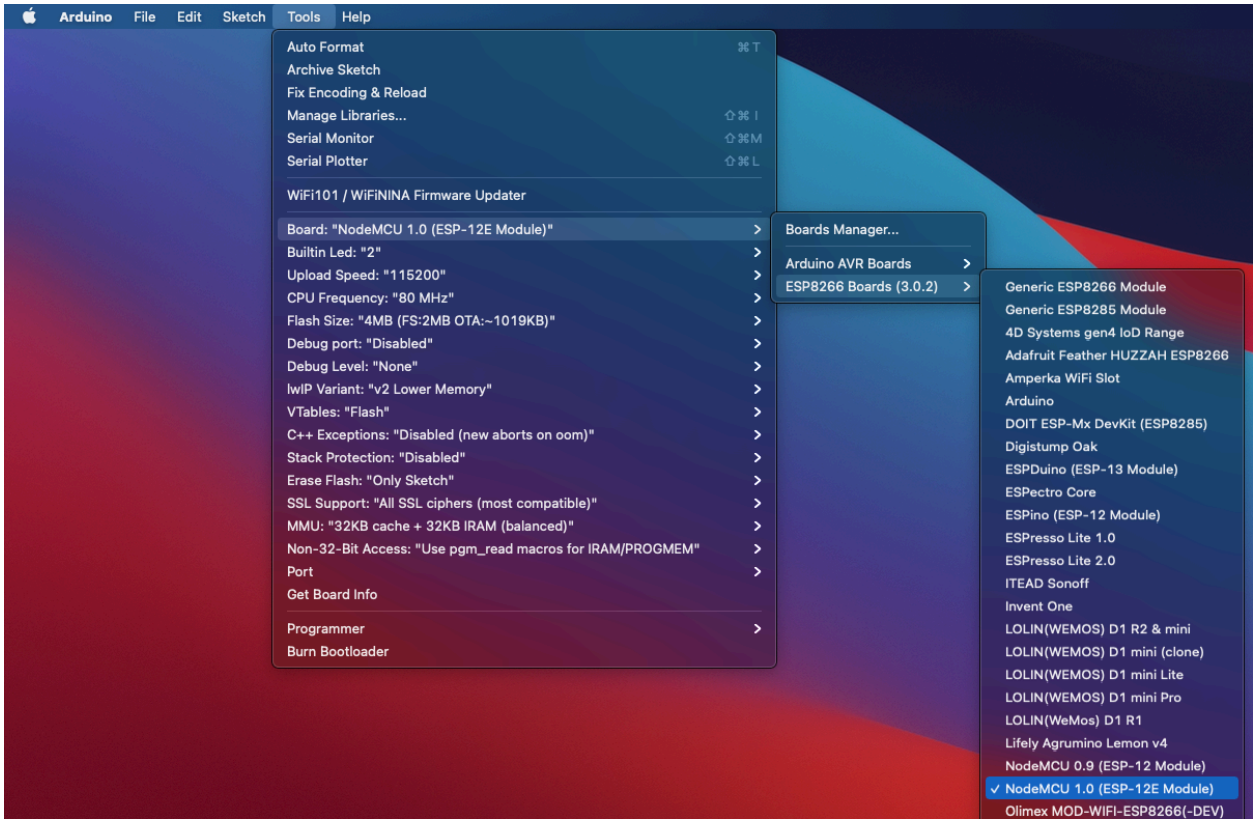
Adafruit IO with NodeMCU

If you are using your own computer, start with Part A. If you are using a lab computer, you can continue with Part B. If you are unable to connect to AirPennNet-Device in Part B, check that you are using a NodeMCU that has been registered by the staff. It should have a green label on the bottom with a number.

If you are working in another location that does not have AirPennNet-Device available, then you will need to use either a phone hotspot or your own internet connection.

Part A: NodeMCU ESP8266 USB Driver Installation

1. Follow all these steps on this tutorial up to "**Setup ESP8266 Support**".
2. At the **Setup ESP8266 Support** step, go to Tools > Board > ESP8266 Boards (3.0.2) > NodeMCU 1.0 (ESP-12E Module)



3. For the port selection, on a Windows, you should select the matching COM port. **On a Mac, you should see something similar to "SLAB_USBtoUART". If you do not, then reinstall the USB driver.**

Part B: Hotspot Connection

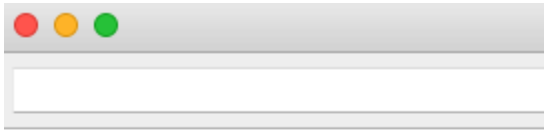
You will need to connect your NodeMCU to AirPennNet-Device. If you are working from another location, replace the SSID and pw accordingly. If AirPennNet-Device does not work, a phone hotspot will be required.

1. Below is the information for AirPennNet-Device:
   SSID: AirPennNet-Device
   Pw: penn1740wifi
2. We're going to check if the nodeMCU can successfully connect to the internet. On the Arduino IDE, go to **File** > **Example** > **esp8266webserver** > **helloserver**. Replace the ssid and the password with the information in step 2 above.

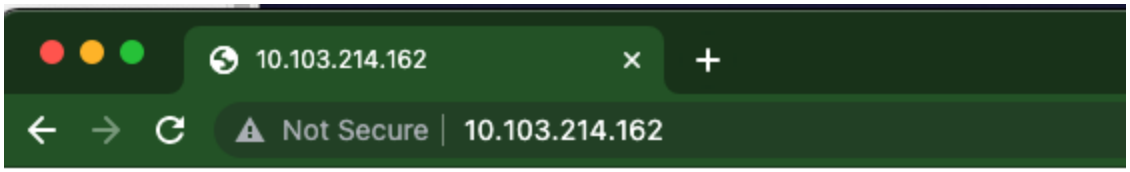
```
HelloServer §
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>

#ifndef STASSID
#define STASSID "AirPennNet-Device"
#define STAPSK  "penn1740wifi"
#endif
```

3. Upload the code, open the serial monitor (make sure the baud rate is 115200), and then go to the IP address in your web browser.



```
{llSSISlSI   S lS "ISS S SSI BS cSSnoS$noS
.........................
Connected to AirPennNet-Device
IP address: 10.103.214.162
MDNS responder started
HTTP server started
```

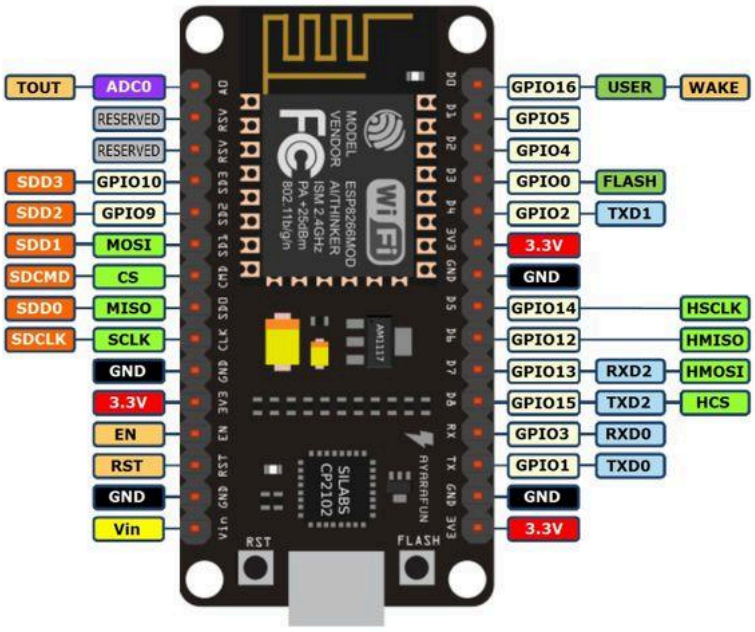



```
hello from esp8266!
```

The NodeMCU should have already been registered by the Detkin staff. If you are using AirPennNet-Device as your hotspot and the connection was unsuccessful, talk to a member of the Detkin staff.

NodeMCU Pinouts

Note the correspondence between **Pin Name on the Board** and **Pin Number in Arduino IDE** in the diagram below. The **Pin Number in Arduino IDE** is the pin number that you will use in the Arduino IDE.

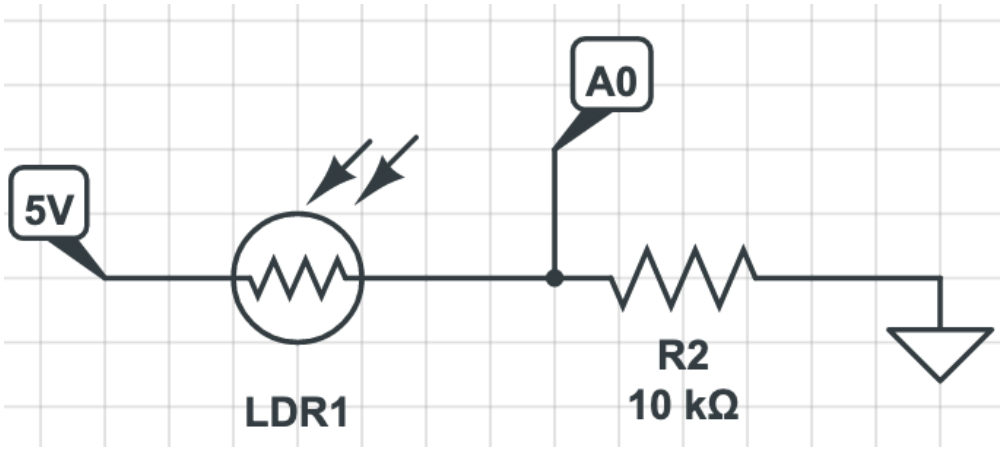


Pin Name on the Board	Function	Pin Number in Arduino IDE	Alias Name in Arduino IDE
D3	GPIO 0	0	D3
TX	GPIO 1	1	D10
D4	GPIO 2	2	D4
RX	GPIO 3	3	D9
D2	GPIO 4	4	D2
D1	GPIO 5	5	D1
D6	GPIO 12	12	D6
D7	GPIO 13	13	D7
D5	GPIO 14	14	D5
D8	GPIO 15	15	D8
D0	GPIO 16	16	D0, LED_BUILTIN
A0	ADC0	A0	analog_ip

Part C: Starter Lab

Schematic

Procure a photoresistor and 10kohm resistor and wire your circuit according to the schematic below.



Arduino IDE Setup

1. Open the Arduino IDE.
2. On your computer desktop and not in the Arduino IDE, navigate to the folder where you downloaded the github repository. Open the *adaruit_io_example* folder and then click on *adafruit_io_example.ino*. Compile and upload the code in the repo onto the NodeMCU. Note that this is just a sample code and you should alter the code to fit your application.

Adafruit IO

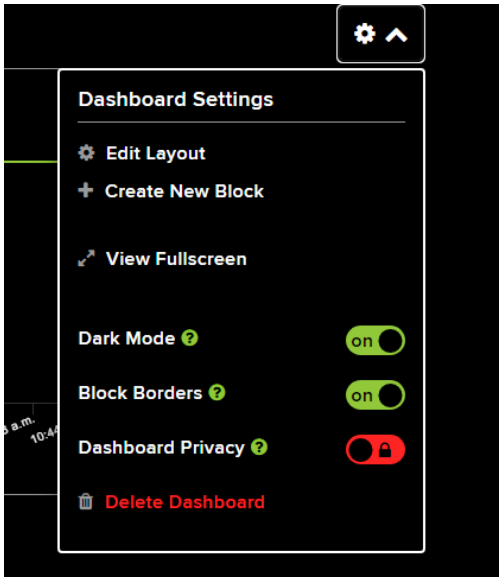
1. Create an account at io.adafruit.
2. Click **My Key**

3. Going back to the Arduino IDE, modify **config.h** so it has the appropriate username and key.

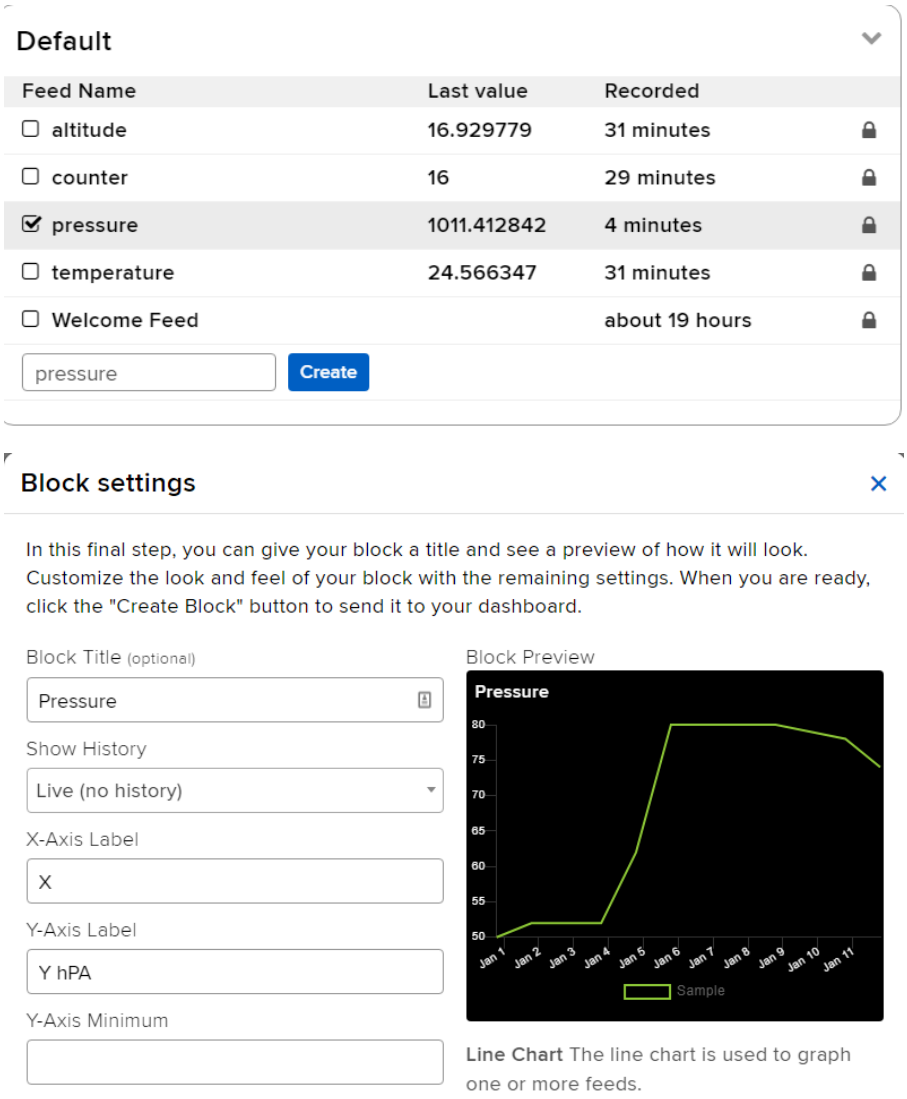
```
e218_lab7_adafruitio | config.h
/************************ Adafruit IO Config *************************

// visit io.adafruit.com if you need to create an account,
// or if you need your Adafruit IO key.
#define IO_USERNAME "your_username"
#define IO_KEY "your_key"
```

4. Back to the io.adafruit.com website, click on **Dashboard** > **Welcome Dashboard** (or a new one that you made). Under the gear icon on the right hand side, select **Create New Block** > **Line Chart.**

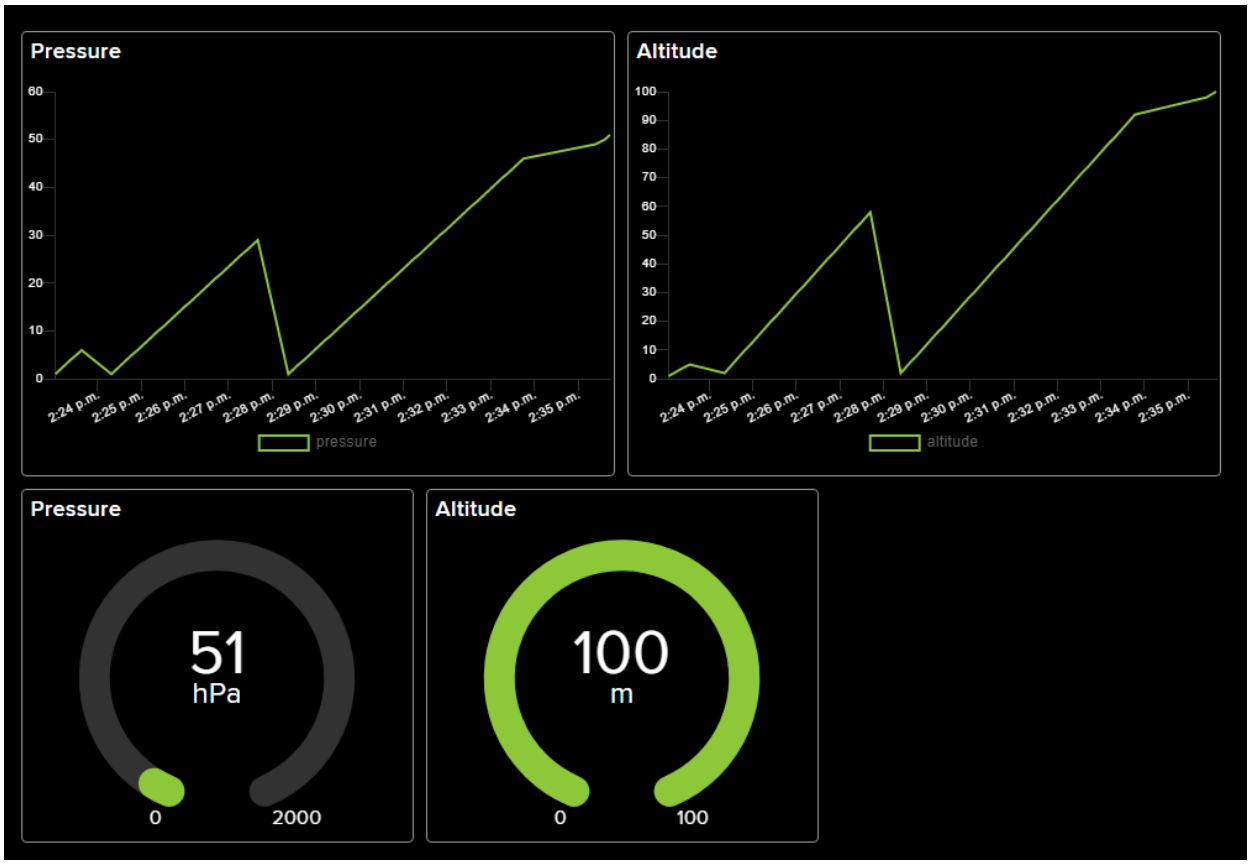


5. Create a new feed called "pressure" (case sensitive). Select it and click "Next step >"



6. Create another Gauge block for the pressure feed.
7. Repeat steps 8-10 for a feed called "altitude" (case sensitive).
8. Shine a light on the photocell and then cover the sensor with your hand. Observe the values that you see on your dashboard.

9. Your final dashboard should look similar to the diagram below but with different values.



10. If the data needs to be exported, follow [these instructions]. You will not need to do this for this lab. The measurements will be recorded manually.