Development of Satellite Subsystems for Use in a Bench Satellite

Samuel Jeffery
20029004

Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering (Electronic) in the Faculty of Engineering at Stellenbosch University.

Supervisors: Dr H. W. Jordaan and Dr L. Visagie
Department of Electrical and Electronic Engineering

November 2022

UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Plagiaatverklaring / *Plagiarism Declaration*

1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.
   Plagiarism is the use of ideas, material and other intellectual property of another's work and to present is as my own.

2. Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.
   I agree that plagiarism is a punishable offence because it constitutes theft.

3. Ek verstaan ook dat direkte vertalings plagiaat is.
   I also understand that direct translations are plagiarism.

4. Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.
   Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism

5. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.
   I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

20029004 Studentenommer / *Student number*	Handtekening / *Signature*
S. J. Jeffery Voorletters en van / *Initials and surname*	**21/11/2022** Datum / *Date*

Abstract

This thesis addresses the need for a satellite-like system to bridge the gap between satellite development and satellite flight missions. The solution takes the form of a bench satellite, BenchSat, that can emulate satellite capabilities and interactions in a controlled environment on the ground. BenchSat utilises surrounding facilities, particularly an air bearing facility and an existing ground station (TIM Ground Station), to produce a realistic test environment for satellite development.

Through research and literature review, the form and subsystems for BenchSat were selected to best suit the requirements of the Electronic Systems Laboratory. During BenchSat's system-level design three subsystems were identified and defined, namely an electrical power subsystem, a radio communication subsystem, and an onboard computer subsystem. Each subsystem is designed to include a microcontroller and accept external power so that it can function on its own.

The subsystems were designed and implemented separately with each subsystem completing a different requirement for BenchSat. Firstly, the electrical power subsystem delivers power to the BenchSat satellite bus, additional payloads, and the carrier cart that moves around the air bearing facility. Secondly, the radio communication system connects to the TIM Ground Station to allow experiments with BenchSat to be controlled remotely from an active ground station as they would be during a real flight mission. Finally, the onboard computer subsystem fulfills multiple tasks including controlling the air bearing carrier cart movement, monitoring movement, and controlling experiments and demonstrations.

Ground station software and hardware were developed on the TIM Ground Station to communicate with BenchSat and a second ground station was built within the Electronic Systems Laboratory to provide an easier route to testing.

Finally, a fully integrated BenchSat connected to the air bearing carrier cart and moving around the air bearing facility was demonstrated. During this demonstration, multiple actions were executed and BenchSat was in constant connection with the TIM Ground Station. The demonstrations showed that BenchSat met the system requirements to a high level.

Uittreksel

Hierdie tesis spreek aan die behoefte vir ’n satellietagtige stelsel aan om die gaping tussen satellietontwikkeling en satellietvlug missies te oorbrug. Die oplossing neem die vorm aan van ’n banksatelliet, BenchSat, wat satelliet vermoëns en interaksies kan emuleer in ’n beheerde omgewing op die grond. BenchSat gebruik die beskikbaarde fasiliteite, veral ’n luglaerfasiliteit en ’n bestaande grondstasie (TIM Grondstasie), om ’n realistiese toetsomgewing vir satellietontwikkeling te produseer.

Deur navorsing en literatuurstudie is die vorm en substelsels vir BenchSat gekies wat die beste te pas by die vereistes van die Elektroniese Stelsels Laboratorium. Tydens BenchSat se stelselvlakontwerp is drie subsisteme geïdentifiseer en gedefinieer; naamlik ’n elektriese kragsubstelsel, ’n radiokommunikasiesubstelsel en ’n aanboordrekenaarsubstelsel. Elke substelsel is ontwerp met ’n mikrobeheerder ingesluit en aanvaar eksterne krag sodat dit op sy eie kan funksioneer.

Die substelsels was afsonderlik ontwerp en geïmplementeer, met elke substelsel wat ’n ander vereiste vir BenchSat voltooi. Eerstens die elektriese kragsubstelsel lewer krag aan die BenchSat-satellietbus, bykomende vragte en die draerwa wat om die luglaerfasiliteit beweeg. Tweedens, die radiokommunikasiestelsel koppel aan die TIM Grondstasie om toe te laat eksperimente met BenchSat wat op afstand vanaf ’n aktiewe grondstasie beheer moet word soos dit tydens ’n regte vlugmissie sou wees. Uiteindelik die aanboordrekenaarsubstelsel vervul die veelvuldige take, insluitend die beheer van die beweging van die luglaer-karretjie, monitering van beweging, en beheer van eksperimente en demonstrasies.

Grondstasie sagteware en hardeware was op die TIM Grondstasie ontwikkel om met BenchSat en ’n tweede grondstasie te kommunikeer was binne die Elektroniese Stelsels Laboratorium gebou om ’n makliker roete vir toetsing aan te bied.

Laastens, ’n volledige integreerde BenchSat wat aan die luglaer-karretjie gekoppel is en om die luglaerfasiliteit beweeg, was gedemonstreer. Tydens hierdie demonstrasie is verskeie aksies uitgevoer en BenchSat was in konstante verbinding met die TIM Grondstasie. Die demonstrasies het aangedui dat BenchSat tot ’n hoë vlak die stelselvereistes voldoen het.

Acknowledgements

I would like to thank:

- My supervisors, Dr Willem Jordaan and Dr Lourens Visagie, for their guidance and advice throughout this project.
- CubeSpace for funding me in this endeavor.
- My parents Doug and Jean Jeffery for their unending love and support.
- Daniella Copeman for keeping me sane.
- The other members of the ESL for making the last two years worthwhile.
- The staff at the E&E Department, especially Jenny Martin, Wessel Croukamp and Johan Arendse.

Contents

List of Figures

List of Tables

Acronyms and abbreviations

ADC	Analog-to-Digital Converter
ADCS	Attitude Determination and Control System
C&DH	Command and Data Handling
CAN	Controller Area Network
COTS	Commercial Off-The-Shelf
DET	Direct Energy Transfer
EPS	Electrical Power System
ESL	Electronic Systems Laboratory
ESOC	European Space Operating Centre
FCDPP	FUNcube Dongle Pro Plus
FCS	Frame Check Sequence
FSK	Frequency Shift Keying
GUI	Graphical User Interface
GPIO	General Purpose Input Output
GPOS	General Purpose Operating System
I2C	Inter-Integrated Circuit
MCU	MicroController Unit
MPPT	Maximum Power Point Tracking
NBFM	Narrow Band Frequency Modulation
OBC	Onboard Computer
PCB	Printed Circuit Board
PV	Photovoltaic
RCS	Radio Communication System
RF	Radio Frequency
RSSI	Received Signal Strength Indicator
RTOS	Real-Time Operation System
SDR	Software-Defined Radio

SPI	Serial Peripheral Interface
TT&C	Telemetry, Tracking and Command
UART	Universal Asynchronous Receiver-Transmitter
UHF	Ultra High Frequency
VHF	Very High Frequency

Chapter 1

Introduction

1.1. Background

Successful satellite component designs need to move from a stage where they are developed in a controlled environment on the ground to where they can be launched and serve a purpose onboard a satellite. However, there is a large gap between an experimental design and a component that can be flown onboard a satellite. This project aims to narrow that gap by providing a system that can emulate satellite capabilities and interactions in a controlled environment on the ground.

The Electronic Systems Laboratory (ESL), part of the Electrical and Electronic Engineering Department at Stellenbosch University, is aiming to design and launch a satellite mission, entitled DockSat. DockSat involves the launch of two CubeSats together as one unit. The mission would require the satellites to separate and reattach using a novel docking system [1].

There is a need for a platform that can assist the DockSat mission and others like it. Two ways in which a platform could assist DockSat and other missions have been identified:

1. Begin the development process in the ESL for a satellite system that would form the base for a future CubeSat.
2. Create a satellite-like base that components (such as a docking mechanism) can attach to for development and testing.

This platform would take the form of an ESL designed bench satellite, to be named BenchSat. In this report, the term bench satellite refers to a satellite system that is used in a laboratory environment. A bench satellite would allow for the testing and development of a wide variety of satellite systems, including hardware and software, with no financial risk [2].

Bench satellites such as EyasSat have been used in the past to allow researchers to develop systems that work in a satellite-like environment [2]. An example of this is the attitude determination and control subsystem (ADCS) software developed on EyasSat in 2014 [3]. An ESL bench satellite is preferable to a commercial bench satellite because it will be less expensive and it can be designed to interface with existing facilities.

Satellite bus components will be designed for BenchSat to be the building blocks of a new satellite. The satellite bus will be made up of an electrical power subsystem (EPS), a radio communication subsystem (RCS) and an onboard computer subsystem (OBC). These three subsystems will make up the satellite bus and allow for future components and payloads to be attached.

BenchSat will add to existing facilities to extend their current functionality. These facilities include an air bearing facility and a functional ground station known as the TIM Ground Station. The air bearing facility is used to simulate orbital dynamics on the ground and the TIM Ground Station is equipped with powerful, movable antennas that have been used in the past to track active missions and satellites.

1.2. Context

BenchSat will connect with the Stellenbosch University infrastructure including the air bearing facility and the TIM Ground Station.

1.2.1. Air Bearing Facility

Figure 1.1: Air Bearing Carrier Cart on the Air Bearing Facility Glass Platform

The air bearing facility enables almost frictionless movement by an air bearing carrier cart on a glass platform. Due to this low friction movement the carrier cart can emulate orbital spacecraft dynamics with three degrees of freedom - two translational and one rotational [4]. This movement allows for the demonstration of many different satellite procedures, such as formation flying or debris capture [4]. The carrier cart actuates around the glass platform using thrusters that are controlled by solenoids on the carrier cart.

BenchSat will be placed on the carrier cart during experiments. During experiments, BenchSat will need to control and monitor the carrier cart's movement around the air bearing facility. To control the movement BenchSat will need to provide power to the solenoids and send command signals to activate the thrusters.

1.2.2. TIM Ground Station

Figure 1.2: TIM Ground Station Antenna Array

The TIM Ground Station communicates over VHF and UHF channels. It uses a Kenwood TS2000 transmitting radio to send messages on the VHF band and it receives messages on the UHF band with a FunCube Dongle Pro Plus [5].

BenchSat needs to communicate with the TIM Ground Station during experiments. With this connection, the TIM Ground Station can control and monitor experiments remotely, as if the experiment were taking place in space.

1.3. Motivation

The motivation for this project is to facilitate satellite research. BenchSat will provide a space-like environment for satellite researchers to develop projects in space-related fields. It will equip researchers with the ability to integrate their projects with existing hardware and enable realistic simulations and testing. The low-risk nature of the simulated environment will provide a perfect place to develop systems and algorithms. Newly designed systems that are not ready for the risks and costs involved in a flight test can be tested and developed using BenchSat. In this way, BenchSat will help to bridge the gap between systems that are in development and those that are space-ready.

In the ESL, projects are given to individual students and therefore are often focused on a single task. These projects will benefit from having the option to integrate into a prebuilt bench satellite that includes general systems. Furthermore, the projects can rely on BenchSat to provide basic functionality. In some cases this functionality may remove the need to design power and computing systems, allowing students to focus on work that relates to their project. Researchers working with algorithms running on a computer can use BenchSat's processor to test if their algorithms will work on a satellite microcontroller.

BenchSat will be similar in size, processing power, power distribution, and communication method to a 1U CubeSat. Although the functionality of BenchSat is similar to that of a flight satellite, it does not have the associated cost, time or risk. Furthermore, since BenchSat will not have a flight test the components will not need to be radiation hardened for space, the spatial constraints can be relaxed to simplify the design, and components will not need redundant backups as they can be replaced if they break. These factors will make BenchSat a valuable testing tool and show that processes developed on BenchSat will be relevant to satellite research.

Leveraging existing Stellenbosch University facilities will enable BenchSat to create a more realistic testing environment than is currently available. The TIM Ground Station and the air bearing facility mentioned in Section 1.2 are useful tools that can be used to test satellite components. If a component like BenchSat can interface with the TIM Ground Station and the air bearing facility at the same time to create a connected system, the benefits of each tool will be compounded. This connected system would enable system integration testing. A mission can be executed in a realistic environment using the air bearing facility while being controlled by satellite-like components on BenchSat and being monitored by the TIM Ground Station as it would monitor a flight test in space.

Although the environment created by BenchSat is not a perfect representation of a flight test, it will assist with the development of satellite projects within the ESL. BenchSat will make use of existing infrastructure at Stellenbosch University thus creating an environment that is as close to what would be experienced in orbit as possible. Furthermore, it will allow for more in-depth testing than would be possible in a simulation or with components functioning on their own.

1.4. Problem Definition

Problem Statement: This project addresses the need for a satellite-like system to assist with the development of satellite projects within the ESL. This assistance includes developing basic functionality (power, communication and computing) and creating an interface with existing infrastructure.

High-Level Goal: The overall goal of this project is to create a bench satellite that can be used by members of the ESL to create satellite-related systems.

Secondary Goal: The system should leverage existing infrastructure at Stellenbosch University such as the air bearing facility and the TIM Ground Station.

1.5. Project Objectives and Scope

There is a need for a satellite-like system to assist with the development of satellite projects within the ESL. This assistance includes developing basic functionality (power, communication and computing) and creating an interface with existing infrastructure.

This project aims to meet that need through the design, construction and practical implementation of a bench satellite system, referred to as BenchSat, to be used in conjunction with facilities currently available at Stellenbosch University. The scope of this project is focused on producing a bench satellite that is capable of:

1. Controlling experiments and storing gathered data.
2. Providing power for itself, additional payloads and the air bearing carrier cart.
3. Connecting to the TIM Ground Station.
4. Controlling the air bearing carrier cart.

In order to show that the objectives have been achieved a demonstration of BenchSat's functionality will be executed using the air bearing facility and the TIM Ground Station.

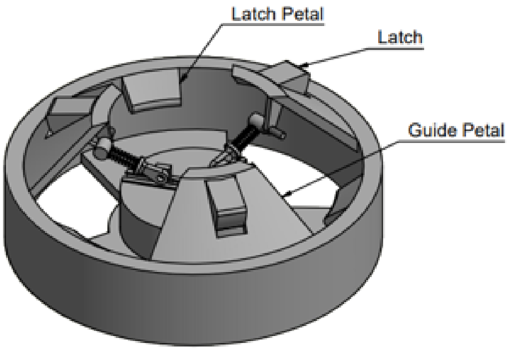
This demonstration will show the capabilities listed above. During the demonstration BenchSat will move through multiple states while recording movement data. It will power itself and the carrier cart during the demonstration without any wires attached. The demonstration will be controlled by the TIM Ground Station and a consistent connection will be managed. Finally, BenchSat will demonstrate control by directing the carrier cart to move in selected ways.

1.6. Use Cases

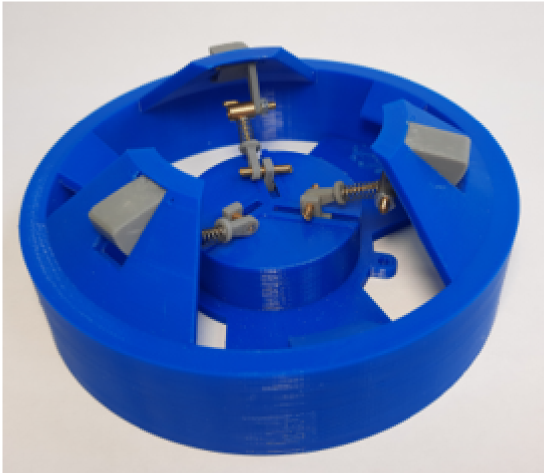
Two use cases for BenchSat have been identified as examples that demonstrate how BenchSat would benefit a new satellite system.

The first example expands on the DockSat mission mentioned in the background section. DockSat aims to perform multiple undocking and redocking of a 1U and 2U satellite in orbit [1]. For this mission, a novel docking mechanism will need to be developed and tested. Figure 1.3 shows an initial design for a docking mechanism to be used on DockSat.

Using BenchSat and the air bearing facility, a realistic docking procedure can be executed



(a) Labeled 3D Drawing of Docking Mechanism



(b) 3D Printed Docking Mechanism

Figure 1.3: Prototype DockSat Docking Mechanism

to test the novel docking mechanism. BenchSat enables the docking mechanism to be tested in a low friction environment without needing other DockSat components. BenchSat will include interfaces to control the air bearing carrier cart that would not be useful to include on DockSat. The BenchSat onboard computer will be able to process docking control systems or additional systems can be integrated onto BenchSat. Using BenchSat in this way streamlines the development process since integral components like the docking mechanism can start to be tested before the satellite bus is complete.

A second use case is the development of an ADCS component such as a reaction wheel. A reaction wheel uses momentum to control the rotation of a satellite about a particular axis. A reaction wheel would be attached to BenchSat to receive power and control signals. The air bearing facility then enables experiments where the rotation of the air bearing carrier cart could be controlled by the reaction wheel. During such experiments, BenchSat would control the air bearing carrier cart actuators, supply power to the reaction wheel, provide input signals to the reaction wheel, and record experimental data. This assistance would enable a developer to focus on design issues that are relevant to the reaction wheel and not the surrounding systems.

1.7. Thesis Outline

The following is the outline of the work put forward in this thesis.

Chapter 2 is a literature study of other bench satellite systems. The study focuses on the problems addressed by bench satellites and the outcomes achieved from using them.

Chapter 3 is the system-level design of BenchSat. The chapter shows the overall system architecture and defines the interfaces between each subsystem. A functional description covers the correct functionality of BenchSat and will introduce the key points to test the system.

Chapter 4 covers the background, design, implementation and testing of the BenchSat EPS.

Chapter 5 covers the background, design, implementation and testing of the BenchSat RCS.

Chapter 6 covers the background, design, implementation and testing of the BenchSat OBC.

Chapter 7 discusses the ground station interface. The chapter covers both the software and hardware used on the ground station and the methods used to implement full-duplex communication.

Chapter 8 details system integration and the final demonstrations.

Chapter 9 concludes the project by providing an overview of the executed work, drawing conclusions from the outcomes of the project and suggesting recommendations for future work.

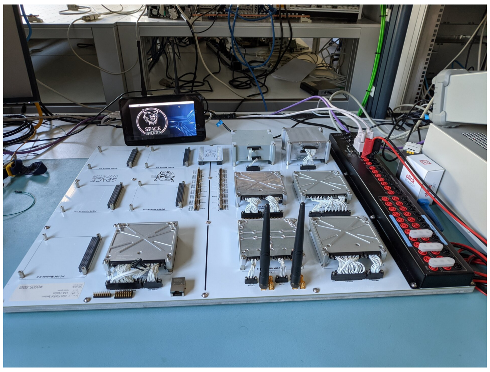
Chapter 2

Literature Study

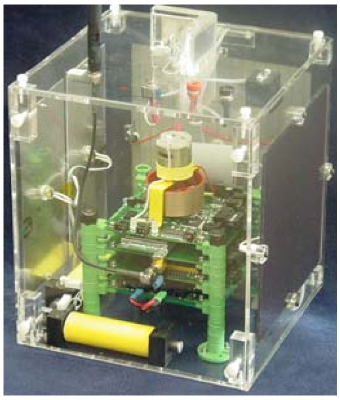
This chapter is a literature study that discusses other bench satellite systems. Six example bench satellites are introduced and reviewed to provide insight into the design of BenchSat. This process includes grouping bench satellites into types, identifying common problems that are dealt with and common methods that are used.

2.1. Types of Bench Satellites

The term bench satellite covers a wide variety of satellite systems that can vary depending on their application. FlatSats are a subgroup of bench satellites that have a distributed structure. Components in a FlatSat are not stacked but rather connected by a shared motherboard that distributes power and connects communication channels [6]. A FlatSat makes it possible to operate a full satellite while retaining the flexibility to swap systems in and out and try different setups. The distributed layout allows a user to have access to individual components that may not be reachable in a flight configuration [7]. This access is particularly useful for debugging and measurement purposes. EIRFLAT-1, the EMM FlatSat and the OPS-SAT FlatSat mentioned in Sections 2.2.4, 2.2.5 and 2.2.6 respectively are all examples of FlatSats.



(a) Flatsat: European Space Agency FlatSat [6]



(b) Satellite Model: EyasSat [2]

Figure 2.1: Types of Test Satellites

Other bench satellites that do not have a FlatSat form factor will be referred to as satellite models. Satellite models often have the same goals as a FlatSat but they still have the structure of a normal satellite. EyasSat, Pujllay and HEPTA-Sat mentioned in Sections 2.2.1, 2.2.2 and 2.2.3 respectively would all be categorised as satellite models. While FlatSats are often built to emulate or test a specific satellite, satellite models tend to be standalone systems that are used for educational or development purposes but not for flight.

2.2. Examples of Bench Satellites

The six bench satellites that were reviewed are briefly discussed in this section.

2.2.1. EyasSat

EyasSat was developed by the United States Air Force Academy to be satellite hardware that could be used to educate students about satellite engineering. The goal of the project was to separate satellite development from clean rooms and allow students to work with a satellite in the classroom without any financial risk [2].

Figure 2.2: Integrated EyasSat [8]

The project had two main specifications for EyasSat. Firstly, it should be an affordable and sustainable microsatellite for use in a laboratory to instruct students on space systems engineering. Secondly, it should contain the “six traditional satellite subsystems” as well as provide space for additional subsystems such as propulsion and experimental payloads. The six traditional subsystems are structural, electrical power, data handling, communications, attitude determination and control, and thermal subsystems [9].

2.2.2. Pujllay

Pujllay is a CubeSat engineering model developed by students from the National University of San Martin in Argentina for entry into a CubeSat design competition [10].

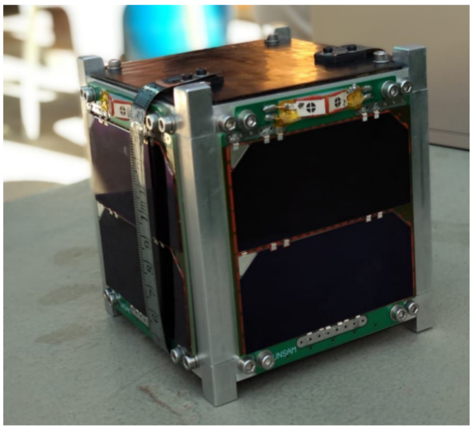


Figure 2.3: Pujllay CubeSat Model Assembled [10]

Pujllay was not designed to be immediately ready for space but rather the team planned to add and replace parts to make it space-ready in the future. The subsystems included in Pujllay are structure, communication, power, command and data handling, thermal control, and attitude determination. The fully assembled Pujllay CubeSat is shown in Figure 2.3.

2.2.3. HEPTA-Sat

HEPTA-Sat (Hands-on Education Program for Technical Advancement Satellite) is a low cost small satellite that is used for education. HEPTA-Sat, shown in Figure 2.4, consists of three boards including a sensor board, an EPS and communications board, and a command and data handling board. The boards are physically connected with spacers and connected electrically via a pin socket [11].

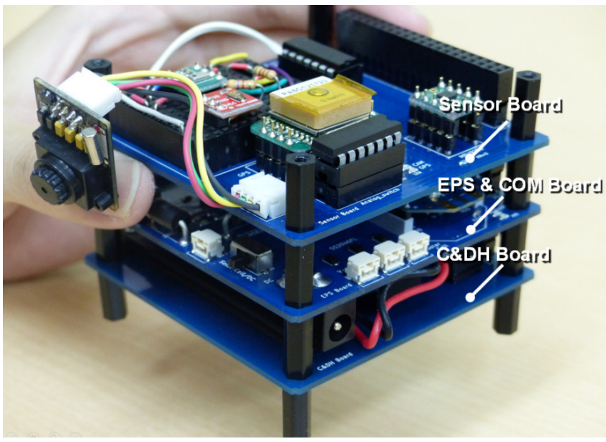


Figure 2.4: HEPTA-Sat with Labelled Subsystems [11]

HEPTA-Sat is used as an educational tool in a training program. The program focuses on assembling, integrating and testing the satellite to enable a user to experience the development process of a nano-satellite in a short time.

2.2.4. EIRFLAT-1

EIRFLAT-1 is a FlatSat that was designed to test the Educational Irish Research Satellite (EIRSAT-1). It aimed to ensure compatibility between all of the spacecraft's components and to allow pre-assembly system-level testing [7]. EIRFLAT-1 is made up of electrical interface boards, which connect satellite subsystems when they are laid out in a flat configuration. As can be seen in Figure 2.5, each active system is connected to an area on an electrical interface board. The electrical interface board shares power, ground and communication connections to all of the active systems and the distributed layout makes it easier for a user to test each system.

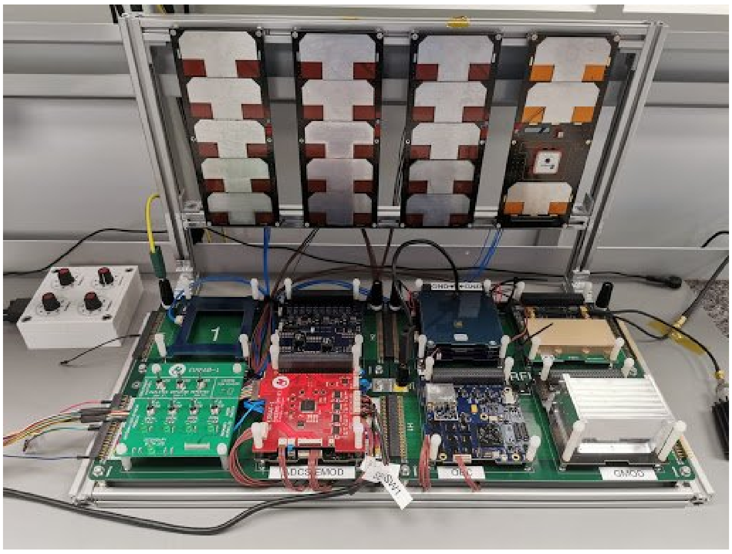


Figure 2.5: EIRFLAT with an Integrated Satellite [7]

EIRFLAT-1 was successfully used to test both EIRSAT-1 and other CubeSats. The system enabled satellite components to be tested in isolation, as part of a system, and at the overall system level before they are launched [7].

2.2.5. Emirates Mars Mission

The Emirates Mars Mission (EMM) is an interplanetary spacecraft mission that was launched in 2020. The mission was to study the Martian atmosphere for one Martian year [12].

To better test this satellite the EMM FlatSat was built. The EMM FlatSat's purpose was to provide a flight-like representation of the EMM spacecraft. The FlatSat's objectives included spacecraft integration, development of procedures and software, testing and anomaly investigation [13].

The FlatSat is made up of three types of components: engineering models that represent satellite components with similar or identical hardware, ground support equipment to configure the simulation environment and emulate the spacecraft interfaces, and simulation models that run on the ground support equipment. The EMM FlatSat is a highly accurate representation of the EMM spacecraft and it enabled rigorous testing and development of the systems that were used [13].

2.2.6. OPS-SAT

OPS-SAT is a 3U CubeSat that is currently in orbit having been launched on 18 December 2019. It is reconfigurable to enable experiments to be run onboard and new software can be uploaded to it from a ground station [14]. OPS-SAT is used in conjunction with a FlatSat copy of OPS-SAT laid out in a flat configuration at the European Space Operations Centre (ESOC) [14]. All code that is tested on OPS-SAT was first tested on the OPS-SAT FlatSat for safety reasons and so that issues could be corrected before the more complex test.

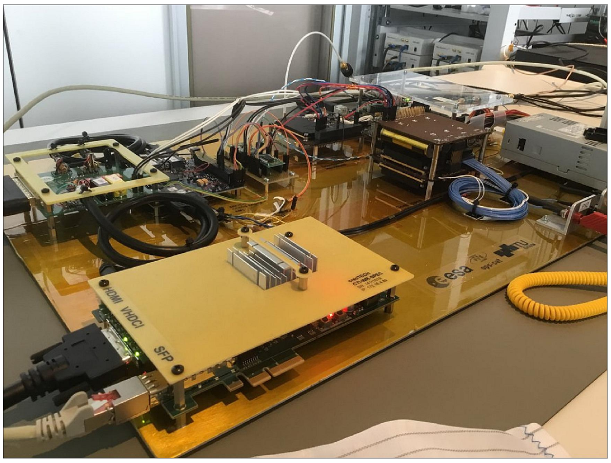


Figure 2.6: OPS-SAT FlatSat at ESOC [15]

Examples of work that has been tested on the OPS-SAT FlatSat include a set of image processing algorithms developed for BeaverCube-2 [16] and autonomous orbit control and satellite status monitoring algorithms developed at KTH in Sweden [17]. OPS-SAT enabled rapid prototyping, testing and validation for both of these projects at no cost to the experimenters.

2.3. Bench Satellite Outcomes

The common goal among all the reviewed bench satellites is that they increase the accessibility of satellite systems by providing a low risk environment in which to use and test satellite systems. The different systems reviewed had different reasons for achieving this goal. The main reasons that were identified are education, testing and development.

Education in this context relates to teaching students of all skill levels about satellite systems. Bench satellites that are designed to be robust and produced in large quantities like EyasSat or HEPTA-Sat are ideal for this application. Bench satellites provide students with a realistic and tangible satellite experience without the financial risk of using expensive space-ready components.

FlatSats tend to focus on testing applications. EIRFLAT-1 and the EMM FlatSat are examples of satellites that are mainly focused on testing a specific spacecraft. They are designed alongside a target satellite and used to iron out issues that arise during development. Such FlatSats are usually made available earlier on in the project, before engineering and flight models of the satellite are integrated, to allow software development to proceed in parallel with the satellite bus development.

Since the development of satellite systems is a broad concept, bench satellites can aid development in multiple ways. OPS-SAT and its FlatSat are existing satellites that outside users can use to develop software in a realistic environment. New subsystems can be added to EyasSat and developed using EyasSat's other subsystems such as the electrical power subsystem or the flight computer. A bench satellite like Pujllay was designed to enter a competition, but it is a bench satellite that can be improved and updated to become a real satellite.

2.4. Subsystems Used

The subsystems present on the bench satellites reviewed in this chapter were recorded and the results are shown in Table 2.1.

	System	**EyasSat**	**Pujllay**	**HEPTA**	**EIRFLAT**	**EMM**	**OPS**
1	Command	x	x	x	x	x	x
2	Communication	x	x	x	x	x	x
3	Power	x	x	x	x	x	x
4	ADCS	x	x		x	x	x
5	Structure	x	x	x		x	
6	Thermal	x	x			x	

Table 2.1: Traditional Satellite Systems Used by Reviewed Bench Satellites

The subsystems listed in the table are the six traditional subsystems identified in [9]. The most common subsystems are command, communications, power and ADCS. BenchSat will need to implement the functionality of all four of these subsystems. A structural subsystem was also present in all three satellite models, which demonstrates the need for extra support in a stacked satellite.

2.5. Summary

This chapter reviewed six bench satellite systems that have been used in multiple applications. The systems were categorised based on their type and their function. A summary of these results is shown in Table 2.2. Relevant subsystems were identified based on the reviewed systems.

Bench Satellite	FlatSat	Sat Model	Education	Testing	Development
EyasSat		x	x		x
Pujllay		x			x
HEPTA-Sat		x	x		
EIRFLAT-1	x			x	x
EMM	x			x	
OPS-SAT	x			x	x

Table 2.2: Summary of Reviewed Bench Satellite Systems

The outcomes of this chapter will inform the designs both for BenchSat as a whole and for its subsystems. The next chapter will cover the system-level design of BenchSat.

Chapter 3

System-Level Design

This chapter describes BenchSat's system-level design. It includes the desired system characteristics, a definition and explanation of the subsystems included on BenchSat, the system architecture including the internal architecture and the interfaces to other systems, and a functional description of how the system will be used during experiments.

3.1. Desired System Characteristics

BenchSat aims to fulfill the objectives defined in Section 1.5. The desired characteristics of BenchSat are listed below:

1. BenchSat should have an onboard computer that has enough processing power to facilitate experiments, control payloads and store experimental data as per point 1 of the project objectives defined in Section 1.5.
2. BenchSat should include power and communication subsystems as included in all bench satellites reviewed in Section 2.2.
3. BenchSat should power the satellite bus, the air bearing carrier cart and any attached payloads as per point 2 of the project objectives defined in Section 1.5.
4. BenchSat should be able to provide switched power to attached payloads with protection and monitoring to reduce the chances of damaging components and provide feedback during payload testing.
5. BenchSat should be able to communicate over VHF (uplink) and UHF (downlink) channels to a ground station. These channels are selected to interface with the TIM Ground Station capabilities defined in Section 1.2.2, as per point 3 of the project objectives defined in Section 1.5.
6. BenchSat should control the actuators on the air bearing carrier cart to enable movement on the air bearing facility's glass platform as per point 4 of the project objectives defined in Section 1.5.

7. BenchSat should have a CubeSat form factor due to its relationship to DockSat and to fit onto the air bearing carrier cart.

8. Subsystems should be implemented as separate modules, similar to satellites like EyasSat and HEPTA-Sat to increase modularity.

9. Each subsystem on BenchSat should be able to function on its own (if provided with regulated power) to enable standalone subsystem testing.

10. Subsystems should be connected by a shared bus network to allow internal communication.

By achieving the characteristics listed above, BenchSat will be able to achieve the objectives of this project.

3.2. Subsystem Definition and Explanation

Six traditional CubeSat subsystems were identified in Section 2.4. These subsystems are command, communication, power, ADCS, structure and thermal.

The BenchSat bus consists of three main subsystems that execute the functions required of the satellite. The subsystems include an onboard computer subsystem (OBC) as the command subsystem, an electrical power subsystem (EPS) as the power subsystem and a radio communication subsystem (RCS) as the communication subsystem. The EPS will provide power to the BenchSat satellite bus, additional payloads and the air bearing carrier cart. The RCS facilitates communication with the TIM Ground Station over a radio frequency communication link. The OBC will perform multiple roles, including controlling experiments, controlling air bearing carrier cart movement, and providing connections and processing power to be available for additional payloads.

There is no dedicated ADCS subsystem on BenchSat, however, ADCS functionality is encapsulated in the OBC which controls movement when BenchSat is connected to the air bearing carrier cart. Standoffs connect the subsystems to form a minimal structural subsystem while still allowing access to each subsystem. No thermal subsystem will be added.

Clear boundaries are set between the subsystems to encourage modularity and enable the development of subsystems in isolation. The benefits of modularity are:

1. It allows substitutions for improvements and upgrades.

2. It simplifies the process of adding new components.

3. It reduces the requirements on the OBC processor by distributing computing.

4. It enables standalone subsystem testing.

3.3. System Architecture

Figure 3.1 illustrates BenchSat's overall system architecture. The three different types of systems involved in the architecture are indicated by colour. The light blue blocks are the subsystems that form BenchSat's internal architecture. These subsystems are mechanically connected within BenchSat. The dark blue blocks are external components that are within this project's scope. The red blocks are components that are relevant to BenchSat's design but are outside of the scope of this project.

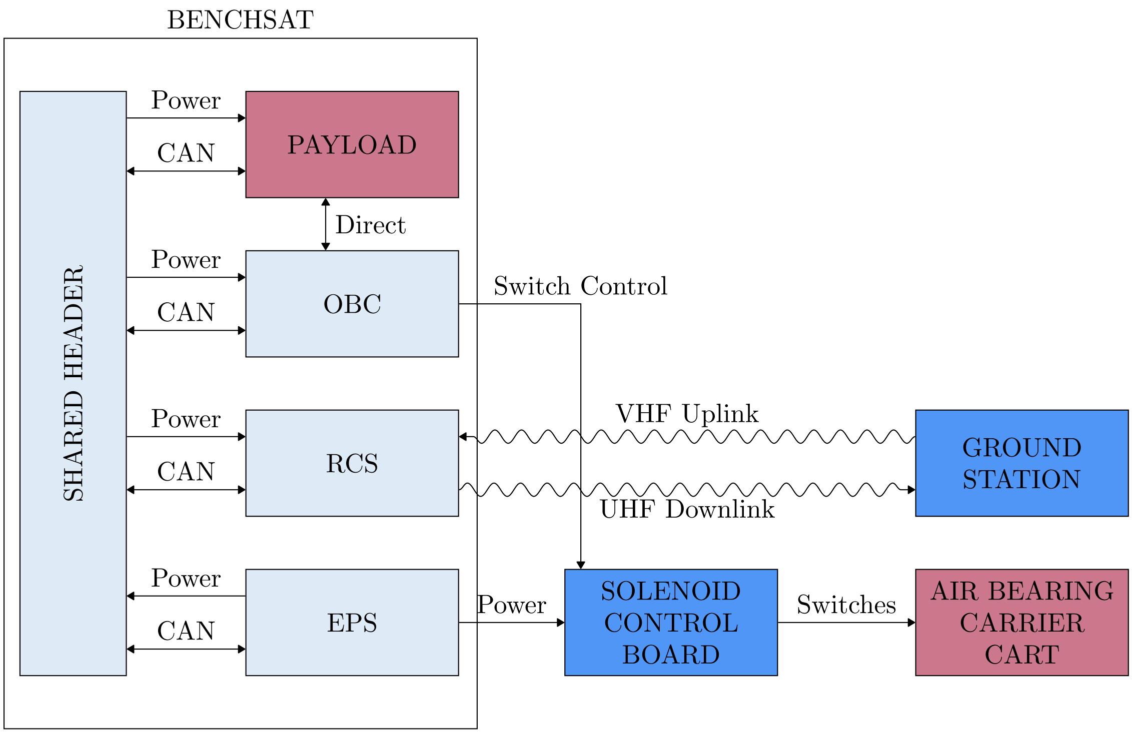


Figure 3.1: BenchSat System Architecture Diagram including Surrounding Systems

3.3.1. Internal Architecture

BenchSat has a distributed architecture where each of the three main subsystems (OBC, RCS and EPS) functions autonomously. The subsystems are electrically connected by a shared header that distributes power and connects each subsystem to a shared communication bus. A structural subsystem supports the connections so that the header is not under mechanical stress.

Each subsystem is connected to a Controller Area Network (CAN) communication bus. CAN is a robust network that can allow multiple devices to communicate on the same bus [18]. CAN is also a well-known communication bus that has been used on satellites in the past [19].

Power is distributed through the shared header from the EPS to the other subsystems. All power distributed through the shared header is monitored and controlled by the EPS. Unswitched system power is shared with the other BenchSat subsystems so that they are always active. Switched power channels are shared with attached payloads and are switched on and off by the EPS. Power usage data and commands to switch different channels on and off are communicated to the EPS controller via the shared communication bus.

3.3.2. BenchSat Interfaces

The interfaces depicted in Figure 3.1 are the connection to a payload, the connection to the ground station and the connection to the air bearing carrier cart via the solenoid control board.

Ideally, a payload will attach to BenchSat via the shared header and add to the physical structure of BenchSat. Through the shared header the payload will have access to switched power from the EPS and the shared communication bus. A payload that is connected in this way will function as an additional subsystem on BenchSat.

To enable diverse payloads, the OBC has the option to connect directly to a payload. This direct connection means that a payload does not have to be able to communicate on the shared communication bus and can connect to the OBC via a different means of communication. This option is to accommodate simpler payloads that may need to be entirely controlled by the OBC.

The TIM Ground Station is connected to BenchSat via a radio frequency connection between the ground station and the BenchSat RCS. The uplink to BenchSat is a VHF band transmission and the downlink to the TIM Ground Station is a UHF band transmission. The channels are separated from each other so they do not cause interference. The connection is maintained during testing and messages can be sent in either direction at any time. A ground station controller will be developed to work with BenchSat. The user interface, radio signal generation, and communication protocols of the ground station controller fall within the scope of this project.

BenchSat connects to the air bearing carrier cart through the solenoid control board. Battery power is connected directly from the EPS to the solenoid control board. This power is used to drive the solenoids on the air bearing carrier cart. The solenoid control board has four switches that are controlled by the OBC. The OBC has four output pins that are connected to each of these switches. Each switch connects to one of the four solenoids on the air bearing carrier cart. The air bearing carrier cart was designed in a previous project and its functionality is not within the scope of this project.

3.4. Functional Description

3.4.1. Primary Experimental Configuration

Figure 3.2 shows the interactions between BenchSat and the surrounding facilities during an experiment using the air bearing facility and the TIM Ground Station. When BenchSat and the air bearing carrier cart are functioning together they will be referred to as the compound satellite.

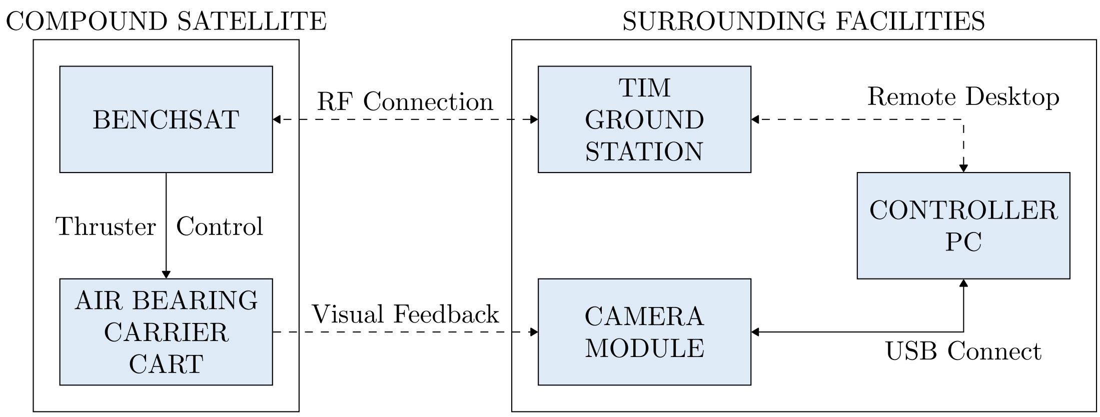
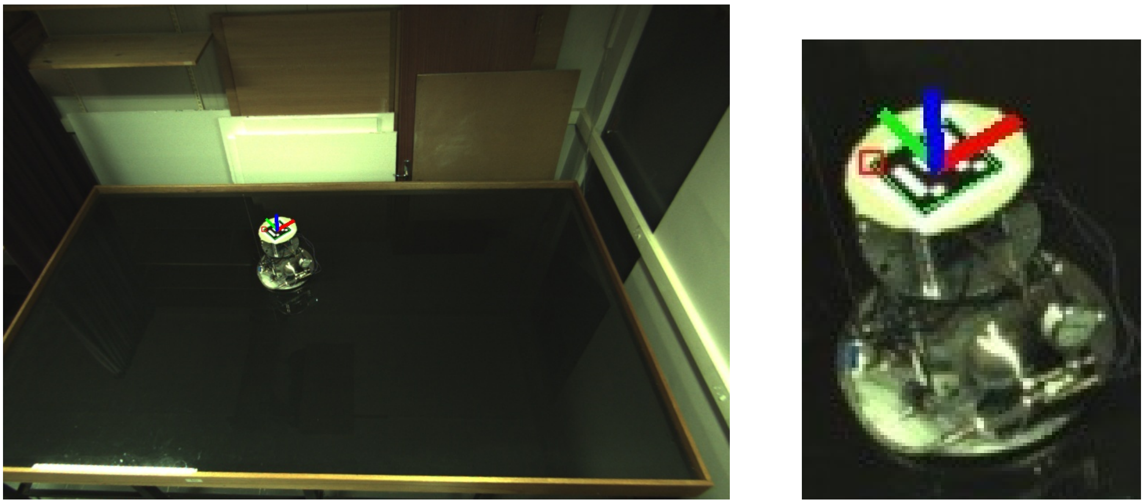


Figure 3.2: Interaction between BenchSat and Surrounding Facilities

As discussed in Section 3.3, the TIM Ground Station creates a radio frequency connection to the compound satellite that can be used to send commands and request information. The TIM Ground Station is controlled remotely by a controller PC, which is the user interface for experiments in this configuration. A user manages experiments from the controller PC by sending commands to the compound satellite via the TIM Ground Station. This flow of command is the same as it would be if the compound satellite were on a flight test in space.

Experiments using the compound satellite are focused on its movement around the air bearing facility glass platform. BenchSat controls the compound satellite with the air bearing carrier cart thrusters. Feedback from this movement can be recorded by an overhead camera module. The camera module connects directly to the controller PC and provides images of the air bearing facility. The controller PC identifies an aruco marker on the compound satellite in the image feedback for pose estimation. The poses are recorded for later review. An example of the image feedback is shown in Figure 3.3a. Figure 3.3b is a zoomed-in perspective of the air bearing carrier cart in Figure 3.3a to show a visual representation of the pose estimation.

Data from an experiment is recorded in three main ways. Data can be recorded in permanent storage on the OBC, it can be transmitted back to the TIM Ground Station or it can be captured by the camera module. Other possible methods would be based on the payload being tested.



(a) Image Feedback from Camera Module **(b)** Pose Estimation Results

Figure 3.3: Camera Module Feedback

3.4.2. Secondary Experimental Configuration

A second ground station is situated in the ESL. Because of its proximity to the air bearing facility and because it is dedicated to ESL projects this second ground station helps to simplify testing. This ground station is known as the ESL Ground Station. By using the ESL Ground Station, the TIM Ground Sation can be bypassed, as shown in Figure 3.4. The ESL Ground Station will be expanded upon in Chapter 7. The benefit of the ESL Ground Station is that it gives more control to a user. If the TIM Ground Station is not able to be used because the ground station is busy tracking a satellite, there is load shedding or any other issue, the ESL Ground Station can still be used to run experiments and develop systems.

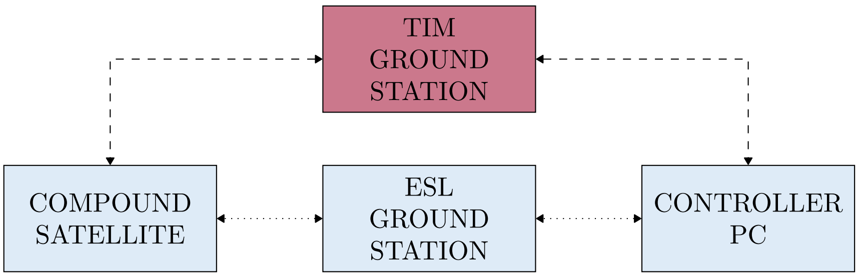


Figure 3.4: Ground Station Bypass Connection

3.4.3. Physical Experimental Configuration

Figure 3.5 shows a sketch of the layout of the different systems that connect to BenchSat. The TIM Ground Station is physically separated from the rest of the components in a different building. The other systems are located around the air bearing facility to simplify experiments that involve multiple systems. The controller PC is connected to the TIM Ground Station via a remote desktop connection to allow real-time control.

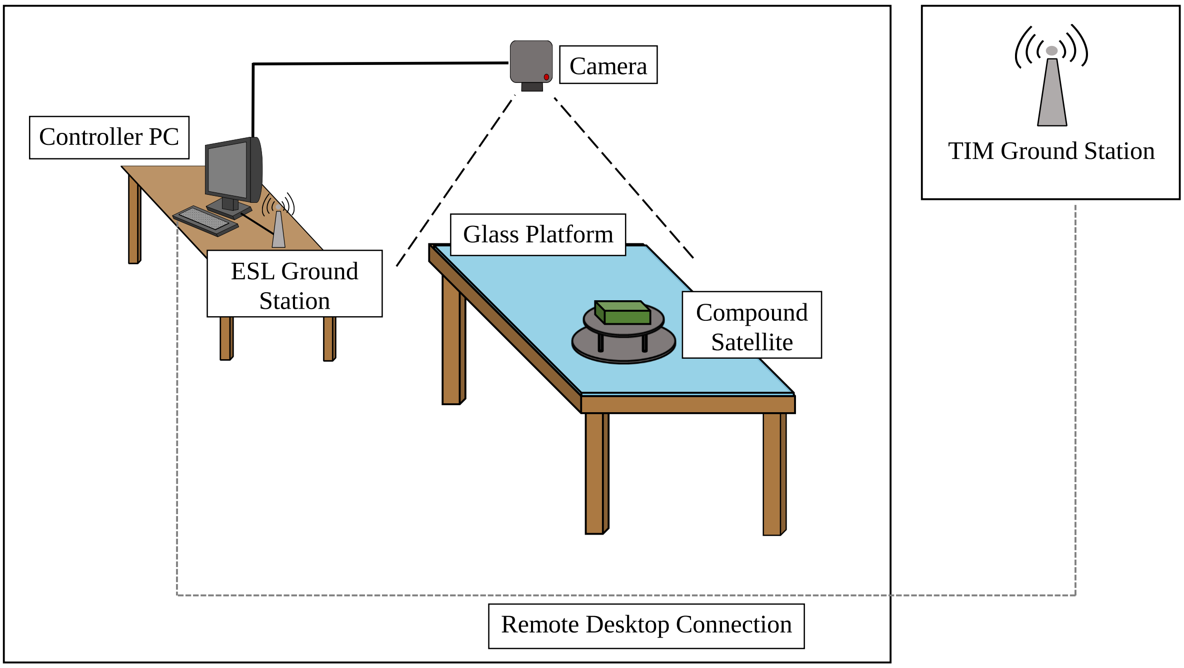


Figure 3.5: Sketch of Experiment Layout

3.5. Summary

This chapter has laid out the overall designs for BenchSat. The desired system characteristics were stated with reasons. The OBC, the RCS and the EPS were chosen to be BenchSat's three primary subsystems while all six traditional subsystems were discussed. The system architecture discussed BenchSat's internal power and CAN connections as well as the external interfaces to the TIM Ground Station, the air bearing carrier cart and attached payloads. Lastly, the functional description described the standard experimental configuration.

The next chapter will cover the EPS design, implementation and testing processes. The EPS is the first of the three BenchSat subsystems to be discussed.

Chapter 4

Electrical Power Subsystem

This chapter covers the background, design, implementation and testing of the BenchSat Electrical Power Subsystem (EPS).

4.1. Background

4.1.1. The Functions of an EPS on a Satellite

The EPS on a satellite is the subsystem that controls power generation, power conversion, energy storage and power distribution [20]. The interactions between these functions on an EPS are shown in Figure 4.1.

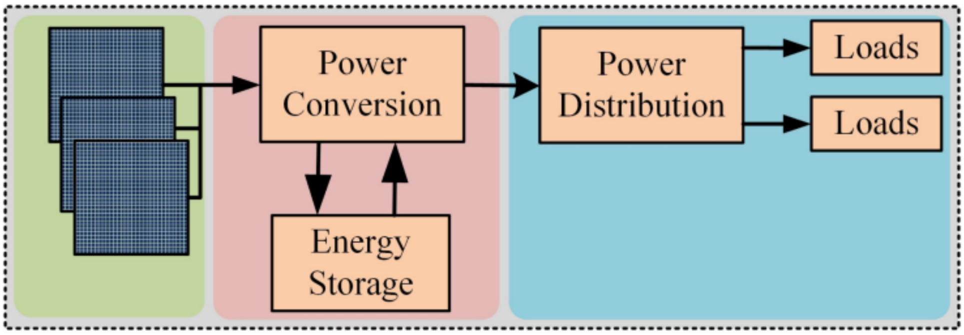


Figure 4.1: Interactions between Components of a CubeSat EPS [20]

The most common method of power generation on a satellite is photovoltaic (PV) panels that are placed on the outside of the satellite or folded and deployed after launch. These panels provide the primary source of power for a satellite during its lifetime. Since part of a satellite's orbit will inevitably be in an eclipse, where PV panels will not generate power, the spacecraft must also have some form of energy storage. Energy storage is used when the PV panels are not active or when power demands exceed the power that the PV panels can deliver. Currently, the most commonly used storage system is lithium-ion batteries. Other battery chemistries have been used in the past and more advanced technologies are being investigated [21].

The power that is generated by the PV panels or drawn from storage must be converted before it can be used by the other subsystems on a CubeSat. The method of drawing power from these sources is complex as the power conversion system must control the incoming power from the PV panels, decide whether to draw from or charge the energy storage system and regulate the power going out to loads at a consistent voltage.

Once the power has been converted into a usable form the EPS distributes it to attached loads. The distribution can be shared along voltage rails to attached systems or switches can be used to manage which components receive power. The need for additional monitoring varies between systems and depends on the application.

4.1.2. EPS Architecture Classifications

From [20], EPS architecture can be generally classified by a few characteristics, namely conversion stages, converter location, and interface with PV panels. These classifications are shown in Figure 4.2. They define the structure of how the EPS handles the conversion step between the energy sources and the loads. The following sections discuss the benefits and drawbacks of the mentioned characteristics.

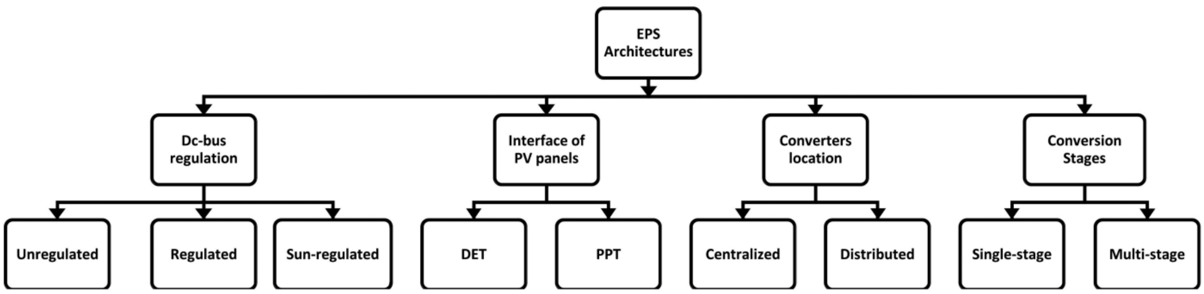


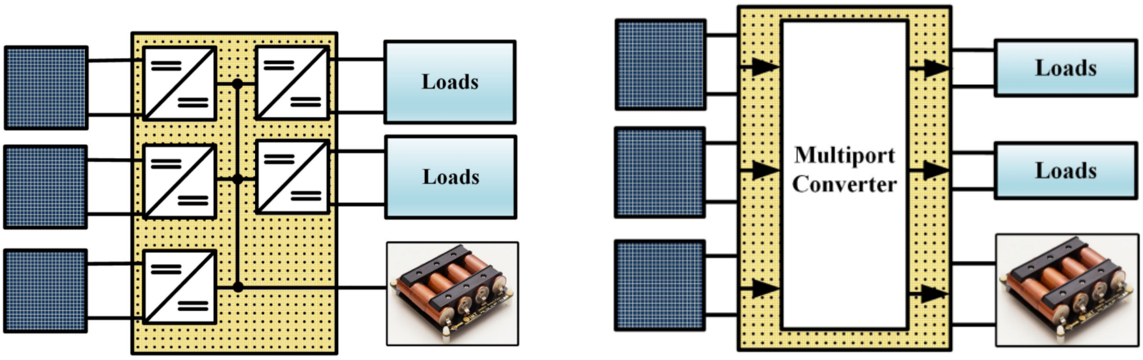
Figure 4.2: EPS Architecture Classification [20]

Conversion Stages

The power conversion step can be done by a single multiple-input multiple-output converter in a single-stage configuration or by many single-input single-output converters in a multi-stage application.

Single-stage power conversion, shown in Figure 4.3b, requires a single converter to perform multiple actions including power conversion, power regulation and power distribution. This single-stage option has a lower component count, can be more efficient with losses localised on one device, and has a small footprint. However, the single-stage option increases the system complexity and the requirements of a single device. A single-stage configuration has not yet been used in CubeSat applications but it has been implemented in smaller satellites with single-cell battery systems and on exploratory rovers [20].

The more commonly used option is a multi-stage power conversion application, shown in Figure 4.3a, with multiple converters each dedicated to a single action. This configuration requires more space and decreases efficiency but it is practical to build and can be adapted to meet a wide range of system needs.



(a) Multiple Stage Power Conversion **(b)** Single Stage Power Conversion

Figure 4.3: EPS Conversion Stage Options [20]

Converter Location

Converter location refers to where on a satellite the output regulators are placed and how power is distributed throughout a system.

A centralised architecture consists of placing all power regulators in one area (such as on a dedicated EPS) and then providing the other subsystems with regulated power. The conversion stage options shown in Figures 4.3a and 4.3b are both examples of centralised architectures.

In a distributed architecture the EPS handles the solar panels and battery inputs and then provides the rest of the system with a given bus voltage. Figure 4.4 shows an example of a distributed architecture where each regulator is on a separate printed circuit board (PCB). The bus voltage can be a specific value or it can be sun-regulated [22].

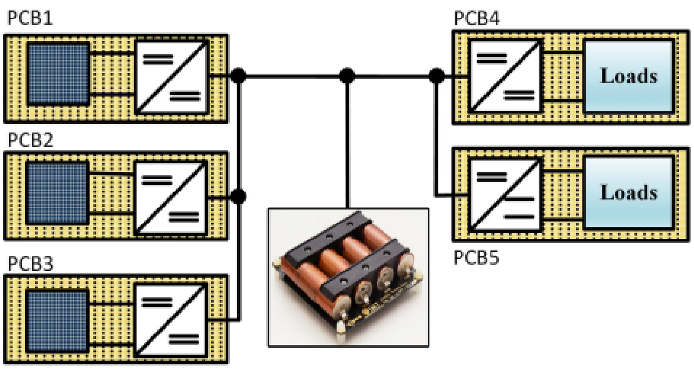


Figure 4.4: Distributed Power Conversion [20]

The centralised EPS is the most common implementation as it simplifies the requirements for the attached systems and it is space efficient to place all power components in a single location. Although the distributed EPS has more complex requirements of attached subsystems, it allows for increased modularity.

Interface with PV Panels

The two main interfaces between a CubeSat and its PV panels are direct energy transfer (DET) and maximum power point tracking (MPPT).

In a DET configuration, PV panels are connected through series diodes directly to the storage system and loads. For this reason DET is the simpler of the two interfaces to implement. However, if the voltage output from the PV panels drops below the battery or system voltage level the system is forced to shut off.

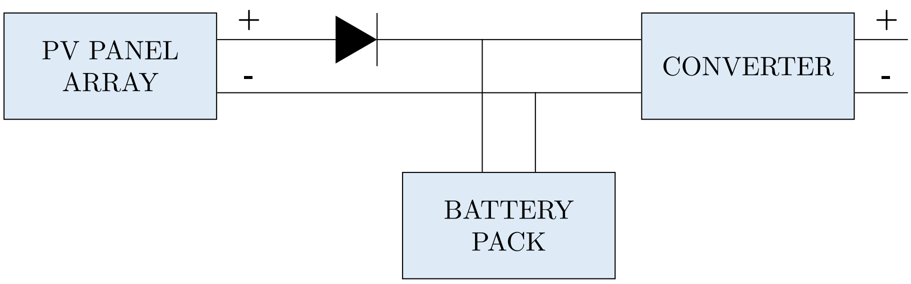


Figure 4.5: DET Solar Setup

MPPT utilises DC-DC converters to adjust the PV panel's output voltage to ensure that the system is always extracting the maximum power possible from the PV panels. This addition allows the system to better handle varying operating conditions, such as sun inclination and panel temperature [23]. The main drawback of MPPT is that it is complex to implement and requires multiple converters. Despite this drawback, MPPT increases the efficiency of solar panels to such an extent that it is used on almost all modern CubeSats [20].

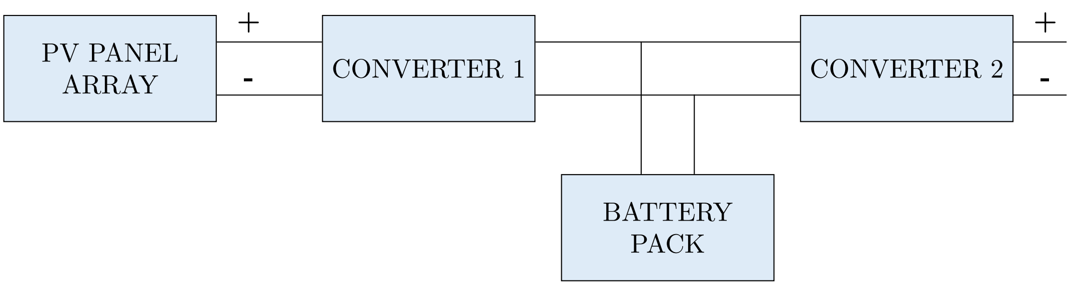


Figure 4.6: MPPT Solar Setup

4.1.3. Review of Existing EPS

The BenchSat EPS will have similar requirements to that of current industry products. The iEPS from ISIS and P31u from GOMspace, shown in Figure 4.7, are power systems that are designed to be used on 1U to 3U satellites.

The iEPS and the P31u both have the option to use two to four batteries in multiple configurations. This flexibility in power input allows both EPSs to be suitable for many different satellite mission profiles. Each provides regulated 5V and 3V3 power rails and supplies an unregulated power line.

The general specifications of the iEPS and the P31u vary slightly between the systems. The P31u documentation states that it can provide an output power of 30 W. It also has a 5V bus with a 4 A maximum output current and a 3V3 bus with a 5 A maximum output current. The iEPS can deliver 20 W on the 5V bus with a maximum current of 4 A. Both systems have multiple channels that can be switched on and off and both systems provide payload protection in the form of overcurrent protection and reverse current protection. Overcurrent and reverse current protection are important to protect both the payload and the satellite bus in unexpected conditions, such as a short circuit or if a user incorrectly connects a payload.

GOMspace tested the performance of the P31u by measuring its converter efficiency and line loss. Both of these tests are worthwhile to categorise the functioning of the BenchSat EPS.

(a) iEPS [24]

(b) P31u [25]

Figure 4.7: Reviewed EPSs

4.2. Design

The main function of the BenchSat EPS is to achieve the desired system characteristic 3 in Section 3.1 which is to provide power to the BenchSat satellite bus, attached payloads and the air bearing carrier cart. Additionally, the EPS should provide protection and monitoring to attached systems and the satellite bus as per desired characteristic 4. This section defines the EPS design process to achieve the desired functionality.

4.2.1. EPS Requirements

The EPS requirements are broken into primary requirements relating to power delivery and secondary requirements relating to protection, monitoring and charging.

Primary Requirements

1. Rated regulated power
2. Multiple power channels

The EPS must provide BenchSat and its attached loads 5V and 3V3 regulated power channels as well as an unregulated input voltage channel at between 7.6 V (2S battery configuration) and 15.2 V (4S battery configuration). In addition the regulated channels must be capable of handling a maximum current of 4 A and a continuous current of 1 A.

Based on the examples reviewed in Section 4.1.3, the EPS must have four 5V power channels and four 3V3 power channels. One power channel at each voltage level will be unswitched and will be used as a dedicated system power channel to provide power to the satellite bus.

Secondary Requirements

1. Load protection
2. Voltage measurement
3. Current measurement
4. Energy storage and charging

Load protection refers to protecting the circuit from unexpected malfunctions. The EPS must protect the satellite bus and payloads against over-current, under-voltage and over-voltage as well as provide reverse current protection. These protections should be present on all regulated power channels.

The EPS must be able to measure the voltage of the input channel, the 5V rail, and the 3V3 rail. The EPS must measure the voltage of each level regularly and store the data to be available upon request.

It is also necessary to measure the current usage of each regulated channel. The EPS must measure the current use of all active channels regularly and other subsystems should be able to acquire any current reading upon request.

Energy storage and charging refers to the battery system. The EPS must allow for balanced battery charging and the battery pack must be able to supply sufficient power to the rest of the system to meet its requirements.

BenchSat will predominantly be used indoors and therefore there is no requirement for it to have solar charging capabilities. Furthermore, the air bearing carrier cart has a time limit of approximately 30 minutes [4] and solar charging is not useful in this amount of time. Solar power generation is, however, a fundamental part of all CubeSats, therefore BenchSat will be built to be able to accommodate the attachment of solar systems.

4.2.2. System Architecture

As noted above, solar charging is not included in the system design as it is not a requirement for the BenchSat EPS; nevertheless, the implementation of the system will accommodate the later addition of solar charging systems. As the solar charging system will be separate from the EPS architecture it can be either MPPT or DET.

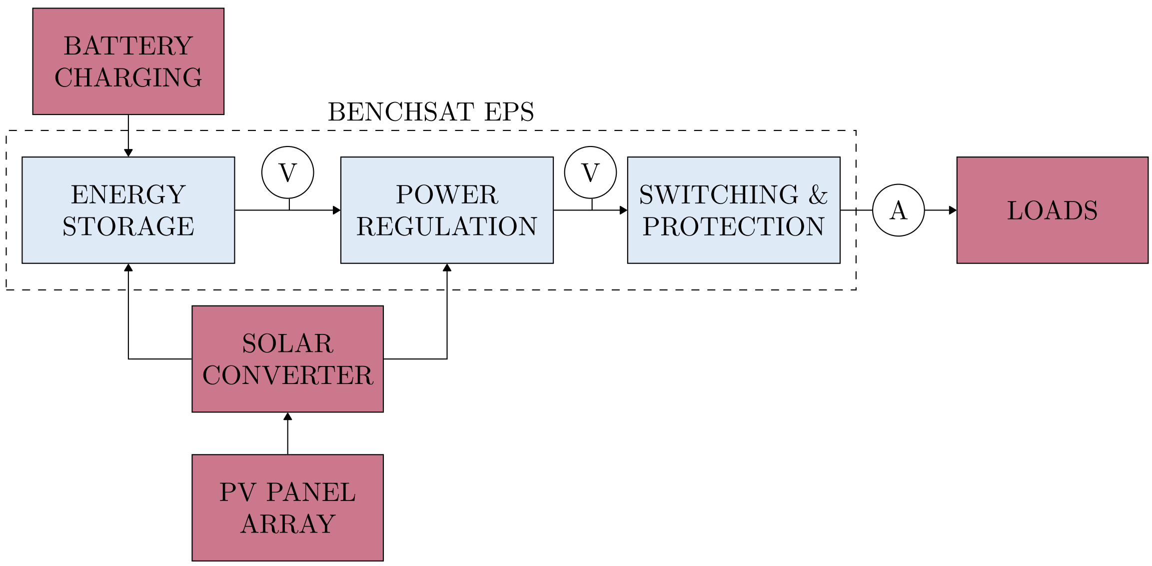


Figure 4.8: EPS Power Flow and Architecture

Figure 4.8 shows the implemented power flow and architecture for the BenchSat EPS. The architecture includes energy storage and a battery system that can be charged using an external charger.

The EPS has a multistage power conversion architecture with a dedicated power converter for output power regulation and two separate attached systems for battery charging and solar control. Separating these systems creates flexibility and simplifies the design. The increased number of components is not relevant as space is not the primary issue for BenchSat.

The power regulation stage takes the input voltage and regulates it to the desired bus voltages of 5V and 3V3. The switching and protection section then splits the power into multiple channels. Protection is linked to switching because the protection functionality is available in many electronic switches.

The output power regulators and switches are placed on a single PCB in a centralised architecture and regulated power is shared with the attached components. Although a distributed architecture may be beneficial in some cases, for this project the BenchSat EPS provides the attached subsystems with regulated power.

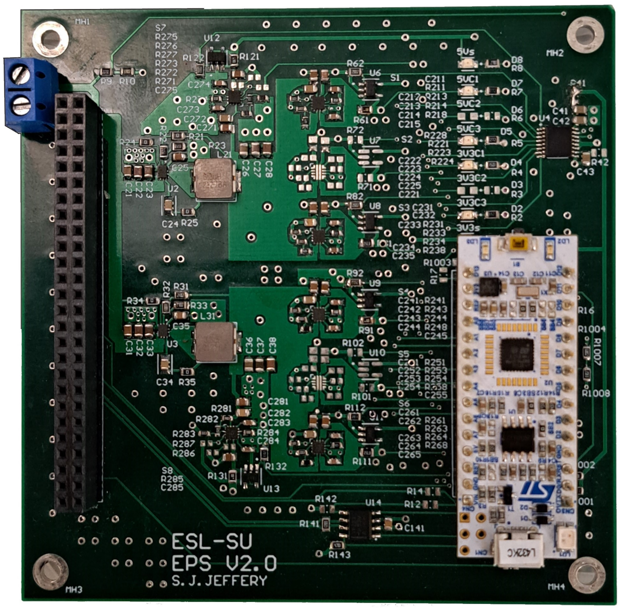
Should a future project wish to move towards a distributed system, the BenchSat EPS will provide an unregulated channel that can be regulated at that stage. A distributed system option could encourage payloads to be more modular and include all required components onboard.

Finally, the overall BenchSat design requires individual modules to be self-sufficient, as per desired system characteristic number 8 in Section 3.1. To meet this requirement the EPS microcontroller manages and monitors the EPS. All commands to any component on the EPS are sent through the microcontroller. The physical hardware on the EPS is connected to the rest of the system via the power connections. The microcontroller connects to the satellite bus and communicates information over the shared CAN communication bus. Voltage sensing components are placed on the input voltage and after the power regulators. Current sensing components are placed between the switches and the loads. The EPS microcontroller uses feedback from these components to monitor the system.

4.3. Implementation

4.3.1. Hardware Implementation

The architecture laid out in Section 4.2.2 was designed and manufactured onto two PCBs, a power conditioning PCB and a battery PCB. The PCBs conform to the BenchSat requirements and structures to ensure that the EPS can connect to the BenchSat subsystems. The resultant systems are shown in Figures 4.9a and 4.9b. The design schematics and PCB layouts are shown in Figures A.1 and A.2 in Appendix A.



(a) Power Conditioning PCB

(b) Battery PCB

Figure 4.9: EPS Boards

In order to meet the design requirements, all of the components in the EPS power flow are at a minimum 4 A rated and the system can handle a wide range of input voltages including, 2S, 3S, and 4S battery sources.

The battery board (Figure 4.9b) designed for this project accommodates two 18650 Li-Ion batteries in a 2S formation. In addition, the battery board has multiple headers to allow for power sharing with the solenoid control board and to interface with a solar charger. If a dedicated solar controller is used, the designed implementation can bypass the switch on the battery board.

Power is transferred through BenchSat's shared header from the battery board to the power conditioning board. Two switch mode regulators on the power conditioning board create the 5V and 3V3 rails. Switch mode regulators are used because of their efficiency and their ability to handle high currents. E-switches are used for each regulated channel to protect attached payloads and the satellite bus. The different functions of the EPS are tested and examined in Section 4.4. The power conditioning board shares the unregulated battery power output, two unswitched system channels (a 5V channel and a 3V3 channel), and six switched channels (three 5V and three 3V3) with the rest of the system.

4.3.2. Functional Implementation

During operation, the EPS will only run processes in three cases:

1. Prompted by a CAN message
2. Prompted by a schedule
3. Prompted by a warning

After being prompted the EPS then has three general responses that it can undertake:

1. Communicate: Send a CAN message - warning, information, request
2. Action: Set or reset switches
3. Measure: Take a reading from one of the peripheral devices

Using these basic prompts and actions the EPS can be run as a basic finite state machine. The main program flow runs sequentially using drivers to interface with the relevant components.

At a scheduled frequency (every 100 ms) the EPS will sample all the active channels to measure current and voltage. This is not enough to follow transient responses but the goal of measuring the channels is to monitor the system and not to provide in-depth analysis. After each sample, the data will be stored on the EPS and are accessible through the component drivers. The EPS will not transfer the data to the OBC. This implementation leverages the fact that BenchSat is modular so rather than repeat data on two devices, the data is stored in temporary memory on the EPS. It is easy and efficient for the OBC to request data as required.

These functions can be expanded as components or systems are expanded. If a solar charging system is added, further actions can involve interacting with the solar controller.

4.4. Testing

Experiments are performed on the EPS to test the subsystem's functionality. The tests varied from measuring the accuracy of sensors to comparisons with other similar systems. The tests conducted included an EPS efficiency test, a voltage drop-off test, a current limiting test, and a power usage test.

The basic test setup is shown in Figure 4.10. Multimeters are used to measure voltage and current at selected points and power is sourced either from a separate power supply or the EPS batteries. A test PC controls the experiments using commands sent over a UART connection. The UART connection can control the switches and receive telemetry from the EPS microcontroller.

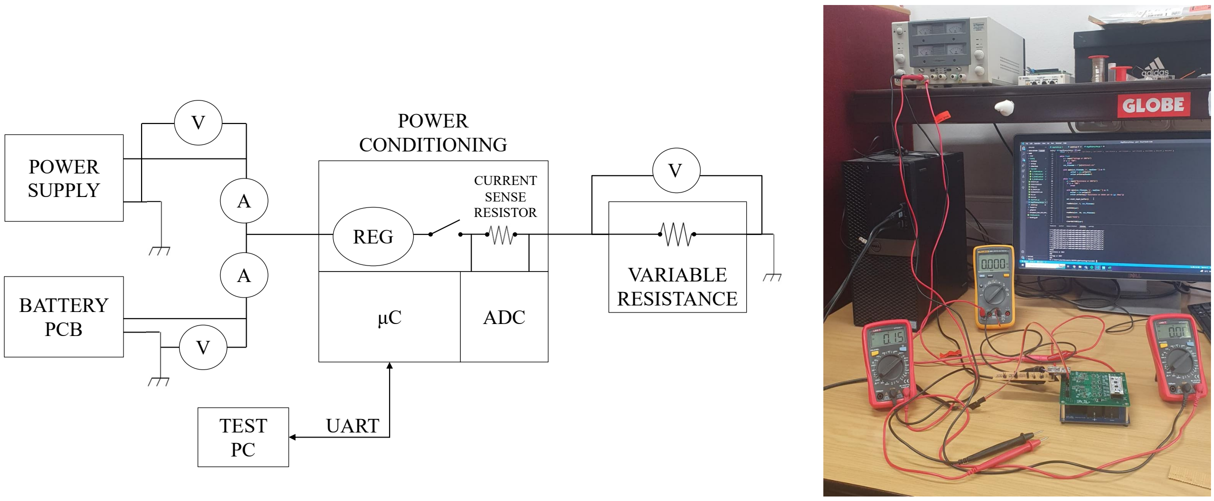


Figure 4.10: EPS Test Setup Description

4.4.1. EPS Efficiency Test

Test Setup and Background

Calculating the efficiency of an EPS is important for determining the satellite power budget. In this case, efficiency is important to compare with other systems to find out where the system can be improved.

EPS efficiency (η) is calculated by measuring the output power of the EPS:

$$\eta = \frac{P_{\text{output}}}{P_{\text{input}}} = \frac{\sum V_{\text{sys}} \cdot I_{\text{sys}} + \sum \frac{V_{\text{load}}^2}{R_{\text{load}}}}{V_{\text{input}} \cdot I_{\text{input}}} \tag{4.1}$$

The outcome of this test is the EPS' efficiency. The output power is controlled by varying a resistance network attached to the EPS. The input voltage and current are also measured with multimeters. The load voltage and resistance are measured with a multimeter. The system current is processed using an INA169 current sensor and the voltage is processed using a voltage divider. The outputs of the current sensor and the voltage divider are connected to an onboard analog-to-digital converter (ADC) where the final measurement is performed. Using Equation 4.1, the input and output power can be calculated and the efficiency determined. The experiment is repeated for different input voltages.

Results and Discussion

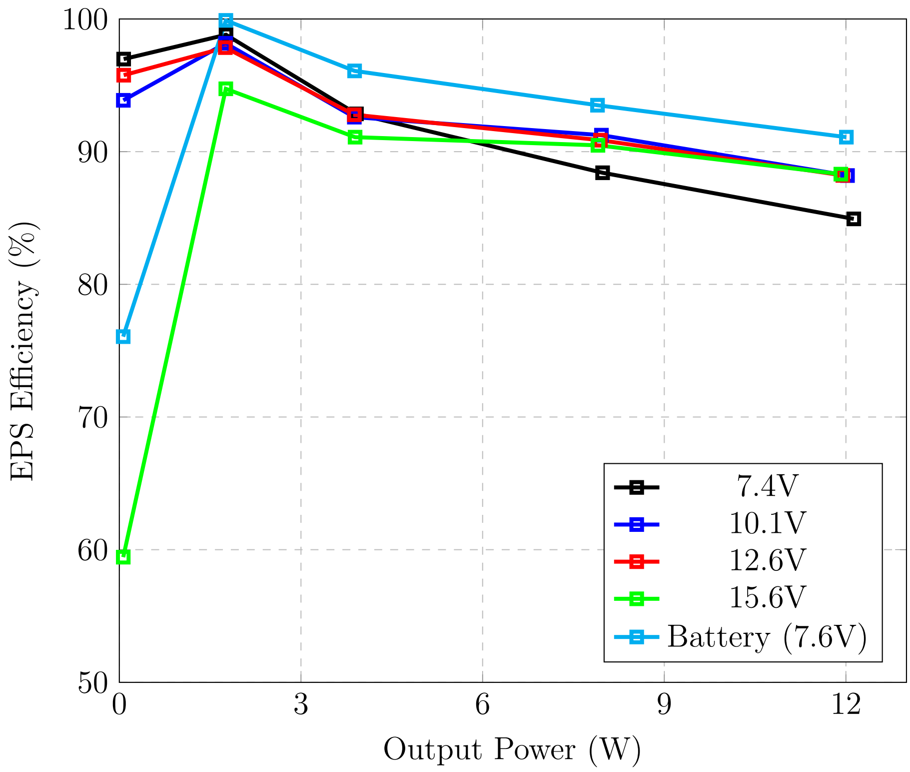


Figure 4.11: EPS Efficiency Output Curve at Different Input Voltages

The result of the efficiency testing is shown in Figure 4.11. The efficiency is shown to vary between 85 and 95 percent under various loads at different input voltages. The initial increase in efficiency at very low power is assumed to be due to inaccuracies in measuring low currents where a slight offset is significant. The efficiency generally decreases as output power increases. These results are similar to those seen for the P31u [25]. A comparison of the two results is shown in Figure 4.12 This test also demonstrates that the EPS functionality is not compromised when connected to the battery system.

The results of this test demonstrate that the converter is working correctly and the EPS can deliver power to attached loads.

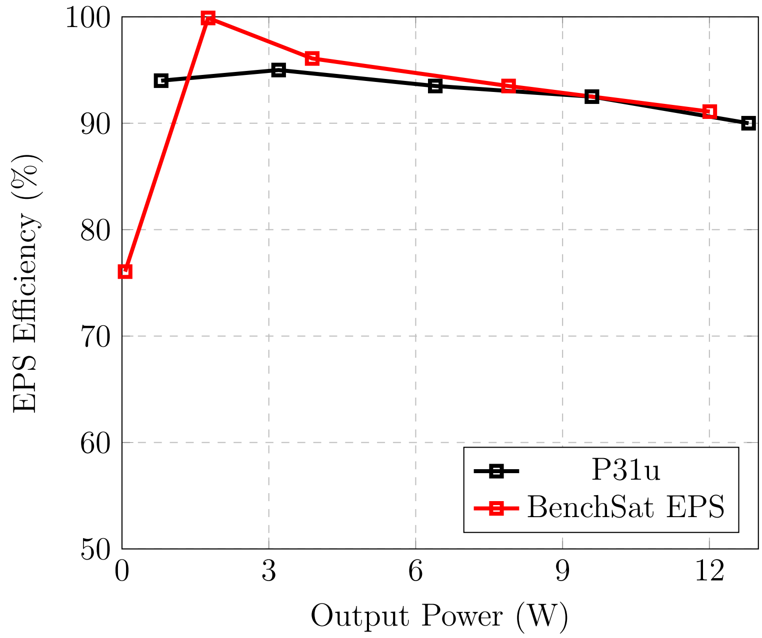


Figure 4.12: EPS Efficiency Comparison between the BenchSat EPS and the P31u

4.4.2. Voltage Drop-off Test

Setup and Background

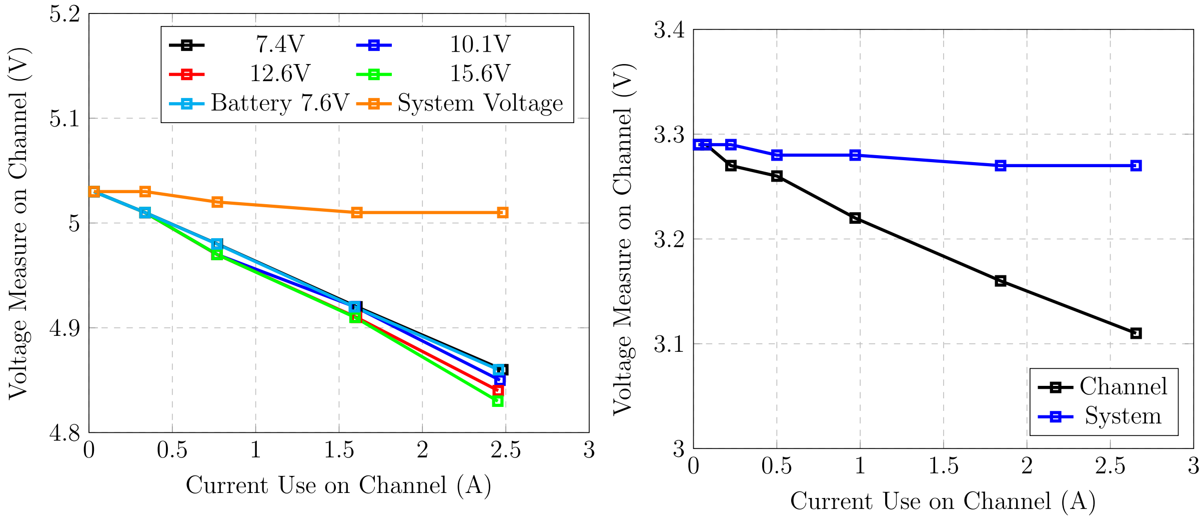
The voltage drop-off or line loss is the drop in voltage due to resistance in the switches, connectors and PCB tracks. It is important to measure voltage drop-off to give the user an accurate representation of the voltage output that can be expected for a given load. The converter voltage is also monitored in this test. If the converter output voltage drops other systems could be affected by a large load decreasing the voltage across the entire bus. For the 5V test, multiple input voltages are used to see if the input voltage impacts the result.

Ideally there would be no drop across the switch. However, voltage drop-off is expected to increase as the current increases. Therefore this experiment compares channel voltage to channel current.

A resistor circuit is used to vary the current draw of the system. The channel voltage and the system voltage are measured using a multimeter. Current is calculated using Ohm's law.

Results and Discussion

The results shown in Figure 4.13 demonstrate the results of the three questions raised above, namely (i) what is the voltage drop-off at different output currents, (ii) is system voltage affected by increased load current and (iii) does input voltage affect the voltage drop-off.



(a) EPS 5V Voltage Drop-off versus Current at Varying Input Voltages

(b) EPS 3V3 Voltage Drop-off versus Current

Figure 4.13: EPS Voltage Drop-Off Test

Firstly the line loss of 3.5% at 2.5 A on the tested 5V channel is high. In comparison, the voltage drop-off at the same current for the GOMspace EPS is less than 1%.

Secondly, both figures demonstrate that despite voltage drop-off on the active channel the system channel stays constant dropping less than 1%.

Finally, Figure 4.13a clearly shows that input voltage does not impact the output voltage. This result can be seen with all the data (excluding the system channel) tracking each other.

The results show that the voltage decreases linearly. This linearity demonstrates that the drop-off is due to the resistance within the switch and the surrounding components. The expected resistance in the measured switch is 25 mΩ from the current sense resistor and 28 mΩ through the switch [26] for a total of 53 mΩ. Based on the results shown in Figure 4.13a the observed path resistance is 70 mΩ.

The voltage drop-off needs to be addressed in future work. The problem is possibly due to the layout of the switch used on the power conditioning board. The expected resistance could also be decreased by removing the current sense resistor and using the switches' current sensing functionality, although there might be a loss in accuracy.

4.4.3. Current Limiting Test

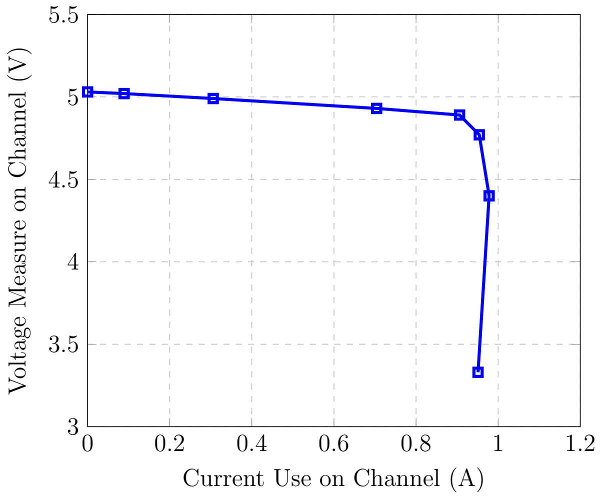
Setup and Background

Part of the functionality of the EPS is to set a current limit on specific channels. This test demonstrates how that limit works by comparing output current and channel voltage.

The switch should limit the current by decreasing the output voltage once the current limit is reached. The current is not cut off completely and excess power is dissipated in a dedicated resistor.

As in the previous tests, the current is controlled by a variable resistance network. The test was done with a current limit of 1 A.

Results and Discussion



(a) EPS 5V Channel 1A Current Limit **(b)** EPS 3V3 Channel 1A Current Limit

Figure 4.14: EPS Hardware Current Limiting Test

As can be seen in Figure 4.14, the current in both cases reaches a maximum current and then does not move further. The downward slope of the line initially is due to line losses discussed in Section 4.4.2.

In this case, the current is not switched off by the EPS, but rather it is limited. The current can be switched off in software by monitoring the current sensor and turning the channel off if it reaches a selected value.

4.4.4. EPS Power Usage Test

Setup and Background

The power usage on the EPS is calculated using its current monitoring components. None of the power connections are active and only the power used by the EPS components and microcontroller is recorded. The information is used to categorise the overall functioning of the BenchSat EPS.

Results and Discussion

The results of the power usage test are shown in Table 4.1.

3V3 Current	5V Current	3V3 Power	5V Power	Total Power
6.16 mA	9.90 mA	20.3 mW	49.5 mW	69.8 mW

Table 4.1: Power Use on the EPS

The EPS itself uses minimal power and will not influence the functioning of attached payloads. Further, as shown in the voltage line loss testing in Section 4.4.2, the main power losses in the system are in the switches, so the power delivery to payloads will not be influenced by the EPS' power usage.

4.5. Summary

This section covered the background of a CubeSat EPS including the function that it fills in a standard CubeSat and a discussion of EPS architecture classifications. Specific focus was then placed on two commercial off-the-shelf (COTS) EPSs. The next two sections focused on the design and implementation of the BenchSat EPS, including the design requirements, design architecture and hardware and software implementations. Finally, the results of the testing were displayed and discussed.

The BenchSat EPS can provide sufficient power to the rest of the system and meets the requirements set out in Section 4.2.1. The converter efficiency is in line with other industry EPSs and shows that the BenchSat EPS does not have major power losses.

The hardware design and implementation can be deemed a success as each component onboard the EPS is functional. This includes the regulators, switches, current and voltage sensing and the microcontroller with the various communication methods including CAN, I2C and UART. However, as mentioned in Section 4.4.2, this method of current sensing leads to line losses. Future work should include investigating a different method of current sensing and further improvements to the PCB design to reduce the losses in the switch.

The regulator and switch choices in this project were forced due to supply issues. Future work could investigate different options with more convenient footprints. The current footprints require a skilled technician to place and, in a student context, it would be useful to be able to make adjustments without requiring a high level of skill and equipment.

The next chapter will follow a similar structure and focus on the radio communication subsystem.

Chapter 5

Radio Communication Subsystem

This chapter discusses the design of the BenchSat radio communication subsystem (RCS). The RCS connects BenchSat and a ground station via radio frequency communications. This chapter will cover background information, design considerations, hardware and software implementation, and system testing with a discussion of the results.

5.1. Background

5.1.1. The Functions of an RCS in a Satellite

The functions of the radio communication subsystem in a satellite include transmitting data and telemetry to a ground station, receiving commands from a ground station and communicating with other satellites [27]. A full communications system includes a ground segment of ground stations on earth and a space segment onboard the satellite.

A radio frequency (RF) communication link consists of a radio transmitter, a free space communication channel and a radio receiver. This link creates a one-way connection where data can be sent to the transmitter, transferred on a high-frequency electromagnetic wave and received at another location where the data can be used. The task of the radio communication subsystem onboard a satellite is to transmit the signals on the downlink and receive signals on the uplink.

5.1.2. Frequency Bands

The frequencies used to transmit radio signals are grouped into frequency bands with upper and lower limits. Selected frequency bands are shown in Table 5.1. Higher frequency bands allow for higher data rates, however, they lead to increased free space losses. Free space losses account for the decrease in a signal's power as it travels further away from the transmitter. These losses need to be offset by higher transmission power and antenna gain.

The need for higher data rates has driven a move to higher frequency bands and newer small satellites have implemented S, X and K band communication systems [27]. However, VHF

Band	Frequency Range
VHF	30 to 300 MHz
UHF	300 to 1000 MHz
S	2 to 4 GHz
X	8 to 12 GHz
K	12 to 40 GHz

Table 5.1: Selected Frequency Bands

and UHF are generally popular for amateur radio and telemetry, tracking and command (TT&C). This popularity is because they have low free space losses, which makes it easier to transmit signals at a long range when a large data rate is not required. Furthermore, VHF and UHF transmission can often make use of simple dipole antennas and are less dependent on accurate ground station pointing, making it possible to communicate with the satellite regardless of its orientation.

Frequency bands can be further broken down based on their use. Amateur radio bands can be used without a licence and are used for a wide range of purposes including wireless experimentation. The South African amateur radio bands in the VHF and UHF range are from 144 MHz to 148 MHz and from 430 MHz to 440 MHz respectively.

5.1.3. Satellite Communication Architecture

An RF communication system consists of a controller, a radio, an amplifier, and an antenna. Transmission occurs when the radio receives a message from the controller. The radio modulates an electronic wave to generate an output signal containing the data it received in the message. The signal is then passed through a low noise amplifier to increase the output power before it is transmitted by an antenna, which increases and focuses the strength of the RF signal [27].

To receive, an antenna picks up a transmitted signal and passes it to a low noise amplifier. The amplifier increases the power and decreases the noise, enabling the radio to decode the received signal. The received signal is then transferred to the controller for processing [27].

5.1.4. Modulation

Modulation is the process of modifying a consistent repeating signal to transfer information. The existing signal is called a carrier signal and the information modulated onto it is the baseband signal [28]. A carrier signal is normally a high-frequency sinusoidal signal with a constant amplitude, frequency and phase. These characteristics are modified during modulation to encode the baseband signal onto the carrier signal.

Digital modulation is used to transfer non-continuous digital data. For this process the baseband signal is limited to specific levels. The main benefit of digital modulation is that it limits the effect of noise since the digital data is limited to a finite set of values. Digital signals can be transferred via a normal radio if the audio being transmitted is limited to specific frequencies. The audio is transferred via radio and the digital data is recovered on the receiver side.

5.1.5. Digital Modulation

BenchSat is required to transmit digital data rather than audio so the focus of this project is on digital modulation.

Frequency Shift Keying

Frequency shift keying (FSK) is a modulation technique where the frequency of the carrier signal varies to transmit a digital message. As can be seen in Figure 5.1, when the modulated wave has a higher frequency a digital 1 is transmitted and when it has a lower frequency a digital 0 is transmitted.

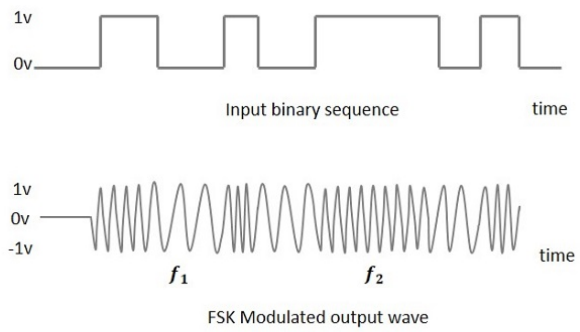


Figure 5.1: FSK Explanation [29]

FSK is the modulation technique used during this project to be compatible with the TIM Ground Station, which uses FSK modulation [5]. FSK is simple to use and is often used in amateur radio systems.

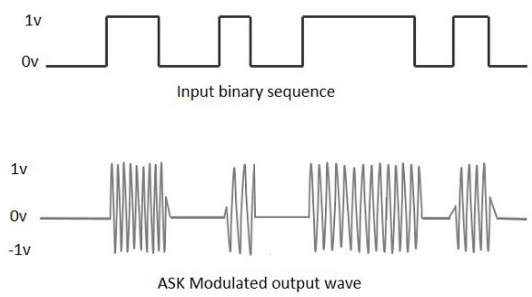
Audio Frequency Shift Keying (AFSK) is an early form of FSK that was used to send digital data using basic radios. An audio tone is set to either 1200 Hz or 2200 Hz to encode data, and can then be transmitted with a standard FM radio. AFSK packet radio had a standard baud rate of 1200 bit/s [30].

To increase the maximum possible data rate James Miller developed a new FSK modulation standard. This standard is G3RUH and it is a 9600 baud modem. The increase in speed is enabled by expanding transmission frequencies outside of the audio spectrum and using the full capabilities of a Narrow Band Frequency Modulation (NBFM) radio [30].

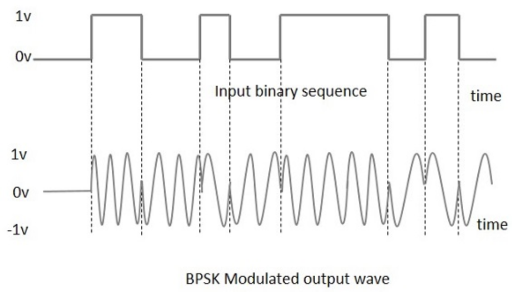
Comparison with other modulation techniques

Amplitude Shift Keying (ASK), shown in Figure 5.2a is a form of modulation where the amplitude of the carrier wave is varied to transmit the signal. ASK is rarely used in satellite systems because the power requirements are greater than other modulation techniques and therefore require larger transmitters [31].

Phase Shift Keying (PSK) is a modulation technique where the phase of the carrier signal is varied to encode information. When compared with FSK, PSK has a lower bandwidth as it only uses a single frequency. However, it is susceptible to phase disturbances that can lead to loss of information [31]. Binary phase shift keying (BPSK) is shown in Figure 5.2b.



(a) ASK Depiction [29]



(b) BPSK Depiction [29]

Figure 5.2: Other Modulation Techniques

5.1.6. Telemetry, Tracking and Command

The telemetry, tracking and command system on a satellite enables the ground station to confirm that the satellite is performing correctly. The three main tasks of the TT&C system include monitoring the status of the satellite, calculating the satellite's position, and controlling the satellite via specific commands [32]. TT&C baud rates can range from 40 bps to 10 kbps [31]. These functions on BenchSat will be performed by the RCS.

5.1.7. AX.25

AX.25 Amateur Packet Radio Link Layer Protocol is an encoding protocol that is often used in amateur radio applications. AX.25 has been used with packet radio modems like G3RUH and AFSK [30]

Packet radio messages are grouped into small segments of data called frames. There are three general types of AX.25 frames: information frames, supervisory frames and unnumbered frames [33]. These frame constructions are shown in Tables 5.2 and 5.3.

AX.25 is a useful example to use as a reference when designing a packet radio connection. Two components of AX.25 are discussed below.

Flag	Address	Control	Info	FCS	Flag
01111110	112/224 Bits	8/16 Bits	N·8 Bits	16 Bits	01111110

Table 5.2: AX.25 Unnumbered and Supervisory Frame Construction [33]

Flag	Address	Control	PID	Info	FCS	Flag
01111110	112/224 Bits	8/16 Bits	8 Bits	N·8 Bits	16 Bits	01111110

Table 5.3: AX.25 Information Frame Construction [33]

Flags and Bit Stuffing

As seen in Table 5.2, the AX.25 flag can be identified as six consecutive "1" bits in a row. Flags identify the start and end of frames so it is useful for them to be unique and not appear at any other point in the packet.

Bit stuffing is the process of adding extra bits to a message to preserve meaning. To ensure that the flag bit sequence does not appear accidentally in a frame, the packet handler monitors the bits being added to the packet for any sequence of five consecutive "1" bits. If five consecutive "1" bits are found the packet handler inserts a "0" bit after the fifth "1" bit. The receiver then discards any "0" bit that immediately follows five "1" bits [33]. Bit stuffing has the added benefit of breaking up long runs which can be difficult for a receiver to decode.

Frame Check Sequence

The frame check sequence is a value that is calculated by both the sender and the receiver to ensure that no data has been lost in transmission. The value is based on the data transmitted. The standard for AX.25 is a 16-bit cyclic redundancy check.

5.1.8. Review of Existing RCS

Cape Peninsula University of Technology's (CPUT) CMC and GomSpace's NanoCom AX100, shown in Figure 5.3a, are examples of commercially available VHF/UHF CubeSat transceivers. Many of the features on these transceivers will be relevant to the design of the BenchSat RCS.

The frequency bands covered by the two devices have similar ranges and are focused predominantly around the amateur bands mentioned in Section 5.1.2. The CMC is a full-duplex transceiver with a set VHF uplink and UHF downlink while the NanoCom AX100 is a half-duplex transceiver with only a single antenna. The NanoCom AX100, therefore, transmits and receives in the same frequency band.

The CMC is primarily used for TT&C with baud rate options of 1.2 kbps and 9.6 kbps. The CMC is compatible with the AX.25 protocol. The NanoCom AX100 is also compatible

(a) CPUT CMC

(b) GomSpace NanoCom AX100

Figure 5.3: Reviewed RCSs

with AX.25 but includes other framing options as well. The NanoCom AX100 can support baud rates from 0.1 kbps to 38.4 kbps.

The CMC uses an I2C interface for command, telemetry, and user data enabling it to be controlled by most systems. The controller that uses the CMC is required to have I2C capabilities but since I2C is so widely used this limitation is unlikely to be a problem. The NanoCom AX100 has a wider variety of interfaces and can communicate with I2C, UART, and CAN. The diversity of interfaces increases the system's flexibility to communicate with a controller.

The CMC's transmit power can be set between 27 dBm and 33 dBm and its sensitivity is -117 dBm. The NanoCom AX100 transmit power can be adjusted between 24 dBm and 30 dBm and it has a sensitivity of -137 dBm.

5.2. Design

The function of the BenchSat RCS is to provide a connection between BenchSat and a ground station. This function is listed in desired system characteristic number 4 in Section 3.1. The RCS enables BenchSat to be controlled by a ground station and to transmit data to a ground station.

5.2.1. Requirements

Design requirements are split into primary requirements and secondary requirements. The primary requirements relate to the RCS' communication capabilities. The secondary requirements relate to the functioning of the RCS on BenchSat.

Primary Requirements

1. FSK UHF (430 MHz-440 MHz) transmit capabilities

2. FSK VHF (144 MHz-148 MHz) receive capabilities

The frequency ranges are based on commonly accepted amateur frequency bands and the capabilities of the ground station [5]. The receiver on the ground station works in the UHF band and the transmitter works in the VHF band. Therefore, RCS must transmit on the UHF band and receive on the VHF band to be compatible.

The bands selected in point 1 and 2 above are beneficial because lower frequencies have reduced free space losses and therefore propagate further through space. This lower free space loss allows the system to transmit further with less power, which is important for low-cost satellites.

As mentioned in Section 5.1.5, the RCS uses FSK modulation which is compatible with the TIM Ground Station and there is a history of its use in the amateur radio setting.

Secondary Requirements

1. Full-duplex communications

2. Baud rate: Minimum 1200 bps, Goal 9600 bps

3. 500 m range

4. Onboard computing and storage capabilities

5. CAN communication capabilities

6. Limited power use ($<$ 220 mW peak, $<$ 100 mW mean)

The RCS is required to be full-duplex so that it does not miss messages sent to it while transmitting.

The baud rate requirement is based on the capabilities of the CMC reviewed in Section 5.1.8 and existing standards like G3RUH and AFSK [30]. To attain useful TT&C a minimum of 1200 bps is required. However, if the system can handle 9600 bps it would show a high level of functionality and would make the system comparable to commercially available options.

The practical range required makes it possible to transmit and receive from the TIM Ground Station to the testing area in the ESL. This distance is shorter than 500 m, but components are specified to have a range of at least 500 m to mitigate the interference caused by the surrounding buildings. Future work may include testing done at a longer range, such as high-altitude balloon flights.

The onboard computer should be powerful enough to handle multiple simultaneous actions including transmitting RF messages, receiving RF messages, and communicating with the rest of BenchSat.

CAN has been selected as the primary method of communication for BenchSat. This decision leverages the fact that each module has a controller onboard that is capable of using CAN to communicate.

Due to limited power capabilities on a satellite; the RF module should be able to function on low power. The RCS should not influence the overall capabilities of the power system and its ability to power payloads. The specifications were chosen using the CMC as a reference point.

5.2.2. System Architecture

The RCS architecture is adapted from the general architecture discussed in Section 5.1.3. Additionally, the RCS has a dedicated microcontroller that is used to control the communication system. This dedicated controller acts as a buffer between the BenchSat subsystems, such as the OBC, and the radio connection to the ground station.

On the RCS, the radio and amplifier sections have been combined into one by using modules to perform the RF communications. As can be seen in Figure 5.4, the UHF and VHF modules connect directly to the antennae without the need for amplifiers, impedance matching or a balun.

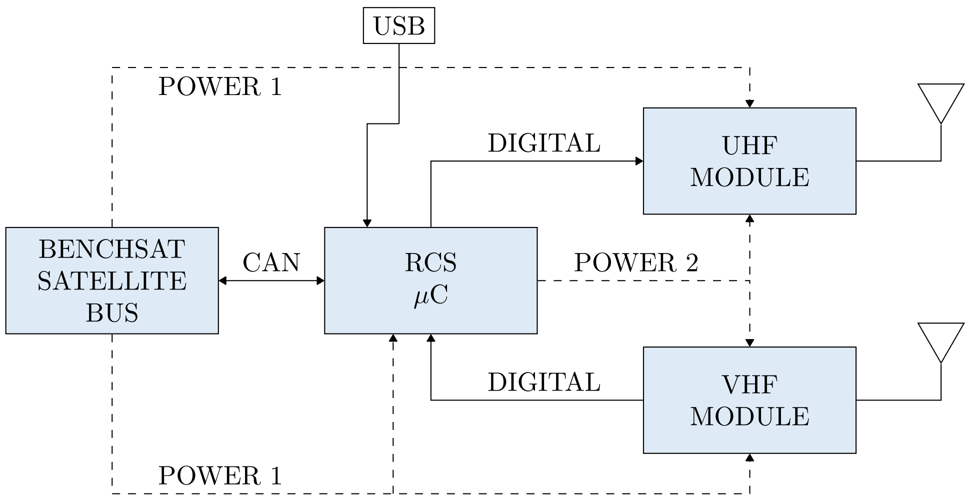


Figure 5.4: RCS Architecture

Figure 5.4 also shows that the RCS functions with two antennae. The antennae are separate from one another and work in different frequency bands so the signals do not interfere with each other. Separate antennae and modules enable the RCS to be a full-duplex system.

RF processing is done within the transmit and receive modules. Digital data are transferred between the radio frequency modules and the RCS microcontroller. Data processing is done onboard the microcontroller. The microcontroller is connected to the rest of BenchSat by the shared CAN-bus. Unless a CAN message's destination is the RCS, the microcontroller acts as a buffer connecting the BenchSat CAN-bus to the RF link between BenchSat and the ground station. Temporary storage on the RCS microcontroller is used to implement this buffer successfully.

All the components are chosen as low power modules. Due to the low range requirements, the power required can be reduced. If the range is increased the power usage will also need to increase.

The architecture provides the option for the RCS to be powered by BenchSat or via USB. During normal functioning the RCS draws power from BenchSat via the shared header, marked Power 1 in Figure 5.4. During standalone testing the RCS can also be powered via USB through the RCS microcontroller module, marked Power 2 in Figure 5.4. A jumper connects the microcontroller module power to the rest of the RCS. This jumper is removed during normal functioning to stop the two regulators from interfering with one another during use.

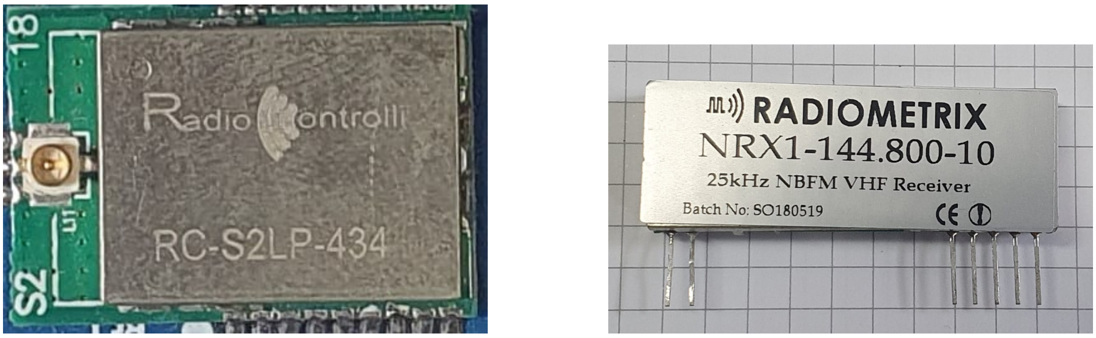
5.3. Implementation

5.3.1. Hardware Implementation

RF Module Selection

The use of COTS RF modules removes the need for complex circuit design because all RF circuitry is located onboard the modules. The requirements for the radio communication modules selected for BenchSat are based on the RCS requirements, BenchSat's capabilities and the TIM Ground Station capabilities. The RC-S2LP-434 module was chosen for the RCS transmitter and the NRX1-144.8MHz module was chosen for the RCS receiver. The modules are shown in Figure 5.5 and were selected based on their FSK capabilities, range, frequency options and power usage.

Each device provides a digital connection to the RCS microcontroller and connects to an antenna with a $50\,\Omega$ impedance connection [34] [35]. The modules handle RF signal filtering, RF matching, and timing.



(a) UHF Module: RC-S2LP-434

(b) VHF Module: NRX1-144.8MHz

Figure 5.5: RCS RF Modules

RCS PCB

The architecture defined in Section 5.2.2 was designed, manufactured and populated on a single two-layer PCB to complete the RCS. The completed RCS is shown in Figure 5.6. The design schematic and PCB layout are shown in Figures A.3 and A.4 in Appendix A. The RCS connects to BenchSat via the shared header and standoffs. The RCS connects to the antennae with two side mount SMA connectors attached to the PCB.

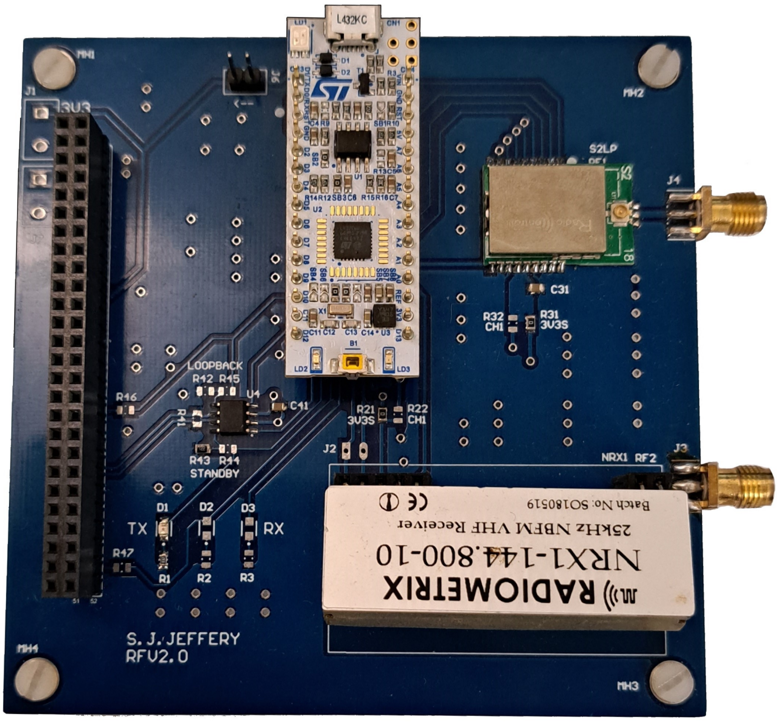


Figure 5.6: Assembled Radio Communication Subsystem

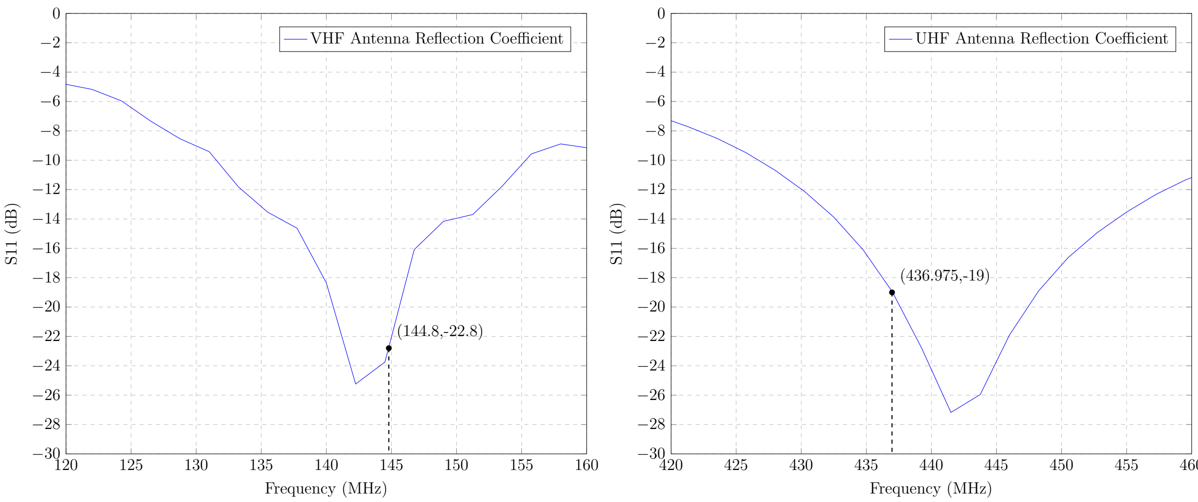
5.3.2. Antenna Selection

As mentioned in Section 5.2.2, the RCS includes two antennae: a transmit antenna for the downlink and a receive antenna for the uplink. The transmit antenna is a stubby antenna tuned to 433 MHz with a gain of 2 dB [36]. The receive antenna is a quarter-wave monopole antenna that is tuned to the VHF band. The estimated gain of a quarter wave monopole antenna is 3 dB [37]. During tests using the air bearing facility, the air bearing carrier cart base is used as the ground plane for the monopole antenna. The antennae used are shown in Figure 5.7.

(a) UHF Stubby Antenna **(b)** VHF Monopole Antenna

Figure 5.7: RCS Antennae

The reflection coefficient (S11) for both antennae was measured empirically and the results are shown in Figures 5.8a and 5.8b. S11 represents the magnitude of the power reflected from an antenna and is used to measure how effectively power is radiated by the antenna. Since the coefficient measures reflection, lower reflection values mean that more power is radiated by the antenna. An S11 value lower than -10 dB is considered acceptable.



(a) VHF Antenna **(b)** UHF Antenna

Figure 5.8: Reflection Coefficient Output Graphs

The exact frequencies used are 436.975 MHz and 144.8 MHz for the downlink and uplink respectively. From Figures 5.8a and 5.8b, the S11 values are -22.8 dB and -19 dB respectively showing that the antennae are well matched to the frequencies that they cover.

5.3.3. Link Budget

A link budget quantifies the performance of a radio communication link by accounting for the various gains and losses between the transmitter and the receiver. Since the antennae and RF modules have been selected a link budget can be calculated. The link budget is calculated for the connection between the TIM Ground Station and the ESL air bearing facility where experiments will take place.

From [38], the received transmission power for both the uplink and the downlink is:

$$P_{\mathrm{RX}} = P_{\mathrm{TX}} - L_{\mathrm{TX}} + G_{\mathrm{TX}} - L_{\mathrm{PATH}} + G_{\mathrm{RX}} - L_{\mathrm{RX}} \tag{5.1}$$

where P_{RX} is the received power, P_{TX} is the transmitted power, L_{TX} is the losses in the transmitter, G_{TX} is the transmit antenna gain, L_{PATH} is the propagation losses between the transmit and receive antennae, G_{RX} is the receive antenna gain and L_{RX} is the losses in the receiver.

The propagation losses can be estimated as the free space loss which is:

$$L_{\mathrm{PATH}} = L_{FS} = 32.45 + 20 \cdot \log d + 20 \cdot \log f \tag{5.2}$$

where L_{PATH} is the propagation loss, L_{FS} is the free space loss, d is the distance listed in kilometers, and f is the carrier frequency in megaHertz.

The excess power received over the minimum level that can be received is the fade margin. The minimum level is approximated to the sensitivity of the receiver. The fade margin is:

$$F = P_{\mathrm{RX}} - S_{\mathrm{RX}} \tag{5.3}$$

where F is the fade margin, P_{RX} is the received power and S_{RX} is the receiver sensitivity.

Using these equations the link budget can be calculated as shown in Table 5.4.

Band	**Uplink**	**Downlink**
TX Power (dBm)	50	16
TX Gain (dB)	14.4	2
Free Space Losses (dB)	69.64	79.24
Estimated Losses (dB)	5	5
RX Gain (dB)	3	18.9
RX Power (dB)	**-10.24**	**-47.34**
RX Sensitivity (dBm)	-80	-60
Fade Margin (dB)	**72.76**	**12.66**

Table 5.4: Link Budget

All the information used in Table 5.4 is from [5] [34] [35] or based on the assumptions listed below:

- L_{TX} and L_{RX} can be summarised into estimated losses and given a value of 5 dB for both the uplink and downlink.
- The TIM Ground Station receiver sensitivity is -60 dB, which is 20 dB above the measured noise floor under normal conditions.
- The distance between TIM Ground Station and the ESL test area is 500 m as stated in the secondary requirements in Section 5.2.1.

From Table 5.4 it can be seen that both the uplink and the downlink have fade margins greater than 10 dB. These results indicate that the connection should be successful.

5.3.4. Packet Handling

Data needs to be packaged correctly to be transmitted over an RF link. The data is placed into a packet that can be transmitted and received. The implemented packet is broken down in Table 5.5.

Component	**Layout**	**Description**
Preamble	Repeated (10)	The connection is asynchronous, the preamble helps to link the two devices
Start Flag	01111110	Identifier that shows that a message is starting or ending
ID	3·(8 bits)	Used to identify the sender
DATA	n·8 bits	Sent data
CRC16	2 byte CRC	Cyclic redundancy check to ensure that the correct data has been transmitted and received
End Flag	01111110	Identifier to mark the end of the message
Postamble	Repeated (10)	Sent to pad the message to separate it from the noise that may follow

Table 5.5: Packet Breakdown

The packet is based on the AX.25 packet structure that was discussed in Section 5.1.7. The same constituent parts are present but some lengths have been changed. However, the packet can be easily updated to fully utilise AX.25.

5.3.5. Digital Connection to Modules

As shown in Section 5.2.2, the RCS microcontroller controls the RF modules via digital communications. The two RF modules, the S2LP for UHF and NRX1 for VHF, have different interfaces and this section covers how the microcontroller controls these modules.

UHF: S2LP

The connection to the S2LP is an SPI and GPIO link. The S2LP is programmed via the SPI connection and selected outputs and interrupts can be programmed onto GPIO pins. The S2LP is a transceiver with multiple frequency options so the same RF board can be repurposed for many other functions if required.

The S2LP has multiple options that can be controlled over the SPI connection depending on the application. These options are set via registers onboard the S2LP. To send messages, data is sent to a first-in, first-out (FIFO) register where the message is stored until a transmit command is sent. The S2LP must be commanded by functions on the RCS controller, however, using the FIFO and other storage registers it can complete tasks on its own. This means that while the S2LP is transmitting a message the RCS controller is not busy, allowing the system to run as a full-duplex system.

An interrupt is used to signal when a message has been fully transmitted so another message can be sent. Without this interrupt, commands could be overlapped and messages may be lost.

VHF: NRX1

The NRX1 has three outputs connected to the RCS microcontroller. Since the module is exclusively a receiver it has no inputs from the microcontroller. The three outputs are audio, digital logic and received signal strength indicator (RSSI) [35].

An ADC setup on the microcontroller is used to read the RSSI output. The RCS microcontroller uses this output to measure the power of any received signal. The audio output has not been used in this project. However, additions can be added to use this output if required during future work.

The NRX1 outputs the received data onto the digital logic pin. The received data is a binary signal that is either one or zero. To decode the information, the microcontroller uses a timer interrupt to trigger the system to sample the pin. The interrupt is triggered at a rate that is eight times faster than the transmitted baud rate. The process for decoding the information is shown in Figure 5.9. The system counts the number of consecutive signals of a given value. When the received value changes the counter is divided by eight and the data is stored in temporary memory. The piece of data that is stored is the number of consecutive bits received.

If the incoming message flag is set the data is saved in a buffer. Otherwise, the system attempts to identify the start flag and ID, from the packet described in Table 5.5, in received data. If consecutive signals define the correct flag and ID, the system will set the incoming message flag. When the end flag is received the data packet is complete; the

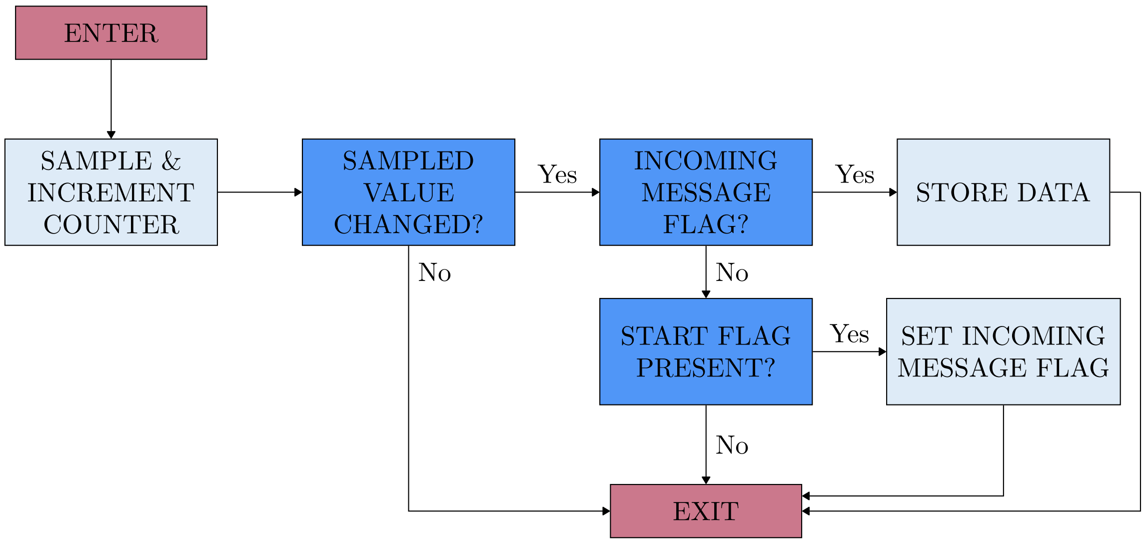


Figure 5.9: NRX1 Reciever Timer Interrupt Flowchart

incoming message flag is reset and the main program is notified to process the received information stored in the buffer.

This processing is done based on individual bits that are received. Knowing every bit that is received is useful for debugging and allows for greater control of the system. This control is beneficial as it enables the implementation of any communication protocol, bit stuffing or frame checking.

Using this method to decode the messages leverages a benefit of the distributed system. The powerful, dedicated RCS microcontroller is capable of more intensive processing, such as running a timer interrupt at a high frequency.

5.3.6. Functional Implementation

This section focuses on the main control loop for the RCS microcontroller. The functions described in Section 5.3.5 and the CAN driver are separated from this loop and only provide it with buffers for information and flags when they are available or need to be processed.

The RCS is a communication module so functions will happen based on commands from other BenchSat systems or the ground station. Therefore, the inputs that will trigger an action on the RCS can be listed:

1. A CAN message from another BenchSat system
2. An RF message from the ground station
3. An internally scheduled process

Each of these triggers will prompt the main loop of the RCS controller to run a function to first determine what is being asked of the system and then execute that task. After receiving a prompt the system has the following options:

1. Pass on communications destined for other systems

2. Run a function onboard the RCS

Passing on communications involves translating information between CAN and radio communications. If a message from the ground station that is destined for the OBC is received, the RCS repackages the message as a CAN message and sends it on. In the opposite case, the RCS repackages a CAN message from the OBC and sends it to the ground station over RF.

RCS functions can include reprogramming the S2LP, setting LEDs, running unit tests or processing telemetry. These processes are defined depending on the experiment.

5.4. Testing

The following section discusses the tests used to analyse the functionality of the BenchSat RCS. All of the tests, other than the power test in Section 5.4.5, use the RCS as a standalone module. The standalone setup is shown in Figure 5.10.

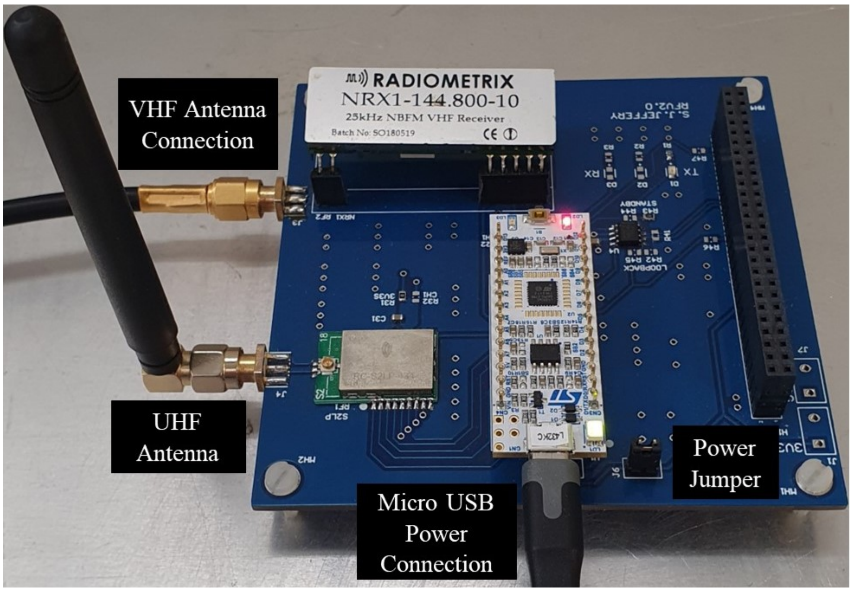


Figure 5.10: RCS Standalone Testing Setup

5.4.1. Transmit Baud Rate Test

Setup and Background

The transmit baud rate test measures the accuracy and ability of the RCS to transmit at varying baud rates. The RCS is programmed to send 20 messages with 50 bytes of data encoded and a 16-bit CRC. The successful transmission of each message is confirmed by a separate receiver. The TIM Ground Station and the ESL Ground Station were used for this test. The TIM Ground Station is at the maximum distance required for BenchSat functionality.

Results and Discussion

The results of the transmit test at varying baud rates are shown in Table 5.6.

Baud	**ESL Ground Station**		**TIM Ground Station**	
Rate	Messages RX	Percent RX	Messages RX	Percent RX
1200	20	100	20	100
4800	20	100	20	100
9600	20	100	20	100
19200	20	100	20	100

Table 5.6: Transmit Baud Rate Test Results

As can be seen in the results the system outperforms the requirements and can transmit at double the expected rate with 100% accuracy.

5.4.2. Receive Baud Rate Test

Setup and Background

The receive baud rate test is used to test the accuracy and ability of the RCS to receive data at varying baud rates. During each test, the RCS received 20 messages with 50 bytes of data and a 16-bit CRC. If the message passed the CRC test it was recorded as a success. During the test, messages are sent by the ESL Ground Station and by the TIM Ground Station at the maximum expected range. The method of transmission is discussed in Chapter 7.

This test measures the RF connection between the ground station and the RCS as well as the RCS's ability to decode information with the algorithm described in Section 5.3.5.

Results and Discussion

Figure 5.11 shows the results of the RX baud rate test. The results show that the system can handle messages sent from both transmitters at the goal baud rate of 9600 bps.

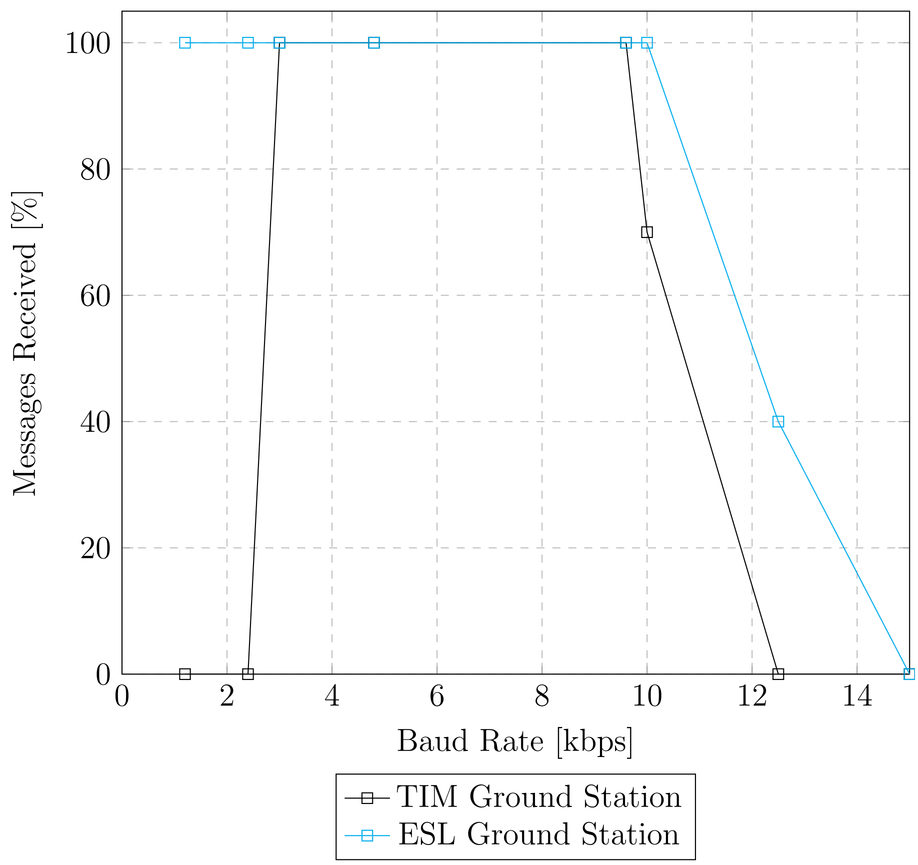


Figure 5.11: Percentage of Messages Received by RCS at Different Baud Rates

As can be seen in Figure 5.11, when the transmission baud rate is greater than 9600 bps the RCS handles messages from the ESL Ground Station better than it handles messages from the TIM Ground Station. This difference is likely due to the increased signal noise on the longer-range message. Errors are therefore more prevalent on the longer connection than on the shorter one.

A drawback of the TIM Ground Station transmitter is that it cannot send messages with a baud rate lower than 3 kbps. The physical hardware that is used to transmit requires an increased baud rate to remain active. When data is sent at lower baud rates the transmitter switches off to conserve power. The result is still considered a success since a higher baud rate can be achieved. Future work can cover investigating the existing TIM Ground Station components to solve this problem.

Using the ESL Ground Station the RCS was able to receive all messages up to a baud rate of 10 kBps. 10 kBps is the theoretical maximum for the NRX1 receiver [35].

5.4.3. Transmission Speed Test

Setup and Background

The speed of transmission is limited by the data rate. However, it is useful to measure how quickly the system can transmit a new message after the previous one was completed. This test measures the downtime between messages that are successfully received.

To find the minimum downtime possible, the RCS is programmed to send a new message as fast as possible after the last message was sent. The results are then captured by the receiver and measured to show the gap in time between messages. This test also measures the total message length and packet data length to make further observations about the packet. The packet data is counted as the data between the start flag and the FCS.

Results and Discussion

Figure 5.12 shows an example of the feedback received. From this feedback, the time at specific points in the message can be measured to calculate the length of each part of a given message. These measurements are summarised in Table 5.7. Each test consisted of 20 messages and all were successfully received and recorded by the receiver.

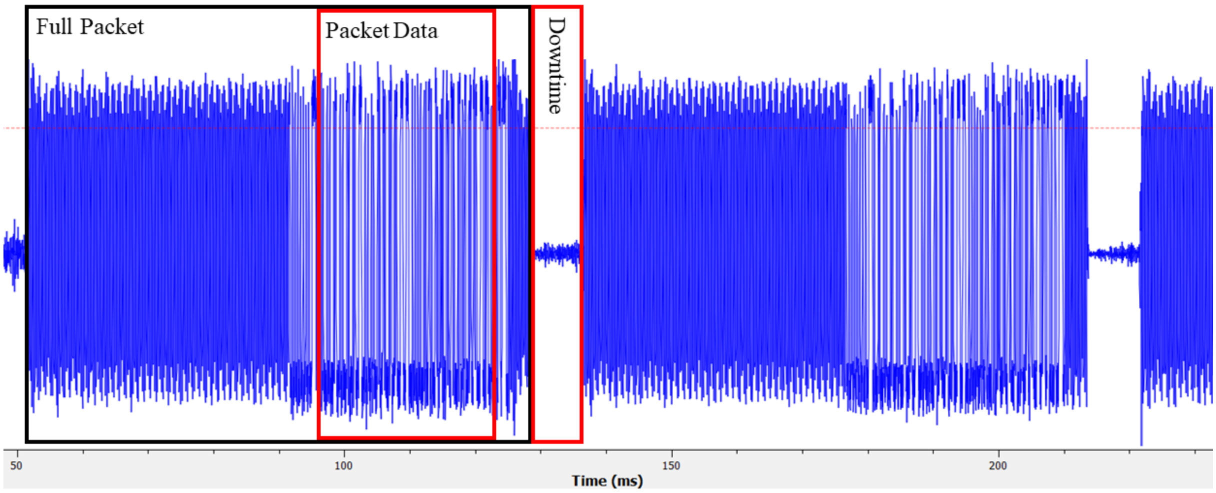


Figure 5.12: Received Data from Transmission Speed Test

Data (bytes)	Total Packet (ms)	Packet Data (ms)	Down (ms)
15	64	14	8
30	77	27	8
45	90	40	10

Table 5.7: Speed Test Timing Results

The time taken to send the data within the packet increases proportionally to the data length. The variation from a perfect relationship is due to bit stuffing, which is allocated based on the data so it is not proportional. The rest of the packet remains constant throughout and consistently takes 50 ms to be delivered.

There is a measured downtime of 8-10 ms between messages. The downtime between messages increases slightly as the messages get longer and more time is required to format the packet and populate the FIFO, but the difference is small.

5.4.4. Full-Duplex Test

Setup and Background

A full-duplex test is used to confirm that the system can handle messages traveling in both directions at the same time. This test checks if there is interference between the two channels used by the RCS and if the RCS processor can handle TX and RX at the same time. To test the system's capability for full-duplex communications the system is commanded to transmit as fast as possible, similar to the transmission speed test in Section 5.4.3.

Results and Discussion

The TIM Ground Station transmitted 20 messages and all 20 messages were received by the RCS. In that period the RCS was able to transmit 413 messages, which demonstrates that it was constantly transmitting. The fact that the RCS was able to receive each message confirms that the RCS has the processing power to handle full-duplex communications. Further, it shows that the different wavelengths did not affect each other as the transmissions did not affect the incoming messages.

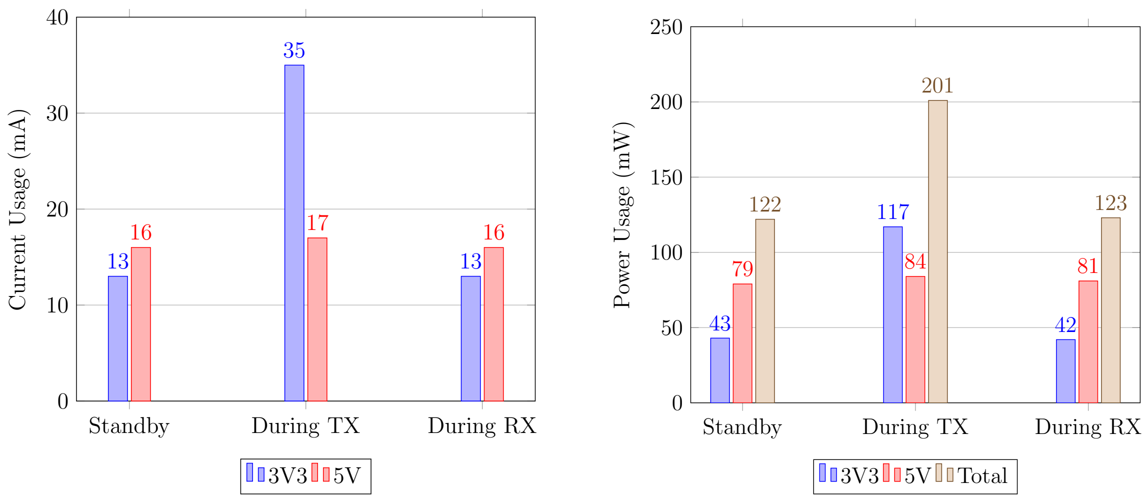
5.4.5. RCS Power Test

Setup and Background

A power test measures the RCS's power usage and whether or not it falls within the system requirements. The BenchSat stack is set up to include the RCS and the EPS. The current measurements from the EPS are transmitted to a control PC during testing. These data allow the current measurements to be recorded. A baseline usage is recorded with only the EPS connected. The RCS is then attached and the power use is measured. The test is repeated with the RCS in standby mode, during message transmission and while a message is received. From these results, the RCS power use can be calculated.

Results and Discussion

The power testing confirms that the RCS power use is in line with the requirements. As shown in Figure 5.13, the maximum instantaneous power used is 201 mW, which is less than the minimum requirement of 220 mW. These results confirm that the power use on the RCS will not affect the overall functioning of the system.



(a) RCS Current Usage per Channel **(b)** RCS Power Use

Figure 5.13: RCS Power Usage Test Results

The current usage test results, shown in Figure 5.13, confirm the correct functioning of the devices on the RCS and that the system is working as expected. Firstly the 5V power requirement stays constant throughout the test. This result makes sense as only the microcontroller uses the 5V power and thus no variation is expected.

Secondly, the 3V3 current increases by approximately 20 mA between the standby and TX states. This result is the same as the expected value for current use stated in the S2LP datasheet. This result is evidence that the S2LP is functioning correctly. The standby current use of 16 mA accounts for the NRX1, CAN transceiver and standby state of the S2LP.

5.5. Summary

This section covered the process of researching, designing and building the BenchSat RCS. The RCS was designed to use modular components where possible to simplify the design process. The RCS was then fully tested to determine its functionality.

The RCS meets all the primary and secondary requirements laid out in Section 5.2.1. The system design is streamlined so the component count can be minimised. Testing demonstrated that all of the implemented components are functioning as desired.

In terms of power use and data rate, the RCS achieves an outcome that is comparable to the CMC reviewed in Section 5.1.8. The RCS's transmit power and receive sensitivity are not as good as the reviewed components. Future work can include choosing more powerful components. However, for BenchSat, the required range is small so increased power and sensitivity are not required.

Future work could also include improving the algorithms used for transmitting and receiving. In the current implementation, if a message needs to be transmitted but the S2LP is already sending a message the RCS will pause processing to wait for the transmitter to become free. In this project the message volume is too low for this issue to cause a problem. However, the issue could be solved by implementing a buffer system to queue messages until they can be sent.

Further, the algorithm for receiving messages could be improved to utilise the RSSI to flag when a message is possibly being received, rather than checking every bit that the NRX1 outputs as it does in the current implementation.

The next chapter will cover the background, design, implementation and testing of the BenchSat onboard computer subsystem. The OBC is the final BenchSat subsystem that will be discussed.

Chapter 6

Onboard Computer Subsystem

This chapter covers the background, design, implementation and testing of the BenchSat onboard computer subsystem (OBC).

6.1. Background

6.1.1. Functions of an OBC in a Satellite

The OBC plays multiple roles in the standard functioning of a satellite. The primary goal of the OBC is to manage the tasks taking place on the satellite. These tasks can include managing communication between subsystems, and monitoring and maintaining the correct functioning of the entire system. The OBC is the main source of computing power onboard a satellite. The primary OBC hardware is a microcontroller that is connected to the other subsystems with digital communication links and protocols. On smaller satellites the functions of multiple subsystems can fall under the scope of the OBC.

To simplify the design, a single microcontroller can cover the functionality of the command and data handling (C&DH) subsystem, the attitude determination and control subsystem (ADCS) controller and a payload processor. These subsystems are highlighted in Figure 6.1 to show their relationship to the rest of the subsystems in a satellite. Each of the highlighted subsystems fulfills a specific role on a satellite, as explained briefly below.

<table>
<tr><th>SATELLITE PAYLOAD</th><th>SATELLITE BUS</th></tr>
<tr><td rowspan="4">PAYLOAD</td><td>RCS</td></tr>
<tr><td>EPS</td></tr>
<tr><td>C&DH</td></tr>
<tr><td>ADCS CONTROLLER</td></tr>
<tr><td>PAYLOAD
PROCESSOR</td><td>ADCS SENSORS
& ACTUATORS</td></tr>
</table>

Figure 6.1: Internal Layout of a Satellite

C&DH Subsystem

One of the OBC's primary functions is command and data handling. A C&DH subsystem performs two main functions: it distributes commands to other subsystems and it gathers and processes mission data for use on the satellite or to downlink to a ground station [31]. The C&DH subsystem can often include additional components and functions such as a real-time clock, subsystem health monitoring, or security features like data encoding.

The complexity of the C&DH subsystem is directly proportional to the complexity of a spacecraft. The greater the number of subsystems on a spacecraft, the more monitoring and processing functionality is required on the C&DH subsystem.

ADCS Controller

The ADCS is used to stabilise the satellite and point it in a chosen direction if commanded during a mission. This task requires the satellite to determine its attitude using sensors and then control itself using actuators [31].

The ADCS controller is responsible for running algorithms and performing calculations necessary to achieve the desired orientation [39]. The feedback from the attitude sensors is used in the calculations and the ADCS controller commands the actuators to execute the result to correctly orientate the satellite.

Payload Processor

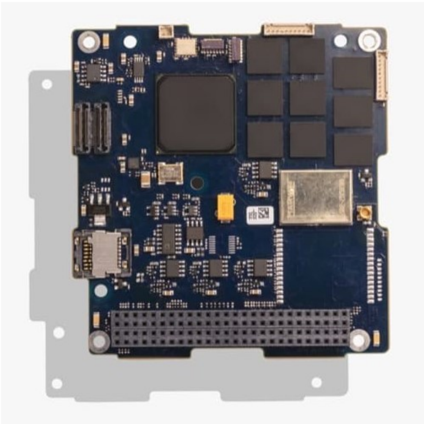
The payload is the component of the satellite that interacts with the outside world to achieve mission objectives [31]. The task and makeup of the payload will vary depending on the mission.

A payload processor is used to control the actions of the payload, record data, process data, and communicate with the satellite bus to transfer data back to the ground. An earth observation camera is an example of a payload. In this instance, the payload processor will command the satellite when to capture and store images and when to process the images, for example, if compression is required. Once the tasks have been completed the payload processor will transfer the results to the satellite bus to be transmitted to the ground or stored.

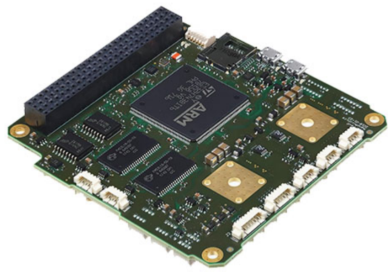
6.1.2. Review of Commercial OBCs

Three commercial off-the-shelf (COTS) OBCs were reviewed to assist in determining relevant components and requirements for the BenchSat OBC. These OBCs are the SatBus 3C2 from NanoAvionics, the Kryten-M3 from AAC Clyde Space and the EnduroSat OBC.

All three are advanced OBC units that can be used on their own as well as integrated into a system. The systems are shown in Figure 6.2.



(a) AAC Clyde Space Kryten-M3 [40]



(b) NanoAvionics SatBus 3C2 [41]

(c) EnduroSat OBC [42]

Figure 6.2: Examples of COTS OBCs

The primary hardware on each OBC is the microcontroller that executes all of the systems processes. Of the reviewed OBCs, the EnduroSat OBC and the SatBus 3C2 both use ARM Cortex M7 cores while the Kryten-M3 has a Cortex M3. Each OBC has access to a form of permanent storage, being a microSD card for EnduroSat and the SatBus 3C2; and extended flash memory for the Kryten-M3. All three systems have multiple communication options, including CAN, SPI, I2C and UART interfaces.

The EnduroSat OBC and the SatBus 3C2 both include onboard sensors such as magnetometers and gyroscopes. These additions demonstrate how the functions of an OBC and other systems can overlap.

From [40], the Kryten-M3 typically draws 400 mW of power. This value is more than what would be expected of the BenchSat OBC as it does not have as many external components. Although it is not a COTS OBC, the ADCS OBC designed in [39] had a minimum power of 130 mW and a peak of 435 mW. These values will be used to define a power requirement for the BenchSat OBC.

The features of the different COTS OBCs are summarised in Table 6.1.

	EnduroSat	**SatBus 3C2**	**Kryten-M3**
Core	ARM Cortex-M7	ARM Cortex-M7	ARM Cortex-M3
Maximum clock speed	480 MHz	400 MHz	50 MHz
Flash memory	2 MB	2 MB	4 GB
RAM	1 MB	1 MB	8 MB
Permanent storage	8 GB	Up to 32 GB	Flash only
Gyroscope	SPI Interface	3	None
Magnetometer	1	1	None
CAN, I2C, SPI, UART	Yes	Yes	Yes

Table 6.1: Comparison of COTS OBCs

6.1.3. Software Architecture Options

Two software implementations are reviewed to guide the design of the BenchSat OBC software architecture. The first is the system used by Botma in [39], which breaks the program flow into a foreground application and background tasks and services.

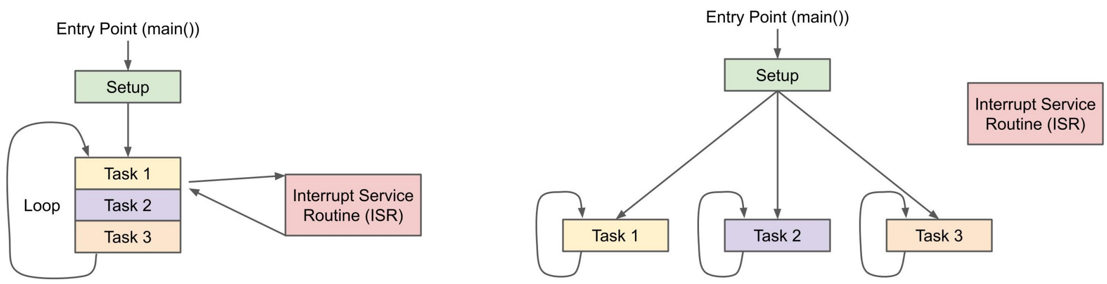
The foreground application is responsible for achieving the system's primary task. In the context of BenchSat, this foreground application would be to manage a demonstration on the air bearing table by managing movement commands, communications with other systems, sensor data and any other tasks related to the payload.

The background tasks are for managing emergencies, housekeeping and interrupts. These tasks do not necessarily complete the exact function of a specific test or demonstration but they are essential for the OBC to run correctly. The tasks include incrementing counters, managing communication nodes, and managing memory.

The other software architecture examined is the PX4 autopilot software, which aims to produce a highly reactive system. A reactive system is responsive and message-driven, which means that timing and sharing of information are viewed as priorities by the system. The PX4 software architecture splits code into self-contained modules. These modules communicate with each other via a shared bus that implements a publish and subscribe system. The outcome of this system is that whenever new information is published other systems can choose when to access it, if at all. The PX4 system ensures that all information is immediately available to any component while simultaneously ensuring that information can be received safely by modules so that they can process messages whenever they have time [43].

6.1.4. Firmware Implementation

Bare metal systems and real-time operating systems (RTOS) are two methods of running code on a microcontroller. Bare metal implementations are known as super loops because all tasks, other than interrupts, run sequentially within an infinite loop, as shown in Figure 6.3a. An RTOS is a step up from a bare metal system. An RTOS enables the program to stop and start processes concurrently allowing for multitasking as shown in Figure 6.3b.



(a) Super loop **(b)** RTOS

Figure 6.3: Super Loop vs Multitasking Applications [44]

General purpose operating systems (GPOS) are operating systems like Windows or Linux that focus on user interaction and interface. Embedded forms of GPOS can be run on powerful microcontrollers, such as embedded Linux running on a Raspberry Pi, but the overhead cost of these operating systems is too high for most microcontrollers. RTOS have minimal overheads and bare metal systems have no overheads.

The main benefit of an RTOS over a bare metal system is the ability to multitask. Multitasking enables the system to be scaled up to execute complex processes with many separate tasks. The system can also place priorities on different tasks enabling the system to meet real-time deadline requirements [45]. However, enabling the system to multitask increases overhead costs and complexity. In instances where multitasking is not necessary and sequential execution is sufficient, a bare metal system is preferable to an RTOS as it is easy to use, debug, and understand.

6.2. Design

The BenchSat OBC is responsible for the functions mentioned in Section 6.1.1. These include controlling and monitoring the satellite bus during experiments (C&DH), controlling the air bearing carrier cart's movements during experiments using onboard sensors and actuators (ADCS), and communicating with the payload to execute the experiment (payload processor). This section covers the design requirements and system architecture used to build the BenchSat OBC.

6.2.1. Requirements

The OBC design requirements are split into primary and secondary requirements. The primary requirements relate to how it controls BenchSat and interfaces with the air bearing carrier cart. The secondary requirements relate to how the OBC will connect to payloads and provide useful data during experiments.

Primary Requirements

1. ARM Cortex M4 processor (minimum)
2. Solenoid controllers

The OBC is the main computer for BenchSat and therefore needs to have enough computing power to run the experiments, including docking and ADCS operations, on the air bearing table. To facilitate a more efficient design, the microcontroller peripherals must be built into the OBC.

Solenoid controllers on the OBC are required to control the air bearing carrier cart's movements. As defined in Section 3.3.2, the OBC requires four output pins to control the four solenoids on the air bearing carrier cart.

Secondary Requirements

1. Movement and attitude sensors
2. CAN capabilities with multiple additional communication options
3. Permanent storage
4. Low power usage

The OBC covers ADCS functionality onboard BenchSat as well as its other tasks. Attitude sensors like magnetometers and gyroscopes are required to provide feedback during experiments, especially when they involve the air bearing facility. The sensors can also be integrated into future work to develop control systems on BenchSat.

The OBC is required to have CAN functionality as it is the primary communication method on BenchSat. As discussed in Section 3.3, payloads have the option to connect directly to the OBC. To enable any sort of payload connection the OBC communication options should include CAN, I2C, SPI, and UART, as shown in the review of commercial OBCs in Section 6.1.2.

Permanent storage is required to record data that is gathered during experiments. The data can be used when analysing a completed experiment or if recalled by the processor during an experiment.

The OBC on a satellite is normally switched on in all operational modes. This requires the OBC to use minimal power or to have efficient sleep modes. Based on the OBCs reviewed in Section 6.1.2, the limit for the OBC has been defined as 200 mW during standby and 400 mW during peak usage.

6.2.2. System Architecture

Due to the nature of modern microcontroller options, many of the computing functions on the OBC can be performed by a single chip. This focus is different from the EPS and the RCS where components such as regulators or RF communication modules perform the primary actions. On the OBC, the microcontroller unit (MCU) performs the important actions and the surrounding components support it in its tasks.

MCU Architecture

The MCU requires enabling components to provide computing power. The architecture for these components is shown in Figure 6.4.

Power to the MCU is filtered by decoupling capacitors to prevent power surges and noise. To ensure accurate timing, external oscillators are connected to the MCU. Although the MCU includes an internal oscillator, an external crystal oscillator is more accurate under varying thermal conditions.

The architecture allocates space for a programming header. A dedicated programmer is used to flash code onto the MCU via a serial wire debug (SWD) connection.

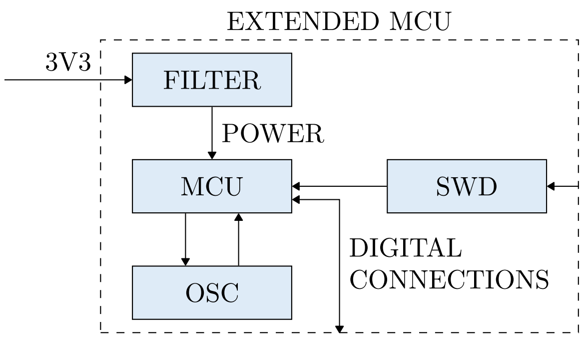


Figure 6.4: MCU Architecture

Outer Architecture

The “functionality” components shown in Figure 6.5 are directly related to the system requirements stated in Section 6.2.1, namely permanent storage, solenoid control and attitude sensors. The MCU is connected to these devices via digital connections and they are used to expand the OBC functionality to enable it to execute additional tasks. The components labeled “useability” in Figure 6.5 provide physical inputs to and feedback from the system during experiments.

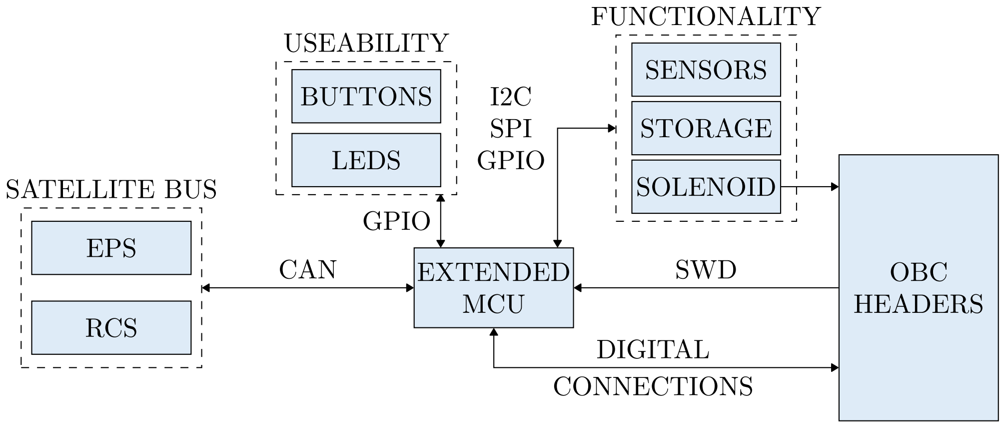


Figure 6.5: OBC Main Architecture

The OBC interacts with other systems via the BenchSat shared header and dedicated headers on the OBC. The three most important connections are CAN, the solenoid outputs and the serial wire debug. CAN connects through the shared header and allows the OBC to connect to BenchSat. The solenoid outputs connect directly to the solenoid control board as shown in Figure 3.1 and the serial wire debug connects directly to an external programmer which is not on the OBC.

Additional connections, including SPI, I2C, UART and GPIOs, are connected to both the shared header and dedicated OBC headers to meet the OBC requirements.

6.2.3. Software Architecture

The OBC software architecture is based on the options discussed in Section 6.1.3. In the architecture, a core OBC program runs the tasks required to achieve the overall goal of an experiment or mission. The functions of the surrounding components are then separated from the core OBC program via component drivers. With this architecture the core program is not dependent on the components surrounding it and vice versa.

The benefit of this architecture is modularity. Drivers publish their data to shared registers that the core program can access if required. If a component is changed a new driver file is added, but the core program remains unchanged. If the mission changes, the core program is rewritten, but the same drivers can still be used. This modularity allows for one system to change without affecting another.

6.3. Implementation

6.3.1. Hardware Implementation

Using the architecture from Section 6.2.2, the BenchSat OBC was designed and manufactured on a 4-layer PCB. The OBC PCB connects to the rest of BenchSat via the shared header. The final PCB is shown in Figure 6.6 and the design schematic and PCB layout are shown in Figures A.5 and A.6 in Appendix A.

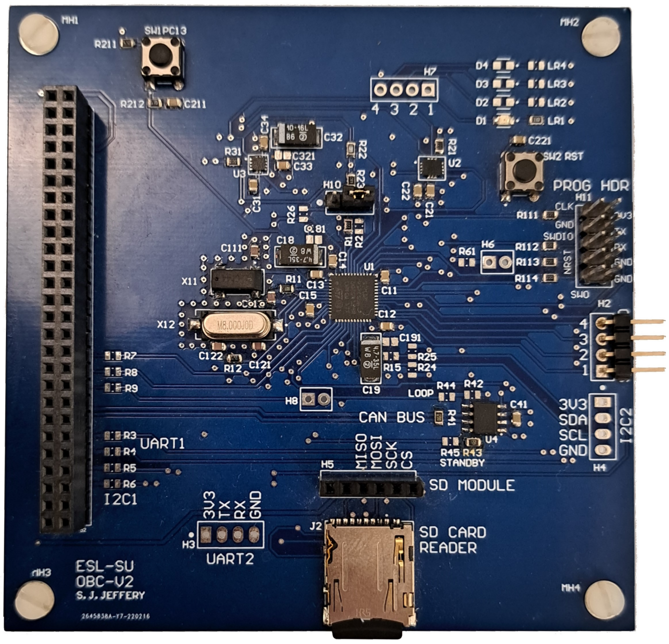


Figure 6.6: BenchSat OBC PCB

6.3.2. Microcontroller Selection

The chosen microcontroller for the BenchSat OBC is the STM32F412. The OBC MCU is more powerful than the other systems' processors to enable the OBC to perform at a higher level of processing. The capabilities of the STM32F412 are compared with that of the EnduroSat OBC MCU in Table 6.2.

	BenchSat OBC MCU STM32F412	**EnduroSat OBC MCU**
Core	ARM Cortex M4	ARM Cortex M7
Maximum clock speed	100 MHz	400 MHz
Flash memory	1 MB	2 MB
RAM	256 kB	1 MB
External FRAM	No	Yes
Radiation Hardened	No	Yes

Table 6.2: Comparison of BenchSat OBC Microcontroller and EnduroSat OBC Microcontroller

The BenchSat OBC MCU is not as powerful as the EnduroSat MCU, but it has many of the same features. The differences can be fixed in future iterations by upgrading the BenchSat MCU from the STM32F412 to a more advanced microcontroller. The comparison in Table 6.2 shows that the BenchSat OBC will be able to represent a satellite flight computer during tests and demonstrations.

The BenchSat OBC MCU and its surrounding components are shown in Figure 6.7. The power filtering capacitors and oscillators are placed around the MCU as close to it as possible. The programming header is on the side of the PCB to allow for access when integrated on BenchSat.

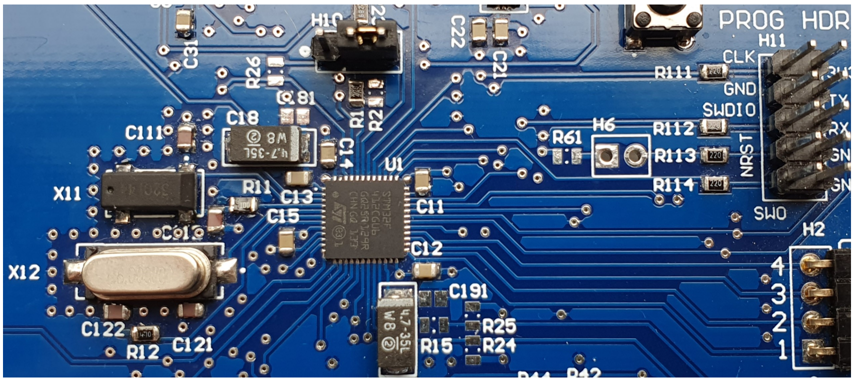


Figure 6.7: STM32F412 and Surrounding Components on BenchSat OBC

6.3.3. Onboard Functionality

This section covers the components that achieve the functionality mentioned in Section 6.2.2. As seen in Figure 6.5 three components are listed under "functionality", namely, sensors, permanent storage, and solenoids. These functions and how they are implemented are discussed in this section.

Permanent Storage

Permanent storage on the OBC is fulfilled by a microSD card. Using SPI, the device can be written to and read from when needed. The SD card connection allows the system to run experiments and record data while not using a ground station or having a direct connection to a PC.

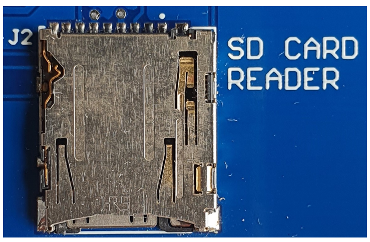


Figure 6.8: SD Card Holder

Magnetometer

The OBC magnetometer is used to calculate BenchSat's orientation using the Earth's magnetic field. CubeSats use magnetometers extensively as part of their attitude sensing and determination [46]. BenchSat's orientation can be defined as its rotation about its vertical axis. Therefore, the magnetometer calculations can be simplified to use only X and Y readings.

The magnetometer used is the LIS2MDL. The LIS2MDL is a low power, 3-axis magnetometer which the MCU connects to via an I2C connection. The LIS2MDL and its surrounding components are shown in Figure 6.9.

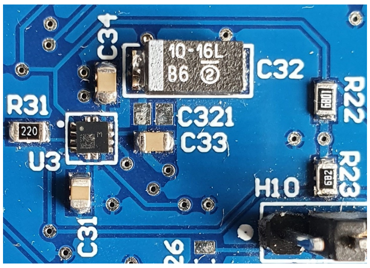


Figure 6.9: LIS2MDL and Surrounding Components

To use a magnetometer the system must be calibrated to deal with distortions. Hard-iron corrections account for distortions produced by materials that show constant additions to the field received by the magnetometer. These distortions can be caused by permanent magnets or electric current flows [47]. The hard-iron offsets in the X and Y planes are:

$$X_{\text{Offset}} = \frac{X_{\text{Max}} + X_{\text{Min}}}{2} \quad (6.1a) \quad \text{and} \quad Y_{\text{Offset}} = \frac{Y_{\text{Max}} + Y_{\text{Min}}}{2} \quad (6.1b)$$

BenchSat's orientation (ψ) in degrees can be calculated using the calibrated magnetometer readings [48]. The orientation of BenchSat is:

$$\psi = \arctan\left(\frac{Y_{\text{Reading}} - Y_{\text{Offset}}}{X_{\text{Reading}} - X_{\text{Offset}}}\right) \cdot \frac{180}{\pi} \tag{6.2}$$

Accelerometer and Gyroscope

The inertial measurement unit (IMU) used on the BenchSat OBC is the ISM330. The ISM330 is a 3D accelerometer and gyroscope that the MCU connects to via an I2C connection which it shares with the magnetometer. The ISM330 and its surrounding components are shown in Figure 6.10.

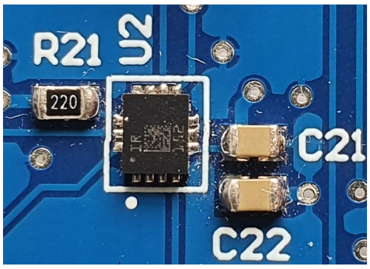


Figure 6.10: ISM330 and Surrounding Components

Gyroscopes are often used on satellites for attitude determination since the rotational movement of a satellite is important for controlling its attitude [46]. The gyroscope will be used to record the rotational velocity of BenchSat about its vertical axis during experiments using the air bearing facility. The accelerometer is used to measure the movement of BenchSat in the XY plane during experiments using the air bearing carrier cart.

Solenoid Control

The OBC controls the air bearing carrier cart through a header connection for four GPIOs. These GPIOs connect to the solenoid control board that directs power from the batteries to the solenoids on the air bearing carrier cart based on the GPIOs' states.

The solenoid control driver has two inputs: thrust duration and movement direction. The solenoid movement functionality is tested during the full system tests using the air bearing facility.

6.3.4. Functional Implementation

The OBC is the main controller for experiments using BenchSat. Since experiments will vary depending on the application, the OBC functional implementation is dynamic and can be changed based on user requirements. However, a framework can be defined for a basic test using the OBC.

During operation processes on the OBC can be triggered when:

1. A CAN message is received
2. A button is pressed
3. A timer expires

Based on these inputs the OBC has many options. These include:

1. Activating solenoid switches
2. Gathering and recording data from sensors

3. Commanding the RCS to send a beacon to the ground station
4. Storing received data in permanent storage

The OBC's capabilities can be expanded to use additional components and run more complex algorithms.

The OBC uses a super loop implementation to handle inputs and outputs. The super loop implementation means that the code executes sequentially and triggers are either monitored in the super loop or by interrupts. Although an RTOS is not required in this implementation, the chosen OBC MCU is capable of implementing an RTOS if future experiment parameters require it.

6.4. Testing

6.4.1. MCU Programming Test

An MCU programming test was used to verify that the OBC MCU and its peripherals are connected and working. The OBC was programmed to blink an LED every second, using the external crystal oscillator for timing. Figure 6.11 shows the OBC connected to a PC with an ST-Link programmer.

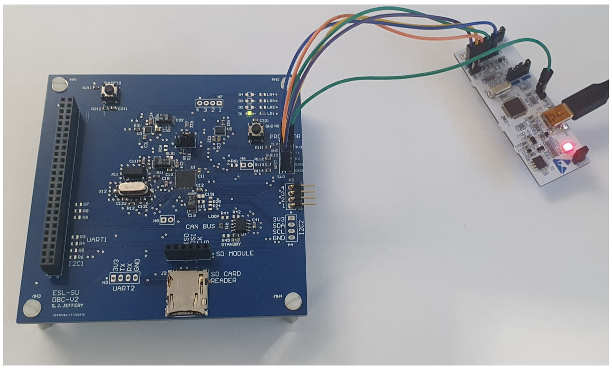


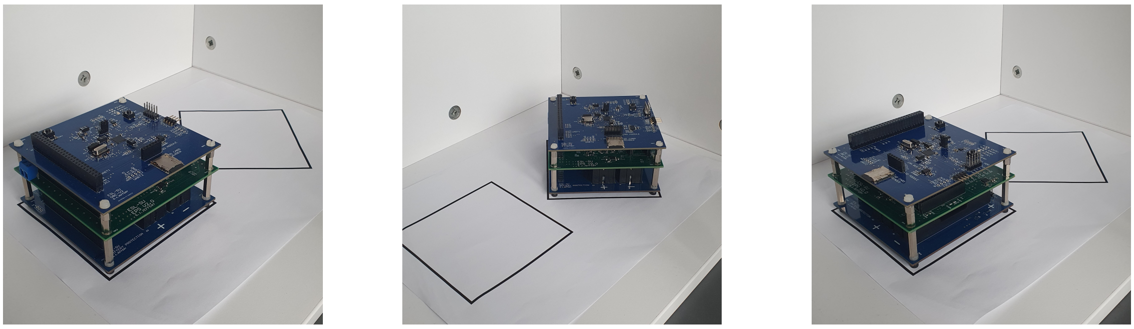
Figure 6.11: OBC Standalone Setup with ST-Link Programmer

This test confirmed that the MCU, power filtering, debugging setup and crystal oscillator are all functioning as expected.

6.4.2. Magnetometer Functionality Test

Setup and Background

The magnetometer functionality test demonstrates the accuracy and calibration of the OBC magnetometer. The OBC is rotated around its vertical axis as it would be on the air bearing carrier cart. The independent variable in the experiment is the absolute orientation that is calculated based on the OBC's placement. The dependent variable is the orientation measured by the compass. The SD card is used to record the data stored during testing. Measurements are taken at 45° intervals while the OBC is stationary. Figure 6.12 shows the first three positions of the test.



(a) Position 1 **(b)** Position 2 **(c)** Position 3

Figure 6.12: First three Magnetometer Test Positions

Results and Discussion

The test was executed for two full rotations and the positions were recorded relative to the initial direction. Since it is relative to the initial direction the error is zero in the first position. The results of the test are shown in Figure 6.13.

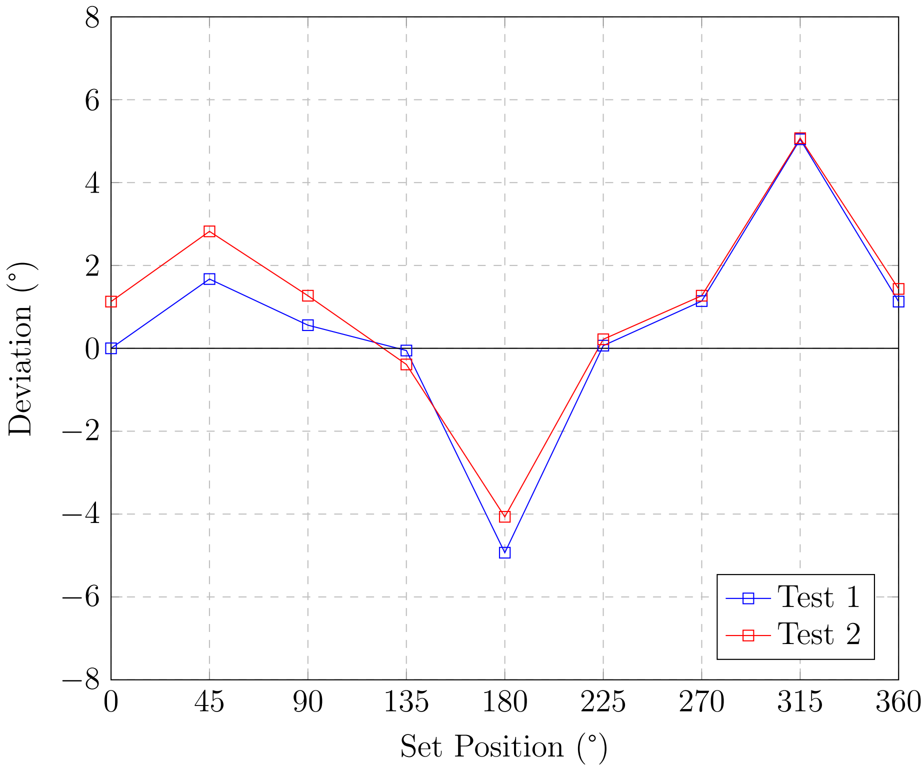


Figure 6.13: Magnetometer Orientation Output Deviation versus Position

At all points during these two rotations, the error is within 5° of the expected value.

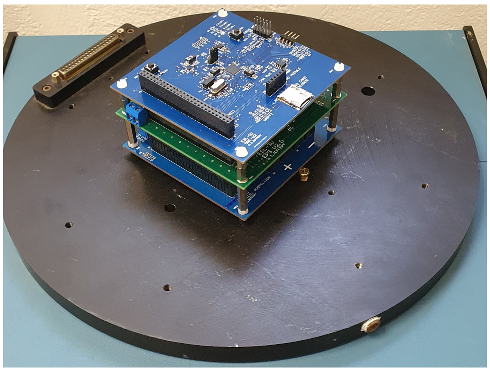
Heading accuracy of less than 5° is considered a low error value by [49] and is accurate enough to be useful on the air bearing facility.

Figure 6.13 shows that the errors in the two rotations follow a similar pattern. These errors could be due to soft-iron distortions, which are a result of materials that can influence the magnetic field while not producing a field [47]. Future work could involve improving the sensor to account for soft-iron distortions.

6.4.3. Gyroscope Accuracy Test

Setup and Background

The gyroscope accuracy on the BenchSat OBC is tested using a rate table that can be set to a chosen rotational velocity. The independent variable for the test is the rate table's rotational velocity and the dependent variable is the value measured by the gyroscope. The rate table is set to multiple values in both the clockwise and anti-clockwise directions. The system is calibrated to minimise error based on the recorded data. Three test points are chosen randomly to assess the system.



(a) OBC Test Setup on Rate Table

(b) Rate Table Display

Figure 6.14: OBC Gyroscope Test Setup

Results and Discussion

The gyroscope measured the rotational velocity with a high degree of accuracy. The accuracy tends to decrease as the rotational velocity increases with the least accurate measurements at the extreme ends of the test. All of the test points results are within 1% of the chosen speed.

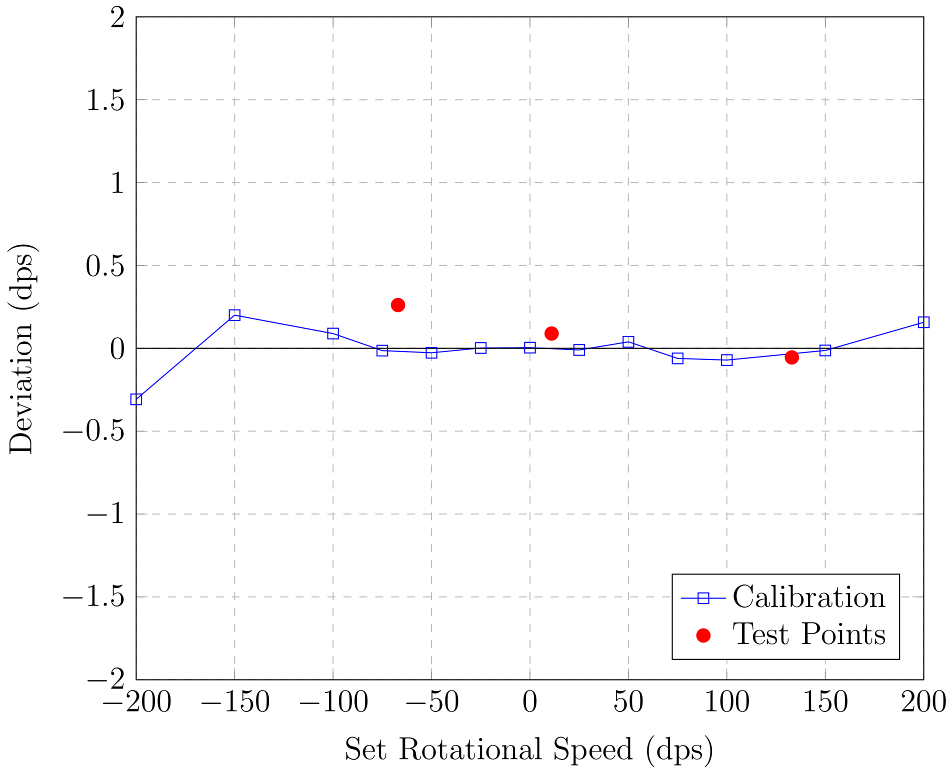


Figure 6.15: Gyroscope Output Deviation from Expected Value at Different Rotational Speeds

6.4.4. Accelerometer Accuracy Test

Setup and Background

The accelerometer accuracy test uses gravitational acceleration to evaluate the accelerometer's ability to detect and measure acceleration. The test is varied by placing the OBC in different stationary positions, which changes the orientation of the accelerometer. The test's independent variable is the expected acceleration based on the gravitational acceleration in different positions. The dependent variable is the measured acceleration by the accelerometer.

Results and Discussion

The acceleration recorded had a maximum error of $0.1\,\mathrm{m\,s^{-2}}$. From a similar inclination test in [50] a total error of $22\,\mathrm{mG}$ (or $\pm\,0.2\,\mathrm{m\,s^{-2}}$) was deemed acceptable.

The onboard accelerometer is accurate enough to provide useful feedback to a user who is testing components on the air bearing facility.

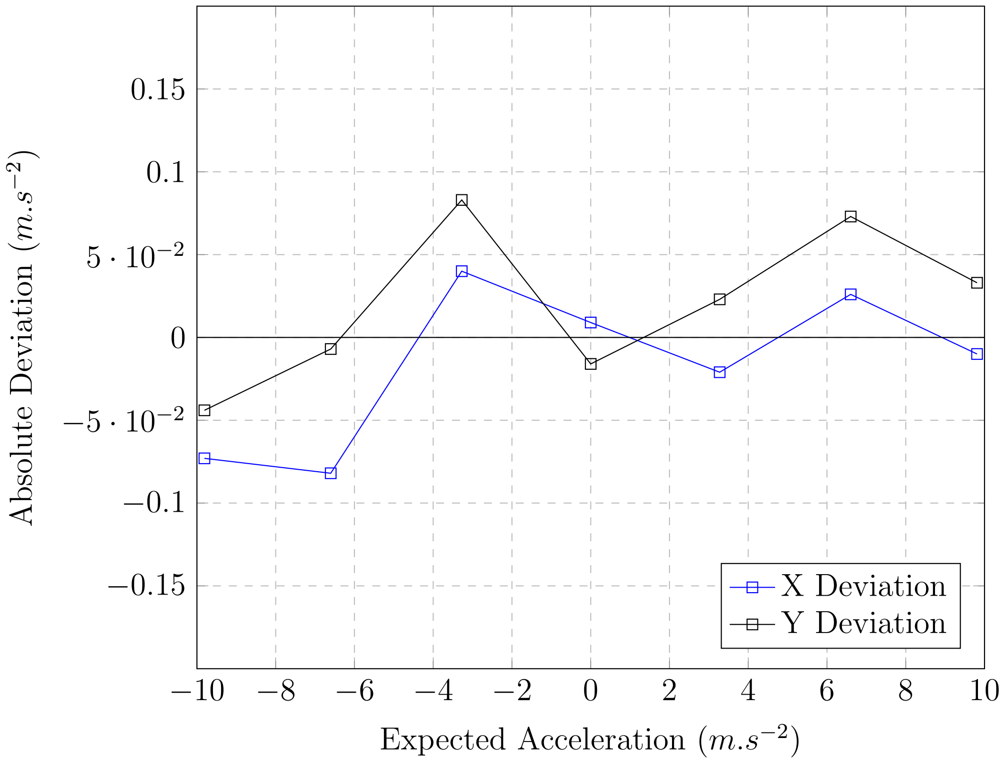


Figure 6.16: Accelerometer Output Deviation from Expected Output in Different Positions

6.4.5. SD Card Write Test

Setup and Background

The SD card interface on the OBC was tested by writing to a .csv file during the gyroscope test in Section 6.4.3. This test is to demonstrate that the SD card writer can be reliably used during BenchSat experiments.

Results and Discussion

The SD card writer is successfully able to write to a .csv as shown in the results in Figure 6.17.

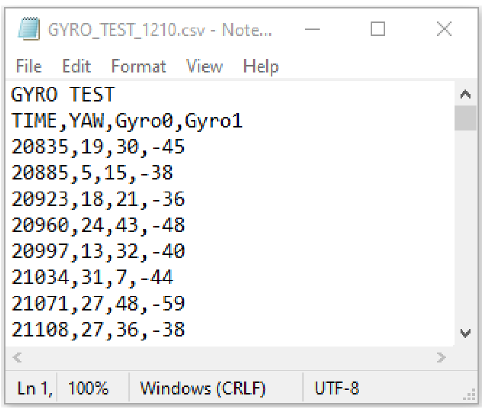


Figure 6.17: Recorded Data using the SD Card Writer

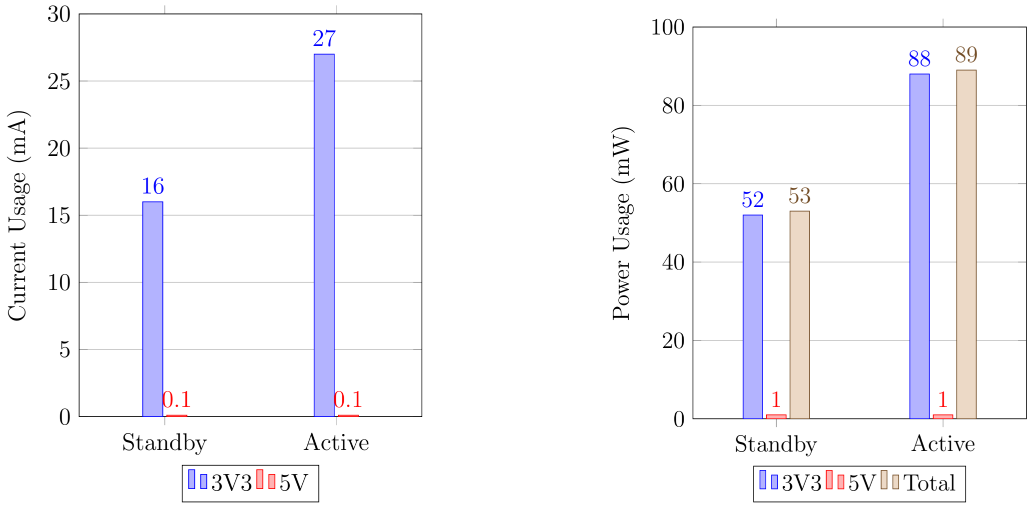
6.4.6. OBC Power Test

Setup and Background

A power usage test is used to measure the power used by the OBC during normal operation. This result is needed to quantify BenchSat's power needs. The EPS is used to measure the amount of current that is being supplied to the OBC during this test. The OBC is tested in two configurations. Firstly in standby mode with only core functions running and secondly in active mode where the SD card, IMU and magnetometer are used.

Results and Discussion

The OBC standby and active current and power usage data are summarised in Figure 6.18.



(a) OBC Current Usage

(b) OBC Power Usage

Figure 6.18: OBC Power Test Results

Since almost all of the components on the OBC use 3V3 power, the 5V power use is extremely low. It is noted that a benefit of not using a module for the operation of the OBC is lower power use. This benefit is demonstrated in this test where the total power usage on the OBC is 53 mW in standby mode. In contrast, the RCS processor uses 79 mW in standby mode as shown in Section 5.4.5. The total power usage on the OBC during active functioning is 89 mW which is well within the limits defined in Section 6.2.1.

6.5. Summary

This chapter introduced the concept of a satellite OBC. The background section covered other OBCs, including COTS options. The design and implementation of the BenchSat OBC were then discussed, including the requirements and components used. A focus was placed on the functionality of the BenchSat OBC as those features are part of what makes the BenchSat OBC useful during experiments.

The OBC functionality was thoroughly tested and the results are shown in Section 6.4. Each component on the OBC was shown to be functioning as designed and the OBC movement sensors accurately recorded movement. The movement data will be used during final system testing to provide feedback on BenchSat's movements around the air bearing facility.

The OBC can be improved with future iterations of the design. The chip used for the OBC processor was chosen when there was a severe supply shortage of microcontroller chips. A more suitable chip either with a stronger processor or lower power usage could be selected depending on the desired application.

The OBC requires an external ST Link to be programmed so it is programmed in a similar way to a development board. A future iteration of the OBC could upgrade to use a USB connection for programming. USB is a more advanced connection that would simplify the connection process and remove the need for an additional programmer such as an ST Link.

This chapter completed the discussion of BenchSat subsystems. The next chapter will cover the implementation of the ground stations that interact with BenchSat.

Chapter 7

Ground Station

This chapter covers the two ground stations that are used in this project and the relevant software to handle radio communication signals. The methods used to transmit and receive signals are covered for both ground stations.

7.1. Architecture

BenchSat is expected to communicate with a ground station to simulate a real flight test. The BenchSat radio communication subsystem (RCS) was discussed in Chapter 5 and the ground stations connect to that subsystem to communicate with BenchSat.

Figure 7.1 shows a breakdown of the radio frequency connection between a ground station and BenchSat. On the right side of the figure, the BenchSat RCS components are filled in as they were discussed in Chapter 5. The left side represents the ground station. The sections where the components are discussed are marked in the figure.

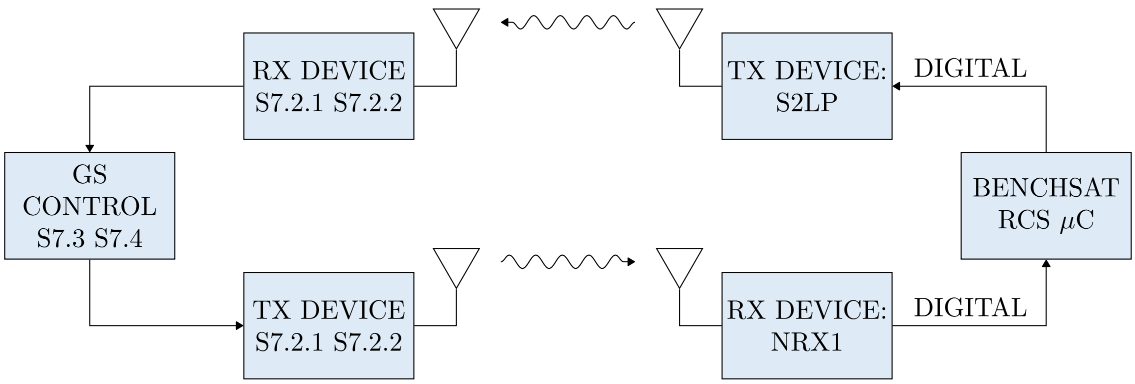


Figure 7.1: Radio Frequency Connection between a Ground Station and the BenchSat RCS

7.1.1. Requirements

The requirements for the communication link were defined in Section 5.2.1. The ground station is required to implement frequency-modulated, full-duplex communications with the BenchSat RCS at a baud rate of 9.6 kbps. The ground station should receive in the UHF frequency band and transmit in the VHF frequency band. The existing TIM Ground Station has existing infrastructure which was a consideration for the stated requirements.

7.1.2. Software Defined Radio

Software-defined radio (SDR) is a method of radio communication that uses PC software to perform the modulation and demodulation of radio signals [51]. The two relevant programs are SDR-Sharp (SDR#) and GNU Radio.

SDR#

SDR# is a PC-based digital signal processing (DSP) program for software-defined radios. It is generally used for proof of concept and allows easy access to DSP for a user [52].

The software supports many types of standard receiver hardware, including both the FUNcube Dongle Pro+ (FCDPP) and the RTLSDR. The software includes narrow-band frequency modulation (NBFM) and demodulation, which can be used to demodulate received radio frequency signals and output the baseband signal.

SDR# can be used to control multiple variables to ensure that the signal is received correctly. Notable variables that are controlled include frequency, gain and squelch. The frequency is chosen to match the frequency transmitted by the RCS, gain increases the power of the received signal, and squelch blocks low-power noise from interrupting the signal.

GNU Radio

GNU Radio is radio control software that can be used to control radio signals. It is a free, open-source software system that is used by amateurs and professionals to implement software radios. It is highly customisable allowing for unique flow charts.

GNU Radio works by building flowcharts out of predefined blocks and embedded Python blocks. Flowcharts are groups of connected blocks. Blocks perform varying tasks in the program flow to achieve the desired goal.

GNU Radio can interface with hardware components in various ways. GNU Radio can connect to a PC's audio connection and SDR devices with predefined blocks.

7.2. Implementation

The ground station will be implemented on two levels:

- The existing TIM Ground Station
- An ESL Ground Station

As stated in Section 1.2.2, one of BenchSat's goals is to interact with the existing TIM Ground Station. This connection would allow experiments with BenchSat that are controlled from the TIM Ground Station to be performed just as they would be during a satellite mission in space.

In addition to the TIM Ground Station it would be useful for the ESL to have a smaller ground station that can be used for local experiments on BenchSat. The TIM Ground Station is used to monitor active missions. This activity means it will not always be available to a BenchSat user. Therefore an ESL Ground Station that is dedicated to being used with BenchSat would be beneficial to the development of radio communication systems.

The ESL Ground Station should emulate the TIM Ground Station as closely as possible so that systems can be ported between the stations with ease. Ideally, communications will be developed on the ESL Ground Station and then ported to the TIM Ground Station for live testing.

7.2.1. TIM Ground Station

The capabilities of the TIM Ground Station are stated in [5]. Relevant to this project are the devices the system uses for receiving and transmitting radio frequency signals. The facility uses a Kenwood TS2000 radio to transmit and a FCDPP to receive. The devices are shown in Figure 7.2.

(a) Receiver: FUNcube Dongle Pro+ **(b)** Transmitter: Kenwood TS2000

Figure 7.2: TIM Ground Station Hardware

Both of these devices are connected to directional antennae that can be programmed to track satellite missions. The TIM Ground Station radio communication capabilities are controlled by a PC located in the ground station and can be accessed remotely. The PC connects to the Kenwood over audio that is generated by GNU Radio. The FCDPP connects with a USB plugin to connect to SDR#. This interaction is shown in Figure 7.3, which demonstrates how the TIM Ground Station hardware and software interact.

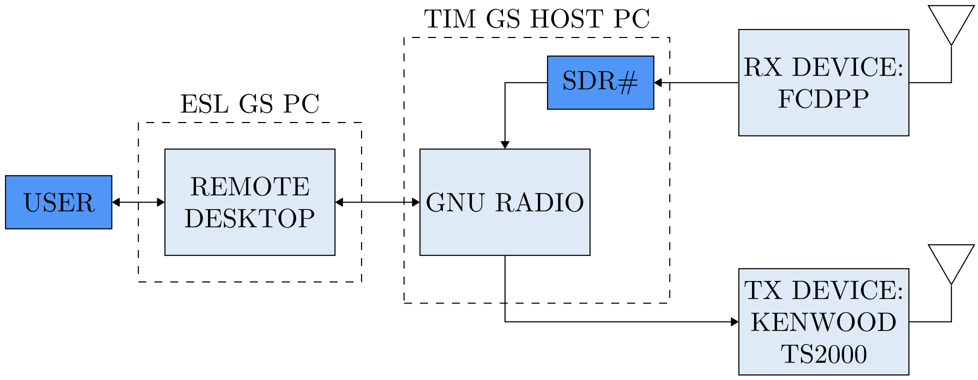


Figure 7.3: TIM Ground Station Connections

7.2.2. ESL Ground Station

The ESL Ground Station is made up of a dedicated PC that connects to a transmitter and a receiver. Due to the proximity to the test area, small antennas are used for the devices. The facility uses a NI USRP-2901 to transmit outgoing messages and an RTLSDR to receive incoming messages. The devices are shown in Figure 7.4.

(a) Receiver: RTLSDR

(b) Transmitter: NI USRP-2901

Figure 7.4: ESL Ground Station Hardware

Both devices connect to a PC via USB. The ESL Ground Station is controlled through the software interfaces onboard the PC. The RTLSDR can be used with both GNU Radio and SDR#. The USRP has a dedicated block in GNU Radio that can be used for transmitting radio frequency signals. These connections are shown in Figure 7.5, which demonstrates the way the ESL Ground Station hardware and software interact.

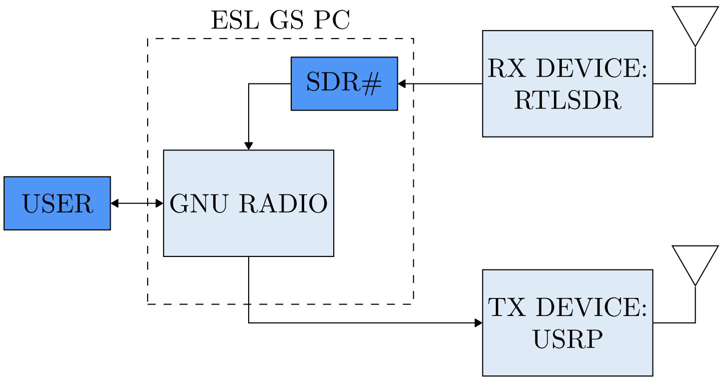


Figure 7.5: ESL Ground Station Connections

7.2.3. Interface

The interface to both ground stations has two layers. Firstly there is a graphical user interface (GUI) in GNU Radio. The GUI is used to input commands and display received messages. Using visual display blocks in GNU Radio this interface can also be used to debug flowcharts. The second layer is the Python interface. This can be used to send predefined messages at given times or to record received data to files.

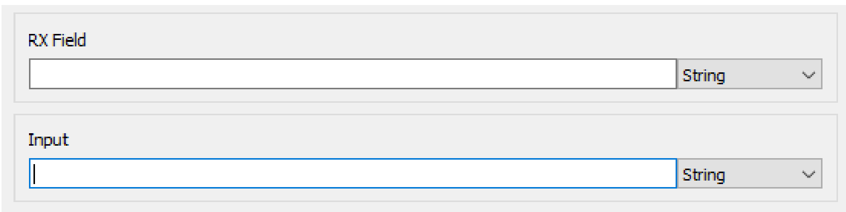


Figure 7.6: GNU Radio GUI Features

7.3. Transmission (TX)

Transmitting from the ground station can be broken into two parts. The first part is generating a baseband signal that includes all the information that will be transmitted. The second part is the modulation and transmission of the baseband signal over radio frequency communication channels. The two ground stations use the same method to generate the baseband signal, but different methods to transmit the signal due to the different hardware implementations. Figure 7.7 shows a combined view of the methods for transmitting on both ground stations.

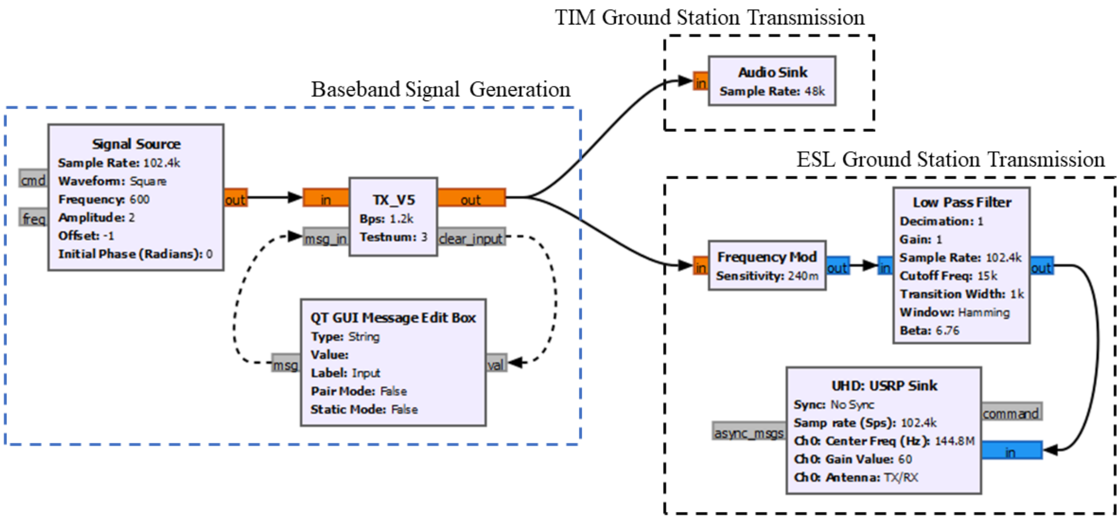


Figure 7.7: GNU Radio Transmit Flow Chart

7.3.1. Baseband Signal Generation

The TX_V5 block is the primary controller of the signal generation system. The TX_V5 block is a synced embedded Python block. This type of GNU Radio block allows a user to write Python code to execute a required task while still following the normal GNU Radio flow [53]. In the ground station context, this block encodes the binary data to be transmitted.

The TX_V5 block has two inputs and two outputs depicted in grey and orange. Firstly, the grey input and output that are connected to the Message Edit Box are related to the user interface that is shown in Section 7.2.3. The grey input to the TX_V5 block allows a user to supply commands by typing into the input field. The grey output from the TX_V5 block clears the input field once a command has been recorded. All the commands are stored for later analysis.

Secondly, the orange input to the TX_V5 block is a repeating square wave from the Signal Source block. The repeating square wave is shown in Figure 7.8.

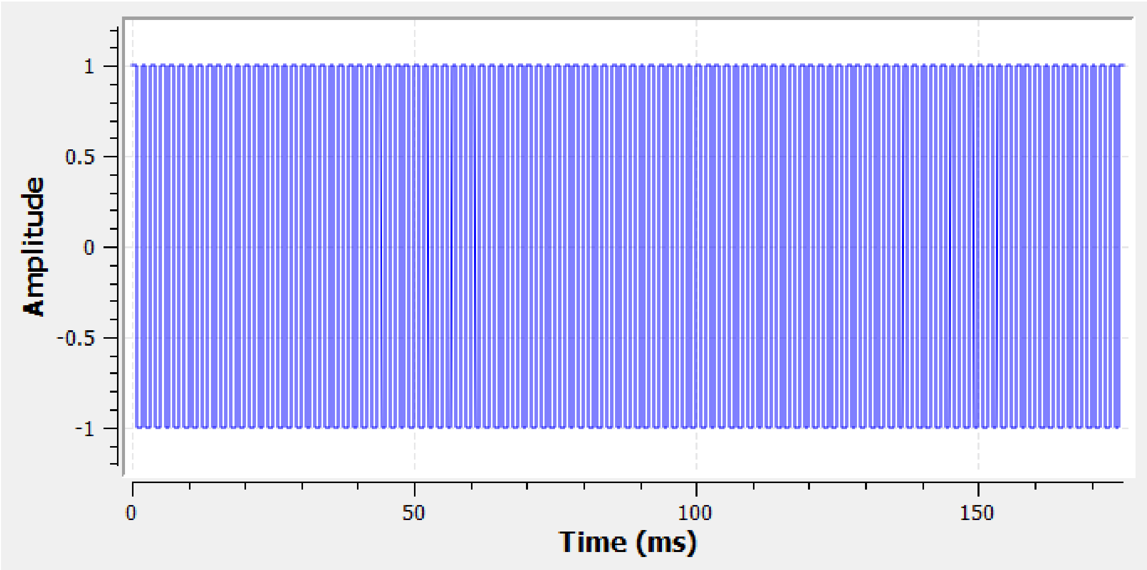


Figure 7.8: Signal Source Output: Repeating Square Wave

This repeating signal is used to set the baud rate since it outputs consistently at a chosen frequency. The TX_V5 block uses the rising and falling edges to define the start and end of bits. The baseband signal baud rate is twice the square wave frequency because both edges are used to define the boundaries between bits.

To send a transmission, the TX_V5 block implements the packet handling that was discussed in Section 5.3.4. Transmission data is encoded into a packet. Variables such as the flag, preamble, postamble and frame check sequence are defined within the TX_V5 block. Once the packet has been generated it is encoded onto the incoming signal resulting in the baseband signal shown in Figure 7.9.

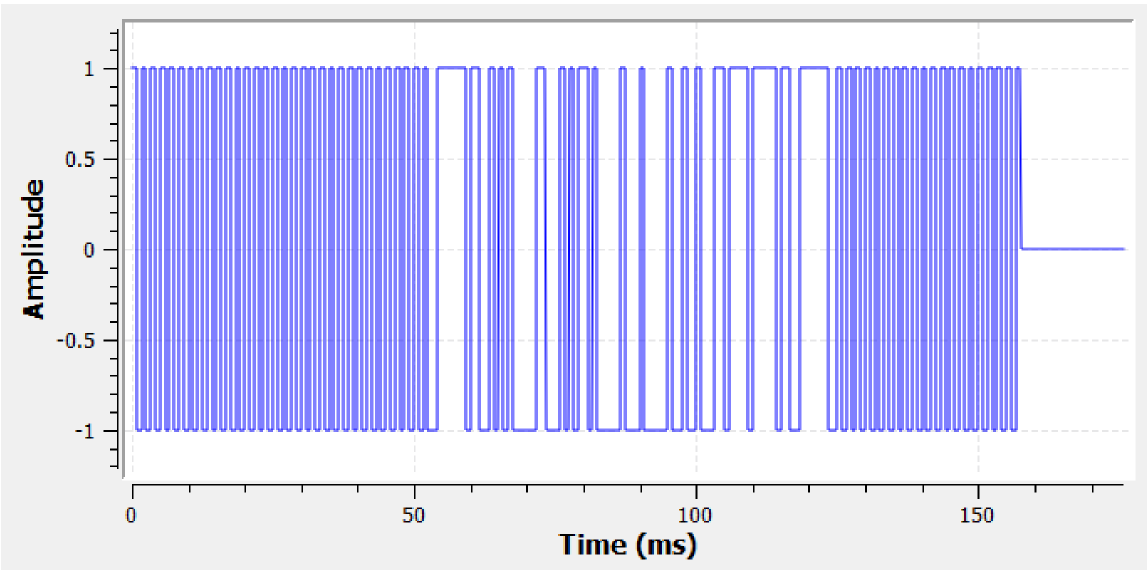


Figure 7.9: Baseband Signal with Embedded Data for Transmission

The baseband signal is passed to the transmission phase of the flowchart in Figure 7.7

where the method depends on which ground station is being used.

7.3.2. Message Modulation and Transmission

The signal is transmitted using different methods depending on the active ground station.

ESL Ground Station

The ESL Ground Station uses GNU Radio for all processing to send a transmission. The blocks used in this processing are depicted in the box marked "ESL Ground Station Transmission" in Figure 7.7. GNU Radio modulates the signal with the Frequency Mod block. The frequency deviation of the transmission is controlled by the magnitude of the input signal and a sensitivity variable. These values have been tuned for the best results when communicating with the BenchSat RCS. Figure 7.10 shows the frequency modulated output signal when the embedded signal generated in Section 7.3.1 is the input.

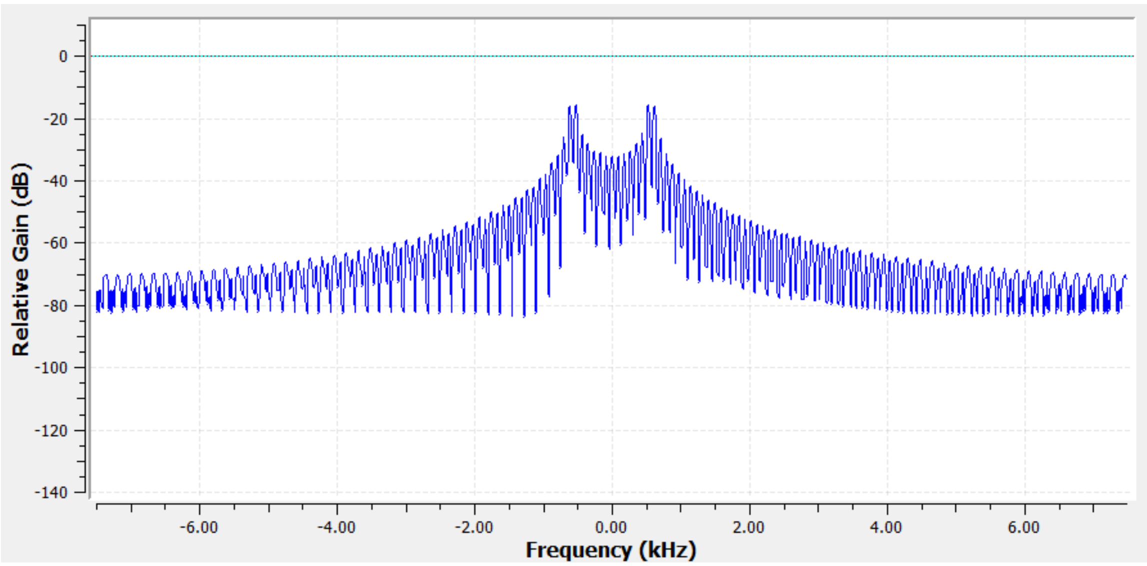


Figure 7.10: Frequency Modulated Baseband Signal

The frequency-modulated output is then filtered to reduce noise that is not useful to the signal. The Low Pass Filter block in Figure 7.7 filters the frequency modulated input to output the signal shown in Figure 7.11.

The filtered signal is then passed to the USRP Sink block. This block controls the USRP device to set the gain and the centre frequency. The centre frequency is set to 144.8 MHz to match the receiver on the BenchSat RCS. The USRP Sink block is the end of the flowchart and it controls the NI-USRP 2901 to transmit the signal to BenchSat.

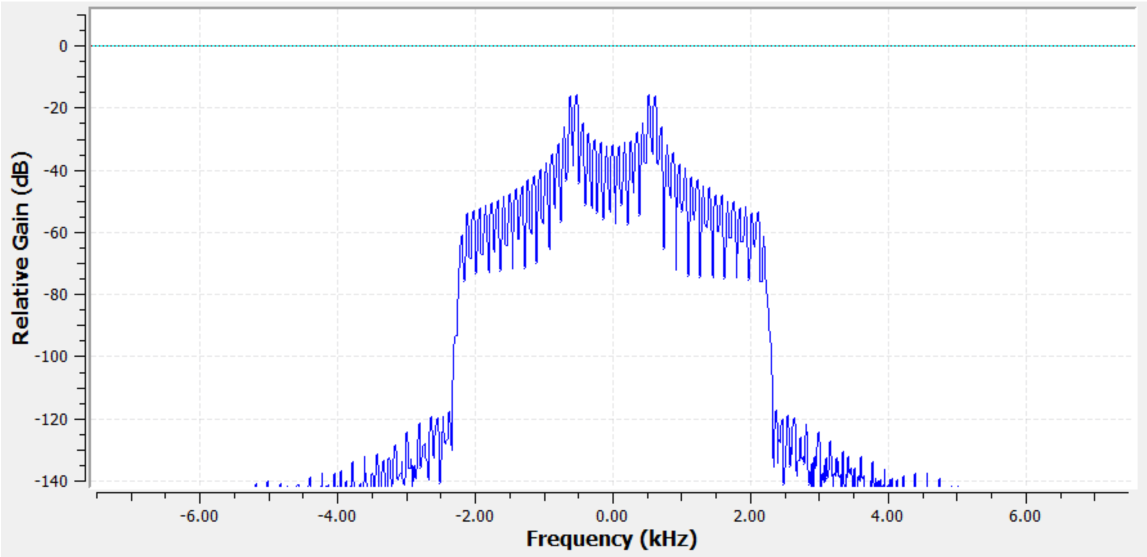


Figure 7.11: Filtered Frequency Modulated Baseband Signal

TIM Ground Station

The only GNU Radio block in the box labeled “TIM Ground Station Transmission” in Figure 7.7 is an Audio Sink block. For the TIM Ground Station, the baseband signal is directed to the audio sink where the information is handled by hardware from that point onwards.

The baseband signal is passed via a USB audio link to the Kenwood TS2000, mentioned in Section 7.2.1. The Kenwood TS2000 defines the gain and centre frequency of the transmission. These values can be controlled using the dedicated ground station software installed on the TIM Ground Station PC.

The ground station hardware controls modulation and transmission to achieve the same outcome as the ESL Ground Station. Although the TIM Ground Station transmission inputs are more opaque than the ESL Ground Station it is important to achieve the same results since the TIM Ground Station has the additional functionality required to communicate with active satellites. The ESL Ground Station is beneficial in this situation as it can be used to gain insight into what the TIM Ground Station hardware is achieving.

7.3.3. Transmission Results

The transmission results of both ground stations are compared in Figure 5.11 in Section 5.4.2. The ESL Ground Station was able to accurately transmit for all baud rates up to 10 kbps. The TIM Ground Station was able to transmit at baud rates between 3 kbps and 9.6 kbps.

Both systems can transmit at 9.6 kbps, which demonstrates that the transmission systems on both ground stations are working correctly. A drawback of the TIM Ground Station is that the system does not pick up baud rates lower than 3 kbps so the system is unable to transmit at that rate. It does not pick up the slower signals because the system stops transmitting if the line stays constant for too long. Bit stuffing and higher baud rates are therefore required to stop the transmitter from going dormant.

7.4. Receiver (RX)

The receiving side of the ground station can also be broken into two parts. The first part is to receive the radio frequency signal and demodulate it to retrieve the baseband signal. The second part is decoding the baseband signal. It is possible to execute this whole process in GNU Radio; however to fully utilise the existing setup in the TIM Ground Station the parts are split between GNU Radio and SDR#. The implemented method allows both ground stations to utilise the same system for receiving messages so that the ESL Ground Station can function exactly like the TIM Ground Station.

7.4.1. Signal Receiving and Demodulation

SDR# is used to receive and demodulate the incoming signal. SDR# has built-in drivers for the receive hardware used on both ground stations, namely the FCDPP and the RTLSDR. Figure 7.12 shows a message being received from BenchSat with SDR#.

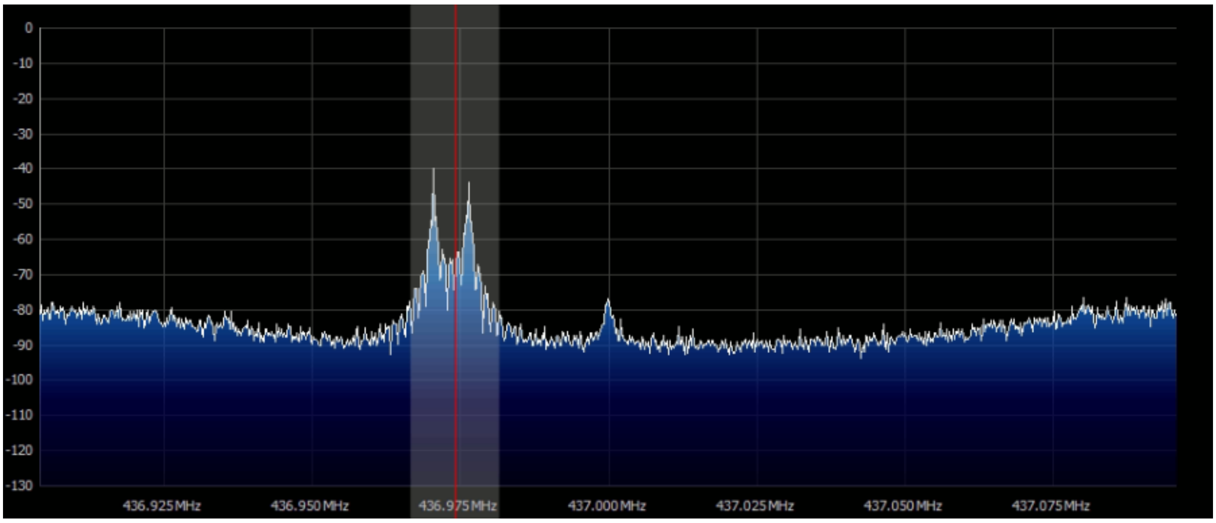


Figure 7.12: SDR Sharp Receive

In Figure 7.12, the red line marks the chosen centre frequency and the highlighted area is the bandwidth. The system is set to a centre frequency of 436.975 MHz and bandwidth of 15 kHz. The squelch is tuned to block out all background noise that is not part of a message from BenchSat.

SDR# demodulates frequency-modulated messages and outputs baseband signals onto an audio output that can be received by GNU Radio.

7.4.2. Baseband Signal Decoding

Figure 7.13 shows the GNU Radio receive flowchart used to decode incoming signals.

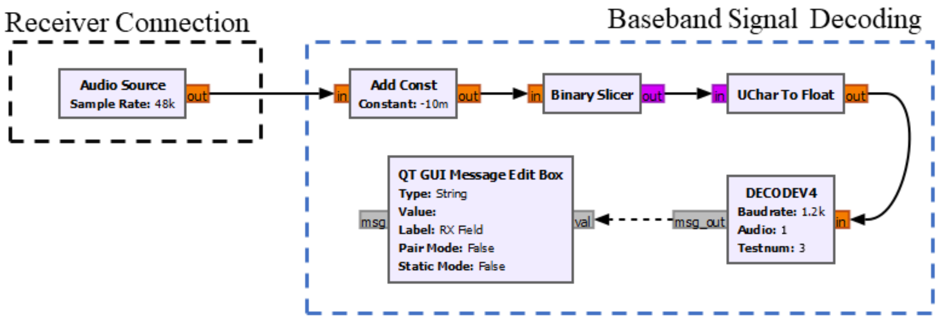


Figure 7.13: GNU Radio Receive Flowchart

The "Baseband Signal Decoding" box receives the demodulated audio signal from SDR# from an Audio Source block in GNU Radio. This Audio Source block makes up the "Receiver Connection" box in Figure 7.13 and it provides the input to the rest of the flowchart. The demodulated audio signal that is received from SDR# is shown in Figure 7.14.

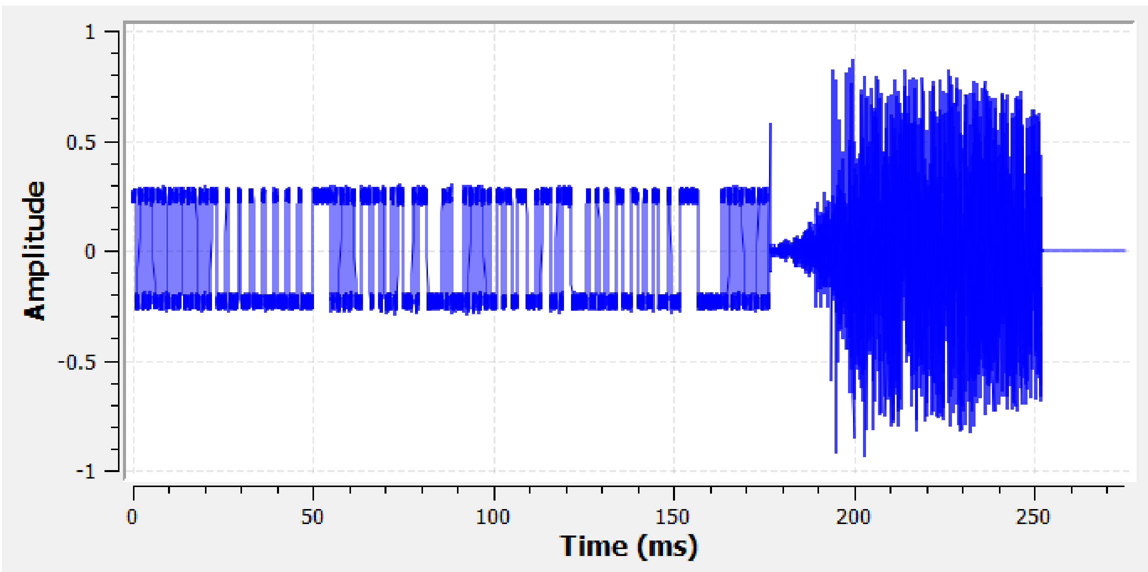


Figure 7.14: Demodulated Audio Signal received from SDR#

Individual bits can be identified in the incoming signal shown in Figure 7.14 but processing is required to decode the information. Processing is performed within the "Baseband Signal Decoding" box. The Add Const block fine-tunes the signal to ensure that the binary slicer receives the correct information. The signal is then cleaned to clearly define the signal into binary options using the Binary Slicer block. The output from the binary slicer is a character output so the signal is turned back into a usable format by the UChar to Float block. The cleaned binary signal is shown in Figure 7.15.

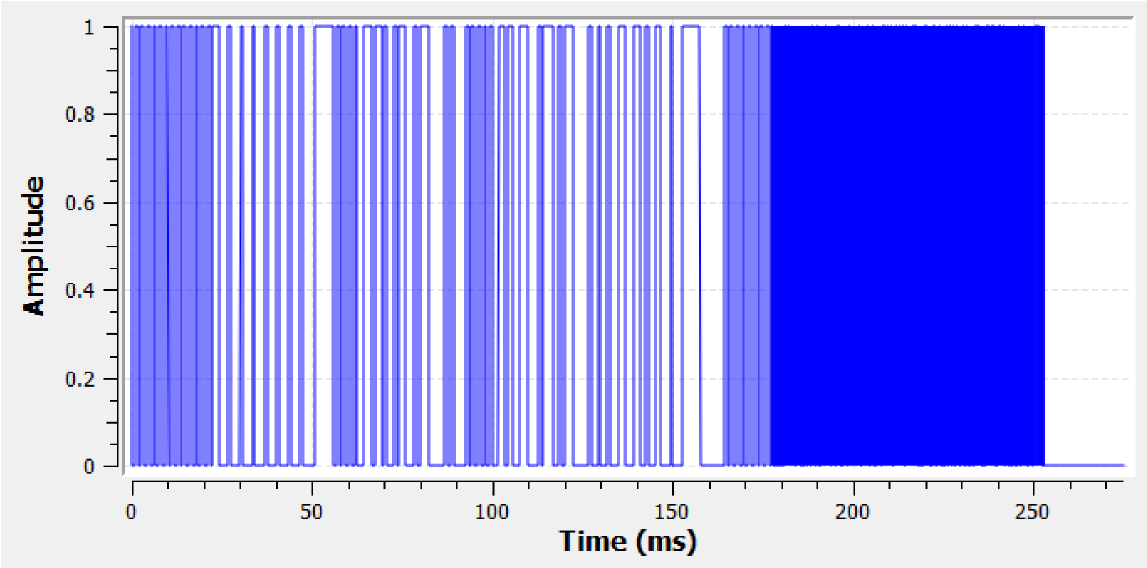


Figure 7.15: Cleaned Binary Signal

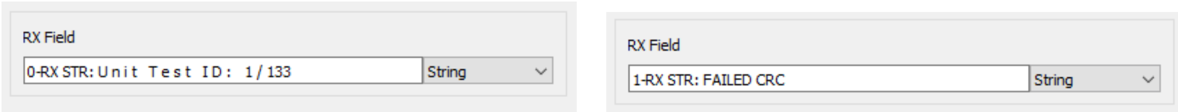
The DecodeV4 block receives the binary signal as a sequence of samples where each bit lasts multiple consecutive samples. The length of a single bit can be found with

$$l = \frac{f_s}{b} \tag{7.1}$$

where l is the length of a bit in samples, f_s is the sample rate and b is the signal baud rate. Using the indices of where the signal changes its value and the length of a bit, it is possible to generate a binary array containing all of an incoming message's information.

While a signal is being received the program saves all the index values. Once a message is complete the program creates and processes the binary array. Decoding a message from BenchSat includes identifying the start and end flags, checking the frame check sequence and removing extra bits created during bit stuffing on the transmit side. If a processed message passes all of its checks it can be displayed on the GUI and saved to storage for later analysis. The noise at the end of the signal is ignored because it comes after the end flag.

The ground station supplies basic feedback directly to the user through the GUI. If the message passes all of the system checks the result is displayed as depicted in Figure 7.16a; otherwise the error is displayed as depicted in Figure 7.16b.



(a) Unit Test Message Output **(b)** Incorrect Message Output

Figure 7.16: Examples of Receiver Feedback

7.4.3. Receiver Results

The receiver on both ground stations was tested in the RCS transmit tests as discussed in Section 5.4.1. At all of the baud rates, both ground stations can receive the signal with 100% effectiveness. During testing, the receivers of both ground stations functioned as required and were able to decode received messages.

7.5. Summary

This chapter covered the design, implementation and use of two ground stations that can be used to interact with BenchSat. The first part of the chapter covered the background of the ground station implementation and introduced the concept of two ground stations. The hardware and the software used to implement the ground stations were also discussed. The chapter then explained the methods for transmitting and receiving messages in depth.

The ground stations implemented to communicate with BenchSat are considered a success as they can communicate with 100% consistency at the goal baud rate of 9.6 kbps. However, both the transmit and receive implementations can be improved in future work.

For the transmit implementation, future work should include enabling the TIM Ground Station to transmit at baud rates below 3 kbps. To do this, the hardware used to transmit will need to be updated. The goal should be to make the workings of the TIM Ground Station more transparent to other users.

For the receiver implementation, the main drawback of the system is that it needs the incoming audio to be clipped at the start and the end to work. An example of this clipping is shown in Figure 7.14 where the line returns to zero after the noise subsides. Future work should include improving the system to be able to constantly receive audio and identify flags that show when a message is being received. This upgrade would enable the system to handle more noise and not require the squelch on SDR# to be tuned as finely as it currently needs to be.

This chapter covered the final component of the BenchSat system architecture. The next chapter will cover system integration and demonstrations of BenchSat functioning together with the air bearing facility and the ground stations.

Chapter 8

System Integration and Evaluation

In the previous chapters, the BenchSat subsystems and surrounding components have been defined and developed. This chapter focuses on the integration of the BenchSat subsystems and the surrounding components. The integrated system is then used in multiple demonstrations to show its functionality.

8.1. System Integration

This project aims to narrow the gap between an experimental design and a component that can be tested and flown onboard a satellite by creating a system that can emulate satellite capabilities and interactions in a controlled environment on the ground.

BenchSat, a bench satellite system that is aimed at development and testing is the outcome of this project. Chapter 3 introduced and discussed the design of the BenchSat system. The design process resulted in a modular system made up of three subsystems, namely an EPS, an RCS and an OBC. The final stage of this project is to integrate the three subsystems to create BenchSat. Once the subsystems have been integrated, BenchSat can be tested with the air bearing carrier cart and the TIM Ground Station to demonstrate that it achieves the functionality described in the project scope.

System integration is the process of connecting different subsystems to create a single larger system. The BenchSat subsystems are connected in a stack. The subsystems in the stack are integrated on top of one another and connections are made through a shared header that connects all of the subsystems in the stack. Due to the modularity of the subsystems, the number of connections between the systems is limited to system power channels and a CAN-bus.

Each subsystem is tested to confirm its standalone functionality but during integration it is important to confirm that power is distributed correctly and that the primary communication channel, the CAN-bus in this case, is working correctly. LEDs are used during integration to confirm that the connections are working.

Firstly, all of the systems blink LEDs during normal functioning to show that they are connected to power and functioning correctly. If they do not blink or blink at the incorrect frequency it shows that there is an issue.

Secondly, the CAN driver includes a unit test to check that it is functioning as expected. The test involves transmitting messages, containing a frequency and a number of pulses, between subsystems. If the subsystems accurately display the data using an LED, then the communication system is functioning correctly. The test is initiated by a button on the OBC and it involves all connected subsystems.

The set of steps to integrate the subsystems is listed below. These steps reduce risk and test functionality wherever possible.

1. To start the integration the EPS is connected to a power supply. Using a power supply instead of the EPS batteries allows a current limit to be set. This limit protects the system if a component is connected incorrectly or causes a short circuit. If the LED on the EPS begins to blink at the correct frequency it is assumed that the EPS is working correctly and integration can continue.

2. The OBC is the second subsystem to be added to the stack. Once integrated, if the LED begins to blink at the correct frequency, it can be assumed that power has been distributed correctly and that it is functioning as expected. The CAN-bus connection between the two systems is tested by pressing a button on the OBC to run a CAN-bus unit test. If both the power distribution and CAN-bus are functioning correctly integration can continue.

3. The RCS is the third and final subsystem to be added to the stack. As with the OBC, if the LED on the RCS begins to blink at the correct frequency once the RCS is connected it is assumed that the RCS is receiving power and functioning correctly. Once again a button on the OBC is pressed to run the CAN unit test to ensure that messages are shared between all three subsystems. This confirms that the CAN-bus is functional.

4. The fourth step is to disconnect BenchSat from the power supply and switch the EPS to use battery power.

5. BenchSat is active once all three subsystems are present and functioning with no external power supply. The integrated subsystems comprising BenchSat are shown in Figure 8.1.

Figure 8.1: Integrated BenchSat: EPS Battery PCB (Bottom Layer), EPS Power Conditioning PCB (Second Layer), OBC (Third Layer), RCS (Top Layer)

6. BenchSat is then connected to the air bearing carrier cart and either the TIM Ground Station or the ESL Ground Stations to expand functionality.

7. To integrate BenchSat with the air bearing carrier cart, BenchSat is placed on the air bearing carrier cart payload platform. Power from the EPS and the solenoid switch outputs on the OBC are connected to the solenoid control board.

8. The connection between BenchSat and the air bearing carrier cart is tested by firing all four actuators in succession while the air bearing carrier cart is immobile. This test confirms the power delivery to the air bearing carrier cart solenoids and confirms that each of the switch outputs is connected to the correct solenoid.

9. The RCS is active when BenchSat is powered on. The VHF antenna is attached to the base of the air bearing carrier cart, which functions as a ground plane.

10. A unit test is sent to the RCS from the ESL Ground Station. The RCS then responds with an acknowledgment to confirm that the message was received. If this communication is successful, the process is repeated using the TIM Ground Station.

11. BenchSat is then fully integrated and the combination of BenchSat and the air bearing carrier cart will be referred to as the compound satellite, as it was in Section 3.4. The compound satellite is shown in Figure 8.2.

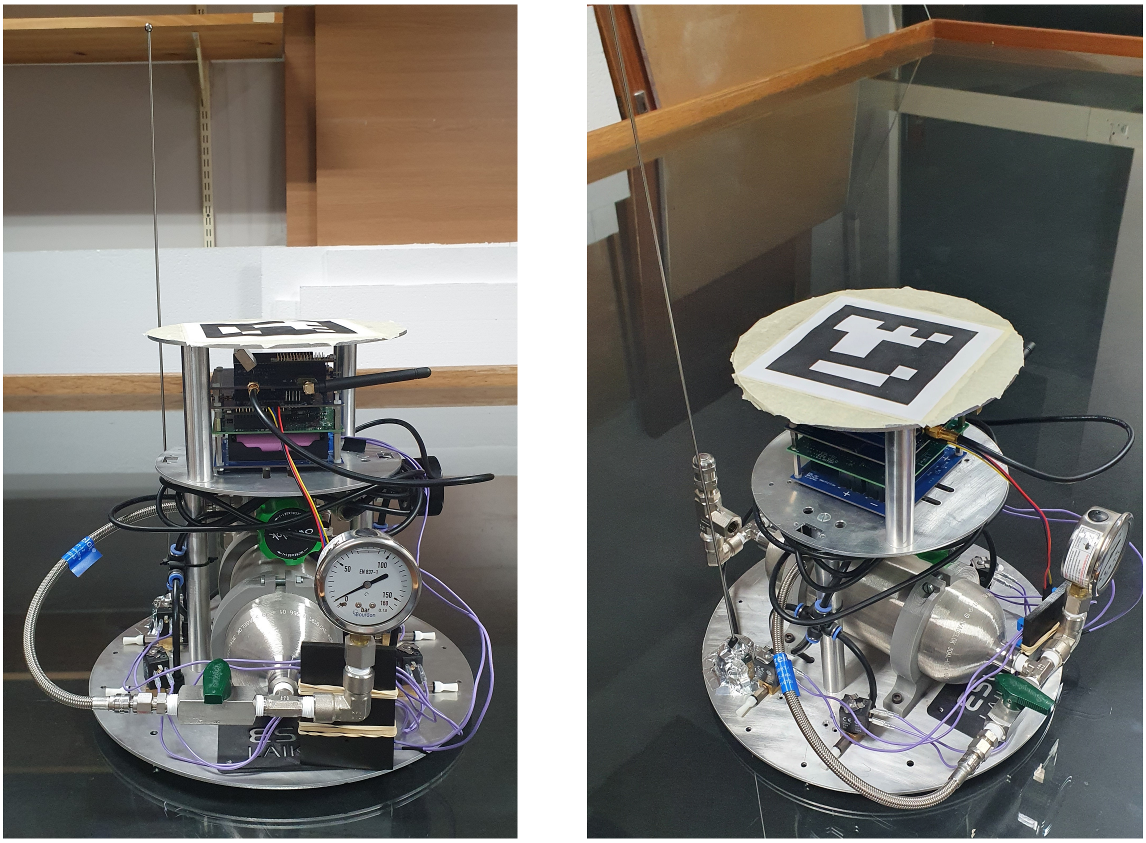


Figure 8.2: Compound Satellite: BenchSat Connected to the Air Bearing Carrier Cart

8.2. Demonstrations

8.2.1. Desired Demonstration Outcomes

The purpose of this section is to demonstrate the capabilities of BenchSat when it is integrated with the air bearing carrier cart and connected to either the TIM Ground Station or the ESL Ground Station. The capabilities that will be demonstrated are as follows:

1. Control experiments while moving through multiple states.
2. React to commands from a ground station during an experiment.
3. Gather and store data from attached sensors.
4. Demonstrate internal communication on BenchSat using the CAN-bus.

5. Provide power to the satellite bus, additional payloads[1] and the air bearing carrier cart.
6. Communicate with both the TIM Ground Station and the ESL Ground Station.
7. Use the actuators to move the air bearing carrier cart forward.
8. Use the actuators to rotate the air bearing carrier cart.

These capabilities are the desired demonstration outcomes and they demonstrate the functionality defined in the project scope and objectives in Section 1.5.

8.2.2. Demonstration Overview

Three demonstrations are used to cover the functionality listed in Section 8.2.1. These demonstrations are: a Spin Test, where the compound satellite executes a multiple step process that involves rotation based on a single command; a Forward Test, which is similar to the Spin Test except it involves forward movement; and a Command Test, where the compound satellite executes both rotational and forward movement based on multiple direct commands from a ground station. Each demonstration is repeated to make use of the TIM Ground Station and the ESL Ground Station.

The information recorded during the demonstrations is constant. It includes gyroscope, accelerometer, and magnetometer readings onboard BenchSat, communication logs between BenchSat and the active ground station, and image feedback from the overhead camera. During each demonstration, BenchSat transmitted beacons to the active ground station every five seconds. Although many of the features overlap between the demonstrations, each setup and the discussion that follows highlights different compound satellite capabilities and functions.

8.2.3. Demonstration One: Spin Test

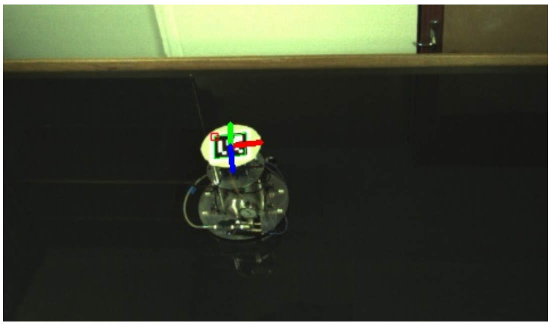
The Spin Test demonstrates the compound satellite's ability to perform a rotational movement using its thrusters. Feedback on this movement is provided by the overhead camera facility and by sensors on the compound satellite. The rotation goes through multiple phases of movement, which include an initial stationary phase, an acceleration phase, a constant velocity phase, a deceleration phase, and a final stationary phase.

Further to demonstrating the compound satellite's ability to perform rotational movement, the Spin Test also demonstrates its ability to move through multiple states and its ability to record data from the magnetometer and gyroscope on the BenchSat OBC. The test

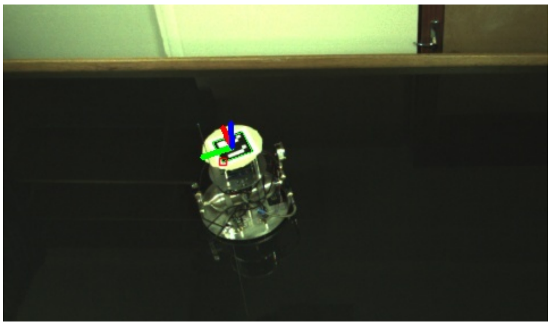
[1]A specific payload was not designed so could not be tested. However, this point was demonstrated in Section 4.4 where the EPS' ability to deliver power to attached loads was tested.

was executed multiple times using both ground stations to initiate the movement. The results shown in this section are from a single test that was controlled by the TIM Ground Station. Similar results were achieved using the ESL Ground Station.

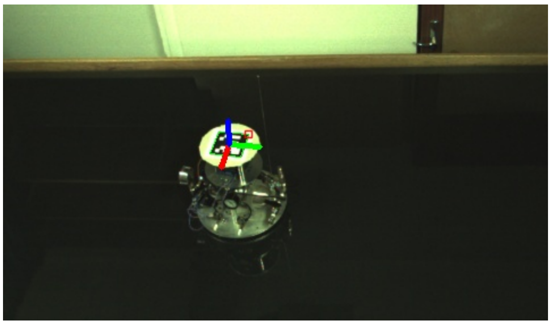
Figure 8.3 shows camera feedback of the compound satellite at four stages during the test. The figure shows that the compound satellite has rotated through a complete rotation. This visual feedback confirms that the compound satellite is capable of using thrusters to rotate the air bearing carrier cart as per desired demonstration outcome 8 in Section 8.2.1.



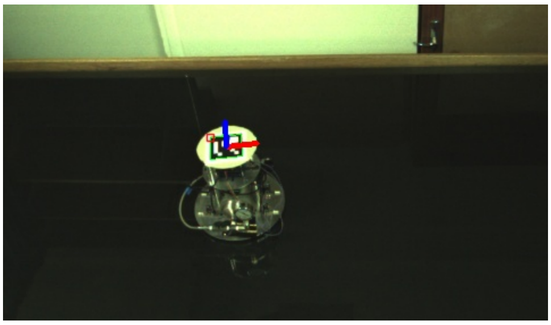
(a) Position 1: $t < 0s$



(b) Position 2: $t = 3.8s$



(c) Position 3: $t = 7.8s$



(d) Postion 4: $t = 11.8s$

Figure 8.3: Compound Satellite Image Feedback from Overhead Camera during the Spin Test

Figure 8.4 shows the compound satellite rotational velocity data from the gyroscope. From the feedback, the phases of movement that the compound satellite passes through can be identified.

The first stationary phase is at $t < 0s$ where the compound satellite has no rotational velocity. At $t = 0s$ the thrusters fire and the constant acceleration phase begins and lasts while $0s < t < 2s$. The constant velocity phase lasts from $t = 2s$ until $t = 10s$. The velocity decreases slightly in this due to friction between the compound satellite air bearing and the glass platform. The compound satellite then enters the deceleration phase for $10s < t < 12s$. In an ideal situation the compound satellite would come to a complete stop since the acceleration time was equal to the deceleration time. However, as a result of the velocity decrease due to friction during the constant velocity phase, the deceleration overshoots and the compound satellite does not come to a complete stop. A control system

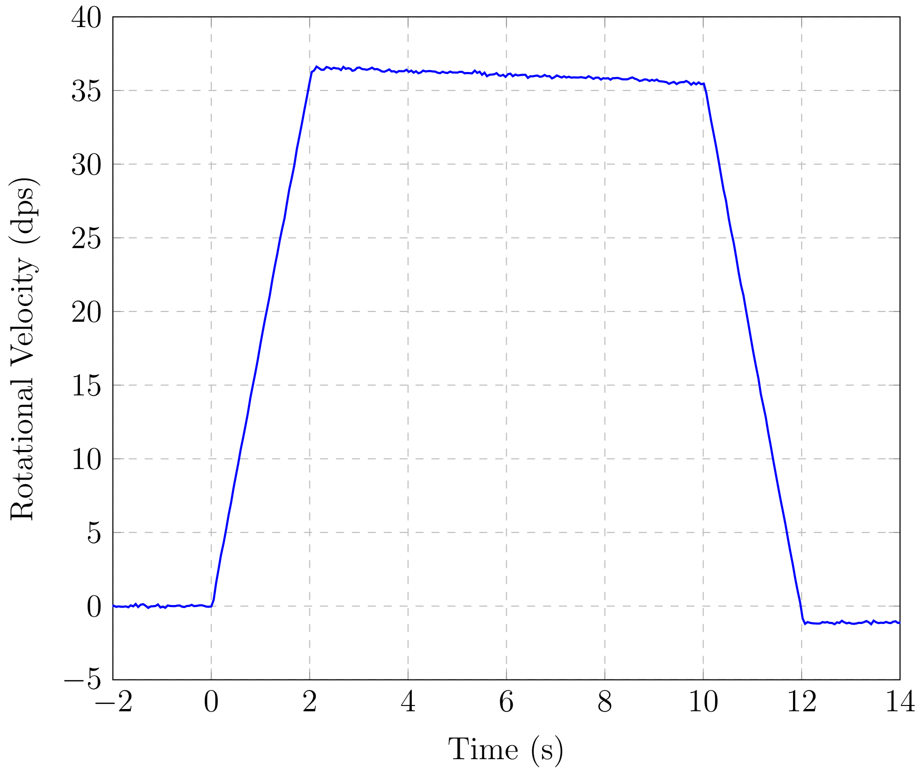


Figure 8.4: Rotational Velocity of the Compound Satellite versus Time during Spin Test as recorded by the OBC Gyroscope

algorithm that accounts for friction can be designed to enable more accurate control of the compound satellite during future experiments and demonstrations. Once the deceleration is complete the final stationary phase is achieved for $t > 12s$.

This movement demonstrates that the compound satellite can control experiments while moving through multiple states as per desired demonstration outcome 1 in Section 8.2.1. This result is demonstrated by the compound satellite's ability to enter an experimental state based on a ground station command and then to move through multiple states based on the internal system timing. The feedback from the gyroscope demonstrates the ability to gather and store data as per desired demonstration outcome 3 in Section 8.2.1.

Figure 8.5 shows the orientation of the compound satellite as it rotates during the Spin Test based on magnetometer and camera feedback. The positions shown in Figure 8.3 are marked in the figure to provide context to the images.

The recorded magnetometer feedback demonstrates the compound satellite's ability to record magnetometer feedback as per desired demonstration outcome 3 in Section 8.2.1. Additionally, Figure 8.5 compares the orientation feedback from the magnetometer and the overhead camera. This feedback can be expected when using the compound satellite during testing. Based on the rotational velocity results it is expected that the orientation would change consistently between $t = 2s$ and $t = 10s$ while the compound satellite is in

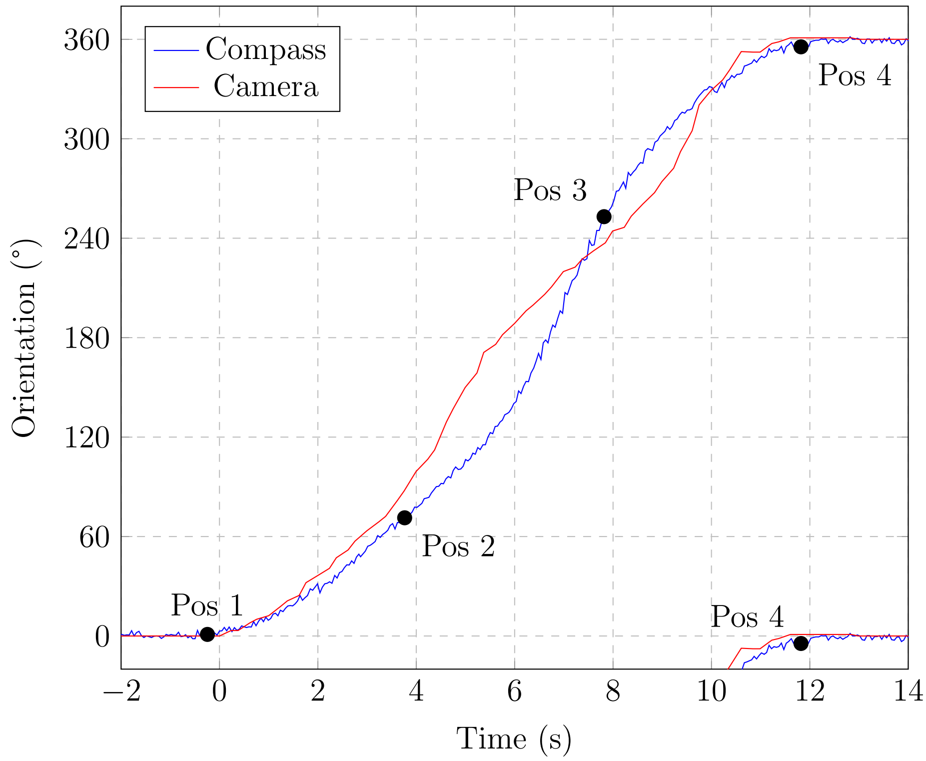


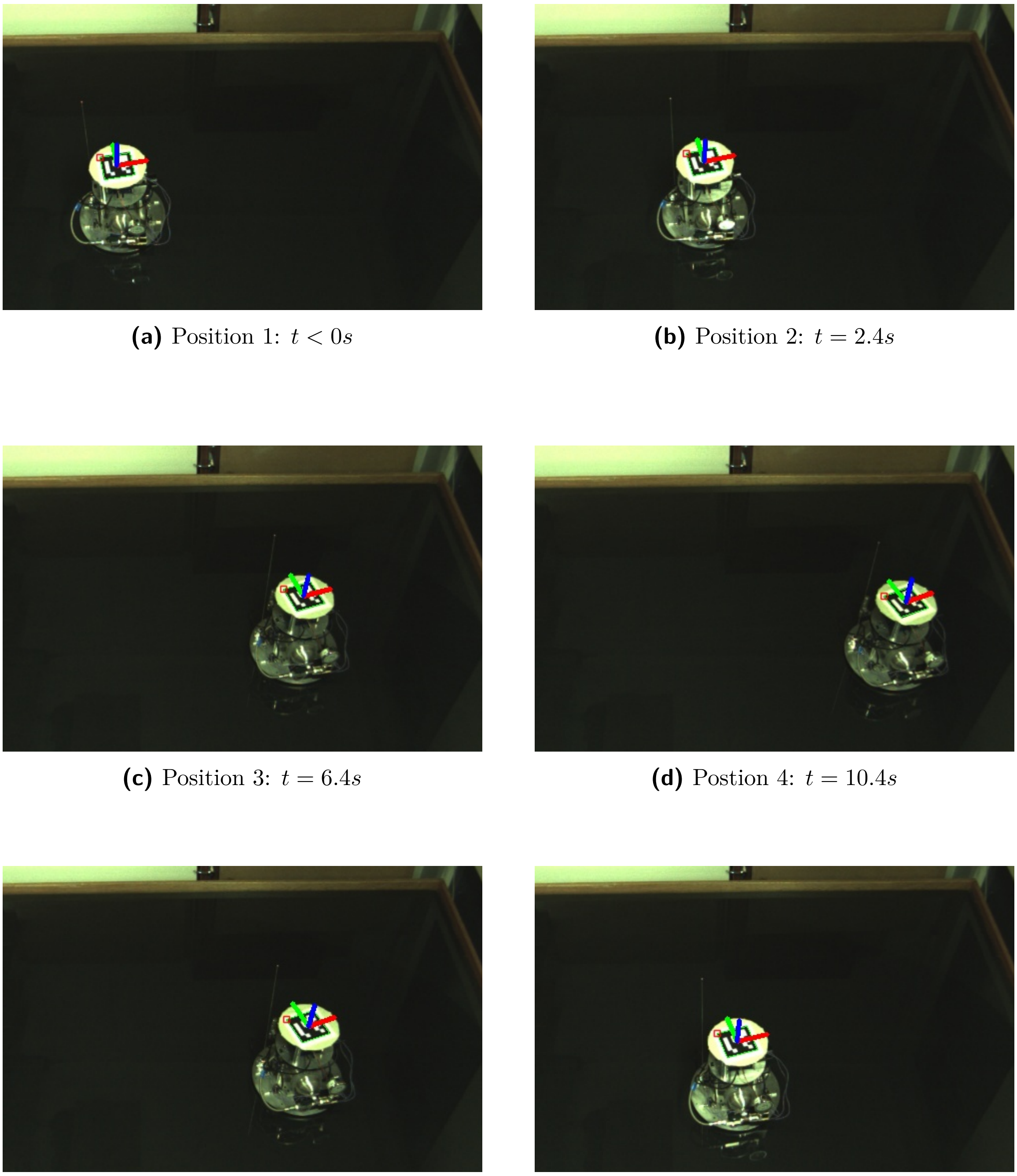
Figure 8.5: Orientation of the Compound Satellite versus Time during the Spin Test as recorded by the OBC Magnetometer and the Overhead Camera System

the constant velocity phase. However, the orientation measurements tend to vary and the camera feedback has a clear deviation. These variations can be handled through camera and magnetometer calibration in future implementations.

8.2.4. Demonstration Two: Forward Test

The Forward Test is similar to the Spin Test, however, it involves using the thrusters for forward movement instead of rotation. The Forward Test is tracked and monitored by the overhead camera and acceleration measurement is recorded by the OBC accelerometer.

The compound satellite accelerates forward for two seconds using its thrusters. The satellite then travels for eight seconds before thrusting in the opposite direction for two seconds to decelerate. The demonstration was repeated using both ground stations. The results shown in this section are from a single test that was controlled by the TIM Ground Station. The overhead camera feedback, shown in Figure 8.6, depicts the movement of the compound satellite.



(a) Position 1: $t < 0s$

(b) Position 2: $t = 2.4s$

(c) Position 3: $t = 6.4s$

(d) Postion 4: $t = 10.4s$

(e) Position 5: $t = 14.4s$

(f) Postion 6: $t = 18.4s$

Figure 8.6: Compound Satellite Image Feedback from Overhead Camera during the Forward Test

Figure 8.6 shows the compound satellite moving to the right based on the initial acceleration (positions 1 to 4) and then moving back towards the left based on the second thrust (positions 5 and 6). The second thrust was implemented to decelerate the compound satellite, however, due to friction the compound satellite had lost most of its velocity so the second thrust caused the satellite to move left.

The movement of the compound satellite and its position as recorded by the overhead camera can be seen in Figure 8.7. The positions from the images in Figure 8.6 are marked to demonstrate the compound satellite movement as time progresses. The time between the marked positions is constant.

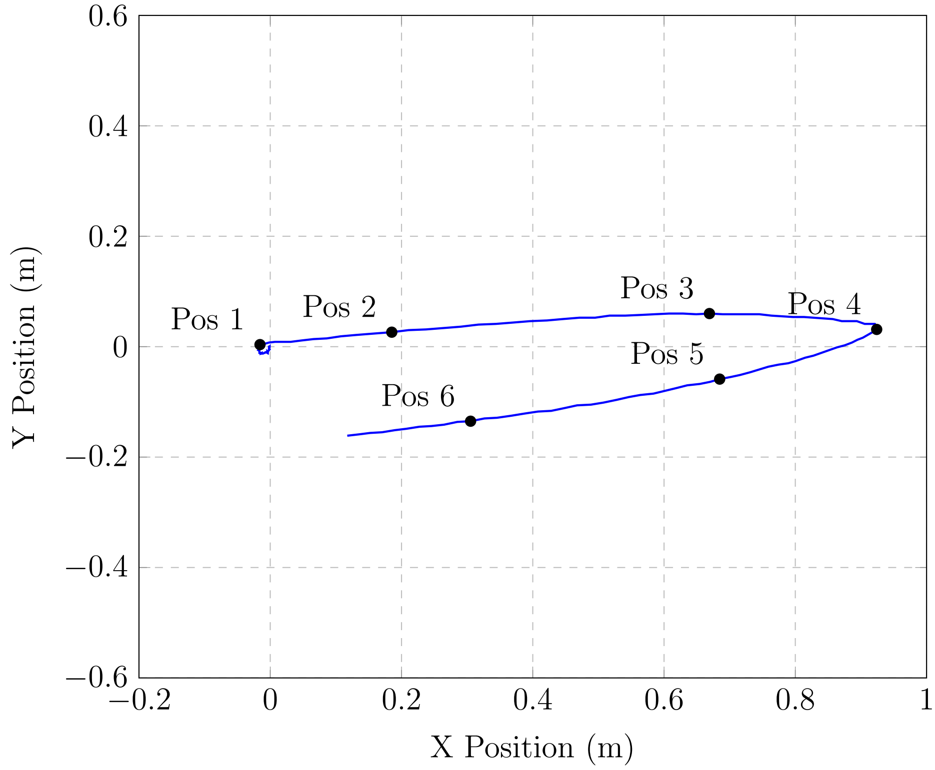


Figure 8.7: Compound Satellite Displacement during the Forward Test with Positions in Figure 8.6 marked

The velocity of the compound satellite can be characterised using the data in Figure 8.7. Since the time separation between the marked positions is constant, it can be noted that the maximum velocity occurs between positions 2 and 3 and positions 5 and 6. Ideally, the carrier cart would have come to a halt at position 4 without accelerating in the opposite direction to the initial movement. A feedback control algorithm can account for friction in the air bearing in future projects.

Despite not being ideal movement, the results shown in Figures 8.6 and 8.7 demonstrate the compound satellite's ability to move forward as per desired demonstration outcome 7 in Section 8.2.1.

Figure 8.8 shows the accelerometer feedback on the compound satellite during the Forward Test. The accelerometer noise is filtered using a digital first-order low-pass filter. The filtered acceleration shows clear spikes at $0 < t < 2s$ and $10s < t < 12s$ when the thrusters are active. These results demonstrate the compound satellite's ability to gather and store data from the accelerometer as per desired demonstration outcome 3 in Section 8.2.1.

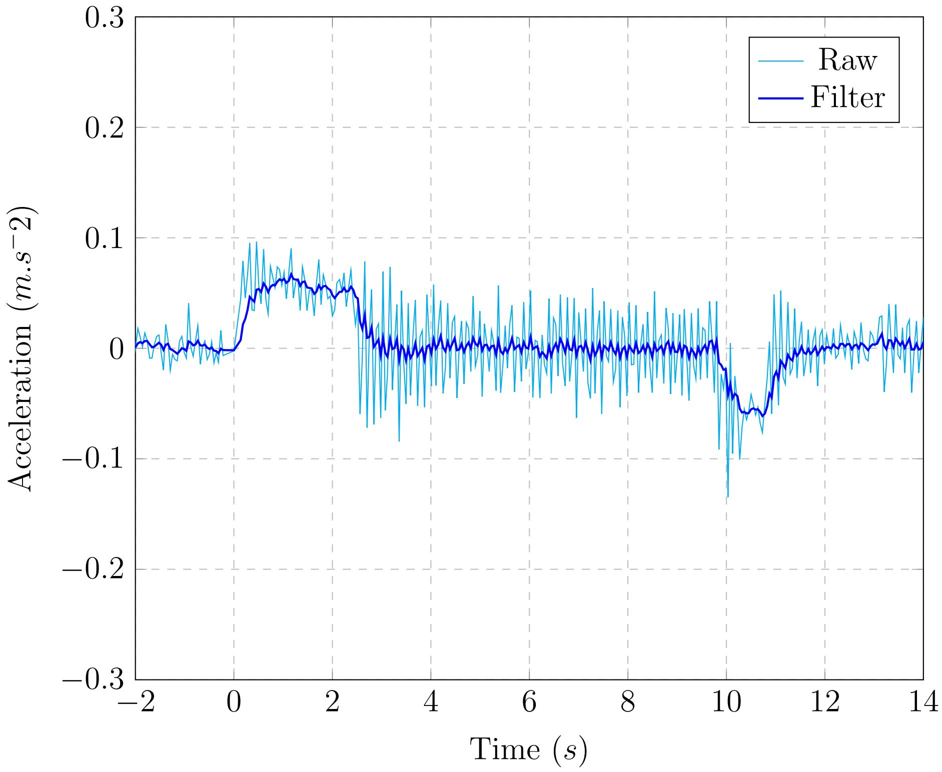


Figure 8.8: Acceleration of the Compound Satellite versus Time during the Forward Test as recorded by the OBC Accelerometer

8.2.5. Demonstration Three: Command Test

The Command Test demonstrates the communication capabilities of the compound satellite. The demonstration covers both external communications between the compound satellite and the ground stations and internal communications between subsystems on the compound satellite using the CAN-bus. The demonstration was repeated using the TIM Ground Station and the ESL Ground Station, with similar functionality and results. The results shown in this section are from a single test using the ESL Ground Station.

Table 8.1 shows the commands sent to the compound satellite during the Command Test.

No.	GS Time	Command	Details
1	5.2	Unit Test	*1.2.18*
2	12.7	Start Record	*1.0.22*
3	23.4	Forward Thrust	*2.0.3.0.2.5*
4	32.0	Reverse Thrust	*2.0.3.0.1.5*
5	44.6	Rotate Clockwise	*2.0.3.0.3.5*
6	54.4	Rotate Anticlockwise	*2.0.3.0.4.5*
7	64.6	End Record	*1.0.22*
8	77.2	Unit Test	*1.2.18*

Table 8.1: All Commands Sent during the Command Test

For each command that was sent during the Command Test, an acknowledgment was received from the compound satellite within two seconds. Along with acknowledgements the compound satellite also broadcasted status beacons every five seconds. The beacons were stored by the ground station software. Selected beacons from the Command Test are shown in Table 8.2.

No.	Ground Station Time	BenchSat State
1	0	Standby
3	10.0	Standby
4	15.1	Record
9	40.4	Record
14	65.6	Standby

Table 8.2: Selected Beacons from the Command Test

Tests to achieve the results seen in Tables 8.1 and 8.2 were repeated using both ground stations. These results demonstrate a two way connection between the compound satellite and both the TIM Ground Station and the ESL Ground Station as per desired demonstration outcome 6 in Section 8.2.1.

Figure 8.9 shows the changes in the compound satellite's attitude during the Command Test. The commands from Table 8.1 are marked on the figure to provide context to the movements.

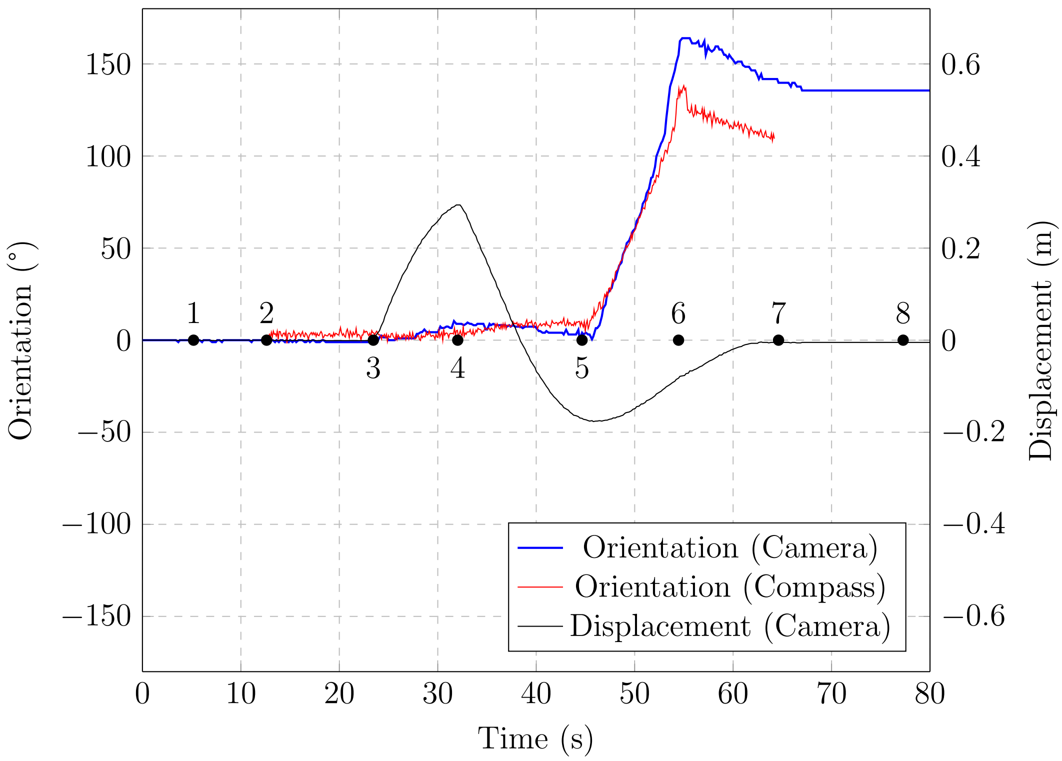


Figure 8.9: Orientation and Displacement of the Compound Satellite versus Time during the Command Test as Recorded by the OBC Magnetometer and the Overhead Camera System, with commands from Table 8.1 marked

Figure 8.9 shows how a ground station can be used to control the compound satellite during an experiment. Commands from the ground station can start and stop the compound satellite from recording data (commands 2 and 7 respectively), activate the thrusters to move the compound satellite forward and back (commands 3 and 4 respectively), and activate the thrusters to rotate the compound satellite (commands 5 and 6). For each of these commands, the results in Figure 8.9 show that the compound satellite responds immediately and moves correctly. These results demonstrate that multiple movements can be executed using the compound satellite when controlled by ground station commands as per desired demonstration outcome 2 in Section 8.2.1.

The EPS updates the OBC with current measurements for both the 5V and 3V3 system power channels. Messages containing current data are sent every 100 ms over the CAN-Bus from the EPS to the OBC to be recorded in permanent storage. The current measurements from the 5V channel and the 3V3 channel, filtered to remove noise, are shown in Figure 8.10.

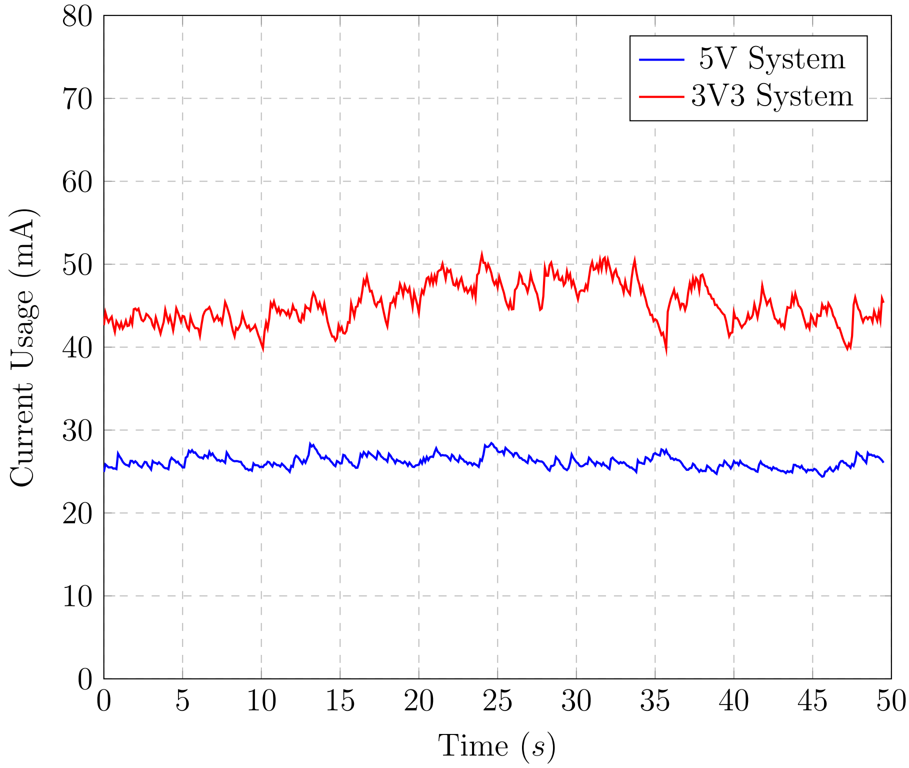


Figure 8.10: Compound Satellite Filtered Current Usage versus Time during Command Test

This result demonstrates the internal communication on BenchSat using the CAN-Bus as per desired demonstration outcome 4 in Section 8.2.1.

The power usage shown in Figure 8.10 demonstrates that power was successfully delivered to all components on the compound satellite as per desired demonstration outcome 5 in Section 8.2.1. Furthermore, the functionality shown in all three demonstrations shows that power was successfully shared with all components to enable them to complete the prescribed tasks.

To confirm that the power measurements recorded during the Command Test are in line with the expected functioning of the compound satellite, the recorded values are compared with the values recorded during standalone testing.

Based on the functions required during Command Test it can be assumed that the EPS and the RCS are predominantly in standby mode as the RCS will only transmit for a small fraction of the overall time and the EPS does not have any loads attached. The OBC is in active mode since it actively uses sensors and solenoids during the test. Based on these assumptions and information from Table 4.1 in Section 4.4.4; Figure 5.13 in Section 5.4.5; and Figure 6.18a in Section 6.4.6 the cumulative 3V3 channel current is 46 mA and the cumulative 5V channel current is 26 mA.

Figure 8.10 shows that the 3V3 channel current varies between 40 mA and 50 mA while the 5V channel varies between 25 mA and 30 mA. The current measurements in Figure 8.10 are in the expected range based on standalone testing. This comparison shows that the BenchSat subsystems were functioning similarly during the Command Test to how they functioned during standalone testing.

8.3. Summary

This chapter discussed the integration of the compound satellite. The integration first involved the completion of the BenchSat stack with all subsystems present. BenchSat was then integrated with the air bearing carrier cart and the ground station to form a compound satellite.

The functionality of the compound satellite was demonstrated in three tests. The three demonstrations included the Spin Test, Forward Test, and Command Test. Each demonstration highlighted different desired outcomes to show that the overall project scope has been achieved.

Table 8.3 links the desired demonstration outcomes listed in Section 8.2.1 to the three demonstrations.

No	Capabilities	Spin	Forward	Command
1	Experiment Multiple States	✓	-	-
2	Experiment Based on GS Commands	-	-	✓
3a	Gather and Store Data (Magnetometer)	✓	-	-
3b	Gather and Store Data (Gyroscope)	✓	-	-
3c	Gather and Store Data (Accelerometer)	-	✓	-
4	Internal Communications on CAN-Bus	-	-	✓
5	Provide Power	-	-	✓
6	Communicate with TIM GS & ESL GS	-	-	✓
7	Move Carrier Cart Forward	-	✓	-
8	Rotate Carrier Cart	✓	-	-

Table 8.3: Overview of the BenchSat Demonstration Outcomes

The data in Table 8.3 illustrates that all of the desired demonstration outcomes were achieved. The next chapter will summarise and conclude the project.

Chapter 9

Conclusion

The goal of this thesis was to design and build a bench satellite system to enable realistic experiments in the ESL. This chapter will cover a summary of the work done for this project, the outcomes of the project, and future work and recommendations.

9.1. Summary

This thesis addressed the need for a bench satellite system to be developed for use within the ESL. The need was met through the design and implementation of a bench satellite called BenchSat, which is a combination of modular subsystems that can function on their own. Existing bench satellites were reviewed to identify important concepts that would be involved in the design of BenchSat.

The system-level design for BenchSat was discussed using the bench satellites reviewed in the literature study for guidance. BenchSat was designed to be made of separate subsystems each of which is modular and capable of functioning on its own. A system architecture was created to define the interactions between subsystems on BenchSat as well as the interactions between BenchSat and the surrounding facilities.

Three subsystems were developed to integrate into BenchSat. They are an EPS, an RCS, and an OBC. Each subsystem was designed, implemented, and tested as a standalone subsystem. The subsystems fulfill different aspects of BenchSat's desired characteristics: the EPS fulfills the power generation and distribution requirement, the RCS executes interactions with the ground station, and the OBC enables integration with the air bearing carrier cart and manages and controls experiments.

In addition to BenchSat, this project involved developing ground station systems that could use the existing facilities to interact with BenchSat. Two ground stations were used in the project; one external and available for testing as well as live satellite missions; and one internal and setup in the ESL. Both ground stations were discussed in terms of the hardware and software used. Although the different ground stations implemented different hardware, a similar software implementation was used on both ground stations.

Finally, to demonstrate that the overall design of BenchSat was successful in achieving its goals, the system was used in three demonstrations to show the extent of the system's capabilities. These demonstrations involved all three subsystems integrated into BenchSat while utilising the air bearing facility and a selected ground station.

The demonstrations showed that BenchSat successfully achieved the desired outcomes of this thesis. It is a capable bench satellite that provides a realistic testing environment and can be used to develop future projects.

9.2. Outcomes

To address the problem defined in Section 1.4, the goal of the project was to develop and implement a satellite-like system, that could provide power, communications and computing to developing projects in the ESL. This goal was met through the development of BenchSat. BenchSat with all the subsystems integrated is pictured in Figure 8.1.

To decide whether BenchSat can be considered functional, objectives that provided high-level benchmarks with which to judge BenchSat were defined in Section 1.5. These objectives were shown to have been achieved in the system demonstrations discussed in Section 8.2 and during the standalone testing discussed in Sections 4.4, 5.4, and 6.4. The objectives and how they were achieved are outlined below:

1. Controlling experiments and storing gathered data
    - Demonstration One, the Spin Test, discussed in Section 8.2.3 demonstrated this objective. The demonstration showed that BenchSat was able to manage an experiment through multiple states while gathering information from multiple sensors.
2. Providing power for itself, additional payloads and the air bearing carrier cart
    - The regulator efficiency testing in Section 4.4.1 showed how power can be directed to a load.
    - During all three demonstrations the system showed that it can power itself and the air bearing carrier cart.
3. Connecting to the TIM Ground Station
    - The connection to the ground station was tested in Section 5.4 and the system was shown to communicate while meeting all of the system requirements.

- During the demonstrations in Chapter 8 it was also shown that the ground station could be used to receive beacons, initiate tests, and command tests.

4. Controlling the air bearing carrier cart

   - During the demonstrations described in Section 8.2, BenchSat showed the ability to perform the two primary movements of the carrier cart; namely to move straight (Forward Test in Section 8.2.4) and to rotate (Spin Test in Section 8.2.3).

9.3. Future Work and Recommendations

BenchSat has been designed to be used by future projects to develop and build satellite systems in the ESL. The components of BenchSat could be upgraded to become the satellite bus of a satellite such as DockSat. For the DockSat implementation to become a reality extensive work would need to be done, including upgrading components to be radiation-hardened and developing solar, thermal, and structural subsystems.

However, there are some functions of BenchSat that can be addressed in future work to improve its functionality as a development tool and enable it to be used in more environments and more efficiently. This future work could include:

1. Investigating the EPS current measurement system to see if line loss can be reduced. The in-line resistor is causing voltage drops across the switch it would be beneficial to future projects for these losses to be reduced.

2. Replacing the switches and regulators on the EPS. The switches and regulators were chosen during a global shortage of components and therefore there was a lack of choice. Future iterations of the EPS design should use components with equivalent or better functionality, but with convenient footprints to allow students to resolder the components rather than having to rely on a technician.

3. Extending the range of the RCS to enable testing in a variety of environments and locations, including high-altitude balloons. An expanded range will also be required to develop space-faring radio equipment.

4. Improving the RCS receive algorithm to utilise the RSSI feature on the RCS receiver would improve the subsystem's functioning and noise rejection.

5. Developing the TIM Ground Station hardware to be able to transmit at any baud rate. Furthermore, the hardware and software on the TIM Ground Station should be investigated to make the system transparent to any user.

6. Implementing a control system for the air bearing facility on the OBC. A control system would enable complex movement sequences that would be needed during experiments that require accurate movement, such as a docking demonstration.

7. Upgrading the RCS and EPS microcontrollers so they no longer use development boards. This upgrade would save space and power.

8. Upgrade the connecting header to use fewer pins or more advanced technology than the current header. Since the communications architecture has limited connections, the connecting header does not need to be as big as it currently is. A future iteration could reduce the number of shared pins or implement a different technology such as backplane connectors.

BenchSat is a satellite-like system that aims to assist in the development of other satellite systems within the ESL. To demonstrate the capabilities of the final system, BenchSat was integrated with the air bearing carrier cart and controlled by the TIM Ground Station to move around the air bearing facility. This demonstration displayed the full functionality of BenchSat and showed that the project objectives were met.

Bibliography

[1] L. Visagie, W. Jordaan, R. Waller, A. Burger, A. Barnard, and S. Jeffery, "Dock-sat: Cubesat undocking and docking mission." Palaiseau, France: 12th European Cubesat Symposium, 11 2021.

[2] D. J. Barnhart, J. J. Sellers, C. A. Bishop, J. R. Gossner, J. J. White, and J. B. Clark, "Systems analysis and systems engineering eyassat: A revolution in teaching and learning space systems engineering," 2005.

[3] C. J. Groenewald, "Attitude determination and control system for eyassat for hardware in the loop application," 4 2014. [Online]. Available: http://hdl.handle.net/10019.1/86653

[4] D. C. Jansen, "The development of a test facility for satellite experiments," 2021. [Online]. Available: http://scholar.sun.ac.za/handle/10019.1/109916

[5] W. Steyn and J. Lochner, "Tim ground station survey," 2022.

[6] ESA, "Flatsat — a ground-based testbed for cubesats at esa/estec," 10 2021. [Online]. Available: https://www.eoportal.org/other-space-activities/flatsat#some-background

[7] J. Reilly, D. Murphy, and M. D. et al, "Eirflat-1: A flatsat platform for the development and testing of the 2u cubesat eirsat-1." Universitat Politècnica de Catalunya, 4 2022. [Online]. Available: https://upcommons.upc.edu/handle/2117/368979

[8] D. J. Barnhart, "Eyassat user's manual v4," 2010.

[9] O. Ritchey, J. Clark, J. White, T. White, D. Barnhart, and J. Sellers, "Eyassat: Transforming the way students experience space systems engineering." ASEE Conferences, 2004, pp. 9.602.1–9.602.15. [Online]. Available: https://peer.asee.org/13297

[10] T. I. Burroni, C. Mucanna, and M. I. Escobar, "Pujllay: the first nanosatellite engineering model developed in unsam," p. 4, 2019. [Online]. Available: https://www.researchgate.net/publication/346327905

[11] M. Yamazaki, M. I. Khalil, T. Zengo, T. Umezawa, R. Futamata, K. Maeda, N. Fukushima, and F. Seki, “Hepta-sat training program: International knowledge transfer using hands-on type cubesat education,” vol. 2018-October, 2018.

[12] H. E. S. Amiri, D. Brain, and O. S. et al, “The emirates mars mission,” *Space Science Reviews*, vol. 218, p. 4, 2 2022. [Online]. Available: https://link.springer.com/article/10.1007/s11214-021-00868-x#citeas

[13] A. AlSuwaidi, “Emirates mars mission flight simulator: Flatsat overview.” IEEE, 3 2020, pp. 1–11. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9172253

[14] D. J. Evans, “Ops-sat: Operational concept for esa’s first mission dedicated to operational technology.” American Institute of Aeronautics and Astronautics, 5 2016.

[15] E. S. Agency, “Opened-out ‘flatsat’ for cubesat testing,” 10 2021.

[16] S. Kacker, A. Meredith, and K. Cahoy, “Machine learning image processing algorithms onboard ops-sat,” *36th Annual AIAA/USU Conference on Small Satellites*, 8 2022. [Online]. Available: https://digitalcommons.usu.edu/smallsat/2022/all2022/65/

[17] A. Surivet, “Integration and validation of a nanosatellite flight software,” 2021. [Online]. Available: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1583521&dswid=-2023

[18] N. Instruments, “Controller area network (can) overview,” 9 2020. [Online]. Available: https://www.ni.com/en-za/innovations/white-papers/06/controller-area-network--can--overview.html

[19] J. C. Godinez, E. Orsel, and J. Guo, “Controller area network in nanosatellites,” 3 2016. [Online]. Available: https://indico.esa.int/event/120/contributions/484

[20] Amarendra, V. Khadkikar, M. S. E. Moursi, H. Zeineldin, N. Al-Sayari, and K. E. Al Hosani, “A comprehensive review on cubesat electrical power system architectures,” *IEEE Transactions on Power Electronics*, vol. 37, pp. 3161–3177, 2022.

[21] K. B., E. J. Brandon, R. V. Bugga, M. C. Smart, S. C. Jones, F. C. Krause, W. C. West, and G. G. C. Bolotin, “Energy storage technologies for small satellite applications,” *Proceedings of the IEEE*, vol. 106, pp. 419–428, 2018.

[22] R. Burt, “Distributed eps in a cubesat application,” *SSC11*, 2013. [Online]. Available: https://digitalcommons.usu.edu/smallsat/2011/all2011/40/

[23] I. T. Osman, "Design and implementation of eps (electrical power system) of a cubesat," 9 2012. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1032.5750&rep=rep1&type=pdf

[24] ISIS, "ieps type a, type b and type c datasheet," 8 2019. [Online]. Available: https://www.isispace.nl/wp-content/uploads/2019/04/ISIS-iEPS2_0-DS-00001-iEPS_Datasheet-1_0.pdf

[25] GOMspace, "Nanopower p31u datasheet," 1 2022.

[26] TI, "Tps25946 efuse with bidirectional current support," 5 2021. [Online]. Available: https://www.ti.com/lit/ds/symlink/tps25946.pdf

[27] S. Weston, "Small spacecraft technology state of the art report," pp. 232–240, 10 2021. [Online]. Available: https://www.nasa.gov/sites/default/files/atoms/files/soa_2021_1.pdf

[28] E. Media, "Practical guide to radio-frequency analysis and design." [Online]. Available: https://www.allaboutcircuits.com/textbook/radio-frequency-analysis-design/

[29] "Digital communication tutorial," 2022. [Online]. Available: https://www.tutorialspoint.com/digital_communication/index.htm

[30] J. Miller, "9600 baud packet radio modem design," pp. 135–140, 7 1988. [Online]. Available: https://www.amsat.org/amsat/articles/g3ruh/109.html

[31] J. R. Wertz and W. J. Larson, *Space Mission Analysis and Design*, 2003, vol. Third Edition.

[32] A. N. Guest, "Telemetry, tracking, and command (tt{&}c)," pp. 1067–1078, 2013. [Online]. Available: https://doi.org/10.1007/978-1-4419-7671-0_69

[33] W. A. Beech, D. E. Nielsen, and J. Taylor, "Ax.25 link access protocol for amateur packet radio," 7 1998. [Online]. Available: https://www.tapr.org/pdf/AX25.2.2.pdf

[34] ST, *S2-LP Ultra-low power, high performance, sub-1 GHz transceiver*, 9 2021. [Online]. Available: https://www.st.com/resource/en/datasheet/s2-lp.pdf

[35] Radiometrix, *NRX1 Low cost VHF Narrow Band FM receiver*, 3 2015. [Online]. Available: http://www.radiometrix.com/files/additional/nrx1.pdf

[36] O. W. Solutions, *BY-433-02 433MHz Antenna Specification*, 2022. [Online]. Available: https://cdn.shopify.com/s/files/1/2219/1447/files/BY-433-02_Datasheet.pdf?v=1595932522

[37] W. Kissick, W. Ingram, J. Vanderau, and R. Jennings, “Antenna system guide,” pp. 17–20, 4 2001. [Online]. Available: https://www.ojp.gov/pdffiles1/nij/185030.pdf

[38] C. S. Inc, “The link budget and fade margin,” 9 2016. [Online]. Available: https://s.campbellsci.com/documents/us/technical-papers/link-budget.pdf

[39] P. Botma, “The design and development of an adcs obc for a cubesat,” 12 2011.

[40] A. C. SPACE, “Command and data handling kryten-m3,” 7 2020. [Online]. Available: https://www.aac-clyde.space/wp-content/uploads/2021/10/AAC_DataSheet_Kryten.pdf

[41] nano avionics, “Cubesat on-board computer – main bus unit satbus 3c2,” 2022. [Online]. Available: https://nanoavionics.com/cubesat-components/cubesat-on-board-computer-main-bus-unit-satbus-3c2/

[42] ENDUROSAT, “Datasheet obc type i,” 3 2022.

[43] PX4, “Px4 architectural overview,” 10 2021. [Online]. Available: https://docs.px4.io/main/en/concept/architecture.html

[44] S. Hymel, “What is a real-time operating system (rtos)?” 1 2021. [Online]. Available: https://www.digikey.co.za/en/maker/projects/what-is-a-realtime-operating-system-rtos/28d8087f53844decafa5000d89608016

[45] C. Gregersen, “Bare metal vs. rtos vs. os: What’s the best for your iot solution,” 4 2021.

[46] X. Xia, G. Sun, K. Zhang, S. Wu, T. Wang, L. Xia, and S. Liu, “Nanosats/cubesats adcs survey.” IEEE, 5 2017, pp. 5151–5158.

[47] C. Konvalin, “Compensating for tilt, hard-iron, and soft-iron effects,” 12 2009. [Online]. Available: https://www.fierceelectronics.com/components/compensating-for-tilt-hard-iron-and-soft-iron-effects

[48] Z. Chu, C. Chen, Y. Liu, Y. Wang, and X. Lin, “Magnetic orientation system based on magnetometer, accelerometer and gyroscope,” *CAAI Transactions on Intelligence Technology*, vol. 2, pp. 166–172, 12 2017. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/trit.2017.0024

[49] M. Hölzl, R. Neumeier, and G. Ostermayer, “Analysis of compass sensor accuracy on several mobile devices in an industrial environment,” pp. 381–389, 2013. [Online]. Available: https://www.researchgate.net/publication/267154787_Analysis_of_Compass_Sensor_Accuracy_on_Several_Mobile_Devices_in_an_Industrial_Environment

[50] A. Fan, "How to improve the accuracy of inclination measurement using an accelerometer," *Analog Dialogue*, vol. 52, 2 2018. [Online]. Available: https://www.analog.com/media/en/analog-dialogue/volume-52/number-1/how-to-improve-the-accuracy-of-inclination-measurement-using-an-accelerometer.pdf

[51] V. K. Garg, "Fourth generation systems and new wireless technologies," pp. 23–1–23–22, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123735805500570#cesec5

[52] PE0SAT, "Sdrsharp," 2022. [Online]. Available: https://www.pe0sat.vgnet.nl/sdr/sdr-software/sdrsharp/

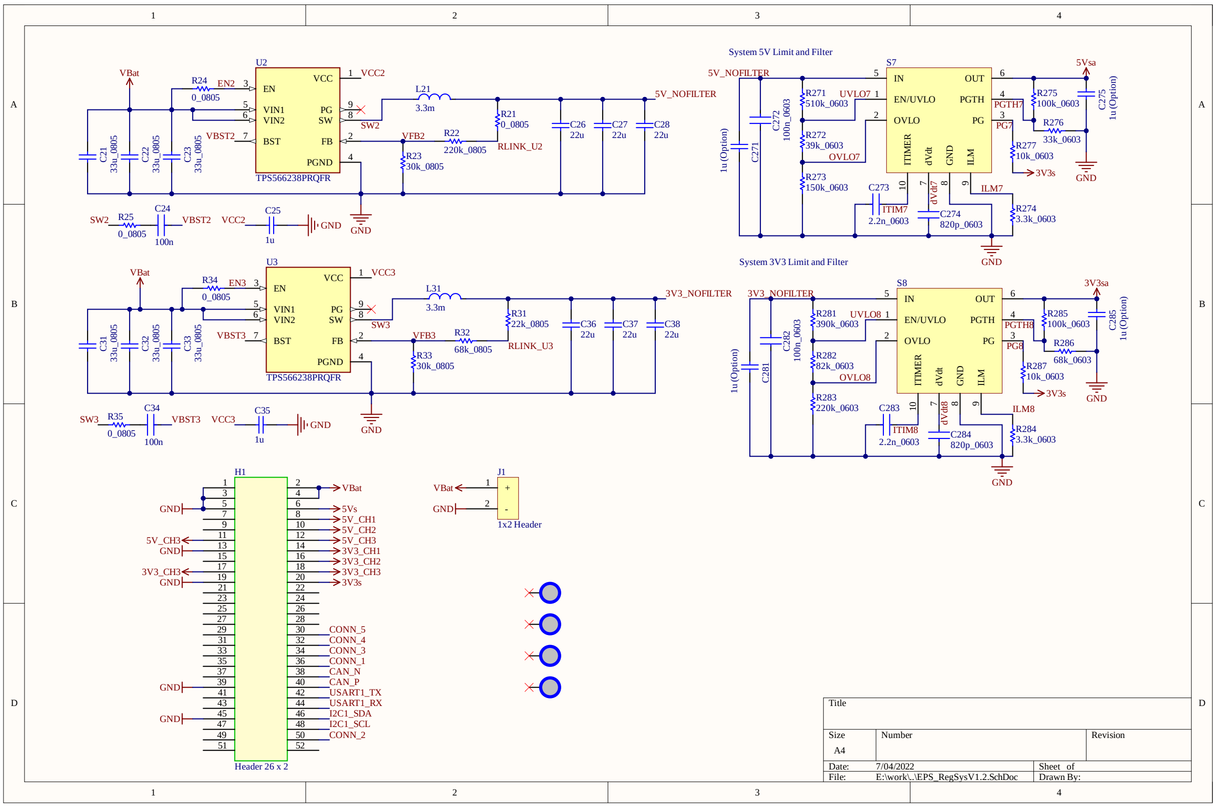
[53] "Gnu radio wiki," 2022. [Online]. Available: https://wiki.gnuradio.org/index.php?title=Main_Page

Appendix A

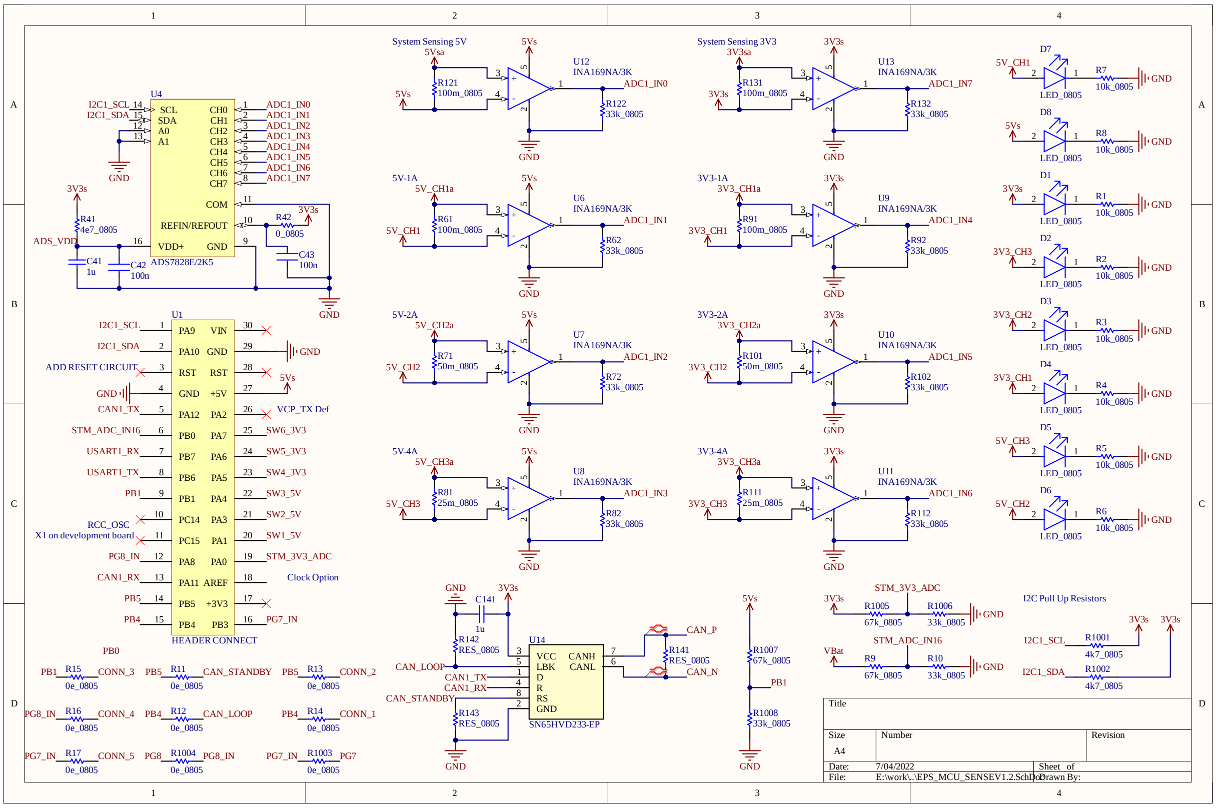
PCB Schematics and Diagrams

A.1. EPS

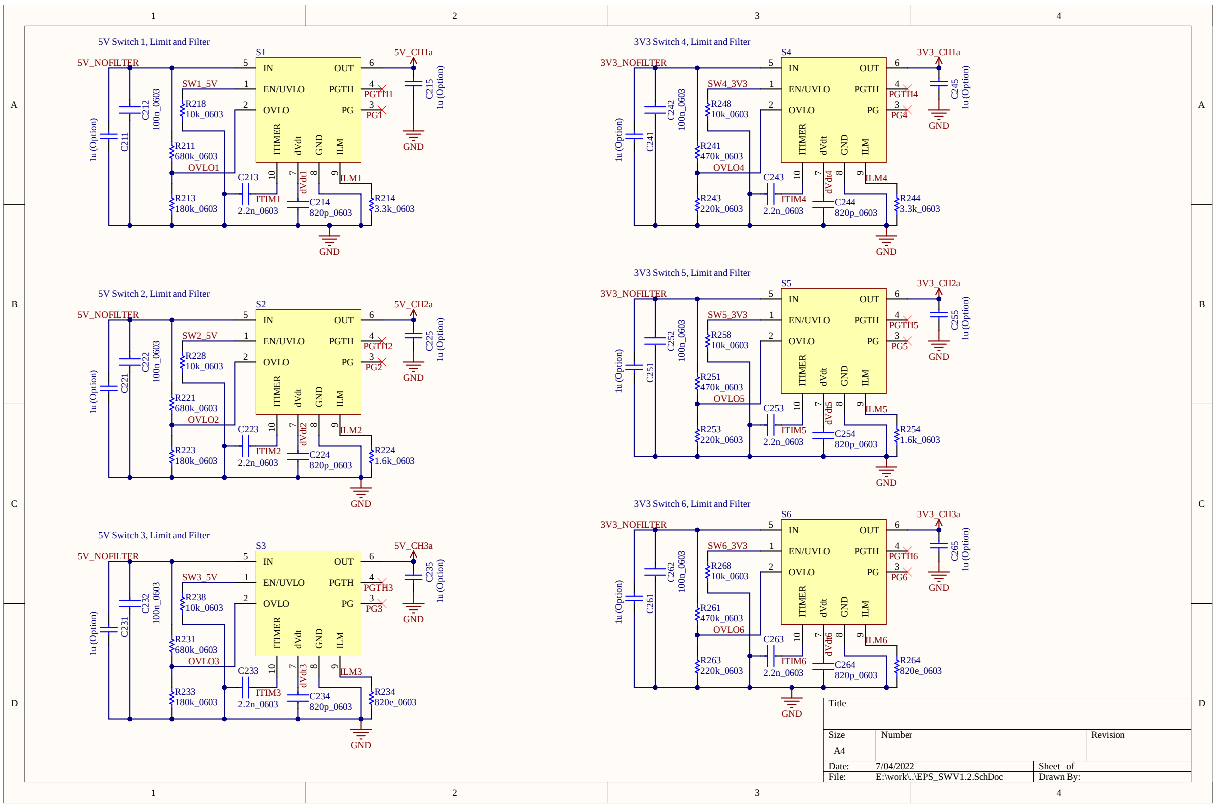
A.1.1. Schematics



(a) EPS Schematic Page 1



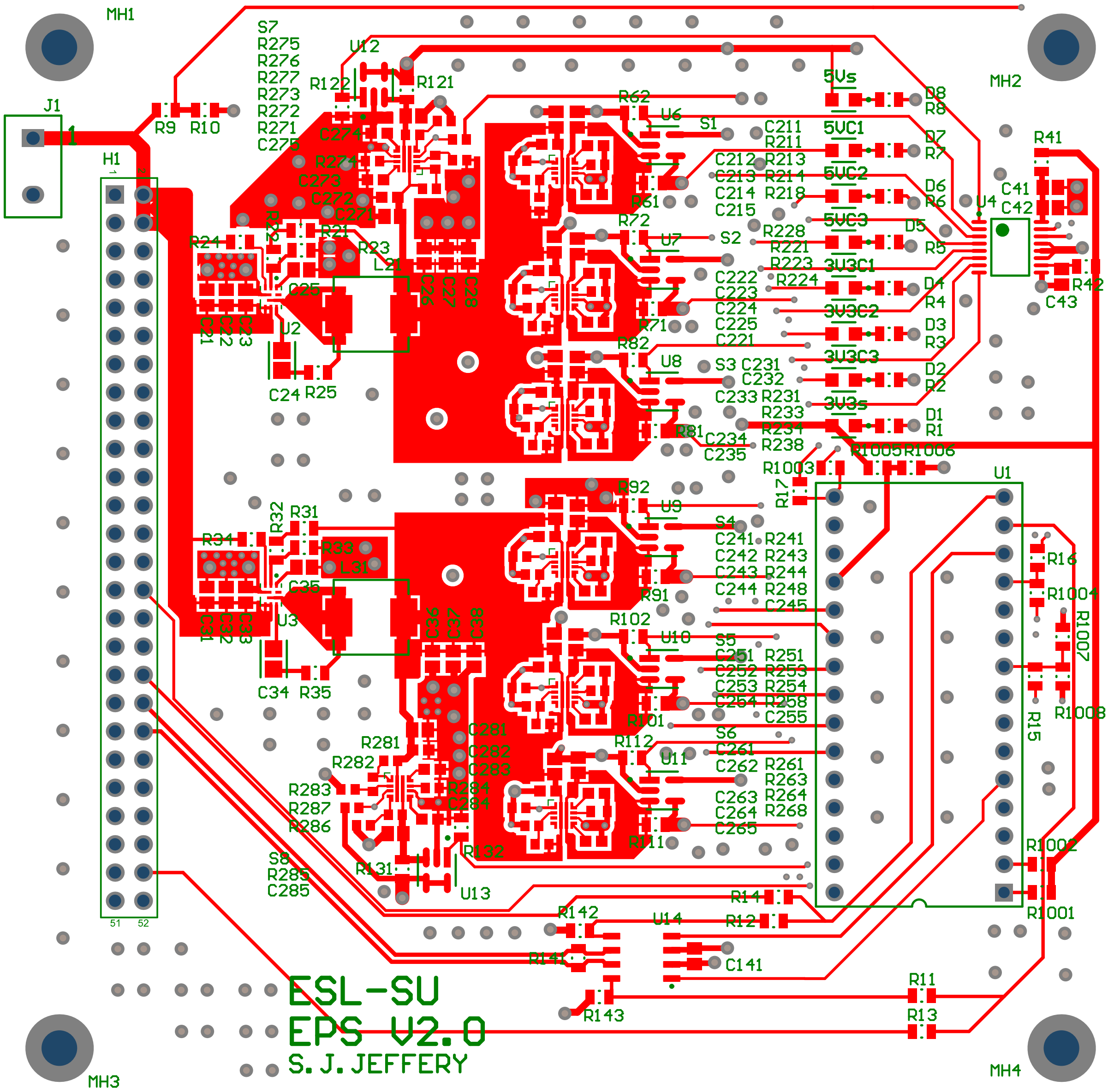
(b) EPS Schematic Page 2



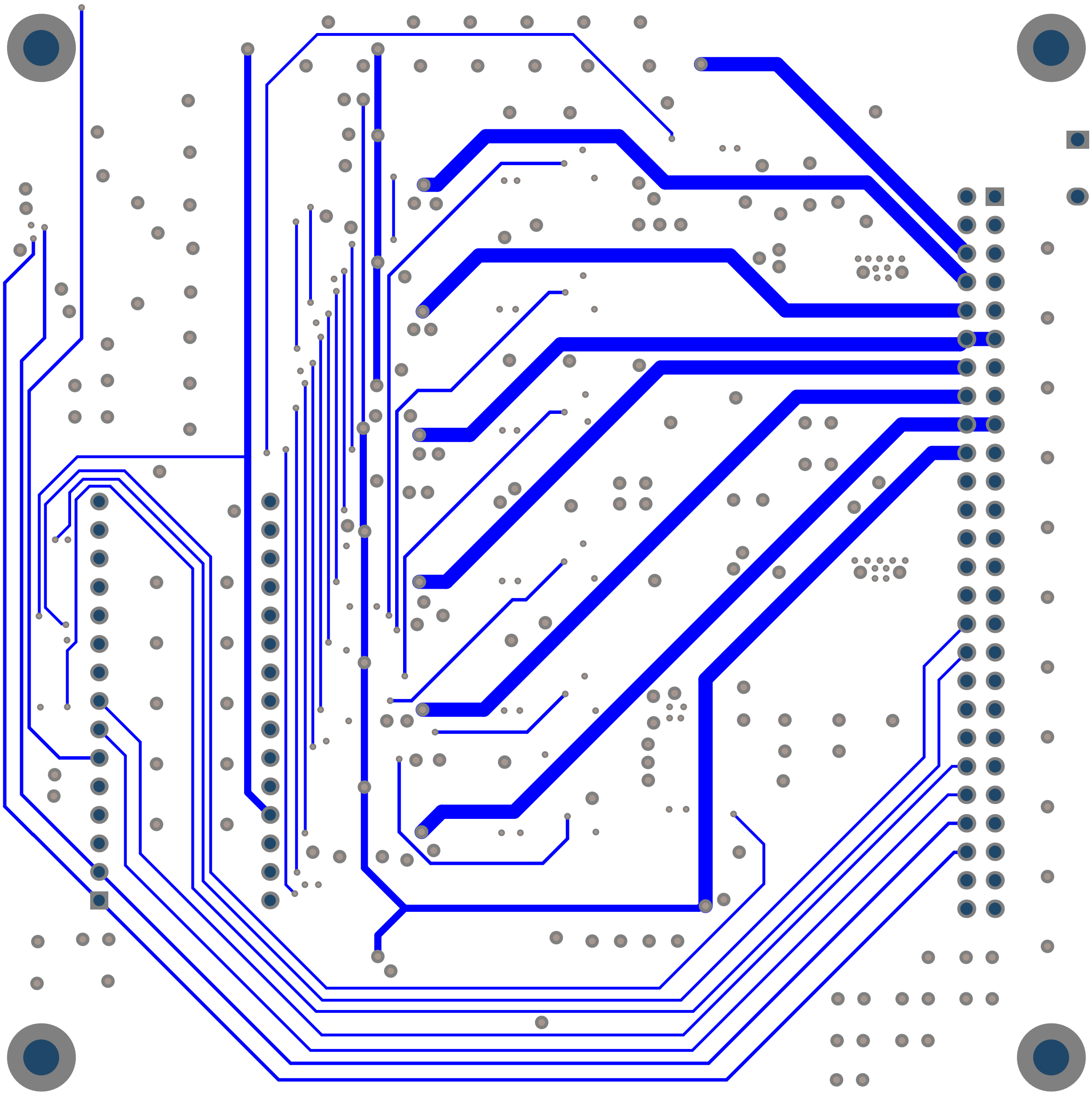
(c) EPS Schematic Page 3

Figure A.1: EPS Schematic Diagrams

A.1.2. PCB Layouts



(a) EPS PCB Top Layer



(b) EPS PCB Bottom Layer

Figure A.2: EPS PCB

A.2. RCS

A.2.1. Schematic

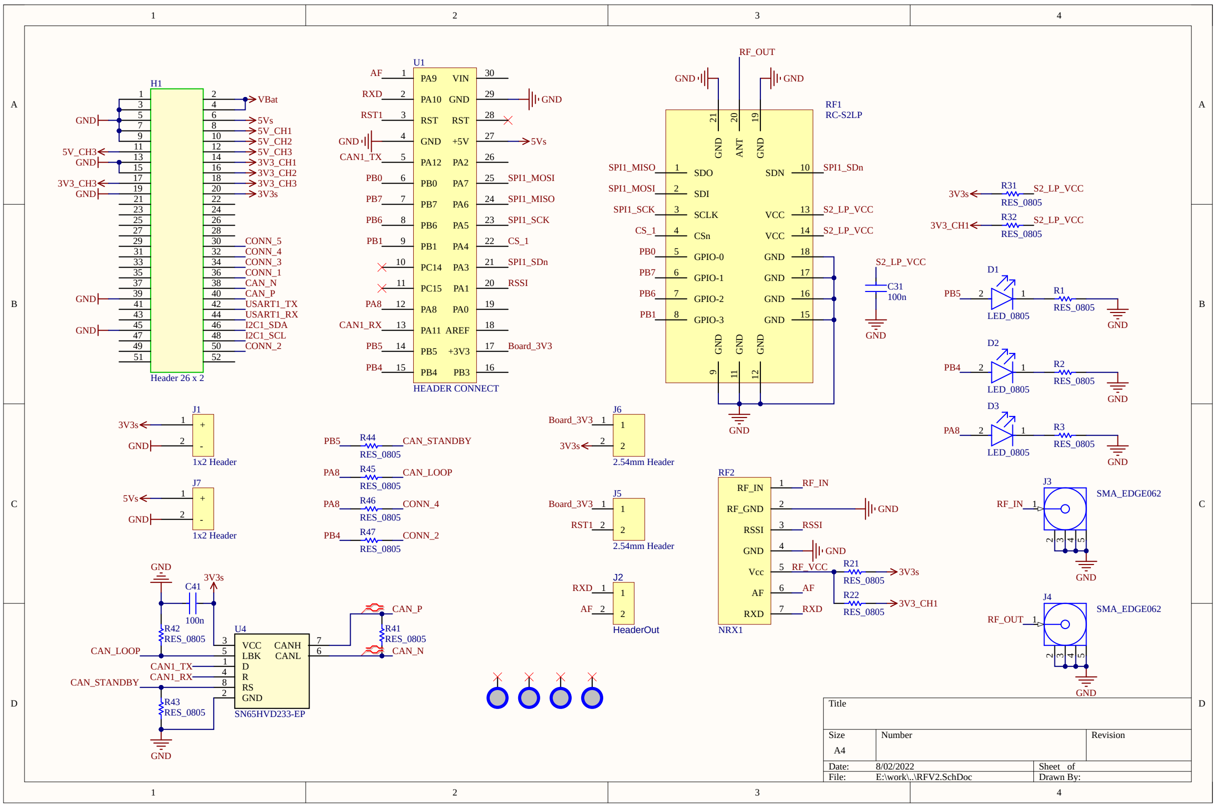
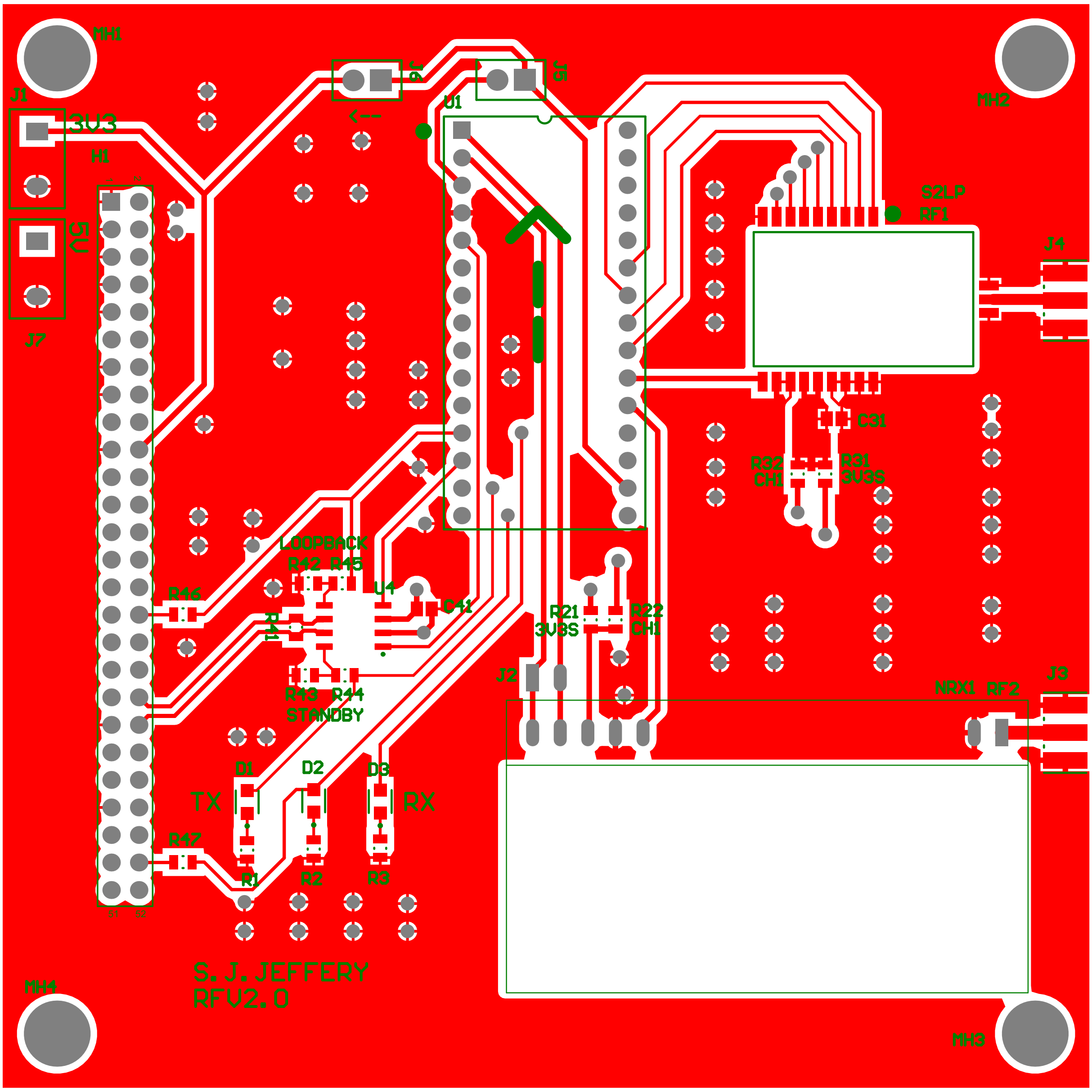
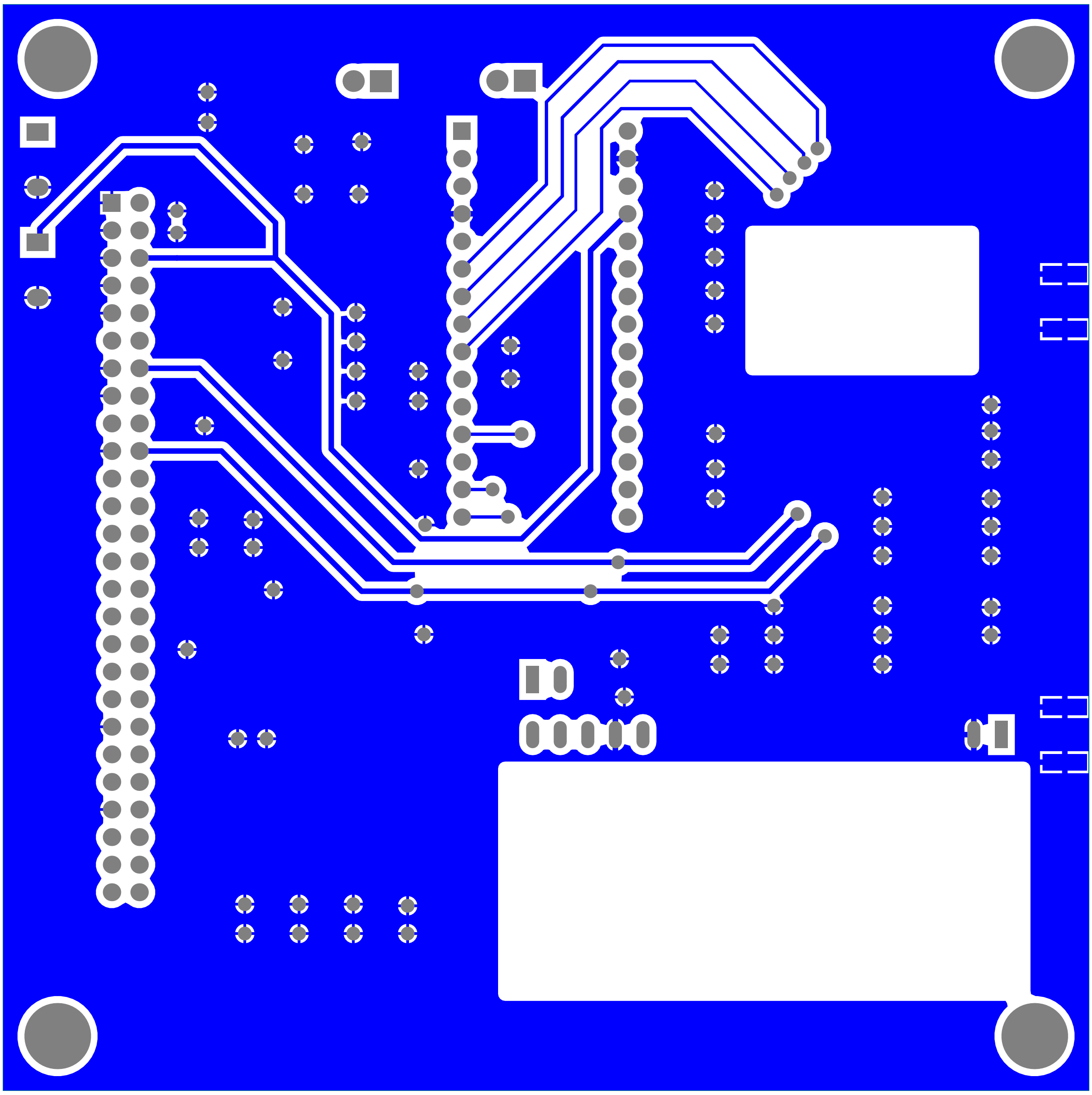


Figure A.3: RCS Schematic Diagram

A.2.2. PCB Layouts



(a) RCS PCB Top Layer



(b) RCS PCB Bottom Layer

Figure A.4: RCS PCB

A.3. OBC

A.3.1. Schematic

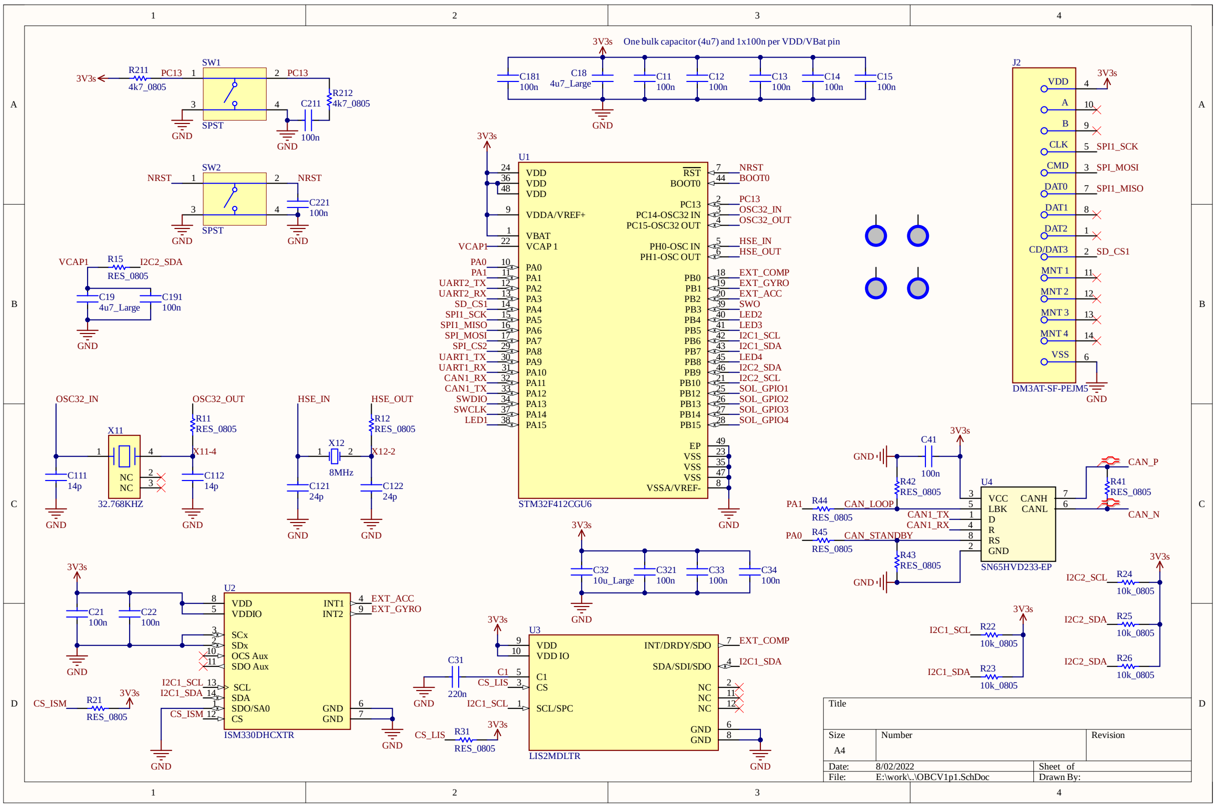
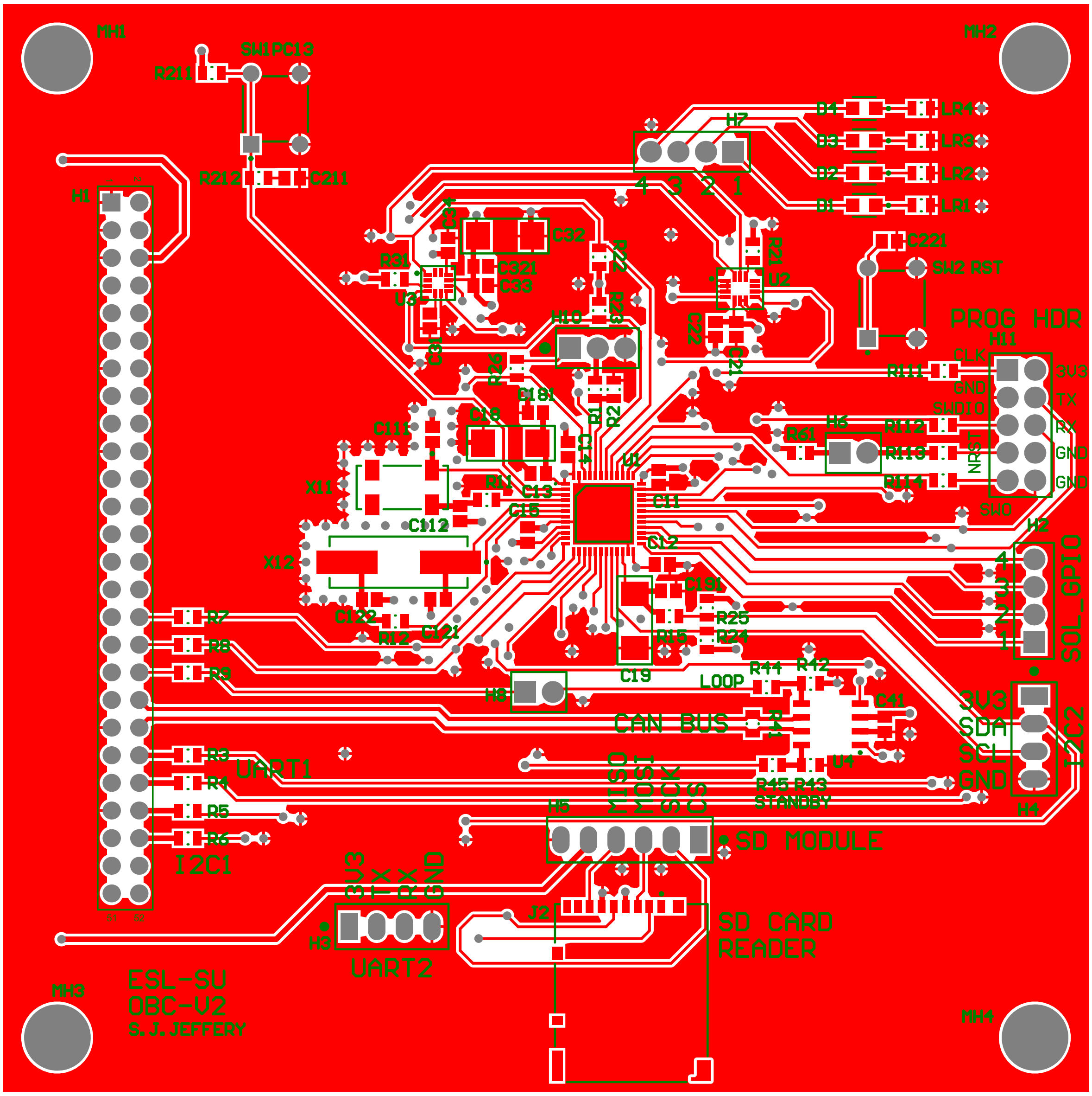
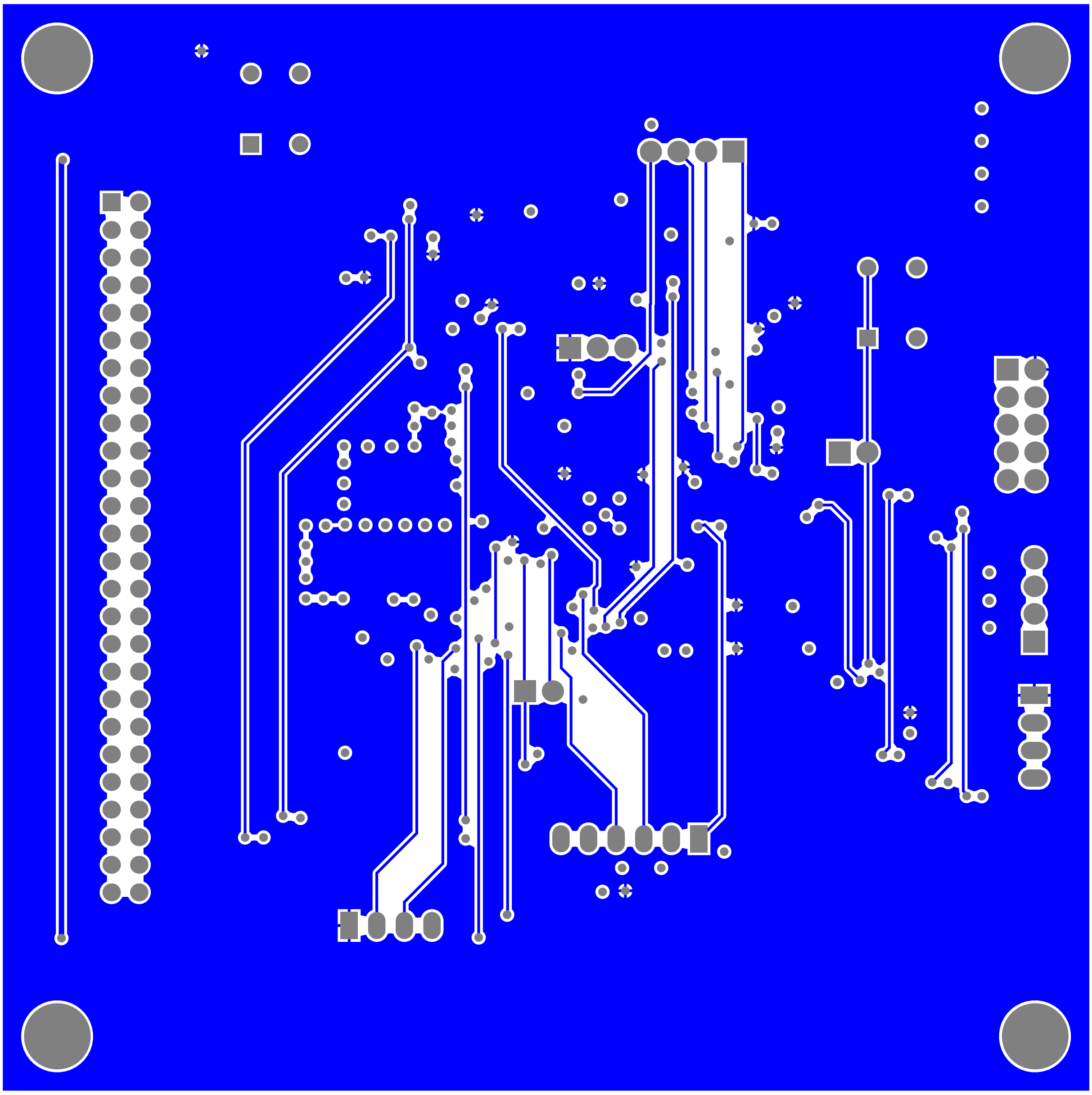


Figure A.5: OBC Schematic Diagram

A.3.2. PCB Layouts



(a) OBC PCB Top Layer



(b) OBC PCB Bottom Layer

Figure A.6: OBC PCB