Department of Electrical, Computer
& Biomedical Engineering
Faculty of Engineering & Architectural Science

Ryerson Tubesat

EDP Project Report

Introduction

As space exploration is continuingly being pushed to its boundaries from space exploration to commercialized satellites, one of the leading space exploration technology that is being developed are miniature satellites. With more companies being involved with miniature satellites such as tubesats, the appears more of a need for them then before. As tubesats are getting cheaper and cheaper to build and are relatively light weight when compared to traditional satellites. This means that corporations have a higher incentive to invest in these types of satellites. These satellites can be used for various reasons. These reasons include measuring the temperature, atmosphere and taking pictures. As more advancements are taking place within this space, our capstone design is to build a communications system that meets the requirements for the Ryerson tubesat.

While designing the communications system for the tubesat, there were many requirements. This ranged from the size of the system to the power consumed by the system and the frequency of the system. These were designed by the tubesat team and were the parameters with what we needed to build the communications system with. The communications system that was tasked to build was ultimately supposed to transfer and receive a picture that the tubesat would take. This is in order to take images in orbit that would be received at a ground station to observe the image. As this project requires knowledge from the members in communication, power and software the students building the communications have developed these skills over the past three years. Having the system split into these 3 areas each student was separately responsible for either the power, software or the communication of the tubesat. The ultimate goal for this project is to design a communications system for the Ryerson tubesat team that can successfully transmit and receive data continuously while ensuring it meets power criteria.

Acknowledgements

We would like to thank Dr Krishna Kumar for allowing us the opportunity to build a communications system for the Ryerson Tubesat Team. It is thanks to him that we were able to work with such an interesting project. We would also like to thank Ph.D Candidate and RyeTubesat Lead Mike Alger for his guidance and support throughout the semester.

Certification of Authorship

Owais Nakhuda, Bethany Santos and Ho Yin Samuel Yeung declare that we are the authors of this document and that any sources or information that is added is credited to its authors. Any data or theories which were used is also credited ranging from calculations or source code. This report is strictly written for the project of developing a tubesat for the Ryerson tubesat team.

Table of Contents

Abstract

The purpose of this project was to create a communications system for a satellite that was being designed for the ryerson tubesat team. This required us to design within the constraints that the team had developed. These constraints were to deal with power, weight and size of the design as these were already determined before we had any input. At first we had to determine what communications systems they had in place. The system were mostly treated as black boxes. As this appeared to be a time consuming process we decided to develop our own design in order to save time and get a better understanding of the communications system. We successfully designed our own communications system using an additional arduino. We also were able to meet the power constraints of the satellite. While conducting tests to ensure our communications system worked, we were able to determine that we were able to transmit and receive data files. However, we were not able to receive image files from the camera. This was due to the non-aliasing issue as we were not able to determine the cause. We were also not able to test the communications system with an in orbit satellite as we did not have sufficient time. Overall this project was a great success. We were able to design a new communication system with all the schematics in altium. This is a huge benefit as we are not dealing with black boxes anymore. We also successfully managed to receive and transmit data continuously. However, we were not able to receive the packets for image files in the correct order. This makes our project a success.

Objectives

The objective of this project is to design a communications system for the Ryerson Tubesat Team. The communication system must meet the specified power, antenna and size requirements that has been listed by the tubesat team.

Background

The power subsystem in the TubeSat is to provide power for the various components and subsysems of the TubeSat to ensure optimal operation. Various subsystems require power for auxiliary components, such as the radio, data transmitters, receivers, onboard camera, as well as any active control systems used for guidance or orientation. The general requirements for the power subsystem, in relation to the entire TubeSat, is to provide necessary power for 3 months in a Low Earth Orbit. Additionally, the power subsystem must be able to capture solar energy and recharge the onboard batteries in order to limit the amount of batteries needed for the mission and therefore reduce weight on the entire system. The power system must be able to operate during any event period, either while in eclipse or in view of the sun, meaning it should be able to conserve power for eclipse and keep the battery at nominal charge during direct contact with the sun.

The software behind the Command and Data Handling (C&DH) subsystem for the TubeSat must be able to operate fully automated with exceptions for specific command execution sent from the ground station. It must also be able to fully execute functions successfully and in a timely manner that conform to the limitation of the TubeSat circumstances; these limitations can be power, timing, or position constraints. Like all software, it must be implemented using a microprocessor as the main controlling hardware device on board. Other satellites that have been created used either a PIC, ARM, or an arduino microcontroller. For this design project, arduino microcontroller will be used. The purpose of the software is to be able to gather data from on board sensors such as IMU’s, temperature sensors, and lux and send the gathered data to ground station with no loss in data.

Theory and Design

For the RyeTubeSat Project, there are two main concepts that needed to be considered for the design: The Power and Link Budgets. The Power Budget is the total available system power that can be used for the system and its functions. Theoretically, one full orbit of the Tubesat takes one hour. For half that orbit, the tubesat will be able to generate power from the solar cells to restore energy used from the satellite. The Tubesat's daily power budget during operations can be calculated with the following equation:

$$10 \geq (P_{sensor} * t_{meas}) + (P_{comm} * t_{send}) - (P_{sun} * t_{charge}) \ \ \textbf{[Wh]}$$

where 10 is the estimated work-case maximum battery ampacity in [Wh]. The main assumption with this equation is that system operations do not resume until the battery is restored. Figure 1 shows the estimated System Power Consumption and restoration on a daily basis. The Main operational assumptions about the system were made:

- One full orbit takes an hour
- Half of the orbit time will be in the sunlight (regardless of the time of day)
- The system does not perform any major measurements while in sunlight (i.e., all payload functions and instrumentation are powered off)
- The solar panels do not provide any power to the system while measurements are being taken

	Active [W]	Idle [W]
Payload	0.0028	0.000045
	0.0022	0.000045
	0.0005	0.00009
	0.0004	4.32E-06
	0.0234	1.05E-05
	0.375	0
Total	0.4043	0.000195
Wh	1.94064	2.34E-05

	Active [W]	Idle [W]
Comm	0.55	0
	0.135	0.135
	1.25	0
	0.01	0.01
Total	1.945	0.145
Wh	0.2334	0.696

Orbit time **(hours)**	1
Charging time per orbit **(hours)**	0.5
sensing time per orbit **(%)**	40
tranamitting time per orbit **(%)**	1
Solar Cell Power [W]	0.0636
Wh restored per orbit	0.0318

		Total Wh Available (Assuming Battery Full; ideal conditions)	10
HOURS ACTIVE per DAY	4.8	sensing	2.63664
	0.12	communicating	0.233423
		Used Wh today	2.870063
		Restored Wh/day	0.7632
		days to restore used power	3.760565
		estimated operating time remaining w/o charging (hours)	7.129937

Figure 1. calculation Table used for Power Budget Estimations

The Link budget is used to determine the theoretical operating distance given the communication system's specifications. With the following equation:

$$\mathbf{P_{RX}[dBm] = P_{TX}[dBm] + G_{sys}\ [dB] - L_{sys}\ [dB] - L_{ch}\ [dB] - M[dB]}$$

where L_{ch} is the freespace path loss:

$$L_{ch} \approx Freespace\ Path\ Loss = 20D + 20F \quad 20\frac{4}{c} - G_{ant_{tx}} - G_{ant_{rx}}$$

Figure 2 shows the estimated Link budget available to the System for downlink communications. In order to generalize the calculations, the following assumptions were made about the system in regards to its operating specifications:

- The antenna gain is the same for the receiver and transmitter
- The Receiver sensitivity for the transceiver and Ground station are equal

Fill out info for sections highlighed in yellow						
Equations Used Referenced from Here		Ideal conditions	Worst case (Guess)	Worst case (suggested assumption)		
Free Space Path Loss Equation Referenced from Here						
fade margin [dBm]	Additional compensation from unknown factors	0	100	30	20	10
Transmitter	Maximum Transmission Power [dBm]	47				
	Frequency [MHz]	433				
Receiver	Sensitivity [dBm]	-160				
Antenna	Gain [dBm]	21				
Losses	freespace[dBm]	92.71395521				
assume 700km total distance	300					
	Estimated Link Budget (Maximum Path Loss)	135.2860448	35.28604479	105.2860448	115.286	125.286
	Maximum Range[km]	320.4898968	0.003204899	10.13478041	32.04899	101.3478

Figure 2. Calculation Table used for System Link Budget Analysis

The main bulk of the software in the TubeSat goes into the Command and Data handling as well as in communication. The end goal of this project is for the TubeSat to launch and reach low earth orbit (LEO) and to be fully functional on its own while in LEO. This means that the satellite must be fully automated, with a few exceptions when the satellite receives commands from ground station. The satellite must be able to collect accurate data from various sensors mounted on board such as a acceleration, satellite orientation, light intensity, temperature, as well as take a real time photo of the earth all in a timely manner. This makes scheduling of when different functions should run at certain times extremely important as the satellite will only be able to operate these functions during a time window that lasts for a few minutes ranging from 5-10 minutes while it is in orbit. Because of this most of the code structure will be sequential rather than concurrent, where the code will check for a condition one at a time before it moves on the next condition statement. For example, the code will check if it is time to start collecting data from the sensors, if no, it moves on to check if it is time to take a photo of the earth, if not, then it moves on to check if it has received commands from ground station, if not then it executes remaining tasks, and loops back again to iterate through the conditions it needs to check. Rather than the code checking for these conditions in simultaneously and having tasks run concurrently, also known as parallel programming. Parallel programming was not implemented for the TubeSat simply because it is more difficult when it comes to timing and syncing everything together, and is more prone to bugs, and errors since different code modules may be handled by different groups of people. If the code is not well interlaced with one another, a variable could be changed unintentionally by another program without being noticed wich can cause serious issues in the satellite, and can be difficult to debug.

In theory, the main goal of the C&DH software is to capture data using Serial Peripheral Interface (SPI) communication protocol from sensors: Inertial Measurement Unit (IMU), lux, thermometer, camera, and save it to an onboard SD card. The satellite will wait and continuously poll to check if it has received a signal from the ground station. Once a signal from the ground station has received it gives the satellite a 'go' to retrieve all the saved data from the SD card and send the data over serial to ground station. Before it reaches orbit, the satellite must know when to turn on. It is critical that the satellite turn on once it has reached LEO and is in equilibrium; this is because if the satellite turns on at the wrong time (ie. during launch) it will record incorrect/corrupted data, and not only that, the frequency in which the satellite used to operate and communicate with the ground station can interrupt with the rocket frequency and can cause interruption with the communication between ground station and the rocket. So there are dire consequences if the satellite turns on at the wrong time, and therefore must be taken care of with meticulously in the code ensuring successful operation. The initial design of the software flow for the TubeSat start up is entirely dependent on an onboard timer, but was later changed to a switch plus an onboard timer. For the initial design, after launch, the TubeSat will execute *mission level code* where the TubeSat checks to see if it is undergoing launch, It does this by reading acceleration values from the IMU and if the code reads values that exceeds 3Gs after 3 consecutive checks, then we can conclude that the TubeSat is in motion and is still launching. After it has confirmed this, the code will start a 45 min timer for when all satellite functionalities will begin (ie, hardware initialization and checking, read and store values from sensors, poll for ground station commands, etc). Since the launch duration is approximately 20 mins, this gives the satellite a 25 min safety windows to ensure it does not start up during launch.

The software for the C&DH was implementing using arduino modules; the initial software design was implemented on Arduino Minis, but would later switch to Arduino Nano due to simpler, more robust data handling, as well more efficient use for memory. There is a total of two Arduino microprocessors used for this subsystem: one for command and data handling such as to activate timed execution for data collection and storing without any misinterpretation and/or loss of data in the process, and the other for communications used as a communication modem to handle transmission of collected data from the onboard sensors to the radio and antenna boards and finally to ground station. The payload board contains a total of 4 sensors: 2 temperature sensors, lux, and an IMU which house an accelerometer, a gyroscope, and a magnetoscope. The overview of hardware and software interactions is seen below in Figure 3.

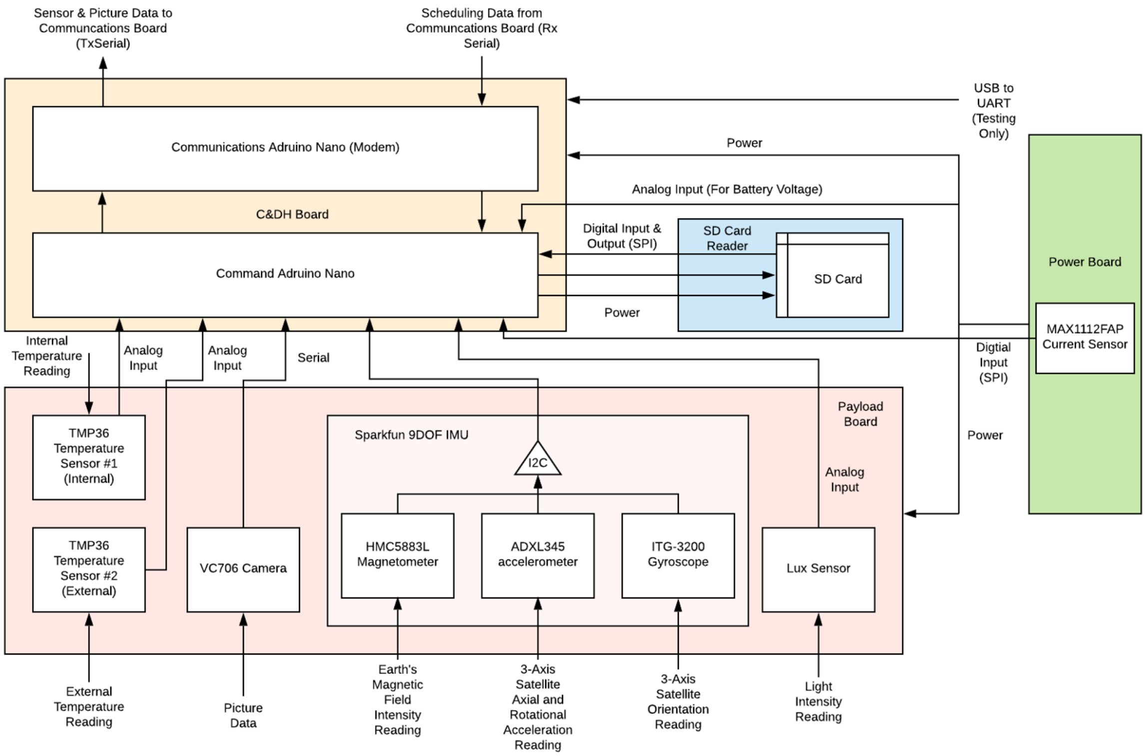

Figure 3. hardware interfaces and software interactions between the C&DH Board, Communications Board, SD Card Reader, Payload Board

Figure 4 below shows a high level view of the software of the satellite. During start up, it declares all global variables and define all constants, then it goes on to setup where it initializes the baud rate for serial communication and all the required hardware such as SD card reader and breakout, IMU, etc. After this the code runs an infinite loop where in consistently gets the current time and checks this against two timer interval. One determines if it is time to collect data from the sensors and the other to determine if it is time to take a snapshot of the earth and store collected data and picture on to the SD card. Once if completes the checks and executes the required tasks, it goes on to check for any commands it might have received from ground station and executes the tasks associated with those commands.

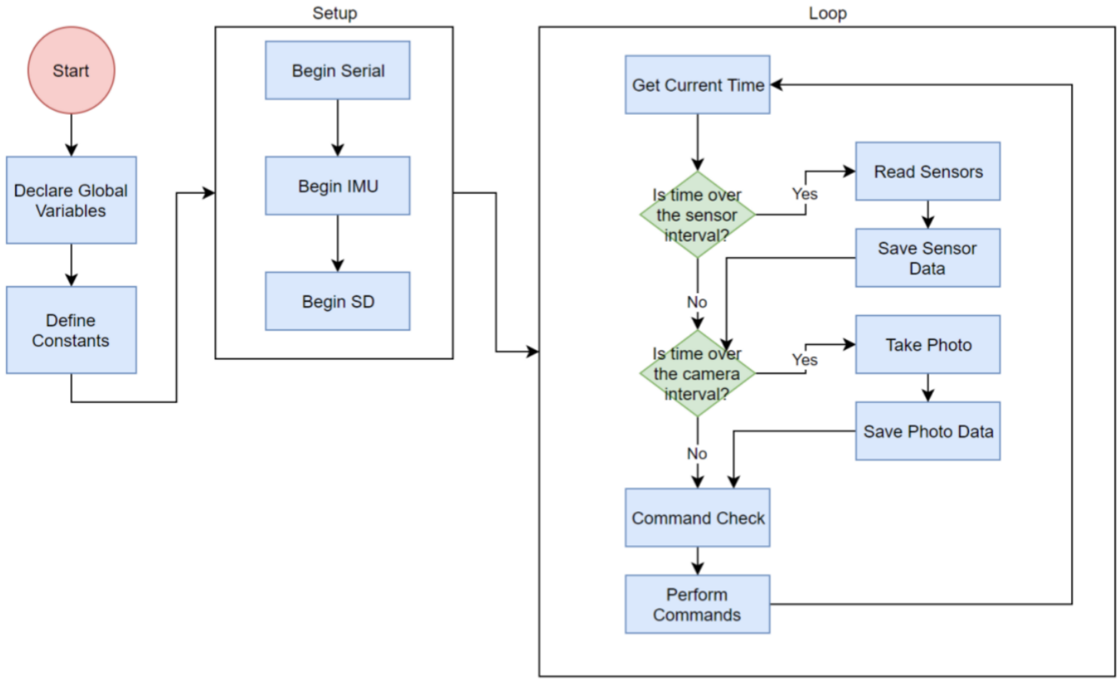

Figure 4. High level view of TubeSat C&DH software structure

Figure 5 below visualizes the software structure and sequence on how the code reads data files. It has two security checks before reading the file: (1) it checks to see if the file exists and (2) if the file has been opened. If these two conditions are not met, then the code exits, if it is met, then the code proceeds to determine the file size to anticipate how much data it will read. It then reads data 64 bytes at a time and entered into a buffer to be sent of serial. This sequence is repeated until it reaches the end of the file.

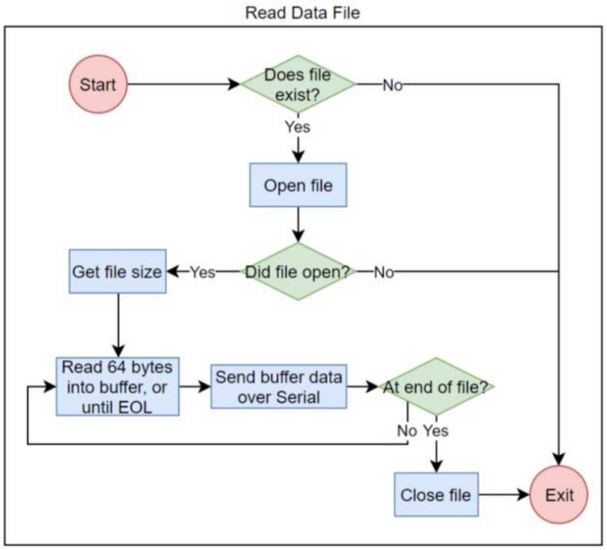

Figure 5. Software breakdown structure of how it reads files

Figure 6 below shows the breakdown of the softwares decision path for each command received from ground station. The 'XX' from each command represent a number, and this number represent different meaning depending on the type of command it is associated with. The command DXX and PXX sets a 'go' flag for data retrieval from the sensors and for photo retrieval respectfully. in this case the XX represents the file number to ensure no duplicates in data file. The command SXX and CXX are scheduling commands that set timer for the when the camera takes a photo and for when the satellite reads data from sensors respectfully. For this case, the XX are numbers in seconds.

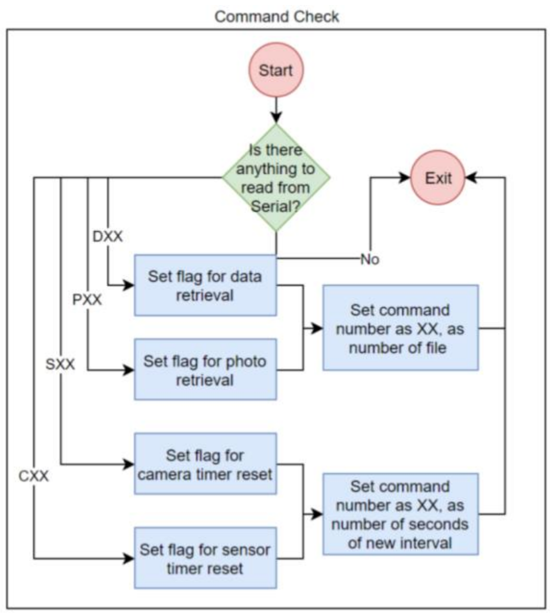

Figure 6. Software breakdown structure of how TubeSat checks for incoming commands from ground station

Figure 7 below goes into more detail on the software's flow for commands from ground station. It is a sequential check of conditions in the following order: data flag, photo flag, sensor timer flag, and camera timer flag. If any of the conditions are met, it proceeds to execute the tasks for that commands as described in the Figure below. The code waits until all tasks required are finished before it proceeds to check the next condition.

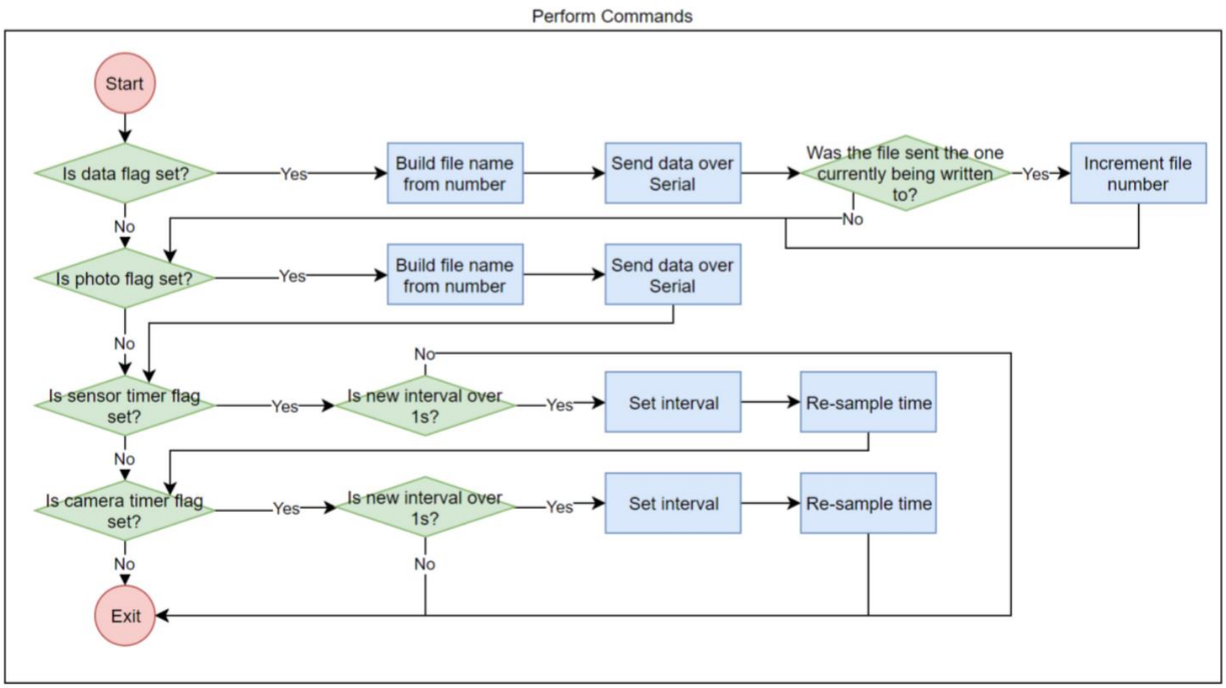

Figure 7. Software breakdown structure of performance of each commands retrieved from ground station

Figure 8 below describe the software structure for the ground station. The ground station is the only point of contact of the satellite on earth. The ground station software platform is MATLAB and compared to the fully automated satellite, the ground station is mostly user operated where the software is composed of modules that can be called at the discretion of the ground station operator. The second block of the software structure labeled "Transmit Command Prompt" is where the ground station operator is prompted to enter a command that branches out to three possible execution paths as seen below. From the ground station, the operator can direct the satellite what data to send over at what time, as well as to send changes in scheduling for satellite functions execution times.

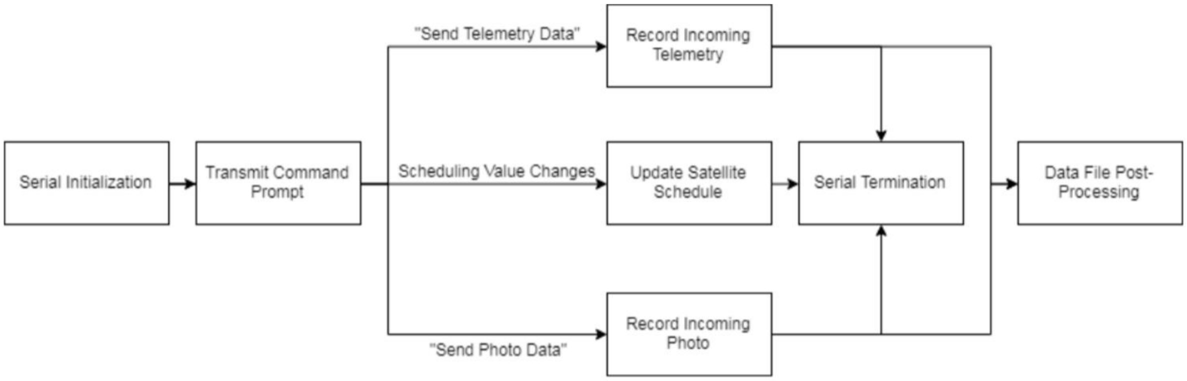

Figure 8. Ground Station Software breakdown structure

Alternative Designs

Due to the nature of the RyeTubeSat Project, the majority of the System Design was already locked in and could not be revised. This included, but not limited to, the Following Components:

- Battery Capacity and Voltage
- Power DIstribution System
- Solar Panel Capacity and Voltage
- Transmission Frequency and Power
- Onboard Instrumentation and equipment

However, certain parts of the system, such as the Logic Controller and Communication Modem, were isolated enough from the other Subsystems of the Tubesat that revisions could be made to the system without any issues.

The Payload and Communications portions of the Satellite needed to be redesigned, due to the majority of the system components being treated as black boxes and incapable of troubleshooting should problems occur. This was especially true for the original Transceiver Modem, which consisted of Three separate PIC microcontroller, each containing another portion of the Modem Protocol (Transmission, Receiving, AX.25 Encoding, etc.). During Testing, it was discovered that one of the PICs status lights could not be turned off. This status light turned on immediately as the system was powered on. Troubleshooting the hardware and software of the communication PIC yielded no results, as the system, with all features unchanged worked previously.

Another issue was the onboard System Memory. Due to the quantity of sensors and system states, it was determined during development that external memory would need to be added in order to safely record the payload data during the tubesats operational lifespan.

The two main issues noted above justified a redesign of the communications board and payload board. The main change was to swap out the original three communications PICs with an open-source AX.25 modem using the arduino. The next change that needed to be made was the addition of external memory in the form of an SD Card Reader. Figure 9a) shows the revised schematic diagram for the Tubesat's Controller Circuit.

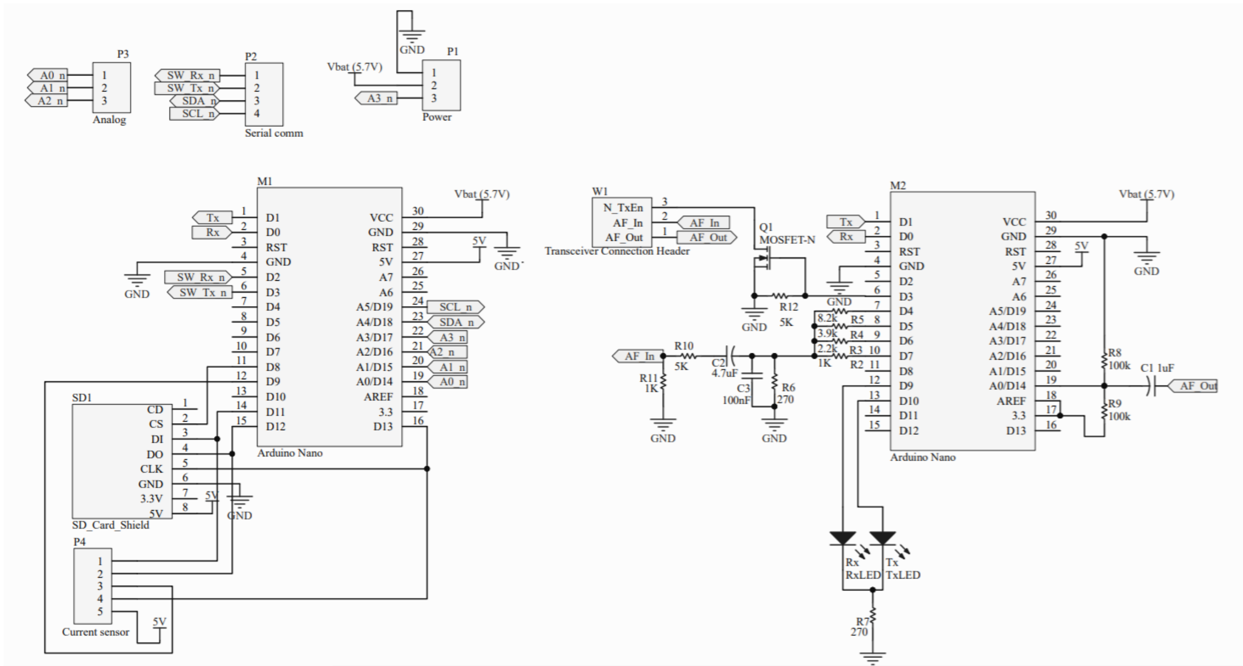

Figure 9a). Controller Board Schematic with new AX.25 Modem and SD Card Reader

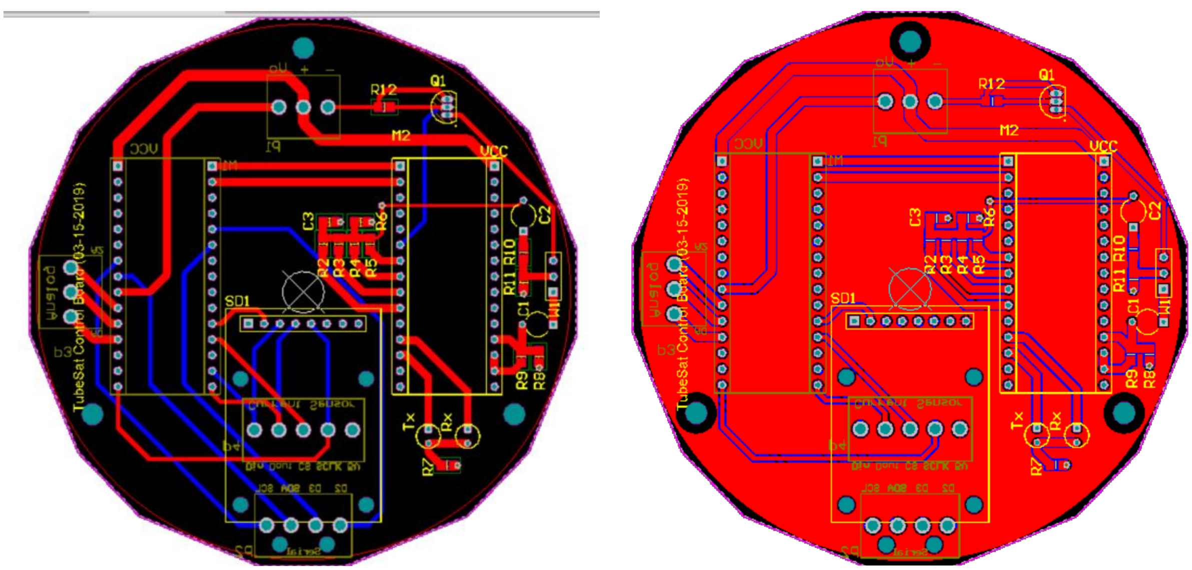

Figure 9b), c) PCB Layout of the Controller Circuit without the Power Plane (Left) and With the Power Plane (right)

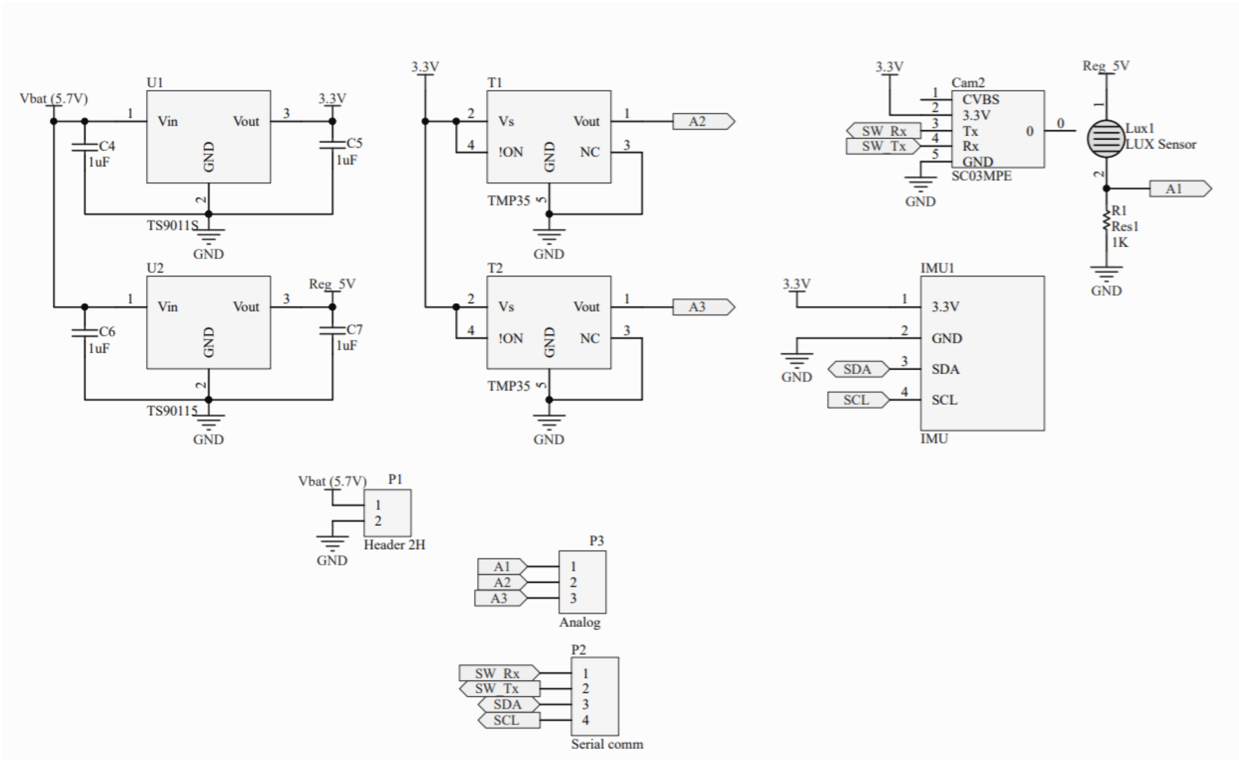

Figure 10a). Board Schematic

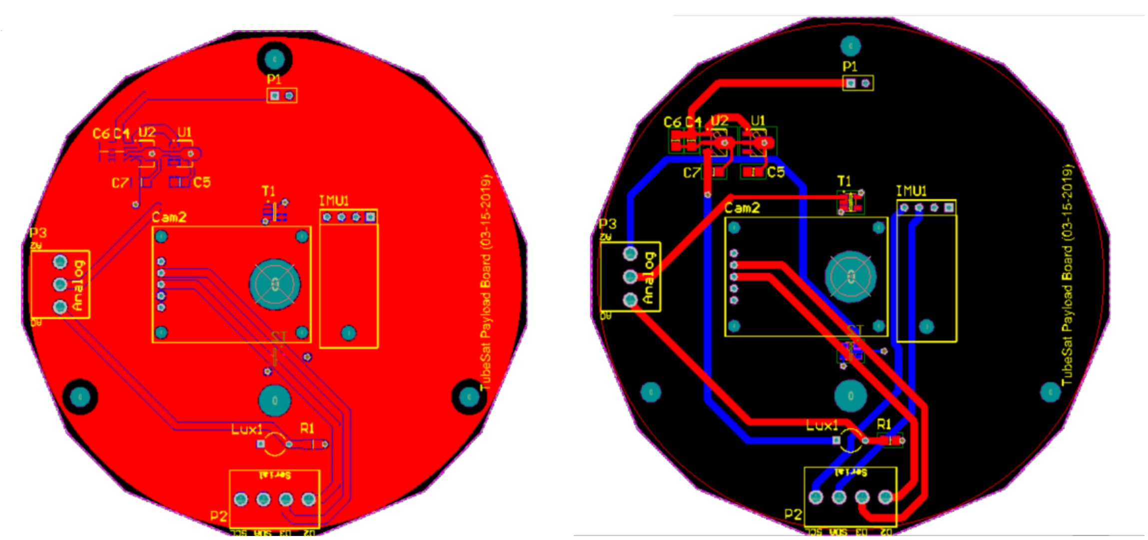

Figure 10b) and c). *PCB Layout of the Payload Circuit with the Power Plane (Left) and without the Power Plane (right)*

Material/Component Cost

Antenna PCB Component List

- K1 CONN_3 Farnell 973- 1083
- P1 CONN_2 Farnell 973- 1601
- PA1 PAD_ANT
- PA2 PAD_ANT
- SW1 SWITCH_INV Manufacturer SSP- Taush 050A http://www.tauch.com.tw/Product- SS.htm
- TU1 TUBESAT_PANEL
- U4 SMA FARNELL 124-890 or 124- 8989

Power Management Board Component List

- B1 BATTERIE Li-Ion
- Propel 3.7V 5200mA
- C1 220pF 1206 Farnell 940-6468
- C2 4.7uF 1206 16V Farnell 922-7946
- C4 22uF 1206 16V Farnell 952-7800
- C5 4.7uF 1206 16V Farnell 922-7946
- C6 4.7uF 1206 16V Farnell 922-7946
- C7 0.1uF 0805 Farnell 940-6387
- C8 1uF 1206 25V Farnell 922-7873
- C9 1uF 1206 25V Farnell 922-7873
- C10 100nF 0805 Farnell 940-6387
- D1 CRS06 Farnell 130-0795
- D2 CRS06 Farnel 130-0795
- D3 CRS06 Farnel 130-0795
- D4 CRS06 Farnel 130-0795
- D5 CRS06 Farnel 130-0795
- D6 CRS06 Farnel 130-0795
- D7 CRS06 Farnel 130-0795
- D8 CRS06 Farnel 130-0795
- D9 CRS06 Farnel 130-0795
- L1 10uH 10x10 shielded Epcos Farnell 742-9444
- P1 CONN_2 Farnell 973-1601
- P2 CONN_2 Farnell 973-1601
- P3 CONN_2 Farnell 973-1601
- P4 CONN_2 Farnell 973-1601
- P5 CONN_2 Farnell 973-1601
- P6 CONN_2 Farnell 973-1601
- P7 CONN_2 Farnell 973-1601
- P8 CONN_2 Farnell 973-1601
- P9 CONN_2 Farnell 973-1075
- P10 CONN_2 Farnell 973-1075
- P11 CONN_2 Farnell 9731199
- P12 CONN_2 Farnell 973-1075
- P13 CONN_3 Farnell 973-1083
- P18 CONN_5 Farnell 973-1105

- P21 TST Test Point
- P22 TST Test Point
- P23 TST Test Point
- P24 TST Test Point
- P25 TST Test Point
- P26 TST Test Point
- P27 TST Test Point
- P28 TST Test Point
- P29 CONN_2 Farnell 973-1075
- R1 1.5 0805 Farnell 933-3959
- R2 1.5 0805 Farnell 933-3959
- R3 1.5 0805 Farnell 933-3959
- R4 1.5 0805 Farnell 933-3959
- R5 1.5 0805 Farnell 933-3959
- R6 1.5 0805 Farnell 933-3959
- R7 1.5 0805 Farnell 933-3959
- R8 1.5 0805 Farnell 933-3959
- R9 470K 1206 Farnell 933-7458
- R10 27K 1206 To be adjusted
- R12 12K 1206 Farnell 933-7075
- R13 47K 1206 Farnell 933-7350
- R16 180K 1206 To be adjusted for V between 4.1 to 4.2V
- R17 R 1206 To be adjusted
- RV1 50K Farnell 514-858
- RV2 50K Farnell 514-858 TU1 TUBESAT_PANEL PCB
- U1 MAX9929 Farnell 1673101
- U2 MAX9929 Farnell 1673101
- U3 MAX9929 Farnell 1673101
- U4 MAX9929 Farnell 1673101
- U5 MAX9929 Farnell 1673101
- U6 MAX9929 Farnell 1673101
- U7 MAX9929 Farnell 1673101
- U8 MAX9929 Farnell 1673101
- U9 LT3021 Farnell 127-3637
- U11 LM2731 Farnell 818-1640
- U12 MAX1112 Farnell 137-9848

Radiometrix Transceiver PCB Component List

- C1 22uF 2112 16V Farnell 952-7800
- C2 4.7uF 1206 16V Farnell 922-7946
- C3 4.7uF 1206 16V Farnell 922-7946
- C4 100nF 1206 Farnell 940-6557
- P1 CONN_5 Farnell 973-1105
- P2 CONN_2 Farnell 973-1075 (Molex)
- P3 CONN_6 Farnell 973-1113 (Molex)
- P4 CONN_2 Farnell 973-1075 (Molex)
- P5 CONN_2 Farnell 973-1075 (Molex)
- R1 47K 0805

- R2 R 0805 To be adjusted
- R3 47K 0805
- R4 R 0805 To be adjusted
- R5 R 0805 To be adjusted
- R6 R 0805 To be adjusted
- R7 10K 0805
- R8 1K 0805
- R9 R 0805
- R10 R 0805
- R11 R 0805
- TU1 TUBESAT_PANEL PCB
- U1 TR2M Radiometrix
- U2 AFS2 Radiometrix
- U3 SMA Farnell 116-9631
- U5 LT3021 Farnell 127-3637
- U6 LT3021 Farnell 127-3637

Microhard Transceiver PCB Component List

- C1 22uF SM1210 16V Farnell 952-7800
- C2 22uF SM1210 16V Farnell 952-7800
- C3 100nF SM0805 Farnell 940-6387
- C4 10nF SM0805 FARNELL 940-6352
- C5 22uF SM1210 16V FARNELL 952-7800
- P2 CONN_8 FARNELL 973-1121
- P3 CONN_8 Farnell 973-1652
- P5 CONN_5 FARNELL 973-1105
- P6 CONN_4 FARNELL 973-1091
- R1 10K SM1206 FARNELL 110-0218
- R2 10K SM1206 FARNELL 110-0218
- R3 10K SM1206 FARNELL 110-0218
- R5 4.7K SM0805 FARNELL 109-9805
- R6 R SM0805
- RV1 50K Farnell 514-858 TP1 TST Test
- Point TP2 TST Test
- Point TU1 TUBESAT_PANEL PCB
- U1 NANO Modem Microhard N920 or N2400 Connector: see Comments on Drawing
- U2 LP38842MR-ADJ
- PSOP-8 FARNELL 128-6792

Microcontroller PCB Component List

- C1 4.7uF 1206 16V Farnell 922-7946
- C2 4.7uF 1206 16V Farnell 922-7946
- C3 18pF 0805 Farnell 721-992
- C4 18pF 0805 Farnell 721-992
- C6 100nF 1206 Farnell 940-6557
- C7 100nF 1206 Farnell 940-6557
- C8 100nF 1206 Farnell 940-6557

- C9 100nF 1206 Farnell 940-6557
- C10 39pF 0805 Farnell 722-030
- C11 100nF 1206 Farnell 940-6557
- D1 LED
- K1 CONN_3 Farnell 973-1083
- K2 CONN_3 Farnell 973-1083
- P1 CONN_5 Farnell 973-1105
- P2 CONN_4 Farnell 973-1091
- P3 CONN_5 Farnell 973-1105
- P4 CONN_7 Farnell 973-1121
- P5 CONN_2 Farnell 973-1601
- P6 CONN_4 Farnell 973-1091
- P7 CONN_2 Farnell 973-1601
- P8 CONN_2 Farnell 973-1601
- P9 CONN_2 Farnell 973-1601
- R1 R 0805 To be adjusted
- R2 47K 0805 Farnell
- R3 R 0805 To be adjusted
- R4 R 0805 To be adjusted
- R5 R 0805
- R6 100K 0805 Farnell 933-3738
- R7 100K 0805 Farnell 933-3738
- R8 R 0805 To be adjusted
- R9 R 0805
- R10 R 0805 To be adjusted
- R11 R 0805
- R12 R 0805
- R13 10K 0805 Farnell 933-3720
- R14 10K 0805 Farnell 933-3720
- R15 1K 0805 Farnell 933-3711
- R16 10K 0805 Farnell 933-3720
- RV1 1K TU1 TUBESAT_PANEL
- U1 LT3021 Farnell 127-3637
- U2 MAX9929 Farnell 167-3101
- U3 BASICX-24
- U4 MX614
- U5 LM311N
- U6 PIC16F627
- X1 3.579545MHz Farnell 153 8756 Farnell 164 0858

Measurement and Testing Procedures

C&DH Software Test Results

Throughout the C&DH software design process, various test were conducted to ensure changes made do not interfere or hinder functionality during integration and to make sure that all functions are performing accurately and as expected. Table 1 below lists in detail the test that have been conducted for the C&DH subsystem.

Test	Date Performed	Pass/Fail	Notes
C&DH Arduino Test	01/18/19	Pass	Arduino has been tested and is in full working order
Arduino upload Test	01/09/19	Pass	Arduino upload test completed, software complies and runs as expected
Communication via USB Test	01/09/19	Pass	Communication between Arduino and PC completed, communication establishes as expected
Camera Snapshot Test	01/23/19	Pass	Camera snapshot test completed and verified multiple times, camera performs as expected
SD Writing Test	02/01/19	Pass	Camera snapshot function and sensor functions can write to SD as expected
SD Reading Test	02/05/19	Pass	Camera and sensor data can be read off SD as expected
Full Code Test	02/13/19	Pass	Test of full current code loop (snapshot, write to SD, read from SD at ground station) has been completed and is working
C&DH Radio Communication Test	03/27/19	Pass	Communications and C&DH software is fully integrated and performs as expected, 64-byte packets created and send as expected
C&DH Long Duration Test	N/A	N/A	To be completed by after integration between all subsystems achieved

Table 1. Outline of C&DH software test and results

Ground Station Software Test Result

Communication connection and transmission of data was tested to ensure a connection between the satellite and ground station is established and that the ground station is able to receive accurate data from the satellite. Figure 11 below shows what is received at ground station after prompting a single command for telemetry retrieval. A total of 17 out of the 23 packets sent from the TubeSat were received demonstrating a 74% data retrieval success rate. The missing data can be due to noise present from other devices within the vicinity of the test area. On the other hand, the ground station was able to successfully retrieve and organize data sent from the TubeSat.

```
1.00 0.00 0.00 2.00 2.85 26.14 476.65 142.00 96.00 -17.       {281
0 2776.00 4100.00 -0.11 -0.94 -0.31 0.00 1.00 0.00 0.00 4.00 2. {282
0 2756.00 4106.00 -0.12                                         {283
-0.94 -0.31 0.00 1.00 0.00 0.00 6.00 2.86 26.00 532.21 145.00 8 {284
.00 -2.00 2680.00 2770.00 4066.00 -0.12 -0.94 -0.31 0.00 1.00 0 {285
00 4111.00 -0.11 -0.94 -0.31 0.00 1.00 0.00 0.00 10.00 2.88 26. {286
0.31 0.00 1.00 0.00 0.00 12.00 2.88 26.04 497.49 145.00 86.00 - {287
53.00 2701.00 2695.00 4071.00 -0.12 -0.94 -0.31 0.00 1.00 0.00  {288
044.00 -0.11 -0.94 -0.31 0.00 1.00 0.00 0.00 16.00 2.88 25.90 5 {289
4042.00 -0.11 -0.94 -0.                                         {290
31 0.00 1.00 0.00 0.00 18.00 2.89 26.04 502.01 138.00 72.00 -10 {291
00 2694.00 2723.00 4090.00 -0.11 -0.94 -0.31 0.00 1.00 0.00 0.0 {292
.00 -0.11 -0.94 -0.32 0.00 1.00 0.00 0.00 22.00 2.88 26.04 495. {293
0.00 1.00 0.00 0.00 24.00 2.88 26.09 496.67 120.00 99.00 18.00  {294
654.00 2753.00 4019.00 -0.11 -0.94 -0.31 0.00 1.00 0.00 0.00 2. {295
.11 -0.93 -0.32 0.00 1.00 0.00 0.00 4.00 2.89 25.95 526.12 126. {296
0 81.00 2.00 2707.00 2678.00 4020.00 -0.11 -0.94 -0.31 0.00 26. {297
```

Figure 11. Telemetry Data

Figure 12 below shows what the ground station received after prompting for photo data retrieval from the TubeSat. For a reconstruction of a JPEG image, it is essential that 100% of the data must be transmitted and received. As seen in the figure below, the data received came included extra symbols and characters that is not payload information specific, because of this, the JPEG formatting was lost during transmission and the post processing of data required meticulous cleaning up.

Figure 12. Picture Data

Power Test Result

Testing of the solar panels involved rotating them through various incident angles relative to the incoming ‘sun’ vector. To ensure repeatability of our testing, a motorized apparatus was created that would automatically rotate the panel through preset incident angles while voltage and current values were recorded. The apparatus was controlled via an Arduino UNO microcontroller with a MATLAB interface. The interface allowed the user to enter voltage and current at each incident angle which would then be recorded in MATLAB and converted into visual graphs upon completion of the test.

Battery Discharge Testing

The Battery Discharge Test was performed using a 5-Watt, 10 Ohm power resistor and emergency cut-off switch in the event of a fault. The test was performed based on the assumption that the power rating of the resistor, when compared to the maximum power draw of the battery, would be able to sufficiently tolerate any resistance fluctuations that may occur on the resistor during testing. As such, it can be assumed that the resistsance of the resistor stayed at around 10 ohms. The open circuit voltage levels were then monitored with an Arduino while the current and close circuit voltage were measured using a multimeter every hour over the course of the discharge cycle.

The test was performed at room temperature (21°C and -20°C). The following equipment was used to perform the battery discharge test:

- Tenergy Lithium-ion 3.7V 5200mAh battery
- Tenergy battery charger
- 10-Ohm 5-Watt power resistor
- Jumper wires
- Vacuum chamber and vacuum pump
- Freezer
- Arduino Nano
- Multimeter

Power system integration test (Rotating Platform)

A mock version of the tubesat was created for the purposes of mounting the solar panels in their appropriate orientation and to then simulate the satellite slowly rotating with respect to the sun. The goal of the test was to validate the full power system and determine its flight readiness. The test allowed live acquisition of current data from each of the 8 solar panels, the battery voltage was also monitored every 15 minutes. The apparatus consisted of 3D printed tube sat frames, a slip ring, stepper motor and an Arduino Leonardo microcontroller. A picture of the setup can be seen in the figure below.

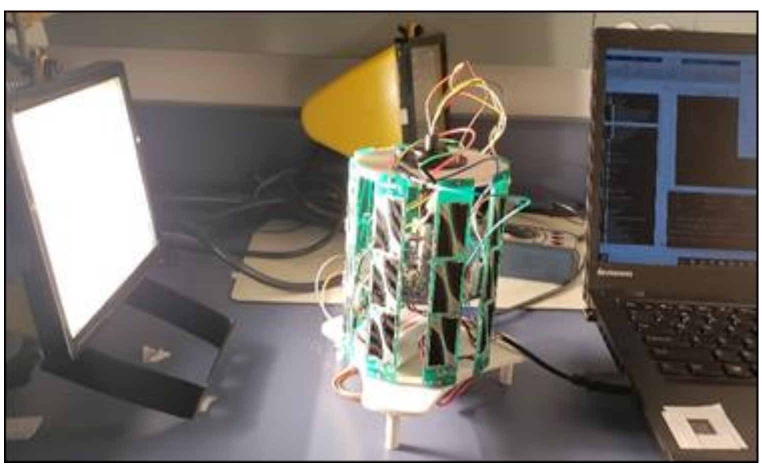

Figure 12.: Rotating integration test setup

Measurement Results

Table 2: Properties of Panel and TrisolX Wing

Property	Panel	Cell
Voltage (Open Circuit)	7.86 V	2.62 V
Voltage (Max Power)	7.00 V	2.33 V
Current (Max Power)	29.1 mA	14.5 mA
Power (Max)	204.4 mW	34 mW
Efficiency	28%	28%

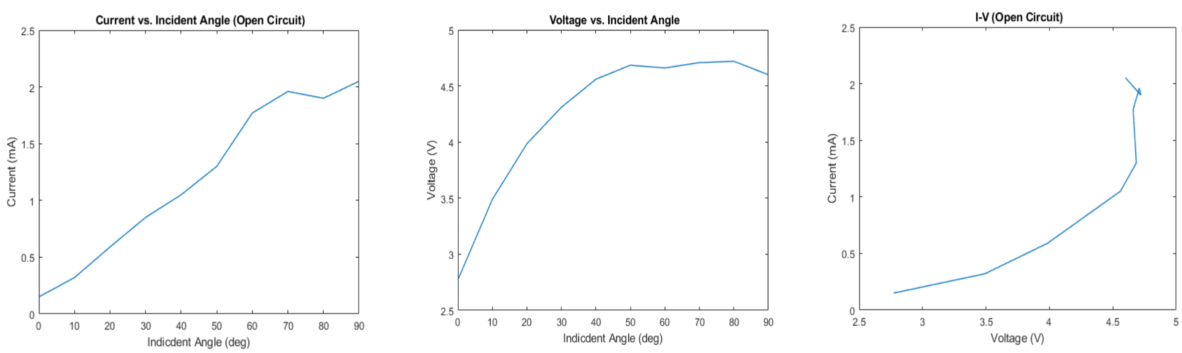

Figure 13: Open Circuit Test Results from Panel B

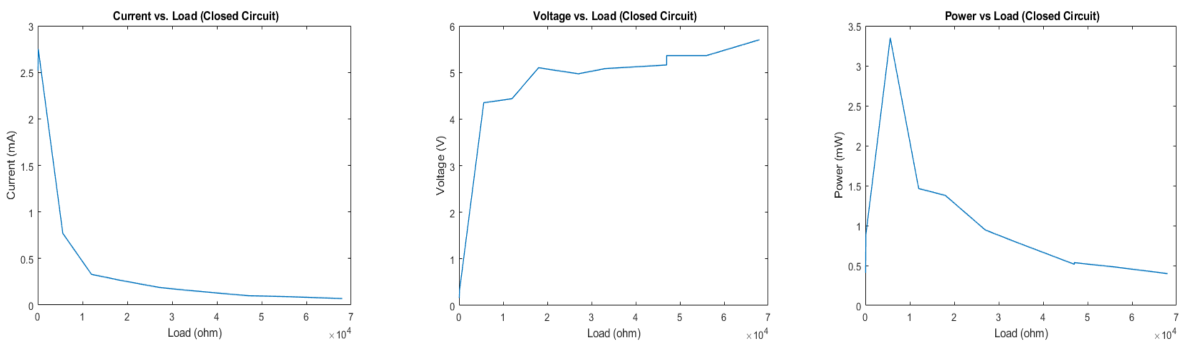

Figure 14: Closed Circuit Test Results from Panel B

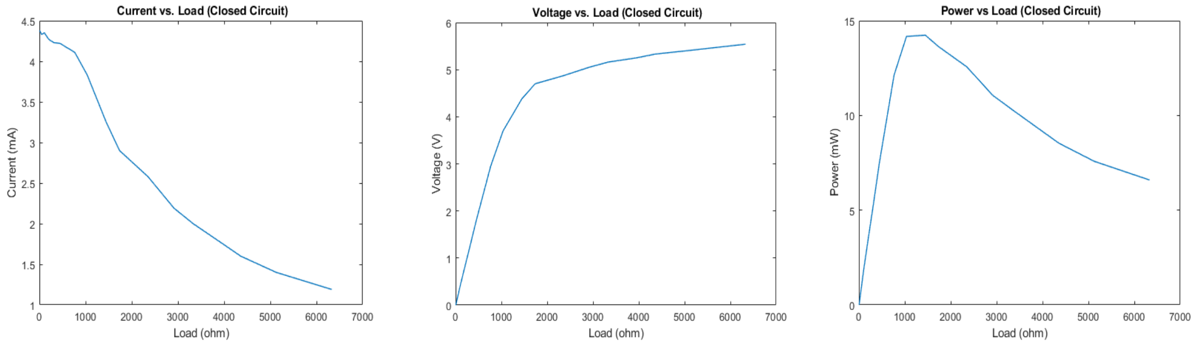

Figure 15: Closed Circuit Testing Panel B (New Light Source, more data points)

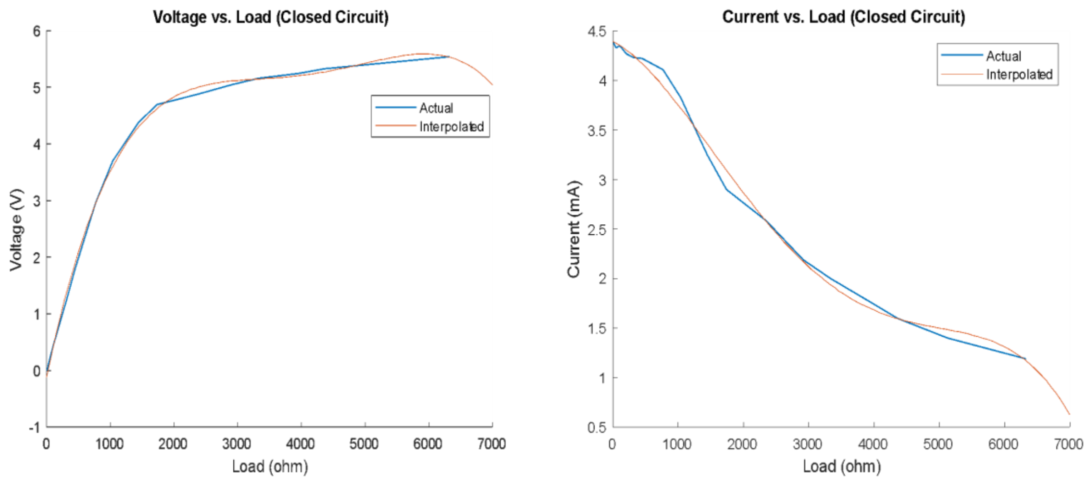

Figure 16: Computational model of based on test results (Closed Circuit)

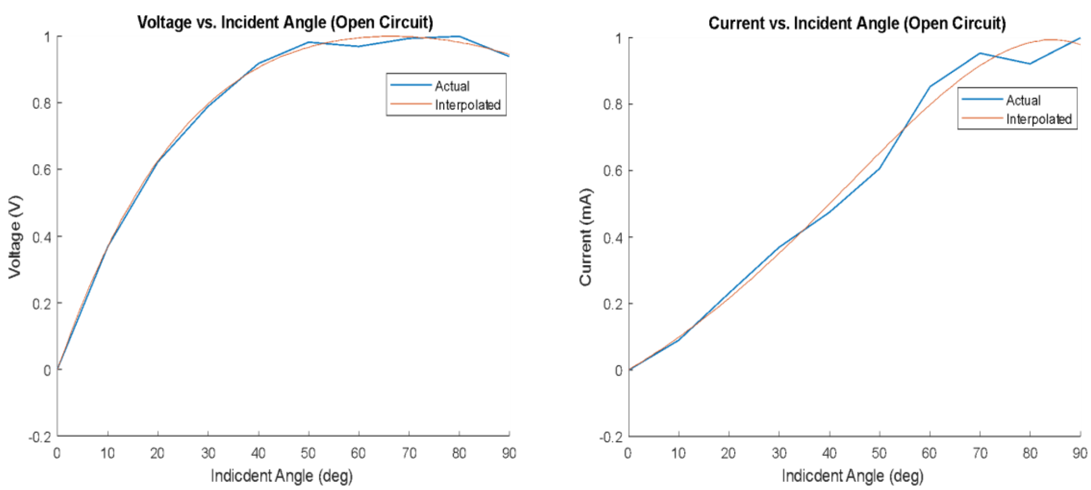

Figure 17: Computational model based on test results (Open Circuit)

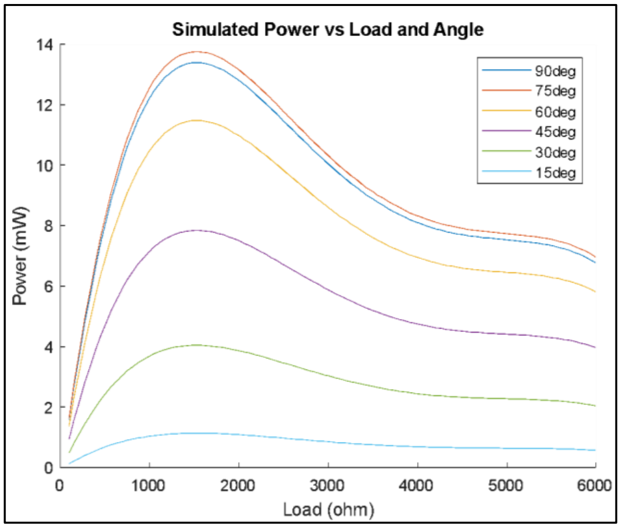

Figure 18: Sweep of computational solar panel model under various loads and incident angles

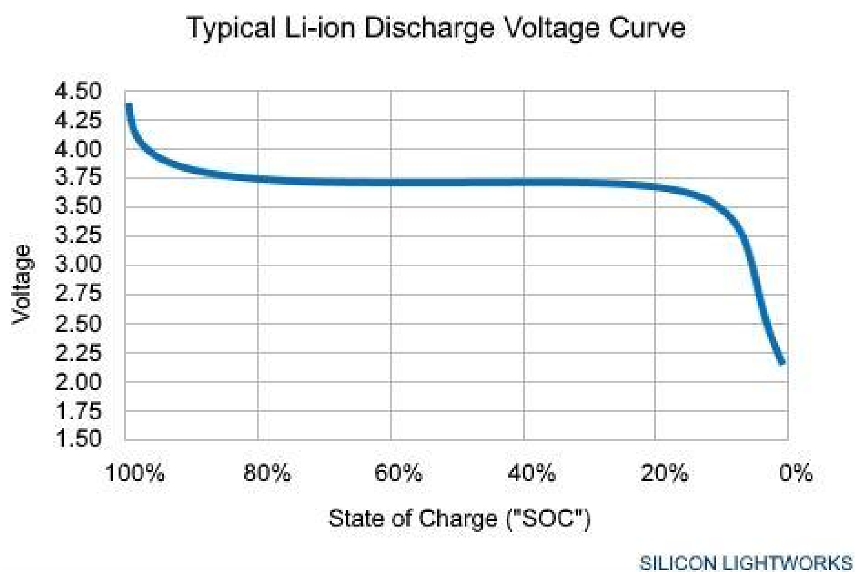

Figure 19: Typical Discharge Curve for Li-Ion battery [1]

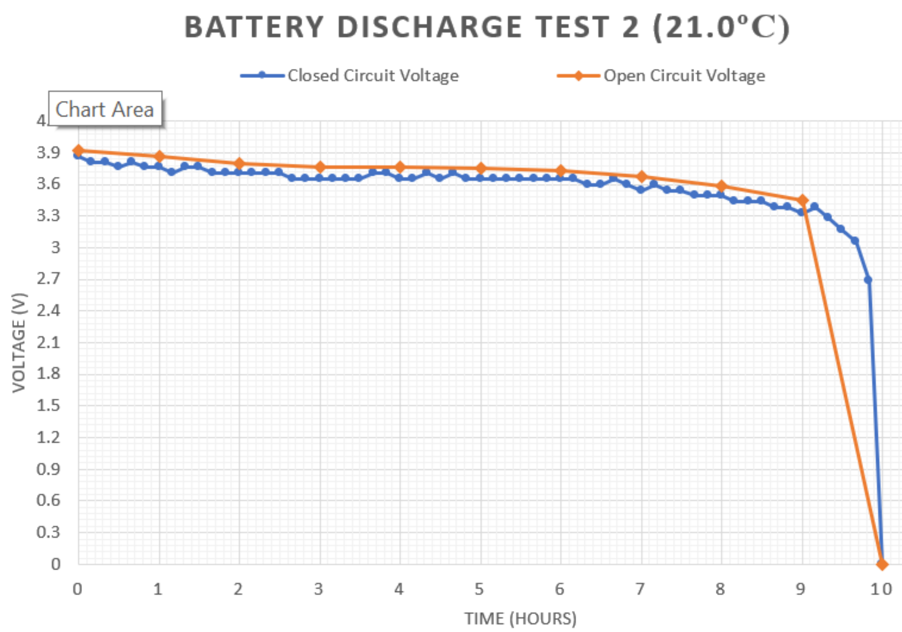

Figure 20: Discharge Open and Closed-Circuit Voltage (20°C)

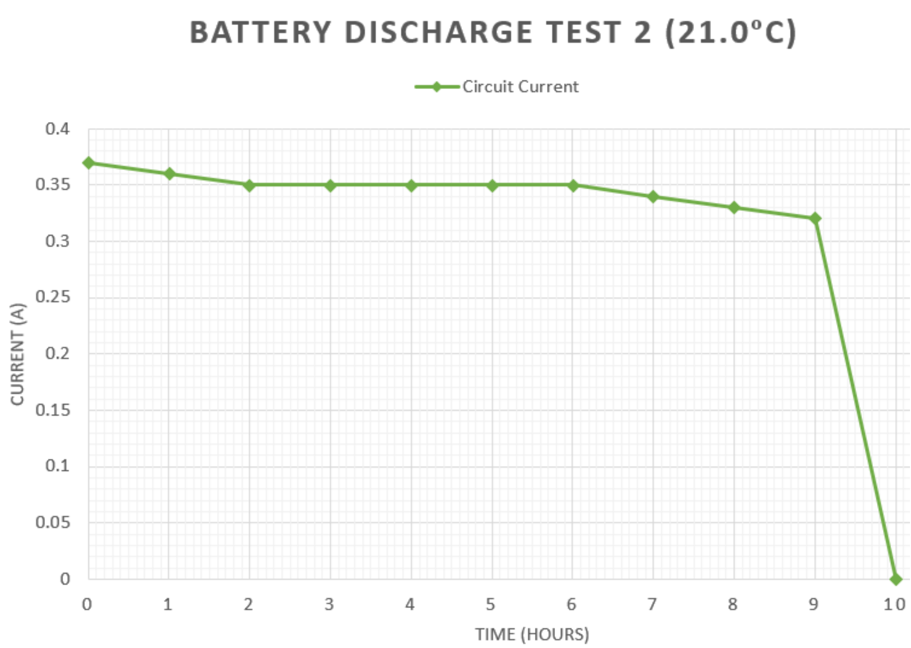

Figure 21: Discharge Current (21°C)

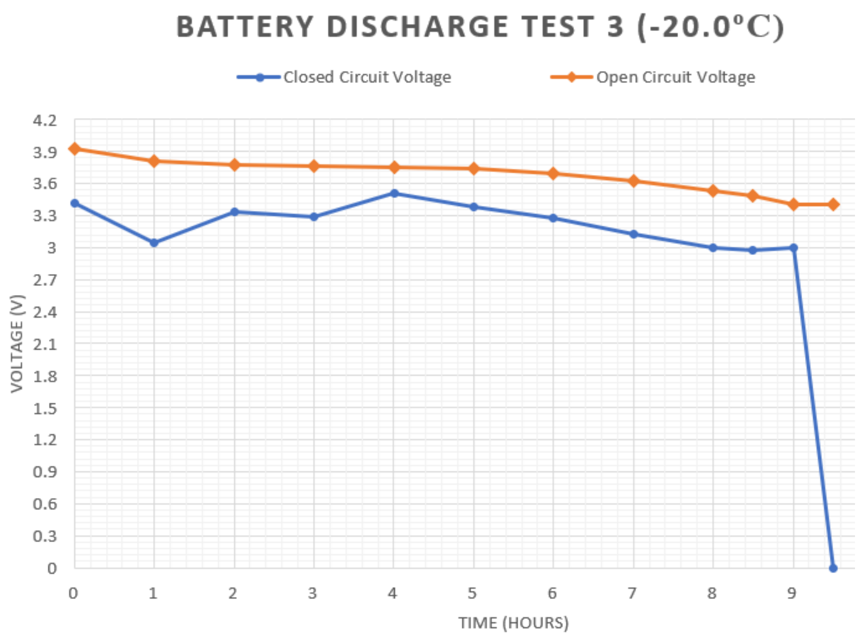

Figure 22: Discharge Open and Closed-Circuit Voltage (-20°C)

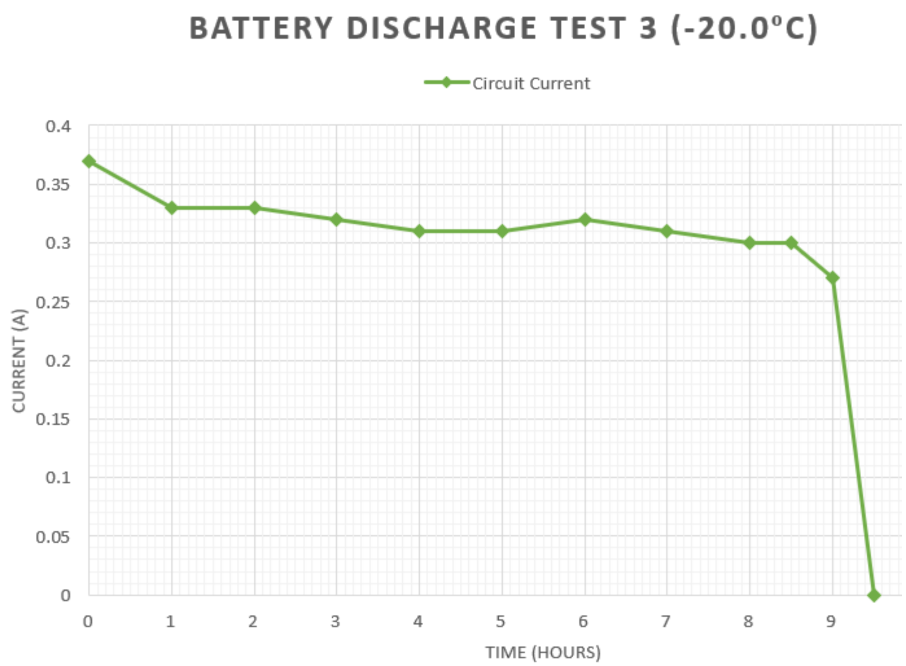

Figure 23: Discharge Current (-20°C)

Table 3: Vacuum Test Results

	Before	After
Thickness	19.9 mm	19.9 mm
Length	66.9 mm	66.9 mm
Width	37.0 mm	37.0 mm
Mass	94 ± 1 g	94 ± 1 g
Voltage	3.93 V	3.81 V

Post Analysis

The solar panel system, assuming best case scenario at worst case efficiencies, will be unable to sufficiently power the system during operation. As such, it is imperative that the system operate in cycles of monitoring and regeneration. In addition further work on the attenna and ground station receiver needs to be done in order to improve the packet reception. Current sensitivity is insufficient for data transfer of images.

Conclusions

The main systems components (Solar Panels, Battery supply, Sensory Instrumentation, Communication Modem, etc.) have been finalized for the RyeTubesat Project. Continued work on the Tubesat Project consists of simplifying control system components (i.e., the microcontroller) still need work done in order to accommodate the required functionality as well as simplifying the system design (i.e., a single microcontroller for all functionality as opposed to several microcontrollers of small memory for simple tasks). Furthermore, in accordance with aerospace standards, weight and power studies would need to be performed in regards to the addition of a Triple Modular Redundancy System on the tubesat. Should such a system be deemed insufficient, alternative methods of fault handling would need to be considered. However, as of today,the main priority of the RyeTubeSat project should be finalizing the control hardware and firmware for the tube sat functionalities. System fault handling can be considered later due to the short operating time of the project in orbit.

References

[1] (For a journal paper) A. Author and B. Arthur, “The title of a reference paper,” *Journal of Excitement*, vol. 1, no. 1, pp. 123-234, Jul. 1999.

[2] R. Lucky, “Automatic equalization for digital communication,” *Bell Syst. Tech. J.*, vol. 44, no. 4, pp. 547–588, Apr. 1965.

[3] (For a book) H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.
[4] (For a conference paper) G. Juette and L. Zeffanella, "Radio noise currents," *IEEE Summer power Meeting*, Dallas, TX, June 22–27, 1990.
[5] (Thesis style) J. Williams, "Narrow-band analyzer," Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.
[6] (Patent style) J. Wilkinson, "Nonlinear resonant circuit devices," U.S. Patent 3 624 12, July 16, 1990.
[7] (Standard style) *Letter Symbols for Quantities*, ANSI Standard Y10.5-1968.
[8] (Report style) E. Reber, R. Michell, and C. Carter, "Oxygen absorption in the Earth's atmosphere," Aerospace Corp., Los Angeles, CA, Tech. Rep. TR-0200, Nov. 1988.
[9] (Handbook style) *Transmission Systems for Communications*, 3rd ed., Western Electric Co., Winston-Salem, NC, 1985, pp. 44–60.
[10] (Basic Book/Monograph Online Sources) J. Author. (year, month, day). Title (edition) [Type of medium]. Volume(issue). Available: http://www.(URL)
[11] J. Jones. (1991, May 10). Networks (2nd ed.) [Online]. Available: http://www.atm.com