**Air quality monitoring using IoT**

**Phase 3:**

***Introduction***

we are going to build an ESP32 Air Quality Monitoring System using Nova PM SDS011 sensor, MQ-7 sensor, and DHT11 sensor. We will also be using an OLED Display module to display Air Quality Values. The Air Quality Index (AQI) in India is based on eight pollutants, PM10, PM2.5, SO2 and NO2, CO, Ozone, NH3, and Pb. However, it is not necessary to measure all of the pollutants. So we are going to measure the concentration of PM2.5, PM10, and Carbon Monoxide to calculate the Air Quality Index. The AQI values will be published on Adafruit IO so that we can monitor it from anywhere.

**Components Required**

ESP32

Nova PM Sensor SDS011

0.96’ SPI OLED Display Module

DHT11 Sensor

MQ-7 Sensor

Jumper Wires

**Sensors**

**Nova PM Sensor SDS011**

The SDS011 Sensor is a very recent Air Quality Sensor developed by Nova Fitness. It works on the principle of laser scattering and can get the particle concentration between 0.3 to 10μm in the air. This sensor consists of a small fan, air inlet valve, Laser diode, and photodiode. The air enters through the air inlet where a light source (Laser) illuminates the particles and the scattered light is transformed into a signal by a photodetector. These signals are then amplified and processed to get the particle concentration of PM2.5 and PM10. We previously used Nova PM Sensor with Arduino to calculate the concentration of PM10 & PM2.5.

SDS011 Sensor Specifications:

Output: PM2.5, PM10

Measuring Range: 0.0-999.9μg/m3

Input Voltage: 4.7V to 5.3V

Maximum Current: 100mA

Sleep Current: 2mA

Response Time: 1 second

Serial Data Output Frequency: 1 time/second

Particle Diameter Resolution:≤0.3μm

Relative Error: 10%

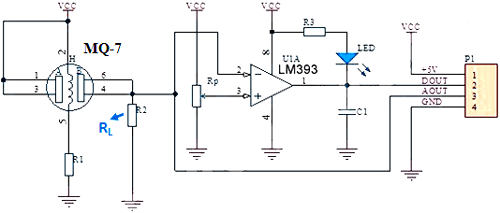
Temperature Range: -20~50°C

**0.96’ OLED Display Module**

OLED (Organic Light Emitting Diode) is a kind of Light Emitting Diode that is made using organic compounds that excites when the electric current is allowed to flow through them. These organic compounds have their own light hence they don’t require any backlight circuitry like normal LCDs. Because of this reason, OLED display technology is power efficient and widely used in Televisions and other display products.

**MQ-7 Sensor to Measure Carbon Monoxide (CO)**

MQ-7 CO Carbon Monoxide Gas Sensor Module detects the concentrations of CO in the air. The sensor can measure concentrations of 10 to 10,000 ppm. MQ-7 sensor can either purchased as a module or just as a sensor alone. Previously we have used many different types of Gas sensors to detect and measure various gas, you can also check them out if you are interested. In this project, we are using the MQ-7 sensor module to measure Carbon Monoxide concentration in PPM. The circuit diagram for the MQ-7 board is given below:



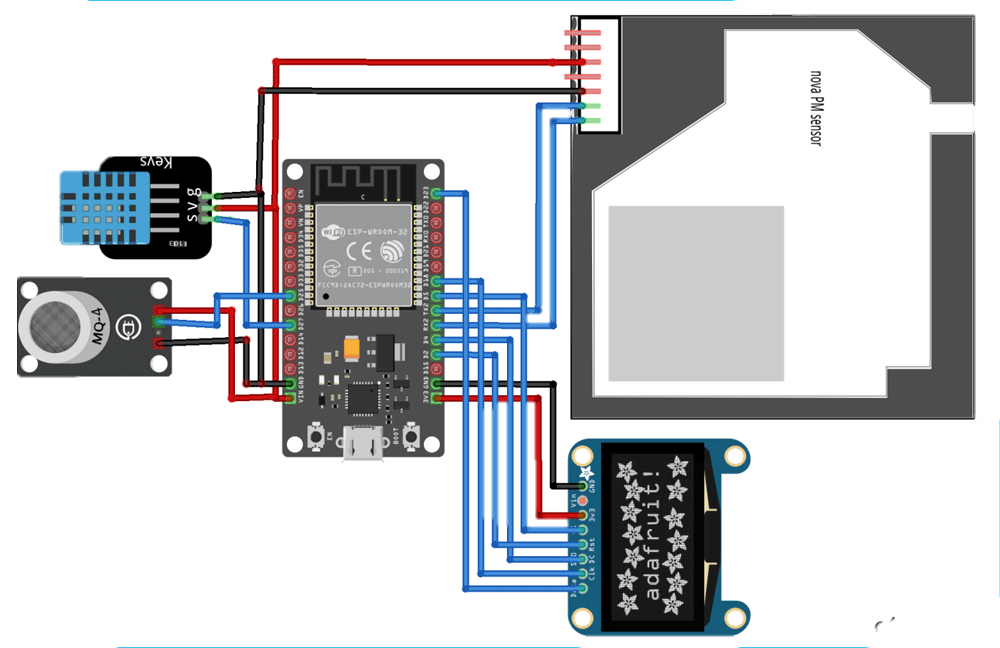
The load resistor RL plays a very important role in making the sensor work. This resistor changes its resistance value according to the concentration of gas. The MQ-7 sensor board comes with a Load resistance of 1KΩ that is useless and affects the sensor readings. So to measure the appropriate CO concentration values, you have to replace the 1KΩ resistor with a 10KΩ resistor.

**Air Quality Index Calculation**

The AQI in India is calculated based on the average concentration of a particular pollutant measured over a standard time interval (24 hours for most pollutants, 8 hours for carbon monoxide and ozone). For example, the AQI for PM2.5 and PM10 is based on 24-hour average concentration and AQI for Carbon Monoxide is based on 8-hour average concentration). The AQI calculations include the eight pollutants that are PM10, PM2.5, Nitrogen Dioxide (NO2), Sulphur Dioxide (SO2), Carbon Monoxide (CO), ground-level ozone (O3), Ammonia (NH3), and Lead (Pb). However, all of the pollutants are not measured at every location.

Based on the measured 24-hour ambient concentrations of a pollutant, a sub-index is calculated, which is a linear function of concentration (e.g. the sub-index for PM2.5 will be 51 at concentration 31 µg/m3, 100 at concentration 60 µg/m3, and 75 at a concentration of 45 µg/m3). The worst sub-index (or maximum of all parameters) determines the overall AQI.

**Circuit Diagram**



***Code***

import machine

import time

import dht

import urequests

from umqtt.simple import MQTTClient

from machine import I2C, Pin

from machine import ADC, SPI

# DHT11 sensor

dht\_pin = Pin(14, Pin.IN, Pin.PULL\_UP) # DHT11 sensor pin

dht\_sensor = dht.DHT11(dht\_pin)

# MQ7 sensor

mq\_pin = Pin(33, Pin.IN) # MQ7 sensor analog pin

# Nova SDS011 sensor

uart = machine.UART(1, tx=17, rx=16, baudrate=9600, txbuf=256)

uart.init(9600, bits=8, parity=None, stop=1)

# OLED display

i2c = I2C(scl=Pin(22), sda=Pin(21))

oled = ssd1306.SSD1306\_I2C(128, 64, i2c)

# MQTT configuration

mqtt\_server = "mqtt.eclipse.org"

mqtt\_topic = "your\_topic" # Replace with your MQTT topic

mqtt\_client\_id = "esp32\_air\_quality\_monitor"

mqtt\_client = MQTTClient(mqtt\_client\_id, mqtt\_server)

def read\_dht():

dht\_sensor.measure()

temperature = dht\_sensor.temperature()

humidity = dht\_sensor.humidity()

return temperature, humidity

def read\_mq():

mq\_value = ADC(mq\_pin).read()

return mq\_value

def read\_sds011():

data = uart.read(10)

if data[0] == 0xAA and data[1] == 0xC0:

pm25 = (data[3] \* 256 + data[2]) / 10.0

pm10 = (data[5] \* 256 + data[4]) / 10.0

return pm25, pm10

return None

def send\_data\_to\_mqtt(temperature, humidity, mq\_value, pm25, pm10):

data = {

"temperature": temperature,

"humidity": humidity,

"mq\_value": mq\_value,

"pm25": pm25,

"pm10": pm10

}

mqtt\_payload = ujson.dumps(data)

mqtt\_client.connect()

mqtt\_client.publish(mqtt\_topic, mqtt\_payload)

mqtt\_client.disconnect()

def display\_data\_on\_oled(temperature, humidity, mq\_value, pm25, pm10):

oled.fill(0)

oled.text("Temp: {:.1f} C".format(temperature), 0, 0)

oled.text("Humidity: {:.1f}%".format(humidity), 0, 12)

oled.text("MQ Value: {}".format(mq\_value), 0, 24)

oled.text("PM2.5: {:.1f}".format(pm25), 0, 36)

oled.text("PM10: {:.1f}".format(pm10), 0, 48)

oled.show()

while True:

try:

temperature, humidity = read\_dht()

mq\_value = read\_mq()

pm25, pm10 = read\_sds011()

if pm25 is not None and pm10 is not None:

send\_data\_to\_mqtt(temperature, humidity, mq\_value, pm25, pm10)

display\_data\_on\_oled(temperature, humidity, mq\_value, pm25, pm10)

time.sleep(60) # Adjust the delay as needed

except Exception as e:

print("An error occurred:", e)

time.sleep(10)

**Running the system:**

1.upload the pythonscript to esp32 board .

2.connect sensors to the appropriate pins on esp32.

3.connect the power supply to esp32.

4.the output are shown in the display and stores in cloud platform.