Design of Low-cost Telecommunications CubeSat-class Spacecraft

Adnane ADDAIM, Abdelhaq KHERRAS and El Bachir Zantou
Centre for Space Research and Studies, EMI
Morocco

1. Introduction

The CubeSat-class Spacecraft hardware has volume, mass, and power limitations more extreme than in other satellites. In fact, the CubeSat-class spacecraft is a Cube with dimensions 10cm x 10cm x 10cm, mass one kilogram maximum and the available power is about 2Watt by mounting the solar cells on the CubeSat structure.

The CubeSat-class spacecrafts have the advantages of being able to perform as a test bed for new core space technologies to be applied to space programs, for much lower cost, shorter schedule, and less risk. For this reason, the world leaders in space technology, including the US and Europe, are focusing their efforts on smaller satellites under the motto of "Faster, Cheaper, Better" that can perform missions traditionally assigned to large/medium satellites in the past.

In general, the conventional design of a satellite is based on the modular architecture where each satellite subsystem has hardware and software autonomy. In the case of a CubeSat-class satellite, this kind of architecture did not prove its effectiveness since a significant number of CubeSats could not make a success of their missions. The goal of this chapter consists in developing a new design methodology of the CubeSat-class satellites.

Each satellite includes many subsystems; each subsystem has a role to play aboard the satellite. In practice, in the field of the satellites design, each satellite subsystem, which has dedicated hardware and software, is assigned to one team of experts because each satellite subsystem belongs to one branch of science. On the other hand, there are connections between the various subsystems. Therefore, each team works in coordination with other teams to achieve the tasks of its subsystem. Now, the question is: Do we have legitimacy to integrate several subsystems in only one. The answer is simple. For the satellites larger than the Microsatellite, it is not conceivable to integrate several subsystems in only one. On the contrary, to ensure the correct operation of the satellite considering the importance of the mission and the significant budget allocated, we usually proceed to the redundancy of some key equipment aboard the satellite in spite of the complexity that adds.

In the case of CubeSat-class spacecraft, considered as the other extreme of the category of the small satellites, given the imposed constraints (available electrical power is about 2Watt, mass ≤ 1kg and volume of 10x10x10 cm3) and the restricted financing (university), we find all legitimacy to integrate several subsystems in only one. This approach will save energy, will reduce the mass and will save space aboard the satellite. The key of this integration is to

replace some hardware by software. For example, two hardware modems could be replaced by programming one Digital Signal processor (DSP).

In this chapter, we will describe in details the design of Low-cost Telecommunications CubeSat-class Spacecraft by using one DSP processor with multitasking operating system which integrates all the intelligences of the satellite. Due to the environment in space and the constraints of the CubeSat-class spacecraft, different measures had to be taken when the satellite was designed.

2. Satellite mission description

The mission of the Telecommunication CubeSat-class satellite is to provide various services, such as mobile localization of ships, message exchange between terminals and data collection from autonomous weather stations in inaccessible.

The Telecommunications satellite will have four modes: Store-and-Forward mode, Digipeater mode, Beacon mode and Command mode. It will use only one communication protocol, the AX.25 protocol, to communicate with the earth segment (terminals and the control station). This protocol provides Connection-Oriented and Connectionless transmissions. The connectionless transmission method will be used in digipeater and beacon modes, whereas the connection-oriented method will be used in store-and-forward and Command modes.

In store-and-forward mode, the terminals use the Slotted Aloha multiple access (Addaim et al., 2008) to send their packets to the satellite and wait for an acknowledgement from the satellite, which stores the correct packets in its on-board storage system, and delivers this to the destination terminal in a later time in the same original format. The packets are position-stamped and time-stamped. Between the storage and the retrieval of the packet, the satellite moves around its orbit and the earth rotates on its axis. These movements change the location of the satellite's footprint, and the satellite effectively carries the packets from terminal to terminal situated in any place all over the world. In this mode, terminals will not use the existing AX.25 firmware. Instead, they will use a stop and wait ARQ (Automatic Repeat Request) protocol which will be implemented only at build-in-house terminals (Zantou & Kherras, 2004). When the terminal transmits its packets, it will wait for acknowledgement and will set its timer. If this timer expires without reception of an acknowledgement, the entire packet is scheduled for retransmission after a random interval time until receiving the acknowledgement.

In the Digipeater mode, which is a real-time bent-pipe relay, all the radio amateur stations, situated all over the world, could transmit their packets without waiting for an acknowledgement. The received packet will be tested if there is an error in the packet. The erroneous packet will be rejected and the correct packet will be retransmitted after adding the satellite's callsign in the Address field of the AX.25 packet. This mode is useful when both the source and destination radio amateur stations are at the same satellite's footprint. In this mode, the satellite could work with the existing APRS satellites or future ones in constellation

In the Beacon mode, the satellite sends periodically the basic telemetry (Temperatures, currents and Battery voltage) to the earth to know the life of the satellite. This telemetry is encoded in an APRS (Automatic Packet Reporting System) Telemetry format in the Information field of AX.25 packet and is identified by the character T as following:

T#SSS,111,222,333,444,555,11111111, where SSS is the sequence number followed by the five 3 digit analog values and the eight binary values. A Control station could gather the satellite APRS telemetry data from the radio amateurs all over the world via Internet. The gathered data will be archived to a database and be accessed by Internet.

In the Command mode, the control station sends telecommand (TC) to the On Board Computer aboard the satellite and receives the housekeeping data telemetry (TM) from it to know the health status of the satellite.

This Telecommunication CubeSat satellite has two communications channels (see Fig. 1), which can be multiplexed over the same transceiver. The first channel carrying the TM/TC data (the application layer) between the satellite and the control station whereas the second channel carrying the payload data (the application layer) between the satellite and the Terminals. On bottom of the application layer, we used the AX.25 protocol (Data Link layer), which is an amateur radio adaptation of the ITU-T X.25 protocol. The AX.25 supports the requirements for amateur satellite communication. The AX.25 packets are sent over the radio links (the Physical layer), which use GMSK (Gaussian Minimum Shift Keying) for the Command mode, and AFSK (Audio Frequency Shift Keying) for the payload modes, over the same 145 MHz (uplink and downlink) VHF band. The use of one frequency for both uplink and downlink keeps the system overall cost very low.

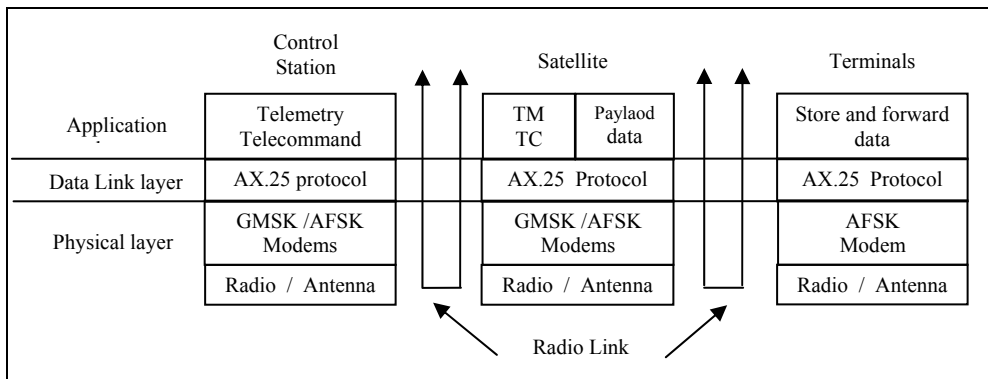|	Control Station	Satellite		Terminals
Application	Telemetry Telecommand	TM TC	Paylaod data	Store and forward data
Data Link layer	AX.25 protocol	AX.25 Protocol		AX.25 Protocol
Physical layer	GMSK /AFSK Modems	GMSK /AFSK Modems		AFSK Modem
	Radio / Antenna	Radio / Antenna		Radio / Antenna

Radio Link

Fig. 1. Satellite Communication Layers

At VHF frequencies, the antennas, receivers and transmitters for both the ground and the space segment, are readily available and inexpensive (Paffet et al., 1998). The Doppler shift, is kept at 3 KHz or below for the satellite parameters (altitude, minimum elevation angle, frequency band) (Jamalipour, 1998), which can be ignored.

In general case, the transmitted signal is corrupted by channel noise, by Doppler and multipath effects so that the bit decoding is not easy. In our case, the Doppler effect is negligible, and the mutipath effect is also negligible using low data rate.

The link budget, shown in Tab. 1, is calculated in the worst case when the satellite, with altitude of 650 Km, is just rising above the horizon with the elevation angle E = 10°.

To enhance the radio link in Command mode, between the control station and the satellite, we added a Forward Error Correction (FEC) by using convolutional coding and Viterbi decoding (Proakis, 1989).

Terminal Parameters	
Antenna gain	0 dBi
Transmitted power	5 W
Antenna Feed Loss	0.5 dB
T_{syst}	2000 k
Control Station Parameters	
Antenna gain	13 dBi
Antenna Feed Loss	1,5 dB
Transmitted power	5 W
T_{syst}	2000 k
Satellite Parameters	
Antenna gain	0 dBi
Transmitted power	1 W
Antenna Feed Loss	0.5 dB
T_{syst}	5000 k
Channel Parameters	
Bandwidth	15000 Hz
Free Space Loss	- 141 dB [*1]
Additional Losses	3.5 dB
E_b/N_0 required	13.6 dB[*2]
Uplink Link Margin for AFSK modulation	
E_b/N_0 estimated	21.42 dB[*3]
Margin	7.82 dB
Downlink Link Margin for AFSK modulation	
E_b/N_0 estimated	18.42 dB[*3]
Margin	4.82 dB
Uplink Link Margin for GMSK modulation	
E_b/N_0 estimated	26.9 dB[*3]
Margin	13.3 dB
Downlink Link Margin for GMSK modulation	
E_b/N_0 estimated	23.9 dB[*3]
Margin	10.3 dB

Table 1. Budget Link

[*1] Free Space Loss, FSL is calculated by (Maral & Bousquet, 2002):

$$FSL_{db} = 32.45 + 20\log\left(D_{km}\right) + 20\log\left(F_{MHz}\right) \qquad (1)$$

where F_{MHz} is the transmit frequency, and D_{km} is the distance or range of the satellite, calculated by:

$$D_{km} = \sqrt{\left(R_T + h\right)^2 - \left(R_T.\cos(E)\right)^2} - R_T.\sin(E) \qquad (2)$$

[*2] Required E_b/N_o gives an operating bit error rate of 2×10^{-5} (Proakis, 1989), assuming non-coherent demodulation of FSK.

[3] An estimation of the E_b/N_o is obtained from:

$$\frac{E_b}{N_0} = \frac{P.L_i.G_t.L_s.L_a.G_r}{k.T_s.R}$$ (3)

Where:

P = Transmitted Power
L_i = Feed Losses
G_t = Transmit antenna gain
L_s = Free Space Loss (FSL)
L_a = Tx Path Losses (Miscellaneous)
G_r = Receive antenna gain
K = Boltzmans constant
T_s = System temperature
R = Data rate of the system

3. Cubesat satellite architecture

3.1 Definitions

The satellite consists of two principal parts, namely: the payload and the platform. The payload (camera, repeater …) is defined by the satellite mission (earth observation, telecommunications …). This payload can not function correctly in the space environment without platform. The platform is made up of several subsystems; each one has its own role. The payload must be inside a structural subsystem (frame) for, on the one hand, being used as interface with the vehicle launch and, on the other hand, protecting the satellite from the space environment effects after its placing in orbit. The various electronic equipments on board the satellite need to be supplied by the electric power. Thus, the satellite needs a power subsystem which is responsible for the generation, the storage and the distribution of the electric power. The temperature in the space environment can vary between – 170 °C (in eclipse) and + 100 °C (in full sun). Thus, the satellite needs a thermal control subsystem which ensures that the temperature of each equipment on board remains within the correct operation temperatures interval. The payload components, like the antenna and the camera, must be pointed in a precise earth geographical area. The satellite will thus need an attitude control subsystem which will be responsible of controlling the orientation of the satellite in space (its attitude).

These various satellite subsystems, including the payload, must be controlled remotely. The need for data handling subsystem is then felt. Its role is to receive remote commands from earth control station and to execute them on board the satellite. For that, the data handling subsystem needs some communication equipments as transceiver and antennas.

In sum, the subsystems, which belong to the platform and offer their services to the payload, are (see Fig. 2):

- The structural subsystem (SS) ;
- The power subsystem (PS);
- The thermal control subsystem (TCS);
- The attitude control subsystem (ACS);
- The data handling subsystem (DHS).

Taking into account the miniaturization of the satellites, a new nomenclature appeared. Table 2 gives an idea of these satellite categories according to their weight.
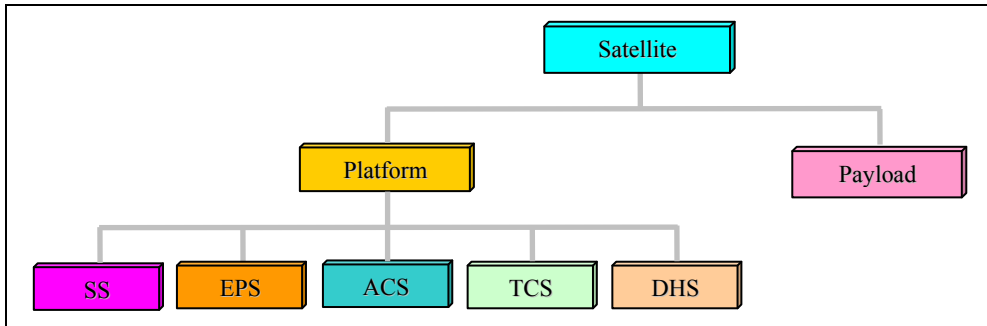
Fig. 2. Various satellite subsystems

Type of satellite	Weight (Kg)
Minisatellite	**100 - 1000**
Microsatellite	**10 - 100**
Nanosatellite	**1 - 10**
Picosatellite	**< 1**

Table 2. Nomenclature of satellites according to their weight

3.2 Satellite architecture overview

The Data Handling Subsystem is considered as an interface between the control station and the other satellite subsystems. The Data Handling Subsystem modules include communication subsystem (antennas, modulator, demodulator) and on board computer which controls the satellite subsystems. But how these subsystems are attached to the on board computer, two principal architectures are distinguished.
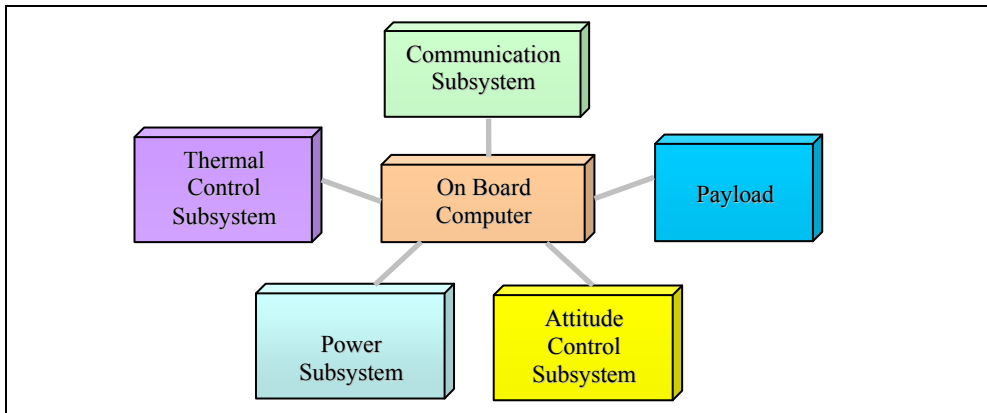

Fig. 3. Satellite Star Architecture

Star Architecture (see Fig. 3): the On Board computer (OBC) has to run separate data lines to each subsystem. In this architecture, the OBC should have many peripheral interfaces equal to the number of satellite subsystems on board and perhaps there will be a problem of wiring.
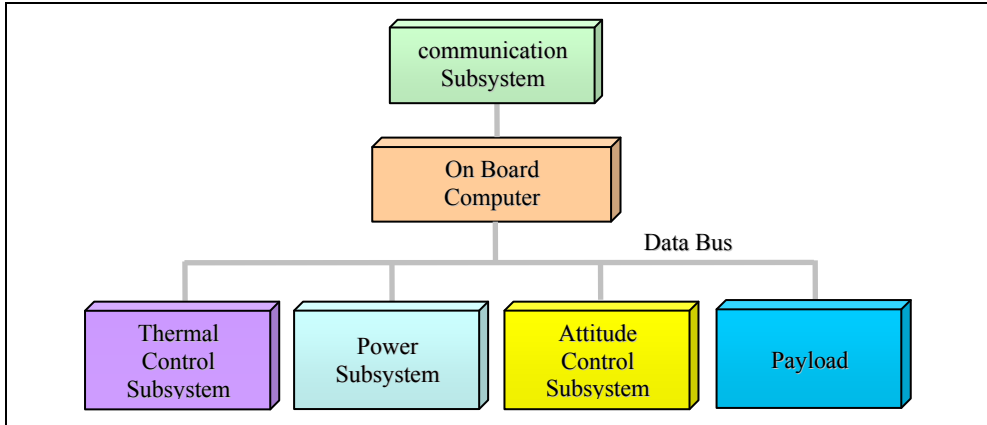
Fig. 4. Satellite Architecture with linear bus

bus Architecture (see Fig. 4): in this case, all satellite subsystems are connected to a bus which is similar to Local Area Network (LAN) aboard the satellite. In general, we used a Master-slave protocol. The three simple Master-slave bus commonly used in the embedded system are: CAN (Controller Area Network), SPI (Serial Peripheral Interface) and I²C (Inter Integrated Circuit).

	CAN	SPI	I²C
Low Power	3	5	5
Availability as microcontroller peripheral	2	5	4
Slave hardware complexity	2	5	3
Slave Software complexity	2	5	3
Wiring Complexity	4	3	5
Scalability to two Slaves	5	4	5
discrete logic interfacing ease	1	4	1
Sum	19	31	26
Key: 5= best, 1= worst			

Table 3. Master-Slave Buses study in case of two slaves

A trade study, comparing three buses, was conducted as seen in Table 3, in the case of two slaves. Notice that speed was not listed as a determining factor since the buses used similar data rates, all of which were plenty fast enough for CubeSat-class satellite needs. In the case of two slaves, the scalability of SPI bus is almost the same as I²C and CAN buses. According to Table 3, SPI bus won with the high score compared to CAN and I²C buses because of its simplicity. In the case of four slaves, the selected bus will be the I²C bus.

Obviously, the satellite can have a hybrid architecture made up of the two architectures (star and with a distributed linear bus). In this case, for example, payload subsystem is connected directly to the on board computer whereas the others subsystems are connected all by the means of one data bus.

3.3 Cubesat concept

The space age began in 1957 with the launch of Sputnik 1 weighing 80 kg. Six months later, the United States had launched an "orange" of 8 kg in space. The Pentagon was aware from the beginning that the dimensions and the weight have a great importance. However, space industry has moved towards larger, more elaborative spacecraft, because this may reduce launch costs and increase the longetivity of space investments. This was one reason for building huge launchers, like the Ariane-5 of the European Space Agency.

The prohibitively high costs of these space projects have restricted first-hand access to space to a handful of nations and international agencies. In the Nineties, research institutions are trying to go against this trend by developing microsatellites. This is mainly possible because state-of-the-art VLSI digital microelectronics enable very sophisticated functions to be achieved without having the constraints of small mass, volume and power.

Traditionally, the design of this type of satellites lasted several years and made their realization impracticable in the university world. Even in the case of realization, it appears difficult to find a launch opportunity. From this point of view, the CubeSat program was proposed in 1999 by professor Robert Twiggs from Stanford University (the USA). The goal of this program is to establish a uniform standard for the design of Nanosatellites and therefore it will increase the number of launches per year. The fundamental specification of this design requires that the satellite must be cubic with 10 cm in length (see Fig. 5) and its weight does not exceed 1 kg. The satellites fit into the CubeSat deployer, the P-POD. This carrier device holds a number of Cubesat what reduces the launch costs. In general, launch of Cubesat uses a modified SS-18 ICBM (Intercontinental Ballistic Missile) Russian or American, at a minimal cost of approximately 45.000 dollars. These days the CubeSat Initiative is a global congregation of universities and private companies from all over the world, trying to advance small satellite technology as tools of education and research.
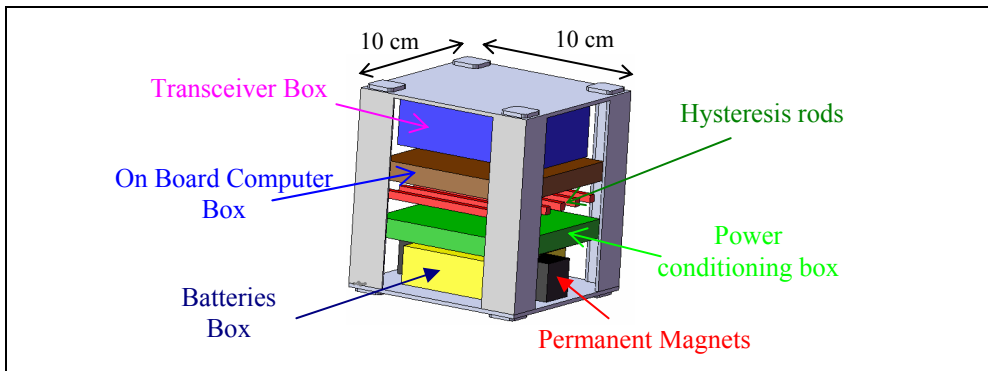

Fig. 5. Internal Arrangement of the Telecommunication CubeSat-class spacecraft

3.4 Cubesat subsystems requirements

The requirements of Cubesat-class spacecrafts are significantly different from their ancestors conventional satellites, and new design techniques are needed to meet these evolving requirements.

The unique requirements of the Telecommunication Cubesat satellite demonstrate this evolution. The design needs to be relatively inexpensive while at the same time

computationally robust. It should consume limited power and must support the space environment by reducing susceptibility to radiation and thermal effects.

In a previous text, we supposed that all the satellite subsystems are intelligent. i.e.; each satellite subsystem has its own microcontroller.

In the case of a Cubesat-class satellite where there are many constraints in terms of power consumption, mass and size. To design the Telecommunication Cubesat satellite, the adopted approach consisted in the integration of all the intelligences of the various satellite subsystems in only one intelligent subsystem. This intelligent subsystem that we called Data Handling Subsystem will integrate the Telemetry/Telecommand functions, the Telecommunication payload as well as the active part of the thermal control subsystem.

Fig. 6 represents the general architecture of the Telecommunication Cubesat satellite in such way that the attitude control subsystem will be designed completely with passive methods, i.e., on the one hand, it does not have any programmed microcontroller to carry out an algorithm of attitude control and, on the other hand, it does not consume any electrical power provided by the power subsystem.
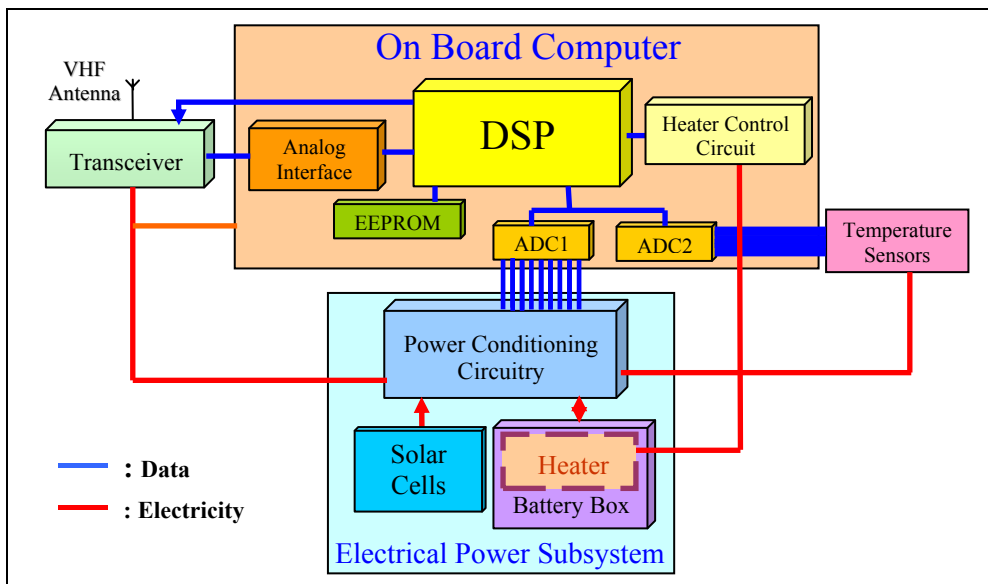

Fig. 6. General Architecture of the Telecommunication Cubesat Satellite

4. Cubesat satellite platform subsystems design

4.1 Space environment

An important issue to consider, which affects all electronic devices in space environment, is radiation. It can lead to various types of problems. These problems range from operational malfunctions to physical damage of the devices.

CMOS technology is preferred for space applications because of its high noise margins and low static power requirements. Scaling and integration are other advances CMOS technology has over other semiconductor technologies. On the other hand, CMOS is susceptive to two types of space radiation effects caused by electrons and protons trapped

by the terrestrial magnetic field: Total Ionizing Dose (TID) and Single Event Effects (SEE). TID effects are the result of accumulated exposure to ionizing radiation. SEE are the result of a single high-energy particle that strikes the device.

The total dose radiation exposure is measured in rads. The term rad (radiation absorbed dose) quantifies the total radiation exposure of a material. One rad(Si) is equal to 10×10^{-6} W of energy absorbed per gram of silicon. The total dose radiation threshold of a device is the minimum level of rad(Si) that will cause device failure. Typical CMOS technology devices can survive nearly 5 krads before physical damage occurs (Pisacane & Moore, 1994).

However, single-event effects are significantly more hazardous to the satellite and can result in either Single Event Upset (SEU) or Single Event Transient (SET) (Poivey et al., 2003) or Single Event Latchup (SEL). SEU effects are internal device memory bit changes (0 becomes 1 and vice versa) that can cause erroneous instruction execution and SET is a transient voltage pulse that can cause erroneous data transmission over a bus. SEU and SET effects are considered soft-errors and do not cause physical damage to the devices. In contrast, SEL effect is a hard-error, which leads to a high current-flow through the device. If not remedied quickly, latch-up can cause permanent damage.

4.2 Satellite structural subsystem

After placing the satellite in orbit, the satellite structure materials must protect the electronic components against the ionizing radiation. The Aluminium thickness of the satellite structure determine the life time of the satellite. Many measurements made on board various satellites allow us to know the energies of the electrons and protons trapped on various altitudes. The space agencies, like ESA (the European Space Agency), established radiation models from those measurements campaigns. From this point of view, we used ESA software called SPENVES to calculate the necessary shielding to protect the electronic components in space environment. The studies determined that the satellite will experience a total radiation dose of approximately 400 rads per month when we chose Aluminium shielding of 2 mm (see Fig. 7), taking into account the constraint of the Cubesat-class satellite
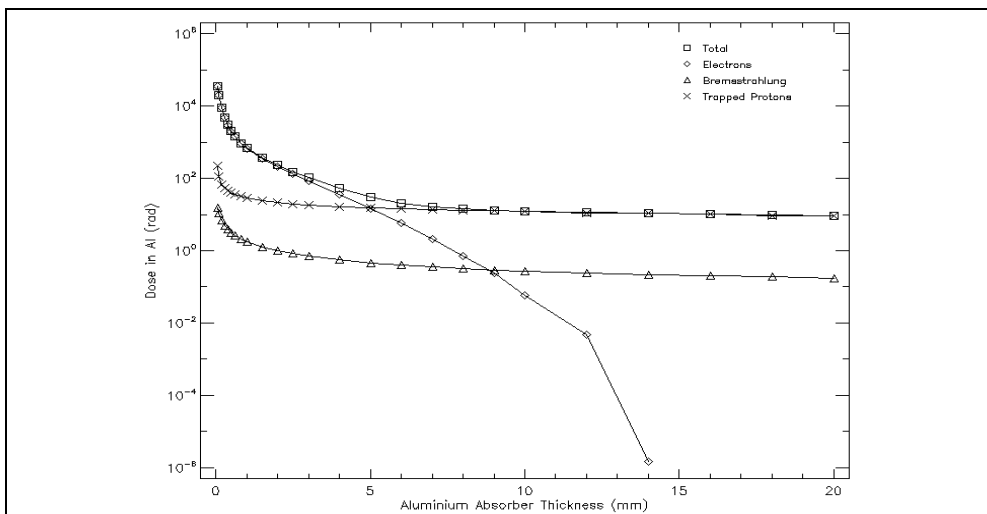

Fig. 7. Total Ionizing Dose for one month in function of the Aluminium Thickness

weight. The Aluminium shielding of 2 mm, which will be shared between the thickness of both the satellite structure and the electronic boxes, will allow the satellite to survive nearly 13 months. The studies also estimate that the satellite will experience approximately 2 bit-errors per 100 days, and only a small percentage of these will result in a latchup condition. If one in a thousand events cause latch up (Pisacane & Moore, 1994), the satellite will not observe any latch-up during its mission life.

4.3 Electrical power subsystem

The solar cells will be attached to the Cubesat structure, thus avoiding the need for solar panels and the necessary deployment system. The surface area for mounting solar cells is significantly reduced compared to conventional satellites, which results in less power generation ability. The solar cells chosen were spectrolab's Dual Junction Solar Cells, with an efficiency of 20% and dimension of 69 mm × 32 mm. They are not the most efficient available, but they provide the best value with low cost. These solar cells generate a power of 2.26 W which is a constraint
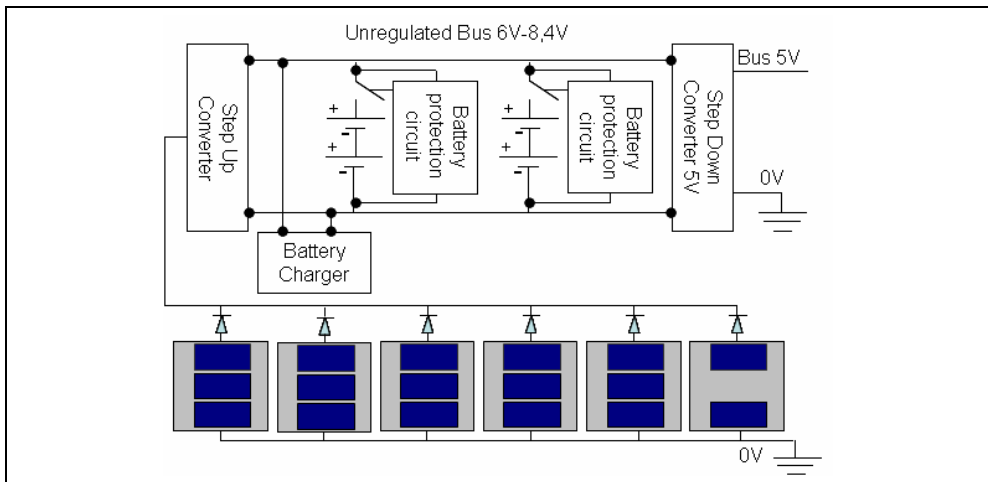

Fig. 8. Power Conditioning Circuitry Architecture

The battery Lion polymer DLP455660 from Danionics was chosen due to its extremely low weight, small size and good capacity. Four batteries were confined in a dedicated battery pressure box which will provide housing with sufficient stiffness to prevent the batteries from expanding against the high vacuum in Low Earth Orbit (LEO).

The Power Conditioning Circuitry (PCC), as shown in Fig. 8, comprises, one step-up converter based on MAX1771 circuit to provide the voltage to the unregulated bus, one battery charger based on ADP38101R-8.4 circuit with two battery protection circuits UCC3911 to protect the Li–Ion batteries from overcharging and overdischarging, and finally one step-down converter based on MAX1771 circuit to provide the voltage to the 5 V bus.

4.4 Thermal control subsystem

The advantage of using a limited number of electronics boards aboard the satellite is to provide a flexible internal configuration of components to arrange heat dissipating in such a

way that the excess heat is dissipated to the surroundings of the satellite as efficient as possible (Rotteveel, 2006).

Satellites that use a passive attitude control require a more homogenous thermal control where each side of the satellite body has more or less identical optical properties. As a result of the body mounted solar arrays, the optical properties of the different body panels are relatively homogeneous. This reduces the options to control passively the temperatures for the Nanosatellite. Therefore, the satellite will carry a combination of passive and active solutions for the thermal control. The passive control methods are in the form of thermal coatings (Wertz & Larson, 1999) of electronics boxes. For the active control, one heater and one temperature sensor are attached to the batteries box in order to protect them from cold temperature during the eclipse. The On Board Computer uses a timer to read periodically the batteries temperature among others telemetry data. The heater will be activated by the On Board Computer each time the batteries temperature drops below 5 °C (see Fig. 9). A polyimide heater from Minco Products has been selected. It is ideal for applications with space and weight limitations, or where the heater will be exposed to vacuum.
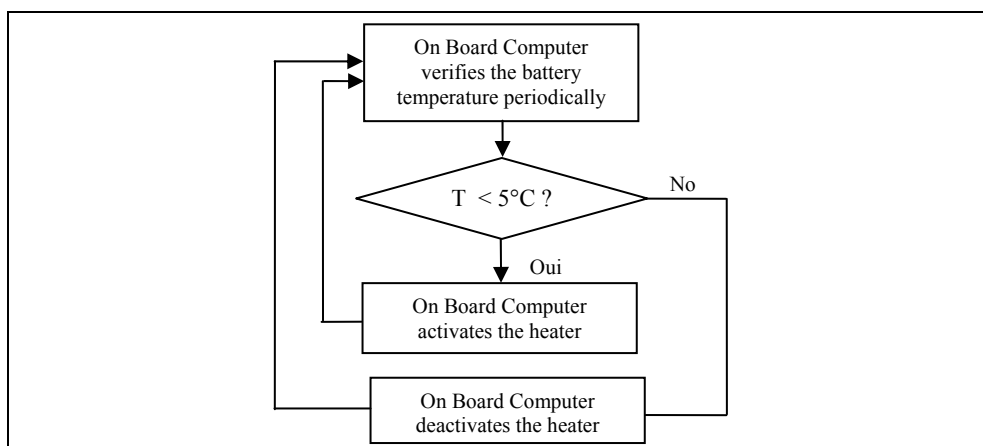

Fig. 9. Flow chart of the battery temperature control

4.5 Attitude control subsystem

To cut down the cost of the Telecommunication CubeSat-class satellite, all the nonessential elements is suppressed. An active attitude, which consumes the energy, is avoided by using omnidirectional antenna.

In the case of early launch CubeSats, the CubeSats that have adopted permanent magnets as passive attitude control, still alive and operated well. On the contrary, other Cubesats that incorporated active attitude control had many problems and was unstable.

The attitude control consists of two permanent magnets placed perpendicular to Six Hysteresis rods (see Fig.5). In fact, if one would think of the Earth as a giant bar magnet, then two parallel permanent magnets, using Neodymium Fer Bore (N35) material, aligned along an axis, would try to passively align the satellite according to the Earth's magnetic field lines. The satellite will spin around this axis and will be slowing down using six hysteresis rods (Alloy49) attached perpendicularly to this axis. The force caused by the hysteresis rods opposes the change in motion of the satellite. Also, it stabilizes the power

generation and provides uniform thermal distribution to the satellite. In order to point the antenna to the earth, the antenna side of the satellite should be made as heavy as possible, acting like a gravity gradient. The antenna is mounted close to the transceiver on the center of the side opposite to the magnet side. Fig. 10 shows the satellite motion on its orbit.
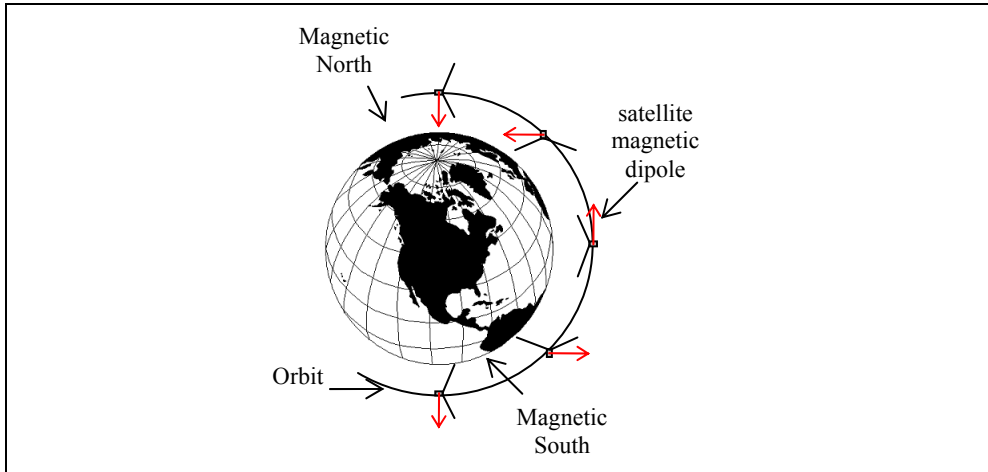

Fig. 10. Satellite motion on its orbit

5. Data handling subsystem design

In the satellite modular architecture (Wertz & Larson, 1999), each single subsystem has a dedicated hardware and software. The Cubesat has important constraints on cost, power and mass, and especially on size. The approach that has been taken in this research consists of the integration of the maximum subsystems within the same unit taking into account that single subsystems can be setup without modifying the operation of the remaining subsystems.

As said before, the Data Handling Subsystem will integrate the Telecommunication Payload, the Telemetry/Telecommand functions as well as the active part of the thermal control subsystem. In this section, we will describe in detail the main considerations and solutions chosen during the design of the Data Handling Subsystem based on the fixed point DSP processor. The design was split in two parts: The Hardware and Software Architectures.

5.1 Data handling subsystem hardware architecture

The Data Handling Subsystem hardware architecture is composed of three main parts (Fig. 11): an on board computer, Sensors and Control signals, and a VHF transceiver with associated antenna.

5.1.1 On board computer

The On Board Computer board as illustrated in Fig. 11 is a hardware board in which is implemented the flight software. The flight software controls the whole operations of the satellite and is built around a 16 bits DSP TMS320C5416 from Texas Instrument (Texas

Instrument, 2002). The hardware board includes, moreover, one analogue interface connected to the Radio Frequency module (Transceiver and antenna), two analogue to digital converters for the acquisition of the Telemetry housekeeping data, a EEPROM memory containing the flight software, a JTAG port for the flight software programming and finally the control signals, by using some DSP Input/Output signals, in particular, to activate or deactivate the heater. Fig. 11 illustrates the hardware architecture of the on board computer. All the components on the on board computer logic board are chosen with SMD (Surface Mounted Device) packages for space saving and high density mounting in order to minimize the weight and dimensions of the logic board.
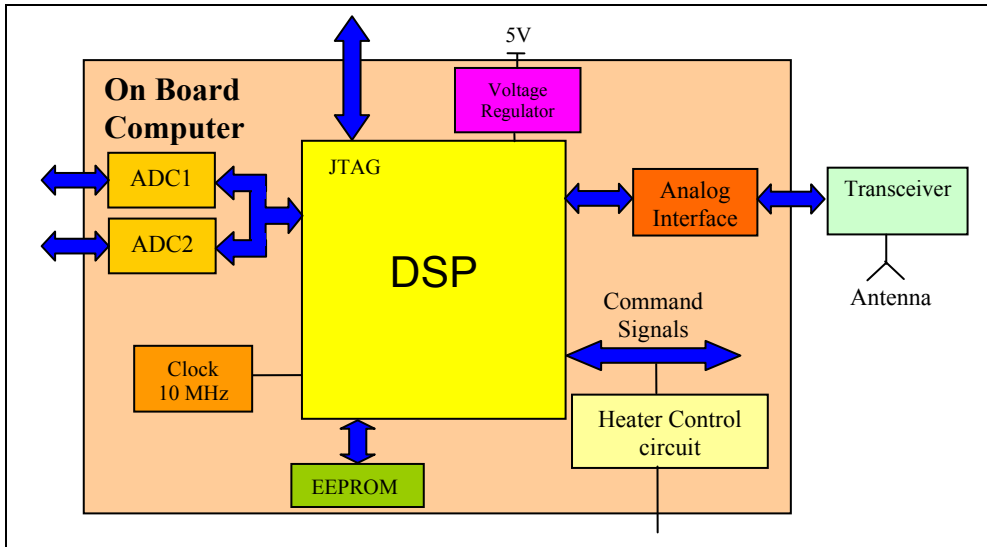

Fig. 11. The hardware architecture of the on board computer

Processor

The Cubesat projects generally use different general purpose processors for the on board computer. The AAU CubeSat used C161PI microcontroller from Infineon, the CUTE-1 CubeSat used an 8-bit H8/300 CPU and The RinconSat CubeSat used a PIC16C77 microcontroller from Microchip Corporation.

As on board computer for the Telecommunication CubeSat-class satellite, the TMS320C5416 DSP efficient power consumption, from Texas Instrument was chosen for its ease of implementing the physical layer (AFSK and GMSK modems) and the data link layer (AX.25 protocol), and also for the availability of many reports applications, which reduce the design time of the satellite. The advantage of implementing AFSK and GMSK modems as software is to eliminate the need for external modem hardware needed by other satellites (Hunyadi et al., 2002), and to minimize the dimensions of the on board computer logic board which must be less than 10 cm x 10 cm.

The DSP processor is connected to VHF transceiver through the AIC (Analogue Interface Circuit) interface. The TMS320C5416 contains an adjustable PLL which allow us to multiply the frequency of Quartz to reach the maximum of 160 MHz. However, the operation clock frequency of TMS320C5416 is selected to 40 MHz for the following reasons: to minimize the

energy consumption, to minimize the heat generation and also to avoid the Single Transient Event effect (SET) (Poivey et al., 2003).

Memory

The On Board Computer is an embedded system where the flight software is implemented in an external memory. To immunize against SEU effect, the application program is loaded in the external Programmable Read Only Memory (PROM). In general, the PROM memory is used, because it is very robust to radiation and in wide range of temperature. In the event that the APRS OBDH unit countermeasures fail, a Fail Safe Circuit is used to perform a satellite reset at regularly scheduled intervals.

The boot mode is chosen to be the SPI EEPROM boot mode using the AT25256A EEPROM memory of 32 Kwords of 8 bits which operates in the temperature range from -40 up to 125 °C. The SPI EEPROM mode is the only mode that does not require the addition of electronic components to control the transfer of the flight software into the memory program of TMS320C5416 for a faster execution. The DSP TMS320C5416 contains 128K × 16 on-chip RAM and 16K × 16 on-chip ROM.

Analogue Interface

The analogue interface or AFE (Analogue Front End) carries out several operations which will make it possible to adapt the signals between the DSP and Radio Frequency (RF) module. These operations are practically amplification, filtering and conversion. Fig. 12 illustrates the internal architecture of AFE interface which is chosen to be the TLC320AC021 analog interface from Texas Instrument.

The TLC320AC021 (Texas Instrument, 1996) (AIC), includes both analog-to-digital and digital-to-analog converters in the one package. This device incorporates a band pass switched capacitor antialiasing filter, a 14 bit conversion process for both the ADC and DAC, and a low pass switched capacitor output reconstruction filter. In addition, the AIC also provides a direct serial interface to the TMS320C5416. Use of this AIC greatly simplified the hardware necessary to provide an analogue interface to the DSP development system. The TLC320AC021 is specified for the −40 ◦C to 85 ◦C temperature range.
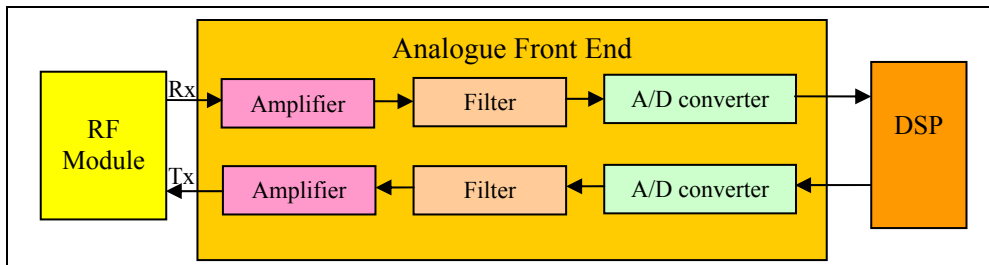

Fig. 12. the internal architecture of AFE interface

Analogue Digital Converter

The Interface to the different 17 sensors is done by using the TLV2556 circuit, which is a serial Analogue Digital Converter (ADC) with 11 analogue multiplexed channels. The serial bus, connected to the DSP, is based on the SPI specification.

The TMS320C5416, which is the Master, selects one of the two TLV2556, used as a slave mode. The TLV2556I is characterized for operation from −40 ◦C to 85 ◦C.

5.1.2 Transceiver

A COTS amateur radio transceiver was adopted to be the main flight radio due to power, weight, and time constraints (Horan, 2002; Lu, 1996). The transceiver, integrated inside the payload, operates in amateur VHF band and consists of the "guts" of a low cost Yaesu VX-1R (Hunyadi et al., 2002), arguably one of the smallest and lightest handhelds on the market. The radio is two stackable double sided PCBs measuring approximately 5×5 cm². The small size of CubeSats limits the amount of energy provided by solar arrays; therefore power availability is a constraint on both the spacecraft processor and the communications systems. Power is supplied from 5 V bus to radiate only 1 Watt RF power which achieves a positive budget link (see Table 1). Current consumption for the receiver and transmitter is 150 mA and 400 mA, respectively. Only slight modifications will be required to make this component space worthy. We use only the RF parts of the VX-1R.

The transceiver is interfaced with the DSP processor by means of AF (Audio Frequency) and PTT (Push To Talk) command signals. AF signals consist of transmit and receive signals of the AFSK and GMSK modems which carry data packet, whereas PTT command signal allows the DSP to choose the transmitting and receiving frequencies of the radio. The transmitting and receiving frequencies will be fixed on the 145.825 satellite frequency. The FM signal output from the transceiver is fed to an omnidirectional antenna.

5.1.3 Antenna

We have previously said that the attitude control of the satellite is based on the Earth magnetic field. As a result, the pointing direction of the satellite depends on its latitude. Therefore, the satellite antenna must be omnidirectional to cover the earth Terminals placed in all latitudes. For the satellite antenna, the turnstile antenna (Milligan, 2005) will be used. It consists of two pairs of half-wavelength dipole antenna positioned orthogonal to each other with a relative phase difference of 90 degree to achieve circular polarization. In theory of transmission line, a 90-degree phase shift can be obtained by adding a ¼ wavelength transmission line to the feed of the antenna. However, due to the limited available space in the satellite and to the long ¼ wavelength transmission line at VHF frequencies; implementation using lumped elements (Bahl, 2003) is desirable. To make easier the deployment of the antenna, the measuring tapes will be used as the four elements that consists the circularly polarized antenna.

5.2 Flight software architecture

The designed flight software of the Telecommunication CubeSat-class satellite is made to boot-up automatically from the external EEPROM memory and then to configure the TMS320C5416 DSP to carry out the various functions of the on board computer. The flight software is a real time software which treats the functions of the on board computer dependently of the material environment of the TMS320C5416 DSP. Its architecture was built according to two points of view:

A static point of view: This describes how the flight software activities are organized. All the functions to be realized by the software are broken up into modules (objects) at the same time as the definition of the relations between these modules. Those are again broken up. The software is presented in a hierarchy of modules. The modules with the lowest level are coded in the programming language. They are tested then assembled to recompose the module of the higher level and so on up to the highest level.

A dynamic point of view: This describes how, in time, the flight software activities will be processed (we speak about the software behaviour). These activities are organized in tasks

which share in time the resources of TMS320C5416 DSP (CPU, Memory) by assigning a priority to each task (depending on the priority of each interruption).

The assembler programming is used for the modules which require real time processing like the modules including in the communications module (COM module). In addition, the programming in C is used for the programming of the other modules of the flight software to reduce the time of their developments.

5.2.1 Static architecture of the flight software

Fig. 13 shows the static architecture of the flight software which is a modular architecture organized in hierarchical levels. It is designed to boot-up from a Main Control Program (MCP) acting as the principal module of the first level. Its principal function is to route the information inside the various software modules of the on board computer. The modular architecture of the flight software makes it possible to test the functionality of each module independently of the others.
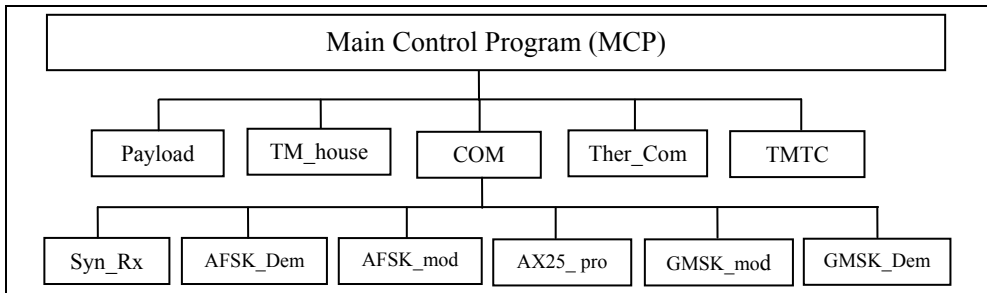

Fig. 13. Static Software Architecture of the on board computer

TM _ house	acquires of the telemetry data of the type "housekeeping".
Payload	carries out the missions of the Nanosatellite payload.
Ther_Con	activate ou deasactivate the heater.
TMTC	executes the telecommands and sends all type of telemetries.
Syn_Rx	has as a function the detection of the beginning of the received packet.
AX25_Pro	carries out the packing and the unpacking of the AX.25 packet.
AFSK_Mod	carries out the AFSK modulation.
AFSK_Dem	carries out the AFSK demodulation.
GMSK_Mod	carries out the GMSK modulation.
GMSK_Dem	carries out the GMSK demodulation.

Table 4. Various modules included in the flight software

5.2.1 Dynamic architecture of the flight software

Once the satellite is separated from the launch vehicle or after a reset, the TMS320C5416 DSP loads the software, stored in the external EEPROM memory, into its memory program for a faster execution of the software. The first part of the software contains the instructions necessary to configure the various registers and to initialize the various peripherals of the TMS320VC5416 DSP. Once these initializations are finished, the program is put on idle state waiting for an interruption either from the Timer or from the analogue interface. The flow chart of this main program is given by Fig. 14.

```
                    ┌─────────────────────────────┐
                    │  Load of the flight software │
                    │     from the EEPROM memory   │
                    └─────────────────────────────┘
                                  │
                    ┌─────────────────────────────┐
                    │ Initialization of the DSP    │
                    │ registers and initialization │
                    │      of the variables        │
                    └─────────────────────────────┘
                                  │
                            < RINT0=1 ? >──── Yes ───┐
                              No │ Non                │
                            < TINT=1 ? >              │
                                │ Yes                │
                           ┌──────────┐              │
                           │ TM  house│              │
                           └──────────┘              │
                           ┌──────────┐              │
                           │ Ther  Con│              │
                           └──────────┘              │
                           ┌──────────┐              │
                           │ AX25  Pro│              │
                           └──────────┘              │
                           ┌──────────┐              │
                           │ AFSK  Mod│              │
                           └──────────┘              │
                           ┌──────────┐              │
                           │  Syn  Rx │◄─────────────┘
                           └──────────┘
                          < Detection ? >──── No
                                │ Yes
                         ┌────────────────┐
                         │ Stop of the Timer│
                         └────────────────┘
                         ┌──────────┐
                         │ AFSK  Dem│
                         └──────────┘
                         ┌──────────┐
                         │ AX25  Pro│
                         └──────────┘
                       ┌────────────────┐
                       │ Packet decoding│
                       └────────────────┘
             Yes ──< Is the packet from control station? >── No
               │                                              │
         ┌──────────┐                                   ┌──────────┐
         │  TMTC    │                                   │ Payload  │
         └──────────┘                                   └──────────┘
               │            ┌──────────────────┐            │
               └───────────►│ Restart of the Timer│◄─────────┘
                            └──────────────────┘
```
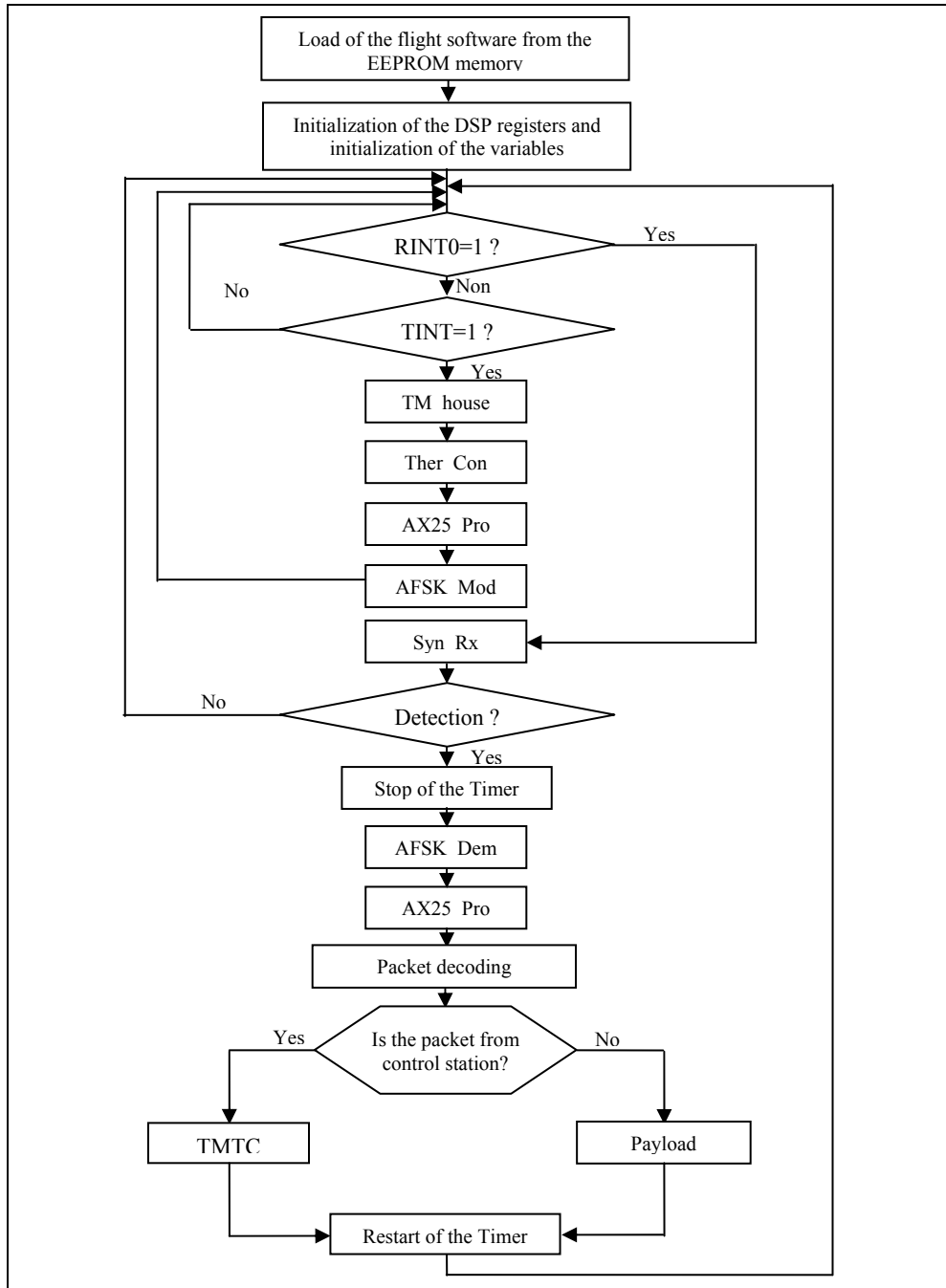
Fig. 14. The main program flow chart of the on board computer

If the interruption will come from the timer (TINT=1), the main control program (MCP) will call first the "TM_house" module, then it will control the battery temperature by the "Ther_Con" module. After that, it will use the "AX25_Pro" and "AFSK_Mod" modules for sending the APRS beacons and, at the end, it will return to its idle state.

In addition, if the interruption will come from the analogue interface (RINT0=1), the MCP will control the reception of the data by the "Syn_Rx" module. If the beginning of the packet is detected, the MCP will stop the Timer counting, and then it will call the "AFSK_Dem" module. At the end of the demodulation, the MCP will test the received packet if it contains an error by the "AX25_Pro" module. The MCP will reject the erroneous packet and will respond to the request of the correct packet. If the packet is coming from the control station, the MPC will give the command to the "TMTC" module to take over. If it is coming from terminals, the "Payload" module will take over. At the end of these tasks, the MPC will return to its idle state and will continue the counting of the Timer from its value at the moment of stop.

5.3 DSP implementation

In this research, we give a particular attention to software optimization due to small amount of memory available on DSP. Today, "C" compilers offer many features to facilitate DSP programming but they do not guarantee the best code optimization. We have defined a methodology approach based on two design steps: algorithmic optimization, and optimal programming using assembler language for the modules which need the real time. By an optimal software design that takes the maximum advantages from DSP architecture facilities, good real time experimental performances have been obtained for the on board computer implemented modules. The implemented modules include AX.25 protocol, AFSK modulation, AFSK demodulation, Convolutional encoding and Viterbi decoding, GMSK modulation, and GMSK demodulation. These modules were tested in real time using two DSP platforms, one for the transmission process and the other for the reception process. The used platforms are TMS320C54X DSKPlus kit (Texas Instrument, 1997) and TMS320C5416 DSK kit (Texas Instrument, 2002), which are compatible. Fig. 15 illustrates the interconnection of the two development boards DSKC542 and DSKC5416.
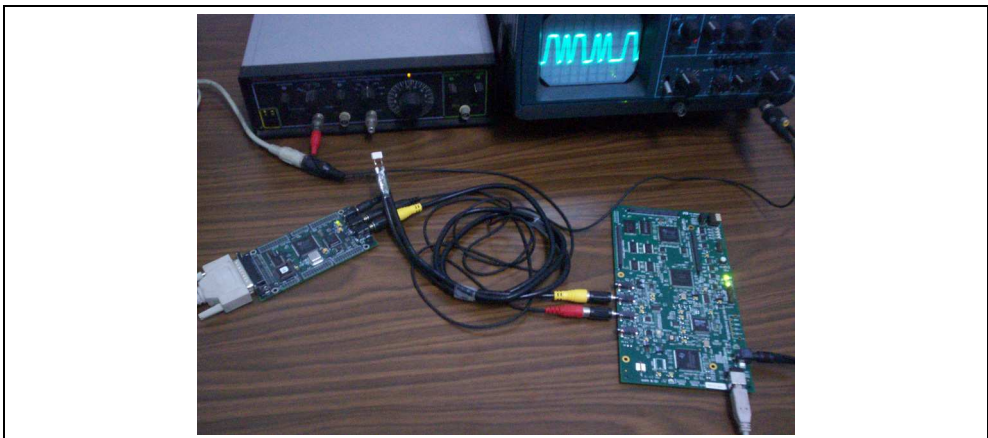

Fig. 15. The interconnection of the DSKC542 and DSK5416 boards

5.3.1 AX.25 communication protocol

The AX.25 protocol (TAPR, 1997) is based on HDLC format. The general frame format used for data transmission is shown in Fig. 16.
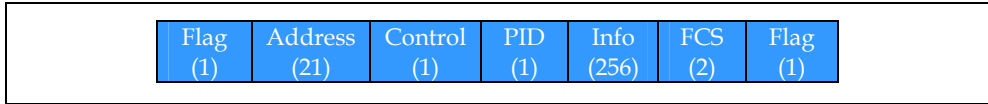
Flag (1)	Address (21)	Control (1)	PID (1)	Info (256)	FCS (2)	Flag (1)

Fig. 16. The AX25 frame format where number of bytes are in parenthesize

Briefly described the functionality of each field in the frame is:

Flag: Indicates start and stop of the frame

Address: Contains sender address, receiver address and satellite address

Control: Identifies the type of frame which in this case is an information frame

PID: Identifies the type of top-level protocol

Info: Contains from zero to 256 bytes of data

FCS: The Frame Check Sequence which will be described in detail below.

The AX.25 protocol uses the standard CRC-CCITT code based on a 16-bit polynomial G(x) to calculate FCS field.

$$G(x) = x^{16} + x^{12} + x^5 + 1 \tag{4}$$

For CRC-CCITT code implementation on DSP, the truncated polynomial of 0x1021 is used with initial value of 0xFFFF.

5.3.2 AFSK modulation

A direct digital frequency synthesis technique (DDS or DDFS) combined with a digital-to-analog converter (DAC) provides an approach to implementation of an Audio Frequency Shift Keying (AFSK) modulator. Samples of a sine wave are generated directly from the sine look-up table. The primary function of AFSK modulator is the following: Given a stream of binary data a_0, a_1, a_2,.. , a_k for each element a_k [0, 1], generates a corresponding signal of frequency, f_0 = 1200Hz or f_1 = 2200Hz, for the duration of a bit period. The 1200Hz tone represents the binary "1" and the 2200Hz tone represents the binary "0" as indicated in the standard Bell 202. The AFSK modulator generates the two tones by stepping through the same sine table.

Suppose f is the desired frequency. The signal to generate is x (t), with:

$$x(t) = \sin(2\pi f t) = \sin(\varphi(t)) \tag{5}$$

and $t = nT_s$ where T_s is the sampling period. Because of Shannon's theorem, $1/T_s$ is chosen greater than $2f$. The phase φ belongs to the interval [0,2π]. It is possible to suppress the time consuming comparison tests with 2π. This can be achieved by using circular buffer as intrinsic feature of TMS320C54X processors. The size N of the table can be calculated in order that the step index S is an integer for both frequencies. This condition can be written as:

$$N f_0 T_s = S_0 \quad and \quad N f_1 T_s = S_1 \tag{6}$$

The size of the step index determines the output signal frequency. At the bit rate f_b of 1200 Hz, an interruption of AIC is sent to the DSP. To generate the frequency f_0 = 1200 Hz (respectively f_1 = 2200 Hz), the sine table of size N = 120 is read with an integer step index equal to S_0 = 6 (resp. S_1=11).

For the implementing of the AFSK modulation on DSP, we used the sampling frequency of 24 KHz and data rate of 1200 bps which corresponds to 20 samples per bit. The steps and the sine samples are represented as 16 bit integer numbers. Fig. 17 represents the output of the AFSK modulator with the following bits of inputs [-1 1 1 1 -1 1 -1 -1 1 1].
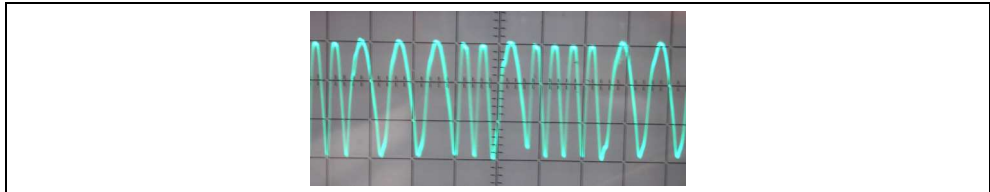
Fig. 17. The AFSK signal

5.3.3 AFSK demodulation

We used a bit-per-bit demodulation as the classical non-coherent demodulation scheme. The received AFSK signal is sent to DSP from the transceiver via the TDM serial port after being converted from analog to digital signal by AIC. The DSP implementation of the AFSK demodulator is illustrated in the Fig. 18.
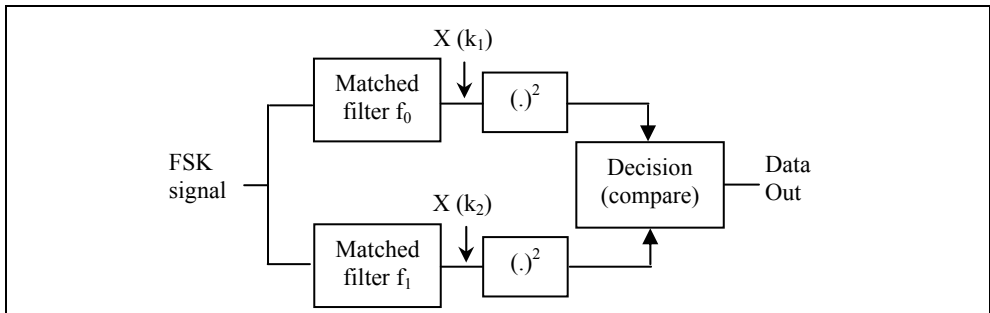
Fig. 18. General diagram of AFSK demodulation

We used the Goertzel algorithm (Oppenheim, 1999) to demodulate the AFSK signal, which can be interpreted as a matched filter for each frequency k as illustrated in Fig. 18. The transfer function $H_k(z)$ corresponds to the kth Goertzel filter:

$$H_k(z) = \frac{1 - e^{j2\pi k / N} z^{-1}}{1 - 2\cos\left(2\pi k / N\right)z^{-1} + z^{-2}} \tag{7}$$

A further simplification of the Goertzel algorithm is made by realizing that only the magnitude squared of X(k), which represents the energy of the received signal, is needed for tone detection. It eliminates the complex arithmetic and requires only one coefficient, a_k = cos($2\pi k/N$), for each $|X(k)|^2$ to be evaluated. Since there are two possible tones to be

detected, we need two filters described by (7). We conclude that the Goertzel algorithm is a Discrete Fourier Transform, calculated from a second degree recursive filter, easy to implement on DSP. In Our case, we compare only the two energies of the two AFSK frequencies to determine which AFSK tone has been received.

The synchronization is performed by detecting the first change to the received signal by using the Syn_Rx module. After processing 20 samples for each bit and calculated the energy at each of the two frequencies, the Goertzel Algorithm then decides which AFSK tone has been received. The sampling frequency is chosen to be 24 KHz because it is the highest sampling frequency available in the AIC. Also to detect the frequency 1200 Hz (resp. 2200 Hz), we used k = 1 (resp. k = 1.83). For M = 20, we have a_1 = 0.951 and a_2 = 0.839, which are corresponding to frequencies 1200 Hz and 2200 Hz respectively. The format of each variable in the algorithm was being chosen suitably taking into account that we had used a 16 bit fixed point DSP.

5.3.4 GMSK modulation

The GMSK modulation is a Continuous Phase Modulation (CPM) with a modulation index h=0.5. A modulated GMSK signal can be expressed, over the time interval $nT_b \leq t \leq (n+1)T_b$, as:

$$s(t) = A\cos\left(2\pi f_0 t + 2\pi h \sum_{k=-\infty}^{n} d_k \int_{-\infty}^{t} g(\tau - kT_b)d\tau\right)$$
(8)

where d_k: sequence of data information = ±1,

and $g(t) = \dfrac{1}{2T_b} rect(t/T_b) * h_g(t)$ with $rect(t) = 1 \quad for \quad |t| \leq 0,5$

$h_g(t)$ is the pulse of Gaussian function, T_b is the symbol period, B is the 3dB bandwidth of the Gaussian prefilter, and $g(t)$ is the response of the transmitted rectangular pulse to the pre-modulation filter.

By deriving the phase signal, the CPM can also be seen like Frequency Modulation (FM). The instantaneous frequency F_i is given by:

$$F_i(t) = f_0 + h \sum_{k=-\infty}^{n} d_k g(t - kT_b)$$
(9)

In the expression (9), h represents the proportionality constant of the modulator and is expressed in Hertz per volt. The baseband signal m(t) to be transmitted is written then, in the interval $nT_b \leq t \leq (n+1)T_b$, in the form of:

$$m(t) = \sum_{k=-\infty}^{n} d_k g(t - kT_b)$$
(10)

In theory, the duration of Gaussian filter is infinite, but in practice, we limit the function $h_g(t)$ to the few period bits over which it is significantly not zero. This duration is inversely proportional to B. For a product $BT_b = 0.5$, we consider that $h_g(t)$ is not zero over 2 bits. The convolution product of $h_g(t)$ with a rectangle function of duration T_b lasts $3T_b$, which affects

the half preceding bit and the half following bit. The Fig. 19 represents the response of Gaussian lowpass filter for BT_b = 0.5 over three bits to a rectangular pulse of duration T.

The implementation of filter convolution product requires multiple instruction processing inducing a lot of calculation time. To respect timing constraints we propose an optimized implementation code based on Lookup table of the Gaussian filter response (Fig. 19). For the implementing of the GMSK modulation on DSP, we used the sampling frequency of 24 KHz with 5 samples per bit which corresponds to data rate of 4800 bps. For data stream of [1 -1 1 1 1 -1 -1 1], the corresponding GMSK baseband signal is given by the Fig. 20.
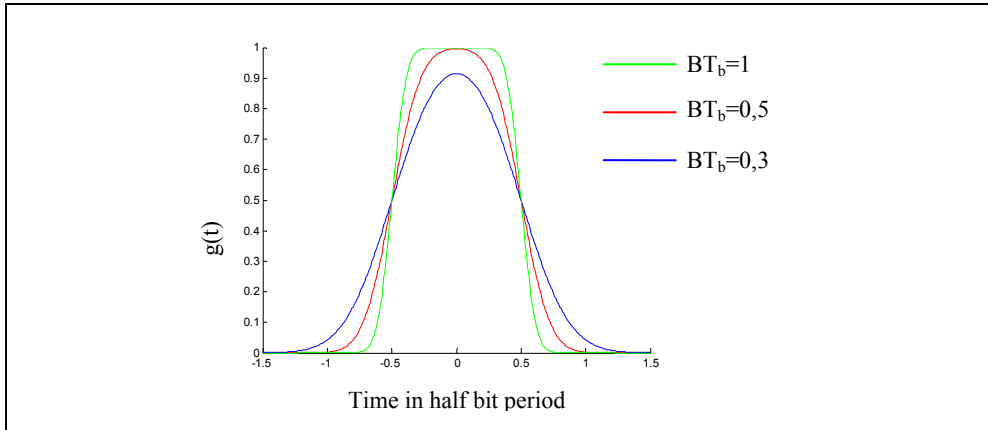
Fig. 19. Gaussian filter response in function with BT_b parameter
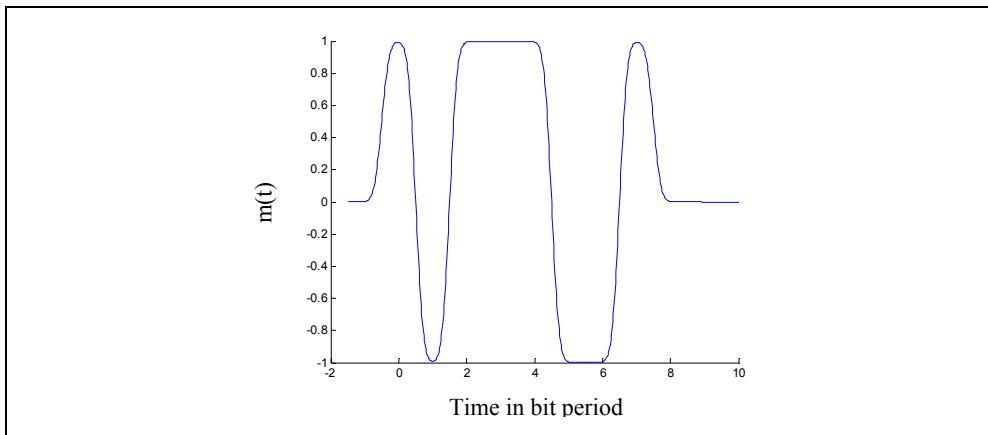
Fig. 20. Baseband GMSK output signal

5.3.5 GMSK demodulation

We used the classical non-coherent demodulation scheme, which performs a bit-per-bit demodulation and it does not require recovery of the carrier phase and frequency. Analysis of the GMSK baseband signal (Fig. 20) permits the identification of eight types of shapes

corresponding to binary states transition. The GMSK demodulator must extract the phase from the modulated signal and, by using a transition shape classification, decode the transmitted bit.

According to Fig. 21, we have four transition shapes for a binary "1", and four transition shapes for a binary "0". We store only two predictive transitions, (b) and (f), on the DSP memory as look-up tables. Based on the lookup tables, the demodulator uses the Absolute distance d_e, which shows the better performance, as matching function to classify the GMSK signal transitions, and determine the transmitted bit.
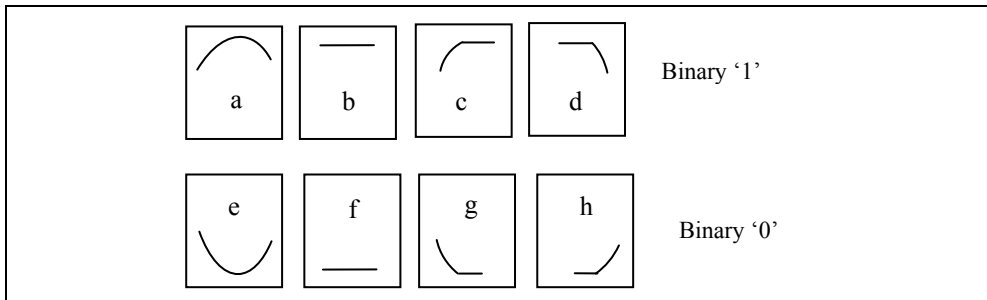
Fig. 21. Eight binary states transitions

$$d_{abs}(x, y) = \sum_{j=1}^{n} \left| x_j - y_j \right| \tag{11}$$

The demodulation of the GMSK signal is processing to perform the shape comparison of binary transition based on the look up tables. The minimum Euclidean distance d_e is evaluated and the decoded bit is determined. The synchronization is performed by using the Syn_Rx module. The C54x DSP family has a dedicated instruction for faster execution of the Absolute distance.

6. Conclusion

As the satellite community transitions towards inexpensive distributed small satellites, new methodologies need to be employed to replace traditional design techniques. The ongoing research will contribute to the development of these cost saving methodologies. The goal of the integration of all the intelligences of the various satellite subsystems in only one intelligent subsystem is to minimize component expenditures while still providing the reliability necessary for mission success.

Associating low cost ground terminals with a low cost Telecommunication CubeSat-class satellite will allow universities to access space communications with a very economical system. The present work, dealing with the design of the Low-cost Telecommunication CubeSat-class spacecraft, shows hardware and software solutions adopted to cut down the system cost. The hardware utilizes commercial low cost components and the software is optimized using assembler language. The On Board Computer unit is small device that can be mounted on any small satellite platform to serve telecommunications applications such as mobile localization and data collection. By using a single CubeSat satellite and low-cost

communications equipments, Telecommunications systems can be kept at the extreme low end of the satellite communications cost spectrum.

7. References

Addaim, A.; Kherras, A. & Zantou, B. (2008). Design and Analysis of Store-and-Forward Data Collection Network using Low-cost Small Satellite and Intelligent Terminals, *Journal of Aerospace Computing, Information and Communications,* Vol. 5, No. 2, (February 2008) page numbers (35-46)

Bahl, I. (2003). *Lumped Elements for RF and Microwave Circuits*, Artech House, first ed.

Gérard, M. & Bousquet, M. (2002). *Satellite Communication Systems*, John Wiley & Sons; fourth edition

Horan, S. (2002). Preparing a COTS radio for flight – lessons learned from the 3 corner satellite project, *Proceedings of 16th Annual/USU Conference on Small Satellites*, Logan, Utah, USA

Hunyadi, G.; Klumpar, D.; Jepsen, S.; Larsen, B. & Obland, M. (2002). A commercial off the shelf (COTS) packet communications subsystem for the Montana EaRth-Orbiting Pico-Explorer (MEROPE) CubeSat, Proceedings of IEEE Aerospace Conference

Jamalipour, A. (1998). *Low Earth Orbital Satellites for Personal Communication Networks*, Norwood, MA: Arthech House

Lu, R. (1996). Modifying off-the-shelf, low cost, terrestrial transceivers for space based application, *Proceedings of the 10th Annual AIAA/USU Conference on Small Satellites*, Logan, September 1996, Utah, USA

Milligan, T. (2005). *Modern Antenna Design*, second ed., Wiley

Oppenheim, A.; Schafer, R. & Buck, J. (1999). *Discrete-Time Signal Processing*, second ed., Prentice Hall

Paffet, J.; Jeans, T. & Ward, J. (1998). VHF-Band Interference Avoidance for Next-Generation Small Satellites, *Proceedings of 12th AIAA/USU Conference on Small Satellites*, Logan, Utah, USA

Pisacane, V. L., & Moore, R. C. (1994). *Fundamentals of Space Systems*, New York: Oxford University Press

Poivey, C.; Buchner, S.; Howard, J. & Label, K. (2003). *Testing Guidelines for Single Event Transient*, NASA Goddard Space Flight Center, 30 June, 2003.

Proakis, J. (1989). *Digital Communications*, McGraw-Hill, (Second Edition)

Rotteveel, J. (2006). Thermal control issues for nano- and picosatellites, *Proceedings of Space Technology Education Conference*, Germany, May 2006, Braunschweig.

TAPR, (1997). AX.25 Link Access Protocol for Amateur Packet Radio, TAPR, version 2.2

Texas Instruments, (1996). TLC320AC01 data manual single-supply analog interface circuit, SLAS057D

Texas Instruments, (1997). DSKplus User's Guide, SPRU191

Texas Instruments, (2001). TMS320C54X DSP: CPU and peripherals, SPRU131G.

Texas Instrument, (2002). TMS320VC5416 DSK Technical Reference,

Wertz, R. & Larson, W. (1999). *Space Mission Analysis and Design*, Microcosm, (third ed.)

Zantou, B. & Kherras, A. (2004). Small Mobile Ground Terminal Design for a Microsatellite Data Collection System, *Journal of Aerospace Computing, Information and Communications,* Vol. 1, No. 9, (September 2004) page numbers (364–371)

Aerospace Technologies Advancements

Edited by Thawar T. Arif

Space technology has become increasingly important after the great development and rapid progress in information and communication technology as well as the technology of space exploration. This book deals with the latest and most prominent research in space technology. The first part of the book (first six chapters) deals with the algorithms and software used in information processing, communications and control of spacecrafts. The second part (chapters 7 to 10) deals with the latest research on the space structures. The third part (chapters 11 to 14) deals with some of the latest applications in space. The fourth part (chapters 15 and 16) deals with small satellite technologies. The fifth part (chapters 17 to 20) deals with some of the latest applications in the field of aircrafts. The sixth part (chapters 21 to 25) outlines some recent research efforts in different subjects.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

INTECH
open science | open minds