# **Communication and Data Handling Subsystem Design**

|  |  |  |
| --- | --- | --- |
| **PDR** | **CDR** | **Rationale** |
| Main processor was STM32F103 | Main processor is changed to ATMega-328. | Due to problems faced in programming STM32F103 and because of previous experience of team in working with ATMega-328 and also meets all the requirements. |

### Below is the communication Subsystem overview :

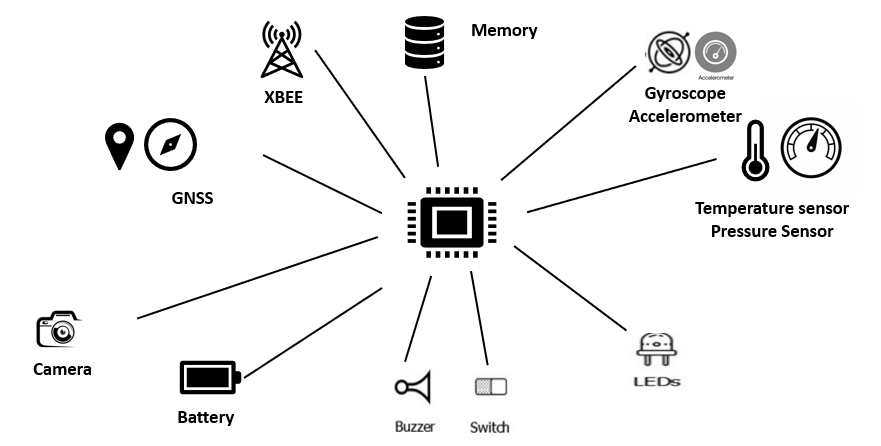


Fig. Communication and Data Handling Overview

Changes made after PDR in communication subsystem and data handling :

# Telemetry

* Upon powering up, the CANSAT collects the required telemetry at a 1 Hz sample rate.
* The telemetry data is being transmitted with ASCII comma separated fields followed by a carriage return in the following format :
* <TEAMID>,<TIMESTAMPING>,<PACKETCOUNT>,<ALTITUDe>,<PRESSURE>,<TEMP>,<VOLTAGE>,<GPSTIME>,<GPS LATITUDE>,<GPS LONGITUDE>, <GPS ALTITUDE>,<GPS SATS>,<ACCELEROMETER DATA>, <GYRO SPIN RATE>, <FLIGHT SOFTWARE STATE>,<ANY OPTIONAL DATA>
* *The received telemetry for the entire mission will be saved on the ground station computer as a comma-separated value (.csv) file with name of file as 2022-ASI-002.csv*

The NET ID/PAN ID of the Xbee is set to team no. as per the requirements and its setting is shown below :

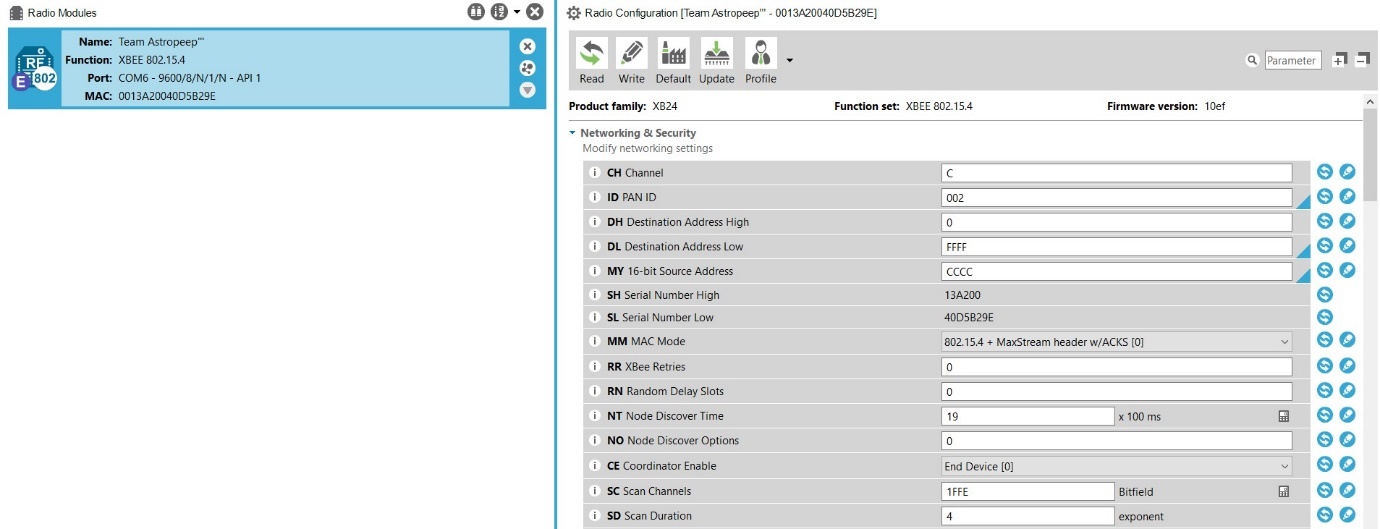


Fig. . Xbee 1 Setting for NET/PAN ID

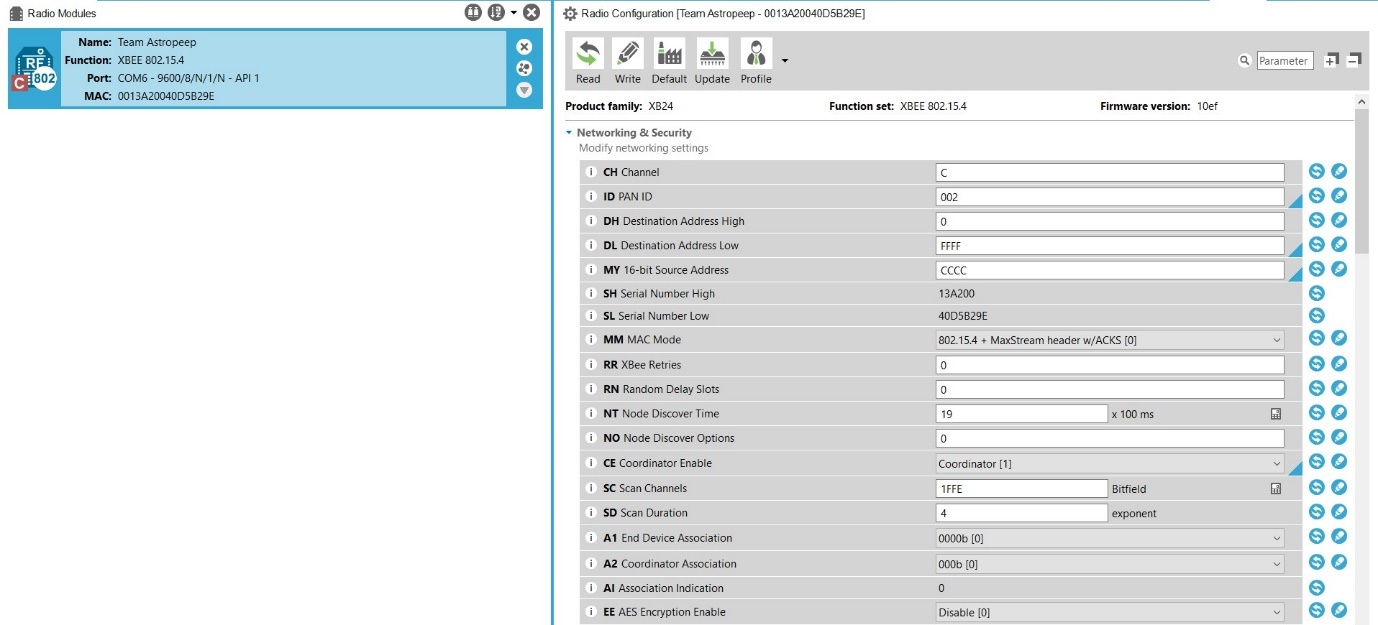


Fig. Xbee 2 Setting for NET/PAN ID

## TELEMETRY DATA FORMAT

|  |  |  |
| --- | --- | --- |
| TM Parameter | Function | Resolution /Format |
| <TEAM ID> | Team Number | 2022ASI-002 |
| <TIME STAMPING> | Time since the initial power | Seconds |
| <PACKET COUNT> | Count of transmitted packets | Integer |
| <ALTITUDE> | Altitude in units of meters and must be relative to ground | 0.1 meters |
| <PRESSURE> | Measurement of atmospheric pressure | 1 pascal |
| <TEMP> | Temperature in Celsius | 0.01 |
| <VOLTAGE> | Voltage of the CANSAT power bus | 0.01 Volts |
| <GPS TIME> | Time generated by the GPS receiver | Seconds |
| <GPS LATITUDE> | Latitude generated by the GPS receiver | 0.0001 degrees |
| <GPS LONGITUDE> | Longitude generated by the GPS receiver | 0.0001 degrees |
| <GPS ALTITUDE> | Altitude generated by the GPS receiver | 0.1 meters |

# Communication and Link Budget

Some changes are done in which we revisited the Gain calculation of receiving and transmitting antenna and we rectified accordingly as shown below.

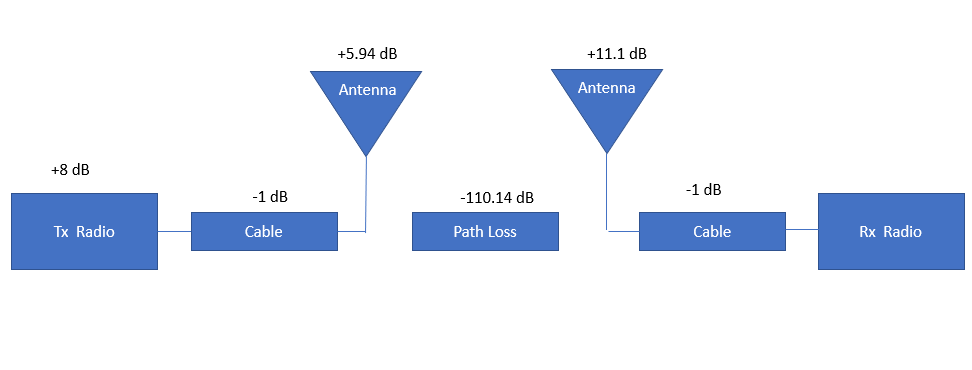


Fig. Communication and Link Budget

## Calculation :

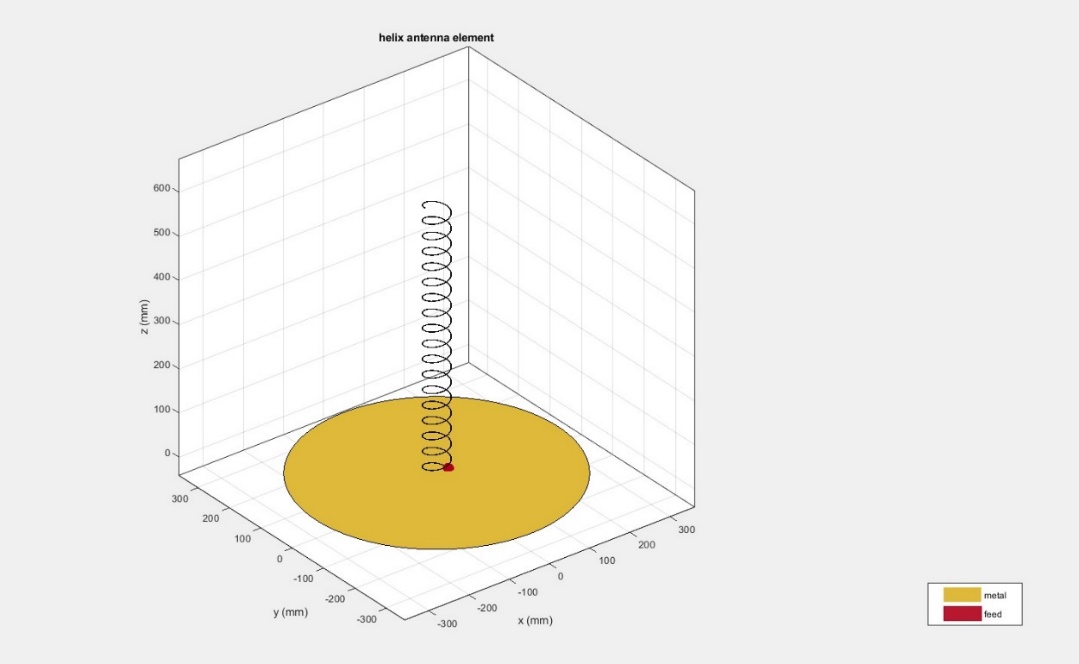
* Transmitted Power : 8 dB
* Transmitter Antenna Gain : 5.94 dB
* Transmitter Loss : 1 dB
* Free Space Loss : 110.14 dB
* Miscellaneous Loss : 1 dB
* Receiver Antenna Gain : 11.1 dB
* Receiver Loss : 1 dB
* Received Power : -84.14 dB
* Receiver Sensitivity : -101 dB
* Link Margin : 16.86 dB

# Antenna Selection

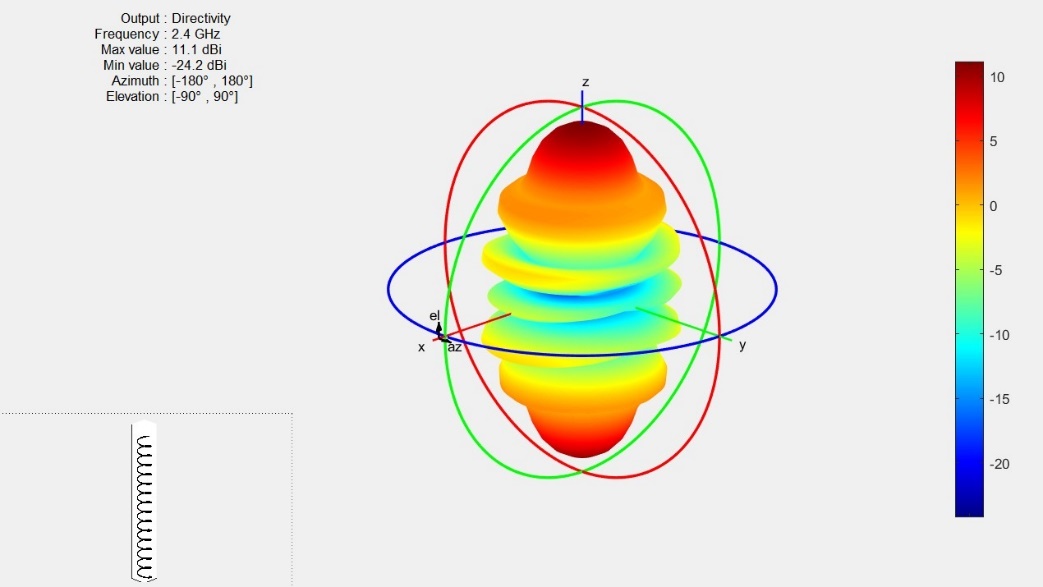
There is no changes since PDR just we simulated the radiation pattern of antennas and found the gains accordingly.

### GCS Antenna

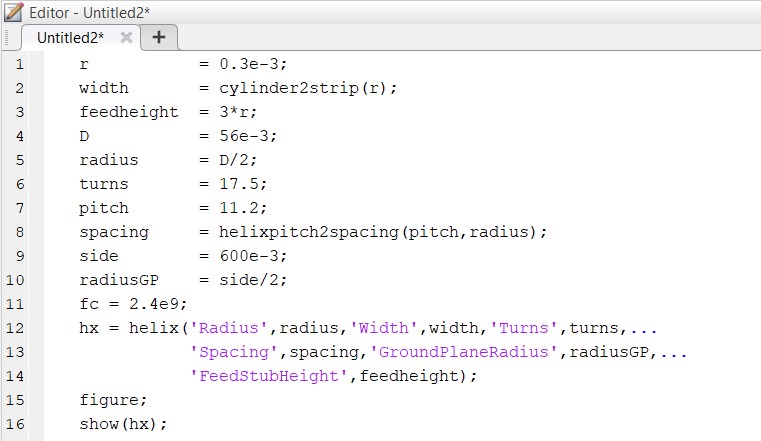
Antenna Design :



Antenna Radiation Pattern :

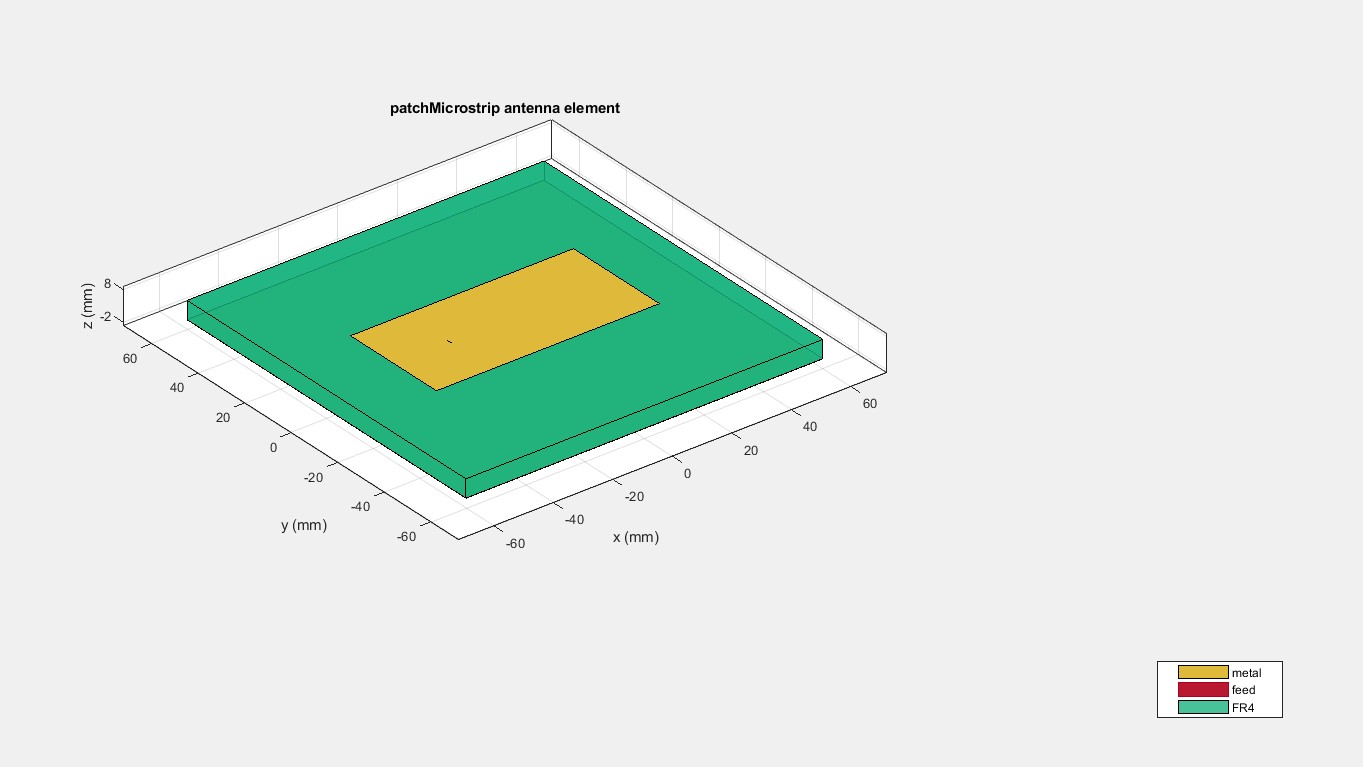


Antenna Design Code :

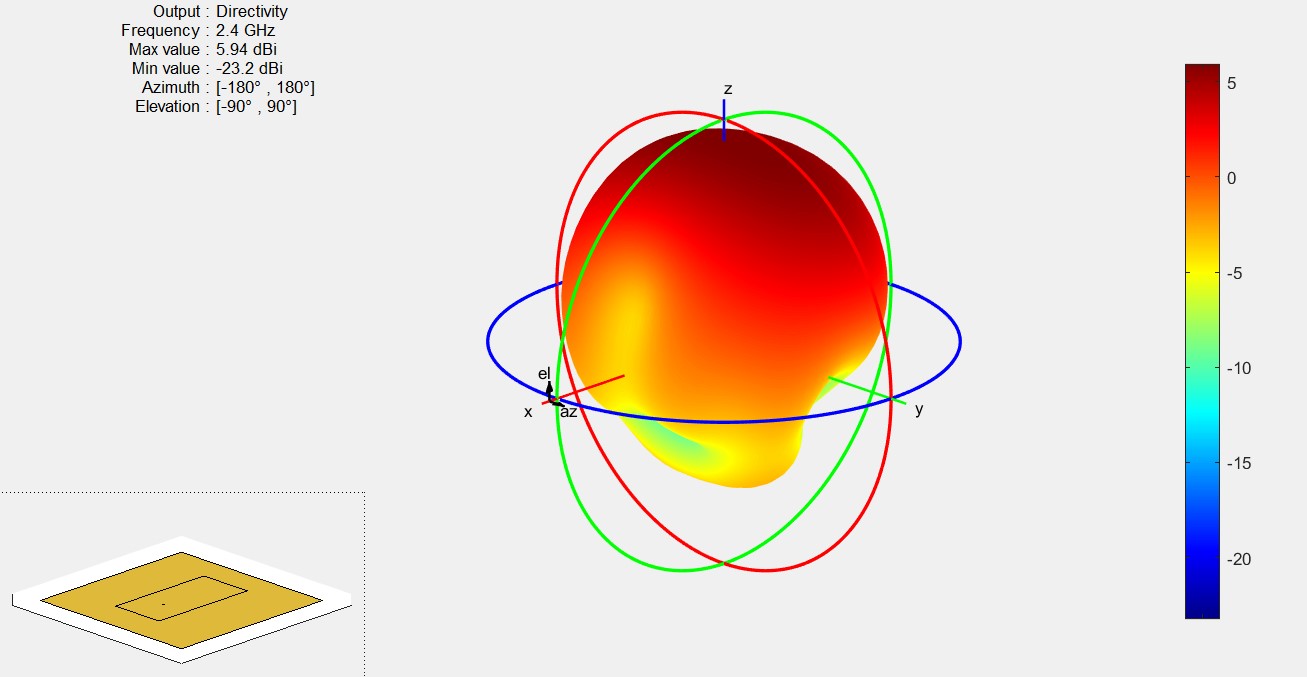


### CANSAT Antenna

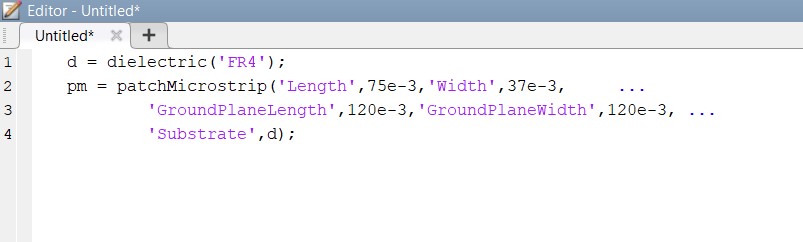
Antenna Design :



Antenna Radiation Pattern :



Antenna Design Code :



The gain obtained after analysis are :

* GCS Antenna – 11.1 db
* Cansat Antenna – 5.94 db

An after calculating we rectified the changes in the Communication and Link Budget.

# Electrical POWER SUBSYSTEM

### There are only some minor changes in Electrical power subsystem as mentioned below :

* The battery configuration is changed from series to parallel which made the step down process more efficient.
* The position of switch is also modified from bottom to panel side of CANSAT.

## Electrical Block Diagram

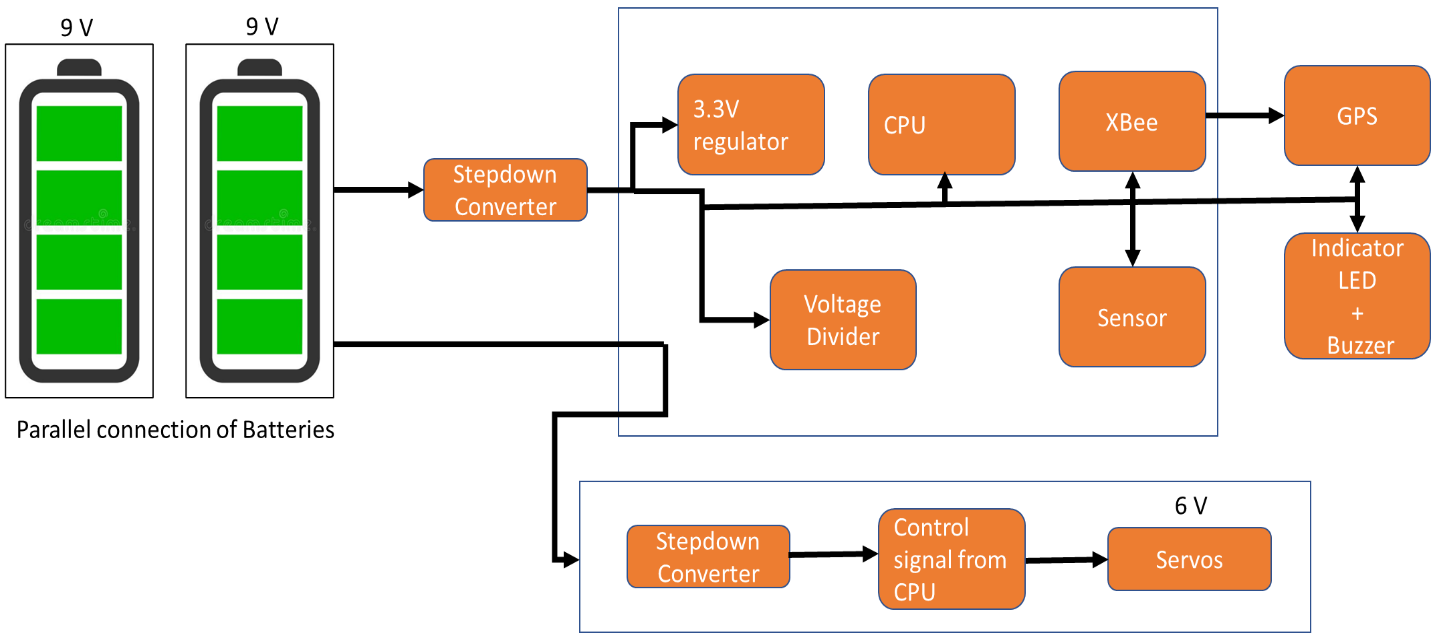


Fig. Electrical Power Subsystem Block Diagram

# 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

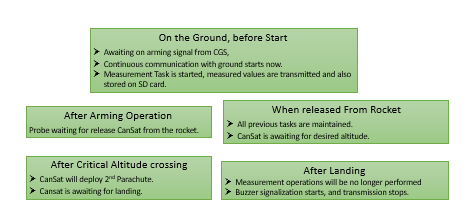


Fig. Flight Software Design Overview

### Momentary Power Cut Precaution Strategy :

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.

### There is no changes since PDR.

### 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.

### Progress Since PDR :

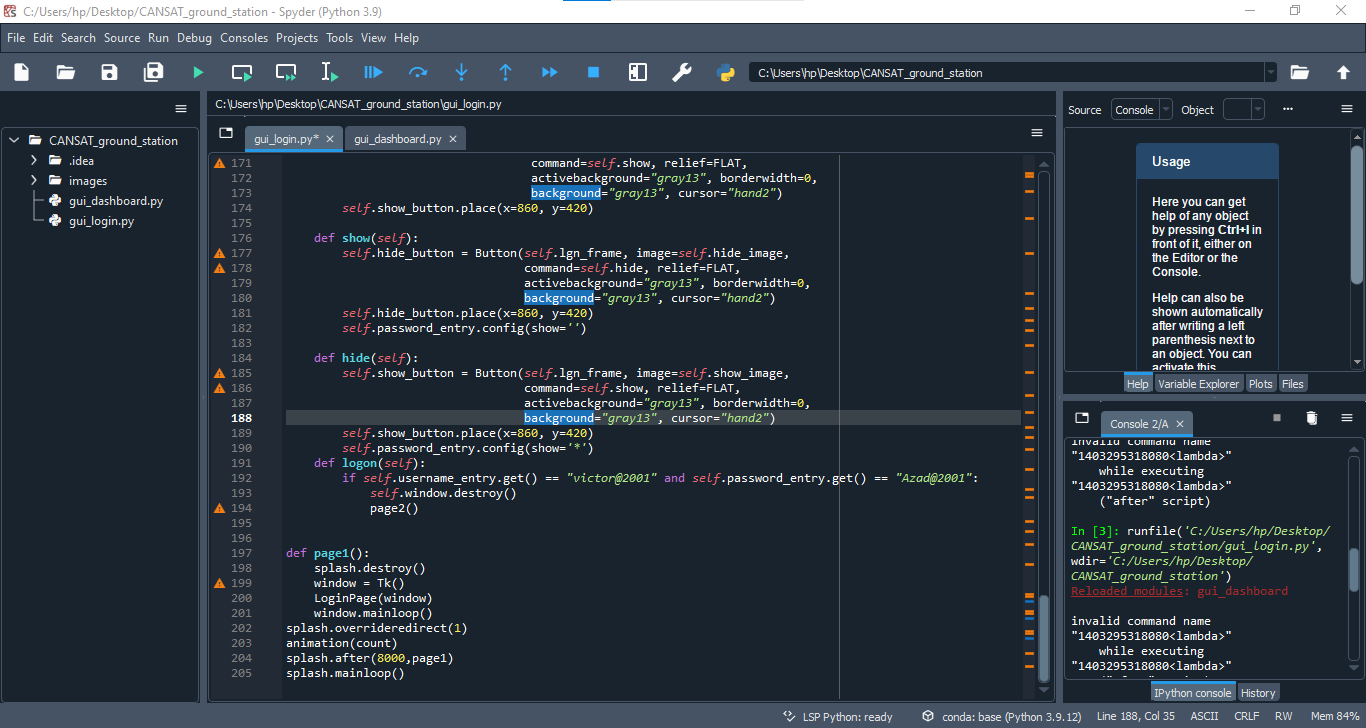
* The login page of GUI is completed and currently working on dashboard page of GUI and the overall data receiving software is tested for sensors individually.
* The development of helical antenna of GCS is started.

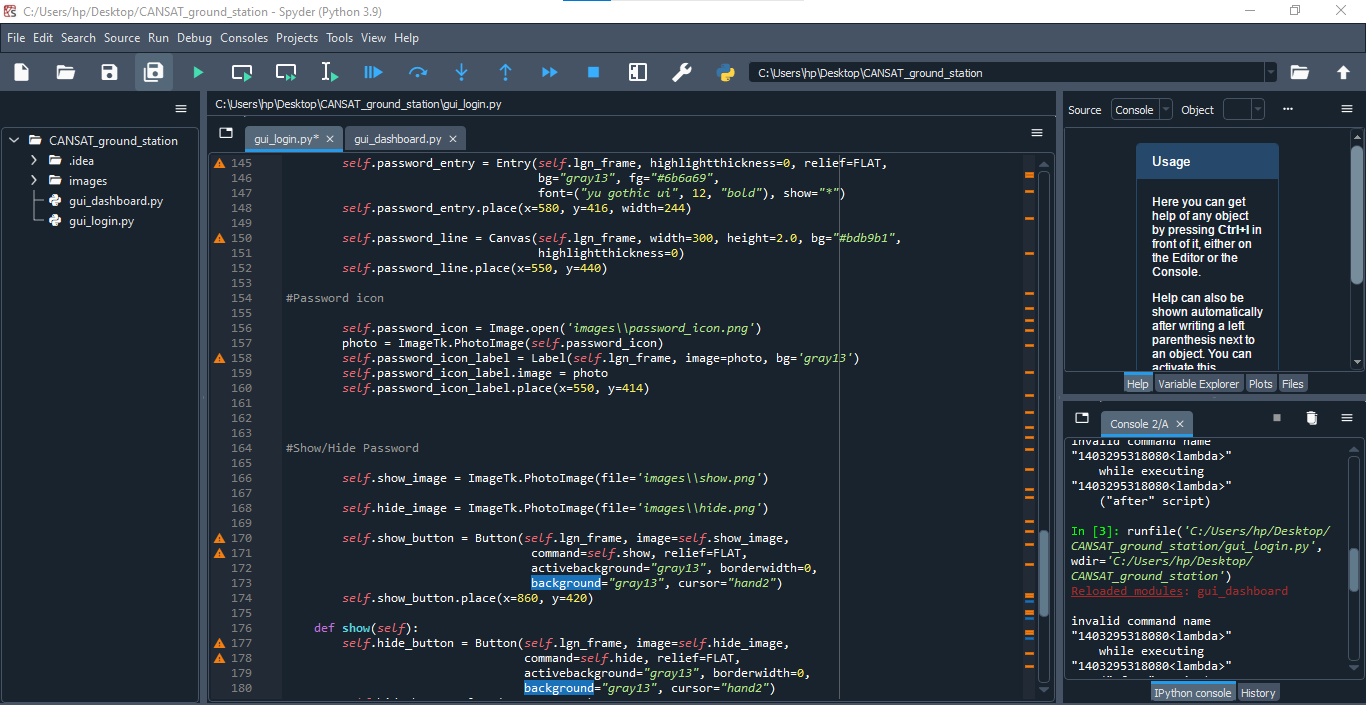
### Ground Control Station GUI :

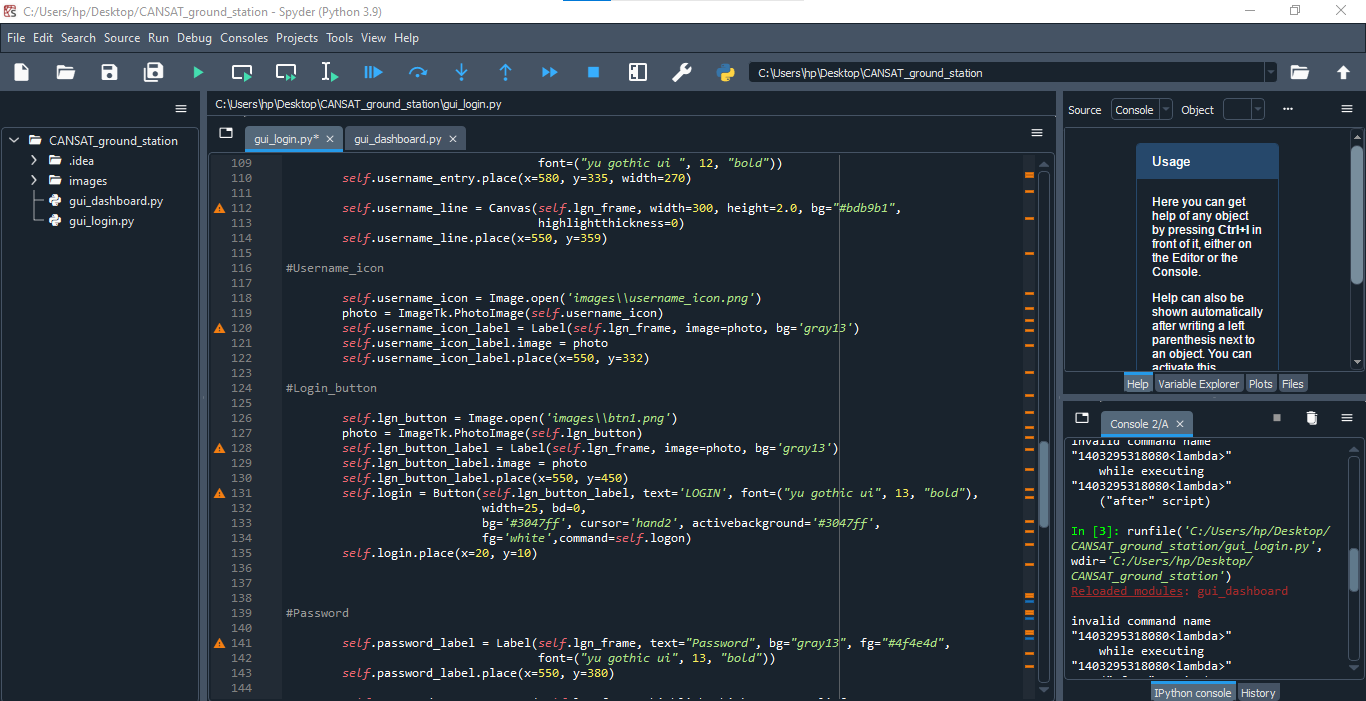
Fig. GCS GUI login page

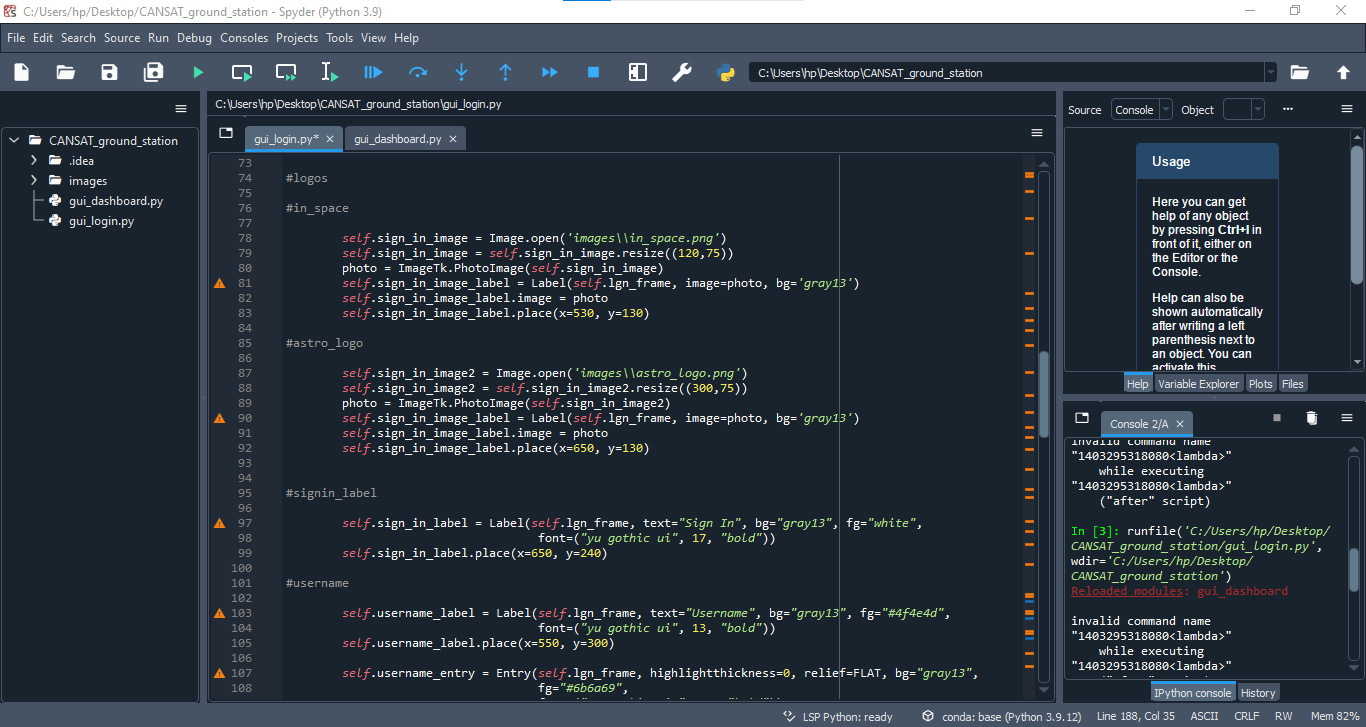
The GUI is build using SPYDER IDE using Python language and the work is still in progress the login page is build as shown above and the dashboard work is in progress and for testing we individually tested the sensor by sending data using our microprocessor and plotted graph in the SPYDER IDE using Python function matplotlib.

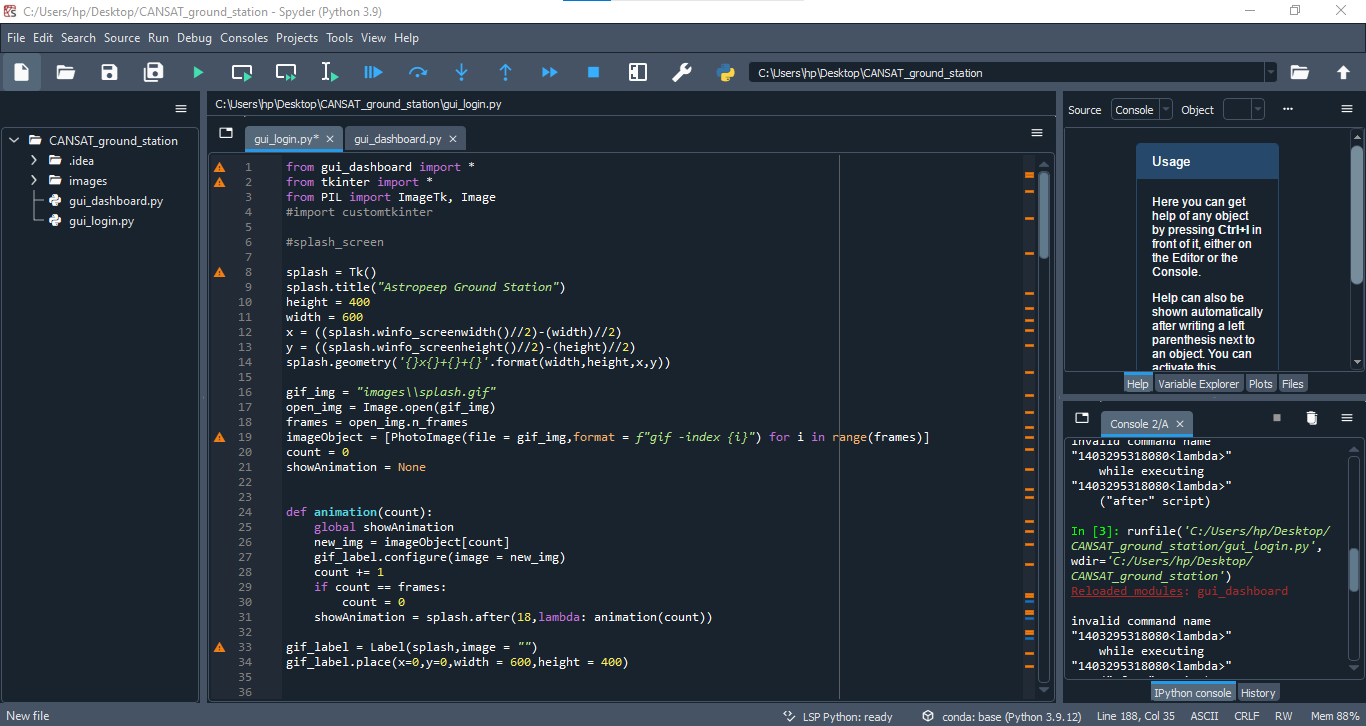
The screenshots of code written in SPYDER IDE are attached below :











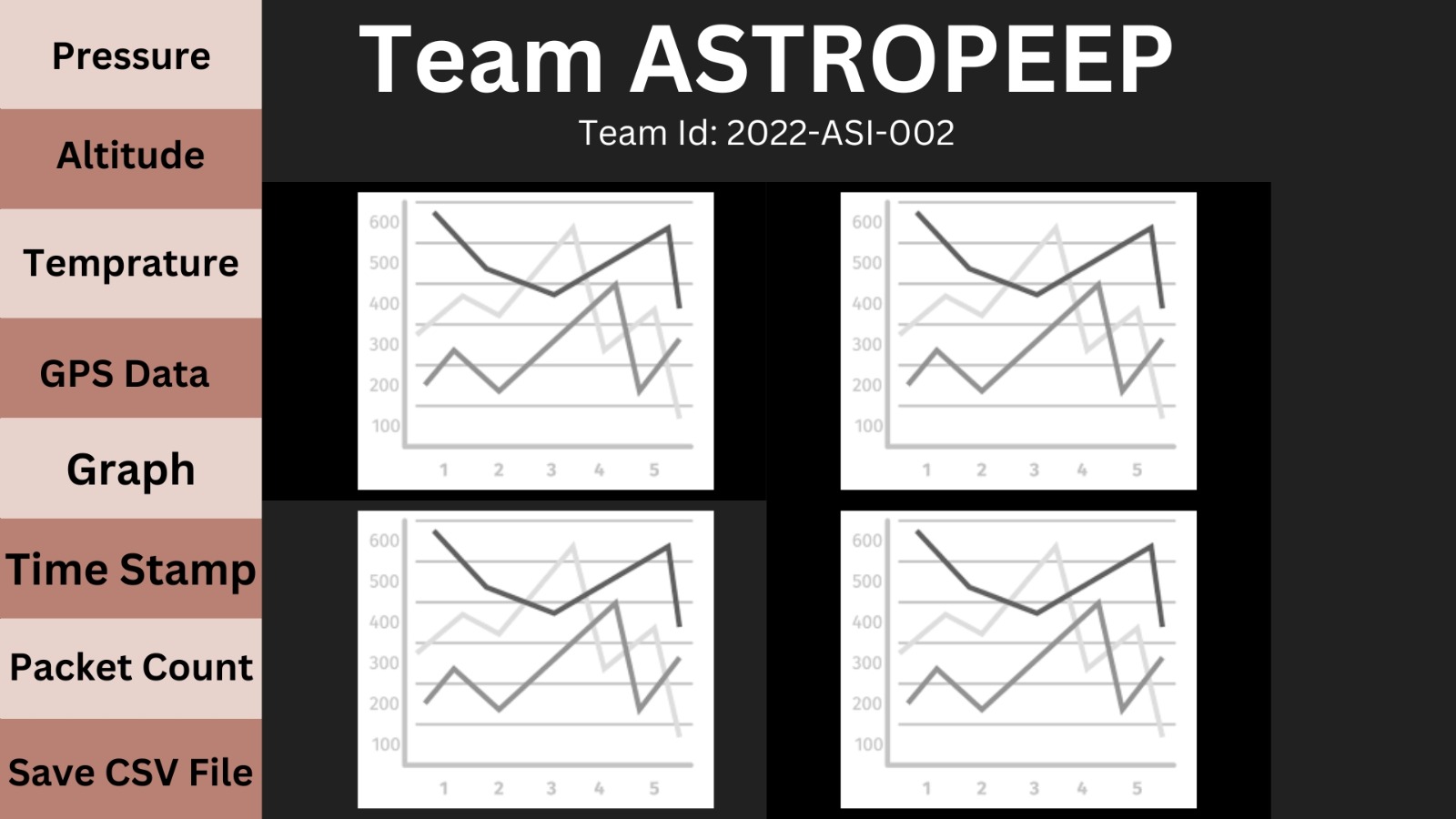
The development of dashboard were all the data will be plotted is in progress and after development the dashboard will be like as shown below :

Fig. GCS GUI Dashboard

# **CANSAT algorithm**

The CANSAT algorithm is developed in C/C++ language using Visual Studio Code and Arduino IDE and the flow chart of the algorithm is shown below :

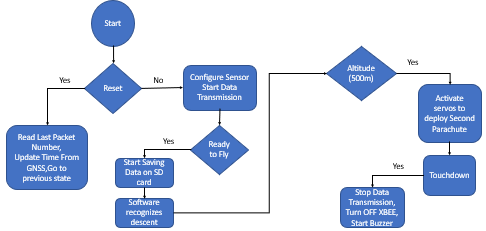


Fig. Cansat Algorithm Flow Chart

# **Ground Station final test results**

The Cansat Ground Station final test results is still in progress as the test of data collection is done individually for all the sensors and in next step we will be combining the overall data into our GUI.

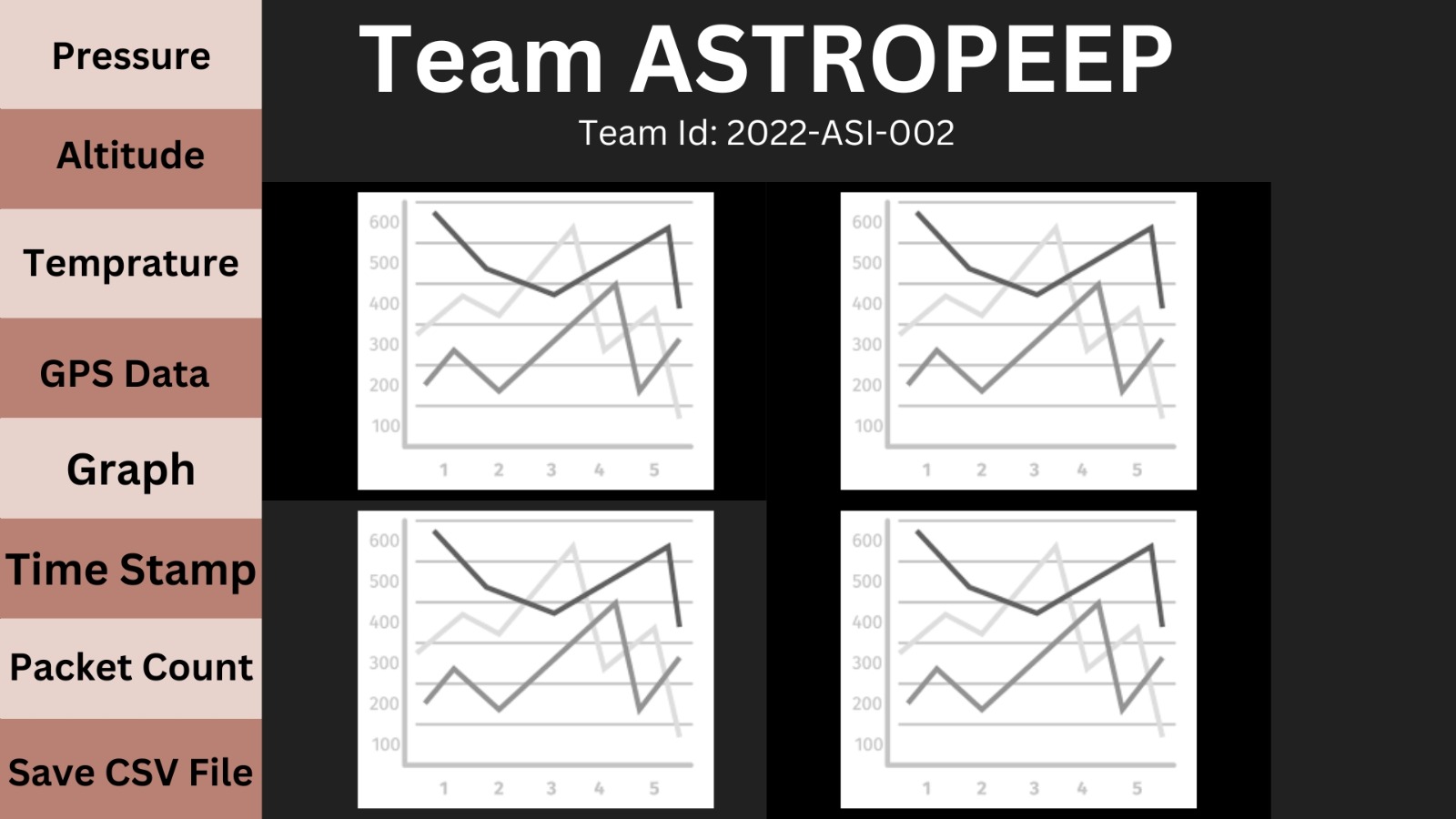
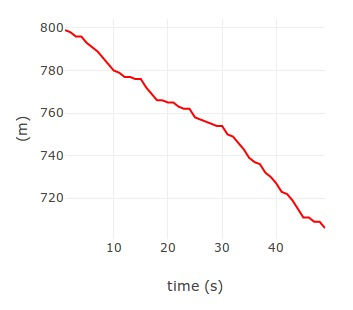
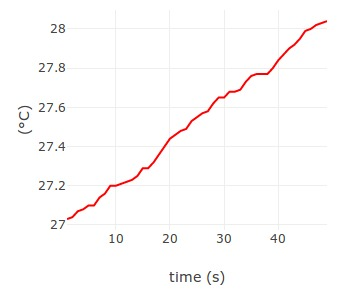
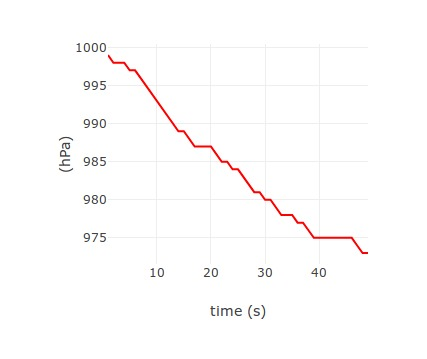
Our GUI work is also still in progress and final glimpse of our GUI is shown below :

Fig. CANSAT GUI Dashboard

The test result plots of the individual sensor data are shown below :

Altitude Test Plot :

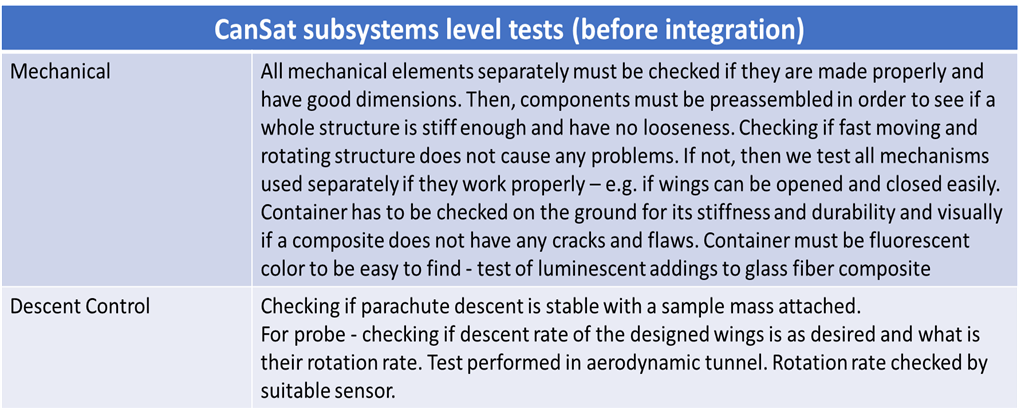
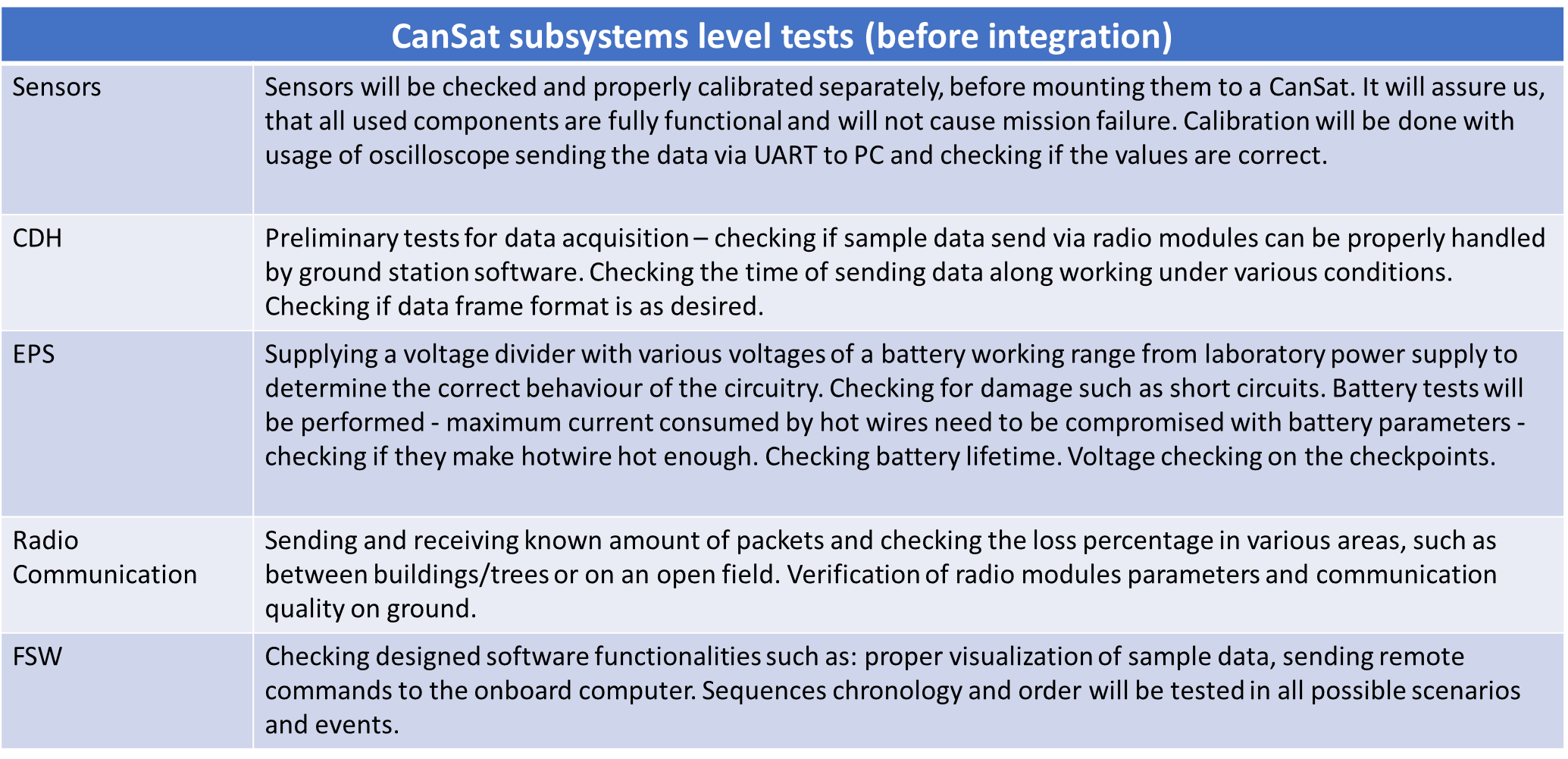
Pressure Test Plot :

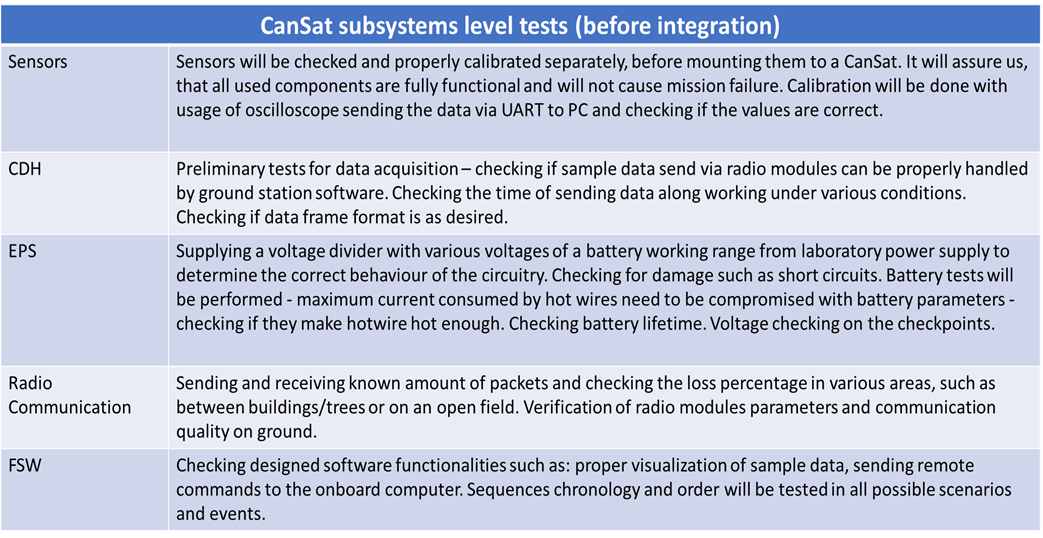
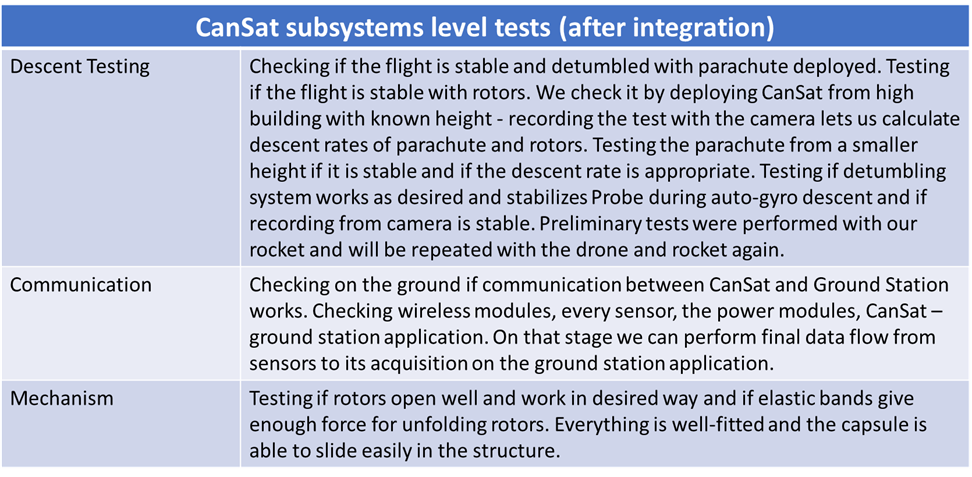
Temperature Test Plot :

The testing of some of the components are still in progress like GPS , Gyroscope and Accelerometer.We are still working on the testing of the above mentioned components.

# **CANSAT Integration and Testing**

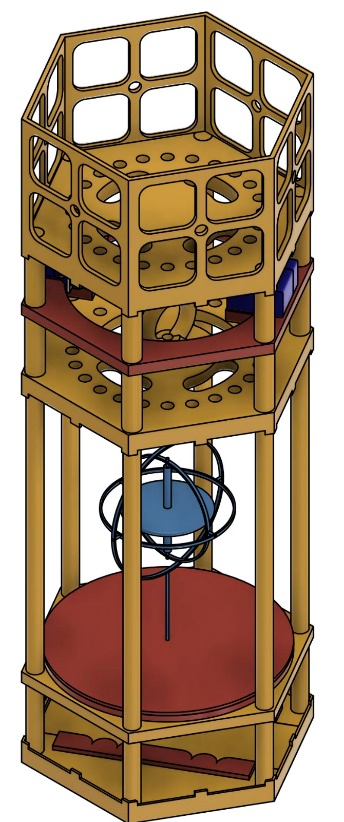
### Testing before Integration :





The above mentioned steps are for testing the cansat before integration and after this testing we will go ahead for integration.

### Integration of CANSAT :

* As our primary structure fabrication process which will let be a homogenous structure with no mechanical joints.
* Other than this the outer body panels will be connected using the shock absorbing screws.
* Batteries , PCB and Servo motors are connected to the particular damper used which are then connected to the structure of CANSAT using screws and other mechanical joints as shown below.

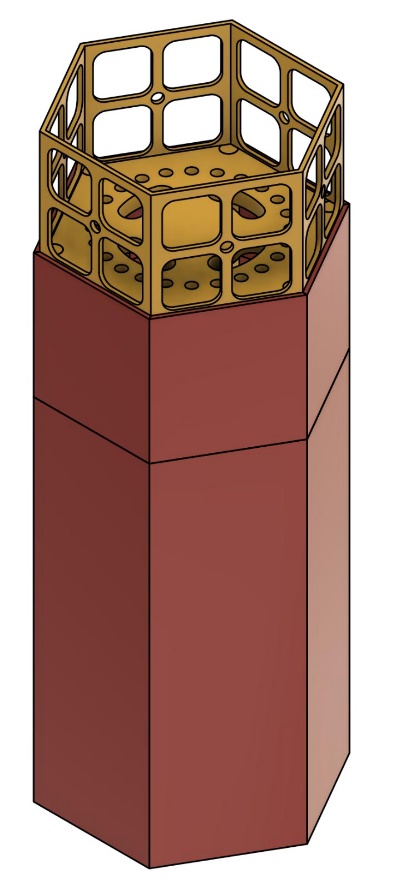
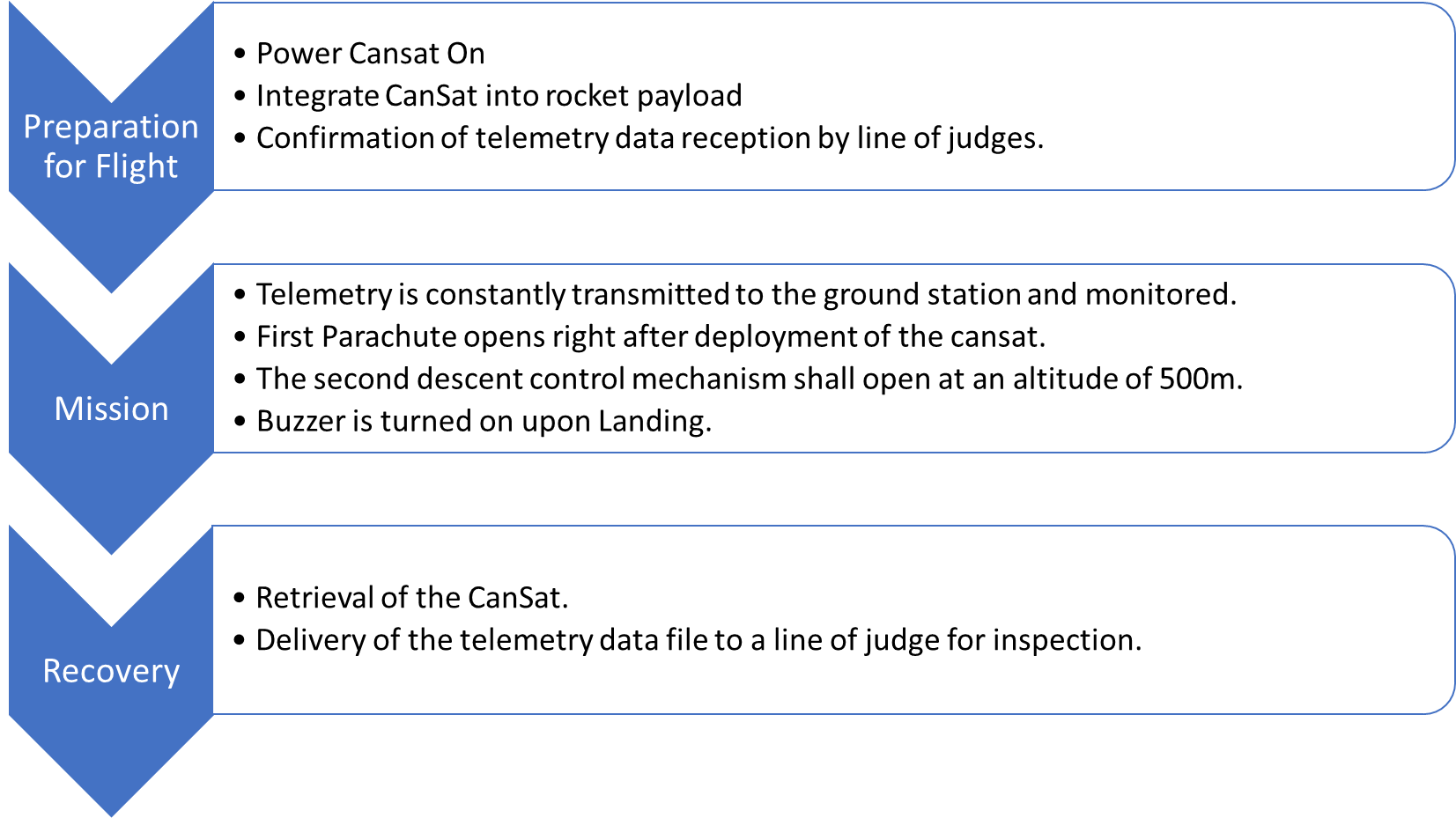
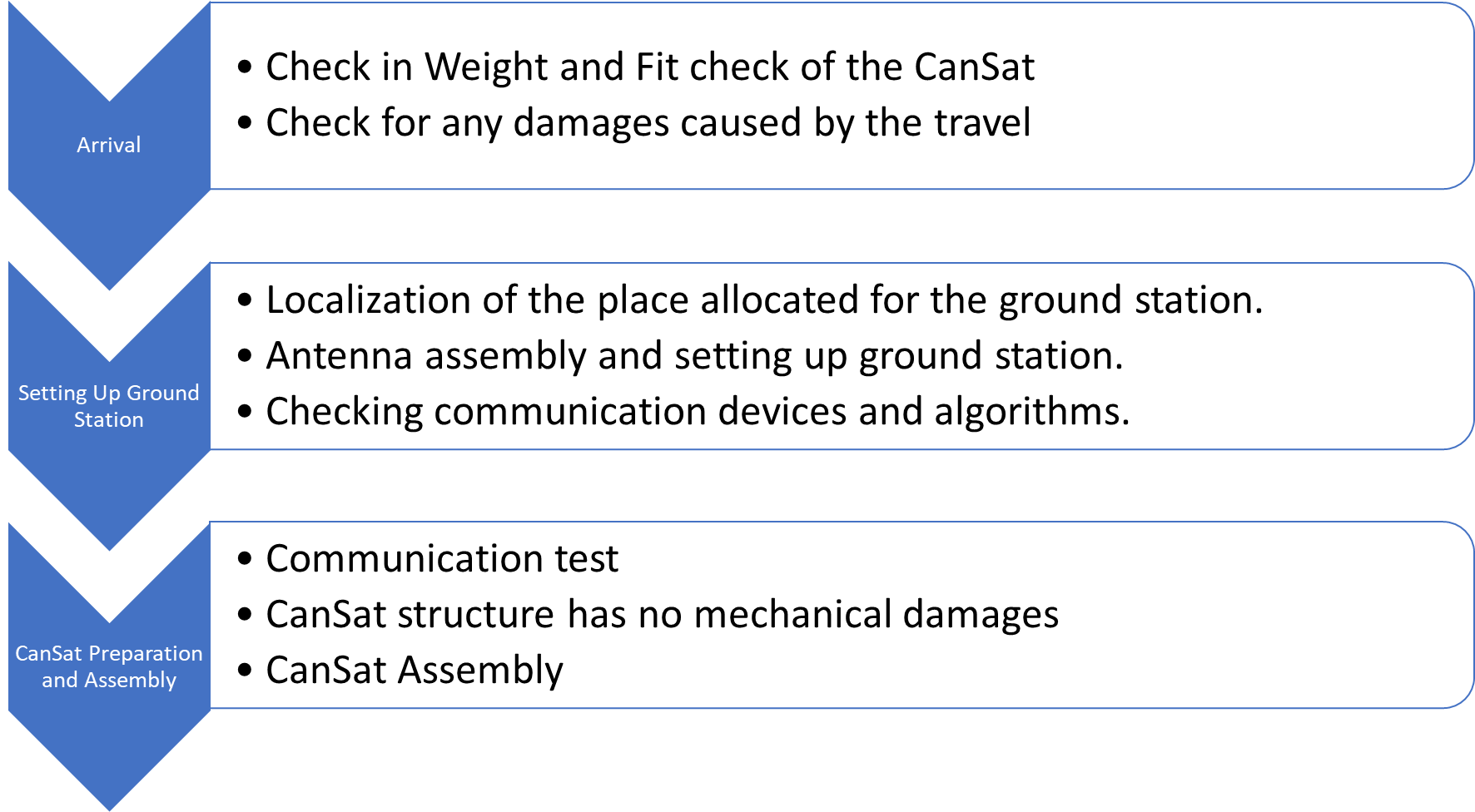


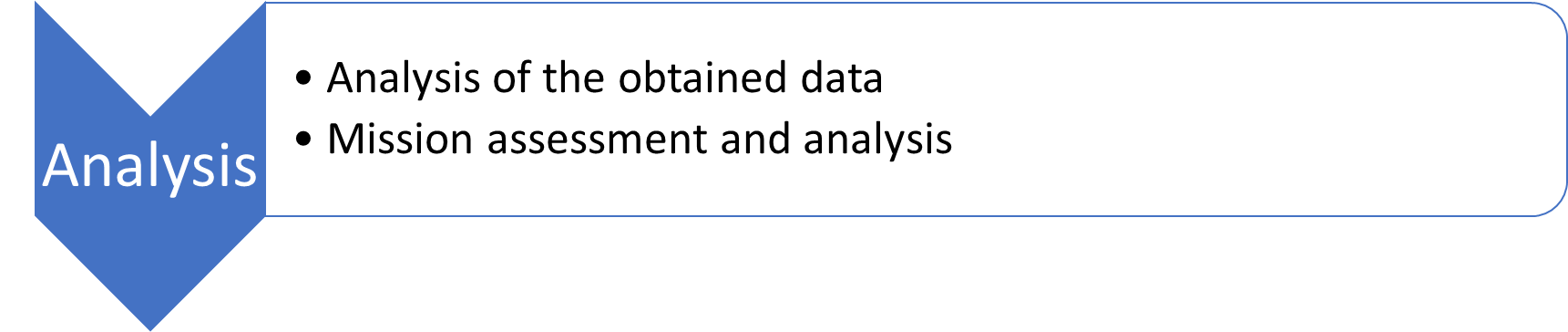
Fig. CANSAT Integration Stage 1

Fig. CANSAT Integration Stage 2

# **Mission Operation and Analysis**







# **Pre-flight requirement-check-analysis**

# Logistics and transportation:

For the Logistics and Transportation:

* We will be packing the fragile and non secured elements independently using cartons and bubble wraps.
* Whereas other all the elements will be packed in a wooden sheet box by using Thermocol sheets, Bubble Wraps, Cardboard and Straps as per the requirements of the fabricated model of CANSAT.

# **CANSAT ready to launch and final comments**

* The testing were made on similar kind of prototypes and we found the rectifications in the model as observed.
* Accordingly we have made changes and as of that our software part is almost done however in Hardware our work is still in progress.
* Due to the delay in the delivery of the materials and the electronic components required for the final fabrication of the CANSAT.
* The work will be completed in accordance to the availability of material however all the analysis , simulations and test results are completed through the alternative scale models of the similar materials.

# Conclusion

### Accomplishments

* Telemetry is tested.
* GCS GUI is developed up to some extent.
* Antenna testing is completed.
* Mechanical analysis (Stress-strain , Vibration and Fit check) is completed.

### Unfinished work

* The fully integrated electronic & mechanical subsystem are not entirely functional yet and needs to be tested.
* Prototypes are being tested and more refinements are required.
* Flight software is in progress.

### Next stage of development

* Final Fabrication work completion
* Code Review
* Testing of integrated system
* Assembly of complete CANSAT
* Integrated testing of CANSAT.