TI Designs

EtherCAT® Master Stack for TI Sitara™ CPU Family

TI Designs

TI Designs provide the foundation that you need including methodology, testing and design files to quickly evaluate and customize the system. TI Designs help *you* accelerate your time to market.

Design Resources

TIDEP0043	Tool Folder Containing Design Files
TMDSICE3359	Tool Folder
AM3359	Product Folder

TI E2E™ Community

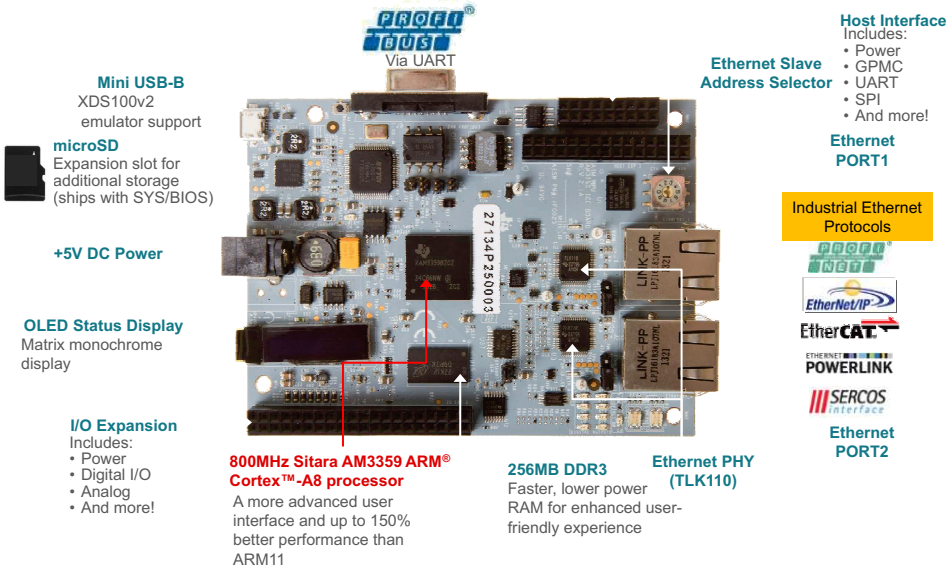
ASK Our E2E Experts
WEBENCH® Calculator Tools

Design Features

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

Featured Applications

- EtherCAT Programmable Logic Control System (PLC)
- 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.

1 Introduction

This TI design document presents TI's Sitara AM335x implementing an EtherCAT Master using Acontis EtherCAT master stack. The Acontis' EtherCAT Master stack (also known as EC-Master) is a highly portable software stack, when combined with a high-performance TI Sitara CPU, provides a sophisticated EtherCAT solution that customers 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 port to a different OS or even 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 µsec.

2 EtherCAT Protocol

EtherCAT is an IEEE 802.3 Ethernet-based fieldbus system. EtherCAT defines a new standard in communication speed. EtherCAT is inexpensive to implement ,which lets the system use fieldbus technology in applications which previously had to omit fieldbus. EtherCAT is an open technology which is standardized within the International Electrotechnical Commission (IEC). The technology is supported and powered by the EtherCAT Technology Group, which is an international community of users and vendors. The protocol is suitable for both hard and soft real-time requirements in automation technology. One of the primary advantages of EtherCAT is its ability to support automation applications requiring 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 addressed to it on the fly and inserts its data in the frame as the frame is moving downstream. The frame is delayed only 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 to actively send an EtherCAT frame; all other nodes only forward frames downstream. This capability permits the network to achieve over 90% of the available network bandwidth and 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 using a special Ethertype identifier (0x88A4). EtherCAT communications consists of several EtherCAT telegrams Each telegram serves a particular 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 exchange between EtherCAT controller (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.

For slave-to-slave communication, two mechanisms are available. Upstream devices can communicate to downstream devices within the same cycle and thus extremely fast. The other option is using the freely configurable slave-to-slave communication which runs through the master device. This option requires two bus cycles (although not necessarily two control cycles).

3 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 well 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 PRU coprocessor (two real-time cores). This PRU can be used for communication protocols such as EtherCAT master and slave, Profinet, EthernetIP and Sercos among others. In this TI design, EtherCAT master uses AM335x Common Platform Ethernet Switch (CPSW) which frees PRU for other protocols or applications.

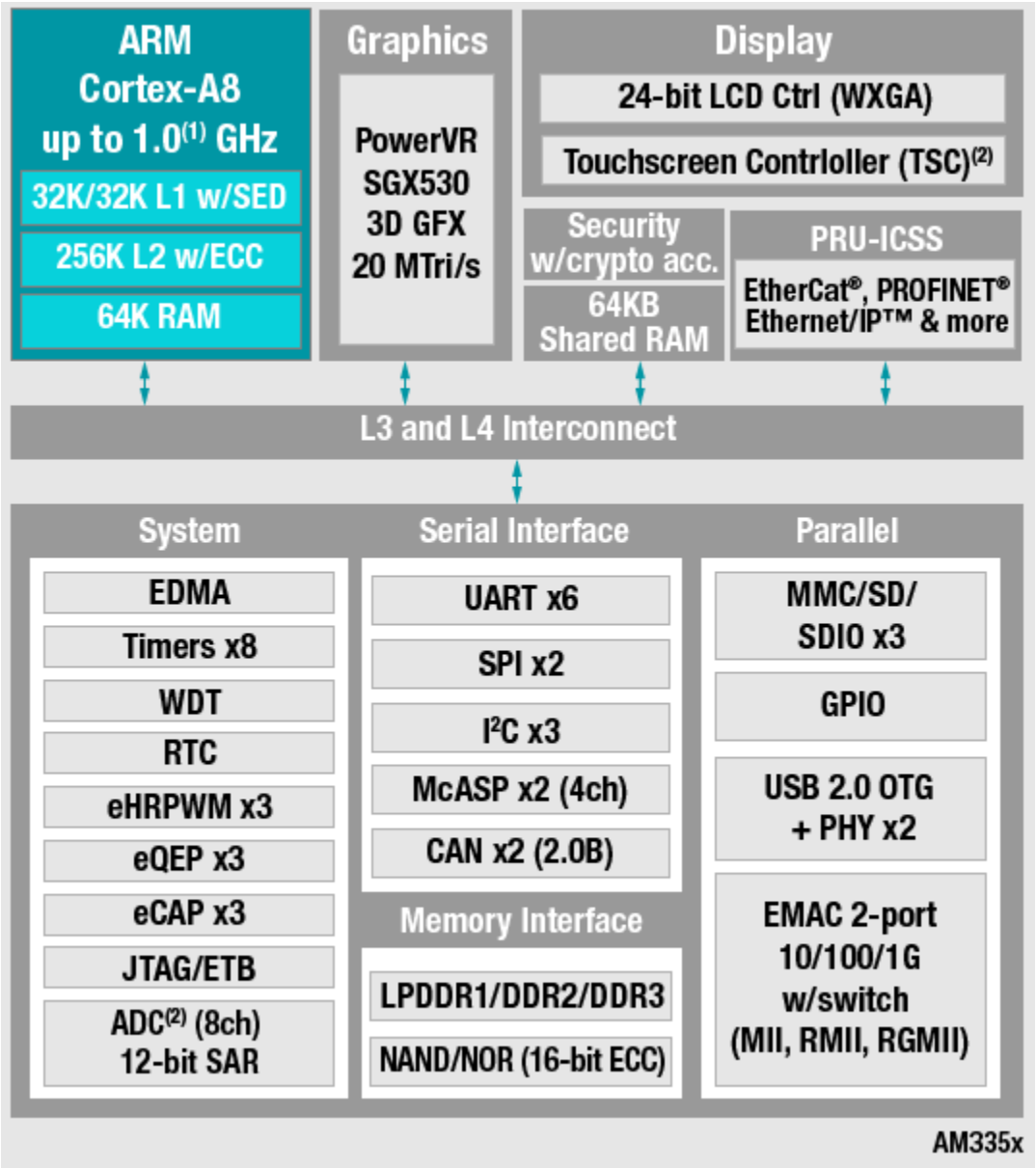


Figure 1. Block Diagram

4 Acontis EtherCAT Master Architecture

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 a XML file whose format is fixed in the EtherCAT specification ETG.2100. 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.

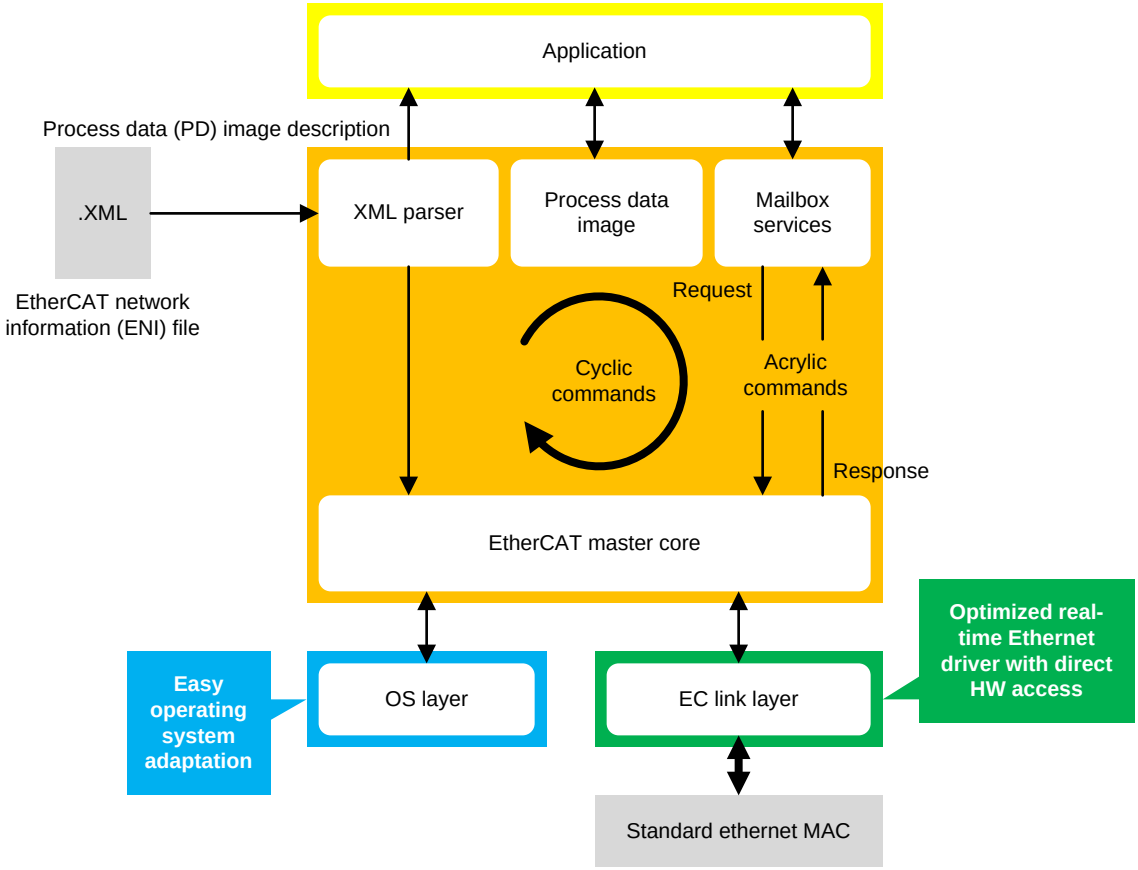


Figure 2. Acontis EC-Master Component Model Architecture

5 Getting Started

5.1 Hardware

The hardware to support this design is as follows:

- A TMDSICE3359 (AM335x ICE v2 board)
- A EtherCAT slave device. In this design, AM437 IDK.
- A PC with terminal connection (for example, TeraTerm through USB)
- A Windows® PC with minimum 2GB RAM

5.2 Software

The software to support this design is as follows:

- Acontis EC-Master V2.7.2.12 for SYSBIOS
- AM335x SYSBIOS Industrial SDK v1.1.0.8
- Code Composer Studio™ 6.0.0.00040 Compiler v5.1 or higher
- SYSBIOS 6.41.04.54
- XDC v3.30.06.67 or higher
- A network development kit (NDK) 2.24.01.18
- A serial console terminal application (for example, TeraTerm, minicom, and HyperTerminal)
  If using AM437x IDK as a slave, download the Industrial SDK 2.1.0.1 pre-build binaries

6 Preparing the Application

EtherCAT Slave: If using AM437x as a slave device, refer to Appendix A.

ENI File: If using the Acontis EC-Engineer tool to create an ENI file, refer to Appendix B.

EtherCAT Master: Install Industrial SDK version 1.1.0.8 on the host PC.

1. Unzip EC_Master_Sysbios_SDK_Eval.
2. Convert ENI file to the C file with array.
   Use the bin2header.exe file from Industrial SDK tools at </ASDK_path>\sdk\tools\bin2header. See the following parameters:
   bin2header.exe eni.xml MasterENI.c MasterENI_xml_data
3. Add the file size information at the end of your new MasterENI.c file. See the following example:
   unsigned int MasterENI_xml_data_size = 16426;

   NOTE: File size prints in the console.
4. Open CCS and import the EC-Master demonstration project.
   (a) Navigate to File.
   (b) Navigate to Import.
   (c) Navigate to CCS Projects.
   (d) Navigate to <ECmaster_installation_path>\Acontis_EC_master\EC_Master_Sysbios_SDK_Eval\Workspace\SYSBIOS\EcMasterDemo
   (e) Click OK.
   (f) Click Finish.
   (g) Check that SYSBIOS, XDC, and compiler versions are correct in the CCS project properties.
   (h) Click Clean Project.
   (i) Click Build Project.

After building the project, the EcMasterDemo.out output application is in the debug or release folder.

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 100000 -perf " \
                                    "-cpsw "                             \
                                        "1 "    /* port */               \
                                        "1 "    /* mode */               \
                                        "1 "    /* priority */           \
                                        "m "    /* master flag */        \
                                        "1 "    /* PHY address */        \
                                        "1"    /* PHY connection mode: RGMII */
```

- -auxclk refers to the clock period in µsec
- -t specifies the time to run the demonstration application in msec
- -perf enables job performance measurement

7 Running the Application

To run the application, do as follows:

1. Power on the EtherCAT slave device.
2. Confirm Ethernet bus is connected to the correct port.
3. Connect a USB cable to the ICE v2 board.
4. Set up a terminal connection with the Windows PC.
5. Configure the host serial port as follows: 115200 baud, no parity, 1 stop bit, no flow control
6. Open CCS.
7. Create a target configuration.
   (a) Navigate to View.
   (b) Navigate to Target Configuration.
   (c) Right-click New Target Configuration.
   (d) Create a file name (for example, ICEv2.ccxml).
   (e) Set the connection to XDS100v2 USB Emulator.
   (f) Set the board or device to ICE_AM3359.
   (g) Click Target Configuration.
   (h) Select Cortex-A8.
   (i) Enter the initialization script as follows: \..\..\..\am335x_sysbios_ind_sdk_1.1.0.8\sdk\tools\gel\ICE\TMDXICE3359_v2_1A.gel.
   (j) Click Save.
   (k) Right-click ICEv2.ccxml to launch the configuration.
   (l) Right-click Cortex-A8 to connect to the target.
   (m) Click CPU (Reset HW).
   (n) Click Scripts.
   (o) Click AM335x System Initialization.
   (p) Click AM3359_ICE_Initialization.
   (q) Click Load Program.
   (r) Click Browse project.
   (s) Click EcMasterDemo.
   (t) Click Debug or Release.
   (u) Click EcMasterDemo.out.
   (v) Click Resume.

See Appendix C for an example of output console.

8 EtherCAT Master Benchmark

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

To benchmark the EtherCAT master on AM335x, the test includes seven commercially-available slave devices (EK110, 2xEL2004, 2xEL1004, 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 seconds and TSC profiling data was collected for the following cycle jobs.

- 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.

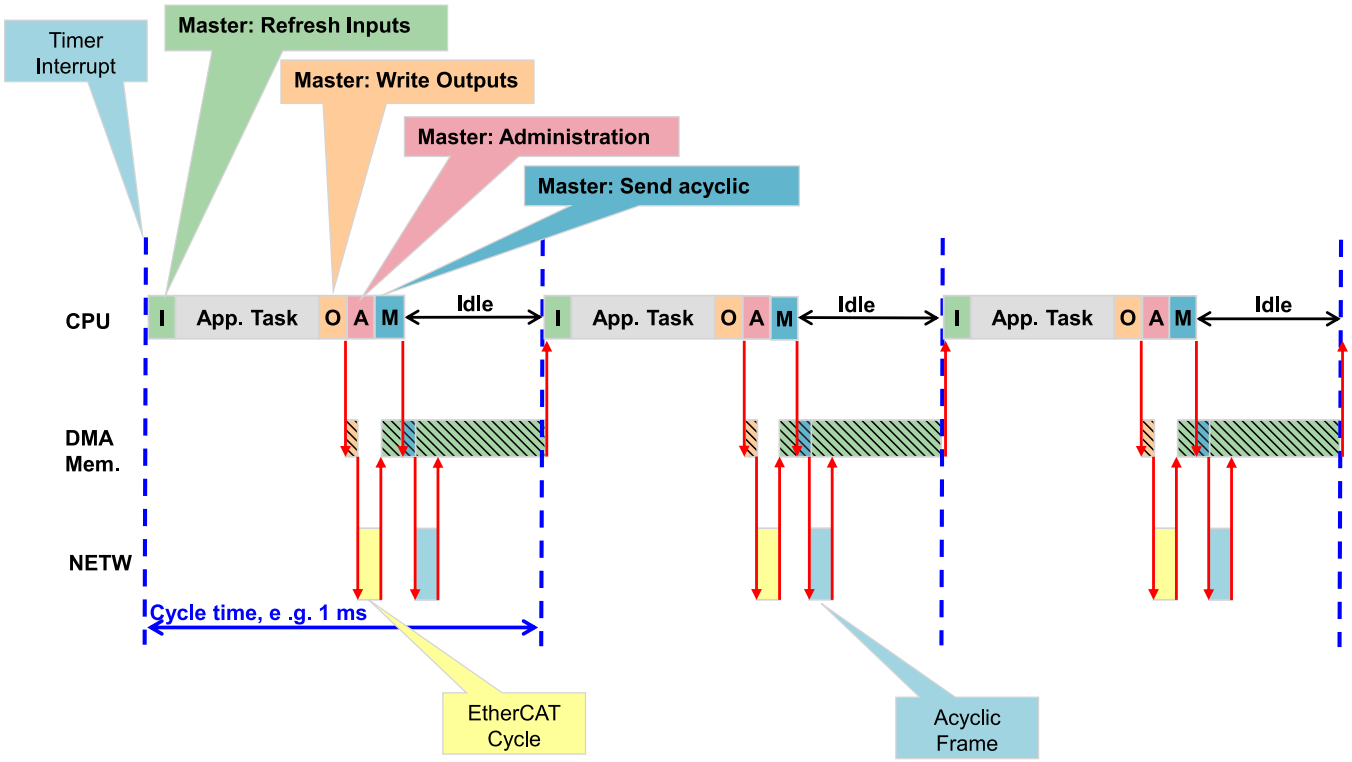


Figure 3. Bus Timing Diagram

9 Test Data

Table 1 shows the measurements (average CPU load).

Table 1. AM335x EC-Master CPU performance

		Sitara AM335 600 MHz NIC: CPSW TI-RTOS (SYS/BIOS)
Number	**EC-Master Job**	**Average µsec**
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 on Sitara Processors White Paper*, (187SPRYD)
2. EtherCAT Website
3. *Acontis EC-Master Stack Class B Version 2.7 Document*, <ECmaster_installation_path>\Acontis_EC_master\EC_Master_Sysbios_SDK_Eval\Doc
4. *AM335 Sysbios User Guide and Getting Started Guide*, </ASDK_path>\am335x_sysbios_ind_sdk_1.1.0.8\sdk\docs

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 degree at Javeriana University, Colombia. Since joining TI in 2009, Paula has been working on different SoC and multi-core DSP platforms; developing applications for high-performance video codecs, syntechtic aperture radar (SAR), and industrial communication protocols.

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 you have HP USB disk storage format tool v2.0.6 portable.
2. Run the HP USB disk storage format tool v2.0.6 portable executable. (The executable detects the SD card 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 Start.

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

5. Install the IASDK prebuild binaries, Industrial SDK 2.1.0.1.
6. Copy MLO from <Install_path>\sysbios_ind_sdk_prebuilt_02_01_00_01\Bootloader\SD\am437x_release to the SD card.
7. Copy the EtherCAT slave application from <Install_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 packests

   Alternatively, use an EtherCAT master, like TwinCAT®, to test master–slave connectivity.

For details using TwinCAT with TI IDK boards, go to http://processors.wiki.ti.com/index.php/SYSBIOS_Industrial_SDK_02.01.00.01_User_Guide#EtherCAT.

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

1. Register for a free evaluation version at http://www.acontis.com/eng/products/downloads/index.php.
2. Install the EC-Engineer tool on your PC.
3. Connect your EtherCAT slave to your computer. (If using 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 OK.
8. Select 2000 as the Cycle Time (µs).
9. Select your network adapter as the slave connected to the local system.

NOTE: If using 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 <Install_path>\sysbios_ind_sdk_2.1.0.1\sdk\examples\ethercat_slave\esi\TiEtherCATLib.xml.
- Open the file.
- Navigate to Network.
- Click Scan EtherCAT Network.
- Click Export ENI after the slaves are found.

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 FAQ - Troubleshooting

- If code hangs while Reading PHY register MDIO, add the following line in main.c:
  GPIOInit();
  UTILsSetBoardType(3); //Added to fix board type to ICEv2
  s_boardType = UTILsGetBoardType();
  s_uartInstance = InitUart(s_boardType)
- For additional troubleshooting, refer to section 4.2, Error Codes in *Acontis EC-Master Stack,* class B, version 2.7 document,
  <ECmaster_installation_path>\Acontis_EC_master\EC_Master_Sysbios_SDK_Eval\Doc .

IMPORTANT NOTICE FOR TI REFERENCE DESIGNS

Texas Instruments Incorporated ("TI") reference designs are solely intended to assist designers (“Buyers”) who are developing systems that incorporate TI semiconductor products (also referred to herein as “components”). Buyer understands and agrees that Buyer remains responsible for using its independent analysis, evaluation and judgment in designing Buyer’s systems and products.

TI reference designs have been created using standard laboratory conditions and engineering practices. **TI has not conducted any testing other than that specifically described in the published documentation for a particular reference design.** TI may make corrections, enhancements, improvements and other changes to its reference designs.

Buyers are authorized to use TI reference designs with the TI component(s) identified in each particular reference design and to modify the reference design in the development of their end products. HOWEVER, NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY THIRD PARTY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT, 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 components 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 such information 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 ARE PROVIDED "AS IS". TI MAKES NO WARRANTIES OR REPRESENTATIONS WITH REGARD TO THE REFERENCE DESIGNS OR USE OF THE REFERENCE DESIGNS, EXPRESS, IMPLIED OR STATUTORY, INCLUDING ACCURACY OR COMPLETENESS. TI DISCLAIMS ANY WARRANTY OF TITLE AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, QUIET ENJOYMENT, QUIET POSSESSION, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS WITH REGARD TO TI REFERENCE DESIGNS OR USE THEREOF. TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY BUYERS AGAINST ANY THIRD PARTY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON A COMBINATION OF COMPONENTS PROVIDED IN A TI REFERENCE DESIGN. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, SPECIAL, INCIDENTAL, CONSEQUENTIAL OR INDIRECT DAMAGES, HOWEVER CAUSED, ON ANY THEORY OF LIABILITY AND WHETHER OR NOT TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, ARISING IN ANY WAY OUT OF TI REFERENCE DESIGNS OR BUYER’S USE OF TI REFERENCE DESIGNS.

TI reserves the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques for TI components are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

Reproduction of significant portions of TI information in TI data books, data sheets or reference designs is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards that anticipate dangerous failures, monitor failures and their consequences, lessen the likelihood of dangerous failures and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in Buyer’s safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed an agreement specifically governing such use.

Only those TI components that TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components that have ***not*** been so designated is solely at Buyer's risk, and Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265