



EXAMENSARBETE INOM ELEKTROTEKNIK,
AVANCERAD NIVÅ, 30 HP
STOCKHOLM, SVERIGE 2018

Functional Testing of the MIST Satellite

MALTE GRUBER

Acknowledgements

I would like to thank my thesis supervisor Mykola Ivchenko for his support during this thesis. I would also like to thank the MIST team for a fun and productive semester. A special thanks to Sven Grahn, Agnes Gårdebäck, Theodor-Adrian Stana and Andrii Berezovskyi for the interesting and productive discussions regarding the thesis work and the MIST project as a whole.

Abstract

The MIST satellite is a 3U CubeSat being developed at KTH, with a large part of the work being carried out by student teams. In this thesis methods for testing the MIST satellite are proposed and some of the methods are implemented and evaluated. Two pieces of testing hardware are implemented and evaluated. One of these is an experiment simulator that emulates the electrical load of an experiment subsystem. The second hardware unit is a sun sensor simulator that can feed the satellite artificial illumination data by simulating the current output of the satellite's sun sensors. A method for creating automated test and checkout procedures in a distributed testing environment using modern web-development tools is proposed. This method is demonstrated by showing how it can be used to verify the operation of the experiment simulators.

Sammanfattning

Satelliten MIST är en 3U CubeSat som utvecklas vid KTH, utvecklingsarbetet drivs i huvudsak av studenter. I detta examensarbete föreslås, implementeras och utvärderas metoder för testning av MIST satelliten. Testutrustning bestående av två hårdvaruenheter föreslås, utvecklas och utvärderas. En av hårdvaruenheterna är en experimentsimulator som kan emulera den elektriska lasten från satellitens experiment. Den andra hårdvaruenheten är en solsensormockup som kan mata satelliten med artificiella belysningsdata genom att härma strömmen från satellitens solsensorer. En metod för att skapa automatiserade tester för MIST satelliten föreslås. Denna metod grundar sig på moderna web-utvecklingsverktyg. En demonstration av metoden visas tillsammans med experimentsimulatorerna.

Acronyms

- ADC** Analog to Digital Converter.
- ADC** Attitude Determination and Control.
- CLI** Command Line Interface.
- CSKB** CubeSat Kit Bus.
- DAC** Digital to Analog Converter.
- EEPROM** Electrically Erasable Programmable Read-Only Memory.
- EPS** Electrical Power System.
- EQM** Engineering Qualification Model.
- FM** Flight Model.
- I2C** Inter-Integrated Circuit.
- IGIS** Innovative Solutions In Space Generic Interface System.
- LSB** Least Significant Bit.
- MOSFET** Metal Oxide Semiconductor Field Effect Transistor.
- MPPT** Maximum Power Point Tracking.
- OBC** On Board Computer.
- PCB** Printed Circuit Board.
- SPI** Serial Peripheral Interface.
- UART** Universal Asynchronous Receiver-Transmitter.

Contents

1	Background	1
1.1	The CubeSat standard	1
1.2	Functional testing	1
1.3	The MIST project	1
1.4	The SEAM project	2
2	Technical background	3
2.1	Solar panel simulator	3
2.2	The MIST satellite	4
2.2.1	Onboard computer	4
2.2.2	FlatSat and the CubeSat Kit Bus	5
2.2.3	Attitude Determination and Control	5
2.2.4	Magnetorquer	6
2.2.5	Sun sensors	6
2.2.6	OBC daughterboard	7
2.2.7	Electrical Power System	7
2.2.8	IGIS board and daisy chain harness	8
2.3	European Cooperation for Space Standardization	8
2.4	RAMSES	9
2.5	Elveti	9
2.6	CubeSat failure modes	9
2.7	Constraints and limitations	9
3	Proposed testing approach	11
3.1	Motivation for testing	11
3.2	Equipment for testing system	11
3.2.1	Experiment simulator	12
3.2.2	Breakout board	12
3.2.3	Sun sensor simulator	13
3.3	Test manager	13
4	Experiment simulator	15
4.1	Principle of operation	15
4.2	Power resistor configuration	17
4.3	Experiment simulator software	19
4.4	Performance and evaluation	19

5	Sun sensor simulator	23
5.1	Injection of sun sensor stimuli	23
5.2	Initial assumptions	23
5.3	Sun sensor simulator circuit	24
5.4	Changes after new information	25
5.5	Error sources	25
5.6	Performance and evaluation	26
6	Breakout board	29
6.1	Voltage monitor	29
6.2	Schematic	29
7	Measurement of sun sensors	33
7.1	Characterization method	33
7.2	Sun sensor soldering	33
7.3	Measurement procedure	34
7.4	Measurement results	35
7.5	Consequences	36
8	Software stack	39
8.1	Network and communication	39
8.2	Experiment simulator interface	39
8.3	Test manager	40
8.4	Evaluation using experiment simulators	41
9	Dynamic load of digital systems	43
9.1	ARM based micro computer	43
9.2	Dynamic load of AX100 UHF radio	43
10	Future work	45
10.1	ADC testing	45
10.2	Experiment simulators dynamic loads	45
10.3	Creating test cases	45
10.4	Computer readable OBC debug messages	46
10.5	Test equipment assembly	46

1 Background

1.1 The CubeSat standard

The CubeSat standard was first introduced in 1999 and is a standard defining the form factor of so-called nanosatellites. As the name suggests, the CubeSat standard is based on the concept of cubes. Each cube measures 10x10x10 cm with a maximum allowed mass of 1 kg [1]. The CubeSat standard also imposes limitations regarding the materials and substances that can be used in a CubeSat. CubeSats are launched as "piggybacks" on launches of larger satellites [2].

Once in orbit, the satellites are deployed from the launch vehicle using deployers, the deployers are containers that hold the satellites on rails. A large spring ensures that a satellite is pushed out of the deployer once the deployer door opens [2]. A CubeSat deployer is shown in Figure 1.1. While in the deployer, the satellite has to be completely powered down with the exception of real-time clocks. The satellite is only to be activated after it is deployed, the satellite is enabled with one or more deployment switches. After deployment, the satellite has to adhere to a 45-minute radio silence [1].

Once in orbit, a CubeSat is no longer under any size restriction and is free to deploy antennas, solar panels, and other contraptions. CubeSats are gaining in popularity and the number of CubeSats is expected to grow rapidly in the coming years.

1.2 Functional testing

The term functional testing is broad and can be taken to mean many things. In the context of this thesis, it is used to describe the process of testing space hardware using testing tools. The purpose of these tests is to ensure that the hardware is functional and that it meets its specified requirements. Hardware in this context can mean hardware such as components in subsystems, subsystems themselves or the spacecraft as a whole.

1.3 The MIST project

The MIniature Student saTellite (MIST) is a 3U CubeSat that is being developed at the Royal Institute of Technology KTH. The satellite is a student project and much of the work is done by students. The MIST satellite has

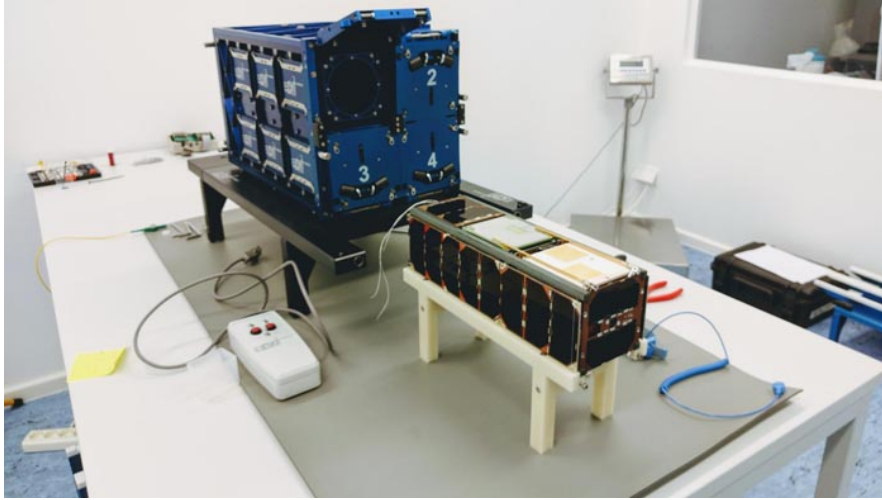


Figure 1.1: A CubeSat deployer for four 3U satellites. The deployer spring is in its deployed position and can be seen in the open hatch. In front of the deployer sits a 3U CubeSat.

seven payloads in the form of experiments, both from KTH and external organizations. RATEX-J is a solid state radiation detector. It will be delivered by the Swedish Institute of Space Physics IRF. Piezo LEGS is delivered by Piezomotor AB in Uppsala and is a linear piezo-electric motor. CUBES is a KTH experiment that will measure radiation with a silicon photomultiplier based detector. SiC-in-space is delivered by KTH ICT and aims to test Silicon carbide semiconductors. MoreBac is an experiment from KTH that will attempt to resuscitate bacteria once in orbit. SEUD is delivered by KTH and aims to study single event upsets. NanoProp is a cold gas propulsion system delivered by NanoSpace in Uppsala [3].

1.4 The SEAM project

The Small Explorer for Advanced Missions (SEAM) is a project partially funded by the European Union [4]. It is a 3U CubeSat designed to measure electric and magnetic fields in the ionosphere. The satellite has two deployable booms that are equipped with a fluxgate magnetometer and an electric field sensor [5]. The development of the satellite is led by KTH.

2 Technical background

This chapter presents the technical background of matters relating to the thesis work. Previous work done on testing equipment for the MIST satellite is introduced, and a description of the core systems of the MIST satellite is given. Finally, constraints and limitations on the thesis work related to equipment availability are discussed.

2.1 Solar panel simulator

The solar panel simulator is a system for simulating the power output of the satellites solar panels. The system can be divided into a software part and a hardware part. The software consists of a simulator with an orbit propagation for determining the location of the satellite. To determine the illumination of the satellite, ray tracing is used. The ray tracing takes into account the shadowing of the satellite due to solar panels and other protruding parts of the satellite. The simulator software contains an electrical model of a solar panel and can produce the solar panel voltage-current curve for multi-cell configurations. An illustration of the solar cell curve from a single solar cell is shown in Figure 2.1. The product of this simulation is a vector containing the current-voltage curve. This vector is sent to the solar panel simulator hardware.

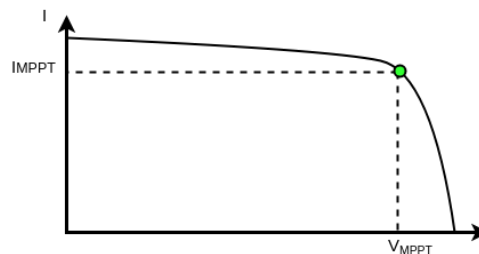


Figure 2.1: The typical IV-curve of a single solar cell and its maximum power point.

Each solar panel simulator is controlled by a microcontroller with a connection to the computer running the simulation software. The microcontroller generates a voltage which is used as a control voltage for a current source. The microcontroller continuously samples the output voltage of the current source using its built-in analog to digital converter. For every sample, the current-voltage vector is used as a lookup table. The measured voltage is

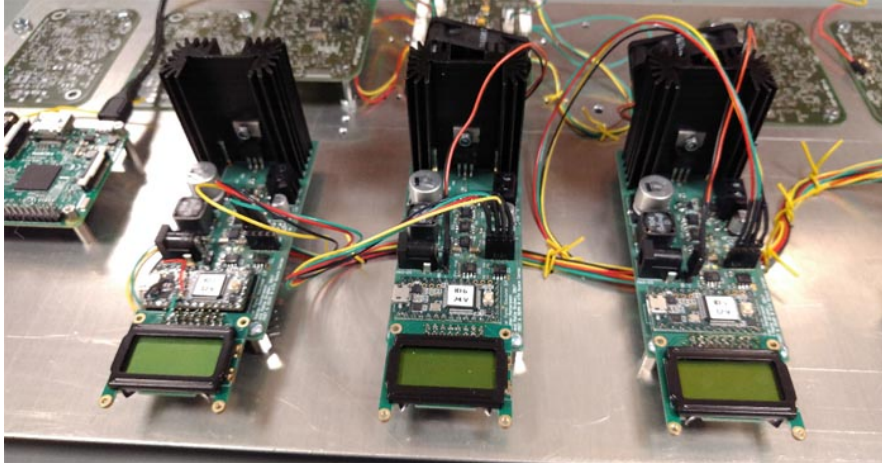


Figure 2.2: The three solar panel simulators developed for the MIST project.

compared to the lookup table and the target current is set to the current needed to stay on the current-voltage curve. The output current is set to this target current. This way the solar panel simulator can simulate the electrical behavior of one or more solar cells. The simulator has been tested with the electrical power system of both the MIST and the SEAM satellites. The maximum power point trackers were able to track the simulated maximum power point on both satellites. In the MIST project, three simulators are used, one for each input channel of the p31u EPS. The simulators are shown in Figure 2.2

2.2 The MIST satellite

In this section technical information about the core systems of the MIST satellite is provided. The description focuses on systems that are important in the context of testing and is not intended to be a complete description of all subsystems of the satellite.

2.2.1 Onboard computer

The onboard computer (OBC) manages many aspects of the satellite's operation such as collecting telemetry, interpreting telecommands and running the calculations for Attitude Determination and Control (ADC). The OBC hardware is supplied by Innovative Solutions In Space and is a single PCB measuring 95.93 x 90.14 mm. The PCB has a header for integration into the CubeSat Kit Bus described in subsection 2.2.2. The main processor of the OBC is the Atmel ARM9 AT91SAM9G20 running at 400 MHz. The OBC has a second supervisory microcontroller which is always on and acts as a watchdog

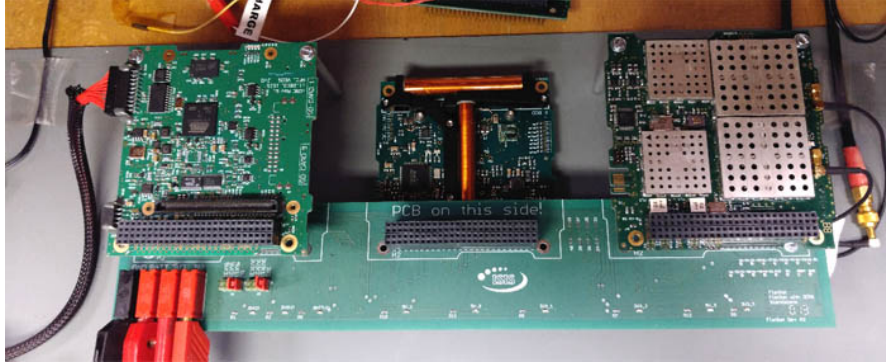


Figure 2.3: FlatSat Assembly of the MIST satellite, to the left is the OBC, in the middle is the disconnected magnetorquer and to the right the UHF radio.

as well as a power manager. The OBC has 32 MB SDRAM, and the program code is stored on 1 MB of NOR Flash memory. Critical data can be stored on a 256 kB FRAM memory. There are two redundant switchable micro SD-cards for non-critical bulk storage [6].

The OBC has several connectors. A connector relevant for testing is the connector J1 which provides access to a JTAG interface as well as a debug UART interface. This interface can be used for displaying debug text to the user during development.

The OBC PCB can hold a daughter board which provides access to more pins of the OBC main processor. This board is described in subsection 2.2.6

2.2.2 FlatSat and the CubeSat Kit Bus

In the context of the MIST project, the term flatsat is describing an assembly where the satellite components are connected but not fully integrated. The purpose of a flatsat is to provide easy and safe access to the subsystems. In the MIST project, a large PCB is used to connect three subsystems horizontally as shown in Figure 2.3. The subsystems that connect to the PCB are intended to be stacked in the integrated satellite. The stack is loosely based on the PC/104 standard. In the context of CubeSats, the stack connector is referred to as a CubeSat Kit Bus and is a 4x26 pin header connector [7].

2.2.3 Attitude Determination and Control

The Attitude Determination and Control system is responsible for keeping the satellite as close as possible to its reference orientation. It acquires the attitude of the satellite using sun sensors described in subsection 2.2.5. The

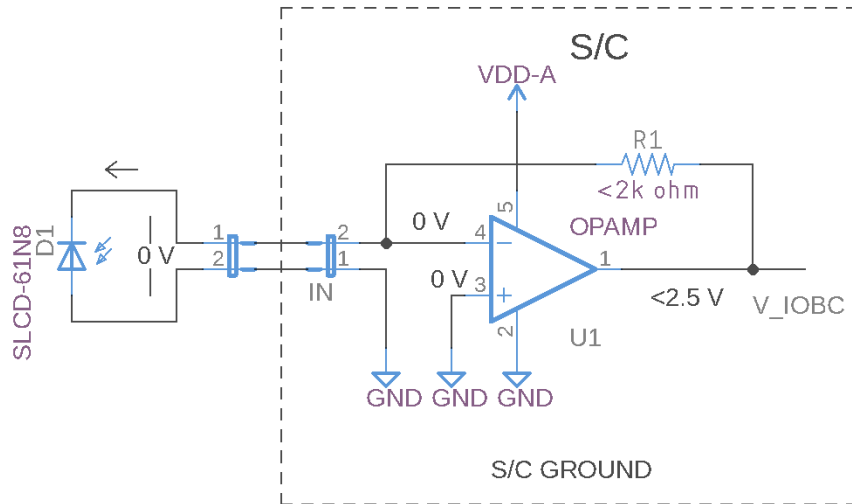


Figure 2.4: Schematic of possible sun sensor current-to-amplifier circuit using transimpedance amplifier.

measured illumination is supplied to a control algorithm and the algorithm calculates the adjustment needed to point the satellite in its reference direction. The output of the algorithm is fed to the magnetorquers that physically rotate the satellite, and then the process repeats. In the MIST project, the ADC algorithm will run on the OBC.

2.2.4 Magnetorquer

The MIST satellite has three magnetorquers that will be used to orient the spacecraft. The magnetorquers are the main actuators of the ADC system. The magnetorquers are mounted on a PCB shown in Figure 2.3. A magnetorquer is made up of a coil. If the coil is in a magnetic field, any current flow through the coil will result in a torque on the coil. This torque can be used to orient the spacecraft in low earth orbit where the magnetic field of the earth is strong.

2.2.5 Sun sensors

The MIST satellite has a total of seven sun sensors. These sensors are mounted on all sides of the satellite, with one per side. The sensors are a part of the Attitude Determination and Control system and are used to determine the orientation of the satellite. The sun sensors are photodiodes with model number SLCD-61N8.

2.2.6 OBC daughterboard

The OBC daughterboard is a PCB that mounts to the OBC described in subsection 2.2.1. There are different connectors on the FM and EQM versions of the daughterboard. The flight model version provides wire-to-board connector access the OBC board; the engineering model provides access using a 2.54 mm pitch header connector as well as USB type-A and type B connectors. Relevant to this thesis is that the daughterboard holds the current to voltage converters that are used to convert the currents from the sun sensors to voltages which the OBC can sample with its analog to digital converters.

The daughterboard uses a transimpedance amplifier to convert the currents from the sun sensors into voltages. The anodes of the diodes connect to satellite ground, the cathodes connect to the OBC daughter board.

The voltage for a given current $i_{cathode}$ can be described as

$$v_{ADC} = i_{cathode}G$$

where G is the gain of the amplifier. Two plots were provided by Innovative Solutions In Space, labeled "1k2" and "750R". The gain in these plots are $G = 1200VA^{-1}$ and $G = 750VA^{-1}$ respectively. A good match for these plots is the equivalent circuit schematic shown in Figure 2.4. The gain of this circuit is

$$v_{ADC} = i_{cathode}R_1$$

2.2.7 Electrical Power System

The electrical power system (EPS) of the MIST satellite is the GOMspace p31u and consists of a PCB measuring 96 x 90 mm. This PCB has a header for integration into the CubeSat Kit Bus [8]. The purpose of the EPS is to convert the power generated by the solar panels to power that the rest of the satellite can use. It also ensures that the solar panels produce as much power as possible with a technique called maximum power point tracking (MPPT). Maximum power point tracking ensures that the EPS draws current from the solar panels in such a way as to produce a voltage that maximizes the power output, this is illustrated in Figure 2.1. The p31u used in MIST has an onboard battery that is charged by the power from the solar panels. The battery voltage is converted to multiple voltages that power the various subsystems of the satellite.

Table 2.1: The configuration of the switchable output channels of the EPS and their mapping to the experiment subsystems.

Channel	Voltage	Current limit	Subsystem
1	3.3V	2A	Cubes
2	5V	2A	MoreBac
3	5V	2A	SiC
4	5V	2A	NanoProp
5	5V	2A	RatexJ
6	3.3V	2A	SEUD

When ordering a p31u EPS, several parameters have to be specified. The p31u for the MIST satellite has six output channels with, the configuration of these channels is shown in Table 2.1.

Since the EPS is the first system to turn on and the last system to turn off the main pull-up resistors of the I²C bus are connected to an internal 3.3 V rail of the EPS.

2.2.8 IGIS board and daisy chain harness

Innovative Solutions In Space Generic Interface System (IGIS) is a PCB that mounts on top of the CSKB stack. It provides access to the various pins of the CSKB in the form of wire-to-board connectors. Relevant to this thesis are the so-called top and bottom stack connectors. These connectors connect the experiments to the EPS and the OBC. The connectors are the start of two daisy chains, one for the top stack and one for the bottom stack. The purpose of these daisy chains is to limit the number of cables connecting to the IGIS board. Each experiment that is not at the end of a chain has an input and an output connector. The experiment passes the connection from the IGIS board or the previous experiment to its output connector, pins going to the experiment are not passed on.

2.3 European Cooperation for Space Standardization

The European Cooperation for Space Standardization (ECSS) is an organization that provides specifications for space-related activities [9]. The ECSS divides its standard into four main branches. **M**: Space project management branch. **Q**: Space product assurance branch. **E**: Space Engineering branch. **U**: Space sustainability branch. While many of these standards are of interest to the MIST project the standards most relevant to this thesis are in the Q and E branches. In the Engineering standards ECSS-E-ST-00 there are several standards about testing. The ECSS-E-ST-10-03C details the standards for testing

of space products. The document describes the general requirements of a test program including documentation, tolerances and retesting. The standard ECSS-E-ST-70-32C describes the Procedural Language for Users in Test and Operations (PLUTO) and its programmatic details. ECSS-E-70-41A defines the CCSDS standard which is used for the MIST satellite's communication link.

2.4 RAMSES

RAMSES (Rocket and Multiple Satellite EMCS Software) is a tool for supporting satellite operations. It can be used during development and space operations. Central to RAMSES is a database where the structure of telemetry and telecommand packets are defined. RAMSES is distributed and uses a broadcast-style network that utilizes a UDP multicast based proprietary protocol. A PLUTO node can be added to RAMSES to support testing of the spacecraft during development [10]. The RAMSES software is used in the SEAM project.

2.5 Elveti

Elveti is the ground station software used in the MIST project. Elveti uses CCSDS packets for communication with the satellite. Elveti has a scripting functionality based on Python scripts [11] but it cannot run PLUTO procedures which are supported by the RAMSES ground support software.

2.6 CubeSat failure modes

Statistical analysis of CubeSat failures shows that issues with the EPS were the most common cause of failure after 30 days, causing 40% of the failures. The second most affected subsystems are the Radio and the OBC [12].

2.7 Constraints and limitations

At the time of writing this thesis, the MIST satellite is under development. While much of the system design is known, many design decisions have yet to be taken. The limited specifications have a direct impact on the work in this thesis as it is not trivial or sometimes not even possible to design or implement tests for a system in these early stages. Much of the hardware has yet to be delivered. The software for the OBC and the ground support software are yet to be implemented.

Many tests rely on being able to test a loop of some sort, such as sending a telecommand and getting a specific reply. These tests are not yet possible to carry out in a meaningful way. While it is possible to suggest testing strategies, it is not always possible to try these strategies in their final form. The focus of this thesis, therefore, is to define a strategy for testing the satellite. Where possible, tools for testing are developed, however, these tools are created to be flexible and allow for future changes. At times new information has become available during the thesis work which has changed the design.

3 Proposed testing approach

In this chapter, an approach to testing the MIST satellite is proposed and discussed. The chapter begins with a broad discussion about how testing could be approached for the MIST satellite. Based on this discussion, a number of hardware units are proposed. Furthermore, possible methods for data collection and test automation are discussed. It is important to note that this is not intended to be a specification but a motivation and background for work presented later in the thesis.

3.1 Motivation for testing

While there are many tests that a satellite can be subjected to, this thesis is limited to functional testing of subsystems and the integrated satellite. The focus of testing should be on system level tests with integrated subsystems. For instance, testing the EPS alone is valuable in terms of verifying the correct behavior of the EPS itself. However, running the EPS while integrated into the satellite could possibly reveal issues that only occur in an integrated environment. These system level tests should be conducted by generating stimuli and feeding it to the satellite. The satellite output should be collected and analyzed to verify compliance with specifications. It is important that tests can be carried out during environmental testing. This ensures that subsystems and subsystem interactions are working in situations that are close to the real operational conditions. Testing equipment must be easy to transport and should be able to run off-site.

There should be some centralized way to control and monitor tests. Tests should be automated and frequently executed to have a better chance of detecting rare, intermittent, faults. The nature of the ever-changing workforce in MIST makes this especially important as changes made to the satellite can be continually verified.

3.2 Equipment for testing system

The proposed layout of the testing system for MIST is shown in Figure 3.1. This layout uses a broadcast style network, this network includes the test manager described later in this chapter. Future nodes that are not within the scope of this thesis, such as an inertial model for testing the ADC software are also shown in Figure 3.1. The ADC testing hardware includes the sun sensor simulator described later in this chapter but also a magnetorquer sensor that

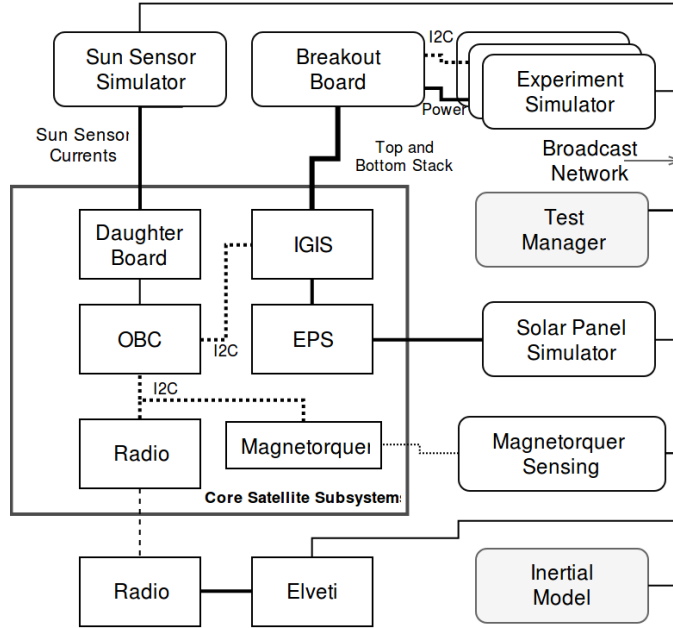


Figure 3.1: Possible layout of the test system for MIST including the proposed hardware.

measures the output of the magnetorquer. The magnetorquer sensor is not within the scope of this thesis.

3.2.1 Experiment simulator

Work is already underway to create simulators for the software side of the experiments but there exists no equipment for hardware simulation. The proposed solution is to construct experiment simulators that simulate the hardware aspects of the experiments. The experiment simulators should simulate the impact of the experiment on the electrical power system by drawing power in a way that mimics the real experiment as closely as possible. The experiment simulators should be able to facilitate communication with the satellite through the I²C bus using the same I²C buffer that is required to be included in the experiments. For controlling the simulators and gathering voltage and current measurements, a PC interface is needed.

3.2.2 Breakout board

To realistically simulate the experiments their cables should be included in the simulation. As described in subsection 2.2.8, the experiments are connected in a daisy-chain configuration. This configuration could cause issues related to contact resistance and interference between wires. Hence

it is important to verify that this connection scheme will work in practice. This can be done by creating a breakout board PCB that has connectors that are daisy-chained in the same way as the experiments. Each experiment would be represented on this PCB, and the experiment simulators could be connected to this PCB. The I²C lines should also be present to verify that the specified cable length will not cause issues.

3.2.3 Sun sensor simulator

When testing the ADC system, the sun sensors need to be illuminated. It is not practical to physically illuminate the sensors. Therefore a replacement for the sun sensor diodes should be constructed in the form of a sun sensor simulator. This simulator should produce the same current as the sun sensor diodes. The simulator should have some way of connecting to a PC for receiving the illumination data.

3.3 Test manager

If tests are to be automated as discussed in section 3.1, the testing system for the MIST satellite will need some form of central control. The ECSS Standard ECSS-E-ST-70-32C describes how the PLUTO language can be used to create procedural test scripts. The RAMSES ground control system described in section 2.4 supports PLUTO scripts. PLUTO Scripts are used in the SEAM project but they are limited to sending telecommands and receiving telemetry over a CCSDS compatible communication channel. In the MIST project, the Elveti ground support software is used which does not support PLUTO scripts. Elveti supports python scripts, but they are primarily intended to send telecommands.

For the MIST project, there exists no centralized way of commanding the custom test equipment. The test equipment does not support CCSDS or standardized ground support protocols which are supported by RAMSES or Elveti. In addition, the Elveti software is already procured and acquiring additional software would be costly.

To integrate the test equipment into an existing, closed source, ground support software suite could take a considerable amount of time and effort. Therefore a custom test manager could be beneficial, both in terms of labor-hours and cost. This test manager could send commands to the satellite or any piece of test equipment and monitor responses to verify that the satellite operates normally.

This test manager would not be able to conduct complicated tests such as verifying that the ADC algorithm returns nominal results but it would be

useful for tests that issue a command which generates a response that can be verified. If the test manager could communicate with all available test equipment, the satellite, and the ground station equipment, a wide variety of tests could be conducted.

4 Experiment simulator

In the previous chapter, several pieces of testing hardware were proposed. Among these were the experiment simulators. In this chapter, the experiment simulators are described in detail. The experiment simulators are credit-card sized PCBs that each simulates one experiment subsystem. Their purpose is to be placeholders for the real experiments during the development of the satellite as described in subsection 3.2.1. The experiment simulators can simulate the experiments power demands by dynamically drawing power from the EPS. In total there are six simulator boards, each representing one experiment connected to a switchable output channel of the EPS.

4.1 Principle of operation

The experiment simulators are made up of individual PCBs, one of which is shown in Figure 4.1. An experiment simulator is made up of two electrically isolated sections. One side has a USB connection which allows access to a USB-to-UART converter, the other is the experiment simulator electronics. The two sides are linked with a bidirectional digital isolator that carries the signals of the UART interface, the schematic is shown in Figure 4.4. A microcontroller controls an electrical load and measures its voltage and current. The experiment simulator electronics are made up of the following parts:

Electrical Load: The main feature of the experiment simulator is the ability to emulate the power consumption of the experiments with an electrical load made up of resistors and switching transistors. The resistors each have one end connected to the incoming EPS power channel; the other side is connected to the drain of one MOSFET switching transistor. The source of the transistor connects to ground and the gate connects to a GPIO pin of the microcontroller. When the gate is pulled low the power resistor is effectively disconnected from the EPS power channel, when the gate is high the resistor is connected to ground and draws current from the EPS. The exact configuration of the resistors is described in section 4.2.

The electrical load is designed to support resistor configurations that draw a substantial amount of current. The MOSFET transistors (DMG3414U) were chosen for their fast switching times of less than 10 ns and low on-resistance of 29 m Ω at a gate voltage of 2.5 V [13]. In addition, the most significant resistance is made up of three resistors, each being controlled

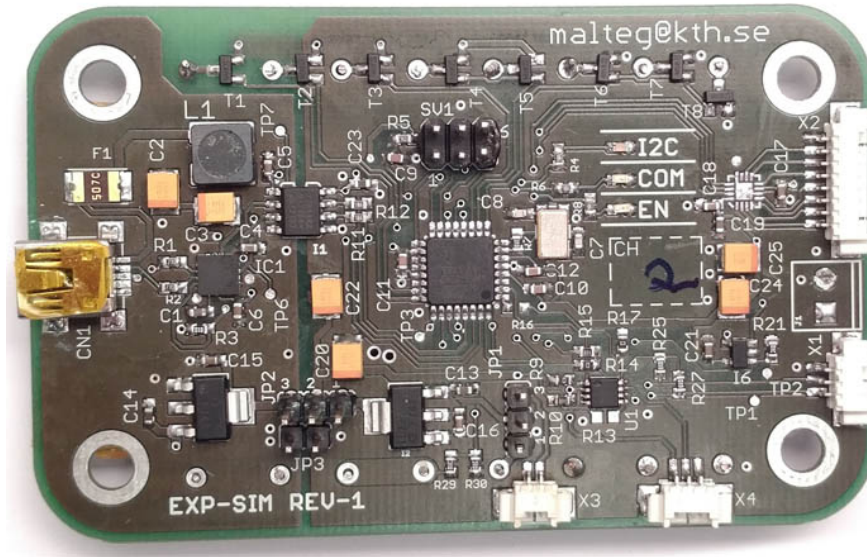


Figure 4.1: Top view of the experiment simulator PCB.

by an individual transistor, the second most significant resistance has two resistor-transistor pairs. In total there are eight transistors and five switchable channels. The exact configuration is shown in Figure 4.5.

Microcontroller: Central to the experiment simulator is a microcontroller; this microcontroller is the ATMEGA328PB. The microcontroller has the Arduino UNO bootloader and can be programmed with the UART interface using the Arduino IDE. The microcontroller is connected to an ISP header which is used to flash the bootloader. The reset line of the microcontroller is connected to a resistor-capacitor divider. This divider will pull the reset line low at power-up. As the resistor charges the capacitor the reset line will go high. The schematic of the microcontroller is shown in Figure 4.3. The microcontroller actually used for the experiment simulators is the ATmega328PB. However, the intended microcontroller is the ATmega328P which is used in the Arduino UNO. This microcontroller is being replaced with the ATmega328PB and was not in stock. Therefore the PB variant was used instead. It is backward compatible with the P version [14].

I²C buffer: The experiments are required to have a I²C buffer, this buffer is also present in the experiment simulator. The buffer is the TCA4311A from Texas Instruments, its pinout is shown in Figure 4.2. The EN and READY pins are intended to be connected to the host [15]. In the experiment simulator, they are instead connected to the microcontroller of the experiment-simulator. The I²C buffer can be powered in two ways; either directly from

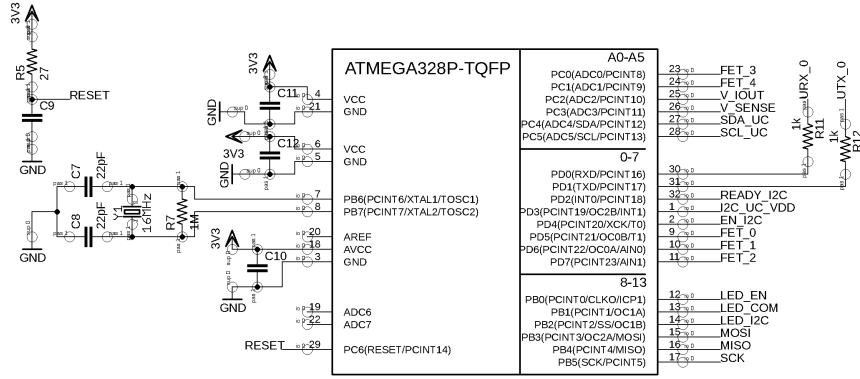


Figure 4.3: Schematic of the microcontroller and the reset circuitry.

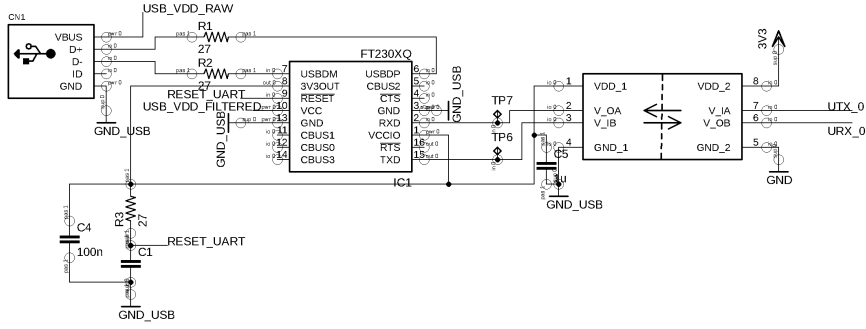


Figure 4.4: To the left is the USB connection which connects to a USB-UART converter. The UART lines connector to the bidirectional digital isolator to the right.

Where $\xi(i)$ represents the corresponding bit in the binary control number M and the state of its MOSFET

$$\xi(i) = H(M \& 2^i)$$

$$H(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$$

V_{bus} is the voltage of the p31u output channel.

M is the setting of the resistor array.

$\&$ is a bit-wise AND operation.

N is the number of switchable resistor channels

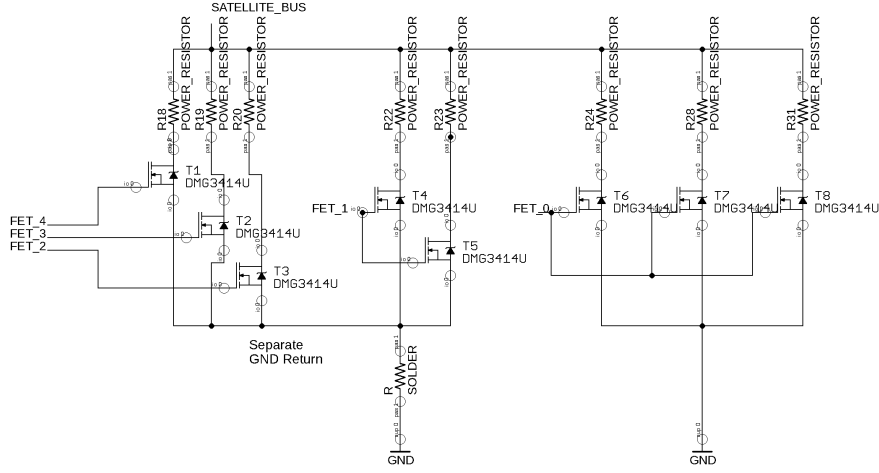


Figure 4.5: The power resistors and the switching transistors of the electrical load.

The total current drawn from the output channel is described by

$$I_{total} = \sum_{i=0}^{i < N-1} \frac{V_{bus}}{R_i} \xi(i)$$

4.3 Experiment simulator software

The experiment simulators can be commanded over a Command Line Interface (CLI) using the USB-UART connection. The commands start with a single character that is the command ID. If the command has an argument it is appended as an integer value, the command is terminated with a newline character.

Each experiment simulator has a unique ID that is used to identify the PCB when connected to a host computer. This ID is stored in the microcontrollers EEPROM. To set the ID, an EEPROM unlock command has to be sent with a key that prevents accidental changes of the experiment simulator ID. The key is 14120.

4.4 Performance and evaluation

The measured load of the experiment simulator is shown in Figure 4.6 . The board is populated with resistors that provide a full-scale resistance of 18.75Ω , the input voltage representing an EPS channel was 3.3 V. In this configuration the maximum load of the resistors is 170 mA (Some power

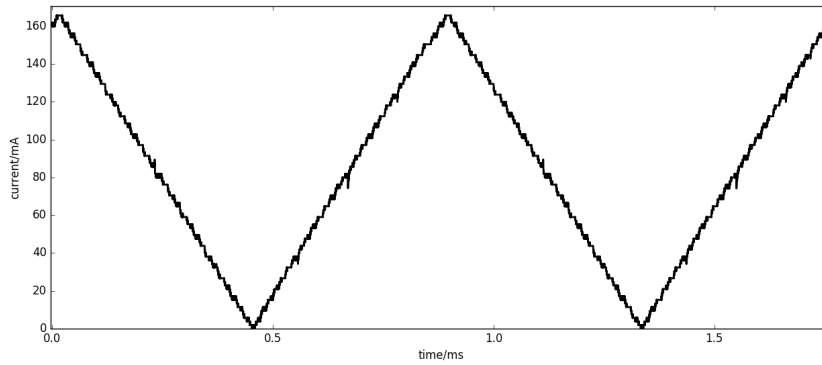


Figure 4.6: The dynamic current draw of an experiment simulator. In this demonstration, a triangle wave is generated by the experiment simulator. The individual steps of the 5-bit resistor network can be seen as steps in the plot.

is lost in the cables). For this demonstration, the board is programmed to output a triangle wave load. The fastest full-scale switching speed without any software optimizations is shown in Figure 4.7. The experiment simulator has been tested with resistors that have a full-scale load of 2 A which is the maximum of any EPS channel as shown in Table 2.1.

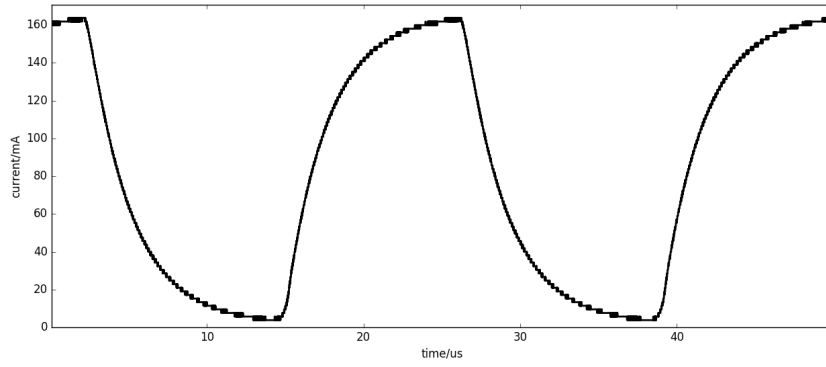


Figure 4.7: Demonstration of the experiment simulators switching capability. Here the load is switched from zero to full scale at a high rate.

5 Sun sensor simulator

In chapter 3 the need of injecting sun sensor readings into the satellite was discussed. In this chapter a simulator that simulates the electrical behavior of the sun sensors is presented.

5.1 Injection of sun sensor stimuli

It is possible to inject sun sensor stimuli in multiple locations. The first option is to remove only the sensors and replace them with a simulator that emulates the electrical behavior of the sun sensor diodes. The second option is to inject stimuli as voltages in place of the current to voltage converter on the OBC daughterboard. In this case the simulator would connect directly to the OBCs analog to digital converters. The third option is to do the injection in software. In this scenario, the OBC code for reading the analog to digital converters would be replaced by code that instead injects simulated readings.

Each method has advantages and drawbacks. As the chapter name suggests, the option that was chosen for the sun sensor simulator was to simulate the photodiodes themselves. This option requires a more complex circuit for the simulator compared to the other injection methods, however, it tests the complete chain from current-to-voltage conversion, analog to digital conversion and the software interface to the analog to digital converter.

5.2 Initial assumptions

At the start of this work, information about how the sun sensors interface to the satellite was limited and during this work, more information has become available. At the start of this work, the only available information was the model number of the sun sensors (SLCD-61N8). Therefore some assumptions had to be made. The initial circuit assumed a current flow into the current-to-voltage amplifier of the OBC daughter board. The second assumption was that the input voltage of the OBC daughter board would not change for changes in input current. Furthermore, it was assumed that the voltage at the input would be the same as the ground reference of the satellite. Put another way, the voltage across the photodiode was expected to be zero.

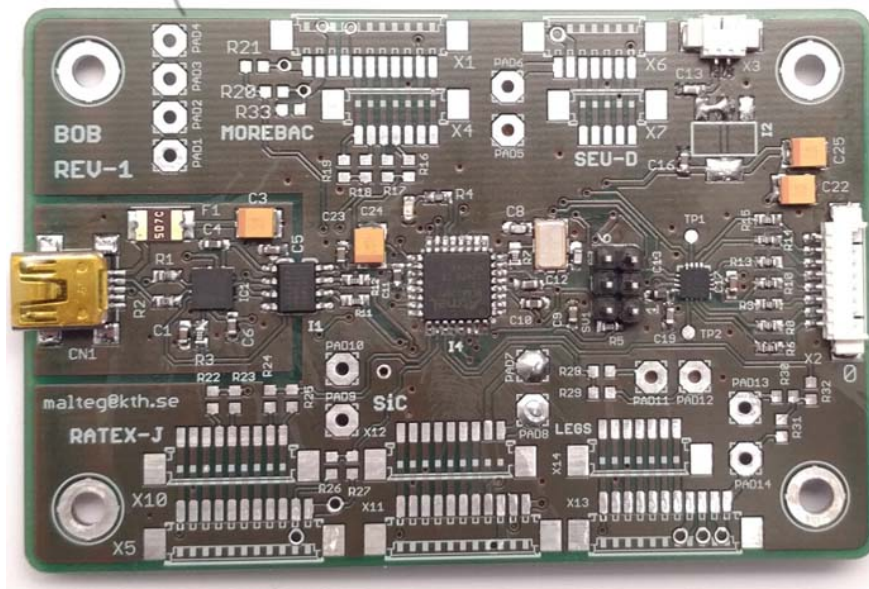


Figure 5.1: The sun sensor simulator PCB.

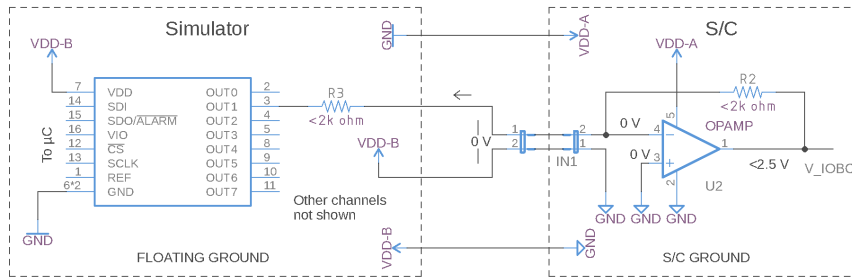


Figure 5.2: Schematic of sun sensor simulator after new information from Innovative Solutions In Space.

5.3 Sun sensor simulator circuit

The sun sensor simulator uses the same PCB as the breakout board described in chapter 6. This PCB uses a very similar configuration of the microcontroller, digital isolator and USB-UART converter configuration as presented in chapter 4 and will not be described in detail here. The sun sensor simulator PCB is shown in Figure 5.1. The two main components of the sun sensor simulator are the microcontroller and a Digital to Analog Converter (DAC). The microcontroller controls the DAC with an SPI interface. The DAC schematic is shown in Figure 5.3 and is the DAC70508MRTET which is a 14-bit voltage output R-2R DAC with a rail-to-rail capable output buffer with a short circuit current of 30 mA to VDD and 35 mA to GND [16]. Seven of the

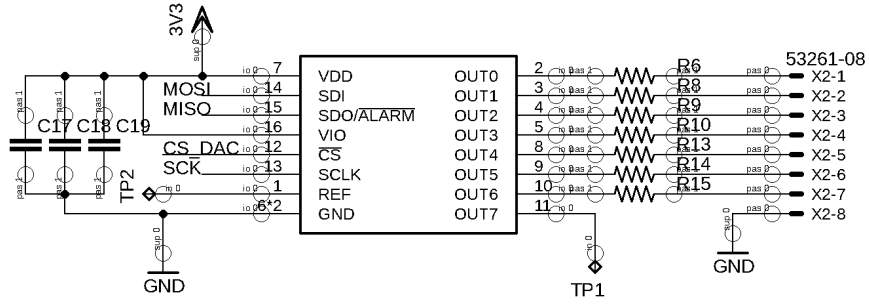


Figure 5.3: Schematic of DAC used for sun sensor simulator.

DAC channels are connected in series with a resistance R . Assuming that the current-to-voltage converter has a low input impedance, the current from a DAC output would be the same as the voltage of the DAC divided by the resistance of the series resistor $I_{SIM} = (V_{DAC} - V_{INPUT})/R$. This is valid since the voltage drop over the current-to-voltage converter is close to ground for all I_{SIM} , that is $V_{INPUT} = 0$.

5.4 Changes after new information

During this work, details of the sun sensor handling were made available in written communication with Innovative Solutions In Space. One of the assumptions described in section 5.2 was wrong: Instead of the anode, the cathode connects to the input of the daughter board. All other assumptions described in section 5.2 were correct. In addition, it was made clear that the circuit used for sensing the current is a transimpedance amplifier. While the exact circuit was not given the configuration shown in Figure 2.4 matches the provided information.

To avoid having to order new PCBs, changes were instead made to how the sun sensor simulator is connected to the satellite. Since the simulator section of the PCB is floating it is possible to choose what reference it should have with respect to the ground of the satellite. By choosing the ground reference such that the ground of the simulator is lower than the ground of the satellite the DAC will sink current from the satellite instead of sourcing it. This schematic of this configuration is shown in Figure 5.2.

5.5 Error sources

The configuration described in section 5.4 introduces new error sources. In the previous configuration, the main error source was the DAC. With the

changes to the simulator reference point, the main error source becomes the supply voltage. This can be explained as follows. Let the supply voltage be described as $V_{supply} = V_{set} + \Delta_{supply}$ where Δ_{supply} is the fluctuation in supply voltage. It is important to note that V_{supply} is floating with respect to the satellite and is always tied to the ground of the simulator.

In the initial configuration described in section 5.3, the simulator is less susceptible to differences in supply voltage Δ_{supply} since the simulator ground (GND_{SIM}) is the same as the satellite (GND_{SAT}). Hence the ground of the DAC's reference (GND_{REF}) is directly tied to the satellite ground. By assuming that the reference is independent of Δ_{supply} the variation in output current is zero.

By referencing the simulator at a point different from GND_{SIM} , GND_{REF} will change with respect to GND_{SAT} by $-\Delta_{supply}$. This means that the output of the DAC will also change with $-\Delta_{supply}$.

A second error can arise if the output of the DAC is not able to reach GND_{SAT} , then there will always be a current which cannot be removed. It is therefore important V_{supply} is chosen such that it is included in the DAC's output range (which is only dependent of the DAC's reference and independent of V_{supply}).

Since the precision of the DAC is typically better than one LSB [16] Δ_{supply} will have a significant impact on the performance of the sun sensor simulator and V_{supply} should be generated with a stable voltage source.

5.6 Performance and evaluation

The sun sensor simulator performance is shown in Figure 5.4. All seven channels are behaving linearly for currents less than 5 mA. The simulator is able to sink currents close to 0 mA. However, due to the limitations in output range discussed in section 5.5 there will always be a small stray current when the simulator is set to output 0 mA. With 680 Ω resistors this stray current was less than 6 μA for all channels. Further as shown in Figure 5.4 the simulator was able to sink 6 mA. This is four times more than the expected maximum current from the sun sensors. Hence, the resistors could be replaced with higher value resistors. This will lower the 6 μA stray current. The update speed of all eight channels of the ADC using the UART interface has been tested at an update rate of 40 Hz. This test was done with a text-based transfer using a command line interface on the sun sensor simulator. In other words, if needed the software can be optimized to achieve much higher update rates.

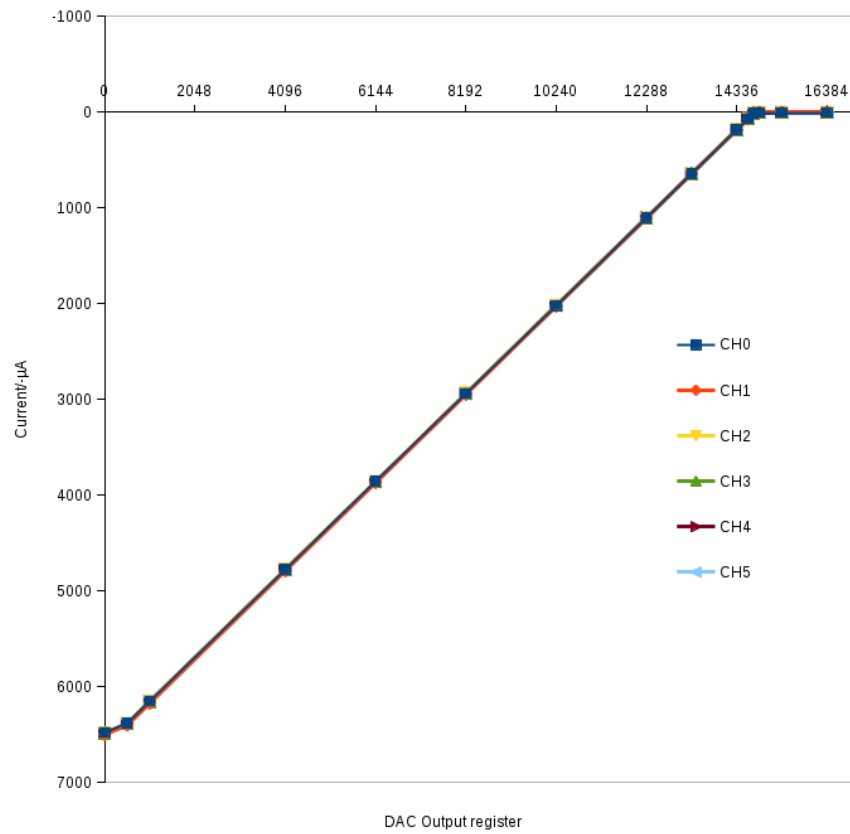


Figure 5.4: Full scale output range of the sun sensor simulator with 680 Ω converter resistors. The DAC reference is 2.5 V with a gain of 2.

6 Breakout board

The experiments of the MIST satellite are connected to the core of the satellite using a daisy chain of PicoBlade connectors. The daisy chain starts at the IGIS board described in subsection 2.2.8. From the IGIS board, two sets of daisy chains carry both power for the experiments and the connections for the I²C bus. The experiments that are not at the end of the daisy chain will each have one input and one output connector, an experiment subsystem will connect to the relevant power rails for the specific subsystem. The remaining channels and the I²C bus is passed to the output connector.

To verify that this configuration will not cause issues due to factors such as contact resistance or cable length and to organize the test equipment a breakout PCB was designed. This PCB simulates the daisy chain by providing connectors that are the same as the connectors of the experiments. The PCB provides a low impedance path to the power tap of point of each experiment and access to the I²C bus as shown in Figure 6.1.

6.1 Voltage monitor

The breakout board uses the same PCB as the sun sensor simulator described in chapter 5. The microcontroller used for the sun sensor simulator can also be used to sample the voltages on the top and bottom stack. The voltages are measured by connecting high resistance voltage dividers to the positive rails of the stack. Note that the ground of the breakout board is not connected to any point in the stack. The satellite ground and the breakout board ground need to be equal for voltage measurements to be available. The sensor board and the breakout board could be implemented on the same physical PCB. However, this is not advisable since the currents carried in the top and bottom stack could cause noise in the sun sensor simulator. Additionally with the grounding scheme described in section 5.4 the voltages cannot be sampled.

6.2 Schematic

The connections of the breakout board are shown in Figure 6.1. On the PCB the input and output connectors for the experiments are grouped per stack as shown in Figure 6.2. For the top stack, the IGIS top stack connector connects to the input connector of MoreBac (X1). The output connector of MoreBac is X4 which connects to SEU-D. For SEU-D the input connector is connector X6, and the output is X7 which connects to NanoProp. NanoProp

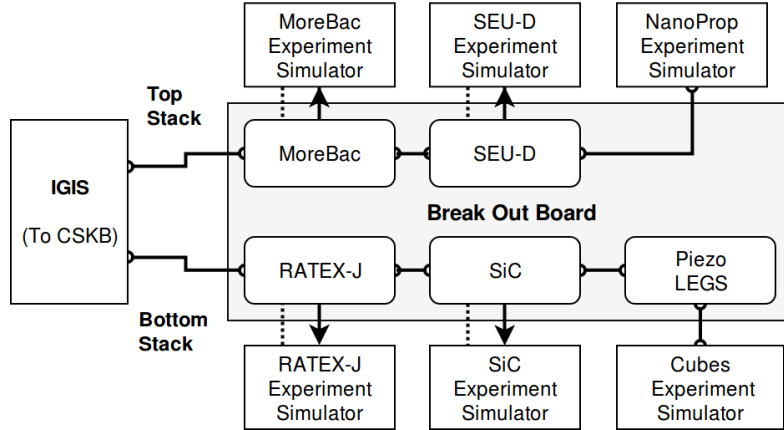


Figure 6.1: Illustration of the breakout board daisy-chain harnessing. The Arrows represent low impedance paths for passing power to an experiment simulator. The dotted lines represent the I²C connection. The lines with the half-circles represent the satellites harnessing. Note that LEGS is not simulated in this configuration since it does not have a dedicated EPS power channel.

is not represented on the PCB. The connection is instead made directly from the breakout board connector X7 to the experiment simulator with the same gauge cables that are used for the stack. This connector structure is the same for the bottom stack with Cubes being at the end of the chain as shown in Figure 6.1.

There are five connectors on the bottom of the breakout board that each connect to the I²C lines to experiment simulators represented on the PCB. The connector is pin mapped to the experiment simulator I²C connector shown in Figure 4.2.

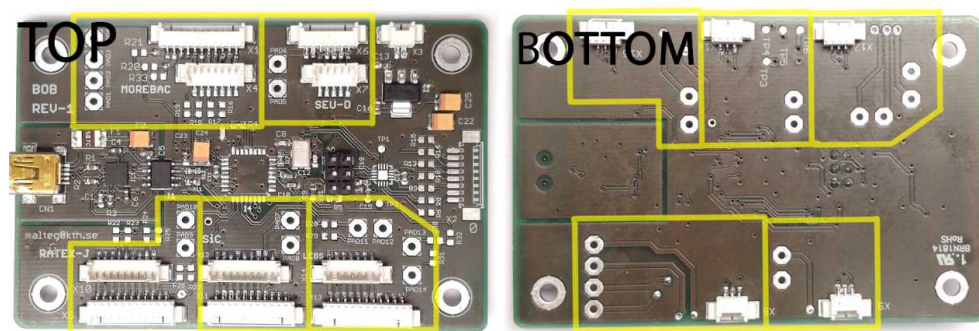


Figure 6.2: The top and bottom of the breakout board. The yellow overlays mark the connector groups for the five experiments represented on the board. For the upper row of the top view, they are MoreBac (Left), SEUD (Right). For the lower row, they are RATEX-J (Left), SIC (Middle) and LEGS (Right). For the top view the large with pico-blade connectors are the stack connectors. On the bottom view the I²C connectors for the experiment simulators can be seen.

7 Measurement of sun sensors

The satellite's primary source of attitude information comes from the sun sensors. It is therefore important that they are well understood. Ideally, the sensors should be measured in such a way as to confirm that the information given in the datasheet is accurate, however, characterizing an optical device is not trivial. In the case of this device, it is intended to be used in low earth orbit at an altitude of approximately 600 km. At this altitude, the atmosphere is very thin. To reproduce the full range of properties of the light at this altitude on earth is challenging. The sunlight that reaches the ground is filtered by the atmosphere which absorbs some wavelengths. The exact resulting spectrum will depend on environmental factors such as humidity and cloud coverage [17]. In low earth orbit, factors such as the sunlight reflecting of the earth and the moon will further complicate the reproduction of the lighting environment. The testing done in this thesis limits the lighting conditions to ground light by using sunlight on a clear day. These tests do not deal with absolute measurements of light intensity but rather comparing the sun sensors and their performance as well as evaluating how sensitive they are to misalignment due to soldering.

7.1 Characterization method

If the sun sensor diode is tilted relative to the incoming sun vector \vec{s} the area of the sensor that will see the light decreases. If the sun sensor with area A has a normal vector \vec{n} and assuming that the incoming sunlight is parallel the area of the sun sensor that will see the sunlight \vec{s} can be described as

$$A' = (\vec{s} \cdot \vec{n})A$$

This is illustrated in Figure 7.1 and is the method used for varying the amount of light reaching the sensor for these measurements.

7.2 Sun sensor soldering

The SLCD-61N8 photodiode is a small bare silicon chip. The bottom of the component is metalized and is connected to the cathode of the diode. The top side has an active area that is the light-sensitive diode junction. Next, to this, there is a small metallized pad which is the anode [18]. Three such chips were soldered to a piece of experiment PCB material. The solder mask

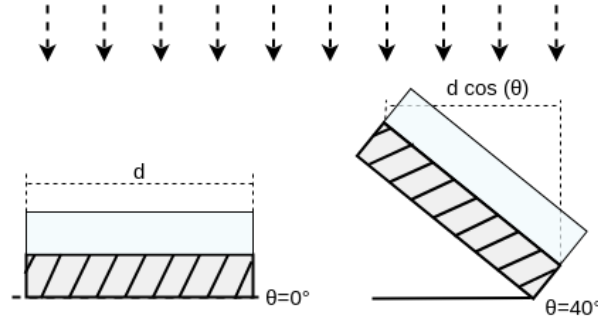


Figure 7.1: Illustration of how the angle of a sun sensor relative to the incoming light relates to the area of the diode exposed to the light.

was removed so that an area of copper, slightly larger than the diode, was exposed. The surfaces were checked for protrusions and solder was applied to the copper using a soldering iron. This solder was then removed using a copper de-soldering wick, leaving a thin layer of solder. On this layers, the diodes were placed after applying soldering flux. A small hot air gun was used to solder the sensors to the PCB with a temperature of 320°C. A small wire was soldered to the anodes of the chips, connecting them to individual pads on the PCB. Cables were soldered to the PCB so that the anode and cathode each had their individual cables. The result is shown in Figure 7.2

7.3 Measurement procedure

The sun sensor PCB described in section 7.2 was mounted to a piece of plywood measuring 100x4 cm. A rod was mounted to the board such that it was perpendicular to the surface, the height of the rod relative to the surface is 47 mm. A scale was added along the length of the board. The shadow cast by the rod on this scale indicates the sun angle. The scale has degree markings in five degree increments, the position of the markings was calculated using the following formula with $l = 47$

$$b = l \tan(2\pi(\theta/360))$$

Three multimeters were used to measure the current from the three sun sensors. To verify that the multimeters agreed, the three multimeters and a single sun sensor were connected in series. This way the same current passed through all three multimeters. The meters agreed with a maximum deviation of less than 1 μA . This was tested at both extremes of the measured sun sensor output at about 50 μA and 600 μA .

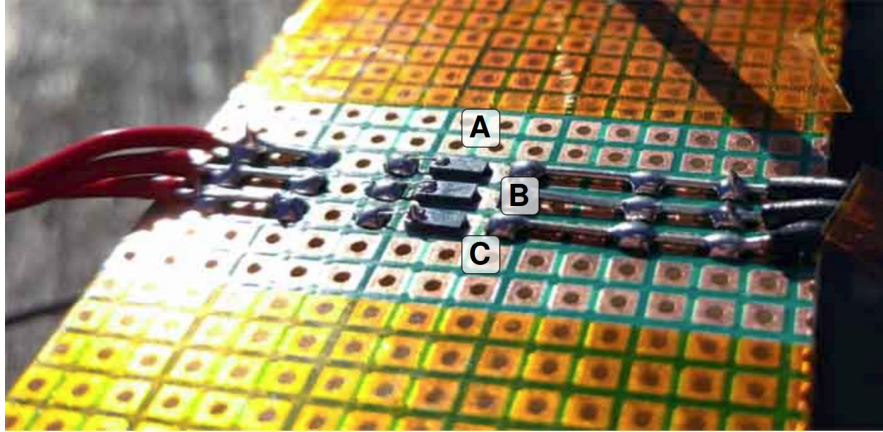


Figure 7.2: The three sun sensors A,B and C soldered onto a circuit board. The red cables to the left in the picture connect to the anodes of the sun sensors.

For the actual measurements, the indicator rod was used to place the sun sensors at an angle relative to the sun and the respective currents were recorded for this angle. Measuring all angles takes about 5 minutes, and the sunlight will change in this period, however, the sensor readings for a specific angle are all taken at the same time using the hold button on the multimeters.

7.4 Measurement results

The result of the measurements of the three sensors taken on a cloud-free day is shown in Figure 7.3. There are two things that stand out. Sensor A produces a significantly higher current than sensors B and C. The side of sensor A is not shadowed by other sensors as is the case for the remaining sensors. However, the difference is most notable at zero degrees and should therefore not be a consequence of shadowing from other sensors. Sensor A was re-soldered and it is possible that this is the cause of the difference, but it is curious that this would result in a higher output current.

In Figure 7.4 the difference between the sun sensors B and C is shown. The two series are from separate measurements made to verify that the difference is real. It seems that the difference is not constant but a function of the sun angle. This could indicate that the sensors are unevenly soldered despite the actions taken during the soldering process to ensure a flat and even application.

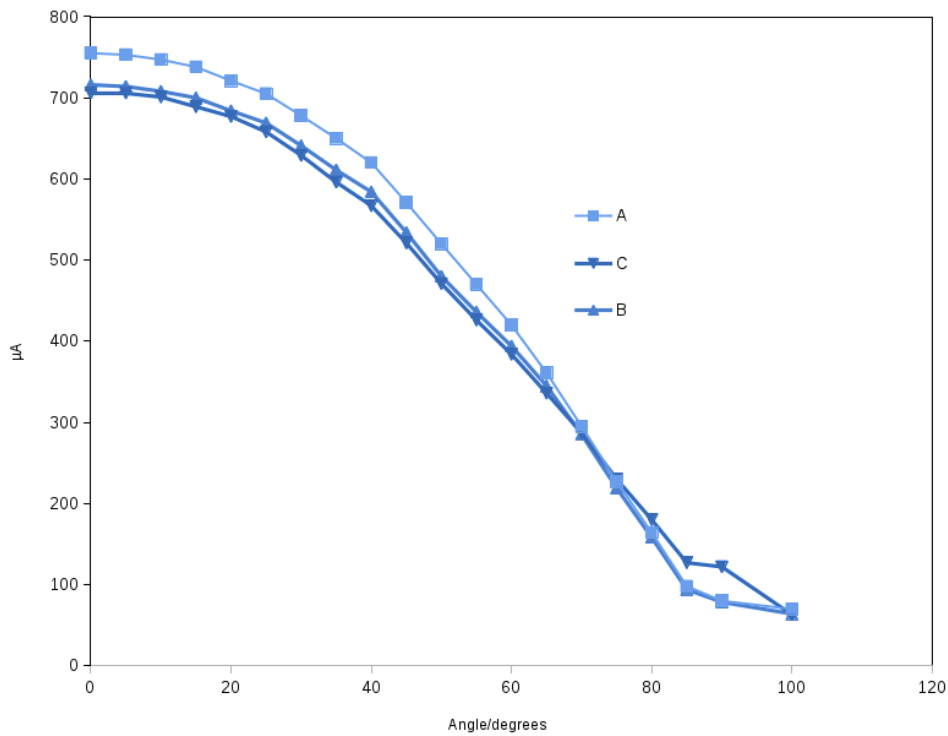


Figure 7.3: The current output of sun sensors A,B and C for angles between 0° and 90°.

7.5 Consequences

The natural next step would be to repeat this experiment with more sensors. However, the sensors are expensive and the results would not be very useful beyond showing the nature of the variation between sensors with statistical rigor. Instead, a more pragmatic approach is to carefully note the output of each sensor of the fully integrated satellite. Any offset should be calibrated out in the ADC code as to not cause pointing error due to an unknown bias caused by the sun sensors.

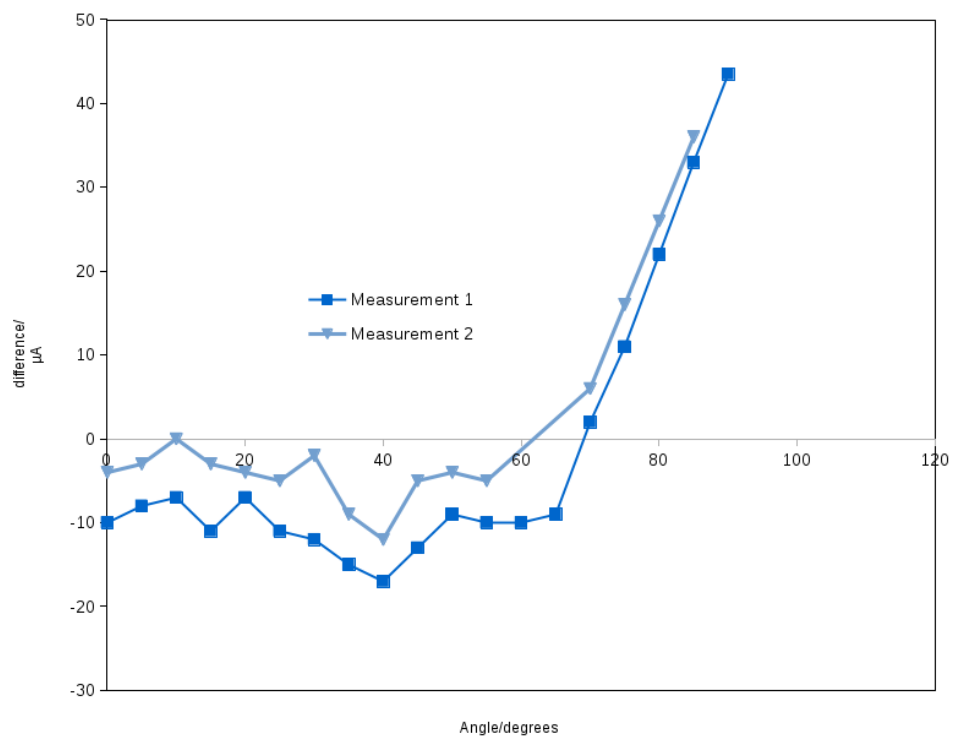


Figure 7.4: Difference between the current output of sun sensors B and C. The measurement were repeated and the results are shown as series 1 and 2.

8 Software stack

During this thesis work, a number of software components were developed. While they are not part of the primary objectives of this thesis they do provide insight into the testing strategy described in this thesis. The major software components are therefore briefly described here.

8.1 Network and communication

As the testing system grows connecting every piece of test equipment and ground support equipment to the same computer will become non-trivial. As proposed in section 3.3 some form of central control is needed if the satellite is to be tested automatically. A method investigated in this work is to use web sockets. Web sockets are a fairly new protocol based on TCP sockets. The idea is to facilitate instantaneous communication without the handshaking procedure needing to be executed except for when first establishing the connection [19]. The configuration used in this thesis was an echo-server approach. This server has one single purpose, to act as a distributor of messages between clients. If a client sends a message, the server will forward this message to all other clients except the originator. An advantage of this approach is that communication can be achieved between clients, even when connected to networks with strong restrictions such as university networks. Other approaches such as UDP multicasts used by the RAMSES software described in section 2.4 would cause problems in such an environment. The web socket approach requires only one server outside the restricted network to facilitate immediate and bidirectional communication between all clients.

8.2 Experiment simulator interface

The experiment simulators were connected to the server described in section 8.1 using a program that operates in the following way. The program scans all available UART connections on a device, these connections are next polled for a device identity. If an identity is received the device is added to a list of client devices. The program next establishes a single web socket connection to the echo server. This is illustrated in Figure 8.1. Any client on the network can now control the experiment simulators with the commands described in section 4.3. Commands are sent as a JSON object named *es* that contains the CLI command and the device identifier number. An example of setting the load to 12 for unit 4 would look like this:

At the start of the test, the *stage* method is called, it clears the message buffer and sets the name of the test section.

```
if(step.ready()):
    self.stage("Demonstration of current measurement")
```

In the next two steps, the test issues commands for setting the load to 12 and requests a current measurement. The measurement is returned from the experiment simulator and placed in a message buffer.

```
if(step.ready()):
    self.sendCommand({'es':{'cmd':"L12",'id':self.id}})
if(step.ready()):
    self.sendCommand({'es':{'cmd':"C",'id':self.id}})
```

The final step is the verification step. After having issued requests to the test equipment the message buffer of the testing framework will now contain the responses. The test can register verifiers such as *ValueTest* and *RangeTest* using the *check* method. The verifiers are used by the framework to test the messages in the message buffer.

```
if(step.ready()):
    self.check(ValueTest(["es","id"],self.id))
    self.check(RangeTest(["es","current"],60,65))
    self.runVerification()
```

The verifiers require the responses from the experiments to be sent as JSON objects. The verifier *RangeTest*(["es","current"],60,65) will verify that the element value of the key *es.current* is between 60 and 65.

8.4 Evaluation using experiment simulators

The test manager described in section 3.3 was used to create checkout tests for the experiment simulators described in chapter 4. The load of an experiment simulator was connected to a known voltage in series with a known resistance. A test class was created that issues commands to change to load and evaluated the measured voltage and current returned by the experiment simulator (The series resistor ensures a change in voltage for different loads). The class was written in such a way as to be reusable for the rest of the experiment simulators by passing the id in the constructor. This test worked well and was able to detect a wrongly placed resistor in one of the experiment simulators. The approach of using web sockets and the test manager to

create automated tests show promise and is recommended to be used in the MIST project.

9 Dynamic load of digital systems

In this chapter measurements of the current consumption of devices similar to those flown on CubeSats are presented. The purpose of these measurements is to provide a visualization of the need for dynamic load simulation. The measurements were taken with an oscilloscope and a current to voltage converter constructed from a modified experiment simulator.

9.1 ARM based micro computer

Figure 9.1 shows the current consumption of a raspberry pi zero. The measurement was taken during the power-up sequence of the raspberry pi. At time 0.7 s power is applied, the large spike in is a result of the bypass capacitors charging up. Next, the power consumption gradually increases as the boot sequence progresses.

9.2 Dynamic load of AX100 UHF radio

The dynamic current consumption of a UHF radio from the SEAM satellite ground station is shown in Figure 9.2. This radio is identical with the radio that is in the SEAM satellite. The load shown in Figure 9.2 is during microprocessor-only operation without the transmitter enabled. The radio current draw periodically spikes, this is most likely a periodic task waking up, resulting in increased power consumption. The large peak at 12 ms is the microcontroller executing a task list command.

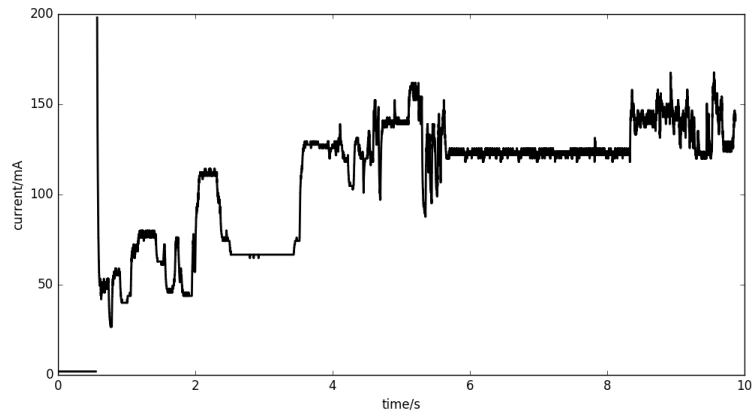


Figure 9.1: Dynamic power draw during bootup of a raspberry pi zero.

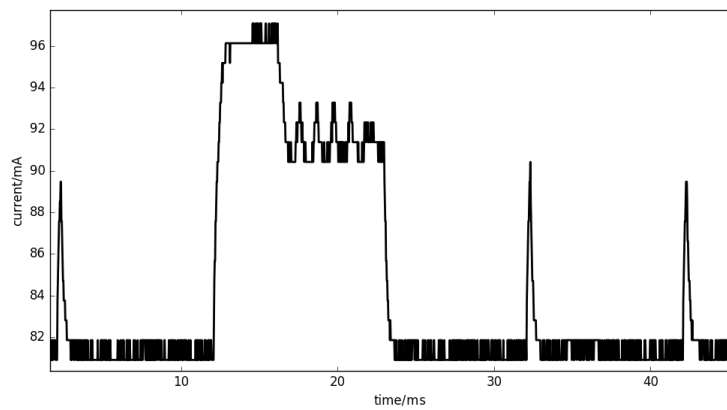


Figure 9.2: Dynamic power draw of a AX100 UHF radio during a task utilization measurement by the radio.

10 Future work

The test equipment developed in this thesis adds to a growing suite of testing tools for the MIST satellite. However, some work still remains until full coverage of the satellite can be achieved. In this chapter, future work is discussed, the discussion includes comments and suggestions from the author on how this work could be done.

10.1 ADC testing

The ADC system will have to be tested. By using the sun sensor simulator and measuring the magnetorquer output the ADC code could be tested on the actual satellite. Some form of an inertial model of the satellite will have to be created which can generate sun sensor data and update the inertial model based on the physical magnetorquer output. This model would have to be able to do ray tracing in order to output the sun sensor data. It could be combined with the solar panel simulator to create realistic power data based on the orientation of the inertial model of the satellite.

10.2 Experiment simulators dynamic loads

The experiment simulators will have to be programmed as to have a load profile that approximates the real experiment subsystems. For this information about the load of the experiments is needed. However, even randomized loading should be effective for testing the output converters of the EPS.

10.3 Creating test cases

The proposed testing system in this thesis has mostly dealt with the layout and structure of the testing system, however, actual tests need to be defined and executed on the satellite hardware. The example in section 8.4 of controlling the experiment simulator should be a good starting point. At the time of writing, much is yet to be decided regarding the OBC software structure and telecommanding philosophy. Creating tests might be a good way to see what the expectations of such a system are and how to approach the task of creating it. Letting the tests drive the design is a common practice in software engineering and might be a good approach for the MIST project as well, even in the physical domain.

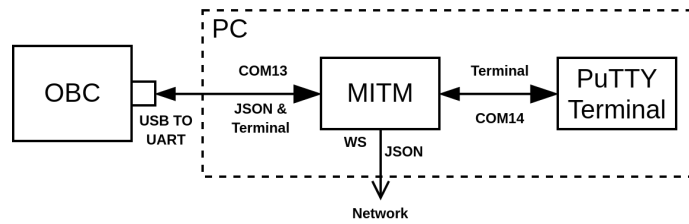


Figure 10.1: Interception scheme of computer readable debug messages from the OBC UART console.

10.4 Computer readable OBC debug messages

The OBC has a UART connection for ground support debug messages. These messages are printed out in a UART terminal client such as PuTTY. Commands can be sent to the OBC through this interface as written commands which are interpreted by the OBC byte-by-byte. The following extension to this debug terminal is suggested. A man-in-the-middle (MITM) program is placed such that it intercepts all the incoming data sent by the OBC. An illustration of this is shown in Figure 10.1. It would examine this data to see if it contains computer-readable code such as JSON. If so these messages would be extracted and forwarded to the testing network using the network structure described in section 8.1. Text that does not contain computer readable messages will be forwarded to the UART terminal. The user would not notice any change from normal operation. Debug messages could easily be generated in OBC with the following C-code

```

sprintf(buf, "{numStoredTelecommands:%d}", numStored);
printf(buf); //This is the computer readable message
printf("This is a normal debug message\r\n")

```

This approach has the advantage that debug messages can easily be generated and used by the rest of the testing system without having to define custom debug telemetry. Another advantage is that it allows testing of the satellite during periods when the radio is silent (Which will be the case for much of the satellite's operation). Additionally, this approach leaves the I²C bus unaffected since no debug telemetry would need to be passed to the radio.

10.5 Test equipment assembly

The test equipment for the MIST satellite needs to be connected to the satellite in a controlled and safe manner. A large aluminum plate has been prepared for mounting the experiment simulators, the solar panel simulators and the

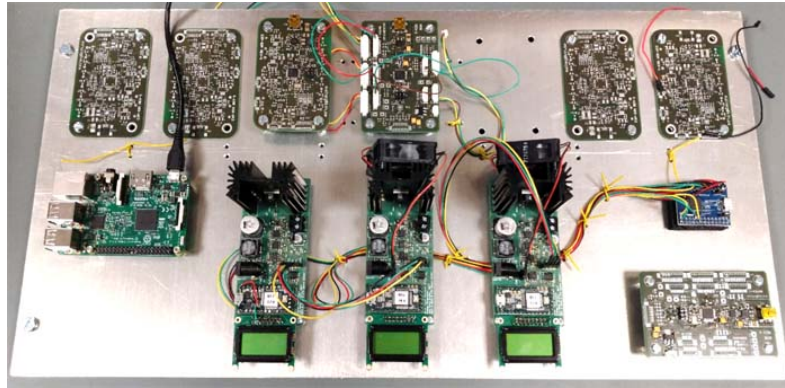


Figure 10.2: The test equipment mounted to an aluminum plate. On the top are the experiment simulators and the breakout board. The three PCBs in the middle are the experiment simulators. The PCB in the lower right is the sun sensor simulator.

sun sensor simulators, as well as other support equipment. The beginnings of this assembly is shown in Figure 10.2.

Bibliography

- [1] California Polytechnic State University. Cubesat design specification rev.13, 2014. <https://www.herox.com/cubesat-challenge/resource/129> (accessed 2018-07-17).
- [2] M. A. Swartwout. A brief history of rideshares (and attack of the cubesats). pages 1–15, March 2011. 2011 Aerospace Conference. DOI:10.1109/AERO.2011.5747233.
- [3] MIST. Basic facts. <https://mistsatellite.space/basic-facts> (accessed 2018-06-28).
- [4] European Commission. Small Explorer for Advanced Missions, 2017. https://cordis.europa.eu/project/rcn/188846_en.html/ (accessed 2018-05-13).
- [5] Spaceflight 101. SEAM Satellite. <http://spaceflight101.com/meteor-m-2-1/seam> (accessed 2018-06-28).
- [6] Innovative Solutions In Space. Isis-obc datasheet, 2014. (Datasheet).
- [7] CubeSat Kit. I/O, Analog, Control and More . <http://www.cubesatkit.com/content/electronics/bus.html> (accessed 2018-06-28).
- [8] GOMspace. Nanopower p-series datasheet p31u / p31us v9.0, 2014.
- [9] European Cooperation for Space Standardization. Ecss, 2018. <http://ecss.nl/> (accessed 2018-06-28).
- [10] Milan Battelino and Christian Svärd. Ramses -a modern and flexible checkout and operational ground system for small satellite projects. 2012. SpaceOps 2012 Conference DOI: 10.2514/6.2012-1293189.
- [11] Solenix GmbH. Elveti - features tour, 2017. <https://www.cubesatshop.com/wp-content/uploads/2017/05/Elveti-Features-Tour-01.01.pdf> (accessed 2018-06-28).
- [12] Jasper Bouwmeester Martin Langer. Reliability of CubeSats – Statistical Data, Developers’ Beliefs and the Way Forward . 2016. Annual AIAA/USU Conference on Small Satellites 2016.
- [13] Diodes Incorporated. N-channel enhancement mode mosfet, 2013. (Datasheet).

- [14] Atmel. Differences between ATmega328/P and ATmega328PB, 2016. https://www.pololu.com/file/0J1464/Atmel-42559-Differences-between-ATmega328P-and-ATmega328PB_ApplicationNote_AT15007.pdf (accessed 2018-06-28).
- [15] Texas Instruments. TCA4311A Hot Swappable 2-Wire Bus Buffers, 2014. (Datasheet).
- [16] Texas Instruments. DACx0508 Octal, 16-, 14-, 12-Bit, SPI, Voltage Output DAC with Internal Reference, 2018. (Datasheet).
- [17] H. Horvath. Atmospheric light absorption—a review. *Atmospheric Environment. Part A. General Topics*, 27(3):293 – 317, 1993. First Ibero-American Conference on the Atmospheric Environment, CIA-MAA91/ACAE91.
- [18] Silonex. SLCD-61N8 solderable planar photodiode. (Datasheet).
- [19] Internet Engineering Task Force. The WebSocket Protocol, 2011. <https://tools.ietf.org/html/rfc6455> (accessed 2018-06-28).

TRITA TRITA-EECS-EX-2018:457