

Smart Air Purification System and Jupyter Analysis

Embedded Systems Project

Vimal Jain (16CSU419)

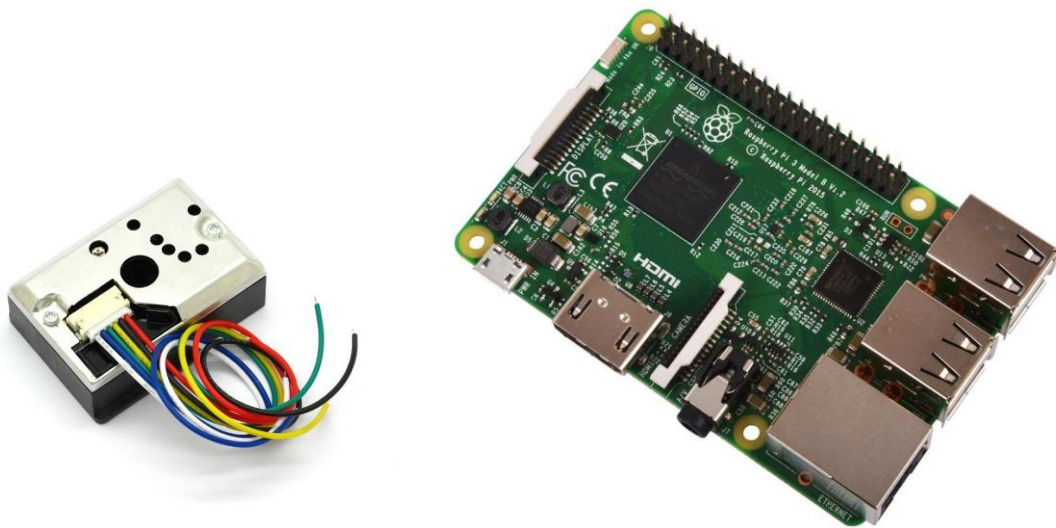
Vivek Choudhary (16CSU427)

Yogesh Kataria (16CSU435)

Objective: The goal is to continuously monitor the room dust samples and trigger the air purification system ON/OFF, and further doing analysis on the data.

Apparatus: Following are the apparatus –

- Arduino UNO
- Raspberry Pi
- Dust Sensor (Sharp GP2Y10)
- Connecting wires.



Quad-core 64-bit Broadcom BCM2837 ARM Cortex-A53 SoC processor running at 1.2 GHz



ATmega328

Some code snippets:

```
float read_dust() {
  /*
   The LED control pin (pin 3 on the SHARP dust sensor module) is active low.
   This means that a LOW ( 0V ) value will turn the LED on and a HIGH ( VCC ) value will turn
the LED off.
  */
  digitalWrite(dust_led, LOW); // turning the led to ACTIVE HIGH
  delayMicroseconds(sampling_time); // delay for pulse mapping [1]

  dust_read = analogRead(dust_in); // read the dust measurement

  delayMicroseconds(delt_time); //[1]
  digitalWrite(dust_led, HIGH); // turning the sensing IR led to LOW
  delayMicroseconds(sleep_time); // [1]
  dust_read = dust_read * (5.0 / 1024.0); // mapped 5 voltage values to 1023 integer value
  dust_read = (dust_read * 0.17) - 0.1;

  delay(500);
  return dust_read;
}

dust_samp = read_dust();
if (dust_samp >= 0.10) { // if density is more than 0.10 then turn of the fan of filter
  analogWrite(fan_out, 255);
}
else { // as the dust factor decreases turn off the fan
  analogWrite(fan_out, 0);
}
```

Work flow: We can brief the workflow in the following ways-

1. The dust sensors continuously monitor the dust.
2. The readings are sent to the UNO
3. From UNO the reading goes to cloud sheets.
4. We then do statistical analysis on the data.

Applications: We can use the technology in the following fields-

1. Gas leakage in a chemical factory: The Idea is to establish a large-scale appliance with gas modulation system which identifies what gas is being leaked or is being used in a certain module, this application has a vast scope of possibility.

2. Smart dust modulation in a room: the air purifiers today sometimes run even if the dust density is not that much fatal, moreover, sending data to cloud for further analysis is not provided, with a bottleneck of price factor.
3. Modularity: Another problem with today's products is the **scalable module**. Those applications are not meant to be expanded and the functionality of the systems cannot be improved and the applications hence is static, but here we can scale it at a much smaller size, we talk about this in the next point.

Further Scalability of the Project: The scalability is defined is idealized in following manner-

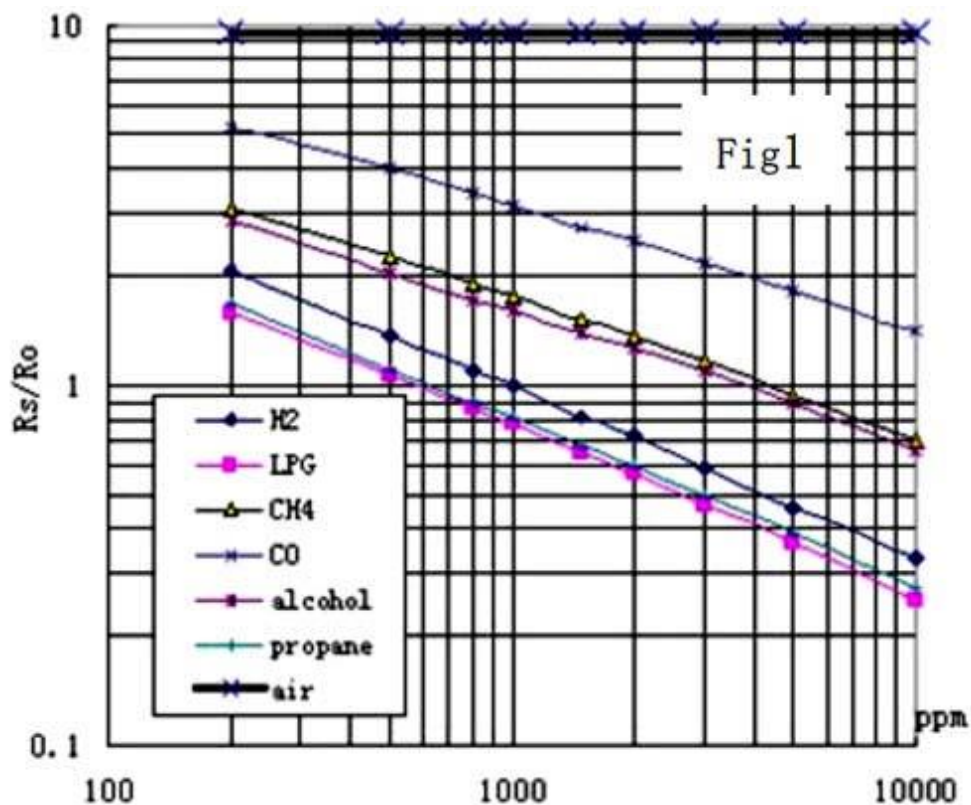
- Detecting the GAS intensity: The idea is to detect and plot the intensity if a specific gas and regulate/Identify the loophole.

We can use the sensors of MQ series, they are designed to detect a range of gases and also comes in the form of a potentiometer to set the sensitivity, following are the available types of the sensors.

Sensor Name	Gas to measure
MQ-2	Methane, Butane, LPG, Smoke
MQ-3	Alcohol, Ethanol, Smoke
MQ-4	Methane, CNG Gas
MQ-5	Natural gas, LPG
MQ-6	LPG, butane
MQ-7	Carbon Monoxide
MQ-8	Hydrogen Gas
MQ-9	Carbon Monoxide, flammable gasses
MQ131	Ozone
MQ135	Air Quality
MQ136	Hydrogen Sulphide gas
MQ137	Ammonia
MQ138	Benzene, Toluene, Alcohol, Propane, Formaldehyde gas, Hydrogen
MQ214	Methane, Natural Gas
MQ216	Natural gas, Coal Gas

MQ303A	Alcohol, Ethanol, smoke
MQ306A	LPG, butane
MQ307A	Carbon Monoxide
MQ309A	Carbon Monoxide, flammable gas

We can also deduce the gas PPM using the datasheet provided –



The graph is a SemiLog graph and can be interpreted as X, Y using the formula –

$$\text{Slope}(m) = \frac{\log y_0 - \log y_1}{x_0 - x_1}$$

Bibliography/References:

Raspberry pi : <https://www.raspberrypi.org/>

Arduino: <https://www.arduino.cc/>

Particle Sensor: <http://arduinodev.woofex.net/2012/12/01/standalone-sharp-dust-sensor/>

Gas Sensor: <https://components101.com/mq2-gas-sensor>

Semilog graph Calibration: <https://msu.edu/course/fsc/441/dvcalc.html>

Github repo: <https://github.com/PaulNicolasHunter/air-quality-analysis>