

# EtherCAT<sup>®</sup> on Sitara<sup>™</sup> Processors



**Maneesh Soni**  
*Systems Manager*  
*Arm<sup>®</sup> Microprocessor Group*  
*Texas Instruments*

# **EtherCAT® is among the leading communications standards based on Ethernet that is used increasingly for networking and communications in the industrial or factory environment.**

---

The EtherCAT communication technology was invented by Beckhoff Automation in Germany and later standardized by the EtherCAT Technology Group (ETG).

Texas Instruments, Inc. (TI) is the first semiconductor company to license EtherCAT technology. TI has integrated EtherCAT into all of the Sitara™ processors. To enable EtherCAT, TI has built upon its programmable real-time unit (PRU) technology to create a unified front-end for industrial communications and bring EtherCAT and other industrial standards to its growing platform of Arm-based microprocessors.

TI has also brought the software, hardware and tools together to streamline the development of EtherCAT-based products with Sitara devices. Additionally, the industrial grade temperature and long life-cycle support make Sitara a compelling choice for EtherCAT and other industrial networking applications.

The integration of EtherCAT into Sitara processors enables best-in-class functionality at lower cost. For example, the Sitara AM335x processor-based integration of EtherCAT meets or exceeds all required features and performance benchmarks, including key EtherCAT features such as distributed clocking and end-to-end latency of less than 700 nanoseconds (ns). In addition to the capabilities of Sitara processors, TI streamlines the development of EtherCAT products by supporting design engineers with a wide range of related software, hardware and development tools.

## **Introduction to EtherCAT**

EtherCAT (Ethernet for Control Automation Technology) is a real-time industrial Ethernet standard for industrial automation applications, such as input/output (I/O) devices, sensors and programmable logic controllers (PLCs). It was originally developed by Beckhoff Automation GmbH but is now overseen by the EtherCAT Technology Group that was set up to help with proliferation of the EtherCAT standard. Today, there are over 1,900 member companies from 52 countries that create and deploy EtherCAT-compatible products. Ethernet has seen unparalleled adoption in diverse applications, but in industrial environments it is still not efficient enough for small amounts of data exchange, it has low determinism for real-time operation, and it works with only star topology in which the network nodes must be connected through switches. EtherCAT technology adds certain features on Ethernet and enforces certain

configurations to make it a very efficient network technology for automation while fully conforming to the Ethernet specifications. The design of EtherCAT enables any standard PC to be used as an EtherCAT master and communicate with EtherCAT slaves, which are specialized devices compliant with the EtherCAT specification. Together, the master and slave EtherCAT devices can be used in all devices in the factory network – automation controllers, operator interfaces, remote input/output units, sensors, actuators, drives and others.

## Technology

EtherCAT improves upon traditional Ethernet by implementing “on-the-fly” processing where the nodes in the EtherCAT network read the data from a frame as it passes through. All EtherCAT frames originate from the EtherCAT master which sends commands and data to the slaves. Any data to be sent back to master is written by the slave into the frame as it passes through.

This helps eliminate the need for point-to-point exchange of small-sized frames between master and individual slaves and drastically improves the efficiency of communication. However, it also means that each slave must have two Ethernet ports and be able to let the frame pass through while reading from or writing to the passing frame and therefore, specialized hardware is required in the slave devices. As a result of these improvements, the usable bandwidth in a 100-Mbps network running EtherCAT is more than 90 percent as compared to less than 5 percent for networks where the master must separately communicate with each slave node.

## EtherCAT telegram

As illustrated in **Figure 2**, the EtherCAT telegram is encapsulated in an Ethernet frame and includes one or more EtherCAT datagrams destined to the EtherCAT slaves. Such Ethernet frames use the EtherCAT type in the header or they can be

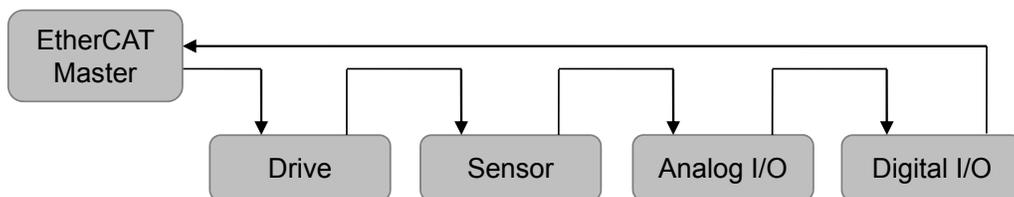


Figure 1. Example EtherCAT network.

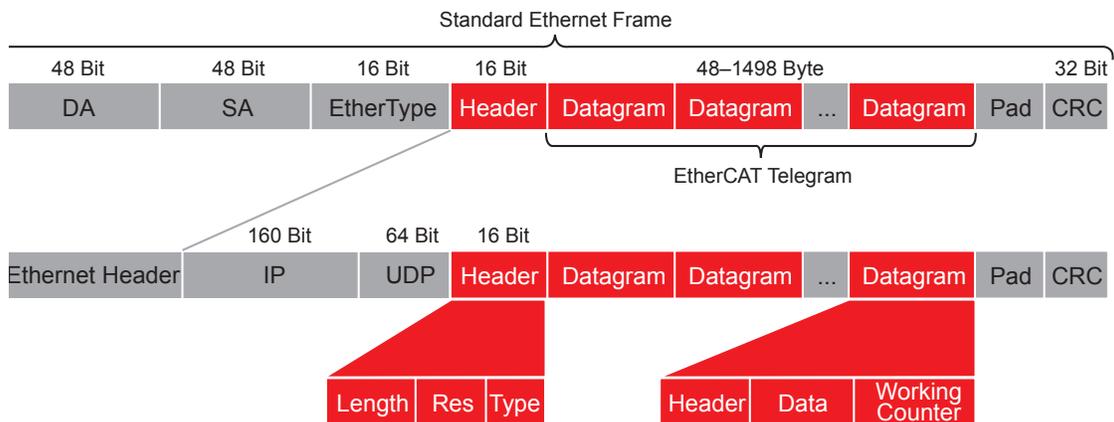


Figure 2. EtherCAT telegram.

packed with the IP/UDP header. When the IP header is used, the EtherCAT protocol can also be used across network routers.

Each EtherCAT datagram is a command that consists of a header, data and a working counter. The header and data are used to specify the operation that the slave must perform, and the working counter is updated by the slave to let the master to know that a slave has processed the command.

### *Protocol*

Each slave processes EtherCAT packets “on-the-fly” in that it receives the frame, parses it and takes action if the address specified in an EtherCAT datagram matches its own address, and forwards the entire datagram from its second port while also updating the contents and the CRC of the packet. Through the datagrams, the EtherCAT master addresses the entire address space of up to 4 GB in which up to 65,536 EtherCAT slaves, each with 65,536 addresses, can be located. The EtherCAT datagrams do not have any restriction on the order in which the slaves are addressed with respect to the actual position of slave nodes in the network.

There are different types of EtherCAT data transmissions – cyclic and acyclic. Cyclic data are the process data that are transferred at periodic intervals or cycle times. Acyclic data is usually non-time-critical data that can be large in size and usually exchanged in response to a controller command. Some acyclic data, such as diagnostic data, can be critical and have demanding timing requirements. EtherCAT handles these different data transmission requirements through optimized addressing schemes – physical addressing, logical addressing, multiple addressing and broadcast addressing.

To handle various addressing schemes, each slave has a fieldbus memory management unit (FMMU).

The FMMU units in each slave enable the EtherCAT protocol to treat various slave devices as part of a 4-GB large memory space with slave spaces mapped in it. The EtherCAT master assembles a complete process image during the initialization phase and then makes even bit-level accesses to slave devices via a single EtherCAT command. This capability makes it possible to communicate practically with any number of input/output (I/O) channels across large and small devices spanning the entire fieldbus network via a standard Ethernet controller and standard Ethernet cable.

### *Performance*

As a result of hardware-based FMMU and on-the-fly processing, the EtherCAT network performs at very high levels of efficiency. It enables cycle times of the order of microseconds to communicate from controllers to field devices. The communication efficiency is no longer a bottleneck in industrial networks and brings it in line with the computation speeds of contemporary industrial PCs. For instance, the increased performance makes it possible to run the current loop, in addition to the position loop, for distributed drives over EtherCAT.

### *Topology*

The EtherCAT standard supports any topology – line, star or tree – and the bus structures common in fieldbus networks can also be realized with EtherCAT. Since the EtherCAT interface is present on I/O devices, there is no requirement for any Ethernet switching hardware. With the 100-m range of copper links and even longer with optical links, EtherCAT can span over thousands of devices spread across a large geographical area. For short distances, such as on back-plane, EtherCAT uses E-bus, a differential signaling technology.

### Distributed clocking

To realize simultaneous actions in industrial nodes installed away from each other, it is necessary to synchronize their internal clocks. EtherCAT accomplishes this by sampling the timestamps for the ingress and egress of an EtherCAT packet on every slave node as it traverses the network. The master uses the timestamp information provided by the slaves to accurately calculate the propagation delay for each individual slave. The clocks in each slave node are adjusted based on this calculation and thus, these clocks are synchronized to within 1  $\mu$ s of each other. An additional advantage of the accurately synchronized clocks is that any measurements taken can be linked to the synchronized time and remove the uncertainty associated with the jitter in the communication between devices.

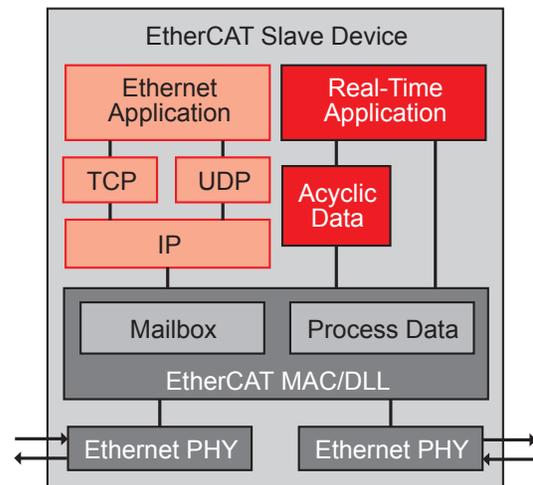
### Device profiles

In industrial automation, use of device profiles is a very common method to describe the functionality and parameters of the devices. EtherCAT provides interfaces to existing device profiles so that legacy fieldbus devices can be easily upgraded to use EtherCAT. Some of such interfaces are CAN application layer over EtherCAT (CoE) and Servo drive profile over EtherCAT (SoE) that enable use of CANOpen<sup>®</sup> and SERCOS<sup>®</sup> by taking advantage of the mapping of their data structures to EtherCAT.

### Components of an EtherCAT node

Each EtherCAT node (**Figure 3**) has three components – the physical layer, the data link layer and an application layer.

The physical layer is implemented using 100BASE-TX copper, 100BASE-FX optical fiber or E-bus based on LVDS signaling. The MAC is implemented either in a specialized ASIC or an FPGA as per the EtherCAT standard specifications. Beyond the



**Figure 3.** Components of an EtherCAT node.

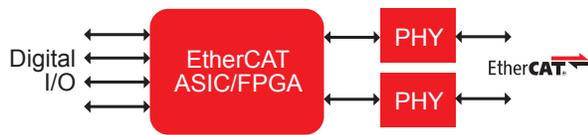
MAC is the industrial application that takes care of application-specific behavior and a standard TCP/IP and UDP/IP stack to support Ethernet-based device profiles. Depending on the complexity of the device, the EtherCAT node can be implemented in hardware or it could be a combination of hardware and software running in an embedded CPU.

### Compliance

To ensure broad interoperability among devices designed with EtherCAT interfaces, the EtherCAT Technology Group (ETG) has several programs for ensuring compliance with the technical specifications. These programs include the conformance test tool (CTT), which is a software program for testing compliance; the plug-fests where members can meet and test against one another's devices; and certification labs in Germany and Japan where formal certification tests are performed. To meet minimum conformance requirements a device has to pass the protocol test using the conformance test tool at the time of its first release to the market. Optionally, vendors can choose to get their products certified in any of the authorized certification labs. The ETG website provides detailed information on procedure and location of certification labs.

## Typical EtherCAT® node

A typical EtherCAT node that is in use today has architecture similar to one of the illustrations below. Many of the simple EtherCAT devices such as digital I/O can be created using single FPGA or ASIC solutions available today. A simplified version of such architecture is shown in **Figure 4**. Such architecture is well suited for cost-sensitive simple I/O nodes that do not require software and all functionality is implemented in hardware.



**Figure 4.** Basic Digital I/O EtherCAT device.

In the EtherCAT nodes where additional processing power is needed, an external processor, often with on-chip Flash memory, is connected to the EtherCAT ASIC/FPGA for handling the application-level processing. Such devices could be sensor applications, for instance, where the processor is required to operate the sensor, implement the device driver and run the EtherCAT protocol stack. The cost of such architecture is higher than that for simple digital I/O devices and it comes with the flexibility that developers can select a processor that suits their needs and cost targets.



**Figure 5.** EtherCAT with ASIC and external processor.

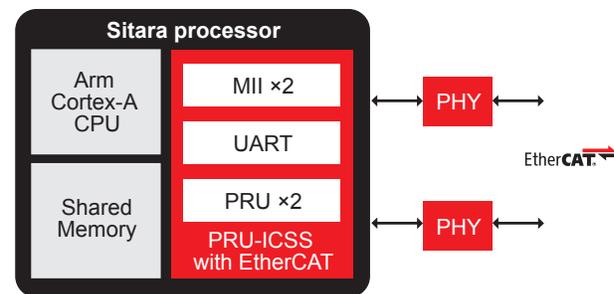


**Figure 6.** Integrated EtherCAT with processor.

In yet another approach, the EtherCAT implementation is one of the peripherals in the device that has an integrated CPU. Many FPGA devices have the capability to configure a processor in the FPGA or already have an integrated processor. Some vendors provide ASICs with both EtherCAT and a suitable processor on the device. The FPGAs are flexible but depending on the CPU selection, there is a risk that cost or operating frequency targets are challenging to meet.

## EtherCAT solution from TI

TI has integrated EtherCAT functionality into the Sitara processors. These devices integrate an Arm core with a cornucopia of other peripherals and interfaces that make them attractive devices for building industrial automation equipment.



**Figure 7.** EtherCAT slave on TI Sitara processors.

The Sitara processors integrate the programmable real-time unit industrial communication subsystem (PRU-ICSS), which supports very low-level interaction with the MII interfaces. This capability enables the PRU-ICSS to implement specialized communication protocols such as EtherCAT.

The entire EtherCAT MAC layer can be encapsulated in the PRU-ICSS through firmware. The PRU-ICSS processes EtherCAT telegrams on-the-fly, parses them, decodes the address and executes EtherCAT commands. Interrupts are used for any communication required with the Arm core where the EtherCAT stack (Layer 7) and the industrial application is running. The PRU-ICSS also performs frame forwarding in the reverse direction. Since the PRU-ICSS can implement all EtherCAT functionality, the Arm core can be utilized for complex applications or a lower-speed Arm core can be deployed for simpler and cost-constrained applications, such as distributed I/O.

To complete the EtherCAT solution with the Sitara processors, Ethernet PHY devices such as TI's TLK105L, TLK106L, DP836X0, DP83822 or DP8384x are required.

### EtherCAT software architecture

Three major software components comprise an EtherCAT slave implementation on one of TI's Sitara processors. The first is the micro-code that implements Layer 2 functionality in the PRU; the second is the EtherCAT slave stack that runs on the Arm core and the third is an industrial application that is dependent on the end equipment in which this solution is used. Additional supporting components, such as the protocol adaptation layer and device drivers are provided by TI in the Processor Software Development Kit (SDK). Irrespective of whether a TI-tested EtherCAT stack is used or another, the architecture illustrated in **Figure 8** on the following page is designed to work without changes. This EtherCAT solution is also independent of the OS and any adaptations can be made by referring to the PRU-ICSS firmware API guides.

In EtherCAT Layer 2, the PRU real-time cores share the tasks of datagram processing, distributed clocking, address mapping, error detection and handling and host interface.

PRUs also emulate EtherCAT register space in the internal shared memory. With their deterministic realtime processing capability, the PRUs handle EtherCAT datagrams with consistent and predictable processing latency. The Sitara processor with TI's DP83822 Ethernet PHY device exhibits a low latency which makes TI's implementation one of the leading EtherCAT slave solutions.

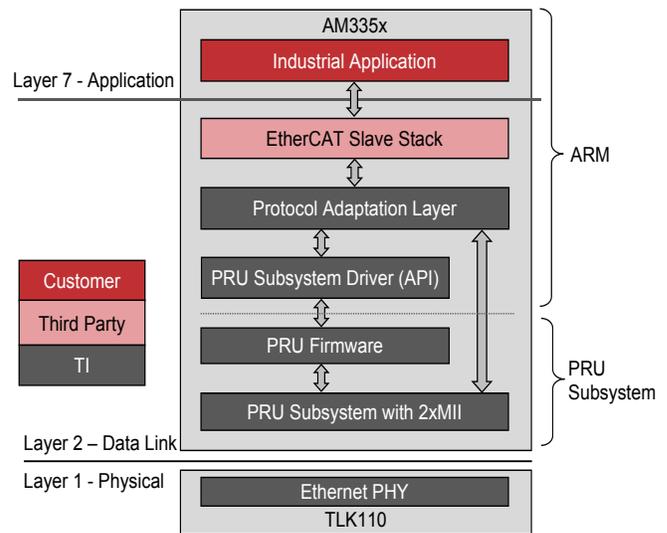


Figure 8. Software architecture for EtherCAT slave.

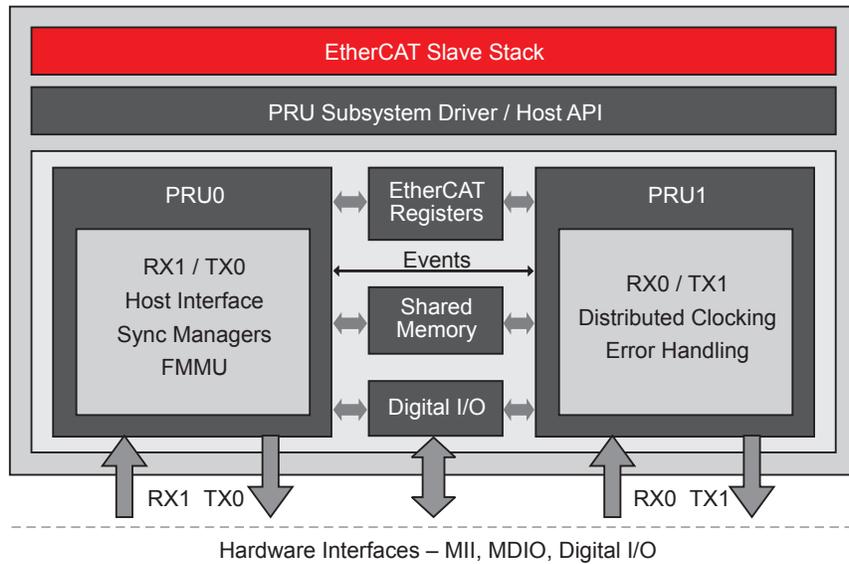


Figure 9. EtherCAT firmware architecture.

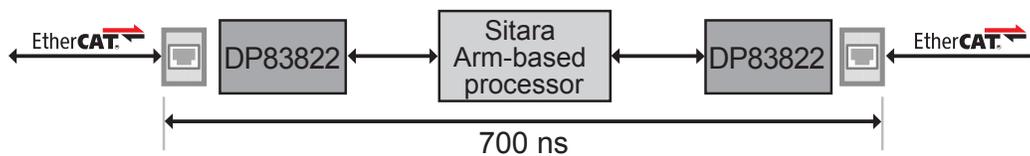


Figure 10. EtherCAT RX-TX latency.

## Key EtherCAT parameters

The key attributes of an EtherCAT slave implementation on the Sitara processors are provided in the [EtherCAT firmware datasheet](#).

## Easy EtherCAT integration

TI has streamlined the process of integrating EtherCAT with the Sitara processors. All the tools and software code required to integrate EtherCAT slaves are available as part of these processors' software development kits (SDK). On each development platform, the SDK includes firmware for the EtherCAT protocol, software drivers, hardware initialization routines, adaptation layer for the stack API, EtherCAT protocol stack and the application itself. The supporting documentation with the SDK enables one to modify and build new features into the application.

To facilitate the integration of the EtherCAT protocol stack, TI has also closely collaborated with Beckhoff Automation to validate EtherCAT Slave Stack Code on the Sitara processors. The Beckhoff code has been adapted to work on the Sitara processors and it has been tested to ensure that the integration is seamless for customers. Customers are expected to become ETG members (required to market EtherCAT products) and get entitled to obtain a free copy of the Beckhoff stack directly via the ETG website before taking their product to market. A copy of the EtherCAT stack from Beckhoff is also included in the Processor SDK for evaluation, development and test purposes.

For a typical use case, the EtherCAT firmware, the stack, the drivers and the high-level operating system (if needed) or a real-time OS kernel are all reused from the respective software development kit. There is usually only one file to be modified by the user when the user application is being developed.

## Power consumption

EtherCAT implementations on Sitara devices benefit from a low-power Arm core and system architecture, which eliminates the need for a fan or heat sink.

For instance, in most use cases, the peak power of the AM335x processor is under 1 W. For EtherCAT applications, the power consumption is less than 1 mW per MHz of Arm CPU speed.

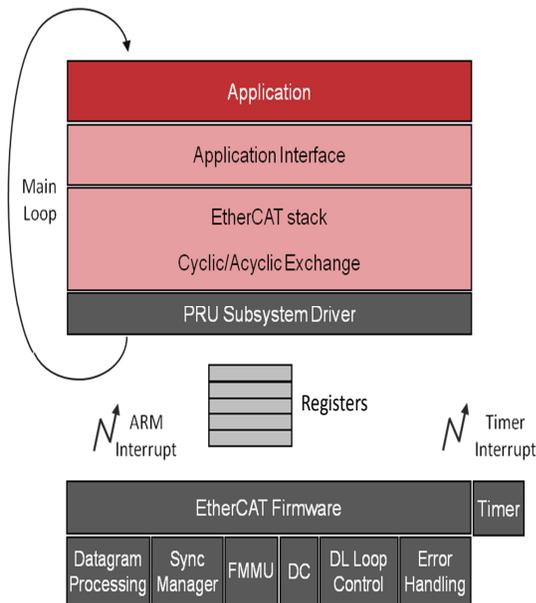


Figure 11. EtherCAT software integration.

## Integrating EtherCAT on end products

In order to integrate EtherCAT slave into industrial equipment, customers can use TI's EtherCAT slave implementation and complete their design process using the evaluation copy of the EtherCAT Slave Stack Code provided in the Processor SDK. The Slave Stack Code is originally obtained from Beckhoff and it is available to all ETG members for no charge.

Customers can also use a slave stack from a different vendor or develop their own. The customer should use Conformance Test Tool to pass all tests. Optionally, they can then get the product certified by EtherCAT certification labs and may also perform broader interoperability tests at the EtherCAT plug fests.

## Development tools for EtherCAT implementation

TI provides Evaluation Module (EVM) development platforms for its Sitara processors with comprehensive design data to assist customers with their implementations. All design data for these EVMs such as schematics and layout is available for accelerating development of customer designs. For more information on the tools available for specific processors, [click here](#).

In addition, TI also collaborates with external vendors for an additional development platform targeted for industrial applications.

## Summary

TI offers integrated EtherCAT slave and master capability on Sitara processors targeted for industrial I/O, sensor, PLC and human machine interface (HMI) systems. The integration of EtherCAT with a powerful yet low-power Arm core results in lower-cost end products without compromise on the functional or operational requirements. TI also offers the transceivers with built-in isolation for the industrial communication interfaces such as EtherCAT, PROFIBUS, CAN, RS-485 and more. With comprehensive software and hardware development tools, worldwide support and an active E2E™ developer community, customers can look forward to greatly simplified EtherCAT integration with the added benefit of significant cost savings – as much as 30 percent!

**Important Notice:** The products and services of Texas Instruments Incorporated and its subsidiaries described herein are sold subject to TI's standard terms and conditions of sale. Customers are advised to obtain the most current and complete information about TI products and services before placing orders. TI assumes no liability for applications assistance, customer's applications or product designs, software performance, or infringement of patents. The publication of information regarding any other company's products or services does not constitute TI's approval, warranty or endorsement thereof.

The platform bar and Sitara are trademarks of Texas Instruments. All other trademarks are the property of their respective owners.

## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), 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, 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 ([www.ti.com/legal/termsofsale.html](http://www.ti.com/legal/termsofsale.html)) or other applicable terms available either on [ti.com](http://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.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2020, Texas Instruments Incorporated