

This paper discusses the following topics:

- “49.1 Overview” on page 1
- “49.2 ABI protocol overview” on page 1
- “49.3 ACE-CCDL protocol overview” on page 2
- “49.4 ACB protocol overview” on page 2
- “49.5 Configuring the KAD/UBM/104 to parse and packetize traffic” on page 8

49.1 Overview

The KAD/UBM/104 is an 8-channel Actuator Bus Interface (ABI), Actuator Control Equipment-Cross Channel Data Link (ACE-CCDL), or Actuator Control Bus (ACB) serial bus monitor and packetizer. You can select which mode to use on a module-wide level and then coherently parse up to 255 messages, while packetizing all traffic on a channel-by-channel basis.

49.2 ABI protocol overview

The ABI protocol uses a 1Mbps RS-485 Bi-Phase-S encoded physical layer.

There are two types of messages used in the ABI protocol: ABI write messages and ABI read messages.

49.2.1 ABI write messages

ABI write messages are made up of 12 bytes of header, followed by up to 255 bytes of data, followed by 2 bytes of Cyclic Redundancy Check (CRC). ABI write messages are preceded and followed by an idle pattern of up to 7 μ s of Bi-Phase-S 1s.

The following figure shows the format of an ABI write message.

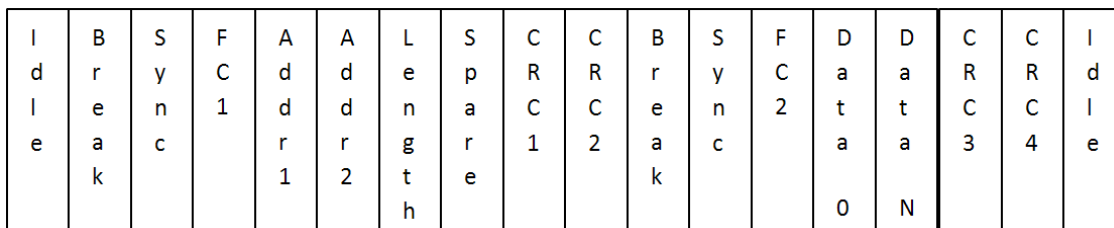


Figure 49-1: ABI write message format

The Break field is a 12-bit 0 pattern preceded by a 0 start bit and completed by a 1 stop bit.

The Sync field is an 8-bit value of 0x55 preceded by a 0 start bit and completed by a 1 stop bit.

The function code (FC1) for ABI write messages is 0x11 preceded by a 0 start bit and completed by a 1 stop bit.

The address field is made up of 2 bytes (1 byte each for Addr1 and Addr2), each preceded by a 0 start bit and completed by a 1 stop bit.

The Length field is an 8-bit value preceded by a 0 start bit and completed by a 1 stop bit that describes the length of the data that follows.

The Spare field reads 0x00 preceded by a 0 start bit and completed by a 1 stop bit.

The first CRC field (CRC1) is the high byte of CRC calculated on the function code field (preamble) and the Data field (preamble), using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

The second CRC field (CRC2) is the low byte of CRC calculated on the function code field (preamble) and the Data field (preamble), using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

The second function code (FC2) for ABI write messages is 0xFE preceded by a 0 start bit and completed by a 1 stop bit.

All Data fields are preceded by a 0 start bit and completed by a 1 stop bit.

The third CRC field (CRC3) is the high byte of CRC calculated on the function code field and the Data field, using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

The fourth CRC field (CRC4) is the low byte of CRC calculated on the function code field and the Data field, using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

49.2.2 ABI read messages

ABI read messages look just like write messages on the physical layer except for the following:

- They have a different function code, which is 0x01 preceded by a 0 start bit and completed by a 1 stop bit.
- They have a gap of between 6.4 μ s to 8.8 μ s after the second function code and again after the fourth CRC.

NOTE: The KAD/UBM/104 does not flag CRC errors within ABI messages. CRC data is treated as packet payload data.

The KAD/UBM/104 uses a combination of function code and address values to uniquely identify each ABI message.

49.3 ACE-CCDL protocol overview

Write messages in the ACE-CCDL protocol are identical to the write messages in the ABI protocol. There is no read message type in the ACE-CCDL protocol.

When the KAD/UBM/104 is programmed to operate in ACE-CCDL mode, all messages that do not have the ABI write function code (0x11) are packetized and flagged as error messages.

The KAD/UBM/104 uses a combination of function code and address values to uniquely identify each ACE-CCDL message.

49.4 ACB protocol overview

The ACB protocol uses a 1Mbps RS-485 Bi-Phase-M encoded physical layer. This is basically the inverse of the ABI line code.

There are three message types in the ACB protocol, each of which works in a master-request followed by a slave-response format.

There are always two types of slave response to each message type: good response or error response.

The three message types are:

“49.4.1 ACB Read Holding Registers messages” on page 2

“49.4.2 ACB Write Multiple Registers messages” on page 4

“49.4.3 ACB Read/Write Multiple Registers messages” on page 6

49.4.1 ACB Read Holding Registers messages

49.4.1.1 Master request

The request from master part of these ACB Read Holding Registers messages is an 8-byte message preceded by a minimum gap of 40 μ s where the bus is reading Bi-Phase-M 1s and is followed by a gap of 40 – 50 μ s of Bi-Phase-M 1s.

S	S	F	R	R	Q	Q	C	C	E
D	L	C	D	D	R	R	R	R	D
E	A	1	A	A	D	D	1	2	E
L	V		D	D					L
I	E		1	2					I
M					1	2			M

Figure 49-2: Request from master message

SDELIM = 40 – 50 μ s of Bi-Phase-M 1s.

SLAVE = Slave address identifier of slave node on the ACB bus, preceded by a 0 start bit and completed by a 1 stop bit.

FC1 = Function code of ACB Read Holding Registers messages, reads 0x03, preceded by a 0 start bit and completed by a 1 stop bit.

RDAD1 = High byte of the read address of the start memory location on the slave node the ACB master is requesting data from, preceded by a 0 start bit and completed by a 1 stop bit.

RDAD2 = Low byte of the read address of the start memory location on the slave node the ACB master is requesting data from, preceded by a 0 start bit and completed by a 1 stop bit.

QRD1 = High byte of quantity of reads of the number of addresses to be read from the slave, starting at the above read address, preceded by a 0 start bit and completed by a 1 stop bit.

QRD2 = Low byte of quantity of reads of the number of addresses to be read from the slave, starting at the above read address, preceded by a 0 start bit and completed by a 1 stop bit.

CRC1 = High byte of the CRC for the ACB read holding request, calculated across all fields of the request, excluding the delimiters using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

CRC2 = Low byte of the CRC for the ACB read holding request, calculated across all fields of the request, excluding the delimiters using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

EDELIM = Minimum 40 μ s of Bi-Phase-M 1s.

49.4.1.2 Good data response

A good data response from a slave may contain up to 250 bytes of data, preceded by 3 bytes of header, and followed by 2 bytes of CRC; the full message is appended by message delimiter patterns of minimum 40 μ s of Bi-Phase-M 1s.

S	S	F	L	D	D	D	C	C	E
D	L	C	e	a	a	a	R	R	D
E	A	2	n	t	t	t	C	C	E
L	V		g	a	a	a	1	2	L
I	E		t	0	1	N			I
M			h						M

Figure 49-3: Good data response message

SDELIM = Minimum 40 μ s of Bi-Phase-M 1s (same delimiter as EDELIM described in the request; the end delimiter for the request is the start delimiter for the response).

SLAVE = Slave address identifier of slave node on the ACB bus, preceded by a 0 start bit and completed by a 1 stop bit.

FC2 = Function code of ACB Read Holding Registers messages, reads 0x03, preceded by a 0 start bit and completed by a 1 stop bit.

Length = Count of the number of bytes of data that follows, preceded by a 0 start bit and completed by a 1 stop bit.

Data 0 – N = Data bytes, preceded by a 0 start bit and completed by a 1 stop bit.

CRC1 = High byte of the CRC for the ACB read holding request, calculated across all fields of the response, excluding the delimiters using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

CRC2 = Low byte of the CRC for the ACB Read Holding Registers message, calculated across all fields of the response, excluding the delimiters using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

EDELIM = Minimum 40 μ s of Bi-Phase-M 1s.

49.4.1.3 Error Response

An error response from a slave is a 5-byte message appended by message delimiter patterns of minimum 40 μs Bi-Phase-M 1s.

S	S	F	E	C	C	E
D	L	C	R	R	R	D
E	A	2	R	C	C	E
L	V		C	1	2	L
I	E					I
M						M

Figure 49-4: Error response message

SDELIM = Minimum 40 μs of Bi-Phase-M 1s (same delimiter as EDELIM described in the request; the end delimiter for the request is the start delimiter for the response).

SLAVE = Slave address identifier of slave node on the ACB bus, preceded by a 0 start bit and completed by a 1 stop bit.

FC2 = Function code of ACB Read Holding Registers messages, reads 0x83, preceded by a 0 start bit and completed by a 1 stop bit.

ERRC = Error code, ACB protocol error code, preceded by a 0 start bit and completed by a 1 stop bit. 0x01 = illegal function, 0x02 = illegal data address, 0x03 = illegal data value, 0x04 = slave device failure.

CRC1 = High byte of the CRC for the ACB read holding request, calculated across all fields of the response, excluding the delimiters using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

CRC2 = Low byte of the CRC for the ACB Read Holding Registers message, calculated across all fields of the response, excluding the delimiters using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

EDELIM = Minimum 40 μs of Bi-Phase-M 1s.

49.4.2 ACB Write Multiple Registers messages

49.4.2.1 Master request

The request from the master part of these ACB Write Multiple Registers messages can contain up to 246 bytes of data preceded by a 7-byte header and followed by 2 bytes of CRC.

S	S	F	W	W	Q	Q	D	D	C	C	E
D	L	C	R	R			A	A	R	R	D
E	A	1	A	A	W	W	T	T	C	C	E
L	V		D	D	R	R	A	A			L
I	E								1	2	I
M			1	2	1	2	0	N			M

Figure 49-5: Request from master message

SDELIM = 40 – 50 μs of Bi-Phase-M 1s.

SLAVE = Slave address identifier of slave node on the ACB bus, preceded by a 0 start bit and completed by a 1 stop bit.

FC1 = Function code of ACB Write Multiple Registers messages, reads 0x10, preceded by a 0 start bit and completed by a 1 stop bit.

WRAD1 = High byte of the write address of the start memory location on the slave node the ACB master is writing data to, preceded by a 0 start bit and completed by a 1 stop bit.

WRAD2 = Low byte of the write address of the start memory location on the slave node the ACB master is writing data to, preceded by a 0 start bit and completed by a 1 stop bit.

QWR1 = High byte of quantity of writes of the number of addresses to be written to on the slave, starting at the above write address, preceded by a 0 start bit and completed by a 1 stop bit.

QWR2 = Low byte of quantity of writes of the number of addresses to be written to on the slave, starting at the above write address, preceded by a 0 start bit and completed by a 1 stop bit.

Data 0 – N = Data bytes preceded by a 0 start bit and completed by a 1 stop bit.

CRC1 = High byte of the CRC for the ACB write request calculated across all fields of the request, excluding the delimiters using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

CRC2 = Low byte of the CRC for the ACB write request calculated across all fields of the request, excluding the delimiters using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

EDELIM = Minimum 40 μ s of Bi-Phase-M 1s.

49.4.2.2 Good data response

A good data response is made up of 8 bytes prepended and appended by message delimiter patterns of up to 50 μ s of Bi-Phase-M 1s.

S	S	F	W	W	Q	Q	C	C	E
D	L	C	R	R	W	W	R	R	D
E	A	2	A	A	R	R	C	C	E
L	V		D	D	1	2	1	2	L
I	E								I
M			1	2					M

Figure 49-6: Good data response message

SDELIM = Minimum 40 μ s of Bi-Phase-M 1s (same delimiter as EDELIM described in the request; the end delimiter for the request is the start delimiter for the response).

SLAVE = Slave address identifier of slave node on the ACB bus, preceded by a 0 start bit and completed by a 1 stop bit.

FC2 = Function code of ACB Write Multiple Registers messages, reads 0x10, preceded by a 0 start bit and completed by a 1 stop bit.

WRAD1 = High byte of the write address of the start memory location on the slave node the ACB master is writing data to, preceded by a 0 start bit and completed by a 1 stop bit.

WRAD2 = Low byte of the write address of the start memory location on the slave node the ACB master is writing data to, preceded by a 0 start bit and completed by a 1 stop bit.

QWR1 = High byte of quantity of writes of the number of addresses to be written to on the slave, starting at the above write address, preceded by a 0 start bit and completed by a 1 stop bit.

QWR2 = Low byte of quantity of writes of the number of addresses to be written to on the slave, starting at the above write address, preceded by a 0 start bit and completed by a 1 stop bit.

CRC1 = High byte of the CRC for the ACB write request calculated across all fields of the request, excluding the delimiters using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

CRC2 = Low byte of the CRC for the ACB write request calculated across all fields of the request, excluding the delimiters using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

EDELIM = Minimum 40 μ s of Bi-Phase-M 1s.

49.4.2.3 Error response

An error response from a slave is a 5-byte message prepended and appended by message delimiter patterns of up to 50 μ s of Bi-Phase-M 1s.

S	S	F	E	C	C	E
D	L	C	R	R	R	D
E	A	2	R	C	C	E
L	V		C	1	2	L
I	E					I
M						M

Figure 49-7: Error response message

SDELIM = Minimum 40 μs of Bi-Phase-M 1s (same delimiter as EDELIM described in the request; the end delimiter for the request is the start delimiter for the response).

SLAVE = Slave address identifier of slave node on the ACB bus, preceded by a 0 start bit and completed by a 1 stop bit.

FC2= Function code of ACB write error response, reads 0x90, preceded by a 0 start bit and completed by a 1 stop bit.

ERRC = Error code, ACB protocol error code, preceded by a 0 start bit and completed by a 1 stop bit. 0x01 = illegal function, 0x02 = illegal data address, 0x03 = illegal data value, 0x04 = slave device failure.

CRC1 = High byte of the CRC for the ACB write error response calculated across all fields of the response, excluding the delimiters using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

CRC2 = Low byte of the CRC for the ACB write error response calculated across all fields of the response, excluding the delimiters using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

EDELIM = Minimum 40 μs of Bi-Phase-M 1s.

49.4.3 ACB Read/Write Multiple Registers messages

ACB Read/Write Multiple Registers messages consist of a request that contains up to 246 bytes of write data and a response that contains up to 250 bytes of read data.

49.4.3.1 Master request

The request from master part of these ACB read/write messages can contain up to 246 bytes of data, preceded by an 11-byte header, followed by 2 bytes of CRC, and book-ended by message delimiter patterns of up to 50 μs of Bi-Phase-M 1s.

S	S	F	R	R	Q	Q	W	W	Q	Q	D	D	C	C	E
D	L	C	D	D	R	R	R	R	W	W	A	A	R	R	D
E	A	1	A	A	D	D	A	A	R	R	T	T	C	C	E
L	V		D	D			D	D			A	A	1	2	L
I	E														I
M			1	2	1	2	1	2	1	2	0	N			M

Figure 49-8: Request from master message

SDELIM = 40 – 50 μs of Bi-Phase-M 1s.

SLAVE = Slave address identifier of slave node on the ACB bus, preceded by a 0 start bit and completed by a 1 stop bit.

FC1 = Function code of ACB read/write messages, reads 0x17, preceded by a 0 start bit and completed by a 1 stop bit.

RDAD1 = High byte of the read address the start memory location on the slave node the ACB master is requesting data from, preceded by a 0 start bit and completed by a 1 stop bit.

RDAD2 = Low byte of the read address the start memory location on the slave node the ACB master is requesting data from, preceded by a 0 start bit and completed by a 1 stop bit.

QRD1 = High byte of quantity of reads of the number of addresses to be read from the slave, starting at the above read address, preceded by a 0 start bit and completed by a 1 stop bit.

QRD2 = Low byte of quantity of reads of the number of addresses to be read from the slave, starting at the above read address, preceded by a 0 start bit and completed by a 1 stop bit.

WRAD1 = High byte of the write address of the start memory location on the slave node the ACB master is writing data to, preceded by a 0 start bit and completed by a 1 stop bit.

WRAD2 = Low byte of the write address of the start memory location on the slave node the ACB master is writing data to, preceded by a 0 start bit and completed by a 1 stop bit.

QWR1 = High byte of quantity of writes of the number of addresses to be written to on the slave, starting at the above write address, preceded by a 0 start bit and completed by a 1 stop bit.

QWR2 = Low byte of quantity of writes of the number of addresses to be written to on the slave, starting at the above write address, preceded by a 0 start bit and completed by a 1 stop bit.

Data 0 – N = Data bytes preceded by a 0 start bit and completed by a 1 stop bit.

CRC1 = High byte of the CRC for the ACB read/write request, calculated across all fields of the request, excluding the delimiters using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

CRC2 = Low byte of the CRC for the ACB read/write request, calculated across all fields of the request, excluding the delimiters using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

EDELIM = Minimum 40 μ s of Bi-Phase-M 1s.

49.4.3.2 Good data response

A good data response may contain up to 250 bytes of read data preceded by 3 bytes of header, prepended and appended by message delimiter patterns of up to 50 μ s of Bi-Phase-M 1s.

S	S	F	R	D	D	C	C	E
D	L	C	D	A	A	R	R	D
E	A	2		T	T	C	C	E
L	V		C	A	A	1	2	L
I	E		N					I
M			T	O	N			M

Figure 49-9: Good data response message

SDELIM = Minimum 40 μ s of Bi-Phase-M 1s (same delimiter as EDELIM described in the request; the end delimiter for the request is the start delimiter for the response).

SLAVE = Slave address identifier of slave node on the ACB bus, preceded by a 0 start bit and completed by a 1 stop bit.

FC2 = Function code of ACB read/write messages reads 0x03, preceded by a 0 start bit and completed by a 1 stop bit.

Data 0 – N = Data bytes preceded by a 0 start bit and completed by a 1 stop bit.

CRC1 = High byte of the CRC for the ACB read/write good response, calculated across all fields of the response, excluding the delimiters using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

CRC2 = Low byte of the CRC for the ACB read/write good response, calculated across all fields of the response, excluding the delimiters using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

EDELIM = Minimum 40 μ s of Bi-Phase-M 1s.

49.4.3.3 Error Response

An error response from a slave is a 5-byte message, prepended and appended by message delimiter patterns of up to 50 μ s of Bi-Phase-M 1s.

S	S	F	E	C	C	E
D	L	C	R	R	R	D
E	A	2	R	C	C	E
L	V		C	1	2	L
I	E					I
M						M

Figure 49-10: Error response message

SDELIM = Minimum 40 μ s of Bi-Phase-M 1s (same delimiter as EDELIM described in the request; the end delimiter for the request is the start delimiter for the response).

SLAVE = Slave address identifier of slave node on the ACB bus, preceded by a 0 start bit and completed by a 1 stop bit.

FC2= Function code of ACB read/write error response, reads 0x90, preceded by a 0 start bit and completed by a 1 stop bit.

ERRC = Error code, ACB protocol error code, preceded by a 0 start bit and completed by a 1 stop bit. 0x01 = illegal function, 0x02 = illegal data address, 0x03 = illegal data value, 0x04 = slave device failure.

CRC1 = High byte of the CRC for the ACB read/write error response, calculated across all fields of the response, excluding the delimiters using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

CRC2 = Low byte of the CRC for the ACB read/write error response, calculated across all fields of the response, excluding the delimiters using $X^{16} + X^{12} + X^5 + 1$ preceded by a 0 start bit and completed by a 1 stop bit.

The KAD/UBM/104 uses a combination of slave address, function code, and address values to uniquely identify each ACB message.

EDELIM = Minimum 40 μ s of Bi-Phase-M 1s.

49.5 Configuring the KAD/UBM/104 to parse and packetize traffic

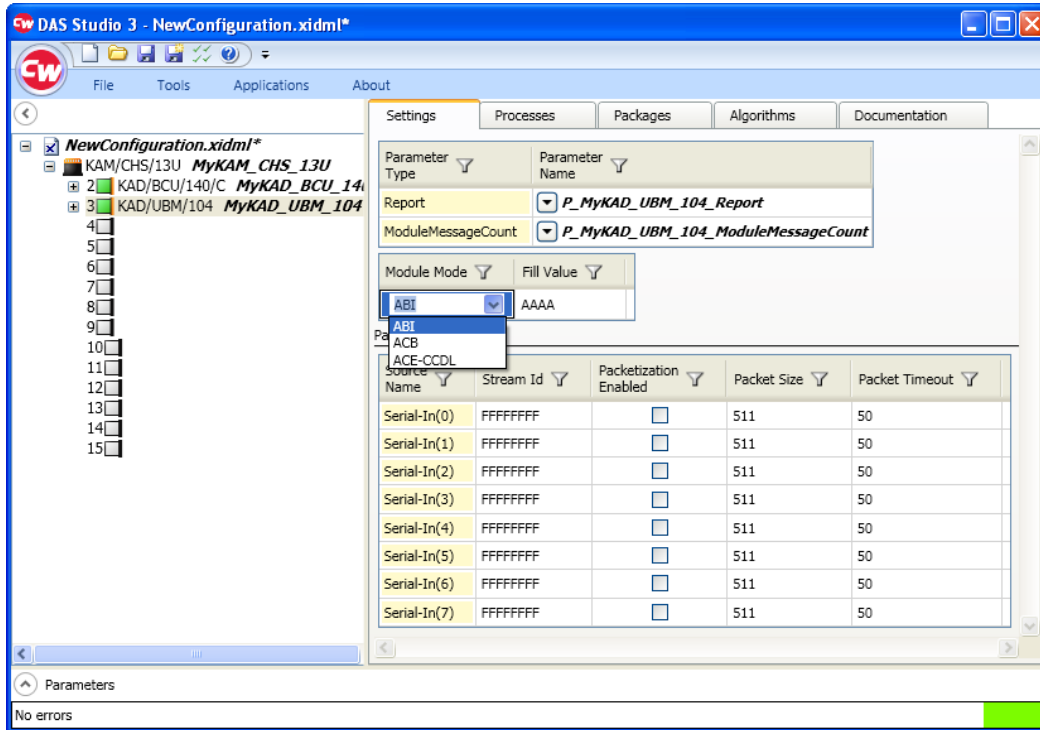
DAS Studio 3 is used to create a configuration which contains the various elements which make up your data acquisition system. You then use this configuration file to manage and program these elements. To see how hardware is represented in the DAS Studio 3 graphical user interface, see Figure 1 in the DAS Studio 3 User Manual.

NOTE: In this technical note, the terms *message* and *package* have the same meaning. The term package is used consistently in the software GUI interface, while the term message is used in the hardware specifications.

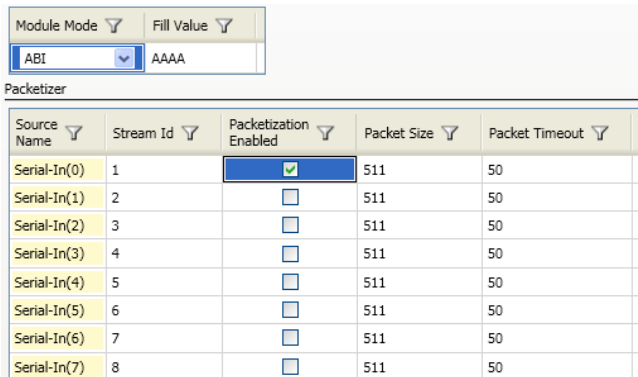
49.5.1 Setting up the packetizer

For this example, the configuration file contains a KAM/CHS/13U (chassis), a KAD/BCU/140/C (controller module), and the KAD/UBM/104.

1. Ensure the KAD/UBM/104 is selected in the Navigator on the left of the screen so that it is in context in the **Settings** tab.



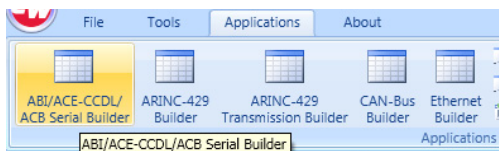
2. Click the drop-down arrow in the **Module Mode** field and select a mode.
3. Assign a unique stream ID to each channel and enable **Packetization Enabled** for channels of interest.



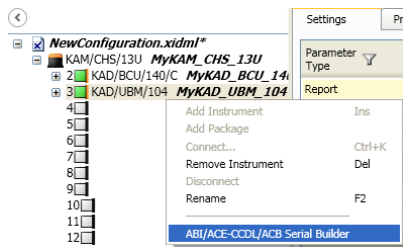
49.5.2 Setting up the parser

Refer to the following to define rules for the module to identify and parse data:

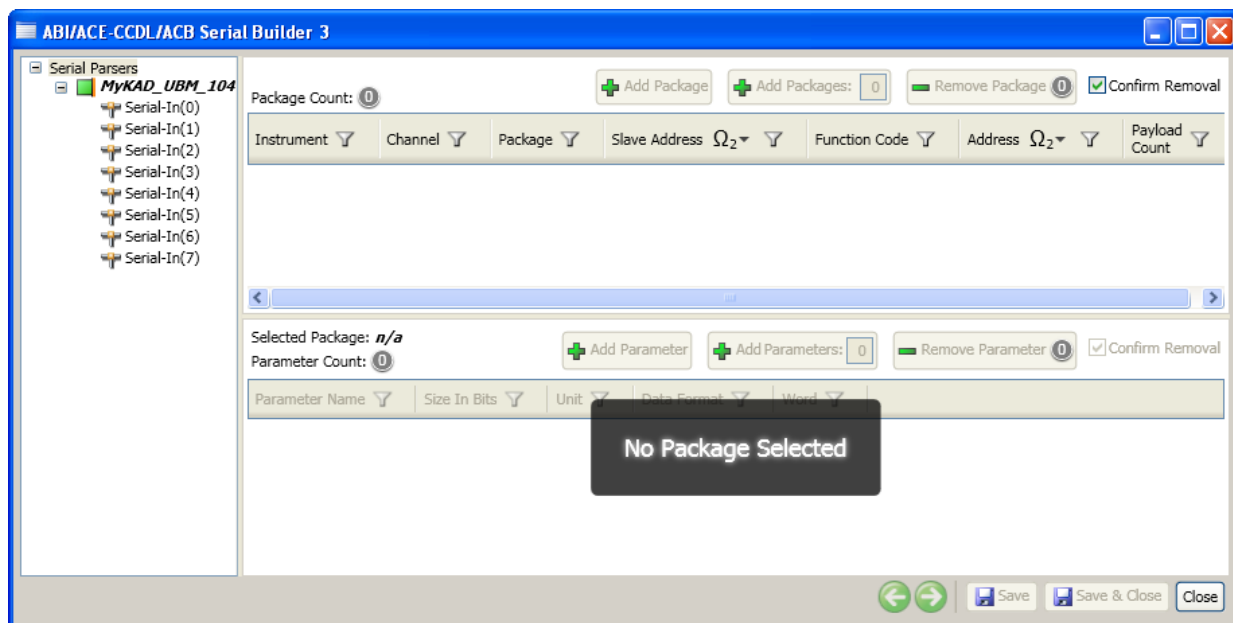
1. On the **Applications** menu, click **ABI/ACE-CCDL/ACB Serial Builder**.



ABI/ACE-CCDL/ACB Serial Builder may also be opened by right-clicking on the KAD/UBM/104 in the Navigator.



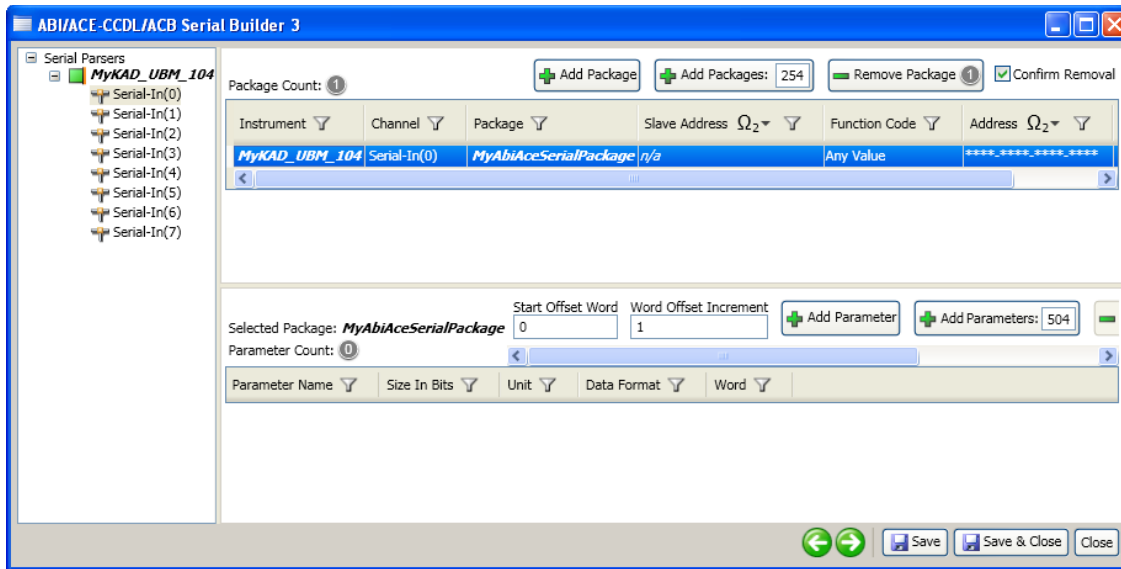
The **ABI/ACE-CCDL/ACB Serial Builder 3** dialog box opens.



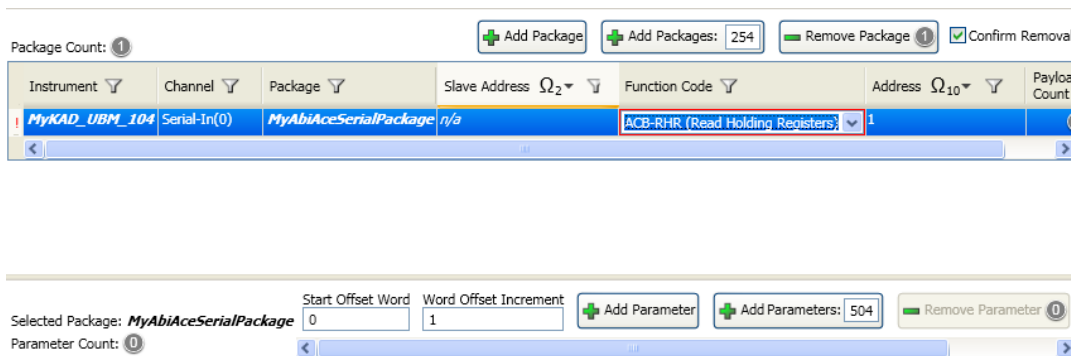
For dialog box navigation, see “Builder application GUI overview” in the *DAS Studio 3 User Manual*.

2. To define a parsing rule for ABI messages, select a channel in the Navigator and then click **Add Package**. To add multiple packages, type the number of packages in the **Add Package** field and then click **Add Packages**. (The maximum number of packages you can add is shown in the field.)

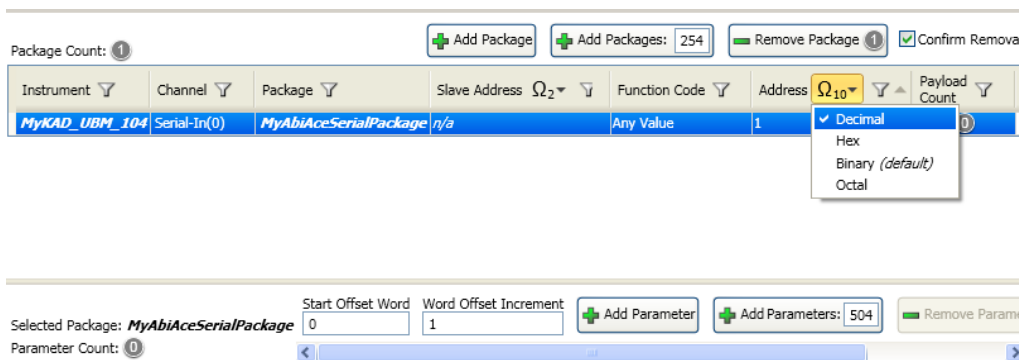
The package is added.



- Click the **Function Code** drop-down arrow (the drop-down arrow is only visible after you click twice in the field) and select the type of package you want.
If an invalid package type is selected for the current module mode setting (such as **ACB Read Holding Registers** when the module is in ABI mode) the **Function Code** field is highlighted in red to indicate the error and the **Save** function is disabled until the error is resolved.



- Type a unique identifier for the package in the **Address** field.
By default the **Address** field is ******_****_****_******, where ***** represents a binary wild card bit. This 16-bit value can be changed to hexadecimal, decimal, or octal by replacing the default string with 0000000000000001. This removes the wild cards and activates the **Hex**, **Decimal**, or **Octal** options. Wild cards are only represented in **Binary** mode.



- To add a parameter to the parsed package, click **Add Parameter**.
To add multiple parameters, type the number of parameters in the **Add Parameter** field and then click **Add**

Parameters. (The maximum number of parameters you can add is shown in the field.)

Package Count: 1

Instrument: MyKAD_UBM_104, Channel: Serial-In(0), Package: MyAbiAceSerialPackage, Slave Address: n/a, Function Code: Any Value, Address: *****, Payload Count: 10

Selected Package: MyAbiAceSerialPackage, Start Offset Word: 0, Word Offset Increment: 1, Add Parameter, Add Parameters: 494, Remove Parameter

Parameter Name	Size In Bits	Unit	Data Format	Word
MyAbiAceSerialPackage.MyParameter	16	BitVector	BitVector	0
MyAbiAceSerialPackage.MyParameter1	16	BitVector	BitVector	1
MyAbiAceSerialPackage.MyParameter2	16	BitVector	BitVector	2
MyAbiAceSerialPackage.MyParameter3	16	BitVector	BitVector	3
MyAbiAceSerialPackage.MyParameter4	16	BitVector	BitVector	4
MyAbiAceSerialPackage.MyParameter5	16	BitVector	BitVector	5
MyAbiAceSerialPackage.MyParameter6	16	BitVector	BitVector	6
MyAbiAceSerialPackage.MyParameter7	16	BitVector	BitVector	7
MyAbiAceSerialPackage.MyParameter8	16	BitVector	BitVector	8
MyAbiAceSerialPackage.MyParameter9	16	BitVector	BitVector	9

Buttons: Save, Save & Close, Close

6. Click **Save & Close** to complete the parser rules setup.

49.5.3 Parsing ACB messages

As stated previously, the KAD/UBM/104 uses a combination of slave address, function code, and address values to uniquely identify each ACB package.

When the Module Mode setting is set to ACB mode (see the following figure), the ABI/ACE-CCDL/ACB Serial Builder reads this setting and makes the Slave Address setting available using a wild card as default (see Figure 49-12 on page 13).

Settings

Parameter Type: Report, Parameter Name: P_MyKAD_UBM_104_Report

ModuleMessageCount: P_MyKAD_UBM_104_ModuleMessageCount

Module Mode: ACB, Fill Value: AAAA

Source Name	Stream Id	Packetization Enabled	Packet Size	Packet Timeout
Serial-In(0)	1	<input checked="" type="checkbox"/>	511	50
Serial-In(1)	2	<input type="checkbox"/>	511	50
Serial-In(2)	3	<input type="checkbox"/>	511	50
Serial-In(3)	4	<input type="checkbox"/>	511	50
Serial-In(4)	5	<input type="checkbox"/>	511	50
Serial-In(5)	6	<input type="checkbox"/>	511	50
Serial-In(6)	7	<input type="checkbox"/>	511	50
Serial-In(7)	8	<input type="checkbox"/>	511	50

Figure 49-11: Module Mode set to ACB

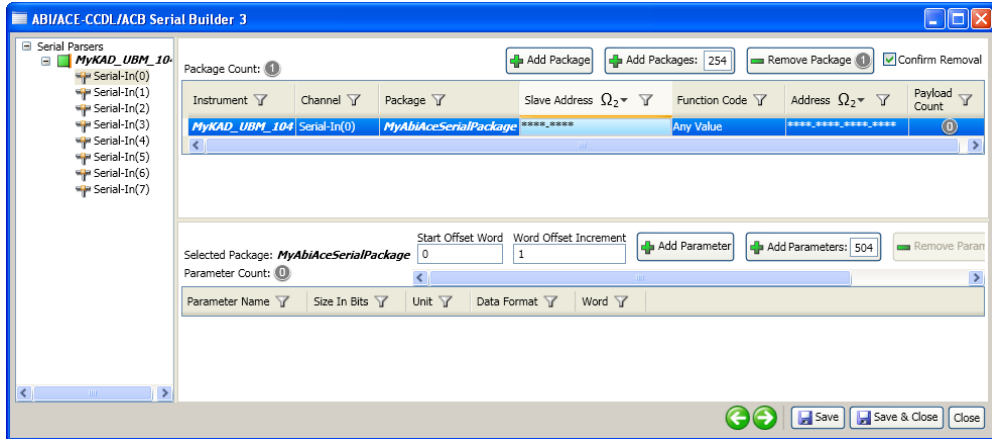


Figure 49-12: Slave Address setting in ACB mode

This Slave Address setting is only available in ACB mode.

This page is intentionally blank