

DSPOWERLINK OSP Control Group

8



On EPL Network



July 15, 2008

Through a joint effort between the B&R Automation and DSP Control Group engineers, it is now possible to share I/Os between B&R CPUs and DSPCG Motion Controllers on an EPL Network.

This document describes how to set up a B&R CPU to transmit and receive 32 bits of data to and from an MX4 motion controller Managing Node on an EPL Network.

- *NOTE:* Please see the Mx4 Powerlink manual for information on configuring and using an Mx4 as an EPL managing node. Also, the reader is assumed to be familiar with Automation Studio's interface and features.
- 1. Start Automation Studio v. 3.0.71.10 or higher.
- 2. From the File pull-down menu, select New Project.
- 3. In the window that appears, enter a program name and the path to the project; then click Next.

New Project Wizard		×
Description of the project	Enter the base parameters for the new project (name, path, description,). Name of the project: DSPowerlink Path of the project: C:\projects\DSPowerlink Browse C:\projects\DSPowerlink Browse C:\projects\DSPowerlink Browse Use default target (AR000 - Automation Runtime W32)	
		-
	<u>N</u> ext > Cancel	

4. Enter a CPU name and configuration name, or leave the defaults as they are and click Next.



5. Select your CPU model from the list provided and click <u>Next</u>.



6. Confirm your project settings, check Launch Insert Object Wisard box and click Finish.

New Project Wizard			X
	Location of the project:		
	C:\projects\DSPowerlink\		
	CPU module:	Power supply module:	
	X20CP1484		_
	Name of the configuration:	Name of the PLC:	
	DSPowerlink_Config1	XC20	
	Version of the target Automation	Runtime:	
18 - 10 - 10 - 10 - 10 - 10 - 10 - 10 -	V2.94		
🔽 Launch Insert Object V	√izard		
	< <u>B</u> ack	<u>F</u> inish Cance	

7. An Add Object window will appear next. Select Program in the left pane, New Program in the right pane, and then click <u>Next</u>.

ategories:	Templates:	
Package Program Data Object Motion Usualization OPC File	Existing Program	
new program		

8. Enter a program name (**Test1** for this example) and description, then click <u>Next</u>.

🗧 New Program		×
A new program		
	Program name:	Test1
	Description:	A new program
	What kind of files do you want	to generate?
	Data type declaration:	Test1.typ
	Variable declaration:	Test1.var
	8	
	< Back Next >	<u>Cancel</u> <u>H</u> elp

9. Select **ANSI C** as the language, and specify whatever program structure (separate or combined **Init, Cyclic**, and **Exit** programs) you prefer, then click <u>Next</u>.

New Program		X
	Program name: Description:	Test1 A new program
	What kind of files do you wa	ant to generate?
R.	Init program:	Test1Init.c
	Language:	ANSI C
	Cyclic program:	Test 1Cyclic.c
	Language:	ANSI C
	Exit program:	Test1Exit.c
	Language:	ANSI C
	Merge init, cyclic and ex	tit program into one file
	Deale Nexts	
	< Back Next >	<u>Lancel</u> <u>H</u> elp

10. Select **Yes**, to active **CPU** to assign the new program objects to the active CPU, then click **Finish**.



11. In the left pane of Automation Studio, select the Logical View tab, right click on file Test1.var and select Open As Table from the drop-down menu.

TSPowerlink/DSPowerlink_C	onfig1 - Automation Studio Beta 3.0.7	1.9 <u> </u>
File Edit View Insert Open P	roject SourceControl Online Tools W	/indow Help
'a é é l g 🗼 Pa	® ∞ ∝ × ⇔ ď ₫ ₽	* # E ¥ %
Project Explorer	· · · ×	
Object Name	Description	
DSPowerlink Global.typ Global.var Global.var Dibraries Diraries Dir	Global data types Global variables Global libraries A new program Local data types Local data types U cool watables Open As Table Open with Explorer Add Object Cut Copy Paste	
Logical View Configu	Delete Select All Rename Move Up Move Down Expand/Collapse • Zoom In Zoom Out Zoom 100% Properties	

In the window which appears in the right pane, right-click and select **Add Variable** from the drop-down menu.

🧭 Test	1::Test1.var	[Variable Declaration	ı]*				_ 🗆 🗙
🔗 A						-	
Name			Туре	& Reference	🔒 Constant	Value	Description [1]
	* COPYRIGH	Π-					
	* Program: Te * File: Test1. * Author: * Created: Ju	est 1 var ly 15, 2008					
	* Local variat	oles of program Test1					
		Add Variable					
I		Add Comment					
		Cut					
		Сору					
		Paste					
		Delete					
		Select All					
		Rename					
		Move Up					
		Move Down					
1		Sort Up 🕨					
		Sort Down					
•		Columns +					

Give the variable a name (Output32 for this example). In the Type column, select UDINT.

COPYRIGHT - Program: Test 1 File: Test I var Author: Created: July 15, 2008 Local variables of program Test 1	DPYRIGHT ogram: Test1 e: Test1 var thor: asted: July 15, 2008 asted: July 15, 2008 asted: July 15, 2008 asted: July 15, 2008 asted: July 15, 2008	n Test 1 [UDINT		0	l
* Program: Test 1 * File: Test 1.var * Author: * Created: July 15, 2008 * Local variables of program Test 1	ogram: Test1 e: Test1 var thor: eated: July 15, 2008 cal vanables of program ut32	n Test 1 UDINT		0	I
* Local variables of program Test1	cal variables of program	I Test1		0	
	ut32	UDINT		0	
Output32 UDINT UDINT UDINT UDINT					

12. Repeat the previous step to add another variable called **Input32**, also of type **UDINT**.

me	Туре	& Reference	🔒 Constant	Value	Description
* COPYRIGHT					
* Program: Test1 * File: Test1.var * Author: * Created: July 15, 2008					
 Local variables of prog 	ram Test1				
🤌 Input32	UDINT			0	
Output32	UDINT			0	

- 13. Press CTRL-SHIFT-S to save the variables created in the two previous steps.
- 14. Going back to the left pane of Automation Studio, select the **Physical View** tab, right-click on the CPU, and select **Open IF3 Powerlink Configuration** from the drop-down menu.



15. In the right pane, set Cycle time to 2000 ? sec, and Multiplexing prescale to 1, and mode to controlled node. Also, under Advanced, set node number to 1 and Interface handled as multiplexed station to off. All other settings may be left at their defaults.

ie	Value	Description	
🧼 Operating mode	POWERLINK V2		
- 🔷 MTU size	300		
	100 MBit		
- POWERLINK parameters			
Activate POWERLINK communication	on		
< Cycle time [µs]	2000		
🚺 Multiplexing prescale	1		
🚺 🛷 Mode	controlled node		
🕂 🛗 Advanced			
····· 🤣 Node definition	set explicitely		
🤣 Node number	1		
🤣 Ready flag	Set by system		
Interface handled as multiplexed station	off		
🤣 Optimization	data throughput		
POWERLINK NAT subnet	192.168.101.0	Last number should be 0 and is ignored	
Dynamic channels			
🗄 📲 🚮 Channel 1			
🤣 Name	TODO		
🧼 🤣 Data type	BOOL		
🛶 🤣 Direction	Output		
Ethemet parameters			

Also in the right pane, right-click on the Dynamic channels branch of the tree and select
 Append Element from the drop-down menu. For the Channel name, enter
 OutputChannel32, for the Data type select UDINT, and for Direction select Output.

ie.	Value	Description
Operating mode	POWERLINK V2	
🔗 MTU size	300	
	100 MBit	
POWERLINK parameters		
Activate POWERLINK communication	on	
🤣 Cycle time [µs]	2000	
🛊 🔷 Multiplexing prescale	1	
🔷 Mode	controlled node	
🖕 👘 Advanced		
www. 🛷 Node definition	set explicitely	
🤣 Node number	1	
🚸 Ready flag	Set by system	
🛷 Interface handled as multiplexed station	off	
	data throughput	
POWERLINK NAT subnet	192.168.101.0	Last number should be 0 and is ignored
Dynamic channels Insert Element		
E- Channel 1 Append Element		
Name	TODO	
Data type	BOOL	
Direction Pacte	Output	
Ethemet parameters		
Activate Ethemet c	on	
st J XC2U.CPU [Select All		
Kerdile		
Move Down		
Charge A		

17. Repeat the previous step to create a channel named InputChannel32, also of type UDINT, but with Direction set to Input. Press CTRL-SHIFT-S to save.

🖌 xc	20.CPU	IF3 POWERLINK Configuration]*			
Name	5		Value	Description	-
		Advanced			
	T	Node definition	set explicitely		
		Node number	1		
		🔷 Ready flag	Set by system		
		Interface handled as multiplexed station	off		
			data throughput		
		POWERLINK NAT subnet	192.168.101.0	Last number should be 0 and is ignored	
	ġ,	Pynamic channels			
		🚽 , 😭 Channel 1			
		Name	OutputChannel32		
		🧼 Data type	UDINT		
		, Ø Direction	Output		
		🕂 🚰 Channel 2			
			InputChannel32		
			UDINT		
		🔷 Direction	Input		
		i Channel 3			
K.		🔹 Name	TODO		
		🔌 Data type	BOOL		
		📖 į 🔌 Direction	Output		
E	- 1	Ethernet parameters			
	-	Activate Ethemet communication	on		
		A 11 1	1 10 11		

18. In the left pane of Automation Studio, select the **Configuration View** tab. Select **XC20**, right-click **IoMap.iom** and select **Open As Table** from the drop-down menu.

File Edit View Insert Open Project Source Control Online Tools Window Help Project Explorer Configuration Batch DSPowerlink_Config1 [Active] Hardware hc XC20 Cut Courfiguration file CPU system configuration file CPU system configuration Add Configuration Add Configuration Add Object Cut Copy Paste Insert Delete Select All Rename Move Up Move Down Columns Expand/Collapse Zoom In Zoom Ut Zoom In Zoom In Zoom In Properties	DSPowerlink/DSPowerlink_Config1 - Automation Studio Beta 3.0.71.9					
Project Explorer Configuration Batch Hardware topology AcConfig 1 Project Explorer Project Explorer <th>File Edit View Insert Open</th> <th>Project Source Control</th> <th>Online Tools Window Help</th>	File Edit View Insert Open	Project Source Control	Online Tools Window Help			
Project Explorer - × Configuration Batch Description Image: State of the state of	1 1 1 1 1 1	b @ ∽ ~ X ↔				
Configuration Batch Description Image: Strategy of the	Project Explorer		* X			
DSPowerlink_Config1 [Active] Hardware hc XC20 XC20 Cpu.sw Cpu.per PVMap.vvm Prof. Sysconf br ArConfig.tc Sysconf br Sysconf br Sysconf br Sysconf br Sysconf br Add Configuration Add Configuration Add Object Cut Copy Paste Insert Delete Select All Rename Move Up Move Down Columns Expand/Collapse Zoom In Zoom 00t Zoom 100% Properties	Configuration	Batch	Description			
The Logical View Contemporter Properties al View	DSPowerlink_Config Hardware.hc XC20 Cpu.sw Cpu.sw Cpu.per PVMap.vvm ArConfig.tc sysconf.syc sysconf.syc	Open with Explorer Open As Text Open As Table Add Configuration Add Object Cut Copy Paste Insert Delete Select All Rename Move Up Move Down Columns Expand/Collapse Zoom In Zoom (J00%)	Hardware topology Files belonging to this PLC Software configuration Declaration of permanent variab IO mapping file PV mapping file Runtime configuration file CPU system configuration CPU system configuration			
	Logical View	Properties	al View			

19. In the right pane, right-click and select Add Mapping from the drop-down menu. Set the mapping Direction to Output, the ChannelWidth to 32 Bit, the IO Address to IF3.OutputChannel32, and set Task Class to Automatic.

rection	Channel Width	IO Address	Task Class
COPYRIGHT -			
 File: IoMap.iom Author: Created: July 15, 2008 			
* IO mapping file			
• Output	32 Bit	IF3.OutputChannel32	Automatic
Output	32 Bit	IF3.OutputChannel32	Automatic
Output	32 Bit	IF3.OutputChannel32	Automatic
Output	32 Bit)IF3.OutputChannel32	Automatic
Cutput	32 Bit	IF3.OutputChannel32	Automatic
Cutput	32 Bit)IF3OutputChannel32	Automatic
Output	32 Bit)IF3OutputChannel32	Automatic

20. In the **PV or Channel Name** column, double-click in the mapping's row. A button should appear at the right side of the box, which, when clicked, opens a window that allows the user to select the C variable to which to map this channel. Under the name of your C program, double-click on variable **Output32**.



🧭 A				
Nidth	IO Address	Task Class	PV or Channel Name	Inverse
	IF3.OutputChannel32	Automatic	Test 1.Output 32	
			KON NO DE	

21. Repeat the previous step to create an input channel mapping between **IF3.InputChannel32** and the C variable **Input32**.

32 Bit	IF3.InputChannel32	Automatic	Test1.Input32
32 Bit	IF3.OutputChannel32	Automatic	Test1.Output32
	32 Bit 32 Bit	32 Bit IF3.InputChannel32 32 Bit IF3.OutputChannel32	32 Bit IF3.InputChannel32 Automatic 32 Bit IF3.OutputChannel32 Automatic

22. Going back to the Logical View tab in Automation Studio's left pane, you can edit the C program Test1Cyclic.c to set the value of Output32. Note that Output32, which maps to IF3.OutputChannel32, is an output *from* the CPU *to* the Managing Node Mx4. The high 16 bits of Output32 will be returned in DSPL variable EPL_INP1_REG, while the low 16 bits of Output32 will be returned in DSPL variable EPL_INP2_REG.

Variable Input32 will be mapped such that when a DSPL command such as

EPL_outp_on(0x1234, 0x5768)

is run on the Managing Node Mx4, Input32's contents will be 0x12345678.

C Test1::Test1Cyclic.c [ANSI C]*	- D ×
18 0 匡 2	-
+ COPYRIGHT	
* Program: Test1 * File: TestlCyclic.d * Author: * Created: July 15, 2008	
* Implementation of program Test1	
<pre>#include <bur plctypes.h=""></bur></pre>	
<pre>#ifdef _DEFAULT_INCLUDES #include <asdefault.h> #endif</asdefault.h></pre>	
<pre>void _CYCLIC Test1Cyclic(void) { UDINT _ ver1:</pre>	
Output32 = 0x12; var1 = Input32; }	
	•

23. Press CTRL-SHIFT-S to save **Test1Cyclic.c** program. To map this program into a cycle, select **Physical View** tab and right-click on the CPU (e.g., **X20**). Select **Open Software Configuration**.



24. Look for your program Test1. Drag your program to an appropriate cycle time (e.g., 10 ms, 20 ms etc.).

iect Name	Version	Transfer	Size (hut	Source	Source File	Description
CPU CPU	Verdiori	Transfor	0120 (0)1	500100	0001001110	Description
Cvclic #1 - [10 ms]						-
Cyclic #2 - [20 ms]						
Cvclic #3 - [50 ms]						
Cvclic #4 - [100 ms]		-				
Cvclic #5 - [200 ms]	1					1
Cyclic #6 - [500 ms]						1
Cyclic #7 - [1000 ms]						
Cyclic #8 - [10 ms]						-
I Test1	1.00.0	UserROM	0	Test1	\Cpu.sw	A new progr
🗍 Data Objects						
🚱 Nc Data Objects						
🐱 Visualisation						
🎒 Binary Objects						
🖓 Library Objects						
Configuration Objects						

25. Now you may Rebuild your C program. To do so, select **Project** from the top menu bar and Chose **Rebuild** option.

DSPowerlink/DSPowerlink_	Con	fig1 - Automation Studio Beta 3.0.71.9
File Edit View Insert Open	Proj	ect Source Control Online Tools Window Help
i 🕆 🖻 🖬 🖬 👗 🖻		Build Configuration F7
Project Explorer		Rebuild Configuration Ctrl+F7
Model no. C XC20 C X88 X20CP1484 USB 1		Grean Connguration Build Cross Reference Iransfer To Target Ctrl+F5 Stop Build
		B <u>a</u> tch
		Update Library Declarations Change Runtime Version Update System Components Generate Bus Navigator Source
		Settings

26. The last step is transfer your program to the CPU - a familiar task for those familiar with Automation Studio. That is, select **Project** from the top menue bar and choose **Transfer To Target**.