How to use

CL-AIR

SMART AIR QUALITY MONITOR

ENVIRONMENTAL SENSOR DOCUMENTATION

Purpose of the System

The purpose of CL-Air (Clean-Air) is to measure indoor air quality; in particular, PM2.5 and PM10 particle concentrations, humidity, and levels of CO_2 (carbon dioxide), CO (carbon monoxide), CH_4 (methane) and NH_3 (ammonia). The user will be alerted when these metrics breach preset safety thresholds, so that they may make reasonable adjustments to their environment.

PM (particulate matter) sensors measure the concentration of tiny particles in the air, typically categorised by their diameter. PM2.5 refers to particles with a diameter of 2.5 micrometers or smaller, while PM10 refers to particles with a diameter of 10 micrometers or smaller. These particles can include dust, pollen, mold, and other pollutants, which can penetrate deep into the lungs and even enter the bloodstream, posing significant health risks. This system is particularly beneficial for individuals with respiratory diseases such as asthma and COPD, as well as those with cardiovascular conditions, as these groups are highly sensitive to air pollution. The system is also valuable for monitoring recovery environments for patients with respiratory infections, where air quality and humidity significantly impact healing.

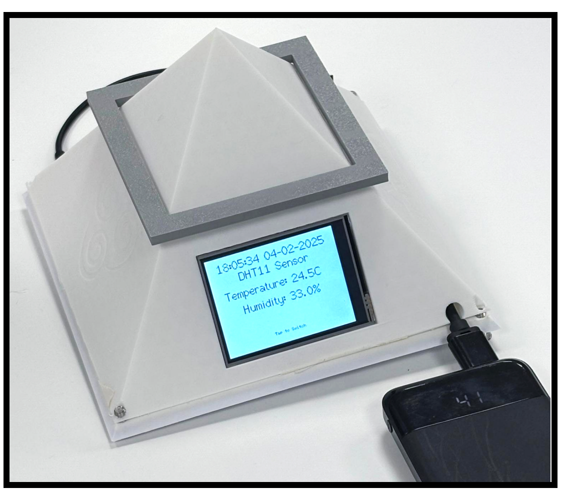

Figure 1: *The air quality monitor displaying readings for temperature and humidity.*

.

List of Components

Components	Supplier	Price
Nova PM SDS011 Laser PM2.5 Sensor x 1	AliExpress	£14.18
2.8 Inch TFT LCD Touch Screen Shield Display Module ILI9341 (240×320) x 1	AliExpress	£3.74
ESP32 x 1	AliExpress	£1.30
DHT11 x 1	AliExpress	£0.82
MQ-135 x 1	AliExpress	£0.82
10kΩ Resistor x 2	AliExpress	£0.65
Charmast 10,000mAh Power Bank x 1	Amazon	£9.99
Elegoo Starter Kit (Wires,Active Buzzer)	Provided by KCL	N/A
Total Cost		**£** 31.50

Table 1: *List of all components used to build the device, along with the supplier and price for each component.*

Block Diagram

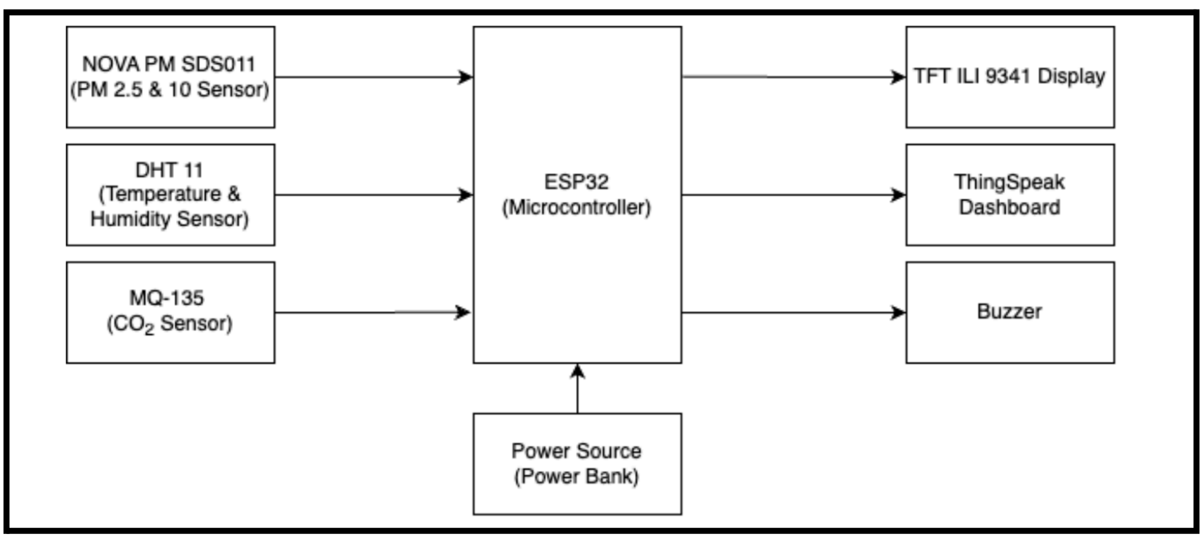

Figure 2*: The block diagram for CL-Air demonstrating general device design and processes. The device is controlled via the ESP 32 microcontroller which takes inputs from three sensors and outputs them to an online dashboard (ThingSpeak), touchscreen display and triggers an active buzzer. A power bank is used to power the microcontroller from which all other components receive power.*

.

Circuit Diagram

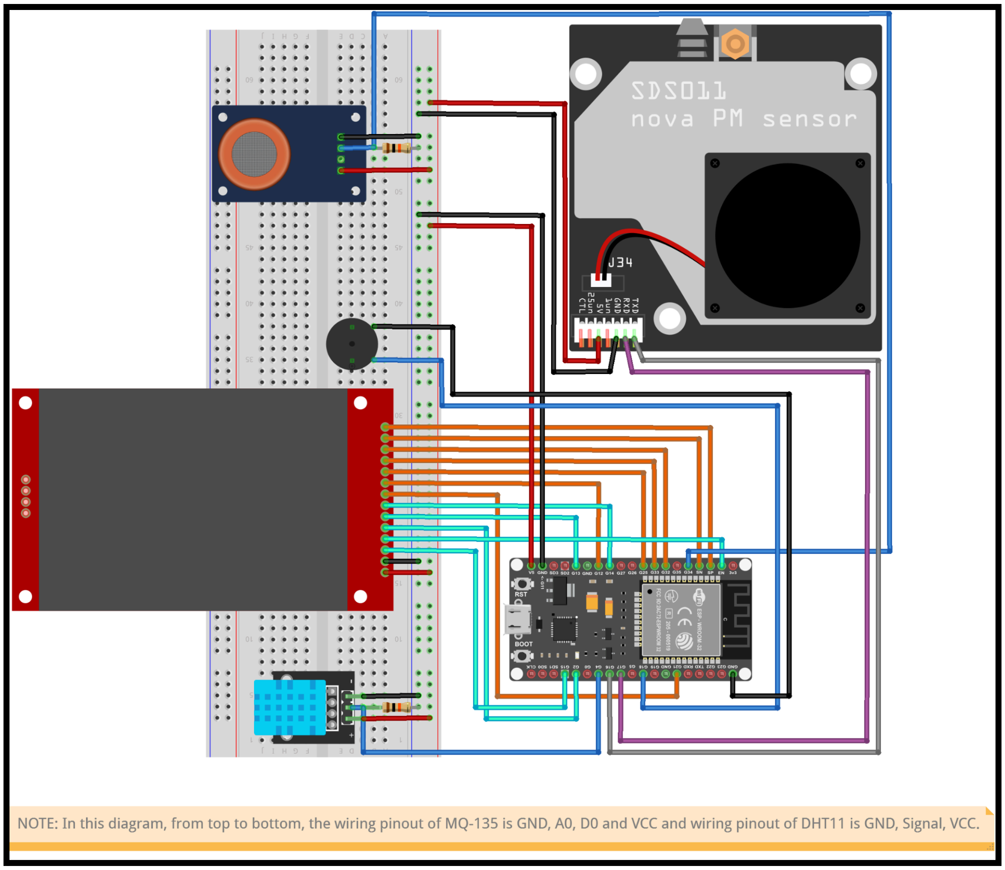

Figure 3: *Circuit diagram for CL-Air showing the electrical connections between components (created on Fritzing)*

Wiring Diagram

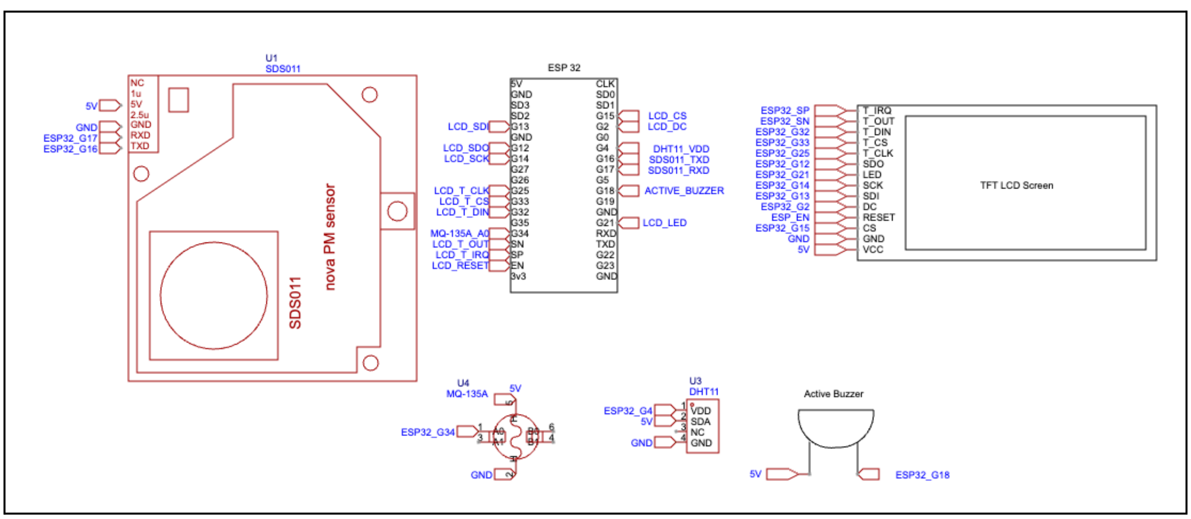

Figure 4: *The wiring diagram for CL-Air showing the specific pin connections between all the components (created on EasyEDA)*

.

Pin Connections

TFT LCD Touchscreen	**ESP32**
T_IRQ	VP
T_OUT	VN
T_DIN	G32
T_CS	G33
T_CLK	G25
SDO(MISO)	G12
LED	G21
SCK	G14
SDI(MOSI)	G13
D/C	G2
RESET	EN
CS	G15
GND	GND
VCC	5V

Table 2: Pin connections between display and ESP 32

Active Buzzer	**ESP32**
POSITIVE	D18
GND	GND

Table 3: Pin connections between passive buzzer and ESP 32

SDS011 Nova PM Sensor	**ESP32**
RXD	G17
TXD	G16
GND	GND
VCC	5V

Table 4: Pin connections between PM sensor and ESP 32

MQ-135 Sensor	**ESP32**
A0 (ANALOG OUT)	G34 *(and to GND through 10kΩ resistor)*
GND	GND
VCC	5V

Table 5: Pin connections between MQ 135 sensor and ESP 32

DHT11 Sensor	**ESP32**
SIGNAL	G4 *(and to GND through 10kΩ resistor)*
GND	GND
VCC	5V

Table 6: Pin connections between DHT 11 sensor and ESP 32

.

System Operations

Data Collection

The Nova PM SDS011 sensor uses a laser scattering method to detect PM2.5 and PM10 levels in the air. Inside the sensor, a laser diode emits a beam of light that interacts with airborne particles. A photodetector then measures the intensity of the scattered light, which corresponds to the concentration of particles in the air. The sensor converts this data into digital signals, providing real-time measurements of PM2.5 and PM10 concentrations in micrograms per cubic meter (µg/m³). Temperature and humidity are measured by the DHT11 sensor using a capacitive humidity sensor and a thermistor to gauge the surrounding air conditions. The MQ-135 sensor detects levels of CO2, CO, CH4, and NH3 by measuring changes in the resistance of its sensing material when exposed to different gases.

Data Processing

The ESP32 microcontroller plays a crucial role in this system. It receives data from all the sensors, processes it, and determines if any of the measured values exceed predefined thresholds. The DHT11 sensor outputs a digital signal, which the ESP32 processes to deliver accurate readings of temperature (in °C) and relative humidity (in %). The MQ-135, on the other hand, outputs an analogue signal, which the ESP32 converts to a digital signal for further processing.

To calculate the concentration of each gas, the sensor's response is compared to pre-defined calibration curves. These curves are represented by the equation: $Rs/R0 = A * C^B$ where A and B are constants specific to the gas being measured. Rs is the sensor resistance in the presence of the target gas and R0 is the sensor resistance in clean air; C is the gas concentration. Below is a graph depicting these curves. The graph shows how the sensor's resistance changes in response to different gases, which is useful for calibrating the MQ-135 sensor.

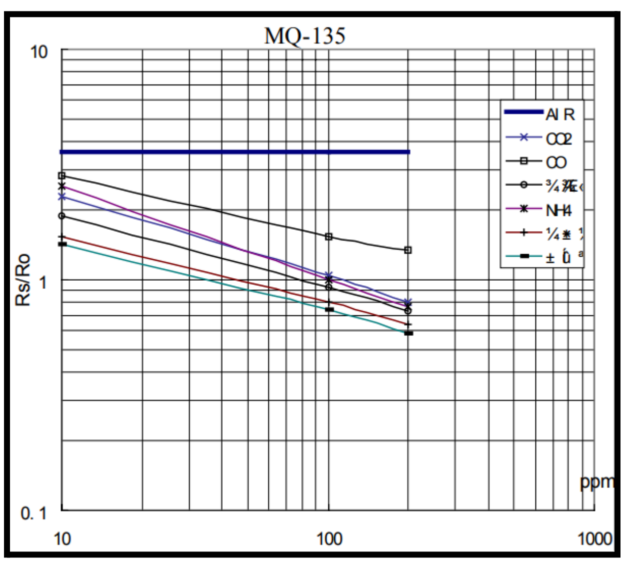

Figure 5: *Sensitivity characteristics of the MQ-135 for various gases (from MQ-135 datasheet).*

.

As mentioned previously, the ESP32 data is used to determine if any of the measured values exceed predefined thresholds. If any threshold is breached an alert is triggered. The details surrounding this are included in the 'Display' section below.

The ESP data is also transmitted to ThingSpeak via a Wi-Fi connection. ThingSpeak is an analytics service that can store the data securely. In the case of this air quality sensor, data is stored in a channel called 'Sensor Data', which contains 8 fields corresponding to the data collected from sensors. Specifically, temperature, humidity, PM2.5, PM10, CO2, CO, CH4 and NH3. Graphical visualisations are created for each field and can be viewed using ThingSpeak. ThingSpeak also enables users to export the data collected as a CSV file with timestamps.

Display

The TFT LCD screen displays real-time sensor readings. Users can interact with it via the touch screen. Tapping the screen changes the display to show a different set of sensor values. If an alert condition is met (the set thresholds for sensor values are breached), the user will be notified. The screen will turn red, reading 'ALERT' and telling the user which sensor value has been breached and a buzzer will sound. If the user acknowledges the alert by tapping the screen, the alert screen will disappear, the buzzer will stop and the sensor values will be displayed as normal. There is a cooldown period of 1 hour, such that if the user has acknowledged the alert, the sensor will not alert them of the same breach for this period of time.

Instructions To Use CL-Air

1. Powering the Device

Connect the environment monitor to a power bank via a wired USB connection. Ensure the Wi-Fi source is connected. Wait for a few seconds while the system initialises and starts collecting air quality data. The screen will turn on and display the sensor data.

2. Display & Readings

The 2.8-inch TFT LCD screen will show real-time measurements of PM2.5 and PM10 levels (µg/m^3) humidity (%), levels of CO_2, CO, CH4 and NH3 (ppm). With this we can detect the real time monitoring of air quality changes. The touch screen allows the user to switch between three different sets of sensor data which are collected by three sensors (SDS011, DHT11 and MQ-135).

.

3. Interpreting the data

The sensor alarm will be set off when warning levels of the parameters are breached. Below is a table displaying the threshold values, which set the sensor off if exceeded. Healthy levels of each metric are also included for reference. Any value between healthy and warning will be considered moderate - not enough to set off the alarm.

	PM2.5	PM10	CO2	CO	CH4	NH3
Healthy Levels	≤ 15 µg/m³	≤ 45 µg/m³	≤ 800 ppm	≤ 9 ppm	≤ 3 ppm	≤ 3 ppm
Warning	35 µg/m³	50 µg/m³	1500 ppm	10 ppm	10 ppm	5 ppm

***Table** 7: Table showing healthy levels of each parameter and warning threshold values.*

4. Taking actions based on parameters

- If PM2.5 or PM10 is too high, avoid opening windows if outdoor pollution is high. Use an air purifier if available.
- If CO_2 levels are too high, open windows and doors for better airflow.
- If humidity is too high or low, use a humidifier (for low humidity) or a dehumidifier (for high humidity).

5. Maintenance and Troubleshooting

- Clean the Nova PM SDS011 sensor regularly to ensure the air can pass it smoothly.
- Avoid placing the device near direct heat sources to prevent sensor breaks.
- If no readings appear, restart the device and check the power source.
- If the buzzer keeps sounding, check air quality and take necessary steps to improve ventilation.

.

Extra Features

Three additional features of this air quality sensor are Thingspeak, the buzzer and touch screen. Details of their use are shown below.

1. **ThingSpeak**

   ThingSpeak collects, stores and visualises sensor data. The ESP32 can send data to ThingSpeak via HTTP or MQTT. Our implementation specifically uses HTTP; this can be seen in the code. Thingspeak's graphical representations can be manipulated to show a different number of results. Timescale and y-axis scale can also be manipulated. Historical data can be accessed and data can be exported as a CSV file, including timestamps.

2. **Buzzer**

   The circuit uses an active buzzer which alerts the user when set thresholds for measured environmental data values have been breached. While the buzzer sounds, a red alert screen appears on the screen. It reads 'ALERT' and notifies the user of the issue - e.g. 'CO2 too high!'. When the user taps the screen, acknowledging the alert, the buzzer stops. Following this, it doesn't sound (for the same breach) for 1 hour.

3. **Touch Screen**

   The air quality sensor uses the 2.8 Inch TFT LCD Touch Screen Display Module, which is able to detect the x and y location of the user's touch as well as pressure. This implementation allows the user to intuitively switch between screens displaying different sensor data by tapping. There are 3 screens which correspond to the 3 different sensors used (DHT11, SDS011 and MQ-135). Each screen displays the sensor name, date, time and specific data collected.

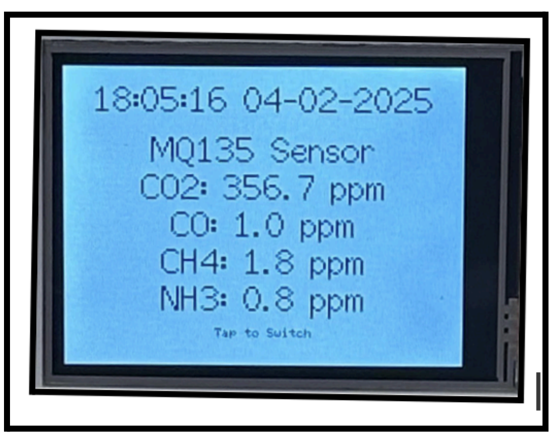

Figure 6: *The touch screen displaying readings from the MQ-135 sensor.*

.

Example Graphs of Captured Data

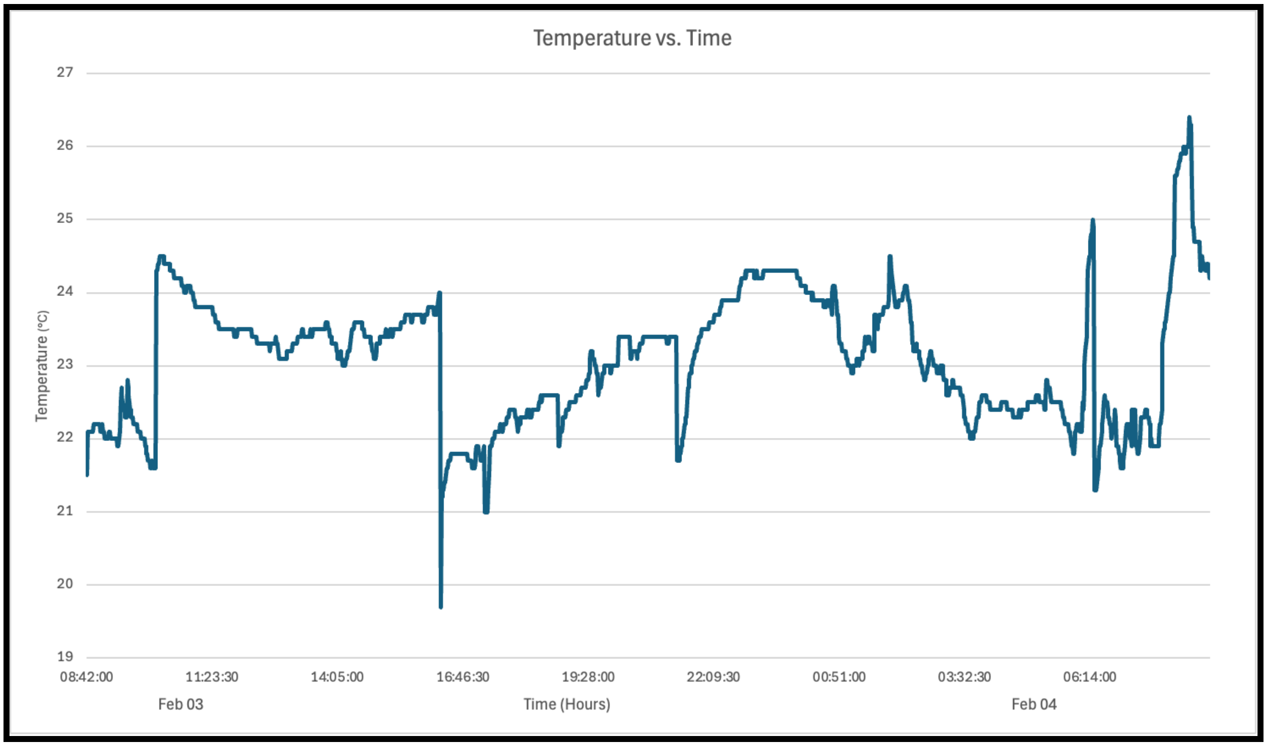

Figure 7: *Visualisation of variation of temperature in an indoor environment over a 24 hour period.*

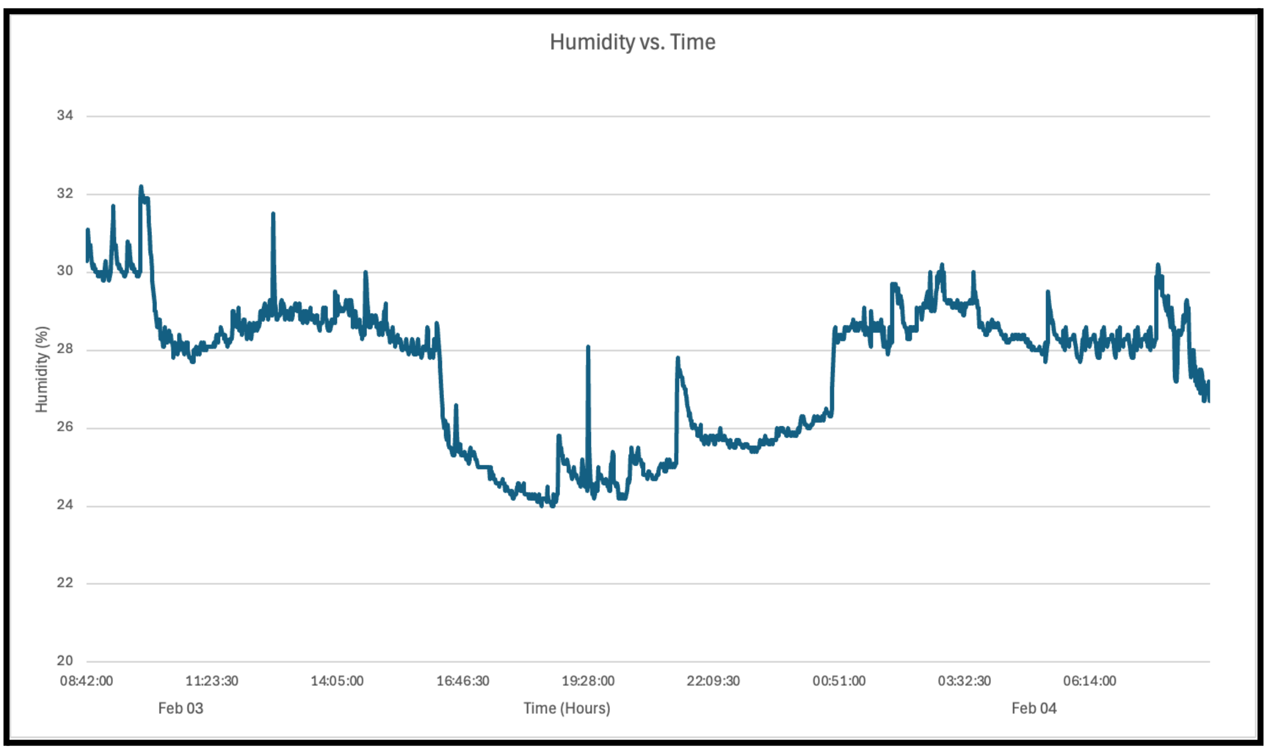

Figure 8: *Visualisation of variation of humidity in an indoor environment over a 24 hour period.*

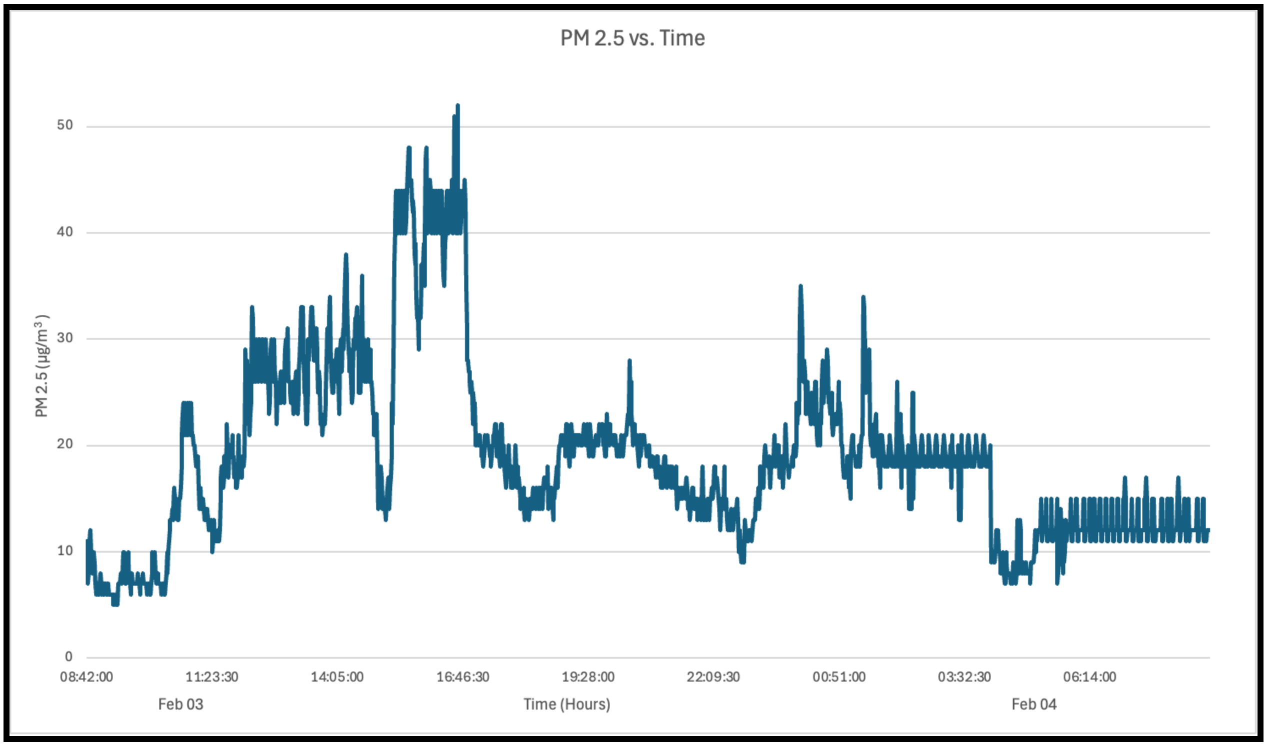

Figure 9: *Visualisation of variation of PM 2.5 in an indoor environment over a 24 hour period.*

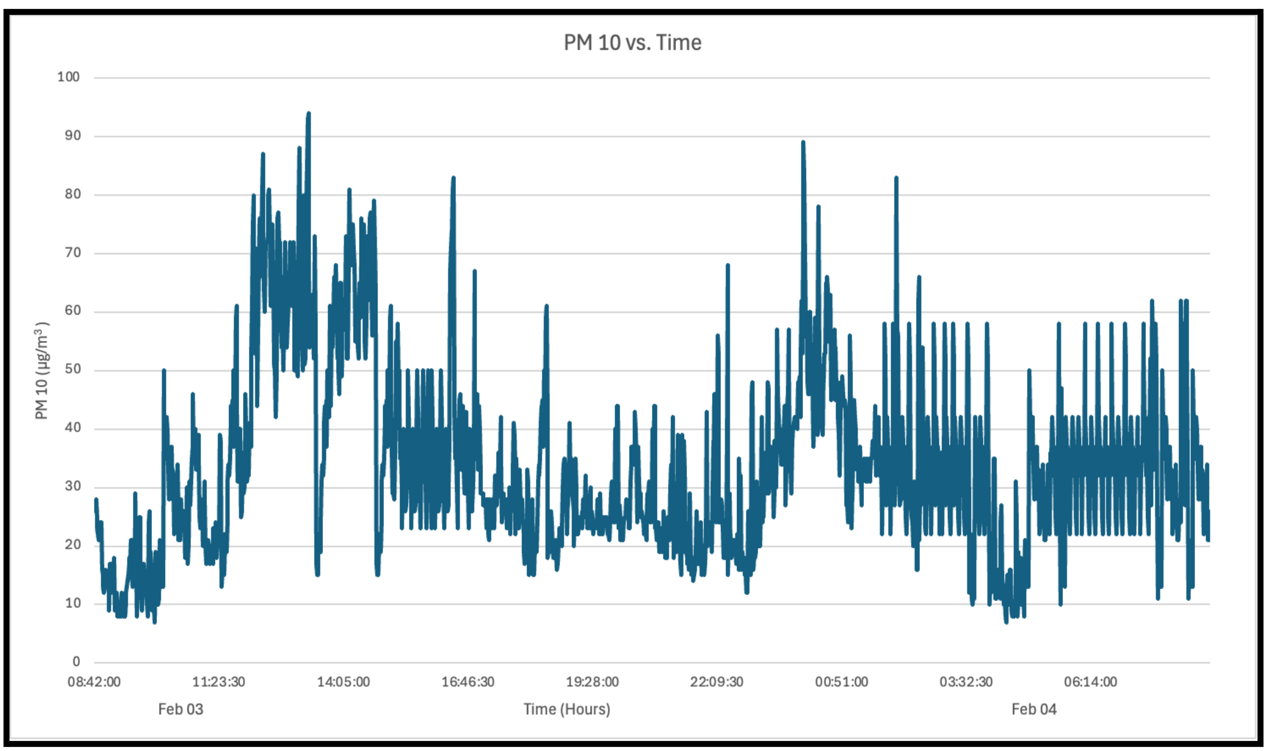

Figure 10: *Visualisation of variation of PM 10 in an indoor environment over a 24 hour period.*

.

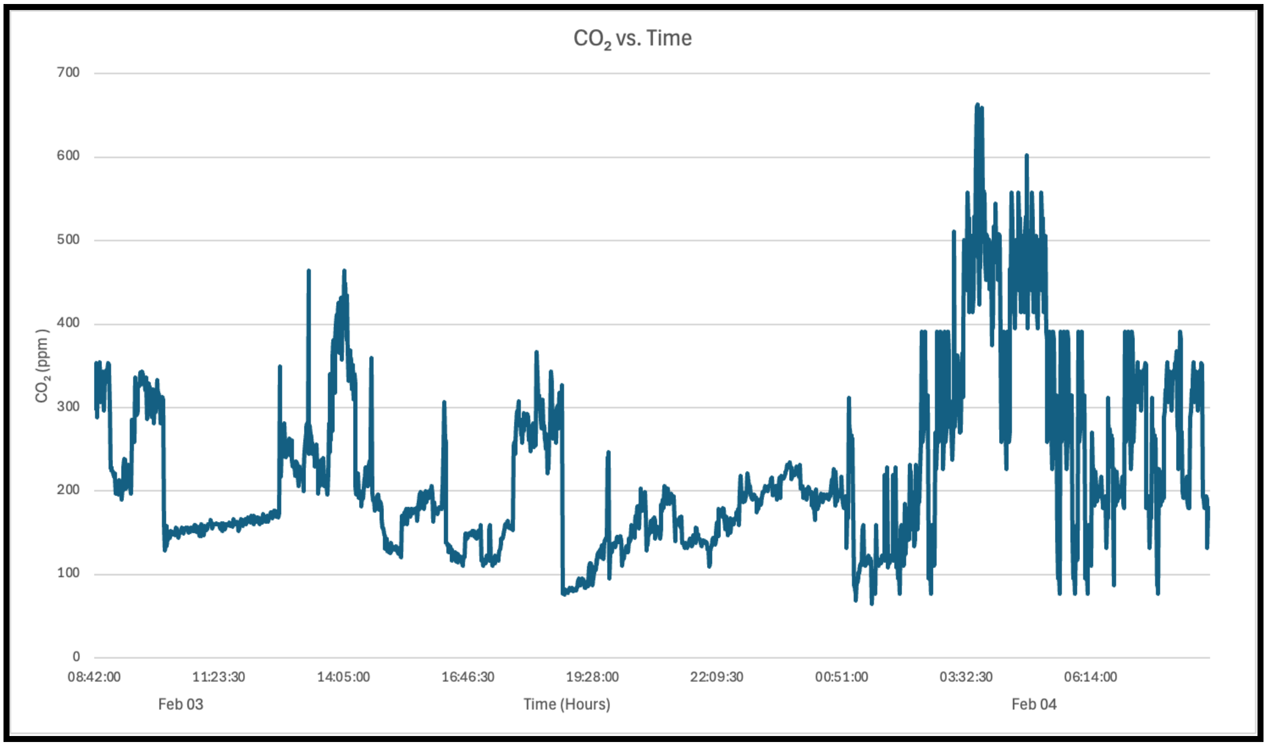

Figure 11: *Visualisation of variation of CO2(Carbon Dioxide) in an indoor environment over a 24 hour period.*

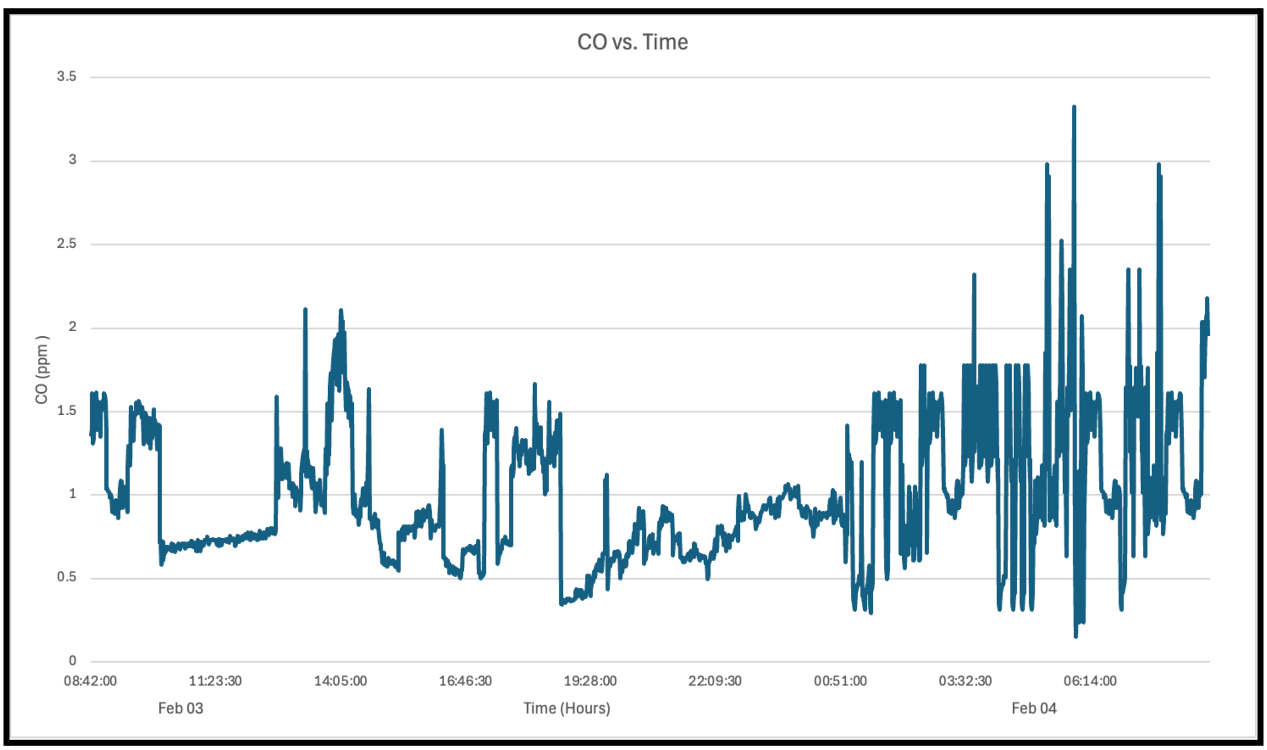

Figure 12: *Visualisation of variation of CO(Carbon Monoxide) in an indoor environment over a 24 hour period.*

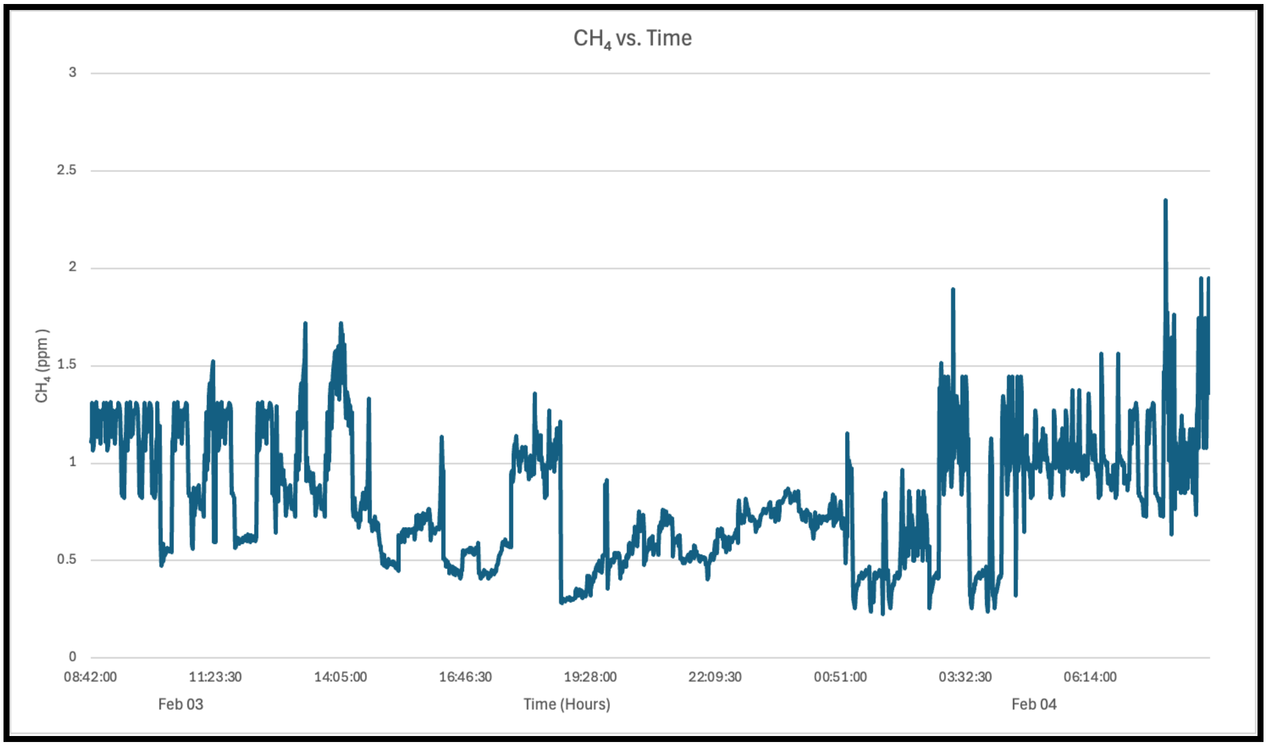

Figure 13: *Visualisation of variation of CH4 (Methane) in an indoor environment over a 24 hour period.*

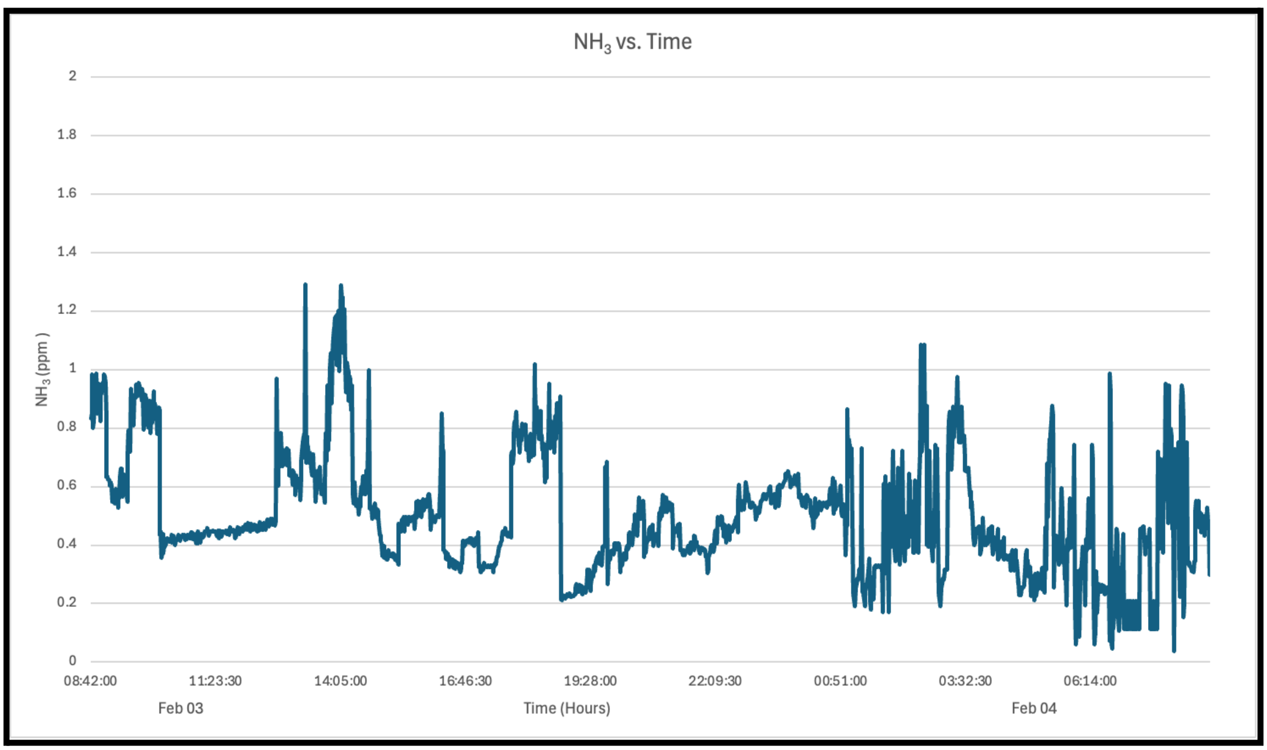

Figure 14: *Visualisation of variation of NH3 (Ammonia) in an indoor environment over a 24 hour period.*

.

Analysis of Visualised Sensor Data

From the visualisations, it is clear that levels of PM2.5 are lower than levels of PM10 on average, as expected due to the fact that PM10 includes both fine and coarse particles but PM2.5 just includes fine particles. There are spikes visible on both the PM2.5 and PM10 graphs where the sensor alarm would have sounded due to thresholds being breached (see interpretation section for threshold details). Potential reasons for the breach could include candles being lit, living near a construction site, etc. Humidity remains fairly stable over a 24 hour period, with readings between 24% and 32% approximately (in line with expected values from multiple sources). Levels of CO, CH_4 and NH_3 do not vary much over the 24 hour period as expected.

Share Point Link

https://emckclac-my.sharepoint.com/:f:/g/personal/k22016336_kcl_ac_uk/Ei8PreGrX4ZIgeRNyDlDrvcBnTlnDoZreqGHgQ1x6UnR8A?e=2B2uhk

Contains: Source Code, Video Demonstration, and 24 Hour Data

References

1. Random Nerd Tutorials (n.d.) *ESP32: TFT LCD Touchscreen Display – 2.8 inch ILI9341 240×320 (Arduino IDE)* https://randomnerdtutorials.com/esp32-tft-touchscreen-display-2-8-ili9341-arduino/
2. Cirkit Design (2024). *How to Use the Nova PM Sensor*. [online] Cirkitdesigner.com. https://docs.cirkitdesigner.com/component/0abf02de-7360-4c66-bc75-b5a82a4d46fb/nova-pm-sensor
3. Cognixia (2015). *How to connect DHT11 Temperature sensor with Arduino*. [online] https://www.cognixia.com/blog/connect-dht11-temperature-sensor-arduino/.
4. Alexander Bessman (2021). *Measuring CO2 with MQ135*. PSLab.io. [online] https://pslab.io/blog/measuring-co2-with-mq135/
5. Futurlec (n.d.). *MQ-135 Gas Sensor Datasheet*. https://www.futurlec.com/Datasheet/Sensor/MQ-135.pdf

.

Appendix

1.1 Task Distribution

Project Plan **-** *Prithvish G.*

Component Sourcing- *Prithvish G.*

Build

Wiring and Circuit Building - *All Group Members*
Source Code and Circuit Design - *Ammarah L. & Prithvish G.*
Sensor Housing Design - *Prithvish G.*
3D Printing of the Housing - *Prithvish G. & Hanyu Zhu*

'How To' Documentation

How to Introduction - *Ammarah L. & Sehwan*
List of Components, Suppliers & Prices - *Ammarah L.*
Block Diagram - *Sehwan*
System Operation - *Ammarah L.*
Circuit Diagram - *Ammarah L.*
Wiring Schematic - *Sehwan*
Pin Connections - *Ammarah L.*
Usage Instructions - *Sehwan & Hanyu Zhu*
Extra Features - *Ammarah L.*
Document Formatting- *Prithvish G.*

Data Collection

ThingSpeak Setup - *Prithvish G. & Ammarah L.*
Example Captured Data Graphs from CSV - *Sehwan*

Video Demonstration **-** *Prithvish G. and Hanyu Zhu*