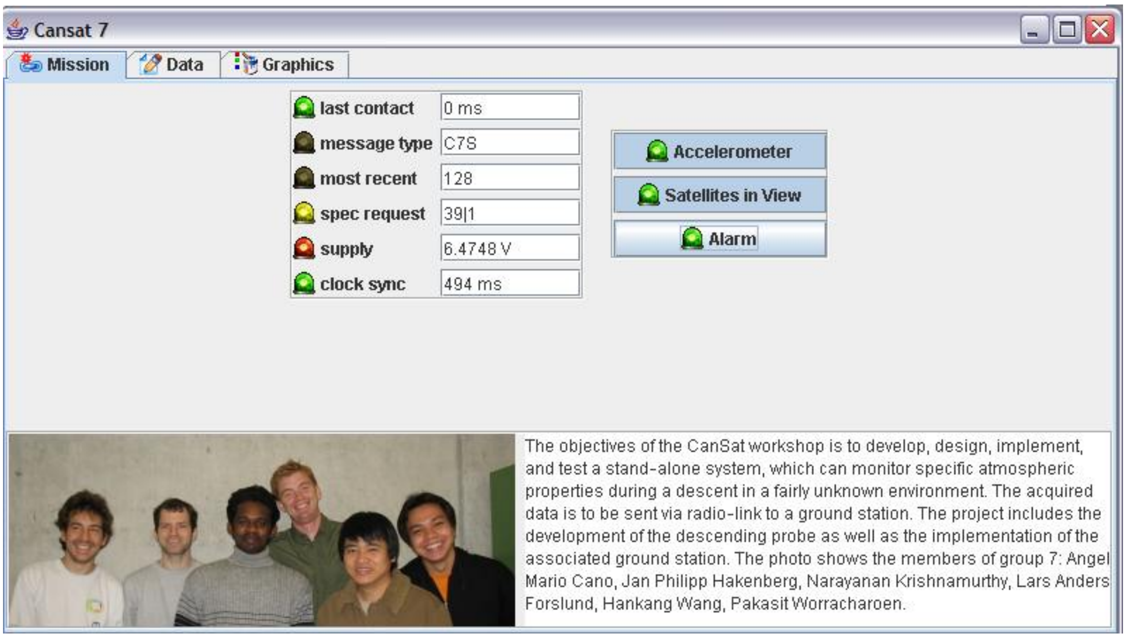
23 January 2007

Cansat · Milestone III

Test and Evaluation



Squad 7

Cano, Angel Mario	1543440
Forslund, Lars Anders	1543952
Hakenberg, Jan Philipp	1537190
Krishnamurthy, Narayanan	1543083
Wang, Hankang	1543524
Worracharoen, Pakasit	1543191

Table of Contents

Introduction

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

In the first chapter of 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.

Project management

The below table updates the status of our project. The team members are contributing effectively to their assigned responsiblities. Even though, we had delay in accomplishing the milestone 2, all the members in our group were still able to catch up with the schedule to complete the milestone. Our group has completed the basic requirement needed to complete the project and currently we are planning to improve the system to achieve the first spot.

Scheduling

			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																	■			

Status of project

STATUS		O C T	N O V	D E C	J A N

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

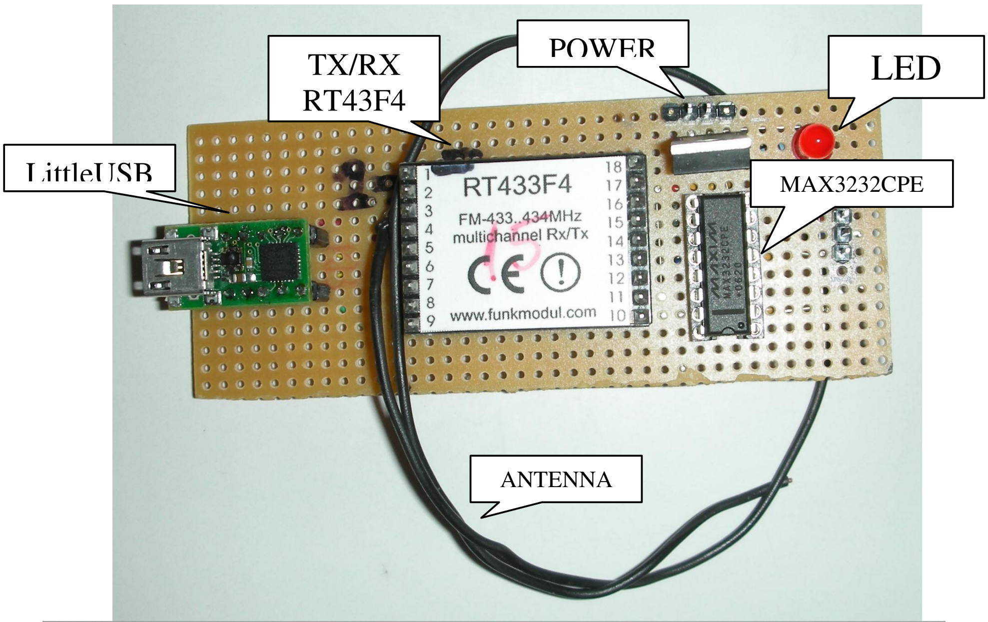
S.No	Activity	Responsiblity	Date	Status	Remarks	
1	**M1: Mission Analysis and Planning**					
	Define Task	TEAM	10.10.06		Completed	
	Allocation of task	TEAM	16.10.06		Completed	
	Time schedule/ Work Plan	TEAM	20.10.06			
	Identify the Different Subsystems	TEAM	23.10.06		Completed	
	System Architecture	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- CLEA
	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**			

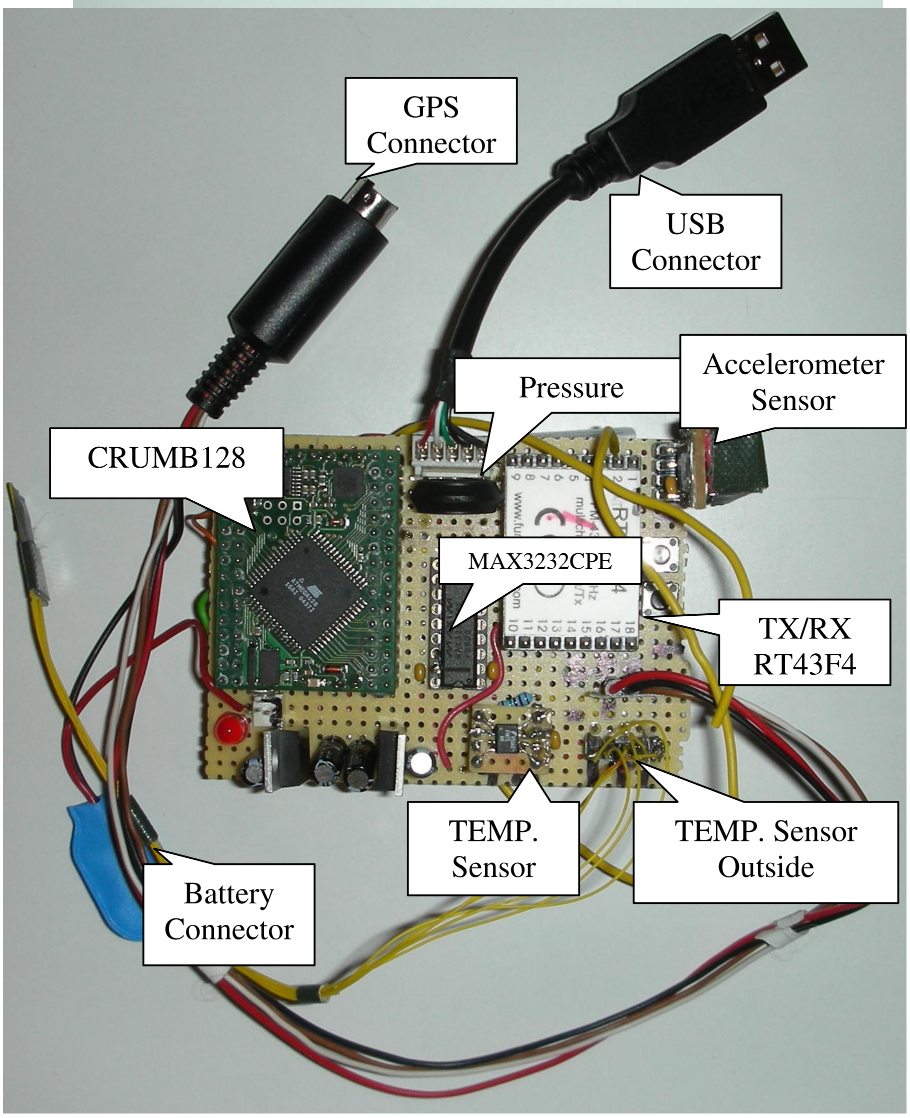
Log of debugging procedures

3W 1H for CANSAT Project

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	To 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).

State of the hardware





Error and accuracy calculations

Below, we discuss what are the errors that might be inherent in the measurements of the sensors, we have installed on our cansat. We discuss every sensor separately.

Pressure sensor

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.

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.

Accelerometer

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

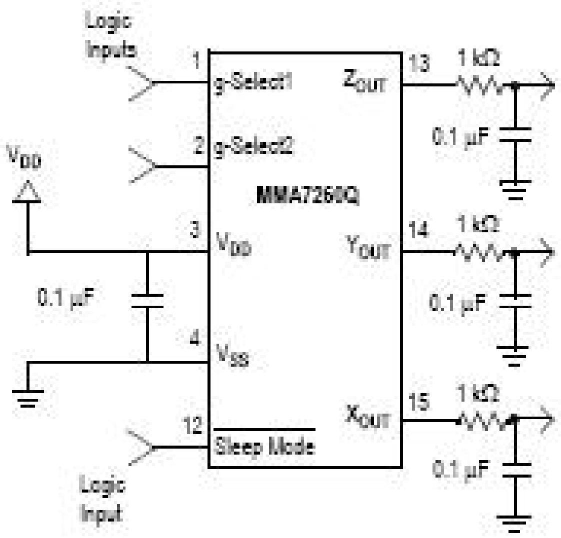


Figure 5. Accelerometer with Recommended Connection Diagram

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.

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;
}
```

GPS

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.

Material Selection

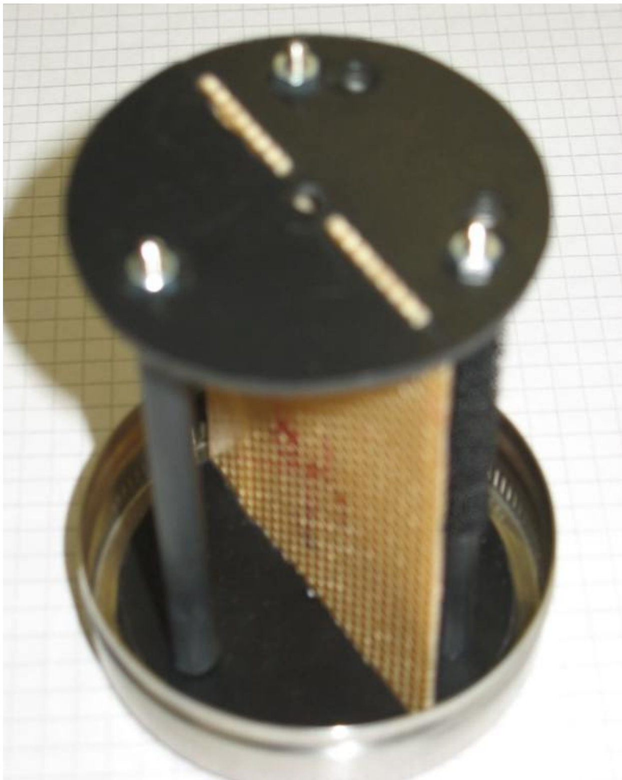
Steel was chosen for the material .Steel has a high yield strength (280-600 Mpa), it has a much greater density (7850 kg/m^3) than aluminum, making it desirable from a structural strengh perspective. Other possible materials are much more expensive than aluminum and steel.
Preliminary calculations have 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.

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

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.



Special features

We were able to encorporate several extra features to our cansat. However, they do not influence the overall mission goal in a negative way.

Power supply indicator

Our cansat sucks about 0.2 A. When we run the cansat with a 9V battery, it is of great advantage to know the voltage provided by the battery. The voltage measurement is implemented by a resistor bridge that puts a voltage on an analog-to-digital-converter pin of the microcontroller. As a consequence, the maximum supply voltage is 24 V.

On the title image, we show, how the supply voltage is displayed on the mission control graphical user interface.

Alarm

We have installed a speaker on the cansat. From the groundstation, we can uplink a command to switch the speaker on, or off. While performing the tests, we used this signaling feature, to indicate the person, who was carrying the cansat, to stop walking.

It is also a good criterion to quickly check, when the uplink stops, or restarts working.

EEPROM storage

We are able to store athmospheric data for more than 1 min 45 sec at a sampling rate of 1 frame/sec. That means we can bridge a fairly long time interval of downlink failure. However, even if this time interval is extended, the groundstation adapts to the situation, and requests messages that the cansat is able to provide.

Reliability and stabilty

Reliability and stability is inherent in our design.

- There exist no possibility for any device (cansat, or groundstation) to remain in a stalling state.
- If messages are not transmitted, they can either be considered not important, such as
    1. real time accelerometer information
    2. evoke uplink from grounstation
    3. satellites in view

  or they will be retransmitted, such as
    1. athmospheric data

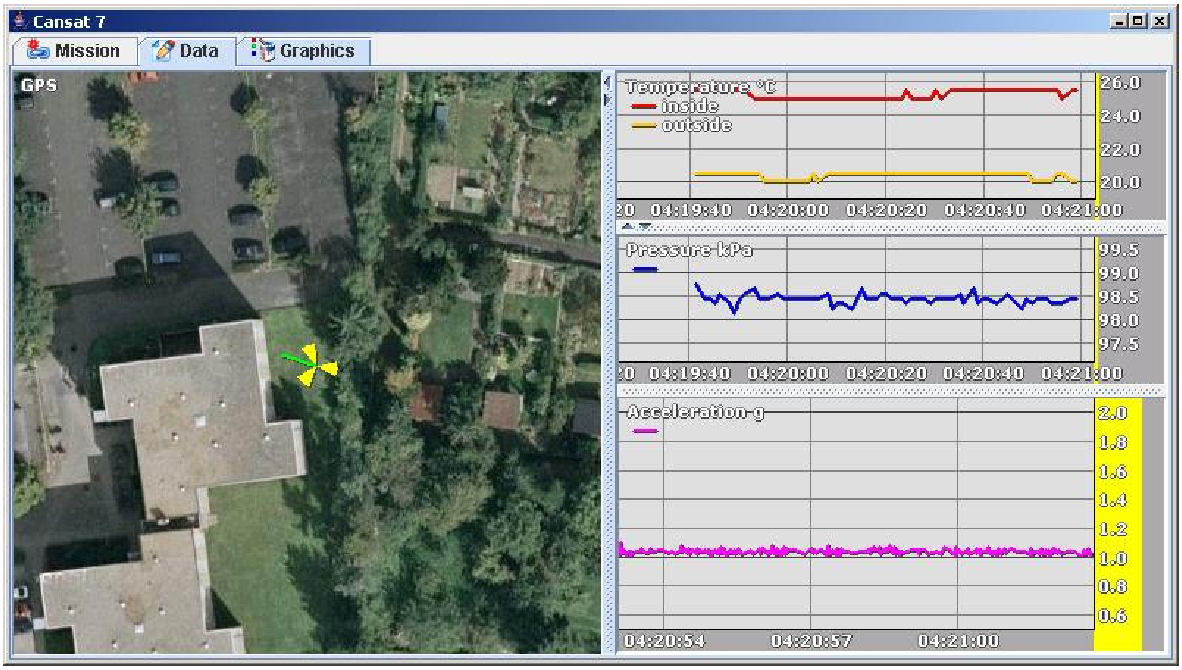
<u>In all test performed, we never suffered a crash.</u>

Improvements since the second milestone

Monitoring of the sensor data

Before we discuss the testing of the sensors, we demonstrate, how the Groundstation software allows us to monitor the sensor values.

The graphics are self explanatory. The data appears in principle as shown below.



The current position, supplied by GPS, is centered smoothly upon update. We have explained the method in our previous report.

The temperature and pressure is plotted on a different time scale than the acceleration. The high resolution monitor of the acceleration meter is required to capture sharp events such as shocks.

Communication design, final version

In this section, we describe

- the *downlink*, i.e. the transmission of data from the cansat to the groundstation, and
- the *uplink*, vice versa

Our group is assigned the frequency 434.11 MHz. Since the hardware can use only one frequency at the same time, data cannot be transmitted and sent simulaneously.

Previous testing has shown, that if the transmission is noticed, the received bytes are rarely corrupted. However, to check the coherency of the messages, we append a checksum to each message.

After submitting the second milestone, we have decided to add an accelerometer to our cansat. Thus, we made slight modifications to our protocol.

Downlink:

Overview:

bytes	header	body	interpretation	frequency
42	C7S	msgL msgH tinsL tinsH toutL toutH presL presH GPS xor	contains athmospheric data	1000 ms
10	C7A	accxL accxH accyL accyH acczL acczH xor	accelerometer values	10 ms

60	C7V	$GPGSV (no checksum)	satellites in view	3000 ms
7	C7L	portB supplyL supplyH xor	portB and supply voltage; initiates uplink from groundstation	1000 ms

Once the GPS frame with time, longitude, and latitude ($GPGGA) arrives the microcontroller from the GPS sensor, we read temperature and pressure values as well. The overall information is stored in the EEPROM (4096 byte) of the microcontroller. GPS information ($GPGGA) arrives about every second. The frame we store into EEPROM consist of 38 byte. The cansat stores up to 107 athmospheric frames into the memory, before data is lost. This means, we are able to bridge a communication failure of about 1 min 45 sec.

While the GPS sensor is not responding, we fire the accelerometer values to the groundstation.

The $GPGSV header is comprised within the GPS sensor message approximately every 3 sec. Then, we send the C7V frame, which essentically consists of the lists of satellites in view. To process this information at the groundstation is subject to future work – but not later than our final presentation.

The C7L frame encodes the status of portB, but also the supply voltage. When operating the cansat by a 9V battery, the monitoring of the supply voltage at the groundstation is extremely useful.
However, the C7L frame also initiates the uplink commands by the groundstation. Meanwhile, the cansat will not send any messages for about 75 ms.

Uplink:

Overview:

bytes	header	body	interpretation	frequency
8	C7M	portB ident reqL reqH xor	portB, command id, request value	1000 ms

As stated above, upon receiving the C7L frame the groundstation performs the uplink. For simplicity, there is only the C7M frame to parse. Via this frame, the groundstation can operate the devices at portB:

- operate accelerometer (on/off), which is useful in case power is an issue
- downlink "satellites in view" C7V frame
- switch on-board speaker (on/off)

At the groundstation, the devices are controled from the following graphical user interface. The LEDs in the JToggleButton[]s indicate the actual state of each device, which was last confirmed by the cansat.

Lets assume, that there has been a communication failure in the past, and several messages have not reached the groundstation. Then by setting the integer req equal to the index of one of the missing messages, will cause the cansat to not only send the most recent athmospheric data, but also the message req, which has been recorded in the past. The value ident is a measure how urgent this request is, as we will see in the following example:

Cansat 7

Mission | Data | Graphics

Time	Satellites	Latitude	Longitude	T_INS [°C]	T_OUT [°C]	Pres [kPa]	Comment
15:25:53.212	0	49.46556 N	9.576508 E	25.5	19.0	98.44618	#21
15:25:54.212	0	49.46556 N	9.576508 E	25.5	19.0	98.55469	#22
15:25:55.212	0	49.46556 N	9.576508 E	25.5	19.0	98.44618	#23
15:25:56.212	0	49.46556 N	9.576508 E	25.5	19.0	98.66319	#24
15:25:58.212	0	49.46556 N	9.576508 E	25.5	19.0	98.33768	#25
15:26:11.212	0	49.46556 N	9.576508 E	25.5	19.0	98.44618	#36
15:26:13.212	0	49.46556 N	9.576508 E	25.5	19.0	98.44618	#37
15:26:14.212	0	49.46556 N	9.576508 E	25.5	19.0	98.33768	#38
15:26:15.212	0	49.46556 N	9.576508 E	25.5	19.0	98.44618	#39

Here, the messages from 26 to 35 are (still) missing. The current implementation of the grounstation, will set

```
ident=2, req=26
```

If the message reaches the cansat, the satellite will send the current athmospheric data frame (of index >39) followed by the athmospheric data of index 26, and 27. Ident is the number of messages that cansat will additionally send.

Antennas

In order to communicate between the ground station and the cansat, antennas are required. The satellite and ground station use the same antenna type.

We are sending on the frequency of 434.11 MHz. We apply the wavelength over two rule, and thus come to a length of the antenna of about 35 cm.

Data Flow

The following performance analysis shows, that our C implementation on the microcontroller, as well as the design of our communication – on a more abstract level - suits our hardware ideally. The main objective concerning the data flow, is to ensure that the sampling, and transmitting frequency of the accelerometer does not drop below a certain value.

The table on the next page protocols the traffic between cansat and groundstation. We protocol

- the header of the message,
- the time of recording (at groundstation) after launch in msec
- the time between two subsequent messages

The last information is redundant, however, makes the table easier to read. For instance

...	
C7A 3906 15	accelerometer values were received 3906 msec after launch
C7V 4016 110	"satellites in view" were received 4016 msec after launch, which is 110 msec after the previous message
...	

In the table, we highlight all messages except the accelerometer frame. This gives a good picture of the relations between real-time data, and single atmospheric frames.

C7A 1375 672	C7A 2844 94	C7A 4750 31	C7A 5875 0	C7A 5594 0	C7A 7078 15	**C7M** 8891 0
C7A 1391 16	**C7S** 2906 62	**C7S** 4781 31	C7A 5891 16	C7A 5609 15	C7A 7109 31	C7A 9000 109
C7A 1453 62	C7A 2922 16	C7A 4797 16	C7A 5906 15	C7A 5609 0	C7A 7125 16	C7A 9016 16
C7A 1547 94	**C7S** 2969 47	**C7L** 4813 16	C7A 5922 16	C7A 5625 16	C7A 7125 0	C7A 9016 0
C7A 1656 109	C7A 2984 15	**C7M** 4813 0	C7A 5938 16	C7A 5641 16	C7A 7141 16	C7A 9031 15
C7S 1719 63	**C7S** 3031 47	C7A 4922 109	C7A 5953 15	C7A 5641 0	C7A 7172 31	C7A 9047 16
C7A 1719 0	C7A 3047 16	C7A 4938 16	C7A 3906 15	C7A 5656 15	C7A 7172 0	C7A 9047 0
C7L 1734 15	**C7L** 3047 0	C7A 4953 15	**C7V** 4016 110	C7A 5688 32	C7A 7188 16	C7A 9063 16
C7M 1734 0	**C7M** 3047 0	C7A 4984 31	C7A 4031 15	C7A 5688 0	C7A 7188 0	C7A 9094 31
C7A 1844 110	C7A 3172 125	C7A 5000 16	C7A 4047 16	C7A 5703 15	C7A 7203 15	C7A 9109 15
C7A 1875 31	C7A 3188 16	C7A 5000 0	C7A 4063 16	C7A 5703 0	C7A 7266 63	C7A 9109 0
C7A 1875 0	C7A 3203 15	C7A 5016 16	C7A 4078 15	C7A 5719 16	C7A 7344 78	C7A 9109 0
C7A 1891 16	C7A 3219 16	C7A 5016 0	C7A 4078 0	C7A 5734 15	C7A 7422 78	C7A 9125 16
C7A 1906 15	C7A 3234 15	C7A 5031 15	**C7V** 4172 94	C7A 5750 16	C7A 7516 94	C7A 9141 16
C7A 1906 0	C7A 3234 0	C7A 5047 16	C7A 4188 16	C7A 5750 0	**C7S** 7578 62	C7A 9141 0
C7A 1922 16	C7A 3250 16	C7A 5078 31	C7A 4203 15	C7A 5766 16	C7A 7578 0	C7A 9156 15
C7A 1938 16	C7A 3250 0	C7A 5078 0	C7A 4203 0	C7A 5781 15	**C7S** 7609 31	C7A 9156 0
C7A 1953 15	C7A 3266 16	C7A 5094 16	C7A 4281 78	C7A 5797 16	C7A 7656 47	C7A 9188 32
C7A 1953 0	C7A 3297 31	C7A 5094 0	C7A 4406 125	C7A 5797 0	**C7S** 7703 47	C7A 9188 0
C7A 1969 16	C7A 3297 0	C7A 5109 15	C7A 4500 94	C7A 5813 16	C7A 7703 0	C7A 9203 15
C7A 1969 0	C7A 3313 16	C7A 5141 32	C7A 4594 94	C7A 5828 15	**C7L** 7719 16	C7A 9203 0
C7A 1984 15	C7A 3328 15	C7A 5172 31	**C7S** 4656 62	C7A 5844 16	**C7M** 7719 0	C7A 9219 16
C7A 1984 0	C7A 3328 0	C7A 5172 0	C7A 4672 16	C7A 5844 0	C7A 7828 109	C7A 9234 15
C7A 2016 32	C7A 3344 16	C7A 5188 16	**C7S** 4719 47	C7A 5875 31	C7A 7828 0	C7A 9234 0
C7A 2031 15	C7A 3359 15	C7A 5188 0	C7A 4750 31	C7A 5875 0	C7A 7844 16	C7A 9250 16
C7A 2047 16	C7A 3375 16	C7A 5203 15	**C7S** 4781 31	C7A 5891 16	C7A 7844 0	C7A 9266 16
C7A 2047 0	C7A 3375 0	C7A 5219 16	C7A 4797 16	C7A 5906 15	C7A 7875 31	C7A 9281 15
C7A 2063 16	C7A 3391 16	C7A 5234 15	**C7L** 4813 16	C7A 5922 16	C7A 7875 0	C7A 9281 0
C7A 2078 15	C7A 3391 0	C7A 5234 0	**C7M** 4813 0	C7A 5938 16	C7A 7906 31	C7A 9297 16
C7A 2094 16	C7A 3406 15	C7A 5250 16	C7A 4922 109	C7A 5953 15	C7A 7906 0	C7A 9313 16
C7A 2094 0	C7A 3438 32	C7A 5266 16	C7A 4938 16	C7A 5953 0	C7A 7922 16	C7A 9313 0
C7A 2109 15	C7A 3438 0	C7A 5266 0	C7A 4953 15	C7A 5969 16	C7A 7938 16	C7A 9328 15
C7A 2109 0	C7A 3453 15	C7A 5281 15	C7A 4984 31	C7A 5969 0	C7A 7938 0	C7A 9328 0
C7A 2125 16	C7A 3469 16	C7A 5313 32	C7A 5000 16	C7A 5984 15	C7A 7953 15	C7A 9344 16
C7A 2156 31	C7A 3484 15	C7A 5313 0	C7A 5000 0	C7A 5984 0	C7A 7969 16	C7A 9359 15
C7A 2156 0	C7A 3484 0	C7A 5328 15	C7A 5016 16	C7A 6016 32	C7A 7969 0	C7A 9375 16
C7A 2203 47	C7A 3500 16	C7A 5328 0	C7A 5016 0	C7A 6016 0	C7A 7984 15	C7A 9375 0
C7A 2203 0	C7A 3516 16	C7A 5344 16	C7A 5031 15	C7A 6031 15	C7A 7984 0	C7A 9391 16
C7A 2203 0	C7A 3516 0	C7A 5359 15	C7A 5047 16	C7A 6031 0	C7A 8000 16	C7A 9406 15
C7A 2203 0	C7A 3531 15	C7A 5359 0	C7A 5078 31	C7A 6109 78	C7A 8016 16	C7A 9406 0
C7A 2219 16	C7A 3531 0	C7A 5375 16	C7A 5078 0	C7A 6203 94	C7A 8031 15	C7A 9422 16
C7A 2250 16	C7A 3563 32	C7A 5391 16	C7A 5094 16	C7A 6281 78	C7A 8031 0	C7A 9438 16
C7A 2250 0	C7A 3578 15	C7A 5406 15	C7A 5094 0	C7A 6391 110	C7A 8047 16	C7A 9438 0
C7A 2266 16	C7A 3578 0	C7A 5406 0	C7A 5109 15	**C7S** 6453 62	C7A 8047 0	C7A 9453 15
C7A 2266 0	C7A 3594 16	C7A 5422 16	C7A 5141 32	C7A 6453 0	C7A 8063 16	C7A 9469 16
C7A 2281 15	C7A 3609 15	C7A 5453 31	C7A 5172 31	**C7S** 6484 31	C7A 8078 15	C7A 9469 0
C7A 2297 16	C7A 3625 16	C7A 5453 0	C7A 5172 0	C7A 6516 32	C7A 8094 16	C7A 9484 15
C7A 2297 0	C7A 3625 0	C7A 5469 16	C7A 5188 16	**C7S** 6563 47	C7A 8094 0	C7A 9484 0
C7A 2328 31	C7A 3641 16	C7A 5484 15	C7A 5188 0	C7A 6578 15	C7A 8109 15	C7A 9500 16
C7A 2328 0	C7A 3672 31	C7A 5500 16	C7A 5203 15	**C7L** 6578 0	C7A 8109 0	C7A 9516 16
C7A 2344 16	C7A 3672 0	C7A 5500 0	C7A 5219 16	**C7M** 6578 0	C7A 8125 16	C7A 9531 15
C7A 2344 0	C7A 3672 0	C7A 5516 16	C7A 5234 15	C7A 6688 110	C7A 8141 16	C7A 9531 0
C7A 2344 0	C7A 3688 16	C7A 5516 0	C7A 5234 0	C7A 6703 15	C7A 8141 0	**C7V** 9625 94
C7A 2375 31	C7A 3703 15	C7A 5531 15	C7A 5250 16	C7A 6719 16	C7A 8156 15	C7A 9656 31
C7A 2391 16	C7A 3719 16	C7A 5547 16	C7A 5266 16	C7A 6719 0	C7A 8172 16	C7A 9656 0
C7A 2406 15	C7A 3719 0	C7A 5547 0	C7A 5266 0	C7A 6734 15	C7A 8188 16	C7A 9672 16
C7A 2406 0	C7A 3734 15	C7A 5563 16	C7A 5281 15	C7A 6750 16	C7A 8188 0	C7A 9672 0
C7A 2422 16	C7A 3750 16	C7A 5594 31	C7A 5313 32	C7A 6750 0	C7A 8203 15	C7A 9688 16
C7A 2438 16	C7A 3766 16	C7A 5594 0	C7A 5313 0	C7A 6766 16	C7A 8219 16	**C7V** 9813 125
C7A 2438 0	**C7V** 3844 78	C7A 5609 15	C7A 5328 15	C7A 6781 15	C7A 8219 0	C7A 9828 15
C7A 2469 31	C7A 3875 31	C7A 5609 0	C7A 5328 0	C7A 6797 16	C7A 8234 15	C7A 9844 16
C7A 2484 15	C7A 3875 0	C7A 5625 16	C7A 5344 16	C7A 6813 16	C7A 8234 0	C7A 9844 0
C7A 2297 16	C7A 3891 16	C7A 5641 16	C7A 5359 15	C7A 6813 0	C7A 8266 32	C7A 9859 15
C7A 2297 0	C7A 3891 0	C7A 5641 0	C7A 5359 0	C7A 6844 31	C7A 8266 0	C7A 9859 0
C7A 2328 31	C7A 3906 15	C7A 5656 15	C7A 5375 16	C7A 6859 15	C7A 8266 0	**C7V** 9953 94
C7A 2328 0	**C7V** 4016 110	C7A 5688 32	C7A 5391 16	C7A 6875 16	C7A 8281 15	C7A 9984 31
C7A 2344 16	C7A 4031 15	C7A 5688 0	C7A 5406 15	C7A 6875 0	C7A 8297 16	C7A 10000 16
C7A 2344 0	C7A 4047 16	C7A 5703 15	C7A 5406 0	C7A 6891 16	C7A 8313 16	C7A 10016 16
C7A 2344 0	C7A 4063 16	C7A 5703 0	C7A 5422 16	C7A 6906 15	C7A 8313 0	C7A 10063 47
C7A 2375 31	C7A 4078 15	C7A 5719 16	C7A 5453 31	C7A 6906 0	C7A 8328 15	C7A 10156 93
C7A 2391 16	C7A 4078 0	C7A 5734 15	C7A 5453 0	C7A 6938 16	C7A 8344 16	C7A 10250 94
C7A 2406 15	**C7V** 4172 94	C7A 5750 16	C7A 5469 16	C7A 6953 15	C7A 8406 47	C7A 10328 78
C7A 2406 0	C7A 4188 16	C7A 5750 0	C7A 5484 15	C7A 6953 0	C7A 8484 78	**C7S** 10406 47
C7A 2422 16	C7A 4203 15	C7A 5766 16	C7A 5500 16	C7A 6953 0	C7A 8578 94	C7A 10406 0
C7A 2438 16	C7A 4203 0	C7A 5781 15	C7A 5500 0	C7A 6953 0	C7A 8672 94	**C7S** 10453 47
C7A 2438 0	C7A 4281 78	C7A 5797 16	C7A 5516 16	C7A 7000 47	**C7S** 8719 47	C7A 10469 16
C7A 2469 31	C7A 4406 125	C7A 5797 0	C7A 5516 0	C7A 7000 0	C7A 8750 31	**C7S** 10531 62
C7A 2484 15	C7A 4500 94	C7A 5813 16	C7A 5531 15	C7A 7000 0	**C7S** 8781 31	C7A 10563 32
C7A 2500 16	C7A 4594 94	C7A 5828 15	C7A 5547 16	C7A 7031 15	C7A 8797 16	**C7L** 10563 0
C7A 2547 47	**C7S** 4656 62	C7A 5844 16	C7A 5547 0	C7A 7047 16	**C7S** 8859 62	**C7M** 10563 0
C7A 2656 109	C7A 4672 16	C7A 5844 0	C7A 5563 16	C7A 7063 16	C7A 8875 16	C7A 10656 93
C7A 2750 94	**C7S** 4719 47	C7A 5875 31	C7A 5594 31	C7A 7063 0	**C7L** 8891 16	C7A 10672 16

Testing

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.

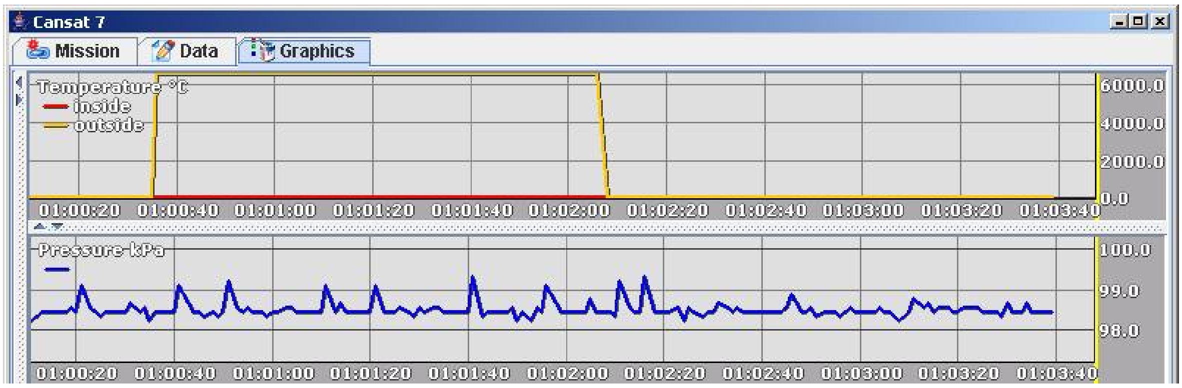
Simulating EMV conditions

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:

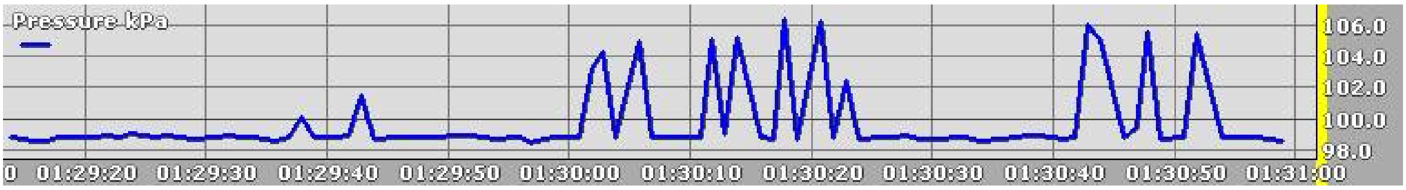


In later test, we show that this problem has been overcome.

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.

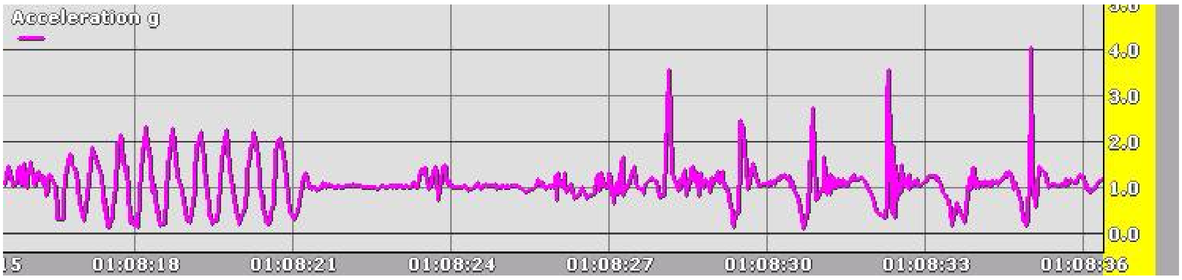
As the graph shows, using this technique we are able to increase the pressure by 8 kPa.



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.

Future work

Cansat breadboard

The breadboard we are currently using has been our testing/development board, the configuration has changed over time. The location of the components on the breadboard is not optimal, the length of the cables is too long, this could lead one of them to get disconnected, shortcut, etc. Besides, the size of the board is not optimal for the mechanical structure.

What will be done for improving is, now that we know how the final hardware will be, redesign the breadboard for optimal performance, implement connectors where there are not, make a cleaner hardware, which we plan will reduce potential error sources.

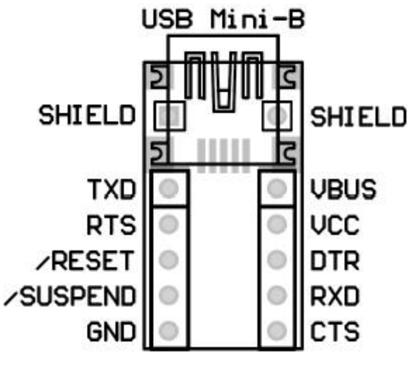
Groundstation circuit

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. The aim is that we will be able to implement it successfully for the final version.

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.



We try to put littleUSB into our circuit board. The problem is we can get the data from CanSat properly, but we cannot send the data from Ground Station to our CanSat. For further development, we will solve this problem.

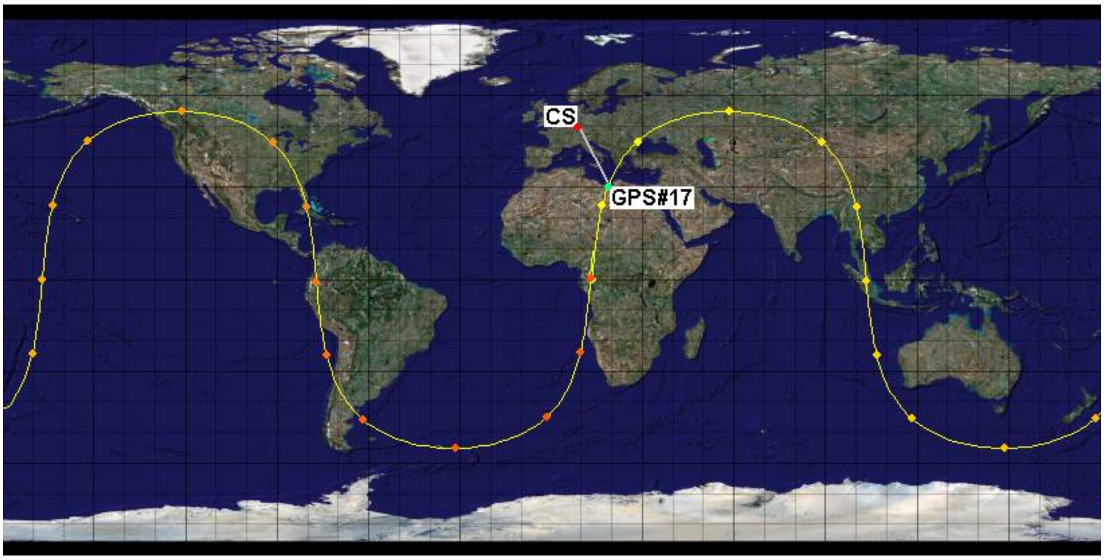
Satellites in view

Before the presentation is due, we intend to implement a nice view on the “Satellites in view”. This data is already sent by the cansat. However, the groundstation code does not process this data so far.

There exist 24 GPS satellites. Each cycles the Earth at an

- orbital radius of 26,600 km, with
- inclination 55°, and
- Eccentricity 0.

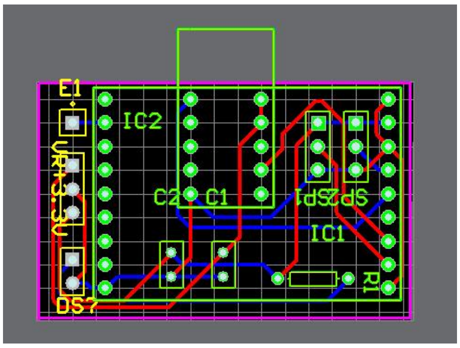
4 GPS satellites share an orbit each. From the $GPGSV frame that the GPS sensor fires about every 3 seconds, we are able to parse azimuth, elevation, and transmit power [dB] of each satellite that is in view of our sensor. Having this information, we can plot the evaluation of the orbits of the GPS satellites that are near to cansat.

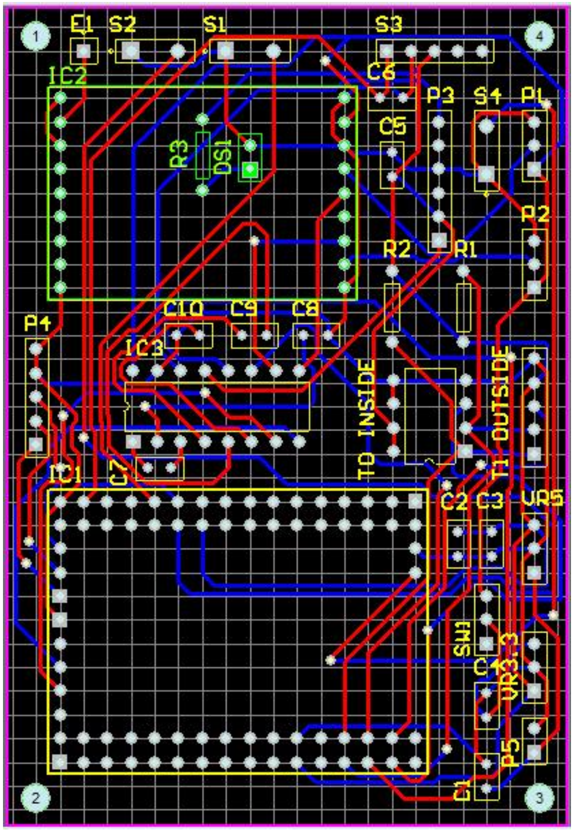


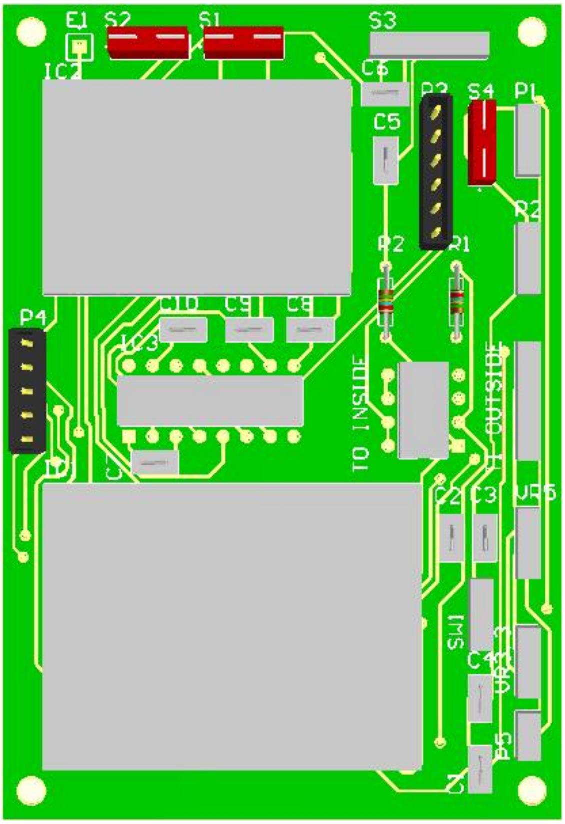
Printed Circuit Board

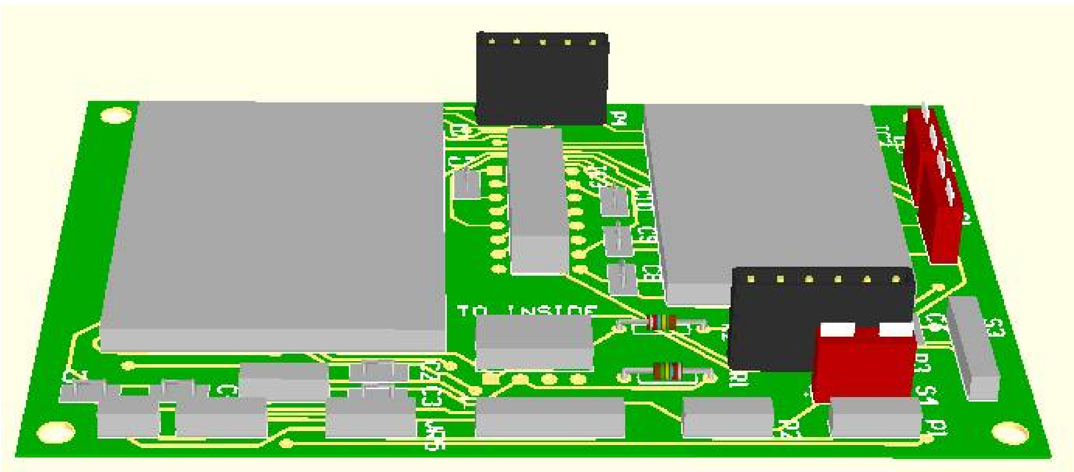
In fact, our team has designed PCB for CanSat circuit and ground station circuit. We created PCB by using Altium Designer2004. But because of some technical problems, we cannot send them to the company. At first, we sent it to Becker-Mueller company. Unfortunately, they cannot open our files. After that we send it to PCB-Pool company. They can accept our files. Unfortunately, it cannot be on time. Moreover, to order PCB in Germany costs so expensive. For further development, we will send our PCB to Thailand. It will costs much cheaper (10 times)

Here is our ground station PCB, and CanSat PCB.






E1
S2
S1
S3
IC2
C6
P3
S4
P1
C5
R2
R2
R1
P4
C10
C9
C8
IC3
TO INSIDE
TO OUTSIDE
C2
C3
VR5
SW1
C4
P5
C1


P4
TO INSIDE
R1
S4
P1
S3
R2
VR5

Conclusions

In this milestone, we have gathered all data, that was asked for: We discussed the accuracy of all sensors we have aboard the cansat. We report about the testing proceedures, that clearly prove, that our cansat meets the mission requirements.

Until the final presentation, our group will not rest to improve the cansat system design, to win the contest. We will distribute the future work that improves our cansat, evenly among the team members.

As all test have been a success, so also the final presentation will be. Our cansat deserves to take the trip to Kirnua, to be tested in high altitude, at challenging atmospheric conditions.

References

[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.