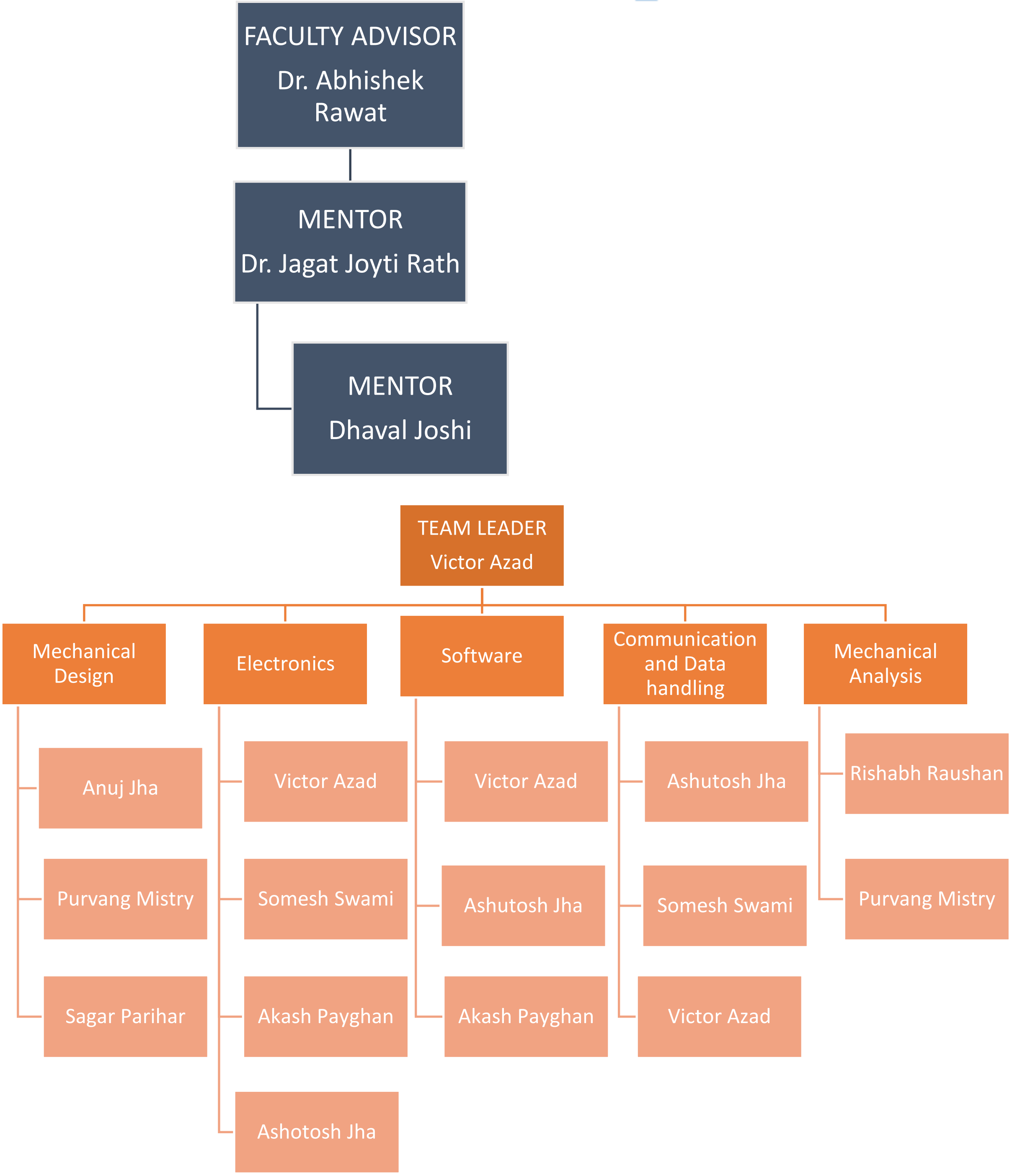
CANSAT 2022-23

CRITICAL DESIGN REVIEW (CDR)

TEAM ID: 2022ASI-002

TEAM NAME: ASTROPEEP

Team Description



FACULTY ADVISOR
Dr. Abhishek Rawat

MENTOR
Dr. Jagat Joyti Rath

MENTOR
Dhaval Joshi

TEAM LEADER
Victor Azad

Mechanical Design
- Anuj Jha
- Purvang Mistry
- Sagar Parihar

Electronics
- Victor Azad
- Somesh Swami
- Akash Payghan
- Ashotosh Jha

Software
- Victor Azad
- Ashutosh Jha
- Akash Payghan

Communication and Data handling
- Ashutosh Jha
- Somesh Swami
- Victor Azad

Mechanical Analysis
- Rishabh Raushan
- Purvang Mistry

List of Acronyms

Acronyms	Meaning
3D	3-Dimensional
A	Analysis
cm	Centimeter
D	Demonstration
dB	Decibel
EPS	Electrical power Subsystem
FRR	Flight Readiness Review
FSW	Flight Software
GCS	Ground Control System
gm	Gram
GPS	Global Positioning System
GS	Ground Station
GUI	Graphical User Interface
Hz	Hertz
I	Inspection

I2C	*Inter-Integrated Circuit*
IC	*Integrated Circuits*
LED	*Light Emitting Diode*
PC	*Personal Computer*
QHA	*Quasi Harmonic Function*
RTC	*Real Time Clock*
T	*Test*
VM	*Verification Method*
GHz	*Giga Hertz*
s	*Second*
csv	*Comma separated values*
hrs	*Hours*
PCB	*Printed Circuit Board*
mAh	*Milli ampere hour*
m	*Meter*
V	*Volts*
FSS	*Flight Software State*

System Overview

MISSION SUMMARY

- ***Innovative Mechanical Gyro-control system that shall demonstrate the descent control of the CANSAT.***
- ***CANSAT descent control system that shall open at an altitude of 500 m.***

CANSATs will be launched to an altitude of 800.0 m to 900.0 m from the ground level and above the launch site & deployed near the peak altitude. During the ejection from the rocket orientation of the CANSAT is not controlled. The CANSAT must remain intact during the course of the entire mission and send the data to the ground station through a telemetry link.

Sr.no.	*Requirement*	*Priority*	*Fulfillment*	*VM*			
				A	*I*	*T*	*D*
1	*Total mass of the CANSAT shall be under 0.700 kg (+/- 0.050 kg)*	*High*	*Mass of Cansat is within limits.*	*x*			*X*
2	*CANSAT shall fit in a cylindrical body of 0.125 m diameter x 0.310 m height.* *Tolerances are to be included to facilitate container deployment from the rocket fairing.*	*High*	*Cansat is designed with appropriate dimensions.*	*X*			*X*
3	*Any sharp edges on the container body shall be avoided as it can cause interfere during the CANSAT ejection from the rocket.*	*High*	*No sharp Edges are present.*	*X*			*X*

4	*Color of the CANSAT body shall be fluorescent i.e., pink, red or orange, and shall embody the Indian flag.*	*High*	*Color will be fluorescent and will embody Indian Flag.*				*X*
5	*The CANSAT shall consist of necessary sensors to provide the following mandatory Real-time datasets: Position data, altitude, pressure, temperature, orientation data, power data & system status.*	*High*	*Cansat has necessary sensors required for measuring mandatory data.*	*X*			*X*
6	*Each data field shall be displayed in real-time on the ground station user interface/software.*	*High*	*GUI will be developed for displaying Data.*				
7	*CANSAT shall also record the data and save it into an onboard SD card in case of telemetry connection loss.*	*High*	*The data shall be recorded in SD Card.*				
8	*All electronics shall be enclosed and shielded from the environment. No electronics can be exposed except for sensors. There must be a structural enclosure.*	*High*	*No Electronics is exposed.*	*x*			*X*
9	*CANSAT structure shall be built to survive 15 Gs of launch acceleration & 30 Gs of shock.*	*High*	*CANSAT will survive required shock.*				*X*
10	*The CANSAT shall have an external power switch with an indicator light or sound for being turned on or off, in order to avoid the de-assembling of CANSATs on the launch pad.*	*High*	*CANSAT has an external power switch.*	*x*			*X*
11	*The CANSAT shall contain a total of 2 descent control mechanisms, to be used at different stages while descent.*	*High*	*has 2-descent control mechanism.*				*x*

12	***CANSAT shall immediately deploy the first parachute after ejection from the rocket.***	***High***	***1st parachute will be immediately deployed after ejection.***				***X***

Sensor Systems Summary and Test results

Selected Component	Type	Function	Interface
Adafruit BMP 388	Pressure and temperature sensor	Measures air pressure and temperature both in the payload and the container	I2C
Matek SAM M8Q	GPS	Gets payload coordinates (latitude and longitude), GPS time, and GPS satellite	Serial
Voltage divider	Voltage Sensor	Measures voltage of the payload's battery	ADC
MPU6050	Acceleration and Orientation Sensor	Orientation and Acceleration Measurement	ADC

Sensor	Model	Function
Altimeter Sensor	BMP280	Altitude Measurement
Air Pressure and Air Temperature Sensor	BMP280	Pressure and Temperature Measurement
Acceleration and Orientation Sensor	MPU6050	Orientation and Acceleration Measurement
GPS Receiver Sensor	Matek SAM M8Q	Latitude, Longitude, Altitude, Time, Satellites number Measurement

Voltage Sensor	***Voltage Divider , CPU's ADC Converter***	***Power Status***

Payload Air Pressure Sensor Summary (1 of 2)

Sensor-Adafruit BMP 280

Pressure Range (hPa)	*Operating Voltage (V)*	*Weight (g) / Dimension (mm)*	*Current Consumption (uA)*	*Resolution (Pa)*	*Accuracy (hPa)*	*Interface*	*Data Format*
300-1250	*1.65 – 3.6*	*1.2 21.6 x 16.6*	*2.7*	*0.016*	*±0.08*	*I2C, SPI*	*Float XXXXX (Pa)*

Altitude above sea level can be calculated from Barometric Equation below. However, output from sensor is very noisy so that a kalman filter will be used to estimate the pressure value.

Barometric Equation

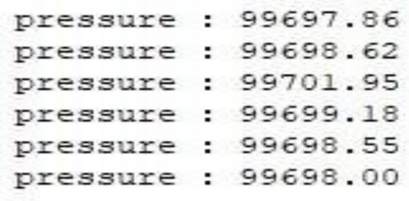
$$h = 44330 \times \left(1 - \left(\frac{P}{P_0}\right)^{\frac{1}{5.255}}\right)$$

h = calculated altitude $[m]$

P = sensed pressure $[Pa]$

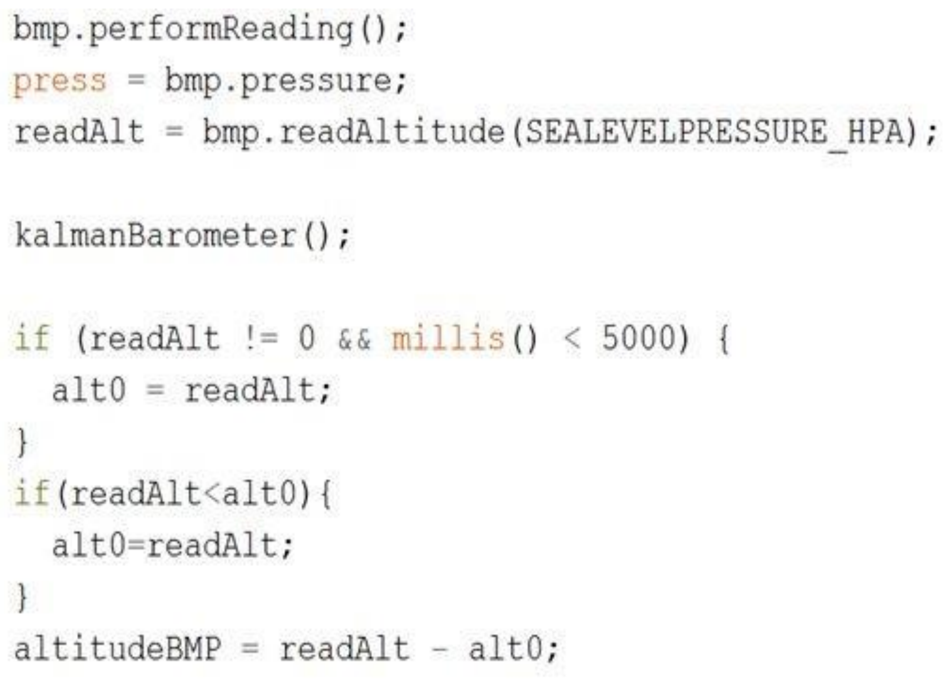
Po = pressure at ground level $[Pa]$

Sample Output



```
pressure : 99697.86
pressure : 99698.62
pressure : 99701.95
pressure : 99699.18
pressure : 99698.55
pressure : 99698.00
```

Data Processing

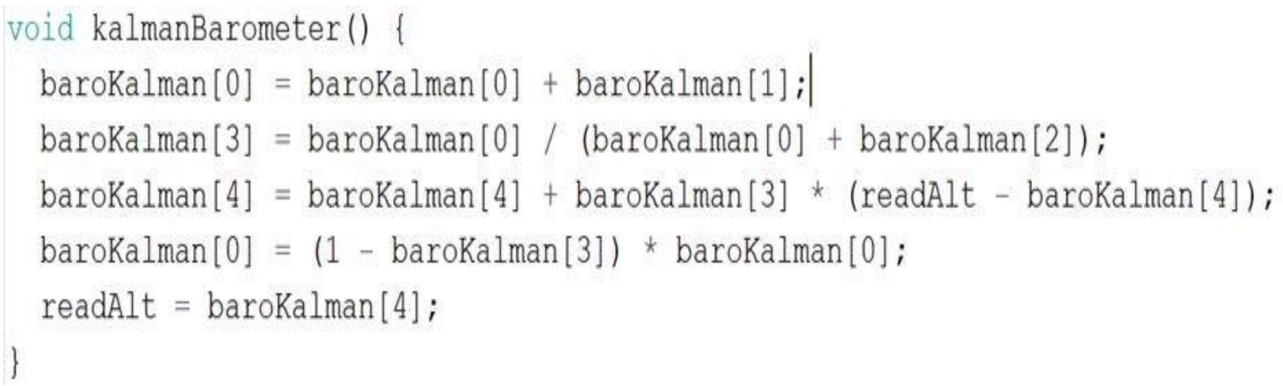


```
bmp.performReading();
press = bmp.pressure;
readAlt = bmp.readAltitude(SEALEVELPRESSURE_HPA);

kalmanBarometer();

if (readAlt != 0 && millis() < 5000) {
  alt0 = readAlt;
}
if(readAlt<alt0){
  alt0=readAlt;
}
altitudeBMP = readAlt - alt0;
```

Kalman Filter



```
void kalmanBarometer() {
  baroKalman[0] = baroKalman[0] + baroKalman[1];
  baroKalman[3] = baroKalman[0] / (baroKalman[0] + baroKalman[2]);
  baroKalman[4] = baroKalman[4] + baroKalman[3] * (readAlt - baroKalman[4]);
  baroKalman[0] = (1 - baroKalman[3]) * baroKalman[0];
  readAlt = baroKalman[4];
}
```

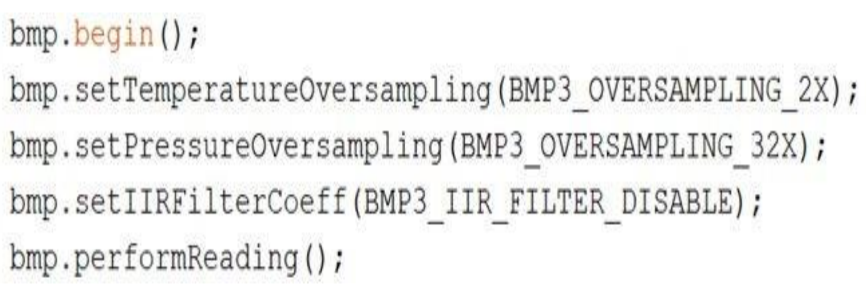
Payload Air Pressure Sensor Summary (2 of 2)

Temperature data will be collected and processed with the help of Adafruit_BMP3XX library. No further processing is needed as the reading is very stable.

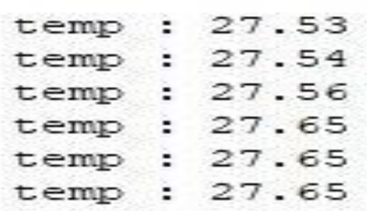
Sensor				Adafruit BMP 280			
Temperature Range (°C)	**Operating Voltage (V)**	**Weight (g) / Dimension (mm)**	**Current Consumption (uA)**	**Resolution (Pa)**	**Accuracy (hPa)**	**Interface**	**Data Format**
0 – 65	1.65 – 3.6	1.2 21.6 x 16.6	2,7	0.016	±0.08	I2C, SPI	Float XX.XX (C)

Data Processing

Sample Output



```
bmp.begin();
bmp.setTemperatureOversampling(BMP3_OVERSAMPLING_2X);
bmp.setPressureOversampling(BMP3_OVERSAMPLING_32X);
bmp.setIIRFilterCoeff(BMP3_IIR_FILTER_DISABLE);
bmp.performReading();
```



```
temp : 27.53
temp : 27.54
temp : 27.56
temp : 27.65
temp : 27.65
temp : 27.65
```

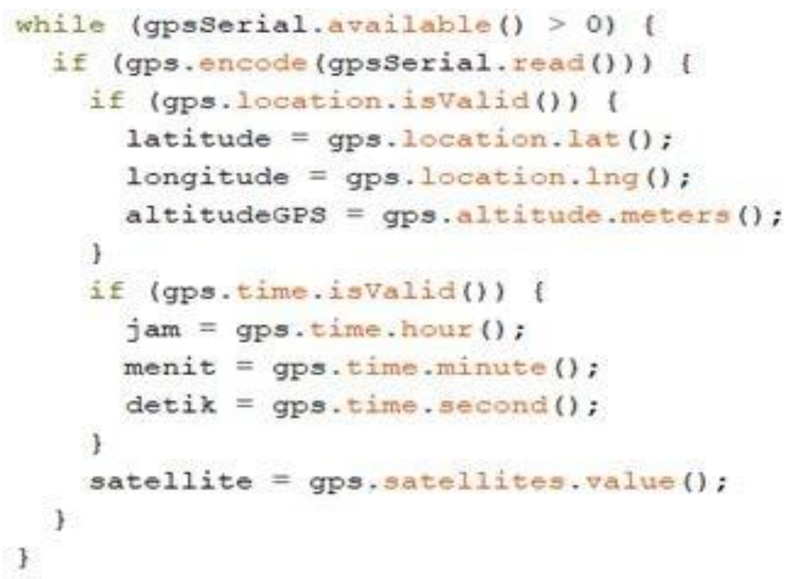
GPS Sensor Summary

Sensor Matek SAM M8Q + Compass

Tracking Sensitivity (dBm)	Operating Voltage (V)	Weight (g) / Size (mm)	Current Consumption (mA)	Channel	Accuracy (m)	Interface	Update Rate (Hz)	Data Format
-165	4-6	7 20 x 20 x 10	29	72	~2.5	UART	18	Float and Integer

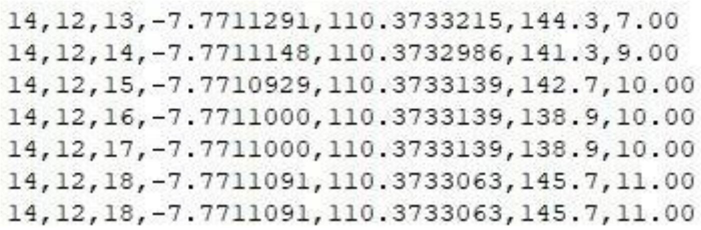
Longitude, latitude, and the other data will be collected and processed with the help of Tiny GPS Plus library by Mikal Hart. This sensor can lock onto GPS satellite quickly.

Data Processing



```
while (gpsSerial.available() > 0) {
  if (gps.encode(gpsSerial.read())) {
    if (gps.location.isValid()) {
      latitude = gps.location.lat();
      longitude = gps.location.lng();
      altitudeGPS = gps.altitude.meters();
    }
    if (gps.time.isValid()) {
      jam = gps.time.hour();
      menit = gps.time.minute();
      detik = gps.time.second();
    }
    satellite = gps.satellites.value();
  }
}
```

Sample Output



```
14,12,13,-7.7711291,110.3733215,144.3,7.00
14,12,14,-7.7711148,110.3732986,141.3,9.00
14,12,15,-7.7710929,110.3733139,142.7,10.00
14,12,16,-7.7711000,110.3733139,138.9,10.00
14,12,17,-7.7711000,110.3733139,138.9,10.00
14,12,18,-7.7711091,110.3733063,145.7,11.00
14,12,18,-7.7711091,110.3733063,145.7,11.00
```

Left to right:
Hour, minute, second, latitude, longitude, altitude, satellite

Source: https://github.com/mikalhart/TinyGPSPlus/blob/master/src/TinyGPS%2B%2B.h

Payload Voltage Sensor Summary

Name	Range (V)	Error rate (%)	Interface	Data Format
Processor Analog Pin (Voltage Divider)	0 - 5	0.03	ADC	Float X.XX (Volt)

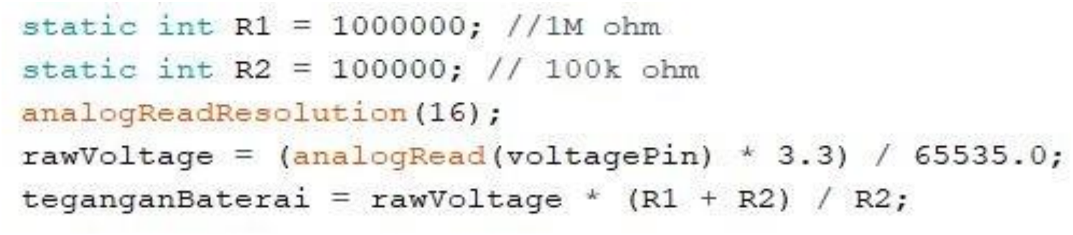
Battery voltage is measured using the ADC port trough a voltage divider, the following resistors are used in the circuit.

R1 = 1 MΩ

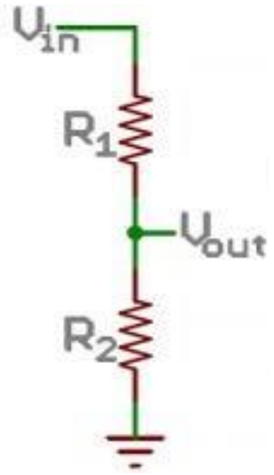
R2 = 100 kΩ

The ATSAMD Processor ADC has maximum resolution of 32-bit with 3.3 volt analog reference, but the resolution used is 16-bit resolution. So, the maximum accuracy of this on board voltage sensor is 50 uV..

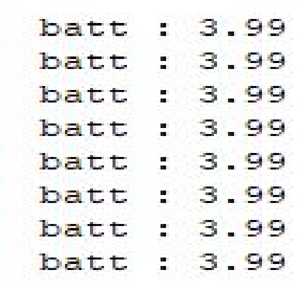
Data Processing



```
static int R1 = 1000000; //1M ohm
static int R2 = 100000; // 100k ohm
analogReadResolution(16);
rawVoltage = (analogRead(voltagePin) * 3.3) / 65535.0;
teganganBaterai = rawVoltage * (R1 + R2) / R2;
```



Sample Output



```
batt : 3.99
batt : 3.99
batt : 3.99
batt : 3.99
batt : 3.99
batt : 3.99
batt : 3.99
batt : 3.99
```

Gyroscope Sensor Selection Summary :

Sensor	Range (°/s)	Resolution (bit)	Accuracy (°/s)	Data Interface	Operating Voltage (V)	Cost (Rs.)	Dimension (mm)	Weight (gm)
MPU6050	±250	16	1.9	I2C/IIC	3-5	120	4 X 4 X 0.9	3

Reason for selecting MPU6050 :

- Light weight, small size and Low Cost.
- Using same sensor for acceleration and orientation measurement will reduce the cost and weight of Cansat.

Flight Software design

Overview of the CanSat FSW Design

- The CanSat will collect sensor data then save to SD card and send to ground station via XBee.
- The CanSat will deploy 2nd Parachute deploy mechanism after reaching 500m altitude.
- The buzzer will keep beeping after landing until turned off with power switch.

Programming Language

- C/C++ for CanSat container
- Python/C++ for ground station

Development Environment

- Arduino IDE
- Visual Studio Code
- Spyder

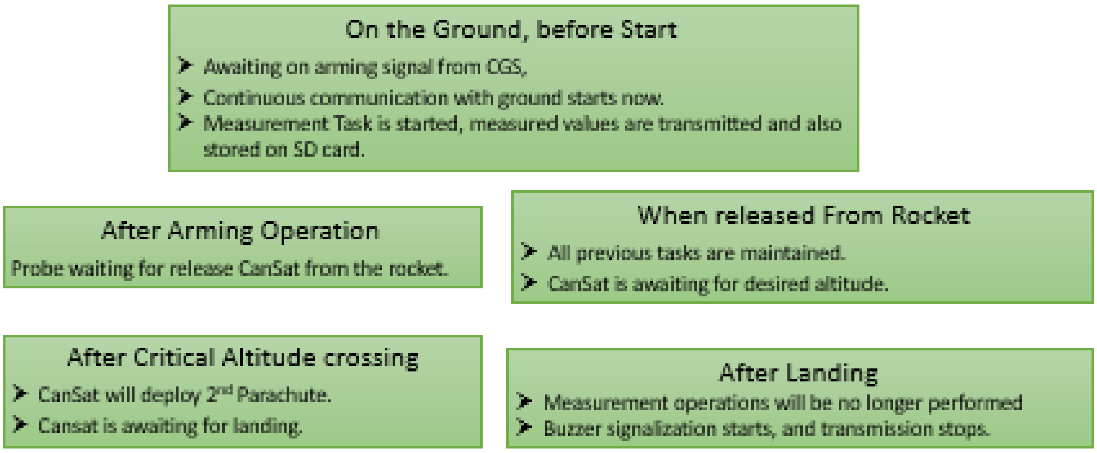
There is no change since PDR.

Calculation :

- Transmitted Power : 8 dB
- Transmitter Antenna Gain : 7 dB
- Transmitter Loss : 1 dB
- Free Space Loss : 110.14 dB
- Miscellaneous Loss : 1 dB
- Receiver Antenna Gain : 14 dB
- Receiver Loss : 1 dB
- Received Power : -84.14 dB
- Receiver Sensitivity : -101 dB
- Link Margin : 16.86 dB

Formula used for Received Power Calculation :

$$P_{out} = P_t + G_t - L_t - L_{fs} - L_m + G_r - L_r$$



On the Ground, before Start

- Awaiting on arming signal from CGS,
- Continuous communication with ground starts now.
- Measurement Task is started, measured values are transmitted and also stored on SD card.

After Arming Operation

Probe waiting for release CanSat from the rocket.

When released From Rocket

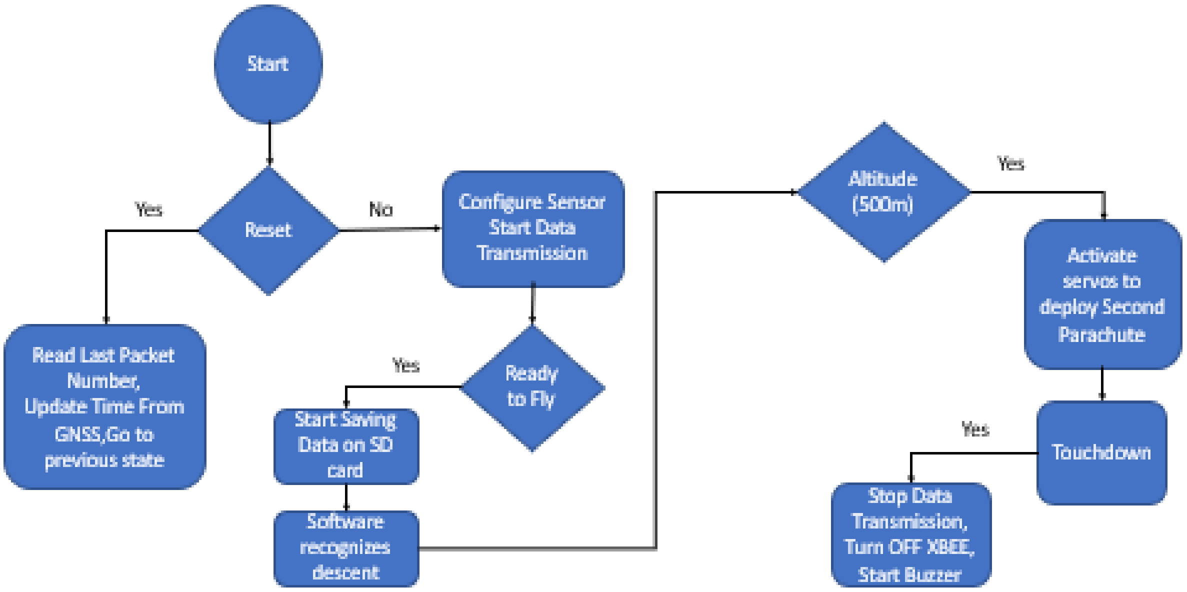
- All previous tasks are maintained.
- CanSat is awaiting for desired altitude.

After Critical Altitude crossing

- CanSat will deploy 2nd Parachute.
- Cansat is awaiting for landing.

After Landing

- Measurement operations will be no longer performed
- Buzzer signalization starts, and transmission stops.


Start
Reset
Yes
No
Read Last Packet Number, Update Time From GNSS,Go to previous state
Configure Sensor Start Data Transmission
Ready to Fly
Yes
Start Saving Data on SD card
Software recognizes descent
Altitude (500m)
Yes
Activate servos to deploy Second Parachute
Touchdown
Yes
Stop Data Transmission, Turn OFF XBEE, Start Buzzer

As far as power management goes, all devices are ON for all time except two (Servo motor for Deployment of 2nd Parachute and Buzzer for recovery) as Servo motor will deployed only for some time after crossing critical altitude (500m) and Buzzer will be turned ON after landing for location and recovery of CanSat.

After reset of processor in flight mode, previous state of mission together with last packet count can be retrieved from flash.

This information with current GNSS time is sufficient to continue mission.

GCS Design and Overview

Overview

- Main Computer is a laptop running GS application. The application is configured in GUI so that an appropriate port and baud rate can be selected.
- QHA antenna receives data from the probe (i.e. CanSat CPU) and transmits commands.
- Xbee PRO S2C module forwards and receive data to form communication link between CanSat and GCS.
- GS application, written in C++/Python, saves and displays data. It also saves received data in a local area network.

Data Flow and Components

- The antenna receives data over radio from the probe.
- Using Xbee explorer cable , forwards data to the Xbee module.
- Xbee PRO S2C module forwards data to GS's Main Computer using micro USB to USB cable.
- Software developed in C++ parses and then displays data in engineering units and save them into a CSV file.
- Main computer is capable of sending predefined commands to the probe.

Specification

- Ground Station can operate on battery at ease up to 4 hours and longer with an extra power bank.
- Ground Station team member carries an umbrella to cast a shadow on the main computer in case of a sunny day.

- Before start the main computer is kept in a special thermal case.
- Main Computer runs Windows offline and all possible automatic updates are turned off.

Overview of the GCS Software Design

- The GUI will collect sensor data from Cansat and send command to CanSat via XBee.
- The GUI will do real time plotting of data using matplotlib.

- The GUI will also contain terminal tabs for sending predefined and custom commands to CanSat.

Programming Language

- Python/C++

Development Environment

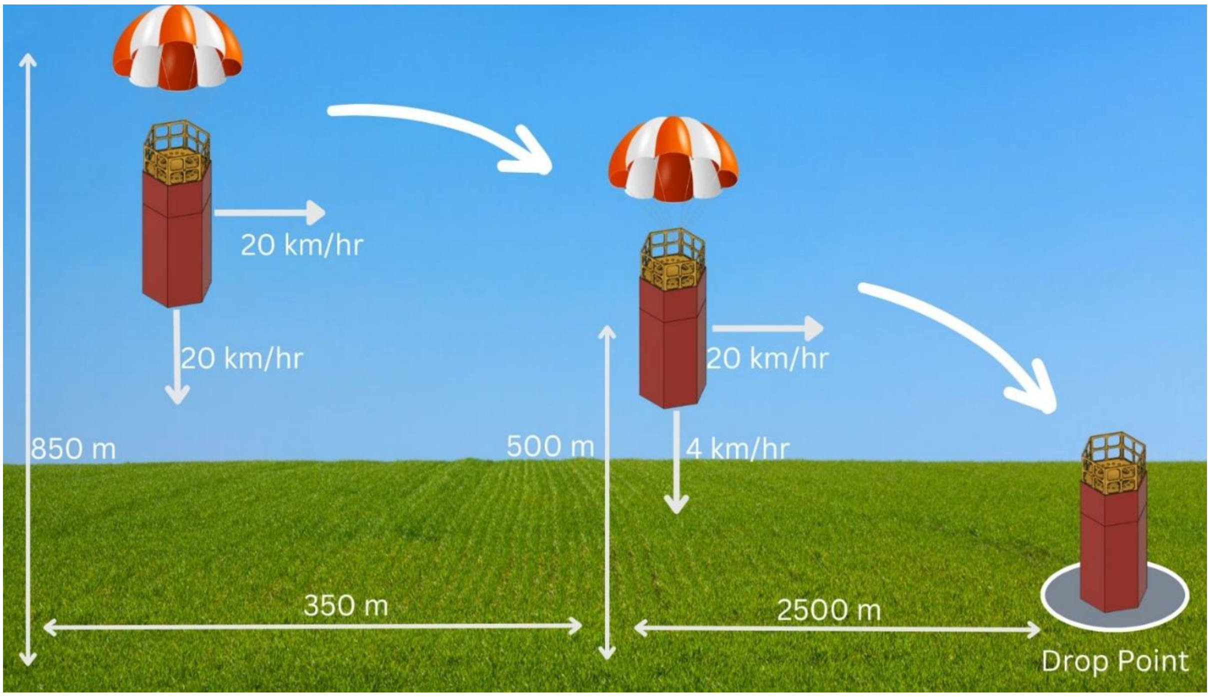
- Visual Studio Code
- Spyder

Software specifics

- Real-time data is received via micro USB-USB cable. Application is analyzing the beginning and end of each telemetry frame. After being received, a frame is divided into parts and directed into the data flow and then the whole frame or its part is sent to software listeners like in e.g. Database module or a real time chart module.

- GUI is equipped with indicators which inform about mission phase and onboard computer system state.
- Real-time plotting software relies on matplotlib chart components which are configured to listen to data.
- Matplotlib charts are optimized for handling real time data.
- Command software and interface is included in the application as the Terminal Tab.
- It is possible to send predefined and custom commands to the probe with the Terminal Tab.
- The CSV file is created by writing every telemetry frame with an End Of Line sign at the end to a file with a proper name.

Descent Control


20 km/hr
20 km/hr
850 m
500 m
20 km/hr
4 km/hr
350 m
2500 m
Drop Point

Actuator Summary

Block Diagram of Altitude control loop and Actuator control loop

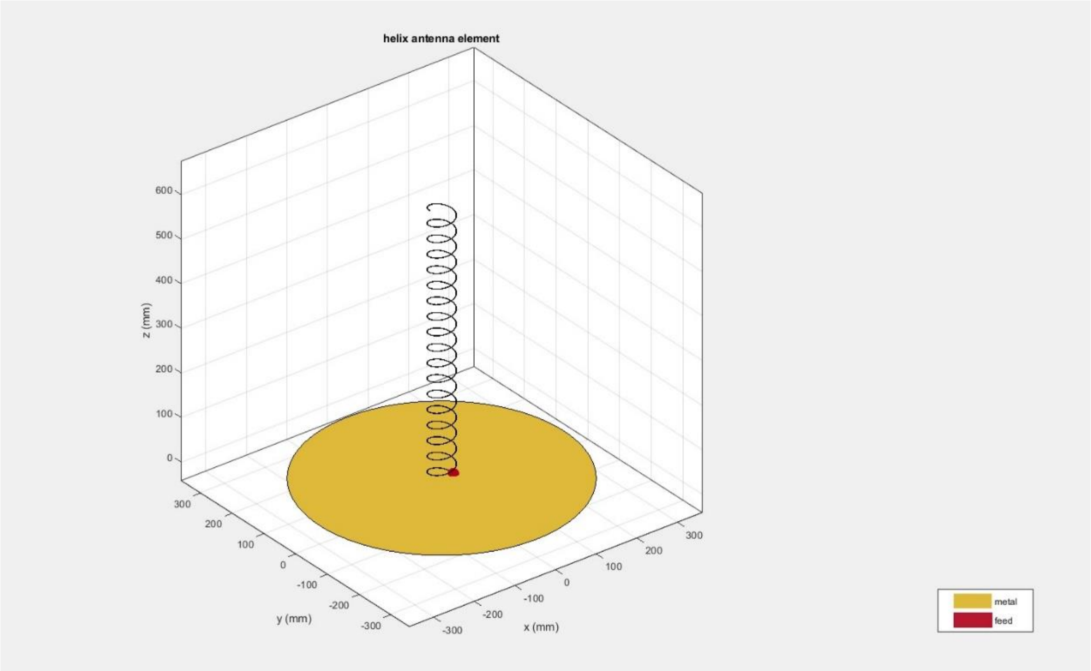
Payload Subsystem

Housekeeping Subsystem

Changes made from PDR stage

GCS Design and Overview

Antenna



Subsystem level testing and results

CANSAT Integrated module simulated results

AstroPeep

Team Id:2022ASI-002

Environmental test results

Design details

CANSAT algorithm

TTLC link margin

Ground Station final test results

CANSAT Integration and Testing

AstroPeep

Team Id:2022ASI-002

Mission Operation and Analysis

Team Id:2022ASI-002

AstroPeep

Pre-flight requirement-check-analysis

Team Id:2022ASI-002

AstroPeep

Logistics and Transportation

CANSAT ready to launch and final comments

Conclusion