

Rick Suel  
Paul Goodjohn



# IoT with ESP8266

Week of Learning 2018



# Class Pre-Work

1. Install latest Arduino software from:

<https://www.arduino.cc/en/Main/Software>

2. Install the Silicon Labs virtual com port (VCP) driver for your laptop's operating system. This driver will allow your computer to recognize the ESP8266 device as a "com port" when you plug it in via USB.

<https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>

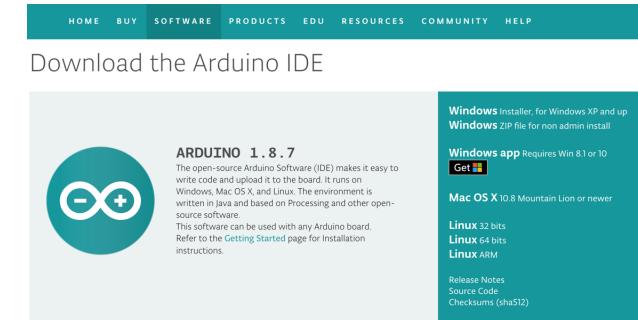
3. Create an account on Adafruit IO. This is the cloud platform we will be using:

<https://io.adafruit.com/>

4. Create an account on IFTTT. We will be using this platform to create events based on your device:

<https://ifttt.com/>

Get these links here: <https://github.com/rdsuel/esp-class>

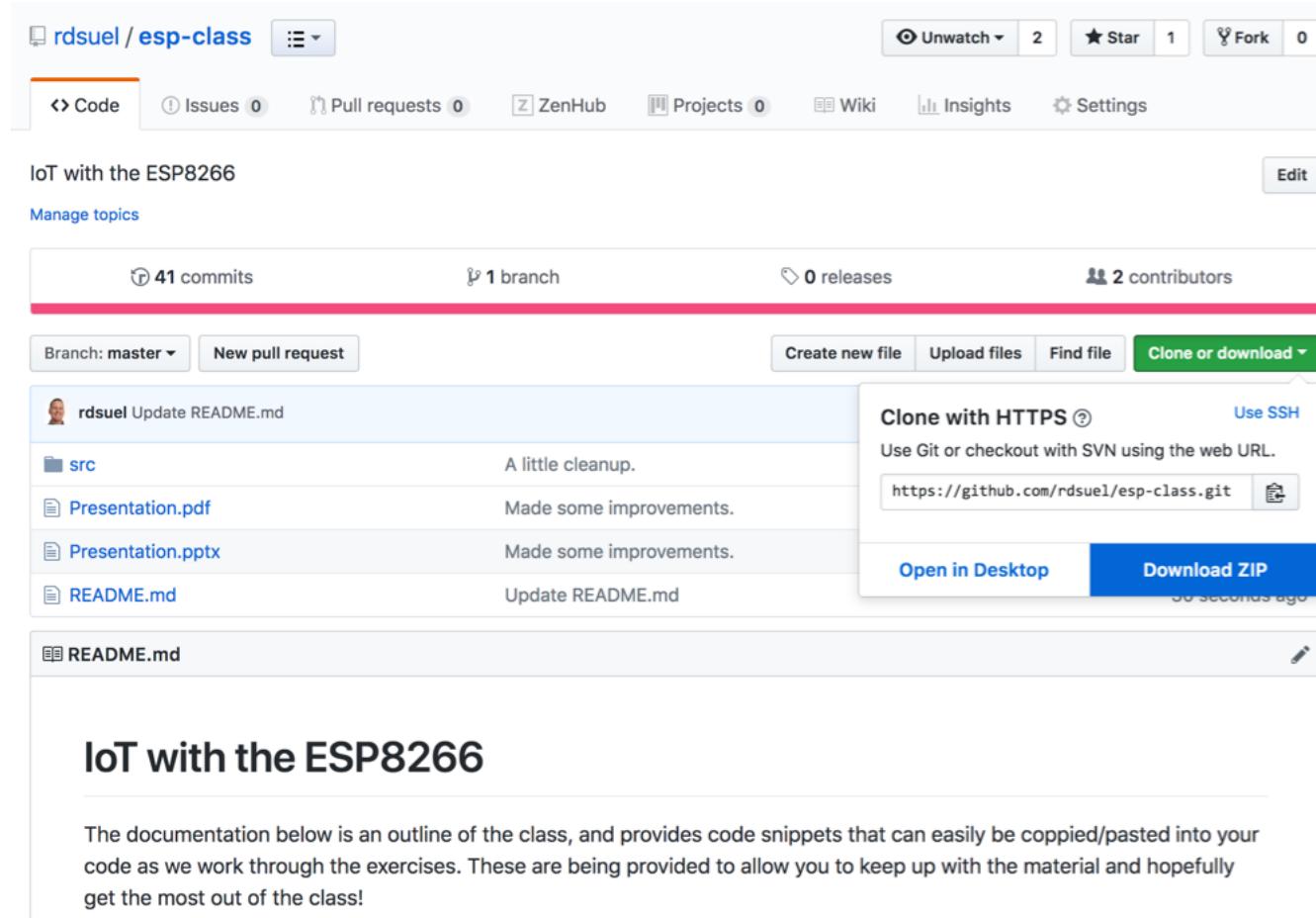


The screenshot shows the Arduino Software (IDE) download page. At the top, there is a navigation bar with links for HOME, BUY, SOFTWARE, PRODUCTS, EDU, RESOURCES, COMMUNITY, and HELP. Below the navigation bar, a section titled "Download the Arduino IDE" is shown. It features a large button labeled "Get IDE" with a "Windows app" icon. To the left of the button is the Arduino logo, which is a teal circle containing a white infinity symbol and a plus sign. To the right of the button, there is descriptive text about the software, including its version (1.8.7), supported platforms (Windows, Mac OS X, and Linux), and download links for Windows (Installer and ZIP), Mac OS X (ZIP), and Linux (32-bit, 64-bit, ARM). There are also links for Release Notes, Source Code, and Checksums (sha512).



# Class Materials

The source code and presentation for this course can be found here:  
<https://github.com/rdsuel/esp-class>



A screenshot of a GitHub repository page. The repository name is `rdsuel / esp-class`. The page shows basic statistics: 41 commits, 1 branch, 0 releases, and 2 contributors. A pull request button and a clone/download button are visible. The repository contains files like `src`, `Presentation.pdf`, `Presentation.pptx`, and `README.md`. The `README.md` file is open, displaying the following content:

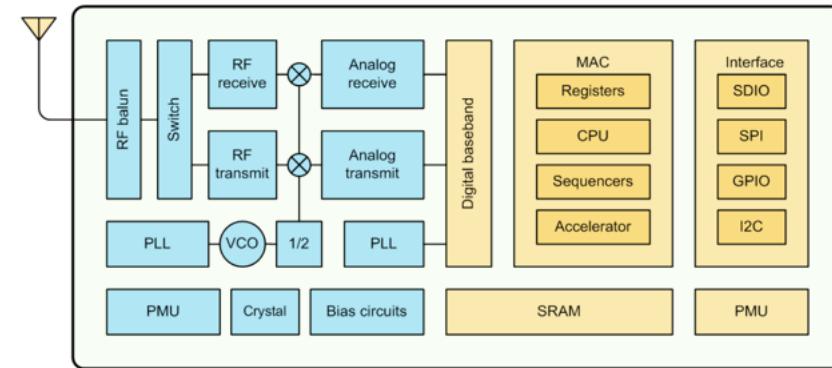
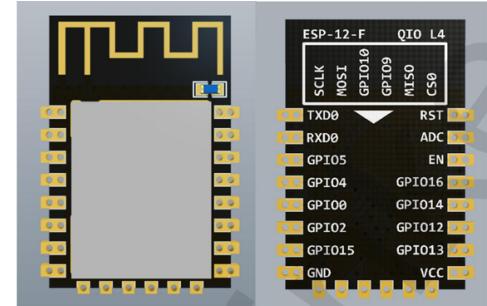
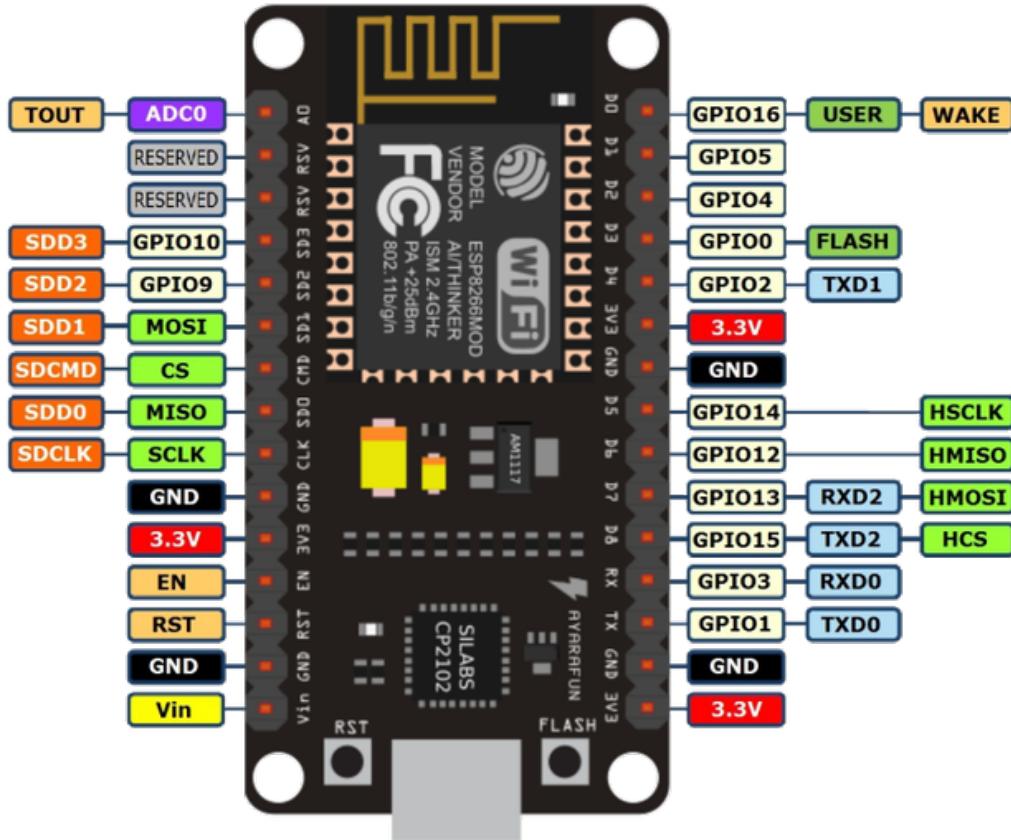
## IoT with the ESP8266

The documentation below is an outline of the class, and provides code snippets that can easily be copied/pasted into your code as we work through the exercises. These are being provided to allow you to keep up with the material and hopefully get the most out of the class!

# The ESP8266

<http://esp8266.net/>

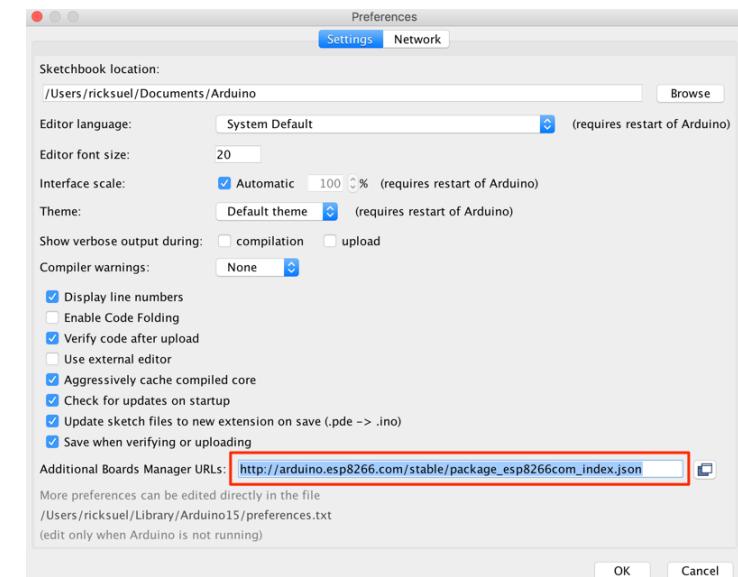
<http://esp8266.github.io/Arduinoversions/2.0.0/doc/reference.html>



# Setup

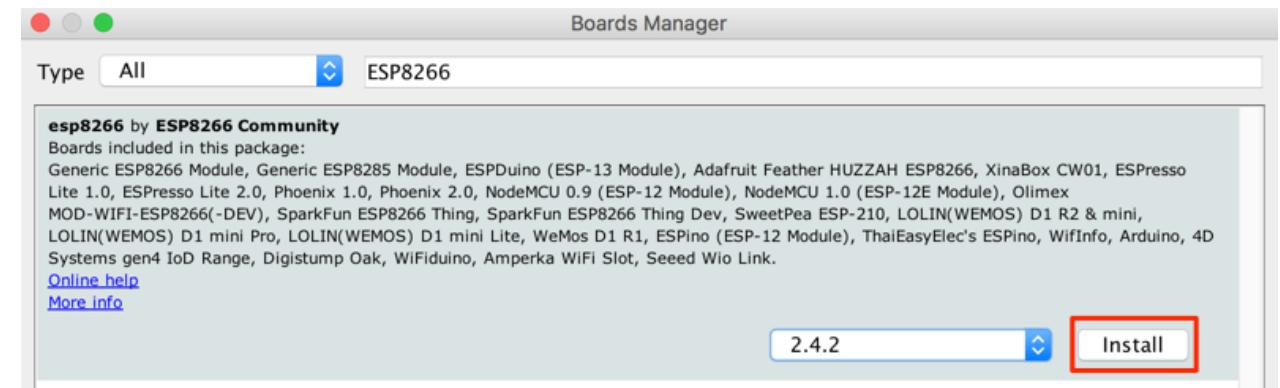
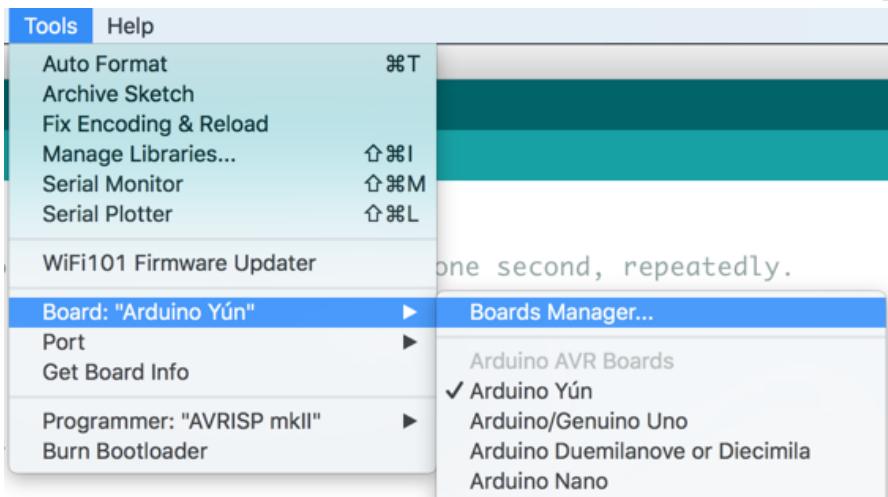
- Add Boards Manager URL:

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



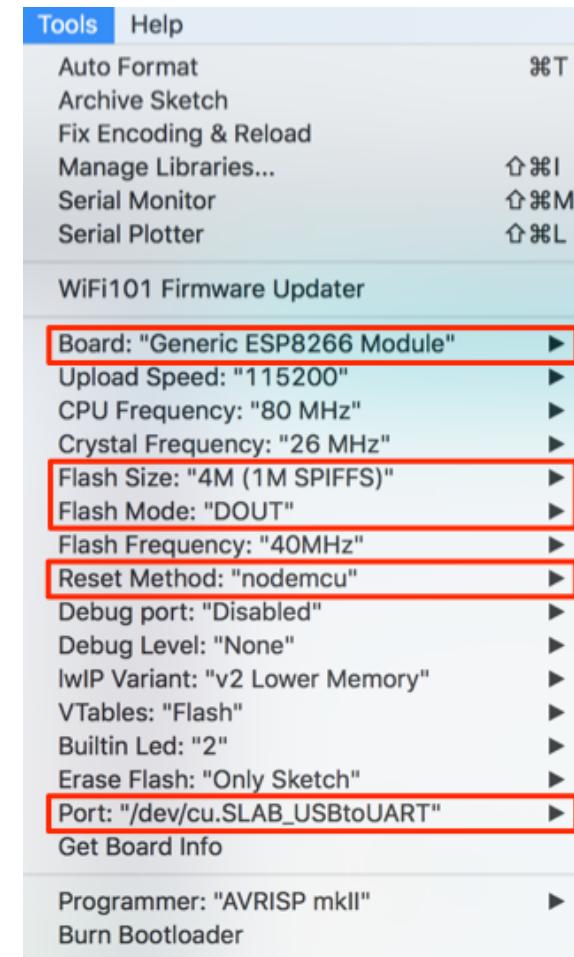
- Install the ESP8266 boards package:

Tools → Board → Boards Manager



# Setup (Continued)

- Configure the connection parameters for the device:
  - Board: “Generic ESP8266 Module”
  - Flash Size: 4M (1M SPIFFS)
  - Flash Mode: “DOUT”
  - Reset Method: “nodemcu”
  - Port: choose the one for your device



# Install Arduino Libraries

We will need several libraries installed for this class:

- Display Library (“ssd1306”):

**ESP8266 and ESP32 Oled Driver for SSD1306 display** by Daniel Eichhorn, Fabrice Weinberg Version 4.0.0 **INSTALLED**  
A I2C display driver for SSD1306 oled displays connected to an ESP8266 or ESP32 A I2C display driver for SSD1306 oled displays connected to an ESP8266 or ESP32  
[More info](#)

Select version

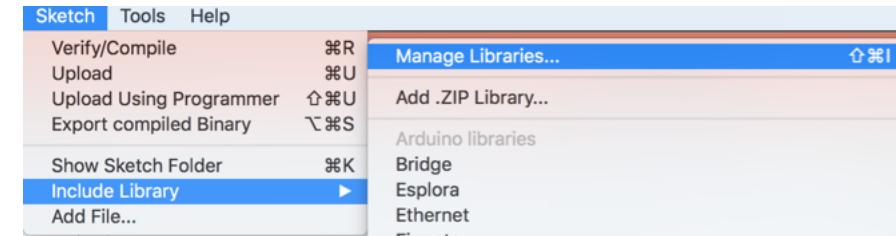
- Temp Sensor Libraries (“DHT” and “adafruit unified sensor”):

**DHT sensor library** by Adafruit Version 1.3.0 **INSTALLED**  
Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors  
[More info](#)

Select version

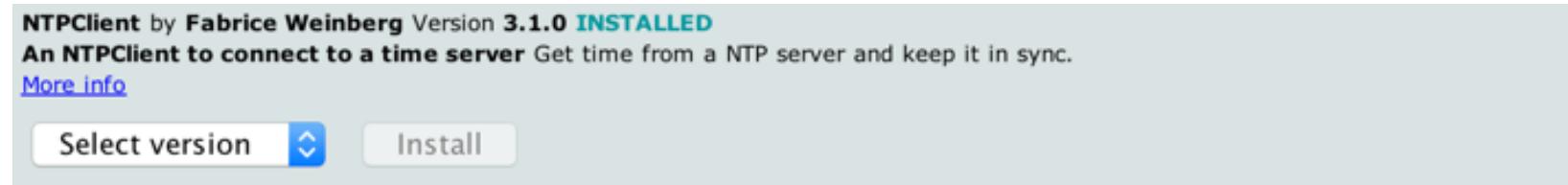
**Adafruit Unified Sensor** by Adafruit Version 1.0.2 **INSTALLED**  
Required for all Adafruit Unified Sensor based libraries. A unified sensor abstraction layer used by many Adafruit sensor libraries.  
[More info](#)

Select version

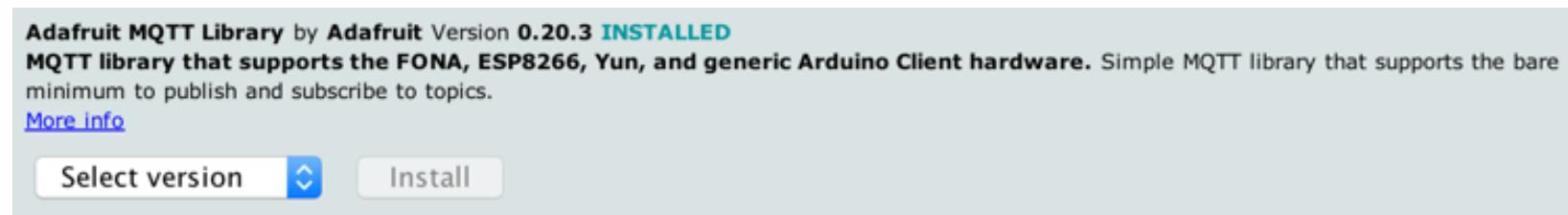


# Install Arduino Libraries (Continued)

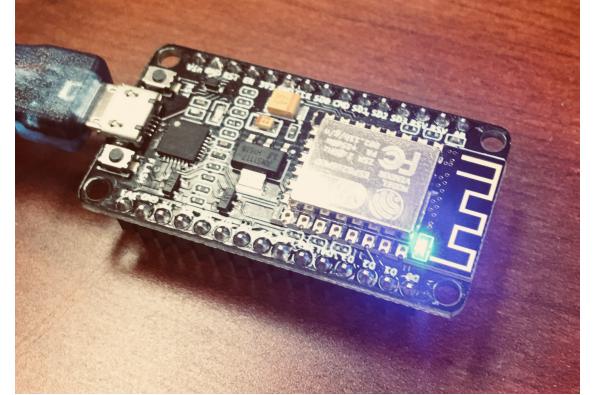
- NTP Client Library for getting time over the internet:



- Adafruit MQTT Library for sending data to <https://io.adafruit.com>



# Exercise 1: Blinky



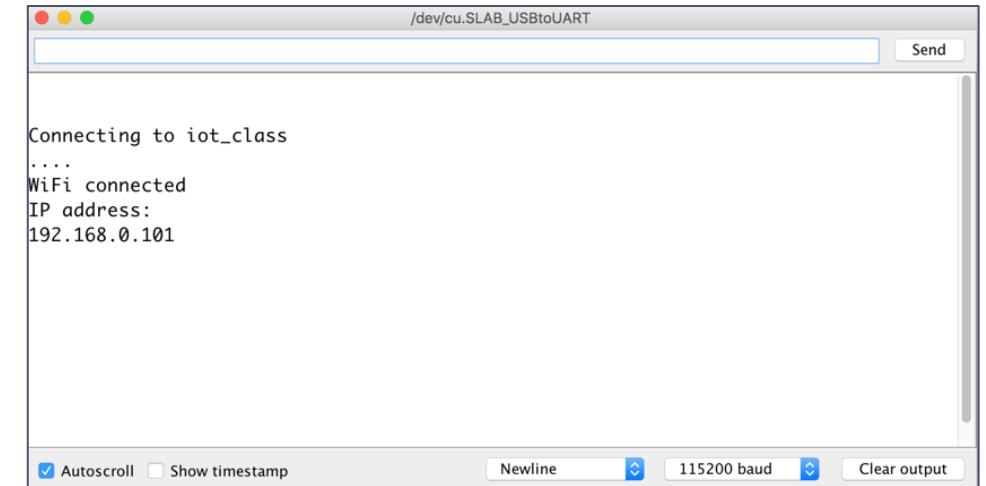
## Goal:

Confirm you are able to load applications to your device by blinking the on-board LED.

## Steps:

1. Load the ESP8266 → Blink example.
2. Switch over to the “scheduler-based” blink architecture. This will serve as the platform for the future exercises.

# Exercise 2: Connect to Wifi



## Goal:

Connect your device to the "iot\_class" network and print your IP address to the serial window.

## Steps:

1. Update the "student.ino" template file so that it connects to the "iot\_class" network. The password is "password", so obviously it is very secure!
2. Once the device is connected, print the IP address to the serial window.
3. Congratulations, your esp8266 is now online. We'll do more with this connection shortly!

# Exercise 3: OLED Display

## Goal:

Add support for the OLED display and display the current by pulling it from the internet using an NTP server (time.nist.gov). Update the time every second.

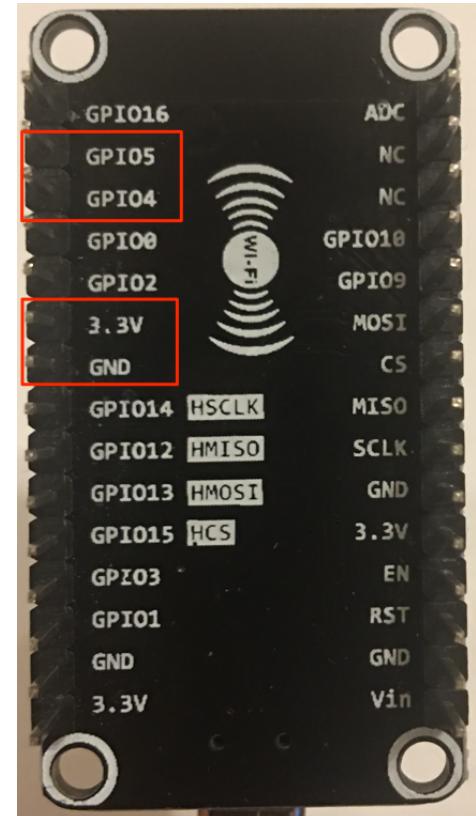


## Steps:

1. Wire the ESP8266 to the display as defined here:

| ESP8266 | Display |
|---------|---------|
| GPIO5   | SDA     |
| GPIO4   | SCK     |
| 3.3V    | VDD     |
| GND     | GND     |

2. Periodically read the time from the NTP server and write it to the display.



# Exercise 4: DHT22 Temp/Humidity Sensor



## Goal:

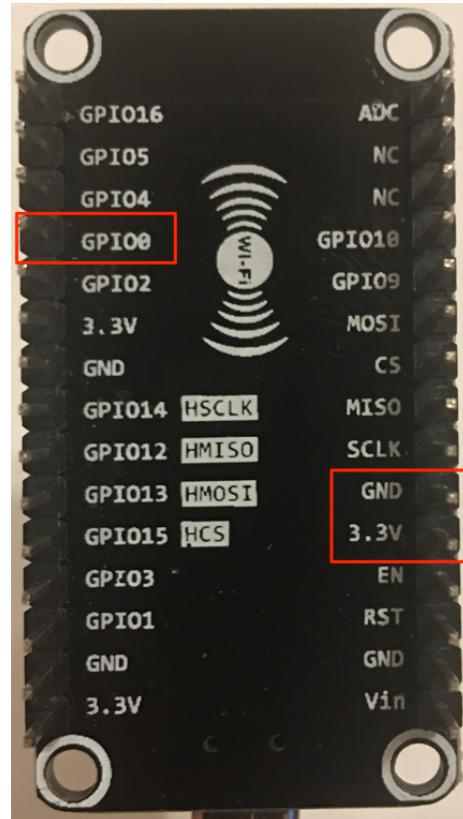
Add support for the DHT22 Temp/Humidity sensor. Display the temperature and humidity along with the current time on the OLED display.

## Steps:

1. Wire the ESP8266 to the sensor as defined here:

| ESP8266 | Sensor |
|---------|--------|
| GPIO0   | DAT    |
| 3.3V    | VCC    |
| GND     | GND    |

2. Periodically read the temperature and humidity, and write them to the display.



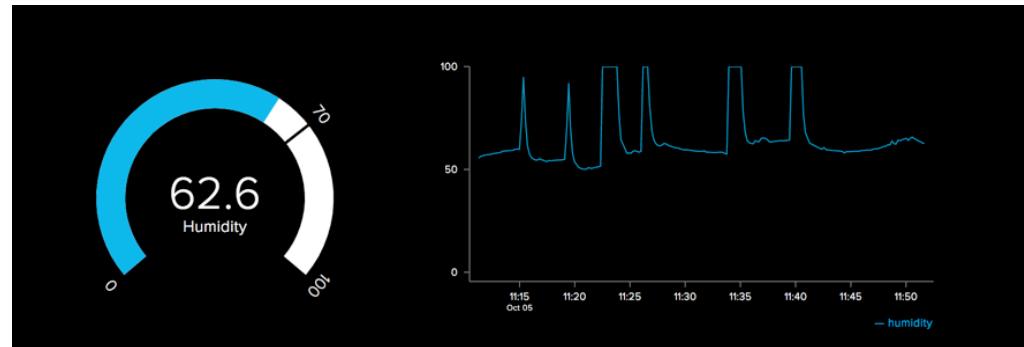
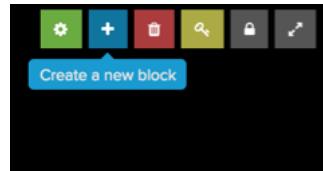
# Exercise 5: Adafruit IO Cloud Platform

## Goal:

Use MQTT to publish temperature and humidity values to the Adafruit IO platform, and create a GUI for displaying them.

## Steps:

1. Log into <https://io.adafruit.com>.
2. Create a feed called “temperature”.
3. Create a feed called “humidity”.
4. Create a dashboard called “IoT Class”.
5. Add blocks to the dashboard for temp/humidity.
6. Add MQTT/adafruit support to your sketch.
7. Compile and upload to your device.



### Block settings

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title (optional)

Gauge Min Value

Gauge Max Value

Gauge Width

Gauge Label

Low Warning Value

High Warning Value

Optional: If no low warning value is given, the gauge will only change color when the value is out of bounds.

Test Value

Published Value

Optional: If no high warning value is given, the gauge will only change color when the value is out of bounds.

Block Preview  
Humidity (%)

Value

Optional: If no low warning value is given, the gauge will only change color when the value is out of bounds.

Test Value

Published Value

Optional: If no high warning value is given, the gauge will only change color when the value is out of bounds.

### Block settings

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title (optional)

Show History

X-Axis Label

Y-Axis Label

Y-Axis Minimum

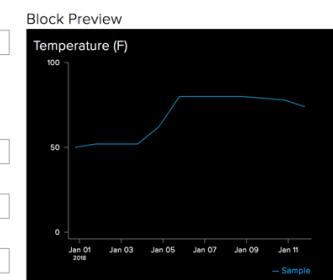
Leave blank to automatically detect.

Y-Axis Maximum

Leave blank to automatically detect.

Decimal Places

Number of decimal places to display, defaults to 4.



Line Chart The line chart is used to graph one or more feeds.

# Exercise 6: IFTTT Humidity Alerts

## Goal:

Create an IFTTT “applet” to send you a text message (or email) when the humidity exceeds 70%.

## Steps:

- Log into your <https://ifttt.com> account
- Create this new applet:

if +this then that

