Design of Store and Forward Data Collection Low-cost Nanosatellite

A. Addaim, A. Kherras, B. Zantou
Centre for Space Research and Studies
Mohammadia School of Engineers
B.P 765, Rabat, Morocco,
addaim@emi.ac.ma, kherras@emi.ac.ma, zantou@crers.gov.ma

Abstract—The satellite, as small as the Cubesat concept, requires employment of limited small boards. In the modular architecture, each single subsystem has a dedicated hardware and software. The approach, which has been taken in this paper, based on two design directives: 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, and the elimination of nonessential elements. In this paper, we will describe in details the design of integrated store-and-forward APRS (Automatic Packet Reporting System) payload and OBDH (On Board Data Handling) subsystems using one chip based on the fixed point DSP (Digital Signal Processor) to design a low-cost Nanosatellite. By an optimal software design that takes the maximum advantages from DSP architecture facilities, good real time experimental performances have been obtained for the APRS payload and OBDH implemented modules. Due to the environment in space and the constraints of the Nanosatellite, different measures had to be taken when the system was designed.

TABLE OF CONTENTS

1. INTRODUCTION

As the satellite industry becomes more commercialized, large expensive satellites will give way to multiple inexpensive Nanosatellites. However, the requirements of Nanosatellites are significantly different from their ancestors, and new design techniques are needed to meet these evolving requirements. The unique requirements of the store-and-forward APRS Nanosatellite demonstrate this evolution. The design needs to be relatively inexpensive while at the same time computationally robust. It must support the space environment by reducing susceptibility to radiation and thermal effects. It should provide the necessary interfaces for each subsystem and consume limited power. We designed [27] a low-complexity AFSK and GMSK modems for a satellite communication link where the constraints on satellite power budget, antenna size, and data rate are such that the available transmit power, aboard the Nanosatellite, is 1 Watt.

In this paper, we will describe in details the main considerations and solutions chosen during the design of the integration of APRS store-and-forward Payload and OBDH subsystems using the same unit based on the fixed point DSP. The design was split in two parts: The Hardware and Software Architectures.

2. SATELLITE MISSION DESCRIPTION

The mission of the Nanosatellite is to provide various APRS services, such as mobile localization of ships and data collection from autonomous weather stations in inaccessible sites using a store-and-forward payload, as shown in Fig. 1. The APRS was developed by Bob Bruninga [1]. The Cubesat project [2] aims to send a satellite, with dimensions 10x10x10 cm and mass one kilogram into Low Earth Orbit (LEO). The orbit, with altitude of 650 Km, will be achieved by means of low cost dedicated launch using Dnepr vehicle Launch [3]. The Nanosatellite uses the AX.25 [4] packet radio protocol, at 1200 bit/s for APRS payload, and at 4800 bit/s for Telemetry/Telecommand (TM/TC) operations.

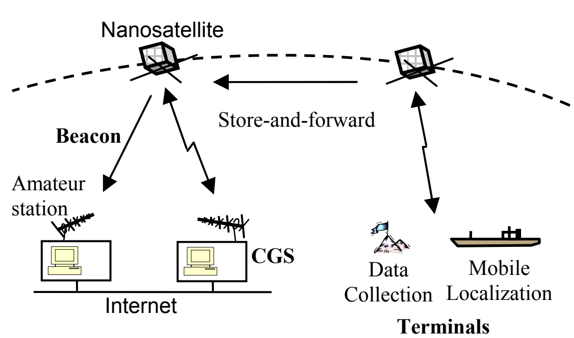


Figure 1. APRS services using LEO Nanosatellite

1
[1] 1-4244-0525-4/07/$20.00 ©2007 IEEE
[2] IEEEAC paper #1573, Version 3, Updated December 28, 2006

The satellite has three modes: Store-and-Forward APRS mode, Beacon mode and Command mode.

The AX.25 protocol provides Connection-Oriented and Connectionless transmissions. The connectionless transmission method will be used in beacon mode, whereas the connection-oriented method will be used in store-and-forward APRS and Command modes.

In store-and-forward APRS mode, the terminals use the slotted Aloha multiple access [6] to send their messages to the Nanosatellite, which stores the correct messages in an on-board storage system, and delivers this to the destination Central Ground Station (CGS) in a later time. Between the storage and the retrieval of the message, the Nanosatellite moves around its orbit and the earth rotates on its axis. These movements change the location of the satellite's footprint, and the Nanosatellite effectively carries the message from terminals all over the world to the CGS station.

In the Beacon mode, the Nanosatellite sends periodically the basic telemetry (Temperatures, currents and Battery voltage) to the earth to know the life of the Nanosatellite. This telemetry is encoded in an APRS Telemetry format. A CGS station could gather data from the radio amateurs all over the world. The gathered data will be archived to a database and be accessed by Internet.

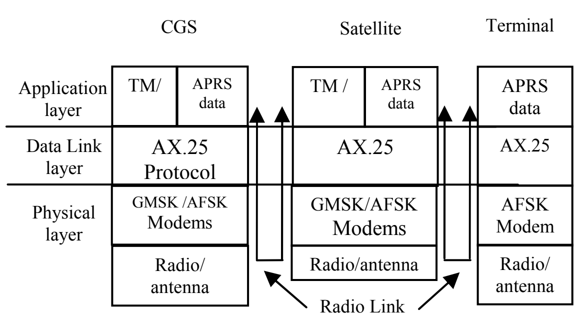


Figure 2. Satellite Communication Layers

In the Command mode, the CGS sends telecommand to the OBDH aboard the Nanosatellite and receives the housekeeping data (the major telemetry) from the OBDH to know the health status of the Nanosatellite.

The Nanosatellite has two communications channels (Fig. 2), which can be multiplexed over the same transceiver. The first channel carrying the TM/TC data (the application layer) between the Nanosatellite and the CGS station whereas the second channel carrying the APRS data (the application layer) between the satellite and the Terminals. On bottom of the application layer, we used the AX.25 protocol, which is an amateur radio adaptation of the ITU-T X.25 protocol. The AX.25 is used in APRS packet radio networks with AFSK modem, and supports the requirements for amateur satellite communication. The AX.25 packets are sent over the radio links (the Physical layer), which use AFSK (Audio Frequency Shift Keying) for APRS data, and GMSK [7] (Gaussian Minimum Shift Keying) for TM/TC data, over the 145 MHz (uplink and downlink) VHF band.

At VHF frequencies the antennas, receivers and transmitters for both the ground and the space segment, are readily available and inexpensive [10]. The doppler shift, is about ±3 KHz maximum for the Nanosatellite parameters (altitude, frequency band) [8], which can be ignored.

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

Table 1
Budget Link

Terminal Parameters	
Antenna gain	0 dBi
Transmitted power	5 W
Antenna Feed Loss	0.5 dB
T_{syst}	2000 k
CGS Parameters	
Antenna gain	13 dBi
Transmitted power	5 W
Antenna Feed Loss	1,5 dB
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 APRS Mode	
E_b/N_0 estimated	21.42 dB*3
Margin	7.82 dB
Downlink Link Margin for APRS Mode	
E_b/N_0 estimated	18.42 dB*3
Margin	4.82 dB
Uplink Link Margin for Command Mode	
E_b/N_0 estimated	26.9 dB*3
Margin	13.3 dB
Downlink Link Margin for Command Mode	
E_b/N_0 estimated	23.9 dB*3
Margin	10.3 dB

*1 Free Space Loss, FSL is calculated by [28]:

$$\mathrm{FSL}_{db} = 32.45 + 20\log\left(\mathrm{D}_{km}\right) + 20\log\left(\mathrm{F}_{MHz}\right) \quad (1)$$

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

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

*2 Required E_b/N_o gives an operating bit error rate of $2x10^{-5}$ [29], assuming non-coherent demodulation of FSK.

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

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

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

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 negligable by using VHF transceiver, and the mutipath effect is also negligible by using low data rate.

To enhance the radio link in Command mode, between the CGS station and the Nanosatellite, we added a Forward Error Correction (FEC) by using convolutional coding and viterbi decoding [11].

3. Nanosatellite Subsystems Design

3.1 Subsystems Requirements

To cut down the cost of the Nanosatellite, all the nonessential elements have been suppressed. Both an active attitude control and pointing system have been avoided by using omnidirectional antenna and having no elements (sensors) that could need some kind of pointing or positioning. The satellite will spin around its main inertial-momentum axis and will be passively orientated according to the Earth's magnetic field. However, in the case of early launch cubesats, the cubesats adopted permanent magnet as passive attitude control still alive and operated well. On the contrary, other cubesats incorporated active attitude control occurred raise many problems and was unstable. If one would think of the Earth as a giant bar magnet, then a parallel four bar magnet attached to the Cubesat would try to point towards the direction of the opposite pole of the Earth. Finally the satellite can flight parallel to magnetic field.

Therefore, thermal control is carried out with thermal conducts instead of active elements [14]. The solar cells will be attached to the satellite 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.

An important issue to consider, which affects all electronic devices in space, is radiation. Radiation can lead to various types of problems. These problems range from operational malfunctions to physical damage of the devices. Radiation effects can be broken into two categories: total radiation dose and Single-Event Effects (SEE) [30]. Using radiation models developed by NASA, the studies determined that the Nanosatellite will experience a total radiation dose of approximately 25 rads. Typical CMOS technology devices can survive nearly 5 krads before physical damage occurs [15]. However, single-event effects are significantly more hazardous to the Nanosatellite and can result in either Single Event Upset (SEU) or Single Event Transient (SET) [31] or Single Event Latch-up (SEL). SEU effect is an internal device memory change that can cause erroneous operation and SET is a transient voltage pulse that can cause logical output change. 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. The studies estimate that the Nanosatellite will experience approximately four bit-errors per day, and only a small percentage of these will result in a latchup condition. If one in a thousand events cause latch up, the Nanosatellite will observe less than two latch-ups per year [15].

Operating devices at slower frequencies reduce the probability of SET effect. Designs should function using the slowest clock rates possible that yield acceptable performance. Based on Nanosatellite mission parameters (orbit, inclination, life expectancy), the total dose radiation can be considered negligible by shielding electrical components, but SEE effects need to be considered throughout the design.

3.2 The OBDH Subsystem Design

The OBDH subsystem is one of the spacecraft subsystems like Tracking Telemetry Command Subsystem (TTCS), Electrical Power Subsystem (EPS), Attitude Control Subsystem (ACS) and Thermal Control Subsystem (TCS). All these subsystems (Fig. 3) belong to spacecraft and offer their services to the payload.

The OBDH subsystem is considered as an interface between the CGS station and the Nanosatellite subsystems. As illustrated in the Fig. 3, the OBDH modules include modulator, demodulator, telemetry encoder, telecommand decoder and on board computer which controls the satellite bus. The satellite bus is chosen distributed linear bus because of several advantages. Instead of having to run separate data lines from the OBDH to each subsystem, the subsystems can now be daisy-chained on the Master-slave bus (see Fig. 3). It is very easy to add, remove, or change the attached subsystems, with minimal or no changes to other nodes. It requires a well-designed communication protocol governing when processors can transmit data over the bus.

In the satellite modular architecture [14, 16], 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 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. In this paper, we will integrate APRS store-and-forward Payload and OBDH subsystems on the same unit.

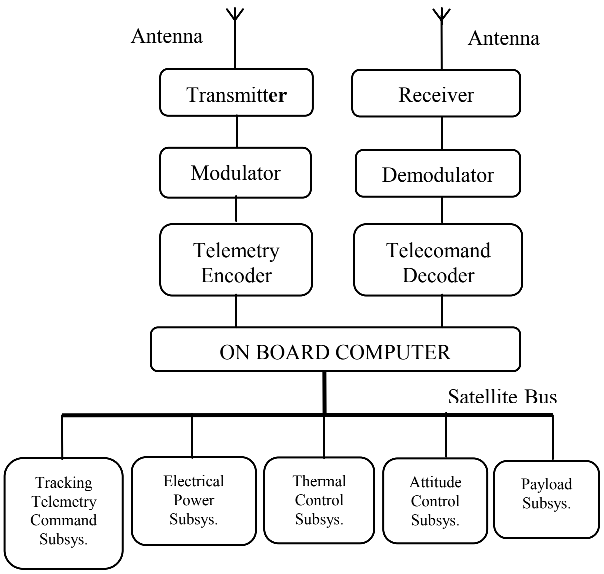


Figure 3. Satellite Modular Architecture with distributed linear Bus

The satellite bus, which is similar to Local Area Network (LAN) aboard the satellite, connects the different subsystems to the OBDH subsystem. 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). The satellite bus was designed to consist internally of two subsystems TAS (Telemetry Acquisition Subsystem), instead of TTCS, and EPS.

A trade study, comparing three buses, was conducted as seen in Tab. 2. 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 needs. In the case of two slaves, the scalability of SPI bus is almost the same as I²C and CAN buses. According to Tab.1, SPI bus won with the high score compared to CAN and I²C buses because of its simplicity. Thus, the SPI bus was selected as the satellite bus.

Table 2
Master-Slave Buses study in case of two slaves

	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

To immunize against SEU effect, the application program is loaded in the external 8K word 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. Although rare, latch-up situations present the greatest hazard to the OBDH subsystem. A latch-up event results in an abnormally high current-flow through a device.

The EPS design must be able to detect high current-flow through each board and turn off the board accordingly. SEL protecting circuit or smart fuse, which shuts off the electrical power automatically to a device, is the only feasible remedy for clearing a latch-up condition. The SEL protecting circuits lie between the power bus connection and the power pins of the boards. When the current-flow into the board rises above a user's predetermined threshold, the SEL protecting circuit disables instantaneously power to the board. The EPS microcontroller measures cyclically each board current-flow and determines which board needs to be reset. Finally, the EPS microcontroller provides a reset signal to the SEL protecting circuit and logs the event. In the event that the OBDH microcontroller countermeasures fail, a Fail Safe Circuit is used to perform a satellite reset at regularly scheduled intervals. A system reset can also be commanded after a telemetry session.

3.3 The Store-and-Forward APRS Payload Design

The Nanosatellite carries aboard a store-and-forward APRS payload operating in Amateur (digital) mode with VHF uplink and downlink operating in amateur frequency band. The store-and-forward APRS payload provides digital communication facilities between a number of terminals [5], situated all over the world, and the CGS station. The whole process of the store-and-forward payload can be seen in the flowchart shown in Fig.4. The design of the APRS store-and-forward payload was designed to take orders. The payload always listens to the receiver. The preamble, that is the beginning of a packet, must be detected firstly. The synchronization will be achieved after the preamble is detected. The received message will be tested if there is error in the message. The erroneous message will be rejected and the correct message will be decoded. The CGS station requests to read memory and download APRS data.

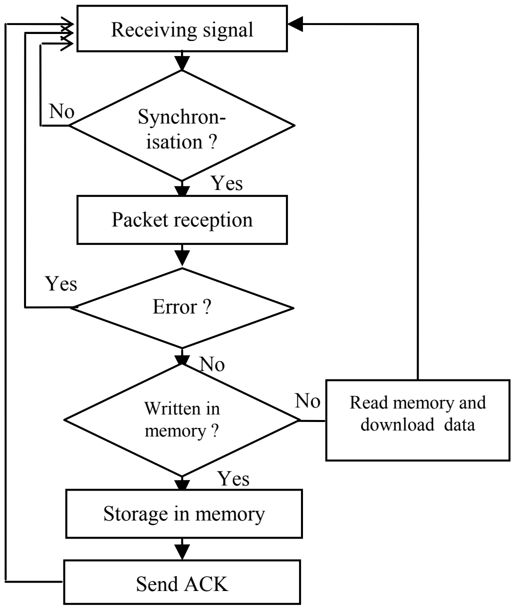


Figure 4. Store-and-Forward Payload flowchart

When the Nanosatellite receives correct APRS packets from terminals, the packets will be stored in SRAM memory, and then the Nanosatellite sends an acknowledgement because when the terminal transmits its APRS 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. An acknowledgement indicates a complete success of the transmission.

Due to SEU effect, the data will be encoded by hamming algorithm [17] before being stored in SRAM memory to correct errors affected by SEU effect. The data stored in SRAM memory will be downloaded by the CGS station after being decoded by hamming algorithm.

The SRAM memory storage has to be small for many reasons. The capacity of the store-and-forward APRS system using the Nanosatellite is limited [6]. Also, the execution time of hamming algorithm periodically for high memory storage consumes a lot of energy, which is limited for the Nanosatellite. The SRAM memory is designed to consist of 1Mb which contains the APRS data.

The use of Commercial Off The Shelf (COTS) components combined with mission specific tailored shielding, error correction and latchup protection can be applied on a wide spectrum of mission scenarios [18].

Mass, size and power consumption constraints must be fulfilled. As a result, integrated circuits based on CMOS technology have been used. We have chosen the DSP microcontroller TMS320C542, from Texas Instrument, connected to storage capacity memories. The DSP is especially chosen for its ease of implementing the physical and the data link layers, and also for the availability of many reports applications, which reduce the design time.

4. Hardware Architecture

The hardware architecture is composed of three main parts (Fig. 5): an APRS OBDH unit, built around a DSP TMS320C542 microcontroller [21], a satellite bus designed around the SPI-bus [20] and a VHF transceiver with associated antenna.

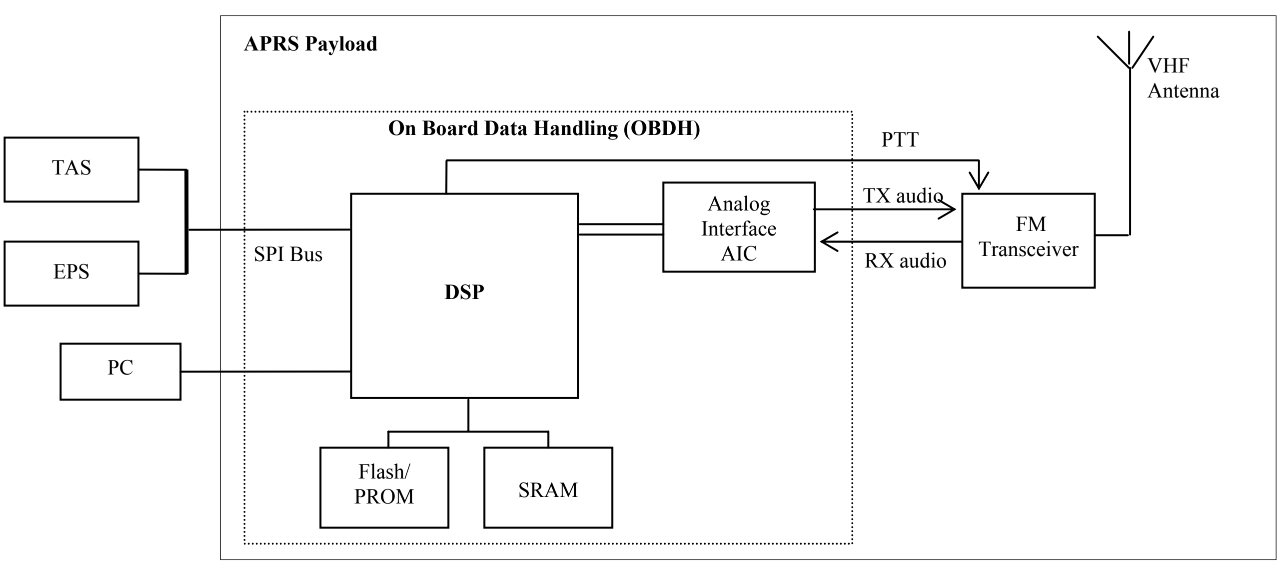


Figure 5. Hardware Integrated Architecture of Store-and-forward APRS Payload and OBDH subsystems

The internal DSP has parallel port HPI and two bidirectional serial ports BSP and TDM. The radio communication data stream from the DSP processor uses TDM interface which is connected to VHF transceiver through the AIC (Analog Interface Circuit) interface. The HPI parallel port is used to connect the APRS OBDH unit to a PC or laptop for booting application programs and reconfiguration.

The BSP serial port was dedicated to interface the OBDH to the Nanosatellite subsystems. The power can be supplied to the APRS OBDH unit with a nominal 5V battery. Due to the environment in space and the constraints of the Nanosatellite, different measures had to be taken when the system was designed.

4.1 The APRS OBDH Unit

The APRS OBDH unit as illustrated in Fig. 5 is a logic board, which controls the whole operations of the payload, built around a 16 bits DSP TMS320C542 [21] (efficient power consumption), from Texas Instrument, running at 40 MHz with 10K words RAM dual access and 2K words ROM for boot software. The DSP TMS320C542 is chosen for its ease of use, processing power, and intrinsic features and size.

The design has the advantage of implementing AFSK and GMSK modems as software, therein eliminating the need for external modem hardware needed by other systems [22], and minimizing the dimensions of the logic board which must be less than 10cm x 10cm.

The OBDH unit will be built in two versions; an engineering model and a flight model. The engineering model is used for prototype and the flight model for launch. The prototype was built in order to test the functionality of the hardware and debug the software. For the prototype Printed Circuit Board (PCB), flash memory was used instead of PROM. This is done in order to allow the developing and testing of software without having to change PROM each time new software were to be tested. The SRAM memory 64k x16 stores the APRS data. A Hamming code (7, 4), which generates extra bits to be stored together with the data in this memory, is made to correct errors in the data stored in the SRAM memory. These extra bits allow the OBDH to detect and correct a limited number of errors when we read the stored data. The Hamming code (7, 4) is able to correct 1 error in 4 bits or 2 errors in byte.

The TLC320AC01 [23] (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 TMS320C542. Use of this AIC greatly simplified the hardware necessary to provide an analog interface to the DSP development system. For the flight model, we will use TLC320AC021[23], which is specified for the –40°C to 85°C temperature range instead of using TLC320AC01, which is specified for the 0°C to 70°C temperature range.

All the components on the 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.

4.2 The Satellite Bus

The satellite bus is based on the SPI specification. As such, all subsystems on the data bus have a built-in SPI microcontroller. The SPI bus was supported, by using the BSP port (buffered serial port) of the DSP, with no hardware modifications are needed to be taken, with transfer rate of 2 Mbps.

The SPI bus consists of four signals: master out slave in (MOSI), master in slave out (MISO), serial clock (SCK), and active-low slave select (/SS). As a master/slave protocol, communications between the master and selected slave use the unidirectional MISO and MOSI lines, to achieve data rates of 2Mbps in full duplex mode. A master (in this case the DSP) is the device which initiates a data transfer on the bus with the selected device and generates the SCLK clock signal to permit transfers. The SPI protocol allows for four different clocking types, based on the polarity and phase of the SCKL signal. It is important to ensure that these are compatible between master and slaves. The SPI-bus is configured, monitored and controlled by setting different internal registers: (BSPCE, BSPC) [21].

4.3 The Transceiver

A COTS amateur radio transceiver was adopted to be the main flight radio due to power, weight, and time constraints [24]. The transceiver, integrated inside the payload, operates in amateur VHF band and consists of the "guts" of a low cost Yaesu VX-1R [22], arguably one of the smallest and lightest handhelds on the market. The radio is two stackable double-sided PCBs measuring approximately 5x5cm. Power is supplied from 5V bus for 1Watt RF power which achieves a positive budget link [9]-[10]-[11]. Current consumption for the receiver and transmitter is 150mA and 400mA, respectively. Only slight modifications will be required to make this component space worthy. The transceiver is interfaced with the OBDH APRS unit by means of AF (Audio Frequency) signals and PTT (Push To Talk) command signal as illustrated by Fig. 5. AF signals consist of transmitting and receiving 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 AFSK/FM signal output from the transceiver is fed to an omnidirectional antenna.

4.4 The antenna

We have previously said that the attitude control of the Nanosatellite is based on the Earth magnetic field. As a result, the pointing direction of the satellite depends on its latitude. Therefore, the Nanosatellite antenna must be omnidirectional to cover the earth stations placed in all latitudes. For the Nanosatellite antenna, the turnstile antenna [12]-[13] 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 Nanosatellite and to the long ¼ wavelength transmission line at VHF frequencies; implementation using lumped elements [25] is desirable. To make easier the deployment of the antenna, we use measuring tapes as the four elements that consists the circularly polarized antenna.

5. Software Architecture

The designed software is made to boot-up and afterwards serve the OBDH. The Fig. 6 shows the software architecture of the OBDH APRS unit.

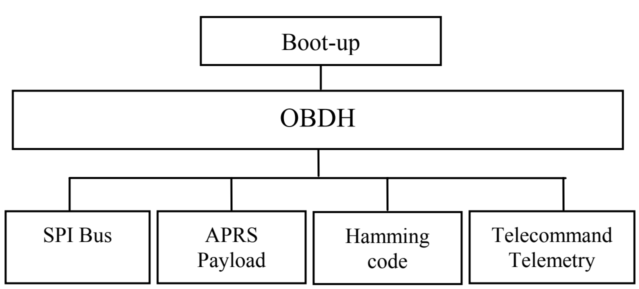


Figure 6. Software architecture of OBDH APRS

As seen on the Fig. 6, the OBDH could execute different functions on the OBDH APRS unit. These functions acted as hardware drivers for the OBDH. By using these drivers different tasks are made transparent to OBDH.

5.1 The Boot-up sequence

One of the most vital functions of the OBDH is to configure the DSP. The DSP is designed to always start by contacting the PROM module.
Once the satellite has been separated from the launcher, the EPS is power supplied and reset the DSP. After that, the EPS started a timer and the DSP continued its boot-up. This timer should then be reset in specific time intervals. The DSP loads the program code stored in the PROM to the protected RAM for a faster execution. The first part of the PROM module is made to contain the basic software needed to set up the different parameters internally in the DSP.
When the DSP had finished the boot-up and started the OBDH, the first thing the OBDH do is to write a RESET command to the EPS. This is done using the SPI bus. When the EPS received the command, it reset its timer. If the boot-up of the OBDH failed, this timer would timeout. This indicates to the EPS that the current boot-up has failed. EPS would then turn off the OBDH and restart it. This routine would be continued until the OBDH boots successfully. After the activation, the OBDH is designed to become a passive unit taking orders from the CGS station.

5.2 The NSPI Protocol

The Nanosatellite SPI protocol (NSPIP) was designed on top of the SPI-bus. The complete protocol therefore consisted of the command and telemetry packets designed especially for the Nanosatellite. The master initiates a NSPIP exchange by transmitting a command packet. As seen in Fig. 7, the NSPIP command packet consists of 3 fields: Command, Module number, and an 8-bit checksum. The module number identifies one sensor or a set of sensors depending on the type of the NSPIP command packet.

Command	Module number	Checksum
1 byte	1 byte	1 byte

Figure 7. NSPIP command packet format

Subsystems reply by telemetry packets, which start with an 8-bit length field, followed by a variably sized data portion and finally a checksum byte for simple error detection. The errors could be introduced by SET effect. The seven (7) least significant bits of the length field encode the integer number of bytes in the data packet, including the length byte itself and the trailing checksum byte. This format is illustrated in Fig. 8.

Length	Data packages	Checksum
1 byte	1-126 byte	1 byte

Figure 8. NSPIP telemetry packet format

The checksum was calculated in the following way:

$$[0xFF] \ - \ \sum[data] = [checksum] \tag{4}$$

All the data packages and the length sent in one transmission are added together, the 8 bit result of this calculation is then subtracted from 0xFF, and this yielded the CRC checksum. The data is checked in the following way:

$$[checksum] \ + \ \sum[data] = [0xFF] \tag{5}$$

The sum of received data is added with the checksum. If the result is different from 0xFF the data is corrupted and would be retransmitted.

When the satellite receives a telecommand from the CGS station, the OBDH will be in the Command mode. In this case, the OBDH distributes NSPIP command to a specific satellite subsystem. For collecting housekeeping data, the TAS subsystem would deliver all its housekeeping data to the OBDH. Then, the housekeeping data will be formatted in AX.25 packet and will be transmitted to the CGS station. Also, The NSPIP commands serve to write data to a unit or to read data from a specific sensor in the unit. For the transmission and the reception of any telecommand, the OBDH performs GMSK modulation and GMSK demodulation functions respectively. The OBDH reads periodically basic telemetry from a specific unit. The basic telemetry will be formatted in APRS telemetry beacon and will be transmitted to the radio amateur stations on the ground using AFSK modulation.

5.3 The DSP Implementation [27]

In this paper, to cut down the cost of the Nanosatellite, we give a particular attention to software optimization due to use of small amount of 8K word PROM memory containing the application program. 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. By an optimal software design that takes the maximum advantages from DSP architecture facilities, good real time experimental performances have been obtained for the APRS payload and OBDH 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 [19] and TMS320C5416 DSK kit [26], which are compatible.

6. Conclusions

As the satellite community transitions towards inexpensive distributed Nanosatellites, new methodologies need to be employed to replace traditional design techniques. This paper will contribute to the development of these cost saving methodologies. The goal of the integrated store-and-forward APRS Payload and OBDH subsystems design is to minimize component expenditures while still providing the reliability necessary for mission success.
Associating low cost ground terminals [5] with a low cost Nanosatellite allows developing countries to access space communications with a very economical system. The present work, dealing with the design of the integrated store-and-forward APRS Payload and OBDH subsystems, 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 OBDH APRS unit is small device that can be mounted on any small satellite platform to serve applications such as mobile localization and data collection. By using a single Nanosatellite and low-cost communications equipments, store-and-forward systems can be kept at the extreme low end of the satellite communications cost spectrum.

Acknowledgement

This work was supported by the University Network of Space Techniques and Sciences (RUSTE), Morocco.

References

[1] Bob Bruninga “A Description of APRS by WB4APR, Packet Users Notebook”, CQ Magazine, Dec 1993.

[2] Waydo, S., D. Henry, and M. Campbell, “CubeSat Design for LEO-Based Earth Science Missions,” IEEE Aerospace Conference Proceedings, Vol. 1, pp. 435-445, 2002.

[3] M. Obland et al. "The Montana State University NASA Space Grant Explorer-1 Science Reflight Commemorative Mission", 15th Annual AIAA/USU conference on small Satellites, Utah, USA, 2001.

[4] TAPR, AX.25 Link Access Protocol for Amateur Packet Radio, TAPR, version 2.2, 1997.

[5] E.B. Zantou, A. Kherras, "Small Mobile Ground Terminal Design for a Microsatellite Data Collection System " Journal of Aerospace Computing, Information, and Communication, vol.1 no.9, 2004.

[6] A. Addaim, A. Kherras, E. B. Zantou, A. ER-RADI “ Design of APRS Network using Low Cost Nanosatellite”. The Seventh IFIP International Conference on Mobile and Wireless Communications Networks (MWCN 2005), Marrakech, Morocco, 2005.

[7] Sundberg, C., “Continuous Phase Modulation”, IEEE Communications Magazine, vol. 24, no. 4, April 1986.

[8] A. Jamalipour, “Low Earth Orbital Satellites for Personal Communication Networks”, Norwood, MA: Arthech House, 1998.

[9] G. Maral, M. Bousquet, «Satellite Communication Systems », John Wiley & Sons; 4th edition, May, 2002.

[10] J. Paffet, T. G. Jeans, J. Ward, "VHF-Band Interference Avoidance for Next-Generation Small Satellites," 12th AIAA/USU Conference on Small Satellites, Logan, Utah, USA, 1998.

[11] J. Proakis, « Digital Communications » McGraw-Hill, Second Edition, 1989.

[12] Thomas A. Milligan, "Modern Antenna Design", Wiley, 2nd Ed., June, 2005

[13] N. Bornman, K. Palmer "VHF Antenna System for a microsatellite (SUNSAT)", Proceeding of Symposium on small satellites and control systems, Stellenbosch, 1994.

[14] R. Wertz, W. Larson "Space Mission Analysis and Design" 3rd Edition, Microcosm, 1999.

[15] Pisacane, V. L., and R. C. Moore, "Fundamentals of Space Systems", New York: Oxford University Press, 1994.

[16] CNES, "Cours de Techniques & Technologies des Véhicules Spatiaux", vol. 3, 1998.

[17] O. Pretzel, "Error correcting codes and finite fields", New York: Oxford University Press, 1992.

[18] Bernd Gerlach, Fritz O. Gliem "Miniaturized Data Processing Unit for Use on Small Satellites", 12th AIAA/USU Conference on Small Satellites, 1998.

[19] Texas Instruments "DSKplus User's Guide", SPRU191, 1997.

[20] Motorola "MC68HC11 Reference Manual", Prentice Hall, 1989.

[21] Texas Instruments "TMS320C54X DSP: CPU and Peripherals", SPRU131G, 2001.

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

[23] Texas Instruments "TLC320AC01 Data Manual Single-Supply Analog Interface Circuit" SLAS057D, October 1996.

[24] Stephen Horan, "Preparing a COTS Radio for Flight - Lessons Learned from the 3 Corner Satellite Project", 16th. Annual/USU Conference on Small Satellites, 2002.

[25] Inder Bahl, "Lumped Elements for RF and Microwave Circuits", Artech, 2003.

[26] Texas Instrument "TMS320VC5416 DSK Technical Reference," 2002.

[27] A. Addaim, A. Kherras, E. B. Zantou," DSP Implementation of Integrated APRS Payload and OBDH Subsystems for Low-cost Nanosatellite", Proceedings of the 4S symposium, Small Satellite Systems and Services, Sardinia, Italy, 2006.

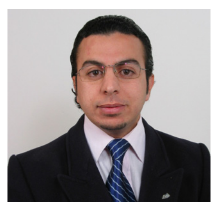
[28] Gérard Maral, Michel Bousquet, " Satellite Communication Systems" John Wiley & Sons; 4 edition (May, 2002).

[29] John G. Proakis, "Digital Communications". McGraw-Hill, Second Edition, 1989.

[30] Sophie Duzellier "Radiation effects on electronic devices in space", Aerospace Science and Technology, Volume 9, Issue 1, January 2005.

[31] C. Poivey, S. Buchner, J. Howard, K. Label, "Testing Guidelines for Single Event Transient", NASA Goddard Space Flight Center, 30 June, 2003.

BIOGRAPHY



Adnane Addaim *was born in Casablanca, Morocco, in 1975. He received the bachelor degree in Electronics from the Faculty of Sciences Ain-Chok, Hassan II university, in 1998, and the Master degree in Aerospace from Mohammadia School of Engineers (EMI), Mohammed V university, in 2001. He is currently working toward the Ph.D. degree with Electrical Engineering Department of EMI.*

From 2002 to 2003, He was employed at SIEMENS.AG, as Microwave Engineer, responsible of the implementation of the supervision network of the Meditelecom GSM Network, in Morocco. He is member of Centre for Space Research and Studies (CRES). His current research interests include signal processing, satellite communication systems and access satellite networks.

Abdelhaq Kherras *was born in Marrakech, Morocco, in 1949. He received the master degree in EEA (Electronic Electrotechnic and Automatic) from Paul Sabatier University, Toulouse, the doctorate thesis in USTL, Montepellier university, in 1980, and the Ph.D. degree from the Liege University, Belgium, in 1990.*

Since 1982, he has been a professor of Electronics in Electrical Engineering Department of EMI. He is currently the director of Centre for Space Research and Studies (CRES), and National Coordinator of University Network of Space Techniques and Sciences (RUSTE). His current research interests include space technology and satellite communication systems.

El Bachir Zantou *was born in Beni-Mellal, Morocco in 1966. In 1988, he graduated from the Aviation Royal School (ERA) with an engineer degree in the aeronautics. He received the Engineer degree in electronics and communications from the EMI, in Rabat in 1992, and the Master degree in space communications from Supaero the ENSAE, in Toulouse, France in 1994. He obtained the Ph.D degree from the EMI in May 2006.*

He is member of the Royal Centre for Space research and Studies (CRERS). He participated in the project of the Moroccan Microsatellite Maroc-Tubsat launched by the CRERS in 2001, and developped the user ground terminals. His current research interests include satellite communication systems and LEO network simulations.