ISTANBUL TECNICAL UNIVERSITY

CANSAT 2011

TEAM HEZARFEN

258

CRITICAL DESIGN REPORT

March 29, 2011

Presentation Outline

2011 Texas Annual CanSat Competition

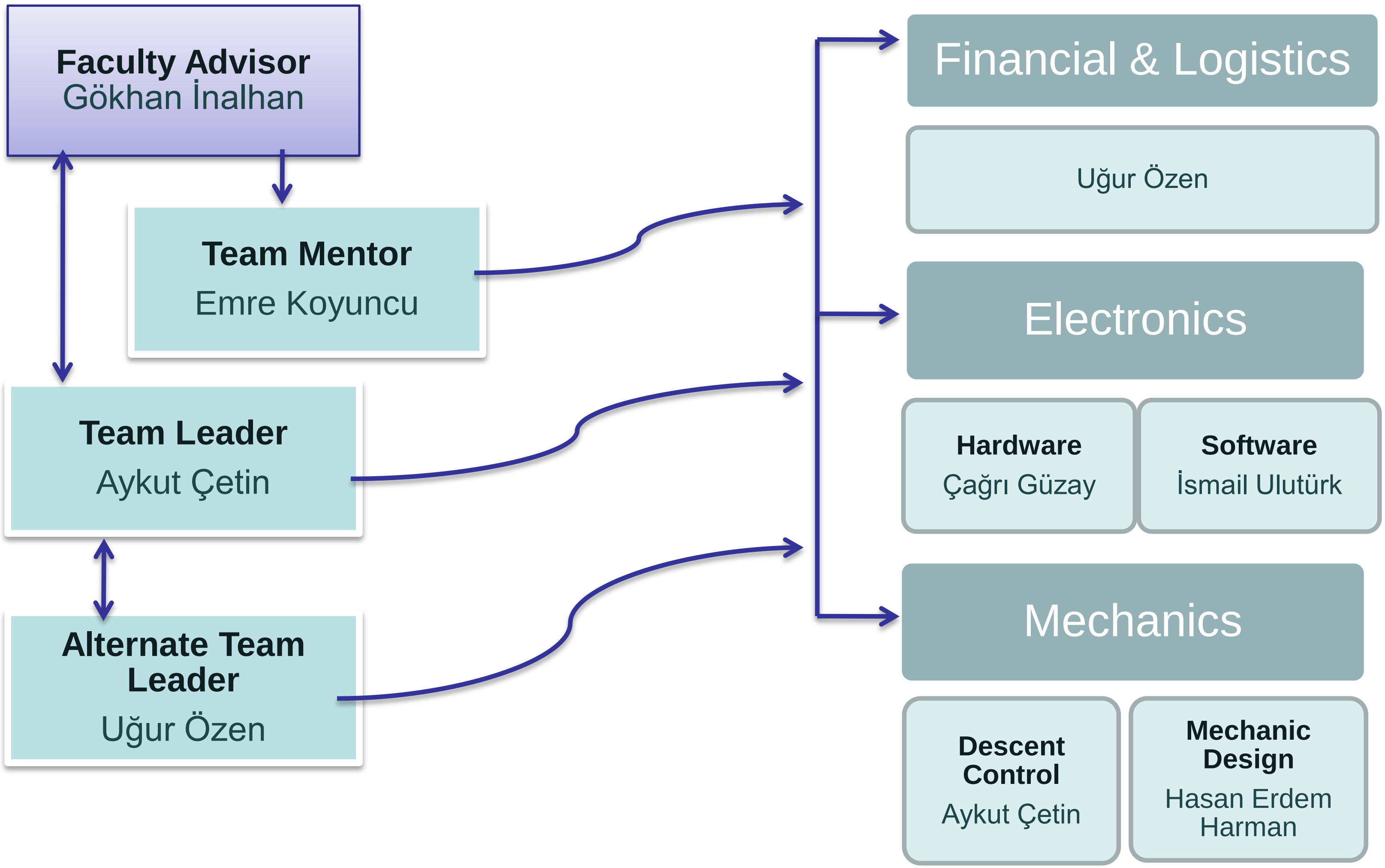
Content

Presentation Outline

Presentation Outline

NO	NAME	COURSE	POSITION
1	Emre Koyuncu	Graduate	Team Mentor
2	Aykut Çetin	Undergraduate	Team Leader Descent Control Authorized
3	Uğur Özen	Undergraduate	Alternate Team Leader Financial&Logistic Authorized
4	Çağrı Güzay	Undergraduate	Electronic-Hardware Authorized
5	İsmail Ulutürk	Undergraduate	Electronic-Software Authorized
6	Hasan Erdem Harman	Undergraduate	Mechanic Design Authorized

Team Organization



Acronyms

A → Analysis
ACL → Acceleration
ADC → Analog digital converter
ALT → Altitude
API → Application programming interface
CAM → Camera
CDH → Communication and Data Handling
D → Demonstration
DCS → Descent Control System Requirements
EEPROM → Electrically Erasable Programmable Read-Only Memory
EPS → Electric Power System
FSW → Flight Software
GPS → Global Positioning System
I →Inspection
MSR → Mechanical System Requirements
RF → Radio Frequency
SEN → Sensor Subsystem Requirement
SPI → Serial Peripheral Interface
SR → System Requirements
T → Test
TTL → Transistor - Transistor Logic
TEM → Temperature
UART →Universal synchronous asynchronous receiver/transmitter
VM → Verification Method

Systems Overview

Uğur ÖZEN

Mission Summary

The Main Obective:

→The main purpose of Cansat is that provide egg safety from launch to landing

Other Objectives:

→Launch Cansat

→Separate Cansat two parts; lander and carrier

→Control descent of lander speed of 3-5m/s

→Control descent of carrier speed of 4,5-6,5m/s

→Collect data using sensors to send ground station

Bonus Objective:

→Calculate lander collision force

Summary of Changes Since PDR

Electronic System Changes

→ We will use a MicroSD Card instead of SD Card in Lander because it is smaller with the same functionality.

→ We left the camera out, because we choose measuring the impact force of the lander as bonus objective.

→ Slight changes in the data packet format.

→ For Carrier, radio module (AC4790) is fed by a seperate regulator. Others are fed from microcontroller board.

Mechanic System Changes

→ Satellite diameter is decreased from 72 mm to 68 mm.

→ We will use body restricted mechanism instead of magnets because of it's low cost,simplicity and reliability.

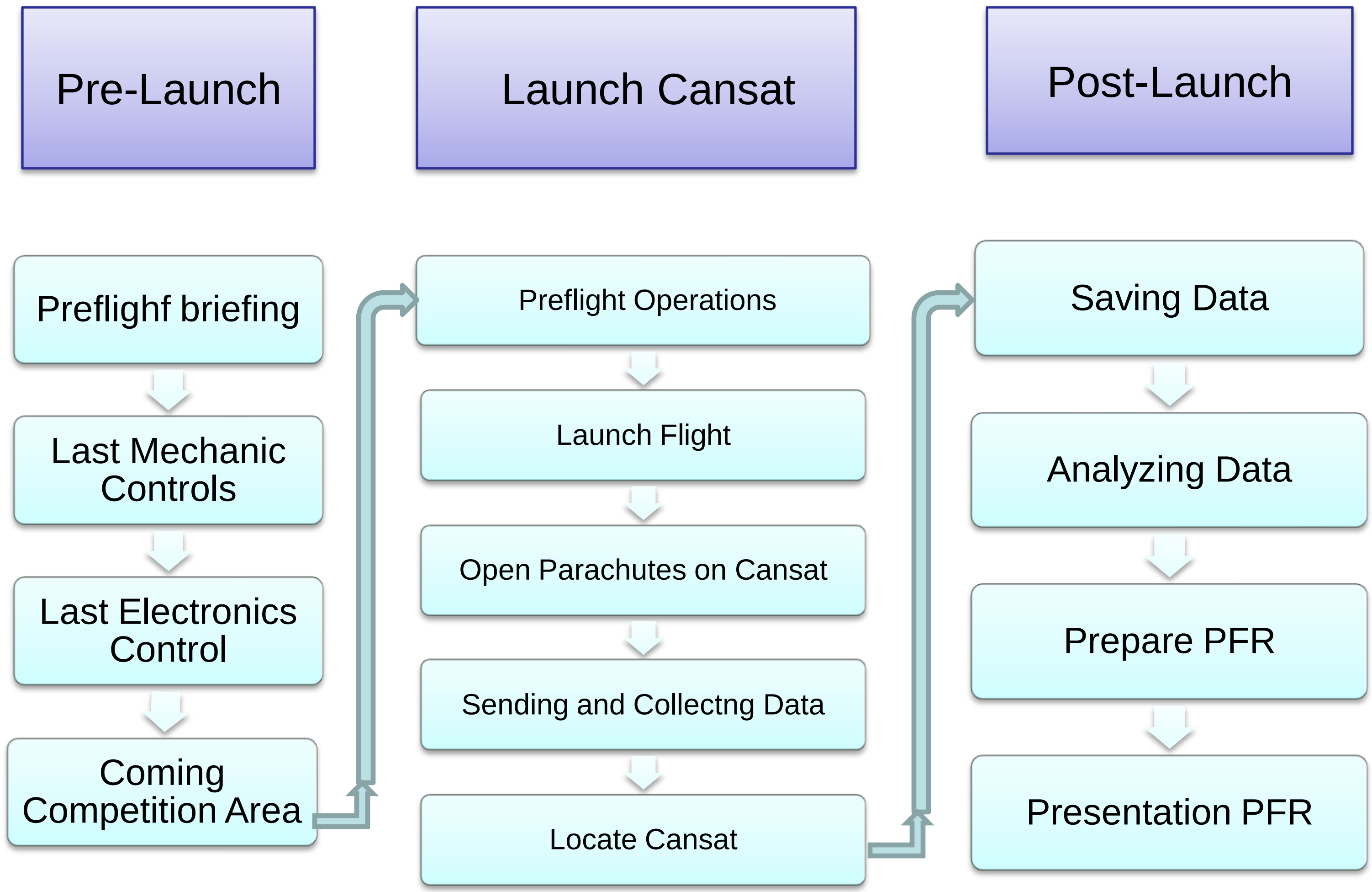
System Requirements

ID	Reqirements	Rationale	Priority	Parent	Children	VM			
						A	I	T	D
SR-01	Total mass of cansat shall not be more than 500gr. (exculuding egg)	competition requirement	High		MSR-01		X	X	
SR-02	The cansat shall fit inside the cylindirical of 72mm diameters 279mm in length.	competition requirement	High		MSR-02		X		
SR-03	Cansat egg placed inside shpuld be recovered safely	competition requirement	High		MSR-03			X	X
SR-04	The cansat shall deploy from the launch vehicle payload section and no protrusions	To be easy leaving from rocket	High		MSR-04			X	
SR-05	The descent control system shall not use any flommable or pyrotechnic devices	competition requirement	High						
SR-06	The average descent rate of cansat carrier after deployment of the lander shall be 4m/s	competition requirement	High		DCS-01	X		X	X
SR-07	The average descent rate of cansat lander after deployment of the lander shall be 5,5m/s	competition requirement	High		DCS-02	X		X	X
SR-08	The lander and the carrier need to be seperated at 500 meters altitude and the paracute need to be opened when it reaced	competition requirement	High		DCS-03	X		X	X

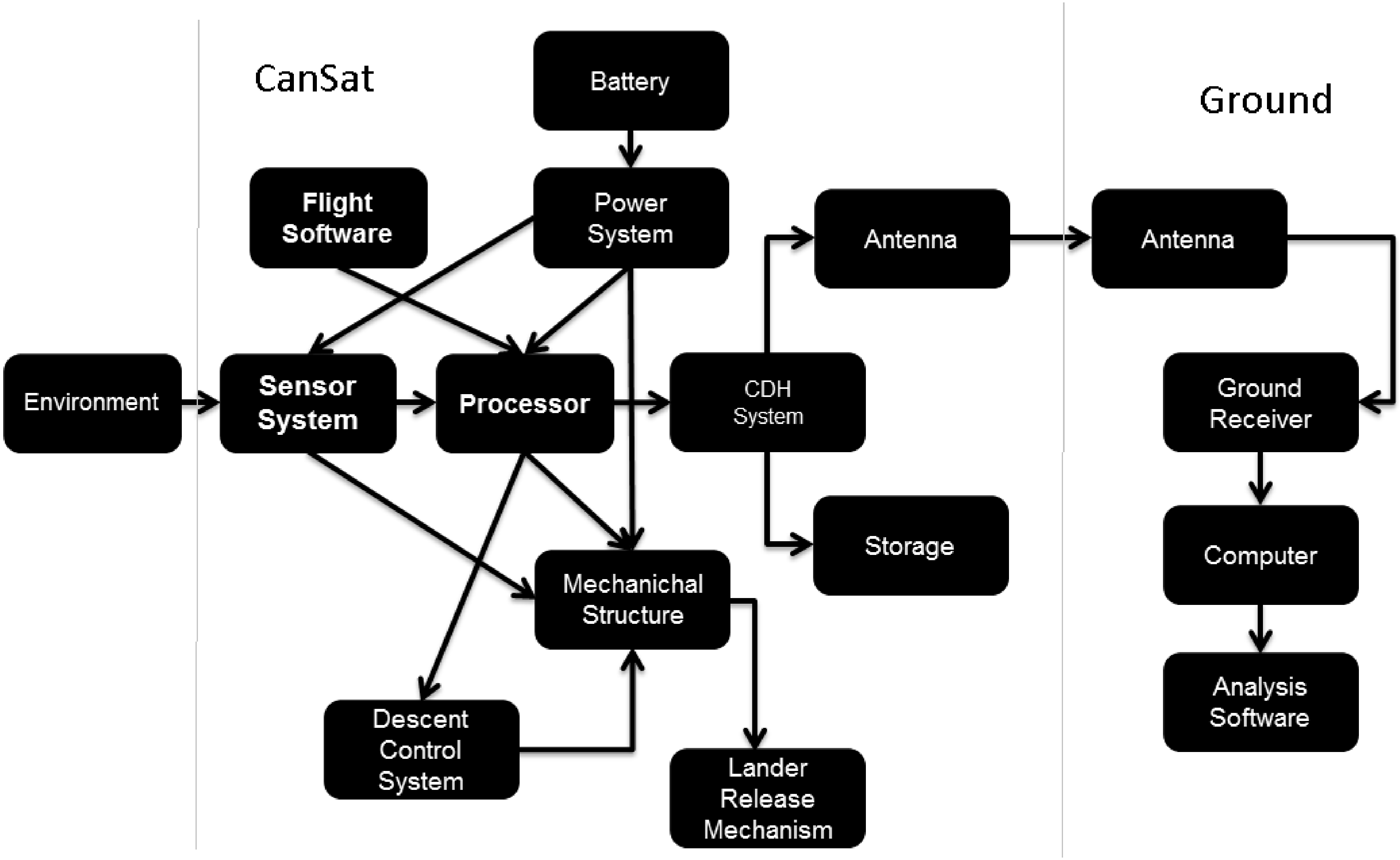
System Requirements

ID	Requirements	Rationale	Priority	Parent	Children	VM			
						A	I	T	D
SR-09	The cansat communication radio shall be the Laird AC4790-200	competition requirement	High		CDH-08 FS-01			X	X
SR-10	The cansat communications shall use the api packet format	competition requirement	High		CDH-08			X	
SR-11	The cansat shall autonomously terminate telemetry transmissions within 5 minutes of landing. This shall be verified via ground control station activities	competition requirement	High		CDH-09			X	X
SR-12	All telemetry shall be displayed in real –time during launch and descent in engineering unit.	competition requirement	High		GCS-05 GCS-06 GCS-07			X	
SR-13	During descent the carrier shall transmit following telemetry data once every two second.	competition requirement	High		CDH-01 FSW-04			X	
SR-14	Lander descent telemetry shall be stored on –board for post processing following retrieval of the lander.	competition requirement	High		CDH-03 FSW-05			X	
SR-15	The cost of cansat flight hardware shall be under 1000$ (ground tools are excluded)	competition requirement	Medium				X		
SR-16	The cansat and asociated operations shall comply with all field safety regulations.	competition requirement	Medium				X		

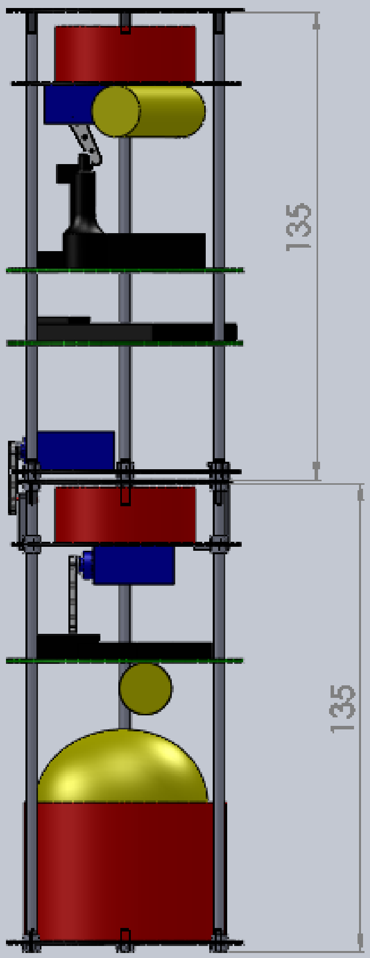
System Concept of Operations



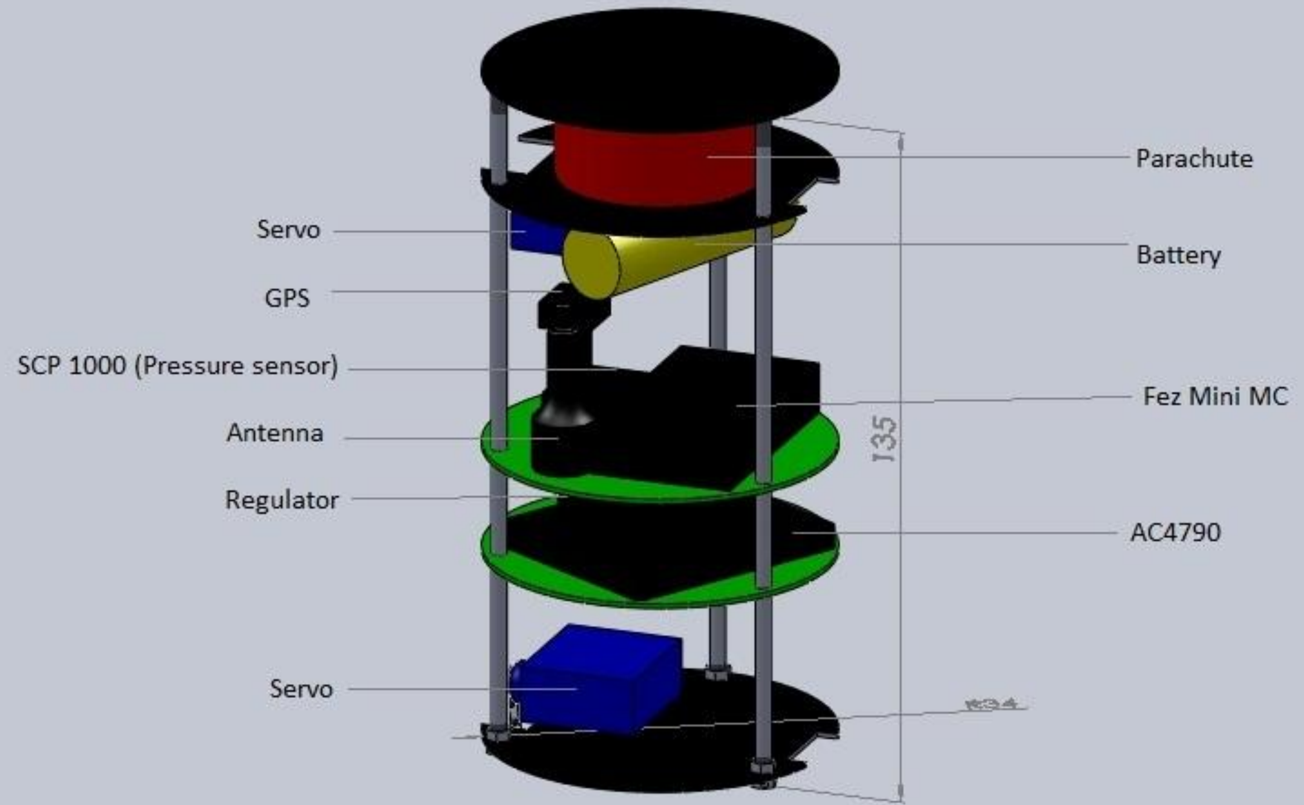
Context Diagram of Cansat



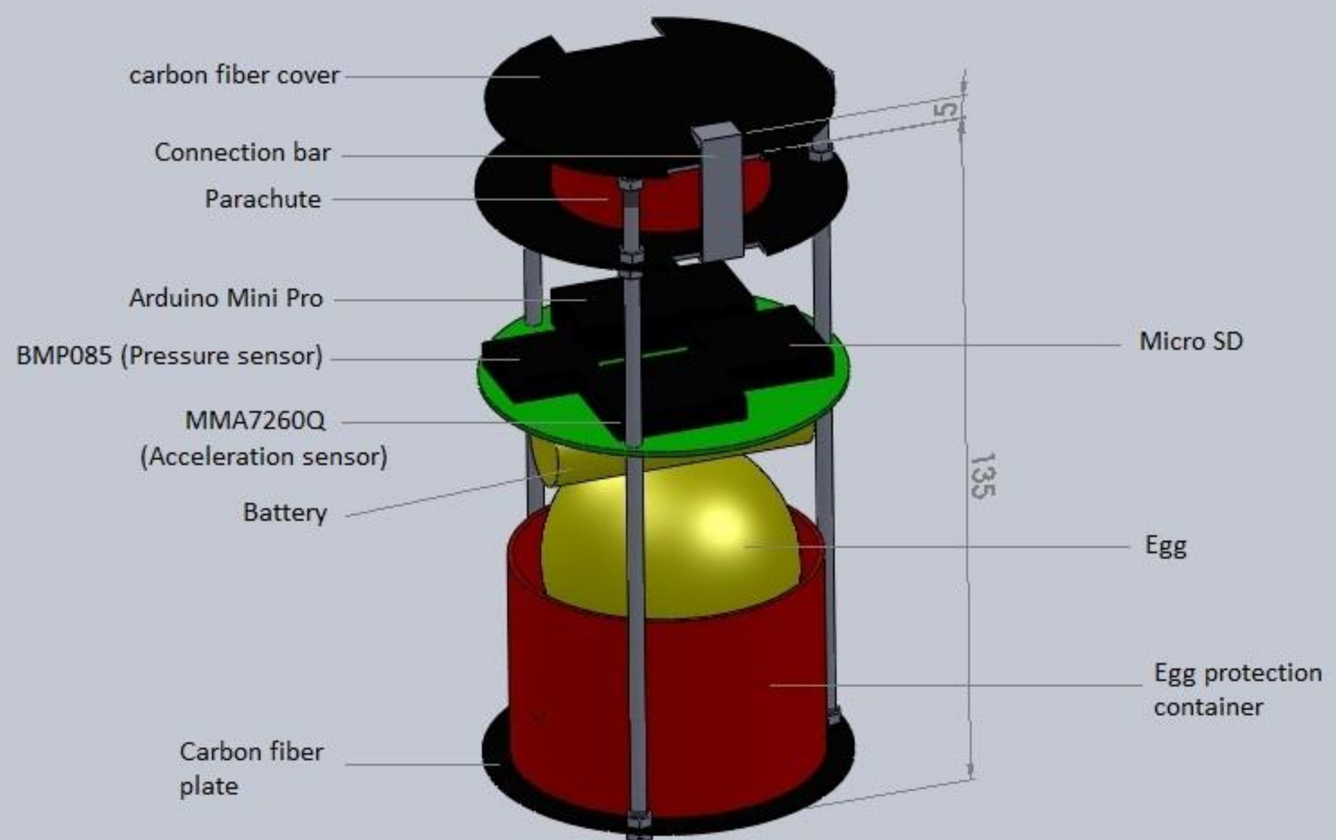
Physical Layout



Carrier

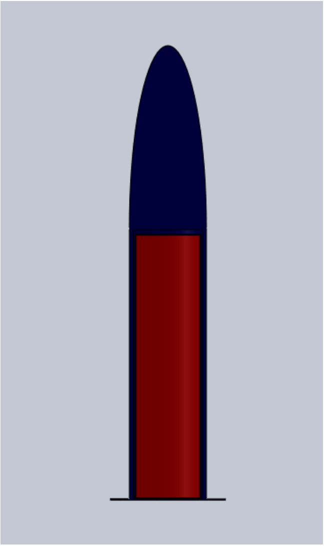

Parachute
Servo
Battery
GPS
SCP 1000 (Pressure sensor)
Fez Mini MC
Antenna
135
Regulator
AC4790
Servo

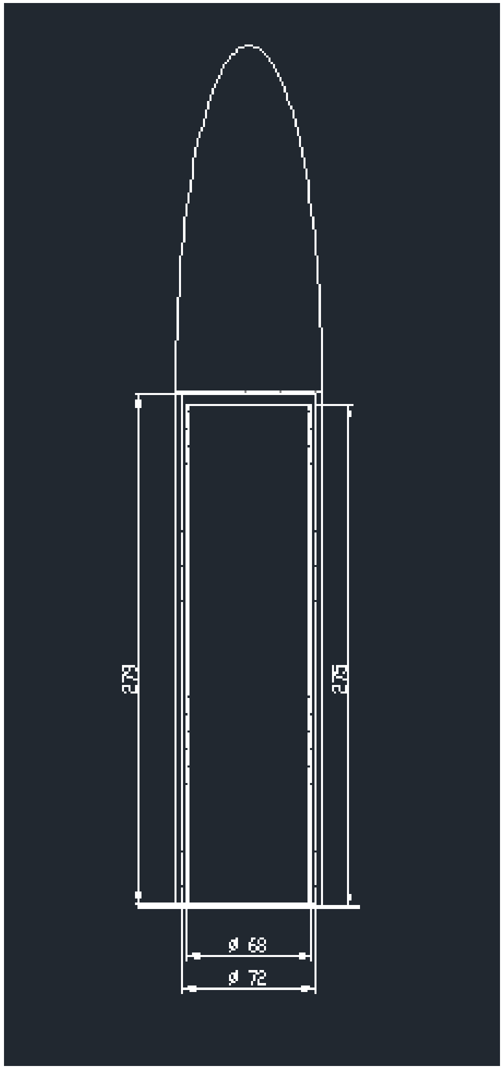
Lander


carbon fiber cover
Connection bar
Parachute
Arduino Mini Pro
BMP085 (Pressure sensor)
MMA7260Q
(Acceleration sensor)
Battery
Carbon fiber
plate
Micro SD
Egg
Egg protection
container
5
135

Launch Vehicle Compatibility

Diameter of cansat is designed 68mm to put the cansat into
72 mm diameter of rocket. Cansat will be put into rocket as a hole body.
Because 4 mm free space in diameter is obtained by this design.
Parachutes will be placed properly between two plates.
Not having an unexpectable event prior to launch day,
a 72mmdiameter cylindrical tube presenting rocket was made
and the cansat was put and taken off the tube. The test was succesful.





Sensor Subsystem Design

Çağrı GÜZAY

Sensor Subsystem Overview

Carrier sensor subsystem

PRE

VTI Technologies-SCP1000

Used for :

Determination of altitude without GPS

Determination of Lander's descent rate

TEM

VTI Technologies-SCP1000

Used for :

Determination of air temperature

Including temperature data to determine the altitude with pressure

GPS

Locosys - 20 Channel LS20126 GPS Receiver

Used for :

Positioning of the Lander

Obtaining other GPS data

Sensor Subsystem Overview

Lander sensor subsystem

PRE

Bosch – BMP085

Used for :

Determination of altitude

Determination of Carrier's descent rate

TEM

Bosch – BMP085

Used for :

Determination of air temperature

Including temperature data to determine the altitude with pressure

ACL

Freescale – MMA7260Q

Used for :

Determination of impact force , occurred by landing of the Carrier

Sensor Subsystem Requirements

ID	Requirement	Rationale	Priority	Parent	Children	VM			
						A	I	T	D
SEN-01	Operating voltage range for all sensor must be between 3.3V and 6V (Carrier & Lander)	Battery and regulator are selectied by these values.	High	EPS-03		X	X	X	X
SEN-02	More than two of Carrier's sensors should not have UART data interface.	Microcontroller has only 3 UART interfaces. One of them is used for AC4790.	High	CDH-10		X		X	
SEN-03	More than two of Carrier's sensors should not have SPI.	Microcontroller has limited SPI.	High	CDH-10		X		X	
SEN-04	One of the Lander sensors must have ADC data interface.	Microcontroller has limited SPI (Lander).	High	CDH-11		X		X	
SEN-05	Other sensors of the Lander should have SPI.	Microcontroller has another interface as SPI (Lander).	High	CDH-11		X		X	
SEN-06	Temperature sensor must have a range of at least 0ºC-50ºC (Carrier).	Air conditins of launch location.	Medium			X		X	

Sensor Subsystem Requirements

ID	Requirement	Rationale	Priority	Parent	Children	VM			
						A	I	T	D
SEN-07	Temperature sensor should have high resoluiton (Carrier)	The higher resolution, the more accurate measurement	Medium			X		X	
SEN-08	Non-GPS altitude measurement sensor (pressure sensor) should have wide range (Carrier)	Pressure measurement is done altitudes between 0m and 1000m.	High			X		X	
SEN-09	Non-GPS altitude measurement sensor (pressure sensor) should have high resoultion (Carrier)	The higher resolution, the more accurate measurement	High			X		X	X
SEN-10	Accuracy and measurement range of the GPS should be at high level (Carrier)	To obtain the closest position	Medium				X	X	X
SEN-11	Pressure sensor of the Lander should have high resolution and range	To obtain more accurate measurements	High			X		X	X
SEN-12	Acceleration sensor of the Lander should have high resolution and range	To obtain more accurate measurements	High			X		X	

Sensor Changes Since PDR

- **All sensors, indicated in PDR, are tested for variable conditions. These tests result with any change of sensors for both Carrier & Lander except for integrated acceleration sensor of GPS module.**
- **We do not need to use of acceleration sensor on Carrier. On the other hand, We decided to measure Lander’s force of impact with the ground. So, imaging sensor on the Carrier was eliminated. This issue will be at the last page of ‘Sensor Subsystem Design’ section.**

Carrier GPS Summary

Characteristics of the selected GPS module are given in following table.

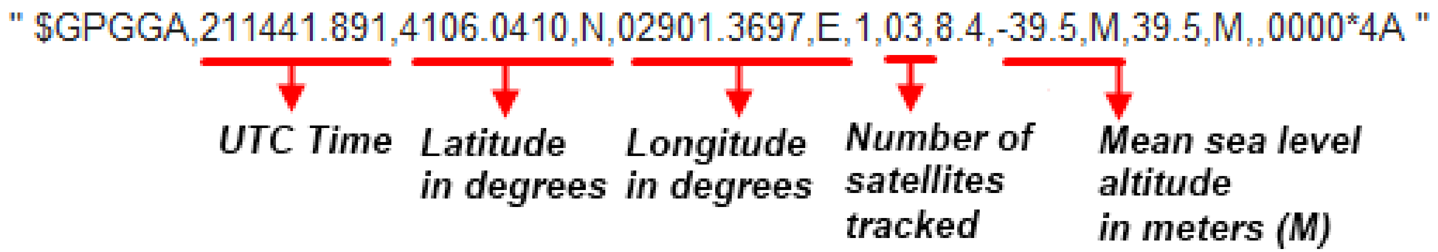
Module	Physical Characteristics		Electrical Characteristics				Cost
	Dimensions	Weight	Nominal operating	Accuracy	Update Rate	Data Interface	
Locosys - LS20126 GPS Receiver	29x12 mm	10 gr	31mA @ 3.3V	<5 meters	1 Hz	UART	$59.95

Main reasons for this selection: (Locosys - LS20126 GPS Receiver)
- Low cost
- Low size

Accuracy is specified as +- 5 meters by the producer of the GPS.Also, this GPS module is tested and It has been found suitable for use.

Carrier GPS Summary

GPS module communicates with microcontroller via serial interface. GPS data is transmitted to microcontroller every second.
Testing data is compared with a handheld GPS data. It is seen that accuracy of the selected GPS module matches with our expectations.
We are getting GPS data by parsing NMEA strings.
We are using $GPGGA string because it holds all the information we need.



<u>Sample data given above is taken at our lab.</u> All requirements of the GPS section will be accomplished by this data format. In addition to accuracy of GPS, Google Earth software gives us our coordinates as **4106.0556,N** and **02901.1834,E**.
Microcontroller stores on board and sends to (via AC4790) ground station computer this data. We will make a software to monitor all flight data on computer.

Carrier GPS Summary

```
$GPGGA,211440.891,4106.0410,N,02901.3697,E,1,03,8.4,-39.5,M,39.5,M,,0000*4B
$GPGGA,211441.891,4106.0410,N,02901.3697,E,1,03,8.4,-39.5,M,39.5,M,,0000*4A
$GPGGA,211442.891,4106.0409,N,02901.3696,E,1,03,8.4,-39.5,M,39.5,M,,0000*40
$GPGGA,211443.891,4106.0409,N,02901.3696,E,1,03,8.4,-39.5,M,39.5,M,,0000*41
$GPGGA,211444.891,4106.0408,N,02901.3696,E,1,03,8.4,-39.5,M,39.5,M,,0000*47
$GPGGA,211445.891,4106.0408,N,02901.3696,E,1,03,8.4,-39.5,M,39.5,M,,0000*46
$GPGGA,211446.891,4106.0407,N,02901.3696,E,1,03,8.4,-39.5,M,39.5,M,,0000*4A
$GPGGA,211447.891,4106.0407,N,02901.3696,E,1,03,8.4,-39.5,M,39.5,M,,0000*4B
$GPGGA,211448.891,4106.0407,N,02901.3696,E,1,03,8.5,-39.5,M,39.5,M,,0000*45
$GPGGA,211449.891,4106.0406,N,02901.3695,E,1,03,8.5,-39.5,M,39.5,M,,0000*46
$GPGGA,211450.891,4106.0406,N,02901.3695,E,1,03,8.5,-39.5,M,39.5,M,,0000*4E
```

This data is generated by Locosys – LS20126 GPS Receiver and stored on SD-card by the microcontroller at our tests.

Carrier GPS Summary

An opinion: Retrieval of Carrier according to GPS data

Google Earth software also uses **kml** file format. This file format looks like an **xml** file. After landing of Carrier, We will generate a **kml** file according to GPS longitude and latitude data. By integrating this file in Google Earth, We will be able to see exact position of our Carrier on the map.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>Carrier is here!</name>
    <description>According to telemetry,
    Carrier is located at these coordinates</description>
    <Point>
      <coordinates>Latitude,Longitude values</coordinates>
    </Point>
  </Placemark>
</kml>
```

A simple kml file (sample)

Carrier Non-GPS Altitude Sensor Summary

Characteristics of the selected non-GPS altitude sensor are given in following table.

Module	Physical Characteristics		Electrical Characteristics				Cost
	Dimensions	Weight	Nominal operating	Resolution	Range	Data Interface	
VTI Tech. – SCP1000	20x19 mm	-	2.4-3.3V	2Pa	30-120 kPa	SPI	$34.95

Main reasons for this selection: (VTI Technologies- SCP1000 Barometric Pressure Sensor)

•Its high accuracy and data resolution. (At 17-bit resolution mode it has the accuracy of 2 Pa which equals to about 15 cm of air column when used to calculate altitude.)

•Integrated temperature sensor. (It fulfills two of our needs and is not so expensive, so a good choice.)

Carrier Non-GPS Altitude Sensor Summary

Data is output by a SPI bus in two parts. First part is the most significant 8 bits of the data and the second part is the least significant 16 bits of data.

Pseudo code for calculating pressure output from sensor:
pressureData = ((msPart << 16) | lsPart)

Pressure data is read in decimal format and is converted to Pascal with the following formula taken from the datasheet:

$$Pres[Pa] = \frac{Pres[dec]}{4} = 0.25 \cdot Pres[dec],$$

Carrier Non-GPS Altitude Sensor Summary

Altitude determination is done by given equation:

$$P = P_0 \exp\left[\frac{-A\,z}{T}\right]$$

P: *Measured pressure* (Pa)

P_0: *Reference pressure* $(Sea\ level, Pa)$

A: *Coefficient* $(avg.\,0.0342)$

z: *Altitude* $(in\ meters)$

T: *Temperature* $(in\ Kelvin)$

By arranging of variables,

$$z = \frac{T}{A}\ln\left[\frac{P_0}{P}\right]$$

We are able to calculate the altitude both on the Cansat or ground station.

Note: It is not mentioned about obtaining of the temperature data. All about air temperature measurement and processing are at next pages.

Carrier Non-GPS Altitude Sensor Summary

18815181,"C","Read init."
18815723,"R",26.40,1007.71
18826056,"C","Read init."
18826598,"R",26.40,1007.68
18836937,"C","Read init."
18837561,"R",26.45,1007.68
18847908,"C","Read init."
18848859,"R",26.40,1007.68
18859206,"C","Read init."
18859753,"R",26.45,1007.71
18870101,"C","Read init."
18870647,"R",26.40,1007.70
18881101,"C","Read init."
18881771,"R",26.40,1007.68
18892107,"C","Read init."
18892649,"R",26.45,1007.69
18902981,"C","Read init."
18903522,"R",26.40,1007.70

This data is generated by SCP1000 Barometric Pressure sensor and stored on SD-card by the microcontroller at our tests.

Data format:
timestamp ,“action type”, data

timestamp: Calculated from the time passed since the microcontroller started.
Action type: “C” stands for comment, “R” stands for reading
Data: Comment when action type is “C”, and readings when action type “R”
Sensor readings are formatted as: Temperature in Celsius, Pressure in hectopascals.

(To measure both air pressure and temperature, We use the same sensor. So, given test data includes test data from air temperature sensor section.)

Carrier Air Temperature Summary

Characteristics of the selected air temperature sensor are given in following table.

Module	Physical Characteristics		Electrical Characteristics				Cost
	Dimensions	Weight	Nominal operating	Resolution	Range	Data Interface	
VTI Tech. – SCP1000	20x19 mm	-	2.4-3.3V	14 bits	-20 to 70 ºC	SPI	$34.95

Main reasons for this selection: (VTI Technologies- SCP1000 Barometric Pressure Sensor)

•Integrated pressure sensor

•With “Carrier non-GPS Altitude Sensor” section, air temperature measurement is necessary for both altitude determination and mission objective . Integrated pressure-temperature sensor will accomplish these objectives.

Carrier Air Temperature Summary

Data is output by a SPI bus, from one 16bit register.
Data resolution is 14 bit, with one bit being the sign bit.
The temperature in Celsius is calculated from data output from sensor according to following formula taken from the datasheet:

$$Temp[°C] = \frac{Temp[dec]}{20} = 0.05 \cdot Temp[dec]$$

Note: To use temperature data in altitude equation, It is converted to Kelvin unit by adding 273.

Lander Pressure Sensor Summary

Characteristics of the selected presure sensor are given in following table.

Module	Physical Characteristics		Electrical Characteristics				Cost
	Dimensions	Weight	Nominal operating	Accuracy	Range	Data Interface	
Bosch – BMP085	16x16 mm	-	1.8-3.6V	3Pa	30-110 kPa	I^2C	$19.95

Main reasons for this selection: (Bosch – BMP085 Pressure Sensor)

•Low cost
•High accuracy

(This sensor has an integrated temperature sensor. We planned to use air temperature data to determine the descent rate of Lander.)

Lander Pressure Sensor Summary

Data is output by a I^2C bus. Retrieving of the data from this sensor is a little complicated. Full algorithm of this process is at product datasheet *(http://www.bosch-sensortec.com/content/language1/downloads/BST-BMP085-DS000-05.pdf) .*
All configurations are done by the datasheet.
Sensor gives pressure data in Pa and temperature data in 0.1 Celsius.

1036 , 99865 , 260 , 12689
2004, 99862 , 260 , 12715
3023, 99867 , 260 , 12671
4030, 99863 , 260 , 12706
5021, 99854 , 260 , 12785
6034, 99860 , 260 , 12732
7009, 99866 , 260 , 12680
8015, 99867 , 260 , 12671
9003, 99863 , 260 , 12706
10020, 99859 , 260 , 12741
11012, 99852 , 260 , 12802
12035, 99850 , 260 , 12820

This data is generated by BMP085 Barometric Pressure sensor and stored on micro SD-card by the microcontroller at our tests.

First column: millisecond
Second column: Pressure (Pa)
Third column: Temperature (01. ºC)
Temperature: To calculate altitude
Fourth column: Altitude (cm)
Altitude: To calculate descent rate

(Altitude calculation is given at Carrier Non-GPS Altitude Sensor Summary section.)

However, sensor stands still. Altitude and pressure do not steady. To overcome this issue, We planned to use a median filter at software of Lander’s microcontroller.

Lander Impact Force Sensor Summary

We decided to work on Lander impact force determination.
Characteristics of the selected sensor are given in following table.

Module	Physical Characteristics		Electrical Characteristics				Cost
	Dimensions	Weight	Nominal operating	Sensitivity	Range	Data Interface	
Freescale – MMA7260Q	25x25 mm	-	3.3V	800mV/g	±6g	ADC	$19.95

Main reasons for this selection: (Freescale – MMA7260Q Accelerometer)

- Low cost
- Output is analog.

This sensor has four different range selection (1.5g/2g/4g/6g). The more range, the less sensitivity. Also, if range increases, output decreases. Considering output signal of accelerometer, we select 4g or 6g range.

Lander Impact Force Sensor Summary

This sensor is tested and test results about accuracy and data processing are given below.

•Whatever range is selected, sensor gives analog data scaled from 0 to 1023 (this means as a voltage level from 0V to 3.3V).
•If the sensor stands still, it gives about 450 as an analog value.
•Sensor accuracy is near to our expectations.

Sensor gives analog signal to microcontroller. This signal is sampled and stored on micro SD-card.

Strategy of measurement Lander impact force:
•Sensor data should be collected at a rate of at least 100Hz and stored on-board. (We are collecting this data at a rate of about 200Hz and storing in micro-SD card).

•When Lander touches to the ground, sensor gives a pulse (higher valued analog data).

•After retrieval of the Lander, we will graph this data and find peak value via a computer. We convert this value to voltage level (0V-3.3V). This voltage level can be translated to g value by selected g-range resolution.

Lander Impact Force Sensor Summary

G- Range	Resolution
1.5g	800mV/g
2g	600mV/g
4g	300mV/g
6g	200mV/g

Strategy of measurement Lander impact force: (continued)

After finding of touching acceleration of Lander, we use the eqation given below.

$$Momentum = mass\ x\ velocity : P = mv$$

$$Impact\ force = Change\ of\ momentum\ by\ time: I = \frac{dP}{dt}$$

$$*\ mass\ of\ Lander\ does\ not\ change$$

$$I = m\frac{dv}{dt} = m\,a : mass\ x\ acceleration$$

Finally,
Impact force can be calculated easily. Since, mass of Lander and touching to the ground acceleration are known.

Lander Impact Force Sensor Summary

9297 , 499
9304 , 499
9308 , 498
9314 , 500
9320 , 499
9326 , 499
9332 , 499
9336 , 499
9342 , 499
9349 , 499
9355 , 500
9361 , 499
9365 , 499
9371 , 498
9377 , 499
9383 , 499
9388 , 499
9394 , 499
9400 , 499

This data is generated by MMA7260Q acceleration sensor and stored on micro SD-card by the microcontroller at our tests.

First column: millisecond
Second column: Acceleration value (in millivolts)

According to test data:
Sensor stands still. At this condition, It gives between 450 and 500 millivolts.

Instant change of these values means Lander touch to the ground.

Descent Control Design

AYKUT CETIN

Descent Control Overview

- The descent system includes a paper parachute to keep the descent velocity as a constant.
- The system includes 3 micro servos(6.6grams).
- One servo is to separate the lander and carrier (number 1).
- Two are the open the lander's (number 2) and the carrier's (number 3) parachute.
- When the cansat separating from the rocket the carrier cansat parachute will be open (number 3 servo). When the cansat falls and reaches 500meters altitude, number 1 servo will separate the lander and the carrier then number 2 servo will open the lander's parachute. So that the controlled descent starts with them and continues to the ground.
- We chose the parachute:
- *low weight (paper parachute)
- *no need to motor control system only control three servos
- *Minumum energy (to open the servo 50mAh energy needed and motor control system needs about 1500mAh energy. So that low battery weight.)
- Paper parachute system weight is about 50 grams. Motor system weight is about 250 grams.

Descent Control Changes Since PDR

- We added the parachutes calculations also we stated material of parachutes.
- We added our new parachutes pictures.

Descent Control Requirements

ID	REQUIREMENT	Rationale	Priority	Parent	A	I	T	D
DCS-01	PARACHUTE(The descent should be at a constant velocity which is 4 m/s for carrier	To make a system which has a constant descent rate	High	SR-06	×		×	×
DCS-02	PARACHUTE(The descent should be at a constant velocity which is and 5.5 m/s for lander.	To make a system which has a constant descent rate and to protect the egg from the crash	High	SR-07	×		×	×
DCS-03	MICROSERVOS (the lander and carrier need to be separated at 500m altitude and the parachutes need to be opened when reaced to 500 meters	To create a basic and a useful system using microservos and magnet.	High	SR-08	×		×	×

Volume calculations: for each servo 5,56 cm^3
for each parachute 12,5 cm^3 closed and 16746 cm^3 open condition

Mass: 30 gr for each parachute
6,3 gr for each servo

Carrier total mass of requirement:36,3 gr

Lander total mass of requirements: 42,4 gr

4 m/s descent rate for carrier

5.5 m/s descent rate for lander

Descent Control Hardware Summary

- **We don’t use any actuator.**
- **Our passive components**

Lander parachute size: r=>> r = 36cms
Lander parachute volume= >>12214.512cm^3
Carrier parachute size: r=>> 42cms
Carrier parachute volume=>>19296.193cm^3

Easy to use, no need to motor control, low power consumption.

- **Descent rate is calculated from change of altitude. Altitude is obtained from pressure sensor. Descent rate will be calculated on ground station computer.**
- **Telemetry data includes millisecond and altitude (in meters). On ground station computer, We make descent rate calculated with Matlab.**

- **We do not use any actuator.**

Descent Rate Estimates

$$r = \sqrt{2F_{drag} / \pi C_d \rho v^2}$$

Where,

π = 3.14

ρ = 1.146 kg/m^3 (density of air at 35 °C)

C_d =1.5 (drag coefficient of the chute for a hemisphere chute)

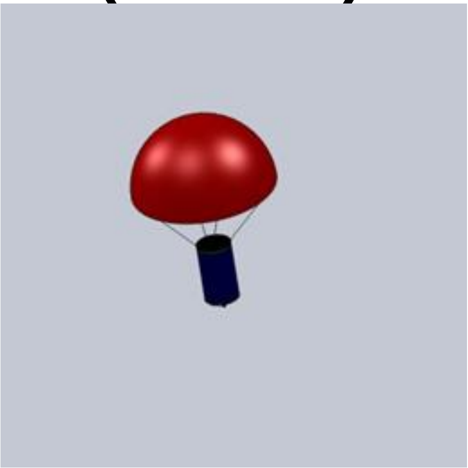
v = terminal velocity achieved (from mission requirements)

F_{drag} = mg (drag force exerted by the air)

Descent Rate Estimates

For the given range of descent velocity for carrier 4.0m/s(+/-1m/s) and for lander 5.5m/s(+/-1m/s)

Radius of parachutes:



- **FOR lander average velocity 5.5m/s >> r = 36cms**

  carrier velocity average 4.0m/s >> r = 42cms

Mechanical Subsystem Design

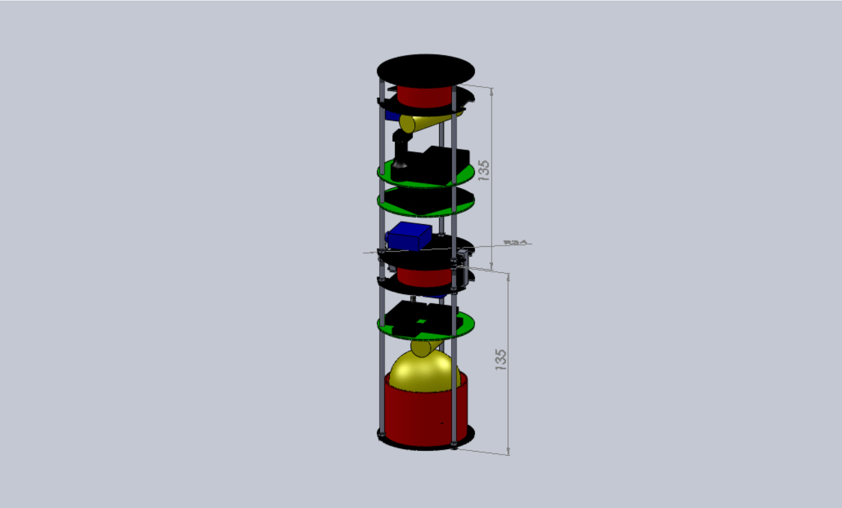
HASAN ERDEM HARMAN

Mechanical Subsystem Overview

Main objective of mechanical design is to place the cansat elements in a robust,protective and low weigth strucutre that carry out all the functions in the competition.

The mechanical system consists,

1. **Structure of the cansat**
2. **Carrier-lander interface**
3. **Egg protection system**
4. **Placing Descent control elements and helping them to work**
5. **Placing electronics components into the cansat**



overview of the cansat

Mechanical Subsystem Overview

The major structure elements of mechanical system can be listed as,

<u>Lander</u>

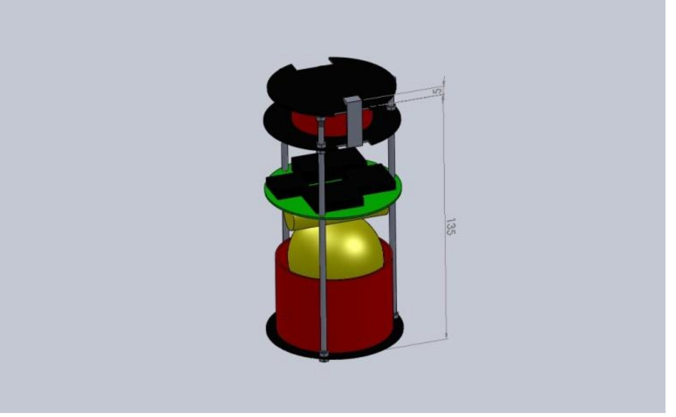
The structure
Silver steel (building the body)
Carbon fiber plates (building the body)
Epoxy
Glue and pin bolt, silicon tube and special dough
Egg protection system.
Servo

<u>Carrier</u>

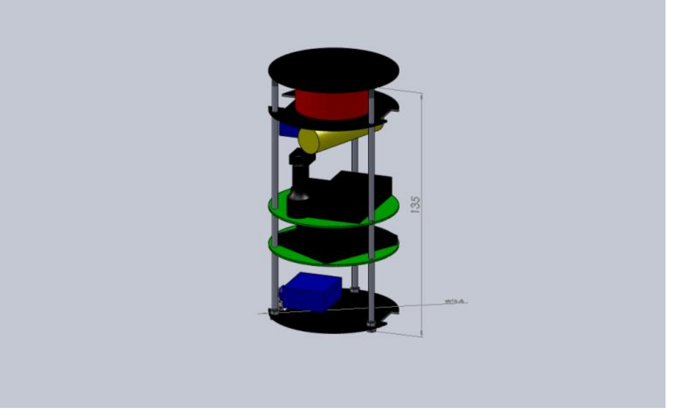
The structure
Silver steel (building the body)
Carbon fiber plates (building the body
Epoxy
Glue and pin bolt, silicon tube
Servo

The cards which includes electronic elements are connected to bars. Silicon tubes are used to hold them stable in their location. The tube has a friction between the bars . It keeps the cards hold their location.

The lander and carrier are connected by bars which are element of serating system. It will be explained in the next slides.



lander layout

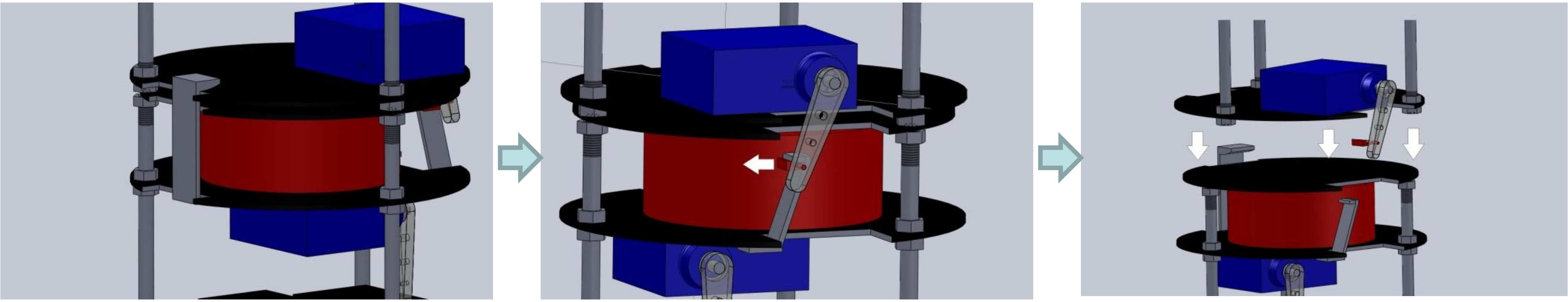


carrier layout

Mechanical Subsystem Changes Since PDR

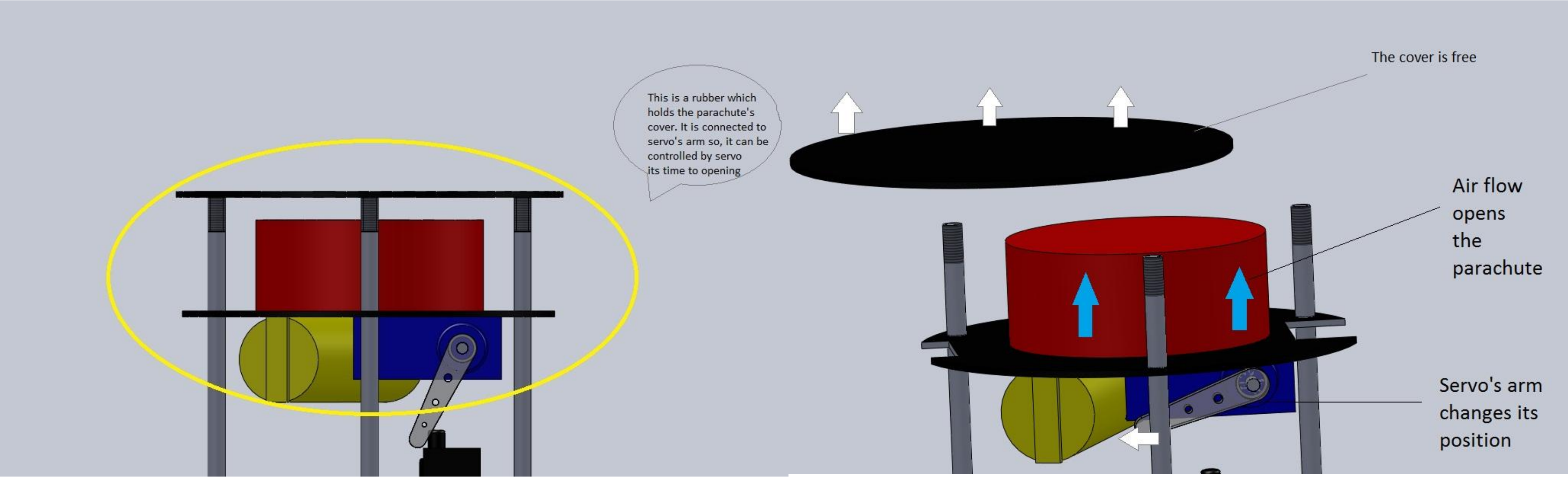
1.The value of cansat diameter has been changed to provide cansat leaving the rocket easily. İt is reduced to 68 mm diameter. The reason why we have changed the diameter is not to have a problem while putting cansat into the rocket and to provide the cansat leave the rocket easily.

2. Carrier lander interface is modified.
For the interface of lander and carrier, a simple design has been discovered. At this design, magnets aren't used for connection. Instead of this, two bars have been added to design. One is stable and connects the carbon fibers of both lander and carrier. The other one is connected to servo. On the servo's arm edge, there is a small piece of metal shown in red.
The item provides us restricted form connection. So magnets aren't used for the connection. When it's time to seperating, servo changes its placements, and then the connection of restricted form is broken.
It is a simple way than the old one. The figures drawn in solidworks are below.



Mechanical Subsystem Changes Since PDR

3. The placement of servos for opening parachute has been changed. The parachutes are placed between two carbon fiber plates at this design. The plates are fixed by a rubber which is controlled by servos. The rubber covers the plates and hold them fixed.When its time to opening parachutes, the servo's arm releases the rubbers, so the connection is broken. The upside plate is free now and it will take off. After that, air flow will open the parachute. This is explained on figures below.



Note: Yellow line is a rubber. It is drawn like a line because it is hard to be drawn as real.

Mechanical System Requirements

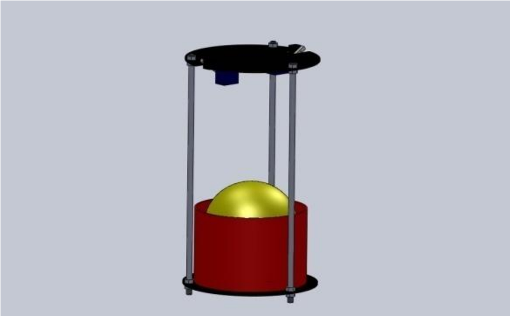
ID	REQUIREMENT	RATIONALE	PRIORITY	PARENTS	A	I	T	D
SR-01	Total mass of the cansat will not be more than 500 gr excluding egg.	The rules says total mass should be under 500 gr.	High	SR-01		×	×	
SR-02	Cansat will fit in a cylindrical diameter of 72 mm and 279 mm in length	The rules says the lenght of cansat should be 279mm	High	SR-02		×		
SR-03	Cansat egg placed inside should be recovered safely	There will be no damage on the egg when the lander landed on the ground	High	SR-03			×	×
SR-04	There will be no protrusions until cansat deployment from rocket payload	The rules says there will be no protrusions		SR-04				
SR-05	Placement of gps antenna, transceiver antenna and camera module	The antenna and camera should be placed on a suitable place on the carrier	Medium	SR-05			×	×

Egg Protection Overview

We think using a speacial organic dough for egg protection that it will damp the force while the lander crashing onto the earth. The dough damps the force using pnomatical systems on air bubbles. So that the egg is exposed low momentum forces.

The egg is generally breaks because of a very big force applies on a small area of the egg's surface and it causes sudden temperature changes.

The prepared dough provides explore the stress on the half of the egg's surface and explore the heat.



Also egg will be protected using string not to take out of its place.

TRADE AND SELECTION

<u>The reasons using dough instead of spring</u>

Force cannot be explored on the half of the surface of the egg when using spring

The dough is cheaper from spring and if spring is chosen, the spring would have to assembled and more application for the assembly would needed.

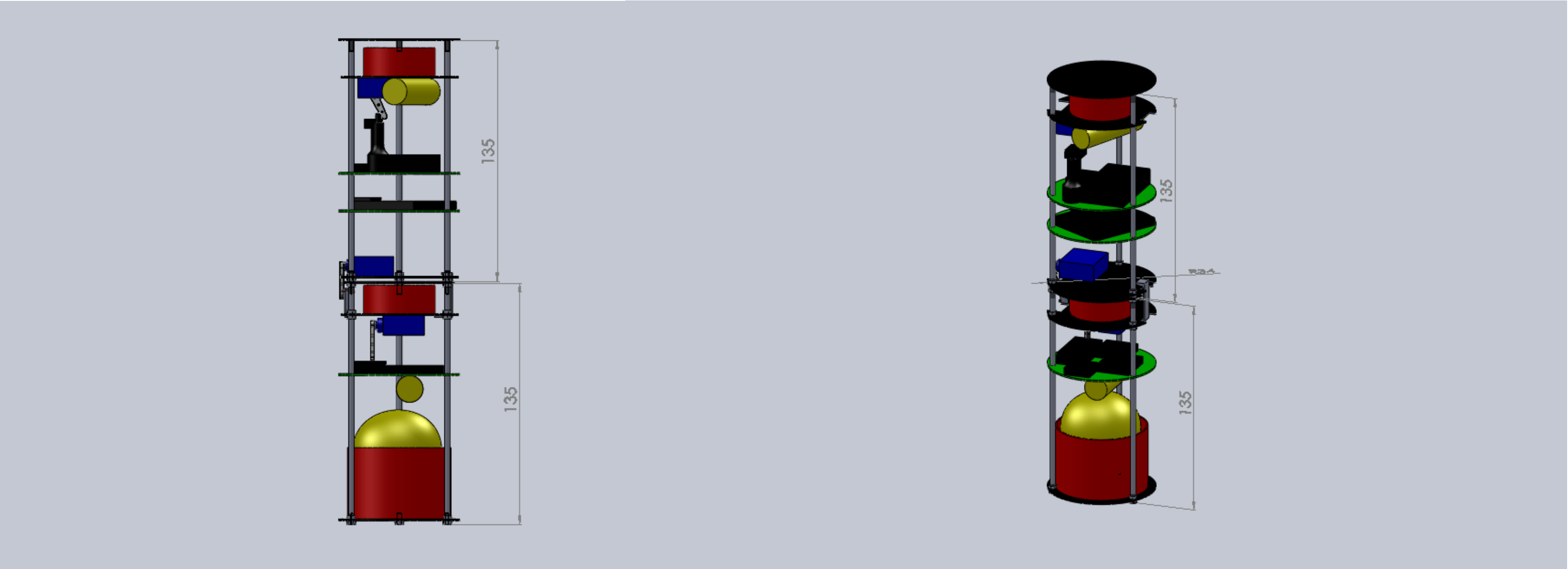
<u>The reason using dough instead of sponge</u>

The coefficient of damping force of sponge is not enough to protect the egg from breaking.

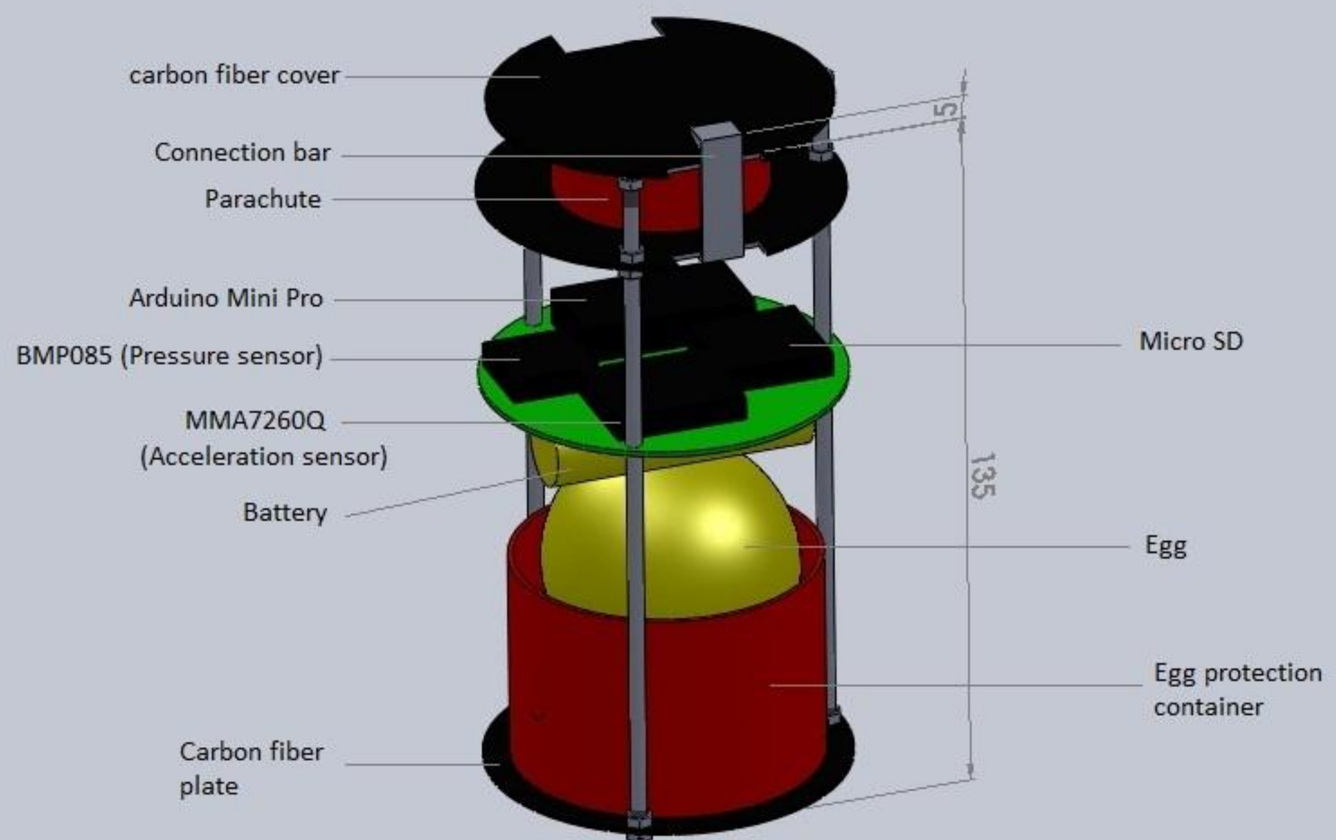
If sponge is chosen we would use more volume to damp equivalent force.

Mechanical Layout of Components Trade & Selection

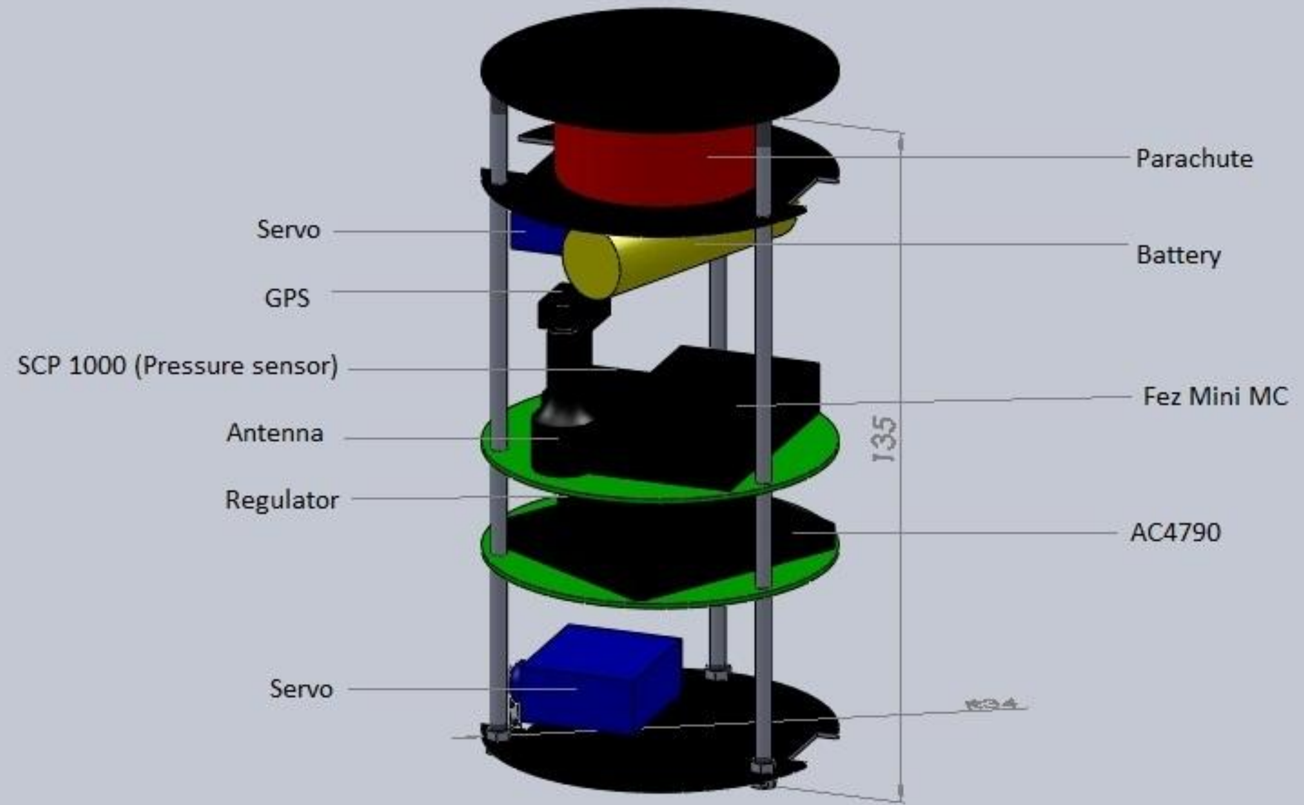
- **The carrier is on the top and the lander is on the down side. It is beacause the lander is carrying the egg and it is heavier than carrier. In addition this, egg protection is provided by placing egg protection system on the bottom of lander and fixing it there. All servos placing and parachutes placing are adjusted for this position of layout.**
- **The pbc cards on which electronics attached, are placed in the center of both carrier and lander. The elecronics do not move or change their place, so they are fixed to steel bars with silicon tubes.**
- **Servos are placed on bottom of carbon fiber plates not to attach to parachutes. So there will be no problem when opening the parachutes.**
- **The batteries are placed on the bottom of the carbon fiber plates. For fixing them there, a flat surface like plates is required.**



Lander layout of components



Carrier layout of components



Material selection

<u>Silver steel:</u>

Chromium-vanadium alloyed cold work tool steel with high wear resistance, high edge holding properties, good machinability and toughness properties,

1. Yield strenght is high

2. Cheap,easy to find

3. Easy to machinability

4. Low density Density : at 20 °C 7,80 kg/dm3

<u>Carbon fiber</u>

1. High strength

2. High flexibility

3. Hard to machinability for carbon fiber but

4. Very low weight

5. We already have in our laboratory

<u>Glue and pin bolt</u>

For assembling the parts to the strucure

<u>Silicon tube</u>

To fix the PCB to the body easily

<u>Servo</u>

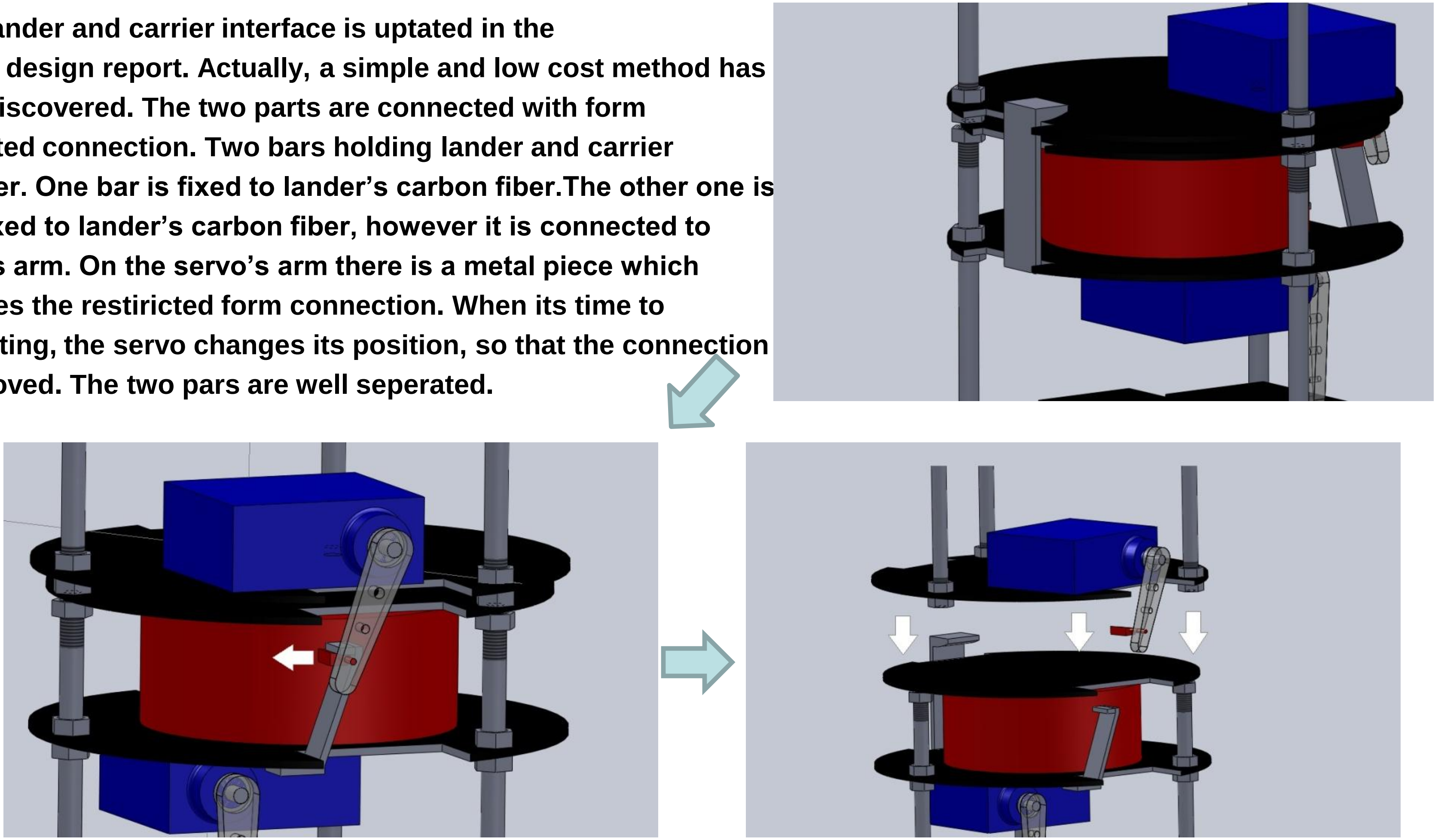
Servo will be used for carrier-lander interface. İt will provide the seperating.

Servo will also be used for opening the parachutes.

Servo provides controlling the seperating and opening parachutes.

Carrier-Lander Interface

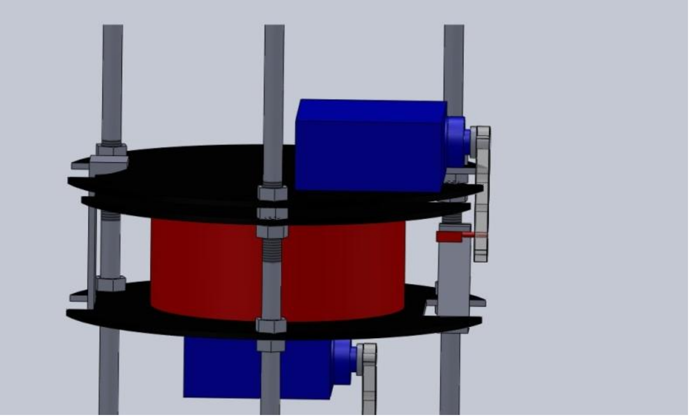
The lander and carrier interface is uptated in the critical design report. Actually, a simple and low cost method has been discovered. The two parts are connected with form restricted connection. Two bars holding lander and carrier together. One bar is fixed to lander's carbon fiber.The other one is also fixed to lander's carbon fiber, however it is connected to servo's arm. On the servo's arm there is a metal piece which provides the restiricted form connection. When its time to seperating, the servo changes its position, so that the connection is removed. The two pars are well seperated.



Carrier-Lander Interface

As shown in the figure on the previous slide, the descent control element, parachute, is placed between two carbon fiber plates. Not creating any attachment of seperating system and parachute, servos are placed under the carbon fiber plates and parachutes are placed on the center of plates.

First ,the seperating will occur, then opening parachute will occur. In this case, there will be no problem for working system.



descent control elements placement

Mass Budget

ELEMENTS	MASS	SOURCE
Carbon-fiber plate	6,7gr	Measured
M3 bold	0,3 gr	measured
Silver-Steel Bar	8,1 gr	Measured
Silicon tube	6,3 gr	Estimated
Servo	40 gr	Product catalog
Battery	50 gr	Product catalog
Dough	11,8 gr (changeable)	Estimated
Egg	60 gr	Estimated
Egg container	5 gr	Estimated
Parachute	30 gr	Product catalog
Holding bars	20 gr	Measured
Total mass of prototype which has just structural elements such as string, carbon fiber plates and bolts	41,4 gr	Sum
Estimated total mass of lander	200 gr	Sum
Estimated total mass of carrier	150gr	Sum
Allocated mass for egg payload	80gr	Estimated

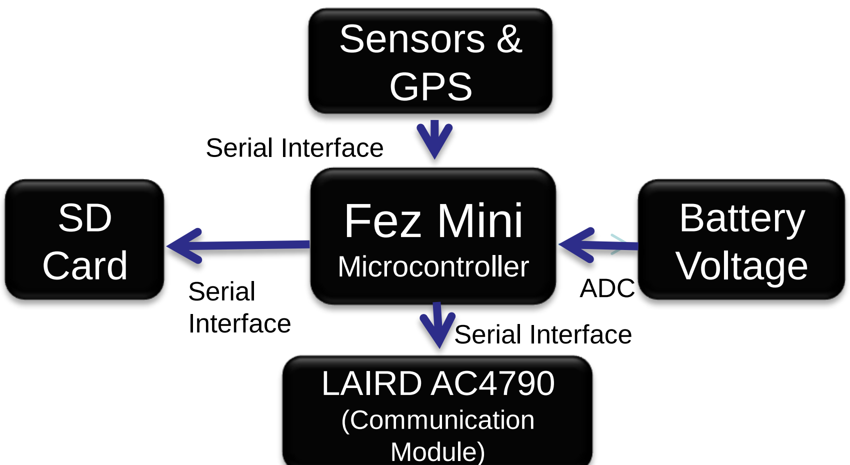
Communication and Data Handling Subsystem Design

İsmail ULUTURK

CDH Overview

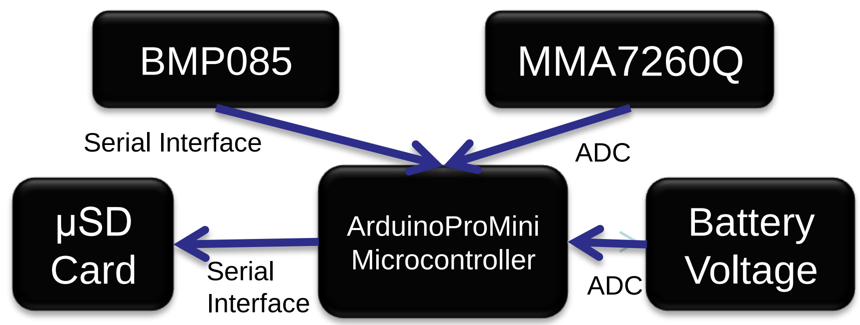
Carrier:

- Fez Mini is the microcontroller board that will handle all communication and data.
- SD Card is used for storing a detailed flight log to be able to see where system fails on tests and providing backup for telemetry in case of a communication failure.
- Laird AC4790 is the radio module that will transmit and recieve messages to/from ground station.
- Details of battery voltage measurement system is discussed at EPS section.
- SCP1000 is used as both temperature and altitude sensor. It is interfaced via SPI by Fez Mini.
- Lycosys LS20126 is used as GPS, it is interfaced via Serial UART interface by Fez Mini.



Lander:

- Arduino Mini is the microcontroller board that will handle all communication and data.
- Micro-SD Card is used for storing a detialed flight log to be able to see where system fails on tests and also for storing the requested data on the board for later retrival.
- Details of battery voltage measurement system is discussed at EPS section.
- Bosch BMP085 is used as both temperature and pressure sensor. It is interfaced via a two wire interface (I^2C) by Arduino Mini.
- MMA7260Q will be used to measure the impact force on the ground. It outputs data in analog format so it is read by ADC of the Arduino Pro Mini.



Ground Station:

- Laird CL4790-USB will be used as reciever on ground station, connected to a laptop PC.
- Custom software written to handle the communications and data will be running on the PC.

CDH Changes Since PDR

- **We changed SD Card on Lander with a Micro-SD Card because it is much more smaller with essentialy same functionality.**
- **We left the camera out, because we choose measuring the impact force of the lander as the bonus objective.**
- **There are slight modifications in the data package format, discussed in further slides.**

CDH Requirements

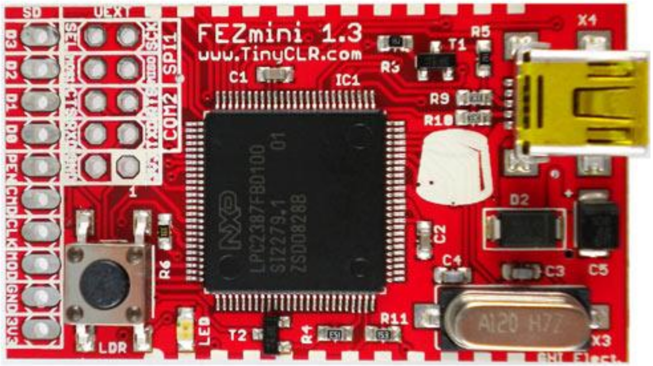
ID	Requirement	Rationale	Priority	Parent(s)	Children	VM			
						A	I	T	D
CDH-01	Transmission of Telemetry Packets at 0.5Hz	Air to ground telemetry transmission is required by the competition	High	SR-13				X	
CDH-02	Handling all sensor data	Sensor data should be converted to a format ready to transmit.	High					X	
CDH-03	Transmission of GPS Data at 0.5Hz	Air to ground GPS data transmission is required by the competition	High	SR-14				X	
CDH-05	Capturing impact accelometer data	Required to calculate the impact force.	Medium			X			X
CDH-06	Data Backup	Data backup to be used in case of a communication failure	Low			X		X	
CDH-07	Flight Log	Detailed flight log to be stored on board for later retrival.	Low			X		X	

CDH Requirements

ID	Requirement	Rationale	Priority	Parent(s)	Children	VM			
						A	I	T	D
CDH-08	Communication shall use Laird AC4790-200 radio with API packet format.	Competition Requirement.	High	SR-09 SR-10				X	
CDH-09	Communications shall be teriminated after landing detected	No need to transmit data after landing	High	SR-11					X
CDH-10	Carrier's controller board shall have enough interfaces for sensors.	Controller needs to read data from all sensors.	High			X			
CDH-11	Lander's controller board shall have enough interfaces for sensors.	Controller needs to read data from all sensors.	High			X			

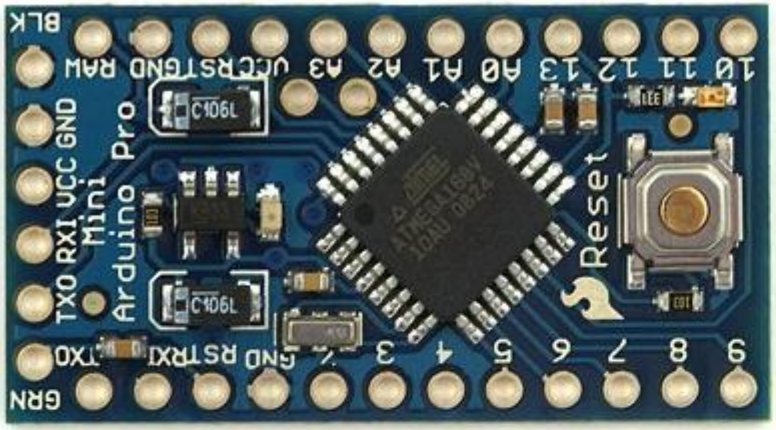
Processor & Memory Selection

- **Fez Mini is our selected processor board for Carrier, It features :**
  - a 72Mhz 32-bit LPC2387 ARM microcontroller from NXP
  - 3xTTL and 1xRS232 serial UART data interfaces.
  - 2x SPI and I^2C data interfaces.
  - 1x CAN bus.
  - 6x Analog Inputs
  - About 148KB of free user FLASH and about 62KB of free user RAM.
  - On board 3.3V regulator
- **Its capabilities far exceeds our requirements, but lesser boards are not really cheaper or smaller and Fez Mini makes prototyping and testing really fast so there is no reason in using a different board.**
  - We have a working prototype that logs all sensor and GPS reading in the SD-Card, at desired data rate.



Processor & Memory Selection

- **Arduino Pro Mini is our selected processor board for Lander, It features :**
  - a 8-bit Atmel ATmega168 microcontroller running at 8Mhz.
  - 1xTTL serial UART data interfaces.
  - 1 x SPI and 1 x I^2C data interfaces.
  - 6x Analog Inputs
  - About 14KB of free user FLASH and about 1KB of free user RAM.
  - On board 3.3V regulator
- **Since we are using a Micro-SD Card for data storage, on board memmory is only needed temporarily, and board is perfectly capable for everything we need.**
  - We have a working prototype that logs readings in the Micro-SD card at desired data rate.



Carrier Antenna Selection

- **Our antenna is Lairdtech 0600-00019:**
  - Electrical Characteristics:
    - Frequency : 902Mhz ~ 928Mhz
    - VSWR : 2
    - Gain : 2dBi
    - Impedance: 50Ohms
    - Price : 14$
  - Mechanical Characteristics:
    - Mass: 22 Grams
    - Length: 182.88mm

Data Package Definitions

- **AC4790:**
  - We are using API packet format:
- **Transmit Package:**

0x81	Payload Data Length (0x01 - 0x80)	Session Count Refresh	Transmit Retries/Broadcast Attempts	Destination MAC (2,1,0)	Payload Data

- Payload Data Length is the length of our data we are sending. It should be the MAC adress of our sender.(Carrier or ground station depending on the phase)
- Destination MAC is the MAC adress of the target receiver.
- The 7 byte overhead is not sent over RF(the part till the Payload Data), but they are used by device and present in buffer.
- **Send Data Complete Package:**

0x82	RSSI	RSSI*	0x00: Failure 0x01: Success

- RSSI is a value that shows the strenth of the received signal, there are methods that calculate distance or even location based on this value, but with only one transmitter and receiver, it is too hard to have a stable result, so we are just ignoring it.
- Failure or Success indicates if the tranmission succeed or not, according to this, same package will be sent again and again until it is successful or a new package is ready to sent.
- **Receive Package:**

0x81	Payload Data Length (0x01 - 0x80)	RSSI	RSSI*	Source MAC (2,1,0)	Payload Data

- Source MAC is the MAC adress of the sender. It should be the MAC adress of our sender.(Carrier or ground station depending on the phase)
- Payload Data is the data we are sending.

Data Package Definitions

- **AC4790:**
- **Serial Interface:**
  - AC4790 communicates with a TTL UART interface, so we can directly connect it to our microcontroller without a voltage interface converter.
  - AC4790 is connected to UART1 of our communication board, communicating at 115,200 baud rate.
  - The data we are sending to AC4790 via UART is stored on a buffer in AC4790 and sent trough RF when certain events occur. Thouse events could be:
    - Interface Timeout
    - RF Packet Size
  - Interface Timeout is set to maximum value for selected baud rate (0x02 = 2ms), and transmission is mainly controlled by RF Packet Size. Packets are always in the same size so unless a failure occurs, data is transmitted by the RF Packet Size event.

Data Package Definitions

- **GPS:**
  - We will use NMEA sentences for interfacing with GPS.
    - NMEA sentences all start with ‘’$’’ and end with <cl><rf> The data that GGA sentence holds is enough for us, so parsing only that sentence will be enough for us.
    - Data is comma delimited in NMEA sentences.
    - An Example NMEA GGA sentence I copied from our SD Card logs at our labs location:
    - $GPGGA,211441.891,4106.0410,N,02901.3697,E,1,03,8.4,-39.5,M,39.5,M,,0000*4A
    - Format of the sentence is present in the datasheet of the GPS and explained in Sensors subsection.
  - **Serial UART Interface at 9600Bps is used to interface with GPS. When only GPS data is needed, there is no need to send any message to GPS, it automatically starts sending out data at 1Hz when powered on.**
  - **We are reading the data sent trough UART, finding the sentence that starts with $GPGGA and parse it to get the data we want, following the format in the datasheet.**

Data Package Definitions

- **SCP1000 Pressure and Temperature Sensor:**
  - SCP1000 is interfaced via SPI bus. It has a simple command interface explained in its datasheet.
  - SPI is a two-way synchronous protocol, so master recieves after it transmits.
  - Data is requested from specific register of the sensor via commands explained in the datasheet, and the value of the requested register is recieved from the sensor.
  - Temperature data and pressure data is read from different registers, and pressure data is read from two registers, one being most significant 8-bit part and one being least significant 16-bit part.
    - Example Reading : (mostSignificantPart << 16) | leastSignificantPart

    - Example Communication:
    - Sending data 0x84 requests temperature data and the microcontroller will recieve the 16bit content of the temperature data register from the sensor.

Data Package Definitions

- **SD Card:**
  - We are using the SD Card in SPI mode, so interfacing is done via SPI bus.
  - FAT-32 file system is handled by the .NET Micro Framework.
  - We are using the routines required to interface with the card from the framework FEZ provides.

  - SD Card SPI interface is too long and complicated to be explained here, it can be found in the Section 7 of the SDCard Physical Layer Specifications from SD Association.

Communication Configuration

- We will use API Packet format in the adressed mode, as specified in the mission guide.
  - We will also use API Send Data Complete feature to ensure that all of our data reaches its destination.
- We will rely on RF Packet Size for transmission since our packets will always be same size. We will set Interface Timeout to a big value we choose by tests, unless a failure occurs, transmission will be made by RF Packet Size.
- We will set System ID and RF Channel Number same with the module in the CanSat and the module at the ground station to be able to initiate sessions and communicate.
- We will write the configuration on the EEPROM of the module, so we won't need to configure the module On-The-Fly everytime we boot it.
- Since the API Send Data Complete function ensures successful transmission, we won't need to take many precautions, a simple checksum will be enough in our application side.

Communication Configuration

Prelaunch	• Establish communication link by sending out start communication packet and recieving acknowledgement packet.
Ascension	• Send data packets to the ground station. Communication is one way in this phase
Descension	• Send data packets to the ground station. Communication is one way in this phase.
Ground	• Stop transmiting data packets after landing detected, send terimination packet to ground station untill recieving acknowledgement packet. Communication is two way in this phase.

- During all phases, all sensor readings and other crucial data about the operation of CanSat and objectives are stored in a log on SD-Cards, at both Carrier and Lander.
- When on ground, carrier sends termination packet, ground station recieves it and answers it and carrier stops communication after recieving the answer. If the answer never comes, that means there is a communication failure or ground-station is out of range, so carrier still stops communication after 3 minutes are past.

Carrier Telemetry Format

Payload Data format:

Start Byte → Seconds since launch → Sensor Data → GPS Data → Checksum

- ’ Sensor Data will include:
  - Air temperature
  - Air pressure
  - Battery voltage
- All data will be sent in ASCII character format, because it is a developed standard.
- Different data will be seperated by comma(‘,’) characters.
- Checksum is included to check the integrity of the recieved package.
- Start Byte is ‘’$’’. We removed the End Byte, because the package size is fixed so the information of the end of package is already known.
- The item with the slowest refresh rate in the list is GPS data, and it has a refresh rate of 1Hz, so we are planning to send data at 0.67Hz(once every 1.5 seconds)

Carrier Telemetry Format

Payload Data format:

Start Byte | Seconds since launch | Sensor Data | GPS Data | Checksum

- Example Frame:
  - ‘’$,45,25,1019,6600,211441.891,4106.0410,N,02901.3697,E,03,39.5,M,163’’
- $ is Start Byte
- 45 is Seconds since launch
- 25 is temperature in Celcius
- 1019 is pressure in hPa
- 6600 is battery voltage in mV
- 211441.891 is UTC time
- 4106.0410 is latitude, N indicates North
- 02901.3697 is longitude, E indicates East
- 03 is the number of sattelites tracked
- 39.5 is Meam Sea Level Altitude, M indicates meters.
- 163 is the checksum, calculated by adding all bytes in the frame modulus 255.

Autonomous Termination of Transmissions

- **We will terminate transmissions after pressure data is stable for 10 seconds.**

- **We will verify that transmissions are terminated by the carrier by sending a pre-defined end-of-transmission package to the ground station**

- **When on ground, carrier sends termination packet, ground station recieves it and answers it and carrier stops communication after recieving the answer.**

- **If the answer never comes, that means there is a communication failure or ground-station is out of range, so carrier still stops communication after 3 minutes are past. Then ground systems should be stopped manualy by hand because without communication, GCS can't know whats happening.**

  (3 minutes are more than enough, but since the limit is 5 minutes, it is not a problem)

Carrier Locator Device Summary

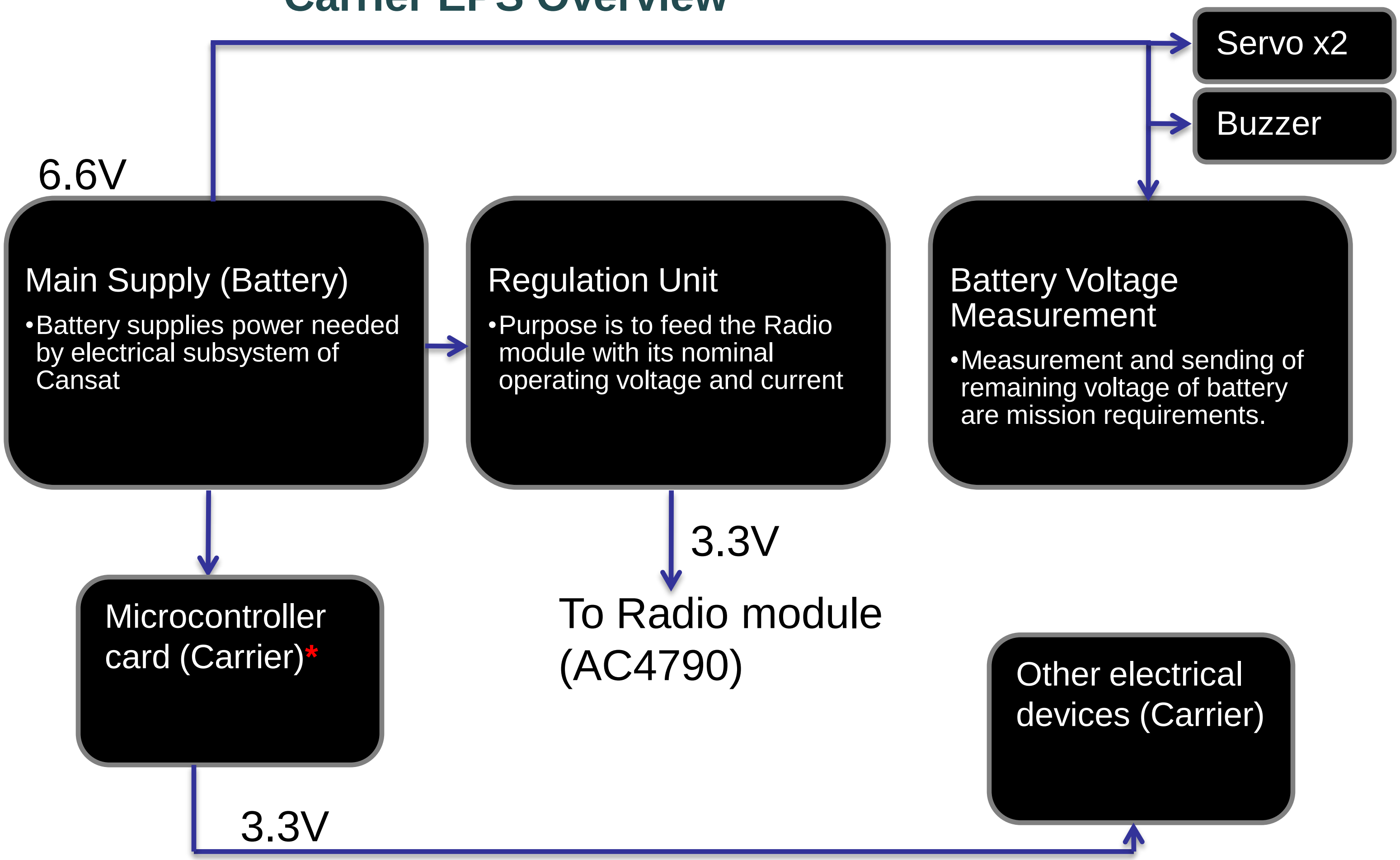
- **Visual indicators might be hard to notice on some conditions, so both carrier and lander will activate a buzzer after landing.**
- **Locator devices will activate after carrier or lander detects that they have landed on the ground.**
- **Landing detection will be made by using the method discussed at the last slide, for terminating of the transmission.**
- **Locater devices will be shut down by a push-button on CanSat.**
- **Specifications of the buzzer we are using:**
  - Operating Voltage: 3-5.5 (We will use 5.1V)
  - Mean Current: < 35mA
  - Sound Output: Typical 95 dBA, Minimum 85 dBA
  - Mean Power Consuption: 5.1V * 35mA = 178.5 mW
- **The buzzer is driven by applying a square wave with %50 duty at 2048Hz. We will do that using one of the hardware PWM outputs of the Fez Mini. Since 35mA is too much for microcontrollers digital pins, there will be a driving transistor.**

Electrical Power Subsystem Design

Çağrı GÜZAY

EPS Overview

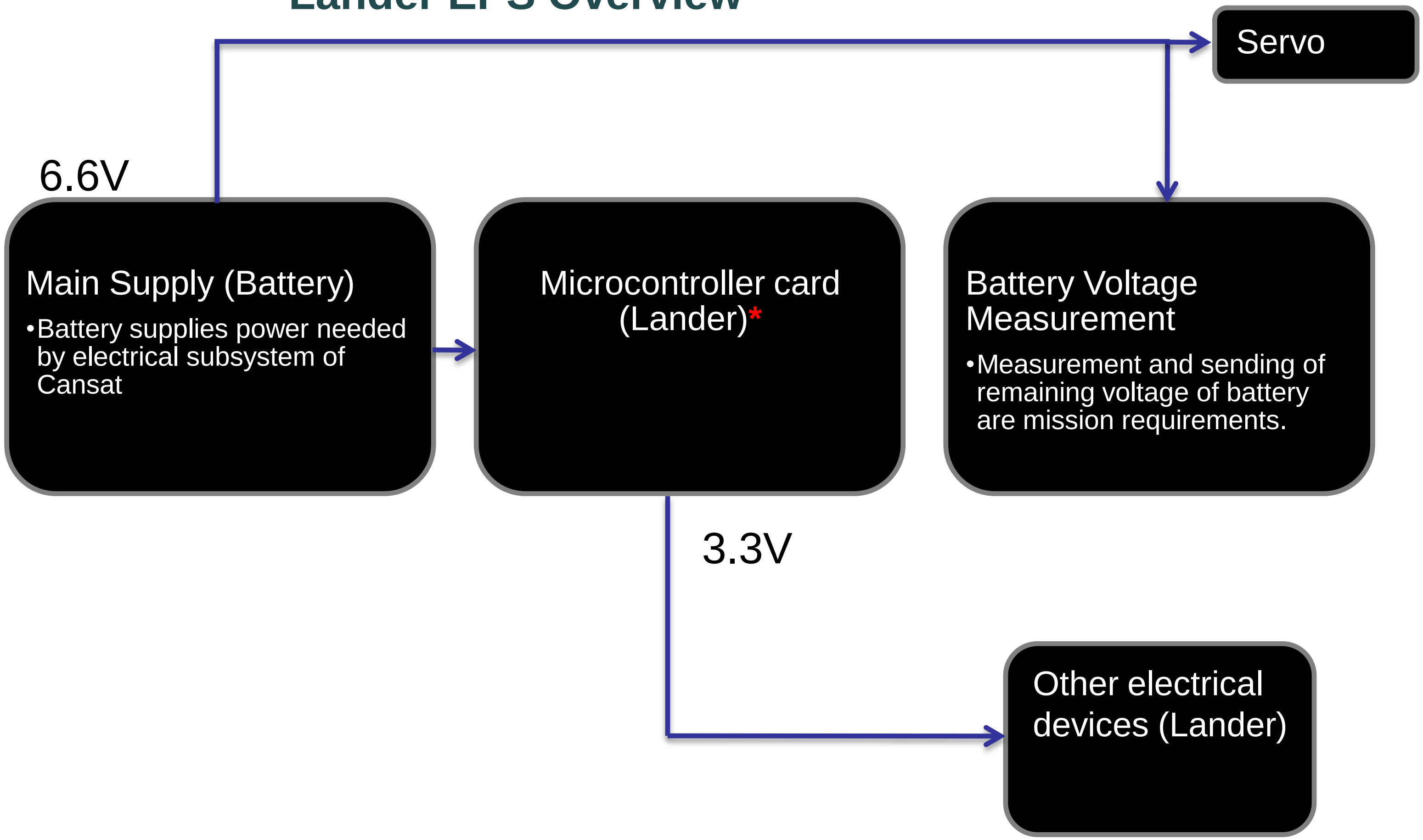
Carrier EPS Overview



Other electrical devices do not draw much current. We planned and tested to use on board regulator of the microcontroller to feed them.

EPS Overview

Lander EPS Overview



Other electrical devices do not draw much current. We planned and tested to use on board regulator of the microcontroller to feed them.

EPS Changes Since PDR

- **For Carrier, only Radio module is fed by regulator unit. Other devices are fed by microcontroller board.**
- **On Carrier, buzzer and servos are fed directly.**
- **For Lander, It is not needed any regulation unit. All devices are fed by microcontroller board.**
- **On Lander, servo is fed directly.**

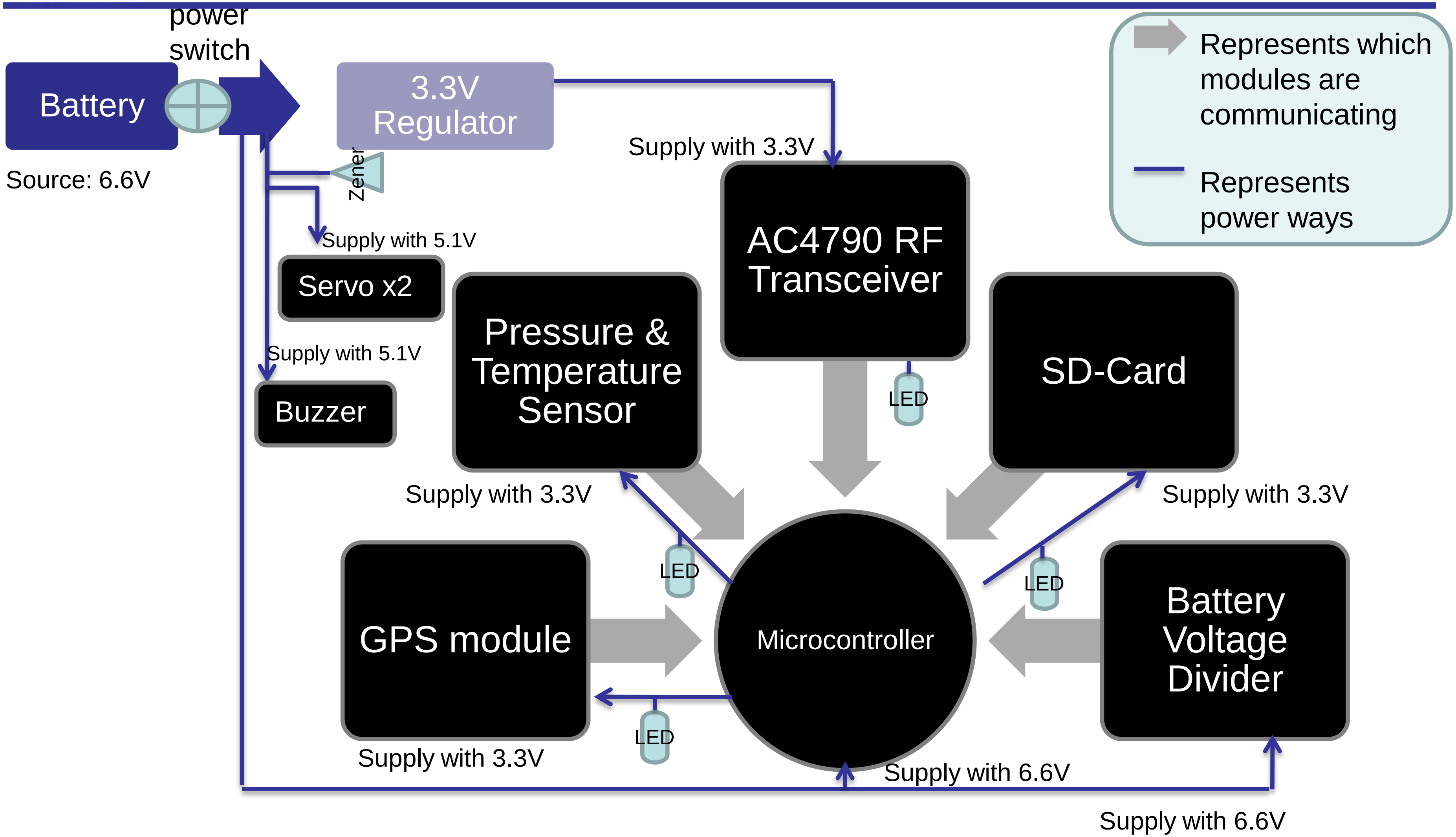
EPS Requirements

ID	Requirement	Rationale	Priority	Parent	Children	VM			
						A	I	T	D
EPS-01	All components should be supplied with an unique battery (Carrier).	There is only one voltage supply, battery.	High		EPS-05	X		X	
EPS-02	All components should be supplied with an unique battery (Lander).	There is only one voltage supply, battery.	High		EPS-04 EPS-05	X		X	
EPS-03	A 3.3V regulator must be used in Carrier.	AC4790 needs very smooth3.3V.	High					X	
EPS-04	The battery of Lander should have between 3.3V and 12V.	It is planned to be used any regulator in Lander. Internal 3.3V regulator of microcontroller card will supply all components.	High	EPS-02			X	X	
EPS-05	The battery voltage must be higher than 3.3V both Carrier and Lander.	Voltage level of the battery decreases, using of it.	High	EPS-01 EPS-02				X	X
EPS-06	Voltage measurement circuit should draw negligible current	Drawing current effects battery	Medium			X		X	

EPS Requirements

ID	Requirement	Rationale	Priority	Parent	Children	VM			
						A	I	T	D
EPS-07	Voltage measurement circuit should draw negligible current	Drawing current effects battery	Medium			X		X	
EPS-08	The battery should supply more than all needed power (Carrier)	Power insufficiency makes system unstable.	High			X	X	X	
EPS-09	The battery should supply more than all needed power (Lander)	Power insufficiency makes system unstable.	High			X	X	X	

Carrier Electrical Block Diagram



Carrier Electrical Block Diagram

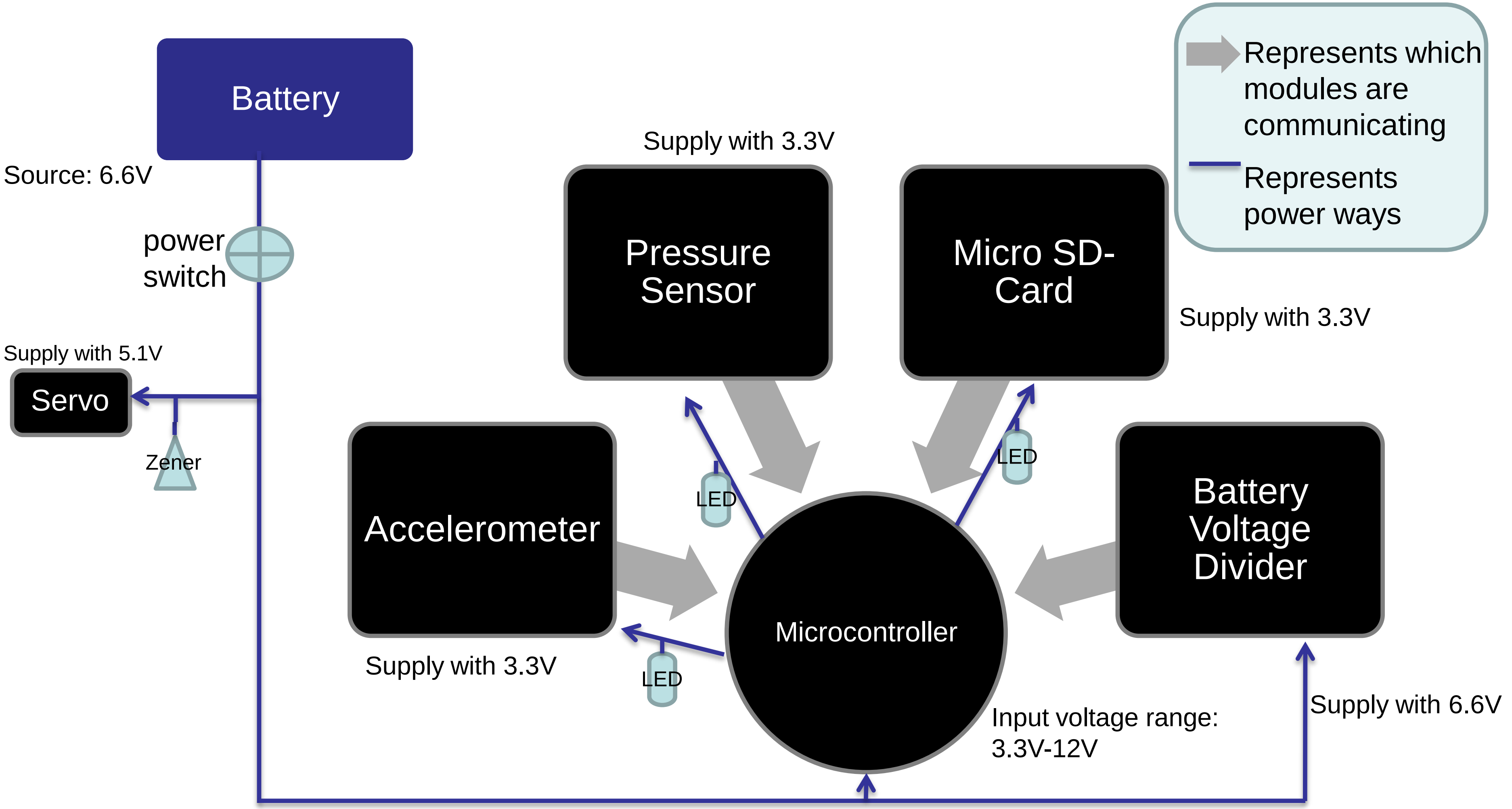
Radio module (AC4790) needs very smooth input voltage. We decided to use a buck converter to convert 6.6V level to 3.3V level. Other electrical devices except from servo are fed from FEZ Mini (microcontroller card). The current drawn by those devices are not harmful for microcontroller card –this information is verified by datasheets and our tests. Microcontroller card has on board 3.3V regulator.

Buzzer and servos are fed by 5.1V voltage level. This is achieved with using a zener diode (5.1V)-resistor combination at their inputs. Buzzer is controlled by a transistor. When Carrier lands, microcontroller activates the transistor in order to run the buzzer.

Power switch is placed before all electrical devices. So It is the main controller.

To control input voltages of GPS module, pressure & temperature sensor and SD-card unit, a LED-resistor combination is connected in parallel at their inputs. This combination should not draw much current. Radio module (AC4790) has indicator LED pin. FEZ mini has own LED on the board showing power condition.

Lander Electrical Block Diagram



Lander Electrical Block Diagram

Accelerometer, pressure sensor and micro SD-card unit are fed from Arduino Pro Mini (microcontroller card). According to microcontroller's datasheet, It can supply the current up 150milliAmps. Also, It has own 3.3V regulator. We mounted accelerometer, pressure sensor and micro-SD card unit on the microcontroller board. While running with full function, average current drawn from battery is about 20milliAmps (peak 25milliAmps). So, there is a lot of margin for microcontroller's regulator.

Servo is fed by 5.1V voltage level. This is achieved with using a zener diode (5.1V)-resistor combination at its input.

Power switch is placed before all electrical devices. So It is the main controller.

To control input voltages of accelerometer, pressure sensor and micro SD-card unit, a LED-resistor combination is connected in parallel at their inputs. This combination should not draw much current. Arduino Pro Mini has own LED on the board showing power condition.

Carrier Power Budget

Module	Power characteristics				Usage ratio
	Voltage	Current	Power	Source	
Microcontroller	3.3V	103mA	340mW	Datasheet/ Measurement	100%
GPS module	3.3V	35.2mA	105.6mW	Datasheet/ Measurement	100%
Pressure & Temperature Sensor	3.3V	25μA	.0825mW	Datasheet	100%
AC4790 RF Transceiver	3.3V	68mA	200mW	Datasheet	100%
Servo motors (2)	5.1V	20mA	204mW	Datasheet	10%
Buzzer	5.1V	30mA	153mW	Datasheet	30%
Indicator LEDs (4)	3.3V	5mA	66mW	Estimation	100%
Battery Voltage Divider	-	-	negligible	Estimation	100%

Total power consumption	~783mWh
Power Available	6600mWh
Margins	5817mWh

Needed voltage	3.3V-6.6V
Battery supply	6.6V

Lander Power Budget

Module	Power characteristics				Usage ratio
	Voltage	Current	Value	Source	
Microcontroller	3.3V-12V	30mA	100mW	Datasheet/ Measurement	100%
Accelerometer	3.3V	500µA	1.32mW	Datasheet	100%
Pressure Sensor	1.8V-3.6V	1mA	3.3mW	Datasheet	100%
Servo motors	5.1V	20mA	102mW	Datasheet	10%
Indicator LEDs (3)	3.3V	5mA	49.5mW	Estimation	100%
Battery Voltage Divider	-	-	negligible	Estimation	100%

Total power consumption	~164.32mWh
Power Available	6600mWh
Margins	6435.68mWh

Needed voltage	3.3V-5.1V
Battery supply	6.6V

Power Source Summary

Selected power source for Carrier and Lander is the same. Its specifications are listed below.

Module	Physical Characteristics		Electrical Characteristics		Cost
	Length & Diameter	Weight	Nominal capacity	Voltage	
A123 (LiFePO4)	40x10mm	39gr	1100mAh	3.3V	$8.06

We choose A123 (LiFePO4) for Carrier and Lander.
Because of,

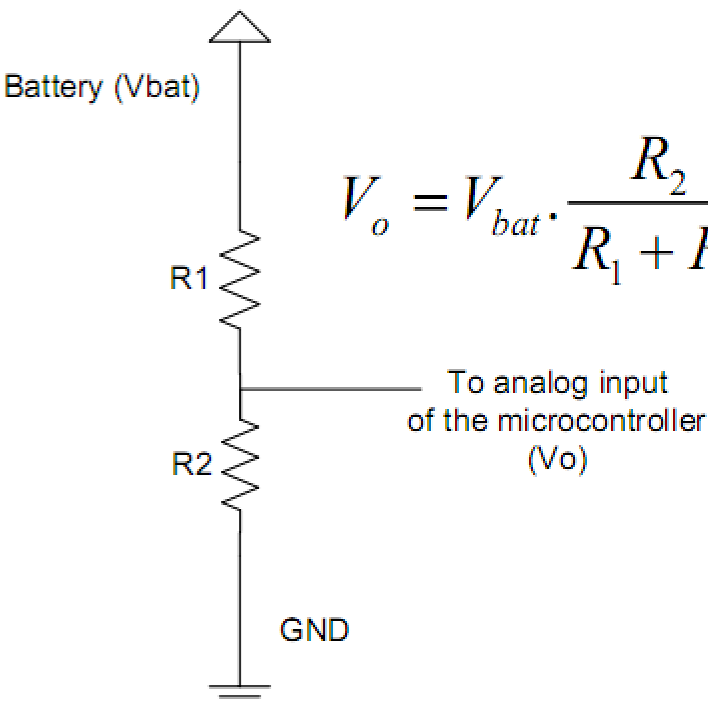
- Excellent light mass
- Optimum mini size
- Cheap and easy charge
- Confident. It's made of fireproof material

We planned to connect 2 batteries in serial on each Cansat. Voltage level will be at 6.6V and battery capacity is 1100 milliAmps for an hour.

Power budgets are obtained in previous slides for both Carrier and Lander. It is planned to be used the same battery for these two. Budget tables show total power consumption for an hour. Selected batteries satisfy with these results. Moreover, there are very much margins for Carrier and Lander.

Battery Voltage Measurement

Battery voltage measurement is done by reading microcontroller's ADC (analog-digital converter) channel. Microcontrollers can take maximum 3.3 Volts* as ADC input. Therefore, battery voltage divider is used to read voltage measurement. Figure shown below represents how to be done this operation.



$$V_o = V_{bat} \cdot \frac{R_2}{R_1 + R_2}$$

We choose this way, because:

- Simple way to divide the voltage
- If resistors are selected as high valued, the current drawn by these resistors is negligible
  - This selection will done with a multimeter readings.
- Weight and coverage area are negligible

* Our selected microcontrollers run at 3.3V.

Flight Software Design

İsmail ULUTÜRK

FSW Overview

- We will use C# with Microsoft Visual Studio Microframework SDK for carrier.
- We will use C with Arduino IDE for lander.
- Flight software:
  - Carrier should release the lander at the right time.
  - Lander should be aware when its released.
  - Read all sensors and GPS data and prepare the data packet for RF Transmission.
  - Store all read data and detailed flight log on SD-Card.
  - Communicate with ground station.
  - Parachutes should be deployed on time.

FSW Changes Since PDR

- **Descent is not controled by the FSW anymore.**
- **Flowcharts are redrawn in a more clear and understandable manner.**
- **We have developed a proof of concept project and it is working on actual hardware so there is no need for any other changes.**

FSW Requirements

ID	Requirement	Rationale	Priority	Parent(s)	Children	VM			
						A	I	T	D
FS-01	Software shall control transceiver via serial interface(CARRIER)	Required for air/ground communications.	High	SR09				X	
FS-04	Software shall transmit data packets at a rate of 0.50Hz at least. (CARRIER)	0.50Hz Telemetry and GPS data transmission is a competition requirement.	High	SR-13	FS-08			X	X
FS-05	Software shall back-up all data read on a SD-CARD. (BOTH)	To be used in case of transmission failure.	Medium	SR-14				X	
FS-06	Software shall keep a log of important flight events and subsystems (BOTH)	Will be used at tests to spot errors and also it will aid us in interpretetion of sensor data.	Low					X	X
FS-07	Shall capture Lander's impact force.(LANDER)	Bonus Objective	Medium					X	

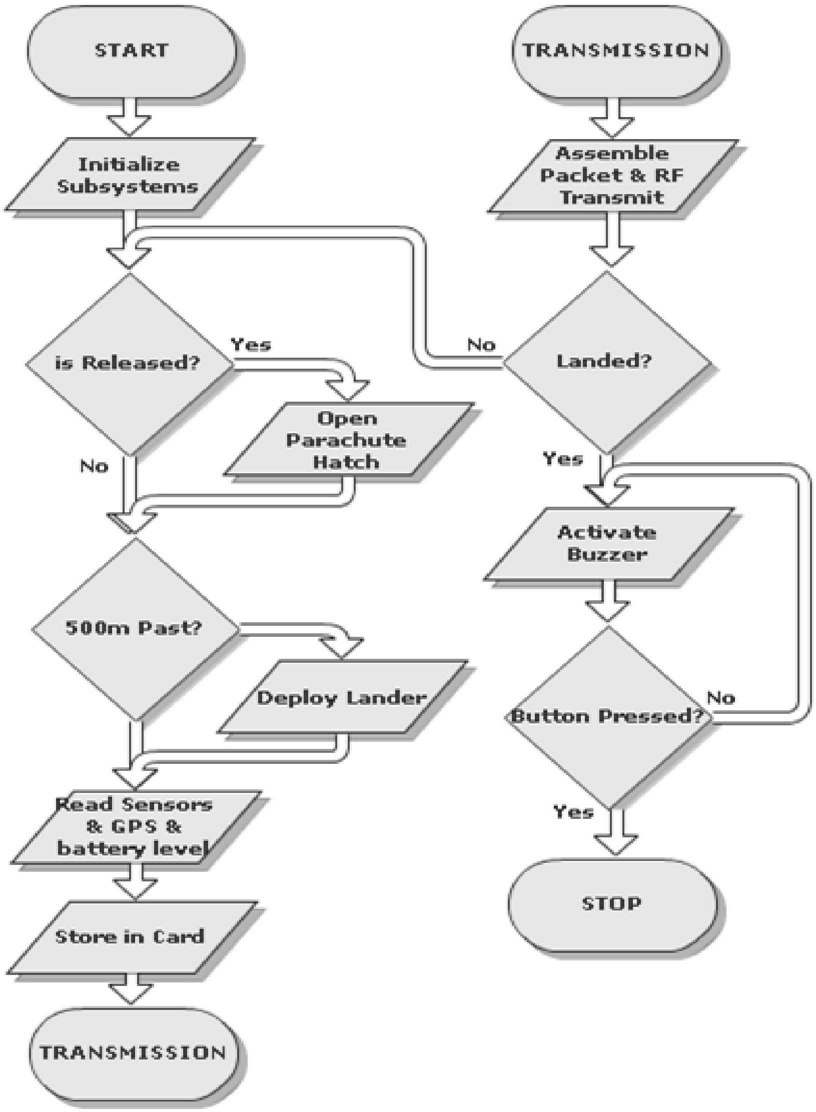
FSW Requirements

ID	Requirement	Rationale	Priority	Parent(s)	Children	VM			
						A	I	T	D
FS-08	Sensors should be sampled at a rate no less than 0.5Hz(BOTH)	Competition requirement.	High	FS-04				X	
FS-09	Parachutes should be deployed on time.(BOTH)	Deploying parachutes early might cause tangling.	High					X	
FS-10	Lander should be deployed at 500m (CARRIER)	Competition Requirement.	High					X	

Carrier Cansat FSW Overview

- Pressure Sensor and Temperature sensor is integrated in one device, SCP1000. It is interfaced via SPI interface at 1Hz. We are using it in triggered mode.
- GPS is interfaced via Serial UART protocol at the Baud Rate of 9600. It outputs data at 1Hz automatically(which is its fastest rate), without any trigger and read by the microcontroller board from UART2 port.
- Communication with ground station is made by the RF link between AC4790 and CL4790. AC4790 is interfaced by Serial UART protocol, connected to a microcontroller from UART1 port at the baud rate of 115,200 .
- Data is stored to a SD Card via SPI protocol by the microcontroller board.
- Data could be easily transmitted at the rate required rate of 2Hz.
- For communications(SPI and UART) and FAT32 file system routines we are using .NET Micro Frameworks standard driver classes and for SD Card SPI mode routines, we are using the class FEZ Framework provides.
- Servo motors are controlled by the duty of a square wave. We are using Output Compare function of the FEZ Framework for that.

Carrier Cansat FSW Overview



Note: 500m past is determined by calculating the altitude from pressure and temperature readings, using the formula present in the Sensor Subsystem part.

Note: Packets are always transmited until landing, because doing so won't complicate the design and we can always use extra data in analysis.

Note: Release from rocket is determined by seeing a stable decrease in altitude for a specified period. We choose that because adding extra sensors will add extra weight and complexity.
More details about the altitude tracking present in the next slide.

Carrier Software Flow Diagram or Pseudocode

- **Pseudocode:**

```
InitializeSubsystems();
startListeningToCommunications();
while(true){ //Infinite Loop
        if( checkRelease() == true ){ //check if cansat seperated from rocket
              openParachuteHatch();
        }
        temp = SCP1000.getTemperature(); //Read Temperature
          pres = SCP1000.getPressure(); //Read Pressure
          logSensorReading(temp,pressure); //Log readings to SD
          while(GPS.readingPresent == false){ //Wait for new GPS data
          Pause(10);// Pause thread for 10ms. Wait for GPS reading.
        }
          gpsData = GPS.getReading(); //Read GPS
          batLevel = readBatteryLevel(); // Read Battery Level
                     if(calculateAltitude(pressure,temp) <= 500){
                         if(Lander.isDeployed == false){
                             Lander.Deploy(); //If 500m reached, deploy lander
                          }
                         Package.Assemble(temp,pres,gpsData,batLevel);
                     Radio.Transmit(Package); //Send package via RF
                }
              if(checkLanding()== true){//Look if pressure sensor reading
                    break;                              //are stabilized.
              }
}
```

Continuing:

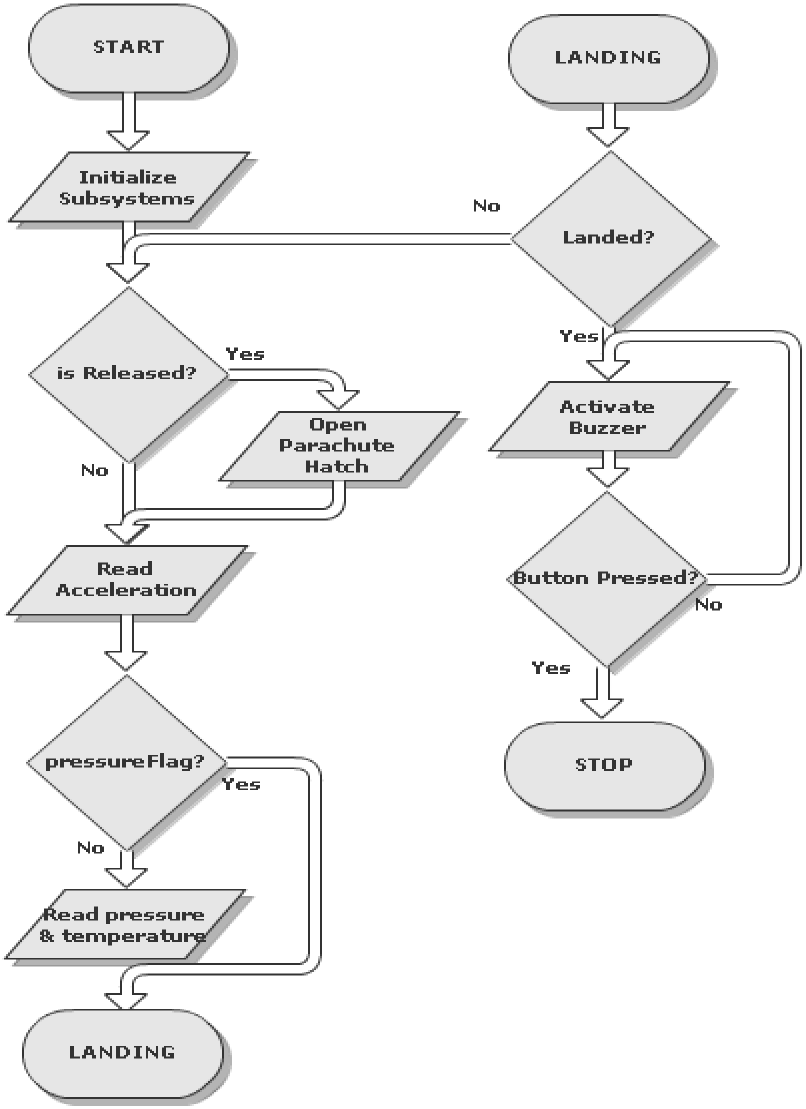
```
Radio.Terminate(100,30);//Send termination package with 100ms
 //interval 30 times or until acknowledgement package arrives. Then
// terminate the RF link.
Buzzer.Activate();
if(button.pressed){
   Buzzer.Deactivate();
}
```

Note: We are using a multi-threaded approach in the carrier software. Sensors are sampled periodically and required calculations/converitons are done in a seperate thread and main program is notified by events when data is ready, so the main program only gets the already prepared data. So information like altitude is always present, calculated using the newest sensor data.

Lander Cansat FSW Overview

- Temperature and Pressure sensors are integrated in one device, BMP085. BMP085 communicates with I^2C protocol, sampled at 5Hz.
- We are using the BMP085 sensor in ultra-high resolution mode. It is read by triggering.
- Data is stored in a MicroSD Card via SPI protocol.
- Arduino Wire Library is used for I^2C communication routines and Arduino FAT library is used for FAT16 filesystem routines and SD Card control.
- Servo motors are controlled by the duty of a square wave. We are using PWM library of Arduino for that.

Lander Cansat FSW Overview



Note: Lander starts reading sensors and storing data as soon as it starts and continues untill landing detected. Since we have almost unlimited storage capacity with the MicroSD Card, this gives us additional data to analyze without any disadvantage.

Note: We are planning to detect release by creating a electrical contact on the restriction mechanism that holds and releases the lander. The contact will be connected to one of the digital IO ports of Lander so Lander will know when the contact is lost.

Note: pressureFlag is set in a periodicaly occuring timer interrupt vector in a desired data rate. 5Hz in this case.

Lander Flow Diagram or Psuedocode

- **Pseudocode:**

```
InitializeComponents();
pressureFlag = 0;
setTimerInterrupt(200); //Configure timer interrupt to hapen every 200ms
   while(true){ //Infinite Loop
if(checkRelease() == true){
     openParachuteHatch(); //Release the parachute
}
if(pressureFlag == 1){
            pressureFlag = 0;
            pres = readPressure();   //Read pressure
            temp = readTemperature(); //Read temperature
            bat = readBatteryLevel();
            storeReading( ‘P’, pres);
            storeReading(‘T’,temp); //Store readings in SD Card.
            storeReading(‘B’,bat);
       }
      accel = readAcceleration(); //Read Acceleration Sensor
      storeReading(‘A’,accel); // Store acceleration data to SD Card.
       if(checkLanding() == true){
           break; // If landed break the loop
       }
 }
  ActivateBuzzer();
  while( buttonPressed == false){
  Pause(10); //Wait for button to be pressed
  }
  DeactivateBuzzer();
```

In the Interrupt Vector://Occurs at the rate of 5Hz
pressureFlag = 1;

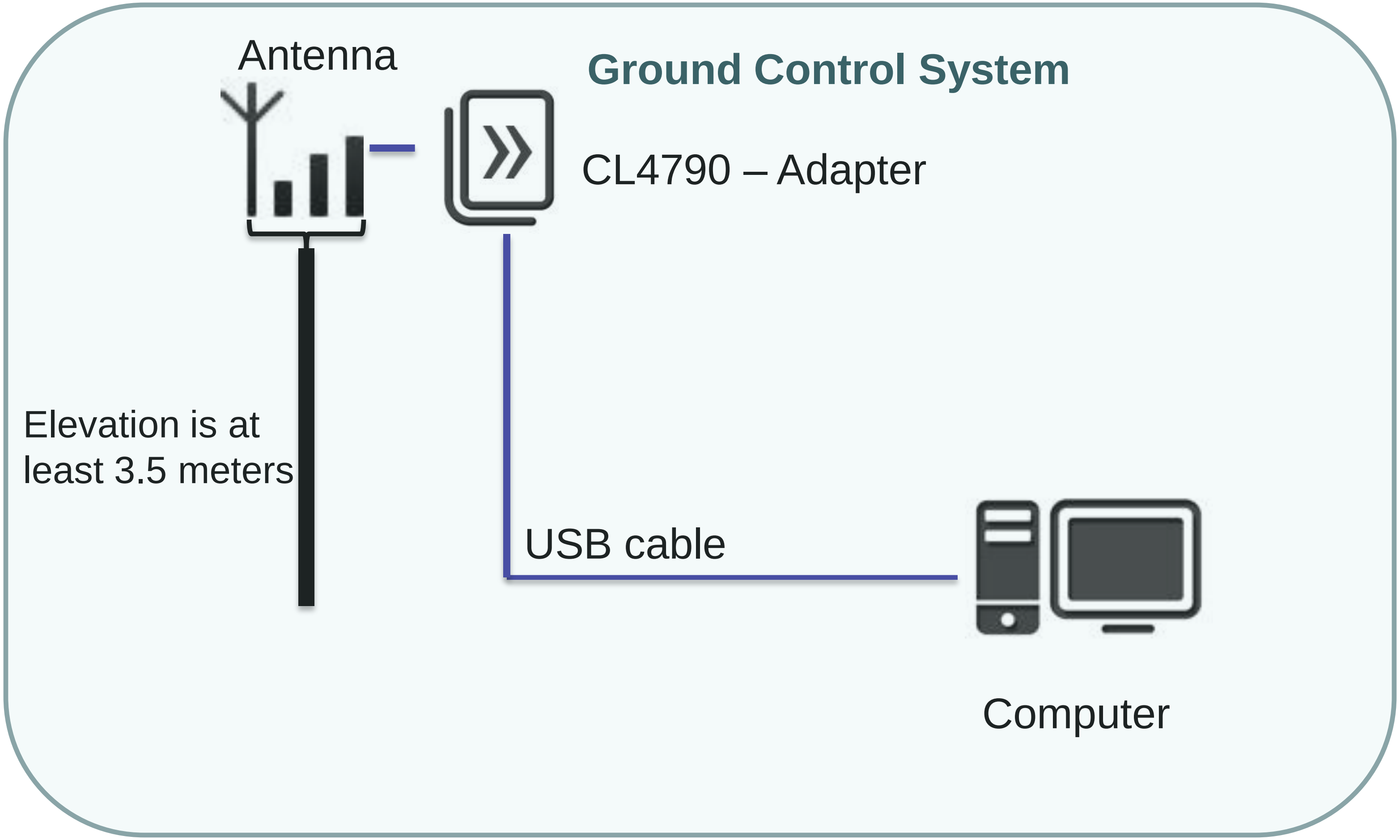
Note: Acceleration sensor is oversampled, not to lose the acceleration at the impact time.

Pressure, temperature and battery measurements are done at 5Hz, because we can do so and extra data will help us to analyze the results.

Ground Control System Design

Çağrı GÜZAY

CanSat (Hezarfen)



GCS Requirements

ID	Requirement	Rationale	Priority	Parent	Children	VM			
						A	I	T	D
GCS -01	Antenna and adapter should be placed at least 3.5 meters above the ground	Clear position of the antenna prevents disconntecting of communication	High				X		
GCS-02	Antenna should stand in facing point to Cansat	Clear position of the antenna prevents disconntecting of communication	High				X		
GCS-03	Adapter range must be higher than 1.5kilometers.	Vertical and horizontal distances (1km) give about 1.5 km (Triangular approximation).	High			X			
GCS-04	Analysis software must be fast enough and be useful	Miscalculation gives wrong results.	High	SR-12		X		X	
GCS-05	Data should be received properly from Carrier	To do accurate analysis	Medium	SR-12		X	X	X	

GCS Requirements

ID	Requirement	Rationale	Priority	Parent	Children	VM			
						A	I	T	D
GCS -07	It is needed to be drawn graphics for some measurements	To obtain time related variations	Medium	SR-12			X		X
GCS-08	Some measurements should be processed	Altitude, descent rate etc. are not measured directly. These are observed from some measurements by some calculations	High			X	X	X	
GCS-09	Some measurements should be compared with calculations	To be sure measurements are correct	Low			X		X	

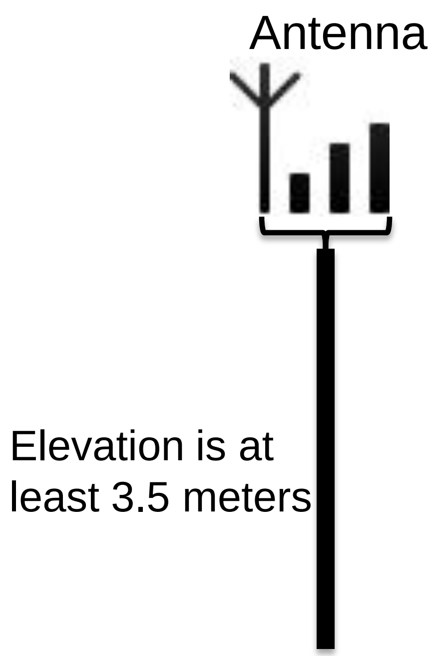
GCS Antenna Selection

We planned to use the antenna coming with CL4790. This antenna has 2 dBi gain. In PDR, We would make only antenna elevated and connect it to CL4790 with shielded antenna cable. If we use this cable, gain decreases. To keep gain at 2dBi, We planned to elevate CL4790. After that, We use long (~3.5m) USB cable to connect it to ground station computer.

Antenna should;
•be elevated.
•be across to the Cansat.
•be open place.

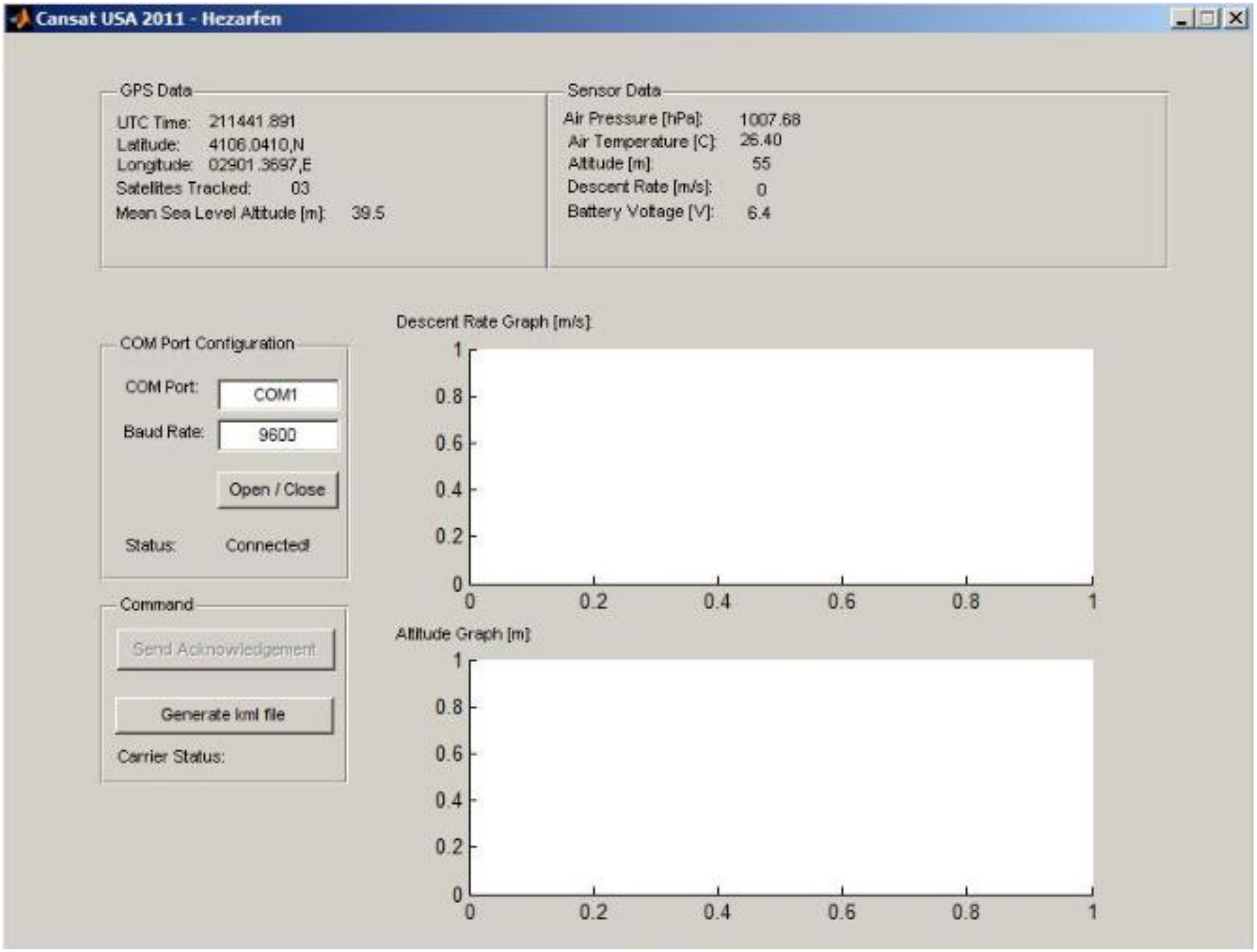
Maximum vertical distance from Cansat to GCS will be 1 km.
Maximum horizontal distance from Cansat to GCS may be 2 km.

Predicted Distance	**2.24 km**
Distance Available	~6 km
Margins	~4 km



We planned to make a portable mast with plastic pipes.

GCS Antenna Selection



Screenshot of the our GCS software is left. When Carrier stands still this data is taken.

<u>Note:</u>
We could not complete descent rate and altitude graphs yet.

CanSat Integration and Test

İsmail Ulutürk

The integration of subsytems

Firstly, the structure of cansat is build. The mechanical subsystems are integrated on the structure. The descent control systems such as parachutes are assembled. In orderly, EPS, CDH, FSW systems are integrated to provide the connection of all subsytems. Finally all subsytems are work together properly as planned.

Mechanic Subsystem Tests

→ Prototype building

→ Seperating test

Descent Control Test

→ Unlocked parachute test

→ Opening Parachute test

Electronic System Test

→ Sensor Subsystem Test

→ CDH Subsystem Test

→ EPS Test

→ FSW Test

→ GCS Test

Cansat Integration and Test Overview

First building cansat structure → Mechanical subsystem → Descent control system

Set up EPS → Set up CDH → Set up FSW system

Test Equipment

→Tools for assembly
→Airflow supplier
→Multimeter
→Computer
→Oscilloslope

Sensor Subsystem Testing Overview

- **Carrier:**
  - Pressure & temperature sensor sends data to microcontroller. This data is monitored on computer.
    - Aim: To obtain air pressure, air temperature, altitude, descent rate.
    - Constraints: This sensor is tested but not fully integrated. We could move it by a little in the room. So, change of pressure, temperature and altitude are at very low level.
    - Pass/fail: After testing, We examined test data. The data fits our reference sensor, placed on our lab. and its results are trusted. We can say pressure & temperature sensor of Carrier works fine.
  - GPS module sends data to microcontroller. This data is monitored on computer.
    - Aim: To obtain UTC time, coordinates, number of tracked satellites and mean sea level altitude.
    - Constraints: We have a limited movement in the room. We could not see coordinates in high level. We have no chance to know number of tracked satellites.
    - Pass/fail: We compared coordinates with Google Earth. As expected us, results have a negligible differences. Also, We control the UTC time data. Our region has +2 hour specification. So, our testing time is forward from the obtained UTC time. On the other hand, mean sea level altitude is different a little from source of meteorology institute. Number of tracked satellites is difficult to verify. Finally, GPS module works as our expectations.

Note: Test data is placed in *“Sensor Subsystem Design”* section.

Sensor Subsystem Testing Overview

- **Lander:**
  - Pressure sensor sends data to microcontroller. This data is monitored on computer.
    - Aim: To obtain air pressure, altitude, descent rate -(air temperature).
    - Constraints: This sensor is tested but not fully integrated. We could move it by a little in the room. So, change of pressure, altitude -(temperature) are at very low level.
    - Pass/fail: After testing, We examined test data. The data fits our reference sensor, placed on our lab. and its results are trusted. We can say pressure sensor of Lander works fine.

Note: Test data is placed in *“Sensor Subsystem Design”* section.

Lander Impact Force Sensor Testing

We decided to work on Lander impact force determination.

Acceleration sensor sends data to microcontroller. This data is monitored on computer.

Aim: To obtain impact force of Lander’s touching to the ground.

Constraints: We have a limited movement in the room. We could not simulate of Lander’s landing.

Pass/fail: We give some pulses to the acceleration sensor with hand. At that moment, sensor data is stored. After testing, this data is graphed and, peak values of pulses are obtained. According to these values, We roughly calculate acceleration of impact. Since We know mass of Lander, impact force can be calculated easily.

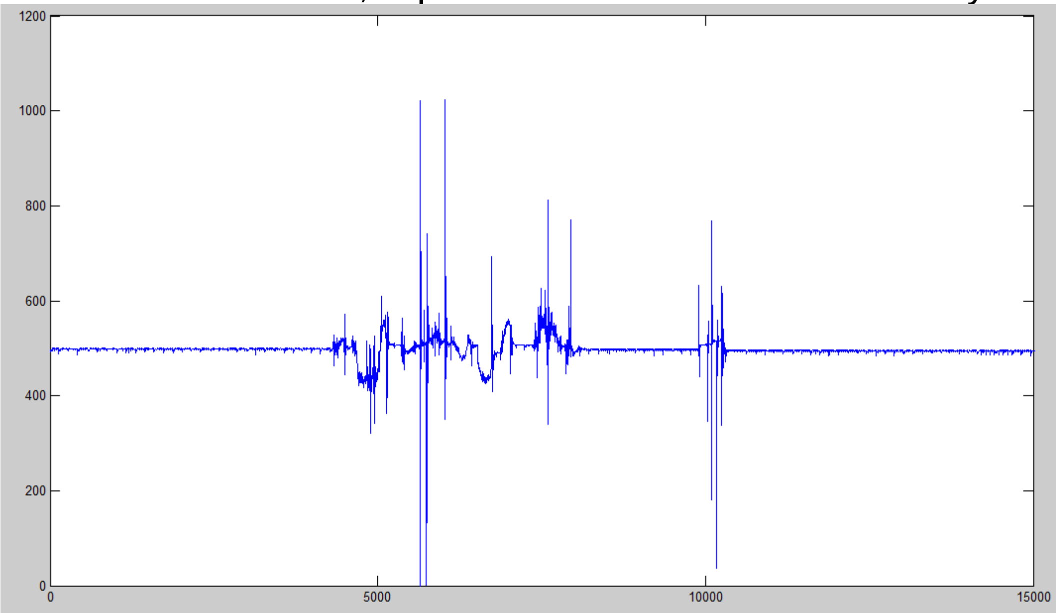


Figure at left shows the acceleration data in our tests. Y-axis is voltage level. These values are converted acceleration value by selected g-range of sensor.

Note: Test data is placed in *“Sensor Subsystem Design”* section.

DCS Subsystem Testing Overview

1-) Unlocking Parachute Test:

The parachute systems of the lander and the carrier have been entegrated to mock up of cansat. The lander's parachute and the carrier's parachutes are located between different two carbon fiber plates. The top of the carbon fiber plates are locking plate. These carbon fiber locking plates will be positioned by micro servos. Firstly we tried this lock system during the drop test in our laboratory. We accomplished this test manually. The unlock parachutes systems are working.

2-) Opening Parachute Test:

After unlocking, the lander's and the carrier's parachute will be self opening with air flow and parachutes will fill with air. We did this test on the our vertical wind tunnel in our laboratory. We observed the opening parachutes. Opening parachute tests were succesfully done.

Mechanical Subsystem Testing Overview

1. Prototype Building

A prototype has been built in order to design and CAD. It is for ensuring to have a robust structure. In this case some structure elements such as steel bars, carbon fiber plates,egg protection, bolts and nuts have been used.The constraints are not having all the elements. So, a simple structure has been built. The test provide us to get the mechanical layout of the cansat. Photos of prototype are shown right side.

2.Seperating Test

Servo controlled seperating system will be used for our cansat. So servo should be placed in a suitable position and it should have a free space to work properly. In this case we assembled servo on the carbon fiber plate to see how it works. The is test is accomplished by our hands. The servo's arm length has been adjusted and connection of the lander and carrier is adjusted using parts for the assembly. So we get the knowledge of how the connection should be broken and where to place the servo.

CDH Subsystem Testing Overview

- We created a test bed for CDH subsystem of carrier, that could be seen in the right. We put sensors,GPS and SD Card in a box and put them out of window and took readings for a while.
  - Pressure reading were stable, and responsive to changes and they were similar to results we got from other trusted and known to be working sensors.
  - GPS was outputing stable data at the specified 1Hz data rate.
  - Temperature sensor was outputing stable and responsive data. When it is taken close to a heat source could easily be seen from the logs.
  - SD Card was working as expected, we were able to read our logs by just pluging it to a computer.
  - We tested the radios AC4790 and CL4790 but we didn’t fully integrated them to our systems yet, it is the first thing we will do next. We successfuly communicated with ground station from carrier on tests, but did not implemented the communication part to the flight software yet. As soon as we implement that, we will test it by starting the systems and taking the CanSat far away in the line of sight. If we can successfully transmit all the telemetry data from 1.5km distance, than we won’t have any problem in the mission.
  - We made all test in our room, so we could not test the systems in a big scale. After radios are fully implemented we will do a bigger scale test.(like throwing it from roof.)

CDH Subsystem Testing Overview

Example Log of Carrier:

6674646,’’C",’’Starting to Read."

6675642,"R",10.50,1014.05

6686338,’’R","$GPGGA,211441.891,4106.0410,N,02901.3697,E,1,03,8.4,-39.5,M,39.5,M,,0000*4A’’

6686488,"C",’’Starting to Read"

6687237,"R",10.60,1014.09

6697839,’’R","$GPGGA,211442.891,4106.0409,N,02901.3696,E,1,03,8.4,-39.5,M,39.5,M,,0000*40’’

6697994,"C", ’’Starting to Read."

6698618,’’R",10.50,1014.05

6709224,’’R","$GPGGA,211443.891,4106.0409,N,02901.3696,E,1,03,8.4,-39.5,M,39.5,M,,0000*41’’

6709385,"C", ’’Starting to Read."

6709998,"R",10.50,1014.04

6720916,’’R","$GPGGA,211444.891,4106.0408,N,02901.3696,E,1,03,8.4,-39.5,M,39.5,M,,0000*47’’

6721076,"C", ’’Starting to Read."

6721680,"R",10.50,1014.10

6732316,’’R","$GPGGA,211445.891,4106.0408,N,02901.3696,E,1,03,8.4,-39.5,M,39.5,M,,0000*46’’

First column is timestamp, second column is ‘R’eading or ‘Comment’, third column is data, GPS NMEA string or temperature,pressure.

CDH Subsystem Testing Overview

- We created a test bed for CDH subsystem of lander. We put sensors and MicroSD Card together and took readings to the card.
  - Pressure reading were generally stable and correct but it had a little fluctuation when it was standing still. We get over it by implementing a simple median filter in the software.
  - Temperature sensor was outputing stable and responsive data. When it is taken close to a heat source could easily be seen from the logs, like carrier.
  - MicroSD Card was working as expected, we were able to read our logs by just pluging it to a computer.
  - Acceleration sensor was also giving stable readings, at a fast rate.Since we want the acceleration data at the moment at impact, reading fast is important here. There is an example reading in the next slide.
  - We made all test in our room, so we could not test the systems in a big scale. After radios are fully implemented we will do a bigger scale test.(like throwing it from roof.)

millis, accel
188 , 484
194 , 483
200 , 484
204 , 484
210 , 483
217 , 483
221 , 483
227 , 482
233 , 483
237 , 483
243 , 483
249 , 484
253 , 483
260 , 484
266 , 482
270 , 484
276 , 483
282 , 483
286 , 483
292 , 483
299 , 484
303 , 483

millis, Pressure, Temperature*0.1, Altitude(cm)
1036 , 100133 , 266 , 10366
2037 , 100137 , 266 , 10331
3037 , 100132 , 266 , 10375
4034 , 100128 , 266 , 10410
5038 , 100121 , 266 , 10471

Note:
Readings are normally stored in the same file, but because the file is so big and acceleration readings are taken in a much more higher data rate than the pressure readings, we seperated them for this slide.

EPS Testing Overview

- **For one time, (for both Carrier and Lander)**
  - After installation of all circuits, We will make short circuit test for any misconnection.
  - We measure the battery voltage with multimeter.
- **Carrier:**
  - We put a LED-resistor combination at some electrical devices; pressure & temperature sensor, GPS module, SD-card unit. Also, We mount a LED to AC4790 indicator pin. Microcontroller has own LED indicating power condition.
    - Aim: To control all electrical devices, if they are supplied with correct voltage levels.
    - Constraints: Broken down LEDs may be deceptive. Current consumption is important point. So, We do not use the indicator LEDs with buzzer and servo motors.
    - Pass/fail: Until this time, all devices have worked properly. Test data is taken with this configuration. After installation of LEDs, We will have more control.
  - We put a power switch on battery.
    - Aim: To make power fully controlled.
    - Constraints: -
    - Pass/fail: Until this time, We have not used this switch. We guess it will work as expected.

EPS Testing Overview

- **Lander:**
  - We put a LED-resistor combination at some electrical devices; pressure sensor, acceleration sensor, micro SD-card unit. Microcontroller has own LED indicating power condition.
    - Aim: To control all electrical devices, if they are supplied with correct voltage levels.
    - Constraints: Broken down LEDs may be deceptive. Current consumption is important point. So, We do not use the indicator LEDs with buzzer and servo motors.
    - Pass/fail: Until this time, all devices have worked properly. Test data is taken with this configuration. After installation of LEDs, We will have more control.
  - We put a power switch on battery. This makes full power control.

FSW Testing Overview

- We tested FSW for taking readings from all sensors and storing them on SD Cards for both carrier and lander. We let the system run for a long time and it worked without any problem.
  - We looked for stability of the data and the readings were stable.
  - We looked for correctness of the data and conversion operations, and they were correct.
  - We tested if we can use FAT32 filesystem on SD Cards, and we succeed.
  - We put all these together, run a test and it worked without any problems.
- The next step for FSW is testing communication routines. After they are tested, we will be ready to make a simulation flight.

- If the FSW works as expected in simulation flight, then it means it is ready.
  - Working as expected means:
    - Reading and storing all sensor data
    - Handling the communication
    - Releasing the lander on time
    - Opening the parachute
    - Detecting landing, activating buzzer and stoping communication.
    - Deactivating buzzer when button pressed.

GCS Testing Overview

- **AC4790 Radio module communicated with ground station (CL4790 - Adapter). A few data sent by AC4790.**
  - Aim: To communicate Carrier with ground station computer.
  - Constraints: We could not test the GCS with full integration.
  - Pass/fail: In first sight, communication between AC4790 (radio module of Carrier) and CL4790 (adapter of ground station computer) is accomplished.
- **Ground station computer uses Matlab software.**
  - Aim: To monitor all telemetry data.
  - Constraints: Matlab may disconnect from CL4790 rarely.
  - Pass/fail: Communication is achieved. We can monitor telemetry data on Matlab.

Mission Operations & Analysis

Hasan Erdem Harman

Overview of Mission Sequence of Events

1. Preflight briefing
2. Last mechanic control
3. Last electronic control
4. Coming competition area
5. Placing competition platform
6. Preparing materials
7. Making egg protection system
8. Putting egg in to the CanSat
9. On cansat
10. Put cansat into rocket
11. Launch rocket
12. Leaving cansat from rocket

Overview of Mission Sequence of Events

Measuring height of cansat → Seperate cansat on 500meters → Open parachutes → Control of descent speeds

Collect data using sensors → Calculate collusion force → Data saving on ground station → Locate Cansat

Off cansat → Analyzing saving data → Preparing PFR → Presentati on PFR

Lander Landing Coordinate Prediction

- **We will use the sensor values transmitted between the launch and the deployment of the lander to estimate the horizontal speed of the CanSat with respect to the ground at the moment of release.**
- **Lander will most likely have a horizontal speed with almost similar direction, but the amplitude might vary.**
- **We will write our own analysis script in MATLAB.**
- **Since carrier is sending telemetry data, we have the GPS values of CanSat, and if we assume that our DCS is working, we approximately know the falling speed. When we wanted to know the location of lander, we will run our MATLAB script with those data and get the approximation.**

Cansat Location and Recover

- We will make predictions about the location of Lander with the method explained in the previous slide.
- We will make predictions about the location of Carrier using the transmitted GPS data, the direction of its horizontal speed calculated using transmitted data, we will also keep the wind speed in mind.
- Both Carrier and Lander will have loud buzzers activated after landing, so finding them should not be so hard after we have their locations roughly.

Mission Rehearsal Activities

- **Radio Link Check Procedure:**
  - We set up the link by connecting the CL4790 to the computer and AC4790 to the controller, and we managed to send messages from carrier to ground station, however the AC4790 is not implemented as it will be used in the competition yet so we have not tested it fully yet.
- **Loading egg payload:**
  - We made the protective dough and placed it in the case of the egg. Then we put the egg in and dropped it in to the ground from the fifth floor of our dormitory and it did not break.
- **Telemetry processing, archiving, and analysis:**
  - We created a test bed for reading all telemetry data and storing them on SD Cards in forms of flight logs with extra data. And we used a MATLAB script to do the processing on the data and checked if they are correct. We are archiving all test results in the SD Card and on our computers as well, we did not delete not even one sensor reading since we started working.

Mission Rehearsal Activities

- **Powering on/off CanSat:**
  - We will have a main power switch to power on/off CanSat. We will test it by observation.
- **Launch configuration preparations:**
  - We will fully assemble the CanSat and make measurements to make sure it is at our expected dimensions. Sensors and SD Cards have LEDs indicating their status. We will use the LED on board of FEZ Mini as general ready-for-flight led, if there is a failure in the initialization of electrical subsystems, we will be warned by that LED.
- **Recovery:**
  - We will make a demonstrative flight when our CanSat is complete, before competition. It will be on a smaller scale, but will be enough for us to see if all the systems are working properly, including recovery.

Management

Uğur ÖZEN

Status of Procurements

- **Status of procurements**
  - All sensors and component procurements are ordered except camera. Because bonus objective is selected as caculating collusion force.
  - Software developed.
  - Protype is produced.

Cansat Budget

Hardware Costs

	Category	Model	Quantity	Unit Cost	Determination
ELECTRONiC	Controller Board (Lander)	Arduino Pro Mini 328-3.3V/8MHz	1	19$	Actual
	Controller Board (Carrier)	Fez Mini	1	50$	Actual
	GPS Receiver (Carrier)	20Channel LS20126 GPS Receiver	1	60$	Actual
	Temperature + Pressure Sensor	SCP1000 with Breakout Board	1	35$	Actual
	Accelometer (Lander)	MMA7260Q	1	20$	Actual
	RF Transciever (Carrier)	AC4790-200M	1	51$	Actual
	RF Antenna (Carrier)	Laird 0600-00019	1	12$	Actual
	Pressure Sensor (Lander)	BMP085 with Breakout Board	1	20$	Actual
MECHANiC	Structure Materals	Silver steel	1	40$	Actual
	Parachute		2	10$	Actual
	Servo	Hitec HS-65HB Micro Carbonite	3	12$	Actual
	Egg protective mix	Dough	1	7$	Estimate
	SUBTOTAL			**350$**	

Cansat Budget

Ground System Costs

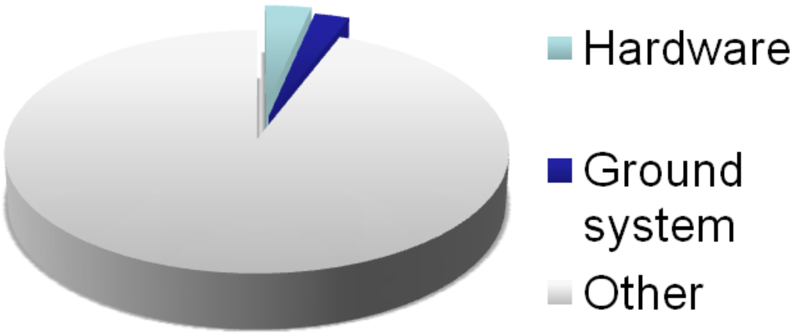
Category	Model	Quantity	Unit Costs	Determination
Antenna Cable	RG195 ANT Cable RPSMA Jack/Plug	1	56$	Actual
Wireless Adapter	Laird Tech. – CL4790	1	170$	Actual
Computer	Asus F3Sv	1	-	Actual
Miscallenous	Wires, expansion equipment		74$	Estimate
SUBTOTAL			**300$**	

Other Costs

Category	Quantity	Unit Costs	Determination
Travel	6	1000$	Actual
Hotel	3	700$	Estimate
Van Rental	1	1000$	Estimate
Food	6	350$	Estimate
SUBTOTAL		**11200$**	

Income-Cost Equilibirium

Category	Cost
Hardware	350$
Ground System	300$
Others	11200$
Suprise Costs	4150$
Total Cost	16000$
Income	16000$



→Source of project income is Rectorship of Istanbul Technical University, Istanbul Ticaret Odası, TAI and Tubitak.

→The project can be fit with income and cost equilibirium

Program Schedule

Date		Mechanics	Electronics	Fin. & Log.	Academic Schedule	High Level Task
		Categories				
November	1.week	-	-	Team come together	School first term Quizes Homeworks	Getting information about competition
	2.week					
	3.week	Recognation of competition task	Recognation of competition tasks	Recognation of competition tasks		
	4.week					
December	1.week	Design mechanic systems	Design electronic systems	Searching for Sponsorship		Doing subsystems design
	2.week					
	3.week	Determining of using materials	Determining of using materials			
	4.week					
January	1.week	Buying materials	Buying materials		Midterm Final Exams	Assign final materials which are used on competition
	2.week	Testing mechanical materials	Testing electronical materials	Prepare PDR		
	3.week					
	4.week				Midterm Holiday	
February	1.week	Break	Break	Break		Break
	2.week				Starting second term	
	3.week					
	4.week					

Program Schedule

<table>
<tr><th colspan="2"></th><th colspan="5">Categories</th></tr>
<tr><th colspan="2">Date</th><th>Mechanics</th><th>Electronics</th><th>Fin. & Log.</th><th>Academic Schedule</th><th>High Level Task</th></tr>
<tr><td rowspan="4">March</td><td>1.week</td><td rowspan="2">Put together subsystems</td><td rowspan="2">Put together subsystems</td><td rowspan="2">Searching for testing area</td><td rowspan="11">School second term

Quizes
Homeworks</td><td rowspan="4">Production of cansat prototype</td></tr>
<tr><td>2.week</td></tr>
<tr><td>3.week</td><td rowspan="2">-</td><td rowspan="2">Creating ground system & software</td><td rowspan="2">Prepare CDR</td></tr>
<tr><td>4.week</td></tr>
<tr><td rowspan="4">April</td><td>1.week</td><td rowspan="4">Testing cansat</td><td rowspan="4">Testing cansat</td><td rowspan="2">Buying fly tickets</td><td rowspan="4">Testing Cansat</td></tr>
<tr><td>2.week</td></tr>
<tr><td>3.week</td><td rowspan="2">-</td></tr>
<tr><td>4.week</td></tr>
<tr><td rowspan="4">May</td><td>1.week</td><td rowspan="4">Last controls and evaluating of cansat mechanic systems</td><td rowspan="4">Last controls and evaluating of cansat electronic systems</td><td rowspan="4">Hotel reservations
&
Taking visa</td><td rowspan="4">Full control of cansat and evaluating</td></tr>
<tr><td>2.week</td></tr>
<tr><td>3.week</td></tr>
<tr><td>4.week</td><td rowspan="2">End year Final Exams</td></tr>
<tr><td rowspan="2">June</td><td>1.week</td><td rowspan="2">-</td><td rowspan="2">-</td><td>Preparing cansat competition</td><td rowspan="2">Participating to cansat competition</td></tr>
<tr><td>2.week</td><td>Cansat competition</td><td></td></tr>
</table>

Conclusions

Major accomplishments

→ The team is come together

→ Subsystems are designed

→ Materials are selected

→ Cost and income are balanced

→ Software is developed

→ Subsystems are tested

→ Passport applications were started

→ Cansat protype is produced

Major unfinishes work

→ We will test cansat for all duties.

We were succesfull all duties until now.

We will go on according to schudle until competition.