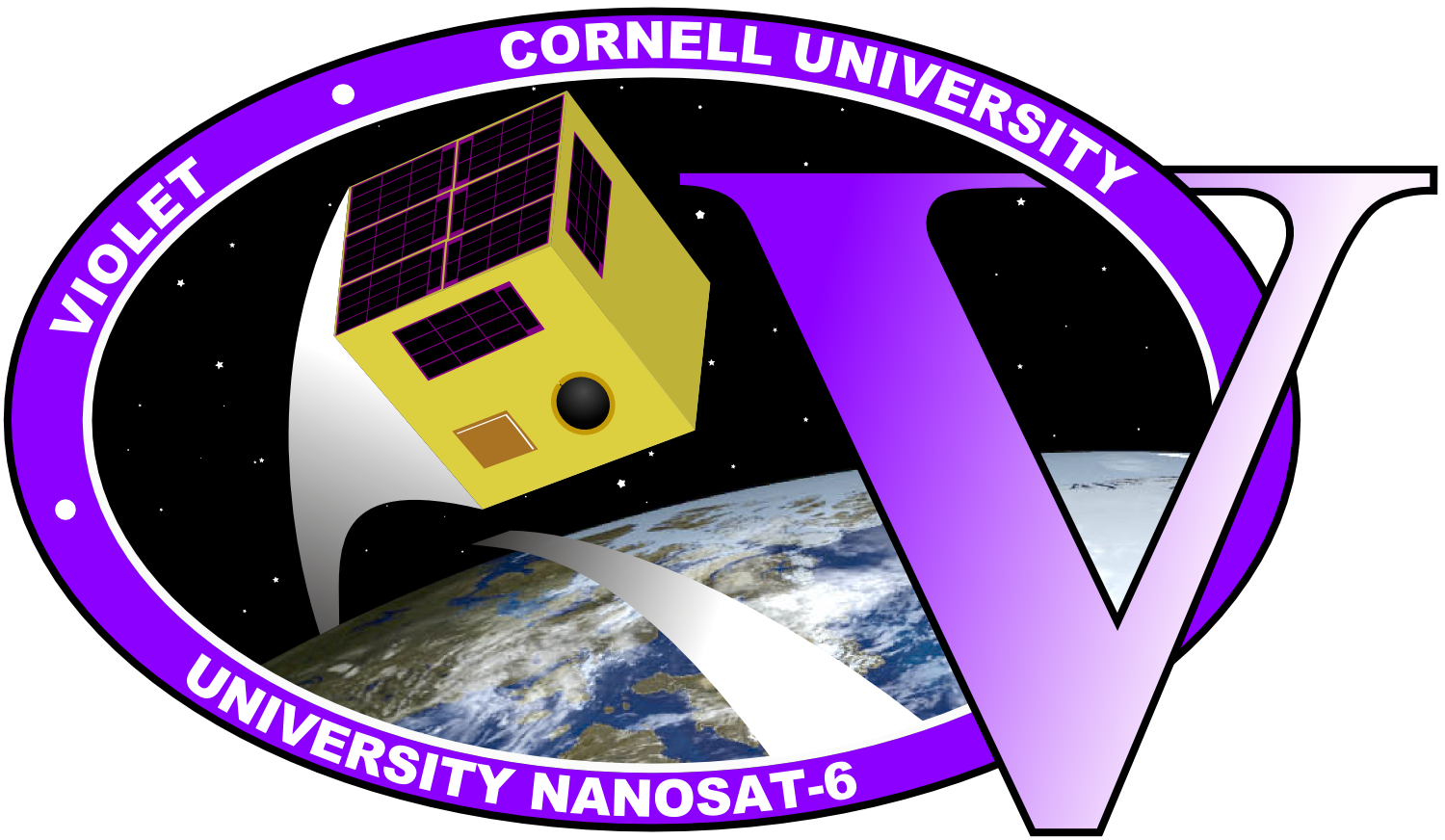
**Cornell University**

University Nanosat-6 Program

**Radio Interface Control Document**



|  |  |
| --- | --- |
| **Document ID:** | VIOLET-TC-010 |
| **Release Date:** | 12/4/2012 |

**Signature Page**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Authored |  |  |  | 12/2/12 |
| Action Taken |  | Jeffrey Niu – jn286@cornell.edu  PREP, Violet Satellite Project |  | Date |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Action Taken |  |  |  | Date |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Action Taken |  |  |  | Date |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Action Taken |  |  |  | Date |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Action Taken |  |  |  | Date |

**Document Revisions**

PREP – Preparer – Any qualified individual who serves as the preparer of technical data or who initiates an engineering action. The preparer is responsible for obtaining a supplementary engineering review and maintains overall responsibility if the supplementary review is not sought.

REA – Responsible Engineering Authority – Any qualified individual that reviews and approves the program’s technical data. The REA should have knowledge of the information presented in the document. There may be more than one applicable REA per document.

FAB – Fabricator/Assembler – The individual(s) responsible for fabricating or assembling the part.

IT – Integration and Test – The individual(s) responsible for planning, executing, and documenting the integration and testing of payload elements.

SY – Systems Engineer – The individual(s) responsible for defining system requirements and for ensuring the overall satisfaction of these requirements. Also responsible for identifying and resolving conflicts and discrepancies of an engineering nature.

QA – Quality Assurance – The individual(s) responsible for ensuring that parts are manufactured according to engineering specification. For the University Nanosat Program, QA may be performed by any individual that is independent of the Fabricator/Assembler and that has either witnessed the proper assembly of the part or can confirm that the part complies with all applicable engineering specifications.

CM – Configuration Manager – The individual(s) responsible for ensuring the accuracy of as-built documentation and for archiving all documentation related to the history of a part.

PM – Program Manager – The individual responsible for resource management and for overall program success.

PI – Principal Investigator – The individual responsible for overall mission design and its contribution to the scientific or engineering community.

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Description** | **Date** | **Approval** |
| **-** | **Initial Release for I&T**  Authored by Jeffrey Niu | **12/2/12** |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. Introduction 1

1.1. Overview 1

1.2. Scope 1

1.3. Applicable Documents 1

2. Physical Properties 2

2.1. Dimensions 2

2.2. Connectors 2

2.3. Operating Temperature 2

3. Electrical Properties. 3

4. Communication Interface 3

4.1. Structure of the Lithium Packet 4

4.1.1.Lithium Header Structure 4

4.1.2.Payload 6

4.1.3.Payload Checksums 6

4.2. AX.25 Structure 6

4.2.1.Beacon Use 7

**Listing of Figures**

Figure 1: Packets Overview………………………………………………………………………3

Figure 2: Lithium Packet Structure.………………………………………………………………4

Figure 3: Information Frame...……………………………………………………………………6

Figure 4: Control Field……………………………………………………………………………7

Figure 5: PID Field………..………………………………………………………………………3

**Listing of Tables**

Table 1: List of OP codes………………….………………………………………………...……5

1. Introduction
   1. Overview

This document provides an overview of the Li-1 radio. It includes the physical and electrical properties as well as the packet structure and other communication features of the radio.

* 1. Scope

This document describes the Li-1 Radio electrical and communication interfaces for interconnectivity between CDH and T&C.

* 1. Applicable Documents
* [Li1\_Radio\_Interface\_Manual](https://cusat.cornell.edu:8889/violet/Shared%20Documents/ELEC%20-%20Electrical/Data%20Sheet/Li1_Radio_Interface_Manual.pdf)
* [Lithium-User\_Manual](https://cusat.cornell.edu:8889/violet/Shared%20Documents/ELEC%20-%20Electrical/Data%20Sheet/Lithium-User_Manual.pdf)

1. Physical Properties
   1. Dimensions

Mass: 48-52g

Height: 10mm (max)

Width: 32-33mm

Length: 64-65mm

* 1. Connectors

RF: SMA, MCX

Serial interface: 3.3V UART (15pin)

Programming Port: 10 pin 1.27mm connector with .1” JTAG header

* 1. Operating Temperature

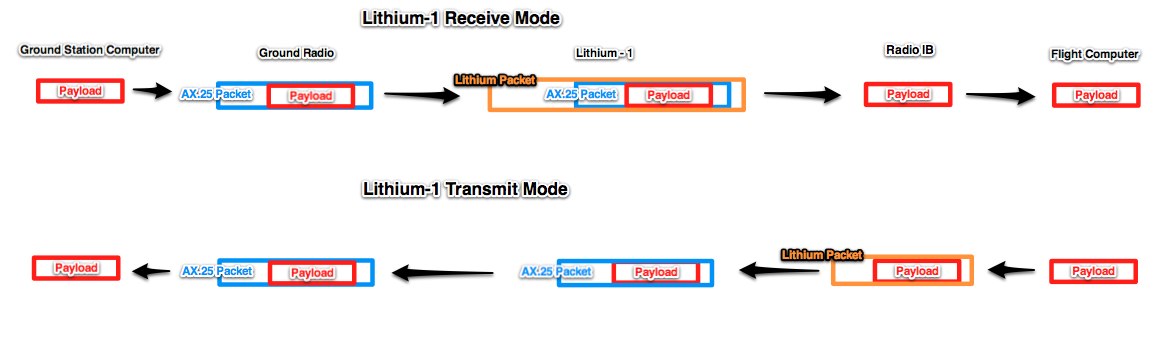
-30°C to +70°C

1. Electrical Properties.

|  |  |
| --- | --- |
| Input Voltages: | Digital: 3.3V  Radio Power Supply: 5-9V |
| Output Transmit Power: | 250mW – 4W |
| Power Usage: | Receive: < 200mW  Transmit: < 10W |
| Frequencies: | 130-450 MHZ (we use 437.4) |
| Modulation: | FSK/GMSK |
| Data Rate: | 9.6kbps |
| Protocol Support: | AX.25 |

1. Communication Interface

The Li-1 is a half-duplex radio with adjustable communication parameters. The radio communicates using a subset of the AX.25 packet protocol, with a custom lithium packet protocol between the radio and the interface board. The overall transmission process is illustrated in figure 1.

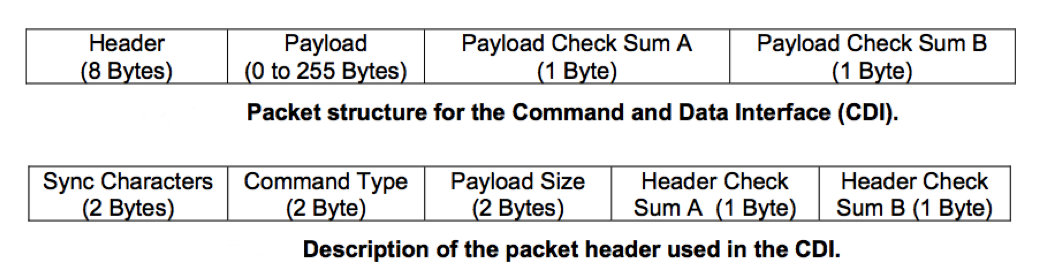
Figure 1:

note: the packet structure shown at each component is what will be leaving that component

* 1. Structure of the Lithium Packet

The packet format of the lithium wrapper is shown in figure 2 below:

Figure 2



* + 1. Lithium Header Structure
       1. Sync Characters

There are two bytes for Sync Characters

Sync Character 0: 0x48 (or “H”)

Sync Character 1: 0x65 (or “e”)

* + - 1. Command Type

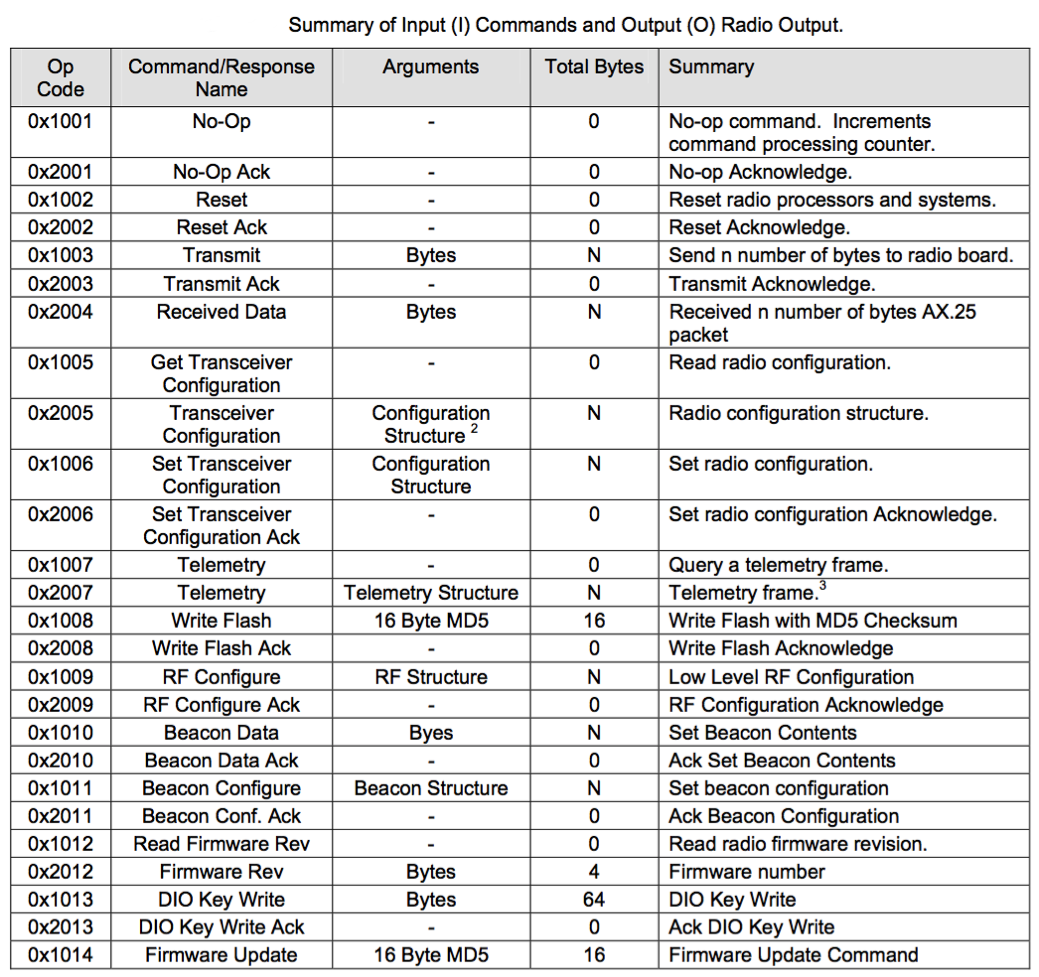
First byte signals whether data is entering or leaving the radio

I-messages are entering the radio and have first byte 0x10

O-messages are leaving the radio have first byte 0x20

Table 2 details the full list of commands, where Op Code is the two bytes of Command Type

Table 1



* + - 1. Payload Size

Can be one of three options depending on the operation

- will be nothing if it is a non-data command being sent from the computer to the radio

- will be 0x01 0x01 for acknowledge

- will be 0xFF 0xFF for not-acknowledge for a response being sent from the radio to the computer will be N, the size of the payload, when data is being carried

note: the payload size is the size of the actual payload, the AX.25 wrapper does not count towards the size of the payload

* + - 1. Header Checksums

Two checksum bytes based on the 8-bit Fletcher algorithm summarized below:

A buffer of N bytes, Buffer[N], contains data over which the checksum is to be calculated. The two checksum values (CK\_A and CK\_B) are 8-bit unsigned integers only. Note, if you implement it with larger sized integers, be sure to mask both CK\_A and CK\_B with 0xFF after the calculations complete to ensure they are 8-bit. Pseudo-code for checksum calculation is given below. For header checksum, Buffer[N] is the first 6 bytes of the header.

CK\_A = 0, CK\_B = 0

For(I=0;I<N;I++)

{

CK\_A = CK\_A + Buffer[I]

CK\_B = CK\_B + CK\_A

}

This loop calculates CK\_A and CK\_B which are then appended to the header.

* + 1. Payload

Following the header is the payload, which can be up to 255 bytes long.

* + 1. Payload Checksums

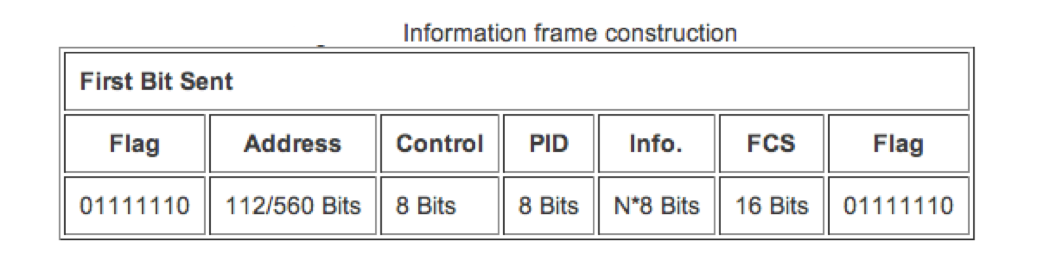
The two bytes of payload checksum are calculated across all pertinent bytes of the message excluding the two sync characters of each message (“He”).

* 1. AX.25 Structure

<http://www.tapr.org/pub_ax25.html#2.2.13>

Radio transmissions are sent in small blocks of data called frames. Each frame is made up of several smaller groups, called fields. The fields of an I (information) frame are shown in figure 3.

Figure 3



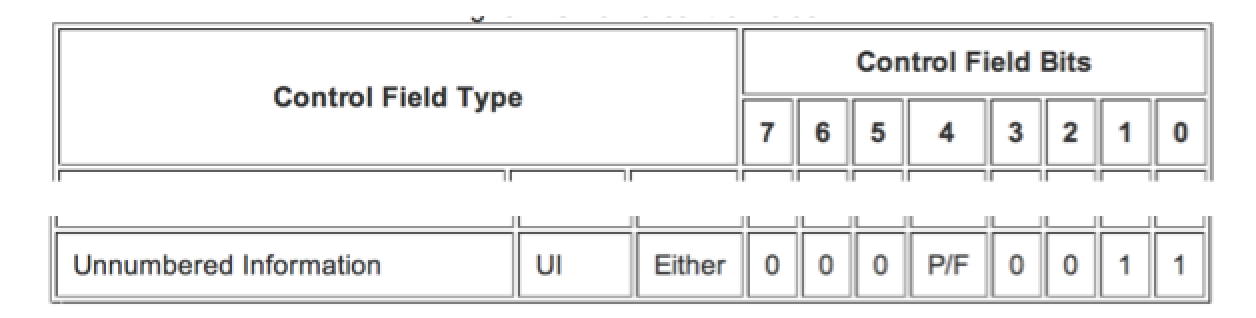
Only the handling of UI (unnumbered information) frames is implemented. UI frames are similar to information (I) frames, except they are unnumbered so there is no detection of order. UI frames also contain the PID and information fields of I frames. Since these frames are not acknowledgeable, if one gets lost, there is no way to recover it. A received UI frame with the P bit set shall cause a response to be transmitted. This response shall be a DM frame when in the disconnected state or a RR (or RNR, if appropriate) frame in the information transfer state.

The flag field is one octet long. Since the flag is used to delimit frames, it occurs at both the beginning and end of each frame. Two frames may share one flag, which would denote the end of the first frame, and the start of the next frame. A flag consists of a zero followed by six ones followed by another zero, or 01111110 (7E hex). As a result of bit stuffing, this sequence is not allowed to occur anywhere else inside a complete frame.

The address field is used to identify both the source of the frame and its destination.

The control field is used to identify the type of frame being passed. The control field for a UI frame will always be as follows:

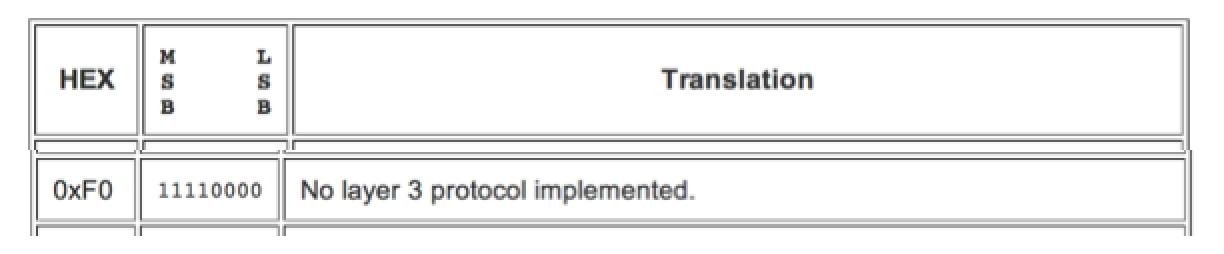
Figure 4



where P/F is the poll/final bit. The P/F bit set to P requires a response from the destination; set to F does not require a response. Either means the frame can represent either a command or response.

The PID (protocol identifier) field identifies what kind of layer 3 protocol, if any, is in use. We have no layer 3 protocol, so the following PID is used:

Figure 5



The info field conveys user data and can be up to 256 bytes long.

The frame check sequence is a sixteen-bit number calculated (using CRC) by both the sender and receiver of a frame. It is used to insure that the frame was not corrupted during transmission.

* + 1. Beacon Use

The beacon is a message that is transmitted intermittently based on user setting. The beacon data consists of up to 256 byes and is set using the write beacon data message. The UART message to load the beacon data matches the standard Li-1 interface format. The beacon data can be updated at any time during normal powered operation.

The Beacon is enabled by setting the beacon interval in the beacon configuration message to a value greater than zero. Each digit of interval corresponds to 2.5 seconds of delay. For example, a beacon configuration set with an interval of five will result in a beacon being transmitted every 12.5 seconds.

Beacon amplifier power level is the configured power level. Beacon Interleaving is automatically performed during a communication session unless the internal transmission buffers are full at the designated time of beacon transmission. If messages are currently in the buffer, the beacon transmission will be added to the buffer and will be transmitted upon its turn.