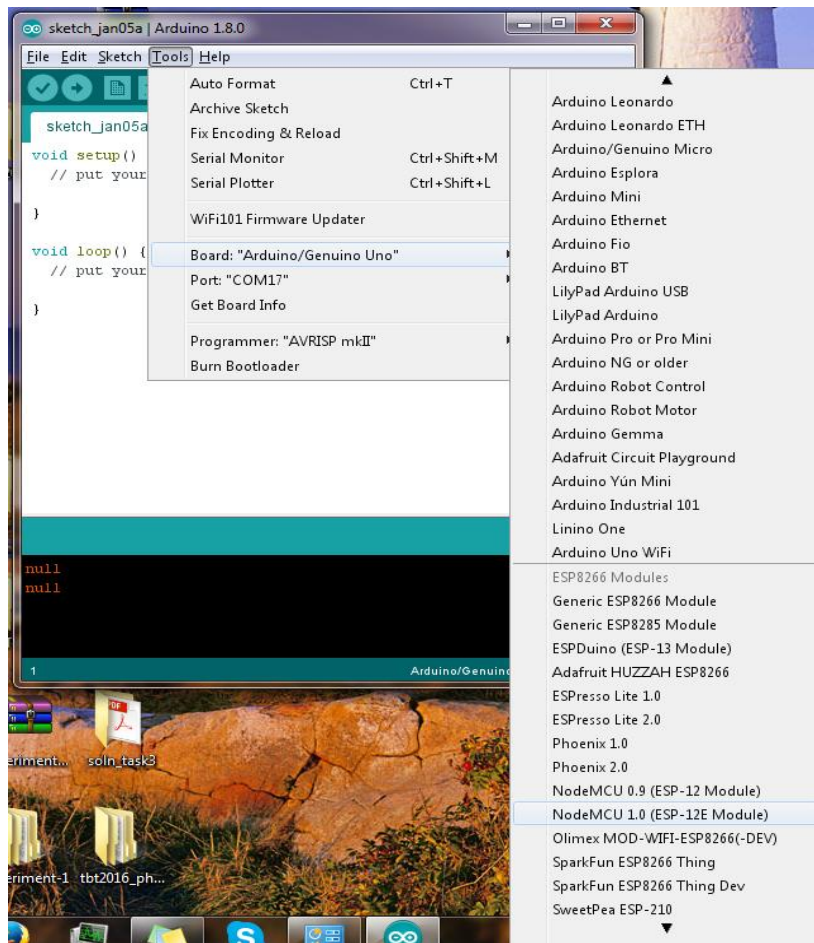
Figure 4: Selecting NodeMCU 1.0 Board.

Step 9: Connect your NodeMCU to your computer using USB cable. Please identify **NodeMCU** from Figure 5**.**
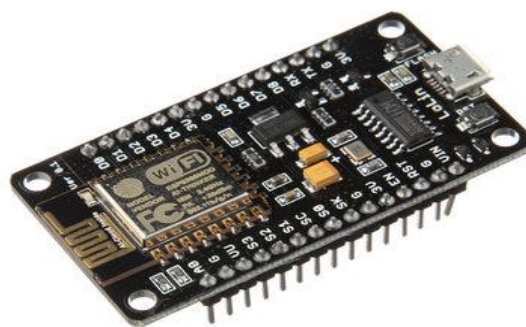

Figure 5: NodeMCU

Step 10: Select the correct "COM" port in Arduino software as shown in Figure 6. Follow these options: **Tools>>Port>>COM**(as per your system)

NOTE: If you are not able to find the COM port, please check your computer's Device Manger and look for "Ports". USB-SERIAL CH340 will be available with correct COM port number.
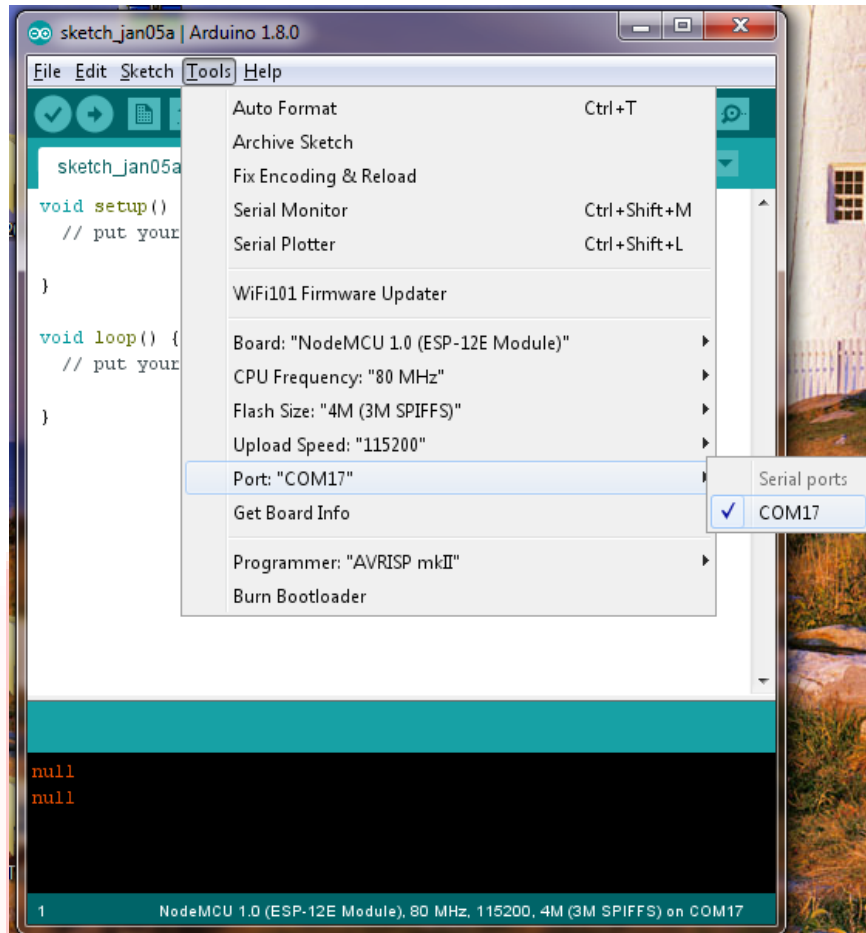
Figure 6: Selecting the correct COM port

Step 11: Follow these options: **File>>Examples>>ESP8266>>Blink.** A window will appear as shown in Figure 7. This is a program for testing your NodeMCU using a RGB LED. Upload this program by clicking on **Upload** button. Please wait until the upload is complete.

NOTE: To write your program for NodeMCU, please select: File>>New. A new window will appear to write your programs. Upload button is given to burn your program on NodeMCU hardware.

```
/*
  ESP8266 Blink by Simon Peter
  Blink the blue LED on the ESP-01 module
  This example code is in the public domain

  The blue LED on the ESP-01 module is connected to GPIO1
  (which is also the TXD pin; so we cannot use Serial.print() at the same time)

  Note that this sketch uses LED_BUILTIN to find the pin with the internal LED
*/

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

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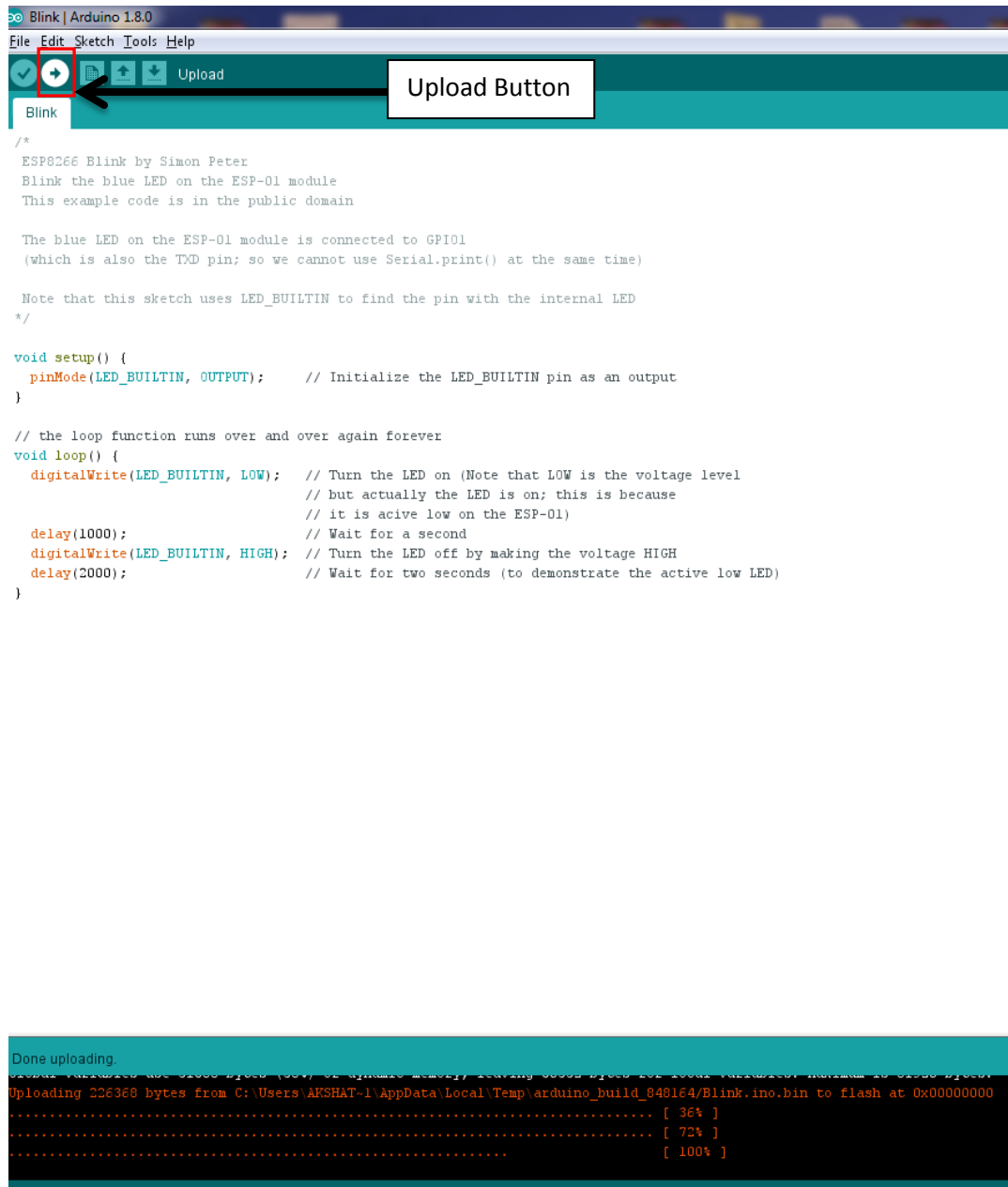

**Figure 7: Uploading the Blink program in NodeMCU.**

Step 12: Take the RGB led given in the kit. Connect **two pins** of RGB led to NodeMCU pins. Connect the RGB led according to Table 1. Figure 8 and Figure 9 is given to identify the pin configurations of RGB led and NodeMCU.

Pins	RGB led pins	NodeMCU pins(pin name written on NodeMCU)
1	Common Cathode	G
2	Any one pin from R,G,B	D0

Table 1: NodeMCU and RGB led connections.

NOTE: Longest pin of RGB led is Common Cathode. Pin names is written over NodeMCU. Do not disconnect your NodeMCU from your computer.
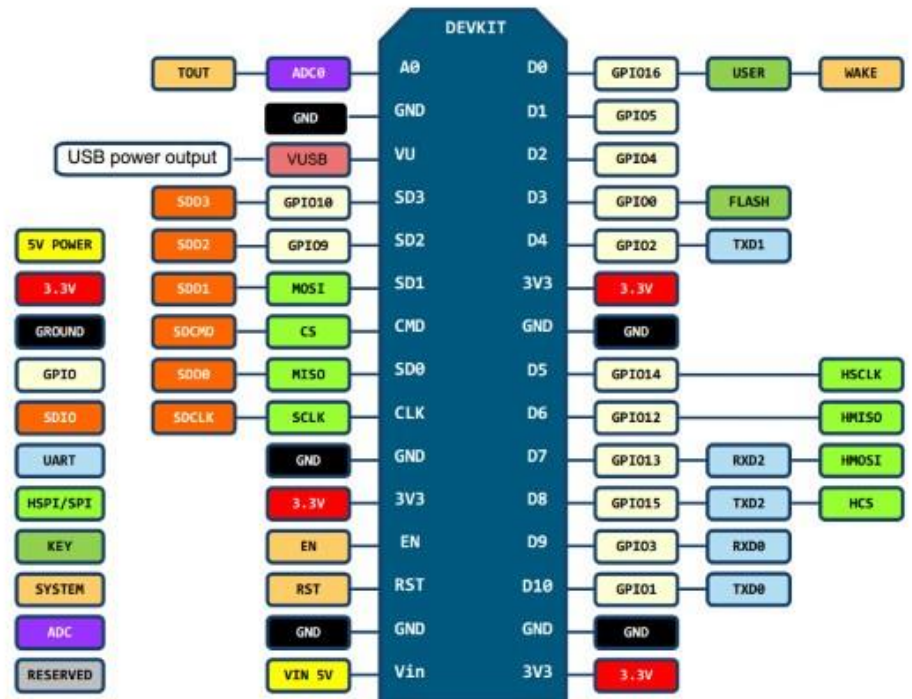
PIN DEFINITION
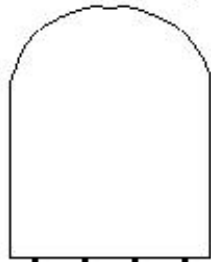
Common Cathode (-)

B   G   R

Common Cathode (-)

D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.

Figure 8: RGB led   **Figure9: NodeMCU Pin Configuration**

Step-13: Take Servo Motors and connect its 3 pins to Node_MCU. Please identify the pins of servo as **VCC, GND and Signal**. Also, upload a program on Node_MCU for generating appropriate PWM on the **Signal** pin of the Servo. You can follow this link to test your Servo motors. The link given is specific to Arduino Uno board, you might have to make some changes in order to control the Servo using Node_MCU.

After following the above steps the Node_MCU, RGB led and Servo Motors will be tested. You will see the components functioning; this will ensure that your components are working properly.

--END--