**Project Blue Horizon - Increment V**

**Enhanced Morse Decoder**

**Interface Control Document**

**(ICD)**

|  |  |
| --- | --- |
| **Revision** | DRAFT |
| **Issue Date:** |  |

|  |  |
| --- | --- |
| **Document Number:** |  |
| **Status:** | Release |
| **CDRL No:** |  |
| **DOORS™ Module Reference:** | N/A |
| **Classification:** |  |

Sean Winfree



Revision Status

The content of this document is wholly defined by the issue number and, where appropriate, the revision letter detailed below. The document will be reproduced in its entirety when any change has been incorporated and approved.

This document is under Configuration Control and any changes to the document will be subject to a change request via Configuration Management.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Issue/ Revision** | **Incorporated**  **by** | **Revision**  **Date** | **Affected**  **Pages** | **Comments** | **CR**  **Applied** | **Authorising**  **CN/POD** |
| - | Sean Winfree |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Authorization Notification

Prepared by: Sean Winfree, Chief Eng.

Authorized by: Sean Winfree

ASCEND

Contents

[1. Introduction 5](#_Toc279325148)

[2. Overview 6](#_Toc279325149)

[2.1 HF Uplink 7](#_Toc279325150)

[2.2 HF Datalink 7](#_Toc279325151)

[3. Enhanced Morse Decoder Interfaces 8](#_Toc279325152)

[3.1 Electrical Interface 8](#_Toc279325153)

[3.2 Mechanical Interface 10](#_Toc279325154)

[4. Serial Interface 11](#_Toc279325155)

[4.1 Content 12](#_Toc279325156)

[4.1.1 Control Messages 12](#_Toc279325157)

[4.1.2 Data Request Messages 14](#_Toc279325158)

Figures

[Figure 1: Morse Code CW 7](#_Toc279325107)

[Figure 2: Enhanced Morse Decoder System View 8](#_Toc279325108)

[Figure 3: EXAMPLE ONLY 10](#_Toc279325109)

Tables

[Table 1: Electrical Interfaces 8](#_Toc279325103)

[Table 2 - Message Format 11](#_Toc279325104)

# Introduction

This document covers the interfaces between Enhanced Morse Decoder and th Enhanced HF Provider. The messaging between the Air Segment and Ground Segment are defined in SW-005 ICD Messaging Inteface.

# Overview

The Enhanced Morsed Decoder translates Constant Wave (CW) pulses, received at a defined frequency, into ASCII characters and publishing the charaters over a serial interface.

The decoder begins by evaluating the received analog signal for the detection of a pulse. Once a pulse edge is detected the sample is evaluated to ensure correct pulse length timing. The sequenses of pulse are then converted to ASCII characters as defined in the following  **HF Datalink** section.

All characters, defined in Figure 4: Morse Code CW, are available for the Uplink message fields expext ***content***. The Uplink message ***content***, defined in Table 3 - HF Uplink Data Format, requires numeric transcoding. The transcoding is defined in SW-005 ICD Messaging Inteface..

## HF Uplink

The messaging between the Air Segment and Ground Segment are defined in SW-005 ICD Messaging Inteface. The following Datalink information is for reference only.

## HF Datalink

The HF Datalink simply use standard HF CW (also known as Morse) communications. The CW standard is well known, and consists of a continuous phase CW signal modulated as high or low to indicate the presence of a symbol. This mode of communication encodes ASCII characters as a series of bits with varying length and spacing. Short bits are known colloquially as "dits" and will be represented here as a "." while long bits are known colloquially as "dahs" and will be represented here as "dahs".

Transmission speed, bit length and bit spacing define completely the aspects of the signaling. Transmission speed in standard CW is measured in Words Per Minute (WPM). In our application, WPM is restricted to be precisely 12. Using this value, we can compute the dit length as 1.2s/12 and the dah length as three times this unit length. In-character spacing is equal to one dah, and between-word spacing is equal to three dahs.

CW can be decoded by collecting short and long bits separated by in-character spaces (of length single dah) where characters may be separated by recognizing a between-word space (of length three dahs). We use the well accepted international Morse Code standards, repeated here for convenience:

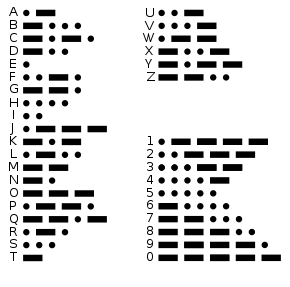


Figure : Morse Code CW

# Enhanced Morse Decoder Interfaces

The following sections define the physical and electrical interfaces between the Enhanced Morse Decoder, the Rockmite Radio and Enhanced HF Provider.

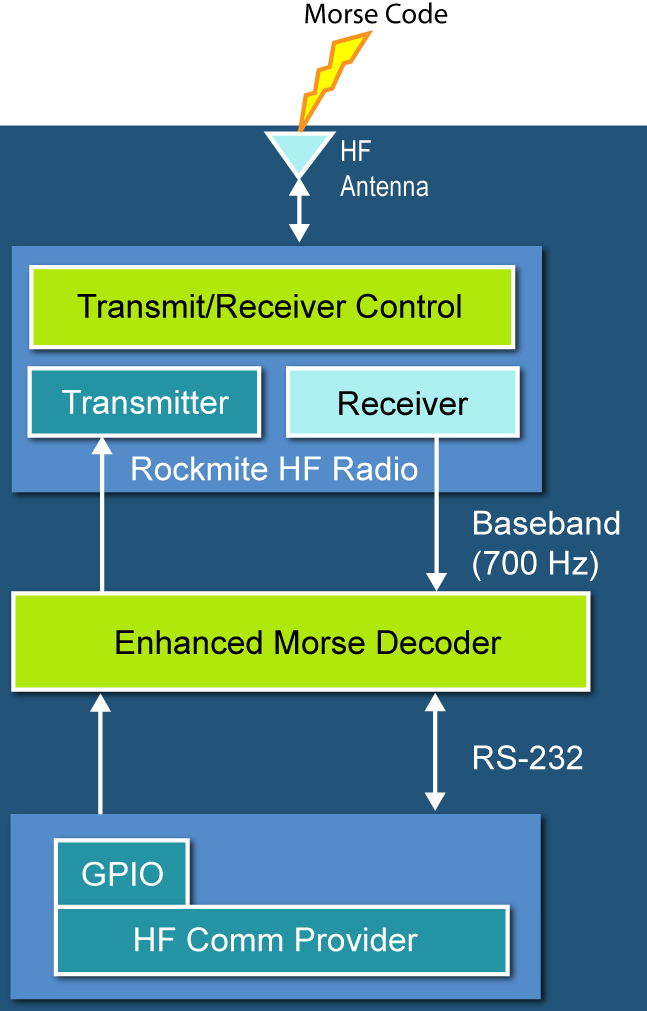


Figure : Enhanced Morse Decoder System View

## Electrical Interface

The following section defines all interfaces between the Enhanced Morse Decoder and connected subsystems.

Table : Electrical Interfaces

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Connector-Pin** | **Name** | **Subsystem**  **Interface** | **Description** | **Direction.** |
| J1-1 | HF\_IN | Rockmite HF Output | RF Analog signal  650 – 750Hz | IN |
| J1-2 | GND | Rockmite Ground | GND | GND |
| J2-1 | ??V | Power Supply | ??V Power | Power |
| J2-2 | GND | Power Supply | GND | GND |
|  |  |  |  |  |
|  |  |  |  |  |

## Mechanical Interface

<Connections, Cabling> Probably really simple. Possible to show the PWB with the reference designators showns so we know where to plug into.



Figure : EXAMPLE ONLY

# Serial Interface

The HF Uplink messages are decoded by the Enhanced Morse Decoder and transmitted to the Enhance HF Provider via a serial interface. In Table 3 we illustrate the fields to be sent via uplink. These messages will be transmitted over serial except for a transcoding of integers of the **content** message field. Note that there are no spaces between the fields when transmitting over the serial interface.

Table - Message Format

|  |  |  |
| --- | --- | --- |
| Message Element | Data Type | Notes |
| Header | UP (to Air Seg) | Constant header consisting of characters “UP” for uplink |
| Counter | Unsigned Integer  (4 characters) | Incremental message counter, monotonically increasing |
| Destination | Unsigned Byte  (1 character) | Integer destination address  3 = Air Segment 1  4 = Air Segment 2 |
| Source | Unsigned Byte  (1 character) | Integer source address  1 = Radar Tower  2 = Huron Campus |
| Relay Flag | Unsigned Byte  (1 character) | Not utilized |
| Message Type | Enum | “ap” = Abort Payload  “ab” = Abort Ballast  “db” = Drop Ballast  “st” = Request Extended Status  “rs” = Request System Status  “ps” = Periodic Status  “xs= Extended Status  “ac” = Acknowledgement  “pq” = power  “ss” = system status  “ll” = Illegal Command |
| *Content* | *Described by Presentation Layer* | |
| Terminator | X | Fixed message termination character |

Thus we can indicate a standard header to the message might look as follows:

UP QQAB C A Q ST X

Hdr message src dst fwd cmd terminator

counter

A standard header with the content field looks as follows:

UP QQAB C A Q AP QQQ QQQ X

Hdr message src dst fwd cmd Content 1 Content 2 terminator

counter

**Note** the spaces are intentionally used here to separate the fields **for** **display only**. The Air Segment does not detect spaces, and as such **space delimiting the fields is not required**.

**Note** that the field widths are fixed.

## Content

There are two classes of messages sent from the Ground Segments to the Air Segment: control messages and data request messages.

Note that we show only the content field, it is assumed that the header described in the is included.

A word of caution: All messages requiring transcoding of integers will be documented carefully. The content field supports up to three integer digits (no floating point) and the **content field must be repeated**. The repetition of the content field allows for simple error detection.

### Control Messages

*Note: the integer content of these messages must be encoded as three transcoded decimals using the translation scheme given in the Network layer. Note that the content field is always repeated.*

The following Air Segment control messages are available:

**Abort Payload (Message Type=AP)**

|  |  |  |
| --- | --- | --- |
| Content Element | Data Type | Notes |
| Abort Code | Unsigned Integer | Decimal abort confirmation code |

An example message where the abort code is all zeros looks like this:

UP QQAB C A Q AP QQQ QQQ X

**Abort Ballast (Message Type=AB)**

|  |  |  |
| --- | --- | --- |
| Content Element | Data Type | Notes |
| Abort Code | Unsigned Integer | Integer abort confirmation code |

An example message where the abort code is all zeros looks like this:

UP QQAB C A Q AB QQQ QQQ X

**Drop Ballast (Message Type=DB)**

|  |  |  |
| --- | --- | --- |
| Content Element | Data Type | Notes |
| Ballast Drop | Unsigned Integer | Integer Ballast Drop confirmation code |

An example message where the Drop Ballast code is all zeros looks like this:

UP QQAB C A Q DB QQQ QQQ X

Note that the codes for the aborts and ballast drop will be defined as mission specific to obfuscate the commands from malicious execution.

**Power Command (Message Type=PQ)**

|  |  |  |
| --- | --- | --- |
| Content Element | Data Type | Notes |
| Device | Enum | 0 = APRS  1 = BALLAST  2 = VX3R  3 = HF40  4 = STROBE  5 = GPS1  6 = GPS2  7 = BATTERY\_WARMER  8= HF30  (note, sparse numbers required, numbers not listed here will execute no-op) |
| Action | Unsigned Integer | 0 = Turn off  1 = Turn On |

An example message where the Power Command for VX3R VHF Radio code is 2 looks like this:

UP QQAB C A Q PQ QQB QQB X

### Data Request Messages

All data messages use values defined in for the message content.

**Status Request Message (Message Type=ST)**

|  |  |  |
| --- | --- | --- |
| Content Element | Data Type | Notes |
| *Empty* | | |

An example status request message looks like this:

UP QQAB C A Q ST X

**System Status Request Message (Message Type=RS)**

|  |  |  |
| --- | --- | --- |
| Content Element | Data Type | Notes |
| *Empty* | | |

An example status request message looks like this:

UP QQAB C A Q RS X