CANSAT 7

Project Report

Submitted in partial fulfillment for the requirement of Ist Semester in Space Masters of University of Wuerzburg

Submitted by

Cano, Angel Mario-1543440

Forslund, Lars Anders-1543952

Hakenburg, Jan Philipp -1537190

Krishnamurthy, Narayanan-1543083

Wang, Hankang-1543524

Worracharoen, Pakasit-1543191

5th February 2007

Department of Informatics

UNIVERSITY OF WUERZBURG

CONTENTS

CHAPTER I

INTRODUCTION

1.1 CANSAT OBJECTIVE

The CanSat mission objectives are listed below in order of significance:

1. Collect and transmit atmospheric data.
2. Demonstrate command and control capability.
3. Show that an inexpensive design will survive launch loads.
4. Create prototype data telemetry system and signal design for future implementation and improvement.

The main objective of the initial CanSat project is to show that the design of a small, inexpensive data telemetry system is possible. The system will have both receiving and transmitting capabilities and will be incorporated with a complement of analog and digital sensors. Although the sensor data is important for verification of the telemetry objectives, achieving individual sensor precision is not a primary goal of this CanSat.

This report updates all technical details, which were stated in the previous milestones.

We put emphasis on recently added features. This is mainly

- Printed circuit board for both, cansat and groundstation
- Display of Satellites in View (concerns groundstation software)

The protocol has been explained in milestone 3. There have not been any modifications since then. In this document, we rather give an abstract overview on the communication flow.

1.2 STUDENT SATELLITE PROGRAM

Several universities have completed CanSat projects in the past, each with different missions.The University of Tokyo's project, Gekka-Beijing, consisted of three CanSats that attempted to rotate a satellite, gather temperature and pressure data, and to use a camera. They have also designed a satellite to test systems that will later be used in a Cube-Sat project.Other universities such as Arizona State University and Kyushu University have designed satellites to test a tracking system and to collect temperature data. Most of these projects are used simply to give students experience in a hands-on project. They must deal with ordering equipment, designing systems, making reports, and working in a group. This will give them an idea of what to expect when they enter the work force.

1.3 MISSION REQUIREMENT

The CanSat project is about designing a device that measures temperature, pressure and global position. The device should weigh no more than five hundred grams and have a volume no bigger than five hundred millilitres. It should be able to telecommunicate with a groundstation, sending information to a computer in order for the signal data to be processed and presented in a user interface.

This project spans over fields including telecommunication, electronics and computer programming. Moreover, it will require skills in project planning and working as a group.

1.4 ANALYSIS & DESIGN

In the second milestone, we take a closer look at the functionality of the different subsystems of the CanSat – the structure, electronics and software. We also give a detailed description of the various subsystem tests we have performed and will perform in the future. Following the fixed timeline, this report also aims to evaluate our progress so far.

We have devised advanced approaches and technology to meet the mission requirements. For instance, our protocol is highly sophisticated. In fact, the final protocol is so sophisticated that we had to invent a simpler protocol to test and debug our devices.

We summarize what we have accomplished in milestone two:

- The cansat structure is manufactured, and is now being optimized in terms of weight, performace, material characteristics, and overall stability.
- The groundstation circuit works and can be configured in running time by the groundstation software.
- The groundstation software runs stably and already meets all the requirements, for instance the graphical display and the file input output. In a later section we point out the most relevant code, which handles the input stream.
- The cansat circuit is functioning. We have already programmed the cansat to send pressure data to the groundstation.

1.5 VALIDATION AND TESTING

The cansat is to measure atmospheric data. Therefore, on the cansat there are temperature sensors, and pressure sensors installed. In addition, GPS information is available, but also acceleration.

In this milestone, we discuss the accuracy of the sensor data, that reaches the microcontroller. We mainly distinguish two sources of error: Virtually, any measuring device comes with an inherent deviation from the accurate value. Usually, this type of tolerance is described in the respective datasheet. But also, the conversion of the analog signal (in case of pressure, and accelerometer data) depends on the state of the microcontroller.

The second part of this Milestone is all about testing, and making sure, the cansat fulfills the mission requirements:

- Collecting and transmitting athmospheric data
- data storage in case of communication failure, and retransmission later on
- receiving and reacting upon commands from the groundstation
- graphical display of the data in real time

<u>All of our tests reveil, that we meet (and overshoot) the requirements. All systems are running properly.</u>
Our cansat is able to record data from the last 1 min 45 sec. Cansat provides the groundstation, with the remaining current supply voltage, which is important, when operated by 9V battery. As another highlight, we have uplink from groundstation, to remote control various subsystems of the cansat. For instance, we can activate and deactivate the accelerometer.

We begin devising and performing tests for each sensor separately. Whenever possible, we crosscheck our results with a reference source, such as a thermometer. After this, we have calibrated each sensor the best we can.

However, we also perform several complete system tests. For instance, we will place our cansat inside a freezer (of a fridge). This checks recovery from communication failure, but also the outside temperature sensor. Another important test is to carry the cansat away from the groundstation, to analyse the average performance.

For performing a system test, the different criteria to monitor are:

- Data flow
- Communication errors
- Sensor accuracy and rate of transmission
- Effects of vibrations, and medium shocks

In particular, we have taken the opportunity given by the robotics lab, to place our cansat on the remote controlled 4-wheel drive off-road robot. The test was successful, in that the hardware (and software) was not influenced by the vibrations cause by driving.

CHAPTER II

REVIEW OF LITERATURE

Of course to accomplish the cansat project, we have to read a lot of literature. Everybody has the bible and Shakespear for good night lecture. Also we get inspiration from computer games. Seriously, modern state of the art computer software has motivated to design the java program as we did.

The complete grid layout and operational mode of the groundstation mission surveillance is carried out by some javax swing model. Without literature, we would not have found out about javax swing buttons and toggle buttons.

Also in the Spacecraft design systems handbook from nasa, we have found useful hint for building small light weight satellites. Unfortunately we were not able to include a solar panel, because we don’t have enough resources and knowledge to include a solar cell without breaking the cell and not overcharging the unchargeable battery.

In conclusion, the books we took into consideration are also very expensive, so that we mainly had to take copies of the relevant pages from the library. The woman from the library is nice, although she is not extraordinarily pretty, like the Swedish girls. Everybody is looking forward to go to Sweden, because the literature is not so expensive.

Also, we are excited to find out about the prices of the computer games in Scandinavia. We will reach the place if the airplanes are not caught by terrorists.

Also in the Spacecraft design systems handbook from nasa, we have found useful hint for building small light weight satellites. Unfortunately we were not able to include a solar panel, because we don’t have enough resources and knowledge to include a solar cell without breaking the cell and not overcharging the unchargeable battery.

CHAPTER III

PROJECT MANAGEMENT:

3.1. INTRODUCTION

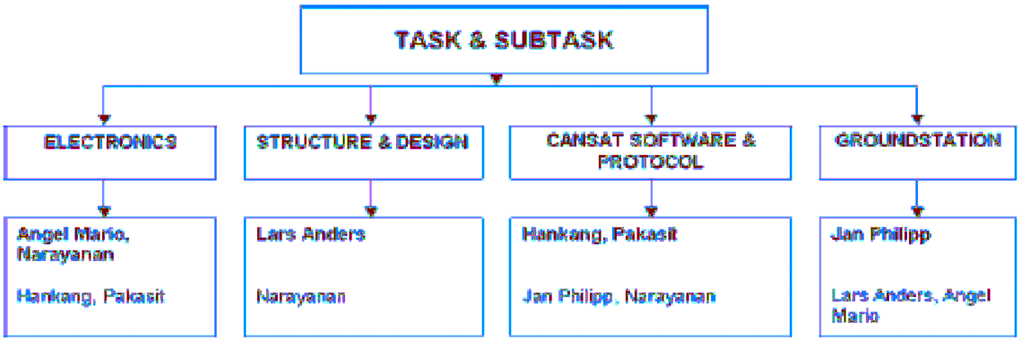
As a part of the project management, we implement the idea of distributed management. Our team does not have a designated leader. However, the figure below indicates the responsibilities of each squad member.

To handle the workload required for this project, the responsibilities were split between the team members. Jan was in charge of the ground station software and Anders was in charge of the structures. Angel was in charge of handling the electronics components. Pakasit, Hanker and Narayanan were responsible for the onboard software and the protocol. The members were are not restricted to work only in their area, but they specialize in one area and are responsible for making sure that it meets its requirements. This let to the improvement of the final product.

To make the time plan and work logging we used different techniques. For the timing plan we used the gandchart, for work logging we used a new concept of traffic light status remainder. To troubleshoot the issues raised during different phases we used the concept of 3W1H (**W**hat, **W**hen, **W**ho and **H**ow).To monitor the status of the project we had check points and review meetings at regular intervals to meet the time plan.

By Angel Mario: Electronic design of the Cansat and its subsystems is completed in its early phase, this includes: Subsystem identification and conceptual integration. Subsystem identification
Within the Cansat Electronic design we find the following subsystems: Power distribution, Data acquisition, Processing and Interfacing.
Lars Anders: I am primarily in charge of the construction of the structure of the CanSat.
Since the construction is an early part of the project, I will later help develop the Java user interface. My background is engineering physics.
Jan Philipp: The core of the GroundStation software is already implemented.As the lead programmer, I depend on Pakasit, and Hankang mainly concerning the communications protocol, timing, and testing. I am looking forward, to make the application as flexible and easy to use as possible. However, stability and recovery from communication loss are the main priorities.

Hankang, Wang: responsible for main program, communication module, and communication protocol improvement, system intergration, system debug, document organization.
Narayanan: Narayanan will be responsible for the Project management .In the initial stages,he will be working on the Cansat Circuit design and the circuit layout. As a part of the cansat programming ,he will be doing the data acquisition of the pressure sensor and the Microcontroller programming.
Worracharoen, Pakasit: responsible for GPS data-acquisition module, temperature dataacquisition module, and especially help to hardware debug.



3.2. TIME PLAN

To make the timing plan for the whole project we used the Gand chart. We initially made a brain storming to identify the tasks involved in this project. Then we started distributing the tasks based on the milestone requirements. As we had distributed structure, the responsibilities changed with respect to the milestones. The dates were fixed based on the individual opinion, workload required and given deadline. The Gand chart is as shown below.

Based on the position of the project during each milestone the tasks and the dates were refined to accommodate new improvements on the system. For example PCB, was a unplanned activity, even still we were able to match up with the deadlines. This carried lots of risk, but the team members were so supportive to take the decision and accomplish the milestone as planned.

The tables shown below gives a idea about the time planning of the whole project.

			OCTOBER				NOVEMBER				DECEMBER				JANUARY				FEBRUARY		
S.No	**Activity**	**Responsibility**	7	14	21	30	7	14	21	30	7	14	21	30	7	14	21	30	7	14	21
1	**M1: Mission Analysis and Planning**																				
	Define Task	TEAM	■																		
	Allocation of task	TEAM		■																	
	Time schedule/ Work Plan	TEAM		■	■																
	Identify the Different Subsystems	TEAM		■	■																
	System Architecture	TEAM	■	■	■																
	Hardware requirement/Structural Req	ANDE		■	■	■															
	Circuit design	ANG/NAR		■	■	■															
	Data Transfer Protocol outline	HANK/PAK		■	■	■															
	CANSAT Programming outline	PAK/NAR			■	■															
	JAVA Programing/Front End(Ground station)	JAN		■	■	■															
	Final review meet of Mission 1	TEAM				■															
	Submit report					■															
2	**M2:Implementation and Integration**																				
	Integration of Ground Station	HANK/PAK				■	■	■	■	■											
	Ground Station Programming	JAN					■	■													
	Testing of ground station/Hardware	JAN/PAK						■	■												
	Structure Fabrication	ANDE/NAR					■	■	■												
	Implement Power supply/ATMEL	ANG					■														
	Implement Pressure/Temp sensor	ANG/JAN					■	■													
	Implement GPS	HAK					■	■													
	Implement RT433/Max	ANG/HAK						■	■												
	Review Meeting-check point	TEAM							■												
	Implement test grounds for each subsystems	HAN/ANG/NAR						■	■	■											
	Integration of CANSAT	ANG							■	■											
	CANSAT Programming	PAK/NAR							■	■	■										
	Integrate various subsystems together	ANG								■	■	■									
	Final Review meet of Mission 2	TEAM									■										
	Submit report											■									
3	**M3:Test Evaluation**																				
	Accelerometer (Addition)	JAN/ANG										■	■	■							
	PCB design/Fabrication	PAK/NAR/Manuf											■	■	■						
	Protocol/GS Interface	JAN/HAN										■	■	■	■	■					
	Circuit Debugging/Final	HAN/ANG														■					
	Software Debugging	JAN/PAK														■	■	■			
	Review meeting - Check point	TEAM														■	■				
	Implement test grounds for the complete system/structure	HAN/ANG/NAR/ANDE														■	■	■			
	Final Review meet of Mission 3	TEAM														■	■				
	Report Submission																	■			
4	**M4:Final Presentation**																				
	Complete project presentation	TEAM																	■		

3.3 WORK LOGGING:

The work logging was done by a uniquely designed process called the TRAFFIC light status. Where three different kinds of status are defined, RED,YELLOW and GREEN. The green color indicates that the task is as per the schedule. The yellow color indicates that the completion of the task is delayed due to issues. The red color indicates the status as out of reach. Fortunately ,in this project there was no red marking .The below table gives the idea about our work logging system.

STATUS		OCT	NOV	DEC	JAN

TRAFFIC LIGHT STATUS	Color
Project on Track	
Delayed Completion	
Out of Deadline	

S.No	Activity	Responsiblity	Date	Status	Remarks	
1	**M1: Mission Analysis and Planning**					
	Define Task	TEAM	10.10.06		Completed	
	Allocation of task	TEAM	16.10.06		Completed	
	Time schedule/ Work Plan	TEAM	20.10.06			
	Identify the Different Subsystems	TEAM	23.10.06		Completed	
	System Architecture	TEAM	23.10.06			
	Hardware requirement/Structural Req	ANDE	25.10.06		Completed	
	Circuit design	ANG/NAR	27.10.06		Completed by/28.10.06	
	Data Transfer Protocol(Algorithm)	HANK/PAK	27.10.06		Completed	
	CANSAT Programming	PAK/NAR	27.10.06			
	JAVA Programing/Front End	JAN	27.10.06			
	Final review meet of Mission 1	TEAM	30.10.06			
	Submit report		**31.10.06**		Resubmit	17.11.06

2	**M2:Implementation and Integration**					
	Integration of Ground Station	HANK/PAK	3.11.06		Completed	Test- CLEAR
	Ground Station Programming	JAN	3.11.06		Completed	yet to be tested
	Testing of ground station/Hardware	JAN/PAK	10.11.06		Completed	Test clear
	Structure Fabrication	ANDE/NAR	17.11.06		Completed	
	Implement Power supply/ATMEL	ANG	17.11.06		Complete	
	Implement Pressure/Temp sensor	ANG	20.11.06			Complete
	Implement GPS	HAK	22.11.06			complete
	Implement RT433/Max	ANG/HAK	24.11.06			complete
	Review Meeting-check point	TEAM	28.11.06			
	Implement test grounds for each subsystems/Description	HAN/ANG/NAR	28.11.06			
	Integration of CANSAT/Hardware	ANG	28.11.06		Complete	
	CANSAT Programming/Detailed description	PAK/NAR	1.12.06			
	Protocol /Detailed Description	HANK	9.12.06			
	Integrate various subsystems together	ANG	9.12.06		Completed on 10.12.06	
	Final Review meet of Mission 2	TEAM	9.12.06			
	Submit report		**12.12.06**			
3	**M3:Test Evaluation**					
	Accelerometer (Addition)	JAN/ANG	20.12.06		Completed	
	PCB design/Fabrication	PAK/NAR/Manuf	27.12.06		Waiting for deleivery..	
	Protocol/GS Interface	JAN/HAN	4.01.07		Completed	
	Circuit Debugging/Final	HAN/ANG	12.01.07			
	Software Debugging	JAN/PAK	15.01.07		Completed	
	Review meeting - Check point	TEAM	15.01.07			
	Implement test grounds for the complete system/structure	HAN/ANG/NAR/ANDE	18.01.07		to be completed on 26.01.07	
	Final Review meet of Mission 3	TEAM	21.01.07			
	Submit report		**23.01.07**			
4	**M4:Final Presentation**					
	Complete project presentation	TEAM	**06.02.07**			

3.4. 3W1H PROCESS

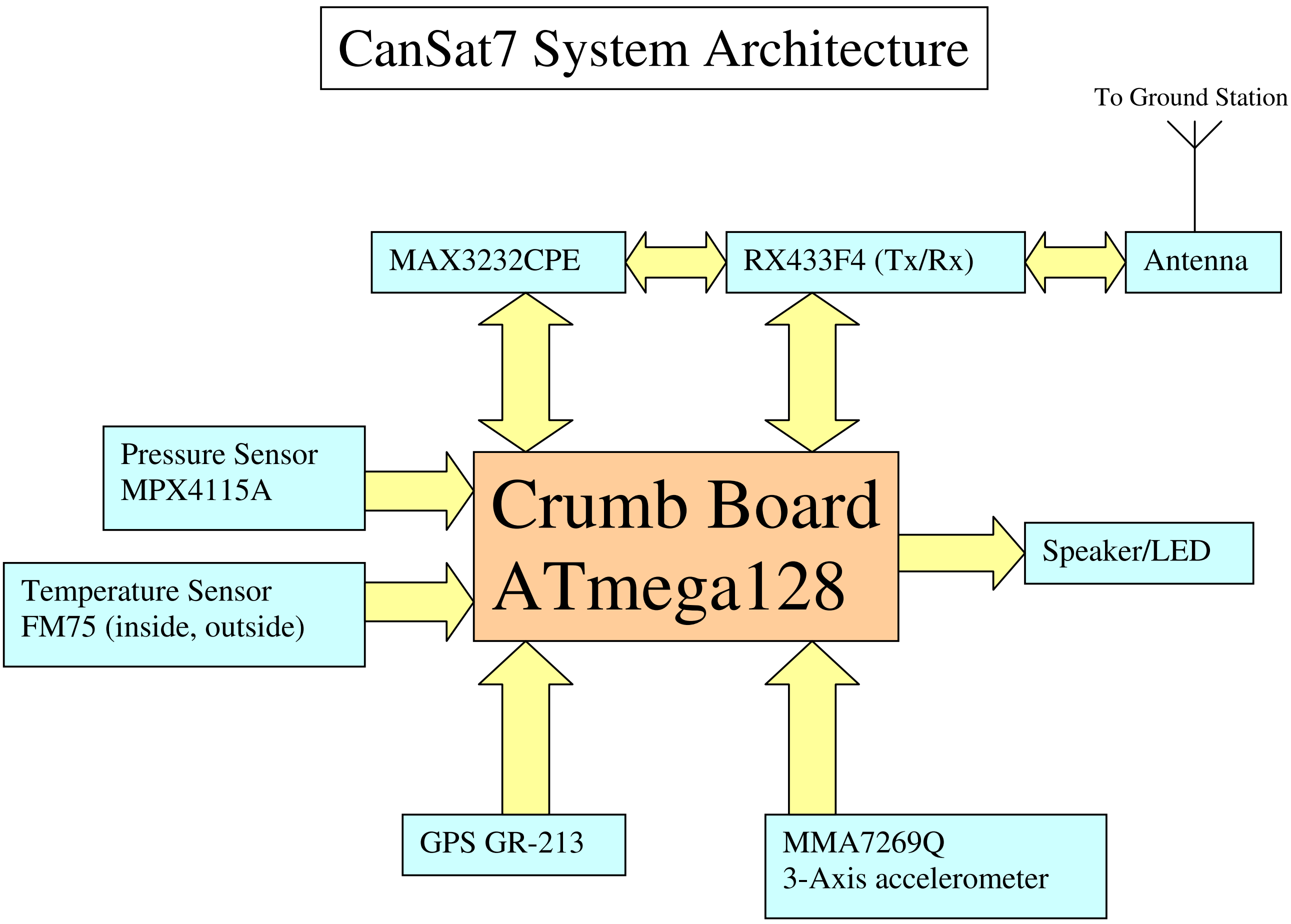
The 3W1H is a process (What, When, Who and How), this tool is used to plan the troubleshooting for a raised issue. It defines, what problem has raised, when it is going to be solved, Who is responsible for it and How it is going to be solved. For any issue, if we apply this method, it will be easy to find the solution and accomplish the given task at the right time. The issues will be raised in the review meeting, where the member who finds the problem in his task will raise in the meeting and a optimum solution will be deduced from the discussion .The below table show the 3W1H process.

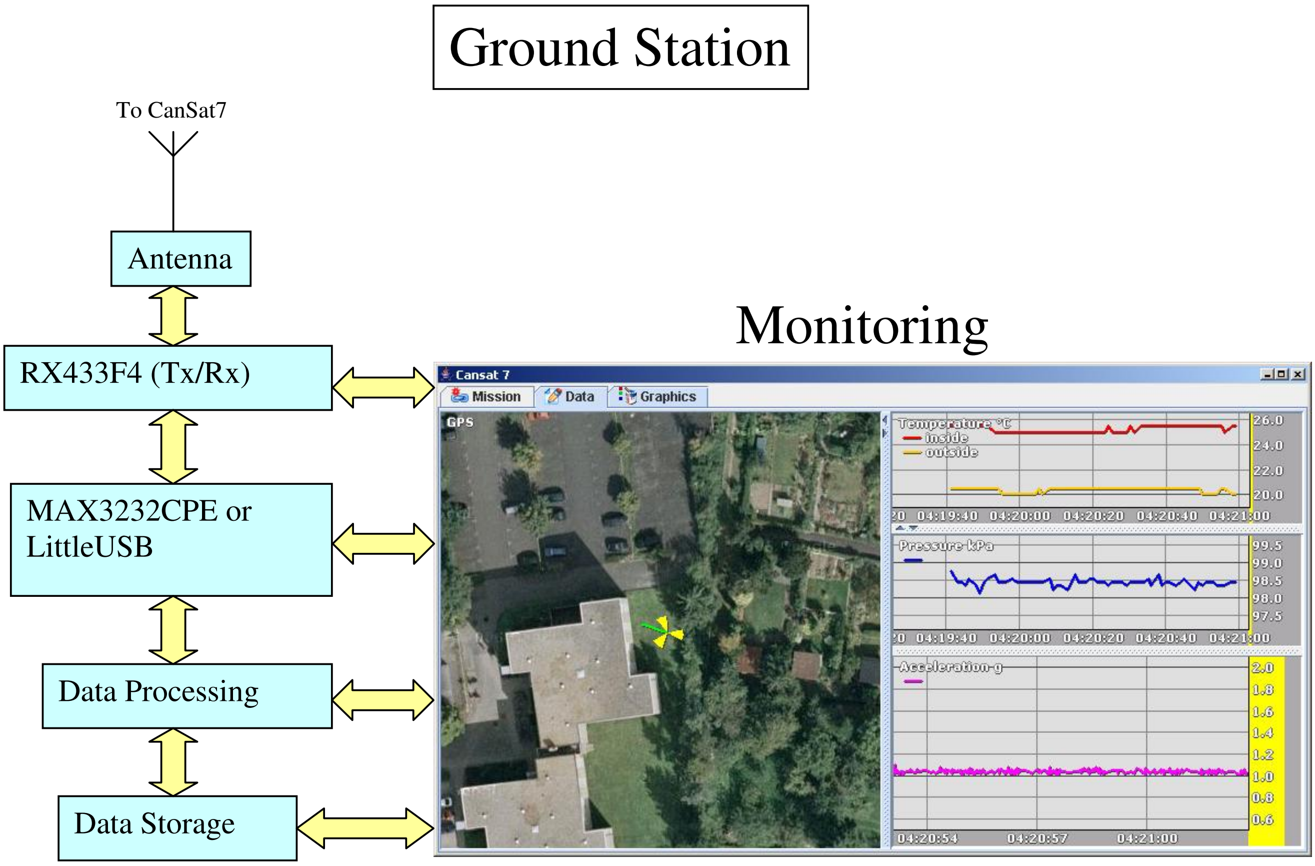
S.No.	WHAT	WHEN	WHO	HOW
1	Circuit design/Power management(M1)	27.10.06	ANG/NAR	Correcions were done for schematics and power management
2	Report Resubmission	14.11.06	TEAM	By Shuffling the Existing Report Layout
3	Integration of Ground Station	10.11.06	HANK/PAK	Cross checking the Continuty
4	Ground Station Programming	14.11.06	JAN	By debugging the program and check the integration with the hardware
5	Testing of ground station/Hardware	14.11.06	JAN/HANK	T o check the circuit and debug the Software
6	Structure Fabrication	22.11.06	AND/NAR	By rechecking the Compatablity of the Circuit board
7	Implement Pressure/Temp sensor	28.11.06	ANG/NAR	Add sensor to current design after stablishing communication
8	Implement GPS	28.11.06	ANG	Add sensor to current design after adding pressure/temp sensor
9	Implement RT433/Max	28.11.06	ANG	Already installed(circuit correction)
10	PCB fabrication	26.01.07	PAK	The design will be sent to the Manuf on 22.01.07 and get it done on 26.01.07(tentative).

CHAPTER IV

ELECTRONICS

4.1 SYSTEM ARCHITECTURE





4.2 ONBOARD &GND ST SPECIFICATION

<u>CanSat7 Circuit</u>

- Refer the Appendix for Schematics

<u>Ground Station Circuit</u>

To communicate with ground station software perfectly, we need a good Ground Station Circuit. That means good qualification, low noise, energy saving and so on.

Our ground station circuit has some special features such as stability in circuit, very small size, a good antenna and no battery using.

- Refer the Appendix for Schematics

<u>Ground Station Circuit V1</u>

The MAX3232 transceivers

The MAX3232 transceivers have a proprietary low-dropout transmitter output stage enabling true RS-232 performance from a 3.0V to 5.5V supply with a dual charge pump. The devices require only four small 0.1µF external chargepump capacitors. The MAX3232 are guaranteed to run at data rates of 120kbps while maintaining RS-232 output levels.

Figure 2 Ground Station V1

<u>Ground Station Circuit V2 (Prototype)</u>
Tiny USB-to-UART Interface Module

LittleUSB is a universal interface adapter module for connecting microcontrollers, FPGAs, etc. to the USB bus. It is based on Silicon Laboratories' CP2102 USB-to-UART interface chip, which converts data traffic between USB and UART formats. littleUSB includes a complete USB 2.0 full-speed function controller, bridge control logic and a UART interface with transmit/receive buffers and handshake signals.

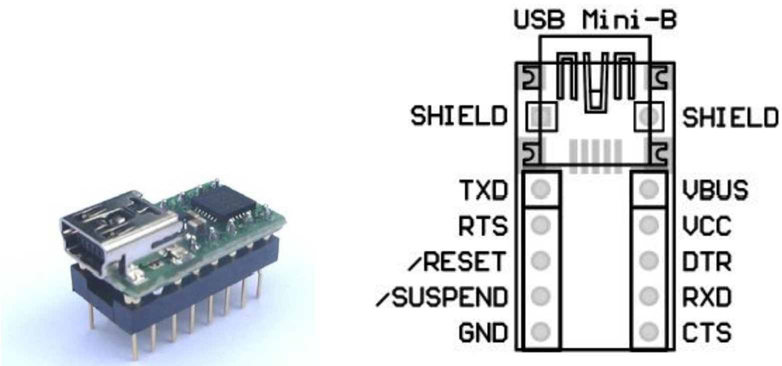


Figure 1 Tiny USB-to-UART Interface Module

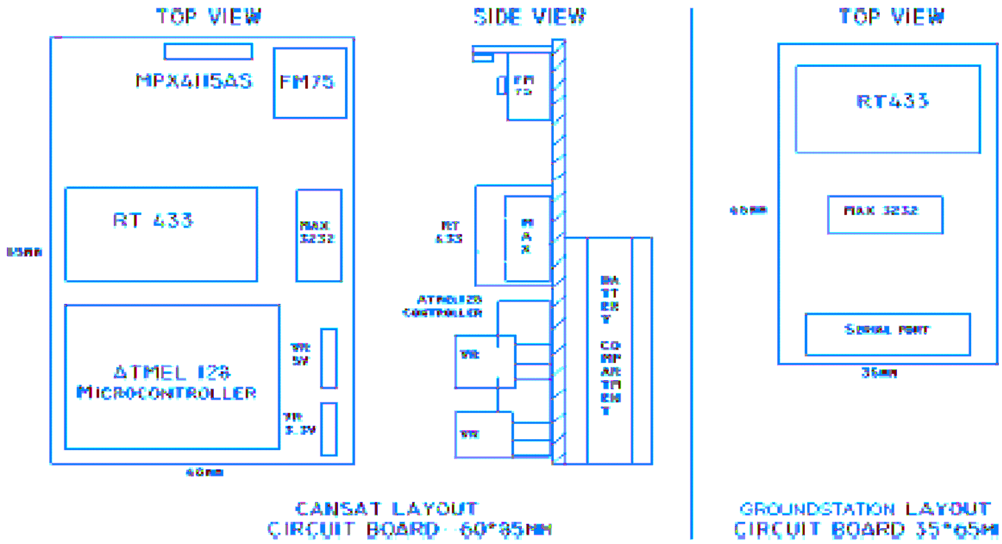
According to Ground Station Circuit V1, it works properly but there are some disadvantages, which are using 9V battery, bigger in size and using serial to USB cable, for instance. The ground station circuit V2 was developed.

The implementation of a USB-based interface from chip45.com has been tested; it already delivers 3.3V CMOS signals and power from USB port in the computer.

With this, we save the use of the MAX3232 for converting RS-232 from serial port; we can directly connect the signals to the RT433F4 transceiver, we also save the 9V battery.

The implementation of this interface is still on test phase. Currently we are using RS-232 serial port as we are having some difficulties with the new interface. Moreover, we have burnt a littleUSB and we have to wait for the new one. The aim is to be able to implement it successfully for the final version.

4.3 BOARD LAYOUT



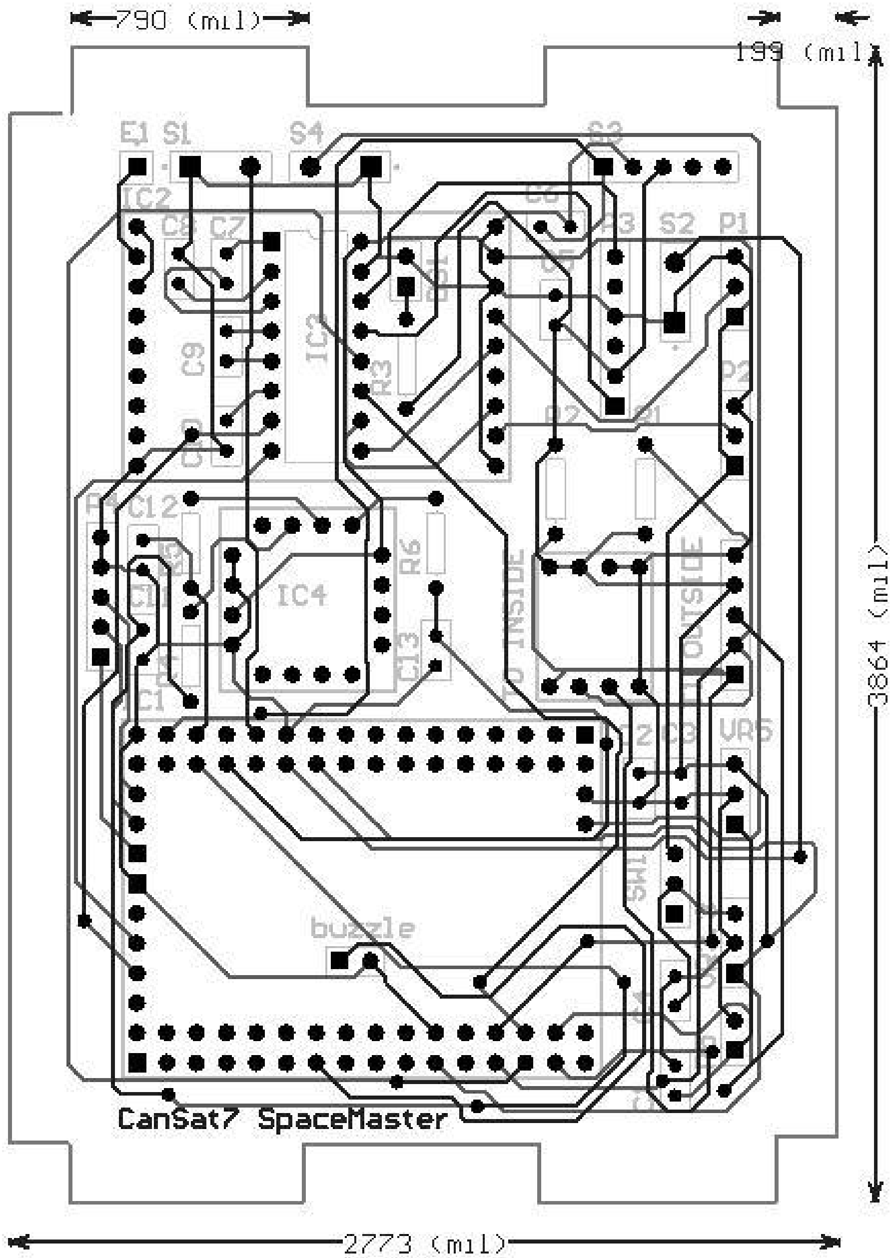


Figure 1 CanSat7 Layout

buzzle : Buzzle
C1 – C12 : Capasitors (100pF)
E1 : Antenna
IC1 : Crumb128 (Crumb board)
IC2 : RT433F4 (Tx/Rx)
IC3 : MPX3232CPE
IC4 : MMA7269Q 3-Axis accelerometer
P1, P2 : Header3 (jumper)

P3 : GPS Connector
P4 : USB Connector
R1 – R11 : Resistors (1k)
SW1 : Battery Switch
S3 : MPX4115A (Pressure Sensor)
S1, S2, S4 : Reset Switch
T1 OUTSIDE : FM75 (Temperature Sensor)
T0 INSIDE : FM75 (Temperature Sensor)
VR5 : TS2940CZ-5,0 (Voltage regulator +5V)
VR3.3 : TS2940CZ-3,3 (Voltage regulator +3.3V)

Ground Station V1 Layout

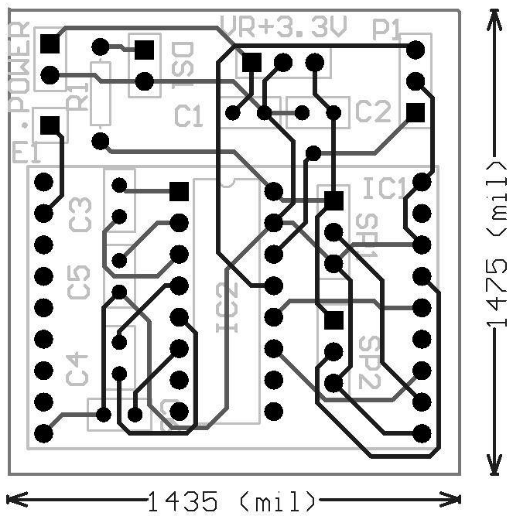


Figure 2 Ground Station V1 Layout

C1, C2 : Capacitor 10 uF
C3, C4, C5, C6 : Capacitor 100pF
DS1 : LED
E1 : Antenna
IC1 : RT433F4 Tranceiver/Receiver
IC2 : MAX3232CPE
POWER : Power Supply connector
P1, SP1, SP2 : Header3 (jumper)
R1 : Resistor 1k
VR+3.3V : TS2940CZ-3,3 (Voltage regulator +3.3V)

Ground Station V2 Layout

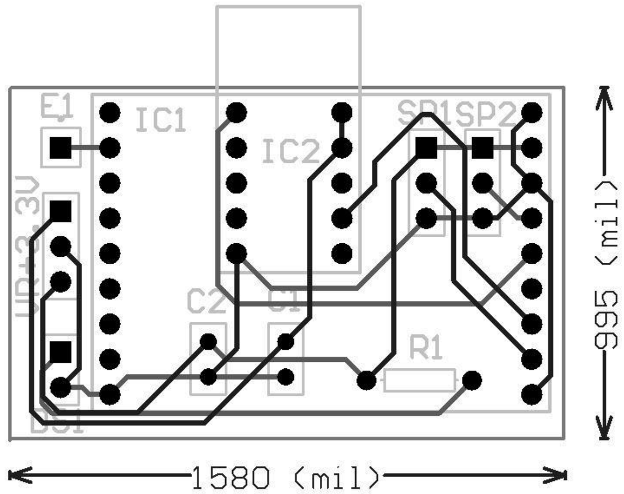


Figure 2 Ground Station V2 Layout

C1, C2 : Capacitor 10 uF
E1 : Antenna
IC1 : RT433F4 Tranceiver/Receiver
IC2 : LittleUSB
R1 : Resistor 1k
SP1, SP2 : Header3 (jumper)
VR+3.3V : TS2940CZ-3,3 (Voltage regulator +3.3V)

4.4 POWER MANAGEMENT

As source of power on the CanSat device, we will choose either a 9*V* Alkaline Battery with approximately 500*mAh*, or a 9*V* NiCd Battery with approximately 250*mAh*. In the following table, we summarize the power consumption of the active components of the CanSat device.

Qty	Part No	Description	Normal to Max Current		Operating Voltage		Power Consumed	
1	ATmega128	Microcontroller	35	mA	5	V	175	mW
1	RT433F4	Transceiver module	60	mA	3.3	V	198	mW
1	FM75	Temp. sensor - I2C interface	10.25	mA	5	V	51.25	mW
1	MPX4115A	Pressure sensor	8	mA	5	V	40	mW
1	Holux GR-213	GPS receiver	80	mA	5	V	400	mW
1	MAX3232CPE+	RS232 2xDr./Rec. 3-5, 5V DIP	20	mA	3.3	V	66	mW
1	MAX3221CPE+	RS232 2xDr./Rec. 3-5, 5V DIP	20	mA	5	V	100	mW
1	TS2940CZ-5,0	Voltage-Reg +5,0 V	30	mA	9	V	270	mW

1	TS2940CZ-3,3	Voltage-Reg +3,3 V	30	mA	5	V	150	mW
1	MMA7260Q	3-axis accelerometer	0.5	mA	3.3	V	1.65	mW
		Total	**293.75**	**mA**		**Total**	**1451.9**	**mW**
	Source of Energy							
					Estimated running time in hours			
	9V Alkaline Battery	Aprox	500	mAh	1.70212766			
	9V NiCd Battery	Aprox	250	mAh	0.85106383			

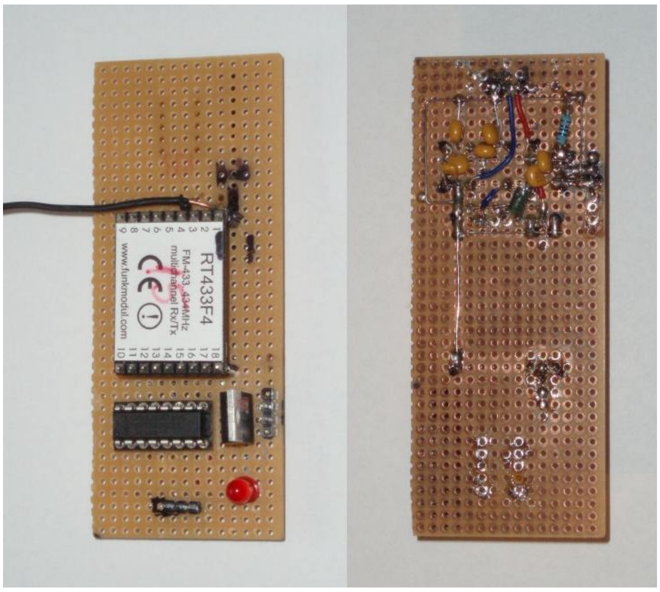
Note:
In practical, the estimated running time is shorter than in the theory. We lost more energy because of resisters, buzzle, LED and so on.

4.7 SOLDER BOARD-PHASE I

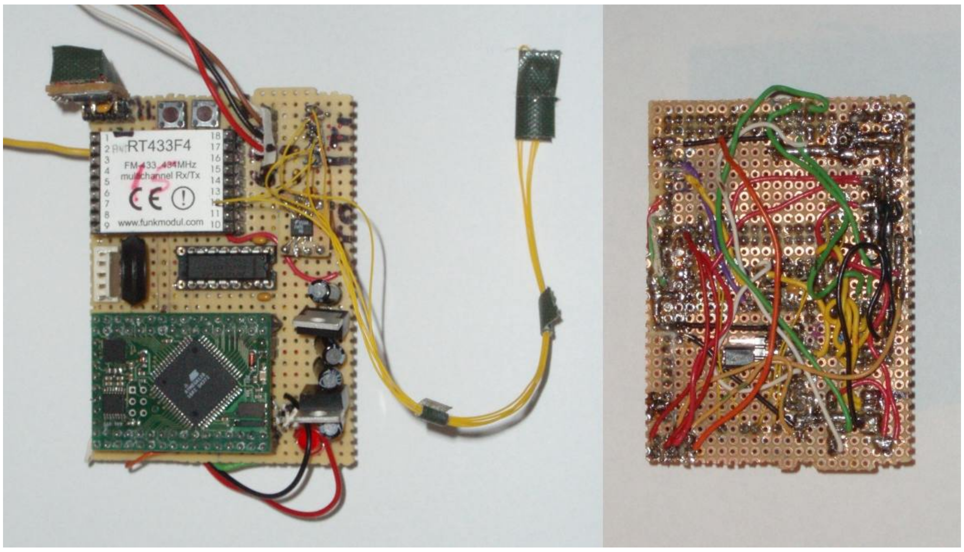
For rapid prototyping, we use a breadboard. Of course, this allowed us

- to test all the sensors
- to install the best antenna
- to add extra stuff, which we did not plan for in the beginning, such as accelerometer and speaker

The first circuit we soldered was the groundstation circuit. The picture shows the front and the back side.



The cansat prototype circuit was soldered and debugged during the Chirstmas holidays.



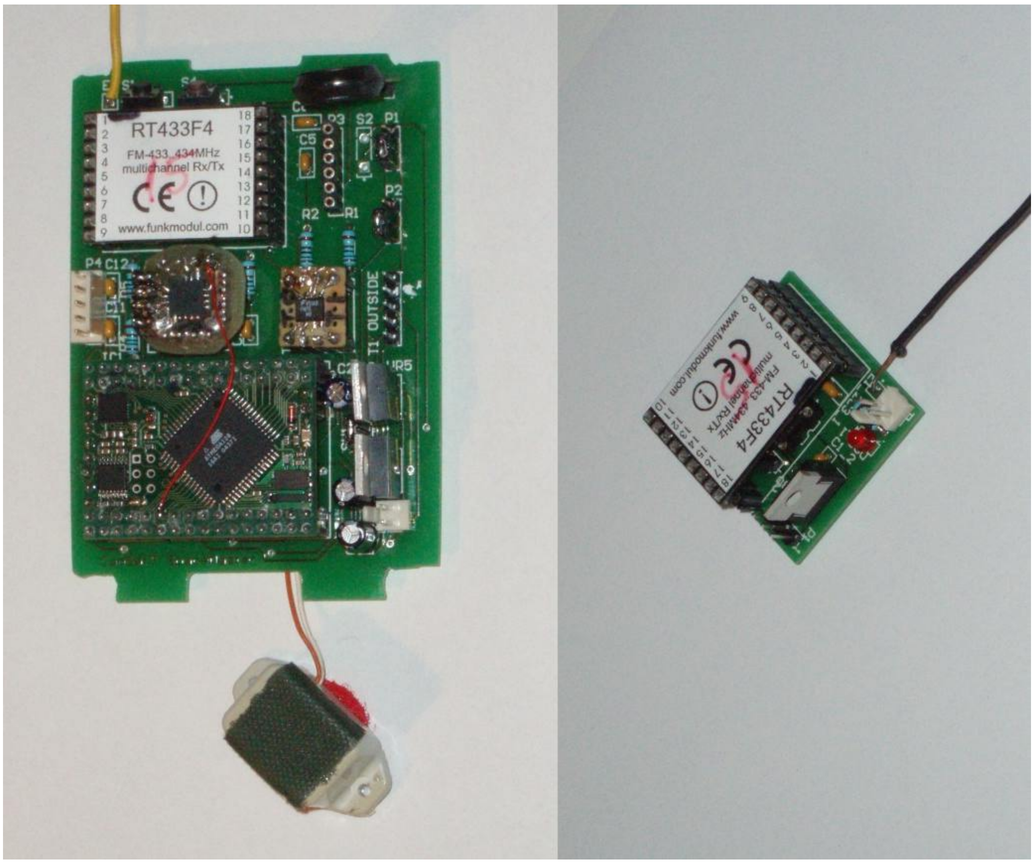
4.8 PCB DESIGN-PHASE II

After we have

- tested all the sensors
- determined the best antenna
- added and optimized all extra stuff such as accelerometer and speaker

we designed a printed circuit board. After we have received the board from the commercial producer, the mounting of the components on both groundstation and cansat was completed in less than a day.

The picture below shows the final result. The open pins are reserved for GPS sensor, and the outside temperature sensor. Cables to which will go thru the lid of the can.


RT433F4
FM-433_434MHz
multichannel Rx/Tx
www.funkmodul.com
OUTSIDE

CHAPTER V

STRUCTURE

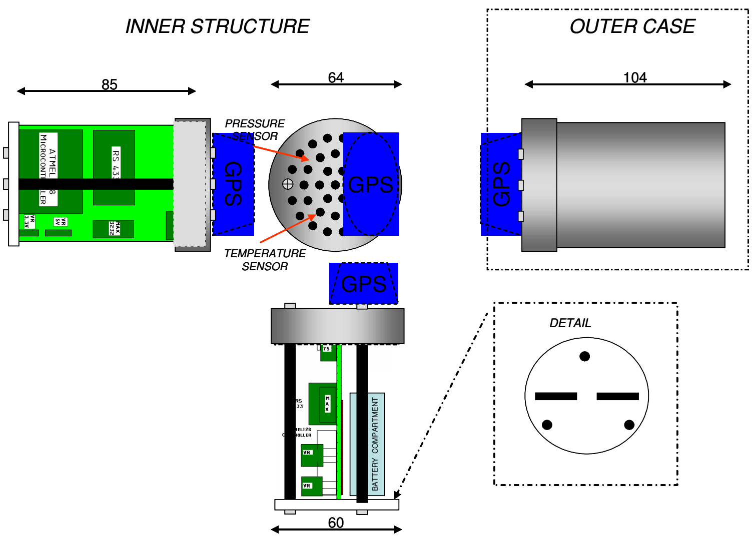
5.1 STRUCTURAL REQUIREMENT

The CanSat has many structural requirements to fulfil. The weight and volume should be the same as a ½ litre soft drink can, i.e. a weight of less than 500 grams and a volume of less than 500 ml.

This structure should be able to withstand moderate accelerations and shocks, as well as providing basic environmental protection for the electronic components.

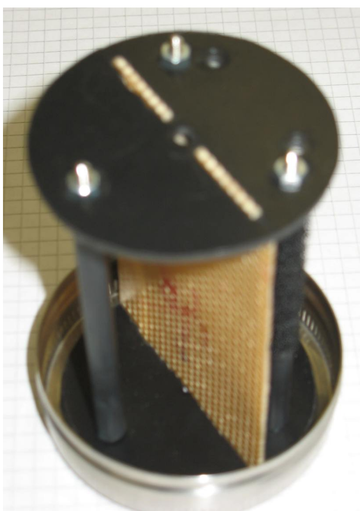
5.2 DESIGN LAYOUT

We are building our CanSat using a stainless steel jar bought in a kitchenware store. This jar has many suitable properties, especially the screw-on lid perforated with 2mm holes. The volume of the jar is approximately 330 ml, well under the designated maximum volume of 0.5 litre. The weight of the jar is 110 grams.



The choice of using this case instead of building one specifically designed for this purpose has pro's and cons. We've had to adapt our design for the case, instead of the other way round. Although sometimes limiting, this gives a different type of challenges, similar to what you would encounter working on an actual satellite. Fortunately for us, the case has not given us any real problems.

As for the construction, we plan to build the whole structure into the lid, leaving the actual jar virtually untouched. This design will ensure easy maintenance, as the circuit board can be accessed simply by screwing the lid of.



The inner structure is based on circular plastic plates that are attached to three cylindrical rods. The rods are also made of stainless steel, and are composed of smaller screw-together rods. The rods will be covered in plastic tape for isolation. Two rectangular slits on each of the plates, and a fitting contour of the circuit board, keeps the circuit board in place. The rods do not make contact with the circuit board.

The battery is mounted on the two rear rods. Not having found a suitable battery compartment, we've decided to go with a Velcro construction to attach the battery to the rods. The components of the circuit board will be mounted on the other side of the circuit board.

As the GPS receiver is said to require “open sky” to work properly, we decided to put it outside the lid, connected by a cord. This will increase the total volume, but we will still be far from exceeding the maximum volume. The cord is will be led through one of the holes of the lid. This feature also provides some perks, for instance not having to place all the CanSat in “open sky” to get GPS to work properly.

The pressure and temperature sensors will be placed just under the holes on the lid, giving them direct contact with the outside environment. We will try to make the whole construction sealed by blocking the rest of the holes (besides the hole used for the GPS receiver cord).

From an aesthetic viewpoint, the jar’s appealing design will give our CanSat a professional look and feel.

We have weighed in the components, inner structure and jar to approximately 300 grams. This leaves 200 grams for battery, wiring et c, which to our knowledge is more than enough.

The construction has almost been completed, having been altered several times.

5.3 MATERIAL SELECTION

The material for the outer structure is stainless steel. This material has a lot of favorable properties, mainly related to its durability.
However, stainless steel still has some drawbacks, the main one being its electric conductance. This has to be avoided by isolation and shielding. At no point in the design is there a direct contact between circuitry and metal case.
The circular plates are made of a plastic. This is a very favorable material; strong, isolating and easy to work with.
The rods are also made of stainless steel, but have been covered in a layer of rubber to provide isolation. These rods are low-maintenance and easy to make into any specific length.

Barometric altitude

We've introduced in our system a barycentric altimeter. This is entirely a software feature, which uses the hypsometric equation to calculate altitude from temperature and pressure. The formula looks like this:

$$h = z_2 - z_1 = \frac{R\overline{T}}{g} \ln\left(\frac{p_1}{p_2}\right)$$

where $z1$ and $z2$ are geometric heights at pressure levels $p1$ and $p2$, respectively; R is the gas constant for dry air; T is the mean temperature of the layer; and g is gravity.

As of now, we are using this formula merely to calculate the height over sea-level, and we therefore set our $p1$ value to the standard 101.3 kBar.

The hypsometric equation is derived from the hydrostatic and the ideal gas law, and is a simplification where meteorological aspects such as low- and high pressure systems and air humidity are not taken into consideration.

It is difficult to make clear error approximations with this formula, since weather conditions are notorious for their unpredictablity. It could be noted, however, that sea-level pressure typically varies within a few millibars during the day. Similarly, the R constant is somewhat affected by water humidity. In addition to this, errors from the temperature- and pressure meter will naturally propagate into this formula.

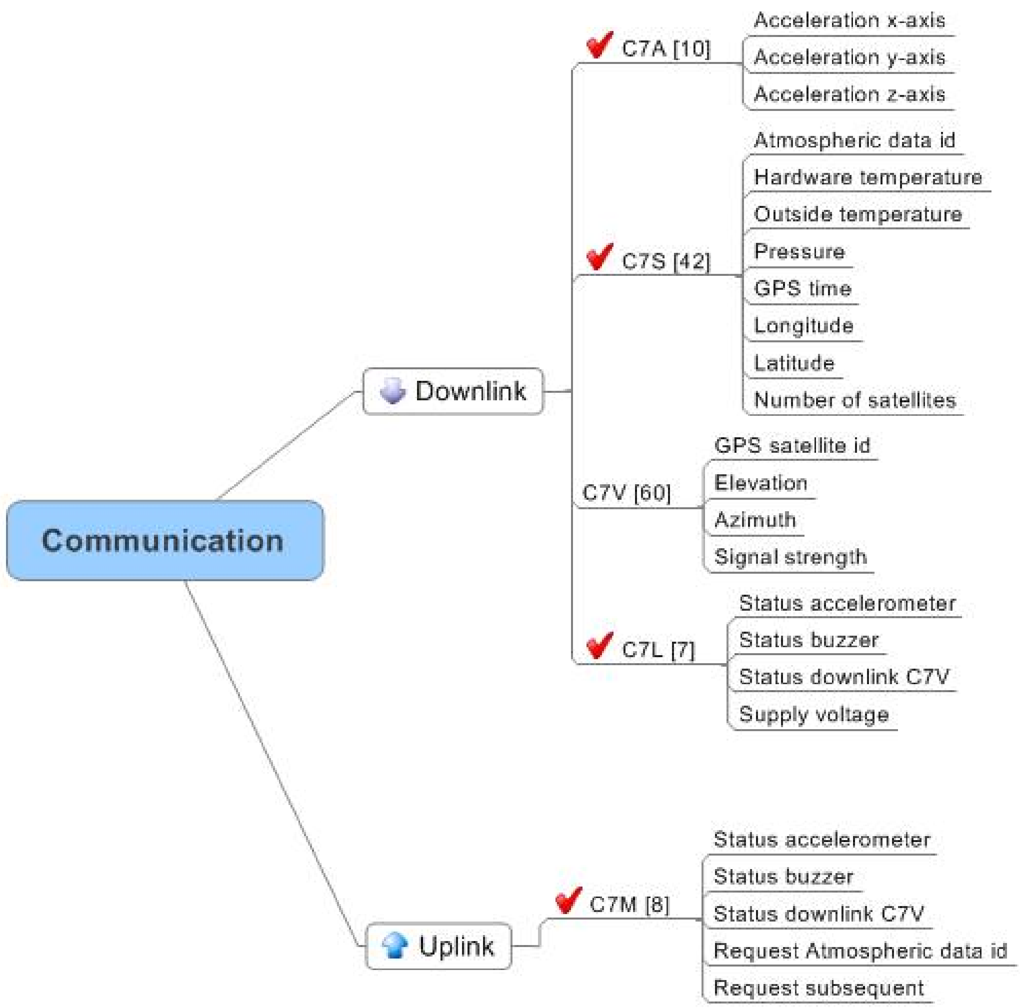
The barometric altimeter should therefore be regarded with a pinch of salt.

CHAPTER VI

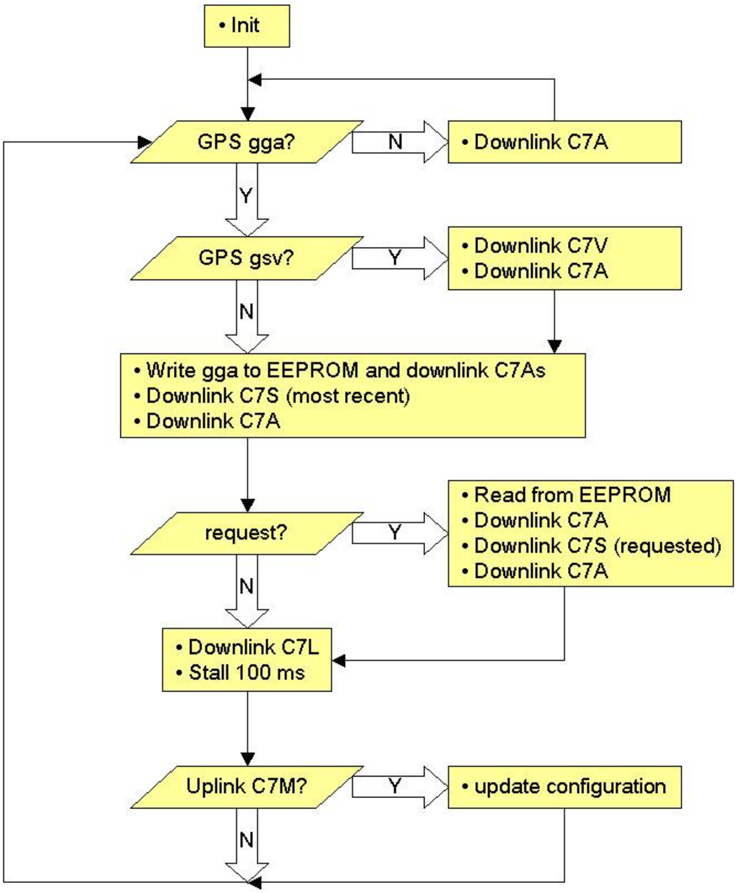
6. CANSAT SOFTWARE AND DATA HANDLING
6.1 Communication Protocol

The protocol consists of various frames. The bit-structure of these frames have been layout in milestone 3. There have not been any modifications since then. In this document, we rather give an abstract overview on the communication flow.

Below, the graphic shows the various frames are sent between cansat and groundstation. The check icon next to the frame header indicates, that we append a checksum to the frame. This will enable us, to identify a corrupt frame on the receiver side. The number between the brackets is the size of the frame in bytes. For instance, the C7S frame consists of 42 bytes.



6.2. Cansat Software



Not all competing groups, use the EEPROM to store data. Therefore, we state the necessary functions to manipulate the EEPROM. The code is motivated by the examples given in the Atmega128 datasheet. In total, there is 4kB EEPROM data available.

```
#define EERE 0
#define EEWE 1
#define EEMWE 2

void EEPROM_write(unsigned int uiAddress, unsigned char ucData) {
  while(EECR & (1<<EEWE)) ; /* Wait for completion of previous write */
  EEAR = uiAddress;         /* Set up address and data registers */
  EEDR = ucData;
  EECR |= 1<<EEMWE;         /* Write logical one to EEMWE */
  EECR |= 1<<EEWE;          /* Start eeprom write by setting EEWE */
}
```

```
unsigned char EEPROM_read(unsigned int uiAddress) {
  while(EECR & (1<<EEWE)) ; /* Wait for completion of previous write */
  EEAR = uiAddress;         /* Set up address register */
  EECR |= 1<<EERE;          /* Start eeprom read by writing EERE */
  return EEDR;              /* Return data from data register */
}
```

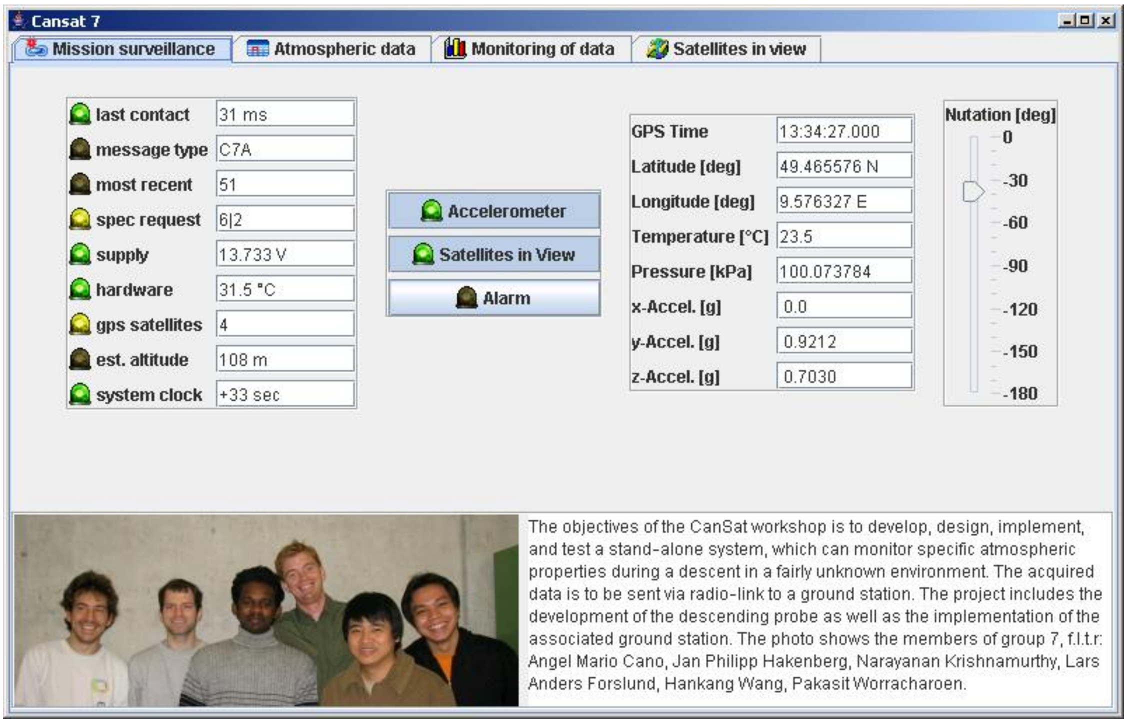
6.3 Groundstation Software

The grounstation program is compatible with Java 1.5.

Mission surveillance

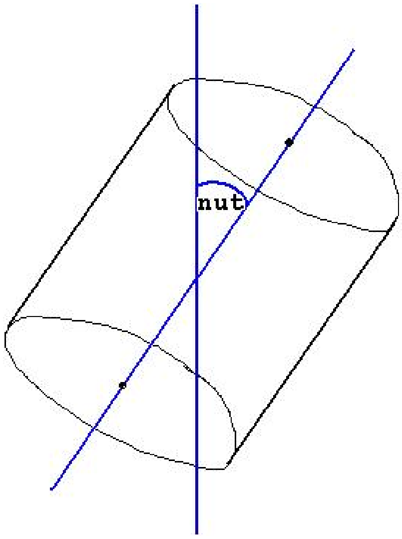
The mission surveillance frame

- shows hardware and communication parameters
- enables the user to control devices on board
- shows the most recent sensor values
- displays the current nutation of the cansat



Nutation of the cansat

Our cansat is equipped with an accelerometer. From this sensor, we obtain the gravitational effect of all three orthogonal axes: a_x, a_y, a_z. Therefore, we can compute the nutation of the can. Adjusted to the mounting of the sensor on the board, the formula for the angle is.

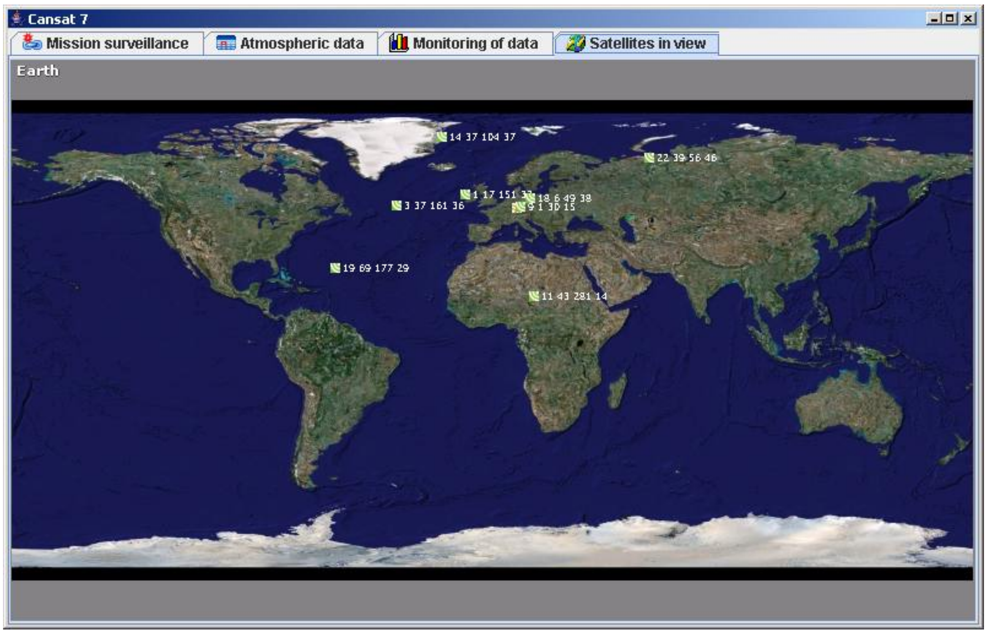


$$nut = \mathrm{ArcCos}\left[\frac{a_y}{\sqrt{a_x^2 + a_y^2 + a_z^2}}\right]$$

We have explained all other panes atmospheric data, and monitoring in milestone 3.

Satellites in View

The last pane of the groundstation program shows a map of the earth. This is new! The orange icon indicates the position of the cansat. Also, we plot the subsatellite points of the gps spacecrafts that are in view of the cansat. They appear in green together with their id number, elevation, azimuth, and signal strength with respect to the gps receiver. In our graphic, no subsatellite point should appear beyond a latitude of 55°.



According to Wikipedia, there are 24 gps satellites distributed equally among six circular orbital planes. The orbital planes are centered on the Earth, not rotating with respect to the distant stars. The six planes have approximately 55° inclination and are separated by 60° right ascension of the ascending node (angle along the equator from a reference point to the orbit's intersection). The orbital radius is 26600 km.

From the GPS receiver, we get the following information

- satellite id: 1, 2, …, 24
- elevation: 0, 1, …, 90 deg
- azimuth: 0, 1, …, 360 deg
- signal strength: 0, 1, 2,… dB

The satellites in view information is transmitted about every 3 seconds. We agree of the following design criteria:

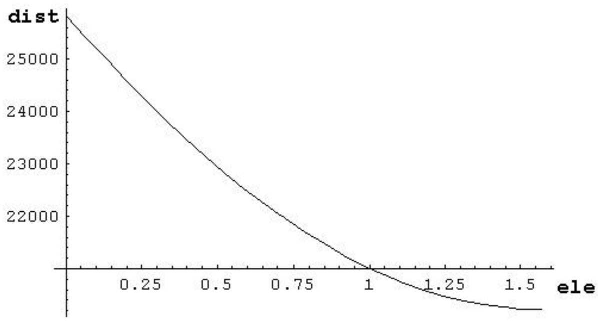
- a GPS satellite is only displayed if its signal strength is greater than 0.
- a GPS satellite that has not been updated for 7.5 seconds is considered to be out of view

What remains is to compute the subsatellite point of the gps satellite on our atlas. We are given the following parameters

lat	=	cansat latitude
lon	=	cansat longitude
ele	=	elevation of gps satellite

azi = azimuth of gps satellite

First, we compute the distance |cansat↔gps satellite| as



$$dist = \frac{0.5\grave{}\left(-12756.28\grave{}\,Sin[ele] + 53200.\grave{}\,\sqrt{0.9425056958284809\grave{}\,Cos[ele]^2 + 1.\grave{}\,Sin[ele]^2}\right)}{Cos[ele]^2 + Sin[ele]^2}$$

Next, we compute the point in 3d, where the gps satellite is in space.

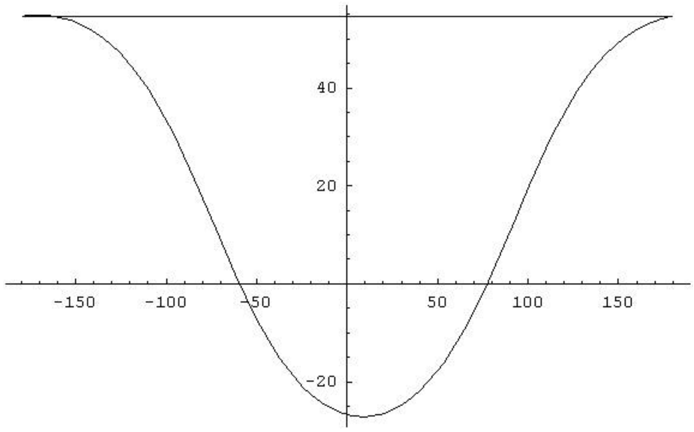
$$pos = \begin{pmatrix} Cos[lat]\,Cos[lon]\,(6378.14 + dist\,Sin[ele]) - \\ \quad dist\,Cos[ele]\,(Cos[lon]\,Sin[azi]\,Sin[lat] + Cos[azi]\,Sin[lon]) \\ dist\,Cos[azi]\,Cos[ele]\,Cos[lon] + (Cos[lat]\,(6378.14 + dist\,Sin[ele]) - \\ \quad 1.\,dist\,Cos[ele]\,Sin[azi]\,Sin[lat])\,Sin[lon] \\ dist\,Cos[ele]\,Cos[lat]\,Sin[azi] + (6378.14 + dist\,Sin[ele])\,Sin[lat] \end{pmatrix}$$

The only thing that remains is to project this point onto the atlas. This is done by the well kown projection of spherical nature.

$$\left\{\sqrt{x_1^2 + x_2^2 + x_3^2},\ ArcTan[x_1,\ x_2],\ ArcTan\left[\sqrt{x_1^2 + x_2^2},\ x_3\right]\right\}$$

Then we have longitude and latitude of the subsatellite point. Lets make an example:

We keep the location of cansat at Wuerzburg. We set the elevation to 0, but let the azimuth range from 0 to 360.



The final java code looks as follows:

```java
double lat=0.0174533*cur_gpsy;
double latCos=Math.cos(lat);
double latSin=Math.sin(lat);
double lon=0.0174533*cur_gpsx;
double lonCos=Math.cos(lon);
double lonSin=Math.sin(lon);
double e=0.0174533*(90-myTok[SEL]);
double eCos=Math.cos(e);
double eSin=Math.sin(e);
double a=0.0174533*myTok[SAZ];
double aCos=Math.cos(a);
double aSin=Math.sin(a);
double d=.5*(-12756.3*eSin+53200*Math.sqrt(0.942506*eCos*eCos+eSin*eSin))/(eCos*eCos+eSin*eSin);
double x1=latCos*lonCos*(6378.14+d*eSin)-d*eCos*lonCos*aSin*latSin-d*aCos*eCos*lonSin;
double x2=d*aCos*eCos*lonCos+latCos*(6378.14+d*eSin)*lonSin-d*eCos*aSin*latSin*lonSin;
double x3=d*eCos*latCos*aSin+(6378.14+d*eSin)*latSin;
double gpsx=57.2958*Math.atan2(x2,x1);
double gpsy=57.2958*Math.atan2(x3,Math.sqrt(x1*x1+x2*x2));
```

CHAPTER VII

SYSTEM TESTIND AND VALIDATION

7.1 STRUCTURE TESTING

The physical testing of our CanSat structure is subject to an inherent problem: Since the only have one CanSat, testing the limits before it breaks is not a wise decision. In fact, any physical testing that poses a even a minute threat to the well-being of our prototype, poses an equal threat to the project as a whole. Fortunately, our mission requirements don't demand any extraordinary robustness of our structure, but only that the CanSat survive moderate accelerations and shocks. This is very unspecific, and would to us be interpreted not as an incentive to conduct advanced testing, but merely as a rule of thumb in the design and handling of our system.

What we can do is to make some basic calculations on the robustness of our stainless steel jar:

Test RC-car

On Friday, January 13th 2006, we tested the communication between CanSat and ground station, placing the CanSat on a RC-car (courtesy of the good people in the robotics lab) and driving it around the lawn and testing terrains of the university campus.

The test was successful. The GPS, thermometer and pressure meter all returned plausible results. As expected, our communication ended when the distance extended fifty meters, but was reestablished as soon as the vehicle returned in range.

The physical testing of our CanSat structure is subject to an inherent problem: Since the only have one CanSat, testing the limits before it breaks is not a wise decision. In fact, any physical testing that poses a even a minute threat to the well-being of our prototype, poses an equal threat to the project as a whole. Fortunately, our mission requirements don't demand any extraordinary robustness of our structure, but only that the CanSat survive moderate accelerations and shocks. This is very unspecific, and would to us be interpreted not as an incentive to conduct advanced testing, but merely as a rule of thumb in the design and handling of our system.

What we can do is to make some basic calculations on the robustness of our stainless steel jar:

Steel was chosen for the material .Steel has a high yield strength (280-600 Mpa), it has a much greater density (7850 kg/m^3) than aluminum, making it desirable

from a structural strengh perspective. Other possible materials are much more expensive than aluminum and steel.
Preliminary calculations have been done to determine the lowest area that would be able to withstand the maximum stress. With a yield strength of 35 MPa and a maximum acceleration of 40 g's, the following calculations were performed:

$$\sigma_y = \frac{F}{A}$$

where F is the force and A is the area.

$F = ma = 110\ g * 40 * 9.805\ m/s^2 = 43.27\ KN$
$\sigma_y = 300\ MPa$
$A = F/\sigma_y = 1.37e\text{-}4\ m^2$

Our can has no points with an area lower than the calculated amount, so it should be able to withstand the highest stress. This was also a conservative calculation: we used the highest stress and the lowest yield strength, so we will actually be able to have a lower area and still not have structural failure.

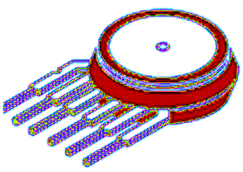
7.2 PRESSURE TEST

7.2.1 Test Requirement

Pressure sensor measures the pressure and transmits the data which is analog voltage to the AVR microcontroller. It needs the analog-digital-converter to transform the acquired data. The electrical characteristic is shown on the form below.

Pressure range	15~115 Kpa
Supply voltage	4.85~5.35 V (Typical: 5.1)
Supply current	7.0 mA
Storage/Operating temperature	-40°+125°
Accuracy	±1.5%

Table 8-1: Electrical characteristics of MPX4115A



PIN NUMBER			
1	V_{out}	4	N/C
2	Gnd	5	N/C
3	V_S	6	N/C

Fig. 8-1 Pin description of the pressure sensor

There is one pressure sensor inside the CanSat. The sensor has temperature compensation system from –40° to +125°C. This allows it to function properly during most of the BEXUS mission. According to the data of previous BEXUS mission, the pressure could reach down to one tenth of pressure of ground level. There is no lower limit for the pressure sensor.

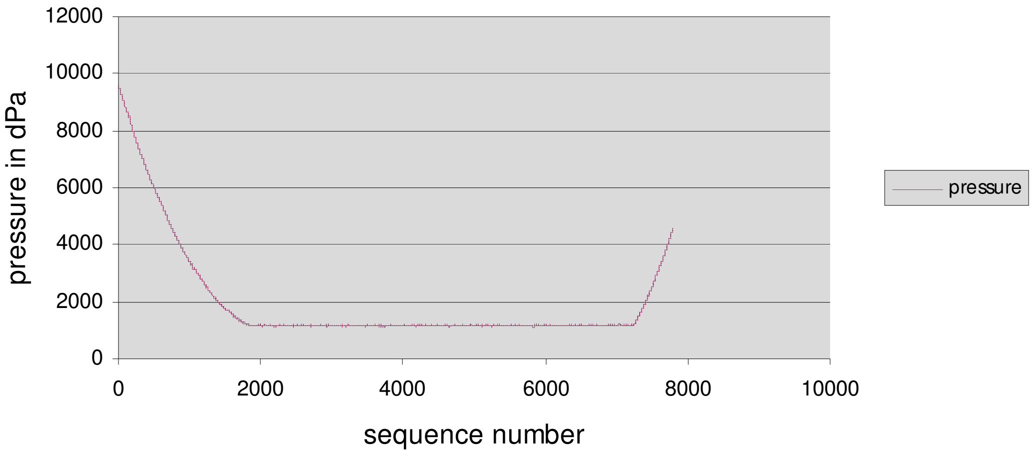


Fig. 8-2 Previous BEXUS mission pressure sensor data

Table 8-2 shows the operational temperature range of each of the CanSat components. For components outside the CanSat probe, mission temperature range might overlap operational range. However, according to the temperature data of previous BEXUS experiment (Fig. 8-2), temperature drops below -20 degrees only once, and this is only a short period. (Note: the temperature drop at the end of the graph happened after the mission finished.) We concluded that the operational temperature range is acceptable for most of the mission duration.

Table 8-2. Operation Temperature Range of Components and Estimated Temperature Range of the Mission

Outside the CanSat		
	Operation Temperature Range	Mission Temperature Range
Photobarrier	-40 to 75	-35 to 50 (Previous BEXUS Test) -50 to 20 (Temp. & Height Graph)
Temperature Sensor	-40 to 75	
GPS	-45 to 100	
Humidity Sensor	-25 to 85	
Inside the CanSat		
	Operation Temperature Range	Mission Temperature Range
Pressure Sensor	-40 to 125	10 to 50
AVR	-40 to 125	
Transceiver	-20 to 70	

Level Converter	0 to 70	

7.2.2 Test Result

The MPX4115A is an integrated pressure sensor measuring pressure from 15 to 115 kPa and providing an output of 0.2 to 4.8V. Those measurements involve a maximum error of 1.5% from 0°C to 85°C. This device is very easy as it has only three pins to be connected: supply (5V), ground and output. Then, the output is analogical. The output voltage is directly proportional to the pressure measured from 15kPa to 115kPa.

The maximum power consumption for this pressure sensor is
P=Umax.Imax=5.35*0.01=0.0535W

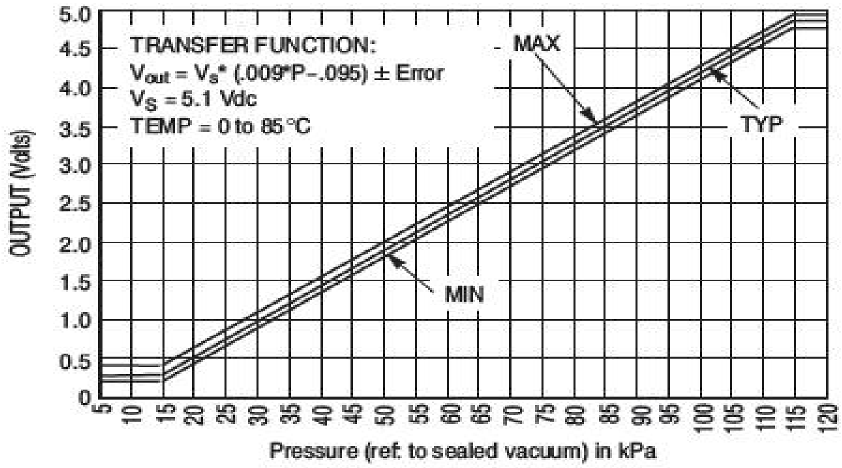


Fig. 8-3 relation between sensor output and signal pressure input

Equation below shows the sensor output signal relative to pressure input. Typical minimum and maximum output curves are shown for operation over 0 to 85°C temperature range. The output will saturate outside of the rated pressure range.

Nominal Transfer Value: $V_{out} = V_S \times (0.009 \times P - 0.095)$
$\pm$ (Pressure Error x Temp. Factor x $0.009 \times V_S$)
$V_S = 5.1 \pm 0.25$ Vdc

The data we received is: pressure: 101.2 kPa. This is quite reasonable value for atmosphere pressure

To test the pressure sensor we need a piece of rubber or soft plastic hose no more than a quarter inch in diameter. A pet store that sells fish has the ideal size air hose. Run the program that continuously displays the pressure readings. Cut a piece about 6 inches to a foot long. Looking at the sensor board, place one end of the hose over the hole on the metal side of the sensor. With the free side of the hose, put the air through with your mouth. You should see the pressure reading increase. If not, readjust the hose on the pressure sensor until you have a better seal. Make sure the end of the hose is flat and smooth.

As the graph shows, using this technique we are able to increase the pressure by 8 kPa.

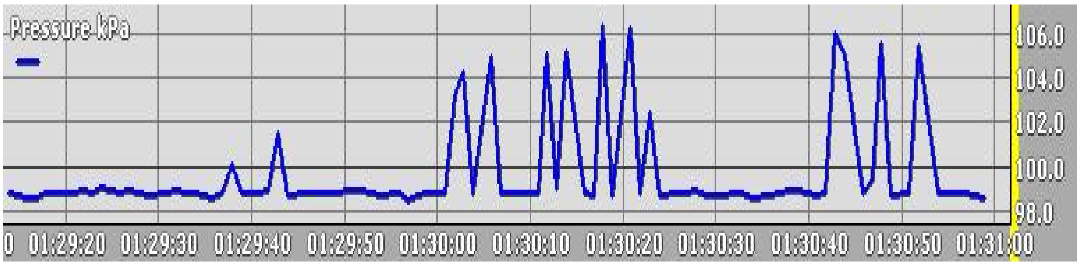


Fig. 8-4 Pressure Sensor Data

7.2.3 Error Corrections And Calibration

In order to collect data from the sensors, the signals from the sensors need to be converted into digital numbers that the processor can handle. The sensors generate a varying voltage based on what is measured and the processor cannot understand it. The processor is digital so it only understands ones and zeros.

An analog to digital converter or ADC allows a processor to measure voltages. The world outside the computer does not have discrete steps such as on/off, high/low, one/zero. It’s an analog world. The ADC allows the computer to measure the analog world with the ADC. The ADC is used to measure the voltage and convert it to a digital number that the computer can use. The computer can take that digital number and process it to calculate the pressure.

The processor has an interface component called and analog to digital converter or ADC for short. The ADC converts a voltage to an integer number. The integer number can be used to calculate the voltage that was measured. The microcontroller has a 10bit ADC. This gives an integer range of 0 to 1024 which covers 0 to 5 volts. The following equation determines the voltage measured:

voltage = measured / 1024 * 5

If the ADC generated an integer number value of 512 then the voltage is 512 / 1024 * 5 or 2.5 volts which is half the voltage range and half the ADC range. 1024 is the number values that the ADC can generate. With an ADC value of 512, the voltage is half the maximum voltage which is 2.5 volts.

The pressure sensor measures the atmospheric pressure and generates a voltage proportional to the air pressure. The higher the air pressure, the higher the voltage.

An equation is provided by the manufacturer:
$V = 5.0(0.009P *0.095)$

- V is the voltage and P is the air pressure in kilopascals.
- The pressure sensor is connected to pin P0.

The equation for the pressure sensor needs to be solved for P.
V = 5.0(0.009P *0.095)
V = (5.0 * 0.009 * P) (5.0 * 0.095)+Error Factor
V = 0.045 * P 0.475
V + 0.475 –Error factor= 0.045 * P
((V + 0.475)-Error Factor)/0.045 = P
P = ((V + 0.475)-Error Factor) * 22.222
P = 22.222 * V + 10.556 –(22.222*EF)

Accuracy

As stated in the data sheet, the sensor accuracy depends on the outside temperature. In average, we expect an error of the order of 1.5 kPa. But more important, the pressure sensor is not able to indicate pressure below 15 kPa. As the previous year cansat evaluation has shown, beyond this limit the pressure sensor will keep indicating a pressure of 15 kPa.

Fig. 8-5 Pressure Sensor Data
Sources of Error

Above, we give an evaluation of the raw pressure information, which we receive from our cansat approximatelzy every second. As one can see, the pressure varies around 0.5 kPa, even if cansat it at rest inside a room. We expect the noise on the reference ground for the analog to digital conversion to cause these randomlike oscillations.

7.3 TEMPERATURE TEST
7.3.1 Test Requirement
There are 2 temperature sensors in the CanSat project. One is used to measure the exterior environment's temperature of the CanSat and the other is to measure the

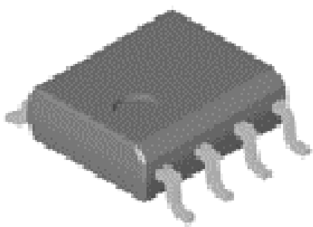


Fig. 8-6: 8Pin SOP

temperature of the interior working environment. Both of them have 8 pins which have the SMBus compatible serial digital interface, and their output data is digital because of the Delta-Sigma analog-to-digital converters within the sensors. The electrical characteristic is shown on the form below.

Table 8-1: The Temperature Sensor

Operating Voltage	2.7V~5.5V
Operating Current	<250uA (low)
Self Heating	0.2℃ MAX in still air
Temperature Range	-40℃~+125℃
Precision Calibrated	±1℃ from 0℃ to 100℃ Typical

To look at the pin description of the temperature sensor, it has O.S, Thermal Alarm output, OS (Over-limit Signal) support Interrupt and Comparator modes. OS is active, if the user-programmable trip-temperature is exceeded. When the temperature falls below the trip-temperature plus the user-programmable hysteresis limit, OS is disabled.

Pin #	Name	Direction	Description
1	SDA	Input/Output	**Serial Data.** Open drain to I/O-data pin for two-wire interface.
2	SCL	Input	**Serial Clock.** Clock for 2-wire serial interface.
3	O.S.	Output	**Over-Limit Signal.** Open drain thermostat output that indicates if the temperature has exceeded user-programmable limits. Default is active low.
4	GND	Supply	**Ground**
5, 6, 7	A0, A1, A2	Input	**Address LSBs.** User selectable address pins for the 3 LSBs of the serial interface address.
8	V_{DD}	Supply	**Supply Voltage**

Table 8-2: Pin Description of FM75 (1)

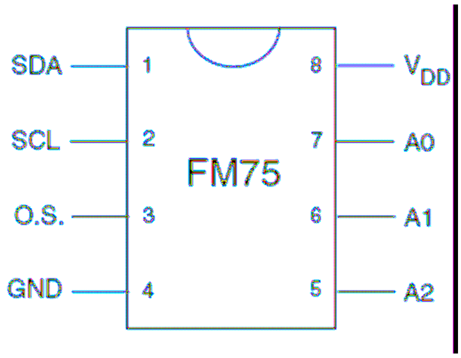


Fig. 8-7 Pin description of FM75 (2)

7.3.2 Test Result

The sensors outside the CanSat indicated around 4 degrees Celsius. This is reasonable with respect to the temperature of 2.3 degrees of weather forecast. The difference is only 2 degrees.
The thing we should notice is the temperature inside the CanSat was decreasing gradually. We can conclude that the temperature inside the CanSat was rather high before the operation due to the workshop inside the robotic hall, and were gradually reaching the temperature outside.
As one can see, the outside temperature sensor measures a quick drop of about 1 °C per second. However, when the temperature drops below 0 °C, we do not convert the sensor values correctly. This ruins our plot as well:

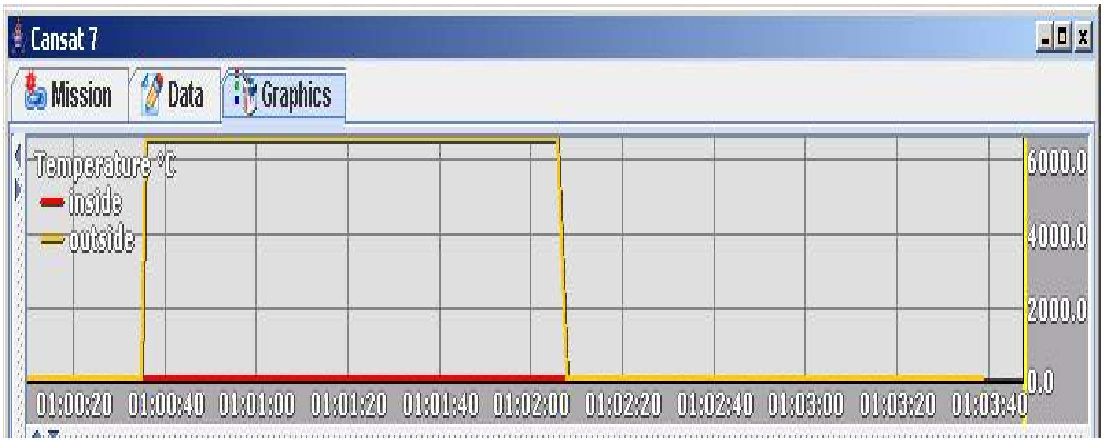


Fig. 8-8 Temperature Sensor Data
In later test, we show that this problem has been overcome.

7.3.3 Error Corrections And Calibration

FM75 temperature sensor uses SMBus, two wire serial interfaces. The SDA/SCL ports need pull-up resister R1~ R2, and the values of them are both 1533 Ohm. The system can choose the temperature sensors by the address line. Shown in the form:

Table 8-5 Address Description of Temperature Sensors

	A1	A2	A3
FM75(2)~~~~0	0(GND)	0(GND)	0(GND)
FM75(1)~~~~1	1(Vcc)	0(GND)	0(GND)

So there is A0/A1/A2 in FM75(1) are all connected to the Vcc and GND and the address pins in the FM75(2) is connected to GND. Output digital data is up to 12 bit resolution, which can be converted into degrees (Celsius) using following equation.

When BIT12 = 0 (Positive degrees): Temperature = 0.0625 x (BIT0-11 in decimal) (degC)

When BIT12 = 1 (Negative degrees): Temperature = -128 + 0.0625 x (BIT0-11 in decimal) (degC)

12 bit resolution provides 0.0625 deg temperature resolution.
11 bit resolution provides 0.125 deg temperature resolution.
10 bit resolution provides 0.25 deg temperature resolution.
9 bit resolution provides 0.5 deg temperature resolution.

Data Interface: I2C (TWI) Digital Bus (5V), connected to SCL(PortD 0), SDA(PortD 1) of AVR

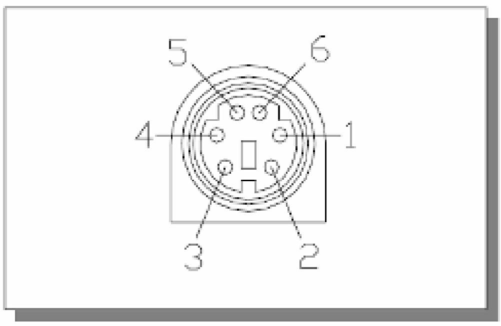
7.4 GPS TEST

7.4.1 Test Requirement

GPS receiver communicates with other electronic utilities via compatible dual-channel through RS-232 or TTL and saves critical satellite data by built–in backup memory. With low power consumption, the GR-213 tracks up to 20 satellites at a time, re- acquires satellite signals in 100 ms and updates position data every second.

This GPS receiver can measure the parameters of the CanSat’s positions and transmit them to the AVR microcontroller by the RS232 or TTL output. In this project, TTL level is chosen because the RS232 serial port on the microcontroller is connected to the transceiver.

GPS receiver has a PS2 female composite connecter, the figure below shows its function.



Pin	Signal	
	RS-232	RS232+TTL
1	Tx	TX(RS232)
2	+5VDC	+5VDC
3	NC	Tx(TTL)
4	Ground	Ground
5	DGPS IN	Rx(TTL)
6	Rx	RX(RS232)

N. C. = No Connection

We do the testing of GPS receivers with the computers. The baud rate of GPS is modulated to be 4800 bps. The place is just outside the house.

The light of GPS is on: the power connection is correct. There are no data received: checked it and then got the reason that the TX and RX are switched.

The light of GPS is on: the power connection is correct. Position data received after modifying the sequence of the cables in the connector.

7.4.2 Test Result

The Global Positioning System is one of the most accurate sensors onboard the CanSat and there was really not much need to test it as such. However, given the easy testing procedure for this, we decided that it should be done.We fixed the coordinates of 3 different points in the city and university with Google Earth along with their latitude and longitude data. Then we took the GPS module connected to the CanSat to these three places and checked the values. The places mentioned in Google Earth and the measured values are shown below.

The Cansat was carried along in a certain path to track the position of the system on the Map. The communication system was tested using a alarm fixed to the cansat. The alarm was switched on and off frequently to check the sending and receiving of the data. The other parameters as the temperature and pressure were also monitored. The communication between the ground station and the Cansat was perfect till a range of 100m and the uplink was fluctuating beyond 120m approximately but the downlink was working perfect. The screenshots shown the test results. The inference from the test was, that the fluctuation in the uplink was due to the antenna or may be due to the delay caused by USB to COM conversion. The trajectory of the system was plotted and as shown in the below figure.
This GPS position is rather westward than we have expected. However the difference is several meters, which is reasonable error for normal GPS system.
The following figure shows the path of CanSat, which is the path the Robot-Car took. This is also reasonable path.

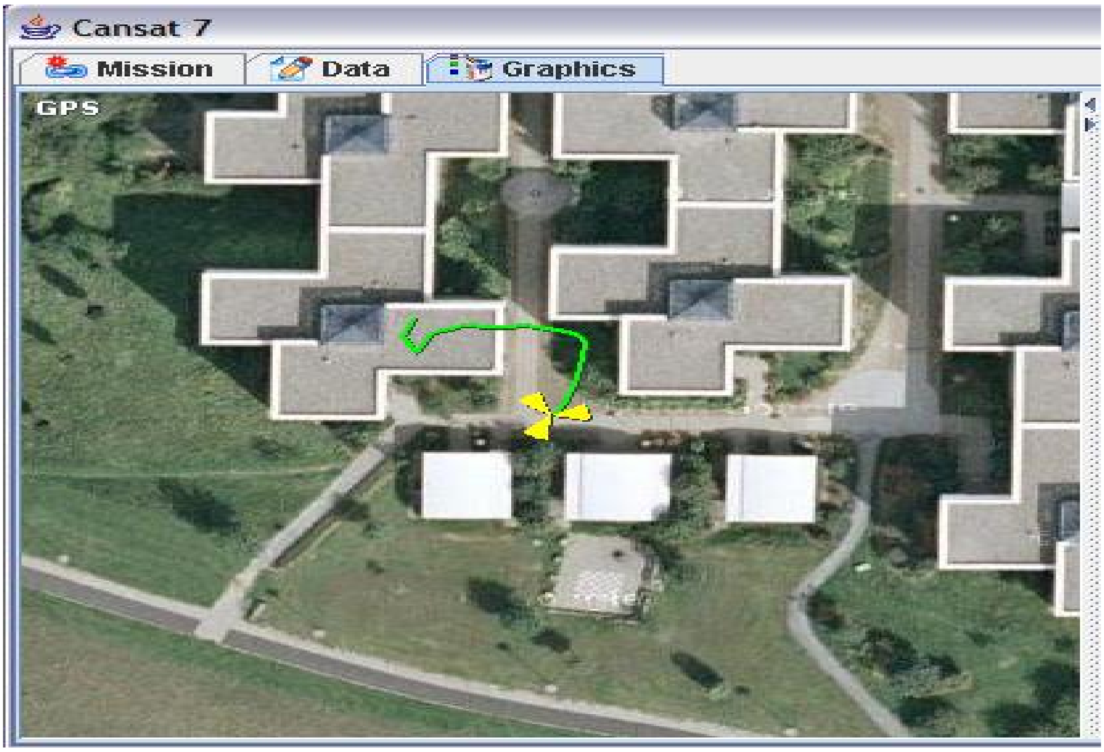


Fig. 8-9 GPS Data Display

7.3.3 Error Corrections And Calibration

In milestone 2, we have explained in detail, what the GPS data, received from the sensor is comprised of. Wikipedia is a good source, to understand how the longitudinal, and latitudinal information is computed.

Accuracy and error sources

The position calculation by a GPS receiver requires the current time, the position of the satellite and the measured delay of the received signal. The position accuracy is primarily dependent on the satellite position and signal delay. Since GPS signals propagate nearly at the speed of light, this represents an error of about 3 meters. This is the minimum error possible using the GPS signal.

Atmospheric effects:

Changing atmospheric conditions change the speed of the GPS signals as they pass through the Earth Atmosphere and ionosphere. These effects are minimized when the satellite is directly overhead. Humidity also causes a variable delay. This effect is much more localized, and changes more quickly than the ionospheric effects, making precise compensation for humidity more difficult. Altitude also causes a variable delay, as the signal passes through less atmosphere at higher elevations.

Multipath effects:

GPS signals can also be affected by multipath issues, where the radio signals reflect off surrounding terrain; buildings, canyon walls, hard ground, etc. These delayed signals can cause inaccuracy. Multipath effects are much less severe in moving vehicles. When the GPS receiver is moving, the false solutions using reflected signals quickly fail to converge and only the direct signals result in stable solutions.

7.5 ACCELEROMETER TEST

7.5.1 Test Requirement

After the submission of the second report, we decided to add an accelerometer to our cansat device. Mr Ziegler has given us permission to extract the accelerometer from a previous cansat. We have modified its mount, and added the filtering, which is suggested in the data sheet

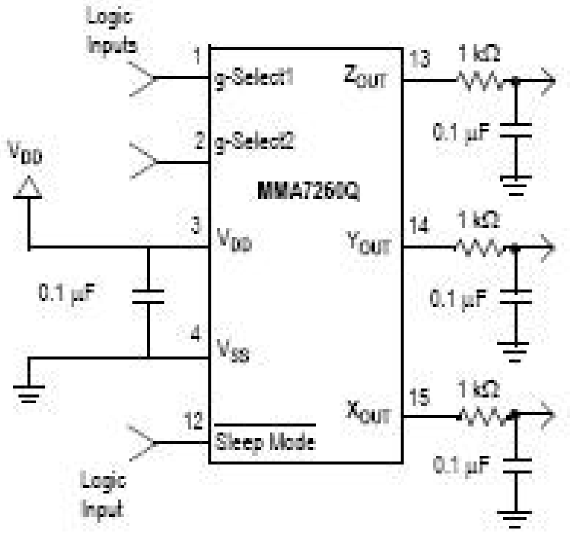


Figure 5. Accelerometer with Recommended Connection Diagram

The accelerometer has three analog outputs: One for each Euler axis. When the cansat is in rest, it will indicate that there is 1 g acceleration upon the sensor. According to the data sheet, we have the following output values, received by the microcontroller:

- When a single axis aligns with the horizon, the analogous output ideally will be 512 (on a scale of 0 to 1023).
- When a single axis points downwards, i.e. towards the center of the earth, the sensor will ideally display 360 (on a scale of 0 to 1023).
- On the other hand, when a single axis points upward, the analog output will be 656 (on a scale of 0 to 1023).

An arbitrary orientation of the sensor will cause a linear interpolation between the extremal values. If the sensor experiences a shock, the output pins can attain values on the full scale of 0 to 1023.

7.5.2 Test Result

Objective: To make sure that the CanSat and its electronic part endure special vibration and acceleration condition of balloon.
CanSat experiences severe vibration and acceleration condition on balloon as shown in the next figure.

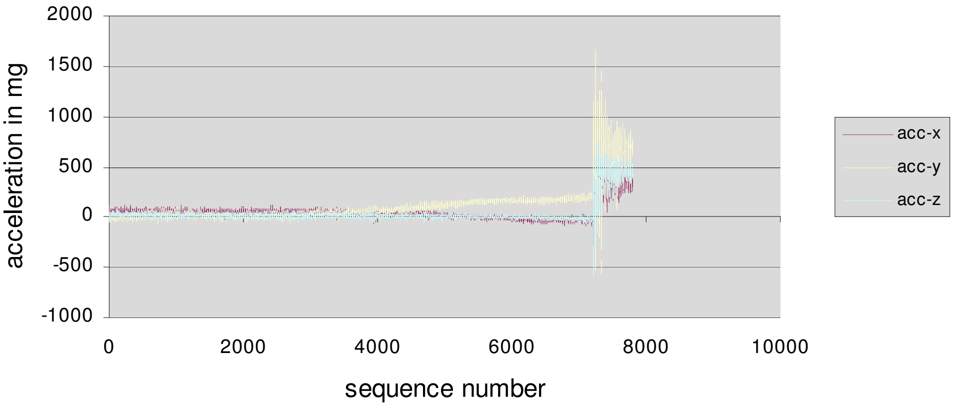


Fig. 8-10 Acceleration Data

(We can neglect the acceleration end of the graph, which occurred after the balloon experiment.) In order to make sure that the CanSat surely work in this acceleration condition, we have to test every function of CanSat after vibration and acceleration test with vibration test facility.

7.5.3 Error Corrections And Calibration

Accuracy

We operate the sensor in the following way: We are performing the analog-to-digital conversion every 20 ms in average, to be able to capture instantaneous, possibly intensive shocks. Meanwhile we might be communicating to other devices, so there will be a lot of noise in the reference voltage. As a result, despite hardware filtering, the output values read by the microcontroller vary with an amplitude of 5. For example, the values around 512 will range between 507 to 517. This is indicated by the plots, we show at a later point of this report.

Sources of Error

We hypothise, that our individual sensor shows signs of aging. When in rest, the norm of the vector resulting from the inputs is not constant. That means, the orientation influences the interpretation of measurements, although it should not have. However, we manage to calibrate the sensor, so that the acceleration computed is *around* 1 g when in rest.

Barometric altitude

We intend to introduced in our system a barocentric altimeter. This is entirerly a software feature, which uses the hypsometric equation to calculate altitude from temperature and pressure. The formula looks like this:

$$h = z_2 - z_1 = \frac{R\overline{T}}{g} \ln \left(\frac{p_1}{p_2}\right)$$

where z1 and z2 are geometric heights at pressure levels p1 and p2, respectively; R is the gas constant for dry air; T is the mean temperature of the layer; and g is gravity.
As of now, we are using this formula merely to calculate the height over sea-level, and we therefore set our p1 value to the standard 101.3 kBar.
The hypsometric equation is derived from the hydrostatic and the ideal gas law, and is a simplification where meterological aspects such as low- and high pressure systems and air humidity are not taken into consideration.
It is difficult to make clear error approximations with this formula, since weather conditions are notorius for their inpredictablity. It could be noted, however, that sea-level pressure typically varies within a few millibars during the day. Similarily, the R constant is somewhat affected by water humidity. In addition to this, errors from the temperature- and pressuremeter will naturally propagate into this formula.
The barometric altometer should therefore be regarded with a pinch of salt.

```
public static float computeHeight(float P,float T) {
  //temperature needs to be given in Kelvin.
  float TKelvin = T + (float)273.15;
  float PGround = (float) 101.325; //kilopascal - ground pressure
  float R = (float) 287.04; //J kg-1 K- Gas constant for dry air
  float g = (float) 9.82;//Gravitational acceleration
  float h = -(R*TKelvin)/g * (float)Math.log(P/PGround);
  System.out.println(h + "  " + PGround+ "  " +R+ "  " +g);
  return h;
}
```

7.6 INTEGRATED TEST

Objective: To make sure that whole CanSat functions work when they are integrated.
Tests and experiments that we have done are “subsystem-independent” or “sensor-independent” tests. Besides these tests, we have to do “integrated” test, which check every CanSat functions with integrated electric board.

Radio controlled car test (Structure testing)

The physical testing of our CanSat structure is subject to an inherent problem: Since the only have one CanSat, testing the limits before it breaks is not a wise decision. In fact, any physical testing that poses a even a minute threat to the well-being of our prototype, poses an equal threat to the project as a whole. Fortunately, our mission requirements don't demand any extraordinary robustness of our structure, but only that the CanSat survive moderate accelerations and shocks.

This is very unspecific, and would to us be interpreted not as an incentive to conduct advanced testing, but merely as a rule of thumb in the design and handling of our system.

On Friday, January 13th 2007, we tested the communication between CanSat and groundstation, placing the CanSat on a RC-car (courtesy of the good people in the robotics lab) and driving it around the lawn and testing terrains of the university campus.

The place in which the cansat has to be tested should have enough space for testing what the maximum distance before losing communication is. Slopes and variations on the terrain are necessary for testing changes in pressure and acceleration. Trees should be available so there are changes in sun incidence, varying temperature and changes in the number of satellites visible to GPS sensor.
On the Ground Station:

Real time monitoring of data and storage, in order to be able to compare to what the results should be, for example with the GPS, the information should describe the pattern of the circuit.

In the case of changes of directions and slopes the accelerometer should return data which is logical, by orders of g forces. The cansat will be exposed to different temperatures, shadows, sunrays, indoors, night, and ice.

For testing data loss and recovery of connection, the cansat will be taken out of range of communication, and perform some maneuvers. Then will be taken back into range and should restore communication and transmit not transmitted data.

The test was successful. The GPS, termometer and pressuremeter all returned plausible results. As expected, our communication ended when the distance extended fifty meters, but was reestablished as soon as the vehicle returned in range.

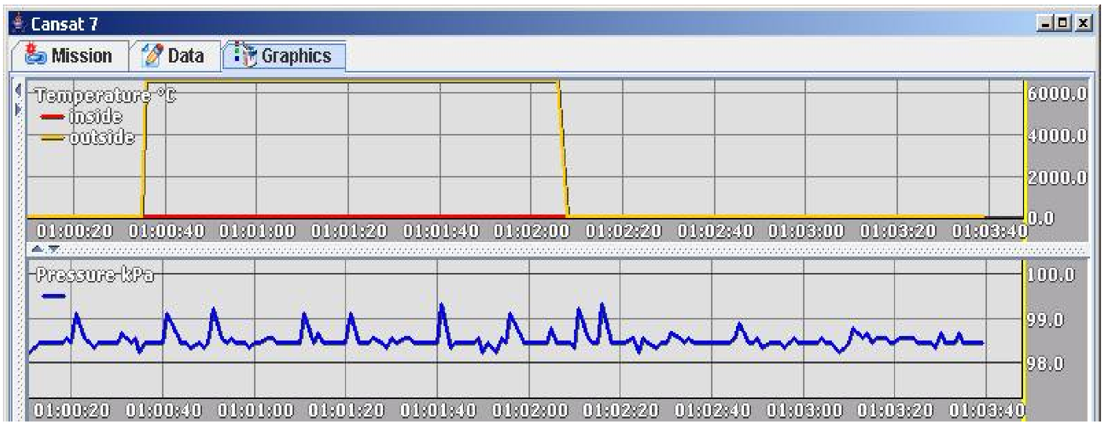
Simulating EMV conditions

We carry out the following test: We switch on the cansat device, and place it inside of the freezer of a fridge for about 2 min. The cansat is operated via a 9V battery. During this time, there cannot exist communication with groundstation. The “atmospheric data” is stored to the EEPROM and sent to groundstation, when we remove the cansat from the fridge.

Result: When we remove the cansat from the fridge, we obtain the following past data from the "past", which is the desired behaviour.

Time	Satellites	Latitude	Longitude	T_INS [°C]	T_OUT [°C]	Pres [kPa]	Comment
00:00:24.045	0	49.465466 N	9.577039 E	24.0	10.0	98.44618	#30
00:00:25.042	0	49.465466 N	9.577039 E	24.0	9.0	98.33768	#31
00:00:26.044	0	49.465466 N	9.577039 E	24.0	7.5	98.44618	#32
00:00:28.042	0	49.465466 N	9.577039 E	24.0	6.0	98.44618	#33
00:00:29.045	0	49.465466 N	9.577039 E	24.0	5.0	98.44618	#34
00:00:30.045	0	49.465466 N	9.577039 E	24.0	4.0	98.44618	#35
00:00:31.042	0	49.465466 N	9.577039 E	24.0	3.0	98.66319	#36
00:00:33.045	0	49.465466 N	9.577039 E	24.0	1.5	98.44618	#37
00:00:34.042	0	49.465466 N	9.577039 E	24.0	1.0	98.55469	#38
00:00:35.042	0	49.465466 N	9.577039 E	24.0	0.0	98.229164	#39
00:00:36.047	0	49.465466 N	9.577039 E	24.0	6552.6	98.44618	#40
00:00:38.046	0	49.465466 N	9.577039 E	24.0	6550.1	98.44618	#41
00:00:39.045	0	49.465466 N	9.577039 E	24.0	6550.1	98.44618	#42
00:00:40.042	0	49.465466 N	9.577039 E	24.0	6549.6	98.44618	#43
00:00:41.044	0	49.465466 N	9.577039 E	24.0	6549.6	99.09722	#44
00:00:43.047	0	49.465466 N	9.577039 E	24.0	6549.6	98.66319	#45
00:00:44.045	0	49.465466 N	9.577039 E	24.0	6548.1	98.44618	#46
00:00:45.045	0	49.465466 N	9.577039 E	24.0	6547.6	98.44618	#47
00:00:46.044	0	49.465466 N	9.577039 E	24.0	6546.6	98.33768	#48

As one can see, the outside temperature sensor measures a quick drop of about 1 °C per second. However, when the temperature drops below 0 °C, we do not convert the sensor values correctly. This ruins our plot as well:

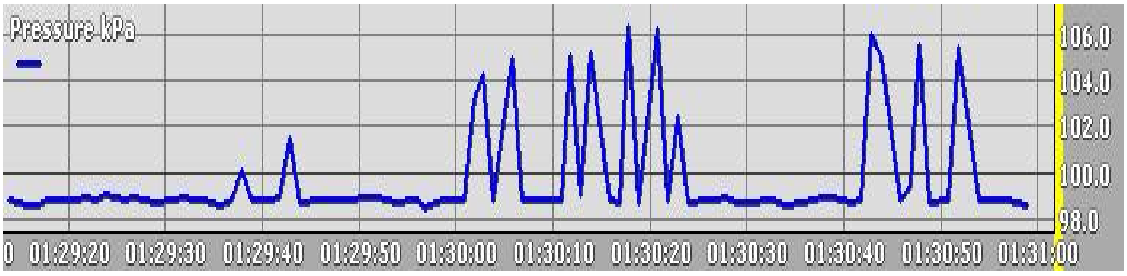


In later test, we show that this problem has been overcome.

Pressure variation

To test the pressure sensor we need a piece of rubber or soft plastic hose no more than a quarter inch in diameter. A pet store that sells fish has the ideal size air hose. Run the program that continuously displays the pressure readings. Cut a piece about 6 inches to a foot long. Looking at the sensor board, place one end of the hose over the hole on the metal side of the sensor. With the free side of the hose, put the air through with your mouth. You should see the pressure reading increase. If not, readjust the hose on the pressure sensor until you have a better seal. Make sure the end of the hose is flat and smooth.

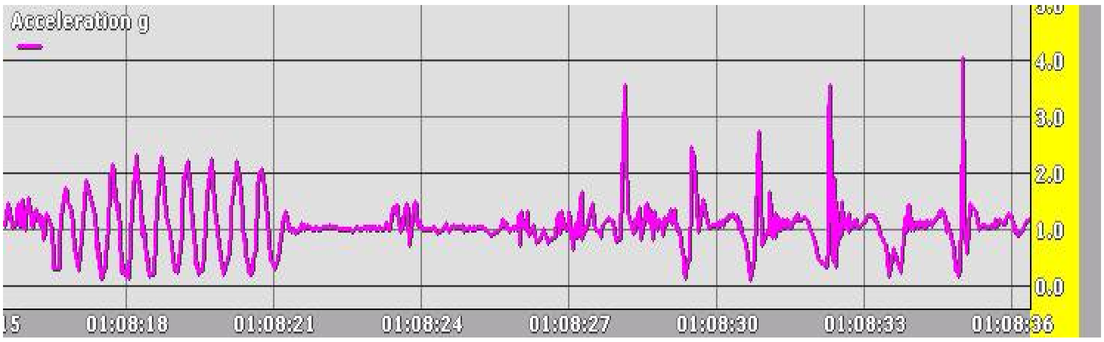
As the graph shows, using this technique we are able to increase the pressure by 8 kPa.



Motions and vibration

The following recording of the accelerometer originates from

- waving the cansat (01:08:17-01:08:21) and
- shocking the cansat (01:08:27-01:08:36) in equidistant time intervals.



Unfortunately, we do not have a reference to cross check the results.

Calibration of the satellite images

We use the satellite images Google provides, to enhance the monitoring of the position of the cansat. We do not access the web to acquire the images, instead we take screen shots and integrate them in our program. Currently, our program features maps of

- the Earth
- the student housing at Peter-Schneider-Straße
- Robotics hall JMUW
- the student housing Galgenberg

Below, we have a view on a part of the Peter-Schneider-Straße, Würzburg. To obtain the GPS coordinates at the points we simply place our GPS sensor on the solid ground and let cansat record the messages. The values for longitude, and latitude as shown below were obtained having contact to over 9 GPS satellites. Thus, we may assume the values are accurate.

Assuming further that the Earth is locally flat, we may use linear interpolation to map different GPS coordinates onto the map. The scaling factors do also apply when the cansat is at the robotics hall.

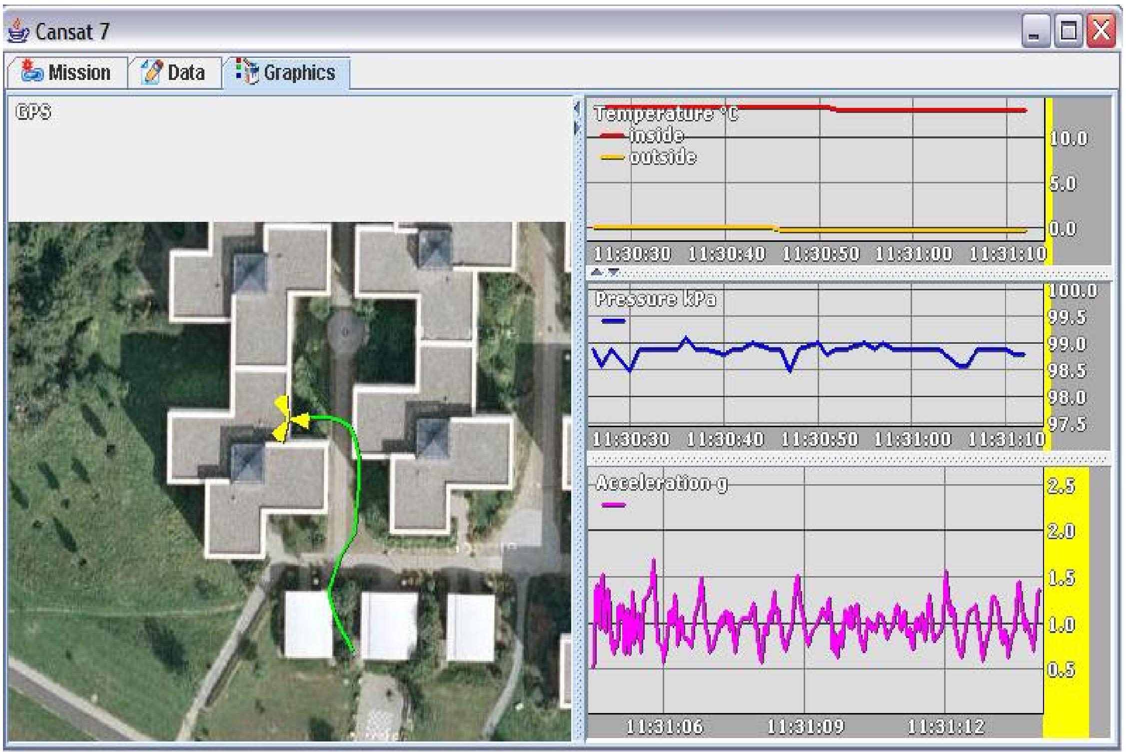
Note that Google does provide GPS coordinates, but we have found them to be erroneous.

Performing this experiment, we have noticed that in open space, away from buildings the GPS receiver might have contact to up to 12 satellites. Whereas the sensor accuracy of the device, when placed close to a building is poor.

System evaluation at Galgenberg

The Cansat was carried along in a certain path to track the position of the system on the Map. The communication system was tested using a alarm fixed to the cansat. The alarm was switched on and off frequently to check the sending and receiving of the data. The other parameters as the temperature and pressure were also monitored. The communication between the ground station and the Cansat was perfect till a range of 100m and the uplink was fluctuating beyond 120m approximately but the downlink was working perfect. The screenshots shown the test results. The inference from the test was, that the fluctuation in the uplink was due to the antenna or may be due to the delay caused by USB to COM conversion. The trajectory of the system was plotted and as shown in the below figure.

Screen shot of Cansat 7 right after the launch of the mission. The GUI shows the difference in reading from temp sensors.



Screen shot of Cansat 7 right before the end of the mission.

CHAPTER VIII

8.1Conclusions

This final report summarizes all milestones before. However, we did not take the copy paste approach to add the material. Instead, we have digested the material and reformulated the paragraphs and pictures.

We put emphasis on the new stuff that have been added to the Cansat 7 complete characteristics. The can and the pcb are mounted now, and we are ready for the presentation.

We hope that during the final presentation our mission sample will work. So far during no testing, the mission yielded for any reason. However, this still not allows us to believe our software is free of bugs.

We have heard that other groups have problems to make their cansat work. Look at all our nice screen shots we have painted. If one neglects our documentation skills we really deserve the trophy.

Our protocol is very advanced. We think of publishing the protocol in some computer graphics journal. But the time is limited and so far nobody wants to write it so that we meet the requirements.

Please don't make us redo the report. We have put all our skill and force and talent to design and make the pictures. The math we explain is very advanced. Our internal interpersonally group structure has been very advantageous. We did not have a fight and the work load has been evenly distributed. This will also show in our presentation, because everybody, each and one after another, will speak about an issue that has been raised during the cansat creating time.

CHAPTER VIII

REFERENCE

- [ATmega128] 8-bit AVR Microcontroller with 128K Bytes In-System Programmable Flash, Atmel Corporation, 2004
- [CP2102] Single-Chip USB to UART Bridge}, Silicon Laboratories, 2004
- [FM75] Low Voltage 2-Wire Digital Temperature Sensor with Thermal Alarm, www.fairchildsemi.com
- [GR213] HOLUX GR-213, GPS Receiver, User's guide, HOLUX Technology Inc., 2005
- [Kr00] Guido Krueger: Go To Java 2, Handbuch der Java-Programmierung, 2. Auflage, Addison-Wesley, 2000
- [JP7T] Fadil Beqiri: JP7-T Family GPS-Receiver Hardware description}, FALCOM GmbH, www.falcom.de
- [MAX3232] 3.0V to 5.5V, Low-Power, up to 1Mbps, True RS-232, Transceivers Using Four External Capacitors, Maxim Integrated Products, 1999
- [MPX4115A] Integrated Silicon Pressure Sensor for Manifold Absolute Pressure, Altimeter or Barometer Applications, Motorola, Inc., 2001, www.freescale.com
- [RT433F4] Infoblatt 433MHz-FM-Mehrkanal-Transceiver, Ingenieurbuero fuer Elektronik, Wildpoldsried, 2004, www.funkmodul.com
- [TS2940] 1A Ultra Low Dropout Fixed Positive Voltage Regulator, TSC, 2003
- [WQ05] Wang Hankang, Li Quancheng, Jiang Hai: CDMA based distribution power net automatization and data transmission system}, Power Engineering Conference, 2005. IPEC 2005. The 7th International.