

Real-time Control Meets Real-time Industrial Communications Development – Part 3



Brian Fortman

In part 1 of this blog series, I gave an overview of the market opportunity for EtherCAT® slave stack support on C2000™ microcontrollers (MCUs) and an introduction to a three-phased solution to jump-start slave stack development. In part 2, I outlined what is different and compelling about the TI C2000 MCU EtherCAT implementation. In this installment, I'll break down the three phases of the development of your own slave-node application and how the TI EtherCAT package in C2000 controlSUITE™ software supports each phase.

Given the previously outlined complementary aspects of EtherCAT technology and C2000 MCUs for motion control applications – and because of customer requests – TI applied its embedded software development experience and C2000 MCU architecture knowledge to create a set of software solutions designed to aid in the development of EtherCAT slave nodes on C2000 real-time control MCUs. The software package will help you:

- Evaluate the stack for a slave node.
- Validate the correct operation/hookup of your custom hardware.
- Create a complete slave-node application using your hardware and a stack distributed by the EtherCAT Technology Group.

Step 1: Evaluate the C2000 EtherCAT Node (Stack and Hardware)

The first software solution included in the C2000 EtherCAT support is meant to make it easy to quickly get familiar with the basic solution. Leveraging the TwinCAT PC software for the EtherCAT master node, the simple “echoback” example application and you can build a multinode EtherCAT network on your bench in minutes.

The EtherCAT “slave demo” project is a binary image release for the EtherCAT controlCARD, optimized for the C28x CPU, that includes the slave stack binary version. So all you need to do is program the C2000 MCU and connect a CAT5 cable to the EtherCAT master.

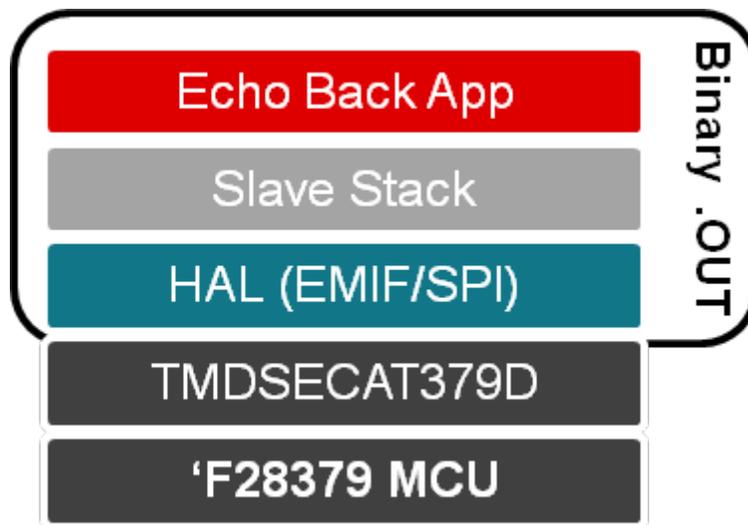


Figure 1. C2000 MCU Out-of-the-box EtherCAT Slave Demonstration and Evaluation Solution

This evaluation project will help you answer questions like: What is the transfer latency from the external EtherCAT controller (ET1100) to the C2000 MCU? How much central processing unit (CPU) bandwidth does this solution take? What is the memory footprint of the hardware abstraction layer (HAL) and the slave stack?

After downloading the latest controlSUITE software, you can install the demonstration and evaluation example by executing the setup file, “EtherCAT_Slave_Demo_Code_v01_00_00_00_setup,” in the controlSUITE development kit project.

Step 2: Validate Your EtherCAT Hardware

The second software solution shifts from being based on the TI controlCARD tool to helping you develop your own hardware. The EtherCAT community refers to the interface to the programmable controller as the processor data interface (PDI). Recall that you can find reference hardware designs for both the parallel PDI, “ASYNC16” (C2000 MCU external memory interface [EMIF]) and serial PDI (C2000 MCU serial peripheral interface [SPI]). This software project will help you verify that hardware interfaces you have built “from the MCU to the wire” are correct.

Since the hardware abstraction layer (HAL) test application can communicate with a TwinCAT master node, the software project helps you validate that the physical PDI interfaces (EMIF and SPI port) to the EtherCAT controller (such as the ET1100) – as well as the physical layer (PHY) connections and Ethernet magnetics – are all connected properly.

You do not need an EtherCAT slave stack to validate the hardware connection in this case. Unlike step 1, the HAL test project is available as source code to aid in validation or to adapt C2000 MCUs to other physical interfaces for EtherCAT controllers.



Figure 2. C2000 MCU HAL Test Application

Step 3: Create Your Own EtherCAT Slave-node Solution

Finally, the EtherCAT application reference project provides a step-by-step example when it’s time to integrate your validated hardware with the EtherCAT stack. This project contains the source code necessary to build an example EtherCAT slave node (initialization, HAL, echoback application) except for the EtherCAT slave stack itself. You must obtain the slave stack through the EtherCAT Technology Group; it is distributed as part of the slave stack configuration (SSC) tool.

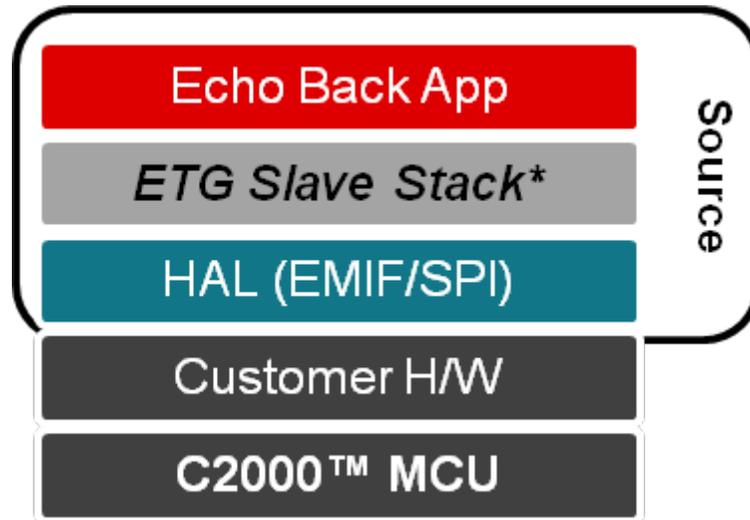


Figure 3. C2000 MCU Native EtherCAT Slave Stack Reference Development Framework

In the event that your version of the configuration tool does not include support for C2000 MCUs, the controlSUITE project includes a patch that will make C2000 MCUs (and the ASYNC16 and SPI PDI options) available to you in the drop-down processor selection dialog box.

Two project configurations are available:

- A random access memory (RAM)-based project, which is usually helpful during the initial stages of a project when code changes are frequent (less time spent programming flash).
- A flash-based project, when the code is more mature and ready for testing through power-cycling of the target C2000 MCU.

With the release of the EtherCAT Slave and C2000 Delfino™ MCU controlCARD kit and EtherCAT solution reference in controlSUITE, software you can jump-start the development of an EtherCAT slave stack running directly on C2000 MCUs.

Additional Resources

- Purchase the [C2000 EtherCAT controlCARD](#).
- To learn more about TI's EtherCAT slave-node solutions for C2000 MCUs, check out:
 - The [DesignDRIVE training portal](#).

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on [ti.com](https://www.ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2023, Texas Instruments Incorporated