**SETUP BEFORE WORKSHOP TO TEST:**  
**Set-up arduino for wemos:**

1. Open Arduino and plug in wemos to computer using the USB cable
2. Click Tools > Board > LOLIN(WEMOS) D1 R2 & mini

If this Board **doesn’t** show up on the list, follow these steps:

* 1. Click File > Preferences > Additional Board Managers
     1. Paste the following into the box:

<http://arduino.esp8266.com/stable/package_esp8266com_index.json>

* 1. Click Tools > Boards > Boards Manager
     1. Click Type > Contributed
     2. Find **esp8266 by ESP8266 Community** and click install
  2. Click Tools > Board > LOLIN(WEMOS) D1 R2 & mini

1. Click Tools > Upload Speed > 115200
2. Click Tools > Port > COMx, where x is any number greater than 1.
3. Copy paste this code:

void setup() {

pinMode(BUILTIN\_LED, OUTPUT); // set up the LED

}

void loop() {

digitalWrite(BUILTIN\_LED, HIGH); // turn off LED

delay(500); // wait a half second

digitalWrite(BUILTIN\_LED, LOW); // turn on LED

delay(500); // wait a half second

}

1. Upload it to the wemos using this button: If it manages to upload you should see a light on the wemos flashing every second! The code is turning the light on for 0.5s then off for 0.5s. Everything within the { } brackets is looped / repeated. This is a good example code to show the students.

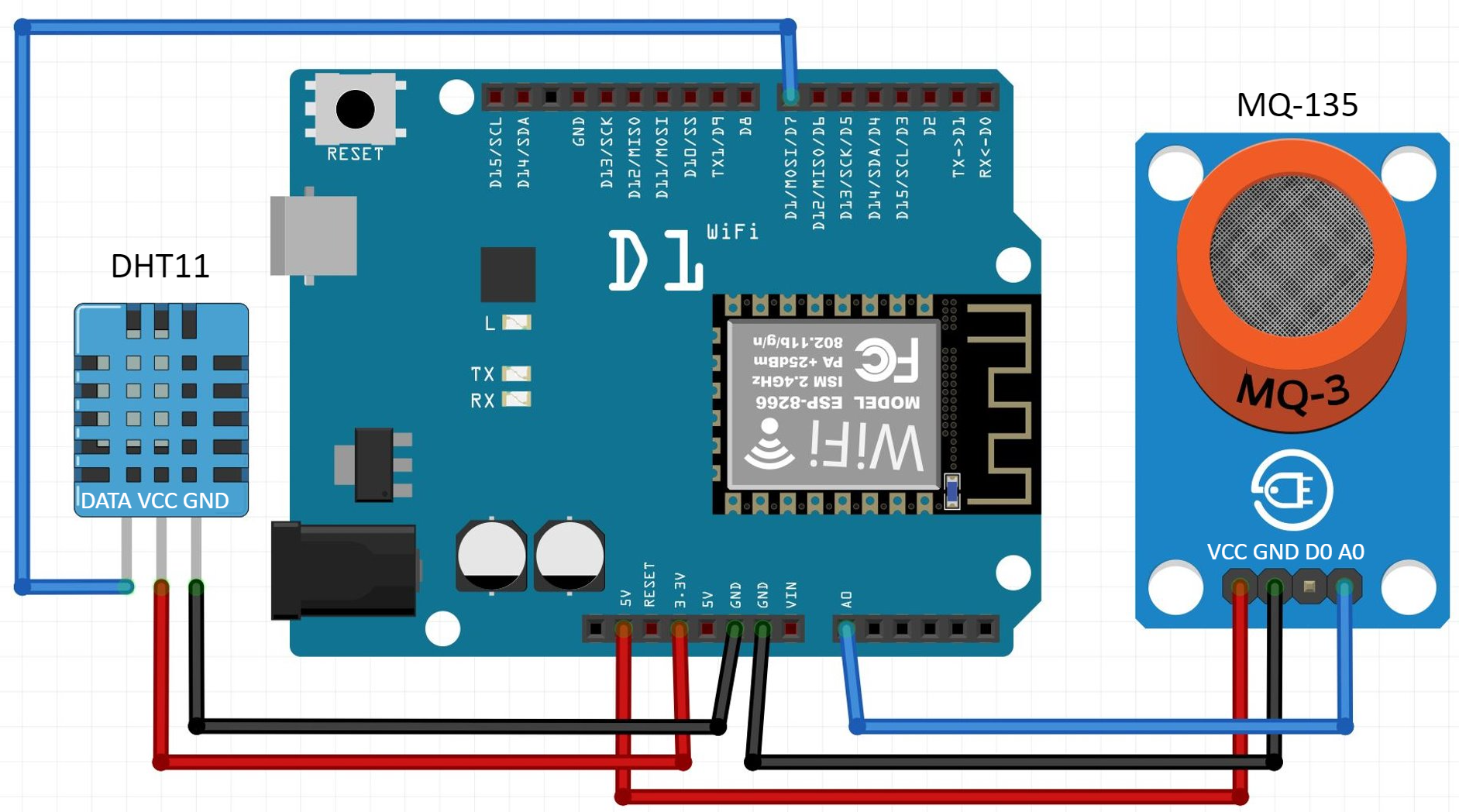
**Code to talk about with the students:**

**void setup()**  - function called at start will run only once

**void loop()** –everything within {} is looped consequitively after setup has finished

**delay(x)** – pause program for x number of milliseconds

**Upcoming wiring:**



**Electronics session pt 1:**

Start with the LED code above to demonstrate a loop and delays maybe?

**MQ-135 air quality sensor.** [Data sheet](https://www.olimex.com/Products/Components/Sensors/Gas/SNS-MQ135/resources/SNS-MQ135.pdf)Talk about MQ-135 with the students:

* General “air quality” sensor – gasses: CO2, Smoke, NH3, NOx, Alcohol, Benzene
* Sensor has a layer of SnO2 - Tin Oxide which has a higher resistance / low conductivity in clear air. When there’s increase in polluting gasses, the resistance of sensor drops / conductivity increases and gives out a higher voltage reading.
* Analog vs digital – a range of values through wemos’ analog pin. Digital pins only have high and low.
* Can see a small spike by breathing on the sensor, much bigger spike by lighting a lighter near the sensor. I got over 100ppm for breathing and over 300ppm for lighter.

1. Wire up MQ-135 -> wemos, using the analog pin. (left to right on the PCB back of MQ-135) Do on webcam if poss
   1. A0 -> A0
   2. Skip D0.
   3. GND -> GND
   4. VCC -> 5V
2. Download MQ-135 library: <https://github.com/ViliusKraujutis/MQ135> (click Clone or download > Download ZIP). in arduino click sketch > library and add the zip
3. add the following to the top of your code. Ask the students for a rough temperature and humidity values. Explain this is used to help calculate a ppm value for air quality.

#include "MQ135.h"

MQ135 mq135\_sensor = MQ135(A0);

float temperature = 21.0;

float humidity = 50.0;

1. Below this add:

void setup(){

Serial.begin(115200);

}

void loop(){

float a = mq135\_sensor.getCorrectedPPM(temperature, humidity);

Serial.print(a);

delay(5000);

}

1. Upload the code, open serial monitor (make sure baud is 115200). The sensor takes ~60 seconds to heat up. Values should print from the sensor. Try breathing on the sensor to change the value – or even a lighter

**Code to talk about with the students:**

**Serial.begin(x)** - set the data rate x bits per second (baud)

**Serial.print(x)** - this prints x to the serial port, then readable on the serial monitor

Actual calculation happening (see more in the [source library](https://github.com/ViliusKraujutis/MQ135/blob/master/MQ135.cpp)) based off [datasheet](https://www.olimex.com/Products/Components/Sensors/SNS-MQ135/resources/SNS-MQ135.pdf)

int val = analogRead(\_pin);

return ((1023./(float)val) - 1.)\*RLOAD;

return getResistance()/getCorrectionFactor(t, h);

**Electronics session pt 2:**

**DHT11 temp / humidity sensor:**

1. Wire up DHT11 -> wemos; (left to right on DHT11):
   1. S -> D7
   2. Middle pin -> 3V3
   3. **-** -> GND
2. Download library zip: <https://github.com/beegee-tokyo/DHTesp> (click Clone or download > Download ZIP) in arduino click sketch > library and add the zip
3. add the following to the top of your code:

#include "DHTesp.h"

DHTesp dht;

1. Add the following to the loop() after “if (client.connect(server,80))”:

humidity = dht.getHumidity();

temperature = dht.getTemperature();

Serial.print("humidity: ");

Serial.print(humidity);

Serial.print(", temp: ");

Serial.println(temperature);

1. Upload code and check serial monitor for temp & humidty readings. Tell the students this is now adjusting the ppm of the MQT-135 air quality sensor

**Coding session**

**Sign in to Thingspeak:**

1. Thingspeak.com. Create an account as each one is limited to four channels.
2. Click Channel > New Channel
   1. Give it a name
   2. Create channel
3. Sharing > Share channel view with everyone (so students can view)
4. API Keys > Write API Key. You’ll need this API key for the code below.

**Create thingspeak/IoT test code**

1. Copy code from below. Zip of code [here](https://uob.sharepoint.com/:u:/t/grp-sceemoutreach/EfOABRiQSFxNu3IDu3D5qDkB4F-FHGFT5bQN1nOn-WT3Zg?e=90zAgK)
2. Fill in API Key, Wifi name & password
3. Upload to wemos with button
4. Once uploaded, click the serial monitor button 
   1. Change baud to 115200
   2. You should see the wifi connect and the value get sent to teamspeak
   3. Every 20 seconds it will send a new value to team speak (currently integer t which is incrementing by 1 every loop.
   4. Have a look for your value on teamspeak coming in live!
   5. Now you’ve created an IOT device, let’s add a sensor.

**IoT Code:**

#include <ESP8266WiFi.h>

// replace with your channel’s write API key (on the API Keys tab) and your wifi name and password

String writeApiKey = "write api key here";

const char\* ssid = "wifi name here";

const char\* password = "wifi password here";

const char\* server = "api.thingspeak.com";

int t = 1;

WiFiClient client;

void setup(){

Serial.begin(115200);

delay(10);

WiFi.begin(ssid, password);

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED){

delay(500);

Serial.print(".");

}

Serial.println("WiFi connected");

}

void loop()

{

if (client.connect(server,80)) {

String postStr = writeApiKey;

postStr +="&field1=";

postStr += String(t);

client.print("POST /update HTTP/1.1\n");

client.print("Host: api.thingspeak.com\n");

client.print("Connection: close\n");

client.print("X-THINGSPEAKAPIKEY: "+writeApiKey+"\n");

client.print("Content-Type: application/x-www-form-urlencoded\n");

client.print("Content-Length: ");

client.print(postStr.length());

client.print("\n\n");

client.print(postStr);

Serial.print("Value: ");

Serial.println(t);

Serial.println("Sending data to Thingspeak");

t = t + 1;

}

client.stop();

Serial.println("Waiting 20 secs");

// thingspeak needs at least a 15 sec delay between updates

// 20 seconds to be safe

delay(20000);

}

**Code to talk about with the students:**

**char** – store a character

**string** – store zero or more characters

**Int t –** store a number – in this case “a” you’re sending to thingspeak

**a = a + 1** - incrementing integer a (within the loop)

**Add the sensors to the IoT Code**

[Full Wemos to Thingspeak code with sensors etc￼](https://uob.sharepoint.com/:u:/t/grp-sceemoutreach/EbG6S2Sth-hDliw94CMqwVkBUhUB0B6nLdlennUAHxAv6Q?e=Syja1r)

**At the top:**

#include "DHTesp.h"

DHTesp dht;

#include "MQ135.h"

MQ135 mq135\_sensor = MQ135(A0);

float temperature = 21.0;

float humidity = 50.0;

#include <SoftwareSerial.h>

**At the end of setup:**

dht.setup(D7, DHTesp::DHT11);

**In void loop:**

if (client.connect(server,80)) {

humidity = dht.getHumidity();

temperature = dht.getTemperature();

Serial.print("humidity: ");

Serial.print(humidity);

Serial.print(", temp: ");

Serial.println(temperature);

float a = mq135\_sensor.getCorrectedPPM(temperature, humidity);

String postStr = writeApiKey;

postStr +="&field1=";

postStr += String(a);

postStr +="&field2=";

postStr += String(humidity);

postStr +="&field3=";

postStr += String(temperature);

**Friday Matlab Session**

**Matlab to analyse data:**

Go to <https://matlab.mathworks.com/> and login with the same account from thingspeak.

**Useful code:**

[data,time] = thingSpeakRead(channelID, 'Numdays', x);

For reading the last number of days.  
Replace channelID with your ID and with x number of days (can’t read more than 8000 points or ~3 days)

[data,time] = thingSpeakRead(channelID,'DateRange',[datetime('May 26, 2020'),datetime('Jun 1, 2020')])

For reading specific dates. Months only 3 letters.

plot(time,data)

To plot a graph of the sensors over time.  
  
hold on

To keep this graph when adding more lines  
  
avedata = movmean(data,x)

Getting the moving mean. Comparing to x number of previous and future points  
  
plot(time,avedata)

Plot a graph of the moving mean.