SOES EtherCAT Slave Example

Getting started with EtherCAT and Infineon XMC4

rt:labs

Workflow

1. Overview and Requirements
2. Setup hardware, DAVE and EtherCAT SDK
3. Define the EtherCAT application with EtherCAT SDK
4. Implement the application in DAVE
5. Test the application with EtherCAT SDK

rt:labs

Workflow

1. Overview and Requirements
2. Setup hardware, DAVE and EtherCAT SDK
3. Define the EtherCAT application with EtherCAT SDK
4. Implement the application in DAVE
5. Test the application with EtherCAT SDK

rt:labs

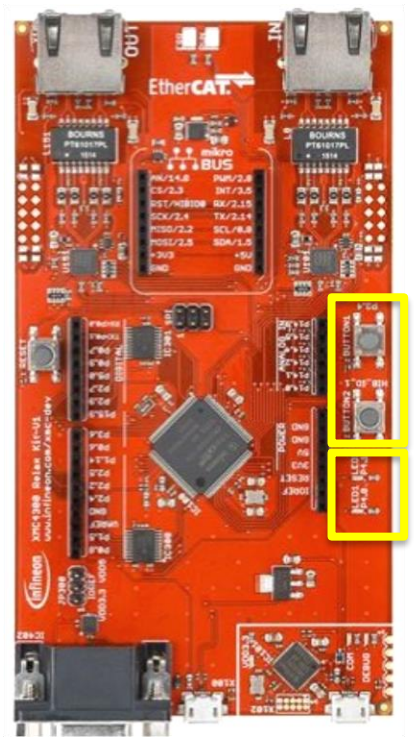
Overview

This example demonstrate how to implement an XMC4300/XMC4800 Relax Kit based EtherCAT slave using rt-labs EtherCAT SDK integrated in DAVE and open source EtherCAT slave stack SOES (Simple Open EtherCAT slave).

While following this getting started guide you will get an introduction to EtherCAT and its components without having any previous experience with EtherCAT.

The goal is to create a simple free-running Input / Output EtherCAT slave that is conformant to the EtherCAT specification and will pass the EtherCAT Conformance Test.

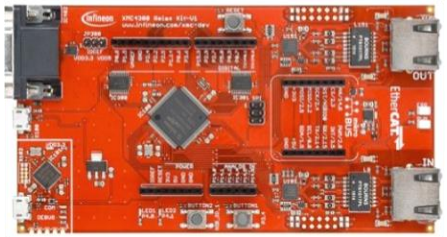
The application will make use of the XMC4300/XMC4800 relax kits on board BUTTONS and LEDS, the documentation present pictures, numbers and names for XMC4300 Relax Kit but it is easy to adopt to fit XMC4800 Relax Kits, simply point out correct GPIOs for the XMC4800.



rt:labs

Requirements

- XMC4 Relax EtherCAT Kit



- RJ45 Ethernet Cable

- Windows Laptop
  - DAVE v4
  - rt-labs EtherCAT SDK
  - Wireshark w/ WinPCAP

- Micro USB Cable (Debugger connector)

rt:labs

Requirements - free downloads

- Install DAVE , get @ Download DAVE
- Install EtherCAT SDK , get@ rt-labs
- EtherCAT slave stack SOES, get @ OpenEtherCATsociety SOES
- Wireshark, get @ Wireshark.org

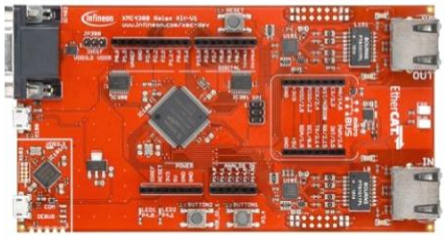
Workflow

1. Overview and Requirements
2. Setup hardware, DAVE and EtherCAT SDK
3. Define the EtherCAT application with EtherCAT SDK
4. Implement the application in DAVE
5. Test the application with EtherCAT SDK

rt:labs

Setup - hardware

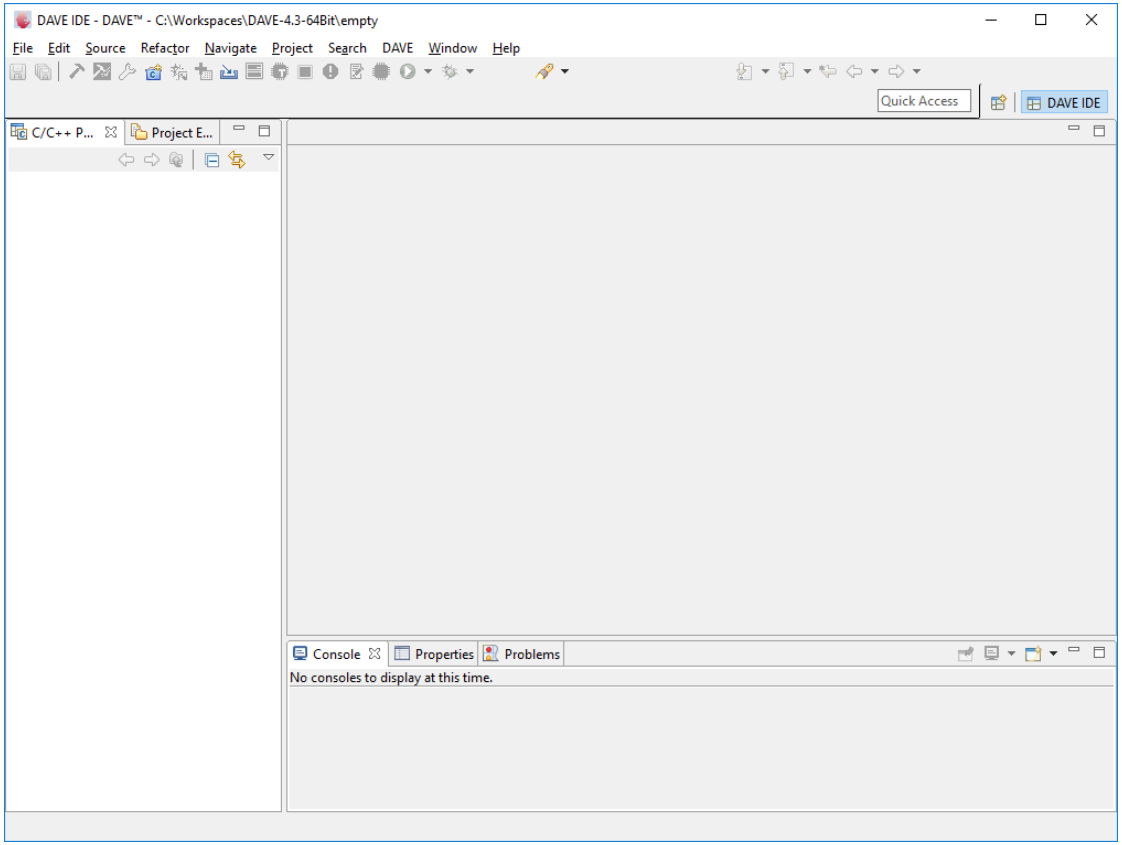
- Micro USB Cable from laptop to debug connector



- Ethernet cable from laptop port to Relax Kit IN-port

rt:labs

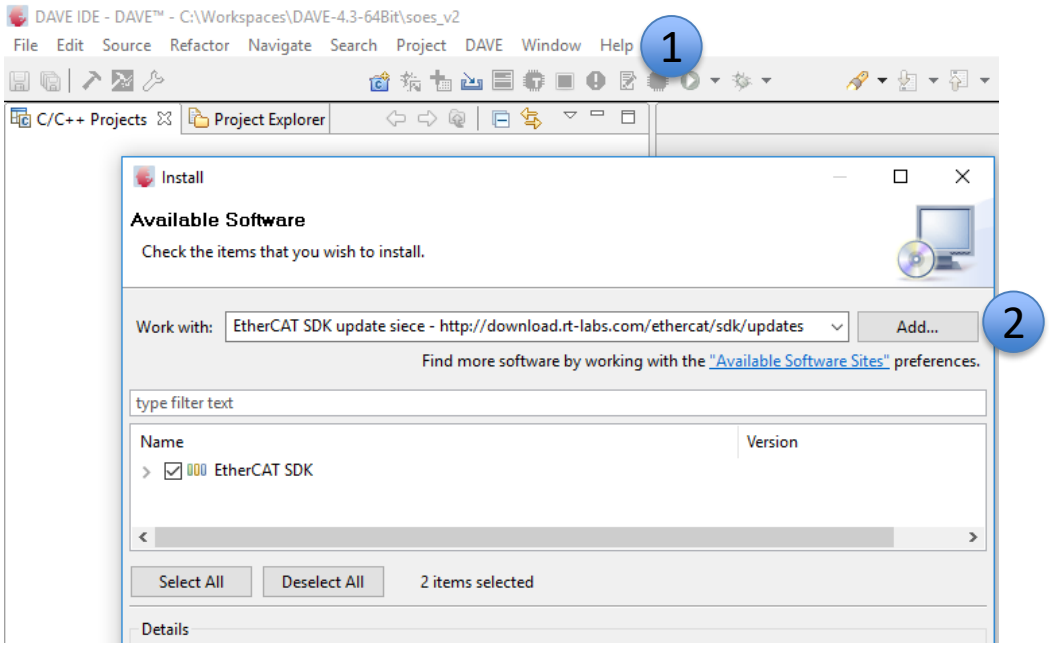
Setup - Start DAVE



rt:labs

Setup- Install EtherCAT SDK

1. Go to menu: Help-> Install New Software
2. Add Add… download site: http://download.rt-labs.com/ethercat/sdk/updates

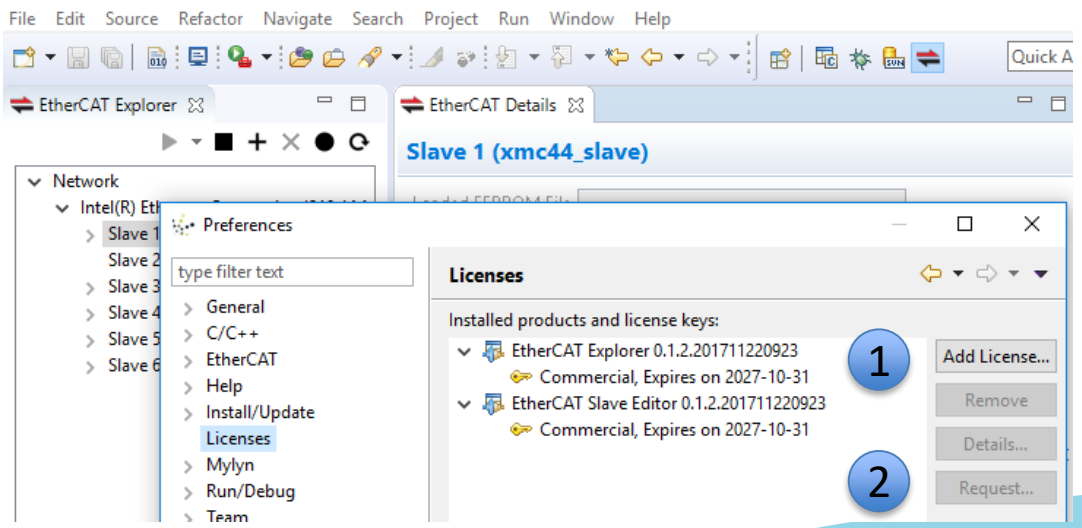


rt:labs

Setup- Add EtherCAT SDK license

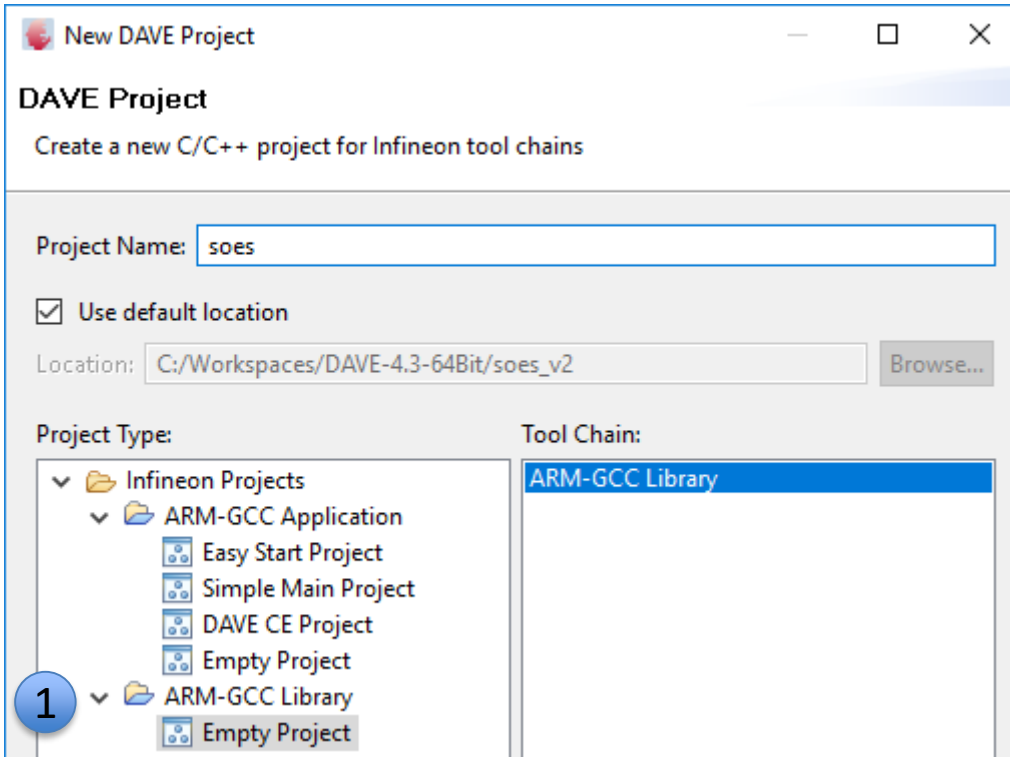
A license is required to be able to use the EtherCAT SDK, to install a license Go to: Windows->Preferences->Licenses. There are two options,

1. Add License..., browse an existing license file
2. Request an evaluation license by providing name and e-mail, shortly an e-mail with a license will be sent to given e-mail. Use option 1 to add the license.



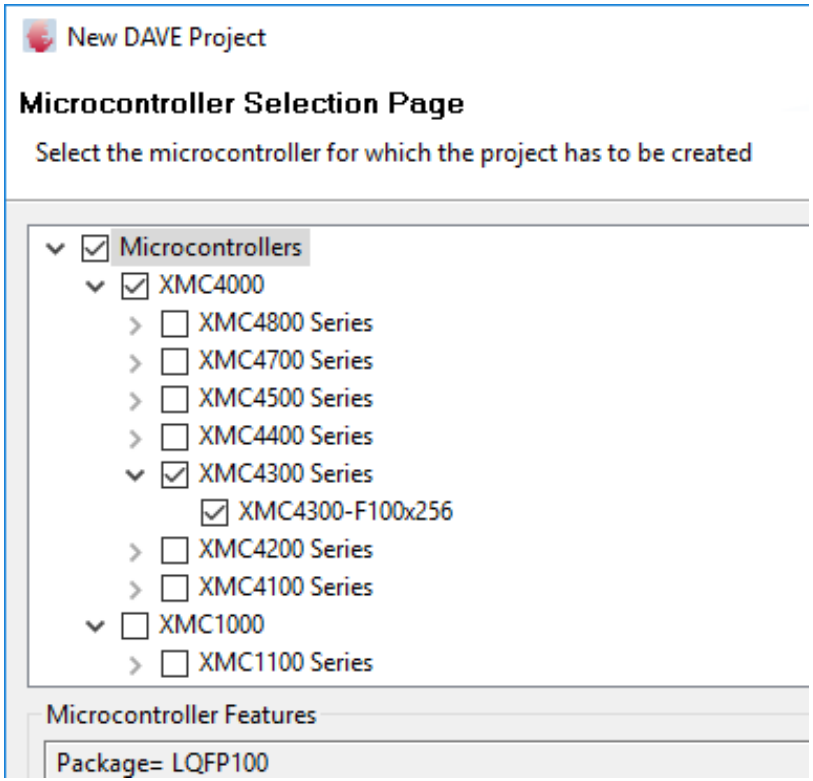
Setup - Create the EtherCAT slave stack library project

1. Go to menu: File->New->DAVE Project-><Project Type of your choice>



rt:labs

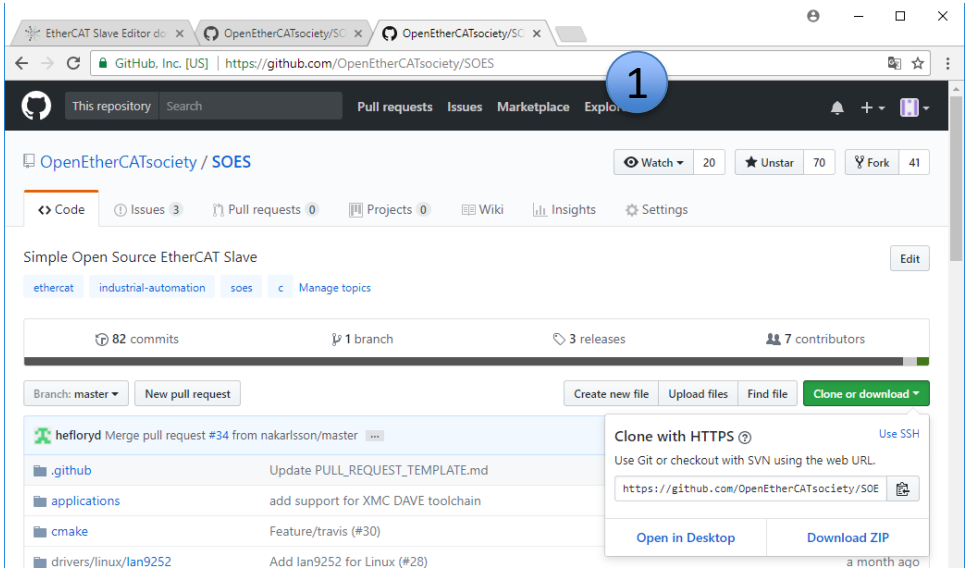
Setup - Choose XMC4300 or XMC4800

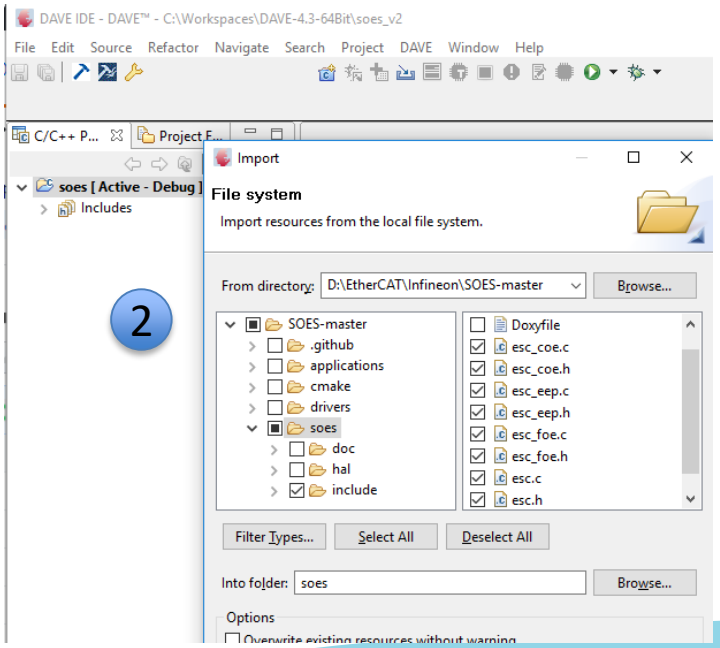


rt:labs

Setup – import files to slave stack project

1. Download or Clone the EtherCAT Slave Stack SOES@ GitHub OpenEtherCATsociety/SOES
2. Import generic parts of the slave stack \soes and soes\include

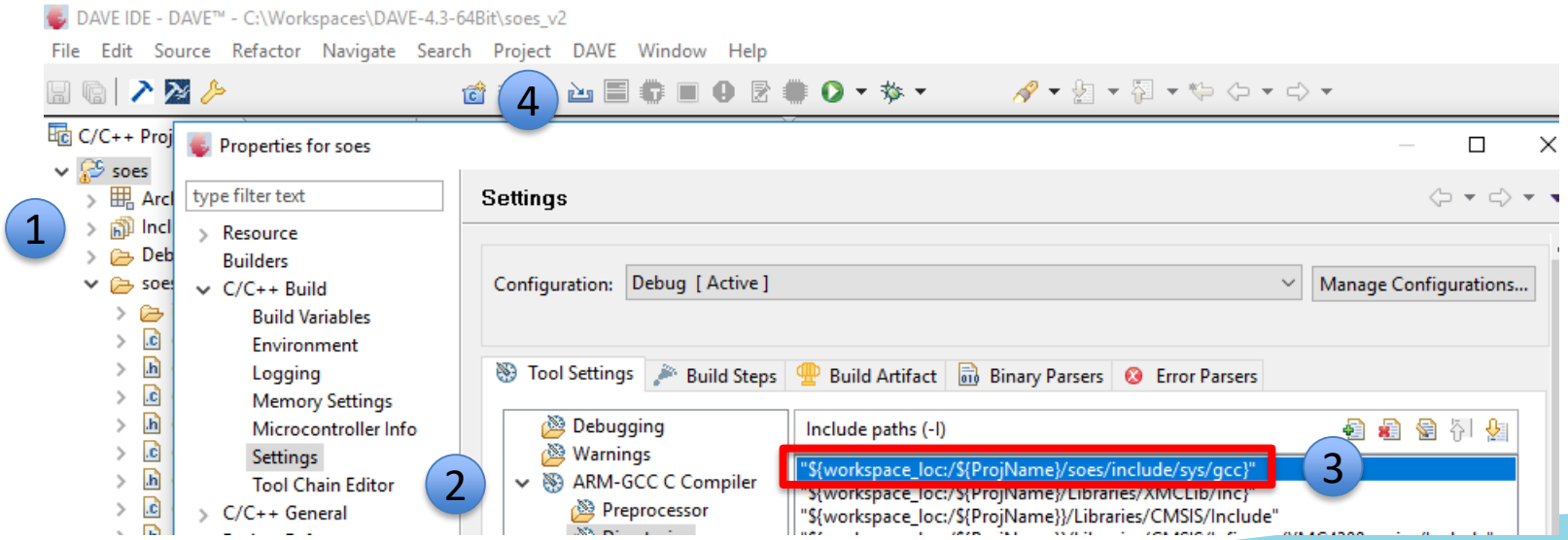




rt:labs

Setup -Add slave stack includes and build

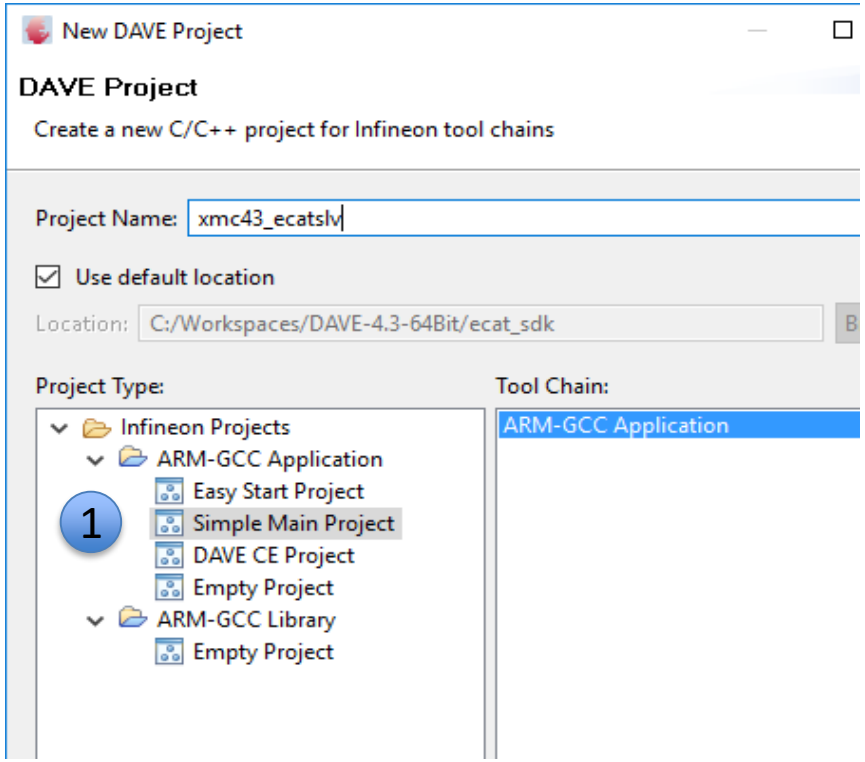
1. Right click the Library <Project> and select Properties
2. Go to: Properties -> C/C++ Build -> Settings -> ARM-GCC C Compiler
3. Add the marked includes
4. Goto Project and Build All



rt:labs

Setup - Create the EtherCAT slave application project

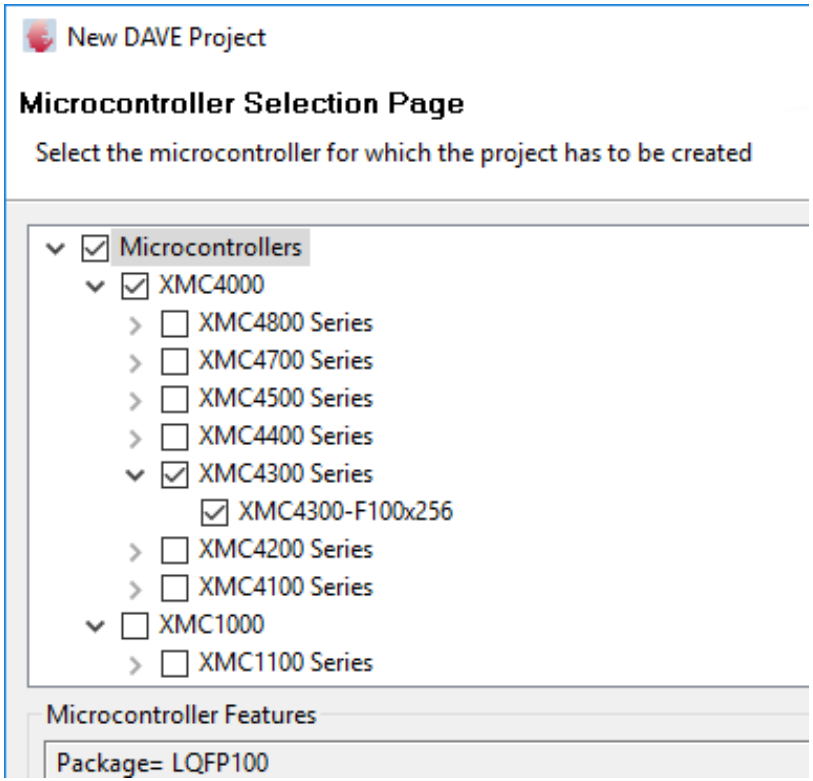
1. Go to menu: File->New->DAVE Project-><Project Type of your choice>



OBS! Using DAVE CE may cause pin conflicts not detected by DAVE since SOES also do some hardware setup

rt:labs

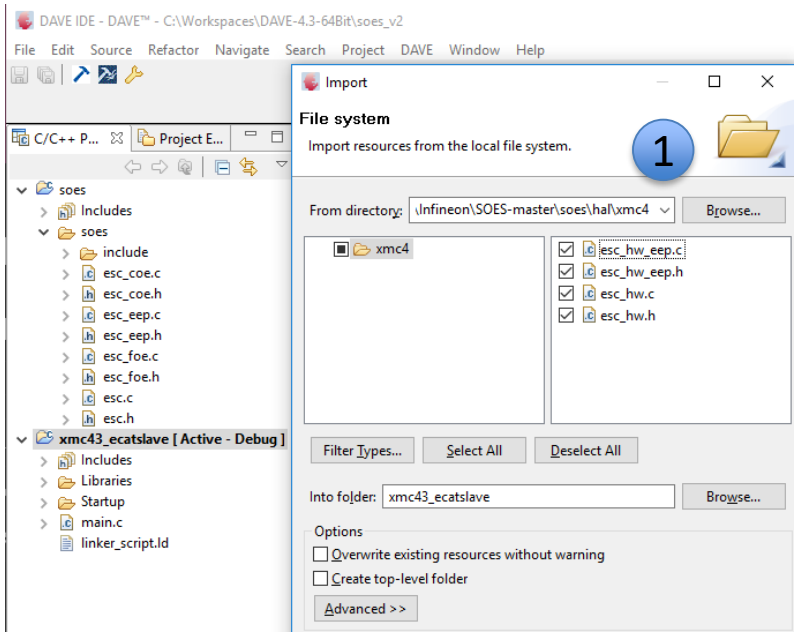
Setup - Choose XMC4300 or XMC4800



rt:labs

Setup – import files to application project

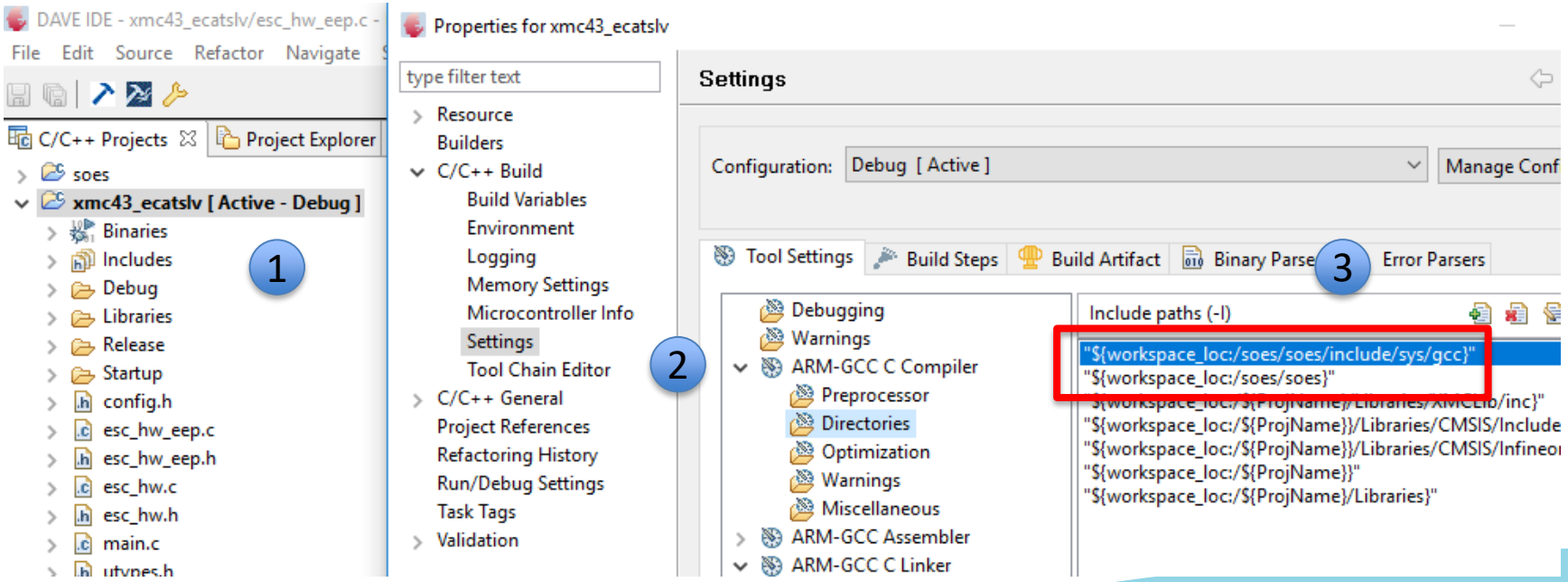
1 Import SOES DAVE XMC hardware abstraction layer parts of the slave stack \soes\hal\xmc4



rt:labs

Setup -Add slave stack includes

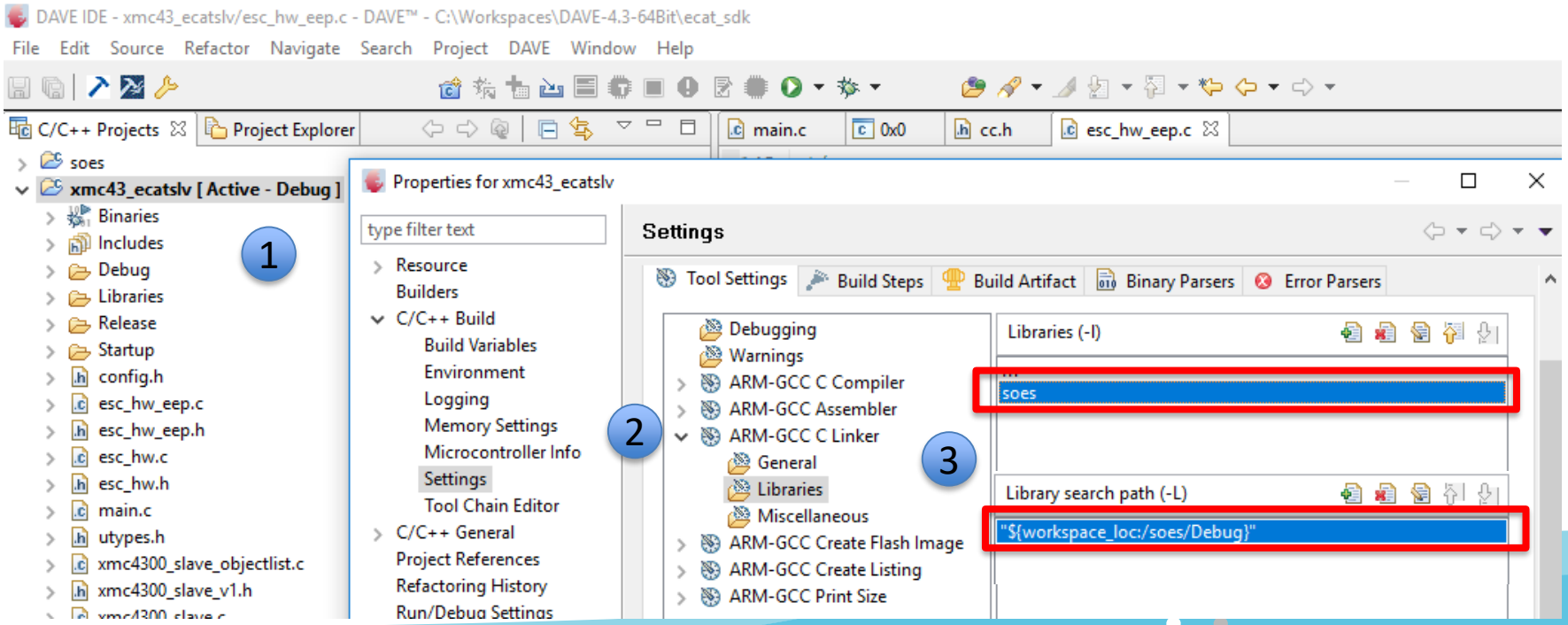
1. Right click the Application <Project> and select Properties
2. Go to: Properties -> C/C++ Build -> Settings -> ARM-GCC C Compiler
3. Add the marked includes



rt:labs

Setup -Add slave stack library

1. Right click the Application <Project> and select Properties
2. Go to: Properties -> C/C++ Build -> Settings -> ARM-GCC C Linker
3. Add the marked library and search path



rt:labs

Workflow

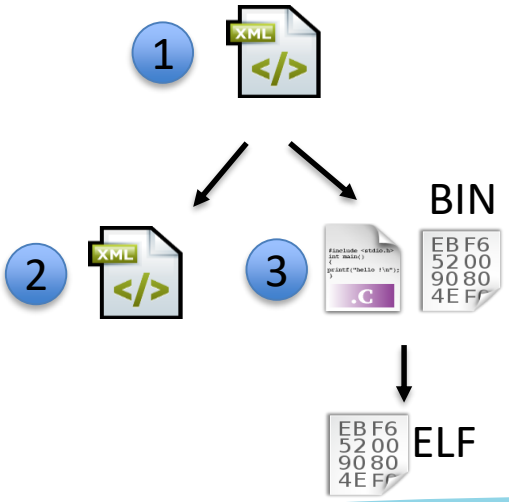
1. Overview and Requirements
2. Setup hardware, DAVE and EtherCAT SDK
3. Define the EtherCAT application with EtherCAT SDK
4. Implement the application in DAVE
5. Test the application with EtherCAT SDK

rt:labs

EtherCAT SDK - Slave editor

In this example we develop an EtherCAT slave with fixed process data and an Object Dictionary. To run the CTT (**C**onformance **T**est **T**ool) we need to provide three objects, EEPROM, OD (**O**bject **D**ictionary) and ESI-file(**E**therCAT **S**lave **I**nformation). The EEPROM and OD are loaded on the target for online configuration, the ESI is used for offline configuration. The information in these three objects overlap and must be coherent to pass the conformance test. The EtherCAT SDK help you generate these objects and keep the information aligned throughout the EtherCAT slave lifetime.

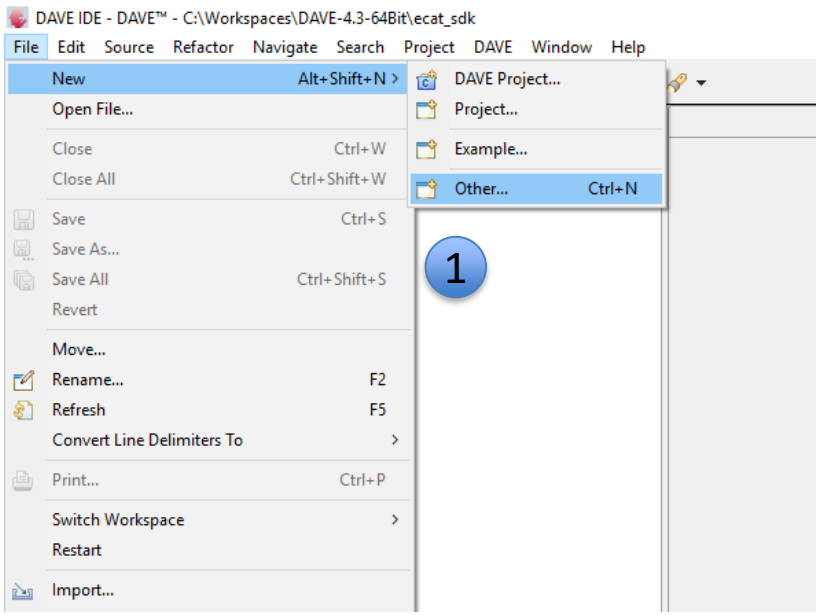
1. The EtherCAT slave editor create and store the information in an esx-file, only used by the Slave Editor
2. The tool generates a conformant ESI-file needed to run CTT
3. The tool generates a conformant OD and EEPROM that will be included in the application project. The OD is a C array/struct compiled with the C application, since we emulate the EEPROM we will convert the EEPROM binary to a linkable object and link it to the resulting application ELF

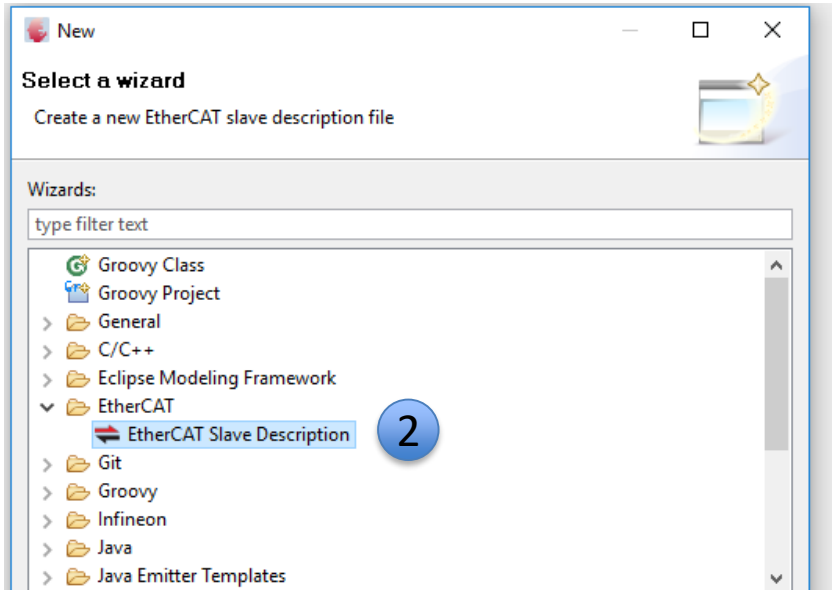


rt:labs

EtherCAT SDK - Slave editor

1. Go to menu: File->Other
2. Select Wizard: EtherCAT->EtherCAT Slave Description

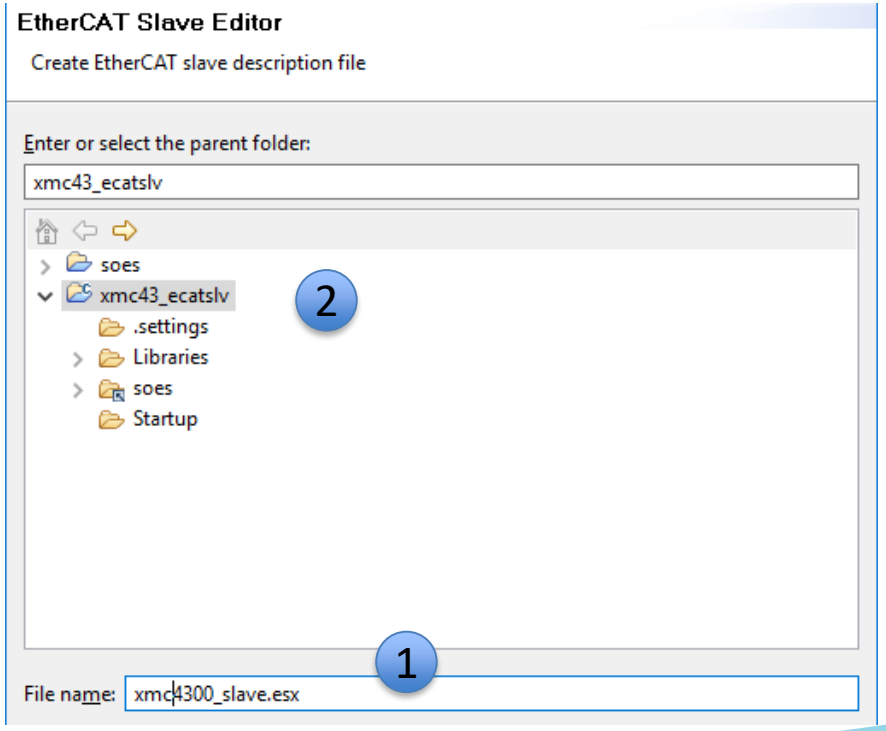




rt:labs

EtherCAT SDK - Slave editor

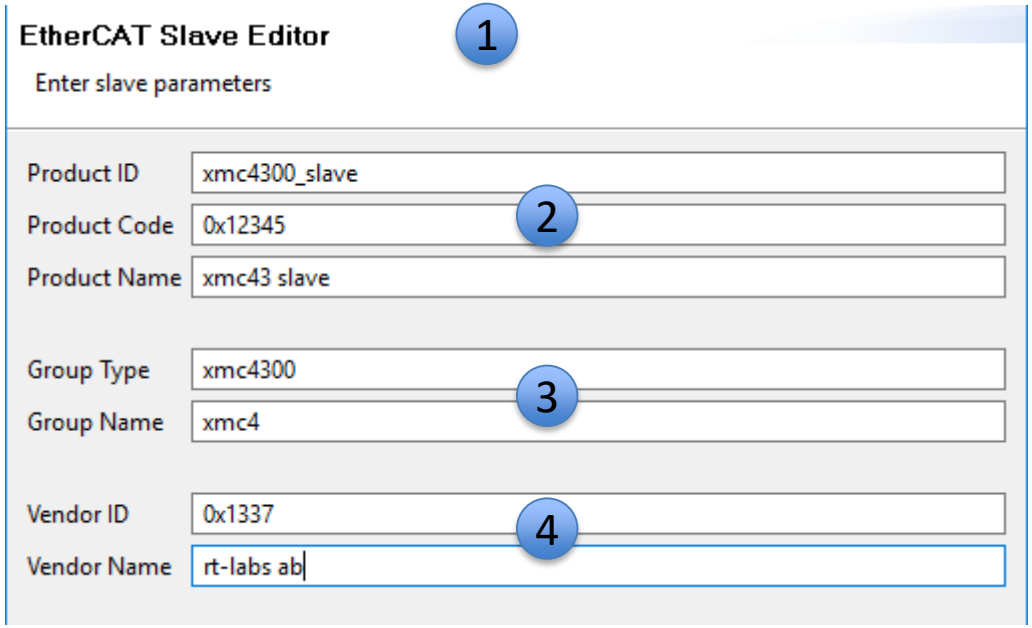
1. Create a slave description project file
2. Point out where to place it relative to the work space



rt:labs

EtherCAT SDK- Slave editor

1. Enter the base parameters
2. Product information, used to uniquely identify the slave device
3. Group information, give you possibility to group slaves
4. Vendor information, vendor ID is provided by at [ETG]

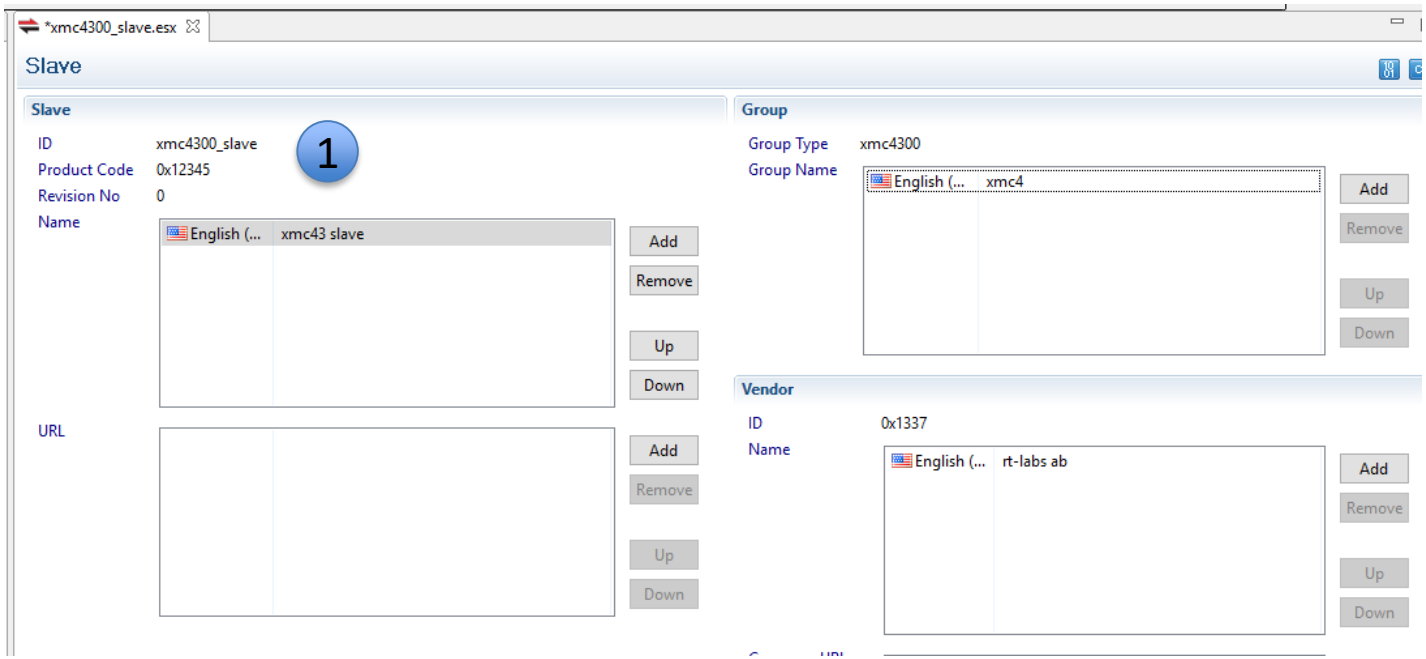


rt:labs

EtherCAT SDK -Slave editor

Slave TAB: Edit and add base parameters.

1 To modify texts, click the label and edit in-place

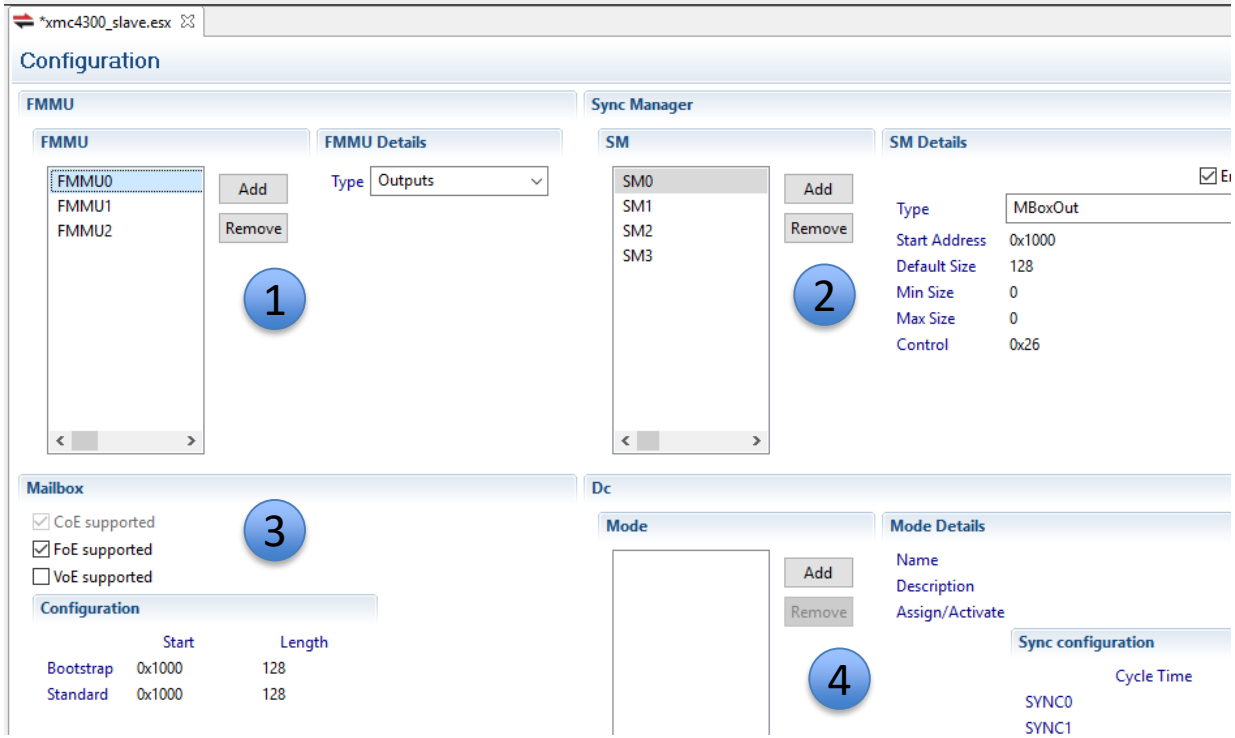


rt:labs

EtherCAT SDK - Slave editor

Configuration TAB: Basic configuration settings

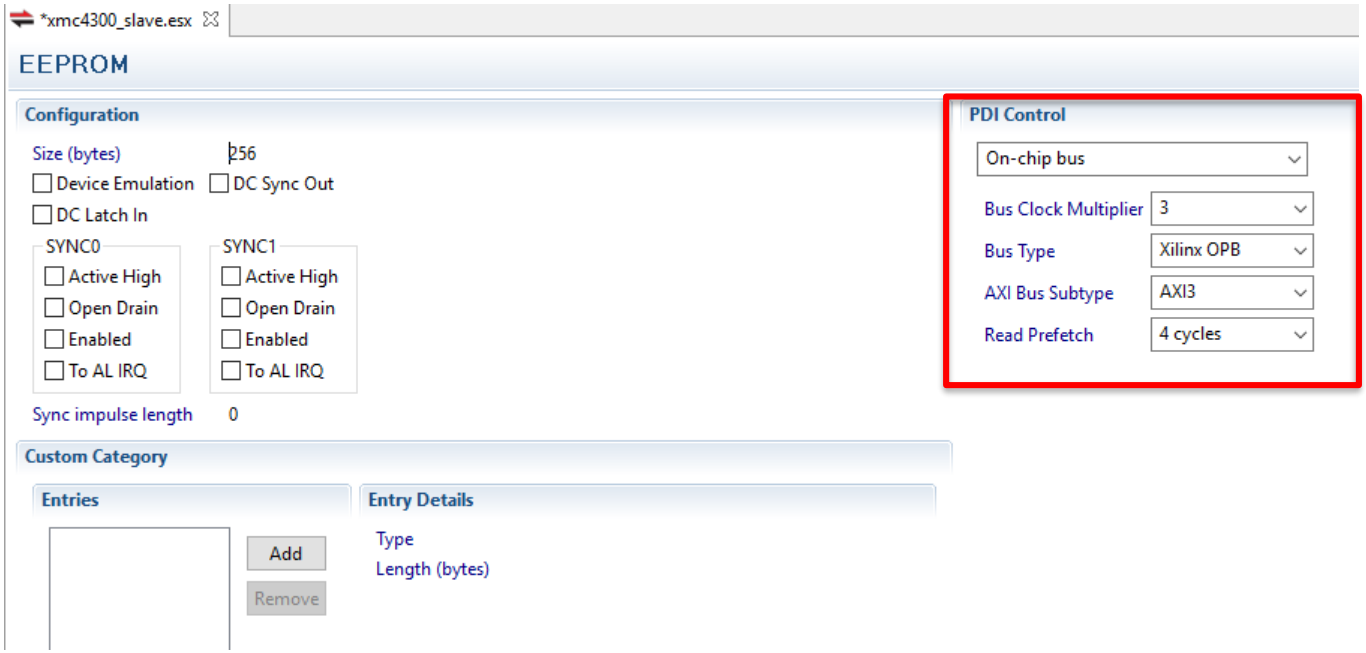
1. FMMU types
2. SM types, address, control, size
3. Supported protocols and configuration
4. DC – leave as is since the example is a free-run slave



rt:labs

EtherCAT SDK - Slave editor

EEPROM TAB: Use this PDI Control settings, they're aligned with ReadOnly ESC values according to the XMC ref manual.

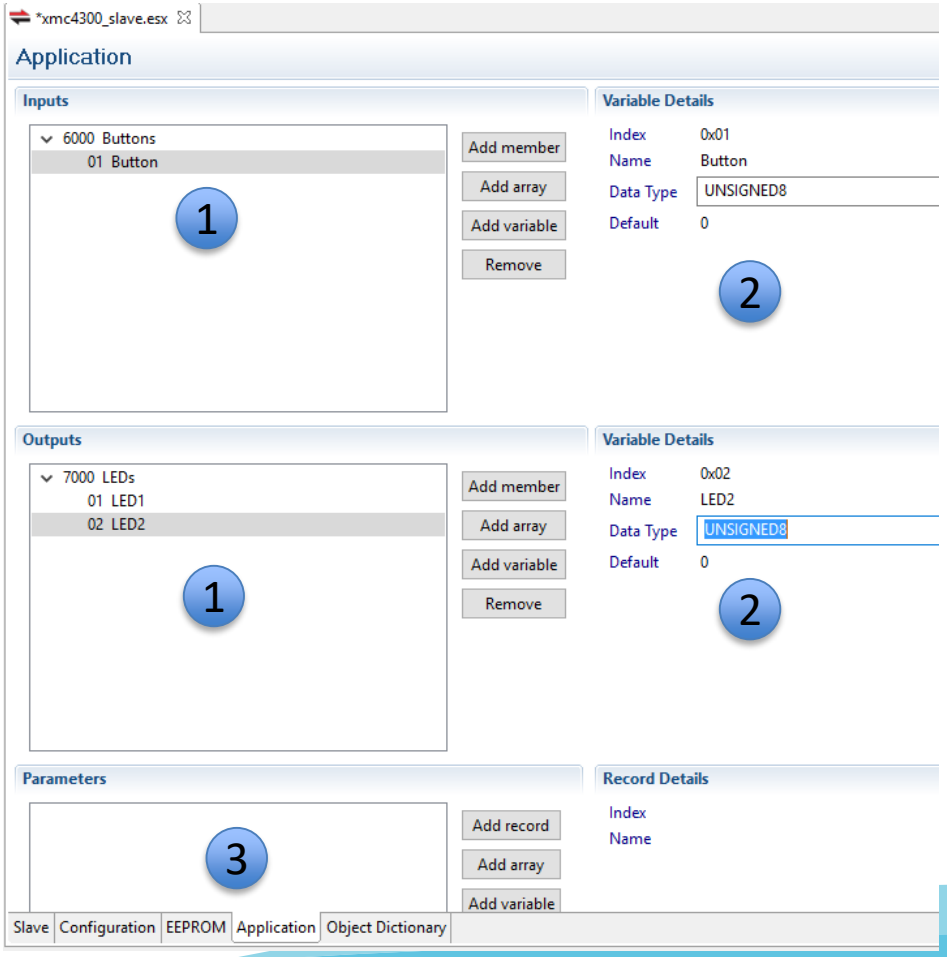


rt:labs

EtherCAT SDK - Slave editor

Application TAB: Add application IO, the Slave Editor will automatically use the correct index ranges according to the Modular Device Profile. The specified PDOs will also automatically be added to the SyncManager , RxPDO and TxPDO objects.

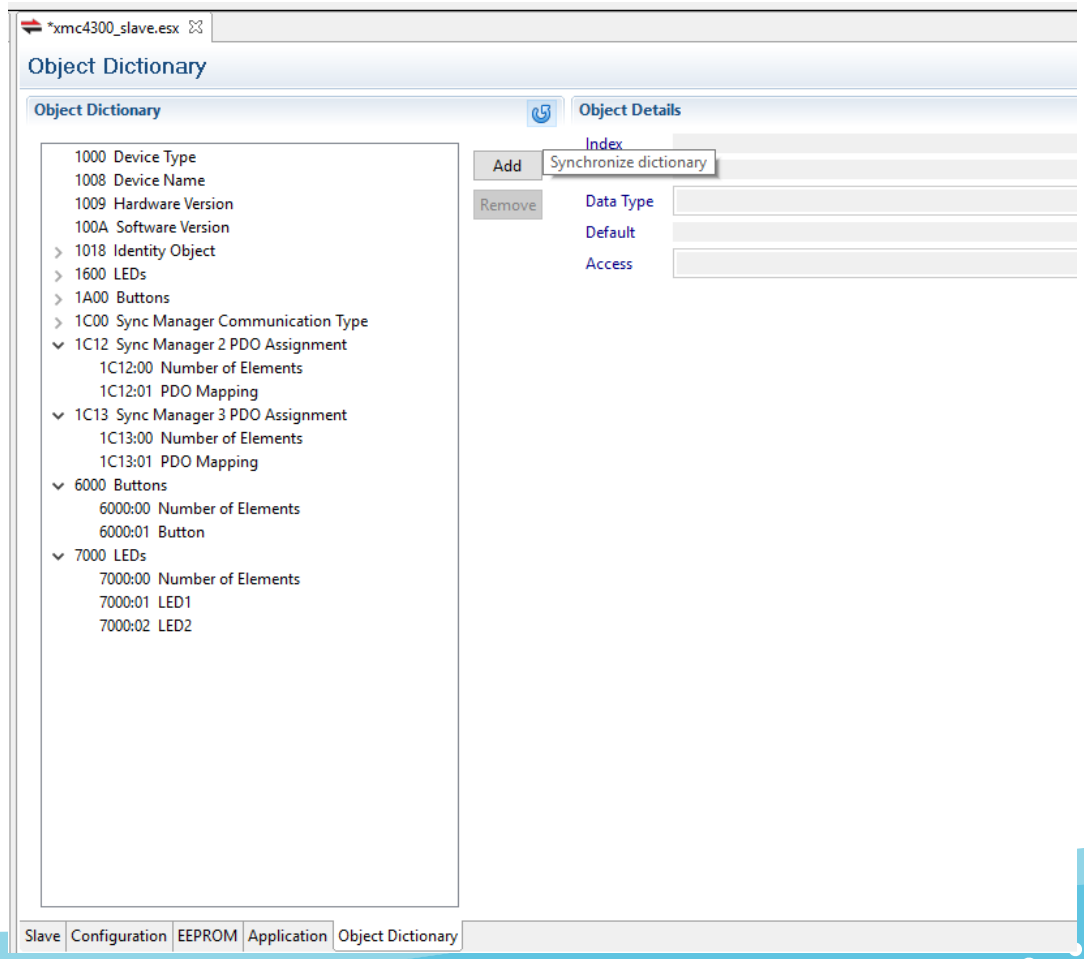
1. Add PDO record which automatically will be added to the TxPDO/RxPDO objects
2. Add PDO record variable with name, datatype and default value
3. Add configuration parameters (optional)



rt:labs

EtherCAT SDK - Slave editor

Object Dictionary TAB: Overview of the resulting Object Dictinay

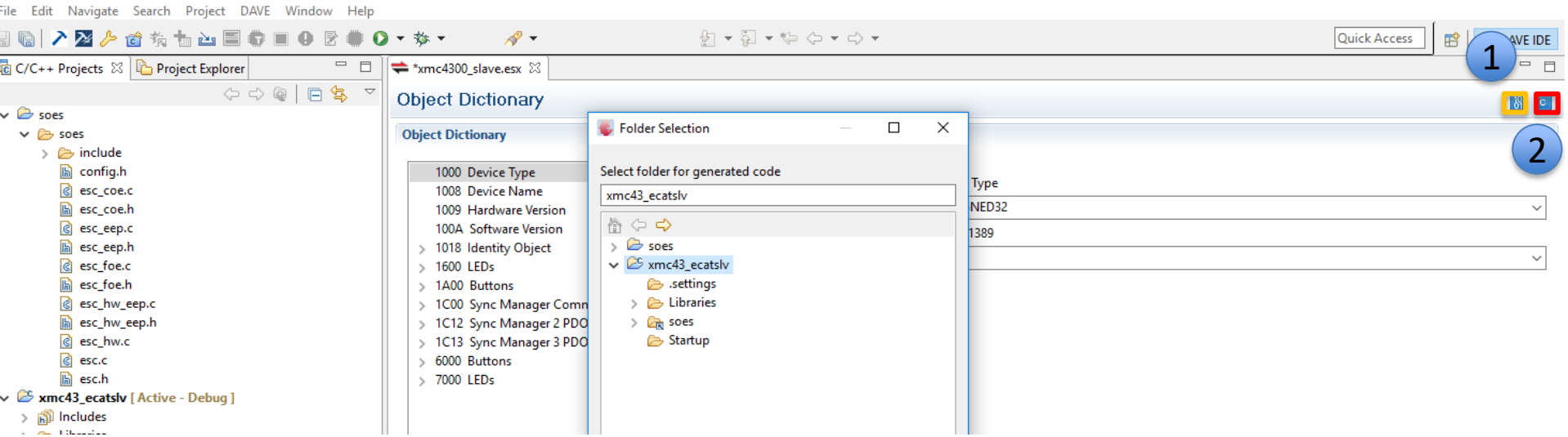


rt:labs

EtherCAT SDK - Slave editor

Generate output:

1. Press button to generate ESI data XML & EEPROM
2. Press button to generate C code, Object Dictionary & Slave application



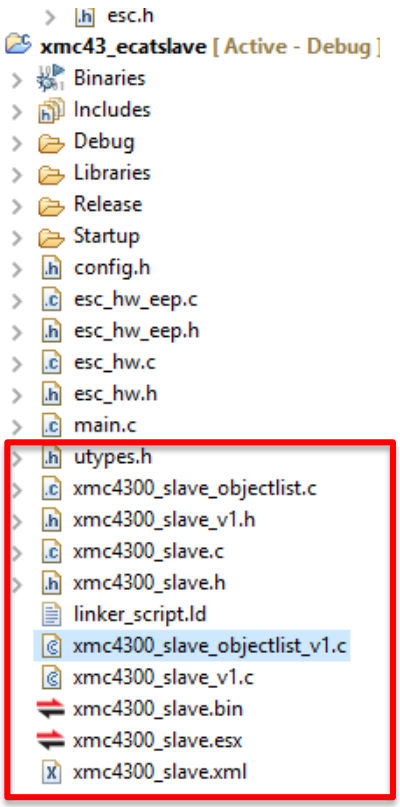
EtherCAT SDK - Slave editor

Overview of the generated output:

- **utypes.h**, types for user defined application data
- <slave_editor project**>_objectlist.c**, Object Dictionary
- <slave_editor project>_**slave.[c,h**], slave stack application functions and declarations
- <slave_editor project>_**slave.bin**, EEPROM
- <slave_editor project>_**slave.xml**, ESI file

Don't EDIT this files since the will be overwritten if code is generated again.

- Files with post fix _v1 are generated for SOES v1 (old), exclude from build from Resource Configruations -> Exclude from Build



rt:labs

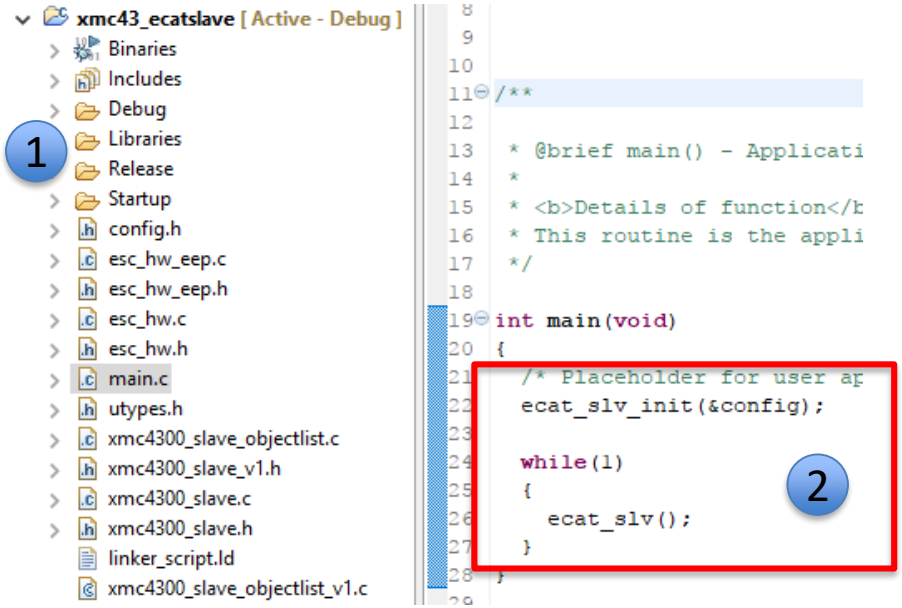
Workflow

1. Overview and Requirements
2. Setup hardware, DAVE and EtherCAT SDK
3. Define the EtherCAT application with EtherCAT SDK
4. Implement the application in DAVE
5. Test the application with EtherCAT SDK

rt:labs

Implement Application – integrate stack

1. Add application code in a NOT generated file.
2. Add calls to the SOES slave stack , **ecat_slv_init** to initialize the stack and **ecat_slv** to run the stack periodically in free running mode.



Implement Application – configure stack

1. The stack is configured by passing a configuration arg to **ecat_slv_init**. Consult the user manual for more information on different options, for a free-run slave the configuration below will do. Copy config from samples in soes\appliciations

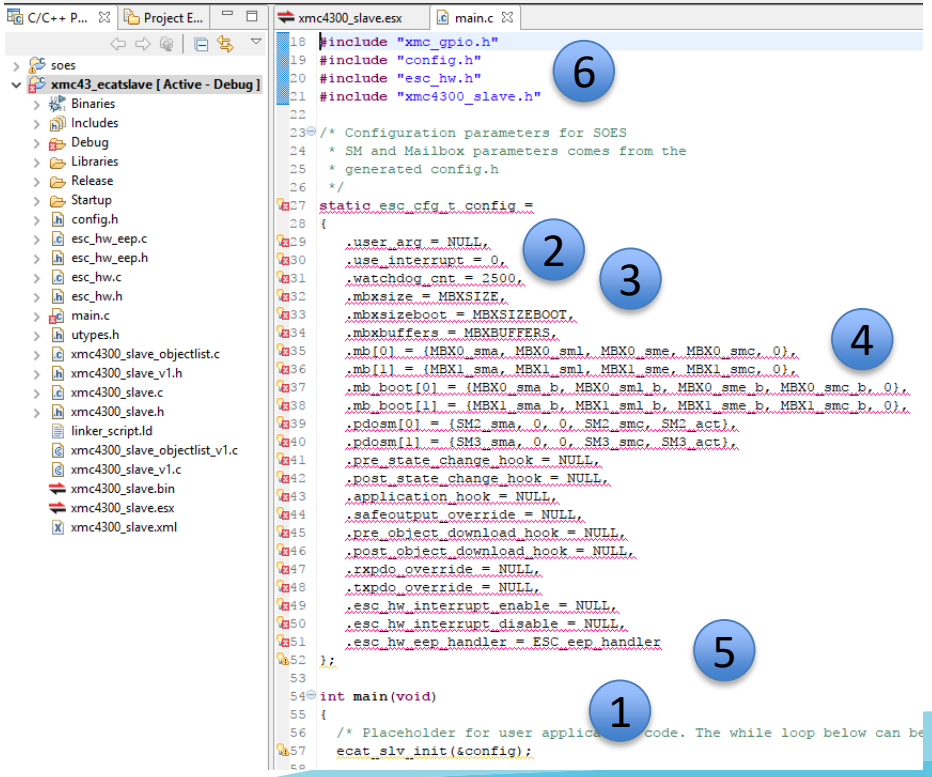
2. use_interrupt = 0

3. watch_dog = <your cycle count>

4. mbx, mb, mb_boot, pdo_sm pass values generated from the SlaveEditor tool in config.h

5. XMC must have an emulated EEPROM handler

6. Add application includes



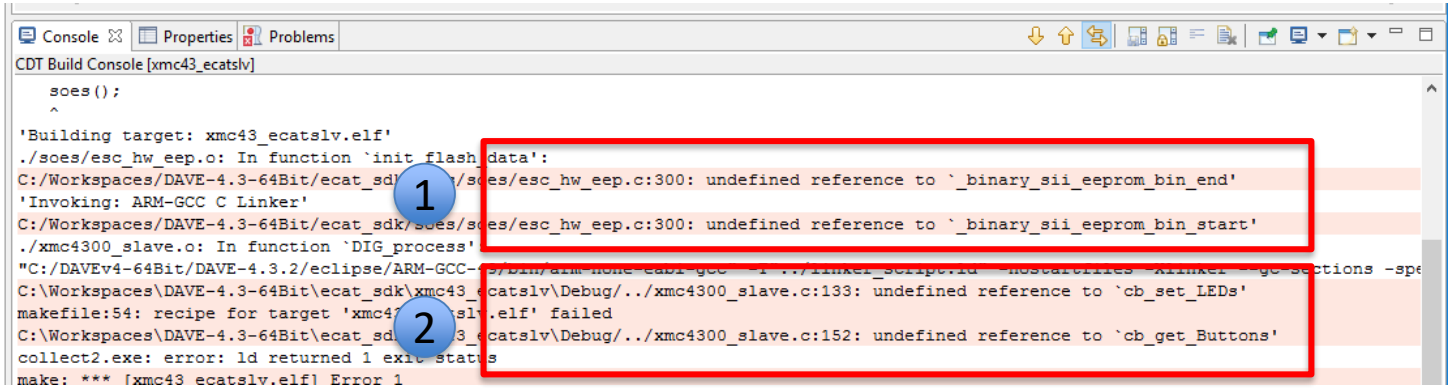
rt:labs

Implement Application – Compile

First time compiling is expected to fail

1. Undefined references to "_binary_sii_eeprom_bin_end" and "_binary_sii_eeprom_bin_start" are extern declared variables in the stack to the emulated EEPROM not yet added. (How to fix in coming slides)

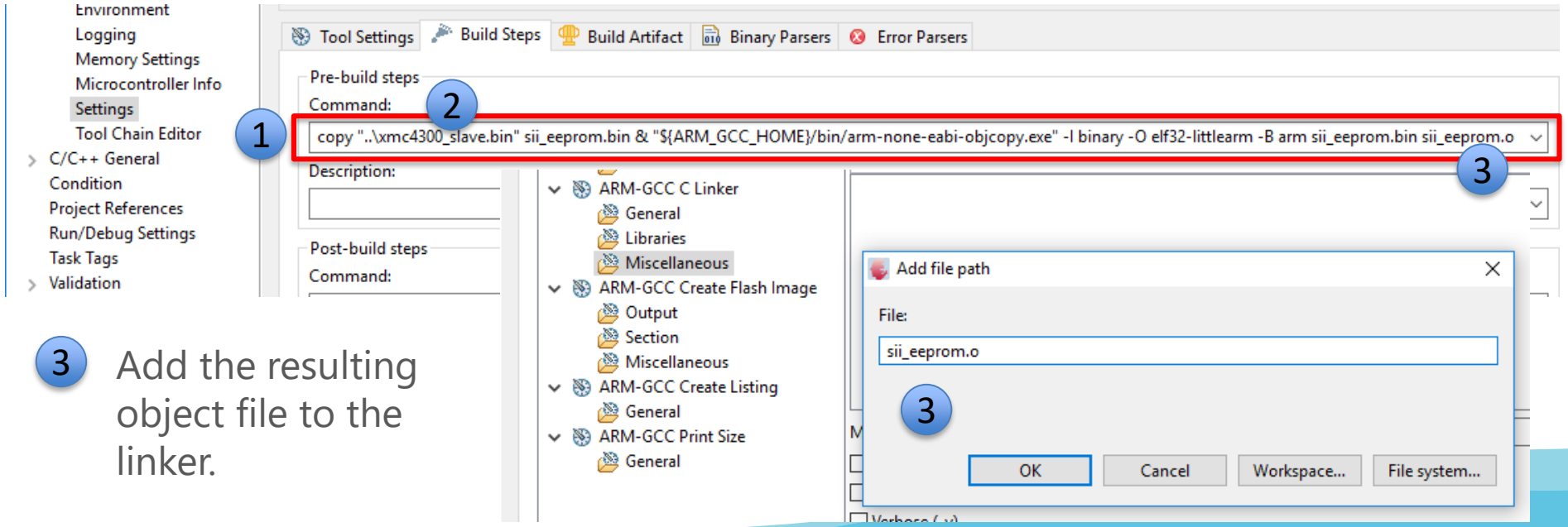
2. cb_set_XX and cb_get_YY are slave stack callbacks for handling our added application IO, these have been generated by the EtherCAT SDK Slave Editor (How to fix in coming slides)



Implement Application - EEPROM

Right click the Application <Project> and select Properties

1. Add an pre-build option to copy and create a linkable object file.
2. The only required change is to add your EEPROM bin in the copy command, eg. replace **xmc4300_slave**.bin with your name.



3. Add the resulting object file to the linker.

Implement Application - GPIO

Add application code in a NOT generated file.

1. Add init of GPIO
2. Add LEDs callbacks
3. Add Buttons callbacks

```c
int main(void)
{
  XMC_GPIO_Init(GPIO_BUTTON1, &gpio_config_btn);
  XMC_GPIO_Init(GPIO_LED1, &gpio_config_led1);
  XMC_GPIO_Init(GPIO_LED2, &gpio_config_led2);

  ecat_slv_init(&config);

  while(1)
  {
    ecat_slv();
  }
}
```

```c
/**
 * This function gets input values and updates Rb.Buttons
 */
void cb_get_Buttons(void)
{
    Rb.Buttons.Button1 = (uint8_t)XMC_GPIO_GetInput(GPIO_BUTTON1);
}
/**
```

```c
/**
 * This function sets output values according to Wb.I
 */
void cb_set_LEDs(void)
{
    if(Wb.LEDs.LED1)
    {
        XMC_GPIO_SetOutputHigh(GPIO_LED1);
    }
    else
    {
        XMC_GPIO_SetOutputLow(GPIO_LED1);
    }

    if(Wb.LEDs.LED2)
    {
        XMC_GPIO_SetOutputHigh(GPIO_LED2);
    }
    else
    {
        XMC_GPIO_SetOutputLow(GPIO_LED2);
    }
}
```

rt:labs

EtherCAT SDK - Agile EtherCAT development

- The whole idea with the slave editor is to keep EEPROM, ESI and OD aligned, you should always be CTT ready.
- To continue editing, just double click the Slave Editor project file, re-generate the code, build and debug.
- Test the slave with EtherCAT SDK – EtherCAT Explorer

rt:labs

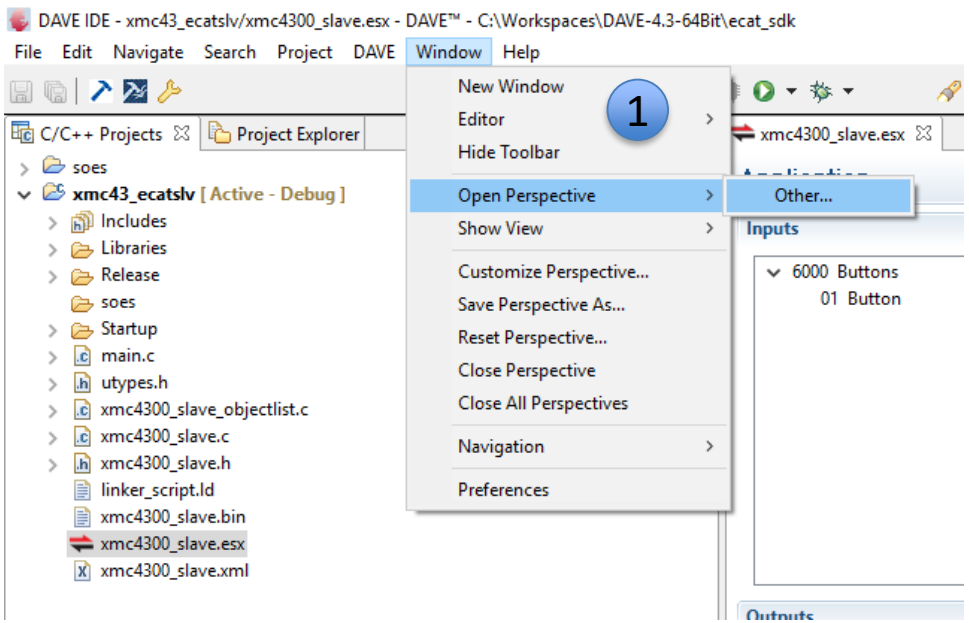
Workflow

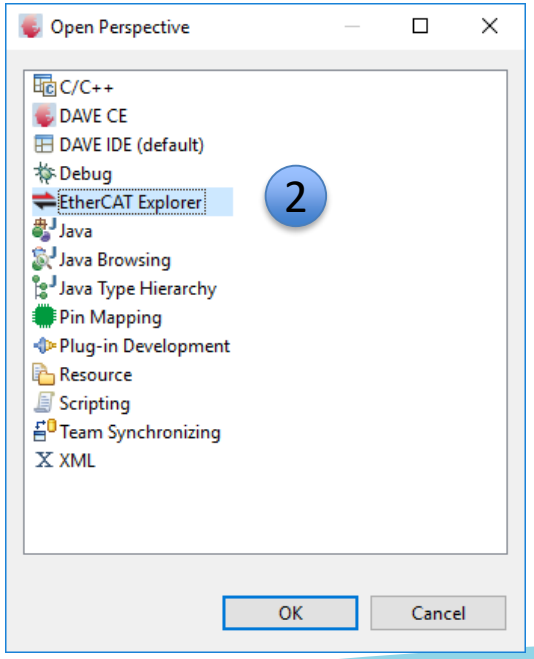
1. Overview and Requirements
2. Setup hardware, DAVE and EtherCAT SDK
3. Define the EtherCAT application with EtherCAT SDK
4. Implement the application in DAVE
5. Test the application with EtherCAT SDK

rt:labs

Test Slave – Open EtherCAT Explorer

1. Go to menu: Windows -> Open Perspective -> Other …
2. Select EtherCAT Explorer

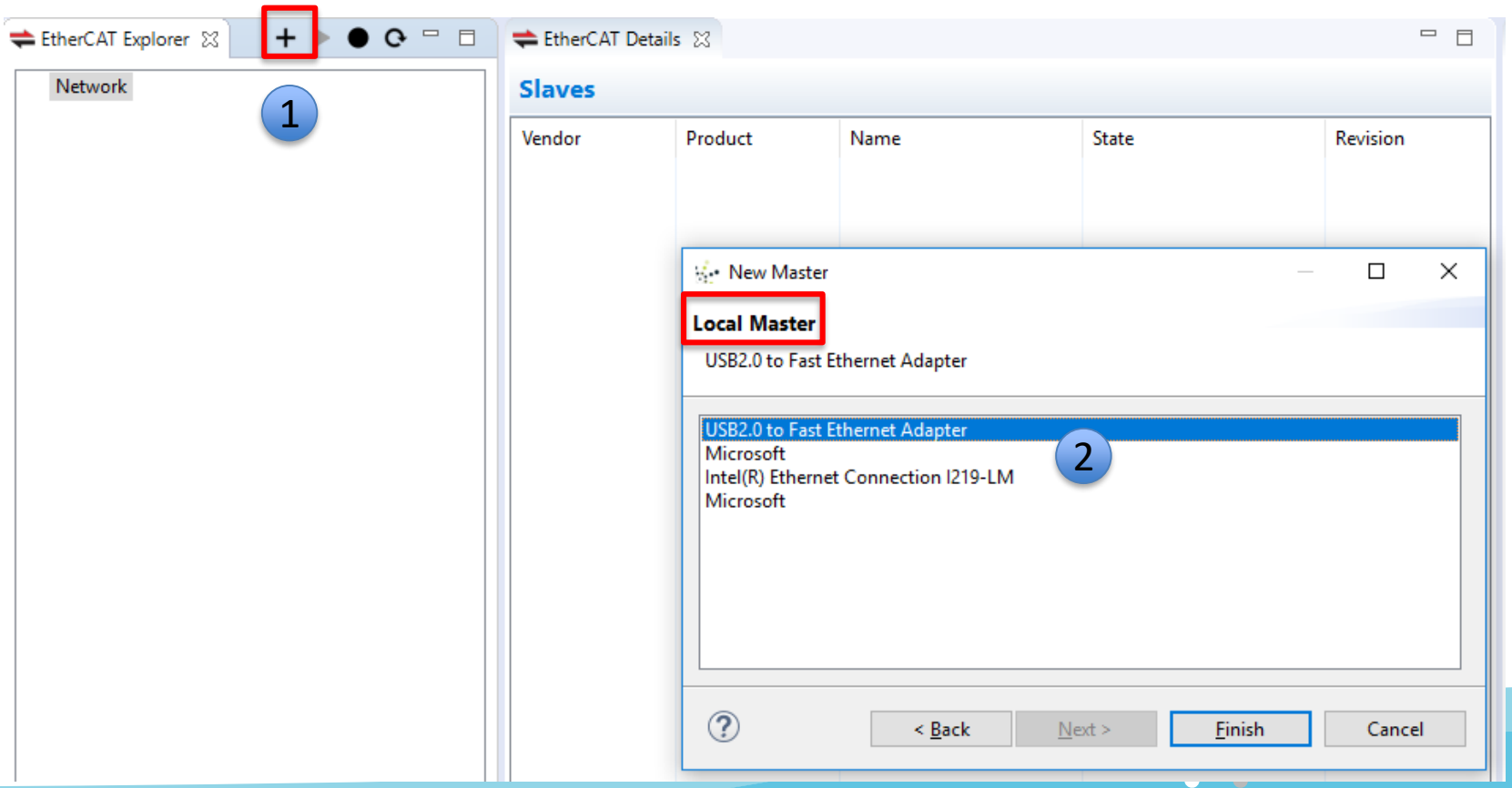




rt:labs

Test Slave- Select NIC for Master

1. Open and select NIC for a Local Master (Remote is when running EAP)
2. Select one of available adapters (Ethercat Explorer require Winpcap)

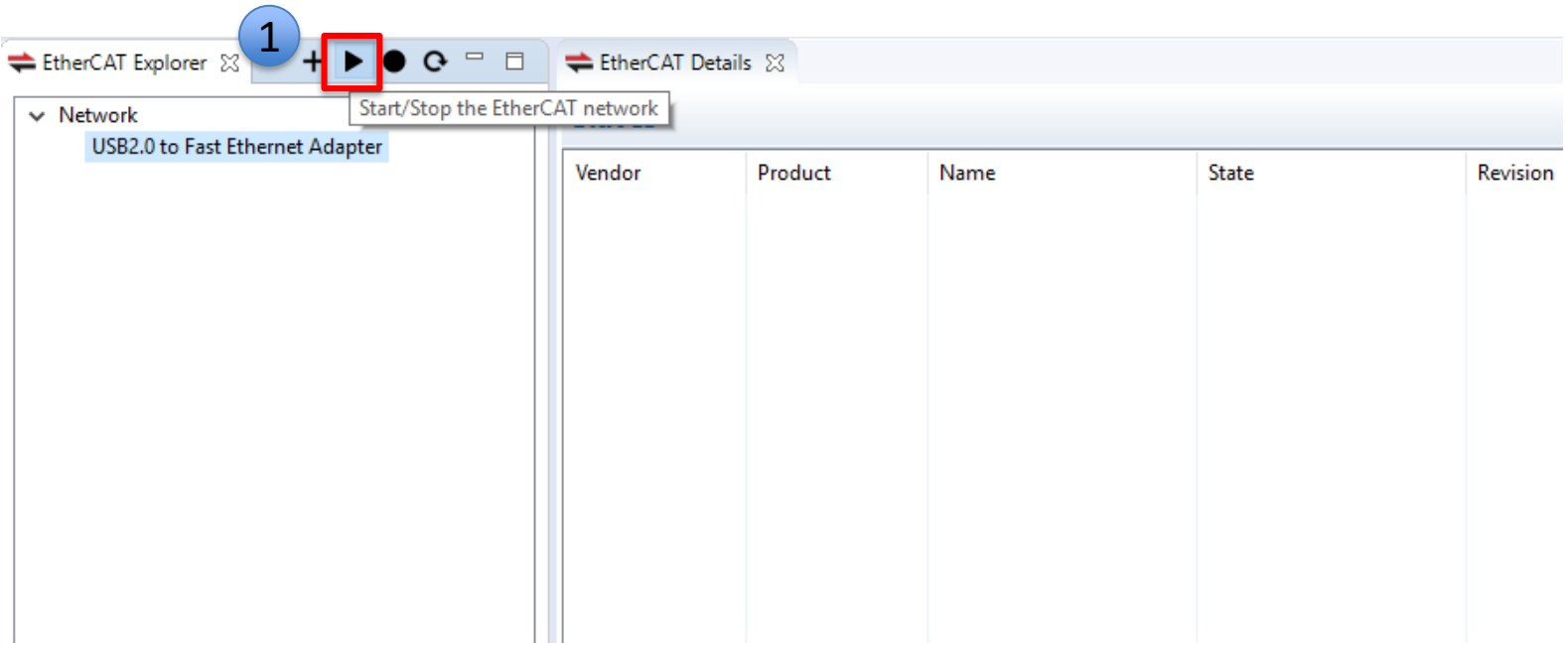


rt:labs

Test Slave - EtherCAT Explorer

Make sure the EtherCAT Relax Kit slave is started and connected to the Laptop

1. Start the master with the arrow. A started master is shutdown with the arrow.



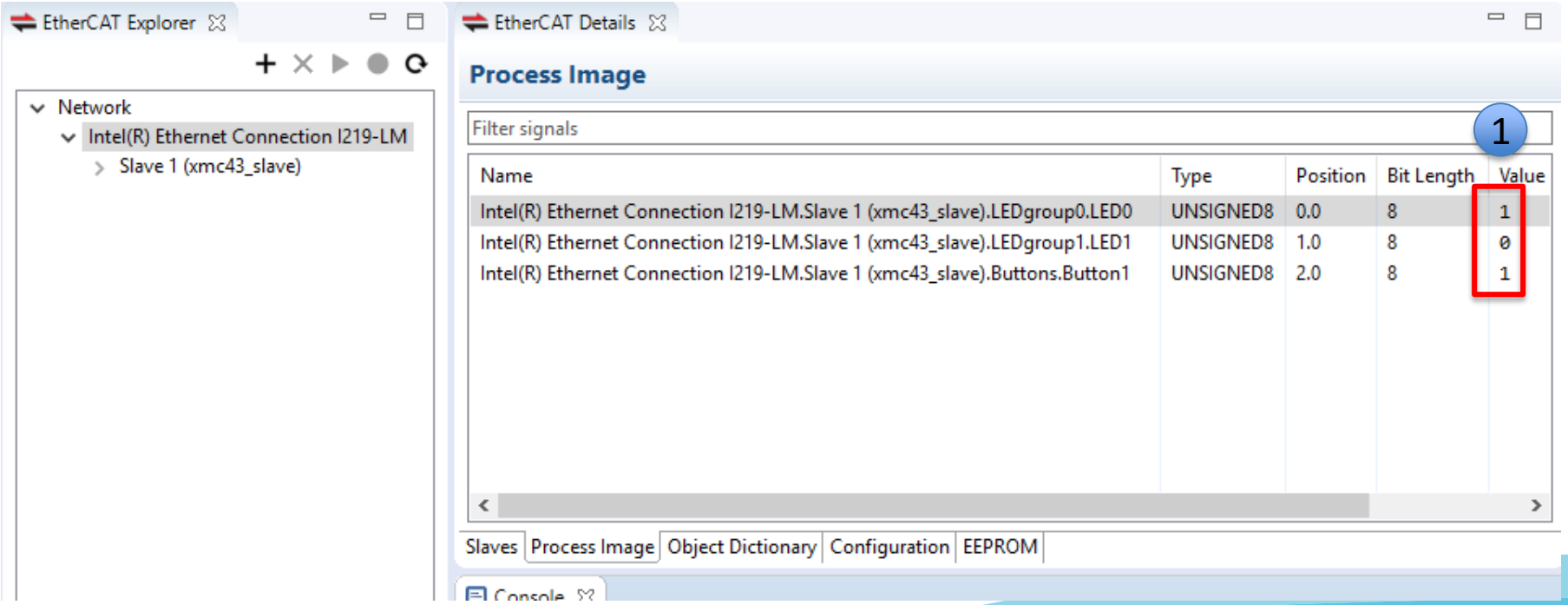
rt:labs

Test Slave- Examine the ProcessData

1 Explore the slave process data,

Action: Press the Relax Kit button , **Result:** The value in the GUI should toggle

Action: Click a LED and enter value 1, **Result:** The LED should go On on the Relax Kit.



rt:labs

EtherCAT SDK - EtherCAT Explorer

1 Explore the slave Object Dictionary. Refresh values with arrow

EtherCAT Explorer

- Network
  - USB2.0 to Fast Ethernet Adapter
    - Slave 1 (xmc43_slave)

EtherCAT Details

Slave 1 (xmc43_slave)

1

Filter parameters

Index	Name	Datatype	Access	Value
1000	Device Type			
1008	Device Name			
1009	Hardware Version			
100A	Software Version			
1018	Identity Object			
1600	LEDs			
1600:00	Number of Elements	UNSIGNED8	RO	2
1600:01	LED1	UNSIGNED32	RO	1879048456
1600:02	LED2	UNSIGNED32	RO	1879048712
1A00	Buttons			
1A00:00	Number of Elements	UNSIGNED8	RO	1
1A00:01	Button	UNSIGNED32	RO	1610613000
1C00	Sync Manager Communi...			
1C12	Sync Manager 2 PDO Assi...			
1C13	Sync Manager 3 PDO Assi...			
6000	Buttons			
6000:00	Number of Elements	UNSIGNED8	RO	1
6000:01	Button	UNSIGNED8	RO	1
7000	LEDs			

1

Slaves | Process Image | Object Dictionary | Configuration | EEPROM

rt:labs

Final step – run CTT

Untitled.ctp* - EtherCAT Conformance Test Tool

File Tool Action Help

Project Explorer

- Untitled*
  - Tests
    - EtherCAT
      - TF-1300 EtherCAT Slave Informatior
      - TF-1100 Data Link Layer
      - TF-1200 EtherCAT State Machine
      - TF-1201 ESM - Explicit Device Ident
      - TF-2300 AL CoE SDO Service
      - TF-2302 CoE SDO Complete Access
      - TF-2301 AL CoE Object Dictionary
      - TF-4100 CiA402 OD
  - EtherCAT Devices
    - EtherCAT
      - ProcessImage
      - Slave1 xmc4300_slave[Default]*
        - TxPdos
        - RxPdos

Test | Slave1 xmc4300_slave[Default] | Start Page

General Info | EEPROM | Memory | ESI | Process Data | Mailbox | Startup | CoE Online | State Machine

C:\Program Files (x86)\EtherCAT Conformance Test\DeviceDescriptions\xmc4300_slave.xml

- DeviceType
  - Invisible
  - Physics = YY
  - Crc32
  - Type
  - HideType (0)
  - AlternativeType (0)
  - Name (1)
  - Comment (0)
  - URL (0)
  - Info
  - GroupType
  - Profile (1)
  - Fmmu (3)
  - Sync Manager (4)
  - RxPdos (1)
  - TxPdos (1)
  - Mailbox
  - Distributed Clocks
  - Slots
  - EtherCAT Slave Controller

Attributes of DeviceType	
Invisible	
Physics	YY
Crc32	
DeviceType	
Type	xmc4300_slave
HideType	No Items
AlternativeType	No Items
Name	Items: 1
Comment	No Items
URL	No Items
Info	not specified
GroupType	xmc4300
Profile	Items: 1
Fmmu	Items: 3
Sync Manager	Items: 4
RxPdos	Items: 1
TxPdos	Items: 1
Mailbox	specified
Distributed Clocks	not specified
Slots	not specified

Name
Detailed name of device shown by a configuration tool (not used for identification)

Logger

211 Success | 0 Warnings | 0 Errors | 135 Skipped | 1163 Outputs | 4770 Verbose

Log No.	Time Stamp	TestCase	Info
1	2017-01-21 17:11:24:467	Secondary V...	Successful | Valid Vendor ID in ESI file (0x1337)
3	2017-01-21 17:11:24:492	Device.Revis...	Success | ESI element Device:RevisionNo is used.
11	2017-01-21 17:11:24:507	ESI Port Typ...	Success | Port configuration elements in ESI are valid.

Logger | Tool Log

Licensed to: RT-Labs AB

rt:labs

EtherCAT provided by rt-labs

Visit us at :

http://www.rt-labs.com/ethercat/why-choose-rt-labs/

https://github.com/OpenEtherCATsociety

rt:labs