

TI Designs

EtherCAT® Master Reference Design for AM335x CPSW



TI Designs

This TI Design details an EtherCAT® Master interface operating on the Sitara™ AM335x processor using the EC-Master stack from acontis. This EtherCAT Master solution can be used for EtherCAT-based PLC or motion control applications. EtherCAT Master is profiled on the Common Platform Ethernet Switch (CPSW) ports of the AM335x processor. This EtherCAT Master implementation on the AM335x platform can achieve cycle times of less than 100 μ s.

Design Resources

[TIDEP0043](#)

[TMDSICE3359](#)

[TMDSIDK437x](#)

[AM3359](#)

Tools Folder

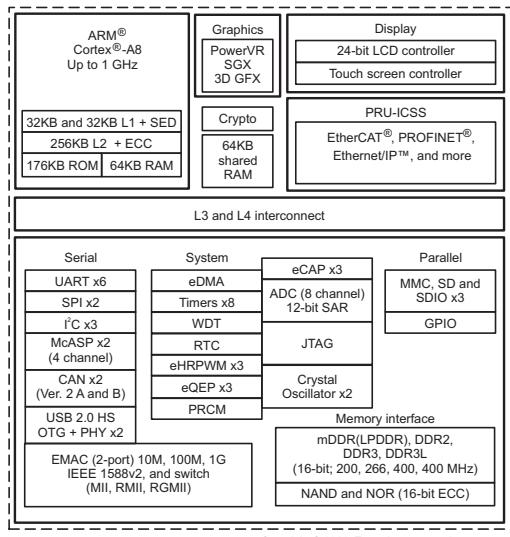
Tools Folder

Tools Folder

Product Folder



[ASK Our E2E Experts](#)

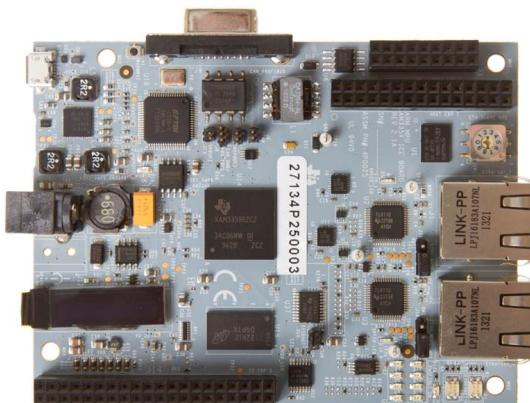


Design Features

- EtherCAT Class A or Class B Master Stack According to ETG.1500 Specification
- High-Performance CPSW Ethernet Driver for Maximum EtherCAT Performance
- EtherCAT Feature: Pack Cable Redundancy Using Two CPSW Ports or Two ICSS_PRU Ports
- EtherCAT Feature: Pack Hot Connect to Support Flexible Configuration
- Supports Various Operating Systems: Linux®, TI-RTOS (SYS/BIOS), StarterWare, VxWorks®, and QNX™

Featured Applications

- EtherCAT Programmable Logic Control (PLC) System
- EtherCAT Motion Control Application
- EtherCAT Interface Boards
- EtherCAT Industrial Communication Gateways



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

ARM, Cortex are registered trademarks of ARM Limited.

EtherCAT, TwinCAT are registered trademarks of Beckhoff Automation GmbH & Co. KG.

Linux is a registered trademark of Linus Torvalds.

Windows is a registered trademark of Microsoft Corporation.

QNX is a trademark of QNX Software Systems Limited.

VxWorks is a registered trademark of Wind River.

All other trademarks are the property of their respective owners.

1 Introduction

This TI Design document presents the TI Sitara AM335x implementing an EtherCAT Master using acontis EtherCAT Master stack (EC-Master). The acontis EtherCAT Master stack is a highly portable software stack, and when combined with a high-performance TI Sitara CPU it provides a sophisticated EtherCAT solution that users can use to implement EtherCAT communication-interface boards, EtherCAT-based PLC, or EtherCAT-based motion control applications.

The EC-Master architectural design does not require additional tasks, making it easier to transport code to a different OS or to bare-metal systems. Due to this architecture combined with the high-speed Ethernet driver, it is possible to implement applications on AM335x platform with short cycle times less than 100 μ sec.

2 EtherCAT® Protocol

EtherCAT is an IEEE 802.3 Ethernet-based fieldbus system. EtherCAT is a new standard in communication speed. Because EtherCAT is inexpensive to implement, the system can use fieldbus technology in applications which previously omitted fieldbus. EtherCAT is an open technology that is standardized within the International Electrotechnical Commission (IEC). The technology is supported and powered by the EtherCAT Technology Group (an international community of users and vendors). The protocol is suitable for both hard and soft real-time requirements in automation technology. A primary advantage of EtherCAT is that it supports automation applications that require short data-update times with low communication jitter and reduced hardware costs.

In the EtherCAT protocol, the EtherCAT Master sends a telegram that passes through each node. Each EtherCAT Slave device reads the data that is addressed to it as soon as the data is detected. Then, the slave device inserts the data into the frame as the frame moves downstream. The frame is delayed by hardware-propagation delay times. The last node in a segment (or branch) detects an open port and sends the message back to the master using the full-duplex feature of Ethernet technology. The EtherCAT Master is the only node within a segment that actively sends an EtherCAT frame. All other nodes forward the frames downstream. This capability permits the network to achieve over 90% of the available network bandwidth, prevents unpredictable delays, and guarantees real-time system response.

The EtherCAT protocol is optimized for process data transfer and is transported within the Ethernet frame by using a special Ethertype identifier (0x88A4). EtherCAT communications consist of several EtherCAT telegrams. Each telegram serves a specific memory area of the logical process image, up to 4GB. The data sequence is independent of the physical order of the Ethernet terminals in the network.

In addition to data exchanges between the EtherCAT Master and Slave, EtherCAT is also suitable for communication between controllers (master to master). Freely addressable network variables for process data and a variety of services for parameterization, diagnosis, programming, and remote control cover a range of requirements. The data interfaces for master-to-slave and master-to-master communication are identical.

Two mechanisms are available for slave-to-slave communication. The first option is to use upstream devices to quickly communicate to downstream devices within the same cycle. The other option is using the freely configurable slave-to-slave communication that runs through the master device. The second option requires two bus cycles (although not necessarily two control cycles).

3 Block Diagram

3.1 TI Sitara™ Overview

The Sitara AM335x processor is low-power device based on the ARM® Cortex®-A8 RISC. This system on a chip (SoC) features a broad range of integrated peripherals that are suitable for industrial applications. Sitara processors support multiple operating-frequency ranges from 300 MHz for simple applications and up to 1 GHz for more complex high-performance applications. The AM335x processor is configured with one Programmable Real-Time Unit (PRU) coprocessor (two real-time cores). This PRU can be used for communication protocols such as EtherCAT Master and Slave, PROFINET, Ethernet/IP and Sercos. In this TI Design, EtherCAT Master uses an AM335x Common Platform Ethernet Switch (CPSW) which frees the PRU for other protocols or applications.

Figure 1 shows the functional block diagram.

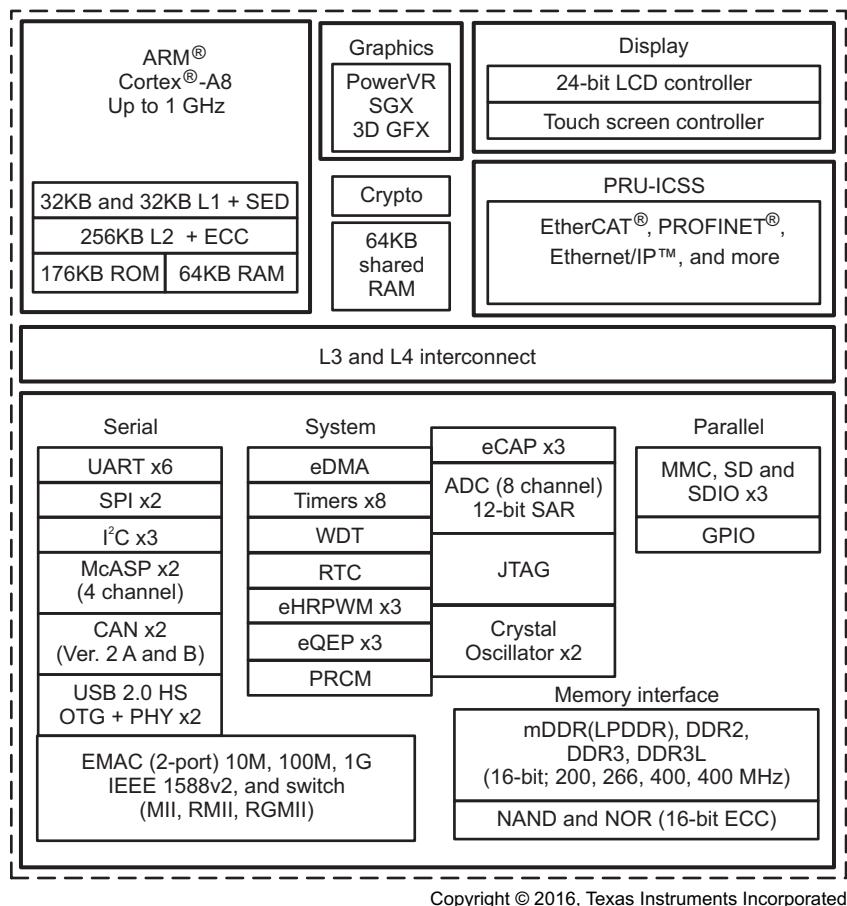


Figure 1. AM335x Block Diagram

4 acontis EtherCAT® Master Architecture

Figure 2 shows the module architecture for the acontis EC-Master. The EC-Master stack is divided into the following modules:

- EtherCAT Master Core
 - In the core module, cyclic (process data update) and acyclic (mailbox) EtherCAT commands are sent and received.
- Configuration Layer
 - The EtherCAT Master is configured using an XML file whose format is fixed in the EtherCAT specification ETG.2100. The EC-Master contains an OS-independent XML parser.
- Ethernet Link Layer
 - This layer exchanges Ethernet frames between the master and the slave devices.
- OS Layer
 - All OS-dependent system calls are encapsulated in a small OS layer.

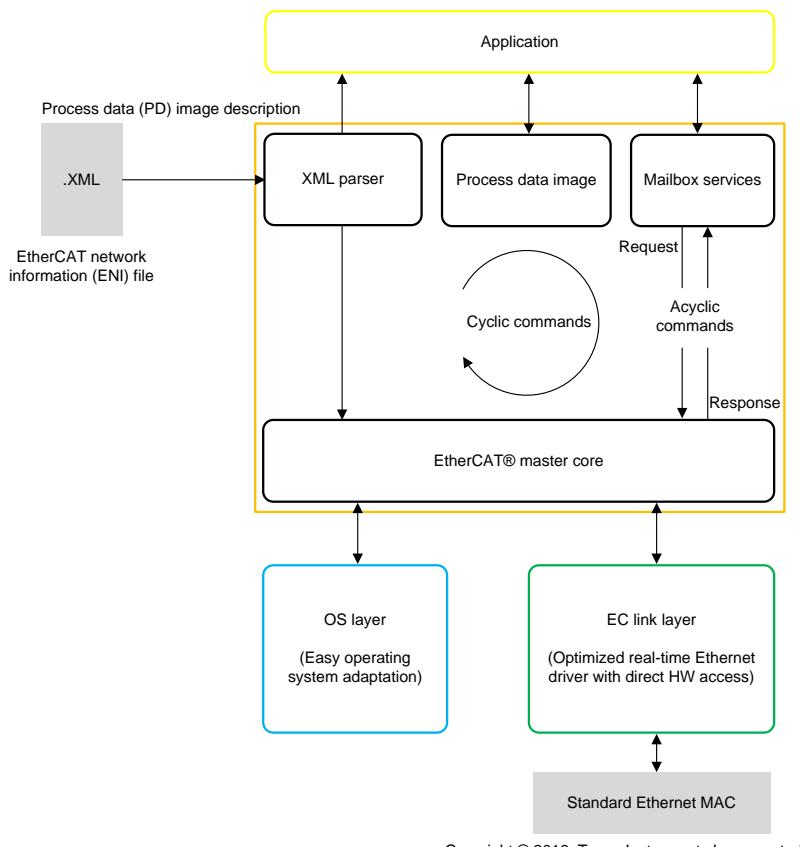


Figure 2. acontis EC-Master Component Model Architecture

5 Getting Started

5.1 Hardware

The following list shows the hardware required to support this design.

- TMDSICE3359 (AM335x ICE V2 board)
 - Select the CPSW using jumpers J18 and J19 between pin 1 and 2
 - Select pin 1 and 2 shorted on J5 for booting
- EtherCAT Slave device (TMDSIDK437x)

NOTE: This TI Design was tested using the TMDSIDK437x as the EtherCAT slave.

- PC with terminal connection (for example, Tera Term through USB)
- Windows® PC with minimum 2GB RAM

5.2 Software

The following list shows the software required to support this design.

- [Acontis EC-Master V2.9 for SYS/BIOS](#)
- [AM335x SYS/BIOS Industrial SDK v1.1.0.8](#)

NOTE: After installing Industrial SDK, add or change IA_SDK_HOME in the PC Environment Variables to point to the SDK root directory.

- Code Composer Studio™ 6.1.3 Compiler v5.1 or higher
- [SYS/BIOS 6.41.04.54](#)
- [XDC v3.30.06.67](#) or higher
- Serial console terminal application (for example, TeraTerm, minicom, and HyperTerminal)

If using the AM437x IDK as a slave, download [Industrial SDK 2.1.0.1 pre-build binaries](#)

6 Preparing the Application

This TI Design was tested using the AM437x as an EtherCAT Slave. To use the AM437x as the slave device, refer to [Appendix A](#).

The TI Design prebuilt software packages include the ENI file for a topology where the AM437x is the only slave device that is connected. If the user switches to a different slave-bus topology or wants to recreate the ENI files for the AM437x, users can use the acontis EC-Engineer tool. To create an ENI file with the EC-Engineer Tool, refer to [Appendix B](#).

1. Install [Industrial SDK version 1.1.0.8](#) on the host PC
2. Unzip [EC_Master_Sysbios_SDK_Eval](#)
3. Consider and select one of the following:
 - (a) If the slave bus is the AM437x, copy the MasterENI.c file from the prebuilt package and paste inside <EC-master_installation_path>\\Workspace\\SYSB\\OS\\
 - (b) If the slave bus is not the AM437x, convert the ENI file (eni.xml) to a C file with array
 - (i) Obtain the eni.xml file by following the instructions in [Appendix B](#)
 - (ii) Download and install [Industrial SDK](#)

(iii) Use the Industrial IDK converter tool found in </ISDK_installation_path>\sdk\tools\bin2header to convert the eni.xml file

NOTE: The following are example parameters to use with bin2header.exe:

bin2header.exe eni.xml MasterENI.c MasterENI_xml_data

Add the file size information at the end of the new MasterENI.c file. See the following code example.

```
unsigned int MasterENI_xml_data_size = 16426;
```

The file size prints in the console.

4. Open CCS and import the EC-Master demonstration project.

(a) Navigate to *File*→*Import*→*CCS Projects*

(b) Navigate to

<ECmaster_installation_path>\contis_EC_master\EC_Master_Sysbios_SDK_Eval\Workspace\SY
SBIOS\EcMasterDemo

(c) Click OK, and then click Finish

(d) Check that SYSBIOS, XDC, and compiler versions are correct in the CCS project properties

(e) Click the Clean Project option

(f) Click the Build Project option

The EcMasterDemo.out output application is in the Debug Folder or Release Folder folder after building the project.

NOTE: Users can load and run EcMasterDemo.out from the prebuilt package.

To change the hardcoded parameters for the demonstration, use DEMO_PARAMETERS in ATEMDemoConfig.h. See the following snippet of code.

```
#define DEMO_PARAMETERS      "=auxclk 2000 -v 2 -t 10000 -perf " \
    "-cpsz " \
    "1" /* port */ \
    "1" /* mode */ \
    "1" /* priority */ \
    "m" /* master flag */ \
    "1" /* PHY address */ \
    "1" /* PHY connection mode: RGMII */
```

- -auxclk: the clock period in μ s
- -t: specifies the time to run the demonstration application in ms
- -perf: enables job-performance measurement

7 Running the Application

Use the following instructions to run the application.

1. Power on the EtherCAT Slave device
2. Confirm that the Ethernet bus is connected to the correct port
 - Find the port that was configured on DEMO_PARAMS; the options are:
 - if Port = 1, use ETH0
 - if Port = 2, use ETH1
 - If using the AM437x as the EtherCAT Slave, connect a cable to PRUETH0 (J6)
3. Connect a USB cable to the ICE v2 board
4. Set up a terminal connection to the Windows PC

5. Configure the host serial port as follows (see [Figure 3](#)): 115200 baud, no parity, 1 stop bit, no flow control

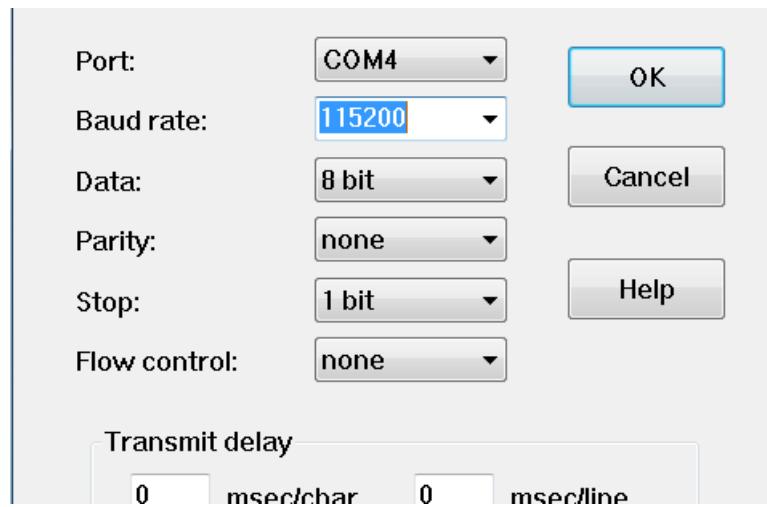


Figure 3. Serial Port Settings

6. Open CCS

7. Create a target configuration

- Navigation to *View*→*Target Configuration*.
- Right-click *New Target Configuration*
- Create a file name (for example, ICEv2.ccxml)
- Set the connection to XDS100v2 USB Emulator
- Set the board or device to ICE_AM3359
- Click *Target Configuration* and select Cortex-A8
- Enter the initialization script as follows: `..\..\..\am335x_sysbios_ind_sdk_1.1.0.8\ sdk\ tools\ gel\ ICE\ TMDXICE3359_v2_1A.gel`
- Click the *Save* button
- Right-click *ICEv2.ccxml* to launch the configuration
- Right-click Cortex-A8 to connect to the target
- Click *CPU (Reset HW)*
- Click *Scripts*
- Click *AM335x System Initialization*
- Click *AM3359_ICE_Initialization*
- Click on *Load Program*, and then click *Browse Project*
- Click *EcMasterDemo*
- Click *Debug or Release*
- Click *EcMasterDemo.out*
- Click the *Resume* button

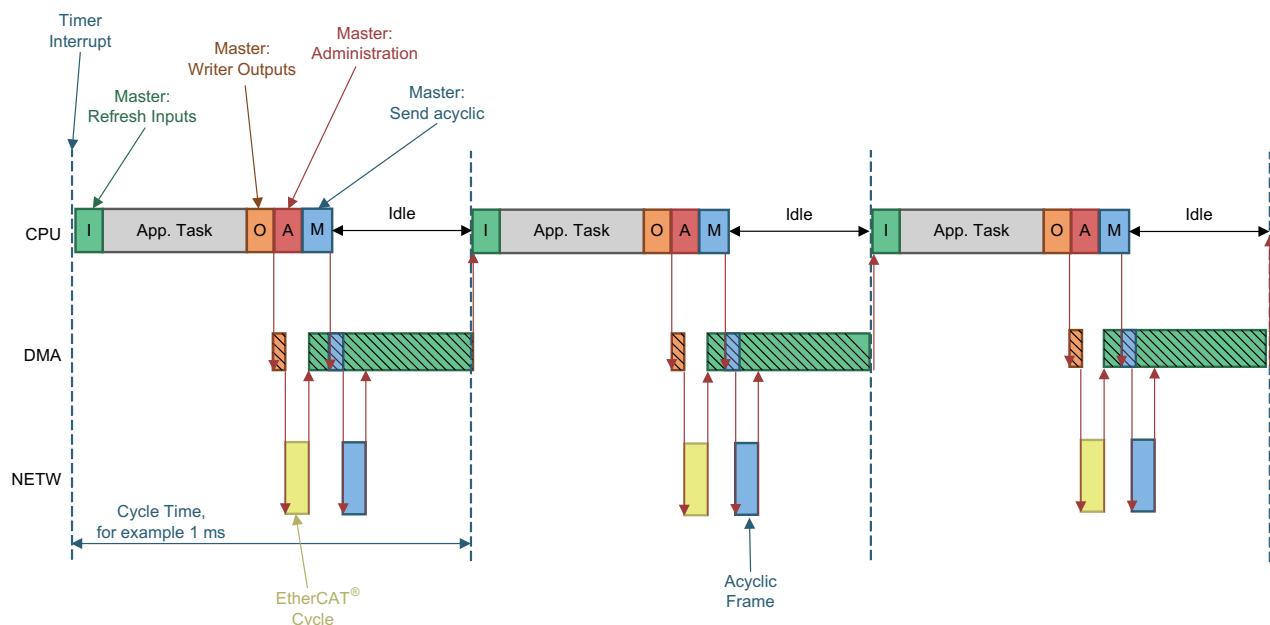
See [Appendix C](#) for an example of an output console.

8 EtherCAT® Master Benchmark

In this design, the EtherCAT Master has no internal tasks. As a result, EtherCAT functions as a driver for the application. This implementation brings some benefits such as no synchronization issues between application and EtherCAT Master. The application controls the timing of events, enabling the cyclic portion to run within an interrupt service routine (ISR).

To benchmark the EtherCAT Master on AM335x, the test includes seven commercially-available slave devices (EK110, 2x EL2004, 2x EL1004, EL4132, EK1110). The configuration used a frame load size of 579 bytes with 512 bytes of process data and a mailbox transfer to EL4132. The EtherCAT Master ran for 10 s, and TSC profiling data was collected for the following cycle jobs (see [Figure 4](#)).

- I: EtherCAT Master refresh inputs
- O: EtherCAT Master writer outputs
- A: EtherCAT Master administration functions
- M: EtherCAT Master acyclic datagrams and commands
- App: The application processes inputs and creates output values



Copyright © 2016, Texas Instruments Incorporated

Figure 4. Bus Timing Diagram

9 Test Data

[Table 1](#) shows the measurements (average CPU load) in [Figure 4](#).

Table 1. AM335x EC-Master CPU performance

Sitara AM335 600 MHz NIC: CPSW TI-RTOS (SYS/BIOS)		
NUMBER	EC-MASTER JOB	AVERAGE μ s
1	I: Process Inputs	16
2	O: Send Outputs	10
3	A: Administration	9
4	M: Send Acyclic Frame	4
	Total CPU Time	39

10 Design Files

10.1 Bill of Materials

To download the bill of materials (BOM), see the design files at [TIDEP0043](#).

10.2 Software Files

To download the software files, see the design files at the [TIDEP0043](#).

11 References

1. EtherCAT.org, *Technical Introduction and Overview*, ([EtherCAT](#))
2. Texas Instruments, *EtherCAT ® on Sitara™ Processors*, White Paper ([SPRY187](#))

12 About the Author

PAULA CARRILLO is a software engineer for the Embedded Processing group at TI. She obtained her MSEE from Florida Atlantic University and her Bachelor's Degree at Javeriana University, Colombia. Since joining TI in 2009, Paula has been working on different SoC and multicore DSP platforms developing applications for high-performance video codecs, synthetic aperture radar (SAR), and industrial communication protocols.

13 About the Author—acontis technologies

STEFAN ZINTGRAF graduated as a Software Engineer in 1987 and began working as software engineer in Sulzer and LP Elektronik. After several years working as team leader in LP Elektronik he co-founded acontis technologies GmbH in 2001, working as general manager. Stefan took over acontis' EtherCAT development activities in 2005 after being responsible for the acontis Windows® real-time products. He is also responsible for the fast growing EtherCAT market in Asia in worldwide sales activities.

Revision History

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

Changes from Original (September 2015) to A Revision	Page
• Updated Hardware Subsection	5
• Updated Software Section.....	5
• Updated acontis EC-Master from version 22.7.2.12 to 2.9	5
• Updated CCS version from 6.0.0.00040 to 6.1.3	5
• Updated the instructions in Preparing the Application Section.....	5
• Updated Running the Application Section	6
• Updated Troubleshooting Section	15

Appendix A AM437x Running EtherCAT Slave

A.1 *Preparing the SD Card*

Prepare the SD card by file allocation table (FAT) formatting it as follows:

1. Ensure the HP USB disk storage format tool v2.0.6 is portable
2. Run the HP USB disk storage format tool v2.0.6 as portable executable. The executable detects the SD card that is plugged into the reader. If undetected, direct the executable to the new disk.
3. Choose FAT32 if the SD card is greater than 4GB. If not, use FAT
4. Click the Start button

NOTE: After formatting, the card can be populated by the files.

5. Install the ISDK prebuilt binaries ([Industrial SDK 2.1.0.1](#))
6. Copy MLO from
`<Installation_path>\sysbios\ind\ sdk\prebuilt_02_01_00_01\Bootloader\SD\am437x_release` to the SD card
7. Copy the EtherCAT Slave application from
`<Installation_path>\sysbios\ind\ sdk\prebuilt_02_01_00_01\ethercat_slave\am437x_release`
8. Connect the Ethernet cable to the ICCS port J6
9. Use Wireshark to test the ecat (EtherCAT) packets

Alternatively, use an EtherCAT Master, such as TwinCAT®, to test master-to-slave connectivity

For details using TwinCAT with TI IDK boards, see the user guide at [wiki.ti.com](#).

Appendix B acontis EC-Engineer Tool for Creating an .ENI File

1. Register for a free evaluation version at www.acontis.com
2. Install the EC-Engineer tool on the PC
3. Connect the EtherCAT slave to the computer (if using the AM437x, connect the Ethernet cable to J6)
4. Open the EC-Engineer tool
5. Select Online Configuration
6. Select the EtherCAT Master unit (Class A) as the master unit
7. Click the OK button
8. Select 2000 as the Cycle Time (μs)
9. Select the desired network adapter as the slave that is connected to the local system.

NOTE: If using the AM437x as the slave, add TI's IDK.xml to the ESI manager as follows:

- Open the ESI manager
- Add the IDK.xml file
- Browse to
`<Installation_path>\sysbios\ind\ sdk_2.1.0.1\ sdk\ examples\ ethercat_slave\ esi\ TiEtherCATLib.xml`
- Open the file
- Navigate to the Network option
- Click Scan EtherCAT Network
- Click Export ENI after the slaves are found

Selecting Export ENI exports eni.xml.

Appendix C Application Console Output

TI Industrial SDK Version - IASDK 1.1.0.8

Device name : AM3359

Chip Revision : AM335x ES1.2 [PG2.1]

SYS/BIOS EcMaster Sample application running on ICE V2

Full command line: -auxclk 2000 -v 2 -t 100000 -perf -cpsw 1 1 1 m 1 1

Run demo now with cycle time 2000 usec Using AuxClock

=====

Initialize EtherCAT Master

=====

EC-Master V2.7.2.12 (Eval) for SYSBIOS Copyright acontis technologies GmbH @ 2015

CPSW INF: Port 1, Prio 1, Flags [Polling] [Master], MAC 00:00:00:00:00:01

CPSW INF: CPSW3G found. CPSW INF: HW-Id: 0x0019, RTL: 0, Major: 1, Minor: 0xc

CPSW INF: PHY found. Id=0x2000a211

CPSW INF: Restart PHY auto negotiation

CPSW INF: PHY auto negotiation completed

Evaluation Version, stop sending ethernet frames after 60 minutes!

PDI Watchdog expired - Slave Slave_1001 [TIESC-002]: - EtherCAT address=1001

Bus scan successful - 1 slaves found

Number : 0

Vendor ID : 0xE000059D = ----

Product Code: 0x54490002 = Unknown

Revision : 0x00000001 Serial Number: 0

ESC Type : TI Sitara (0x90) Revision: 2 Build: 947

Bus AutoInc Address: 0 (0x0)

Bus Station Address: 1001 (0x3e9)

Bus Alias Address : 0000 (0x0)

Connection at Port 0: yes Port 1: no Port 2: no Port 3: no

SlaveID at Port 0: 65536 Port 1: -1 Port 2: -1 Port 3: -1

Config Station Address: 1001 (0x3e9)

PD IN Byte.Bit offset: 0.0 Size: 32 bits

PD OUT Byte.Bit offset: 0.0 Size: 32 bits

EtherCAT network adapter MAC: 00-00-00-00-00-01

=====

Start EtherCAT Master

```
=====
Master state changed from <UNKNOWN> to <INIT>
Master state changed from <INIT> to <PREOP>
Master state changed from <PREOP> to <SAFEOP>
Master state changed from <SAFEOP> to <OP>
```

Job times during startup <INIT> to <OP>:

```
=====
PerfMsmt 'JOB_ProcessAllRxFrames' (avg/max) [usec]: 29.6/ 91.4
PerfMsmt 'JOB_SendAllCycFrames ' (avg/max) [usec]: 18.2/ 36.5
PerfMsmt 'JOB_MasterTimer ' (avg/max) [usec]: 50.7/263.0
PerfMsmt 'JOB_SendAcycFrames ' (avg/max) [usec]: 21.3/ 64.4
PerfMsmt 'Cycle Time ' (avg/max) [usec]: 1624.0/2011.7
PerfMsmt 'myAppWorkPd ' (avg/max) [usec]: 1.7/ 5.8
```

```
=====
PerfMsmt 'JOB_ProcessAllRxFrames' (avg/max) [usec]: 32.9/ 48.4
PerfMsmt 'JOB_SendAllCycFrames ' (avg/max) [usec]: 30.6/ 39.2
PerfMsmt 'JOB_MasterTimer ' (avg/max) [usec]: 52.3/ 61.5
PerfMsmt 'JOB_SendAcycFrames ' (avg/max) [usec]: 7.0/ 15.9
PerfMsmt 'Cycle Time ' (avg/max) [usec]: 1998.1/2014.2
PerfMsmt 'myAppWorkPd ' (avg/max) [usec]: 2.2/ 9.0
```

Appendix D Troubleshooting

If the code hangs while reading PHY register MDIO, add the following line of code in main.c:

```
GPIOInit();  
UTILsSetBoardType(3); //Added to fix board type to ICEv2  
s_boardType = UTILsGetBoardType();  
s_uartInstance = InitUart(s_boardType);
```

If the following error is printed in the console, the SPI likely prefetched.

```
001181 : CPSW ERR: mdio ACK missing  
001181 : CPSW INF: mdio ref 2 error  
001181 : CPSW ERR: PHY initialization failed
```

The two solutions to this error are:

- Change the boot jumper to NOR Pin 1 and 2 shorted on J5, or
- Erase SPI flash
 - For assistance erasing flash memory, see wiki.ti.com

For additional troubleshooting, refer to section 4.2 Error Codes in *Acontis EC-Master Stack Class B Version 2.7 Document* by navigating to
<ECmaster_installation_path>\Acontis_EC_master\EC_Master_Sysbios_SDK_Eval\Doc.

Users should also refer to *AM335 Sysbios User Guide and Getting Started Guide* by navigating to
</ASDK_path>\am335x_sysbios_ind_sdk_1.1.0.8\ sdk\docs.

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 equipment includes, without limitation, all medical devices identified by the U.S. Food and Drug Administration as Class III devices and 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 non-compliance with the terms and provisions of this Notice.

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