DEPARTMENT OF PHYSICS

Determination of Particulate Matter using Near Space Balloon Payloads

PROJECT REPORT

Submitted by,

Sarath Krishnan G   200011024031

Under the Guidance of

Dr. Manesh Michael   Dr. Binu Kumar
Assistant Professor, Department of Physics,   Postdoctoral Research Fellow,
Bharata Mata College, Thrikkakara   Indian Institute of Astrophysics
Visiting Associate, IUCCA, Pune   Bangalore

In partial fulfilment of the requirements for the degree of

Master of Science in Space Science

2020-2022

DEPARTMENT OF PHYSICS

CERTIFICATE

Certified that the minor project work titled "Determination of Particulate Matter using Near Space Balloon Payloads" is carried out by Mr. Sarath Krishnan G (200011024031) who is a bonafide student of Bharata Mata College, Thrikkakara in partial fulfilment for the award of degree of Master of Science in Space Science of the Mahatma Gandhi University, Kottayam the year 2020-2022. It is certified that all corrections indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirement in respect of project work prescribed by the institution for the said Degree.

Signature of Internal Guide   Signature of External Guide

Dr. Manesh Michael   Dr. Binu Kumar

Signature of HoD   Signature of Principal

Dr. Anu Philip   Dr. Johnson K M

Ref: SPG/PCL/N.07/09/22

CERTIFICATE

To whomsoever it may concern

This is to certify that Sarath Krishnan G (Reg no: 200011024031), MSc Space Science student of Department of Physics, Bharata Mata College, Thrikkakara, Kerala (Affiliated to Mahatma Gandhi University, Kottayam) has completed project work at the Indian Institute of Astrophysics, Bangalore during   March 2022 to August 2022.

He has worked on the project titled "**Determination of Particulate Matter using Near Space Balloon Payloads**". This project was aimed to design and develop a Particulate Matter counter payload for hosting onboard the High Altitude Balloon platform. He has designed developed and tested the payload on the ground as well on a tethered balloon platform. He has conducted numerous land surveys and collected sufficient data for calibrating the system as well as for a quantitative decision making. During the course of the project he demonstrated excellent design and problem solving skills. He also had a self-driven attitude to venture into new arena of contemporary research in the field of near space payload developments. His performance was par excellence.
We wish him best of luck for all his future endeavors

26/09/2022
Bangalore

Binukumar Gopalakrishnan
Post-Doctoral Research Fellow
Indian Institute of Astrophysics
Koramangala, Bangalore-560034
Email: binukumar@iiap.res.in
Mob: 9035903915

Affiliated to MG University, Re-accredited by NAAC with 'A+' Grade, ISO 9001-2015 Certified

DECLARATION

I, Mr. Sarath Krishnan G (200011024031 student of fourth Semester MSc Space Science, Department of Physics, Bharata Mata College, Thrikkakara, hereby declare that the project titled entitled "Determination of Particulate Matter using Near Space Balloon Payloads" has been carried out by us and submitted in partial fulfilment of fourth semester requirements in Master of Science in Space Science of Mahatma Gandhi University, Kottyam during the year 2020 - 2022.

Further we declare that the content of the dissertation has not been submitted previously by anybody for the award of any degree or diploma to any other university.

We also declare that any Intellectual Property Rights generated out of this project carried out at IIA will be the property of Bharata Mata College, Thrikkakara and I will be one of the author of the same.

Place: Thrikkakara

Date: 28/09/2022

      Name   Signature

Sarath Krishnan G (200011024031)

ACKNOWLEDGEMENT

ABSTRACT

Near Space Balloons (NSBs) are unmanned aerial balloons filled with lighter than air gases that fly up to the altitudes of 25-35 km above sea levels. NSBs offer exclusive advantages like simplistic design, flexible flight scheduling, and a cost-effective way of reaching the edge of space to perform distinguished experiments. However, they possess major disadvantages like unstable airframe dynamics, poor orientation control, and payload recovery issues.

The particulate matter is the particle pollution which is a mixture of solid particles and liquid droplets found in the air. The PM2.5 is the fine particle which is mostly seen in the atmosphere. This particle is considered more harmful to humans.

This study mainly focuses on building and launching a payload to measure the concentration of Particulate matter present in the atmosphere. At low altitude, the particle changes drastically. The main objective of our project is mainly to detect the concentration of PM2.5. The instrument counts the particle of size below 2.5µm. The change in value is evaluated in both air and ground level. With the help of Arduino microcontroller, an instrument is designed to measure the particulate matter. The required datas are collected by conducting ground survey, tethered flight and actual flight. This datas from the surveys are used to plot and evaluate the particle count. From the plot, the variation of particulate matter with different parameters at various conditions are studied and interpreted.

Contents

Chapter 1

Chapter 2

Chapter 3

Chapter 4

CHAPTER 1

1.1 <u>INTRODUCTION</u>

The high-altitude balloons are used for collecting atmospheric data for long periods that can reach beyond the majority of thick atmosphere, past the region of clouds and airflow that may interfere with measuring data. As such, they provide the perfect opportunity to measure weather data.

1.2 <u>HIGH-ALTITUDE BALLOON</u>

High-altitude balloons are without crew members, these balloons are usually filled with helium or hydrogen gas. Because helium gas is expensive, we usually use Hydrogen. balloons once filled with the gas and released through the stratosphere. Generally above sea level, they attain an altitude between 37 and 18 km. A balloon name was BU60-1. It reached altitude of 53.0 km in 2002. It is a record altitude. Above sea levels serial balloons filled with lighter than air gases that fly up to the altitudes of 25-50 km. Usually these are filled with hydrogen or helium gas and expand as they ascend through the earth's atmosphere.

It generally contained electronic components such as cameras, satellite navigation systems , GPS receivers , radio transmitters.

The high-altitude balloon is a weather balloon in in common type. Balloons are launched in to the near space. In the earth atmosphere area, which named as Armstrong limit where pressure reduces to the point that a living organism cannot survive without a Karman line and pressurized suit. In that place astrodynamics must take over from aerodynamics to maintain flight.

1.2.1   **USES**

  As research balloons, we use high-altitude balloons. by hobbyist, and for educational purposes. They have so many common uses including atmospheric and climate research, meteorology, collection of imagery from near space amateur radio applications and submillimeter astronomy.

High-altitude balloons have major role in space tourism and telecommunications. 2 infinity and world view Enterprises are private companies which involve to develop both commercial

purposes and crewed high-altitude balloons for scientific research and tourism. The applications of high-altitude balloons is that communication relays.

1.2.2 <u>**MOTION OF BALLOON**</u>

The balloon travels through the upper atmosphere, the external pressure on the balloon decreases with altitude. Therefore, the balloon's surface area is gradually increases. The streamlines of airflow get disturbed and its resistance to wind causes it to spin or rotate as it rises. The payload airframe and the parachute have a contribution to these rotations too, payload also experiences pendulum motion and rotation about its suspension-like axis while in flight up to burst altitude.

1.2.3 <u>**DISADVANTAGES**</u>

The balloons have advantages and disadvantages. The major disadvantages include unstable airframe, poor orientation control, poor flight control, weight &power constraints, payload recovery issues, and more. This project is mainly concentrated to solve the issues regarding the instability of the payload airframe and its poor orientation control, also taken into consideration the weight and constraints.

1.3 <u>**ARDUINO**</u>

The Arduino environment gives an opportunity for beginners who have no software or electronics experience, by using Arduino, we can build objects that can respond to control light, sound, touch, and movement. Not only light sculptures, games, interactive furniture, and even interactive clothing but also Arduino has been used to create an amazing variety of

things, including musical instruments robots.

Educational programs, particularly Arduino used by many artists and designers who want to easily and powerfully create prototypes but do not need understanding of the technical details behind creations. ,In the case of nontechnical people, the software includes plenty of example code to demonstrate how to use the Arduino board's various facilities

1.3.1   ARDUINO SOFTWARE

The Arduino software program is called sketches. Using the Arduino integrated development environment (IDE) it can be created. The Integrated development environment allows converting this into instructions and writing and editing code that Arduino hardware understands. The IDE

also transfers those instructions to the Arduino board.

1.3.2 <u>**ARDUINO HARDWARE**</u>

The code executed in the Arduino board. The board can only respond and control to electricity. The board contains various types of components including sensors that convert some aspects values of the physical world to electricity management.   The sensors include ultrasound distance sensors, accelerometers, and switches. The board can sense it or actuators. Actuators are things like lights, displays, LEDs, motors, and speakers.

There are different types of highly functional boards that we can use for software Arduino and the Arduino contain a particular type USB port that gives connectivity and power   for uploading the software in to the Arduino board.

1.3.3 <u>**Arduino Uno Pin out Configuration**</u>

	Pin Category	**Pin Name**
I.		
•	power	Vin, 3.3V, 5V,GND
•	Reset	Reset
•	Input/output Pins	Digital Pins 0 - 13
•	Analog pins	A0 – A5
•	Serial	0(Rx),1(TX)
•	TWI	A4 (SDA), A5 (SCA)
•	SPI	10 (SS),11 (MOSI),12 (MISO),13
•	External Interrupts	2, 3
•	Inbuilt LED	13
•	PWM	3, 5, 6, 9,11
•	AREF	AREF

I. **Details**

- Vin: when using an external power source Input voltage to Arduino.

- 3.3V : Maximum current draw is 50mA. by on-board voltage regulator 3.3V supply generated.

- 5V : Regulated power supply used to power other components on the board and microcontroller.

- Reset : Resets the microcontroller.

- GND : ground pins.

- A0 – A5 : Used to provide analog input in the range of 0-5V

- External Interrupts (2,3) : To trigger an interrupt.

- 0(Rx), 1(Tx) : Used to receive and transmit TTL serial data.

- PWM (3, 5, 6, 9, 11) : Provides 8-bit PWM output.

- Digital Pins (0 - 13) : Can be used as input or output pins.

- Inbuilt LED (13) : To turn on the inbuilt LED

.

- SPI : Used for SPI communication

- TWI : Used for TWI communication.

- AREF   : To provide a reference voltage for input voltage.

1.3.4   <u>**Arduino Uno Technical Specifications**</u>

- Microcontroller Recommended Input Voltage: 7-12V

- DC Current on I/O Pins: 40 mA

- Operating Voltage: 5V

- Microcontroller: ATmega328P – 8-bit AVR family Input Voltage Limits: 6-20V

- Digital I/O Pins: 14 (6 provide PWM output)

- Analog Input Pins: 6 (A0 – A5)

- SRAM: 2 KB

- DC Current on 3.3V Pin: 50 mA

- EEPROM: 1 KB

- Frequency (Clock Speed): 16 MHz

- Flash Memory: 32 KB (0.5 KB is used for Bootloader)

1.3.5   <u>**major Arduino Boards**</u>

Arduino Nano, Arduino pro Minni, Arduino Mega, Arduino Due, Arduino MKR1000 Wi_FI
Board, Arduino Leonardo

Overview

The version Uno of Arduino contain microcontroller board based on 8- bit ATmega328p microcontroller. Along with AT mega 328p, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. these components helps the microcontroller Arduino UNO had 14 digit input /output pins (PWM output 6 pin), 6 an ICS header, a USB connection ,a reset button, A power barrel jack, and analog input pins   .

To use Arduino Board

- plug the board by using USB port connection with your computer

- checking of the green LED which use as a power indicator on the illuminates

- In the center of the board contain an orange LED which flash on and off

- to prepare an Arduino sketch by using the integrated development environment (IDE)

- sketches which define what the board will do

- The sketches have editor area which helps to edit code and view

- by using the example blink, the first check led os blinking

- compelling means that needs to be converted by an Arduino controller chip

18

- first upload the blink sketch and run it

- using the USB port, connect Arduino board to computer

- cable load the Blink sketch into IDE and a next sketch the name od board

- we would get a drop-down list of available serial ports on   computer

- Each machine would have a different type of combination of serial parts

- A sketch must create and save

- we create a sketch and save it. to launch new sketch we must go through editor window

- we will go to the File menu and select new

```
const int led Pin = 13; // LED connected to digital pin 13


void setup ()


{


pin Mode ( led Pin, OUTPUT);


}


void loop ()


{
```

```
digital Write (led Pin, HIGH); // set the LED on


 delay (2000); // wait for two seconds


 digital Write (led Pin, LOW); // set the LED off


 delay (2000); // wait for two seconds


 }
```

- To upload a code, we need click on the upload button

The IDE does not allow the spaces

the IDE will automatically replace these with valid characters

the Arduino files with extension. pde

1.3.6 <u>Communication</u>

Arduino can be used to communicate with a computer, another Arduino board, or other microcontrollers. The AT mega 328Pmicrocontroller provides UART TTL(5 V) serial communication which can be done using digital pin 0 (Rx)and digital pin 1 (TX). An ATMEGA16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The AT mega 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on windows, a.inf files are

required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. There are two Rx and Tx LEDs on the Arduino board which will flash when data is being transmitted via the USB -to- serial chip and USB connection to the computer /A software serial library allows for serial communication on any of the UNO's digits pins. The AT mega 328p also supports SPI communication and 12C. Including the Arduino software wired library which help to simplify the 12C bus.

Chapter 2

2.1   Particle Counters

A particle counter is a sensor or an instrument used to detect and count the fine physical particle. The nature of counting is based on light scattering. A high intensity light source is used to make brighter the particles as they pass through the detection chamber. When the particle passes through the light source, the photo detector measures the dispersion of reflected light. This measurement is then converted to PWM output signal. Particle counters are available in three categories. They are aerosol, liquid and solid. Aerosol particle counter is most commonly used.

2.1.1   DSM501A

Model – DSM501A

Make - SAMYOUNG S & C co.

We have used SAMYOUNG's DSM501A model dust sensor module in this project. This dust sensor module is a compact sized particle density sensor. This dust sensor is designed to detect

the particle of the size from 1.0 micrometer to 2.5 micrometer, like cigarette smoke, house dust, tick, spore and pollen that are known cause for respiratory disease and allergy. Particles of up to 10 micrometer can be detected using this dust sensor.

2.1.2   Data Sheet

Data sheet of a sensor gives the complete details about the sensor module that includes dimensions, electrical characteristics, block diagram, components, pinout description, pinout diagram etc.

2.1.3 Dimension
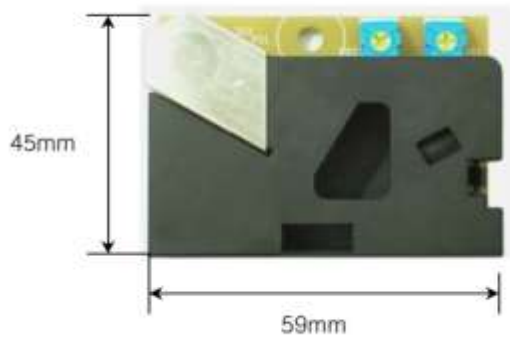
2.1.4   Electrical characteristics

- Supply voltage   : DC5V±10%
- Power consumption   : 90mA
- Operating temperature range : -10~ +65^{0}C
- Operating humidity range   : 95%RH or less
- Storage condition   : -20~ +80^{0}C
- Dimension   : W59 * H45 * D20 (mm)
- Detectable particle size   : 1µm(minimum) to 2.5µm(maximum)
- Output signal   : PWM (pulse width modulation)
- Time for stabilization   : 1 minute after power turned on

2.1.5   BLOCK DIAGRAM

23
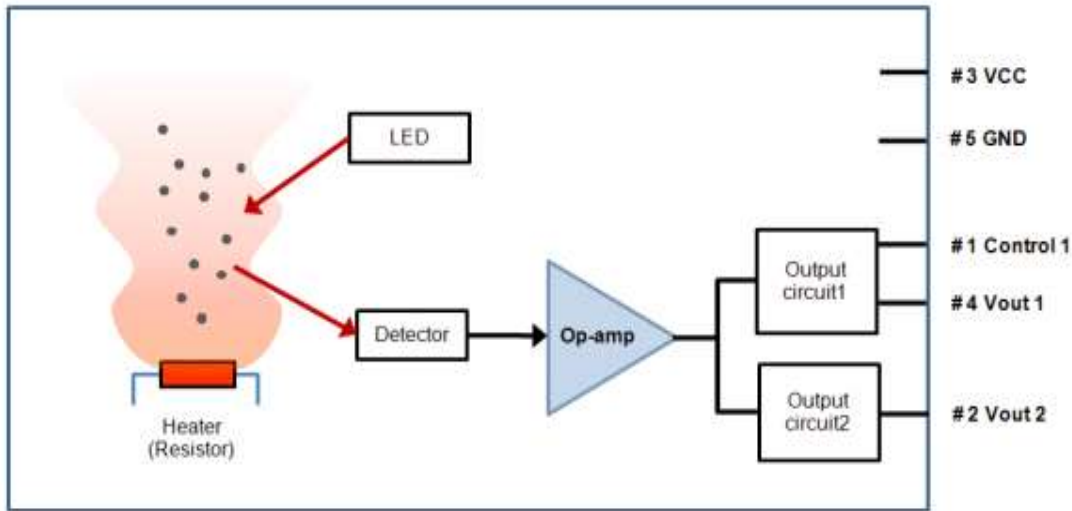
DSM501A consists of:-

➢ **Heater :** A resistor is used as the heater in this module. The heat creates upward current of air which draws outside air into the module.

➢ **Light Emitting Diode Lamp :** The high intensity light source(LED) is used to illuminate the particle as they pass through the detection chamber.

➢ **Lens :** A convex lens is used here to converge the reflected light which passes through the lens. The lens is coated with anti static and anti dust polymer.

➢ **Photodiode Detector :** The detector is used to detect and measure the dispersion of reflected light, when they pass through the laser source.

➢ **Signal Amplifier Circuit :** The amplifier circuit is used to convert the measurements from detector to PWM(Pulse Width Modulation) output signal.

➢ **Output drive circuit 1 :** This output circuit gives the PWM output of PM2.5

➢ **Output drive circuit 2 :** This output circuit gives the PWM output of PM1.0

2.1.6   PINOUT I/O Description

Pin number	Pin name	Description

24

#1	Control	Vout 1 control
#2	Vout 2	Vout 2 output(PWM)
#3	Vcc	Positive power supply
#4	Vout 1	Vout 1 output(PWM)
#5	GND	Ground

Control (Pin #1) : This pin is used for tuning the sensitivity when Vout1 is used.

V out 2 (Pin #2) : The port gives a PWM output for the density of the particles over 1.0 micrometres.

Vcc (Pin #3) : DC 5V positive power supply is used for this sensor module.

V out 1 (Pin #4) : The port gives a PWM output for the density of the particles over 2.5 micrometres.

Ground (Pin #5) : This pin is used for the Ground.

2.1.7   ALGORITHM of DSM501A

STEP 1   :   START

STEP 2   :   Input / call library KarserDSM501.h, SPI.h, SD.h

STEP 3   :   Define variables PM 1.0 and PM 2.5

STEP 4   :   Setup : Initialize sensors PM-DSM501A and SD card

STEP 5   :   Read Sensor : Detect particle size of 1 µm(PM 1.0) and detect particle size

of 2.5µm(PM 2.5)

STEP 6   :   If value < 0 then go to STEP 4, else go to STEP 7

STEP 7   :   Serial monitor enabled : if YES go to STEP 8,else go to STEP 9

STEP 8   :   Debugging output : display sensor readings to serial monitor only while tethered
to USB

STEP 9   :   Check SD card is present : if YES go to STEP 10, else go to STEP 11

STEP 10   :   Save data to SD card : open a file and write values of PM 1.0 and PM 2.5 to the
SD card in txt format. Then go to STEP 12

STEP 11   :   SD card failed : blink the LED in the Arduino, then go to STEP 5

STEP 12   :   Delay : take 1 second delay before reading the next value, then go to STEP 5

2.1.8 Flow chart of

```
START  →  Call Library  →  Define Variables
                                    ↓
                         Initialize Sensors
               Ye        DSM501A and SD card
                                    ↓
          If                Read Sensor
         Value <      ←    Detect PM 1.0 and PM 2.5
          0
          ↓ No
                              ↑
      Serial
     Monitor
     enabled
  Debugging  ← Yes
  g
  Output
          ↓ No
                                    SD card
      Check SD                      Failed
      card is    →    No
      present
          ↓ Ye

      Save Data
      to SD card
          ↓

      Delay 1 sec
```

2.1.9   **System Architecture**

> **Heater :** A resistor is used as the heater in this module. The heat creates upward current of air which draws outside air into the module.

> **Light Emitting Diode Lamp :** The high intensity light source(LED) is used to illuminate the particle as they pass through the detection chamber.

> **Lens :** A convex lens is used here to converge the reflected light which passes through the lens. The lens is coated with anti static and anti dust polymer.

> **Photodiode Detector :** The detector is used to detect and measure the dispersion of reflected light, when they pass through the laser source.

> **Signal Amplifier Circuit :** The amplifier circuit is used to convert the measurements from detector to PWM(Pulse Width Modulation) output signal.

➢ **Output drive circuit 1 :** This output circuit gives the PWM output of PM2.5

➢ **Output drive circuit 2 :** This output circuit gives the PWM output of PM1.0

2.1.9   **Working Protocol**
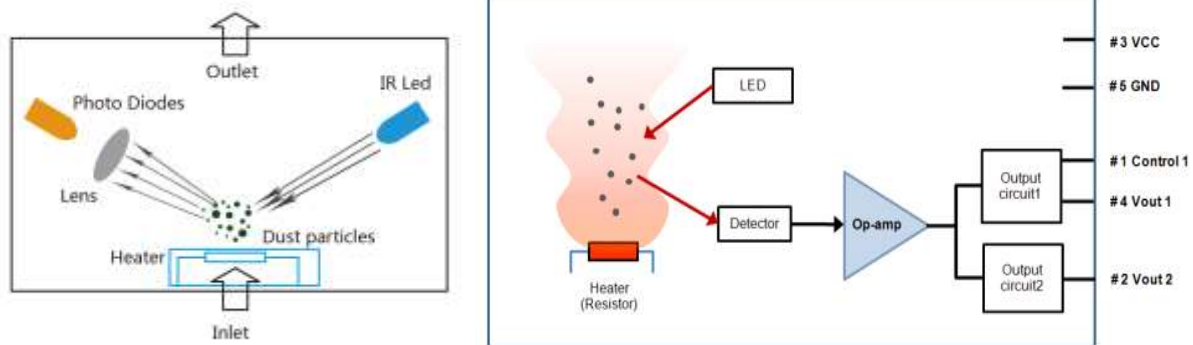
Let's see how this dust sensor module DSM501A works. First of all this dust sensor module must be placed vertically. DC 5V positive power supply is used for this sensor module. After applying DC 5V power supply to Pin #3, turn ON the power supply. The first one minute after turning the power supply ON is the warm-up time. The dust particles are entering into the detection chamber through the inlet. There is a resistor at the inlet which acts as heating element. The heat creates upward current of air which draws outside air into the module. Inside the detection chamber there is laser source. An infrared LED is used as the light source here. When the particle is passes through the light source, the infrared light is reflected by hitting the particles. The reflected light hitting the dust particles is then passes through the focusing lens and then enters into the photo detector. A convex lens is used here as the focusing object. The photo detector detects and measures the dispersion of reflected light. This measurement is then converted to PWM output signal with the help of an operational amplifier (op-amp). These

29

electrical signals are sending to a microcontroller (arduino) through two output pins, Vout1 and Vout2. The microcontroller converts the electrical signals from the sensor module into digital signal. We can see the sensor readings through a serial monitor only while tethered to USB or we can store the output data on an SD card. Here we use an SD card to write the digital output data.

A command code called interrupt is included in the code, so when other modules are integrated, even if the next values of other modules deviate from the normal time stamp of 1 second delay for next values, the DSM501A interrupt the others and maintain the time stamp of 1 second delay. This command will give priority to DSM501A values when integrating with other modules.

2.2   Testing and Calibration in Ambient condition

DC 5V power supply should be given to this sensor module in an ambient condition. Generally, the first one minute after turning the power supply ON is taken as the warm-up time. During the warm-up time, this sensor module reads the value of PM2.5 and PM 1.0 as zero values. After the warm-up time the first few values will rise to a high value or the sensor read the values as high. These values are totally wrong values. This is a major drawback of this sensor module.

2.2.1   Testing in simulated working condition

Normally DC 5V power supply is given to DSM501A, but ±10 % change can be given in simulated working condition. That is 4.5V can be given as the minimum input voltage and 5.5V can be given as the maximum input voltage. If an adhesive particle such as oil etc gets inside the module, which may cause malfunction by sticking to the optical part. It was not possible to operate this module in the locations where the condensation occurs frequently and when it was operated in such locations, the output values attained its low saturation level. In highly polluted areas, the output values are more likely to saturate at its high level.

2.2.2 <u>**DHT11 Humidity & Temperature Sensor**</u>

DHT11- is a digital humidity and temperature sensor. DHT 11 is used to analyse the Humidity and temperature. The humidity and temperature sensor combine with a digital signal output which already calibrated. DHT11 has technology to sense the temperature and humidity of its surroundings. It shows high reliability and power full long- term stability. The humidity measurement unit contains resistive type component. Temperature contains NTC connects to a high -performance 8 -bit microcontroller and temperature measurement component, fast response, cost-effectiveness, anti-interference ability and offering excellent quality.

Every DHT11 calibrated in laboratory. That calibration coefficients are stored as programmes by using OTP memory. This type of memory helps to sensor's internal signal detecting process. The system integration used by a single-wire serial interface technology that makes not only easy and quick but also low power consumption and up-to 20meter signal transmission.

ISINWEI

2.2.3   <u>**Humidity sensor**</u>

Humidity means that the measure of water vapour contains in the air. All the places have different Humidity because climate and whether influences the humidity of that place. The level of humidity that occur in air makes so many biological, physical, and chemical process. The measurement values of humidity is very important according to industrials like semiconductor industry, control system industry etc… The Air atmosphere contain various type of gases including nitrogen, argon, oxygen etc….

By measuring humidity, we can analysis the amount of moisture present in the gases. Mainly humidity sensors divided into two types based on measurement units.

➢ **Relative humidity sensor**

➢ **Absolute humidity sensor**

2.2.4   <u>**Temperature sensor**</u>

Temperature sensors are a particular type of electronic devices which accurately detects the temperature of its surroundings and transforms the incoming data into digital output to control record. There are different types of temperature sensor available in market. Some are the direct contact with the physical target, but some are the indirect contact with the physical target. These behaviour of temperature sensors called non-contact temperature sensors.

Temperature sensors quantify the quantity of amount heat or coldness that produced by a system. These properties make digital output or an analogue. Mainly temperature sensors are used for different types of industrial works and chemical processes. Here we are using a combination of humidity sensor and temperature sensor called DHT11.

Temperature sensor have basic working principle based on the voltage in its diode. The detecting temperature variations is directly related to the resistance of this diode. The diode have resistance which detect and transformed into readable of this diode. The main temperature readable values such that Fahrenheit, Kelvin, Centigrade etc...

2.2.5   <u>**DHT11 Pinout Configuration**</u>

No:	Pin - Name	Description

	DHT11 Sensor	
1	Vcc	Power supply - 3.5V to 5.5V
2	Data	No connection and hence nor used
3	Data	Outputs both Humidity and Temperature through serial data
4	Ground	The circuit is connected to ground

For the DHT11 Sensor module

| 1	VCC	Power supply 3.5V to 5.5V
2	Data	Outputs both Humidity and Temperature through serial Data
3	Ground	The circuit is connected to ground

2.2.6   <u>DHT 11 Specifications</u>

- **Temperature Range: 50°C and 0°C**

- **Operating Voltage: 5.5V to 3.5V**

- **Output: Serial data**

- **Operating current: 0.3mA (measuring) 60uA (standby)**

- **Accuracy: ±1°C and ±1%**

- **Humidity of Range: 90% to 20%**

- **Resolution: Temperature and Humidity both are 16-bit**

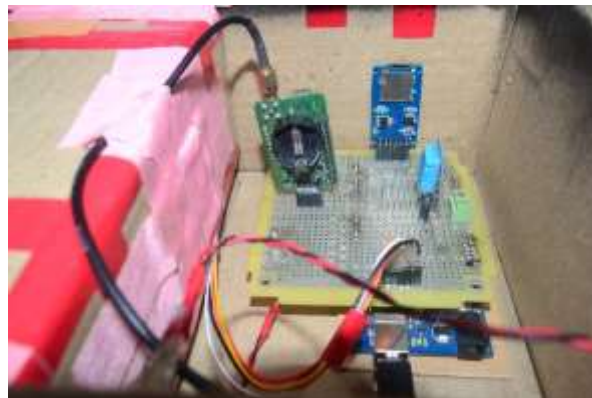**2.2.7**   <u>**Assembly sequences**</u>
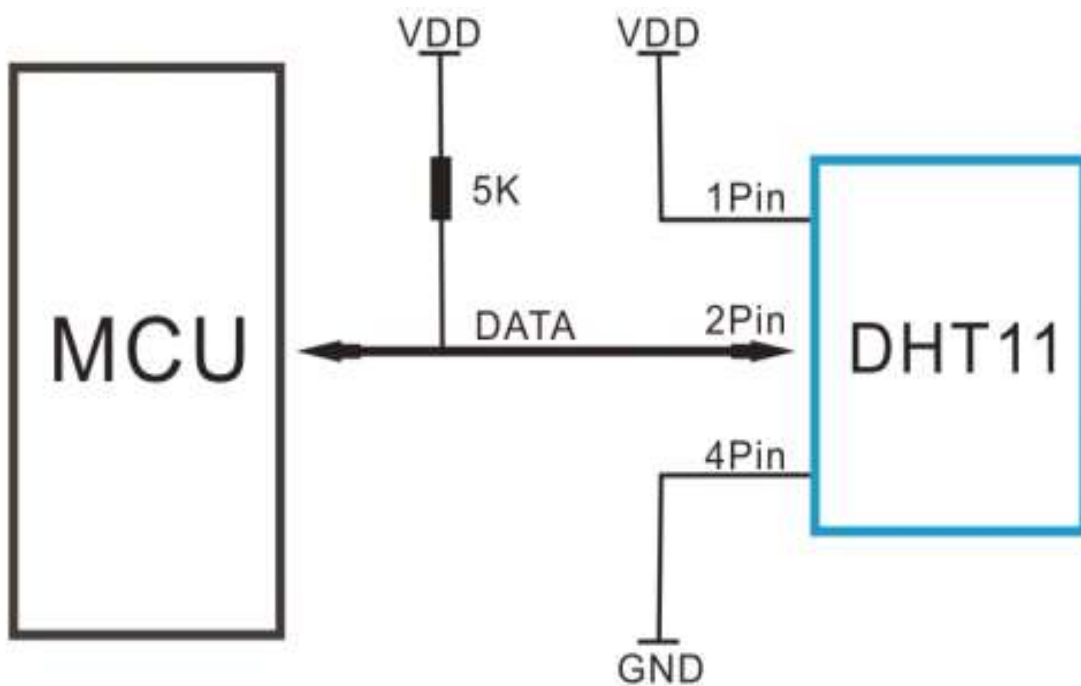
2.2.8 <u>Communication process</u>

DHT 11 changes the low power – consumption mode to running mode by MCU starts message. DHT 11 sends back response signal which contains 40 -bit data that includes the temperature and relative humidity information to MCU. After data is collected, DHT 11 retired into low power-consumption mode.


2.2.9 <u>MCU sends starts signal to DHT 11</u>

When the MCU starts communication with DHT11, the program of Microcontroller unit will set Data single-bus voltage level from high to low and process. To ensure DHT's detection of MCU 's signal, we take at least 18ms.

2.3   <u>DHT Sensor Responses to MCU</u>

When DHT sensor detects the starts signal, it would send out low voltage signal to MCU. Then DHT sensor's programme prepare for the data transmission. While the data send the MCU, bits of data begin with low voltage power level the signal decided whether data bits "0" or "1". If a response signal indicates always high- voltage-level, it shows that DHT sensor is not work properly and check the connections.
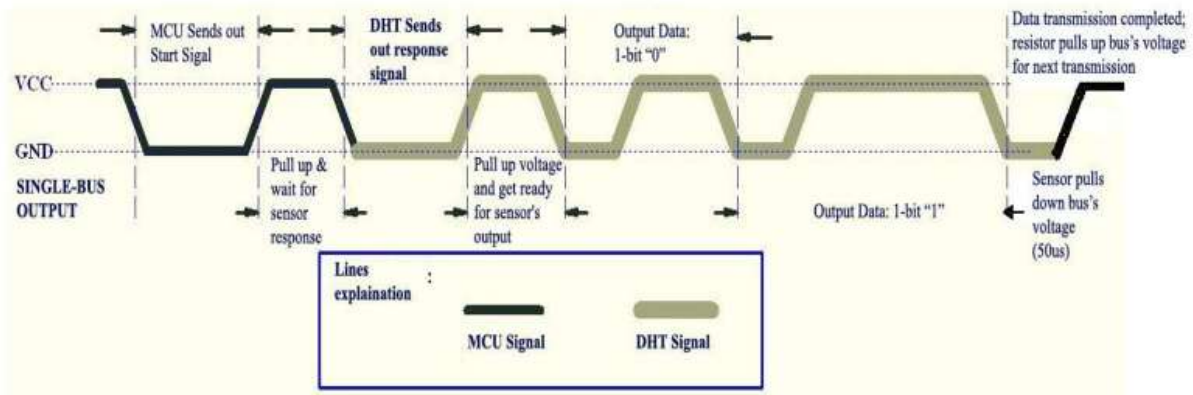
2.3.1   <u>**ALGORITHAM**</u>

  Step 1 – Start

  Step 2 – input /call libraries DHT.h, SPI.h, SD.h

  Step 3 – Define variables

  Step 4 – set up intialise sensor DHT sensor and SD card

  Step 5 – Read humidity temperature in degree celsius

  Step 6 – If value < 0 then go to step -4 else go to step -7

  Step 7 – Serial monitor -if yes go to step 8 else go to step 9

  Step 8 – Debugging output:

  display sensor readings to serial monitor only while tethered to USB.

  Step 9 - Check SD card is presented :

  if yes go to step 10 else go to step 13

  Step 10 – Display and write:

  SD card failed' in serial monitor and SD card blink LED in Arduino t hen go to step 13
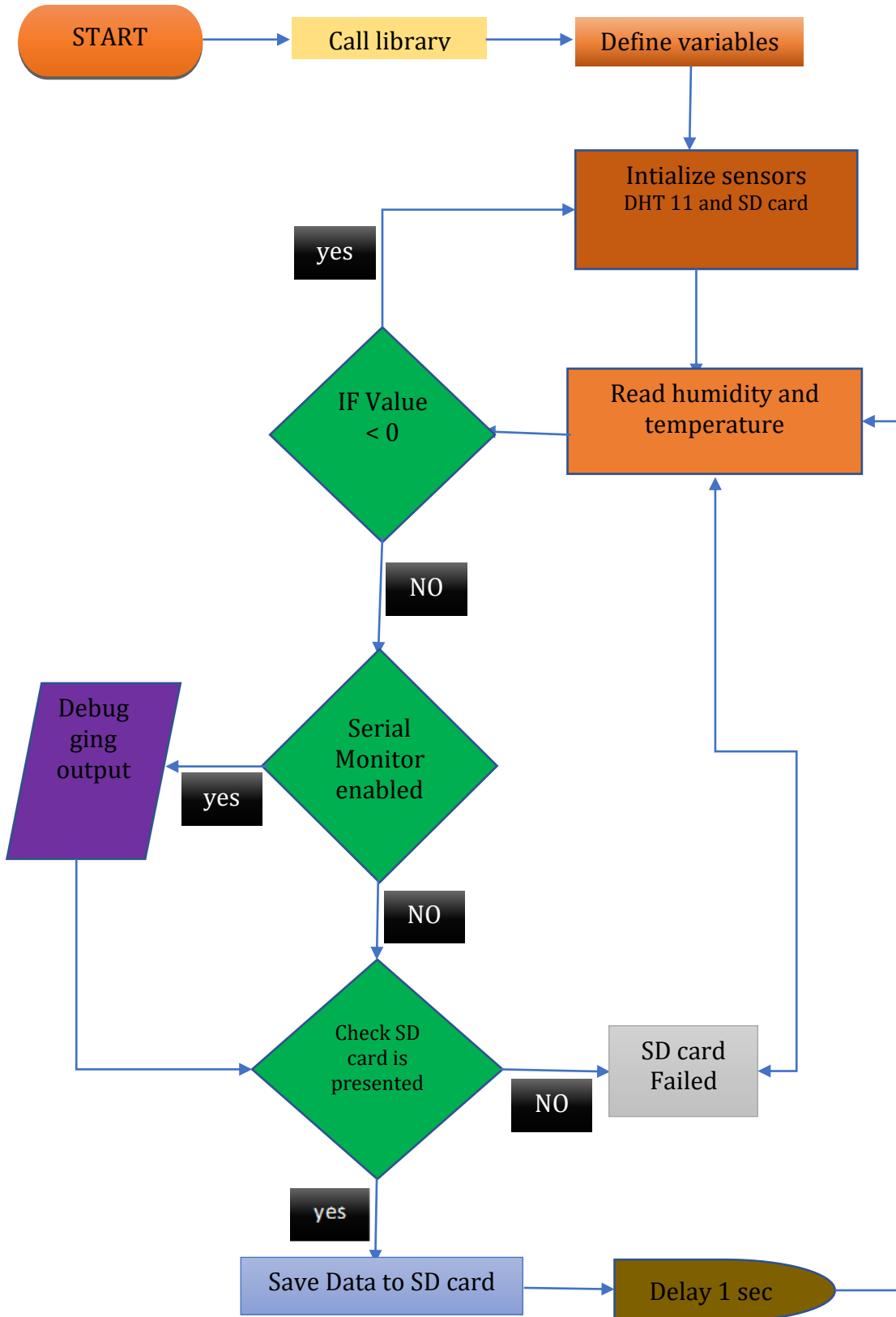
Step 11 – Save to SD card:

Write the SD card with obtained temperature and humidity ,date and time then save as .txt file.

Step 12 – Display from getting next Step for 1 second for synchronise with other modules then go to step 5

2.3.2

Flow chart of DHT

START → Call library → Define variables

Define variables →

Intialize sensors
DHT 11 and SD card

yes

Intialize sensors → Read humidity and temperature

IF Value < 0 ← Read humidity and temperature

NO

Serial Monitor enabled

yes → Debug ging output

NO

Check SD card is presented

NO → SD card Failed

yes

Save Data to SD card → Delay 1 sec

2.3.3   <u>DHT11 SENSOR   - WORKING PRINCIPLE</u>

DHT11 not only senses the humidity but also temperature. DHT11 sensor contains thermistor for sensing temperature and capacitive humidity sensing element. According to humidity, capacitor have two electrodes. That contain moisture holding substrate as a dielectric between them. By changing capacitive values then we get corresponding humidity levels. In the case of IC measure process, changing resistance values and convert them into digital type. This sensor made up of tiny ceramic semiconductors. The sensor has negative coefficient thermistor, which make a decrease in its values of resistance by increase in temperature. To get larger resistance value then we make smallest change in temperature values. The temperature range of DHT11 is that 0 to 50 degree Celsius with a 2- degree accuracy . In the case of humidity sensor, It provides 20 to 80% with 5% accuracy. Sensors sampling rate is1 HZ. IT helps to read in every second. DTH11 has small voltage from 3 to 5 volts. The current is 2.5mA while measuring. DHT sensor have numerous pins which including Data pin, GND, VCC,   and a not connected pin. To make smooth communication between Dht11 sensor and microcontroller, we use a pull -up resistor which have capability of 5k to 10k ohms.

2.3.4   <u>Application</u>

This sensor has numerous applications including measurement of   temperature and humidity values in heating, air condition systems and ventilation. DHT11 mainly used to measure whether area for analysis current condition of whether. That process helps to predict the upcoming whether situations. Sensor of humidity used as a safety measure. The widely using applications of DHT 11 are Here

- **Measure temperature and humidity**

- **Local Weather station**

- **Automatic climate control**

- **Environment monitoring**

2.3.5   **TESTING**

2.3.6   **Testing and calibration in Ambient condition**

Dc 5v power supply should be given to this sensor in an ambient condition Generally the first one minute after turning the power supply ON is taken as the warm – up time During the warm-up time, this sensor module reads the values of temperature and humidity as zero values. Not only quality of connection wires but also distance with DHT sensor affect the out-put data from DHT sensor.

2.3.7   <u>Testing in simulated working condition</u>

In simulated working condition DHT 11 sensor shows 3% RH signal shift. By using the operation condition sensor can change the calibrated status gradually. Typically, the sensor use DC 5V. But its maximum value 5.5V and minimum value is 3V. The Vapour from chemical materials affects the DHT's sensitive-elements and sensitivity. High amount of chemical substance may permanently damage the sensor. Similarly long-term strong radiations and sunlight may interfere the database and DHT's performance. We have proper quality of connection wire and distance of communication to test simulated working condition.. These types of locations gave low saturation level values or high saturation level values.

2.3.8   <u>GPS Module</u>

    The Global Navigational Satellite System (GNSS) is the constellation of autonomous geo-spatial satellites that contribute highly accurate positioning, navigation and timing (PNT) measurements worldwide. GNSS provide global coverage. GPS (Global Positioning Satellite) is one of the many GNSS that provide PNT measurement which is operated by U.S Armed force. Likewise Europe – Galileo, Russia – GLONASS, China – BeiDou, India – NavIC, Japan – QZSS. GPS can work in all weather conditions with line of sight communication with 4 or more GPS satellites for accuracy.

Fig no. 1: Chip of NEO 6 series

<u>Model</u> – NEO -6Q-0-000

<u>Make</u> – u-blox ®

Fig no. 2: u-blox NEO-6Q-0-000

There are many GPS modules in the market but U-blox provide large variety of GPS modules in different price range and sensitivity. There are series of modules like UBX, CAM, MAX, NEO, LEA, EVA, SAM, ZOE, MIA and ZED, all these belongs to excellent GPS module series. From these series we opted NEO 6 series. Because NEO 6Q is an easy to interface, low power consumption, budget friendly, high performance which can track 22 satellites with receiver type of 50 channel u-blox 6 engine which can locate anywhere in the world.

AssistNow autonomous data is the data which generate accurate satellite orbital data that is reliable for few days that estimating the position without connectivity also with fast position fix within those days. NEO 6Q's oscillator is TCXO (Temperature Compensated Crystal Oscillator) which provides more stable and precise clock signal to receiver than standard crystal oscillator used in same series (NEO 6M).

The manufacturing of NEO 6Q's production site is certified under ISO 16949, and the product NEO 6Q is qualified with ISO 16750.

NEO 6Q belongs to NEO 6 series which are generally known as versatile u-blox 6 GPS modules. The main highlight of this modules are USB, UART, DDC (PC compliant) and SPI interface for connectivity, the module is available in TCXO and crystal versions, module has onboard RTC crystal to enhance the productivity by faster warm and hot starts, and power supply.

NEO-6 modules are Moisture Sensitive Devices (MSD) in according to the JEDEC /IPC specification. NEO-6 modules are rated at MSL (Moisture sensitivity level) level 4.

2.3.9 <u>GPS Performance / Working Protocol</u>

Let's begin with the power supply which can be in the range of 2.7V to 3.6V. The type of antenna is patch antenna, which are mostly ceramic, flat and metal body which is mounted to a metal plate. For optimal performance the position of placing this antenna is very crucial. The placing of this antenna should be oriented parallel to geographic horizon. There shouldn't be any obstacles between GPS receiver and sky to maintain direct line of sight with as many as possible with visible satellite.

For knowing the values from GPS module, we have to connect module to computer by establish a serial communication using USB-UART adapter. The connection with Arduino and NEO 6Q is as follows,

NEO-6Q GPS Module	USB to Serial Converter (Arduino)
VCC	3.3V
GND	GND
TX	RX
RX	TX

Firstly, we have to run serial passthrough (raw GPS data) through which we can obtain all the informations that the GPS provide, which include latitude, longitude, altitude, satellite speed, time, date, number of satellites tracked, horizontal dilution of position and height of geoid (mean sea level). Some NMEA (National Marine Electronics Association) sentences are;

- $GPGGA – basic GPS NMEA message (which provide accuracy data and 3D location)
- $GPGSA – Active satellites and GPS DOP
- $GPGSV – GPS satellite detailed information
- $GPGLL – Geographic Longitude and Latitude
- $GPRMC – Essential GPS data (Position, time, velocity)
- $GPVTG – Track made good and ground speed

Fig no. 3: serial passthrough when searching for satellites

If the GPS is not locked with sufficient satellites then the values will be random which implies module need more time, line of sight is interrupted or some other reasons. We can find an LED on the NEO 6Q GPS module that indicates current status of the position fix. If the LED is not blinking then it's searching for satellites and if the LED is blinking in every second then the GPS module has found enough satellites in locating.

Fig no. 4: Serial passthrough when locked to satellites

We use Arduino IDE to communicate with Arduino MEGA. For connecting GPS module to Arduino, we need to download and install the 'TinyGPS' library for parsing NMEA data streams from GPS module. The default baud rate is 9600 for NEO 6Q. The values can be viewed in serial monitor.

Once we find the GPS is working perfectly then we can upload the source code which consist GPS module, SD card and Arduino MEGA. If Arduino is connected to computer then we can see the data via serial monitor simultaneously the data will be stored in SD card in '.txt' format.

```
File Edit Format View Help
Latitude,Longitude,Altitude,Date,Time
12.7789916,77.6434326,913.70,16/6/2022,6:42:58.
12.7789936,77.6434326,913.60,16/6/2022,6:43:1.
12.7789936,77.6434326,913.60,16/6/2022,6:43:1.
12.7790002,77.6434326,913.50,16/6/2022,6:43:4.
12.7790002,77.6434326,913.50,16/6/2022,6:43:4.
12.7790002,77.6434326,914.40,16/6/2022,6:43:7.
12.7790002,77.6434326,914.40,16/6/2022,6:43:7.
12.7790250,77.6434402,914.20,16/6/2022,6:43:10.
12.7790250,77.6434402,914.20,16/6/2022,6:43:10.
12.7790384,77.6434326,915.60,16/6/2022,6:43:13.
12.7790384,77.6434326,915.60,16/6/2022,6:43:13.
12.7790670,77.6434326,915.80,16/6/2022,6:43:16.
12.7790670,77.6434326,915.80,16/6/2022,6:43:16.
12.7790880,77.6434097,916.70,16/6/2022,6:43:19.
12.7790880,77.6434097,916.70,16/6/2022,6:43:19.
12.7791090,77.6434020,918.10,16/6/2022,6:43:22.
12.7791090,77.6434020,918.10,16/6/2022,6:43:22.
12.7791385,77.6434097,918.10,16/6/2022,6:43:25.
12.7791385,77.6434097,918.10,16/6/2022,6:43:25.
12.7791700,77.6434097,917.70,16/6/2022,6:43:28.
12.7791700,77.6434097,917.70,16/6/2022,6:43:28.
12.7791900,77.6434097,917.30,16/6/2022,6:43:31.
12.7791900,77.6434097,917.30,16/6/2022,6:43:31.
12.7791967,77.6434249,917.40,16/6/2022,6:43:34.
12.7791967,77.6434249,917.40,16/6/2022,6:43:34.
12.7791948,77.6434555,917.50,16/6/2022,6:43:37.
12.7791948,77.6434555,917.50,16/6/2022,6:43:37.
12.7791967,77.6434707,917.00,16/6/2022,6:43:40.
12.7791967,77.6434707,917.00,16/6/2022,6:43:40.
12.7791996,77.6434860,916.90,16/6/2022,6:43:43.
```

Fig no. 5: Values that stored in SD card when GPS was locked with satellites

2.4.0   Testing and calibration in ambient conditions

For locking to at least 3 satellite, it takes 26 seconds in ambient temperature (cold and warm start). U-blox NEO 6Q is a sensible module of sensitivity in navigation and tracking of -162dBm. The sensitivity varies from ambient, that is hot and cold (without aid) is -157dBm and -148dBm respectively. Velocity accuracy of the module while moving is 0.1m/s. horizontal position accuracy is about 2.5m.

It is always recommended to supply 3.3V even though the module is 5V tolerant, so we can connect to any 5V logic microcontroller. NEO 6Q has power consumption of 111mW when continuous 3V and 33mW when power save mode. There should be a backup power of 1.4V to 3.6V with 22µA current for uninterrupted values from GPS module.

2.4.1   Testing in simulated working conditions

Due to Assist Now Autonomous the acquisition is faster. For locking to the satellite, it takes 1 second in hot and aided starts. The sensitivity differs from cold and hot start as -148dBm and -157dBm respectively. The maximum altitude measure is up to 50000m and maximum

49

velocity up to which module work perfectly is 500m/s. Operating beyond the specified limits will affect GPS module reliability. The GPS module has same operating and storage temperature of -40°C to 85°C. Configurable Time pulse frequency range varies from 0.25Hz to 1kHz.

2.4.2 <u>System architecture</u>

Symbol	Min. (mm)	Typ. (mm)	Max. (mm)
A	15.9	16.0	16.6
B	12.1	12.2	12.3
C	2.2	2.4	2.6
D	0.9	1.0	1.3
E	1.0	1.1	1.2
F	2.9	3.0	3.1
G	0.9	1.0	1.3
H		0.82	
K	0.7	0.8	0.9
M	0.8	0.9	1.0
N	0.4	0.5	0.6
Weight		1.6g	
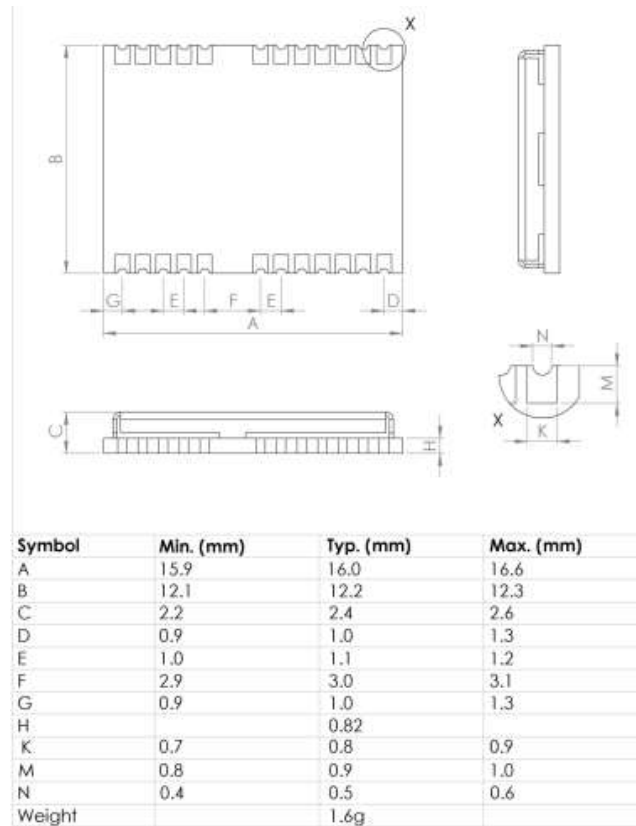
Fig no. 6: dimension of the chip used in NEO 6Q.

2.4.3
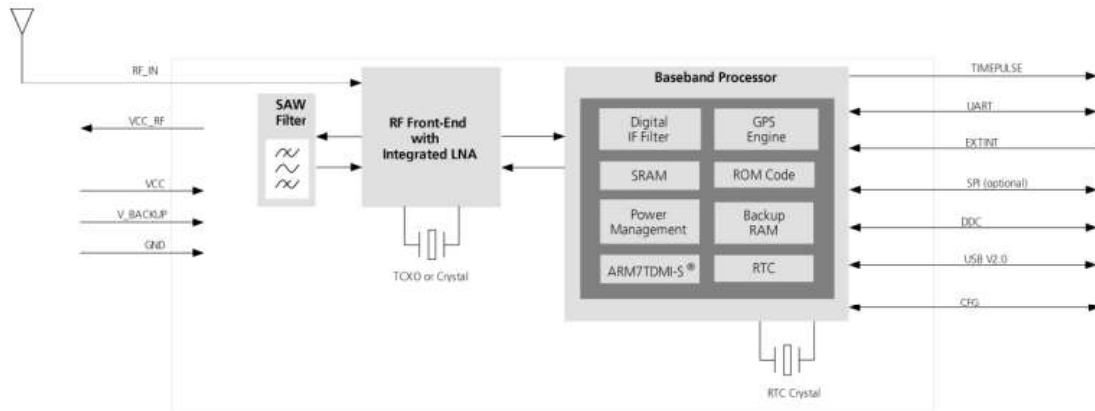
Block diagram

Fig no. 7: detailed block diagram

2.4.4   **Pinout**
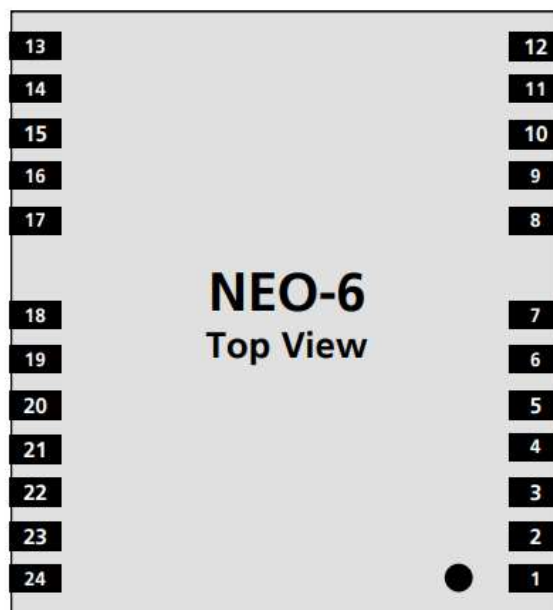
Fig no.   8: Pin assignment

Number	Name	I/O	Description
1	Reserved	I	Reserved
2	SS_N	I	SPI Slave Select
3	TIMEPULSE	O	Timepulse (1PPS)
4	EXTINT0	I	External Interrupt Pin
5	USB_DM	I/O	USB Data
6	USB_DP	I/O	USB Data
7	VDDUSB	I	USB Supply
8	Reserved		Pin 8 and 9 must be connected together. (See Hardware Integration Manual)
9	VCC_RF	O	Pin 8 and 9 must be connected together. (Output Voltage RF section)
10	GND	I	Ground
11	RF_IN	I	GPS signal input
12	GND	I	Ground
13	GND	I	Ground
14	MOSI/CFG_COM0	O/I	SPI MOSI / Configuration Pin. Leave open if not used.
15	MISO/CFG_COM1	I	SPI MISO / Configuration Pin. Leave open if not used.
16	CFG_GPS0/SCK	I	Power Mode Configuration Pin / SPI Clock. Leave open if not used
17	Reserved	I	Reserved
18	SDA2	I/O	DDC Data
19	SCL2	I/O	DDC CLOCK
20	TxD1	O	Serial Port 1
21	RxD1	I	Serial Port 1

2.4.5   Explanation of codes
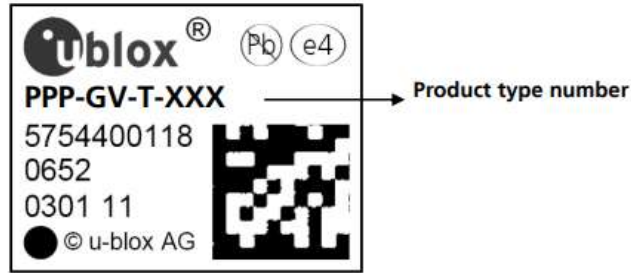
Fig no. 9: Sticker in the chip of GPS module

Elaboration of the codes mentioned in the chip of u-blox NEO-6Q

Format	Structure
Product's Name	PPP-GV
Ordering Code	PPP-GV-T
Type Number	PPP-GV-T-XXX

GPS module used : NEO-6Q-0-000

Code	Meaning	Example
PPP	Family (Product)	NEO
G	Generation (Product)	6 = u-blox6
V	Variant	T = Timing, R = DR, etc. Module used : Q
T	Quality Grade / Option	Describes standardized quality grade or functional element, i.e., Automotive grade, flash size, etc.
XXX	Product Details	Describes options or product details such as hardwave and software revision, cable length, etc.

2.4.6   <u>Algorithm for GPS with SD card</u>

STEP 1:   START

STEP 2:   Input / Call libraries:

                 TinyGPS.h, SPI.h, SD.h

STEP 3:   Define variables

STEP 4:   Setup:

                 Initialise sensors NEO 6Q-GPS, SD card

STEP 5:   GPS locked then go to STEP 6 else go to STEP 4

STEP 6:   Encode GPS values; Serial passthrough

STEP 7:   Call out library info for extracting variables from GPS serial

                 passthrough. i.e., latitude, longitude, altitude, date, time

STEP 8:   Serial monitor enabled, if YES go to STEP 9 else go to STEP 10

STEP 9:   Debugging output:

                 Display serial passthrough and all required variable to serial monitor
                 only while tethered to USB. Then go to STEP 10

STEP 10:   Check SD card:

                 SD card found, then go to STEP 12 else go to 11

STEP 11:   Display and write 'SD card failed' in serial monitor and SD card, blink
                 LED in Arduino, then go to STEP 13.

STEP 12:   Save to SD card:

                 Write the SD card with obtained latitude, longitude, altitude, date, time
                 then save as .txt file.

STEP 13:   Delay:

Delay from getting next step for 1 second for synchronise with other modules, then go to STEP 5

2.4.7   <u>Flow chart</u>
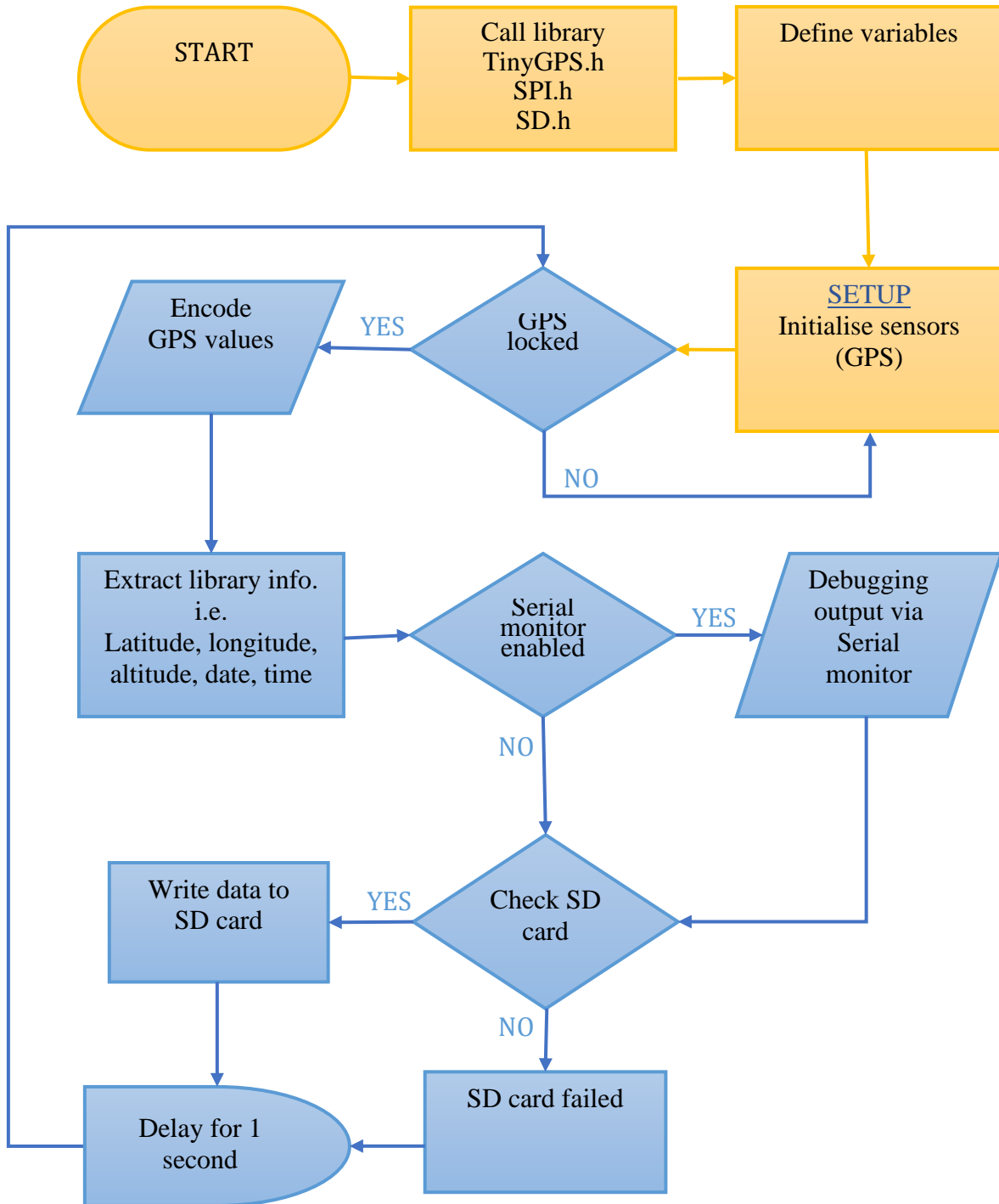
3. <u>Design and Implementation</u>

3.1 <u>Action plan</u>

The objective of our project was to study about the particulate matter present in atmosphere. For this we are planning to design and integrate some sensor modules that measure the concentration of particulate matter in ground and near space. To integrate, we plan to purchase some sensor modules like GPS, Temperature - humidity sensor and Particle counter. Particle counter is the sensor used to measure particulate matter, GPS module is for tracking the exact location.

Our plan was to construct a payload by integrating these sensor modules, then to conduct ground survey with this payload in Bangalore city. After the ground survey, construct a payload box using Styrofoam and then insert the payload inside the box and conduct tethered flight up to 1000 meter from sea level.
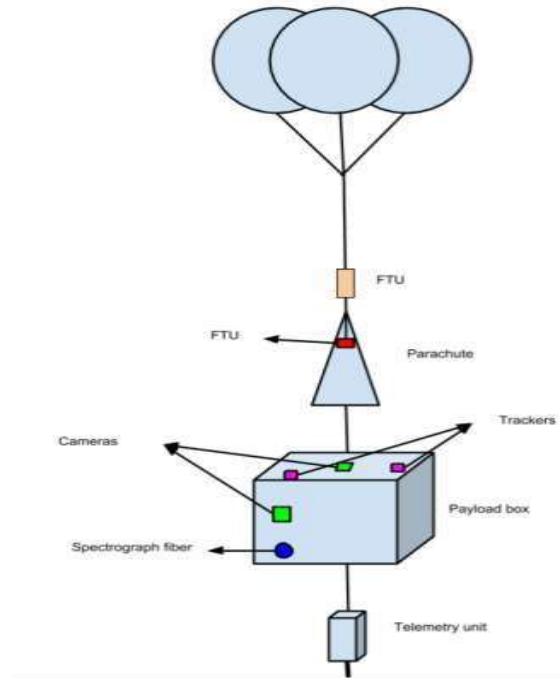
Fig no.3.1 : Model of actual flight

Followed by the tethered flight we considered for an actual flight which will be launched from IIA campus in Hoskotte. For actual flight we need permissions from higher authorities for the clearance due to which the balloon consisting the payload will reach to a height of 32KM from the sea level, and the crash site can be in the radius of approximately 120KM from the launched site.

3.2   Project Procedure

Fig no.3.2: Logo of our project

We named our project name as JAStro, inspired by the name AstroSAT designed by Indian Institute of Astrophysics, Bangalore. As we know the main objective of this mission was to find the concentration of particulate matter in the atmosphere. So we need a precise, cost effective and compact sized particle counter. There are numerous models available in the market, within it the best option was SAMYOUNG DSM501A which was bought from the authorized dealer 'element14'. A controller for operating all sensors modules, a microprocessor or microcontrollers is necessary, so we selected Arduino UNO and Arduino MEGA as microcontroller and raspberry py4 as microprocessor. To accompany this instrument, we included GPS, temperature – humidity sensor, battery and SD card module. We bought GPS module u-blox NEO-M8N-0-10 and u-blox NEO-6Q-0-000 for a backup. For temperature and humidity sensors, DHT11 was the best option because both temperature and humidity sensors are available in one module. For powering up the whole component, we opted for Lithium ion (Li-ion) of different size and capability according to our need and power-bank of 10000 mAh capacity. For data memory we used a 3.3V SD card module. Other accessories like breadboard, jumper wires, purf-board and digital camera for capturing the visuals during the tethered and actual flights.
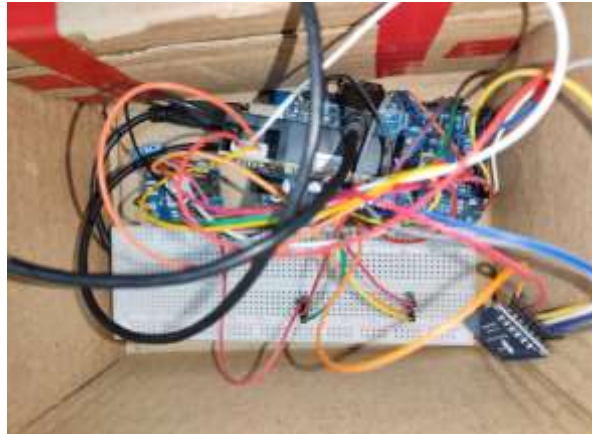
Fig no.3.3: Connected all components using breadboard

It was necessary to test and calibrate each components with Arduino. As an initial step, we wrote the source codes for temperature and humidity sensor in Arduino IDE. For that, the module tested in different conditions, places and time. The priliminary test results where obtained in static and dynamic conditions. There where some issues faced while connecting DHT11 with Arduino, that was sometimes LED indicator blinks or turned off, which indicated the power supply was entered to ground or VCC/GND disconnected. It was later rectified by reconnecting its pins according to its data sheet.

Arduino's efficiency and memory was tested by running the source codes of gyroscope, altimeter and initial measurement unit (IMU). It was found that some arduino was not able to meet the needs. We could use J-tag for reprogramming the Arduino but due to time shortage and complexity in using it, later it was kept back and opted the backup Arduino. Beside microcontroller, microprocessor (Raspberry py4) taken into consideration, although microcontroller was efficient and enough for the project, so we only considered microcontroller (Arduino). When considering Arduino UNO as microcontroller that of 32 KB, after uploading various codes into its flash memory for stand alone purpose, The Arduino UNO failed to save some codes into memory. So we discarded UNO and decided to use Arduino MEGA throughout the project, which had flash memory of 256 KB. Arduino MEGA is a microcontroller board based on ATmega2560 chip, which is capable to connect multiple sensors.

After the successful testing of DHT11, we considered the   particle counter DSM501A for testing and calibrating with Arduino MEGA. The pinout signs where not mentioned in the component for output 1 and 2 which made an unsureness. Later it was rectified by the guidance of data sheet and official site of the producer. Since the cables of this component are short and very brittle that it might fluctuate the result, so we re-soldered with some better wires and tested. With the help of Informations from data sheet, source code was written in the Arduino IDE. 2 types of codes where generated which gives the output in terms of concentration and the other is in terms of particle count. The module tested in different conditions, places and time. The results were uploaded to SD card via codes.

Particle counter and temp-humidity sensors were combined and source codes were written in Arduino IDE. SD card module was also integrated with these two modules and also made the respective changes to the source code. Values of particulate matter in different temperatures and humidities are obtained and datas are stored in the SD card.

```
12:59:18.053 -> Humidity = 79.00               Temperature,Humidity,Particle2.5,Particle1.0
12:59:19.037 -> Temperature = 30.20
12:59:19.084 -> Humidity = 79.00               27.70,89.00,16.78,2.58,
12:59:19.084 -> pm10: 0.00 pm2.5: 0.00
12:59:20.116 -> Temperature = 30.20            27.70,89.00,13.92,2.46,
12:59:20.116 -> Humidity = 79.00
12:59:20.116 -> pm10: 0.00 pm2.5: 0.00         27.70,89.00,7.92,1.89,
12:59:21.102 -> Temperature = 30.20
12:59:21.148 -> Humidity = 79.00               27.70,89.00,11.29,1.07,
12:59:21.148 -> pm10: 0.00 pm2.5: 0.00
12:59:22.180 -> Temperature = 30.30            27.50,89.00,12.77,2.08,
12:59:22.180 -> Humidity = 79.00
12:59:22.180 -> pm10: 0.00 pm2.5: 0.00         27.50,89.00,8.85,4.83,
12:59:23.166 -> Temperature = 30.30
12:59:23.213 -> Humidity = 79.00               27.70,89.00,13.26,2.62,
12:59:23.213 -> pm10: 0.00 pm2.5: 0.00
12:59:24.245 -> Temperature = 30.30            27.70,89.00,8.03,1.04,
12:59:24.245 -> Humidity = 79.00
12:59:24.291 -> pm10: 0.00 pm2.5: 0.60         27.70,89.00,14.03,2.13,
12:59:25.276 -> Temperature = 30.30
12:59:25.276 -> Humidity = 79.00               27.70,89.00,21.90,4.92,
12:59:25.276 -> pm10: 0.00 pm2.5: 0.00
12:59:26.300 -> Temperature = 30.30            27.60,89.00,18.15,4.74,
12:59:26.300 -> Humidity = 79.00
12:59:26.354 -> pm10: 0.00 pm2.5: 0.00         27.60,89.00,14.22,2.37,
12:59:27.339 -> Temperature = 30.30
12:59:27.339 -> Humidity = 79.00               27.60,89.00,17.58,4.07,
12:59:27.386 -> pm10: 0.00 pm2.5: 0.00
12:59:28.371 -> Temperature = 30.40            27.60,89.00,18.07,3.91,
                                               27.70,93.00,11.88,2.11,
                                               27.70,93.00,35.89,4.20,
```

Fig no. 3.4: values of PM and DHT11 in Serial monitor and SD card

For locating position of the payload, we used GPS module of u-blox NEO-6Q engine with accompanying library 'TinyGPS++' in Arduino IDE. The actual latitude and longitude of Madiwala, Bangalore was 12.926°N and 77.612°E. But using this code, we got latitude and longitude of Madiwala as 12.5541°N and 77.3698°E which had an error of 49.88KM from the actual position. The working and accuracy of GPS module was verified by running serial passthrough codes in Arduino IDE which gives all the information that the GPS module capture.

It takes 20 seconds in outdoor and around 1 minute in indoor to lock the GPS module with satellite. GPS module should lock with atleast 4 satellite to give accurate location. We had numerous codes regarding GPS module. Using the first code, GPS module as able to lock faster but the latitude, longitude and altitude values was false and also the values were not continuous. So we switched the GPS module to u-blox NEO-M8N with the same library 'TinyGPS++'. The values were continuous but still obtained the corrupted value and the value of altitude was absent. We combined all the 4 modules in Arduino MEGA and codes where compressed to a single code. While running this code, the GPS values were interrupted by the particle counter. We thought the problem was due some technical issues of ground's connection and checked it, but even then the error continued. It was due to prioritising the particle counter. So we approached an expert for the guidance,as per his opinion; the particle counter was designed in the manner that if GPS module is integrated with PM counter, the values of GPS becomes zero and PM counter value dominates. The solution suggested was to introduce a master-slave in microcontroller, that 2 Arduinos are mutually connected to each other by either communication port or I2C port. So, we considered Arduino MEGA as master which consists of particle counter and temp-humidity sensor, by the time slave Arduino (UNO) consists of GPS module. It was practically complicated toexecute, due to lack of time. By thorough inspection in hardware and software, we found that the problem was due to the interrupt command in the source code for the particle counter and that interrupt was necessary for the particle counter for its smooth running. The GPS module was switched back to u-blox NEO-6Q. There was a program crash between the interrupt command and TinyGPS++ library. Later it was rectified by changing the library to TinyGPS. The whole instrument was working perfect from then. For additional information to the data, we encoded date, time and altitude.

```
16:10:28.713 -> $GNGLL,,,,,,V,N*7A
16:10:29.679 -> Temperature = 28.70
16:10:29.722 -> Humidity = 84.00
16:10:29.722 -> pm10: 0.00 pm2.5: 0.00
16:10:29.768 -> GPS data : $GNRMC,,V,,,,,,,,,,N*4D
16:10:29.811 -> $GNVTG,,,,,,,,N*2E
16:10:29.811 -> $GNGGA,,,,,0,00,Temperature = 28.70
16:10:30.822 -> Humidity = 83.00
16:10:30.870 -> pm10: 5.25 pm2.5: 0.00
16:10:30.870 -> GPS data : $GNRMC,,V,,,,,,,,,,N*4D
16:10:30.906 -> $GNVTG,,,,,,,,N*2E
16:10:30.954 -> $GNGGA,,,,,0,00,Temperature = 28.70
16:10:31.916 -> Humidity = 83.00
16:10:31.957 -> pm10: 31.03 pm2.5: 0.34
16:10:31.957 -> GPS data : $GNRMC,,V,,,,,,,,,,N*4D
16:10:32.003 -> $GNVTG,,,,,,,,N*2E
16:10:32.047 -> $GNGGA,,,,,0,00,Temperature = 28.70
16:10:33.055 -> Humidity = 83.00
16:10:33.055 -> pm10: 8.06 pm2.5: 0.00
16:10:33.100 -> GPS data : $GNRMC,,V,,,,,,,,,,N*4D
16:10:33.145 -> $GNVTG,,,,,,,,N*2E
16:10:33.145 -> $GNGGA,,,,,0,00,Temperature = 28.70
16:10:34.137 -> Humidity = 83.00
16:10:34.181 -> pm10: 14.02 pm2.5: 0.00
16:10:34.181 -> GPS data : $GNRMC,104034.00,V,,,,,,,,,,N*61
16:10:34.258 -> $GNVTG,,,,,,,,N*2E
16:10:34.258 -> $GNGGA,156
```

```
Temperature,Humidity,Particle2.5,Particle1.0,Latitude,Longitude,D
ate,Time
34.90,60.00,9.01,0.13,0.0000000,0.0000000,5/30/2022,5:52:5,
34.90,60.00,9.01,1.56,0.0000000,0.0000000,5/30/2022,5:52:6,
34.90,60.00,10.65,1.78,0.0000000,0.0000000,5/30/2022,5:52:8,
34.90,60.00,13.40,0.71,0.0000000,0.0000000,5/30/2022,5:52:9,
34.90,60.00,18.59,0.73,0.0000000,0.0000000,5/30/2022,5:52:10,
34.90,60.00,5.84,0.86,0.0000000,0.0000000,5/30/2022,5:52:11,
34.80,60.00,5.33,1.61,0.0000000,0.0000000,5/30/2022,5:52:13,
34.80,60.00,16.98,0.00,0.0000000,0.0000000,5/30/2022,5:52:14,
34.80,60.00,20.89,0.31,0.0000000,0.0000000,5/30/2022,5:52:15,
34.80,60.00,8.99,1.41,0.0000000,0.0000000,5/30/2022,5:52:16,
34.80,60.00,7.32,1.82,0.0000000,0.0000000,5/30/2022,5:52:17,
34.80,60.00,17.65,0.63,0.0000000,0.0000000,5/30/2022,5:52:19,
34.90,60.00,21.35,1.36,0.0000000,0.0000000,5/30/2022,5:52:20,
34.90,60.00,7.66,1.34,0.0000000,0.0000000,5/30/2022,5:52:21,
34.90,61.00,5.24,2.19,0.0000000,0.0000000,5/30/2022,5:52:22,
34.90,61.00,19.75,0.82,0.0000000,0.0000000,5/30/2022,5:52:24,
34.90,61.00,22.28,1.02,0.0000000,0.0000000,5/30/2022,5:52:25,
34.90,61.00,9.32,2.64,0.0000000,0.0000000,5/30/2022,5:52:26,
34.90,61.00,13.95,1.11,0.0000000,0.0000000,5/30/2022,5:52:27,
34.90,61.00,19.06,0.50,0.0000000,0.0000000,5/30/2022,5:52:29,
34.80,61.00,21.09,0.94,0.0000000,0.0000000,5/30/2022,5:52:30,
```

Fig no. 3.5: GPS value not found when combined

Whole wiring was done in the breadboard for reliability. As all the hardware and software problems were resolved, therefore the perforated circuit board (PCB) was designed for soldering the sensors. The soldering was done to resist vibrational damages and loose contact.An indicator LED from pin 13 of Arduino MEGA was taken out.It was to ensure that the locked GPS values are fed into SD card and also know whether the instrument is functioning properly. If the indicator LED is off, it means that theinstrument's power is out. If the LED is blinking, the SD card is not present. If the LED is on, it implies all the datas are stored into SD card.All works regarding hardware and software were done successfully. With this payload, test reading where taken and finally the payload was ready for the survey.

Temperature,Humidity,Particle10,Particle25,Latitude,Longitude,Alt
itude,Date,Time

27.50,95.00,13.12,3.65,12.9265108,77.6211395,923.60,0/0/2000,14:2
8:32.
27.40,95.00,20.97,4.80,12.9265108,77.6211395,923.60,0/0/2000,14:2
8:32.
27.40,95.00,13.66,2.91,12.9265108,77.6211395,923.60,0/0/2000,14:2
8:32.
27.50,95.00,5.86,2.06,12.9265060,77.6211395,922.80,18/6/2022,14:2
8:36.
27.50,95.00,9.72,3.80,12.9265060,77.6211395,922.80,18/6/2022,14:2
8:36.
27.30,95.00,38.23,3.54,12.9265060,77.6211395,922.80,18/6/2022,14:
28:36.
27.30,95.00,8.13,2.92,12.9265060,77.6211395,922.80,18/6/2022,14:2
8:36.
27.60,95.00,9.79,1.57,12.9264993,77.6211471,919.10,18/6/2022,14:2
8:38.
27.60,95.00,15.64,4.89,12.9264993,77.6211471,919.10,18/6/2022,14:
28:38.
27.60,95.00,6.23,2.24,12.9264993,77.6211471,919.10,18/6/2022,14:2
8:38.
27.60,95.00,9.27,0.86,12.9264993,77.6211471,919.10,18/6/2022,14:2
8:38.
27.40,95.00,18.38,6.63,12.9264993,77.6211471,919.10,18/6/2022,14:
28:38.

Fig no. 3.6: Whole combined data in SD card

For the safety of the instrument, a carrier box was make with cardboard, which was designed to suite all the component within it. Initially the box was planned to make with styrofoam material as considering for higher durability and strength. Due to bad weather, the permission to conduct actual flight was denied by Air traffic control (ATC), Indian air force and balloon department of IIA, which was the higher authorities. Due to those reasons the material was confined to cardboard. The dimension of the box was 40cm length, 20cm width and 12cm height. A hole is made at the center of the box having dimension of12 cm length, 8 cm breadth, 8 cm height for capturing the particles of volume 768 cm³. The particle counter's air inlet was kept bulged to the cavity. The payload box had 2 compartments; one, to keep the circuit with sensors and second for power bank. The antenna of the GPS module was clamped outside the box for satellite locking. Indicator LED was also kept outside in the visible region.

Dimension of the box

- Length = 40cm
- Breadth = 20 cm
- Height = 12 cm

64

Fig no. 3.7: Final payload

3.3   Ground survey

After all the laboratory works, we have done numerous testing and calibrationswhich include drop test and vibration test. Thenwe move on to ground survey. The first ground survey was taken from IIA to hostel and vise versa at 31st May 2022.The particulate mattercount of this region was taken everyday till 17th June 2022.

3.3.1   Lalbhag – Tourist area

At 4th June 2022 survey was conducted in Lalbhag Botanical park. Lalbhag botanical park is a tourist area which is spread over 0.97 square kilometers, was taken survey from 2pm to 5pm. Most of the area in Lalbhag botanical park was surveyed. The survey was interrupted by the rain at 5pm, so were forced to turn off the payload. The temperature varied from 31.1°C to 39°C and humidity varied from 56 to 69. The highest value of PM 2.5 was 984.28 microgram/m³ at 2:07pm which was at the time of arrival, the reason of this value might be due to traffic at the entrance of Lalbhag. The lowest value of PM 2.5 was 0.9 microgram/m³ at 4:39pm.The lowest value of PM 1.0 was 0.21 at 4:32pm and highest value of PM 1.0 was 70.64 at 4:36pm, there was a slight wind during that time. The average value of PM2.5 at Lalbhag is 37.43µg/m^3 and the average value of PM1.0 is 9.44 µg/m^3.

Fig no. 3.8: Surveying from Lalbhag Botanical Park

3.3.2   <u>Residential - Urban Area</u>

To find the particulate matter of our local area which consists of residential, urban and rural areas which spread in the radius of 2 KM from our hostel.

South-East area was surveyed from 4:49pm to 6:08pm at 11st June. On this day average PM2.5 value was 51.77 µg/m^3 which varied from 298.43µg/m^3 at 4:51pm to 1.63 µg/m^3 at 5:02pm, while average PM1.0 value was 11.17 µg/m^3 which varied from 89.37 µg/m^3 at 4:55pm to 1.76 µg/m^3 at 5:00pm. The routes was Madiwala – Koramangala – BBMP park – Wipro park – Madiwala police station. The temperature of the day was 32°C and humidity 70.

Fig no. 3.9 : Survey taken with Yulu in south-east area

North area was surveyed from 6:17pm to 7:48pm at 12nd June. The average value of PM2.5 of this day was 71.49 µg/m^3 which varied from 154.87 µg/m^3 at 6:58pm to 2.15 µg/m^3 at 7:30pm. While, average value of PM1.0 was 3.38 µg/m^3, which varied from 23.66 µg/m^3 at 7:12pm to 0.66 µg/m^3 at 7:10pm. The average humidity and temperature of the day was 77 and 30°C respectively. The route covered was hostel – 20th main road – forum mall.

South-West area was surveyed from 4:20pm to 6:42pm at 13rd June. On this day, the average value of PM2.5 was 91.53 µg/m^3 which varied from 536.47 µg/m^3 at 5:04pm to 1.62 µg/m^3 at 5:09pm while average value of PM1.0 was 7.37 µg/m^3 which varied from 27.54 µg/m^3 at 5:09pm to 0.7 µg/m^3 at 5:09pm. The average temperature and humidity of the day was 32°C and 75 respectively. The route surveyed was hostel – Balaji theatre – Thavarekara.

3.3.3   Traffic area - Silk board

The Karnadaka government has installed air pollution detector which measures real-time air quality index (AQI). That detector provides PM2.5, PM10, O3, NO2, SO2, CO, temperature, pressure, humidity and wind speed. In 15th June we took survey that covered the area near Silk board, where the air pollution detector was located. The primary aim was to compare our PM counter with that highly précised equipment. We visited those areas at evening time. The streets were busy with traffic. The board displayed PM10 and PM2.5 with 83 µg/m^3 and 29 µg/m^3

respectively, average standard measurement covering 24 hours was 68 µg/m^3 for PM2.5 and 100 µg/m^3 for PM10. The average value of our particle counter at that time was 26.13 µg/m^3 which was similar to the value displayed by the board. PM2.5 varied from 1021 µg/m^3 to 3.64 µg/m^3, while average value of PM1.0 was 3.344 µg/m^3 which varied from 72.81 µg/m^3 to 1.0 µg/m^3. The temperature we measured and displayed on the board was 30℃ and 22℃ respectively, while the humidity we measured and displayed was 80.62 and 76 respectively.

Fig no. 3.10: Silk board survey– Air Quality Index board

3.3.4 **Industrial area – Jigini**

The place Jigini belongs to outskirts of Bangalore. Where many industries like granite, chemicals, food export, steel, bricks etc. production unit belong. A day was dedicated for taking survey in Southern part of Bangalore which was 16th June 2022. The day had average temperature and humidity of 33℃ and 70 respectively. The time taken to cover this area was approximately 1 and half hour, that is from 10:56 am to 12:13 pm. The particle count from each industry was taken. The average value of PM2.5 was 50.36 µg/m^3 which varied from 236.52 µg/m^3 at 11:00 am to 4.13 µg/m^3 at 12:02 pm. While the average value of PM1.0 was 9.44 µg/m^3 which varied from 52.52 µg/m^3 at 11:00 am to 0.33 µg/m^3 at 11:00 am. The particle

count of this region was slightly higher than the other regions and it was due to the pollution formed from the industries.

Fig no. 3.11 : Jigini industrial area surveying

3.3.5 <u>Silent Area – SVYASA</u>

Around 12:00 pm we were completed the data collection from the industrial area Jigini and go to SVYASA, the first education yoga institution in India. This region can be remarked under silent zone. This institution is located at a hilly-forest area. So this region will have the least pollution rate compared to other regions. The average value of PM2.5 in this region is measured as 24.34 µg/m^3 which ranges from 171.73 µg/m^3 at 12:33 pm to 3.3 µg/m^3 at 12:36 pm. While the average value of PM1.0 is 5.42 µg/m^3 which ranges from 33.68 µg/m^3 at 12:41 pm to 0.44 µg/m^3 at 12:42 pm. The average temperature of this region is measured as 35°C even though it felts like 28°C   and the average humidity is about 70.

Fig no. 3.12 : surveying silent area - SVYASA

3.3.6 <u>CITY – Cubbon Park - Vidhana Soudha</u>

On 16th June, after surveying the southern part we switch to the central part of Bangalore. We took a loop around Cubbon park, Vidhana Soudha, Hudson Circle and nearby places. The average temperature of this area is 34℃ and the average humidity is 62. The average value of PM2.5 was 16.96 µg/m^3 which ranged from 485.97 µg/m^3 at 5:29pm to 1.9 µg/m^3 at 6:21pm. The average value of PM1.0 was 3.19 µg/m^3 which ranged from 56.27 µg/m^3 at 5:00pm to 0.72 µg/m^3 at 5:32pm. Since the air traffic here high ,so we expected very high values for PM counter, but obtained very low values for both PM2.5 and PM1.0. The reason for this decrease might be, the area we covered the most was inside Cubbon park which is a calm place.

Fig no. 3.13: Surveying areas around Vidhana Soudha

3.3.7   <u>Bangalore Palace – Vasanth Nagar</u>

The average value of PM2.5 at this place was around 14 µg/m^3 ranging from 102.79 µg/m^3 at 8:00pm to 1.66 µg/m^3at 7:31pm. The average value of PM1.0 was 2.61 µg/m^3 ranging from 17.85 µg/m^3 at 7:53pm to 0.63 µg/m^3 at 7:51pm. It is assumed that the decrease in the readings is due to the night time. The average temperature throughout this survey was 32℃ and the average humidity was 63.74. This area was found to be having less traffic in comparison to the other city regions. The experiment was concluded for the day due to rain.

3.3.8 Travelling

The following data corresponds to ones taken while travelling to SVYASA:

- PM 2.5- average value 28.13 µg/m^3
- PM1.0 – average value 4.52 µg/m^3
- Average humidity : 62
- Average Temperature : 38℃

Upon travelling to Hudson – Bangalore Palace via Majestic

- PM2.5- average value 15.96 µg/m^3

71

- PM1 – average value 2.84 µg/m^3

- Average humidity : 64

- Average Temperature: 34℃

Vasanth Nagar – Cubbon via Majestic had the following readings:

- PM2.5 – average value 12.76 µg/m^3

- PM1- average value 2.70 µg/m^3

- Average humidity : 66

- Average Temperature : 33

3.4   <u>**Tethered Flight**</u>

After completing the ground surveys successfully, we entered the second section of this project – **tethered flight.** We conducted two flights. On 19/6/2022, upon reaching IIA, preliminary set-up for the flights was prepared. The experiment though initially was set to be in Hoskote campus, due to various factors, it was shifted to the main campus. Balloon was filled up with hydrogen and the payload box was attached to it after double-checking its condition to check whether the weight as being balanced, whether the balloon was moving smoothly etc. Once it was checked and confirmed, for 10 minutes, the balloon was tied at a height of 50m to see whether any fluctuations occurred at lower heights. The experiment started after all these check-ups. Cameras were set-up at different places in the ground and in order to control the tethered flight, we went to the dome are in the roof top. The first flight was conducted between 12:39pm – 3:12pm. The wind speed was low as it was noon and so the balloon went up smoothly to 1041m from 841m (altitude values from sea-level). IIA had a payload box containing GM counter that was planned to attach with our payload, but unfortunately due to weight-balance issue that a single balloon can hold upto 3 (maximum), which is recommended for smooth flight. At the time of launch as to reduce weight, payload that belonged to IIA was withdrawed. The temperature throughout this tethered flight was 33℃ and humidity was 65. The average value of PM2.5 count was 44.94 µg/m^3, while the average of PM1.0 was 13.08 µg/m^3.

After taking some break, we initialized tethered flight 2 with introducing digital camera. Some images during the flight is mentioned below. We launched the flight from 3:33pm till 6:35pm. Due to high wind speed, the payload box was spinning. The movement was not as smooth as

previously launched tethered. Due to turbulence in the air, we had to control the tangential motion of the balloon by lowering and rising accordingly. The average temperature throughout the flight was 32°C and average humidity was 69.6. The average value of PM2.5 was 27.21 µg/m^3 while average value of PM1.0 was 7.74 µg/m^3. The second launch was from the top floor of the institution at the altitude 908.1m and went upto 1039.3m (from the sea-level). As the intensity of the wind increased gradually with time. So, we were forced to conclude the tethered flight by 6:35pm.

Fig no. 3.13: During tethered flight at a height of 150m from launch site.

4.   Results and discussion

Major objective of tethered balloon launch and ground survey were to observe and measure the particulate matter count at low altitudes. Based on the experiment the following are the results to be discussed. Firstly, the particulate matter counts when kept static, dynamic during the ground survey and during tethered flight datas were obtained with respect to time, latitude, longitude, altitude, temperature and humidity are plotted, studied and interpret. The results are mentioned in 2D and 3D graphical representation and pictorial representation of area surveyed with the help of Google Earth.

4.1   Static ground survey

The particulate matter datas obtained in 18th June 2022 at Hostel which is shown in figure 4.1. The latitude and longitude of the hostel obtained was 12.92346⁰ and 77.61638⁰ respectively, and altitude of 901.1m above sea-level. From this graph, we can conclude that static values are less when compared to dynamic values.
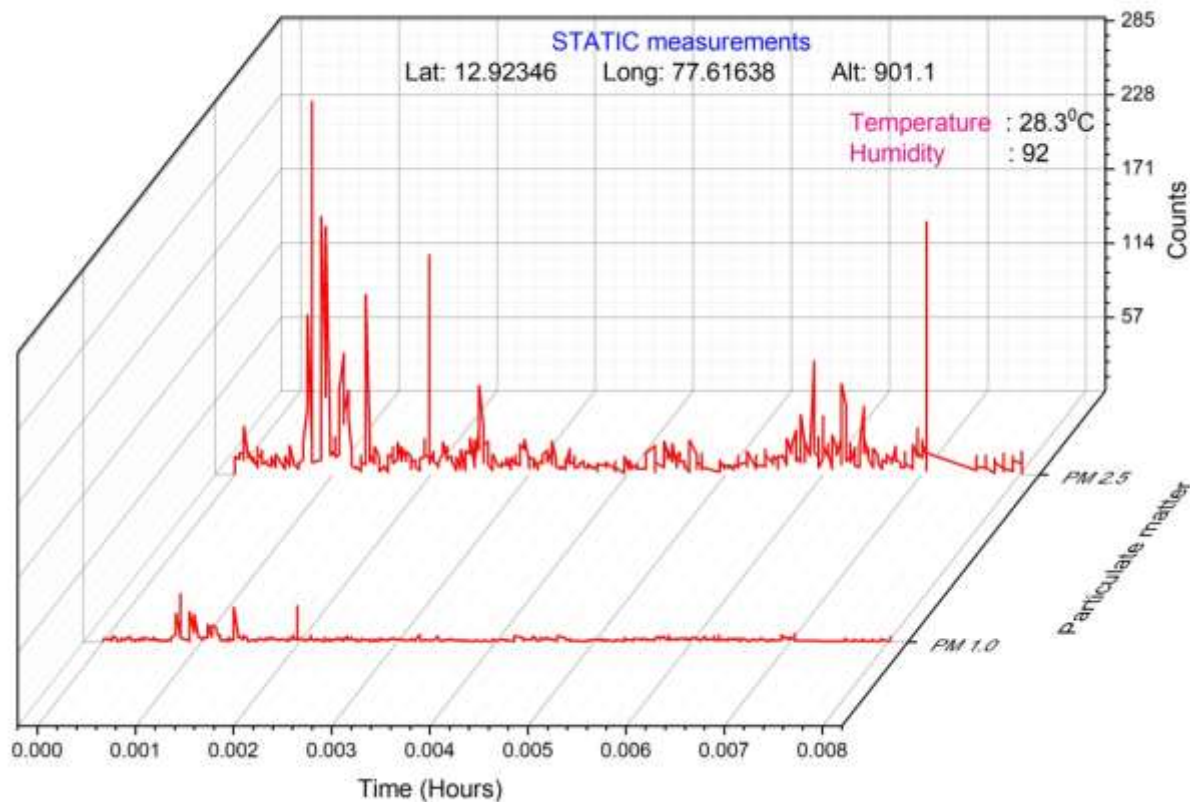

Fig 4.1 : Static measurements of Hostel

74

The particulate matter datas obtained in 17th May 2022 at IIA Balloon lab which is shown in figure 4.2. From this graph, we can find that the values of both particle count values are low due to which the values were measured in a closed, isolated room rather than exposed traffic area.
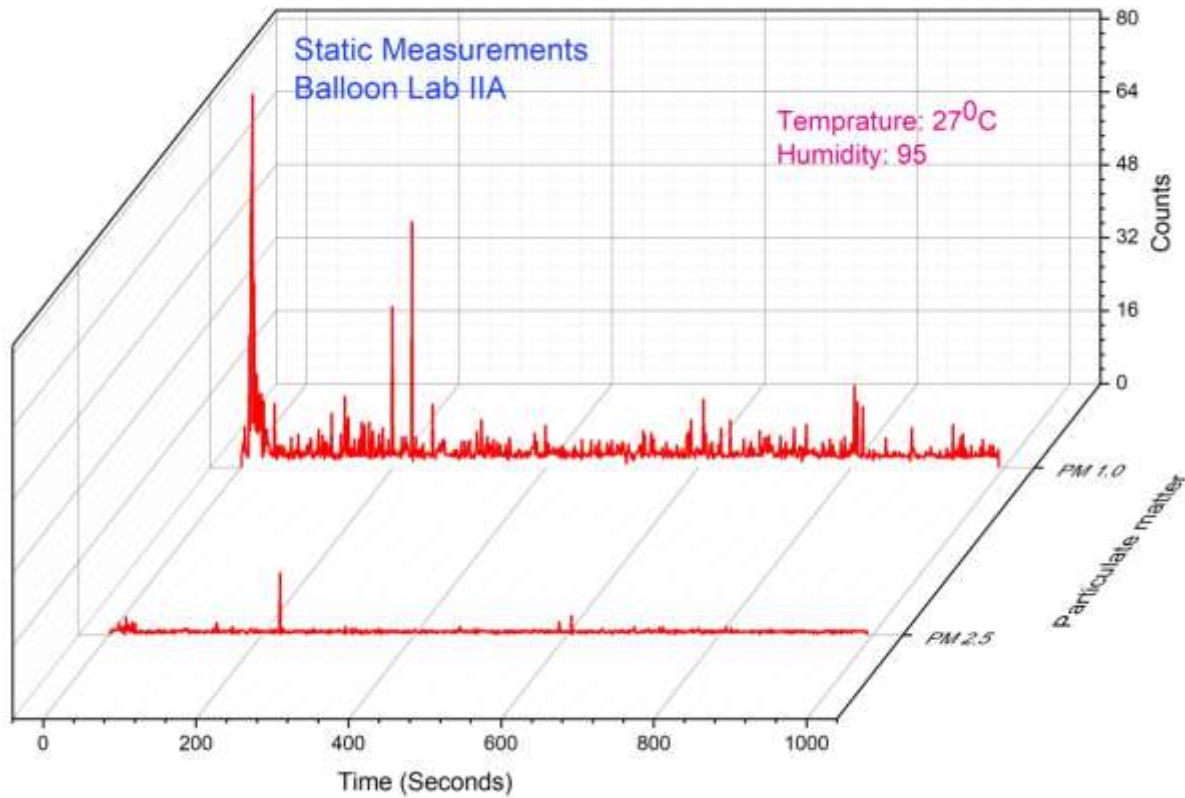

Fig 4.2 : Static balloon lab data

While observing figure 4.2, the average value of PM2.5 was 3.85 µg/m^3 and while comparing with figure 4.1, were the average value of PM1.0 is 17.66 µg/m^3. The distance between 2 locations are just 850m but the value of IIA balloon lab has dropped drastically. The reason for the difference can be the exposure of the room with the environment. The hostel is located near the Madiwala junction, so the air get polluted due to traffic, while IIA balloon lab is located within the IIA main building.

4.2   Dynamic ground survey

When the sensors are in motion, the values differ especially the particle counter. When more air pass through the cavity, the chance of counting particulate matter increases.   Some of dynamic area's graph are plotted below. For taking survey, we relied on public transport, electric vehicle (YULU, electric scooter), cycle and by walking.

4.2.1   Commercial area
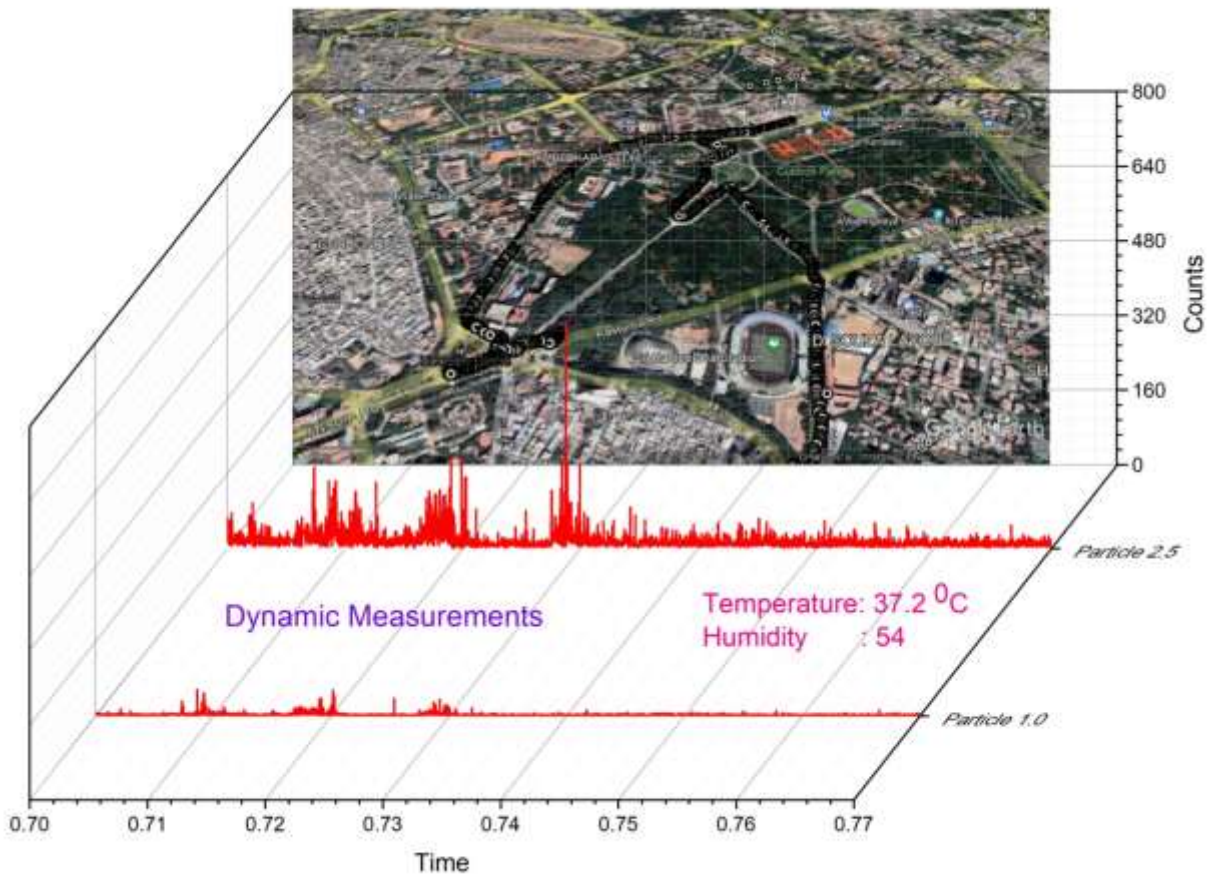

Fig 4.3 : Dynamic – Cubbon park-Vidhana Soudha

On 16th June 2022, the   average dynamic value of PM2.5 was 23.58 $\mu g/m^3$ while average value of   PM1.0 was 4.12 $\mu g/m^3$ with temperature and humidity of 37.2°C and 54 respectively. The figure 4.3 represents 2 types of information that one of it is 3D graphical representation and the other is pictorial representation of area surveyed with the help of Google Earth. The peak is marked at the value 485.97 $\mu g/m^3$ at 5:29pm. Where the pictorial representation shows the area covered while taking the survey.

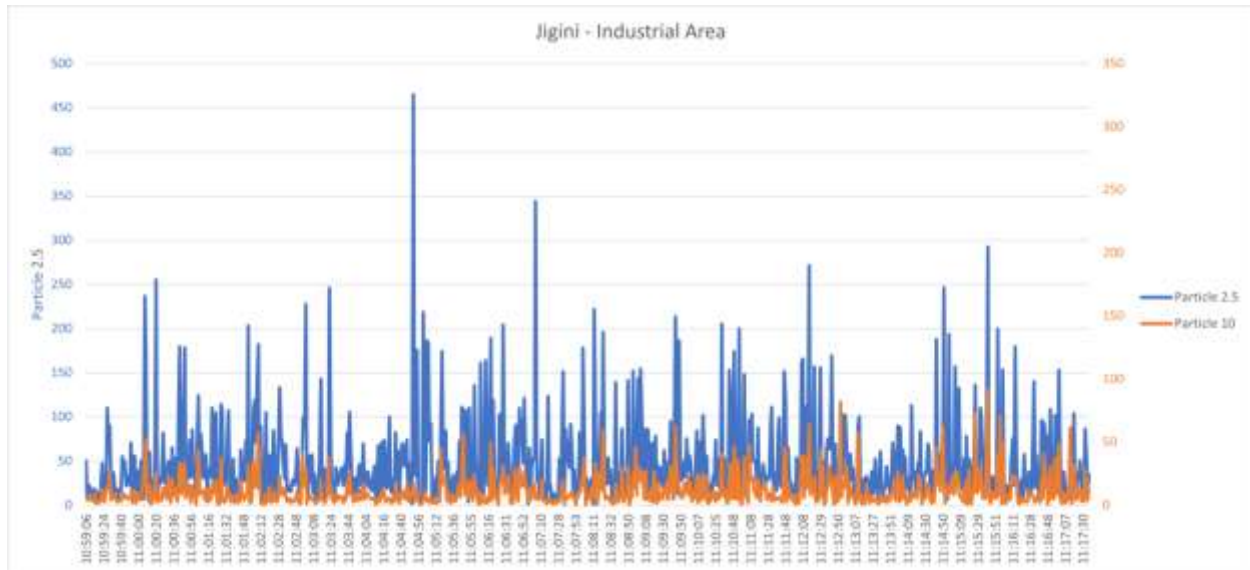## 4.2.3   <u>Industrial area</u>

Fig 4.3 : Datas collected from Industrial areas

We surveyed the industries in Jigini. As we expected the values were higher than other region. The values obtained was reasonable in which the approximate average value of PM2.5 was 48.894 µg/m^3 while average value of PM1.0 was 10.996 µg/m^3. These are observations taken from the fraction of actual datas, for convenience in graphical representation. The cause of these value are due to production and manufacturing units present there, which create includes granite cutting, production of electrical components and other chemicals.

4.2.4   <u>**Silent area**</u>
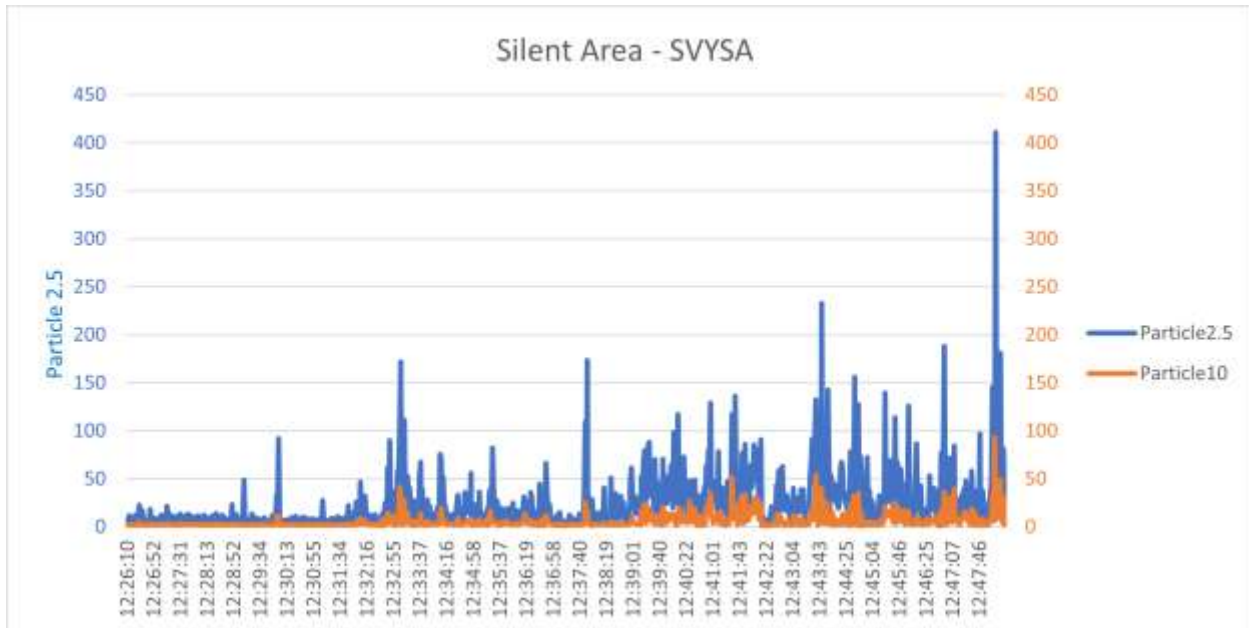
Fig 4.4 Values of PM1.0 and PM2.5 in silent area

SVYASA is a yoga university situvated near forest area in southern part of Bangalore. Figure 4.4 represent few datas collected from this university. As per this graph, the average value of PM2.5 is 23.41 µg/m^3 and average value of PM1.0 is 5.62 µg/m^3. The particle count is comparitively low due to the eco-friendly nature. The high peaks in PM2.5 might be due to sudden movements or wind at that time.

4.2.5   <u>**Residential**</u>
area
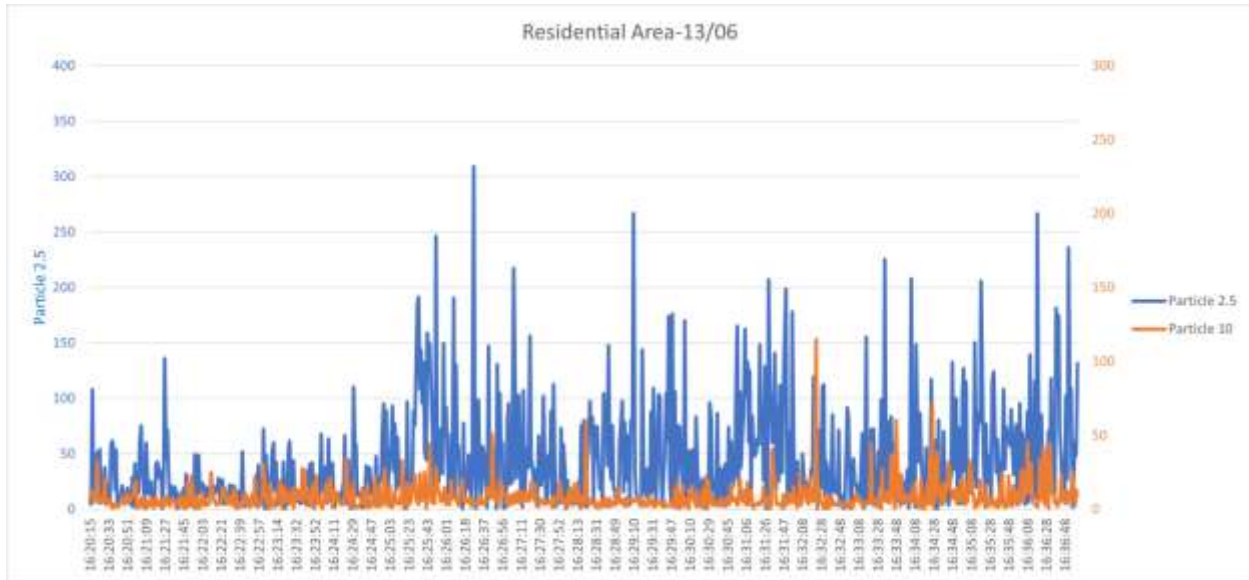
Fig 4.5 : Values of PM1.0 and PM2.5 of residential area

On 13rd June, we took survey of some residential area near hostel, covering Madiwala, Tavarekere park and Gurappanapalya junction. The figure 4.5 shows the variations in PM1.0 and PM2.5, where the particulate matter count is high due to traffic in the rural road. The surveyed time was the peak time with highest traffic. The average value of PM2.5 was 45.73 µg/m^3 while that of PM1.0 was 8.36 µg/m^3. Another reason for high value might be due to considering less datas from actual data for fitting the graph.

4.2.6 <u>**Tethered flight**</u>
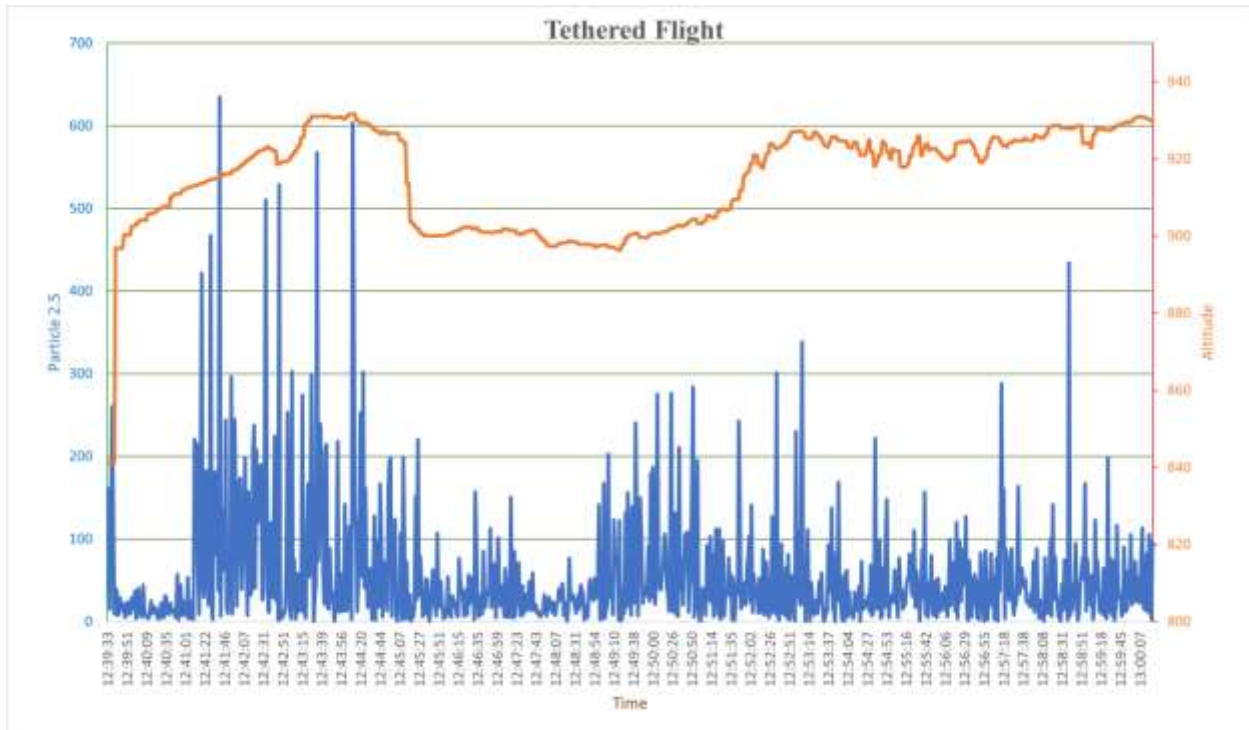
Fig 4.6 : Tethered flight althitude and PM2.5 values

Tethered flight was launched from IIA badminton court. The figure 4.6, shows only the initial values of tethered flight. It is observed that the PM2.5 values are very high, that shoot up to 634.61 $\mu g/m^3$ and had higher value to some extent. The reason for this might be the powder within the balloon, for the protection and durability of the latex balloon. Due to the pressure difference in balloon and atmosphere, the balloon raised quickly to around 935m then the wind dragged the motion.

Some arial videos and photos were taken asn some of them are shown below.

Fig no. 4.7: Arial photo taken from 1 minute to 5 minute of launch

80

Fig no. 4.9: Some photos captured during tethered flight

5.   <u>Conclusions and future works</u>

5.1   <u>Conclusion</u>

The data obtained from ground survey and tethered flight, the graph of particulate matter were plotted with different parameters. The particulate matter count we obtained were different in static and dynamic conditions. The values were varying, which shows that the particulate matter count is not a constant.

The 3 different scenarios we came through was static, dynamic and tethered flight. In which dynamic readings were taken in industrial area, silent area and residential area.

Our findings are;

- The particulate matter count in dynamic condition is relatively higher than static condition.
- In dynamic condition, Industrial area had the highest particulate matter count than all other region and the least particulate matter count was measured in silent area.
- In tethered flight, as altitude increases, the particulate matter count increases.
- Although the ground survey and tethered flight were different approach, the value of particulate matter in industrial area remained high.
- As humidity increases, the chance of measuring particulate matter decreases.

5.2   <u>Future works</u>

- In coming days, we are planning to conduct actual flight with the permission of higher authorities, which can provide more data and hence, we can analyse the particulate matter in each location. We are planning the follow;

- To complete a project with collaboration with government as part of environmental survey

- To submit the project to an international journal, to do so we need more datas

- Conduct awareness programs regarding air pollution with the observed data.

- To conduct actual flights in different parts in India and then internationally.

5.3   Reference

[1]

An Overview of High-Altitude Balloon Experiments at the Indian Institute of Astrophysics,
https://doi.org/10.48550/arXiv.1506.01538

Margarita Safonova, *Akshata Nayak*, *A. G. Sreejith*, *Joice Mathew*, *Mayuresh Sarpotdar*, *S. Ambily*, *K. Nirmal*, *Sameer Talnikar*, *Shripathy Hadigal*, *Ajin Prakash*, *Jayant Murthy*

Web Resources

https://en.wikipedia.org/wiki/High-altitude_balloon

https://www.robotshop.com/media/files/PDF/ArduinoMega2560Datasheet.pdf

https://www.arduino.cc/

https://www.farnell.com/datasheets/1682209.pdf

https://en.wikipedia.org/wiki/Raspberry_Pi

https://opensource.com/resources/raspberry-pi

http://www.haoyuelectronics.com/Attachment/DSM501A/DSM501.pdf

Images

https://aqicn.org/sensor/shinyei/

https://www.researchgate.net/figure/Diagram-of-dust-sensor-block_fig1_325548144

http://www.luftmy.net/sensor-news-273.html

https://thinkrobotics.in/products/pm2-5-dsm501a-dust-sensor

https://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/

https://www.elprocus.com/a-brief-on-dht11-sensor/

https://www.linquip.com/blog/temperature-sensors/

Data Sheet:
https://content.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf?utm_source=en%2Fimages%2Fdownloads%2FProduct_Docs%2FNEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf

Datasheet (summarized):
https://content.u-blox.com/sites/default/files/products/documents/NEO-6_ProductSummary_%28GPS.G6-HW-09003%29.pdf

NEO 6 info: https://lastminuteengineers.com/neo6m-gps-arduino-tutorial/

Neo 6Q pic and details taken from https://www.electroschematics.com/neo-6m-gps-module/

Datasheet- https://components101.com/sensors/dht11-temperature-sensor/
https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet/
Basic information-
https://components101.com/sensors/dht11-temperature-sensor

5.4   <u>Appendix</u>

```
/* for mega

 SD Card MISO=50, CLK=52, MOSI=51, CS=4

 DHT DATA=6

 PM 2(red)=3, 3(white)=5V(VCC), 4(yellow)=2, 5(orange)=GND

 GPS Rx-4, Tx-3

*/


#define DUST_SENSOR_PIN_PM10   3   //Must be the pins that

#define DUST_SENSOR_PIN_PM25   2   //support interrupts


#define INTERVAL_COUNTDOWN 1000

#define INTERVAL_READ 1000

#include <TinyGPS++.h>

#include <KarserDSM501.h>

//#include <TinyGPSPlus.h>

#include <SPI.h> //Library for SPI communication (Pre-Loaded into
Arduino)
```

```
#include <SD.h> //Library for SD card (Pre-Loaded into Arduino)

#include "DHT.h"

#define DHTTYPE DHT11 // DHT 11


uint8_t DHTPin = 6;

DHT dht(DHTPin, DHTTYPE);

float Temperature;

float Humidity;

float PM_10;

float PM_25;

float GPS;

float lat;

float lon;

const int chipSelect = 53; //SD card CS pin connected to pin 4 of Arduino


// ISRs forward declaration

void pm10_handleInterrupt();

void pm25_handleInterrupt();

// init pm10 and pm25 instances

KarserDSM501 pm10(DUST_SENSOR_PIN_PM10, pm10_handleInterrupt);

KarserDSM501 pm25(DUST_SENSOR_PIN_PM25, pm25_handleInterrupt);

// handle ISRs

void pm10_handleInterrupt() { pm10.handleInterrupt(); }
```

```
void pm25_handleInterrupt() { pm25.handleInterrupt(); }


unsigned long timer = 0;

TinyGPSPlus gps;


TinyGPSCustom pdop(gps, "GNGLL", 1); // $GPGSA sentence, 15th element
TinyGPSCustom hdop(gps, "GNGLL", 3); // $GPGSA sentence, 16th element
TinyGPSCustom hdot(gps, "GNGLL", 5); // $GPGSA sentence, 5th element
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);

  Serial.begin(9600);

  Serial1.begin(9600);

  pinMode(DHTPin, INPUT);

  pinMode(chipSelect,OUTPUT);

  dht.begin();

  Initialize_SDcard();

}
char raw_data[500];



void loop()
```

```
{
  Write_SDcard();

  Read_TempHum();

  Read_PM();

  char v;

  while (Serial1.available()>0)

  { v=Serial1.read();

    gps.encode(v);

    Serial.print(v);

  }
// check if gps is connected
 /* if (millis() > 5000 && gps.charsProcessed() < 10)

  {

    Serial.println(F("No GPS detected: check wiring."));

    while(true);

  }*/

  //LED off if gps data not correct

  if(int(atof(pdop.value()))/100) < 1) digitalWrite(LED_BUILTIN, LOW);

  else digitalWrite(LED_BUILTIN, HIGH);


  print_gps_data();

  delay(1000);      //Wait for 1 seconds before writing the next data

}
```

```
/*

Read the particel counter

*/

void Read_PM()

{


  if (millis() >= timer + INTERVAL_READ)

    {

      timer += INTERVAL_READ;

      PM_10 = pm10.readPM();

      PM_25 = pm25.readPM();

      Serial.println("pm10: "+String(PM_10)+" pm2.5: "+String(PM_25));

    }


}




void print_gps_data()

{
```

```
    //Latitude

    Serial.print(atof(pdop.value())/100,7);

    Serial.print(",");

    //Longitude

    Serial.print(atof(hdop.value())/100,7);

    Serial.print("\n");

    //Date

    Serial.print(gps.date.month());

    Serial.print(F("/"));

    Serial.print(gps.date.day());

    Serial.print(F("/"));

    Serial.print(gps.date.year());

    Serial.print("\n");

    //Time

    Serial.print(gps.time.hour());

    Serial.print(F(":"));

    Serial.print(gps.time.minute());

    Serial.print(F(":"));

    Serial.print(gps.time.second());

    Serial.print(F("."));

    Serial.print("\n");

}
```

```
/*

Read the temperature and humidity data

*/


void Read_TempHum()

{

    Temperature = dht.readTemperature();

    Humidity = dht.readHumidity();

    Serial.print("Temperature = ");

    Serial.println(Temperature);

    Serial.print("Humidity = ");

    Serial.println(Humidity);


}



/*

Initialise SD card

*/

void Initialize_SDcard()

{

    // see if the card is present and can be initialized:
```

```
//Blink if sd card is not initialised

while (!SD.begin(chipSelect))

{

    Serial.println("Card failed, or not present");

    digitalWrite(LED_BUILTIN, HIGH);     // turn the LED on (HIGH is the
voltage level)

    delay(500);                               // wait for a second

    digitalWrite(LED_BUILTIN, LOW);      // turn the LED off by making
the voltage LOW

    delay(500);

}

 delay(1000);

 digitalWrite(LED_BUILTIN, HIGH);

 // open the file. note that only one file can be open at a time,

 // so you have to close this one before opening another.

 File dataFile = SD.open("LoggerCD.txt", "w+");

 // if the file is available, write to it:

 if (dataFile)

 {

dataFile.println("Temperature,Humidity,Particle10,Particle25,Latitude,Long
itude,Date,Time"); //Write the first row of the excel file


    dataFile.close();
```

```
    }

}



/*

Write to SD card

*/

void Write_SDcard()

{

    // open the file. note that only one file can be open at a time,

    // so you have to close this one before opening another.

    File dataFile = SD.open("LoggerCD.txt", FILE_WRITE);

    // if the file is available, write to it:

    if (dataFile)

    {


      dataFile.print(Temperature); //Store date on SD card

      dataFile.print(","); //Move to next column using a ","

      dataFile.print(Humidity); //Store date on SD card

      dataFile.print(","); //Move to next column using a ","

      dataFile.print(PM_10); //Store date on SD card

      dataFile.print(","); //Move to next column using a ","

      dataFile.print(PM_25); //Store date on SD card
```

94

```
      dataFile.print(","); //Move to next column using a ","

      dataFile.print(atof(pdop.value())/100,7); //Store date on SD card

      dataFile.print(","); //Move to next column using a ","

      dataFile.print(atof(hdop.value())/100,7); //Store date on SD card

      dataFile.print(","); //Move to next column using a ","

      dataFile.print(gps.date.day());

      dataFile.print("/");

      dataFile.print(gps.date.month());

      dataFile.print("/");

      dataFile.print(gps.date.year());

      dataFile.print(","); //Move to next column using a ","

      dataFile.print(gps.time.hour());

      dataFile.print(":");

      dataFile.print(gps.time.minute());

      dataFile.print(":");

      dataFile.print(gps.time.second());

      dataFile.print(","); //Move to next column using a ","

      dataFile.println(); //End of Row move to next row

      dataFile.close(); //Close the file

    }

   else

   Serial.println("SD card writing failed");

  }
```