

TI Designs: TIDEP-0105

AMIC110上のDDRレスEtherCAT®スレーブのリファレンス・デザイン



概要

EtherCAT® (Ethernet for Control Automation Technology)は成長を続け、産業用の主要なイーサネット・ネットワークとしての地位を固めつつあります。このDDRレスEtherCATリファレンス・デザインは、マルチプロトコル産業用通信SoC (システム・オン・チップ)であるAMIC110に、低コストのDDRレスEtherCATスレーブを実装する際のリファレンス・デザインとして活用できます。また、このリファレンス・デザインは、フル機能のEtherCATスレーブ・スタック全体をSoCの内部メモリで実行できる能力も示します。このリファレンス・デザインは、外付けのASICやDDRが不要になるため、システムの部品表(BOM)および基板面積を大幅に削減できます。さらに、通常はEtherCATで使用している外部メモリ転送が不要になった結果、ネットワーク接続型の産業用ドライブや通信モジュールのような各種アプリケーションでは、高速処理から多くの利点を得ることができます。

リソース

TIDEP-0105	デザイン・フォルダ
AMIC110	プロダクト・フォルダ
DP83822H	プロダクト・フォルダ
PRU-ICSS産業用SW	プロダクト・フォルダ



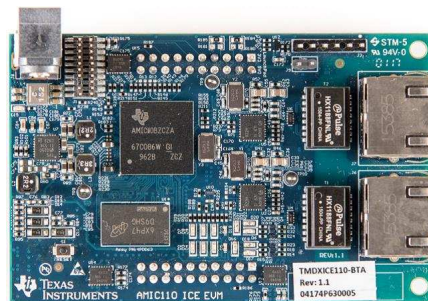
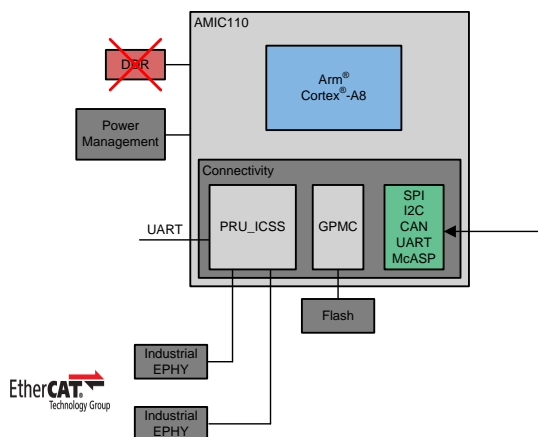
[E2E™エキスパートに質問](#)

特長

- EtherCAT Technology Group (ETG)のEtherCATスレーブ準拠性テスト・ツール(CTT)に合格
- EtherCATスレーブ・スタック全体を内部メモリでホスト
- PRU-ICSSファームウェアにより、8つのフィールドバス・メモリ管理ユニット(FMMU)と同期マネージャ(SM)をサポート
- SYNC0/SYNC1を配信クロック(DC)付きで生成
- 拡張されたリンクロス検出によるループ制御
- 外部メモリ・アクセスに関連するレイテンシが除去されるため、システム性能が向上
- オプションとして、C2000™ MCU、TMS320F28379Dと接続し、低コスト、高性能の産業用ドライブ・ソリューションを提供可能

アプリケーション

- 産業用ロボット通信モジュール
- CPU (プログラマブル・ロジック・コントローラ)
- 通信モジュール
- ACドライブの有線通信とワイヤレス通信
- サーボ・ドライブの有線通信とワイヤレス通信





使用許可、知的財産、その他免責事項は、最終ページにあるIMPORTANT NOTICE(重要な注意事項)をご参照くださいますようお願いいたします。英語版のTI製品についての情報を翻訳したこの資料は、製品の概要を確認する目的で便宜的に提供しているものです。該当する正式な英語版の最新情報は、www.ti.comで閲覧でき、その内容が常に優先されます。TIでは翻訳の正確性および妥当性につきましては一切保証いたしません。実際の設計などの前には、必ず最新版の英語版をご参照くださいますようお願いいたします。

1 System Description

EtherCAT, invented by Beckhoff Automation in Germany and later standardized by the ETG, is a real-time, industrial, Ethernet standard for industrial automation applications, such as input/output (I/O) devices, communication modules, sensors, and programmable logic controllers (PLCs).

Traditional Ethernet has seen unparalleled adoption in diverse applications, but in industrial environments it is still not efficient enough for small amounts of data exchange, due to its lower determinism for real-time operation and also works in which the network nodes must be connected through switches. 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 the master is written by the slave onto the frame as it passes through.

Many simple EtherCAT devices such as digital I/Os can be created using single FPGA or ASIC solutions available today. In 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 application-level processing. The cost of such architecture is higher than that of simple digital I/O devices, but it comes with flexibility in that developers can select a processor that suits their needs. In yet another approach, the EtherCAT implementation is one of the peripherals in the device that has an integrated CPU. Many FPGA devices can configure a processor in the FPGA or already have an integrated processor. The FPGAs are flexible, but depending on the CPU selection there is a risk that costs or operating frequency targets will be challenging to meet.

To meet the demand of cost-sensitive, industrial automation applications, this TI Design presents a reference design for a completely new, compact implementation that provides a low-cost, DDR-less, EtherCAT Slave with the AMIC110, a multiprotocol programmable industrial communications SoC. Significant system BOM and board savings are achieved with the solution by eliminating an external ASIC and DDR. In addition, the software- and firmware-based architecture and the PRU-ICSS Industrial Communications suite can scale to support multiple industrial Ethernet and fieldbus communication standards.

1.1 Key System Specifications

表 1. Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
EtherCAT commands	NOP, ARPD, APWR, APRW, FPRD, FPWR, FPRW, BRD, BWR, BRW, LRD, LWR, LRW, ARMW and FRMW	All supported
Number of ports	2 MII ports	Connection between the PHY (DP83822) and MAC (AMIC110)
Number of FMMUs and SMs	Up to 8	Fieldbus memory management unit and sync managers
Process data RAM	8KB	From PRU shared RAM
Distributed clock	Yes	Supports SYNC0, SYNC1, LATCH0 and LATCH1 signals
Conformance test	Pass	3.2.2.2

2 System Overview

2.1 Block Diagram

Figure 1 shows the industrial protocol software architecture built with programmable real-time unit (PRU) technology on the AMIC110, which aligns with other Sitara™ family processor-based industrial application products. A highlighted feature of the Industrial Ethernet protocol EtherCAT on the AMIC110 device is that it is DDR-less, which is achieved by deep optimization from the bootloader to protocol drivers.

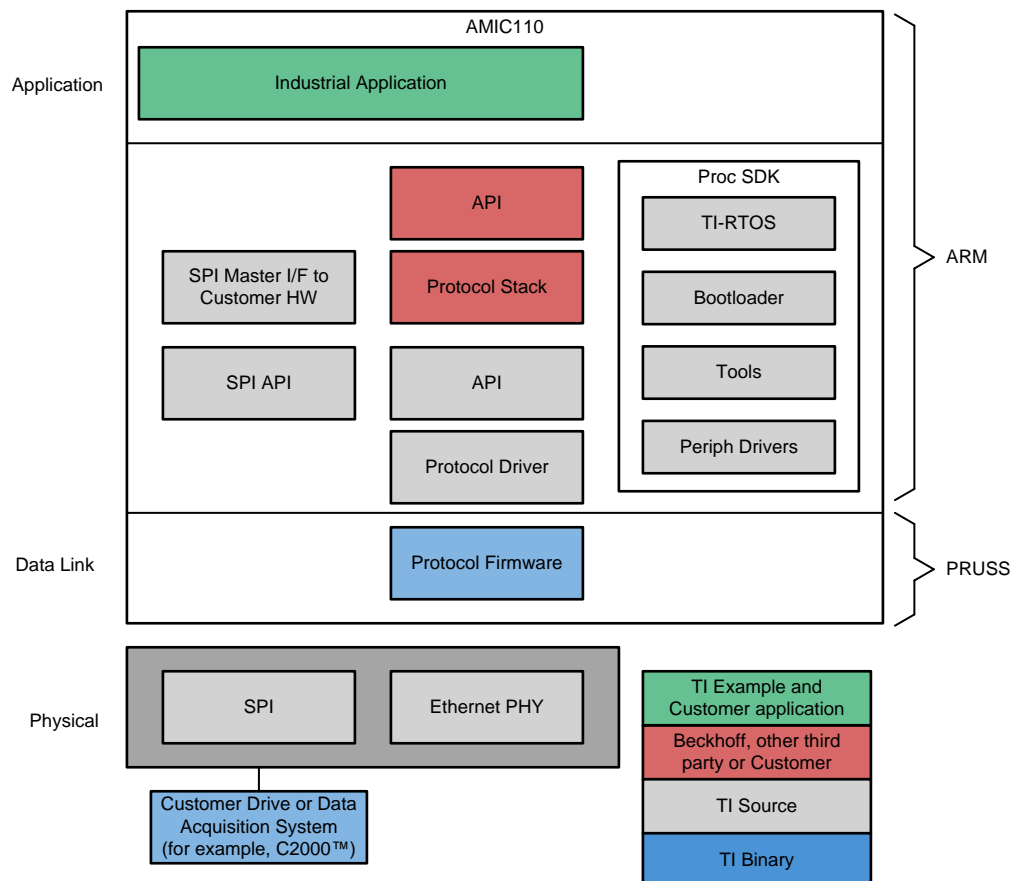


Figure 1. TIDEP-0105 Software Architecture Block Diagram

Three major software components, the physical layer, data link layer, and application layer, comprise the EtherCAT slave implementation on the AMIC110 device, similar to other Sitara processors from TI.

The first component is the physical layer, PHY, which provides all physical layer functions needed to transmit and receive data over standard twisted-pair cables. AMIC110 processors with the DP83822 Ethernet PHY device, as shown in Figure 2, the AMIC110 ICE hardware block diagram, exhibit a low latency.

In EtherCAT layer 2, the data link, the PRU real-time cores execute the tasks of datagram processing, distributed clocking, address mapping, error detection and handling, and host interface. PRUs also emulate the EtherCAT register space in the internal shared memory. With their deterministic real-time processing capability, the PRUs handle EtherCAT datagrams with consistent and predictable processing latency.

The third component is the EtherCAT slave stack, which runs on the Arm® processor and industrial application that is dependent on the end equipment in which this solution is used. For the application layer connection, different process data interfaces (PDI) are available. Typical interface options vary from 32-bit to 8- or 16-bit parallel I/O interfaces or serial interfaces like SPI.

Additional supporting components, such as the protocol adaptation layer and device drivers, are provided in the Processor SDK from TI, a unified software platform for TI-embedded processors.

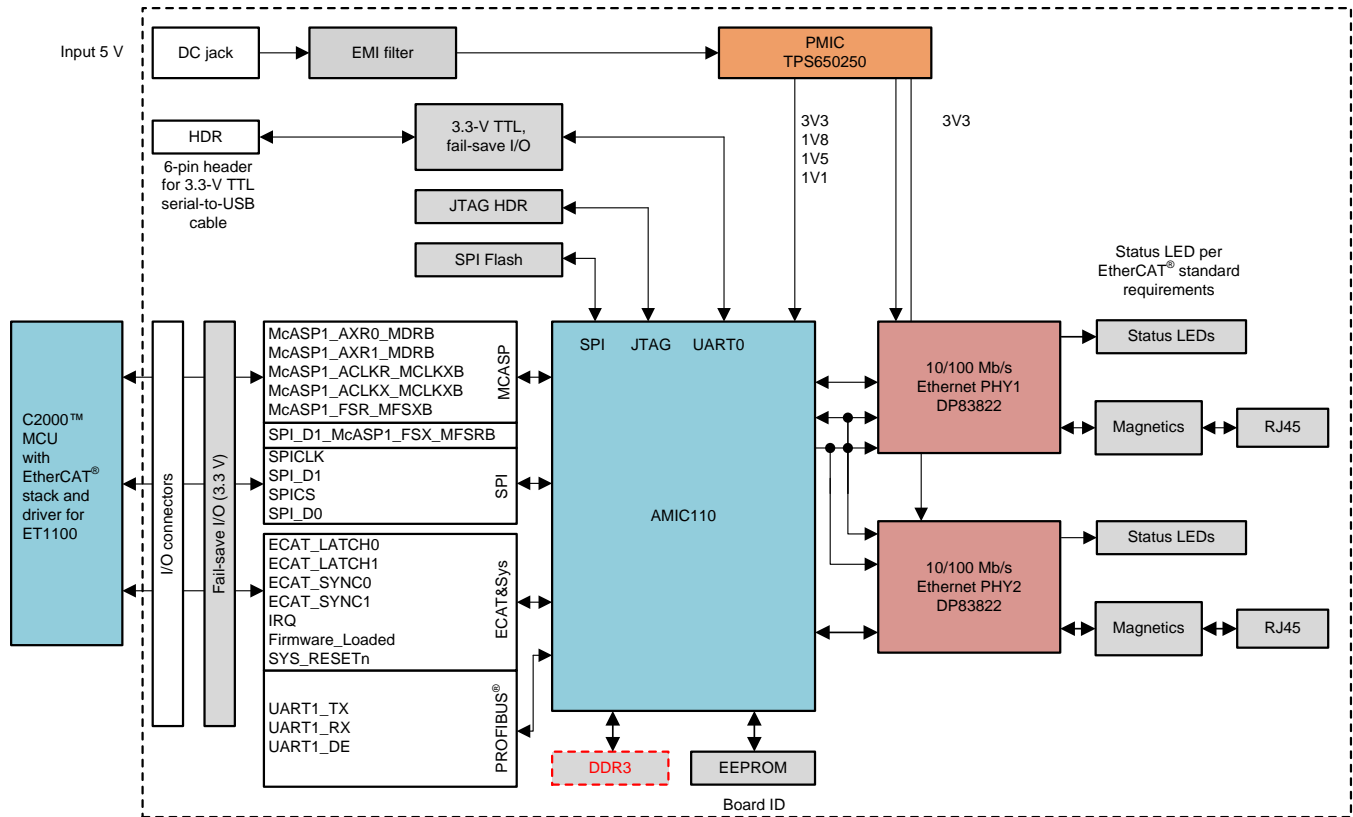


図 2. AMIC110 Industrial Communication Engine

2.2 Highlighted Products

2.2.1 AMIC110 Sitara™ SoC

The AMIC110 device is a multiprotocol, programmable, industrial communications processor, which provides ready-to-use solutions for most industrial Ethernet and fieldbus communications slaves, as well as some masters. The device is based on the Arm Cortex®-A8 processor, peripherals, and industrial interface options. The AMIC110 microprocessor is an ideal companion-communications chip to the C2000 family of microcontrollers for connected drives.

Figure 3 shows the subsystems contained in the AMIC110 microprocessor. The microprocessor unit (MPU) subsystem is based on the Arm Cortex-A8 processor. The PRU-ICSS is separate from the ARM core, allowing independent operation and clocking for greater efficiency and flexibility. The PRU-ICSS enables additional peripheral interfaces and real-time protocols such as EtherCAT, PROFINET IRT, EtherNet/IP™, PROFIBUS, Ethernet Powerlink, Sercos III, and others. Additionally, the programmable nature of the PRU-ICSS, along with its access to pins, events, and all SoC resources, provides flexibility in implementing fast real-time responses, specialized-data handling operations, custom peripheral interfaces, and offloading tasks from the other processor cores of the SoC.

Figure 3 shows the AMIC110 microprocessor functional block diagram.

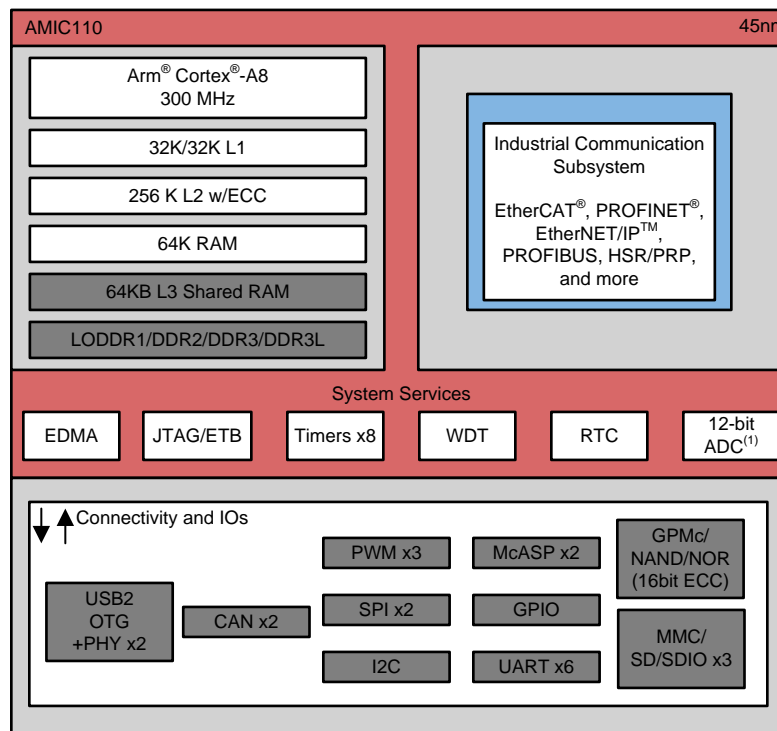


Figure 3. AMIC110 Microprocessor Functional Block Diagram

Key features:

- Up to 300-MHz Sitara, ARM Cortex-A8, 32-bit RISC processor:
 - NEON™ single instruction multiple data (SIMD) coprocessor
 - 32KB of L1 instruction and 32KB of data cache with single-error detection (parity)
 - 256KB of L2 cache with error correcting code (ECC)
 - 176KB of on-chip boot ROM
 - 64KB of dedicated RAM
 - Interrupt controller (up to 128 interrupt requests)
- On-chip memory (shared L3 RAM):
 - 64KB of general-purpose, on-chip memory controller (OCMC) RAM
 - Accessible to all masters
- Industrial communication subsystem (PRU-ICSS):
 - Supports protocols such as EtherCAT, PROFIBUS, PROFINET, EtherNet/IP, and more
 - Two PRUs
 - 32-bit load/store RISC processor, capable of running at 200 MHz
 - 8KB of instruction RAM with single-error detection (parity)
 - Single-cycle, 32-bit multiplier with 64-bit accumulator
 - Enhanced GPIO module provides shift-in/out support and parallel latch on the external signal

2.2.2 DP83822 Ethernet Physical Layer Transceiver

The DP83822 is a low-power, single-port, 10/100 Mbps, Ethernet PHY. The DP83822 provides all the physical layer functions needed to transmit and receive data over both standard twisted-pair cables or connecting to an external fiber optic transceiver. Additionally, the DP83822 device provides flexibility to connect to a MAC through a standard media independent interface (MII), reduced media-independent interface (RMII), or reduced gigabit media independent interface (RGMII). The DP83822 device offers integrated cable diagnostic tools, built-in self-test (BIST) and loopback capabilities for ease of use. The DP83822 device supports multiple industrial buses with fast link-down timing as well as Auto-MDIX in forced modes.

Figure 4 shows a simplified schematic of the DP83822.

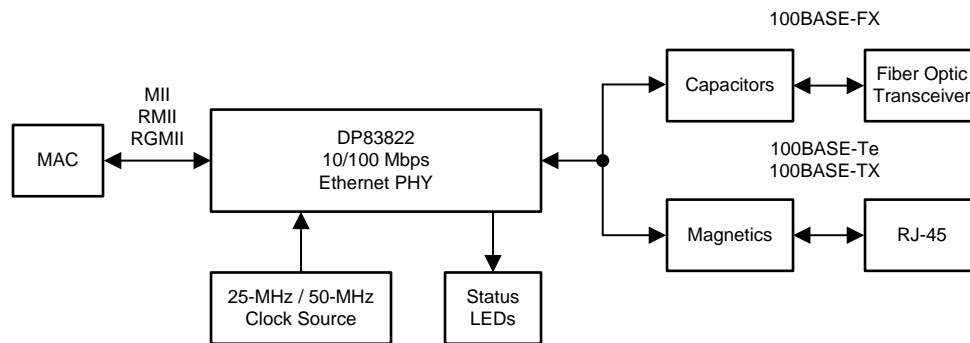


Figure 4. DP83822 Simplified Schematic

Key features:

- IEEE 802.3u compliant: 100BASE-FX, 100BASE-TX, and 10BASE-T
- MII, RMII, and RGMII MAC Interfaces
- Low-power, single-supply options: 1.8-V AVD < 120 mW, 3.3-V AVD < 220 mW
- Start of frame detect for IEEE 1588 time stamp
- Fast link-down timing
- Auto-crossover in forced modes
- BIST
- MDC / MDIO interface

2.3 System Design Theory

To implement the EtherCAT Slave DDR-less system on the AMIC110 with limited on-chip memory, many optimization techniques are applied. The optimization ranges from the AMIC110 second boot loader (SBL) to the Processor SDK RTOS drivers and EtherCAT protocol. Table 2 lists the available on-chip memory in the AMIC110 device.

Table 2. On-Chip Memory in AMIC110

TYPE	START ADDRESS	SIZE	DESCRIPTION
SRAM	0x40200000	0x00010000	64KB internal SRAM
L3 OCMC	0x40300000	0x00010000	64KB L3 OCMC SRAM
M3 SHUMEM	0x44D00000	0x00004000	16KB M3 shared unified code space
M3 SUDMEM	0x44D80000	0x00002000	8KB M3 shared data memory

The process of integrating EtherCAT with the Sitara processors has been streamlined. All the tools and software code required to integrate EtherCAT slaves are available as part of the processor SDK RTOS and PRU-ICSS Industrial SDK. The SDKs include the PRU-ICSS firmware, software drivers, hardware initialization routines, adaptation layer, EtherCAT protocol stack, and the application.

2.3.1 SBL Optimization

The SBL sets up the PLL clocks, powers on the I/O Peripherals, initializes the DDR, loads the application image into DDR, and brings the slave cores out of reset for applicable SoCs. A variety of boot modes such as QSPI, UART, MMCSD, NAND, and MCSPI are available in typical Sitara devices. The SBL uses part of the internal memory that has to be reduced to allow the EtherCAT stack and application to fit in.

2.3.1.1 Reducing SBL Size

Earlier versions of the AM335x/AMIC110 SBL in the Processor SDK RTOS for MCSPI boot mode: 34KB (release) and 57KB (debug).

The following changes have been implemented to reduce the size of the SBL for this use case:

- Bypass or remove SBL code and data for DDR setup
- Remove board and SoC detect functions to make the SBL specific for the AMIC110
- Remove console utilities to allow direct application boot

Implementation:

- Setup using the build option USE_DDR to configure the DDRLESS option in the source code
- Implementation generates cut-down version of board, utilities, device and bootloader components

2.3.1.2 Wakeup PRU

The EtherCAT application contains two, 8KB, constant arrays which contain the PRU firmware that gets copied into the OCMC memory and gets copied into the PRU IRAM memory. To avoid the copy, a mechanism was implemented in the SBL to load the application from the flash to PRU0 and PRU1 IRAM, which also avoids use of an additional 16KB of memory in the OCMC. To load the PRU0 and PRU1 IRAMs, the Arm needs to enable the PRU using PRCM.

Implementation:

- Setup build option ENABLE_PRU to perform the PRU wake-up sequence in the bootloader code.
- The option wakes up PRU0 and PRU1 and with flush data RAM

2.3.1.3 Storing and Loading Copy of TIESC EEPROM Data and PRU IRAM Binaries

The EtherCAT application has a requirement to save TIESC EEPROM data in nonvolatile memory that can be used to restore the setup. This data is also stored as a constant in the application and copied, to be used by the EtherCAT application during normal operation. This data was moved to the SPI flash and loaded in by the bootloader for the application to consume.

Implementation:

- Put binary images into TI Image format and load them based on the configuration file that specifies the number of binaries to load and the offset from which they are read into the device memory.
- Customization allows for loading of PRU firmware in IRAM and DATA RAM and also additional industrial constant arrays, like TIESC, into device memory

Figure 5 shows the SBL and EtherCAT memory map on the AMIC110 device. The .bss section of the EtherCAT application is designed to overlap with the SBL for maximum use of on-chip memory, because the .bss section is loaded only after the SBL has loaded the application and exited.

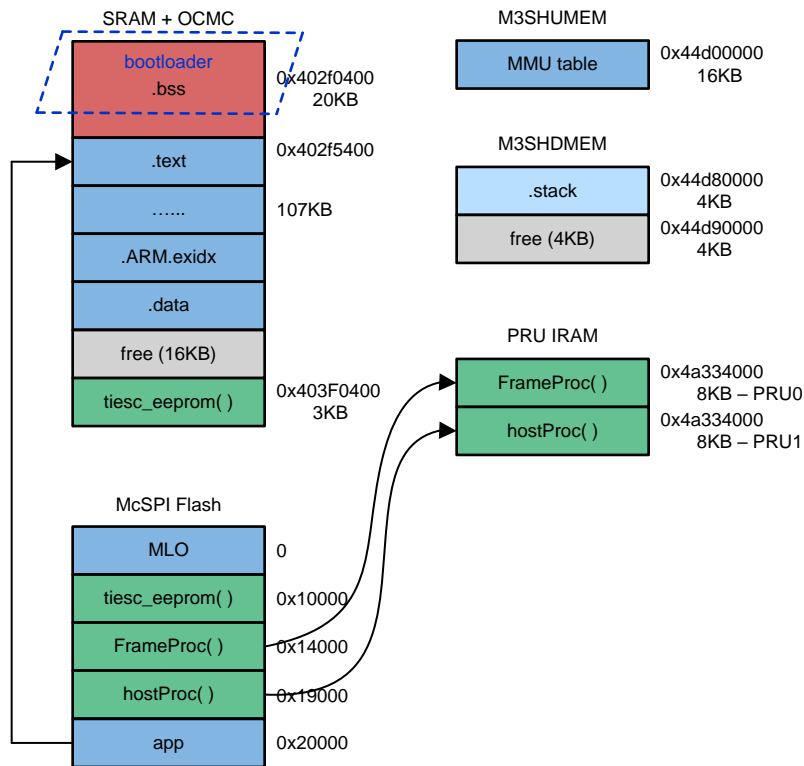


Figure 5. SBL and EtherCAT® Memory Map on AMIC110

2.3.2 Host Driver and Protocol Optimization

2.3.2.1 Memory Use Analysis

Using the object dump tool (arm-none-eabi-objdump.exe) from Arm GCC, the memory sections of the EtherCAT slave can be retrieved from the executable ARM binary (see Table 3).

Table 3. Memory Sections From Arm® GCC Compiler

LOADED SECTION	UNLOADED SECTION
.c_int00	.comment
ti.sysbios.family.arm.a8.mmuTableSection	.ARM.attributes
xdc.meta	.debug_aranges
.text	.debug_info
.rodata	.debug_abbrev
.vectors	.debug_line
.ARM.exidx	.debug_frame
.data	.debug_str
.bss	.debug_loc
.stack	.debug_ranges

These sections can be categorized into two types during code execution – loaded and unloaded. The loaded sections are the areas being investigated.

2.3.2.2 Optimization Techniques

A number of techniques were used to reduce the EtherCAT application fit into the AMIC110 on-chip memory.

1. The first step is to eliminate code support for unused features of the device, the ICE board, and in TI-RTOS. For example, in this application the I²C interface, SPI write, and the UART/ text/ printf support could be eliminated in both the application and in RTOS.
2. The next step is to minimize the RTOS debug and error handling components to what is necessary for the runtime application. Here, the error handlers and exception stack size are reduced, and unnecessary functions such as RTOS logging, RTS Thread protection, and stack overrun checking are eliminated.
3. The third step is to eliminate any unnecessary operations and optimize the remaining functions. For example, because the application runs only on a dedicated hardware configuration, the device configuration is static and does not need to be read from an EEPROM. Reduce functional representations to an optimum size. This reduction can be done by using bit arrays in place of word arrays. Optimize SYSBIOS to use a custom code configuration of just the necessary components. Evaluate the Stack and Heap use and set their reserved sizes to optimum values.
4. The last step is to compile the code and link the code with the appropriate settings to minimize the executable size and pack the executable into the available memory regions. These optimizations follow:
 - Use Thumb mode.
 - Optimize for size -Os.
 - Use NEON and hardware floating point.
 - Place each function in its own section.
 - Place data items in their own sections.
 - Disable all debugging information.
 - Enable global GNU optimizations, -flto, -fuse-linker-plugin.
 - Separate portions of the program and data to achieve good memory use.
 - Partition each relatively self-contained code block into sections, because external section references require additional program memory and cycles.

3 Hardware, Software, Testing Requirements, and Test Results

3.1 Required Hardware and Software

3.1.1 Hardware

- [AMIC110 ICE](#)
- [XDS200 JTAG emulator](#)
- Ethernet cable
- Power supply: 5 V, DC $\pm 10\%$ at 1.2 A

3.1.2 Software

- [CCSv7.3.0.19](#) – Code Composer Studio™
- [TwinCAT 3.1](#) – eXtended Automation Engineering (XAE) from ETG
- [EtherCAT stack version 5.11](#) from ETG
- [PROCESSOR-SDK-RTOS-AM335X 04_01_00_06](#)
- [PRU-ICSS-ETHERCAT-SLAVE v1.00.05](#)

3.2 Testing and Results

3.2.1 Test Setup

1. Set up the hardware as shown in [Figure 6](#). The other end of the Ethernet cable and XDS200 JTAG emulator connect to the PC.

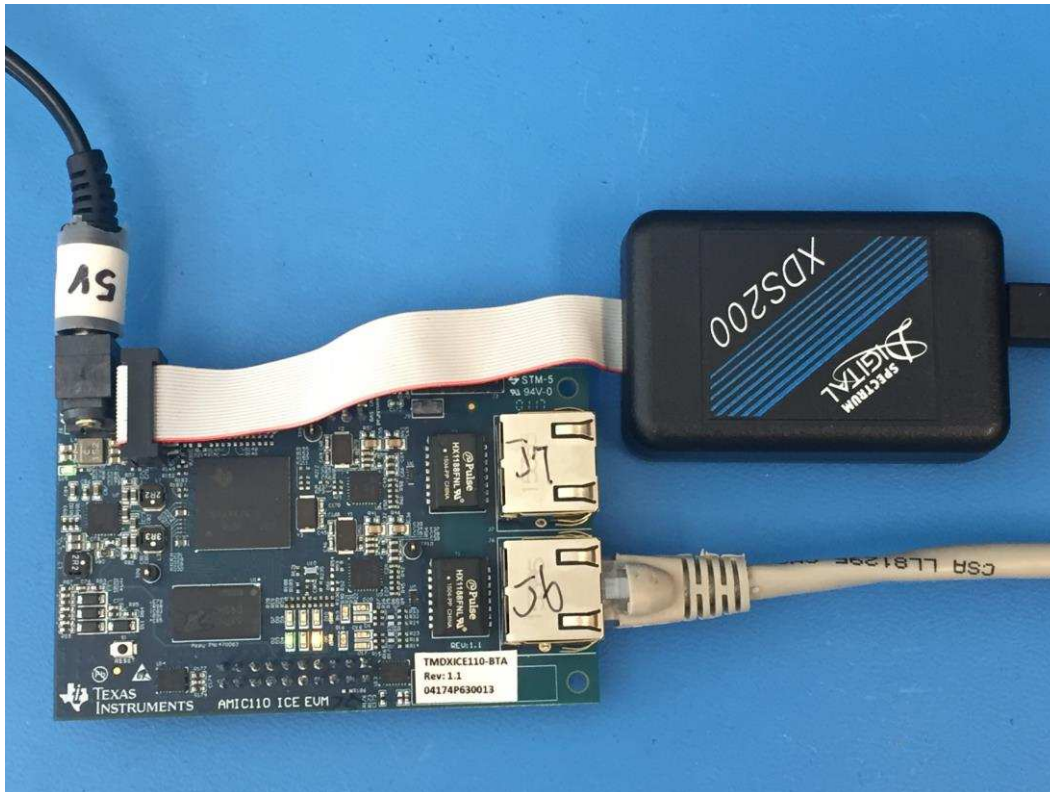


図 6. AMIC110 ICE Hardware Setup

2. Install the software listed in [3.1.2](#).
3. Follow the [PRU-ICSS EtherCAT user guide wiki](#) to create the full EtherCAT stack application.


```
projectCreate.bat AMIC11x arm ethercat_slave_full
```
4. Apply the PDK patch for Thumb mode. For the DDR-less EtherCAT application to build, it is critical that the Processor SDK is built in Thumb mode. A patch file is included in the PRU-ICSS-ETHERCAT-SLAVE v1.00.05 package at [INSTALL-DIR]/protocols/pdk_patches/04.01.00/AM335x_PDK_1_0_8_thumb_mode.patch. After applying this patch the Processor SDK PDK needs to be cleaned and rebuilt. Follow the [PDK rebuild procedure](#).
5. Follow the instructions in the [Building full feature EtherCAT Slave Application wiki](#) to build the full EtherCAT stack application.
6. Follow the instructions in the [On-chip Memory \(DDRless\) Execution of EtherCAT Slave Application wiki](#) to flash the bootloader, EtherCAT binaries, and application. During normal operation, users are required to flash the SBL in 0x000000 and the app at 0x20000. In DDR-less mode, the SBL (MLO) must load additional binaries to PRU0 IRAM and PRU1 IRAM and store TIESC EEPROM. The following flash memory map (see [Figure 7](#)) was set to flash the application binaries. Currently, the SBL (MLO) for DDR-less mode must be built from [Processor SDK v4.2](#), with [this patch](#). The patch is not required for Processor SDK RTOS v4.3 and later. A porting guide to rebase the PRU-ICSS-ETHERCAT-SLAVE v1.00.05 to Processor SDK v4.2 is in the *Processor SDK 4.2 Migration Guide of EtherCAT* section of the user's guide.

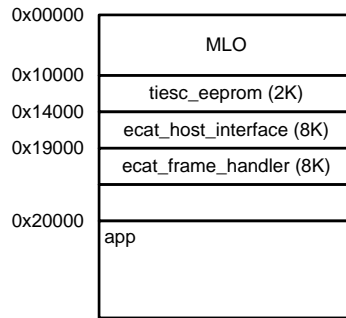


図 7. Flash Memory Map

7. Reboot the AMIC110 ICE after the binaries are flashed to the SPI flash memory.

3.2.2 Test Results

3.2.2.1 Detect DDR-Less EtherCAT® Slave Device With TwinCAT

Follow [these instructions](#) to set up the TwinCAT.

1. Launch the TwinCAT and create a new project, for example iceAMIC110, as shown in 図 8.

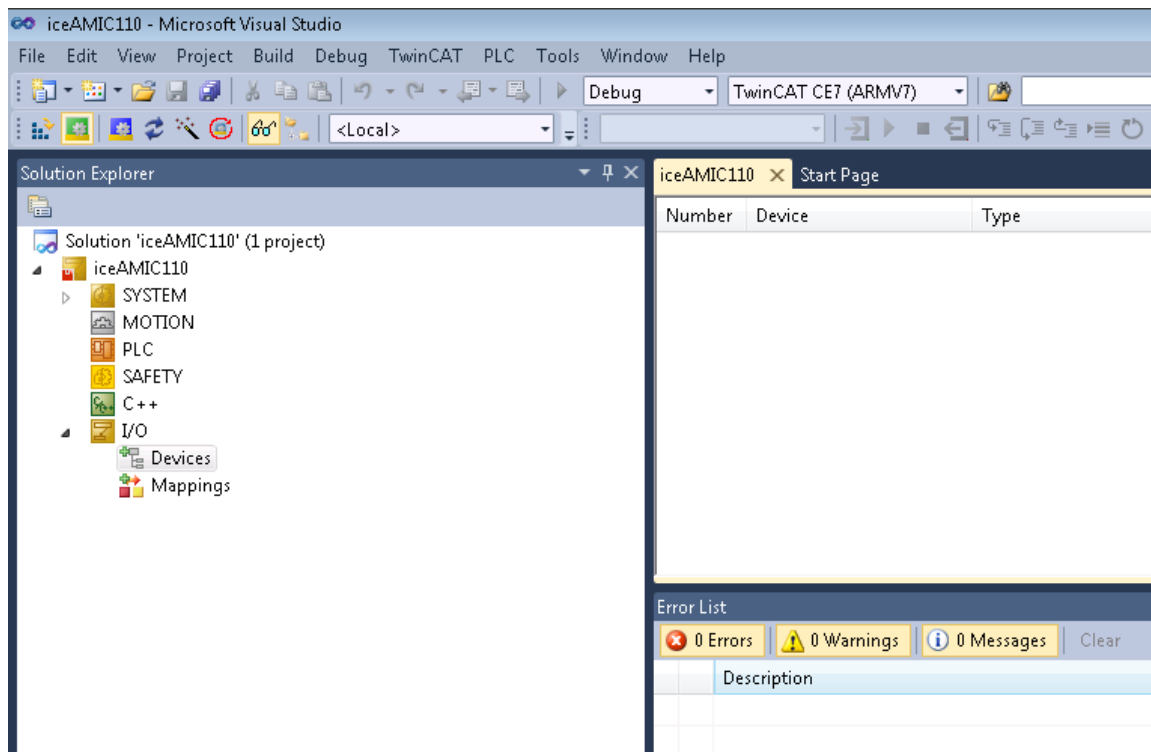


図 8. New Project in TwinCAT

2. Next:

- Scan the device and box, then click yes when the Active Free Run window prompts.
- Check that the TI box is online and the current status is OP, as shown in [Figure 9](#).

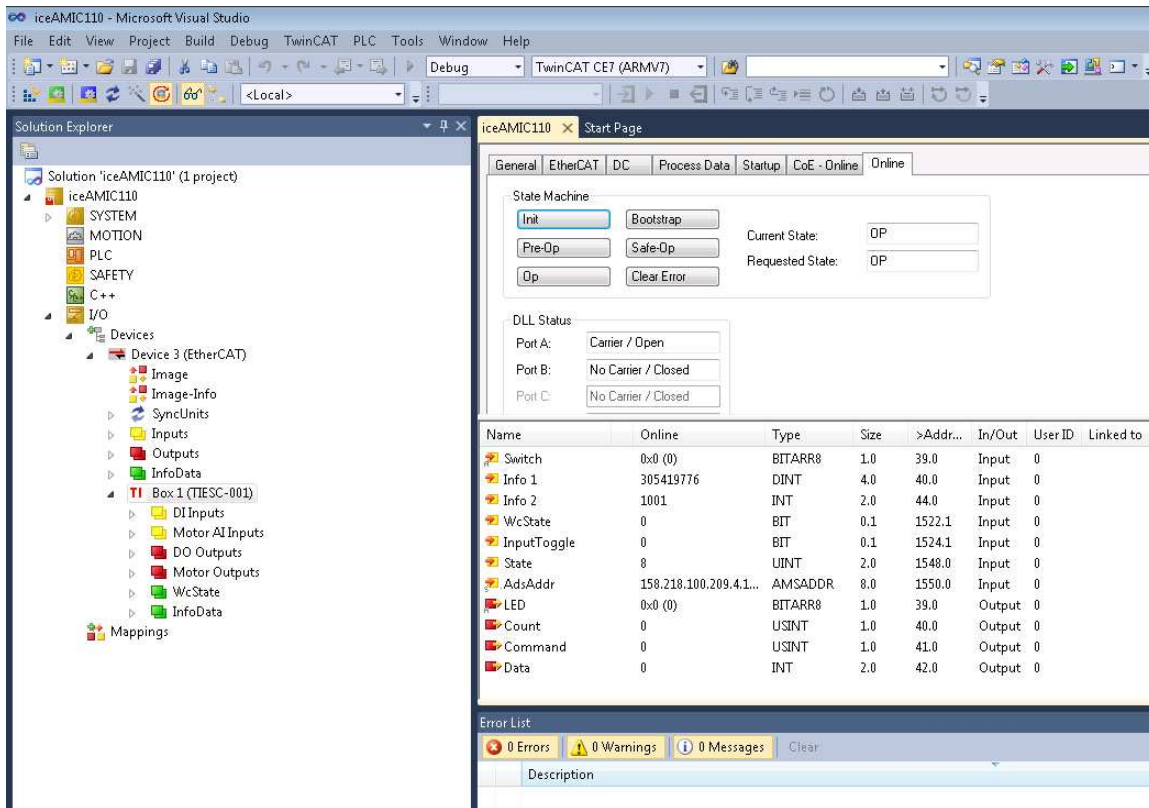


Figure 9. TIESC-001 in OP State

3.2.2.2 Conformance Test

The DDR-less EtherCAT slave on the AMIC110 device successfully passes the CTT from ETG. [Figure 10](#) and [Figure 11](#) show zero errors when using the EtherCAT device.

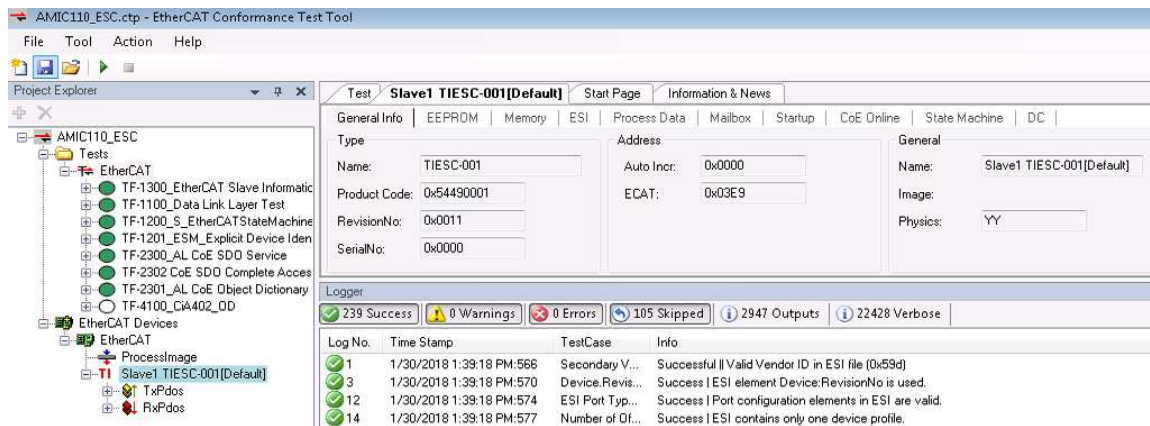


Figure 10. EtherCAT® Conformance Test Result

Type	VendorId	Product Code	RevisionNo	ESC Type/Revision/Build	Firmware Version
TIESC-001	0x0000059D	0x54490001	0x00000011	0x90/1/400	SII: 0 CoE: 5.11

Hardware Version	Errors	Warnings	Skipped	Succeeded	Details
0	0	0	105	239	Slave1_TestReport

Figure 11. EtherCAT® Conformance Test Report

4 Design Files

4.1 Schematics

To download the schematics, see the design files at [TIDEP-0105](#).

4.2 Bill of Materials

To download the bill of materials (BOM), see the design files at [TIDEP-0105](#).

4.3 PCB Layout Recommendations

4.3.1 Layout Prints

To download the layer plots, see the design files at [TIDEP-0105](#).

4.4 Altium Project

To download the Altium project files, see the design files at [TIDEP-0105](#).

4.5 Gerber Files

To download the Gerber files, see the design files at [TIDEP-0105](#).

4.6 Assembly Drawings

To download the assembly drawings, see the design files at [TIDEP-0105](#).

5 Software Files

To download the software files, see the design files at [TIDEP-0105](#).

6 Related Documentation

1. Texas Instruments, [EtherCAT® on Sitara™ Processors Marketing White Paper](#)
2. Texas Instruments, [EtherCAT® Slave and Multi-Protocol Industrial Ethernet Reference Design](#)
3. Texas Instruments, [AM335x and AMIC110 Sitara™ Processors Technical Reference Manual](#)
4. Texas Instruments, [DP83822 Robust, Low Power 10/100 Mbps Ethernet Physical Layer Transceiver Data Sheet](#)

6.1 商標

E2E, C2000, Sitara, Code Composer Studio are trademarks of Texas Instruments.

NEON is a trademark of Arm Limited (or its subsidiaries).

Arm, Cortex are registered trademarks of Arm Limited (or its subsidiaries).

EtherCAT is a registered trademark of Beckhoff Automation GmbH.

EtherNet/IP is a trademark of ODVA, Inc.

すべての商標および登録商標はそれぞれの所有者に帰属します。

7 About the Authors

GARRETT DING is a software applications engineer for the Embedded Processing Group at Texas Instruments, where he is responsible for developing reference design solutions and providing technical support to customers for the industrial segment. Garrett earned his master of science in electrical engineering (MSEE) from NanJing University of Science and Technology, China.

MANMOHAN MANDHANA is a software engineer for the Embedded Processing Group at Texas Instruments, where he is responsible for developing Industrial Communication/Fieldbus and Control protocols using PRU-ICSS technology available in the Embedded Processor Sitara and Keystone product lines. Manmohan earned his bachelor of technology in electrical engineering from the Indian Institute of Technology Bombay, India.

DAVID ZAUCHA is an applications engineer at Texas Instruments, where he is responsible for supporting customer applications in the Industrial Communications segment. David has been with TI since 1999 and has been involved in designing and supporting products in analog and embedded systems. David earned his bachelor of science (BSEE) at the University of Massachusetts and his MSEE at the University of Rochester.

RAHUL PRABHU is a software applications engineer for the Embedded Processing Group at Texas Instruments, where he is responsible for supporting customer applications using the Sitara and Keystone product lines. Rahul brings to this role his extensive experience and knowledge in RTOS application development and system integration. Rahul earned MS in electrical and computer engineering from the University of Houston.

TIの設計情報およびリソースに関する重要な注意事項

Texas Instruments Incorporated ("TI")の技術、アプリケーションその他設計に関する助言、サービスまたは情報は、TI製品を組み込んだアプリケーションを開発する設計者に役立つことを目的として提供するものです。これにはリファレンス設計や、評価モジュールに関係する資料が含まれますが、これらに限られません。以下、これらを総称して「TIリソース」と呼びます。いかなる方法であっても、TIリソースのいずれかをダウンロード、アクセス、または使用した場合、お客様(個人、または会社を代表している場合にはお客様の会社)は、これらのリソースをここに記載された目的にのみ使用し、この注意事項の条項に従うことに合意したものとします。

TIによるTIリソースの提供は、TI製品に対する該当の発行済み保証事項または免責事項を拡張またはいかなる形でも変更するものではなく、これらのTIリソースを提供することによって、TIにはいかなる追加義務も責任も発生しないものとします。TIは、自社のTIリソースに訂正、拡張、改良、およびその他の変更を加える権利を留保します。

お客様は、自らのアプリケーションの設計において、ご自身が独自に分析、評価、判断を行う責任がお客様にあり、お客様のアプリケーション(および、お客様のアプリケーションに使用されるすべてのTI製品)の安全性、および該当するすべての規制、法、その他適用される要件への遵守を保証するすべての責任をお客様のみが負うことを理解し、合意するものとします。お客様は、自身のアプリケーションに関して、(1) 故障による危険な結果を予測し、(2) 障害とその結果を監視し、および、(3) 損害を引き起こす障害の可能性を減らし、適切な対策を行う目的での、安全策を開発し実装するために必要な、すべての技術を保持していることを表明するものとします。お客様は、TI製品を含むアプリケーションを使用または配布する前に、それらのアプリケーション、およびアプリケーションに使用されているTI製品の機能性を完全にテストすることに合意するものとします。TIは、特定のTIリソース用に発行されたドキュメントで明示的に記載されているもの以外のテストを実行していません。

お客様は、個別のTIリソースにつき、当該TIリソースに記載されているTI製品を含むアプリケーションの開発に関連する目的でのみ、使用、コピー、変更することが許可されています。明示的または黙示的を問わず、禁反言の法理その他どのような理由でも、他のTIの知的所有権に対するその他のライセンスは付与されません。また、TIまたは他のいかなる第三者のテクノロジーまたは知的所有権についても、いかなるライセンスも付与されるものではありません。付与されないものには、TI製品またはサービスが使用される組み合わせ、機械、プロセスに関連する特許権、著作権、回路配置利用権、その他の知的所有権が含まれますが、これらに限られません。第三者の製品やサービスに関する、またはそれらを参照する情報は、そのような製品またはサービスを利用するライセンスを構成するものではなく、それらに対する保証または推奨を意味するものでもありません。TIリソースを使用するため、第三者の特許または他の知的所有権に基づく第三者からのライセンス、もしくは、TIの特許または他の知的所有権に基づくTIからのライセンスが必要な場合があります。

TIのリソースは、それに含まれるあらゆる欠陥も含めて、「現状のまま」提供されます。TIは、TIリソースまたはその仕様に関して、明示的か暗黙的にかかわらず、他のいかなる保証または表明も行いません。これには、正確性または完全性、権原、続発性の障害に関する保証、および商品性、特定目的への適合性、第三者の知的所有権の非侵害に対する黙示的保証が含まれますが、これらに限られません。

TIは、いかなる苦情に対しても、お客様への弁済または補償を行う義務はなく、行わないものとします。これには、任意の製品の組み合わせに関連する、またはそれらに基づく侵害の請求も含まれますが、これらに限られず、またその事実についてTIリソースまたは他の場所に記載されているか否かを問わないものとします。いかなる場合も、TIリソースまたはその使用に関連して、またはそれらにより発生した、実際の、直接的、特別、付随的、間接的、懲罰的、偶発的、または、結果的な損害について、そのような損害の可能性についてTIが知らされていたかどうかにかかわらず、TIは責任を負わないものとします。

お客様は、この注意事項の条件および条項に従わなかったために発生した、いかなる損害、コスト、損失、責任からも、TIおよびその代表者を完全に免責するものとします。

この注意事項はTIリソースに適用されます。特定の種類の資料、TI製品、およびサービスの使用および購入については、追加条項が適用されます。これには、半導体製品(<http://www.ti.com/sc/docs/stdterms.htm>)、評価モジュール、およびサンプル(<http://www.ti.com/sc/docs/sampterms.htm>)についてのTIの標準条項が含まれますが、これらに限られません。