

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

## IAP\_SRWDA-SAT

1<sup>st</sup> Project Report (2012-2013)

Conception and Definition phase

Initial document: Ras Nhache/Heidelberg, 6 June 2013

Last update: 16.01.2014

Authors:

Samir Mourad

Mahmoud Zohby



A member institute of AECENAR  
[www.aecenar.com/institutes/iap](http://www.aecenar.com/institutes/iap)



# Content

<b>ABSTRACT .....</b>	<b>4</b>
<b>1 IAP SPACE PROGRAM 2012 - 2015 .....</b>	<b>6</b>
1.1 PROGRAM OVERVIEW .....	6
1.2 SCIENTIFIC PROGRAM .....	6
1.2.1 Pulsar Radio Astronomy .....	6
1.2.2 The SRWDA Project.....	7
1.2.3 Ground Station for Radio Astronomy: IAP_SRWDA .....	7
1.3 SATELLITE DESIGN AND CONSTRUCTION .....	7
1.4 OLD PROGRAM (LAST UPDATED DEZ. 12) .....	8
1.5 AIM OF IAP_SRWDA-.SAT PROJECT .....	8
<b>2 PROJECT MANAGEMENT.....</b>	<b>9</b>
2.1 ORGANISATION .....	9
2.2 WORKING PACKAGES AND TIME PLAN .....	9
2.2.1 IAP 2013 .....	9
2.3 COSTS .....	10
2.3.1 IAP 2012 .....	10
2.3.2 IAP 2013 .....	10
2.3.3 IAP 2014 (planned).....	10
<b>3 BASICS .....</b>	<b>11</b>
3.1 SATELLITE BUS DESIGN .....	11
3.1.1 Example: 10 ccm Student Nanosatellite OUFTI-1 of University of Liège .....	11
3.2 ENTWURF DES ENERGIEVERSORGUNGSSYSTEMS .....	12
3.3 ENTWURF DES LAGEREGLUNGSSYSTEMS .....	14
3.3.1 Sensors.....	14
3.3.2 Actuators.....	15
3.4 MISSION SIMULATION.....	15
3.5 SHUTTLE SYSTEM .....	16
<b>4 CANDIDATE SOFTWARE TOOLS.....</b>	<b>17</b>
4.1 POSSIBLE DEVELOPMENT ENVIRONMENT FOR COMMUNICATION AND CONTROL .....	17
4.1.1 Scilab .....	17
4.1.1.1 Zusatzpakete .....	18
4.1.1.2 Scicos.....	19
4.2 NEARSPACECRAFT SIMULATOR DEVELOPMENT .....	20
<b>5 SPECIFICATION.....</b>	<b>21</b>
<b>6 SYSTEM CONCEPT.....</b>	<b>22</b>
6.1 OVERVIEW .....	22
6.1.1 Satellite Bus .....	22
6.1.2 Propulsion System.....	22
6.1.3 Power Supply System.....	33
6.1.4 Structures and Mechanisms .....	35
6.1.5 Thermal Control System .....	35
6.1.6 Flight Control System (FCS) on board of satellite – Attitude control system (ACS) .....	37
6.1.6.1 Gear .....	38
6.1.7 Command and Data Handling .....	39
6.2 EDUCATION AND PUBLIC OUTREACH.....	39
6.3 FIRST MOCKUP MODEL.....	40

6.4	PROTOTYPE.....	41
<b>7</b>	<b>MISSION SIMULATION 2012: SHUTTLE FLIGHT SIMULATION INTO ORBIT.....</b>	<b>42</b>
<b>8</b>	<b>TELEMETRY (SATELLITE BUS COMMUNICATION) CONCEPTION.....</b>	<b>46</b>
8.1	FROM SBEP PROJECT.....	46
8.2	SATELLITE SIDE.....	47
8.2.1	<i>System Design</i> .....	47
8.2.2	<i>Working packages</i> .....	48
8.3	GROUND STATION SIDE.....	49
8.3.1	<i>GUI Alternative 1: Using the old alternative Lotte program</i> .....	49
8.3.2	<i>GUI Alternative 2: In Scilab</i> .....	50
8.4	TRANSCIVER TECHNOLOGY.....	50
8.4.1	<i>Specification</i> .....	50
8.4.1.1	STEP 1 (until 15 June 2013).....	51
8.4.1.2	STEP 2 (until 30 June 2013).....	52
8.4.1.3	STEP 3: Improvements.....	53
8.4.1.4	Antenna improvement.....	54
8.4.1.5	Channel Coding.....	54
8.5	PAYLOAD COMMUNICATION: SCS-SMS INCLUDING AES.....	54
<b>9</b>	<b>TELEMETRY (SATELLITE BUS COMMUNICATION) PROTOTYPE.....</b>	<b>55</b>
9.1	OVERVIEW.....	55
9.1.1	<i>Synthesize transceiver STD-402</i> .....	55
9.2	BASE STATION SIDE.....	55
9.3	SATELLITE SIDE.....	55
9.3.1.1	Hardware part.....	56
9.3.1.2	ACU actuator.....	57
9.3.1.3	GPS sensor.....	57
9.3.1.4	PIC board.....	57
9.3.1.5	Antenna control board.....	61
9.3.2	<i>Software part</i> .....	61
9.3.2.1	Communication code.....	61
9.3.2.2	Control and stability code.....	63
9.3.2.3	Coding code.....	63
9.3.2.4	Antenna control code.....	63
9.4	SYSTEM CODING.....	63
9.4.1	<i>Introduction</i> .....	63
9.4.2	<i>Source Coding</i> .....	64
9.4.2.1	Using Huffman theory.....	65
9.4.2.2	Write the Huffman code (c programming).....	67
9.4.3	<i>Encryption</i> .....	71
9.4.3.1	Advanced Encryption Standard (AES).....	71
9.4.3.2	AES algorithm.....	71
9.4.3.3	Encryption and Decryption AES flow chart.....	75
9.4.3.4	AES program (C programming).....	76
<b>10</b>	<b>SCIENTIFIC PROGRAM.....</b>	<b>77</b>
10.1	DATA COLLECTION FROM GROUND STATION.....	77
10.1.1	<i>System Design</i> .....	77
10.2	SATELLITE BASED RADIO ASTRONOMY SENSOR.....	78
<b>11</b>	<b>LABORATORIES.....</b>	<b>79</b>
	<b>LITERATURE.....</b>	<b>80</b>

## Abstract



IAP - Institute for Astronomy and Astrophysics

مركز ابحاث لعلم الفلك و فيزياء النجوم



## Satellite System for Radio Astronomy

### IAP\_SRWDA-SAT

Last update: 17 December 2013

Electric Propulsion for orbit and attitude stabilization

First Mockup Model

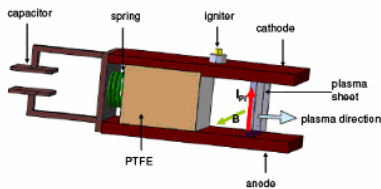
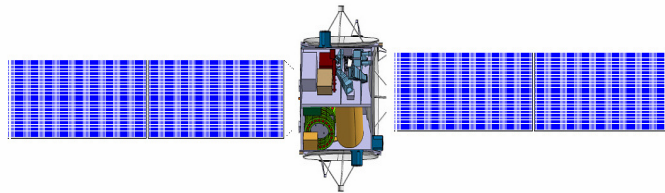


Figure 1. Working Principle of an I-MPD



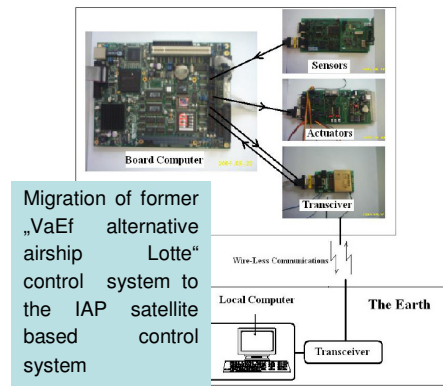
Detection of HI Lines in Interstellar Media

### Actual Working Packages:

- Refinement of the first Mockup Model
- Development of Prototype with parts of IAP ECS and alternative Lotte FCS
- Mission Simulation
- Simple Prototype for the MPD Propulsion Device

### Financial and man power needs for 2014:

- 2-3 persons (e.g. as master theses)
- 4.500 USD material



### Contact

Samir Mourad

Mobile Lebanon ++961 76 341 526

Mobile Germany ++49 (0)176 4039 2038

Email: samir.mourad@aecenar.com





IAP - Institute for Astronomy and Astrophysics

مركز أبحاث لعلم الفلك و فيزياء النجوم

<http://aecenar.com/institutes/iap>



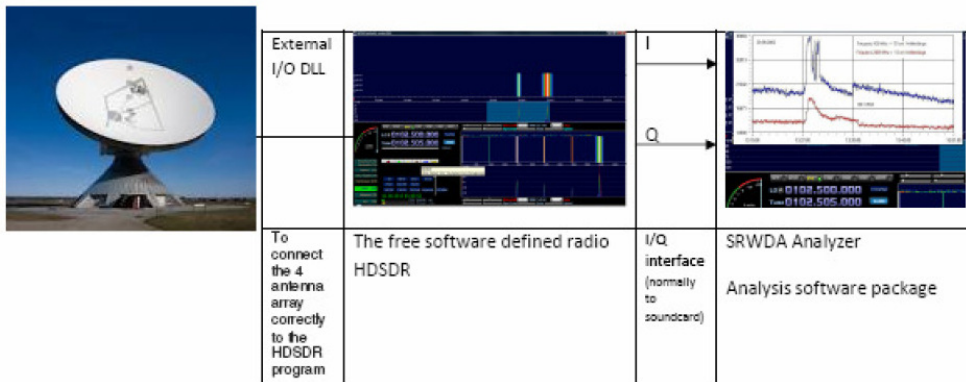
## Ground Station for Radio Astronomy

### IAP\_SRWDA

Last update: 06 December 2013

#### The SRWD project

The radio astronomical IAP project supernova radio wave detector and analyzer (SRWDA) aims to detect and analyze HI radio signals from supernova remnants. At the base station a set of antennas, which gives also direction information of the signals, is aimed to be connected to a computer which acts as Software Defined Radio (SDR). Afterwards an analyzing program is aimed to be installed. Later the detectors are planned to be installed on satellites to improve the resolution and to suppress disturbing signals from earth stations.



#### Actual Working Packages:

- Development of Prototype with parts of IAP ECS

#### Financial and man power needs for 2014:

- 1 persons (e.g. as PhD thesis) with student support (e.g. as master thesis)
- 10.000 USD material

#### Contact

Samir Mourad  
 Mobile Lebanon ++961 76 341 526  
 Mobile Germany ++49 (0)176 4039 2038  
 Email: samir.mourad@aecenar.com

# 1 IAP Space Program 2012 - 2015

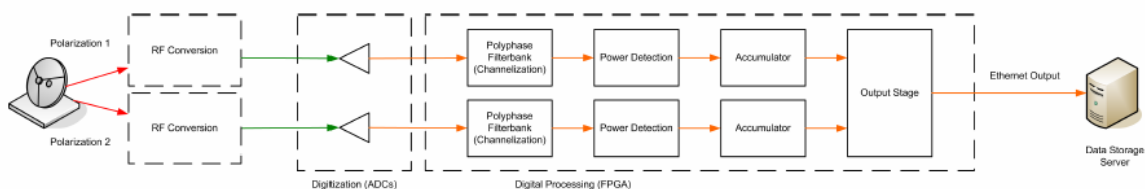
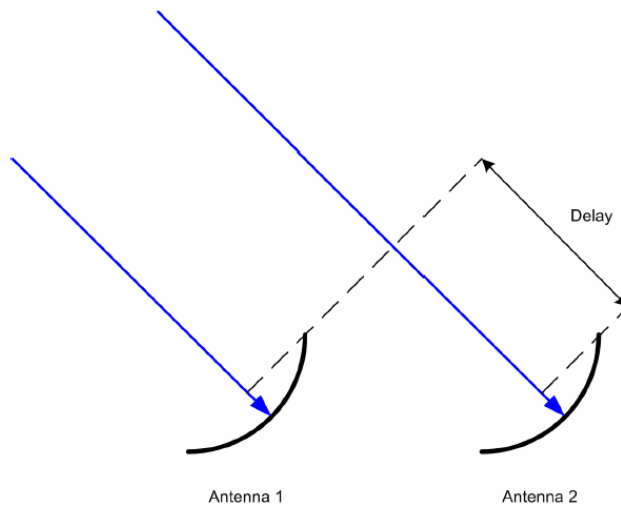
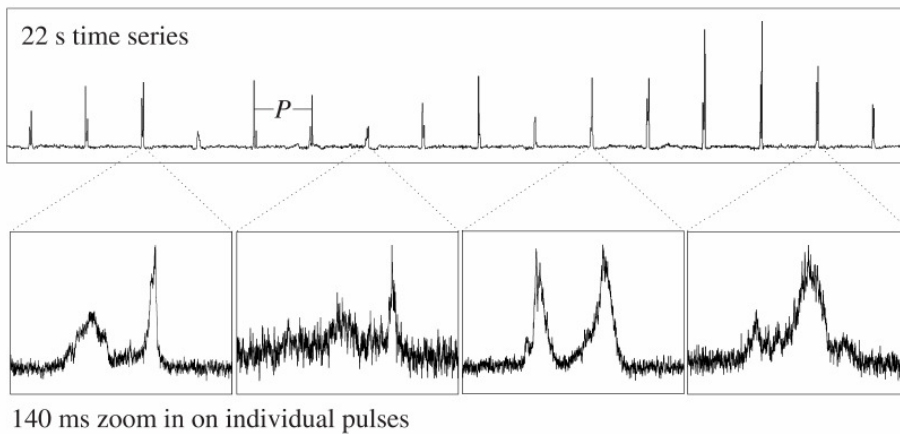
## 1.1 Program Overview

The program has three parts or aspects:

- Scientific Program
- Satellite Design and Construction
- Design and Construction Shuttlesystem

## 1.2 Scientific Program

### 1.2.1 Pulsar Radio Astronomy



### 1.2.2 The SRWDA Project

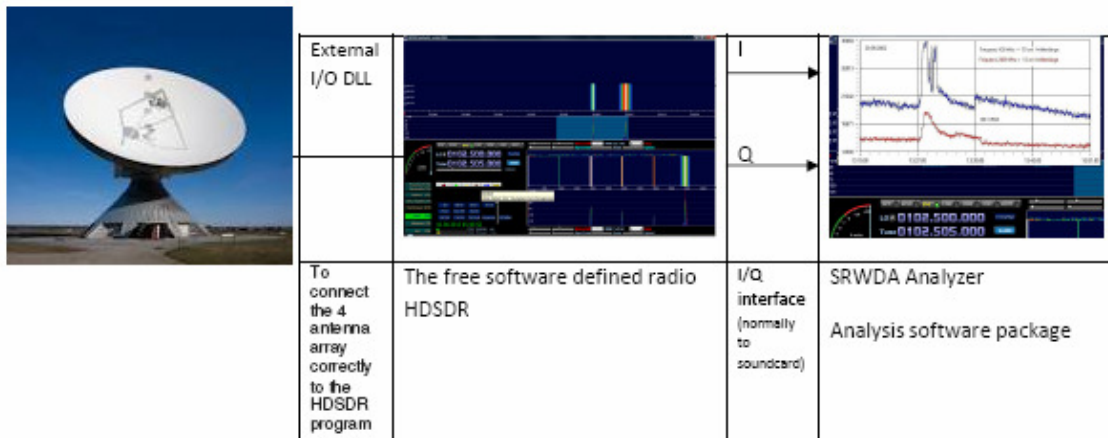
The radio astronomical IAP project supernova radio wave detector and analyzer (SRWDA) aims to detect and analyze HI radio signals from supernova remnants. At the base station a set of antennas, which gives also direction

information of the signals, is aimed to be connected to a computer which acts as Software Defined Radio (SDR). Afterwards an analyzing program is aimed to be installed.

Later the detectors are planned to be installed on satellites to improve the resolution and to suppress disturbing signals from earth stations.

### 1.2.3 Ground Station for Radio Astronomy: IAP\_SRWDA

Last update: 06 December 2013



#### Actual Working Packages:

- Development of Prototype with parts of IAP ECS

#### Financial and man power needs for 2014:

- 1 persons (e.g. as PhD thesis) with student support (e.g. as master thesis)
- 1.000 USD material

Actually a master thesis is undergone in this project [Kassar 2014].

### 1.3 Satellite Design and Construction

Die Architektur einer Satellitenmission

Der Entwurf einer Satellitenmission

Entwurf und Planung eines Satelliten

Entwurf des Energieversorgungssystems

Entwurf des Lageregelungssystems

Entwurf des Antriebssystems

Performance Evaluation of Propulsion Systems

Description of the Analysis Tool

Entwurf der Funkverbindung

Entwurf des Borddatenverarbeitungssystems

Entwurf des Thermalkontrollsystems

Entwurf der Struktur und Mechanismen

Zuverlässigkeit und Produktsicherung

## 1.4 Old Program (last updated Dez. 12)

Das IAP-Programm umfasst für den Zeitraum Jan 2013 – Dez 2015 folgende Einzelpakete:

- A. Programmpaket Mission (Start bis LEO – Übergang in Erdflucht – Einschwenkung in Mondorbit – Abtrennung des Landemoduls vom Mutterschiff – Landung des Landemoduls auf dem Mond – Start und Flug zum Mutterschiff im Mondorbit)
- B. Programmpaket wissenschaftliche Mission (Messen von kosmischer Strahlung, Vorauswertung, Senden zur Erde)

Programmpaket A:

- SW-Programmumgebung unter Linux
- Implementierung von PN Navigation und anderen regelungstechnischen Algorithmen (inkl. Lenkung), Benutzung von MATLAB/Simulink (Student Vers. bei Amazon 77 EUR)
- Implementierung von Image Processing (Bildsequenzen)
- Benutzung von CFD/Numerical Combustion SW
- Migrierung des alternative Lotte Systems zu einem Landemodul

Programmpaket B:

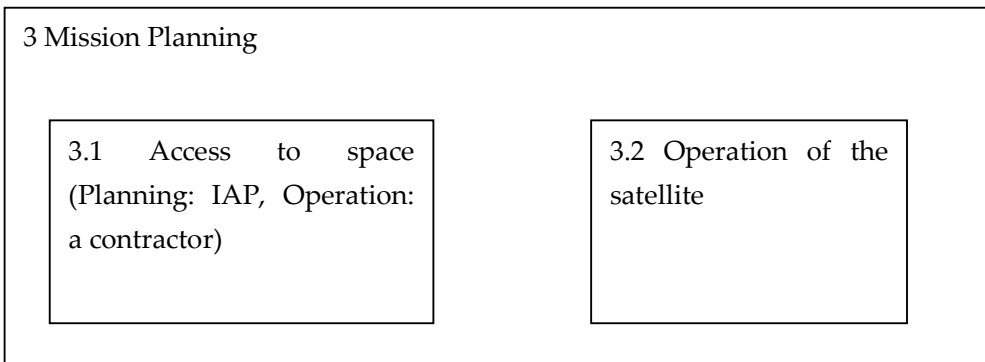
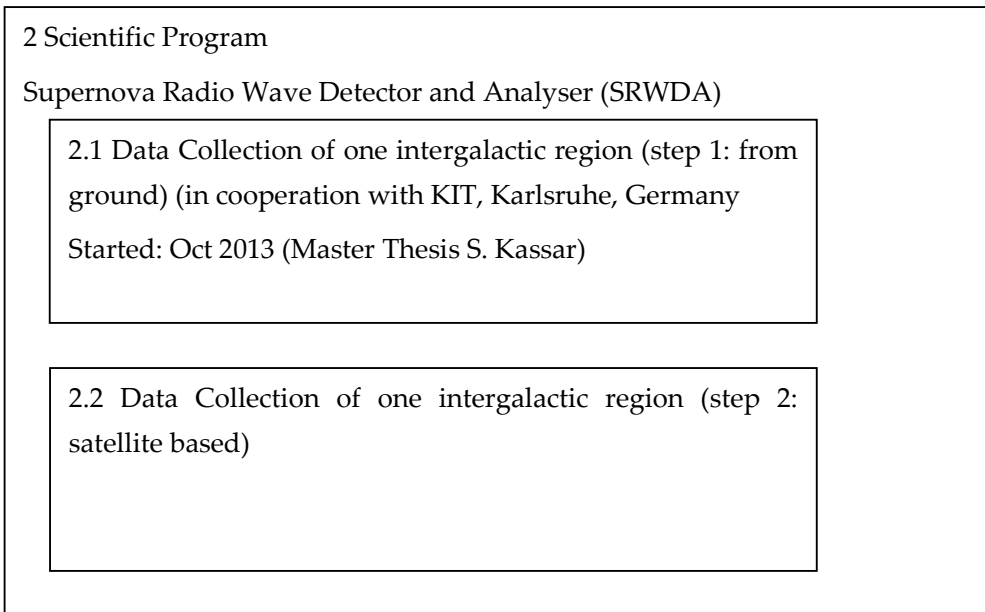
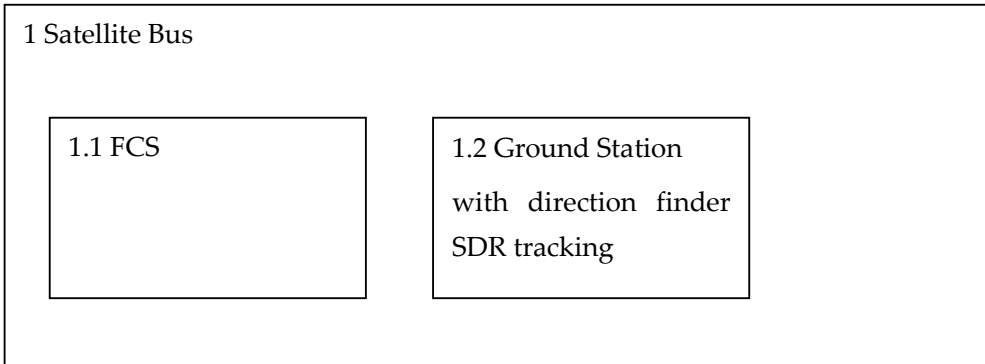
- SW-Programmumgebung unter Linux
- Wissenschaftliches teilchenphysikalisches Programmpaket
  
- MATLAB/SIMULINK Einarb. Satellitenregelung
- SIM-Programm Einarbeitung

## 1.5 Aim of IAP\_SRWDA-.SAT project

- initialization of a small private space program (mission planning, communication, telemetry (ground tracking), FCS) which is expendable to commercial usage
- initialization of astrophysics scientific program

## 2 Project Management

### 2.1 Organisation



### 2.2 Working Packages and Time Plan

#### 2.2.1 IAP 2013

June – Dec 13 : Conception Phase

Milestone (Dec 13): First Mock-up model

## 2.3 Costs

### 2.3.1 IAP 2012

July – September 2012

Mission Simulation

Costs: 500 \$

### 2.3.2 IAP 2013

2013	Materialkosten	Arbeiterlöhne	Einnahmen	Ausgaben/Mona	Bilanz/Monat	Bilanz/Jahr
Januar				0	0	
Februar				0	\$0	
März				0	\$0	
April				0	\$0	
Mai		-\$800		-800	-\$800	
Juni	-\$300	-\$800		-\$1.100	-\$1.100	
Juli		-\$960		-\$960	-\$960	
August				\$0	\$0	
September		-\$800		-\$800	-\$800	
Oktober	-\$150	-\$760		-\$910	-\$910	
November		-\$800		-\$800	-\$800	
Dezember		-\$480		-\$480	-\$480	
	-\$450	-\$5.400				-\$5.050

### 2.3.3 IAP 2014 (planned)

		Engineer need (MM)	Specialized Worker/Facharbeiter (MM)	Student (1/2 MM)	Personnel Costs	Material Costs	Duration (months)	Needed Staff
IAP_SRWDA-SAT	mock-up model				\$1.000	\$300	1	1 Student (A1)
	specification prototype				\$1.000		1	1 Student (A1)
	prototype (COM, FCS) aus ECS u. alt.Lotte übernehmen				\$4.000	\$1.200	6	1 Master Student (electrical engineering)
	mission simulation				\$2.000		1	1 Student (A1)
	Specification MPD propulsion system				\$1.000		3	1 Student (A1)
	MPD propulsion system				\$5.000	\$2.000		
IAP_SRWDA-SAT Ground Station	Migration from IAP_SRWDA Ground Station				\$3.000	\$1.000	1	1 Student (A1)
IAP_SRWDA Ground Station	Integration Ground Station Prototype					\$1.000	3	1 Master Student (electrical engineering) Suheeb

Sum: 22.500 \$

## 3 Basics

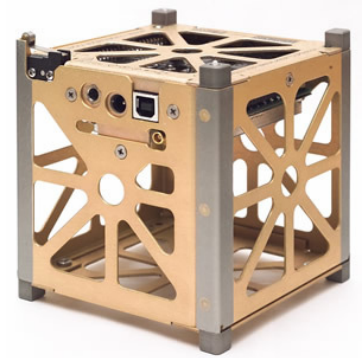
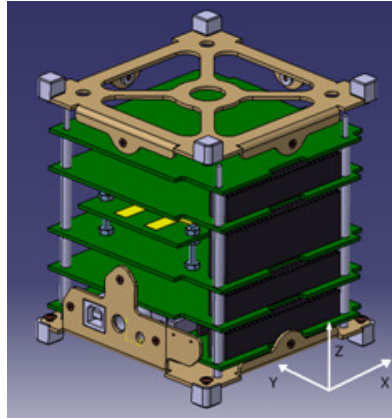
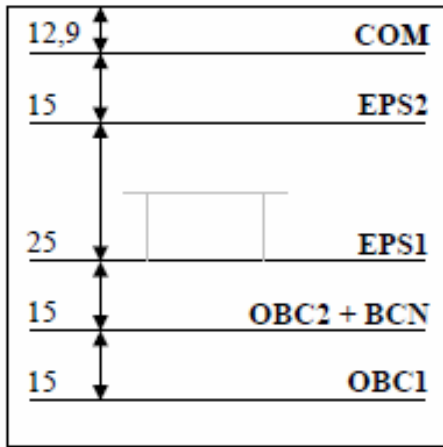
### 3.1 Satellite Bus Design

#### Satellitenentwurf

Kapitel	Thema
1	<a href="#">Die Architektur einer Satellitenmission</a>
2	<a href="#">Der Entwurf einer Satellitenmission</a>
3	<a href="#">Entwurf und Planung eines Satelliten</a>
4	<a href="#">Entwurf des Energieversorgungssystems</a>
5	<a href="#">Entwurf des Lageregelungssystems</a>
6	Entwurf des Antriebssystems
6.1	<a href="#">Performance Evaluation of Propulsion Systems</a>
6.2	<a href="#">Excercise: Propulsion System Selection</a>
6.3	<a href="#">Table for Excercise Results</a>
6.4	<a href="#">Description of the Analysis Tool</a>
7	<a href="#">Entwurf der Funkverbindung</a>
8	<a href="#">Entwurf des Borddatenverarbeitungssystems</a>
9	<a href="#">Entwurf des Thermalkontrollsystems</a>
10	<a href="#">Entwurf der Struktur und Mechanismen</a>
11	<a href="#">Zuverlässigkeit und Produktsicherung</a>

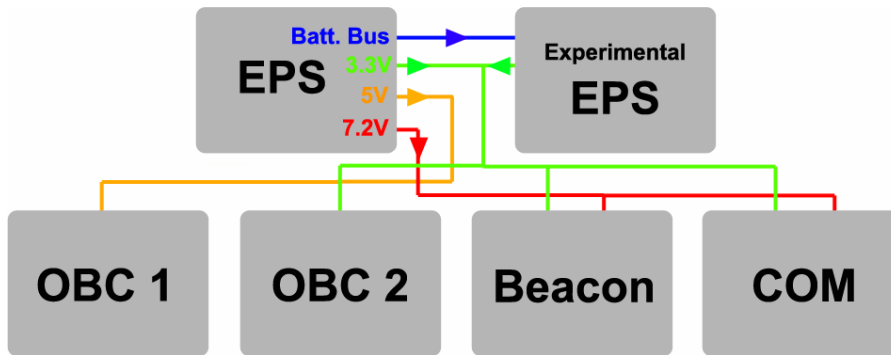
#### 3.1.1 Example: 10 ccm Student Nanosatellite OUFTI-1 of University of Liège

(OUFTI-1 is designed for communications in D-STAR, an amateur-radio communication protocol)

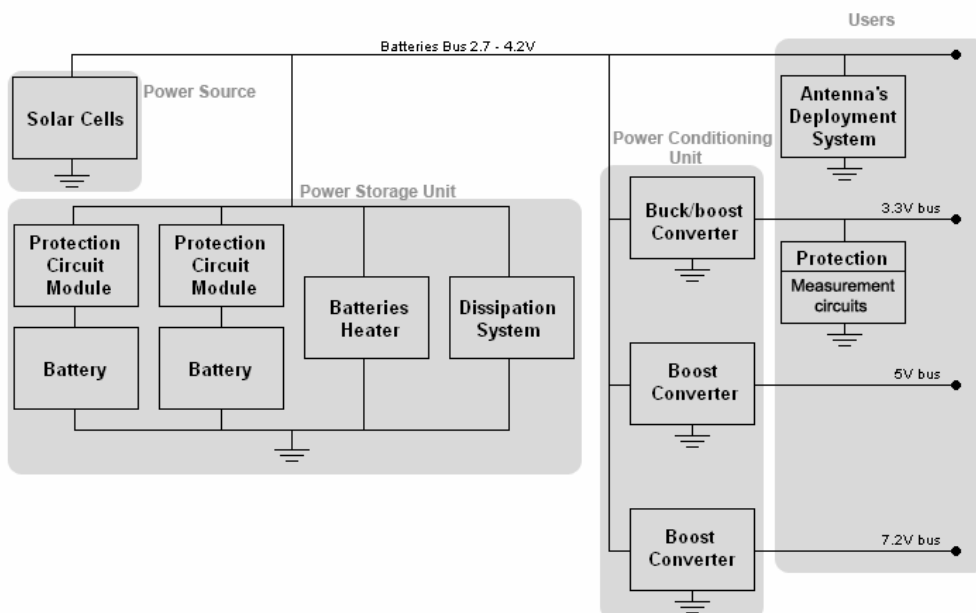


### 3.2 Entwurf des Energieversorgungssystems

base: Pierre THIRION, Design and Implementation of On-board Electrical Power Supply of Student Nanosatellite OUFTI-1 of University of Liège (Master Thesis)

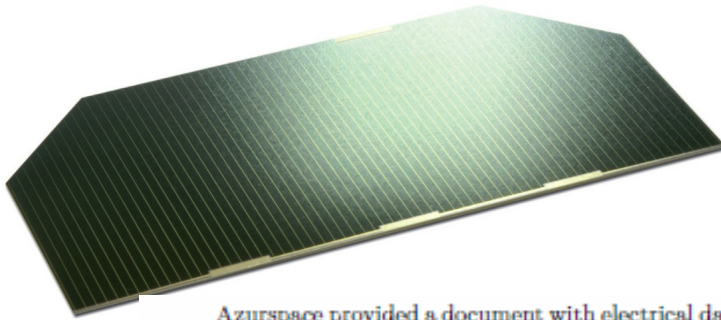


OUFTI-1  
subsystems  
and power  
buses



Block  
diagram of  
EPS.





The commercialized model, the “3G - 28%” , has an efficiency of 28% at Begin Of Life (BOL), the dimensions of the solar cells are 80mm x 40mm with cropped corners, for a total surface of 30.18cm<sup>2</sup>.

Azurspace provided a document with electrical data for the 30% cells (table 3.1). The short circuit current is thus 528.15 mA for one cell, and the current at max. power is 513.06mA.

Open Circuit Voltage $V_{OC}$ [mV]	2,716
Short Circuit Current $J_{SC}$ [mA/cm <sup>2</sup> ]	17.5
Voltage at max. Power $V_{pmax}$ [mV]	2,427
Current at max. Power $J_{pmax}$ [mA/cm <sup>2</sup> ]	17.0
Maximum Power $P_{pmax}$ [mW/cm <sup>2</sup> ]	41
Average Efficiency $\eta_{bare}$ [%]	30.1

Table 3.1: Electrical characteristics of Azurspace 3G - 30% solar cells at 28°C and for the AM0 spectrum [23].

### Electrical Design of EPS

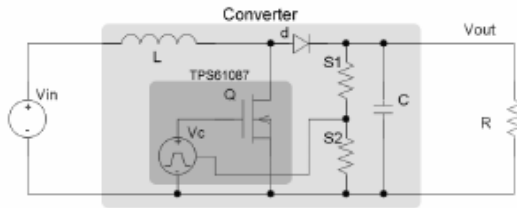


Figure 5.6: Simplified schematics of the 7.2V converter.

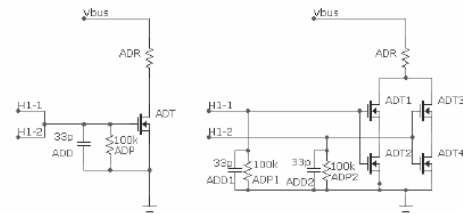


Figure 5.42: Propositions for the antennas deployment system.

## Matlab code for the I-V curves of solar panels

The first function gives the current in one solar cell as a function of the voltage (in V), the insolation G (with  $G_{nom} = 1350W/m^2$  as unity), and the temperature (in C).

function I = Cell\_GaAs(V,G,TaC)

%Code produced by the team from the CubeSat SwissCube

%and modified by Philippe Ledent (OUFTI-1) (september 2008)

% Pierre Thirion (OUFTI-1) (april 2009)

%

%Model of 30% efficiency solar cell from Azurspace : I=f(V,T)

%Use of function : I = Cell\_GaAs(V,G,TaC)

%V = Voltage on cell terminals [V]

%G = relative insolation [-] (G=1 => 1367 W/m<sup>2</sup>)

%TaC = temperature of the cell in operation [Celsius]

%Boltzman constant

k = 1.38e-23;

%Electric charge

q = 1.60e-19;

.....

.....

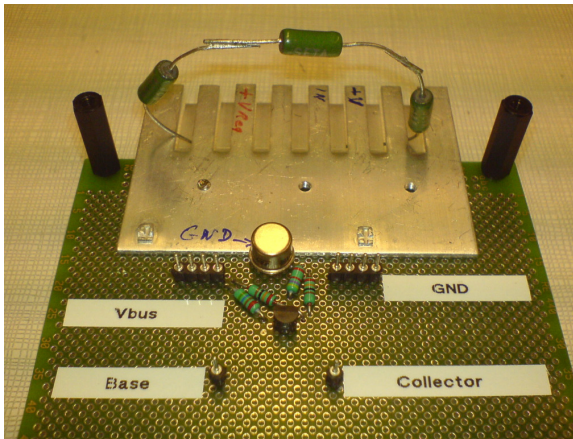


Figure C.1: Prototype of the shunt regulator.

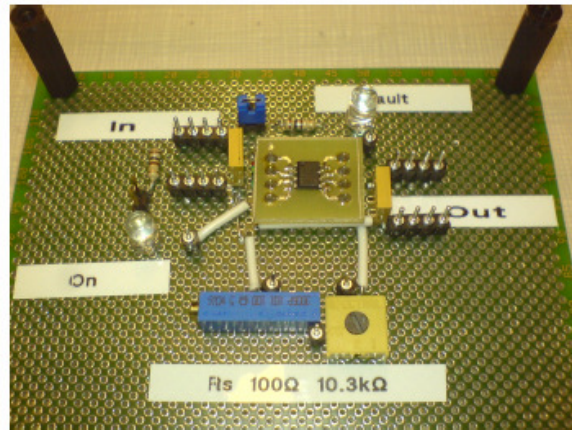


Figure C.2: Prototype of the protection circuit.

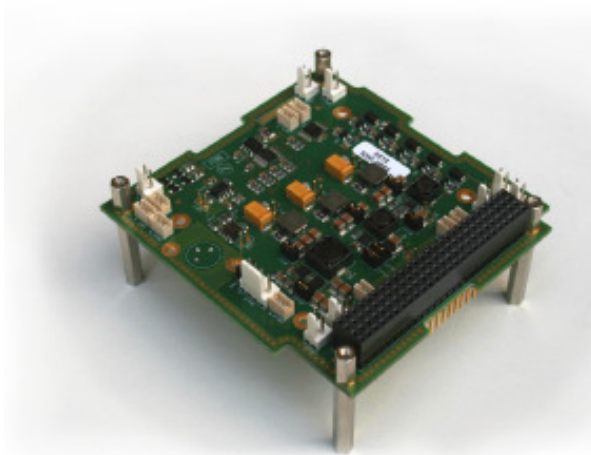
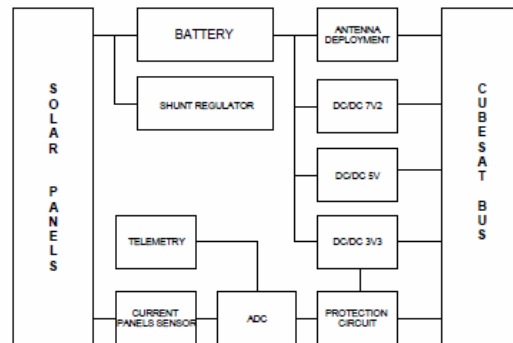


Figure C.3: Engineering model of the EPS card.

DESIGN TOP LEVEL :



### 3.3 Entwurf des Lageregelungssystems

#### 3.3.1 Sensors

Rate gyros

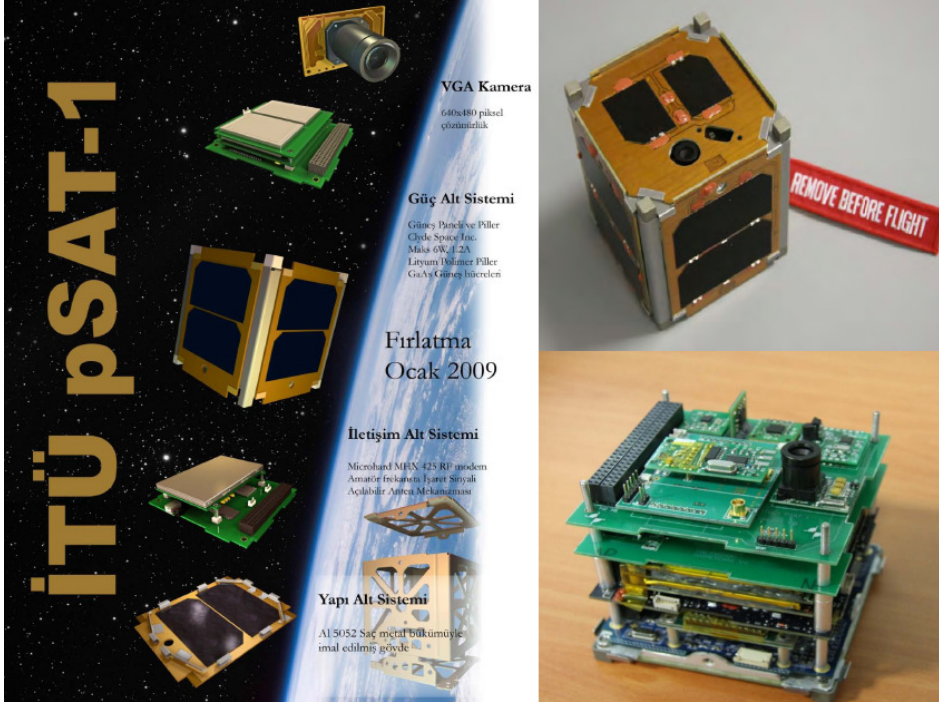
Star tracker camera

### 3.3.2 Actuators

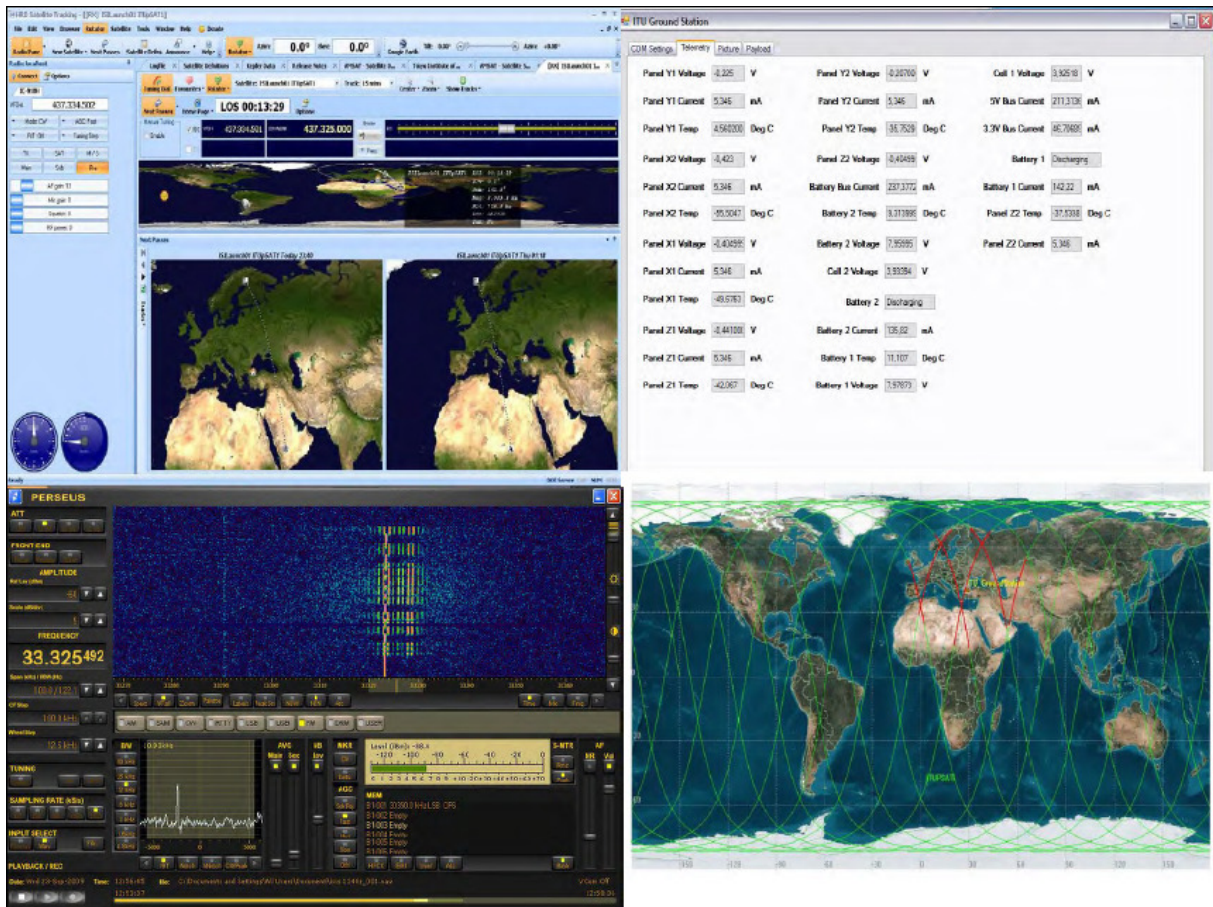
MPD Electrical thrust

## 3.4 Mission Simulation

From [Inalhan 2009]:







### 3.5 Shuttle System



Integration des Engineering Modells der Oberstufe

## 4 Candidate Software Tools

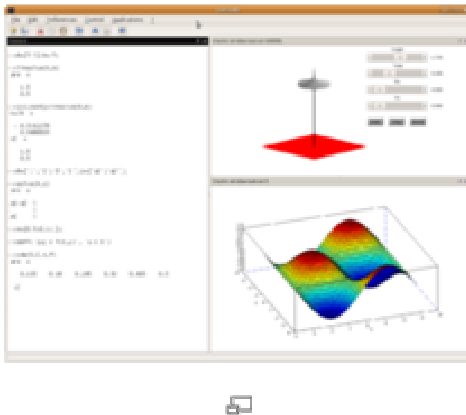
### 4.1 Possible Development Environment for Communication and Control

Communication and Control shall be developed under Matlab/Simulink or Octave or Scilab/Scicos. Then there must be generated autocode, which has to be put on a suitable HW.

#### 4.1.1 Scilab

aus Wikipedia, der freien Enzyklopädie

#### Scilab



Bildschirmfoto von Scilab

#### Basisdaten

<b><u>Entwickler</u></b>	<u>Scilab Enterprises</u>
<b><u>Aktuelle Version</u></b>	5.4.0
<b><u>Betriebssystem</u></b>	<u>Linux, Unix, Mac OS, BSD, Windows</u>
<b><u>Kategorie</u></b>	<u>Numerische Mathematik</u>
<b><u>Lizenz</u></b>	<u>CeCILL</u>
<b><u>Deutschsprachig</u></b>	deutsche Version verfügbar

[www.scilab.org](http://www.scilab.org)

Scilab ist ein umfangreiches, leistungsfähiges und freies Software-Paket für Anwendungen aus der numerischen Mathematik, das ehemals am Institut national de recherche en informatique et en

automatique (INRIA) in Frankreich seit 1990 als Alternative zu MATLAB entwickelt wurde und seit 2003 vom Scilab-Konsortium weiterentwickelt wird. Seit Juli 2008 ist das Scilab-Konsortium Mitglied der Digiteo Foundation. Seit Juli 2012 erfolgt die Herausgabe und Entwicklung durch Scilab Enterprises.

Die Funktionalität und Syntax von Scilab/Scicos ist zu weiten Teilen mit der von MATLAB/Simulink identisch, und es gibt Konverter von MATLAB nach Scilab. Scilab wird u. a. für technische und wissenschaftliche Anwendungen in Lehre, Forschung und Industrie eingesetzt. Scilab stellt von sich aus u. a. Funktionen für folgende Bereiche bereit: 2D- und 3D-Plots aller gängigen Formen auf der Grundlage von gnuplot (oder/und LabPlot), numerische lineare Algebra, Polynom-Berechnungen, Statistik, Regelungs-technik, digitale Signalverarbeitung, sowie I/O-Funktionen zum Lesen und Schreiben von Daten, unter anderem auch als Sounddateien im WAVE-Dateiformat. Darüber hinaus existiert eine Vielzahl fertiger Scilab-/Scicos-Lösungsskripte und Funktionsbibliotheken von Anwendern aus aller Welt.

Implementiert ist Scilab/Scicos in C, erweiterbar ist es aber auch durch Module, die in Scilab/Scicos selbst oder in anderen Sprachen verfasst wurden, z. B. FORTRAN oder C, für die definierte Schnittstellen existieren. Es liefert definierte Schnittstellen zu bereits verfügbaren oder selbst frei programmierbaren Toolkits (Toolboxen) wie Scicos, SPI, LabVIEW, Modelica, aber auch MuPAD. Mit Erscheinen der Version 5.0 wurden viele graphische Funktionalitäten sowie die Benutzeroberfläche in Java neu programmiert.

### **Abgrenzung**

Aufgrund der kaum vorhandenen algebraischen Funktionalitäten gehört Scilab nicht zu den Computeralgebrasystem (CAS) wie zum Beispiel Maxima, Maple oder Mathematica, die, im Unterschied zur rein numerischen Mathematik, auch symbolische Verfahren unterstützen.

#### **4.1.1.1 Zusatzpakete**

Im Lieferumfang von Scilab sind mächtige zusätzliche Werkzeuge verfügbar, so unter anderem:

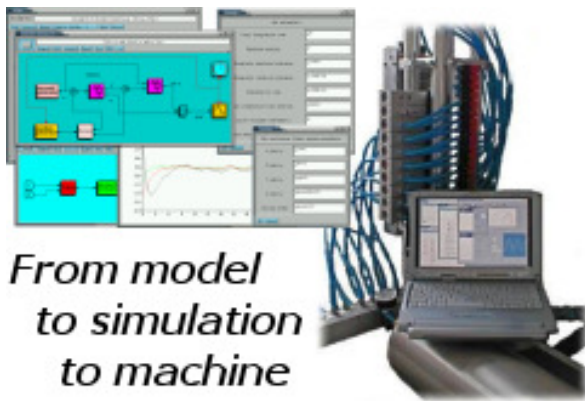
- **SciNotes**, ein speziell für Scilab entwickelter Texteditor und Debugger, der das Bearbeiten von Scilab-Skripten vereinfacht. Er unterstützt unter anderem:
  - Hervorheben der Scilab-Syntax
  - Automatische Identifizierung von Funktionen
  - Autovervollständigen
  - Direktes Ausführen von Skripten in Scilab
  - Registernavigation
  - Kommentarfunktion von Zeilen und Absätzen
  - Zeilennummerierung
  - Hilfefunktion

- **Xcos** (ehemals Scicos), ein Programmpaket ähnlich Simulink, zur graphischen Modellierung und Simulation dynamischer Systeme. Xcos 1.0 basiert auf Scicos 4.2 und ist zu diesem vollständig kompatibel <sup>[1]</sup>. Xcos enthält eine Sammlung graphischer Icons. Diese lassen sich ähnlich einer elektronischen Schaltung miteinander verdrahten. Es wird dabei unterschieden zwischen Zeitinformationen (rote Leitungen) und den eigentlichen Signalen (blaue Leitungen). So erstellte Programme bzw. Blockschaltbilder lassen sich auch in ein C-Programm umwandeln. Mit einem entsprechenden C-Compiler lassen sich die Programme dann auf jede beliebige Zielplattform, zum Beispiel auch auf Mikrocontroller, übertragen. Die aktuelle Scicos-Version 4.4 (Dezember 2009) ist in der ebenfalls freien, auf Scilab 4.1.2 basierenden, Entwicklungsumgebung ScicosLab verfügbar. Es steht ein freier Compiler für Modelica-Modelle zur Verfügung, der es erlaubt, Modelica-Modelle direkt in Scilab / Xcos zu verwenden und zu bearbeiten <sup>[2][3]</sup>.
- **ATOMS** (engl: *AutomaTic mOdules Management for Scilab*) ermöglicht das automatische Installieren und Aktualisieren von weiteren zusätzlichen Programmpaketen.

#### 4.1.1.2 Scicos

From [www.scicos.org](http://www.scicos.org) :

Scicos: Block diagram modeler/simulator



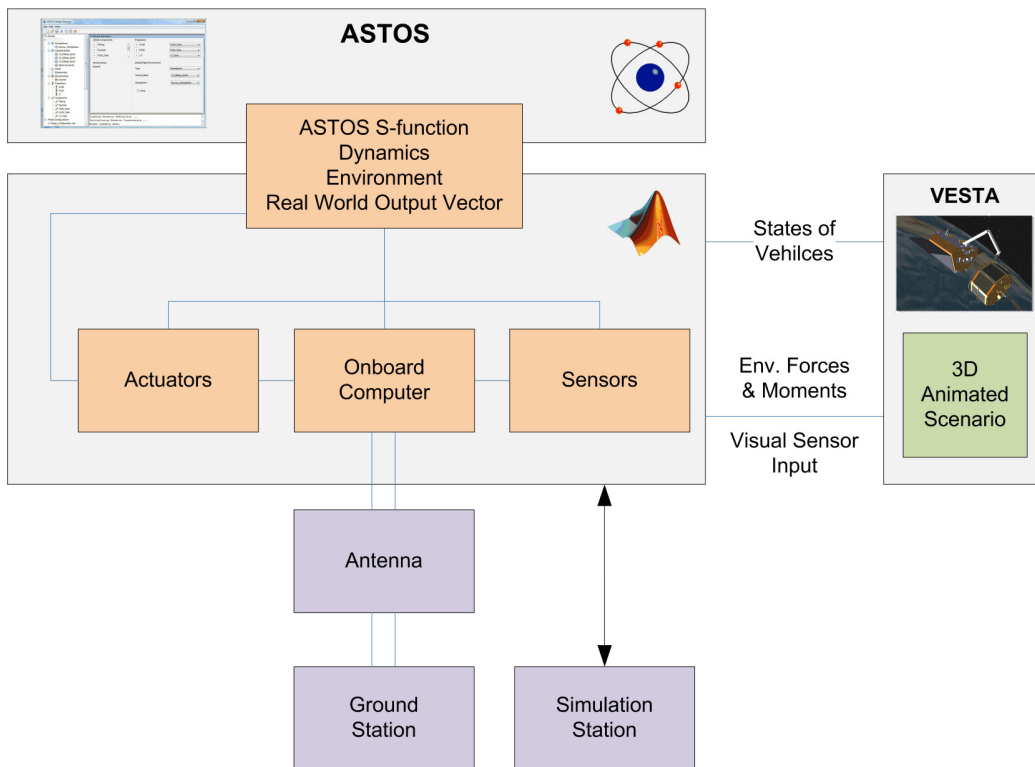
**Scicos** is a graphical dynamical system modeler and simulator developed in the Metalau project at INRIA, Paris-Rocquencourt center. With **Scicos**, user can create block diagrams to model and simulate the dynamics of hybrid dynamical systems and compile models into executable code. **Scicos** is used for signal processing, systems control, queuing systems, and to study physical and biological systems. New extensions allow generation of component based modeling of electrical and hydraulic circuits using the Modelica language.

**With Scicos you can:**

- Graphically model, compile, and simulate dynamical systems
- Combine continuous and discrete-time behaviors in the same model
- Select model elements from Palettes of standard blocks

- Program new blocks in C, Fortran, or Scilab Language
- Run simulations in batch mode from Scilab environment
- Generate C code from Scicos model using a *Code Generator*
- Run simulations in real time with and real devices using Scicos-HIL
- Generate *hard real-time* control executables with Scicos-RTAI and Scicos-FLEX
- Use implicit blocks developed in the Modelica language.
- Discover new Scicos capability using additional toolboxes.

## 4.2 NearSpacecraft Simulator Development



The tool SRSIM is a multi-purpose spacecraft simulation test bench, employed primarily to validate Guidance, Navigation and Control System designs for the various near space missions the IFR (University of Stuttgart) is involved in. The Matlab/Simulink based SRSIM is a six degree of freedom space dynamics simulator including advanced modeling of the space environment through the ASTOS software package. This allows to precisely include effects such as solar radiation pressure, atmospheric drag and magnetic field interactions. In addition, it includes an advance visualization environment called VESTA, that is also responsible for the calculations of any surface-related disturbances and provides input to optical sensors such as Navigation Cameras and Star Trackers.



## 5 Specification

[SysReq1] There shall be built a ground station receiver for radio astronomy at 21 cm.

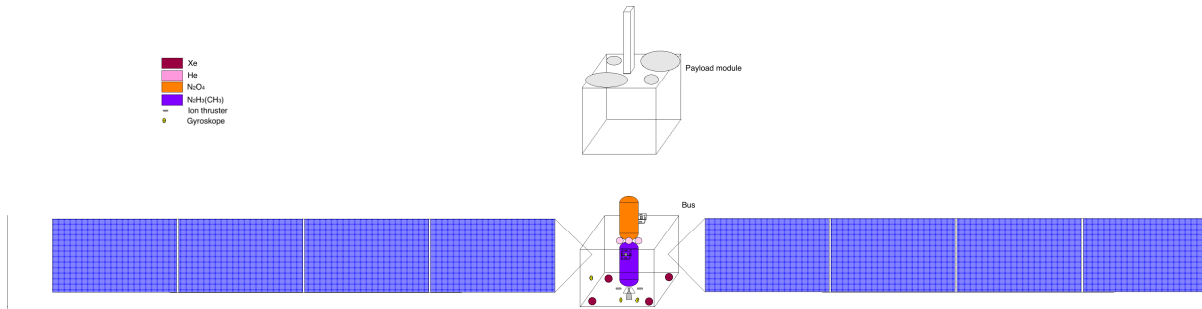
[SysReq1] There shall be built a satellite prototype.

[SysReq3] The satellite has a magnetoplasmadynamic (MPD) thruster. It shall be a *Self-Field MPD*.

## 6 System Concept

Many is taken from [Laufer 2009].

### 6.1 Overview



#### 6.1.1 Satellite Bus

The bus necessary for such a satellite consists of the following subsystems: propulsion system, power supply system, structures and mechanisms, thermal control, communication, command and data handling, attitude determination and control.

#### 6.1.2 Propulsion System

SIMP-LEX is a low mass, robust and easy-to-integrated system consisting of a cluster of four pulsed MPD thrusters using solid propellant (PTFE, Polytetrafluorethylene, a.k.a. Teflon). A pulsed plasma thruster consists of four main parts in general: a capacitor bank, two electrodes, the propellant and a spark plug (see figure 3.4). After the capacitors are charged, the spark plug is triggered forming a short-time arc discharge. This discharge ionizes the solid propellant along its surface closing the main circuit through a plasma sheet formed by the ablated and ionized propellant. The circuit yields a current loop resulting in an induced magnetic field. The magnetic field component perpendicular to the plasma current leads to a Lorentz force and accelerates the plasma creating an impulse. The MPD thrusters provides an average thrust of approx. 1.5 mN each with a specific impulse of 1950 s pulsed at 0.5 Hz and consuming around 120 W per engine. Using a pulsed plasma thruster system as the main engine during the cruise phase demands lifetime tests and investigations especially of the solid propellant feed system (see figure 3.5) since an expected operational lifetime of up to 20,000 hours is required. For this case, the system is designed to feed the propellant from the side using a helical shape with up to two loops on each side for the *LUNAR MISSION BW1* [60].

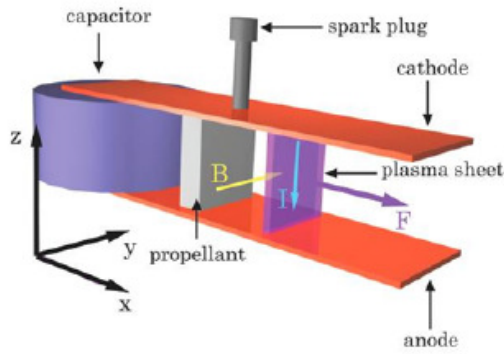


Figure 3.4: Working Principle of SIMP-LEX[60, 48]

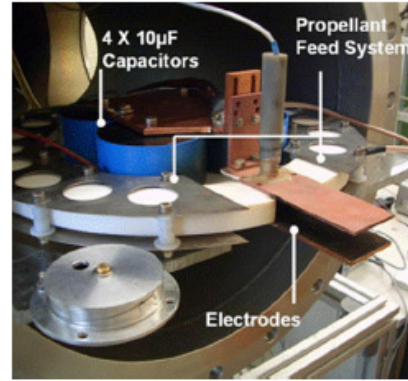


Figure 3.5: Test of SIMP-LEX propellant feed system[60, 48]

From [Nawaz et. al. 2005]:

### I. Introduction

At the *Institut für Raumfahrtssysteme* in Stuttgart, Germany, an instationary pulsed magnetoplasmadynamic thruster (I-MPD, also referred to as pulsed plasma thruster, PPT) is being built. This project was initiated when the *Small Satellite Program* was started at IRS in 2002. This program includes the launch of four satellites, developed at this institute. One of the satellites, BW1, is an all electrical satellite bound to the moon<sup>[1]</sup>. Two types of electric thrusters will provide the  $\Delta v$  necessary. One of them is a thermal arcjet with a thrust level of around 100 mN<sup>[2]</sup>; the other is the I-MPD thruster SIMP-LEX (Stuttgart Instationary Magnetoplasmadynamic Thruster for Lunar Exploration). The first will be used for phases that require higher

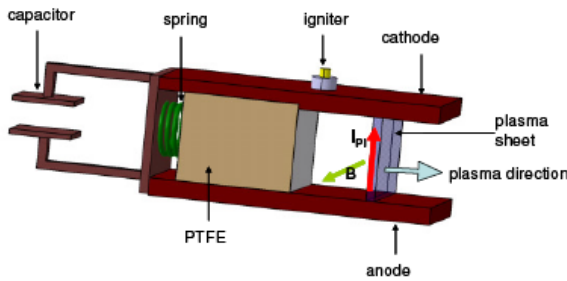


Figure 1. Working Principle of an I-MPD

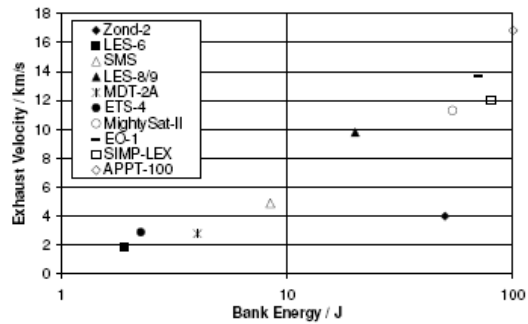


Figure 2. Overview of Exhaust Velocity over Bank Energy for Various I-MPDs<sup>[3,4,5,6]</sup>

thrust levels, such as the ascent phase immediately after separation from launcher, elevating the orbit above the Van Allen belt. A cluster of I-MPDs serve as cruise propulsion system. The I-MPD was chosen as cruise thruster because of its simple design, its reliability as well as its capability to easily adjust to different power levels due to its pulsed energy release without loss of performance. On small satellites, power is usually one of the tightest constraints. Although SIMP-LEX is the first I-MPD thruster in Stuttgart, the development of such a thruster at IRS can draw from over 30 years of expertise in the field of electric propulsion.

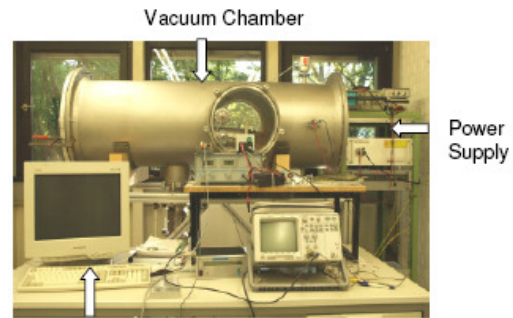
In principle, a parallel plate I-MPD consists of two parallel electrode plates between which a solid propellant block (Polytetrafluorethylen, PTFE) is placed, see Fig.1. The capacitor is charged externally and discharges across the surface of the PTFE, forming PTFE-plasma, once the igniter is fired. An electrical circuit with current  $I_p$  and its own magnetic field  $B$  is formed. Consequently, the plasma is accelerated outwards along the electrodes according to the Lorentz law resulting in an impulse bit on the thruster in the opposite direction.

Figure 2 shows an overview of several I-MPD thrusters. Filled icons indicate the mission was already launched. SIMP-LEX will have a bank energy of around 80 J and its expected exhaust velocity is 12 000 m/s. In order to serve as a cruise propulsion system the cluster of I-MPDs will have to provide a total average thrust of 6 mN.

### A. Test Facility Setup

In its final stage, the test facility for the I-MPD will be comprised of three vacuum chambers: one in which the thrust is measured, one for conducting life expectancy tests, and one for optimizing components and geometry of the thruster. Currently, the first two have been setup, where the facility in which the thrust is measured also serves for geometry and current investigations. It is shown in Fig. 3. The pressure in the vacuum chambers reaches about  $10^{-5}$  mbar prior to firing the thruster.

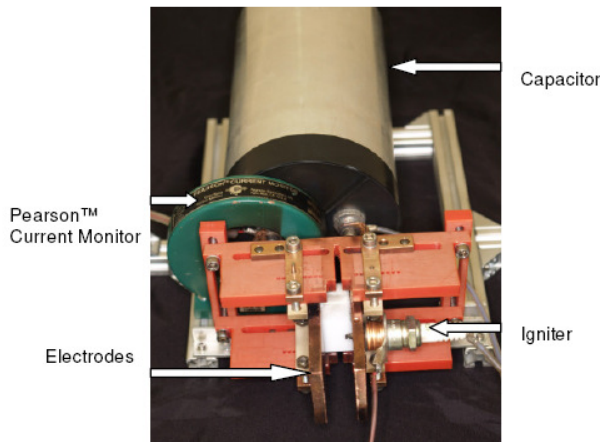
The electronics providing the voltage to the capacitor as well as the voltage peak to the spark plug was developed by ASP GmbH in Friedrichshafen. Its bread board stage allows for different voltage settings for charging the capacitor as well as various pulse frequencies in the automatic firing mode.



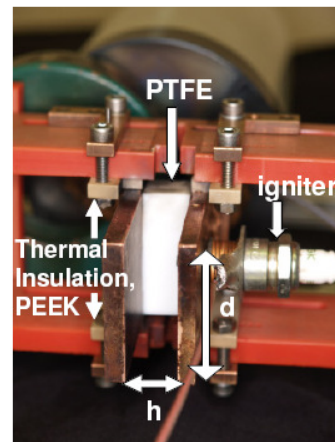
Data Readout at 200MHz  
**Figure 3. One of the I-MPD Test Facilities**

### B. Thruster Setup

The current setup of the I-MPD thruster is shown in the pictures in Fig. 4. It shows its parallel plate geometry, as well as its highly modular setup. This setup allows for easy changes of components and geometry which are currently being optimized. Fig. 4 a) shows the configuration of the capacitor, the copper electrodes and the igniter. The distance  $h$  between the electrodes in Fig. 4 b) can be changed incrementally by using a different set of holes in the red half shell for mounting the electrode plates, as seen in the left picture. The width  $d$  of the electrodes as well as their length can be varied freely. The current values for the thruster can be seen in



**Figure 4a. Current Setup of SIMP-LEX**



**Figure 4b. Close-Up Front View of SIMP-LEX**

Table 1. However, they are still subject to optimization. L is the distance the plasma covers from the propellant surface to the end of the electrodes. First tests at a pulse frequency of 1.2 Hz showed that it is necessary to insulate the PE (Polyethylen) half shells (red) thermally from the plasma and the copper electrodes. This was realized by placing PEEK spaceholders between the electrodes and the half shell and by moving the half shell backwards, away from the plasma.

Capacity	1 x 40 $\mu$ F
Maximum Voltage	2200V
h	20 mm
d	40 mm
L	47 mm

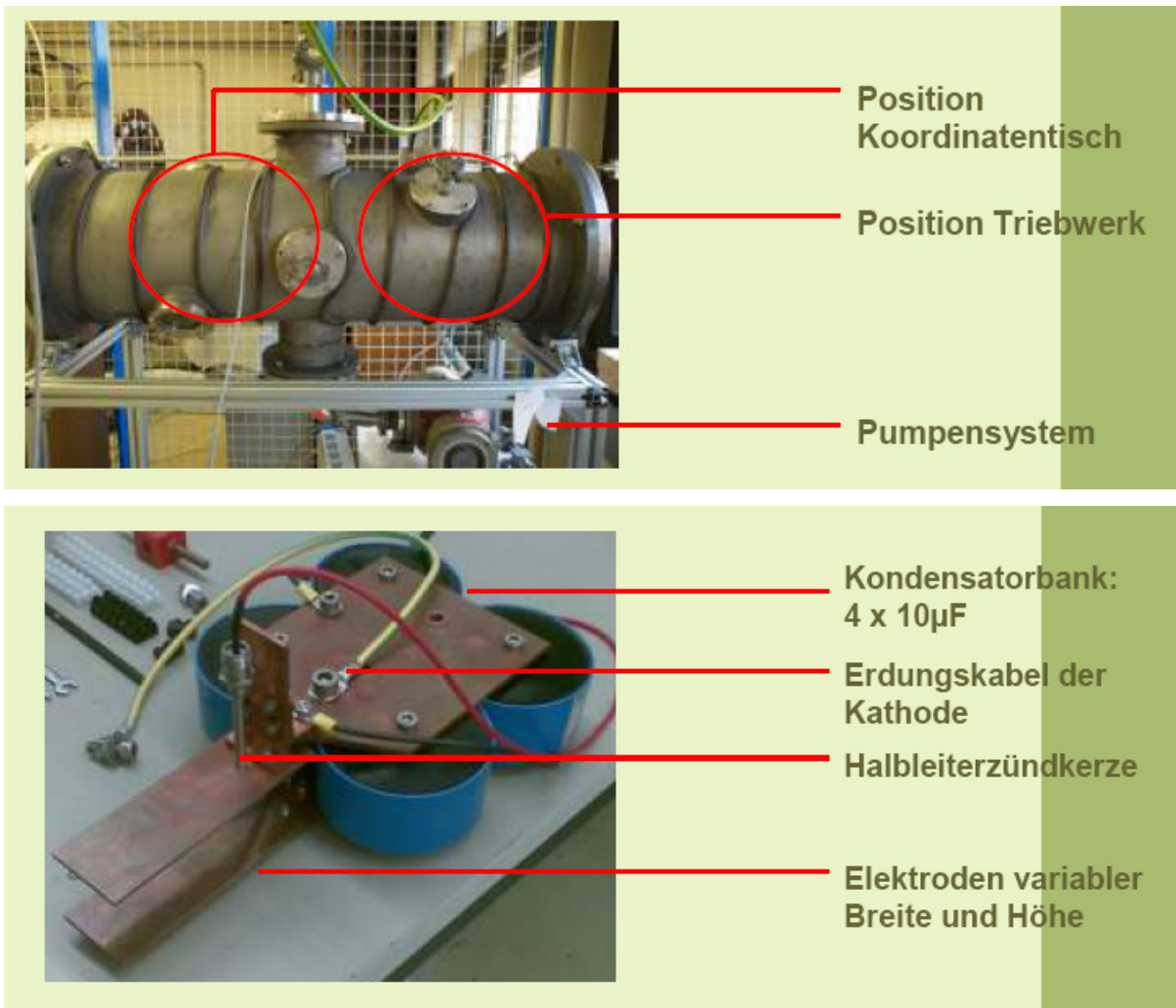
**Table 1. Current I-MPD Design Parameters**

**C. Measurement of Plasma Current and Mass Bit Ablated**

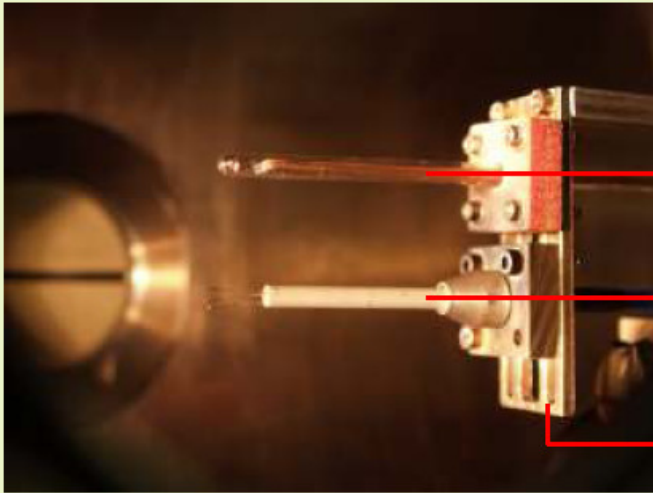
Figure 4 also shows the Pearson™ current monitor, placed between the electrodes and the capacitor. It serves as a Rogowski coil and outputs a voltage proportional to the current through the coil.

The mass bit is measured by comparing the weight of the propellant bar before and after at least 150 pulses. The balance used has a precision of 10  $\mu$ g.

From [Böhrk et. al. 2007]:



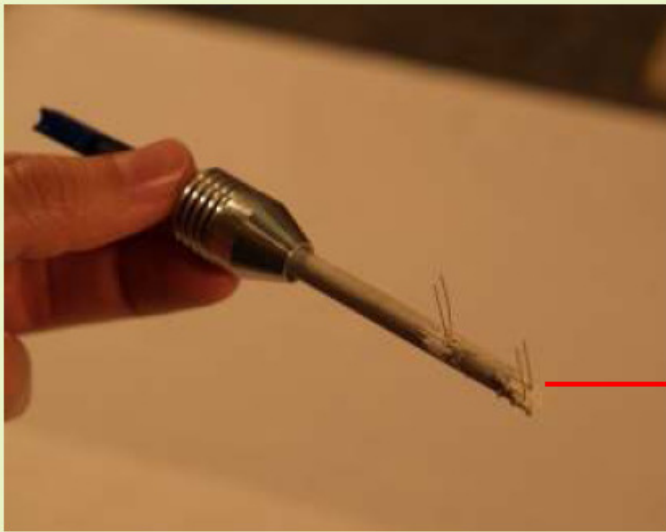




Magnetfeldsonde

Elektrostatische  
Tripelsonde

Sondenhalterung auf  
Koordinatentisch



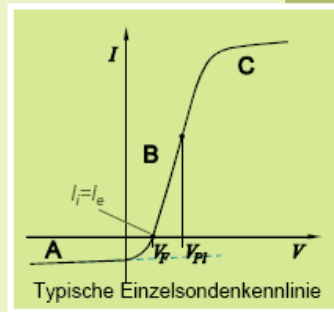
Elektrostatische  
Flugzeitsonde

$$\lambda_e \approx \lambda_i \approx \lambda_{ei} = \frac{16\pi(\epsilon_0 k T_e)^2}{e^4 n_e \ln \Lambda}$$

$$\Lambda = 12\pi \frac{(\epsilon_0 k T_e)^{3/2}}{n_e^{1/2} e^3}$$

$$\lambda_D = \sqrt{\frac{\epsilon_0 k T_e}{e^2 n_e}}$$

$$r_L = \frac{mv_{\perp}}{ZeB}$$

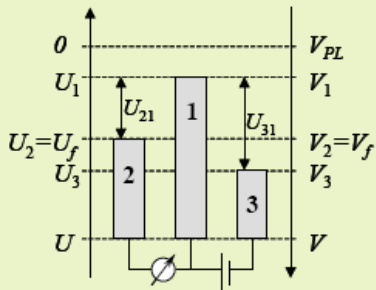


$r_p$	$0,2 \times 10^{-3} \text{m}$	$s$	$2 \times 10^{-3} \text{m}$
$\lambda_D$	$3 \times 10^{-6} \text{m}$	$\lambda_{e,i}$	$1,3 \times 10^{-2} \text{m}$
$m_i$	$5,14 \times 10^{-26} \text{kg}$ [5]	$r_{L,i}$	$3 \times 10^{-2} \text{m}$
		$r_{L,e}$	$1 \times 10^{-6} \text{m}$

- Plasma des I-MPD: dünne ( $r_p > \lambda_D$ ) stoßfreie ( $\lambda_{e,i} > r_p$ ) Raumladungsschicht
- $r_{L,e} < r_p < r_{L,i}$  moderates B-Feld
- Elektronenenergien maxwellverteilt

[5] Scharlemann, C., Investigation of Thrust Mechanisms in a Water fed Pulsed Plasma Thruster, Ph.D.-Thesis, Ohio State University, 2003

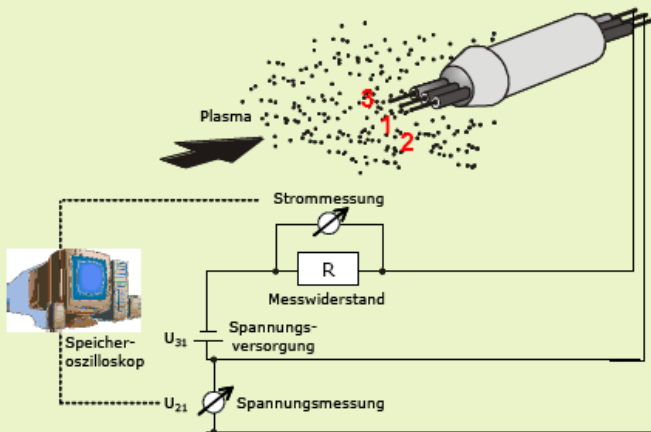
### Elektrostatische Tripelsonde



$$I_1 = A j_i(U_1) - A j_e \exp\left(-\frac{eU_1}{kT_e}\right)$$

$$I_2 = 0 = A j_i(U_2) - A j_e \exp\left(-\frac{eU_2}{kT_e}\right)$$

$$I_3 = I_1 = A j_i(U_3) - A j_e \exp\left(-\frac{eU_3}{kT_e}\right)$$



## Elektrostatische Tripelsonde

$$I_1 = A_{j_i}(U_1) - A_{j_e} \exp\left(-\frac{eU_1}{kT_e}\right)$$

$$I_2 \neq 0 = A_{j_i}(U_2) - A_{j_e} \exp\left(-\frac{eU_2}{kT_e}\right)$$

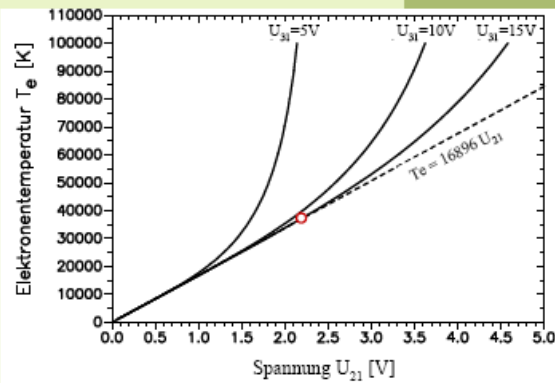
$$I_3 = I_1 = A_{j_i}(U_3) - A_{j_e} \exp\left(-\frac{eU_3}{kT_e}\right)$$

$$\frac{I_1 - I_2}{I_1 - I_3} = \frac{1}{2} = \frac{1 - \exp(-eU_{21}/kT_e)}{1 - \exp(-eU_{31}/kT_e)}$$

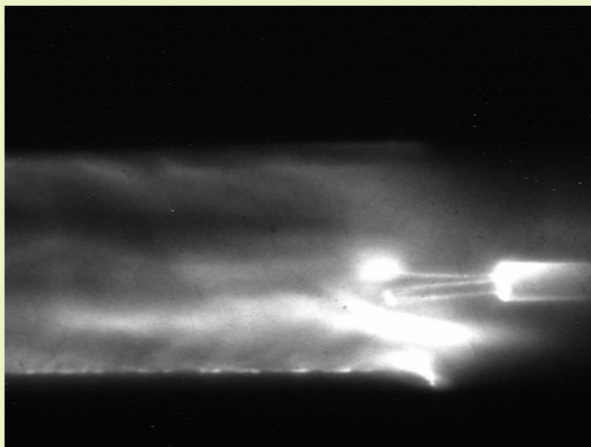
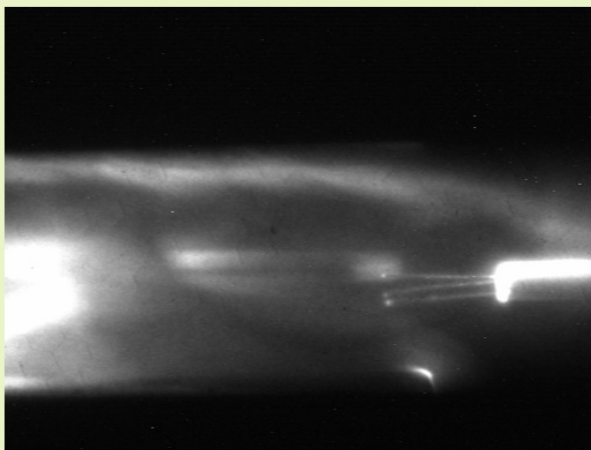
$$U_{21} = \frac{kT_e}{e} \left[ \ln 2 - \ln \left( 1 + \exp\left(-\frac{eU_{31}}{kT_e}\right) \right) \right]$$

Der Wert der Exponentialfunktion wird für relativ große Doppelsondenspannungen ( $U_{31} \geq 10V$ ) vernachlässigbar klein. Somit gilt in guter Näherung :

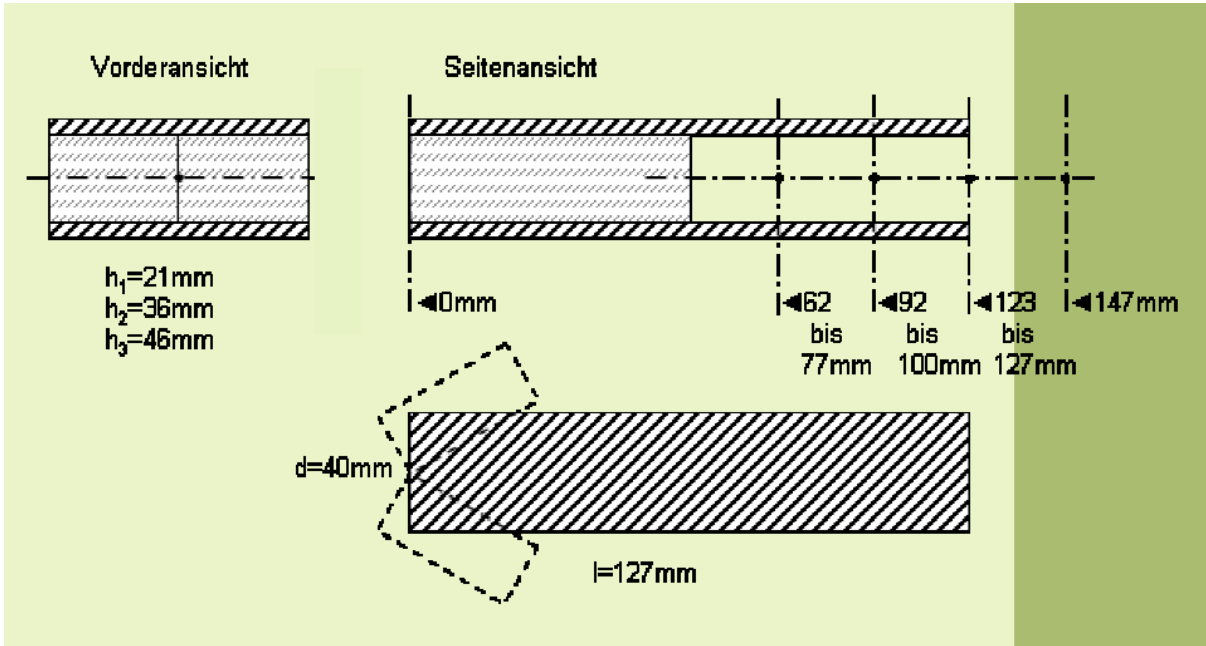
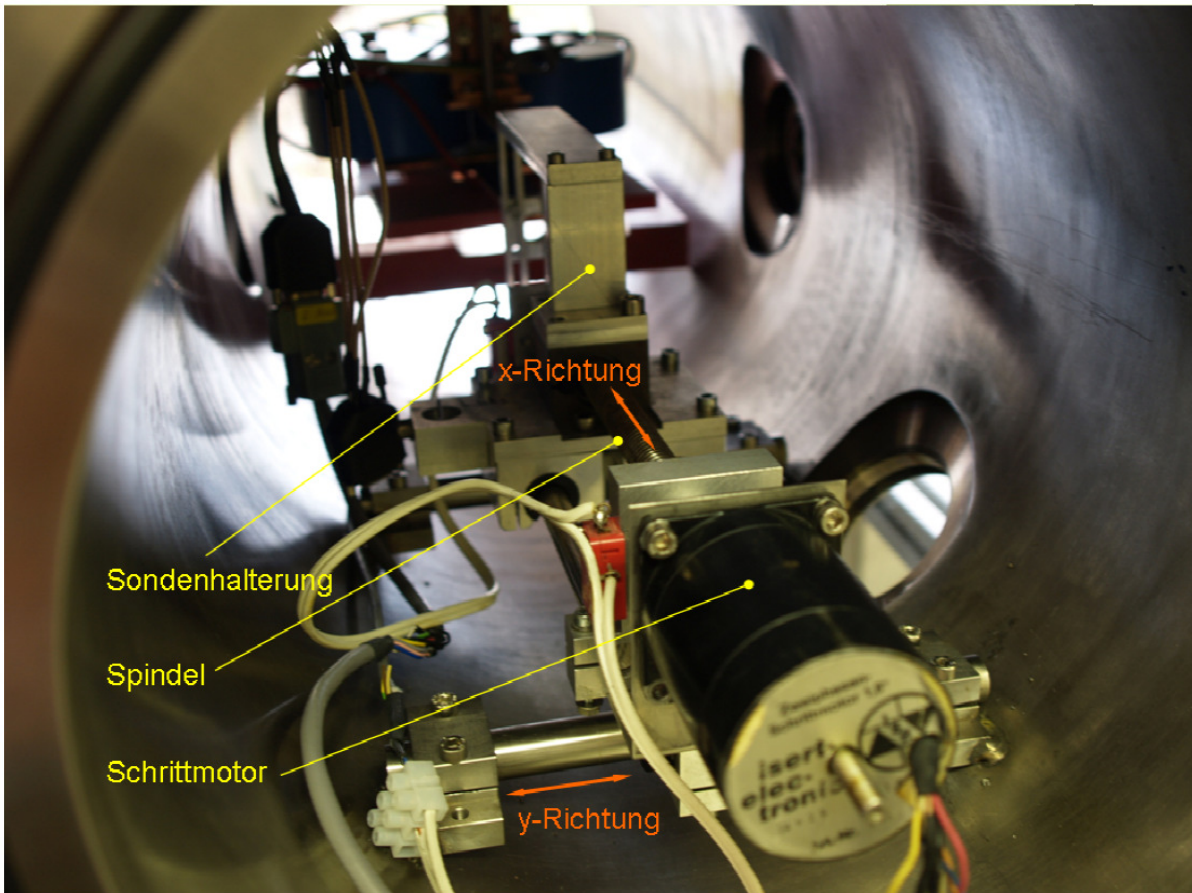
$$T_e = \frac{eU_{21}}{k \ln 2}$$



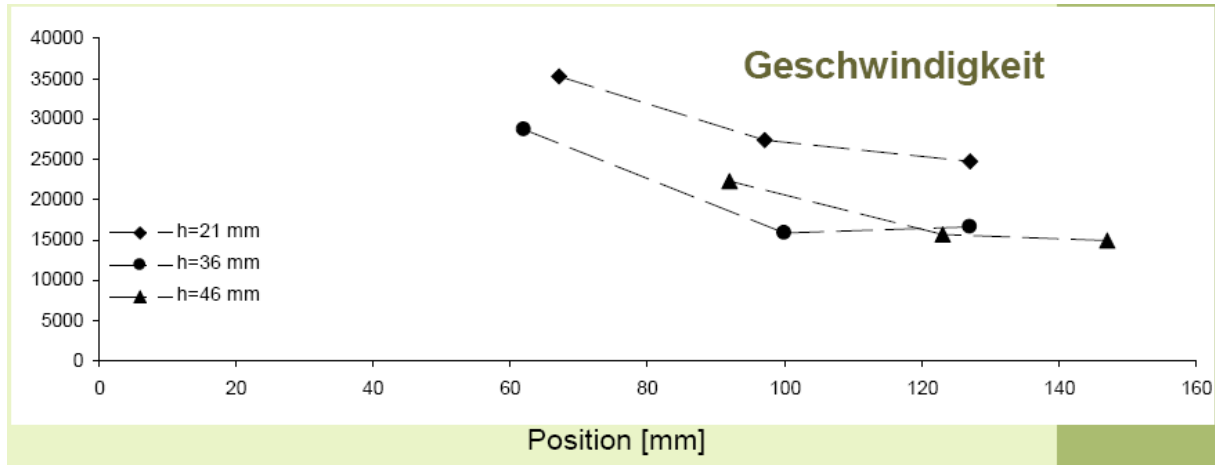
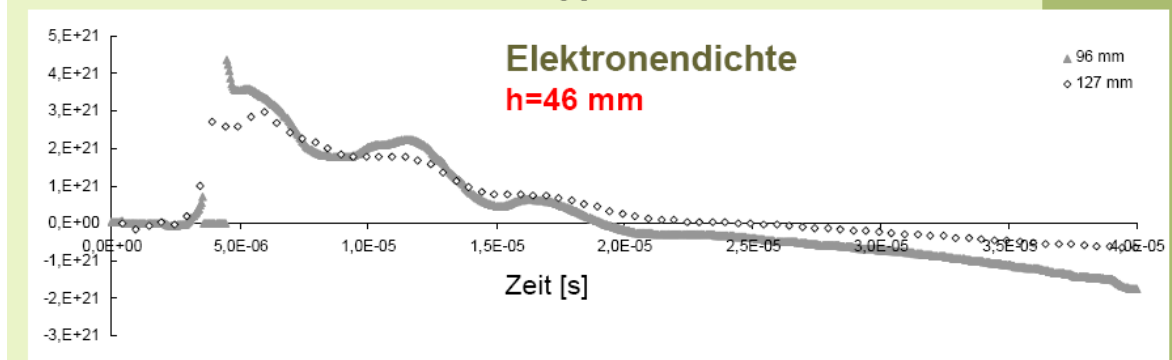
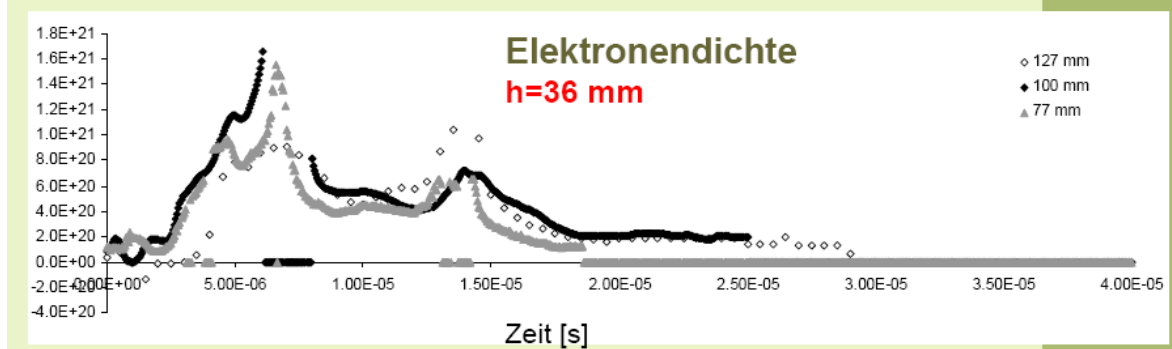
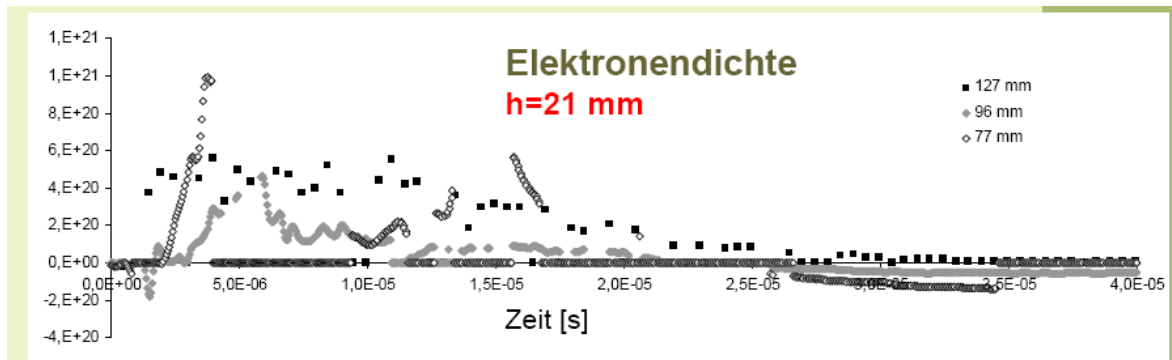
Elektronentemperatur  $T_e$  in Abhängigkeit von  $U_{21}$  und  $U_{31}$

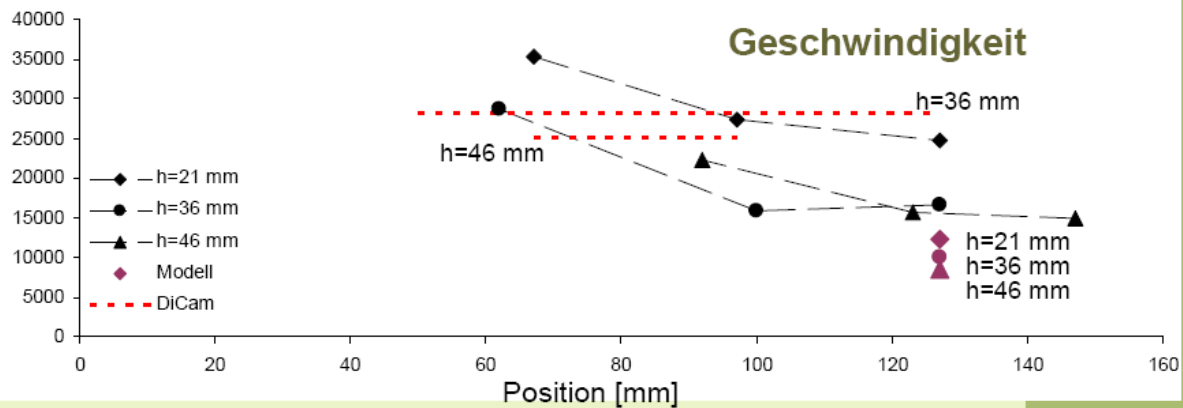
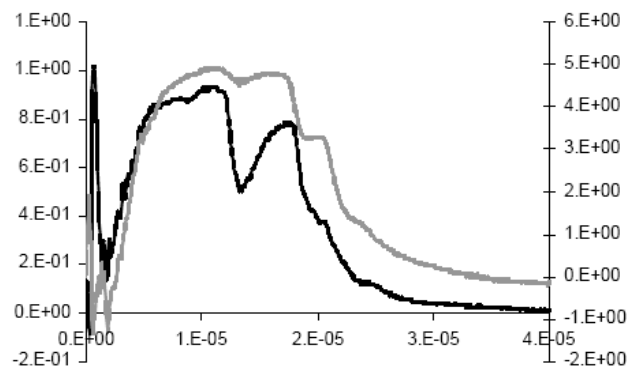












**Elektrostatische Sondenmessungen am I-MPD Simp-LEX erfolgreich durchgeführt**

**Messungen zwischen den Elektroden**

**Vergleichbarkeit mit Ergebnissen aus „Messungen“ mit Hochgeschwindigkeitskamera**

**Höchste Elektronentemperatur bei h=21 mm:  $T_e=20-30\text{eV}$**

**Höchste Elektronendichte bei h=46 mm:  $n_e=4 \times 10^{21}\text{m}^{-3}$**

**Höchste Geschwindigkeit bei h=21 mm:  $v=25-35\text{ km/s}$**

### 6.1.3 Power Supply System

#### The Li-ion advantage

Today's satellite makers are looking for technologies that reduce the overall size of the satellite because of the costs of manufacturing and launching the satellite. A lighter, smaller battery allows more mass/volume on the satellite for more transponders (communication satellites), heavier science components (NASA missions) or lower launch costs. Although Ni-H<sub>2</sub> significantly reduces satellite weight compared to NiCd, lithium-ion (Li-ion) technology reduces weight even more due to its higher specific energy.

Lower thermal dissipation is another source of weight reduction for Li-ion. Combined with higher energy and better charge efficiency, this lets the satellite maker reduce the size of the satellites' solar panels.

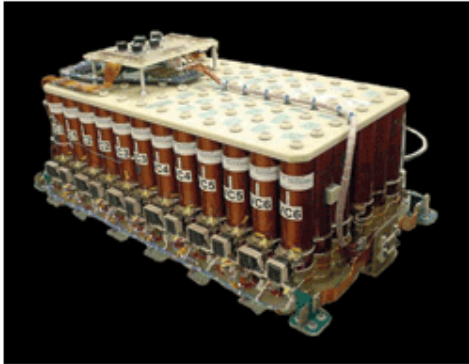


Fig. 2. Soft batteries used in satellites

Li-ion has numerous other advantages over NiCd and Ni-H<sub>2</sub>, including low self-discharge, wide operating temperature, and superior charging efficiency. Li-ion batteries retain about 96% to 98% of their charge even after a month of storage and need no recharging when they are about to be launched, as is the case with Ni-H<sub>2</sub>. Li-ion batteries can operate from 0° to 40°C, and they are easily recharged, further simplifying operation and maintenance. ■

For more on batteries, visit <http://www2.electronicproducts.com/Power.aspx>.

### Base: 10 ccm Student Nanosatellite OUFTI-1 of University of Liège (OUFTI-1 is designed for communications in D-STAR, an amateur-radio communication protocol)

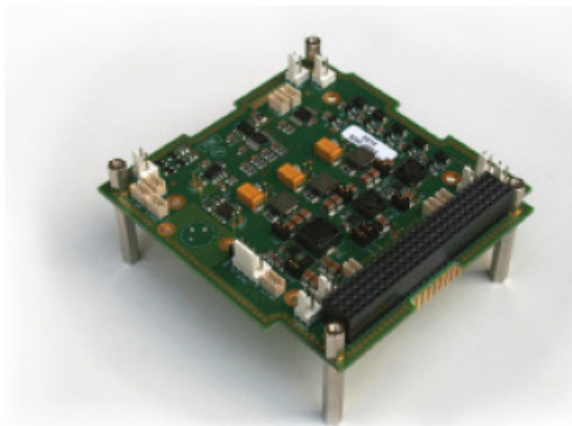
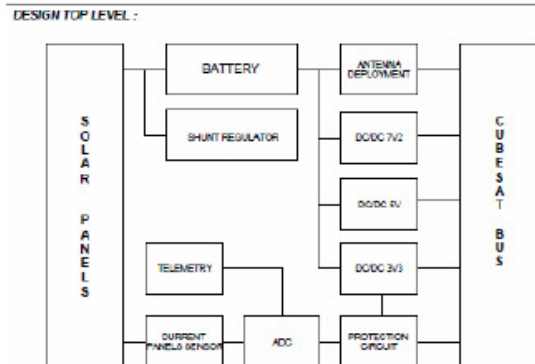


Figure C.3: Engineering model of the EPS card.





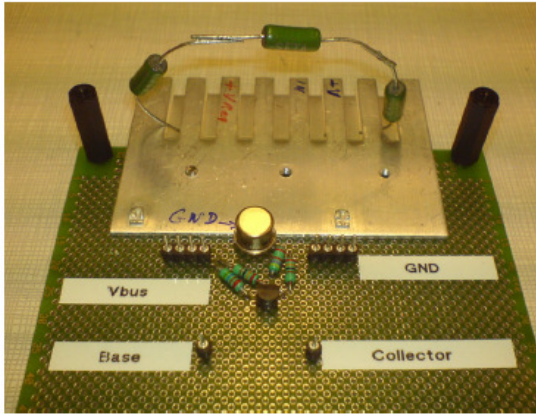


Figure C.1: Prototype of the shunt regulator.

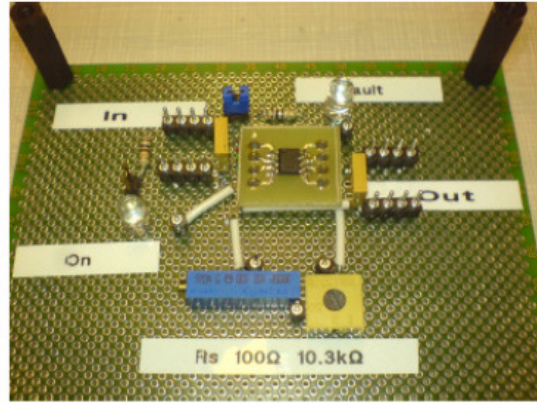


Figure C.2: Prototype of the protection circuit.

## Rechargeable Lithium-ion Batteries

### QL015KA



#### PERFORMANCE

- Lithium ion rechargeable cell
- Battery design for long cycle life (>100,000 cycles)

#### CHEMISTRY

- LNCAO

#### MAIN APPLICATIONS

- Military
- Aerospace

#### KEY FEATURES

- Domestic production
- Long LEO and GEO cycle life
- High reliability
- Zero-Volt™

#### SPECIFICATIONS

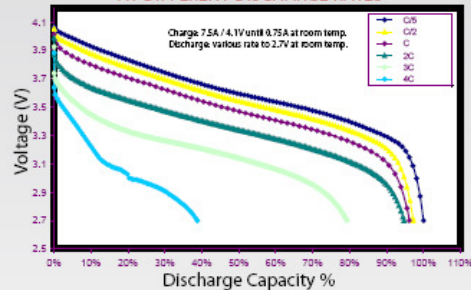
##### ELECTRICAL CHARACTERISTICS

Nominal Capacity	15 Ah
Nominal Voltage	3.6 V
Maximum Recommended Continuous Discharge Current	1 C
Operating Discharge Temperature	15°C to 35°C

##### PHYSICAL CHARACTERISTICS

Height	88.3 mm
Width	54.5 mm
Thickness	38.0 mm
Weight	380 g
Volume	182 cc
Gravimetric Energy Density	142 Wh/kg
Volumetric Energy Density	296 Wh/L

#### CELL LEVEL VOLTAGE CURVES AT DIFFERENT DISCHARGE RATES



### QL1820



#### PERFORMANCE

- Lithium ion rechargeable cell
- Battery design for long cycle life (>20,000 cycles)

#### CHEMISTRY

- LNCAO

#### MAIN APPLICATIONS

- Military
- Aerospace

#### KEY FEATURES

- Domestic production
- Long LEO and GEO cycle life
- High reliability
- Zero-Volt™

[www.quallion.com](http://www.quallion.com)

Quallion LLC | 12744 San Fernando Road | Sylmar, CA 91342 | 818.833.2000

### 6.1.4 Structures and Mechanisms

Initial investigations were performed in preparation for a full scale mock-up and to check the feasibility of the accommodation of all subsystems[66]. Early trade-off evaluations led to structural options to accommodate and position subsystems and components[25]. Further studies refined the selected structural options in preparation for the spacecraft prototype (see figures 3.7 and 3.8)[52]<sup>5</sup>.

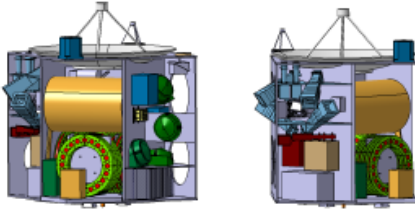


Figure 3.7: Structure option — View on propulsion module[52]      Figure 3.8: Structure option — View on star trackers and other subsystems[52]

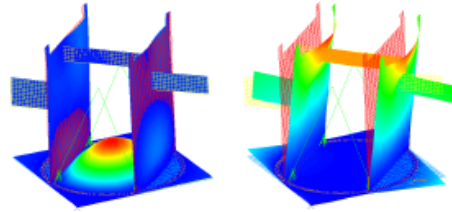
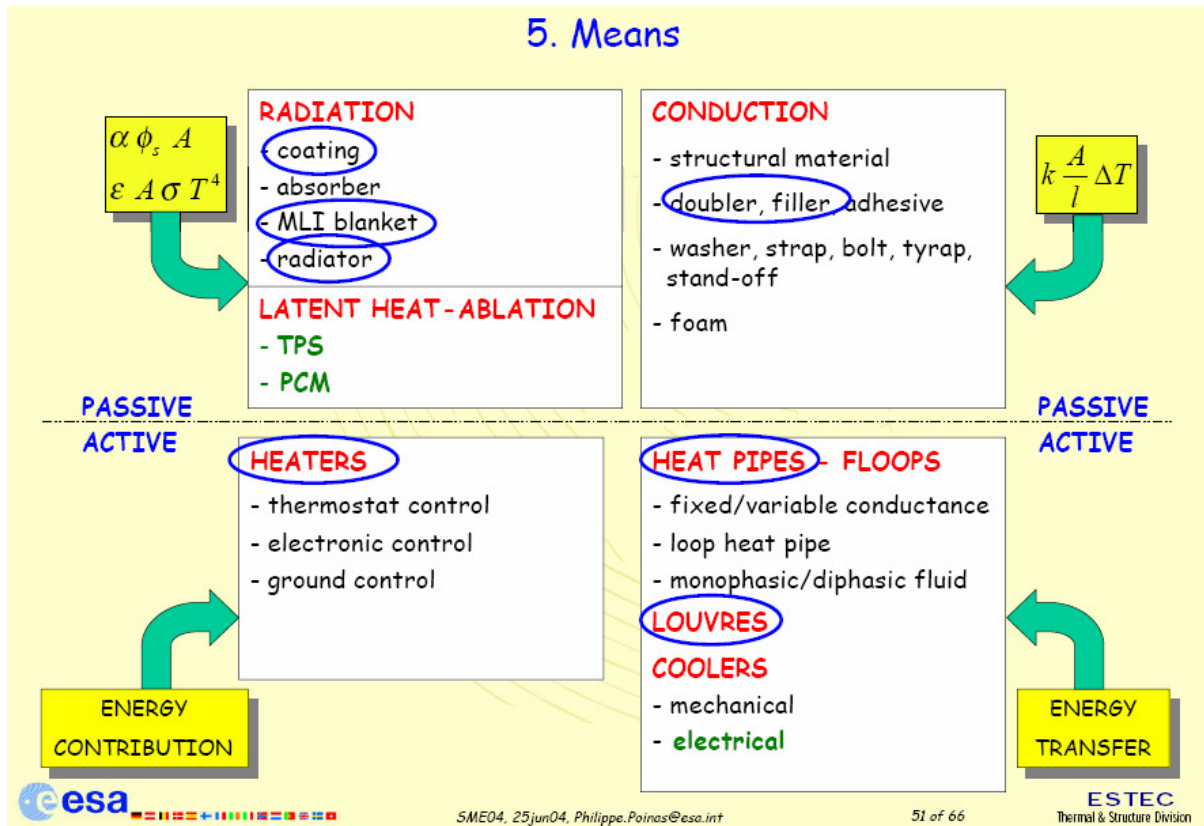


Figure 3.9: FEM analysis, longitudinal eigenfrequency[52]      Figure 3.10: FEM analysis, lateral eigenfrequency[52]

Analysis of different materials were performed to reduce the structural mass. Initial options using carbon fibre fold core sandwich material was replaced by honeycomb sandwich structures due to less complex production and usage as well as availability[26, 52]. Further studies must still be performed in order to determine the final structural design of the satellite bus incorporating all mechanisms and solar panel structure.

### 6.1.5 Thermal Control System

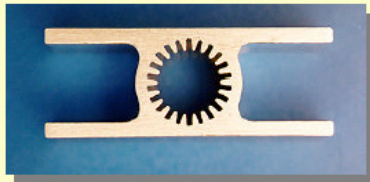


## 5. Means

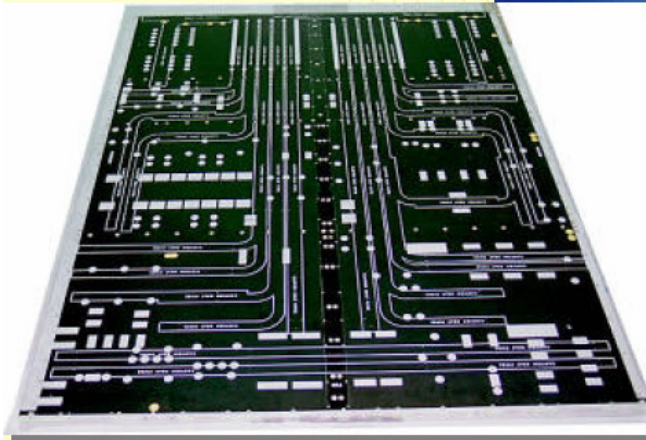
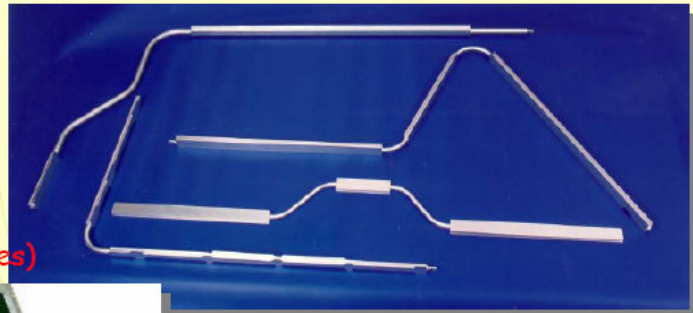
- **Passive Systems Pros/Cons**
  - no mechanical moving parts or moving fluids, no power consumption
    - simple to design/implement/test
    - low mass and cost
    - highly reliable
  - BUT low heat transport capability
    - except heat pipes
- **Active Systems Pros/Cons**
  - mechanical moving parts or moving fluids or electrical power required
    - complex design
    - generate constraints on S/C design and test configurations
    - high mass and cost
    - less reliable than PTC means

### 5. Means - ATC - Heat Pipes

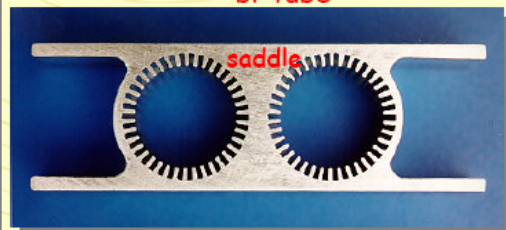
Ø 12 mm



Telecom Panel Heat Pipe (Swales)



bi-tube





## 5. Means - PTC - Radiation, Radiators



### 6.1.6 Flight Control System (FCS) on board of satellite - Attitude control system (ACS)

All possible elements of [Mourad et. al. 2006] shall be used.

### 6.1.6.1 Gear

From: Robert H. Berning III and Olivier Viout, Development of the Upgraded DC Brush Gearmotor FOR SPACEBUS PLATFORMS



Figure 2—Upgraded brush gearmotor (UBGM).

the existing motor design revealed inefficient processes, high brush drag and use of discontinued brush material.

The brush assembly consists of a carbon-composite brush, shunt wire, cap and spring. Eight different brush materials were considered and four were selected for testing. All brushes were tested for motor performance, resistance, drag, spring force, brush wear, commutator wear, smearing, debris and manufacturing yield. Brush material options 2 and 3 were eliminated due to low motor torque. Brush material option 1 was selected due to poor performance of option 4 in a vacuum. Table 2 lists development brush performance. The selected brush material is softer than the existing brush material, resulting in higher motor torque, lower brush drag and less commutator wear.

The motor trade study considered all assemblies and machined parts. The producibility of the existing motor is poor, due to the need for frequent rework resulting in high production costs. Stack fabrication, coating and attachment methodology were upgraded to current Moog procedures. All uncontrollable and unnecessary processes were replaced or eliminated. For instance, existing BGM commutators are machined after final armature assembly, putting the completed armature at risk. UBGM commutator processing was moved to the piece part level to lower the risk to hardware. The soldering process was updated to the current standard. Table 3 shows increased motor torque with new brush mate-

rials, design and manufacturing changes.

The overall development of the upgraded brush gearmotor was successful. All issues discovered during the Six-Sigma process were addressed. After development was completed, a qualification unit (Fig.2) was fabricated to specification, using production processes and tooling. The unit was subjected to qualification testing that included vibration, thermal-vacuum exposure and life tests. The qualification unit successfully passed all qualification and life tests with no findings.

#### Lessons Learned






While the upgraded brush motor development and qualification were successful, the methodology in some areas needs improvement. The following documents the major lessons learned during development and qualification:

*Understand derived requirements.* A firm understanding of the requirements (actual and derived) is needed prior to development. At the onset of the development process, the gearhead bushings were identified as a cause of BGM performance problems. A total redesign of the gearhead was started, with heritage design practices, processes and software utilized in the new gearhead. Gear design parameters were optimized to allow for greater allowable tolerances and used compatible material combinations to reduce galling and thermal expansion issues. Optimization of the gearhead for producibility adversely affected performance, however.

Since the BGM motor torque output is relatively low, it

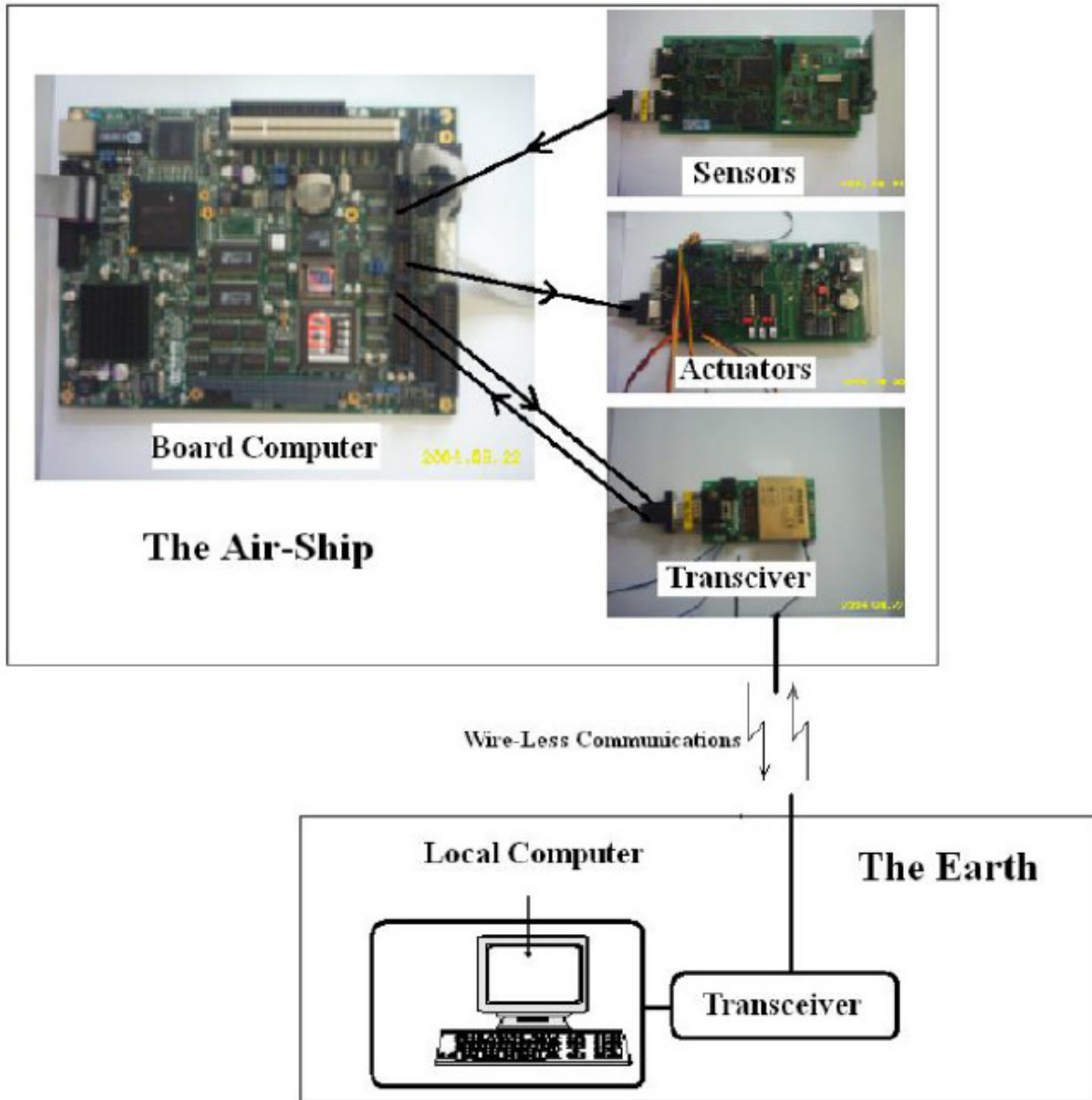
continued

Table 4—Brush Wear

				
New Brush	A1	A2	B1	B2
Percent Reduction	~14%	~14%	~14%	~17%

### 6.1.7 Command and Data Handling

All possible elements of [Mourad et. al. 2006] shall be used.



Also the elements from SBEP conception (May-July 2013).

See chapter "Telemetry"

### 6.2 Education and Public Outreach

To gain visibility within the scientific and engineering community as well as in the general public an education and public outreach program will be an important element of the project. Up-to-date communications involves classical instruments like press releases for broadcasting and print media coverage but also online media like web-based home pages, blogs and wiki-based information systems. Initiating partnerships with telecommunication and web media providers should favorably

To link the scientific community and the general public amateur astronomy groups will also be involved in payload experiments and the organization and management of ground-based observations (e.g. space and ground based imaging of transient lunar phenomena). These groups will establish a network to education and public outreach institutions like public observatories, science museums, planetariums and others to support an successful outreach program.

### 6.3 First Mockup Model

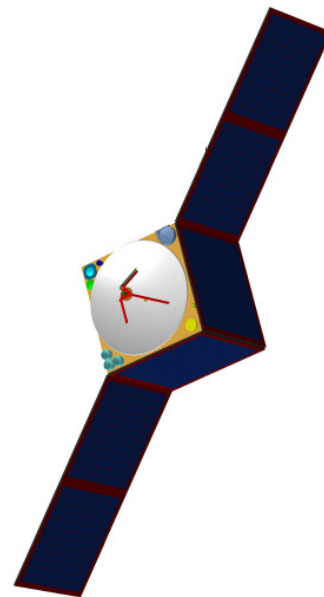


Figure 3.18: Accommodation of the strawman payload (left corner: MICSE, upper corner: SPOSH (bbo), PAICAM2 (dark bbo), right corner: LUBENA, center: TUES2[6])

## 6.4 Prototype

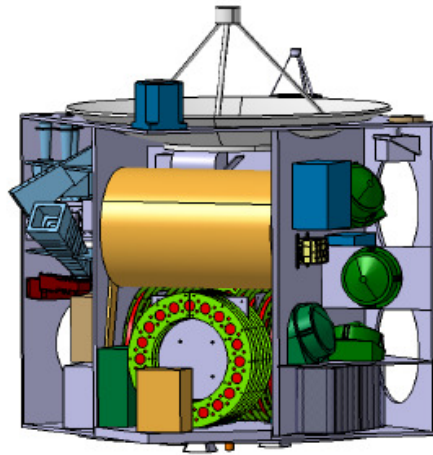


Figure 6.5: Prototype design — View on propulsion module[52]

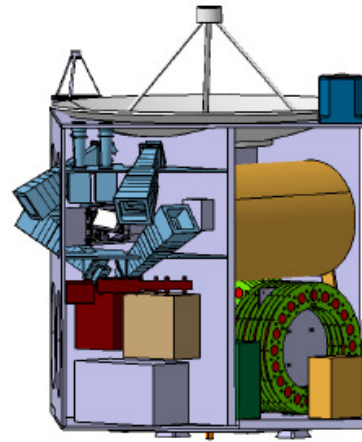


Figure 6.6: Prototype design — View on star trackers and other subsystems[52]



## 7 Mission Simulation 2012: Shuttle Flight simulation into orbit

The image is a composite of four panels related to a shuttle flight simulation. The top-left panel shows a 3D cutaway model of a shuttle, revealing internal components like the orbiter and external tank. The top-right panel shows a 3D perspective of the shuttle's trajectory, with a red and green light indicating its path. The bottom-left panel is a screenshot of a flight simulation interface, featuring a globe of Earth and a control panel with buttons for 'Quit', 'Start', and 'ERD-FLUCHT'. The bottom-right panel is a terminal window displaying a grid of coordinate data for various points along the shuttle's path.

gamma_317	- 43.121498	h_317	- 1887127.250000	x_317	- 1990971
gamma_318	- 43.119838	h_318	- 1895233.875000	x_318	- 1999628
gamma_319	- 43.116578	h_319	- 1903342.000000	x_319	- 2008287
gamma_320	- 43.114189	h_320	- 1911451.500000	x_320	- 2016948
gamma_321	- 43.111649	h_321	- 1919562.500000	x_321	- 2025612
gamma_322	- 43.109188	h_322	- 1927674.875000	x_322	- 2034278
gamma_323	- 43.106728	h_323	- 1935788.500000	x_323	- 2042946
gamma_324	- 43.104267	h_324	- 1943903.500000	x_324	- 2051616
gamma_325	- 43.101807	h_325	- 1952019.750000	x_325	- 2060289
gamma_326	- 43.099350	h_326	- 1960137.375000	x_326	- 2068963
gamma_327	- 43.096893	h_327	- 1968256.250000	x_327	- 2077640
gamma_328	- 43.094437	h_328	- 1976376.250000	x_328	- 2086318
gamma_329	- 43.091980	h_329	- 1984497.500000	x_329	- 2094999
gamma_330	- 43.089523	h_330	- 1992620.000000	x_330	- 2103682
gamma_331	- 43.087067	h_331	- 2000743.625000	x_331	- 2112366
gamma_332	- 43.084610	h_332	- 2008868.375000	x_332	- 2121053
gamma_333	- 43.082153	h_333	- 2016994.250000	x_333	- 2129741
gamma_334	- 43.079697	h_334	- 2025121.125000	x_334	- 2138432
gamma_335	- 43.077240	h_335	- 2033249.125000	x_335	- 2147124
gamma_336	- 43.074783	h_336	- 2041378.125000	x_336	- 2155818
gamma_337	- 43.072330	h_337	- 2049508.125000	x_337	- 2164514
gamma_338	- 43.069878	h_338	- 2057639.250000	x_338	- 2173212
gamma_339	- 43.067425	h_339	- 2065771.250000	x_339	- 2181912
gamma_340	- 43.064972	h_340	- 2073904.250000	x_340	- 2190613

#include "/home/iap/gtk+-uebungen/gtk-2.0/gtk/gtk.h"



```
#include <unistd.h>

void loeschen_funk(GtkWidget *widget, gpointer daten);
int raketenflugberechnung_ebeneErde();

int main(int argc, char *argv[])
{
    GtkWidget *hauptfenster;
    GtkWidget *schalter;
    GtkWidget *vbox;

    gtk_init(&argc, &argv);

    hauptfenster = gtk_window_new(GTK_WINDOW_TOPLEVEL);
    gtk_widget_set_usize(GTK_WIDGET(hauptfenster), 180,120);
    gtk_window_set_title(GTK_WINDOW(hauptfenster), "Mondprojekt");

    gtk_signal_connect(GTK_OBJECT(hauptfenster), "destroy",
        GTK_SIGNAL_FUNC(loeschen_funk), NULL);
    gtk_container_set_border_width(GTK_CONTAINER(hauptfenster), 20);

    vbox = gtk_vbox_new(TRUE, 0);
    gtk_container_add(GTK_CONTAINER(hauptfenster), vbox);

    /*Schalter einrichten*/
    schalter = gtk_button_new_with_label("Raketenflug berechnen");
    gtk_signal_connect(GTK_OBJECT(schalter), "clicked",
        GTK_SIGNAL_FUNC(raketenflugberechnung_ebeneErde), 0 );
    gtk_box_pack_start(GTK_BOX(vbox), schalter, TRUE, FALSE, 0);
    gtk_widget_show(schalter);

    /* Fenster sichtbar machen. */
}
```

```

gtk_widget_show(vbox);
gtk_widget_show(hauptfenster);

gtk_main();

g_print("main() wird nun beendet.\n");
return 0;
}

void loeschen_funk(GtkWidget *Widget, gpointer zdaten)
{
    g_print("Beenden : Destroy-Signal wurde empfangen.\n");
    gtk_main_quit();
}

/*-----*/
/*      Berechnung      */
/*-----*/

/*globale Variablen und Arrays.*/

float v[500];
float F = 30000;
float F_W = 0;
float c_W = 0.8;
float A_R = 0.8;
float luft_dichte = 0.002;
float m = 1000;
int g = 10;
float gamma_grad[500];
float gamma_rad;
float sin_gamma;
float cos_gamma;
float h[500];

```

```
float x[500];
int delta_t = 2;
int i;

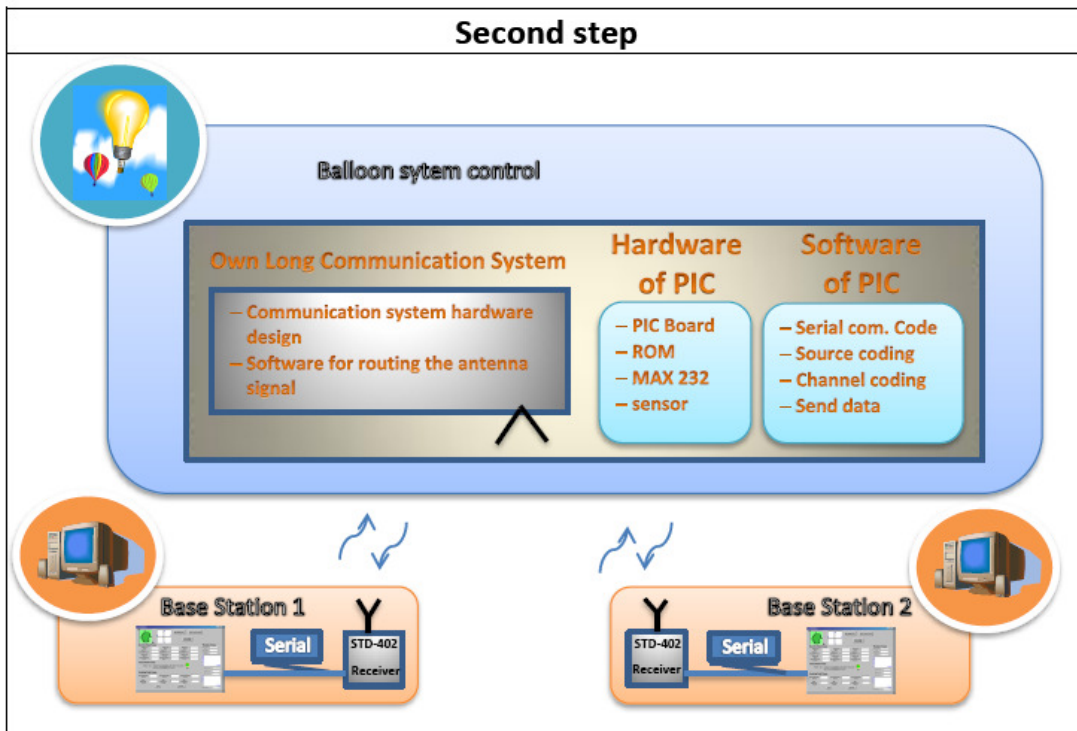
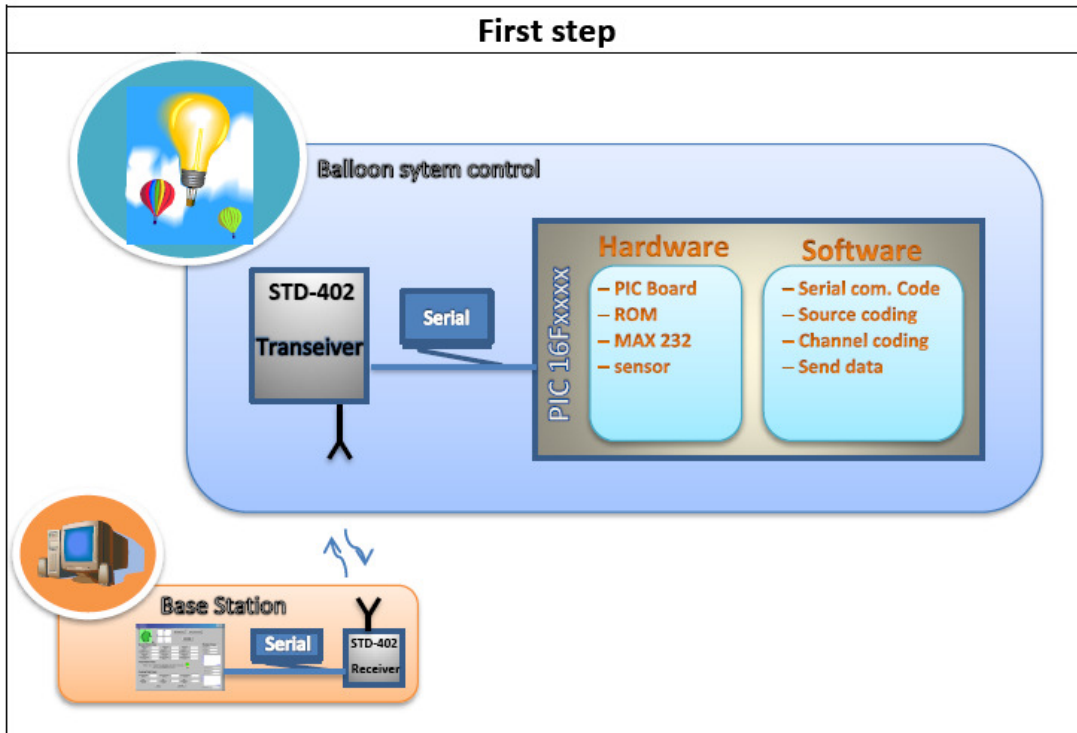
int raketenflugberechnung_ebeneErde()
{
    v[0] = 0;
    gamma_grad[0] = 90;
    h[0] = 0;
    x[0] = 0;

    for(i = 0; i < 500; i++)
    {
        sleep(2);
        gamma_rad = gamma_grad[i] * 3.1415/180;
        sin_gamma = sin(gamma_rad);
        cos_gamma = cos(gamma_rad);
        v[i+1] = ( (F-F_W) / m - (g*sin_gamma) ) * delta_t + v[i];
        F_W = c_W * A_R * (luft_dichte/2) * v[i] * v[i];
        gamma_grad[i+1] = ((-2*g)/(v[i]+v[i+1]))*cos_gamma*delta_t+gamma_grad[i];
        h[i+1] = 0.5 * (v[i] + v[i+1]) * sin_gamma * delta_t + h[i];
        x[i+1] = 0.5 * (v[i] + v[i+1]) * cos_gamma * delta_t + x[i];
        gamma_grad[2] = 45;

        if (m < 10)
            m = m - 10;
        printf("v_%d = %f m/s \t gamma_%d = %f \t h_%d = %f \t x_%d = %f \n" \
            , i, v[i], i, gamma_grad[i], i, h[i], i, x[i]);
    }
    return 0;
}
```

## 8 Telemetry (Satellite Bus Communication) Conception

### 8.1 From SBEP project<sup>1</sup>



<sup>1</sup> From [IAP\_SBEP]

## 8.2 Satellite side<sup>2</sup>

### 8.2.1 System Design

In general, the microcontrollers has to do the following (see figure 8.1):

- Establish a communication with the base station
- Control the Stratosphere balloon (stability and travel system)
- Source coding, encrypt code and channel coding
- Do a phased array

In our project we use the C programming language to program the microcontroller. A side to the software, there are some hardware should be build also will discuss it later.

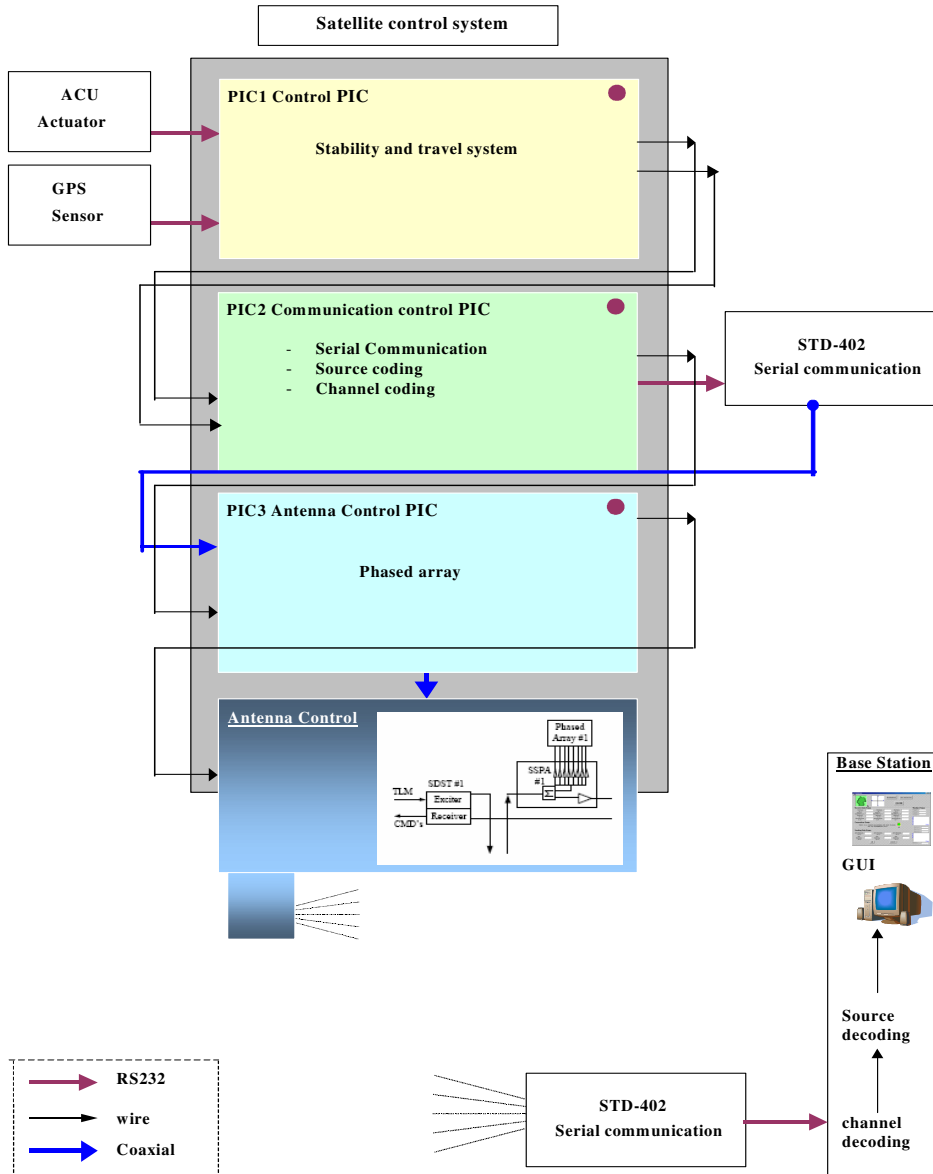


Figure 8.1: Satellite control system

<sup>2</sup> Migration from [IAP\_SBEP]

As we see in **figure1** the first PIC is connected to the ACU actuator ([Mourad et. al. 2006], Thesis of Jamal Ebeidieh) and to the GPS sensors board ([Mourad et. al. 2006], project of Mohamed Subhan) and connected to the base PIC to interact. The second PIC which is the base PIC is connected to the other PIC and to the STD-402 transceiver. The third PIC is connected to the base PIC and to the circuit of antenna control for a phased array antenna goal.

### 8.2.2 Working packages

WP No.	Working package content	Time span, material costs	Development environment (HW, SW)	Responsible	Status
1	Development of the base station GUI	0,5 MM	PC, Windows, VC++		
2	Development of Control for Stabilization of the stratoBallon (using alternativeLotte HW), scilab/xcos	2 MM			
2a	Requirements				
2b	Lotte MatrixX Code -> MATLAB				
2c	Generating Autocode, Loading on alternativeLotte Control Card				
3	Installation of a communication system for sending the data from the ballon to the base station (using a commercial radio sonde) (Payload 1)	Sep			
4		Sep			
4a	Modeling of communication path to stratosphere with octave	1 MM			
4b	Development of HW	3 MM			



### 8.3 Ground station side



#### 8.3.1 GUI Alternative 1: Using the old alternative Lotte program

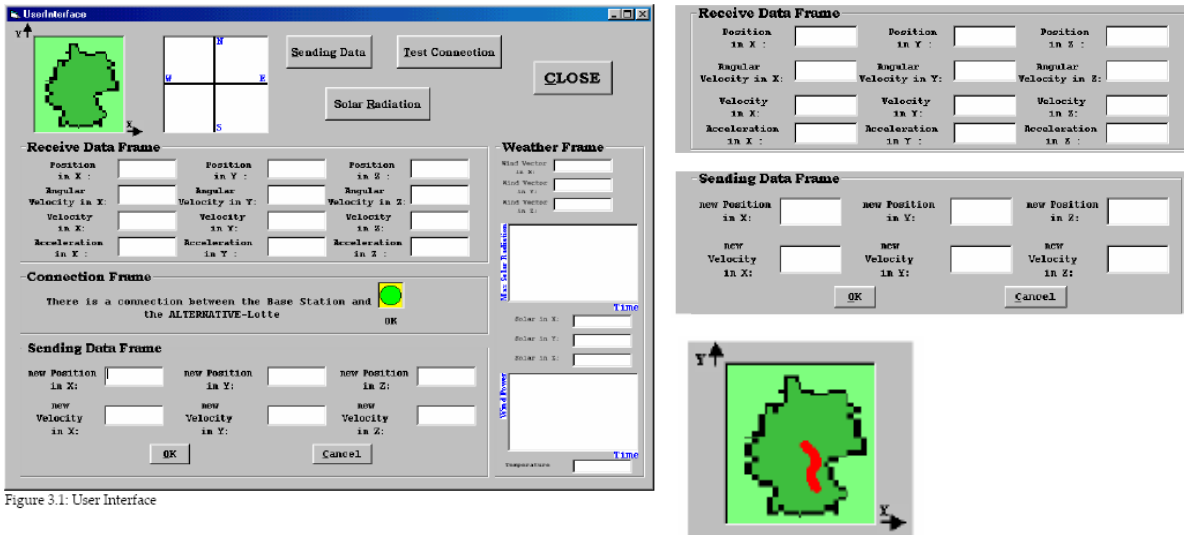
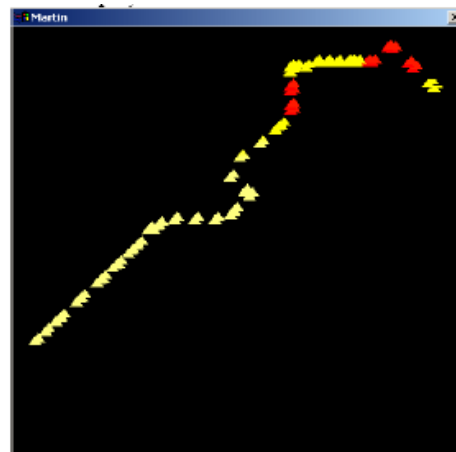
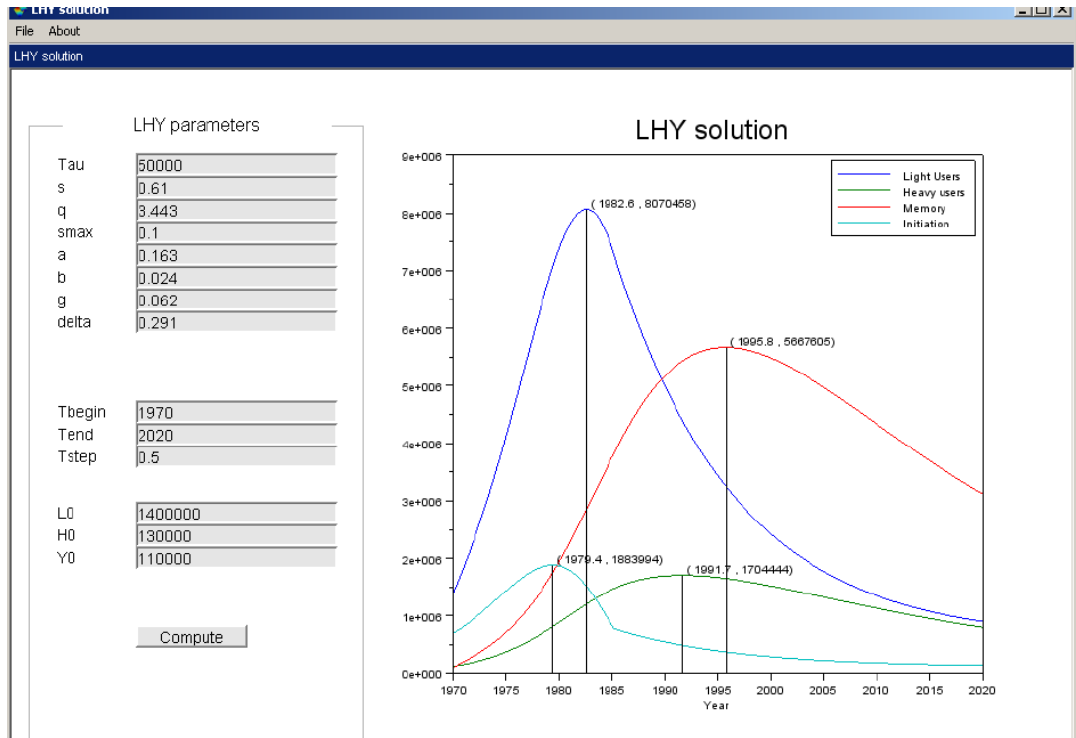


Figure 3.1: User Interface



Advantage: already programmed, only little changes necessary

### 8.3.2 GUI Alternative 2: In Scilab



See LHY\_Tutorial\_GUI.pdf

## 8.4 Transceiver Technology

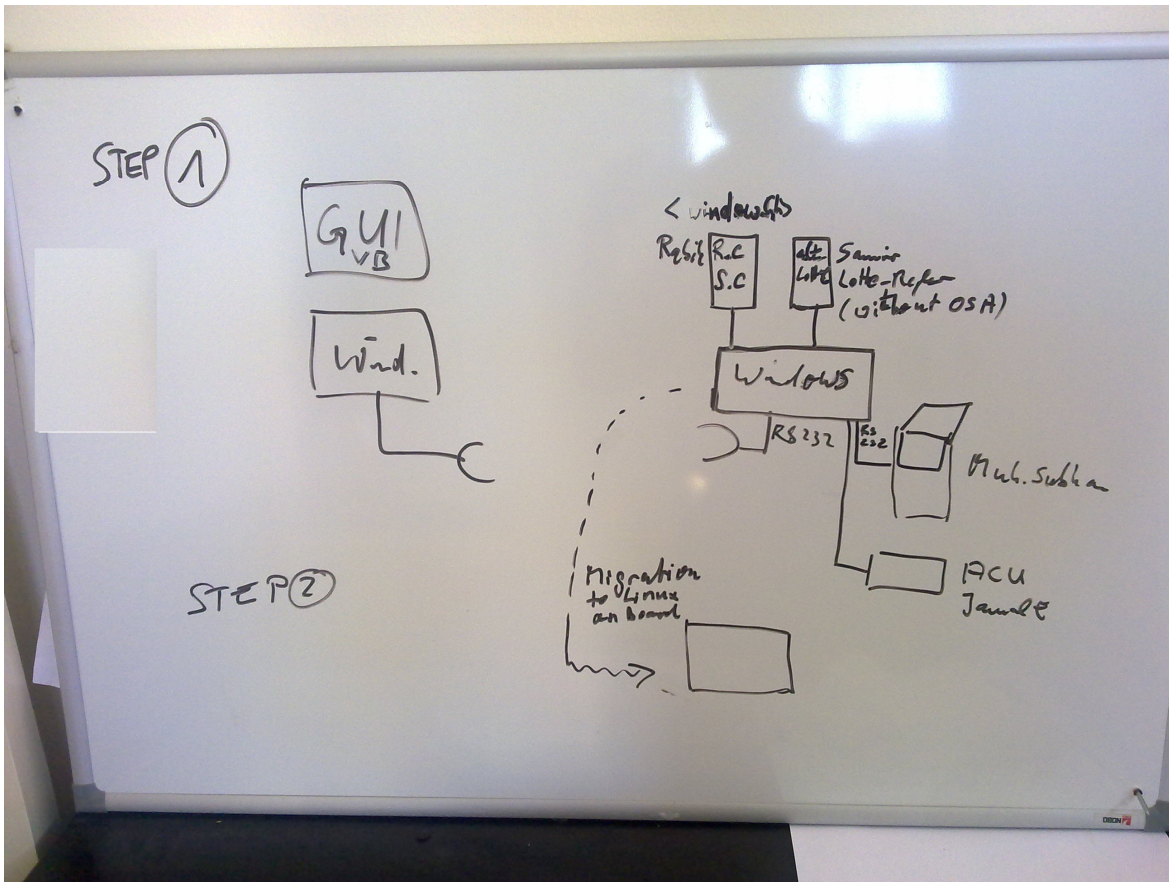
The transceiver technology shall use channel coding as on satellite communication technology (turbo codes). There exist MATLAB satellite communication simulation packages.

The design can be started based on the alternative Lotte transceiver (see Ref. [Mourad et. al. 2006], Thesis of Rabih al-Farkh). To improve the sending and receiving the antenna shall be modified.

There be used phased array technology both at sending and at receiving site.

### 8.4.1 Specification

It was decided, that the GUI and communication system of the former "alternative Lotte" project (Alternative 1) shall migrated, because this will reduce the development time insha Allah.



The ground station is a PC.

GUI:

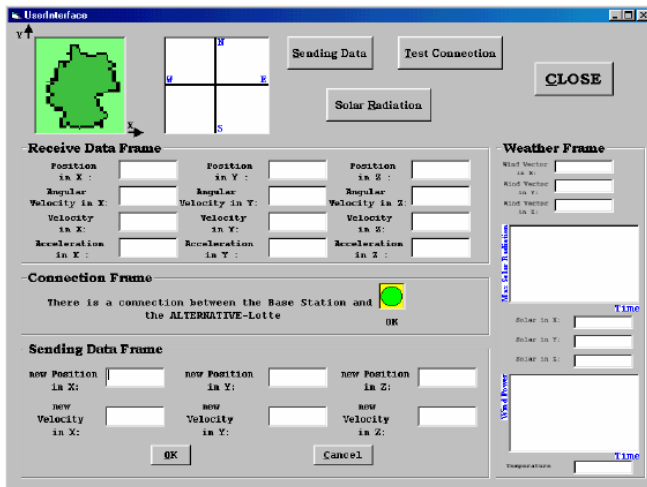


Figure 3.1: User Interface

### 8.4.1.1 STEP 1 (until 15 June 2013)

Communication and Control modules of the board computer shall be integrated on a Windows PC, HW (Sensor card, actuator card and comm.) shall be mounted with 3 serial ports (RS 232) to the PC.

At this stage also a GUI for the board computer is implemented (migration of KAHROUB solar circuit windows program and integration of the code of Rabih (Receive.C, Send.C) into it).

FCS is from

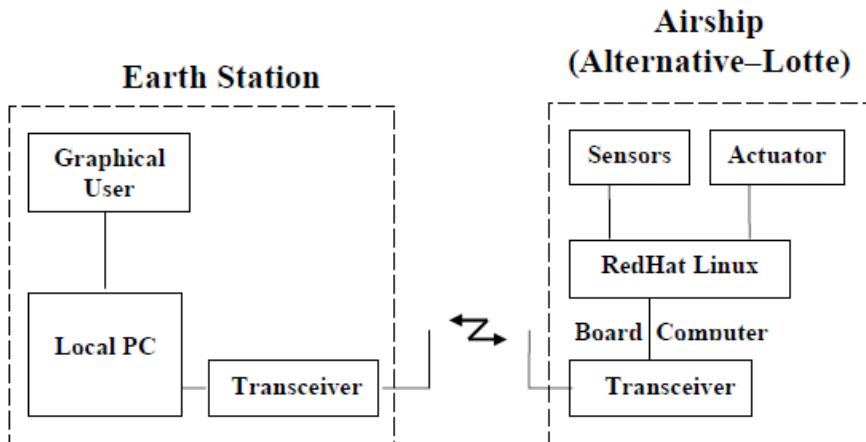
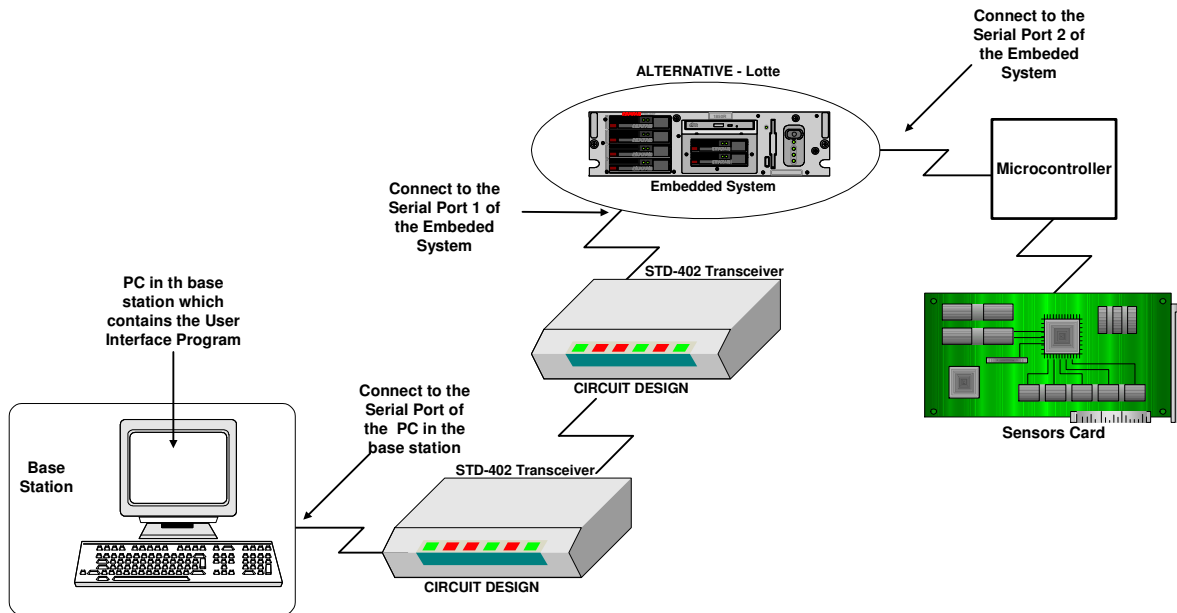
D:\AECENAR\IAP\Alternative Lotte\Regelung an Bord\osa\_lotte\lotte\_regler neufassung von Textdatei.txt

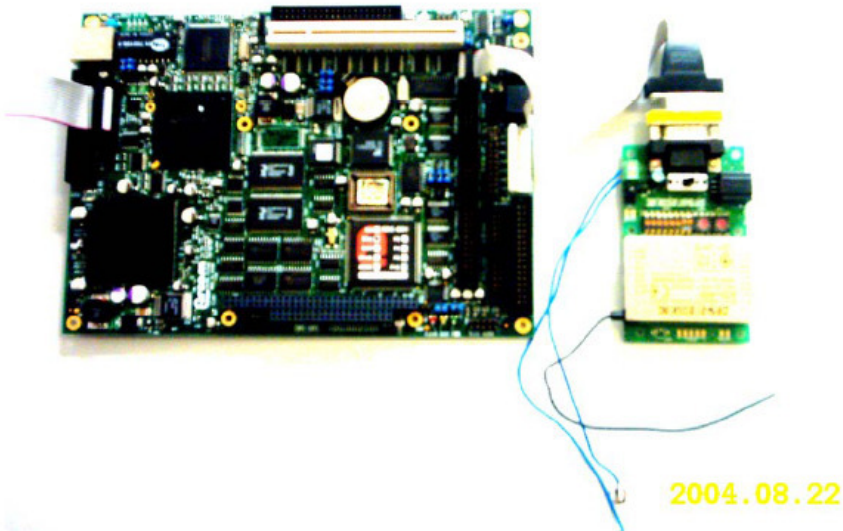
(Migration of former MATRIX autocode file D:\AECENAR\IAP\Alternative Lotte\Regelung an Bord\osa\_lotte\COPY of Lotte\_Regler.c)

**Development environment for STEP 1**

Visual Basic 6.0 (for Ground Station PC), C Compiler: MinGW, MSYS

**8.4.1.2 STEP 2 (until 30 June 2013)**





The communication is at 433 MHz. C Compiler for the microcontroller of the board.

### 8.4.1.3 STEP 3: Improvements

The two STD-42 transceivers can only communicate at distances < 500 m.

The SBEP system must have the ability to communicate up to about 60 km, because the balloon shall go up to 40 km (stratosphere) and due to unavoidable drifts the distance between the base station and the balloon will extend 50 km.

For this reason the following improvements shall be done:

1. send and receive antenna lobes shall be directed with phased array antennas.
2. channel coding must be added before sending to the user data

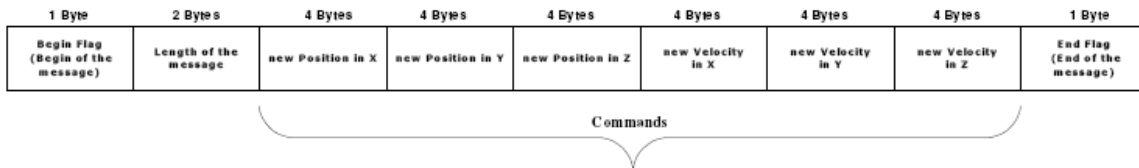


Figure 3.13: User Data Protocol \_ Send Message

and decoded after receiving the data.

### 8.4.1.4 Antenna improvement

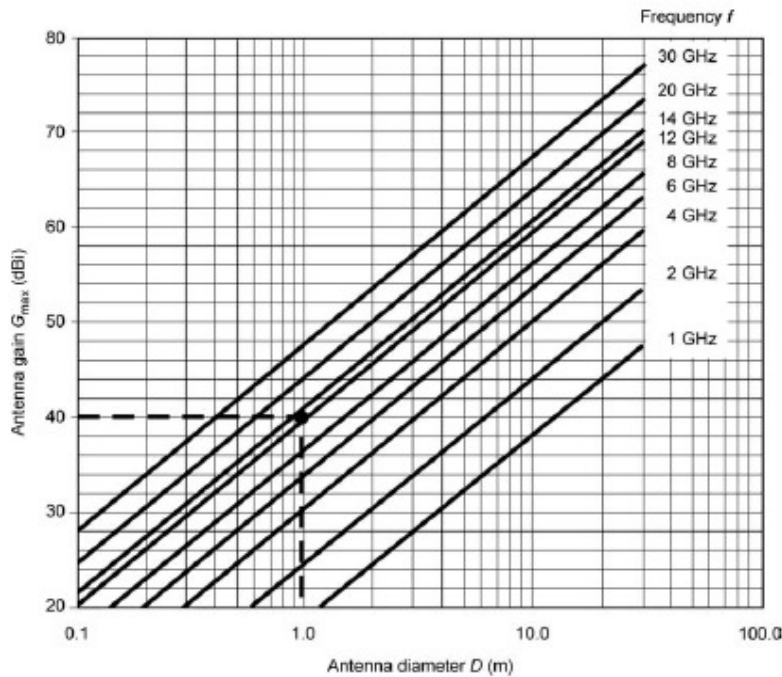


Figure 5.2 Maximum antenna gain as a function of diameter for different frequencies at  $\eta = 0.6$ . A 1 m antenna at 12 GHz has a gain of 40 dBi.

whose value depends on the chosen illumination law. For uniform illumination, the coefficient has a value of  $58.5^\circ$ . With non-uniform illumination laws, which lead to attenuation at the reflector boundaries, the 3 dB beamwidth increases and the value of the coefficient depends on the particular characteristics of the law. The value commonly used is  $70^\circ$  which leads to the following

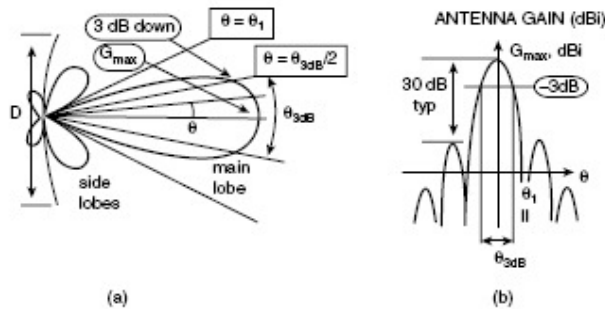


Figure 5.3 Antenna radiation pattern: (a) polar representation and (b) Cartesian representation.

From [Maral et. al. 2009].

### 8.4.1.5 Channel Coding

There shall be a implemented a turbo code.

## 8.5 Payload Communication: SCS-SMS including AES

See [IAP\_ECS]



## 9 Telemetry (Satellite Bus Communication) Prototype<sup>3</sup>

### 9.1 Overview

In our project, the satellite communicates with a base station wirelessly by two Synthesize transceivers STD-402. The base station is a computer connected to a Synthesize transceiver and on the computer there is a GUI user interface, in the satellite there are microcontrollers connect together and the hall system connect to the base station by a Synthesize transceiver.

#### 9.1.1 Synthesize transceiver STD-402

The transceiver to be used is **MB-STD-RS232**. It is a bi-directional semi-duplex radio modem having RS232 serial interface. It uses CIRCUIT DESIGN's standard 434 MHz FM Narrow Band transceiver module **STD-402** transceiver for RF part. This transceiver was selected because of its frequency of 434 MHz. For this frequency there is no extra permission necessary. Another reason is that this transceiver is a cheap one.

Further details see [Mourad et. al. 2006], Thesis of Rabih al-Farkh

### 9.2 Base station side<sup>4</sup>

In the base station side, a GUI user interface should build to interact with the satellite by sending and receiving data.

See [Mourad et. al. 2006], Thesis of Rabih al-Farkh, Chapters 3-4

### 9.3 Satellite side

In the satellite side, there are some PIC microcontrollers which should be programmed to do its work. In general, the microcontrollers aim to do the follow (see figure1):

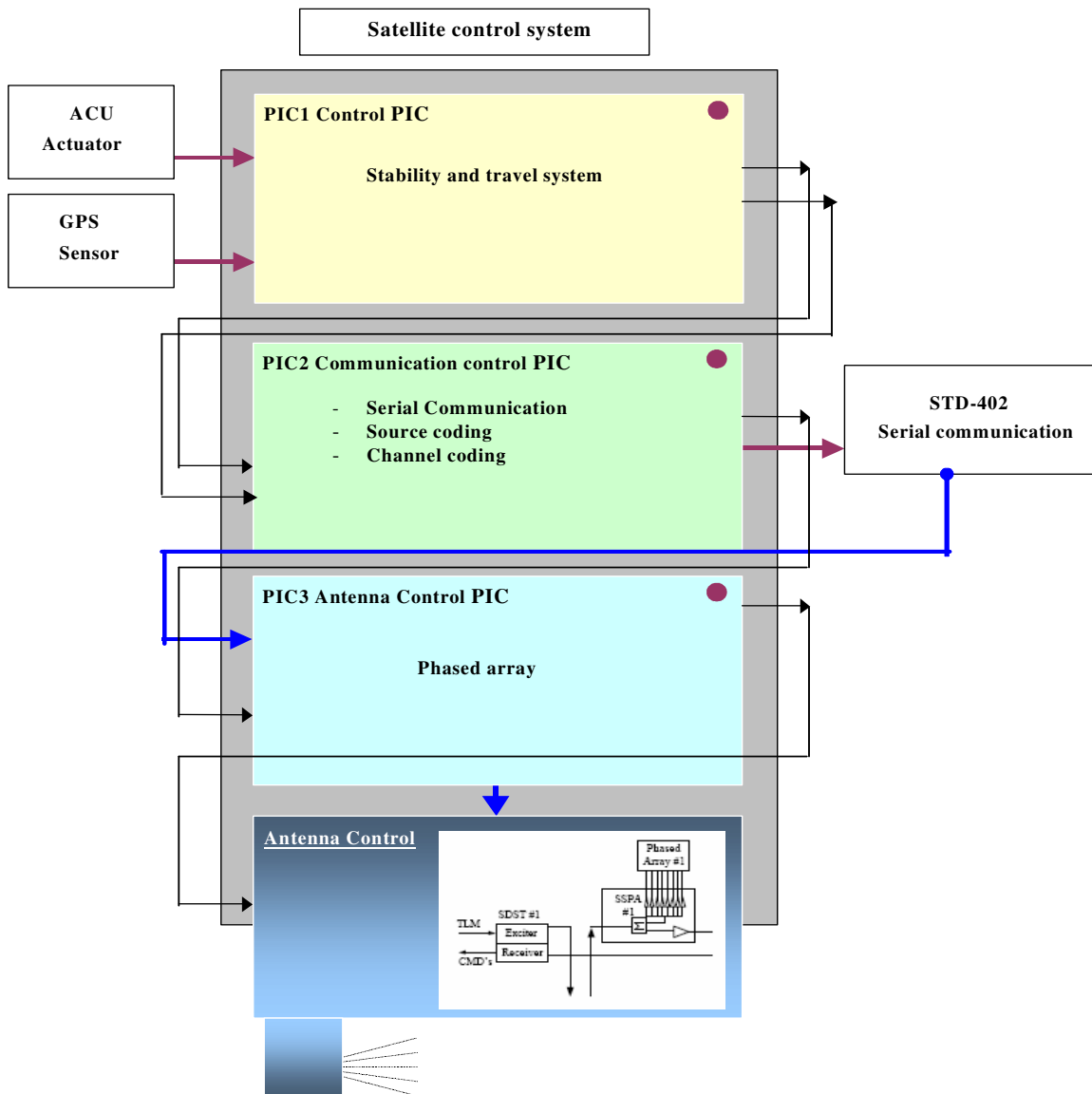
- Establish a communication with the base station
- Control the Stratosphere balloon (stability and travel system)
- Source coding, encrypt code and channel coding
- Do a phased array

In our project we use the C programming language to program the microcontroller. A side to the software, there are some hardware should be build also will discuss it later.

---

<sup>3</sup>

<sup>4</sup> most of this information is from the **Alternative-lotte** project of **Rabih al\_Farkh**



**Figure1: Satellite control system**

As we see in **figure1** the first PIC is connected to the ACU actuator (project of Jamal Ebeidieh) and to the GPS sensors board (project of Mohamed Subhan) and connected to the base PIC to interact. The second PIC which is the base PIC is connected to the other PIC and to the STD-402 transceiver. The third PIC is connected to the base PIC and to the circuit of antenna control for a phased array antenna goal.

### 9.3.1.1 Hardware part

We can divide the hardware part of our project as follow:

### 9.3.1.2 ACU actuator<sup>5</sup>

The ACU actuator is the project of Mr. Jamal Ebeideh. You can go back to his project report as a reference for this part.

### 9.3.1.3 GPS sensor<sup>6</sup>

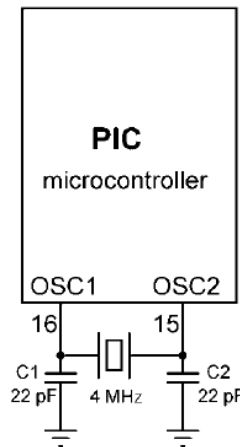
The ACU actuator is the project of Mr. Mohamed Subhan. You can go back to his project report as a reference for this part.

### 9.3.1.4 PIC board<sup>7</sup>

The PIC microcontroller needs a minimum support component to do its work. A PIC microcontroller, even though it may have been programmed, is not of much use unless it is supported by a number of components, such as the timing components and the reset circuit.

#### Timing components and design

A PIC microcontroller requires an external clock circuit (some PIC microcontrollers have built-in clock circuits) to function accurately. For accurate timing applications, the clock circuitry consists of a crystal, and two small capacitors. Figure bellow shows the circuit diagram of a PIC microcontroller with a 4-MHz crystal clock circuit. The crystal and the capacitors are connected to the OSC1 and OSC2 inputs of the microcontroller.



#### Power source and Reset circuit

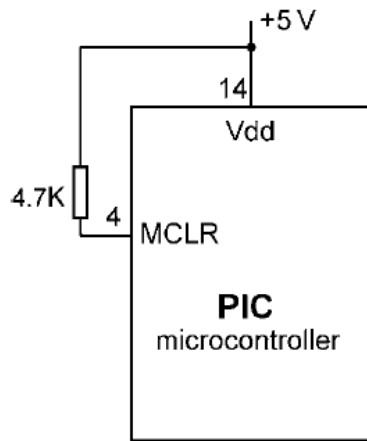
A PIC microcontroller starts executing the user program from address 0 of the program memory when power is applied to the chip. As shown in Figure bellow, the reset input (MCLR) of the microcontroller is usually connected to the 5V supply voltage through a 4.7K resistor.

---

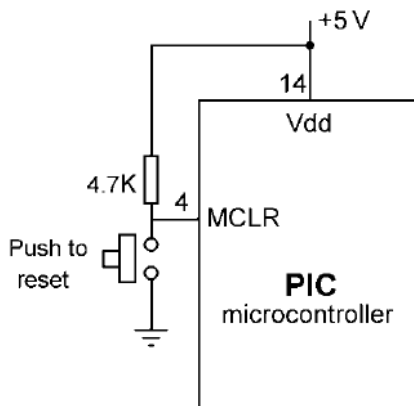
<sup>5</sup> Source: <https://www.aecenar.com/download/ACUactuator>

<sup>6</sup> Source: <https://www.aecenar.com/download/GPSsensors>

<sup>7</sup> Some information of this part is copy with change from: PICBASIC PROJECT for Dogan Ibrahim



There are many applications where the user may want to force reset action e.g. by pressing an external button so that the program re-starts to execute from the beginning. External reset is very useful during microcontroller-based system development and testing. Figure bellow shows how an external reset button can be connected to a PIC microcontroller. Normally the MCLR input is at logic 1, and goes to logic 0 which resets the microcontroller when the reset button is pressed. The microcontroller goes back to the normal operating mode when the button is released.

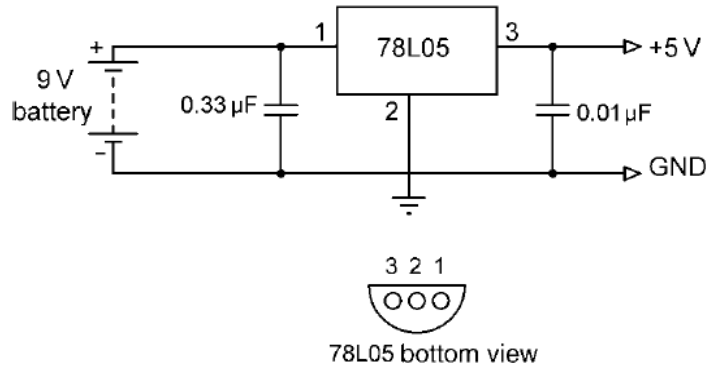


## Power supply

Every electronic circuit requires a power supply to operate. The required power can either be provided from a battery, or the mains voltage can be used and then reduced to the required level before it is used in the circuit (e.g. a mains adaptor). In this section, we shall look at the design of a power supply circuit to power our PIC microcontroller circuits.

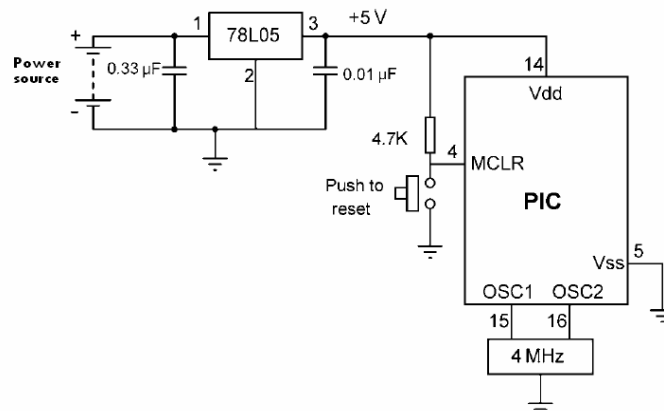
PIC microcontrollers can operate from a power supply voltage in the range 2 to 6V. The standard power supply voltage in digital electronic circuits is 5 V and this is the voltage with which the PIC microcontrollers are mostly operated. Unfortunately, it is not possible to obtain 5 V using standard alkaline batteries only. The nearest we can get is by using three batteries, which gives 4.5 V and this is not enough to power standard logic circuits. The simplest solution to drop the voltage from 9 to 5 V is by using a potential divider circuit using two resistors. Although a

potential divider circuit is simple, it has the major disadvantage that the voltage at the output depends on the current drawn from the circuit. As a result of this, the output voltage will change as we add or remove components from our circuit. Also, the output voltage falls as the battery is used. A voltage regulator circuit is needed to convert the 9 V battery voltage into 5V, independent of the current drawn from the supply. A basic voltage regulator circuit consists of a regulator integrated circuit and filter capacitors. Figure bellow shows a low-cost voltage regulator circuit using the 78L05-type voltage regulator IC, and two filter capacitors. 78L05 is a 3-pin IC with a maximum current capacity of 100 mA.

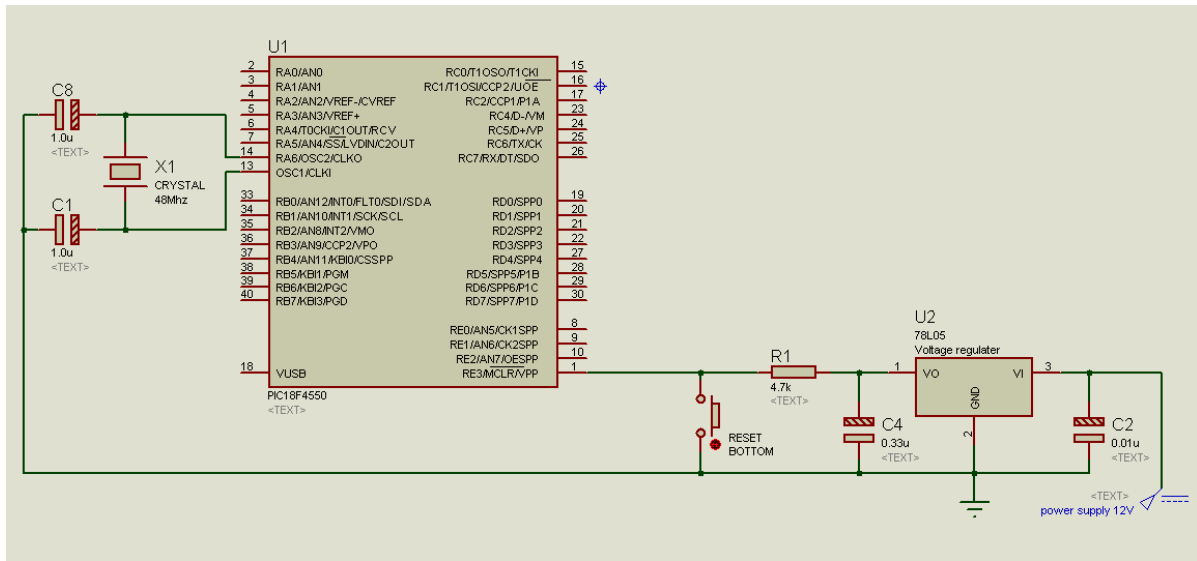


One of the pins of 78L05 is connected to the +V terminal of the battery in parallel with a 0.33-uF capacitor. One of the pins is connected to the -V terminal of the battery. The third pin provides the +5 V output and a 0.01-uF capacitor should be used in parallel with this pin. In applications where a larger current is required, the 7805 regulator IC can be used. This is pin compatible with the low-power 78L05 and it has a maximum current capacity of 1 A. 78L05 should be used with a suitable heat-sink in applications drawing more than a few hundreds of mill-amperes.

The complete circuit diagram of our PIC based basic system, together with the power supply, is shown in Figure bellow. The circuit is now fully functional, what is required now is to write our program and load it into the program memory of the microcontroller.



This circuit is tested on simulation software (Proteus isus7) under the design bellow:

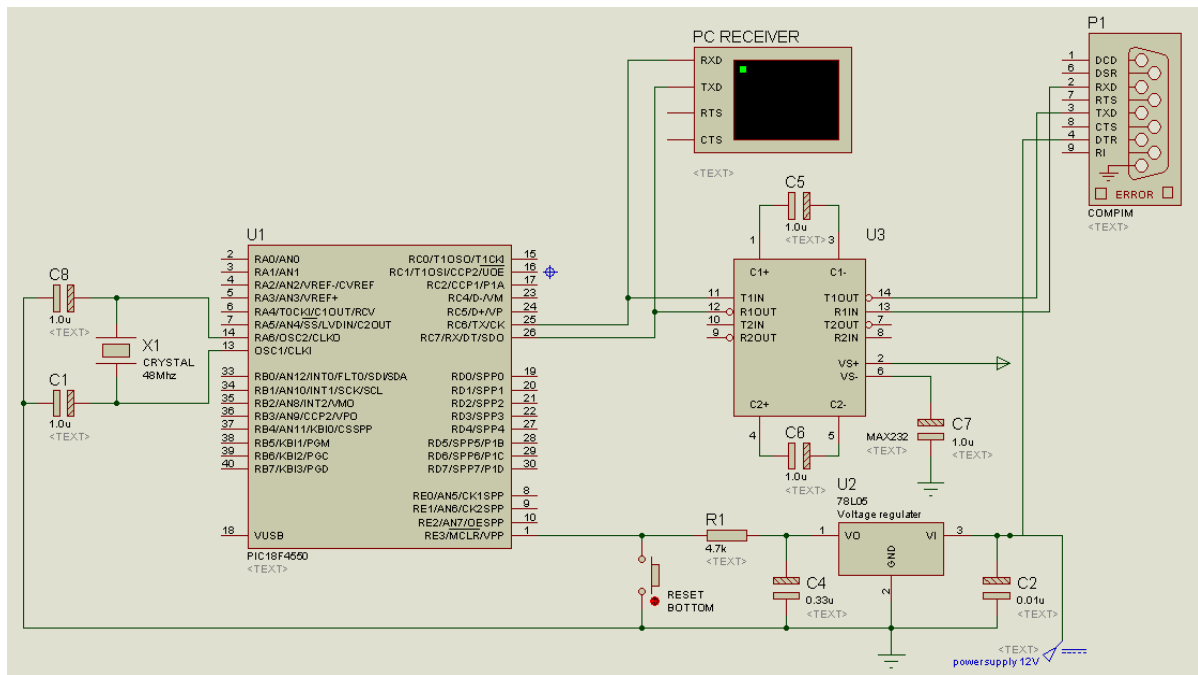


### Serial communication component

PIC microcontroller has serial communication module (USART) which operates using CMOS logic levels which changes between +5V and 0V to represent logic 1 and 0. However computer serial port (RS232C) operates in different voltage levels. It represents logic 0 with -10V and logic 1 with +10V. So obviously we can't directly connect PIC MCUs USART pins to computer. We need to convert voltage levels. For this task there is specially designed serial level converter ICs available in most common one is MAX232 which is used here. This IC receives signals from -10 to +10V from computer side and converts them into 0 and 5V which can be used in microcontroller

When using MAX232 IC it is essential to connect C2 capacitor as close as possible to MAX232 chip. Otherwise your circuit may not work. Also it is good idea to keep serial cable from computer to MAX232 reasonably short. The new Proteus design is shown in the figure bellow:





The virtual terminal was added only for testing purpose.

### 9.3.1.5 Antenna control board

Not yet . . .

## 9.3.2 Software part

We can divide the software part of our project as follow:

### 9.3.2.1 Communication code <sup>8</sup>

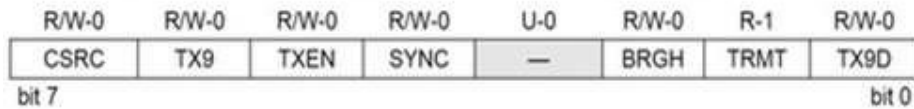
To communicate with external components such as computers or microcontrollers, the PIC micro uses a component called **USART – Universal Synchronous Asynchronous Receiver Transmitter**. This component can be configured as:

- A Full-Duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers
- A Half-Duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

To enable the serial communication with PIC microcontroller we must set different parameters within two registers:

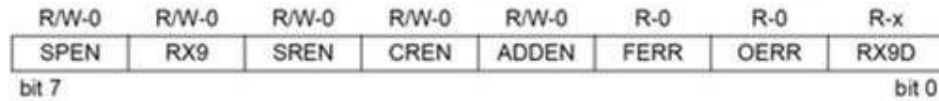
1. TXSTA – Transmit Status and Control Register

<sup>8</sup> Reference: some information in this part was copied with change from: [http://www.microcontrollerboard.com/pic\\_serial\\_communication.html](http://www.microcontrollerboard.com/pic_serial_communication.html)



MicrocontrollerBoard.com

## 2. RCSTA – Receive Status and Control Register



MicrocontrollerBoard.com

The send information will be stored inside **TXREG** register, which acts as a temporary buffer storage of information prior to transmission. While the receive information will be store in the **RSR** register, which acts as a temporary buffer storage.

Each transmission is transmitted in the particular rate (BAUD). The baud rate is measured in units of **bps** (bit per second). This is done by setting the system clock to the value needed. To do so, we need to “write” a hexadecimal number to the **SPBRG** register. The value written to the **SPBRG** register set the clock cycle to the value we want for the BAUD rate.

The size of **SPBRG** register is 8-bit. In asynchronous mode, the baud rate of transmission of the information can be set to high speed or to low speed. The rate selection, as already seen, is made by the BRGH bit in TXSTA register:

1 = High speed

0 = Low speed

For each baud rate we need to calculate the value being placed in the SPBRG differently:

$$SPBRG = (F_{osc} / (16 \times \text{Baud rate})) - 1, \quad BRGH = 1 \text{ High Speed}$$

$$SPBRG = (F_{osc} / (64 \times \text{Baud rate})) - 1, \quad BRGH = 0 \text{ Low Speed}$$

In our case, we have:  $F_{osc}=48\text{Mhz}$ ,  $\text{Baud rate}=9600$

$$\text{For High Speed} \Rightarrow SPBRG = 0x137 \quad \text{hex}$$

$$\text{For Low Speed} \Rightarrow SPBRG = 0x4D \quad \text{hex}$$

Then the code for serial sending is (C programming):

```
void SendToSerial(char m)
{
    SPBRG = 0x4D; // 4D hex or 77 decimal (baud rate=9600), Low
                 speed: SPBRG = (Fosc / (64 x Baud rate)) - 1
    TXSTA = 0x22; // determining the setting for the
```

```
transmitter
    RCSTA= 0x90; // determining the setting for the
receiver
    TXREG = m;
    while (PIR1bits.TXIF==0) {}
}
```

### 9.3.2.2 Control and stability code

Not yet ...

### 9.3.2.3 Coding code

This code part will discuss in the system coding chapter (next chapter), and includes:

- Source Coding and Decoding code
- Encryption and Decryption code
- Channel Coding and Decoding code

All of this code will discuss in C language.

### 9.3.2.4 Antenna control code

Not yet ...

## 9.4 System coding

In the engineering sense, coding can be classified into four areas:

- Encryption: to encrypt information for security purpose.
- Data compression: to reduce space for the data stream.
- Data translation: to change the form of representation of the information so that it can be transmitted over a communication channel.
- Error control: to encode a signal so that error occurred can be detected and possibly corrected.

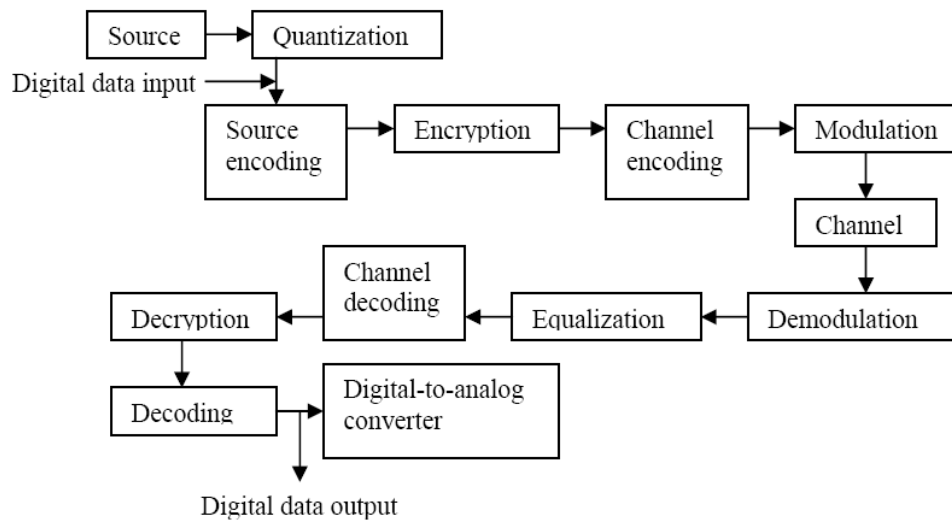
### 9.4.1 Introduction<sup>9</sup>

The main aim of any communication schemes is to provide error-free data transmission. In a communication system, information can be transmitted by analog or digital signals. For analog means, the amplitude of the signal reflects the information of the source, whereas for digital case, the information will first be translated into a stream of '0' and '1'. Then two different signals will be used to represent '0' and '1' respectively. As can be referred to the following illustration, the main advantage of using digital signal is that errors introduced by noise during the transmission can be detected and possibly corrected. For wireless communication channels, noise can be introduced in various ways.

---

<sup>9</sup> Reference: [http://www.mathdb.org/notes\\_download/elementary/number/ne\\_N6.pdf](http://www.mathdb.org/notes_download/elementary/number/ne_N6.pdf)

Figure 1 shows the flow of a simple digital communication system:



### 9.4.2 Source Coding

Suppose a word 'Zebra' is going to be sent out. Before this information can be transmitted to the channel, it is first translated into a stream of bits ('0' and '1'). The process is called **source coding**. There are many commonly used ways to translate that. For example, if ASCII code is used, each alphabet will be represented by 7-bit so-called the code word. The alphabets 'Z', 'e', 'b', 'r', 'a', will be encoded as

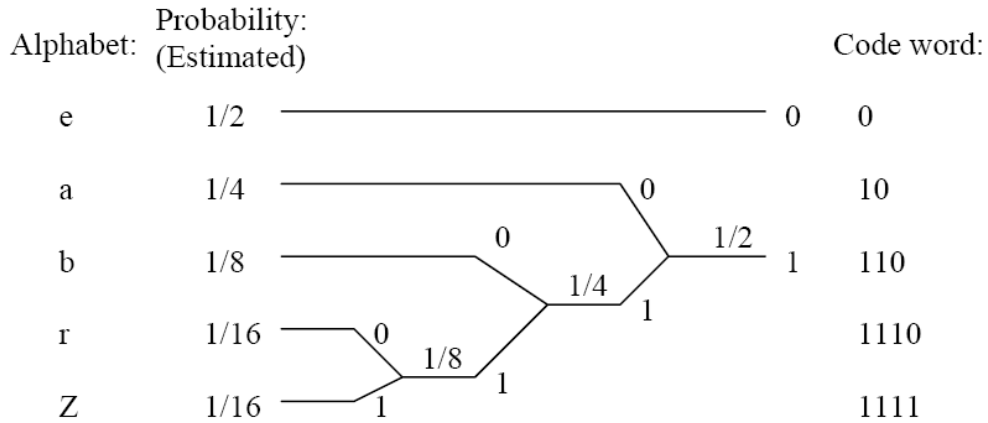
'1010101', '0110110', '0010110', '0010111', '0001110'

The **ASCII code** is an example of fixed-length code, because each of the code word is of the same length (7 bits). However, in the view of efficient communication, the occurrence of 'Z' is not as often as that of 'e' and 'a'. If there is a way of encoding information such that the alphabets with higher probability of occurrence are assigned with shorter code words, and longer for the other letters which seldom come out, then on the whole it may be able to conserve the number of bits to be sent to the channel while sending the same information. This is what the variable length code can do. The following illustrates the **Huffman Codes**, which was developed by David Huffman in 1951.

Rules:

- ✓ Order the symbols ('Z', 'e', 'b', 'r', 'a') by decreasing probability and denote them as  $S_1$ , to  $S_n$  ( $n = 5$  for this case).
- ✓ Combine the two symbols ( $S_n, S_{n-1}$ ) having the lowest probabilities. Assign '1' as the last symbol of  $S_{n-1}$  and '0' as the last symbol of  $S_n$ .
- ✓ Form a new source alphabet of  $(n-1)$  symbols by combining  $S_{n-1}$  and  $S_n$  into a new symbol  $S'_{n-1}$  with probability  $P'_{n-1} = P_{n-1} + P_n$ .
- ✓ Repeat the above steps until the final source alphabet has only one symbol with probability equals to 1.

Figure 2: The process of coding the word 'Zebra' by Huffman Code



### 9.4.2.1 Using Huffman theory

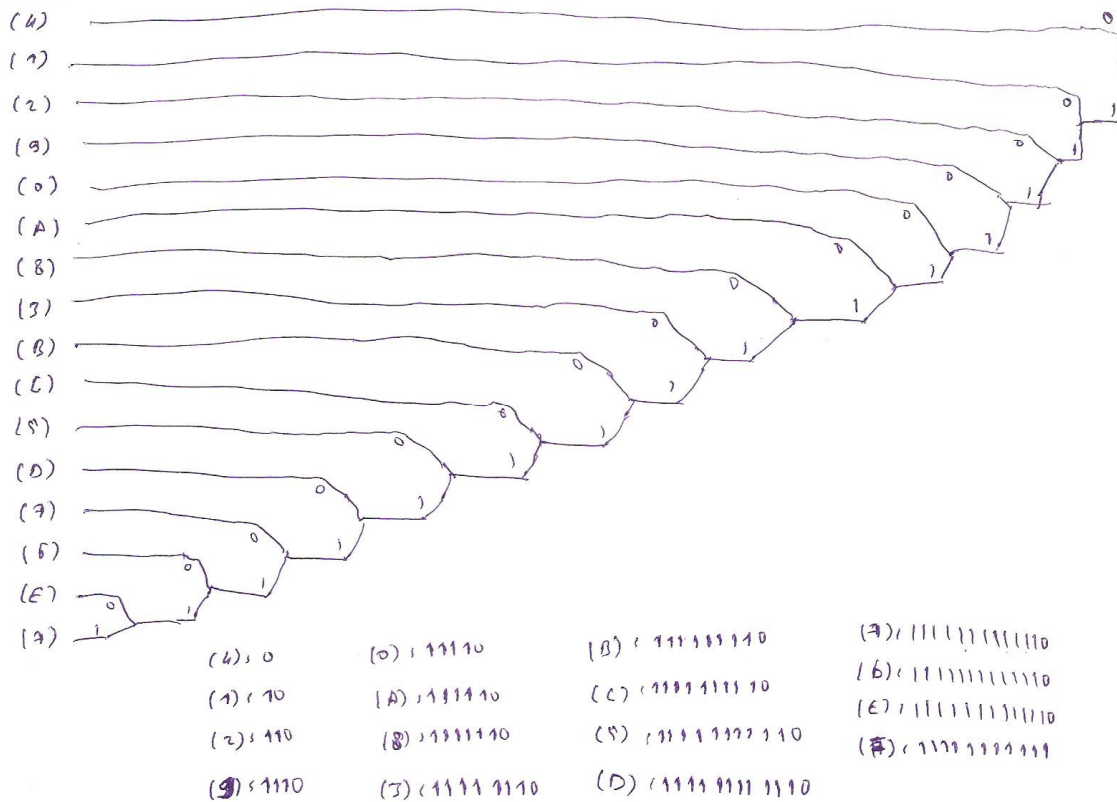
In the stratosphere balloon project, the Stratosphere balloon send to the base station a data message contain hexadecimal character with message length equal to 116. Hexadecimal data mean that we need a tree of 16 roots (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F) to use it in the Huffman coding strategy.

The probability of each character can't know by us but we can use some message data and calculate the percentage in it. For this purpose, I use 10 data message and the percentages were as follow:

- |               |               |               |              |
|---------------|---------------|---------------|--------------|
| (0): 70/1160  | (1): 161/1160 | (2): 106/1160 | (3): 59/1160 |
| (4): 202/1160 | (5): 58/1160  | (6): 41/1160  | (7): 43/1160 |
| (8): 61/1160  | (9): 75/1160  | (A): 67/1160  | (B): 54/1160 |
| (C): 50/1160  | (D): 47/1160  | (E): 40/1160  | (F): 36/1160 |

Then we can sort the character as follow:

$$4 > 1 > 2 > 9 > 0 > A > 8 > 3 > B > C > 5 > D > 7 > 6 > E > F$$



By trying to solve a Huffman code using this tree we get:

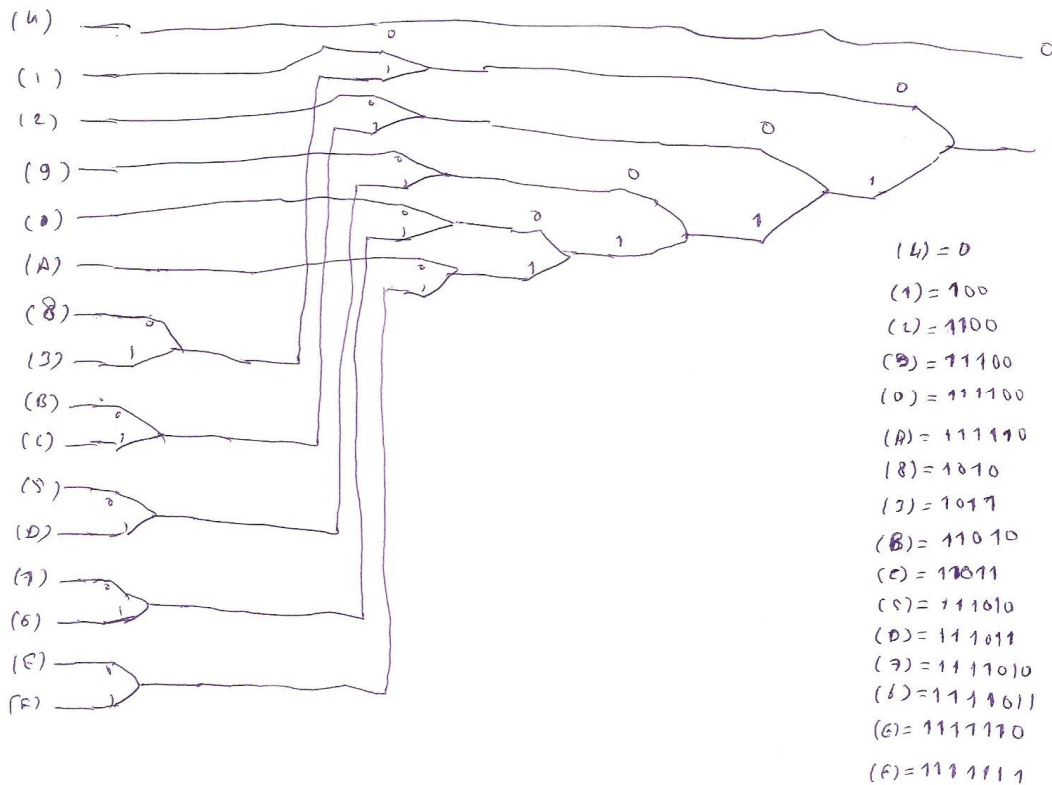
Let try with this msg: 41A73C42

The code will be: 010111110111111111111111101111111011111111100110

We can advanced this tree to reduce the number of bits, the tree bellow reduce the number of bits for each character







By trying to solve the same Huffman code using the new tree we get:

Message : 41A73C42

The code will be: 0100111110111101010111101100

9.4.2.2 Write the Huffman code (c programming)



As we see in the last part that each Hex characters have one binary code as the following table:

Table1: our Huffman code table

0	111100	4	0	8	1010	C	11011
1	100	5	111010	9	11100	D	111011
2	1100	6	1111011	A	111110	E	1111110
3	1011	7	1111010	B	11010	F	1111111

Now, let's start with the coding side:

The **coding** part is done by 16 (if) conditions for the 16 Hex characters we have it:

Let's start:

```
void Source_Coding(char msg[])
{
    int i, lenght;
    char c0[7] = "111100";
    char c1[7] = "100";
    char c2[7] = "1100";
    char c3[7] = "1011";
    char c4[7] = "0";
    char c5[7] = "111010";
    char c6[7] = "1111011";
    char c7[7] = "1111010";
    char c8[7] = "1010";
    char c9[7] = "11100";
    char cA[7] = "111110";
    char cB[7] = "11010";
    char cC[7] = "11011";
    char cD[7] = "111011";
    char cE[7] = "1111110";
    char cF[7] = "1111111";
    lenght = strlen(msg);
    for(i=0; i <= lenght; i++){
        if (msg[i] == '0')    Send_STRING(c0);
        else if (msg[i] == '1') Send_STRING(c1);
        else if (msg[i] == '2') Send_STRING(c2);
        else if (msg[i] == '3') Send_STRING(c3);
        else if (msg[i] == '4') Send_STRING(c4);
        else if (msg[i] == '5') Send_STRING(c5);
        else if (msg[i] == '6') Send_STRING(c6);
        else if (msg[i] == '7') Send_STRING(c7);
        else if (msg[i] == '8') Send_STRING(c8);
        else if (msg[i] == '9') Send_STRING(c9);
```

```

        else if (msg[i] == 'A') Send_STRING(cA);
        else if (msg[i] == 'B') Send_STRING(cB);
        else if (msg[i] == 'C') Send_STRING(cC);
        else if (msg[i] == 'D') Send_STRING(cD);
        else if (msg[i] == 'E') Send_STRING(cE);
        else if (msg[i] == 'F') Send_STRING(cF);

        //      else set the error bit in the sending message
    }

//      Delay10KTCYx(1000);
}

```

In the other hand, in the **decoding** side is done also by some (if) conditions as follows:

```

For(i=0; i<= msglength; ){
If (code[i] = 0) {i++; m = '4'; break;}
Else {
    // the first 1 on the code
    I++;
    if (code[i] = 0){ // the code now is: 10
        if (code[i] = 0) {i++; m = '1'; break;}
        else { // the code now is: 101
            i++;
            if(code[i] = 0) {i++; m = '8'; break;}
            else {i++; m = '3'; break;}
        }
    }
    Else { // the code now is: 11
        I++;
        If(code[i] = 0) { // the code now is: 110
            I++;
            If(code[i] = 0) {i++; m = '2'; break;}
            Else {
                i=i+2;
                If(code[i] = 0) { m = 'B'; i++; break;}
                Else { m = 'C'; i++; break;}
            }
        }
    }
}

```



### 9.4.3 Encryption<sup>10</sup>

One of the most widely used block cipher algorithms is the **Data Encryption Standard (DES)**, adopted in 1977 by the American National Standards Institute (ANSI).

After more than twenty years of use with continuous aging due to advances in cryptography, the National Institute of Standards and Technology (NIST). On 2 October 2000 the NIST announced that the new encryption technique, named **Advanced Encryption Standard (AES)**, would use the Rijndael algorithm, designed by two well-known specialists, Joan Daemen and Vincent Rijmen from Belgium.

#### 9.4.3.1 Advanced Encryption Standard (AES)<sup>11</sup>

AES is based on a design principle known as a substitution-permutation network, and is fast in both software and hardware. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, the Rijndael specification *per se* is specified with block and key sizes that may be any multiple of 32 bits, both with a minimum of 128 and a maximum of 256 bits.

AES operates on a 4×4 column-major order matrix of bytes, termed the *state*, although some versions of Rijndael have a larger block size and have additional columns in the state. Most AES calculations are done in a special finite field.

The key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input, called the plaintext, into the final output, called the cipher-text. The number of cycles of repetition is as follows:

- 10 cycles of repetition for 128-bit keys.
- 12 cycles of repetition for 192-bit keys.
- 14 cycles of repetition for 256-bit keys.

Each round consists of several processing steps, each containing five similar but different stages, including one that depends on the encryption key itself. A set of reverse rounds are applied to transform cipher-text back into the original plaintext using the same encryption key.

#### 9.4.3.2 AES algorithm

1. KeyExpansion—round keys are derived from the cipher key using Rijndael's key schedule.
2. InitialRound
  1. AddRoundKey—each byte of the state is combined with the round key using bitwise **xor**.
3. Rounds
  1. SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.

---

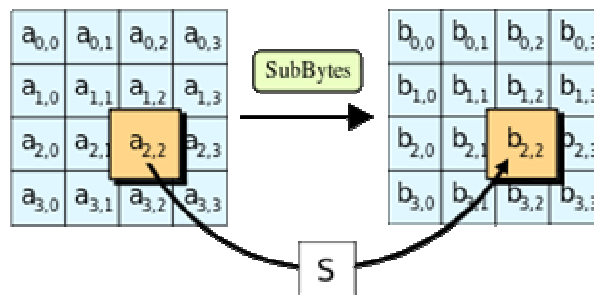
<sup>10</sup> Reference: Microchip AN821 Advanced Encryption Standard Using the PIC16XXX page:1

<sup>11</sup> Reference: [http://en.wikipedia.org/wiki/Advanced\\_Encryption\\_Standard](http://en.wikipedia.org/wiki/Advanced_Encryption_Standard)

2. `ShiftRows`—a transposition step where each row of the state is shifted cyclically a certain number of steps.
  3. `MixColumns`—a mixing operation which operates on the columns of the state, combining the four bytes in each column.
  4. `AddRoundKey`
4. Final Round (no `MixColumns`)
1. `SubBytes`
  2. `ShiftRows`
  3. `AddRoundKey`

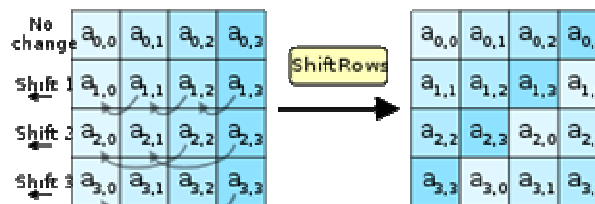
### The SubBytes step

In the `SubBytes` step, each byte in the state matrix is replaced with a SubByte using an 8-bit substitution box, the Rijndael S-box. This operation provides the non-linearity in the cipher. The S-box used is derived from the multiplicative inverse over  $GF(2^8)$ , known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box is also chosen to avoid any fixed points (and so is a derangement), and also any opposite fixed points.



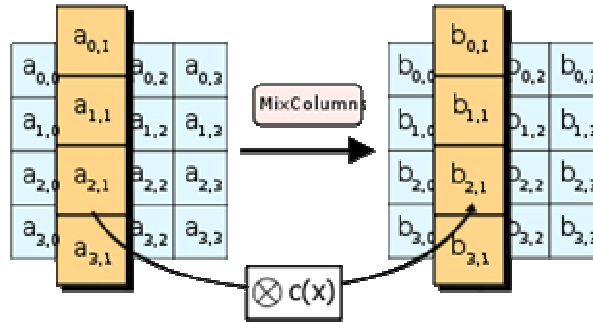
### The ShiftRows step

The `ShiftRows` step operates on the rows of the state; it cyclically shifts the bytes in each row by a certain offset. For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. For blocks of sizes 128 bits and 192 bits, the shifting pattern is the same. Row  $n$  is shifted left circularly by  $n-1$  bytes. In this way, each column of the output state of the `ShiftRows` step is composed of bytes from each column of the input state. (Rijndael variants with a larger block size have slightly different offsets). For a 256-bit block, the first row is unchanged and the shifting for the second, third and fourth row is 1 byte, 2 bytes and 3



bytes respectively—this change only applies for the Rijndael cipher when used with a 256-bit block, as AES does not use 256-bit blocks. The importance of this step is to make columns not linear independent. If so, AES becomes four independent block ciphers.

**The MixColumns step**



In the MixColumns step, the four bytes of each column of the state are combined using an invertible linear transformation. The MixColumns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. Together with ShiftRows, MixColumns provides diffusion in the cipher.

During this operation, each column is multiplied by the known matrix that for the 128-bit key is:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

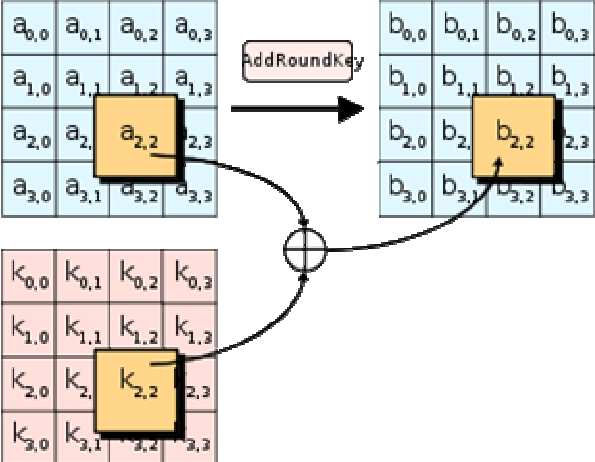
The multiplication operation is defined as: multiplication by 1 means no change, multiplication by 2 means shifting to the left, and multiplication by 3 means shifting to the left and then performing xor with the initial unshifted value. After shifting, a conditional xor with 0x1B should be performed if the shifted value is larger than 0xFF.

In more general sense, each column is treated as a polynomial over GF(28) and is then multiplied modulo x4+1 with a fixed polynomial c(x) = 0x03 · x3 + x2 + x + 0x02. The coefficients are displayed in their hexadecimal equivalent of the binary representation of bit polynomials from GF(2)[x]. The MixColumns step can also be viewed as a multiplication by a particular MDS matrix in a finite field. This process is described further in the article Rijndael mix columns.

**The AddRoundKey step**



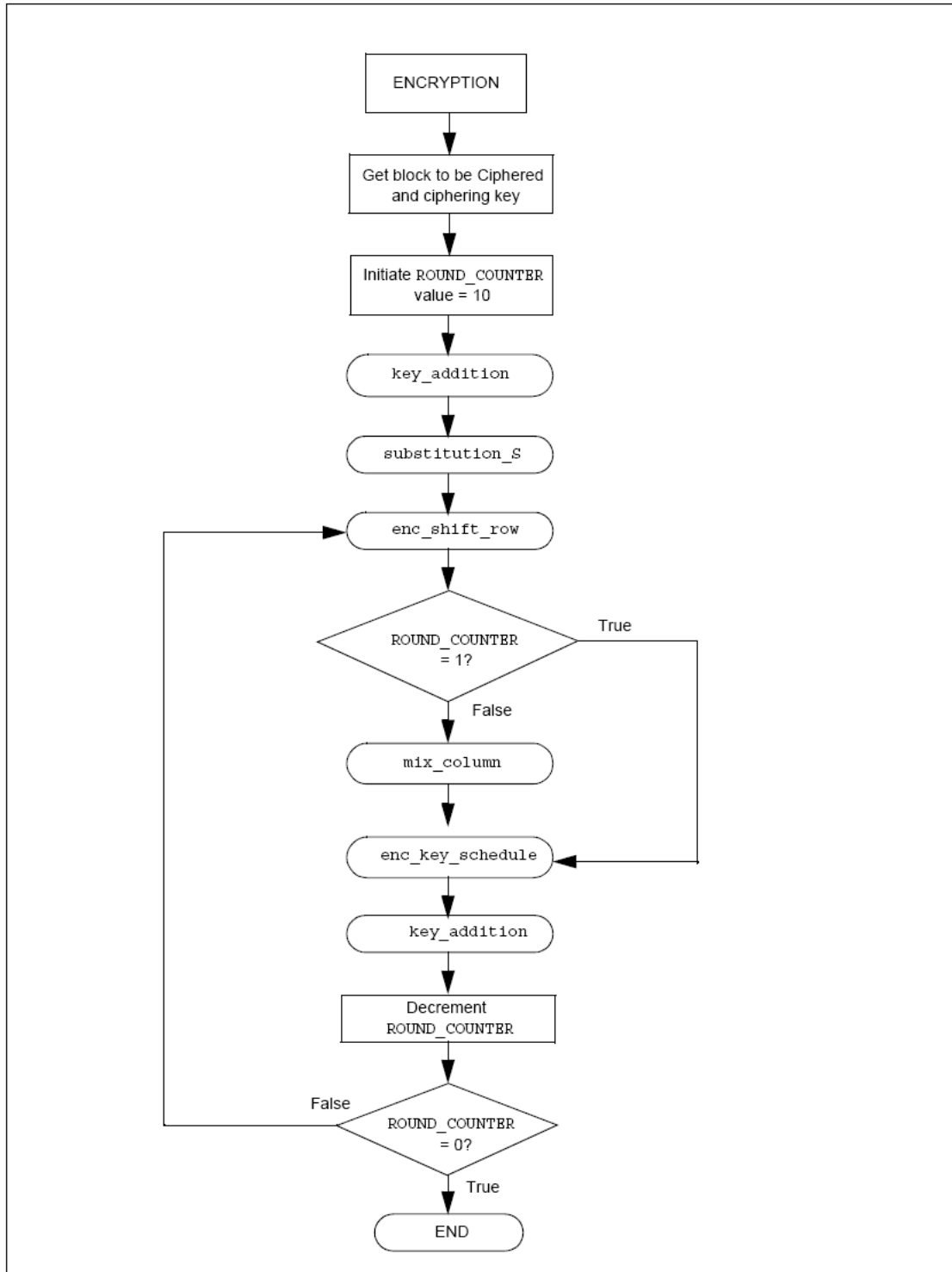
In the AddRoundKey step, the subkey is combined with the state. For each round, a subkey is derived from the main key using Rijndael's key schedule; the subkey is the same size as the state. The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR.



each  
each  
The  
byte  
byte

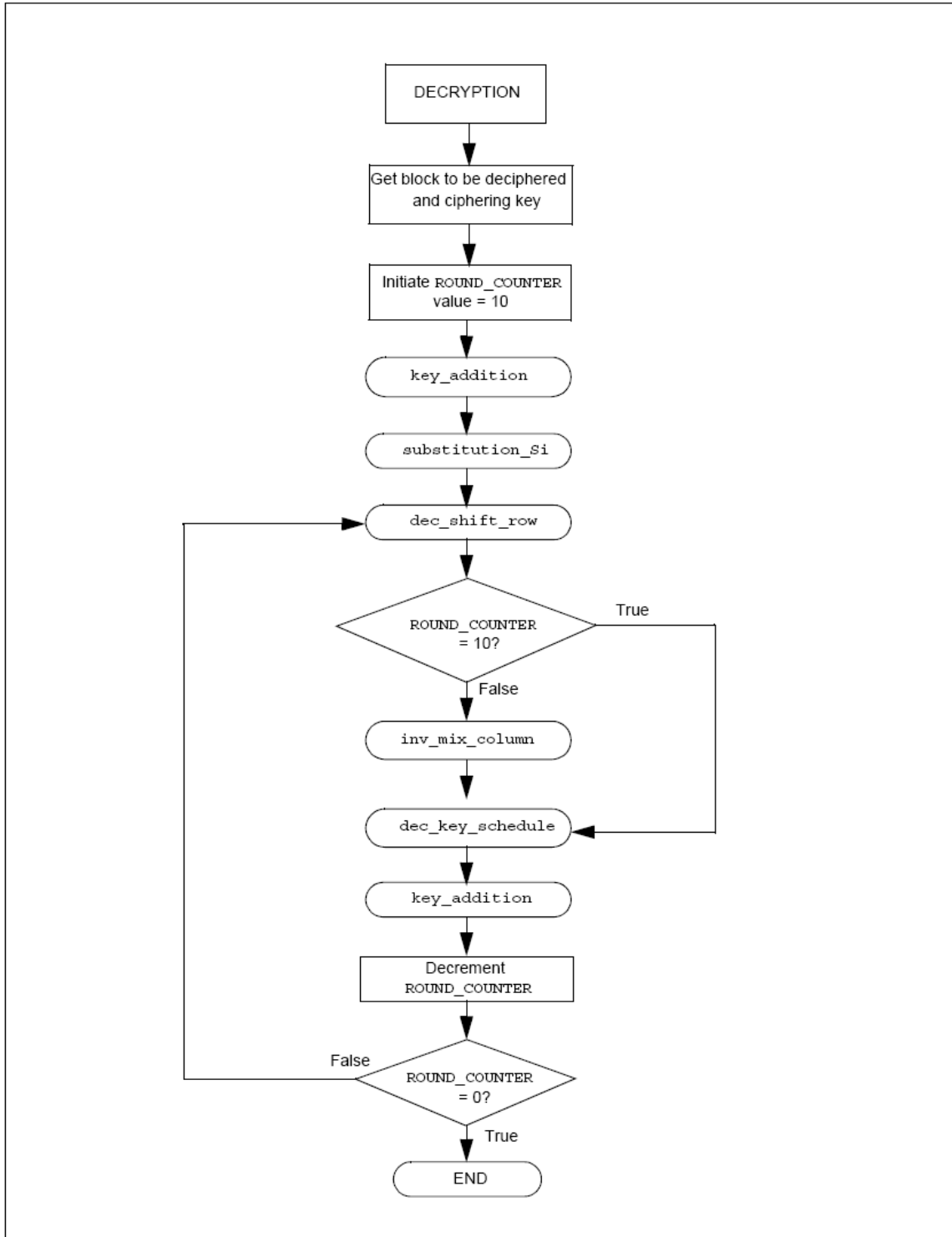
9.4.3.3 Encryption and Decryption AES flow chart <sup>12</sup>

Encryption flow chart



<sup>12</sup> Reference: Microchip AN821 Advanced Encryption Standard Using the PIC16XXX page:18, 19

### Decryption flow chart



#### 9.4.3.4 AES program (C programming)

See [IAP\_ECS]

# 10 Scientific Program

## 10.1 Data Collection from Ground station<sup>13</sup>



Samir Mourad

### The SRWD project

The radio astronomical IAP project supernova radio wave detector and analyzer (SRWDA) aims to detect and analyze radio signals from supernovas. At the base station a set of antennas, which gives also direction information of the signals, is aimed to be connected to a computer which acts as Software Defined Radio (SDR). Afterwards an analyzing program is aimed to be installed.

Later the detectors are planned to be installed on satellites to improve the resolution and to suppress disturbing signals from earth stations.

Master Thesis

### Initial Prototype for a base station for supernova radio wave detector and analyzer (SRWDA)

			External I/O DLL		I	
					Q	
System of antennas to get directional information	Coaxial adapter for each antenna	DAB+ USB Stick for each antenna	To connect the 4 antenna array correctly to the HSDR program	The free software defined radio HSDR	I/Q interface (normally to soundcard)	SRWDA Analyzer Analysis software package
<a href="http://www.amazon.de/dp/B007E9095">http://www.amazon.de/dp/B007E9095</a>	<a href="http://www.amazon.de/dp/B007E9095">http://www.amazon.de/dp/B007E9095</a>	<a href="http://www.amazon.de/dp/B007E9095">http://www.amazon.de/dp/B007E9095</a>				

### Detailed description and working plan

2 weeks -> Chapter 1 of Master Thesis	Literature research (radio astronomy, SDR) and specification of master thesis
4 weeks -> Chapter 2 of Master Thesis	Installing the development environment: Visual C++ 2008 Express with qt 4.7 (1 week)  Initial installation of HSDR with one antenna (ExtIO is the free program zadeg.exe) and introductory learning of HSDR program package (1 week)  Short introduction to winrad open source program code to get an understanding of the more complex HSDR code (1 week)  Introduction to communication technology (e.g. base band, mixing, I/Q signal) (1 week)
4 weeks -> Chapter 3 of Master Thesis	Migrating existing External I/O DLL ExtIO_Demo.dll from <a href="http://www.hdsdr.de">www.hdsdr.de</a> , which is suitable for one antenna to an external IO DLL which is suitable for the antenna array  Testing the ExtIO_Array.dll with the four antennas and HSDR
1 week -> Chapter 4 of Master Thesis	Determination of Requirements for SRWDA Analyzer (in cooperation with IAP supervisor)
9 weeks -> Chapter 5 of Master Thesis	Design and Implementation of SRWDA Analyzer
2 weeks -> Chapter 6 of Master Thesis	Test of the system with moving sending objects (a sender from a moving car)
4 weeks	Final writing and correction
Sum: 26 weeks (= 6 months)	

### Keywords:

Radio Astronomy, Software Designed Radio, Communication Technology, Visual C++, qt

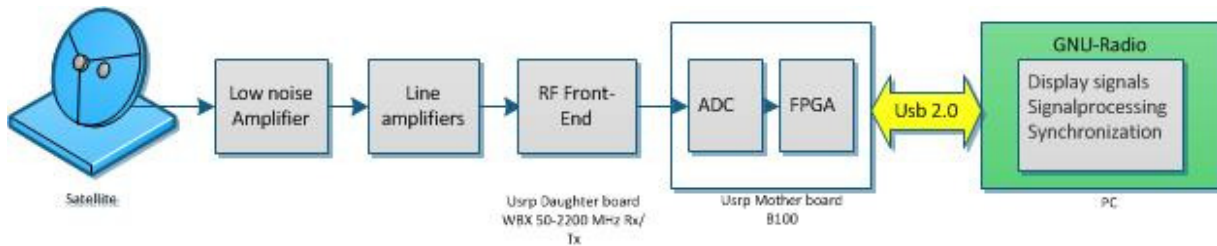
### Contact:

Dipl.-Ing. Dipl.-Inform. Samir Mourad

Email: [samir.mourad@aecenar.com](mailto:samir.mourad@aecenar.com)

Mobil: 0176 93516187 or 00961 76341526

### 10.1.1 System Design



<sup>13</sup> From [Kassar 2014].



## 10.2 Satellite Based Radio Astronomy Sensor

Migration of Ground Based Sensor of Suheeb Kassar Master Thesis to a Satellite based system.

## 11 Laboratories



Figure 4.2: Satellite Integration Laboratory, view from the “airlock” area (Image: IRS/Univ. Stuttgart)



Figure 4.3: Mission Control Center, view on one working place (Image: STW, IRS/Univ. Stuttgart)

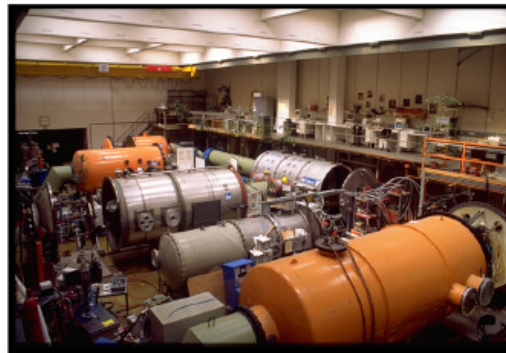


Figure 4.4: IRS vacuum chamber and plasma wind tunnel laboratory (Image: IRS/Univ. Stuttgart)

## Literature

[Kassar 2014] Suheeb Kassar, *Prototype for a base station for supernova remnant HI line radio wave detector and analyzer (SRWDA)*, IAP, AECENAR / KIT, Karlsruhe/Ras Nhache, April 2014 (planned)

**Space Mission Analysis and Design**, W. Larson, J. Wertz, Kluwer, 1999

**Elements of Spacecraft Design**, C.D. Brown, AIAA, 2002

**Space Vehicle Mechanisms**, P. Conley, New York, 1998

**Spacecraft Structure and Mechanisms**, T.P. Sarafin, Kluwer, 1995

**Raumfahrtssysteme: eine Einführung mit Übungen und Lösungen**, E. Messerschmidt ; S. Fasoulas. - Berlin u.a.: Springer, 2000. 533 S.

**Handbuch der Raumfahrttechnik**, W. Hallmann; W. Ley, München, Wien, Hanser 1999, 792 S.

[Laufer 2009] Lunar Mission BW1: Scientific Objectives and Small Satellite Concept, Rene Laufer, 2010

[Mourad et. al. 2006] Mourad S.(Ed.): Mourad S., Forstmann S., Subhan M., E.J., Farkh R., Abdelhaq N., Faquir M., Mikou M.Y., Ammar A., Flight Control System for an Wind Data Measuring Airship (Project Report for 2000 - 2006), Karlsruhe, 2006

[Nawaz et al. 2005] Investigation and Optimization of an Instationary MPD Thruster at IRS, IEPC-2005-208, *Presented at the 29<sup>th</sup> International Electric Propulsion Conference, Princeton University, October 31 – November 4, 2005*, Anusheh Nawaz\*, Monika Auweter-Kurtz, Georg Herdrich and Helmut L. Kurtz. *University of Stuttgart, Institut für Raumfahrssysteme, 70550 Stuttgart, Germany*

[Böhrk et. al. 2007] Elektrostatische Sonden im I-MPD Simp-LEX, Hannah Böhrk, Anusheh Nawaz, Uwe Bauder, Bastian Steiner, Oliver Troll, IRS Stuttgart, 2007

[Inalhan 2009] Applied Space Education and Research at ITU CAL Assoc. Prof. Gokhan Inalhan, Istanbul Technical University, Faculty of Aeronautics and Astronautic, Director of Controls and Avionics Laboratories

[Thirion] Pierre THIRION, Design and Implementation of On-board Electrical Power Supply of Student Nanosatellite OUFTI-1 of University of Liège (Master Thesis)

[IAP\_ECS] Emergency Control System, IAP 2013

[IAP\_SBEP] Zohby, Mahmoud; *Concept for Stratospheric Ballon Control System*, IAP, 2013

[Maral et. al. 2009] SATELLITE COMMUNICATIONS SYSTEMS - Systems, Techniques and Technology, Fifth Edition, Gerard Maral, Ecole Nationale Superieure des Telecommunications, Site de Toulouse, France, Michel Bousquet, Ecole Nationale Superieure de l'Aeronautique et de l'Espace (SUPAERO), Toulouse, France, Revisions to fifth edition by Zhili Sun, University of Surrey, UK, with contributions from Isabelle Buret, Thales Alenia Space, 2009