5-21-2019

Applying Internet of Things principles to spacecraft

Brian K. Dixon
Rowan University

Recommended Citation

Dixon, Brian K., "Applying Internet of Things principles to spacecraft" (2019). *Theses and Dissertations*. 2667.
https://rdw.rowan.edu/etd/2667

APPLYING INTERNET OF THINGS PRINCIPLES TO SPACECRAFT

by

Brian K. Dixon

A Thesis

Submitted to the
Department Electrical and Computer Engineering
College of Engineering
In partial fulfillment of the requirement
For the degree of
Master of Science in Electrical and Computer Engineering
at
Rowan University
May 8, 2019

Thesis Chair: John L. Schmalzel

Acknowledgements

I would first like to thank my thesis advisor, Dr. Schmalzel, of the Electrical and Computer Engineering Department at Rowan University. His door was always open whenever I needed help or had a question about my research or writing. He consistently allowed this paper to be my own work but steered me in the right the direction whenever he thought I needed it.

I would also like to acknowledge Dr Krchnavek and Dr. Shin of the Electrical and Computer Engineering Department at Rowan University for their assistance on the MemSat Project which was the catalyst for this work.

I would also like to acknowledge my colleagues Russel Trafford and Adam Fifth for their hard work and dedication to the MemSat Project which was the catalyst for this work.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Abstract

Brian Dixon
APPLYING INTERNET OF THINGS PRINCIPLES TO SPACECRAFT
2018-2019
John Schmalzel, Ph.D.
Master of Science in Electrical and Computer Engineering

This thesis proposes the adaptation of the IEEE 1451 Transducer Electronic Data Sheet (TEDS) for usage in spacecraft subsystems. In the 1990s, TEDS initially were developed as a standardized method to provide metadata required for the operation of a transducer by a microcontroller or data acquisition device. The metadata provides information that identifies and documents key characteristics of the transducer, thereby facilitating plug-and-play interoperability within a system and across networks. An overarching goal of this thesis is to make a case for adapting and extending the TEDS concept as a means for self-describing critical physical components of a CubeSat nanosatellite. This work explores the potential to adapt electronic data sheets to support a more complex system of systems not defined by existing TEDS framework and templates. CubeSat application is assessed and demonstrated utilizing the eXtensible Electronic Data Sheet (XEDS) provision that is described in the IEEE 1451 standard.

Table of Contents

Table of Contents (Continued)

Table of Contents (Continued)

List of Figures

Figures Page

List of Tables

Tables Page

Chapter 1

Introduction

Introduction to Nanosatellites

Nanosatellites are satellites that are between 1 and 10kg in mass. Satellites in this mass range can be traced back to the OSCAR 1 satellite, which in 1961 was a 5kg satellite carrying an amateur radio that was launched as a secondary payload on a US Air Force rocket [1]. Since the 1990s, the quantity of nanosatellites being developed and flown has grown substantially in the United States and internationally [2]. Currently, a large majority of the nanosatellites flown are of the CubeSat type [3], [4].

Compared to their larger counterparts, nanosatellites tend to require lower overall design complexity, which helps lower costs. The combination of low cost and modest complexity is a factor that contributes to nanosatellites' ability to successfully occupy a niche on the smaller end of the satellite spectrum. This aligns with a "faster-better-cheaper" strategy that gained traction during the late 20th century among the National Aeronautics and Space Administration (NASA), the U.S. Department of Defense, and others [5].

Challenges of Nanosatellites

The increased popularity of nanosatellites is linked to the ability to conduct certain types of satellite missions at relatively low cost on relatively compact timelines. However, for a particular nanosatellite to successfully meet its mission goals, budget, and development timeline, some significant challenges must be addressed. Cost challenges encompass not only managing the price of the satellite's component parts, but also personnel costs, and expenses associated with critical supporting items such as tools, testing equipment, software, and laboratory space.

Development times can be shortened the more that commercial off-the-shelf (COTS) components can be incorporated into the design, rather than devoting time to developing custom components. The creation and adoption of widely used set of standards greatly facilitates interoperability of COTS components, thereby reducing development times. Given the low cost and rapid development cycle, it is not uncommon for a project team or institution to work on a set of successive nanosatellite missions. In such cases, managing costs and development timelines can benefit from the opportunity to adapt and reuse components from prior missions.

Objectives of the Thesis

In broad terms, an overarching objective of this thesis is to contribute toward enhancing the ability of CubeSat developers to reliably meet their budgets and development timelines. More specifically, a main goal of this thesis is to propose a standard method that allows the subsystems of a nanosatellite, both structural and electronic, to self-describe themselves. While the focus on CubeSats comes from the need to prove these concepts on a smaller scale, these concepts and methods can apply to any complex system-of-systems. Approaches are identified that extend concepts and methods encompassed in the IEEE 1451 standards that have proven pivotal in facilitating the interoperability and plug-and-play capabilities of smart transducers that are in modern electronics and Internet of Things applications.

This goal is to be accomplished via the use of user-defined Extensible TEDS (XEDS) derived from the IEEE 1451.4 standard. An XEDS data structure contains the information to make a subsystem machine-describable. By making the constituent parts of a nanosatellite machine-describable, the difficulty of development of the nanosatellites can be eased. It is the main goal of this thesis to propose and describe a method of using the XEDS to lower the difficulty inherent in creating a nanosatellite. It is plausible that if standards such as those proposed within this thesis were refined and adopted, they could contribute to reducing the amount of time required to design, assemble, test, debug, and retest CubeSats.

Presented in Chapter 2 is background information that summarizes important concepts and prior work upon which the novel elements developed in this thesis are built. Chapter 3 outlines the concept and usage case for a ‘spacecraft bus,’ with Chapter 4 mating that concept with the proposed XEDS implementation discussed above. Chapter 5 characterizes the electrical communications and storage needed to implement the XEDS in a nanosatellite. It also contains a description of the test setup used to validate the viability of the chosen methods. Chapter 6 is composed of the discussion and conclusions drawn from this thesis and the associated research that led to its formation. The final chapter discusses future work and the need to fully validate the concepts posited in this thesis and the next steps to be taken.

Chapter 2

Background

Applications of Nanosatellites

The growth in popularity of nanosatellites is being propelled by the continued miniaturization of electronics with their small size and relatively low costs allowing more entities access to spaceflight capabilities. These attributes also make nanosatellites well suited for use in educational missions [6].

Though there exists a large variety of nanosatellite missions with a variety of objectives, the ability to use many commercial off-the-shelf (COTS) components when building an individual nanosatellite both decreases costs and shortens development times for all types of missions.

The small size and mass of nanosatellites makes it possible to launch several on the same launch vehicle. For example, in 2017 Planet Labs deployed a “flock” of eighty-eight imaging nanosatellites into orbit aboard a single rocket launched from India [7]. Some other benefits of nanosatellites are the ability to use more, smaller, cheaper satellites to provide greater capabilities than can be created with fewer, larger satellites. This can be achieved by creating networks of interconnected satellites acting as a distributed system. Terms such as constellation missions and formation flying are used to describe different types of multi-satellite missions that adopt this general approach [8], [9]. Though the concept of multi-satellite systems dates back to the 1970s, when topics such as the design of orbital patterns for twenty-five satellites for communications and global positioning system (GPS) navigation were being developed [10], nanosatellite technology today has made larger multi-satellite systems of smaller satellites more feasible and cost-effective, though managing a constellation of nanosatellites introduces its own operational challenges [9]. Global imaging systems [7] and satellite-based internet and mobile phone communications services [11], [12], [13] are two examples of commercial applications of nanosatellite constellations. Earth science missions are the most common type of science-driven multi-satellite missions [8], [14].

CubeSats and MemSat as Examples of Nanosatellites

Much of the increased popularity of nanosatellites can be attributed to the subset of nanosatellites known as CubeSats. The CubeSat specification was developed by The California Polytechnic State University (hereafter referred to as CalPoly) and Stanford University in 1999 [15], [3] and since then 875 CubeSats have been launched [4].

The CubeSat specification covers nanosatellites from 1.33kg to 8kg. There are four different sizes from smallest to largest being designated as 1U, 2U, 3U, and 6U. Due to their different sizes, all the specifications that apply to CubeSats of different sizes are not identical; however, the electrical and communications requirements are uniform across the different sizes [15]. The Rowan University MemSat project, the development of which was the catalyst for much of this work, is example of a 1U CubeSat [16].

The attributes of low cost, relatively simple design, and the availability of plug-and-play components make CubeSats well suited for use as educational missions. In recent years, NASA’s Educational Launch of Nanosatellite initiative (ELaNa) has partnered with universities, non-profits, and high schools to provide CubeSat launch opportunities to promote hands-on science, technology, engineering and mathematics (STEM) education on topics aligned with NASA’s goals [17], [18]. However, nanosatellites also have value for applications other than those linked to educational institutions, as they are increasing used for commercial, military, and science missions [6], [19], [20].

The MemSat mission, which was Rowan University's first CubeSat mission, was launched as part of the ELaNa program. MemSat aimed to compare, in a low orbit space environment, the performance characteristics of memristors and standard, silicon-based memory. The mission's overarching goal was to test the supposition that memristors are potentially more durable in spacecraft implementations [16].

Nanosatellites as Systems of Systems

The term systems of systems (SoS) generally refers to a collection of multiple networked, interconnected systems, with the specifics of any SoS generally being application dependent. The systems that make up a SoS are networked together so that they can cooperate in a synchronous manner to achieve functionality that is greater than each individual system could provide on its own [21].

The system-of-systems concept applies to a plethora of applications and has achieved recognition within the systems engineering discipline, including technical committees Institute of Electrical and Electronics Engineers (IEEE's) Reliability Society [22], and the IEEE's Systems, Man, and Cybernetics Society [23].

Virtually all modern satellites can be considered systems of systems, because they are generally composed of a group of interconnected electrical subsystems, networked together. For satellites, the SoS encompasses not only system components that are physically located on the launched satellite, but also includes ground-based systems, such as the tracking station, and radios. Onboard the satellite itself, the spacecraft bus consists of this system of systems, along with the structural elements of the satellite, but excludes the satellite's payload. The standardization of CubeSat bus designs is intended to facilitate and simplify development while allowing, within constraints, flexibility in the ultimate design of a specific payload [6], [24]. Together, this group of systems works together to perform a satellite's given mission.

Prospects for Shortening Small Satellite Development Cycle

The physical development of CubeSats is governed by the relatively comprehensive CubeSat specification set out by CalPoly [15]. However, existing electrical standards are less comprehensive for CubeSat applications. While standards do exist around battery capacity and inhibiting electrical operation while stowed, there currently are no widely accepted guidelines establishing methods of intra-satellite communication and associated integration, outside of existing serial communication protocols. The creation and adoption of a comprehensive electrical standard would allow development teams to more readily utilize COTS components, which has the potential to substantially lower the length of a CubeSat development cycle.

In theory, the goal of having standards developed and adopted to enhance plug-and-play compatibility for satellites is analogous to the successes realized for plug-and-play compatibility for personal computer components [25]. In practice, for satellites, the path toward comprehensive plug-and-play compatibility of electronic components can at times be slower than desired, as different approaches can be pursued simultaneously to address issues, potentially complicating the establishment of a consensus standard [25].

One of the major barriers to the rapid development and deployment of complex systems of systems CubeSats using COTS components is the lack of guaranteed physical and electrical compatibility between sub-systems. For example, problems arising from a lack of adequate electrical standards can manifest as multiple subsystems requiring the same header pin for different functions, or different satellite components having been built to operate using different data protocols, or even power rail incompatibilities. In addition, there is no guarantee that each subsystem's processor is going to be operating on the same data communication protocol as each other or the satellite's mainboard. For the MemSat mission, design barriers that needed to be overcome due to a lack of complete plug-and-play functionality included communication protocols (pin assignments), interference between components (e.g., antenna SMA connectors hitting frames), and subsystems with incompatible heights that required header modification.

The interoperability challenges present in the development of small satellites are broadly analogous to those present in many other instruments, systems, and systems of systems that contain transducers [26], [27].

IEEE 1451 and Transducer Electronic Data Sheets

In response to the interoperability challenges that can arise in the nearly ubiquitous applications of transducers in modern systems, the IEEE 1451 standard was established to promote plug-and-play compatibility [27], [28]. A benefit of these widely accepted standards is that they apply to a range of applications, devices, and systems. However, the existing IEEE 1451 standards do not encompass all of the specific needs of satellites. Luckily, the IEEE 1451 standards contain provisions for extending the standards in ways that address the needs of specialized applications, such as those associated with satellite buses for small satellites, including MemSat. Minimizing or eliminating incompatibilities across electronic components can be addressed by the development and widespread adoption of standardized electronic data sheets [29], designed with certain types of systems in mind.

In response to the interoperability challenges that can arise in the many applications of transducers in modern systems, the development of the IEEE 1451 family of standards was initiated in the mid-1990s [27], [30] and continues to be developed today, to meet emerging needs [28], [31]. IEEE 1451 aims to facilitate the manufacturing and use of a very wide variety of smart transducers by specifying a set of transducer interface standards that promote interoperability of transducers within a specific device and across various types of networks, including the Internet of Things [31], [32], [33], [34], [35].

A critical element of the IEEE 1451 standards is the definition of Transducer Electronic Data Sheet (TEDS) [36]. In effect, TEDS consist of manufacturer-supplied metadata that provides documentation, in a standardized form, that serves multiple purposes. TEDS enable a transducer to identify itself to the network, so as to reduce the amount of manual configuration needed when adding transducers to a system. Additionally, TEDS associated with IEEE 1451.4 contain a standard of how to interface with and subsequently use the analog signal from the transducer. The standardization of messaging formats and protocols reduces the amount of software needed to allow transducers to communicate across a system.

TEDS reside within the infrastructure at a relatively high position, so that they are largely independent of the details of the physical layout and transmission layers [30]. TEDS typically are stored on nonvolatile memory physically on the transducer, though provisions exist for virtual TEDS that may be accessed over networks [37]. Generally, the TEDS reside on the transducer interface module (TIM) part of a smart transducer. When queried for information contained in the TEDS, the TIM responds by passing the requested metadata. The TIM also can contain the hardware required to convert the transducer-generated signal so that it is compatible with the Network Capable Application Processor (NCAP) portion of a smart transducer and vice versa (i.e., the TIM has analog-to-digital and digital-to-analog converters, as well as signal conditioning apparatus) [33].

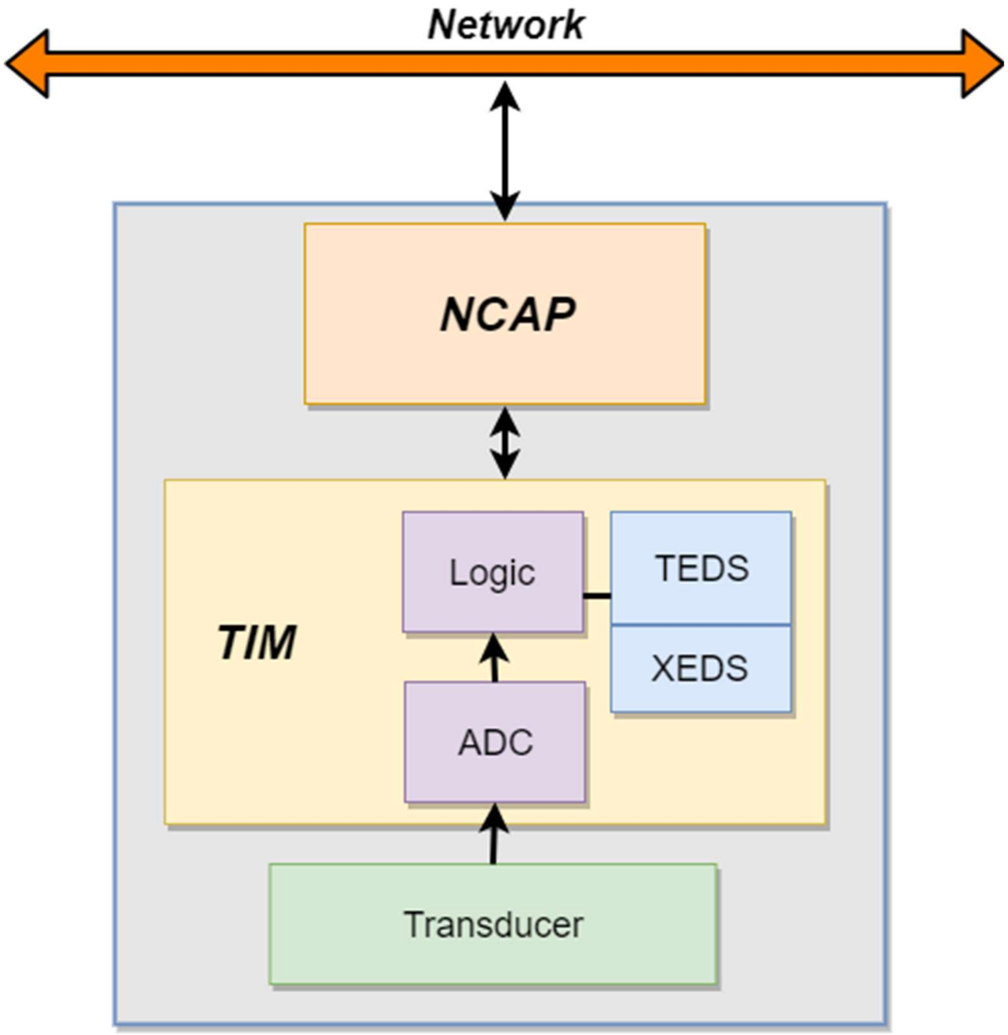

Figure 1. Transducer Interface Module

A benefit of the IEEE 1451 set of standards is that they apply to a range of applications, devices and systems. However, the existing family of IEEE 1451 standards do not encompass all of the specific needs of satellites. The standard TEDS templates defined in IEEE 1451.4 apply to many common types of transducer applications, but not to all [29].

User-defined Extensible TEDS (XEDS)

Because the TEDS templates that are part of the IEEE 1451.4 standard do not apply to all types of transducers, the standards contain options for specialized user-defined TEDS known as eXtensible Electronic Data Sheet (XEDS or sometimes xTEDS) [37]. The structure for a subsystem XEDS can be patterned along the lines of the TEDS described in IEEE1451.4. As is the case for TEDS, the XEDS metadata can be written following the open standards set forth for the Extensible Markup Language (XML) [38].

The first block of any XEDS includes the basic TEDS metadata outlined in the 1451 standard, including the manufacturer's ID number, the subsystem's model number, version letter, version number, and serial number. The remainder of the XEDS is to contain any and all information that encompasses any additional necessary identifying characteristics of the subsystem and information required for interfacing with and operating the subsystem. Recently, at Rowan University, the concept of developing XEDS to address CubeSat communication subsystems needs has been pursued [39].

The Internet of Things (IoT)

The Internet of Things, also known as 'IoT', is a somewhat vague term that is used to describe the interconnectivity of individual networked devices that together provide new functionalities for a wide variety of applications [40], . Enabled in large part by technological advances in smart sensors, local, wireless and mobile networks, and the internet, and by multiple co-evolving communications protocols and standards, the general concept of the IoT is to readily connect devices in various locations in order to perform the functions of a new application [41]. Commonly traced back to the late 1990s, the popular IoT concept has evolved at a rate such that by 2017 it could be characterized as having gone through three generations of development [42]. Thus far, a consistent and crucial aspect of the rapid development of IoT has been the availability of relatively low-cost smart objects that can be readily connected in IoT applications due to their built-in networking abilities and self-identification properties, which are some of the same features the IEEE 1451 has sought to address since its inception, in part by the implementation of standard self-documenting electronic data sheets.

The IoT concept applies to relatively low cost, everyday consumer items as well as to high end, high-tech commercial, engineering, infrastructure management and research applications [39]. In the consumer space, controlling and monitoring remote devices from a mobile phone is a common type of IoT application. Well known IoT applications of this class consist of things such as Nest Labs' learning thermostat, security cameras, doorbells, alarm systems, and smoke alarms [43]. Other consumer-focused IoT applications include wearable items, including medical related devices [44], and assorted home automation technologies beyond those already described (e.g., 'smart' LED lightbulbs, Wi-Fi controllable electrical outlets, etc.). However, there is a growing concern over the lack of a unified security standard for the data transmitted and contained inside of these networks [45].

Chapter 3

Spacecraft Bus Characteristics

Definition of a Spacecraft Bus

A spacecraft bus, which sometimes is referred to as a satellite bus, is a general specification on which a family of satellites is based [46]. A spacecraft bus encompasses a satellite's electrical and physical infrastructure, including control systems, power supply, structure and propulsion. With respect to a CubeSat, this includes the command and data handling (C&DH) system, the electrical power system (EPS), the attitude control system (ACS), the radio communications systems, including its antennas, and the general structural components.

Rapid and cost-effective nanosatellite development is aided by the commercial availability of relatively comprehensive, standardized spacecraft buses for several different sizes and types of nanosatellites [47]. Customization of a COTS spacecraft bus allows designers to meet the needs of a particular mission and its payload. The payload is not considered part of the spacecraft bus. However, in order to allow more resources to be available for payloads, spacecraft bus designs often seek to minimize the power demands and mass of the bus itself [48].

Definition of a 1U CubeSat

A 1U CubeSat, such as the Rowan University MemSat, is an 11x10x10cm 1.33kg nanosatellite with a 10x10x10cm internal volume. This internal volume and 1kg mass make up one unit of CubeSat, for which the 1U designation is an abbreviation. As is true for all CubeSat missions, MemSat must adhere to a set of requirements that provide design specifications for many mechanical, electrical, operational, and testing factors [15].

As is typical for CubeSats, MemSat contains a C&DH system, EPS system, ACS systems (which in the case of MemSat is passive), a radio tracking beacon, and a radio transceiver to communicate with the surface. It is also composed of a structure to hold all of these systems, along with their associated and antennas and photovoltaics.

Select Elements of a Sample Spacecraft Bus for a 1U CubeSat

This section presents a sample spacecraft bus specification as it pertains to a spacecraft bus based on the Pumpkin Aerospace 1U CubeSat Chassis and utilizing Pumpkin Aerospace motherboard (P/N 710-00484, RevD, non-pass-through) [49]. This is the bus that was chosen for the MemSat mission.

Motherboard

The Pumpkin motherboard's main electrical connections are conducted across two, 52 pin IBM style headers. Ground is connected to the chassis and delivered through pins H2.31 for analog ground and H2.29, H2.30, H2.32 for digital ground. VBATT is on H2.45 and H246, with battery voltages usually in the range +7V to +10V. USB power (+5V_UBS) is assigned H1.32. System power (+5V_SYS) is on H2.24 and H2.25, whereas VSS_SYS is on H2.27 and H2.28. Other notable pin designations include H1.01 through H1.24 and H2.01 through H2.24 being dedicated to General IO. Also, User Pins are assigned to H1.47 through H1.52 and H2.47 through H2.52. The full pinout can be seen in Table 1 and Table 2 and is consistent with documentation provided by the manufacturer in CubeSat Kit Flight Motherboard Rev. D [49].

Table 1.

RevD Motherboard Header 1 Pinout

Header 1			
Pin	Function	Pin	Function
1	IO.23	27	Sense
2	IO.22	28	VREF1
3	IO.21	29	-RESET
4	IO.20	30	VREF2
5	IO.19	31	OFF_VCC
6	IO.18	32	+5V_USB
7	IO.17	33	PWR_MHX
8	IO.16	34	-CTS_MHX
9	IO.15	35	-RTS_MHX
10	IO.14	36	-DTR_MHX
11	IO.13	37	-DTR_MHX
12	IO.12	38	TDX_MHX
13	IO.11	39	RDX_MDX
14	IO.10	40	VBACKUP
15	IO.09	41	SCL_SYS
16	IO.08	42	VBACKUP
17	IO.07	43	SCL_SYS
18	IO.06	44	RSVD0
19	IO.05	45	RSVD1
20	IO.04	46	RSVD2
21	IO.03	47	USER0
22	IO.02	48	USER1
23	IO.01	49	USER2
24	IO.00	50	USER3
25	-Fault	51	USER4
26	VREF0	52	USER5

Table 2.

RevD Motherboard Header 2 Pinout

Header 2			
Pin	Function	Pin	Function
1	IO.47	27	VCC_SYS
2	IO.46	28	VCC_SYS
3	IO.45	29	DGND
4	IO.44	30	DGND
5	IO.43	31	AGND
6	IO.42	32	DGND
7	IO.41	33	S0
8	IO.40	34	S0
9	IO.39	35	S1
10	IO.38	36	S1
11	IO.37	37	S2
12	IO.36	38	S2
13	IO.35	39	S3
14	IO.34	40	S3
15	IO.33	41	S4
16	IO.32	42	S4
17	IO.31	43	S5
18	IO.30	44	S5
19	IO.29	45	VBATT
20	IO.28	46	VBATT
21	IO.27	47	USER6
22	IO.26	48	USER7
23	IO.25	49	USER8
24	IO.24	50	USER9
25	+5V_SYS	51	USER10
26	+5V_SYS	52	USER11

Physical Structure

The physical characteristics of the of the spacecraft bus exist to ensure the physical compatibility of the subsystems that the spacecraft bus contains. The major physical parameters that will be enumerated here can be sorted into two categories; namely, ones for internal compatibility between subsystems, and the external dimensions that allow compatibility with CubeSat deployers. The internal parameters are: the board mounting standoffs, the standard internal board footprint, the header locations, and the board-to-board envelope. The external parameters include the gross external dimensions, the rails that interface with the internals of the deployer, the separation springs and deployment switch locations, and the remove-before-flight pin location.

Physical characteristics of a satellite bus may be documented with an XEDS. This section presents key aspects of the physical bus structure used in the MemSat mission. Note that Figures 3 through 11 were developed by the author by editing publicly available CubeSat Kit 3D CAD models [49] that are made available courtesy of Pumpkin, Inc., to be freely used within CAD systems to create 2D and 3D illustrations [50], [51].

The first physical characteristic to be defined is the existence and location of the board holding standoffs. These standoffs are 4-40 threaded and riveted into the bottom of the satellite chassis. There are four standoffs, and their locations can be seen in Figure 2, with the dimensions given being the distance to the center of each standoff from the center of the bottom face of the chassis, in millimeters.

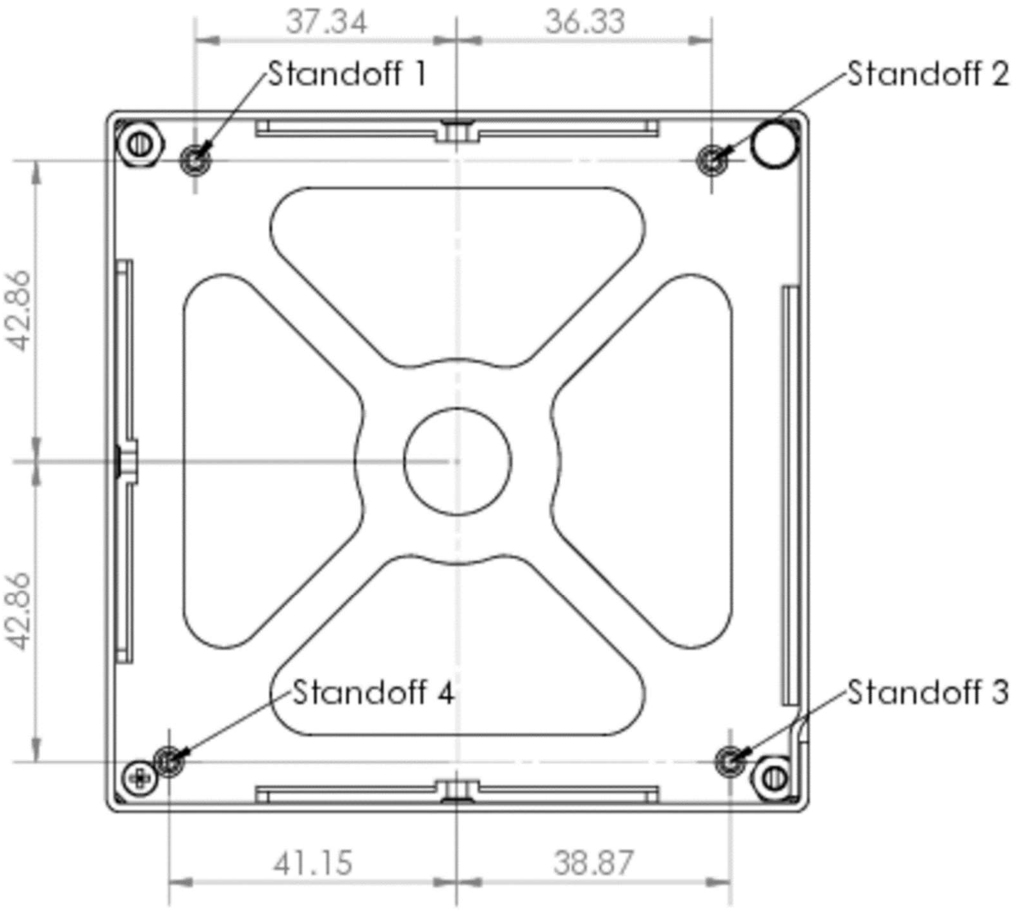

Figure 2. Standoff Mounting Dimensions - Chassis (mm)

The location of standoff 1 is 37.34mm to the left of center, and 42.86mm above the center of the bottom face of the chassis. The location of standoff 2 is 36.33mm to the right of center, and 42.86mm above the center. The location of standoff 3 is 38.87mm to the right of center, and 42.86mm below the center. The location of standoff 4 is 41.15mm to the left of center, and 42.86mm below the center.

The standard footprint for the internal circuit boards can be seen in Figure 3. Though not all of these dimensions are critical, they provide a maximum available area for the design of any internal circuit boards. Boards may be smaller than this, but they must stay within the 90.17mm by 95.76mm form factor and contain mounting provisions for the standoffs and headers also mentioned in this chapter.

The header mounting location can be seen in Figure 4. The headers consist of two, 52-pin IBM pass through style headers. The pinouts for these headers can be found in an earlier section of this chapter. All of the following locations reference the circuit board footprint in the same orientation as it is in Figure 3 The edges of the headers are located 13.97mm from the left side of the circuit board footprint, 12.70mm from the right edge of the board footprint, and 5.33mm from the top of the footprint, with the final edge being 7.62mm below that third edge. Header 1 is the upper of the two headers, with header 2 being the lower, with respect to the circuit board footprint orientation presented in both Figure 4 and Figure 5.

These circuit boards must fit inside of an envelope that consists of the circuit board footprint and starts 5.97mm from the bottom face of the spacecraft chassis and ends 10.25mm below the top of the spacecraft chassis, for a total of 83.28mm of height. The components attached to these boards must fit between the internal surfaces of the chassis, a distance of 98.50mm, as depicted in Figure 6.

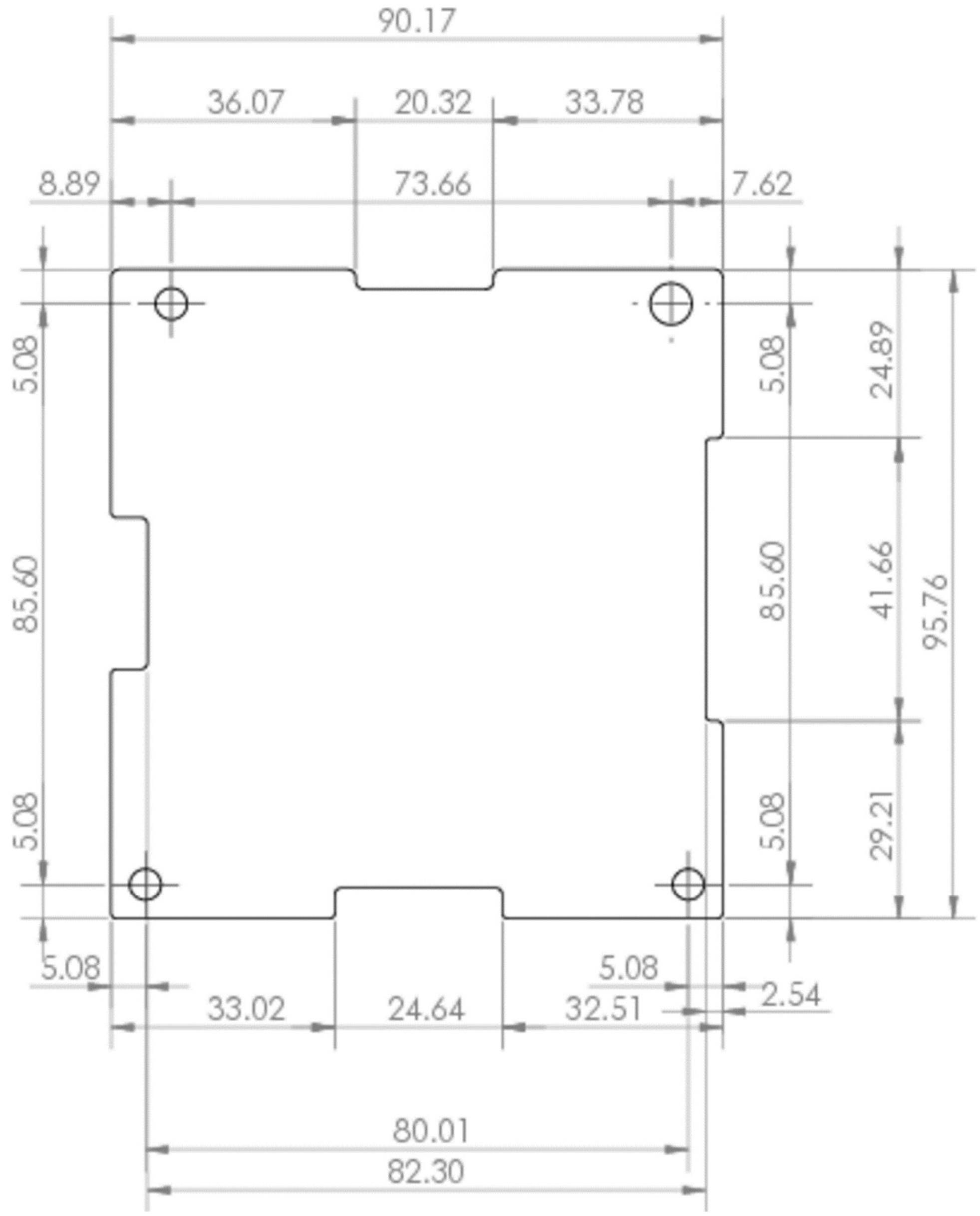

Figure 3. Header mounting dimensions (mm)

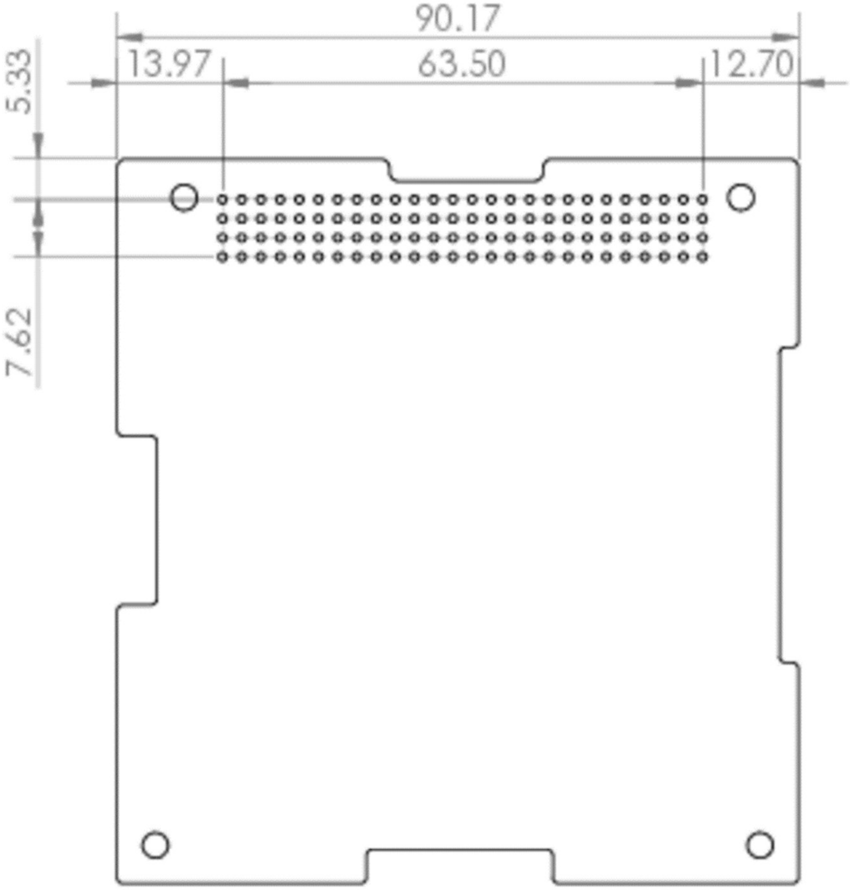

Figure 4. Internal Board Dimensions (mm)

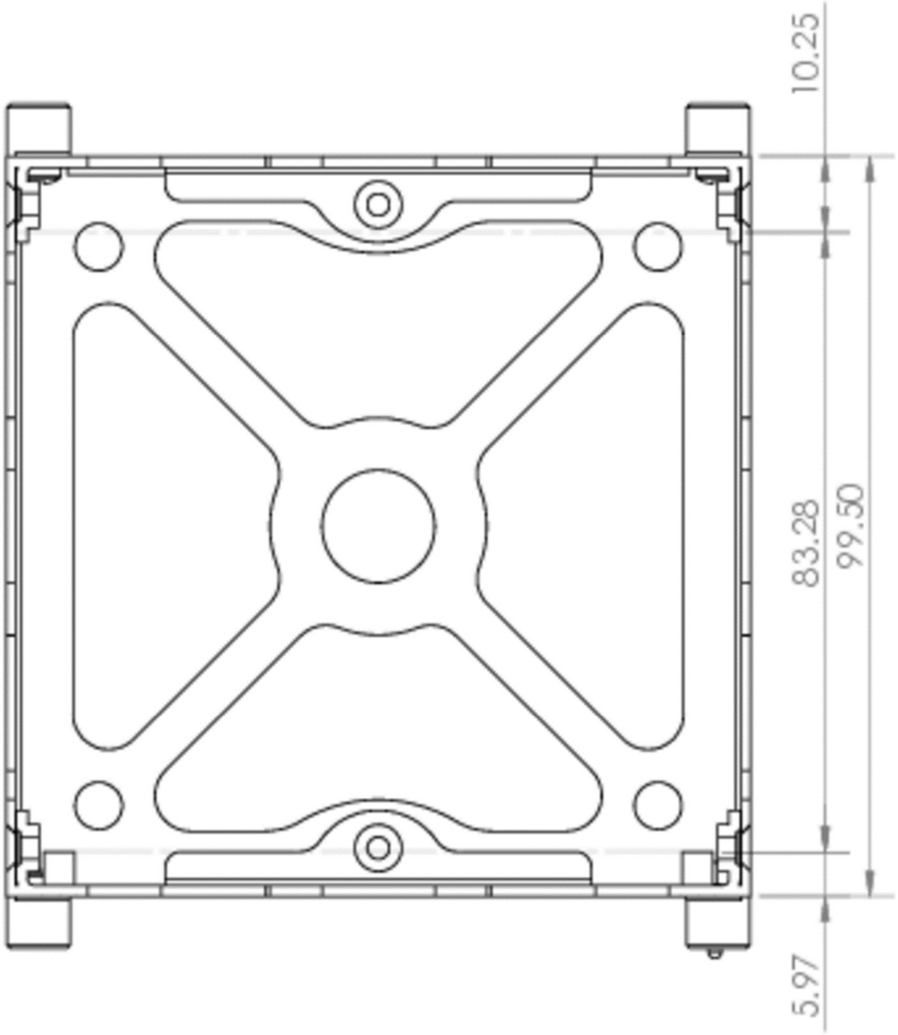

Figure 5. Chassis Z size constraints (mm)

The remainder of the physical parameters are ones that are utilized to ensure compatibility with the satellite deployment mechanism used for the launch and deployment of CubeSats. These include the overall physical dimensions of the chassis, the size and shape and hardness of the rail section of the chassis that interfaces with the deployer unit, in addition to the locations of the deployment springs and switch.

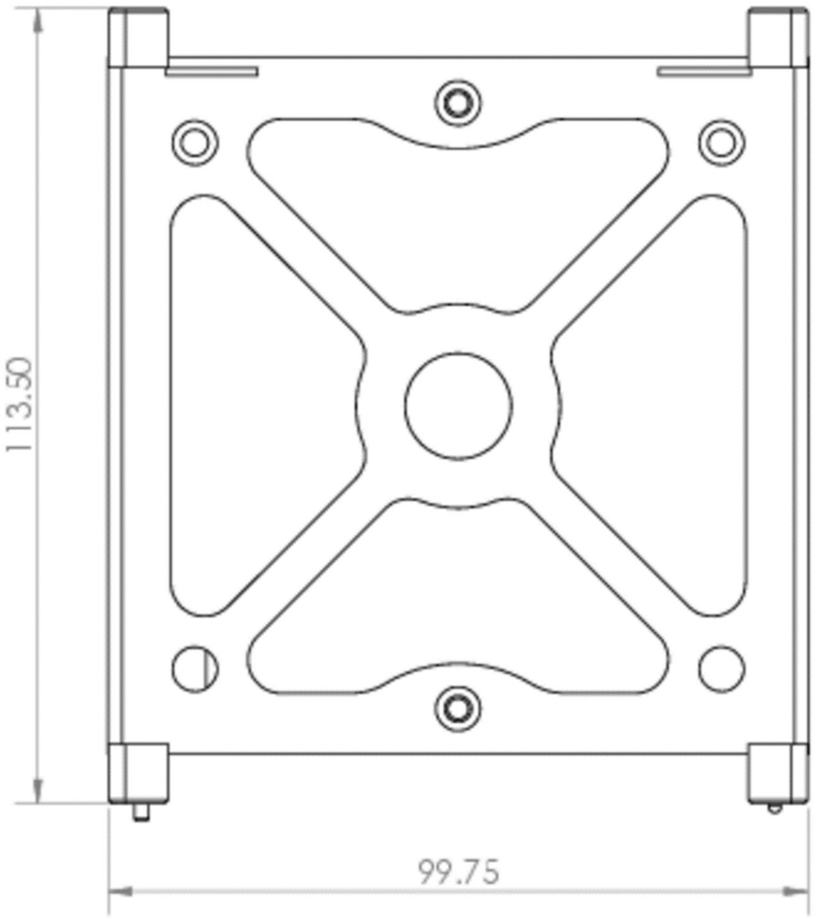

Figure 6. XZ External Dimensions (mm)

The overall external dimensions are presented in Figures 7 and 8, with the Z and Y dimensions being the same at 99.75mm. The overall Z dimension for the extended interface rails is 113.50mm with the main chassis casing being 99.50mm in height.

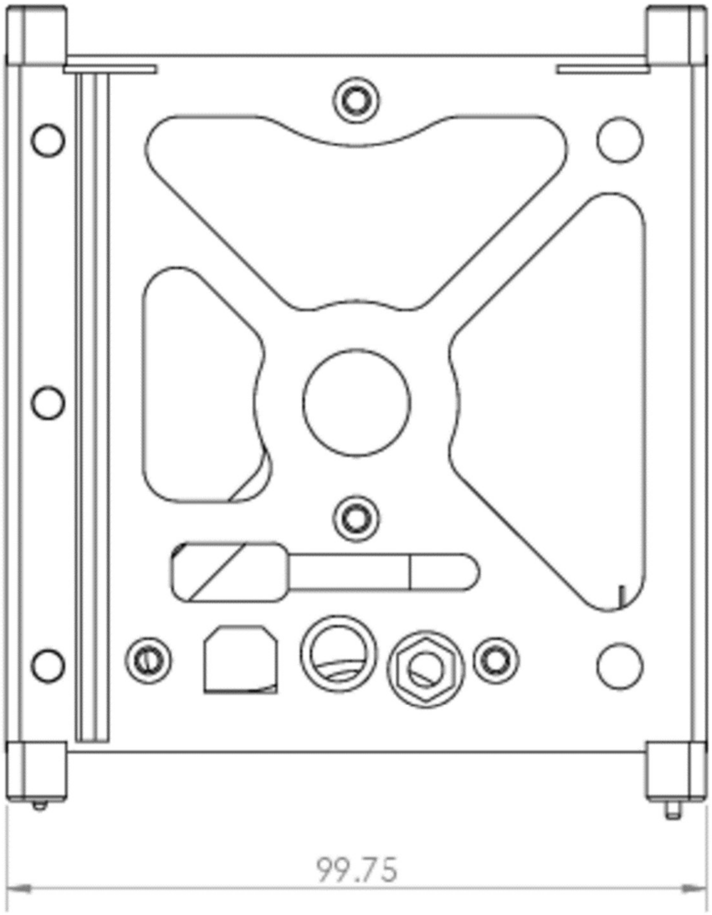

Figure 7. Chassis Y Size Constraints (mm)

The Extended Rails that protrude from the main case can be seen in Figure 9. These four protrusions on each side extend 7.00mm out from the main body of the spacecraft chassis and consist of an 8.50mm wide square with a 2.36mm radius on the outside corner, a trait which continues down the entire length of the spacecraft chassis.

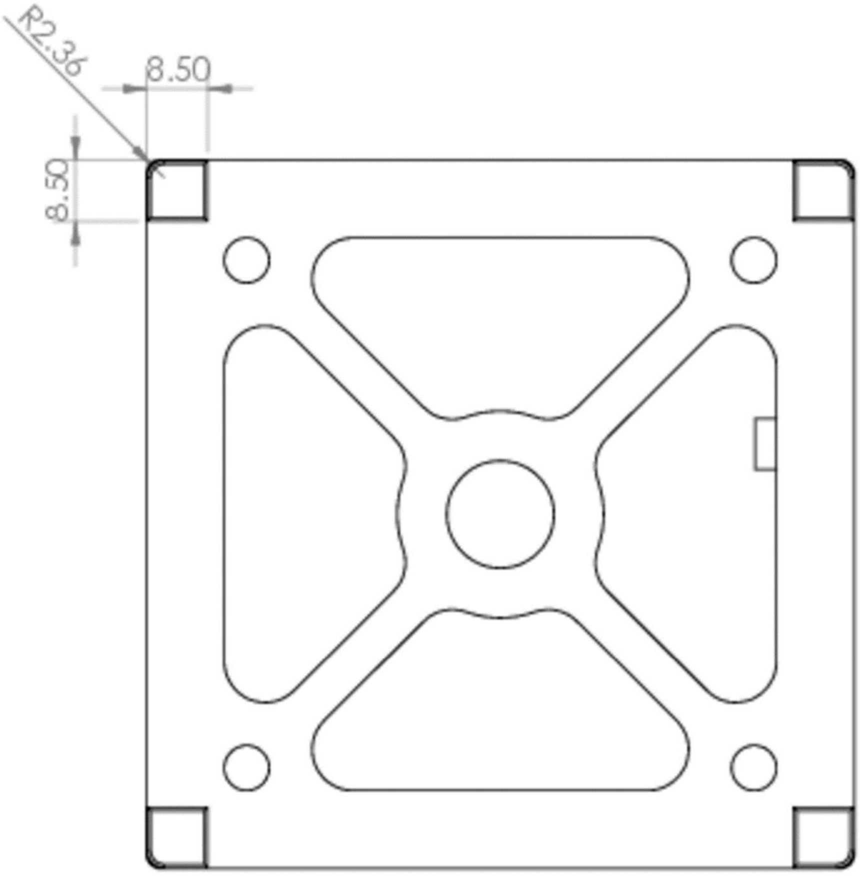

Figure 8. Rail End Dimensions (mm)

The areas of the main body of the spacecraft chassis that exist between these extended protrusions require a different type of surface finishing, specifically a hard anodizing that is at least Rockwell C 65-70 [15]. The purpose for this is twofold, firstly it prevents the spacecraft chassis from being damaged by the deployer during launch. Secondly, anodizing prevents the chassis from cold-welding itself to the internal structure of the deployer during the launching process and during any periods of extended storage. These specially treated areas can be seen in Figure 10 (they are the hatched areas) and consist of the area 8.50mm in from the corners of the spacecraft chassis.

The bottom face of the spacecraft chassis contains the Deployment Springs and the Deployment Switch, the locations of which can be seen in Figure 10. The Deployment Springs are of CalPoly design and consist of an 8-36 threaded body, with 0.14 lbs. of initial force and 0.9 lbs. of final force over a 4mm throw. The switch is a space rated switch attached to a spring plunger that extends outside of the rail foot upon deployment starting up the satellite. As depicted in Figure 11, a remove-before-flight pin is located on a face of the spacecraft chassis. Once inserted, the pin renders the spacecraft electrically inoperable, until such time as it is removed.

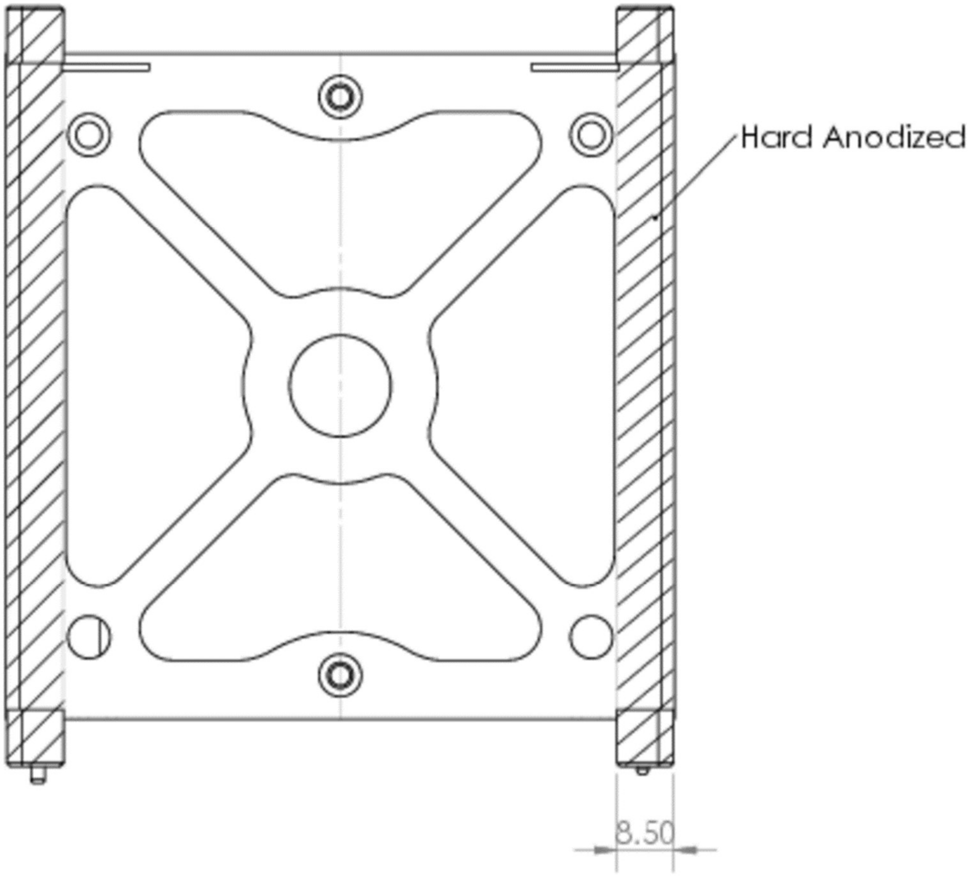

Figure 9. Hard Anodizing Rails (mm)

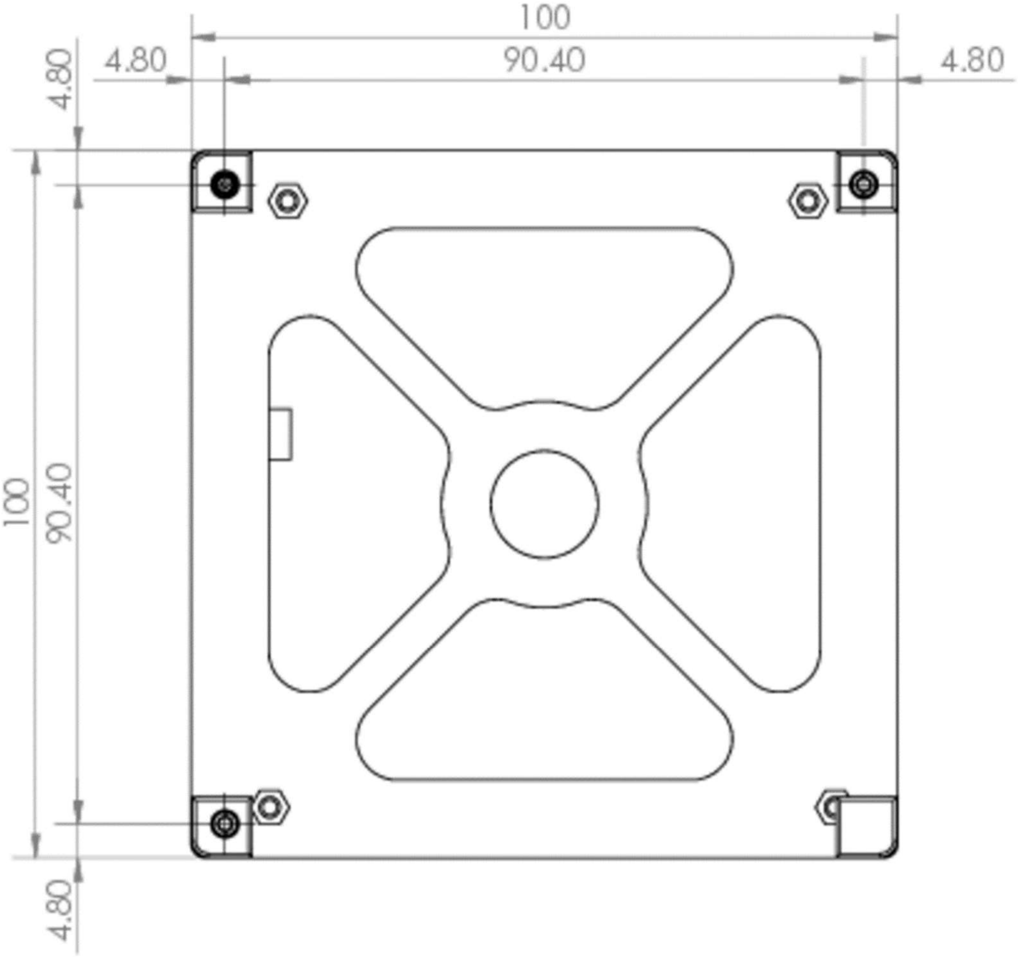

Figure 10. Deployment Apparatus locations (mm)

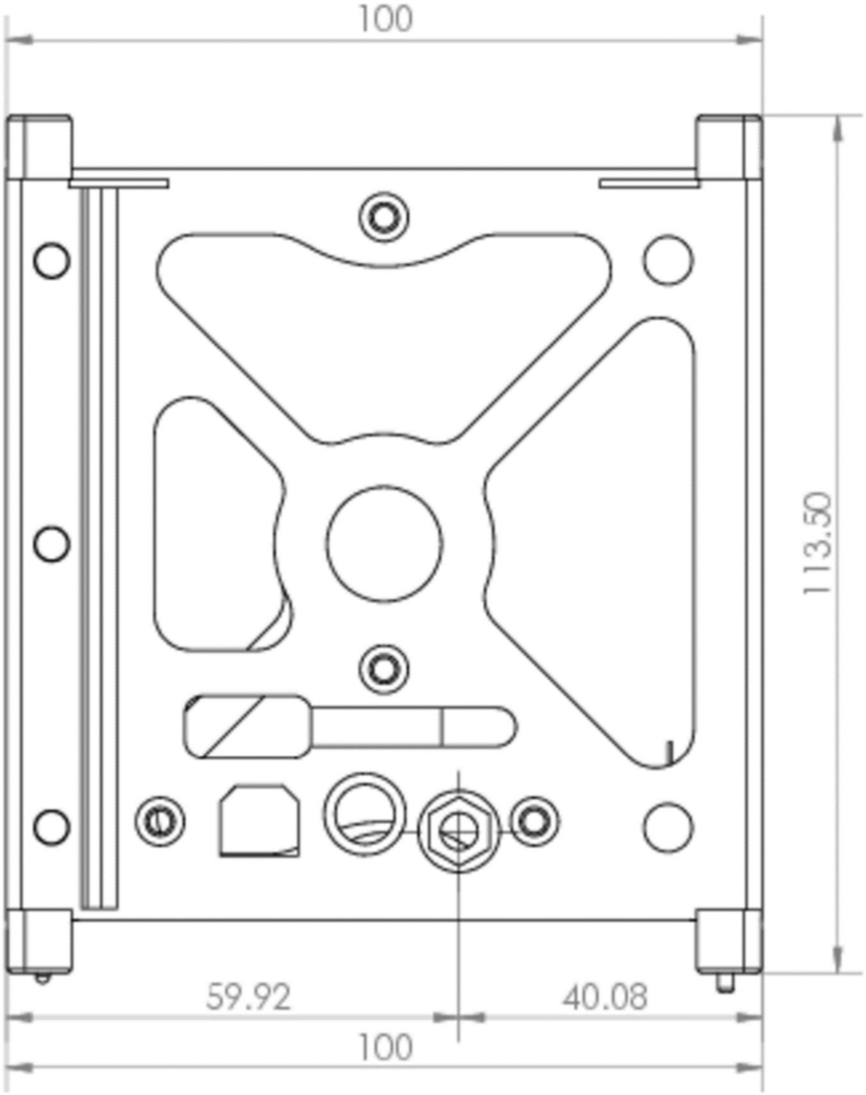

Figure 11. Remove-Before-Flight Pin location (mm)

The chassis that is a part if this particular satellite bus must allow for the internal mounting of boards of the given form factor, while still conforming to the requirements of the general CubeSat standard. Those standards include, but are not limited to, the 100Wh limit on chemical storage, meeting NASA outgassing requirements, the limits on potential debris, the center of mass being within 2cm of the geometric center of the chassis, and the chassis rail size and surface finish requirements. Additionally, to conform to this satellite bus, the CubeSat must contain a number of other qualifying subsystems and functionalities. These include for the communications subsystem the usage of a 70cm radio transceiver for data up and down link, in addition to a 70cm radio beacon for tracking the satellite. There must also be a separate 2m receiver that can receive signals to perform certain administrative functions. These functions must include the ability to shut off the satellite permanently, and to disable any onboard cameras that could be pointed at the Earth when flying over certain geographic regions. The ACS onboard the CubeSat must be passive, and the CubeSat itself can not contain any chemical thrusters. There also must be at least one available position in the internal board stack for the experimental payload.

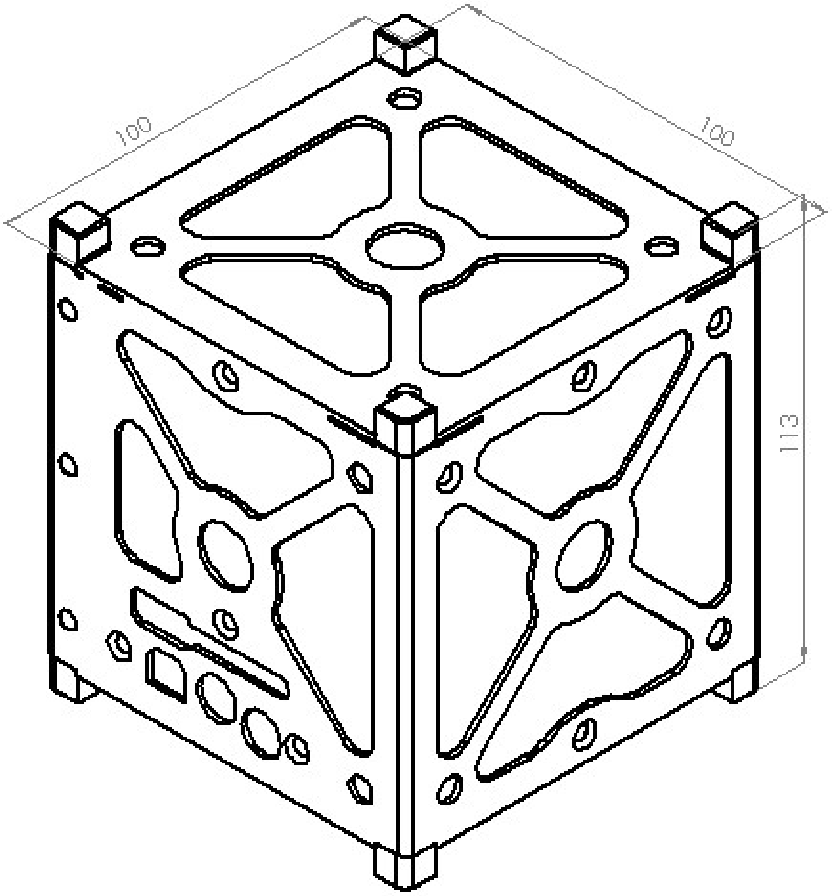

Figure 12. Isometric View

Chapter 4:

Defining XEDS for Spacecraft BUS Components

This chapter presents information on how to translate a spacecraft bus specification into the XEDS format as outlined in the IEEE 1451.4 standard for user generated TEDS [27]. A TEDS can consist of four major sections: the basic TEDS, the transducer templates, the calibration templates, and the user generated TEDS.

Overview of TEDS (IEEE 1451.4 – 2004)

The basic TEDS section contains generic identification metadata that is required across all types of smart transducers that comply with the 1451 standard. This includes the compliant device's manufacturer ID, model number, version letter, version number, and serial number. The manufacturer ID comes from a table that is included in the 1451 standard. Companies can request a manufacturer ID number from the IEEE, which is then added to the standard.

In order to adopt and adapt the TEDS concept that traditionally has been applied to transducers, so that it is applicable for developing XEDS for a spacecraft bus, it is useful to review and appreciate established methods by which TEDS are used to encode information about transducers. For example, the IEEE 1451.4 standard offers several templates, each of which applies to a different common type of transducer. The transducer template segment of a TEDS contains the information needed to make a particular transducer machine describable, in effect, functioning as a machine-readable data sheet. Because functionality of different transducer types varies, the specific information that the different templates stipulate as being required varies. The list of

TEDS templates that apply to transducers is presented in Table 3. This table includes the transducer type, and its template number as described in the standard.

Table 3.

TEDS Transducer Template List

Transducer Type	Template Number
Accelerometer & Force	25
Charge Amplifier (w/ attached accelerometer)	26
Microphone with built-in preamplifier	27
Microphone Preamplifiers (w/ attached microphone)	28
Microphones (capacitive)	29
High-Level Voltage Output Sensors	30
Current Loop Output Sensors	31
Resistance Sensors	32
Bridge Sensors	33
AC Linear/Rotary Variable Differential Transformer (LVDT/RVDT) Sensors	34
Strain Gage	35
Thermocouple	36
Resistance Temperature Detectors (RTDs)	37
Thermistor	38
Potentiometric Voltage Divider	39
Charge Amplifier (w/ attached force transducer)	43

The calibration segment of a TEDS is an optional segment that can be added to any transducer TEDS. The purpose of these templates is to add information so that sensitivity variations inherent in different physical devices can be accounted for and normalized for ease of usage. There exist three main types of calibration templates, each designed for a different type of calibration approach. The three calibration template types are listed, along with their template ID number, in Table 4. Calibration table templates are appropriate when using real-world measurements / interpolation for calibration purposes. When an equation for a curve is suitable, a calibration curve (polynomial) template is used. And a frequency response table is for devices such as accelerometers.

Table 4.

TEDS Calibration Templates List

Calibration Type	Template Number
Calibration Table	40
Calibration Curve (Polynomial)	41
Frequency Response Table	42

The last part of a TEDS, the user TEDS, can be considered a separate type of TEDS that is open for users to customize. In some applications, user TEDS can supersede the transducer template as the main vector of information in a given TEDS. A user TEDS contains whatever information the user of that TEDS deems necessary. It is from this provision that the XEDS that are discussed in the following chapters take their inspiration and form.

Sample TEDS of thermocouple

A thermocouple is a widely used electronic temperature measurement device that functions by measuring the voltage that develops between wires made of different metals that are connected at two junctions. One junction is at the temperature to be measured and the other cold junction serves as a reference. Calibration allows conversion of the voltage generated due to the temperature difference between the junctions into the temperature at the measurement junction. Many types of thermocouples are made for applications that span different temperature ranges and environmental conditions [52] [53].

The IEEE 1451.4 TEDS template for a generic thermocouple contains information designed to uniquely identify a particular thermocouple and several of its performance characteristics. Table 5 depicts the IEEE 1451.4 template for the TEDS of a generic thermocouple, which is identified as Template 36 in the standards [27]. Each line item in the table represents a data field in the TEDS, and specifies its bit length, data type and the appropriate data range. The TEDS itself is broken down into two major sections, the basic TEDS, and the class specific TEDS.

The basic TEDS data contains information generic to all smart transducers using the TEDS standard; namely, manufacturer, model number, version letter and version number.

Table 5.

TEDS Template for a Thermocouple

Section	Description	Bits	Data Type (and Range)
Basic TEDS	Manufacturer ID	14	INT (17-16381)
	Model Number	15	INT (0-32767)
	Version Letter	5	INT (0-63)
	Version Number	24	INT (0-16777215)
Thermocouple	Template ID	8	Integer (value = 50)
	Transducer electrical signal type	-	Assign = 0, “Voltage Sensor”
	Minimum temperature	11	ConRes (-273 to 1,770, step 1)
	Maximum temperature	11	ConRes (-273 to 1,770, step 1)
	Minimum electrical output	7	ConRes (-25E-3 to 0.1 step 1E-3)
	Maximum electrical output	7	ConRes (-25E-3 to 0.1 step 1E-3)
	Mapping method	-	Assign = 3, “Thermocouple”
	Thermocouple type	4	B, E, J, K, N, R, S, T, or non-std.
	Cold junction compensation required	1	CJC Required or Compensated
	Thermocouple resistance	12	ConRelRes (1 to 319k, ±0.155%
	Sensor response time	6	ConRelRes (1E-6 to 7.9, ±15%)
	Calibration date	16	DATE
	Calibration initials	15	CHR5
	Calibration period	12	UNINT
	Measurement location ID	11	UNINT
	Total	179	

The class specific TEDS always include a template ID number, which in addition to the information included in the basic TEDS, identifies the transducer to any C&DH unit, such as a microcontroller, data acquisition device or computer interface, to which the transducer is mated. Additionally, the TEDS contain information that must be known to operate the transducer properly. In the case of the thermocouple, the class specific TEDS includes the operational data for the transducer which consists of the following three types of information: (a) the template ID for a thermocouple, which is specified to be 36, (b) what type of electrical signal it generates, which is a voltage signal, and (c) its minimum and maximum temperatures, which correspond to the minimum and maximum electrical outputs. These three types of information, together with the thermocouple mapping method and type, cold junction compensation requirements, and response time, provide a C&DH device the data needed to fully utilize a given thermocouple.

The remaining segments of data in the class specific TEDS for the thermocouple includes relevant administrative data, including the date of the last calibration, initials of the calibrator, how long until the calibration needs to be repeated, and the location of the thermocouple in question. These last four items can be written by the user to provide up-to-date information.

Definition of XEDS

The eXtensible Electronic Data Sheet (XEDS) are derived from the IEEE 1451.4 standard. Specifically, it comes from the section that outlines the use of user-defined TEDS in IEEE 1451.4. The purpose of using XEDS in this capacity is to allow the subsystems of a system-of-systems to be machine describable. This is to say that a given subsystem's XEDS should be able to describe its purpose, usage and control methods to the central command and data handling system. An XEDS is composed of the same basic elements of a standard TEDS, but with the transducer template replaced with the template for the sub-system that needs describing.

Defining Classes of XEDS for CubeSats

The XEDS template depicted below is that of a 1U CubeSat Chassis. The additional XEDS classes required to fill out the remainder of the satellite bus include the C&DH Board, the communications board or boards with subclasses for the 70cm transceiver, 70cm Beacon, and the dedicated 2m receiver. The remaining classes include the electrical power system or EPS, external photovoltaic systems and associated deployment devices, attitude control system (ACS), antennas and their respective deployment system.

A possible list of additional XEDS classes and sub-classes is below:

- Chassis
- Attitude Control System
- Command & Data Handling
- Communications
  - 70cm Band data transceiver
  - 70cm Band beacon
  - 2m Band Receiver
- Antenna Systems
  - Antenna module
  - Antenna deployment system
- Electrical Power system
- Photovoltaic Array
  - Photovoltaic module
  - Photovoltaic deployment system

Defining XEDS for a Physical Component

Each datum included in the chassis segment of the XEDS is included in the data sheet to provide a benefit to either the system integration of the satellite itself or for the designers and systems integrators of the CubeSat that the chassis is meant to be a part of. The items can be divided along those lines into two categories, primarily benefiting the satellite or primarily benefiting the designers and systems integrators.

There are eight data fields that are considered as primarily benefiting the satellite's C&DH system. These are:

- Template ID
- Serial Number
- System Type
- Ground Plane
- Number of Inhibits
- Number of Rail Type Inhibits
- Min Operating Temp
- Max Operating Temp

The Template ID is necessary to tell the C&DH unit or any other reader what to expect for the contents of the remainder of the TEDS. Also, the serial number and systems type are necessary to identify if a system is in a valid configuration. The minimum and maximum operating temperature are included to allow for the C&DH unit to maintain its operating temperature utilizing any onboard heating mechanism (i.e. battery heaters included in the satellites electrical power system). The number of inhibits and the number of rail-type electrical inhibits inform the C&DH system allowing it to account for the post launch time delay begin operating after launch, without needing it to be hardcoded into the satellite.

Of the remaining data fields, ten of them are more beneficial to the satellite's designers and systems integrators.

- Mass
- Standoff Thread
- Standoff Locations
- Internal dimensions
- Modulus of Elasticity
- Coefficient of Thermal Expansion
- Tensile Strength
- Thermal Conductivity
- Volume Resistivity
- External Dimensions

All of the fields listed above provide information that is critical to the satellite's other components being successfully integrated into the completed satellite. Having the chassis component's critical physical attributes, with respect to systems integration, stored on the individual component's TIM as an XEDS, allows for systems designers and integrators to quickly and easily access the most relevant information to ensure that a given satellite configuration will function, meet the given specification, and be allowed to fly.

The mass of the chassis and its associated hardware allows for quick calculation of the mass budget that a given chassis provides. The internal board mounting standoffs, along with the internal dimensions give systems designers and integrators the information needed to determine if a given COTS subsystem is compatible with the chassis that they have in front of them. The external dimensions are included so that it can be readily determined if a given chassis is within the acceptable parameters for a given type of CubeSat deployer.

Sample XEDS for a 1U CubeSat Chassis

A CubeSat chassis is an integral part of the satellite bus. Table 6 depicts the proposed XEDS Template for a CubeSat Chassis. Included in this template is the basic TEDS as outlined in the thermocouple example along with the class specific XEDS for a 1U CubeSat chassis of the sample spacecraft bus. The contents of the class specific XEDS can be broken down into three major categories: dimensional, operational, and interface details.

The dimensional component of the satellite chassis XEDS contains the internal and external dimensions of that chassis. The external dimensions are recorded with respect to the face-to-face distance of the farthest apart faces on each axis. The internal dimensions are recorded with respect to the face to face distance of the closet faces on the inside of the structure.

The operational component of the satellite chassis XEDS contains the relevant characteristics of the chassis with respect to its usage as a CubeSat. These characteristics include the mass of the chassis to allow for mass budgeting during the design process, and to supply the data to C&DH for its use. Also included is the operational temperature range of the chassis, and the number of electrical inhibitor switches and their types, in addition to if the chassis is a ground plane and its serial number and usage type. This data allows the C&DH system of the CubeSat to determine if the chassis it is connected to is the correct chassis for that system.

The Interface component of the satellite chassis XEDS contains the locations of the internal board mounting stand offs. Their locations are given with respect to the geometric center of the bottom face of the chassis. This can be seen in Figure 3.

Table 6.

Proposed XEDS Template for a CubeSat Chassis

Section	Description	Bytes	Data Type (and Range)
Basic TEDS	Manufacturer ID	14	INT (17-16381)
	Model Number	15	INT (0-32767)
	Version Letter	5	INT (0-63)
	Version Number	24	INT (0-16777215)
Chassis	Template ID	8	Integer (value = 50)
	Serial Number	24	INT (0-16777215)
	System Type	2	INT (0-Flight, 1-Engineering, 2-Test)
	Ground Plane	1	BOOL (0-No, 1-Yes)
	Standoff Thread	15	ASCII
	Standoff 1 X coordinate	8	SIGNINT (.01mm)
	Standoff 1 Y coordinate	8	SIGNINT (.01mm)
	Standoff 2 X coordinate	8	SIGNINT (.01mm)
	Standoff 2 Y coordinate	8	SIGNINT (.01mm)
	Standoff 3 X coordinate	8	SIGNINT (.01mm)
	Standoff 3 Y coordinate	8	SIGNINT (.01mm)
	Standoff 4 X coordinate	8	SIGNINT (.01mm)
	Standoff 4 Y coordinate	8	SIGNINT (.01mm)
	Min Operating Temp	8	INT (.1 K)
	Max Operating Temp	8	INT (.1 K)
	Units of CubeSat	4	INT (1-6)
	Internal X dimension	8	INT (.01mm)
	Internal Y dimension	8	INT (.01mm)
	Internal Z dimension	8	INT (.01mm)
	External X dimension	8	INT (.01mm)
	External Y dimension	8	INT (.01mm)
	External Z dimension	8	INT (.01mm)
	Mass	8	INT (.01g)
	Number of Inhibits	2	INT (0-4)
	Number of Rail Type Inhibits	2	INT (0-4)
	Modulus of Elasticity	16	INT (N/mm2) (MPa)
	COEF of Thermal Expansion	16	INT 10^{-7}K−1
	Tensile Strength	16	INT (N/mm2) (MPa)
	Thermal Conductivity	8	INT W/m*K
	Volume resistivity	8	INT 0.1nOhm*m
	Total	172	

Table 7.

Sample XEDS for 1U CubeSat Chassis

Section	Description	Value
Basic TEDS	Manufacturer ID	1400
	Model Number	1
	Version Letter	D
	Version Number	1
Chassis	Template ID	50
	Serial Number	10000001
	System Type	0
	Ground Plane	1
	Standoff Thread	Apr-40
	Standoff 1 X coordinate	-3734
	Standoff 1 Y coordinate	4286
	Standoff 2 X coordinate	3633
	Standoff 2 Y coordinate	4286
	Standoff 3 X coordinate	3887
	Standoff 3 Y coordinate	-4286
	Standoff 4 X coordinate	-4115
	Standoff 4 Y coordinate	-4286
	Min Operating Temp	80
	Max Operating Temp	-40
	Internal X dimension	9850
	Internal Y dimension	9850
	Internal Z dimension	11350
	External X dimension	9975
	External Y dimension	9975
	External Z dimension	9975
	Mass	14469
	Number of Inhibits	2
	Number of Rail Type Inhibits	0

The XEDS can also be stored and shown as an XML (eXtensible Markup Language) [39] that can be stored in various locations on the spacecraft bus. This markup format allows for easy reading of the XEDS by both computer systems and the engineers working on a given project. Using the XML format to encode the virtual storage of XEDS has two major benefits. XML's wide-spread use and modest complexity lowers implementation barriers, leading to ease of use due to commonality. Another benefit is that an XEDS structure encoded using XML can be made to mirror the byte wise storage of a TEDS or XEDS on a physical chip. Encoding the data in XML allows for the information to be easily machine-readable, which is an integral part of creating a machine-describable system. The structural layout described in 1451.3 allows all XEDS for a given system to be in either XML or byte-wise physical storage. Using both the TEDS and XML as templates for how to store system metadata is a boon for developing a method to create a fully machine-describable system. The XEDS shown in Table 7 can be seen in XML format below.

```
<XML>
<XEDS>
<Manufacturer ID> 1400 </Manufacturer ID>
<Model Number> 00001 </Model Number>
<Version Letter> D </Version Letter>
<Version Number> 000001 </Version Number>
<Template ID> 50 </Template ID>
<Serial Number> 10000001 </Serial Number>
<System Type> 0 </System Type>
<Ground Plane> 1 </Ground Plane>
<Standoff Thread> '4-40' </Standoff Thread>
<Standoff 1 X coordinate> -3734 </Standoff 1 X coordinate>
<Standoff 1 Y coordinate> 4286 </Standoff 1 Y coordinate>
<Standoff 2 X coordinate> 3633 </Standoff 2 X coordinate>
<Standoff 2 Y coordinate> 4286 </Standoff 2 Y coordinate>
<Standoff 3 X coordinate> 3887 </Standoff 3 X coordinate>
<Standoff 3 Y coordinate> -4286 </Standoff 3 Y coordinate>
<Standoff 4 X coordinate> -4286 </Standoff 4 X coordinate>
<Standoff 4 Y coordinate> -4286</Standoff 4 Y coordinate>
<Min Operating Temp> 80 </Min Operating Temp>
<Max Operating Temp> -40 </Max Operating Temp>
<Internal X dimension> 9850 </Internal X dimension>
<Internal Y dimension> 9850 </Internal Y dimension>
<Internal Z dimension> 11350 </Internal Z dimension>
<External X dimension> 9975 </External X dimension>
<External Y dimension> 9975 </External Y dimension>
<External Z dimension> 9975 </External Z dimension>
<Mass> 14469 </Mass>
<Number of Inhibits> 2 </umber of Inhibits>
<Number of Rail Type> 0 </Inhibits Number of Rail Type
Inhibits>
<Modulus of Elasticity>290</Modulus of Elasticity>
<Coefficient of Thermal Expansion>238</Coefficient of
Thermal Expansion>
<Tensile Strength>228</Tensile Strength>
<Thermal Conductivity>290</Tensile Strength>
<Tensile Strength>499</Tensile Strength>
```

```
</XEDS>
</XML>
```

Chapter 5

Example Implementation

The contents of this chapter detail the design methodology and implementation of a proof of concept demonstration of applying eXtensible Electronic Data Sheet (XEDS) developed as part of this thesis project to document and communicate key information about a 1U CubeSat chassis. The goal is to demonstrate that a microcontroller functioning as a basic C&DH board can differentiate between information contained in user-defined XEDS that resides on three chips. The hardware used in the demonstration consists of a Texas Instruments MSP430 microcontroller and three ADESTO RM24C256DS resistive memory chips. The ADESTO resistive memory chips are the same type that flew on the MemSat mission.

In this demonstration, the Texas Instruments microcontroller is programmed to be able to use information contained in the XEDS' to determine whether the chassis is the expected, acceptable chassis or not. The determination is made based on two pieces of information; namely, the twenty-four-bit chassis serial number and the two-bit system type metadata entry contained in the user-defined XEDS as outlined in Chapter 3 Table 6.

Each of the three resistive memory chips contains copy of the sample chassis XEDS shown in Chapter 3 Table 7, but with slight variations in the serial number and system type metadata that distinguish each chip from the other two. The XEDS on chip #1 contains what is considered to be the correct eight-digit integer serial number (10000001) and correct system type (0, which signifies "flight") expected by the microcontroller. Chip #2's XEDS contains mostly the same data, including the correct serial number, but with a different system type code (1, which indicates "engineering").

Chip #3's XED contains a different, incorrect serial number (1000000 2), but contains the desired system type code (0, for a "flight" designation).

XEDS Architecture Description

The General XEDS Architecture consists of a C&DH unit and the individual memory chips (analogues to the TIMs in an IEEE 1451.4 system) of each subsystem. In this system, the C&DH unit would function as the master controller, with all the additional subsystems, and their respective memory chips in this case, functioning as the slaves in the network. This can be seen depicted in Figure 15. The C&DH unit would need to be able to both read and write to the individual memory components. This is so that it can function not only to read and use the XEDS, but to write to them. This writing capability allows for updating individual XEDS with things such as up to data calibration data, or changes in configuration to due to damage in flight.

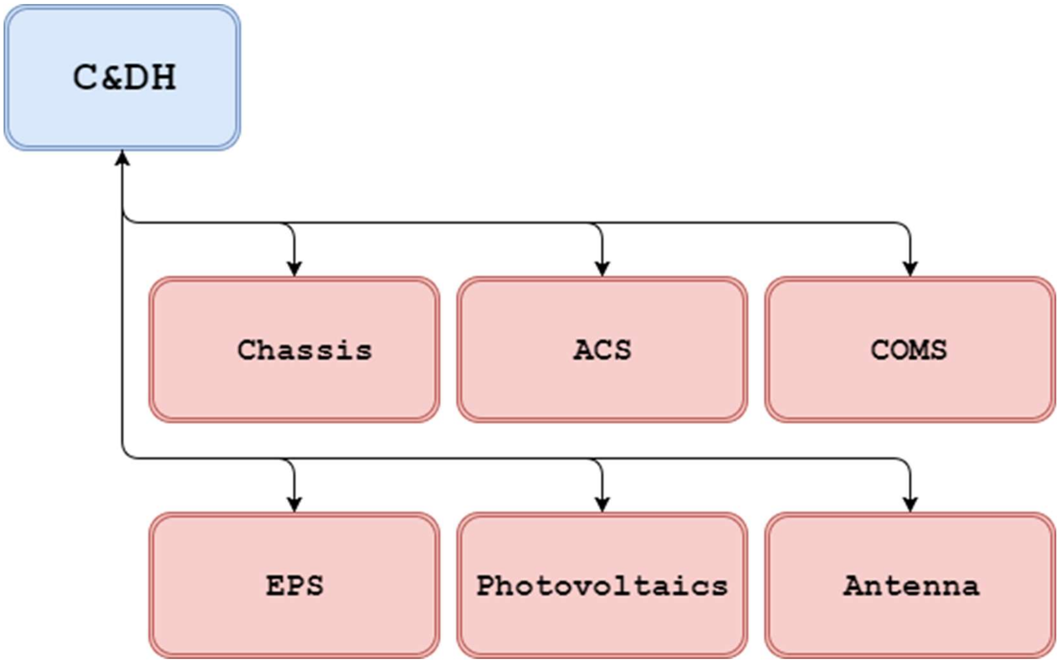

Figure 13. XEDS Bus for a CubeSat

Chip Implementation Considerations

The method of implementation for chips that are to be physically and electrically integrated into a satellite has a few basic requirements. Firstly, a chip needs to be affixed to the spacecraft structure in a secure enough manner to survive launch and deployment. Secondly, the chip needs to connect to the central C&DH unit in such a way that it can be easily accessed by the system. Thirdly, it also must be accessible when the spacecraft is not fully assembled, so that the chip's XEDS can be read and written to for administrative purposes. These chips can be either non-volatile flash memory, or what was used in this the implementation described here, resistive memory. Various communication protocols can be used to integrate a chip into a satellite's system, each with its own advantages and disadvantages.

Communication protocols

There are several electronic communications protocols that are suitable for facilitating communication between the memory chips containing the XEDS and the C&DH system. In this section, information on six protocols are presented. The six are commonly known by their abbreviated names: UART, I2C, SPI, CANBUS, MIL 1553, and 1-wire.

The UART protocol differs from the others described in this section as it is not specifically a protocol per se, but rather is a name for the hardware that uses this method of serial communication. A Universal Asynchronous Receiver-Transmitter or UART is a concept that dates back to the early days of the telegraph, wherein data is sent via pulling the signal line high or low, similar to the dots and dashes of the aforementioned telegraph. The basic functionality of asynchronous serial communication was conceptualized by Gordon Bell of DEC with PDP-1 in 1959 [54], [55].

A UART bus/modem consists of a Clock generator, the input and output registers, and then the additional logic to control reading and writing. There is no synchronized clock being sent between a pair of UARTS, as the clock speeds must be selected before the start of a data transfer.

UART is not ideal. These data rates are measured in baud, meaning the number of times a signal changes its state per second. Common UART speeds, or baud rates, are 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200, with 9600 baud being the most common, especially among applications were speed isn’t critical. The major downsides to using UART for our XEDS bus is the completely serial nature of the connection and a lack of a built-in addressing protocol. This leads to an addition layer of complexity that does not make its usage not ideal.

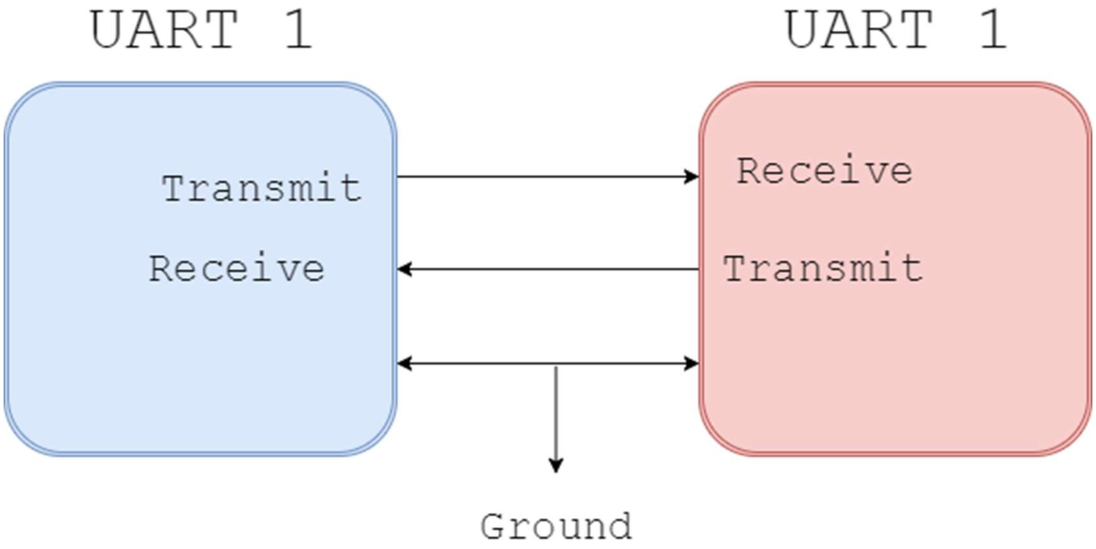

Figure 14. UART Bus

The Inter-Integrated Circuit (I2C – pronounced 'I squared C') is a serial half duplex electronic communications protocol. It was developed in 1982 by NXP Semiconductors (then Philips Semiconductor) [56]. I2C is commonly used in applications where the speed of a device's communication to a processor is not a high priority, The I2C protocol functions on a master-slave system, with the master generating the clock signal and being the only node that can initiate communication on the bus. It commonly functions at 100kHZ, 400kHz, but in some applications can reach speeds up to 3.4MHz [56]. I2C also functions on a seven-bit address system, allowing up to 127 slaves per master, with the possibility to use I2C switches and expanders to reach even more chips.

The I2C protocol operates over a two-wire bus, which along with power and ground means that an I2C memory chip requires two pins plus common power and ground, or four wires for external subsystems, in order to function. This allows the I2C bus for the XEDS storage to take up very little in the way of the electrical connections, leading to easier packaging into systems. Another benefit of I2C is that it has a built-in addressing system, unlike UART. This allows for easier integration of the individual I2C memory chips into a single bus for the usage of the C&DH system.

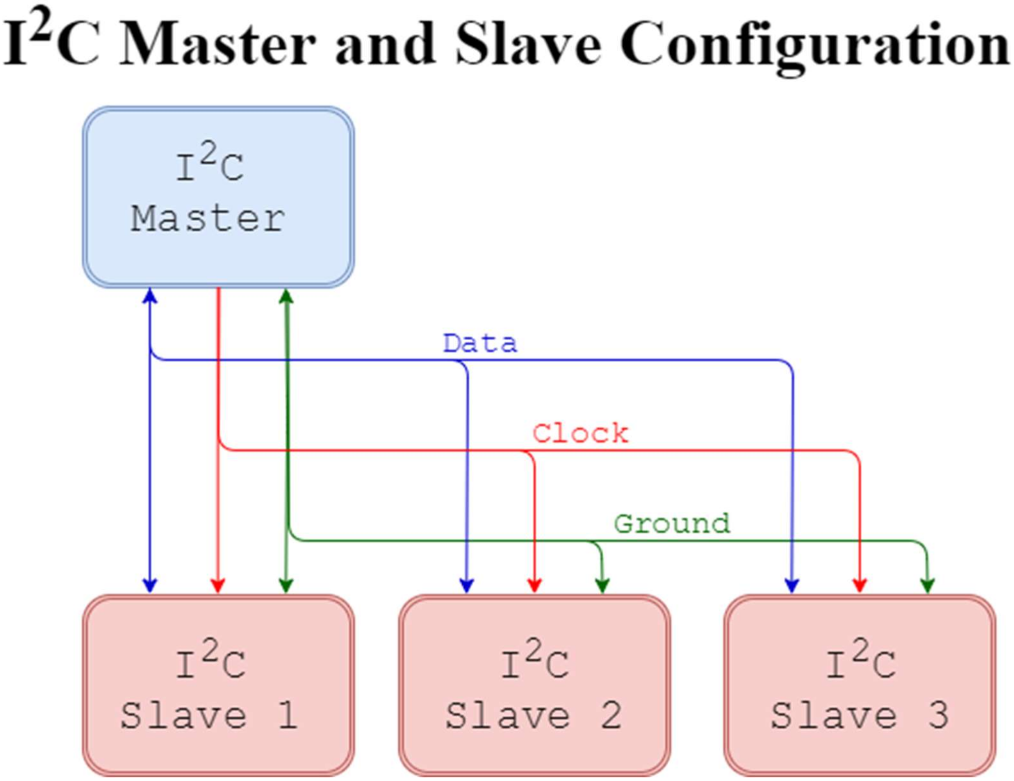

Figure 15. I2C Bus with Multiple Slaves

The Serial Peripheral Interface (SPI) is another type of serial communication interface, that was originally developed by Motorola in the 1980's [57], [58]. The interface differs from I2C due to its usage of a single master in its master-slave architecture as opposed to the multi-slave systems possible with I2C. Another noteworthy difference between the two protocols is the full duplex nature of SPI allowing to it to perform read and write operations at the same time, theoretically increasing the speed at which the bus performs. In addition, SPI does not have a maximum clock speed like I2C does [59]. When building a compact device, such as a nanosatellite, this at times can lead to packaging complications in an already cramped environment [58].

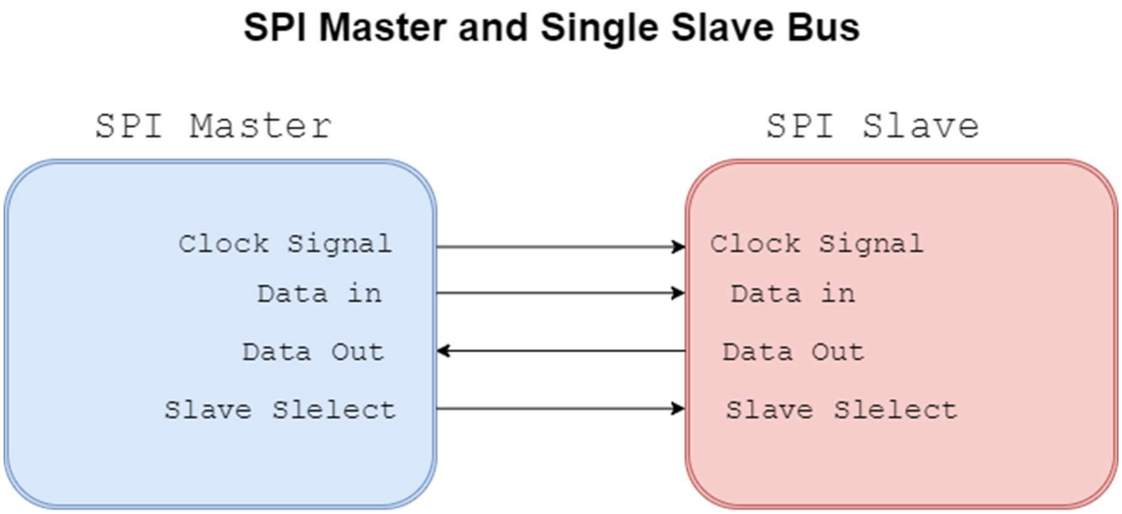

Figure 16. SPI Bus with a Single Slave

The Controller Area Network (CAN bus) is an electronic communications protocol developed primarily for automotive applications [60], [61], [62]. The main advantage that CAN bus has over the other electronic communications protocols discussed here is that CAN bus is a host-less system. This means that all the nodes on the CAN bus network are of equal status, with an added benefit of all nodes seeing all communications that are sent on the network.

A shortcoming of using CAN bus in a spacecraft bus application is the need to preset all nodes with unique node identifier. In the case of different components having the same node id, (either via different vendors setting duplicate node identifier, or user configuration errors) the network will simplify fail to start. This failure condition is not a desired outcome in the context of doing the systems integration of a satellite, as it adds an extra layer of compatibility that can inhibit the ability for plug-and-play operation.

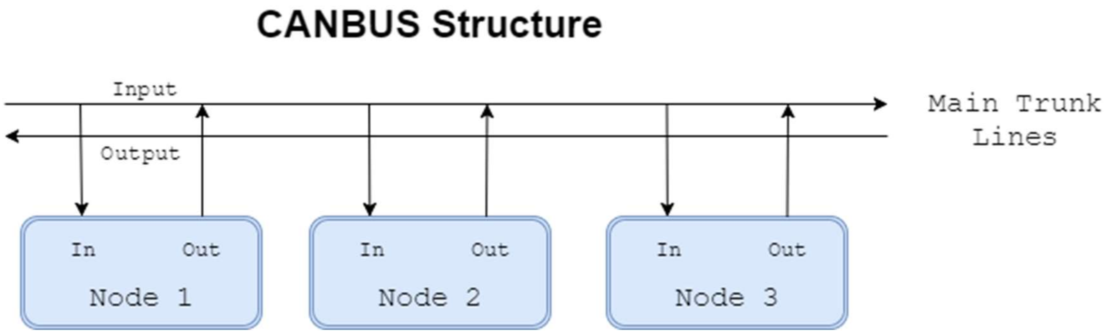

Figure 17. CANBUS Node Structure

MIL-STD-1553 is a military standard that defines an electronic communications protocol used for military application published by the United States Department of Defense [63]. Originally designed with the intent of being used primarily for avionics in

1973, it has also become commonly used in spacecraft data handling systems. The first major usage of the standard was in the avionics suite of the General Dynamics F-16 Fighting Falcon. MIL-STD-1553 is also known as NATO STANAG 3838 AVS and used around the world [64]. MIL-STD-1553 has a sister standard, known as of MIL-STD-1773, which uses optical connections instead of electrical ones. The base composition of a single MIL-STD-1553 Bus consists of a pair of wires with a 70–85 Ω impedance at 1MHz, which is capable of 1.0 Megabits per second, or 1 bit per microsecond.

A benefit of using MIL-STD-1553 is that there is a built-in capacity for redundancy in the system. Any data handling system built on the MIL-STD-1553 architecture has the ability to run a fully capable backup bus controller identical to the systems primary bus controller.

As seen in Figure 17, MIL-STD-1553 system consists of a pair of bus controllers, generally two to four redundant bus connections, a bus monitor, and a number of remote terminals. The bus monitor allows for monitoring of the communications that travel through the bus. The remote terminals can be either connected to a subsystem via another electronic communications protocol or be an embedded part of the subsystem.

The relatively widespread usage of MIL-STD-1553 in modern spacecraft applications means that a reasonable amount space rated hardware that is compatible with this protocol is already is available, which can be viewed as advantageous.

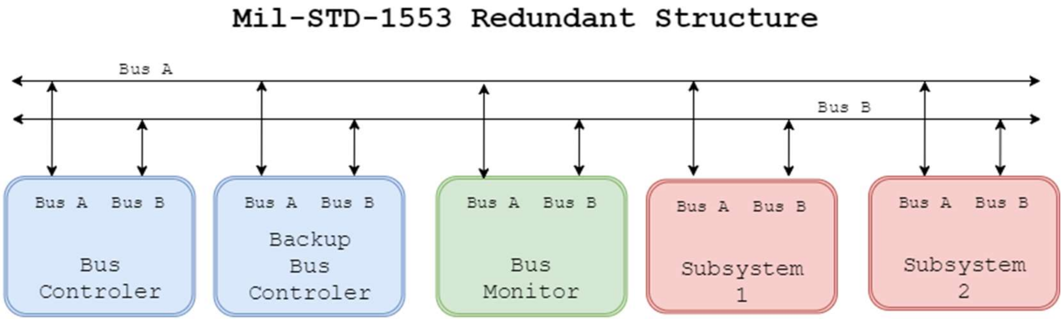

Figure 18. Mil-STD-1553 Bus with Redundant Controllers and Busses

The 1-Wire communications bus is a protocol/system developed by Dallas Semiconductor Corp (which has since been acquired by Maxim Integrated Products) [65]. The purpose of the 1-Wire protocol is to provide a low speed data communication network over just a single wire and a ground [66]. It is also possible to create a network of 1-Wire devices, but to do so requires an additional master device, known as a MicroLAN.

The 1-Wire bus consists of a single powered wire, and then a ground. Power for the attached device is therefore cut intermittently during data transmissions. This is mitigated by the inclusion of a capacitor in the device to keep the internals of the connected device powered during transmitting and receiving.

To use a 1-Wire bus for the XEDS network, a MicroLAN must be used. The single master in the MicroLAN would be either the C&DH main processor or a 1-Wire bridge, which Maxim Integrated Products produces [67], to turn an I2C into 1-Wire connection. Utilization of this bridge would allow for the XEDS of physical subsystems (such as the satellite chassis), or subsystems not contained in the main body of the satellite (such external photovoltaic arrays) to be integrated into an I2C XEDS network. The benefit of doing so would be to ease packaging even more, by lowering the number of connections to these difficult to reach places from four to two wires, and in some cases only adding one addition conductor, if there is already a ground going to a given external panel [68].

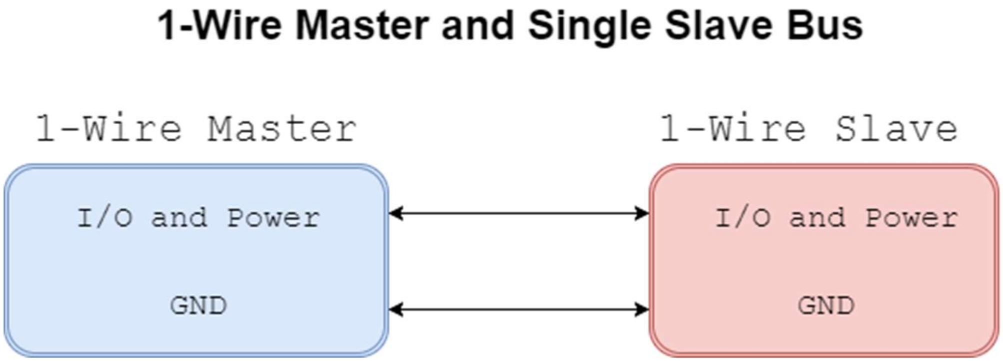

Figure 19. 1-Wire Bus with single slave

Wireless Space Plug-and-Play

The sections above have document various, select details concerning a series of communications protocols. These protocols and standards address factors related to modern satellites in general, though the primary focus here is on nanosatellites, and accordingly takes into account the difficulties of working inside of the tight confines of a nanosatellite.

This Wireless Space Plug-and-Play Architecture is attempting to achieve a similar outcome utilizing a wireless XBEE derived transceivers (XBEE is a COTS wireless transceiver that uses the IEEE 802.15.4 protocol [69]) to transmit TEDS between subsystems [70]. However, wireless communications among subsystems are not considered suitable in nanosatellite applications at this time, as the size and additional power consumption of these transceivers, in addition to the close proximity of the subsystems, renders it a non-feasible.

Memory Architectures

Flash Electrically Erasable Programmable Read-Only Memory (EEPROM) is a type of nonvolatile computer memory that is commonly used to store data in high-speed high-density applications [71]. It is comprised of an array of floating gate transistors. Flash EEPROMs are therefore susceptible to radiation damage, as high energy particles can destroy the gate of a given transistor and render the bit dead. A similar issue can be seen with the dead pixels in the digital images taken from the International Space Station [72].

An alternative memory construction that may alleviate this problem is that of resistive memory. There are two types of resistive memory that will be discussed in this section, memristive memory, and the conductive-bridging random access memory (CBRAM) that is used in this implementation [73]. Memristive memory is based on memristors, which are nonlinear resistive circuit elements first proposed by Leon Chua in 1971 [74]. However, it was not until 2007 that engineers at HP Labs in Palo Alto, California reported on the creation of the first practical memristor implementation [75]. The HP memristor was created using three layers consisting of a titanium dioxide substrate between a titanium electrode and a platinum electrode. The memristor functions off of the oxygen atoms in between the two electrodes changing their position in response to charge moving across the memristor. After a point, the resistance cannot increase anymore and the memristor has reached a point known as the hysteresis point, where it will function as a passive resistor until the flow of electricity through it is reversed. It is this effect that allows a memristor to be used as a memory cell.

The CBRAM used in the implementation being studied in the current work functions similarly to the HP created memristor but differs in its method of manufacture. Whereas the oxygen atoms in a memristor must be supplied at the time of its creation, electrodes provide the metal ions that are needed for the CBRAM to function. Both of these types of memory can be read by passing a known voltage over the cell, and measuring the resultant resistance, as long as the voltage used is underneath the threshold to change the state of the cell. Due to its construction not requiring the potently fragile gates that exist in flash memory, both memristive and CBRAM based memory chips could be more durable in space-borne applications, which is why the CBRAM chip used in this test implementation, the ADESTO RM24C256DS [76], is one of the types of chips flown on the Rowan University MemSat CubeSat mission [16].

Describe Implementation

The following segment contains a more detailed description of the example test implementation. The goal of this test implementation is to take three separate ADESTO RM24C256DS CBRAM [76] chips and a Texas Instruments MSP430G2553 [77] microcontroller in an MSP-EXP430G2 LaunchPad development tool and emulate connecting three different XEDS configurations and returning a verdict on if that XEDS configuration is the correct one for the ‘spacecraft’ that is being assembled.

The three RM24C256DS chips had all three of their address select pins pulled to ground. This gave them the resulting I^2C binary address of ‘1010000’ or a hex address of 0x50. These chips were manually selected by only providing power to the chip that was being read. This is to simulate all potential chassis for a given satellite bus having the same I2C address to allow for compatibility, and to emulate each chip being hooked up to the hypothetical satellites C&DH system for validation. Each of the three RM24C256DS chips was loaded with a variation of the sample XEDS for a 1U CubeSat chassis seen in Table 7. The variations in the XEDS by chip can be seen in Table 8.

Table 8.

XEDS Variation Across Test Chips

Chip #	1	2	3
Serial Number	10000001	10000001	10000002
Type	0 (Flight)	1 (Engineering)	1 (Engineering)

All other fields not shown are identical to each other and to the fields found in Table 7. The procedure for the test was to individually write the XEDS to each chip. Then a separate comparator program was loaded onto the MSP430G2553 chip. This program was written to compare the serial number and system type on the selected chip whenever a push button was pressed. The program was looking for a serial number of ‘10000001’ type designation of ‘0’ indicating that the chassis in question was flight hardware, with a matching serial number. Upon matching both parameters, the program would activate a green LED next to the trigger button. If one of both parameters did not match with the expected values, the program activated a red LED next to the trigger button. The nature of this test implementation was to design and emulate the fact that a given XEDS is going to be written very few times, while it most likely will be read many times, due to the nature of it being stable metadata.

Chapter 6

Discussion

Additional Thoughts

This chapter contains the discussion of the information presented in this thesis and the conclusions drawn from them. First, in order to provide some context to the discussions that follow below, some foundational topics will be revisited.

This thesis was written primarily through the lens of nanosatellite development and systems integration. More specifically, the spacecraft bus and the chassis described in the Extensible Electronic Data Sheets (XEDS) that are central to this work are based on products from Pumpkin Aerospace [49], [50], [51], that were used in the development of the Rowan University MemSat [16], which was a 1U CubeSat. Additionally, many of the observations and impressions expressed in this section regarding challenges that can be encountered when designing, constructing, and testing a nanosatellite, as well as standards and procedures that can facilitate the process, are derived from the experiences of the author working on the Rowan University MemSat project during 2017 and 2018. Accordingly, elements of a systems engineering perspective are interwoven with discussion of specifics related to the electronic components, applicable protocols, and evolving standards most closely connected with this project.

Nanosatellites are satellites that are between 1 and 10kg in mass. Therefore, CubeSats which weigh from 1.33 to 7.98kg are considered nanosatellites. The CubeSat specification was developed by CalPoly [15] to provide universities and other, smaller entities the ability to produce small spacecraft for performing research at a fraction of the cost of larger more traditional satellites. It is in this vein that this thesis approached the idea of further reducing the costs, both in time and money, of developing a CubeSat by adapting and applying the proven Internet of Things-based principles of the IEEE 1451 standard and its Transducer Electronics Data Sheets (TEDS) [28].

Much prior work has applied a system engineering approach to spacecraft in general [48], including CubeSat design approaches (e.g., [78]) and reviews of nanosatellite mission successes and failures (e.g., [79], [80], [81]). Though it is beyond the scope of this work to comprehensively test systems engineering hypotheses, the following discussion is informed by and appreciative of prior works and their previously documented general lessons learned, even as they are re-learned as challenges arise in the context of a specific, new university-based satellite application.

From its inception in 2008, NASA's Educational Launch of Nanosatellites (ELaNa) program and its CubeSat Launch Initiative (CSLI) have been popular and successful at providing a relatively low cost means for universities and others to fully participate in the design, construction, and testing of nanosatellites before launch and in the gathering and analysis of mission data after launch [17], [82]. Within this framework, university-class CubeSats are considered to be good mechanisms by which to provide hands-on, high-tech educational opportunities that often feature technology demonstrations as a mission goal [27]. The MemSat mission can be considered a representative example of university-based CubeSat projects, in that it served both an educational purpose by affording several Rowan engineering students opportunities to learn by working on a nanosatellite project, and it did so in in pursuit of a mission goals aimed to test the durability, reliability, and other performance characteristics of memristive memory chips in a space environment [16], [39], [81].

The approximately twenty-nine-month timeline of Rowan's MemSat mission, from NASA's approval of the project as part of the ELaNa program to its deployment from the International Space Station (ISS), is not atypical of the relatively rapid lifecycle associated with university-based CubeSat missions. The announcement that MemSat was among the twenty candidate nanosatellites selected by the seventh round of NASA's CubeSat Launch Initiative (CSLI) to be part of the ELaNa program was made on February 18, 2016, and was subject to final negotiations before officially becoming eligible for placement on a launch manifest [83].

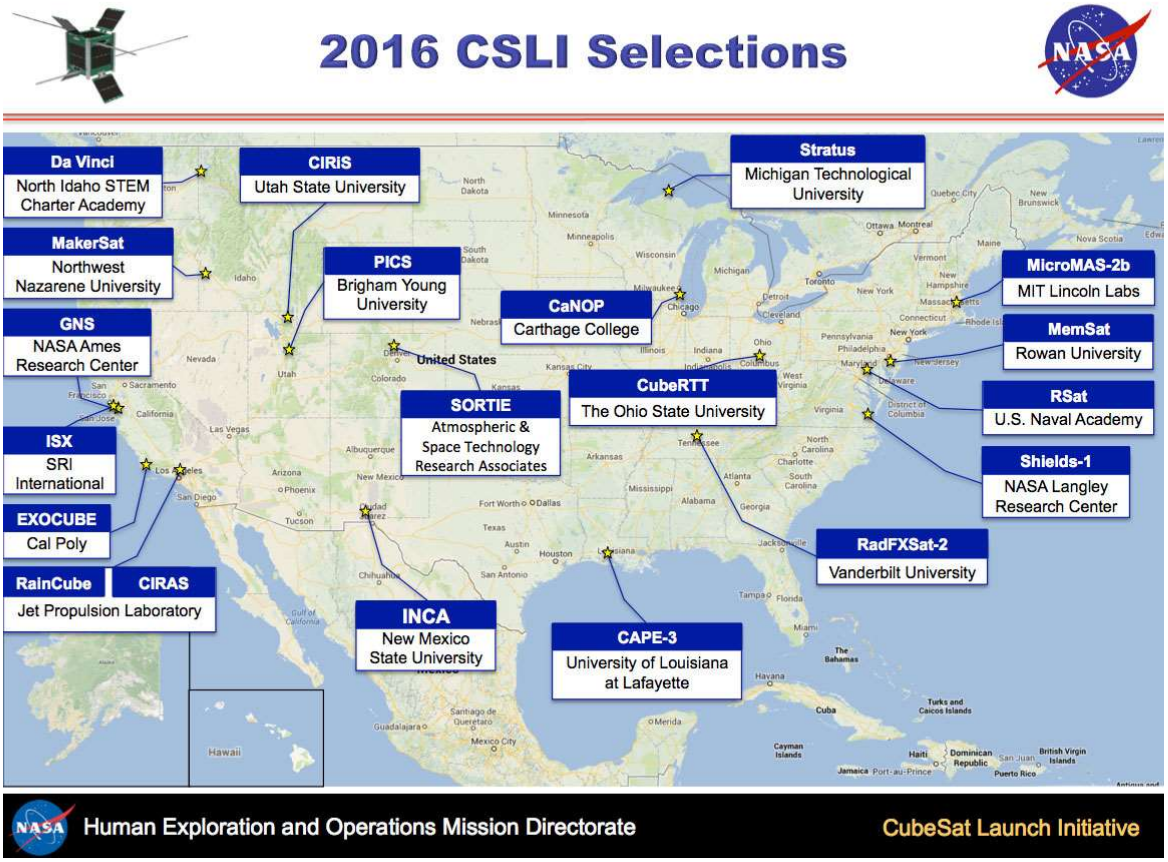

Figure 20. Locations of ELaNa 23 CubeSat Teams (NASA*)*

Twenty-seven months later, on May 21, 2018, MemSat was part of a Cygnus resupply spacecraft mission [84] that was launched from NASA's Wallops Flight Facility in Virginia on an Antares 230 rocket [84], [85]. The Cygnus spacecraft (known both as Cygnus CRS OA-9E and Orbital ATK CRS-9) rendezvoused with the International Space Station (ISS) on May 24, 2018, at which time it was grasped by the ISS robotic arm and successfully guided into its berthing port [86]. On July 13, 2018, the NanoRacks CubeSat deployer system, on which MemSat, six other ELaNa 23 missions, and two commercial CubeSats were loaded (a mixture of 1U, 3U and 6U CubeSats), was successfully deployed from the ISS into low-Earth orbit [87], [88].

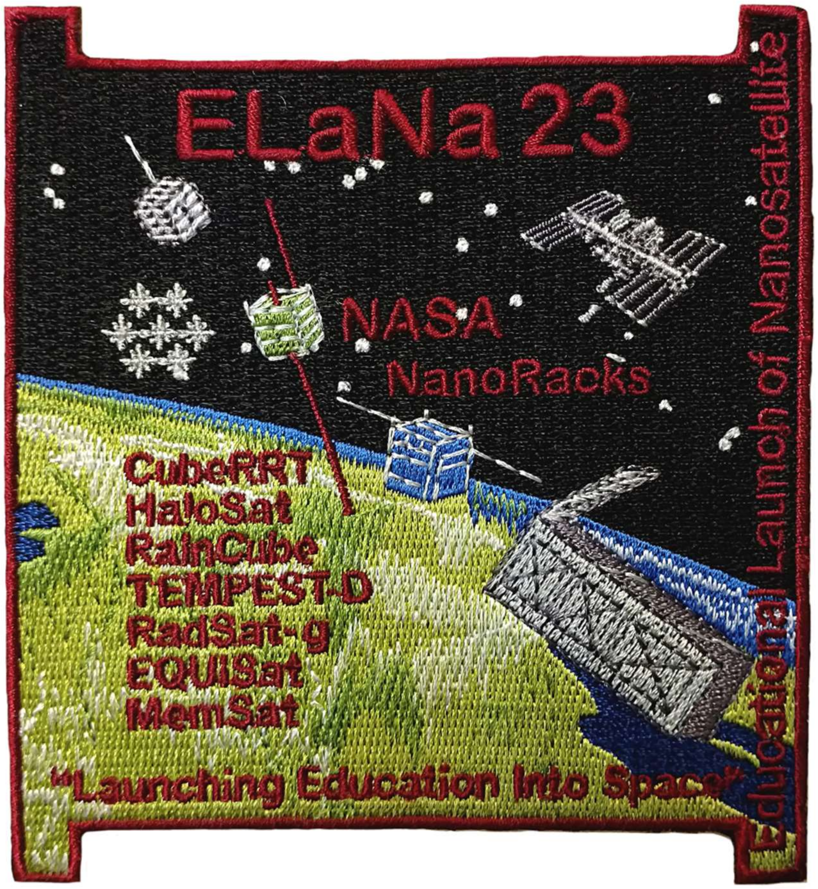

Figure 21. Photograph of the ELaNa 23 mission patch listing MemSat and the other six ELaNa 23 CubeSats that were deployed from the ISS on July 13, 2018.

On the day of MemSat's successful deployment, Dr. John L. Schmalzel, professor of Electrical and Computer Engineering at Rowan University and a Fellow of the IEEE, was quoted in the media as stating, in reference to the MemSat project efforts as a whole, "There's an awful lot of work that goes into this. Not only do you have to design things, but you also have to test things and tests often fail," [89]. Dr. Schmalzel's statement alludes to the amount of effort expended within the approximately twenty-nine-month period that elapsed from the announcement of MemSat's acceptance to its launch and deployment in space, as well as the efforts expended earlier to put together the proposal submitted to NASA. Those efforts include, roughly in sequential order, though some work was done in parallel, hardware, software and systems design, procurement of required parts, assembly, and various tests of individual components and integrated parts of the MemSat system. This encompassed not only the physical CubeSat itself, but also ground tracking station positioned on top of the Rowan University Engineering building and multiple software packages used in MemSat's development, testing, and eventual operation.

In practice, the period of active, hands-on MemSat work at Rowan University by faculty and students was shorter than twenty-nine months. At the start, there was a multi-month lag from the NASA announcement to the initiation of hands-on work. And at the end, MemSat was shipped to NanoRacks, LLC in Houston for integration into the multi-CubeSat NanoRacks deployer months before it was launched in May 2018 and ultimately deployed from the International Space Station in July 2018.

As described previously, the interoperability and plug-and-play capabilities of modern, COTS smart transducers that has resulted from the adoption of common protocols and standards such as IEEE 1451, has contributed markedly to their widespread presence in today's consumer and industrial devices and systems. The proposed XEDS extensions documented in this thesis aim to leverage and expand the IEEE 1451 family of standards to apply to satellite systems.

If adopted, the expanded standards should aid the future development of satellites in general, especially CubeSat missions which are planned, built, and tested on relatively short time lines, at modest costs, by teams consisting of students and faculty. Having

IEEE 1451 compatibility amongst satellite components would be a beneficial advancement that is more evolutionary than revolutionary, but not a cure-all (complete solution) for the diverse set of challenges that can be encountered when developing a CubeSat on tight timelines and limited budgets. Still, if an improved, more comprehensive family of standards were developed and adopted, it likely would improve interoperability and thus shorten the amount of time associated with testing, debugging, and retesting, the probability of a successful mission could be increased.

A study of the first one hundred CubeSats reported that upward of one half of the early university-based CubeSat missions failed to meet their operational objectives [79]. Though categorizing in a binary fashion whether or not a mission is deemed a success can be somewhat subjective, that early assessment is not inconsistent with later studies that review some of the common challenges university CubeSat efforts can encounter that negatively impact the mission meeting its operational objectives. For example, Richard Welle of the Aerospace Corporation used the term “CubeSat Paradigm” to describe university-based satellite programs for which the student participants’ educational experiences are a main goal such that mission failure, while not desirable was tolerable, given the relatively modest financial cost [67]. Welle’s analysis, and that of Massachusetts Institute of Technology graduate student Zachary Decker [81], identify insufficient integrated testing of satellites before they are shipped for launch as being a common pitfall among university CubeSat efforts. Tight timelines and delays in COTS deliveries are factors that contribute to testing strategies that focus more on component testing rather than testing of the integrated system. Though enhanced IEEE 1451 standards developed with satellite applications in mind obviously will not prevent troublesome COTS delivery delays, increased operability supported by improved and expanded standards, such as the XEDS developed in this work, potentially can reduce the time needed for debugging and allow for more comprehensive system testing.

Results of the Test Implementation

The purpose of the test implementation described in Chapter 5 was to show that XEDS can be written and read from relatively inexpensive, readily available resistive/EEPROM memory and be used for a useful function, which in the case of this example was testing if a part was the correct serial number and was flight hardware. The MSP430G2553 that was used polled the selected chip via I^2C and read back the relevant portion of the XEDS, starting from the template number, and encompassing the serial number and hardware type (flight/engineering/test) and compared it against the values that it was searching for.

Conclusions on communication protocols for XEDS implementations in Nanosatellites

Various communications protocols were discussed in Chapter 4 with regards to implementing an IoT network inside the confines of a 1U CubeSat. The following findings resulted from that discussion.

The ideal communication protocol to use for creating a network of XEDS inside of a CubeSat is a pair of protocols. The preferred solution is a combination of I^2C for internal subsystems and 1-Wire for external subsystems and the chassis. These conclusions were reached based on a few factors. Firstly, as CubeSats do not have a lot of free space on the inside for the routing of wires, and a limited number of conductors are available on the internal headers, the two wires required for I^2C and the 1 wire required for a 1-wire bus help to alleviate packaging problems. Secondly, the relatively low data transfer speeds of I2C and 1-Wire are inconsequential due to the nature of XEDS operation, as once the data is accessed, it does not need to be reacquired until the next system startup. Thirdly, the two protocols can work together via the I2C to 1-Wire bridges that Maxim Integrated Technologies produces, such as the DS2484 single channel converter [90] and the DS2480B-800 8-channel converter [67]. Finally, there is widespread commercial-off-the-shelf availability of memory devices and controllers for both protocols. A runner up option would be the MIL-STD-1553 standard due to its widespread use in the aerospace industry leading to availability of hardware, and its built-in ability to accept redundancy out of the box.

Conclusions

In this thesis, user-defined Extensible TEDS (XEDS), as derived from the IEEE 1451.4 standard, were applied to a satellite subsystem. The adapted XEDS data structure contains the information to make a satellite chassis subsystem machine-describable. This was demonstrated for one subsystem, a COTS chassis. Additional templates for the subsystems described as being part of a CubeSat spacecraft bus can be found in the appendix. This concept was tested successfully via physical implementation of a microcontroller and a resistive memory chip. The code for this test can be found via the GitHub link in the appendix.

Literate reviews of the success and failure rates of CubeSat missions [78], [79], [80], coupled with the Rowan University team’s experiences with MemSat in 2018, strongly suggest that problems encountered during the satellite development phase often lead to reduced time being allotted to system testing - a factor that can negatively impact mission success rates. This thesis describes an approach to interoperability through the use of XEDS descriptions of satellite components that facilitate interoperability and plug-and-play like behavior. Further reductions in development and testing time for CubeSats should also result if the XEDS approach is refined and developed into a standard. A standard means that the widest possible user base could benefit.

Chapter 7

Future Work

Propagate XEDS

The concept of a nanosatellite being integrated primarily through XEDS needs to be tested further, however, the full assembly of a CubeSat, or bench test analogue, is outside of the scope of this thesis.

The prototype XEDS standards developed and presented in this work are potentially a first step towards realizing the benefits of self-describing subsystems in nanosatellite missions at some time in the future. The concept of a nanosatellite being integrated primarily through XEDS needs to be expanded, tested further and refined through an iterative process. The eight templates and single successful test performed with XEDS representative of a satellite chassis, though covering a limited set of subsystems, can serve as a roadmap for follow-on work. For example, testing of a fully assembled CubeSat or bench test analog, are reasonable steps that could build upon this work; however, those steps are outside of the thesis scope. Additionally, it is expected that should the XEDS method and associated templates developed herein be deemed promising, they would need to be distributed, refined, expanded, and tested further. This necessary step arises in part because there exists a large variety of configurations for CubeSats (e.g., in size, mission type, and subsystem design) that present a diversity of engineering requirements.

Propose Development of Standard for Nanosatellite Bus

The stated goal of this proposed standard is to enable the subsystems of a compliant nanosatellite to be machine describable in such a manner as to facilitate the plug-and-play operation of the various subsystems. This effort would be undertaken to provide the effect of lowering the cost of nanosatellite development and to shorten both the development and overall lead times for the deployment of these nanosatellites. Therefore, it is the opinion of the author that efforts to develop a standard should be pursued.

The obvious home for such a standard would be within the IEEE 1451 standard family. IEEE 1451.4-2004 describes the possibility of creating user TEDS. An XEDS would follow the same overall structure as other defined TEDS, including a basic TEDS, a specific data template, and the provision for additional calibration templates. The basic TEDS and calibration templates would closely follow the IEEE 1451.4 standard; the data template would be populated with CubeSat specific information.

The data that composes these XEDS shall be stored in one of two potential methods, ether byte-wise on a dedicated memory device, such as the Adesto chips described in chapter 4. Interface to local memory could follow the methods described in 1451.4 for Class 1 devices. Data can also be stored virtually in local storage or in the Cloud.

Several steps need to be undertaken for the proposed standard development effort to be successful including (1) identifying a sponsoring Technical Committee within an IEEE Technical Society (e.g., TC-9 under the Instrumentation and Measurement Society), (2) Obtaining a project authorization request (PAR) from IEEE allows work on the standard to officially occur, and (3) Members of the committee develop the draft standard and advance it to the point when it can be balloted by industry at large.

To further raise awareness for this work, an abstract has been submitted to the SmallSat Symposium 2019 [91], describing the need for standardization and the rationale for how standards of this type can benefit nanosatellite development. The SmallSat Conference provides a venue for discussions with members of the small satellite engineering community.

If efforts to raise awareness are successful with responses of others in the CubeSat community showing sufficient enthusiasm, the next steps would be to undertake the standards development process.

References

[1] National Aeronautics and Space Administration, “OSCAR 1,” NASA Goddard Space Flight Center, Greenbelt, Maryland, NASA Space Science Data Coordinated Archive NSSDCA/COSPAR ID: 1961-034B.

[2] J. Bouwmeester and J. Guo, “Survey of worldwide pico- and nanosatellite missions, distributions and subsystem technology,” *Acta Astronaut.*, vol. 67, no. 7–8, pp. 854–862, Oct. 2010.

[3] J. Chin *et al.*, “CubeSat 101: Basic Concepts and Processes for First-Time CubeSat Developers,” NASA CubeSat Launch Initiative, NP-2017-10-2470-HQ, 2017.

[4] E. Kulu, “Nanosatellite & CubeSat Database.” Aug-2018.

[5] D. Bearden, “Small-satellite costs,” *Crosslink*, vol. 2, no. 1, pp. 32–44, 2001.

[6] H. Heidt, J. Puig-Suari, A. Moore, S. Nakasuka, and R. Twiggs, “CubeSat: A New Generation of Picosatellite for Education and Industry Low-Cost Space Experimentation,” in *Proceedings of the 14th Annual AIAA/USU Conference on Small Satellites, Lessons Learned-In Success and Failure*, 2000, pp. 1–19.

[7] R. Zimmerman, D. Doan, L. Leung, J. Mason, N. Parsons, and K. Shalid, “Commissioning the World’s Largest Satellite Constellation,” in *31st Annual AIAA/USU Conference on Small Satellites, SSC17-X-03*, Logan, Utah, USA, 2017, p. 10.

[8] S. Bandyopadhyay, R. Foust, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, “Review of Formation Flying and Constellation Missions Using Nanosatellites,” *J. Spacecr. Rockets*, vol. 53, no. 3, pp. 567–578, May 2016.

[9] C. Henry, “Fleet details 100 nanosat constellation for Internet of Things connectivity,” *Space News Website*, 06-Dec-2017.

[10] J. G. Walker, “Continuous Whole-Earth Coverage by Circular-Orbit Satellite Patterns,” Royal Aircraft Establishment, Franborough, United Kingdom, RAE-TR-77044, 1977.

[11] J. N. Pelton and B. Jacqué, “Distributed Internet-Optimized Services via Satellite Constellations,” in *Handbook of Satellite Applications*, J. N. Pelton, S. Madry, and S. Camacho-Lara, Eds. Cham: Springer International Publishing, 2017, pp. 249–269.

[12] A. Narayanasamy, Y. A. Ahmad, and M. Othman, “Nanosatellites constellation as an IoT communication platform for near equatorial countries,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 260, p. 012028, Nov. 2017.

[13] T. Bowler, “The low-cost mini satellites bringing mobile to the world,” *BBC News Website*, 23-Feb-2018.

[14] R. Sandau, K. Brieß, and M. D’Errico, “Small satellites for global coverage: Potential and limits,” *ISPRS J. Photogramm. Remote Sens.*, vol. 65, no. 6, pp. 492–504, Nov. 2010.

[15] S. Lee *et al.*, “CubeSat Design Specification Rev 13.” [Online]. Available: https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/56e9b62337013b6c063a655a/1458157095454/cds_rev13_final2.pdf. [Accessed: 17-Feb-2019].

[16] S. Shin, “MemSat experiment webpage: NASA International Space Station.” 2018.

[17] G. L. Skrobot and R. Coelho, “ELaNa—Educational Launch of Nanosatellite: Providing Routine RideShare Opportunities,” in *Proceedings of the AIAA/USU Conference Small Satellites*, Logan, UT, 2012.

[18] National Aeronautics and Space Administration, “About ELaNa.” 03-Aug-2017.

[19] E. National Academies of Sciences and Medicine, *Achieving Science with CubeSats: Thinking Inside the Box*. Washington, DC: The National Academies Press, 2016.

[20] N. Jones, “Mini satellites prove their scientific power,” *Nature*, vol. 508, no. 7496, pp. 300–301, 17-Apr-2014.

[21] C. Keating *et al.*, “System of Systems Engineering,” *Eng. Manag. J.*, vol. 15, no. 3, pp. 36–45, Sep. 2003.

[22] P. Dersin, “Systems of Systems,” IEEE Reliability Society. Technical Committee on Systems of Systems, IEEE-RS-TC-SoS White Paper, Oct. 2014.

[23] IEEE SMC, “IEEE Systems, Man, and Cybernetics Society webpage.” .

[24] S. Chaieb, M. Wegerson, B. Kading, J. Straub, R. Marsh, and D. Whalen, “The OpenOrbiter CubeSat as a system-of-systems (SoS) and how SoS engineering (SoSE) Aids CubeSat design,” 2015, pp. 47–52.

[25] J. C. Lyke, Q. Christensen, and D. Anderson, “Lessons Learned: Our Decade in Plug-and-Play for Spacecraft,” in *28th Annual AIAA/USU Conference on Small Satellites*, 2014.

[26] K. R. Fowler and J. L. Schmalzel, “Sensors: the first stage in the measurement chain. 2. In a series of tutorials in instrumentation and measurement,” *IEEE Instrum. Meas. Mag.*, vol. 7, no. 3, pp. 60–66, Sep. 2004.

[27] T. R. Licht, “The IEEE 1451.4 proposed standard,” *IEEE Instrum. Meas. Mag.*, vol. 4, no. 1, pp. 12–18, Mar. 2001.

[28] National Institute of Standards and Technology, “IEEE 1451 web page.” NIST, Gaithersburg, MD, Jul-2017.

[29] National Instruments, “An Overview of IEEE 1451.4 Transducer Electronic Data Sheets (TEDS),” Aug. 2016.

[30] K. B. Lee and R. D. Schneeman, “A Standardized Approach for Transducer Interfacing: Implementing IEEE-P1451 Smart Transducer Interface Draft Standards,” in *Proceedings of SENSORS Conference ’96*, 1996.

[31] J. Schmalzel, F. Figueroa, J. Morris, S. Mandayam, and R. Polikar, “An Architecture for Intelligent Systems Based on Smart Sensors,” *IEEE Trans. Instrum. Meas.*, vol. 54, no. 4, pp. 1612–1616, Aug. 2005.

[32] K. Lee, “IEEE 1451: A standard in support of smart transducer networking,” in *Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference [Cat. No. 00CH37066]*, Baltimore, MD, USA, 2000, vol. 2, pp. 525–528.

[33] E. Song and K. Lee, “Understanding IEEE 1451-Networked smart transducer interface standard - What is a smart transducer?,” *IEEE Instrum. Meas. Mag.*, vol. 11, no. 2, pp. 11–17, Apr. 2008.

[34] D. Jethwa, R. R. Selmic, and F. Figueroa, “Real-time implementation of intelligent actuator control with a transducer health monitoring capability,” in *16th Mediterranean Conference on Control and Automation*, Ajaccio, France, 2008, pp. 1441–1446.

[35] IEEE Standards Association, “IEEE Project 1451-99 - Standard for Harmonization of Internet of Things (IoT) Devices and Systems web page.” 2017.

[36] National Institute of Standards and Technology, “Transducer Electronic Data Sheet web page.” Feb-2011.

[37] J. Christensen, S. Cannon, B. Hansen, and J. Lyke, “Automatic generation of sdm application source code from xTEDS,” in *SSC10-X-5*, Logan, UT, 2010, pp. 1–8.

[38] World Wide Web Consortium, “Extensible Markup Language (XML) 1.1, second edition.” World Wide Web Consortium, 29-Sep-2006.

[39] K. Konte, R. Trafford, and J. Schmalzel, “Implementing XEDS for a CubeSat communication subsystem,” in *IEEE Conference Proceedings: Sensor Applications Symposium (SAS)*, 2018, pp. 1–5.

[40] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” *IEEE Commun. Surv. Tutor.*, vol. 17, no. 4, pp. 2347–2376, 2015.

[41] F. Mattern and C. Flörkemeier, “From the Internet of Computers to the Internet of Things,” *Inform.-Spektrum*, vol. 33, no. 2, pp. 107–121, Apr. 2010.

[42] L. Atzori, A. Iera, and G. Morabito, “Understanding the Internet of Things: definition, potentials, and societal role of a fast evolving paradigm,” *Ad Hoc Netw.*, vol. 56, pp. 122–140, Mar. 2017.

[43] “Nest Company Homepage.” [Online]. Available: https://nest.com/.

[44] M. Haghi, K. Thurow, and R. Stoll, “Wearable Devices in Medical Internet of Things: Scientific Research and Commercially Available Devices,” *Healthc. Inform. Res.*, vol. 23, no. 1, p. 4, 2017.

[45] W. M. Kang, S. Y. Moon, and J. H. Park, “An enhanced security framework for home appliances in smart home,” *Hum.-Centric Comput. Inf. Sci.*, vol. 7, no. 1, Dec. 2017.

[46] L. E. Cohan, R.-D. Chambers, R. K. Lee, and J. E. Keesee, “Analysis of Modular Spacecraft Bus Design for Rapid Response Missions -RS4 2006-3001,” presented at the 4th Responsive Space Conference, Los Angeles, CA, 2006, pp. 1–9.

[47] P. W. Fortescue, J. Stark, and G. Swinerd, Eds., *Spacecraft systems engineering*, 4th ed. Hoboken, N.J: Wiley, 2011.

[48] B. M. Arnaut, D. D. Viot, L. L. Costa, G. Loureiro, E. E. Bürger, and M. Franchitto, “Systems Engineering Process of a CubeSat from the Perspective of Operations,” *INCOSE Int. Symp.*, vol. 23, no. 1, pp. 458–470, Jun. 2013.

[49] Pumpkin, Inc, “CubeSat Kit Motherboard (MB) Hardware Rev. D (specification sheet).” Pumpkin, Inc., San Francisco, CA, 2009.

[50] Pumpkin, Inc., “3D Models of the CubeSat Kit.” www.cubesatkit.com, 2018.

[51] Pumpkin, Inc., “CubeSat Kit^{TM} Structures web page.” .

[52] P. R. N. Childs, J. R. Greenwood, and C. A. Long, “Review of temperature measurement,” *Rev. Sci. Instrum.*, vol. 71, no. 8, pp. 2959–2978, Aug. 2000.

[53] Capgo Pty Ltd, “Introduction to Thermocouples web page.” 2013.

[54] Digital Equipment Corp., “Programmed Data Processor-l Handbook,” DIgitial Equipment Corporation, Maynard, Massachusetts, Oct. 1963.

[55] C. G. M. Bell J. Craig; Mcnamara, John E., *Computer engineering: a dec view of hardware systems design.* Bedford, Mass.: Digital Press, 2014.

[56] NXP Semiconductors, “I2C-bus specification and user manual rev.6.” [Online]. Available: http://www.nxp.com/documents/user_manual/UM10204.pdf.

[57] Renesas Electronics, “Serial Peripheral Interface (SPI) & Inter-IC (IC2) (SPI_I2C),” Renesas Electronics, AN0303011/Rev1.00, Sep. 2003.

[58] Texas Instruments, “KeyStone Architecture Serial Peripheral Interface (SPI) User Guide,” Texas Instruments, Dallas, TX, SPRUGP2A, Mar. 2012.

[59] F. Leens, “An introduction to I2C and SPI protocols,” *IEEE Instrum. Meas. Mag.*, vol. 12, no. 1, pp. 8–13, Feb. 2009.

[60] S. Corrigan, “Introduction to the Controller Area Network (CAN),” Texas Instruments Inc., Dallas, TX, Application Report SLOA101B, May 2016.

[61] M. Farsi, K. Ratcliff, and M. Barbosa, “An overview of Controller Area Network,” *Comput. Control Eng. J.*, vol. 10, no. 3, pp. 113–120, Jun. 1999.

[62] Robert Bosch GmbH, “CAN specification version 2.0.,” Robert Bosch GmbH, Stuttgart, Germany, Sep. 1991.

[63] U.S. Department of Defense, “Military Standard MIL-STD-1553B,” U.S. Government Printing Office: 1978-603-022/5017, 1978.

[64] D. R. Bracknell, “Introduction to the MIL-STD-1553B Serial Multiplex Data Bus,” Royal Aerospace Establishment, Farnborough, UK, AD-A202 511, Sep. 1988.

[65] Maxim Integrated Products, Inc., “Overview of 1-Wire Technology and Its Use: Tutuorial 1796 web page.” Maxim Integrated Products, Inc., 2008.

[66] D. Awtrey, K. Smith, and D. Lissiuk, “1-Wire Design Guide v1.0,” Springbok Digitronics.

[67] Maxim Integrated Products, Inc., “8-Channel 1-Wire Master DS2482-800 data sheet.” Maxim Integrated Products, Inc., Dec-2014.

[68] Maxim Integrated Products, Inc., “Tutorial 1796 : Overview of 1-Wire Technology and Its Use.” 2016.

[69] Digi International, “Demystifying 802.15.4 and ZigBee,” Minnetonka, MN, USA, White Paper, 2008.

[70] R. Alena, J. Ossenfort, T. Stone, and J. Baldwin, “Wireless Space Plug-and-Play Architecture (SPA-Z),” in *2014 IEEE Aerospace Conference*, 2014, pp. 1–17.

[71] G. Groeseneken, H. E. Maes, J. Van Houdt, and J. S. Witters, *Basics of non-volatile semiconductor memory devices*, vol. n Nonvolatile semiconductor memory technology, W.D.Brown and Joe E.Brewer, Ed. New York: IEEE Press, 1997.

[72] A. Nagamatsu *et al.*, "Space radiation damage to HDTV camera CCDs onboard the international space station," *Radiat. Meas.*, vol. 46, no. 2, pp. 205–212, Feb. 2011.

[73] G. Muller, T. Happ, M. Kund, G. Y. Lee, N. Nagel, and R. Sezi, "Status and outlook of emerging nonvolatile memory technologies," in *IEDM Technical Digest. IEEE International Electron Devices Meeting, 2004.*, San Francisco, CA, USA, 2004, pp. 567–570.

[74] L. Chua, "Memristor-The missing circuit element," *IEEE Trans. Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971.

[75] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, May 2008.

[76] Adesto Technologies, "Adesto Technologies RM24C256DS Advance Datasheet." Adesto Technologies.

[77] Texas Instruments, "Mixed Signal Microcontroller MSP430G2x53, MSP430G2x13," Texas Instruments, Dallas, TX, SLAS735J, May 2013.

[78] S. A. Asundi and N. G. Fitz-Coy, "CubeSat mission design based on a systems engineering approach," in *2013 IEEE Aerospace Conference*, Big Sky, MT, 2013, pp. 1–9.

[79] M. Swartwout, "The First One Hundred Cubesats: A Statistical Look," *J. Small Satell.*, vol. 2, no. 2, pp. 213–233, 2013.

[80] R. P. Welle, "The CubeSat Paradigm: An Evolutionary Approach to Satellite Design," presented at the 32nd Space Symposium, Technical Track, Colorado Springs, Colorado, 2016, pp. 1–12.

[81] Z. S. Decker, "A systems-engineering assessment of multiple CubeSat build approaches," Master of Science in Aeronautics and Astronautics, Massachusetts Institute of Technology, Boston, Massachusetts, USA, 2016.

[82] G. L. Skrobot, "ELaNa - Educational Launch of Nanosatellite Enhance Education Through Space Flight," NASA Kennedy Space Center, Logan, UT, KSC-2011-172, Aug. 2011.

[83] E. Mahoney, "NASA Announces Seventh Round of Candidates for CubeSat Space Missions." NASA, 18-Feb-2016.

[84] K. Hambleton and G. Jordan, “NASA Sends New Research on Orbital ATK Mission to Space Station, release 18-037.” NASA, 21-May-2018.

[85] Rowan Today staff, “Rowan Engineering successfully launches nanosatellite as part of NASA initiative.” Rowan University, 22-May-2018.

[86] V. Cox, “Orbital ATKs Cygnus Spacecraft Successfully Completes Rendezvous and Berthing with International Space Station.” Orbital ATK, 24-May-2018.

[87] NanoRacks, “NanoRacks Completes 14th CubeSat Deployment Mission from International Space Station.” NanoRacks LLC, Houston, Texas, 13-Jul-2018.

[88] B. Keeter, “ISS Daily Summary Report – 7/13/2018.” NASA, 13-Jul-2018.

[89] SNJ Today staff, “Rowan University Student-Built Nanosatellite Launched into Space by NASA.” SNJ Today, 13-Jul-2018.

[90] Maxim Integrated Products, Inc., “Single-Channel 1-Wire Master with Adjustable Timing and Sleep Mode DS2484 data sheet.” Maxim Integrated Products, Inc., Jul-2015.

[91] “SMALL SATELLITE CONFERENCE,” *SMALL SATELLITE CONFERENCE*. [Online]. Available: https://smallsat.org/.

Appendix A

Sample XEDS Template: Command & Data Handling (C&DH)

Section	Description	Bytes	Data Type (and Range)
Basic TEDS	Manufacturer ID	14	INT (17-16381)
	Model Number	15	INT (0-32767)
	Version Letter	5	INT (0-63)
	Version Number	24	INT (0-16777215)
C&DH	Template ID	8	Integer (value = 51)
	Serial Number	24	INT (0-16777215)
	System Type	2	INT (0-Flight, 1-Engineering, 2-Test)
	Ground Plane	1	BOOL (0-No, 1-Yes)
	System Voltage	8	INT (.01V)
	Max Power Draw	8	INT (.01W)
	Idle Power Draw	8	INT (.01W)
	I2C Header	8	INT (1 for Header 1, 2 for Header 2)
	I2C Pin	8	INT (1-104)
	Standoff Thread	15	ASCII
	Standoff 1 X coordinate	8	SIGNINT (.01mm)
	Standoff 1 Y coordinate	8	SIGNINT (.01mm)
	Standoff 2 X coordinate	8	SIGNINT (.01mm)
	Standoff 2 Y coordinate	8	SIGNINT (.01mm)
	Standoff 3 X coordinate	8	SIGNINT (.01mm)
	Standoff 3 Y coordinate	8	SIGNINT (.01mm)
	Standoff 4 X coordinate	8	SIGNINT (.01mm)
	Standoff 4 Y coordinate	8	SIGNINT (.01mm)
	Min Operating Temp	8	INT (.1 K)
	Max Operating Temp	8	INT (.1 K)
	Auxiliary I2C Header 1	8	INT (1 for Header 1, 2 for Header 2)
	Auxiliary I2C Pin 1	8	INT (1-104)
	Auxiliary I2C Header 2	8	INT (1 for Header 1, 2 for Header 2)
	Auxiliary I2C Pin 2	8	INT (1-104)
	Auxiliary I2C Header 3	8	INT (1 for Header 1, 2 for Header 2)
	Auxiliary I2C Pin 3	8	INT (1-104)
	Onboard Memory	8	INT (kB)
	Onboard Storage	8	INT (kB)
	Total	292	

Appendix B

Sample XEDS Template: Electrical Power System (EPS)

Section	Description	Bytes	Data Type (and Range)
Basic TEDS	Manufacturer ID	14	INT (17-16381)
	Model Number	15	INT (0-32767)
	Version Letter	5	INT (0-63)
	Version Number	24	INT (0-16777215)
EPS	Template ID	8	Integer (value = 52)
	Serial Number	24	INT (0-16777215)
	System Type	2	INT (0-Flight, 1-Engineering, 2-Test)
	Ground Plane	1	BOOL (0-No, 1-Yes)
	System Voltage	8	INT (.01V)
	Max Power Draw	8	INT (.01W)
	Idle Power Draw	8	INT (.01W)
	I2C Header	8	INT (1 for Header 1, 2 for Header 2)
	I2C Pin	8	INT (1-104)
	Standoff Thread	15	ASCII
	Standoff 1 X coordinate	8	SIGNINT (.01mm)
	Standoff 1 Y coordinate	8	SIGNINT (.01mm)
	Standoff 2 X coordinate	8	SIGNINT (.01mm)
	Standoff 2 Y coordinate	8	SIGNINT (.01mm)
	Standoff 3 X coordinate	8	SIGNINT (.01mm)
	Standoff 3 Y coordinate	8	SIGNINT (.01mm)
	Standoff 4 X coordinate	8	SIGNINT (.01mm)
	Standoff 4 Y coordinate	8	SIGNINT (.01mm)
	Min Operating Temp	8	INT (.1 K)
	Max Operating Temp	8	INT (.1 K)
	Number of Power rails	8	INT (1-6)
	Rail 1 Voltage	8	INT (.01V) (set voltage to 0 if not in use)
	Rail 1 Header	8	INT (1 for Header 1, 2 for Header 2)
	Rail 1 Pin	8	INT (1-104)
	Rail 2 Voltage	8	INT (.01V) (set voltage to 0 if not in use)
	Rail 2 Header	8	INT (1 for Header 1, 2 for Header 2)
	Rail 2 Pin	8	INT (1-104)

	Rail 3 Voltage	8	INT (.01V) (set voltage to 0 if not in use)
	Rail 3 Header	8	INT (1 for Header 1, 2 for Header 2)
	Rail 3 Pin	8	INT (1-104)
	Rail 4 Voltage	8	INT (.01V) (set voltage to 0 if not in use)
	Rail 4 Header	8	INT (1 for Header 1, 2 for Header 2)
	Rail 4 Pin	8	INT (1-104)
	Rail 5 Voltage	8	INT (.01V) (set voltage to 0 if not in use)
	Rail 5 Header	8	INT (1 for Header 1, 2 for Header 2)
	Rail 5 Pin	8	INT (1-104)
	Rail 6 Voltage	8	INT (.01V) (set voltage to 0 if not in use)
	Rail 6 Header	8	INT (1 for Header 1, 2 for Header 2)
	Rail 6 Pin	8	INT (1-104)
	Number of Ground Rails	8	INT (1-4)
	Ground Rail 1 Header	8	INT (1 for Header 1, 2 for Header 2)
	Ground Rail 1 Pin	8	INT (1-104)
	Ground Rail 2 Header	8	INT (1 for Header 1, 2 for Header 2)
	Ground Rail 2 Pin	8	INT (1-104)
	Ground Rail 3 Header	8	INT (1 for Header 1, 2 for Header 2)
	Ground Rail 3 Pin	8	INT (1-104)
	Ground Rail 4 Header	8	INT (1 for Header 1, 2 for Header 2)
	Ground Rail 4 Pin	8	INT (1-104)
	Total	452	

Appendix C

Sample XEDS Template: Radio Transceiver

Section	Description	Bytes	Data Type (and Range)
Basic TEDS	Manufacturer ID	14	INT (17-16381)
	Model Number	15	INT (0-32767)
	Version Letter	5	INT (0-63)
	Version Number	24	INT (0-16777215)
Radio	Template ID	8	Integer (value = 53)
	Serial Number	24	INT (0-16777215)
	System Type	2	INT (0-Flight, 1-Engineering, 2-Test)
	Ground Plane	1	BOOL (0-No, 1-Yes)
	System Voltage	8	INT (.01V)
	Max Power Draw	8	INT (.01W)
	Idle Power Draw	8	INT (.01W)
	I2C Header	8	INT (1 for Header 1, 2 for Header 2)
	I2C Pin	8	INT (1-104)
	Standoff Thread	15	ASCII
	Standoff 1 X coordinate	8	INT (.01mm)
	Standoff 1 Y coordinate	8	INT (.01mm)
	Standoff 2 X coordinate	8	INT (.01mm)
	Standoff 2 Y coordinate	8	INT (.01mm)
	Standoff 3 X coordinate	8	INT (.01mm)
	Standoff 3 Y coordinate	8	INT (.01mm)
	Standoff 4 X coordinate	8	INT (.01mm)
	Standoff 4 Y coordinate	8	INT (.01mm)
	Min Operating Temp	8	INT (.1 K)
	Max Operating Temp	8	INT (.1 K)
	Frequency	8	INT (.01MHz)
	Max Transmit Power	8	INT (.01W)
	Duplex Type	1	BOOL (0-Half, 1-Full)
	Max Radio Voltage	8	INT (.01V)
	Min Radio Voltage	8	INT (.01V)
	System Voltage	8	INT (.01V)
	Max Power Draw	8	INT (.01W)
	Idle Power Draw	8	INT (.01W)
	Packet Type	15	ASCII
	Impedance	8	INT (.01Ohms)
	Total	308	

Appendix D

Sample XEDS Template: Radio Beacon

Section	Description	Bytes	Data Type (and Range)
Basic TEDS	Manufacturer ID	14	INT (17-16381)
	Model Number	15	INT (0-32767)
	Version Letter	5	INT (0-63)
	Version Number	24	INT (0-16777215)
Beacon	Template ID	8	Integer (value = 53)
	Serial Number	24	INT (0-16777215)
	System Type	2	INT (0-Flight, 1-Engineering, 2-Test)
	Ground Plane	1	BOOL (0-No, 1-Yes)
	System Voltage	8	INT (.01V)
	Max Power Draw	8	INT (.01W)
	Idle Power Draw	8	INT (.01W)
	I2C Header	8	INT (1 for Header 1, 2 for Header 2)
	I2C Pin	8	INT (1-104)
	Standoff Thread	15	ASCII
	Standoff 1 X coordinate	8	INT (.01mm)
	Standoff 1 Y coordinate	8	INT (.01mm)
	Standoff 2 X coordinate	8	INT (.01mm)
	Standoff 2 Y coordinate	8	INT (.01mm)
	Standoff 3 X coordinate	8	INT (.01mm)
	Standoff 3 Y coordinate	8	INT (.01mm)
	Standoff 4 X coordinate	8	INT (.01mm)
	Standoff 4 Y coordinate	8	INT (.01mm)
	Min Operating Temp	8	INT (.1 K)
	Max Operating Temp	8	INT (.1 K)
	Frequency	8	INT (.01MHz)
	Max Transmit Power	8	INT (.01W)
	Duplex Type	1	BOOL (0-Half, 1-Full)
	Max Radio Voltage	8	INT (.01V)
	Min Radio Voltage	8	INT (.01V)
	System Voltage	8	INT (.01V)
	Max Power Draw	8	INT (.01W)
	Idle Power Draw	8	INT (.01W)
	Packet Type	15	ASCII
	Message	32	ASCII
	Impedance	8	INT (.01Ohms)

	Total	340	

Appendix E

Sample XEDS Template: Antenna

Section	Description	Bytes	Data Type (and Range)
Basic TEDS	Manufacturer ID	14	INT (17-16381)
	Model Number	15	INT (0-32767)
	Version Letter	5	INT (0-63)
	Version Number	24	INT (0-16777215)
Antenna	Template ID	8	Integer (value = 53)
	Serial Number	24	INT (0-16777215)
	System Type	2	INT (0-Flight, 1-Engineering, 2-Test)
	Ground Plane	1	BOOL (0-No, 1-Yes)
	System Voltage	8	INT (.01V)
	Max Power Draw	8	INT (.01W)
	Idle Power Draw	8	INT (.01W)
	I2C Header	8	INT (1 for Header 1, 2 for Header 2)
	I2C Pin	8	INT (1-104)
	Standoff Thread	15	ASCII
	Standoff 1 X coordinate	8	INT (.01mm)
	Standoff 1 Y coordinate	8	INT (.01mm)
	Standoff 2 X coordinate	8	INT (.01mm)
	Standoff 2 Y coordinate	8	INT (.01mm)
	Standoff 3 X coordinate	8	INT (.01mm)
	Standoff 3 Y coordinate	8	INT (.01mm)
	Standoff 4 X coordinate	8	INT (.01mm)
	Standoff 4 Y coordinate	8	INT (.01mm)
	Min Operating Temp	8	INT (.1 K)
	Max Operating Temp	8	INT (.1 K)
	Frequency	8	INT (.01MHz)
	Duplex Type	2	INT (0-Half, 1-Full, 2 - Quarter)
	Impedance	8	INT (.01Ohms)
	Deployment Voltage	8	INT (.01V)
	Deployment Current	8	INT (.01A)
	Deployment Pin Header	8	INT (1 for Header 1, 2 for Header 2)
	Deployment Pin Pin	8	INT (1-104)
	Total	278	

Appendix F

Sample XEDS Template: Attitude Control System (ACS)

Section	Description	Bytes	Data Type (and Range)
Basic TEDS	Manufacturer ID	14	INT (17-16381)
	Model Number	15	INT (0-32767)
	Version Letter	5	INT (0-63)
	Version Number	24	INT (0-16777215)
ACS	Template ID	8	Integer (value = 53)
	Serial Number	24	INT (0-16777215)
	System Type	2	INT (0-Flight, 1-Engineering, 2-Test)
	Ground Plane	1	BOOL (0-No, 1-Yes)
	System Voltage	8	INT (.01V)
	Max Power Draw	8	INT (.01W)
	Idle Power Draw	8	INT (.01W)
	I2C Header	8	INT (1 for Header 1, 2 for Header 2)
	I2C Pin	8	INT (1-104)
	Standoff Thread	15	ASCII
	Standoff 1 X coordinate	8	INT (.01mm)
	Standoff 1 Y coordinate	8	INT (.01mm)
	Standoff 2 X coordinate	8	INT (.01mm)
	Standoff 2 Y coordinate	8	INT (.01mm)
	Standoff 3 X coordinate	8	INT (.01mm)
	Standoff 3 Y coordinate	8	INT (.01mm)
	Standoff 4 X coordinate	8	INT (.01mm)
	Standoff 4 Y coordinate	8	INT (.01mm)
	Min Operating Temp	8	INT (.1 K)
	Max Operating Temp	8	INT (.1 K)
	Type	1	BOOL (0-passive, 1-Active)
	Polling Rate	8	INT (.01MHz)
	Control Packet Length	16	INT Bytes
	X-Axis Byte Length	8	INT Bytes
	Y-Axis Byte Length	8	INT Bytes
	Z-Axis Byte Length	8	INT Bytes
	Attitude Packet Length	16	INT Bytes
	X-Axis Read Byte Length	8	INT Bytes
	Y-Axis Read Byte Length	8	INT Bytes
	Z-Axis Read Byte Length	8	INT Bytes
	Total	212	

Appendix G

Sample XEDS Template: Photovoltaic Array

Section	Description	Bytes	Data Type (and Range)
Basic TEDS	Manufacturer ID	14	INT (17-16381)
	Model Number	15	INT (0-32767)
	Version Letter	5	INT (0-63)
	Version Number	24	INT (0-16777215)
Photovoltaics	Template ID	8	Integer (value = 57)
	Serial Number	24	INT (0-16777215)
	System Type	2	INT (0-Flight, 1-Engineering, 2-Test)
	Ground Plane	1	BOOL (0-No, 1-Yes)
	Min Operating Temp	8	INT (.1 K)
	Max Operating Temp	8	INT (.1 K)
	Face	8	INT (1-6 with face 1 being RBF face)
	Nominal Output Voltage		INT (.01V)
	Max Output Voltage		INT (.01V)
	Max Output Current		INT (.01A)
	Total	93	

Appendix H

Sample XEDS Template: Chassis

Section	Description	Bytes	Data Type (and Range)
Basic TEDS	Manufacturer ID	14	INT (17-16381)
	Model Number	15	INT (0-32767)
	Version Letter	5	INT (0-63)
	Version Number	24	INT (0-16777215)
Chassis	Template ID	8	Integer (value = 50)
	Serial Number	24	INT (0-16777215)
	System Type	2	INT (0-Flight, 1-Engineering, 2-Test)
	Ground Plane	1	BOOL (0-No, 1-Yes)
	Standoff Thread	15	ASCII
	Standoff 1 X coordinate	8	SIGNINT (.01mm)
	Standoff 1 Y coordinate	8	SIGNINT (.01mm)
	Standoff 2 X coordinate	8	SIGNINT (.01mm)
	Standoff 2 Y coordinate	8	SIGNINT (.01mm)
	Standoff 3 X coordinate	8	SIGNINT (.01mm)
	Standoff 3 Y coordinate	8	SIGNINT (.01mm)
	Standoff 4 X coordinate	8	SIGNINT (.01mm)
	Standoff 4 Y coordinate	8	SIGNINT (.01mm)
	Min Operating Temp	8	INT (.1 K)
	Max Operating Temp	8	INT (.1 K)
	Units of CubeSat	4	INT (1-6)
	Internal X dimension	8	INT (.01mm)
	Internal Y dimension	8	INT (.01mm)
	Internal Z dimension	8	INT (.01mm)
	External X dimension	8	INT (.01mm)
	External Y dimension	8	INT (.01mm)
	External Z dimension	8	INT (.01mm)
	Mass	8	INT (.01g)
	Number of Inhibits	2	INT (0-4)
	Number of Rail Type Inhibits	2	INT (0-4)
	Modulus of Elasticity	16	INT (N/mm2)(MPa)
	Coefficient of Thermal Expansion	16	INT 10^{-7}K−1
	Tensile Strength	16	INT (N/mm2)(MPa)
	Thermal Conductivity	8	INT W/m*K
	Volume resistivity	8	INT 0.1nOhm*m

	Total	172	

Appendix I

GitHub Repository for testing code

The URL for the GitHub Repository containing the sample code used for testing can be found at “https://github.com/freeride732/S_IOT”.