Chapter 40

IENA and iNET-X packet payload formats

TEC/NOT/067

This paper discusses the following topics:

40.1 Overview

This paper describes the packet header structures and payload structures for both IENA and iNET-X application layer packetization protocols.

NOTE: There is significant commonality regarding the creation and placement of the payload in both iNET-X and IENA application layer packets, the sections describing the IENA packetization formats are for illustrative purposes since the payload specific details have been discussed in the corresponding iNET-X section.

NOTE: To better understand this paper, see "40.6 Recommended reading" on page 22.

NOTE: The notation used in this document is the standard bit ordering and numbering convention used in networking known as *standard big-endian network byte order*. That is, the most significant octet is transmitted first; the left-most bit of the entire field is the most significant bit.

40.2 Transmitting data

The backplane controller packs the acquired data from each of the user-modules into the packet payload. The controller then encapsulates the payload of acquired data as either iNET-X or IENA packets which are subsequently encapsulated and transmitted as UDP/IP packets. The following two figures illustrate the complete UDP/IP-encapsulated iNET-X and IENA packet structures respectively.

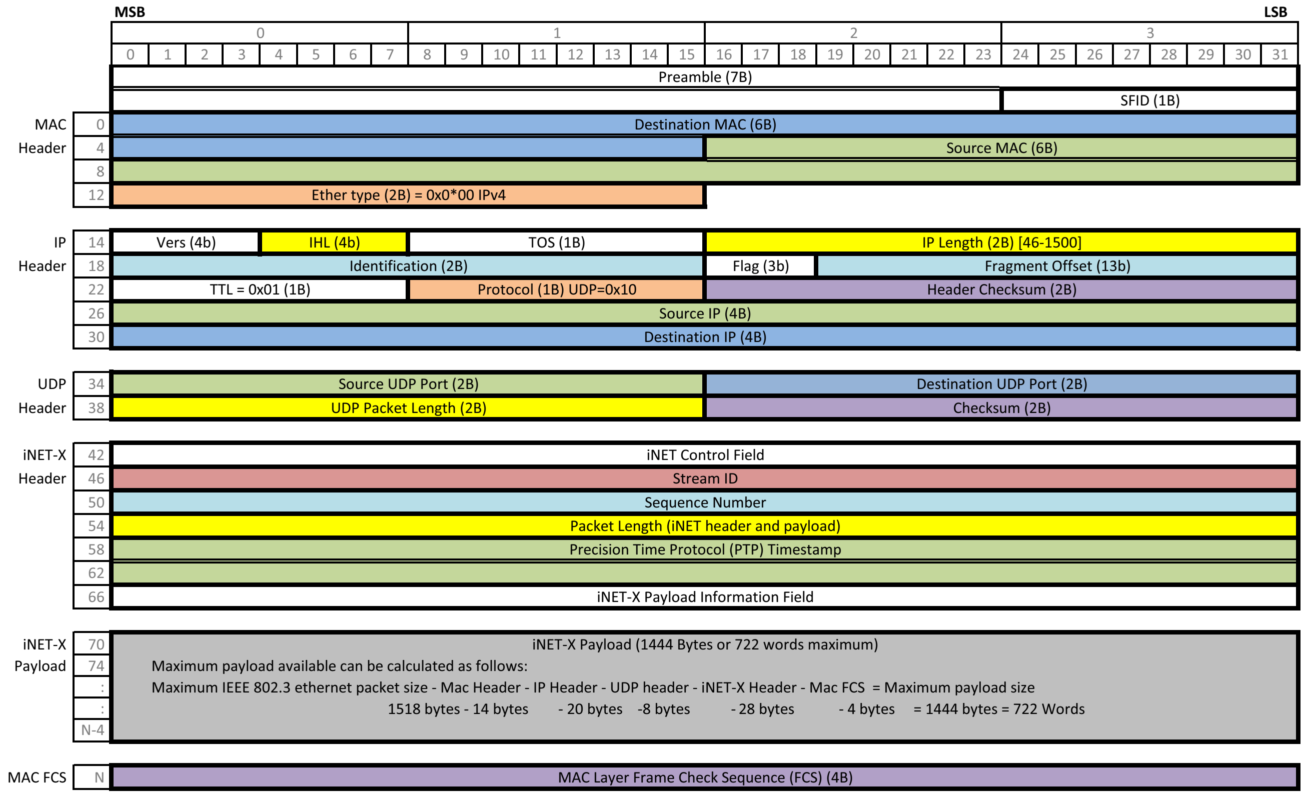

Figure 40-1: iNET-X packet structure

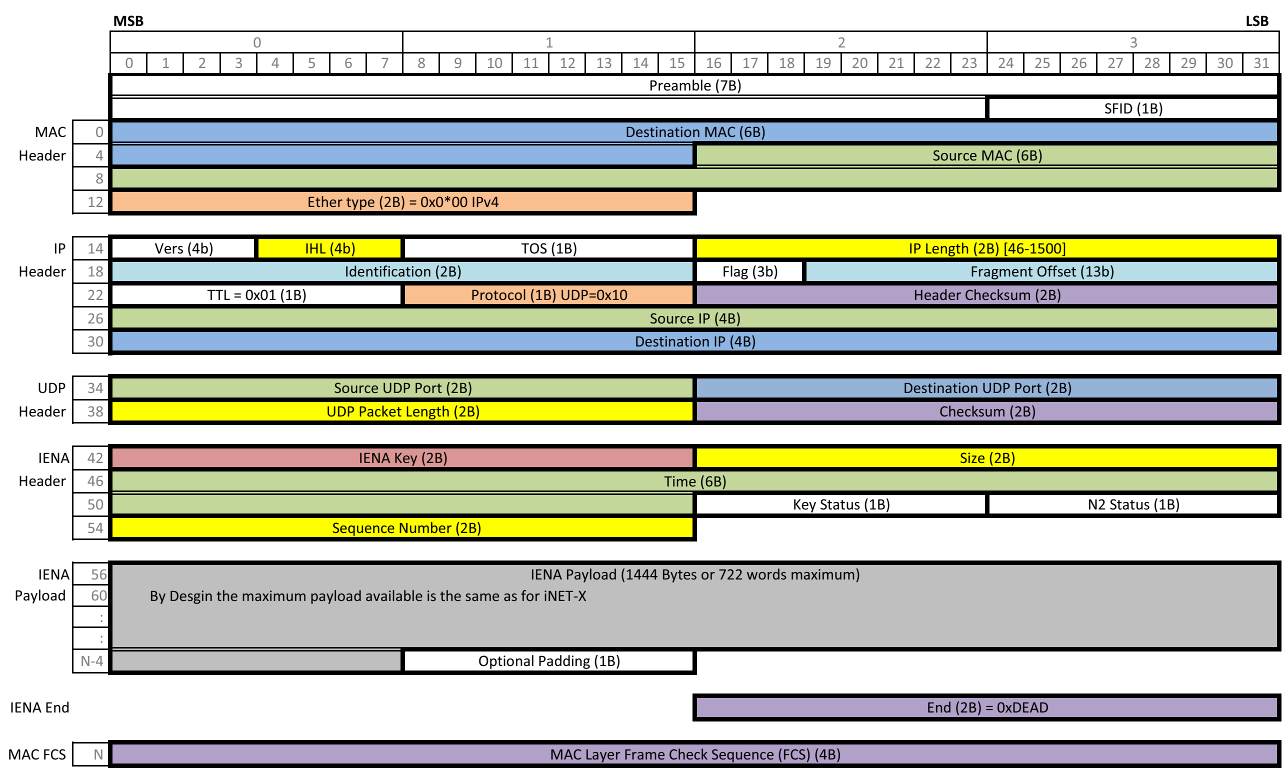

Figure 40-2: IENA packet structure

40.2.1 Packetization rules and recommendations

The following table outlines the packetization rules and recommendations for both IENA and iNET-X packet structures.

Table 40-1: iNET-X and IENA packetization rules and recommendations

Recommendation	iNET-X	IENA
Each packet stream carries a Stream ID.	iNET-X Stream ID (32 bits).	IENA Key (16 bits).
A Data Acquisition Unit (DAU) may transmit more than one packet stream.	Recommended.	Recommended.
It is recommended that the packet stream is transmitted as multicast.	Recommended.	Recommended.
The Stream ID/IENA Key is system-wide unique.	Recommended.	Recommended.
All packets for a given stream contain one type of parameter (that is, messages and parameters are not mixed in the one packet payload).	Recommended.	Recommended.
Supported payload structures.	iNET-X defines the following payload structures: Placed Block Bit-aligned Parser-aligned Event	IENA defines three basic payload structures however only the Positional type is supported. Other types not currently supported: Message Standard
Parameter placement rules.	No restrictions, but recommended as having contiguously placed samples. This applies to Placed payload structures.	Positional type payloads must adhere to strict parameter sample interleaving rules.
The placement of parameters is fully described by XidML.	Recommended.	Recommended.
Fragmentation support: Large payloads exceeding the Maximum Transmission Unit (MTU) of 1500 bytes incurs an onerous packetization and reassembly delay thereby increasing the end-to-end latency.	Fragmentation is not supported by the INET-X standard.	IP layer fragmentation is allowed as part of the IENA standard but it is not recommended.
Packetization latency: The latency from sensor to display should not exceed real-time limits. A typical value of 50ms is recommended.	Recommended.	Recommended.

40.3 iNET-X packet header

iNET-X packets use the standard iNET application layer packet structure as shown in the following figure and are therefore fully compliant and compatible with iNET systems. iNET-X packets have an additional 4-byte extension field, called the iNET-X Payload Information field, appended directly following the standard iNET header.

NOTE: For more information on iNET, go to http://www.inetprogram.org/default.aspx.

The iNET-X Payload Information field contains Curtiss-Wright-specific metadata to facilitate decoding and decommutation of the payload.

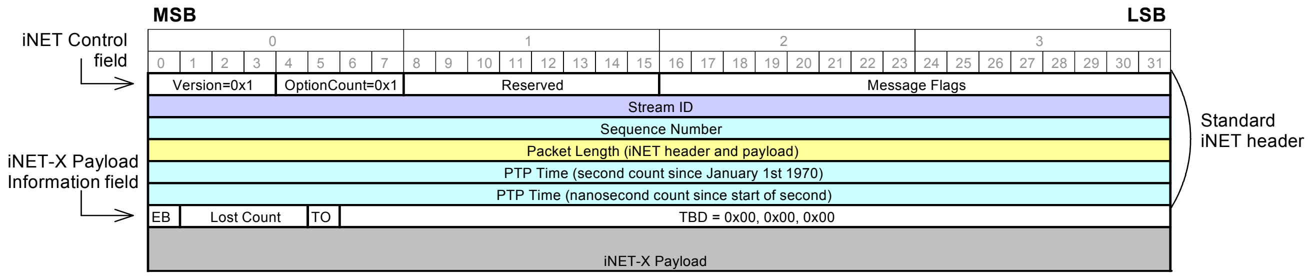

Figure 40-3: iNET packet header

The iNET packet header contains the following elements:

- iNET Control field (32 bits):
  - Bits [0:3] Version: identifies the version of the iNET application layer protocol. The version supported is Version1.
  - Bits [4:7] OptionCount: specifies the number of 32-bit words in the application defined extension field. The value in this field is 0x1, since iNET-X packets comprise a single 32-bit application defined extension field, called the iNET-X Payload Information field, following the standard iNET header.
  - Bits [8:15] Reserved: to be defined and is reserved for future use.
  - Bits [16:31] Message Flags: contains the following fields:
    - Message Fragmentation flags.
    - Message Simulated Data flag.
    - Message Time Synchronization flag.
    - Message Health flag.
    - End of Data flag.

NOTE: These Message flags are not currently supported in iNET-X.

- Stream ID (32 bits): points to a unique (system wide) metadata package description in the XidML. The package description describes the format, structure, and data contained in the iNET-X packet payload.
- Sequence Number (32 bits): increments by one for each packet generated and transmitted with a given Stream ID.
  - A sequence number value of 0 occurs when:
    - The DAU controller is powered on.
    - After the DAU has been programmed.
    - The sequence number has incremented to the maximum value and has wrapped around back to 0.
- Packet Length (32 bits): is the length of the complete iNET-X packet in bytes including the iNET header, iNET-X Payload Information field, the payload, and any padding in the packet. Padding is used when the length of the iNET-X packet does not fall on a 32-bit boundary.
- PTP Time (64 bits): is the timestamp associated with the oldest unit of data in the payload. In the case of analog data, the timestamp relates to the earliest sample contained in the payload. For bus monitor data, the timestamp relates to the first and earliest bit or message captured on the bus. The time format used is unsigned Precision Time Protocol (PTP) version 1 format using PTP epoch where:
  - Bits [0:31] Time = second count since January 1st 1970.
  - Bits [32:63] Time = nanosecond count since start of second.
- iNET-X Payload Information field (32 bits): contains Curtiss-Wright-specific packet metadata. The iNET-X Payload Information field comprises the following fields:
  - EB (Bit 0): Error Bit: the acquired data is validated for parity and bit errors. The EB is used to indicate the integrity of the payload.
    - EB = 1 if there is one or more errors associated with the data (such as bits or messages) contained in the payload.
    - EB = 0 if there are no errors associated with the payload.
    - LostCount (Bits [1:4]): when the First In First Out (FIFO) or buffer is filled with acquired data at a faster rate than iNET-X packets are being generated, data is lost. When this occurs, the Drop Count field indicates the number of iNET-X packet payloads lost due to the buffer becoming saturated.
  - TO (Bit 5): Timeout: iNET-X supports an Aperiodic Transmission (No Transmit When Empty) function to make efficient use of network bandwidth and recording media resources. However, on asynchronous busses during periods of low data rates, the buffer or FIFO may fill slowly. To ensure iNET-X packets are periodically generated to facilitate real-time analysis and processing, during periods of low data rates there is a timeout associated with the data, thereby guaranteeing that a packet is generated.
    - TO = 1 if the iNET-X packet is generated and transmitted as the result of a timeout occurring.

- TO = 0 if the iNET-X packet is generated as a result of the FIFO or buffer being filled to a predefined maximum size.
- Bits [6:31]: Reserved: to be defined and is reserved for future use.

NOTE: When the timeout is reduced from the default value, a higher packetizer rate may be required to ensure that each packet can timeout and still be read.

40.3.1 iNET-X payload formats

iNET-X defines the following payload formats which are suitable for a variety of acquired data types:

- Bit-aligned
- Placed
- Block
- Parser-aligned
- Event

The iNET-X payload format implemented is a feature of the appropriate user module.

The key attributes of iNET-X packets are summarized in the following table.

Table 40-2: iNET-X payload structure overview

iNET-X packet payload type	Examples	Example products	Max payload length	PTP timestamp	Notes and metadata elements
Bit-aligned	Continuously Variable Slope Delta (CVSD) audio	SSR/CHS/00X built-in VDC	N x 4 bytes, not exceeding 1,444 bytes	Time tag of first bit in the payload	• iNET-X packet type: bit-aligned. • Timeout: if applicable, for example 10ms. Data type: string representation of the data type, for example CVSD audio. • Target packet size in bytes, for example N x 32-bit.
Placed	Analog samples, synchronous PCM, and any periodic acquisition	SSR/ADC/126	Fixed and constant length, not exceeding 1,444 bytes	Time tag of oldest sample in the payload	• iNET-X packet type: placed. • For each parameter: - Format: for example offset binary, 2's complement and so on. - Range: maximum, minimum with respect to SI units. • Parameter placement should always start on a 16-bit boundary. • If more than 16-bit, that is 24-bit, must be extended by N x 16-bit.
Block	MPEG-2 transport stream video	SSR/VID/106	N x block length in bytes, not exceeding 1,444 bytes	Time tag of oldest block in the payload	• iNET-X packet type: block. • Timeout: if applicable, for example 10ms. • Data type: string representation of the data type, for example video. • Bytes per block, for example 188. • Target packet size: integral number of blocks, for example 7.
Parser-aligned	MIL-STD-1553, ARINC-429, PCM	SSR/MBM/101 SSR/ABM/102 SSR/ABM/103 SSR/PBM/104 SSR/CBM/105	Variable, not exceeding 1,444 bytes	Time tag of first message in the payload	• iNET-X packet type: Parser-aligned. • Timeout: if applicable, for example 10ms. • Data type: string representation of the data type, for example MIL-STD-1553, ARINC-429, PCM. • Target packet size in bytes.

Table 40-2: iNET-X payload structure overview

iNET-X packet payload type	Examples	Example products	Max payload length	PTP timestamp	Notes and metadata elements
Event, Status, Error	SSR locally or remotely generated pilot event markers and traps	SSR/CHS/00X NET/REC/00X KAD/BIT/102	Variable, not exceeding 1,444 bytes	Time tag of event	• iNET-X packet type: event. • For each event type, an enumeration and string.

40.3.2 iNET-X placed packet format

Placed iNET-X packet structures are designed to cater for tailored packet payloads where parameters are placed at specific locations in the payload as shown in the following figure. This approach is analogous to the placement of parameters in a PCM frame. The format and structure of these placed packets must be described in full, in XidML, for individual parameters and samples to be located and extracted from the packet.

The placement of parameters and/or samples in the iNET-X payload is not restricted or constrained by placement rules. However, the locations of the placed parameters must be fixed and constant for each iNET-X packet generated, with a given stream ID for a given configuration.

Analog samples for multiple channels may be placed contiguously in the payload. Where there are multiple channels of analog data, the number of samples for each analog channel contained in the payload must be an integral number of the lowest common denominator of samples. For example, consider Channel 1 has N samples (lowest common denominator); every other channel must contain Z x N samples, where Z ≤ 1.

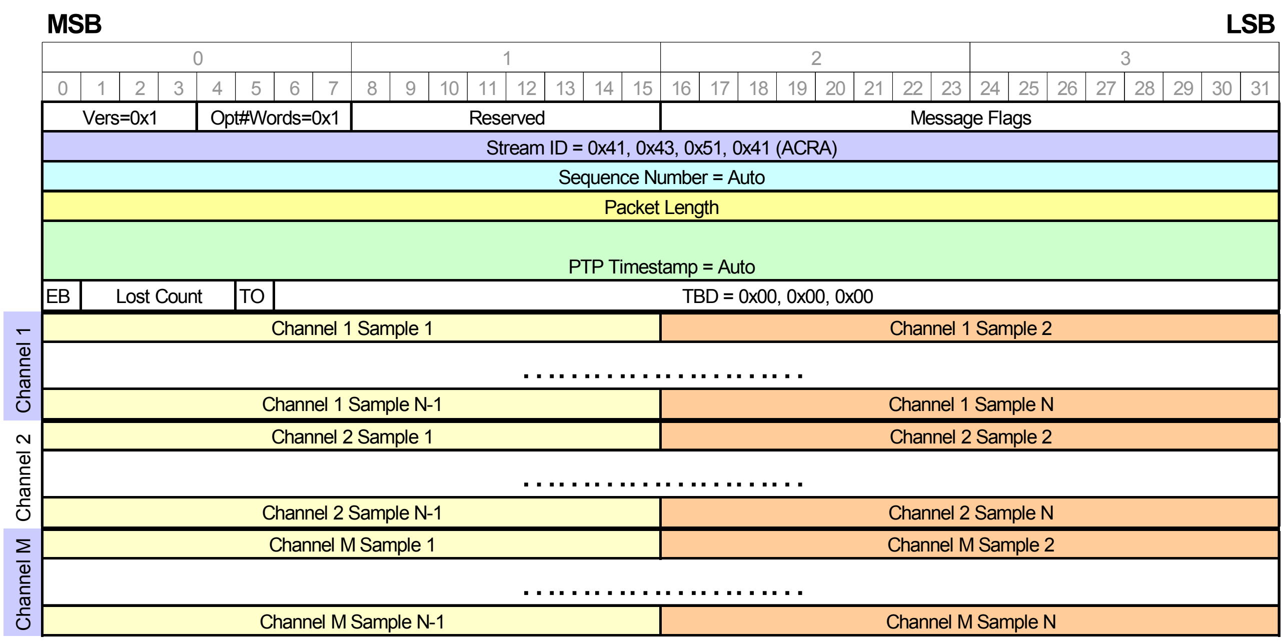

Figure 40-4: Example of iNET-X payload packet format

40.3.2.1 Example - iNET-X placed packet for analog data

Consider an analog module with four input analog channels. The analog parameters are sampled at the start of an even second and at even intervals of time thereafter. The analog data is packetized in a placed iNET-X analog packet and may contain data for one or more analog channels, where each channel may be sampled at equal or differing rates. Samples for each channel are placed contiguously in the analog packet.

Consider sampling four channel parameters at different rates (as shown in the following table) and packetized in a placed iNET-X packet called MyAnalogPacket.

Table 40-3: Analog module sampling configuration and packetization in MyAnalogPacket

Parameter name	Rate (ksps)	Occurrences per packet	First channel-sample byte offset from start of Ethernet frame
MyChannel #1	2	100	68
MyChannel #2	4	200	268
MyChannel #3	1	50	668
MyChannel #4	1	50	768

MyAnalogPacket contains a total of 400 samples per packet and packets are generated 20 times per second, as shown in the following figure.

The placed iNET-X packet, MyAnalogPacket, contains contiguously placed samples for each of the four channels. The first sample of the analog data is located at an offset of 68 bytes from the start of the Ethernet frame. The PTP timestamp in the iNET-X packet header relates to the sampling instant of the first (earliest) sample contained in the payload. The sampling instant of subsequent samples is calculated by adding the channel-sampling interval to the PTP timestamp. The format of the generated analog packets is stored in the XidML file.

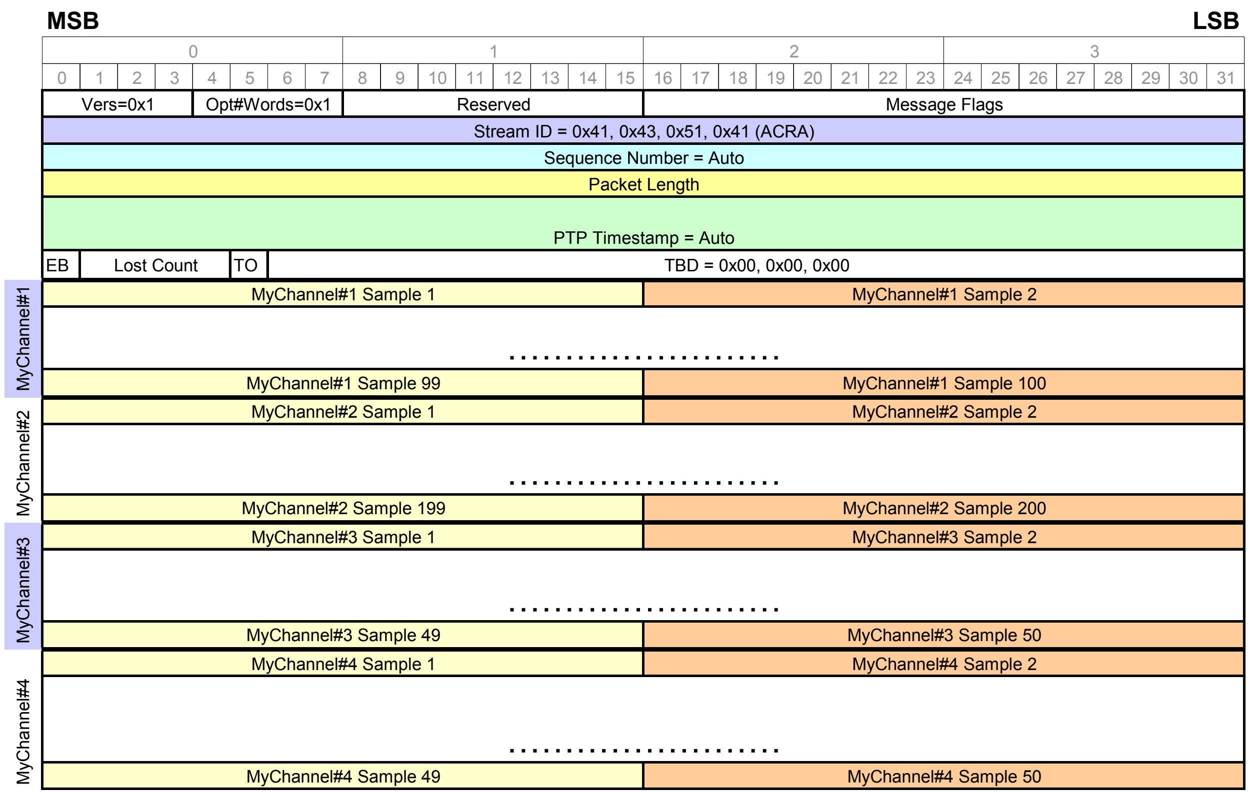

Figure 40-5: 400 analog samples for four channels at different rates in a single iNET-X placed packet

The analog packet transmission properties are described in the following table.

Table 40-4: MyAnalog packet transmission properties

Transmission property	Value
Analog payload size (bytes)	800 (400 samples x 2 bytes)

Table 40-4: MyAnalog packet transmission properties

Transmission property	Value
Total Ethernet frame length (bytes)	874 MAC header 14 bytes + IP 20 bytes + UDP 8 bytes + iNET-X 28 bytes + analog data + MAC Frame Check Sequence (FCS) 4 bytes
Packet rate (packets per second)	20
Total number of parameter samples	400
Total bit-rate (kbps)	139.8

40.3.3 iNET-X bit-aligned packet format

The iNET-X bit-aligned packet format is suited to bit-aligned data streams such as audio CVSD (as opposed to word-aligned or frame-aligned data types). Moreover, this raw packing structure can be applied to any data type stored in a FIFO. The PTP timestamp in the iNET-X packet header refers to the acquisition instant of the first bit placed in an empty FIFO.

40.3.3.1 Example - iNET-X bit-aligned packet format for synchronous CVSD audio data

CVSD encodes voice audio at 1 bit per sample; audio sampled at 64kHz is encoded at 64kbps. The application payload format for CVSD audio data is the same as that used for PCM bit-stream data.

The CVSD packet transmission properties are described in the following table and indicate the packetization structure of 8,000 CVSD, 1-bit, audio samples in a single packet to maximize the storage efficiency. The structure of the iNET-X MyCVSD audio packet is shown in the following figure.

Table 40-5: MyCVSD audio packet transmission properties

Transmission property	Value
CVSD packet payload size (bytes)	1,000 (8,000 x 1-bit samples)
Total Ethernet frame length (bytes)	1,074 MAC header 14 bytes + IP 20 bytes + UDP 8 bytes + iNET-X 28 bytes + CVSD audio bit-stream data + MAC FCS 4 bytes
Packet rate (packets per second)	8
Total number of bytes per packet	1,000
Total bit-rate (kbps)	68.7

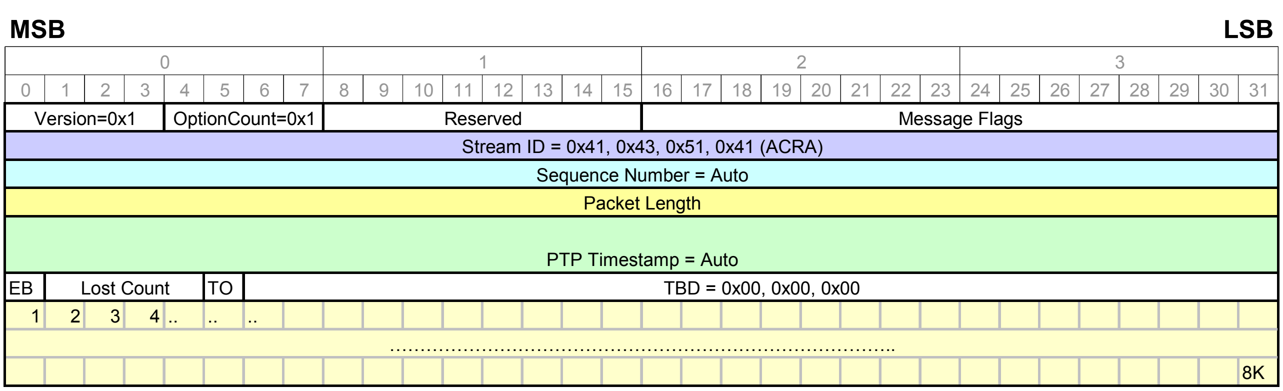

Figure 40-6: MyCVSD bit-aligned iNET-X packet

40.3.4 iNET-X block packet format

The iNET-X block packet format is suited to constant bit-rate data types such as MPEG-2 transport stream video data. An iNET-X block packet consists of an integral number of blocks of data, where each block consists of M words. For example, using MPEG-2 transport stream data, a maximum of seven transport stream chunks can be stored in the payload as shown in the following figure. The PTP timestamp in the iNET-X packet header refers to the acquisition instant of the first block of data in the payload.

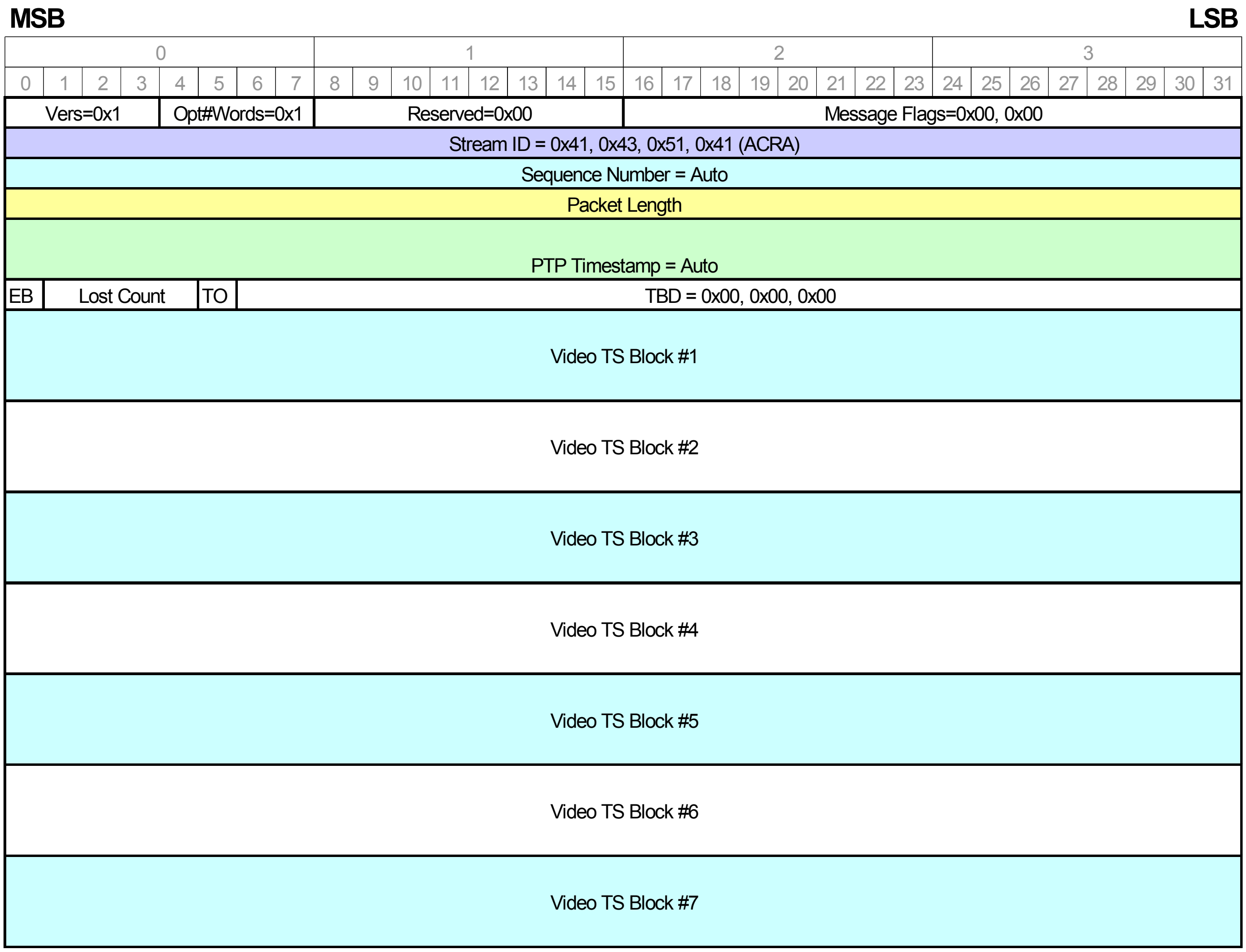

Figure 40-7: iNET-X block packet structure

40.3.4.1 Example - iNET-X block packet format for MPEG-2 transport stream video

Consider an MPEG-4 compression module that embeds the MPEG-4 video in an MPEG-2 transport stream. An MPEG-2 transport stream partitions the MPEG-4 video data into a continuous flow of transport stream chunks where each chunk is 188 bytes. iNET-X block video packets contain an integral number of MPEG-2 transport stream chunks.

If the target encoding bit-rate is 1Mbps, seven transport stream chunks can be stored in each Ethernet frame (7 x 188 bytes = 1,316 bytes of video data in the payload). To achieve a bit-rate of approximately 1Mbps, then 95 packets per second are required to capture the video data. However, to ensure no video data is lost in the event that the video bit-rate exceeds 1Mbps, the video packet storage/transmission rate is rounded up to 100 packets per second.

The video packet transmission properties are described in the following table. The video packet structure is shown in the following figure.

Table 40-6: MyVideo iNET-X block packet transmission properties

Transmission property	Value
Video packet payload size (bytes)	1,316 (7 x 188-byte transport stream chunks)
Total Ethernet frame length (bytes)	1,390 (MAC header 14 bytes + IP 20 bytes + UDP 8 bytes + iNET-X 28 bytes + video data + MAC FCS 4 bytes)
Packet rate (packets per second)	95, rounded up to 100
Total number of transport stream chunks per packet	7
Total bit-rate (Mbps)	1.1

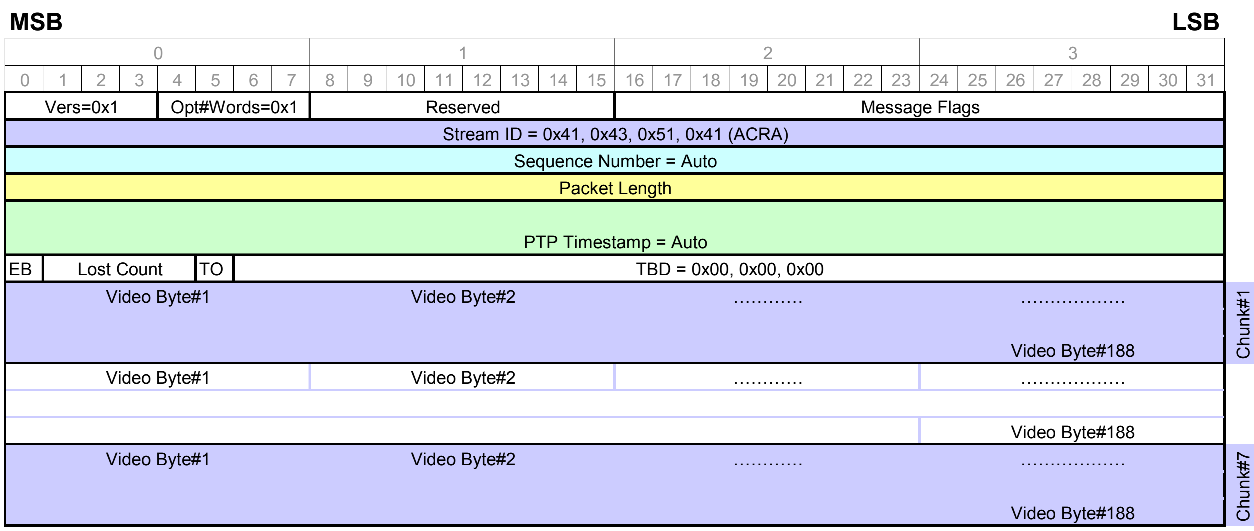

Figure 40-8: MyVideo iNET-X block video packet

40.3.5 iNET-X parser-aligned packet format

There is a diverse range of avionic bus technologies for which traffic may be captured, for example, MIL-STD-1553, PCM, or ARINC-429. The generalized iNET-X payload structure for parser-aligned packets is shown in the following figure.

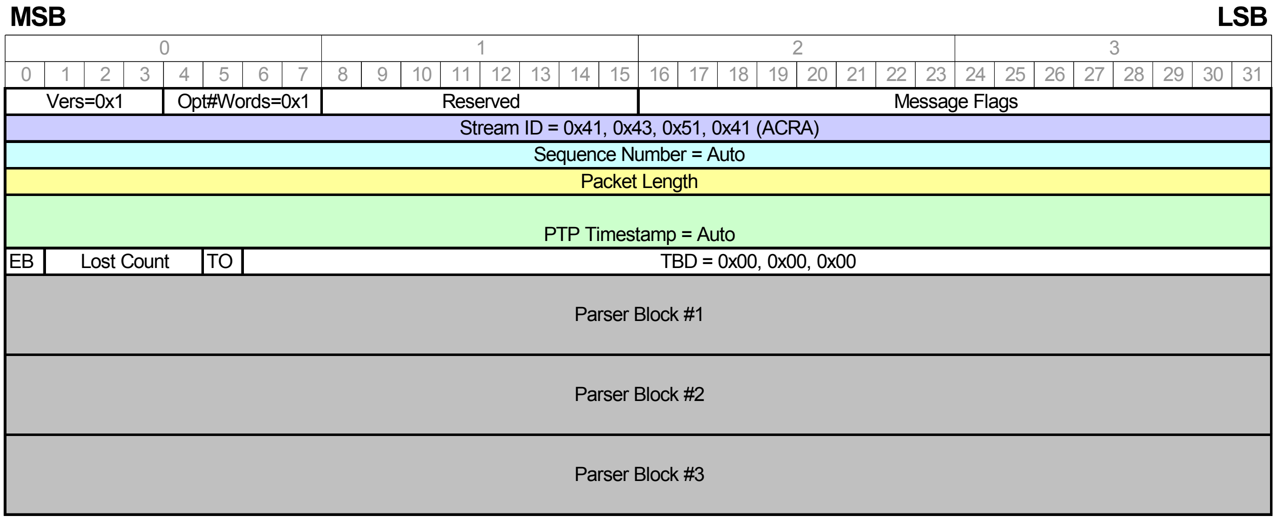

Figure 40-9: Generalized parser-aligned iNET-X packet

As messages are captured on the bus, they are formatted in a parser block. Each parser block begins with a 4-byte parser information word, followed by a 4-byte elapsed time tag and the message data shown in the following figure.

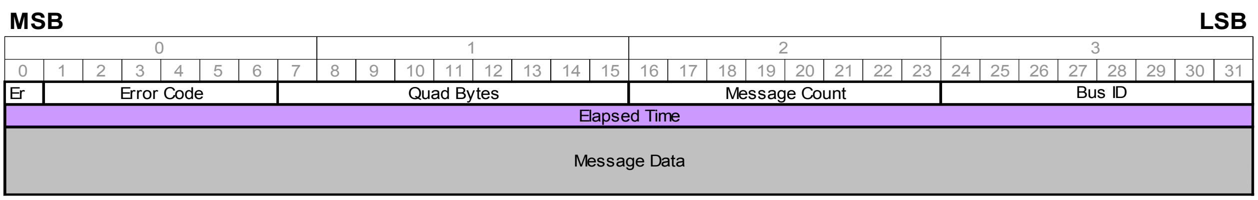

Figure 40-10: iNET-X parser block

A parser block consists of the following fields:

- Parser information word (4 bytes): metadata providing information about the health and status of the message.
  - Er (Bit 0): indicates that an error occurred.
  - Bits [1:6]: TBD.
  - Quad bytes (Bits [7:15]): number of quad bytes. This relates to the length of the parser information word, elapsed time tag, and the message data and padding (N x 4 bytes). For example, a 4-byte message captured from a given bus has a quad-byte value of 3, that is 12 bytes that includes 4 bytes parser information word, 4 bytes elapsed time, and 4 bytes bus message data.
  - Message count (Bits [16:23]): message counter. This is a message counter that relates to the messages contained in the payload. The message counter increments for each message contained in the packet payload and continues to increment across consecutive packets. The message counter resets and wraps around to 0 once it has reached the maximum message count of 0xFF.
  - Bus ID (Bits [24:31]): bus number.
- Elapsed time (4 bytes): time tag as an unsigned offset in nanoseconds that is added to the base PTP timestamp in the iNET-X header.
- Message data (N x 4 bytes): captured bus traffic, padded if necessary to end on 4-byte boundary.

40.3.5.1 Example iNET-X parser-aligned packet format for ARINC-429

Consider an eight-channel ARINC-429 bus monitor where traffic captured on each of the ARINC-429 busses is placed in an iNET-X parser-aligned packet (see the following figure) where each bus has its own unique Stream ID.

As the ARINC messages arrive they are tagged with a 4-byte parser information word and a 4-byte elapsed time word, followed by the 4-byte ARINC message. The parser information word identifies properties of the ARINC-429 message (such as the message counter and the ID of the bus on which the message was received) and marks the health of the message using an error bit and error code. The PTP timestamp in the iNET-X packet header is fixed when the packet is opened for writing and is used as the base timestamp for the whole packet. The PTP timestamp for each ARINC message in the packet can be calculated by adding the elapsed time to this base timestamp. Directly following the elapsed time field is the ARINC-429 message.

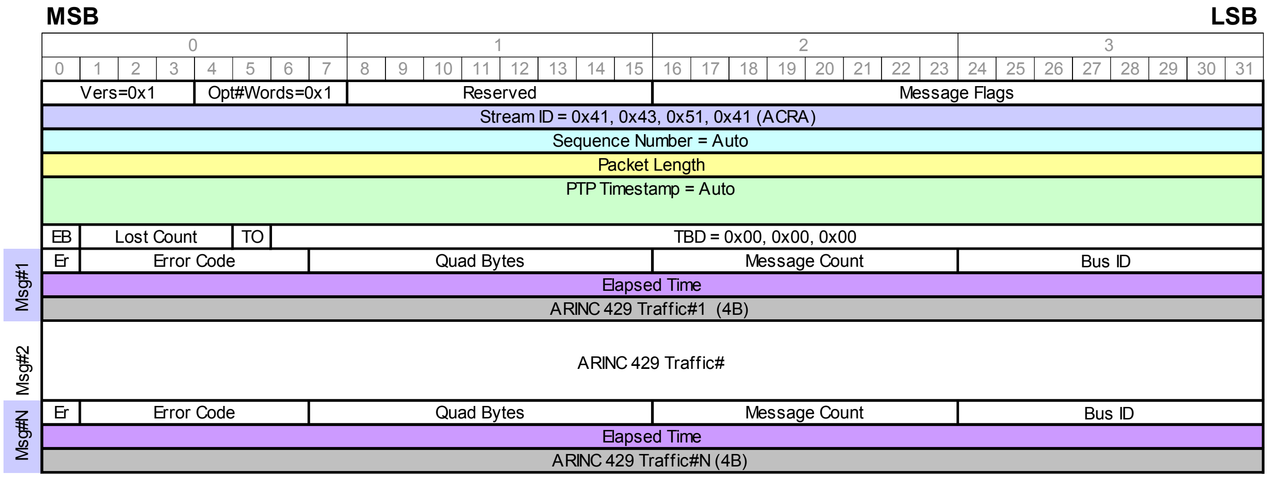

Figure 40-11: ARINC-429 iNET-X parser-aligned packet

For example, to facilitate real-time processing, the minimum payload size of an ARINC-429 packet is given as 1,008 bytes for a given default high-speed bus bit-rate of 100kbps, allowing for 84 ARINC 12-byte blocks to be carried in a single packet. This results in a maximum packet rate of 34 packets per second since the traffic on the bus may be asynchronous. The transmission properties for a 100kbps ARINC bus are summarized in the following table.

Table 40-7: MyARINC packet transmission properties

Transmission property	Value
ARINC-429 packet payload size (bytes)	1,008 bytes (84 ARINC message blocks of 12 bytes per ARINC block)
Total Ethernet frame length (bytes)	1,082 bytes (MAC header 14 bytes + IP 20 bytes + UDP 8 bytes + iNET-X 28 bytes + ARINC-429 data + MAC FCS 4 bytes)
Packet rate (packets per second)	34[1]
Total number of ARINC messages per packet	84 ARINC blocks
Total bit-rate (kbps)	294.3

1. 84 ARINC-429 messages per packet × (32 data bits + 4 inter message gap bits) = 3024 ARINC-429 bits on the bus to fill one packet
100kbps bus speed / 3024 bits on the bus per full Ethernet packet = 33.06 (rounded to 34) packets per second

40.3.6 iNET-X parser-aligned packet structure for MIL-STD-1553 bus monitoring

As MIL-STD-1553 messages arrive, the MIL-STD-1553 protocol tracker logic identifies them and maps them to corresponding transaction identifier codes, as in the following table.

Table 40-8: Transaction identifier codes

Message type	Mnemonic	Transaction ID
Bus Controller to Remote Terminal	BC → RT	0x00
Remote Terminal to Bus Controller	RT → BC	0x01
Remote Terminal to Remote Terminal	RT → RT	0x02
Mode Code without Data	M → S	0x03
Mode Code with Data (R)	MD → S	0x04
Mode Code with Data (T)	M → SD	0x05
Broadcast		
Bus Controller to Remote Terminals	BC → RTS	0x06
Remote Terminal to Remote Terminals	RT → RTS	0x07
Mode Code without Data	M	0x08
Mode Code with Data (R)	MD	0x09
Messages without Status reply		
Bus Controller to Remote Terminal	BC → RT	0x10
Remote Terminal to Bus Controller	RT → BC	0x11
Remote Terminal to Remote Terminal	RT → RT	0x12
Mode Code without Data	M → S	0x13
Mode Code with Data (R)	MD → S	0x14
Mode Code with Data (T)	M → SD	0x15
Remote Terminal to Remote Terminal	RT → RTS	0x17

Only valid MIL-STD-1553 transactions are stored in the packet (see the following figure). If an error occurs, only the parser information word and elapsed time tag are written to the iNET-X packet; the message is then dumped. The error bit and error code (see Table 40-10 on page 15), which are set in the parser information word, indicate the cause of the error.

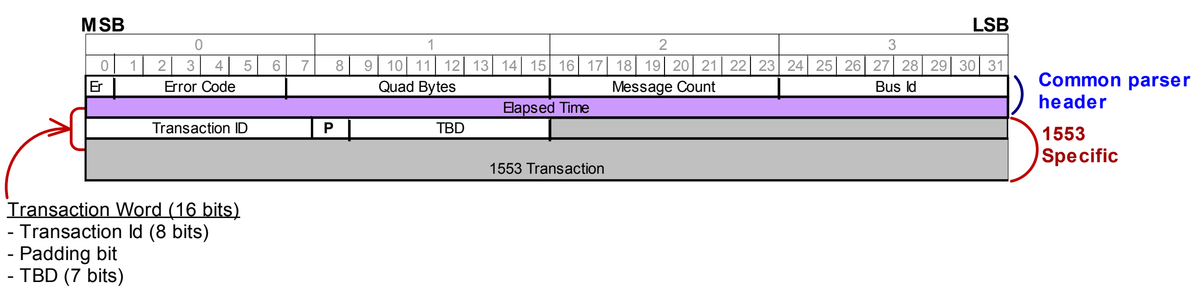

Figure 40-12: MIL-STD-1553 transaction iNET-X parser-aligned packet

In the case where a remote terminal is off-line but it is still desirable to capture data sent to it, you can set options to include Accept Rx Message With No Status and Accept Tx Message With No Status. In such a scenario, the parser information word indicates an error but the 1553 traffic is still captured.

The transaction identifier may be used to indicate when response times are carried in the iNET-X parser-aligned block. The MIL-STD-1553 standard specifies a minimum response time of 4µs and a maximum response time of 12µs. However, the bus controller waits up to 20µs before determining a timeout has occurred. The granularity of response time on the 8MHz bus can be measured to a resolution of 125ns. When a timeout occurs, the response time is set to 0xFF.

In order to facilitate the decoding and decommutation of the MIL-STD-1553 parser-aligned iNET-X packets, the first word of a MIL-STD-1553 transaction is the transaction word where:

- Transaction word (2 bytes): metadata providing protocol tracking information, health, and status of the message.
  - Transaction ID (Bits [0:7]): see Table 40-8 on page 13
  - P (Bit 8): bit to indicate if the parser message has been padded to fall on a 4-byte boundary
  - TBD (Bits [9:15]): TBD

Where a MIL-STD-1553 transaction does not fall on a 32-bit boundary, the MIL-STD-1553 parser-aligned message is padded. In MIL-STD-1553 the maximum number of padding words (16-bit) possible in a single transaction is one. Therefore a single bit is sufficient to indicate if a transaction message has been padded. The following two figures illustrate individual MIL-STD-1553 BC ⟶ RT transactions, with and without padding.

MSB **LSB**

<table>
<tr><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td>Er</td><td colspan="6">Error Code</td><td colspan="9">Quad Bytes=6</td><td colspan="8">Message Count</td><td colspan="8">Bus ID</td></tr>
<tr><td colspan="32">Elapsed Time</td></tr>
<tr><td colspan="8">Transaction ID: RT->BC</td><td>P=0</td><td colspan="7">TBD</td><td colspan="16">Cmd=TX</td></tr>
<tr><td colspan="16">Response</td><td colspan="16">Status</td></tr>
<tr><td colspan="16">Data#1</td><td colspan="16">Data#2</td></tr>
<tr><td colspan="16">Data#3</td><td colspan="16">Data#4</td></tr>
</table>

Figure 40-13: MIL-STD-1553 transaction iNET-X parser-aligned message without padding

MSB **LSB**

<table>
<tr><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td>Er</td><td colspan="6">Error Code</td><td colspan="9">Quad Bytes=7</td><td colspan="8">Message Count</td><td colspan="8">Bus ID</td></tr>
<tr><td colspan="32">Elapsed Time</td></tr>
<tr><td colspan="8">Transaction ID: BC->RT</td><td>P=1</td><td colspan="7">TBD</td><td colspan="16">Cmd=RX</td></tr>
<tr><td colspan="16">Data#1</td><td colspan="16">Data#2</td></tr>
<tr><td colspan="16">Data#3</td><td colspan="16">Data#4</td></tr>
<tr><td colspan="16">Data#5</td><td colspan="16">Response</td></tr>
<tr><td colspan="16">Status</td><td colspan="16">Padding</td></tr>
</table>

Figure 40-14: MIL-STD-1553 transaction iNET-X parser-aligned message with padding

The following figure demonstrates the iNET-X parser-aligned payload structure for MIL-STD-1553 with the following three parsed MIL-STD-1553 transactions in the payload:

- BC ⟶ RT: With four data words transferred.
- RT ⟶ RT: With five data words transferred.
- RT ⟶ BC: With four data words transferred.

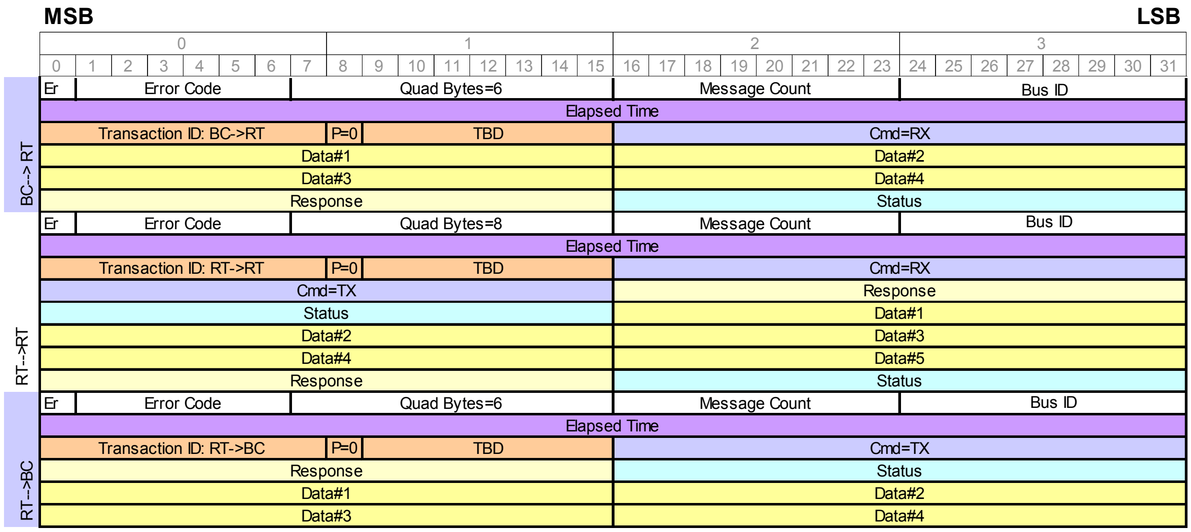

Figure 40-15: iNET-X parser-aligned payload for multiple MIL-STD-1553 transactions

For example, the MIL-STD-1553 bus has a peak bit-rate of 1Mbps. However, the messages and transactions transmitted are variable in length and asynchronous. If it is assumed that the mean MIL-STD-1553 transaction is 14 bytes long (comprising of a command, response, data words, and a status word), the iNET-X parser block structure encapsulating this transaction is therefore 24 bytes (including the MIL-STD-1553 transaction ID, parser information word and the elapsed time). In this case, each iNET-X parser-aligned packet may contain up to 60 parser blocks in the payload (that is, 60 parser blocks x 24 bytes per block which results in 1440 bytes of payload, or 840 bytes of MIL-STD-1553 bus data). To ensure that no data is lost on the bus, packets are generated at a rate of 150 packets per second.

The transmission properties for a 1Mbps MIL-STD-1553 bus are summarized in the following table.

Table 40-9: MyMIL-STD-1553 packet transmission properties

Transmission property	Value
MIL-STD-1553 packet payload size (bytes)	1440 bytes (60 MIL-STD-1553 parser blocks of 24 bytes per block where each block contains 14 bytes of MIL-STD-1553 bus data)
Total Ethernet frame length (bytes)	1514 bytes MAC header 14 bytes + IP 20 bytes + UDP 8 bytes + iNET-X 28 bytes + MIL-STD-1553 data + MAC FCS 4 bytes
Packet rate (packets per second)	150
Total number of MIL-STD-1553 transactions per packet	60 transactions assuming 14 bytes per transaction
Total bit-rate (kbps)	1816.8

Table 40-10: Error codes

Error code	Description	Error code	Description
0_{16}	Reserved for future use.	20_{16}	Expected STS was a data word.
116	Data word did not have enough bits.	21_{16}	Expected STS was invalid.
2_{16}	Data word had bit error.	22_{16}	Expected STS had incorrect RT.
3_{16}	Data word had parity error.	23_{16}	Expected STS had contiguous traffic.
4_{16}	Non-data word did not have enough bits.	24_{16}	Expected STS timed out.
5_{16}	Non-data word had bit error.	25_{16}	Expected STS had no contiguous data word.

Table 40-10: Error codes (continued)

Error code	Description	Error code	Description
6_{16}	Non-data word had parity error.	26_{16}	Reserved for future use.
7_{16}	Reserved for future use.	27_{16}	Reserved for future use.
8_{16}	Expected data word was non-data word.	28_{16}	Reserved for future use.
9_{16}	Expected data word did not have contiguous word.	29_{16}	Reserved for future use.
A_{16}	Expected last data word was not last.	$2A_{16}$	Reserved for future use.
B_{16}	Expected mode data word was non-data word.	$2B_{16}$	Expected Tx CMD of RT to RT(s) had different number of words than Rx CMD.
C_{16}	Expected mode data word has contiguous traffic.	$2C_{16}$	Expected Tx CMD of RT to RT(s) had same RT as Rx CMD.
D_{16}	Reserved for future use.	$2D_{16}$	Expected Tx CMD of RT to RT(s) had contiguous traffic.
E_{16}	Reserved for future use.	$2E_{16}$	Second CMD in RT-RT was not a TX.
F_{16}	Reserved for future use.	$2F_{16}$	Reserved for future use.
10_{16}	Expected first CMD was a data word.	30_{16}	Reserved for future use.
11_{16}	Reserved for future use.	31_{16}	Reserved for future use.
12_{16}	Reserved for future use.	32_{16}	Reserved for future use.
13_{16}	Reserved for future use.	33_{16}	Reserved for future use.
14_{16}	Reserved for future use.	34_{16}	Reserved for future use.
15_{16}	Expected first CMD had contiguous traffic.	35_{16}	Reserved for future use.
16_{16}	Expected first CMD was Rx with no contiguous data word.	36_{16}	Reserved for future use.
17_{16}	Expected first CMD was Mode with no contiguous data.	37_{16}	Reserved for future use.
18_{16}	Expected second STS of RT to RT was data word.	38_{16}	Reserved for future use.
19_{16}	Expected second STS of RT to RT had incorrect RT.	39_{16}	Reserved for future use.
$1A_{16}$	Expected second STS of RT to RT had contiguous traffic.	$3A_{16}$	Reserved for future use.
$1B_{16}$	Expected second STS of RT to RT timed out.	$3B_{16}$	Reserved for future use.
$1C_{16}$	Reserved for future use.	$3C_{16}$	Reserved for future use.
$1D_{16}$	Reserved for future use.	$3D_{16}$	Reserved for future use.
$1E_{16}$	Reserved for future use.	$3E_{16}$	Reserved for future use.
$1F_{16}$	Reserved for future use.	$3F_{16}$	Reset occurred since last read.

Tx = transmit; STS = status; CMD = command.

40.3.7 iNET-X Event packets

The format and structure of an Event packet is shown in the following figure. The recorded errors and events are listed in Table 40-12 on page 21 in the Appendix.

MSB **LSB**

<table>
<tr><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="4">Vers=0x1</td><td colspan="4">Opt#Words=0x1</td><td colspan="8">Reserved</td><td colspan="16">Message Flags</td></tr>
<tr><td colspan="32">Stream ID = 0x41, 0x43, 0x51, 0x41 (ACRA)</td></tr>
<tr><td colspan="32">Sequence Number = Auto</td></tr>
<tr><td colspan="32">Packet Length</td></tr>
<tr><td colspan="32">PTP Timestamp = Auto</td></tr>
<tr><td>EB</td><td colspan="4">Lost Count</td><td>TO</td><td colspan="26">TBD = 0x00, 0x00, 0x00</td></tr>
<tr><td colspan="16">Error/Event Code</td><td colspan="16">Parameter</td></tr>
<tr><td colspan="32">Description (null-terminated)</td></tr>
</table>

Figure 40-16: iNET-X Event packet

40.4 IENA packetization

IENA is the application layer protocol developed by Airbus that defines the packet header and packetization rules for the transmission of acquired data as UDP/IP packets. Similar to iNET-X, IENA partitions logical groupings of data into packet streams, each uniquely identified by an IENA Key. The IENA standard defines the application layer IENA header, shown in the following figure, and payload structures.

MSB **LSB**

<table>
<tr><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="16">IENA Key (2B)</td><td colspan="16">Size (2B)</td></tr>
<tr><td colspan="32">Time (6B)</td></tr>
<tr><td colspan="16"></td><td colspan="8">Key Status (1B)</td><td colspan="8">N2 Status (1B)</td></tr>
<tr><td colspan="16">Seq (2B)</td><td colspan="16"></td></tr>
<tr><td colspan="32">IENA Payload</td></tr>
<tr><td colspan="8"></td><td colspan="8">OPTIONAL Padding (1B)</td><td colspan="16">End (2B) = 0xDEAD</td></tr>
</table>

Figure 40-17: IENA packet header

The IENA packet header contains the following elements:

- IENA Key (16 bits): Points to a unique (system wide) metadata package description in the XidML. The package description describes the format, structure, and data contained in the IENA packet payload.
- Size (16 bits): The length of the complete IENA packet in 16-bit words including the IENA header, the payload, padding, and the IENA Trailer field. Padding is used when the length of the IENA packet does not fall on a 16-bit boundary. The valid range for the Size field is therefore 8 to 32753, which is the maximum possible payload size.
- Time (48 bits): The time of the current year in microseconds since the 1st January.
- Key Status (8 bits): This field is reserved for future use.
- N2 Status (8 bits): This field is specific to Airbus equipment and is not used. This fields has a default value of 0x00.
- Sequence Number (16 bits): Increments by one for each packet generated and transmitted with a given IENA Key and wraps around back to 0. The valid sequence number range is 0 to 65535.
- Padding (8 bits): The IENA standard specifies that packets must be 16-bit word aligned, therefore to achieve alignment it may be necessary to pad the IENA payload to fall on a 16-bit boundary.
- Trailer End field (16 bits): This is specified as being a unique value for all keys where the default value is 0xDEAD.

40.4.1 IENA payload formats

The IENA standard defines three main payload structures. These are called:

- Positional - This payload structure is analogous to the iNET-X placed payload structure in that parameters and samples are placed within the payload.
- Standard - This is similar to the Positional type however samples for a given parameter are placed contiguously in the payload and a 2-byte Parameter Identifier describes each contiguous block of samples.
- Message - This payload structure is designed for asynchronous variable length data sets where each message has an associated Message Identifier and timestamp.

Curtiss-Wright products only support the positional type of IENA payload structure. Other IENA payload structures are not described in this technical note.

40.4.2 IENA

An IENA packet contains a standard IENA layer header and footer. Between these, the data field contains one or more parameters of a specified type. For the KAD/ABM/103 each ARINC-429 message can be formatted as either a D Type or N Type IENA parameter. In both cases, the 16-bit parameter ID for each parameter is composed by combining the bus ID (for channels 0 to 23) with the label and SDI fields from the received message. As shown in the following figure, these fields appear in the message from MSB to LSB in the order SDI[1:0], bus ID[4:0], and label[7:0]. The MSB is always equal to 0.

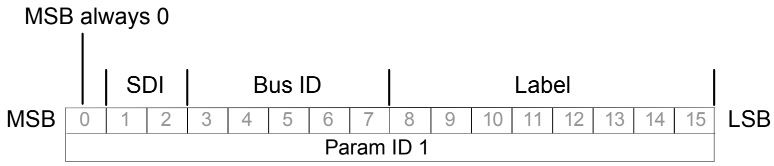

Figure 40-18: Parameter ID from received message

As shown in the following figure, D Type parameters consist of a 16-bit parameter ID, a 16-bit delay field, and the 32-bit ARINC-429 message. The delay field indicates the difference in microseconds between the timestamp in the IENA packet's header and the received timestamp for the specific message contained in the parameter. ARINC-429 messages use 8 bytes per N Type parameter.

	0	1	2	3
	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
#1	Param ID 1		Delay 1	
	ARINC-429 Word 1			
#2	Param ID 2		Delay 2	
	ARINC-429 Word 2			

Figure 40-19: D Type IENA parameter for ARINC-429

As shown in the following figure, N Type parameters include the 16-bit parameter ID and the 32-bit message, but the delay field is omitted and the only timestamp is that of the packet, which relates to the first packetizer message. ARINC-429 messages use 6 bytes per N Type parameter.

0	1	2	3
0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
Param ID 1		ARINC-429 Word 1 (0:15)	
ARINC-429 Word 1 (16:31)		Param ID 2	
ARINC-429 Word 2 (0:31)			
Param ID 3		ARINC-429 Word 3 (0:15)	
ARINC-429 Word 3 (16:31)		…	

Figure 40-20: N Type IENA parameter for ARINC-429

40.4.2.1 IENA positional payload formats

Only the Positional type payload structure is supported, the following subsections provide applied examples of the positional payload structure for use with packetizer and placed parameter payloads. There are a number of rules that govern the placement of parameters within the positional payload structure:

- Length of the parameters: All parameters within the payload are an even number N bytes long where 2 ≤ N ≤ 14.

- All parameters must have the same length.
- The length of the parameters can be determined from the IENA Key status field (that is, N/2)

- Number of parameters in the pattern: The positional payload interleaves parameter samples in a block pattern. The first block pattern contains the first sample for all parameters in the payload, the second block pattern contains the second sample for all parameters, and so on.
- The metadata file: This describes the name of the parameter and the location of each parameter in the pattern. Subsequent sample locations can be inferred knowing the sampling rate and the number of samples for a given parameter in the payload.

The following figure illustrates the interleaved sample placement in block patterns where each parameter is 16 bits wide, that is, N = 2.

Block Pattern #1				Block Pattern #2				Block Pattern ….				Block Pattern #x			
A1	B1	C1	D1	A2	B2	C2	D2	A..	B..	C..	D..	Ax	Bx	Cx	Dx

Figure 40-21: IENA positional payload format

The previous figure illustrates the positional payload structure where each parameter is 2 bytes long, each block pattern contains a single sample for four parameters, and there are X number of these block patterns repeated in the payload. The XidML file is used to specify the location of the parameters in the block.

40.4.2.2 IENA positional packet format for analog

Although IENA positional types specify interleaved placement rules for the parameters, this restriction is not enforced. The following figure illustrates the interleaved placement of the parameter samples while Figure 40-23 on page 20 illustrates the payload structure for contiguously placed parameter samples.

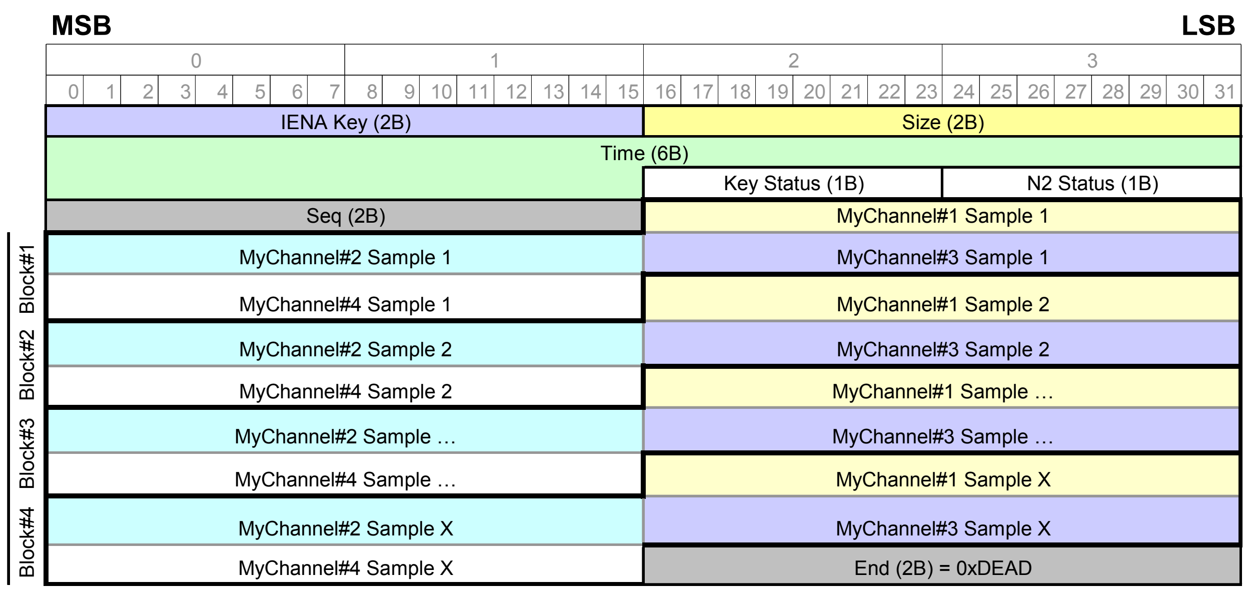

Figure 40-22: IENA interleaved positional placed packet structure

MSB **LSB**

	0		1		2	3
	0 1 2 3 4 5 6 7		8 9 10 11 12 13 14 15		16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	IENA Key (2B)				Size (2B)	
	Time (6B)					
					Key Status (1B)	N2 Status (1B)
	Seq (2B)				MyChannel#1 Sample 1	
Channel#1	MyChannel#1 Sample 2				MyChannel#1 Sample …	
	MyChannel#1 Sample X				MyChannel#2 Sample 1	
Channel#2	MyChannel#2 Sample 2				MyChannel#2 Sample …	
	MyChannel#2 Sample X				MyChannel#3 Sample 1	
Channel#3	MyChannel#3 Sample 2				MyChannel#3 Sample …	
	MyChannel#3 Sample X				MyChannel#4 Sample 1	
Channel#4	MyChannel#4 Sample 2				MyChannel#4 Sample …	
	MyChannel#4 Sample X				End (2B) = 0xDEAD	

Figure 40-23: IENA contiguous positional placed packet structure

40.5 Appendix

Table 40-11: Parser-aligned MIL-STD-1553 Transaction Identifier codes

Message with Status (S) reply	Mnemonic	Transaction ID
Bus Controller to Remote Terminal	BC ⟶ RT	0x00
Remote Terminal to Bus Controller	RT ⟶ BC	0x01
Remote Terminal to Remote Terminal	RT ⟶ RT	0x02
Mode code without data	M ⟶ S	0x03
Mode code with Data (receive)	MD ⟶ S	0x04
Mode code with Data (transmit)	M ⟶ SD	0x05
Broadcast messages	**Mnemonic**	**Transaction ID**
Bus Controller to Remote Terminals	BC ⟶ RTs	0x06
Remote Terminal to Remote Terminals	RT ⟶ RTs	0x07
Mode code without data	M	0x08
Mode code with Data (receive)	MD	0x09
Messages without Status (S) reply	**Mnemonic**	**Transaction ID**
Bus Controller to Remote Terminal	BC ⟶ RT	0x10
Remote Terminal to Bus Controller	RT ⟶ BC	0x11
Remote Terminal to Remote Terminal	RT ⟶ RT	0x12
Mode code without data	M ⟶ S	0x13
Mode code with Data (receive)	MD ⟶ S	0x14
Mode code with Data (transmit)	M ⟶ SD	0x15

Table 40-12: Event and error codes

	Name	Error/Event code	Parameter	Description Zero terminated ANSI text
Event	Event Pressed	0x100	Event number	EVENT PRESSED
	Event Start	0x101	Event number	EVENT START
	Event Released	0x102	Event number	EVENT RELEASED
	Start	0x103	Event number	START
	Stop	0x104	Event number	STOP

Table 40-12: Event and error codes

	Name	Error/Event code	Parameter	Description Zero terminated ANSI text
Remote Event	Remote Event Pressed	0x200	Event number	REMOTE EVENT PRESSED
	Remote Event Start	0x201	Event number	REMOTE EVENT START
	Remote Event Released	0x202	Event number	REMOTE EVENT RELEASED
	Remote Start	0x203	Event number	REMOTE EVENT START
	Remote Stop	0x204	Event number	REMOTE EVENT STOP
Error	Packet Dropped	0x800	Number of packets dropped	PACKET(S) DROPPED
	PTP Grandmaster Lost	0x900	Event number	GRANDMASTER LOST
	PTP Time Reliable	0x901	Event number	TIME RELIABLE
	PTP Clock Jumped	0x902	Event number	CLOCK JUMPED
	PTP Time Unreliable	0x903	Event number	TIME UNRELIABLE

40.6 Recommended reading

To better understand this paper, read the documents listed in the following two tables.

Table 40-13: Data sheets

Document	Description
KAD/ADC/126	Accelerometer ADC (current excitation, programmable analog gain, 25kHz b/w) - 4ch at 100ksps. Explains the electrical interface, possible sampling rates and configuration.
SSR/CHS/001	Ethernet multi-role recorder (CompactFlash®, voice connector) - 4 user-slots. Explains the STOP/START switch, LCD display and power consumption.
KAD/MBM/101	Dual redundant MIL-STD-1553 bus packetizer Explains the electrical interface, possible sampling rates and configuration.
KAD/VID/106	H.264 video encoder (analog video input) - 1ch Explains the electrical interface, possible sampling rates and configuration.
DAS Studio 3	Explains the graphical user interface to set up the SSR/CHS/00X.

Table 40-14: Technical notes

Document	Description
TEC/NOT/051 - Ethernet frames, Wireshark® and FAT32	Explains the rationale for large Ethernet frames that are close to but not exceeding 1500 bytes. Describes the FAT32 file system on the CompactFlash card.

NOTE: For more information on XidML, go to http://www.xidml.org.