EE 344 : Electronic Design Lab

Group No. : MON-20

Sanket Potdar 20D070071
Rituraj Chaudhari 20D070065
Simran Tanwar 20D070078

April 28, 2023

Project P-12: IoT enabled air pollution level meters
Faculty Reviewers:
Prof. Gaurav Kasbekar and Prof. V Rajbabu
RA/ Mentor Alotted:
Sravani G. and Chandralekha

Contents

1 Design Details

1.1 Block Diagram

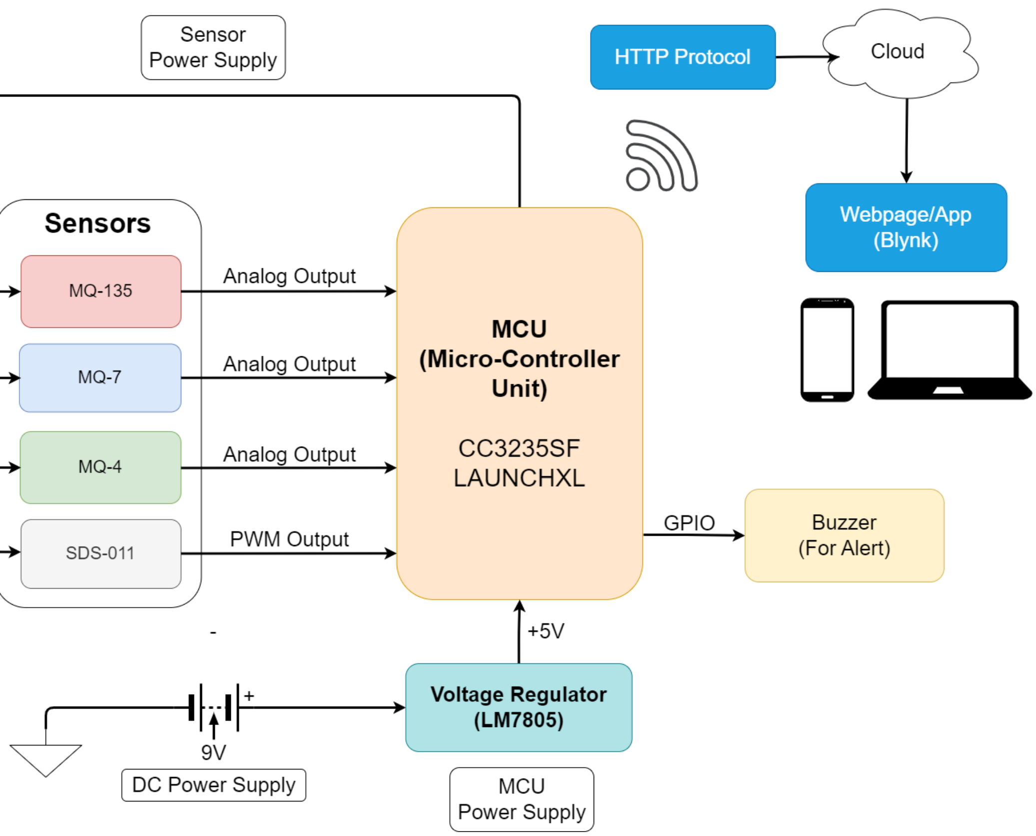


Figure 1: Block Diagram for the project

The block diagram mainly consists of 4 parts - **Sensor, Voltage Regulator Circuit, MCU(Micro-Controller Unit), and cloud setup**.

- **Sensors**: This consists of all the sensors namely the gas sensors MQ-135, MQ-4, MQ-7 and the particulate matter sensor SDS-011. We use the analog inputs for the gas sensors and the PWM pins for SDS-011. All the sensors receive power from the 5V output of the MCU launchpad.

- **Voltage Regulator Circuit**: It is the main power suppy system for our entire circuit. We use a 9V battery for the power source and use LM7805 to step down the 9V to 5V as a source for the launchpad. It also consists of 2 capacitors each of 0.1 uF between the input pin and ground and the output pin and ground of the LM7805 IC acting as decoupling capacitors for our 9V power supply.

- **MCU (CC3235SF-LAUNCHXL and Cloud setup)**: The MCU has ADC inputs that take in analog value from each sensor and then we calibrated all sensors to convert the raw ADC values to PPM values. A buzzer is attatched through GPIO pin of the board to alert the user. These values are then transferred using the inbuilt Wi-Fi module of the board using HTTP protocol to the Blynk server. Cloud setup also consists of the mobile and laptop devices which we use to display our sensor values on the Blynk console.

1.2 Schematic

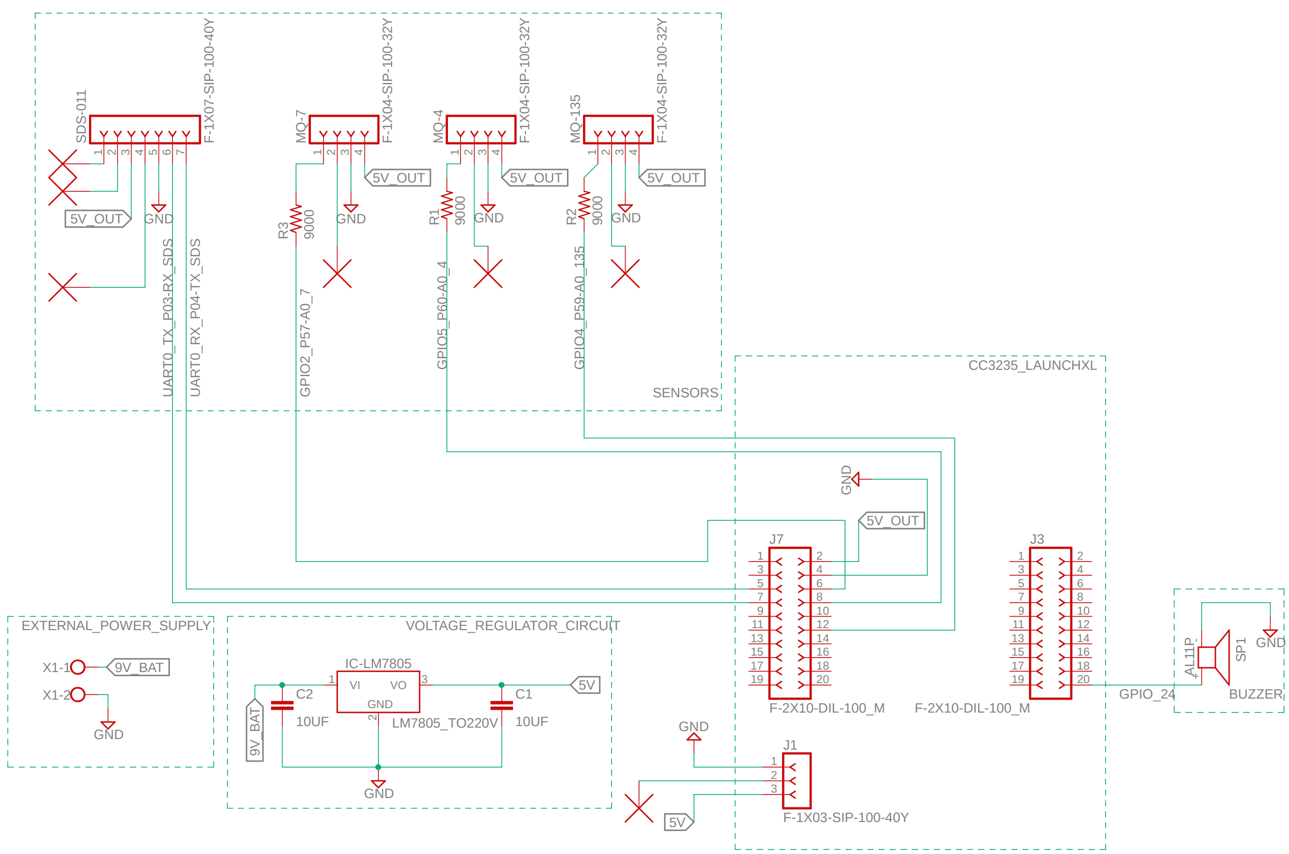


Figure 2: Circuit Schematic

1.3 PCB Design

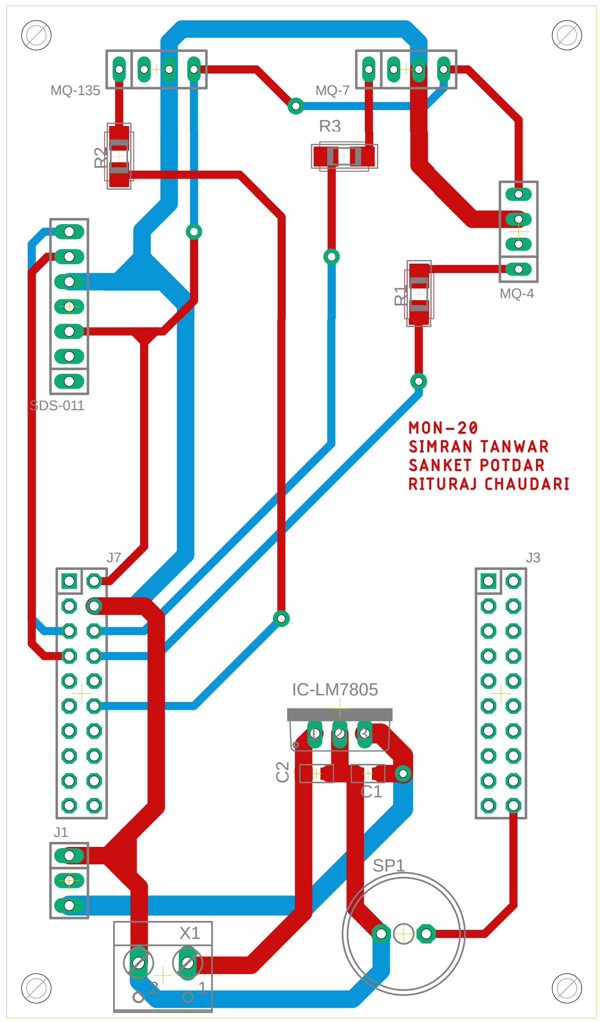


Figure 3: PCB Design

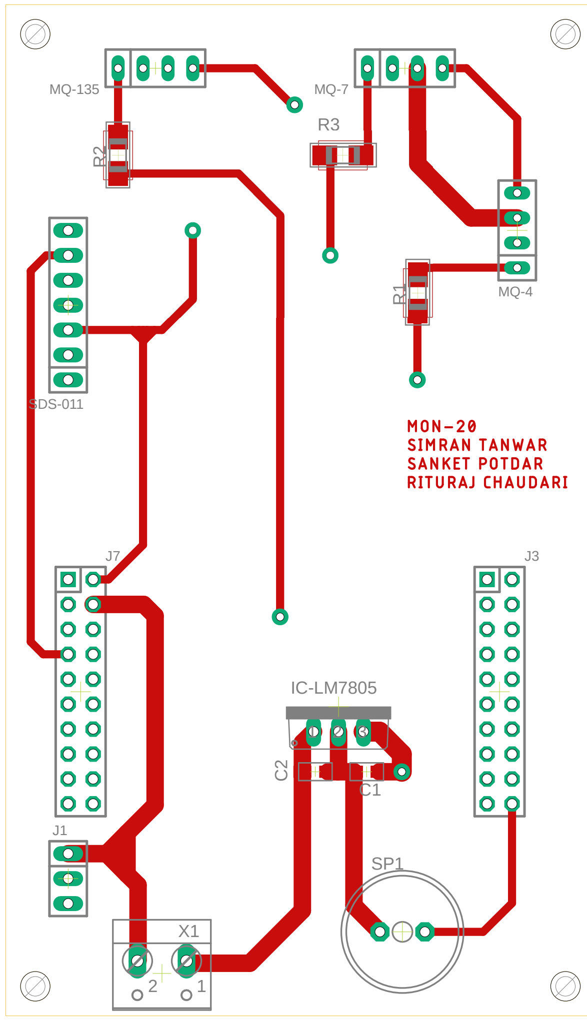


Figure 4: PCB Top Layer

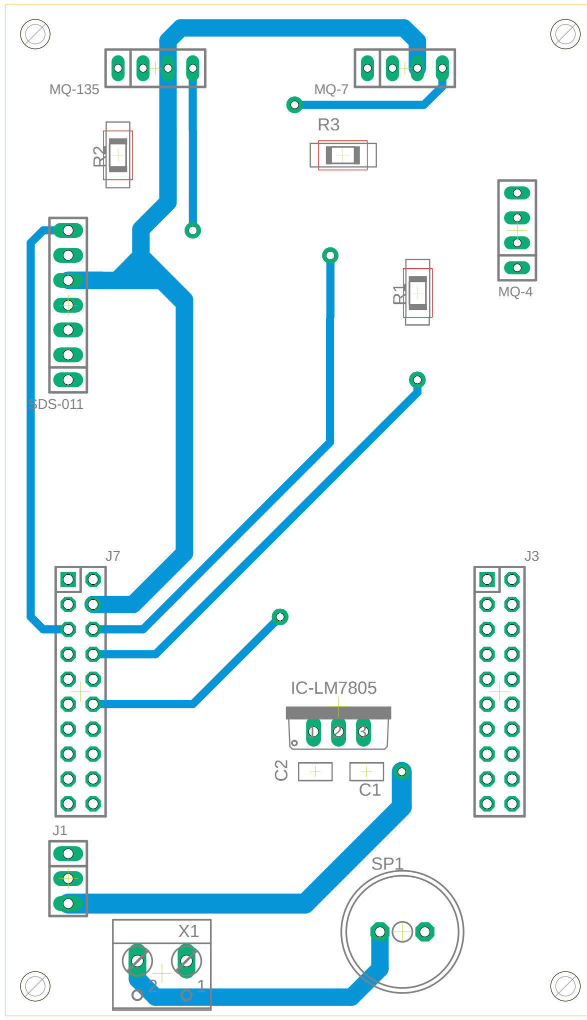


Figure 5: PCB Bottom Layer

PCB Design Explained: PCB design is such that the PCB board contains two 2x10 female pin headers at its bottom which will be used to interface the Launchpad and will be connected to two 2x10 male connectors on the Launchpad from the top. We are also going to use female connectors for interfacing our sensors and input power supply. The voltage regulator circuit is built-in on the PCB board using LM7085, capacitors, and the attached female pin header connected to the 9V battery. We also placed the decoupling capacitor very near the regulator as it protects the IC by filtering out any excessive noise. Also, we increase the width of the power lines as compared to the remaining connecting lines.

1.4 CAD Design

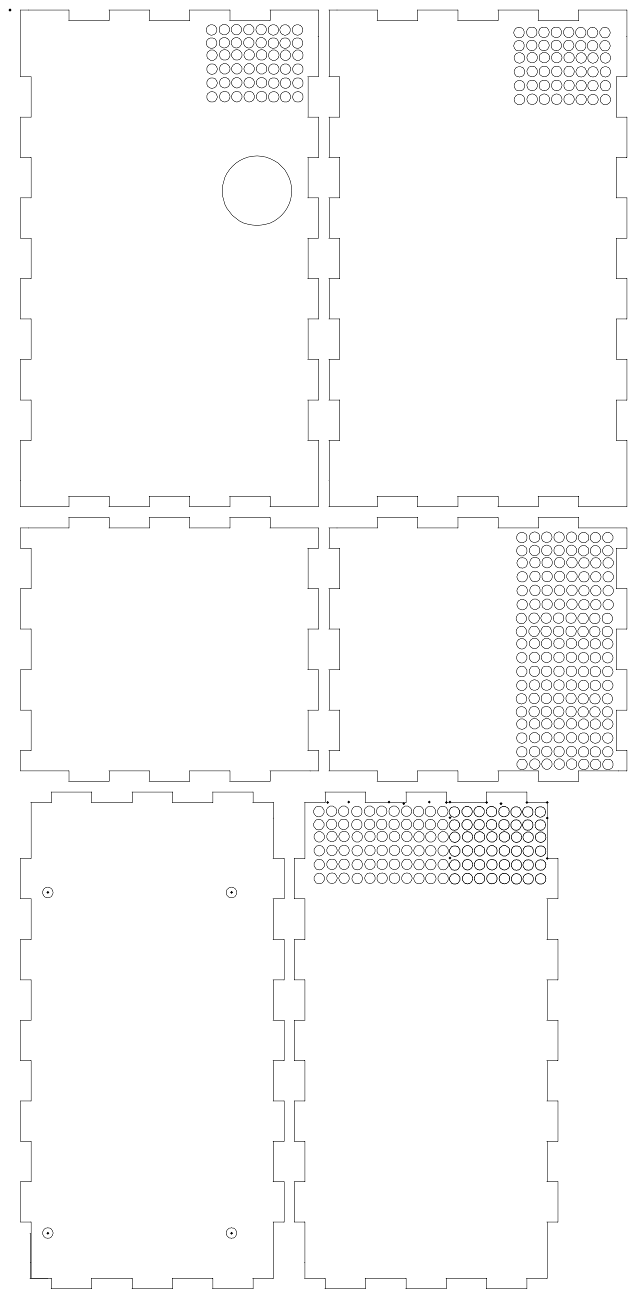


Figure 6: CAD Design Layout

Description

The box was prepared using MakerCase. It consists of a box with finger slots of given dimensions. It was then exported to Fusion 360 where we gave the holes according to our design. We created a mesh with hole size 3mm for air to flow into the sensors and we made another large hole for the MQ-4 sensor to protrude out from the side. Small 4 holes are for the screw attachment of the MCU Launchpad. We then cut the following layout on the acrylic sheet using laser print and assemble the box.

2 Bill of Materials

Sr. No.	Component	Quantity	Price per unit	Total Price	Comments
1.	MQ-135	1	Rs.180	Rs.180	-
2.	MQ-7	4	Rs.190	Rs.190	-
3.	MQ-4	4	Rs.230	Rs.230	-
4.	9V Duracell Battery	2	Rs.285	Rs.570	-
5.	CC3235SF-LAUNCHXL	-	-	-	available in WEL lab
6.	SDS-011	-	-	-	Did not use in project
7.	Audio Source(Buzzer)	-	-	-	available in WEL lab

Table 1: Components Required

3 Results and Observations

3.1 Plots

Here are the snapshots of the sensor value outputs obtained on the Blynk app and web server.

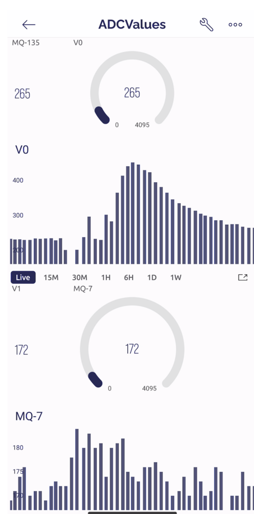


Figure 7: Screenshot from Values obtained on Blynk mobile app

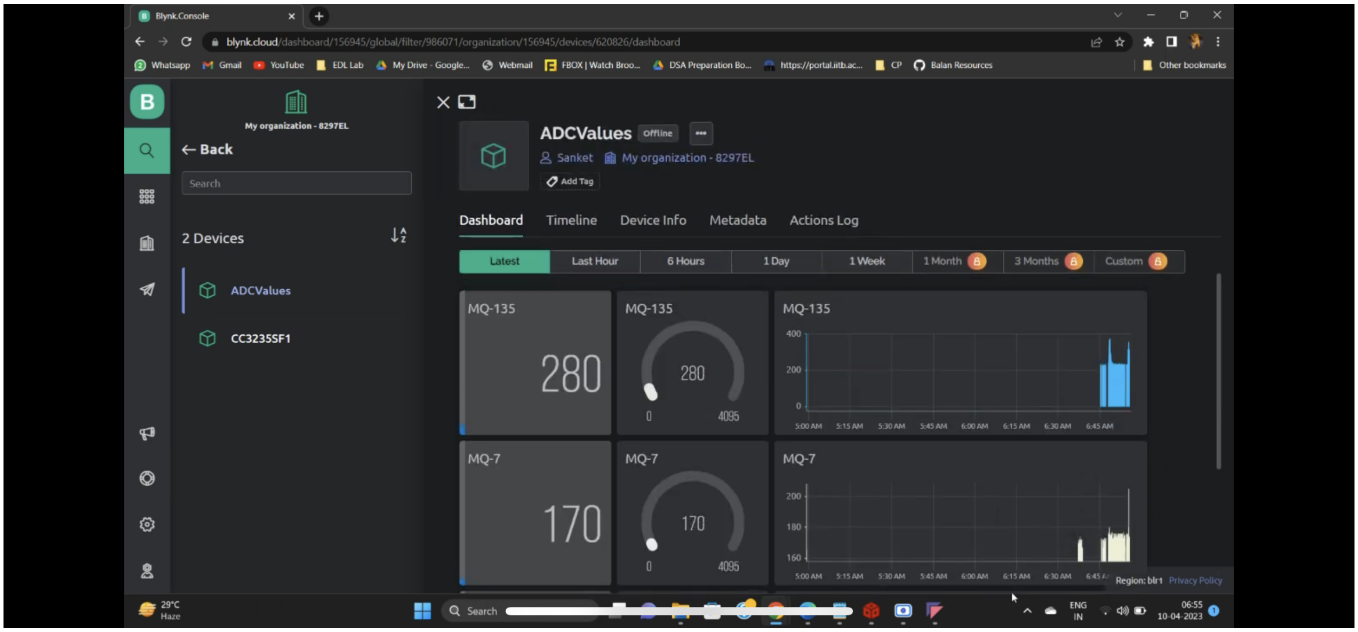
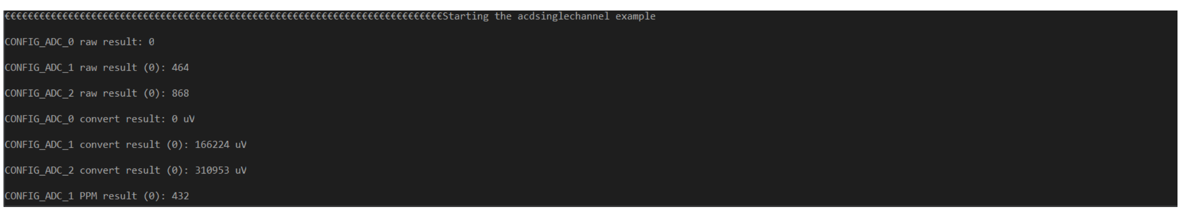


Figure 8: Screenshot from Values obtained on Blynk console on Laptop

3.2 Video Links

- Demo Video
- Blynk web dashboard recording
- Blynk Mobile App Recording

3.3 Screenshots



```
€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€€Starting the acdsinglechannel example

CONFIG_ADC_0 raw result: 0

CONFIG_ADC_1 raw result (0): 464

CONFIG_ADC_2 raw result (0): 868

CONFIG_ADC_0 convert result: 0 uV

CONFIG_ADC_1 convert result (0): 166224 uV

CONFIG_ADC_2 convert result (0): 310953 uV

CONFIG_ADC_1 PPM result (0): 432
```

Figure 9: RAW ADC values and calibrated PPM values obtained on the terminal of CC studio

3.4 Photographs

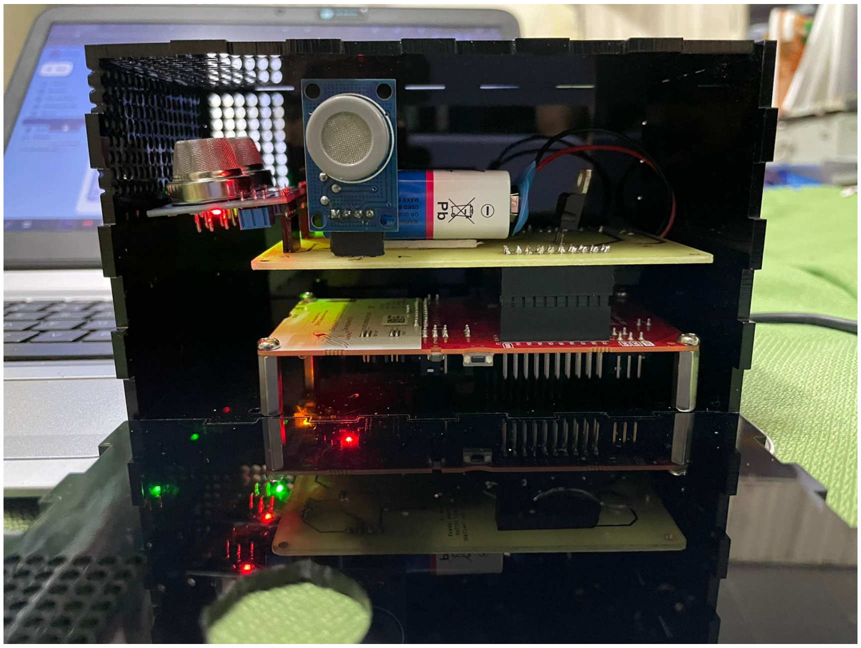


Figure 10: Sensor unit setup - Sensors, battery, PCB and MCU Launchpad

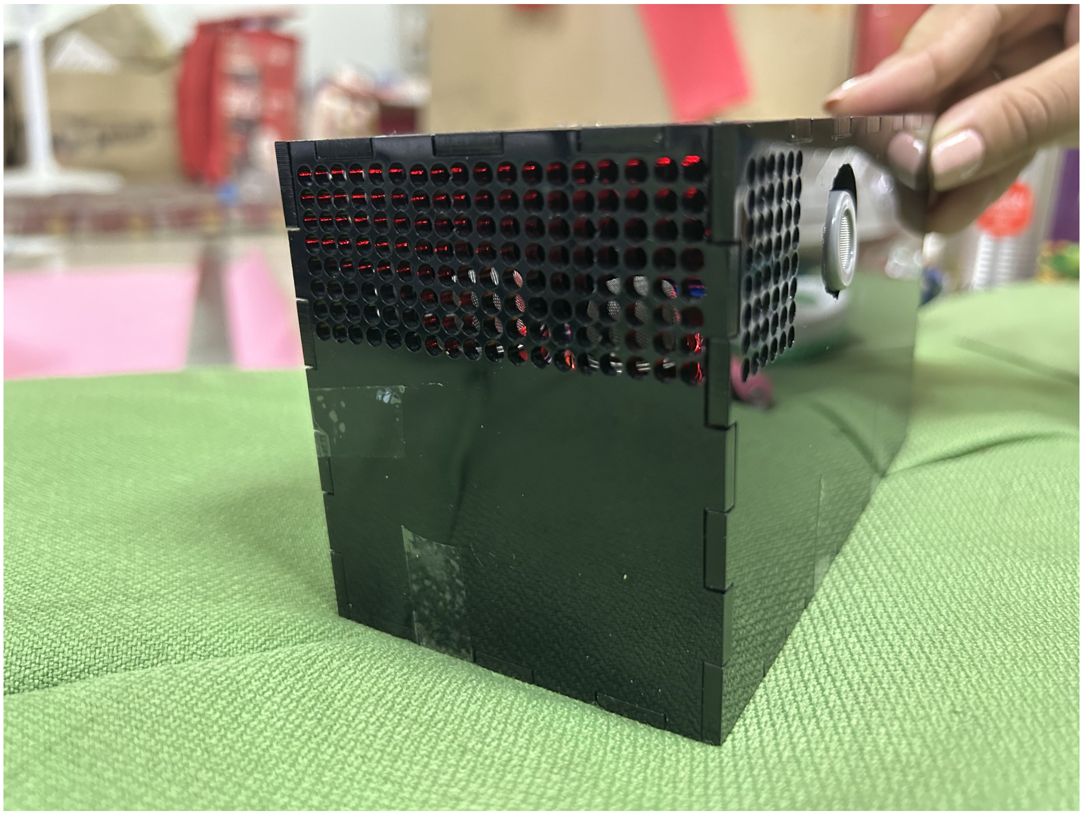


Figure 11: CAD Design

Figure 12: Mobile and Laptop setup
Note: We could check the comment on the quality of air - "SAFE VALUE"

4 Conclusions

We calibrated our sensors and transferred the PPM values over Wi-Fi (we used a portable mobile hotspot in our case) using the HTTP protocol and display it on the Blynk app and Blynk desktop platform. We could power the circuit and MCU using a 9V Duracell battery and transfer the data over Wi-Fi with our whole setup as a black box working independently without the need of connecting to the USB terminal. We could see the live sensor data and could store the data of up to a year in Blynk and show graphs/charts and live values on the Blynk console of both mobile and laptop. It also printed the quality of air, if the PPM values exceed a certain threshold. It shows outputs as - "SAFE VALUE", "HARMFUL FOR LONG", and "HARMFUL VALUE" depending on the thresholds.

5 Future Work

- **Better Battery**: The normal 9V battery did not provide enough constant power for the Wi-Fi module to work. The Duracell battery worked but got drained too quickly. Instead, we could use the 7.4V LiPo battery with a 5200mAh rating or a 12V battery with a 10000mAh rating. These would work for a long duration of time without the need to replace them. The drawbacks to these are that, they have a large size as compared to our current battery which requires us to create a separate battery slot in CAD design and change it. Also, they are a lot costlier. The Lipo battery costs around 3k rs.

- **SDS-011 PPM sensor**: EPA establishes an AQI for five major air pollutants regulated by the Clean Air Act. Of these, the particulate matter PM 2.5 is the most harmful and important. It is an air pollutant that is a concern for people's health when levels in air are high. PM2.5 are tiny particles in the air that reduce visibility and cause the air to appear hazy when levels are elevated. PM2.5 or PM10 is generally the primary pollutant in the air around us.
  The SDS011 can get the particle concentration between 0.3 to 10um in the air. It gives direct digital output via serial protocol at 9600bps and also over 2 PWM channels for PM2.5 and PM10 particles separately. This can be connected to the PWM pins on the board and calibrated accordingly.

- **Calibration in Controlled environment**: Actual calibration involves having a known gas concentration in your environment as your standard for calibration. As, we did't have the exact actual ppm values in our environment, we used the Atmospheric CO2 for calibration. Using a controlled PPM environment for the calibration of sensors could give tremendously better and much accurate results for our PPM values.

- **Analytics part**: Work on the analytics part and design a better algorithm like the AQI algorithm to determine the quality of air and mention how safe it is. For now, we have just used a threshold on the gases, but we could design a complex algorithm depending on PPM values obtained and develop a formula to determine the quality of air. This algorithm may include weighted average, polynomial, or any other functions.

- **Blynk Notifications**: It is possible to customize sounds for notifications coming from Blynk. In-app push notifications from a device triggered by events configured in the Blynk.Console. Each channel can be configured to turn on/off all notifications or use custom sounds, and device event notifications can be configured as critical alerts in some cases. This can be used to alert the user through mobile device in case the quality of air goes critically down and can be harmful.

- **Buzzer**: For now, we did not implement the buzzer part. This could alert any user nearby our sensor unit is alerted when the quality of air is bad. It can be attached using the GPIO which can be coded to give output voltage when PPM exceeds a certain threshold.

- **Blynk Sound**: The web dashboard of our Device has a widget of "Sound and Alert" which can be triggered by a datastream that we can manually set up and can be used to alert the user. This can typically be used in case of a gas leak at home when the user is not present there.

- **Rechargeable Battery**: The battery of our IoT units needs to be replaced when they drain out. There is a limitation to it in the case of inaccessible places like high rises. For the comfort of the user, we could also explore the option of rechargeable batteries, in which case, the battery will receive an external connection from the mains with an

AC-DC adapter circuit suitable for charging our battery.
We could also use mains for our power supply. We first need to downconvert it and change it from AC to DC voltage in a suitable range of around 5V. Mains would be present at almost every place we install in the city - like highrise buildings, roadside lamps, etc. For a backup power source, we can use a battery. In order to implement this, we will need to create an automatic power supply switching circuit which is a type of latching circuit that can be simply created using components like diodes, transistors, and relays. MAX6326 can also be used for efficient noiseless switching from a battery to an external power source and vice-versa.