SOFTWARE DEVELOPMENT OF C & DH USING FREERTOS FOR STUDSAT-2

A Project Report Submitted By

Bhuvanesh S K	**4NM16EC026**
Pooja Pai	**4NM16EC087**
Shraddha S	**4NM16EC130**

Under the Guidance of

Mr. Durga Prasad
Associate Professor

in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Engineering
in
Electronics and Communication Engineering

from
Visvesvaraya Technological University, Belagavi

NITTE EDUCATION TRUST | NMAM INSTITUTE OF TECHNOLOGY
Nitte-574110, Karnataka, India

July-2020

NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India
ISO 9001:2015 Certified, Accredited with 'A' Grade by NAAC
☎:08258 - 281039 – 281263, Fax: 08258 – 281265

Department of Electronics and Communication Engineering

Certificate

Certified that the project work entitled ***Software Development of C&DH using FreeRTOS for STUDSAT-2*** *is a bonafide work carried out by* ***Bhuvanesh S K (4NM16EC026), Pooja Pai (4NM16EC087) & Shraddha S (4NM16EC130)*** *in partial fulfillment of the requirements for the award of* ***Bachelor of Engineering Degree in Electronics and Communication Engineering*** *prescribed by Visvesvaraya Technological University, Belagavi during the year 2019-2020. It is certified that all corrections/suggestions 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 requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.*

Signature of the Guide **Signature of the HOD**

Signature of the Principal

Semester End Viva Voce Examination

	Name of the Examiners	Signature With Date
1.	_____	_____
2.	_____	_____

Abstract

Project STUDSAT-2 is India's Twin Nano Satellite Mission. It hosts four payloads namely Inter-Satellite Link (ISL) to prove in-orbit Communication between satellites, Drag Sail Mechanism to de-orbit the satellite at the end of mission, Automatic Identification System (AIS) for ship Identification and Tracking and a Beacon data relay for HAMs. The satellite system of STUDSAT-2 consists of six subsystems i.e.Payload, Attitude Determination and Control System(ADCS), Electrical Power System(EPS), Command and Data Handling (C&DH), Mechanical Structure and Communication sub-system are the main functional blocks of the satellite. C&DH sub-system which is responsible for on-board control, data handling, decision making and processing. The software deployed on top of FreeRTOS operating system should handle multiple tasks based on their priority levels. The project aims at designing such software on top of real-time operating system. Multiple tasks which are supposed to be demonstrated are Timer call back function with ten threads for six current sensors and four temperature sensors. The programming section involves in the design such that the tasks are scheduled using pre-emptive priority scheduling algorithm. The software section of the project aims at designing and demonstrating command and data handling ability of the system using six current sensors and four temperature sensors to demonstrate that software will run concurrent tasks.

Acknowledgement

Our project would not have been successful without the encouragement, guidance and support by various personalities. First and foremost, we would like to express our sincere gratitude towards our project guide **Mr. Durga Prasad**, Associate professor, Department of Electronics and Communication Engineering, N. M. A. M. Institute of Technology, Nitte, for his guidance, encouragement and inspiration throughout the project work.

We are thankful to **Mr. Guruprasad C Kulkarni** and **Ms. Pooja K P**, Research Associates, Team STUDSAT, Department of Electronics and Communication Engineering, N. M. I. T. , Bangalore for providing us an opportunity to be a part of the wonderful project "STUDSAT-2" , and also a 10-days training for the same at N. M. I. T. , Bangalore. We express our gratitude for their constant support by providing us the necessary components and guidance to overcome the challenges throughout the project.

We extend our gratitude to **Dr. Rekha Bhandarkar**, Professor and Head, Department of Electronics and Communication Engineering, N. M. A. M. Institute of Technology, Nitte, for her encouragement and providing necessary facilities in the department.

We wish to acknowledge the support of **Dr. Niranjan N. Chiplunkar,** Principal, N. M. A. M. Institute of Technology, Nitte, for providing motivational academic environment.

We would like to express our sincere thanks to all the teaching and non-teaching staff of the department of Electronics and Communication Engineering, N. M. A. M. Institute of Technology, Nitte, for their patience and help during our project work.

Finally, we would like to thank all our friends and well-wishers who have helped us whenever needed and supported us.

Bhuvanesh S K
Pooja Pai
Shraddha S

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

STUDSAT-2, aims at proving in-orbit separation, establish inter- satellite communication. Out of six subsystems of the satellite, Command and Data Handling is the brain of the satellite which is responsible for on-board control and processing. The project aims at designing a software on top of real-time operating system to handle multiple tasks based on their priority levels. In this chapter, the description of the project structure with background studies done for dealing with it are explained.

1.1 General Introduction

STUDSAT-1, India's first pico-satellite, was successfully launched under the guidance of Indian Space Research Organisation (ISRO) [1]. The new mission, STUDSAT-2; a twin nano satellite, aims at proving in - orbit separation, establish inter - satellite communication, implementation of drag sail technology to de-orbit the slave satellite and capturing earth image using Complementary Metal-Oxide Semiconductor (CMOS) camera [2]. The satellite system of STUDSAT-2 consists of six subsystems i.e. Structure, Attitude Determination and Control System (ADCS), Electrical Power System (EPS), Command and Data Handling (C&DH), Communication and Payload. The ADCS is responsible for maintaining the desired orientation of the satellite in the orbit. The satellite employs a combination of sensors and actuators to achieve the same. EPS is the major subsystem which supplies regulated power for proper functioning of all the subsystems. The solar energy is harnessed by the solar panels and converted to electrical energy which is stored in Li-ion batteries. The objective of Communication and Payload system is to establish communication link with ground station and transfer the payload and telemetry data to the ground station using full-duplex mode of communication. The structure of the satellite is designed considering the dimensions 30 cm x 30 cm x 20 cm. C&DH subsystem is the brain of satellite, i.e. responsible for on-board control and processing. All the subsystem operations are mastered by an On-Board Computer (OBC) system. The sensor data and other important information about the outer space environment has to be down-linked to the earth station. C&DH will manoeuvre the transmission of all these information. Real-time computing is where system correctness not only depends on the correctness of the logical result but also on the result delivery time. So the operating system should have features to support this critical requirement to render it to be termed a Real-time Operating System (RTOS).

The RTOS should have predictable behaviour to unpredictable external events. The project is to develop software for C&DH using FreeRTOS for STUDSAT-2. C&DH regulates the processing of the tasks according to their priorities and all the subsystems are controlled by the OBC.

1.2 Problem Definition

To demonstrate delay of 45 minutes using Timer before the solar panel deployment and to design multitasking operation for C&DH system using the software coding and demonstrate with the help of six current sensory modules and four temperature sensory modules.

1.3 Objectives of the Project

The project objectives are:
1.) To switch the Satellite to 'normal' mode post ejection after a delay of 45 minutes, prior to which it will be in power suspended mode.
2.) To deploy the antenna once the Satellite attains 'Normal' mode to establish communication.
3.) To initiate an event based separation of Satellite, after confirming the attitude stability.
4.) To deploy the solar panels once the Satellite is separated.
5.) To read and record the Telemetry logging data like Current sensor data and Temperature sensor data.

1.4 Methodology

The first task is to use an inbuilt Real-time clock for Post Ejection where the satellite switches to normal mode after a delay of 45 minutes.This is followed by deployment of solar panels once the satellite is been separated and stabilized. The second task is to interface the temperature sensor with microcontroller to collect temperature data. The third task is to interface six current sensors in the I2C protocol format to communicate with the microcontroller as master and collect the bus voltages and shunt current through the shunt resistor. Second and third tasks are pertaining to Telemetry logging data.

1.5 Literature Survey

The background work discusses the need for RTOS in the project and also aims at a case study on RTOS selection for the project, software development for Telecommand and Telemetry handling for STUDSAT-2. A good RTOS should have these properties: Multi-tasking and preemptable, should be able to identify the deadline within which the task should be executed, should be capable of predictable synchronization, should have sufficient priority levels and should possess predefined latencies. A Process Scheduler schedules different processes to be assigned to the CPU based on particular scheduling algorithms. Following are the scheduler plans that are studied: First-Come First- Served (FCFS) Scheduling, Shortest-Job-Next (SJN) Scheduling, Priority Scheduling, Shortest Remaining Time, Round Robin (RR) Scheduling, Multiple- Level Queues Scheduling. Case study of RTOS selection is done through referring different papers on satellites and their RTOS and comparative analysis is done.

Customized RTOS: Some have used a customized RTOS which is more flexible. However, basic features are retained and additional features are added such as RTOS used in skCube [3] Satellite uses only one static pointer and also provides emulation of lockstep with double scheduling whereas DANDE [4] has preemptive scheduling driven by watchDog reset and also has an interrupt to be raised in case of overvoltage in the hardware parts.

XilKernel: XilKernel RTOS implemented on Spartan-6 Field Programmable Gate Array (FPGA) [5]. The disadvantages of the work are FPGA sensitivity to ionizing radiation is not fully known, research and testing techniques for tolerance to error are yet to be determined.

Table 1.1: List of different satellites with the specifications of RTOS used

Satellite	**Design**	**Type of RTOS**	**onBoard computer**	**Algorithm**	**Reference Paper**
Studsat 2	Nitte Meenakshi Institute of Technology, Bangalore, India	FreeRTOS	STM32 ARM Cortex M4	Priority-based preemptive	[6]
Parikshit Nanosatellite	Manipal Institute of Technology, Manipal, India	Micrium RTOS	STM32 F207	Preemptive Scheduling	[7]
Moroccan University Nano Satellite	Moroccan University, Morocco	XilKernel	Spartan 6	Preemptive Scheduling	[5]
UGMSat 1	Satellite and Aerospace Research Group, Yogyakarta, Indonesia	CooCox CoOS RTOS	NXP LPC1769, 32 Bit ARM Cortex M3	Preemptive priority	[8]
Twin nanosatellite StudSAT2	Nitte Meenakshi Institue of Technology, India	Free RTOS	STM32F4 Discovery	Priority and preemptive	[9]
skCube	Slovak University of Technology, and Slovak Academy of Science	Customised RTOS	MSP430 Launchpad	Scheduler	[3]
DANDE	Colorado Space Grant Consortium, Boulder (COSGC), CO; NASA	Customised RTOS	Atmel NGW 100 mkII	Preemptive Scheduling	[4]
IiNUSAT 1	Indonesian National Institute of Aeronautics and Space, Bogor, Indonesia	CooCox CoOS RTOS	ARM9	Priority-based and preemptive	[10]

CooOx CoOs: CooOx CoOs used in IiNUSAT 1 [10] and UGMSat 1 [9] is an embedded RTOS which is free and open-source but being specially designed for Cortex M4 makes it difficult to interface with other devices. The platforms supported by the OS are ICCARM, ARM GCC, and GCC. Otherwise, it is fairly good

RTOS to be used for nano-satellites.

Micrium III RTOS: One major advantage of Micrium III is that it is highly scalable [7]. This RTOS occupies very little memory and does not impose restrictions on the declaration with respect to the number of Kernel objects as long as it does not go out of memory range of the RTOS. Considering this feature, to avoid priority inversion it also has 256 priority levels which makes the preemptive scheduling more effective and efficient.

FreeRTOS: FreeRTOS is a free and open-source RTOS which implies that it can be tweaked as per the user's needs. FreeRTOS also supports a wide range of platforms also which makes it more versatile and desirable [8]. It's design has been developed to fit on very small embedded systems and implements only a very minimalist set of functions, a very basic handle of tasks and memory management, just sufficient Application Programming Interface (API) concerning synchronization, and absolutely nothing is provided for network communication, drivers for the external hardware, or access to a files-system. Since most of the code is written in the C programming language, it is highly portable and has been ported to many different platforms. Its strength is its small size, making it possible to run where most (or all) other real-time operating systems won't fit [6]. FreeRTOS is designed to be simple, portable and concise. FreeRTOS has much smaller requirements when it comes to the amount of memory. FreeRTOS has a simple highest priority first scheduler. It has much less scheduling overhead.

1.6 Organization of the Report

The rest of the report is organized as follows:
Chapter 2 gives an insight into STUDSAT and its missions.
Chapter 3 details the hardware components and software tools used for implementation.
Chapter 4 explains methodolgy, calibration, block diagram, flowchart and schematics of all the modules.
Chapter 5 details the testing and results of the project.
Chapter 7 presents conclusion and future work.

Chapter 2
STUDent SATellite (STUDSAT)

STUDSAT is a student satellite conceptualized, designed and implemented by undergraduate students across India. This will be the smallest satellite ever developed by any Indian University indigenously. The mission of the project is to design and experiment on various subsystems of satellite with a major objective of providing hands-on experience in the design, fabrication and realization of a space mission at a minimum cost. The uniqueness of this project is it has been initiated by students.

2.1 STUDSAT-1

STUDSAT-1 is a pico-satellite, a miniaturized satellite, successfully launched on 12 July 2010 from Satish Dhawan Space Centre into a sun synchronous orbit. This was the first pico-satellite launched by India, as well as the smallest satellite launched indigenously by ISRO. The mission life was stated to be six months. The satellite had all the subsystems which are present in the bigger satellites. The various subsystems were:

- Structure
- Attitude Determination and Control System (ADCS)
- Communication
- Command and Data Handling (C&DH)
- Electronic Power System (EPS)
- Ground Station
- Payload

The satellite was close to being a cube when compared with the common satellites. It had dimensions of 10 cm x 10 cm x 11.35 cm. It weighed just 950 gm and had a volume of 1.1 litres and hence fell into the category of pico-satellites. The satellite was launched in aPolar Satellite Launch Vehicle (PSLV) into a 647 km solar synchronous orbit. The functional objective of the satellite was to perform remote sensing and capture images of the surface of the earth using its camera of resolution 95 m.

Figure 2.1: Prototype of STUDSAT-1

The pico-satellite was designed to operate in Low Earth Orbit (LEO) at an altitude of 647 km. The payload of the satellite was a CMOS camera which was capable of capturing images with a ground resolution of 90 m. Prototype for STUDSAT-1 is shown in Figure 2.1.

2.2 STUDSAT-2

The Team STUDSAT has continued the legacy and is building Twin Nano satellites for proving the concept of Inter-Satellite Link (ISL). The design of the twin-satellites, STUDSAT-2A and STUDSAT-2B are of the dimensions 30 cm x 30 cm x 20 cm and weighing less than 10 kg. The main goal of the STUDSAT-2 project is to develop a low-cost small satellite, capable of operating small scientific or technological payloads where real time connectivity is provided by ISL. The prototype of STUDSAT-2 is shown in Figure 2.2.

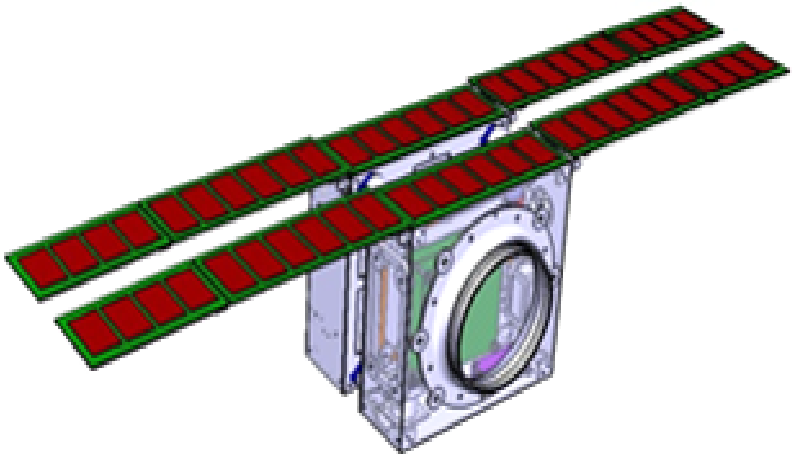


Figure 2.2: Prototype of STUDSAT-2

2.2.1 Project Organization of STUDSAT-2

The consortium of STUDSAT-2 consists of the following Engineering institutions affiliated to Visveswarayya Technological University (VTU), Belagavi with NMIT, Bengaluru as the lead Institute.

- Nitte Meenakshi Institute of Technology (Lead College), Bengaluru
- N.M.A.M. Institute of Technology, Nitte, Karkala
- R.N.S. Institute of Technology, Bengaluru
- M.S. Ramaiah Institute of Technology, Bengaluru
- Sri Sidhartha Institute of Technology, Tumkur
- Nagarjuna College of Engineering and Technology, Bengaluru
- Siddaganga Institute of Technology, Tumkur

2.2.2 Specifications of STUDSAT-2

The specifications of STUDSAT-2 satellite are as follows:

- Category: Nano satellites
- Mass: 5 kg and 4.5 kg
- Dimension: 30 cm x 30 cm x 20 cm
- Payload: CMOS Camera, Inter Satellite Link(ISL), Automatic Identification System (AIS).
- Resolution: 40 m
- Swath: 40 x 30 km
- Inter-Satellite Link Frequency: S-band (2.4 GHz 256 Kbps)
- Antenna type: Patch antenna
- Communication:
    - Downlink: UHF (434 MHz - 437 MHz), Antenna type: Monopole
    - Uplink: VHF (144 MHz - 147 MHz), Antenna type: Monopole
    - Beacon: UHF (434 MHz - 437 MHz), Antenna type: Monopole

- Attitude Determination and Control System (ADCS): 3-Axes Stabilization with nadir pointing accuracy less than 1 degree
- Sensors: GPS, Sun sensor, Magnetometer, Gyro
- Actuators: Magnetic Torquer Coils and Reaction Wheel
- Controller: ARM Cortex M4
- OS: FreeRTOS
- Battery: Lithium Ion Battery(5.2 Ah)
- EPS Efficiency: Above 85%
- Solar Panels: Body mounted and deployed with solar cells of 18% efficiency
- Mission lifetime: 1 year

2.2.3 Orbit Details

The orbit details of STUDSAT-2 are:

- Orbit: Sun synchronous polar orbit
- Altitude: 600 to 750 km
- Orbit Period: 98 minutes (approx.)
- Inclination: 98 degrees (approx.)
- Equatorial Crossing: 9:30-10:30 AM
- Eccentricity: 0
- Launcher: PSLV
- Satellite Separation Rate: 1 km per day
- ISL separation: Up to 100 km

2.2.4 Objectives of STUDSAT-2

The objectives of STUDSAT-2 are divided into three groups such as primary, technical and secondary objectives.

1. Primary Objectives

    - To prove the concept of establishing a communication link between the two satellites (Inter-Satellite Communication) and with the Ground Station.
    - To obtain a well-defined image of a certain region of the world using the CMOS camera and transmit to the Earth station using Twin satellites.
    - To prove the concept of temporal resolution, where the ability to collect imagery of the same area of the earth's surface at different periods of time and at different angles which is one of the most important elements for applying stereo imaging.

2. Technical Objectives

    - To demonstrate Inter-Satellite Communication between STUDSAT-2A and STUDSAT-2B via S-band frequency.
    - To capture the images of earth using a CMOS camera with a higher resolution of 83 metres and swath of 63.17 km x 40.32 km.
    - To monitor and control the sub-systems with the help of a 32-bit Reduced Instruction Set Computer (RISC) core controller ARM Cortex M4.

3. Secondary Objectives

    - To demonstrate a successful inter-university, inter-disciplinary and interdepartmental cooperation and coordination.
    - To develop leadership qualities and scientific temperament among students.
    - To promote space technology in educational institutions and encourage research and development in miniaturized satellites.

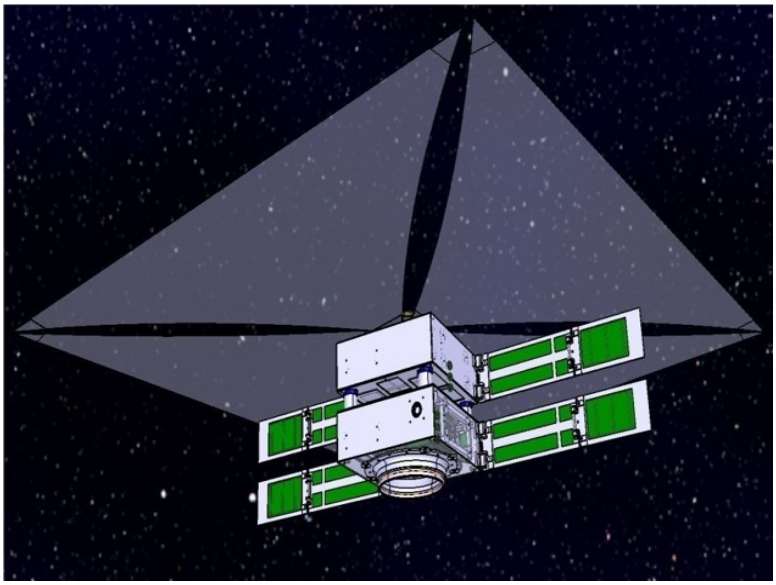


Figure 2.3: STUDSAT-2

The satellite payload includes an on-board GPS receiver, 3-axes attitude stabilization system, reaction wheels, efficient power system with deployable solar panels, full duplex communication system using UHF and S-bands, beacon in Morse code, antenna deployment and various other systems that are essential for a successful space mission. The two satellites will have capabilities to establish communication (Inter-Satellite Link) with each other and also the Master Control Unit (MCU) at ground segment. The satellites are in alongthe-track constellation architecture with the STUDSAT-2A sending position and velocity data to the STUDSAT-2B through Inter-Satellite Link. STUDSAT-2 Model is shown in Figure 2.3

2.3 On-Board Computer (OBC) Architecture of STUDSAT-2

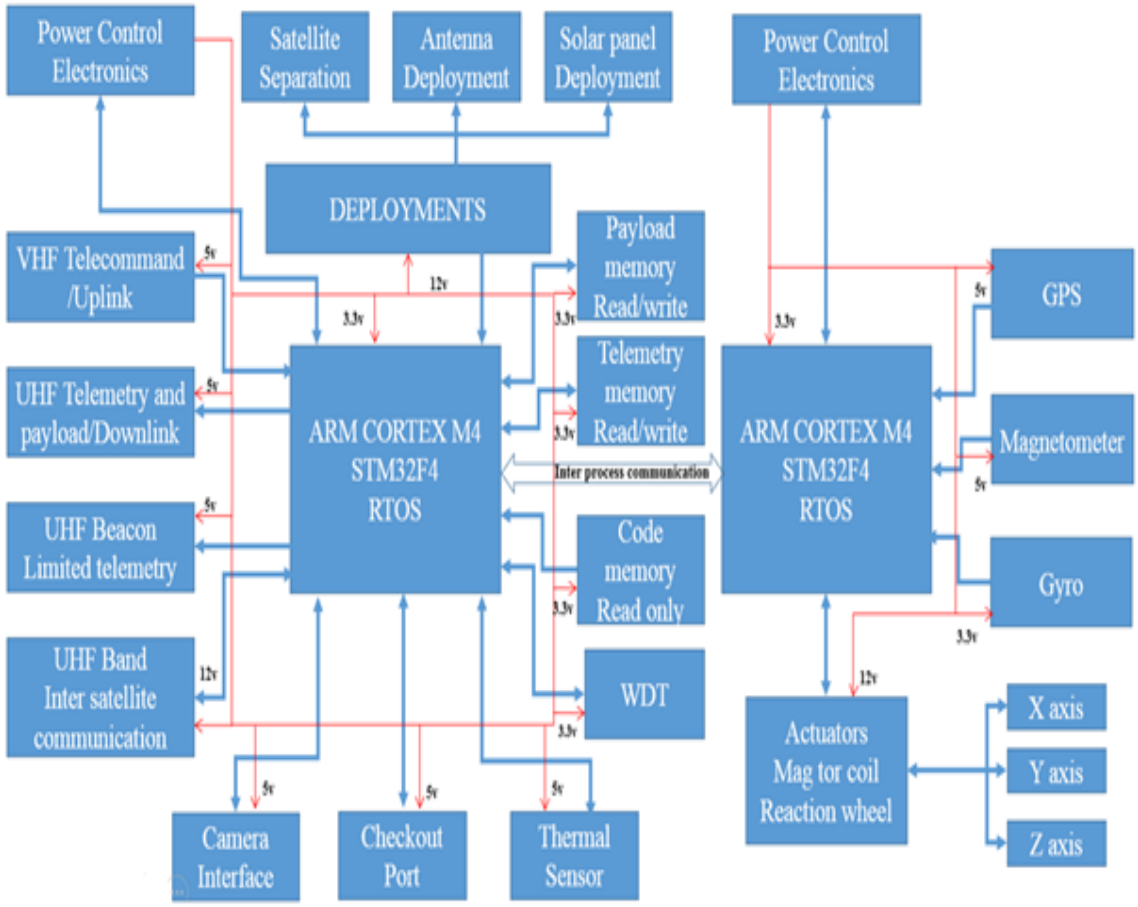


Figure 2.4: Architecture OBC STUDSAT-2

OBC considered for STUDSAT-2 is 32 bit ARM Cortex M4 based STM32F407VGT6 microcontroller which has built-in Digital Signal Processing (DSP) and Floating Point Unit (FPU) features. The system operates at 168 MHz and is strong enough to execute 210 Dhrystone Million Instructions per Second (DMIPS). The advantage of different low power modes like stop, standby, sleep with Real-Time Clock (RTC) enables the system to switch between these power modes. Figure 2.4 shows the OBC architecture of STUDSAT-2. The core element of the OBC subsystem is the dual identical microcontrollers. One monitors C&DH and the other takes care of ADCS of the satellite. Both the microcontrollers are intended to accomplish ADCS and C&DH tasks in parallel.

Chapter 3
Hardware and Software Description

3.1 Hardware

3.1.1 STM32F407VGT6 Discovery Board

The ARM Microcontroller STM32F407VGT6 Discovery will be the OBC. The STM32 F407VGT6 Discovery is a Cortex M4 series based 32-bit microcontroller. One of the major features of the ARM Cortex M4 series is the hardware acceleration of floating-point operations. On the STM32 devices, clocks are known as the hardware and peripheral clocks and are controlled by the Reset and Clock Control (RCC) group of registers.
Since there are more than 32 on-chip peripherals as shown in Figure 3.1, there are actually two registers used to switch on a clock: RCC AHB1ENR and RCC AHB2ENR for hardware clock and RCC APB1ENR and RCC APB2ENR for the peripheral clock. The clock is controlled by set/reset registers. So to turn ON a system, bit in the peripheral clock enable register should be set and to turn that same peripheral OFF, a bit in the corresponding RCC AHBxRSTR register should be set.

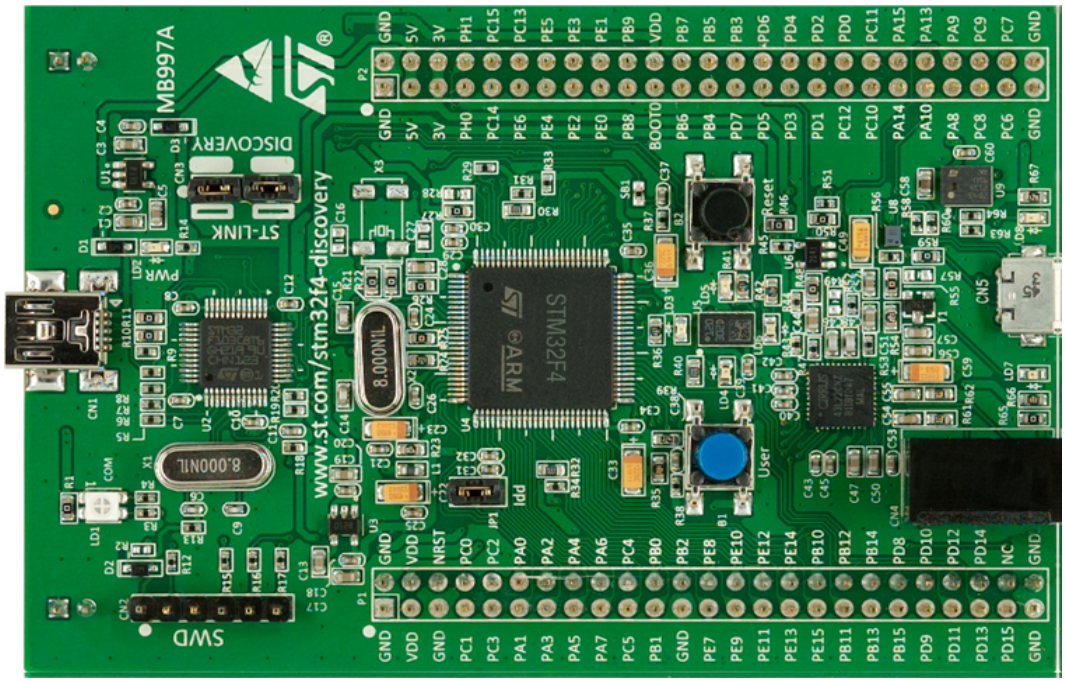


Figure 3.1: STM32F407VGT6 Discovery Board

Features of ARM Microcontroller:

The features of ARM Microcontroller are given below.

- STM32F407VGT6 High-Performance and DSP 32-bits MCU, 168 MHz Cortex-M4 CPU, 1 MB Flash, 192 KB RAM, 4 KB backup RAM.
- Advanced DMA channels, 6 USART, 3 SPI, 3 I2C, 3 ADC, 2 DAC, RTC, CRC unit, Ethernet, 2 USB OTG, two independent watchdogs.
- On-board ST-LINK/V2 with selection mode switch to use the kit as a standalone STLINK/ V2 (with SWD connector for programming and debugging).
- Board power supply: through USB bus or from an external 5 V supply voltage.
- External application power supply: 3.3 V and 5 V.
- On-board Micro-Electro-Mechanical Systems (MEMS) digital accelerometer.
- On-board MEMS digital microphone.
- On-board audio DAC with class-D speaker driver.
- USB OTG micro-AB connector.
- Eight LEDs:
  – Four user LEDs, LD3 (orange), LD4 (green), LD5 (red) and LD6 (blue).
- Four status LEDs:
  – LD1 (red/green) for USB communication.
  – LD2 (red) for 3.3 V power on.
  – 2 USB OTG LEDs - LD7 (green) VBus and LD8 (red) over current.

3.1.2 USB-UART Bridge

Universal Asynchronous Receiver Transmitter (UART) as shown in Figure 3.2 is used to connect computer to OBC for verification and testing purposes, offers a flexible means of data exchange with external equipment requiring an industry-standard NRZ asynchronous serial data format. The UART offers a very wide range of baud rates using a fractional baud rate generator. It supports asynchronous one-way communication and half-duplex single-wire communication. It allows multiprocessor communication. High-speed data communication is possible by using the DMA for multi-buffer configuration.

The main features of UART are:

- Half duplex, asynchronous communication.
- NRZ standard format (Mark/Space).
- Single-wire half-duplex communication.
- Configurable multi-buffer communication using Direct Memory Access (DMA).

Figure 3.2: UART to USB brigde

3.1.3 INA219 Current Sensor

INA219 is a high side DC current sensor breakout board capable of monitoring power consumption in a circuit. INA219 chip on the breakout board measures both the high side voltage and DC current draw over I2C with 1% precision. This chip as shown in Figure 3.3 is much smarter in a way that it can handle high side current measuring, up to +26 V DC, even though it is powered with 3 or 5 V. It will also report back that high side voltage, which is great for tracking battery life or solar panels.

Figure 3.3: INA219 current sensor

3.1.4 TMP75 Temperature Sensor

The TMP75 device from Figure 3.4 is digital temperature sensors ideal for Negative Thermal Co-efficient (NTC) and Positive Thermal Co-efficient (PTC) thermistor replacement. The devices offer a typical accuracy of $\pm 1°C$ without requiring calibration or External Component Signal conditioning. IC temperature sensors are highly linear and do not require complex calculations or look-up tables to derive the temperature. The on-chip 12-bit ADC offers resolutions down to 0.0625°C. The devices are available in the industry-standard LM75 SOIC-8 andMSOP-8 footprint. Multiple sensors are used and they will report back temperature fluctuations in a different part of satellites.

Figure 3.4: TMP75 temperature sensor

3.1.5 SDIO SDCard

SDIO may refer to Secure Digital Input Output, a type of Secure Digital card interface. So SD-Card reader as shown in Figure 3.5 with SDIO interface is used for recording all data of sensors.

Figure 3.5: SD Card

3.2 Software

3.2.1 STM32CubeMX & STMCubeIDE

STM32CubeMX and STM32CubeIDE are graphical tools and Integrated Development Tools that allows a very easy configuration of STM32 Microcontrollers and Micro-processors, as well as the generation of the corresponding initialization C code for the Arm Cortex-M core or a partial Linux Device Tree for the Arm Cortex-A core, through a step by step process. Figure 3.6 is the commercial icon of these tool chains. Free forum support, or optional commercial support and licensing available for STM32 tool chains.

Figure 3.6: STM32CubeMX and STM32CubeIDE

Four main aspects of CubeMX involve:

- Configuring GPIO port.
- Configuring peripherals.
- Configuring MiddleWares (FreeRTOS, FATFS).
- Configuring clocks in graphical interface.

3.2.2 HTerm

HTerm is a JS library that provides a terminal emulator program for the serial interface (COM), which runs on Windows and Linux. Figure 3.7 is the commercial symbol. It is reasonably fast, reasonably correct, and reasonably portable across browsers. It is useful debugging tool for serial communication applications.

Figure 3.7: HTerm

3.2.3 Keil μvision MDK v5

Keil MDK is the complete software development environment for a wide range of Arm Cortex-M based microcontroller devices. MDK includes the μVision IDE and debugger, a cross compiler Arm C/C++ compiler, and essential middleware components. Figure 3.8 is it's trademark icon.

Figure 3.8: Keil

3.2.4 FreeRTOS

FreeRTOS is a class of RTOS that is designed to be small enough to run on a microcontroller or microprocessor. A microcontroller or a microprocessor is a small and resource real-time constrained processor that incorporates, on a single chip, the processor itself, Read Only Memory (ROM or Flash) to hold the program to be executed, and the Random Access Memory (RAM) needed by the programs it executes. Typically the program is executed directly from the read only memory. Microcontrollers and microprocessor are used in deeply embedded applications (those applications where one never actually see the processors themselves or the software they are running) that normally have a very specific and dedicated job to do. The size constraints, and dedicated end application nature, rarely warrant the use of a full RTOS implementation - or indeed make the use of a full RTOS implementation possible. FreeRTOS™ wherein the commercial symbol is as shown in Figure 3.9 therefore provides the core real-time scheduling functionality, inter-task communication, timing and synchronization primitives only. This means it is more accurately described as a real-time kernel, or real-time executive. Additional functionality, such as a command console interface, or networking stacks, can be then be included with add-on components.

Figure 3.9: FreeRTOS

FreeRTOS features ideal for the project includes

- FreeRTOS demonstration builder core-preemptive, cooperative and hybrid configuration options.
- FreeRTOS-MPU supports the Arm Cortex-M3 Memory Protection Unit (MPU).
- Designed to be small, simple and easy to use. Typically a demonstration builder core binary image will be in the region of 4K to 9K bytes.
- Very portable code structure predominantly written in C.
- Supports both tasks and co-routines.
- Queues, binary semaphores, counting semaphores, recursive semaphores and mutexes for communication and synchronization between tasks, or between tasks and interrupts.
- Mutexes with priority inheritance.
- Supports efficient software timers.
- Powerful execution traces functionality.
- Stack overflows detection options.
- Pre-configured demo applications for selected single board computers allowing out of the box operation and fast learning curve.
- Free forum support, or optional commercial support and licensing.
- No software restriction on the number of tasks that can be created.
- No software restriction on the number of priorities that can be used.

Chapter 4
Methodolgy and Implementation

4.1 Methodology

4.1.1 Block Diagram and Schematic Diagrams

4.1.1.1 Block Diagram

The project is built around the STM32F407VG Discovery board. The six INA219 current sensors and four temperature sensors are interfaced to the board. The shunt current values and bus voltage values read by INA219 current sensors and, the temperature values read by TMP75 temperature sensors are sent to STM32F407VG Discovery board.This communication follows I^2C protocol. The values so received by the board is sent to the computer via UART bridge where it is displayed on serial monitor display. This helps for debugging purposes. The data read by the board is also stored in the SD card to achieve telemetry logging of the obtained sensor values. The block diagram of the project and used for experimentation is as shown in Figure 4.1.

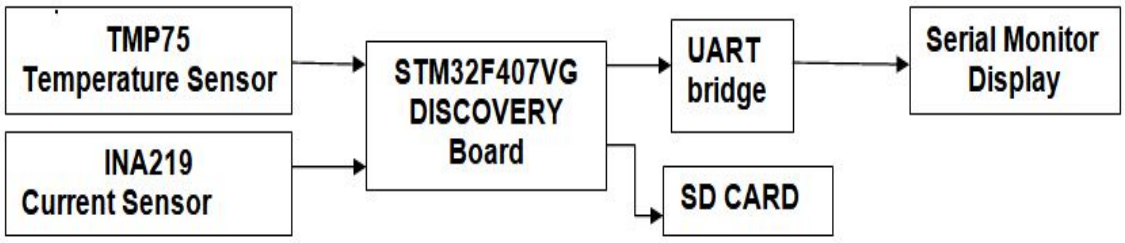


Figure 4.1: Block Diagram of the project

4.1.1.2 Schematic Diagrams of INA219 Current Sensor and TMP75 Temperature Sensor

The essential current,voltage,wattage ratings of the current sensor INA219 are studied from the datasheets and the pin diagram is compared with the each of the six current sensors in the current sensor PCB and slave addresses of each of the current sensors are noted down. The schematic of the six current sensors is as shown in Figure 4.2. Similarly, datasheet of the temperature sensor TMP75 is studied and the schematic of four temperature sensors is as shown in Figure 4.3.

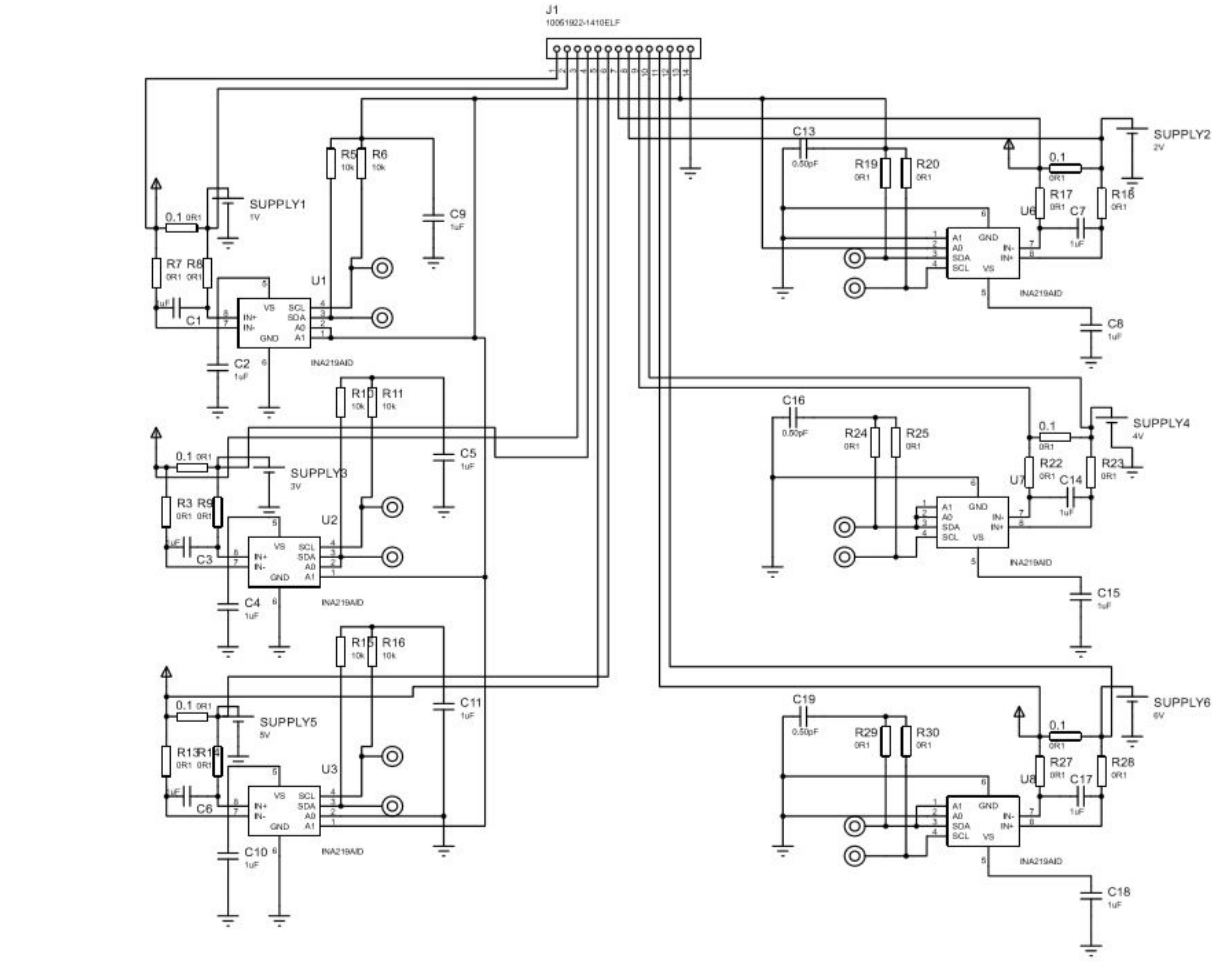


Figure 4.2: Schematic of 6 INA219 sensors

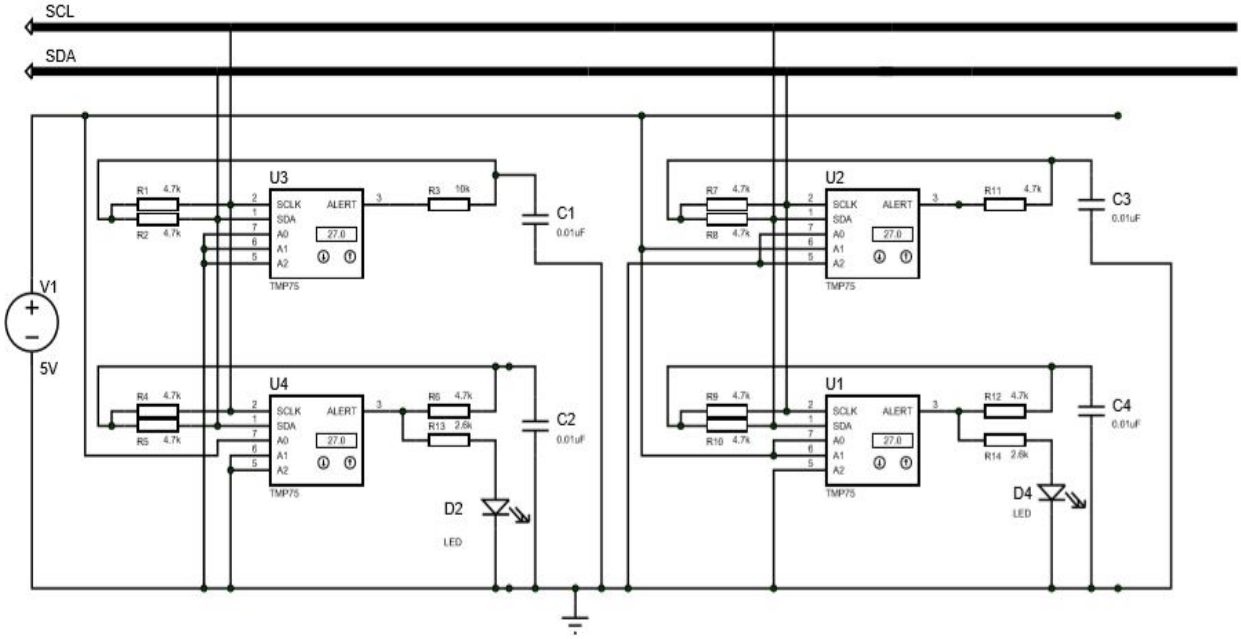
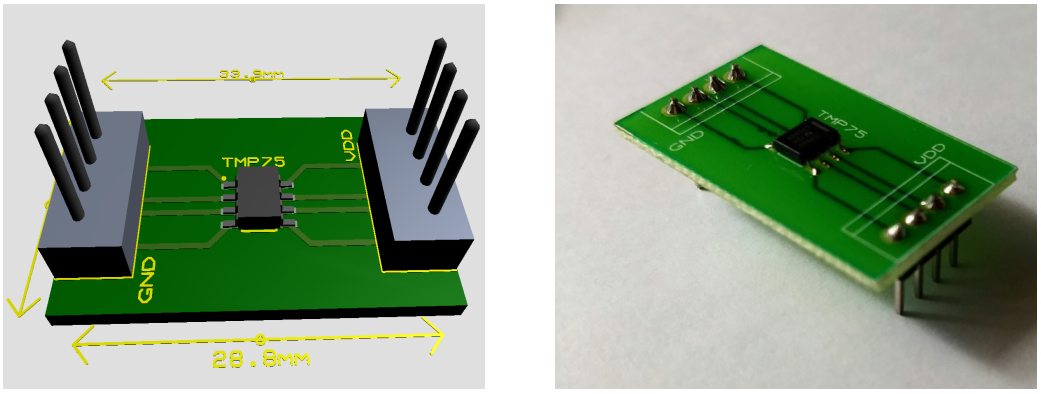


Figure 4.3: Schematic of 4 TMP75 sensors



(a) 3D PCB Proteus image (b) PCB Image

Figure 4.4: PCB of TMP75

4.2 Implementation

4.2.1 Hardware Set-up

The Figure 4.5 is the set-up of 6 DC Sources and connected to 6 INA-219 Current sensor modules interfaced to STM32 board with I2C interface and a serial UART module. Reading of voltage and current values begins after the delay set by call back functions.

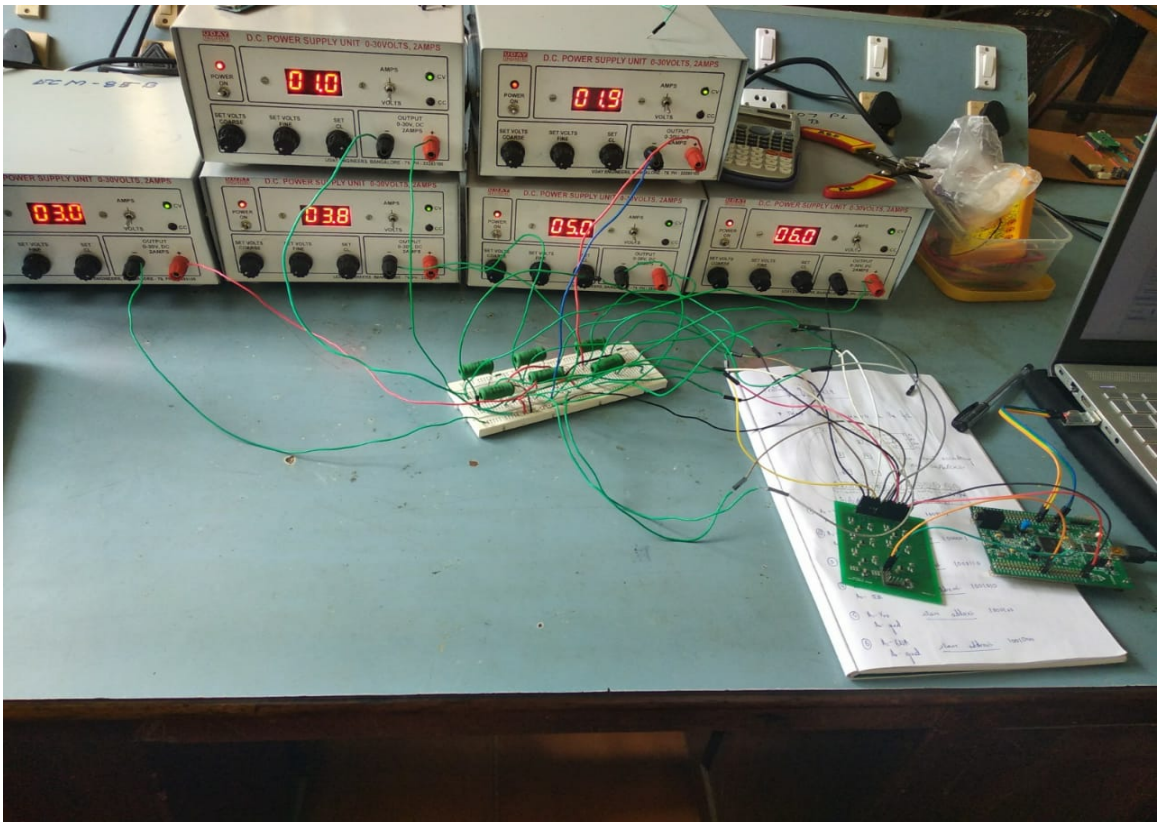


Figure 4.5: Hardware Set up of μ-Controller, INA219 Module and six DC Source

4.2.2 Flow chart

The first task is to use an inbuilt Real-time clock to generate a delay of 45 minutes. This is the duration where the satellite switches to normal mode, post ejection. The second task is to interface the four temperature sensor with microcontroller to collect temperature data. After confirming the attitude stability, an event-based separation of satellites is initiated.The third task is to interface six current sensors in the I2C protocol format to communicate with the microcontroller as master and collect the bus voltages and shunt current through the shunt resistor. The sensor vaues are continuously read from the sensors and are further used. The flow chart of the project is as depicted in Figure 4.6.

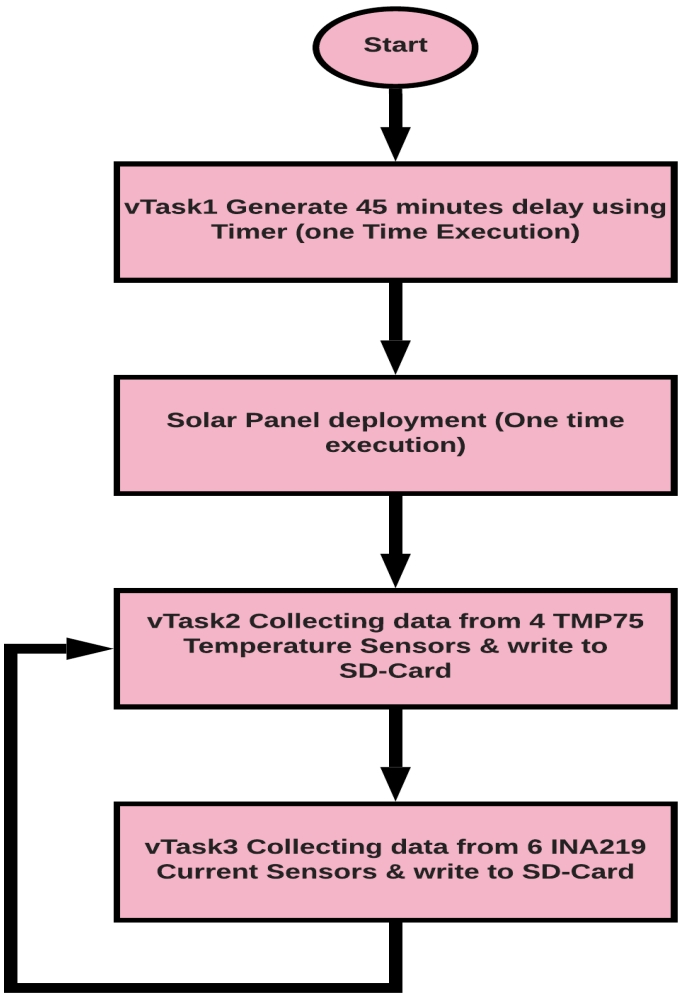


Figure 4.6: flowchart of the software showing how tasks are done

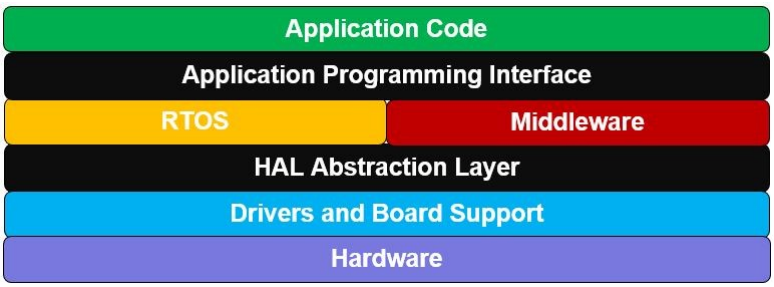


Figure 4.7: Software Abstraction layers

4.2.3 Procedure for implementation

To interface the four temperature sensors and six current sensors the pinouts, communication interfaces like SPI, I2C, UART, clock configuration is initialised using STMCubeMX. Along with this, middlewares like FATFS file systems, FreeRTOS, timers, semaphores are declared. The software has various abstraction level from API to Middleware and to HAL(Hardware Abstraction Layer) as shown in Figure 4.7. This generates a code with input-output pin declarations and empty threads declarations. Library files for current sensor and temperature sensor which constitutes of necessary function declaration and definitions are written. These library files are included in the main program and tasks are called into the empty threads. This is programmed using Keil μvision5 or STM32 Cube IDE. There are 11 threads: one default thread, six threads for six current sensors and four threads for four temperature sensors. The delay of 45 minutes is achieved using the timer inbuilt in STM32F407VG Discovery board as it takes 45 minutes switch on to the normal mode post ejection. During this duration, this task has priority aboveNormal hence, none of the other tasks run during this time. Post the delay, the four TMP75 temperature sensors read temperature of the surroundings and the six INA219 current sensors read shunt currents and bus voltages. The so read values are displayed on the computer using HTerm software using the UART protocol. The temperature sensor and current sensor data so collected are stored into an SD card and hence telemetry logging of data can be achieved and task analysis can be done using middlewares.

4.2.4 Calibration of INA219 sensors Interfacing

The INA219 current sensors are to be calibrated to obtain appropriate shunt current values. The calibration is done according to the following calculations: The maximum bus voltage for the current sensor is specified to be 16V. Also the shunt voltage across each sensor differential inputs is fixed to be 0.04V. The resistance of Rshunt =0.1Ω with 5W power rating is used for testing. Therefore the maximum

possible current is 0.4A.
Different cases corresponding to different gains. Cases 1,2,3 and 4 correspond to the gains of 1,2,4 and 8 respectively.
shunt VBUS_MAX=1V
R_SHUNT=0.1Ω

Case 1:

To obtain gain = 1

VSHUNT_MAX = 0.04V

1. Max possible current=$\frac{Vshunt\_Max}{R\_Shunt}$ = 0.4A

2. Max Expected current =0.4A

3. Possible range of LSBs (15 bit-12 bit)
=Max_Expected Current/2^{15} =0.0000122 –>12μA/bit
=Max_Expected Current/2^{12} =0.0000977 –>98μA/bit

4. Choosing 0.00005 50μA/bit as LSB

5. Calibration reg value =trunc($\frac{0.04096}{Current\_LSB*R\_Shunt}$) =8192(0x2000H)

6. Power_LSB =20*Current_LSB =0.001 1mW/bit

Case 2:

To obtain gain = 2

VSHUNT_MAX=0.08V

1. Max possible current =$\frac{Vshunt\_Max}{R\_Shunt}$ = 0.8A

2. Max Expected current =0.8A

3. Possible range of LSBs(15 bit-12 bit)
=Max_Expected Current/2^{15} =0.0000244 –>12μA/bit
=Max_Expected Current/2^{12} =0.0001953 –>98μA/bit

4. Choosing 0.00005 50μA/bit as LSB

5. Calibration reg value =trunc($\frac{0.04096}{Current\_LSB*R\_Shunt}$) =8192(0x2000H)

6. Power_LSB =20*Current_LSB =0.001 1mW/bit

Case 3:

To obtain gain = 4

VSHUNT_MAX=0.16V

1. Max possible current =$\frac{Vshunt\_Max}{R\_Shunt}$ = 1.6A

2. Max Expected current =1.6A

3. Possible range of LSBs(15 bit-12 bit)
=Max_Expected Current/2^{15} =0.0000488 –>12μA/bit
=Max_Expected Current/2^{12} =0.0003906 –>98μA/bit

4. Choosing 0.00005 50μA/bit as LSB

5. Calibration reg value =trunc($\frac{0.04096}{Current\_LSB*R\_Shunt}$) =8192(0x2000H)

6. Power_LSB =20*Current_LSB =0.001 1mW/bit

Case 4:

To obtain gain = 8

VSHUNT_MAX=0.32V

1. Max possible current = Vshunt_Max / R_Shunt = 3.2A

2. Max Expected current =3.2A

3. Possible range of LSBs(15 bit-12 bit)
=Max_Expected Current/2^{15} =0.0000976 –>12μA/bit
=Max_Expected Current/2^{12} =0.0007812 –>98μA/bit

4. Choosing 0.00005 50μA/bit as LSB

5. Calibration reg value =trunc($\frac{0.04096}{Current\_LSB*R\_Shunt}$) =8192(0x2000H)

6. Power_LSB =20*Current_LSB =0.001 1mW/bit

4.2.5 Calibration of TMP75 sensors Interfacing

The devices offer a typical accuracy of ± 1 °C without requiring calibration or external component signal conditioning. Device temperature sensors are highly linear and do not require complex calculations or look-up tables to derive the temperature. TMP75 modules have been split into 4 different boards and each mini-board with one TMP75 IC as shown in Figure 4.4.

Chapter 5
Testing and Results

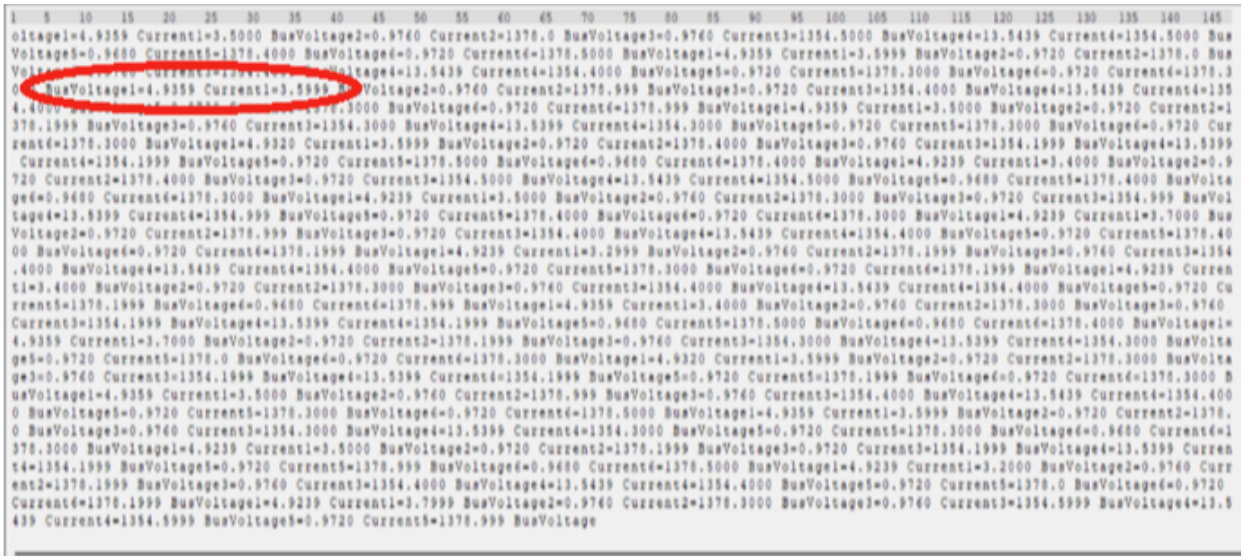


Figure 5.1: The result obtained from current sensor

As per the hardware and software described in the previous chapters, the ornamentation is carried out. This chapter discusses the tests and result obtained relevant to the C&DH section of the Satellite.

Image from Figure 5.1 shows the H-term Software's UART display of the different threads running and sends back the sensor data. This consists of INA219's voltage value as well as sensory values from six different locations. Multiple sensors corresponding to different threads can be read in similar manner. The red mark highlights one of the sensory data from a INA219 Current sensor. The bus voltage supply is 5V and load resistance is of 10KΩ. The bus voltage is 4.9359V and the shunt current is 3.5999x10^{-4}A.

Chapter 6
Conclusion and Future Scope

6.1 Conclusion

The part of the work involved abstract layering of Software using FreeRTOS for controlling the attributes of C&DH subsection of the Satellite. Timer with and without RTC, calibration and testing of single INA219 current sensor as well as integration of six INA219 current sensors using FreeRTOS are successfully demonstrated. Its been observed that Timer with RTC will provide more lag than timer without RTC so the integration of the system was using a timer without RTC. Displaying of current sensor values on serial monitor has been successfully observed. Integration of timer and six INA219 sensor modules with FreeRTOS where threads corresponding to current sensors are blocked until timer reaches given time limit. TMP75 IC Interface software implementation and Hardware designing have been completed. Threads linked to temperature sensor are scheduled along with threads linked to current sensor successfully. Final integration along with middle wares like FreeRTOS, FatFs file-system was successful using STM32CubeIDE.

6.2 Future Scope

In future, the plan is to increase the efficiency of the system. Further analysis of the RTOS system using licensed software like Percepio's Tracealyzer or free software SEGGER's SystemView can boost the software significantly. The log file recording to an SD-card requires one to add FatFs in middleware which would make the binary file to cross 32 KB limit. This limit could be problematic if the IDE used to integrate rest of the modules with C&DH module is limited version of Keil MDK. Instead shifitng the entire software design to free software STM32CubeIDE would save the time and would increase the efficiency. And using SDIO interface, SD-Card to record all the physical values would provide a better insight to the entire system for the later analysis of the health and condition of the satellite. Theoretically, prediction of the variations in physical values prior to the occurring using this collected data would be instrumental for the maintenance of the satellite system.

Bibliography

[1] C. Angadi, Z. Manjiyani, C. Dixit, K. Vigneswaran, G. S. Avinash, Prithvi Raj Narendra, S. Prasad, H. Ramavaram, R. M. Mamatha, G. Karthik, H. V. Arpan, A. H. Sharath, P. Sashi Kiran, and K. Visweswaran, "Studsat: India's first student pico-satellite project," in *2011 Aerospace Conference*, 2011, pp. 1–15.

[2] S. R. Hegde, D. Sahay, S. Sandya, G. M. Sandeep, Muralidhara, and K. V. Nikhilesh, "Design and development of inter-satellite separation mechanism for twin nano satellite — studsat-2," in *2016 IEEE Aerospace Conference*, 2016, pp. 1–8.

[3] J. Slačka and M. Halás, "Safety critical rtos for space satellites," in *2015 20th International Conference on Process Control (PC)*, 2015, pp. 250–254.

[4] C. M. Cooke, "Implementation of a real-time operating system on a small satellite platform," Ph.D. dissertation, Colorado Space Grant Consortium, Boulder, CO, 2012.

[5] A. Hanafi, M. Karim, I. Latachi, T. Rachidi, S. Dahbi, and S. Zouggar, "Fpga-based secondary on-board computer system for low-earth-orbit nanosatellite," in *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, 2017, pp. 1–6.

[6] B. Rajulu, S. Dasiga, and N. R. Iyer, "Open source rtos implementation for onboard computer (obc) in studsat-2," in *2014 IEEE Aerospace Conference*, 2014, pp. 1–13.

[7] C. Nagarajan, K. Kinger, F. Haider, and R. Agarwal, "Performance analysis of micrium rtos in the computer of a nano-satellite," in *2015 IEEE Aerospace Conference*, 2015, pp. 1–9.

[8] K. Lamichhane, M. Kiran, T. Kannan, D. Sahay, H. G. Ranjith, and S. Sandya, "Embedded rtos implementation for twin nano-satellite studsat-2," in *2015 IEEE Metrology for Aerospace (MetroAeroSpace)*, 2015, pp. 541–546.

[9] A. Putra, T. Priyambodo, and N. Mestika, "Real-time operating system implementation on obc/obdh for ugmsat-1 sequence," vol. 1755, 07 2016, p. 170009.

[10] A. Tomar. (2014) Coocox:user guide for coocox coos operating system. [Online]. Available: https://www.element14.com/community/docs/DOC-42435/l/coocoxuser-guide-forcoocox-coos-operating-system