MYD-C437X-PRU RTOS Development Guide

MYiR™ Make Your Idea Real

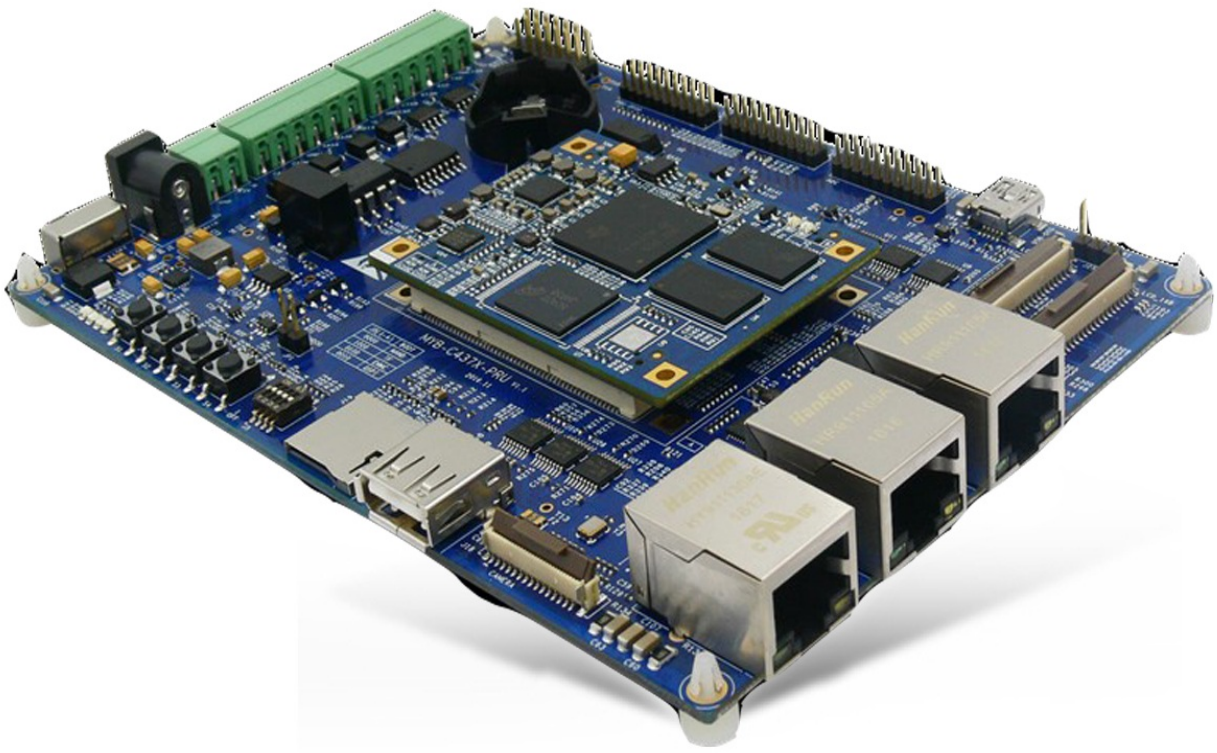

Table of Contents

MYD-C437X-PRU RTOS Development Guide

Introduction

This document quickly provides the information you need most while evaluating and developing SYSBIOS-based RTOS applications on MYD-C437X-PRU development board,
it includes the content of setup development environment, building RTOS application and running a EtherCAT slave demo application.

Version History:

Version	Description	Time
V1.0	Initial Version	2017.1.16
V2.0	Add Startup section, adjust some details	2018.6.12

Hardware Version:

This document applies only to MYD-C437X-PRU development board of myirtech.

1. Software Resources

Along with the MYD-C437X-PRU development board, we provide a SDK of RTOS, it includes the fundamental platform software and tools for development,deployment and execution of RTOS-based applications. The content of SDK for RTOS is in the directory *05-RTOS_Source*. It is shown as a table below.

Table 1-1 Software Resouces List

Category	Name	Description	source
Bootloader	startware bootloader	Responsible for system initialization and booting SYSBIOS kernel, loading SYSBIOS application	YES
Kernel	sysbios	TI SYSBIOS kernel bios_6_45_01_29	YES
Driver	emac	am437x emac driver	YES
Driver	gpio	gpio driver	YES
Driver	gpmc	gpmc driver	YES
Driver	i2c	i2c driver	YES
Driver	icss_emac	icss emac driver	YES
Driver	mcasp	mcasp driver	YES
Driver	mmcsd	mmc host driver	YES
Driver	pruss	pruss driver	YES
Driver	spi	spi driver	YES
Driver	uart	uart driver	YES
Application	ethercat_slave	EtherCAT slave demo	YES
Application	ethernet_mac	ethernet mac demo	YES
Application	ethernetip_adapter	ethernet ip demo	YES
Application	example_utils	misc example utils	YES
Tools	xdctools	xdctools_3_32_01_22_core	exe
Tools	Win32DiskImager	TF/SD image programer	exe
Tools	Format Tools	TF/SD Format tools	exe
Tools	CCS6	Code Composer Studio IDE 6.2.0	exe
Tools	TWinCAT	TC31-Full-Setup.3.1.4020.28	exe

2. Deploy Development Enviornment

The following section covers the setup of hardware, deployment and verification of software development environment.

Hardware Preparation:

- One Host PC
- One XDS100V3 compatible emulator
- One MYD-C437X-PRU development board
- One TF card, one USB to TTL converter and one CAT5 ethernet cable

Connect the TTL Debug Interface J25(Debug UART in the following picture)to PC with a USB to TTL converter and set the baudrate of serial port on host PC to 115200-8-n-1. Ethernet interface J6(Giga Ethernet in the following picture) is corresponding to MAC0 of AM437X; J26 and J27 are industrial ethernet interface shown as PRUETH0 and PRUETH1 in the following picture.
If you wat to debug with JTAG emulator, please connect the xds100v3 compatible emulator to J22(TI-JTAG interface).
A high level picture of MYD-C437X-PRU is shown below:

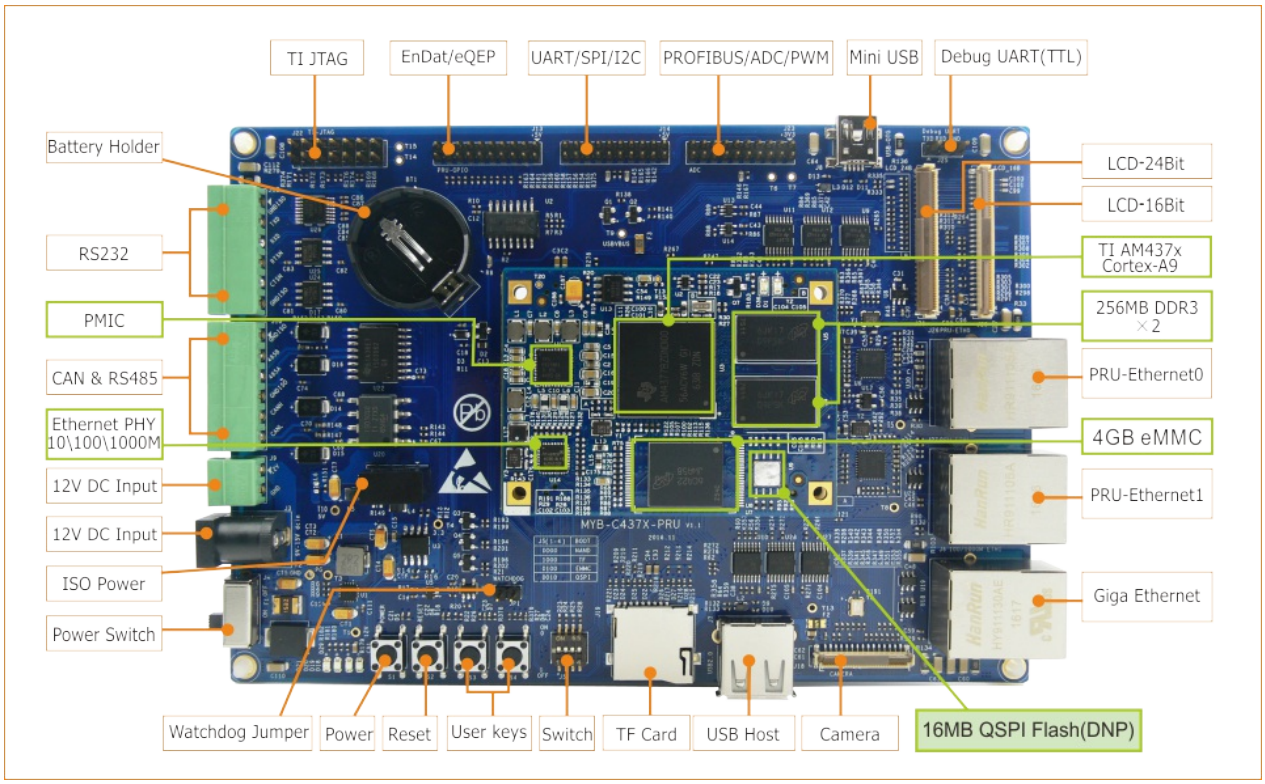

Figure 2-1 MYD-C437X-PRU Interface Definition

Software Preparation:

- Ubuntu12.04 64bit Desktop
- Windows7/Windows10 Host PC
- Code Composer Studio IDE 6.2.0

Install Code Composer Studio IDE 6.2.0:

Code Composer Studio IDE is the default software tools for development and debugging with TI SYSBIOS-based RTOS. There are serval versions of CCS on the website of TI, we choose the version 6.2.0.00050
as default. To install CCS, please refer to the instructions provided in the CCSv6 Getting Started Guide, we set the default install path as *c:/ti*.

During installing CCS, we can choose to control what is installed for processor architecture. For AM437x processor, we should choose `Sitara 32-bit ARM Processors`, as shown below:

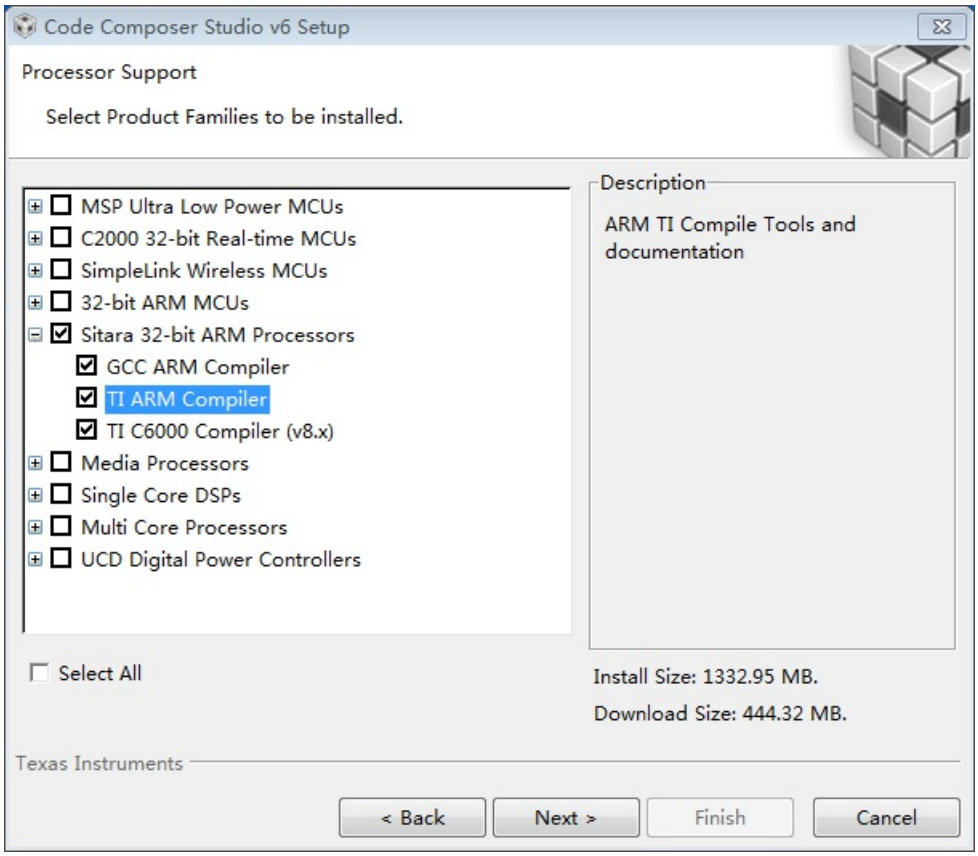

Figure 2-2 Choose processor on CCS

During installing CSS, we can choose some additional tools(add-ons), as shown below. We can also install them from App Center of CCS after installing.

Figure 2-3 Choose Add-ons from CCS App Center

The support for connecting a JTAG to the EVM to debug software is included in the CCS package. In some cases, additional modifications to configuration GEL files are provided separately. To get the latest version, perform a CCS Check for Updates, as shown below:

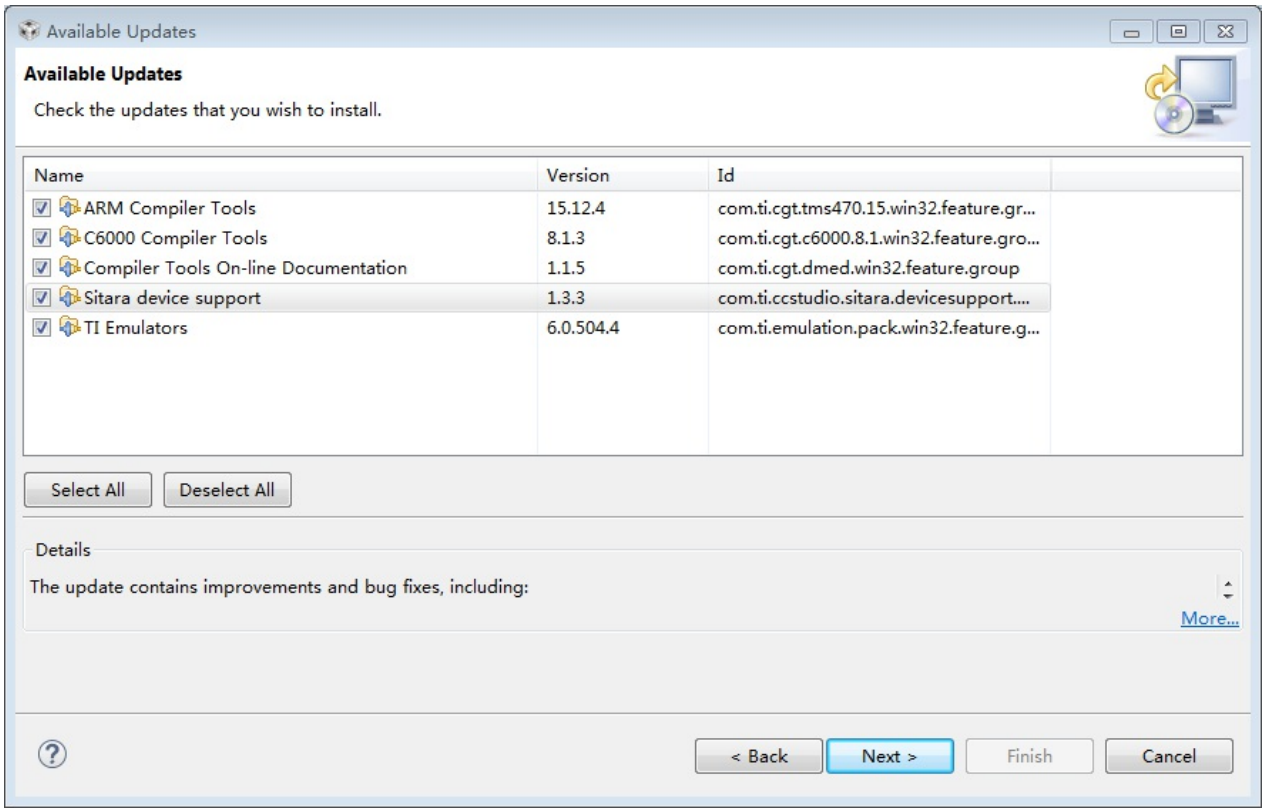

Figure 2-4 Update Sitara Device Support List

MYD-C437X-PRU is not in the official Sitara support list of TI, so it needs the user to add it manually, and the process is very simple. Extract the 05_RTOS_Source/MYIRTECH/myir_sitara_device_support_1.3.3.zip from the factory attached file. The directory structure is as follows：

```
├── common
│   └── targetdb
│       └── boards
│           └── mydidk437x.xml
└── emulation
    └── boards
        └── mydidk437x
            └── gel
                ├── AM437x_EVMs.gel
                ├── AM437x_MMRs.gel
                ├── AM437x_Status.gel
                ├── AM43xx_EMIFconfig_HWlvl.gel
                ├── AM43xx_ICSSconfig.gel
                ├── AM43xx_PLLconfig.gel
                ├── idk_am437x.gel
                └── mydidk437x.gel
```

Copy mydidk437x.xml file to *C:\ti\ccsv6\ccs_base\common\targetdb\boards* directory; copy mydidk437x directory to *C:\ti\ccsv6\ccs_base\emulation\boards* directory.

After that, we should restart CCS, and then create target configuration for `MYD-C437X-PRU` as shown in Figure 2-5 as below. Please choose `View -> Target Configurations` of mainmenu, right click the window of `Target Configuration`, choose `New Target Configuration` menu item in the popup window to create a new target configuration.
We named this new target configuration as `mydidkc437x.xml`, after saving the target configuration by choosing `Save` button, we can choose `Test Connection` button to verify the connection.
If we test connection successfully, we can set it as default by right clicking it and choosing `Set as Default` menu item.

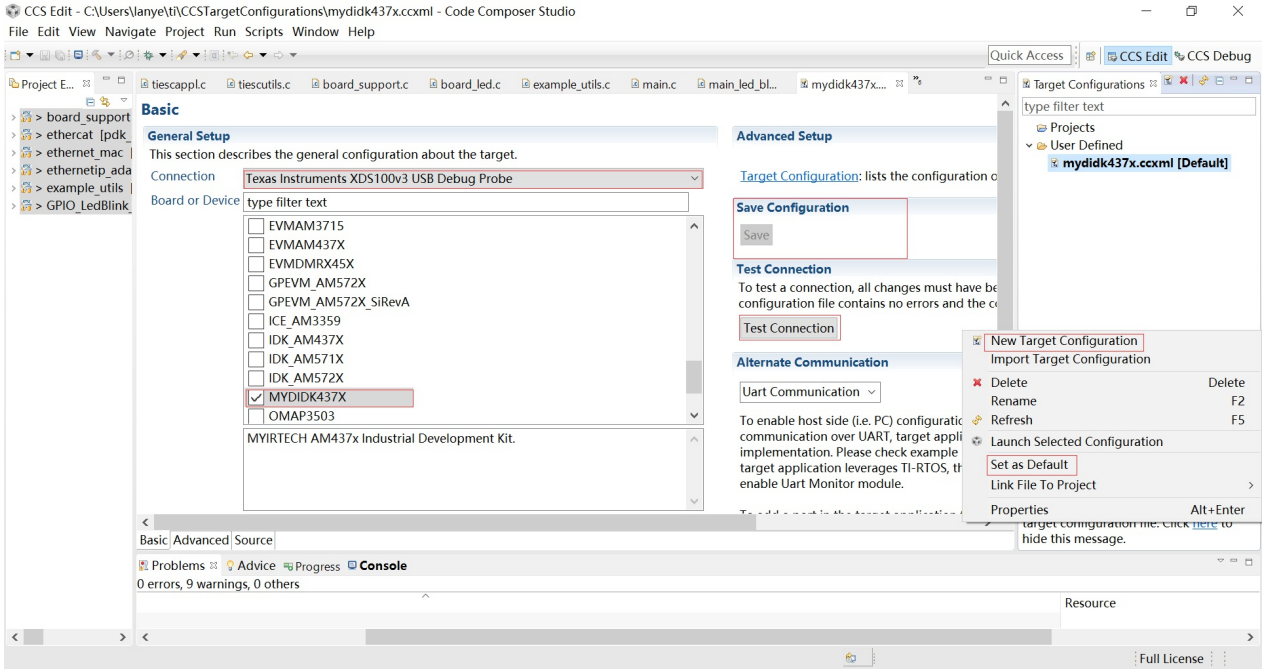

Figure 2-5 Create Target Configuration for MYDIDK437X

3. Build System

SYSBIOS-based RTOS is an open souce real time operation system designed by TI, the following section covers the migration of SYSBIOS kernel to MYD-C437X-PRU development board of myirtech. For details about RTOS, please refer to the WIKI page of TI at http://processors.wiki.ti.com/index.php/Category:SYSBIOS.

Preparation:

RTOS system was built based on `Processor RTOS SDK` of TI, we choose the SDK version processor_sdk_rtos_am437x_3_01_00_06 as default. The Windows install package of processor_sdk_rtos_am437x_3_01_00_06 is located in *05-RTOS_Source\TI*. Customers can also download from the website of TI. We install the processor rtos sdk to *c:/ti* as default, and the content of the sdk is shown as blow(includes the CSS installed before).

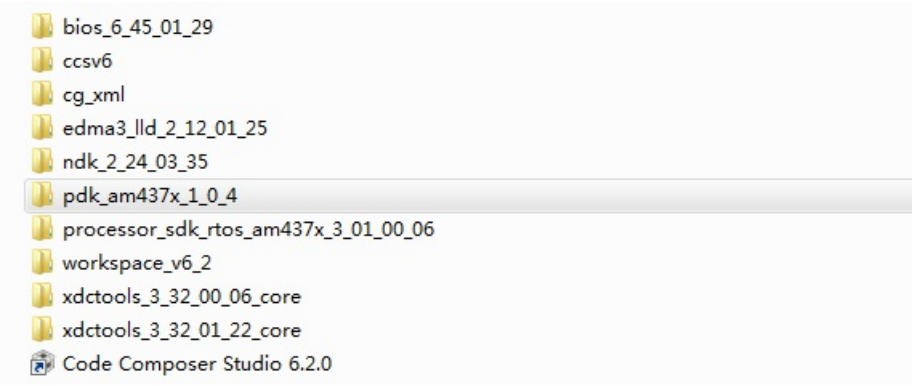

Figure 3-1 TI RTOS SDK Directory

As an evaluation platform especially designed for industrial communicaton application, we will introduce an EtherCAT slave application in the subsequent section. The EtherCAT slave application applies the official industrial SDK `sysbios_ind_sdk_02.01.02.02` from TI, customers can get the Windows install package from TI's website, or find it in *05-RTOS_Source\TI* of our release package.
Customers do not need to install sysbios_ind_sdk_02.01.02.02, because it is somewhat conflicted and duplicate with processor_sdk_rtos_am437x_3_01_00_06. We have extracted some industrial communication applications from sysbios_ind_sdk_02.01.02.02 and integrated them with processor_sdk_rtos_am437x_3_01_00_06, please copy and merge *05-RTOS_Source\MYIRTECH\pdk_am437x_1_0_4* to *C:\ti\pdk_am437x_1_0_4* manually.

Run RTOS Applicaton:

We should rename the RTOS application to `app`, then the bootloader will load the app to RAM and transfer execution control to `app`. The following section focuses on the mmc boot mode of RTOS application.

- Prepare a TF Card formated by HP USB Disk Storage Format Tool 2.0.6 in *03-Tools* of our release package, as shown below:

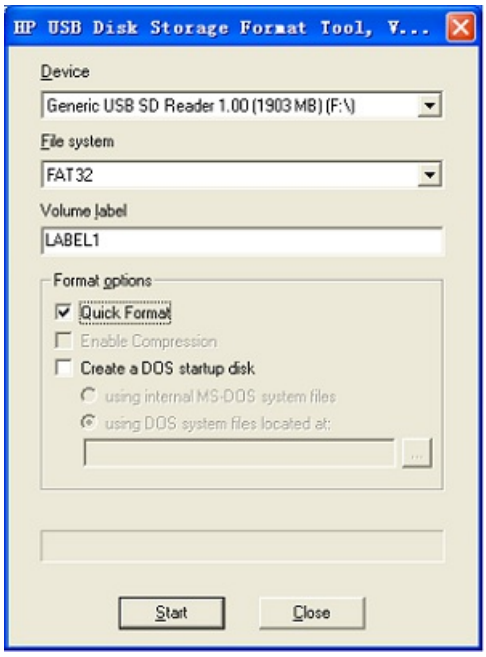

Figure 3-2 Format TF Card

- Copy `MLO` and `app` from *05-RTOS_Source\MYIRTECH\processor_sdk_rtos_am437x_3_01_00_06\prebuilt-sdcards\mydidk437x\sd_card_files* to the root of the prepared TF Card.

`MLO` is the bootloader of RTOS, `app` is the prebuilt EtherCAT slave RTOS application.

- Power off the MYD-C437X-PRU development board and set it to mmc boot mode by J5, connect the TTL Debug Interface J25 to PC with a USB to TTL converter and set the baudrate of host pc serial port to 115200-8-n-1.
- Power on the MYD-C437X-PRU development board, the EtherCAT slave application will run and output message to debug terminal on host pc as below. The details about the usage of this application will be explained in chapter 4.

```
StarterWare Boot Loader
BOARDInit status [0x0]
 SoC                   : [AM43XX]
 Core                  : [A9]
 Board Detected        : [MYDIDK437X]
 Base Board Revision   : [UNKNOWN]
 Daughter Card Revision: [UNKNOWN]
Copying application image from MMC/SD card to RAM
Jumping to StarterWare Application...

TI Industrial SDK Version : IASDK 2.1.2.2
Device name     : AM43XX
Chip Revision   : AM437x [PG1.2]
ARM Clock rate  : 600 MHz
Device Type     : EtherCAT Device

TI EtherCAT Demo Application Build - 3.5.0 - running on MYDIDK437X
SYNC0 task started

SYNC1 task started
```

3.1 Set Environment

TI has provided the processor_sdk_rtos_am437x_3_01_00_06 on Windows and Linux for RTOS development, we focus on the version of Windows in this document only.

After the installation of processor_sdk_rtos_am437x_3_01_00_06, The SDK build environment can be configured on Windows and Linux by running an environment setup script located in the top-level RTOS SDK folder. We can enter the sdk directory and set the environment as below:

```
C:\ti\processor_sdk_rtos_am437x_3_01_00_06>setupenv.bat
Optional parameter not configured : CG_XML_BIN_INSTALL_PATH
REQUIRED for xdc release build
Example: set CG_XML_BIN_INSTALL_PATH=C:/ti/cg_xml/bin
Optional parameter not configured : DOXYGEN_INSTALL_PATH
REQUIRED for xdc release build
Example: set DOXYGEN_INSTALL_PATH=C:/ti/Doxygen/doxygen/1.5.1-p1/bin
**********************************************************************
Environment Configuration:
    LIMIT_SOCS                : am437x
    LIMIT_BOARDS              : evmAM437x idkAM437x skAM437x mydidk437x
    PDK_INSTALL_PATH          : /ti/PDK_AM~1/packages/
    C6X_GEN_INSTALL_PATH      : C:/ti/ccsv6/tools/compiler/ti-cgt-c6000_8.1.0
    TOOLCHAIN_PATH_A15        : C:/ti/ccsv6/tools/compiler/gcc-arm-none-eabi-4_9-2015q3
    TOOLCHAIN_PATH_A8         : C:/ti/ccsv6/tools/compiler/gcc-arm-none-eabi-4_9-2015q3
    TOOLCHAIN_PATH_A9         : C:/ti/ccsv6/tools/compiler/gcc-arm-none-eabi-4_9-2015q3
    TOOLCHAIN_PATH_M4         : C:/ti/ccsv6/tools/compiler/ti-cgt-arm_15.12.1.LTS
    FPULIB_PATH               : C:/ti/ccsv6/tools/compiler/gcc-arm-none-eabi-4_9-2015q3
/lib/gcc/arm-none-eabi/4.9.3/fpu
    CROSS_TOOL_PRFX           : arm-none-eabi-
    XDC_INSTALL_PATH          : C:/ti/xdctools_3_32_00_06_core
    BIOS_INSTALL_PATH         : C:/ti/bios_6_45_01_29
    IPC_INSTALL_PATH          : C:/ti/ipc_3_43_02_04
    EDMA3LLD_BIOS6_INSTALLDIR : C:/ti/edma3_lld_2_12_01_25
    NDK_INSTALL_PATH          : C:/ti/ndk_2_24_03_35
    IMGLIB_INSTALL_PATH       : C:/ti/imglib_c66x_3_1_1_0
    UIA_INSTALL_PATH          : C:/ti/uia_2_00_03_43
    IPC_PLATFORM: UNKNOWN
    IPC_ALT_PLATFORM:
    PROC_SDK_INSTALL_PATH     : C:/ti/processor_sdk_rtos_am437x_3_01_00_06
**********************************************************************
Changing to short name to support directory names containing spaces
current directory: C:/ti/processor_sdk_rtos_am437x_3_01_00_06
PROCESSOR SDK BUILD ENVIRONMENT CONFIGURED
**********************************************************************
```

path2dos is located in xdctools. If you get the following error during running setupenv.bat, please add *C:\ti\xdctools_3_32_01_22_core\packages\xdc\services\io\release* into system variables or user variables, and restart the Windows command shell.

```
'path2dos'  is not recogzed as an internal or external command
```

The SDK level makefile can be used to compile SDK sub-components after the build environment has been configured. The Processor SDK for RTOS provides the following targets by default which will invoke the corresponding component targets. We can use `gmake help` to check the supported targets.

```
C:\ti\processor_sdk_rtos_am437x_3_01_00_06>gmake help
Standard Targets:
    help      - Prints target information
    all       - Builds all Component targets
    clean     - Cleans all Component targets

Component Targets:
    pdk       - Builds all targets within the PDK top-level makefile
    pdk_clean - Cleans all targets within the PDK top-level makefile

    NOTE: Instructions for rebuilding all other components
          installed with Processor SDK can be found in each
          component's sub-directory
```

3.2 Build Bootloader

Bootloader of TI SYSBIOS-base RTOS is located in startware, it supports power-on-reset bootstraps for the board. It initializes board, loads application from the memory device to DDR and transfers control to application.

Source Reference:

```
C:\ti\pdk_am437x_1_0_4\packages\ti\starterware\bootloader
```

Bootloader execution sequence:

```
1. Power on Reset
2. ROM Bootloader(RBL)
     - Platform configuration and initialization
            - DPLL and clock settings for MPU, I2C, MMCSD, USB, SPI, QSPI, Ethernet etc.
     - Checks Sysboot pins and choose booting device
            - If no valid bootloader found on booting device, RBL checks for next booting
              device. The sequence depends on RBL execution flow and Sysboot pins
     - RBL gets image size and load address by checking TI Image Header appended on
     bootloader binary(.bin). Check binary formats.
     - Loads the binary to internal memory at the Load address fetched from TI Image Header
     - Passes control to Secondary Bootloader(SBL)
3. Secondary Bootloader(SBL)
     - Configure PLL and Initialize DDR
     - Configure PRCM and PinMux  for Boot Peripherals
     - Copies application image to DDR
     - Passes execution control to Application
4. Application execution
```

Build Bootloader:

After setup the environment by setupenv.bat, we can build bootloader in startware by running gmake as below:

```
C:\ti\processor_sdk_rtos_am437x_3_01_00_06>gmake
```

After compiling bootloader, we will get serval binary files under *C:\ti\pdk_am437x_1_0_4\packages\ti\starterware\binary\bootloader\bin\am43xx-evm\gcc* directory as shown below.

```
bootloader_boot_mmcsd_a9host_debug.bin
bootloader_boot_mmcsd_a9host_debug_ti.bin
bootloader_boot_mmcsd_a9host_release.bin
bootloader_boot_mmcsd_a9host_release_ti.bin
bootloader_boot_nand_a9host_debug.bin
bootloader_boot_nand_a9host_debug_ti.bin
bootloader_boot_nand_a9host_release.bin
bootloader_boot_nand_a9host_release_ti.bin
bootloader_boot_qspi_a9host_debug.bin
bootloader_boot_qspi_a9host_debug_ti.bin
bootloader_boot_qspi_a9host_release.bin
bootloader_boot_qspi_a9host_release_ti.bin
bootloader_boot_uart_a9host_debug.bin
bootloader_boot_uart_a9host_debug_ti.bin
bootloader_boot_uart_a9host_release.bin
bootloader_boot_uart_a9host_release_ti.bin
```

Each binary file is corresponding to a boot mode as the file name suggests. The following table shows the meaning of different binary formats.

Binary format	Mode	Details
.bin	QSPI bootloader,UART bootloader and app	QSPI bootloader binary should in .bin format and Application binary should be in _ti.bin format. UART bootloader and applications are loaded in .bin format through XMODEM in terminal.
_ti.bin	MMCSD bootloader and app,NAND bootloader and app,MCSPI boot loader and app,QSPI app.	.bin binaries are converted to _ti.bin format by adding Image size(4bytes) and Image load address(4bytes) as image header. MMCSD, NAND and MCSPI boot loaders and application binaries should be appended with ti image header.

MMC Boot Mode:

bootloader_boot_mmcsd_a9host_debug_ti.bin or bootloader_boot_mmcsd_a9host_release_ti.bin is used for booting via tf/sd card. It should be renamed to `MLO` and be copied to the root directory of prepared tf/sdcard.

Debug Bootloader:

Bootloader in startware can be debugged by CCS. Firstly, we should import bootloader to CCS by choosing main menu of CSS: `File->Import->CCS Projects` , then choose `Select search-directory` directory as *C:\ti\pdk_am437x_1_0_4\packages\ti\starterware*. Combined with bootloader, we should import the following dependent libraries: board,dal,soc,utils,ff9b_lib, mmcsd_lib, xmodem_lib, qspi_lib, device, nand_lib.

> Note: Choose `Automatically import reference projects found in same search-directory` checkbox.

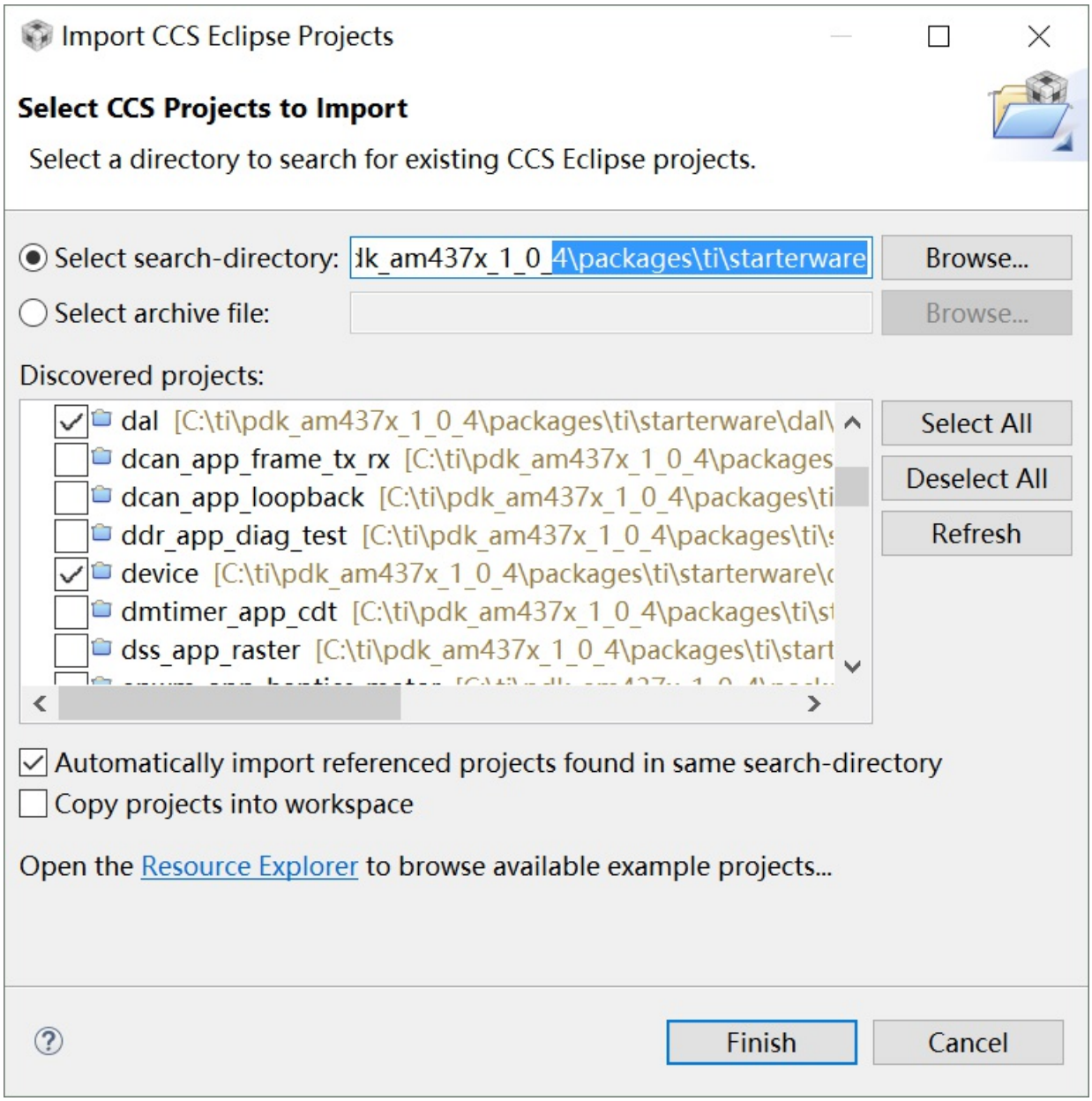

Figure 3-2-1 Import Bootloader

After importing bootloader, right clikc bootloader project in `Project Explorer` window, and then choose `Build Configurations->Set Active->AM43xx_mmcsd_boot_debug` in the popup menu. Right clikc bootloader project and dependent project in `Project Explorer` window, and then choose `Build Project` to build bootloader.

Before debugging bootloader, we need to set debug configuration for it.

```
1. Right click bootloader project, choose `Debug As->Debug Configurations...` in the popup menu,
   then `Debug Configurations` sub windows popups
2. Right clikc `Code Composer Studio-Device Debugging` item in `Debug Configurations` window,
   and choose `New` in the popup menu
3. Name the configuration to `bootloader`, and set the parameters as shown in figure 3-2-3
   and figure 3-2-4 below
4. Choose `Apply` to save the configuration
5. Chose `Debug` to enter debug view of bootloader as shown in figure 3-2-4 below.
     then we can debug bootloader step by step
```

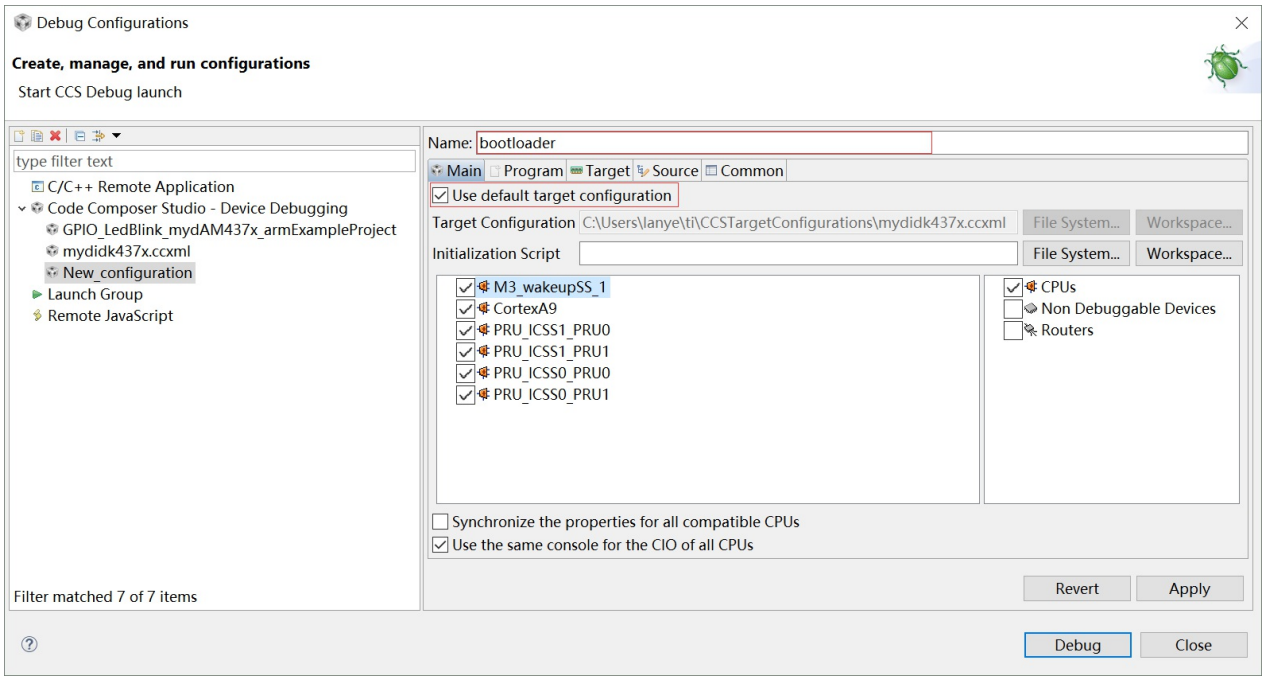

Figure 3-2-2 Debug Configurations Settings-1 of Bootloader

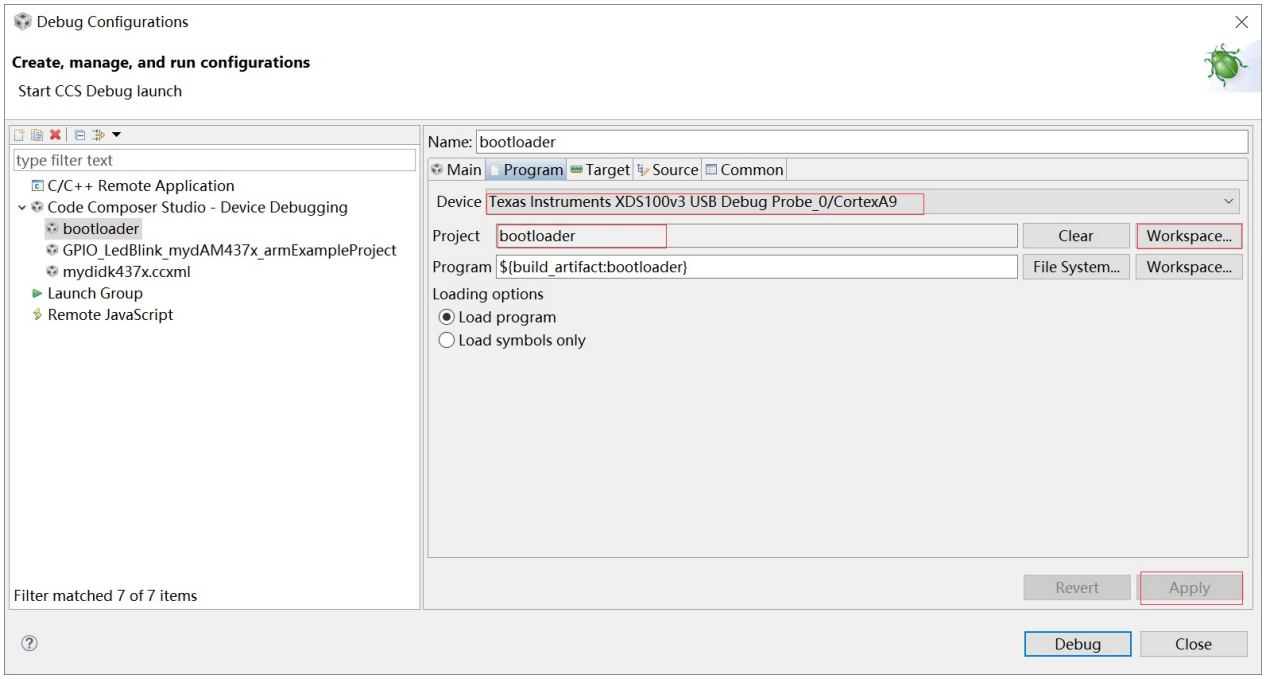

Figure 3-2-3 Debug Configurations Settings-2 of Bootloader

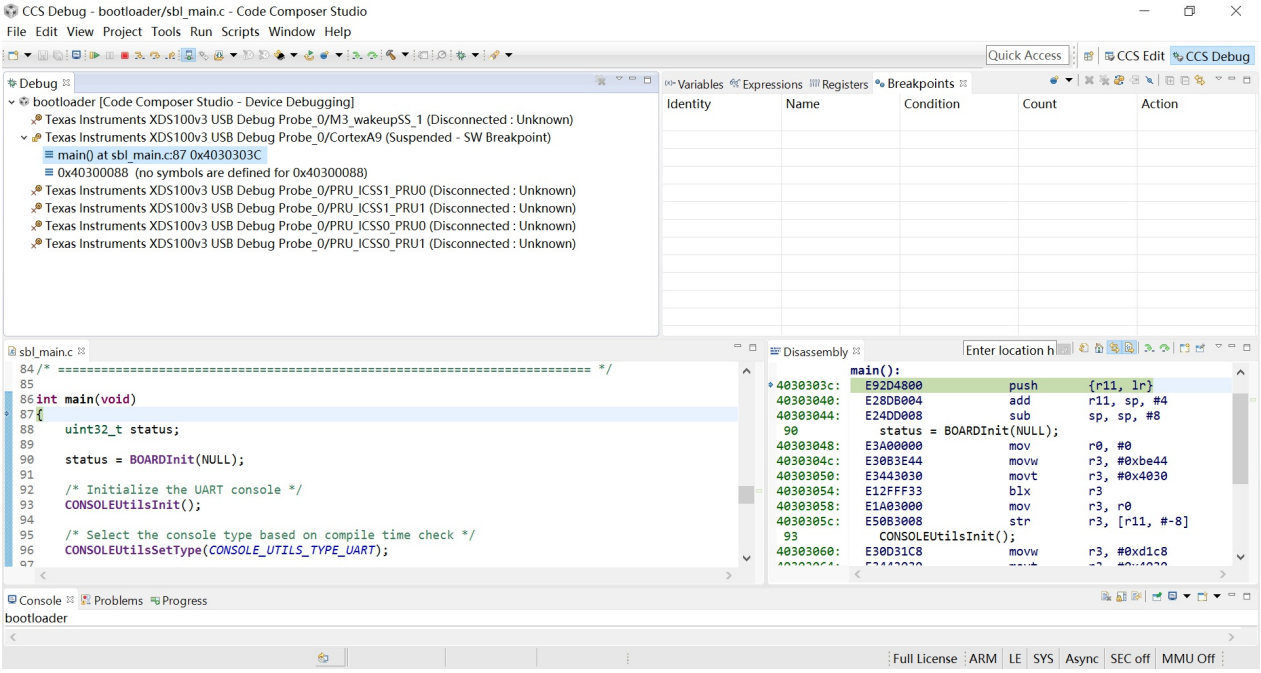

Figure 3-2-4 Debug View of Bootloader

After finish debugging bootloader on CCS, please build bootloader in Windows command shell with gmake for production.

3.3 Build RTOS Application

The following section covers the creation, compiling, debugging and running of a RTOS application.

PDK Introduction:
The content of PDK in processor_sdk_rtos_am437x_3_01_00_06 is installed at *C:\ti\pdk_am437x_1_0_4*, it includes some CCS plugins, support packages and examples for SYSBIOS-based RTOS.
We also put the industrial communication examples extracted from sysbios_ind_sdk_02.01.02.02 to the PDK. Firstly, we introduce three important scripts in *C:\ti\pdk_am437x_1_0_4\packages*.

```
pdksetupenv       -- Set the environment of PDK
pdkProjectCreate  -- Create RTOS application project
pdkAppImageCreate -- Create RTOS application image after finish compiling the project,
                     and rename it to `app`
```

Create PDK Project:

We use pdkProjectCreate.bat script to create PDK projects on Windows as shown below:

```
pdkProjectCreate.bat [soc] [board] [endian] [module] [processor] [pdkDir]
 Description:     (first option is default)
  soc         -   AM335x / AM437x / AM571x / AM572x
  board       -   all
                 -or-
                  Refer to "pdk_<soc>_<version>\packages\ti\board\lib
                  for valid board inputs for the soc
  endian      -   little / big
  module      -   all
                  -or-
                  fatfs / gpio / i2c / icss_emac / mmcsd / nuimu / nimu_icss /
                  pcie / pruss / spi / uart / usb
  processor   -   arm / dsp / m4
  pdkDir      -   THIS FILE LOCATION / "C:\ti\pdk_<device>_<version>\packages"
  Example:
  a) pdkProjectCreate.bat
             - Creates all module projects for the AM335x soc for arm little endian
  b) pdkProjectCreate.bat AM437x
             - Creates all module projects for the AM437x soc for arm little endian
  c) pdkProjectCreate.bat AM437x mydAM437x
             - Creates all module projects for mydAM437x device for arm little endian
  d) pdkProjectCreate.bat AM572x evmAM572x little
             - Creates all module projects for evmAM571x device for arm little endian
  e) pdkProjectCreate.bat AM571x evmAM571x little i2c dsp
             - Creates i2c module projects for evmAM571x device for dsp little endian
```

Currently, PDK provides a GPIO controlled LED RTOS applicaton for MYD-C437X-PRU develoment board，customers can use the following command to create the CCS project.

```
C:\ti\PDK_AM~1\packages>pdkProjectCreate.bat AM437x mydAM437x
================================================================================
Configuration:
   SOC             :   AM437x
   BOARD           :   mydAM437x
   ENDIAN          :   little
   MODULE          :   all
   PROCESSOR       :   arm
   PDK_SHORT_NAME  :   C:\ti\PDK_AM~1\packages\
================================================================================
Checking Configuration...
Complete
================================================================================
   PDK_PARTNO         : AM437
   PDK_ECLIPSE_ID     : com.ti.pdk.am437x
   RTSC_PLATFORM_NAME : ti.platforms.evmAM437X
   RTSC_TARGET        : gnu.targets.arm.A9F
   CCS_DEVICE         : "Cortex A.AM4379.IDK_AM437X"
********************************************************************************
Detecting all projects in PDK and importing them in the workspace
 C:\ti\PDK_AM~1\packages\\MyExampleProjects
Detected Test Project: GPIO_LedBlink_mydAM437x_armExampleProject
--------------------------------------------------------------------------------
Creating project 'GPIO_LedBlink_mydAM437x_armExampleProject'...
Done!
Copying macro.ini
Copied         1 File.
No projects detected
Project generation complete
********************************************************************************
```

Now, a GPIO controlled LED RTOS applicaton was created. The CCS project files are loacted in *C:\ti\PDK_AM~1\packages\MyExampleProjects\GPIO_LedBlink_mydAM437x_armExampleProject*.

Compile and Run the PDK Project:

- Import the PDK project to CCS as below:
    1. Open CCS
    2. Choose `File->Import` in CCS main menu, the `Select search-directory` dialog popups as shown in Figure3-3-1 below.
    3. Select search-directory C:\ti\PDK_AM~1\packages\MyExampleProjects\ GPIO_LedBlink_mydAM437x_armExampleProject
    4. Finish importing

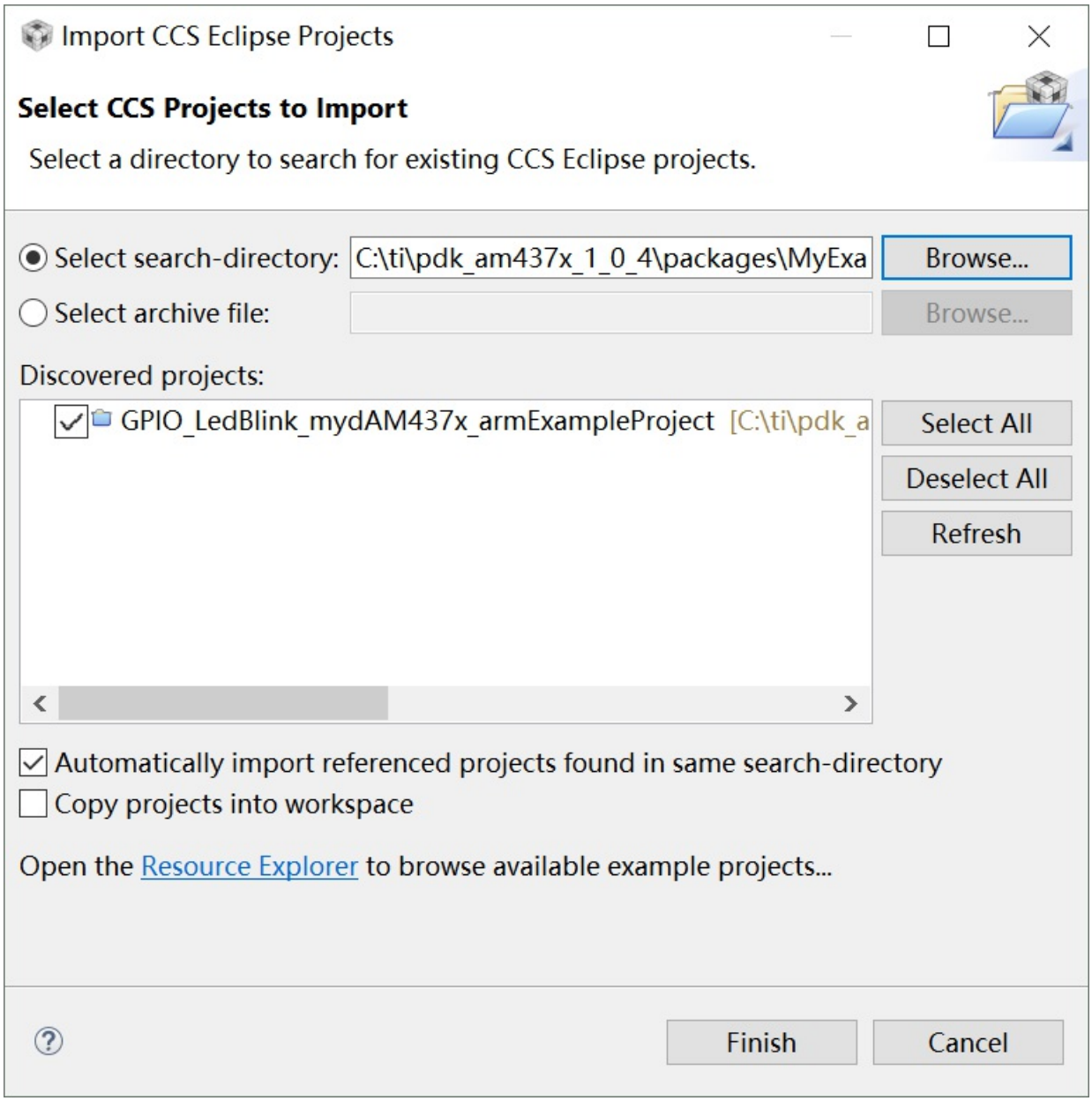

Figure 3-3-1 Import gpio_ledblik Project

- Compile the PDK Project

Right click the imported project in CCS `Project Explorer`, and then choose `Build Project` item of the popup menu. During compiling the project, if there is an error shown as below, please right click the imported project, and then choose `Properties` to set the project properties.

Choose `Build->GNU Compiler->Runtime` and set `-fmpu` to `neon` as shown in figure 3-3-2 below:

```
error: GPIO_LedBlink_mydAM437x_armExampleProject.out uses VFP register arguments,
```

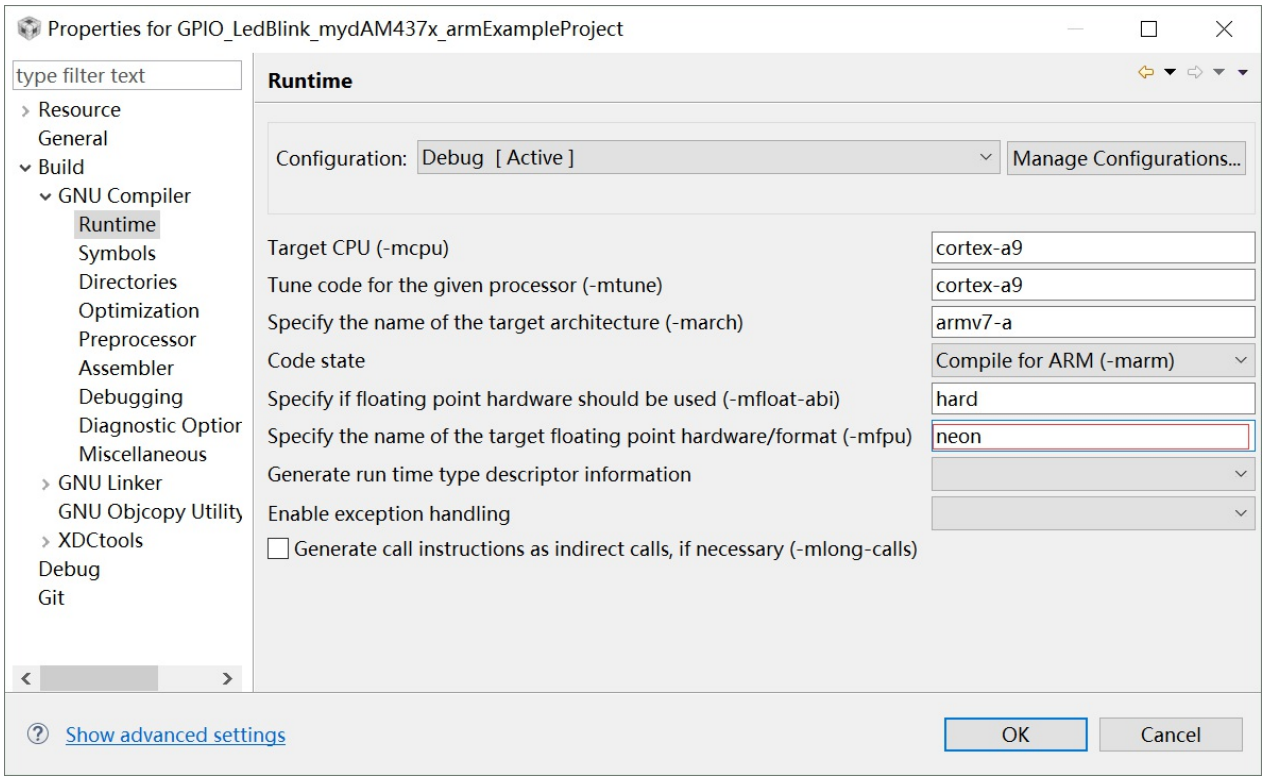

Figure 3-3-2 Project Properties Settings for gpio_ledblink

- Debug the PDK Project

After finish compiling the project successfully, we can debug the project. Before debugging the project, we need to set debug configuration for it:

1. Right click the project, choose `Debug As->Debug Configurations...` in the popup menu, then `Debug Configurations` sub windows popups
2. Right clikc `Code Composer Studio-Device Debugging` item in `Debug Configurations` window, and choose `New` in the popup menu
3. Name the configuration to `GPIO_LedBlink_mydAM437x_armExampleProject`, and set the parameters as shown in figure 3-3-3 and figure 3-3-4 below
4. Choose `Apply` to save the configuration
5. Chose `Debug` to enter debug view of GPIO_LedBlink_mydAM437x_armExampleProject as shown in figure 3-2-4 below. then we can debug step by step

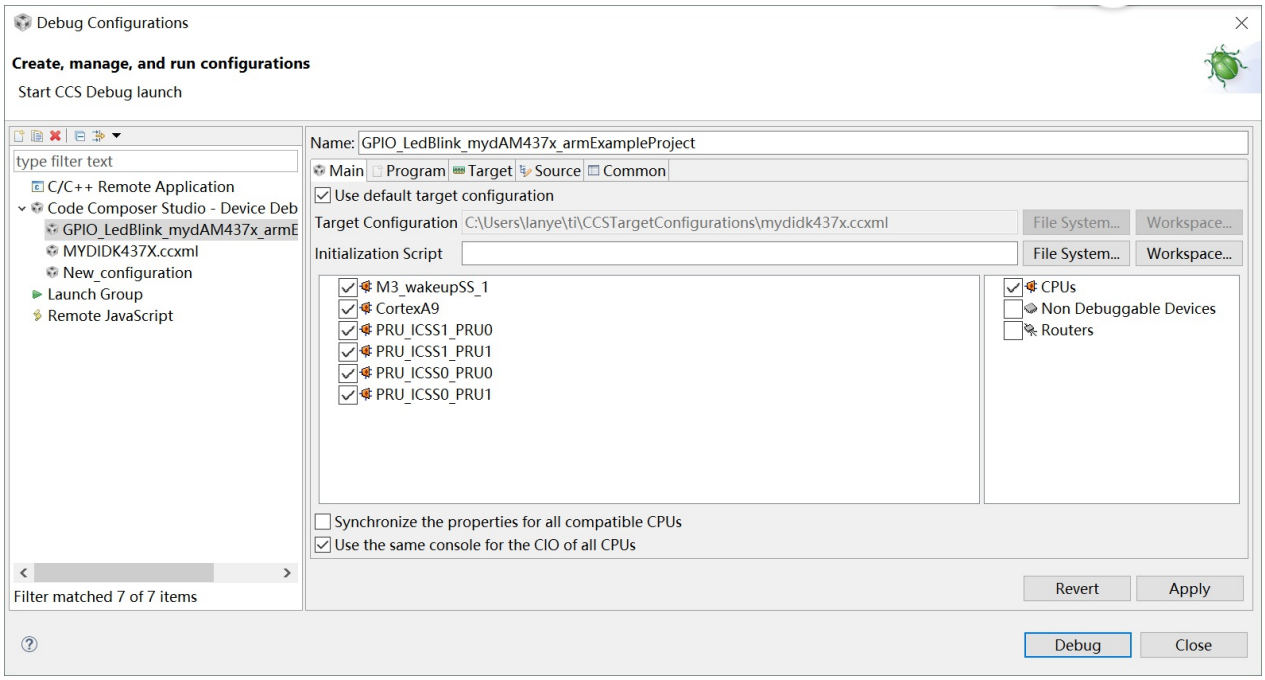

Figure 3-3-3 Debug Configurations Settings-1 of gpio_ledblink

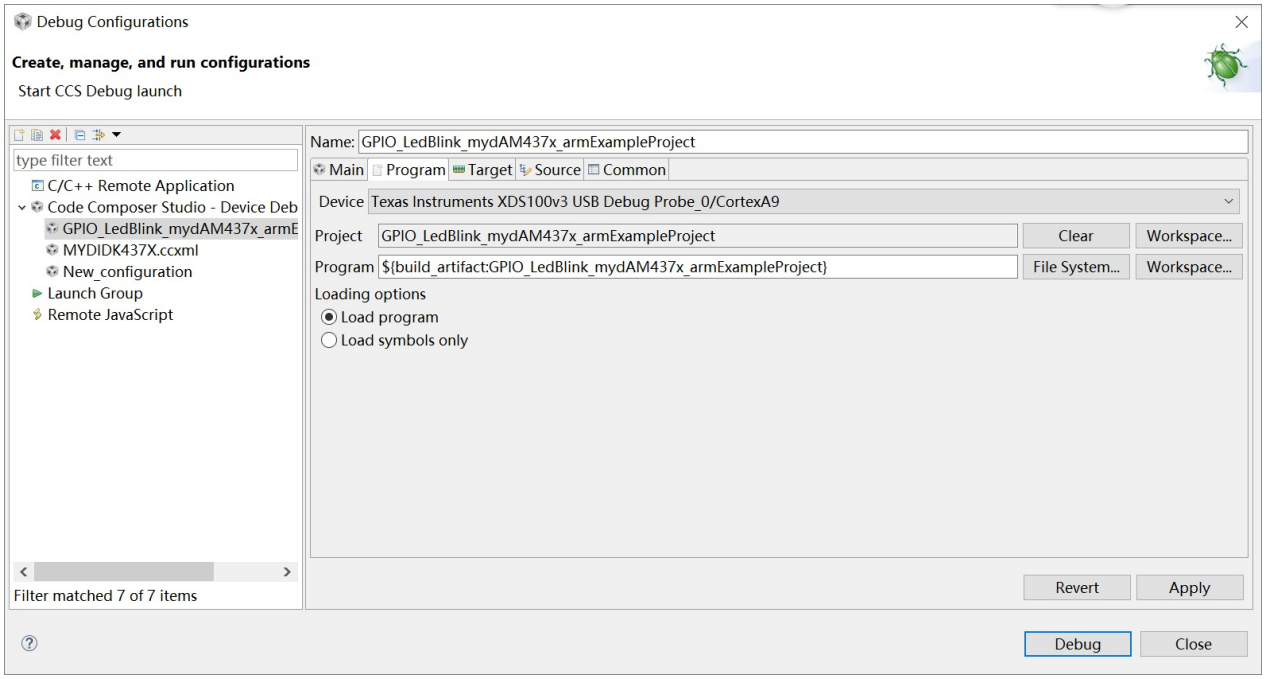

Figure 3-3-4 Debug Configurations Settings-2 of gpio_ledblink

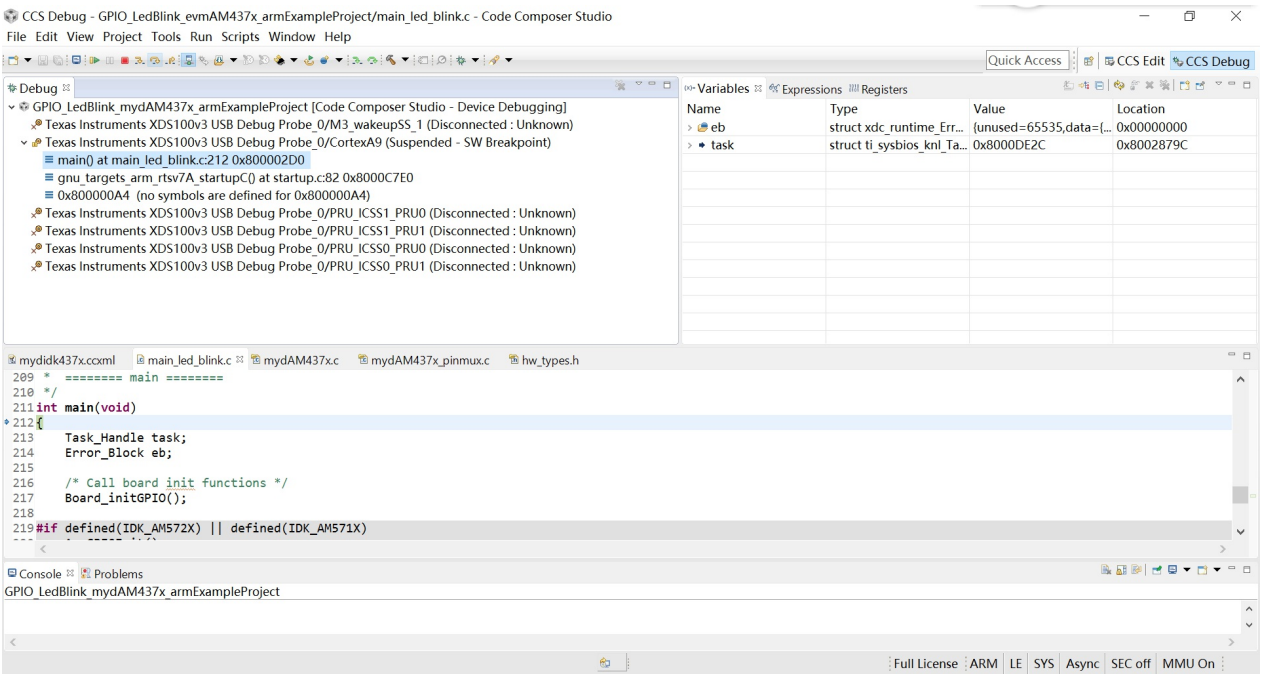

Figure 3-3-5 Debug View of gpio_ledblink

- Run the PDK Project

We can run the PDK project by press `F8` shortcut key during debugging in CSS.

The compiled program image is located in the _C:\ti\pdk_am437x_1_0_4\packages\MyExampleProjects\GPIO_LedBlink_mydAM437x_armExampleProject\Debug _directory

```
app                                          main_led_blink.d
ccsObjs.opt                                  main_led_blink.o
configPkg/                                   makefile
GPIO_LedBlink_mydAM437x_armExampleProject.bin  objects.mk
GPIO_LedBlink_mydAM437x_armExampleProject.map  sources.mk
GPIO_LedBlink_mydAM437x_armExampleProject.out  subdir_rules.mk
GPIO_log.d                                   subdir_vars.mk
GPIO_log.o                                   UART_soc.d
GPIO_mydAM437x_board.d                       UART_soc.o
GPIO_mydAM437x_board.o
```

The app file can be directly used to program the QSPI Flash and TF card. This file is generated by the GPIO_LedBlink_mydAM437x_armExampleProject.bin file and then adds the file header information. The generation process is as follows:

```
tiimage.exe 0x80000000 NONE GPIO_LedBlink_mydAM437x_armExampleProject.bin app
```

tiimage.exe is located in the *C:\ti\pdk_am437x_1_0_4\packages\ti\starterware\tools\ti_image* directory. The routines have completed the app generation step automatically when compiling.

For production, we can copy `MLO' and the` app` file build in CCS to prepared TF Card. Power off the MYD_C437X_PRU development board and set J5 to mmc boot mode,
Power on the MYD-C437X-PRU development board, the gpio_ledblink application will run, we can get the following output message in host pc debug terminal.

```
StarterWare Boot Loader
BOARDInit status [0x0]
 SoC                   : [AM43XX]
 Core                  : [A9]
 Board Detected        : [MYDIDK437X]
 Base Board Revision   : [UNKNOWN]
 Daughter Card Revision: [UNKNOWN]
Copying application image from MMC/SD card to RAM
Jumping to StarterWare Application...

GPIO Led Blink Application
```

Press and release the `S3` keypad on MYD-C437X-PRU development board circularly, the blue LED(D36) will toggle on and off.

3.4 Start method

The MYD-C437X-PRU development board factory image file is located in the 02-Images\rtos-images directory of the CD-ROM. The QSPI directory is the image of the ethercat_slave application for QSPI programming and startup,The TF directory is the ethercat_slave application image for the TF card startup.

TF card startup

TF card startup requires two files, a boot file named MLO, and an application file named app.

Both boot and application files are files ending in *_ti.bin.

The boot file is located at *C:\ti\pdk_am437x_1_0_4\packages\ti\starterware\binary\bootloader\bin\am43xx-evm\gcc*

```
bootloader_boot_mmcsd_a9host_debug_ti.bin
bootloader_boot_mmcsd_a9host_release_ti.bin
```

The debug version is the same as the release version, except that the debug version has more information for CCS simulation debugging. Copy one of the files to the SD card and rename it MLO.

The application file uses the QSPI program as an example. The file is located under *C:\ti\pdk_am437x_1_0_4\packages\ti\starterware\binary\qspi_app_flash_writer\bin\am43xx-evm\gcc*

```
qspi_app_flash_writer_a9host_debug_ti.bin
qspi_app_flash_writer_a9host_release_ti.bin
```

The debug version is the same as the release version, except that the debug version has more information for CCS simulation debugging. Copy one of the files to the SD card and rename it app.

QSPI startup

QSPI boot is divided into QSPI programming and QSPI program running.

QSPI programming is also done with the help of TF card booting. The TF card contains the following files.

```
app
boot
config
ethercat
MLO
```

MLO is the TF boot file, and the app is the TF card boot QSPI application file.

The specific source of MLO files and app files has been described in the above TF card startup.

Config is a configuration file written by QSPI. It contains QSPI boot file and QSPI application programming information. The configuration file is as follows:

```
boot 0x0
ethercat 0x80000
```

The boot file is located in the *C:\ti\pdk_am437x_1_0_4\packages\ti\starterware\binary\bootloader\bin\am43xx-evm\gcc* directory:

```
bootloader_boot_qspi_a9host_debug.bin
bootloader_boot_qspi_a9host_release.bin
```

Copy one of the files to the SD card and rename it boot.

The ethercat application file is located under the *C:\ti\pdk_am437x_1_0_4\packages\ti\examples\ethercat_slave\am437x_release* directory:

```
ethercat_ti.bin
```

Copy the file to the SD card and rename it ethercat.

After the above documents are ready, insert the TF card into the development board, set the development as TF card startup, and give the development board the power. From the serial port, see the following information to show that the programming is completed.

```
StarterWare Boot Loader
BOARDInit status [0x0]
 SoC                  : [AM43XX]
 Core                 : [A9]
 Board Detected       : [MYDIDK437X]
 Base Board Revision  : [UNKNOWN]
 Daughter Card Revision: [UNKNOWN]
Copying application image from MMC/SD card to RAM
Jumping to StarterWare Application...


StarterWare QSPI Flash Writer!!
BOARDInit status [0x0]
 SoC                  : [AM43XX]
 Core                 : [A9]
 Board Detected       : [MYDIDK437X]
 Base Board Revision  : [UNKNOWN]
 Daughter Card Revision: [UNKNOWN]
Flash Manufacturer ID = ef
Device ID = 17
Copying boot to QSPI Flash
Copying ethercat to QSPI Flash
Changing read to quad mode
Read mode has been changed to Quad mode
SUCCESS!!!
Flashing completed
```

After the programming is completed, set the boot mode of the development board to QSPI mode. Unplug the SD card, power off, and then power on. See the following information from the serial port to show that the QSPI startup is successful.

```
StarterWare Boot Loader
BOARDInit status [0x0]
 SoC                  : [AM43XX]
 Core                 : [A9]
 Board Detected       : [MYDIDK437X]
 Base Board Revision  : [UNKNOWN]
 Daughter Card Revision: [UNKNOWN]

Copying Header of the application image
Copying image from flash to DDR

Jumping to StarterWare Application...

TI Industrial SDK Version : IASDK 2.1.2.2
Device name     : AM43XX
Chip Revision   : AM437x [PG1.2]
ARM Clock rate  : 600 MHz
Device Type     : EtherCAT Device

TI EtherCAT Demo Application Build - 3.5.0 - running on MYDIDK437X
SYNC0 task started

SYNC1 task started
MYDIDK437X Industrial SDK LED= 0
```

4. Industrial Application

MYD-C437X-PRU development board provides a evaluation platform especially designed for industrial communication. In TI's sysbios_ind_sdk_02.01.02.02, there are serval examples for industrial communication, this section focuses on the most typical EtherCAT slave application.

Hardware Preparation:

- One host PC with ethernet，2GB RAM or above
- One MYD-C437X-PRU development board
- One CAT5 ethernet cable
- One USB to TTL converter to connect J25 of MYD_C437X_PRU and host PC, set the baudrate serial port of PC to 115200-8-n-1

Software Preparation:

- Windows 7/10 PC
- Beckhoff TWinCAT V3.1.4020.28
- CCS 6.2.0
- ethercat slave rtos applicatoin

Host PC running BeckhOFF TwinCAT V3.1.4020.28 works as EtherCAT master, a MYD-C437X-PRU development board running ethercat_slave application works as EtherCAT slave. For best performance - it is recommended to use a compatible NIC card listed Supported Network Controller by Beckhoff Ethernet Driver (http://infosys.beckhoff.com/index.php?content=../content/1031/tcsystemmanager/reference/ethercat/html/ethercat_supnetworkcontroller.htm&id=10112) . Note, please always check ethernet adapter is listed below "installed and ready to use devices" before attempting to run TwinCat demo.

Build EtherCAT Slave Application：

EtherCAT Slave application is built by CCS. Open CCS, choose `File->Import` , the `search-directory` dialog will pop up, choose *C:\ti\pdk_am437x_1_0_4\packages\ti\examples\ethercat_slave* in `search-directory` and import it to CCS. Then we can compile, debug and run the EtherCAT slave application as explained in Chapter 3-3.

> Note: EtherCAT slave application depends on some libraries in PDK, so we should compile PDK with gmake as shown below:

```
C:\ti\processor_sdk_rtos_am437x_3_01_00_06>setupenv.bat
C:\ti\processor_sdk_rtos_am437x_3_01_00_06>gmake pdk
```

or

```
C:\ti\pdk_am437x_1_0_4\packages>pdksetupenv.bat
C:\ti\pdk_am437x_1_0_4\packages>gmake
```

After building the `ethcat_slave` project, a binary file named with ethercat_ti.bin will be created in *C:\ti\pdk_am437x_1_0_4\packages\ti\examples\ethercat_slave\am437x_debug*, then we can rename `ethercat_ti.bin` to `app` , copy it with `MLO` to prepared TF card and run EtherCAT slave application from TF Card.

Test Steps:

To test TI EtherCAT slave sample app, please refer to the following steps:

- Install TwinCAT V3.1.4020.28

A one month evaluation is available for free download from the Beckoff website TwinCAT 3 Download (http://www.beckhoff.co.in/english.asp?download/tc3-download-xae.htm). Select `the eXtended Automation Engineering (XAE)` mode of installation

- Communicaton between TwinCAT master and MYD-C437X-PRU, control the LEDs on MYD-C437X-PRU in TWinCAT as below:

1. Copy *C:\ti\pdk_am437x_1_0_4\packages\ti\examples\ethercat_slave\esi\TiEtherCATLib.xml* to the install directory of TwinCAT `<Drive>:\TwinCAT\3.1\Config\Io\EtherCAT\` ; Copy*C:\ti\pdk_am437x_1_0_4\packages\ti\protocols\ethercat_slave\ecat_appl\esi\TI_ESC_CTT.xml* to the install directory of TwinCAT `<Drive>:\TwinCAT\3.1\Config\Io\EtherCAT\` .
2. Open TwinCAT XAE(VS2013)
3. Create a new TwinCAT XAE project: File->New Project->TwinCAT Project
4. Install TwinCAT RT Ethernet intermediate driver: TwinCAT->Show Real Time Ethernet Compatible Devices; choose the compatible ethercat and select Install to install
5. Connect CAT5 Ethernet cable from TwinCAT PC to EtherCAT IN/Port0 (J26) of MYD-C437X-PRU
6. Scan Boxes: In Solutin Explorer, choose the project just created->I/O-> right click `Devices->Scan` . If the `Scan Boxes` is gray, then choose `TWinCAT->Restart TwinCAT(Config Mode)` in main menu
7. TI Boxn(ti-esc) will be detected automatically. Now select Device1 (EtherCAT) and goto TwinCAT and then Reload Devices.
8. A dialog will pop asking Load I/O Devices, select Yes.Next dialog asks confirmation to Activate Free Run - select Yes. This will put TI ESC into OP mode as shown in figure 4-1. Alternatively, you can use "Toggle Free Run State" bottom.

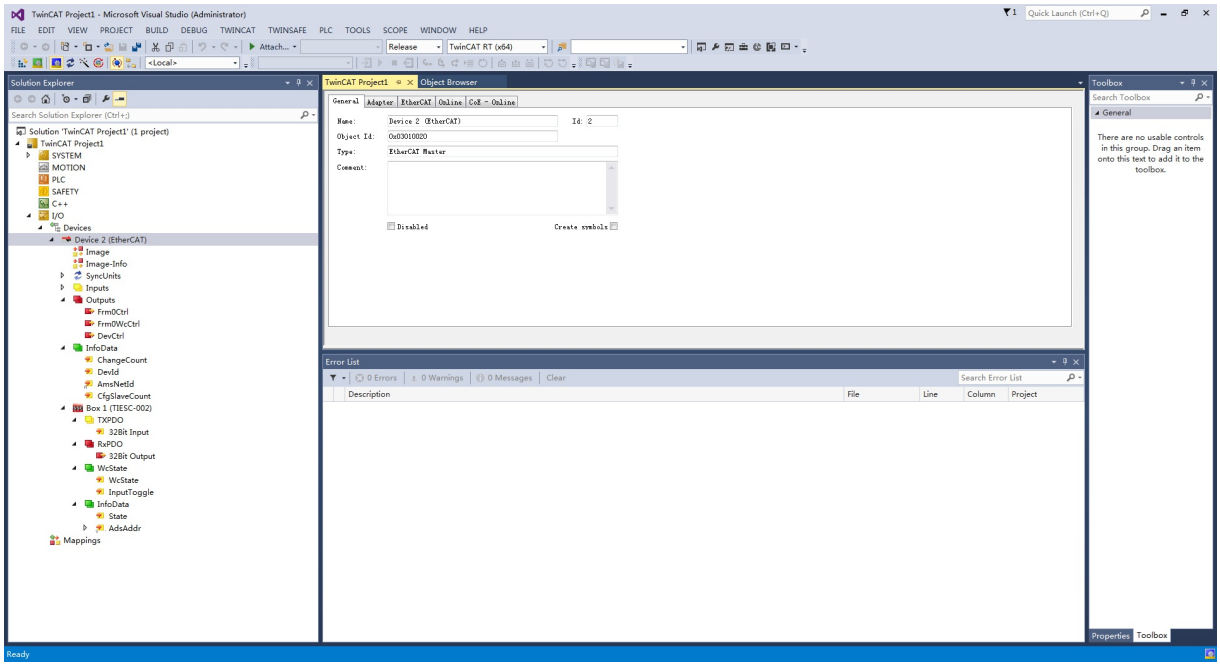

Figure 4-1 TI Boxn(ti-esc) detected in TwinCAT

9. Choose `Boxn(TI-ESC)->RxPDO->32Bit` , in the sub window at right, choose `Online->Write` , and write number 0~15 as shown in Figure 4-2 below. Observe the state of the blue LEDs on MYD_C437X_PRU development board. If you write number `15` , all the four blue LEDs will be turned on.

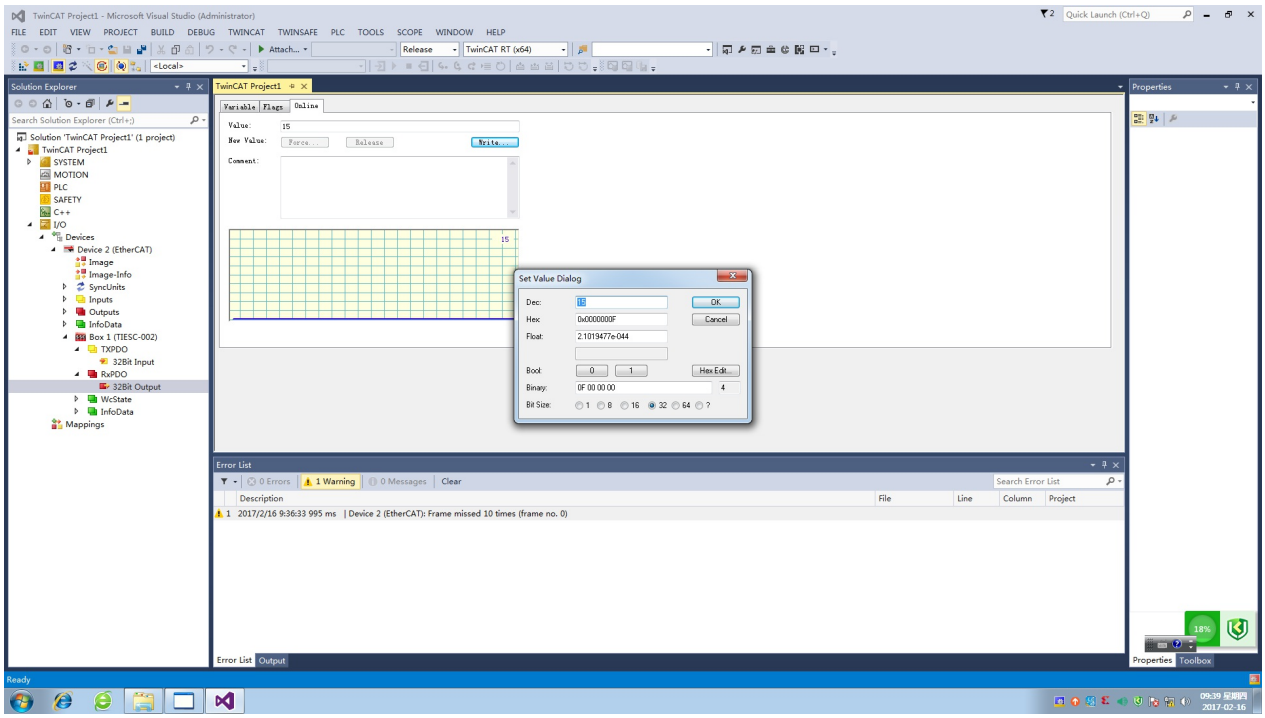

Figure 4-2 Control LEDs on MYD_C437X_PRU in TwinCAT

Appendix A Warranty & Technical Support Services

MYIR Tech Limited is a global provider of ARM hardware and software tools, design solutions for embedded applications. We support our customers in a wide range of services to accelerate your time to market.

MYIR is an ARM Connected Community Member and work closely with ARM and many semiconductor vendors. We sell products ranging from board level products such as development boards, single board computers and CPU modules to help with your evaluation, prototype, and system integration or creating your own applications. Our products are used widely in industrial control, medical devices, consumer electronic, telecommunication systems, Human Machine Interface (HMI) and more other embedded
applications. MYIR has an experienced team and provides custom design services based on ARM processors to help customers make your idea a reality.

The contents below introduce to customers the warranty and technical support services provided by MYIR as well as the matters needing attention in using MYIR's products.

Service Guarantee

MYIR regards the product quality as the life of an enterprise. We strictly check and control the core board design, the procurement of components, production control, product testing, packaging, shipping and other aspects and strive to provide products with best quality to customers. We believe that only quality products and excellent services can ensure the long-term cooperation and mutual benefit.

Price

MYIR insists on providing customers with the most valuable products. We do not pursue excess profits which we think only for short-time cooperation. Instead, we hope to establish long-term cooperation and win-win business with customers. So we will offer reasonable prices in the hope of making the business greater with the customers together hand in hand.

Delivery Time

MYIR will always keep a certain stock for its regular products. If your order quantity is less than the amount of inventory, the delivery time would be within three days; if your order quantity is greater than the number of inventory, the delivery time would be always four to six weeks. If for any urgent delivery, we can negotiate with customer and try to supply the goods in advance.

Technical Support

MYIR has a professional technical support team. Customer can contact us by email (support@myirtech.com), we will try to reply you within 48 hours. For mass production and customized products, we will specify person to follow the case and ensure the smooth production.

After-sale Service

MYIR offers one year free technical support and after-sales maintenance service from the purchase date. The service covers:

Technical support service

- MYIR offers technical support for the hardware and software materials which have provided to customers;

- To help customers compile and run the source code we offer;
- To help customers solve problems occurred during operations if users follow the user manual documents;
- To judge whether the failure exists;
- To provide free software upgrading service.

However, the following situations are not included in the scope of our free technical support service:

- Hardware or software problems occurred during customers' own development;
- Problems occurred when customers compile or run the OS which is tailored by themselves;
- Problems occurred during customers' own applications development;
- Problems occurred during the modification of MYIR's software source code.

After-sales maintenance service

The products except LCD, which are not used properly, will take the twelve months free maintenance service since the purchase date. But following situations are not included in the scope of our free maintenance service:

- The warranty period is expired;
- The customer cannot provide proof-of-purchase or the product has no serial number;
- The customer has not followed the instruction of the manual which has caused the damage the product;
- Due to the natural disasters (unexpected matters), or natural attrition of the components, or unexpected matters leads the defects of appearance/function;
- Due to the power supply, bump, leaking of the roof, pets, moist, impurities into the boards, all those reasons which have caused the damage of the products or defects of appearance;
- Due to unauthorized weld or dismantle parts or repair the products which has caused the damage of the products or defects of appearance;
- Due to unauthorized installation of the software, system or incorrect configuration or computer virus which has caused the damage of products.

Warm tips:

1. MYIR does not supply maintenance service to LCD. We suggest the customer first check the LCD when receiving the goods. In case the LCD cannot run or no display, customer should contact MYIR within 7 business days from the moment get the goods.
2. Please do not use finger nails or hard sharp object to touch the surface of the LCD.
3. MYIR suggests user purchasing a piece of special wiper to wipe the LCD after long time use, please avoid clean the surface with fingers or hands to leave fingerprint.
4. Do not clean the surface of the screen with chemicals.
5. Please read through the product user manual before you using MYIR's products.
6. For any maintenance service, customers should communicate with MYIR to confirm the issue first. MYIR's support team will judge the failure to see if the goods need to be returned for repair service, we will issue you RMA number for return maintenance service after confirmation.

Maintenance period and charges

- MYIR will test the products within three days after receipt of the returned goods and inform customer the testing result. Then we will arrange shipment within one week for the repaired goods to the customer. For any special failure, we will negotiate with customers to confirm the maintenance period.
- For products within warranty period and caused by quality problem, MYIR offers free maintenance service; for products within warranty period but out of free maintenance service scope, MYIR provides maintenance service but shall charge some basic material cost; for products out of warranty period, MYIR provides maintenance service but shall charge some basic material cost and handling fee.

Shipping cost

During the warranty period, the shipping cost which delivered to MYIR should be responsible by user; MYIR will pay for the return shipping cost to users when the product is repaired. If the warranty period is expired, all the shipping cost will be responsible by users.

Products Life Cycle

MYIR will always select mainstream chips for our design, thus to ensure at least ten years continuous supply; if meeting some main chip stopping production, we will inform customers in time and assist customers with products updating and upgrading.

Value-added Services

1. MYIR provides services of driver development base on MYIR's products, like serial port, USB, Ethernet, LCD, etc.
2. MYIR provides the services of OS porting, BSP drivers' development, API software development, etc.
3. MYIR provides other products supporting services like power adapter, LCD panel, etc.
4. ODM/OEM services.

MYIR Tech Limited

Address: Room 04, 6th Floor, Building No.2, Fada Road, Yunli Smart Park, Bantian, Longgang District, Shenzhen, Guangdong, China 518129

Support Email: support@myirtech.com

Sales Email: sales@myirtech.com

Phone: +86-755-22984836

Fax: +86-755-25532724

Website: www.myirtech.com