Using DAS Studio 3 to configure the KAD/CBM/103

TEC/NOT/074



This technical note describes how to use DAS Studio 3 to configure the KAD/CBM/103 to parse messages. This paper discusses the following topics:

- "45.1 Overview" on page 1
- "45.2 About the CCDL/MCDL protocol" on page 1
- "45.3 Using DAS Studio 3 to configure the KAD/CBM/103" on page 2

45.1 Overview

The KAD/CBM/103 is a 4-channel Cross Channel Data Link/Motor Controller Data Link (CCDL/MCDL) bus monitor. It can parse up to 127 unique messages per channel, with up to 65 bytes of data per message.

The CCDL/MCDL message structure is shown in the following figure. Messages are separated by gaps that are a minimum of 11 null bits wide. Bytes inside messages are transmitted without gaps.



Figure 45-1: Typical CCDL/MCDL message structure

The header is composed as shown in the following figure.

Bits 15 -11	Bit 10	Bits 9 - 6	Bits 5 - 0
Message ID	Master/slave bit	Freshness counter	Message length

Figure 45-2: Composition of header bytes

Elements of the header are described as follows:

Message ID: a unique 5-bit identifier per message.

Master/slave bit: identifies whether the message came from a master unit (1) or a slave unit (0).

Freshness counter: 4-bit counter which increments every time a particular message ID is sent.

Message length: the number of data bytes contained in the current message.

45.2 About the CCDL/MCDL protocol

The CCDL/MCDL protocol uses a command-reply format. CCDL master devices request data from slave devices, which send replies using the same message ID. Considering the following example where a master device requests data with a message ID of 0, the header breakdown is as follows:

Message ID = 00000

Master bit = 1

Freshness counter = 0000

Message length = ******

When a slave device receives this request, it replies using the same message ID, incrementing the freshness counter by 1. The header breakdown is then:

Message ID = 00000

Master bit = 0



Freshness counter = 0001

Message length = ******

This continues until all required data from the slave has been received by the master.

45.3 Using DAS Studio 3 to configure the KAD/CBM/103

DAS Studio 3 is used to create a configuration which contains the various elements which make up your data acquisition system. You may 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.

45.3.1 Creating a basic configuration

This section describes how to use DAS Studio 3 to create a basic configuration which includes the KAD/CBM/103.

1. On the Quick Access Toolbar or the File menu, click New.



2. To add a chassis which represents the hardware you have connected, right-click on the overview node and click **Add Instrument**.



Instruments Palette opens. For information on Instruments Palette settings, see the DAS Studio 3 User Manual.

- 3. On the **DAU** tab, select the chassis connected and then click **Add**.
- 4. Click + to expand the chassis node.
- Empty slots appear under the chassis indicating where modules can be added.
- 5. To add a controller module which represents the hardware you have connected, right-click on empty slot 2 and click Add Instrument.

Instruments Palette opens.

6. Select a controller module, for example a KAD/BCU/140/C, and click Add.

cw Instruments Palette	(D) ===	and the second		
CW	Ethernet Other PCM	1		
	Name 🍸	Part Reference 🍸	Short Description ${\mathbb Y}$	Long Description 🍸
New Actor Component	MyKAD_BCU_105_D	KAD/BCU/105/D		KAM-500 backplane controller with Ethernet ence
,	MyKAD_BCU_140	KAD/BCU/140		KAM-500 Controller With Ethernet Transmitter ar
6	MyKAD_BCU_140_B	KAD/BCU/140/B		KAM-500 Controller With Ethernet Transmitter ar
Import	MyKAD_BCU_140_C	KAD/BCU/140/C		KAM-500 Controller With Ethernet Transmitter ar
	MyKAD_BCU_140_X1	KAD/BCU/140/X1		KAM-500 Controller With Ethernet Transmitter ar



The module is added to slot 2.



- 7. To add the KAD/CBM/103 you have connected, right-click on an empty slot and click **Add Instrument**. **Instruments Palette** opens.
- 8. On the Bus Monitor tab, select the KAD/CBM/103 and click Add.

(cw Instruments Palette	fearable of	fragmenter og		
1	CW	Analog Bus Monitor Oth	er Switch Transmitt	er	
		Name 🍸	Part Reference 🍸	Short Description ∇	Long Description 🍸
	New ACKA Component	MyKAD/CBM/107	KAD/CBM/107		A
	,	MyKAD_ABM_101	KAD/ABM/101		8-Channel ARINC-429 bus monitor with
	S	MyKAD_ABM_102	KAD/ABM/102		8-Channel ARINC-429 bus monitor
	Import	MyKAD_ABM_102_B	KAD/ABM/102/B		ARINC-429 bus monitor PARSER/packe
		MyKAD_ABM_102_B_EM1	KAD/ABM/102/B/EM1		ARINC-429 bus monitor PARSER/packe
		MyKAD_ARI_001_B	KAD/ARI/001/B		8 channel ARINC-429 bus monitor with
		MyKAD_CBM_102	KAD/CBM/102		Four-channel CAN Bus Monitor
		MyKAD_CBM_103	KAD/CBM/103		Four-channel CCDL/MCDL PARSER.
		MyKAD_CBM_104	KAD/CBM/104		Four-channel CSDB Parser.
		MUKAD CPM 105	KAD/CRM/105		Four channel CAN Pur Monitor

The module is added to the previously empty slot.

45.3.2 Setting bit-rates for the KAD/CBM/103

After adding the KAD/CBM/103 to your configuration, you can set bit-rates for channels.

- 1. If required, click + to expand the KAD/CBM/103 node and the Inputs node.
- 2. On the **Settings** tab, set the bit-rate for each CCDL channel by changing the values in the **Baud Rate** field. Also, change the value in the **Fill Value** field as required.





45.3.3 Adding CCDL packages

After setting bit-rates for each channel, add CCDL packages to each channel.

1. To add a package, right-click the channel where you want to add the package and click Add Package.



Packages Palette opens.

2. On the Packages Palette, select MyCCDLPackage and click Add.

cw Packages Palette							
CIW.	Generic						
New ACRA Component	Name 🍸	Instrument 🍸	Package Rate (Hz) ア	Туре 🍸	Subtype 🍸	Short Description $arbox$	Long Desc
	MyCANBusMessage		1	CCDL			
	MyCCDLPackage		1	CCDL	MCDL		
This File	MyCSDBPackage		1	CSDB			
This File	MyEBMPackage		1	CCDL			
,	MyGenericPackage		1	CCDL			
6	MyMBIPackage		1	CCDL			
Import							

3. On the **Packages** tab, set the **Message Identifier** field in the **Package Properties** pane.

octungs	Processes	Packages	Algorithms	Documentation		
 Links 						
Name 🍸		1	ype 7	Bit Rate 🍸	Source V	Channel 🍸
Link_MyKAD	_CBM_103_C	CDL-In(0)	CDL	n/a	MyKAD_CBM_103	CCDL-In(0)
 Package Principal 	operties					
 Package Pr 	operties					
 Package Pri ■ Name 	operties Me	essage Identifi	ier 🍸 Master/	Slave 🍸		

NOTE: Values must be in hex and in the range 00 to 1F.

4. In the **Master/Slave** field, select the message type. Options available are: **Master**, **Slave** or **Both**.



45.3.4 Adding parameters to a CCDL package

This section describes how to add parameters to the CCDL packages already defined.

1. In the Placed Parameters pane, click the Import Parameter button.

Settings	Processes	Package	s A	lgorithms	Documentation		
Links							
Name 🍸			Type 5	7	Bit Rate 🍸	Source V	Channel 🍸
Link_MyKA	MyKAD_CBM_103_CCDL-In(0) CCDL				n/a	MyKAD_CBM_103	CCDL-In(0)
Package P	Properties						
- 6							
Name 🍸	M	essage Iden	tifier 🍸	Master/Slav	ve 7		
MyCCDLPa	ckage 00			Master	•		
Content							
Placed Param	neters						
Impor	t Parameter						

Parameters Palette opens.

NOTE: 32 x 16-bit parameters are required.

- 2. To filter the list of parameters so that only 16-bit parameters are displayed, click the $\frac{1}{2}$ icon in the **Bits** column heading and type **16** in the text box that appears.
- 3. Click the Contains button and select Equals from the filtering choices.

	Parameters			
cw			Bits	7
ew ACKA Componi	Name 🍸		16	
4			Con	tains 🔻
	My 16-bit BitStream Parameter	No filte	ring	
Import	My 16-bit BitVector Parameter	Begins	with ith	
	My 16-bit IEEE-754-Float-Double Parameter	Contair	15	
	My 16-bit IEEE-754-Float-Single Parameter	Equals		
	My 16-bit OffsetBinary Parameter	Is empt	ty	
	My 16-bit SignedMagnitude Parameter	Is not e	mpty	

- 4. Select the first 32 x 16-bit parameters available.
- For information on using the Shift and Ctrl keys for selecting multiple fields, see the DAS Studio 3 User Manual.
 5. Select the Use My Renaming Rules radio option.
- 6. Select the **Rename To** check box and type a unique parameter name in the box provided.
- For information on the Use My Renaming Rules and Rename To fields, see the DAS Studio 3 User Manual.



7. Click Add.

On the Packages tab, 32 data words are added to the CCDL message.

4 5	6 7	8 9																						
4 5	0 7	8 9																						
		1 1	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
			_				-	-	-		-		_	-		-		-		-			-	_
												Defau	t Occi	urren	ces: 1			-	- I	.00%		<u></u>		Ŧ
										-				-										
Y	Data Bits	Y	Of	ffset W	Vords	Y																		
DATA	16		0																					
ATA 1	16		1				-																	E
AIA_1	10		-				-																	
DATA_2	16		2				_																	
DATA_3	16		3																					
DATA_4	16		4																					
DATA_5	16		5																					
DATA_6	16		6																					-
	♥ DATA DATA_1 DATA_2 DATA_2 DATA_3 DATA_4 DATA_5 DATA_6	The second sec	Y Data Bits Y DATA 16 DATA_1 16 DATA_2 16 DATA_3 16 DATA_5 16 DATA_6 16	Y Data Bits Y O DATA 16 0 DATA_I 16 1 DATA_2 16 2 DATA_3 16 3 DATA_5 16 5 DATA_6 16 6	Y Data Bits Y Offset V DATA 16 0 DATA_1 16 1 DATA_2 16 2 DATA_3 16 3 DATA_4 16 4 DATA_5 16 5 DATA_6 16 6	Y Data Bits Y Offset Words DATA 16 0 DATA_I 16 1 DATA_2 16 2 DATA_3 16 3 DATA_5 16 5 DATA_6 16 6	Y Data Bits Y Offset Words Y DATA 16 0 DATA_1 16 1 DATA_2 16 2 DATA_3 16 3 DATA_5 16 5 DATA_6 16 6	Y Data Bits Y Offset Words Y DATA 16 0 DATA_1 16 1 DATA_2 16 2 DATA_3 16 3 DATA_4 16 4 DATA_5 16 5 DATA_6 16 6	Y Data Bits Y Offset Words Y DATA 16 0 DATA_1 16 1 DATA_2 16 2 DATA_3 16 3 DATA_5 16 5 DATA_6 16 6	Y Data Bits Y Offset Words Y DATA 16 0 DATA_1 16 1 DATA_2 16 2 DATA_3 16 3 DATA_5 16 5 DATA_6 16 6	Y Data Bits Y Offset Words Y DATA 16 0 DATA_1 16 1 DATA_2 16 2 DATA_3 16 3 DATA_4 16 4 DATA_5 16 5 DATA_6 16 6	Y Data Bits Y Offset Words Y DATA 16 0 DATA_1 16 1 DATA_2 16 2 DATA_3 16 3 DATA_4 16 4 DATA_5 16 5 DATA_6 16 6	Y Data Bits Y Offset Words Y DATA 16 0 DATA_1 16 1 DATA_2 16 2 DATA_3 16 3 DATA_5 16 5 DATA_6 16 6	Y Data Bits Y Offset Words Y ATA 16 0 ATA_1 16 1 ATA_2 16 2 ATA_3 16 3 ATA_5 16 5 ATA_6 16 6	Y Data Bits Y Offset Words Y DATA 16 0 DATA_1 16 1 DATA_2 16 2 DATA_3 16 3 DATA_5 16 5 DATA_6 16 6	Y Data Bits Y Offset Words Y Default Occurrences: 1 YATA 16 0 <td< td=""><td>Y Data Bits Y Offset Words Y DATA 16 0 DATA_1 16 1 DATA_2 16 2 DATA_3 16 3 DATA_5 16 5 DATA_6 16 6</td><td>Y Data Bits Y Offset Words Y ATA 16 0 ATA_1 16 1 ATA_2 16 2 ATA_3 16 3 ATA_4 16 4 ATA_5 16 5 ATA_6 16 6</td><td>Y Data Bits Y Offset Words Y ATA 16 0 ATA[1] 16 1 ATA[2] 16 2 ATA[3] 16 3 ATA[4] 16 4 ATA[5] 16 5 ATA[6] 16 6</td><td>Y Data Bits Y Offset Words Y ATA 16 0 ATA_1 16 1 ATA_2 16 2 ATA_3 16 3 ATA_5 16 5 ATA_6 16 6</td><td>Product Occurrences: 1 ■ ■ m m 100% Y Data Bits Y Offset Words Y Data 16 0 Data 1 16 1 Data 2 16 2 Data 3 16 3 Data 4 16 4 Data 5 16 5 Data 6 6 6</td><td>Y Data Bits Y Offset Words Y ATA 16 0 ATA[1] 16 1 ATA[2] 16 2 ATA[3] 16 3 ATA[4] 16 4 ATA[5] 16 5 ATA[6] 16 6</td><td>Default Occurrences: 1 Immunolity I</td><td>Default Occurrences: 1 100% 0 Y Data Bits Y Offset Words Y DATA 16 0 DATA_1 16 1 DATA_2 16 2 DATA_3 16 3 DATA_5 16 5 DATA_6 16 6</td></td<>	Y Data Bits Y Offset Words Y DATA 16 0 DATA_1 16 1 DATA_2 16 2 DATA_3 16 3 DATA_5 16 5 DATA_6 16 6	Y Data Bits Y Offset Words Y ATA 16 0 ATA_1 16 1 ATA_2 16 2 ATA_3 16 3 ATA_4 16 4 ATA_5 16 5 ATA_6 16 6	Y Data Bits Y Offset Words Y ATA 16 0 ATA[1] 16 1 ATA[2] 16 2 ATA[3] 16 3 ATA[4] 16 4 ATA[5] 16 5 ATA[6] 16 6	Y Data Bits Y Offset Words Y ATA 16 0 ATA_1 16 1 ATA_2 16 2 ATA_3 16 3 ATA_5 16 5 ATA_6 16 6	Product Occurrences: 1 ■ ■ m m 100% Y Data Bits Y Offset Words Y Data 16 0 Data 1 16 1 Data 2 16 2 Data 3 16 3 Data 4 16 4 Data 5 16 5 Data 6 6 6	Y Data Bits Y Offset Words Y ATA 16 0 ATA[1] 16 1 ATA[2] 16 2 ATA[3] 16 3 ATA[4] 16 4 ATA[5] 16 5 ATA[6] 16 6	Default Occurrences: 1 Immunolity I	Default Occurrences: 1 100% 0 Y Data Bits Y Offset Words Y DATA 16 0 DATA_1 16 1 DATA_2 16 2 DATA_3 16 3 DATA_5 16 5 DATA_6 16 6

45.3.5 Placing incoming CCDL parameters

After adding parameters, it is necessary to place them into an outgoing package at a rate defined by you.

- 1. If required, click + to expand the controller node and the **Outputs** node.
- 2. Right-click the **Ethernet** link and select **Add Package**.



Packages Palette opens.

3. On the **iNET-X** tab, select the *MyPlacediNET-XPackage* package type and click Add.

On the **Packages** tab (ensuring the controller module is in context), an empty iNET-X packet with a **Stream ID** of **00** is created.

Settings	Processes	Package	es	Algorithms	Documentation					
 Links 										
Name 🍸			Type	Y	Bit Rate 🍸	Source V	Channel 🍸			
Link_MyK	AD_BCU_140_C_I	Ethernet	Ethern	et	n/a	MyKAD_BCU_140_C	Ethernet			
∧ Package	Properties									
Name 🍸		Rate (Ha	z) 7	Type 🕤	Sub Tj	ype 🍸 Stream	ID 7 (Destination IPA 🍸	Destination MAC 🍸	Destination UDP Pc
MyPlacedi	iNET-XPackage	1		INet-X	Placed	00	2	35.0.0.1	01-00-5E-00-00-01	0
•										•

4. Set values for the Rate (Hz), Stream ID, Destination IPA, and Destination UDP Port as required.



- 5. To place the CCDL parameters into this packet, click the **Import Parameter** button. **Parameters Palette** opens.
- 6. On the **Parameters Palette**, select all 32 of the CCDL data words previously created and click **Add Reference**. On the **Packages** tab, the CCDL parameters are added to the outgoing packet.

Placed Paramet	iers					
- 4						
Colour 🍸	Name 🍸 🛛 🔺	Data Bits 🍸	Occurrences ∇	Offset Bytes ∇	Actual Rate 🍸	Source Chase
	CCDL_DATA_10	16	1	28	16	MyKAM_CHS
	CCDL_DATA_11	16	1	30	16	MyKAM_CHS
	CCDL_DATA_12	16	1	32	16	MyKAM_CHS
	CCDL_DATA_13	16	1	34	16	MyKAM_CHS
	CCDL_DATA_14	16	1	36	16	MyKAM_CHS
	CCDL_DATA_15	16	1	38	16	MyKAM_CHS
	CCDL_DATA_16	16	1	40	16	MyKAM_CHS
	CCDL_DATA	16	1	42	16	MyKAM_CHS
	CCDL_DATA_18	16	1	44	16	MyKAM_CHS
	CCDL_DATA_19	16	1	46	16	MyKAM_CHS

45.3.6 Verifying configuration

Use the Verify tool to check that the current configuration contains no errors.

1. On the Quick Access Toolbar or the Tools menu, click Verify.

	File	Tools	
Program	Verify	MO	'n

When verification is complete, a message displays to confirm whether the configuration was verified successfully.

TIP! Details of the verification are available in the Message Server window. To display the Message Server window, double-click its icon in the notification area.

Double-click to open Message Server	Message Server Ksetup 97007 29/10/2013 16:23:35:4163: Multi-ChassisScheduler: Starting 97007 29/10/2013 16:23:36:3194: Multi-ChassisScheduler: Done. 97004 29/10/2013 16:23:36:3274: Starting schedule for chassis MyKAM_CHS_13U 97002 29/10/2013 16:23:36:275: Starting generation of EEPROM image for module KAD/CBM/103 type #0A60 97002 29/10/2013 16:23:36:275: Starting generation of EEPROM image for module KAD/BCU/140/C type #089D 97002 29/10/2013 16:23:36:6385: Information: Backplane utilization factor 0.0063, Ticks as NDP 496840, Ticks per cycle 500000, Number of transfers 39 97003 29/10/2013 16:23:36:6425: Finished EEPROM generation for module KAD/BCU/140/C type #25e9 97003 29/10/2013 16:23:36:6425: Finished EEPROM generation for module KAD/BCU/140/C type #25e9 97003 29/10/2013 16:23:36:6425: Finished EEPROM generation for module KAD/BCU/140/C type #25e9 97003 29/10/2013 16:23:36:6425: Finished EEPROM generation for module KAD/BCU/140/C type #25e9 97003 29/10/2013 16:23:36:6425: Finished EEPROM generation for module KAD/BCU/140/C type #25e9 97003 29/10/2013 16:23:37:755E Compilation complete EVPROM generation for module KAD/BCU/140/C type #0460 97005 29/10/2013 16:23:37:755E. Compilation Complete EVPROM generation for module KAD/CBM/103 type #0A60 97005 29/10/2013 16:23:37:755E. Compilation Complete EVPROM generation for module KAD/2BM/103 type #0A60 97005 29/10/2013 16:23:37:755E. Compilation Complete EVPROM generation for module KAD/2BM/103 type #0A60 97010 29/10/2013 16:23:37:755E. Compilation Complete EVPROM generation for module KAD/2BM/103 type #0A60 97010 97010 97010 97010 97010 97010 97010 97010 97010 97010 97010 97010 97010 97010 97010 9701 9701



This page is intentionally blank