

A Flexible and Autonomous Communication Subsystem for Constellations of Small-Satellites

Juan A. Fraire, Esteban G. Kocian
 Servicios Tecnológicos Integrados
 Av. Fuerza Aérea 3939, Córdoba, Argentina; +54-9-3512446010
 jfraire@sti-tech.com.ar

ABSTRACT

During the past 20 years, space communications technologies have shown limited progress in comparison to commercially available solutions on Earth. Such an evolution was not mandatory as Earth-to-Satellite (ESL) communication architecture is still conceived as a telecommand in the uplink, and telemetry and payload data on the downlink. However, communication systems designed to cope with this asymmetry are not suitable for inter-satellite (ISL) links which require a non-hierarchical and unassisted operation. In particular, ISLs can be of significant importance with the advent of small-satellite as they allow achieving common mission objectives by taking advantage of a distributed architecture. Nevertheless, since small-satellite platforms are resource-constrained, accounting with separate ESL and ISL subsystems can sometimes be prohibitive. Therefore, we envisioned the design of an autonomous and flexible communication subsystem capable of operating over a wide variety of scenarios. To this end, we present the recently developed STI-PRX-01: a CCSDS Proximity-1 protocol modem that can operate with different radio-frequency front-ends under varying communication parameters which can be autonomously negotiated upon link establishment between spacecrafts (for ISLs), or ground segment (for ESLs). Between these parameters we highlight frequency channel, modulation, datarate, error-correcting codes, transmitting power, etc. Furthermore, the chosen protocol allows for unassisted recovery of lost or corrupted frames enabling the system to efficiently operate over marginal and variable link budgets. In this paper, we describe the architecture that supports this error-recovery mechanisms and dynamic parameter negotiation to later provide an overview of the hardware of STI-PRX-01. Finally, we supply and analyze the first measurements obtained from the engineering model in order to demonstrate the benefits of this novel communication approach.

INTRODUCTION

Distributed small-satellite constellations are emerging as an alternative to traditional big single satellites missions which are expensive to develop, launch and operate¹. In particular, a constellation offers numerous advantages such as wider coverage, payload redundancy or combination, distributed storage and processing, incremental launching methodologies, on-orbit satellite replacement, among others that increase mission reliability² and performance³. However, in order to be able to implement such missions, an appropriate communication subsystem must embrace unprecedented autonomy and flexibility characteristics.

In general, traditional satellite transponders either assume that one of the end nodes is operated from a ground station on earth, or that an intermediate node performs as a remotely managed bent-pipe or under a relay scheme such as TDRSS. Furthermore, a few constellation missions embraced inter-satellite links (ISL) such as Iridium, for which the protocol and link properties remains proprietary⁴. As a result, autonomy is not a common feature in exiting satellite

communication systems, especially for Earth-observation missions with scientific or socio-economical purposes.

In the other hand, the highly mobile environment onto which Low-Earth Orbit (LEO) satellites operate also imposes major challenges to communication subsystems. In particular, the orbital dynamic of LEO-based constellations might develop on a wide variety of radio-frequency (RF) ranges and relative antenna pointing situations. Therefore, the on-board communication subsystem must be able to accommodate a self-adapting feature in order to dynamically configure the transponder to make the best out of the inter-spacecraft communication opportunity. Indeed, such flexibility also encompasses the better usage of Earth-to-Satellite (ESL) links.

Since conventional and available small-satellite transponders and protocols such as CCSDS Telemetry and Telecommand (TM/TC) lacks of the autonomy and flexibility required for conforming satellite constellations, a different approach is required to fulfill

this need⁵. Unfortunately, since the LEO constellation mission characteristic (distance, node density, reliability, power consumption, data-rate, etc.) can drastically differ from one another, there is not a clear consensus nor a general agreement on which is the best ISL approach⁶.

On the one hand, existing protocols such as IEEE 802.11 (*WiFi*) provide enough flexibility and autonomy, but it is designed to perform in short distances. Despite previous work on extending *WiFi* range to several Km exists⁷, they provoke a significant loss of performance due to the conversational overhead. Others protocols such as HDLC have also been considered, but they lack of the required reliability for space applications⁵. Since modifying existing standards is too risky for the general space application, the CCSDS recommends the usage of Proximity-1 which specifically emerges as a long-range space-oriented protocol with sufficient autonomy and flexibility⁷.

Despite Proximity-1 protocol was successfully implemented⁹ and validated between Mars orbiters and landers¹⁰, to the best of our knowledge there is no available experience on this protocol application on LEO constellations. Therefore, Servicios Tecnológicos Integrados (STI) and the Argentinian Space Agency (CONAE), devoted to the observation of the Earth, envisioned the development of STI-PRX-01 modem for providing autonomous and flexible ISL and ESL communications to future constellations of small-satellites. At the time of writing, we account with two functional prototypes of the modem which we describe and analyze throughout this document (Figure 1).

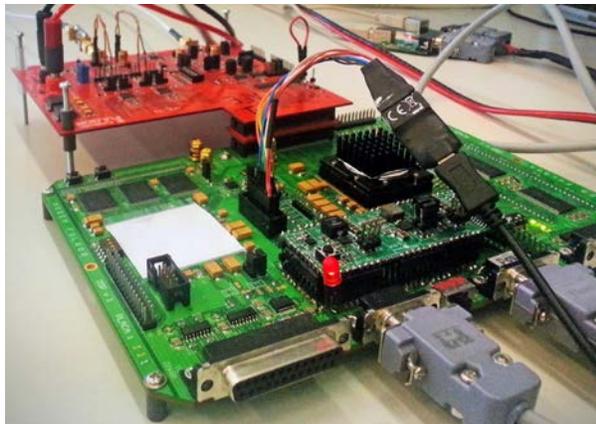


Figure 1: Proximity-1 modem prototype

Proximity-1 Overview

The CCSDS Proximity-1 protocol is intended for cross-support purposes on proximity links. Proximity links are defined as being short range, bidirectional (full and half-duplex), fixed or mobile radio links to

communicate among landers, rovers, orbiting constellations, and orbiting relays. Proximity links have short time delays, moderate signal power, and short, independent sessions⁸.

Within the OSI communications model, Proximity-1 fits in the physical and link layer as it not only defines radio-frequency specifications such as modulation and coding, but also session establishment and termination, a variable frame size, quality of service and a negotiable frequency division multiple access (FDMA) medium access control (MAC) scheme. Figure 2 illustrates this organization.

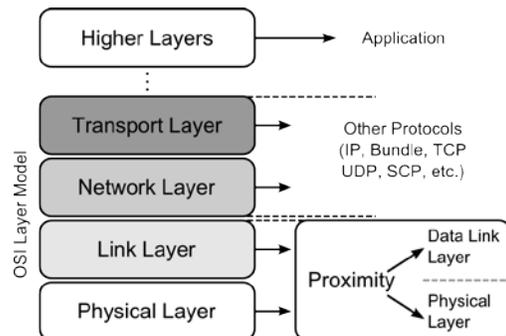


Figure 2: Proximity-1 within the OSI Layer Model

Proximity-1 supports two types of data transmission services on the link layer level: sequence controlled (SEQ) and expedited (EXP). The former provides a reliable, in-order, gap and duplication free frame transfer via a go-back-n ARQ strategy to guarantee that no data is lost nor corrupted within the session. The reception acknowledgement in Proximity-1 is transmitted by means of a Proximity Link Control Word (PLCW). On the other hand, EXP service allows for a higher priority yet unconfirmed mechanism for non-critical data or to accommodate protocols with higher layer reliability mechanisms.

On the physical layer side, Proximity-1 standard specifies a channel assignment for UHF band, leaving the S-Band channel structure as a future development. In order to improve the original datarate, we propose an S-Band frequency assignment which is able accommodate larger bandwidth channels. Also, we choose to implement an optional forward-error correction code (Low Density Parity Check or LDPC) to boost the transponder capability.

Modem Environment

The modem is intended to be implemented and used both for ISLs and ESLs in a small satellite on LEO orbits, but other platforms and scenarios might also be considered. The design assumes an on-board computer will manage and configure the modem operation (via

directives and status reports) and also provide and accept the information to send and receive respectively (Tx and Rx data). However, all link and physical operations are performed autonomously by the modem without any external intervention. Furthermore, the modem design is also suitable for implementation on a ground station for ESLs establishment. Figure 3 illustrates and summarizes the platform environment on which the Proximity-1 modem is expected to perform.

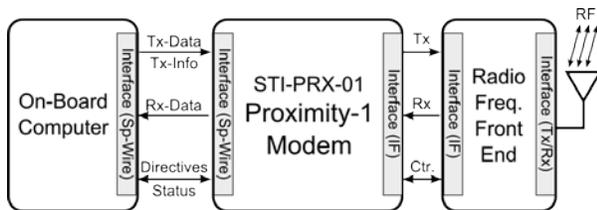


Figure 3: Modem Environment

On the radio-frequency side, the modem provides an intermediate frequency (IF) output suitable to be further up-converted to the mission-specific spectrum assignment. In particular, the current architecture targets an S-Band central frequency with the channel disposition shown in Figure 4. This is a channelization proposal to be sent to the CCSDS committee for consideration as a feasible spectrum management strategy for the yet undefined S-Band. To this end, the modem interface in IF is tuned to 100MHz and accounts for a total bandwidth of the S-Band. This implies that channel selection and isolation is performed by the modem as we further explain in the implementation section.

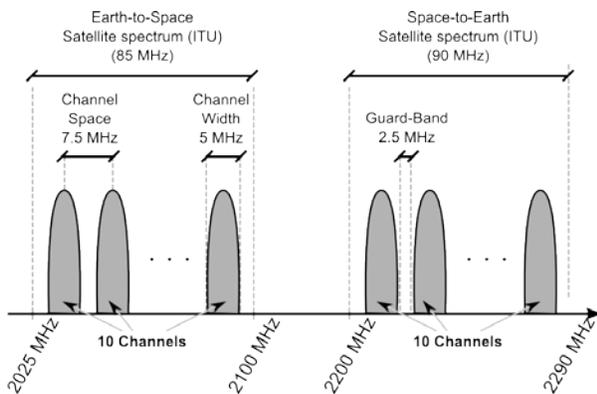


Figure 4: S-Band Channel Assignment

In the forthcoming sections we will provide an overview of the modem architecture and the specific implementation decisions. Next, we summarize several performance measurements obtained from the modem prototypes to finally draw the final conclusion and further work.

FUNCTIONAL ARCHITECTURE

In this section we provide a general architecture of the functionalities the modem was expected to accomplish. Indeed, most of them are derived from the CCSDS Proximity-1⁸ specifications from which the reader can obtain further details.

In general the Proximity-1 transponder functionality can be analyzed and described based on the layered model shown in Figure 2. In particular, the transponder embraces both Data Link Layer (DLL) and physical layer (PHY) whose activities can in turn be divided in transmitter and receiver chain. Indeed, the DLL interfaces with the on-board computer (Tx and Rx data, Tx info, directive and status) and the PHY with the RF-Front end (analog RF Tx and Rx) as shown in Figure 3 and 5.

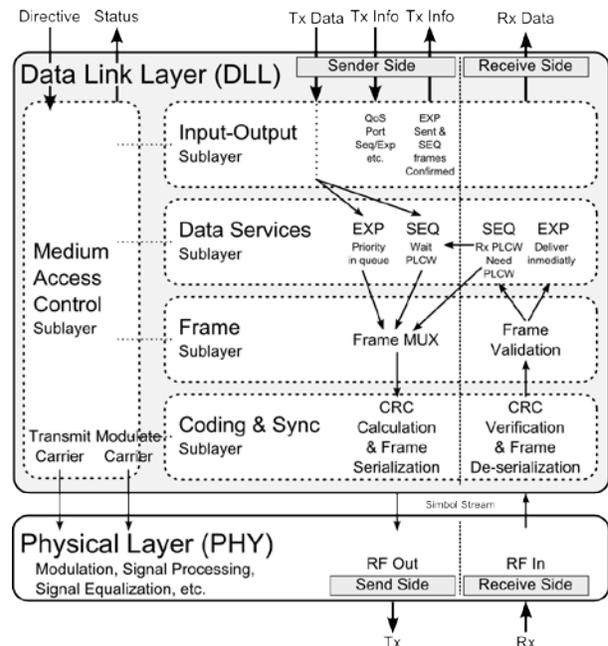


Figure 5: Modem Functional Architecture

The DLL layer is subdivided in several sub-layers: input-output (I/O), Data services (DS), Frame (FSL), and Coding and Synchronization (CS). The I/O layer is basically responsible for interfacing with the platform by accepting and formatting Tx data and informing its transmission (EXP frames) or the confirmation reception (SEQ frames) on the sender side, and for delivering Rx data. Furthermore, the modem can interpret hard-commands which can trigger specific hardware level signals for contingency purposes.

The Data Services sub-layer is probably the most critical module as it is in charge of guaranteeing quality of service (QoS) and priority frame management as per

the Tx info received through the I/O sub-layer. To achieve this purpose, the DS executes the Frame Operations Procedures (FOP-P) which is the stateful logic that waits for the reception confirmation (PLCW) on the sender side, and transmits the confirmation on the reception side. Indeed, the sender and receiver side are highly integrated in this layer to support the conversational go-back-n based protocol. The specific behavior of this reliability algorithm can be found on the Proximity-1 blue book⁸.

One step lower in the internal DLL architecture, the frame (FLS) sub-layer is in charge of sending and receiving frames from the CS sub-layer. In other words, it performs the final frame formatting operation in the DLL chain. PLCWs to be sent are created in this layer by demand of the DS sub-layer on the receiver side by means of a *need PLCW* flag. The later indicates that a confirmation frame is to be generated in order to let the modem on the other side of the link know that a SEQ frame was received and decoded correctly. During the data transfer stage, the FSL layer determines the ordering and interleaving of PLCW and data frames on the sender side, and checks frames integrity and format on the receiver. If the frame is corrupted, it is directly discarded whether if it is of type EXP or SEQ. The FOP-P logic (DS layer) will be able to recover the lost frame when a timeout is detected on the transmission modem.

Next, the C&S sub-layer is responsible for receiving frames from FSL, optionally codify them, calculate a 32 bit long cyclic redundancy code (CRC) of the frame and deliver a symbol stream to the PHY layer. At this stage the frame is structured as a Proximity Link Transfer Unit (PLTU) as shown in Figure 6. The interested reader can find specific header fields and functions on the Proximity-1 CCSDS blue book⁸. On the reception side this layer is in charge of delimiting and determining if a received frame is corrupted by using the CRC code.

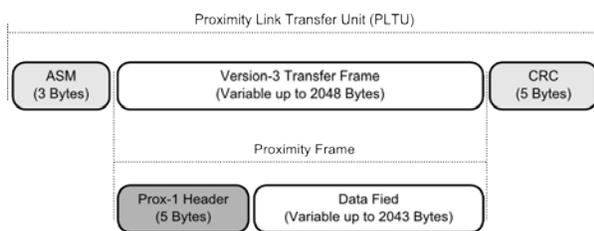


Figure 6: Proximity-1 Frame Structure

The Medium Access Control (MAC) sub-layer is the responsible for controlling the status of all the layers. In particular, the MAC manages the communication sessions by establishing, maintaining, and terminating

them, configuring the PHY layer parameters such as transmission rates, frequencies, among others.

Finally, the physical layer is the interface between protocol-specific algorithms and radio-frequency signal in the output. Indeed, since C&S sub-layer delivers a symbol stream, the PHY layer is responsible for properly condition those symbols in an analog signal on the sender side, and to effectively decode them on the receiver side. This layer is responsible of coping with analogic phenomena such as noise filtering, Doppler Effect, clock synchronisms, frequency shifts, among others.

IMPLEMENTATION DETAILS

In this section we describe how the functions previously described are implemented in the specific STI-PRX-01 architecture. In general, the design was driven by the Space Telecommunications Radio System (STRS) Hardware Architecture Standard¹¹, under the small-mission classification with particular considerations as specified below.

In particular, the implementation of the modem can be classified in three blocks: the processing module, the control module and the analog module. This rationale allows for a split reliability and performance (performability) strategy where the control module is required to provide high availability while the processing module is expected to deliver both a large logical capacity and performance. Finally, the analog module implements all analog components required to properly feed the analog to digital (ADC) and digital to analog (DAC) converters before reaching the DSP module. In general, all modules accounts with specific debugging interfaces that allows for collecting a detailed status of the algorithms both for channel status measurements and scientific purposes.

As a result, the processing module is implemented in a high-performance defense-grade RAM-based Kintex-7 FPGA from Xilinx (XQ7K325T1RF900M)¹². The critical components of the logic implementation in this FPGA (such as states machines and recursive loops) were redounded for reliability. This component will account for a LEON3-FT processor and general digital signal processing (DSP) logic which basically determines the performance of the modem. This module interfaces with a triple-modular-redundancy (TMR) radiation-tolerant SD-RAM memory structure (2Gbit capacity), the control module, and the analog front end, particularly with the digital to analog converter (DAC) on the transmitter side, the analog to digital converter (ADC) on the receiving side, and other control signals. Finally, the processing module interfaces with the on-board computer (OBC) via a SpaceWire (SW) port.

On the other hand, the control module is based on a high-reliability anti-fuse RTAX-1000S from MicroSemi¹³ which is in charge of controlling, configuring, and reporting the status of the processing module. Indeed, the Kintex-7 is configured by means of a Slave Serial Configuration mode¹² from a master device on the RTAX FPGA which verifies and transmits the bitstream from a TMR set of Nor-Flash radiation-tolerant memories (128Mbit).

Indeed, both Xilinx and MicroSemi FPGAs have commercial and compatible versions which speed up the development and prototyping process. Also, both the clock and power distribution, including DC/DC, Low-Drop-Out (LDO), and EMI filters, are designed with flight-grade quality components. Finally, the full modem design was mounted on a 14-layer Printed Circuit Board (PCB) where 6 of them are allocated to routing and 8 to power and grounding planes. To wrap up, Figure 7 illustrates the implementation architecture described in this first part of the section.

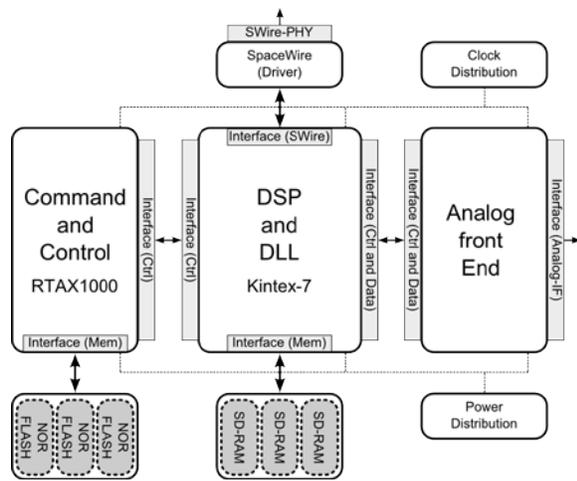


Figure 7: Modem Implementation Architecture

In the following sub-sections we provide a higher detail of the implementation of the Data-Link-Layer (DLL), Digital-Signal-Processing (DSP) and Analog-Front-End (AFE).

Data Link Layer

On the sender side, the DLL layer is responsible for accepting user data from the on-board computer (via SW) to later submit it to protocol-specific algorithms to finally send the frame to the PHY layer. On the receiver side, the DLL receives the serial output from the receiver to later process the protocol data units before being delivered to the SW port. On the north-bound interface towards the OBC, the DLL accepts user data, and directives (configurations), and transmits received

user data, notifications (status reports), and flow-control information.

In particular, the transmitting chain of the DLL is implemented in software (LEON3-FT processor) by means of a *state-driven*, while the receiver by a *modeless* (i.e. non-stateful tasks) approach. Furthermore, the RTEMS operating system provides a general and transparent service platform that allows the processor to access a shared bus where both the SpaceWire port and DSP module are connected.

Finally, despite the codification and synchronization are part of the DLL layer in the Proximity-1 specification (Figure 5), for performance reasons, we implement most of its functionality in hardware within the Digital Signal Processing (DSP) module. The DSP module will be described in the following sections.

Digital Signal Processing

The DSP module implements part of the Physical layer (PHY) of the Proximity-1 specification, but also it encompasses hardware aspects of the coding and sync (C&S) sub-layer of the DLL as shown in the Figure 8. In general, the latter allows for higher performances since executing coding and synchronization in software can impose severe requirements to the modem processor. Therefore, the DSP communicates via an I&Q signal with the AFE (described below) in the south-bound, and directly with the DLL processes running on the LEON processor on the north-bound interface.

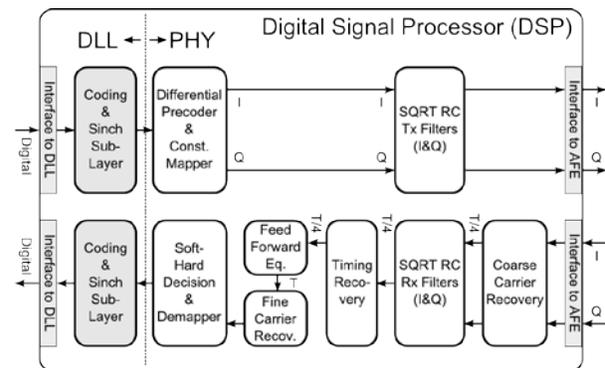


Figure 8: DSP Implementation Architecture

Regarding the C&S sub-layer, the Proximity-1 specification provides three options: a) No coding, b) Convolutional Coding (CC) (rate 1/2) concatenated with Reed-Solomon (RS), and c) Low-Density-Parity-Check (LDPC) coding (rate 1/2) and 1024 word-size. We found that LDPC option outperforms CC+RS by about 0.5dB and does not require code interleaving, bending the coding selection towards LDPC. Therefore, we implemented an LDPC 1/2 scheme which main

drawback is the memory requirement which in turn is not an issue in our particular hardware design (Kintex-7). Also, we included a higher rate LDPC of 5/4 for higher flexibility and performance. Indeed, both implementations include a pseudo-randomizer in order to guarantee enough bit-transitions in the transmitter module, and a scaled min-sum decoder in the receiver. Furthermore, it is worth adding that this module accounts for a noise power estimator which can be used as a means of determining signal quality in order to dynamically select and negotiate parameters in the future. As a result, our implementation of the Proximity-1 Modem accounts for three feasible C&S configurations:

1. No coding: Higher bit-rate (no overhead), lower Bit Error Rate (BER) tolerance.
2. LDPC (5/4): Medium bit-rate (20% overhead), medium BER tolerance.
3. LDPC (1/2): Low bit-rate (50% overhead), higher BER tolerance.

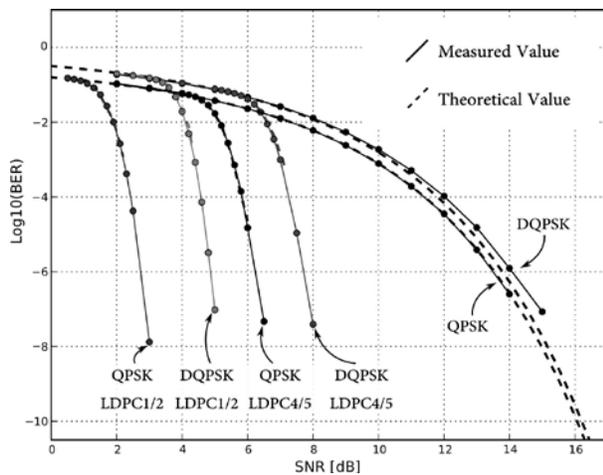


Figure 9: Performance of LDPC 1/2 and 4/5 when used in QPSK and DQPSK

It is worth recalling that the resulting coding configurations can be defined in-flight by means of remote or local directives. Also, the overall DSP baud-rate (hence, the final bit-rate) can be configured in-orbit as per Table 1 parameters. In particular a low-data rate mode based on Binary-Phase-Shift-Keying (BPSK) allows for a more robust communication, while a high-data rate exploits Quadrature-Phase-Shift-Keying (QPSK) throughput. As a result, the configurability of these properties allows for drastically boosting the modem flexibility to cope with a highly varying set of channel types (distance, depointing, interference, BER, etc.). Figure 9 illustrates the measured and theoretical BER for each of the QPSK configurations.

Table 1: DSP Baud and Bit Rates

Baud Rate	Modulation	Bit Rate
128 KSym/Sec	BPSK/DBPSK	128 Kbps
1024 KSym/Sec	QPSK/DQPSK	2048 Kbps

On the transmitter side of the PHY part of the DSP, we highlight the BPSK/QPSK modulator and differential encoder, and the transmitter filter. In particular, we designed the DSP with a differential encoding (DBPSK and DQPSK: both optional for BPSK and QPSK in Proximity-1 specification) in order to mitigate a well-known phenomenon known as the *Cycle Slip* (CS). The CS effect is generated by phase noise in the channel and due to carrier recovery limitations which can derive in a change of phase within a block of data. This, in turn, provokes burst errors which are particularly difficult to correct with Forward Error Correcting Codes (FECs). However, the synchronization markers of LDPC codes allow overriding the stated phase ambiguity allowing the usage of plain BPSK and QPSK with better performances (Figure 9). As a result the modem accounts for BPSK, DBPSK, QPSK, and DQPSK modulations (all with no coding, LCDP 4/5, or LDPC 1/2) with the constellation and differential mapper illustrated in the Figure 10.

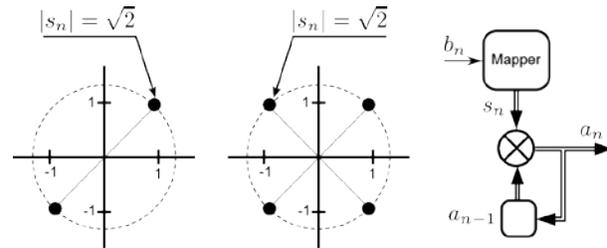


Figure 10: BPSK and QPSK Constellation and Differential Mapper

Finally, the transmitted signal power is concentrated within the required frequency to minimize interference with lateral bands by means of a square-root-raised-cosine filter (SQRT)¹⁴. The filters are designed with a poly-phasic approach (relaxes computational requirements) with a total of 64 coefficients, and a roll-off of 0.5 providing more than 35dB of lateral side-lobes suppression.

The receiving side of the DSP is composed of a digital automatic gain control (AGC), reception filters, coarse carrier recovery (CCR), timing recovery (TR), a fractional speed equalizer (FSE), a fine carrier recovery (FCR), and a differential decoder. Also, the DSP controls the channel selection which is made by a quadrature demodulator on the AFE which in turns delivers the complex signal to the reception chain of the DSP.

Before filtering, a Coarse Carrier Recovery (CCR) implemented by means of a Differential Power Measurement (DPM)¹⁵ is needed since decision-based approaches cannot deliver a frequency offset beyond 128KHz for the high data-rate mode which is not sufficient for space communications. Then, the first filter on the receiving side of the DSP module is a decimation filter implemented by means of a Cascade Integrator-Comb (CIC) Finite Impulse Response (FIR) filter. Finally, and similarly with the transmitting chain, a raised-cosine filter with an oversampling of 4 (4096 KHz for the high data-rate mode) minimizes residual interference effects.

Next, Timing Recovery (TR) is set to extract the received signal frequency tone used in the remote transmitter. Indeed, clock differences and Doppler Effect are mitigated by the TR module. Typically, a *spectral line*¹⁴ method is used where the received signal is modified by a non-linear operation such as a square transformation. In this module, a Wave Differential Method (WDM)¹⁶ and a Rotational Frequency Detector (RFD)¹⁶ allows to recover synchronism in presence of phase shift. In particular, the implementation of TR module relies on Mixed-Mode Clock Manager (MMCM) included in the Kintex-7 FPGA.

Despite frequency-dependent dispersion in satellite links can be disregarded, we chose to implement a reduced Fractional Spaced Equalizer (FSE)¹⁴ filter with only 5 coefficients to cope with possible dispersions provoked by the transmitter filters and temporal gain variations that cannot be tackled by the AGC. However, the FSE equalizer cannot generally compensate fine carrier offsets requiring of a Fine Carrier Recovery (FCR) module implemented by means of a Decision Directed (DD) mechanism¹⁴. Finally, a de-mapper demodulates the BPSK or QPSK symbol before delivering the bit-stream to the C&S sub-layer.

Analog Front End

In general, the Analog-Front-End (AFE) is responsible for taking the I&Q signaling from the DSP to provide the intermediate frequency (IF) to the Radio-Frequency Front-End component, which is out of scope of this documentation. The same remains true for the inverse signal-path on the receiver side. In particular, the IF interface delivers or accept a channelized signal similar to the one shown in S-Band in Figure 3, but with a case frequency of 100MHz. Specifically, Table 2 list the expected channel frequencies. Next, figure 11 depicts the AFE architecture.

As shown on the Figure 11, on the transmitting side, the AFE accounts for a DAC, analog filters, and an analog modulator. Since the maximum symbol frequency is

1024 KHz, the digital to analog converters (DAC) are chosen with a sampling frequency of 8192 KHz. This implies that the transmitting oversampling has a factor of 8. On the other hand, the 4th-order Bessel-type low-pass filters delivers an output signal of 2.5MHz bandwidth. The latter in combination with the raised-cosine filters from the DSP, provides maximum co-channel interference.

Table 2: IF Channel Frequencies

Channel Number In Band A	Channel Number In Band B	Frequency
A1	B10	105.3 MHz
A2	B9	112.8 MHz
A3	B8	120.3 MHz
A4	B7	127.8 MHz
A5	B6	135.3 MHz
A6	B5	142.8 MHz
A7	B4	150.3 MHz
A8	B3	157.8 MHz
A9	B2	165.3 MHz
A10	B1	172.8 MHz

The receiving side is composed of analog de-modulator, and an analog to digital converter (ADC). A quadrature demodulator is controlled from the DSP chain in order to perform the channel selection. The complete demodulation is performed on the reception chain of the DSP so as to exploit several well-known algorithms that improves the recovery performance.

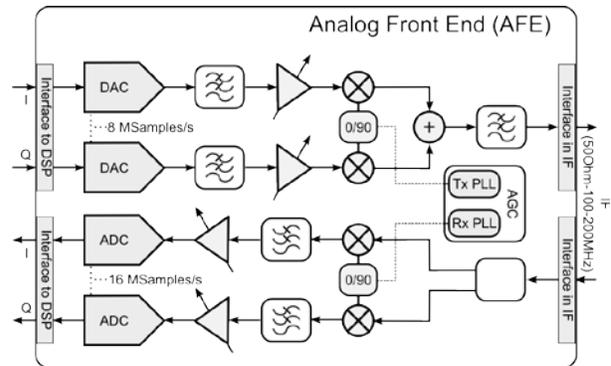


Figure 11: Analog Front End Implementation Architecture

In particular, the filtering in RF in the receiving chain is also performed by a 4th-order Bessel filter with a 2.5MHz bandwidth. Furthermore, one of the main features of the receiving part of the AFE is its ability to demodulate signals with significant frequency offset which can be product either of electronic component

imperfections (oscillators) and Doppler shift. By design the modem is able to cope with offsets of more than 750 KHz, with even better performance (1MHz) for the low-data-rate mode (128Kbps). Figure 12 illustrates the resulting spectrum of the receiving signal delivered to the DSP input with a co-channel rejection of more than -20dB, and a 1MHz frequency offset signals. Indeed, the DSP algorithm increases this channel rejection beyond -35dB as explained in the previous section.

Once filtered, the signal is fed to two ADCs (one for each I and Q complex component) for which the sampling rate is controlled from the synchronism recovery mechanism from the DSP. The sampling rate is 16384 KHz which allows implementing efficient algorithms in the DSP to drastically reduce the co-channel impact for both high and low data-rate modes. Finally, the input signal gain is dynamically adjusted by means of an Automatic Gain Control (AGC), converted to I&Q signaling, and delivered to the DSP module.

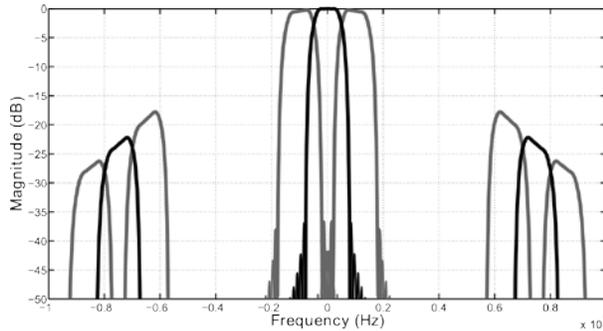


Figure 12: AFE Rx Output (to DSP in I&Q)

RESULTS AND MEASUREMENTS

In this section we provide the results and measurements of the STI-PRX-01 Proximity-1 modem in a stressed scenario as those expected in ESL and ISL links.

Scenario Description

The scenarios we are evaluating to demonstrate the performances of the Proximity-1 modem are described below. On the one hand, we configure a DQPSK high data-rate mode without code protection, and on the other hand, a DQPSK high data-rate mode with a LDPC 4/5 coding scheme. This will allow us to measure the tradeoff between error recovery by FEC coding and higher layer's algorithms such as Proximity-1 SEQ frames.

For both scenarios, we configure an intermediate noise generator to deliver a signal with a 12dB of SNR, and also, frequency offset of 500 KHz for channels A9 and B2. Furthermore, a clock shift is enforced to 200ppm deriving in an additional frequency offset of 32 MHz. This scenario configuration allows verifying and testing

the modem implementation in a stressed RF environment. Regarding traffic configuration, we are transferring a 77.8 KB image once with EXP QoS and another with SEQ with secure delivery on the DLL protocol layer. The maximum frame size is set to 2000 Bytes and the SEQ re-transmission window to 5 frames.

Results

In this section we summarize and analyze the analog, digital, and protocol measurements obtained from the two modem prototypes in the scenario described above. In general, all parameters plotted can be obtained by means of the modem debugging interface. This interface was specifically designed to test, troubleshoot, and provide enough telemetry information to perform the modem monitor while on flight.

On the physical side, Figure 13 depicts an eye diagram for the QPSK signal in a 12 dB SNR environment. A total of 10 100ns step on the Y axis suggest a symbol time of 1us which corresponds to a 1Million Bauds per seconds, which on QPSK (2 bits per baud) implies a 2Mbps data-rate.

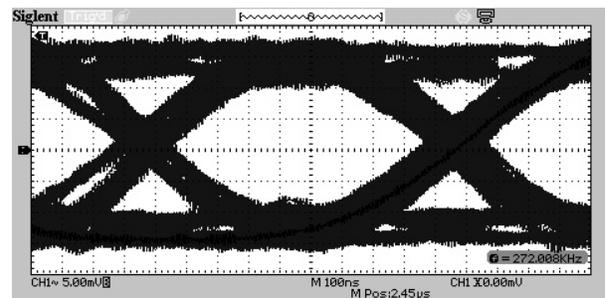


Figure 13: Eye diagram of the QPSK Signal

In Figure 14 we show the spectral frequency of TX at IF in the 105,3MHz channel. As can be appreciated the spurious components are 60 dB below carrier at the adjacent channel.

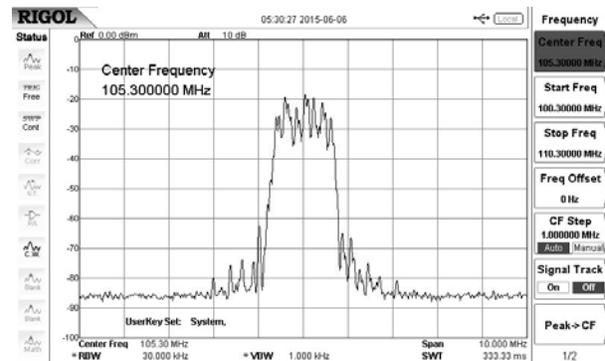


Figure 14: Spectral frequency of the 105.3MHz Channel in the transmitter

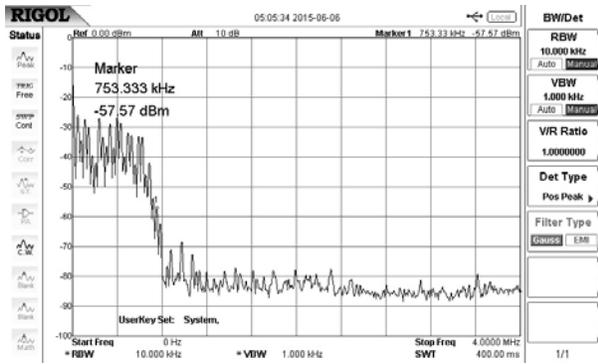


Figure 15: Detailed spectral frequency of the signal in IF

A more detailed spectrum can be seen in Figure 15, where a -60dB marker is observed at a 753.3 KHz (1.5 MHz total) bandwidth. Next, in figure 16, we illustrate the combined spectrum of two IF Tx channels where two markers measure the channel frequency distance of 7.5 MHz as previously described. In this figure the two channels are clearly separated and no interference is measured, validating the analog baseband filtering.

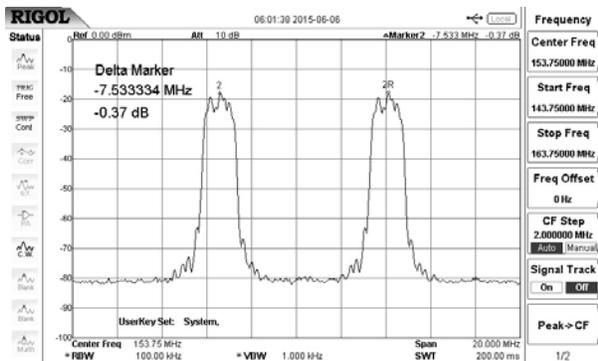


Figure 16: Spectral frequency of two adjacent channels without interference

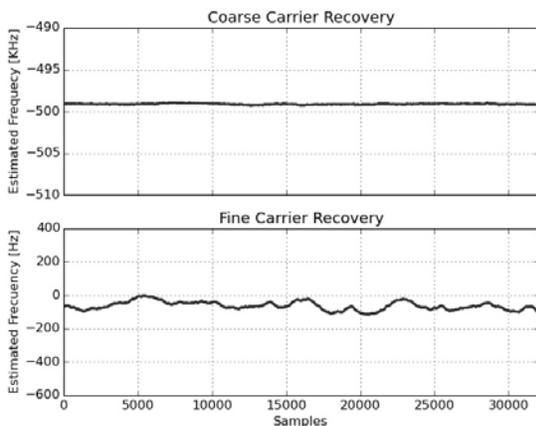


Figure 17: Coarse and Fine carrier recovery compensating large frequency offsets

Next, on the digital signal processing module, we use the debugging interface to access a set of DSP variables. Figure 17 illustrates both the coarse and fine carrier recovery performance. In particular, the coarse carrier recovery is able to cope with the major offset of 500 KHz approximately, while the fine carrier recovery adjusts the signal for less drastic frequency variations.

Another important part of the DPS is the timing recovery and automatic gain control (AGC). The Figure 18 illustrates their behavior in the proposed scenario. In particular, the 200ppm offset is properly detected by the timing recovery module.

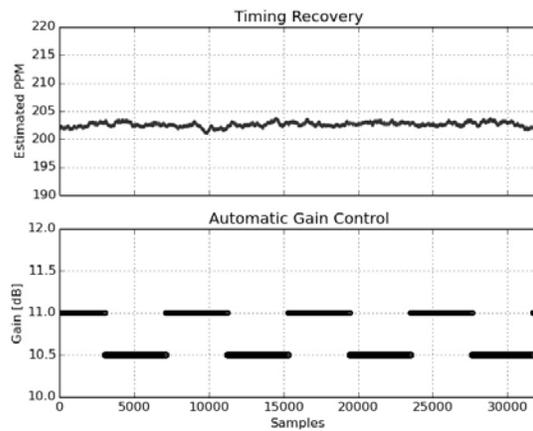


Figure 18: Timing recovery and automatic gain control

On the equalizer module, Figure 19 illustrates the real and imaginary part of the equalizer for the signal.

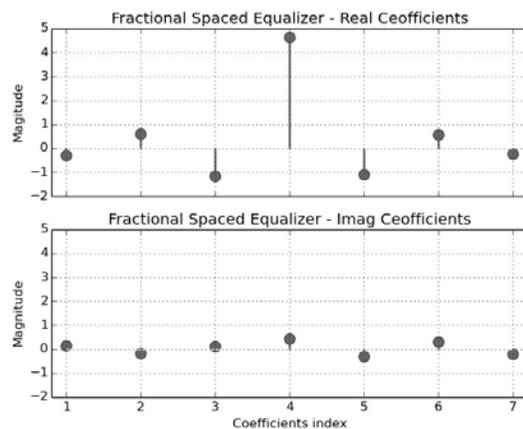


Figure 19: Equalizer coefficient values for the proposed scenario

Furthermore, Figure 20 illustrates the constellation of the demodulated symbols not only for the DQPSK signal but also for a DBPSK for illustration purposes. Since the output of the fractional speed equalizer (FSE)

do not account for frequency offset corrections, the constellation cannot be recovered properly until the fine carrier recovery (FCR) compensates this phenomena. This type of information is also obtained from the debugging interface.

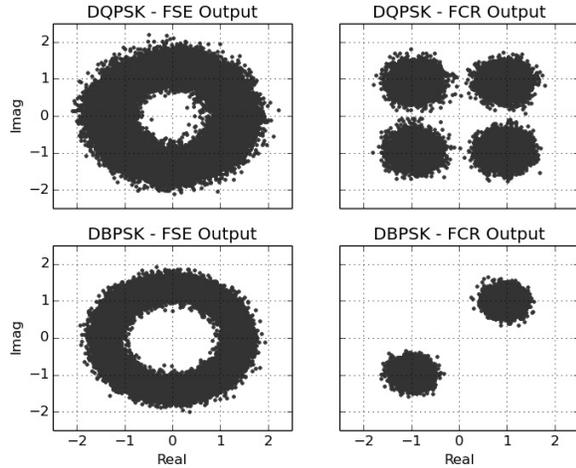


Figure 20: DBPSK and DQPSK constellation maps

Finally, on the Data Link Layer (DLL), we analyze the frames sent, delivered, corrupted, and re-transmitted in the proposed scenario. The reader should refer to figure 4 to properly interpret the protocol stage on which the listed statistics are taken from.

Table 3 provides a list of the frame count for each of the DLL sub-layer for the first execution of the scenario where no LDPC code is applied to the signal. In particular, the table includes statistics from the data service (DS) and frame sub-layer (FSL) for each of the modem in the test bed (modem A and B).

Table 3: DLL Measurements (No FEC Code)

Parameter	Modem A	Modem B
Data Serv. EXP Tx	132	132
Data Serv. SEQ Tx	132	132
Data Serv. SEQ Re-Tx	1259	1211
Data Serv. Total Tx	1523	1475
Data Serv. EXP Rx-ok	42	49
Data Serv. SEQ Rx-ok	132	132
Data Serv. SEQ Rx-Gap	283	329
Data Serv. SEQ Rx-Rep	13	10
Data Serv. Total Rx	470	520
Frame. PLCW Tx	256	282
Frame. PLCW Rx	111	98

In the proposed scenario, both modems send two instances of a single file using a different quality of service interface for each of them: expedited services (EXP) and sequence controlled (SEQ). As a result, the image is accommodated in a total of 132 Proximity frames for EXP and SEQ. However, the 12 dB of SNR of the channel induces enough noise to corrupt the frames. Indeed, despite 132 EXP frames were transmitted from modem A and B, only 41 and 49 were successfully received respectively. On the other hand, the SEQ service provides a retransmission feature that avoids the loss of corrupted data when no confirmation is received from the other end of the link. Therefore, modem A and B had to retransmit a total of 1259 and 1211 frames respectively until the full payload of 132 frames is received correctly at the receiver. This re-transmissions based on go-back-n derives in collateral effects such as frames received in gap or repeated as shown in the table. Also, several confirmation frames (PLCW) are also lost in the noisy channel forcing both modems A and B to send more than 132 confirmations (256 and 258 respectively). It is worth clarifying that a single PLCW can confirm more than a single frame, hence, there no specific need of receiving a total of 132 PLCW in the receiver.

Table 4: DLL Measurements (LDPC 4/5)

Parameter	Modem A	Modem B
Data Serv. EXP Tx	132	132
Data Serv. SEQ Tx	132	132
Data Serv. SEQ Re-Tx	1	3
Data Serv. Total Tx	265	256
Data Serv. EXP Rx-ok	132	132
Data Serv. SEQ Rx-ok	132	132
Data Serv. SEQ Rx-Gap	0	0
Data Serv. SEQ Rx-Rep	3	1
Data Serv. Total Rx	267	265
Frame. PLCW Tx	132	132
Frame. PLCW Rx	128	131

On the other hand, when enabling LDPC 4/5 coding, the transmitted frames drop drastically as seen in Table 4. This is product of the effect of the LDPC code which provides error correction on the C&S sub-layer. The latter improves the delivery of the frames avoiding the protocol to have to react and re-transmit the corrupted data. In this context, only 3 frames were lost in the B to A channel, while 1 was corrupted in the A to B path, requiring a minimal intervention from the upper layer go-back-n protocol.

As a results and measurement closing remark, the transmitted data in the LDPC mode required 250 frames (in average) in contrast with the 1500 (in average) in the execution without coding gain, providing a significant saving (about 80%) in channel usage time and energy at the expense of a minimal overhead (25%). The latter is a clear evidence of the modem flexibility, adaptability, and autonomy which were the main requirement from which this communications sub-system was designed upon. Indeed, a better channel condition without frame loss would have rendered the usage of the non-coding communication scheme more convenient than the LDPC 4/5.

CONCLUSION

In this article we introduced the need of an autonomous and flexible communication sub-system able to cope with the special scenario of mixed earth-to-satellite (ESL) and inter-satellite links (ISL). Once we verified the physical and algorithmic suitability of the CCSDS Proximity-1 protocol, STI-PRX-01, a specification-compliant modem architecture was proposed. However, a few considerations were discussed such as an S-Band channel assignment.

Through the paper we provided details on the technical decisions taken in order to effectively implement a Proximity-1 modem, ranging from Analog Front End (AFE), Digital Signal Processing (DSP) and Data-Link Layer (DLL) module. As a result, we verified the implementability of this architecture in a functional prototype on which we executed a basic, yet interesting, demonstration of the modem capabilities. Indeed, we have shown that the modem is able to efficiently adapt to a varying range of channel conditions by simply configuring its parameter while on-flight.

We left as further work the development of an interface to a mass storage system where the satellite platform can directly store the data to be downloaded. Thereafter, the modem would autonomously calculate further contacts or routes towards a destination in a networked environment of small-satellites. Therefore, such a version of the modem would become a simple plug-and-play data management system for satellite constellations.

In general, the authors are confident that the flexibility features included in STI-PRX-01 constitutes not only an evolution in modern inter-satellite communications systems, but also a paradigm shift towards higher-layers protocols in order to provide better performances than the traditional Telemetry and Telecommand architecture.

References

1. Z. Yoon, W. Frese, K. Briess, and A. Bukmaier, Mission Design of a S-Band Network of cooperative nanosatellites, International Workshop on Satellite Constellations and Formation Flying, 2013.
2. O. Brown and P. Eremenko. The value proposition for fractionated space architectures. In AIAA-2006-7506, AIAA Space 2006, San Jose, CA, 2006.
3. Fractionated spacecraft: the new sprout in distributed space systems, J Guo, DC Maessen, EKA Gill - 60th International Astronautical Congress: IAC 2009.
4. T.P. Garrison, M. Ince, J. Pizzicaroli, and P.A. Swan. Systems engineering trades for the iridium constellation. Journal of Spacecraft and Rockets, 34(5):675-680, Oct 1997.
5. Kerri L. Kusza and Michael A. Paluszek, Intersatellite Links: Lower Layer Protocols for Autonomous Constellations.
6. Paul Muri, Janise McNair, A Survey of Communication Sub-systems for Intersatellite Linked Systems and CubeSat Missions, Journal of Communications, Vol. 7, No. 4, Apr 2012.
7. Consultative Committee for Space Data Systems (CCSDS). Proximity-1 Space Link Protocol Rationale, Architecture, and Scenarios, Informational Report, CCSDS 210.0-G-2, Dec 2013.
8. P. Walz, CCSDS Proximity-1 Links Mars Mission Communications Success, Disney Film and Interplanetary Internet, CCSDS Press Release, Mar 2006.
9. T. Vladimirova and K. Sidibeh, WLAN for Earth Observation Satellite Formations in LEO, IEEE Bio-inspired Learning Intelligent Systems for Security (BLISS), 2008.
10. C. Edwards, Jr., T. Jedrey, E. Schwartzbaum, and A. Devereaux, The Electra Proximity Link Payload for Mars Relay Telecommunications and Navigation, 54th International Astronautical Congress of the International Astronautical Federation, September 2003.
11. 7 Series FPGAs Configuration. User Guide. UG471
12. RTAX-S/SL and RTAX-DSP Radiation-Tolerant FPGAs

13. 13. J. Barry, E. Lee, and D. Messerschmitt. Digital Communication. KAP, 2004.
14. 14. T. Albery and V. Hespelt. A New Pattern Jitter Free Frequency Error Detector. IEEE Transactions on Communications, Vol. Com-37, No. 2, Feb. 1989.
15. 15. O. E. Agazzi, C. J. Tzeng, D. G. Messerschmitt, and D. A. Hodges. "Timing Recovery in Digital Subscriber Loops". IEEE Transactions on Communications, Vol. Com-33, No. 6, Jun 1985.
16. 16. Space Telecommunications Radio System (STRS) Hardware Architecture Standard (Release 1.0 Hardware Section) NASA/TP—2008214471