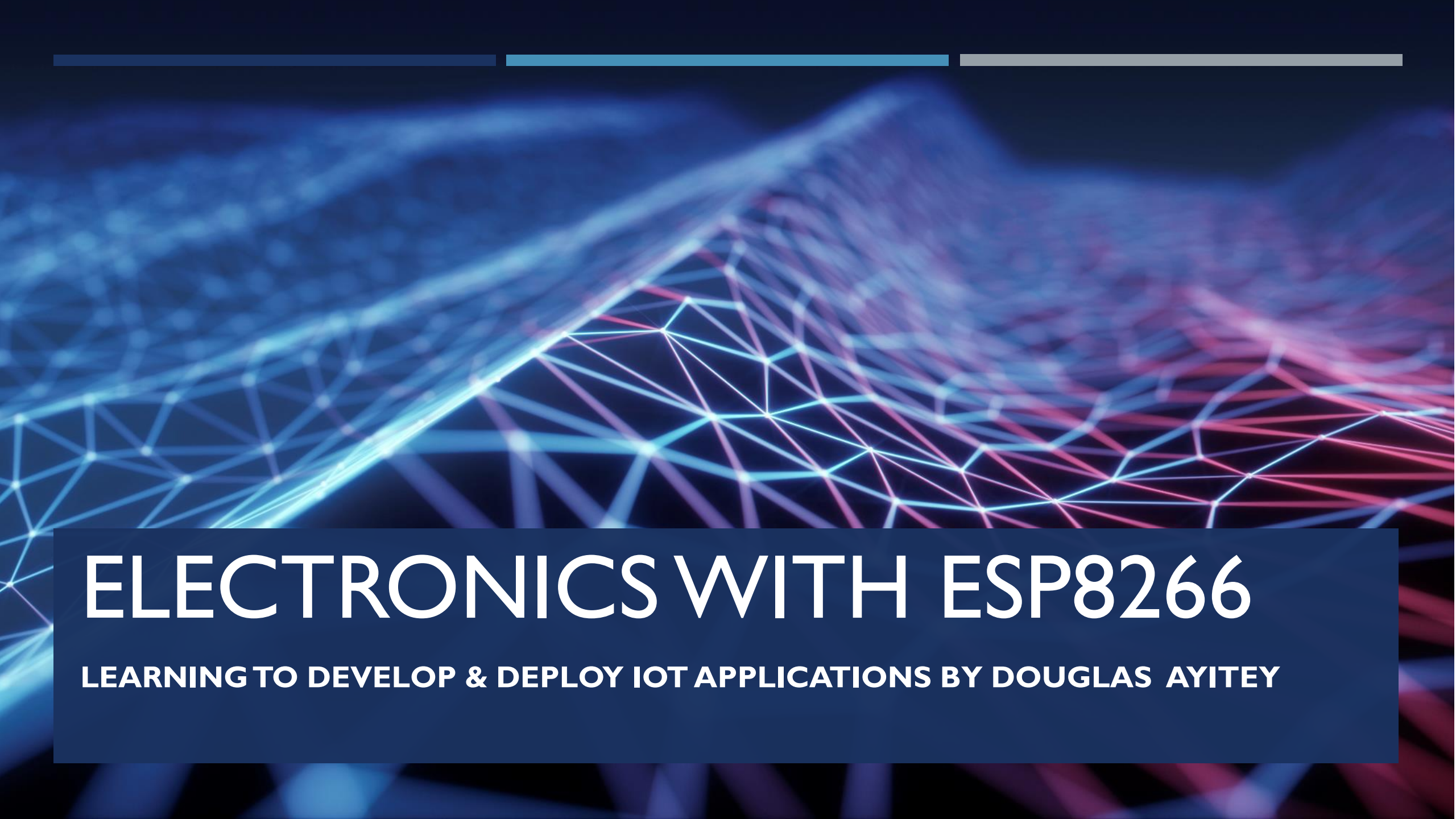
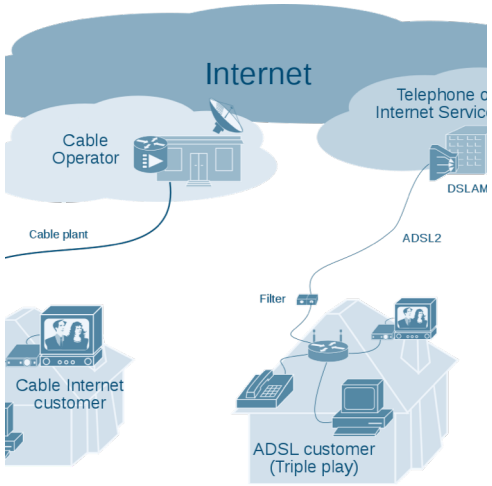
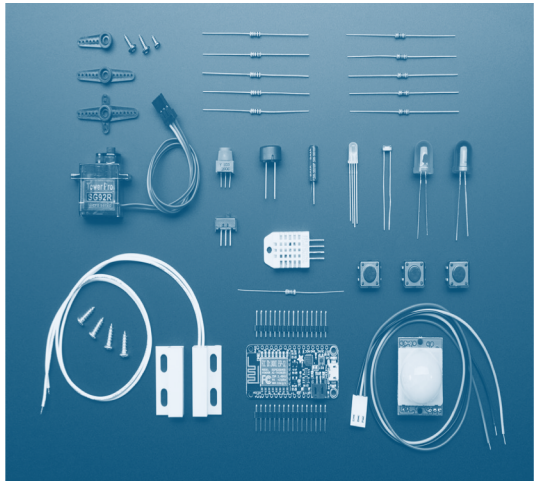


ELECTRONICS WITH ESP8266

LEARNING TO DEVELOP & DEPLOY IOT APPLICATIONS BY DOUGLAS AYITEY



Network



Components

Computer

TECH REQUIREMENTS

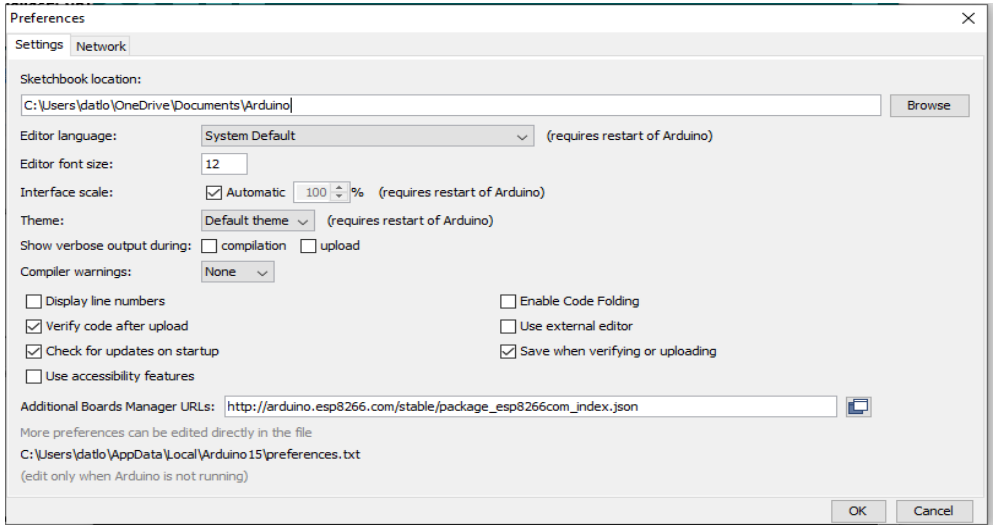
INSTALLING THE ARDUINO IDE FOR THE ESP8266

- Downloading and installing the Arduino IDE and getting the board definitions for our ESP8266 module.

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

Download the Arduino IDE

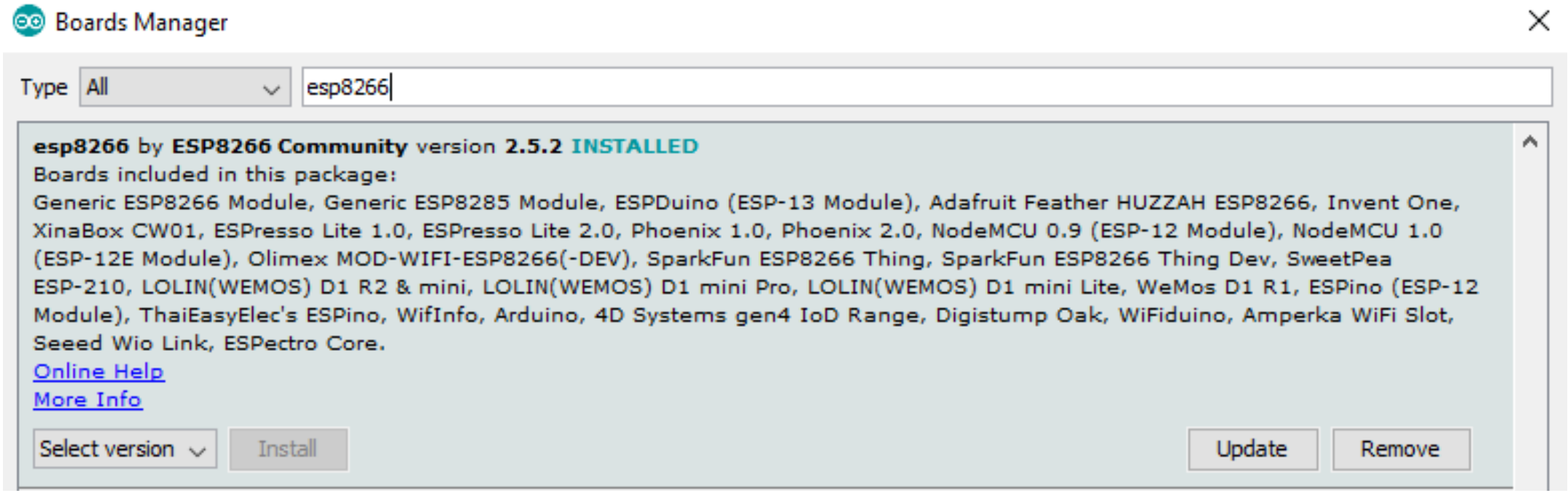
- To get the ESP8266 definitions. Open the preference window in the Arduino IDE from File | Preferences



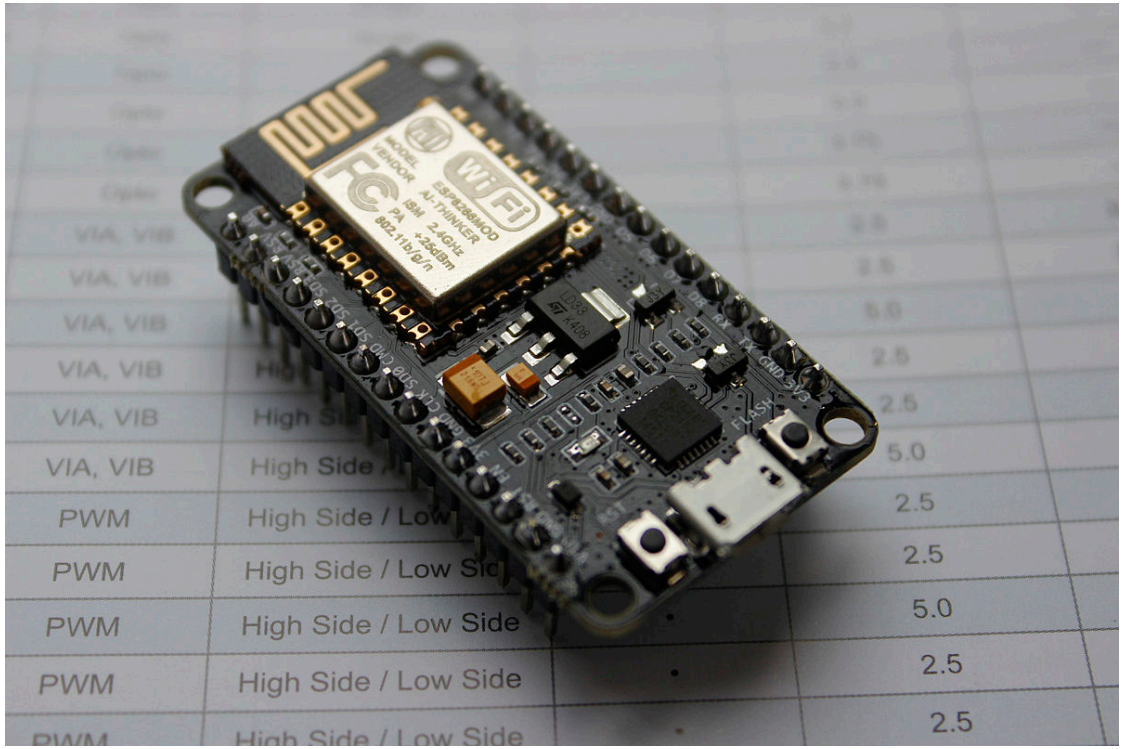
- Copy this URL: http://arduino.esp8266.com/stable/package_esp8266com_ index.json.

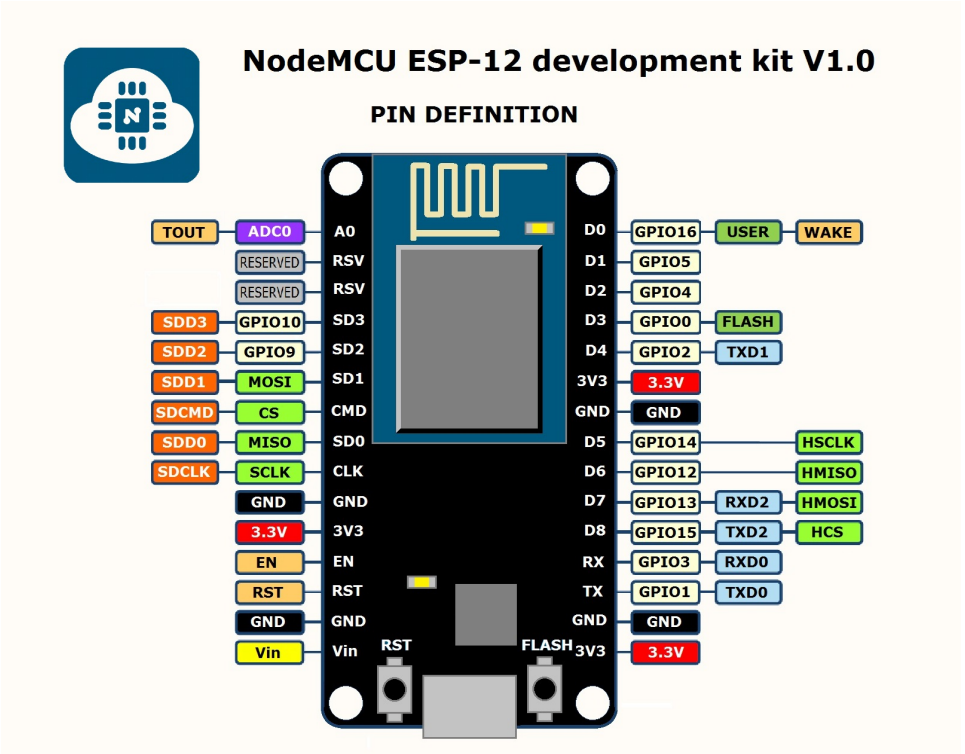
INSTALLING THE ARDUINO IDE FOR THE ESP8266

- Open the board manager from the **Tools | Board** menu and install the ESP8266 platform.



ALL ABOUT ESP8266





REQUIRED COMPONENTS

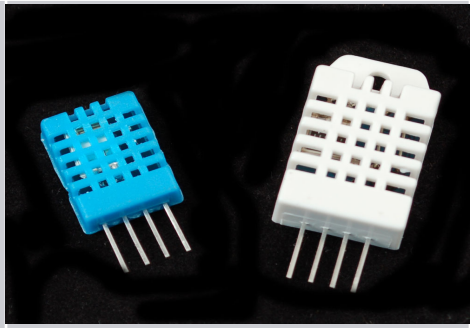
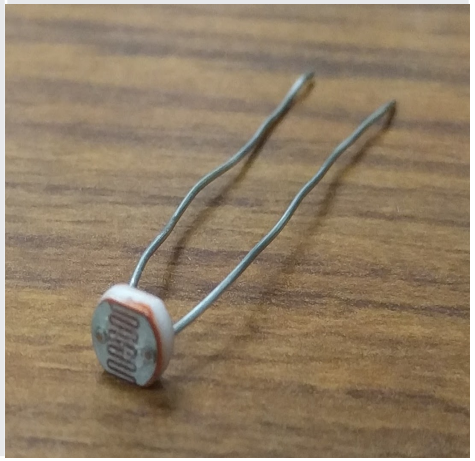
The components we will use include:

- Breadboard
- Jumper wires
- Assorted Resistors
- Transistor (2N2222, 2N3904)
- Micro USB cable
- Sensors
  - DHT11/22
  - Photocell

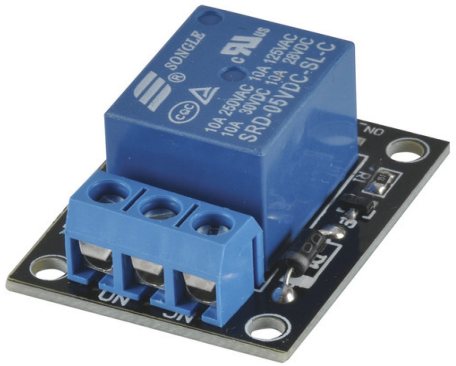
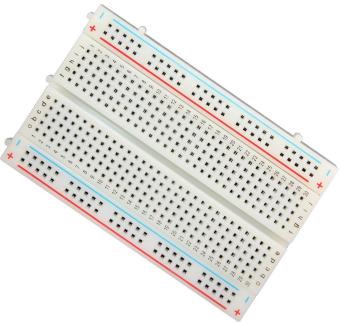
Actuators:

- Relay

COMPONENTS DETAILS

Component	Description	Image
DHT11/22	The DHT11 is a digital temperature and humidity sensor. It uses a thermistor and capacitive humidity sensor to monitor the humidity and temperature of the surrounding air, and produces a digital signal on the data pin. A digital pin on the ESP8266 can be used to read the data from the sensor data pin.	
Photocell	A photocell is a light sensor that changes its resistance depending on the amount of incident light it is exposed to. They can be used in a voltage divider setup to detect the amount of light in the surroundings. In a setup where the photocell is used in the VCC side of the voltage divider, the output of the voltage divider goes high when the light is bright and low when the light is dim. The output of the voltage divider is connected to an analog input pin and the voltage readings can be read:	

COMPONENTS DETAILS

Component	Description	Image
Relay	A relay is a switch that is operated electrically. It uses electromagnetism to switch large loads using small voltages. It consists of three parts: a coil, spring, and contacts. When the coil is energized by a high signal from a digital pin on the ESP8266, it attracts the contacts, forcing them closed. This completes the circuit and turns on the connected load. When the signal on the digital pin goes low, the coil is no longer energized and the spring pulls the contacts apart. This opens the circuit and turns off the connected load:	
Breadboard	A breadboard is used to temporarily connect components without soldering. This makes it an ideal prototyping accessory that comes in handy when building circuits	

COMPONENTS DETAILS

Component	Description	Image
Potentiometer	A potentiometer is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider. If only two terminals are used, one end and the wiper, it acts as a variable resistor or rheostat.	
Resistors	Resistors are electronic components which have a specific, never-changing electrical resistance. The resistor's resistance limits the flow of electrons through a circuit. They are passive components, meaning they only consume power (and can't generate it). Resistors are usually added to circuits where they complement active components like op-amps, microcontrollers, and other integrated circuits.	

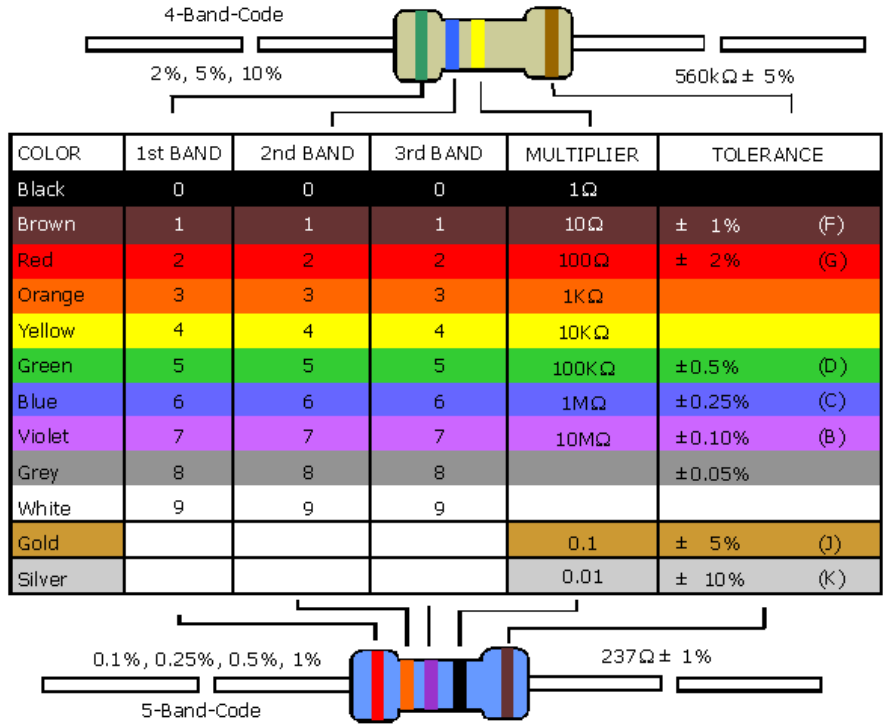
MORE ON RESISTORS AND APPLICATIONS

By using:

- Colour charts
- Multimeter
- Online calculato

https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-resistor-color-code-4-band

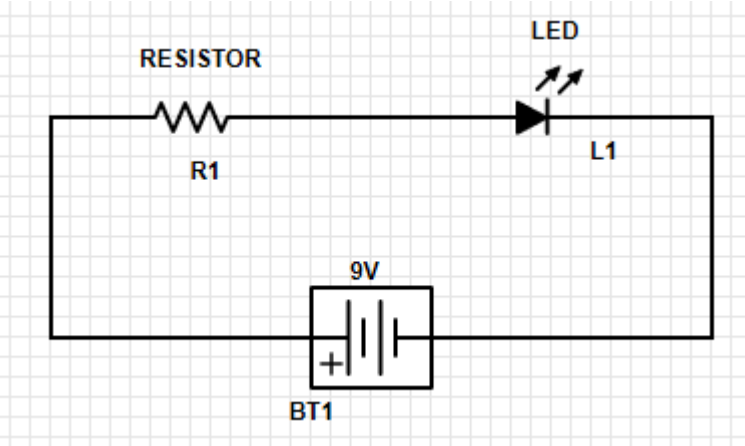
DECODING THE COLOUR BAND



COLOR	1st BAND	2nd BAND	3rd BAND	MULTIPLIER	TOLERANCE	
Black	0	0	0	1Ω		
Brown	1	1	1	10Ω	± 1%	(F)
Red	2	2	2	100Ω	± 2%	(G)
Orange	3	3	3	1KΩ		
Yellow	4	4	4	10KΩ		
Green	5	5	5	100KΩ	±0.5%	(D)
Blue	6	6	6	1MΩ	±0.25%	(C)
Violet	7	7	7	10MΩ	±0.10%	(B)
Grey	8	8	8		±0.05%	
White	9	9	9			
Gold				0.1	± 5%	(J)
Silver				0.01	± 10%	(K)

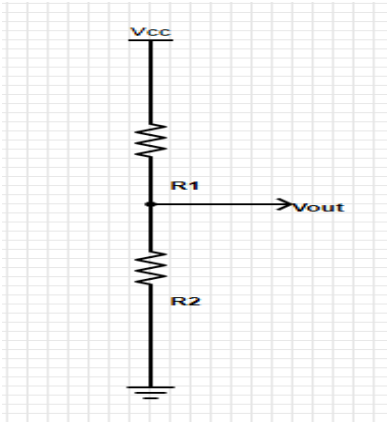
RESISTOR APPLICATIONS

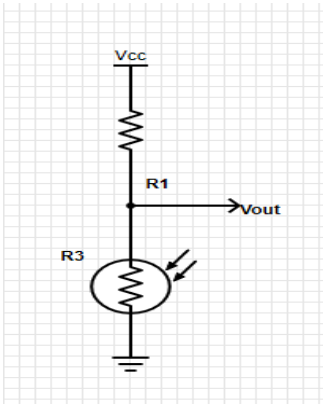
- Resistors are key in making sure LEDs don't blow up when power is applied. By connecting a resistor in series with an LED, current flowing through the two components can be limited to a safe value.



Current Limiting

- A voltage divider is a resistor circuit which turns a large voltage into a smaller one. Using just two resistors in series, an output voltage can be created that's a fraction of the input voltage.

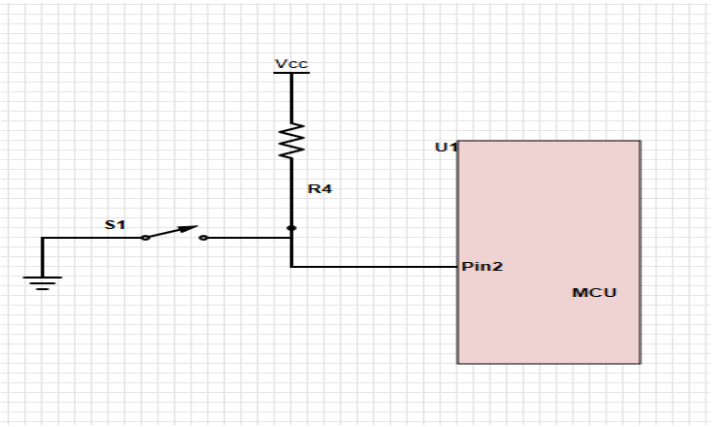




Voltage Divider

PULL-UP RESISTOR

- A pull-up resistor is used when you need to bias a microcontroller's input pin to a known state. One end of the resistor is connected to the MCU's pin, and the other end is connected to a high voltage (usually 5V or 3.3V).
- Without a pull-up resistor, inputs on the MCU could be left floating. There's no guarantee that a floating pin is either high (5V) or low (0V).
- Pull-up resistors are often used when interfacing with a button or switch input. The pull-up resistor can bias the input-pin when the switch is open. And it will protect the circuit from a short when the switch is closed.



PROJECTS

- Connecting the ESP8266 to WiFi Network
- Controlling an LED
- Reading the status of the GPIO pin
- Reading sensor data
  - Using DHT12 sensor, to obtain ambient temperature and humidity.
  - Remotely monitor the sensor data using aREST.
- Light Intensity Monitoring over the Cloud.
- Monitoring temperature & humidity in real-time.

CONNECTING ESP8266 TO WI-FI NETWORK

- Write a code and then upload it to the board.
- The code is to connect the board to the local Wi-Fi network and print the IP address of the board.

```
// Import required libraries
#include <ESP8266WiFi.h>
// WiFi parameters
const char* ssid = "*********";
const char* password = "**********";
```

```
void setup() {
// Start Serial
Serial.begin(115200);
// Connect to WiFi
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
  } Serial.println("");
Serial.println("WiFi connected");
// Print the IP address
Serial.println(WiFi.localIP());
}
```

```
void loop() {
  // put your main code here, to run
repeatedly:

}
```

CONTROLLING AN LED

- Light up the LED by programming the ESP8266 chip.
- **Modify the code to blink the LED continuously.**

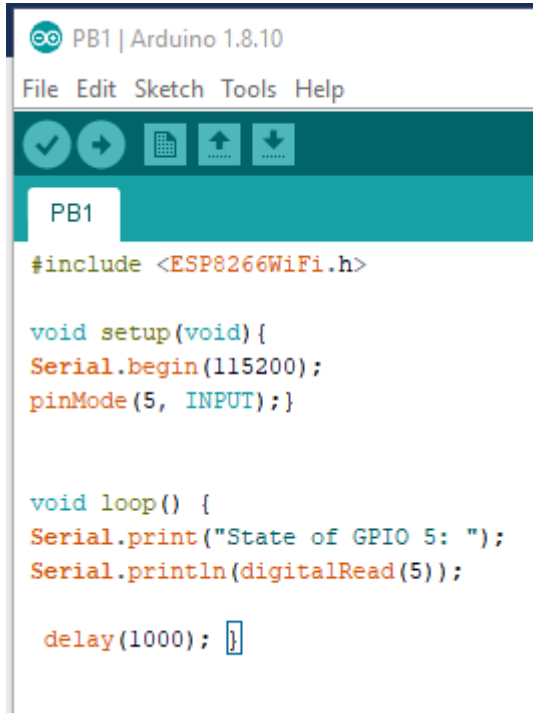
```
//Import the ESP8266 WiFi library
#include <ESP8266WiFi.h>

void setup() {
  // put your setup code here, to run once:
//Serial.begin(115200);

pinMode(5, OUTPUT);

digitalWrite(5, HIGH);

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

READING THE STATE OF GPIO PIN

- Remove the LED and the resistor.
- Now, simply connect this pin (GPIO 5) of the board to the positive power supply on your board with a wire, applying a 3.3V signal on this pin.
- This is the result you should get in the Serial monitor:

  State of GPIO 5: 1

- Connect the pin to the ground, and the state should go to 0.



READING DIGITAL SENSOR DATA

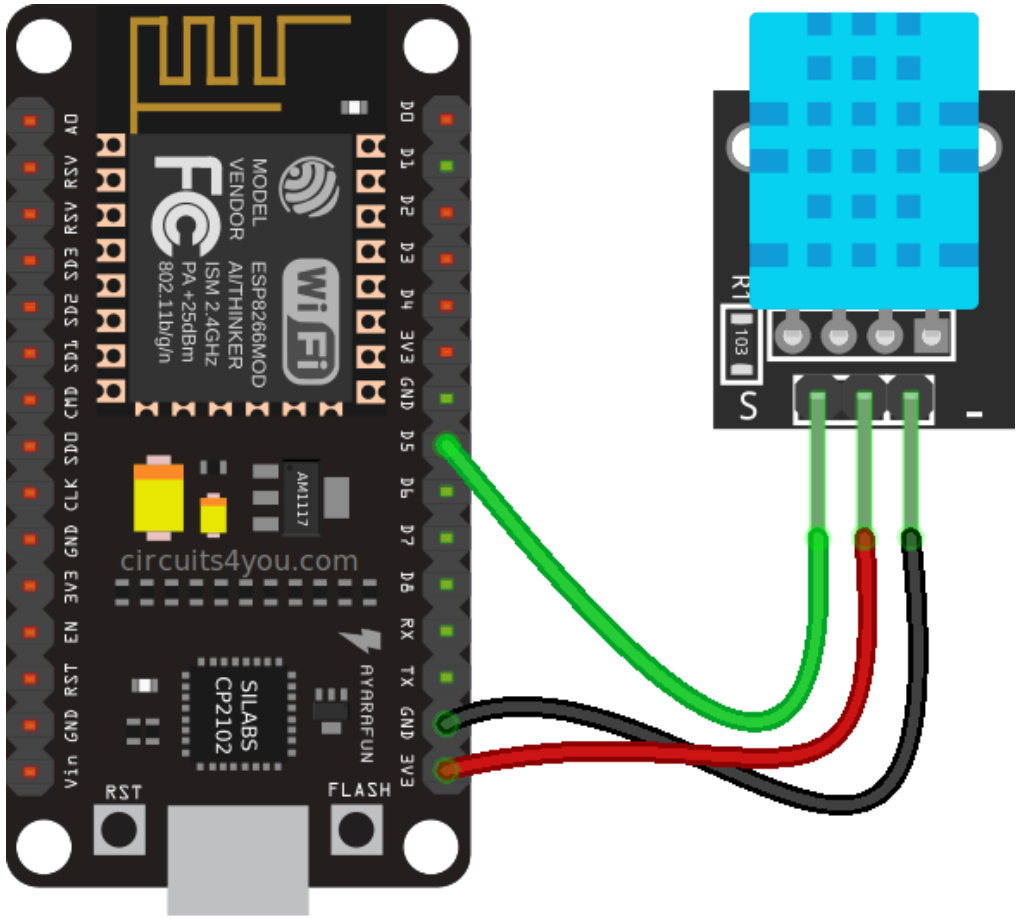
- Using DHT12 sensor, to get ambient temperature and humidity.
- Remotely monitor the sensor data using aREST.
- Add the following libraries

1. DHT Sensor Library:

https://github.com/adafruit/DHT-sensor-library

2. Adafruit Unified Sensor Lib:

https://github.com/adafruit/Adafruit_Sensor



- Include the require libraries

  #include "ESP8266WiFi.h" #include <aREST.h> #include "DHT.h"

- Set the pin that the DHT sensor is connected to

  #define DHTPIN 5 #define DHTTYPE DHT11

- Declare an instance of the DHT sensor

  DHT dht(DHTPIN, DHTTYPE, 15);

- Insert your own Wi-Fi name and password

  const char ssid = "wifi-name"; const char* password = "wifi-pass";*

- Define two variables that will hold the measurements of the sensor

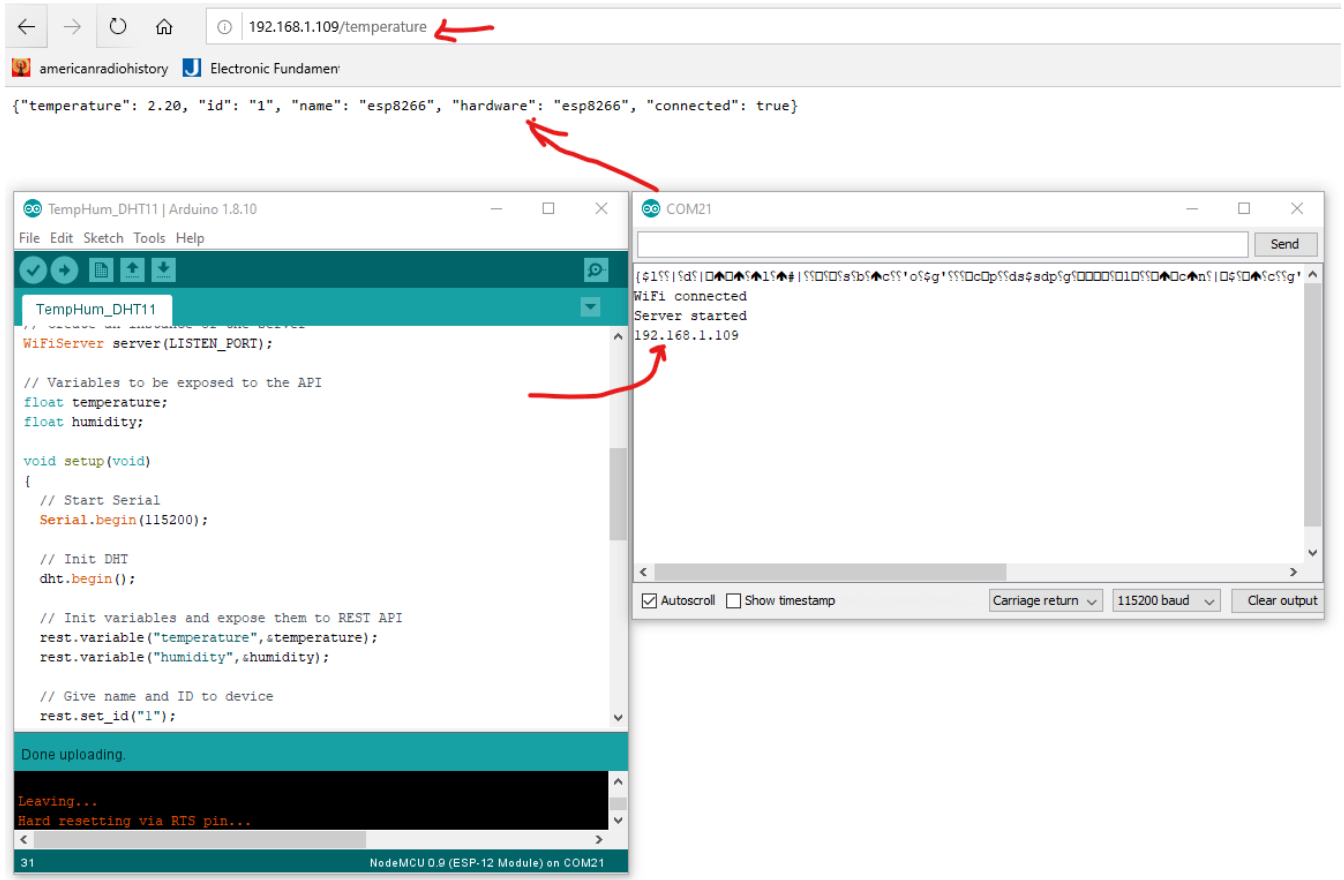
  float temperature; float humidity;

- initialize the sensor in setup function *dht.begin();*

- Input the variables to the aREST API, so we can access them remotely via Wi-Fi. *rest.variable("temperature",&temperature); rest.variable("humidity",&humidity);*

- Read the measurements from the sensor

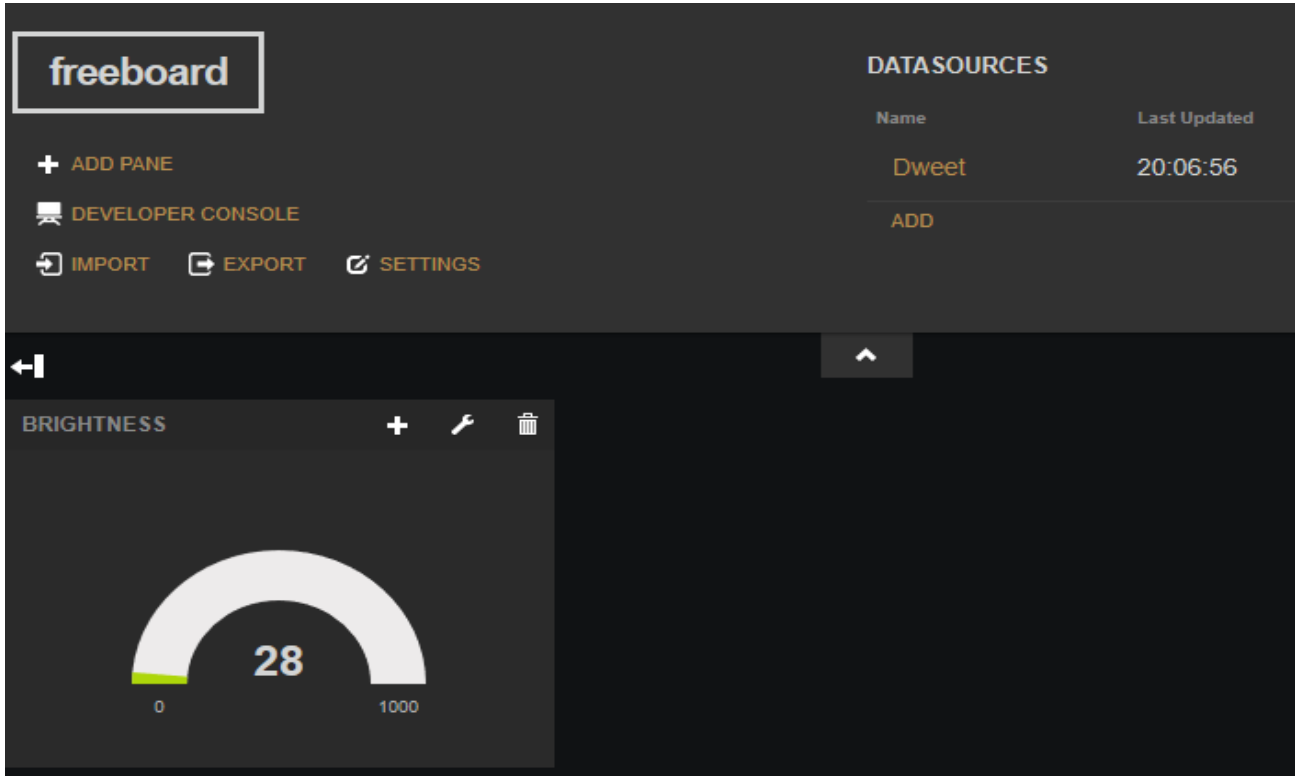
  humidity = dht.readHumidity(); temperature = dht.readTemperature();

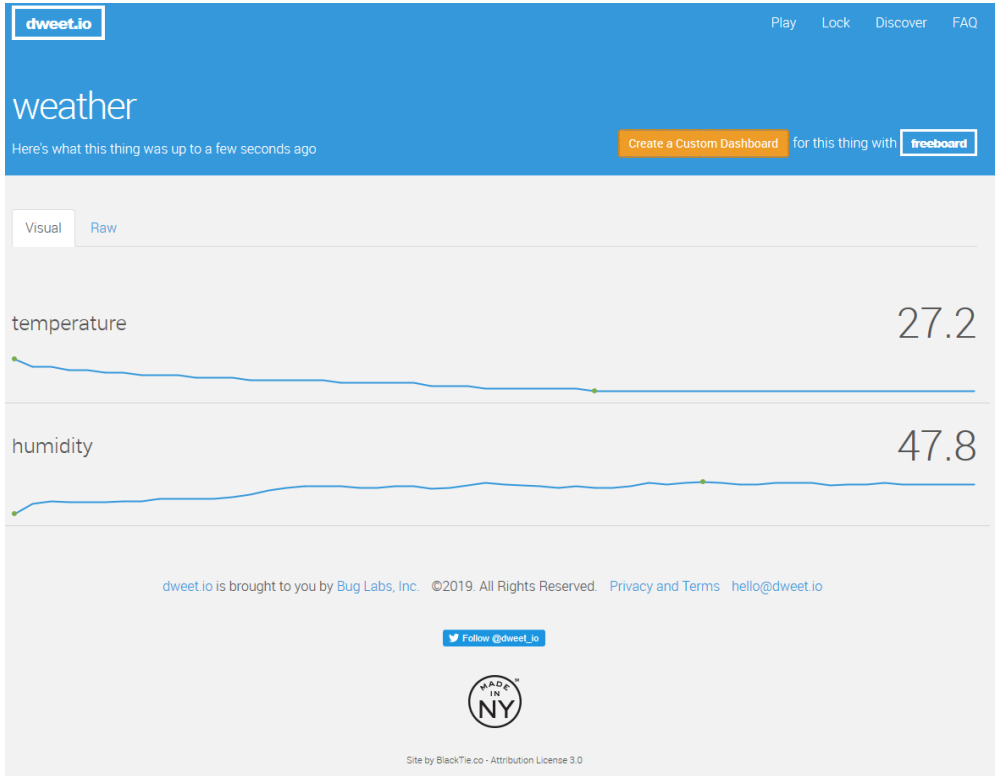


CREATING DASHBOARD

- Create an account at https://www.freeboard.io/
- Create a dashboard to link the datasource from Dweet.io as defined in your code.
- Add a gauge widget to display the sensor data

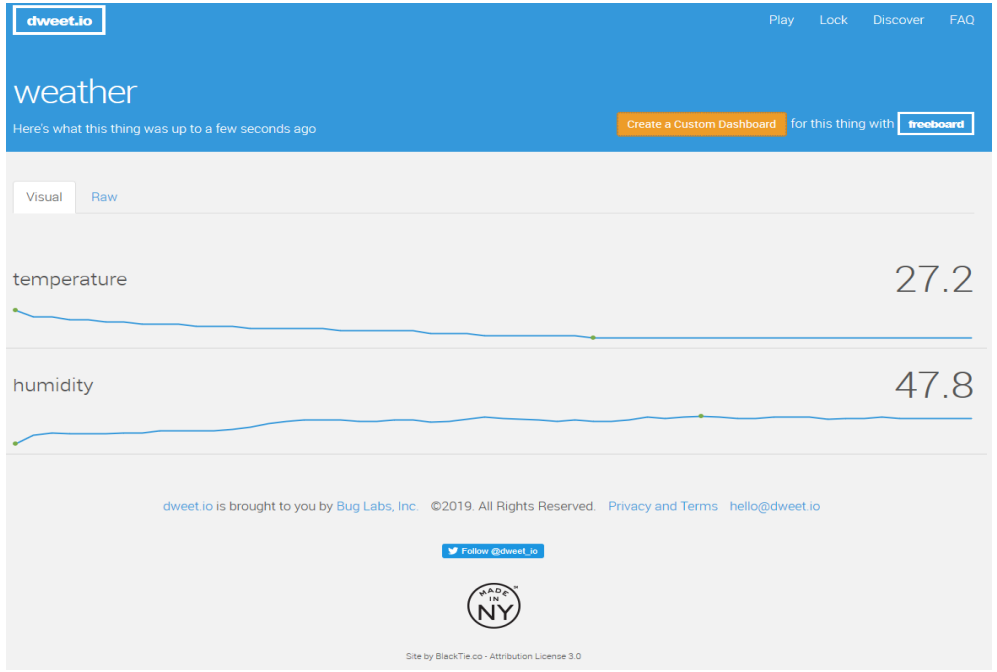
LIGHT INTENSITY MONITORING

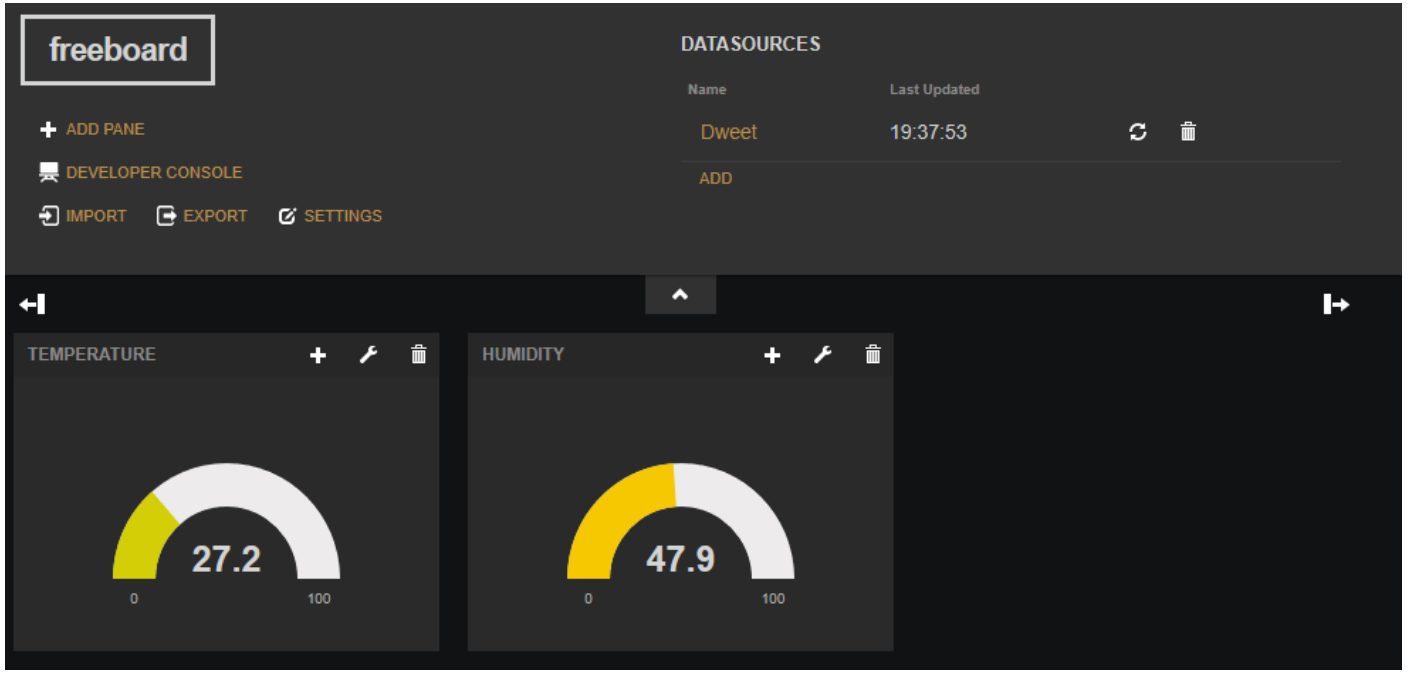




Using ESP8266 to remotely monitor light brightness.

READING DATA ONLINE





Results from dweet.io and freeboard.io

THANK YOU

FOLLOW ME:

TWITTER: @DATLOGIX

DATLOGIX@OUTLOOK.COM

WHATSAPP: 0205613190