

ECE 492
Computer Engineering Design Project
Group 3

Satellite Ground Station

Cube Satellite Ground Station Control Software

Jacob Ortt
jortt@ualberta.ca

Bryson Peeters
bpeeters@ualberta.ca

Ranek Kiil
ranek.kiil@ualberta.ca

Andrew Keller
awkeller@ualberta.ca

Summary: Ground station to track cube satellite pass across the sky and communicate with it.

Abstract

Over the course of the semester we were able to build a ground station that will communicate with the AlbertaSat cube satellite once it launches. While some of the hardware for the ground station is not the final equipment we hope to use (in particular, we are seeking a better antenna, radio, and TNC) we were able to construct a prototype with similar devices borrowed from different sources. The two major goals of the project, tracking and communications, were both accomplished. For tracking, software was written that can predict where a satellite will be and can accurately point an antenna at this location in the sky, and prototype hardware was built. Compatibility between the hardware and software was tested, and the hardware was calibrated to ensure accurate operation. We confirmed the hardware functionality by tracking and listening to other small satellites which have already launched. For communications, we were able to reliably send packets of information between two amateur radios using the same Cubesat Space Protocol that will be used with the satellite. Once the specific TNC hardware for the satellite and ground station is available, the input and output of our simulation code can be directed through this equipment instead. All that is left is to further define the commands that will be communicated with the satellite, in cooperation with teams implementing the satellite firmware itself.

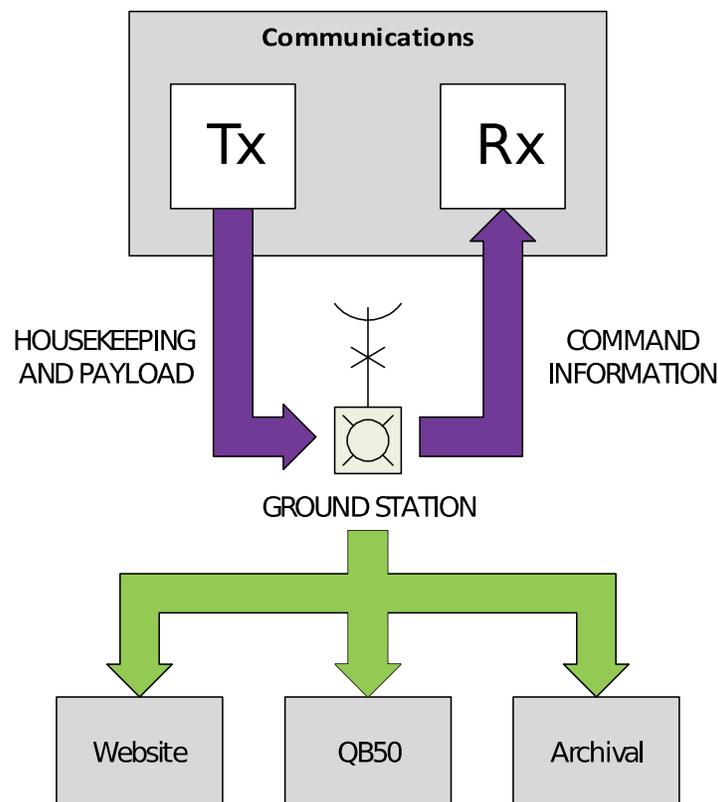
Table of Contents

Abstract.....	2	Software Design.....	15
Table of Contents.....	2	Test Plan	18
Functional requirements.....	3	Testing Orbital Tracking Software:.....	18
Design and description of operation	3	Testing Communications Software:.....	18
Description of Operation.....	4	Testing Ability to Receive Signals:.....	18
Bill of Materials	5	Results of Experiments and Characterization.....	19
Radio Hardware.....	5	Safety	19
Demo Hardware.....	6	Environmental Impact.....	19
Sources of Reusable Design Units	7	Sustainability.....	19
Datasheet	7	References	20
Background Reading.....	7	Appendices	21
Hardware Design.....	9	Quick Start Manual.....	21
Ground Station Overview	9	Future Work	21
Antenna Specifications.....	9	Certifications.....	21
Rotor Specifications	10	Hardware Documentation.....	23
Calibration of Antenna.....	12	Source Code.....	24
Downlink Telemetry Budget.....	14		

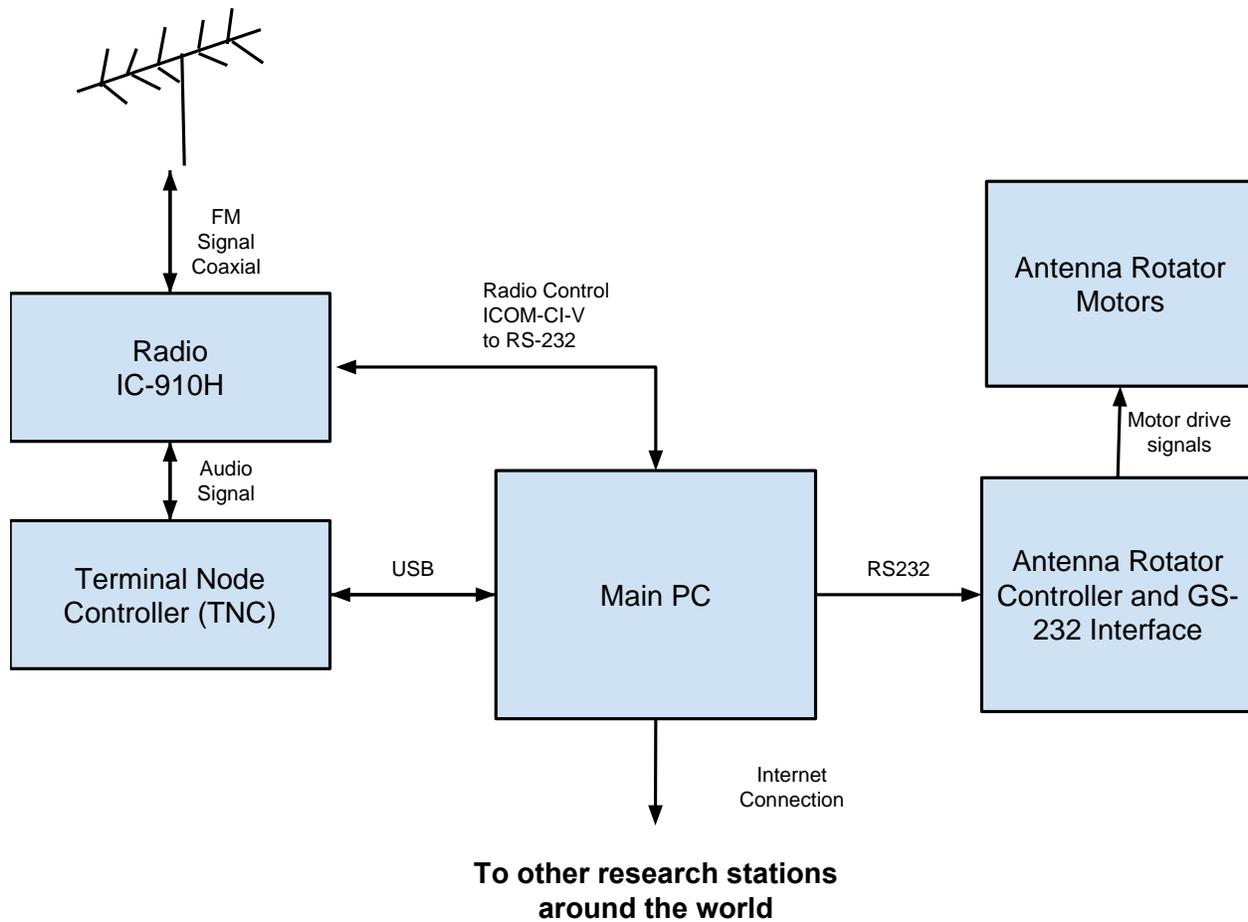
Functional requirements

1. Operate radio modem, including protocols (CSP) for communicating with satellite.
 - a. Functional, though using TNC built into Kenwood TH-D72 radios, rather than final hardware.
2. Command satellite, including the transmission of queued commands and the requesting of data downloads.
 - a. Not implemented, as satellite command set is not yet well defined.
3. Retrieve satellite trajectory data from the internet and use to calculate azimuth and elevation relative to the University of Alberta North Campus.
 - a. Fully functional.
4. Interface and command a 2-axis rotor that steers a directional antenna to track satellites.
 - a. Fully functional.
5. Archive downlinked scientific payload, and distribute over internet.
 - a. Not implemented, as satellite payload format is not yet well defined.

Design and description of operation



Dataflow diagram provided by AlbertaSat[3]. Purple indicates radio signals used to communicate with the satellite, while green will be standard TCP/IP over the internet.



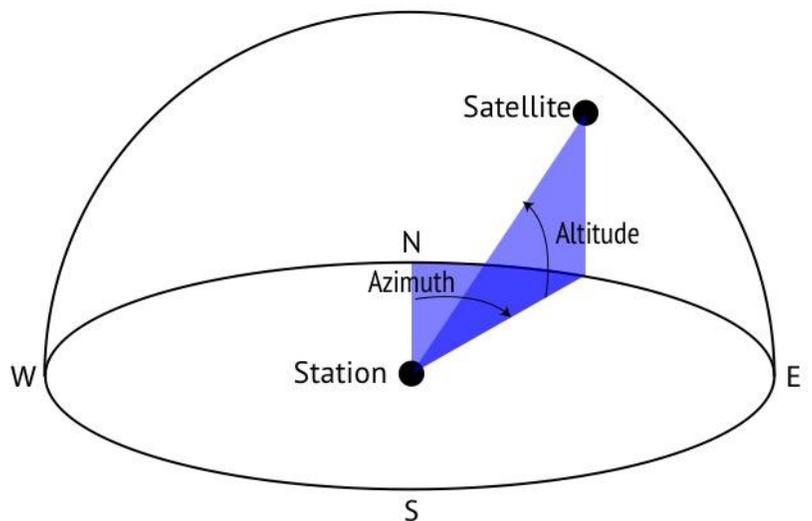
Block diagram of final hardware components which are expected to make up the permanent ground station.

Description of Operation

Using two-line element set data from NORAD, the ground station software models the satellite's orbit, enabling it to compute the satellite's topocentric azimuth and elevation relative to the ground station's physical location at the University of Alberta (53.528, -113.527). This data will be sent through the rotator controller, which has been calibrated to control a 2-axis rotor to point a directional antenna at the satellite as it passes overhead.

When a known satellite is overhead, the ground station starts a communications process, which establishes a connection to the satellite. This connection then enables the ground station to send commands to the satellite, including commands to request downlink of the scientific data.

After basic operation of the ground station has been tested, opportunities will be explored for how data can be better shared and presented with the public. Visualization or scientific analysis may be possible extensions if further



details on data schema and format are confirmed soon.

Bill of Materials

The main hardware components are an antenna, radio, and two motors in a rotator assembly for re-positioning the antenna. A rotator controller is also used to connect between low-current digital commands to high-current motor control. A standard PC is used to coordinate the operation of the rotator and radio, pulling updated satellite trajectories from the internet and pushing retrieved data out.

The data rate that will be coming in from the radio is still unknown to us because the design of the communications system from the side of the satellite is still incomplete. We believe that whatever the data rate the satellite is sending out our computer will be able convert and store it easily, since the ground station computer will be an off-the-shelf system not limited by the same power and size requirements as the satellite. The data from the satellite will likely be received as audio (usually several hundred kbps) or as standard serial (typically several thousand kbps), in either case these data rates will be negligible for a standard off-the-shelf system.

Additional components used, including the second radio and an Arduino, were used to prototype a simple satellite for demonstration purposes. Once the actual satellite hardware has been clearly specified, there will no longer be a long-term need for these parts.

Radio Hardware

Parts have been proposed and are being provided by AlbertaSat [1].

Component	Description	Specification	Cost in CAD	Order Status	Supplier	Datasheet
Power/SWR Meter Daiwa CN-103LN	Power and SWR meter for antenna testing	Frequency Range: 140 - 525 MHz	\$110	Received	http://radioworld.ca/103l-p-2473.html	http://www.daiwa-industry.co.jp/radio/hame/products/manual_cn101.pdf
Radio IC-910H	PC Controllable radio.	Frequency coverage: 144 - 148 MHz, 430 - 450 MHz, 1240 - 1300 MHz	Used - Hard to get.	Not Ordered	Discontinued	http://www.tel.uva.es/personales/genso/docs/IC-910H_Manual.pdf
Radio Replacement TS-2000	Radio featuring dual DSP units and RS-232 control support.	HF to 1.2 GHz and 144 - 440 MHz	\$1700	Not Ordered	http://www.hamradio.com/detail.cfm?pid=H0-005387	http://www.kenwoodusa.com/UserFiles/File/UnitedStates/Communications/AMA/Manuals/TS-2000-Owner-Manual.PDF
Power Supply FP-1030A	Antenna power supply	provides 13.8 VDC at 25 amps	\$310	Not Ordered	http://radioworld.ca/1030a-p-2137.html	http://radiomanual.info/schemi/ACC_power_supply/Yaesu_FP-1030A_user.pdf
Radio Control Cable CT-17	Allows the connection of ICOM radios to the RS-232 port on a PC	ICOM to RS-232 converter	\$195	Not Ordered	http://radioworld.ca/ct-17-p-1628.html	http://www.tel.uva.es/personales/genso/docs/ICOM_CT-17_Manual.pdf
TNC Gomspace	Terminal Node	USB interface and KISS-TNC	\$1000	Not Ordered	http://gomspace.com/index.php?	http://gomspace.com/documents/GS-DS-

TNC	Controller with USB interface and is KISS-TNC compatible	protocol compatible			p=products-tnc1	TNC-4.0.pdf
SPID RAS Rotor	Complete Azimuth and Elevation control solution	Rotator has a 450 degree rotation range and elevation has 180 degree rotation.	\$2,999	Not Ordered	http://www.yaesu.com/indexVS.cfm?cmd=DisplayProducts&ProdCatID=104&encProdID=79A89CEC477AA3B819EE02831F3FD5B8&DivisionID=65&isArchived=0	http://www.yaesu.com/indexVS.cfm?cmd=DisplayProducts&ProdCatID=104&encProdID=79A89CEC477AA3B819EE02831F3FD5B8&DivisionID=65&isArchived=0
Rotor Controller MD-01	Digital controller for Rotor	Communication with PC over USB	\$876	Not Ordered	http://www.hamradio.com/detail.cfm?pid=H0-002778	http://polysat.calpoly.edu/download/earthstation/documents/gs-232b.pdf
UHF Yagi Antenna 436CP42	Yagi antenna for satellite communications	Frequency: 430 - 438 MHz	\$456	Not Ordered	http://www.hamradio.com/detail.cfm?pid=H0-000725#results	http://www.ecse.rpi.edu/courses/CStudio/ham_radio_docs/436CP42UG%20manual.pdf
Mount Hardware for Ant.			\$700	Received		
Polarity Switch PS-70CM	Polarity switcher for reversing the polarity direction on the Yagi antenna	Frequency: 300 - 500 MHz	\$213.6	Not Ordered	http://www.hamradio.com/detail.cfm?pid=H0-005457#results	http://static.dxengineering.com/global/images/instructions/msq-ps70cm.pdf
Total Estimate			\$8559.60			

Demo Hardware

Component	Description	Specification	Cost in CAD	Order Status	Supplier	Datasheet
Arduino Uno	microcontroller board	16 MHz ATmega328, 2 KB RAM	\$25.00	Not Ordered		
Pair of Handheld Amateur Radios	Handheld Radios with built in TNC. Can be used in place of the Gomspace TNC and radio for testing	144MHz/430 MHz FM bands. AX.25 TNC with KISS support.	\$0 (Provided by Collin Cupido)	Received	http://www.keenwood.ca/Communications/AmateurRadio/Portables/TH-D72A	http://www.keenwood.ca/Communications/AmateurRadio/Portables/TH-D72A

Total Estimate**\$25.00**

Sources of Reusable Design Units

We plan to implement the majority of the ground station software in Python running on Linux. Ubuntu is the target operating system, though and effort has been made to stick to a cross-platform design when possible. Python provides a vast array of general computing modules and the open source community surrounding Python is quite diverse. For scientific computing, the SciPy and NumPy libraries provide most needed functionality to implement vector math, solve optimization problems, and more. The Skyfield Python library performs the calculations necessary to return altitude and azimuth values for the satellites, given a time and location. The most important libraries used are listed below; a full set of requirements is available in the requirements.txt file of the source code.

- Numpy: <http://www.numpy.org/>
- PyEphem: <http://rhodesmill.org/pyephem/>
- AstroPy: <http://www.astropy.org/>
- PySerial: <http://pyserial.sourceforge.net/>

Cubesat Space Protocol (CSP) (<https://github.com/GomSpace/libcsp>) has been used for packet communications between the Ground Station and the satellite. CSP provides basic socket interfacing similar to IP and TCP. The library is written in C but a partially complete set of Python bindings are included that we then expanded upon to interface with a set of radios via the KISS interface. The source size is approximately 600 kB and the compiled library is 184 kB.

Datasheet

The PC that will be controlling the ground station will be connected to the antenna motors using a single USB port. It will communicate with the Terminal Node controller using a single USB port. All other communication with other researchers will be done over the internet via ethernet.

Signal Name	Number of Wires in Bus	Location of Signal
Satellite Data Received	2 (USB)	TNC to PC
Satellite Data Transmitted	2 (USB)	PC to TNC
Antenna Radio Signal	1 (Coaxial)	Antenna to TNC (analog)
Antenna Position Data	2	PC to Antenna Rotor Controller
Radio Control	2	PC to Radio

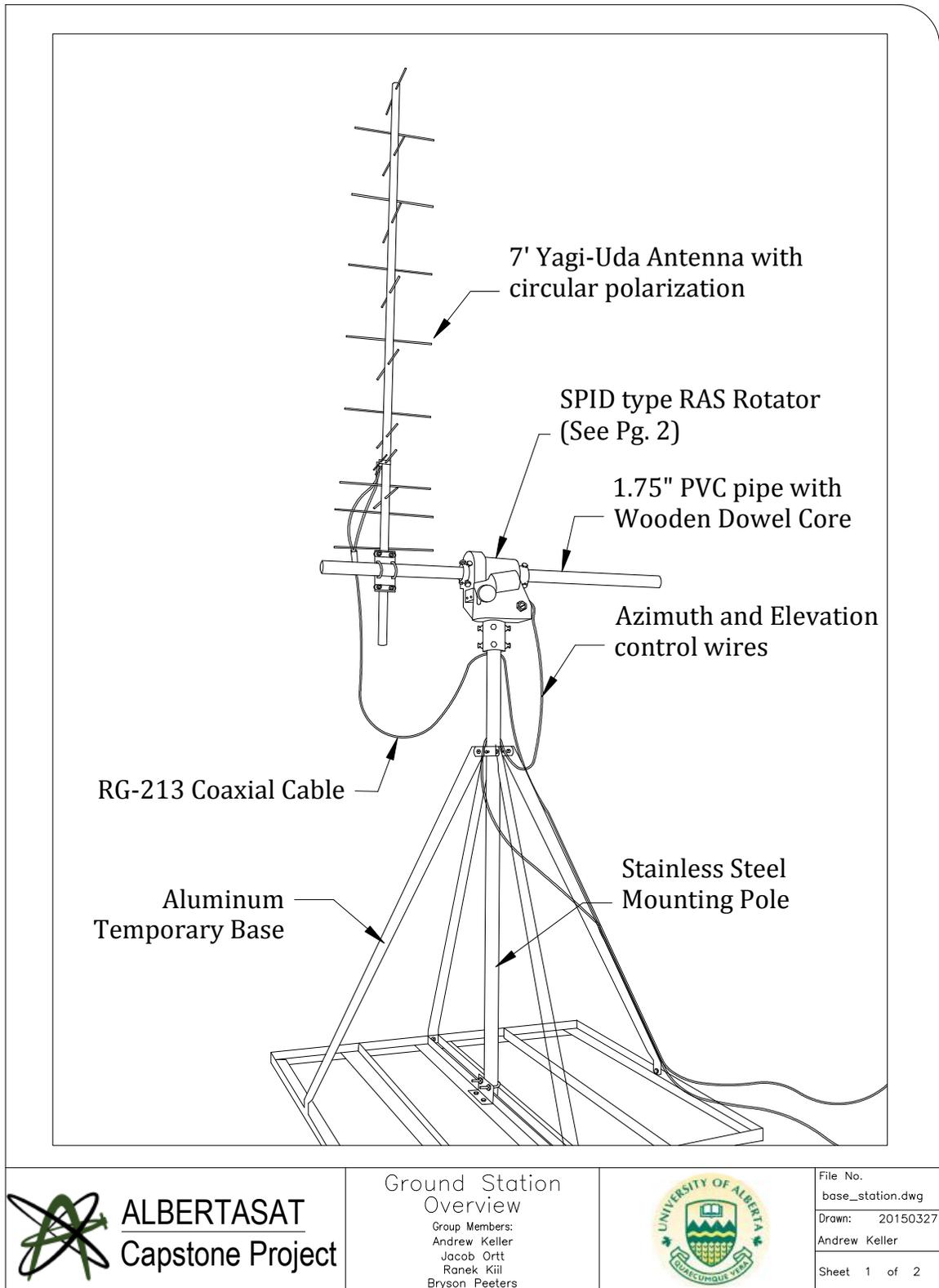
Background Reading

- Good resource on polarity switching: <http://sv1bsx.50webs.com/antenna-pol/polarization.html>
- Textbook on satellite design with good chapter on ground stations : *Space Mission Analysis and Design*, 3rd ed., Wiley J. Larson and James R Wetzr, 1999
 - TH-D72 Commands: <http://www.worldwidedx.com/threads/kenwood-th-d72-tnc-commands.92416/>
 - Has information on how to operate a Kenwood TNC. It's not the same TNC as ours but it works the same it seems: <http://www.repeater-builder.com/kenwood/pdfs/tm-d700a-aprs-packet-manual.pdf>
 - This thread provides an exact example of how to send a packet between two D72 radios: <http://forums.radioreference.com/amateur-radio-data-transmissions/285449-can-someone-explain-how-i-send-packets-using-tnc.html>
 - TCP/IP over AX.25: <http://www.tldp.org/HOWTO/AX25-HOWTO/x94.html>

- The creator of the Skyfield library talking about good Python API design:
<https://www.youtube.com/watch?v=FVEEndIwOSA>

Hardware Design

Ground Station Overview



Antenna Specifications

Due to the modifications made to the existing antenna the following specifications are for a commercially available antenna that has similar characteristics to the one we used:

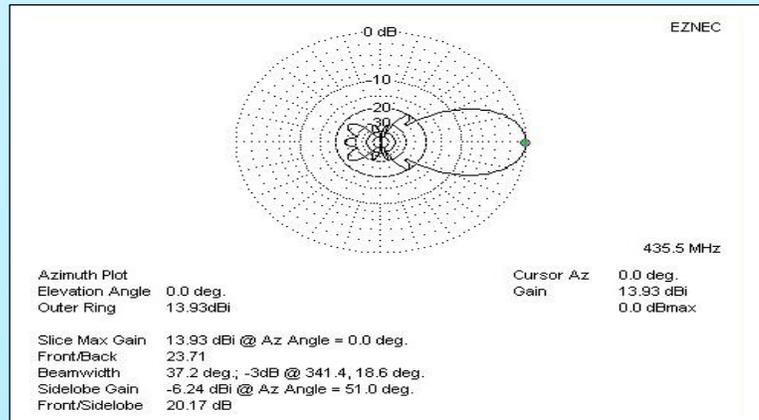
70 CM Circular Polarized Satellite Yagi

Model No. 70CM-8ELSat

Price-\$249.00



Boom Length-67"
 Boom dia. -1 1/4" dia.
 No. of elements-8 vertical 8 horizontal
 Freq. range-430-438 MHZ
 Feed Point Impedance-50 Ohms
 Feed Connector- type N
 Power Handling-400 watts
 Forward Gain-11.40 dbdc
 Front to Back ratio-53 db
 Beamwidth Azimuth @ 3db- 40.0 degrees
 Beamwidth Elevation@ 3db-43.6 degrees
 Mast mounting dia.-1 1/4" to 2" dia.

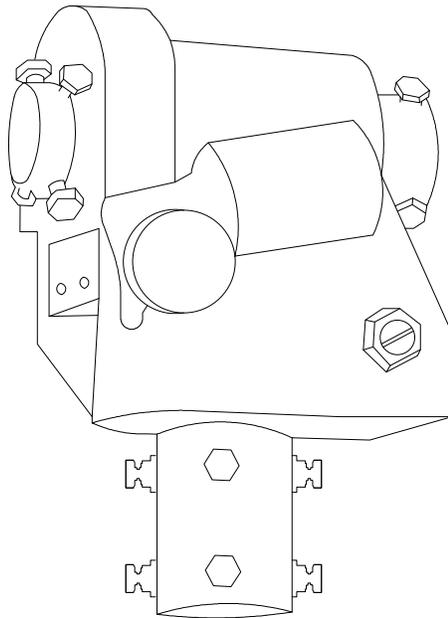


Designed for satellite operation and has circular polarization (RHCP) with both sets of elements optimized in satellite band. Unit is supplied with phasing lines and T connector and typical vswr in satellite band with phasing lines connected, 1.2 to 1. All connections use high quality type N connectors. Constructed of the finest materials, 6061 corrosion resistant aluminum. Elements are 3/16" dia. except for the driven element which is 1/4" dia. (solid rods). Uses superior T match system and coaxial balun for 50 ohm balanced load. Unit is CAD designed for the mechanical structure and computer optimized for electrical characteristics. Antennas are constructed with the utmost precision and all hardware is stainless steel. Designed to last indefinitely while giving superior performance.

Specifications Obtained from:
<http://gulfalphaantennas.com/>

Rotator Specifications

SPID RAS Rotator



Notes:

Required Voltage: 24-12 VDC

360 Degree Rotating Time: 60-120 sec.

Maximum Mast Diameter: 50mm

Maximum Mounting Pipe Diameter: 66mm

Turning Torque: 158-366Nm

Brake Torque: 1582Nm

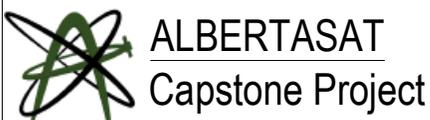
Max. Vertical Load: 250kg

Precision: 1°

Rotating angle azimuth: 360°

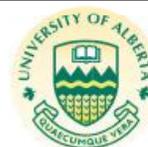
Rotating angle elevation: 180°

Weight 13kg



Ground Station
Rotator Specs.

Group Members:
Andrew Keller
Jacob Ortt
Ranek Kill
Bryson Peeters



File No.

base_station.dwg

Drawn: 20150327

Andrew Keller

Sheet 2 of 2

Calibration of Antenna

One of the major challenges to making our antenna functional in the real world is to calibrate it so that it reliably points in the right direction. Even if we have designed software that predicts the satellite's path exactly and turn the rotator to track the satellite with a high precision, the antenna will not be pointed in the correct direction if it is not calibrated correctly. This means we need to set where north is so that when instructed to point at 0° , it points precisely north. Some compensation for elevation may also be required if the surface the rotator is on is not level. This lab section will discuss three different approaches to calibration: the basic approach, single point of reference, and triple point of reference.

Basic Approach:

The simplest approach to calibrating the antenna is to use a level and a compass. The rotator sits on a vertical pole that can be adjusted. Using a level this pole can be adjusted to be vertical and the rotator can be adjusted to be level with the pole. The rotator can be set to be 90 degrees (straight up) on the controller and then the pole can be adjusted to also be perfectly vertical according to the level. If all of these components are level according to this test then the altitude put out by the controller should be accurate. If 90 degrees is correct on the controller then we assume that all other angles will also be correct.

The direction that the rotator is pointed can then be calibrated using a compass. The controller altitude is then set to 0 degrees to point at the horizon and the controller azimuth is set to be 0 degrees to point north. A compass is then used to find true north and the rotator is adjusted to face in that direction. An important point to note here is that magnetic north in Edmonton is not very close to true north. The magnetic declination in Edmonton is +14.8 degrees. This means that magnetic north is 14.8 degrees east of true north. To point to rotator at true north it should be pointed 14.8 degrees west of the compass point.

We used this approach at first to get our antenna in a generally correct direction. The major problem with this approach however is that compasses are not a very accurate tool when standing on top of a large metal building. We found that just by moving a couple of meters from where we were the needle point could move by up to 20 degrees. This is because of the warp in the earth's magnetic field caused by all of the metal around. In fact the antenna itself is metallic so if the compass is moved close to the antenna for easy alignment it will move into a completely incorrect direction. On top of this, the levelling approach was also fairly imprecise. We found it to be quite difficult to level to pole using just the alignment screws holding it in place. It can be mostly leveled but was still a couple of degrees off at the end. For these reasons we decided to use more accurate methods.

Single Point-Of-Reference:

The next method of calibration is to point the antenna at a known object in the sky and calibrate it that way. This known object could be the moon, the sun, or a star. The main procedure is to point the antenna at this celestial object and calculate how far off the controller's azimuth and altitude values are from the true values of that object. This offset can then be saved and used later to calculate what controller values should be used to point the antenna in a certain true direction. The coordinate system and formulas are shown below.

$$\begin{array}{lll} \text{True Altitude} = A & \text{Controller Altitude} = a & \text{Altitude Offset} = \alpha \\ \text{True Azimuth} = Z & \text{Controller Azimuth} = z & \text{Azimuth Offset} = \beta \end{array}$$

$$(1) A = a + \alpha$$

$$(2) Z = z + \beta$$

The first step is to choose a celestial body and find its true altitude and azimuth (A and Z). We used the moon for our calibration because it is easy to find in the sky and you can also line the antenna up to in by line of sight unlike the sun. We used an online resource to find its true azimuth and altitude at the specific time (<http://www.dailymoonposition.com/>).

The second step is to align the antenna with the celestial body and read the controller's values of the azimuth and altitude (a and z). Using formulas (1) and (2) above the offset values (α and β) can be calculated.

Finally these offset values can be incorporated into the prediction software to point the controller in the correct direction. For instance if the antenna should be set to an elevation of 45 degrees, then the controller should be set to an elevation of $(45 - \alpha)$ degrees instead in order to achieve the true elevation.

This calibration technique was used by us, however we did not need to incorporate these calibration offsets into the satellite prediction software. The controller itself allowed us to set these offsets making the process much faster. This calibration also allowed us to give what appeared to be accurate measurements of the sun and the moon (much more accurate than from the basic approach) so we were satisfied with it. That being said there are some inherent problems with this calibration approach. It does not take into account how level the rotator itself is with the horizon. It assumes that if the rotator is set to a certain altitude it will maintain that altitude no matter what azimuth it is pointed in. This unfortunately is not the case. If the rotator is crooked on the pole or the pole itself is crooked then the altitude offset will change based on the azimuth.

Triple Point-Of-Reference:

The triple point-of-reference calibration technique would accommodate for a crooked rotator by assuming that the controller's altitude measurements are relative to a sloped plane. This means that the altitude offset changes relative to the azimuth. The altitude offset (α) would no longer be a constant but instead be given by the following formula.

$$(3) \alpha = \alpha_0 + b \cdot \sin(Z - \Omega)$$

Which when added to formula (1) gives the following

$$(4) A = a + \alpha_0 + b \cdot \sin(Z - \Omega)$$

α_0 = average altitude offset

b = steepness of tilt

Ω = direction of tilt

To calculate these values three different calibration points would be needed. This could possibly be done at night using three different known stars or during the day by tracking the path of the sun or moon across the sky at different points in the day. From each of these calibration points the values of A, a, and Z would be recorded. By adding them into equation (4) three equations would be created each with three unknown constants. The system of equations would then have to be solved to find the constants α_0 , b, and Ω . This is theoretically the most precise way to calibrate the antenna. We never actually performed this calibration however because the beam width of our antenna is about 40 degrees. This calibration would only change our current values by about a degree so it would not be worth our time. It would also be very difficult to automate because the resultant system of equations does not have a simple solution.

Downlink Telemetry Budget

Satellite Ground Station		
Parameter:	Value:	Units:
<i>Spacecraft:</i>		
Spacecraft Transmitter Power Output:	0.5	watts
In dBW:	-3.0	dBW
In dBm:	27.0	dBm
Spacecraft Total Transmission Line Losses:	5.9	dB
Spacecraft Antenna Gain:	2.2	dB _i
Spacecraft EIRP:	-6.8	dBW
<i>Downlink Path:</i>		
Spacecraft Antenna Pointing Loss:	0.3	dB
S/C-to-Ground Antenna Polarization Loss:	4.1	dB
Path Loss:	148.4	dB
Atmospheric Loss:	1.1	dB
Ionospheric Loss:	0.4	dB
Rain Loss:	0.0	dB
Isotropic Signal Level at Ground Station:	-161.1	dBW
<i>Ground Station (EbNo Method):</i> ----- Eb/No Method -----		
Ground Station Antenna Pointing Loss:	0.1	dB
Ground Station Antenna Gain:	11.4	dB _i
Ground Station Total Transmission Line Losses:	1.6	dB
Ground Station Effective Noise Temperature:	714	K
Ground Station Figure of Merit (G/T):	-18.7	dB/K
G.S. Signal-to-Noise Power Density (S/No):	48.7	dBHz
System Desired Data Rate:	1200	bps
In dBHz:	30.8	dBHz
Telemetry System Eb/No for the Downlink:	17.9	dB
Demodulation Method Selected:	AFSK/FM	
Forward Error Correction Coding Used:	None	
System Allowed or Specified Bit-Error-Rate:	1.0E-04	
Demodulator Implementation Loss:	0	dB
Telemetry System Required Eb/No:	21	dB
Eb/No Threshold:	21	dB
System Link Margin:	-3.1	dB
<i>Ground Station Alternative Signal Analysis Method (SNR Computation):</i> ----- SNR Method -----		
Ground Station Antenna Pointing Loss:	0.1	dB
Ground Station Antenna Gain:	11.4	dB _i
Ground Station Total Transmission Line Losses:	1.6	dB
Ground Station Effective Noise Temperature:	714	K
Ground Station Figure of Merit (G/T):	-18.7	dB/K

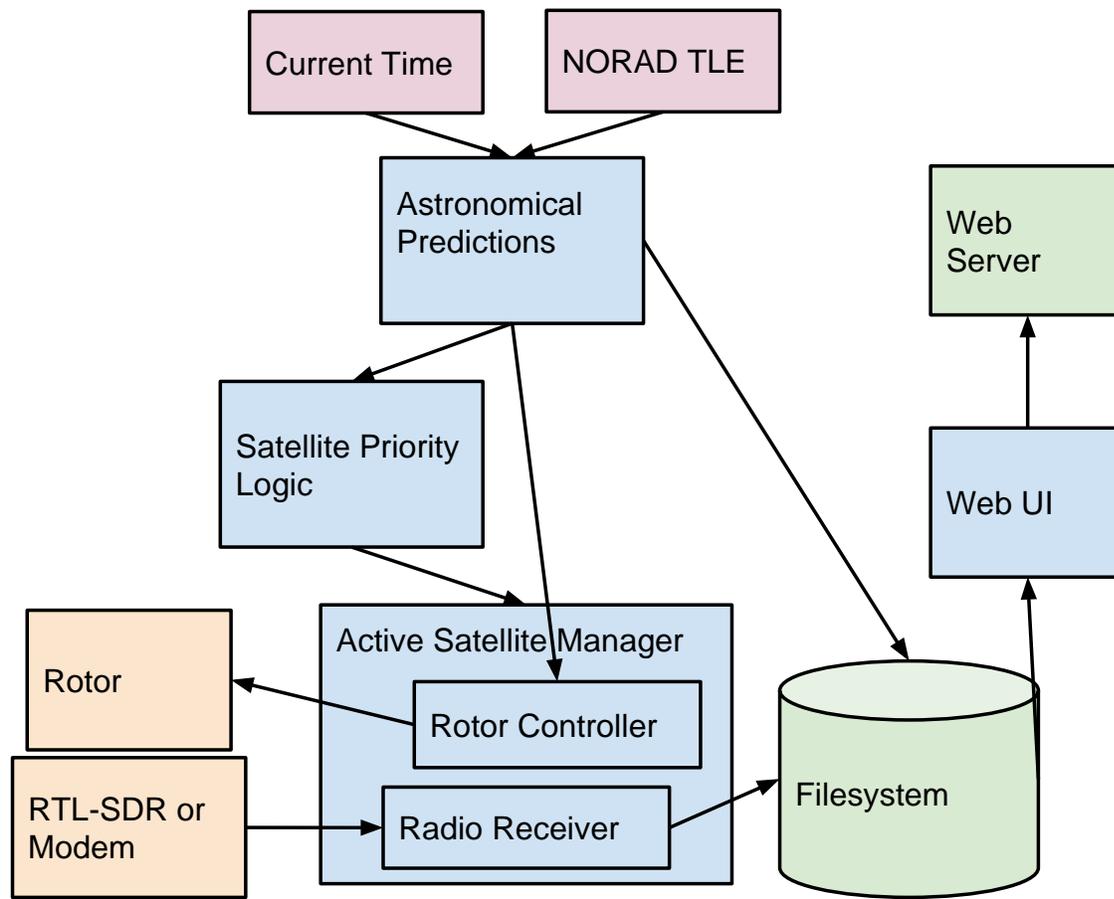
Signal Power at Ground Station LNA Input:	-151.4	dBW
Ground Station Receiver Bandwidth (B):	10,000	Hz
G.S. Receiver Noise Power ($P_n = kTB$)	-160.1	dBW
Signal-to-Noise Power Ratio at G.S. Rcvr:	8.7	dB
Analog or Digital System Required S/N:	16.6	dB
System Link Margin	-7.9	dB

Note: For a list of the values used for the above calculations see Hardware Documentation in the Appendix

As shown in the table above our downlink values concerning the proposed satellite range from -3.1dB to -7.9dB depending on which calculation method is used. In order for us to be able to receive a signal from the ground station we require a “System Link Margin” that is positive. The obtained negative values do not come as a surprise. Our antenna lacks the gain required to hear satellites that are ‘quieter’. We were, however, able to listen to ‘louder’ satellites. Satellites that are transmitting on the 70cm B.W. are typically pretty quiet so many of the satellites we attempted to listen too could not be heard. To solve this problem we could add a low noise amplifier (LNA) to the system or increase the gain of the Antenna.

Software Design

- Astronomical Predictions
 - Downloads latest two-line-element sets
 - Predicts current location of satellite
 - Predicts next satellite pass time
- Satellite Priority Logic
 - Given list of satellites to track and priorities of which to listen to, controls which satellite the rotor control software listens to and which radio receiver is active
- Rotor Controller
 - Given predicted satellite trajectory, controls rotor
- Radio receiver
 - Given satellite type, records received data
 - May also be involved with transmission
 - Stores received data packets or files in the file system



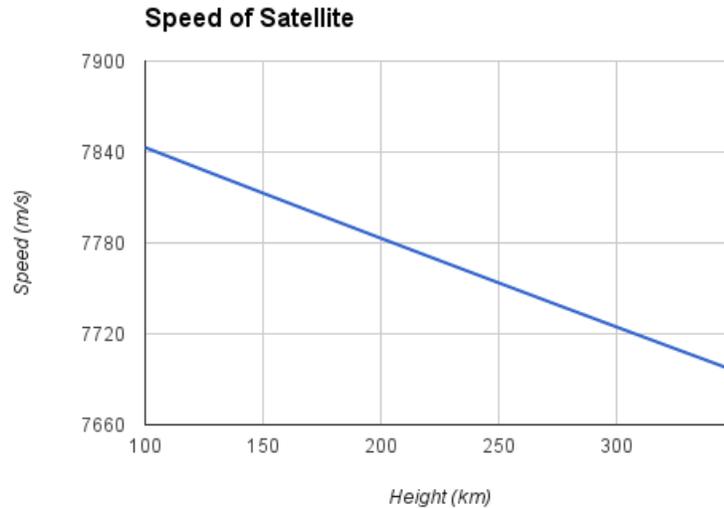
Doppler Shift

One of the major design challenges that has to be faced when designing a satellite ground station is dealing with the Doppler shift of the signal. The Doppler Effect is most commonly seen when listening to the sound of fast moving vehicles. If a car drives by you at a high speed, the sound it gives off will sound higher in pitch when it is coming towards you and lower in pitch when moving away. This same effect can be seen on the electromagnetic waves used for communication with satellites. Even though the satellite communications moved at the speed of light there is still a slight Doppler Shift.

The first thing that needs to be calculated in order to find the Doppler shift is the speed of the satellite. The speed of the satellite depends on the height of its orbit. Our satellite will begin its orbit at a height of about 350km and end by burning up in the atmosphere at about 90km. This means that the software will have to constantly update its value for the satellite speed based on its current height. The formulas and results for this calculation is shown below.

$$v_{\text{orbit}} = (\mathbf{r}_{\text{orbit}} * g_{\text{orbit}})^{\wedge} (1/2)$$

$$g_{\text{orbit}} = g (\mathbf{r}_{\text{earth}} / \mathbf{r}_{\text{orbit}})^{\wedge} 2$$



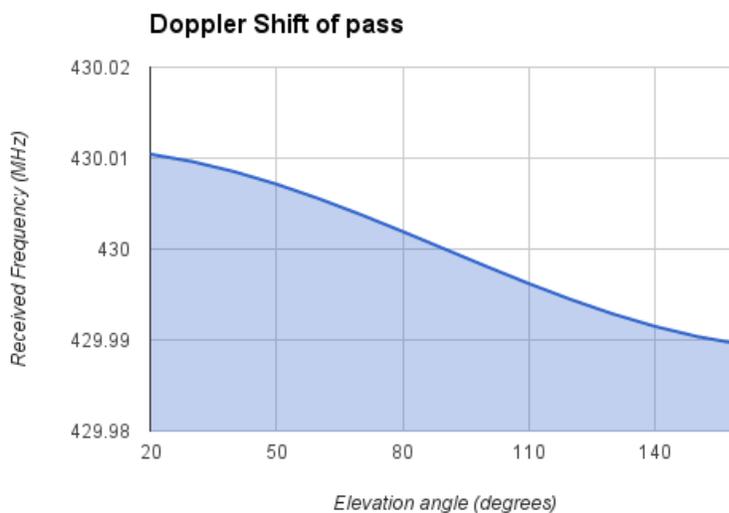
This shows that the speed of the satellite in its orbit will range from about 7700 m/s to about 7850 m/s during its lifetime. For the rest of the calculations we will assume that the satellite is currently orbiting at 250km with a speed of 7754m/s. This value only represents the satellite's perpendicular speed to the earth's surface. The important value will actually be the change in distance between the satellite and the observer. This value is given by the following formula:

$$\text{distance_change} = \text{speed} * \cos(\text{altitude})$$

Where altitude is the angle the satellite is located above the horizon. This change (v_s) can then be used in the following formula to find the Doppler shift of the signal

$$f_{\text{doppler}} = f_0 * (c / (c + v_s))$$

By adding these values into the equations the Doppler shift of the signal can be found at any point while it passes overhead. An example is given below of the frequency being received from a satellite that is transmitting at 430MHz flying directly overhead.



As shown in the diagram, the communication frequency will vary by about 20kHz during a pass. This will have to be taken into account in the final design of the communications software. Unfortunately we will not be able to implement this portion of the design yet because we do not yet have access to the radio that will be used for the final ground station. All of the software testing that we will do is using HAM radios so we will simply be tuning the radio by hand. It is also worth noting that this is essentially the worst case Doppler shift. The satellite will almost never be travelling directly overhead. If it is passing by to the side of the ground station then the Doppler shift will be lower.

Test Plan

The AlbertaSat satellite will not be launched for many months after our project is complete. For this reason we have to find ways to test the ground station without a working prototype of the in-flight module. It is likely that we will test components of our design like orbital tracking on known satellites like the international space station. It is also unknown at this moment when we will be able to receive the radio equipment for the ground station. It is likely that in order to test our communication code we will have to use different communications technology than what was intended in our design. One proposed way of doing this would be to connect the ground station and the in-flight module (which is in production) to a cheap short range radio like a Zigbee radio. An even simpler way would be to simply use an RS232 connection to the in-flight test board.

Testing Orbital Tracking Software:

The orbital tracking software can be tested using a compass, a protractor, a level, and the International Space Station. The ISS can be seen by the naked eye so if a pass is calculated by our software it can be tested at night without the need of the antenna and radio. The steps to completing this are below.

1. Calculate next night time pass of the ISS
2. Create a tube out of a roll of paper
3. Have one member of the team sit facing a direction that is part way through the pass (possibly an azimuth half way between rise and fall azimuth.
4. Have this member follow the path of the satellite with the paper tube until it is directly in front of him then stop
5. Have a second member check the altitude angle of the paper tube to see if it is close to the expected altitude that was calculated at that moment.

This test is not very accurate but would be a simple and fast way to see if our orbit calculator is giving reasonable calculations. For more accurate calculations we would need to have the rotator set up. We could then mount a camera to it and check that the ISS stays in the center of the picture as the rotator set-up follows the orbit.

Testing Communications Software:

The communications module was tested via a set of handheld amateur radios. The radios supported AX.25 over KISS which is the same interface that will be used with the TNC from Gomspace that we are waiting on. Commands were wrapped in Cubesat Space Protocol (CSP) packets and then transmitted and received via the radios using the CSP protocol layer.

Testing Ability to Receive Signals:

The QB50 group has already launched two test satellites that we can use to test our reception capabilities. When we have a radio ready these satellites will be extremely valuable in testing that we are correctly receiving QB50 protocol.

Results of Experiments and Characterization

Output values from the Skyfield-based code written for this project match with the output of other software tools, including the Gpredict software, and websites which offer satellite prediction.

Using the ground station setup, without a TNC, we recorded audio of several satellites as they passed. The most promising result to date was the recording we made of the KKS-1 satellite on April 7. A CW beacon was detected at 437.385 KHz and decoded to spell the repeating message, "2 8-52-11 MINAMI-SENJYU ARAKAWA-KU TOKYO-TO JAPAN". This is the address of the Tokyo Metropolitan College of Industrial Technology where the satellite originates.

Safety

Since the ground station will be mounted on the roof of a building on campus, installation and maintenance poses a safety risk. However, the ongoing operation is unlikely to pose a major risk to humans, as roofs are typically uninhabited.

The safety risk of installation will be mitigated by requiring that all members who venture onto the roof take part in a roof training course. The assumption is that other people on the roofs of university buildings will also have received this training, and therefore will exercise caution when working on the roof. Every member of our group took the Roof Access Orientation course and received our roof access cards.



Environmental Impact

Due to the high altitude position of the antenna and rotator, there is a potential risk for birds which might fly close to the roof of the building the antenna is mounted on. However, we feel the low speed of rotation of the rotator is unlikely to cause permanent harm to any animals.

Sustainability

Power consumption estimates

Constantly powered parts:
Computer: 15 W

Controller: 0.20 A @ 15.3 V = 3.1 W

Moving motor:

Motor: 3.0 A @ 21.6 V = 64.8 W

Active Communications:

Radio: 2 W

A lower power computer (like a Raspberry Pi, which draws under 2 watts) would be sufficient for these purposes if power saving was critical.

Average:

18.1 W constantly = 159 kWh annual

Estimated 1% motor usage = 5.68 kWh annually

Estimated 10% active radio usage = 1.75 kWh annually

Total estimated annual usage: 166 kWh

Based on 166 kWh consumption per year, burning coal to power the plant would release approximately 332 kWh.

Based on an average daily consumption of 455Wh for our device and an average daily generation of 440Wh / m² of solar panels, we would require 1.03m² of solar panels to power our device.

References

- [1] Collin Cupido (ccupido@ualberta.ca), “Fwd: Ground station,” E-mail to Bryson Peeters (bpeeters@ualberta.ca) 17 January 2015.
- [2] *Space Mission Analysis and Design*, 3rd ed., Wiley J. Larson and James R Wertz, 1999,
- [3] AlbertaSat, *Data Flow Diagram* [PDF], Available: <http://albertasat-software.wikispaces.com/file/view/Data+Flow+Diagram.pdf>
- [4] B. Rhodes. (2013). “Skyfield and 15 Years of Bad APIs”. *PyCon Canada*. Available: <https://www.youtube.com/watch?v=FVEEndIwOSA>
- [5] B. Rhodes. (2015). “Skyfield: Elegant Astronomy for Python”. Available: <http://rhodesmill.org/skyfield/>
- [6] *LPC 1769/68/67/66/65/64/63 Product Data Sheet* Rev. 9.5, NXP Semiconductors, 24 June 2014. Available: http://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.pdf
- [7] Chris Liecht. “Welcome to PySerial’s Documentation”, Available: <http://pyserial.sourceforge.net/>
- [8] Gomspace. "The Cubesat Space Protocol", Available: <https://github.com/GomSpace/libcsp>

Appendices

Quick Start Manual

Dependencies:

- Python 2
- Pip
- Virtualenv
- HamLib

In a command prompt execute the following:

```
git clone https://github.com/ranek/albertasat-groundstation.git
cd albertasat-groundstation
virtualenv venv
source venv/bin/activate
pip install -r requirements.txt
rotctld -m 901 -r /dev/ttyUSB0 -s 9600 -t 4533
```

```
python demo.py
```

Future Work

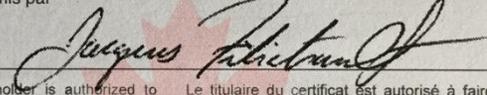
1. Coordinate with satellite coms teams and implement all commands to be communicated with the satellite.
2. Ensure interoperability with communications code on satellite, including testing and/or development of satellite software libraries.
3. Share data with researchers all over the world via internet, including push to QB50 team.
4. Provide an interface for visualization, analysis, and archival of data.

Eventually, Priority will be based on the age of the data and number of previous download attempts, in an attempt to download the oldest and most frequently failed datasets first. Other priority schemes may be made available so that data can be downlinked based on newest data first or based on the interest of the data.

Certifications

In order to operate the amateur radios, two group members received their certificate of proficiency in amateur radio.

 Industry Canada	Industrie Canada	CERTIFICATE OF PROFICIENCY IN AMATEUR RADIO	CERTIFICAT DE COMPÉTENCE EN RADIO AMATEUR
Certificate Holder - Titulaire du certificat			
Ranek Kiil			
Qualifications - Compétences			
Basic with Honours			
Call Signs - Indicateurs d'appel			
VA6RAK			
FOLD / PLIER			
Issued by - Émis par			
			
The certificate holder is authorized to operate amateur radio apparatus in accordance with the regulations made pursuant to the <i>Radiocommunications Act</i> .		Le titulaire du certificat est autorisé à faire fonctionner des appareils radio amateur conformément aux règlements établis en vertu de la <i>Loi sur la radiocommunication</i> .	
Number - Numéro	Issue Date - Date d'émission		
20150000529	27 March 2015		
IC 2755 (2010/11)			

 Industry Canada	Industrie Canada	CERTIFICATE OF PROFICIENCY IN AMATEUR RADIO	CERTIFICAT DE COMPÉTENCE EN RADIO AMATEUR
Certificate Holder - Titulaire du certificat			
Bryson Peeters			
Qualifications - Compétences			
Basic			
Call Signs - Indicateurs d'appel			
VE6BPZ			
FOLD / PLIER			
Issued by - Émis par			
			
The certificate holder is authorized to operate amateur radio apparatus in accordance with the regulations made pursuant to the <i>Radiocommunications Act</i> .		Le titulaire du certificat est autorisé à faire fonctionner des appareils radio amateur conformément aux règlements établis en vertu de la <i>Loi sur la radiocommunication</i> .	
Number - Numéro	Issue Date - Date d'émission		
20150000414	03 March 2015		
IC 2755 (2010/11)			

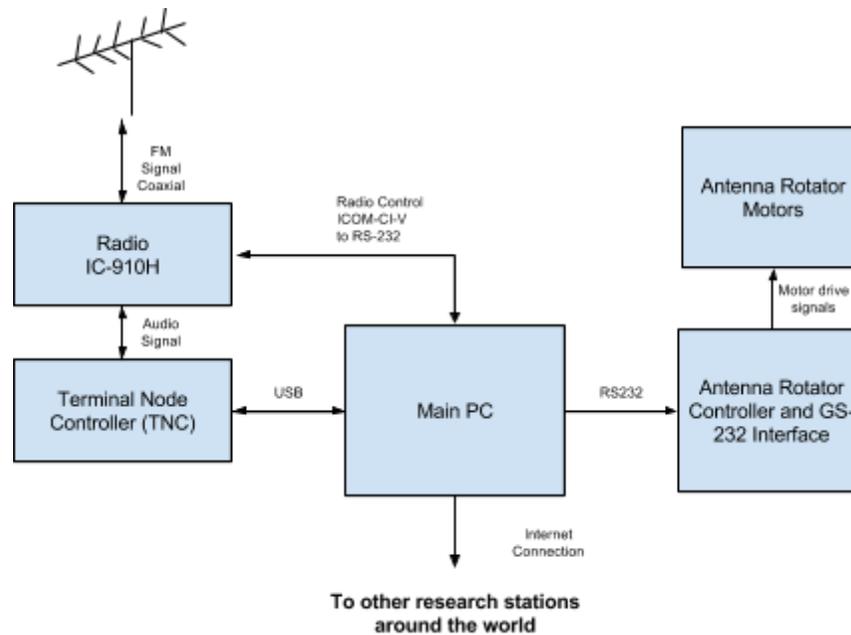
Hardware Documentation

For Hardware manuals for the MD-01 Controller and RAS rotator use the following link:

<http://alfaradio.ca/downloads/manuals/>

Downlink Budget Variables

Orbit		
Height of Apogee	400km	
Height of Perigee	400km	
Elevation Angle	10°	
Frequency		
Downlink Frequency	437.500MHz	
Transmitter		
Transmitter Power	0.5W min (2.0W max)	
Receiver		
Line A length	0.3m	
Line B length	7.5m	
Line C length	0.3m	
Cable Type	Belden 8267 (RG-213)	
Cable loss/meter	0.58dB	
Number of Connectors	4	
Estimated Noise Level:	-130dBm	
Bandwidth	15.0kHz	
“Sky” Temperature	483K	
Ground Station Feedline Temperature	290K	
Communications Receiver Front End Temperature	290K	
Antenna		
Type	Yagi-Uda	
Boom Length	1.371m	
Elements	8	
Max Gain	11.4dBiC	
Beamwidth	40°	
Pointing Losses		
Estimated Pointing Error of Ground	5°	
Polarization Loss		
Axial ratio of Tx Antenna	30.00dB	
Axial ratio of Rx Antenna	1.00dB	
Bit Transfer Rate	9600bps	



Source Code

See <https://github.com/ranek/albertasat-groundstation>

The latest functioning code is available in the master branch.

Source	Description
AX25-Config	Configuration folder including necessary files for AX25 support on Linux
csp	CSP source including modified Python Bindings
.gitignore	.gitignore keeps undesirable files out of git repository.
AX25-Config	Folder containing necessary configuration files for AX25 support.
ax25-init.sh	Initialization script for setting up AX25.
cdr.py	Prints a sequence of (altitude, azimuth) pairs from next pass of satellite to terminal.
client.py	Simple client that sends 'Hello World' over KISS.
commands.py	Commands for the communications module.
communications	Folder that holds stubs for different communications modules.
communications_unit_test.py	A few unit tests for communications.
communications.py	Main communications module. All threads are contained here.
config	Satellite configuration file.
demo.py	Simulates pass of satellite and controls rotator.
exit_kiss_test.py	Script showing how to exit KISS mode so that radio settings can be changed.
packets.py	Packet structures for the different types of packets.
README.markdown	General README file.
requirements.txt	List of Python package requirements; install with pip install -r requirements.txt.
satellite.py	Dummy satellite for testing purposes.
server.py	Simple server for listening to incoming connections
web	Experimental web interface.
webgen.py	Script to export satellite pass information into JSON for experimental web interface.