## TI Designs EtherCAT® Interface for High Performance C2000<sup>™</sup> MCU

# U Texas Instruments

## Description

This reference design demonstrates how to connect an EtherCAT ET1100 Slave Controller to a C2000 Delfino<sup>™</sup> MCU. The interface supports both demultiplexed address/data busses for maximum bandwidth and minimum latency, and a serial peripheral interface (SPI) mode for low pin-count EtherCAT communication. The Slave Controller offloads the processing of 100 Mbps Ethernet-based fieldbus communication, thereby, eliminating CPU overhead for these tasks.

## Resources

http://www.ti.com/tool/TID M-DELFINO-ETHERCAT TMS320F28377D DP83822 LAUNCHXL-F28377S

Design Folder Product Folder Product Folder Product Folder



## Features

- High-Performance Real-Time Control MCU Paired
   With Low-Latency Ethernet-Based Communication
- High-Bandwidth, Low-Latency Interface to Beckhoff ET1100 EtherCAT Slave Controller
- Supports Both Asynchronous Parallel and SPI Connections
- Glueless Interface
- Eliminates CPU Overhead for EtherCAT Frame Processing

### Applications

- Industrial Drives
- Servo Motor Drives
- Manufacturing Robotics
- CNC Machinery
- Remote I/O





BoosterPack (SPI) Format



Ccard DaughterCard (SPI/EMIF) Format





An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

#### 1 System Overview

#### 1.1 System Description

The TIDM-DELFINO-ETHERCAT TI design describes a pair of plug-in modules that use a Beckhoff ET1100 EtherCAT Slave Controller (ESC) and TI Ethernet PHYs to enable the creation of EtherCAT slave nodes, when coupled with a C2000 MCU. The modules are available in both daughtercard format (for TI F2837x ControlCARDs) and in BoosterPack<sup>™</sup> format (for C2000 LaunchPad development kits). The daughtercard format uses a 60-pin high-density connector that can support both asynchronous parallel and SPI interfaces, while the BoosterPack format utilizes a standard 40-pin header and supports a SPI interface only.

This TI design illustrates how to set up the EtherCAT BoosterPack and daughtercard modules, initialize the ET1100 subsystem for first use, and install and configure the Beckhoff TwinCAT 3 software for use as an EtherCAT master in a test setup. Example code is provided to configure both SPI and EMIF interfaces and run simple read/write tests across an EtherCAT network.



Figure 1. TIDM-DELFINO-ETHERCAT BoosterPack Plug-in Module and C2000 Launchpad

#### 1.2 Key System Specifications

PARAMETER	DESCRIPTION	VALUE	DELAY (1)
ТА	Read-to-Write Turnaround time	1	5 ns
RHOLD	Address and CSn hold after OEn LH edge	1	5 ns
RSTROBE	Read Strobe time in units of EMIF Clocks	64	320 ns
RSETUP	Address and CSn to OEn assertion delay	1	5 ns
WHOLD	Write Hold time after WE deassertion	1	5 ns
WSTROBE	Write Strobe (WE) width	2	10 ns
WSETUP	Address and CSn setup time to WE assertion	1	5 ns
EW	Extended Wait Mode	ENABLE	-
SS	Strobe Select Mode	DISABLE	-

Table 1.	EMIF	Configuration	and	Timina	Settinas	for	ET1100	PDI







## 1.3 Block Diagram

Figure 4 shows the block diagram of the SPI (for the C2000 LaunchPad) and EMIF (for the C2000 ControlCARD) interfaces. In both configurations, the 'F2837x runs the EtherCAT slave stack while the ET1100 is used to offload the EtherCAT Slave Controller (ESC) frame processing, FMMU, and SyncManager operations.

Certain pins on the ET1100 are used on both the SPI and asynchronous interfaces, which requires the '74CBTLV3257 mux and '74CBTLV3245 buffer to steer these signals to the appropriate GPIO on the 'F2837x MCU.

The EtherCAT BoosterPack and Add-on cards feature an on-board DC-DC 5V–3.3 V converter, which allows the 3.3 V Vcc to be sourced from either the C2000 LaunchPad/ControlCARD, or generated locally from an off-board 5V source.



Figure 3. Block Diagram





Figure 4. F2837x SPI and EMIF Connections to the EtherCAT Slave Controller

## 1.4 Highlighted Products

### 1.4.1 TMS320F28377D

The Delfino TMS320F2837x is a powerful 32-bit floating-point microcontroller unit (MCU) designed for advanced closed-loop control applications such as industrial drives and servo motor control; solar inverters and converters; digital power; transportation; and power line communications. Complete development packages for digital power and industrial drives are available as part of the powerSUITE and DesignDRIVE initiatives. The F2837x supports a new dual-core C28x architecture that significantly boosts system performance while integrated analog and control peripherals allow designers to consolidate control architectures and eliminate multiprocessor use in high-end systems.

In the TIDM-DELFINO-ETHERCAT design, the F2837x receives EtherCAT data from the ET1100 through either a serial (SPI) interface or an asynchronous parallel memory interface (EMIF). Figure 5 shows the use of EMIF2, but either EMIF can be used to interface to the ET1100. Note that GPIO93,94 are for future expansion to address a larger memory space. They are not used in the example code.





Figure 5. TMS320F2837x EMIF Interface to ET1100 EtherCAT Slave Controller



Figure 6. Delfino SPI as EtherCAT PDI

## 1.4.2 DP83822

The DP83822 is a low power single-port 10/100 Mbps Ethernet PHY. It provides all physical layer functions needed to transmit and receive data over both standard twisted-pair cables or connect to an external fiber optic transceiver. Additionally, the DP83822 provides flexibility to connect to a MAC through a standard MII, RMII or RGMII interface.

#### 1.4.3 **TPS62063**

The TPS6206x is a family of highly efficient synchronous step-down DC-DC converters. They provide up to 1.6-A output current. With an input voltage range of 2.7 V to 6 V, the device is a perfect fit for power conversion from 5-V or 3.3-V system supply rails. The TPS6206x operates at 3-MHz fixed frequency and enters power save mode operation at light load currents to maintain high efficiency over the entire load current range. The power save mode is optimized for low-output voltage ripple. For low noise applications, the device can be forced into fixed frequency PWM mode by pulling the MODE pin high.

In this TI Design, the converter enables power to be supplied from either the C2000 LaunchPad, controlCARD, or from an external 5 V source.



System Overview

#### 1.4.4 **SN74LVC1G07**

This single buffer/driver is designed for 1.65-V to 5.5-V  $V_{cc}$  operation. The output of the SN74LVC1G07 device is open drain and can be connected to other open-drain outputs to implement active-low wired-OR or active-high wired-AND functions. The maximum sink current is 32 mA.

In this design, the open-drain buffer connects a GPIO from the F2837x LaunchPad or controlCARD to the ET1100 EtherCAT slave controller reset pin, thereby, enabling independent reset of both devices.

#### 1.4.5 SN74CBTLV3245A

The SN74CBTLV3245A provides eight bits of high-speed bus switching in a standard '245 device pinout. The low on-state resistance of the switch allows connections to be made with minimal propagation delay. The device is organized as one 8-bit switch. When output enable (OE) is low, the 8-bit bus switch is on, and port A is connected to port B. When OE is high, the switch is open and the high-impedance state exists between the two ports.

In this TI Design, the bus switch isolates selected ET1100 outputs from the F2837x controlCARD GPIOs when the EMIF interface is not used (SPI mode). Note that the bus switches are optional in a design using only a single interface type.

#### 1.4.6 SN74CBTLV3257

The SN74CBTLV3257 device is a 4-bit 1-of-2 high-speed FET multiplexer/demultiplexer. The low on-state resistance of the switch allows connections to be made with minimal propagation delay. The select (S) input controls the data flow. The FET multiplexers/demultiplexers are disabled when the output-enable (OE) input is high.

This multiplexer connects selected GPIOs to the ET1100 depending on the interface mode (EMIF or SPI). Note that the multiplexer is optional in a design using only a single interface type.

#### 1.4.6.1 Beckhoff ET1100 EtherCAT Slave Controller (ESC)

The ET1100 device is an EtherCAT Slave Controller (ESC). It handles all communications between the EtherCAT fieldbus and the F2837x interface (either SPI or EMIF).

For more information, see http://www.beckhoff.com.





1.4.7 TMS320F2837x Delfino Microcontroller Functional Diagram

Figure 7. TMS320F2837x Functional Diagram



#### 1.4.8 Ethernet PHY Functional Diagram



Figure 8. DP83822 Functional Diagram

#### 1.4.9 SN74CBTLV3245A Functional Diagram



Figure 9. SN74CBTLV3245A Functional Diagram







Figure 10. SN74CBTLV3257 Functional Diagram

## 1.4.11 ET1100 Functional Diagram

Bechoff data sheet: http://download.beckhoff.com/download/document/io/ethercat-development-products/ethercat\_et1100\_datasheet\_v1i9.pdf



#### Getting Started Hardware

#### 2 Getting Started Hardware

This TI-Design describes two configurations for the EtherCAT plug-in: a BoosterPack form factor compatible with TI LaunchPad Development kit and a F2837x ControlCARD daughtercard. The hardware setup described below is identical for both configurations in terms of EtherCAT connectivity and slave operation. The available Process Data Interfaces (PDIs) differ between the two boards, in that the LaunchPad has only enough pins for a SPI connection, whereas, the controlCARD plugin can support either an EMIF or SPI interface.



# Figure 11. Stacked F2837x LaunchPad and EtherCAT BoosterPack (Left) and ControlCard + EtherCAT Plugin (Right)



Figure 12. EtherCAT Test Setup

## 2.1 EtherCAT Master Configuration Using TwinCAT3

## 2.1.1 Download and Install TwinCAT3 From the Beckhoff website

The TwinCAT3 software is available from the Beckhoff website at http://www.beckhoff.com. Follow the left sidebar to Download  $\rightarrow$  Software  $\rightarrow$  TwinCAT 3  $\rightarrow$  TE1xxx | Engineering. As of this writing, the most recent version of this software is TwinCAT 3.1 – eXtended Automation Engineering (XAE) v 3.1.4018.26.



Getting Started Hardware

#### 2.1.2 Verify the TwinCAT Runtime is Active

 Check for the EtherCAT icon in the notification panel in the lower-right corner as shown in Figure 13. If this is absent, open the notification panel and check in the popup window for the TCSwitchRuntime. Right click on this icon and select Tools → TCSwitchRuntime.





2. Verify that the TCSwitch Runtime is active. The "Deactivate" button should be showing as illustrated in Figure 14. If this button reads "Activate", click that button to start the TCSwitchRuntime.

TcSwitchRuntime	×
	Version 1.11
TwinCAT 3.1 Build 4018 is active	
	-
Switch Deactivate	)
Save Log Clear Log	
MDPUA-Server Service does not exist.	*
Starting TF3300 Scope Server TF3300 Scope Server service enabled successfully. Service start pending TF3300 Scope Server started successfully.	
Starting TcAdsSerialCommServer TcAdsSerialCommServer Service does not exist.	
Starting TcAdsWcfHost Service start type is Manual and hence could not be started. **DONE**	-

Figure 14. TwinCAT Runtime Dialog Box

3. If the TCSwitchRuntime is not found in step (a,b) above, then locate the runtime in the file system. A typical location is: "c:\TwinCAT\TcSwitchRuntime\TcSwitchRuntime.exe". Note that it is NOT commonly found in the Start Menu.

## 2.2 Start TwinCAT3 and Verify That TwinCAT is Running in Visual Studio

- 1. Locate the TwinCAT XAE, which can be found in one of three places:
  - (a) Start menu  $\rightarrow$  under Beckhoff  $\rightarrow$  TwinCAT3  $\rightarrow$  TwinCAT XAE (VS 2010).
  - (b) Desktop icon is shown in Figure 15.



## Figure 15. TwinCAT3 XAE Desktop Icon

(c) Notification panel icon  $\rightarrow$  right click and select TwinCAT XAE (VS 2010).



Figure 16. TwinCAT3 Icon in Toolbar

 Verify that TwinCAT is running under Visual Studio. "TwinCAT" and "PLC" should both appear in the main toolbar as shown in 1. If these menu items are not shown, then the TC3 runtime is NOT running. Go back to step Figure 17 to restart the TC3 runtime.



Figure 17. Visual Studio Menus for TwinCAT3

- 3. Open a new EtherCAT project.
  - (a) File  $\rightarrow$  New  $\rightarrow$  Project.

New Project					
Recent Templates		.NET Framework 4	Sort by: Default		Search Installed Templates
Installed Templates  D Other Project Types TwinCAT Measurem TwinCAT Project  Online Templates	s nent	TwinCAT XA	E Project (XML format)	TwinCAT Project	Type: TwinCAT Project TwinCAT XAE SystemManager Configuration
Name:	TwinCAT Project				
Location:	C:\Users\xxxx	< x\Documents\Visual S	tudio 2010\Projects	•	Browse
Solution name:	TwinCAT Project				Create directory for solution
					OK Cancel

Figure 18. TwinCAT3 New Project Dialog



- 4. Verify that a Realtime Ethernet Adapter is installed.
  - (a) TwinCAT → Show Realtime Ethernet Compatible Devices If no RT adapter is installed, select one from the list of Compatible devices and click "Install", then exit this popup.

hernet Adapters	Update List
Set Installed and ready to use devices(realtime capable) Local Area Connection - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)	Install
Installed and ready to use devices(for demo use only) Compatible devices	Update
Incompatible devices Wireless Network Connection - DW1520 Wireless N WLAN Half-Mini Card Disabled devices	Bind
	Unbind
	Enable
	Disable

Figure 19. TwinCAT3 EtherNet Adapter Dialog

- 5. Scan for the newly installed Realtime adapter by clicking TwinCAT  $\rightarrow$  Scan.
  - (a) A popup indicating TwinCAT has found the adapter that should appear (see Figure 20).

• TwinCAT Project - Microsoft Visu	l Studio	
File Edit View Project Build	Debug TaleCAT PLC Tools Surpe Window Help	
0-2-3-44 × 4	23 10 - Ch - St - Cal P. Release + TwinCAT RT (r54) (	Status + 100 9 10 14
19 G L .!!	三日 ・ 日本は いっつ 日日 ひつ・1	G G C C G G G Latords
iolution Explorer X Toolbox		Projection
3		
Solution 'TwirtCAT Project' (1 pr	sect)	· 2:84
a fwinCAT Project	Trees Without Stand	
III MOTION		
III INC.	V Device 2 (OtherCAT) Local Area Connection (Intel)	K1857/LM Cepter Net
III SAVETY		Concel
· III 50		
<ul> <li>To Devices</li> </ul>		Select Ail
<ul> <li>Im Device 2 (DtherCAS)</li> <li>Image</li> </ul>		Unantest All
of the second second		

Figure 20. TwinCAT3 Discovery of Ethernet Adapter

(b) Click "OK", then click "Yes" to the following two popups (see Figure 21).

Microsoft Visual Studio	Microsoft Visual Studio
Scan for boxes	Activate Free Run
Yes No	Yes No

Figure 21. TwinCAT3 Master Scan for Slaves and Free Run Activation



(c) The screen view shown in Figure 22 should now be visible in TwinCAT3, indicating that the Master and Slave are connected and prepared for use. Note that "Box 1 (TI\_C2kESC)" appears at the bottom of the image, indicating that TwinCAT has discovered the EtherCAT slave. If this is the first time that the slave has been connected and the EEPROM has not yet been programmed, the "Box n ()" label (for example, "TI\_C2KESC") will not be visible.

TwinCAT Project - Microsoft Visual Studio							
File Edit View Project Build Debug Twind	CAT PLC Tools Scope Windo	w Help					
- (*) - (*) 🛃 😹 📓 - (*) - (*)	🛛 - 🚚 - 🖳 🕨 Release	TwinCAT RT (x	64) 🔹	🖄 Status	- 🛛 🖓 😤 🐋 🎉	e 🎫 🔐 💿 👻 🖕	
	전 🕨 🛋 🔁 🗐 🗐 🗐 👘 🖷 전		3 5 - I 😰 🕻	1 🗖 🗢 🔨 🎯 🚱	<local></local>	• •	
Solution Explorer X Toolbox					<ul> <li>Properties</li> </ul>		<b>-</b> ↓ ×
Ph					Device 2 (EtherCAT) Eth	erCAT Master	
					21 41 122		
Solution 'TwinCAT Project' (1 project)					4 Misc		
<ul> <li>TwinCAT Project</li> </ul>					(Name)	Device 2 (EtherCAT)	
> III SYSTEM					Disabled	SMDS NOT DISABLED	
MOTION					ItemType	2	
I PLC					PathName	TIID^Device 2 (EtherCAT)	
SAFETY					Persistent		
S C++					SaveInOwnFile	False	
4 21/O							
The Devices							
Device 2 (EtherCAT)							
Image							
s image-into							
Syncomics							
Poy 1 (TL C2KESC)					Misc		
Mannings							
- Mappings							
Error List				Output			
🔇 0 Errors 🛛 🗥 0 Warnings 🛛 🕕 0 Messages	Clear			Show output from:	winCAT	-   \$   \$ B	🕵 🖃
Description	File	Line Colu	Project				*
							*
				4			•
Creating project 'TwinCAT Project' project creation	successful						
Stating project wind it hoject project creation							



6. The EtherCAT Master is now ready for communication with the Slave device.



## 2.3 Configuring the EtherCAT BoosterPack Interface for EMIF or SPI Operation

The EtherCAT BoosterPack board must be configured prior to powerup to select between the EMIF or SPI interface from the C28x to the ET1100 slave controller. Options for power source and interface type are available through jumpers and/or DIP switches as described in Figure 23 and Table 2.



Figure 23. EtherCAT BoosterPack LED and Switch Locations (Switch and LED locations are identical for the ControlCard plugin version)

Table 2. EtherCAT	LED and Switch	Usage Descriptions

NAME	OPTIONS	DESCRIPTION
Switches/Jumpers		
SW1	L – SPI R – EMIF	Selects between EMIF and SPI Interface Modes
J3	1-2 off-board 2-3 on-board	Off-board: 3.3 V is provided directly from attached LaunchPad or ControlCARD On-board: 3.3 V is generated by the on- board regulator from a separate 5 V supply.
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. For details, see the schematics in Figure 38.

 Additional RUN LED states provided in the ET1100 Digital I/O Signals table of the ET1100 data sheet, which is located at: http://download.beckhoff.com/download/document/io/ethercat-development-products/ethercat\_et1100\_datasheet\_v1i9.pdf.



Getting Started Hardware

## 2.4 Preparing the Addon Board for EtherCAT Communcation Using TwinCAT3

After installing the TwinCAT3 software in the previous section and verifying connectivity between the EtherCAT Master and the slave node, the ESC EEPROM must be programmed to enable communication to the C28x device over either the EMIF or SPI PDI. The following procedure writes the binary file associated with the selected PDI type into the EEPROM:

1. After starting up the TwinCAT software in Section 2.1, the screen should look something like Figure 24. Double clicking the EtherCAT slave (labelled "Box 1" in the figure) brings up the EtherCAT properties window on the right. Click "Advanced Settings".

StrainCAT Project - Microsoft Visual	Studio								X
File Edit View Project Build D	ebug 1	winCAT PLC To	ols Scope	Window	/ Help				
i 🛅 • 🖮 • 对 🖌 🥥 👗 🕯	3 9	· (* - <b>5</b> - <b>5</b>	▶ Release	se 🔻	TwinCA	AT RT (	(x64) 🔹 🔯 Status		-
		-] ▶ ■ -[	⊊⊒ (⊒ ª		à 🖆	**	55-112 0 0 2	🔨 🞯 🚱 🛬 掌	
Solution Explorer X Toolbox	-	TwinCAT Project				_		-	Ч×
<b>P</b>		General EtherCA	T DC Pr	ocess Data	Startup	CoE - 0	Online Online		-
<ul> <li>Solution 'TwinCAT Project' (1 project')</li> <li>TwinCAT Project</li> <li>SYSTEM</li> <li>MOTION</li> <li>PLC</li> <li>SAFETY</li> <li>C++</li> <li>I/O</li> <li>* Devices</li> <li>* Devices</li> <li>* Devices 2 (EtherCAT)</li> <li>* Image</li> <li>* Image</li> </ul>	ect)	Type: Product/Revisior Auto Inc Addr: EtherCAT Addr: Identification Va Previous Port:	TI_C2K 141407 0 1001 Ue: 0 Master	20274 / 18			Advanced Settings		
🤌 🥏 SyncUnits		Name	Online	Туре	Size	>A	In/ Us Linked to		
> Inputs		<sup>≁</sup> Switch 1	0	BIT	0.1	39.0	Inp 0		III
<ul> <li>InfoData</li> </ul>		Switch 2	0	BIT	0.1	39.1	Inp 0		
<ul> <li>Box 1 (TI_C2KESC)</li> </ul>	-	Switch 3	0	BIT	0.1	39.2	Inp 0		-
AL		IwinCAT Proje	ect Pro	perties					
Error List			•	Ψ× Ο	utput				Ψ×
0 Errors 🔒 0 Warnings	0 Messa	ges Clear							·
Description	F <mark>i</mark> le	Line Colu	Project						*

Figure 24. TwinCAT Project EtherCAT Tab



 In the advanced settings popup windows, select "Smart View" to bring up detailed information about the EtherCAT Slave PDI. Click "Write E2PROM". Note that, prior to initialization, this view shows a "blank" configuration.

General	Smart View						
- Behavior	Config Data (evaluated f	rom ESC)	Device Identity (hex)				
- Timeout Settin	E <sup>2</sup> PROM Size (Byte):	128 👻	Vendor Id:	0x000000	0x00000000		
-FMMU / SM	PDI Type:	0	Product Code:	0x000000	00		
-Init Command	Device Emulation	(state machine emulation)	Revison No.:	0x000000	00		
Here Mailbox	SPI / 8 / 16 µC Interface		Serial No.:	0x000000	00		
ESC Access	BUSY Open Dra	in BUSY High Active	Product Revision:				
Configured	22 Dit Interface		Mailbox				
- Enhanced Li - Smart View - Hex Editor - FPGA	WD Open Drain	WD High Active	CoE SoE	EoE	FoE		
	input cutori		Bootstrap Configuration	on			
Memory	Sync Signal Configura	tion	Out Start/Length:	0	0		
	SYNC0 Open Dr.	ain SYNC0 High Active	In Start/Length:	0	0		
	SYNC1 Open Drain SYNC1 High Active	ain SYNC1 High Active	Standard Configuration				
	SYNC1 Enabled	SYNC1 to PDI IRQ	Out Start/Length:	0	0		
	Impulse Length (µs):	0	In Start/Longth:	0	0		
			in Start/Length:	0	U		
	Write E <sup>2</sup> PROM	Read E <sup>2</sup> PROM					

Figure 25. TwinCAT "Smart View" Slave Properties Window Showing Blank Slave

- 3. Click "Browse" to find the desired EEPROM binary associated with either the SPI or EMIF interface in the sw/master\_files directory downloaded from the project site.
  - (a) pdi\_test\_app\_spi.bin for SPI interface
  - (b) pdi\_test\_app\_emif.bin for EMIF interface
  - (c) Click "OK".

ilable EEPROM Descriptions:	Show Hidden Devices	ОК
👹 Beckhoff Automation GmbH & Co. KG		
Vendor with ID 0000059Dh		Cancel
TI C2K SPI ESlaves		
TI_C2KESC (1414070274 / 18)		
		Browco
		Diowse

Figure 26. TwinCAT3 EEPROM File Dialog



#### Getting Started Firmware

www.ti.com

4. After TwinCAT writes the EEPROM (in this case for the SPI), the following "Smart View" window showing a valid configuration with PDI Type = "SPI slave" should appear (see Figure 27).

Advanced Settings					<b>X</b>
General Mailbox Distributed Clock ESC Access ESC Access Configured - Enhanced Li - Smart View Hex Editor - FPGA Memory	Smart View Config Data (evaluated fm E²PROM Size (Byte): PDI Type: Device Emulation SPI / 8 / 16 µC Interface ✓ BUSY Open Drain 32 Bit Interface ✓ WD Open Drain Input Latch	om ESC) 2048 SPI slave (state machine emulation) e M BUSY High Active INT High Active WD High Active	Device Identity (hex) Vendor Id: Product Code: Revison No.: Serial No.: Product Revision: Mailbox Mailbox CoE SoE AoE	0x000005 0x544900 0x000000 0x020000	9D 02 112 00
	Sync Signal Configural SYNC0 Open Dra ✓ SYNC0 Enabled SYNC1 Open Dra ✓ SYNC1 Enabled Impulse Length (µs):	ion SYNC0 High Active SYNC0 to PDI IRQ SYNC1 High Active SYNC1 to PDI IRQ 0.100	Bootstrap Configuration Out Start/Length: In Start/Length: Standard Configuration Out Start/Length: In Start/Length:	4096 5120 4096 5120	128 128 128 128
4 111 4	Write E <sup>2</sup> PROM	Read E <sup>2</sup> PROM			OK Cancel

## Figure 27. TwinCat "Smart View" Slave Properties Window Showing Programmed EEPROM

5. The EtherCAT Slave controller is now prepared for communication to the MCU.

## 3 Getting Started Firmware

A software package (project pdi\_hal\_test\_app) is included with this TI Design that contains a Code Composer Studio<sup>™</sup> (CCS) project designed for the 'F2837x. The project should be imported into CCS v6.0 or later for building and uploading to the target board. Six build configurations are provided to enable use of RAM or Flash-based designs with two board types (ControlCARD and LaunchPad) and two PDI types (EMIF and SPI).

- **NOTE:** For EtherCat Developers: The EtherCAT Technology Group (ETG) recommends membership for parties implementing EtherCAT in a machine or machine line. For additional information about EtherCAT and ETG membership, see the following links:
  - EtherCAT FAQs: https://www.ethercat.org/en/faq.html
  - EtherCAT Technology Group site: https://www.ethercat.org



## 3.1 PDI Test Project CCS Setup (project import)

The PDI test application contains a Hardware Abstraction Layer (HAL) that handles all of the configuration details for the EMIF or SPI interface. This CCS6.x project also performs some simple Read/Write tests to verify correct functioning of communication with the ET1100 Slave controller.

The CCS project files for pdi\_hal\_test\_app can be downloaded from the design directory at http://www.ti.com/tool/TIDM-DELFINO-ETHERCAT. Once downloaded, import this project into CCS 6.0 or later using the procedure below:

- 1. File  $\rightarrow$  Import... (an "Import" popup window appears).
- 2. Select "Code Composer Studio"  $\rightarrow$  CCS Projects. Click "Next".
- 3. Use "Select search-directory" and click "Browse" to find the pdi\_hal\_test\_app source directory.
- 4. Select pdi\_hal\_test\_app from the list of Discovered projects:
  - (a) Leave the "Automatically import referenced projects" box checked.
  - (b) Checking the box for "Copy projects into workspace" is optional.
  - (c) Click "Finish". The CCS project will appear in the project explorer window.

## 4 Testing and Results

The main purpose of this test is to demonstrate the usage of the PDI between the C28x processor and the ET1100 EtherCAT Slave Controller. The EtherCAT board is available as both a piggyback board for the F2837x controlCARD and a BoosterPack for use with TI LaunchPads. Both boards have identical hardware and differ only in the connectors used to attach them to their respective processor board.



Figure 28. EtherCAT BoosterPack Test Setup

The test setup in Figure 28 shows a C2000 LaunchPad with attached EtherCAT BoosterPack and two PCs. The left PC connects to the BoosterPack through a standard Ethernet cable connected to Port 0, and runs TwinCAT 3 software, which provides EtherCAT Master functionality. The right PC connects to the LaunchPad directly through the USB connector and runs the EtherCAT Hardware abstraction layer (HAL) software on the CCS6.x development environment. The HAL software in this example performs the following functions:

- Initialize the C28x hardware and the selected PDI interface (SPI or EMIF)
- Execute read/writes to the ET1100 User RAM
- Execute reads from ET1100 register space



The intent of this project is to demonstrate the usage of the PDI. Therefore, no EtherCAT stack is included in this demo.

Note that the HAL software supports both the SPI and EMIF as PDI interfaces. The user must choose the proper settings when building the project. Correct jumper settings on the BoosterPack are also required for proper operation. These settings are described in Section 2.3.

### 4.1 Running Simple ESC Interface Test on C28x

After downloading the software from the project directory at http://www.ti.com/tool/TIDM-DELFINO-ETHERCAT and importing the project into CCS, perform the following steps to exercise the PDI HAL:

1. Open the example project pdi\_hal\_test\_app in CCS. The file pdi\_test\_appl.c has the main routine. The following code is of interest:

ESC_HWInit()	Initializes the C28x MCU and the PDI
ESC_setupPDITestInterface()	Set-up PDI interface and initialize test variables and ET1100 RAM over PDI
ESC_debugUpdateESCRegLogs()	Keep updating ET1100 registers in a loop. User can add to the list of registers to read in the escRegs data structure.



Figure 29. CCS example project: pdi\_hal\_test\_app



2. Right click on the project and set the desired active build configuration. Figure 30 shows the available configurations, which provides options for board types (LaunchPad and controlCARD plugin), program storage (FLASH and RAM), as well as interface type (EMIF, SPI-A,orSPI-C)



Figure 30. EtherCAT HW Abstraction Layer Project



Testing and Results

For this example, choose build configuration (6\_LAUNCHPAD\_SPIA\_RAM) by right clicking on the project name and selecting "Build Configuration"  $\rightarrow$  Set Active  $\rightarrow$  (6\_LAUNCHPAD\_SPIA\_RAM).

File Edit View Navigate Project Run Sorpts Window Help         Image: Sector Project Explorer Image: Sector Project Image: Sector	CCS Edit - Code	Composer Studio		
Image: Sector balance       Image: Sector balance         Image: Sector balance       Sector balance </th <th>File Edit View</th> <th>Navigate Project Run Scripts</th> <th>Window Help</th> <th></th>	File Edit View	Navigate Project Run Scripts	Window Help	
By register two:		• Ø * • * • • •	⇔ •	Quick Access 😰 🕼 CCS Edit
Close Project       Build Configurations       Manage         Build Configurations       Set Active       1       1 CCARD EMIF FLASH (C2000 EtherCAT software running from FLASH on C2000 controlCARD taiking to ET1100 over EMIF2)         Index       Build All       2.2.CCARD_EMIF_RAM (C2000 EtherCAT software running from FLASH on C2000 controlCARD taiking to ET1100 over EMIF2)         Debug As       Clean All       3.3.CCARD_SPIC_FLASH (C2000 EtherCAT software running from FLASH on C2000 controlCARD taiking to ET1100 over SPIC)         Team       Build Selected       5.5_LAUNCHPAD_SPIC_FLASH (C2000 EtherCAT software running from FLASH on C2000 controlCARD taiking to ET1100 over SPIC)         Compare With       Felsee With       V       6.6_LAUNCHPAD_SPIC_FLASH (C2000 EtherCAT software running from RAM on C2000 LaunchPAD taiking to ET1100 over SPIA)         Velses for from Local History       blems 38       8.8_LAUNCHPAD_SPIC_FLASH (C2000 EtherCAT software running from RAM on C2000 LaunchPAD taiking to ET1100 over SPIA)	Project Explore: Image: Project Explore: Ima	23 CCAPD EMIC B New 2 CCAPD EMIC B New 2 CCAPD EMIC B Add Files Copy Paste Delete Refactor Source Move Refactor Source Move Rename Import Export Show Build Settings Build Project Clean Project Clean Project Rebuild Project Defree b	Ctrl+C Ctrl+V Delete ,	
Index     Build All     2.2_CCARD_EMIF_RAM (C2000 EtherCAT software running from RAM on C2000 controlCARD talking to ET1100 over SPIC)       Debug As     3.3_CCARD_SPIC_FLASH (C2000 EtherCAT software running from RAM on C2000 controlCARD talking to ET1100 over SPIC)       Team     8uild Selected     4_4CCARD_SPIC_FLASH (C2000 EtherCAT software running from RAM on C2000 controlCARD talking to ET1100 over SPIC)       Compare With     5_LAUNCHPAD_SPIA_FLASH (C2000 EtherCAT software running from FLASH on C2000 LaunchPAD talking to ET1100 over SPIA)       Replace With     ✓ 6_LAUNCHPAD_SPIA_RAM (C2000 EtherCAT software running from RAM on C2000 LaunchPAD talking to ET1100 over SPIA)       blems ﷺ     8_BLAUNCHPAD_SPIC_FLASH (C2000 EtherCAT software running from RAM on C2000 LaunchPAD talking to ET1100 over SPIC)		Close Project Build Configurations Make Targets	Manage Set Active	1 1 CCARD EMIF FLASH (C2000 EtherCAT software running from FLASH on C2000 control(CARD talking to ET1100 over EMIF2)
Properties Alt+Enter 1-1 warming, u outers 9 _9_LAUNCHPAD_EMIF1_RAM (C2000 EtherCAT software running from RAM on C2000 LaunchPAD talking to ET1100 over EMIF1) 10 _A_LAUNCHPAD_EMIF1_RAM (C2000 EtherCAT software running from RAM on C2000 LaunchPAD talking to ET1100 over EMIF1)		Make Targets Index Debug As Team Compare With Replace With Replace With Restore from Local History Properties	build All Clean All Build Selected blems II Alt+Enter Description	2.2_CCARD_EMIF_RAM (C2000 EtherCAT software running from RAM on C2000 controlCARD talking to ET1100 over EMIF2)     3.3_CCARD_SPIC_FLASH (C2000 EtherCAT software running from RAM on C2000 controlCARD talking to ET1100 over SPIC)     4.4_CCARD_SPIC_RAM (C2000 EtherCAT software running from RAM on C2000 controlCARD talking to ET1100 over SPIA)     ✓ 6.4_LUNCHPAD_SPIA_FLASH (C2000 EtherCAT software running from RAM on C2000 controlCARD talking to ET1100 over SPIA)     ✓ 6.4_LUNCHPAD_SPIA_FLASH (C2000 EtherCAT software running from RAM on C2000 LaunchPAD talking to ET1100 over SPIA)     ✓ 6.4_LUNCHPAD_SPIC_RAM (C2000 EtherCAT software running from RAM on C2000 LaunchPAD talking to ET1100 over SPIA)     7_7_LUNUCHPAD_SPIC_RAM (C2000 EtherCAT software running from RAM on C2000 LaunchPAD talking to ET1100 over SPIC)     8_8_LAUNCHPAD_SPIC_FLASH (C2000 EtherCAT software running from RAM on C2000 LaunchPAD talking to ET1100 over SPIC)     9_9_LAUNCHPAD_SPIC_FLASH (C2000 EtherCAT software running from RAM on C2000 LaunchPAD talking to ET1100 over SPIC)     10.4_LAUNCHPAD_SPIC_FLASH (C2000 EtherCAT software running from RAM on C2000 LaunchPAD talking to ET1100 over SPIC)     9_9_LAUNCHPAD_SPIC_FLASH (C2000 EtherCAT software running from RAM on C2000 LaunchPAD talking to ET1100 over SPIC)     10.4_LAUNCHPAD_SPIC_FLASH (C2000 EtherCAT software running from RAM on C2000 LaunchPAD talking to ET1100 over SPIC)

Figure 31. CCS Project Build Configuration Management

Start TwinCAT3 on the EtherCAT Master PC as described in Section 2.2. Note that double clicking the EtherCAT slave node ("Box 2" in Figure 32) opens up a properties windows, from which the "Advanced Settings window" can be opened. This is used later in the demo.

TEM	Advanced Settings								
1014	🚍 General	Memory							
IDN Perices Perice 2 (EtherCAT) "Image- "Image- "Image- "Synchrits Synchrits Dutputs Dutputs Dutputs Dutputs Dutputs Dutputs Dutputs Dutputs Dutputs Dutputs Dutputs	General Behavior Timeout Settings Identification FMMU/S M Distributed Clock Distributed Clock ESC Access #PFROM FPGA Memory	Memory Start Offset Lengh: Working Counter: Auto Reload CompectView Use Fixed Addr EfferCAT Stave C Unspecified EfferCAT Stave C Decre EfferCAT Stave C Decre EfferCAT Stave C	0 0 0400 1 Peloed Write ontroller Type	Offs 0000 E 0002 E 0004 P 0008 F 0008 0 0000 0000 0 0000 0000 0 0000 0000 0 0000 0000 0000 0000 0000 0000 0000 E	SC Rev/Type SC Build M/FMMU Cnt orts/DPRAM eatures hys Addr onfigured Statio	Dec 17 3 2056 3848 252 0 0 0 0 0 0 0 0 0 0 0 0 0	Hex 0011 0003 0806 0000 0000 0000 0000 0000 0000	Char 	
		Digital (4)	16 µC (8, a)	0-7	FMMU cnt		8		
		SPI (5)	0 16 μC (10, s)     16 μC (10, s)	8-15	SM cnt		8		
		Bridge (7)	) 8 μC (11, s				-		
								OK	Cancel

Figure 32. TwinCAT3 Advanced Settings Dialog



3. Start the debugger and open the memory browser window as shown in Figure 33. The "escRegs" data array contains a list of ET1100 register addresses and values. These get updated in the ESC\_debugUpdateESCRegLogs() function.

ic pdi_test_appl.c	🗆 🚺 Me	nory Browser 🛛 🔤 Disassembly 🖸	1 × •
1085 ESC_debugAddESCRegsAddress(0x100 /*, "dlc1", 4*/);	Data	▼ escReas	8
<pre>1086 ESC_debugAddESCRegsAddress(0x102 /*, "dlc2", 4*/);</pre>	Datas	121c2 - 0x000121C2 - Mamony Rendering 1> 52	
<pre>1087 ESC_debugAddESCRegsAddress(0x110 /*, "dls", 2*/);</pre>	Data	TELCE - 0x000121C2 (Mentoly Kendening 12 (a)	
1088 ESC_debugAddESCRegsAddress(0x310 /*, "llc1", 2*/);	16-B	Hex - TI Style 👻	
<pre>1089 ESC_debugAddESCRegsAddress(0x312 /*, "llc2", 2*/);</pre>	0x00	121C2 escRegs	
<pre>1090 ESC_debugAddESCRegsAddress(0xE00 /*, "porv", 4*/);</pre>	0x06	121C2 0000 FFFF 0002 FFFF 0004 FFFF 0008 FFFF 0100 FFFF 0102 FFFF 0110 FFFF 0310 FFFF 0312 FFFF 0E00 FFFF (	1510
1091 ESC_debugAddESCRegsAddress(0x510 /*, "miis", 4*/);	0x00	12107 FFFF 0512 FFFF 0514 FFFF 0516 FFFF 0518 FFFF 051A FFFF 1000 FFFF FFFF FFFF FFFF FFFF FF	
1892 ESC_debugAddESCRegsAddress(0x512 /*, "phya", 4*/);	0x00	121EATI_cleanup_ptr	
1893 ESC_debugAddESCRegSAddress(0x514 /*, "phyd", 4*/);	0x00	0121EA 0000 0000	
1094 ESC_debugAddESCKegSAddresS(0X516 /*, mila, 4*/);	0x00	121ECTI_dtors_ptr	
1895 ESC_debugAddESCRegsAddress(0x518 /*, port0, 5*/); 1896 ESC_debugAddESCRegsAddress(0x518 /*, "port1", 5*/);	0x00	121EC 0000 0000	
1007 ESC_debugaddSCRegsAddress(0x10A /* "RAM address" 11*/).	0x06		
1007 ESC_0000ghoutSchegshouress(0x1000 / , Horr address , 11 / ),	0000		
1099	0x00		
1100 //Now wait till EEPROM is loaded	0,00	122170 9104 0000 0000 0000 0000 0000 0000 000	
1101 do	0x00	12200 CPUTATEL 4000 0000 0000 0000 0000 0000	
1102 {	0x06	12208 Coulimer?	
<pre>pdi_control = ESC_readWordNonISR(0x0140 /*ESC_PDI_CONTROL_OFFSET*/);</pre>	0x00	012208 0C10 0000 0000 0000 0000 0000 0000 00	
1104#ifdef INTERFACE_SPI	0x00	12210 CpuTimer0	
1105 } while (((pdi_control & 0xFF) != 0x05)); //SPI PDI	- 0x00	12210 0000 0000 0000 0000 0000 0000 000	9000
1106 <b>#else</b>	0x00	12225 0000 0000 0000 0000 0000 0000 0000	1000
1107 } while (((pdi_control & 0xFF) != 0x08)); //EMIF PDI	0x00	1223A 0000 0000 0000 0000 0000 0000 0000	000
1108 #endif	0x00	1224F 0000 0000 0000 0000 0000 0000 0000	000
1109	0x00	12264 0000 0000 0000 0000 0000 0000 0000	0000
1110	0x00	12279 0000 0000 0000 0000 0000 0000 0000	0000
1111 // NOW EEPNOH is loaded and PDI is either EMIF (10 Dit) of SPI.	≣_ 0x00	1228E 6000 6000 6006 6000 6006 6000 6000 60	000
1112 // CILLOU NAW Fedu/Write tests, on Cillou NAW being at 0x1000 and can go up to 0xPPP	0x00	122A3 0000 0000 0000 0000 0000 0000 0000	000
	• 0x00	17322 0000 0000 0000 0000 0000 0000 0000	000
	0X06	2175CD	1000

Figure 33. Device Debug Memory Window at INIT Time

**NOTE:** The ESC RUN LED will NOT be on during the memory tests, which is described in the following sections. For more information, see the ET1100 data sheet.

## ESC RAM READ TEST

4. Open up the "advanced settings" window in TwinCAT3 and go to the memory browser at ESC Access → Memory and view Start Offset = 1000h as shown in Figure 34.

												l
- General	Memory											
- Timeout Settings	Start Offset	1000	Offs		Dec							
Identification	Length:	0400	1000		4660		0//		Dec		Char	
- Init Commands	Working Counter:	ing Counter: 1	1002		22136		Ulls		Dec	Hex	Char	=
Distributed Clock	-		1004		0		1000		4660	1234	4.	
ESC Access	Auto Reload	Reload	1006		0		1000		22126	5670		
FPGA	Compact View	Compact View         Write           Use Fixed Addr         EtherCAT Slave Controller Type           Outpacefied         Esci 10/20           O Force         ET1100           O ET1200         ET1200	1008		0		1002		22130	30/8	XA	
Memory	Use Fixed Addr		100a		0		1004		0	0000		
	EtherCAT Slave Co		100c				1000		0	0000		
			Unspecified 100e ESC 10/20 1010	0		1008			U	0000		
	ESC 10/20				0	0000						
	© ET1100		1012 1014	0	0 0000							
	C ET1200				0 0	0000	0000					
	PDI Type		1016	_	0	0000						
	Unspecified  Digital (4)  SEL(5)	<ul> <li>16 μC (8, a)</li> <li>8 μC (9, a)</li> <li>16 μC (10, a)</li> </ul>	Bits	Name		Value	Enum					
	<ul> <li>Bridge (7)</li> </ul>	8 µC (11, s)										

Figure 34. TwinCAT3 Memory Read Window



Testing and Results

5. In CCS, view the Expressions window and add "escRegs" as a watched expression. Note that, as data is entered into the window in TwinCAT, the values read on the CCS side are identical.

	🔋 Memory Browser 📟 Disasse	mbly 😭 Expressions 🛛		
	Expression	Туре	Value	Address
② pdjtest_applc ③ etherat_slave_c2&x_halc ☆ 0 1121 // which is equivalent to 0x2800 to 0x2FFF of the C28x EMIF2 address space. ▲ E	Expression Expression	struct esc_et1100_re	0x000121C2@Data	0x000121C2@Da
<pre>1122 //please refer to ETI100 datasheet for more details 1123 for (test_address = ESC_PDI_RAM_START_ADDRESS_OFFSET; test_address &lt;= ESC_PDI_RAM_END 1128.metrix</pre>		unsigned long	0x00001000 (Hex)	0x0000040A@Da
1124 menuar 1125 { 1126		unsigned long	0x56781234 (Hex)	0x00000408@Da
1127     dtest_data = ESC_readDWordNonISR(test_address);       1128     ESC_writeWord(0x8AADF00D, test_address);	🖶 Add new expression			
<pre>1125</pre>	est_address); iness); ist_address);; MOD)			

Figure 35. EtherCAT Write to Memory From CCS Project

## **ESC RAM WRITE TEST**

6. Figure 36 shows program control after execution of the first 32-bit write and 32-bit read from PDI to ET100 address 0x1000.

			🔋 Memory Browser 📟 Disassembly 👫 Expressions 🛛						
			Expression	Туре	Value	Address			
pdi_test_appl.c      12     i ethercat_slave_c28x_hal.c      22		Memory Brow	escRegs	struct esc_et1100_re	0x000121C2@Data	0x000121C2@Da			
1121 // which is equivalent to 0x2800 to 0x2FFF of the C28x EMIF 1122 //please refer to ET1100 datasheet for more details for (test address = ESC PDI RM START ADDRESS OFFSFI: test /	Expression	⋈= test_address	unsigned long	0x00001000 (Hex)	0x0000040A@Da				
1124 #endif 1125 {		e dtest_data	⋈= dtest_data	unsigned long	0xBAADF00D (Hex)	0x00000408@Da			
1126       1127       dtest_dutaESC_readDWordNonISR(test_address);       1128       ESC_writeDWord(0xBnucleaco_test_address);		- Add new ex	Add new expression						
1120         dtest_data = ESC_readWordIonISt(test_datarss);;           1130         if(dtest_data = (Comparing to the state of the st	test_data = ESC_r SC_writeDWord(0xE test_data = ESC_r f(dtest_data != ( {	readDWon BAADF001 readDWon (uint32	rdNonISR(test_address ), test_address); rdNonISR(test_address t)0xBAADF00D)	);					

Figure 36. CCS Project Write to EtherCAT Slave



7. The TwinCAT Master view of the ESC memory address showing the results of the write from CCS.

Advanced Settings									×	
General	Memory									
- Timeout Settings	Start Offset:	1000	Offs	Offs				Dec	Hex	Char
FMMU / SM Init Commands	Working Counter:	1	1000	1000				61453	f00d	
ESC Access	Auto Reload	Reload Write	1004 1006	1002				47789	baad	
	Use Fixed Addr	xed Addr	1008 100a	1003 1003 1004 100c 1006			0	0000		
	EtherCAT Slave C	Controller Type	100c 100e					0	0000	
	© ESC 10/20 © IP core © ET1100 © ET1200 PDI Type © Unspecified ©	ESC 10/20 P core ET1100 ET1200	1010 1012 1014		0	0000		**		
		cified 💿 16 µС (8, а)	Bits	Bits Name Value En			Enum			
	<ul> <li>Digital (4)</li> <li>SPI (5)</li> <li>Bridge (7)</li> </ul>	8 µС (9, а)     16 µС (10, s)     8 µС (11, s)								
								OK Cano	el	

Figure 37. TwinCAT3 Master Read of Data Written to EtherCAT Slave

## 4.2 TwinCAT3 Troubleshooting

Common issues in TwinCAT3 usage:

- Problem: EtherCAT network fails to initialize
  - Other Descriptions: "Reload Devices" fails, "Scan" for devices fails, "Restart EtherCAT in config mode" fails.
- Solutions:
  - Power Cycle the LaunchPad
  - Check to make sure a RealTime Ethernet Driver is available.
    - TwinCAT  $\rightarrow$  "Show RealTime Ethernet Compatible Devices". This opens a popup window below.
    - Look for the first line "Installed and ready to use devices (realtime capable)".
      - In this example, there are NO adapters installed!
    - Select a compatible device; here it is "Local Area Connection  $\rightarrow$  Intel ...." and click "Install".
    - Close the window by clicking "X" in the upper right corner.



#### Testing and Results

www.ti.com

1stallation of TwinCAT RT-Ethernet Adapters									
Ethernet Adapters	Update List								
Installed and ready to use devices(realtime capable) Installed and ready to use devices(for demo use only)	Install								
Compatible devices     Local Area Connection - Intel(R) Ethemet Connection (3) I218-LM #3	Update								
Incompatible devices     Wireless Network Connection - Intel(R) Dual Band Wireless-AC 7265	Bind								
Disabled devices	Unbind								
	Enable								
	Disable								
	☐ Show Bindings								

Figure 38. TwinCAT3 Ethernet Adapter Installation



## 5 Design Files

## 5.1 Schematics

To download the schematics, see the design files at TIDM-DELFINO-ETHERCAT.

## 5.2 Bill of Materials

To download the bill of materials (BOM), see the design files at TIDM-DELFINO-ETHERCAT.

## 5.3 PCB Layout Recommendations

## 5.3.1 Layout Prints

To download the layer plots, see the design files at TIDM-DELFINO-ETHERCAT.

## 5.4 Altium Project

To download the Altium project files, see the design files at TIDM-DELFINO-ETHERCAT.

## 5.5 Gerber Files

To download the Gerber files, see the design files at TIDM-DELFINO-ETHERCAT.

## 5.6 Assembly Drawings

To download the assembly drawings, see the design files at TIDM-DELFINO-ETHERCAT.

## 6 Software Files

To download the software files, see the design files at TIDM-DELFINO-ETHERCAT.

## 7 References

- 1. LAUNCHXL-F28377S C2000 Delfino MCU F28377S LaunchPad Development kit
- 2. TMDXDOCK28377D F28377D Delfino MCU Experimenter Kit
- 3. Code Composer Studio (CCS) Integrated Development Environment (IDE)
- 4. Texas Instruments E2E<sup>™</sup> online community: http://e2e.ti.com/

## 8 Terminology (Optional)

- EMIF: External Memory Interface
- ESC: EtherCAT Slave Controller
- ETG<sup>™</sup>: EtherCAT Technology Group
- LaunchPad: An easy-to-use development tool intended for MCU-based applications.
- MCU: MicroController
- PDI: Process Data Interface
- SPI: Serial Peripheral Interface



## 9 About the Author (Optional)

**ROBERT LANDERS** is a Systems Applications Engineer at Texas Instruments, where he is responsible for developing design solutions for the industrial drives segment. Robert earned his Master of Science in Electrical Engineering (MSEE) from the University of Texas Arlington. He is a member of the Institute of Electrical and Electronics Engineers (IEEE).

**SANTOSH ATHURU** is a Software Engineer at Texas Instruments, where he is responsible for TI C2000 embedded software applications. Santosh earned his Bachelors of Engineering (B.E) from the University of Madras, India. He is a member of the Institute of Electrical and Electronics Engineers (IEEE).



## **Revision History**

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Cł	nanges from Original (April 2016) to A Revision	Page
•	Replaced TLK105 with DP83822 devices throughout the document.	1
•	Updates were made to Section 1.1.	2
•	Updates were made to Section 2.	10
•	Updates were made to Section 2.3.	15

#### **IMPORTANT NOTICE FOR TI REFERENCE DESIGNS**

Texas Instruments Incorporated ('TI') reference designs are solely intended to assist designers ("Designer(s)") who are developing systems that incorporate TI products. TI has not conducted any testing other than that specifically described in the published documentation for a particular reference design.

TI's provision of reference designs and any other technical, applications or design advice, quality characterization, reliability data or other information or services 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 reference designs or other items.

TI reserves the right to make corrections, enhancements, improvements and other changes to its reference designs and other items.

Designer understands and agrees that Designer remains responsible for using its independent analysis, evaluation and judgment in designing Designer's systems and products, and has full and exclusive responsibility to assure the safety of its products and compliance of its products (and of all TI products used in or for such Designer's products) with all applicable regulations, laws and other applicable requirements. Designer represents that, with respect to its applications, it has 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. Designer agrees that prior to using or distributing any systems that include TI products, Designer will thoroughly test such systems and the functionality of such TI products as used in such systems. Designer may not use any TI products in life-critical medical equipment unless authorized officers of the parties have executed a special contract specifically governing such use. Life-critical medical equipment is medical equipment where failure of such equipment would cause serious bodily injury or death (e.g., life support, pacemakers, defibrillators, heart pumps, neurostimulators, and implantables). Such equivalent classifications outside the U.S.

Designers are authorized to use, copy and modify any individual TI reference design only in connection with the development of end products that include the TI product(s) identified in that reference design. HOWEVER, 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 published by TI regarding third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of the reference design or other items described above 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 REFERENCE DESIGNS AND OTHER ITEMS DESCRIBED ABOVE ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING THE REFERENCE DESIGNS OR USE OF THE REFERENCE DESIGNS, 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 DESIGNERS AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS AS DESCRIBED IN A TI REFERENCE DESIGN 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 THE REFERENCE DESIGNS OR USE OF THE REFERENCE DESIGNS, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

TI's standard terms of sale for semiconductor products (http://www.ti.com/sc/docs/stdterms.htm) apply to the sale of packaged integrated circuit products. Additional terms may apply to the use or sale of other types of TI products and services.

Designer will fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of Designer's noncompliance with the terms and provisions of this Notice.

> Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265 Copyright © 2016, Texas Instruments Incorporated