بسم الله الرحمن الرحيم

IAP-SAT

2st Project Report (2014)

Embedded Test and Simulation Environment for A Muster Earth Surveillance Sattelite Prototype

Initial document: Ras Nhache, 20 July 2014

Last update: 11.03.2015

A member institute of AECENAR
www.aecenar.com/institutes/iap

Content

Abstract

IAP
INSTITUTE FOR
ASTROPHYSICS

IAP - Institute for Astronomy and Astrophysics

AECENAR
Association for Economical and Technological Cooperation in the Euro-Asian and North-African Region
www.aecenar.com

مركز ابحـــاث لعلـــم الفلـــك و
فيزيـــاء النجـــوم

Satellite System for Radio Astronomy

IAP_SAT

Last update: 17 December 2013

Electric Propulsion for orbit and attitude stabilization

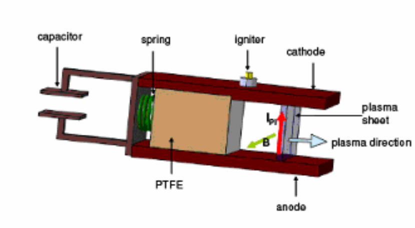
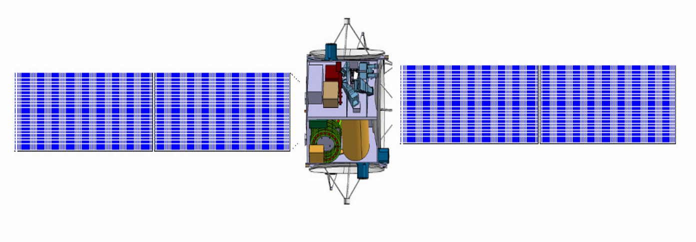
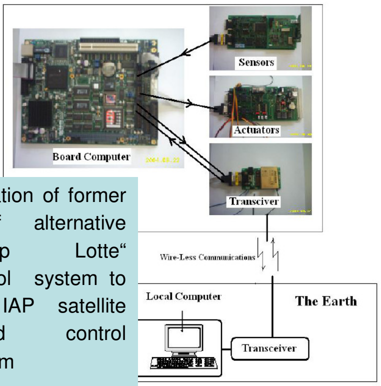


Figure 1. Working Principle of an I-MPD

First Mockup Model



Detection of HI Lines in Interstallar Media



Migration of former „VaEf alternative airship Lotte“ control system to the IAP satellite based control system

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 thesises)
4.500 USD material

Contact

Samir Mourad

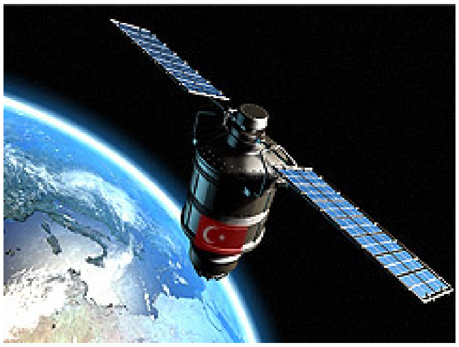
Mobile Lebanon ++961 76 341 526

Mobile Germany ++49 (0)176 4039 2038

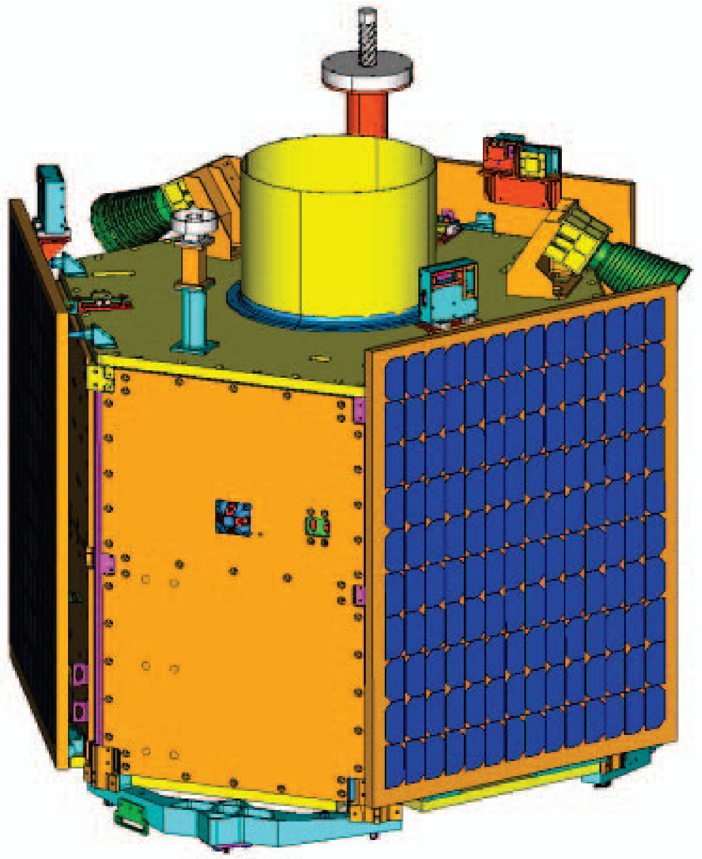
Email: samir.mourad@aecenar.com

1 Mission Overview for IAP-Sat

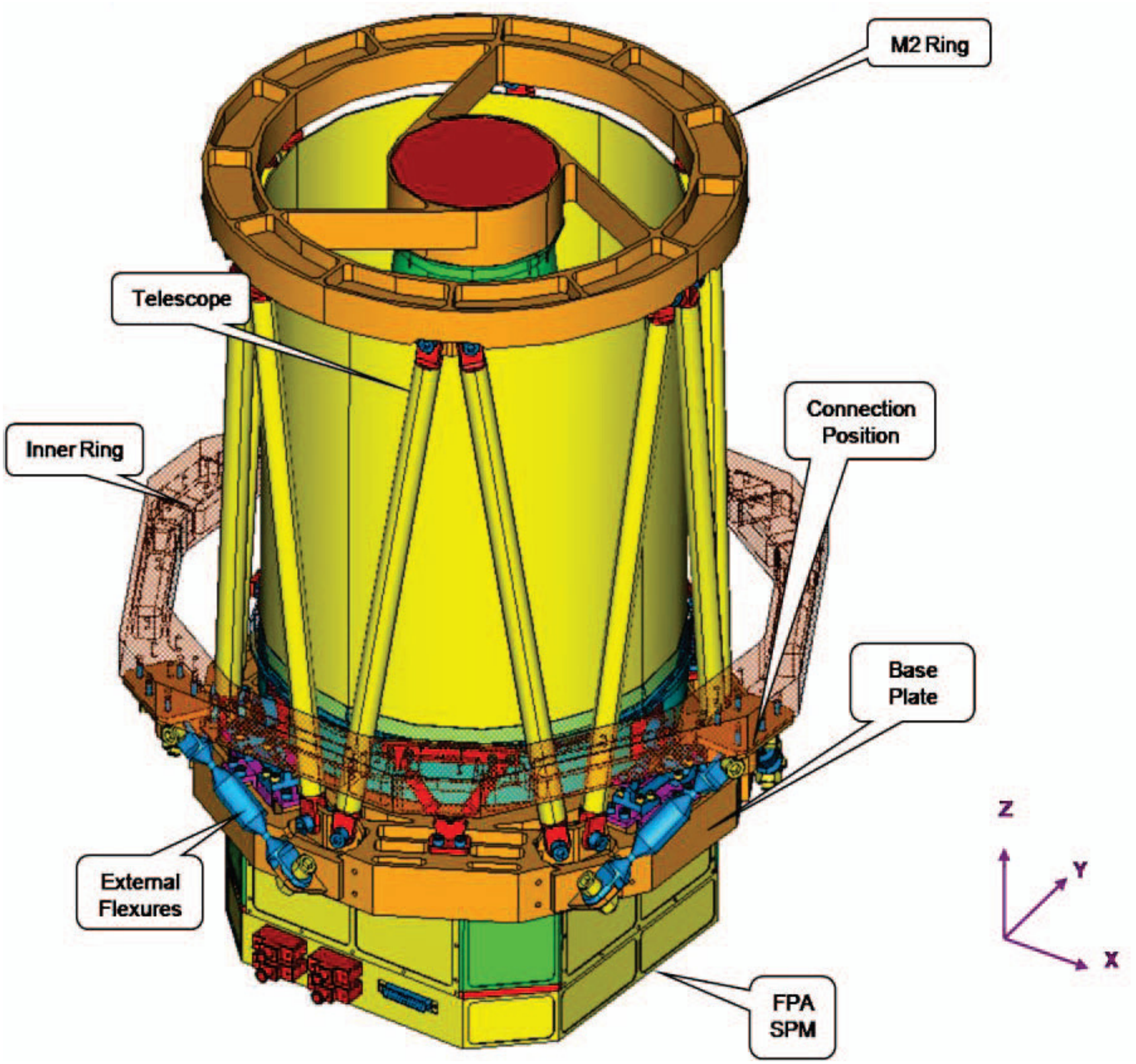
The satellite shall be based on technology of Turkish Göktürk and DubaiSat-1



Göktürk



DubaiSat-1



Dubai Medium Aperture Camera (DMAC)

1.1 Program Overview

1.2 Payload

2 Project Management

2.1 Organisation

1 Satellite Bus

1.1 FCS

1.2 Ground Station

with direction finder SDR tracking

2 Payload: Earth surveillance camera

()

3 Mission Planning

3.1 Access to space (Planning: IAP, Operation: a contractor)

3.2 Operation of the satellite

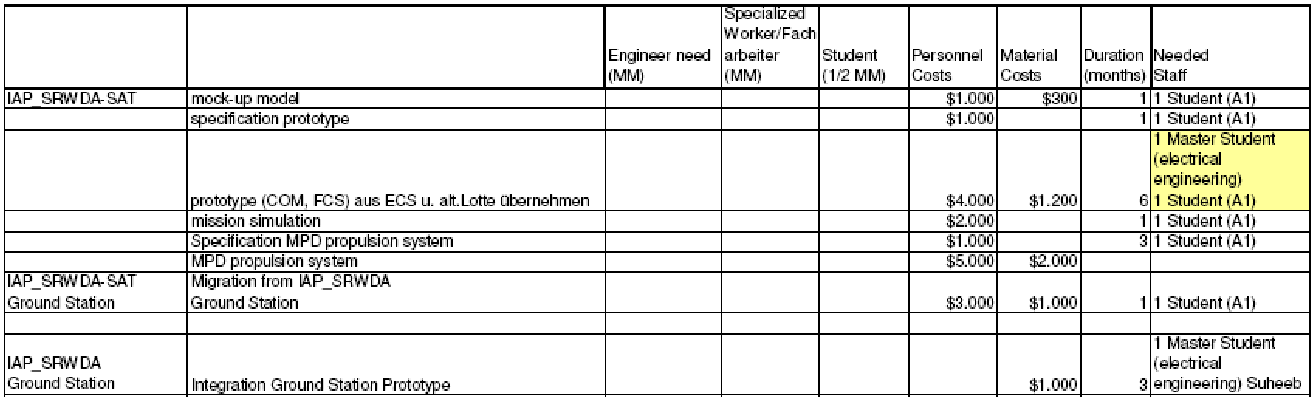
2.2 Working Packages and Time Plan

2.2.1 Working packages

WP No.	Working package content	Time span, material costs	Development environment (HW, SW)	Respon-sible	Status
1	Implementing Space Model, Sensor Model and Actuator Model (only Resistojet) in scicos Sources: M.Yassir Diss.	24.7.-7.8.	HP Server, scicos	A1M, MK, SM	
2	Star camera: Fixing on Orion **-> star coordination card in memory, Vergleich mit aufgenommenem Bild**	8.8.-31.8.		A1M, MK	
3	Resistojet **Thruster Prototype**	24.7.-7.8.	ProE	A2M, BM	
4	Development and Test Bench System Design	31.7-1.8.	Word	SM	

2.3 Costs

2.3.1 IAP 2014 (planned in January 2014)

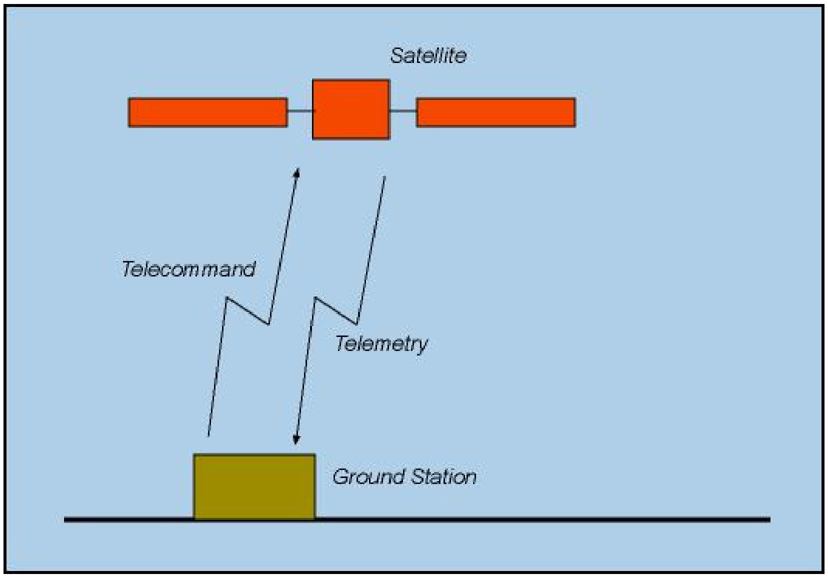


		Engineer need (MM)	Specialized Worker/Facharbeiter (MM)	Student (1/2 MM)	Personnel Costs	Material Costs	Duration (months)	Needed Staff
IAP_SRWDA-SAT	mock-up model				$1.000	$300	1	1 Student (A1)
	specification prototype				$1.000		1	1 Student (A1)
	prototype (COM, FCS) aus ECS u. alt.Lotte übernehmen				$4.000	$1.200	6	1 Master Student (electrical engineering) 1 Student (A1)
	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

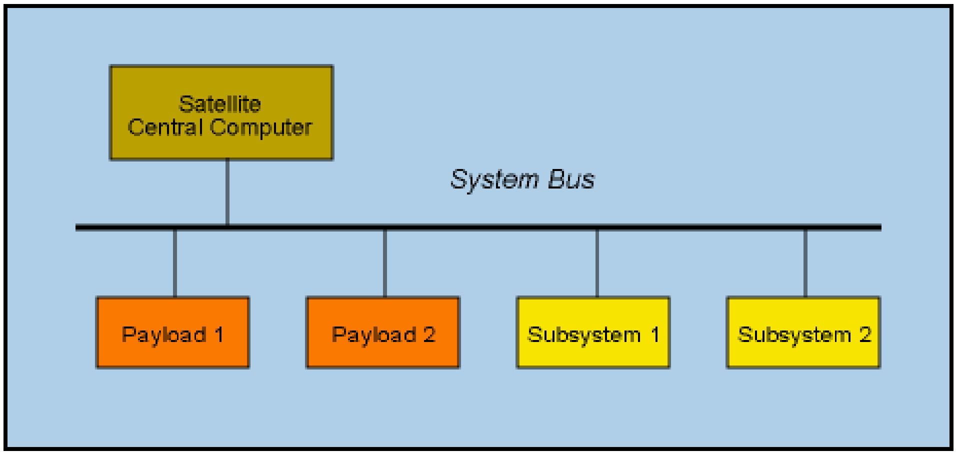
Sum: 22.500 $

3 Basics

3.1 Satellite System Structure



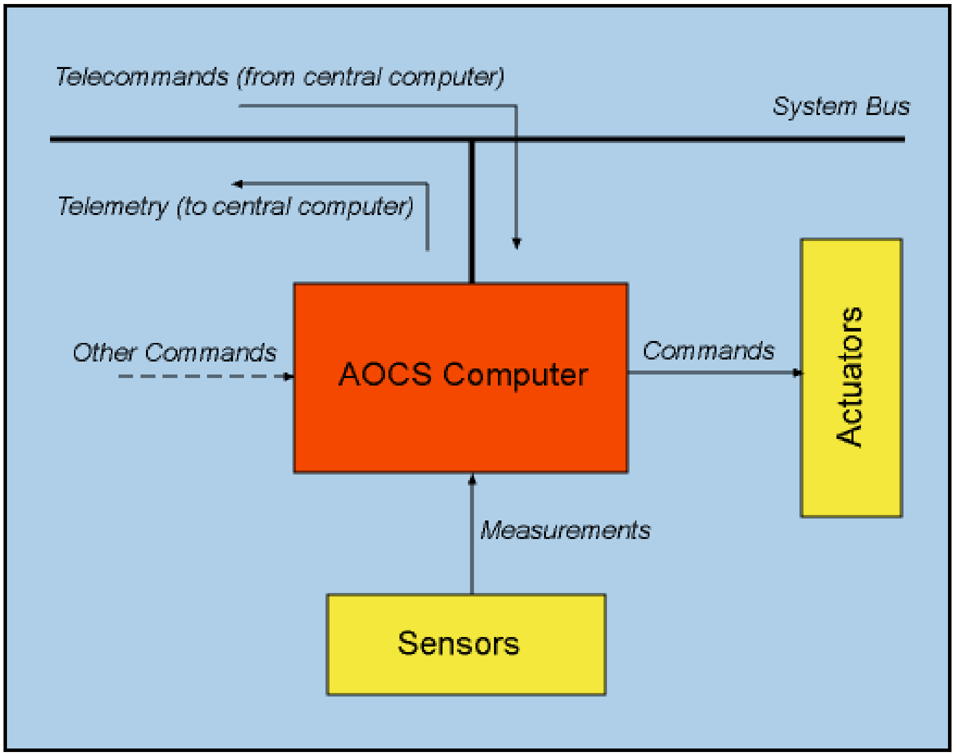
The figure at the side shows a "black box" view of a satellite system. The satellite is controlled from a *ground station* through a radiofrequency link. Depending on the mission, the link with the ground station may be continuous or extend only over a section of the orbit.



The figure at the side shows a typical internal architecture of a satellite system.

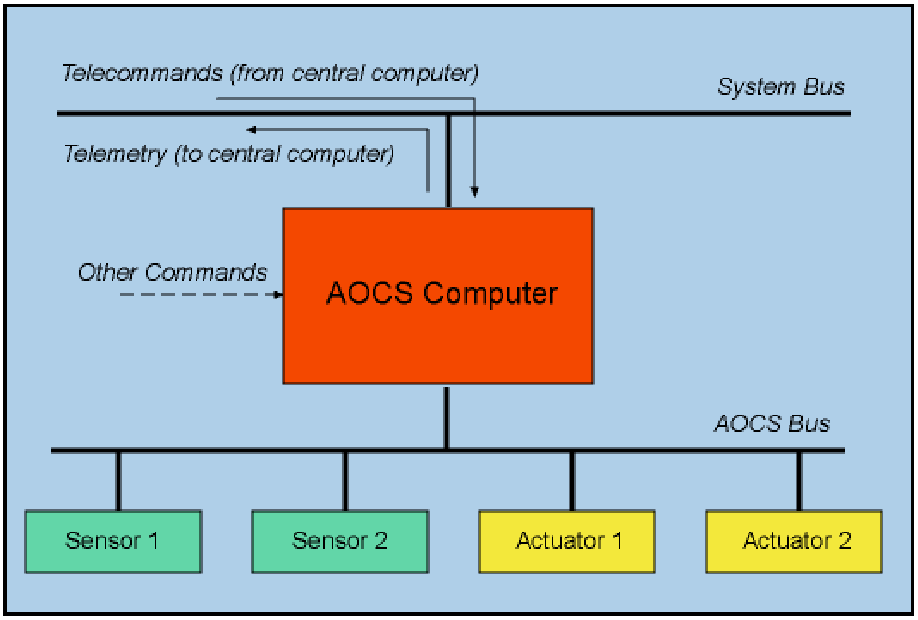
3.2 AOCS Subsystem Structure

The AOCS subsystem is one of the subsystems of a satellite. Its main function is to control the attitude and the orbit of the satellite.



The figure at the side shows an AOCS subsystem with its external interfaces. The figure assumes an AOCS with a dedicated computer separate from the central satellite computer. In an alternative configuration, the AOCS is hosted on the central computer.

The AOCS receives telecommands from the central computer and it forwards telemetry data to it over the system bus. It periodically acquires measurements (on the satellite attitude and orbital position) from a set of sensors and uses them to compute control signals that are sent to a set of actuators. Other commands, for special contingencies (such as for instance forcing the AOCS into safe mode), may be routed over dedicated command lines.



Because of its complexity, the AOCS is often organized around a dedicated AOCS bus over which communications between the AOCS computer and the AOCS sensors and actuators take place. The resulting structure is shown in the figure at the side.

SIMULATION ENVIRONMENT

An essential requirement in order to develop and verify various attitude control algorithms is the availability of space environment model. Such a tool provides necessary infrastructure at the preliminary stage to examine and validate the algorithms prior to hardware implementation.
A simulator was set up to develop and verify attitude control algorithms for the FLP.
The simulator was developed in Matlab/Simulink environment. The top level diagram of the simulator is shown in the Figure 7.1. The structure of the simulator is kept close to the overall structure of ACS algorithms. The yellow blocks in the top level diagram represents the space environment along with the ACS hardware. The blocks in the blue color contain all the functions that have to be implemented on the FPGA-based OBC and the blocks in the green color show the user interface which is required to initialize the inputs and monitor the output of the simulator.
A fabulous support provided by EADS Astrium in a form of Astrium STARS library tool box must be acknowledged here. This tool box contains several key functions which not only helped but also accelerated the development of the space environment.
Each block shown in the top level diagram is discussed briefly in this chapter to provide an overview of the simulator. The blocks and sub-blocks of this simulator are well documented in [44] and [45]. The main blocks of the simulator are

- Space environment

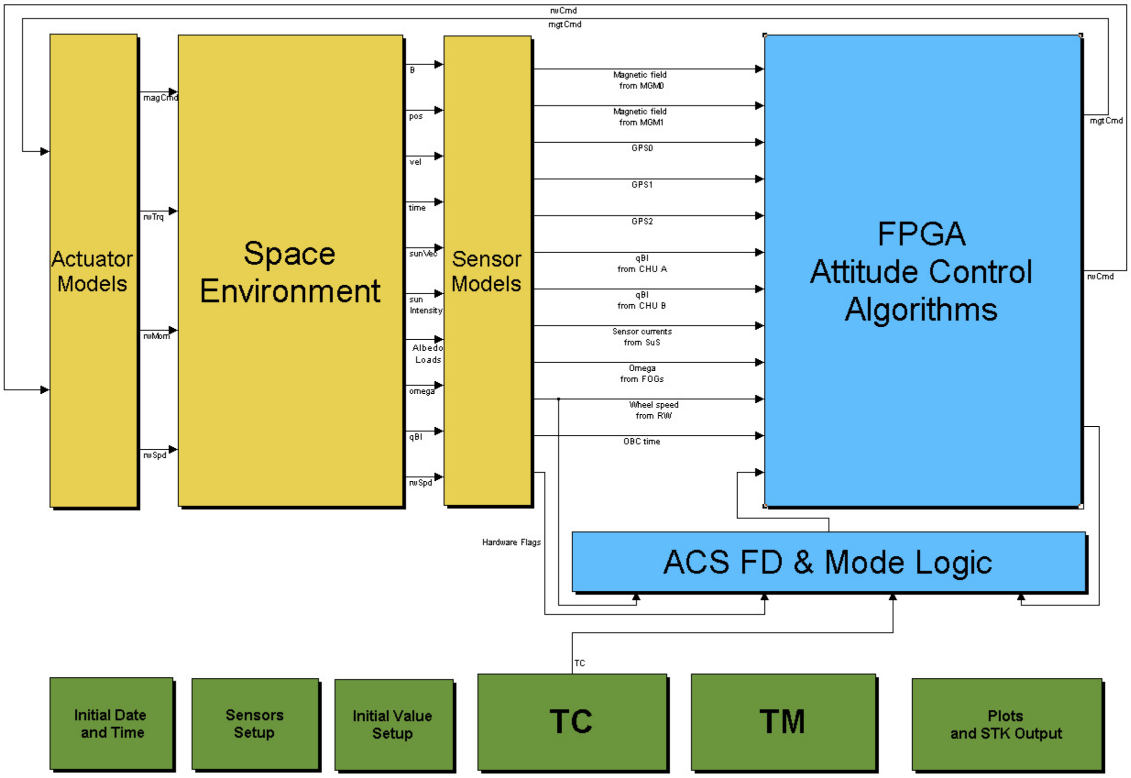


Figure 7.1: Top level diagram of the simulator

- Sensor models
- Actuator models
- User interface
- ACS algorithms
- ACS FD and mode logic

Space environment

This block emulates the space environment for the simulation. The core of this block is the orbit generator which connects all the other blocks. The main blocks in space environment are shown in the Figure 7.2 and are listed below.

1. Epoch clock
2. Orbit propagator
3. Spacecraft dynamics and attitude block
4. Earth's magnetic field model

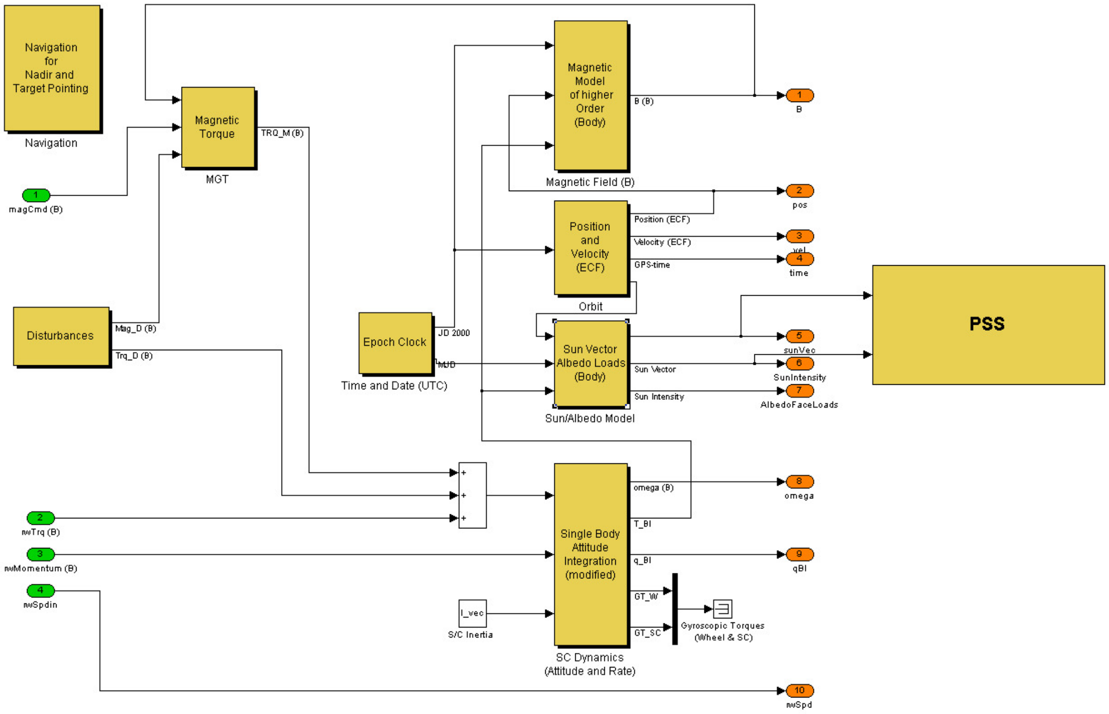
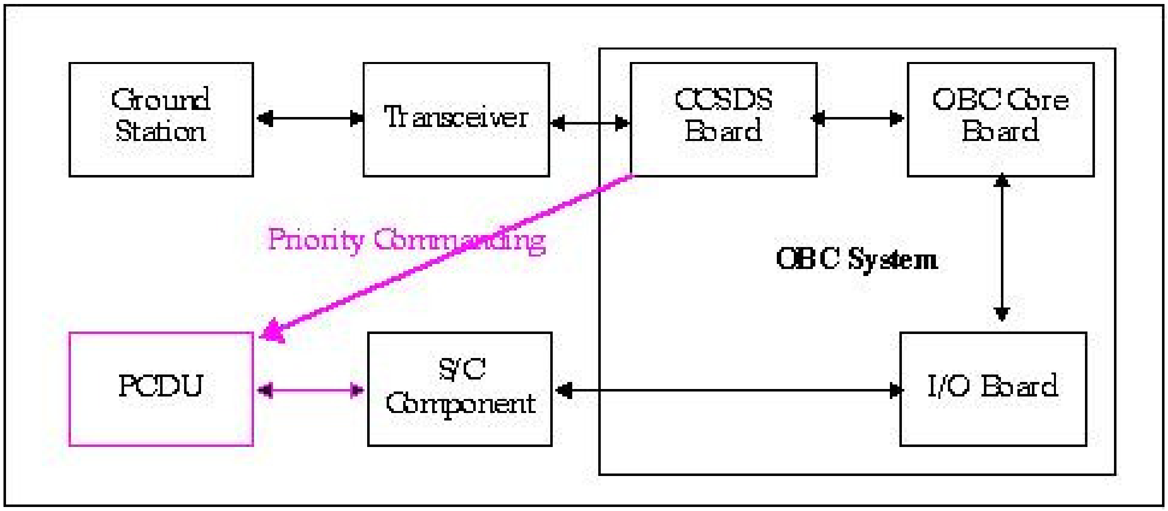
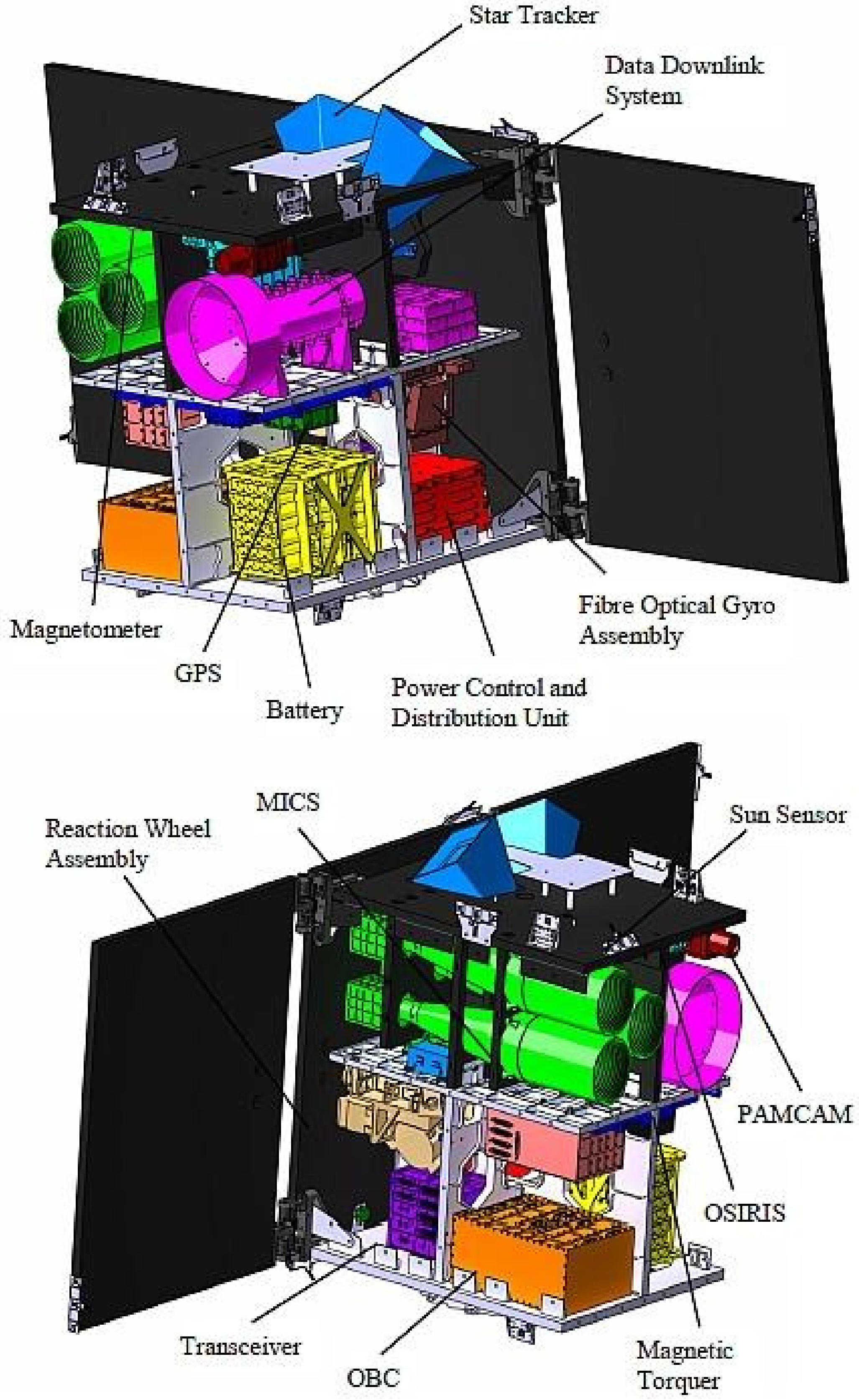


Figure 7.2: Space environment

5. Sun vector and eclipse flag generator
6. Modeling albedo effect
7. Power subsystem (battery) model

3.3 Flying Laptop

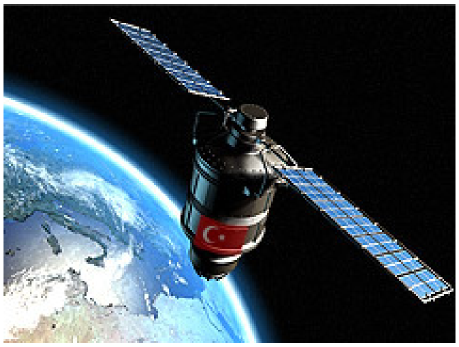

Ground Station
Transceiver
CCSDS Board
OBC Core Board
Priority Commanding
OBC System
PCDU
S/C Component
I/O Board


Star Tracker
Data Downlink System
Magnetometer
GPS
Battery
Power Control and Distribution Unit
Fibre Optical Gyro Assembly
MICS
Reaction Wheel Assembly
Sun Sensor
PAMCAM
OSIRIS
Transceiver
OBC
Magnetic Torquer

3.4 Göktürk project[1]

Göktürk – Project of Reconnaissance and Surveillance Satellite System

Developments in Space Technologies and TAF Satellite Need, as well as the developments experienced with regard to space since 1970s grew in 1990s and led to the emergence of new trends; visual satellites, in particular, became a part of our lives. Extension of usage area of visual satellites, and rise of the benefits gained thanks to innovations in technology have made the necessity of such systems in terms of military indisputable. Within this framework, TURAF initiated studies towards having satellite systems, so that TAF can take its part within this historical evolution process. In order to determine the technical specifications of the electro optical reconnaissance and surveillance satellite foreseen to be acquired in the first phase, satellite image requirements of TAF have been examined and the Air Force Command triggered the Project of Reconnaissance and Surveillance System, in the wake of the evaluations carried out.



Project Description

Within the scope of the Reconnaissance and Surveillance Satellite System named as GÖKTÜRK; a satellite which will be deployed on the low earth orbit (650-700 km) and which carries a very high-resolution Electro optical (E-O) camera and a fixed and mobile satellite earth station to form the earth section of the system will be obtained, with a view to satisfy the satellite image need of TAF for target intelligence, without any territorial water and air space restrictions and in daytime and under favorable weather conditions. In addition thereto, a Satellite Assembly Integration and Test (SAIT) center will be established within the scope of the project.

Technical specifications of Göktürk Project:

- **Electro optical camera system** on the satellite will be able to take 4-band color and black-white images.
- In order to make the best of sunlight, the satellite will be deployed on the **low orbit** (at 650-700 km altitude) compatible with the circular sun.
- It will have imaging capabilities in **spot, stereo, stripe** and **wide-area modes**.
- Communication can be established with the satellite, by means of the earth station based in Ankara, and the images taken by the satellite when it is out of the communication cone will be stored to be transferred to the earth station afterwards.
- In line with the orbit properties, images will be acquired across the world via satellite. On the orbit, a cycle of the satellite is approximately 98 minutes and it will complete 14 cycles per day around the world.

[1] last update: 2011, taken from homepage of Turkish Airforces

Earth Station

The following functions are intended to be created in the earth station to be built within the scope of the project;

- Satellite command control,
- Satellite tasking and image downloading,
- Image processing,
- Planning image requests, archive management and image distribution.

It is further intended that the Satellite Earth Station will have an expansion potential to support operation of the satellites that will take part in the future within the reconnaissance and surveillance satellite system architecture.

Usage Concept

Reconnaissance and Surveillance Satellite System will be used in line with the needs of TAF; in peace, crisis and war. In addition, meeting possible satellite image demands of Public Institutions and fulfilling possible image requests of friendly and allied countries within the scope of peacekeeping operation and/or antiterrorism are further projected.

Activity Calendar

The agreement concerning the project, procurement activities of which were initiated within the Undersecretariat for Defense Industries (UDI) in 2005, was executed on 16 July 2009. The satellite is envisaged to be placed into the orbit in 2013. With the reconnaissance and surveillance satellite system added in the inventory, the reconnaissance architecture will have gained substantial system ability and in the future, efficient imagery intelligence production ability will be obtained by using satellite images together with reconnaissance aircraft and unmanned aerial vehicles.

3.5 Star tracker system[2]

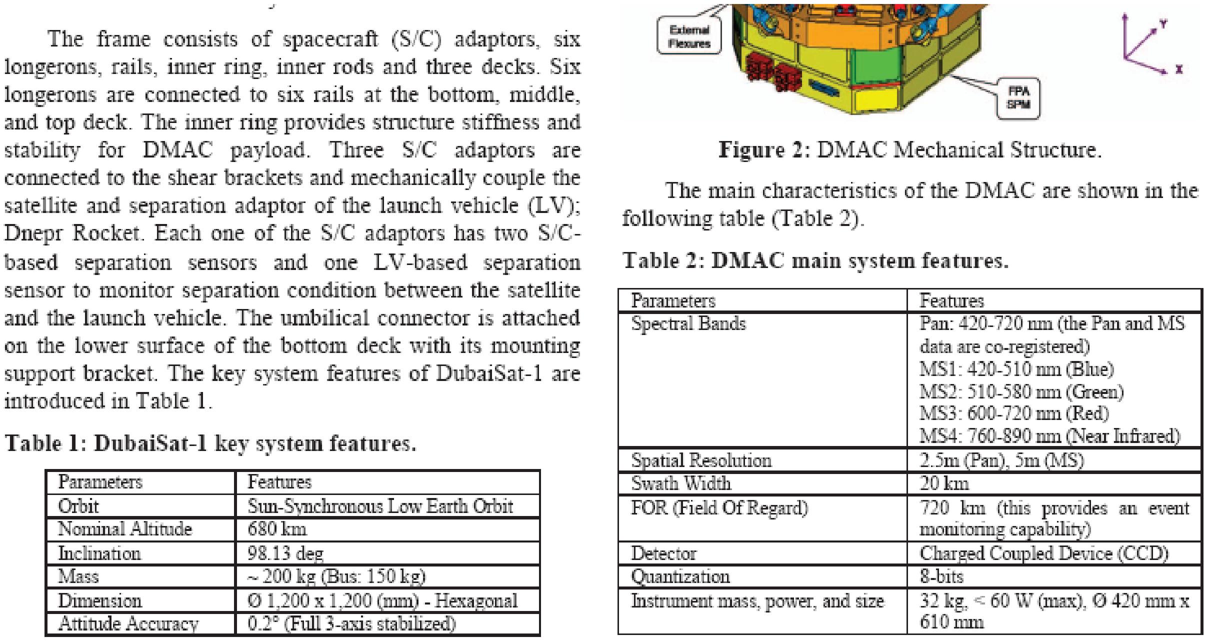
The star tracker system is employed in FLP to obtain the _ne attitude knowledge. The selected sensor is the micro-Advanced Stellar Compass (_ASC) developed by the Technical University of Denmark (DTU). The _ASC is the latest generation of the Advanced Stellar Compass (ASC) which was used in not less than 15 di_erent projects carried out by NASA, ESA, CNES, DLR and other leading space agencies. The _ASC is light in weight is more suitable for the small and micro-satellites.

The Star Tracker System of Flying Laptop consists of two camera head units (CHU), one micro-data Processing Unit (_DPU), and two ba_es as shown in the Figure 3.18. Two CHUs are included in this design to avoid the Star Tracker blinding and to allow a single failure in this system. The _DPU could drive up to 4 CHUs and its electronics is fully redundant. The _DPU uses 486 microprocessor to process the optical data acquired by CHU. It is equipped with 64 kB PROM, 8 MB _ash and 8 MB RAM. During normal operation, the _DPU creates a digital image of the

[2] From [Yassir 2010]

acquired analogue data received from the CHUs every 0.5 s and stores it in the internal RAM. In further computing, the lens distortion is compensated and the image is adjusted for adequately bright objects. The amount of bright objects that are found in an image can be influenced by varying a software

3.6 DubaiSat-1



The frame consists of spacecraft (S/C) adaptors, six longerons, rails, inner ring, inner rods and three decks. Six longerons are connected to six rails at the bottom, middle, and top deck. The inner ring provides structure stiffness and stability for DMAC payload. Three S/C adaptors are connected to the shear brackets and mechanically couple the satellite and separation adaptor of the launch vehicle (LV); Dnepr Rocket. Each one of the S/C adaptors has two S/C-based separation sensors and one LV-based separation sensor to monitor separation condition between the satellite and the launch vehicle. The umbilical connector is attached on the lower surface of the bottom deck with its mounting support bracket. The key system features of DubaiSat-1 are introduced in Table 1.

Table 1: DubaiSat-1 key system features.

Parameters	Features
Orbit	Sun-Synchronous Low Earth Orbit
Nominal Altitude	680 km
Inclination	98.13 deg
Mass	~ 200 kg (Bus: 150 kg)
Dimension	Ø 1,200 x 1,200 (mm) - Hexagonal
Attitude Accuracy	0.2° (Full 3-axis stabilized)

Figure 2: DMAC Mechanical Structure.

The main characteristics of the DMAC are shown in the following table (Table 2).

Table 2: DMAC main system features.

Parameters	Features
Spectral Bands	Pan: 420-720 nm (the Pan and MS data are co-registered) MS1: 420-510 nm (Blue) MS2: 510-580 nm (Green) MS3: 600-720 nm (Red) MS4: 760-890 nm (Near Infrared)
Spatial Resolution	2.5m (Pan), 5m (MS)
Swath Width	20 km
FOR (Field Of Regard)	720 km (this provides an event monitoring capability)
Detector	Charged Coupled Device (CCD)
Quantization	8-bits
Instrument mass, power, and size	32 kg, < 60 W (max), Ø 420 mm x 610 mm

3.7 Software-in-the-Loop (SiL)

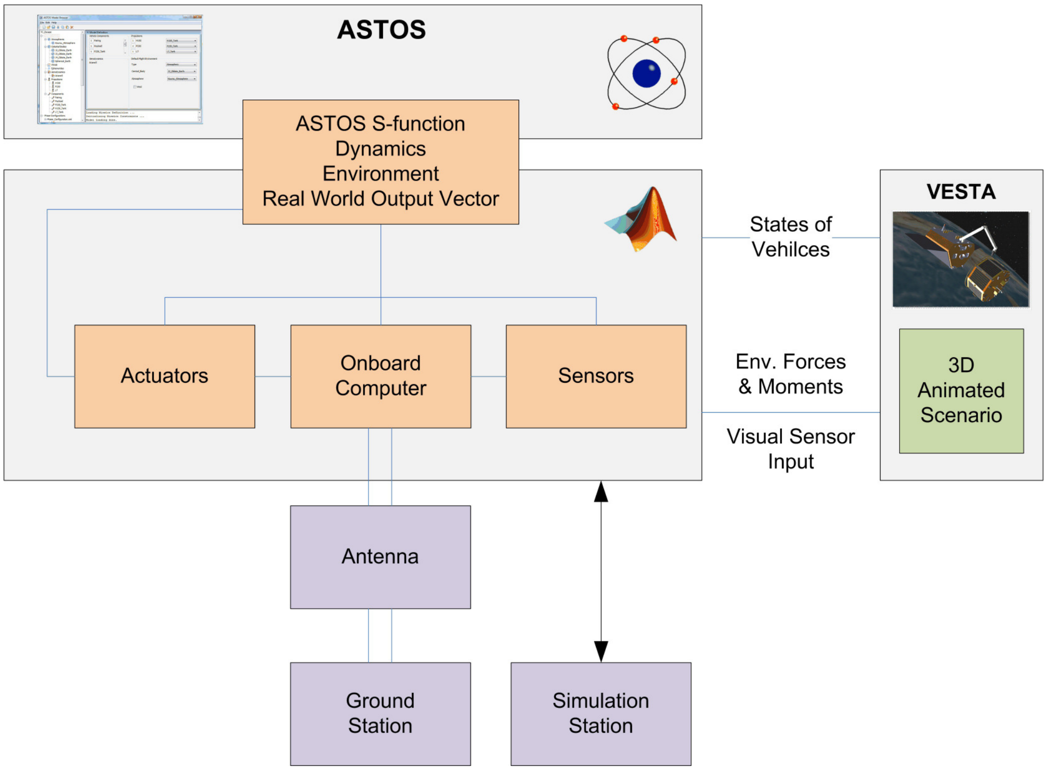
…

3.8 Hardware-in-the-Loop (HiL)

…

3.9 Candidate Software Tools

3.9.1 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.

4 Specification and Concept

4.1 Requirements

[SysReq1] It shall be made a test bench for a similar earth surveillance satellite as DubaiSat-1

[SysReq2] For simulation of the satellite behaviour in space: octave/scicos

[SysReq3] Actuators:

~~Magnetoplasmadynamic (MPD) thruster. It shall be a *Self Field MPD*.~~

[SysReq4] Sensors:

Star camera (

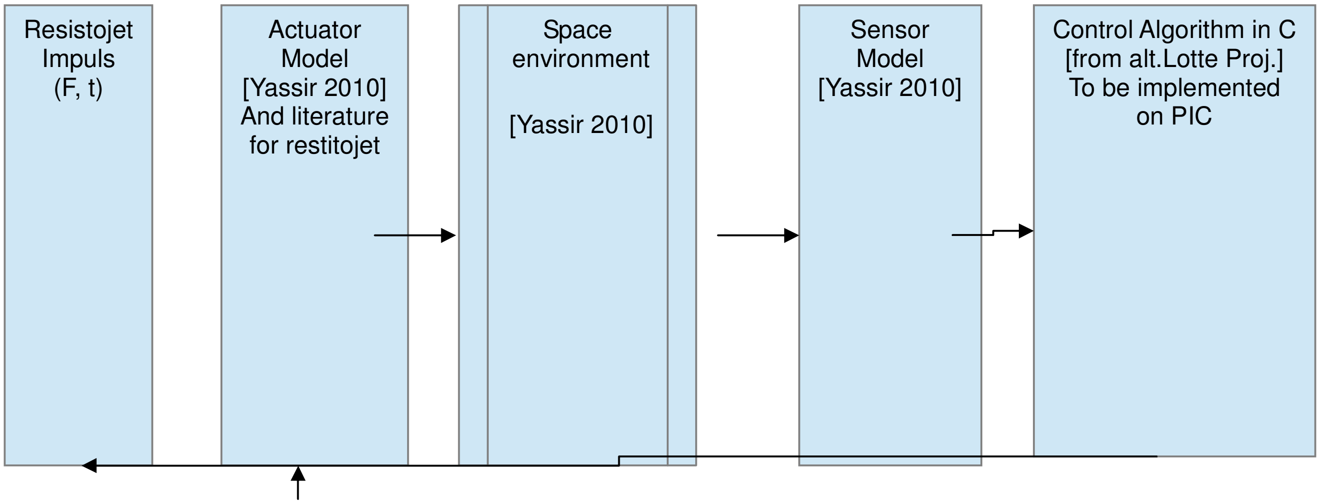
4.2 System Concept

Vorgehensweise:

alle bisherigen Projektbausteine benutzen

Sensors: camera

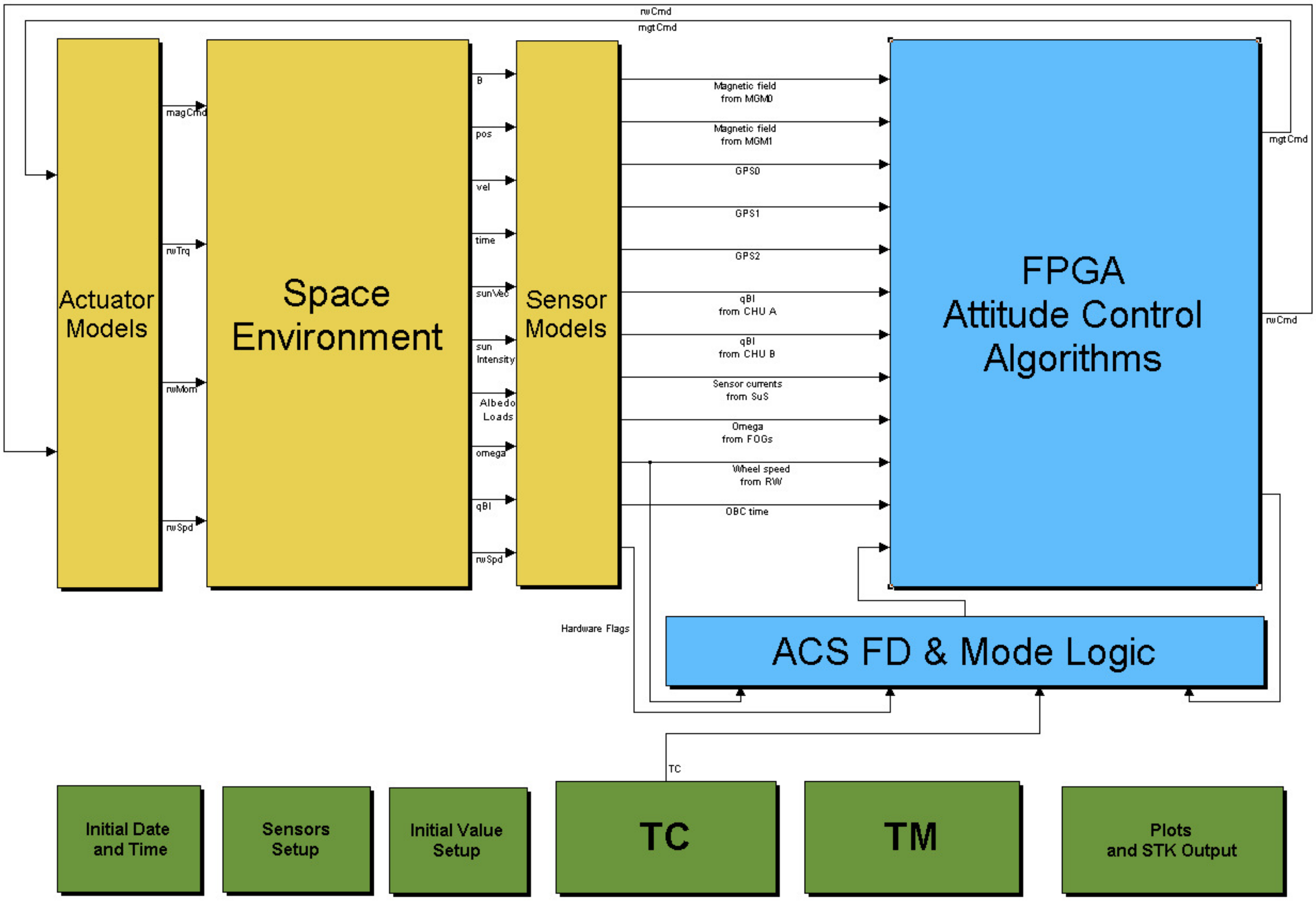
Actuators


Resistojet Impuls (F, t)
Actuator Model [Yassir 2010] And literature for restitojet
Space environment
[Yassir 2010]
Sensor Model [Yassir 2010]
Control Algorithm in C [from alt.Lotte Proj.] To be implemented on PIC

5 Development and Test Bench System Design[3]

5.1 ACS On-Board Algorithms Architecture

5.2 Simulation Environment



5.2.1 Space Environment

This block emulates the space environment for the simulation. The core of this block is the orbit generator which connects all the other blocks. The main blocks in space environment are shown in the Figure 7.2 and are listed below.

1. Epoch clock

2. Orbit propagator

3. Spacecraft dynamics and attitude block

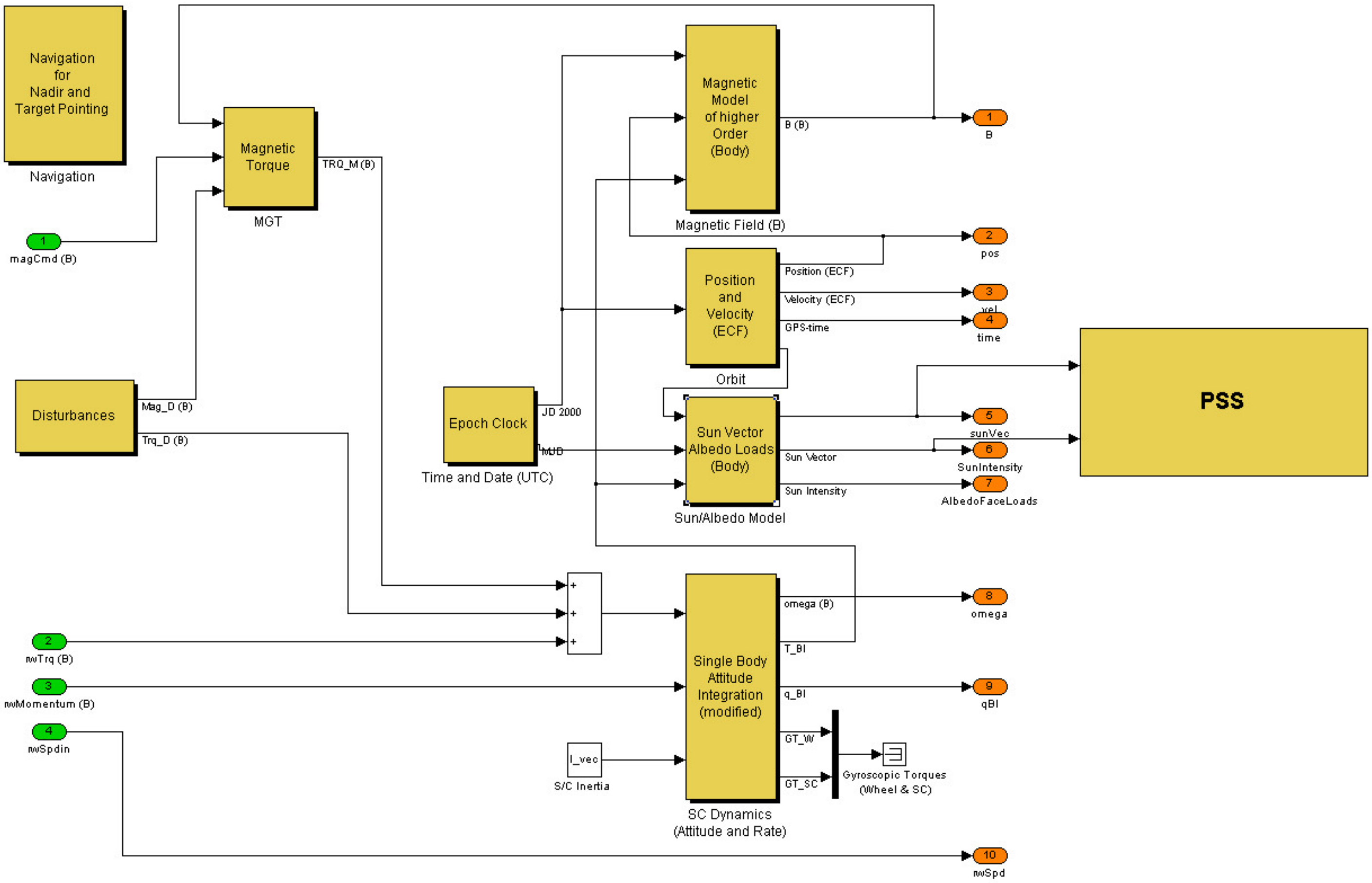
4. Earth's magnetic field model

5. Sun vector and eclipse flag generator

6. Modeling albedo effect

[3] Mostly based on [Yassir 2010]

7. Power subsystem (battery) model



5.2.2 Matlab testing interface

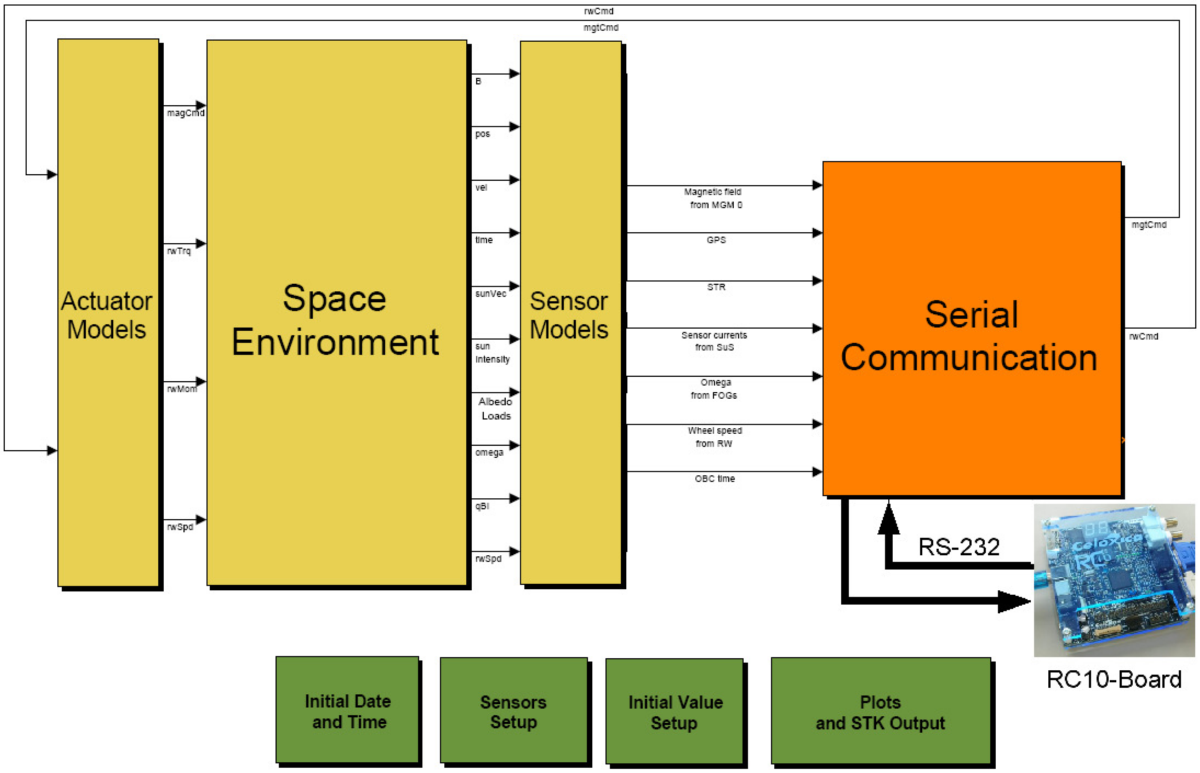
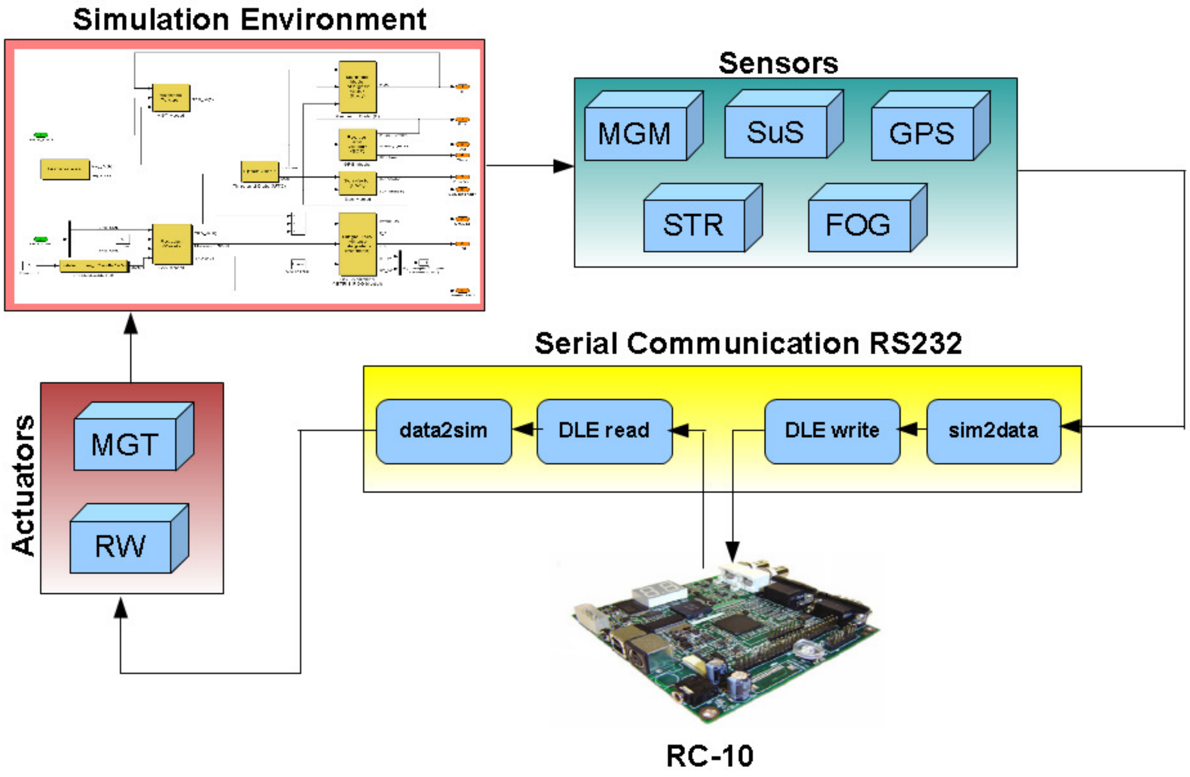


Figure 8.3: Top level diagram of testing algorithms with scicos



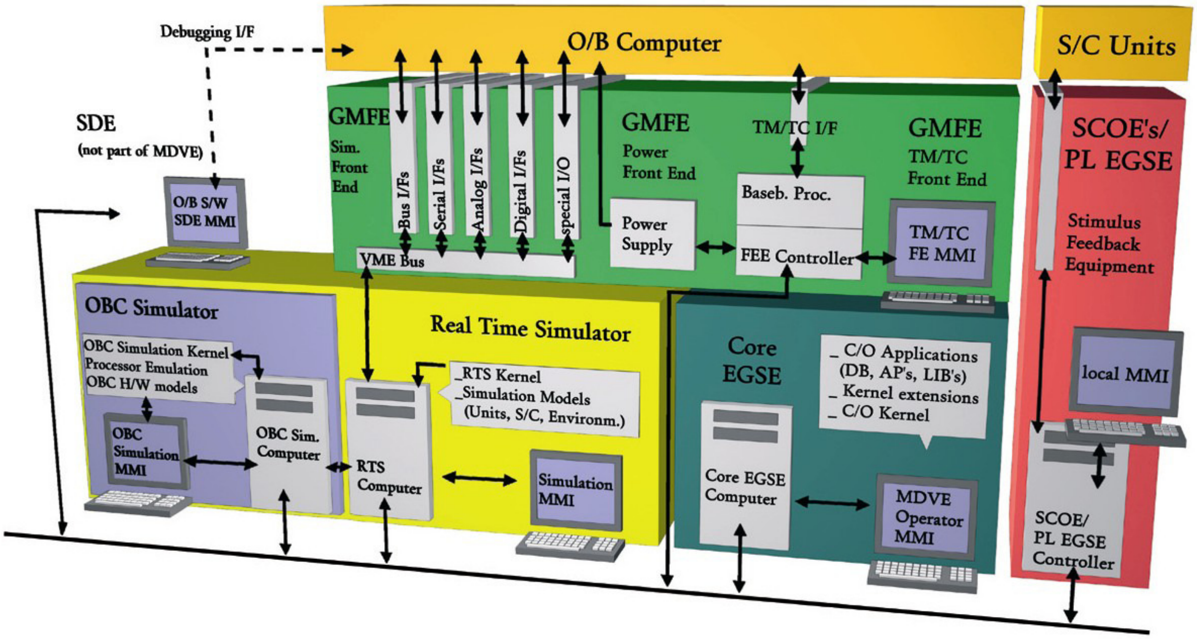


Figure 8.6: Model based development and verification environment

MDVE will be used in various project phases for a number of design, development and veri_cation tasks in di_erent working environments. The most important MDVE standard con_gurations are:

1. Development SVF (DEV-SVF)
2. Software Veri_cation Facility (SVF)
3. System Testbed (STB)
4. Extended Real-Time Testbed (FlatSat)
5. Spacecraft simulator for the mission control center.

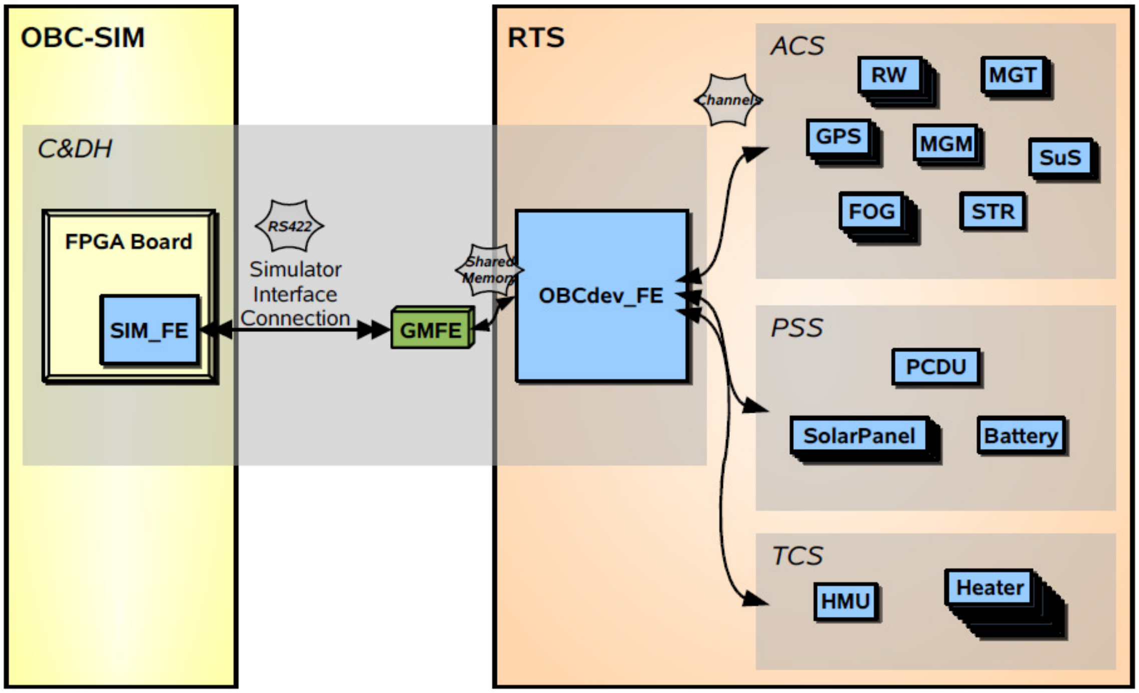


Figure 8.7: Simulator Interface with OBC

5.3 Kantenerkennung

I tried this tutorial: http://www.semipol.de/wp-content/uploads/sites/4/2008/07/tutorial.pdf

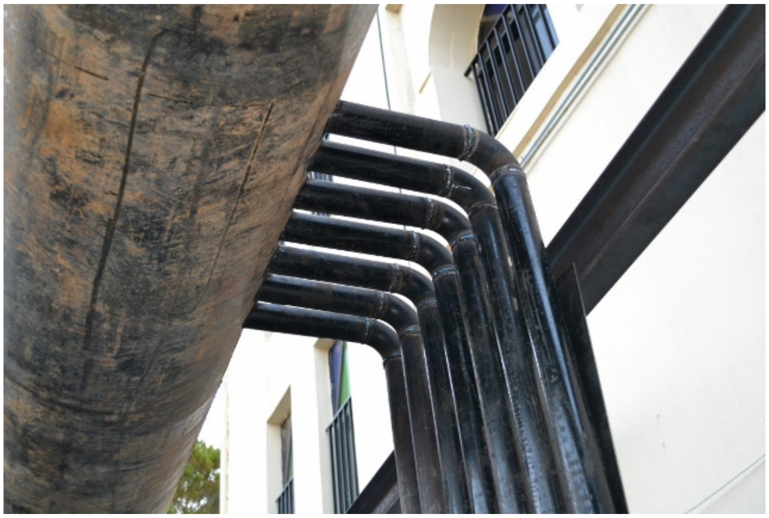
The code:

```
#include<opencv\cv.h>
#include<opencv\highgui.h>
# include <iostream >

using namespace std;


int main (int argc , char * argv []) {

 IplImage * img = cvLoadImage ( argv [1]) ;
 if (! img ) {
 cerr << " Could not load image file : " << argv [1] <<
endl ;
 exit ( EXIT_FAILURE );
 }
 IplImage * binaryImage = cvCreateImage ( cvGetSize (img),
IPL_DEPTH_8U , 1);
 cvSplit (img , binaryImage , NULL , NULL , NULL );
 IplImage * normalized = cvCreateImage ( cvGetSize (img), img
->depth , 1);
 cvEqualizeHist ( binaryImage , normalized );

// cvNamedWindow (" normalized ", CV_WINDOW_AUTOSIZE );
// cvShowImage (" normalized ", normalized );

IplImage * gauss = cvCreateImage ( cvGetSize (img), img ->
depth , 1);
 cvSmooth ( normalized , gauss , CV_GAUSSIAN , 13, 13);

// cvNamedWindow (" gauss ", CV_WINDOW_AUTOSIZE );
// cvShowImage (" gauss ", gauss );

 IplImage * canny = cvCreateImage ( cvGetSize (img), img ->
depth , 1);
 cvCanny (gauss , canny , 40, 130) ;

// cvNamedWindow (" edges ", CV_WINDOW_AUTOSIZE );
// cvShowImage (" edges ", canny );

 IplImage * temp = cvCreateImage ( cvGetSize (img),
IPL_DEPTH_8U , 1);
 IplImage * closed = cvCreateImage ( cvGetSize (img),
IPL_DEPTH_8U , 1);
 IplConvKernel * element = cvCreateStructuringElementEx (3,
3, 1, 1, CV_SHAPE_RECT );
 cvMorphologyEx (canny , closed , temp , element ,
CV_MOP_CLOSE );
 cvReleaseStructuringElement (& element );
 cvReleaseImage (& temp );

 cvNamedWindow (" closed ", CV_WINDOW_AUTOSIZE );
 cvShowImage (" closed ", closed );


//cvNamedWindow (" original ", CV_WINDOW_AUTOSIZE );
// cvShowImage (" original ", img );
 cvWaitKey (0);
 cvDestroyWindow (" original ");
 cvReleaseImage (& img );
return EXIT_SUCCESS ;
}
```

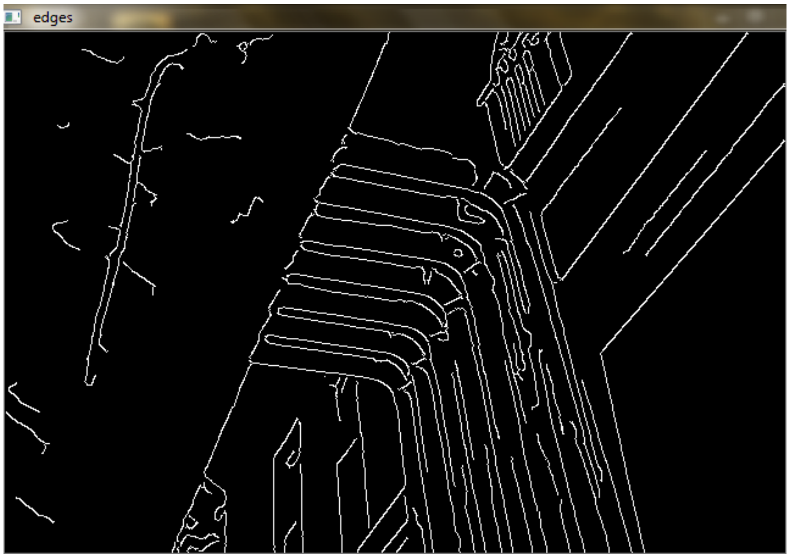
For compiling, this tutorial is helpful: https://www.youtube.com/watch?v=cgo0UitHfp8

example:

Original:

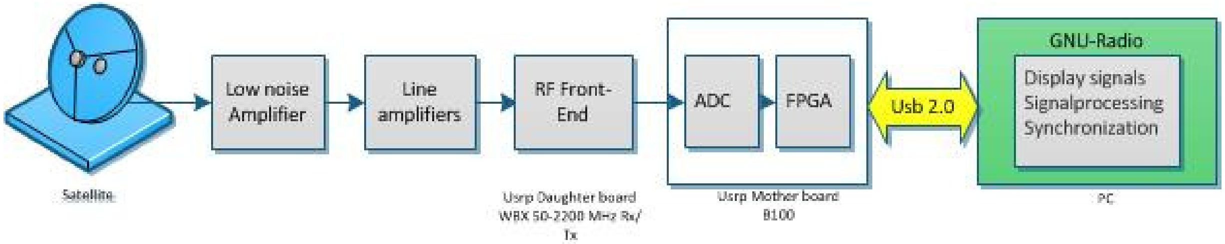


Output:



5.4 Prototype

6 Telemetry (Satellite Bus Communication) Prototype[4]



[4]

7 Payload Program

7.1 Earth Surveillance

7.1.1 System Design

8 Laboratories

Literature

[Yassir 2010] Muhammad Yassir, *Development and Implementation of the attitude control algorithms for the e micro-satellite Flying Laptop*, PhD thesis, IRS, University of Stuttgart, 2010