



Ulyssix UDP Frame Publisher Format

Revision A 11/29/2017

Ulyssix Technologies, Inc
7470 New Technology Way, Suite B, Frederick, MD 21703-9461
Tel: 301-846-4800 ~ Fax: 301-846-0686 ~ www.ulyssix.com

Table of Contents

Introduction	3
Packet Format	4
UDP Packet Counter	4
TAD Data Payload	4
UDP Packet Size	5
Appendix 1 – Converting BCD Time to Decimal Time	6

Introduction

This document details the UDP frame format for the Ulyssix UDP Frame Publisher. The UDP frame format is used in the Altair Software as well as the Ulyssix UDP Parameter Publishing Plug-In for DEWESoft.

The data format begins with a four-byte UDP packet counter and is followed by frame data in the Tarsus Archive Data (TAD) format. For more information about the TAD format, please see the TarsusHS Manual or the Bald Eagle Manual.

A UDP packet is published approximately every 10mS. This is the interrupt speed that the Ulyssix PCM card delivers DMA data to Altair.

Packet Format

The packet has two parts: The first is a four-byte UDP packet counter. The second is the TAD Data Payload.

The packet has a fixed size for a given Altair Configuration setup. The packet size is the four-byte UDP Packet Counter plus the TAD Data Payload size. The TAD Data Payload is determined using the Bit Rate and the Bits per Minor Frame of the telemetry setup.

UDP Packet Counter

The first four bytes in the packet are the UDP Packet Counter. This counter starts at one and increments by one for each packet sent. Each time the UDP Frame Publisher is stopped and restarted, the UDP Packet Counter resets to 1.

The UDP Packet Counter is in Little Endian:

Byte 0 is bits 0-7

Byte 1 is bits 8-15

Byte 2 is bits 16-23

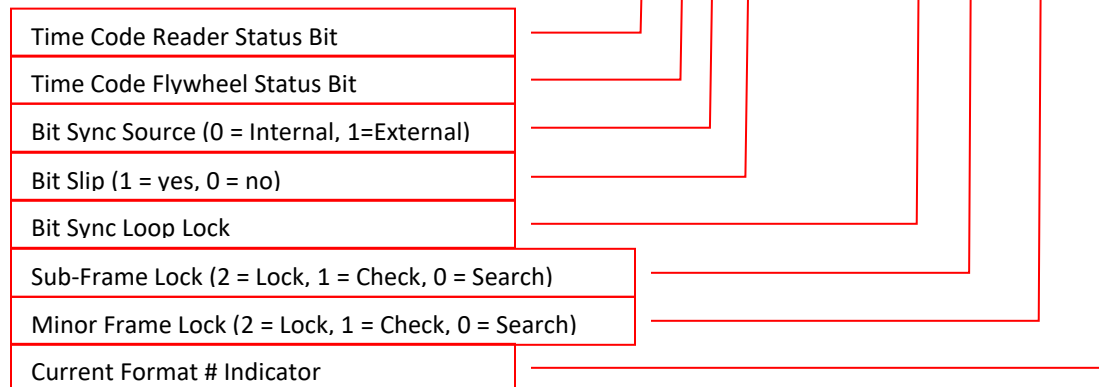
Byte 3 is bits 24-31

TAD Data Payload

The TAD Data Payload contains one or more minor frames in TAD Format. The TAD Minor Frame begins with a 12-byte TAD Minor Frame header. The header contains 8-bytes of IRIG Time data in Binary Coded Decimal (BCD) format and four bytes of Status Word data.

32 BIT WORDS	BITS																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0000				100's days				10's days				1's days				10's hours				1's hours				10's minutes				1's minutes			
1	10's seconds				1's seconds				100's msec				10's msec				1's msec				100's usec				10's usec				1's usecs			
2	Minor Frame Count																TC R Bit	TC F Bit	I/E	Slip	# Sync Errors				B Lock	SF Lock	MF Lock	FFI				

Tarsus Archive Data Header Definition



One minor frame of data follows the twelve byte TAD Minor Frame header. The TAD Minor Frame Data is in Little Endian byte order. If the Frame Sync Pattern is 0xFE6B2840, the first byte after the TAD Minor

Frame Header will be 0x40, the second byte is 0x28, the third byte is 0x6B, and the fourth byte is 0xFE. This pattern continues for the rest of the data. When trying to index 8-bit telemetry words, the pattern is byte 3, byte 2, byte 1, byte 0, byte 7, byte 6, byte 5, byte 4, byte 11, etc.

Each minor frame of data is zero padded to nearest four bytes. Please see the example below:

Examples:

256 bits per Minor Frame	44 bytes (12 bytes of Header and 32 bytes of Minor Frame Data)
264 bits per Minor Frame	48 bytes (12 bytes of Header and 36 bytes of Minor Frame Data)
272 bits per Minor Frame	48 bytes (12 bytes of Header and 36 bytes of Minor Frame Data)

The size of one TAD Minor Frame, called Packed Bytes, is the 12-byte TAD Minor Frame Header plus the TAD Minor Frame size. This can be calculated by the following using C style integer division:

```
//12-bytes for the TAD Header plus the bytes in the minor frame padded up to 4-byte increment
PackedBytes = 12 + (int)((BitsPerMinorFrame + 31) / 32) * 4;
```

The number of TAD Minor Frames per packet is called the Block Size. The Block Size is the number of minor frames that occur in 10mS. The 10mS number is based on the Windows DMA interrupt rate for the Ulyssix PCM card sending data to the Altair software. Please note that the Block Size has a minimal value of 1. If the math generates BlockSize = 0, then it should be corrected to BlockSize = 1.

```
//bit rate divided by bits in a minor frame divided by 100 interrupts per second (10mS)
BlockSize = (int)(BitRate / BitsPerMinorFrame / 100);
```

UDP Packet Size

The UDP Packet Size is calculated by the following:

```
//UDP 4-byte counter plus the bytes in a TAD minor frame multiplied by the number of minor frames in packet
UdpBytes = 4 + PackedBytes * BlockSize;
```

Appendix 1 – Converting BCD Time to Decimal Time

The following pseudo-code converts an input unsigned 64-bit integer of BCD time to

```
//Input unsigned 64-bit BCD time
UInt64 bcd;

//Decimal time for output
UInt64 usecs, msecs, secs, mins, hours, days;

//Time in microseconds since January 1 of the current year
UInt64 ustime;

//Calculate the time elements from BCD input time
usecs = (bcd >> 32 & 0xf) + (10 * ((bcd >> 36) & 0xf)) + (100 * ((bcd >> 40) & 0xf));
msecs = ((bcd >> 44) & 0xf) + (10 * ((bcd >> 48) & 0xf)) + (100 * ((bcd >> 52) & 0xf));
secs = ((bcd >> 56) & 0xf) + (10 * ((bcd >> 60) & 0xf));
mins = ((bcd >> 64) & 0xf) + (10 * ((bcd >> 68) & 0xf));
hours = ((bcd >> 72) & 0xf) + (10 * ((bcd >> 76) & 0xf));
days = ((bcd >> 80) & 0xf) + (10 * ((bcd >> 84) & 0xf)) + (100 * ((bcd >> 88) & 0xf));

//Calculate micro-seconds since January 1 of the current year
ustime = (UInt64)(days * 86400000000) + (hours * 3600000000) + (mins * 60000000) + (secs
* 1000000) + (msecs * 1000) + usecs;
```