IOT BASED AIR QUALITY MONITORING

PHASE 3:Devolpment Part 1

Deploying IoT devices

NodeMCU plays the main controlling role in this project. It has been programmed in a manner, such that, it senses the sensory signals from the sensors and shows the quality level via led indicators. The DHT11 sensor module is used to measure the temperature and the humidity of the surroundings.With the help of the MQ-135 gas sensor module, air quality is measured in ppm. These data are fed to the ThinkSpeak cloud over the internet. We have also provided LED indicators to indicate the safety levels.

STEP 1. Firstly, the calibration ofthe MQ-135 gas sensor module is done. The sensor is set to preheat for 24 minutes. Then the software code is uploaded to the NodeMCU followed by the hardware circuit to calibrate the sensor has been performed.

STEP 2. Then, the DHT11 sensor is set to preheat for 10 minutes.

STEP 3. The result of calibration found in STEP 1 is used to configure the final working code.

STEP 4. The final working code is then uploaded to the NodeMCU.

STEP 5. Finally, the complete hardware circuit is implemented.

HARDWARE MODEL

Hardware Model to Preheat DHT11 Sensor Module .

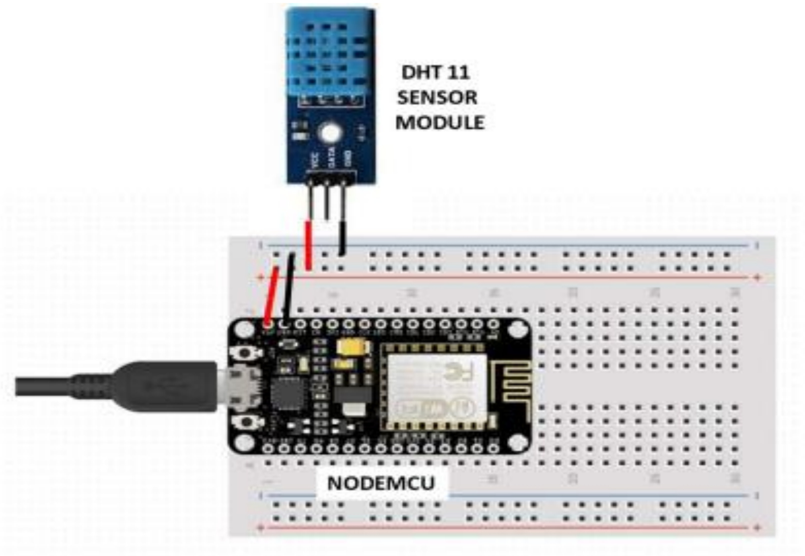
The following steps were performed to preheat the DHT11 sensor module:

STEP 1 : The Vcc pin of the DHT11 sensor module was connected with the VU pin of NodeMCU.

STEP 2 : The Gnd pin of the DHT11 sensor module was connected with the Gnd pin of NodeMCU.

STEP 3 : The NodeMCU is powered with a 12V DC via AC-DC adapter for 20 minutes.

STEP 4 : The setup was then disconnected.



(Circuit Diagram to Preheat the DHT11 sensor module)

Hardware Model to Preheat and Calibrate MQ-135 Gas Sensor Module

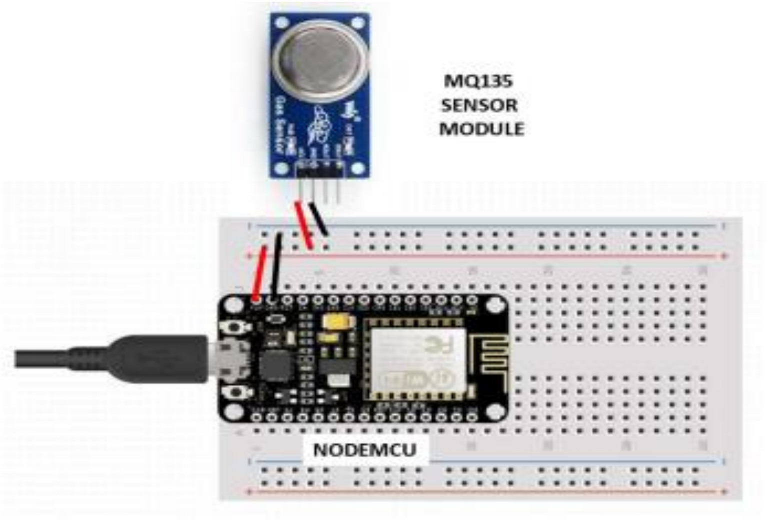
The following steps were performed to preheat the MQ-135 gas sensor module

STEP 1 : The Vcc pin of the MQ-135 gas sensor module was connected with the VU pin of NodeMCU.

STEP 2 : The Gnd pin of the MQ-135 gas sensor module was connected with the Gnd pin of NodeMCU.

STEP 3 : The NodeMCU is powered with a 12V DC via AC-DC adapter for a day.

STEP 4 : The setup was then disconnected.



(Circuit Diagram to Preheat the MQ-135 Gas sensor module)

Final Hardware Model

The following steps were performed to execute the project

STEP 1 : The Vcc pin of the MQ-135 gas sensor module and DHT11 sensor module was connected via Veroboard with an adapter delivering around 5V.

STEP 2 : The Gnd pin of the MQ-135 gas sensor module, DHT11 sensor module and the cathode of the LED indicators was connected via Veroboard with the Gnd pin of the NodeMCU.

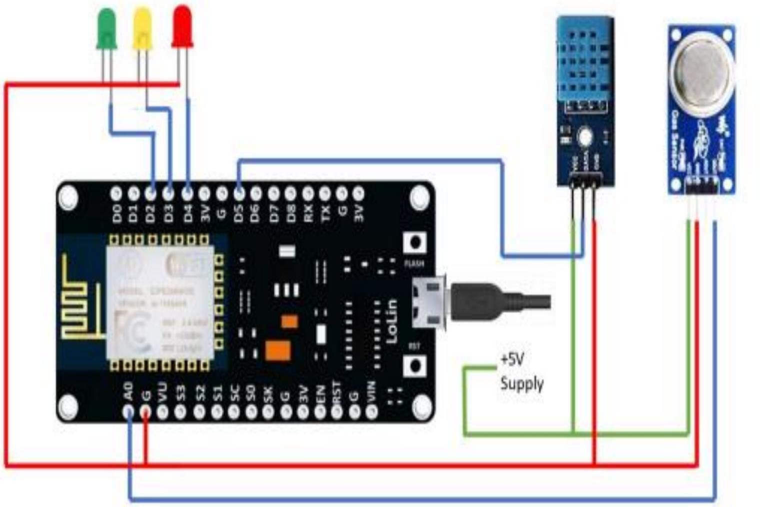
STEP 3 : The analog DATA pin of the MQ-135 gas sensor module was connected with the A0 Pin of the NodeMCU.

STEP 4 : The DATA pin of the DHT11 sensor module was connected with the D0 pin of the NodeMCU.

STEP 5 : The anode of the three LED indicators (green, yellow, and red) were connected to the D2, D3, and D4 pins of the NodeMCU respectively.

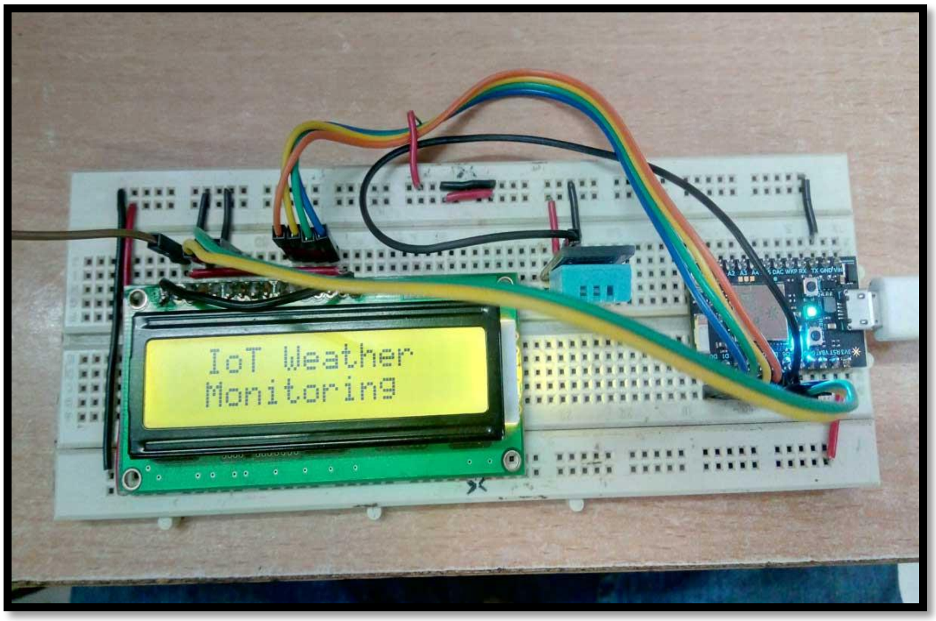
STEP 6 : The software code to execute the project was then uploaded to the NodeMCU.

STEP 7 : The setup was then powered with 9V DC via AC-DC adapter.



0

Circuit Diagram of the setup



DEVELOPING A PYTHON SCRIPT:

```
import streams

# import the wifi interface
from wireless import wifi

# import wifi support
from espressif.esp8266wifi import esp8266wifi as wifi_driver

streams.serial()

# init the wifi driver!
# The driver automatically registers itself to the wifi interface
# with the correct configuration for the selected board
wifi_driver.auto_init()

# use the wifi interface to link to the Access Point
# change network name, security and password as needed
print("Establishing Link...")
try:
   # FOR THIS EXAMPLE TO WORK, "Network-Name" AND "Wifi-Password" MUST BE SET
```

```
    # TO MATCH YOUR ACTUAL NETWORK CONFIGURATION
    wifi.link("Network-name",wifi.WIFI_WPA2,"password")
except Exception as e:
    print("ooops, something wrong while linking :(", e)
    while True:
        sleep(1000)
```