PRATHAM
IIT BOMBAY STUDENT SATELLITE

Critical Design Review
On-Board Computer

By

Pratham Team

Department of Aerospace Engineering,

Indian Institute of Technology, Bombay
July 2015

Contents

*

List of Tables

*

List of Figures

*

Chapter 1

Introduction

1.1 Pratham Ovrview

Pratham is the first satellite designed and built by the IIT Bombay Student Satellite Team. It's defining features are as follows:

- Weight: under 9.7 Kg (without FE ring)
- Size: 28.9 cm $\times$ 29.9 cm $\times$ 46.1 cm
- Solar Panels on 4 sides
- Orbit: 10:30 polar sun-synchronous
- Three pre-deployed monopoles
- Downlink at frequency 437.455 MHz
- Beacon at frequency 145.98 MHz
- Uplink at frequency 437.455 MHz
- Completely autonomous

1.1.1 Mission Objectives

- The main aim of Pratham is to measure the Total Electron Count (TEC) over India.
- It is also recognized that the learning and experience gained from this endeavour is quite valuable by itself regardless of the data obtained from the TEC measurements.
- The mission statement has, therefore, been broken down into of various levels. The successful completion of each level adds to the overall success of the mission.

1.1.2 On-board Computer (OBC) Overview

- The OBC Subsystem comprises of almost all the computing power employed by Pratham.
- The mission statement of Pratham requires the satellite to stay in a stable orbit for a reasonably long period of time and transmit data to the ground-station in order to facilitate measurement of the Total Electron Count (TEC) in the atmosphere.

- The design of Pratham has the various sub-systems like the Communication sub-system and Controls sub-system providing the equipment for communication and controls respectively while the OBC subsystem interfaces with these elements and executes the algorithms to keep the satellite stable and transmit data to earth.

The System Engineer has decided on an operating sequence for the various functions of the satellite in orbit. This sequence orders tasks on the basis of orbital time/position and also orders the execution of select tasks during the response to certain contingencies. The design of the On-board Computer is based on achieving this operating sequence with the maximum efficiency and minimum risk.

1.2 Design Methodology

The design of the On-board Computer sub-system has two aspects – the design of the hardware that forms the physical basis, and the accompanying software.

1.2.1 Hardware Design

The hardware design has been motivated by the requirement to interface with the components provided by the other sub-systems on a reliable basis. The On-board Computer must interface with the Power sub-system to receive information about the health of the various components of the satellite. It must interface with the Attitude determination and Control sub-system to stabilize the satellite, and it must interface with the Communication sub-system to transmit data down to the ground-station. These design and implementation of these interfaces make up the major portion of hardware design. The simplest satisfactory solution, in terms of components and layout has been chosen as this is the first time such a project has been attempted at IITB. Thus, along with reliability, the feasibility of the implementation is a major concern. The satellite's operational sequence translates to a more detailed operational sequence for the On-board Computer, where satellite tasks such as downlink, for example, correspond to a sequence of storage and retrieval from memory, interfacing with the antenna and On-board Computer Power Communication Attitude Determination and Control transmission of suitably encoded data.

1.2.2 Software Design

The software of the On-board Computer is concerned with the implementation of the desired operational sequence. Due to the nature of the system – a satellite in a regular, periodic orbit, the nominal operation of the satellite involves the cyclic execution of a series of predefined tasks in a predetermined order. This allows us to design the software without the need for an operating system or a complicated scheduler. This should greatly simplify the implementation and testing of the On-board Computer. The nature of the system – a satellite far removed from manual control, also imposes certain limitations of the contingencies that can be handled and the methods by which the satellite can attempt to return to nominal operation. The failure of solar panels, for example, or of the actuators is an exigency that cannot be recovered from. On the other hand, the satellite can still function, to a degree, after the failure of the Global Position System (GPS) unit. The system engineer has identified the nature of the failures that can be handled and their appropriate responses. The implementation of these responses is part of the software design.

1.3 Fabrication and Testing

The idea behind a cube-sat is a satellite built by universities using commercially available off-the-shelf components. In keeping with this principle, the components used as part of the On-board Computer will be those that have been used regularly by students- no special space grade components will be used. The procedures for fabrication, on the other hand, will be those that are prescribed by ISRO and the system – including the printed circuit board and the associated soldering of components will be done by space/industry grade manufacturers and will conform to ISRO approved standards. The testing of the On-board Computer comprises of multiple levels. The software and hardware comprising of each interface will be tested individually with the associated components from the other sub-systems. This will be followed by the testing of the entire information pathway which links the power, On-board computer and communication sub- systems. This will be followed by On-board Computer In-Loop Simulations (OILS) which test the complete functioning of the On-board Computer with its implementation of the Control algorithms, Communication tasks and Health Monitoring duties by simulating the actual conditions of space as experienced by the various sensors. These simulations will locate design flaws, if any, and also provide estimates on the reliability of the system as a whole.

Chapter 2

Sub-System Requirements

The various tasks that the On-board Computer sub-system is supposed to perform on the flight model are listed in this section. The efficient and reliable execution of these tasks is the primary motivation for the design of the On-board Computer sub-system.

2.1 Requirements from other sub-systems on OBC

2.1.1 System requirements

The On-board Computer sub-system is required to conduct pre-flight checks to confirm the operational status of select important components like the GPS unit. These checks will be conducted before the satellite is integrated with the launch vehicle. The health status of a number of components will be monitored and stored while the satellite is in orbit. This "Health Monitoring Data" (or HM data) will consist of data regarding various loads as seen by the Power sub-system along with additional temperature data of battery, Low-Noise Amplifier (LNA) and Power Amplifier (PA) will also be stored. Position data will also be sent for payload purpose.

2.1.2 Attitude Determination and Control

The Controls subsystem is primarily concerned with determining the satellite's position and maintaining a stable orbit. The On-board Computer is required to interface with sensors and actuators and execute the control-law calculations that determine the required actuation from the current position.

2.1.3 Sensor Interfacing

The On-board Computer is required to interface with the following sensors:

1. Global Positioning System (GPS) Unit
2. Magnetometer
3. Sun-Sensors

2.1.4 Control Law Execution

The On-board Computer shall execute the control law as designed by the Controls sub-system. This includes performing the requisite numerical calculations with the desired accuracy as well as according to the predefined sequence and timings.

2.1.5 Actuation

The On-board Computer will interface with and actuate the magnetorquers according to the control law. This implies provision of suitable pulse-width-modulated signals for suitable time intervals.

2.1.6 Power

The power subsystem is concerned with acquiring, regulating and distributing power to the various components of the satellite. The Power and On-board Computer subsystem must interact to exchange data about satellite health (HM data).

Response to low-power situations

The Power subsystem will generate a signal indicating a predicted inability to supply adequate power to the On-board Computer subsystem. The data exchanged regularly between the Power and On-board Computer subsystem will indicate any misbehaviour (over-current) on the part of any component. The Power board itself will respond to situations where the OBC malfunctions or low power situations which might arise, and take suitable measures to shut down various components. In all other situations of malfunction the OBC directs the Power board to shutdown components in a defined sequence.

2.1.7 Communication and Ground-Station

For the purposes of monitoring the status of the mission, the satellite will transmit the health of the various components during downlink. In this regard, the OBC sub-system is required to send the health, temperature, latitude, longitude and altitude data in packets encoded using the AX-25 communication protocol, at a maximum rate of 1.2 kbps whenever the satellite is over the Ground- Station.

2.1.8 Integration

To ease the integration of the On-board Computer sub-system with the rest of the satellite, the final Printed Circuit Board (PCB) of the On-board Computer must be 12 cm x 12 cm in dimensions, with a natural frequency above 90 Hz. The final PCB must confirm to the requirements of the harness to be designed by the integration team

2.1.9 Quality

The Quality sub-system requires the On-board Computer hardware to be tested extensively to reduce chances of infant mortality. While the exact testing procedures have not yet been fixed, they must provide an estimate of the probability of failure. For the satellite to have a life of 4 months, the estimate of the probability of failure should be below some threshold. The quality of the software plays a big role in the success of the satellite as a whole. The quality will be ensured by keeping proper coding conventions in mind including naming conventions and extensive in-line comments. The software will be tested by a code walk through to ensure that the software performs the requisite functions in the correct manner. A formal verification of the software seems unnecessary due to the deterministic nature of the system. Finally, the complete electrical subsystem including the OBC will be tested using OILS (On board computer In-Loop Simulations) to ensure that the probability of failure is low.

2.2 Requirements from OBC on other subsystems

In order to execute the above functions, the On-board Computer sub-system places several constraints on other sub-systems:

2.2.1 Power

The Power sub-system is required to supply continuous power of up to 1 Watt at all times. The Power sub-system will also respond to the On-board Computer's requests to turn ON/OFF certain components depending on their need. This information will be communicated to the Power sub-system in the form of one byte every cycle (2 seconds). The Power sub-system will also send HM data regarding the status of each of the major loads on the satellite to the On-board Computer when polled for this data. Also the OBC also tells the power board to turn off the loads in case of over-current situations and also decides when to turn them ON. In case of over-current in the OBC the power board decides to switch off the OBC.

2.2.2 Thermals

The Thermals sub-system will be required to remove excess heat from the On-board Computer PCB, in order to maintain the temperatures of all the components between -40^0C and 85^0C (the operating range of the components).

Chapter 3

Software Design

3.1 Requirements

As part of the pre PDR phase, the Onboard Computer subsystem had completed a major portion of the software design. Software design comprises of formulating solutions to the following issues:

- To design algorithms that would perform the various tasks listed as part of the subsystem requirements. These algorithms must be robust and efficient.
- To write code to drive the various peripherals of the microcontrollers. These codes would form part of a primitive hardware abstraction layer and would provide methods that other developers could use to drive the peripherals without going into the details of microcontroller programming.

3.2 Design process

The software design consisted of 3 parts:

- Writing modules for hardware interfacing
- Writing modules for processing the sensor information obtained above, followed by actuation of magnetorquers
- Integrating the parts above to form the complete system

3.3 Coding Practices

This being a big project, special attention has been given to coding practices so as to make the code understandable and easy to debug. The following practices have been followed:

3.3.1 Naming Convention

Proper naming conventions have been used while writing the software:

- Peripheral related functions: `<instruction>_<peripheral>[_<mode>]` Eg. `read_GPS()`, `init_SPI_trans(<arg>)`
- Peripheral related structs: `<peripheral>_<attribute>` Eg. `MM_reading`
- Names of variables that have significance in the system are descriptive of what they represent Eg. `HM_counter`, command etc.
- Global variables: Start with a capital letter Eg. Transmission, Mode etc.

3.3.2 Comments

The code has been extensively commented and documentation has been produced from the in-line comments using Atmel Studio 6.

3.3.3 Modularity

Code modularity has been ensured to ensure easy testing of separate components and easy integration as well as debugging.

3.4 Hardware Control

In order to execute various tasks, it is necessary to use a number of the ATmega128's peripherals. One part of software design is to write the C code required to drive these peripherals. This effort was further sub-divided into two portions:

3.4.1 Sensor interfacing

The On-board computer interfaces with the GPS unit, the magnetometer, and the sun-sensors. The GPS and Magnetometer are both compatible with the RS232 communication standard. As the ATmega 128 has 2 fully independent UART peripherals, both the magnetometer and the GPS have a dedicated UART peripheral interfaced with them.The sun-sensors uses the analog-to-digital convertor (ADC) peripheral of an atmega 8 which communicates with On-board computer using SPI.

- In order to interface with these components, the UART (Universal Asynchronous Receiver Transmitter) peripheral needed to be configured. This peripheral allows for complete duplex reception and transmission. The configuration of this peripheral for asynchronous use involves the specification of the required baud rate, and the frame format (number of stop bits and the presence/absence of parity bits). Transmission is initiated by simply writing the byte to be transferred to a specific hardware register. The end of transmission or reception is indicated by the corresponding flag in a dedicated status register. The setting of these flags can also be used to trigger appropriate interrupts.
    - The magnetometer will be operated in polling mode. Every time a reading is required, the primary micro-controller will send a 5 byte command to the magnetometer via the UART peripheral. The micro-controller will then enable receive-complete interrupts and wait for a response (wait for the UART receive complete flag to be set and the interrupt to be triggered). The magnetometer will send a 7 byte string of the magnetic field values (2 bytes each for x, y and z directions and one 'end of message' byte, i.e., carriage return (0x0D)). As soon as a byte is received, the setting of the receive-complete flag will trigger an interrupt. The received byte will be buffered in each interrupt. Once 7 bytes

are received, they will be processed. Note that interrupts do not process the received data but only buffer it for processing in the main code.

  - GPS will be run in interrupt mode. It takes around 85 seconds for the first valid reading to arrive. This validity is based on the PDOP value being sent by the GPS and the time has been determined experimentally and discussed in a later chapter. During this period, the receive-complete interrupt of that UART peripheral will be enabled. The GPS unit will send one packet of GPS data every second. Each data packet consists of 8 messages conveying position, velocity, time-date and other information. Each message consists of a number of bytes. All the received bytes will be buffered in the receive-complete interrupt for future processing. The buffered message will be processed for a specific messages like position and velocity using the start and end flags and the relevant information will be used for further processing. The GPS UART interface will be operated at 9600 baud.

- The sun-sensors are similar in operation to photo-diodes. They provide an analog current as output. These currents are converted to analog voltages using an op-amp. The analog voltages are then routed to the Analog-to-Digital convertor peripheral of an atmega8 chip which converts it to digital voltage. The micro-controller (ATmega 128) will use the SPI interface to obtain the data from the atmega 8 chip.

3.4.2 Data Communication

The Onboard Computer design involves the exchange of data between the micro-controllers and several other devices on board the satellite. These data transfers are handled by several on-chip peripherals and codes must be written to drive these peripherals. The data pathways on board the satellite are:

- Power $\Leftrightarrow$ On-board computer
  Health monitoring data and load on-off decision (I^2C)

- Primary Microcontroller $\Leftrightarrow$ Secondary Microcontroller
  Health monitoring and attitude data, commands for downlink (SPI)

- Secondary Microcontroller $\rightarrow$ Downlink
  AX.25 packeted data (UART)

- Secondary Microcontroller $\Leftrightarrow$ EEPROM
  Data storage before downlink (I2C)

- Primary microcontroller $\Leftrightarrow$ ADC
  Sun-sensor Readings (SPI)

3.4.3 Task Execution

The other portion of the software comprises of the code written to execute the various complex tasks assigned to the on-board computer. This again can be subdivided into two parts. The major tasks (like control law execution and downlink) are independent of each other and can be coded as separate blocks (each of these tasks can themselves be broken down into smaller sub-blocks – for example: the control law task includes parsing of the GPS data as a sub- task). The coding of these sub-blocks formed one part of writing codes for task execution. The second part comprises of deciding on a certain execution sequence and putting together the various blocks/sub-blocks according to this operational sequence. The coding of several of these sub-blocks and their integration will be covered as part of the CDR. The list of tasks/sub-blocks that are part of the software design is listed below along with their frequency of execution.

List of Tasks

1. Control Law Implementation
2. Communication Routines
3. Health Monitoring

Control Law Implementation

Data Acquisition
Read from GPS

1. Read GPS – receive data via UART, store sentence in array (This buffering takes place in the UART receive-complete interrupts). Since the GPS has a warm start-up time of around 70 seconds, the GPS must be turned on about 70 seconds before readings are to be taken. (Once every 10 minutes)
2. Parse array and store Latitude, Longitude, Altitude and Time in separate arrays/-variables. (Once every 10 minutes)

Read from Magnetometer

1. Read Magnetometer – receive data via UART, directly receive Bx, By, Bz readings in binary. Store in array/variables (This buffering takes place in the UART receive-complete interrupts). (Polled for readings once every 2 seconds)
2. Convert the stored binary values to decimal values for use in the control law. (Once every 2 seconds)

Read from Sun-Sensor

1. Using the ADC peripheral of the atmega8 on ADC board the analog voltage values of the 6 sun-sensors are converted to digital voltages and stored in an array.(Once every 2 seconds)

Data Processing

Data processing involves the following steps:

1. Use GPS/Orbit propagator and sun-sensor readings along with solar model to get the sun vector.

2. Use the sun-vector along with the magnetic field readings to estimated the attitude quaternions using a quaternion estimation (QuEST) algorithm. The GPS will be used only once every 10 minutes – to correct the orbit estimator. At other times, the orbit estimator itself will be used to determine the satellite's position.

3. Calculate the moment of the satellite.

4. Use the moment to determine which control law to use (De-tumbling / Nominal)

Actuation

The 2 second time frame for actuation comprises of:

1. While reading magnetometer, no PWM (first 1 ms of frame) is executed to remove interference due to magnetic field of torquers.

2. For rest of cycle previously calculated values of current are provided to the torquers using PWM.

3. PWM duty cycles for next cycle are updated in the micro-controller registers after control law is executed. Attitude quaternions/moment are used to calculate necessary PWM signals to magnetorquers using control law.

NOTE: Entire sequence of Data Processing and Actuation steps is done once every 2 seconds. All these routines occur on the primary micro-controller. Thus 2 seconds is our primary frame time, while the GPS read routine occurs once every 10 minutes. Downlink is the only asynchronous event in the system.

3.4.4 Communication Routines

Check Position

Use GPS readings/Orbit Propagator to determine if we are over India or over ground-station or neither; and act accordingly (executed every 2 second cycle):

1. If we are over Ground Station[1] or over India/France, switch on second monopole and switch on CC1101 chip (via downlink board) else put CC1101 to power-down mode.

2. If we are over ground-station – transmit Health montoring and Position (Latitude, Longitude and Altitude) data.

3. If we are over India but not over ground station – transmit Position data.

[1]smaller transmission circle over Mumbai

Transmission

1. Primary micro-controller detects position and sends commands to the secondary micro-controller and the power micro-controller. The Power micro-controller switches on the downlink micro-controller.

2. The downlink micro-controller configures the CC chip.

3. Once successful configuration of the CC1101 chip is done, the secondary micro-controller takes data from the EEPROM flash memory and primary micro-controller then packages it into AX.25 format and sends it to the downlink microcontroller, which sends it over to the CC1101 chip.

4. Data transmission happens continuously while the satellite is over India/ground-station.

3.4.5 Health Monitoring

Data Acquisition

1. Primary microcontroller requests and receives (stores in temporary buffer) load data packet from Power microcontroller (via I2C). (Once every 2 seconds).

2. Primary microcontroller creates a data packet containing the HM data. (Once every 2 seconds)

Data Analysis

1. Loads currently switched on are monitored.

2. If necessary, loads are turned on/off, as in the case when second monopole must be turned on when over ground-station.

Data Storage and Retrieval

1. Primary microcontroller sends packets to secondary microcontroller (via SPI bus) as follows:

   (a) In nominal operation: HM packet (load + temperature data) – once every 2 seconds.

   (b) When over India (and not over Ground-Station): packets containing both position and HM data once is sent every 2 seconds.

   (c) Over Ground Station: packets containing both position and HM data once is sent every 2 seconds.

2. Secondary microcontroller processes the data as follows:

   (a) Packets containing HM data are stored in the EEPROM without packaging.

(b) After the power-on of the CC chip for downlink, the secondary microcontroller reads the EEPROM and transmits the data to the downlink microcontroller after packaging it into AX.25 format.

3. When downlink is in progress, the position and HM data packets are received by the downlink microcontroller from the secondary microcontroller.

3.5 Integration and Scheduler Design

The final portion of software design consists of putting together all the individual blocks to form the execution sequence. The determination of the execution sequence itself depends only on the list of tasks, their nature and the requirements on onboard computer.

Issues in design of execution sequence

- The tasks must be executed in the exact order and at the exact frequency as determined from the sub-system requirements. Thus emphasis must be given not only to the intelligent ordering of tasks –but also to their scheduling in a manner that allows the execution of all the tasks within a specified frame time.
- The presence of asynchronous events – interrupts from various interfaces, the downlink process – complicates the scheduling of blocks into a neat sequence. However, it must be noted that that the presence of interrupts allows for the delays associated with communication and sensor interfacing tasks to be bypassed – removing idle periods.
- The requirement for a fixed frame time necessitates the implementation of a system timer. This timer must be used to trigger execution of tasks at the appropriate times.
- For every task in the task list, worst case execution times must be calculated. It is imperative that the sum of the execution times of all tasks in a frame be smaller than the frame time itself.
- Certain measures must be taken so that when a task does not achieve completion within a reasonable time window, the microcontroller does not “hang”.

Implementation Two features greatly simplify the integration:

- The separation of communication tasks and control tasks allows us to write separate schedulers for the primary and secondary microcontrollers. The scheduler for the primary microcontroller comprises of the control and health monitoring tasks; the scheduler for the secondary microcontroller consists of the routines necessary for communication with downlink microcontroller and memory management.
- The tasks themselves have been broken down in to smaller modules/sub-blocks. Each of these sub-blocks is simple enough for us to reject very complicated scheduling algorithms. Further, the tasks are intrinsically sequential – concurrency is not required and therefore we can safely reject multi-threading / context switching.

A simple round-robin scheduler is proposed for both microcontrollers – tasks are placed in a predetermined sequence in a cycle and in every frame time (2 seconds for the primary microcontroller) all the tasks in the cycle are executed one by one. This simple design is thought to be sufficient for our purposes due to the existence of very few event driven / hard-real-time tasks. Downlink is the only “real-time” task – and the handling of this event (the event: entering the field of vision of the ground-station; the handling: downlink initialization) is taken care of by the primary, secondary and downlink microcontroller col- lectively. The start of each cycle of the primary microcontroller is triggered by the system clock (implemented using one of the 3 timers). This is the same timer that is used to trigger the switching on the GPS unit. The start of downlink will depend on the satellite’s position. This is re-evaluated every frame (2 seconds – either by the GPS or by the orbit estimator). The primary microcon- troller will decide when to initiate/terminate downlink. The secondary microcontroller’s cycle is triggered by commands/data received from the primary microcontroller. If health monitoring data is received, a cycle consisting of the following is triggered:

1. Receive and buffer all the data sent.
2. Once all the data has been received, store it in the EEPROM

If the command for downlink is received, a different cycle consisting of the following is triggered:

1. Command the power microcontroller to switch-on the downlink microcontroller which automatically configures the CC1101 chip.
2. The secondary microcontroller reads the data from EEPROM and transmits it to the downlink microcontroller until a ”Stop Downlink” command is received.

Chapter 4

Hardware Design

4.1 Requirements of the Hardware Design

4.1.1 System requirements

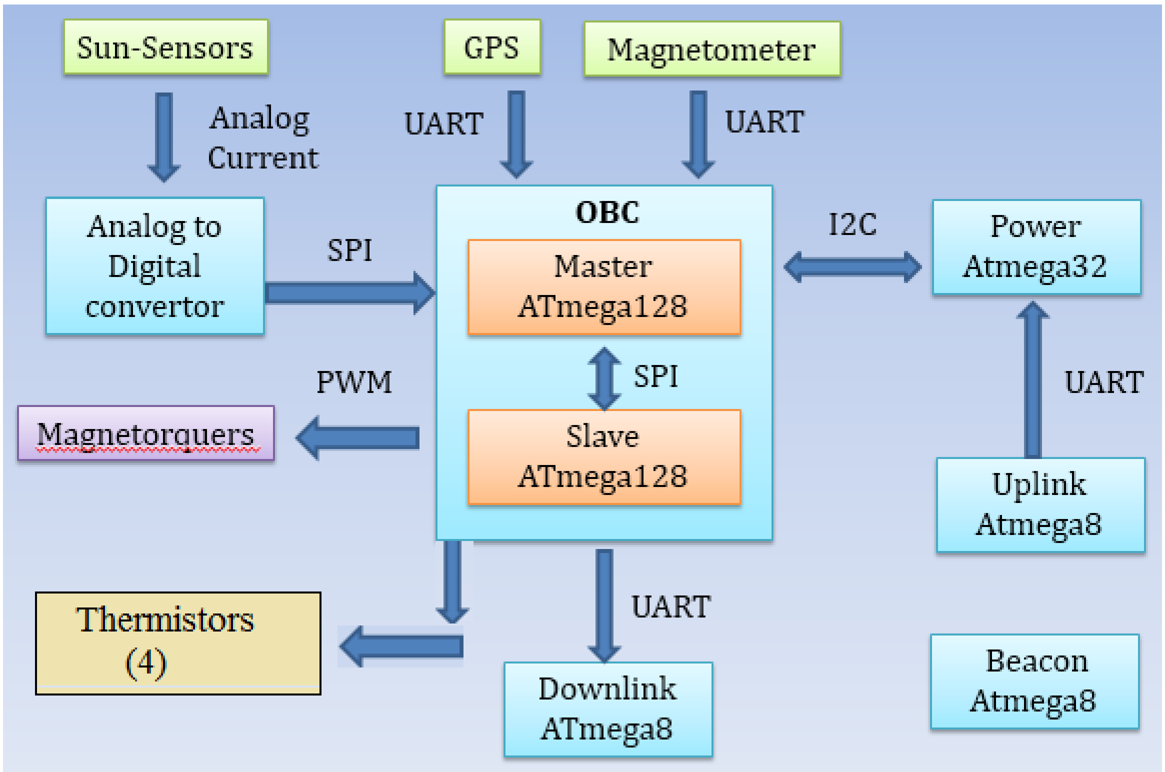
The Hardware of the Onboard Computer must satisfy the following requirements:

1. **Interfaces**: The Onboard Computer must be able to interface with the sensors employed by the ADCS subsystem. It must also interface with the CC1101 circuit employed by the communication subsystem. Finally, it should be able to drive the actuators (magnetorquers) and communicate with the microcontroller used by the power subsystem.

2. **Robustness**: The design must be robust enough to withstand the temperature and radiation conditions found in space. Ideally, space/industrial grade components should have been used, but due to the desire to use COTS components, the design should be robust enough to compensate for any internal failures and still provide some degree of functionality.

3. **Simplicity**: Pratham is the first project of its kind at IIT Bombay. Thus, emphasis was on designing a system using components we were familiar with. This included both the chips and the communication interfaces connecting them.

4.1.2 Basic Design and Interfaces



4.1.3 Microcontrollers

The tasks that have to be performed by the Onboard Computer are quite complex. Thus, we decided to use microcontrollers in place of other options like FPGAs. The main points that tipped the choice in favor of microcontrollers were:

1. Familiarity with their programming and use
2. Ability to change programming quickly, faster development and testing cycles
3. Versatile peripherals, no need for supporting circuitry
4. Easily available locally, cheap, ease of PCB design, fabrication,etc.

The specific microcontroller chosen was the Atmel ATmega 128. This micro-controller had been used for various other embedded systems projects in our institute. It also boasts of an incredible set of peripherals that would greatly simplify the rest of the design. Some of its features are:

1. 8 MIPS throughput at 8 MHz operating frequency
2. Low Power Consumption (3mW/MHz)
3. 4 KB on-chip SRAM, 128 KB on-chip Flash, 4 KB on-chip EEPROM

Some of the peripherals that are essential to the hardware design:

1. 3 timers (two 8-bit and one 16-bit) for PWM generation and system clock
2. Two-Wire Serial Interface (TWI or I2C) to interface with power micro-controller and external memory bank
3. Two independent UART lines to interface primary microcontroller with magnetometer and GPS; one UART line to interface the secondary micro-controller with downlink micro-controller.
4. On-chip Watchdog timer to restart micro-controller in case of system lock-up.

The micro-controller will be operated at the rated clock frequency of 8 MHz at 3.3 V.

The on-chip Flash memory will be used to store the program memory. While it is realized that the use of Flash increases the susceptibility of the system to radiation induced failures, it is felt that the use of an external PROM to store the program memory would overly complicate the design. Precautions against radiation have been taken by coating the PCB's. The execution of the control law in a speed optimized manner necessitates the use of several tables for computational purposes. These tables will be stored in the on-chip EEPROM and the on-chip Flash memories.To provide an accurate record of any failures or shut-downs and to keep a record of what the satellite was doing prior to these failures; a concise history will be maintained in the on-chip EEPROM. This will consist of flags signaling the state of the system before the most recent failure/shut-down. Sample computations, of a similar complexity as that of the final control law, have been executed on these microcontrollers and it has been determined that the amount of on-chip SRAM is sufficient for these purposes.

Need for 2 separate microcontrollers

Introducing 2 separate microcontrollers into the hardware design was necessary for a number of reasons:

1. Logical separation of the ADCS and the Communication portions of the Onboard Computer. The Primary microcontroller handles the interfaces with the sensors and actuators of the ADCS and also runs the control law. The secondary microcontroller interfaces with the CC1101 chip provided by the communications subsystem through the Downlink Microcontroller and handles the various subroutines involved in the downlink of data to earth.
2. Separation of data acquisition and data handling: The primary microcontroller is responsible for the collection of data from the sensors as well as receiving health monitoring data from the Power subsystem. The secondary microcontroller stores this data in the external memory bank and sends the data directly through the UART when indicated by the primary microcontroller

3. Simplification of the Operational sequence: When the satellite is over the ground-station, it must simultaneously execute 2 main functions – attitude determination and control, and downlink of accumulated data. In order to execute these tasks concurrently without the need for multi-threading or context-switching, these tasks are executed on two different microcontrollers.

4.1.4 Inter-Microcontroller Communication Interface

The two microcontrollers will be interfaced using the on-chip SPI peripheral communication interfaces. The primary microcontroller will always initialize any communication (act as a Master) and the secondary microcontroller will be hardcoded to act as the Slave. The SPI interface is a byte-oriented serial interface that is essentially two shift registers (one on each microcontroller) connected together. The transfer of bytes between this shift register is synchronized by a clock generated by the Master. The data carried by this channel will be described later.

4.1.5 Downlink Data

The mission statement requires the onboard computer to transmit two types of data.

1. **Attitude data**: In order to carry out atmospheric tomography, the satellite's position and attitude must be known to reasonable accuracy. The position of the satellite can be accurately tracked by ground-stations and with the aid of agencies such as NORAD. However, the attitude of the satellite can be best determined using the sensor readings transmitted from the satellite.

2. **Health Status**: As Pratham is the first student-satellite initiative at IIT Bombay, data regarding the progress of the mission will be immensely useful to subsequent missions. This data will consist mainly of the health of various components onboard, and the history of any component failures and will be stored on an external flash memory. This data will help us diagnose design faults and determine the gap, if any, between design conditions and operating conditions.

Features of External EEPROM

In order to store HM data, an external EEPROM, 24FC1025 from Microchip, has been chosen as the third component of the onboard computer hardware. The main features of the EEPROM are:

1. 128 KB serial EEPROM

2. Two-Wire Interface (I2C) interface (100 kHz or 400 kHz clock compatible)

3. Low Power Consumption (2.25 mW max)

4. High speed operation – 3ms page write.

5. 128 byte page (buffered write)

4.1.6 Memory Read/Write Interface

The memory is read from or written to using the on-chip I2C interface of the secondary microcontroller. The interface consists of one line for the clock (generated by the secondary microcontroller) and one bi-directional line for the data. The EEPROM allows for sequential and random reads, and page or byte writes.

INTER MICROCONTROLLER SPI BUS

- Bidirectional (uses SPI protocol)
- A permanent master (Microcontroller 1) and a Slave (Microcontroller 2) is used to transmit HM data to Microcontroller 2.
- Also, the access port can be used to program each of the microcontrollers by pulling low the appropriate RESET pin.

POWER STATUS COMMUNICATION BUS

- Bidirectional I2C bus
- Micro-controller 1 is always the master with the power micro-controller always the slave
  - Current status (ON/OFF) of various loads, battery voltage etc. on power bus, sent from power micro-controller
  - Required/desired status (on/off) of various loads sent from Micro-controller 1

UART BUSES

1. 3 independent bi-directional UART lines. Two of them on the primary micro-controller and the third on the secondary micro-controller.
2. GPS line
   - Used to configure GPS
   - Used to receive GPS data at regular intervals
3. Magnetometer line
   - Used to configure magnetometer after launch
   - Used to poll for and receive magnetic field data at regular intervals
4. Downlink line
   - Used to send data for Downlink from EEPROM when above groundstation

EEPROM R/W BUS

1. Bidirectional I2C bus
2. Used to store/retrieve HM data from EEPROM

4.1.7 Auxiliary components

The Onboard Computer main board will also host a few other circuits.

1. **H-Bridges**: The GPIO pins of the micro-controllers will not be able to drive the magnetorquers themselves. They will serve as the logic inputs to a H-Bridge which will then drive the torquers with the required currents. The micro-controller output pins will provide the PWM signal which will act as inputs to the enables of the magnetorquers.
2. **Thermistors**: 4 thermistors of industrial grade are used to monitor temperature of Battery box, Uplink LNA, Downlink's and Beacon's power amplifier, which are thermally critical positions in the satellite. This data is converted to digital format and then combined in the Health monitoring data by secondary micro-controller.

Chapter 5

Hardware Testing

This chapter details the methods which were used to test the sensors and flash memories attached to the OBC. These include the following

- GPS Unit
- Magnetometer Unit
- Magnetorquer

The description of the testing of all communication links on the OBC will be described in the chapter on environmental testing which includes integrated testing of the satellite.

5.1 GPS

The GPS was tested out at place with a clear view of the sky. The terrace of the Aerospace Engineering Department was chosen for this purpose.

5.1.1 Phase 1

The GPS was connected to the primary microcontroller (Master) on the OBC and the output data was transmitted via UART to a PC where it was observed on the serial terminal. The following procedure was then followed:

1. The output of the GPS was first independently observed via UART on the GUI provided by the manufacturer, Accord.
2. On verifying that the GPS puck antenna was sending the correct data, the GPS was connected to the Master OBC and the output was observed on the terminal.

Observation

As expected from the datasheet, after about 80 seconds, the puck antenna started sending the correct data and the array was getting updated in every loop of the Master OBC. The change was observed on the terminal.

5.1.2 Phase 2

After establishing that the GPS is working, the next step was to check the effect of the view factor. The GPS puck antenna was mounted on the zenith side and the solar panel on the zenith reduces the view factor of the puck antenna. Hence, it was necessary to study the effect of the solar panel on the GPS performance.

Testing Environment

The GPS receiver was mounted on the zenith side with the mock solar panel placed appropriately. The satellite was placed on the Nadir side. Therefore, the zenith side was facing the sky.

Observations

The solar panel had negligible effect on the GPS performance and the correct data was sent by the puck antenna in the same time as in the case without the solar panel.

5.1.3 Phase 3

The satellite was turned so that the zenith was no longer on the top, i.e. the zenith was no longer sky-facing. This was done to simulate the worst case scenarios and find out that to what extent can we reduce the view factor. This test was performed in two steps

1. The satellite was placed on the sunside so that the zenith would be one of the lateral faces.

2. The satellite was inverted such that the zenith side was the bottom most side and the Nadir side was sky-facing.

Observations

- In the first case, the time taken by the puck antenna to send the correct data increased to 120-140 seconds.

- In the second case, correct data was not received even after 180 seconds.

5.1.4 Sources of Error

In all the tests performed above, multiple things could further constrain the view factor. For instance, the presence of humans around the GPS puck antenna etc.

5.2 Magnetometer

The testing of magnetometer majorly consists of calibration. Testing the working of the device is done by simply connecting it to a power source, sending a poll command, receiving the data and viewing it on a serial terminal.

5.2.1 Calibration

- Calibration was carried out using a Helmoltz coil setup. The magnetometer was placed on a platform with one of its axes aligned with the axis of the coils and a current was passed through the coils to generate a known magnetic field.

- The field was then recorded by the magnetometer and after the field value settled, a time averaged reading was recorded and compared to the known value.
- The field was varied from -1 to 1 Gauss.
- The calibration curve along all the axes was observed to be linear.

5.2.2 Results

- The result for X axis is as shown below (squared correlation coefficient (R^2) = 1; RMS error: 8.9×10^{-4})

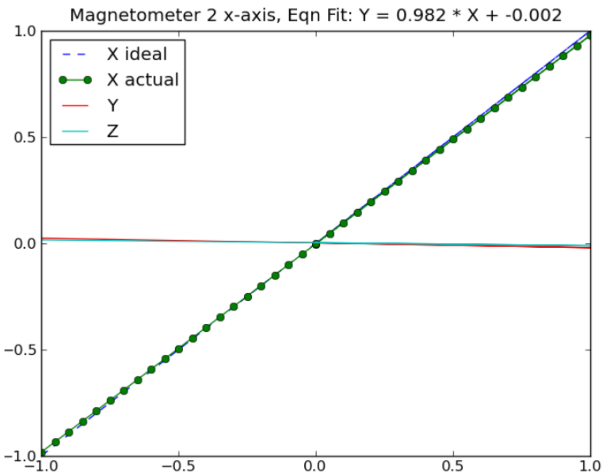


Figure 5.1: Calibration along X-axis

- The result for Y axis is as shown below ($R^2 = 1$; RMS error: 9.5×10^{-4})

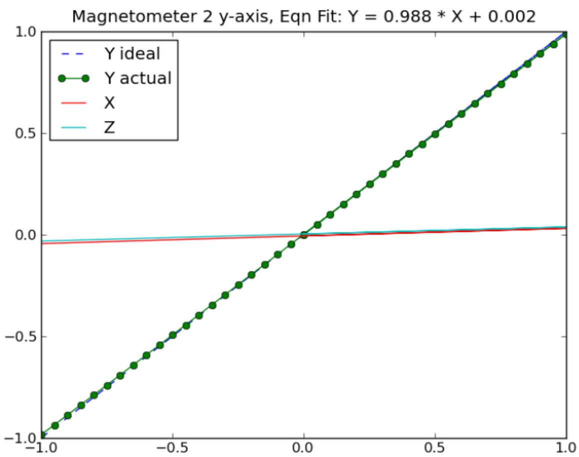


Figure 5.2: Calibration along Y-axis

- The result for Z axis is as shown below ($R^2 = 1$; RMS error: $3.6x \times 10^{-4}$)

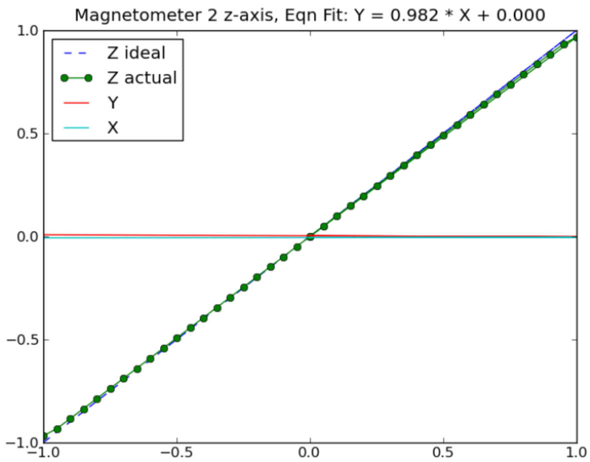


Figure 5.3: Calibration along Z-axis

5.2.3 Sources of Error

- Error due to electronics of the satellite: The magnetometer is surrounded by other satellite electronics, which have a magnetic field of their own. To minimize this error, the magnetometer should be placed as away as possible from the other electronics. Hence the choice of the lagging face, which is relatively free of all other electronics. Also wire routing is done in such a way that wires carrying currents in the opposite direction are placed together. This prevents formation of loops and thus stray moment.

- Error due to the satellite body: Due to the metallic body of the satellite, there is magnetization error which can change the magnetic field. Coarse experiments done by covering the magnetometer with an aluminium box, give an error in the magnetic field of 2 mgauss.

- Mounting error: There will always be a misalignment between the magnetometer axes and the satellite body axes. Tolerances are given to the integrations team for the same.

- Error due to temperature variation: Temperature varies from -10 to 60 degree Celsius. This introduces a scaling error, the maximum value of which is 2.1

- Drift error: The magnetic field reading at the same point varies with time. This is known as drift. Experiments were performed, which put a value to the drift as approx. 2.5 μgauss/s.

- Bias and random error: The maximum bias is 3×10^{-4} gauss. The maximum random error is 52×10^{-4} gauss.

- Sudden changes in magnetic field (of the factors of 4 and 5) will be taken care of by putting checks on the OBC.

5.3 Magnetorquer

The torquer was tested using a helmontz coil setup. The Helmholtz coil has a radius of 20 cm. The axial distance between the Helmholtz coil used was 24 cm. The size of the torquer side is 21 cm.

5.3.1 Process

1. The torquer was suspended between the helmholtz coil through a string ensuring single point of contact at the knot, perpendicular to the magnetic field generated to produce maximum torque.
2. A gauss meter as seen in **??** measures the magnetic field through a probe between the two coils.
3. When the torquers were OFF, the currents were changed in the helmholtz coil till the magnetic field reached between 0.7 and 0.8 gauss (as magnetic field of earth is of this order).
4. Then the torquer was supplied currents at various voltages.
5. On supplying current, angular deflections were observed in the torquer.

5.3.2 Observation

The amplitude of angular deflections were low initially when constant currents were provided but increased when the currents were changed dynamically by changing the input voltage.

Chapter 6

Thermo-vaccum Testing

6.1 Introduction

Thermovac testing for Qualification model of Pratham was done at ISAC from 9th November to 11th November under the guidance of scientists Khened Sir, Ramachandran Sir, Ram kumar Sir and Harish Sir. Thermovac testing aims to test and validate the thermal deign and components sustainability for thermal loads during orbiting of satellite. The satellite has undergone three transient thermal cycles followed by cold soak and hot soak of 3 hrs. The complete testing runs for 37 hours. Battery (for lower side temperature) and the Power amplifier on beacon (for higher side temperature) were the critical components for deciding the thermovac profile. subsection Testing Environment

- Pressure of the chamber: $< 10^{-5}$ mbar
- Lower temperature limit: -5° C on the battery
- Upper temperature limit: +55°C on the Power Amplifier of the beacon board
- Rate of change of temperature: 1°C/minute

During thermovac testing 16 thermistors and 6 thermocouples were used. Thermocouples were provided by ISAC while thermistors were brought by PRATHAM team. Out of 16 thermistors, 12 thermistors are mentioned in the table 6.1. Rest of the 4 thermistors were placed at the location of thermistors: T1, T2, T3 and T4. The data coming out of these thermistors was received at house-keeping telemetry.

SI. No.	Thermistor	Thermistor Location
1	T1	Battery
2	T2	Downlink
3	T3	Beacon
4	T4	Uplink
5	T5	PTH-1
6	T6	PTH-2
7	T7	PTH-3
8	T8	PTH-4
9	T9	Solar Panel (Sunside)
10	T10	Solar Panel (Lagging)
11	T11	Solar Panel (Zenith)
12	T12	Solar Panel (Leading)

Table 6.1: Location of Thermistors

SI. No.	Channels No.	Location
1	1	FE Ring (Lagging Side)
2	2	Anti-sunside Panel
3	3	Chamber Base Plate
4	4	Chamber Base Plate
5	9	Shroud Temperature
6	10	Shroud Temperature

Table 6.2: Thermocouple Location

6.1.1 Testing Sequence

- Functionality check was performed
  - Before closing the chamber
  - After closing the chamber and before de-pressurizing
- After de-pressurizing, the satellite was in OFF state till the battery temperature first reached -5°C in the first thermal cycle
- After that, SNAP was done and the satellite went in sleep for further 5 minutes before turning ON
- House-keeping telemetry was kept on for most of the time till the cold soak began
- Once in the cold soak, the satellite was turned off for 2 hours till the temperature stabilized.
- After that, SNAP was done and functionality check was subsequently performed.
- After the functionality check, the satellite was again turned OFF and the temperature was increased to reach the beginning of the hot soak
- While in hot soak, the satellite was kept OFF for 2 hours for the temperature to stabilize
- After that, SNAP was done and functionality check was subsequently performed.
- After the functionality check, the satellite was again turned OFF and the chamber was allowed to reach room-temperature and was slowly pressurized
- After opening the chamber, the functionality check was performed again.

6.1.2 Functionality Check - Details

All the sensors, actuators and peripherals were monitored throughout the test. House-keeping telemetry was almost continuously ON and payload telemetry was also tested during Hot and Cold soaks (see figure 6.3). The temperature of the critical components were monitored throughout the test. We now present a brief overview of the methodology used to test the critical components/parameters:

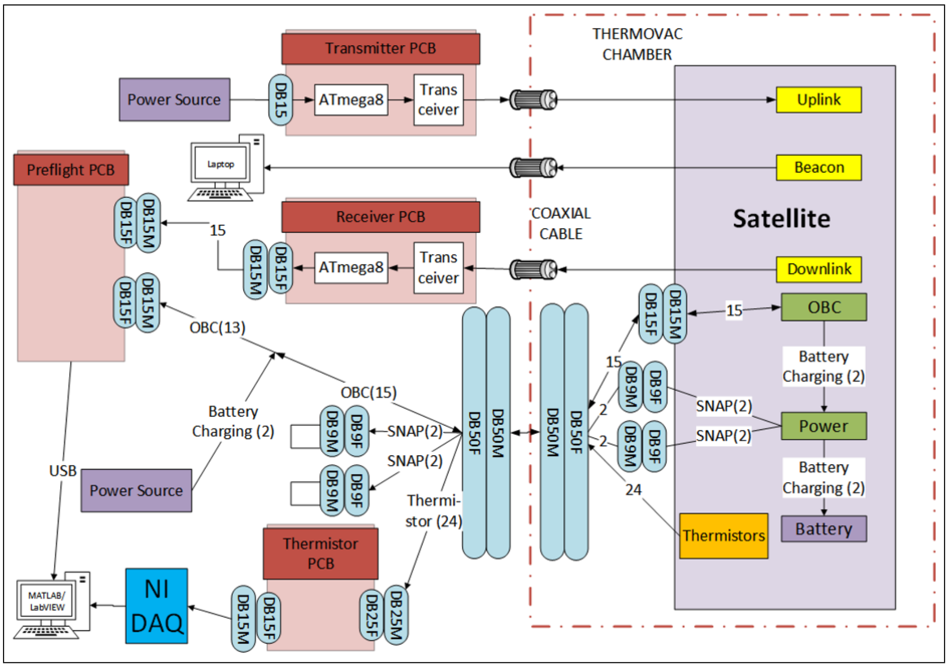


Figure 6.1: Functionality Check Schematic

- **Sun-sensors:**
  - The Analog current was converted to Digital voltage and transmitted as a part of the house-keeping telemetry. The change in value on the application of a torch-light was observed both before and after the tests.
- **GPS:**
  - Transmission of data by the GPS was verified via house-keeping telemetry. The verification of data was not possible as the puck antenna of the GPS does not get give correct data indoors.
- **Magnetorquers:**
  - Command to start each magnetorquer successively for 30 seconds was given and the corresponding changes in the values of the magnetic field indicated by the magnetometer were observed

6.2 Temperature Profiles

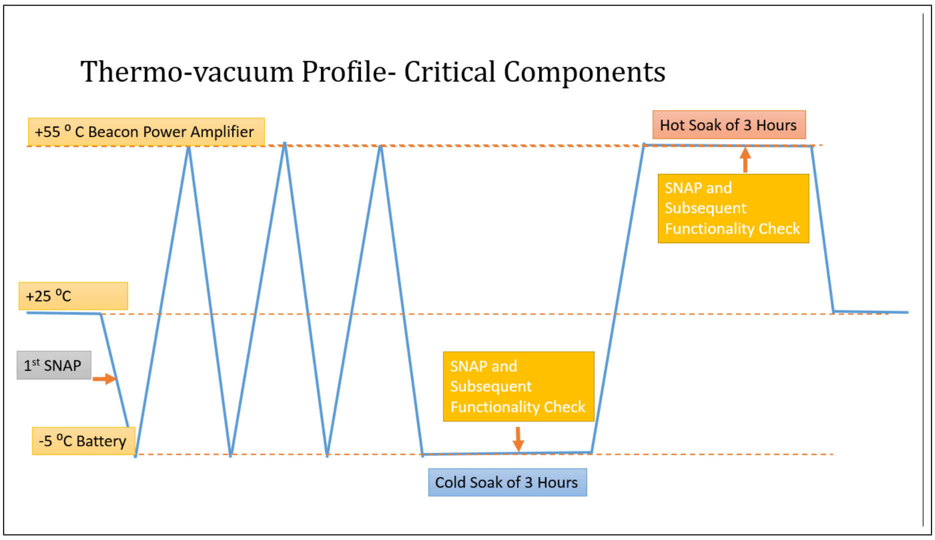


Figure 6.2: Expected Thermo-vacuum Profile

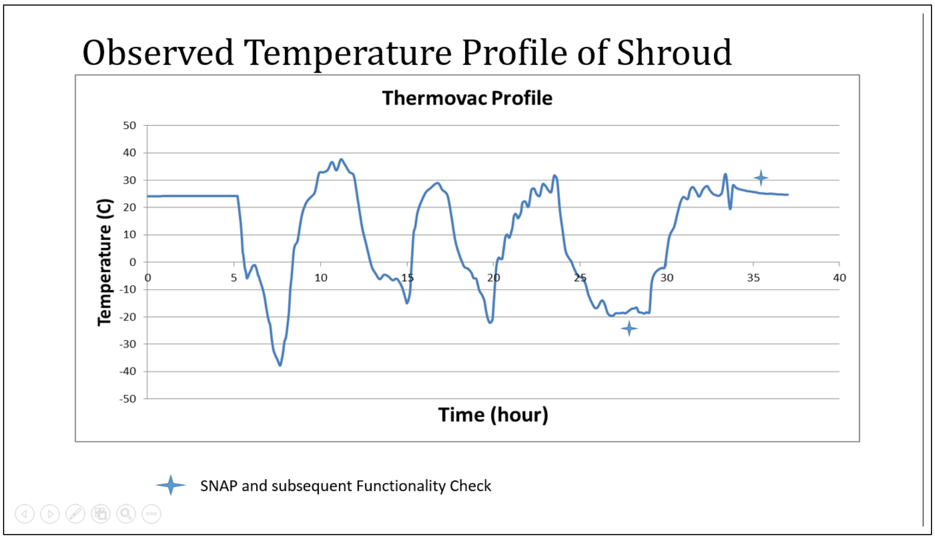


Figure 6.3: Observed Shroud Temperature Profile

6.3 Functionality Check - Results

6.3.1 Health Monitoring Data Results:

Data	**Unit**	**Expected Value**	**Observed Value**
Total Solar Panel Incoming Current	Ampere	0 - 1.2	0 - 1
Battery Voltage	Volts	6-8.4	7.5-8.3
Downlink Voltage	Volts	5	5
OBC Voltage	Volts	3.3	3.28
Over current Byte Status	No over current	1	1
	over current	0	
Load Status	Beacon	0: Off/ 1: On	1
	Torquer	0: Off/ 1: On	1
	GPS	0: Off/ 1: On	1
	Downlink	0: Off/ 1: On	Value switching as expected
	OBC	0: Off/ 1: On	1
	Magnetometer	0: Off/ 1: On	1
Load Consumption Current (Ampere)	OBC+Magnetometer +Beacon	approx 0.75*	Case not observed
	OBC+Magnetometer +Beacon+GPS	approx 1.17*	0.75
	OBC+Magnetometer +Beacon +GPS+Downlink	approx 1.77*	1.1
	OBC+Magnetometer +Beacon +GPS+Uplink	approx 1.77*	0.88

Table 6.3: Health Monitoring Data - Expected and Observed Values

* : Absolute maximum values

Inferences:

- All the critical voltages and currents are within the expected range
- Over-current was not observed anywhere
- The Load consumption currents are much lower than expected as the expected values are the absolute maximum values. Beyond these values, overcurrent signal would have been be generated.

6.3.2 Sensor Results

Data	**Condition**	**Expected**	**Observed**
Sunsensors	During the Test	300-600 (for each sunsensor)	All values within range
	Before and after the test, on the application of torchlight	>600	All values above 600 except sun-side sunsensor
GPS	During the test	-	Transmission by GPS was verified by data in house-keeping telemetry
Magnetometers	During the test	-	Constant values, which changed when magnetorquers were turned ON

Table 6.4: Sensors - Expected and Observed Values

Inferences:

- No problems were observed with any sensors
- The change in magnetic field values when the magnetorquer was turned ON indicate that the magnetometer is working well
- The increase in the values of the sunsensors on the application of a torch-light both before and after the tests indicate that the sunsensors were not damaged during the test. (Note: The sunsensor on the sun-side was not working before Vibration Test itself)

6.3.3 Sensor Results

Data	**Condition**	**Expected**	**Observed**
Sunsensors	During the Test	300-600 (for each sunsensor)	All values within range
	Before and after the test, on the application of torchlight	>600	All values above 600 except sun-side sunsensor
GPS	During the test	-	Transmission by GPS was verified by data in house-keeping telemetry
Magnetometers	During the test	-	Constant values, which changed when magnetorquers were turned ON

Table 6.5: Sensors - Expected and Observed Values

Inferences:

- No problems were observed with any sensors
- The change in magnetic field values when the magnetorquer was turned ON indicate that the magnetometer is working well

- The increase in the values of the sunsensors on the application of a torch-light both before and after the tests indicate that the sunsensors were not damaged during the test. (Note: The sunsensor on the sun-side was not working before Vibration Test itself)

6.3.4 Actuator Results

Condition	Magnetic Field - x component	Magnetic Field - y component	Magnetic field - z component
All torquers OFF	2023	3731	2917
Torquer X turned ON	1944	3594	3592
Torquer X turned OFF and Torquer Y turned ON	2543	1876	640
Torquer Y turned OFF and Torquer Z turned ON	2960	4024	2274

Table 6.6: Changes in Magnetic Field values on Turning ON Torquers

Inferences:

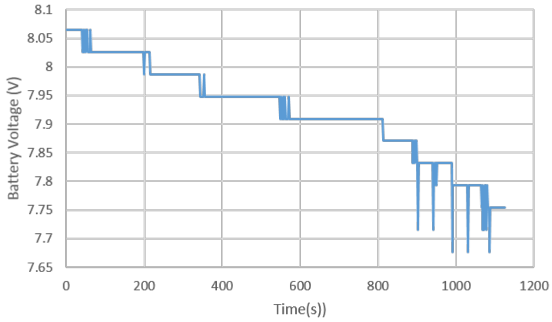
- As can be seen from table 6.6, the change in magnetic field values on turning ON the actuators was satisfactorily observed

Notes:

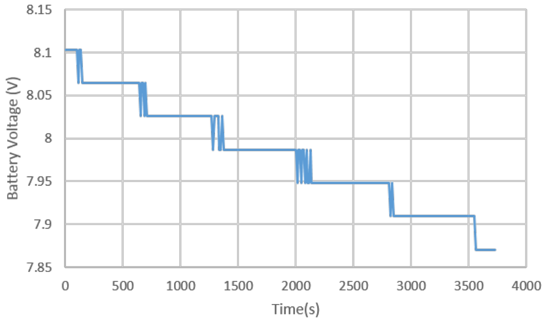
- The magnetorquers were successively turned ON for **30 seconds** duration
- The values were taken after more than 20 seconds of switching to each condition. The magnetometer values had become constant by then.
- X axis: Lagging to Leading
- Y axis: Sun-side to Anti sun-side
- Z axis: Zenith to Nadir

6.3.5 Verification of Payload Telemetry

Cold Soak



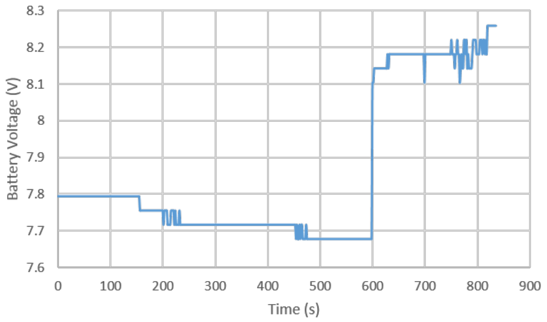
(a) Battery Voltage - Payload Telemetry



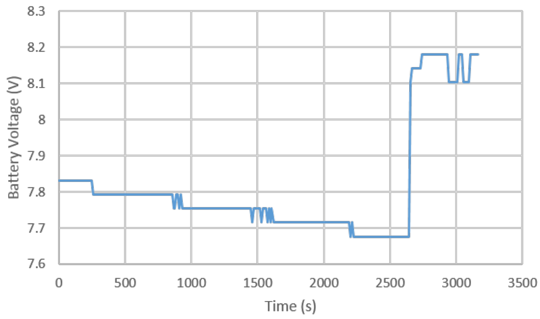
(b) Battery Voltage - House-keeping telemetry

Figure 6.4: Cold Soak

Hot Soak



(a) Battery Voltage - Payload Telemetry



(b) Battery Voltage - House-keeping telemetry

Figure 6.5: Hot Soak

Inferences:

- The payload telemetry data matches well with the house-keeping telemetry data. A byte-by-byte check has also been done to check the same
- It appears that the payload telemetry is more informative. This is correct and the reason is that there is sub-sampling taking place. Only one of every five sets of data stored in the EEPROM is transmitted as house-keeping telemetry.