AMSAT Linear Transponder Module (LTM) Interface Control Document

January 22, 2022
Revision: V1.2

This document may contain proprietary information or material covered by Non-Disclosure Agreements (NDA) if so marked/highlighted. This document is intended to be shared by members of the AMSAT Linear Transponder Module (LTM) design, development, assembly and test team and members of a "Partner University" CubeSat development team. Discussions about, as well as further dissemination of the information contained herein, should not occur without knowledge that the participating recipients are approved for access to this information..

ICD Revision History

Revision	Date	Author(s)	Change Log
0.1	February 28, 2021	E. Skoog (K1TVV) B. Fisher (WB1FJ)	ICD initial draft material
0.2	May 16, 2021	B. Fisher (WB1FJ)	More CAN and downlink details
0.3	June 14, 2021	B. Fisher (WB1FJ)	Combine .2 with UMO's appendix revisions; update telemetry to maximum length
0.4	June 15, 2021	B. Fisher (WB1FJ)	Fix frame size and max CAN sizes
0.5	July 27, 2021	B. Fisher (WB1FJ)	Fix field order in CAN ID; clarify MAX31725 addresses
0.6	August 19, 2021	B. Fisher (WB1FJ)	Add CAN health message types
0.7	September 9, 2021	B. Fisher (WB1FJ)	Add CAN uplink messages
0.8	December 8, 2021	B. Fisher (WB1FJ)	Fix typos, remove host appendix
0.9	January 19, 2022	E.Skoog (K1TVV)	Change ITAR to proprietary
1.0	January 20, 2022	B. Fisher (WB1FJ)	Change telem bits from 1 per data bytes to 8 per data byte. Clear up some text.
1.1	January 21, 2022	B. Fisher (WB1FJ)	Change 32-bit binary downlink to 3 CAN messages

1.2	January 22, 2022	B. Fisher (WB1FJ)	Change 32-bit binary downlink to up to 4 CAN messages

INTRODUCTION

Interface management and control is a System Engineering activity that begins in parallel with the development of system architectures and continues for the life of a system. Likewise, the evolutionary process to define interfaces begins with concept definition and continues throughout the development phase. Interfaces, both internal and external to the system, are documented and formally managed. This activity is intended to ensure compatibility among subsystems being designed and fabricated, and to ensure future interoperability between systems.

System interfaces are both physical and functional. Initially, a system is depicted by a top-level Concept Diagram that identifies its external interfacing elements. Source documents, such as Capability Development Documents, Capstone Requirements Documents, system specifications, and interface documents provide interface requirements to ensure interoperability between systems and that the required capabilities are realized. An operational concept may also provide descriptions, interactions, and requirements between the system and its external elements. An interface definition process aids architecture and requirements evolution in conjunction with the overall system definition process, i.e., interfaces will mature as operational and system requirements mature.

1.1. Overview

An interface, from the perspective of system development, can be identified as any point where a system and something, or someone, meet. This interface point may include other systems, internal hardware, external peripherals, networks, system users, etc.

Interfaces are documented using Interface Control Documents (ICD) that describe inter/intra system workings including any communication rules. ICDs help ensure compatibility between system segments and components and are considered critical elements of effective system design and development.

Interface Control Document (ICD) ≡ describes the interworking of two elements of a system that share a common interface. An ICD may also describe the interaction between a user and a system, a software component and a hardware device, or two software or hardware components. An ICD is typically used where complex interfaces exist between components that are being developed by different teams. It is jointly prepared by the interfacing groups.

For the development team, ICDs define applicable system inputs, outputs, and interfaces to other systems, subsystems, and/or users. An ICD should only describe the interface itself and not the characteristics of the systems connected. An ICD does not normally include details regarding the meaning or intended use of the data being interchanged across the interface. Any design features, functions, or logic supporting a system interface should be detailed in separate design and/or specification documents. Defining an ICD in this way allows other teams to develop connecting systems without being concerned about design details or how data is treated by the other system. This allows development teams to work without knowledge of the business logic or technical aspects behind the interface system. This facilitates modularity and leads to easier maintenance and extension of the interfacing systems.

1.2. Purpose

The purpose of interface management and control is to define interface requirements so as to ensure compatibility between inter-connected pieces of equipment and to provide an authoritative means of controlling the design of the interfaces.

The purpose of an ICD is to clearly communicate a system's external inputs and outputs for all potential interface actions whether or not they are transparent to the system's users. An ICD may describe the system interface to its lowest physical or functional elements (e.g., units, modules, software routines/tasks, etc.) and/or parameter values (e.g., circuits, voltages, impedances).

An ICD describes the relationship between two components of a system in terms of data items and messages passed, protocols used, and the timing and sequencing of interactive events. Specifically, ICDs are generated to:

- Ensure configuration control of the system's interface design to prevent any changes to characteristics that would affect its compatibility with other systems,
- Define and illustrate the physical and functional characteristics of a system in sufficient detail to ensure compatibility of the interface, and verification of that compatibility can be determined from the information in the ICD,
- Facilitate the identification of missing interface data and control the submission of that data, and
- Communicate coordinated design decisions and design changes to all involved participants.

ICD's by nature are requirements documents. They define design requirements and allow integration. They record the agreed-to design solution to interface requirements and provide a control mechanism to ensure that the agreed-to designs are not changed by one participant without the knowledge of and agreement with the other participant(s).

To be effective, ICD's should track a schedule compatible with the design maturation of the system, e.g., initial ICD's should be at the 80% level of detail at Preliminary Design Review (PDR), should mature as the design matures, and should reach the 99% mark near the Critical Design Review (CDR) period.

1.3. Interface Requirements

It is important that requirement statements defining the interface between two systems be stated in clear, unambiguous language. Starting with the type and purpose of the interface, all interface design requirements and constraints must be identified. Explicit definitions of the content and format of every message/file that can pass between the systems and the conditions under which each is to be sent should be detailed. For non-IT interfaces, assembly characteristics, stresses, connections, accessibility requirements, mechanical details/descriptions, etc. should be provided.

Tables, diagrams, and figures should be used to aid in understanding interface characteristics. Reference material such as data dictionaries, protocols, and standards should be identified in an ICD References section.

Interface requirements may be summarized by describing data protocols, communication methods and processing priorities used by the interface, e.g.,

- Data protocols may include messages and ASCII files exchanged,
- Communication methods may include networks or storage media, and

- If specific information processing must be prioritized, formatted and communicated in accordance with schedules (real time – batch – asynchronous – etc.), such information should be clearly highlighted.

1.4. Cooperative Agreement

This Interface Control Document specifies the interface between AMSAT's Linear Transponder Module (LTM) [System #1] and a Partner University's CubeSat Platform [System #2]. The information contained herein is to be used during development and testing by all parties requiring such information, including hardware/software developers, system engineers, and testers responsible for verifying the required performance of the interface. Upon approval by both development organizations, this ICD shall be incorporated into the requirements baseline of each system.

This ICD will be developed and updated regularly over the period of the project by both Partner Universities and AMSAT responsible team members. There will be no unilateral/unpublished changes made to the interface details/requirements included in this ICD. Rather, any mission or design changes that would affect the interface information contained herein will be immediately brought to the attention of the other party so that impact analyses and possible redirection planning may be conducted.

To ensure effective/efficient development of the capabilities detailed in this ICD, Partner University and AMSAT engineering team members will coordinate/collaborate technical/programmatic issues via scheduled and unscheduled communication utilizing documentation generation/exchange/review/revision activities, email, telephone, video-teleconference, network-enabled test events, etc.

Mutual support will be provided to assist in trouble-shooting during initial module (engineering model – EM) testing, Flight Model (FM) integration testing, Preliminary and Critical Design Reviews (PDR/CDR), Mission Readiness Reviews (MRR), and other launch preparation activities as required.

2. SYSTEM INTERFACE DESCRIPTIONS

This ICD specifies the interfaces between AMSAT's LTM and a Partner University CubeSat host platform. It is divided into three main sections: (1) System Description; (2) Structure and Electrical Lines; and (3) Software and Data, to include protocols and uses for the electrical connections.

The terms "will", "shall", or "must" herein mean that the specified action (or lack of in the case of "not") is a requirement placed on either the LTM or the Partner University's CubeSat design. The term "may" means that the specified action is allowed but optional.

The term "opaque to LTM" is sometimes used in these descriptions. LTM does not act on opaque data, but will downlink, save, or pass it to the host platform depending on the context. The term "ignored by LTM" means that LTM does not act on the data, nor does it save or transmit it.

This ICD does, in several places, specify choices to be made by the Partner University host platform. Interface details specific to a Partner University host platform will be provided in an appendix to this ICD. Those details will include which choices the Partner University host platform makes.

2.1. AMSAT Linear Transponder Module (LTM) [System #1]

2.1.1. System Description

The LTM is a 3-board module provided by AMSAT comprising: an LIHU (Legacy Internal Housekeeping Unit, essentially the on-board computer), an ICR (Improved Command Receiver), and an RxTx (the linear transponder receiver - transmitter). It is intended to be used as the radio module for a non-AMSAT developed satellite, herein called “the host platform.” All software for the LIHU is provided by AMSAT.

The LIHU is NOT intended to be used as a flight computer for the host platform. It communicates with the host platform’s flight computer via a CAN bus.

AMSAT’s LTM will provide VHF uplink command and control capability as well as UHF downlink TLM data in support of a Partner University’s on-orbit mission. Consequently, the LTM provides a separate 1200 bps BPSK modulated channel to download daily housekeeping telemetry, verify CubeSat operating parameters, and download operational status, education, and scientific data from the University platform’s subsystems. The design of the transponder includes state-of-the-art integrated circuit components to provide exceptional communications performance, while minimizing size, weight and power. For Amateur Radio Satellite Service authorized use, the LTM will provide V/u transponding capabilities. Amateur radio satellite uplink communication transmissions will require generation of VHF (2 meter) SSB/CW amateur radio signals and downlink signal receipt will require reception of UHF (70 cm) SSB/CW signals.

2.1.1.1. Block Diagram

At the conceptual level, AMSAT’s Linear Transponder Module can be envisioned as a set of major functional blocks as illustrated in Figure 1 below.

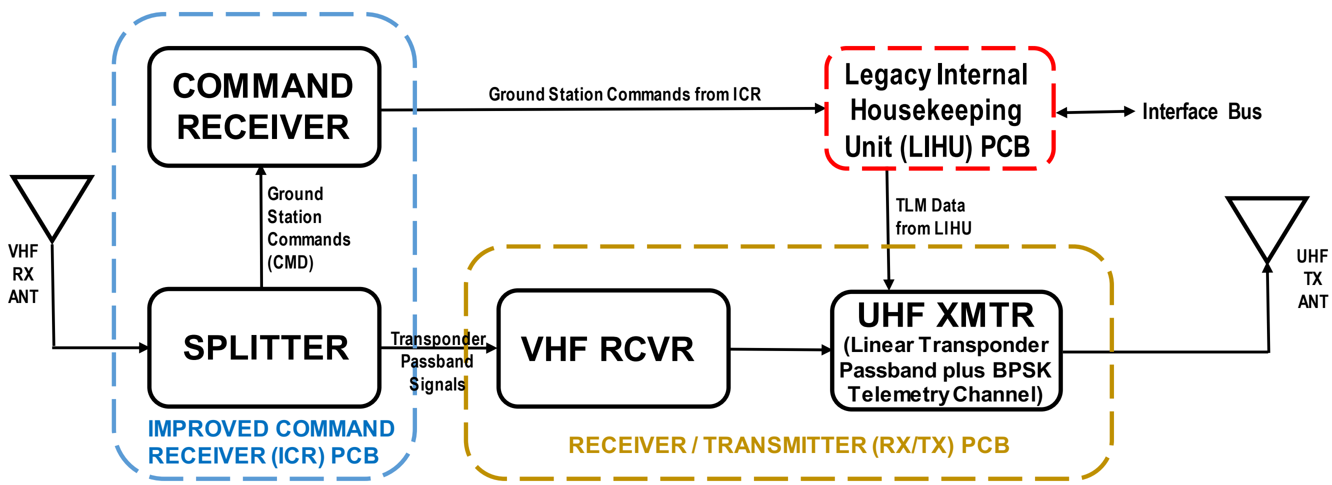


Figure 1: AMSAT Linear Transponder Module (LTM) Top Level Block Diagram

Figure 2 below illustrates the major interface connections between AMSAT’s Linear Transponder Module (LTM) and a University’s CubeSat host platform.

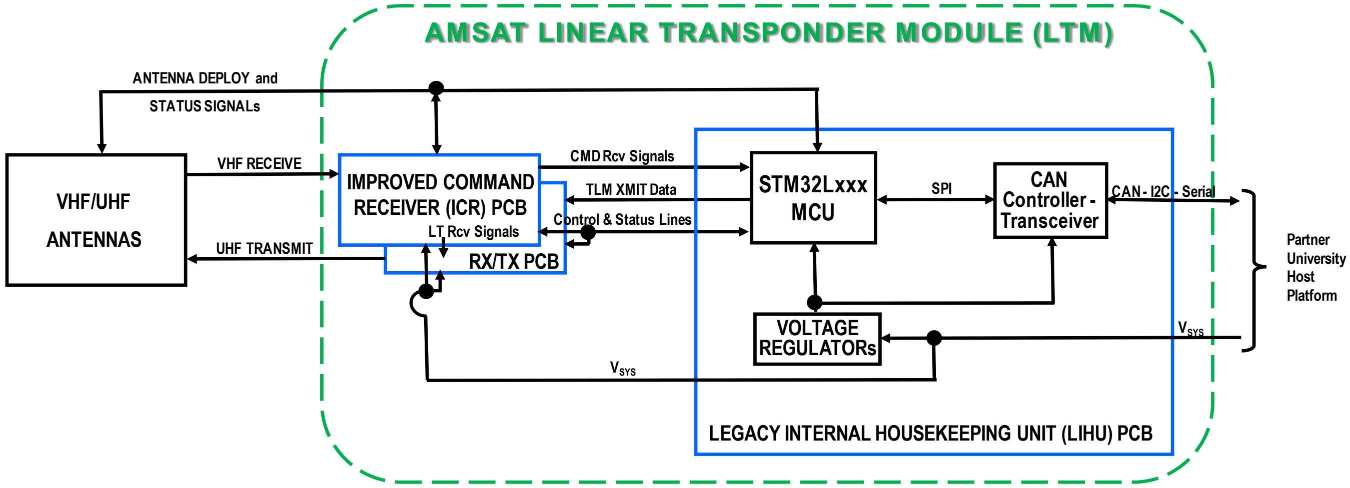


Figure 2: Linear Transponder Module Major Interface Connections

2.1.1.2 Frequencies

The LTM shall exhibit a transponder user passband (3dB) bandwidth of 30 kHz. All uplink-downlink VHF and UHF operational frequencies to be used by AMSAT's LTM shall be Partner University coordinated and authorized through IARU and FCC application, viz.,

UPLINK: 145.xxx MHz to 145.xxx MHz

DOWNLINK: 43X.xxx MHz to 43X.xxx MHz (inverting)

TLM Downlink Channel (dedicated): 435.xxx MHz (1200 bps BPSK)

COMMAND & CONTROL UPLINK CHANNEL: *[redacted]* MHz (exact frequency is sensitive)

2.1.1.3 Performance "Desirements"

Consideration should be given to the extent to which the proposed LTM design is capable of satisfying users' *unconstrained wishes*. This means articulating performance via a set of "desirements." These are merely (performance) requirements that would be ideal IFF they can be achieved by an affordable, viable system design.

When studying an initial desirements list, it is helpful to consider 'corner (use) cases' to ensure the system under development will be beneficial to a wide spectrum of operators. One use-case that will assist in determining the "worst case" baseline design involves a "disadvantaged" ground station; disadvantaged in the sense that some operators' transmit/receive capabilities are less than a normally equipped amateur radio satellite ground station. Link budget calculations confirm that satisfying the 'desire' to accommodate such a low end user station leads to a set of "Design To" requirements. Consequently, the required set of performance parameters for the four main Linear Transponder Module (LTM) signal chains are shown in Tables 1 to 4 below, viz.:

(1) UHF Passband Transmit (TX); (2) VHF Passband Receive (RX); (3) VHF Uplink Command Channel Receive; and (4) the UHF Downlink Telemetry Transmit Channel.

PERFORMANCE PARAMETERS	REQUIRED VALUES	COMMENTS
UHF Downlink Transmit (TX) Frequency Range	TBD to TBD MHz	VHF Passband receive frequencies inverted
Transponder Bandwidth	30 kHz (3 dB)	
Maximum RF Output Power	1W (+30 dBm) max.	Linear Transponder Power Amplifier $P_{out\ peak}$ (SSB full loading; including TLM channel)
Frequency Stability (L.O.)	±25 ppm	-20° C to +50° C
Spurious Products	≥ -20 dBc	
Component Temp. Range	-40° C to +85° C	
DC Power (maximum)	~2.6 W (625 mA @ 5.2V)	Transmitter power is dynamic and depends on transponder loading and Average Power Tracking (APT) performance

Table 1: UHF Passband Transmit (TX) Signal Chain Performance Values

PERFORMANCE PARAMETERS	REQUIRED VALUES	COMMENTS
VHF Uplink (Transponder) Receive Frequency Range	TBD to TBD MHz	
Transponder Bandwidth	30 kHz (3 dB)	
Minimum Discernable Signal (MDS)	≈ -110 dBm	for 10dB S/N SSB signals
Frequency Stability (L.O.)	±25 ppm	-20° C to +50° C
Noise Figure (NF)	<3.0 dB	
Third Order Output Intercept Point (OIP3)	>5 dBm	
IMD Levels	>-50 dBc	
Component Temp. Range	-40 °C to +85° C	
DC Power (Maximum)	~208 mW (40 mA @5.2V)	Includes Command Channel Receiver

Table 2: VHF Passband Receive (RX) Signal Chain Performance Values

PERFORMANCE PARAMETERS	REQUIRED VALUES	COMMENTS
Telecommand Uplink Receive Channel Frequency	TBD MHz	Exact frequency is sensitive information
Channel Bandwidth	≈ 20 kHz	NBFM (± 5kHz Deviation)
Minimum Discernable Signal (MDS)	-110 dBm for a BER of 1 x 10^{-4}	AFSK-FM (see Command and Control Interface Specification) – SENSITIVE DESIGN INFORMATION
Noise Figure (NF)	≤ 3.0 dB	
Component Temp. Range	-40° C to +85° C	

Table 3: VHF Uplink Command Channel Receive Signal Chain Performance Values

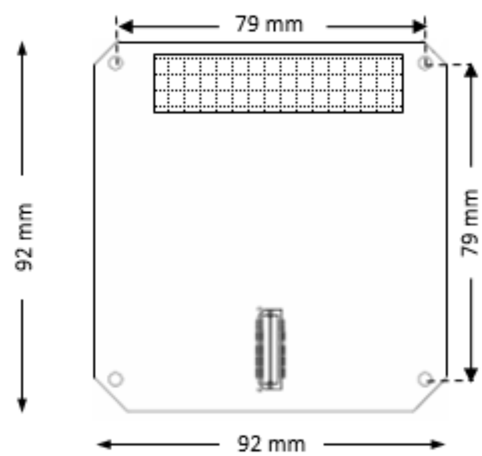
PERFORMANCE PARAMETERS	REQUIRED VALUES	COMMENTS
UHF Telemetry Downlink Transmit (TX) Frequency	TBD MHz	1200 bps (channel rate) BPSK
Telemetry Downlink Channel Maximum Output Power	200 mW (+23 dBm) max.	Peak Power
Frequency Stability (L.O.)	±25 ppm	-20° C to +50° C
Component Temp. Range	-40° C to +85° C	
TLM TX ONLY; DC Power (maximum)	1.1 W (210 mA @ 5.2 V)	@100 mW RF telemetry-beacon RF output

Table 4: UHF Downlink Telemetry Transmit Signal Chain Performance Values

2.1.2. Structure and Electrical Lines

2.1.2.1. Structure

The LTM is not designed to pass PC-104 connector bus lines. Therefore, it must be located at either end of a Partner University's PC-104 compatible PCB stack. A Partner University's interfacing bus connector can be a PC-104 compatible male or female mate. The LTM mechanical design consists of three Printed Circuit Boards (PCB) built to the dimensions shown in Figure 3 below. The LTM's PCBs are 2 and 4-layer FR-4 material with ~8 mm inter-board spacing and interconnected by SAMTEC QTH/QSH connectors, a TX/RX - ICR inter-board RF coax, and a PC-104 bus connector. Figure 3 below also shows a 3D rendering of the three board LTM.



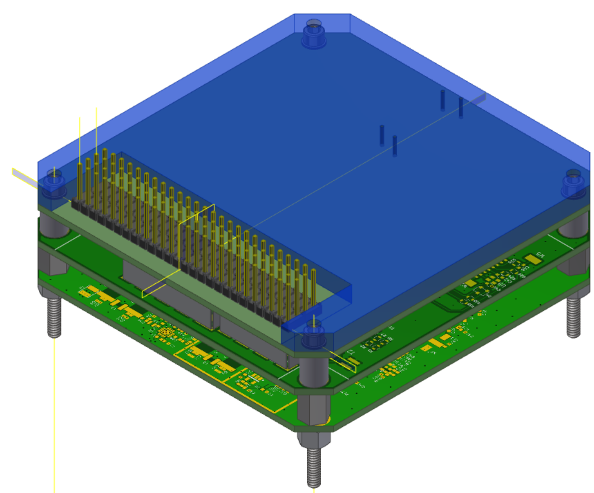


Figure 3: LTM PCB Dimensions and 3D Rendering

2.1.2.2. PC-104

The major signal interface lines between the LTM and the Partner University CubeSat subsystems are provided via a PC-104 connector (see Figure 4). This connector can be provided as either male or female and can be arranged so the LTM can be either on the bottom or the top of the host platform's board stack. Gender and end are specified in the host platform-specific appendix. The LTM receives primary DC power over this connector via two Vsys and two GND pins. These pins are equivalent, i.e., they are connected together at their power source, the host platform's Electrical Power Subsystem (EPS).

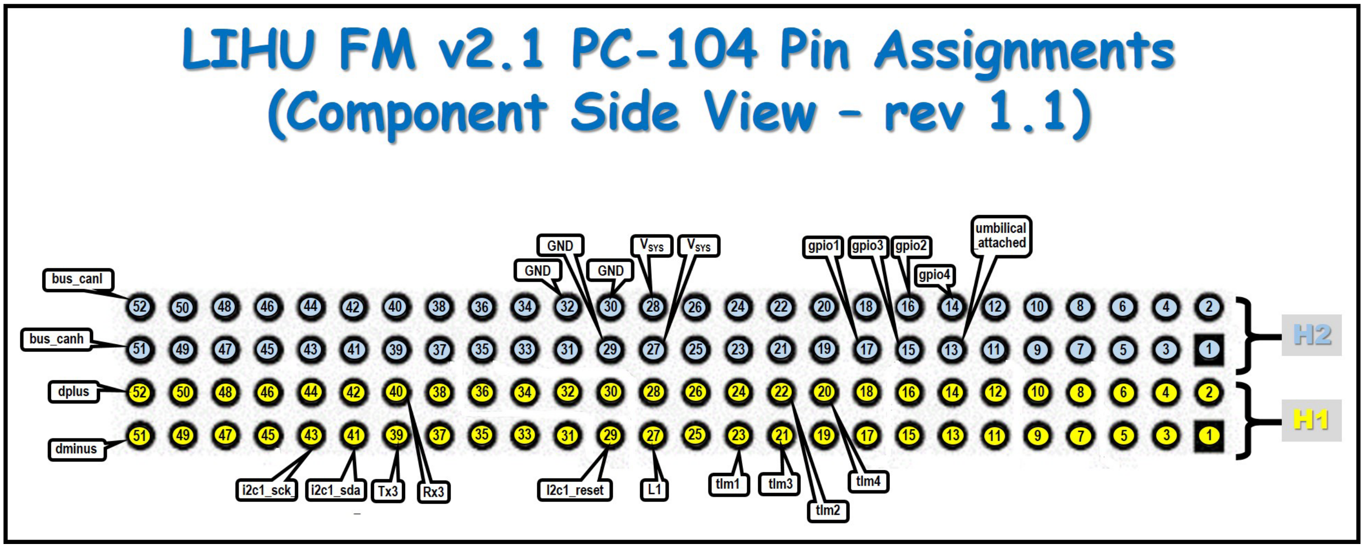


Figure 4: LTM PC-104 pin assignments (component side view)

2.1.2.3. Test and Operational Configurations

The LTM interface functions are designed to provide reliable VHF uplink and UHF downlink operations once the Partner University's CubeSat host platform is in orbit.

However, before this, during development, all required interface functions should be verified through RF and baseband testing via connections between (laboratory simulated) ground station command data and the Partner University's CubeSat platform. To support initial Partner University Lab development/testing, AMSAT will provide an early LTM EM unit along with a Familiarization and Acceptance Test Procedure (ATP) document. This will allow the Partner University's Development Team to get familiar with the LTM design and operation via a laboratory test configuration set-up as illustrated in Figure 5 below:

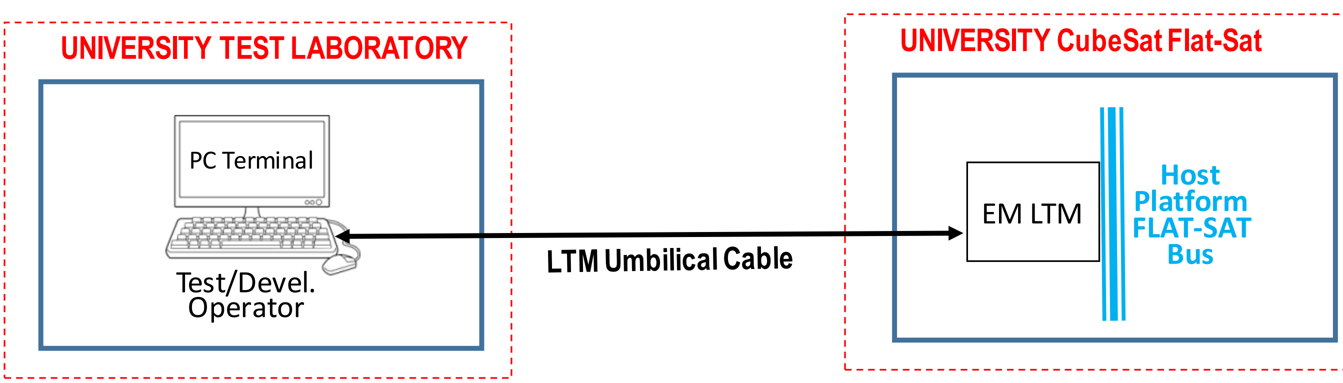


Figure 5: LTM Interface Laboratory Test Diagram

While on orbit, LTM supported ground communications with a University CubeSat should be provided through the operational configuration illustrated in Figure 6 below:

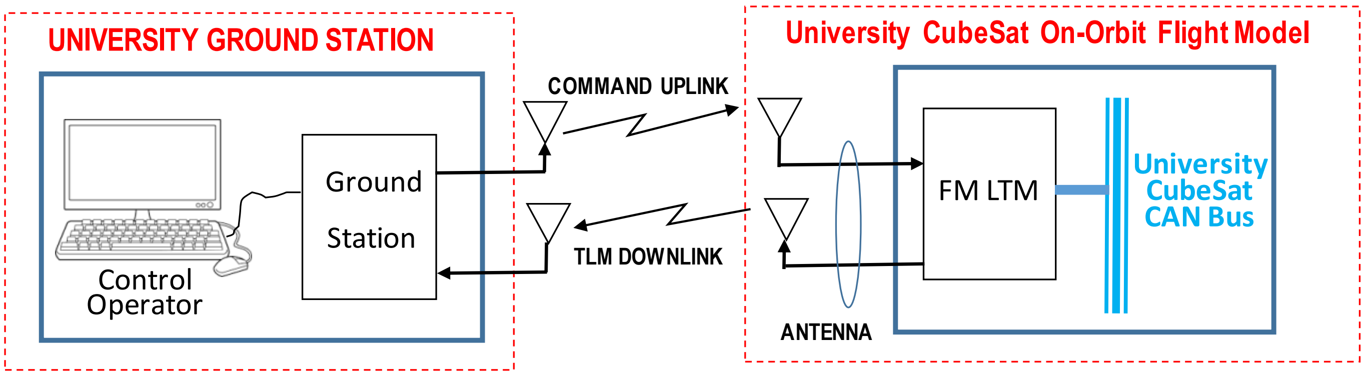


Figure 6: On-Orbit LTM - University CubeSat Communications Diagram

Additionally, LTM subsystem functions will support amateur radio satellite VHF/UHF transponder operations. These functions will be controlled by authorized AMSAT ground stations and will employ some of the same VHF uplink and UHF downlink capabilities utilized by the host platform's other subsystems.

2.1.2.4. Signal Buses

- CAN Bus - The CAN bus will be the main data interface between the LTM and the Partner University's host platform. The CAN bus is a 5V bus on the LTM. However, due to the differential nature of the CAN bus, the host platform may use either a 5V or a 3.3V bus. The CAN bus will run at 125K bits/second. The CAN ID will be of the extended form using 29 bits. CANH and CANL lines are provided on pins H2/51 and H2/52 respectively of the PC-104 host platform interface (see Figure 4 for PC-104 pin details).
- I2C Bus – The LTM provides an Inter-Integrated Circuit (I2C) bus which is slower and less capable than the CAN bus, but which may be used for direct connections to several I2C - equipped chips, viz., the ADS7828 (12-bit, 8 input A-to-D converter chip); the PCA9539 and MAX7320 GPIO devices; the MAX31725 (temp sensor); or to an ISIS CubeSat antenna (if equipped). SCK and SDA I2C lines are available on the LTM PC-104 connector interface on pins H1/43 and H1/41 respectively (see Figure 4). The LTM will provide the required pull-up resistor on the I2C data lines. In addition, pin H1/29 is a "reset" line that is normally HIGH. When this line is lowered and raised, all I2C devices are expected to reset (if they have a reset capability) or power off and on.
- USB Bus – The LTM provides a USB bus or a serial bus for use as an umbilical. There is a USB mini-B connector on the LIHU board itself. In addition, an external USB connection may be elsewhere on the host platform spacecraft; the USBP and USBM lines are available on the PC104 host platform interface pins H1/52 and H1/51, respectively for this purpose. The USB mini-B connector on the LTM can also supply +5V power to the LTM for testing/checkout before it is integrated with the host platform.

2.1.2.5. Serial Lines

There are two UART lines provided for possible use as an umbilical. "UART lines" means that the protocol used is that of RS232, but the voltages are TTL levels such as a USB-to-serial converter or the direct output of a Universal Asynchronous Receiver Transmitter (UART) might produce. These lines are an alternative to USB for console and program loading, and are available on PC-104 pins H1/39 for STx and H1/40 for SRx. See this ICD's host platform appendix for whether the serial lines or the USB lines are used.

2.1.2.6. Analog Telemetry Lines

There are four lines (TLMn) on which the host platform may place analog voltages between 0 and 3.3V to be A to D converted to digital signals and sent to ground stations as telemetry data. These lines are on pins H1/20-23 of the PC-104 connector.

2.1.2.7. Discrete Digital Lines

- Attached (input to LTM) - Pin H2/13 of the PC-104. The 'Attached' signal is an input to the LTM. The intent is that it will be raised when an umbilical connector is attached to the spacecraft. In addition, this signal is raised if a USB connection with power is plugged into the USB mini-B connector on the LTM board itself (see software section 2.1.3.4. for behavior).
- GPIOn (input or output to LTM) - There are four digital signal lines available on the PC-104 connector (pins H2/14-17); two for input and two for output. These can be used either as deployable sense/release signals or as unspecified reads and writes (see software section 2.1.3.4. for behavior).

2.1.2.8. LTM Power Requirements

The host platform's power system is responsible for generating solar power, maintaining battery health, and providing power to all CubeSat subsystems, including the LTM. Electrical power to the LTM shall be supplied via the LTM's V_{sys} lines. The V_{sys} voltage range will be between 5.2 and 6V. Upon deployment from the launch vehicle's dispenser, the powered LTM can be responsible for deploying the Partner University CubeSat's antennas TBD minutes after initial boot up. This a Partner University option (see Appendix).

The host platform design is responsible for the deploy switches and the Remove Before Flight (RBF) pin, and must ensure the LTM remains UNPOWERED after the host platform satellite is integrated, tested, and prepared for launch and before the satellite is released from the launch vehicle, with the following exception. If the 'Attached' signal is HIGH, and an umbilical is plugged into the satellite for battery charging, power **may** be supplied to the LTM. If the 'Attached' signal is HIGH, and an umbilical is plugged into the satellite for testing, power **must** be supplied to the LTM. Table 5 below shows the current design measurements of the power per LTM board.

PCB	ESTIMATES (from Eng Model Measurements)	COMMENTS
Improved Command Receiver (ICR)	200 mW	
Receive/Transmit (RX/TX)	2.6W (P_{OUT-PK} = 30 dBm)	With Average Power Tracking (APT); Power Dissipation will be dynamic, depending on Linear Transponder loading
Legacy Internal Housekeeping Unit (LIHU)	100 mW	
TOTAL DC POWER INPUT	≈ 2.9 Watts (max.)	All power numbers are worst case maximum/peak values with all Communication functions active. Average power dissipation will depend on transmission duty cycle.

Table 5: LTM Power Requirements

2.1.3. Software and Data

2.1.3.1. CAN

- CAN Bus – The CAN bus will be the main data interface between the LTM and the host platform satellite. It will be used by the LTM to send commands received from the ground to the host platform, and to send telemetry information from the host platform to the ground via the LTM's telemetry capabilities. CAN will also be used to pass information between the LTM and the host platform. For example, the LTM can tell the host platform that it has switched to Safe Mode and will not be downlinking all the telemetry from the host platform. Similarly, the host platform can signal the LTM that the LTM should go into Safe Mode to reduce power consumption. The CAN bus may be used internally by the host platform or by the LTM using identifiers compatible with the list specified herein.
- Message Format - CAN messages will comprise a 29-bit ID, and a DLC specifying the number of data bytes, viz., from 0 to 8 data bytes. The format of the IDs required by the LTM is shown in Table 6 below.
- Specific Messages Between LTM and Host – See the tables below.
- Foxtelem Display: The default labels for telemetry values are shown, but they can be customized for each satellite.
- Messages Uplinked from the ground. Any message can be sent from the ground to the host with the following restrictions: 1) Bits 28:25 are not uplinked, and will be filled in by the LTM. They must be ignored by the host. Bits 15:12, the destination field (see below) must be in the host's range (8 or above). Any number of data bytes between 0 and 8 may be sent.

BIT NUMBERS	PURPOSE	Comments and Specific Values When Required
28:24 (5 bits)	**Priority**	**If multiple messages are sent on the same bus simultaneously, the hardware prioritizes lower ID numbers. The host platform and LTM must accept messages with any priority and ignore the priority when interpreting the message. For opaque uplinked CAN messages, bit 24 may be used by the host for any purpose. Bits 28:25 will be set by the LTM software to ensure a high priority**
23:20 (4 bits)	**Source**	**The host platform must put a number 8 or above in this field if the destination is 3 or 0 (that is the LTM). Otherwise, the host platform may use this field as it wishes. The LTM will place 3 in this field when sending to the host platform. For opaque uplinked CAN messages these bits may be used for any purpose**
19:16 (4 bits)	**Type**	**The LTM will treat messages received from the host platform in accordance with their type as follows: 1,9: Telemetry to be downlinked as opaque CAN messages in realtime during safe mode 2,10: Telemetry to be downlinked as opaque CAN messages in realtime during health mode 3: Telemetry to be downlinked as opaque CAN messages in science mode 4: Passing status information (problem requiring a switch to Safe Mode for example) to coordinate actions between the host platform and the LTM. (see software section 2.1.3.7.)**

		5: Telemetry to be displayed on FoxTelem as health data and saved in health whole orbit data **8,9,10: Telemetry to be saved as Whole Orbit Data (WOD) (Note that adding 8 to telemetry types 1 and 2 means to send both in realtime AND saved as WOD)** **The LTM will use the following values when sending messages to the host platform:** **4: Passing status information (such as operational mode, or spacecraft elapsed time) to coordinate actions between host platform and the LTM** **For an opaque uplinked command, this field may be used for any purpose.**
15:12 (4 bits)	**Destination**	**The LTM will only receive messages from the host platform which have these ID bits set to 3 or 0. The LTM will put 8 in this field when transferring uplinked commands or status information to the host platform. The host platform must ignore messages that have this field set to less than 8. The LTM will ignore received messages with number 8 or greater and any such messages may be freely used within the host platform.** **For opaque uplinked commands, this field must be 8 or greater or the LTM will not send the command to the host.**
0:11 (12 bits)	**Message ID**	**For opaque CAN status sent from the host platform, this field is sent as part of the telemetry and is "opaque" to the LTM.** **For opaque uplinked commands sent to the host platform, this field may be used for any purpose.** **For status messages in either direction see Table 7.** **For health messages, see Table 8.**

Table 6: CAN Identifiers

Message ID (Bits 0:11)	Meaning	Data Units and Scale
0	Ignored	None
1	Enter Safe Mode	If LTM is source: Data 0=reason; 0=commanded, 1=low power If host is the source: TBD
2	Enter Health Mode	Data 0=reason; 0=commanded;1=power ok; 2=timeout of science
3	Enter Science Mode	Data 0=reason; 0=commanded; 1=Scheduled Time; Data 1=timeout (in minutes)

Table 7: CAN Status Messages (Type 4)

Message ID (Bits 0:11)	Meaning	Data Units and Scale
8	Solar Panel Temperatures (displayed and labelled on FoxTelem health page)	Data 0=+X, Data 1=-X, Data 2,3=+-Y Data 4,5=+-Z Unit: Degrees C, Scale: 0-255 = -20C to +107.5C
9	Other Temperatures (by default displayed as Temp 0, Temp 1, etc on FoxTelem health page	Up to 8 temperatures in data 0-7 Unit: Degrees C Scale 0-255 = -20C to +107.5C
10	Other Temperatures (by default displayed as Temp 8, Temp 9, etc on FoxTelem health page)	Up to 8 temperatures in data 0-7 Unit: Degrees C Scale 0-255 = -20C to +107.5C
16	Solar Panel Voltages (by default displayed and labelled on FoxTelem health page)	Data 0=+X, Data 1=-X, Data 2,3=+-Y Data 4,5=+-Z Unit: Volts Scale: 0-255 = 0-10V
17	Other Voltages (by default displayed and labelled as Voltage 0, 1, etc on FoxTelem Health Page)	Up to 8 voltages in data 0-7 Unit: Volts Scale: 0-255 = 0-10V
18	Other Voltages (by default displayed and labelled as Voltage 8, 9, etc on FoxTelem Health Page)	Up to 8 voltages in data 0-7 Unit: Volts Scale: 0-255 = 0-10V
24	Binary State Data 1 (by[1] default displayed and labelled as State n)	Up to 32 state bits in data 0-3. Unit: True/False Scale: 0=False, 1=True
25	Binary State Data 2 (by default displayed and labelled as State m)	Up to 24 states in data 0-2 Unit: True/False Scale: 0=False, 1=True
26	Binary State Data 3 (by default displayed and labelled as State n)	Up to 16 states in data 0,1 Unit: True/False Scale: 0=False, 1=True
27	Binary State Data 4 (by default displayed and labelled as State o)	Up to 8 states in data 0 Unit: True/False Scale: 0=False, 1=True
32	UTC Time (from GPS for example)	Data0=Year, Data1=Month (1-12), Data2=Day (1-31), Data3=Hour (0-23), Data 4=Minute (0-59), Data 5=Second (0-59)

Table 8: CAN Downlink Health Telemetry Messages (Type 5)

[1] 32 state bits in all may be divided among the 4 state messages in multiples of 8 at host's option. Binary State Data 1 message must have greater than or equal to BSD 2 >=3>=4.

2.1.3.2. I2C Bus

- Protocol - The LTM is always the I2C master and thus the host platform cannot initiate any I2C transactions. The I2C bus will run in standard mode; speed is not guaranteed, but currently it is 10K bits/second. The LTM, when accessing the I2C devices, will follow the formats specified by the datasheets of the respective devices.
- Devices and Addresses - The LTM will provide I2C support for a limited number of devices as listed in Table 8. The addresses for the available I2C devices are also listed in Table 8. Note that the host platform circuit designer can choose from several addresses for most of these devices; the host platform design is responsible for using the correct addresses. Which I2C devices will be used and how GPIO devices will be used is specified in the host platform Appendix.

I2C ADDRESS	HOST PLATFORM SLAVE DEVICE	COMMENTS
0x31,32	**ISIS Antenna**	**Two center-fed dipole antennas with temperature, deploy sense, deploy command and related (two addresses – one backup)**
0x48-4B	**ADS7828**	**Programmed as single ended; raw data downlinked in telemetry. A maximum of 4 ADS7828s may be used by the host platform.**
0x4C-54[2]	**MAX31725**	**Temperature**
0x55-5F	**MAX7320**	**GPIO I/O**
0x74-77	**PCA9539**	**GPIO I/O**

Table 8: I2C Devices and Addresses
(when not specified, max. number of devices TBD)

2.1.3.3. Umbilical Choices

The LTM provides two choices for an umbilical connection that can be used for testing and software updating after the satellite is assembled. The choices are a serial line or a USB bus. Either can be used to update the code running on the LTM's LIHU, or to enter typed commands to the LTM console, which will either initiate actions in the satellite, or elicit information to be output to the console. The choice of serial line or USB bus is specified in the host platform Appendix.

- USB Bus - The USB bus can be accessed via a mini-B USB connector mounted on the LTM or through a host platform-supplied umbilical connector. In either case, the LTM acts as a USB device to be connected to an external computer where it appears as a serial port (COMn on Windows; /dev/ttyACMn on Linux). When used as a console, testers will connect to this serial port with a terminal emulator to access the LTM. The LTM will accept text commands and provide text information, including LTM health telemetry as well as raw host platform telemetry. When used to load software, a PC (Linux, Windows, or Mac) program supplied by AMSAT can be used to load the LTM code. (See 'Attached' line for related information; section 2.1.3.4.).

[2] Note that the datasheet for the MAX31725 specifies the address as a full 8 bits with the least significant bit always 0. These addresses are listed in the more common 7-bit form. Thus what the ICD calls address 0x4C, the datasheet calls address 0x98.

- UART Lines - The UART lines can be accessed by header pins on the LTM's LIHU or via a host platform-supplied umbilical. The UART lines run at 57,600 baud and, when connected to a customer-supplied PC serial port, provide the same functions as described in the USB Bus paragraph above.

2.1.3.4. Discrete Connections – A discrete connection means a connection that is used for single purpose and has a single source and single destination. The LTM provides 9 discrete connections, described below.

- Analog Telemetry Lines - There are four lines called tlm1-tlm4 on which the host platform may place an analog voltage between 0 and 3.3V. These voltages will be digitized at 8 bits per value collected about every four seconds and sent to the ground by telemetry. The LTM samples these lines asynchronously, although not every sample will be sent by telemetry. It is the host platform's responsibility to ensure that these signals are meaningful at all times. For example, if this voltage varies and only the average is meaningful, the host platform must employ some sort of averaging circuitry.
- Digital Command Lines
  - *Attached (input to LTM):* When high, 'Attached' inhibits the radio from transmitting, avoids some initial testing or timing for faster startup when "on the bench," and avoids sending any signals to release antennas. The intent is that it would be raised when an umbilical connector is attached to the spacecraft. In addition, this signal is raised if a USB connection with power is plugged into the USB mini-B connector on the LIHU board itself. Another function of 'Attached' is that when it is high and the LTM is powered up or reset, the flash loader is started rather than the main flight code. This allows the user to load new AMSAT-supplied code for the LTM using the AMSAT-supplied pyMicroloader. If no code is to be loaded, a terminal command to the boot loader will start the flight code.
  - *GPIO1,2 (output from LTM):* These are two digital outputs (to the host platform) signal lines. They can be used for one of two purposes (choice to be specified in the host platform Appendix):
    (1) As commands automatically issued by the LTM to release antennas or other deployables (see the "Deployables" section 2.1.3.5. below). Note that these are low current logic signals, so if they are used, for example, to heat a burn wire, the host platform must supply a switch (a transistor for example). If this use is chosen, the deployment can also be activated by a ground or umbilical command;
    (2) As a logic signal unrelated to deployables and activated by ground command(s).
  - *GPIO3,4 (input to LTM)*: These are two digital inputs (to the LTM) signal lines. They can also be used for two purposes (choice to be specified in the host platform Appendix):
    (1) As sensors to indicate the deployables have been released (LOW = stowed, HIGH = released). The sensor state will also be telemetered to the ground; or
    (2) As sensors unrelated to deployables to be telemetered to the ground.

    Note: the host platform must choose the same option (1 or 2) for all GPIOs.

2.1.3.5. Software Behavior: startup phases – Every time the LTM flight software starts, following a power cycle, following a reset, or following the first release from the launch vehicle, it passes through several phases of startup before reaching the main software. These phases are described below:

- Flash Loader (Startup Phase 1) - The first phase to execute is the flash loader. If the "Attached" digital signal is high, the flash loader waits for a command from a terminal or the pyMicroLoader software on a PC to either continue to the next phase, or to load software from the attached PC into the LIHU's flash program memory. If pyMicroLoader supplies new code for the remaining startup phases and the main software, this code is written into the flash program memory and then executed. If there is just a command to continue, the existing code is executed. Note that the flash loader code is permanently installed in the processor's flash program storage and cannot be replaced except by having physical access to the LIHU. The flash loader can replace the remaining startup code as well as the main LIHU code.

- POST (several places in startup) - At several places during the remaining startup phases, power-on self-tests (POST) are run, including a CRC check on several sections of the flash program memory, a processor self-test, and a memory test. If any of these tests fail, the behavior is different depending on the state of the 'Attached' digital signal. If the 'Attached' signal is HIGH (for example, when on the ground in the lab) the CRC is recalculated and stored; other failure information is passed on to be displayed on the console. If 'Attached' is LOW for either case, the LIHU is power cycled in an attempt to fix what may be a deposited charge problem.
- Post-release timer (Startup Phase 2) - If 'Attached' is HIGH, or if there has been a successful complete startup since release from the Launch Vehicle, the post-release-timer does not execute, but skips to the "deployable release" phase. Otherwise, the LIHU uses absolutely minimal resources (no memory, no interrupts, no crystal clock—only internal RC time constant timing) to wait for the post-launch delay (about 45 minutes by default; see host platform Appendix if the host platform requires a different time). During the wait time, a 'sonalert' tone generator will be audible as a warning in case the timer is being tested or entered accidentally on the ground.
- Deployable release (Startup Phase 3) - Following the post-release timer, the 'deployable release' phase is entered. Again, if the 'Attached' signal is HIGH, this phase is skipped and the main flight code is entered. If there has been a successful boot previously, and if the deployable sensors indicate that the deployables have been released, this phase is skipped. Otherwise, the deployables are released one at a time. If the sensors show 'released' when the code is first executed, it is assumed there is a problem with the sensors, and the deployables are released anyway with an extra-long burn time (~10 seconds). If the sensors show stowed, deployables are released with a normal length burn time (~5 seconds) and then the sensors are checked to verify success. If the sensors show no-deploy, this procedure is repeated with a longer burn time. Then, execution continues on into the main software.

2.1.3.6.Software Behavior: major functions of the main software - The major functions of the LTM's flight software are to:

- Keep track of time (LTM Timekeeping),
- Receive and execute (or pass to the host platform) ground commands,
- Collect telemetry data from itself and the host platform and format it for downlinking,
- Implement a 1200 bps BPSK telemetry beacon,
- Implement operational modes that choose which telemetry data to send and when,
- Implement a Whole Orbit Data (WOD) collection scheme so telemetry can be collected at all times in orbit and then repeatedly downlinked so it is likely to be seen by some ground station(s), and
- Manage the LTM's amateur radio transponder based on ground commands and the operational mode being executed.

2.1.3.7. Software Operational Modes - The LTM flight software, after startup is complete, has three basic operational modes: (1) Safe Mode, (2) Health Mode, and (3) Science Mode. In addition, the amateur radio transponder can be enabled by ground command, but will only be turned on in Health Mode. The operational mode is saved in non-volatile memory so that if the processor resets, it will re-enter the mode that it was previously in. Science Mode is slightly different in that it includes a timer so that it will return to Safe or Health mode when it times out, or if the processor should reset while it was active.

- Safe Mode: In Safe Mode, the intent is to attempt to reduce the power consumption of the satellite as much as possible. In particular, a CAN message is sent to the host to reduce power, the amateur radio transponder is turned off, the telemetry beacon is changed to be on for approximately 10 seconds (2 frames) every 2 minutes, and the transmitter power is set to low. The telemetry transmitted in Safe Mode is limited to health data from both the LTM and the host platform (including whole orbit health data). When first released from the launch vehicle, the LTM is in Safe Mode and will not change modes until commanded from the ground. In addition, the LTM has the capability of "auto-safe" which causes it to automatically enter and exit Safe Mode when a low voltage bus condition occurs (the host platform Appendix specifies what that voltage will be).

- Health Mode: This is the "normal" mode in which the LTM operates. When in Health Mode, the LIHU continually sends telemetry, including health data, whole orbit data, and data of the host platform's choice. In Health Mode, it is possible to turn on the amateur radio transponder.
- Science Mode: When commanded into Science Mode, the LTM "remembers" its previous mode and sets a timer (specified within the Science Mode command). When the timer expires, the LTM returns to its previous mode. Science Mode is similar to health mode, except that the telemetry information changes to allow for more host platform science data to be sent at the expense of health and whole orbit data.

2.1.3.8. LTM Timekeeping

The LTM keeps track of time as a pair of numbers (epoch, seconds-in-epoch). The epoch number increments when the software initializes, i.e., following a reset. Thus, it is sometimes called a reset number. When the epoch increments, the seconds start at 0. The spacecraft time is guaranteed always to increment if the epoch number is considered more significant than seconds. It is included in every telemetry frame sent to the ground to indicate the time sent, in whole orbit data to indicate the time the data was collected, and in min/max data to indicate the last time a new minimum or maximum was observed.

The LTM is designed to reset occasionally due to failures that might occur from radiation, charged particle impacts, power problems or software bugs. On-orbit previous experience has shown that a reset is common every few days and frequently happens near the South Atlantic Anomaly, implying that charged particles are a likely cause.

The LTM does not have a real-time clock that 'lives' across resets, nor does it have a GPS nor any other source of UTC.[3] However, on the ground, it is possible to correlate spacecraft time with UTC. This is accomplished by comparing the spacecraft time in a telemetry frame with the ground UTC when the telemetry was received, and using that to determine the UTC of the start of each epoch.

Optionally, the host may provide UTC time over CAN (received, for example, by a GPS unit). In this case, the spacecraft time (epoch, seconds) is still used as above, but an occasional telemetry frame that include the UTC time will be sent to allow automatic calibration of the spacecraft time with UTC.

2.1.3.9. Telemetry

- Telemetry Format - Telemetry is downlinked as "frames" using BPSK (Binary Phase Shift Keying) at 1200 bits per second. All telemetry frames are the same size. The data consists of no more than 665 bytes (may vary by satellite), a 4-byte CRC check, and 3 32-byte Reed Solomon (255,223) check blocks for forward error correction (FEC). A frame requires roughly 6.6 seconds to send.

  A frame consists of a header followed by from 1 to 8 "payloads". The header includes the spacecraft time when the frame was assembled, the satellite ID number (assigned by AMSAT), the frame type,

[3] UTC: Coordinated Universal Time—the basis for orbit calculations and most other time functions on earth.

and the operational mode the LTM was in when the frame was assembled. Payloads consist of different kinds of data such as whole orbit data, host platform science data, LTM health data, and host platform health data. Each frame type has a specific set of payloads in a specific order. For example, frame type 1 might have LTM realtime health, LTM min values health, host platform short science, LTM whole orbit data, host platform whole orbit data, etc. Type 2 might be the same but with LTM max values health. Type 3 might be 3 LTM whole orbit data payloads and 3 host platform whole orbit data payloads. The exact definitions of the data will be in a spreadsheet available from AMSAT.
The composition and frequency of frames will depend on the mode of the spacecraft (Safe, Health, or Science) with details TBS.

- Payload Types
    - **Health** – A health payload contains information about the operation of the spacecraft. Besides data from the LTM module (temperatures, radio power, voltages) it will also contain data gathered from the host I2c devices, the host-defined analog lines, host GPIO input lines and type 5 CAN data sent by the host. For data that has numerical values (generally those that come from A/D converters) the LTM also keeps track of the minimum and maximum values and downlinks those in "Min" and "Max" payloads.

    - **Health WOD** – A health WOD payload is nearly identical to a Health payload with these exceptions: 1) A WOD payload is collected and stored in a circular, non-volatile spacecraft buffer every 60 seconds. 2) The spacecraft time and mode is also stored with the WOD payload. 3) Many (TBS) WOD payloads are downlinked per minute, so that each stored WOD payload will be sent multiple times before it is overwritten.
    Health and health WOD payloads are sent (at different rates) in Health and Safe mode.

    - **Science** – A science payload consists of 83 bytes of data from the host spacecraft passed to the LTM over the CAN bus. Science data passed in a CAN message consists of the least significant 12 bits of the 29-bit CAN ID, 4 bits indicating the number of data bytes (0-8), and the specified number of data bytes. Thus, a science payload can consist of between 8 and 40 CAN messages (depending on how much data each one holds).

    - **Science WOD** – Similar to the Health WOD, the Science WOD contains science payloads that are saved in a non-volatile circular buffer. One significant difference is that the host can specify for a given CAN message whether it should be saved in WOD, downlinked immediately, or both. Science and Science WOD payloads are sent only in health mode.

    - **Long Science** – A long science payload fills an entire frame, and thus contains at most 656 bytes, or between 66 and 332 CAN messages. Long Science payloads are sent only in science mode.
- Telemetry Reception (FoxTelem) - FoxTelem is a program that runs on Windows, Linux, or MacOS. It receives telemetry via either a software defined radio dongle (for example the FunCube Dongle) or direct sound card input from another radio. FoxTelem is designed to receive formats for many different satellites and to present the data in tabular or graphic form. It also has the ability to download data that was received around the world from a central server, to output CSV tables of data, and to provide heat maps indicating various signal characteristics based on the area of the sky where the satellite was located when the signal was received. FoxTelem is open-source software (sources on GitHub, username AC2CZ) and is available for download from the AMSAT website. A minimum of version 1.11 (not yet released) will be required for the LTM.

2.1.3.10. Commanding

Only a licensed amateur given permission by the satellite licensee is allowed to send commands to the satellite (in accordance with FCC regulations). Generally, this will mean a licensed Partner University control operator, or an AMSAT control operator, depending on who the licensee is, and what permissions the licensee has given.
The following types of commands can be sent:

- Generic CAN message commands - A command can consist of a CAN message that is 'opaque' to the LTM and which is simply passed to the host platform.
- Specific Commands - Up to 256 commands can be sent that are specific to a host platform. Each of these commands may include as many as four 16-bit arguments. Specific commands must be enabled for each individual AMCOM user. The specific commands will be defined in a separate document.
- LTM Commands – These are commands common to most LTM satellites that are largely used to control the radio and operation modes, as well as override some automatic functions for testing and debugging.

Commands are generated by Windows software called AMCOM and provided by AMSAT. AMCOM output is via a sound card which can be connected to a standard amateur radio transmitter. AMCOM can also turn on some transmitters automatically, either directly or via a small control box supplied by the partner. AMCOM requires a username and password to operate. Each individual user is given a constellation of satellites and commands that s/he is allowed to send, as well as granted or not granted the capability to send commands from an external source via AMCOM. AMSAT provides the authentication file which specifies a user's name, his/her constellation of capabilities, and an initial password. The command uplink format includes forward error correction, whitening, interleaving, and secure authentication.

3 INTERFACE VERIFICATION

This section defines a set of qualification methods to be used to verify that the requirements of the interfaces defined in Section 2 have been met. Qualification methods include:

DEMONSTRATION - operation of interfacing entities that relies on observations not requiring the use of instrumentation, special test equipment, or subsequent analysis.

TEST - operation of interfacing entities using instrumentation or special test equipment to collect data for later analysis.

ANALYSIS - processing of accumulated or generated data obtained from other qualification methods; e.g., reduction, calculation, simulation, interpretation, or extrapolation.

INSPECTION - visual examination of interfacing entities, documentation, etc.

SPECIAL QUALIFICATION METHODS - any special qualification means applied to the interfacing entities, such as special tools, techniques, procedures, facilities, and acceptance limits.

4 ACRONYMS GLOSSARY

ADAC	Attitude Determination & Control
ADC	Analog to Digital Converter
AMCOM	AMSAT Command (s/w pgm)
AMSAT	Radio Amateur Satellite Corporation
APT	Average Power Tracking
ASCII	American Std Code for Info Interchange
ATP	Acceptance Test Procedure
BPSK	Binary Phase Shift Keying
CAN	Controller Area Networking
CDR	Critical Design Review
CONOPS	Concept of Operations
CMD	Command
CRC	Cyclic Redundancy Check
CW	Continuous Wave (Morse Code)
dB	deci-Bel
dBc	dB's below Carrier
dBm	dB's relative to 1mW
DC	Direct Current
DLC	Data Length Code
EAR	Export Administration Regulations
EM	Engineering Model
EMC	ElectroMagnetic Compatibility
EMI	ElectroMagnetic Interference
EPS	Electrical Power System
FCC	Federal Communications Commission
FM	Flight Model
FoxTelem	Fox Telemetry (s/w pgm)
FR-4	Glass Reinforced Epoxy Laminate PCB
GND	Ground
GPIO	General Purpose Input - Output
GPS	Global Positioning System
I2C	Inter-Integrated Circuit
IARU	International Amateur Radio Union
IAW	In Accordance With
ICD	Interface Control Document
ICR	Improved Command Receiver
ID	Identification; IDentifier
IMD	Inter-Modulation Distortion
I/O	Input / Output
IT	Information Technology
ITAR	International Traffic in Arms Regulations
ITU	Ineternation Telecommunications Union
LIHU	Legacy Internal Housekeeping Unit
LO	Local Oscillator
LT	Linear Transponder
LTM	Linear Transponder Module
mA	milli-Ampere
MCU	MicroController Unit
MDS	Minimum Discernable Signal
MHz	Mega Hertz
MRR	Mission Readiness Review
mW	milli-Watt
NDA	Non-Disclosure Agreement
NF	Noise Figure
OBC	On Board Computer
OIP3	Output Intercept Point 3rd (Order)

PC	Personal Computer
PCB	Printed Circuit Board
PDR	Preliminary Design Review
POST	Power On Self Test
ppm	parts per million
QTH/QSH	AMSAT Fox CubeSat bus connectors
RBF	Remove Before Flight
RF	Radio Frequency
RX/TX	Receiver/Transmitter (LTM card)
SAT	Satellite; Saturated
SPI	Serial Peripheral Interface
SSB	Single Side Band
TBD	To Be Determined
TLM	Telemetry
UHF	Ultra High Frequency
USB	Universal Serial Bus
UTC	Coordinated Universal Time
VHF	Very High Frequency
WOD	Whole Orbit Data
XCVR	Transceiver

5 REFERENCES/APPLICABLE DOCUMENTS

1) CubeSat Design Specification, Rev. 13, February 20, 2014, The CubeSat Program Cal Poly SLO
2) ITU Radio Regulations, Edition of 2012, available from http://www.itu.int/publ/R-REG-RR-2012/en
3) Fox-1E Linear Transponder Communications Performance Analysis (Preliminary), AMSAT, DRAFT v0.4, 11 October 2016