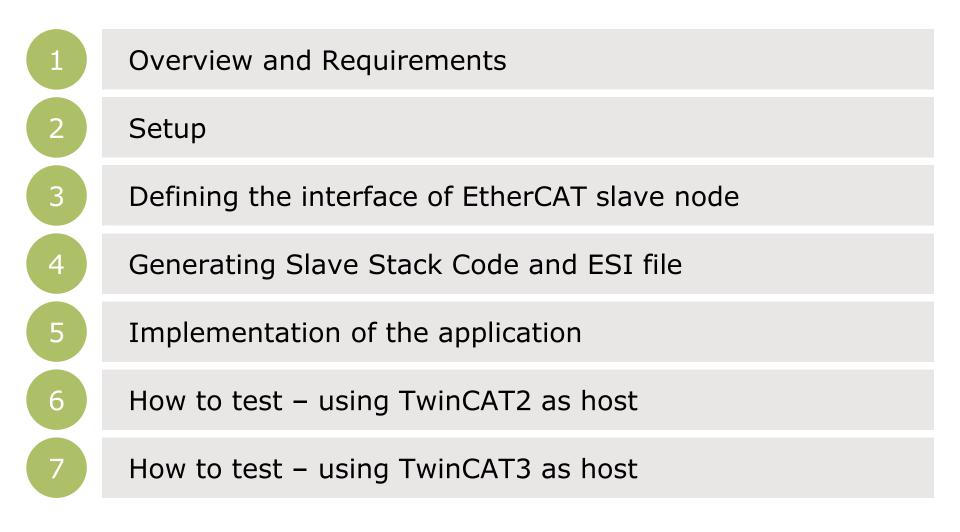
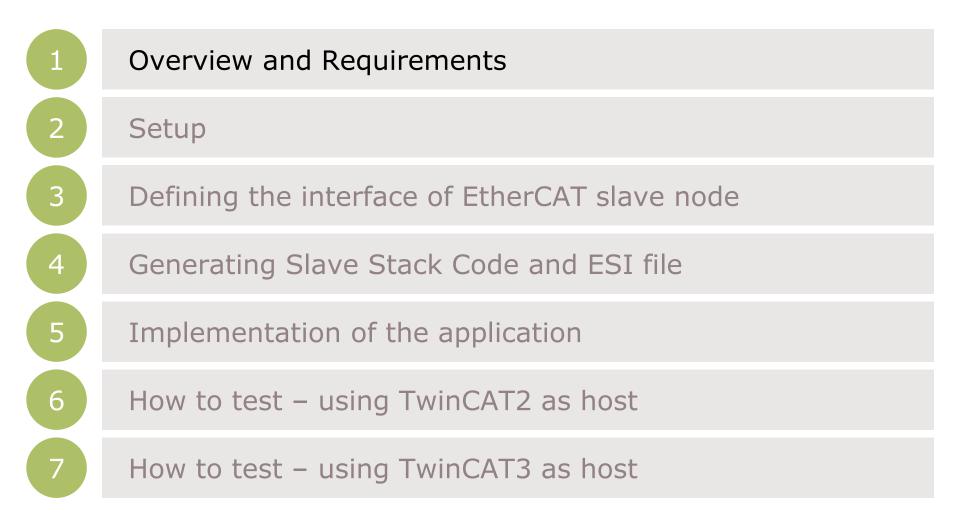
XMC4300 EtherCAT APP SSC Slave Example Getting Started V3.0





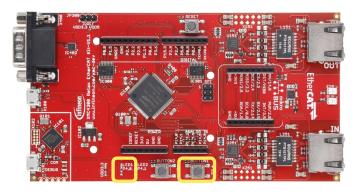






Overview





This example demonstrates the implementation of a EtherCAT slave node using the Beckhoff SSC Tool to generate the slave stack code for "XMC4300 Relax EtherCAT Kit".

While reviewing this example you will see in output direction the EtherCAT

master controlling LED1 on the "XMC4300 Relax EtherCAT Kit". In input direction you will monitor inside the master device the status of BUTTON1. You will observe inside the source code how to modify the mapping of the data structures to the I/Os for your own evaluations and testing. Furthermore you will learn how to modify the data structures and generate a slave stack code which fits to your needs. In this example we will demonstrate how easy it is to setup a proper EtherCAT communication by using the EtherCAT APP.



Requirements

XMC4300 Relax EtherCAT Kit



RJ45 Ethernet Cable



Windows Laptop installed

- DAVE v4 (Version4.1.4 or higher)
- TwinCAT2 or TwinCAT3 Master PLC
- Slave Stack Code Tool Version 5.12



Micro USB Cable (Debugger connector)

Requirements - free downloads





TwinCAT2 (30 day trial; 32bit Windows only) Link: Download TwinCAT2

or



TwinCAT3 (no trial period; usability limited; 32bit and 64bit Windows) Link: Download TwinCAT3

ATTENTION: According our experience TwinCAT is best compatible with Intel[™] ethernet chipset. For details on compatibility with your hardware, additional driver and general installation support please get into contact with your local BECKHOFF support.

Requirements - free downloads





DAVE (v4.1.4 or higher) Link: <u>Download DAVE (Version 4)</u>



EtherCAT Slave Stack Code Tool Version 5.12

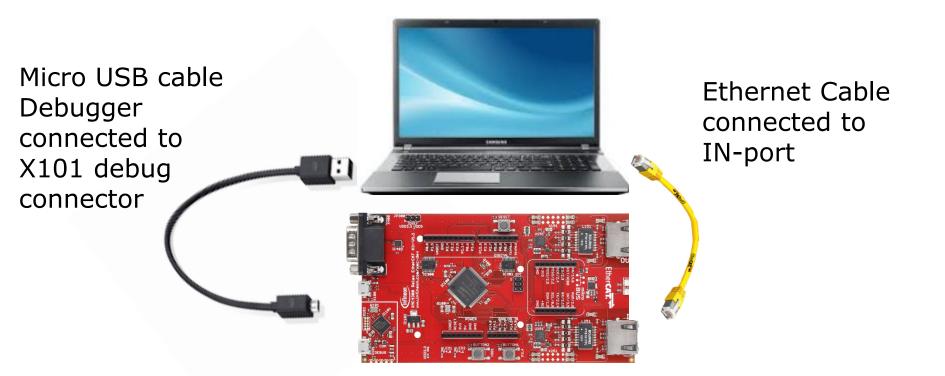
(ETG membership obligatory) Link: <u>Slave Stack Code Tool</u>



	Overview and Requirements
2	Setup
3	Defining the interface of EtherCAT slave node
4	Generating Slave Stack Code and ESI file
5	Implementation of the application
6	How to test – using TwinCAT2 as host
7	How to test – using TwinCAT3 as host

Setup – Hardware







Setup – Import example project into DAVE

🎸 D	AVE CE - DAVE™ - C:\Wo	orkspaces\DAVE-4.1\Ethe
File	Edit Source Refacto	or Navigate Project
	New	Alt+Shift+N ►
	Open File	
	Close	Ctrl+W
	Close All	Ctrl+Shift+W
	Save	Ctrl+S
	Save As	
G	Save All	Ctrl+Shift+S
ď	Rename	F2
\$	Refresh	F5
Ē,	Print	Ctrl+P
	Switch Workspace	+
_	Restart	
2	Import	
è1	Export	
	Properties	Alt+Enter
	Exit	

🍯 Import DAVE Projects		
Import DAVE projects		
Import Existing DAVE Pro	jects	
Select Root Directory		Browse
Select Archive File	$\label{eq:c:Users} C: Users \end{tabular} Scheme C: Users \end{tabular} C: Users \end{tabular} Scheme C: Users \end{tabular} C: Users \$	Browse
Project List:		
XMC4300_Relax	EtherCat_APP_Slave_SSC(XMC4300_Relax_EtherCat_APP_Slave_SSC)	Select All
✓ Copy Projects Into W	orkspace	
?	< Back Next > Finish	Cancel

Select			\mathbb{P}
Select an impo	ort source:		
D/ Control D/ D/ D/ Control D/	AVE Project	Local Library Store	
 ▷ ▷ Plug- ▷ ▷ Run/I ▷ ▷ Team ▷ ▷ XML 			



Setup – Import example project into DAVE

	🌜 D.	AVE C	E - DAV	E™ -	C:\Wo	rkspac	es\DA\	VE-4.1∖E	THERC	AT_D
	File	Edit	Sourc	e R	efacto	r Nav	vigate	Search	n Proj	ect
	H	G	2 🏹	B			20 70 70	5 Q	.e 🗟	Ŧ
	E	C/C++	Project	ts 🖾	۲ <u>م</u>	roject	Explor	er		
						<	> +	Q (- 4	\bigtriangledown
	4	😂 XN	AC4300	_Rela	x_Eth	erCat_	APP_S	ave_SS	C [Acti	ve - I
		Þ 🐝	Binarie	5						
		Þ 🖻	Includ	es						
		۵ 🗁	Dave							
		۵ 🗁	Debug							
			Librari	es						
Ц		4 🕞							_	
1		⊳	🕞 Sro							
- ว			,			ECAT	_SSC_C	onfig.xr	nl	
2			🐴 XM		C.xlsx					
		▶ 🗁	Startup							
		⊳ .c	main.c							
			linker_		.ld					
			solver.	bak						

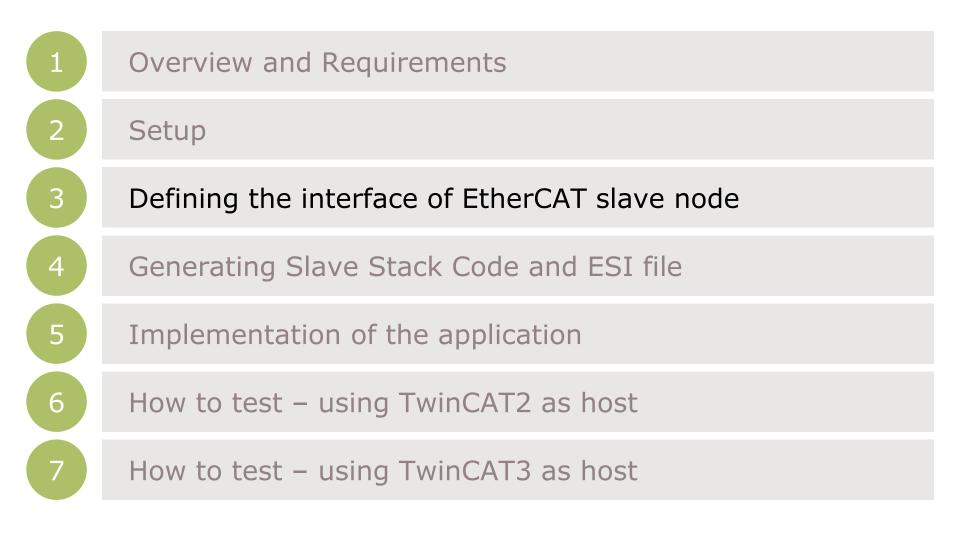
After the project import you will find this project folder structure.

1 The project is nearly complete for building, it only misses the EtherCAT slave stack code. For these files the Src folder has been already prepared.

2 The EtherCAT slave stack code for the XMC4300 can be generated by configuration files. These configuration files are included in the project already.

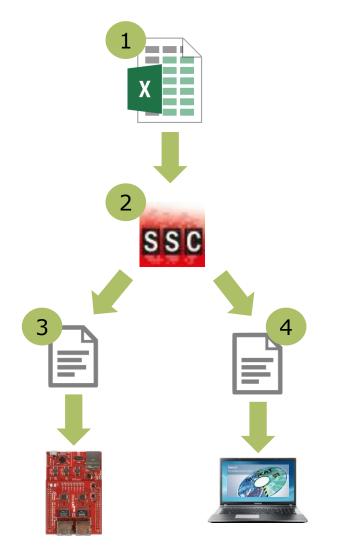
The following slides show in detail how to define your EtherCAT slave node interface and to generate the slave stack code.





The flow to define the EtherCAT slave node interface





1 Take the Excel Worksheet provided inside the example project to define your EtherCAT slave node interface.

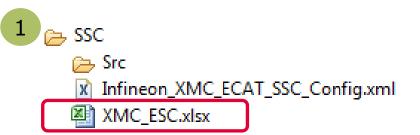
2 The Beckhoff SSC-tool uses the excel sheet as an input to generate the output-files.

3 The generated EtherCAT slave stack code does apply for the XMC4300.

4 The generated EtherCAT Slave Information file (ESI) does apply for the EtherCAT host. There the relevant interface information about the slave is stored.



Defining the interface of EtherCAT slave node



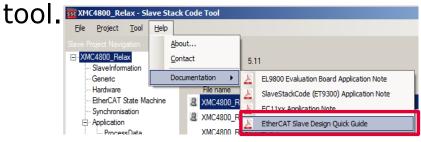
2								
Z	Index	-	ObjectCode ~	SI ~	-	DataType 🔄	-	Name
	ll0x6nnx		Input Data of th	e Module (0x60	00)0 - 0x6FFF)		
	0x6000		RECORD					IN_GENERIC
				0x01		UINT		IN_GEN_INT1
				0x02		UINT		IN_GEN_INT2
				0x03		UINT		IN_GEN_INT3
				0x04		UINT		IN_GEN_INT4
				0x05		BOOL		IN_GEN_Bit1
				0x06		BOOL		IN_GEN_Bit2
				0x07		BOOL		IN_GEN_Bit3
				0x08		BOOL		IN_GEN_Bit4
				0x09		BOOL		IN_GEN_Bit5
				0x0A		BOOL		IN_GEN_Bit6
				0x0B		BOOL		IN_GEN_Bit7
				0x0C		BOOL		IN_GEN_Bit8

ll0x7nnx	Output Data o	f the Module (0x)	7000 - 0x7FFF)	
0x7000	RECORD			OUT_GENERIC
		0x01	UINT	OUT_GEN_INT1
		0x02	UINT	OUT_GEN_INT2
		0x03	UINT	OUT_GEN_INT3
		0x04	UINT	OUT_GEN_INT4
		0x05	BOOL	OUT_GEN_Bit1
		0x06	BOOL	OUT_GEN_Bit2
		0x07	BOOL	OUT_GEN_Bit3
		0x08	BOOL	OUT_GEN_Bit4
		0x09	BOOL	OUT_GEN_Bit5
		0x0A	BOOL	OUT_GEN_Bit6
		0x0B	BOOL	OUT_GEN_Bit7
		0x0C	BOOL	OUT_GEN_Bit8
i i				

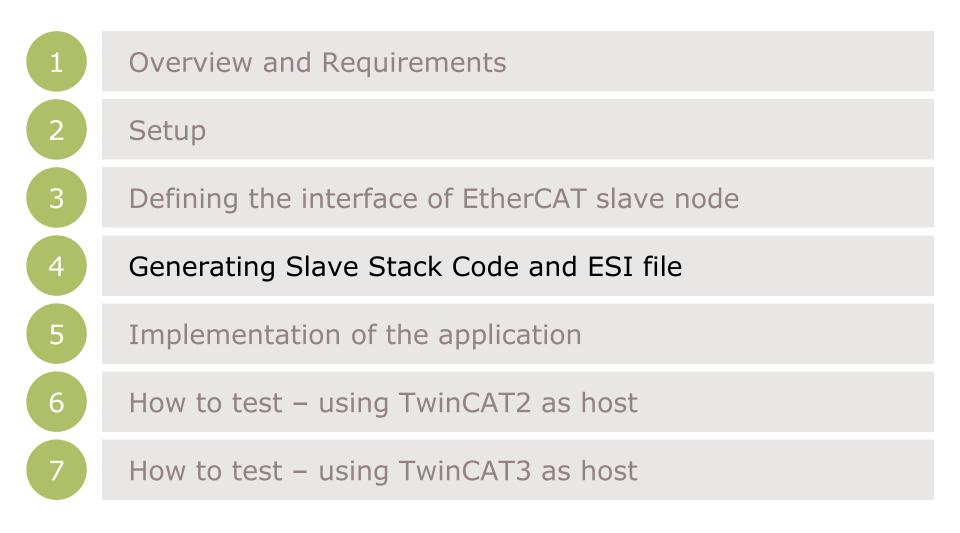
1 Double click on the excel file to open it.

² Check the content of the file. The data defined in both I/O directions is 4x16bit integers and 8x1bit booleans.

3 For further details on how to define your own interface you may want to follow the instructions inside *EtherCAT Slave Design Quick Guide.pdf* inside SSC





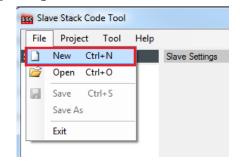




1	ve Stack Code Tool New Project	3
	Default Custom EL980012Axis CIA402 Sample	
	Custom EL9800 2Avis CiA402 Sample This configuration provides a CiA402 sample application for the new EL9800 EtherCAT Evaluation Kit. The corresponding device description is located in file "\esi\SlaveStackCode.xml" (device: EL9800_CiA402) or will be created with this tool.	
	Import	1

2 Import Configuration file(s)		×
🔾 🔾 🗢 📗 « XMC4300_Relax_E	therCat ► SSC ►	م
Organize 🔻 New folder	8==	• 🔳 🔞
Downloads Recent Places	Name	Date modified 13.06.2016 10:14
 □ Libraries □ Documents □ Music □ Pictures □ Subversion □ Videos 	Infineon_XMC_ECAT_SSC_Config.xml	04.05.2016 15:20
19 Computer ᆕ SystemDisk (C:)	~ <	Þ
File name:	▼ xml (*.xml) Open ▼	Cancel

1 Start the sec tool and create a new project File >> New



² Select the configuration file which you find inside the example project.



ave Stack C	ode Tool New Project
O Default	
Custom	Infineon XMC EtherCAT Slave Sample <infineon technologies=""></infineon>
Version: 1.0 NOTE: This	
Shall be set	if the Slave code executes on an XMC4800/XMC4300 device.
Import	ОК

³ Select the Infineon device inside the drop down list and confirm with the OK button. Your project will be created.



EtherCAT Slave* - Slave Stack (Code Tool	4 or biab
File Project Tool Help		
Slave Project Navigation	Slave Settings	
⊡ EtherCAT Slave	Name	Value
SlaveInformation	VENDOR_ID	0x0000034E
Generic Hardware	VENDOR_NAME	Infineon Technologies
- EtherCAT State Machine	VENDOR_IMAGE	424D160500000000000
Synchronisation	GROUP_NAME	Infineon slave
Application ProcessData	GROUP_IMAGE	424DD80200000000000
···· Mailbox	DEVICE_IMAGE	424DD80200000000000
ⁱ Compiler	PRODUCT_CODE	0x0000000
	REVISION_NUMBER	0x0
	SERIAL_NUMBER	0x0
	DEVICE_PROFILE_TYPE	0x00001389
	DEVICE_NAME	XMC_ESC
	DEVICE_NAME_LEN	0x7
	DEVICE_HW_VERSION	1.0
	DEVICE_HW_VERSION_LEN	0x3
	DEVICE_SW_VERSION	5.12

- Check the settings inside SlaveInformation: vendor ID, vendor name, product ID and product code are customer specific and are used by the host to identify the slave.
- Define revision number, serial number, device name, HW/SW version according to your needs.
- The vendor ID/name and product code assigned to infineon may be used for evaluation purpose only. For productive purpose your own vendor ID/name assigned by the EtherCAt Technology Group is obligatory.



File Project To	ol Help			
Slave Project Naviga 🛩	Show Con	flict Window		
	Options			Value 0x0000034E
Generic	EEPROM F	rogrammer		Jofineog Technolo
Hardware	Applicatio	n 🕨	Create new	D000
Synchronisation	Machine	REVISION_NUMBER	Import	000
Application				0~0000000
Organize Vew fold	-	it ▶ SSC ▶ ▼ 4	Search SSC	ر ⊛ ∎ • ≣
	-	ıt ▶ SSC ▶ ▼ ◀	,	
Organize 🔻 New fol	-		,	E 🔹 🚺 🔞
	der	-	,	
Organize Vew fol	der	Name	,	EE 🔻 🛄 🔞
Organize New fold Favorites Desktop Downloads Recent Places	der	Name	,	E ▼ □ 0 Date modified 13.06.2016 10:
Organize New fold Favorites Desktop Downloads	der	Name	,	E ▼ □ 0 Date modified 13.06.2016 10:
Organize New fold Favorites Desktop Downloads Recent Places Libraries	der	Name	,	E ▼ □ 0 Date modified 13.06.2016 10:
Organize New fold Favorites Favorites Desktop Downloads Recent Places Libraries Libraries Documents Music Pictures	der	Name	,	E ▼ □ 0 Date modified 13.06.2016 10:
Organize New fold Favorites Favorites Desktop Downloads Recent Places Libraries Documents Music	der	Name	,	E ▼ □ 0 Date modified 13.06.2016 10:

- 4 Import the EXCEL-sheet which defines the interface of your EtherCAT node.
- ⁵ Select the EXCEL-file provided inside the example project.



6 EtherCAT Slave* - Slave Stack	Code Tool	
File Project Tool Help		
Slave Pr 🕢 Project Update	Ctrl+F	Value 0x0000034E
Create new Slave Fi	iles F5 A IE	Infineon Tech 424D1605000
- Synchronisation	GROUP_NAME	Infineon slave

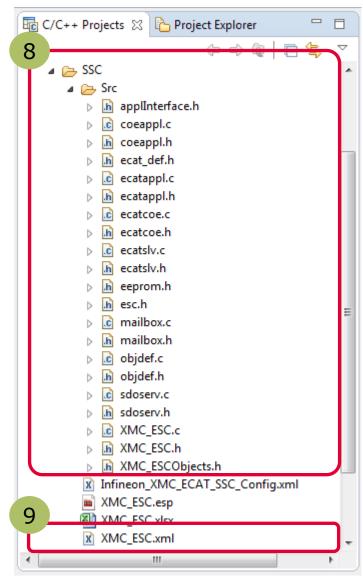
6 Click on Project >> Create new Slave Files to start file generation.

7 In this step the destination folder for the EtherCAT Slave Stack Code and the ESI file can be adapted. For this example it is recommended to take the default settings.

-			
Project File	C:\Workspace	s\DAVE-4.1\ETHERCAT_DC\XMC4300_Relax_EtherCat_APP_S	
	Source Folder	C:\Workspaces\DAVE-4.1\ETHERCAT_DC\XMC4300_Relax_E	Change
	ESI File	C:\Workspaces\DAVE-4.1\ETHERCAT_DC\XMC4300_Relax_	Change
	Doc Folder	C:\Workspaces\DAVE-4.1\ETHERCAT_DC\XMC4300_Relax_E	Change



Find and use your result

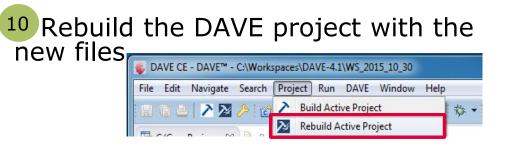


After the generation process the respective files are inside the project space:

⁸ Check the availability of the generated slave stack code

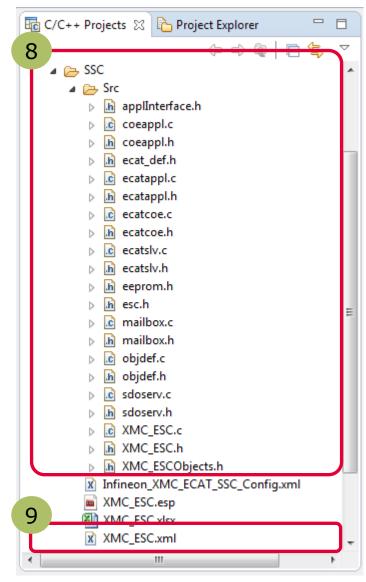
9 Check the availability of the ESI-file and download to the host by these 3 steps:

- Stop TwinCAT System Manager
 Copy the ESI file to your TwinCAT installation
 - C:\TwinCAT\Io\EtherCAT
- 3. Restart TwinCAT System Manager to start re-work of the device description cache.





Find and use your result



After the generation process the respective files are inside the project space:

⁸ Check the availability of the generated slave stack code

⁹ Check the availability of the ESI-file and download to the host by these 3 steps:

- 1. Stop TwinCAT System Manager
- 2. Copy the ESI file to resp. destination for TwinCAT2:

C:\TwinCAT\Io\EtherCAT for TwinCAT3:

C:\TwinCAT\3.1\Config\Io\EtherCAT

3. Restart TwinCAT System Manager to start re-work of the device description cache.

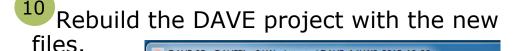


 Image: Search Project Run DAVE Window Help

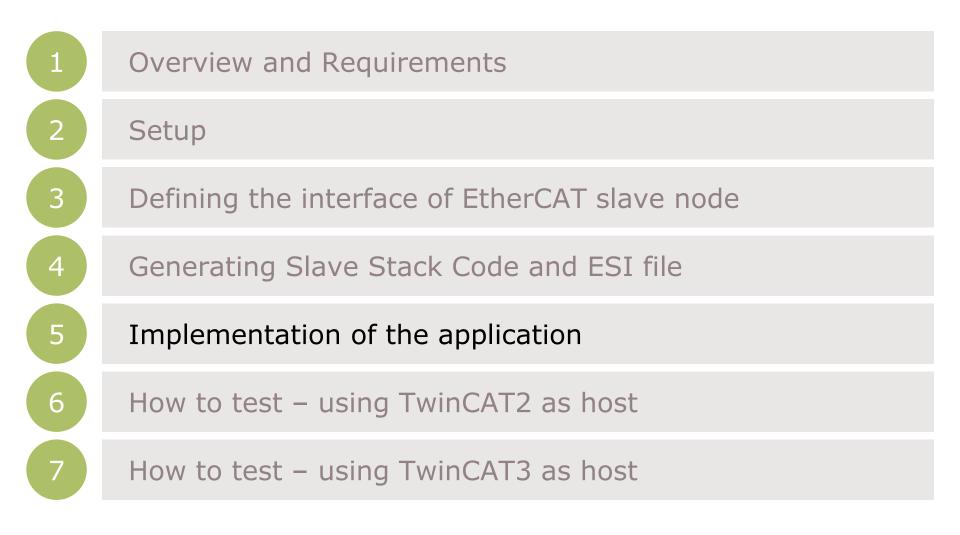
 Image: Search Project Run DAVE Window Help

 Image: Search Project Run DAVE Window Help

 Image: Search Project Run DAVE Rebuild Active Project

 Image: Search Project Run DAVE Rebuild Active Project





Copy data from/to local data to/from ESC memory



Inside the generated file *XMC_ESC.c* the link to your application must be implemented. Modify the source code accordingly which copies the application data to/from ESC memory to the local application memory:

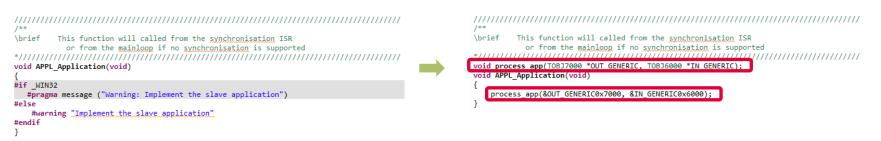
Originally generated code: Modified code: \param pData pointer to input process data \param pData pointer to input process data \brief This function will copies the inputs from the local memory to the ESC memory This function will copies the inputs from the local memory to the ESC memory \brief to the hardware to the hardware void APPL InputMapping(UINT16* pData) void APPL InputMapping(UINT16* pData) #if _WIN32 memcpy(pData,&(((UINT16 *)&IN_GENERIC0x6000)[1]),SIZEOF(IN_GENERIC0x6000)-2); #pragma message ("Warning: Implement input (Slave -> Master) mapping") #else #warning "Implement input (Slave -> Master) mapping" #endif /** pData pointer to output process data \param \param pData pointer to output process data This function will copies the outputs from the ESC memory to the local memory \brief \brief This function will copies the outputs from the ESC memory to the local memory to the hardware to the hardware void APPL OutputMapping(UINT16* pData) void APPL OutputMapping(UINT16* pData) #if WIN32 memcpy(&(((UINT16 *)&OUT_GENERIC0x7000)[1]),pData,SIZEOF(OUT_GENERIC0x7000)-2); #pragma message ("Warning: Implement output (Master -> Slave) mapping" #else #warning "Implement output (Master -> Slave) mapping" #endif }

Implement application specific slave node behaviour



Inside the generated file *XMC_ESC.c* file the function APPL_Application is implemented. This function implements the application specific code to handle input and output... A) ... from mainloop or B) ... if synchronisation is active from ISR Inside main.c of the example, the function *void process_app(TOBJ7000 *OUT_GENERIC, TOBJ6000 *IN_GENERIC);* implements the mapping of the input/output data to buttons and LEDs. Therefore please modify the function APPL_Application to call process_app in the following way:

Originally generated code:



Modified code:

Description – process of input and output

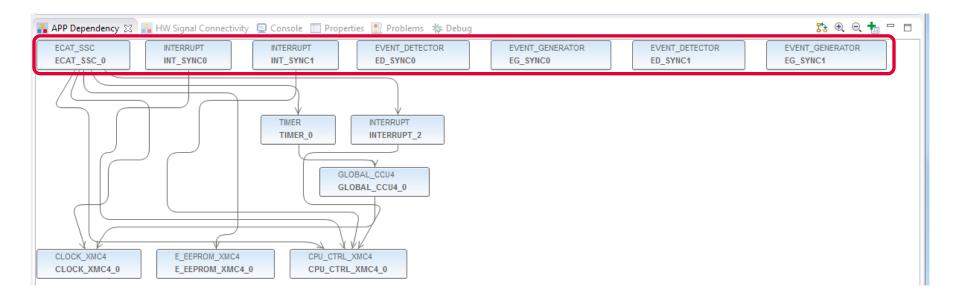


S DAVE CE - XMC4300_Relax	x_EtherCat_APP_Slave_SSC/main.c - DAVE™ - C:\Workspaces\DAVE-4.1\ETHERCAT_DC
File Edit Source Refacto	or Navigate Search Project DAVE Window Help
E G 🔀 🎽 🕨	山 🗉 🖉 ユ つ レ 弓 党 🙆 税 🏪 🔤 🛢 🗊 💷 🕒 🕏 💭 🌮 🖬 🈕 🖉 🗐
<u> </u> 월 ▼ 🖗 ▼ 🗢 ↔ →	▼ Quick Access 🛛 🖻 🖪 DAVE IDE 💽 DAVE CE 🎋 Debug
🖥 с 🛛 🍐 Р 🗖 🗖	<mark>I ain.c ⊠</mark>
0 0 0 0 0 5	840 void process_app(TOBJ7000 *OUT_GENERIC, TOBJ6000 *IN_GENERIC)
~	85 {
⊿ 😂 XMC4300_Relax_Et	<pre>86 /* OUTPUT PROCESSING */ 87 /* Check bitfield set by master OUT GEN Bit18 and set LEDs accordingly */</pre>
A Binaries	<pre>XMC_GPI0_SetOutputLevel(P_LED1, MAP2LEVEL(OUT_GENERIC->OUT_GEN_Bit1));</pre>
▷ manes ▷ manes	<pre>89 XMC GPIO SetOutputLevel(P LED2, MAP2LEVEL(OUT GENERIC->OUT GEN Bit2));</pre>
	90
Dave	91 /* INPUT PROCESSING */
> 🗁 Debug	92 /*Check Button 1 and set IN_GEN_Bit1 which is sent to master accordingly*/
b 🗁 Libraries	93 if (XMC_GPIO_GetInput(P3_4))
SSC	94 IN_GENERIC->IN_GEN_Bit1 = 1;
Startun	95 else
c main.c	96 IN_GENERIC->IN_GEN_Bit1 = 0;
	97 }
📄 IInкer_script.ld	
solver.bak	

Within the slave stack code the function process_app is called. This process_app function process the binary output data (master->slave) to set the LED1 "XMC4300 Relax EtherCAT Kit". The states of the BUTTON1 is checked and propagated to the input data (slave->master).

Description – Overview on used APPs





The ECAT_SSC APP assigns the system resources (automatically done by DAVE by using the respective lower level apps) and pins (by manual configuration) to setup a proper EtherCAT communication. The EVENT_DETECTOR, EVENT_GENERATOR and INTERRUPT APPs are used inside this example to connect the sync_out_0 and sync_out_1 of the ECAT_SSC APP to the interrupt service routines of the SSCstack.

Description – EtherCat ports and physical connection



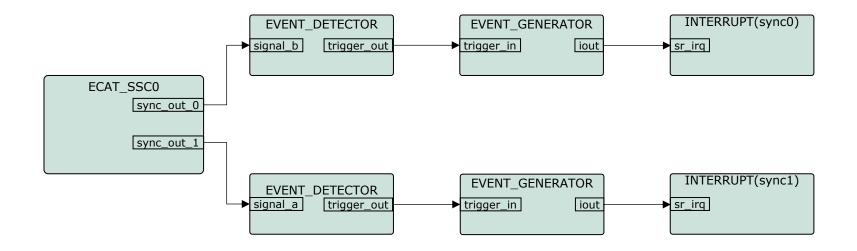
	APP Depend	lency 🛛	🔒 HW S	ignal Connectivity		
	ECAT_SSC		INT	ERRUPT		
	ECAT	Configu	e APP Inst	ance		
		Rename	Instance L	abel	E .	
	//				f	
	+	Add New	/Instance		- F	
	×	Remove.				
	_					
	-	HW Sian	al Connec	tions	ł	
		Manual I	Pin Allocat	tor		
		Manual I	Resource A	Assignment		
		Copy AP	P Configu	ration		
			P Configu			
			PP Config		E .	
2	🍃 Manual Pin Allo	cator		(1000000))		<u> </u>
	Manual Pin Allo			a., anato)		
2				1,000000)		
		0 -	Pin Name	Pin Number (Port)		
	Filter ECAT_SSC	0 -	Pin Name	Pin Number (Port)		
	Filter ECAT_SSC	0 -		Pin Number (Port) #73 (P1.10)	_	
	Filter ECAT_SSC	0 - me APP led_ led_	err link_act_p0	#73 (P1.10) #71 (P1.12)		
	Filter ECAT_SSC	me APP	err link_act_p0 link_act_p1	#73 (P1.10) #71 (P1.12) #95 (P0.11)		
	Filter ECAT_SSC	me APP led_ led_ led_ led_	err link_act_p0 link_act_p1 run	#73 (P1.10) #71 (P1.12) #95 (P0.11) #72 (P1.11)		
	Filter ECAT_SSC	me APP	err link_act_p0 link_act_p1 run :	#73 (P1.10) #71 (P1.12) #95 (P0.11) #72 (P1.11) #93 (P3.3)		
	Filter ECAT_SSC	me APP led_ led_ led_ led_ mda mda	err link_act_p0 link_act_p1 run : o	#73 (P1.10) #71 (P1.12) #95 (P0.11) #72 (P1.11) #93 (P3.3) #94 (P0.12)		
	Filter ECAT_SSC	me APP led_ led_ led_ mdk mdi p0_1	err link_act_p0 link_act_p1 run : o ink	#73 (P1.10) #71 (P1.12) #95 (P0.11) #72 (P1.11) #93 (P3.3)		
	Filter ECAT_SSC	Me APP Ied_ Ied_ Ied_ Ied_ Ied_ P0_1 P0_1	err link_act_p0 link_act_p1 run : o	#73 (P1.10) #71 (P1.12) #95 (P0.11) #72 (P1.11) #93 (P3.3) #94 (P0.12) #68 (P1.15)		
	Filter ECAT_SSC	me APP led_ led_ led_ mda mda mda p0_1 p0_r p0_r	err link_act_p0 link_act_p1 run : o ink x_clk	#73 (P1.10) #71 (P1.12) #95 (P0.11) #72 (P1.11) #93 (P3.3) #94 (P0.12) #68 (P1.15) #78 (P1.1)		
	Filter ECAT_SSC	me APP led_ led_ led_ led_ mdi p0_r p0_r p0_r p0_r	err link_act_p0 link_act_p1 run : o ink x_clk x_clk x_clk x_dv x_err xd0	#73 (P1.10) #71 (P1.12) #95 (P0.11) #72 (P1.11) #93 (P3.3) #94 (P0.12) #68 (P1.15) #78 (P1.1) #78 (P1.1) #54 (P2.6) #75 (P1.4)		
	Filter ECAT_SSC	me APP led_ led_ led_ mda mda p0_1 p0_r p0_r p0_r p0_r p0_r	err link_act_p0 link_act_p1 run : o ink x_clk x_clk x_clk x_clk x_dv x_err xd0 xd1	#73 (P1.10) #71 (P1.12) #72 (P1.11) #72 (P1.11) #93 (P0.12) #94 (P0.12) #78 (P1.1) #80 (P1.9) #54 (P2.6) #75 (P1.4) #57 (P5.1)		
	Filter ECAT_SSC	me APP led_ led_ led_ led_ led_ led_ p0_r p0_r p0_r p0_r p0_r p0_r p0_r p0_r	err link_act_p0 link_act_p1 : o ink x_clk x_clk x_clk x_clv x_dv x_err xxd0 xxd1 xxd2	#73 (P1.10) #71 (P1.12) #95 (P0.11) #72 (P1.11) #93 (P0.12) #94 (P0.12) #76 (P1.1) #80 (P1.9) #54 (P2.6) #75 (P1.4) #57 (P5.1) #56 (P5.2)		
	Filter ECAT_SSC	me APP led_ led_ mda mda mda mda mda mda mda mda	err link_act_p0 link_act_p1 : o ink x_clk x_clk x_clk x_clv x_dv x_err xxd0 xxd1 xxd2	#73 (P1.10) #71 (P1.12) #72 (P1.11) #72 (P1.11) #93 (P0.12) #94 (P0.12) #78 (P1.1) #80 (P1.9) #54 (P2.6) #75 (P1.4) #57 (P5.1)		

1 Right click on the ECAT_SSC APP. From the context menu select "Manual Pin Allocator" to open the pin allocation for the EtherCAT module.

2 Inside Manual Pin Allocator you can configure the EtherCAT ports for your application. For the example provided, the configuration fits to the XMC4300 Relax EtherCAT Kit.

Description – Distributed clock support

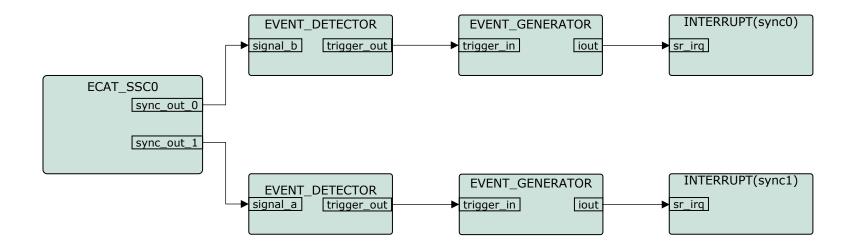




For distributed clock support, the sync0 and sync1 signals coming from the ethercat peripheral are used to trigger interrupts. Inside the interrupt service routines the respective API functions of the SSC protocol stack are called.

Description – Overview on propagating the sync0 and sync1 signals to ISR





EVENT_DETECTOR and EVENT_GENERATOR APPs are instances of the event request unit (ERU) peripheral. Inside this example the ERU is used to propagate the signals sync0 and sync1 to the interrupt service routines.

Please see next slides how to setup this configuration inside DAVETM.

ATTENTION: With the same approach sync0 and sync1 signals can also be connected to other resources. For example: ADC, ports and timers.

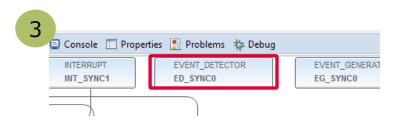


1	
APP Depend	dency 🔀 🕌 HW Signal Connectivity 📮
ECAT_SSC	INTERRUPT
ECAT 📰	Configu e APP Instance
	Rename Instance Label
	Add New Instance
×	Remove
	HW Signal Connections
	Manual Pin Allocator
	Manual Resource Assignment
	Copy APP Configuration
	Paste APP Configuration
	Import APP Configuration
2	
2	
HW Signal Connections	
Filter ECAT_SSC_0	
Filter ECAT_SSC_0	
Source APP Inst	tance Name Source Signal Co Target APP Instance Name Target Signal
2 CAT_SSC_0	sync_out_0 👻> ED_SYNC0 👻 signal_b 💌
	sync_out_1
?	Save Reset Close

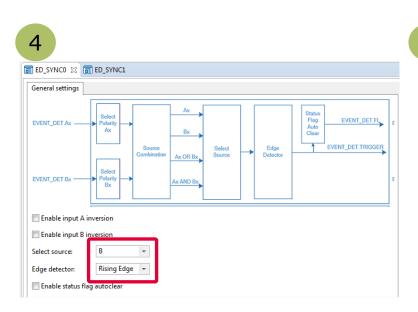
1 Right click on the ECAT_SSC APP. From the context menu select "HW Signal Connections" to open the HW Signal Connection dialog of the ECAT_SSC APP.

2 Connect the sync_out_0 and sync_out_1 signal to the a/b input of the event detection unit.



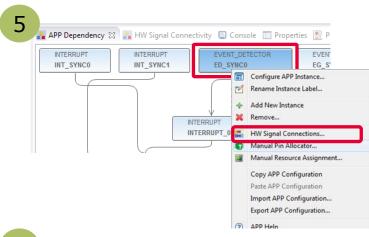


3 Double click on the EVENT_DETECTOR APP for SYNC0 and EVENT_DETECTOR APP for SYNC1.



4 Select the respective source signal ("A" for SYNC0 and "B" for SYNC1) and edge detection "Rising Edge".





HW Sig	nal Connections							
	·							
Filter El	D_SYNC0 🔻							
	Source APP Instance Name	Source Signal		Connect To	Target APP Instan	ce Name	Target Signal	
		Source Signal		Connect To	Target APP Instan	ce Name	Target Signal	
⊿ 🕀	Source APP Instance Name ED_SYNC0				-			
⊿ 🔂		Source Signal trigger_out	- -	Connect To	Target APP Instan		Target Signal trigger_in	Ŧ

HW Signal Connections							
Filter ED_SYNC1 -							
Source APP Instance Name	Source Signal		Connect To	Target APP Instan	ce Name	e Target Signa	
▲ (ED_SYNC1	_						
	trigger_out	Ŧ	>	EG_SYNC1		+ trigger_in	Ŧ
	Not Selected	Ŧ	>	Not Selected		 Not Selected 	-

5 Right click on the EVENT_DETECTOR APP for SYNC0 and SYNC1. From the context menu select "HW Signal Connections " to open the HW Signal Connection dialog of the ECAT_SSC APP.

⁶ Connect the trigger_out signals of the event detection units to the trigger_in signals of the event generation units.



EVENT_GENERA	FOR EVENT_DETECTOR
EG_SYNC0	Configure APP Instance
	Rename Instance Label
+	Add New Instance
×	Remove
	HW Signal Connections
0	Manual Pin Allocator
	Manual Resource Assignment
	Copy APP Configuration
	Paste APP Configuration
	Import APP Configuration
	Export APP Configuration
(?)	APP Help

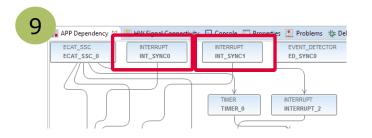
7 Right click on EVENT_GENERATOR for sync0 and sync1. From the context menu select "HW Signal Connections " to open the HW Signal Connection dialog of the EVENT_GENERATOR APP.

	Source Signal	connen	Target APP Instan	ce ivame	Target Sig	Inal
_SYNC0	iout -	· ····>	INT SYNC0		sr ira	
	Not Selected	>	NOT SElected	•	NOT SELECT	eu 🔻
	311100	iout	iout >	iout> INT_SYNC0	iout 👻> INT_SYNC0 👻	iout 👻> INT_SYNC0 💌 sr_irq

🖁 HW Sigi	nal Connections						×
Filter EG	_SYNC1 -						
	Source APP Instance Name	Source Signal		Connect To	Target APP Instance Name	Target Signal	
⊿ 🕀	EG_SYNC1	iout	-	>	INT_SYNC1 -	sr_irq	-
		Not Selected	•	,	NorSciected	Not Selected	

⁸ Connect the iout of the EVENT_GENERATOR APP for sync0 to INTERRUPT APP of sync0. Proceed respectively for sync1.





9 Double click on the INTERRUPT APP for sync0 and INTERRUPT for sync1.

n	
	Interrupt Settings
	Enable interrupt at initialization
	Therrupt Priority
	Preemption priority 30 Subpriority 0
	Interrupt handler: SYNC0IRQHandler

Interrupt Settings
Enable interrupt at initialization
Interrupt Priority
Preemption priority 30 Subpriority 0
Interrupt handler: SYNC1IRQHandler

¹⁰ Set the interrupt service routine for sync0 and sync1 inside the configuration of the respective INTERRUPT APP.



BAVE CE - XMC4300_Relax_EtherCat_APP_Slave_SSC/main.c - DAVE™ - C:\Workspaces\DAVE-4.1\ETHERCAT_DC File Edit Source Refactor Navigate Search Project DAVE Window Help 🖫 🕼 | 🔁 🌌 🌽 | 🕨 🗉 📾 🖄 🗵 👁 🗷 🤜 党 🔞 號 늘 🔤 💭 🗟 🍈 🎒 🖓 📲 🏠 🖉 🔮 🔅 🔹 🗇 🖉 🖉 🖅 🌆 🗸 🖓 🕶 🖓 🕶 🖓 🕶 🔶 📼 료 C/C++ Proj... 💥 🏊 Project Expl... 💼 main.c 🖾 123 124 /** XMC4300_Relax_EtherCat_APP_Slave_SSC [125 Binaries * @brief SYNCØIRQHandler() - EtherCAT Interrupt Routine for SYNCØ 126 Includes 127 128 * Details of function
 Dave * This routine is handling the SYNCO Interrupts and need to call the SSC Stack 129 🕞 Debug 130 b > > Libraries void SYNC0IRQHandler (void) 13: SSC >> 13: { 133 Sync0 Isr(); b > > Startup 13 c main.c 13 IInker_script.id 136 /** solver.bak 137 > 25 XMC4800 Relax EtherCat APP Slave SSC @brief SYNC1IRQHandler() - EtherCAT Interrupt Routine for SYNC1 138 139 140 * Details of function
 * This routine is handling the SYNC1 Interrupts and need to call the SSC Stack 141 14 143 void SYNC1IRQHandler (void) 144 14 Sync1 Isr(); 14

Inside main() the interrupt handlers for sync0 and sync1 are implemented. The implementation is calling the respective functions of the SSC protocol stack.

Description – SSC specific enabling/disabling of interrupts [1/2]



Please see ET9300 application note published by the ETG on details about the SSC code structure and interrupt handling (chapter 4).

In v1.8/2017-11-14 of this document inside chapter 5/hardware access it is specified:

"If interrupts are used also two macros shall be defined "ENABLE_ESC_INT" and "DISABLE_ESC_INT". These shall enable/disable **all four interrupt** sources".

These macros are implemented inside ECAT_APP. Timer- and PDIinterrupt are handled by the ECAT_APP. As Sync0 and Sync1 are routed through ERU (see before) these interrupts need to be handled in addition by the user.

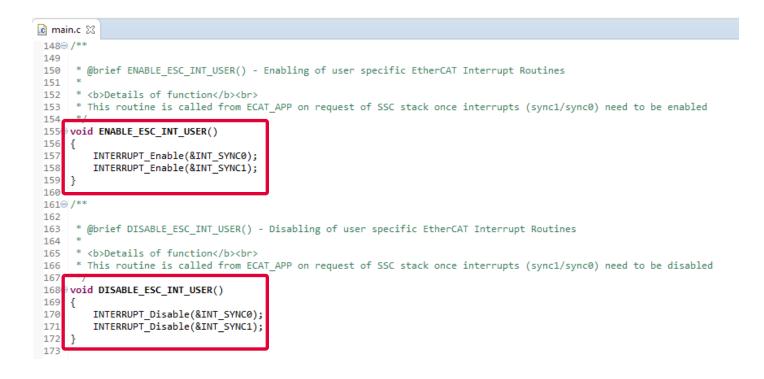
For this purpose ECAT_APP is implementing a callback function for user specific implementation:

ENABLE_ESC_INT_USER and DISABLE_ESC_INT_USER.

Description – SSC specific enabling/disabling of interrupts [2/2]



Within this example you find the implementation of ENABLE_ESC_INT_USER and DISABLE_ESC_INT_USER inside main.c:



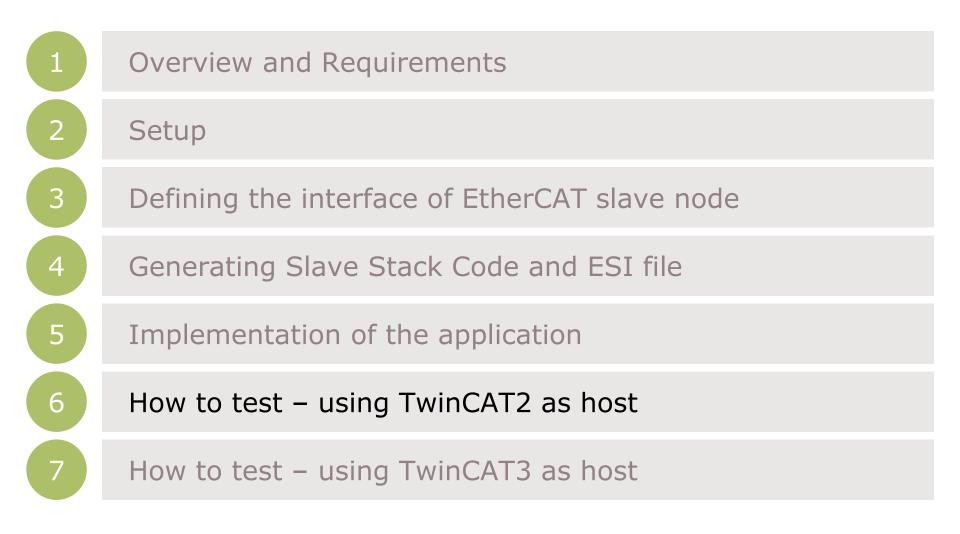


Description – initialization inside main.c

© DAVE CE - XMC4300_Relax_EtherCat_APP_Slave_SSC/main.c - DAVE™ - C:\Workspaces\DAVE-4.1\ETHERCAT_DC					
File Edit Source Refactor Navigate Search F	roject DAVE Window Help				
1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -	र 🔀 🖄 📩 🔤 🕒 🖬 🛛 🖻 🔅	• 😕 🖋 • 📝 🖢 • 🖗 • 득 <			
🔁 C/C++ Proj 🔀 🏠 Project Expl 😐 🗖	͡∎ main.c 🛛				
(→ → @ 🖻 🔄 ▽	99⊖ int main(void)				
State:	<pre>100 { 101 DAVE_STATUS_t status; 102 uint32_t timer=0;</pre>				
Includes	<pre>103 status = DAVE_Init();</pre>	<pre>/* Initialization of DAVE</pre>			
 Dave Debug Libraries SSC Startun Startun Inker_script.ld solver.bak XMC4800_Relax_EtherCat_APP_Slave_SSC 	<pre>104 105 Init_ECAT_Adapt_LED (); 106 Init_Relax_Button(); 107 108 if(status == DAVE_STATUS_FAILURE) 109 { 110 /* Placeholder for error handler of 111 XMC_DEBUG("DAVE APPs initialization 112 while(1U) 113 { 114 } 115 } 116 117 /* Placeholder for user application 118 while(1U) 119 { 120 MainLoop(); 121 } 122 }</pre>	on failed\n");			
	*				

Inside main() DAVE and its APPs (PWM_CCU8, ECAT_SSC) are initialized. InitECAT_Adapt_LED() and Init_Relax-Button() are used to initialize the buttons and LED1 to 8 of the "XMC4300 Relax EtherCAT Kit". Finally the MainLoop is called cyclically to process the state machine of the slave stack code.

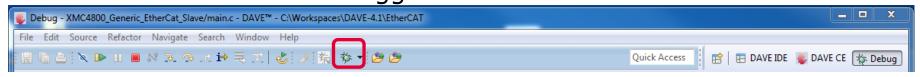






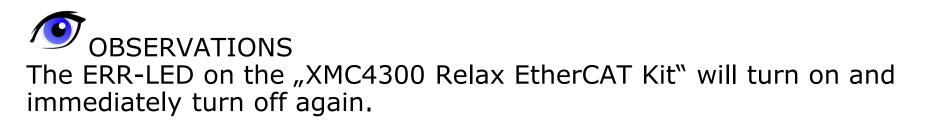
How to test - start the slave to run

ACTIONS 1. Build and download the example application software to the XMC4300 and start the debugger



2. Start the software by the run button





How to test – start the TwinCAT 2 master to run (1/4)



	🛃 Ur	ntitled - TwinCAT System Manager	
1	File	Edit Actions View Options Help	
		📽 📽 🖬 🎯 🖪 🕺 🗛 8	•
		SYSTEM - Configuration	_
		PLC - Configuration	
		1/0 - Configuration	`
		I/O Devi Append Device	
		😭 I <u>m</u> port Device	
		📉 Scan Devices	
		Paste Ctrl+V	
		Paste with Links Alt+ Ctrl+ V	
2	Insert Dev		-
	Туре:	HI/O Beckhoff Lightbus HI/O II/O Lightbus FC200x, PCI	Ok
		II/O II/O Lightbus C1220, ISA II/O II/O Lightbus Master EL6720, EtherCAT	Cancel
		II/O Lightbus Master CX1500-M200, PC104	
		II/O Lightbus Slave CX1500-B200, PC104	
		CANopen DeviceNet / Ethernet I/P	Target Type-
			PC only
			C CX only
		EtherCAT	C BX only
		EtherCAT Automation Protocol via EL6601, EtherCAT Ethernet	C All
	Name:	Device 2	



After starting the TwinCAT System Manager from windows start menu:

1 Right Click I/O-Devices and select "Append Device…"

2 Create an EtherCAT master device by double click

How to test – start the TwinCAT 2 master to run (2/4)



3	General Adapte	r EtherCAT Onli
	⊢ ⊙ Network	Adapter
		OS (NE)
	Description:	Local Area
	Device Name	
	PCI Bus/Slo	t:
	MAC Address	s: d8 9d 67 d
	IP Address:	0.0.0.0 (0.
		Promise
		Virtual
	C Adapter F	Reference
	Adapter:	
	Freerun Cycle (n	ns): 4
4	File Edit Act	ions View Op
	i D 😅 📽 🖡	
	SYSTEM	- Configuration nfiguration

□ ● Network Adapte					
	 OS (NDIS)	C PCI		O DPRAM	
Description:	Local Area Conne	ction (Intel(R) 8	32579L	M Gigabit Netwo	ork Conne
Device Name:	\DEVICE\{03DA	E701-02D1-42A	0-BCE	C.1C1D821B48	51}
PCI Bus/Slot:				Search	
MAC Address:	d8 9d 67 d3 11 6	6		Compatible De	evices
IP Address:	0.0.0.0 (0.0.0.0)		-L		
	Promiscuous N Virtual Device		Netmor	n/Wireshark only)
C Adapter Referen	nce				
Adapter:					7
Freerun Cycle (ms): File Edit Actions	4 +	Help			
🗅 🚅 📽 🖬 🍯) 🖪	l 🙃 🗸 🎒	•
	ation 1 (Ether	nd Box			
‱ ≦≚ Mappings	X Delete	-			
	R Online				
		: <u>N</u> esei : Reload (Cont	fiq Mo	de only)	
		e Delete (Conf	-		
	😭 Export	Device			
		t Box			
	📉 Scan E	Boxes			

ne CoE - Online

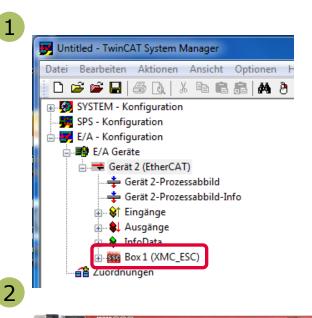


3 Select the network adapter you want to use (search and select). Application hint: In case the device is not found please install the respective device driver by following the instructions given by TwinCAT through the "Compatible Devices…" button.

Right Click EtherCAT master and select "Scan Boxes..."

How to test – start the TwinCAT 2 master to run (3/4)

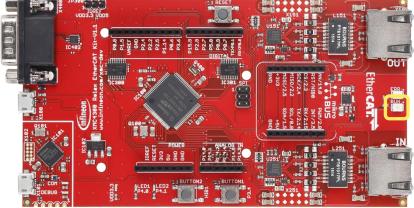






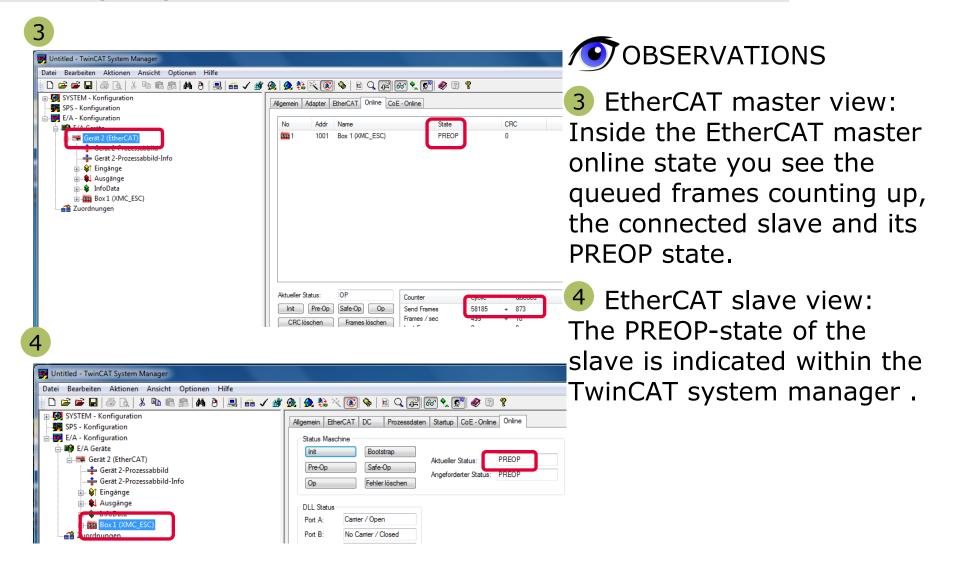
1 The slave appears as a node on the EtherCAT master bus.

2 The RUN-LED is flashing indicating PREOP-state



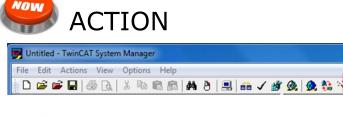
How to test – start the TwinCAT 2 master to run (4/4)





How to test – Setting slave to operational mode

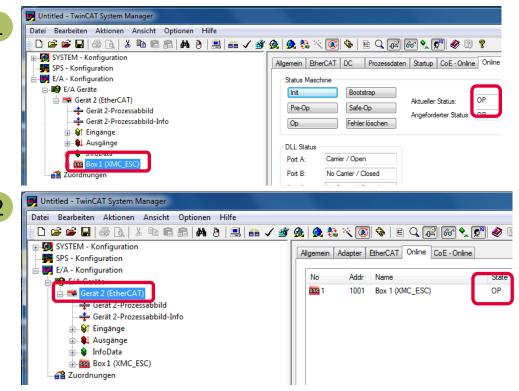




act

Set master device to free run mode

✓ OBSERVATIONS



1 EtherCAT slave view: Online status of slave shows the slave in OP state

2 EtherCAT master view: Online status of master shows the slave in OP state. Cyclic counter is incrementing.

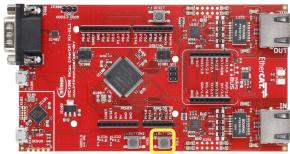
3 "XMC4300 Relax EtherCAT Kit": RUN-LED is static turned on indicating OP-state.

How to test – Monitoring slave inputs on master





While pushing BUTTON1 on "XMC4300 Relax EtherCAT Kit" the button state is updated on the host.





📴 Untitled - TwinCAT System Manager	
Datei Bearbeiten Aktionen Ansicht Optionen Hilfe	
D 🚅 📽 🖬 🚑 [k, % 🖻 🖻 🙈 M 🤌 🖳 🔒 🖌 🗳	<u>&</u> 👲 🗞 🔨 🚱 🗣 E 🔍 🔐 🞯 🗙 🔊 🖉 🦉
👜 🔩 SYSTEM - Konfiguration	Variable Flags Online
E/A - Konfiguration	Wert: 1
⊨-∰ E/A Geräte 	Neuer Wert: Force Aufheben Schreiben
Gerät 2-Prozessabbild	
Gerät 2-Prozessabbild-Info	Kommentar:
😥 🚯 Eingänge	
😥 🕸 🌡 Ausgänge	
😥 💀 😥 InfoData	
Box 1 (XMC_ESC)	
Section 10 CEN IN T	
····· �↑ IN_GEN_INT1 ····· �↑ IN_GEN_INT2	
↓ IN_GEN_INT3	
AT IN COMPACT	
IN_GEN_Bit1	
\$↑ IN_GEN_Bit3	
····◆↑ IN_GEN_Bit4	
····· �↑ IN_GEN_Bit5	

State of IN_GEN_Bit1 changes according to the state of BUTTON1.

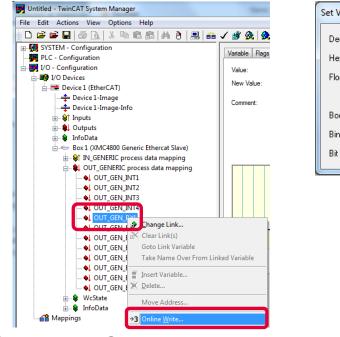
How to test – Setting slave outputs on master (1/2)



ACT

ACTIONS

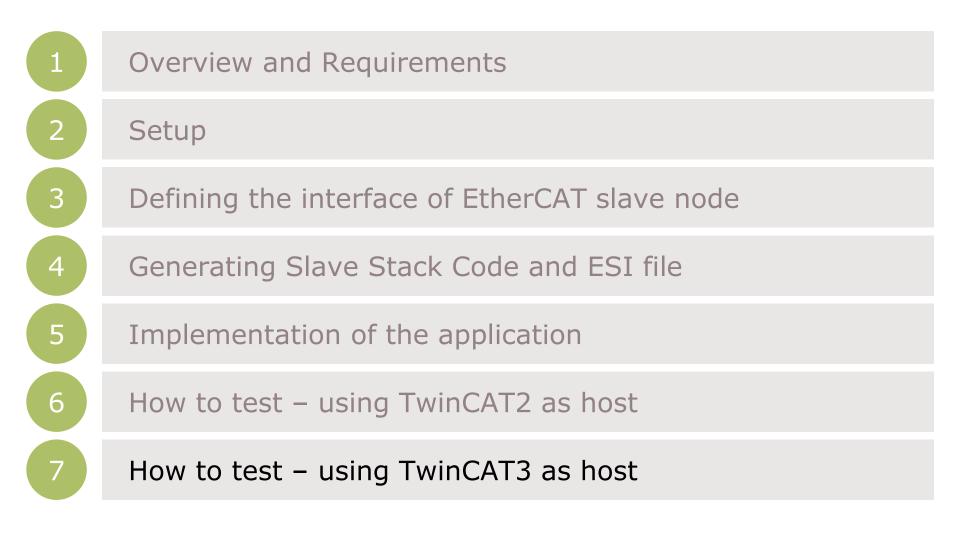
Right click on OUT_GEN_Bit1 of the slave node and select "Online Write..." inside the context menu. Change the value from 0 to 1 to switch on LED1 from 1 to 0 to switch off LED1.



Set Value D	ialog	x
Dec:	1	ОК
Hex:	0x01	Cancer
Float:	1	
Bool:	0 1	Hex Edit
Binary:	01	
Bit Size:	◉1 ◎8 ◎16 ◎32 (64 🔘 ?

OBSERVATION LED1 "XMC4300 Relax EtherCAT Kit" is turned on/off according to OUT_GEN_Bit1 setting.







How to test - start the slave to run

ACTIONS L. Build and download the ex

1. Build and download the example application software to the XMC4300 and start the debugger



2. Start the software by the run button

Sebug - XMC4800_Generic_EtherCat_Slave/main.c - DAVE™ - C:\Workspaces\DAVE-4.1\EtherCAT	
File Edit Source Refactor Navigate Search Window Help	
🗄 🕼 🖻 (♥ 📭) I 🔎 🕅 🌫 👁 🕪 🗮 🕹 (1/4 🙀 🎋 ♥ 😂 😕	Quick Access 🛛 🔁 🛛 DAVE IDE 🛛 😺 DAVE CE 🎼 Debug



How to test – start the TwinCAT 3 master to run (1/4)



File Edit View Project Build Debug TwinCAT PLC To Image: Image	
Solution Explorer × Solution Explorer × Substitution Explorer × Subs	
2 Insert Device	
	Ok Cancel et Type Conly
##/ SERCOS interface Image: Figure 1 ##0 Beckhoff Lightbus Image: Construction 1 ##5 Beckhoff Lightburg Image: Construction 2	X only X only II



After starting the TwinCAT System Manager from windows start menu:

1 Right Click I/O-Devices and select "Add New Item…"

2 Create an EtherCAT master device by double click

How to test – start the TwinCAT 3 master to run (2/4)

Compatible Devices.

Scan



9	General Adapter Ether	CAT Online CoE - Online	
	Network Adapter		
		• OS (NDIS)	
	Description:	LAN-Verbindung (TwinCAT-Intel PCI Ether	rnet Adapter (Gigabit))
	Device Name:	\DEVICE\{44658C39-F4B6-49DB-98AE-61	6E0DC68EE9}
	PCI Bus/Slot:		Search
	MAC Address:	38 63 bb b6 04 60	Compatible Devic
	IP Address:	169.254.115.19 (255.255.0.0)	
		Promiscuous Mode (use with Wireshard	k only)
		Virtual Device Names	
	Adapter Reference	9	
	Adapter:		
4	▲ 🔽 I/O		
	▲ ➡ Device	Add New Item	
	100 mag == 100 mag ===		
	🗢 Sync 🗙	Remove	
	⊳ 🕒 Inpu ⊳ 🖷 Out;	Change NetId	
	Infol	Save Device 1 (EtherCAT) As	
	📸 Mappings	Append EtherCAT Cmd	
		Append Dynamic Container	
		Online Reset	
		Online Reload	

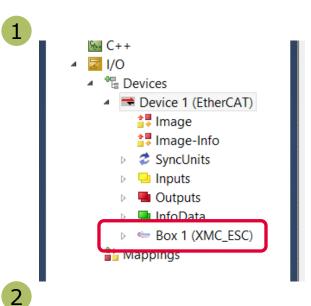


3 Select the network adapter you want to use (search and select). **Application hint:** In case the device is not found please install the respective device driver by following the instructions given by TwinCAT through the "Compatible Devices…" button.

Right Click EtherCAT master and select "Scan Boxes..."

How to test – start the TwinCAT 3 master to run (3/4)

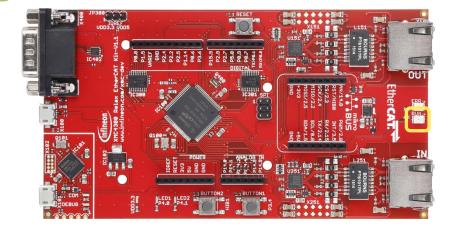






1 The slave appears as a node on the EtherCAT master bus.

2 The RUN-LED is flashing indicating PREOP-state



How to test – start the TwinCAT 3 master to run (4/4)

XMCProject 1 - Microsoft Vis



 Solution Explorer
 • * XMCProject 1 × Start Page

		·비·비·국 ·비·비·국		Frame Selection
	pject_1 × Start Page		- 10. Not - mg -4	-
	ral Adapter EtherCAT	Opling CoE Opling		
Solution XMCProject_1' (1 project) ■ XMCProject_1 (1 project) ■ XMCProject_1 ■ SYSTEM ■ MOTION ■ PLC ■ SAFETY ■ C++ ■ Z I/O ■ Device 1 (EtherCAT) ■ Image-Info ■ Z SyncUnits ■ Inputs	Ad Name	ESC) PREOP C	CRC	
 InfoData e Box 1 (XMC_ESC) Mappings 	Init Pre-Op Safe	Op Op Send Frames Frames / sec Lost Frames Tx/Rx Errors	0 + 11396 + 90 + 0 0 + 0 0 / 17	
Solution XMCProject_1 (1 project)	State Machi	20.		
XMCProject_1 SYSTEM	Init	Bootstrap		
MOTION			Current State:	PREOP
I PLC	Pre-Op	Safe-Op	Requested State:	
SAFETY	Op	Clear Error		
M C++ ▲ Z I/O	DLL Status			
 Bevices 		Carrier / Open		
 Device 1 (EtherCAT) 	Port A:			
🛟 Image	Port B:	No Carrier / Closed		
🛟 Image-Info	Port C:	No Carrier / Closed		
SyncUnits	Port D:	No Carrier / Closed		
Inputs				
Outputs	File Access	over EtherCAT		
A 🖙 Box 1 (XMC ESC)	Downlo	ad Upload		



3 EtherCAT master view: Inside the EtherCAT master online state you see the queued frames counting up, the connected slave and its PREOP state.

4 EtherCAT slave view: The PREOP-state of the slave is indicated within the TwinCAT system manager.

How to test – Setting slave to operational mode

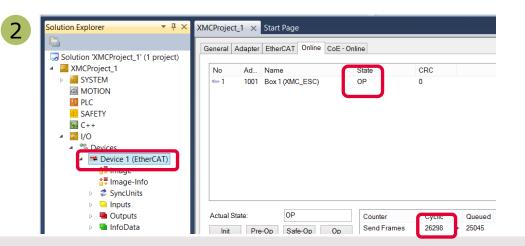




Set master device to free run mode

• OBSERVATIONS

ile Edit View Project Build I	Debug TwinCAT PLC Tools Scope Window Help
🔛 🔤 🧔 🛠 🐻 🔐	li 🖓 🖓 🖓 🦕
	- 三 🕣 🍕 (三 雪 唐 ひ) 🝐 沓 沓 じ じ 🚦 🖼 😫 関 🎉 💭 🏛 🔘 Frame Selection
Solution Explorer 🚽 🖵	× XMCProject_1 × Start Page
	General EtherCAT DC Process Data Startup CoE - Online Online
Solution 'XMCProject_1' (1 project) State Machine
XMCProject_1	
SYSTEM	Init Bootstrap
A MOTION	Current State: OP
PLC	Requested State: OP
SAFETY	Op Clear Error



1 EtherCAT slave view: Online status of slave shows the slave in OP state

2 EtherCAT master view: Online status of master shows the slave in OP state. Frames are no more queued. Cyclic counter is incrementing.

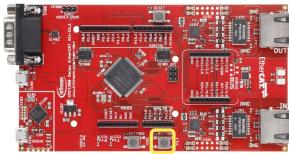
3 "XMC4300 Relax EtherCAT Kit": RUN-LED is static turned on indicating OP-state.

How to test – Monitoring slave inputs on master





While pushing BUTTON1 on "XMC4300 Relax EtherCAT Kit" the button state is updated on the host.





Solution Explorer 🔹 👎 🗙	XMCProject_1 × Start Page	
	Variable Flags Online	
Solution 'XMCProject_1' (1 project) Solution 'XMCProject_1	Value: 1	
AMEPROJECT SYSTEM		
MOTION	New Value: Force Release Write	
PLC	Comment:	
SAFETY		
▲ 🔽 I/O		
▲ 📲 Devices		
 Device 1 (EtherCAT) 	✓	
Image 1. Image-Info		
 SyncUnits 		
Inputs		
Outputs		
▷ 🖳 InfoData ◢ 🗢 Box 1 (XMC_ESC)		
 BOX F(xime_LSC) IN_GENERIC process 	╽║║┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┦╿┽╜╟╌╢	
IN_GEN_INT1		
IN_GEN_INT2		
IN_GEN_INT3		
✓ IN_GEN_Bit1		

State of IN_GEN_Bit1 changes according to the state of BUTTON1.

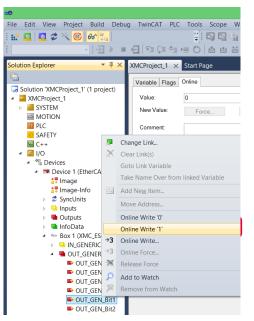
How to test – Setting slave outputs on master (1/2)



NOW

ACTIONS

Right click on OUT_GEN_Bit1 of the slave node and select "Online Write..." inside the context menu. Change the value from 0 to 1 to switch on LED1 from 1 to 0 to switch off LED1.



OBSERVATION LED1 "XMC4300 Relax EtherCAT Kit" is turned on/off according to OUT_GEN_Bit1 setting.



Part of your life. Part of tomorrow.



X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Development Boards & Kits - ARM category:

Click to view products by Infineon manufacturer:

Other Similar products are found below :

SAFETI-HSK-RM48 PICOHOBBITFL CC-ACC-MMK-2443 EVALSPEAR320CPU TMDX570LS04HDK TXSD-SV70 TXSD-SV71 YGRPEACHNORMAL PICODWARFFL YR8A77450HA02BG 3580 32F3348DISCOVERY ATTINY1607 CURIOSITY NANO PIC16F15376 CURIOSITY NANO BOARD PIC18F47Q10 CURIOSITY NANO VISIONSTK-6ULL V.2.0 DEV-17717 EAK00360 YR0K77210B000BE RTK7EKA2L1S00001BE SLN-VIZN-IOT LV18F V6 DEVELOPMENT SYSTEM READY FOR AVR BOARD READY FOR PIC BOARD READY FOR PIC (DIP28) AVRPLC16 V6 PLC SYSTEM MIKROLAB FOR AVR XL MIKROLAB FOR PIC L MINI-AT BOARD - 5V MINI-M4 FOR STELLARIS MOD-09.Z BUGGY + CLICKER 2 FOR PIC32MX + BLUETOOT 1410 LETS MAKE PROJECT PROGRAM. RELAY PIC LETS MAKE - VOICE CONTROLLED LIGHTS LPC-H2294 DSPIC-READY2 BOARD DSPIC-READY3 BOARD MIKROBOARD FOR ARM 64-PIN MIKROLAB FOR AVR MIKROLAB FOR AVR L MIKROLAB FOR DSPIC MIKROLAB FOR DSPIC XL MIKROLAB FOR PIC32 MIKROLAB FOR TIVA EASYAVR V7 EASYMX PRO FOR TIVA C SERIES EASYMX PRO V7 FOR STM32 EASYPIC FUSION V7 MINI-32 BOARD