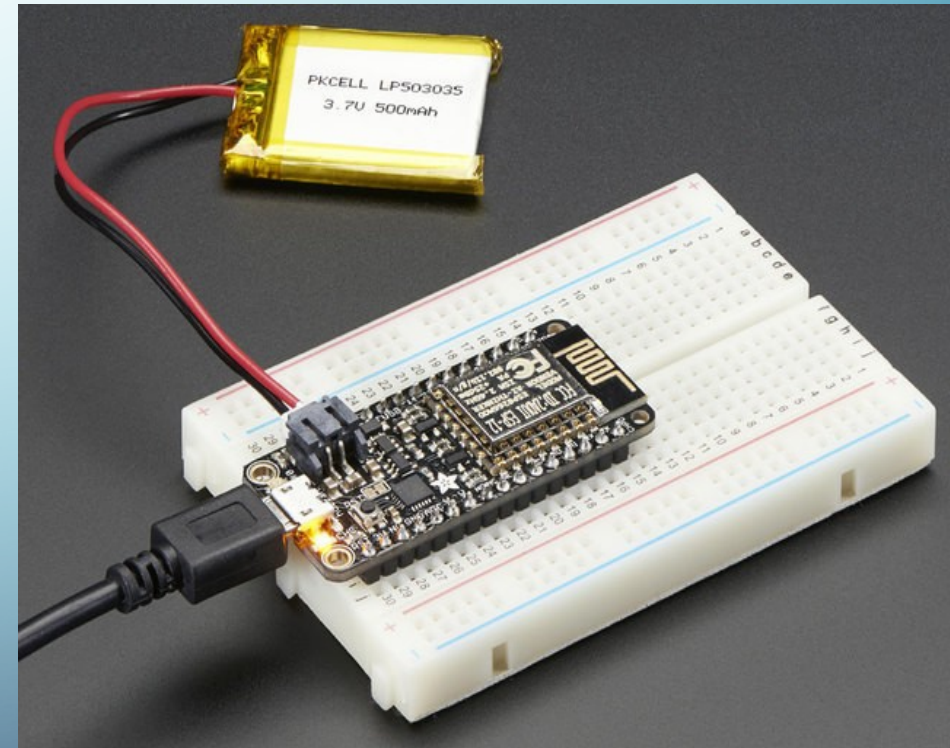


# NETWORKING WITH EMBEDDED SYSTEMS

- Adafruit Feather HUZZAH ESP8266
- Used to achieve wireless communication with your robot
- Standard Micro USB cable
- Comes programmed with NodeMCU's Lua, but you can use the Arduino IDE as well.



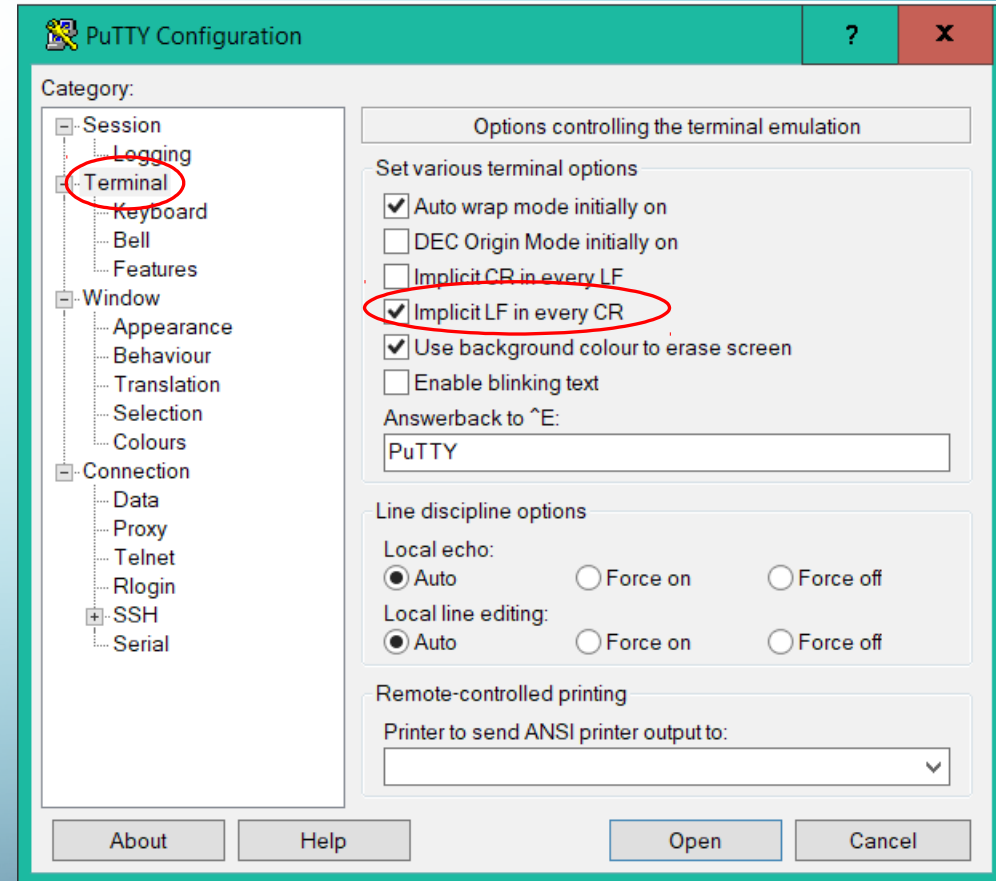
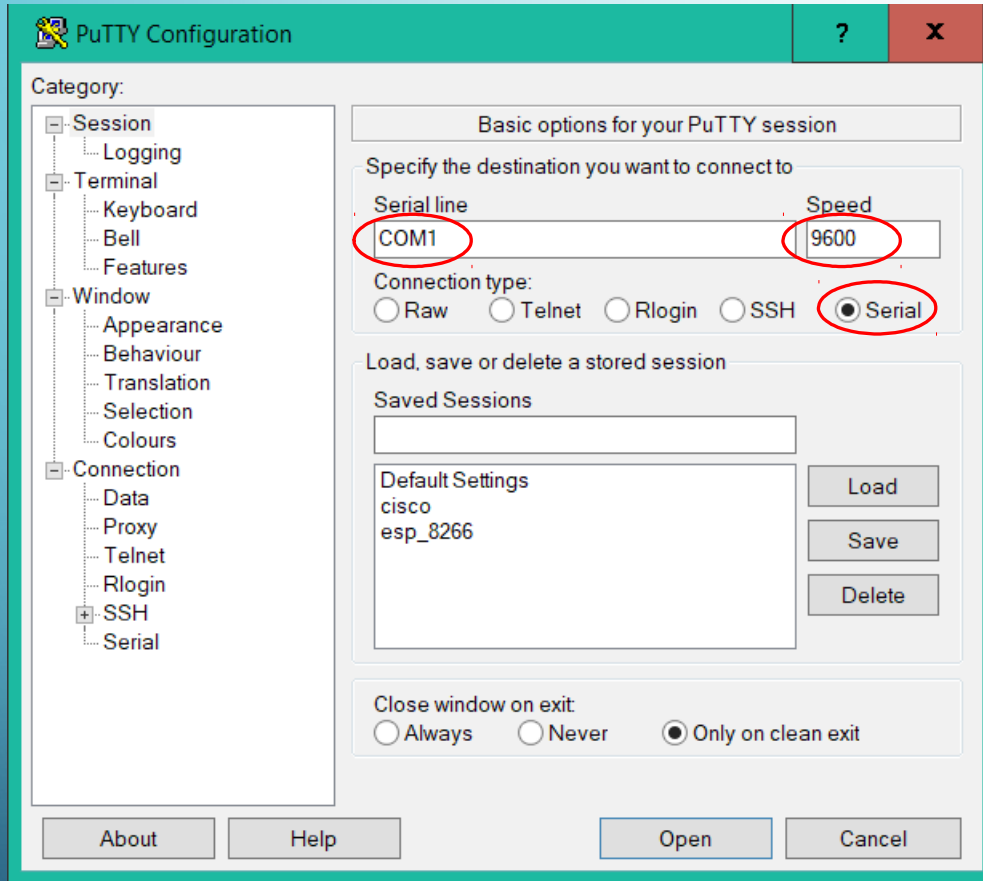
# NODE-MCU LUA-DOWNLOAD DRIVER SOFTWARE

- Use this link to download and install the necessary software so that your COM/Serial port will appear correctly
- <https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx>

# NODE-MCU LUA-PUTTY SETUP

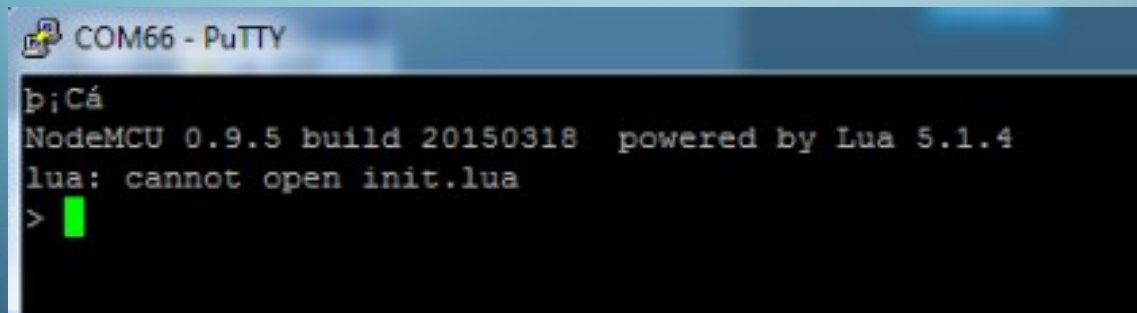
- If you haven't already downloaded Putty, you can do so here:
- <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
- Connect the chip to your computer with the USB Cable
- Use putty to connect to the COM or Serial port your cable is connected to





# NODE-MCU LUA-CONNECTING WITH PUTTY

- Once you're connected, press the reset button on the chip.
- Red and blue lights should flash
- Something similar to this should be printed:

A screenshot of a PuTTY terminal window titled 'COM66 - PuTTY'. The terminal displays the following text: 'p;Cá', 'NodeMCU 0.9.5 build 20150318 powered by Lua 5.1.4', 'lua: cannot open init.lua', and a prompt '>' followed by a green cursor bar.

```
COM66 - PuTTY
p;Cá
NodeMCU 0.9.5 build 20150318 powered by Lua 5.1.4
lua: cannot open init.lua
>
```

# NODE-MCU LUA-LED BLINK CODE

- You can now copy and paste code or type commands to send them to the chip.
- This example code controls pin output to cause an LED to blink.
- You could write similar code to control things connected to other pins.

```
1.  while 1 do
2.      gpio.write(3, gpio.HIGH)
3.      tmr.delay(1000000)    -- wait 1,000,000 us = 1 second
4.      gpio.write(3, gpio.LOW)
5.      tmr.delay(1000000)    -- wait 1,000,000 us = 1 second
6.  end
```

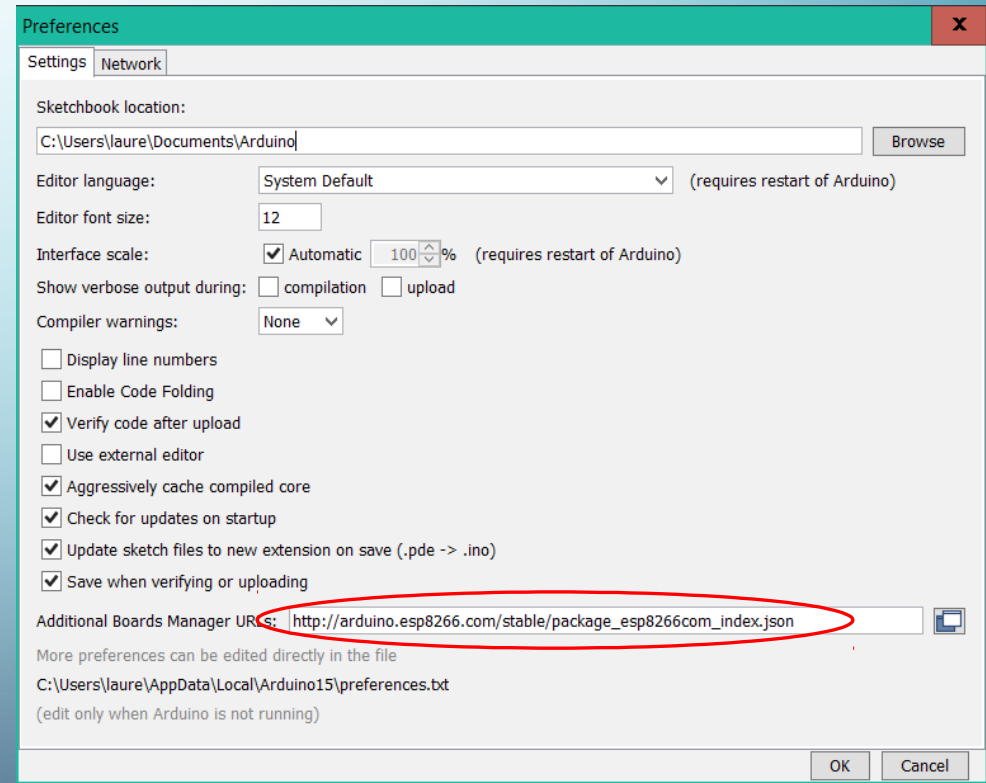
# ARDUINO IDE-DOWNLOAD SOFTWARE

- If you're more familiar with Arduino or C coding, you can program the chip to use that language
- This will erase the NodeMCU Lua programming from the chip. You can redownload it here:
- <https://github.com/nodemcu/nodemcu-flasher>
- If you want to use the Arduino IDE and don't have it, you can download it here:
- <https://www.arduino.cc/en/Main/Software>



# ARDUINO IDE-ADDING BOARD PACKAGE

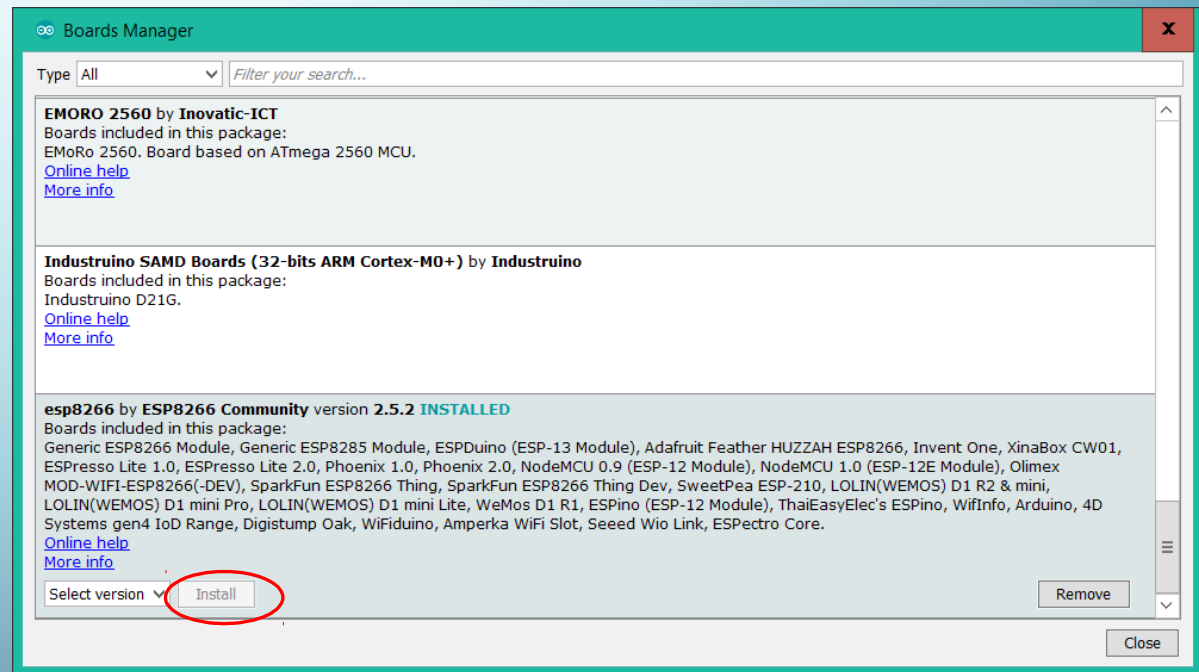
- In the Arduino IDE, go to Preferences under the File tab
- Paste this URL into the Additional Boards Manager URLs slot:
- [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)





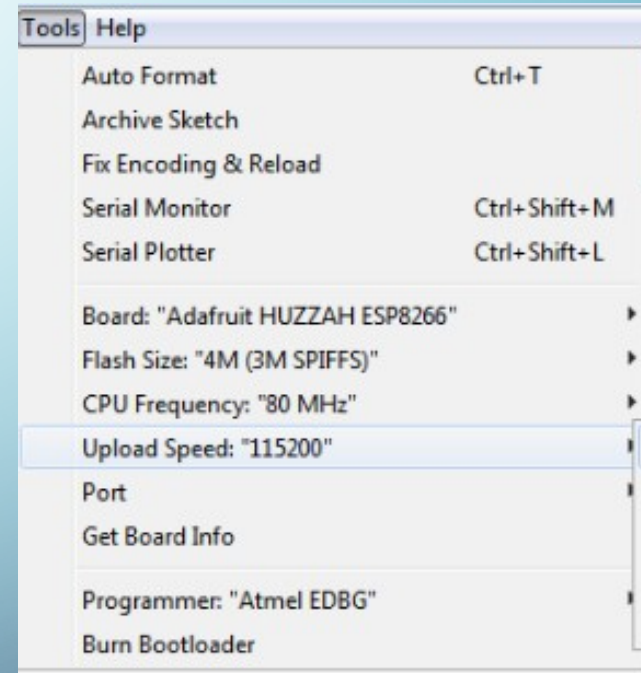
# ARDUINO IDE-INSTALLING THE BOARD PACKAGE

- Click on Board:  
under the Tools tab,  
and choose Board  
Manager
- Click on the ESP8266  
box, and then click  
Install



# ARDUINO IDE-BOARD SETTING

- Go to the tools tab, and choose Adafruit Feather HUZZAH ESP8266
- Set the CPU Frequency to 80 MHz
- Set the flash size to 4M(3M Spiffs)
- Set the upload speed to 115200
- Select the COM port that connects to your USB



# ARDUINO IDE-SOCKET SERVER

- At this point, you can write code and compile and upload it to the board while it's connected via USB just like any Arduino code
- To communicate wirelessly, you need a socket server
- Use code from one of a number of sources to create one

```
#include "ESP8266WiFi.h"

const char* ssid = "MERCURY";
const char* password = "";

WiFiServer wifiServer(80);

void setup() {

  Serial.begin(115200);

  delay(1000);

  WiFi.begin(ssid, password);

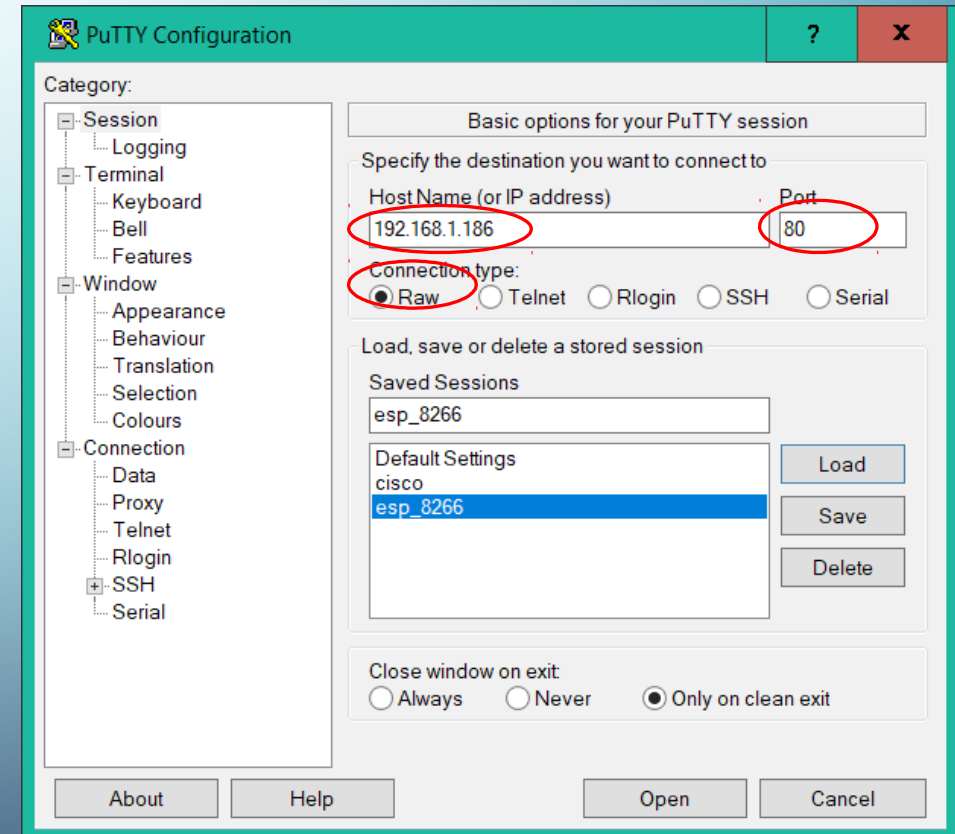
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting..");
  }
  pinMode(0, OUTPUT);
  Serial.print("Connected to WiFi. IP:");
  Serial.println(WiFi.localIP());

  wifiServer.begin();
}
```



# ARDUINO IDE-PUTTY SETTINGS

- Once the Webserver code is uploaded to the chip, go to the Arduino IDE's serial monitor under tools.
- The IP Address of the chip should be printed in the serial monitor once connected.
- Use this address to connect to the chip with Putty, on port 80



# ARDUINO IDE-LED CONTROL EXAMPLE

- Any commands entered in the Putty window will be sent to the chip and processed within the code
- Add code to process input data and control pin output to make the chip respond to certain inputs
- Here, sending 1 in Putty turns the LED on, 0 turns it off

```
void loop() {  
  
    WiFiClient client = wifiServer.available();  
  
    if (client) {  
  
        while (client.connected()) {  
  
            while (client.available() > 0) {  
                char c = client.read();  
                Serial.write(c);  
  
                if (c == '1')  
                {  
                    Serial.write(" on");  
                    digitalWrite(0, LOW);  
                }  
                if (c == '0')  
                {  
                    Serial.write(" off");  
                    digitalWrite(0, HIGH);  
                }  
            }  
  
            delay(10);  
        }  
  
        client.stop();  
        Serial.println("Client disconnected");  
  
    }  
}
```

# REFERENCES

- This code can be generalized to send information to any of the chip's pins, and control any part of a robot.
- For more information:
- <https://techtutorialsx.com/2018/06/02/esp8266-arduino-socket-server/>
- <https://learn.adafruit.com/adafruit-feather-huzzah-esp8266/overview>