Technical Article

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

Brian Fortman

In part one of this blog series, I gave an overview of the market opportunity for EtherCAT slave stack support on C2000 MCUs and an introduction to a three phased solution to jumpstart slave stack development.

Beyond this three-faceted development flow, what has TI done to make our solution more compelling than a typical stack porting exercise? First, we applied our knowledge of the C28 CPU architecture and tuned the software to leverage the CPU services accordingly. For example, we optimized the interrupt handling, direct memory access (DMA) and control (pulse-width modulation [PWM]) synchronization routines to make the best use of the on-chip hardware. In addition, we worked with Beckhoff to update/enhance the released stacks to support data handling so that it's more tailored for the C28 CPU.

Speaking of working with Beckhoff, the C28 slave stack and hardware abstraction layer (HAL) for both Serial Peripheral Interface (SPI) and parallel (external memory interface [EMIF]) communications are supported as part of their Slave Stack Configuration tool (SSC). Based on previous C2000 MCU experience, you will find the reference software solution release "Plugfest-ready."

Supporting both the SPI and EMIF hardware interfaces provides implementation options, enabling system trade-offs on printed circuit board (PCB) routing complexity vs. the lowest latency communications. In addition, while we tested the release directly with the Beckhoff ET1100 device, you will be able to adapt the SPI and EMIF HAL drivers to support other EtherCAT slave silicon solutions, including TI's AMIC110 SoC (SPI – please contact the Sitara™ forum) or field-programmable gate array (FPGA) instances (EMIF or SPI) of the slave hardware such as the Beckhoff ET1816.

Among SPI and EMIF processor data interfaces (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 EMIF slave nodes can be different even when they are both in the same network. EMIF slave nodes can be different when they are both in the same network.

A C2000 MCU-specific feature of TI's EtherCAT software package is the Echoback application example. While the application demonstration is not complex, it simply loops slave-node output data structures back to inputs, viewable using any EtherCAT master or the TwinCAT master as detailed in the "EtherCAT Solution Reference Guide."

By including the Echoback application, TI shows an example of the basic use of the stack software and provides a placeholder for you to create your own applications. Following the Echoback example will make it easier to take advantage of the many optimizations TI has made in the stack and HAL software modules and gives a good example of the EtherCAT slave information (ESI) file as well. Echoback is also included as part of the quick evaluation binary project.

Finally, there is the compelling EtherCAT controlCARD hardware platform itself, Figure 1, that leverages the common 180-pin interface of other C2000 MCU control cards. The EtherCAT controlCARD is physically compatible with any 180-pin C2000 application evaluation module (EVM) or dock, thus making it possible to add EtherCAT connectivity to many existing real-time control application examples. For example, you can use the EtherCAT controlCARD in conjunction with the industrial drives development kit (IDDK) to add real-time connectivity to a servo using fast current loop and/or position manager technology.

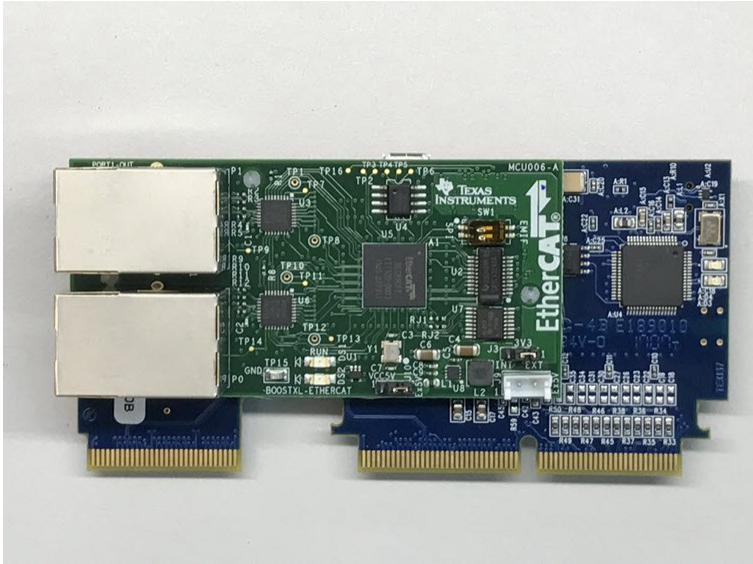

Figure 1. TMDSECATNCD379 EtherCAT Board

Another nice feature is that the EtherCAT controlCARD can operate stand-alone. Powering the hardware with a Micro USB cable eliminates the need for a dock or EVM to supply power. See the "TMDSECATCNCD379D EtherCAT Solution Reference Guide" for details. This can help reduce hardware complexity in multi-node network testing and configuration.

As you can see, the C2000 DesignDRIVE EtherCAT support has taken several additional steps to ease your development of your own slave node and help you get great performance running the stack on our microcontrollers. In the following posts of this series, I will provide more description of the software support provided by each of the three development phases outlined previously.

Additional Resources

To learn more about TI's EtherCAT slave-node solutions for C2000 MCUs, see:

- DesignDRIVE training portal.
- DesignDRIVE applications 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 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