Course Name: Internet of Things Lab **Course code:** 21CSP-344

Experiment – 3.1

Student Name: Updesh Kaur Benipal **UID:** 21ICS1021
Branch: CSE **Section/Group:** 646-B
Semester: 5th **Date of Performance:**
Subject Name: Internet of Things Lab **Subject Code:** 21CSP-344

Aim: To design a weather station by checking Air quality of an environment with the help of IoT.

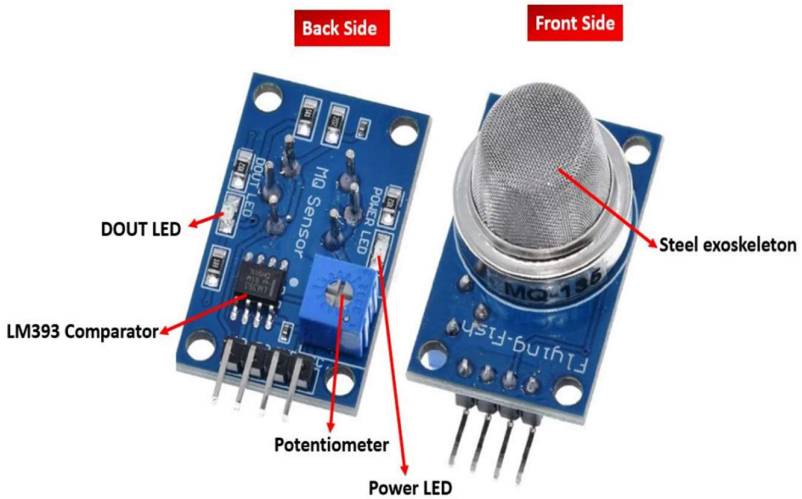
Objectives:
- Learn about actuators.

Software used: Arduino UNO

Hardware used:
- MQ 135 Air Quality Sensor Module
- Arduino Uno R3 board
- Male to female jumper wires

Theory:

Air Quality Sensor: The MQ-135 Air Quality Sensor, part of the MQ series, plays a vital role in detecting various gases in the environment. It is proficient in identifying a wide range of gases, including NH3, NOx, alcohol, Benzene, smoke, and CO2. The sensor's robust steel exoskeleton ensures durability, housing a highly sensitive sensing device within the gas sensor module. The MQ-135 sensor is a versatile tool used in applications where monitoring and maintaining air quality is crucial, such as in air purification systems, industrial safety, and environmental monitoring. Its ability to detect multiple gases makes it a valuable component in ensuring health and safety standards are met.



Course Name: Internet of Things Lab **Course code:** 21CSP-344

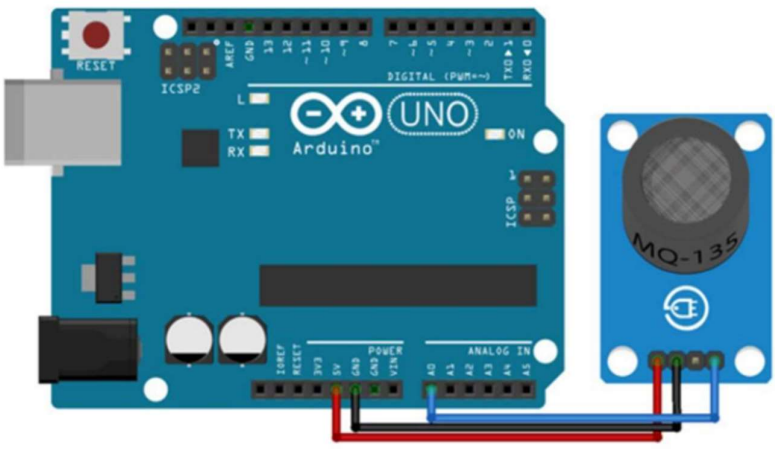
The table below shows some key specifications of the MQ-135 sensor module:

Feature	Description
Operating Voltage	2.5-5.0V
Detecting Concentration	10ppm-300ppm for NH3 10ppm-1000ppm for Benzene 10ppm-300ppm for Alcohol
Load Resistance	Adjustable
Heater Resistance	33Ω ± 5%
Heater Consumption	less than 800mW
Operating Temperature	-10 to 45°C

This sensor has 4 pins:

- 5V: Module power supply – 5 V (Positive power supply pin that powers up the sensor module)
- GND: Ground (Reference potential pin)
- DOUT: Digital output (It also produces a digital signal whose limit can be set using the in-built potentiometer)
- AOUT: Analog output (It generates a signal proportional to the concentration of gas vapors coming in contact with the sensor)
-

Circuit Diagram:

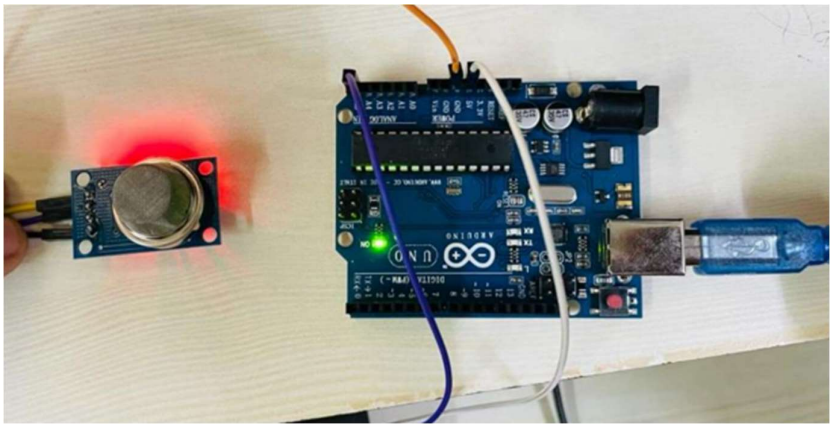


Course Name: Internet of Things Lab **Course code:** 21CSP-344

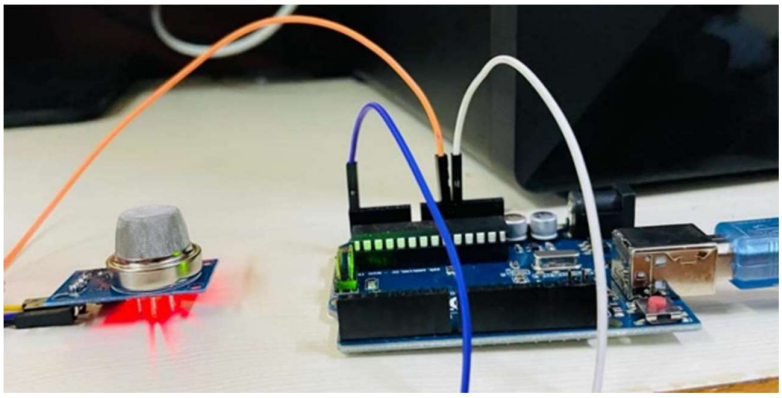
Code:

```
int sensorValue;
int digitalValue;
void setup()
{
Serial.begin(9600);
pinMode(13, OUTPUT);
pinMode(2, INPUT);
}
void loop()
{
sensorValue = analogRead(2);
digitalValue = digitalRead(2);
if (sensorValue > 400)
{
 digitalWrite(13, HIGH);
}
else
 digitalWrite(13, LOW);
 Serial.println(sensorValue, DEC);
 Serial.println(digitalValue, DEC);
delay(1000);
}
```

Result/Conclusion:
The air value was varying between 64 – 65.



Course Name: Internet of Things Lab **Course code:** 21CSP-344



Output Serial Monitor ×

Message (Enter to send message to 'Arduino Uno' on 'COM13')

```
SENSOR VALUE = 65 (NORMAL)
SENSOR VALUE = 65 (NORMAL)
SENSOR VALUE = 65 (NORMAL)
SENSOR VALUE = 64 (IMPURE AIR)
SENSOR VALUE = 64 (IMPURE AIR)
SENSOR VALUE = 64 (IMPURE AIR)
SENSOR VALUE = 65 (NORMAL)
SENSOR VALUE = 64 (IMPURE AIR)
SENSOR VALUE = 65 (NORMAL)
SENSOR VALUE = 64 (IMPURE AIR)
SENSOR VALUE = 64 (IMPURE AIR)
SENSOR VALUE = 64 (IMPURE AIR)
SENSOR VALUE = 64 (IMPURE AIR)
SENSOR VALUE = 64 (IMPURE AIR)
SENSOR VALUE = 64 (IMPURE AIR)
SENSOR VALUE = 65 (NORMAL)
SENSOR VALUE = 64 (IMPURE AIR)
SENSOR VALUE = 64 (IMPURE AIR)
SENSOR VALUE = 65 (NORMAL)
SENSOR VALUE = 65 (NORMAL)
SENSOR VALUE = 65 (NORMAL)
SENSOR VALUE = 65 (NORMAL)
SENSOR VALUE = 65 (NORMAL)
SENSOR VALUE = 65 (NORMAL)
SENSOR VALUE = 65 (NORMAL)
SENSOR VALUE = 65 (NORMAL)
SENSOR VALUE = 64 (IMPURE AIR)
SENSOR VALUE = 64 (IMPURE AIR)
SENSOR VALUE = 64 (IMPURE AIR)
SENSOR VALUE = 65 (NORMAL)
SENSOR VALUE = 64 (IMPURE AIR)
SENSOR VALUE = 65 (NORMAL)
SENSOR VALUE = 64 (IMPURE AIR)
```

Fig: Serial Monitor showing sensor value

Learning Outcomes:

- Learned about Arduino-based actuator control.
- Learned about coding for IoT applications.