



Ulyssix UDP Parameter Packet Format

Revision C 01/18/2017

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
Data Packet	4
Data Packet Header	4
Payload.....	5
Payload Header	5
Payload Data	5
Payload Footer	8
Appendix 1 – Converting BCD Time to Decimal Time	9

Introduction

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

The data format is configurable from the UDP Parameter Publishing graphical user interface (GUI). This document will note the configurable options in the packet description. The configurable options allow the user to determine if the packet is verbose or if the packet has the fewer bytes.

Due to the differences between DEWESoft and Altair, there are some configuration options for the packet that are not available in both. These differences are noted in the document.

Data Packet

The data packet consists of a Data Packet Header and a Payload. The Payload consists of a Payload Header, Payload Data, and Payload Footer.

Data Packet Header

Address: Bytes 0x0 with a length of at least 5-bytes.

The Data Packet Header begins at byte 0x00 with a four-byte Packet Counter in Little Endian byte order. The Least Significant Byte is in 0x00. The Most Significant Byte is in 0x03.

Byte 0x04 is a Control Byte. The bits in the Control Byte indicate the enabled features for this packet. Bit 7 is the Most Significant Bit. Bit 0 is the Least Significant Bit.

Bit 0 is the Time Tagged bit. If this bit is set, then each 32-bit floating point data value is followed by a 64-bit floating point time stamp. If the bit is clear, then the each 32-bit floating point data value is not time tagged. Clearing the Time Tagged bit reduces each sample from 96-bits to 32-bits by excluding the 64-bit floating point time stamp.

Bit 1 is the Parameter Names bit. If this bit is set, then additional data is added to the Data Packet Header. Bytes 0x05 and 0x06 are a 16-bit integer that is Parameter Names String Length. The Parameter Names String Length is the number of characters, including delimiters, in the string of Parameter Names. The Parameter Name String starts at byte 0x07. The Parameter Names String is the concatenation of the Parameter Names delimited with ASCII value 0x1F. This ASCII value 0x1F is the Unit Separator. This ASCII value will not conflict with the characters allowed in a DeweSoft Parameter Name. There is not a delimiter at the end of the string. If the Parameter Names bit is clear, then byte 0x04 is the last byte in the Packet Header and the Payload starts in byte 0x05.

Start	End	Length	Type	Name	Value
0x0	0x3	4	32-bit Integer	Packet Counter	
0x4		Bit 0	Bit	Time Tagged	0 - No Time Data 1 - Time included in Payload
0x4		Bit 1	Bit	Parameter Names	0 - No Param Names 1 - Param Names in Packet Header
0x4		Bit 2	Bit	Data Type (Altair Only)	0 - Floating Point Data and Time 1 - Integer Data and Time
0x4		Bit 3	Bit	Unused	
0x4		Bit 4	Bit	Unused	
0x4		Bit 5	Bit	Unused	
0x4		Bit 6	Bit	Unused	
0x4		Bit 7	Bit	Unused	
0x5	0x6	2	16-bit Integer	Packet Names Length	Optional: depends of Status Bit 1 Little Endian
0x7	Variable	Variable	String	Packet Names	Optional: depends of Status Bit 1 ASCII 0x1F is the delimiter

Payload

The Payload contains the Payload Header, Payload Data, and Payload Footer. The Payload begins after the Data Packet Header. The end of the Data Packet Header depends on the Control Byte. Please see the section Data Header for more details.

Payload Header

Address: Starts at the end of the Packet Header and is 36-bytes in length. All multiple byte words are in Little Endian byte order. The example below assumes that the Control Byte Parameter Names bit is clear.

The 36-byte Payload Header contains a Start String to denote the beginning of the packet. It also contains the Payload Size, Packet Type, Number of Samples in the Packet, Total Samples Acquired, and the Absolute Time. Payload Size is the number of bytes in the payload minus 8-bytes for the Start Packet String and 8-bytes for the End Packet String (Payload Footer). The Packet Type is always 0x00 for a Data Packet. The Samples in the Packet is the total number of samples for all parameters included in the packet. Total Samples Acquired is the total number of samples that have been acquired since the measurement started. Packet Time Stamp is an unsigned 64-bit integer representing the day of year, hour, minute, second, millisecond, and microsecond of the first data point in the packet.

Start	End	Length	Type	Name	Value
0x05	0x0C	8		Start Packet String	0x0706050403020100
0x0D	0x10	4	32-bit Integer	Payload Size	
0x11	0x14	4	32-bit Integer	Packet Type	0 for Data Packet
0x15	0x18	4	32-bit Integer	Samples in Payload	
0x19	0x20	8	64-bit Integer	Total Samples Acquired	
0x21	0x28	8	64-bit Unsigned Integer	Packet Time Stamp	Time of the first data point in the packet. Time is in BCD format.

*this example is for a Control Byte Parameter Names Bit is clear.

The Packet Time Stamp is in Binary Coded Decimal. This makes the Packet Time Stamp human readable in hexadecimal representation of the of the packet. The Packet Time Stamp is in the same format as the time stamp in the Ulyssix TAD Minor Frame Header. The table below shows the value for each nibble in the 64-bit integer.

Upper 32-Bits

15	14	13	12	11	10	9	8
0x0	100's Days	10's Days	1's Days	10's Hours	1's Hours	10's Min	1's Min

Lower 32-Bits

7	6	5	4	3	2	1	0
10's Sec	1's Sec	100's mSec	10's mSec	1's mSec	100's uSec	10's uSec	1's uSec

Payload Data

Address: Starts after the end of the Payload Header. If the Parameter Names bit is clear, then the Payload Data starts in byte 0x29 and is variable in length depending on the number of parameters and the samples for each parameter. The Payload data is in Little Endian byte order.

The Payload Data contains the data for each parameter selected in the GUI. The parameters in the Payload are in the same order as the parameter names appear in the software and the same order as the Parameters String in the Packet Header, if enabled. The time between packets for Altair is approximately 10mS. The time between packets for DeweSoft is controlled by the DeweSoft setting Acquisition Update Rate. Typical values are 33mS to 50mS. If the parameter's sample rate is faster than the packet rate, then the packet will contain multiple samples from the parameter. If the parameter's sample rate is slower than packet rate, then the Parameter Number of Samples is 0x0 and no samples are included. A Data Packet will never be sent if all the Parameters have zero samples.

Each parameter begins with the Number of Samples as a 32-bit integer. Next, is the data as a 32-bit data point and the optional 64-bit time stamp. The time stamp can be omitted by a check box in the Plug-In GUI settings. Bit 0 in the Control Byte indicates if the data is time stamped.

DEWESoft does not have the option for Data Type in the GUI. The 32-bit data point is always a single precision floating point number. The time stamp is always a 64-bit double precision floating point number that represents the seconds since 12:00am January 1st of the current year.

In Altair there are two options. The DataType bit = 0 is the same as DEWESoft. The 32-bit data point is always a single precision floating point number. The time stamp is always a 64-bit double precision floating point number that represents the seconds since 12:00am January 1st of the current year. The Data Type bit = 1 uses raw binary values instead of processed floating point. The data point is a signed 64-bit integer representing the raw bits from the data parameter. The time stamp is an unsigned 64-bit integer that represents the micro-seconds since 12:00am January 1st of the current year.

Start	End	Length	Type	Name	Value
Offset	+3	4	32-bit Integer	Number of Samples	
+4	+7	4	Single Float	Data	
+8	+15	8	Double Float	Time	User selectable
* Data and Time pairs are repeated for the Number of Samples					

Example 1:

There are two parameters in the packet. The first parameter has four samples. The second parameter has two samples. Time tagging is enabled.

Start	End	Length	Type	Name	Value
0x29	0x2C	4	32-bit Integer	Param 1 Num Samples	4
0x2D	0x30	4	Single Float	P1 Samp 1 Data	
0x31	0x38	8	Double Float	P1 Samp 1 Time	
0x39	0x3C	4	Single Float	P1 Samp 2 Data	
0x3D	0x44	8	Double Float	P1 Samp 2 Time	
0x45	0x48	4	Single Float	P1 Samp 3 Data	
0x49	0x50	8	Double Float	P1 Samp 3 Time	
0x51	0x54	4	Single Float	P1 Samp 4 Data	
0x55	0x5C	8	Double Float	P1 Samp 4 Time	
0x5D	0x60	4	32-bit Integer	Param 2 Num Samples	2
0x61	0x64	4	Single Float	P2 Samp 1 Data	
0x65	0x6C	8	Double Float	P2 Samp 1 Time	
0x6D	0x70	4	Single Float	P2 Samp 2 Data	
0x71	0x78	8	Double Float	P2 Samp 2 Time	

Example 2:

There are three parameters in the packet. The first parameter has two samples. The second parameter has one sample. The third parameter has one sample. Time tagging is disabled.

Start	End	Length	Type	Name	Value
0x29	0x2C	4	32-bit Integer	Param 1 Num Samples	2
0x2D	0x30	4	Single Float	P1 Samp 1 Data	
0x31	0x34	4	Single Float	P1 Samp 2 Data	
0x35	0x38	4	32-bit Integer	Param 2 Num Samples	1
0x39	0x3C	4	Single Float	P1 Samp 1 Data	
0x3D	0x40	4	32-bit Integer	Param 2 Num Samples	1
0x41	0x44	4	Single Float	P1 Samp 1 Data	

Example 3:

There are three parameters in the packet. The first parameter has zero samples. The second parameter has four samples. The third parameter has two samples. Time tagging is enabled.

Start	End	Length	Type	Name	Value
0x28	0x2C	4	32-bit Integer	Param 1 Num Samples	0
0x29	0x2C	4	32-bit Integer	Param 2 Num Samples	4
0x2D	0x30	4	Single Float	P1 Samp 1 Data	
0x31	0x38	8	Double Float	P1 Samp 1 Time	
0x39	0x3C	4	Single Float	P1 Samp 2 Data	
0x3D	0x44	8	Double Float	P1 Samp 2 Time	
0x45	0x48	4	Single Float	P1 Samp 3 Data	
0x49	0x50	8	Double Float	P1 Samp 3 Time	
0x51	0x54	4	Single Float	P1 Samp 4 Data	
0x55	0x5C	8	Double Float	P1 Samp 4 Time	
0x5D	0x60	4	32-bit Integer	Param 2 Num Samples	2
0x61	0x64	4	Single Float	P2 Samp 1 Data	
0x65	0x6C	8	Double Float	P2 Samp 1 Time	
0x6D	0x70	4	Single Float	P2 Samp 2 Data	
0x71	0x78	8	Double Float	P2 Samp 2 Time	

Payload Footer

Address: The Payload Footer begins after the last byte of the Payload Data. The Payload Footer is 8-bytes long.

The Payload Footer is the inverse of the Start Packet String.

Start	End	Length	Type	Name	Value
Offset	+8	8		End Packet String	0x0001020304050607

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;
```