

University of Colorado
Department of Aerospace Engineering Sciences
Senior Projects - ASEN 4018

**Thinsat Operational Management
of Commands And Telemetry
(TOMCAT)**

Fall Final Report

Monday 16th December, 2019

Information

0.1. Project Customers

Name: Mark Werremeyer Email: Mark.R.Werremeyer@Raytheon.com
--

0.2. Group Members

Name: Austin Scheck Email: Austin.Scheck@colorado.edu	Name: Cole Kenny Email: Nicholas.Kenny@colorado.edu
Name: Grant Novota Email: Grant.Novota@colorado.edu	Name: Jake Schroeder Email: Jacob.Schroeder@colorado.edu
Name: Katie Steward Email: Katie.Steward@colorado.edu	Name: Lara Buri Email: labu2151@colorado.edu
Name: Lauren De Moudt Email: Lauren.DeMoudt@colorado.edu	Name: Lucas Perry Email: Lupe6259@colorado.edu
Name: Manuel Lindo Email: Manuel.Lindo@colorado.edu	Name: Matthew McCallum Email: Matthew.McCallum@colorado.edu
Name: Srikanth Venkataraman Email: srve1799@colorado.edu	Name: Trevor Weschler Email: Trevor.Weschler@colorado.edu

Contents

0.1	Project Customers	1
0.2	Group Members	1
1	Project Purpose	13
2	Project Objectives and Functional Requirements	13
2.1	Project Objectives and Levels of Success	13
2.2	System Operation and Deliverables	15
2.3	Functional Block Diagram	17
2.4	Project Functional Requirements	17
3	Conceptual Design	18
3.1	Key Design Options Considered	18
3.1.1	Communications Hardware	18
a	Software Defined Radio	18
i	LimeSDR Mini	19
ii	BladeRF 2.0 Micro xA4	19
iii	HackRF One	19
iv	USRP B200mini	20
b	Communication Band	20
c	Antenna	21
d	Half-Duplex Communication	22
3.1.2	Communications Software	22
a	SDR Software	22
b	Modulation Technique	23
3.1.3	Flight Software and On-Board Computer	23
a	Integrated Development Environment	23
i	Arduino IDE	23
ii	Eclipse C++ for Arduino	24
iii	KubOS	24
b	Microcontroller Unit	24
i	ATmega328	25
ii	ATSAMD21G18	25
iii	AT91SAM3X8E	25
iv	AM3358	25
3.1.4	Ground Software	26
3.1.5	Thermal	26
a	Thermal Control System	26
b	Thermal Protection System	27
3.1.6	Power	27
a	Power Supply Types	27
i	Alkaline	27
ii	Lithium Polymer	28
iii	Lithium Ion	28
3.1.7	Structures	28
3.1.8	Scientific Sensor	29
a	Imaging Sensor/Camera	29
b	Magnetometer	30

	c	Inertial Measurement Unit	31
	d	Environmental Sensor Suite	32
	e	Radiation Sensor/Dosimeter	33
3.2		Trade Study Process and Results	34
3.2.1		Communications Hardware: SDR	34
	a	Metrics and Weighting	34
	b	Metric Quantification	35
	c	Trade Results	36
3.2.2		Communications Software	37
3.2.3		Printed Circuit Board	37
3.2.4		Flight Software and On-Board Computer	37
	a	Metrics and Weighting	37
	b	Metric Quantification	37
	c	Trade Results	38
3.2.5		Flight Software IDE	38
	a	Metrics and Weighting	38
	b	Metric Quantification	39
	c	Trade Results	40
3.2.6		Ground Software	41
	a	KubOS	41
	b	Creating an original program	42
	c	COSMOS	42
3.2.7		Power	42
	i	Alkaline	42
	ii	Lithium Polymer	43
	iii	Lithium Ion	43
3.2.8		Structures	43
3.2.9		Science Sensor Type	43
	a	Metrics and Weighting	44
	b	Metric Quantification	45
	c	Trade Results	46
3.2.10		Science Sensor Model Selection	47
	a	Metrics and Weighting	49
	b	Metric Quantification	50
	c	Trade Results	50
3.3		Selection of Baseline Design	51
3.3.1		Communications Hardwre	51
	a	Software Defined Radio	51
3.3.2		Communications Software	51
3.3.3		Printed Circuit Board	51
3.3.4		Flight Software and On-Board Computer	51
3.3.5		Ground Software	52
3.3.6		Thermal	52
3.3.7		Power	53
3.3.8		Structures	54
3.3.9		Science Sensor	55
	a	Science Sensor Type	55
	b	Science Sensor Model	55

4	Requirements Development	55
4.1	Requirement 1.1	59
4.1.1	Requirement 1.1.1	59
4.2	Requirement 1.2	59
4.3	Requirement 1.3	59
4.3.1	Requirement 1.3.1	59
4.4	Requirement 1.4	59
4.5	Requirement 2.1	59
4.5.1	Requirement 2.1.1	59
4.6	Requirement 2.2	59
4.6.1	Requirement 2.2.1	60
4.6.2	Requirement 2.2.2	60
4.7	Requirement 2.3	60
4.8	Requirement 2.4	60
4.9	Requirement 2.5	60
4.10	Requirement 3.1	60
4.11	Requirement 3.2	60
4.12	Requirement 3.3	60
4.13	Requirement 3.4	60
4.13.1	Requirement 3.4.1	61
4.14	Requirement 4.1	61
4.14.1	Requirement 4.1.1	61
4.14.2	Requirement 4.1.2	61
4.14.3	Requirement 4.1.3	61
4.14.4	Requirement 4.1.4	61
4.14.5	Requirement 4.1.5	61
4.14.6	Requirement 4.1.6	61
4.15	Requirement 4.2	61
4.15.1	Requirement 4.2.1	61
4.15.2	Requirement 4.2.2	62
4.15.3	Requirement 4.2.3	62
4.16	Requirement 4.3	62
4.16.1	Requirement 4.3.1	62
4.16.2	Requirement 4.3.2	62
4.16.3	Requirement 4.3.3	62
4.17	Requirement 5.1	62
4.18	Requirement 5.2	62
4.19	Requirement 5.3	62
5	Detailed Design	62
5.1	Communications Hardware	62
5.1.1	Link Budget	63
5.1.2	RF Chain	68
5.1.3	Ground Station Placement	69
5.1.4	Communications Printed Circuit Board	70
5.2	Communications Software	71
5.2.1	Transmission	72
5.2.2	Reception	73
5.3	Printed Circuit Board	73
5.3.1	The Microcontroller Unit	74

5.3.2	Power and Ground	74
5.3.3	Clamping Circuit	77
5.3.4	Electrostatic Discharge Protection	79
5.3.5	Reset Circuitry	79
5.3.6	Clock Circuitry	80
5.3.7	Peripherals	80
5.3.8	USB	81
5.3.9	Non-Volatile Storage	82
5.3.10	Board Layout	83
5.4	Flight Software	83
5.5	Ground Software	86
5.6	Thermal	87
5.6.1	Preliminary 1D, Steady-State Model	87
a	Model Assumptions	88
b	Method and Theory	88
c	Results	89
5.6.2	Preliminary Testing	91
a	Model Assumptions	91
b	Method and Experimental Setup	91
c	Results	92
5.7	Power	93
5.7.1	Power Budget	93
5.7.2	Margin and Built in Capacity	94
5.7.3	Power Distribution	96
a	Manufacturer Provided PCB	97
b	Payload Board Power Regulation	97
5.7.4	Battery Selection	97
5.7.5	Ground Station Power	97
5.8	Structures	98
5.8.1	Requirements	100
5.8.2	Modeling	101
5.9	Science	101
5.9.1	Requirements	101
5.9.2	Physical Integration	102
5.9.3	Mission Selection	103
5.9.4	Coding Algorithm	103
6	Verification and Validation	104
6.1	Communications Hardware	104
6.1.1	Verification	104
a	Transmission Bench Test	104
b	Reception Bench Test	105
c	Short Range Wireless Test	106
6.1.2	Validation	107
a	Long Range Wireless Test	107
6.2	Communications Software	108
6.2.1	Verification	108
a	SDR-SDR bench test with attenuator	108
b	GSW integration test	108
6.2.2	Validation	109

a	Running it on the OBC	109
6.3	Payload Board	109
6.3.1	Verification	109
a	Visually Inspect Payload Board	109
b	Solder Remaining Components	109
c	Flash Board with KubOS	109
d	Use KubOS to Control I/O Programmatically	110
6.3.2	Validation	110
6.4	Flight Software	111
6.4.1	Verification	111
6.4.2	Validation	112
6.5	Ground Software	112
6.5.1	Verification	112
a	COSMOS Command Sending Test	112
b	COSMOS Telemetry Visualization Test	113
c	COSMOS Telemetry Logging Test	114
d	COSMOS Integration Test	115
6.5.2	Validation	115
6.6	Thermal	115
6.6.1	Verification	115
a	Cold Test #2	115
6.6.2	Validation	116
a	Full System Functional Cold Test (Cold Test #3)	116
b	Balloon Flight Validation	117
6.7	Power	117
6.7.1	Verification	117
a	Charge/Discharge Test	117
b	Peak Power Test	117
c	Full System Thermal Test (Cold Test #3)	118
6.7.2	Validation	118
6.8	Structures	118
6.8.1	Verification	119
6.8.2	Validation	119
6.9	Science	120
6.9.1	Verification	120
6.9.2	Validation	120
7	Risk Assessment and Mitigation	121
7.1	Critical Technical Project Risks	121
7.2	Critical Risk Descriptions and Mitigations	122
7.2.1	Balloon Drift	122
7.2.2	RF Noise	123
7.2.3	Thermal	123
7.2.4	Signal Integrity	123
7.2.5	PCB Failure	124
7.2.6	Heat	124
7.3	Critical Logistical Project Risks	124
7.4	Risk Tracking Methodology	124

8	Project Planning	125
8.1	Organizational Chart	125
8.2	Work Breakdown Structure	126
8.3	Work Plan - Gantt Chart	127
8.3.1	Critical Path Analysis	131
8.4	Cost Plan	131
9	Individual Report Contributions	134

List of Figures

1	Raytheon's anticipated first mission with the ThinSat compared with TOMCAT's portion of the project development	15
2	TOMCAT mission concept of operations	16
3	TOMCAT On-board and Ground System Functional Block Diagram	17
4	Onboard Antenna Choice - ANT-916-CW-RAH (From Linx Technologies)	21
5	Ground Antenna Choice	22
6	1T Configuration ThinSat that will be purchased from NearSpace Launch Inc.	29
7	PlanetLabs visible spectrum satellite concept [1]	30
8	Magnetic Field Fluctuations with Solar Wind [2]	31
9	Sample FBD of a state estimator using an IMU [3]	31
10	Temperature Profile Throughout the Atmosphere [4]	32
11	Regener-Pfotzer Maximum [5]	33
12	Insulating Materials at Low Pressures [6]	53
13	CAD model of the fully assembled 2T ThinSat	54
14	Required Eb/No SMAD (SMAD)	64
15	Expected Atmospheric Attenuation (SMAD)	66
16	On-board Receiver RF Chain	68
17	Ground Receiver RF Chain	68
18	Ground Station Offset Concept Visualization	69
19	Required Ground Station Offset	70
20	Custom Communications PCB Schematic	71
21	Full GNU Radio simulation block diagram	72
22	Binary data before and after modulation process	72
23	FFT Spectrum before and after equalizer	73
24	OSD335x C-Sip Functional Diagram	74
25	OSD335x Power Input PCB Schematic	75
26	OSD335x Ground Connections PCB Schematic	76
27	Power Management Connections	77
28	Clamping Circuit	78
29	Clamping Circuit PCB Schematic	79
30	Reset Circuitry PCB Schematic	80
31	Clock Circuitry PCB Schematic	80
32	IMU PCB Schematic	81
33	JTAG PCB Schematic	81
34	USB Ports PCB Schematic	82
35	Micro-SD PCB Schematic	82
36	Custom PCB Board Layout Design	83
37	Flight Software Framework's Freebody Diagram	85
38	Functional Block Diagram for Ground Software	87
39	1D, Steady-State Model With No Insulation	88
40	1D, Steady-State Model With Insulation	89
41	1D, Steady-State Model Results With No Insulation	89
42	Payload Area Results With Aerogel Insulation	90
43	SDR Results with Aerogel Insulation	90
44	Payload Results with Polystyrene Insulation	90
45	SDR Results with Polystyrene Insulation	90
46	Test Setup For Preliminary Thermal Testing	92
47	Actual Test Setup	92

48	Test Results For Battery Area	93
49	Test Results For Payload Area	93
50	Test Results For SDR Area	93
51	Voltage vs Depth of Discharge at Varying Discharge Rates [7]	95
52	Voltage vs Depth of Discharge at Varying Temperatures [8]	95
53	ThinSat Power System FBD	96
54	Battery with PCB CAD from Manufacturer	97
55	Battery within Payload	97
56	ThinSat Power System FBD	98
57	Exploded view of the CAD model with major components labeled	99
58	CAD model of the ThinSat with the tube used to attach to the Gateway to Space balloon flight	100
59	Xsens MT-i 3 AHRS	101
60	MTi 1-Series Pin Diagram	102
61	Payload Board Functional Block Diagram	103
62	ASEN 1400 Flight Data	104
63	Transmission Bench Test Diagram	105
64	Reception Bench Test Diagram	106
65	Communications System Short Range Test FBD	107
66	Communications System Long Range Test FBD	108
67	Example of Control I/O Programmatically	110
68	Testing Interface between instrument and GNU Radio	111
69	Health and Status / Beacon Test	111
70	Query/Mutation Schema for Testing	112
71	Results of the Command Sending Test in COSMOS	113
72	Results of the Telemetry Visualization Test in COSMOS	114
73	Results of the Telemetry Logging Test	115
74	Charge/Discharge Test	117
75	Full System Test	118
76	Drop test will be conducted in this orientation and from 13.5 feet	119
77	Critical project risk table	121
78	Critical project risk matrix	122
79	Critical project risk matrix prior to mitigation	122
80	Risk tracking tool created by TOMCAT	125
81	TOMCAT Org Chart	126
82	TOMCAT Work Breakdown Structure - Updated 12/14/19	127
83	TOMCAT Gantt Chart, Page 1	128
84	TOMCAT Gantt Chart, Page 2	129
85	TOMCAT Gantt Chart, Page 3	130
86	Tentative Cost Plan - Pie Chart	132
87	Tentative Cost Plan - Subsystem Overview	132
88	Tentative Cost Plan - Detailed Overview	133

List of Tables

5	Trade Topics and Weights for the Software Defined Radio	35
6	Explanation of Analytical Scale for the Software Defined Radio	36
7	Trade Study Results for the Software Defined Radio	36
8	Trade Topics and Weights for the MCU	37
9	Explanation of Analytical Scale for the MCU	38
10	Trade Study Results for MCU Selection	38
11	Trade Topics for the IDE	39
12	Explanation of weights for IDE	40
13	Trade Study Results for the Flight Software IDE	41
14	Pros and Cons of KubOS as a ground software	41
15	Pros and Cons of creating an original ground software	42
16	Pros and Cons of COSMOS as a ground software	42
17	Pros and Cons of Alkaline Batteries	42
18	Pros and Cons of Lithium Polymer Batteries	43
19	Pros and Cons of Lithium Ion Batteries	43
20	Trade Metrics for the Science Sensor Selection	44
22	Weights for Science Sensor Selection	46
23	Trade Study Results for Science Sensor Selection	47
24	Characteristics of Each IMU Sensor	47
25	Trade Metrics for the Sensor Model Selection	49
26	Weights for Sensor Model Selection	50
27	Trade Study Results for Sensor Model Selection	50
28	TOMCAT Link Budget: General Parameters	63
29	TOMCAT Link Budget: Data Parameters	63
30	TOMCAT Link Budget: Noise Parameters	65
31	TOMCAT Link Budget: Receiver and Propagation Parameters	65
32	TOMCAT Link Budget: Transmitter Parameters	66
33	TOMCAT Link Budget: Summary	67
34	KubOS Basic Services	84
35	TOMCAT's Mission Applications	85
36	Component Temperature Requirements	87
37	ThinSat Power Budget	94
38	Effect of Temperature on Battery Capacity [8]	95
39	Calculation of Additional Battery Capacity Accommodations	96
40	Ground Station Power Budget	98

Nomenclature

ADC	Analog to Digital Converter
AVR	Automatic Voltage Regulator
BOM	Bill of Materials
C&DH	Command & Data Handling
CC	Command Control
CCD	Charge-Coupled Decide
CCSDS	Consultative Committee for Space Data Systems
CMOS	Complementary Metal-Oxide-Semiconductor
COMM	Communication
CONOPS	Concept of Operations
CRaTER	Cosmic Ray Telescope for the Effects of Radiation
CU	University of Colorado
DDR	Double Data Rate
EEPROM	Electrically Erasable Programmable Read-Only Memory
ELSA	Europa Lander for Science Acquisition
eMMC	Embedded Multi-Media Card
ESD	Electro-Static Discharge
FAA	Federal Aviation Administration
FBD	Functional Block Diagram
FCC	Federal Communications Commission
FSW	Flight Software
GPIO	General-Purpose Input/Output
GSW	Ground Software
I/O	Input/Output
I2C	I-Squared-C (Serial Protocol)
ICD	Interface Control Document
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit
IR	Infrared

ISM	Industrial, Scientific, and Medical
LASP	Laboratory for Atmospheric and Space Physics
LEO	Low Earth Orbit
MCU	Micro Controller Unit
MEMS	Micro-Electro-Mechanical Systems
MLI	Multi-Layer Insulation
NSL	NearSpace Launch Inc.
OBC	On-Board Computer
OS	Operating System
PCB	Printed Circuit Board
RAVEN	Rover and Air Visual Environment Navigation
RF	Radio Frequency
SDR	Software Defined Radio
SMA	SubMiniature version A
STEM	Science, Technology, Engineering, and Mathematics
sUAS	Small Unmanned Aerial System
TOMCAT	Thinsat Operational Management of Commands And Telemetry
USB	Universal Serial Bus
USRP	Universal Software Radio Peripheral
UV	Ultraviolet
WBS	Work Breakdown Structure

1. Project Purpose

Katie Steward

Expenditures to access space are slowly decreasing; however, development, testing, launch, and operating costs of spacecraft and small satellites still remain a huge barrier to entry for anyone looking to enter into the spaceflight industry. While these expenses make it harder for businesses to conduct activities in space, it also makes it difficult for primary schools and STEM organizations to expose their students to space-based research. To combat this, Virginia Space helped pioneer the ThinSat Program, which provides an opportunity for educational institutions to design lightweight, low-cost payloads for picosatellites.[9][10]

The ThinSat is a 1/7th slice of a 1U cubesat, with a fully functioning bus and payload area. The TOMCAT project expands on the ThinSat program, expanding its communication capabilities. By using a software defined radio (SDR), TOMCAT allows the satellite to communicate at any frequency that a company could want, while maintaining the small size of the ThinSat frame. The TOMCAT project is a 2/7th slice of a 1U cubesat, to provide a better, more versatile communication system and more battery power. By conforming to the ThinSat shape, the TOMCAT satellite can be launched into low Earth orbit for a very low price relative to other space launches.

The TOMCAT project can be expanded into infinite applications. Raytheon is specifically interested in two after the project is complete: an on-orbit ground station test asset and a spacecraft to characterize the upper atmosphere as the satellite burns up.

2. Project Objectives and Functional Requirements

Katie Steward

2.1. Project Objectives and Levels of Success

Raytheon has tasked the TOMCAT team to develop and test viable options for PicoSat telemetry and lead the way in developing and testing secondary communications on the ThinSat. Not only would this provide a redundancy to the satellite, an often-desired trait in space missions, but it would also allow for a low-cost option for testing ground-based communication. Following the standards laid out by the Consultative Committee for Space Data Systems (CCSDS)[11], the team will integrate command and telemetry, C&DH hardware and software first onto a FlatSat, then onto a ThinSat bus for verification and validation in the lab and on a high-altitude balloon. Communications will first be established during a bench test of two Software Defined Radios. These radios will then be modified to work at long range. The high-altitude balloon flight will act as a final validation test, in which the payload will be fully integrated with the ThinSat bus and the COMM system will be fully operational. Tests leading up to the balloon flight will also allow for the verification and validation of the wireless COMM system. The ThinSat payload will include a scientific sensor, which will take data and generate telemetry that will be sent through the COMM system to the COMM system on the ground. This will fulfill Raytheon's desire to have some kind of science data measured on the payload that can be sent as telemetry to further verify the capability of the COMM system.

The following table reflects the specific objectives for each of these three cornerstone components. Within each of these, objectives are subdivided into three levels of mission success, with Level I representing marginal, bottom-line performance, and Level III expressing high-tier achievement, implying the fulfillment of both Level I and II objectives in their completion.

Flight Software		
	Command	Telemetry

Level 1	Flight software should receive, parse, and execute commands from the SDR. Flight software should recognize and decode CCSDS-formatted command packets.	Flight software should generate state of payload health telemetry. Flight software should packetize telemetry it generates or receives from the science sensors in CCSDS format. Flight software should send the packetized telemetry to the SDR for downlink.
Level 2	Flight software should generate command counters to verify commands it received. Flight software shall receive and execute payload commands from the on-board bus computer. Flight software should generate command counters for commands it has sent to the science suite.	Flight software should store state of health and science telemetry that is generated so that it can be compared to the telemetry received on the ground.
Level 3	Flight software should generate command counters to verify commands have been recognized as acceptable commands for payload execution.	
Communications		
Level 1	The SDR should be able to receive commands from another SDR over a serial connection. The SDR should be able to transmit telemetry to another SDR over a serial connection.	
Level 2	The SDR should be able to receive commands from another SDR over a wireless connection at a range greater than 5m. The SDR should be able to transmit telemetry to another SDR over a wireless connection at a range greater than 5m.	
Level 3	The ground SDR should be able to receive commands from the flight SDR over a wireless connection during the balloon flight, with a range of 30km. The ground SDR should be able to transmit telemetry to the flight SDR over a wireless connection during the balloon flight, with a range of 30km.	
Science Sensor Suite		
Level 1	Sensor suite should generate science telemetry to send to the payload computer.	
Level 2	The sensor suite should execute commands received from the payload computer to take science data. The sensor suite should return science data to the payload computer. The sensor suite should generate and send state of health telemetry to the payload computer.	
Level 3		
Payload Structure		
Level 1	The payload should mount to the designated payload area on the ThinSat bus.	

Level 2	
Level 3	The payload should survive nominal (9 m/s) impacts from balloon launch and recovery.
	The SDR should be mechanically mounted to the bus for balloon flight.

2.2. System Operation and Deliverables

Raytheon would like to use TOMCAT as a relatively inexpensive on-orbit test asset that can verify the functionality of a ground station. TOMCAT will be launched in the low-earth orbit (LEO) where it can communicate with Raytheon's chosen ground station. TOMCAT will be focusing on beginning the development of long-range communications by using a software defined radio configured for the range of a balloon launch. Raytheon's mission is shown below on the right, and TOMCAT's mission in comparison is shown below on the left.

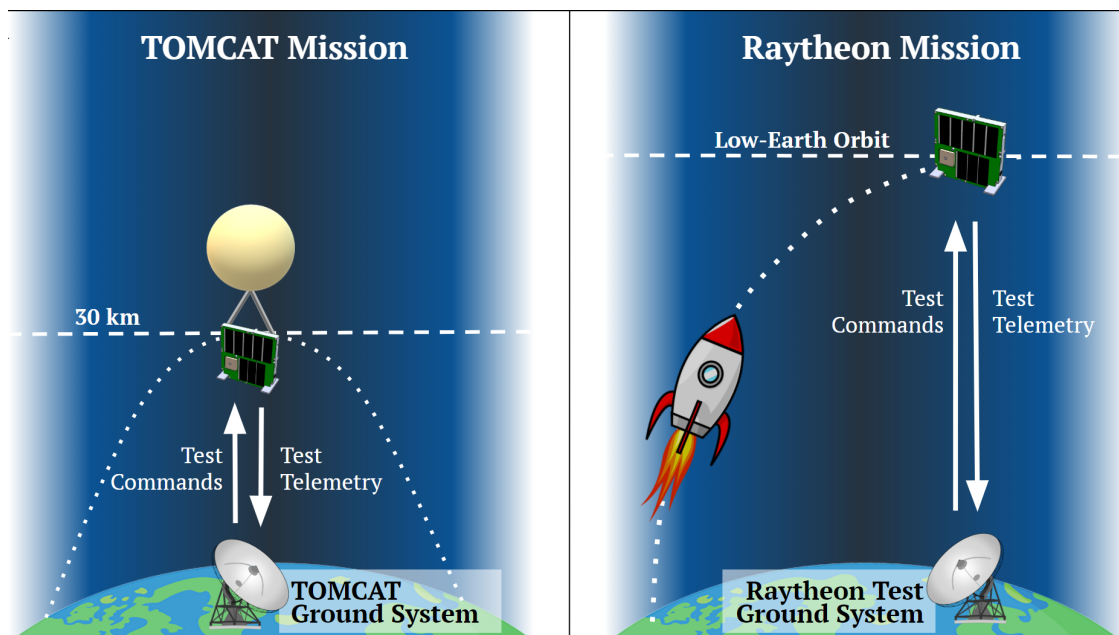


Figure 1. Raytheon's anticipated first mission with the ThinSat compared with TOMCAT's portion of the project development

A more detailed visual of the TOMCAT's end goal balloon flight is shown in the CONOPS below. TOMCAT will be launching on along on a balloon string hosted by the University of Colorado at Boulder's Gateway to Space class. The ground station will start by communicating with the ThinSat while it is on the ground so a two-way communication link can be verified. On-board the ThinSat, the ground commands will travel from software defined radio to the microcontroller to the inertial measurement unit, which will execute the command and send rate telemetry in return. Throughout the duration of the launch the ground system will be in constant communication with the ThinSat. At the peak altitude of the flight (around 30km), the balloon will automatically burst due to pressure, and a parachute will automatically deploy. During the descent, the ground station will continue to maintain two-way communication. The ThinSat is expected to land at a speed of around 9 m/s, where after it can be collected by the TOMCAT team.

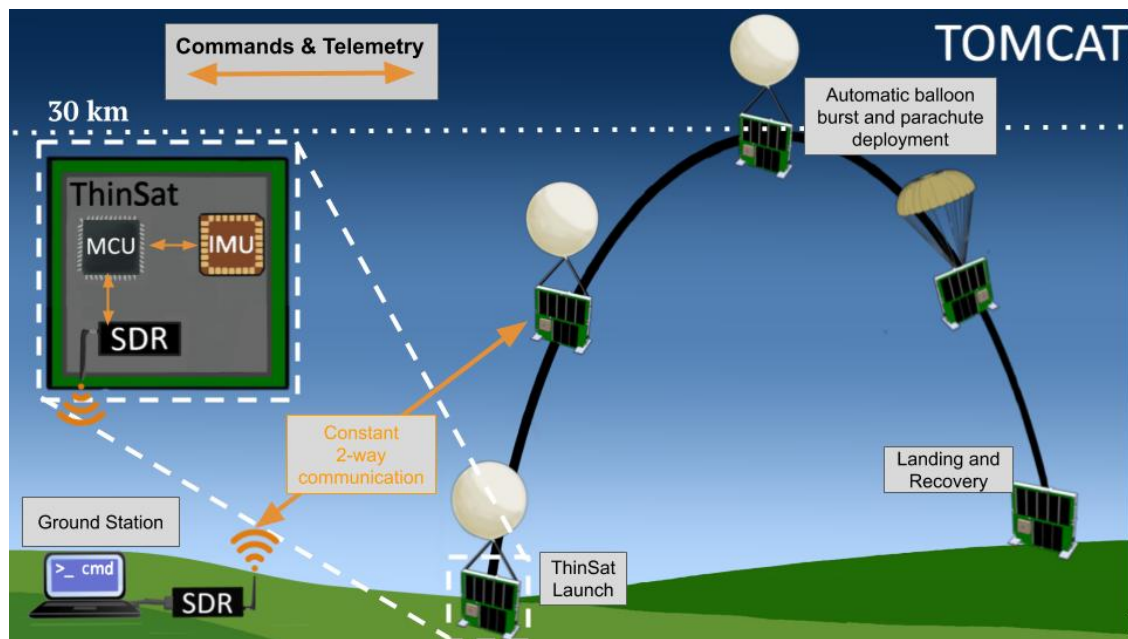


Figure 2. TOMCAT mission concept of operations

At the end of the project, the TOMCAT team will give Raytheon two systems – the TOMCAT ThinSat system and the TOMCAT ground system. The ThinSat system will consist of one purchased frame "slice", and one custom manufactured frame "slice", and all required internal hardware: communications hardware (radio, amplifiers), custom printed payload circuit board (containing an inertial measurement unit and an AM335x microcontroller), insulation, and battery. The software provided for this system will consist of custom flight software based on the framework of KubOS flight software and custom signal processing software based on the framework of GNURadio software. The ground system hardware will consist of a Yagi antenna, software defined radio, amplifiers, and a USB connection to a computer. The ground system software will consist of the same custom signal processing software as the ThinSat system package, as well as custom ground software built off the framework of COSMOS.

2.3. Functional Block Diagram

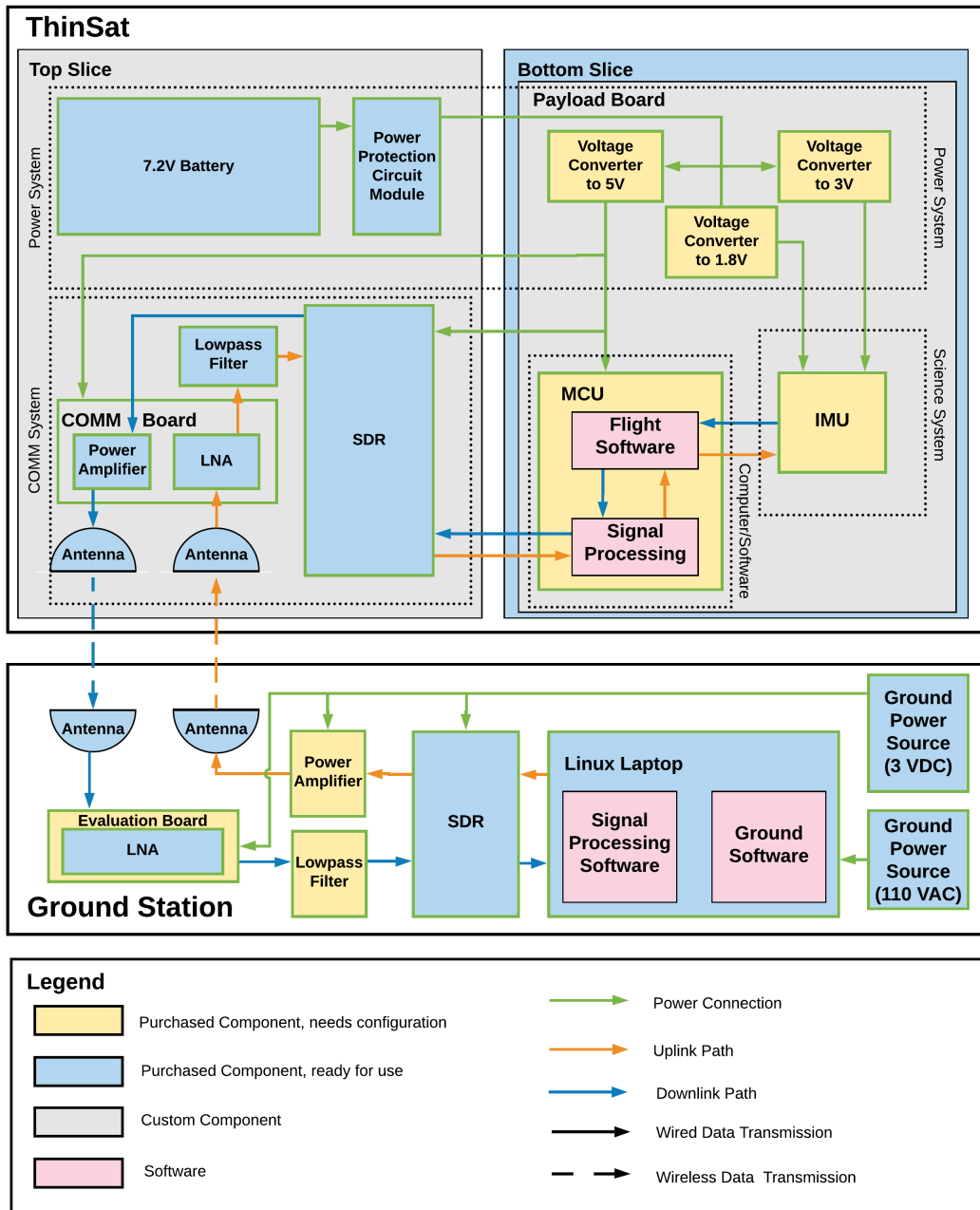


Figure 3. TOMCAT On-board and Ground System Functional Block Diagram

2.4. Project Functional Requirements

In early stages of the project, the TOMCAT team discussed the goals of the project with the Raytheon sponsor. In an effort to design the communication satellite that Raytheon is looking for, the team introduced the following high level functional requirements for the project to succeed.

1. The on-board payload shall interface with the ThinSat bus according to the ThinSat Interface Control Document (ICD)

2. The on-board payload shall be able to communicate with the ground system on a balloon-flight-like range test
3. The payload shall survive and operate in a balloon-flight-like environment for the duration of the proposed balloon flight (1.5 hrs)
4. The on-board electronics radio software, and flight software shall perform all software and hardware tasks necessary for communication on a balloon flight

Since Raytheon has requested that the TOMCAT project will build off of the base ThinSat created by NearSpace and Twigg's Space lab, the project needs to be designed to interface with the base. At a high level, this means that the electronics need to be connected, and all TOMCAT components need to fit within a ThinSat-shaped frame.

In order for the mission to be successful, the satellite needs to be able to communicate in both uplink and downlink. Raytheon has requested that the communications be designed to support a balloon flight, so the communications system needs to be designed to support the maximum range of the proposed balloon flight.

The payload is expected to go to space in the future, so in order to demonstrate a proof of concept that the satellite can withstand near-space conditions, the satellite needs to be designed to survive a balloon flight. Survivability includes working through the thermal and pressure conditions at high-altitude, as well as surviving the impact at the end of the balloon flight.

Other than the communications system, all other systems work to support the functionality or verification of the communication system. Thus, all other systems need to work together to perform all the tasks that are necessary to make the communications system work during flight. This includes everything from basic power requirements to signal processing software for the received signal.

All functional requirements give the design goals of the project, and give the basis for the design requirements developed for the project. The specific design requirements and how they are planning to be met are discussed in detail in a later section.

3. Conceptual Design

3.1. Key Design Options Considered

3.1.1. Communications Hardware

Lauren DeMoudt, Trevor Weschler

a. Software Defined Radio The customer has specified that the ThinSat shall use a software defined radio (SDR) for communications with the payload. Software defined radios differ from traditional radios in that the radio control typically associated with the hardware is done via software. This includes frequency adjustment, filtering, etc. SDRs are becoming more popular because of the flexibility they provide over standard radios is highly desirable. SDRs allow the user to change transmission and receiving parameters with the use of software rather than modifying the hardware. For TOMCAT, the large amount of SDR options were narrowed down to four radios that are suitable for the mission. For future missions, the entire SDR subsystem will ideally fit onto a dedicated ThinSat slice. While this is not a requirement for TOMCAT, this desire does provide criteria to narrow down suitable options from the large amount of SDRs available. The main criteria used to select these radios were minimization of size, mass, and power draw. On a ThinSat, access to all of these criteria is constrained due to the small size of the spacecraft payload area.

i. LimeSDR Mini The first suitable option is the LimeSDR mini available from SparkFun. This SDR is the cheapest available option at \$189.95 while having the lowest mass and size. Where this option separates itself is its large bandwidth capability at 96 MHz. This will allow the entire 902-928 MHz range to be usable if necessary for data transmission.

This SDR has a lower max receiver gain of the SDR options. Along with an average max transmission gain will most likely mean that a separate RF amplifier will be necessary to ensure the communications system works at the distances expected on balloon flight. Additionally, the LimeSDR mini doesn't have much MATLAB support. Since the TOMCAT team is experienced in using MATLAB, being able to interface with the ground SDR with MATLAB is preferable. This will cut down in overall development time for the communication system since unfamiliar software will not have to be used to interface with the ground SDR. On the flight SDR side, this radio has GNU Radio support. GNU Radio is the software that will be used to interface with the flight SDR.

Additionally, the LimeSDR is rated to operate down to -40°C, but is normally operated at 25°C. The lowest temperature TOMCAT is expected to see is -56°C. This means that there will have to be some thermal design in order to ensure the radio continues to work throughout the balloon flight. The extent of this thermal control will depend on modeling the thermal characteristics of the radio.

Pros	Cons
Cheapest viable option	Low receiving gain
Largest bandwidth capability	Average transmission gain
Small size	Limited MATLAB support
Low mass	Typical 25°C operating temperature
Down to -40°C operating temperature	

ii. BladeRF 2.0 Micro xA4 The second suitable option is the BladeRF 2.0 Micro xA4. This SDR has high transmission gain and reasonable reception gain. While investment in a RF amplifier will probably still be necessary, it may be possible to get a cheaper amplifier since the signal will already be amplified. In addition, the BladeRF has more MATLAB compatibility than the LimeSDR. 3rd parties have been developing MATLAB software for this radio since it was released around a year ago. With additional GNU radio software, the team will be less pressed to develop custom software to support the BladeRF's operations.

This radio has the second highest mass and size of the options up for consideration. As future iterations of this project will launch into space, both of these characteristics need to be minimized. The BladeRF is also the 2nd highest cost option. At \$480.00, purchasing two of these radios will consume almost 20% of the total budget allocated for the project. When adding in any external elements such as antennas and RF amplifiers, this option could become a significant portion of the total budget.

Finally, the BladeRF only has a 0°C minimum operating temperature. It requires thermal protection in order to ensure communications can be maintained over the entire balloon flight. This SDR needs more thermal control than the LimeSDR or other radios that are rated to operate to lower temperatures.

Pros	Cons
High transmission gain	Large mass
MATLAB compatibility	Large size
Decent receiver gain	High cost
	0°C operating minimum

iii. HackRF One The third suitable option is the HackRF One. This SDR has the highest receiver gain than any of the other options at 94 dB. Additionally, an entire 3rd party interface has been developed for MATLAB to control the HackRF One. This would make software development for the ground SDR

must simpler. This combined with the GNU support for the flight SDR would streamline the software development process.

Additionally, the HackRF One has a lower power draw than any of the other SDR option. A USB 2.0 interface is used instead of a USB 3.0. The total power consumed is 2.5 W generated by using 500 mA at 5V. The Hack RF One is relatively cheap at \$299.95. This could free up budget for any necessary external elements such as antennas and RF amplifiers.

As the "tinkerers" radio, the HackRF One also has several drawbacks. Despite the high receiver gain, the transmission gain and bandwidth is the lowest of all the options. This reduces the maximum achievable data rate. Additionally, the radio has the highest mass and largest size of the available options.

Perhaps the most concerning characteristic of the HackRF One is that it has no documentation on operating temperature. If selected, much more modeling would be necessary to determine the minimum operation temperature and design a suitable thermal control system. Such a process could complicate the thermal control system which will most likely be essential for other components of the ThinSat.

Pros	Cons
High receiver gain	Low transmission gain
3rd Party MATLAB interface	Narrow bandwidth
Low power draw	Undocumented operating temperature range
Cheap cost	High mass

iv. USRP B200mini The final suitable option is the USRP B200mini. This SDR is similar to the LimeSDR in terms of minimizing total footprint. It has the second smallest size and mass of the options. Additionally, Mathworks has developed USRP support in the MATLAB Communications Toolbox. It provides radio-in-the-loop design and modeling, allowing the user to design and verify SDR systems. Coupled with GNU radio support, the software support for the B200 mini is unmatched by any of the other options. In addition to low mass, small size, and great software support, this radio also has a high transmission gain at 90 dB.

One major downside is that the SDR costs \$759.00 per radio. In addition, the need for amplifiers on both the flight and ground SDRs will increase the overall setup cost further. Finally, the B200mini does not meet the operating temperature requirements out of the box. At -20°C, a thermal control system will need to be developed to keep the SDR operational.

Pros	Cons
High transmission gain	High cost
Mathworks MATLAB and GNU radio support	Low receiver gain
Small size	-20°C operating minimum
Low mass	

b. Communication Band When determining which frequency band to communicate within, the main concerns of the team were accessibility and the effects on communication capabilities between the ground and onboard systems. The initial concern of the team was that many frequency bands require the user to possess some kind of license in order to broadcast within the band. Due to the complexity often involved with obtaining a radio license, the team determined that it would be best to operate within a band designated for amateur radio services that does not require a license. The team consulted the customer and they recommended transmitting in the ISM bands. The ISM bands are designated for industrial, medical, and scientific applications. The bands designated for amateur, unlicensed services considered by the team were the 2.4-2.5 GHz band and the 902-928 MHz band. The higher frequency band allowed for the team to transmit across a wider bandwidth. This would allow for more transmitted data at once. However, transmitting at higher frequencies means that the signal would be less powerful when it reached the onboard system and

thus more difficult to receive and decode. Due to the ISM bands being open for amateur radio services, the noise within these bands is unpredictable. Since the amount of telemetry the team intends to downlink will be small enough that the narrower bandwidth will still be sufficient, the team determined that the bigger concern was the ability to deal with unanticipated noise. Due to this, the team chose to transmit within the lower frequency band because it results in a larger link margin and thus more room for unexpected noise, despite the sacrificed bandwidth. The link margin and budget will be discussed in more detail later in this document.

c. Antenna After conducting research on antenna functionality and design available for SDR applications, the team was able to fairly quickly determine what was desirable without the need to do a full trade study. The most important factor was the frequency range of the antenna. For this project, the team intends to utilize the ISM band frequencies allocated for amateur satellite services. As discussed above, the chosen frequency band encompasses 902 MHz through 928 MHz. For the onboard system, the team noted a preference for omni-directional antennas. Despite the diminished gain, and thus decreased range, the capability to receive and radiate energy in all directions was deemed more important due to the lack of on-board pointing capabilities for the ThinSat. In addition to this, the team needed to minimize the size and weight for the onboard antennas in order to minimize the mass contribution to the system and allow the antennas to fit within the ThinSat dimensions. Due to these constraints, the team determined that a quarter-wave monopole antenna would work best for this application. With quarter-wave antennas, the antenna is a fourth of the length of the wavelength of the radio waves. This allows for the antenna to be shorter than other options that broadcast at the same frequency, thus making this model ideal for this application.

After taking these design preferences into consideration, the specific antenna model chosen for the onboard system for both the receiving and transmitting antennas was the ANT-916-CW-RAH, a quarter-wave monopole antenna, from Linx Technologies shown in Figure 4. This antenna has a length of 47 mm which is small enough to fit within the design space. It possesses an omni-directional radiation pattern as desired and a center frequency in the middle of the ISM band chosen for operation. While there were other antenna choices with similar radiation patterns and roughly the same size, this antenna was primarily chosen due to its relatively high gain of 2.2dBi and the lower bound of the operational temperature extending to -40 degrees Celsius.

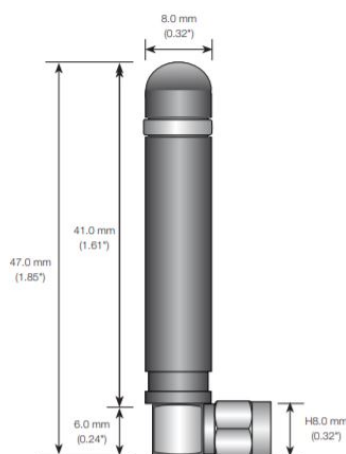


Figure 4. Onboard Antenna Choice - ANT-916-CW-RAH (From Linx Technologies)

The requirements for the antennas for the ground system varied slightly from those of the onboard system. The team initially considered an omni-directional antenna on the ground as well but after noting that there would never be a need for the ground antenna radiate in all directions, the team turned the research efforts towards antennas with higher gain but a more restricted radiation pattern. In addition to this, the mass

and size constraints placed on the onboard system are not an issue with the ground system. Thus, the team could focus more on finding an antenna with a large amount of gain and a desired radiation pattern without having to think too much about size or weight. With this in mind, the team decided on a four element Yagi Antenna (Model A09-Y8NF) from Arcadian Incorporated, shown in Figure 5. Its operational range falls within the 896-980 MHz which encompasses the desired operational zone for the project. The primary justification for choosing this specific model was that it provided a large amount of gain without sacrificing too much beamwidth. To be more specific, this model provides 8.5dBi gain, a 100 degree horizontal beamwidth, and a 70 degree vertical beamwidth. Despite the constrained vertical and horizontal beamwidth compared with an omni-directional antenna, the team is confident that they can adjust the design to compensate for this while still taking advantage of the high gain provided by the Yagi antenna. The specific design choices that account for the beamwidth constraints will be discussed in more detail later in this report.

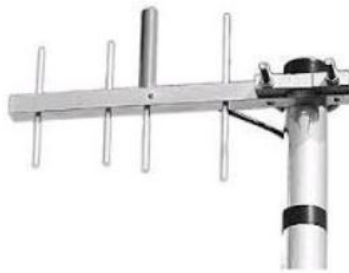


Figure 5. Ground Antenna Choice

d. Half-Duplex Communication In addition to the above considerations, the team also put some thought into how to communicate within the ISM band of choice. Since the bandwidth of this communications band is relatively narrow compared with other bands, 26 MHz, the team expressed a preference to communicate using the entire band rather than using half of the band for commanding and the other for telemetry. The negative side of this design choice is that the team will not be able to send commands and receive telemetry at the same time. In order to account for this, the team will implement a system to ensure that command uplink and telemetry downlink are offset. This will be discussed in more detail later in the report.

3.1.2. Communications Software

a. SDR Software After doing research on the various SDR software options available, two options were identified as the most viable: MATLAB/Simulink and GNU Radio. MATLAB is a closed source proprietary programming language and development environment that has a robust signal processing toolbox. GNU Radio, is an open source project that is popular among academics, hobbyists, and institutions. Three main criteria were used when evaluating the strengths and weaknesses of these two options: compatibility with the SDRs, performance, and ease of use. All of the SDRs considered in the trade study are well supported by GNU radio, with each having vendor-supplied plugins to use with their hardware. The MATLAB support for each SDR varied from having Mathworks supported libraries complete with example code to unofficial 3rd party support. While hardware specific support isn't necessary to interface with the SDRs, it does make it a lot easier. With the exception of the USRP B200mini, the GNU Radio support was equivalent or superior to the MATLAB support. When it comes to ease of use, most of the TOMCAT team have considerable MATLAB experience and limited GNU radio experience. The documentation and online resources for both are good and there they both have active help forums. In addition, both MATLAB and GNU Radio feature graphical programming interfaces that allow for more rapid design work. MATLAB supports standalone C++ code generation to a variety of targets. This means that the software design can be done in MATLAB code or graphically through Simulink, then turned into high performance C++ code that can be run on the

OBC. While GNU Radio supports Python and C++, it only supports Python code generation from graphical designs. If the Python code is too slow, then a C++ version has to be written manually. In addition, GNU Radio code still needs GNU Radio to be installed on the system to run, as opposed to MATLAB generated code which can run standalone. Overall, both options were deemed viable, and the final decision came with the result of the SDR trade. Since the LimeSDR had the worst MATLAB support out of all the options, GNU Radio was selected.

b. Modulation Technique To encode binary data in a transmittable wave, the data must be modulated in some way. The two major forms of modulation considered were phase shift keying (PSK) and frequency shift keying (FSK). Phase shift keying encodes binary data in a carrier wave through phase shifts, while frequency shift keying encodes data in frequency shifts. Though FSK provides a higher signal to noise ratio in general, it was decided that PSK would be used since it is more bandwidth efficient, more power efficient, and provides better bit error rates in noisy environments. Next, a decision was made on whether to use binary phase shift keying (BPSK) or quadratic phase shift keying (QPSK). BPSK uses two phases to encode data, while QPSK uses four. Though they have identical noise performance characteristics and QPSK is more bandwidth efficient, the decision was made to go with BPSK since the implementation is simpler and it still fulfills our data rate requirements. Finally, it was decided to use differential encoding (DBPSK). With differential encoding, the reference phase is the phase of the last sampled signal instead of the carrier reference phase. Since the carrier wave can be distorted during transmission, the received signal can be quite noisy. Differential encoding remedies this at the cost of slightly worse noise performance.

3.1.3. Flight Software and On-Board Computer

Grant Novota, Manuel Lindo, Lucas Perry

The purpose of our OBC and FSW language is to choose a MCU and IDE that will provide reliable development at low risk. Together, these three components should receive, parse, and execute payload commands in CCSDS format, as well as generate state of health telemetry. First, the MCU options were narrowed down to four options that fit the mission's requirements. The unit should be able to have desirable architecture/performance, low power requirements, sufficient GPIO Pin count, and an efficient development kit. The off-the-shelf MCU will not have to fit in the Bus right away. After software development is concluded in the development board, the team will make a custom board prototype with only the necessary components for our mission. This will significantly decrease the size of the on-board computer. The FSW should work exactly the same as long as the same firmware and bootloader are used on the custom board.

a. Integrated Development Environment The IDE's main objective is to ensure a good environment for development of software solutions. This choice becomes especially important when the users are not familiar with FSW development. It is crucial to have an intuitive IDE, with great documentation, and a wide user base. With this in mind, if there is a strong community and resources around the IDE, debugging and module implementation times can be greatly reduced. Furthermore, it is crucial for all collaborators to have access to the FSW development environment. Thus, the IDE should support all conventional operating systems such as Windows, Mac OS, and Linux. Above all, the most important factor is the IDE compatibility with the MCU. In high level development, payload and hardware integration are a big advantage. It must be able to fully communicate with the chosen MCU and, if possible, have comprehensive flight software services.

i. Arduino IDE The Arduino IDE is one of the most user-friendly and extensive libraries available for software development and hardware integration. Its vast libraries and packages offer tremendous capabilities. In addition, it has a strong community and plenty of resources to speed up development. However,

this is only true for low-level functionalities and interfacing with proprietary Arduino hardware. It is more difficult to debug and more expensive than the other options.

Pros	Cons
Vast Libraries and Packages	Deficient Debugging
Very easy for low level development	Simple Code Editor
Supported in all conventional operating systems	Shields away a lot of complexity and limits high-level development
Strong community and resources	More expensive relative to other option

ii. Eclipse C++ for Arduino Similar to the Arduino IDE, the Eclipse C++ Arduino IDE has the benefit of sharing Arduino's variety of libraries to simplify and expand the available resources of the IDE. The Eclipse Arduino Plugin allows for a far more complex range of uses than the simpler Arduino IDE. However, With this complexity comes a few drawbacks. Although the program has a lot of documentation, much of it is outdated and will not cover the depth of issues that could be involved in this project. The second and most notable downside of using Eclipse C++ for Arduino is the complexity of starting out on the program. With all the extra features and complexities comes bulky, difficult-to-interpret code that can also have a very slow execution time. Although the Eclipse C++ for Arduino IDE offered a more in depth and intricate option, the lacking help resources and complexity of the program still hold it back from satisfying the requirements needed for the IDE.

Pros	Cons
Built on one of the most complete IDEs	Fairly complex in initial phase
Access to Arduino's libraries	Reliability issues
Vast Eclipse plug-ins for prototype development	

iii. KubOS The third and final IDE option, KubOS, was initially found from a recommendation from the customer. While it is still an upcoming program, there is a strong support program within their development team that would make it feasible to get questions answered and setbacks resolved. It has a solid set of features for code creation, testing, and debugging, along with a high compatibility level with the MCU. With these features comes the downside of complexity. The team has no previous experience with KubOS, and one of the challenges will be becoming familiar with the system before progress can be made.

Pros	Cons
Comprehensive critical flight software services	Fairly complex in an initial phase
Strong development support	
Payload and Hardware Integration	
Customer's Recommendation	

b. Microcontroller Unit The MCU must be able to interface with and provide all of the control functionality necessary for the operation of the subsystems. It needs to operate within the power budget set by the system's power supply and be able to drive additional power to the subsystems it is controlling. In addition, it must be capable of meeting all of the real-time processing requirements that are set for the system to function properly as a whole. Another important factor to consider when choosing a MCU is the development process. There must be a development kit available that fits within the budget and provides the necessary functionality to develop and test both the software and hardware capabilities of the MCU before the final hardware is constructed.

i. ATmega328 The ATmega is an 8-bit AVR architecture MCU that operates at 20MHz and is commonly found in low profile micro-controller boards. This includes many Arduino-type boards, which have the benefit of being a familiar prototyping tool that is easy to learn and use. It provides 23 GPIO pins. This option also provides the opportunity to develop software using the Arduino IDE and libraries, which greatly simplify the embedded programming process by abstracting from the low-level code needed for the MCU to operate. The power requirement for this MCU is 1.8V - 5.5V, 25mA.

Pros	Cons
Low level of complexity	Minimal computing power
Access to Arduino's IDE and libraries	Low number of GPIO pins
Good documentation	Low efficiency

ii. ATSAMD21G18 The ATSAMD21G18 is a 32-bit MCU based on the ARM architecture that operates at 48MHz. The advantages of using an ARM processor are that they are cheaper and more power efficient than other processors based on other architectures. The high availability and applications support for ARM processors make them a popular choice for embedded computing. This processor is available on an Arduino board, which comes with the prototyping benefits associated with using Arduinos. It also provides 38 GPIO pins. The power requirement for this MCU is 1.62V - 3.6V, 25mA.

Pros	Cons
High efficiency	Medium number of GPIO pins
Good documentation	Medium computing power
Access to Arduino's IDE and libraries	Medium level of software development complexity
	Medium level of hardware complexity

iii. AT91SAM3X8E The AT91SAM3X8E is another 32-bit ARM MCU that provides a step up in performance from the ATSAMD21G18 with a higher clock speed at 84 MHz and more than double the number of GPIO pins at 103. This MCU is also available on an Arduino board. The power requirement for this MCU is 1.62V - 3.6V, 25mA.

Pros	Cons
High efficiency	Medium level of hardware complexity
High number of GPIO pins	Medium level of software development complexity
Access to Arduino's IDE and libraries	
Good documentation	
High computing power	

iv. AM3358 The AM3358 as a 32-bit ARM MCU with enough processing power to support high-level operating systems such as Linux. It is enhanced with graphics processing peripherals and industrial interface options. It also contains subsystems separate from the ARM core that provide more flexibility in implementing fast, real-time responses, specialized data handling operations, custom peripheral interfaces, and in allocating tasks to different processor cores. This MCU's processing capabilities come at the cost of increased complexity and power requirements. The power requirement for this MCU is 3.7V - 5V, 700mA.

Pros	Cons
Ability to run high-level operating systems	Large power requirement
Good documentation	High level of hardware complexity
High efficiency	
Low software development complexity	
Very high computing power	
High number of GPIO pins	

3.1.4. Ground Software

The main high-level objective of the ground software is to send commands, view telemetry in some kind of GUI, and store telemetry in an desired format to be analyzed. On a deeper level, the ground software must also be able to communicate with the Communications Software via a TCP or UCP connection, send and receive data formatted in a CCSDS Space Packet, and be flexible to different data rates for uplink and downlink. The best option, which also came highly recommended by the customer, is COSMOS. It is an open source ground software program created by Ball Aerospace that contains a suite of applications designed to control a set of embedded systems. All the commands being uplinked to the ThinSat and telemetry being downlinked from the ThinSat are processed by COSMOS's Command & Telemetry Server, then passed on to one of its specific applications. The most notable applications being used for the purposes of this project are the Telemetry Grapher, Telemetry Extractor, and Command Sender tools. COSMOS satisfies all requirements of a ground software program as it can communicate with the Communications Software, process data in almost any format, and provides a great user interface to view real-time telemetry data as it is being logged.

3.1.5. Thermal

Lara Buri

Because the end goal of the TOMCAT mission is to send the ThinSat and payload on a high-altitude balloon to 30km, it is important to take into account the cold environment at this altitude. It is expected that the ThinSat will require some type of thermal control to maintain adequate temperatures inside the structure to ensure the battery and electronic components can operate properly. Heritage data from Near Space Launch includes temperature data taken during high-altitude balloon flights for the same ThinSat frame design as TOMCAT. This data suggests that with no thermal protection or control, temperatures inside the ThinSat reach around -40°C. These temperatures are much too cold for a battery to maintain optimal operation and reach the threshold at which the electronic components will get too cold for operation.

a. Thermal Control System The first option to maintain temperatures within the ThinSat and prevent the ThinSat from becoming too cold is to add a thermal control system. This would include a heater and a feedback control loop that would allow the heater to turn on if the ThinSat becomes too cold, and turn off if it becomes too warm. The following table presents the pros and cons of using an active thermal control system.

Pros	Cons
Allows for control of internal temperature	Complicated to model
Lowers risk of failure due to low temperatures	Requires more battery power
	Requires implementation of control law on OBC

b. Thermal Protection System The second option to maintain temperatures within the ThinSat is to add thermal protection, or insulation. This option is a much simpler solution to keep the ThinSat electronics and battery warm and is more cost-effective in terms of both monetary cost and power. The following table presents the pros and cons of using a thermal protection system.

Pros	Cons
Easier to model	Little control of internal temperature
Does not use any power	Will change dimensions of ThinSat
Lower cost option	

3.1.6. Power

Jake Schroeder

While a power supply is available from the ThinSat Bus, the power required from the SDR alone is several times that which is supplied. A secondary or completely standalone power supply enables the ability to power on-board computers that are able to take advantage of more powerful flight software, such as those built off Linux like KubOS. The power supply, like the SDR, does not fall within the weight and size constraints of the original ThinSat payload, but does fall within the new mass requirements specified for the Gateway to Space Launch and the size requirements for the double ThinSat design being used for the high altitude balloon flight. The battery was chosen based on its feasibility for use in a near-space environment and its ability to fulfill power requirements of each of the subsystems for a two hour high altitude balloon flight.

a. Power Supply Types When determining the type of power supply required, it was important to distinguish between the scope of the project to launch on a balloon satellite and the ultimate goal of launching into Low Earth Orbit. One commonly used power supply option for satellites of any size in orbit is a combination of solar panels and batteries since they allow for missions to be recharged almost indefinitely. While solar panels may be useful in the future for use on the payload of ThinSats when they are launched into orbit, in order to operate on a high altitude balloon flight for a maximum of two hours, no long-term power supplies are required. Even for ThinSats launched into extreme low Earth orbit, the typical lifespan is only around five days, meaning batteries could still potentially be useful without the help of generating power from the sun. While there are other energy storage options, nearly all high altitude balloon payloads use some form of battery due to their high energy density, low cost, simplicity, ease of use, and reliability. There are several types of chemical batteries that were considered. For a balloon satellite mission, the most important aspects of a battery considered were operational temperature, physical dimensions, power output, energy density, and feasibility in a vacuum environment.

i. Alkaline Household alkaline batteries are proven to be flight tested and reliable, with a large temperature range from -20°C - 55°C . As with all chemical batteries, power output decreases as temperature decreases. These batteries are inexpensive and are widely available at general stores. Their primary drawback is their inability to be recharged. For single missions this is acceptable, but becomes more difficult when the payload needs to be tested since every test will require a new set of batteries. These batteries are also less power dense with respect to volume compared to Lithium Ion and Lithium Polymer batteries, so physical dimensions were important to take into account when considering Alkaline batteries.

Pros	Cons
High energy:weight ratio	Not rechargeable
Large temperature range	New batteries required for every test
Flight Heritage	Lower energy:volume ratio

ii. Lithium Polymer Lithium polymer batteries are generally very similar in cost and performance to lithium ion batteries, and are commonly used in rechargeable portable electronics. Lithium polymer batteries offer similar life cycles, charging/discharging safety, and performance at lower temperatures. In the past, Lithium Polymer batteries have been used on a number of CubeSat flights as well as previous CU Senior design projects such as RAVEN [12] and ELSA [13], so several resources for similar missions could be used for reference.

Pros	Cons
Rechargeable	Smaller temperature range
Lightweight, flexible casing	Slightly more susceptible to damage from external pressure
High energy density	

iii. Lithium Ion Lithium ion batteries are most commonly found in hand-held electronics such as smart phones and are relatively inexpensive. They have a large operational temperature range from -20°C - 60°C, but decrease in effectiveness as temperature decreases. Lithium ion batteries have slightly higher power densities compared to Lithium Polymer batteries, and while they have similar operating temperature ranges, Li-Ion batteries tend to remain only 55%-66% effective at -20°C compared to 70% effective for lithium polymer batteries. Most commonly, Lithium Ion batteries come in cylindrical cells, which are more power dense, but utilize the available volume inside the ThinSat less efficiently. Additionally, there is an extensive body of research on Lithium Ion batteries detailing the effects of temperature, discharge rate, and other variables because of their wide use in consumer electronics.

Pros	Cons
Rechargeable	Cylindrical Volume
High Energy Density	Less prone to external pressure damage
Large Body of Research	

3.1.7. Structures

Matthew McCallum

Very early on in the project design, it was determined that we would not be able to fit all of the necessary hardware within the payload space of the frame that we would be purchasing from NearSpace Launch Inc. (NSL). This single frame is known as a 1T configuration for the ThinSat. Our ThinSat still needed to be able to interface with the hardware that NSL has designed for the balloon flight and rocket launch, so the only option available to us was to make our ThinSat use the volume of two standard ThinSats by attaching two frames together. The use of two frames like this is known as a 2T configuration. This conversion more than tripled the amount of space available for our components because the second frame does not need to have a section for NSL's hardware, and we can use the space in between the frames that would normally be taken up by cover plates.

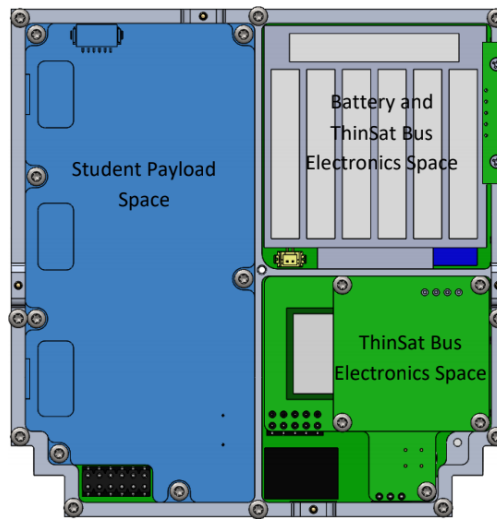


Figure 6. 1T Configuration ThinSat that will be purchased from NearSpace Launch Inc.

3.1.8. Scientific Sensor

Lara Buri, Austin Scheck, Matthew McCallum

Though the primary focus of the TOMCAT mission revolves around communication and data handling (C&DH) systems, the verification of communications can only be done through the transfer of information. This is where the scientific sensor finds its way into the TOMCAT mission. In order to successfully test the complex communication system involving software defined radios, antennas, software for both ground and flight operations, a data generation source needs to be on the satellite.

The team's sponsor, Raytheon, has requested that the on-board scientific sensor provides data that is applicable to both the upcoming balloon flight and an eventual spaceflight mission in Low Earth Orbit.

The types of instruments chosen for the preliminary trade study were researched based on scientific merit, team interest, and Raytheon's interest. These decisions were additionally informed by researching CubeSat and high-altitude balloon missions, analyzing successes and failures involving each of the instruments.

All of the considered options for a scientific sensor are described in the following section. In total, the team considered an imaging sensor/camera, a magnetometer, an inertial measurement unit, an environmental sensor suite, and a radiation sensor/dosimeter.

a. Imaging Sensor/Camera The first option for consideration was an imaging system, such as a visible light camera, ultraviolet (UV), or infrared (IR) imager. There have been many recent CubeSat missions aimed at revolutionizing the Earth-imaging field using satellite constellations [14] as seen in Fig. (7). Due to the ThinSat's size and low cost, very large swarms of these satellites could be used to image the Earth, with each individual satellite focused on one spot over the Earth. Images of Earth in the visible, IR, and UV spectrum provide valuable data that can be used for meteorological and geographical purposes. The team looked specifically at small complementary metal-oxide-semiconductor (CMOS) cameras, charge-coupled device (CCD) imagers, IR sensors, and UV sensors.



Figure 7. PlanetLabs visible spectrum satellite concept [1]

CCDs use thin silicon chips that contain light-sensitive squares. Each square on the chip represents a pixel in the final image. Bright objects in the field of view fill the light-sensitive squares with more electrons than a dim object. Once exposure finishes, the charge is sent from the CCD to a computer for processing. The larger the CCD chip, the more pixels a photo can have, and therefore the better the resolution [15].

With CMOS sensors, the charge buildup on each photo-sensitive square is converted to a voltage and the signal is multiplexed to an analog-to-digital converter. Because CMOS sensors are digital devices, they are inherently faster than CCD chips, but lack in the higher resolution of CCDs. However, in terms of processing complexity, CMOS sensors reduce computing complexity since the signal sent is a voltage, rather than an electron packet like in a CCD [16].

Finally, UV and IR imaging sensors were considered for use on the ThinSat payload. The team was interested in experimenting with different areas of the radiation spectrum when imaging the Earth. IR cameras pointed at Earth during the balloon flight could give data about cloud cover. A UV sensor could characterize how UV radiation varies throughout the atmosphere.

The largest downside to imaging sensors was the large amount of computational power required to convert the sensor readings to images or applicable data. IR cameras require complex computations to convert raw voltage data from the imager to tangible data (typically temperature data) [17]. The CMOS and CCD both present challenges with the large amounts of data required to produce an image with high enough quality.

The table below presents a list of pros and cons for imaging sensors. This list was informed through research on specific types and brands of imaging sensors, with emphasis on CMOS and CCD cameras.

Pros	Cons
Most imaging sensors are relatively low in cost	Some sensors require a large power draw
Large operational temperature range	Due to the need for a lens system, some imaging sensors may be larger than the thickness of the payload space
Provides relevant data for both balloon and space missions	Large amounts of data to store, send, and process

b. Magnetometer A magnetometer measures the strength of electromagnetic fields. They can be multi-axis and provide information on the strength and direction of the electromagnetic field. This sensor would provide applicable data to both aeronautic and astronautic flight, albeit to varying degrees of effectiveness.

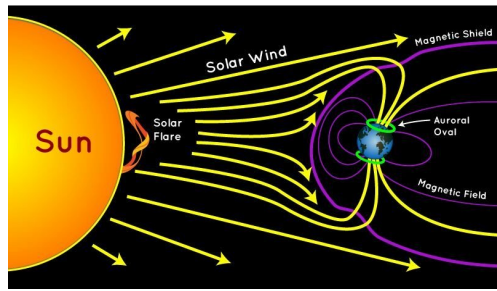


Figure 8. Magnetic Field Fluctuations with Solar Wind [2]

The Earth's magnetic field is constantly changing in response to the solar wind. These changes occur all around the Earth, so a swarm of small satellites with 3-axis magnetometers would be able to simultaneously measure these changes in real time. This data would provide information about the Earth's magnetic field as well as information about the solar wind. The effect of solar wind on Earth's magnetic field is shown in figure 8. Previous space missions have studied the fluctuations in magnetic field and their effects on the aurora [18]. In the atmosphere, a magnetometer acts like a multi-axis compass. This allows it to be used as a reference direction to limit the drift of an accelerometer or as a means to establish an orientation for state estimation applications.

Pros	Cons
Low power draw	High cost for acceptable resolution
Small size	Data not as applicable in a balloon flight as it is for a space mission
Wide operational temperature range	
Good documentation	

c. Inertial Measurement Unit Inertial Measurement Units, or IMUs, can provide 3-axis data on acceleration, rotation, and magnetic flux. They provide useful state information about whatever they are attached to, which can be instrumental in determining causes of error in the case of a malfunction. More advanced applications of these sensors include using IMUs to measure wind gusts by attaching them to a sUAS [19], or using them on a balloon flight to measure atmospheric turbulence [20]. An example FBD for this type of mission is shown in fig. (9).

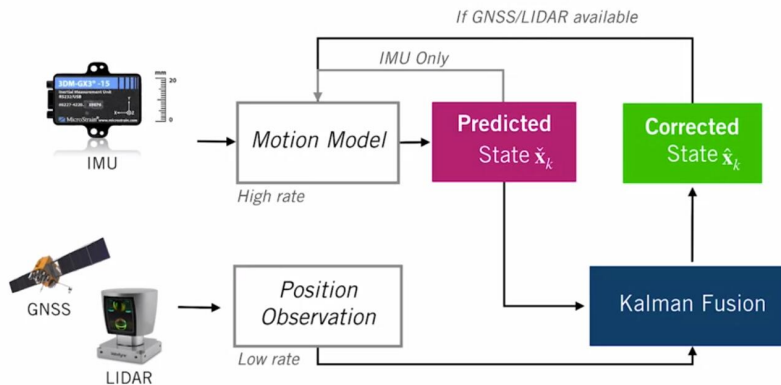


Figure 9. Sample FBD of a state estimator using an IMU [3]

The data from the IMU could be used on both the balloon flight and in space for state estimation. The accelerometer and gyroscopic data can be integrated to get the position and orientation of the satellite if starting conditions are known. However, these measurements are prone to drifting due to the way that the errors propagate. The magnetometer could be used to help mitigate this by using the magnetic field as a reference direction. Additionally, other sensors located on the bus can be used in the state estimation algorithm to further increase its accuracy at the cost of technical and computing complexity.

Pros	Cons
Small size	Possibility of on-board data processing
Wide operational temperature range	Science data is not as applicable for a balloon flight
Wide operational temperature range	High cost for acceptable resolution
Good documentation	
Provides state data that may be useful for space applications	

d. Environmental Sensor Suite An environmental sensor suite was additionally considered for the high-altitude balloon flight. The team was specifically interested in barometric pressure, temperature, and acceleration. Each of these sensors help characterize balloon flights. The payload will experience very low temperatures and extremely low pressures at the 30km altitude the balloon will reach. Potential high acceleration loads are expected when the balloon bursts, when the parachute deploys, and when the ThinSat lands. These loads can be characterized using an accelerometer on the payload.

Data received from the temperature sensor can be directly compared to radiosonde data from the day of the launch. Pressure decreases with altitude, and the maximum altitude of the balloon flight can be determined from the readings from the pressure sensor. Figure 10 shows the typical temperature profile throughout the atmosphere that a temperature on a balloon flight might encounter.

Though environmental sensors such as temperature and pressure yield important and interesting results during a balloon flight, they will not be of much use in a space environment. Due to the lack of air, a pressure sensor in space does not give any valuable data, as with temperature.

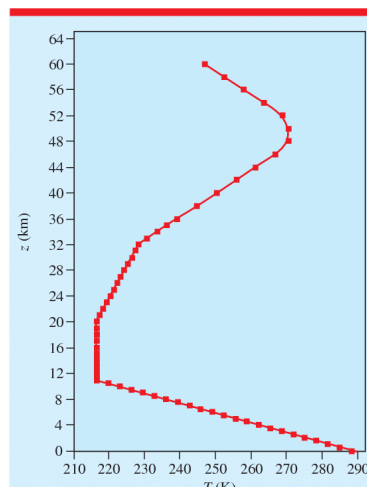


Figure 10. Temperature Profile Throughout the Atmosphere [4]

Pros	Cons
Environmental sensors require low power and don't require extensive data processing	Large amount of space taken up by scientific sensor suite
Large operational temperature range	Sensors are not as useful in a space application.
Low cost	
Good documentation	

e. Radiation Sensor/Dosimeter The final scientific payload considered by the team consisted of a radiation sensor or dosimeter. This mission would be scientifically profound, applicable to both atmospheric balloon flight and space flight, and would provide more insightful data compared to basic environmental readings like temperature and pressure. This mission would allow the team to measure the Regener-Pfotzer Maximum, the point in Earth's atmosphere where ionizing radiation reaches a maximum, occurring around 20 kilometers in altitude (see Figure 11). The data taken would perhaps corroborate recent research suggesting that galactic cosmic ray exposure is becoming an increasingly-prevalent danger for human spaceflight, and even commercial aviation [21] [22]. This astronautic phenomenon could be readily measured during balloon flight, and subsequently compared to research being done by NASA's CRaTER mission on the Lunar Reconnaissance Orbiter.

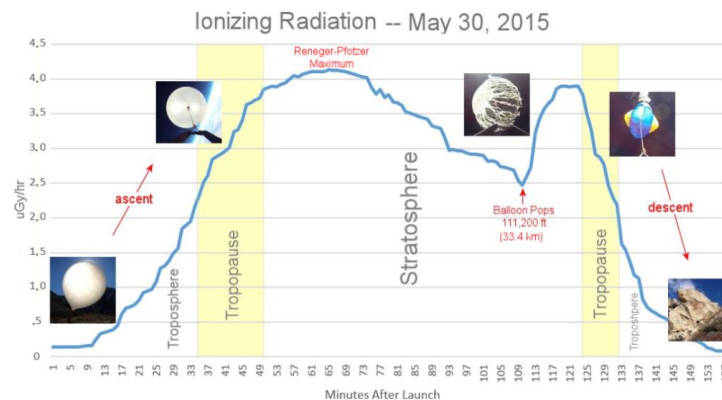


Figure 11. Regener-Pfotzer Maximum [5]

The radiation sensor, unfortunately, presented limitations on sizing, documentation, power requirements, and survivability. Of the two models most thoroughly considered, the Radiation Watch Type 5 Pocket Geiger Radiation Sensor and the SkyFox Labs piDOSE, both require significant volume as compared to other sensors, stretching 16.016 and 23.744 cubic centimeters respectively, and offer little documentation. The lack of documentation provides significant difficulty in the eventual integration of these sensors, especially when writing software for interfacing with the microcontroller unit on the payload board. Additionally, the piDOSE offered by SkyFox labs retails for \$4,300, a value far outside the budget allocated for scientific research, making the Radiation Watch model the only truly viable option of the ones considered thus far.

Geiger counters, apparatuses which count the amount of beta particles and gamma rays in the vicinity, require a high voltage to successfully operate. This component, present on the considered radiation sensors, mandates that voltage must be high enough to generate an electric field through the tube-shaped apparatus. Without this electric field, the sensor cannot accurately assess the surrounding particulate, rendering the whole sensor dysfunctional. Compared to other mission types analyzed above, this mission necessitates hefty sacrifices for more power. Finally, these sensors are limited by thermal conditions, only reaching -20°C in their minimum operational temperature. As the inaugural balloon flight is anticipated to reach temperatures in excess of -50°C, this scientific sensor would require insulation or thermal regulation to properly function during the mission.

In all, the radiation sensor mission, with its limited product documentation, promises impressive results, but at the cost of extensive accommodations for sizing, power, and temperature control.

Pros	Cons
Scientifically-relevant data can be obtained on a balloon flight as well as in orbit	Geiger counters can be large in size due to the need for a Geiger-Mueller tube
Provides relevant data that can be compared to other sources	Little documentation
Sensor easily tested in a lab setting	Requires a higher voltage than provided-transformer necessary to step up the voltage
	Dosimeters and Geiger counters can be very expensive, especially those that are smaller in size

3.2. Trade Study Process and Results

3.2.1. Communications Hardware: SDR

a. Metrics and Weighting The metrics that the team decided upon as determinants of an SDR's compatibility with the scope of the project are conveyed in Table 5 along with the rationale for these choices and associated driving requirements. In addition to this, a weighting scheme was developed to allow the aspects that were deemed more important to have a greater influence on the overall scoring. These metrics and their corresponding weights reflect the needs and priorities of the team as well as the customer.

Table 5. Trade Topics and Weights for the Software Defined Radio

Metric	Weight	Driving Requirement	Rationale
Max Power Draw	0.1	N/A	Power draw should be minimized to enable the smallest possible power supply. Due to the minimal power supplied from the ThinSat bus, an alternate power source will be used to power the SDR and associated components.
Size	0.1	N/A	The size of the SDR must be minimized. While the SDR is not required to fit within the payload space, it will need to attach to the exterior of the bus.
Mass	0.1	N/A	The mass of the SDR must be minimized.
Operating Temperature	0.15	3.1	The SDR must be capable of TX/RX over the expected temperature range.
Cost	0.05	5.1	Cost must fit within project budget. While cost is an important factor to consider, the other components that the team is required to purchase will only take a small portion of the budget. Thus, there is more room for flexibility in this category.
Bandwidth	0.1	2.4,2.5,2.6	The bandwidth is related to the quantity of data that can be transmitted, received, and handled. It is essential for the radio to be able to effectively transmit and receive important command and telemetry data.
Max TX Gain	0.1	2.2,2.4	Gain will drive the maximum transmission distance across which the ground SDR can communicate with the onboard SDR, which will be critical during the balloon flight.
Max RX Gain	0.1	2.2,2.4	Many SDRs have differing TX and RX gain, but the rationale for this metric is the same as above.
Configurability/Compatibility	0.2	2.3	Ease of configurability/compatibility will diminish complexity and time consumption involved with configuring the SDR interface with the Raytheon ground system and OBC.

b. Metric Quantification The ranges across which the metrics were scored were dependant on the available SDRs on the market as well as the project requirements. Because of the much higher power consumption of the SDR over other systems, the range over which this metric is scored is much larger than in other studies. Similar to other studies though, operating temperature scores also serve as a proxy for scoring the complexity of the necessary thermal system that we will have to implement after further analysis. Specifically with regards to configurability and complexity, which is the only metric not supported quantitatively, higher scores were given to SDRs which are supported in MATLAB, as the team has considerable experience working with it.

Table 6. Explanation of Analytical Scale for the Software Defined Radio

Metric	0	1	2	3	4
Max Power Draw	≤ 1000 mA	≤ 750 mA	≤ 500 mA	≤ 250 mA	≤ 100 mA
Size	≥ 90 cm ²	≥ 72.9 cm ²	≥ 55.8 cm ²	≥ 38.7 cm ²	≥ 21.6 cm ²
Mass	≥ 100 g	≥ 80 g	≥ 60 g	≥ 40 g	≥ 20 g
Operating Temperature	untested	25 °	0 °	-25 °	-50 °
Cost	$\geq \$759$	$\geq \$616.74$	$\geq \$474.50$	$\geq \$332.32$	$\geq \$189.95$
Bandwidth	≤ 28 MHz	≤ 45 MHz	≤ 62 MHz	≤ 79 MHz	≤ 96 MHz
Max TX gain	≥ 45 dB	≥ 60 dB	≥ 70 dB	≥ 80 dB	≥ 90 dB
Max RX gain	≥ 65.5 dB	≥ 70 dB	≥ 80 dB	≥ 80 dB	≥ 90 dB
Configurability / Compatibility	3rd Party Configuration SW	Major Configuration SW	Major Configuration SW, limited 3rd party MATLAB support	Major Configuration SW, 3rd Party MATLAB Interface	Major Configuration SW, Mathworks Supported

c. Trade Results Once the metric scale was developed, the various attributes of the four SDRs were rated accordingly and the results are illustrated numerically and visually (color coded) in Table 7. As can be deduced from these results, the leading model choices were the LimeSDR Mini and the USRP B200mini. Although the LimeSDR lost out in the configurability/compatibility and TX gain categories, it made up for it with its superior bandwidth capabilities, modest cost, lower minimum operating temperature, and minute size. The low TX gain can be compensated for by implementing a form of RF amplification in order to achieve the desired communication range. The LimeSDR primarily lost out in the configurability category due to the minimal MATLAB support, which is what the TOMCAT team has the most experience implementing. Thus, the configuration time and complexity on the ground SDR side will be increased slightly but not to an extent that is seen as unreasonable. However, the LimeSDR was noted to also have GNU Radio support. The team anticipates this to be advantageous for interfacing with the on-board SDR.

As a general note, this trade study resulted in an overall under-performance of the selected SDR models, with final scores ranging in the 1s and 2s as opposed to the 2s and 3s like in other subsystem trades. The reason for this is not because of a poor selection of SDR choices, but because of the very specific constraints of the projects resulting in very few SDRs that fulfill all objectives.

Table 7. Trade Study Results for the Software Defined Radio

Metric	Weight	Lime SDR Mini	BladeRF 2.0 Micro	HackRF One	USRP B200mini
Max Power Draw	0.1	0	0	2	0
Size	0.1	4	1	0	3
Mass	0.1	4	1	0	4
Operating Temperature	0.15	3	2	0	2
Cost	0.05	4	2	4	0
Bandwidth	0.1	4	1	0	1
Max TX Gain	0.1	2	4	0	4
Max RX Gain	0.1	1	2	4	0
Configurability/Compatibility	0.2	2	2	3	4
Total	1	2.55	1.7	1.4	2.3

3.2.2. Communications Software

3.2.3. Printed Circuit Board

A trade study was not needed for the printed circuit board.

3.2.4. Flight Software and On-Board Computer

Grant Novota, Manuel Lindo, Lucas Perry

a. Metrics and Weighting The metrics used to conduct the trade study below are described in Table 8. They were also given weight values according to their importance as defined by requirements. These measure the most significant features of the MCUs that are needed to make comparisons in order to select the optimal MCU for the system. We chose not to include other metrics like size, operating temperatures, and cost because all of the MCUs are nearly identical in these areas.

Table 8. Trade Topics and Weights for the MCU

Metric	Weight	Driving Requirement	Rationale
Architecture/Performance	0.4	4.1, 4.2, 4.3	The MCU must be able to meet all of the real-time processing requirements for our operation. It must have the architecture and performance capabilities to execute the required instructions before their deadlines.
Power Requirement	0.1	N/A	The MCU must operate within the power budget.
GPIO Pin Count	0.2	4.1, 4.2.1, 4.2.3	The MCU must be able to interface with all required subsystems
Development Kit	0.3	N/A	The MCU must be compatible with software developed before the final microcontroller hardware is finished with the help of a development board that is reasonably easy to use and within the project budget.

b. Metric Quantification Table 9 includes the details of how each metric is scored. The metric carrying the greatest significance in the trade study is the architecture and performance of the MCU. This is what will determine how capable an MCU is at meeting the real-time processing requirements of the system we will be designing. An MCU based on 8-bit architecture is significantly weaker and less efficient than an MCU based on 32-bit ARM architecture. A 32-bit ARM architecture will have increased processor register memory width and include five stage pipe-lining to further increase the speed that instructions are executed beyond the clock speed of the processor. Speed increases of up to 5 times are attainable over a comparable 8-bit architecture. The MCU's clock speed will determine the likelihood that software instructions are executed by their deadlines, which are set by the requirements of the system.

The next metric considered is the power needed for the MCU to operate. This will influence the amount of power the system needs to be supplied. An MCU with more efficient architecture lowers the power requirement, and increasing clock rates bring increasing power requirements.

GPIO pin count influences how many subsystems are able to be controlled by the MCU. More GPIO pins provide a less constrained control system that is able to interface with more subsystems.

The last metric considered was the development kit for the MCU. The availability of a development board to be purchased early in the development process is crucial in order to have the time to design the flight

software and explore the capabilities of the MCU before a final hardware design is constructed. The cost of the development board needed to fit within the budget constraints of the project.

Table 9. Explanation of Analytical Scale for the MCU

Metric	0	1	2	3	4
Architecture/ Performance	8-bit architecture with a clock speed $\leq$ 20 MHz	8-bit architecture with a clock speed $>$ 20 MHz	32-bit ARM architecture with a clock speed $>$ 30 MHz	32-Bit ARM architecture with a clock speed $>$ 80MHz	32-Bit ARM architecture with extra processor cores and a clock speed $\geq$ 1GHz
Power Requirement	Operating requirements greater than 5V, 200mA	Operating requirements greater than 5V, 100mA	Operating requirements greater than 3.3V, 50mA	Operating requirements greater than 3.3V, 40mA	Operating requirements less than 3.3V, 40mA
GPIO Pin Count	$<$ 20 pins	$<$ 30 pins	$<$ 40 pins	$<$ 50 pins	$<$ 60 pins
Development Kit	A prototyping/development board is not available to purchase	A prototyping/development board costs more than \$200	A prototyping/development board costs less than \$200	A prototyping/development board costs less than \$100	A prototyping/development board costs less than \$70

Table 10. Trade Study Results for MCU Selection

Metric	Weight	ATmega328	ATSAMD21G18	AT91SAM3X8E	AM3358
Architecture/Performance	0.5	0	2	3	4
Power Requirement	0.1	0	4	4	0
GPIO Pin Count	0.2	1	3	4	4
Development Kit	0.2	4	4	4	4
Total	1	1.0	2.8	3.5	3.6

c. Trade Results The MCU trade study results show equal scores for the AT91SAM3X8E and the AM3358. Both of these MCUs have development kits available to be purchased at low cost and they both have a large number of GPIO pins available to use for interfacing with the various subsystems that require controls. The differences between these two MCUs are that the AM3358 is significantly more powerful and complex than the AT91SAM3X8E and as a result, it requires more power. The advantage of the AM3358's computing power is that it is capable of running a Linux kernel; something none of the other MCUs are capable of. This is a significant factor that lowers the difficulty of developing flight software because the operating system abstracts from the hardware and thus, allows the use of high-level and diverse programming. This advantage is well worth the increased power requirement that comes with the AM3358.

3.2.5. Flight Software IDE

a. Metrics and Weighting As discussed previously, the IDE selected plays a large role in providing a good environment for the flight software development. The selected metrics of importance are displayed below. The weighting of each metric is also an important factor of the trade study process, and so these were taken into account based on what was needed most to have a successful IDE to fulfill the requirements of the project as defined by the project requirements.

Table 11. Trade Topics for the IDE

Metric	Weight	Driving Requirement	Rationale
Complexity	0.15	N/A	There is a steep learning curve for some of the more robust IDEs. It can be difficult to understand all of the IDE's capabilities and use them to drive the FSW development forward.
MCU Compatibility	0.3	4.1	FSW developed in the IDE and MCU should be compatible for the payload to be commandable, generate telemetry, and store all telemetry during flight.
OS Compatibility	0.1	5.3	It should support the conventional operating systems to ensure every collaborator can access and modify the FSW.
Popularity	0.2	N/A	Community help and resources are very important in high-level development
Programming Language Support	0.05	N/A	It should support conventional programming languages
Additional Features	0.2	N/A	It should provide a convenient developing environment for code compilation, debugging, and team collaboration

b. Metric Quantification The scoring criteria are described in the following table. In general, metrics which were directly related to project requirements were weighted higher. The compatibility metric scores were determined by comparing the IDE's ability to communicate with either the MCU or a computer operating system during development. This is also how the program language support category was scored. Complexity and Popularity were scored using relative comparisons, with complex and unpopular IDEs being unknown to not just the team, but also the community at large. Conversely, less complex, popular solutions had support from a wide community online and at least one team member of TOMCAT was familiar with the IDE.

Table 12. Explanation of weights for IDE

Metric	0	1	2	3	4
Complexity	Unfamiliar to all team member-s/very unintuitive with minimal documentation	Unfamiliar to all team member-s/very intuitive and step-by-step documentation.	Previously used by or familiar to at least one team member/ very intuitive with step-by-step documentation	Previously used by or familiar to >1 team member-s/very intuitive and step-by-step documentation	At least one team member has expertise/extremely intuitive with step-by-step documentation
MCU Compatibility	The MCU is unable to communicate with the IDE.	The MCU can partially communicate with IDE. None of the IDE's features are compatible with MCU	The MCU can partially communicate with IDE. Some of IDE's features are compatible with MCU	The MCU can fully communicate with IDE. All of IDE's features are very compatible with the MCU.	The MCU can fully communicate with IDE. All of IDE's features are very compatible with the MCU. There is extensive MCU compatible libraries/plugins and, payload and hardware integration.
OS Compatibility	The IDE is not compatible with readily available operating systems	Linux only	Linux and either Mac OS or Windows	Mac OS, Windows, Linux	Mac OS, Windows, Linux, Web Browser
Popularity	The IDE is widely unknown.	The IDE is known in specific community, not very well rated by users.	The IDE is widely known, not very well rated by users.	The IDE is widely known, very well rated by users.	The IDE is industry practice and has outstanding user ratings, with minimal criticism.
Programming Language Support	C,Python/Micro and Arduino not supported				C/C++ or Python/Micro Python or Arduino fully supported
Additional Features	Only code compilation	Code compilation, and simple debugging	Code Compilation, complex Debugging, and Help highlighting	Code Compilation, Run Code, Debugging, Help highlighting	Code Compilation, Run Code, Debugging, Help highlighting, Team Collaboration

c. Trade Results The results of the trade study using the above metrics to judge three options for IDEs is shown below. The narrowed down options for IDEs were Eclipse C++ for Arduino, KubOS, and the Arduino IDE. All of these options were described more in detail in Section 4.

Table 13. Trade Study Results for the Flight Software IDE

Metric	Weight	Eclipse C++ Arduino	KubOS	Arduino IDE
Complexity	0.15	3	1	4
MCU compatibility	0.3	2	4	2
OS Compatibility	0.1	3	3	4
Popularity	0.2	2	4	3
Programming Languages Support	0.05	4	4	4
Features	0.2	3	3	1
Total	1	2.55	3.25	2.6

After using the decided upon criteria to score the Eclipse, Arduino, and KubOS IDEs, it was concluded that the best choice was KubOS. The KubOS operating system scored well on all metrics considered, aside from complexity. As was discussed in the informational paragraph about KubOS, its biggest downside is the unfamiliarity of the program to all the members of the group. KubOS was scored a 1 in complexity because it is believed that although the team has no experience to start with in KubOS, it was intuitive and would be learned swiftly. The KubOS IDE received a 4 in the MCU compatibility metric due to its extensive libraries and plug-ins accessible by the MCU, along with all the communication compatibility requirements. Continuing on KubOS received a 3 in OS compatibility because it works with Mac OS, Windows, or Linux, which are available operating systems usable to the team. The KubOS IDE got a 4 in popularity due to the customer wanting it to be used, and only good ratings were found from those who have used KubOS. KubOS also got a 4 in the Programming languages metric as it simply supports the desired language to create the software development with. Finally, KubOS received a 3 in Additional Features because it surpassed the level 3 requirements but unfortunately did not have a team collaboration component to it, failing to fulfill that requirement.

Overall KubOS was found through this trade study that it will work the best for the purposes of this project. Although complex, it has a large amount of resources, great compatibility and features that will make it ideal for this project.

3.2.6. Ground Software

There were a few options considered for the ground software.

a. KubOS Because it was already being designed as a flight software to receive CCSDS packetized data, KubOS was considered as a possible ground software. Some of the pros and cons of KubOS are displayed below.

Pros	Cons
Built-in packetizing/depacketizing CCSDS formatted data	Not created to act as a ground software
Can receive and transmit data through a UDP/TCP port	No graphical user interface
Already being used for FSW so minimal extra work involved	Would require another PCB Board on the ground side

Table 14: Pros and Cons of KubOS as a ground software

In the end KubOS did not fulfill all the requirements needed for a ground software as it lacked the interface and full scope of a ground system software program. KubOS was decided against as the ground software.

b. Creating an original program One of the options with the best ability to meet the needs of the ground software requirements was to write the code for an original program using C++ or Matlab. Some of the pros and cons considered for this option are shown below.

Pros	Cons
Could be written specifically to fulfill the requirements of the project	Would be extremely difficult and time consuming
Would be very interesting and educational	Out of the scope of the purpose of the project
	Not guaranteed to work

Table 15: Pros and Cons of creating an original ground software

Because creating an original ground software program was not a requirement and would be a huge amount of extra and unnecessary work, this option was decided against.

c. COSMOS As an open source program with a lot of useful applications, COSMOS seemed to satisfy all the necessary requirements. Some pros and cons are shown below for COSMOS.

Pros	Cons
Command Sender application sends command packets to a specific target	Configuration code is written in Ruby so extra work required
Can receive and transmit data through a UDP/TCP port	
Telemetry Grapher and Extractor applications can graph data on a GUI and store it	

Table 16: Pros and Cons of COSMOS as a ground software

In the end COSMOS was chosen as the ground software because it is a very powerful program that easily fulfilled all the necessary requirements. Many of the built in applications were usable for the mission, so the design portion comes in the form of writing configuration files for the tools and creating specific commands to be tested.

3.2.7. Power

Jake Schroeder

Three main types of battery compositions were compared: Alkaline, Lithium Polymer, and Lithium Ion. The primary driving factors for the battery selection were physical volume, power density, operational temperature limitations, ease of use and reuse, and number of informational resources available.

i. Alkaline The pros and cons for Alkaline batteries is shown below:

Pros	Cons
High energy:weight ratio	Not rechargeable
Large temperature range	New batteries required for every test
Flight Heritage	Lower energy:volume ratio

Table 17. Pros and Cons of Alkaline Batteries

While Alkaline batteries are flight proven in high altitude balloon missions and have ratings down to much lower temperatures than typical rechargeable batteries, the inconvenience of not being rechargeable

is a significant reason not to choose them as the payload power supply. Additionally, at least 5 9V batteries would be required to power the mission for 2 hours, and would not fit in the payload volume available.

ii. Lithium Polymer The pros and cons for lithium polymer batteries are shown below. They are fairly similar to those of lithium ion batteries, so the decision to choose either Li-Po or Li-Ion wasn't as clear cut as with Alkaline batteries.

Pros	Cons
Rechargeable	Smaller temperature range
Lightweight, flexible casing	Slightly more susceptible to damage from external pressure
High energy density	

Table 18. Pros and Cons of Lithium Polymer Batteries

Lithium Polymer batteries have flight heritage with past senior projects, and have a fair amount of research behind them. Choosing between Li-Ion and Li-Po batteries primarily came down to what was available to fit within the size constraints of the ThinSat, and also being able to provide the necessary power for the 2 hour flight. Lithium Polymer cells are often rectangular, which would allow for the team to maximize the volume for batteries inside the ThinSat. However, the majority of cells available didn't allow enough room for the communications system, or were overall too large for their given capacities, and the cells that did meet our constraints weren't readily available from reputable companies with shipping within the United States.

iii. Lithium Ion The pros and cons list for typical lithium ion batteries is shown below:

Pros	Cons
Rechargeable	Cylindrical Volume
High Energy Density	Less prone to external pressure damage
Large Body of Research	

Table 19. Pros and Cons of Lithium Ion Batteries

Lithium Ion batteries are widely used commercially and have one of the largest bodies of research surrounding them, with extensive information on their operation in low temperature environments and at different discharge rates. Lithium Ion batteries were chosen by the team to be the best battery option because of their large power capacity, their wide availability in a number of sizes from reputable sources shipping from within the United States, and the large body of research that gave the team confidence in their applicability and usefulness in a high altitude balloon flight.

3.2.8. Structures

A trade study was not needed for the structures.

3.2.9. Science Sensor Type

Lara Buri, Austin Scheck

a. Metrics and Weighting The science sensor selection portion of this endeavor required a two-step process in order to determine the baseline design. First, it was necessary to compare the capabilities of different types of sensors considered in Section 4.3. The sensors were juxtaposed using the metrics and weights provided below, in Table 20. The second step was to select a sensor model based on the type of sensor selected. That is presented in Section 3.2.10 with its own separate trade study.

Table 20. Trade Metrics for the Science Sensor Selection

Metric	Weight	Driving Requirement	Rationale
Power	0.14	1.3	Power is coming from an external battery in the 'top slice'. Though this allows for some room in terms of power usage, it is still best practice to use as little power as possible, in the event that future missions will be using power from the ThinSat bus. Because power is less of an issue, it has not been weighted the highest.
Size	0.17	1.1, 1.1.1	The ThinSat payload area is very small (54.8cm^2), and the team must be able to fit the payload computer, science sensor, and any thermal protection and structural support within the small area. This size constraint merits a higher weight to ensure that the sensor(s) do not violate the payload size requirement.
Survivability	0.17	3.1, 3.3	In order to have a successful balloon flight, the science payload must be able to withstand the extreme environments of near space. This means very low temperatures for a prolonged amount of time, and very low pressure. In order for mission success, the sensors need to be taking data and operating properly at all points of flight. It is very important that the science sensor is operational throughout the entire mission, as the data sent from the sensor will be sent as telemetry, verifying the COMM system capability. This metric was weighted higher because of this.
Cost	0.12	5.1	In order to allow as much budget as possible for the COMM system and the payload computer, the cost of the science sensor needed to be minimized. The weight was given based off of the fact that many sensors are not of concern when it comes to cost, though more expensive options may allow for better data resolution, lower power consumption, smaller size, etc.

Documentation	0.15	N/A	Documentation is key in understanding how the sensors work and how they can be integrated with other boards. This comes in handy when writing software for the sensor that allows it to interface with the payload computer. Documentation was rated high due to its importance in understanding how the science sensor will operate.
Scientific Merit	0.12	N/A	Raytheon has tasked the team with developing ideas for a mission that can be translated to a future space mission. This means choosing science that can be done on a balloon and in space. Though scientific merit is desired, the most important aspect of the mission is to generate data that can be sent as telemetry in order to verify and validate the COMM system, therefore, scientific merit was weighted lower.
Data Processing Requirements	0.13	2.3, 4.2.1, 4.2.2	The amount of data processing must be taken into account when selecting a sensor. Some types of sensors require more extensive on-board processing before data can be sent and received by the ground. The payload computer and the SDR must be able to handle the data rates from the sensor. The lower the data rates and the less on-board processing required, the simpler the scientific aspect of the mission becomes, and more focus can be placed on developing the payload computer and COMM system. This metric was moderately rated because while most sensors do not exhibit high values, there is a sizable range in data processing values for sensors.

b. Metric Quantification With the parameters for the trade study defined, it was necessary to determine the scoring subdivisions for each metric through an objective lens. The following table details what attributes constitute what scores for each of the metrics. The size category works around a maximum value of 12.5 cm², equal to half the total payload size. For realistic operations, this size acted as a feasibility ceiling value. Anything larger potentially compromises the space requirements of the thermal insulation and on-board computer. The power category is limited by the battery payload maximum, and thus, a "zero" score in that category represented anything outside the capabilities offered by the selected battery.

Table 22. Weights for Science Sensor Selection

Metric	0	1	2	3	4
Power	>100mA and/or greater than 5V	≤50 at 3.3V or 5V	≤10mA 3.3V or 5V	≤5mA 3.3V or 5V	<1mA 3.3V or 5V
Size	> 12.5cm ²	≤ 12.5cm ²	≤ 9cm ²	≤ 6cm ²	≤ 3cm ²
Survivability	Sensor can survive temperatures 0 Celsius and above	Sensor can survive temperatures -5 and above	Sensor can survive temperatures -10 Celsius and above	Sensor can survive temperatures -30 Celsius and above	Sensor can survive temperatures -40 Celsius and above
Cost	>\$300	≤\$300	≤\$200	≤\$100	< \$50
Documentation	Little to no documentation or product page only	Bare minimum details provided, such as size, mass, and power requirements	Datasheet available with at least absolute maximum operating values and schematic.	Detailed datasheet available	Highly detailed datasheets, software documentation, external resources available
Scientific Merit	Does not provide data that can be applicable in a space mission and/or is not relevant for a balloon flight		Data is applicable for balloon flight, may be able to be translated to a space mission		Data is directly translatable to a space mission and also gives valuable results for a balloon flight.
Data Processing Requirements	Requires on-board processor running at 200 MHz or greater and/or data requires extensive post-processing	Requires on-board processor running at 160 MHz or greater and/or data requires extensive post-processing	Requires on-board processor running at 130 MHz or above and/or requires extensive post-processing	Requires on-board processor running at 95 MHz or above and/or data may require some post-processing	Requires on-board processor running at 60 MHz or less and/or data requires minimal post-processing

c. Trade Results In conducting the trade study, both the imaging sensor and dosimeter received low scores, mostly due to their large size and power requirements, both relatively-important metrics. The environmental sensor came in very close to the magnetometer and IMU, however its applicability in space is limited and the sensor type was therefore not selected. Both the IMU and the magnetometer scored very close to each other, and since many IMUs can be purchased with a magnetometer built in, the IMU was selected as the team's primary science instrument. It only scored poorly on cost, mostly because the price range of IMUs that were considered varied widely. Additional information of specific IMU models is discussed in Section 3.2.10.

Table 23. Trade Study Results for Science Sensor Selection

Metric	Weight	Imaging Sensor/ Camera	Magnetometer	IMU	Environment Sensor	Radiation Sensor/ Dosimeter
Power	0.14	0	4	3	4	2
Size	0.17	1	3	4	2	0
Survivability	0.17	3	4	4	4	2
Cost	0.12	3	2	2	4	1
Documentation	0.15	2	4	4	4	0
Scientific Merit	0.12	4	3	3	1	4
Data Processing	0.13	0	3	3	4	3
Total	1	1.82	3.34	3.37	3.30	1.61

3.2.10. Science Sensor Model Selection

Lara Buri, Austin Scheck

From the science mission trade study, it was decided that an IMU would be used as the science sensor. This decision was made based on the fact that IMUs have a magnetometer, gyroscope, and accelerometer built in, and more data on the ThinSat's balloon flight could be taken using an IMU. This decision is explored further in Section 3.2.9. As a result, a secondary science sensor trade is conducted to choose an IMU model. The three sensors that were under consideration for this trade study are summarized below:

Table 24. Characteristics of Each IMU Sensor

	SparkFun IMU Breakout - MPU-9250	SparkFun 9DoF IMU Breakout - ICM-20948 (Qwiic)	Xsens MTi 1-Series
Current Draw	12 mA	3 mA	30 mA
Size	5 cm ²	5.88 cm ²	1.46 cm ²
Temperature Range	-40 to 85 C	-40 to 85 C	-40 to 85 C
Price	\$14.95	\$16.95	~\$200
Documentation	Good	Good	Good
Magnetometer Resolution	15000 nT	150 nT	25 nT

1. SparkFun IMU Breakout - MPU-9250

The SparkFun MPU-9250 contains a 9-axis MEMS sensor. The "Breakout" board houses two chips: the MPU-6500, containing a 3-axis gyroscope and a 3-axis accelerometer, and the AK8963, containing a 3-axis magnetometer. The MPU-9250 breakout houses 11 plated through-hole pins around the border of the PCB. These pins include the power interface and the I²C interface, which allows interface between this board and other boards, such as our payload computer. The MPU-9250 uses 16-bit analog-to-digital converters (ADCs) for digitizing all nine axes [23].

2. SparkFun 9DoF IMU Breakout - ICM-20948 (Qwiic)

The ICM-20948 features a 3-axis gyroscope and a 3-axis accelerometer with four selectable ranges, and a 3-axis magnetometer. The "Breakout" board contains a logic shifter and broken out GPIO pins. The chip itself is very low-powered with an I²C interface. The chip also uses a digital motion processor that offloads the computation of motion sensing algorithms from the detectors [24].

3. MTi 1-series

The MTi 1-series IMU features a 3-axis gyroscope, a 3-axis magnetometer, and a 3-axis accelerometer. It contains an on-board 'sensor fusion' algorithm that provides rudimentary data filtering. All sensors are calibrated and tested prior to shipment. These sensors are small in size, low in power consumption, and have I²C interfacing. The sensor can be bought with a development kit that is Arduino compatible and comes with a free download of the MT software suite, which can be used to program the MTi 1-series [25].

Table 25. Trade Metrics for the Sensor Model Selection

Metric	Weight	Driving Requirement	Rationale
Power	0.15	1.3	Though the IMU sensors chosen met the trade study metrics for the previous trade study, power was kept as a metric in order to emphasize the importance of power mitigation.
Size	0.14	1.1, 1.1.1	The sensors chosen are all very small in size, however this metric was retained to emphasize the importance of having small sensors for such a small payload area.
Survivability	0.14	3.1, 3.3	The sensors chosen must be able to survive the low temperatures at 30km altitude.
Cost	0.17	5.1	In order to allow as much budget as possible for the COMM system and the payload computer, the cost of the sensor should be minimized. Cost was given a higher weight due to the fact that the team wanted to save as much budget as possible for purchasing higher-impact items such as the SDR and payload computer supplies.
Documentation	0.2	N/A	Documentation is key in understanding how the sensor works and how it can be integrated with other boards. This comes in handy when writing software for the sensor that allows it to interface with the payload computer. Documentation was rated high due to its importance in understanding how the IMU sensor will operate. Though there are many options for IMUs, some may have better documentation than others, and that should be considered when purchasing.
Magnetometer Resolution	0.2	N/A	The resolution of the IMU is important to consider when choosing the sensor. The magnetometer resolution was chosen as the metric due to the fact that each datasheet for each sensor under consideration had consistent ratings for resolution. A higher resolution sensor will give more accurate results and therefore more valuable data. The highest weight was placed on resolution due to the fact that most of the IMU sensors already fit the criteria set in the previous trade study, giving them all similar point distributions in each category. This metric made distinguishing the models far easier.

a. Metrics and Weighting

Table 26. Weights for Sensor Model Selection

Metric	0	1	2	3	4
Power	$\geq 100\text{mA}$ and/or greater than 5V	$\leq 50\text{mA}$ at 3.3V or 5V	$\leq 10\text{mA}$ 3.3V or 5V	$\leq 5\text{mA}$ 3.3V or 5V	$<1\text{mA}$ 3.3V or 5V
Size	$> 12.5\text{cm}^2$	$\leq 12.5\text{cm}^2$	$\leq 9\text{cm}^2$	$\leq 6\text{cm}^2$	$\leq 3\text{cm}^2$
Survivability	-10 Celsius and above	Sensor can survive temperatures -20 and above	Sensor can survive temperatures -30 Celsius and above	Sensor can survive temperatures -40 Celsius and above	Sensor can survive temperatures -50 Celsius and above
Cost	$>\$300$	$\leq \$300$	$\leq \$200$	$\leq \$100$	$<\$50$
Documentation	Little to no documentation or product page only	Bare minimum details provided, such as size, mass, and power requirements	Datasheet available with at least absolute maximum operating values and schematic.	Detailed datasheet available	Highly detailed datasheets, software documentation, external resources available
Magnetometer Resolution	$>77.5\text{ nT}$	$<77.5\text{ nT}$	$<55\text{ nT}$	$<32.5\text{ nT}$	$<10\text{ nT}$

b. Metric Quantification

Table 27. Trade Study Results for Sensor Model Selection

Metric	Weight	SparkFun IMU Breakout - MPU-9250	SparkFun 9DoF IMU Breakout - ICM-20948 (Qwiic)	Xsens MTi 1-Series
Power	0.15	2	3	1
Size	0.14	4	4	4
Survivability	0.14	3	3	3
Cost	0.17	4	4	2
Documentation	0.2	4	4	4
Magnetometer Resolution	0.2	0	0	3
Total	1	2.76	2.91	2.87

c. Trade Results For the first metric, power, the Xsens MTi 1-Series IMU scored the lowest. This was due to the fact that the sensor can draw up to 30 mA, which was on the high end of our scale. The SparkFun MPU-9250 received a score of two as it can draw up to 12 mA. The SparkFun ICM-20948 received a score of three as it drew the least current of the three sensors.

All three IMUs considered received fours in size as they were all very small ($< 5\text{cm}^2$) and could have easily fit on a small PCB that would not take up too much payload space.

Again, all three IMUs considered received a 4 for documentation. Each sensor had very a thorough datasheet which came with some kind of code documentation as well.

The resolution of the sensor was very important, as the team sought accurate results. Neither SparkFun sensor had a resolution anywhere near as good as the Xsens IMU. The resolution on the Xsens magnetometer

was hundreds of times better than the magnetometers on the SparkFun IMUs. Therefore, both SparkFun sensors received a zero in this category while the Xsens IMU received a 3.

3.3. Selection of Baseline Design

3.3.1. Communications Hardware

a. Software Defined Radio Based on the results of the SDR trade study, the Lime SDR Mini was chosen. While the Lime SDR Mini and the USRP B200mini had similar scores, several key factors made the Lime SDR Mini the best choice. For one, it has a much higher bandwidth than the USRP B200mini. This allows for the potential for higher data rates when receiving telemetry if paired with an amplifier. Also, the Lime SDR has a minimum operating temperature of -40°C , compared to 0°C with the USRP B200mini. Since the minimum temperature expected during the balloon flight is -56°C , the Lime SDR will require a less robust thermal control system, which simplifies the design. The cost of the USRP B200mini also poses a challenge. The cost of the 2 SDRs alone would take up 30% of the budget, and that doesn't include additional necessary hardware like antennas or amplifiers. While one of the strengths of the USRP B200mini over the Lime SDR is its transmission gain, the use of an amplifier can mitigate this shortcoming. Since each of these options would require an amplifier anyways, the effect of the lower gain will be minimal. Finally, while the USRP B200mini has better integration with MATLAB and can also communicate using protocols defined in GNU Radio (compared to only GNU Radio protocols being supported on the LimeSDR), the extensive documentation of GNU Radio will help mitigate the learning curve associated with using the unfamiliar software. Overall, since the LimeSDR performs better in nearly every other category, its selection over the USRP B200mini was obvious. Further, since GNU Radio will be implemented on the flight SDR anyway given the more powerful OBC, implementing it on the ground station SDR should present minimal difficulty.

3.3.2. Communications Software

3.3.3. Printed Circuit Board

The payload space on the Thinsat bus is too small to simply purchase and use an AM335x-based development board like the BeagleBone Black. Instead, a custom PCB must be designed and manufactured to conform to the shape of the payload space and include all of the necessary hardware for system operation. To design this board, we are using the Altium PCB designer software, which includes all of the functionality necessary to develop a PCB schematic, move it to a layout, and generate the design files for manufacturing.

3.3.4. Flight Software and On-Board Computer

After performing a trade study on the MCU to be used for the on-board computer, the AM3358 was chosen. This MCU requires more power than the others; however, it comes with a major advantage of being powerful enough to run a high-level operating system such as KubOS and many other Linux distributions, which none of the other MCU options are capable of doing. This drastically reduces the software design complexity because it allows the use of higher-level programming, which abstracts from the hardware and kernel, and avoids the difficulties that come with low-level embedded programming. The high clock rate of this MCU, along with its multi-core architecture makes it much easier to meet the real-time processing requirements needed to control all of the subsystems successfully. The MCU has plenty of GPIO pins, which will ensure that all of the required interfaces will have connections with the processor. Furthermore, the MCU is included in the Beaglebone Black development board. This development board can be purchased and used to quickly and easily begin prototyping. Since the AM3358 allows the use of Linux and high-level programming, there is more flexibility with what IDE is chosen and what programming languages are used to develop the flight software. Software may be developed with multiple IDEs, so each team member may choose the one that they are most comfortable with. This is the same case for what programming languages

are used. The flight software may be composed of software written in multiple languages thanks to the Linux kernel that will translate instructions for the hardware.

3.3.5. Ground Software

After considering a few options for ground software programs, it was decided that COSMOS fit the necessary requirements best. The program is very powerful with an easily learnable user interface. It has various breakdowns of tool suites, each containing applications to provide ground software function. These suites include Realtime Telemetry Visualization Tools, Realtime Commanding and Scripting Tools, Configuration Tools, and the Command & Telemetry Server. Different applications from each of these tool suites can be used to perform necessary functions for the ThinSat mission.

The Command & Telemetry Server acts a middle man between the external target embedded system and the specific COSMOS tool being used. It processes command packets before sending them off, it receives telemetry and sends it to the correct tool, and it has the ability to log realtime data into a directory in the host computer.

The Command Sender application creates a command (or uses a previously created command) and sends it to the Command & Telemetry Server to be processed and forwarded to the ThinSat. The command packets are formatted according to configuration files, so they can take any form desired, such as CCSDS.

The Telemetry tool suite depacketizes raw telemetry packets forwarded from the Command & Telemetry Server, and applies the desired applications from the tool suite. The Telemetry Grapher tool takes in the depacketized telemetry and plots it in realtime to give a visual of the data being stored. The other important tool in the telemetry suite is the Telemetry Extractor tool. Because the telemetry data logging is done directly by the Command & Telemetry Server, the data has not been depacketized, and so it is stored in a directory in a packetized .cpg data format. The Telemetry Extractor tool takes in these .cpg files and, according to its configuration files, extracts the data into a text file in the same directory.

COSMOS is a very powerful, open source program, that has the applications and capability to fully fulfill the ground software requirements of this mission.

3.3.6. Thermal

Lara Buri

Due to a limited power budget and the fact that Near Space Launch has flown ThinSats with no thermal control or protection, it was decided the TOMCAT mission would use thermal protection (insulation) to maintain internal temperatures at the required levels. This is the simplest solution for thermal design and does not place more constraints on battery selection and complications on the payload computer design or software design.

When choosing an insulation type, the team had to decide between MLI (Multi-Layer Insulation), which is commonly used in orbital space missions, and a foam-type insulation. The following chart was used to make this decision:

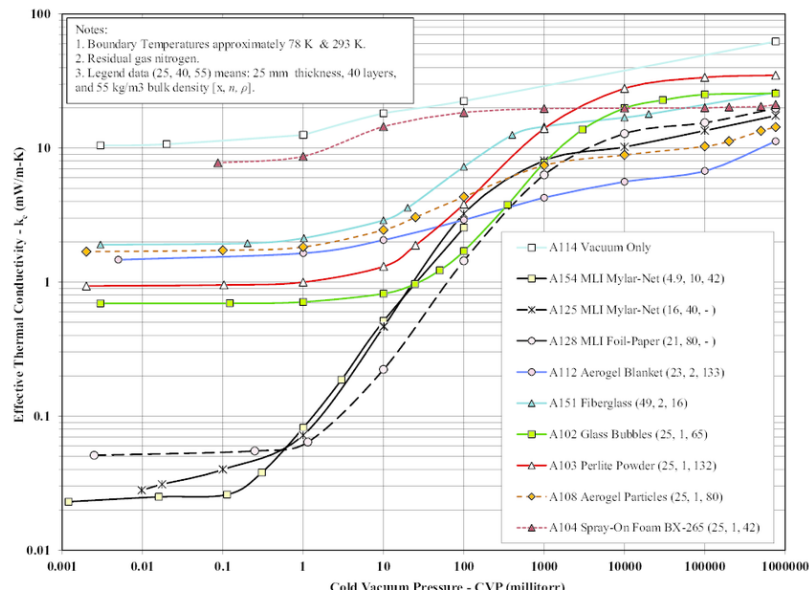


Figure 12. Insulating Materials at Low Pressures [6]

The atmospheric pressure at the maximum balloon flight height of 30km is about 200 Pascals, or 1500 millitorr. Figure 12 shows that MLI performs best at pressures around .01 to .1 millitorr, pressures more indicative of Low Earth Orbit or deep space. At the pressure endured by TOMCAT mission during flight, MLI and foam insulators perform similarly, with thermal conductivities around $.01 \frac{W}{m \cdot K}$. Foam insulation is much cheaper than MLI and can be bought easily at local hardware stores, whereas MLI must be purchased from specific vendors and lead times may be higher due to shipping. For these reasons, it was concluded that a foam insulation would be used for thermal protection. The type and thickness of the foam insulation will be determined through thermal modeling and analysis, presented in later sections.

3.3.7. Power

Jake Schroeder

When choosing the specific battery to use, the TOMCAT team considered the voltage and capacity available from the battery, its operation in the low temperatures experienced during the high altitude balloon flight, component safety, and power regulation and distribution.

Lithium Ion batteries were chosen by the TOMCAT team because of their large power capacity, wide availability in numerous sizes from reputable sources shipping from within the United States. This large body of research gave the team confidence in their applicability and usefulness in a high altitude balloon flight. Lithium Ion batteries are widely used commercially and have one of the largest bodies of research surrounding them, with extensive information on their operation in low temperature environments and at different discharge rates.

Lithium Polymer batteries were not chosen primarily because of size constraints and increased difficulty in finding reputable sources offering batteries with very specific capacity and volume requirements. Lithium Polymer cells are often rectangular, which would allow for the team to maximize the volume for batteries inside the ThinSat. However, the majority of cells available didn't allow enough room for the communications system, or were overall too large for their given capacities. Furthermore, the cells that did meet our constraints weren't readily available from reputable companies with shipping within the United States.

Alkaline batteries were not chosen primarily due to their inability to be recharged and fixed dimension, which did not fit properly within the payload volume available. While Alkaline batteries are flight proven in high altitude balloon missions and have ratings down to much lower temperatures than typical rechargeable

batteries, the inconvenience of not being rechargeable would require new batteries for every test. Additionally, at least 5 9V batteries would be required to power the mission for 2 hours, and would not fit in the payload volume available.

For these reasons, the rechargeable Lithium Ion batteries offered the greatest flexibility in terms of being able to be customized to fit within the available payload space, had a high power density to volume ratio, and were capable of operating at the low temperatures expected in a high altitude balloon flight, with a large body of research behind them to show their feasibility in low temperature environments.

3.3.8. Structures

Matthew McCallum

The ThinSat structure is composed of two frames. The "lower" (from the perspective of Figure 13) frame, as it will now be referred to, is the frame that will be purchased from NearSpace Launch Inc. (NSL). Half of this frame will be taken up by hardware from NSL, and the other half is the payload space. The payload space will contain our payload board. The payload board is a custom PCB that will have the OBC, IMU, and power distribution. The "upper" (from the perspective of Figure 13) frame contains the rest of our hardware. The battery will be above the payload space, and will partially extend into the lower frame due to its thickness. On the other side of the upper frame is the SDR and signal processing components. In Figure 13, the SDR is beneath another custom PCB. This custom PCB will contain signal processing hardware to filter the incoming signals before they are sent to the SDR. There is also a Low Noise Amplifier (LNA) in the upper frame to amplify the signal to ensure that the SDR can read the incoming signal.

The dimensions of the ThinSat are 114.2 mm in length, 111.1 mm in width, and 36.3 mm in height. (The height in Figure 13 does not include the top cover.) These dimension are set by NSL to be able to interface with their hardware. The whole ThinSat is expected to weigh 569 grams. It needs to be under 800 grams for the Gateway to Space balloon flight, so there is a good margin there for any unexpected weight additions. The frame is designed to be able to fit all of the hardware within the volume of the 2T ThinSat configuration. It is also designed to be able to keep all components in place during testing, throughout the balloon flight, and when subjected to impact with the ground at the end of the balloon flight.

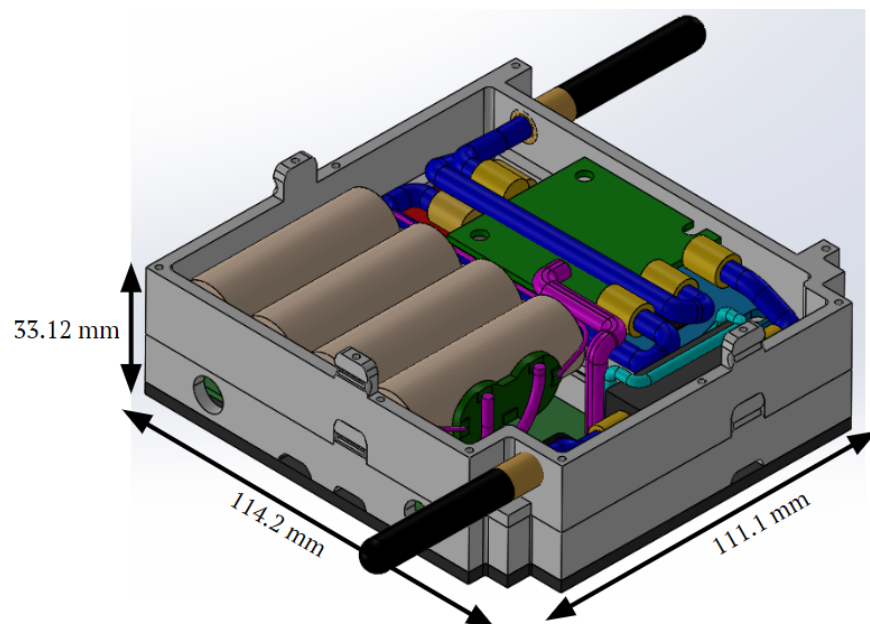


Figure 13. CAD model of the fully assembled 2T ThinSat

3.3.9. Science Sensor

Austin Scheck

Selecting a scientific sensor was a two-step process that first involved choosing a type of scientific sensor before selecting a particular model to integrate. Both steps are detailed below.

a. Science Sensor Type From the results presented in Table 23, it was clear that imaging sensors/cameras and radiation sensors/dosimeters were not ideal choices for scientific sensors for this project. This was primarily derived from the amount of power and size they would take up in the ThinSat, as well as a lack of documentation of each, which would be integral in understanding how they would interface with the payload board.

That left three sensor types to choose from: magnetometers, IMUs, and environmental sensors. Nonetheless, all received very similar scores at the end of the trade study. Based on the overall score, it was decided that the science sensor would ultimately be an IMU. This decision was based on the fact that 9-axis IMUs have built-in magnetometers, and the additional data from the gyroscope and accelerometer could give useful data to characterize the balloon flight. An IMU could be used in future space missions to send and receive data to and from an attitude determination and control system, which is very important for satellites in Earth orbit to measure the state of the spacecraft. For these reasons, an IMU was chosen as the type of science sensor for the TOMCAT mission.

b. Science Sensor Model A second trade study was done to determine the IMU sensor model. The results of this trade are shown in Table 27, and all IMUs scored similarly. When researching IMU sensor options, many of the sensors were similar in size, survivability, and documentation. The most expensive option was the Xsens MTi 1-series IMU, which can cost up to \$200. This steep price results in better magnetometer resolution and overall improved sensor performance. The resolution of the magnetometer on the Xsens IMU was hundreds of times better than that of the SparkFun IMUs.

As a result of its high-fidelity data and fair price range within the project budget, the Xsens 1-Series IMU was selected by the team to be the TOMCAT science sensor for the mission. The accuracy of its acceleration, rotational, and magnetic data far surpasses its competitors and has the potential to provide Raytheon with valuable, insightful, and exploitable information for both balloon flight missions and operations in low Earth orbit.

4. Requirements Development

Katie Steward

After determining the functional requirements of the project, the team broke each one down into sub-requirements that have to be met in order for the functional requirement to be met. The requirements flowdown is given in the following spreadsheet:

#	##	###	Requirement	Motivation	Validation
1			The payload shall interface with the bus according to the Thinsat Interface Control Document (ICD)	Customer has mandated the use of the ThinSat bus	System Demonstration
	1.1		The designed payload board shall fit within the ThinSat payload area as defined by the ICD	Customer has mandated that the SDR, comms hardware, and power system do not need to be in the payload area, but the other hardware components will	The components fit into the ThinSat bus
		1.1.1	A 0.25 mm tolerance shall be used when designing parts to fit within the ThinSat bus area	Conform to ThinSat ICD	The components fit into the ThinSat bus
	1.2		The designed payload board shall have a mass less than 55 grams	Conform to ThinSat ICD	Inspection of mass
	1.3		The designed payload shall interface with the bus computer via the provided payload connector	Conform to ThinSat ICD	The payload computer will interface with the bus computer via a flatsat test
		1.3.1	The payload shall be designed with a DF12(3.0)-20DS-0.5V (86) receptacle	Conform to ThinSat ICD	The components fit together with the provided ThinSat bus port
	1.4		The designed payload shall conform to the 'prohibited items' list as provided in the ICD	Conform to ThinSat ICD	Inspection of data sheets to ensure no component explodes or outgasses during flight
2			The payload shall communicate with the ground system during the balloon flight	Customer has mandated that the ThinSat must be commanded and must send telemetry to the ground during the balloon flight	System Demonstration
	2.1		The payload SDR shall be compatible with the ground SDR	The SDRs must be compatible in order to transfer commands and telemetry	Bench test with SDRs communicating through serial connection
		2.1.1	The ground SDR shall send and receive formatted data bits to/from the payload SDR over a wired connection	The wired connection allows for early C&DH testing	Bench test with SDRs communicating through serial connection
	2.2		The payload SDR shall wirelessly communicate with the ground SDR at a range > 30km	Verify that wireless communication works in preparation for the balloon flight	Long range wireless communication test
		2.2.1	The ground SDR shall wirelessly send and receive formatted data bits to/from the payload SDR	The SDRs must be able to communicate at a distance for the ThinSat to be commandable on a balloon	Bench test with SDRs communicating through wireless connection
		2.2.2	The payload SDR shall wirelessly send and receive formatted data bits to/from the ground SDR	The SDRs must be able to communicate at a distance for the ThinSat to be commandable on a balloon	Bench test with SDRs communicating through wireless connection
	2.3		The SDR shall operate at in ISM frequency band	Payload must use a carrier frequency that complies with the FCC, which is easiest in the amateur bands	Datasheet verification
	2.4		The SDR shall have an uplink rate such that the payload can receive at least one command packet every 10 seconds	The payload must be commandable. Since the data rate can only be determined from a full link budget analysis, the exact data rate is not chosen here	Bench test with SDRs communicating through wireless connection
	2.5		The SDR shall have a downlink rate such that the ground SDR can receive one telemetry packet every 10 seconds	The ground must receive telemetry. Since the data rate can only be determined from a full link budget analysis, the exact data rate is not chosen here	Bench test with SDRs communicating through wireless connection

#	##	###	Requirement	Motivation	Validation
3			The payload shall survive and operate in the environment during all stages of the balloon flight	Team needs to deliver completed product to the customer at the conclusion of the project	System Demonstration
	3.1		The payload shall be able to withstand the temperature range of flight, which includes the coldest point in the flight (-56° C), and a hot day on the ground (38°C)	The payload must work during testing and flight	Turn-on test in a dry-ice cooled cooler, turn-on test in the lab
	3.2		The payload shall withstand the impact from a balloon launch (estimated 9 m/s from vendor)	The payload must still operate after the balloon flight	Drop test from height that mimics 9m/s, followed by a power on test
	3.3		The designed payload shall operate between near-vacuum conditions of peak flight (~0.01 atm) and standard atmospheric pressure	The payload must survive and be usable after flight	Datasheet verification
	3.4		The IMU shall generate rate telemetry during ascent and descent	Telemetry must be generated so that it can be sent to the ground to prove that the C&DH system works	Lab test with IMU moving while on
		3.4.1	Science data shall be able to be collected and updated at least once every second	The telemetry must update fast enough that a model with reasonable resolution can be created	Software test
4			The On Board Computer (OBC), SDR Software, and Flight Software (FSW) shall perform all necessary software tasks for the balloon flight	Each part of the payload must be controlled to ensure that all parts work together to take and send science data during the balloon flight	System Demonstration
	4.1		The payload shall be commandable	Overall goal of the project	Bench test
		4.1.1	The SDR software shall send all command packets it receives to the FSW	Received commands must go to the FSW for processing	Software test
		4.1.2	The FSW shall decode all commands received in CCSDS format	CCSDS format is mandated by the customer	Software test
		4.1.3	The FSW shall recognize errors in command formatting and return an error code	Formatting errors should be noted so that they can be fixed	Bench test with bad commands
		4.1.4	The FSW shall generate command counts for commands received in the correct format	Commands counts show that commands made it to the bus	Bench test with good commands
		4.1.5	The FSW shall generate command counts for commands with formatting errors	Command error counts show that commands made it to the bus, but that there was something wrong with them	Bench test with bad commands
		4.1.6	The FSW shall forward IMU commands to the IMU	Must be able to command the sensor suite	Bench test with sensor suite commands
	4.2		The payload shall generate telemetry	Overall goal of the project	Bench test, balloon flight
		4.2.1	The OBC shall receive raw telemetry from the IMU	The FSW should process the data that the sensor suite collects	Bench test
		4.2.2	The FSW shall packetize all (payload and health) telemetry in CCSDS format	CCSDS format is mandated by customer	Software test
		4.2.3	The FSW shall send packetized telemetry to the SDR software	Telemetry must be sent to the SDR for downlink	Software test
	4.3		The OBC shall store all telemetry during flight	Allows the team to validate downlinked data with stored data	Balloon flight
		4.3.1	The OBC shall store IMU telemetry	Science data can be validated	Software test
		4.3.2	The OBC shall store payload health and safety telemetry	Safety data can be validated	Software test
		4.3.3	The OBC shall store command counts	Payload receipt of commands can be validated	Software test

#	##	###	Requirement	Motivation	Validation
5			Administrative requirements	Capture requirements that are non-technical in nature but still critical to the success of the project	Bookkeeping
	5.1		All purchased components must cost in total equal or less than the amount provided by the Aerospace Engineering Sciences department and other engineering funds obtained by the team	The project cannot cost more money than the team is given	The total cost at the end does not exceed the allocated funds
	5.2		The project must be FAA/FCC compliant	The project must be allowed to legally perform the balloon flight	Necessary permissions are given and limitations are met
	5.3		Any software used or generated by the project team shall be compatible with typical computer operating systems	The software components of the project need to be easy to manipulate and use	Software documentation review

4.1. Requirement 1.1

The primary ThinSat frame has a small area for electronics that should be filled with the payload board. The payload board contains the microcontroller and the inertial measurement unit. The other hardware components are too large to fit in this small area, so the customer decided that it was not mandatory for everything to fit in this area.

4.1.1. Requirement 1.1.1

The ThinSat ICD gives requirements for manufacturing components that the team will need to conform to.

4.2. Requirement 1.2

The ThinSat ICD recommends that the payload in the payload space should have a mass of less than 55 grams. Since only the payload board will be in this area, the other hardware components do not need to be contained within the 55g requirement.

4.3. Requirement 1.3

In mission development after the project, the TOMCAT payload should be able to send telemetry and receive commands from two communications systems– the TOMCAT system and the ThinSat system. In order to communicate with the ThinSat system, the TOMCAT payload needs to connect with the payload connector already on the ThinSat bus. Since the TOMCAT team will not have access to the ThinSat system in the project, the connection only needs to be established so that another team can make the software connection.

4.3.1. Requirement 1.3.1

The project needs to use the specific connector used on the primary ThinSat bus.

4.4. Requirement 1.4

For project safety, no part of the payload can explode or out-gas during flight, which is specified in the ICD.

4.5. Requirement 2.1

In order for the TOMCAT ground station to be able to communicate with the on-board communication system, both software defined radios in the systems need to be able to communicate with each other.

4.5.1. Requirement 2.1.1

At a base level, the SDRs need to be able to send data back and forth. The data that will be sent are commands, which at a basic form are just bits. If this can be performed over a bench test with the SDRs wired together, it shows that the system software works.

4.6. Requirement 2.2

The software defined radios need to be able to send data back and forth over a wireless connection in order to move towards a balloon flight. The maximum altitude of the balloon flight is 30km, so the radio system should be designed to function at least at that range.

4.6.1. Requirement 2.2.1

Over this wireless connection, the ground radio should be able to send commands or other formatted data files to the on-board radio.

4.6.2. Requirement 2.2.2

Over this wireless connection, the on-board radio should be able to send commands or other formatted data files to the ground radio.

4.7. Requirement 2.3

In an effort to avoid the necessity of a radio license, the team chose to operate in an open band that is available to everyone. Additionally, the team wanted to be at a frequency near the range used in space missions. These two requirements lead the team to operate in an ISM band.

4.8. Requirement 2.4

In order for the communication system to be successful, the radio system needs to be able to send commands from the ground to the satellite, even at a slow rate. The team decided that a reasonable, low end uplink rate should allow the system to send commands at least once every second. This allows the system to be successful at a low rate, but work to exceed the requirement.

4.9. Requirement 2.5

In order for the communication system to be successful, the radio system needs to be able to send telemetry packets from the satellite to the ground, even at a slow rate. The team decided that a reasonable, low end downlink rate should allow the system to send packets at least once every second. This allows the system to be successful at a low rate, but work to exceed the requirement.

4.10. Requirement 3.1

In order for the satellite to survive a balloon flight, it needs to withstand the temperature range of flight. This includes a high end for a hot day on the ground as well as a low end of the lowest temperature estimated on the 30km altitude balloon flight.

4.11. Requirement 3.2

In order for the satellite to be recoverable, it needs to survive the impact after the balloon flight.

4.12. Requirement 3.3

In order for the satellite to survive a balloon flight, it needs to function at the low pressure of the altitude-balloon flight.

4.13. Requirement 3.4

In order to verify the communication system, the satellite needs an instrument to generate verifiable data. Thus, the inertial measurement unit needs to generate telemetry during all parts of the balloon flight.

4.13.1. Requirement 3.4.1

In an effort to generate enough telemetry for a packet to be sent every 10 seconds, the team estimated that the IMU should generate telemetry at least once a second.

4.14. Requirement 4.1

In order to verify the communications system, the payload needs to be able to receive and execute commands.

4.14.1. Requirement 4.1.1

After the signal is received by the signal processing software and returned to packet format, it should forward all data it receives to the flight software to be decoded and further processed so that a command can be executed.

4.14.2. Requirement 4.1.2

By customer request, the team needs to use standard CCSDS packet protocol for all of its command and telemetry packets. Thus, the flight software needs to be able to decode one of these packets to understand what command has been sent from the ground.

4.14.3. Requirement 4.1.3

In an effort to have good awareness of if there are too many bit flips in uplink, the flight software should be able to recognize a poorly formatted CCSDS packet or poorly formatted command and give an error.

4.14.4. Requirement 4.1.4

In order to characterize the communications system, the flight software needs to generate a count of the number of good commands it receives and executes.

4.14.5. Requirement 4.1.5

In order to characterize the communications system, the flight software needs to generate a count of the number of bad commands it receives and rejects.

4.14.6. Requirement 4.1.6

The inertial measurement unit is the final destination for a command, so the flight software needs to be able to receive a command and format it such that the inertial measurement unit can understand it and react appropriately.

4.15. Requirement 4.2

In order to verify the communication system, the satellite needs to be able to generate telemetry that can be downlinked.

4.15.1. Requirement 4.2.1

The inertial measurement unit will generate telemetry, so the microcontroller needs to be able to receive that telemetry from the IMU.

4.15.2. Requirement 4.2.2

All telemetry that the flight software receives needs to be formatted in CCSDS format per customer request.

4.15.3. Requirement 4.2.3

The flight software needs to forward all telemetry packets to the signal processing software so that the packets can be modulated and downlinked.

4.16. Requirement 4.3

In the event that the communication system fails, the satellite should store all data so that when it is retrieved, the team can debug the problem. Additionally, if the data is stored, the team can validate the downlinked data with the stored data.

4.16.1. Requirement 4.3.1

To store all data, the microcontroller needs to store the inertial measurement unit science data.

4.16.2. Requirement 4.3.2

To store all data, the microcontroller needs to store all state of health data.

4.16.3. Requirement 4.3.3

To store all data, the microcontroller needs to store the command counters that the flight software generates.

4.17. Requirement 5.1

The project needs to fit within the budget. At a minimum, this can be the \$5000 provided by the department. at a maximum, this can be the \$5000 from the department, the \$3000 from Raytheon for extra funding, and the \$3000 from the Engineering Excellence Fund.

4.18. Requirement 5.2

In order for the project to be able to be successful, it needs to be legally able to perform the balloon flight, both in terms of the communications frequency and power and the balloon flight logistics.

4.19. Requirement 5.3

All software from the project should be able to run on someone's computer who would like to pick up the project after the team has completed it.

5. Detailed Design

5.1. Communications Hardware

Lauren DeMoudt, Trevor Weschler

All of the communications hardware is designed to satisfy functional requirement 2 and all flow down requirements below it.

5.1.1. Link Budget

Trevor Weschler

The link budget is a critical part of the design that ensures that the communication between the ground station and Thinsat can be achieved. The goal is to account for all of the gains and losses in the system to determine if a positive link margin can be achieved. A positive link margin means that the signal is powerful enough to be received at the intended destination. To do this, the signal must overcome losses in propagation and remain a signal level higher than the noise encountered by the system. The first part of the link budget includes general parameters that were derived from design requirements or choices.

Table 28. TOMCAT Link Budget: General Parameters

Parameter	Uplink (Command)	Downlink (Telemetry)	Units
Frequency	0.915	0.915	GHz
Wavelength	0.328	0.328	m
Range	104	104	km

Table 28 shows the general parameters upon which the link budget is built on. As discussed previously, the chosen uplink and downlink frequency is 0.915 GHz. The reasoning is that since this frequency lies in an ISM band, we don't need to get a radio license to use this frequency. Additionally, it has a 26 MHz bandwidth which is more than necessary for TOMCAT purposes. Bandwidth is related to how much data can be encoded on the transmitted analog signal. Since TOMCAT plans to use the same frequency for both uplink and downlink, the communications system will be operated in half-duplex mode. This means that the system will not transmit and receive at the same time. The wavelength parameter is calculated from the chosen frequency.

The range parameter is determined by the maximum altitude that the balloon is expected to reach and the horizontal drift of the balloon. The maximum altitude expected is 30 km. This is a typical altitude of the Gateway to Space balloons. The maximum horizontal drift expected is 100 km. This value is based on historic flights and NOAA information on high altitude weather balloon drift. The worst case scenario for the link budget occurs when the balloon has drifted 100 km and is at the peak altitude of 30 km. The link budget will be computed at this worst case scenario for a range of 104 km (the hypotenuse of the triangle formed by 30 km height and 100 km drift).

Table 29. TOMCAT Link Budget: Data Parameters

Parameter	Uplink (Command)	Downlink (Telemetry)	Units
Bit Error Rate	10^{-6}	10^{-5}	[-]
Modulation Scheme	DPSK	DPSK	[-]
Required Eb/No	11.8	10.6	dB
Data Rate	256	144,000	bps
Pr/No	35.86	62.18	dB-Hz
Design Margin	3	3	dB
Minimum Pr/No	38.88	65.18	dB-Hz

Table 29 shows the data parameters of the link budget. These represent expectations of how TOMCAT will handle the data and the requirements that fall out of those design choices. Since TOMCAT has no customer requirements for bit error rate, suggested values from SMAD were used. These values are 1 in every million bits on the uplink are in error while 1 in every 100,000 bits are in error on the downlink. This is a key parameter because it is one of the criteria that will be used for testing during verification.

Modulation scheme is the method by which data is encoded onto an analog radio signal to be broadcast. The modulation scheme that was chosen for TOMCAT is differential phase shift keying. The reasoning is

that this scheme is fast and accurate. This choice drives the next parameter, required E_b/N_o , or ratio of received energy per bit to noise density. This value can be interpreted as how powerful the signal must be relative to the noise in order for the signal to be received. To determine the required E_b/N_o , SMAD can be used. Using fig. 14, the required E_b/N_o can be computed using the selected bit error rates and modulation scheme.

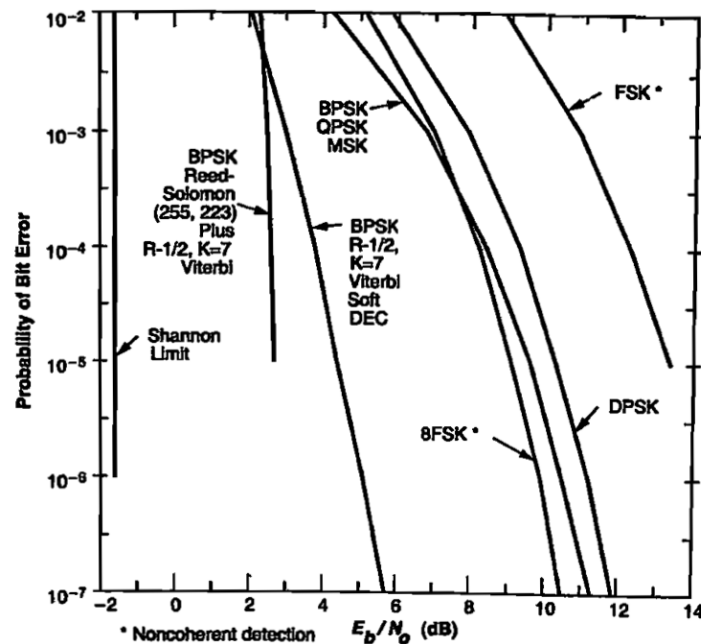


Figure 14. Required E_b/N_o SMAD (SMAD)

The data rate parameter is based on design choices for TOMCAT. Each command is expected to be a maximum of 64 bits long. A design choice was to be able to command at least twice per second. This means uplinking at a minimum of 128 bits per second. However, since the system is operating in half-duplex mode, the uplink multiplied by two again resulting in an uplink rate of 256 bits per second. The downlink is driven by how much data is produced by the IMU. In order to downlink all of the accelerometer data real time, a rate of 72,000 bits per second is needed. Once again, this value must be multiplied by two to account for operating in half-duplex mode bringing the total downlink rate to 144,000 bits per second.

The next parameter P_r/N_o (carrier to noise ratio density) is the main parameter used to compute the link margin. P_r/N_o can be understood as the signal power divided by noise. The difference from E_b/N_o is that P_r/N_o deals with the signal instead of individual bit energy. It is found by multiplying the required E_b/N_o by the data rate. Performing this computation for both the uplink and the downlink provides the values in Tbl. 29. The design margin is required because real systems do not follow Fig. 14 perfectly due to system imperfections. Adding a design margin of 3 dB onto the P_r/N_o gives the minimum P_r/N_o . This value is the P_r/N_o that the system must be able to produce for the signal to reach the destination with enough power to be received.

Table 30. TOMCAT Link Budget: Noise Parameters

Parameter	Uplink (Command)	Downlink (Telemetry)	Units
Receiver Antenna Noise Temperature	873.03	97.7	K
Receiver Cable Loss	0.5	0.5	dB
Receiver Noise Figure	2	2	dB
Receiver Noise Factor	1.6	1.6	[-]
Receiver Noise Temperature	290	290	K
Receiver System Noise Temperature	1098.73	323.40	K
Receiver System Noise Power	-198.19	-203.5	dBW-Hz

Table 30 walks through the noise parameters that must be overcome by the link design. One common way to represent noise is in terms of degrees Kelvin. It is important to note that this is not the temperature of the item being discussed, but rather a measurement of the noise at that component. In general, the noise is usually divided into two parts: the antenna noise temperature and the receiver noise temperature.

The receiving antenna noise temperature is the noise encountered at the antenna. For the uplink, this is the antenna on board and for the downlink, this is the antenna on the ground. The downlink includes noise from the business area, galactic noise, black body noise, Earth noise, and PCB noise. The uplink includes noise from the business area, galactic noise, and black body noise. Note the downlink noise is the most uncertain portion of the link budget and is part of the reason the link budget is so large.

The receiver noise temperature is further divided into two parts: the cable loss and then actual receiver noise. The receiver cable loss is computed based on the choice of cable that TOMCAT is using. Since the same cables are used for the uplink and downlink, the cable loss is the same. The receiver noise figure is a given value in the datasheet for the LimeSDR Mini. The receiver noise temperature is determined by the multiplying the reference temperature by the noise figure minus 1. The typical reference temperature is 290 K. Since the noise figure is 2, the receiver noise temperature is also 290 K.

The final parameter in this section is the receiver system noise temperature. This is the combination of both the receiver antenna noise temperature and receiver system noise temperature. Once again, the downlink value is the most uncertain part of the budget. This is mostly due to the effects of the Sun on the noise. If the balloon drifts close to the Sun, the noise becomes much greater. However, away from the Sun, the noise contribution is relatively modest (about 0.5 dB). Another uncertainty is how much noise will be encountered from other use in the band. This was taken into account with values recommended from SMAD, but more noise may be encountered since the band is open. This uncertainties are mitigated by having a large link margin.

Table 31. TOMCAT Link Budget: Receiver and Propagation Parameters

Parameter	Uplink (Command)	Downlink (Telemetry)	Units
Receive Antenna Gain	2.2	8.5	dBi
Receiver Cable Loss	0.5	0.5	dB
Space Loss	132.04	132.04	dB
Atmospheric Attenuation	0.45	0.45	dB
Polarization Loss	0.2	0.2	dB

Table 31 shows the receiver and propagation parameters. These include parameters based on the hardware choices for the reception system and the losses due to propagation. Often times, the propagation loss is the greatest loss in the system and that is true of the TOMCAT system. Effectively designing to overcome the propagation loss is mandatory for all communications systems.

The receiver parameters include the antenna gain and cable loss. The antenna gain is given in units of dBi. This is reference to an isotropic radiator where all power is equally transmitted in all directions. In

reality, such a radiator does not exist, nor is it necessarily beneficial. A real antenna has a beam width pattern which describes the direction in which the antenna radiates. Since the power is now confined to this area, there is a gain associated with this directionality. TOMCAT is using a monopole antenna on board. This antenna radiates in a hemisphere and has a modest gain of 2.2 dBi. The ground station uses a Yagi antenna which is more directional. Due to the higher directionality, the gain is an increased value of 8.5 dBi. It should be noted that TOMCAT is a static pointing system, otherwise pointing considerations would have to be taken into account here.

The propagation parameters are given in Tbl. 31 as well. The propagation loss is how much power is lost in the signal over the intended transmission distance of 104 km. Equation 1 shows how this value is calculated based on wave length (λ) and distance (R). Additionally, atmospheric attenuation contributes to loss in the signal of 0.45 dB. This value can be found by referencing Fig. 15 taken from SMAD. Finally, polarization loss is estimated to be 0.2 dB (again from SMAD).

$$L_s = 10 \log\left(\left(\frac{\lambda}{4\pi R}\right)^2\right) \quad (1)$$

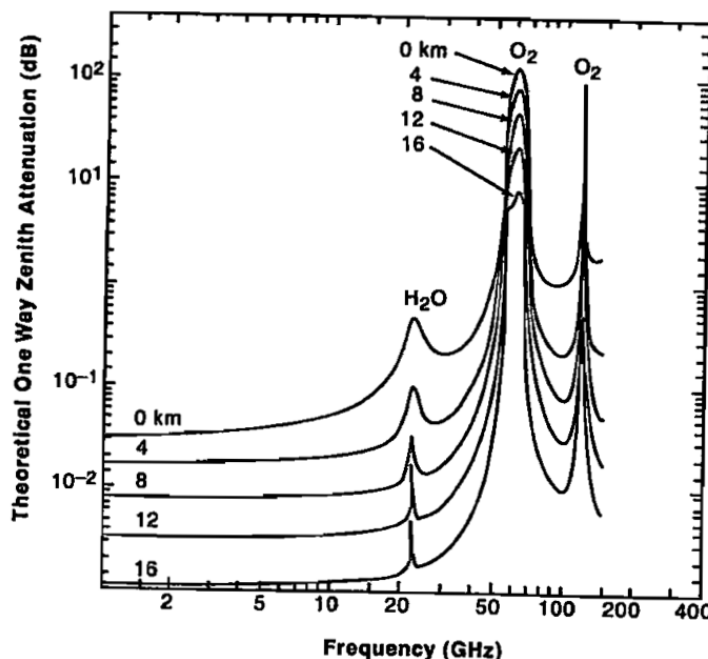


Figure 15. Expected Atmospheric Attenuation (SMAD)

Table 32. TOMCAT Link Budget: Transmitter Parameters

Parameter	Uplink (Command)	Downlink (Telemetry)	Units
Transmit Antenna Gain	8.5	2.2	dBi
Transmission Cable Loss	0.5	0.5	dB
Transmission Power	-2.2	4.1	dB
Power Transmitted	0.6	2.6	W
Effective Isotropic Radiated Power	5.78	5.85	dB

Table 32 gives the parameters associated with the transmission characteristics of the system. These include the gain of the transmission antenna, line loss, transmission power, and effective isotropic radiated

power. In general, it is best to transmit with as much power as possible to reduce to necessity for amplifiers along the RF chain. Choosing a high gain antenna also helps remove the need for more amplifiers in the chain.

The first parameter is the transmit antenna gain. It can be noticed that essentially, the antenna gains have swapped places from the receiver antenna gain line. This is because on the uplink, the yagi antenna is transmitting while the monopole antenna is transmitting on the downlink. The exact same yagi and monopole antennas are being used so the gain for each are still 8.5 dBi and 2.2 dBi respectively. The line loss is also 0.5 dB as the same cables are going to be used.

The next parameter is power transmitted, which is subject to multiple constraints. The driving factor in this case is the legality of broadcasting power in this band. The effective isotropic radiated power cannot be above 6 dB in order to remain legal. Due to this constraint, the power can either be made up in the power transmitted or the antenna gain. On the ground side, the antenna gain is high enough that the power transmitted can be lower at 0.6 W. This has the benefit of being less power intensive to operate. On board, the antenna gain is small, which means the power transmitted must make up more of the effective isotropic radiated power.

Finally, the effective isotropic radiated power is the sum of the transmitted power, antenna gain, and line loss. As mentioned earlier, this value must be lower than 6 dB in order to remain legal. In order to meet this value, easily implementable amplifiers were used to boost the signal. In order to fine tune the power transmitted, the LimeSDR Mini has an onboard amplifier which can be programmed for up to 72 dB of gain. The team will fine tune this value in order to get as close to 6 dB of effective isotropic radiated power as possible. Right now estimates show that 5.78 dB on the uplink and 5.85 dB on the downlink are reasonable. This is due to the step change of 1 dB on the transmitter gain. There isn't a step there makes the effective isotropic radiated power exactly equal 6 dB.

Table 33. TOMCAT Link Budget: Summary

Parameter	Uplink (Command)	Downlink (Telemetry)	Units
Effective Isotropic Radiated Power	5.78	5.85	dB
Propagation Loss	132.69	132.69	dB
Receiver Gain	1.7	8	dBi
System Noise Power	198.19	203.5	dB-Hz
Pr/No	72.98	84.66	dB-Hz
Minimum Pr/No	35.87	62.17	dB-Hz
Link Margin	34.10	19.47	dB-Hz

In summary, Tbl. 33 shows key values that summarize the TOMCAT link budget. The new item that is introduced here is the Pr/No of the designed system. This value can be computed with Eq. 2 where EIRP is effective isotropic radiated power, L_s is space loss, L_a is atmospheric loss, G_r is receiver antenna gain, and T_s is system noise temperature. Note that this equation is in terms of decibals and that 228.6 comes from the logarithm of the boltzmann constant.

$$\frac{P_r}{N_o} = \text{EIRP} - L_s - L_a + G_r - 10 \log(T_s) + 228.6 \quad (2)$$

This Pr/No represents what the designed system is capable of producing. In order to get the final link margin, the minimum Pr/No is subtracted from the Pr/No of the system. This gives a 34.10 dB-Hz and 19.47 dB-Hz margin for the uplink and downlink respectively. The margin is large in order to account for any modeling issues with the link budget such as an underestimation of noise. This also allows more margin in case the balloon drifts further than expected.

5.1.2. RF Chain

Trevor Weschler

The link margin design ensures that the signal can reach the destination and be received. However, a signal must be conditioned before it is processed by the software defined radio. Essentially, filtering must be employed to avoid aliasing in the signal. Despite the capabilities of the software defined radio, it cannot remove signal aliasing through its software. To avoid signal aliasing, a low pass filter is implemented in the reception chain for both the ground and on-board systems. In order to simplify the design, the same chain is used for both the on-board and ground reception systems. Figure 16 shows the on-board receiver chain.

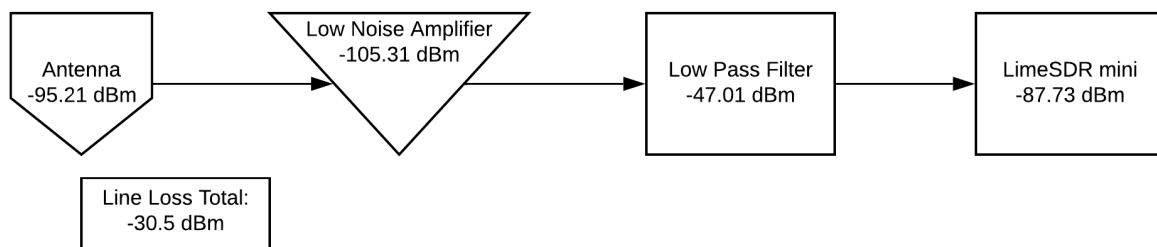


Figure 16. On-board Receiver RF Chain

It is important to note that the introduction of a low pass filter reduces the signal strength. In order to account for this, a low noise amplifier (PMA4-33GLN+) is introduced to the chain. This amplifier boosts the strength of the signal without adding too much noise. At each stage of Fig. 16, the signal strength is printed on the component. In order to be decoded by the radio, the signal must arrive with at least -100 dBm of strength. As indicated, the signal arrives with -87.73 dBm meaning that the signal should be able to be decoded on-board. A similar analysis is done in Fig. 17

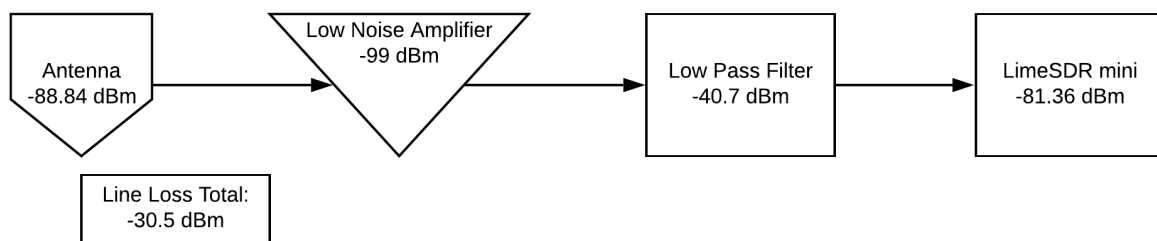


Figure 17. Ground Receiver RF Chain

Figure 17 shows the ground side receiver chain. The same hardware components are used. Note the low pass filter meant to prevent signal aliasing. In addition the signal power at the radio is above -100 dBm ensuring that the signal can be decoded. The difference between the two is that the ground receiving antenna has a larger gain meaning the signal arrives in the chain with more power than the on board version.

The transmission chain is designed to transmit as much power as possible. The layout includes the radio, a power amplifier, and then the antenna. In the transmission case, there is a difference between the on board and ground chains. On the ground, the power amplifier is weaker than on board. This is due to the antenna

gain on the ground being much higher gain than the antenna on-board. The ground side power amplifier gives 20 dB of gain and the on-board power amplifier gives 22 dB of gain. The associated effective isotropic radiated powers are listed in the link budget section.

5.1.3. Ground Station Placement

As mentioned in the conceptual design section, the team chose a four element Yagi antenna for the ground station in order to maximize the gain without sacrificing too much beamwidth. However, the beamwidth is still limited, 100 degrees in the horizontal direction and 70 degrees in the vertical direction. The team was initially concerned that if the ground station was simply placed near the launch site, the balloon would rise outside of the vertical beamwidth. The team would have to respond to this issue by implementing a system to change the pointing of the antenna as the balloon ascended. In order to avoid the complexity involved with automated or manual pointing, the team chose to instead displace the ground station such that the balloon would still be within the vertical beamwidth of the antenna at its maximum altitude of 30 km. A visual representation of this concept is shown in Figure 18.

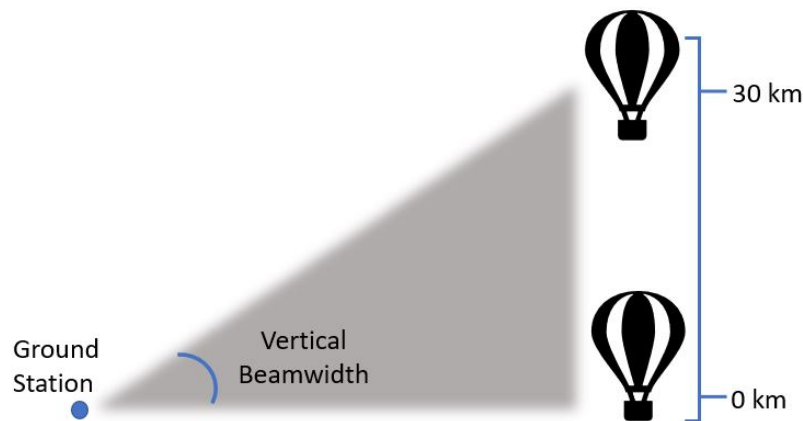


Figure 18. Ground Station Offset Concept Visualization

Figure 19 conveys the result of the geometric analysis implemented to determine the required offset. As can be noted from this image and recalling the 70 degree vertical beamwidth of the ground antenna, the minimum required offset for the ThinSat/Balloon configuration to always remain within view is 10 km.

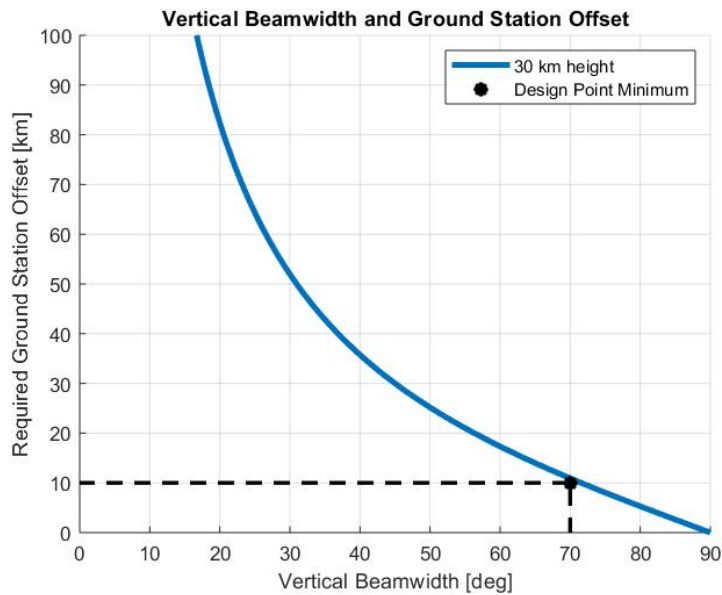


Figure 19. Required Ground Station Offset

Another concern arose after deciding to offset the ground station. This concern was that the geography of the area surrounding the launch site could make it difficult to communicate with the ThinSat until it ascended a sufficient distance above the ground. Thus, the team decided to place a second, identical ground station at the launch site. This ground station will allow the team to communicate with the ThinSat until it passes outside of the vertical beamwidth. The ground station at the launch site will be used to command the ThinSat until telemetry can be heard and understood by the offset ground station. At this point, commanding will be done from the offset ground station while the ground station at the launch site can still provide verification of the communications. The TOMCAT team will remain in constant contact with each other to ensure a smooth transition. With a ground station at the launch site, the team would also be able to verify ThinSat communications prior to the launch. This could also serve as an important trouble shooting tool if something is not working as planned on the day of the launch.

After the discussion of how to adjust the design to account for the constrained vertical beamwidth, the question of the constrained horizontal beamwidth arises. The team consulted balloon flight data from the Gateway to Space balloon flights and it was determined that the balloon flight typically travels in one direction with lateral movement off that direction not approaching distances of any concern with an antenna horizontal beamwidth of 100 degrees. The only concern would be if the balloon drift (maximum of 100 km) had a large drift component across the view of the antenna rather than down range. To account for this, the team will scope out four potential locations near the launch site for the offset ground station and consult the wind predictions on the day of the launch to help choose the best of the four locations for the day of the launch. The launch site has not been determined for the Gateway to Space spring launch yet. As soon as the location is determined, the team will develop a detailed plan of the options for the offset ground station location. It is important to note here that the ground station is very mobile and will not require extensive setup efforts. Thus, finding out where to setup so close to the day of the launch will not be an issue for the team.

5.1.4. Communications Printed Circuit Board

Trevor Weschler

The communications system on board will have its own printed circuit board to hold the low noise amplifier and the power amplifier. The goal of the PCB is to reduced the mass and space required by the

communication system. Without a custom board, each component would need to have its own evaluation board. TOMCAT does not have the physical space available to give each component its own evaluation board. Instead a single PCB will be used and stacked on top of the SDR. The board itself is simple and the schematic is shown in Fig. 20. It should be noted that the ground has no such size requirements and an evaluation board will be used for the low noise amplifier on the ground.

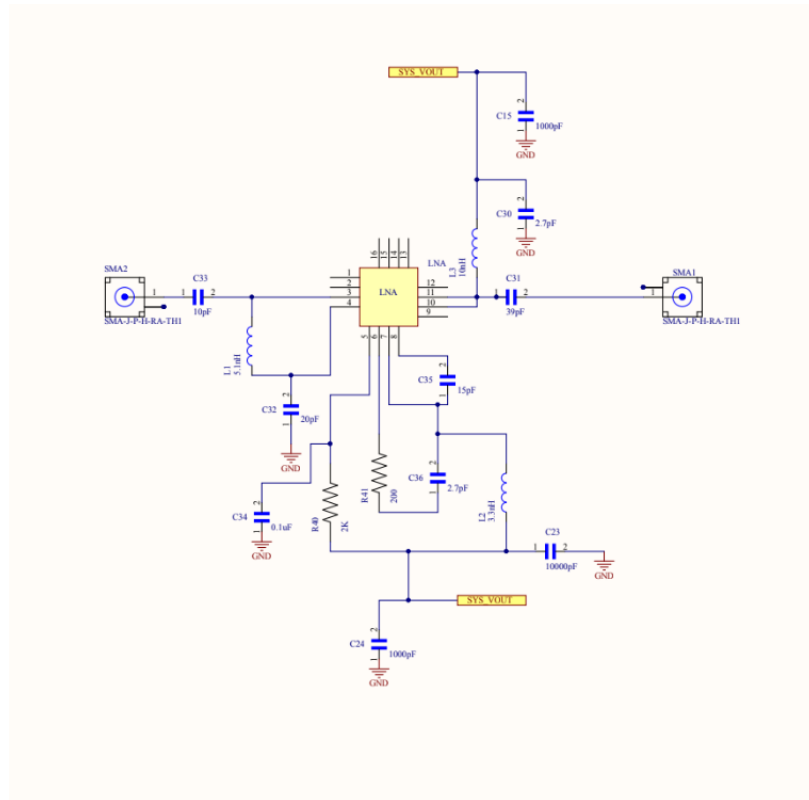


Figure 20. Custom Communications PCB Schematic

5.2. Communications Software

The communication software will provide an interface between ground/flight software and the SDRs. The communication software will run on both the ground station laptop and the OBC, and will be handling transmission and reception on both. A software development kit called GNU Radio was used to develop and test the communication software. The design of the transmission and reception signal processing chains will be described below. To verify the signal processing model would work as intended to modulate, process, and demodulate a signal without having to hardware test, a GNU Radio simulation was constructed. The full block diagram is pictured below.

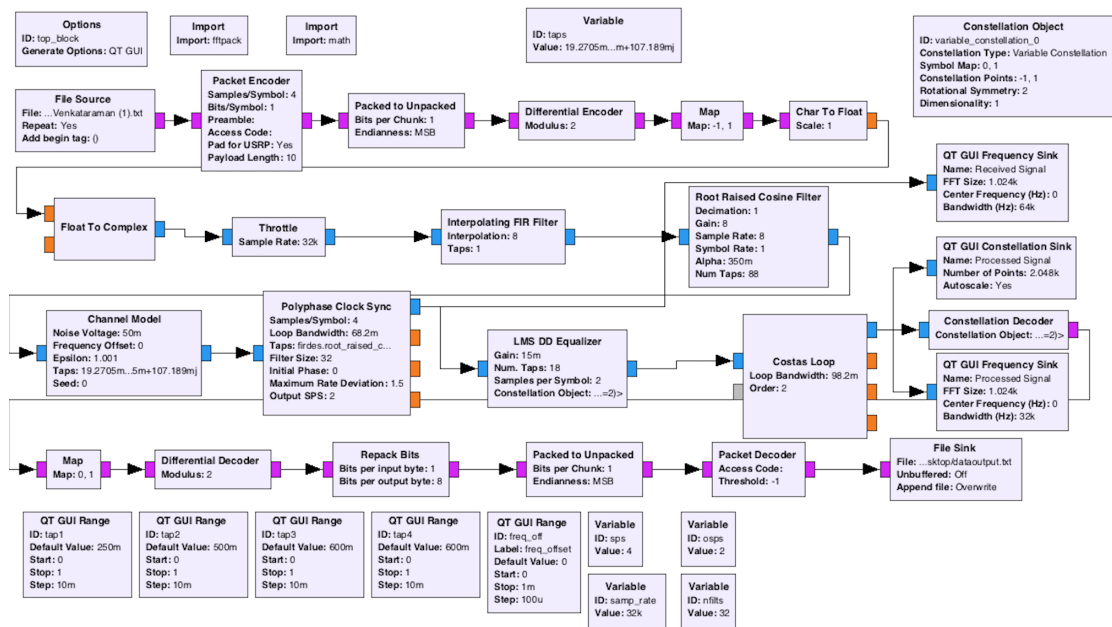


Figure 21. Full GNU Radio simulation block diagram

5.2.1. Transmission

The purpose of the transmission side of the communication software is to take binary data from from ground and flight software over a UDP connection and turn it into a digital signal that can be transmitted by the SDR. The modulation scheme used is called differential binary phase shift keying (DBPSK). It differentially encodes data in phase shifts in a carrier wave. The carrier wave will shift between two phases offset by 180 degrees. A phase shift in the carrier wave represents a one, while a lack of a phase shift represents a 0. Thus, a series of ones and zeros can be encoded in the wave.

Since the square wave generated by modulation has infinite bandwidth, the signal needs to go through a process called pulse shaping to constrain the frequency spectrum to a particular bandwidth. A root raised cosine filter was used to pulse shape the signal.

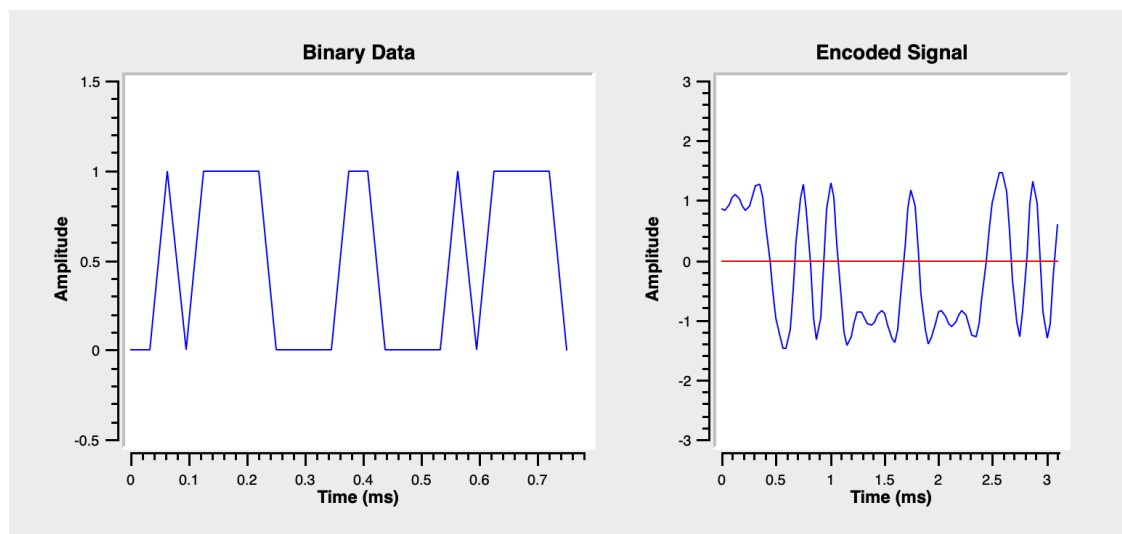


Figure 22. Binary data before and after modulation process

5.2.2. Reception

On the reception end, a digital signal will be taken from the SDR, processed to account for interference introduced during transmission, and decoded back into binary data to be sent to flight or ground software. Since the two SDR's oscillators are not perfectly in sync, the receiving SDR will not sample the incoming signal at the optimal points, which results in a completely unintelligible signal in most practical situations. A polyphase clock synchronization algorithm was used to account for this. Another source of interference is multipath interference, where the transmitted signal reflects off of objects in the environment and arrives at the receiver multiple times. The results is that some frequencies along the bandwidth show increased gain, while others show reduced gain. A least means squared equalizer was implemented to undo the effects of this interference. This equalizer uses knowledge of what the incoming signal should look like to provide an error value for the LMS algorithm.

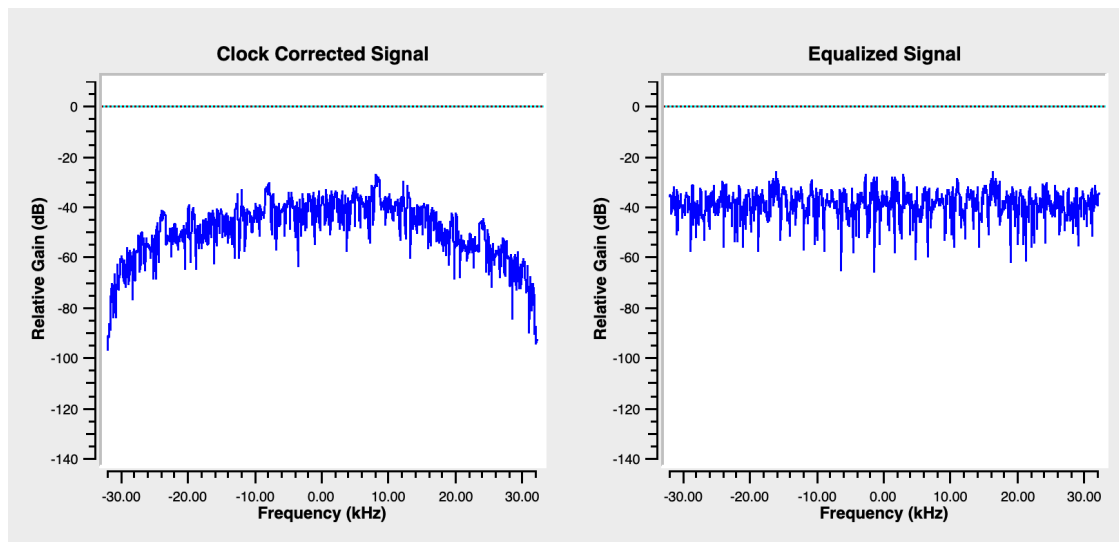


Figure 23. FFT Spectrum before and after equalizer

From this FFT plot, it can be seen that the equalizer is working as intended and boosting the power on lower power frequencies so all the frequencies have roughly the same gain. Since the oscillators of the two SDRs drift relative to each other, there will also be frequency drift in the carrier wave. A Costas loop algorithm was used to account for this drift. Finally, the signal can be demodulated by reversing the modulation process described above.

5.3. Printed Circuit Board

Grant Novota

The custom ThinSat payload PCB is a 4-layer board that integrates many electrical subsystems and was designed through the use of open source reference schematics and component datasheets. The main objectives in the design of the PCB were to design for connectivity, performance, manufacturing, assembly, bring-up, troubleshoot, and debugging. All of these subsystems work together to satisfy the following functional requirement:

FR4: The payload board shall perform all necessary hardware tasks for system operation.

5.3.1. The Microcontroller Unit

The chosen MCU is an AM335x model. The payload PCB utilizes the Octavo Systems OSD335x C-Sip (Complete System-in-Package) surface mount package, which integrates the AM335x MCU with all of the necessary components that are required for a typical embedded system. These components include DDR3 memory, power management, EEPROM, up to 16GB embedded multi-media card (eMMC) Non-Volatile storage, MEMS oscillator, and other passive components. Octavo Systems provides a PCB schematic and footprint library for this part, where the part schematic is split up into 6 separate blocks (A-E) allowing the design to be easily separated across multiple pages of the PCB schematic.

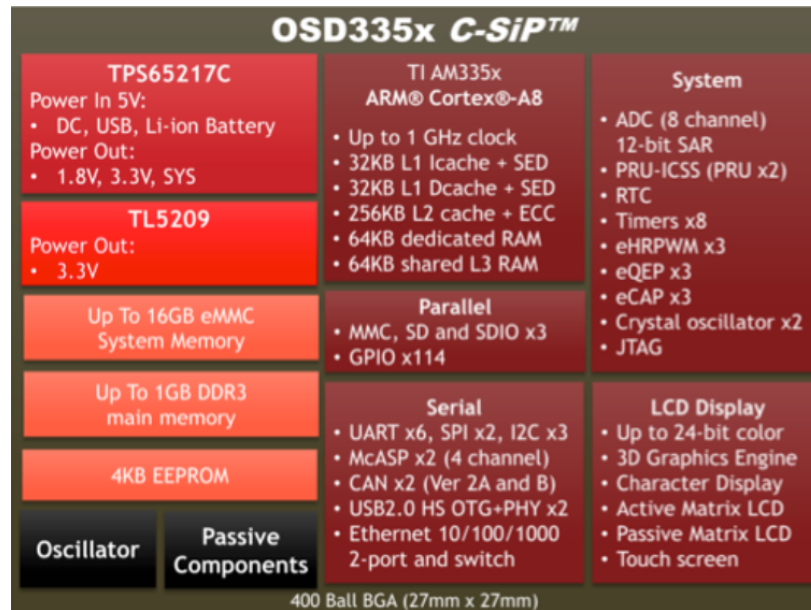


Figure 24. OSD335x C-Sip Functional Diagram

[26]

5.3.2. Power and Ground

To ensure stable operation of the PCB electronics, the power circuitry must be able to meet the power requirements of the device under both normal and extreme conditions. The OSD335x contains internal circuitry to regulate the power on the PCB. The design of this custom PCB also incorporates extra features for power stability such as a clamping circuit and electro-static discharge protection.

Power Input: The OSD335x package drives power to the rest of the electrical components on the PCB and can be powered in a variety of ways. For this design, we have chosen to allow power from either a battery or from a standard USB port. The power inputs of the OSD335x schematic were attached to the appropriate nets to connect the package to 5V power from both the battery and USB circuitry.

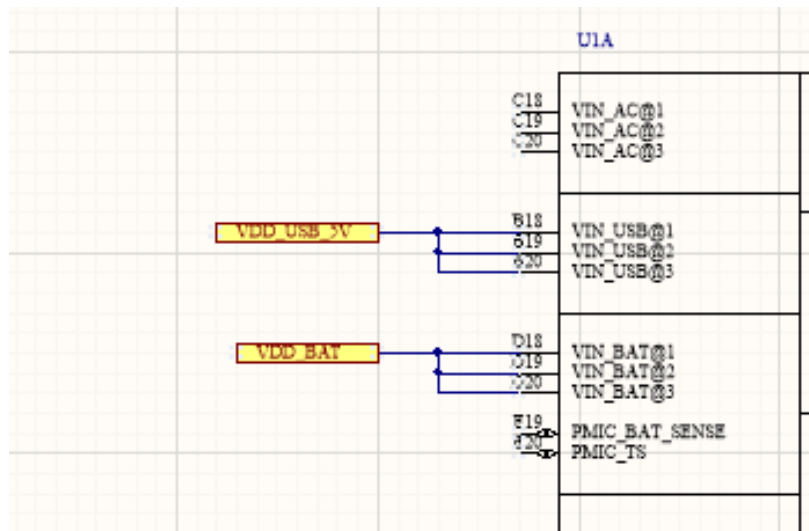


Figure 25. OSD335x Power Input PCB Schematic

Power Output: The power management integrated circuit and low-dropout linear voltage regulator inside the OSD355x provide multiple different power outputs. Some of these outputs are used internally by the OSD355x package and others are available for external use and are described in the datasheet. This PCB design utilizes some of the 5V, 3.3V, and 1.8V outputs to power the rest of the circuitry on the board.

Ground Connections: The OSD335x has several ground connections that are connected together within the package, but to design for stability we have each of these pins connected externally to the ground plane of the PCB to ensure the shortest return current paths. The figure below shows what this looks like in our PCB schematic.

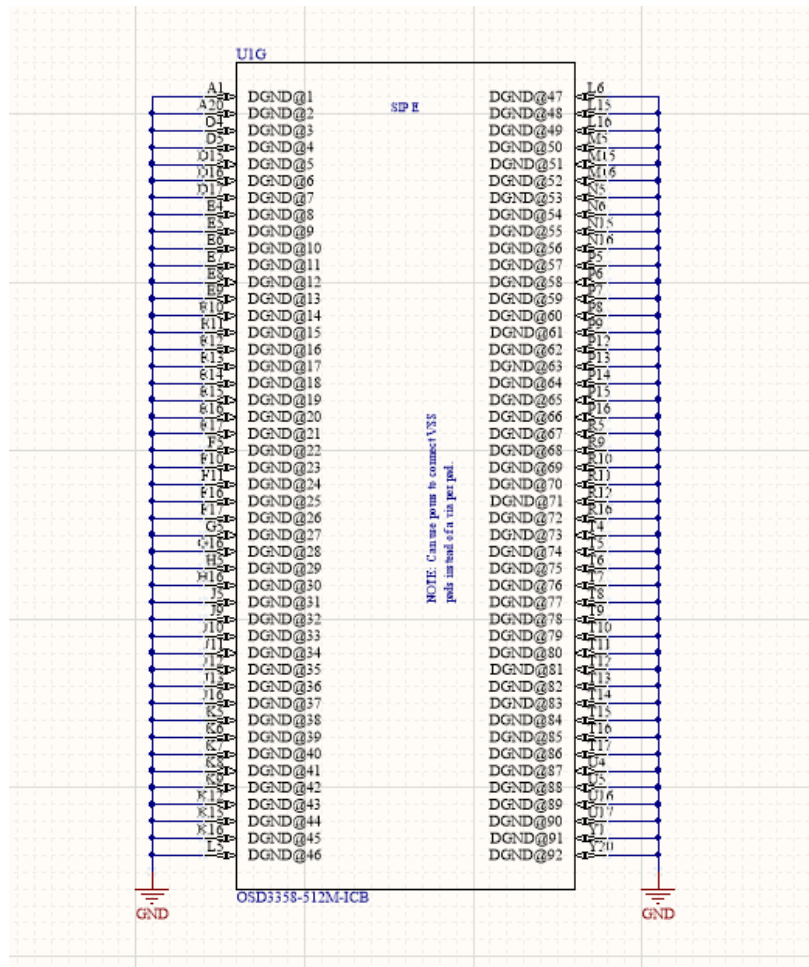


Figure 26. OSD335x Ground Connections PCB Schematic

Power Management: The AM335x MCU communicates with a power management integrated circuit (TPS65217C) inside of the OSD335x package via an I2C interface. Most of the functionality of this integrated circuit can be controlled programmatically through the I2C connection such as the internal buck and boost converter output voltage, linear voltage regulator output voltage, power up/down sequences, and over current thresholds. This I2C connection is also shared with the IMU, so we needed to add extra pull-up resistors to the I2C line to ensure that edges of our signals remain sharp.

In addition to the I2C connections, some more signals needed to be connected between the MCU and power management so that the MCU can be used to regulate power. We also added a mechanical power button to power up the board through the power management integrated circuit.

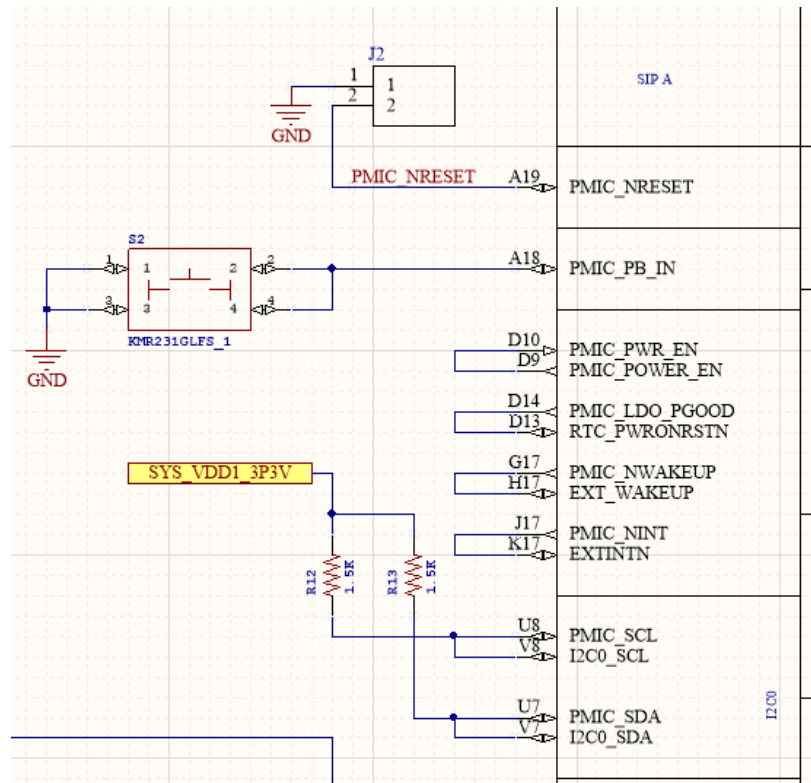


Figure 27. Power Management Connections

5.3.3. Clamping Circuit

AM335x reference designs include a clamping circuit for the MCU package. This clamping circuit maintains the voltage level of an input with respect to another input. The reason for the inclusion of this circuit is to help the system maintain a state that meets the power down requirements of the MCU.

The AM335x datasheet requires that the voltage difference between the power rails (1.8V and 3.3V) be less than 2V during the entire power-down sequence. If we can't guarantee that this voltage difference will be 2V during the whole power-down sequence, then we need a clamping circuit. We have chosen to include the clamping circuit to avoid the possibility of complications when turning off the MCU.

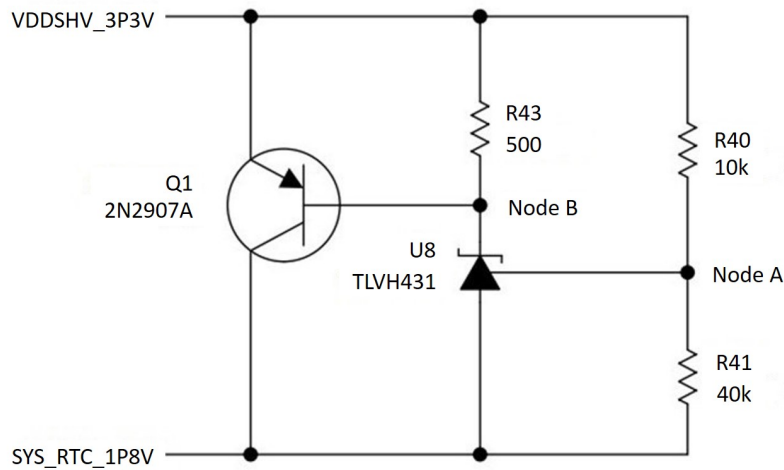


Figure 28. Clamping Circuit

[26]

The clamping circuit maintains a 2V voltage across the PNP transistor (Q1). It does this in the following way:

The shunt regulator (D1) will be off whenever the voltage at its reference input connected to node A is less than 1.24V. It will turn on whenever the voltage at its reference (node A) is greater than 1.24V. the Q1 transistor will allow current to flow from its emitter terminal to the collector terminal when the voltage difference between its base terminal and emitter terminal is greater than or equal to 0.4V. This means that the Q1 transistor acts like a closed switch when the voltage difference between node B and the 3.3V power rail is greater than 0.4V and it acts like an open switch when the voltage difference between node B and the 3.3V power rail is less than 0.4V.

This clamping circuit design was added to the PCB schematic, which can be seen in the figure below.

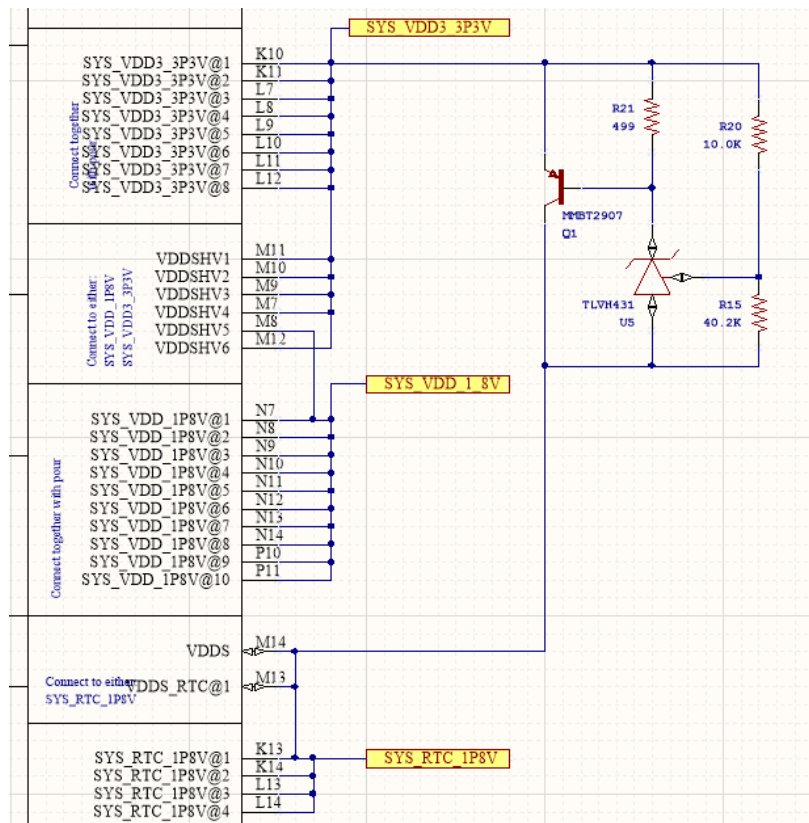


Figure 29. Clamping Circuit PCB Schematic

5.3.4. Electrostatic Discharge Protection

To protect the PCB from ESD damage, which can occur when the board is touched, a ground plane border was included into the design. This border is given the common ground net name (CGND) and is created by using a copper fill along the edge of the PCB, where it will be in contact with mounting screws around the board. This common ground net is connected to the power I/O so that an ESD strike from the source powering the board can be dissipated to the common ground plane and kept from damaging the components on the PCB.

5.3.5. Reset Circuitry

Reset circuitry was included into the design of the PCB to help the MCU recover and resume normal operation whenever it gets stuck in an erroneous state. Additionally, it is used during power-up to make sure the MCU and all of its subsystems start their operation from a known state. A reset button, as well as a supervisor circuit were added to the PCB to allow the board to be reset. The purpose of the supervisor circuit is to perform two functions. It checks if the supply voltage to the board falls below a preset voltage or if the manual reset input is asserted from the mechanical reset button.

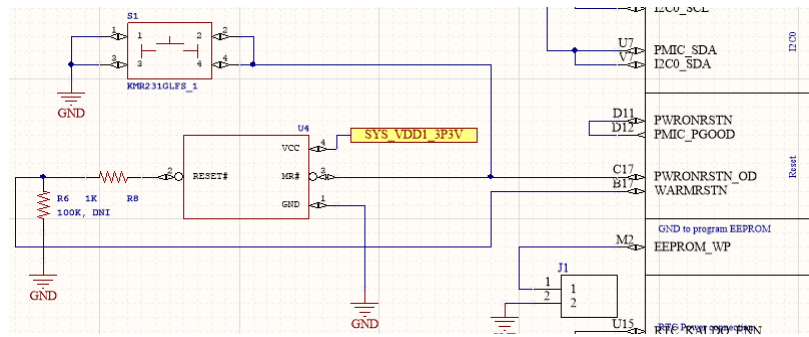


Figure 30. Reset Circuitry PCB Schematic

5.3.6. Clock Circuitry

A system clock is an essential component in any MCU board design. The OSD335x package has two clock inputs for high and low frequency oscillators. For our application, we only need the high frequency input, which will provide a reference for all non-real-time clock functions of the PCB. The crystal clock circuitry and it's capacitor values are specified in the AM335x datasheet.

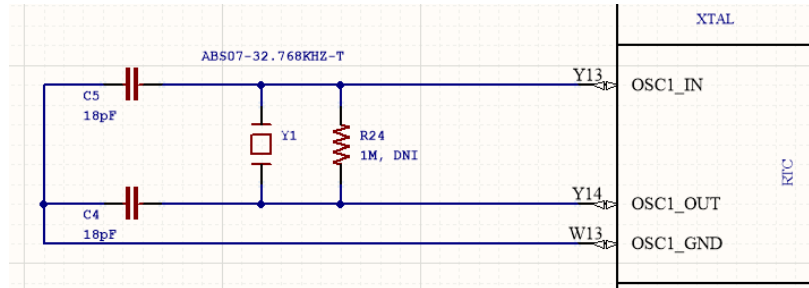


Figure 31. Clock Circuitry PCB Schematic

5.3.7. Peripherals

IMU: The MCU must interface with the IMU via an I2C connection. The I2C pins of the IMU are specified in its datasheet and are connected to the I2C line of the MCU, which is shared with the power management integrated circuit. Since we included pull-up resistors in our design, the signal integrity on the I2C lines will be preserved with this additional load. The rest of the IMU connections for power and ground and are also specified in the datasheet.

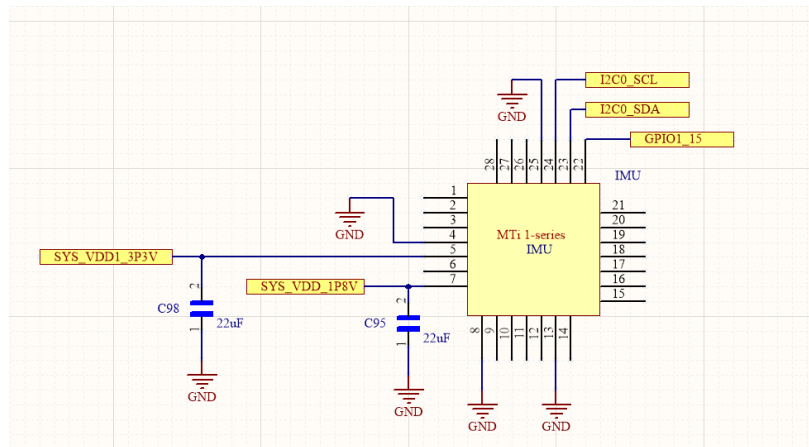


Figure 32. IMU PCB Schematic

JTAG: Our PCB needs a way to load programs onto the MCU. This is done via a JTAG interface using a standard 20-pin connector that supports most JTAG debuggers.

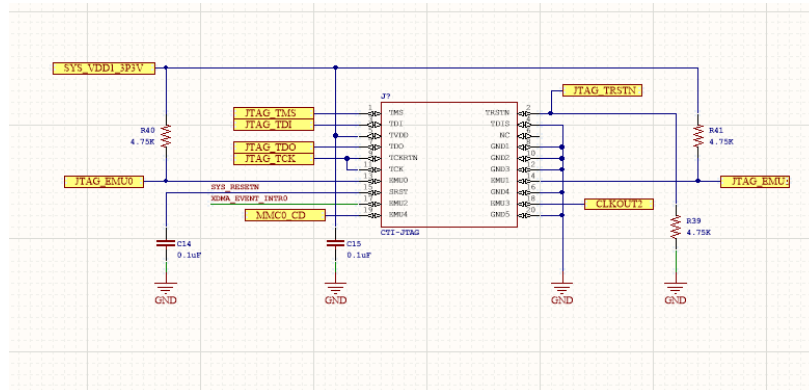


Figure 33. JTAG PCB Schematic

5.3.8. USB

This design includes two USB connectors; one for interfacing with a computer device and another for interfacing with the SDR. These USB connections follow the same standard for all USB A connectors and fuses were added on the power line to protect the board from any power spike.

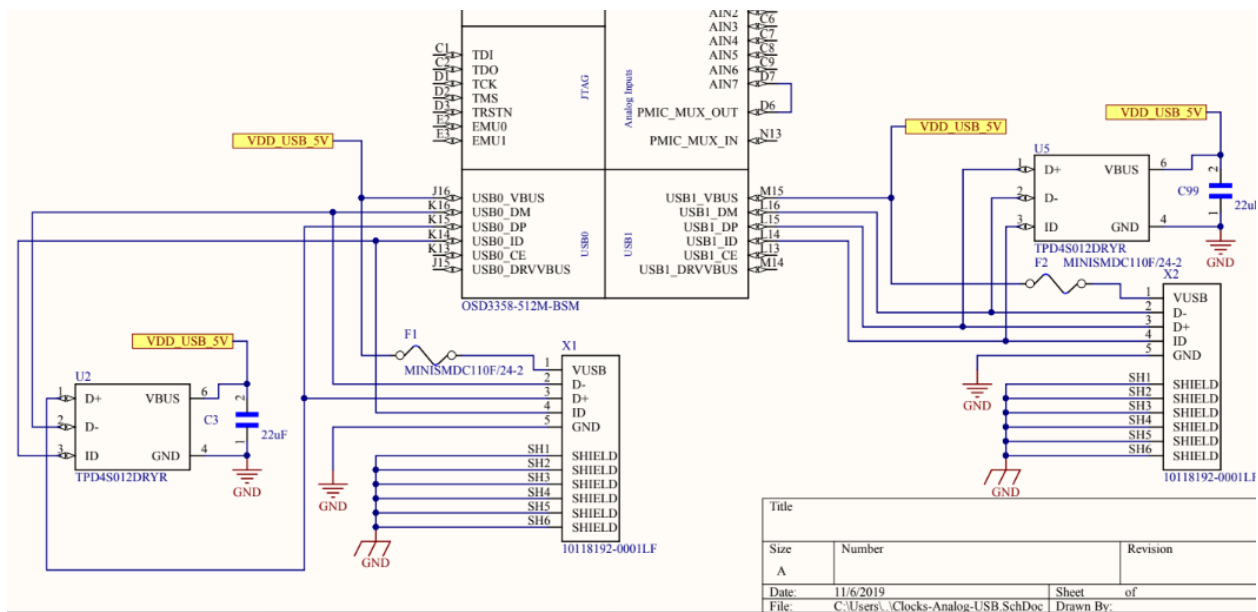


Figure 34. USB Ports PCB Schematic

5.3.9. Non-Volatile Storage

The PCB must be capable of running KubOS, which is a Linux kernel. This requires non-volatile storage such as a micro-SD card slot and embedded multimedia card (eMMC) to be included in the board design. Luckily, the OSD335x surface mount package has an eMMC internally integrated, so an external eMMC does not need to be mounted to the PCB. This leaves us with just the need to design an external micro-SD card connection. All of the pin connections are specified in the datasheets of the OSD335x and micro-SD card slot and decoupling capacitors were added to reduce noise.

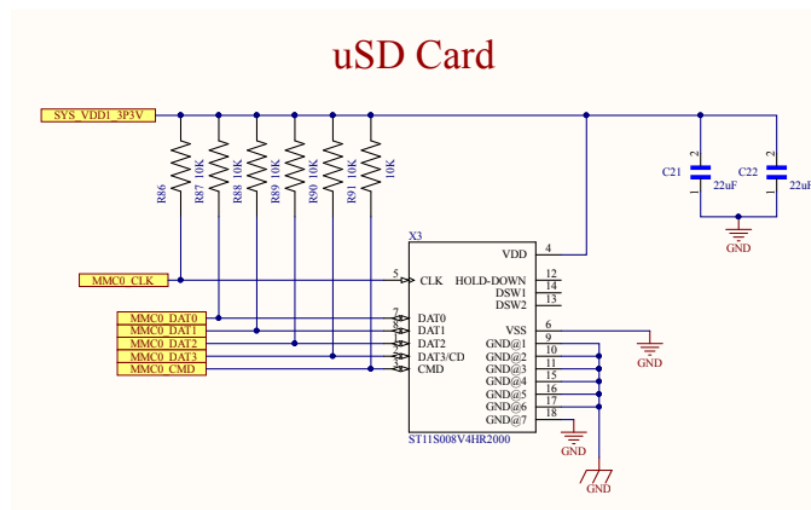


Figure 35. Micro-SD PCB Schematic

5.3.10. Board Layout

The PCB board layout design fulfills the connectivity requirement, where all of the connections between components specified in the schematics have copper traces making these connections. Furthermore, the copper trace widths were varied so that 6 mil traces were used for signals and 10 mil traces were used for power and ground connections. A lab experiment was done to test how much current a copper trace could handle before it began smoking and the results were that a 6 mil trace could handle 1.2 A of current and a 10 mil trace could handle over 5 A. The signal currents will remain well below half of this 1.2 A limit for a 6 mil trace and the power currents will also remain below half of the 5 A limit for a 10 mil trace.

The layout design was also created with the goal of maintaining signal integrity. This was done by following good electrical engineering design practices such as keeping traces as short as possible, using large power and ground planes, and routing the traces to keep mutual loop inductance at a minimum. Mutual loop inductance is where the changing current in one copper trace induces a magnetic field, which then induces a voltage in adjacent traces and therefore, creates unwanted current in the adjacent traces. Avoiding large current loops in the layout, as well as routing traces on the top layer of the board in opposite directions from the traces in the bottom of the board are some of the strategies used to mitigate mutual loop induction and keep the PCB signal noise at a minimum.

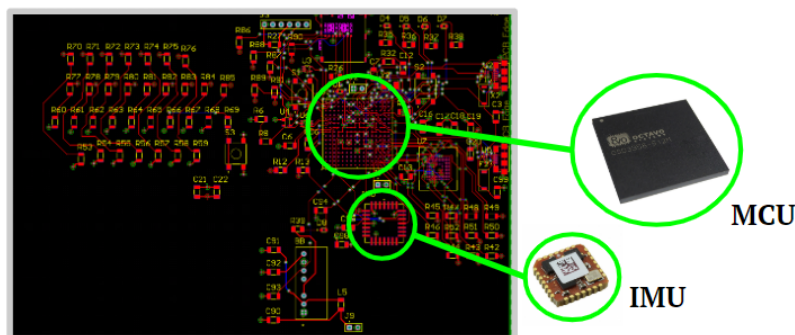


Figure 36. Custom PCB Board Layout Design

5.4. Flight Software

Katie Steward and Cole Kenny

The chosen flight software framework was KubOS. The mission applications running on our micro-processor will be built upon KubOS's software structure. KubOS provides a number of "microservices" while fulfilling basic low-level flight software tasks. These include file transferring capability, monitor services for hardware and telemetry, a telemetry database built on SQLite, shell services for uploading new code, and hardware and communications services which can be configured for a number of different kinds of radios.

The primary advantage to using the KubOS software is that the team can focus on coding the mission software that specifically accomplishes the tasks that the team needs in order to communicate with the IMU and fulfill the project objectives. The built-in KubOS software provides a framework for the low-level services that the TOMCAT mission needs that can be fine-tuned to fully support the mission. KubOS is built off of a Linux system, so the low-level software focuses primarily on the required Linux services. These built-in services are summarized in the table below:

Table 34. KubOS Basic Services

Services	Basic Functionality
File Transfer	Focuses on moving command and telemetry files between different parts of the system. For the TOMCAT software, this will include between the IMU, the signal processing software, and to the telemetry storage on the microcontroller.
Monitor	Primarily works to ensure that all software processes are running as intended. Keeps track of software issues for easier debugging.
Telemetry Database	The telemetry database is stored in the microcontroller memory using a standard Linux SQLite database. This will allow telemetry to be stored for the duration of the balloon flight so the team can compare the downlinked data to the onboard stored data.
Shell	This service is primarily for externally communicating with KubOS software. A Linux shell gives a command line so that Linux operations can be performed on the software while the software is in the development and testing.
Hardware	The hardware tools are low-level functions to perform low-level Linux tasks. This is primarily used to communicate with the microcontroller to execute hardware changes. An example of when this is necessary is when the software needs to query the database or use mutations.
Communications	The communications service will primarily communicate with the signal processing software. It will receive the command packets from the signal processing software and send telemetry packets back to it.

These services provide the low-level foundation for the TOMCAT flight software to be built upon. The following table shows the mission applications that need to be built/modified to support the TOMCAT mission:

Table 35. TOMCAT's Mission Applications

Mission Applications	Basic Functionality
Housekeeping	Reduces mission application complexity and increases fault tolerance by separating critical hardware components in software. This ensures that if some hardware system fails, they don't all fail with it.
Passive Telemetry Collection	Stores collected telemetry to the telemetry database. Polls hardware services periodically to collect data to store.
Deployment	Manages any tasks that need to occur after deployment from a launch vehicle but prior to normal system operation. TOMCAT's focus in this area will be limited due to project scope, but research will be provided to Raytheon.
Beacon	Downlinks critical telemetry (such as those related to system health and diagnostics) at a regular interval separate from the usual telemetry packets. While separate from regular data telemetry, the Beacon application still uses KubOS's communications service.
Operations	Defines the actual mission activities of the satellite.

Figure (37) shows the interactions between the custom designed mission applications and the core services KubOS provides.

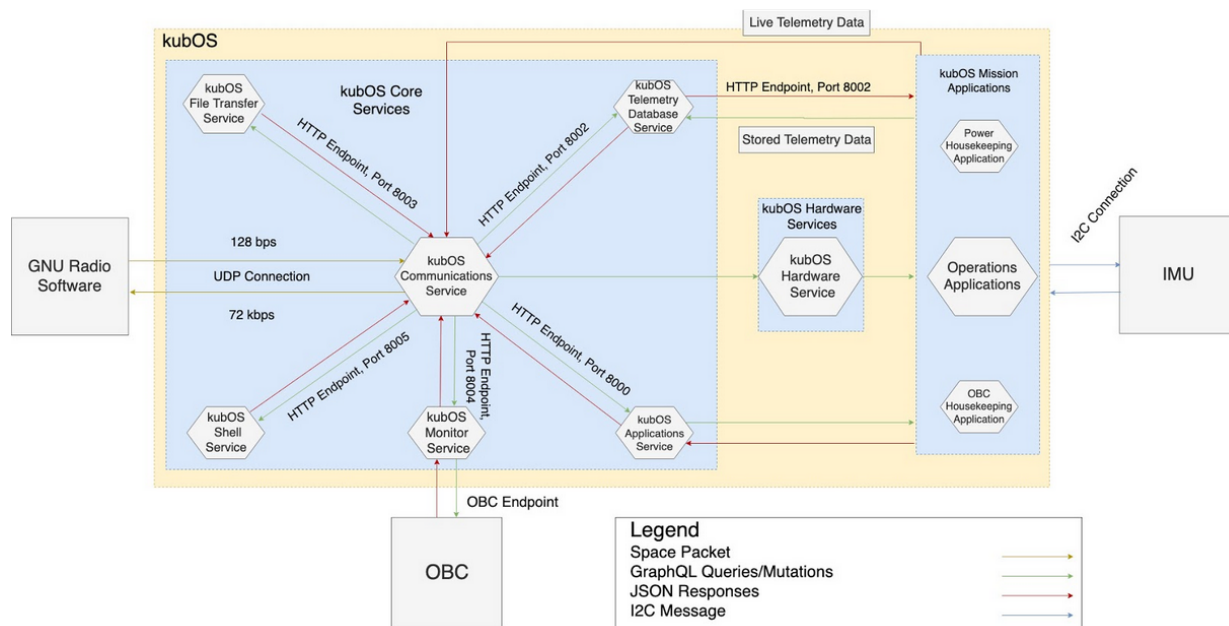


Figure 37. Flight Software Framework's Freebody Diagram

5.5. Ground Software

After deciding that COSMOS ground software and interface satisfied all the requirements, it was necessary to create a plan to assimilate COSMOS's features into the current mission. As stated previously, the main purpose of a ground software is to send commands, and receive and store telemetry. More specifically, it must send commands packetized in CCSDS format over a UDP connection to the communications software, also running on the ground computer. Then it must be able to receive telemetry data at a certain data rate in packetized CCSDS format, have the ability to plot the raw data, and then depacketize and store in text file format. The major functional requirement that the ground software satisfies is:

FR2: The payload shall communicate with the ground system during the balloon flight.

Although COSMOS contains the foundation for everything that is needed for this mission, the design portion comes from writing configuration files. Configuration files define how a tool in the COSMOS suite acts, and what it should be doing. It is crucial to the success of the project that the commands/telemetry packets are in a specific format of CCSDS. Because CCSDS format is a general term, the exact format of the header and the data included in there must be chosen. Once the information that will be contained in the packet is decided upon, the configuration files for the applications being used (namely Command Sender and Telemetry Extractor) must be written to specifically handle that packet type. These edits to the configuration files are written in Ruby, and direct the application to know what format of file to expect for sending and receiving packets.

With these configuration files taken care of, the data can be transferred in the standard manner COSMOS uses. The commands are wrapped in a space packet, and sent using the Command Sender tool up to the ThinSat and on-board computer, then it receives IMU and health status data in a space packet from the ThinSat. This telemetry data is plotted using the Telemetry Grapher, logged from the Command & Telemetry Server, and depacketized and transformed using the Telemetry Extractor.

The general functional block diagram for the workings of the ground software is shown below in Figure 38.

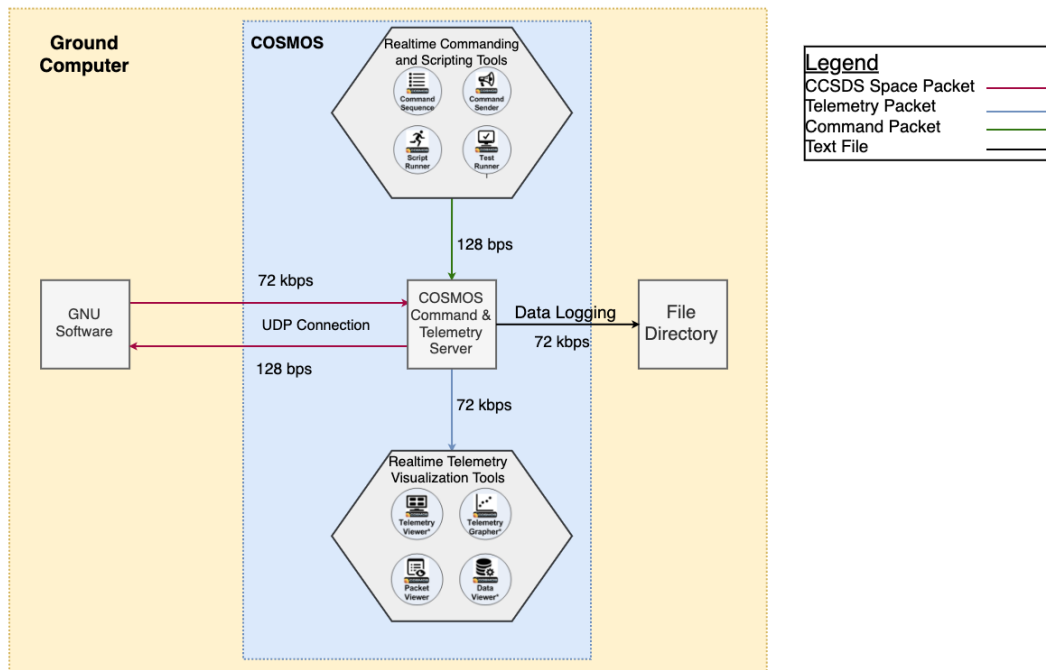


Figure 38. Functional Block Diagram for Ground Software

5.6. Thermal

Lara Buri

In order to understand the effects of the low temperatures experienced during a high-altitude balloon flight, a thermal model is needed to determine the effects of insulation on the TOMCAT system. Two models were developed in order to study the effects of low temperatures and how well insulation maintains acceptable internal temperatures. The motivation of these models is to satisfy design requirement 3.1:

DR3.1 The payload shall be able to withstand the temperature range of flight, which includes the coldest point in the flight (-56°C), and a hot day on the ground (38°C).

To meet this design requirement, the electronic components must not fall below their specified temperature rating as given by their datasheets. These temperature requirements are shown in Table 36.

Table 36. Component Temperature Requirements

Component	Minimum Operating Temperature	Maximum Operating Temperature
Payload Board (Computer + IMU)	-40°C	85°C
SDR	-40°C	85°C
Battery	-10°C	60°C

5.6.1. Preliminary 1D, Steady-State Model

The first model used to analyze the effects of temperature and insulation on the TOMCAT design was a steady-state, one-dimensional thermal model using resistance networks. This model accounts for radiation, convection, and conduction as forms of heat loss from the electronic components inside the ThinSat structure to the atmosphere.

a. Model Assumptions

- Steady-State.
- One-dimensional.
- Ambient temperature is derived from the standard atmosphere.
- Effects of sun loading and Earth's albedo can be neglected.
- Electronics dissipate the same amount of heat as their maximum wattage rating.
- Density and pressure effects can be neglected.

b. Method and Theory First, the electronic components were modeled with no insulation. The surface temperature of the components was calculated to find what the temperature would be if there was no thermal protection of any sort and the component was exposed to the ambient air. Figure 39 shows the mathematical method behind this model. The model is based off of a simple 1D heat balance, in which the heat generated from the electronics is equal to the heat dissipated by radiation and convection. The heat equations for convection and radiation depend on the convection coefficient, h_c , which depends on the amount of turbulence felt by the components, the surface area of the component, A_s , the emissivity of the material, ϵ , and the ambient temperature, T_∞ . The heat equations are set equal to each other and the surface temperature of the component can be found.

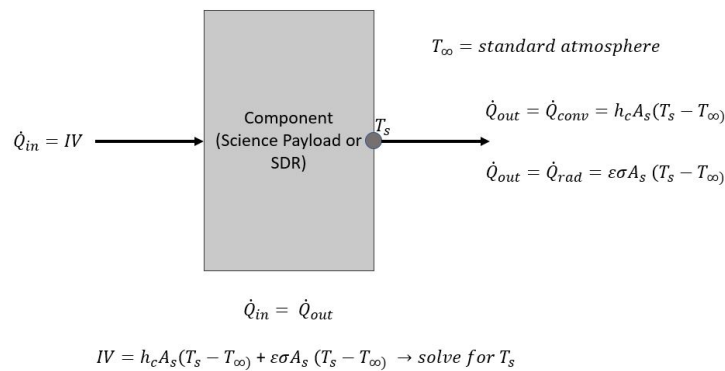


Figure 39. 1D, Steady-State Model With No Insulation

A similar approach is used to determine the effects of insulation. In the example shown in Figure 40 below, insulation is added to two sides of the electronic component. The same concept of heat balance can be utilized but, a thermal resistance network must be used in order to capture the effects of the heat loss due to conduction, convection, and radiation. Adding insulation essentially acts as a 'resistor' to heat flow, so less heat can be lost to the atmosphere via convection and radiation. The new conduction term depends on the surface area of the component, A_s , the thickness of the insulation, L , and the thermal conductivity of the insulation, k . The lower the thermal conductivity, the better the insulator. Once again, the heat loss terms were set equal to the heat gained due to the electronics, and the surface temperature of the electronics was solved for.

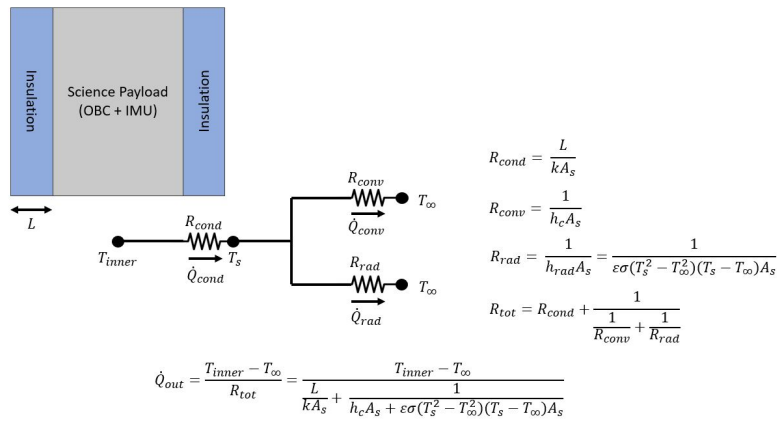


Figure 40. 1D, Steady-State Model With Insulation

c. Results Figure 41 shows the results of the 1D, steady-state analysis with no added insulation. It is evident that with no thermal protection, the electronic components reach or go below their minimum temperature ratings about halfway through the flight, at altitudes between 10km and 25km. This plot shows that there is a need for insulation.

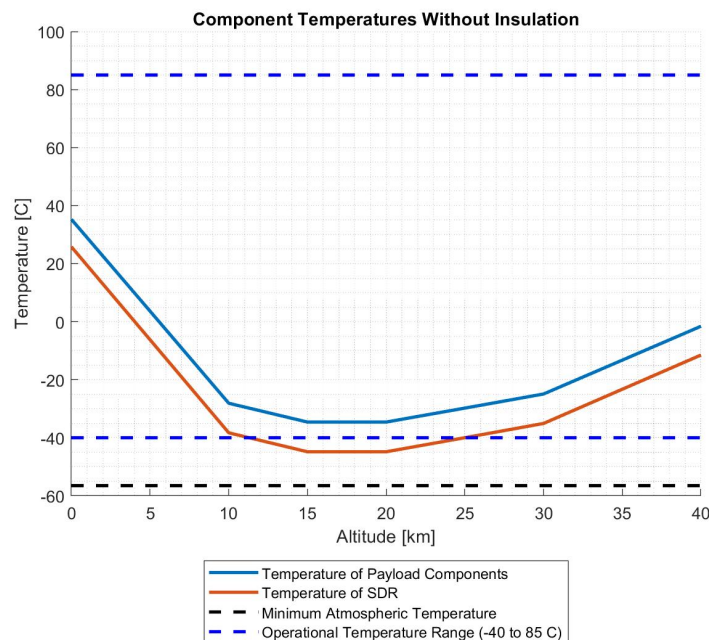


Figure 41. 1D, Steady-State Model Results With No Insulation

Figures 42 through 45 show the results of adding insulation to each electronic component using the method shown in Figure 40 above. Two different types of foam insulation were modeled to study the effects. In figures 42 and 43, aerogel insulation was used. In figures 44 and 45, polystyrene insulation was used. It is evident by these results that by increasing the thickness of the insulation, the temperature will rise significantly. Further, aerogel has a lower thermal conductivity than polystyrene. Therefore, temperatures of the components with aerogel insulation become much hotter than those with polystyrene.

These results also show that the 1D, steady-state model is flawed. These plots show that with the addition of insulation, temperatures of the components will rise to around 600°C and in the case of the results using aerogel insulation, the minimum temperature could be as high as 100°C. This shows that the addition of insulation is creating a scenario in which the system is essentially adiabatic, where no heat can escape and because of this, the internal temperature heats up to unrealistically high values.

The source of this error in the model comes from the fact that the surface areas of the components in question are very small. The heat equations depend directly on the surface area, and the smaller the surface area, the less heat can be lost through radiation and convection.

Though this model does show us that insulation will affect the internal temperatures and that the amount of insulation can influence how warm or cold the internal components become, it is not sufficient to give accurate data and therefore a new approach must be taken.

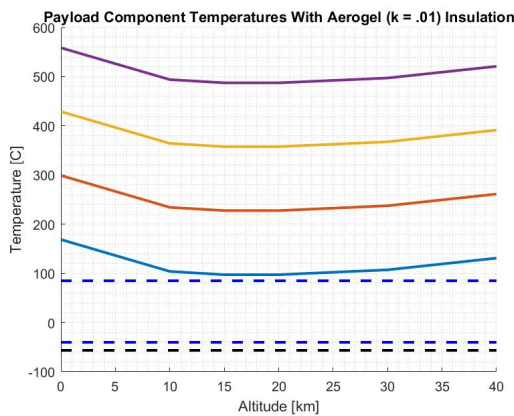


Figure 42. Payload Area Results With Aerogel Insulation

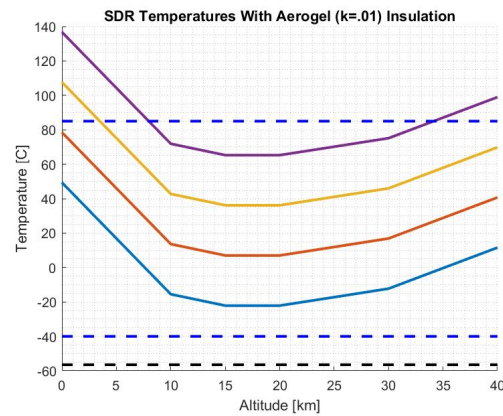


Figure 43. SDR Results with Aerogel Insulation

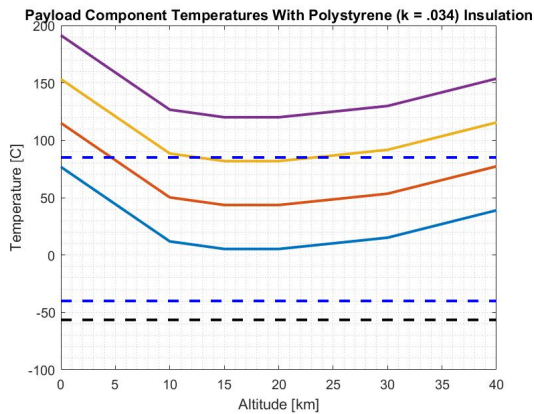


Figure 44. Payload Results with Polystyrene Insulation

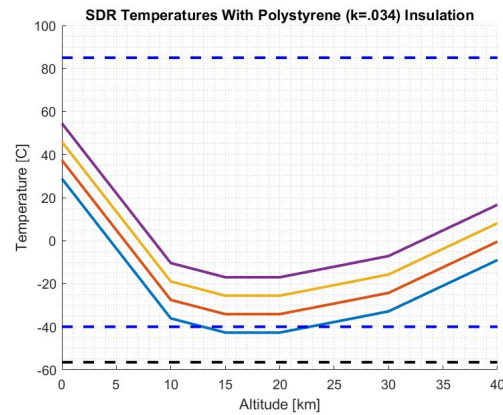
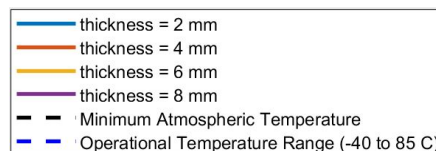


Figure 45. SDR Results with Polystyrene Insulation



5.6.2. Preliminary Testing

Because the steady-state 1D model is not a feasible option for the TOMCAT design, the team decided to run a test using a model of the TOMCAT design with power resistors replacing electronic components to model their heat output. Testing was chosen over higher-fidelity 3-dimensional, transient models due to the limited availability of software such as Thermal Desktop and the lack of team experience with these modeling softwares.

This test simulates flight conditions to create a more robust thermal model that can aid in further testing. This includes battery performance and degradation, which was left out in the preliminary 1D, steady-state thermal model.

a. Model Assumptions

- Electronics dissipate the same amount of heat as their maximum wattage.
- Heat generated by electronics can be simulated using power resistors.
- Convection effects are negligible.
- Effects of sun loading and Earth's albedo are negligible.

b. Method and Experimental Setup Figure 46 shows the test specimen set-up. Power resistors were placed in the locations in which electronics will be housed. Resistors were placed to represent the location of the main electronics in the ThinSat, these resistors are shown by the white boxes in the figure. These resistors are set to produce the same amount of heat that the electronics produce in an attempt to replicate the heat distribution within the ThinSat. For this test, it was assumed that the wattage of the electronic components is equal to the heat dissipated by the electronics. The orange circles represent placement of thermocouples. In order to get a full picture of the heat distribution throughout the ThinSat, thermocouples were placed inside the ThinSat where electronic components will be placed. Additionally, thermocouples were placed on the external faces of the insulation to capture the surface temperature of the ThinSat. Finally, a thermocouple was placed outside and away from the structure to measure the ambient temperature inside the cooler with dry ice.

Each thermocouple was labeled with a number corresponding to the numbers in Figure 46 and sent data to a LabView VI program on a desktop computer. Power was supplied via a desktop power supply. The voltage and current of the power supply depends on the resistance chosen for the power resistors.

Two plastic acrylic sheets were cut to the size of the ThinSat frame. To emulate the actual flight design of the ThinSat, two 3D printed plastic frames were utilized as the test specimens. Each frame had an acrylic backing and the power resistors were placed onto the acrylic inside the frame. The power resistors were secured using aluminum tape while the thermocouples were secured using Kapton tape. Once the internal components of the test specimen were placed and secured, the two ThinSat frames were stacked, with the acrylic covers on either side. The thermocouple wires and the power resistor wires were threaded out of the test specimen and plugged into their respective ports.

For the test, the ThinSat-insulation structure was placed in a cooler filled with dry ice. The test specimen was placed such that there was no direct contact with the dry ice. Power cables were run in and out of the cooler to supply power. The thermocouple cables were run out of the cooler and plugged into the DAQ computer.

The test was run twice, with the first test being a control experiment in which only the acrylic covers were used, while the second test included the insulation. The insulation chosen for the test was $\frac{3}{16}$ inch thick extruded polystyrene foam, with a thermal conductivity of $.03 \frac{W}{m \cdot K}$. Figure 47 shows the test specimen with attached power resistors and thermocouples.

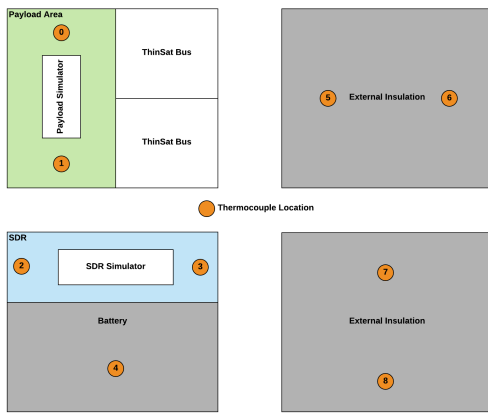


Figure 46. Test Setup For Preliminary Thermal Testing

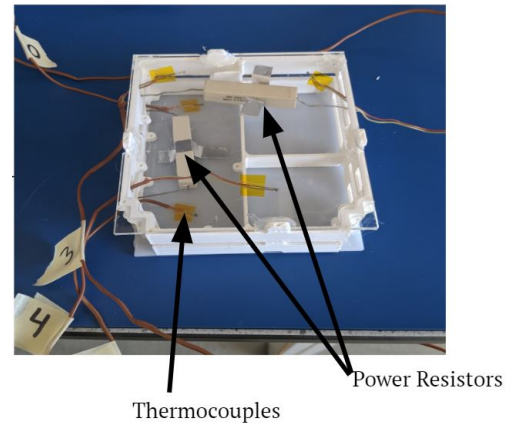


Figure 47. Actual Test Setup

c. Results The ambient temperature thermocouple read temperatures all the way down to -60°C , below the predicted coldest temperature during flight. This means that the temperature conditions were well replicated inside the cooler.

Figure 48 shows the test results for thermocouple 4, with its placement shown in figure 46. The blue line is the control test with no added insulation, while the blue line is the test with one layer of insulation. The red dashed lines represent the maximum and minimum temperature that the battery is allowed to reach without degrading performance. Adding insulation raises the temperature almost 15°C , from around -25°C without insulation to -10°C with insulation. It is evident by this plot that adding more insulation layers should be able to keep the battery warm enough. Insulating the battery directly could also help keep it warm. Further testing will explore more layers of insulation and/or different types of insulation.

Figure 49 shows the test results for thermocouples 0 and 1, with their placements shown in figure 46. The blue color lines represent the control test results with no insulation and the green colored line represents the insulation test results. The solid line indicates results from thermocouple 0 and the dashed line indicates results from thermocouple 1. These results show that even without insulation, the payload area will not reach its lower temperature limit and therefore insulation will only help keep it warmer.

Figure 50 shows the test results for thermocouples 2 and 3, with their placements shown in figure 46. The blue color lines represent the control test results with no insulation and the green colored line represents the insulation test results. The solid line indicates results from thermocouple 2 and the dashed line indicates results from thermocouple 3. It is important to note that thermocouple 2 came loose from the test specimen during the insulation test and its results are therefore not shown on this plot. Once again, it is evident from this plot that even with no insulation, the SDR does not reach its lower temperature limit and therefore insulation will only help keep it warmer.

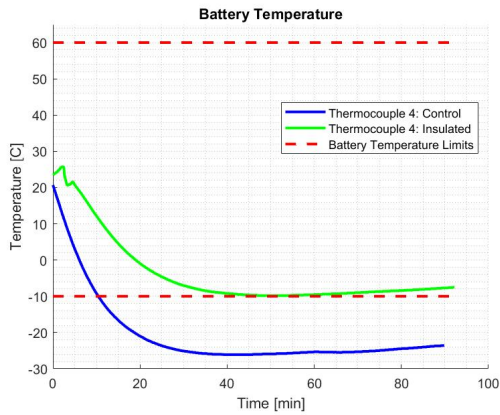


Figure 48. Test Results For Battery Area

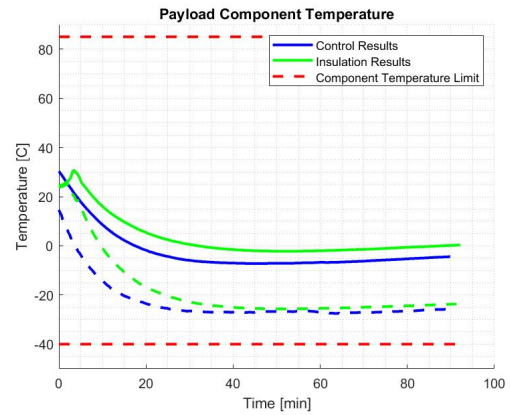


Figure 49. Test Results For Payload Area

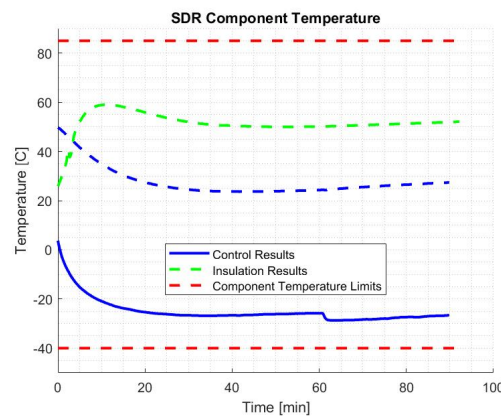


Figure 50. Test Results For SDR Area

This test shows that with an adequate amount of insulation, design requirement 3.1 will be met, as all electronic components and the battery are able to survive the coldest temperatures of flight and never reach temperatures above their upper temperature limit. The main concern coming out of these results is the lack of margin on the temperature of the battery (figure 48). The temperature reached exactly -10°C , right at the lower temperature limit. Any uncertainties in this model could cause that temperature to go lower. This risk and its mitigation will be discussed in Section 8.

5.7. Power

Jake Schroeder

5.7.1. Power Budget

The five main components of the ThinSat are the Inertial Measurement Unit, Software Defined Radio, Low Noise Amplifier, Power Amplifier, and Micro-controller Unit. Each component has its own voltage and current draw requirements to run properly, so a power budget has been created to keep track of the individual power requirements of each device and determine the total current draw and energy required to sustain a two hour balloon flight. In Table 37, the science subsystem is shown in green, the communications subsystem in red, and the flight computer in blue. The total current draw, power draw, and energy required for each

subsystem and the ThinSat as a whole is displayed at the bottom section of the table.

Component	Voltage Req. [V]	Max Current Draw [A]	Max Power Draw [W]	Energy Required (2hrs) [Wh]
XSens MTi-1 IMU	3V (3.3V Pin)	0.015	.045 W	0.09
	Science Total:	0.015	0.045	0.09
SDR	5	0.9	4.5	9
LNA	5	0.152	0.76	1.52
Power Amp	3	0.05	0.15	0.3
	Comms Total:	1.102	5.41	10.82
MCU	5	0.5	2.5	5
	FSW/OBC Total:	0.5	2.5	5
	Total:	1.617	7.955	15.91
	+50% Margin:	2.4255	11.9325	23.865

Table 37. ThinSat Power Budget

There are two different voltages required by the ThinSat components. Therefore, two types of DC/DC converters will be required to first step down the voltage from the battery to 5V and then to 3V. The values given for each component's current draw are based on each device's maximum current draw from their respective specifications. The software defined radio and micro-controller unit draw by far the largest currents, at 900mA, and 500mA respectively. While it is not expected that these maximum currents will be drawn continuously, the SDR will be transmitting and receiving and the MCU will be operating throughout the entire flight, so it is reasonable to expect that their actual current draws won't be significantly lower than the values shown in the power budget.

The maximum power draw of each component is calculated by multiplying its required voltage by its maximum current draw. These values give an idea of the maximum power that will be drawn from the battery during the mission operation. It is important to note that the voltage of the battery will be decreasing as it discharges throughout the flight. In order to maintain the voltages required for each of the components, the DC/DC converters will draw a larger current as battery voltage decreases. Therefore, the maximum current draw from the battery will increase over time, whereas the maximum power draw from the battery will remain constant.

The typical maximum duration for a Gateway to Space high altitude balloon flight is two hours. Therefore, the battery must be able to sustain the maximum power draw requirements for at least two hours. The total energy required for the mission is found by multiplying the power draw by the duration of the flight, which gives the minimum required capacity for the battery.

5.7.2. Margin and Built in Capacity

It is important to note that the battery capacity required by the components alone is 15.91 Wh, but the power requirements for the system as a whole at high altitude conditions may be more. Due to power conversion inefficiencies, small unaccounted for power draws, and losses in battery capacity due to low temperatures, built in additional battery capacity is required.

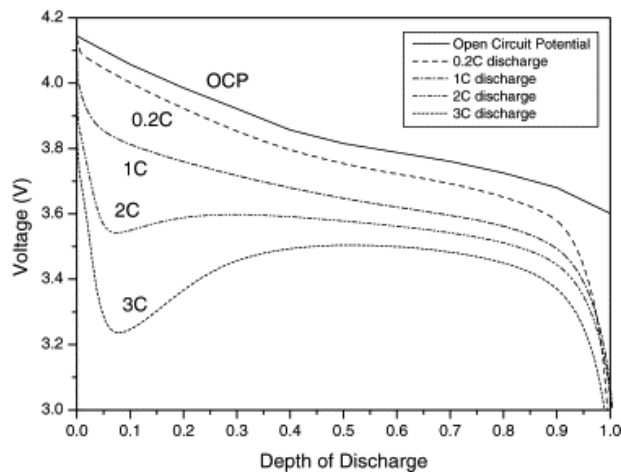


Figure 51. Voltage vs Depth of Discharge at Varying Discharge Rates [7]

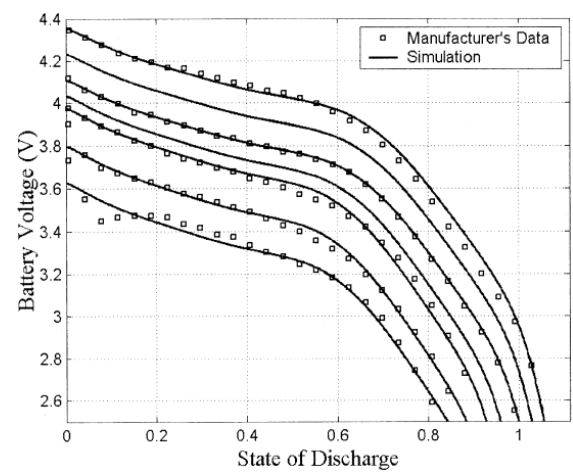


Figure 52. Voltage vs Depth of Discharge at Varying Temperatures [8]

Figures 51 and 52 plot the voltage of a lithium ion battery over the course of its depth of discharge. Depth (or state) of discharge describes the percentage of charge that has been removed from the battery, meaning at depth of discharge of 1.0, the battery has been fully discharged. Both of these plots show that as the battery is discharged, voltage decreases, which requires the use of DC/DC converters to keep the voltage to each of the components constant.

Figure 51 plots the voltage output of the lithium ion battery at different rates of discharge (C rates), where the C rate is defined as the percentage of the battery discharged per hour. Given that the TOMCAT flight is 2 hours, and assuming that the battery is completely discharged in that amount of time, the C rate of the TOMCAT battery is a maximum of 0.5. As shown in the figure, lower C rates allow for the battery to output at higher voltages, and therefore allow for a greater power output. With a relatively low C rate of less than 0.5, the battery for the TOMCAT mission does not need to add built in capacity for high discharge rates.

Figure 52 shows the voltage output of a lithium ion battery versus state of discharge at varying temperatures. From top to bottom, the temperatures plotted are: 45°C, 34°C, 23°C, 10°C, 0°C, -10°C, and -20°C., at a discharge rate of 0.7 C. In this plot, the battery is discharged in each test until it reaches 2.5 V. From this state of discharge plot, the effective depths of discharge for a lithium ion battery discharged to 3.0 V at varying temperatures can be gathered. Table 38 below takes the data from figure 52 and shows the effective battery capacity of a lithium ion at decreasing temperatures.

Temperature	Battery Capacity
10°C	83%
0°C	80%
-10°C	70%
-20°C	65%

Table 38. Effect of Temperature on Battery Capacity [8]

From the first thermal test discussed above, it was shown that the temperature of the battery could reach as low as -10°C with insulation. Since the mission will not be employing the use of active thermal control due to its increased complexity, the battery must be able to power the mission even at very low temperatures. At -10°C, the battery is still within its operational range, but at a cost to its effective battery capacity. From table 38, it is expected that at this temperature, the battery is capable of discharging to 70% its specified

capacity. Therefore, an additional 30% capacity will be built into the power budget.

Since power converters will be employed to maintain a constant voltage for each of the ThinSat components, a small power loss due to DC/DC conversion must be accounted for. The efficiency of a DC/DC converter is typically around 95%, but depends on the voltage received from the battery, which changes over time. To account for conversion losses, a 10% capacity will be built into the power budget. Furthermore, an additional 10% capacity will be built into the power budget to account for any other power losses, such as increased demand from individual components, and power draw from the payload board. These additional battery capacity accommodations are shown in the table below.

Accommodation	Required Additional Capacity
Low Temperatures	30%
DC/DC Converter Efficiency Losses	>10%
Other Efficiency Losses	10%
Total:	50%

Table 39. Calculation of Additional Battery Capacity Accommodations

5.7.3. Power Distribution

Figure 53 shows the connections and power requirements of each of the on-board components.

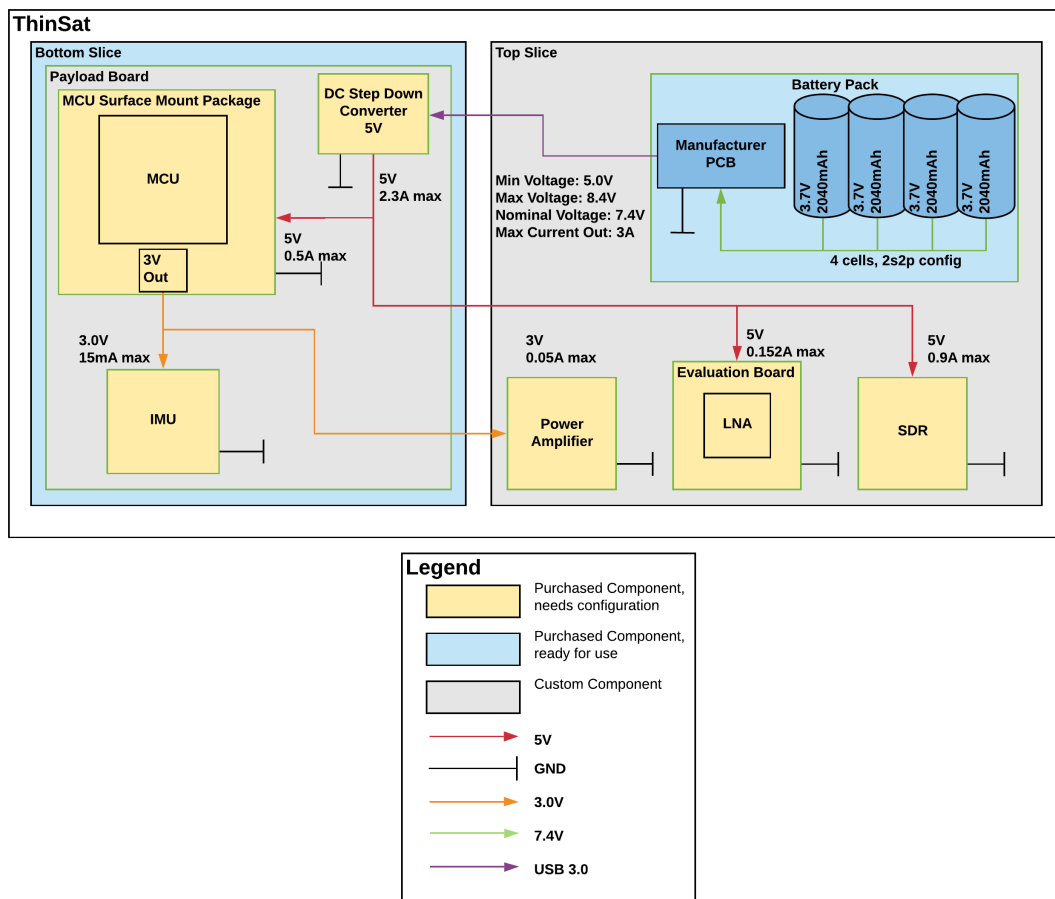


Figure 53. ThinSat Power System FBD

a. Manufacturer Provided PCB To prevent damage to the payload components and the battery itself, a PCB provided by the battery manufacturer will be included attached to the battery. The PCB allows for the batteries to operate in a two series, two parallel configuration, increasing the output voltage of the battery to 7.4V. The PCB allows for safe discharging, with a fuse that disconnects the battery above 3A, which is much greater than the current expected to be drawn at any time during operation. The PCB also regulates battery voltage, preventing the pack as a whole from discharging below 5V, which keeps each individual cell from decreasing below 2.5V, and it allows for safe charging and discharging of the pack in its 2s2p configuration without needing to separate the batteries during charging. It was important to find a PCB that was included with the battery as designed by the manufacturer, rather than to design a fully custom PCB for the battery, which would have an increased likelihood of possibly damaging the battery or other components.

b. Payload Board Power Regulation The payload board, which houses the microcontroller unit and inertial measurement unit, as well as all the power distribution connections, will connect the battery to a 5V DC/DC step down converter capable of supplying 2.3A to the microcontroller surface mount package, the low noise amplifier evaluation board, and the software defined radio. There is a built in 3V out pin on the microcontroller surface mount package, which will be used to provide power to the inertial measurement unit and the power amplifier. The DC/DC step down converters ensure that each component is delivered the proper voltage, and allow for the required current draws. Since the battery is regulated by its own PCB to disconnect once the pack reaches 5V, no step up converter is necessary, which improves the overall efficiency of the power distribution system.

5.7.4. Battery Selection

Because the payload volume of the ThinSat is so small, considerable size constraints for the battery had to be taken into account. The lithium ion 18500 cell is a 3.7V, 2040 mAh battery with dimensions that allow for four batteries to be placed side by side within the payload. It was chosen for its very high energy density compared to other lithium ion batteries and for its shape, which allows for it to be placed snugly within the payload. Using four batteries, the nominal capacity is 30.19 Wh, which is significantly above the 23.86 Wh required capacity for the two hour flight. With the batteries in a 2s2p configuration, stepping the voltage down is required instead of stepping the voltage up, which is less efficient and more demanding on the current draw of the batteries.

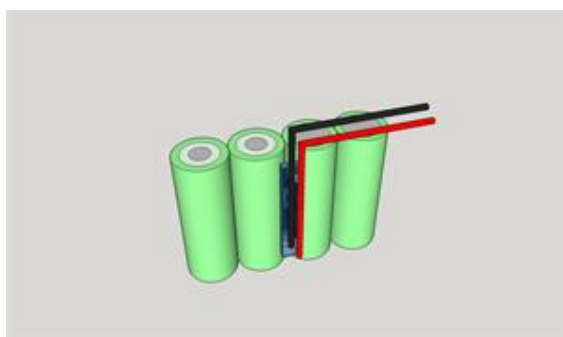


Figure 54. Battery with PCB CAD from Manufacturer

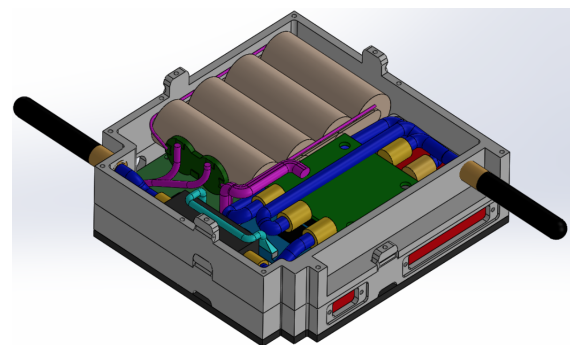


Figure 55. Battery within Payload

5.7.5. Ground Station Power

The ground station consists of four main components: the ground software defined radio, low noise amplifier, power amplifier, and a Linux based laptop to power and interface with the communications equipment. The power requirements for the ground station components are shown in table 40.

Component	Voltage Req. [V]	Max Current Draw [A]	Max Power Draw [W]	Energy Required (3hrs) [Wh]
Ground SDR	5	0.9	4.5	13.5
Ground LNA	5	0.152	0.76	2.28
Ground Power Amp	5	0.1	0.5	1.5
Total:		1.152	5.76	17.28
50% Margin:		1.728	8.64	25.92

Table 40. Ground Station Power Budget

The Functional Block Diagram for the ground station power system is shown below in figure 56.

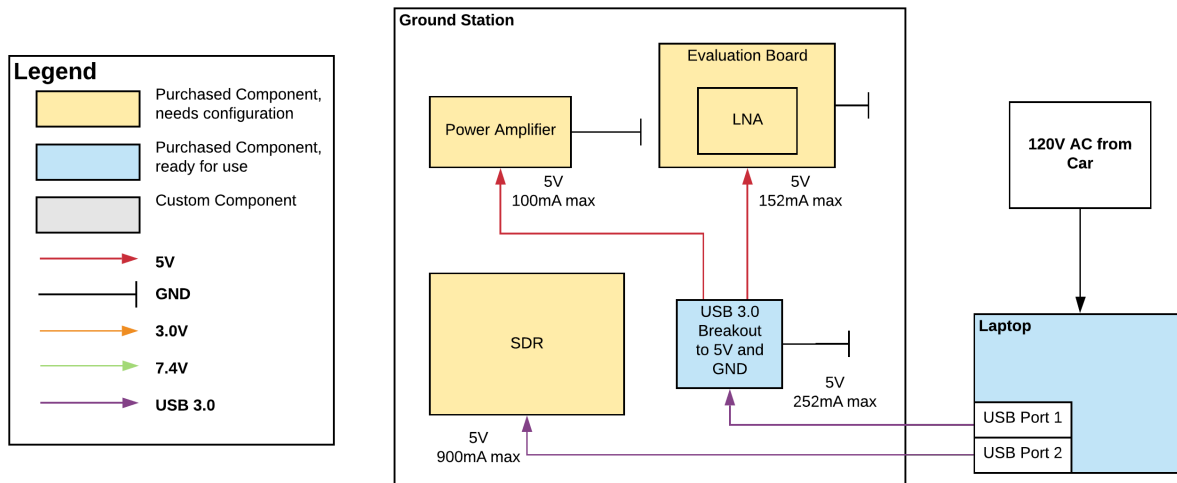


Figure 56. ThinSat Power System FBD

Each of the ground station components requires a 5V power supply, and no more than 900mA of current. A USB 3.0 port can supply up to 900mA of power at 5V, meaning that the software defined radio can be directly connected to the laptop to receive power and to transfer data. The power amplifier and low noise amplifier evaluation board also need 5V with a combined current draw significantly less than the 900mA maximum, meaning a second USB 3.0 port can be used to power these two components separately from the SDR. A simple USB 3.0 breakout to 5V and ground will be used to distribute the power from the laptop to the power amplifier and LNA. A car will always be onsite with the ground station, allowing the laptop to plug in if necessary.

5.8. Structures

Cole Kenny and Matthew McCallum

The structure of the ThinSat has a few key roles. The structure needs to be able to carry all of the components for full system testing, ensure that the components remain connected to each other, and allow the components to survive the impact at the end of the balloon flight. Ensuring that the structure can carry all of the components is done with a CAD model of the system. The CAD model allows for all of the components to be test fit before purchasing the components and physically putting them in place. The CAD model also allows us to determine how the components will be connected to the frame and to each other before anything has to be purchased. The first component designed in CAD software was the top frame. The bottom frame is the one that we are buying from NSL and is designed to function by itself and not attached to another frame. This means that we had to design the top frame to merge with the bottom frame.

To do this the walls of the frame were extended down to meet with the upper edge of the bottom frame. This makes the whole system look more like it is one single unit, and will help to trap warm air inside of the system to help keep the components warm. The top frame was also modified to account for the antenna. The antenna needed places to be mounted and they also needed places where they could be stored for launch. The antenna mounts are simple holes in the frame that the base of the antenna can attach to. For our project the antenna will not move, but for a rocket launch the antenna cannot stick out of the frame. To ensure that the antenna will not be a problem for the customer after the project is over, we accounted for space that the antenna could be folded into for a rocket launch. Ensuring that the components survive impact is done through modeling and testing, which is discussed in the next sections.

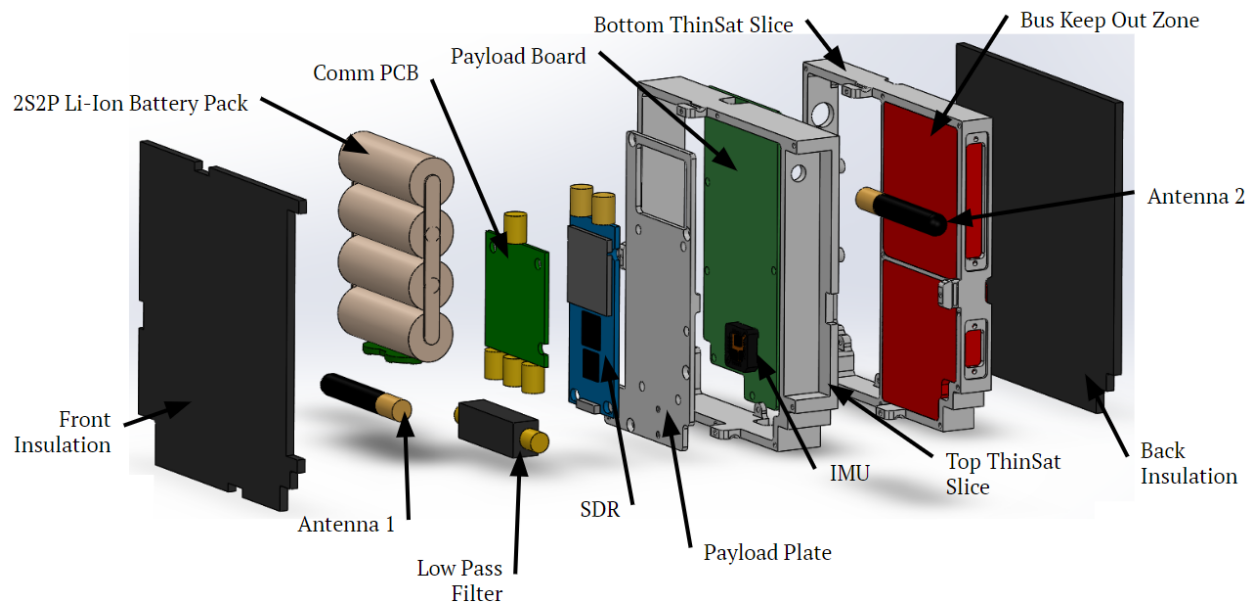


Figure 57. Exploded view of the CAD model with major components labeled

One other critical element to be designed for was being able to interface with the Gateway to Space balloon flight string. This is accomplished by designing a slot on the front of the ThinSat casing to slide in a Teflon tube. This is compliant with the Gateway to Space Balloon Payload ICD, which calls for a 4mm-6.5mm tube for the flight string to pass through. The tube needs to be as close to the center of mass as possible, but since there is not enough internal space for the tube to be able to pass through the center of the ThinSat, the tube is allowed to pass right over the top of the satellite frame and a bolt pattern is provided to add additional mass to shift the center of mass as needed.

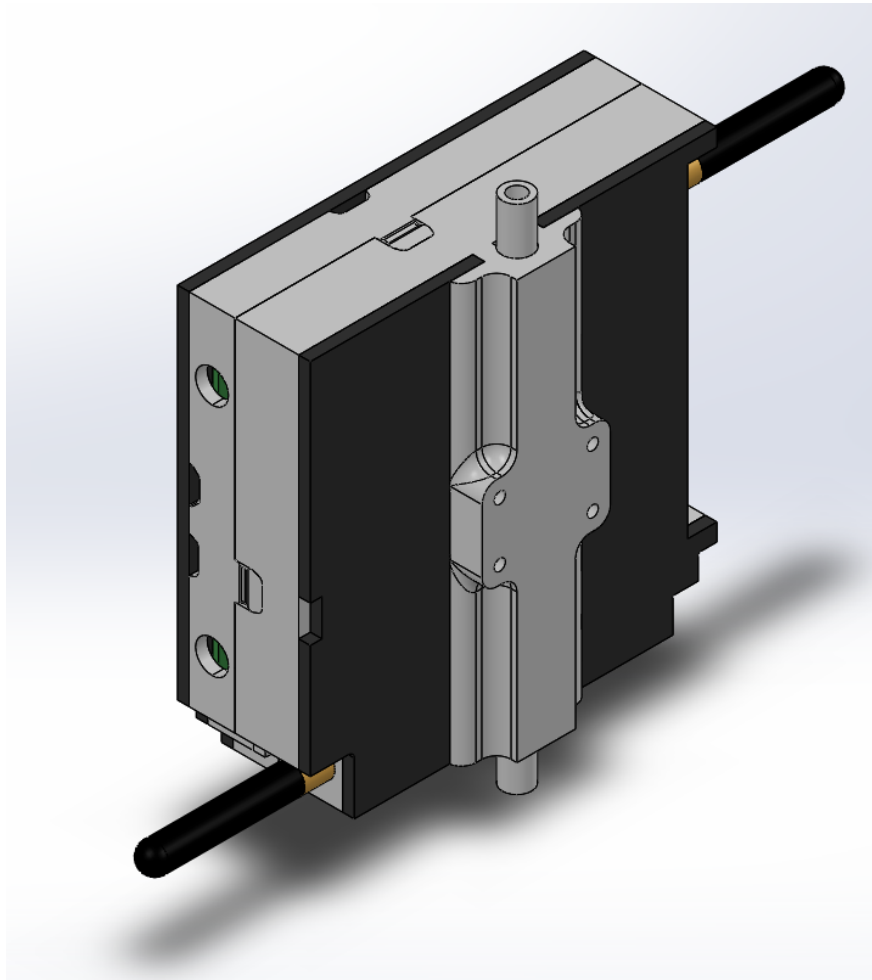


Figure 58. CAD model of the ThinSat with the tube used to attach to the Gateway to Space balloon flight

5.8.1. Requirements

The primary requirement for the ThinSat Structure is functional requirement 3.2

3.2: The Payload shall withstand the impact from a balloon launch (estimated 9 m/s from vendor).

The purpose of this requirement is to ensure that the system remains in an operational state for when it is handed over to the customer. The other roles of the ThinSat structure do not have requirements explicitly stated, but are implied by what a structure should do for any project.

There are two main concerns for meeting requirement 3.2. The first is that the acceleration of the impact causes damage to a component. The second is a component breaking free and impacting another component, causing damage. The first concern is dealt with by choosing components that can withstand the acceleration of impact. All of the components that will go inside the ThinSat can withstand hundreds or thousands of g's of acceleration. This ensures that the acceleration itself will not cause any damage to components. The second concern is dealt with by modeling the component fasteners and testing the system with mass simulates.

5.8.2. Modeling

The main problem with modeling the impact at the end of the balloon flight came from trying to predict the forces involved. The forces were predicted by assuming all of the kinetic energy of the ThinSat went into compressing the frame and calculating the maximum forces that were produced. The connections holding the components to the frame were modelled with multiple methods, but every time the predicted forces were used, the model said that the connections were several times too weak. The only model that returned results that looked somewhat accurate was using the knowledge of what NSL said would survive the impact, and extrapolating that to our components. The model shows that we should be able to connect our components to the frame in a way that will ensure that they will remain in place through the impact. However, the problems with this method are that it is limited to the connection type that NSL uses, and there are a lot of unknowns like the factor of safety that make the result less trustworthy. This leads to the necessity of testing the structure to verify that the components will stay in place through the impact.

5.9. Science

The specific inertial measurement unit utilized by the team will be the Xsens MTi-3 Attitude and Heading Reference System (AHRS) from the 1-Series product line. This product provides the team with a high level of accuracy in its acceleration, rotation, and magnetic data that is valuable in both endo- and exo-atmospheric conditions. This model additionally features the XKF3 Kalman filter algorithm that outputs three-dimensional orientation data in a gravity-referenced frame. This includes drift-free roll, pitch, and true magnetic North-referenced yaw, all supported by a dynamic filter profile.

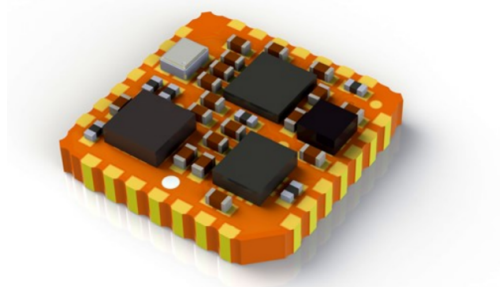


Figure 59. Xsens MT-i 3 AHRS

The sensor meets the team's physical requirements to provide reliable data, being individually calibrated by Xsens. Having sensor data provided in an inertially-referenced frame supplies the team with accessible and easily-applicable information that has the potential to satisfy a wide variety of missions. The MTi-3 AHRS can successfully accomplish state estimation, true magnetic North-referenced navigation, and more.

This section will discuss the MT-i 3's requirements, its physical integration onto the printed circuit board (PCB), what mission it will accomplish, and the code it will execute to do it.

5.9.1. Requirements

One of the reasons an IMU was selected as a scientific sensor for this mission was its relatively marginal power draw and wide envelope of operational temperatures.

The Xsens 1-Series draws 15 mA between 2.16-3.45V during operation. These power requirements are readily supplied by the external battery, and do not drive the power budget higher, like the software-defined radios do. The IMU is not a limiting factor for power on this mission, making it an acceptable choice.

Thermally, the Xsens 1-Series can operate between temperatures of -40°C and 80°C. This mission is concerned with the bottom end of that spectrum, and once again, the IMU is not the limiting factor across the TOMCAT system (the battery must maintain temperatures above -10°C). Preliminary thermal testing has shown that even a thin layer of insulation can protect the payload board from dipping below this minimum operational temperature; there are no valid environmental restrictions on using this specific IMU.

The IMU will not be hindered by its power and temperature requirements during the TOMCAT mission. Both provide acceptable values that are feasible for the full system to support.

5.9.2. Physical Integration

The IMU will be attached to the PCB using a 3D-printed mount, and these components together form part of the "payload board". This payload board is located on the "bottom slice" and houses the microcontroller unit (MCU) and the IMU with its mount on TOMCAT's custom PCB. The mount is visible on the 3D model of the full TOMCAT satellite. With respect to the PCB, the IMU will be mounted in a location such that it does not align with regions of high mechanical stress in the PCB. Its location will be advised by the hardware integration manual provided by Xsens, which specifies tactics for avoiding mechanical stress such as reducing anchor points on the PCB. These considerations will be contemplated further during the manufacturing and integration phase of TOMCAT development.

The IMU itself features 28 pins, 7 per side, which allow it to physically interface with the MCU and PCB. A top-down diagram showcasing the pins is shown below:

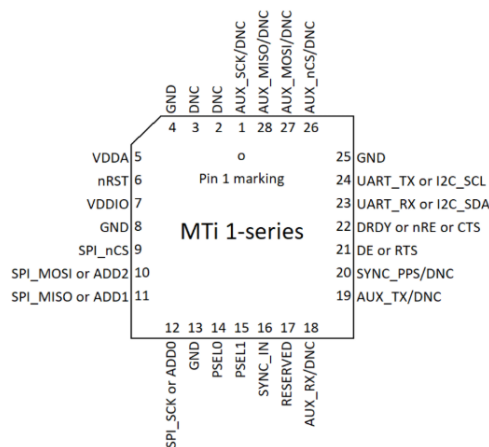


Figure 60. MTi 1-Series Pin Diagram

The IMU has the capability to communicate using SPI, UART, and I²C protocols, and the team has selected to employ I²C, a decision made by the PCB team on account of its relative familiarity and resistance to noise relative to SPI and UART. The pins in particular that will be configured for the TOMCAT mission can be seen below, in the payload board functional block diagram:

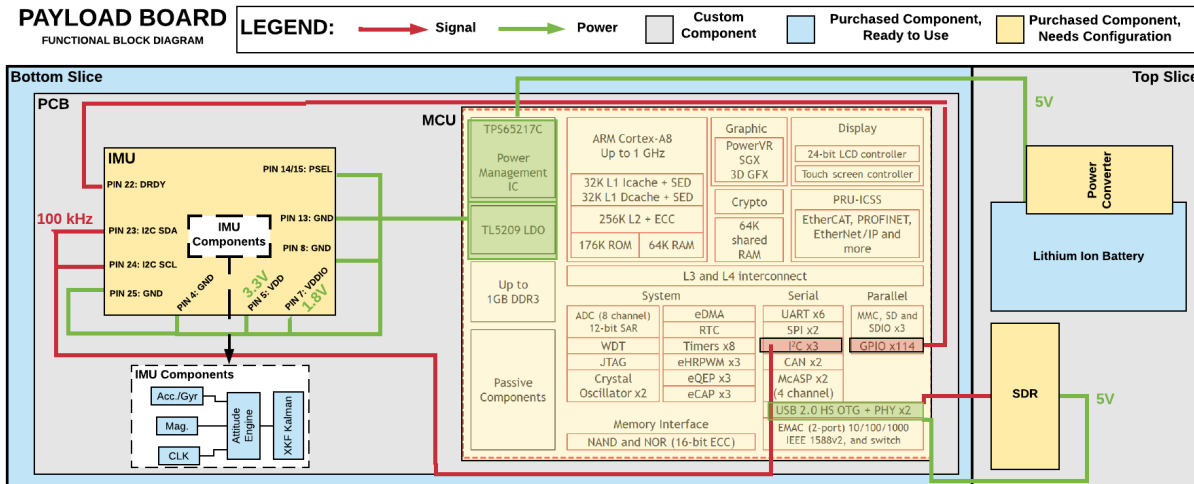


Figure 61. Payload Board Functional Block Diagram

For power, pins 5 and 7 will take in 3.3V and 1.8V respectively. Pins 4, 8, 13, and 25 will act as grounding for the IMU. Pins 14 and 15 will both be connected to power, as this configuration sets the communication protocol to I²C (different combinations of power/ground connections set to SPI, UART half-, and UART full-duplex).

To communicate, the data ready (DRDY) pin will notify the MCU when data is available, and the serial data input/output (I2C_SDA) and serial clock input (I2C_SCL) pins will send/receive messages to and from the MCU at 100 kHz.

The components within the IMU, showcased in the white box, require no integration or configuration, and will generate and process the acceleration, rotation, and magnetic data throughout the flight.

5.9.3. Mission Selection

The primary function of the IMU in the context of this project is to simply provide a source of data for generating telemetry, which will be accomplished by the on-board MCU and SDR. Thus, the team's mission within the scope of the TOMCAT mission is straightforward: determine whether the satellite is in ascent or descent phase during its flight.

The MTi-3 will be an excellent asset for further missions using this ThinSat architecture, however its functionalities will be somewhat underplayed for this project. The needs to be able to verify, in a straightforward manner, whether or not the data being transmitted from the satellite is accurate or erroneous. The major complexity and primary triumph of the TOMCAT mission comes from its communications system, and the team wants to minimize the potential for error coming from any other source. Isolating the communications as the likeliest point of failure, if any, will make product development much more streamlined and achievable within the remaining 5 month time frame.

Additionally, the success or failure of the mission can easily be verified by juxtaposition with an on-board camera, which will be discussed later, in Validation & Verification.

5.9.4. Coding Algorithm

Because the MTi-3 AHRS features the XFK3 fusion algorithm, an Xsens proprietary Kalman filtering system, the three-dimensional data will be reported in an inertially-fixed gravitational frame, which makes the

process of determining ascent versus descent simplistic.

The 3D acceleration data will be the primary indicator of ascent versus descent, specifically the Z-axis, which will be aligned with the force of gravity. When the inertial Z acceleration value exceeds $+2\text{ g}$'s, as informed by the ASEN 1400 data acquired from previous launches, the IMU will output 'ascent' phase. This phase will continue until burst occurs, where acceleration will once again exceed 2 g 's, this time in the inertial -Z. From that point onward, the phase will read 'descent' until landing has occurred, where all 3 acceleration inputs and 3 rotational inputs will zero out within $\pm 0.1\text{ g}$ and $\pm 0.1\text{ rad/s}$. When this condition is reached, the IMU will know that the post-flight phase has been reached. A sample ASEN 1400 Gateway to Space dataset can be seen below.

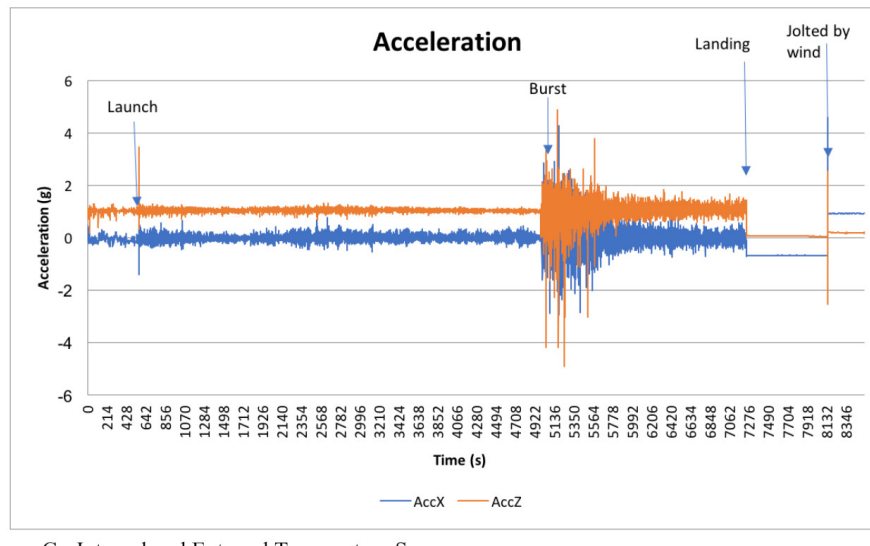


Figure 62. ASEN 1400 Flight Data

The Xsens IMU is compatible with Arduino, Python, MATLAB, C++, and C#. This breadth of languages gives the team plenty of freedom to select whichever language performs the mission best with the least expense to computational power.

6. Verification and Validation

6.1. Communications Hardware

6.1.1. Verification

In order to verify that the individual components of the communications hardware system function as designed, the team developed multiple tests designed to isolate specific components of the overall system.

a. Transmission Bench Test In order to verify the transmission capabilities of the hardware, the team developed a bench test. During the bench test, the LimeSDR Mini will be plugged into the USB port of a team member's laptop and configured to produce a signal across the desired frequency range of 902 to 928 MHz. The output will be hooked up to the laboratory oscilloscope to verify the frequency of the generated waveform as well as the power of the signal as it leaves the SDR. This test will reassure the team that the SDR has been configured appropriately, as it is designed to communicate at a wide range of frequencies and needs to be configured to a specific operational frequency. It will also verify the expected strength of the signal as it leaves the SDR. After this, the low noise amplifier will be incorporated into the system. The resulting signal power will be measured again in order to confirm that the LNA has been configured appropriately to

produce the desired gain. The setup for this test is shown in Figure 63 along with descriptions of what the team is looking for at each step along the way. These expected values were previously discussed in the link budget analysis.

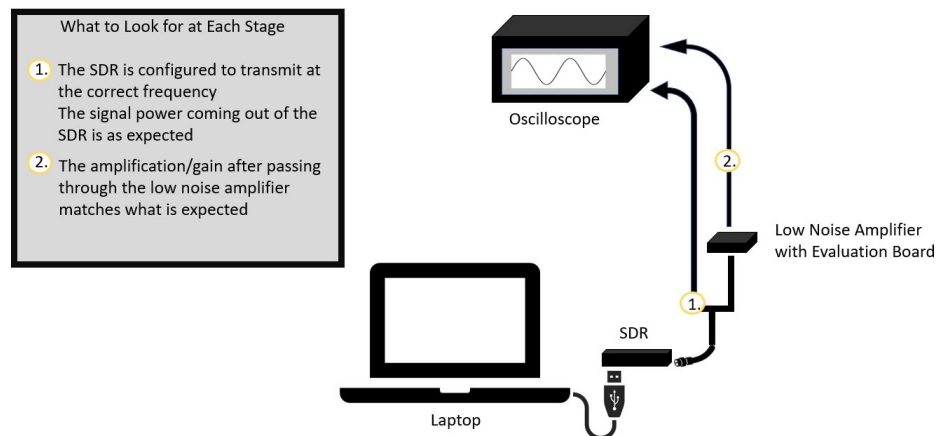


Figure 63. Transmission Bench Test Diagram

b. Reception Bench Test In order to verify the capability of the SDR to receive wave forms within the prescribed frequency band, the team will conduct a reception bench test. This bench test will also be used to verify the reception RF chain. It will involve configuring the function generator to output a wave with a center frequency of 915 MHz and transfer the resulting signal from the function generator, through the low noise amplifier, through the low pass filter, and finally arriving at the SDR. The LimeSDR Mini will be plugged in via USB port to a team member's laptop in order to determine what is being received by the radio. In order to verify that the gain through the low noise amplifier and the losses through the low pass filter are as predicted, the team will measure the received power before and after each component in the chain. As shown in Figure 64, the cable will be hooked up to a splitter. One output will continue through the chain and the other will go to the desktop oscilloscope to measure the power received after each stage. Success will be defined by whether or not the results match within 15% of the RF chain analysis. The frequency of the wave output from the function generator will be varied at intervals both below and above the expected frequency in order to test the ability of the low pass filter to filter out the signals coming it at undesired frequencies. This will be verified by determining what frequencies are present on the other side of the low pass filter.

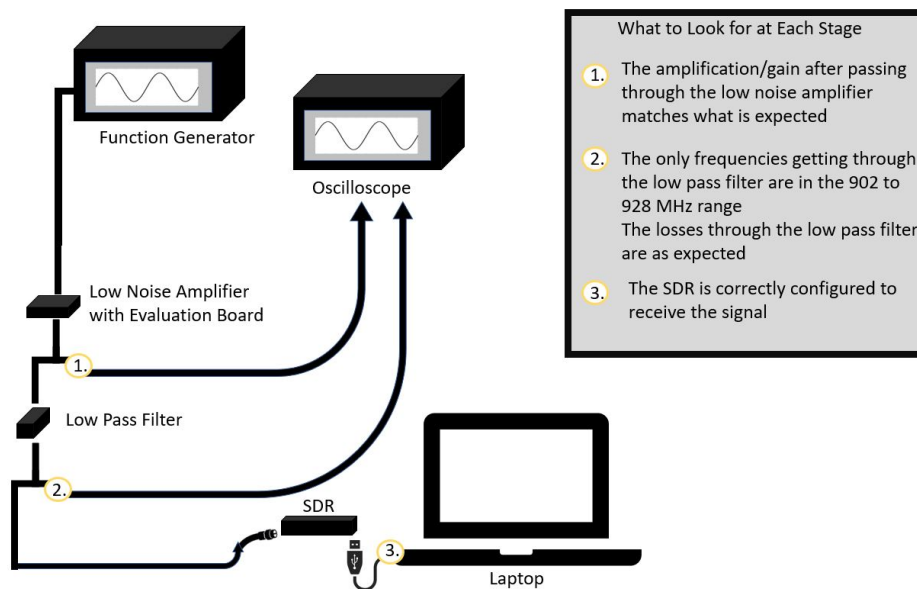


Figure 64. Reception Bench Test Diagram

c. Short Range Wireless Test A short range test will be implemented to verify the wireless communication between the two communications hardware systems. One system will be set up as the ground station and the other will simulate the on-board system. The team deemed it important to first conduct a short range test to isolate the wireless communication component without yet accounting for problems that could present themselves during a long range test, such as more noise interference and space losses.

During this test, the team will be measuring the bit error rate and data rate, and comparing these values to the link budget analysis. This will serve as a quantitative measure. However, the team will also be analyzing the qualitative aspects of the general functionality of the system. This includes the ability of the SDR's to handle the incoming signals. A lot of this has to do with the communications software and will be discussed in further detail later.

For the hardware, the team is primarily concerned with the SDR's abilities to communicate between each other without connecting wires.

During this test, the bit error rate will be measured as a function of the downlink rate. The success of this test will be dependent on if the bit error rate matches the specified bit error rate for the design as previously discussed in the detailed design section.

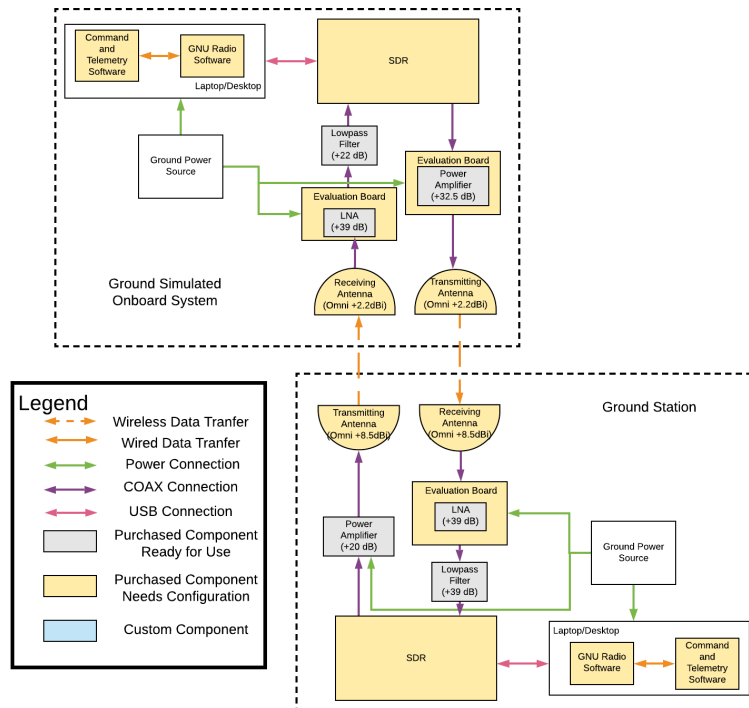


Figure 65. Communications System Short Range Test FBD

The functional block diagram for this short range test can be seen in Figure 65. The ground station in this image can be noted to be identical to the one in the final system FBD. However, the simulated on-board system will not exactly match the on-board system. This allows for the team to test the short range capability without having a fully assembled on-board system. The primary differences are that the team will use an evaluation board for the configuration of the low noise amplifiers rather than needing a completed custom PCB for this stage of testing. In addition to this, the team can use a power source other than that designed for the on-board system. Another key difference is that a laptop or desktop can be used rather than requiring the full integration of the flight software. Each of the components not accounted for here will have their own tests to ensure their functionality.

6.1.2. Validation

The customer's driving requirement for the communications hardware system is conveyed in Functional Requirement 2:

FR2: The payload shall communicate with the ground system during the balloon flight.

In order to validate that the design meets the needs of the customer, the team developed a long range test.

a. Long Range Wireless Test The long range test for the communications hardware system is similar to the short range test but at a longer range and with added attenuation. The team will have a team with the designed ground station on CU Boulder's East Campus and a team with the simulated on-board system on the top of Boulder's Flatirons. This gives a range of 5 km. Since the actual balloon flight communication range will be close to 100 km, the team will be adding 140dB of attenuation in order to simulate this range. The functional block diagram for the long range test looks very similar to the short range test but with attenuation added in as shown in Figure 66.

During this test the team will once again be measuring the strength of the incoming signal and thus verifying the link budget model. The team will also be measuring data rate and bit error rate in order to

add additional verification to the link budget analysis. The expected values and reasoning behind them were discussed in detail in the link budget section above.

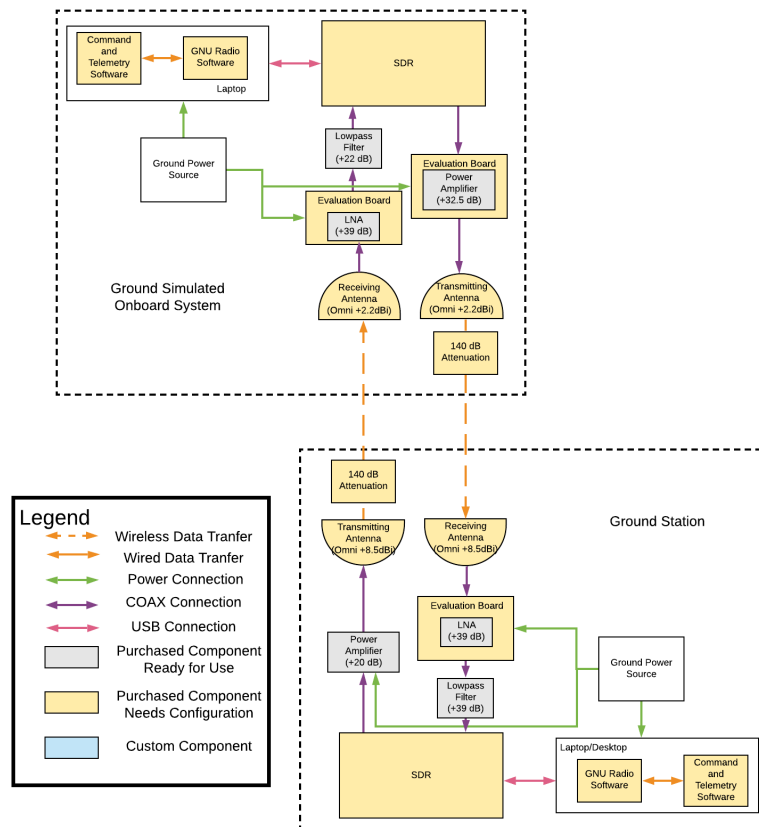


Figure 66. Communications System Long Range Test FBD

It can be noted that the full ThinSat system does not need to be complete in order to conduct this test. This allows the communications team to start testing specifically the communications hardware and software without needing the full on board flight software developed.

6.2. Communications Software

6.2.1. Verification

a. SDR-SDR bench test with attenuator To verify the the signal processing model, a test will be performed where data will be sent between two SDRs wired together. The test setup will consist of two SDRs connected to a computer running only GNU Radio. The two SDRs will be connected by SMA cables with an attenuator in the middle, which will lower the power of the transmitted signal to prevent damage to the receiving SDR. Data will be sent from GNU Radio to the transmitting SDR, then to the receiving SDR. Finally, the received signal will pass through the signal processing chain in GNU radio and be converted back into bits. The bit error rate and data rate will be measured to verify that the signal processing software is performing to the design specifications.

b. GSW integration test To verify the functionality of the software interface between GNU Radio and the flight and ground software, a data transfer test will be performed for each of them. Data will be streamed

over a UDP connection to and from GNU radio and the respective software. Again, the bit error rate and data rate will be measured to judge the success of the test.

6.2.2. Validation

a. Running it on the OBC To test the communication software's functionality on the OBC, a test will be performed similar to the SDR-SDR bench test, except the communication software will be running on the OBC. This test will be used to validate the communication software's compatibility with the OBC. The bit error rate and data rate will be measured to make sure the software runs to spec on the OBC.

6.3. Payload Board

6.3.1. Verification

The verification plan for the payload board is to flash the MCU with KubOS and send/receive data through the I/O connections. This will be done in the following steps:

a. Visually Inspect Payload Board Before assembling the board, it will be visually inspected to find any shorts or missing connections. To aid in this process, a digital voltmeter will be used to check for any shorts. Additionally, we will check for any missing solder pads that may have fallen off during manufacturing and make sure the orientation of the solder pads matches the correct orientation of the I/O devices so that nothing will be blocking I/O connections.

b. Solder Remaining Components Once the board has been visually inspected and no problems are found, the remaining components that were not assembled by the manufacturer will be soldered onto the board by hand. The steps for successfully soldering a PCB are detailed below:

1. Melt a small amount of solder onto one pad of a component.
2. Grab the component and place it in its correct orientation.
3. Reflow the previous solder so the solder grabs onto the component.
4. Let the solder cool. The part should now be locked into place.
5. Solder the rest of the component as it is being held into place.

Using flux makes this process much easier and also ensures a clean solder joint, so we will use plenty of it during the assembly process. To make sure that everything is assembled properly, solder joints will be inspected with a microscope and a digital voltmeter will be used to check for connectivity of the components.

c. Flash Board with KubOS Several steps must be taken to flash the Beaglebone Black with KubOS. Our board uses the same MCU as the Beaglebone Black, so we will perform the following steps:

- Installing the necessary eMMC with the KubOS Linux flashed SD Card.
- Overwriting the eMMC, with U-Boot in order to boot Kubos Linux from the microSD card.
- Connecting to Beaglebone black using FTDI cable to establish a serial connection.
- Flashing the microSD card with the auxiliary SD card image, putting the card back into the microSD slot, and booting up the system. This image contains the Kubos Linux upgrade partition and the second user data partition.

A successful boot up of KubOS will confirm that it was flashed correctly. The way we will find out if KubOS booted successfully is by powering up the board and accessing the terminal through a wired or secure shell connection.

d. Use KubOS to Control I/O Programmatically In order to control the board's gpio pin, there are several Linux functions and procedures to read and write from the pins. There are several steps in this process:

- Any pin that is not dedicated to a previously mentioned peripheral is available for use.
- To interact with a pin from the command line or from a script, the user will first need to generate the pin's device name
- User's application with KubOS Linux's interface is composed to set the pin's direction, value, read the value back, and release the pin.

```
#include <sys/stat.h>
#include <sys/types.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int fd;
int pin = 45;
int value = 1;

/* Define the pin to the system */
fd = open("/sys/class/gpio/export", O_WRONLY);
write(fd, &pin, sizeof(pin));
close(fd);

/* Set the pin's direction */
fd = open("/sys/class/gpio/gpio45/direction", O_WRONLY);
write(fd, "out", 3);
close(fd);

/* Set the pin's value */
fd = open("/sys/class/gpio/gpio45/value", O_WRONLY);
write(fd, &value, 1);
close(fd);

/* Read the value back */
fd = open("/sys/class/gpio/gpio45/value", O_RDONLY);
char strValue[3];
read(fd, strValue, 3);
value = atoi(strValue);
close(fd);

/* Release the pin */
fd = open("/sys/class/gpio/unexport", O_WRONLY);
write(fd, &pin, sizeof(pin));
close(fd);
```

Figure 67. Example of Control I/O Programmatically

To confirm that I/O is being controlled successfully, we will use an oscilloscope to measure the digital signals of the I/O pins and compare the measurements to what we expected from our code.

6.3.2. Validation

The payload board design becomes valid once KubOS is successfully flashed onto the non-volatile storage, the board boots, and the MCU is able to communicate with I/O devices, such as the IMU, programmatically. This will mean that our hardware is capable of running a Linux kernel, signal integrity is preserved, and the board has met its functional requirements because the hardware is able to perform all of the necessary tasks for system operation.

6.4. Flight Software

6.4.1. Verification

The first test is going to verify the functionality of the interface between the GNU Radio (SDR) and Communications Service, as well as, between our designed mission application and the science sensor (IMU). The test's plan includes sending a simple packetized request for IMU telemetry from our ThinSat. Inevitably, this will require the microcontroller board running KubOS Linux, the IMU, and a Laptop simulating a UDP port. The latter will be the communications protocol for most of our system and subsystems. The verification will be successful if the data is successfully passed to the Laptop in the right schema, defined below, and fits within the UDP connections.

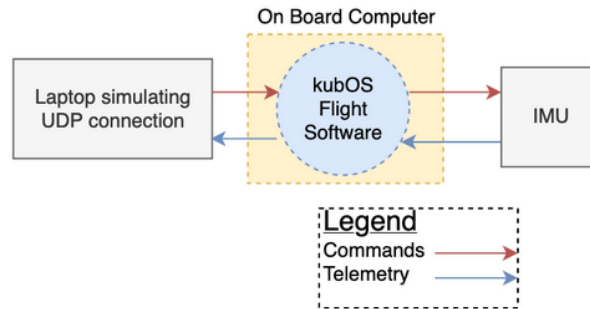


Figure 68. Testing Interface between instrument and GNU Radio

The second test is going to verify the ability to assess the overall health during flight using absolutely minimum telemetry. This test will downlink a subset of critical telemetry data via the mission application and communications service. Once again, the list of required testing materials includes a microcontroller unit running KubOS Linux, a Laptop simulating a UDP connection, and an IMU. Keep in mind both tests will be successful if the correct response is retrieved in the right schema and in CCSDS format.

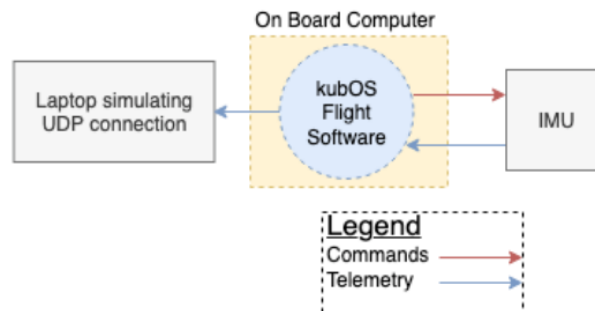


Figure 69. Health and Status / Beacon Test

Depending on what type of system interaction, the commands will take different syntax and data structure. In order to make any changes to a system's state, the software will rely on mutations. On the other hand, if the ground system wants to access the generated data, it will have to submit a query. There are a limited amount of data parameters saved during flight and the laptop/ground system has chronological access each subsystem's data any time point during flight.

Take the following example:

```

{
  "data": {
    "telemetry": [
      {
        "timestamp": 730.855591753,
        "subsystem": "eps",
        "parameter": "current",
        "value": "0.1"
      },
      {
        "timestamp": 730.21144042,
        "subsystem": "eps",
        "parameter": "current",
        "value": "0.1"
      }
    ]
  }
}

```

Figure 70. Query/Mutation Schema for Testing

6.4.2. Validation

After the tests provided in the verification section, the design mission application along with KubOS flight software can be validated by:

- Upon request have the IMU downlink telemetry in CCSDS format .
- Receive live health-status data in CCSDS package without directly requesting it.

Moreover, the following test will attempt to validate the following functional requirements:

FR4 The On Board Computer (OBC), SDR Software, and Flight Software (FSW) shall perform all necessary software tasks for the balloon flight.

It will also check that the payload is commandable and can deliver generated telemetry. Furthermore, if an error message is displayed, missing correct schema, the flight will recognize the error in command formatting.

6.5. Ground Software

6.5.1. Verification

The verification plan for COSMOS ground software can be broken down into four preliminary tests.

a. COSMOS Command Sending Test The first of these four tests to verify the functionality of the ground software was proving that a command can be sent. The steps of the test went as follows:

1. Created a target system to interface with in COSMOS
2. Configured the system to take example temperature measurements
3. Used the Command Sender tool to send a command to the system to start a collection of telemetry
4. Received a confirmation that command had been sent

The results of this test were positive, and can be shown in Figure 71 below.

Target: Command:

Description: Starts a collect on the INST target

Parameters:

Name	Value or State	Units	Description
TYPE:	NORMAL	0	Collect type
DURATION:		3.0	Collect duration
OPCODE:		0xAB	Collect opcode
TEMP:		25.0	C Collect temperature

Command History: (Pressing Enter on the line re-executes the command)

```
cmd("INST COLLECT with TYPE NORMAL, DURATION 3.0, OPCODE 171, TEMP 25.0")
```

cmd("INST COLLECT with TYPE NORMAL, DURATION 3.0, OPCODE 171, TEMP 25.0") sent. (2)

Figure 71. Results of the Command Sending Test in COSMOS

As can be seen in Figure 71, an example COLLECT command was sent to the target system, named "INST". Then the confirmation appears at the bottom of the screen with "sent" at the end.

b. COSMOS Telemetry Visualization Test The purpose of the visualization test was to test the ability of the Telemetry Grapher tool to accurately plot the data to make sure proper data is being received during the process. The steps of this test were as follows:

1. Created a target embedded system in COSMOS to communicate with
2. Began taking real-time temperature health and X,Y,Z Position data through a UDP port into the Command & Telemetry Server (The data collected is taken from a database that COSMOS contains for one of its example target systems)
3. Used the Telemetry Grapher tool to plot the incoming telemetry
4. Confirmed that plots are accurate to the data being received

The results of this test were successful in that it was shown that telemetry data can be accurately received and plotted. Multiple plots can be shown at the same time, and there is a proven method that allows for visualization of real-time telemetry data. This will be applicable to the TOMCAT mission in that we will be able to view the incoming health status, IMU, and housekeeping data as it is received on the ground. A screenshot of the Telemetry Grapher is shown below in Figure 72.

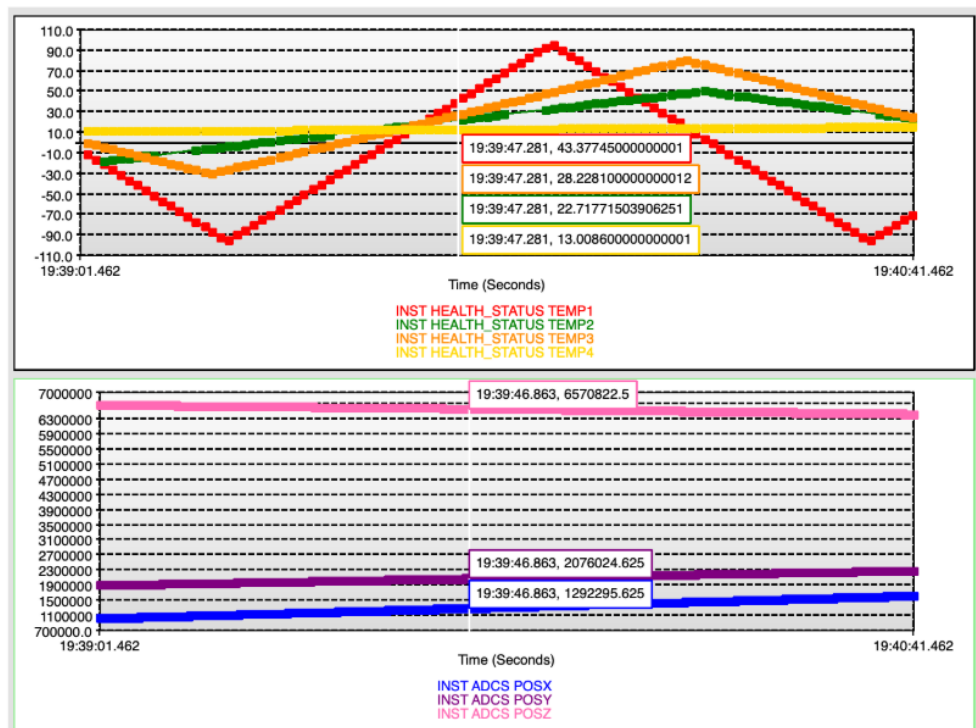


Figure 72. Results of the Telemetry Visualization Test in COSMOS

As can be seen in Figure 72, telemetry data can be plotted as it is received live from a target. The data plotted in Figure 72 is example temperature data with units in °C (on top) and example IMU data with units in ft. (on bottom).

c. COSMOS Telemetry Logging Test The third verification test of COSMOS is to prove that that telemetry data being received is also being stored in a directory. The test was carried out as follows:

1. Created a target system to interface and began collecting live telemetry
2. Manually started and stopped logging in the Command & Telemetry Server
3. Extracted the telemetry data into a text file using the Telemetry Extractor tool
4. Confirmed that data was located in text format in the file directory

The results of the telemetry logging test were successful. The telemetry was able to be extracted into a text file and stored in a file directory on the host computer. A screenshot of the text file containing temperature health status data in °C from the system is shown below in Figure 73.

TARGET	PACKET	TEMP1
INST	HEALTH_STATUS	-45.615449999999996
INST	HEALTH_STATUS	-40.6714
INST	HEALTH_STATUS	-35.72735
INST	HEALTH_STATUS	-30.783299999999997
INST	HEALTH_STATUS	-25.839249999999993
INST	HEALTH_STATUS	-20.895199999999999
INST	HEALTH_STATUS	-15.951149999999998
INST	HEALTH_STATUS	-11.007099999999994
INST	HEALTH_STATUS	-6.063049999999999
INST	HEALTH_STATUS	-1.118999999999998

Figure 73. Results of the Telemetry Logging Test

d. COSMOS Integration Test This fourth test will be to show that COSMOS can be integrated in with the communications software, and the data packet types being dealt with. This test will involve writing the configuration files to narrow down on specific format of commands and telemetry, and testing it by creating a folder, then sending commands to it, and receiving fake telemetry over a virtual UDP connection. Because the exact CCSDS header format has not been decided upon, this integration test will be conducted during the Spring semester.

6.5.2. Validation

As the verification tests have proven, COSMOS ground software can be validated to be able to:

- Command the to start collecting data from the IMU
- Receive CCSDS packetized telemetry data from the with real-time graphing and logging

These validated capabilities are able to satisfy the relevant functional requirement 2:

FR2: The payload shall communicate with the ground system during the balloon flight.

The ground software is a necessary aspect of the communication process as was proven above.

6.6. Thermal

Cold Test #1, as presented in Section 5.6.2, will serve as the basis for a thermal model that will be built upon in the spring semester. This test allowed the team to prove that insulation would be sufficient to keep the temperatures within the ThinSat within the required ranges. Future cold tests in the spring semester will build model fidelity, explore insulation thickness and type, reduce model errors, and study altitude effects. Two more cold tests are scheduled in the spring that will serve as verification and validation for the thermal system. Cold Test #2 will occur in late January or early February and Cold Test #3 will occur in March.

6.6.1. Verification

a. Cold Test #2 Cold Test #2 will serve as a verification test for the thermal protection system. The objectives of this test are as follows:

- Reduce errors in the test setup and improve on cold test design from fall semester.
- Explore insulation types and choose an optimal insulation type and thickness.
- Study altitude effects.

This test will allow us to determine the ideal amount and type of insulation to cover the ThinSat with and will verify the validity of using insulation as the sole thermal system on the ThinSat. The second cold test will be used to increase fidelity of the thermal model and meet the temperature requirements with a sufficient margin to account for uncertainties in testing.

The setup of the second cold test will be very similar to that of the cold test presented in Section 6.6.2. The test specimen will be more accurately representative of the actual TOMCAT design, and though hardware such as the payload board and SDR will not be used, dummy components will be used as placeholders to take up the same amount of space the actual components would take. The same goes for the battery pack. Once again, power resistors will be used to simulate the heat generated by the electronics. These resistors will be placed where each component will be housed and will output the same wattage that expected of each electronic component. Thermocouples will be placed on each side of each power resistor as well as directly on the power resistor to measure the temperature the component itself may rise to.

A control test will be run to measure the temperatures of the ThinSat without insulation. Subsequent tests will be run in which the type and thickness of insulation will be varied. Extruded polystyrene will be tested once again to verify the results of the test presented in Section 6.6.2 and more layers of this material will be added to study the effects of insulation thickness.

The results of this test will inform our final thermal design and the insulation chosen will be used on the balloon-flight design of the ThinSat.

6.6.2. Validation

Both of the validation tests outlined in this section will allow the satisfaction of functional requirement 3 and design requirement 3.1:

DR3.1: The payload shall be able to withstand the temperature range of flight, which includes the coldest point in the flight (-56°C), and a hot day on the ground (38°C).

FR3: The payload shall survive and operate in the environment during all stages of the balloon flight

a. Full System Functional Cold Test (Cold Test #3) Cold Test #3 will be a validation test. The goal is to validate system performance on the ground without the need for a balloon flight.

This test will require the fully functional TOMCAT design with flight hardware (functional payload board and SDR with all supporting electronics), running on battery power. Data from this test will be used to validate the thermal system and prove validity of the insulation system.

The test setup will be similar to that of Cold Test #2, although the heat will now be generated from the actual electronics instead of power resistors. The test specimen will replicate the full TOMCAT design with the insulation chosen from Cold Test #2. Thermocouples will be placed near each electronic component, on each component, and near/on the battery to get a full temperature profile of the entire ThinSat. The test specimen will be placed in a cooler with dry ice and the ambient temperature will be allowed to fall to the minimum flight temperature of -56°C . During this test, the battery voltage will also be monitored to ensure that the battery capacity stays above the required 75%.

The test will be run for the full expected length of the balloon flight, about 2 hours. During this time, flight operations such as the sending and receiving of commands and telemetry will be performed to ensure that the system is working correctly under the expected environmental conditions.

The test will be deemed successful if the temperatures of the electronic components and the battery stay within their required ranges and the battery capacity does not drop below 75% and all flight operations can be performed. This means that all components perform optimally and the battery provides the correct amount of power to each component.

b. Balloon Flight Validation The balloon flight allows for the validation of the thermal system in the actual flight conditions at 30 km. Temperature readings during flight will be used to determine the maximum and minimum temperatures during flight. If these temperatures stay within the required ranges for optimal performance, then the thermal models from verification testing can be deemed successfully validated.

6.7. Power

6.7.1. Verification

In order to verify that the power system works correctly and is safe for the payload components and battery to be operated, a charge/discharge test, peak power test, and full system thermal test will be performed.

a. Charge/Discharge Test The objective of the charge/discharge test is to ensure that the battery performs as expected and is able to be charged and discharged safely. In order to perform this test, the battery will be connected to the charger and voltmeter. The battery's voltage shall be measured in short time intervals until fully charged. Once fully charged, the battery shall be connected to a resistor and voltmeter. The battery's voltage shall be measured in short time intervals until discharged to 5V. The materials required for this test are: the battery pack, voltmeter, and a resistor. In order to pass the charge/discharge test, the charger must stop charging the battery when the maximum voltage of 8.4V is reached. The battery shuts off when the minimum voltage of 5.0V is reached.

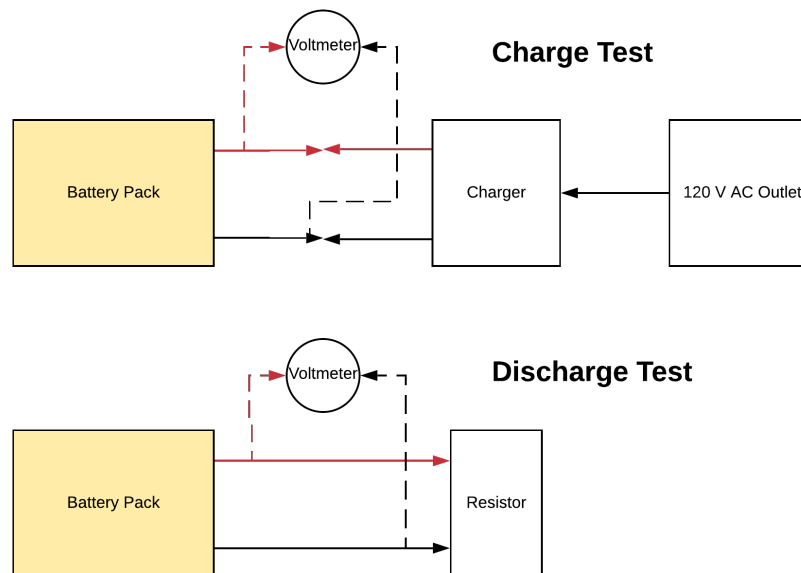


Figure 74. Charge/Discharge Test

b. Peak Power Test The objective of the peak power test is to ensure the battery is capable of discharging enough current during peak demand, such as when the components are all powering on or off, and when the system is simultaneously transmitting and receiving commands and processing the data. In order to perform

this test, the current drawn during peak power draw shall be measured. The current shall be measured when the system is booting up and powering down, and when transmitting and receiving simultaneously while also processing data. The materials required for this test are a power supply, voltmeter and ammeter, and the operational payload. In order to pass the peak power test, the current drawn at peak power draw times must be less than 3A, the maximum discharge rate of the battery. Figure (75) shows a simplified diagram of the peak power test.

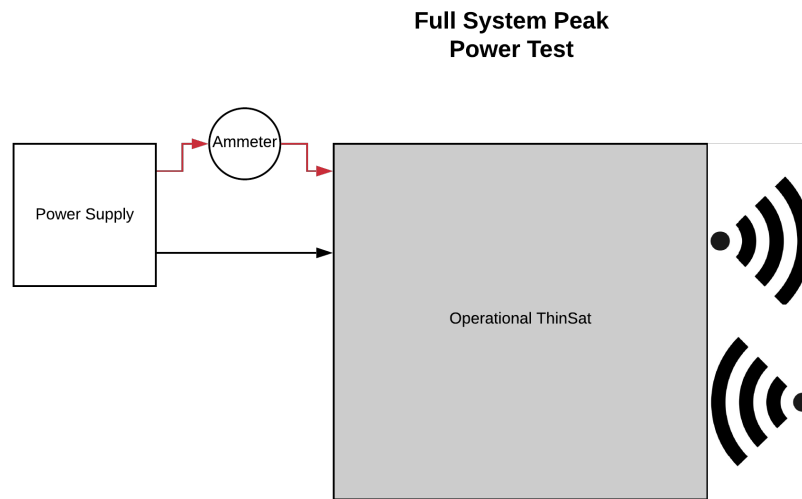


Figure 75. Full System Test

c. Full System Thermal Test (Cold Test #3) The objective of the full system thermal test (Cold Test #3) with respect to the power system is to ensure that the battery is capable of powering the full system at the temperatures expected during balloon flight. In order to perform this test, the system shall be placed in a thermal chamber under balloon flight temperature conditions for 2 hours. The battery voltage and discharge rate shall be measured periodically throughout the duration of the test. The materials required for this test are a voltmeter and ammeter, the battery pack, a thermal chamber, and the operational payload. The objective of this test with respect to the power system is to ensure that the battery powers the system for the duration of the balloon flight and provides the required voltage and current at all times.

6.7.2. Validation

The power system shall be validated by meeting the following requirements:

FR 3.1 The payload shall be able to withstand the temperature range of flight, which includes the coldest point in the flight (-56°C), and a hot day on the ground (38°C).

By passing the full system thermal test, the power system will meet FR 3.1. This allows the team to validate the ThinSat without the balloon flight. The balloon flight allows for further validation of the power system in the actual flight conditions at 30 km. State of charge measurements during flight will be used to determine the performance of the battery during the mission so that the power budget and requirements can be deemed successfully validated.

6.8. Structures

Matthew McCallum

6.8.1. Verification

The requirement that needs verification is Functional Requirement 3.2: The Payload shall withstand the impact from a balloon launch (estimated 9 m/s from vendor). This requirement will be verified by performing a drop test.

For the drop test, a model ThinSat will be constructed with mass simulates so that the system will function identically to the real ThinSat without putting any of the components in danger. The ThinSat model will be dropped from 13.5 feet (4.13 meters) onto multiple surfaces (dirt, grass, maybe snow, and concrete). This height causes the model ThinSat to impact the ground at 9 meters per second, the same as the expected impact speed. After each drop the model ThinSat will be inspected to determine if there was any damage to the frame or if a component broke loose. A ruler will be placed at the landing zone and the impact will be filmed at high speed to enable some calculations about the test. Slow motion video from our phone cameras will allow us to verify the impact speed of the tests by comparing the motion of the ThinSat to the ruler. Also, a high speed camera will be borrowed to film the impact at a much higher frame rate. This will allow for the time of impact to be accurately determined. With an accurate time of impact, a better prediction of the impact forces could be calculated which would allow the model to predict the impact more accurately. This would give us a better model to compare the impact to as well as possibly provide a way to figure out what went wrong if there is a problem with the test.

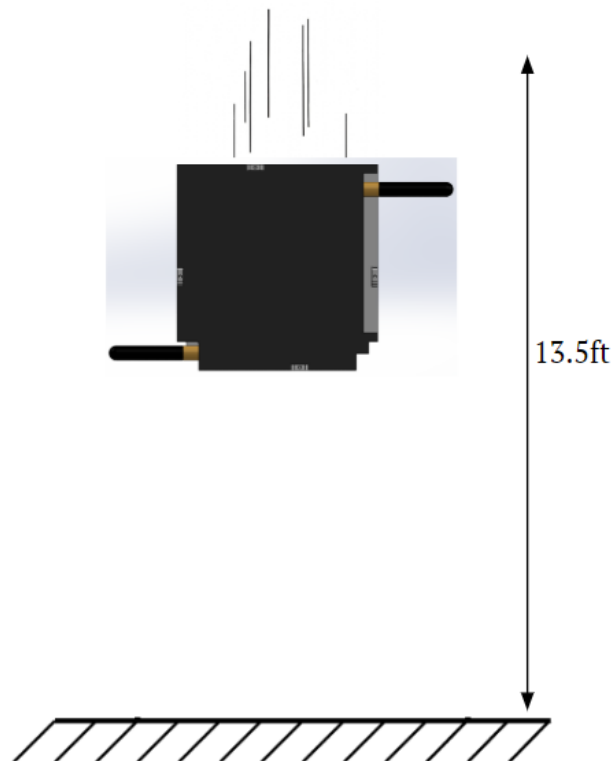


Figure 76. Drop test will be conducted in this orientation and from 13.5 feet

6.8.2. Validation

Validation of FR3.2 will be done by performing a visual inspection and full system test of the ThinSat after the balloon flight. The visual inspection will determine if any components broke loose during impact, if the structure was damaged, or if any components were significantly damaged. The full system test will validate that there was no damage, that could not be confirmed visually, that interferes with the function

of the system. If the system did not receive any significant damage and is still functional after the balloon flight, then FR3.2 will have been validated.

6.9. Science

6.9.1. Verification

The applicable requirements for the IMU include the following:

- FR3.4: The payload shall survive and operate in the environment during all stages of balloon flight
- FR3.4.1: Science data shall be able to be collected and updated at least once every second
- FR4.1: The payload shall be commandable
- FR4.2: The payload shall generate telemetry

These requirements can be demonstrated to be met through a variety of methods. FR3.4 will be proven using full-system cold tests (detailed in Section 6.6), where the IMU will be present in a simulated stratosphere. The team will test to see if the IMU can output data under the thermal flight conditions, where temperatures reach -56°C . If the IMU keeps sending correct data, then the team will confirm that it meets requirements. Correct data, in this case, will be in the form of appropriate measurements of rotation and change in heading - the team will periodically move the cold test chamber, causing a rotation to occur. By confirming that rotational rates are sensible and angles from true north are correct, the team will confirm FR3.4.

FR3.4.1 is satisfied by the Xsens' data output rate of 100 Hz, meaning it can take data 100x more frequent than the requirement of 1 Hz. The team will test for this during the first ranged system test, where IMU data will be collected and transmitted. If more than 1 data point reaches the "ground station" per second, FR3.4.1 will be satisfied. Furthermore, the team can verify before the IMU is integrated into the TOMCAT system by configuring it with an Arduino and testing to see the data rate it outputs at through an Arduino UI.

FR4.1 specifies that the payload must be commandable. The Xsens IMU is configurable with five different coding languages, and the mission could potentially be specified to do many other tasks other than determine ascent versus descent. In the case of this mission, commands will be uploaded to MCU and passed on to the IMU which specify for it to pass on its most recent accelerational, rotational, and magnetic data, and to determine whether it deems the current phase to be ascent or descent. The team can verify that this requirement is met during full system testing or simply on its own, communicating with an Arduino.

Finally, FR4.2 will be verified during full system long-range test involving commands and telemetry. The IMU will be tasked with generating telemetry for the on-board communications system to transmit to the makeshift "ground station." Generating telemetry, in this case, simply means asking the IMU to take data as it normally would in a powered configuration. Verifying this requirement should be exceedingly simple and provides little resistance in overall system verification.

6.9.2. Validation

The customer has requested the fulfillment of two primary objectives in regards to the science mission:

- Telemetry needs to be generated
- Data should be applicable in the atmosphere and in the vacuum of space

As discussed in science verification, the IMU can generate telemetry at rates that far exceed the minimum needed to successfully carry out the TOMCAT mission. As long as the IMU has power, maintains a temperature within its operational envelope, and is properly mounted and interfaced with the printed circuit board, there is no realistic way the IMU could not carry out this functionality, barring a completely dysfunctional unit. The team is confident that their science sensor is to the customer's liking.

The second customer request for a science mission entailed integrating a sensor that could eventually complete a mission in low Earth orbit as well as one for balloon flight. The team kept this request in the forefront of their minds when selecting a science mission, and adopted the IMU partially on the account that it was not hindered by the vacuum of space and relatively resistant to its harsh and dynamic range of temperatures. Acceleration and rotation data are completely unaffected by the presence, or lack thereof, of atmosphere. The Earth's magnetic field is less strong in the space environment, but the magnetometer will still take readings and provide meaningful data. In Raytheon's end-goal mission, taking place in low Earth orbit, the mission can be adapted to conduct state estimation or characterization of Earth's magnetic fields from space.

The team is confident that the MTi-3 AHRS will provide the capabilities necessary to satisfy the demands of their customer and complete a variety of missions, regardless of whether the ThinSat is launched via balloon or rocket.

7. Risk Assessment and Mitigation

Cole

7.1. Critical Technical Project Risks

The following risks outlined in Fig. (77) and Fig. (78) pertain to the critical project elements and would result in failure of the system or significantly reduced performance if an anomaly were to occur. Figure (77) shows the risks post-mitigation, which will be explained below.

Technical Risks					
Subsystem	Shorthand	Description	Probability	Impact	Color Code
Comm	Balloon Drift	Balloon drift outside of ground antenna FOV	Low	Medium	
	T1		1	2	
Comm	RF Noise	Noise and losses overcomes link margin	Medium	Low	
	T2		2	1	
Thermal	Thermal	Low thermal modeling margin results in low confidence of survivability at full system operation	Medium	Medium	
	T3		2	2	
Payload Board	Signal Integrity	Signal integrity is not preserved	Low	Medium	
	T4		1	2	
Payload Board	PCB Failure	PCB manufacturing is incorrect and does not work	Medium	Low	
	T5		2	1	
Thermal	Heat	Heat from the SDR or payload board is localized and builds up in components, leading to overtemps and component failure	Medium	High	
	T12		2	3	

Figure 77. Critical project risk table

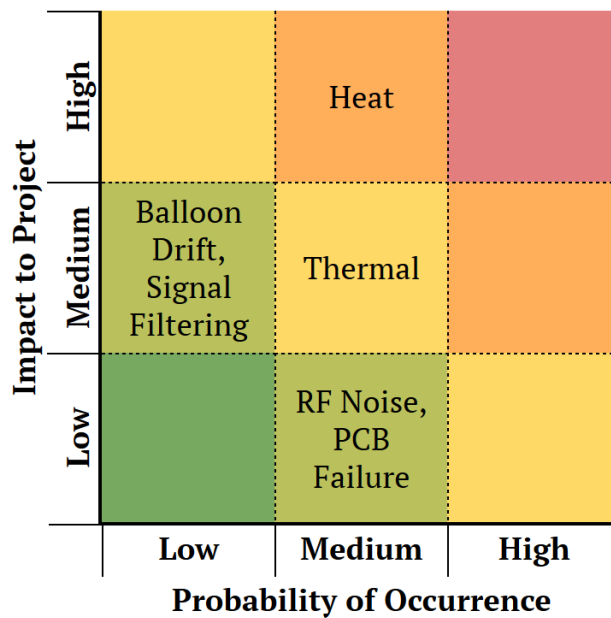


Figure 78. Critical project risk matrix

7.2. Critical Risk Descriptions and Mitigations

Prior to mitigation, the critical risks were all in the upper right quadrant of the risk matrix, as shown in Fig. (79).

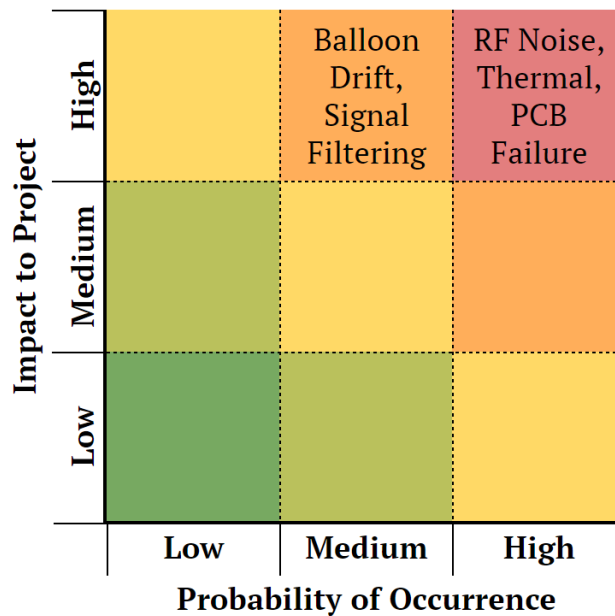


Figure 79. Critical project risk matrix prior to mitigation

7.2.1. Balloon Drift

The balloon drift risk, T1, is primarily concerned with the balloon drifting outside of the field of view of the ground based antenna, causing a loss of communication and the ability to send and receive data. The

ground antenna is fixed and does not have an active pointing system, so after deployment, the field of view is effectively fixed. While the antenna could be manually pointed, since the balloon ascends to over 30 km, it is extremely unlikely that the team would be able to visually locate the balloon to point the antenna.

To mitigate the risk of the balloon drifting outside of the antenna field of view, a ground antenna with a sufficiently large field of view was selected. Specifically, TOMCAT's ground system antenna has a 70° vertical beamwidth and a 100° horizontal beamwidth. Additionally, the primary ground system antenna (and by extension, the ground station itself) is displaced from the launch site by 10 km. This is further discussed in Sec. 5.1.3. The larger beamwidth is situated across the ground (horizontally) since this is the direction the balloon would drift in due to wind. Since effects of the wind are far more uncertain than the ascent of the balloon itself, the larger beamwidth provides more margin.

7.2.2. RF Noise

The RF noise risk, T2, is concerned with noise from other radio equipment on the balloon flight or environmental factors interfering with the communications and impacting the ability to send and receive data. Possible sources of noise include noise from the sun, noise from reflections on clouds or other obstructions, and other radio equipment used to support the balloon flight.

To mitigate this risk, TOMCAT has designed for a very large link margin compared to a typical balloon satellite. This large link margin would be able to overpower any sources of noise and maintain a reliable communications link between the satellite and the ground.

7.2.3. Thermal

The thermal risk, T3, covers the thermal failure of the system due to the satellite not being able to maintain the minimum required temperature for nominal operation. This temperature is defined by the battery, which should remain above -10°C in order to retain enough capacity to power the TOMCAT satellite over the full 2 hour balloon flight. While the temperature dropping below this value would not result in complete system failure, as the satellite would still be able to operate while power remained, it would result in the system not maintaining continuous communication throughout the duration of the flight. While temperature is a concern for other components in the satellite, all other components are rated for operation down to -40°C, so the battery temperature supersedes these components.

The mitigation for this thermal risk is discussed in detail in Sec. (5.6). In short, insulation is being used in order to retain heat from components inside the ThinSat frame itself. The margin on this insulation is currently very small, which is why this risk is still in the "yellow" category post-mitigation. Further testing will be conducted to increase the fidelity of the thermal model and get a more accurate representation of the margin.

7.2.4. Signal Integrity

The signal integrity risk, T4, describes the possibility of noise and loss on signals within the TOMCAT system, such as along wires, through connectors, along traces on the Payload Board, and at solder joints between board components and PCBs. Loss of signal integrity could result in faulty commands being read or faulty telemetry being sent. If these errors are significant, the data may be unreadable and the system would be unable to continue operation.

This risk is being mitigated by using design best practices and online tools, like those provided by Texas Instruments, to analyze and simulate board designs for noise and losses. This will allow the team to understand expectations of noise and signal integrity losses prior to purchasing the board and components.

7.2.5. PCB Failure

The PCB failure risk, T5, addressed the team's overall lack of experience in designing and manufacturing a custom PCB. While the team does have an electrical engineering student with experience in PCB design, few other team members have any experience at all. Additionally, the board is being designed to use hand soldered components where possible, and many of these are surface mount components. Surface mount components are more difficult to solder than through-hole components, so this also presents a risk. Given that all of the components of the TOMCAT satellite are connected to the payload board PCB, failure of this part of the system would result in immediate and unrecoverable mission failure in flight.

The PCB failure risk is being mitigated in multiple ways. From a technical standpoint, our electrical engineering student will be working to teach other members of the team how to design PCBs so that more individuals are able to troubleshoot problems. Logistically, the team is allocating schedule for 4 total board revisions, which is further detailed in Sec 8.3.1. Additionally, the team has received a sponsorship with Advanced Circuits who will be able to manufacture and assemble the boards for almost no cost. These schedule and budget margins allow the team to, in a worst case situation, throw additional time and money at the problem in order to resolve any issues.

7.2.6. Heat

The heat risk, T12, covers the thermal failure of the system due to temperatures getting too hot. The main concern for this arised after CDR and could result in component failures due to going over the manufacturer recommended temperature limits. This is possible due to the lack of convection at higher altitudes, meaning heat generated by components remains localized and near the components themselves.

Currently, no detailed mitigation plan exists for this risk, as it was only recently identified. However, heat can be moved around by careful design of the PCB, so the team will be working with Dr. Rainville to ensure the PCB design can effectively transfer heat as needed.

7.3. Critical Logistical Project Risks

Currently, the only logistical risk the team has identified is the possibility of people not completing their work by assigned deadlines. This is currently allocated as a High probability of occurrence, and Medium impact to the project. This is mostly because this has occurred previously, and other team members then have to pick up the slack to complete required deliverables. While this hasn't resulted in failures yet, as the schedule tightens in the Spring and team members get more loaded up on class and external projects, the impact of not doing assigned work could become more severe.

The mitigation for this is setting internal deadlines with enough schedule margin to account for lack of work. Throughout the Fall semester this was reasonably effective, so we anticipate this mitigation is sufficient.

Currently, there are no significant financial risks to the TOMCAT team. Since the team is receiving additional funding from Raytheon to cover the cost of the ThinSat Education Kit and we have applied for an EEF grant, financial aspects of the project are not a concern. No part of the project would need to be repurchased due to catastrophic failure unless the balloon flight failed. The current budget allows the satellite to be re-manufactured up to three times with margin.

7.4. Risk Tracking Methodology

The team is tracking risk continuously and uses a custom spreadsheet to track critical risks throughout the lifetime of the project. This risk tracking tool is updated weekly to include new high priority risks, shift current risks, and remove low priority risks. Currently, the team is tracking 12 total risks to the project which can be seen in Fig. (80).

Technical Risks					
Subsystem	Shorthand	Description	Probability	Impact	Color Code
Comm	Balloon Drift	Balloon drift outside of ground antenna FOV	Low	Medium	
	T1		1	2	
Comm	RF Noise	Noise and losses overcomes link margin	Medium	Low	
	T2		2	1	
Thermal	Thermal	Low thermal modeling margin results in low confidence of survivability at full system operation	Medium	Medium	
	T3		2	2	
Payload Board	Signal Integrity	Signal integrity is not preserved	Low	Medium	
	T4		1	2	
Payload Board	PCB Failure	PCB manufacturing is incorrect and does not work	Medium	Low	
	T5		2	1	
Structure	Impact Damage	Components are damaged upon impact after balloon flight	Low	Medium	
	T6		1	2	
Comm		Frame desynchronization	Medium	Low	
	T7		2	1	
FSW		GNU Radio and Kubos Linux are not compatible in the Beaglebone Black	Medium	High	
	T8		2	3	
Structure	Fitting	The components are unable to fit within the ThinSat frame	Low	High	
	T9		1	3	
GSW	Configuration Failure	Tool Configuration is incompatible with CCSDS packet format	Low	Medium	
	T10		1	2	
Power	Size	Batteries are too large to fit with insulation inside ThinSat	Medium	Medium	
	T11		2	2	
Thermal	Heat	Heat from the SDR or payload board is localized and builds up in components, leading to overtemps and component failure	Medium	High	
	T12		2	3	
	T13		FALSE	FALSE	

Figure 80. Risk tracking tool created by TOMCAT

Logistical, safety, and financial risks are tracked using their own versions of the same tool.

8. Project Planning

8.1. Organizational Chart

Cole

The organizational chart shown in Fig. (81) shows the leadership roles of all team members. In addition to the technical leadership roles, all team members also serve as technical members of at least one subteam. For example, Trevor is the Communications Hardware Lead, and both Lauren and Srikanth contribute to the communications hardware team in addition to leading the Safety/Test and Communications Software teams, respectively.

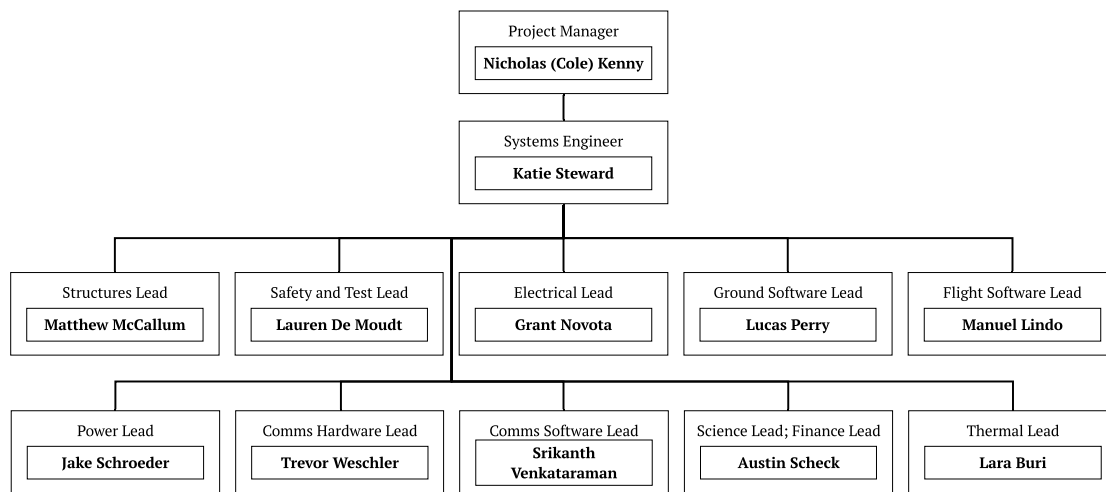


Figure 81. TOMCAT Org Chart

8.2. Work Breakdown Structure

Cole

The WBS shown in Fig. (82) shows an overview of the major work tasks that remain broken down by subteam. Each of these tasks have a number of subtasks which are available in the full Work Plan presented in Section 8.3.

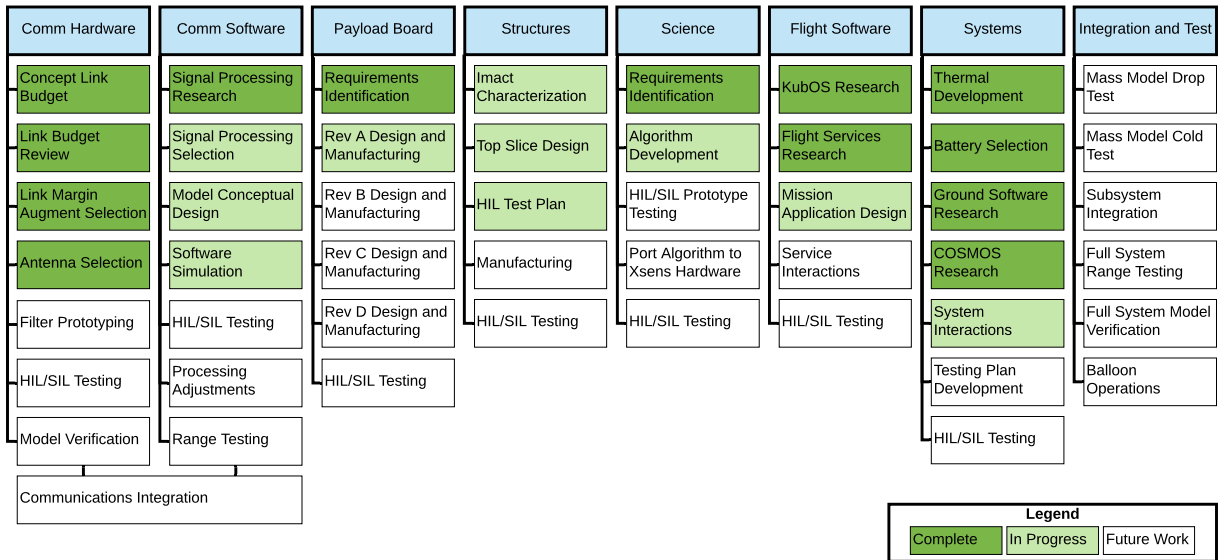


Figure 82. TOMCAT Work Breakdown Structure - Updated 12/14/19

These work tasks were determined from the gantt chart and serve as an easier way to visualize the remaining tasks in the project.

8.3. Work Plan - Gantt Chart

Cole

The gantt chart shown in Fig. (83) through (85) shows a detailed breakdown of all projects tasks in the project and their dependencies. It also outlines the estimated start and end dates for each task which helps inform the overall project schedule and defines the critical path (shown in red). Gantt charts are notoriously difficult to display at regular aspect ratios, so if you'd like access to the Microsoft Project source files, please contact the team directly.

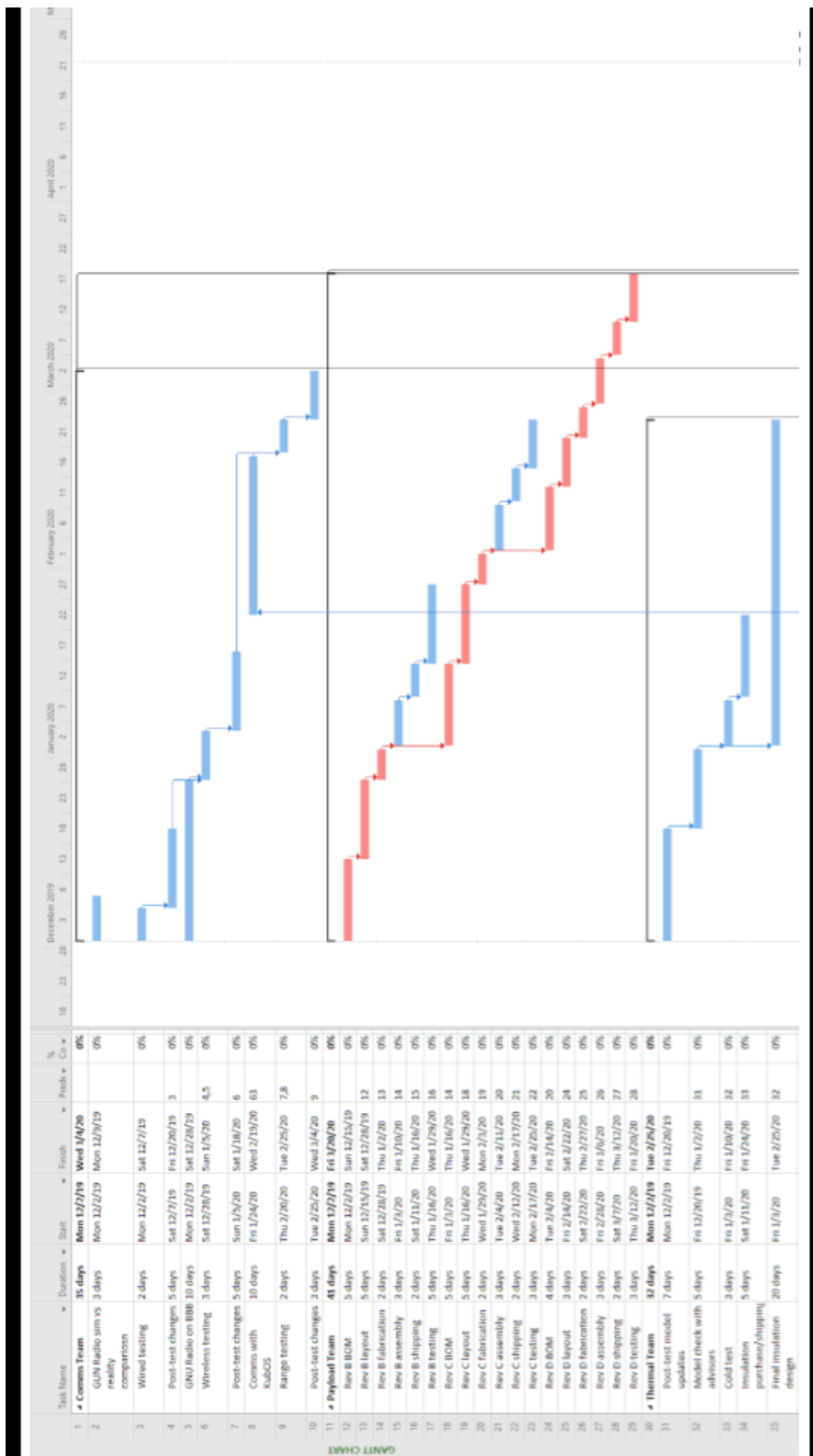


Figure 83. TOMCAT Gantt Chart, Page 1

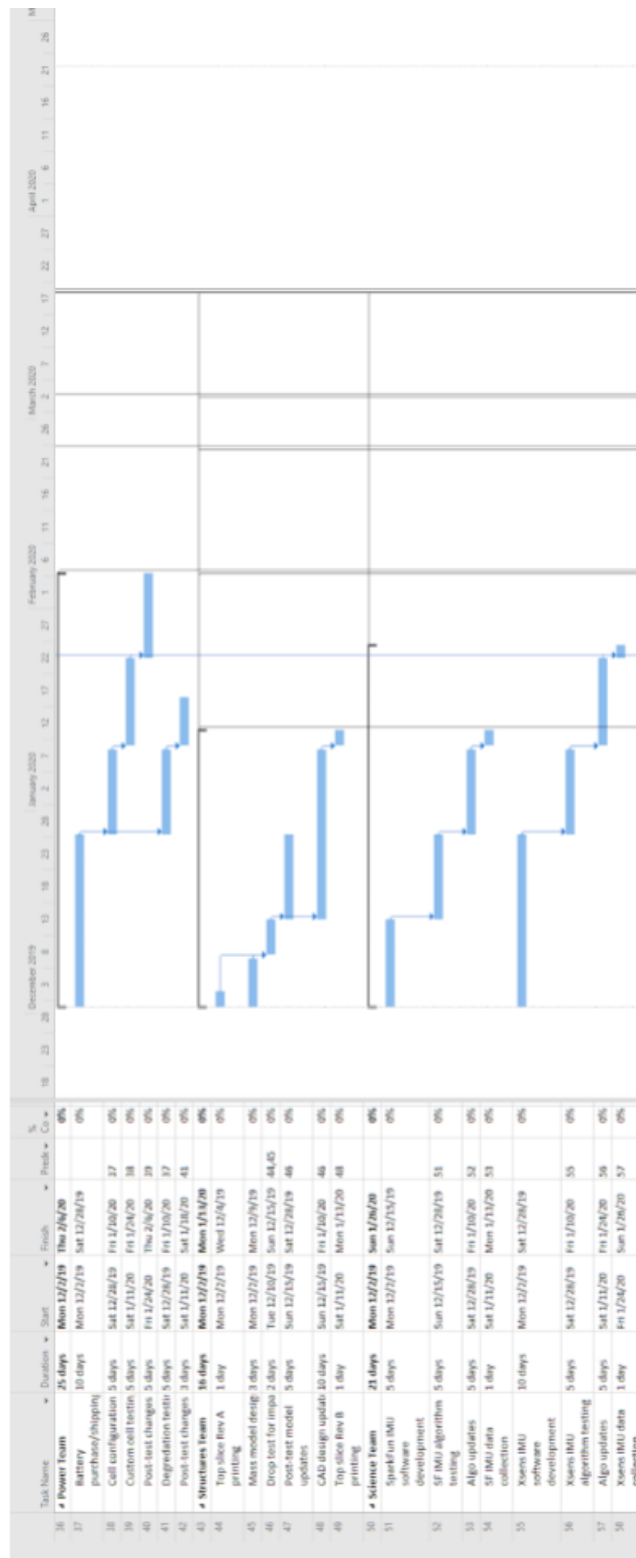


Figure 84. TOMCAT Gantt Chart, Page 2

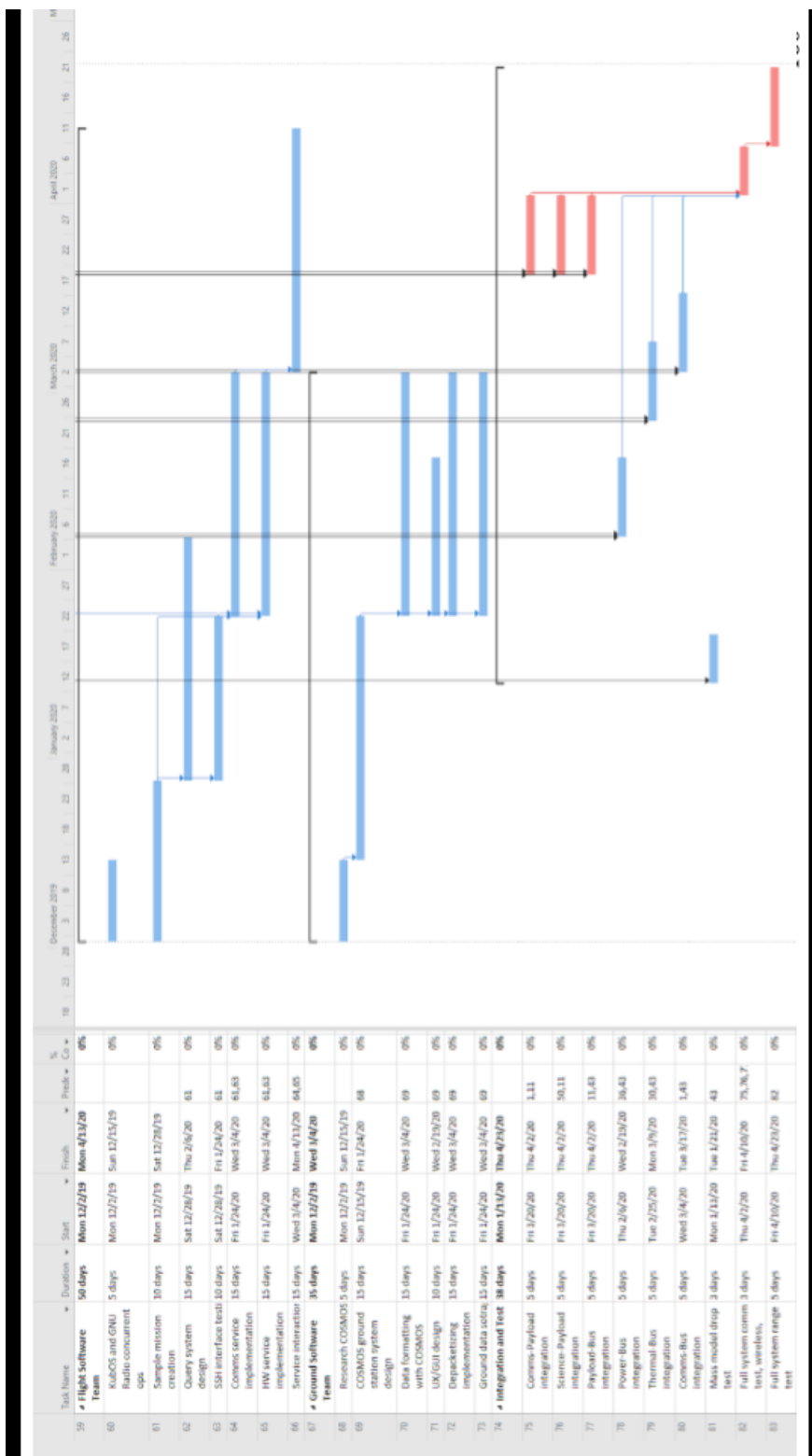


Figure 85. TOMCAT Gantt Chart, Page 3

8.3.1. Critical Path Analysis

The critical path is defined as the continuous string of tasks which collectively have zero slack. By definition, the last task in a project will always be on the critical path. Working backwards, the critical path includes the balloon operations and the highest level system tests, including the final project communications verification, the final cold test. Since these tasks require the use of the TOMCAT satellite, they cannot be completed concurrently.

Flowing back, all tasks in the Payload Board subteam are on the critical path, as this payload board is required to integrate all other parts of the system and is needed in order to to final top level testing.

The team has planned for a total of four board revisions, one in the Fall (complete) and three in the spring semester. Each revision requires the same basic tasks, which make up a full revision cycle. These tasks are:

1. BOM Creation
2. PCB Layout
3. Fabrication
4. Assembly
5. Shipping
6. Testing

While all steps are required to complete each revision cycle, development of the next revision can begin prior to the completion of a current revision. For this project, the TOMCAT team has determined that future revision development can begin after the fabrication step.

Margins in the schedule were automatically calculated based on the task duration and predecessors. By defining the work week to match that of the senior projects schedule and allocating time to defined team members, the Microsoft Project scheduling features produce a reasonable and accurate schedule to work from that doesn't overdraw from any particular teams (assuming the tasks were allocated properly when created).

8.4. Cost Plan

Austin Scheck

The TOMCAT mission is currently on-par with the allocated budget of \$5,000, however additional funding has been applied for from the Engineering Excellence Fund and more has been offered by Raytheon, pending university approval. The tentative cost plan pie chart can be seen below, categorized by project element.

As one can see from the figure, most of the costs will be allocated towards testing. In particular, this includes the Near Space Launch-provided ThinSat (acting as the team's "bottom slice"), a FlatSat, and a slot on one of NSL's balloon launches in Indiana.

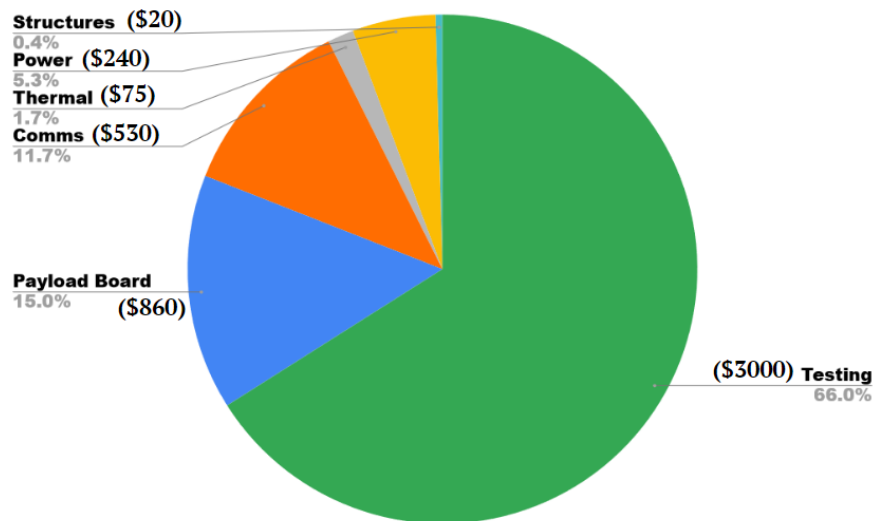


Figure 86. Tentative Cost Plan - Pie Chart

Payload Board: \$860

- MCU (BBB)
- Custom PCB
- MCU Hardware
- IMU

Comms: \$530

- SDR
- SMA Cables
- Antennae
- LNA

Thermal: \$75

- Coolers
- Dry Ice
- Variety of Thermal Insulators

Power: \$240

- Li-Ion Battery
- Charger
- Prototype Fee
- Buck-Boost Converter

Structures: \$20

- PLA Plastic Spool

Testing: \$3000

- ThinSat Bus
- FlatSat
- NSL Balloon Launch

Figure 87. Tentative Cost Plan - Subsystem Overview

With a total cost of \$4,545, this project is nominally lower than the allocated funds. There is a 9.1% margin on the base budget to utilize in the case that additional costs are incurred.

The supplementary funds potentially coming from the EEF (\$2,754.30) and Raytheon (\$3,000) will be helpful in the case that major revisions to the TOMCAT system need to be made at some point next semester. In addition, the team is already paying for a slot on one of NSL's balloon launches, and would like to capitalize on that opportunity. However, because it is located in Indiana, the team would need a

significantly bolstered budget to cover transportation, lodging, and meals. If the Raytheon funding goes through, the team will consider attending the optional launch for full verification of their system.

If that path forward does not pan out, the team has secured flight on a Gateway to Space flight in April of 2020. This will be a free and local launch which will allow them to conduct the same mission as NSL's launch.

The team's full budget in detail to date is shown below. The items highlighted in green are ones that have already been acquired.

FINANCIAL OVERVIEW - Total Purchases											
Component Name	Component ID	Subsystem	Need-By Date	Manufacturer Lead Time	Stock Status	Order-By Date	Qty.	Unit Price (USD)	Total Price (USD)	Notes	Link
Software-Defined Radio	Lime Mini	Comms	11/9/2019	2 Weeks	In Stock	11/6/2019	2	\$159.00	\$318.00	Ordered & Delivered	Link
SMA Cables	ACX1731-ND	Comms			In Stock	11/6/2019	2	16.38	\$32.76	Ordered & Delivered	
Flight Antenna		Comms	TBD			TBD	1	\$49.95	\$49.95		Link
Ground Antenna		Comms	TBD			TBD	1	\$21.95	\$21.95		Link
Inertial Measurement Unit	XSens MTi-3 AHRs	Payload Board	TBD	2 Weeks		TBD	1	\$365.00	\$365.00		
MCU	BBB Wireless	Payload Board	n/a	n/a	n/a	10/7/2019	1	\$76.13	\$76.13	Ordered & Delivered	-
AC/DC Wall Adapter	AC/DC Wall Adapter	Payload Board	n/a	n/a	n/a	10/7/2019	1	\$13.32	\$13.32	Ordered & Delivered	-
PCB Assembly	n/a	Payload Board					1	\$0.00	\$0.00	waiting on quote	4pcb.com
Custom PCB	n/a	Payload Board					2	\$66.00	\$132.00		Link
Microcontroller Unit	OSD3358 <AM3358>(MCU)	Payload Board	Payload Board	10 Weeks (DK)	600+ In Stock (DK)	TBD	1	\$43.73	\$43.73	Digi-Key: OSD3358-512M-BSM RAM: 512 MB	Link
IC Buffer, Non-Inverting (SN74LVC1G07DC KR)	IC Buffer, Non-Inverting (SN74LVC1G07 DCKR)	Payload Board	Soon?	6 Weeks (DK)	11700+ In Stock (DK)	TBD	1	\$0.33	\$0.33	Digi-Key: 296-8486-1-ND	Link
TPD45012	TPD45012	Payload Board	Soon?	6 Weeks (DK)	30000+ In Stock (DK)	TBD	1	\$0.71	\$0.71	Digi-Key: 296-24244-1-N	Link
0472192001 CONN Micro SD Card Hinged Type	0472192001 CONN Micro SD Card Hinged Type	Payload Board	Soon?	12 Weeks (DK)	19000+ In Stock (DK)	TBD	1	\$0.85	\$0.85	Digi-Key: WM6698CT-ND	Link
24 MHz Crystal	24 MHz Crystal	Payload Board	Soon?	9 Weeks (DK)	13000+ In Stock (DK)	TBD	1	\$0.50	\$0.50	Digi-Key: XC1001CT-ND	Link
Micro USB Socket	Micro USB Socket	Payload Board	Soon?	12 Weeks (DK)	55000+ In Stock (DK)	TBD	1	\$0.43	\$0.43	Digi-Key: 609-4613-1-ND	Link
MCU Hardware	MCU Hardware	Payload Board	TBD			TBD	2	\$50.00	\$100.00		
Power Converter	Power Converter	Power	TBD	-	-	-	1	\$40.00	\$40.00	-	-
Battery	Battery	Power	TBD	-	-	-	1	\$200.00	\$200.00	-	-
Passive Insulation	Passive Insulation	Thermal	11/9/19	-	-	11/9/19	1	-	\$15.22	Bought In-Store	-
Cooler	Cooler	Thermal	11/9/19	-	-	11/9/19	2	-	-	Bought In Store	-
Dry Ice	Dry Ice	Thermal	11/9/19	-	-	11/9/19	1	-	\$22.07	Bought In Store	-
Spool of PLA Plastic	Spool of PLA Plastic	Structures	Soon?	In House	In House	In House	1	\$20.00	\$20.00	ITLL has printers we can use. Maybe Aero building?	n/a
ThinSat Bus	ThinSat Bus	Testing	TBD			TBD	1				
FlatSat	FlatSat	Testing	TBD			TBD	1	\$3,000.00	\$3,000.00		n/a
Balloon	Balloon	Testing	TBD			TBD	1				

Figure 88. Tentative Cost Plan - Detailed Overview

9. Individual Report Contributions

Cole Kenny: FFR template generation and administration, Risk and Mitigation, Planning (except cost plan), proofreading

Katie Steward: Project Purpose, Objectives and Requirements, FBD, CONOPS, Requirements Development

Lauren De Moudt: Conceptual Design: Communications Hardware (SDR, Communications Band, Antenna, Half-Duplex Communication), Detailed Design: Communications Hardware (Ground Station Placement), Verification and Validation: Communications Hardware (Transmission Bench Test, Reception Bench Test, Short Range Wireless Test, Long Range Wireless Test)

Austin Scheck: Conceptual Design: Science, Detailed Design: Science, Verification & Validation: Science, Cost Plan

Jake Schroeder: Conceptual Design: Power (Key Design Options Considered, Trade Study Process and Results, Selection of Baseline Design), Detailed Design: Power Budget, Margin and Built in Capacity, Power Distribution, Thermal Considerations, Ground System Power, Verification and Validation: Peak Power Test, Thermal Test

Trevor Weschler: Conceptual Design: Communications Hardware: SDR, Detailed Design: Communications Hardware: (Link Budget, RF chain, Communications PCB).

Grant Novota: Conceptual Design: MCU and Payload Board, Detailed Design: Payload Board, Verification & Validation: Payload Board

Matthew McCallum: Nomenclature, Conceptual Design: Structures, Detailed Design: Structures, Verification & Validation: Structures

Lara Buri: Conceptual Design: Thermal design considerations and baseline design, Science trade studies and baseline design, Detailed Design: Thermal design PDR model and CDR testing, Verification and Validation: Cold Test #2 and #3.

Srikanth Venkataraman: Conceptual Design: Communication Software, Detailed Design: Communication Software, Verification and Validation: Communication Software

References

- [1] “Planet Labs says it’ll acquire Google’s satellite imaging subsidiary, Terra Bella,” , 2017. URL <https://www.geekwire.com/2017/planet-labs-will-acquire-googles-satellite-imaging-subsi-diary-terra-bella/>.
- [2] “Solar wind magnetic field interacts the Earth’s magnetic field,” , 2019. URL https://www.researchgate.net/figure/Solar-wind-magnetic-field-interacts-the-Earths-magnetic-field-fig6_326622449.
- [3] “Coursera State Estimation and Localization for Self-Driving Cars,” , 2019. URL http://tracyw.coding.me/blog/2019/08/01/coursera_state_est05/.
- [4] “Variation of temperature with altitude in the US Standard Atmosphere,” , 2019. URL https://www.researchgate.net/figure/Variation-of-temperature-with-altitude-in-the-US-Standard-Atmosphere-The-lowest-layer-fig1_231131328.
- [5] “What is the Regener-Pfotzer Maximum?” , 2019. URL <https://spaceweatherarchive.com/2018/06/07/what-is-the-regener-pfotzer-maximum/>.
- [6] “Standardization in Cryogenic Insulation Systems Testing and Performance Data,” , 2019. URL https://www.researchgate.net/publication/282556233_Standardization_in_Cryogenic_Insulation_Systems_Testing_and_Performance_Data.
- [7] Chen, S., Wan, C., and Wang, Y., “Thermal analysis of lithium-ion batteries,” , 2005. , URL <http://www.sciencedirect.com/science/article/pii/S0378775304008596>.
- [8] Lijun Gao, Shengyi Liu, and Dougal, R. A., “Dynamic lithium-ion battery model for system simulation,” *IEEE Transactions on Components and Packaging Technologies*, Vol. 25, No. 3, 2002, pp. 495–505. .
- [9] Twiggs, R., et al., “The ThinSat Program: Flight Opportunities for Education, Research and Industry,” *AIAA/USU Conference on Small Satellites*, 2018, p. all.
- [10] Tristanchó, J., “Implementation of a femto-satellite and a mini-launcher,” Master’s thesis, Universitat Politècnica de Catalunya, 2010.
- [11] “CCSDS Protocol,” , 2019. URL <https://public.ccsds.org/about/default.aspx>.
- [12] “Rover and Air Visual Environment Navigation,” , 2018. URL <https://www.colorado.edu/aerospace/current-students/undergraduates/senior-design-projects/past-senior-projects/2017-2018/rover-and-air>.
- [13] “Europa Lander for Science Acquisition,” , 2016. URL <https://www.colorado.edu/aerospace/current-students/undergraduates/senior-design-projects/past-senior-projects/2015-2016/elsa>.
- [14] “Planet Labs,” , 2019. URL <https://www.planet.com/>.
- [15] “How CCD Cameras Work,” , 2019. URL <https://starizona.com/tutorial/how-ccd-cameras-work/>.
- [16] “Imaging Electronics 101: Understanding Camera Sensors for Machine Vision Applications,” , 2019. URL <https://www.edmundoptics.com/resources/application-notes/imaging/understanding-camera-sensors-for-machine-vision-applications/>.
- [17] “SparkFun IR Array Breakout,” , 2019. URL <https://www.sparkfun.com/products/14843>.
- [18] “ANDESITE,” , 2016. URL <https://www.bu.edu/busat/missions/andesite/>.

- [19] Emily A. Ranquist, S. H., and Argrow, B., “Distributed Acceleration Sensing for Small UAS Wind Gust Estimation,” *American Meteorological Society*, 2019, p. all.
- [20] Graeme J. Marlton, K. A. N. P. D. W., R. Giles Harrison, “A balloon-borne accelerometer technique for measuring atmospheric turbulence,” *Review of Scientific Instruments*, 2015, p. all.
- [21] Phillips, D. T., “What is the Regener-Pfotzer Maximum?” , 2018. URL <https://spaceweatherarchive.com/2018/06/07/what-is-the-regener-pfotzer-maximum/>.
- [22] Phillips, D. T., “The Worsening Cosmic Ray Situation,” , 2018. URL <https://spaceweatherarchive.com/2018/03/05/the-worsening-cosmic-ray-situation/>.
- [23] “SparkFun IMU Breakout - MPU-9250,” , 2019. URL <https://www.sparkfun.com/products/13762>.
- [24] “SparkFun 9DoF IMU Breakout - ICM-20948 (Qwiic),” , 2019. URL <https://www.sparkfun.com/products/15335>.
- [25] “MTi 1-series,” , 2019. URL <https://www.xsens.com/products/mti-1-series>.
- [26] “OSD335x Datasheet,” , 2019. URL <https://octavosystems.com/docs/osd335x-datasheet/>.
- [27] Jackson, J., “Fall Final Report (FFR),” Tech. rep., University of Colorado Boulder, 2019.