AXN/UBM/401

RS-232, RS-422 or RS-485 serial bus parser/packetizer - 16ch

CURTISSWRIGHTDS.COM

Key Features

- Monitors up to 16 RS-422/485/232 busses
- Coherently parses traffic and tags for up to 1,024 messages from 4 to 1,024 bytes
- Aperiodic transmission of packetized serial messages including tags as iNET-X parser-aligned payload structure and Chapter 10 UART Data Packets
- Bit-rates from 300 bps to 10,000,000 bps
- 7/8 bits per word with odd, even or no parity
- Programmable start sequence (1 to 8 bytes), stop sequence (1 byte or by fixed length) and idle time

Applications

- Interfacing with serial data links

Overview

The AXN/UBM/401 is used to monitor up to 16 RS-232/422/485 channels. The module can parse (coherently extract specific bytes) and/or packetize (insert messages in packets for Ethernet transmission) each channel at the same time.

The signal type (RS-232/422/485), bits per word, and parity are programmable on a channel-by-channel basis.

In the parser, a total of up to 1,024 complete messages are triple buffered so that the stale indication is message-wide. Each message can be up to 1,024 characters (bytes) long (including start and stop characters). Each message is time tagged; a message is considered found when a start sequence of up to eight specific bytes is received. The end-of-message delimiter is determined by a user-defined stop character or after a specific number of words are received. A message is not updated if any sequence is incorrect.

Independently of the parser, a packet stream is generated for each channel. All received bytes are encapsulated in an Ethernet frame. A programmable gap time allows the module to split the incoming bytes into shorter timestamped sequences. A block header attached to each sequence stores the channel index, length, and the time of reception of each message. These Ethernet frames may be transmitted aperiodically to optimize network bandwidth utilization and memory usage when recording serial traffic. Optionally, the message stream can be filtered such that only certain messages are captured and packetized.

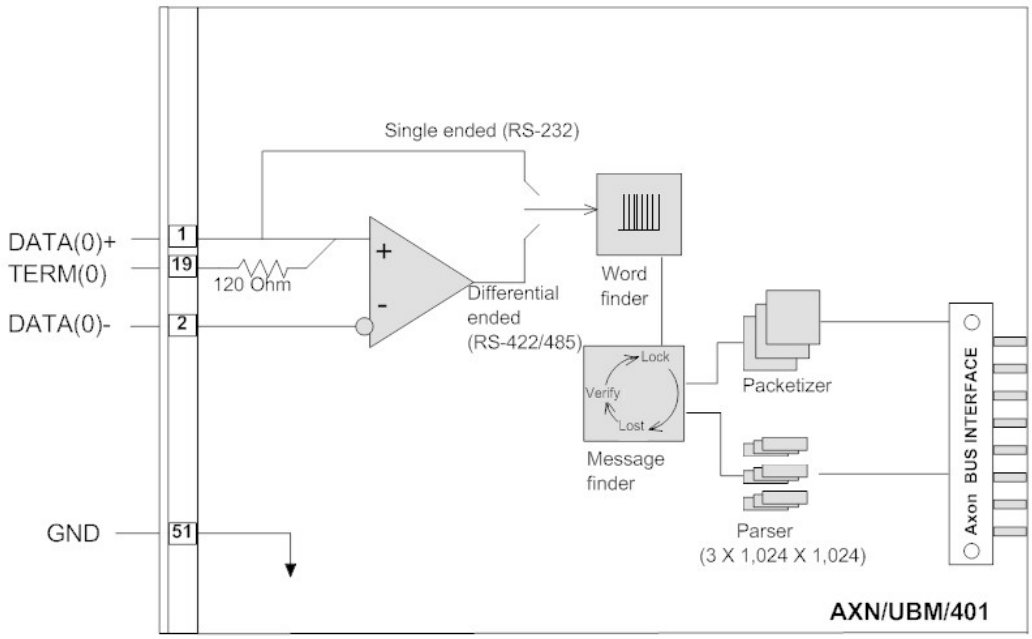

Figure 1: First of 16 channels of the AXN/UBM/401

Specifications

All values provided in the following specification tables are valid within the operating temperature range specified under "Environmental ratings" in the "General specifications" table.

TABLE 1	General specifications				
PARAMETER	MIN.	TYP.	MAX.	UNITS	CONDITION/DETAILS
Slots	–	–	1	–	Can be placed in any user-slot in any combination.
Mass					
	–	60	–	g	
	–	2.12	–	oz	Design metric is grams.
Height above chassis					For recommended clearance requirements see the *CON/KAD/002/CP* data sheet.
bare connector	–	–	11	mm	
bare connector	–	–	0.43	in.	Design metric is millimeters.
Power consumption					
+15V	–	110	135	mA	
total power	–	1.65	2.03	W	Particular combinations of Axon chassis and modules may have power or current limitations. For details, contact Curtiss-Wright support (acra-support@curtisswright.com).
Environmental ratings					
operating temperature	-40	–	85	°C	Chassis base/side plate temperature.
storage temperature	-55	–	105	°C	

TABLE 2	RS-422/RS-485 inputs				
PARAMETER	MIN.	TYP.	MAX.	UNITS	CONDITION/DETAILS
Inputs	–	–	16	–	
Signalling rate					
DATA	0.0003	–	10	Mbps	NRZ-L.
Input voltage					
operating range	-15	–	15	V	Do not exceed operating range.
logic 0	–	–	0.2	V	(190 mV hysteresis) V_{IN+} - V_{IN-}.
logic 1	0.2	–	–	V	(190 mV hysteresis) V_{IN+} - V_{IN-}.
common mode voltage	-7	–	12	V	
overvoltage protection	-15	–	15	V	Voltage in excess of these values can damage input.
ESD protection	–	16	–	kV	Human Body Model.
Input resistance					
between inputs	–	250	–	kΩ	Module powered up and no input termination.
between inputs	–	260	–	kΩ	Module powered down and no input termination.
between inputs	–	120	–	Ω	Module powered up and inputs terminated with external or internal termination resistor.
between inputs	–	120	–	Ω	Module powered down and inputs terminated with external termination resistor.

TABLE 3	RS-232 inputs				
PARAMETER	MIN.	TYP.	MAX.	UNITS	CONDITION/DETAILS
Inputs	–	–	16	–	
Signaling rate					
DATA[15:0]	0.0003	–	1	Mbps	
Input voltage					
operating range	-15	–	15	V	Do not exceed operating range.
logic 0	2.5	–	–	V	
logic 1	–	–	0.6	V	
overvoltage protection	-15	–	15	V	Voltages outside of this range can damage input.
ESD protection	16	–	16	kV	Human Body Model.
Lightning protection					DO-160 Section 22, Waveform 4, Level 3, Pin Injection
voltage	–	–	300	V	
current	–	–	60	A	
Input resistance					
each input to GND	–	5	–	kΩ	Module powered up.
each input to GND	–	130	–	kΩ	Module powered down.

Setting up the AXN/UBM/401

All module setup can be defined in XML using XidML® schemas (see http://www.xidml.org).

Instrument settings

SETUP DATA	CHOICE	DEFAULT	NOTES
Manufacturer	-	-	-
Name	ACRA CONTROL	ACRA CONTROL	Name of manufacturer.
PartReference	AXN/UBM/401	AXN/UBM/401	The instrument part reference.
SerialNumber	AAA1234	AAA1234	Unique name for each module.
Settings	-	-	-
Parser Data Endianness	First byte at low end of word First byte at high end of word	First byte at high end of word	Select byte order in data field of parser slot.
Fill Value	0000 to FFFF	AAAA	All parser slot parameters (data and tag) except the MessageInfo will display this value until a parsed message is received.
Processes	-	-	-
Parser(1023:0)	-	-	-
Channels	-	-	-
Serial-In(15:0) *Serial Input*	-	-	Connection between the module channel and RS-232/422/485 data bus.
Settings *Serial Interface*	-	-	-
Signal Type	RS-232 RS-422 RS-485	RS-422	Type of data stream.
Baud Rate	300 to 10000000	9600	Specifies the number of symbols transmitted per second.
Data Bits Per Word	7 8	8	Bits per incoming data word. Two incoming words are packed into a single 16-bit output word.
Parity	No Parity Even Parity Odd Parity	No Parity	Configure whether parity bit is present in incoming data.
Parity Check	Report failure Not checked	Report failure	How parity failures are reported.
Gap Between Messages	0 to 10000	10	Time gap between consecutive characters required before starting new message. Value expressed in unit of character periods at configured bit rate. Set to 0 to ignore all gaps, a gap of 1 is not supported.
Programmable Termination	Disabled Enabled	Disabled	Enable internal 120 ohm termination resistance. (Note: Not active when power is disabled. Use wiring selectable termination instead if termination is required at all times.)
Settings *Packetizer*	-	-	-
Packetizer Format	iNET-X Chapter 10	iNET-X	Selects the packetizer header format for all channels.
Stream Id	00 to FFFFFFFF	FFFFFFFF	iNET-X stream ID for selected channel if a packet is generated via the assertion of Packetization Enabled. This is a conditional setting and is only active when the Packetizer Format is set to iNET-X.

SETUP DATA	CHOICE	DEFAULT	NOTES
Channel Id	00 to FFFF	FFFF	Chapter 10 channel ID for selected channel if a packet is generated via the assertion of Packetization Enabled. This is a conditional setting and is only active when the Packetizer Format is set to Chapter 10.
Packetization Enabled	True False	False	Enables the generation of a stream of packets containing messages from this channel.
Max Packet Payload Size	200 to 511	511	Size of packet payload in words.
Packet Timeout	10 to 65	50	Generates a packet when the oldest data recorded is this old (ms).
One Message Per Packet	True False	False	Set to True to limit packets to one parser block. After each gap, the next message will start a new packet. Only use this setting if the number of packets scheduled exceeds the number of messages expected and the maximum size message is less than the Packet Size setting.
Packetizer Filter Mode	Pass By Rule Block By Rule Pass All	Pass All	Specifies the filtering mode for the channel. Block By Rule means that messages that match defined filter rules are blocked and all other messages are passed through. Pass By Rule means that messages that match defined filter rules are passed through and all other messages are blocked. Pass All means that alll messages are passed through. This is only applicable when Packetization is enabled.
Packetization Sink	All Controller Only Slot 1 Slot 2 Slot 3 Slot 4 Slot 5 Slot 6 Slot 7 Slot 8 Slot 9 Slot 10 Slot 11 Slot 12 Slot 13 Slot 14 Slot 15 Slot 16	All	Selects which modules the packetizer package will be sent to for transmission or storage.
Processes	-	-	-
Packetizer-Filter-Rule(1023:0)	-	-	This Filter-Rule process is used to map a Filter Action to a Filtering Rule (such as a serial message definition).
Settings	-	-	-
Packetizer Filter Rule Enabled	False True	False	Enables messages to be either Blocked or Passed (depending on the channel Packetizer Filter Mode setting) when they meet the packetizer filtering condition referenced by this process.

Parameter definitions

NAME/DESCRIPTION	BASE UNIT	DATA FORMAT	BITS	REGISTER DEFINITION
Global Parameters				
Report Indicates the status of the monitor.	BitVector	BitVector	16	R[15:0] R(15) ErrorIndicator - 1 indicates an error occurred since last read. R[14:11] BusNumber - The bus the error occurred on. R[10:4] Reserved - Reserved for future use. R[3:0] ErrorCodes - Each bit set indicates that particular error occurred since report word last read. After the report word has been read, further reads will return the last non-zero error code with the ErrorIndicator bit not set. 0001: Parity error. 0010: Bad stop bit. 0100: Too many data words. Stop character not found in 1024 characters. Other: Reserved for future use.
ModuleMessageCount Increments by one each time the parser logic detects a complete message.	BitVector	BitVector	16	R[15:0]
ModuleTemperature Temperature of the AXN/UBM/401 module	Celsius	OffsetBinary	16	R[15:0] R[15:4] ModuleTemperature - Temperature of the AXN/ABM/401 module in steps of 0.0625 C. 0xC90 = -55C, 0x7D0 = 125C R[3:0] Reserved
Parser(1023:0) Parameters				
MessageSize Number of received bytes (including start bytes, for example, 1 + 8 = 9 bytes in a message). Minimum length of message is 2 bytes (1 byte of start sequence plus 1 stop byte or at least 1 byte in case of fixed length message).	Count	OffsetBinary	16	R[15:0]
MessageIrigTime48 48-bit wide IRIG time word.	BitVector	BitVector	48	R[47:0]
MessageTimeHi Hours and minutes time midway through first transmitted bit.	BitVector	BitVector	16	R[47:32] R[15:13] Reserved - Reserved for future use. R[12:7] Hours - BCD Hours 0 to 23. R[6:0] Minutes - BCD Minutes 0 to 59.
MessageTimeLo Seconds and centiseconds time midway through first transmitted bit.	BitVector	BitVector	16	R[31:16] R(15) Reserved - Reserved for future use. R[14:8] Seconds - BCD Seconds 0 to 59. R[7:0] Centiseconds - BCD Centiseconds 0 to 99.
MessageTimeMicro Microsecond time midway through first transmitted bit.	Second	BCD	16	R[15:0] R[15:0] Microseconds - BCD Microseconds 0 to 9999.
MessageCount Received message count.	Count	OffsetBinary	16	R[15:0]

NAME/DESCRIPTION	BASE UNIT	DATA FORMAT	BITS	REGISTER DEFINITION
MessageInfo Stale/skipped indication for this parsed message.	BitVector	BitVector	16	R[15:0] R(15) Empty - ID is empty (no message received). R(14) Stale - 1 indicates this message was read before. R(13) Skipped - 1 indicates this message overwrote another. R[12:0] Reserved - Reserved for future use.
Serial-In(15:0) Parameters				
ChannelMessageCount Increments by one each time the parser logic detects a complete message.	BitVector	BitVector	16	R[15:0]
ChannelByteCount Count of bytes received on this bus.	Count	OffsetBinary	16	R[15:0]
ChannelErrCount Count of errors detected on this bus.	BitVector	BitVector	16	R[15:0]

NOTE: It is recommended that names do not contain any of the following five characters "/><\.

Getting the most from the AXN/UBM/401

Each of the 16 bus connections can be independently selected as RS-232, RS-422 or RS-485. For RS-232 (single ended), use a DATA(x)+ pin and leave the corresponding DATA(x)- pin unconnected. Both data pins are required when a differential mode is selected (RS-422 or RS-485).

The following figure shows how to optionally terminate the RS-422 and RS-485 receivers by using the internal termination resistor provided on the module (TERM[x] pin). It is important to ground each source of RS-232, RS-422 or RS-485. Star grounding provides optimal noise rejection. For details on grounding, see *TEC/NOT/063 - Grounding and shielding of the Axon and Acra KAM-500*.

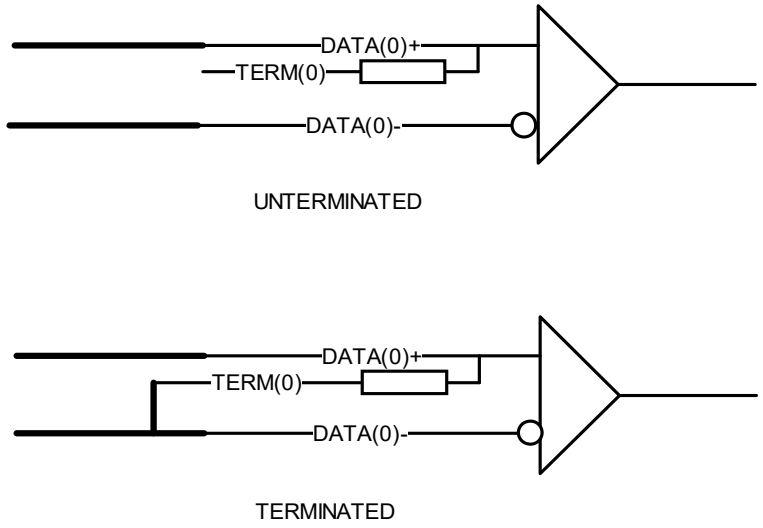

Figure 2: Optional RS-422/RS-485 third pin termination

Maximum message length

The maximum message length is 1,024 bytes. This length includes the bytes used for the start pattern and the end byte (if an end byte is used).

Error codes

Code[1]	DESCRIPTION
1_{16}	Parity error.
2_{16}	Bad stop bit.
4_{16}	Last character not found in 1,024 characters (in report word only; not in packetizer error code).
0_{16}, 3_{16}, 5_{16}, 6_{16}, 7_{16}	Reserved for future use.

1. Error codes are in hexadecimal.

iNET-X Packetizer format

The AXN/UBM/401 recognizes bytes as having a start bit, data bits (seven or eight, programmable), a parity bit (optional), and a stop bit. The AXN/UBM/401 gathers groups of bytes into standard iNET-X parser-aligned blocks. Each output packet can contain multiple blocks. If configured to receive 7-bit bytes, each byte is padded to 8 bits using a zero at the most significant bit before saving. The generalized iNET-X payload structure for parser-aligned packets is shown in Figure 4 on page 9 and examples of parser block formats are shown in Figure 5 on page 10.

A parser block is created for each logical message, based on the gaps between sequences of bytes. A new parser block is created at the start of each iNET-X packet and also when the time gap between successive bytes exceeds a programmable

threshold.

The gap time threshold is programmed in characters. Characters are defined using the Data Bits Per Word setting, Parity setting, and Start/Stop bits.

Figure 3: Serial character structure

By setting the threshold value to just under the minimum gap time expected between messages (dependent on the data source), it is possible to correctly split the bytes of the incoming stream into individual messages, as required.

If a single message continues past the end of one packet with no gap, the succeeding bytes are packetized in a new parser block in the next packet. This is indicated in the current parser aligned block header as well as in the following block's header in the next iNET-X packet. The location of these continuation bits is marked Cn in the example parser blocks in Figure 5 on page 10.

If the gap threshold is set to 0, gap detection is disabled, and in the absence of errors, each iNET-X packet contains a single parser block whose length is controlled by the packet timeout and size settings selected.

Furthermore, the AXN/UBM/401 supports filtered packetizers through the Packetizer Filter Mode setting, (see “Setting up the AXN/UBM/401” on page 4). Packetizer Filter Mode allows messages to be tracked according to classification rules defined using the Serial Builder tool in DAS Studio 3; messages can either be included for packetization (Pass By Rule) or excluded (Block By Rule). Configuration is applied on a channel-by-channel basis. These rules are also considered in the creation of new Parser Aligned Blocks, however, note that gaps greater than the threshold setting on the channel will still trigger the opening of additional Parser Aligned Blocks, regardless of the message classification rule. This will be evident by the expected message being split into multiple parser aligned blocks.

NOTE: Additional packetized messages that contain one zero value byte can occur when there is noise on the incoming signal lines. These may be marked as errors, depending on the parity and error detection settings.

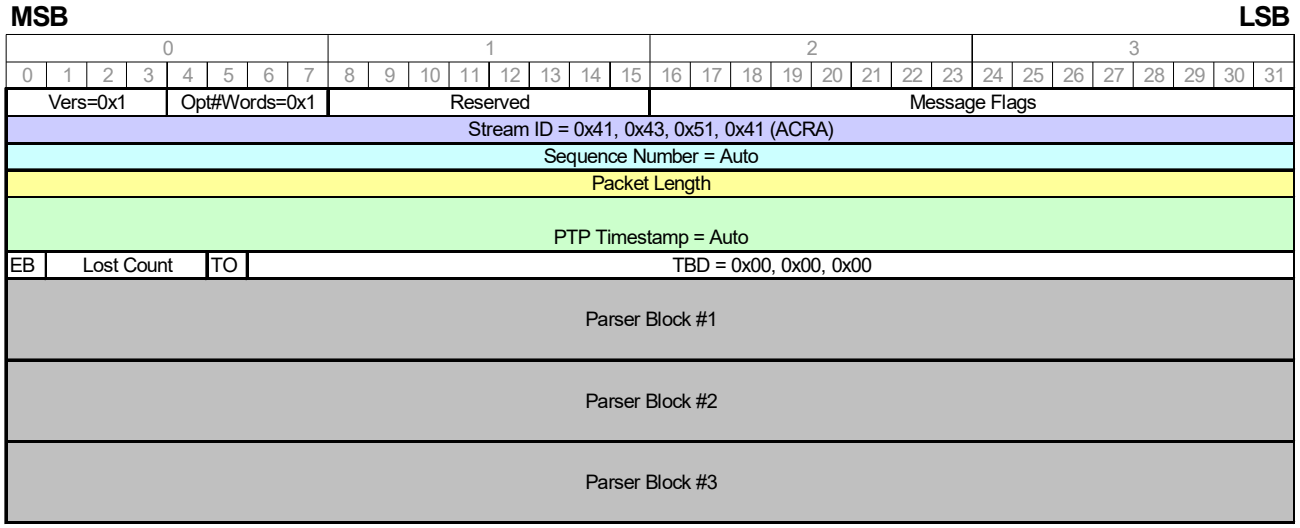

Figure 4: Generalized parser-aligned iNET-X packet

Parser blocks example

UART message parser block (10 characters of data)

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
Er	Error Code							Quad Bytes=5								Message Count								TBD				Bus ID			
Elapsed Time																															
TBD	Cn		P=0					TBD								Data #1								Data #2							
Data #3								Data #4								Data #5								Data #6							
Data #7								Data #8								Data #9								Data #10							

UART message parser block (5 characters of data with 1 byte padding)

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
Er	Error Code							Quad Bytes=4								Message Count								TBD				Bus ID			
Elapsed Time																															
TBD	Cn		P=1					TBD								Data #1								Data #2							
Data #3								Data #4								Data #5								Padding							

UART Message Parser Block (8 characters of data with 2 bytes padding)

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
Er	Error Code							Quad Bytes=5								Message Count								TBD				Bus ID			
Elapsed Time																															
TBD	Cn		P=2					TBD								Data #1								Data #2							
Data #3								Data #4								Data #5								Data #6							
Data #7								Data #8								Padding								Padding							

UART Message Parser Block (13 characters of data (7 bits per character), 1 byte padding)

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
Er	Error Code							Quad Bytes=6								Message Count								TBD				Bus ID			
Elapsed Time																															
TBD	Cn		P=1					TBD								Data #1								Data #2							
	Data #3								Data #4								Data #5								Data #6							
	Data #7								Data #8								Data #9								Data #10							
	Data #11								Data #12								Data #13								Padding							

P = Number of padding bytes added to complete final quadbyte to 32 bits
Cn = 0 0 Complete message
1 0 First fragment of a message that also continues in next packet
0 1 Last fragment of the message continued from last packet and is now complete
1 1 Middle fragment, message continues from last packet and is not finished more in the next packet

UART = Universal Asynchronous Receiver Transmitter)

Figure 5: Parser block formats used in packetizer

Chapter 10 Packetizer format

The IRIG-106 standard is controlled by the Range Commanders Council (RCC) and is a comprehensive telemetry standard for aeronautical applications. Chapter 10 of the IRIG Standard defines the digital Recording Standard.

The Chapter 10 UDP transfer header used on the AXN/UBM/401 is compliant with Format 1, UDP transfer header as defined in Section 10.3.9.1.2 of Telemetry Standards, IRIG Standard 106-17, Chapter 10, July 2017.

The Chapter 10 packetizer format used on the AXN/UBM/401 is compliant with Telemetry Standards, IRIG Standard 106-17, Chapter 10, July 2017, Section 10.6.

http://www.irig106.org/docs/106-17/chapter10.pdf

Section 10.6 of the Telemetry Standards, IRIG Standard 106-17, Chapter 10, July 2017 indicates that the data must be formatted in accordance with Telemetry Standards, IRIG Standard 106-17, Chapter 11, July 2017,

http://www.irig106.org/docs/106-17/chapter11.pdf

The UART Data Packets definition can be found in Section 11.2.12.

For a detailed description on iNET-X packets, see *TEC/NOT/067 - IENA and iNET-X packet payload formats*.

Supported AXN/BCU/40x controller

The AXN/MBM/401 does not function with an AXN/BCU/401 controller; instead it must be used with an AXN/BCU/402 controller. Contact Curtiss-Wright support (acra-support@curtisswright.com) to upgrade an AXN/BCU/401 to an AXN/BCU/402.

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.

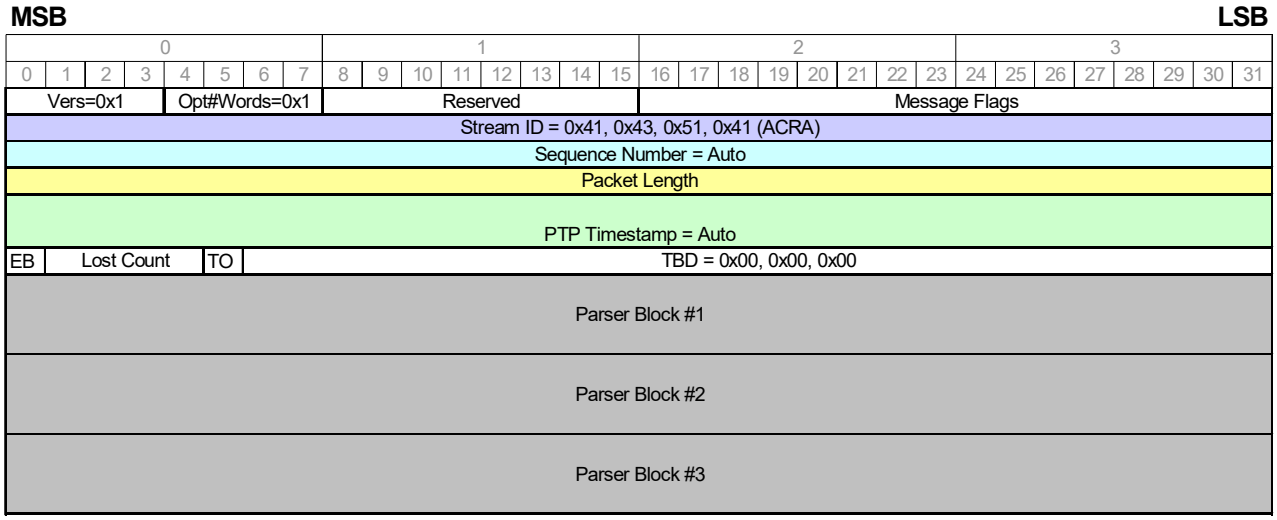

Figure 6: Generalized parser-aligned iNET-X packet

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.

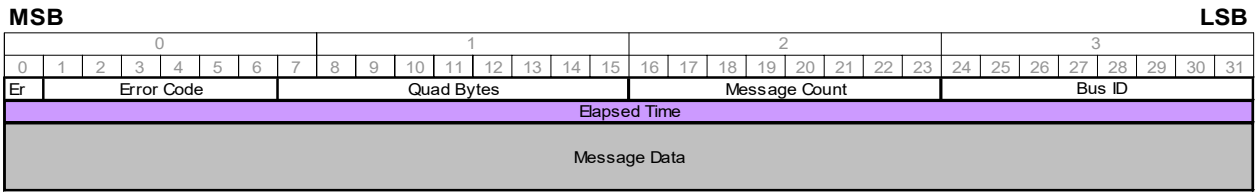

Figure 7: iNET-X parser block

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]: Error code.
  - 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.

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 4	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 the previous table), which are set in the parser information word, indicate the cause of the error.

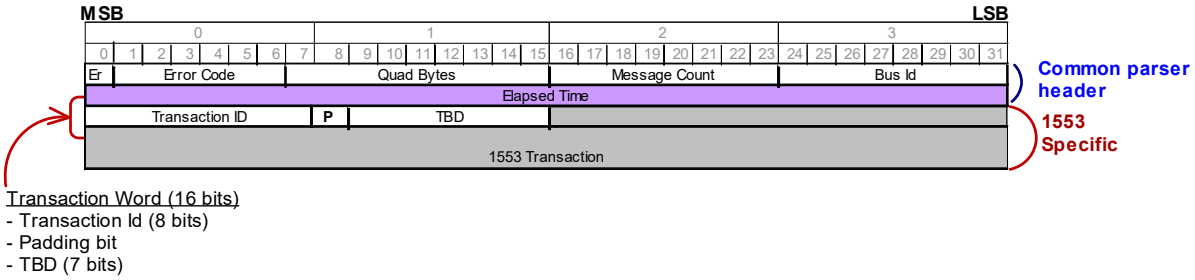

Figure 8: MIL-STD-1553 transaction iNET-X parser-aligned packet

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.

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 4 on page 12
  - 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. Figure 9 on page 13 and Figure 10 on page 13 illustrate individual MIL-STD-1553 BC -> RT transactions, with and without padding.

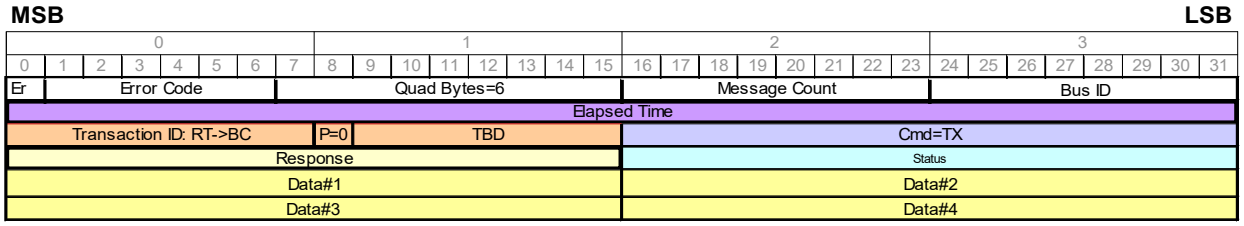

Figure 9: MIL-STD-1553 transaction iNET-X parser-aligned message without padding

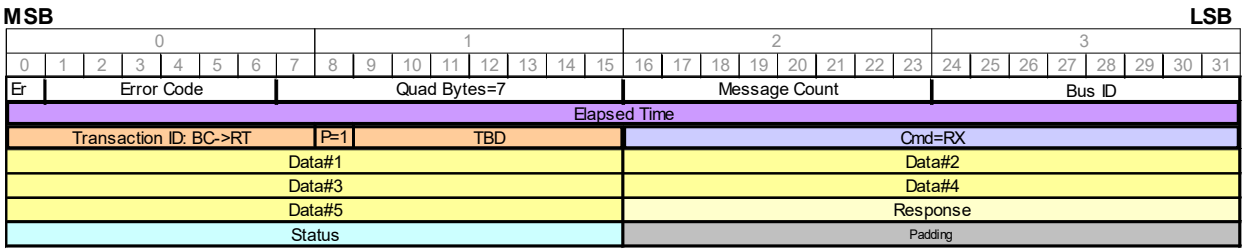

Figure 10: MIL-STD-1553 transaction iNET-X parser-aligned message with padding

Figure 11 on page 14 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.

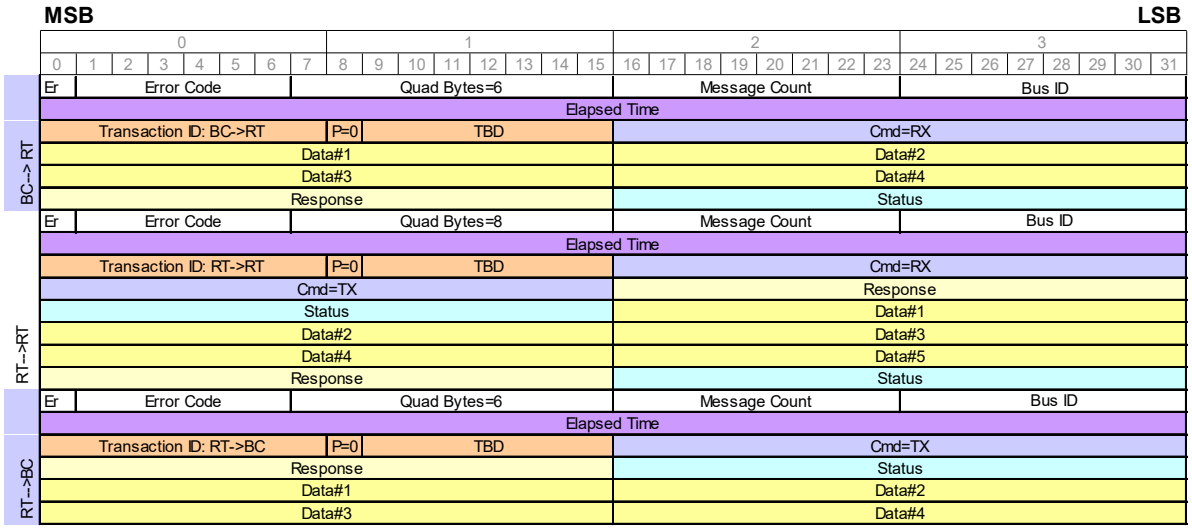

Figure 11: iNET-X parser-aligned payload for multiple MIL-STD-1553 transactions

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).

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

TABLE 5	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)
(packets per second)	150
Total number of MIL-STD-1553 transactions per packet	60 transactions assuming 14 bytes per transaction
bit-rate (kbps)	1816.8

TABLE 6	Error codes		
ERROR CODE	DESCRIPTION	ERROR CODE	DESCRIPTION
0_{16}	Reserved for future use.	20_{16}	Expected STS was a data word.
1_{16}	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.
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.

TABLE 6	Error codes (continued)		
ERROR CODE	DESCRIPTION	ERROR CODE	DESCRIPTION
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.

Connector pinout of the AXN/UBM/401

PIN	NAME	SEE SPECIFICATIONS TABLE	COMMENT
1	DATA(0)+	RS-422/RS-485 inputs, RS-232 inputs	Data in
2	DATA(0)-	RS-422/RS-485 inputs	Data in; connect to pin 19 for 120Ω termination
3	DATA(1)+	RS-422/RS-485 inputs, RS-232 inputs	Data in
4	DATA(1)-	RS-422/RS-485 inputs	Data in; connect to pin 21 for 120Ω termination
5	DATA(2)+	RS-422/RS-485 inputs, RS-232 inputs	Data in
6	DATA(2)-	RS-422/RS-485 inputs	Data in; connect to pin 23 for 120Ω termination
7	DATA(3)+	RS-422/RS-485 inputs, RS-232 inputs	Data in
8	DATA(3)-	RS-422/RS-485 inputs	Data in; connect to pin 25 for 120Ω termination
9	DATA(4)+	RS-422/RS-485 inputs, RS-232 inputs	Data in
10	DATA(4)-	RS-422/RS-485 inputs	Data in; connect to pin 27 for 120Ω termination
11	DATA(5)+	RS-422/RS-485 inputs, RS-232 inputs	Data in
12	DATA(5)-	RS-422/RS-485 inputs	Data in; connect to pin 29 for 120Ω termination
13	DATA(6)+	RS-422/RS-485 inputs, RS-232 inputs	Data in
14	DATA(6)-	RS-422/RS-485 inputs	Data in; connect to pin 31 for 120Ω termination
15	TERM(7)	RS-422/RS-485 inputs	Connect to pin 16 for 120Ω termination
16	DATA(7)-	RS-422/RS-485 inputs	Data in; connect to pin 15 for 120Ω termination
17	DATA(8)-	RS-422/RS-485 inputs	Data in; connect to pin 34 for 120Ω termination
18	GND	Internal ground	
19	TERM(0)	RS-422/RS-485 inputs	Connect to pin 2 for 120Ω termination
20	TERM(9)	RS-422/RS-485 inputs	Connect to pin 38 for 120Ω termination
21	TERM(1)	RS-422/RS-485 inputs	Connect to pin 4 for 120Ω termination
22	TERM(10)	RS-422/RS-485 inputs	Connect to pin 40 for 120Ω termination
23	TERM(2)	RS-422/RS-485 inputs	Connect to pin 6 for 120Ω termination
24	TERM(11)	RS-422/RS-485 inputs	Connect to pin 42 for 120Ω termination
25	TERM(3)	RS-422/RS-485 inputs	Connect to pin 8 for 120Ω termination
26	TERM(12)	RS-422/RS-485 inputs	Connect to pin 44 for 120Ω termination
27	TERM(4)	RS-422/RS-485 inputs	Connect to pin 10 for 120Ω termination
28	TERM(13)	RS-422/RS-485 inputs	Connect to pin 46 for 120Ω termination
29	TERM(5)	RS-422/RS-485 inputs	Connect to pin 12 for 120Ω termination
30	TERM(14)	RS-422/RS-485 inputs	Connect to pin 48 for 120Ω termination
31	TERM(6)	RS-422/RS-485 inputs	Connect to pin 14 for 120Ω termination
32	TERM(15)	RS-422/RS-485 inputs	Connect to pin 50 for 120Ω termination
33	DATA(7)+	RS-422/RS-485 inputs, RS-232 inputs	Data in
34	TERM(8)	RS-422/RS-485 inputs	Connect to pin 17 for 120Ω termination
35	DATA(8)+	RS-422/RS-485 inputs, RS-232 inputs	Data in
36	GND	Internal ground	
37	DATA(9)+	RS-422/RS-485 inputs, RS-232 inputs	Data in
38	DATA(9)-	RS-422/RS-485 inputs	Connect to pin 20 for 120Ω termination
39	DATA(10)+	RS-422/RS-485 inputs, RS-232 inputs	Data in
40	DATA(10)-	RS-422/RS-485 inputs	Connect to pin 22 for 120Ω termination
41	DATA(11)+	RS-422/RS-485 inputs, RS-232 inputs	Data in
42	DATA(11)-	RS-422/RS-485 inputs	Connect to pin 24 for 120Ω termination
43	DATA(12)+	RS-422/RS-485 inputs, RS-232 inputs	Data in
44	DATA(12)-	RS-422/RS-485 inputs	Connect to pin 26 for 120Ω termination
45	DATA(13)+	RS-422/RS-485 inputs, RS-232 inputs	Data in
46	DATA(13)-	RS-422/RS-485 inputs	Connect to pin 28 for 120Ω termination
47	DATA(14)+	RS-422/RS-485 inputs, RS-232 inputs	Data in
48	DATA(14)-	RS-422/RS-485 inputs	Connect to pin 30 for 120Ω termination
49	DATA(15)+	RS-422/RS-485 inputs, RS-232 inputs	Data in
50	DATA(15)-	RS-422/RS-485 inputs	Connect to pin 32 for 120Ω termination
51	GND	Internal ground	
52	CHASSIS	Chassis	

PART NUMBER	DESCRIPTION
AXN/UBM/401	RS-232, RS-422 or RS-485 serial bus parser/packetizer - 16ch

By default, the standard mating connector, CON/KAD/002/CP, is included with each module in the shipment. Its part number will be added to the Confirmation of Order unless an alternative option is specified (see the *Cables* data sheet).

Revision history

REVISION	DIFFERENCES	STATUS
AXN/UBM/401	First release	Recommended for new programs

Supporting software

SOFTWARE	DIFFERENCES
DAS Studio 3	User interface for setup and management of data acquisition, network switches, recorders and ground stations in an integrated environment

Related products

MODULE	DETAILS
GS Works 9	Real-time and post-test data visualization and analysis software

Related documentation

DOCUMENT	DETAILS
DOC/MAN/030	DAS Studio 3 User Manual
DOC/HBK/008	Environmental Qualification Handbook for Axon Products.

This page is intentionally blank