PhoneSat:

Ground Testing of a Phone-Based Prototype Bus

Carmen Felix[a], Benjamin Howard[b], Matthew Reyes[c,d] , Fedor Snarskiy[e], Ryan Hickman[f], Christopher Boshuizen[c,g], William Marshall[c,h]

a. International Space University
b. Brown University
c. NASA-Ames Research Center
d. Lockheed Martin Corp.
e. University of California, Berkeley
f. Google Inc.
g. Logyx Inc.
h. Universities Space Research Association

ABSTRACT

Most of the key capabilities that are requisite of a satellite bus are housed in today's smart phones. PhoneSat refers to an initiative to build a ground-based prototype vehicle that could all the basic functionality of a satellite, including attitude control, using a smart Phone as its central hardware. All components used were also low cost Commercial off the Shelf (COTS). In summer 2009, an initial prototype was created using the LEGO Mindstorm toolkit demonstrating simple attitude control. Here we report on a follow up initiative to design, build and test a vehicle based on the Google's smart phone Nexus One. The report includes results from initial thermal-vacuum chamber tests and low altitude sub-orbital rocket flights which show that, at least for short durations, the Nexus One phone is able to withstand key aspects of the space environment without failure. We compare the sensor data from the Phone's accelerometers and magnetometers with that of an external micro-electronic inertial measurement unit.

Chapter 1 PhoneSat: A Low Cost Small Satellite Concept

1.1 INTRODUCTION

The idea of artificial satellites traces back to early 1900's, when Konstantin Tsiolkovsky started to talk about spacecrafts and orbits around the Earth. Later on, Herman Potocnik suggested the use of a geostationary orbit, but it was not until 1945 when Arthur C. Clarke described the use of communication satellites that the world became interested in this topic. The launch of Sputnik I and the start of the space age triggered a chain of technological developments accelerating the rhythm of human life. Thanks to these advances we now have mobile phones, internet, weather forecast systems, remote sensing capabilities for disaster managements, and much more.

In the beginning, sending satellites in orbit was something only governments and space agencies were able to accomplish, due to the complexity of the technologies involved and the high cost. As electronics become more miniaturized and their capability improved, the cost of building small satellites decreased. As the capability to build small satellites cheaply and with off the shelf components become available, universities, private companies and small communities started to take advantage of small satellites with hands-on projects to gain experience through research. On the other hand, a major goal of space agencies is the reduction of mission cost and operations. Satellites are with no doubts a growing sector in technology and one of the most promising areas in the future. "(the use of) Small spacecraft will increase for future space missions. If the space business is to grow and prosper as commercial aviation has, we must find a way to reduce the costs of using space. Lowering cost is the real challenge for space mission analysis and design" (Wertz & Larson, 2007).

This project took place at the Mission Design Center, Small Spacecraft Division of NASA Ames, a leader in low cost innovative mission design. LCROSS, a project led by Ames, successfully impacted a spacecraft into the moon discovering significant quantities of water. The most important feature of LCROSS is that the spacecraft was approximately a tenth of the cost of regular NASA space missions, and it used commercial components. NASA Ames has also developed Cubesat missions, e.g. PharmaSat which carried biological payloads and whose total mission cost was ~$5m.

The goal of the Small Spacecraft Division at Ames is to develop cost effective space missions for an easy, reliable and frequent access to space. The cost can be kept low by using common and reusable architectures. By utilizing secondary payloads the mission costs can be reduced while increasing at the same time scientific exploration return (Klupar, 2009).

Current spacecraft often still rely on technology developed in the 80's because of its reliability. If we could prove that existing off-the-shelf commercial technology can be used for space applications, this may potentially reduce the cost of a modern satellite from $5-500 M to $5-50 K in production.

There are different types of satellites, depending of their size and weight, like shown in Table 1. A small spacecraft can perform less science, but for a cost of millions of dollars less than an average satellite.

Development Definitions			
	Mass	**Cost**	**Time**
Large	2,000 kg +	1,000 M +	10 years +
Small	750 kg	100 M	2 – 3 years
Mini	250 kg	75 M	2 years
Micro	100 kg	50 M	1.5 years
Nano	1 – 10 kg	5 M	1 year
Pico	100 gm	< 500 k	< 1 year

Table 1 Development definitions for different types of satellites. Source: (Klupar, 2009)

PhoneSat refers to an initiative that NASA took last summer to build a prototype vehicle to test technology based on Commercial off the Shelf (COTS) components at a very low cost. Eleven students from ISU, which participated in the Summer Session Program at Ames, worked during three weeks designing and testing a prototype with a single Degree Of Freedom (DoF) and successfully transferred pictures from the vehicle to a PC acting as a ground station. The base element in their prototype was the LEGO Mindstorm toolkit that is commercially available. A picture of the final prototype is shown in Figure 1.

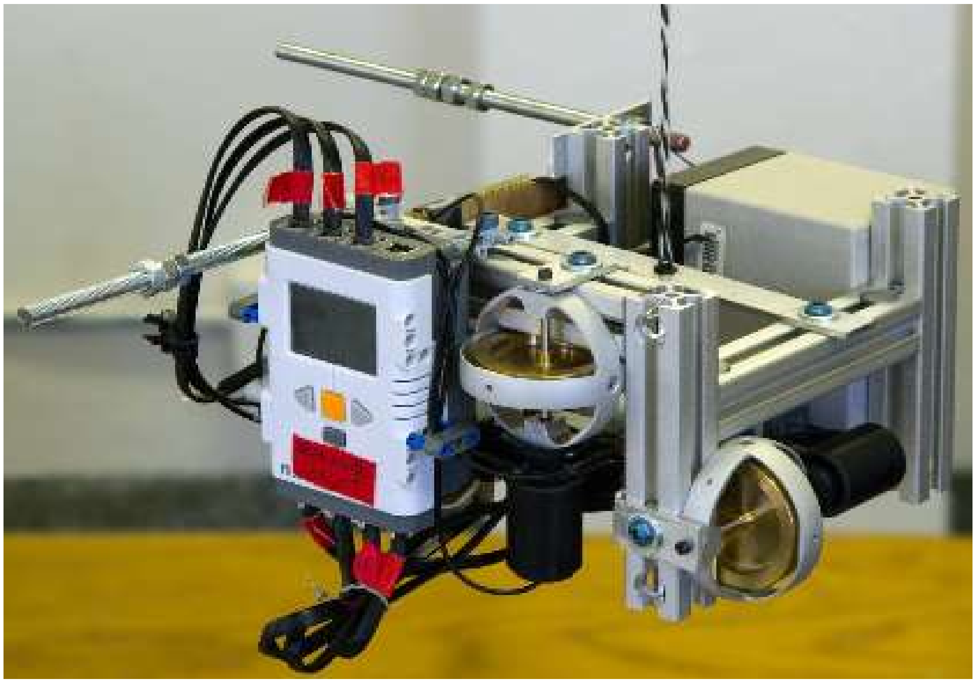


Figure 1 Prototype based in LEGO Mindstorm toolkit. Source: (Paces, 2009)

This summer, NASA designated three students to follow up the initiative to improve systems and capabilities of the test bed, making it more efficient by adding the attitude control of the vehicle with the Google's smart phone Nexus One, and ground-based space qualification testing of this and other COTS components.

Back in 2001, during the ISU annual symposium "Smaller Satellites: Bigger Business", Jim Burke asked a question to space specialists about the possibility to have a fusion of the cellular phone, internet and satellite industries through the development of a common system architecture, driven by the same micro technology (Rycroft & Crosby, 2002). If a mobile phone has the capability to control a satellite, why not using it for that purpose? We want to take the knowledge and capabilities that are in the commercial market and apply them to build a satellite.

A comparison between the first and the second version of the project with their respective characteristics is shown in Table 2.

	1ST VERSION	PHONE SAT VERSION
COST	3,500$	Budget of 5,000$
COMPONENTS	100% COTS	100% COTS
MASS	9 Kg	Not calculated yet, estimated to be approximately 2-3 kg
CONTROL SYSTEM REQUIREMENTS	3 DoF attitude control with 1 deg precision	3 DoF attitude control with 1 deg precision
STRUCTURE	Hypercube surrounded by a sensor ring, Lego-Mindstorm	2U or 3U – aluminum 6063-T6
PRINCIPAL COMPONENT	LegoNXT brick	Google Nexus One cell phone
SENSORS	LegoMindstorms Sensors	Nexus One sensors, magnetometer, sun sensors
POWER SOURCE	12 V lead-acid battery	9 V lead-acid battery – Solar panels
COMMUNICATION SYSTEM	Bluetooth and WiFi	Bluetooth / WiFi / Link Budget with internal / external antennas
SOFTWARE	Matlab, Simulink and C code	Phyton / C++ code
TARGET	Near Earth Orbit	LEO

Table 2 Comparison between first and current version of the project

1.2 PURPOSE OF THE PHONESAT PROJECT

The NASA budget for the fiscal year 2011, published by the Obama Administration last February, gives a priority to technology demonstration and Research and Development programs. A total of $7.8 billion dollars over a five years period will be dedicated to technology demonstrations towards reducing costs of missions and improving capabilities for future exploration missions (NASA, 2010). The PhoneSat project, which tries to demonstrate the use of cellular technology and COTS hardware on nano satellites, is consistent with the new NASA's strategy, The major challenge of the PhoneSat project is the integration of the Google's Nexus One mobile phone, as part of the vehicle. Space projects tend to be expensive for the need of very specialized materials and space certified instruments. The electronics inside of the Nexus One needs to be strong enough to resist mechanical shocks, acoustic and vibration conditions of a launch. At the same time, it has to withstand vacuum, radiation and the temperature variations like those that can be found in orbit.

PhoneSat mission can be stated in a single sentence: to do ground testing to prove the feasibility of a 100% COTS satellite using a smart phone as the main component.

1.3 GOOGLE CELL PHONE: NEXUS ONE

Currently, the electronics that can be found on cell phones is so powerful and advanced that it surpasses the processing requirements for a satellite. The main driver for PhoneSat is the integration of an inexpensive Nexus One Smartphone with the other necessary vehicle componants.

Nexus One is a smart phone sold by Google with the partnership and manufacturing of HTC Corporation (Figure 2). While considering different options for this project the Nexus One Google phone turned out to be the best option to successfully fulfill the requirements that a vehicle's need.

Nexus One uses the Qualcomm Inc.'s Snapdragon baseband processor with 1 GHz clock speed. It has more processing power than most of the current operational satellites. The combination of the Snapdragon processor with the Android 2.1 operating system provides fast performance and a very effective multitasking environment, which allows running many different applications with no problems. The structure that protects the cell phone is a unibody design, which provides high structural rigidity and electronics' protection in case of shocks. Nexus One has 512MB in DRAM, and 4Gbit of internal NAND flash memory, a very good storage capability. Memory can be upgraded with MicroSD cards up to 32GB.

Figure 2 Google smart phone Nexus One. Source: (Rassweiler, 2010)

Table 3 presents a list of the Nexus One's characteristics used in PhoneSat.

NEXUS ONE CHARACTERISTICS	
Physical Dimensions	11.9 cm x 6 cm x 1.2 cm 130 grams weigth
Processor	Qualcomm QSD8250, 1 GHz
Storage	Flash Memory: 512MB RAM: 512MB microSD card: 4GB (expandable to 32GB)
Power and Battery	Li-Ion 1400mAH battery Charges at 980mA from supplied charger
Connectivity	UMTS Band 1/4/8 GSM/EDGE, WiFi, Bluetooth 2.1
Sensors and Location	Digital Compass, Accelerometer and GPS
Camera	5M pixels, Autofocus, 720x480 20 fps
Platform	Android platform 2.1

Table 3 Nexus One Characteristics

These specific characteristics can be utilized into our prototype, particularly to demonstrate the feasibility to have attitude control through Bluetooth communication or serial ports, record the accelerometer and magnetometer data, the use of the camera to take pictures or video, and the use of the antennas of the cell phone for communication links to Ground Stations.

With the Nexus One, the weight and size of the vehicle is kept low, even without removing the display, which is not necessary for the vehicle. Figure 3, shows an exploded view of the phone, which gives an idea of how the electronic components are using just the bottom part of the smart phone.

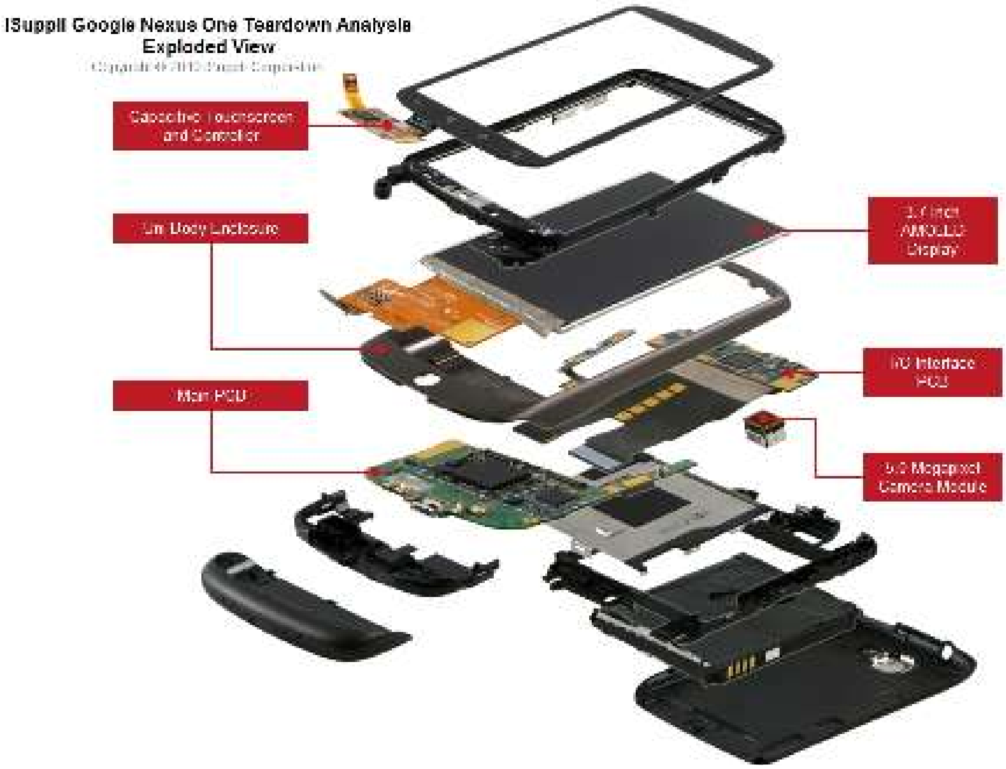


Figure 3 Exploded view of the Nexus One. Source: (Rassweiler, 2010)

An analysis of each of the Nexus One components as potential spacecraft elements is discussed in this report. An alternative from the Nexus One could be the use of the iPhone4, which regarding electronics, has a major advantage over the Google phone: an embedded 3-axis MEMS gyroscope, which is an essential component for a satellite. Testing different cell phones could consume a lot of time, so focusing on just one platform in this stage of the project is important to get progress in one platform quickly.

Another advantage of Nexus One is the fact that the Android Operation System s an open source platform, which makes developing software for PhoneSat easier.

Chapter 2: PHONESAT SYSTEMS

2.1 NEXUS ONE IMPROVING OR REPLACING VEHICLE'S SUBSYSTEMS

PhoneSat is a project for technology demonstration. The objective is to provide a prototype vehicle, but in the most simple way. The subsystems of the prototype vehicle include structures and mechanism, thermal control, attitude and control, communications, power and, command and data handling. In this chapter, I am going to discuss how each of these subsystems is intended to be designed for the PhoneSat.

Based on our requirements, we won't use propulsion, even if some ideas about propulsion have been discussed as part of future development of the PhoneSat. Regarding Command and Data Handling, the Nexus One will act as the main computer and will store data in its SD card.

2.2 SOFTWARE FOR THE PHONESAT

The Nexus One runs on Android 2.1 Operative System. For the first stage of the PhoneSat project, two scripting environments were used: Scripting Layer for Android (SL4A) and the Android Scripting Environment (ASE), both designed for developers. Through these scripts environments, we could edit and execute scripts directly on the Nexus One, while running different phone's applications like video or recording pictures in multitasking.

The language codes supported by the SL4A are Python, Perl, JavaScript, Tcl, and Shell. The language actually used was Python, which is a simple and yet effective language, but the plan is to move to C or C++ on a later phase. Current satellite software is developed in C, C++ and, Matlab; the use of C in the PhoneSat will give the project a more strong software base, and allow a comparison with codes sample from real satellites.

The codes built during the summer are able to read the Nexus One sensors - accelerometer, gyroscope, magnetometer and GPS, and provide their values. The reading changes in time, and depends on the movements of the Smartphone. The information is read and saved into an SDcard, to be analyzed after the test.

Another script was developed to take pictures automatically at pre defined intervals in time, to test the phone's capability to take pictures in space. The default video application of the Nexus One has a limit of 30 minutes of video recording. A modification of the video recording software was necessary to increase the limit, in order to allow the phone to take video as long as possible during launches and flights.

For testing the vehicle on ground and during the first launches, a program for data collection through Bluetooth was developed as well. The first idea for the Bluetooth was to set up the control of the Gyroscopes through the cell phone. Communication from the Nexus One to the motor controller of the gyroscopes, and the reception of the values detected by an IMU was successfully tested.

Software development posed several challenges, since SL4A was unable to execute some of the commands or tasks that we needed for the project. The help of the Google team was really important in order to clear bugs in the Script Environment.

Some of the issues that were discovered during the software developing were that the display of the Nexus One needed to be turned on in order to keep recording video, and in order to take pictures constantly. If the display turns off or goes into sleep mode, the time between pictures changes into a random sequence, and the video recording stops. A lock mechanism was needed to solve this issue. In another case, in the code section that reads sensors and GPS, if the command asks for both readings at the same time, the reading turns out to be inconsistent. The recording of sensors and GPS needed to be though different commands.

Receiving data from the IMU was also something that required a code re-arrangement. In order to save the data correctly and avoid losing information from the IMU, a delay needed to be set up. This delay limited the capability to read data from the accelerometer, magnetometer and gyroscopes to a read every two of four seconds. Since four seconds is an interval too long for a launch time, the data were not as accurate as desired. In order to avoid losing vital information, a separation of these two functions were necessary.

2.3 STRUCTURE, SIZE AND THERMAL DESIGNS

The structural goal of the PhoneSat is to have a small vehicle that weights approximately two kilograms. The first version of the project was a hypercube with a total mass of 9 kg, using the Lego NXT as the computer system. Instead of keep using heavy aluminum parts 8020 as in the first versions, we decided to use a high-strength aluminum (6063-T6) MicroRax structure in order to make the structural framing lighter. Because the pieces of MicroRax are not at Ames yet, the new physical design has not been done.

Before selecting the use of MicroRax, we used the structure of the LegoSat, the first version of this project, to accommodate the cell phone and reaction wheels in a cube. This structure can help in the demonstration of one degree of freedom (DoF) of the PhoneSat. Figure 4 shows a picture of the first iteration of the structure for the PhoneSat.

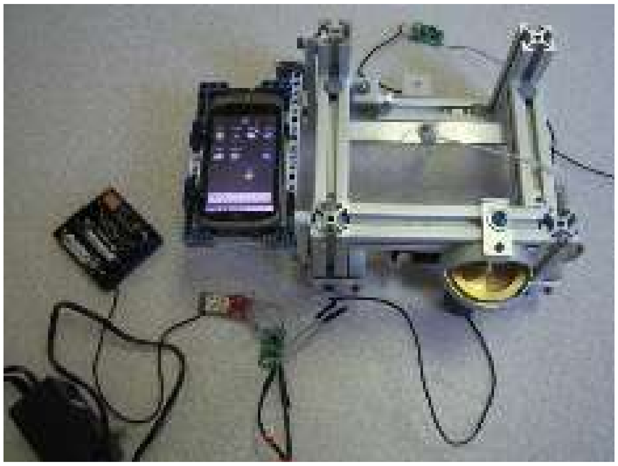


Figure 4 PhoneSat in its first structure model

A structure of “3U” cube, 30x10x10 cm, has been evaluated as the final structure for the PhoneSat, taking into account the possible space we would be able to get for a final launch in March 2011. Since the material is missing, physical tests were not possible. A possible physical accommodation of the elements was discussed instead.

The idea is to have the three reaction wheels in a single cube, the cell phone or cell phones in another one, and all the external electronics and batteries in the third cube. This layout needs to be balanced, so iterations around this idea have to be done. The use of solar panels is considered in order to provide additional power to the Nexus One battery capability. The design of a 2U cube structure will be analyzed in the following months, to be prepared for a possible launch to LEO in November.

Regarding radiation and thermal control in space, we are considering using Mylar sheets to protect the elements inside the PhoneSat and increase its survivability. Even if the requirement is to complete one orbit, taking precautions for radiation can increase the probability to be able to complete more orbits and get more useful data from this project. Fulfill our requirements is our goal, but to do more will be better.

The most critical component in the design, the cell phone, was already tested in vacuum conditions and thermal cycles, from -30 Celsius degrees up to 40 Celsius degrees. More details about testing in chapter 3.

A pre-design of a structure for launch testing was made by Ben Howard, a member of the team. The structure based in acrylic material was done specifically to test a gyroscope, two cell phones and, an IMU during the launches that we had on the 23 and 24 of July. More details about the launches are explained in Chapter 4.

The approximate weight of the final PhoneSat is detailed in Table 5.

SUBSYSTEM	PhoneSat Weight Budget (kg)
Attitude Control – Gyroscopes (3)	1.035
Communications (external antenna)	0.2
Thermal	0.1
Power (3 batteries)	0.4728
Structure (MicroRax)	0.90
Nexus One (2 for redundancy)	0.260
Total	2.96

Table 5 PhoneSat Weight Budget

2.4 ATTITUDE DETERMINATION AND CONTROL SYSTEM

PhoneSat will have a three axis attitude control. In this configuration, the attitude level or torque of each of the reaction wheels helps to reach a specific equilibrium in order to get a pointing precision or accuracy pointing. This will be necessary to point the vehicle towards the sun to charge the battery with solar panels, or to direct the antenna to a specific area on ground.

The first stage of this subsystem is to demonstrate the feasibility of an attitude control based on three super precision gyroscopes that will act as reaction wheels, to provide three Degrees of Freedom (DoF) to the vehicle. Figure 5 shows the accommodation of the gyroscopes-reaction wheels during the first iteration of the PhoneSat structure.

Figure 5 Gyroscopes accommodated in the Phonesat

PhoneSat will perform attitude control by coordinating acceleration values from the Nexus One to the three gyroscopes acting as reaction wheels. Taking into account that the Nexus One has its own accelerometer and magnetometer, some tests were done to study their efficiency; the results will be explained in Chapter 4. The use of a Sun sensor is also considered in the future design.

The first iteration, designed to get a single degree of freedom (DoF), revolved around the combination of a gyroscope working with one motor control through Bluetooth interface. The objective was to convert the code developed for the LEGO Sat in Simulink and Matlab, into Python, which is the easiest language to interact with Android Script Engine (ASE) platform in Nexus One. Since I didn't have access to the entire Simulink or Matlab code, it was really difficult to replicate it completely in Python.

The motors controller drivers that were designed to work with the gyroscopes were not working properly. Even with a basic test code to turn on and off the gyroscope, both motor controllers were reacting differently. We decided to change the motor controllers for simpler ones which were designed to work in a robotic toolkit.

I decided to develop a complete new program in Phyton to make one of the gyroscopes rotate through Bluetooth communication. At the end, we managed to create a communication channel between the Nexus One and a gyroscope, allowing it to rotate at different speeds.

This objective was accomplished thanks to the easy access of the open source platform of the Nexus One phone, and to the Bluetooth communication that was set up between Nexus One and the motor, interfacing with an external Bluetooth and an Arduino controller.

Figure 6 shows the connection between the Gyroscope's motor and the Nexus One.

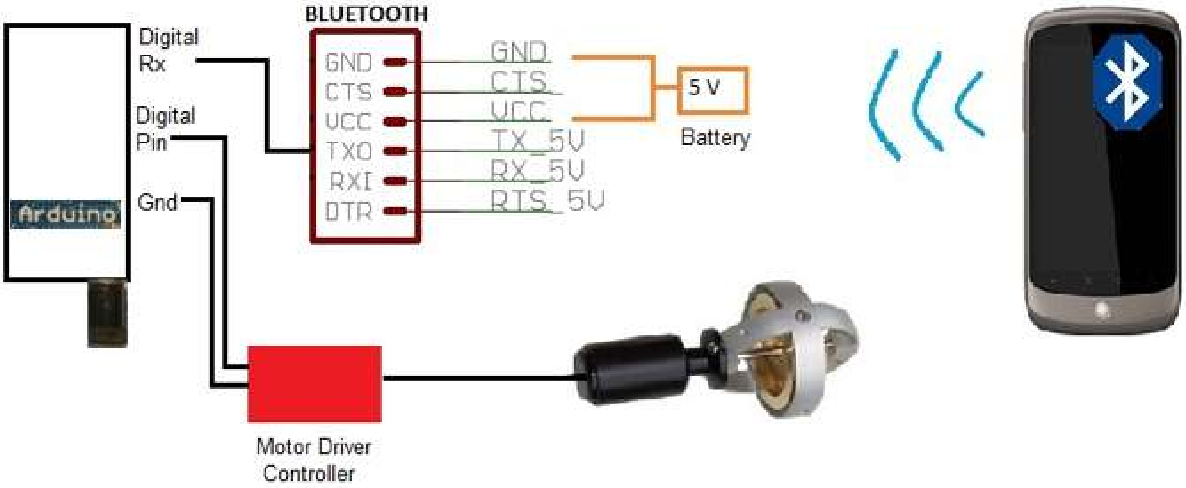


Figure 6 Communication between cell phone and gyroscope

The code was based on the Amarino free source application, developed in C++ language. Some modifications were made in order to read the values of the Nexus One's accelerometer and send them to the motor drive to make it increase or decrease the velocity of the gyroscope. In this way, while the cell phone changes its position, the accelerometer values changes and so the gyroscope changes its velocity. Figure 7, shows the Roll, Pitch and Azimuth axis in the Nexus One.

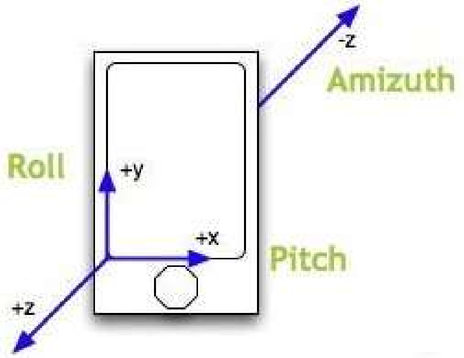


Figure 7 Roll, Pitch and Azimuth Axis of the Nexus One

This achievement demonstrated that the values from the sensors of the cell phone can be used to trigger a change velocity in the reaction wheel. The next step is to get the precise control of the gyroscope velocity to achieve the 3 Degrees of freedom (DoF) for the vehicle. For a video demonstration of the rotation of the gyroscope through the Nexus One's Bluetooth this:
http://www.youtube.com/watch?v=Zb5x7HNaalc

One of the problems we faced with the gyroscopes is that the friction that the motor have with the rings of the gyroscope reduces the velocity at which the wheel starts rotating.

The current gyroscopes can reach a speed up to 12,000 rpm. The amount of times that the gyroscope spins can be increased if the electric motor is detached, but to do that, a system needed to be created to attach and detach the motor to the gyroscope automatically.

The next option to improve the system is to change the motors in the gyroscope. The idea of using brushless motors which have less friction is considered. New small gyroscopes arrived during the last week of the internship to the lab. More tests about the velocity and the functionality for this type of gyroscope need to be done, in order to understand if they can be actually used in space or not. The most direct advantage for these gyroscopes is that they are really small and lighter than the ones currently used.

A summary of the current Attitude Determination and Control subsystem is presented in the table 6.

Component	Number	Mass (Kg)	Power (Watts)
Gyroscopes	3	1.035	14.5 / 2.9
Motor controller	3	Xxx	Xxx
Sun Sensor	1	0.01	
Total		Xxx	xxx

Table 6 ADCS summary

2.5 COMMAND AND DATA HANDLING

The data storage and processing for the PhoneSat will be done by the Nexus One, taking advantage of its 1GHz internal processor. One of the mission requirements is the transmission of a picture from the PhoneSat in space, to the ground station. This operation does not require too much memory space per se, but since we are planning to manage all the data through the cell phone, data rates and memory capacity have to be taken in account.

So far, tests I have done show that internal memory and storage capacity are enough for three hours of data management. . Receiving Gyroscopes, acceleration, magnetometer and, clock information of an IMU for a time period of two hours with a frequency of 10 Hz produces 3.33 MB of data. Data coming from the Nexus One accelerometers, magnetometers, gyroscope and GPS recorded with a frequency of 60 Hz for a time period of 2 hours produces 241 MB.

The Nexus One can use a 32GB SDHD memorycard, which provides enough space to collect data for the entire one orbit period and even more.

The data acquisition volume is shown in the Table 7. It takes in account a pictures every 30 seconds, a reading of Nexus One's sensors with a frequency of 60 Hz, and IMU information at 10 Hz frequency.

	# Samples every 90 minutes	Data volume (Mbits)	Data rate (Hz)
Pictures	180	270	0.033
N1 Sensors	126,202	180	60
IMU	54,000	2.5	10
Gyros?			
Total		452.5	

Table 7 Data acquisition rate during testing phase

2.5 TRACKING, TELEMETRY AND COMMAND SYSTEM

As part of the technology demonstration side of the PhoneSat, a communication link analysis using the internal antennas of the Nexus One was done taking into account an altitude of 1500 km in LEO.

Since the Nexus One antennas are omnidirectional and not designed for long distance communication purposes, they present constraints like a very low radiated power, a factor which affects the size of the ground station antenna.

For the PhoneSat, the telemetry information that we would be interested in collecting is acceleration, angular rate, temperature, voltage, current, magnetic field, and the processor performance. For the first stage, the requirement is just to get a picture. At this stage, the telemetry link will not take into account all the readings coming from the vehicle, as the primary requirement is to provide the downlink for the picture. The uplink was considered in the study, but will probably be handled in the next stage of the PhoneSat project.

For connectivity purposes, the Nexus One internal electronics provides communications in the bands presented in Table 8.

Connectivity	Frequencies MHz
UMTS **1 – 2100, 4 – 1700, 8 – 900**	Band 1 → 1922.4 - 1977.6 Band 4 → 1712.4 – 1752.6 Band 8 → 900
GSM / EDGE	850 → 824.2 – 848.8 900 → 880.2 – 914.8 1800 → 1710.2 – 1784.8 1900 → 1850.2 – 1909.8
WLAN / WiFi / BT 2.1	2412 – 2462

Table 8 Communication bands enabled in the Nexus One. Source: (FCC, 2009)

The chips that handle communication in the Nexus One are the Qualcomm RTR6285 RF Chip transceiver, and the Broadcom BCM4329 chip. Looking at the datasheets of these two chips, I found out that the Qualcomm RTR6285 supports all UMTS bands, but only the bands mentioned in the Table 8 are enabled through the Nexus One's power amplifiers.

The relationships between the frequency bands and the amplifiers are:

- Skyworks Sky77336 → quad band GSM amplifier.
- Skyworks Sky77191 → WCDMA (3G DATA) amplifier and 1700 Mhz
- A5001 → single band amplifier for the UMTS band 1, 2100 Mhz
- A5008 → single band amplifier for the UMTS band 8, 900 Mhz.

The Broadcom BCM4329 chip handles 3G, Bluetooth and FM transceiver. Even if the chip has the capability of FM transmission and reception, the Nexus Ones does not provide the radio option.

The specifications for WLAN, Bluetooth, and FM from the BCM4329 chip are explained in Table 9

TECHNICAL SPECIFICATIONS			
	WLAN	BLUETOOTH	FM
Standard	802.11a/b/g/n	Bluetooth 2.1 + EDR	RDS (E), RBDS (NA)
Modulation	OFDM, CCK, DQPSK, DBPSK	GFSK, DQPSK, 8-DPSK	
Frequency	2.4 – 2.497 GHz 4.9 – 5.85 GHz	2402-2480 MHz	76 – 108 MHz
RF Output Power	2.4 GHz : 18 dBm 5 GHz : 15 dBm	Class 1, Class 2	117 dbuV Receive: -107 dBm

Table 9 Technical Specification for Nexus One connectivity

The datasheet of the BCM4329 chip, specifies that the receive signal path of the Bluetooth and WLAN can be shared. The block diagram is shown in Figure 8.

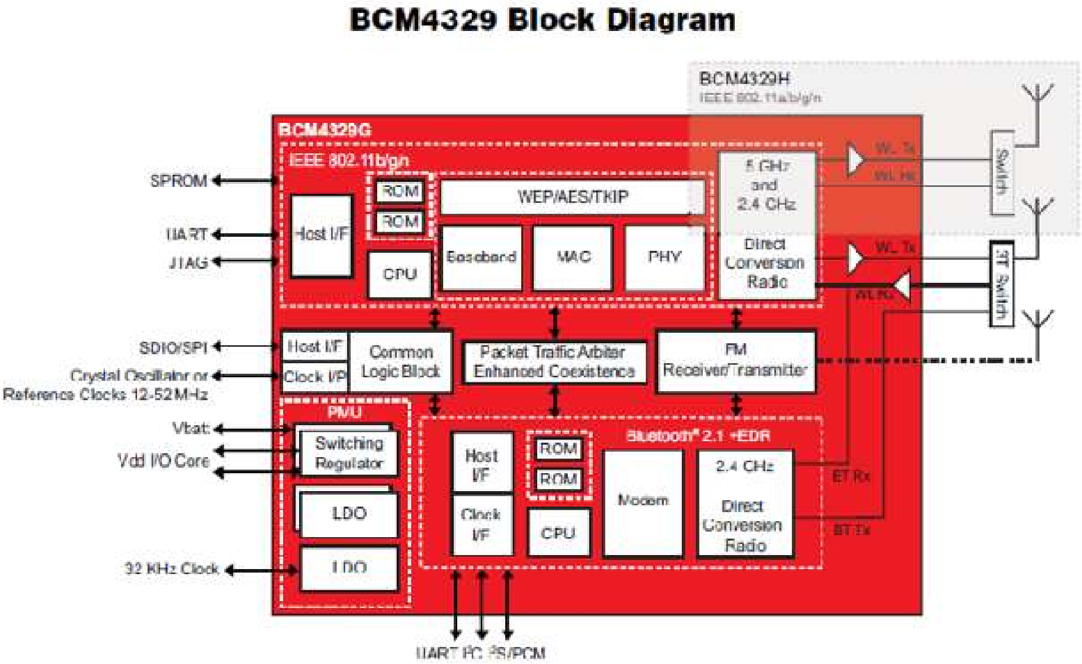


Figure 8 BCM4329 block diagram. Source: (BROADCOM, 2008)

Even if the internal antennas of the Nexus One are not space qualified, the success of the PhoneSat project might bring a new market for its internal electronics. . A similar case was the commercial transceiver Microhard MHX-2400 which was successfully tested on space and now is widely used in small satellites. (Mass, 2007)

Since the nature of the project is to reduce the cost of small satellites, using small antennas for ground stations was a priority. Even if a smaller antenna provides a smaller gain, its beam width is wider and so its pointing error is lower. The gain of a small ground station antenna as well as its system noise temperature can be improved, if necessary, by adding an amplifier.

The analysis was very conservative regarding system losses, since there is no previous experience about using a cell phone communication set for transmission and reception in space.

A link budget calculator was used which allows filling out different options while designing the satellite communication link in order to compare results. The option of using the Ames Mission Design Center's link budget calculator was discarded because the frequencies available and the size of the antennas available didn't match with the Nexus One antenna's specifications.

A feasibility analysis is explained next for the theoretical use of a phone from low earth orbit. Downlink communications have been implemented with a BPSK link at 2 Kbps in UMTS or 1.76 GHz frequency from space and with a 500 bps uplink with 2.16 GHz from the ground. The antenna inside of the Nexus One is omnidirectional and transmits with a power of 0.25 Watts. A multiband dual polarization antenna which provides 21 dB gain, closes the link with a low margin. Even if the requirements call for downlink only, the link budget analyzes communications in both directions.

The uplink budget takes into account the values given in the datasheet for a ground antenna, an altitude of 1500 km in LEO and an elevation angle of 65 degrees. The angle was chosen because angles between 50 degrees and 70 degrees are the most common for operations. The elevation angle is unknown. For this reason, the Link Budget Excel file takes into account angles between 0 and 90 degrees, closing the uplink budget successfully. The link budget for the uplink command control is shown in the Table 10, with input parameters in orange.

Parameter	Symbol	Unit	Uplink	Comments
Frequency	F	GHz	2.16	
Wavelength	Λ	M	0.14	
Transmitter Power	P	Watts	10.00	From antenna datasheet
Transmitter Power	P	dBm	40.00	
Transmitter Power	P	dBW	10.00	
Transmitter Line Loss	L_l	dB	-3.00	Set as a common value
Transmit Antenna Beamwidth	θ_t	Deg	44.00	From antenna datasheet
Transmit Antenna Efficiency	H	-	0.60	From antenna datasheet
Peak Transmit Antenna Gain	G_{pt}	dBi	21.00	From antenna datasheet
Transmit Antenna Diameter	D_t	M	0.22	
Transmit Antenna Pointing Offset	e_t	Deg	4.40	Set to 10% of Antenna Beamwidth
Transmit Antenna Pointing Loss	L_{pt}	dB	-0.12	
Transmit Antenna Gain (net)	G_t	dBi	20.88	
Equiv Isotropic Radiated Power	$EIRP$	dBW	27.88	
Altitude of Satellite	H	Km	1500.00	LEO
Elevation angle	E	Deg	65.00	Spacecraft Elevation angle assumed as 65

Earth Angular Radius	P	Deg	54.06	
Nadir angle	Θη	Deg	20.01	
Earth central angle	$Λearth$	Deg	4.99	
Propagation Path Length	S	Km	1622.09	
Space Loss	L_s	dB	-163.32	
Propagation & Polarization Loss	L_a	dB	-3.00	According to atmospheric attenuation graph less than 0.5 but being conservative higher losses
Receive Antenna Diameter	D_r	M	0.03	Antenna inside N1
Peak Receive Antenna Gain (net)	G_{rp}	dBi	0.00	Omnidirectional Antenna N1
Receive Antenna Beamwidth	$θ_r$	Deg	324.83	
Receive Antenna Pointing Error	e_r	Deg	32.48	Set to 10% of Antenna Beamwidth
Receive Antenna Pointing Loss	L_{pr}	dB	-0.12	
Receive Antenna Gain	G_r	dBi	-0.12	
Noise Temperature of Receiver	T_r	K	253.15	Transform from C to K
Receiver Noise Figure	F	dB	2.73	
Receiver Line Loss	L_r	K	0.89	For a -0.5 Db
System Noise Temperature	T_s	K	573.43	
Data Rate	R	Bps	500.00	Suggested value (no orbit)
Eb/No (1)	E_b/N_o	dB	35.35	
Carrier to Noise Density Ratio	C/N_o	dB-Hz	62.34	
Bit Error Rate	*BER*	-	10^-5	Acceptable value
Required Eb/No (2)	*Req* E_b/N_o	dB	9.60	N1 for BPSK/QPSK
Implementation Loss (3)	-	dB	-3.00	Commonly -2, being conservative -3
Margin	-	dB	22.75	

Table 10 Uplink budget for 2.16GHz

The link budget for the downlink telemetry control, important for PhoneSat basic requirements, is shown in the Table 11, with input parameters in orange.

Parameter	Symbol	Unit	Downlink	Comments
Frequency	F	GHz	1.76	
Wavelength	$Λ$	M	0.17	
Transmitter Power	P	Watts	0.25	
Transmitter Power	P	dBm	24.00	From Nexus One FCC report
Transmitter Power	P	dBW	-6.00	
Transmitter Line Loss	L_l	dB	-3.00	Set as a common value
Transmit Antenna Beamwidth	$θ_t$	Deg	350.00	N1 omnidirectional antenna
Transmit Antenna Efficiency	H	-	0.60	N1 omnidirectional antenna
Peak Transmit Antenna Gain	G_{pt}	dBi	0.00	N1 omnidirectional antenna
Transmit Antenna Diameter	D_t	M	0.03	
Transmit Antenna Pointing Offset	e_t	Deg	35.00	Set to 10% of Antenna Beamwidth

Transmit Antenna Pointing Loss	L_{pt}	dB	-0.12	
Transmit Antenna Gain (net)	G_t	dBi	-0.12	
Equiv Isotropic Radiated Power	*EIRP*	dBW	-9.12	
Altitude of Satellite	H	Km	1500.00	LEO
Elevation angle	E	Deg	65.00	Spacecraft Elevation angle assuming as 65
Earth Angular Radius	P	Deg	54.06	
Nadir angle	Θη	Deg	20.01	
Earth central angle	*Aearth*	Deg	4.99	
Propagation Path Length	S	Km	1622.09	
Space Loss	L_s	dB	-161.53	
Propagation & Polarization Loss	L_a	dB	-3.00	According to atmospheric attenuation graph less than 0.5 but being conservative higher losses
Receive Antenna Diameter	D_r	M	0.22	GS Antenna
Peak Receive Antenna Gain (net)	G_{rp}	dBi	20.90	
Receive Antenna Beamwidth	θ_r	Deg	54.03	
Receive Antenna Pointing Error	e_r	Deg	5.40	Set to 10% of Antenna Beamwidth
Receive Antenna Pointing Loss	L_{pr}	dB	-0.12	
Receive Antenna Gain	G_r	dBi	20.78	
Noise Temperature of Receiver	T_r	K	323.15	Transform from C to K
Receiver Noise Figure	F	dB	3.25	
Receiver Line Loss	L_r	K	0.89	For a -0.5 dB
System Noise Temperature	T_s	K	722.08	
Data Rate	R	Bps	2000.00	Suggested value (no orbit)
Eb/No (1)	E_b/N_o	dB	14.01	
Carrier to Noise Density Ratio	C/N_o	dB-Hz	47.02	
Bit Error Rate	*BER*	-	10^-5	Acceptable value
Required Eb/No (2)	Req E_b/N_o	dB	9.60	N1 for BPSK/QPSK
Implementation Loss (3)	-	dB	-3.00	Commonly -2, being conservative -3
Margin	-	dB	1.41	

Table 11 Downlink budget for 1.76 GHz

Nexus Ones is a product made in collaboration between Google and HTC. For this reason, it was not possible to get the specifications for its internal antennas through Google, since this information is HTC's property. An extensive research was done to get antennas specifications and power consumption of the cell phone for each radio frequency (RF). The parameters for the Nexus One's antenna used in this case were obtained from the

“Electromagnetic Emissions Compliance Report from the Federal Communication Commission” and the “HTC Dream Service Manual”.

For the downlink, the link closed with a low margin assuming a 65 degrees elevation angle. An analysis with angles between 0 and 90 degrees shows a decrease in performance with a negative margin between 0 and 50 degrees. This result suggests that is not reliable to use the Nexus One internal antenna with this frequency for downlink.

A second study, using the frequencies of 2.4GHz, and 4.9 GHz was performed. For the link budget using the 2.4 GHz frequency, I considered a dish antenna of 1.5 m, with a gain of 30 dBi. A link analysis taking into consideration different elevation angles from 0 to 90 degrees was done, and I found that the uplink closes well, while the downlink presents a negative margin for angles between 0 to 20 degrees. A link budget summary for the 2.4GHz frequency at 65 degrees elevation angle is shown in Table 12:

Parameter	Symbol	Unit	Source	Uplink	Downlink
Frequency	f	GHz	Input Parameter	2.40	2.40
Wavelength	λ	M	$\lambda = c / f$	0.12	0.12
Transmitter Power	P	Watts	Input Parameter or P Watts = 10^((dBm-30)/10)	10.00	0.06
Transmitter Power	P	dBm	Input Parameter or P dBm = 10log(P watts)+30	40.00	17.41
Transmitter Power	P	dBW	Input Parameter or P dBW = 10log(P watts)	10.00	-12.59
Transmitter Line Loss	L_l	dB	Input Parameter	-3.00	-3.00
Transmit Antenna Beamwidth	θ_t	Deg	Input Parameter	5.83	291.67
Transmit Antenna Efficiency	η	-	Input Parameter	0.60	0.60
Peak Transmit Antenna Gain	G_{pt}	dBi	Gpt=(Pi^2*Dr^2*n)/λ^2 or =44.3-10*log10(θxθy)	30.00	1.10
Transmit Antenna Diameter	D_t	M	Dt = 21/(θt*f)	1.50	0.03
Transmit Antenna Pointing Offset	e_t	Deg	Input Parameter	0.58	29.17
Transmit Antenna Pointing Loss	L_{pt}	dB	Lpt=-12(et/θt)^2	-0.12	-0.12
Transmit Antenna Gain (net)	G_t	dBi	Gt=Gpt+Lpt	29.88	0.98
Equiv Isotropic Radiated Power	$EIRP$	dBW	EIRP=P+Ll+Gt	36.88	-14.61
Altitude of Satellite	H	Km	Input Parameter	1500.00	1500.00
Elevation angle	ε	Deg	Input Parameter	65.00	65.00
Earth Angular Radius	ρ	Deg	ρ = sin^-1(Re/(Re+H))	54.06	54.06
Nadir angle	$\theta\eta$	Deg	sin (θη) = cos(ε)*sin(ρ)	20.01	20.01
Earth central angle	$\lambda earth$	Deg	θη + λearth + ε = 90 deg	4.99	4.99
Propagation Path Length	S	Km	D=Re*(sin(λearth)/sin(θη))	1622.09	1622.09
Space Loss	L_s	dB	Ls=20log(c)-20log(4*PI)-20log(S)-20log(f)	-164.25	-164.25
Propagation & Polarization Loss	L_a	dB	Figure 13-10	-3.00	-3.00
Receive Antenna Diameter	D_r	M	Input Parameter	0.03	1.50
Peak Receive Antenna Gain (net)	G_{rp}	dBi	Input Parameter	1.10	30.00

Receive Antenna Beamwidth	θ_r	Deg	θr= 21/(Dt*f)	291.67	5.83
Receive Antenna Pointing Error	e_r	Deg	Input Parameter	29.17	0.58
Receive Antenna Pointing Loss	L_{pr}	dB	Lpr=-12(er/θr)^2	-0.12	-0.12
Receive Antenna Gain	G_r	dBi	Gr=Grp+Lpr	0.98	29.88
Noise Temperature of Receiver	T_r	K	Input Parameter	253.15	233.15
Receiver Noise Figure	F	dB	F=1+Tr/To	2.73	2.56
Receiver Line Loss	L_r	K	Input Parameter	0.89	0.89
System Noise Temperature	T_s	K	Ts=Tr+((To*(1-Lr))/Lr) + ((To*(F-1)) / Lr)	573.43	530.96
Data Rate	R	Bps	Input Parameter	500.00	1000.00
Eb/No (1)	E_b/N_o	dB	Eb/No=P+Ll+Gt+Lpr+Ls+La+Gr+228.6-10logTs-10logR	44.51	19.25
Carrier to Noise Density Ratio	C/N_o	dB-Hz	C/No=Eb/No+10logR	71.50	49.25
Bit Error Rate	BER	-	Input Parameter	10^-5	10^-5
Required Eb/No (2)	$Req\ E_b/N_o$	dB	Table 13-11	10.30	10.30
Implementation Loss (3)	-	dB	Estimate	-3.00	-3.00
Margin	-	dB	(1)-(2)+3	31.21	5.95

Table 12 Uplink budget for 2.4 GHz

A link budget summary for the 5GHz frequency is shown in the Table 13:

Parameter	Symbol	Unit	Source	Uplink	Downlink
Frequency	f	GHz	Input Parameter	5.30	5.80
Wavelength	λ	M	$\lambda = c / f$	0.06	0.05
Transmitter Power	P	Watts	Input Parameter or P Watts = 10^((dBm-30)/10)	10.00	0.06
Transmitter Power	P	dBm	Input Parameter or P dBm = 10log(P watts)+30	40.00	17.41
Transmitter Power	P	dBW	Input Parameter or P dBW = 10log(P watts)	10.00	-12.59
Transmitter Line Loss	L_l	dB	Input Parameter	-3.00	-3.00
Transmit Antenna Beamwidth	θ_t	Deg	Input Parameter	4.00	291.67
Transmit Antenna Efficiency	η	-	Input Parameter	0.60	0.60
Peak Transmit Antenna Gain	G_{pt}	dBi	Gpt=(Pi^2*Dr^2*n)/λ^2 or =44.3-10*log10(θxθy)	31.00	1.10
Transmit Antenna Diameter	D_t	M	Dt = 21/(θt*f)	0.90	0.01
Transmit Antenna Pointing Offset	e_t	Deg	Input Parameter	0.40	29.17
Transmit Antenna Pointing Loss	L_{pt}	dB	Lpt=-12(et/θt)^2	-0.12	-0.12
Transmit Antenna Gain (net)	G_t	dBi	Gt=Gpt+Lpt	30.88	0.98

Equiv Isotropic Radiated Power	*EIRP*	dBW	EIRP=P+Ll+Gt	37.88	-14.61
Altitude of Satellite	*H*	Km	Input Parameter	1500.00	1500.00
Elevation angle	ε	Deg	Input Parameter	65.00	65.00
Earth Angular Radius	ρ	Deg	ρ = sin^-1(Re/(Re+H))	54.06	54.06
Nadir angle	θη	Deg	sin (θη) = cos(ε)*sin(ρ)	20.01	20.01
Earth central angle	*λearth*	Deg	θη + λearth + ε = 90 deg	4.99	4.99
Propagation Path Length	*S*	Km	D=Re*(sin(λearth)/sin(θη))	1622.09	1622.09
Space Loss	L_s	dB	Ls=20log(c)-20log(4*PI)-20log(S)-20log(*f*)	-171.13	-171.92
Propagation & Polarization Loss	L_a	dB	Figure 13-10	-3.00	-3.00
Receive Antenna Diameter	D_r	M	Input Parameter	0.03	0.90
Peak Receive Antenna Gain (net)	G_{rp}	dBi	Input Parameter	1.10	32.50
Receive Antenna Beamwidth	θ_r	Deg	θr= 21/(Dt**f)*	132.08	4.02
Receive Antenna Pointing Error	e_r	Deg	Input Parameter	13.21	0.40
Receive Antenna Pointing Loss	L_{pr}	dB	Lpr=-12(er/θr)^2	-0.12	-0.12
Receive Antenna Gain	G_r	dBi	Gr=Grp+Lpr	0.98	32.38
Noise Temperature of Receiver	T_r	K	Input Parameter	253.15	233.15
Receiver Noise Figure	*F*	dB	F=1+Tr/To	2.73	2.56
Receiver Line Loss	L_r	K	Input Parameter	0.89	0.89
System Noise Temperature	T_s	K	Ts=Tr+((To*(1-Lr))/Lr) + ((To*(F-1)) / Lr)	573.43	530.96
Data Rate	*R*	Bps	Input Parameter	500.00	1000.00
Eb/No (1)	E_b/N_o	dB	Eb/No=P+Ll+Gt+Lpr+Ls+La+Gr+228.6-10logTs-10logR	38.63	14.08
Carrier to Noise Density Ratio	C/N_o	dB-Hz	C/No=Eb/No+10logR	65.62	44.08
Bit Error Rate	*BER*	-	Input Parameter	10^-5	10^-5
Required Eb/No (2)	*Req* E_b/N_o	dB	Table 13-11	10.30	10.30
Implementation Loss (3)	-	dB	Estimate	-3.00	-3.00
Margin	-	dB	(1)-(2)+3	25.33	0.78

Table 13 Downlink budget for 5.3 GHz

To ensure the link budget, an easy solution is the use of a bigger antenna. A three meter antenna would do the job, but for a higher cost. In our case, the communication link has to be done for a demonstration technology and not precisely for a satellite that will last years on orbit. By taking fewer losses the link could close easier, but being conservative at this point is necessary.

The information that I received from the Google employees involved in the project was that the antenna's performance was very variable. Their recommendation is to use external antennas for the link budget.

Besides the link performance challenge of the Nexus One's internal antennas, the utilization of its internal antennas would necessarily require access to the phone's native in order to modify the data transmission frequency. The easier and most reliable way to implement communication in PhoneSat is to use external antennas, which can be easily adapted to the Nexus One having the access to the system's native code.

An analysis about the possible external antennas that can be used for PhoneSat communication, indicate that the option of selecting Rubber Duck Antennas onboard of the PhoneSat is the best one. The ones offered by L-com Global Connectivity has been proven to survive several launches and even rocket crashes. The options they offer can be used for 1.9 GHz, 2.4 GHz, 3.5 GHz, 900 MHz, and multi bands.

The circularly polarized patch antennas for ISM Band that Laird Technologies offers are a good option to also test the communication link, acting as ground stations. With this antenna, we can have a high signal reception in areas where the scattering and multipath is high. For testing purposes, an antenna fro 900MHz, with 7.5 dBi gain, beamwidth in both planes of 65 degrees and power consumption of 1 Watt, is enough.

2.6 POWER SYSTEM

To provide the required power for the PhoneSat, we considered four sources: the Nexus One's internal battery, an external battery connected to the cell phone, solar panels, and external batteries for the reaction wheels.

The Nexus One has a Lithium-Ion battery of 3.7 VDC and 1400mAh, with a consumption of 5.18 Whr. Extended tests were performed to measure the battery's charge duration. The result was a 100% charged battery lasts up to three hours while recording and storing video, or four hours recording accelerometer, gps, and external IMU data through Bluetooth, while taking pictures.

Even if the most common batteries used in space are Nickel-Cadium (NiCd) and Nickel-Hydrogen (NiH2), the battery of the Nexus One seems to work fine in vacuum conditions, extreme temperatures, and space like environment.

Calculating the power generated by a solar array in the PhoneSat, the result is that for a 3U structure with a single face solar array of 30% efficiency, just 1.23 Watts are generated.

$$\mathrm{Aa} = \frac{\mathrm{P}}{\mathrm{eff} * 1367}$$

Where:

- *Aa* is the surface of the solar array in m^2
- *P* is power generated in Watts
- *Eff* is the efficiency of the solar array
- 1367 is the solar constant in W/m^2

Since the mission life time is just a single orbit, the power degradation of the solar panel does not plays an important role in the design process. A power budget is analyzed in the Table 14:

Component	Power (W)
Gyroscopes – Reaction Wheels (3)	43.5
Solar arrays drives	0.2
Nexus One operations	5.18
Total	51.3

Table 14 PhoneSat Power Budget

The tests so far have been done using two external 9 V batteries for Bluetooth communications to the IMU, a rechargeable 5 V battery connected externally to the Nexus One, and the full capacity of the Lithium-Ion Nexus One's battery.

More power consumption tests with the overall system working must to be done prior to next launches.

Chapter 3: TESTING THE PHONESAT COMPONENTS

3.1 Vacuum and Thermal Testing

The functionality of one Nexus One and one gyroscope were tested in a vacuum chamber with thermal cycles. This test were done to simulate the space environment and to prove that such hardware could be used in PhoneSat.

The fidelity of the test environment was not the best available for LEO, but it was still good enough to suggest that the Nexus One could be used successfully in extreme temperatures and in vacuum conditions. I developed a testing plan, considering changes of temperature from -30° up to +40° Celsius. The test's duration was approximately two hours. The vacuum condition employed were 30 Torr, which can be augmented to 50 Torr in future tests.

More tests are needed in order to verify the functionality of the cell phone connectivity - Bluetooth, WI-Fi, phone calls and more – and the reliability of the electronics in vacuum conditions.

The tests were conducted with two of the software developers that are supporting the project from Google. The Nexus One and a Motorola cell phone were tested at the same time, running a program to track and record the values of the cell phone sensors. The data was displayed in a Laptop, connected to the cell phones through a serial cable. The test was successful.

The test bed is depicted in Figures 9, 10 and 11.

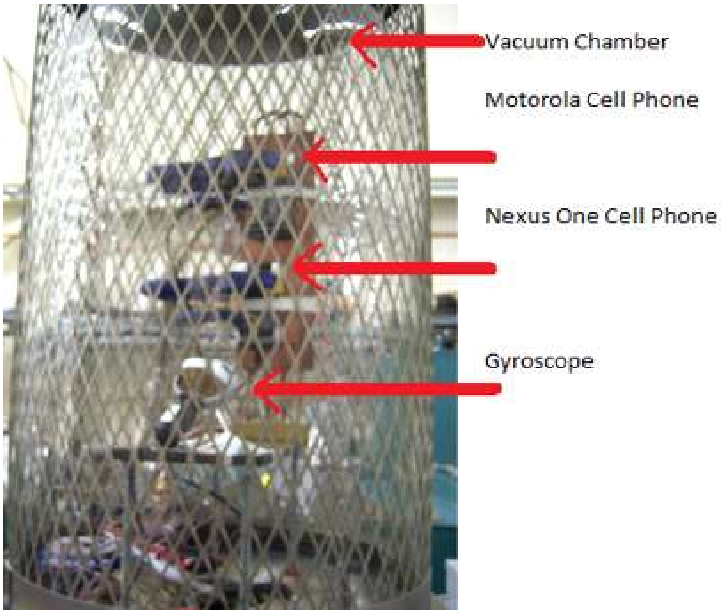


Figure 9 Vacuum Test chamber

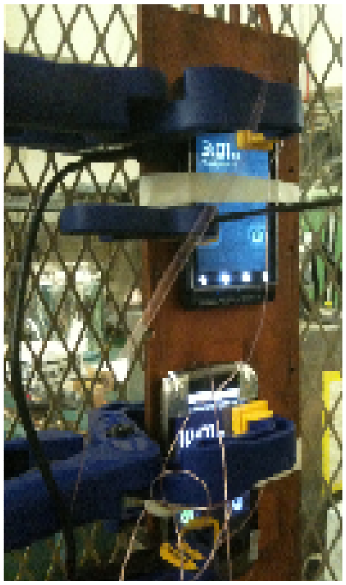


Figure 10 Nexus One and Motorola cell phones inside vacuum chamber

The gyroscope was tested at different speeds, paying attention to possible delays in the starting and stopping times, without showing changes.

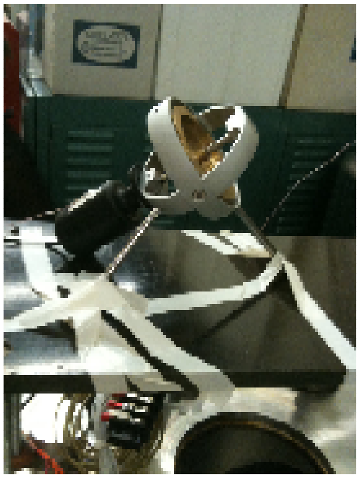


Figure 11 Gyroscope ready for the vacuum test

A graph of the temperature changes during the test can be seen in the Figure 12, which shows the temperature levels recorded by the thermo sensors connected to the cell phones and the gyroscopes. The thermo sensors were connected in the areas where the main electronics and sensors of the Nexus One are located.

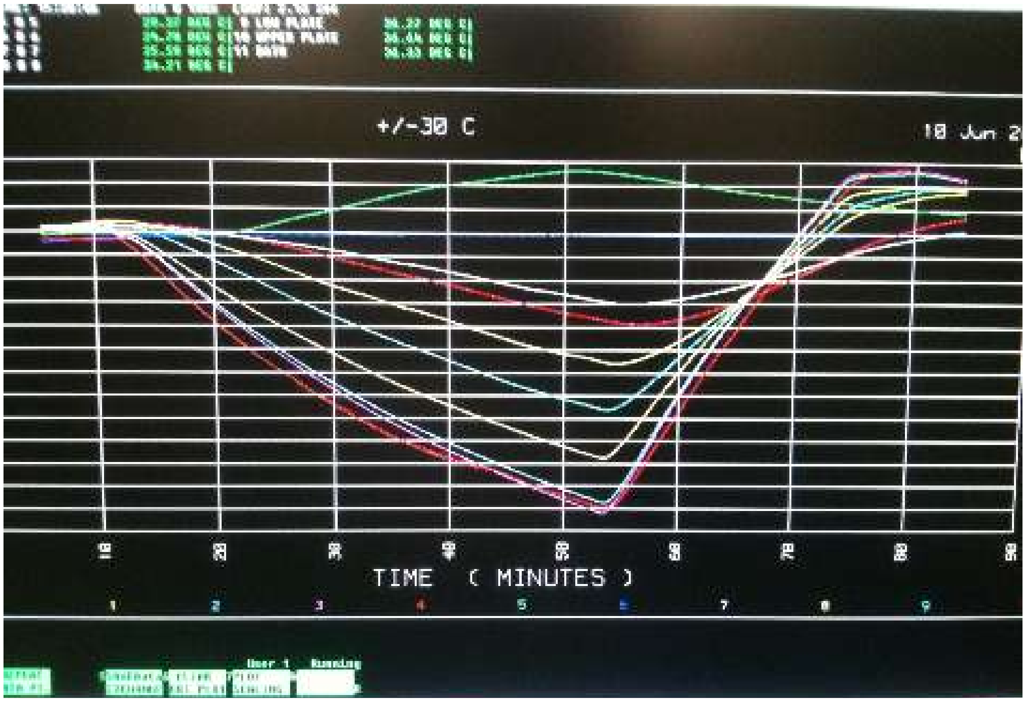


Figure 12 Graph of temperature change

3.2 FUTURE TEST IN SPACE ENVIRONMENT

It's necessary to perform additional tests at Ames facilities to ensure the success in future launches.

Extended vibration and thermal-vacuum tests with cycles up to 24 hrs are planned to be executed in the following weeks. For the vibration test, I developed a test plan which requires going up to 2 G for 10 minutes, then increasing to 5 G for another 10 min, and finally again to 2G for 10 more minutes. We know that the cell phone gets data at 2G, but seems to saturate if it experiences more acceleration. This condition was proved during the last launch. The objective is to test if the data that the Nexus One gets at 2G changes after having performed cycles at higher acceleration, or if this passage does not affect the functionality of the cell phone and its sensors. During these tests, the cell phone will be recording data from gps, accelerometer, magnetometer, and will be receiving data of an IMU through Bluetooth. The idea is to test everything what we have developed so far.

For the thermal-vacuum tests, a long duration test of 24 hours will be done. The maximum temperature suggested is 80 Celsius degrees; the minimum is 20, with 90 minutes cycles with 50 torr of pressure. To perform such test, a bigger vacuum chamber, with temperature range of up to 127 Celsius degree, was requested. An interesting part of this test will be to see if the Nexus One's Li-Ion battery will resists temperatures higher than 60 Celsius degrees, which, is the limit recommended by Google. For this test, the objective is to collect data from a cell phone and a gyroscope with communication through Bluetooth. The structure to put the elements into the vacuum chamber is the one planned to be used for the first launch and discarded for dimensional constraints. The structure could be attached to the chamber test with double side tape, and its dimensions are 15 x 12 x 11 cm. A picture of this structure is shown in figure 13.

Figure 13 Structure for test a gyroscope and two cell phones

I have been in contact with Ames employees, who had access to a radiation chamber last summer, to arrange a radiation test for our components. The standard for space qualification is resisting more than 96 meV-cm^2/mg, and the ideal goal would be to prove if the Nexus One survives to this limit. While still waiting to get more information about the availability and capability of the radiation chamber, the test is now designed to go first up to 40 meV, then to 60 meV, and finally to 96 meV for a couple of hours, if that is possible. Data about the size and dimensions of the Nexus One was sent to the personnel in charge of the chambers.

These tests were scheduled to be done during the last week of July but have been rescheduled to the end of August or beginning of September to address more urgent priorities.

Chapter 4: SUBORBITAL OPPORTUNITIES

4.1 OPPORTUNITY TO TEST OUR HARDWARE

The plan for testing the hardware platform on a low altitude sounding rocket was initially planned for August 7, 2010. Regardless the initial plan, an important boost in the progress of the PhoneSat development came with the sudden availability of a similar flight opportunity in July, 2010. With just a week of preparation, we prepared two sounding rocket flight, which produced enough data and to mark an important stepping stone for the project. Overnight, the project started to get attention and to be considered more serious. We had the chance to strengthen the relationship with external groups and companies, suddenly interested in the PhoneSat project.

On the weekend of the 23 and 24 of July, at the Black Rock Desert in Nevada, the Rocket Mavericks launch event was celebrated with the participation of the PhoneSat team. Initiative, creativity and ingenuity were profused in order to fix the payload on two different rockets: in fact we didn’t knew any detail about size and volume of the available space in the payload bay until two days before our journey to the desert. These flights were organized by a group of rocket amateurs, who are getting a lot of rocketry experience.

These flights were important for the PhoneSat project because they allowed testing g-forces, temperature changes and hardware survivability, with all the attention on our

Nexus One and its sensors. Both flights were an extraordinary success for the PhoneSat development, since we were able to get back data to analyze. This experience allowed us to work with the major amateur rocket community and to start building relationships between them and Ames. And since one of the future goals of the PhoneSat project is to develop new partnerships and increase participatory exploration, this event was very helpful.

4.2 FIRST FLIGHT JULY 23, 2010

On the 23rd of July, a test flight was planned to launch two cell phones and an IMU with Bluetooth communication.

The scheduled flight was planned to happen on a Beagle VI rocket. This rocket was the twin rocket of the Sony Rocket, which was developed by the Mavericks Civilian Space Foundation through a partnership with Sony and Intel, under the Mavericks' education program. This education program gave to a group of students the opportunity to be involved in the design, construction, and development and test flight of a real rocket. The Mavericks Foundation is getting experience in sounding rocket research, while strengthening associations with private rocket organizations, universities and research institutions. In this context, they offered us the opportunity to flight our hardware in this rocket for free.

As we arrived to the launch site, one day before the launch, we had the opportunity to see the payload integration of the first Sony rocket. This visit helped us to figure out the logistics for the flight planned for the next day.

An assessment about testing a gyroscope, one Nexus One for data collection of accelerometers, gyroscope, magnetometer, GPS, and picture shooting, as well as the collection of an IMU information was done for both flights. An additional Nexus One was also prepared for video recording. The gyroscope test was complicated by the little information available about the possible impact of the gyroscope on the rocket's attitude control.

The payload bay volume considered during the preparation of our payload was a cone of six inches diameter and six inches high. The mass of our payload did not pose real constraints. A window on one of the sides of the rocket's wall was considered to be good enough to allow taking pictures. . Base on specifications, an acrylic structure was done by Benjamin Howard. The structure has been described in Chapter 3. In the end, this structure was not used for the flight, because the actual space available inside the payload bay was by far more restricted.

For this flight our primary goal was not to jeopardize the entire flight mission, which had a primary payload. For this reason, we decided not to run the Bluetooth test for the IMU data collection. The space available for our payload allowed us to flight just a single Nexus One. We were expecting to have the opportunity to accommodate the Nexus One in front of the rocket window, but it was not possible since electronics and other systems needed to follow a specific layout. We were expecting to be allowed to drill a hole on the surface of the rocket structure, in order to allow the Nexus to take pictures of the exterior. The hole was considered impractical because of complications with the integration of the primary payload, so we were left with no other choice than taking pictures from the inside.

The program I wrote for the Nexus One did not have an automatic way to start and stop execution, so it was necessary to start the program execution before closing the payload bay. Since we experienced delays between the payload integration, the rocket integration into the launch pad and the ignition, we were worried about the duration of the battery charge. We were unsure if the battery could last in case the program needed to be started more than three hours before the launch. Ironically, even if the test was designed to demonstrate that the Lithium-Ion battery of the Nexus One was capable to work fine in attitudes, temperatures and vibration conditions of a launch, we still had to ensure that the battery had enough capacity for the test itself. For this reason, external 5V batteries were bought to be connected through mini USB to the Nexus One.

The final configuration for the first test flight was a single Nexus One cell phone, reading GPS, magnetometer, accelerometers, and gyroscope data from its internal sensors, and saving them into its SDHC memory card for post-analysis. At the same time, the Nexus One was taking pictures every five seconds. Unfortunately, due to the reasons described above, the pictures were all of the interior area of the payload bay. The Nexus One and the external battery connected through USB were attached to the table of the payload bay, just next to the primary payload's electronics.

The 29 feet, 1100 pounds rocket was aimed to reach an altitude of 220,000 feet, in approximately 18 seconds, under vibration loads of 12-15 g's, reaching mach 1.1 in the descent, deploying a parachute to allow a safe recovery of the payloads.

The two stages rocket accomplished the separation of the first stage and the ignition of the second, experiencing some problems for a hole in the rocket's wall and for a overheat exposure during the ascending. These problems caused the premature deployment of parachutes, and influenced the guidance sensors of the second stage, reducing the speed and the maximum altitude that reachable by the rocket. As a consequence, the descend phase of the rocket did not have a parachute, and the payloads crashed into the ground at Mach 2.8, at a distance no more than two miles away from the launch pad. Regardless the parachutes problem, the launch was considered a success by Sony and the Mavericks Foundation, since it proved that the rocket could be launched by a Sony laptop, main goal of their mission. The Mavericks Foundation will keep working to get space qualification of their parachutes and rocket components.

The Nexus One was recovered as a single piece, while the electronics of the primary payload was completely destroyed. Even if the Nexus One was in a single piece, and still attached to the payload table when recovered, the display was entirely broken. The Google cell phone was a little bit twisted, a consequence of a shock that the entire payload table had after the crash, and for that reason, the back cover and battery were out of place. After analyzing the cell phone, we saw in the upper right corner of the display a mark were the display was hit by a screw coming from the other side of the payload table. During the crash, the screw impacted the display of the Nexus One and the shock wave propagated through the display, breaking it all (Figure 14).

Figure 14 Nexus One after ballistic re-entry of the first launch

We recovered the SDHD memory card in excellent conditions. This way we were then able to analyze the flight data being in this case, the only successful payload of this launch.

With data coming from the Nexus One accelerometer, I plotted an acceleration graph of the entire record. The most interesting part starts right before the ignition, and goes on until the last record available. The Nexus One recorded one sample every two seconds, since that was how it was programmed to do. We noticed that we needed more accurate information through the launch time, so the sample rate for the second launch was increased to 60 Hz.

Small size graphs about the launch data are displayed in Figures 15, 16 and 17. Full quality graphs can be found in the appendix section.

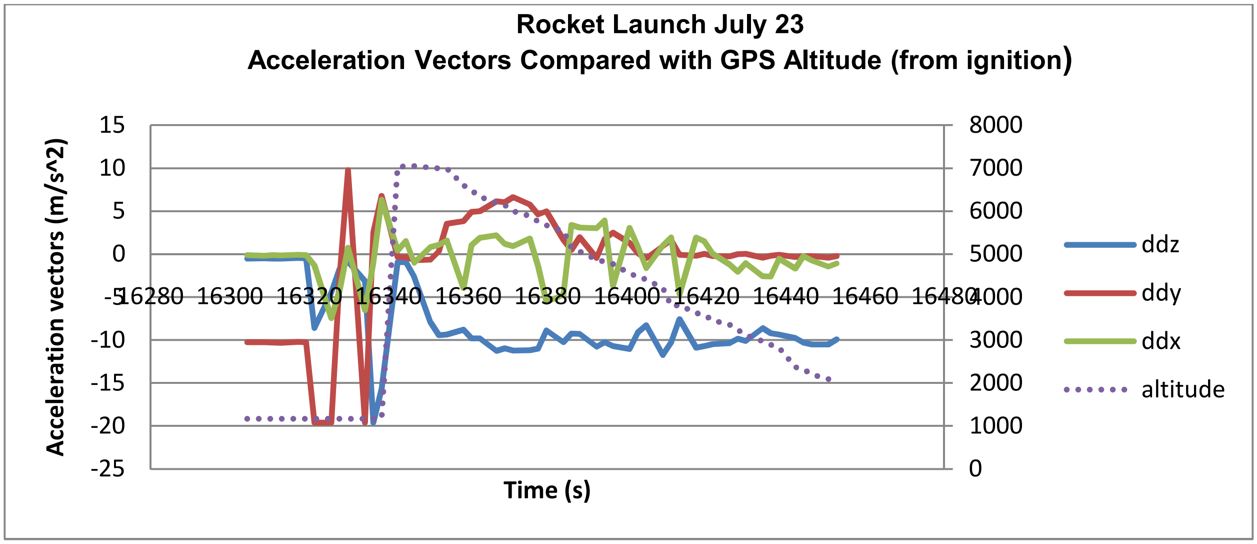


Figure 15 Acceleration Vectors compared with GPS Altitude - Launch July 23

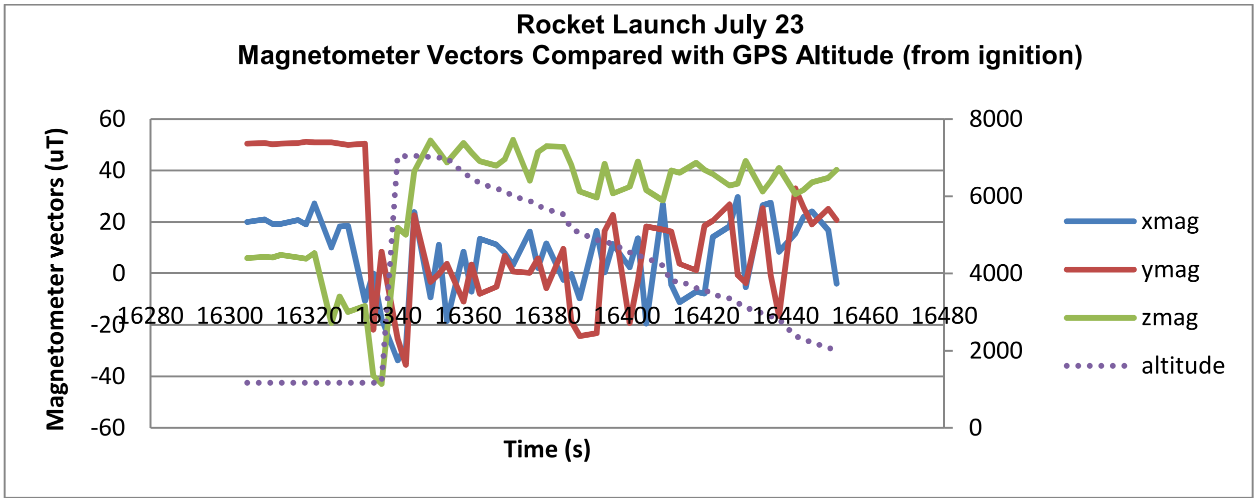


Figure 16 Magnetometer Vectors Compared with GPS Altitude - Launch July 23

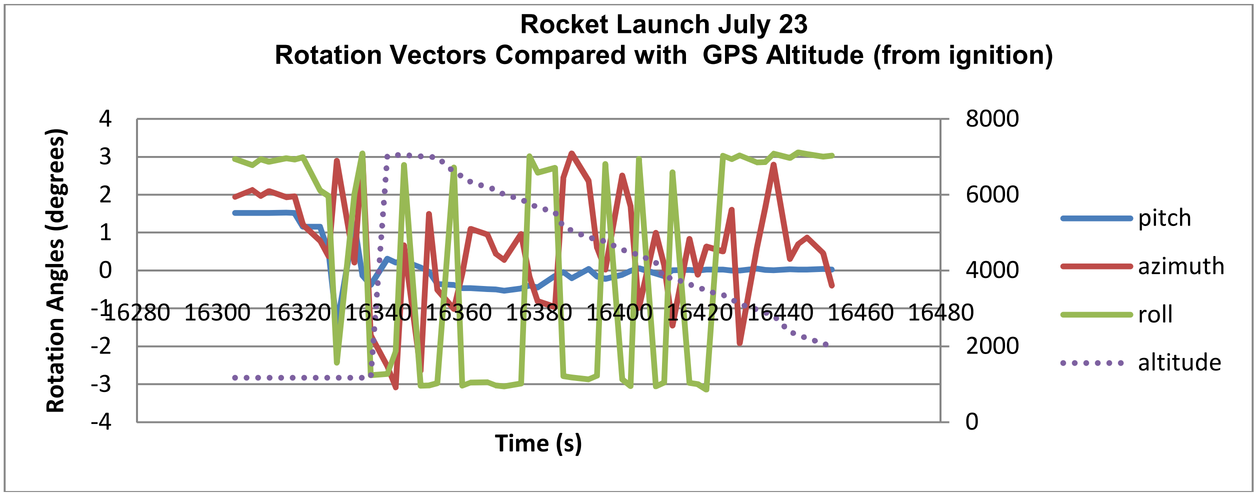
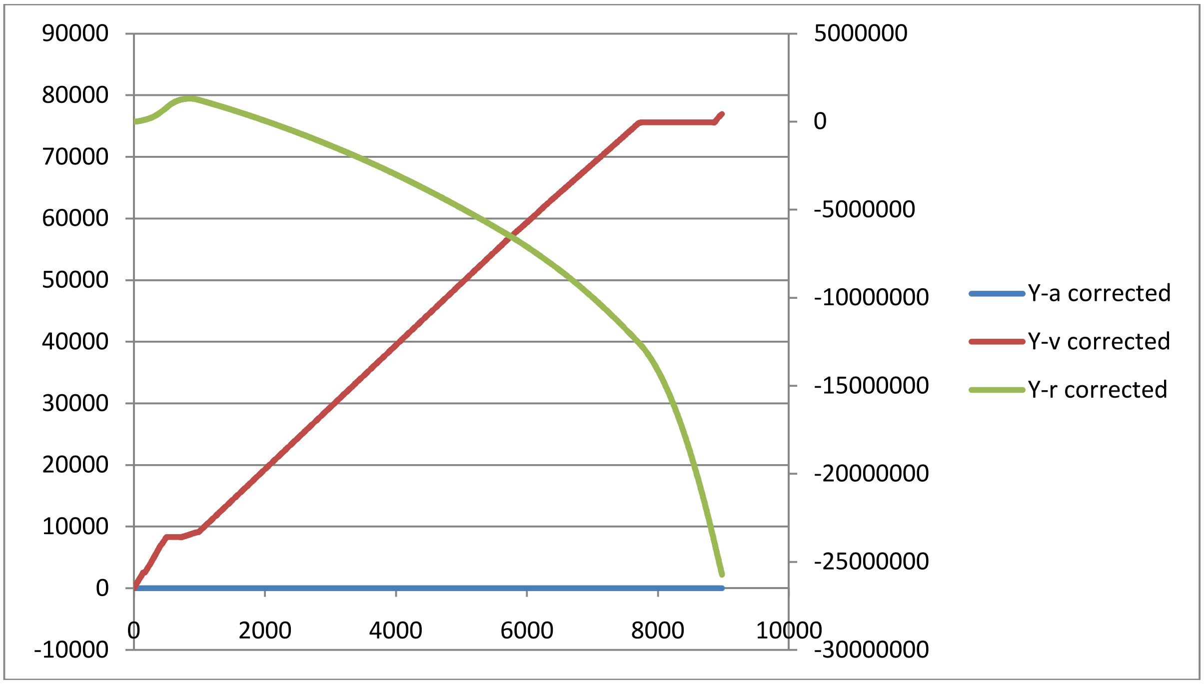


Figure 17 Rotation Vectors Compared with GPS Altitude - Launch July 23

These graphs allow understanding the functionality of the Nexus. According to Figure 15, the Nexus One accelerometer saturates at 2 G. The graph shows that even in rest position, as in the launch pad, the software was recording a -10 m/s^2 acceleration in the vertical axis. Since the recording rate was very low, the changes during the launch could not be fully appreciated.

The maximum altitude recorded by the GPS was around 7,000 meters; on the acceleration graph that point matches with the acceleration values in a maximum point, as zero or close to zero. The data recorded by the GPS sensor in the descending phase shows some points of sudden decrease, this phenomenon is probably a consequence of the delay in the GPS refresh, or a missing sample due to the data rate.



With this flight we proved that the Nexus One resists launching vibrations, to high temperatures, and that its structure is strong enough to survive a high load impact. The data says that the accelerometers can't record values higher than 2Gs, and that the Nexus One magnetometer should be replaced in the PhoneSat final design.

4.3 SECOND FLIGHT JULY 24, 2010

The second flight on July 24, 2010, was a dedicated flight for our payload. The rocket was the James Dougherty's Intimidator-5, with a Cesaroni thermoplastic N4100 motor with 1000 lbs of thrust, reaching an altitude of 28, 000 feet. This particular rocket had successful previous flights, and that gave us more confidence about the recovery of our payload in excellent conditions.

Because it was a dedicated flight, we had the opportunity to be totally involved in the systems integration and rocket integration. The payload designated for this flight was: two Nexus Ones and an IMU. One Nexus cell phone was dedicated for receiving data sensors, and gps, at the same time of receiving the IMU data through Bluetooth, interfacing with an arduino. In this case, we didn't take pictures of the flight. The second Nexus One was dedicated to take video of the flight. For this purpose, we drilled a hole in the wall of the rocket, and the Nexus One camera was aligned to it, while attached to the rocket wall.

For this flight, we received the section of the payload area and nosecone of the rocket, three days before the flight, so we had more time to accommodate our hardware properly. There was a try to accommodate the gyroscope in the nosecone section for test in flight, but there was impossible to fit the Nexus One and the motor in a safe way. The decision to don't test the gyroscope was done for this reason.

The tested hardware was attached to the payload table, while both Nexus One were attached to the wall, while protected in between with foam in order to reduce the probability of damping during flight or landing. No external batteries were attached to the cell phones because 3 hours were enough time to finish the flight and recovery. The hardware was sharing the payload area with the rocket's electronics and its batteries, which were in the other side of the payload table. The programs in both cell phones were started before closing the payload area.

The video program keeps recording as long as you let the application run, and it stops at the moment you touch one of the buttons in the screen. Because of this, the screen didn't have a lock and a constant concern about if the video was running or not was present.

While putting together the different stages of the rocket and accommodating the parachute, we noticed that the sections where difficult to separate. This was a serious problem, because if the stages couldn't separate, the parachute wouldn't be deployed. The reason why this happened, was because the heat experienced in the desert that days was so high in temperature, that the material of the rocket' structure was expanded, so the separation mechanism couldn't work properly. A couple of sand sheets to correct the surface of the rocket structure were necessary. This was an interesting experience, because teach us that contingency can happen in the last moment, if the complete configuration is not tested in advance.

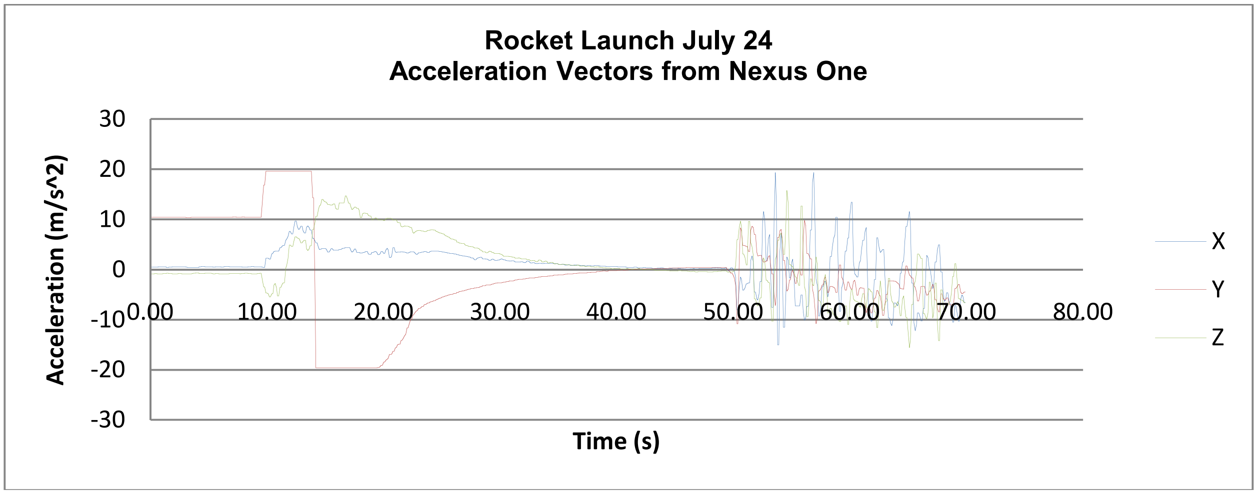
The rocket was successfully launched and, the payload was recovered in perfect conditions, after a perfect flight. According with the data we got from the rocket's electronics, it reached an altitude of 8,111 meters, at mach 1.74, and with maximum acceleration value of 180 m/sec.

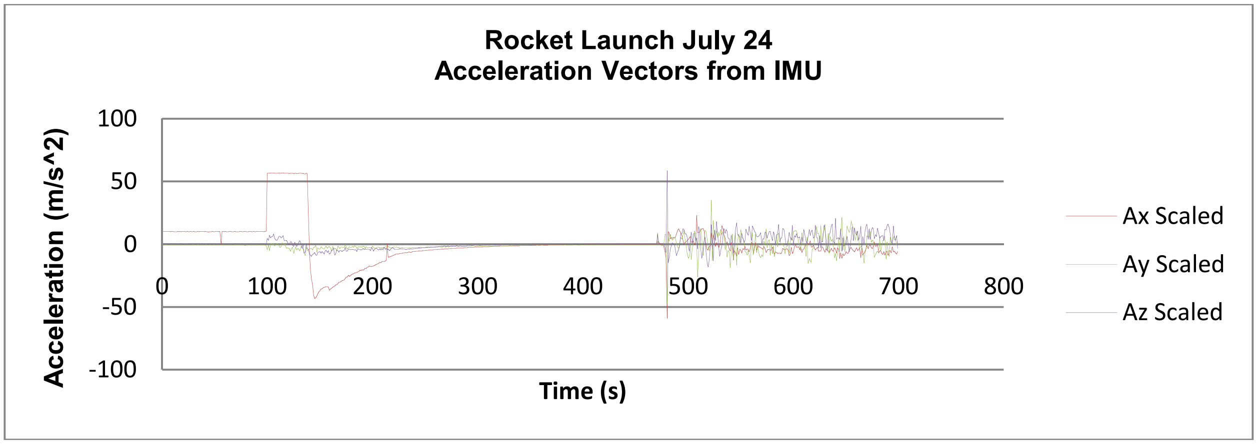
After opening the payload bay, we noticed that the video recording kept going the entire flight, recording 2.5 hours of video. The program that was recording the sensors and IMU data was not running anymore. After getting the data files out of the SDcards, we plot the values to analyze the development of the flight.

The data files we got were, the values of accelerometer, magnetometer, gyroscope and GPS from the Nexus One at a sampling rate of 60 Hz, at the same time we got the data from the IMU about its magnetometer, accelerometer, gyroscope and clock. The video that we got, shows an amazing movie of the entire flight, which at the end, we could match with the data graphs.

After analyzing the video, we noticed that the audio stopped after one hour and eleven minutes, exactly at the moment we were transporting the rocket from the car to the launch pad. In the audio graph signal it seemed that there was an interference that caused the audio to stop. Trying to find what could produce this, I could find any interference source as responsible for this. I spend time analyzing the videos that I recorded for pre-test flight, and I noticed that they were also missing audio, exactly at the same time the rocket video did. More debug to the video application code should be done to fix that.

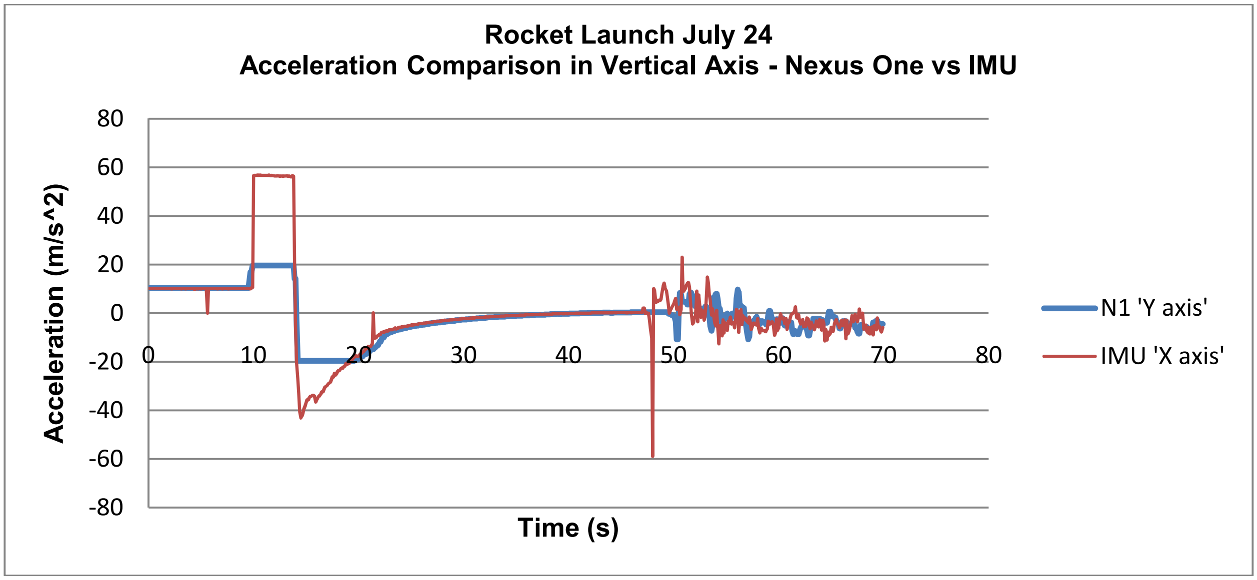
From the IMU, Nexus One sensors and GPS data we plot the graphs in figures 19, 20 and 21. Full size graphs can be found in the Appendix section.





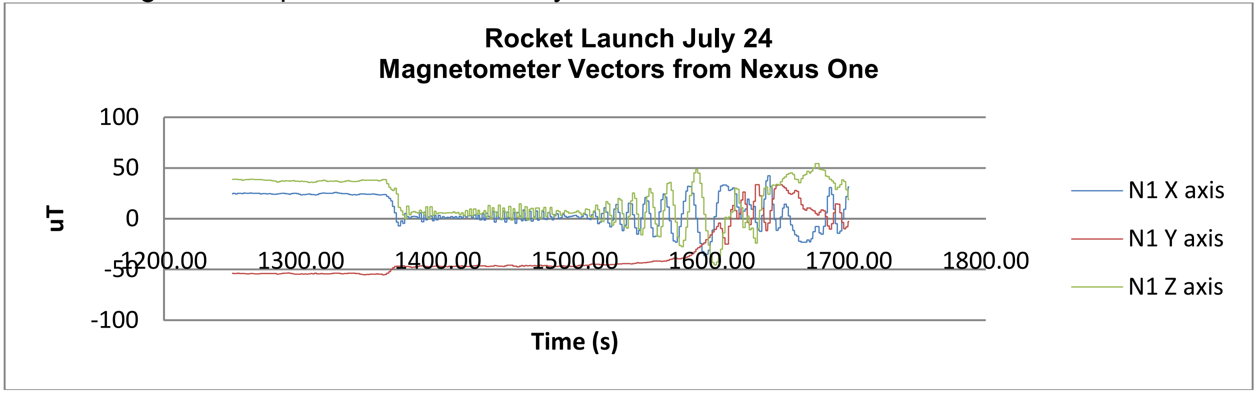
The two acceleration graphs, one from the Nexus One sensors, and the other one from the IMU, match in shape during time. The biggest difference is noticed in the saturation level of the vertical axis, where the Nexus One saturates at 2G, and the IMU saturates at 5Gs.

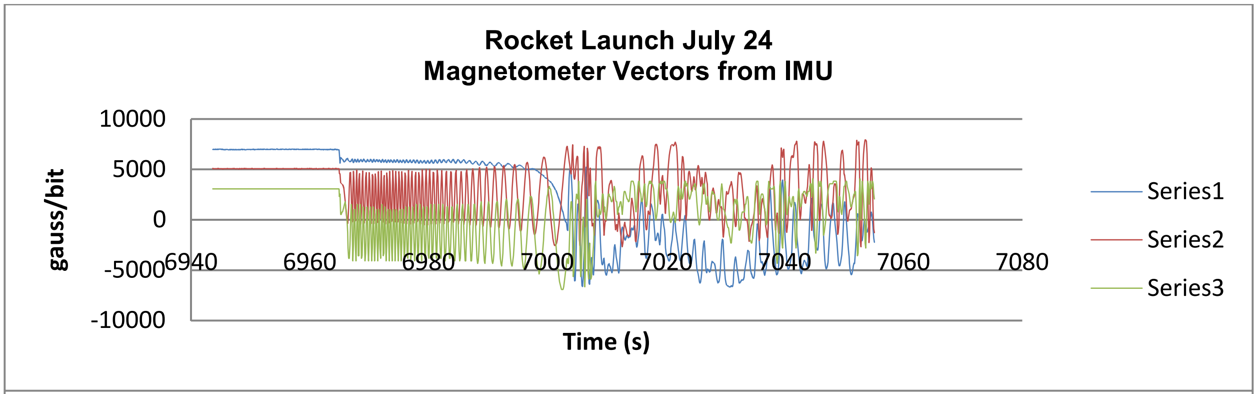
A comparison between accelerometer data in the vertical axis, between the Nexus One and the IMU is shown in the next Figure.

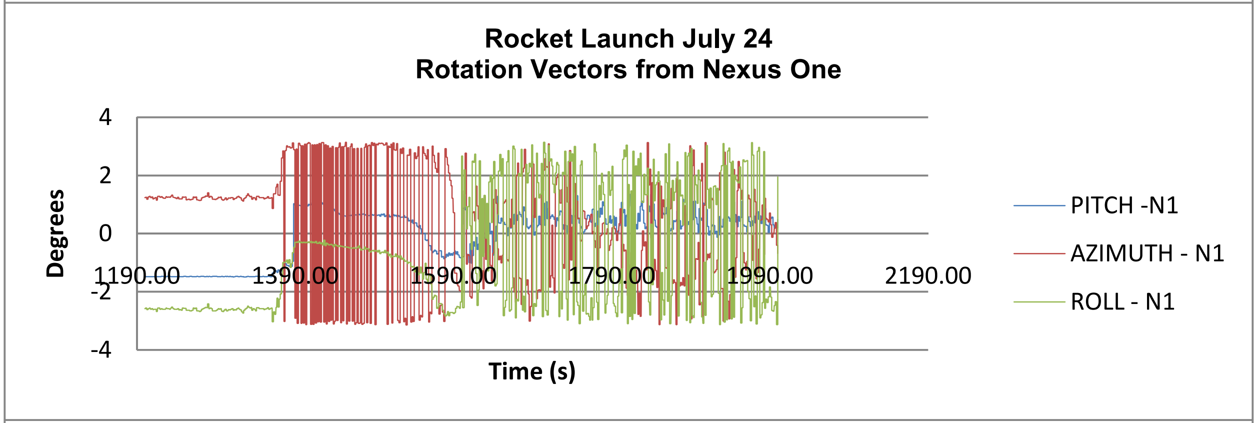


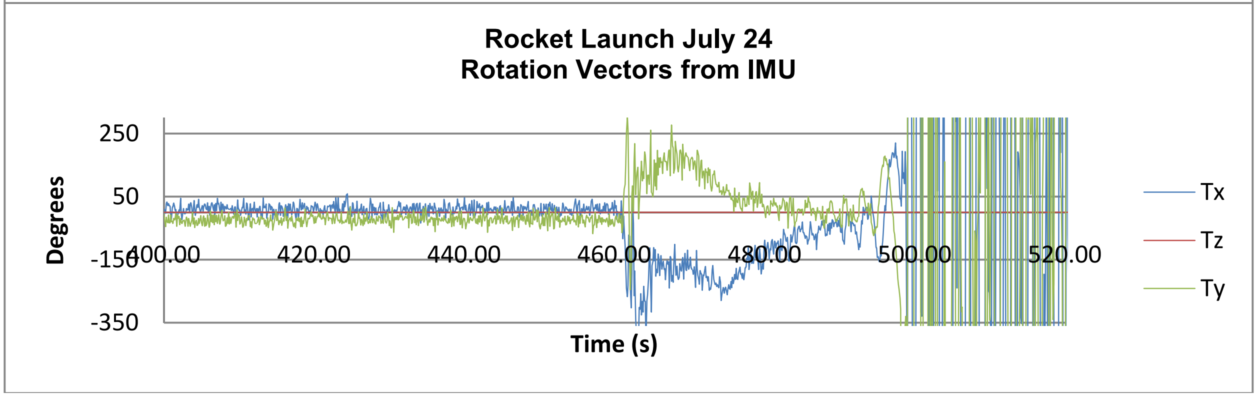
The graph starts just right before the ignition at 1 G, where the rocket was in rest position on the launch pad. At the ignition moment, the difference of saturation levels in Gs, is visible in the first frame of the graph, being at 2Gs for the Nexus One and at 5 G for the IMU, and that is the reason why in that section of the graph the lines are flat. Even if the rocket experienced 18.36 G, the maximum information we could ever got with the used sensors will be 2 and 5Gs, missing important information in that time frame. Following that, gravity des acceleration is seen, while the acceleration starts decreasing with altitude. Around second 47, the difference in acceleration is noticed as the parachute was deployed. This also match with the rocket flight data that we got form James Dougherty, the owner of the rocket, in where the time to apogee is marked as 37.4 sec.

Analyzing the magnetometer and rotation data we got, it’s clearly how the IMU recorded smother values changes, than the Nexus One. We can notice sine waves at a higher frequency during the ignition and ascending phase, which matches with the fact that the rocket keeps rotating all its way up, so the difference between ascending and descending with the parachute it is clearly different.

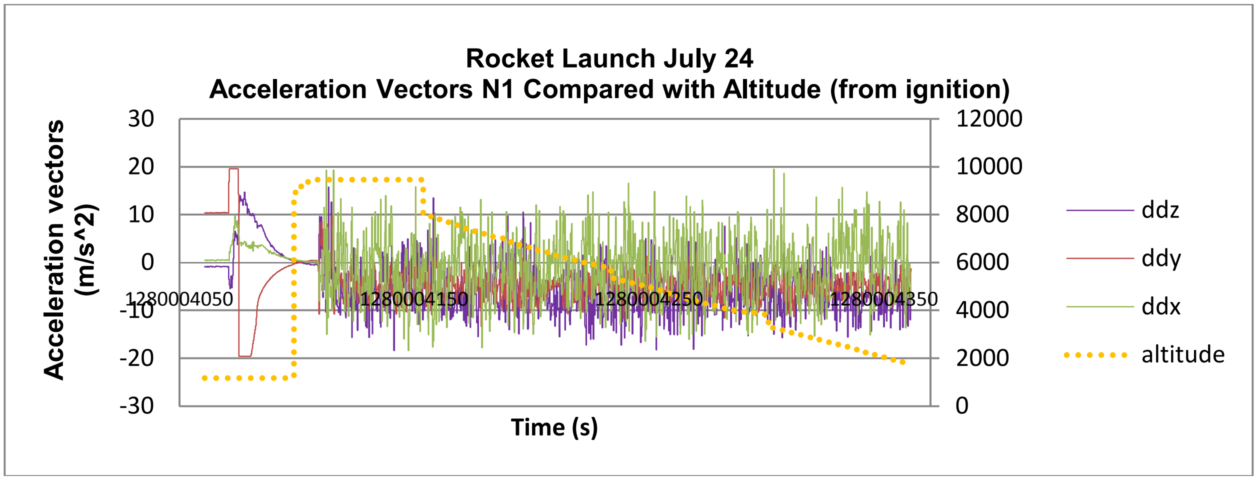






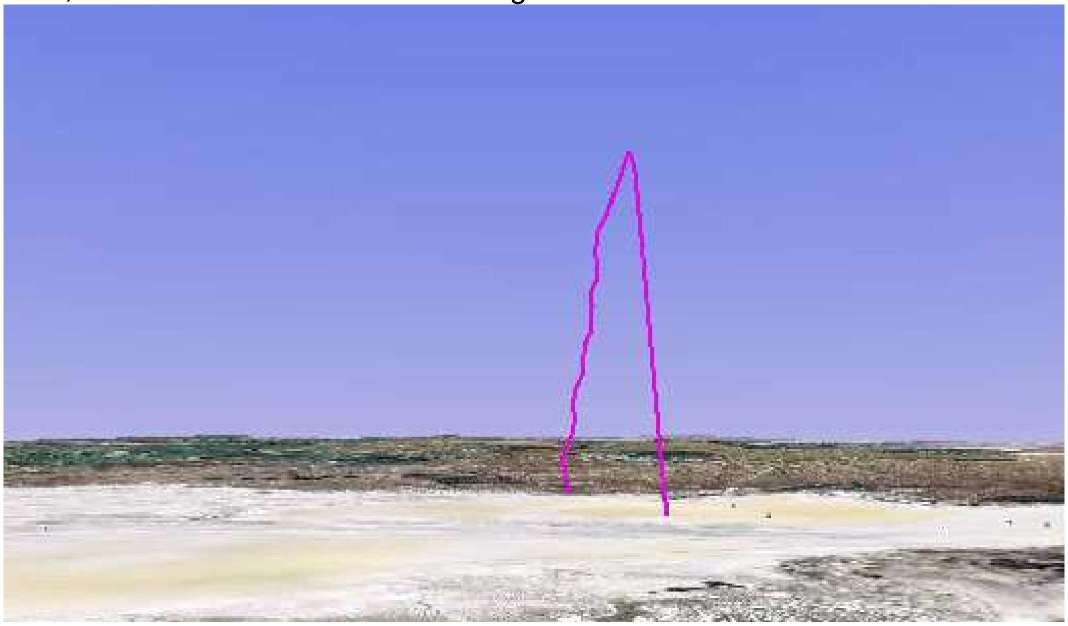


From the Nexus One's GPS, a graph was plotted to compare altitude recorded versus acceleration can be seen in the next Figure, where the maximum altitude matches with the point where acceleration is zero.



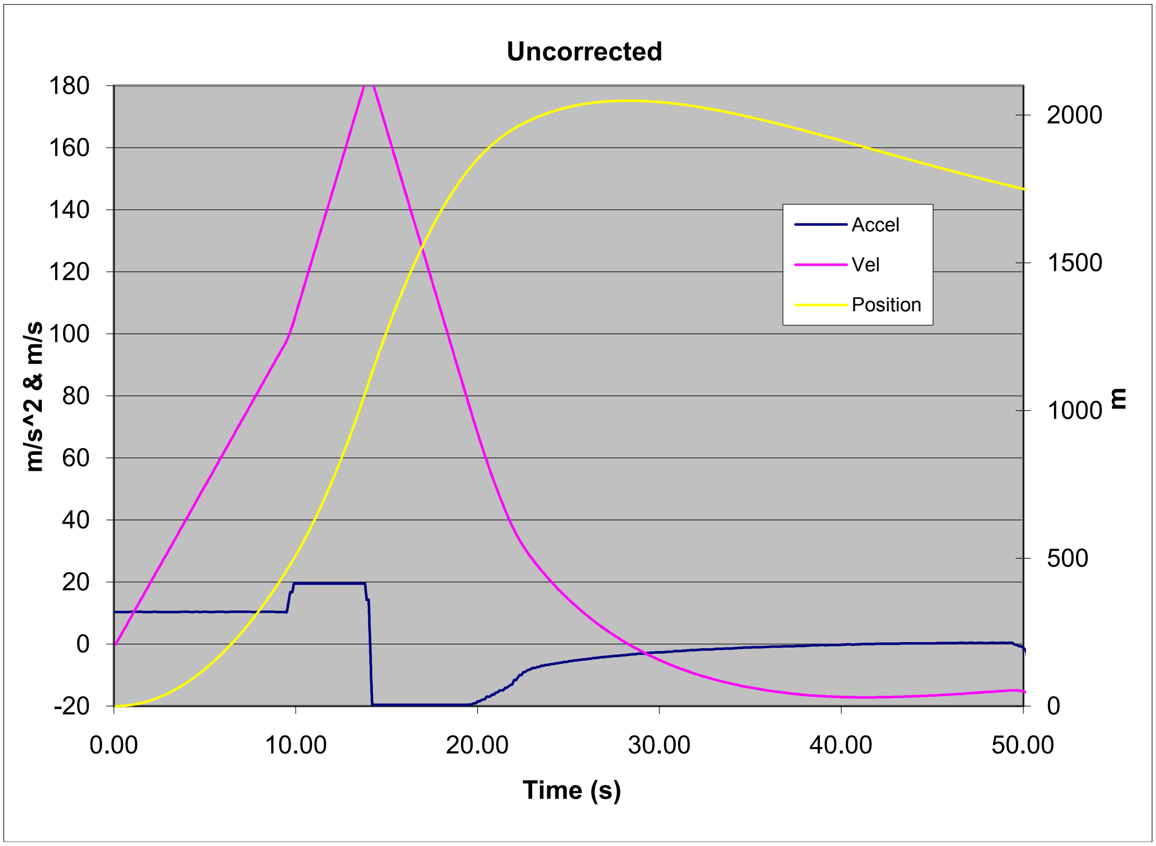
The maximum altitude recorded by the Nexus One was 9,457 meters which is higher than the one recorded by the rocket's electronics, which was 8,111 meters. The rocket got its altitude through barometric way, and probably that is why we notice a difference between maximum altitudes recorded. The data points from the rocket's electronics are not available to be plotted and so, to do a single graph for comparison purposes between our own data.

Taking advantage of Google Earth, and the GPS data we got, a trajectory diagram was plotted, which can be seen in the next figure:

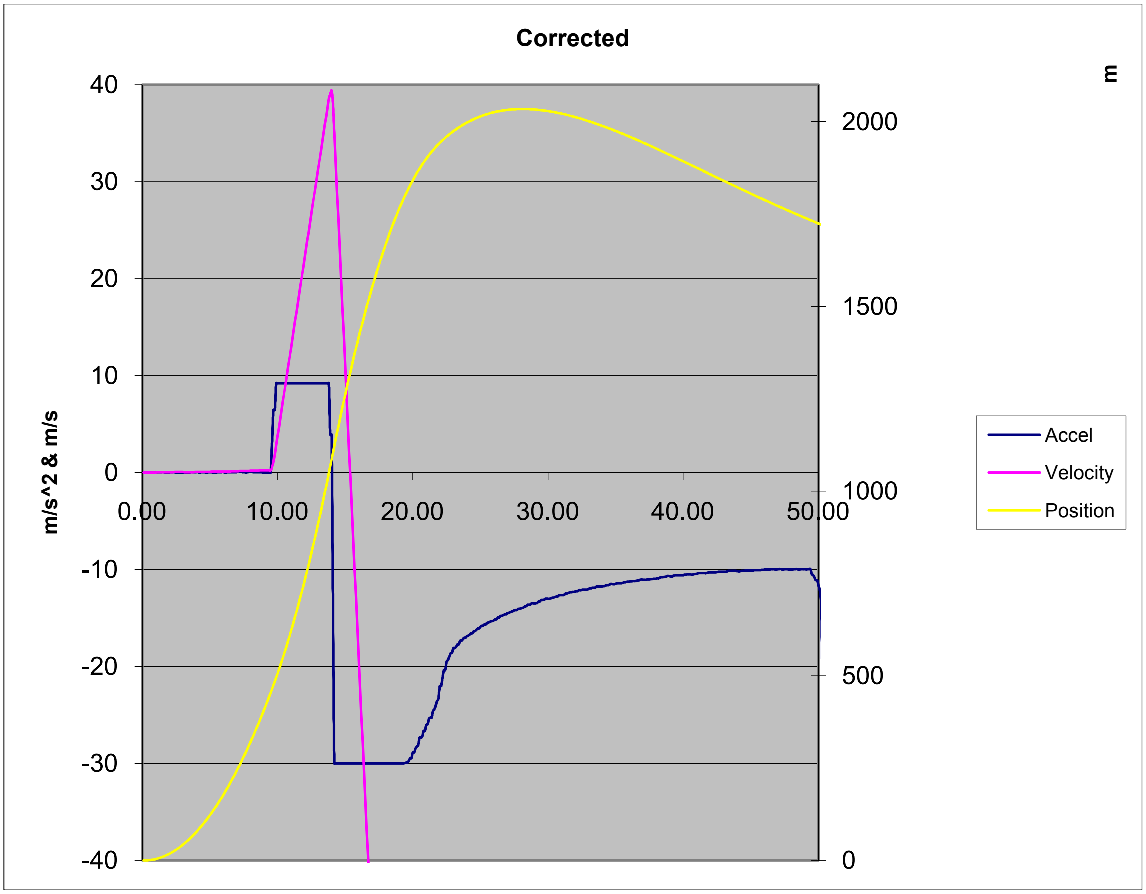


With the acceleration data from the Nexus One and IMU, a velocity and displacement graph was processed. Without modifying the initial value of the acceleration, to zero, the graph shows a maximum velocity of 180 m/s, with a maximum altitude around

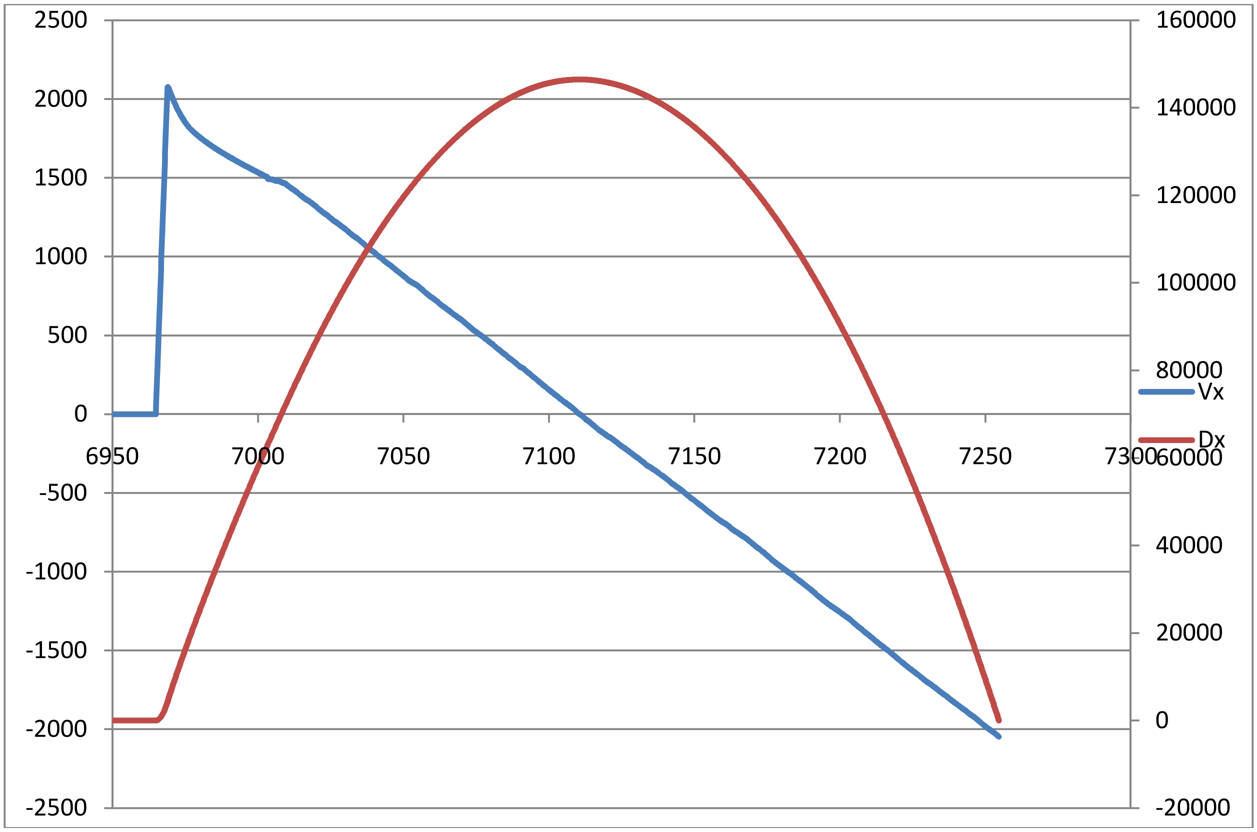
2, 046 meters, which is clearly wrong. Even if the final maximum altitude is wrong, it matches with the acceleration value point of zero.



If I correct the initial values of the acceleration, to zero the plotted graph is the next:



From the acceleration values of the IMU data, the displacement and velocity graph is the next one:



From these graphs, we can conclude that from the accelerometer values of the Nexus One, the velocity that is recording is accurate enough, but the altitude reached complete wrong. The data points that we are missing because of the saturation at 2 Gs, avoid us to get an accurate graph, to analyze the total velocities and final path of the rocket.

After these flight tests, we understood that the Nexus One hardware is resistant enough to survive launches and soft landings, as well as the Li-Ion battery. The electronics inside are recording the data fast enough, but a big gap is presented because of the saturation level of the accelerometers at 2 G. The magnetometers are clearly not good enough if we compared them with the data we can get from the IMU.

Chapter 5: FUTURE DEVELOPMENT

5.1 Improvements in PhoneSat Systems

Before any other flight tests, the subsystems of the PhoneSat should be working, if not perfect, at least should have an 80% of precision. To increase our progress, the 3 degrees of freedom for attitude control should be proven, as well as local wireless communication for data reception.

The software should be improved to have automatic starts and ending points, where we can have more battery life control over the hardware. The data process should be friendlier than the current one, to speed up the post-analysis of flights. The new structure and placement of the new gyroscopes should be tested, in order to have iterations and choose the best option of mass balance and not electronic interference.

For further developments of the PhoneSat project, modifications to the Nexus One could be done in order to reduce mass and space. One of them could be taking off the display, and the keyboard which are not necessary for our satellite, as well as other electronics that are not used. By removing the not used hardware, the mass goes down, and that is a positive point in launch cost. This idea should be studied as well, because of the impact that can have in the original idea of the project, which is to use commercial of the shelf components as the Nexus One, where all the electronics are embedded in one product, instead of asking for each of them to build a specific circuit board.

5.2 More Tests and Flight opportunities

Testing the hardware for radiation, vibration and thermal vacuum cycles should be done, as explained in Chapter 3.

In order to increase the readiness of the PhoneSat components, more flight tests should be done. An excellent test could be to flight the hardware in high altitude balloons, where we could have the opportunity to go up to 160,000 feet. If the PhoneSat components can keep working through a suborbital balloon flight, the readiness level of the Nexus One’s electronics might jump up to level 6 or 7.

The possibility of getting a flight opportunity in a sounding rocket going up to attitudes higher than 100 - 150 kilometers should be interesting to test.

5.3 Industry and student communities

Outline:
Google tech talk
Students interested (like SU)

Chapter 6: CONCLUSION

OUTLINE:

- Target – schools, young generation, inspire people to think out of the box
- Android enable community and schools to work on satellites basing their designs in their own cell phones – something new.
- If satellites can be built with a lower cost, the production number of them could increase, and the availability of them increases too. They become more affordable, and with that space technology became more accessible to the public.
- Giving another use to the technologies of private sectors.
- What about to have a unique platform to manage space exploration?
- It works in space like environment!

Let's change the big size components for miniaturization, expensive projects for cheaper ones, slow systems for faster ones, old technology for new one…

Let's inspire new generations, new groups of students, industries, and specially children to dream, making space more reachable.