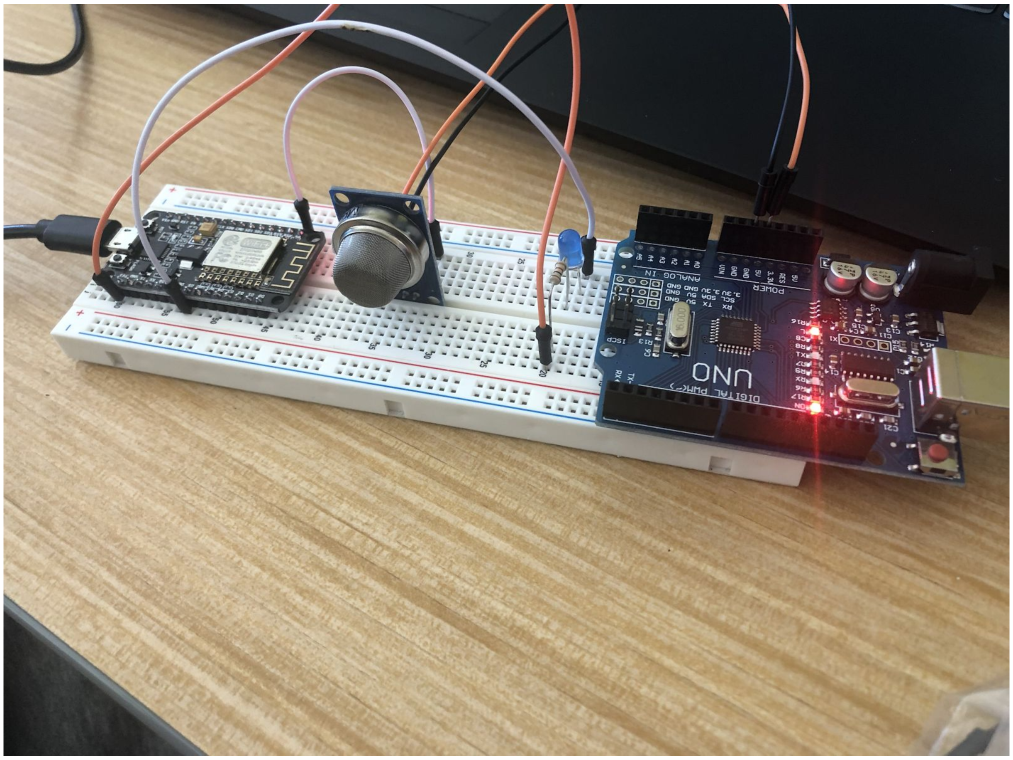
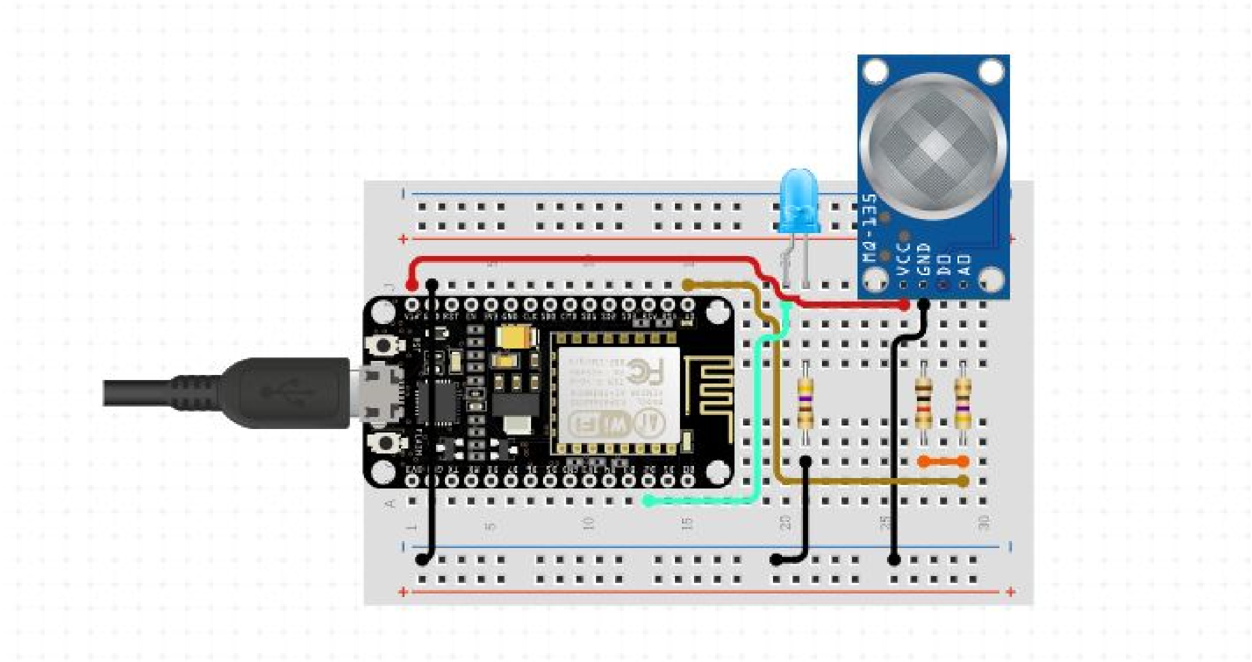
In TOOLS, go to BOARD and go to NodeMCU 1.0 (ESP-12E Module)
//we will be using this one for the wifi module

Hazardous Gas Sensor:



Schematic:



Notes:

The sensor is powered by the 5V from the arduino UNO. One can also use a different power source that can supply the necessary 5V.

The analog output of the sensor is connected to the A0 input of the Node MCU.

The 3.3V output of the NodeMCU is connected to a 1Kohm resistor in series with the LED cathode.

The anode of the LED is connected to the D5 output of the NodeMCU.

The code will take in the analog reading of the sensor and using the ppm command from the MQ135.h library, change the raw data into parts per million (ppm) readings.

The code will then compare this reading to 5 levels:

1. ppm < 50 : firebase airQ value is set to 0
2. 50 < ppm < 100 : firebase airQ value is set to 1
3. 100 < ppm < 150 : firebase airQ value is set to 2
4. 150 < ppm < 200 : firebase airQ value is set to 3
5. 200 < ppm < 300 : firebase airQ value is set to 4
6. 300 < ppm : firebase airQ value is set to 5

Unit Testing:

1. Aim a lighter at the gas sensor. Press and hold down the red button to release the butane gas. Make sure not to ignite the flame. The sensor should read an extremely high value outputting a 5 to the firebase database. In order to test that the other values set, increase the distance between the lighter and the sensor to decrease the density of the butane.
2. Place the gas sensor directly in front of a bottle of acetone or alcohol. The sensor should read high values of ppm and set a value of 5 to the firebase.

Integration Testing:

At the start of the circuit, WiFi is connected to. This will then be used to initiate a connection to the Firebase. The sensor is continuously updating the firebase by sending a signal every few seconds. We tested this by keeping the Firebase database open, where we would then change the states of the sensor, and would then view them on the database in real time.