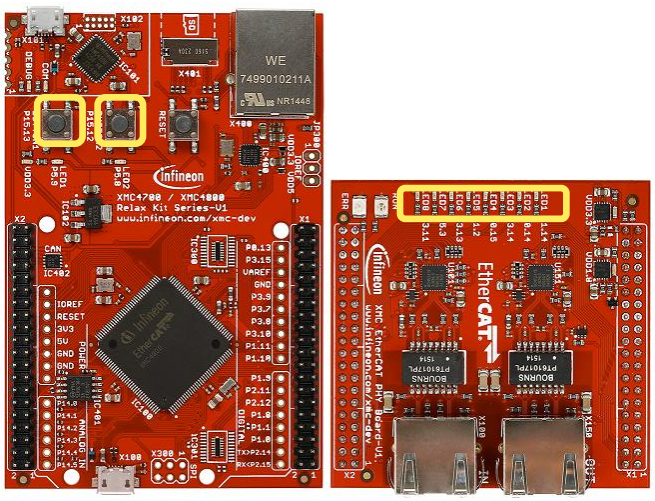
XMC4800 EtherCAT App and SSC Slave Example

Getting Started V1.1

1. 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

1. 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

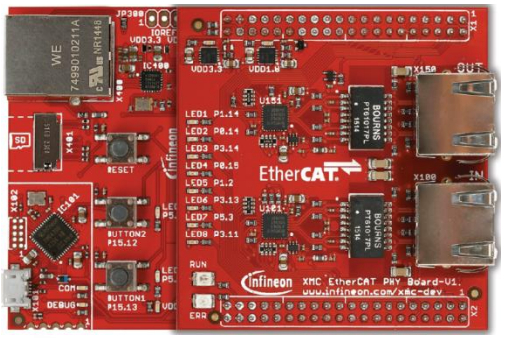
Overview



This example demonstrates the implementation of a EtherCAT slave node using the Beckhoff SSC Tool to generate the slave stack code for „XMC4800 Relax EtherCAT Kit“.

While reviewing this example you will see in output direction the EtherCAT master controlling the 8 LEDs on the „XMC EtherCAT PHY Board. In input direction you will monitor inside the master device the status of the buttons available on the Relax Kit.
You will observe inside source code how you can 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 your needs.

Requirements



XMC4800 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

Micro USB Cable (Debugger connector)

Requirements - free downloads

TwinCAT2 (30 day trial)
(used inside this document as reference)
Link: Download TwinCAT2

or

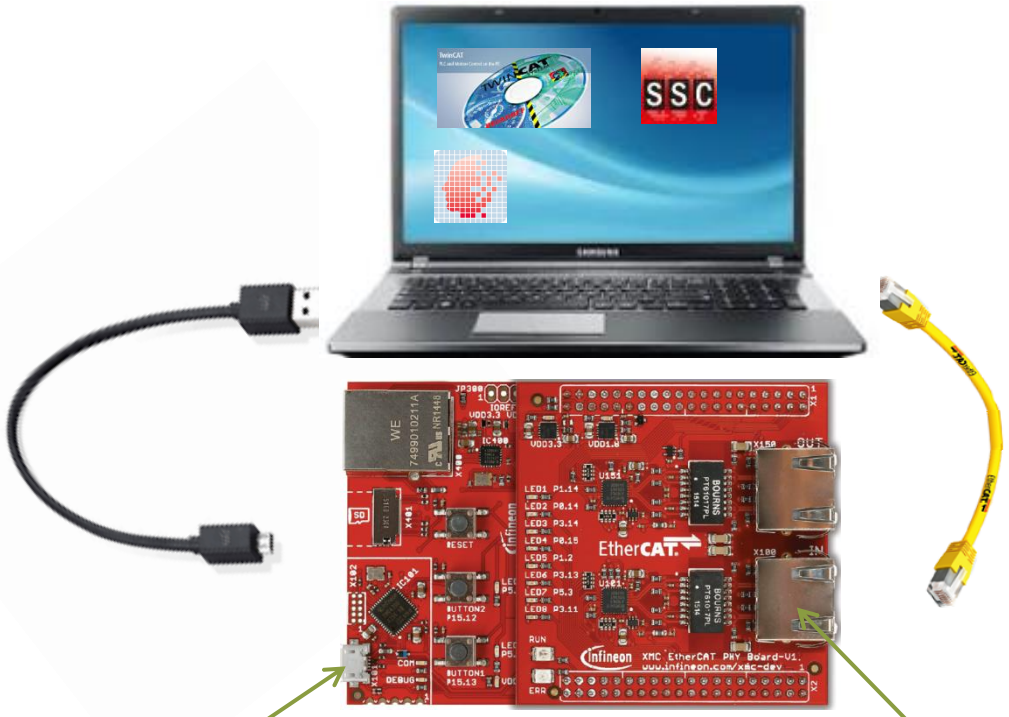
TwinCAT3 (no trial period; usability limited)
Link: Download TwinCAT3

DAVE (v4.1.4 or higher)
Link: Download DAVE (Version 4)

EtherCAT Slave Stack Code Tool
(ETG membership obligatory)
Link: Slave Stack Code Tool

1. 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

Setup
Hardware connection

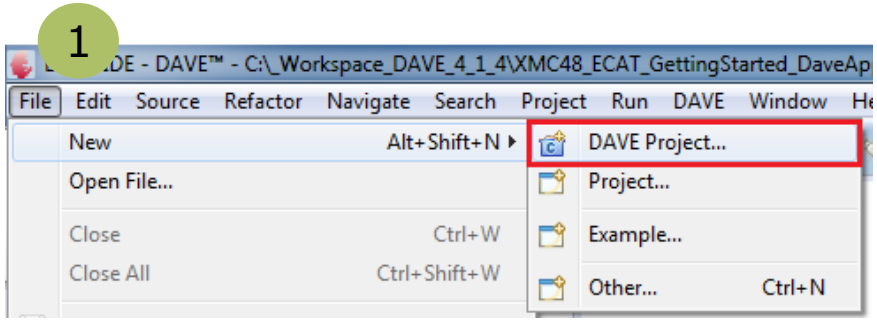


Micro USB cable
Debugger connected to
X101 debug connector

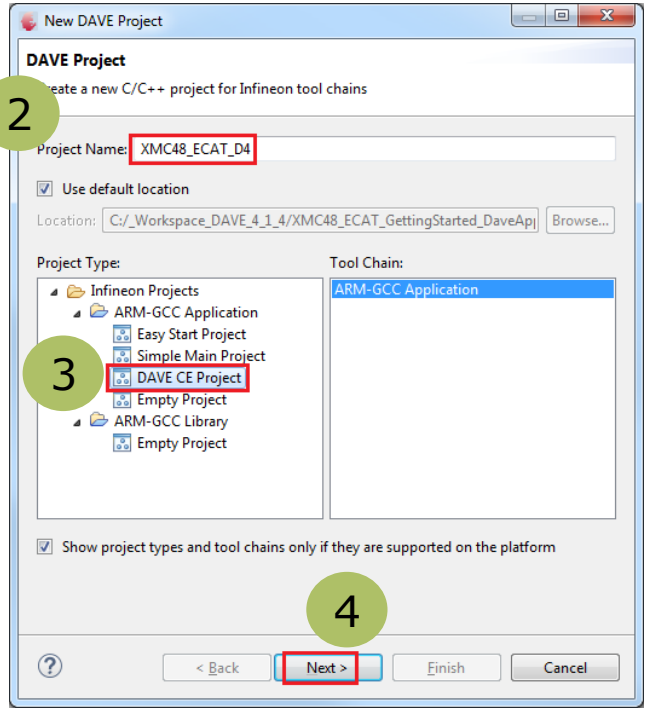
Ethernet Cable
connected to IN-port

Setup
Creating a new DAVE4 project

1. Click on File >> **New** >> **Dave Project**



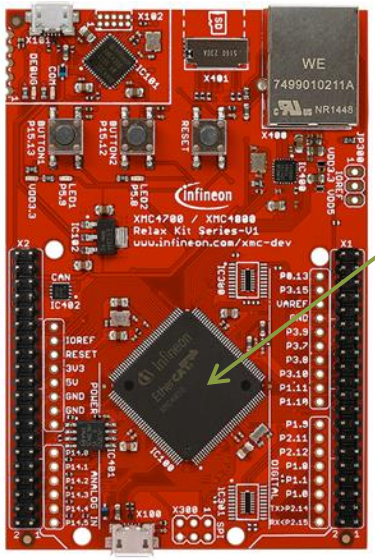
2. Provide a **name** (e.g. XMC48_ECAT_D4) to your project
3. In the Project Type select "**DAVE CE Project**"
4. Click on "**Next**"

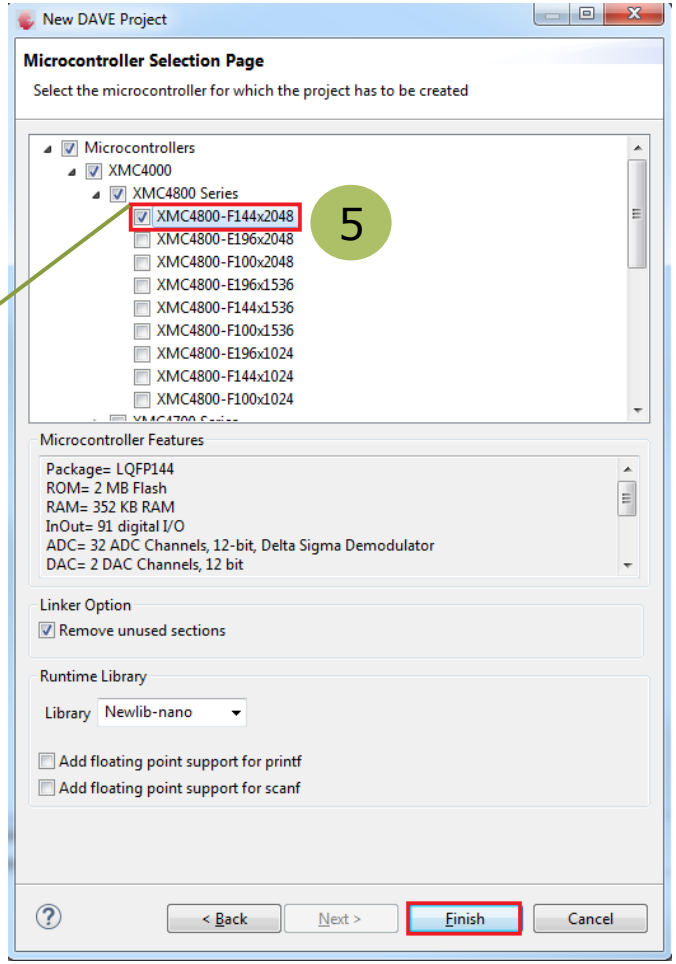


Setup
Creating a new DAVE4 project

5 Select the Microcontroller which is being used in the XMC4800 relaxkit.

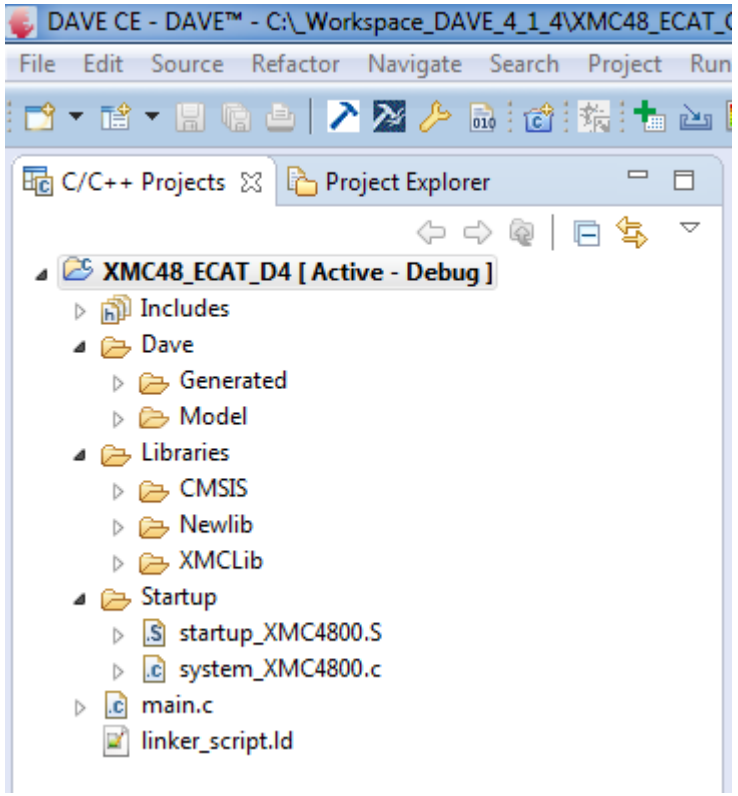
› For this case it is **XMC4800–F144x2048**





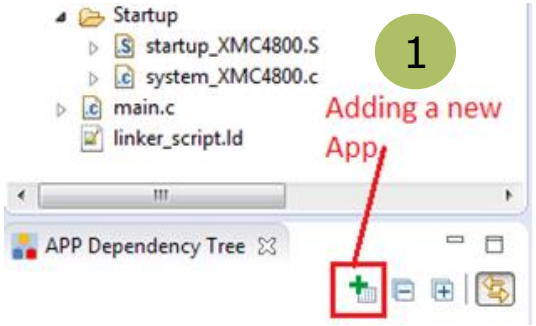
Setup
Creating a new DAVE4 project

- Here you should be able to see that a DAVE4 project is created with the following files and folders.



Setup
App configuration

1. Click on the on the "**Add new App**" icon.



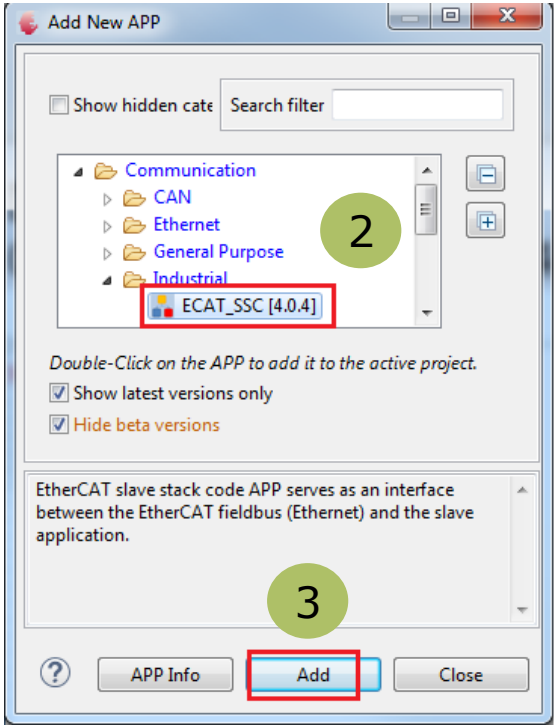
- With that, the **Add New App** window will appear to provide a selection of Apps.

2. Click on "**Communication**" >> "**Industrial**"

- Select the app "**ECAT_SSC**"

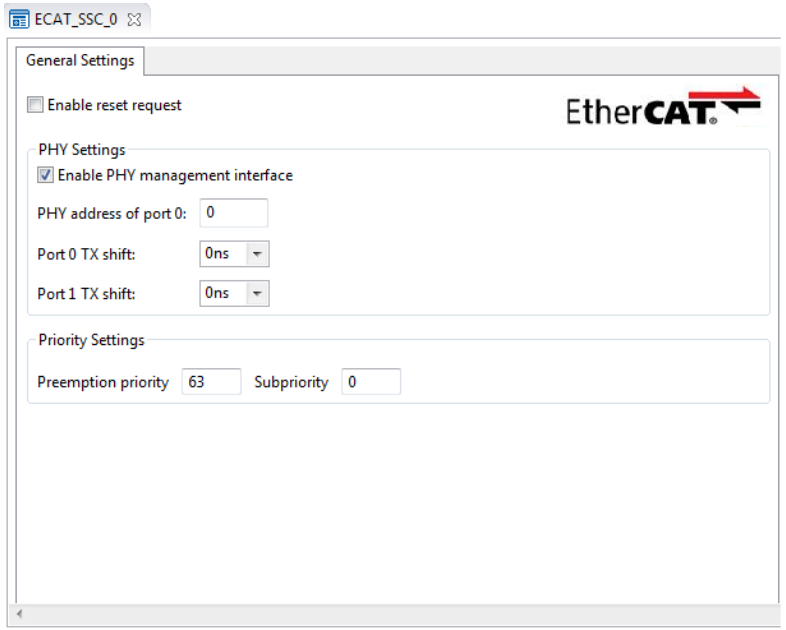
3. Add the selected App by clicking on the "**Add**" button.

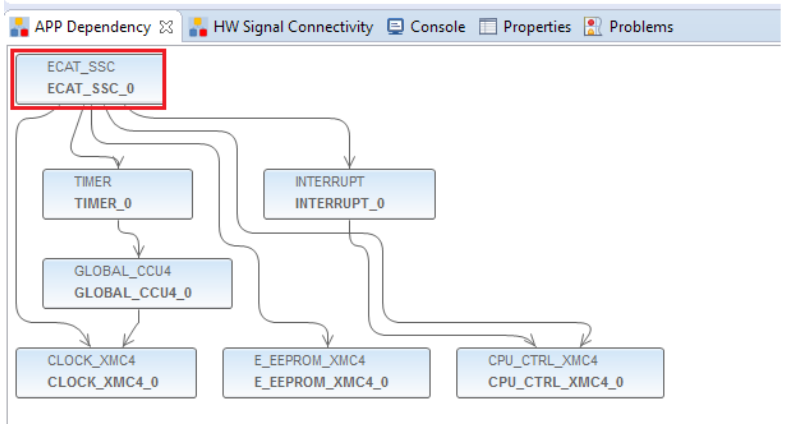
- Finally click on the "**Close**" button after the App is added to the project.



Setup
App configuration

- General Setting of the EtherCAT apps can be done by clicking on the **ECAT_SSC_0 app**
- However, we stay with the default configurations.



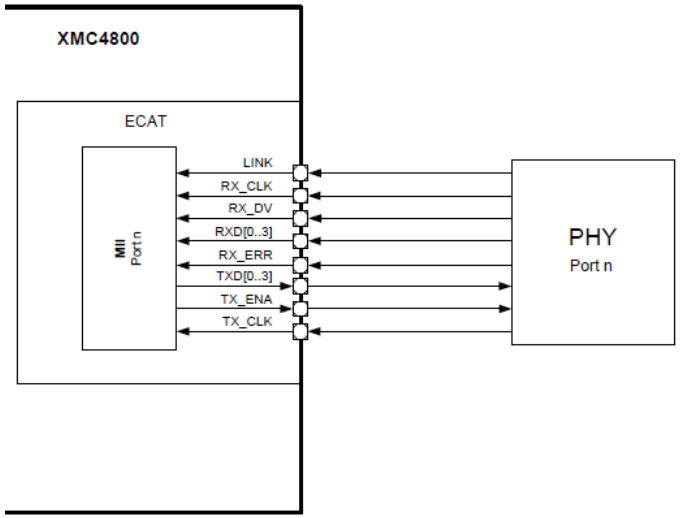


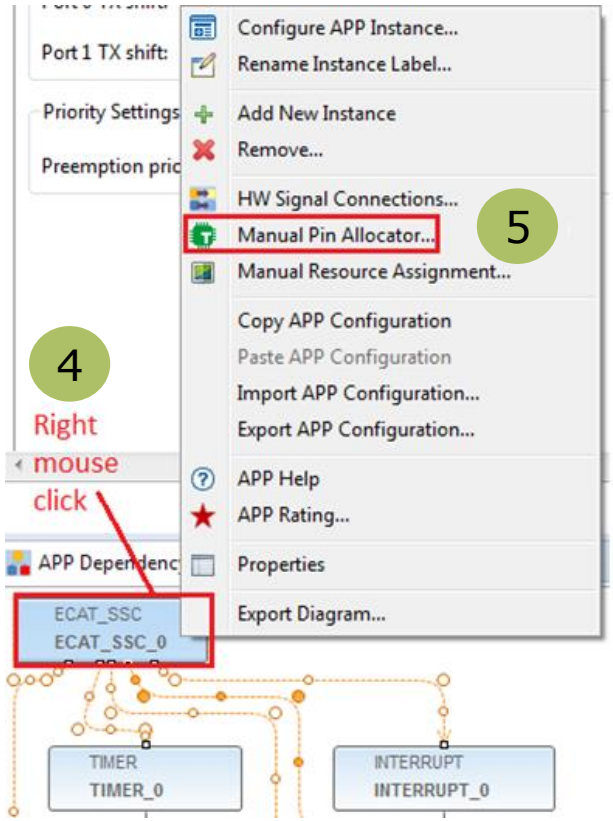
Setup
App configuration

- Now we need to manually assign the XMC pins connection to the PHY

4 To do so, right mouse click on **ECAT_SSC_0** app.

5 Then select **Manual Pin Allocator**





Setup
App configuration

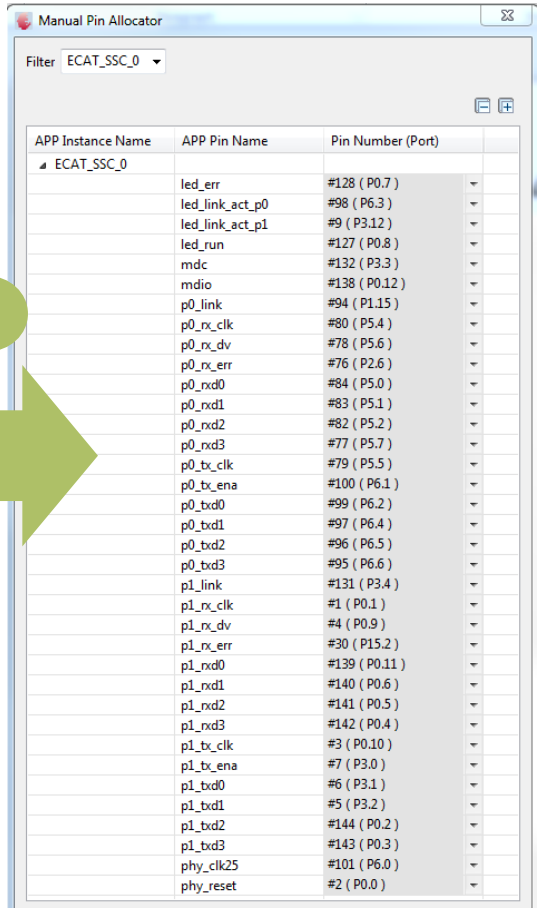
- The PHY configuration consist of **36** IO pins which is connected to the MII ports of the EtherCAT module.

6

- So map the IOs accordingly as shown.

	A	B	C	D
1	APP Instance Name	Resource	Port-Pin	Pin Number
2	ECAT_SLAVE_0	led_err	(P0.7)	#128
3	ECAT_SLAVE_0	led_link_act_p0	(P6.3)	#98
4	ECAT_SLAVE_0	led_link_act_p1	(P3.12)	#9
5	ECAT_SLAVE_0	led_run	(P0.8)	#127
6	ECAT_SLAVE_0	mdc	(P3.3)	#132
7	ECAT_SLAVE_0	mdio	(P0.12)	#138
8	ECAT_SLAVE_0	p0_link	(P1.15)	#94
9	ECAT_SLAVE_0	p0_rx_clk	(P5.4)	#80
10	ECAT_SLAVE_0	p0_rx_dv	(P5.6)	#78
11	ECAT_SLAVE_0	p0_rx_err	(P2.6)	#76
12	ECAT_SLAVE_0	p0_rxd0	(P5.0)	#84
13	ECAT_SLAVE_0	p0_rxd1	(P5.1)	#83
14	ECAT_SLAVE_0	p0_rxd2	(P5.2)	#82
15	ECAT_SLAVE_0	p0_rxd3	(P5.7)	#77
16	ECAT_SLAVE_0	p0_tx_clk	(P5.5)	#79
17	ECAT_SLAVE_0	p0_tx_ena	(P6.1)	#100
18	ECAT_SLAVE_0	p0_txd0	(P6.2)	#99
19	ECAT_SLAVE_0	p0_txd1	(P6.4)	#97
20	ECAT_SLAVE_0	p0_txd2	(P6.5)	#96
21	ECAT_SLAVE_0	p0_txd3	(P6.6)	#95
22	ECAT_SLAVE_0	p1_link	(P3.4)	#131
23	ECAT_SLAVE_0	p1_rx_clk	(P0.1)	#1
24	ECAT_SLAVE_0	p1_rx_dv	(P0.9)	#4
25	ECAT_SLAVE_0	p1_rx_err	(P15.2)	#30
26	ECAT_SLAVE_0	p1_rxd0	(P0.11)	#139
27	ECAT_SLAVE_0	p1_rxd1	(P0.6)	#140
28	ECAT_SLAVE_0	p1_rxd2	(P0.5)	#141
29	ECAT_SLAVE_0	p1_rxd3	(P0.4)	#142
30	ECAT_SLAVE_0	p1_tx_clk	(P0.10)	#3
31	ECAT_SLAVE_0	p1_tx_ena	(P3.0)	#7
32	ECAT_SLAVE_0	p1_txd0	(P3.1)	#6
33	ECAT_SLAVE_0	p1_txd1	(P3.2)	#5
34	ECAT_SLAVE_0	p1_txd2	(P0.2)	#144
35	ECAT_SLAVE_0	p1_txd3	(P0.3)	#143
36	ECAT_SLAVE_0	phy_clk25	(P6.0)	#101
37	ECAT_SLAVE_0	phy_reset	(P0.0)	#2

6

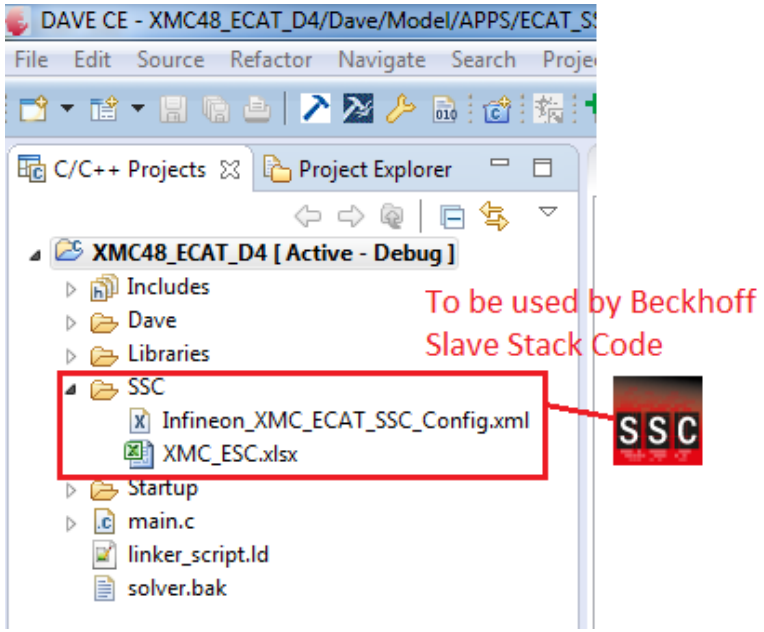


Setup
App configuration

- Now we shall generate the codes base on the configurations of the Apps.

7 Click on "**Generate codes**" icon.

- After the code generation process.
- A **SSC** (Slave Stack Code) folder is created with the following files,
  - **Infineon_XMC_ECAT_SSC_Config.xml** (To be used by Slave stack code)
  - **XMC_ESC.xlsx** (To be used for variable mapping)

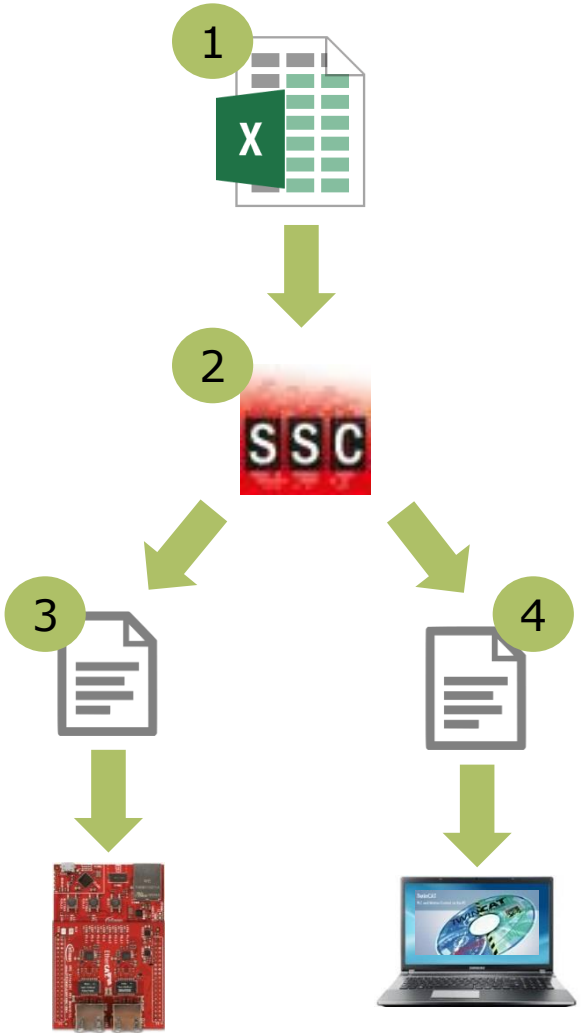


Important note !!!

The next step is to create the Slave Stack Code and ESI (EtherCAT Slave Information) file in folder SSC.

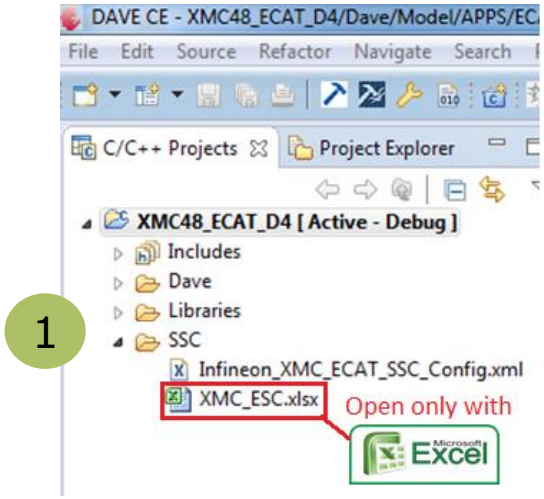
1. Overview and Requirements
2. Setup
3. Defining the interface of EtherCAT slave node
4. Generating Slave Stack Code and ESI file
5. Implementation of the application
6. How to test

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 XMC4800.
4. The generated **E**therCAT **S**lave **I**nformation file (ESI) does apply for the EtherCAT host. There the relevant interface information about the slave is stored.

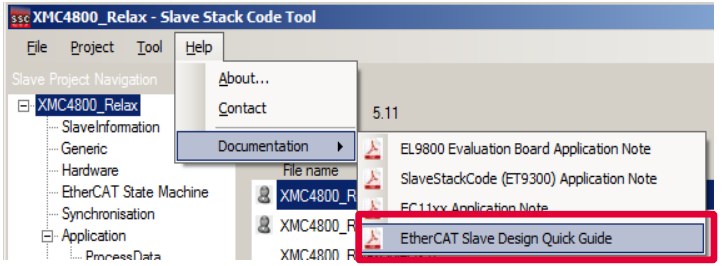
Defining the interface of EtherCAT slave node



2

	Index	ObjectCode	SI	DataType	Name	Default
35	//0x6nnx	Input Data of the Module (0x6000 - 0x6FFF)				
36	0x6000	RECORD			Button	
37			0x01	BOOL	Button1	0
38			0x02	BOOL	Button2	0
39			0x03	pad_14		
40						
41						
42						
43	//0x7nnx	Output Data of the Module (0x7000 - 0x7FFF)				
44	0x7000	RECORD			LED	
45			0x01	BOOL	LED1	0
46			0x02	BOOL	LED2	0
47			0x03	BOOL	LED3	0
48			0x04	BOOL	LED4	0
49			0x05	BOOL	LED5	0
50			0x06	BOOL	LED6	0
51			0x07	BOOL	LED7	0
52			0x08	BOOL	LED8	0
53			0x09	pad_8		

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 tool.

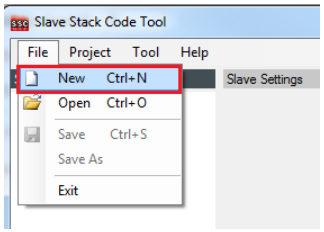


1. Overview and Requirements
2. Setup
3. Defining the interface of EtherCAT slave node
4. Generating Slave Stack Code and ESI file
5. Implementation of the application
6. How to test

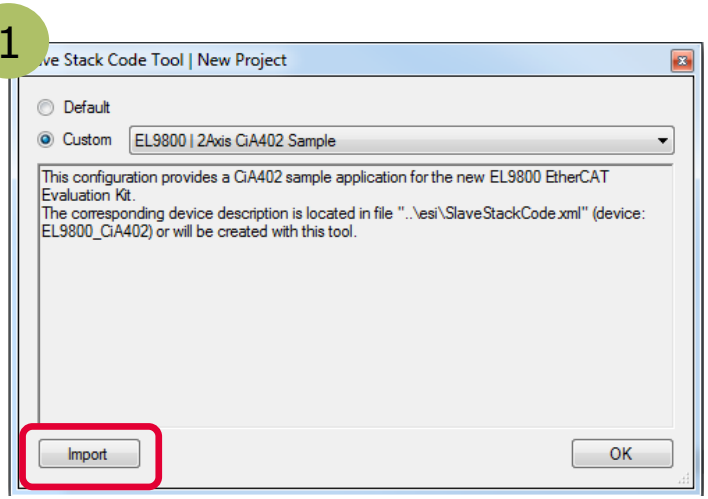
Generating Slave Stack Code and ESI file

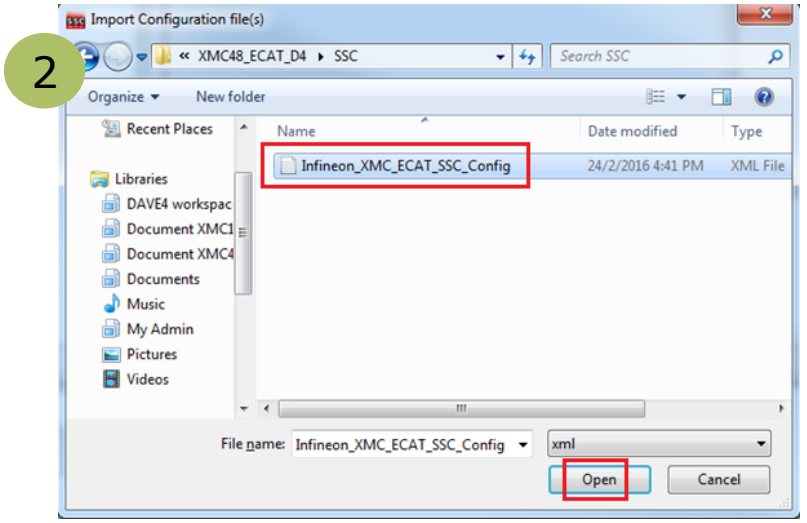
1 Start the SSC tool and this window will pop up automatically.

If not create a new project, by clicking on **File** >> **New**



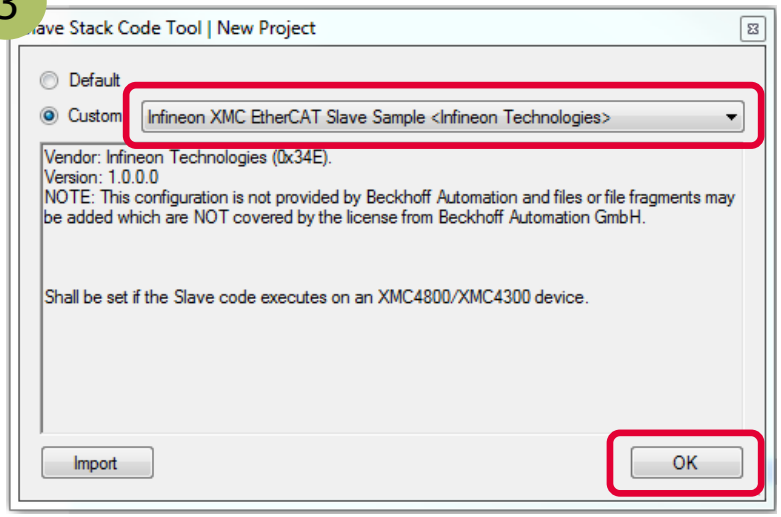
2 Select the slave stack code configuration file for XMC4800 which you find inside the SSC folder.





Generating Slave Stack Code and ESI file

3

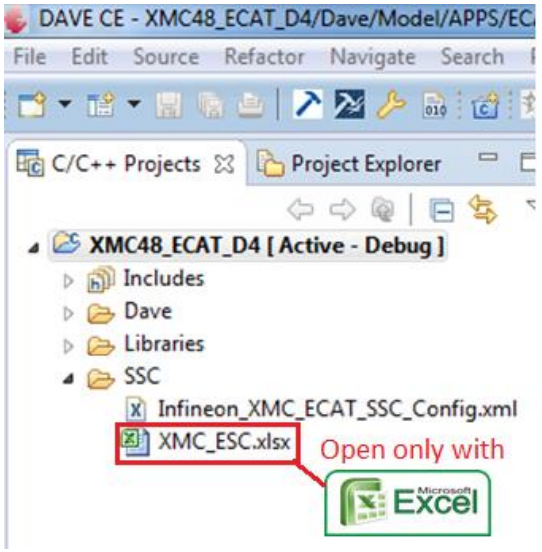


3 Select the Infineon device inside the drop down list and confirm with OK button.
Your project will be created.

Generating Slave Stack Code and ESI file

- Next we are about to do the variable mapping.
- Open the file "**XMC_ESC.xlsx**" in the SSC folder using the **Microsoft Excel**.

Important !!!!
To modify the file "**XMC_ECAT.xlsx**", use only MS Excel and do not use the DAVE4 eclipse viewer !!!

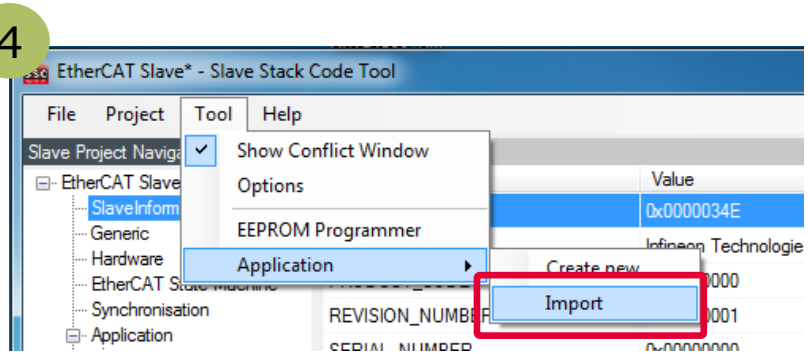


- This excel file already contains both the **Inputs** (2 push buttons) and **outputs** (8 LEDs) variables of the XMC4800 Relax Kits.
- For this training no modification is required, therefore "**save**" and "**close**" the excel worksheet.

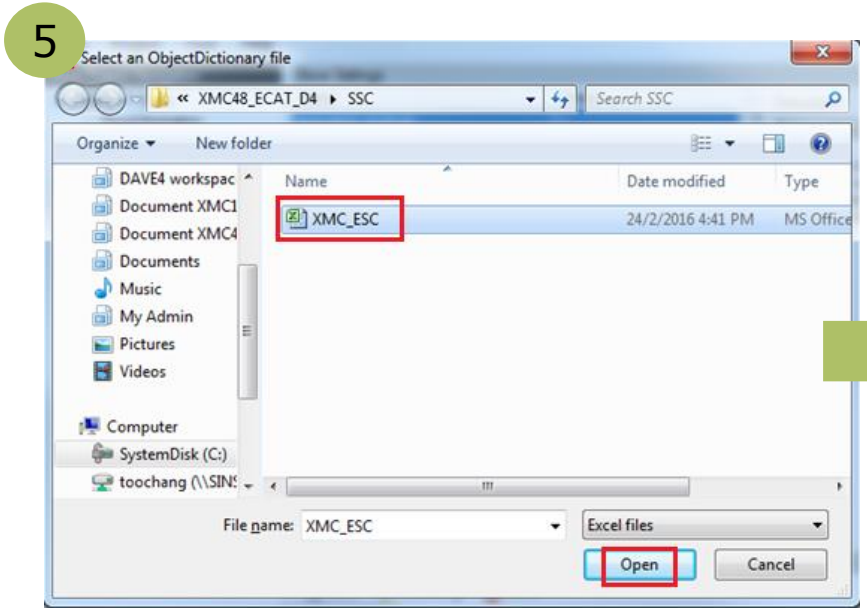
10	Index	ObjectCode	SI	DataType	Name	Default	Min	Max	M/O/C	B/S	Acce
35	//0x6nnx	Input Data of the Module (0x6000 - 0x6FFF)									
36	0x6000	RECORD			Button						
37			0x01	BOOL	Button1	0					ro
38			0x02	BOOL	Button2	0					ro
39			0x03	pad_14							
40											
41											
42											
43	//0x7nnx	Output Data of the Module (0x7000 - 0x7FFF)									
44	0x7000	RECORD			LED						
45			0x01	BOOL	LED1	0					rw
46			0x02	BOOL	LED2	0					rw
47			0x03	BOOL	LED3	0					rw
48			0x04	BOOL	LED4	0					rw
49			0x05	BOOL	LED5	0					rw
50			0x06	BOOL	LED6	0					rw
51			0x07	BOOL	LED7	0					rw
52			0x08	BOOL	LED8	0					
53			0x09	pad_8							

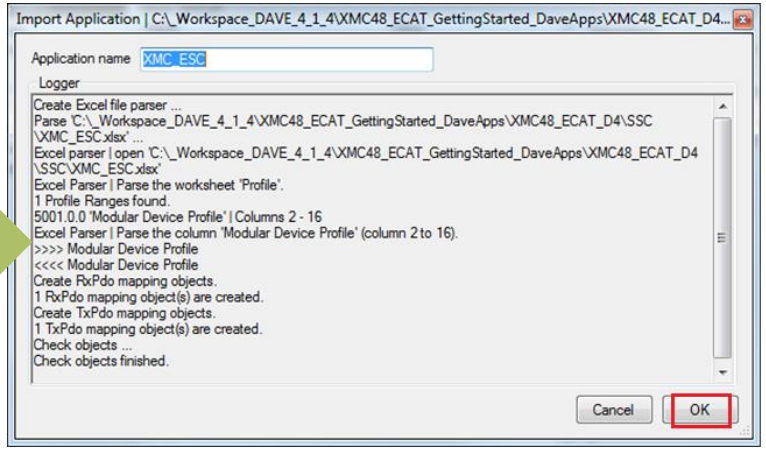
Generating Slave Stack Code and ESI file

4 Import the EXCEL-sheet which defines the interface of your EtherCAT node.



5 Select the EXCEL-file provided inside the SSC folder.



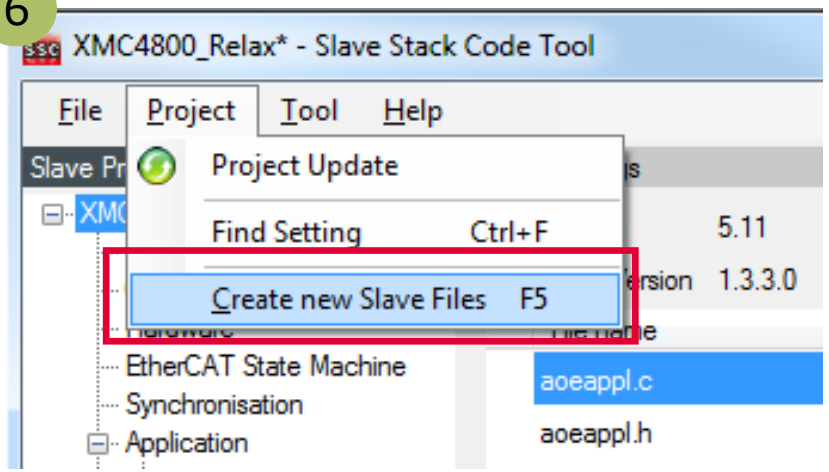


Generating Slave Stack Code and ESI file

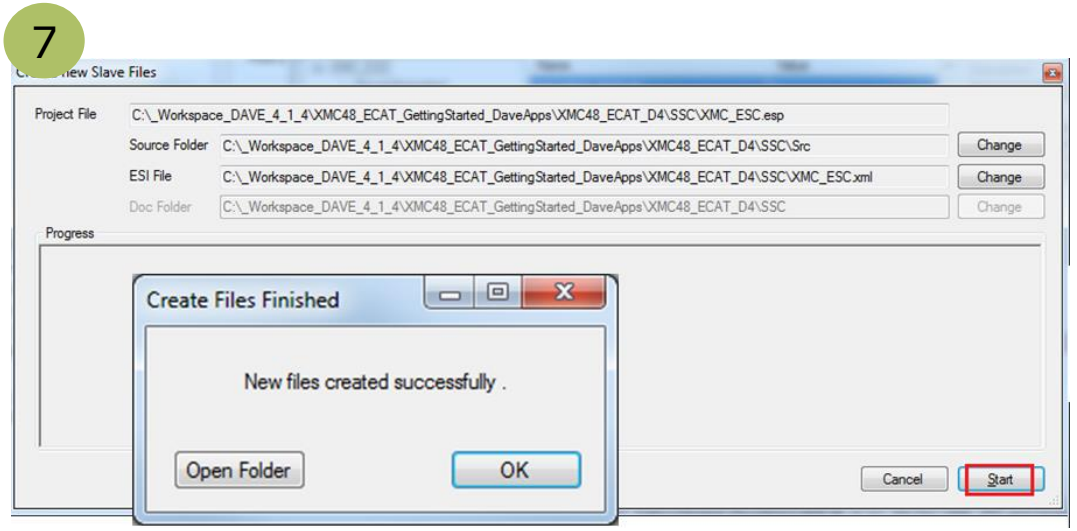
6 Click on **Project** >> **Create new Slave Files** to start file generation.

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.

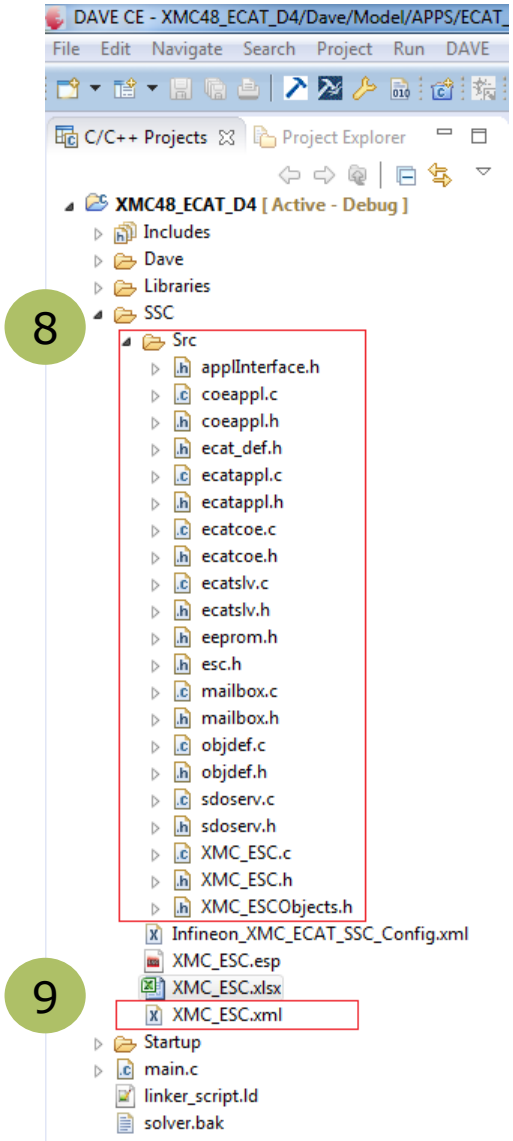
6



7



Generating Slave Stack Code and ESI file



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

Important note !!!!

The ESI file (XMC_ESC.XML) shall be copied into the TwinCAT folder later.

1. 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

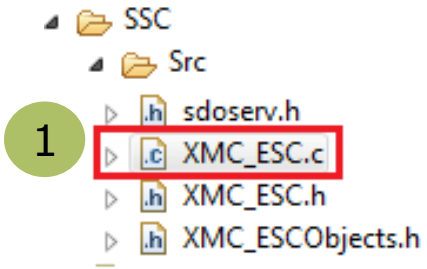
Application coding for XMC_ESC.c

Implement the input and output mapping

1. Open the file "**XMC_ESC.c**" found in the SSC folder

› Replace the codes with the highlighted ones found in both functions:

2. "**APPL_InputMapping**" and
3. "**APPL_OutputMapping**"




```
////////////////////////////////////////////////////////////////////////////////
/**
\param     pData  pointer to input process data

\brief     This function will copies the inputs from the local memory to the ESC memory
           to the hardware
*////////////////////////////////////////////////////////////////////////////////
void APPL_InputMapping(UINT16* pData)
{
   memcpy(pData,&(((UINT16 *)&Button0x6000)[1]),SIZEOF(Button0x6000)-2);
}

////////////////////////////////////////////////////////////////////////////////
/**
\param     pData  pointer to output process data

\brief    This function will copies the outputs from the ESC memory to the local memory
          to the hardware
*////////////////////////////////////////////////////////////////////////////////
void APPL_OutputMapping(UINT16* pData)
{
   memcpy(&(((UINT16 *)&LED0x7000)[1]),pData,SIZEOF(LED0x7000)-2);
}
```

2

3

Application coding for XMC_ESC.c

› Replace the codes with the highlighted ones found in function "**APPL_Application"**

4

```
//////////////////////////////////////////////////////////////////////////////
/**
\brief    This function will called from the synchronisation ISR
          or from the mainloop if no synchronisation is supported
*//////////////////////////////////////////////////////////////////////////////
void APPL_Application(void)
{
// Set LEDs
```

4

```
  if(LED0x7000.LED1)
    PORT1->OMR = 0x40000000;
  else
    PORT1->OMR = 0x00004000;

  if(LED0x7000.LED2)
    PORT0->OMR = 0x40000000;
  else
    PORT0->OMR = 0x00004000;

  if(LED0x7000.LED3)
    PORT3->OMR = 0x40000000;
  else
    PORT3->OMR = 0x00004000;

  if(LED0x7000.LED4)
    PORT0->OMR = 0x80000000;
  else
    PORT0->OMR = 0x00008000;

  if(LED0x7000.LED5)
    PORT1->OMR = 0x00040000;
  else
    PORT1->OMR = 0x00000004;
```

Continue next page...

Application coding for XMC_ESC.c

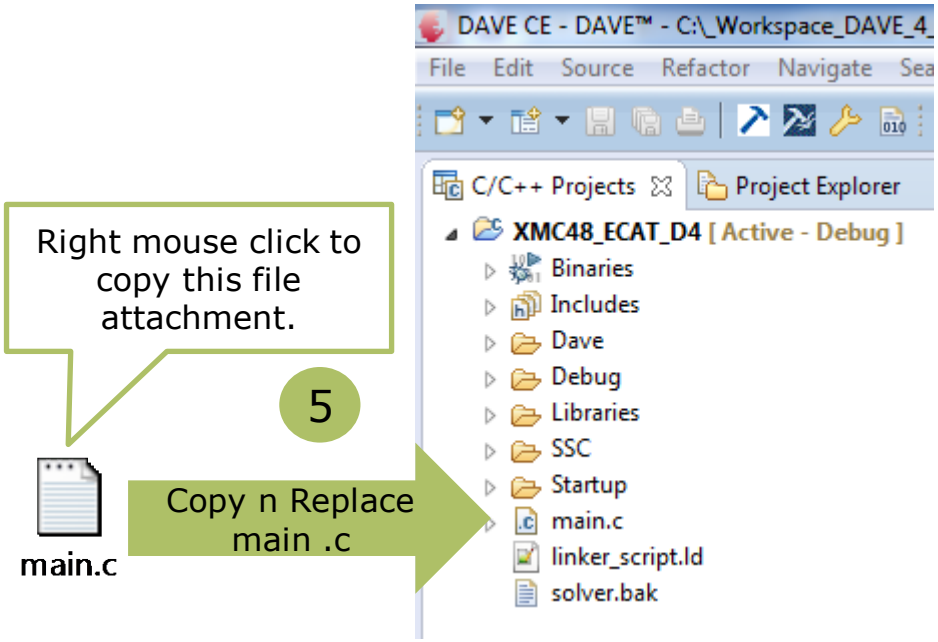
Continue from previous …….

4

```
if(LED0x7000.LED6)
      PORT3->OMR = 0x20000000;
   else
      PORT3->OMR = 0x00002000;

   if(LED0x7000.LED7)
      PORT5->OMR = 0x00080000;
   else
      PORT5->OMR = 0x00000008;

   if(LED0x7000.LED8)
      PORT3->OMR = 0x08000000;
   else
      PORT3->OMR = 0x00000800;

   // Read Button status
   if(PORT15->IN & PORT15_OUT_P13_Msk)
      Button0x6000.Button1 = TRUE;
   else
      Button0x6000.Button1 = FALSE;

   if(PORT15->IN & PORT15_OUT_P12_Msk)
      Button0x6000.Button2 = TRUE;
   else
      Button0x6000.Button2 = FALSE;
}
```

Application coding for main.c

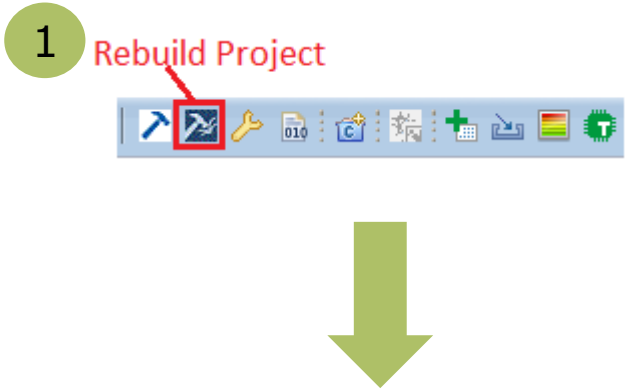
5 Copy and replace the main.c file.

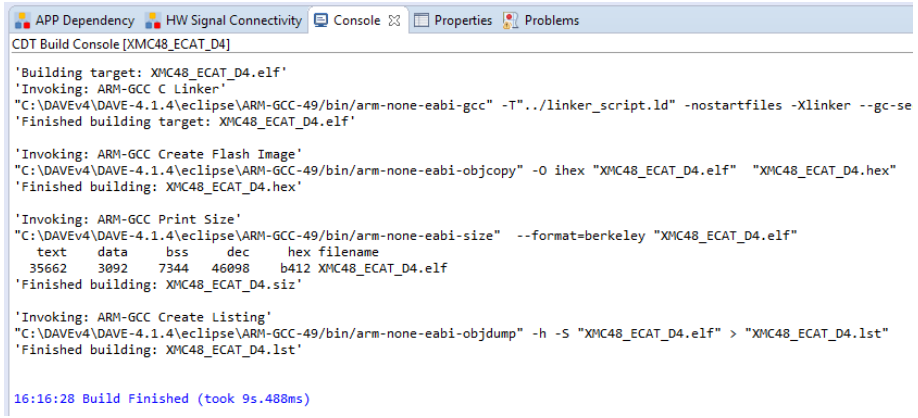


Compile DAVE project

1. Click on the rebuild button to compile the software.

› Ensure that there is no compilation errors.

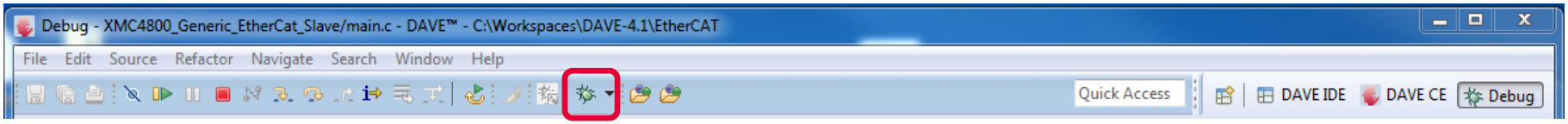




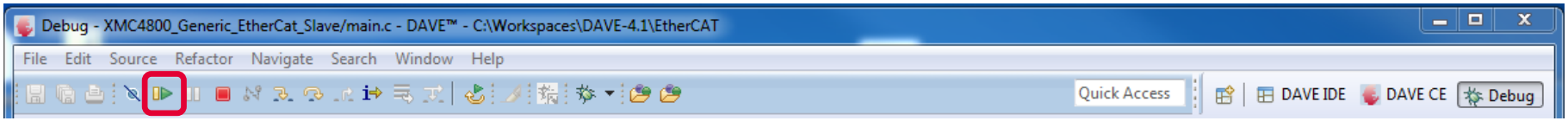
Debug and Run the software

ACTIONS

1. Download software to XMC4800 and start debugger



2. Start the software to run



OBSERVATIONS

1. The ERR-LED on the "XMC EtherCAT PHY Board" will turn on and immediately turn off again.
2. The LED2 on the "XMC4800 Relax EtherCAT Kit" will remain turned on.

1. 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

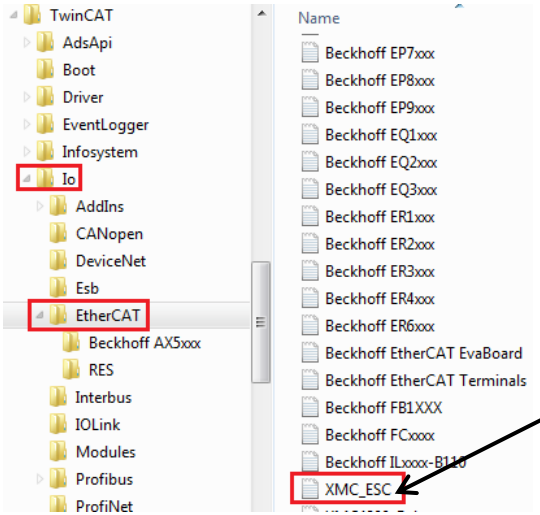
How to test
Start the TwinCAT2 master to run

- Next is to make TwinCAT (Master) know your new EtherCAT Slave Device.

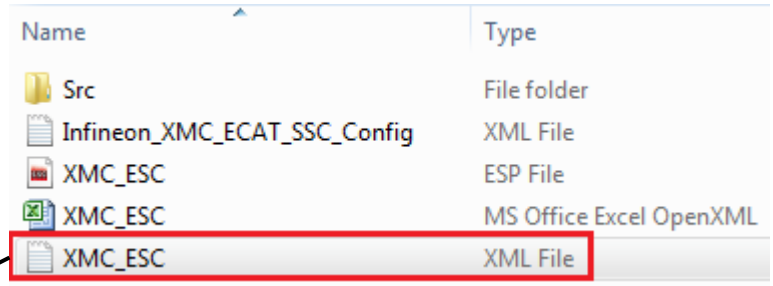
1. Copy the ESI file (**XMC_ESC.xml**) generated by the Slave Stack Tool into the TwinCAT 2 IO directory

C:\TwinCAT\Io\EtherCAT

TwinCAT 2 folder

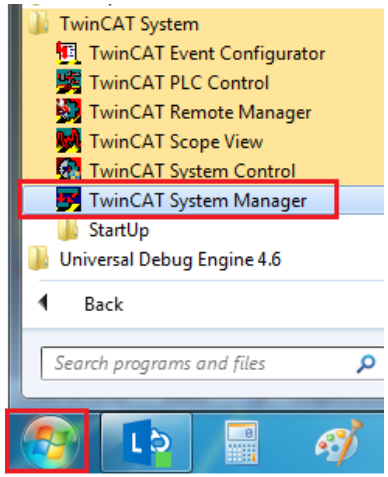


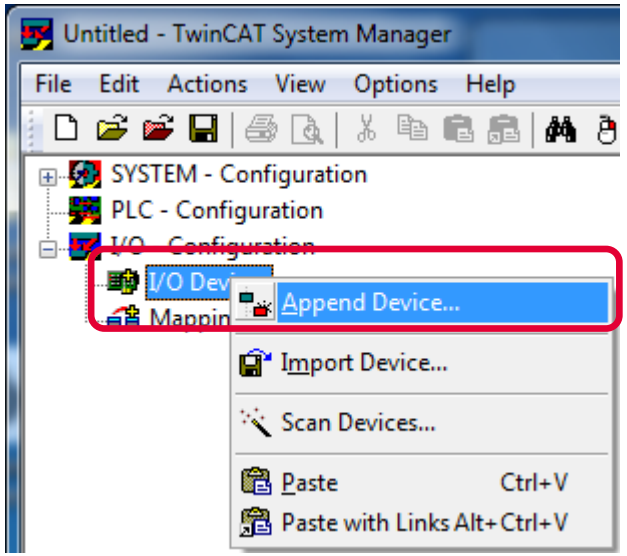
DAVE4 Project folder



1

How to test
Start the TwinCAT2 master to run



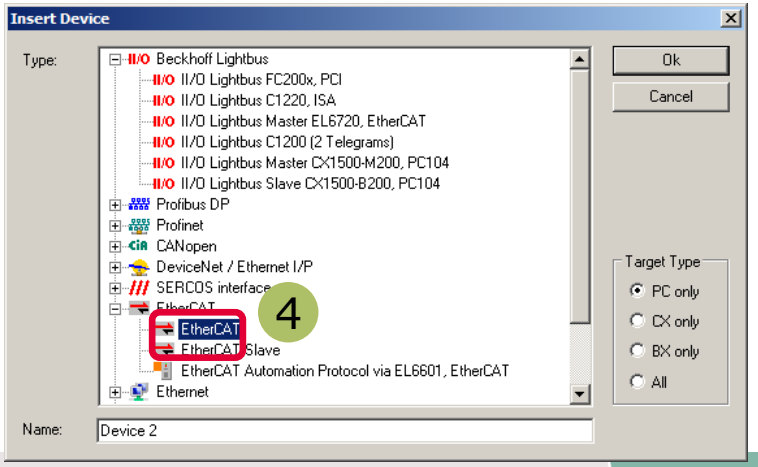


ACTIONS

2 Start the TwinCAT System Manager from windows start menu:

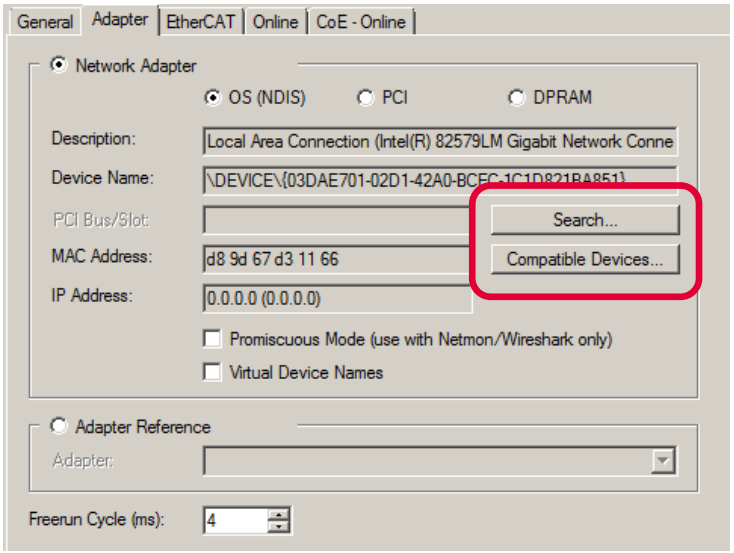
3 Right Click I/O-Devices and select „Append Device…“

4 Create an EtherCAT master device by double click

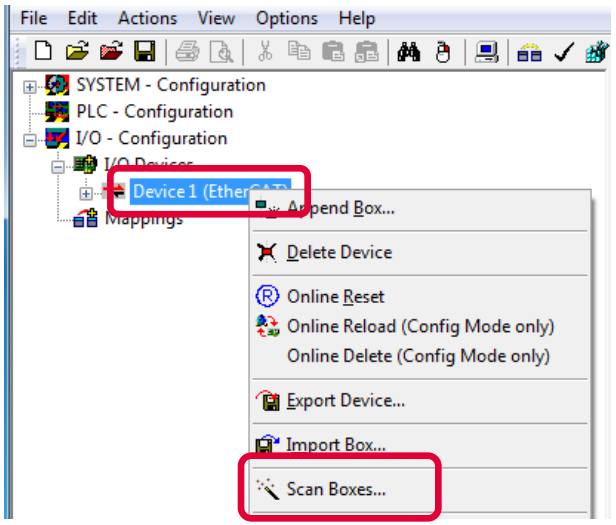


How to test
Start the TwinCAT2 master to run

5



6



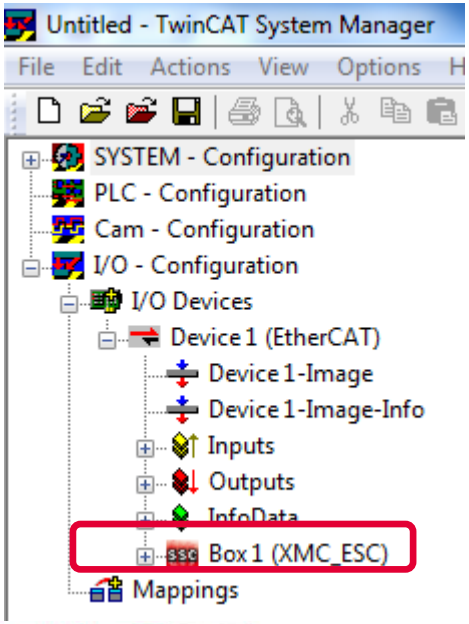
ACTIONS

5 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.

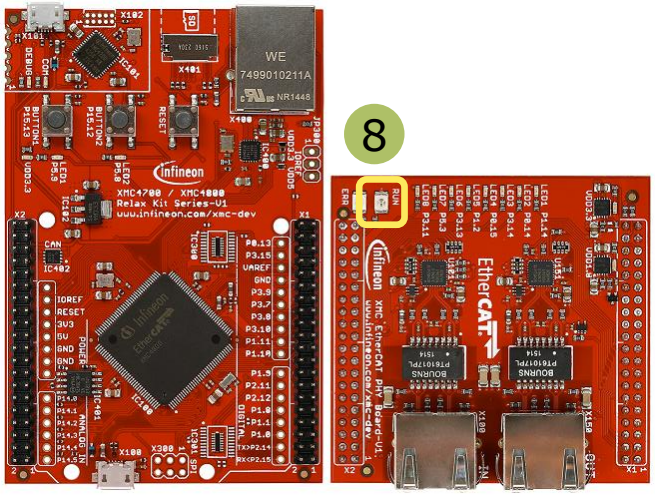
6 Right Click EtherCAT master and select „Scan Boxes…“

How to test
Start the TwinCAT2 master to run

7



8

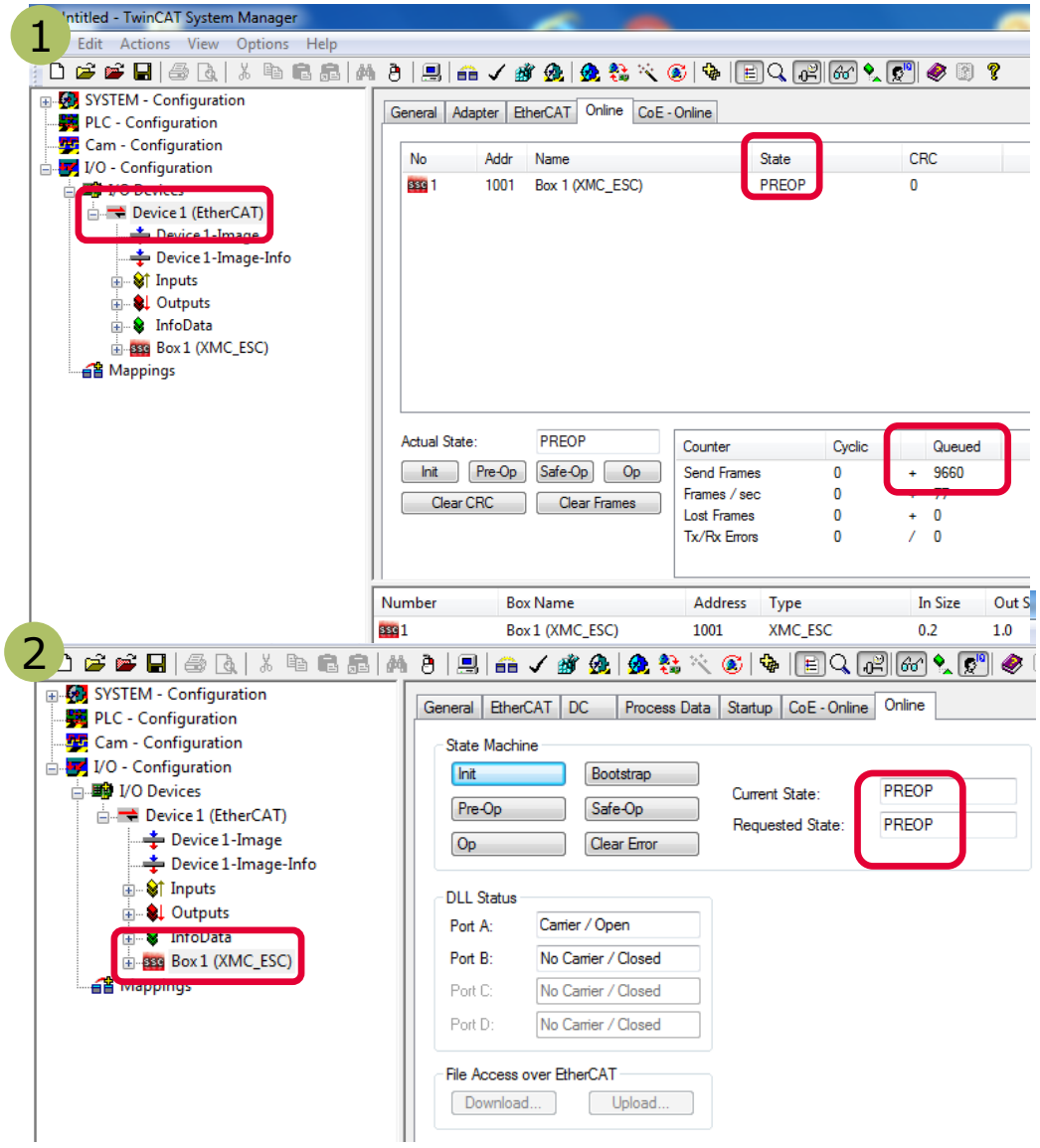


OBSERVATIONS

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

8 The RUN-LED is flashing indicating **PREOP**-state

How to test
Start the TwinCAT2 master to run



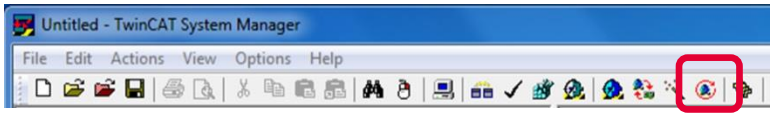
OBSERVATIONS

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

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

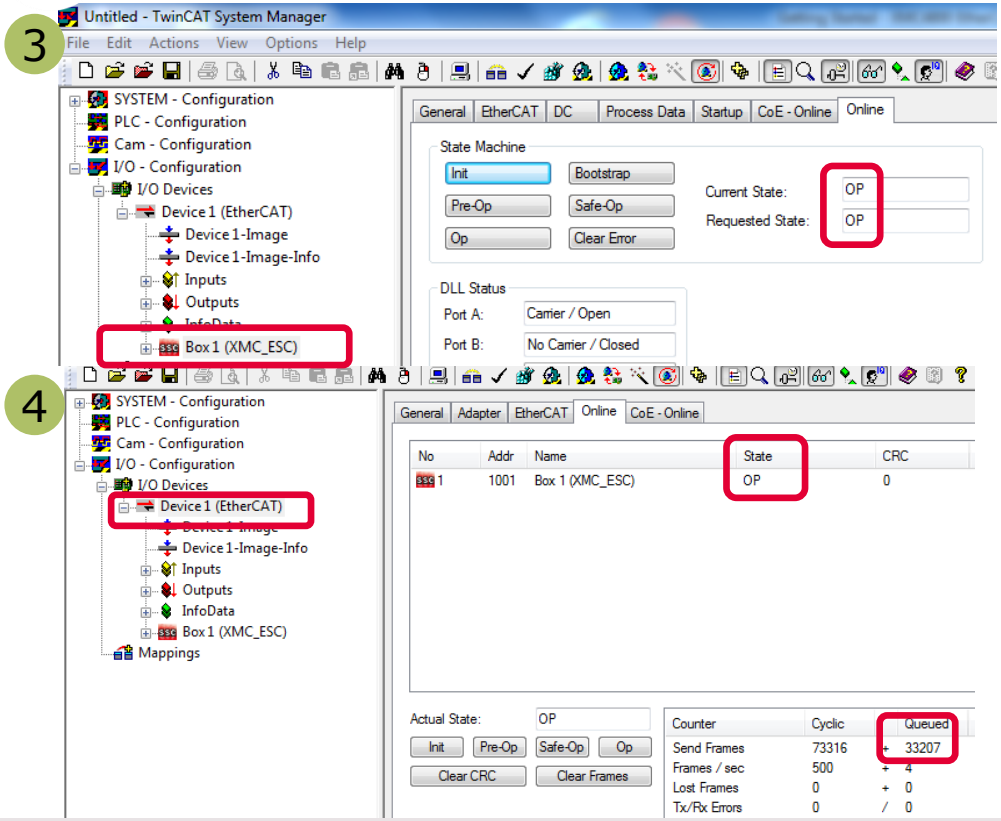
How to test
Start the TwinCAT2 master to run

ACTION



Set master device to **free run** mode

OBSERVATIONS



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

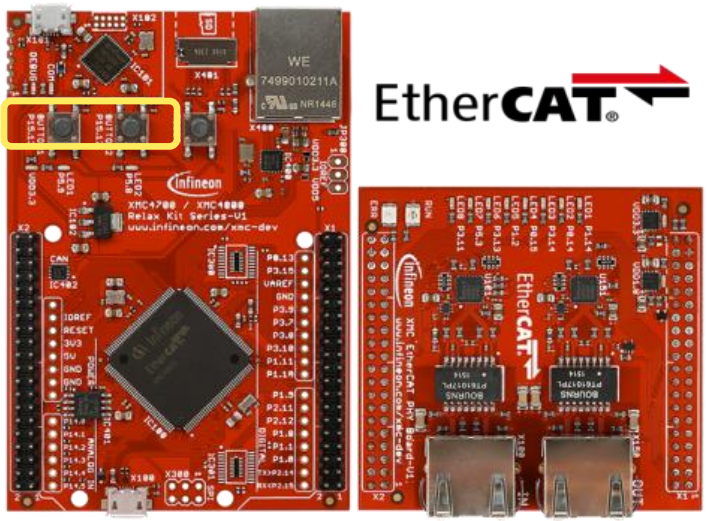
4 EtherCAT master view: Online status of master shows the slave in OP state. Frames are no more queued. Cyclic counter is incrementing.

5 „XMC EtherCAT PHY Board“: RUN-LED is static turned on indicating OP-state.

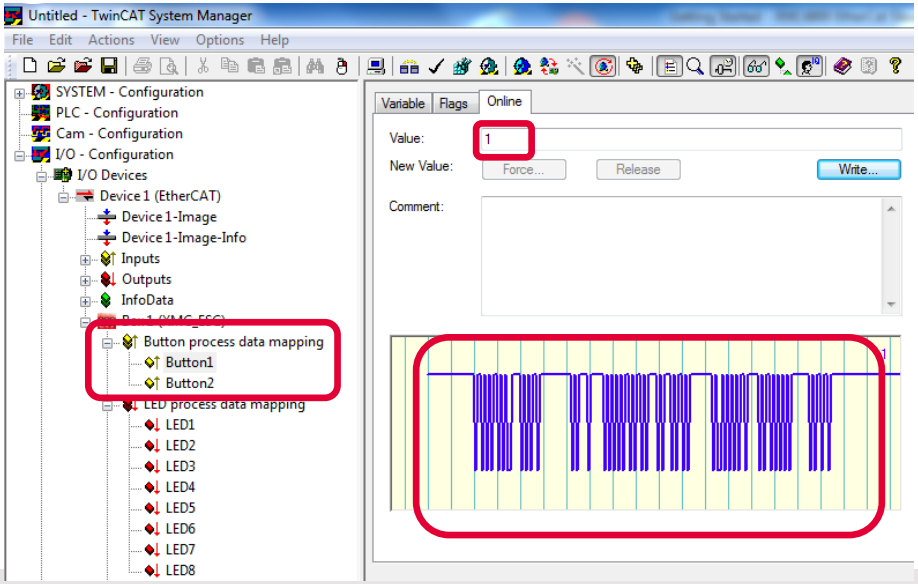
How to test
Monitoring slave inputs on master

ACTIONS

While pushing Button1 or 2 on „XMC4800 Relax EtherCAT Kit“ button state is updated on host.



OBSERVATIONS

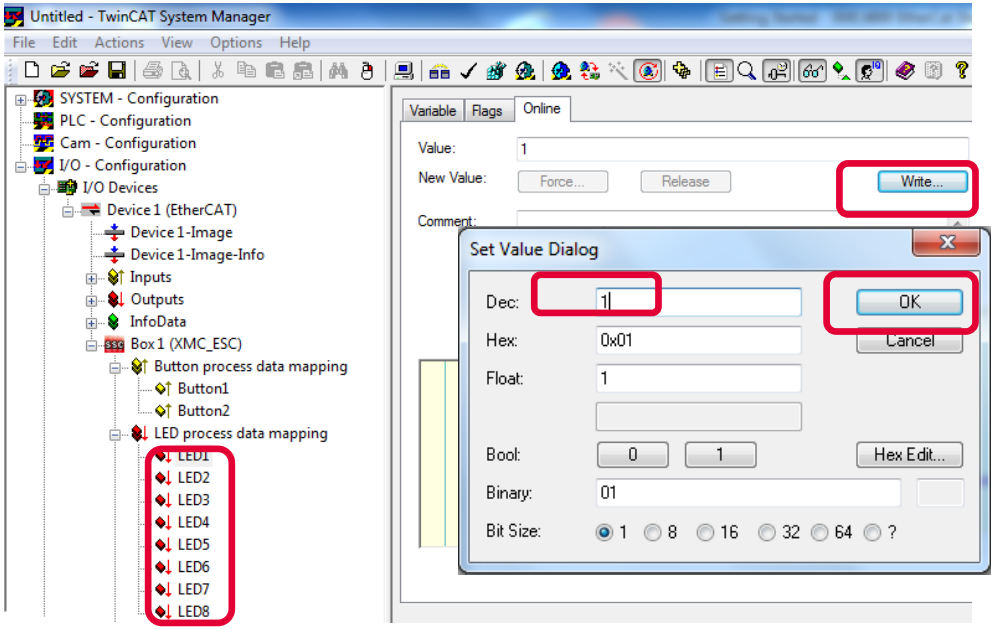


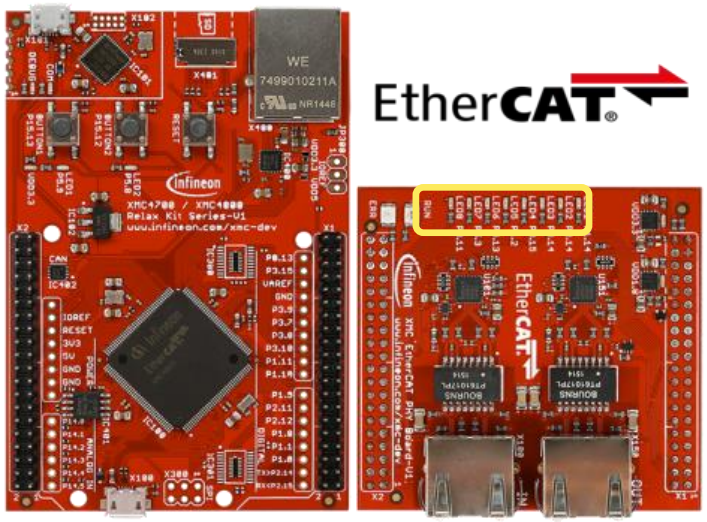
State of Button1/Button2 changes according state of BUTTON1.

How to test
Setting slave outputs on master

ACTIONS

Right click on LED1 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 „XMC EtherCAT PHY Board" is turned on/off according to OUT_GEN_Bit1 setting.

Part of your life. Part of tomorrow.

Infineon