Reference Guide

SPRUIG9–September 2017

TMDSECATCNCD379D EtherCAT® Solution Reference Guide

The scope and purpose of this document is to provide users with a guide to build an EtherCAT® sample application for the TMDSECATCNCD379D hardware kit using the EtherCAT slave stack and C28x EtherCAT sample application sources generated from the ETG slave stack code (SSC) tool. Users can use the SSC tool patches delivered by TI for C28x configuration if the SSC tool does not come built-in with the C28x configurations shown in this document.

Use this document along with the EtherCAT Interface for High-Performance C2000 MCU User's Guide.

The EtherCAT slave stack referred to in this document, and provided with the demo binaries for the TMDSECATCNCD379D kit, is obtained by porting the ETG EtherCAT slave stack generated from the SSC tool onto the C28x MCU. Section 9 of this document provides a link to the ETG website for users to obtain the SSC tool.

This user's guide describes the procedure to generate EtherCAT slave stack sources ported for the C28x MCU using the C28x configuration for SSC tool.

Contents

List of Figures

List of Tables

Trademarks

Code Composer Studio, C2000, LaunchPad are trademarks of Texas Instruments.
EtherCAT is a registered trademark of Beckhoff Automation.

1 Definitions, Abbreviations, and Acronyms

Table 1 lists the definitions, abbreviations, and acronyms.

Table 1. Definition List

Term	Definition
CCS	Code Composer Studio™
COFF	Common object file format (final application output file that can be loaded on to the target and run, sometimes called OUT file)
CTT	Conformance test tool
DIGIO	Digital I/O profile
DP83822	Ethernet PHY from TI
DSS	Debug server scripting (for debug, load, run of COFF files from command line)
EMIF	External memory interface peripheral
ENI	EtherCAT network information
ESC	EtherCAT slave controller
ESI	EtherCAT slave information
ET1100	Beckhoff EtherCAT slave controller
ETG	EtherCAT Technical group
EtherCAT	Ethernet for Control Automation Technology
F2837xD	TMS320F2837xD Delfino microcontroller
HAL	Hardware abstraction layer
PDI	Processor data interface
SSC	Slave stack code
TwinCAT	EtherCAT master software from Beckhoff which runs on a PC
TMDSECATCNCD379D kit	EtherCAT kit from TI C2000™ for TMS320F28379D MCU with ET1100 ESC

1.1 Assumptions

Users are aware of the TMDSECATCNCD379D EtherCAT® Interface for High-Performance C2000 MCU User's Guide and the TI Design for EtherCAT on Delifino.

2 TMDSECATCNCD379D Hardware Kit

Figure 1 and Figure 2 show the hardware kit.

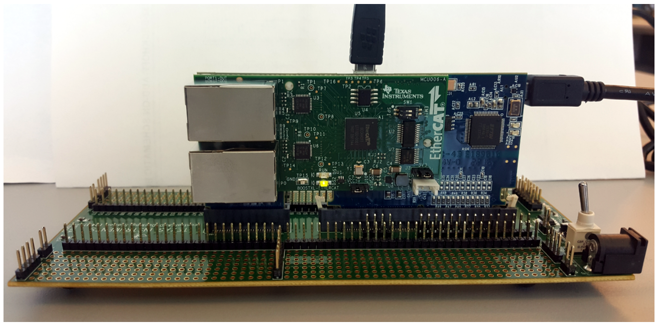

Figure 1. TMDSECATCNCD379D Hardware Kit (Micro USB)

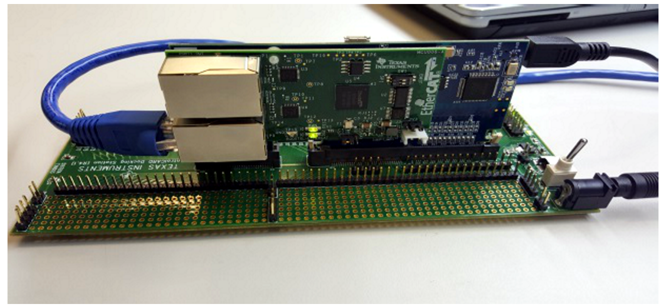

Figure 2. TMDSECATCNCD379D Hardware Kit (5V Supply)

The board can be powered using a Micro-USB cable, as shown in Figure 1, coming in from the top. JTAG can be accessed to program software on the board using a Mini-USB cable, as shown in Figure 2, coming from the right.

When connected to the docking station, the board can also be powered with a 5-V supply; in this case the Micro-USB power, shown in Figure 1, is not needed. Use SW1 to switch between ASYNC16 (EMIF) and SPI PDI (see Figure 3). The switch position must match the PDI for which the slave stack sources are generated by the SSC tool, and also the respective Build Config chosen when building the software project.

The software project build configurations are explained further in the document.

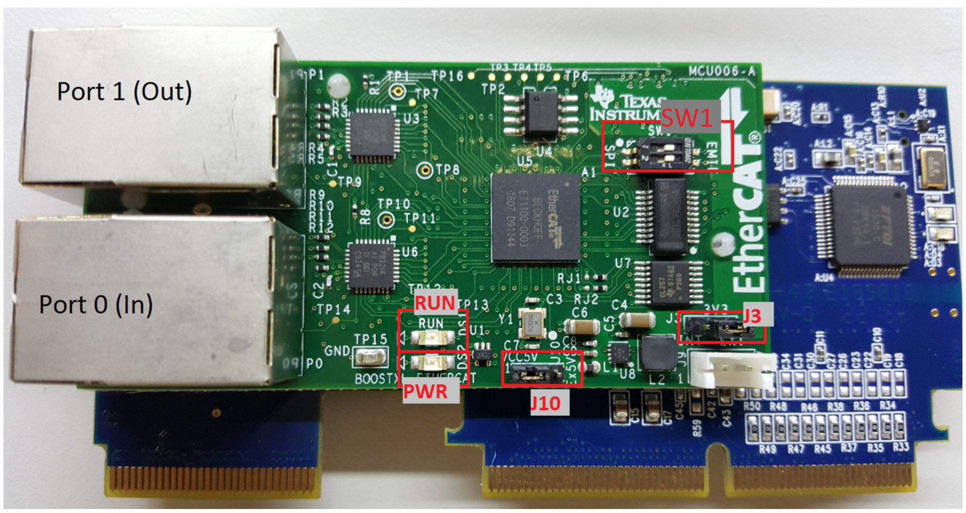

Figure 3. TMDSECATCNCD379D SW1 Position

Table 2 lists the EtherCAT daughter card LED and switch use descriptions.

Table 2. EtherCAT Daughter Card LED and Switch Use Descriptions

Name	Options	Description
Switches and jumpers		
SW1	L – SPI R – EMIF	Selects between EMIF and SPI modes
J3	1 - 2 Offboard 2 - 3 Onboard	Offboard: 3.3 V is provided directly from attached LaunchPad™ or ControlCARD Onboard: 3.3 V is generated by the onboard regulator from a separate 5-V supply.
J10	1 - 2 Onboard 2 - 3 Offboard	Onboard: 5 V provided by LaunchPad or ControlCARD Offboard: 5 V provided externally through header J9.
LEDs		
RUN LED	State machine status[1]	OFF: ET1100 device is in INIT state. ON: ET1100 device is in Operational state.
DS2/PWR LED	3.3-V power	ON indicates 3.3 V is being supplied to the board.

(1) Additional RUN LED states given in Table 54 of the ET1100 data sheet.

NOTE: Ensure J10 and J3 are both set to position 1-2 before powering the kit. Also ensure SW1 is set to the PDI that the user is evaluating for correct operation.

3 Deliverables Explained

The user will work with the following highlighted software deliverables in conjunction with this document (see Figure 4). All the software pertaining to the TMDSECATCNCD379D kit is delivered to users through ControlSuite. The software is in the ControlSuite\development_kits\TMDSECATCNCD379D folders.

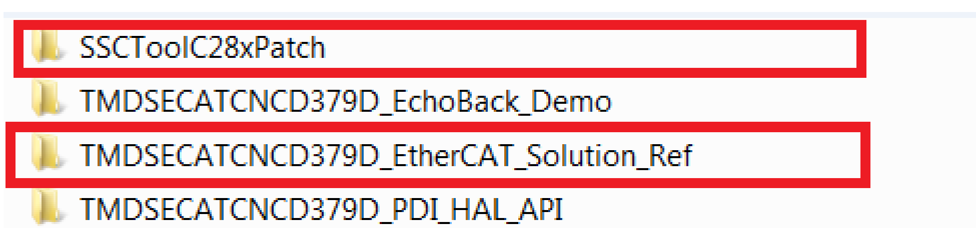

Figure 4. Software Deliverables

- The SSCToolC28xPatch folder contains the SSC tool patches for C28x configuration. As previously mentioned, the SSC tool is the slave stack code generator that users must obtain from the ETG website. Section 9 provides a link for the website.
- The TMDSECATCNCD379D_EtherCAT_Reference folder contains the CCS project for customers to build a full EtherCAT Echoback sample application with the EtherCAT slave stack.

3.1 SSC Tool C28x Patch Folder

The files under the files folder are used by the SSC tool when the user imports the C28xx_Config.xml into the SSC tool (see Figure 5).

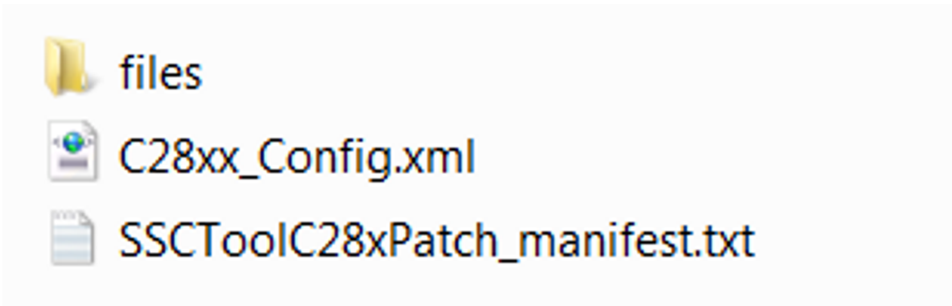

Figure 5. SSC Tool C28x Patch Folder

3.2 TMDSECATCNCD379D_EtherCAT_Reference

This folder contains a CCS project that the user can build to get to the source-level debug of the EtherCAT slave stack and a sample application running on the C28x MCU (see Figure 6).

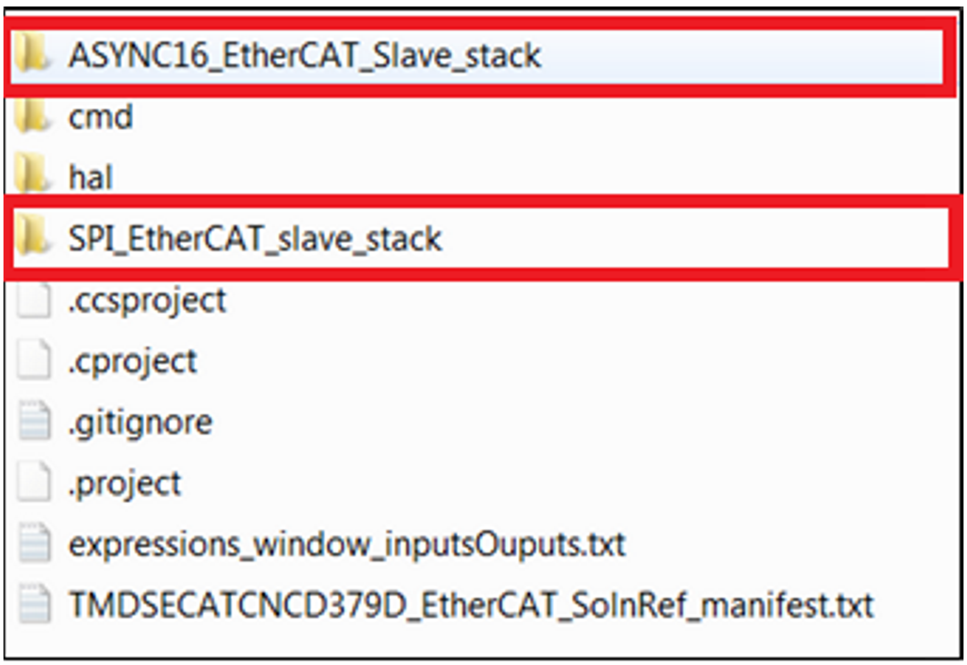

Figure 6. Reference Project Contents

The EtherCAT reference project has two empty placeholder folders (see Figure 6). Users must copy the SSC tool-generated sources for the C28x configuration into these folders.

- When the ASYNC16 option is used, users must copy the SSC tool generated sources to the ASYNC16_EtherCAT_Slave_stack folder.
- When the SPI option is used, users must copy the SSC tool generated sources to the SPI_EtherCAT_slave_stack folder.

The files under these folders are used depending on the build configuration used. Users do not need to populate both folders.

3.2.1 EtherCAT Reference Project Build Configurations

The TMDSECATCNCD379D kit supports two types of PDI for ET1100 (see Table 3). The SPI PDI port is interfaced with the SPI port on the TMS320F28379D MCU, and the ASYNC16 PDI is interfaced with EMIF on the TMS320F28379D MCU.

NOTE: The TMS320F28379D MCU has two EMIF ports: EMIF1 and EMIF2. The software and hardware interfaces use EMIF2 by default, but users who want to use EMIF1 in their own design can do so. The instructions are provided in the EtherCAT® Interface for High-Performance C2000 MCU User's Guide. Similarly, among the SPI ports available on the MCU, the SPI_C port is used by default in the hardware and software, but customers can emulate other SPI options with the provided software by enabling respective macros in the software. The software for using alternate SPI ports is provided, but is disabled by default.

The EtherCAT solution reference project supports build options to generate application code for both SPI and ASYNC16 PDI options.

Table 3. Reference Project Build Configurations

CCS Project Build Configuration	PDI Port	Description
_1_F2837xD_CCARD_EMIF_FLASH	ASYNC16	Software running from flash memory, software retained through power cycle
_2_F2837xD_CCARD_EMIF_RAM	ASYNC16	Software running from RAM, software not retained through power cycle
_3_F2837xD_CCARD_SPIC_FLASH	SPI	Software running from flash memory, software retained through power cycle
_4_F2837xD_CCARD_SPIC_RAM	SPI	Software running from RAM, software not retained through power cycle

The RAM configurations are usually helpful during development or initial stages of the project when the code changes are frequent, such that loading flash for every code change takes too much time compared to code change and test cycle.

The flash configurations are helpful when software is a bit mature and ready to be tested through power cycles of the target device. When building options 1 and 2, only the ASYNC16 slave stack sources folder is used.

When building options 3 and 4, only the SPI slave stack sources folder is used.

4 Generating EtherCAT Slave Stack Sources

TI provides customers with the followings files, which are applied to the SSC tool to generate a slave stack code configuration for the C28x.

NOTE: If the SSC tool natively supports C28x configuration, then there is no need for to use the patches provided.

Users must download the SSC tool from the ETG website: EtherCAT slave stack code – ET9300 from ETG/Beckhoff.

4.1 TI Provided Files

- C28xx_Config.xml – This is the .xml file to be imported into the SSC tool.
- files/c28xxhw.h – This file contains the C28x HAL MAP.
- files/c28xxhw.c – This file contains hardware INIT for the C28x.
- files/c28xx.patch – This file contains the slave stack patches for the C28x.
- files/TMDSECATCNCD379D-EchoBack.c – This file is a sample EtherCAT application for the C28x.
- files/TMDSECATCNCD379D-EchoBack.h – This file is a sample EtherCAT application for the C28x.
- files/TMDSECATCNCD379D-EchoBackObjects.h – This file is a sample EtherCAT application for the C28x.

Figure 7 shows the folder view of the SSC tool patches delivered by TI.

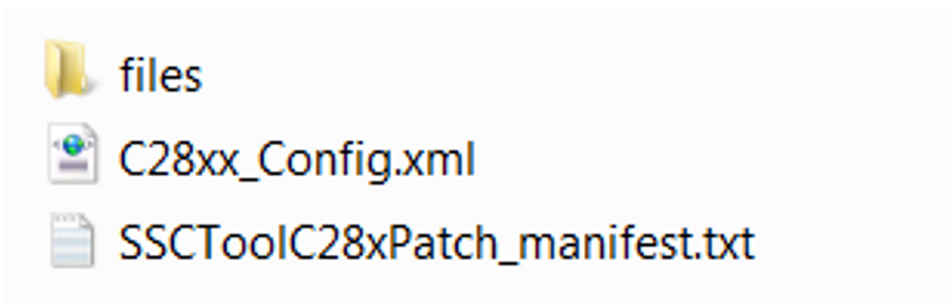

Figure 7. C28x SSC Patches Contents

4.2 Generating SSC for C28x

1. After installing the SSC tool mentioned in Section 4, open the SSC tool and create a new project. The following dialog box appears (see Figure 8).

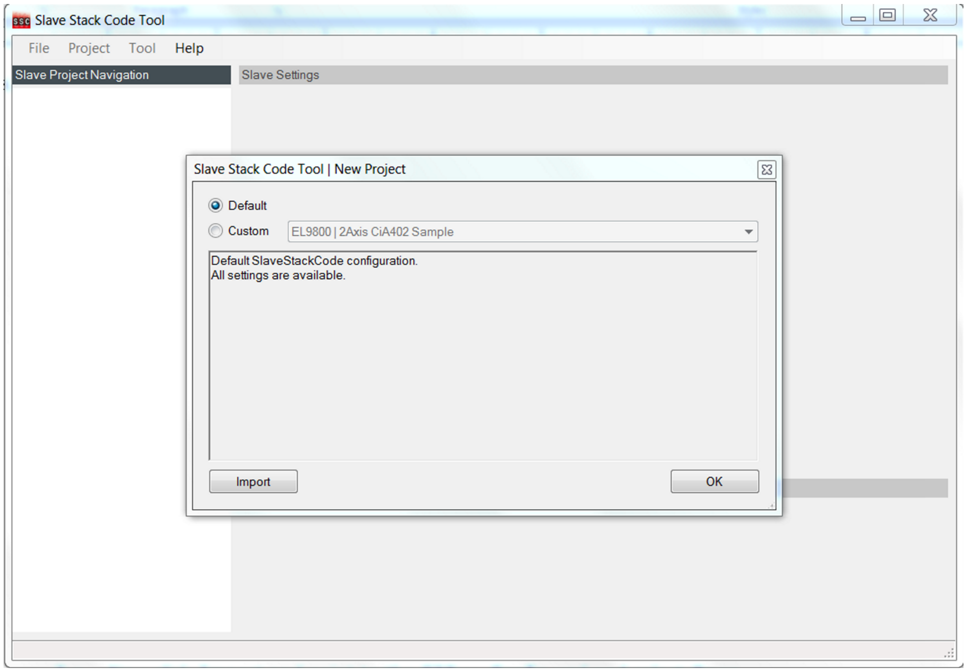

Figure 8. New Project

2. Click Import, point to the C28xx_Config.xml, and then select Open (see Figure 9).

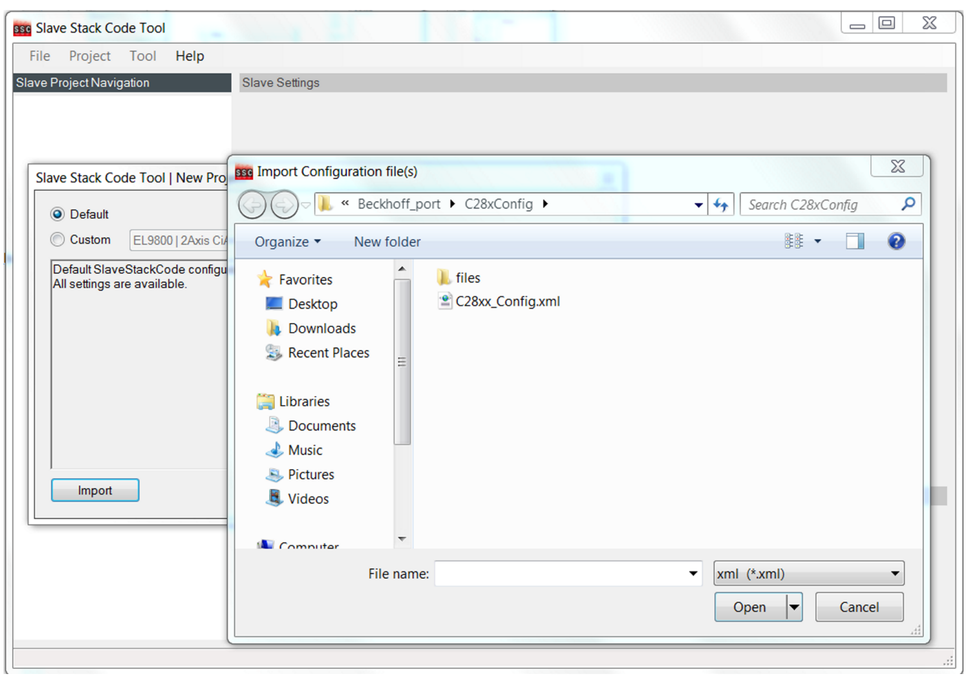

Figure 9. Import Configuration

The following screen is shown when the C28xx_Config.xml is imported for the first time by the SSC tool (see Figure 10).

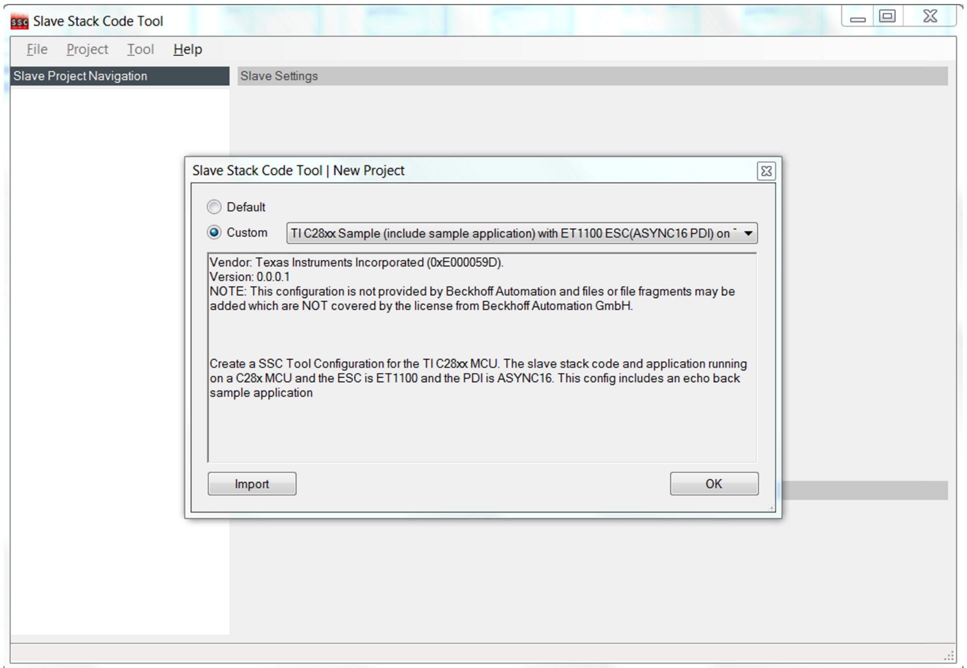

Figure 10. Initial Import

3. When the user selects the drop-down menu, the following options are provided (see Figure 11).

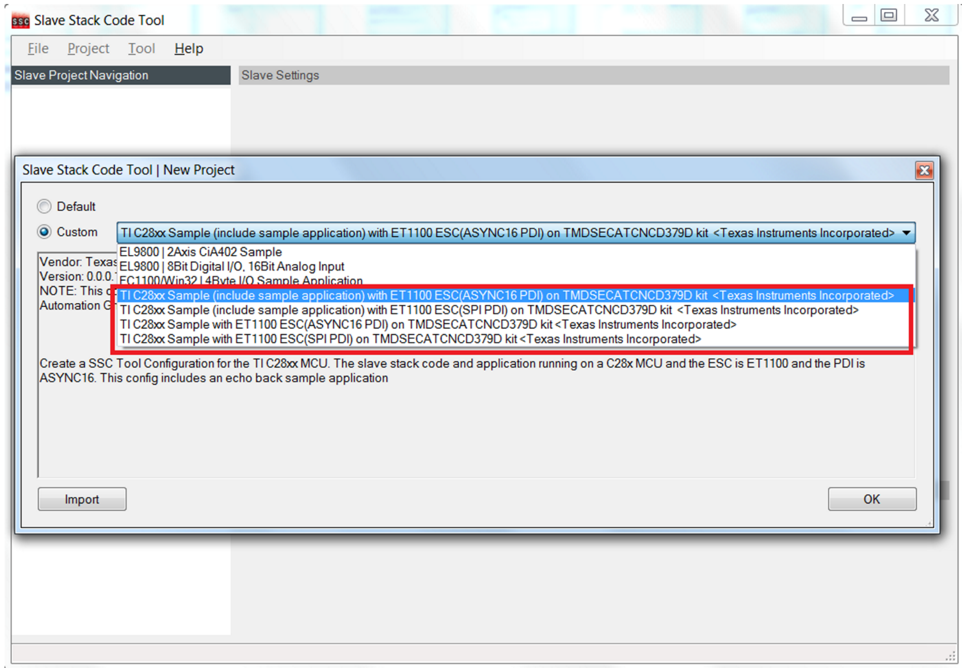

Figure 11. New Project Options

Four options are provided for the C28xx, as follows:

- Option 1 generates EtherCAT slave stack code and EtherCAT EchoBack sample application code for ASYNC16 PDI.
- Option 2 generates EtherCAT slave stack code and EtherCAT EchoBack sample application code for SPI PDI.
- Options 3 and 4 generate EtherCAT slave stack code for ASYNC16 and SPI PDI, without any default EchoBack sample application.

NOTE: Among SPI and ASYNC16 PDIs, there is no difference between the EtherCAT slave stack code and application code. Only the device name and product code differ, so both SPI and ASYNC16 slave nodes can be differentiated when they are both in the same network.

For the EchoBack slave node profiles, the ESI files generated for SPI and ASYNC16 PDIs are also the same except for the device name and product code.

4. Choose an option (preferably one with a sample application), then click OK and click Yes, as shown in Figure 12.

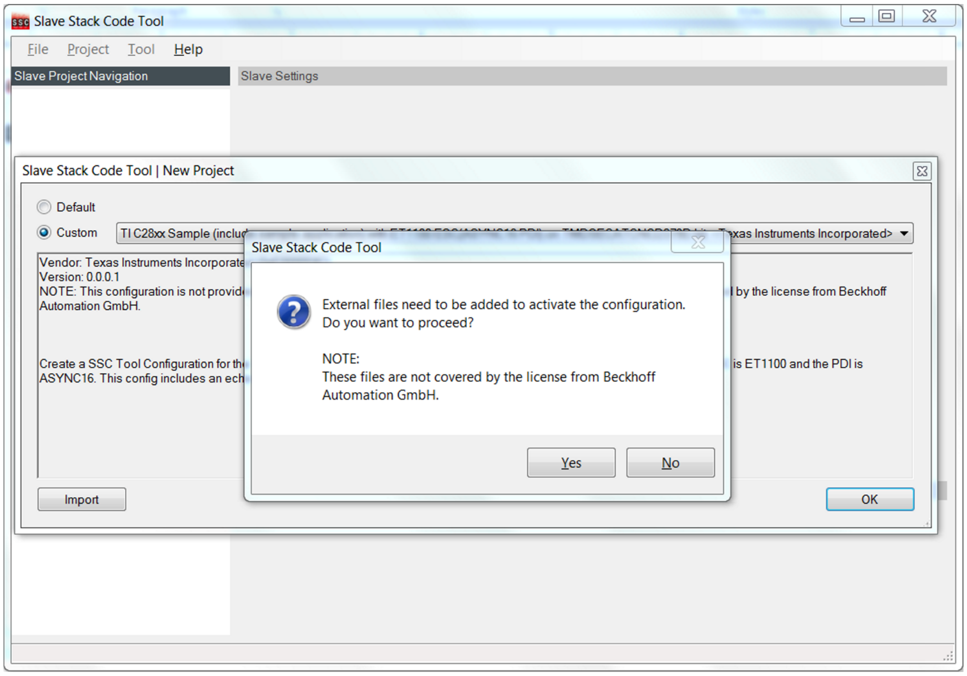

Figure 12. Project Selection

5. Now the C28xx configuration should be imported. Inspect the slave information (see Figure 13).

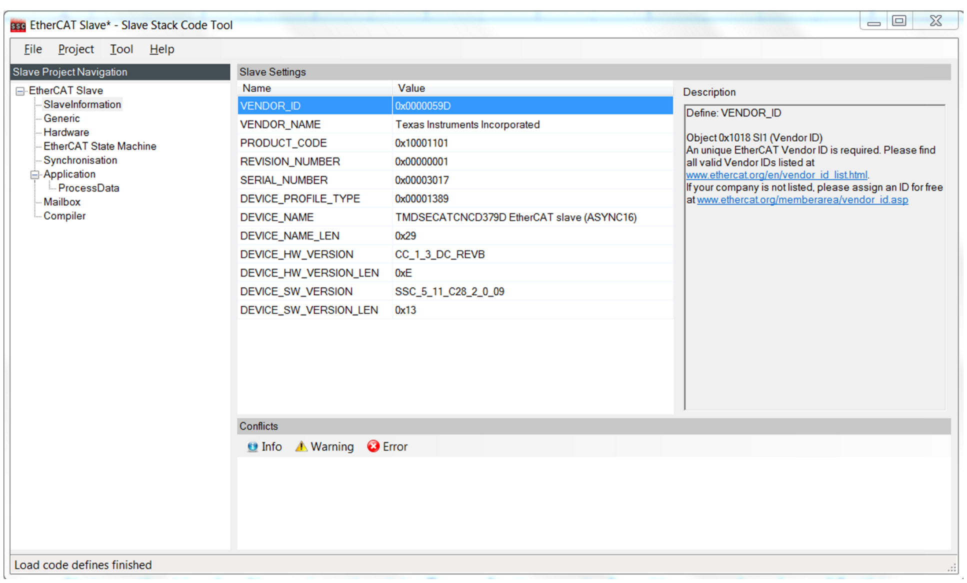

Figure 13. Slave Information

6. Save the SSC Project.

7. Select Project → Create new Slave Files (see Figure 14).

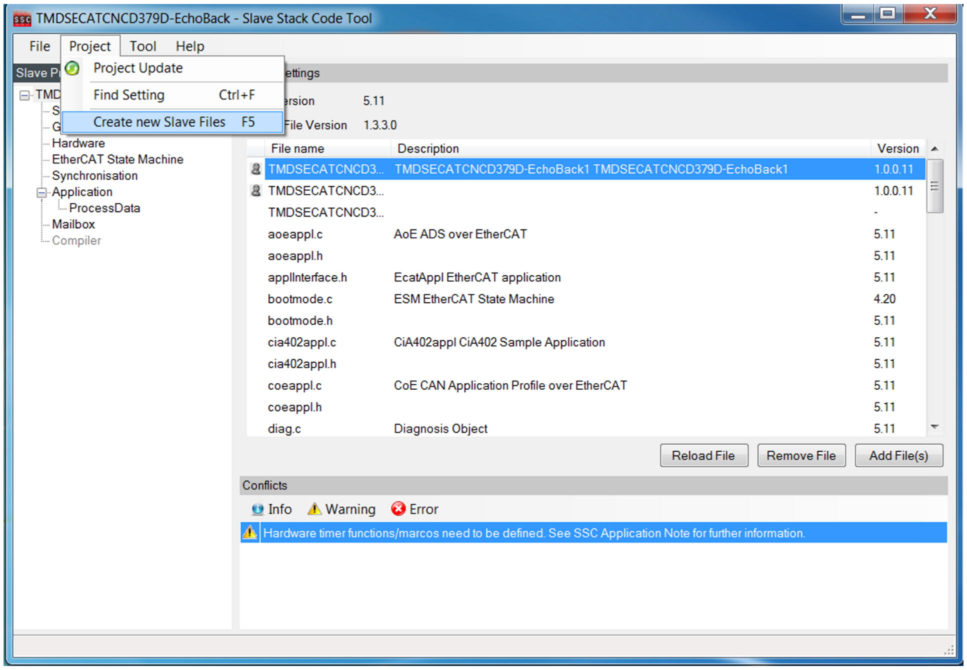

Figure 14. Create New Slave Files

8. Input the Source Folder and ESI File path, or check where it is being saved and click Start (see Figure 15).

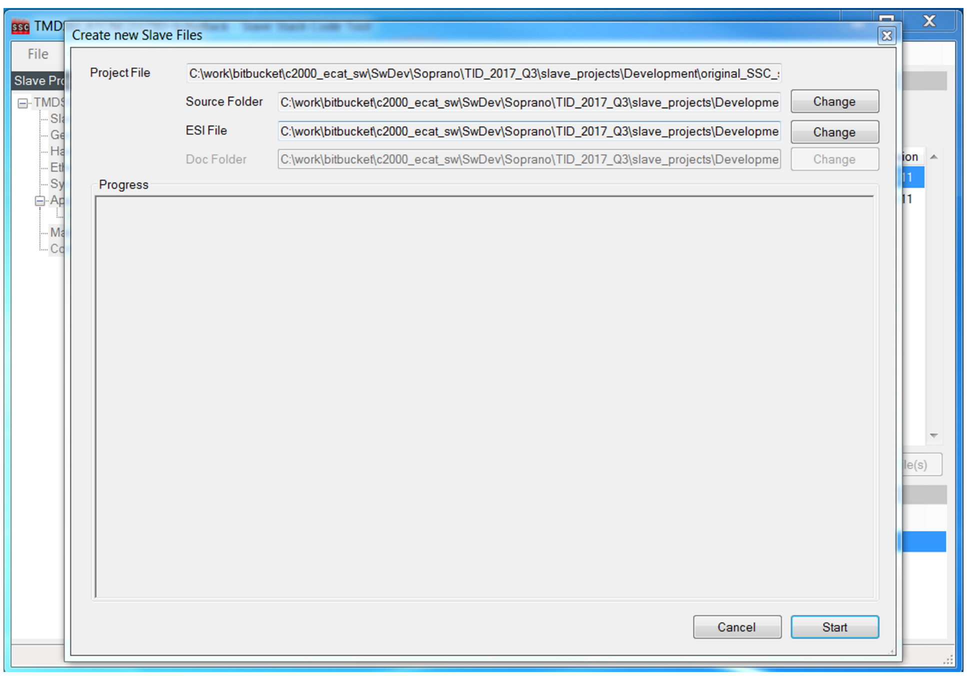

Figure 15. Project File Path Selection

9. The slave node source files must be created. Click OK and then close the screen (see Figure 16).

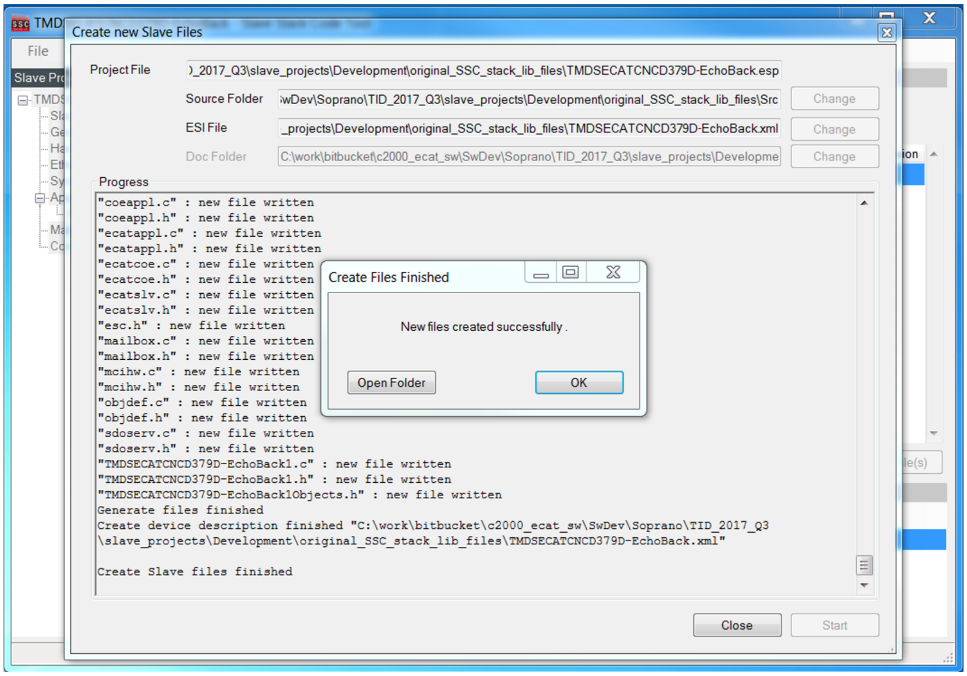

Figure 16. Slave File Creation

10. Inspect the directory in which the files were created (see Figure 17).

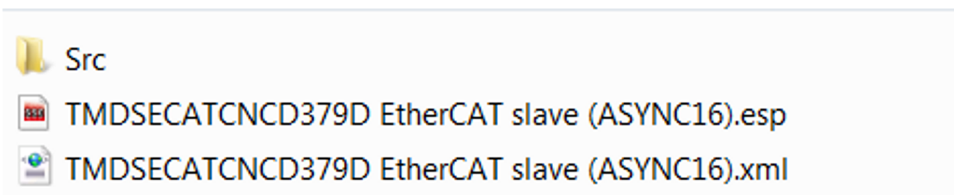

Figure 17. Slave Directory

- The src folder must contain all the slave stack files and the default sample Echoback application that were generated by the tool.
- The *.esp is the slave stack project file for the slave stack tool. Users can open this file in the SSC tool and edit the project as needed and regenerate the files.
- The *.xml is the generated ESI file which must be updated with the EtherCAT master in the network to which this slave node will be connected.

5 Moving SSC Tool-Generated Sources to the Solution Reference CCS Project

Figure 18 shows the EtherCAT slave stack sources generated from the steps in Section 4.2.

Figure 18. Slave Stack Sources

Users must copy all the files under the src folder (see Figure 18) to the EtherCAT Solution Reference CCS project, provided in Section 3.2, and shown in Figure 19.

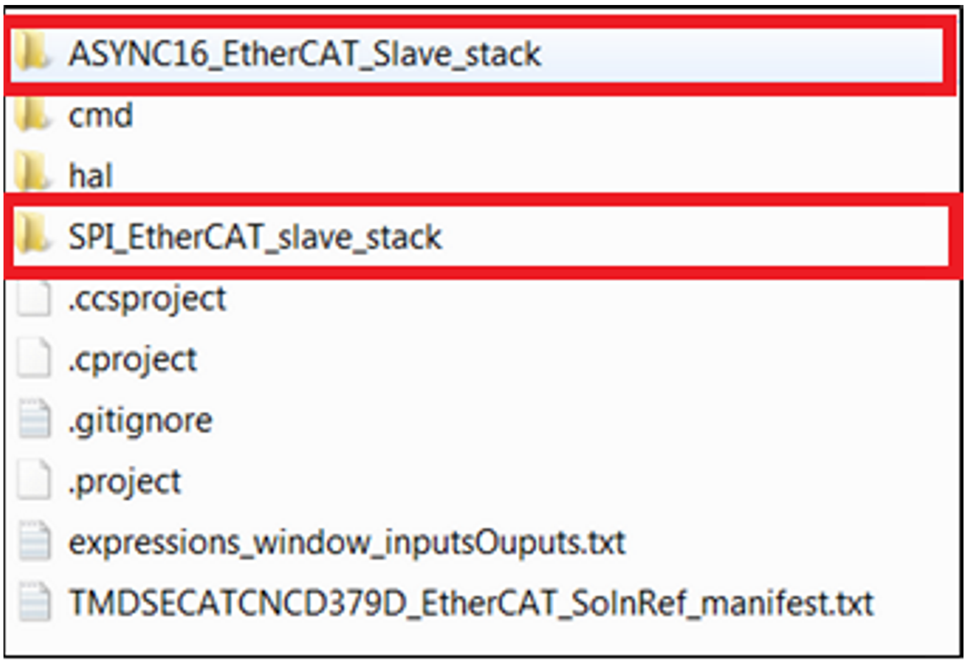

Figure 19. Copy Source Folder Files

If the user generated slave stack sources from the SSC tool for ASYNC16 PDI, then they must copy all the sources to the ASYNC16_EtherCAT_Slave_stack folder. If the user generated slave stack sources from the SSC tool for SPI PDI, then they must copy all the sources to the SPI_EtherCAT_slave_stack folder.

- ASYNC PDI sources → ASYNC16_EtherCAT_Slave_stack folder
- SPI PDI sources → SPI_EtherCAT_slave_stack folder

6 Download Code Composer Studio

Download the latest version of CCS from http://processors.wiki.ti.com/index.php/Download_CCS.

The latest version available at the time of writing this document is CCS v7.2.0.00013, so the steps listed are verified for this version of CCS.

7 Importing and Building CCS Solution Reference Project Using CCS GUI

Users must complete the steps in Section 4 and Section 5 to generate the EtherCAT and demo application sources, and to move the source files to the Solution Reference project.

1. Open CCS v7.
2. Import the project (see Figure 20).

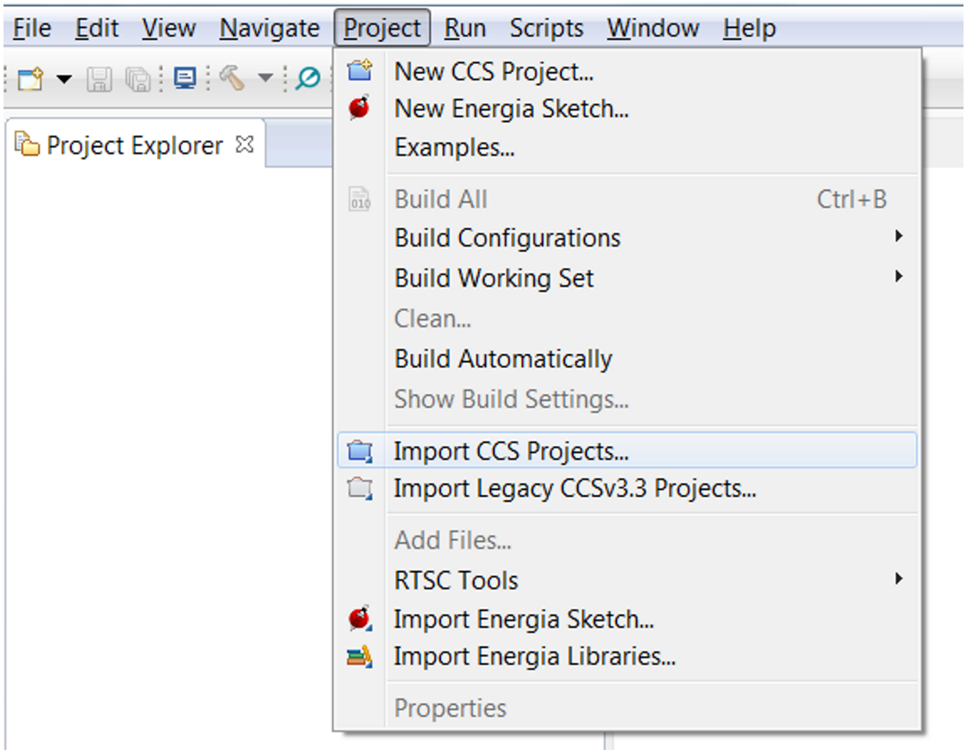

Figure 20. Import Project

3. Navigate to the TMDSECATCNCD379D EtherCAT solution reference project folder and check it (see Figure 21).

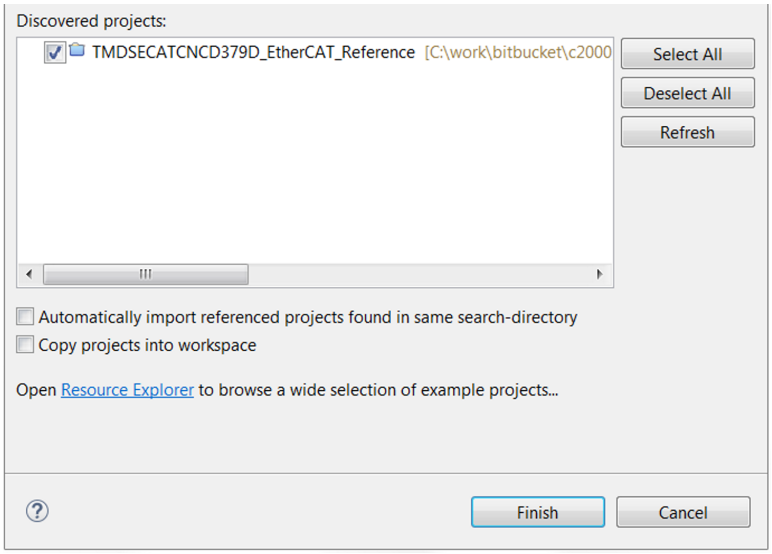

Figure 21. Discovered Projects

4. Click Finish, and the following screen appears (see Figure 22).

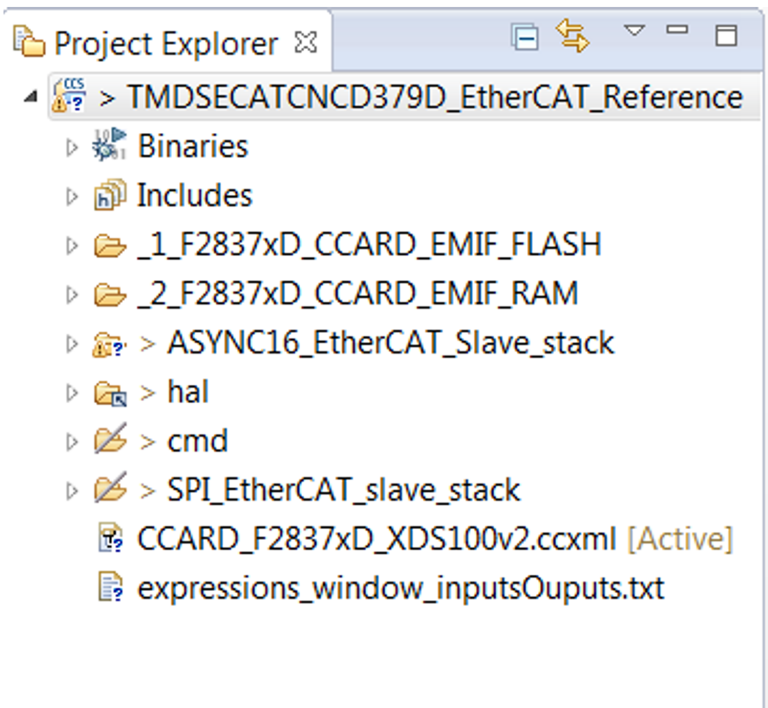

Figure 22. Project Folder

5. Right-click on the project and select the build configuration. Choose the build configuration matching the PDI configuration selected in SSC tool for C28x configuration and the SW1 switch setting on the TMDSECATCNCD379D kit (see Figure 23).

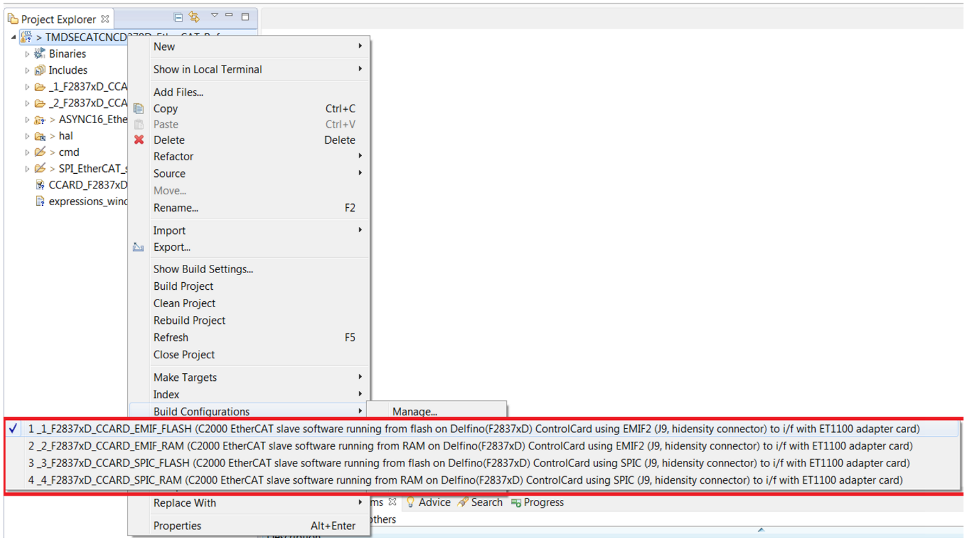

Figure 23. Select Build Configuration

Choose build configuration option 1 or 2 if ASYNC16 PDI is selected, and choose build configuration 3 or 4 if SPI PDI is chosen. To program the application in flash so it is retained through power cycle, choose the build option with *_FLASH. Depending on the build configuration chosen, the slave stack sources in the ASYNC16_EtherCAT_Slave_stack or SPI_EtherCAT_Slave_stack folder is selected automatically for build.

6. Rebuild or build project.
7. Power up the board using a Micro-USB cable or with a baseboard connected to 5-V supply.
8. Connect the Mini-USB cable for JTAG.
9. Launch target configurations by selecting the target configuration, as shown in Figure 24. The target configurations window is visible from view → Target Configurations option.

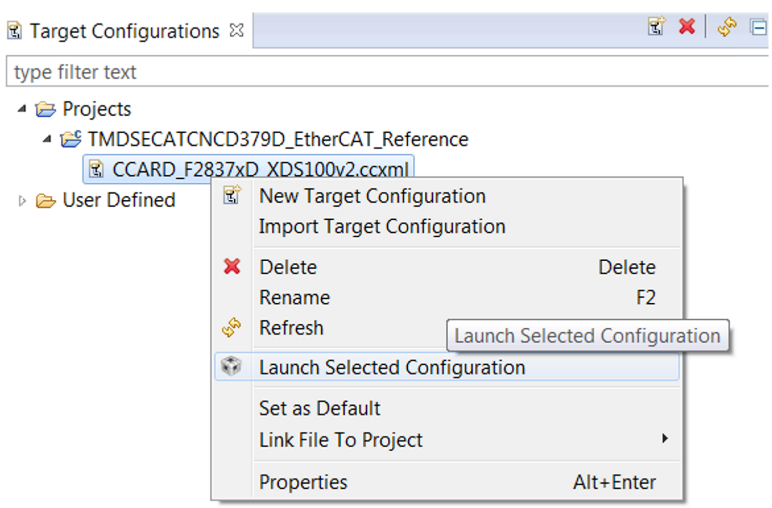

Figure 24. Launch Selected Configuration

10. Right-click and connect to CPU1 (see Figure 25).

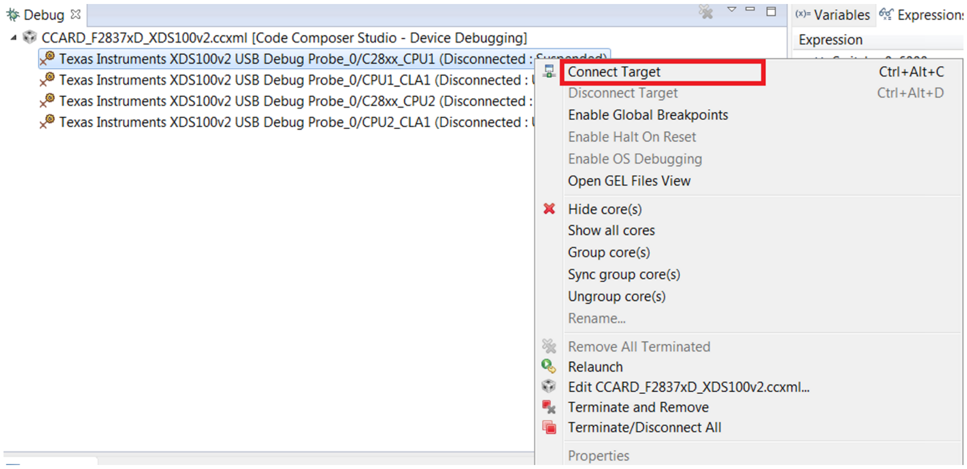

Figure 25. Connect to CPU1

11. Go to Run → Load Program → Browse project..., and select the OUT file just built and click OK. This step should load the program in flash if you built a FLASH build configuration (see Figure 26).

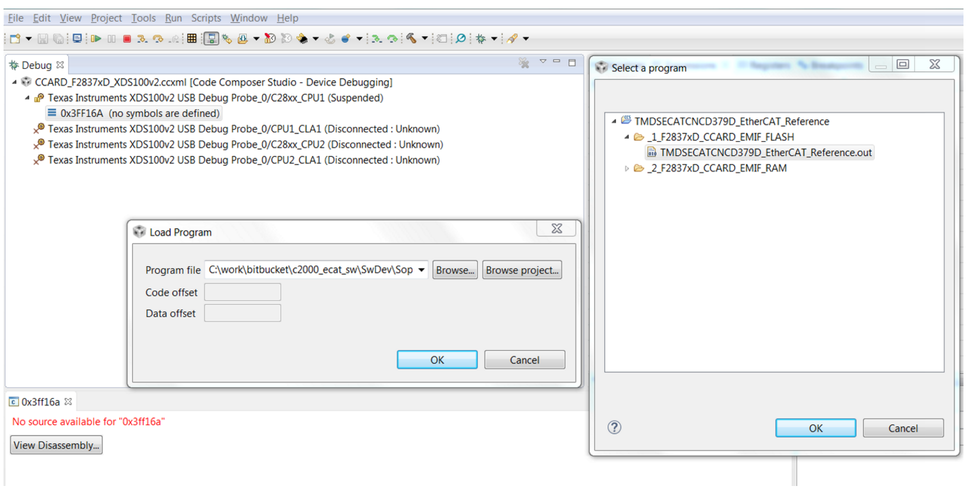

Figure 26. Load Program

12. Wait for the application to load. When it is loaded, the user can run by clicking the play (or F8) button.

8 Programming the ESI File from the EtherCAT Master

The following steps assume the user is using a TWINCAT master. ESI files are needed for the EtherCAT master to recognize the EtherCAT slaves connected in the network. An EtherCAT master builds an ENI file with the ESI files provided.

1. Copy the ESI files generated in Section 4.2, Step 10 to the TwinCAT ESI files folder → C:\TwinCAT\3.1\Config\Io\EtherCAT
   - TMDSECATCNCD379D EtherCAT slave (ASYNC16).xml for ASYNC16 PDI
   - TMDSECATCNCD379D EtherCAT slave (SPI).xml for SPI PDI
2. Connect the TMDSECATCNCD379D EtherCAT kit Port 0/IN to the TwinCAT master.
3. Scan for slaves.
4. Program the EEPROM with the ESI files, as shown in Figure 27. Choose the appropriate ESI file according to your PDI selection.

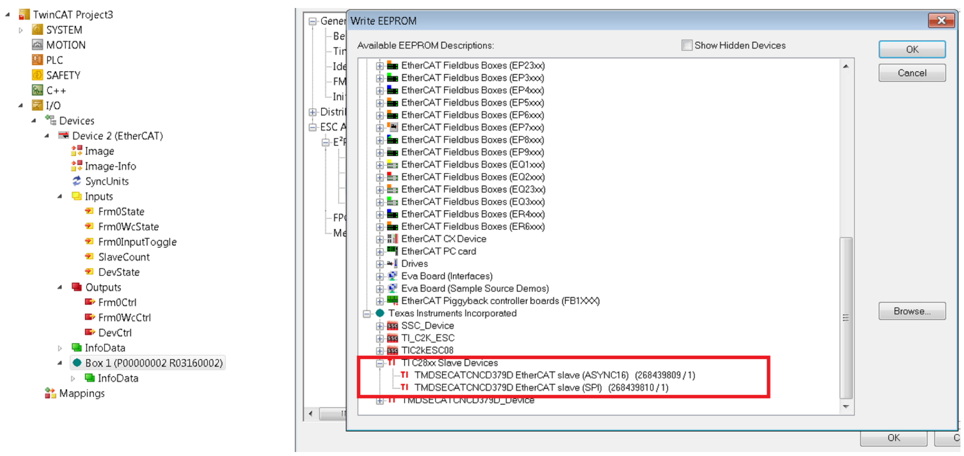

Figure 27. Program EEPROM

5. Rescan the network (see Figure 28).

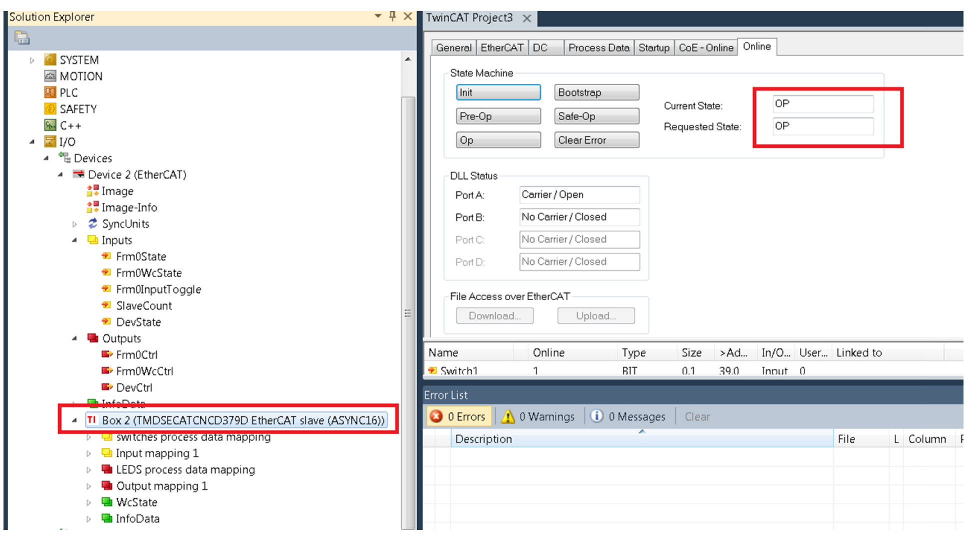

Figure 28. Rescan Network

6. Set the outputs and read the inputs.

The TMDSECATCNCD379D sample Echoback application demo software loops outputs back to inputs. Table 4 lists the input variables.

Table 4. Echoback Application Inputs

Index	Object Code	Sub Index	Data Type	Name	Notes
0x6000	RECORD			Switches	
		0x01	BOOL	Switch1	Switch1 input to master
		0x02	BOOL	Switch2	Switch2 input to master
		0x03	BOOL	Switch3	Switch3 input to master
		0x04	BOOL	Switch4	Switch4 input to master
		0x05	BOOL	Switch5	Switch5 input to master
		0x06	BOOL	Switch6	Switch6 input to master
		0x07	BOOL	Switch7	Switch7 input to master
		0x08	BOOL	Switch8	Switch8 input to master
0x6010	RECORD			DataToMaster	
		0x01	UNSIGNED32	DataToMaster	32-bit data input to master
0x6012	RECORD			TargetModeResponse	
		0x01	UNSIGNED16	ModeResponse	16-bit command mode (0 = position, 1 = speed), feedback to master
0x6014	RECORD			TargetSpeedPosFeedback	
		0x01	UNSIGNED32	SpeedPosFbk	32-bit speed, position feedback to master

Table 5 lists the output variables.

Table 5. Echoback Application Outputs

Index	Object Code	Sub Index	Data Type	Name	Notes
0x7000				LEDs	
		0x01	BOOL	LED1	LED1 output from master
		0x02	BOOL	LED2	LED2 output from master
		0x03	BOOL	LED3	LED3 output from master
		0x04	BOOL	LED4	LED4 output from master
		0x05	BOOL	LED5	LED5 output from master
		0x06	BOOL	LED6	LED6 output from master
		0x07	BOOL	LED7	LED7 output from master
		0x08	BOOL	LED8	LED8 output from master
0x7010	RECORD			DatafromMaster	
		0x01	UNSIGNED32	DatafromMaster	32-bit output var from master
0x7012	RECORD			TargetMode	
		0x01	UNSIGNED16	Mode	16-bit command mode (0 = position, 1 = speed), request from master
0x7014	RECORD			TargetSpeedPosReq	
		0x01	UNSIGNED32	SpeedPosReq	32-bit speed and position request from master

Table 6 lists the loopback operation of sample EchoBack application. The user can set the outputs as follows in Table 6, and observe the value reflected in the respective inputs.

Table 6. Loopback of Outputs and Inputs

Output Value Set by Master	Input Value Looped Back by Slave
LED1	SWITCH1
LED2	SWITCH2
LED3	SWITCH3
LED4	SWITCH4
LED5	SWITCH5
LED6	SWITCH6
LED7	SWITCH7
LED8	SWITCH8
DatafromMaster	DataToMaster
Mode	ModeResponse
SpeedPosReq	SpeedPosFbk

9 References

1. EtherCAT, EtherCAT Slave Implementation Guide
2. Beckhoff, EtherCAT Slave Stack Design Guide
3. Beckhoff, EtherCAT Slave Stack Code – ET9300 from ETG/Beckhoff
4. Beckhoff, ET9300 EtherCAT Slave Stack Code, application note
5. Beckhoff, ET1100, data sheet
6. Texas Instruments, Delfino EtherCAT, TI Design
7. Beckhoff, Beckhoff ET1100 EtherCAT Configuration and Pinout Tool
8. Texas Instruments, F28379D Data Sheet and Technical Reference Manual
9. Texas Instruments, TMS320F2837xD Dual-Core Delfino Microcontrollers, TRM
10. Texas Instruments, TMS320F28377D, product page
11. Texas Instruments, Debug Server Scripting, Wiki page
12. Texas Instruments, CCS Command Line Project Create, Import, and Build Steps, Wiki page

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).