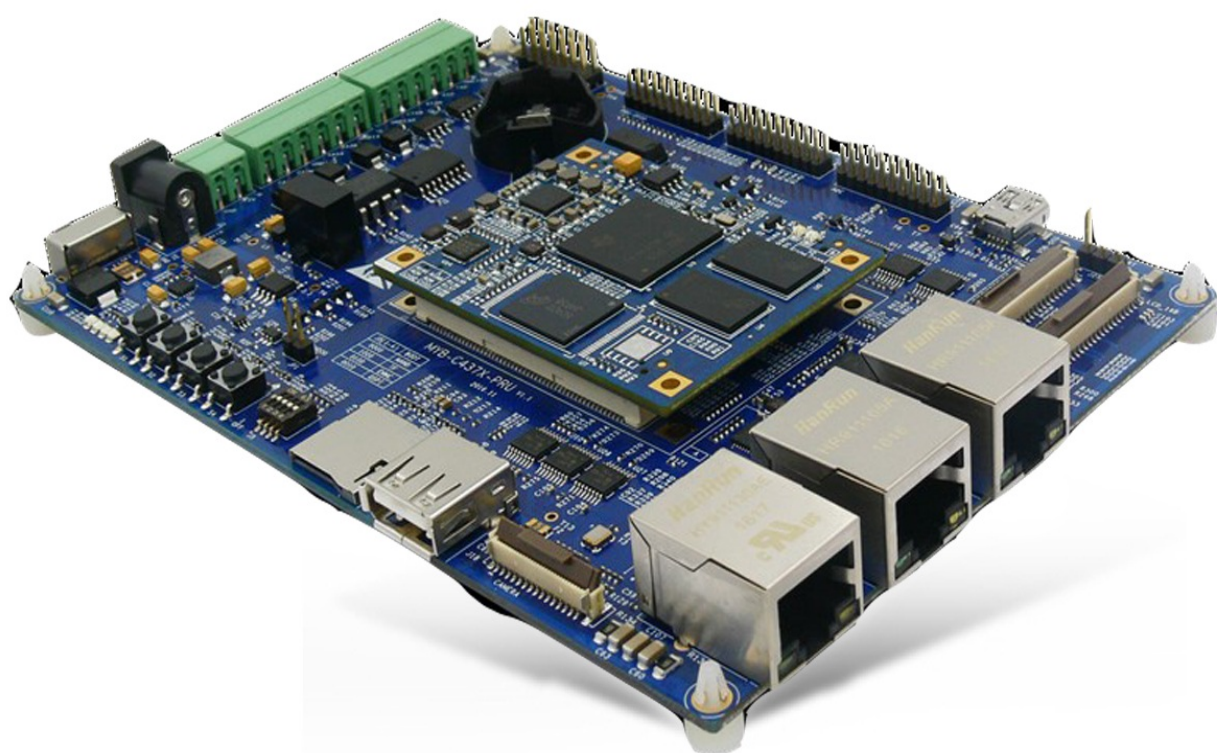


MYD-C437X-PRU RTOS 开发手册

MYiR™ Make Your Idea Real



目錄

前言	0
1. 软件资源介绍	1
2. 部署开发环境	2
3. 构建系统	3
3.1 设置环境变量	3.1
3.2 生成Bootloader	3.2
3.3 生成RTOS应用	3.3
3.4 启动方式	3.4
4. 工业协议应用	4
附录A	5
附录B	6

前言

本文主要讲述如何在MYD-C437X-PRU开发板上安装运行TI SYSBIOS实时系统以及SYSBIOS应用程序的开发流程。其中包括开发环境的搭建、源码编译、应用程序的实例分析、映像的下载等。

版本历史

版本号	描述	时间
V1.0	初始版本	2017.1.16
V2.0	增加启动方式章节，调整一些细节	2018.6.12

硬件版本

本文档仅适用于米尔MYD-C437X-PRU开发板。

1. 软件资源介绍

MYD-C437X-PRU开发板出厂附带TI SYSBIOS系统开发所需要的编译工具链，SYSBIOS内核和各驱动模块的源代码，以及各种配套的开发调试工具，应用开发样例等。

主要内容位于出厂附带资料**05-RTOS_Source**目录下。

本节内容以表格形式描述MYD-C437X-PRU SYSBIOS系统开发所需要的软件资源。

表 1-1 软件资源列表

类别	名称	描述	源码
引导程序	startware bootloader	负责系统初始化和引导SYSBIOS内核,加载SYSBIOS应用	YES
内核	sysbios	TI SYSBIOS系统内核bios_6_45_01_29	YES
驱动	emac	am437x emac 驱动	YES
驱动	gpio	gpio驱动	YES
驱动	gpmc	gpmc驱动	YES
驱动	i2c	i2c驱动	YES
驱动	icss_emac	icss emac驱动	YES
驱动	mcasp	mcasp驱动	YES
驱动	mmcsd	mmc host驱动	YES
驱动	pruss	pruss 驱动	YES
驱动	spi	spi驱动	YES
驱动	uart	uart驱动	YES
应用程序	ethercat_slave	EtherCAT slave 测试程序	YES
应用程序	ethernet_mac	ethernet mac测试程序	YES
应用程序	ethernetip_adapter	ethernet ip测试程序	YES
应用程序	example_utils	其他综合测试例程	YES
工具	xdctools	xdctools_3_32_01_22_core	exe
工具	Win32DiskImager	TF/SD镜像烧写工具	exe
工具	Format Tools	TF/SD格式化工具	exe
工具	CCS6	Code Composer Studio IDE 6.2.0	exe
工具	TWinCat	TC31-Full-Setup.3.1.4020.28	exe

2. 部署开发环境

本节主要介绍开发过程中所使用的软件开发环境选择，硬件调试环境的搭建和集成开发环境CCS的安装，配置以及验证。

硬件调试环境：

- PC主机一台以上
- XDS100V3 兼容仿真器一台
- MYD-C437X-PRU开发板一块
- TF卡一张，USB转TTL电平调试线，CAT5交叉网线各一根

MYD-C437X-PRU硬件调试环境搭建：

将TTL电平调试串口J25（图中Debug UART）通过USB转TTL电平调试线连到PC上，并设置PC端串口的波特率设为115200，数据位为8，停止位为1，无奇偶校验。网口J6(图中Giga Ethernet)对应AM437X EMAC0，网口J26和J27分别对应PRU扩展的PRUETH0和PRUETH1。

如果需要使用JTAG调试，请将仿真器连接到TI-JTAG接口J22。

具体如下图：

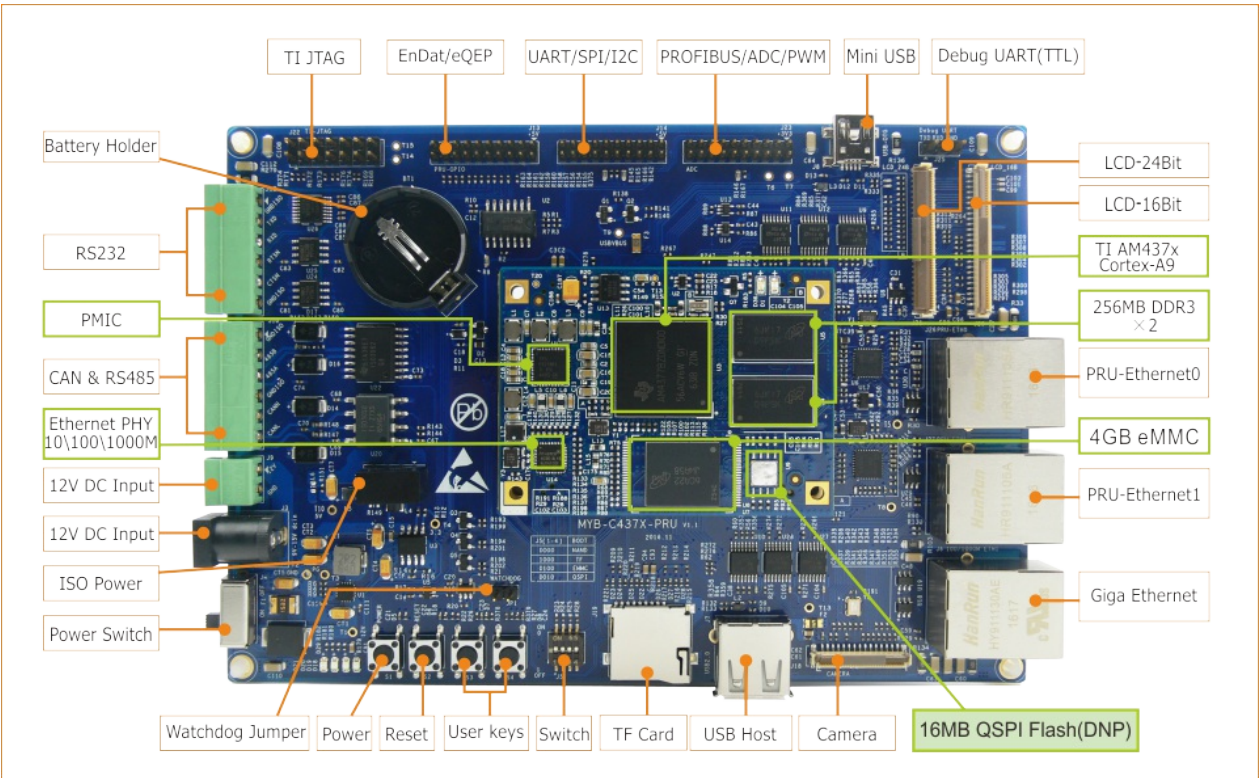


图2-1 MYD-C437X-PRU接口定义

软件开发环境：

- Ubuntu12.04 64位桌面版
- Windows7/Windows10主机
- Code Composer Studio IDE 6.2.0

安装Code Composer Studio IDE 6.2.0:

TI RTOS系统使用Code Composer Studio IDE作为主机端的开发和调试工具，TI的官网提供多个版本免费下载，本手册所使用的版本为6.2.0.00050。

用户可以参照TI官方的CCS指导手册进行安装，默认安装路径为c:\ti。针对AM437X平台的开发，安装的时候需要选择Sitara 32-bit ARM Processors产品系列对应的组件。如下图所示：

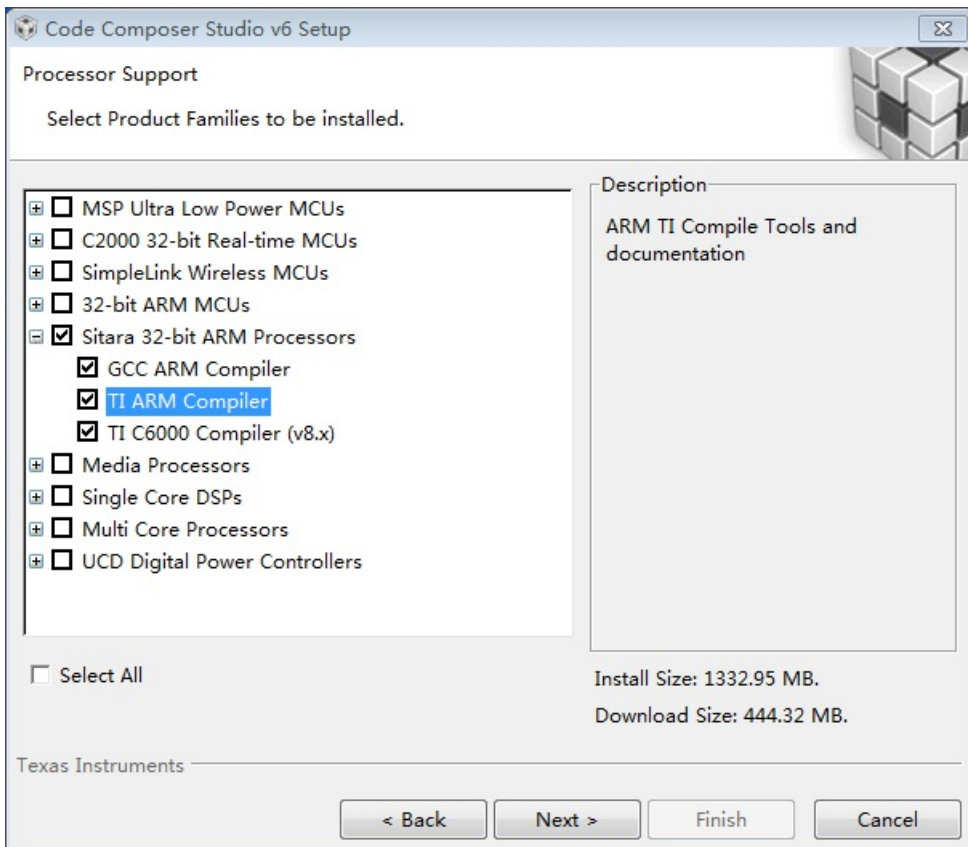


图2-2 CCS安装处理器选择

CCS安装过程中，会选择一些辅助的开发工具（add-ons），如下图所示，如果这里不选择，也可以在安装完成之后，通过CCS的App Center进行安装。



图2-3 CCS App Center工具选择

安装完成之后，CCS包含了JTAG调试组件支持，但是对于某些特定的设备，还需要单独添加一些设备支持文件，GEL文件。用户可以通过升级的方式从TI官网获取Sitara设备的支持。如下图所示：

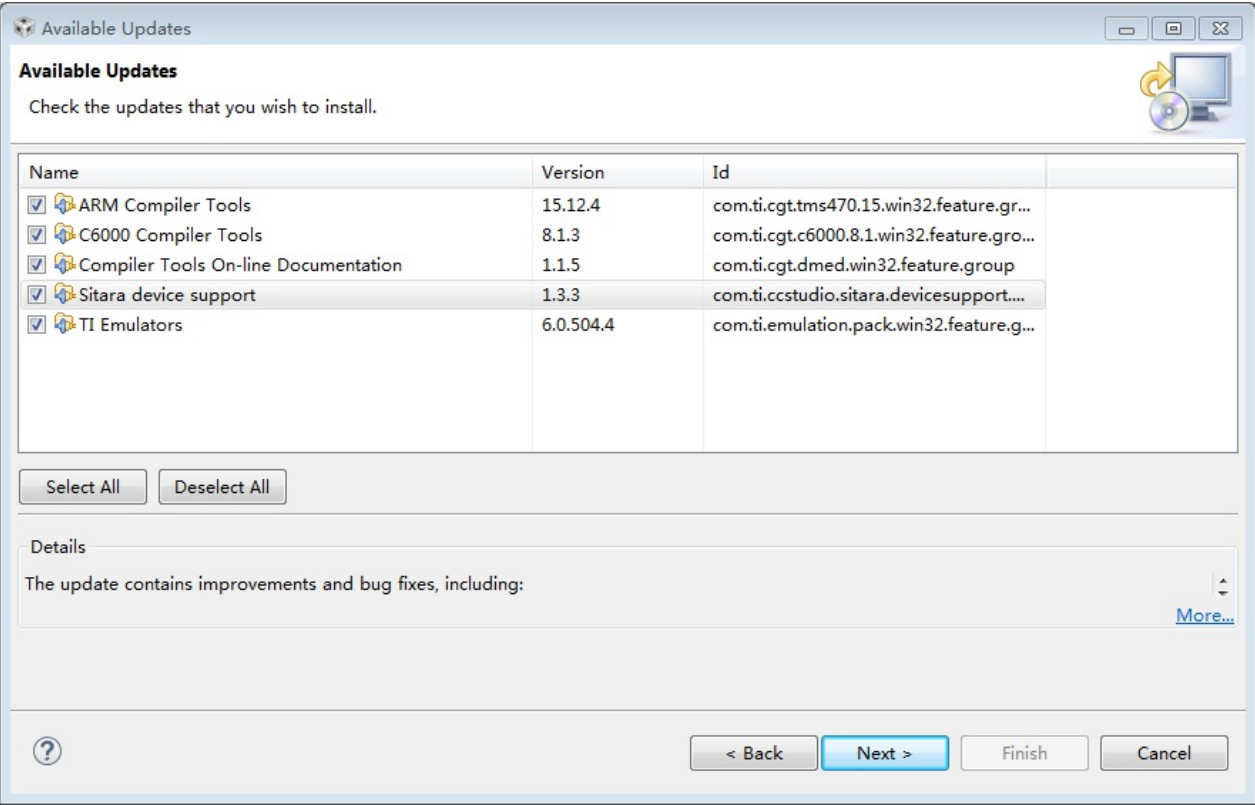


图2-4 升级Sitara设备支持列表

MYD-C437X-PRU并不在TI官方的Sitara支持列表中，因此需要用户手动添加，过程也很简单。将出厂附带资料中05_RTOS_Source/MYIRTECH/myir_sitara_device_support_1.3.3.zip解压出来，目录结构如下

```
├─ common
│   └─ targetdb
│       └─ boards
│           └─ mydidk437x.xml
└─ emulation
    └─ boards
        └─ mydidk437x
            └─ gel
                ├── AM437x_EVMs.gel
                ├── AM437x_MMRS.gel
                ├── AM437x_Status.gel
                ├── AM43xx_EMIFconfig_HW1v1.gel
                ├── AM43xx_ICSSconfig.gel
                ├── AM43xx_PLLconfig.gel
                ├── idk_am437x.gel
                └─ mydidk437x.gel
```

拷贝mydidk437x.xml文件到 C:\ti\ccsv6\ccs_base\common\targetdb\boards目录下；拷贝mydidk437x目录到 C:\ti\ccsv6\ccs_base\emulation\boards目录下。

拷贝完之后重新启动CCS,即可创建MYDIDK437X目标板的配置文件。
主菜单下选择View->Target Configurations。在Target Configuration子窗口下点右键弹出对话框选择New Target Configuration。创建一个针对MYD-C437X-PRU的目标板配置文件，这里命名为mydidkc437x.xml。选择save保存之后，再选择Test Connection进行连接测试。如果测试连接正常，可以通过右键选择该配置文件，在弹出菜单中选择Set as Default，将其设置为默认的目标板配置文件。准备就绪之后即可进行后续的项目调试了，如下图所示

Test Connection进行连接测试。如果测试连接正常，可以通过右键选择该配置文件，在弹出菜单中选择Set as Default，将其设置为默认的目标板配置文件。准备就绪之后即可进行后续的项目调试了，如下图所示：

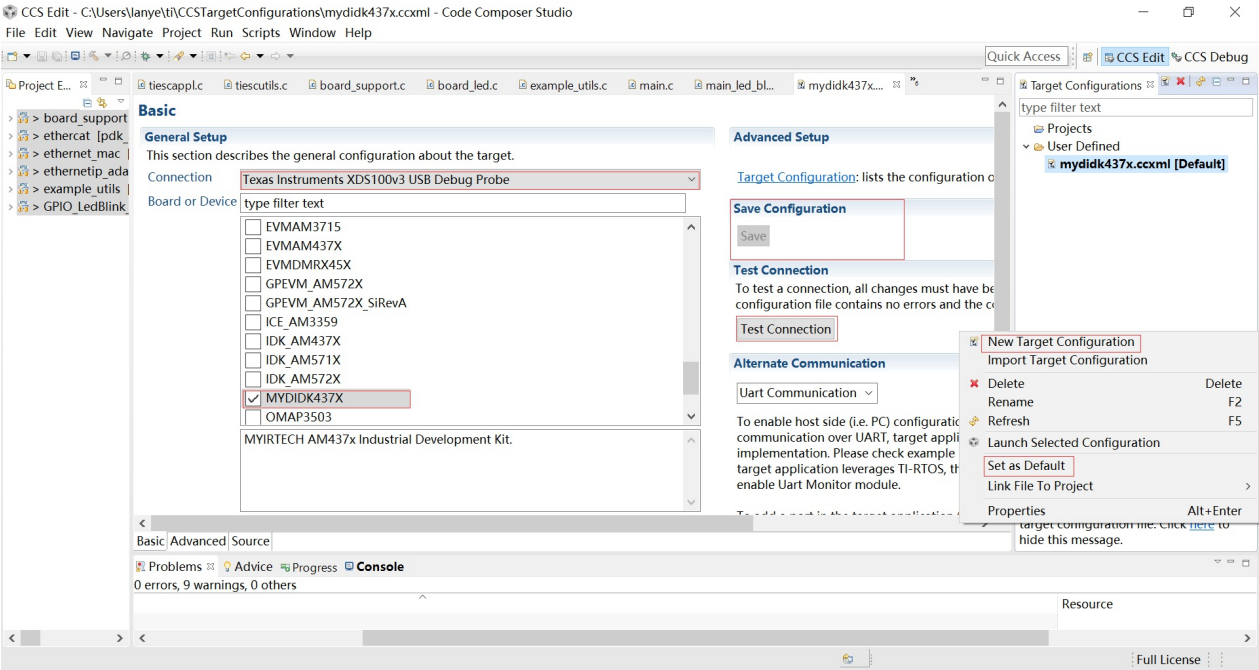


图2-5 创建MYDIDK437X目标板配置文件

3. 构建系统

SYSBIOS实时操作系统(RTOS)是TI开发的一款开源实时操作系统，本节内容主要介绍TI的开源SYS/BIOS实时操作系统(RTOS)在米尔MYD-C437X-PRU上的移植过程。

而关于TI的开源SYSBIOS实时操作系统本身的细节在本手册中不做详细介绍，用户可以查阅TI官方WIKI页面：

<http://processors.wiki.ti.com/index.php/Category:SYSBIOS>。

构建系统前准备：

本节介绍的SYSBIOS实时系统构建过程基于TI专门用于实时操作系统开发的Processor SDK套件展开，所选版本为processor_sdk_rtos_am437x_3_01_00_06。

TI提供了在windows系统下的安装包ti-processor-sdk-rtos-am437x-evm-03.01.00.06-Windows-x86-Install.exe。该安装包位于出厂附带资料05-RTOS_Source\TI目录下，也可以从TI官网下载。

安装时选择默认安装路径为c:\ti，剩下的过程主要是将资料拷贝到c:\ti目录下，安装完成之后目录结构如下所示，其中也包含了之前安装的CCS：

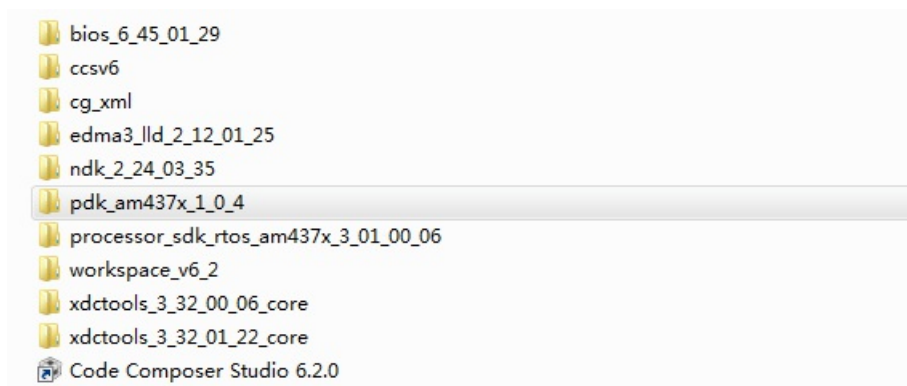


图3-1 TI RTOS SDK目录结构

作为一款面向工业应用的开发板，本手册还会介绍EtherCAT工业协议的RTOS应用。这些应用例程来自于TI官方的sysbios_ind_sdk_02.01.02.02开发包sysbios_ind_sdk_win32setup_02_01_02_02.exe。该安装包位于出厂附带资料05-RTOS_Source\TI目录下。

sysbios_ind_sdk和processor_sdk_rtos存在一些重复和冲突的内容，用户不用安装sysbios_ind_sdk。

为了方便这些应用的编译和生成，我们将这些应用对应的代码从sysbios_ind_sdk提取出来，针对MYD-C437X-PRU做适配之后和PDK集成在一起，代码置于C:\ti\pdk_am437x_1_0_4\packages\examples目录下。用户需要手动将出厂附带资料中05-RTOS_Source\MYIRTECH\pdk_am437x_1_0_4和C:\ti\pdk_am437x_1_0_4进行拷贝合并之后，进行后续的调试和开发。

运行RTOS应用：

构建完成之后的RTOS应用可执行文件需要命名为app，它由位于startware目录下的bootloader进行加载和引导之后运行。下面介绍从TF卡启动并运行RTOS应用的方法。

- 首先使用出厂附带资料中03-Tools目录下的HP USB Disk Storage Format Tool 2.0.6工具来格式化TF卡，如下图所示：

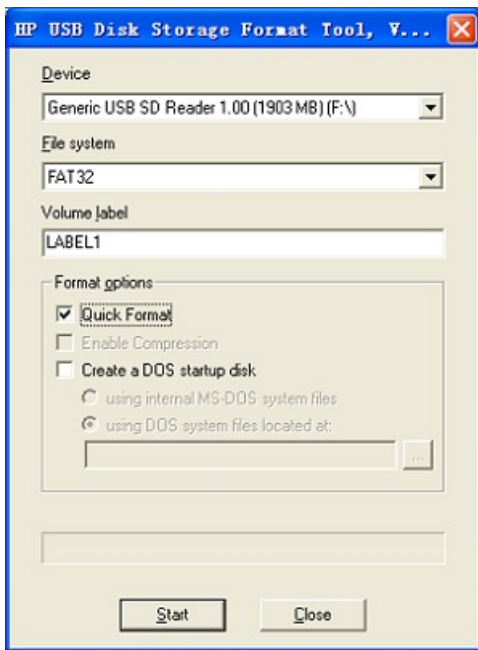


图3-2 格式化TF卡

- 将出厂附带资料中05-RTOS_Source\MYIRTECH\processor_sdk_rtos_am437x_3_01_00_06\prebuilt-sdcards\mydidk437x\sd_card_files 目录下的MLO和app拷贝到TF卡。

其中MLO是通过startware目录下的bootloader编译生成的二进制文件，app为预编译的EtherCAT Slave RTOS应用。

- 将MYD-C437X-PRU设置为TF卡启动模式，连接调试串口，插入准备好的TF卡，重新上电开机。

EtherCAT Slave RTOS应用运行之后，串口调试终端输出结果如下，详细的使用方法，后续有专门的章节介绍。

```
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 设置环境变量

TI提供了Windows和Linux两种操作系统下的SYSBIOS开发工具和使用说明，本手册中SYSBIOS实时操作系统的开发主要是在Windows操作系统下进行，Linux系统下不再详细介绍。

在processor_sdk_rtos_am437x_3_01_00_06安装完成之后，在对应的安装目录中提供了环境变量设置的脚本。我们只需要进入相应目录，执行安装脚本即可。

```
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程序位于xdctools中，脚本执行时，如果出现下面的错误提示，则需要将路径

C:\ti\xdctools_3_32_01_22_core\packages\xdc\services\io\release加入到Windows系统环境变量PATH之中，然后再重新打开命令窗口执行。

```
'path2dos' 不是内部或外部命令，也不是可运行的程序
或批处理文件。
```

在setupenv.bat同级目录还有一个Makefile文件，环境变量设置完成之后，就可以使用gmake进行代码的编译了，首先我们可以使用gmake help查看一下Makefile支持目标组件。

```
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 生成Bootloader

TI SYSBIOS系统所需的引导程序位于startware之下，它实现开发板上电重启之后的引导，对硬件进行初始化，将app加载到DDR中并将控制权转交给app去执行。

代码路径：

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

执行过程：

- 1. 上电复位
- 2. ROM Bootloader(RBL)
 - 平台配置和初始化。
 - DPLL和MPU时钟设置，I2C，MMCSd，USB，SPI，QSPI，Ethernet等初始化。
 - 检查SYSBOOT，确定启动模式
 - 如果没有找到bootloader，RBL检查下一种模式。检查的顺序取决于RBL的执行流程以及SYSBOOT管脚配置。
 - RBL检查bootloader的格式，从附加在bootloader上的Header中获取文件的大小和加载地址。
 - 将二进制码加载到Header指定的内存地址。
 - 控制权转交给第二阶段引导程序(SBL)
- 3. 第二阶段引导程序(SBL) 执行
 - 配置PLL并初始化DDR。
 - 配置外围PRCM和PinMux。
 - 将应用程序(app文件)拷贝到DDR。
 - 控制权转交给应用程序。
- 4. 应用程序开始执行。

编译生成Bootloader

按照上一节的说明配置好环境变量之后，就可以编译生成Bootloader了，编译bootloader的命令如下：

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

编译完成之后，在C:\ti\pdk_am437x_1_0_4\packages\ti\starterware\binary\bootloader\bin\am43xx-evm\gcc目录下得到各种启动模式下的二进制文件，如下所示：

```
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
```

每个二进制文件对应不同启动模式下的bootloader，下面以表格的形式，对生成的这些二进制文件加以说明

二进制文件	模式	详细信息
.bin	QSPI bootloader, UART bootloader和应用程序	QSPI bootloader二进制文件为.bin格式，但应用程序为_ti.bin格式，UART bootloader和应用程序都为.bin格式，通过XMODEM协议传输
_ti.bin	MMCSd bootloader和应用程序, NAND bootloader和应用程序, MCSPI bootloader和应用程序, QSPI应用程序.	.bin二进制格式增加4字节的镜像文件大小信息和4字节的内存加载地址信息之后转变为_ti.bin

生成TF卡启动模式引导程序

bootloader_boot_mmcsd_a9host_debug_ti.bin或bootloader_boot_mmcsd_a9host_release_ti.bin即为TF卡启动模式下的引导程序，不过使用的时候需要将其重命名为MLO，放置到之前准备好的TF卡根目录下。

Bootloader调试

RTOS系统下的Bootloader位于startware中，也可以通过CCS进行调试。在主菜单下选择File->Import->CCS Projects。然后选择Select search-directory为C:\ti\pdk_am437x_1_0_4\packages\ti\starterware。同时导入bootloader和bootloader的依赖库board,dal,soc,utils,ff9b_lib, mmcsd_lib, xmodem_lib, qspi_lib, device, nand_lib。注意勾选Automatically import reference projects found in same search-directory。如下图3-2-1所示。

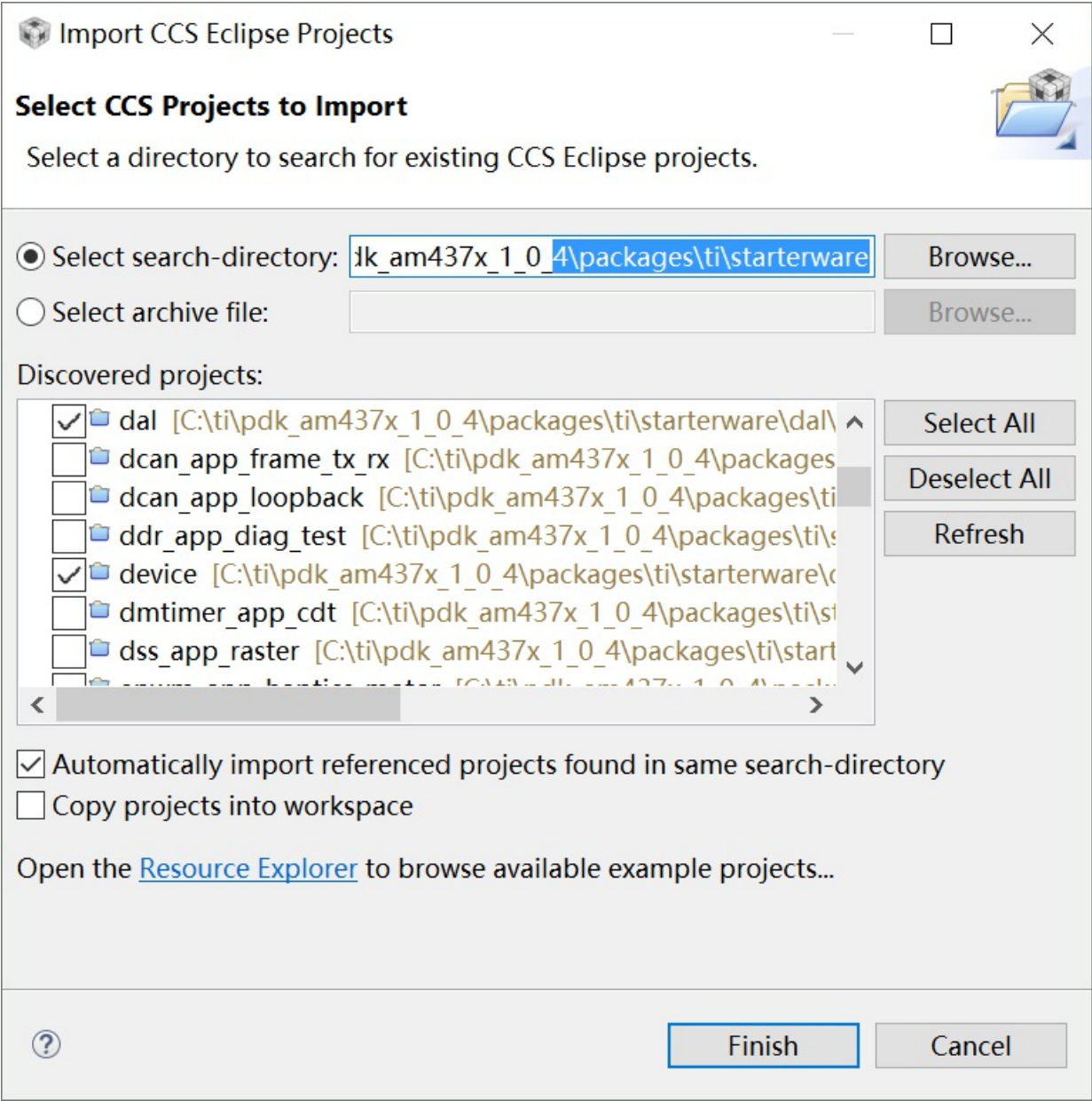


图3-2-1 导入bootloader

导入成功之后，右键选择**bootloader**，然后在弹出菜单下选择**Build Configurations->Set Active->AM43xx_mmcscd_boot_debug**。接下来对所导入的库以及**bootloader**分别进行编译。编译成功之后可以对其进行调试。调试之前如要进行一些调试设置。具体流程如下：

1. 右键选择已编译完成的工程，在弹出菜单中选择**Debug As->Debug Configurations...**，此时会弹出一个对话框。
2. 右键选择**Code Composer Studio-Device Debugging**，在弹出菜单下选择**New**
3. 配置命名为**bootloader**，并对其进行下图3-2-3和图3-2-4所示的参数配置。
4. 选择**Apply**进行配置保存。
5. 选择**Debug**进入调试视图界面，如图3-2-4所示，接下来就可以进行单步调试了。

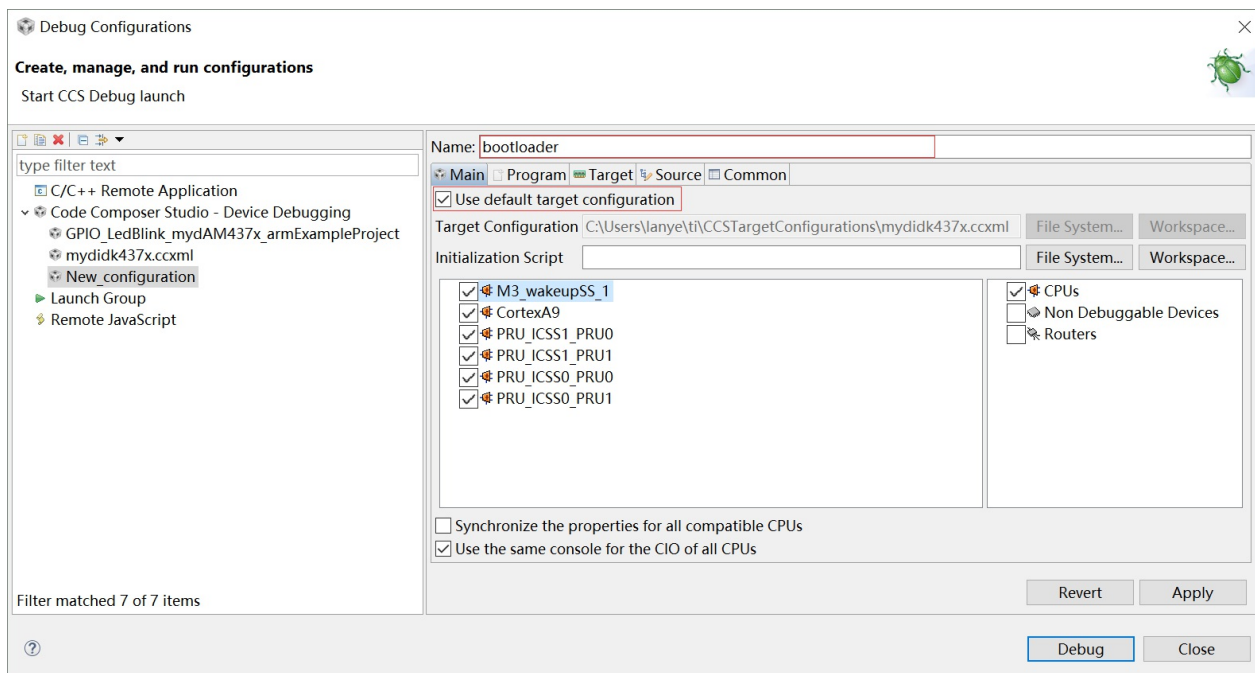


图3-2-2 bootloader项目调试设置一

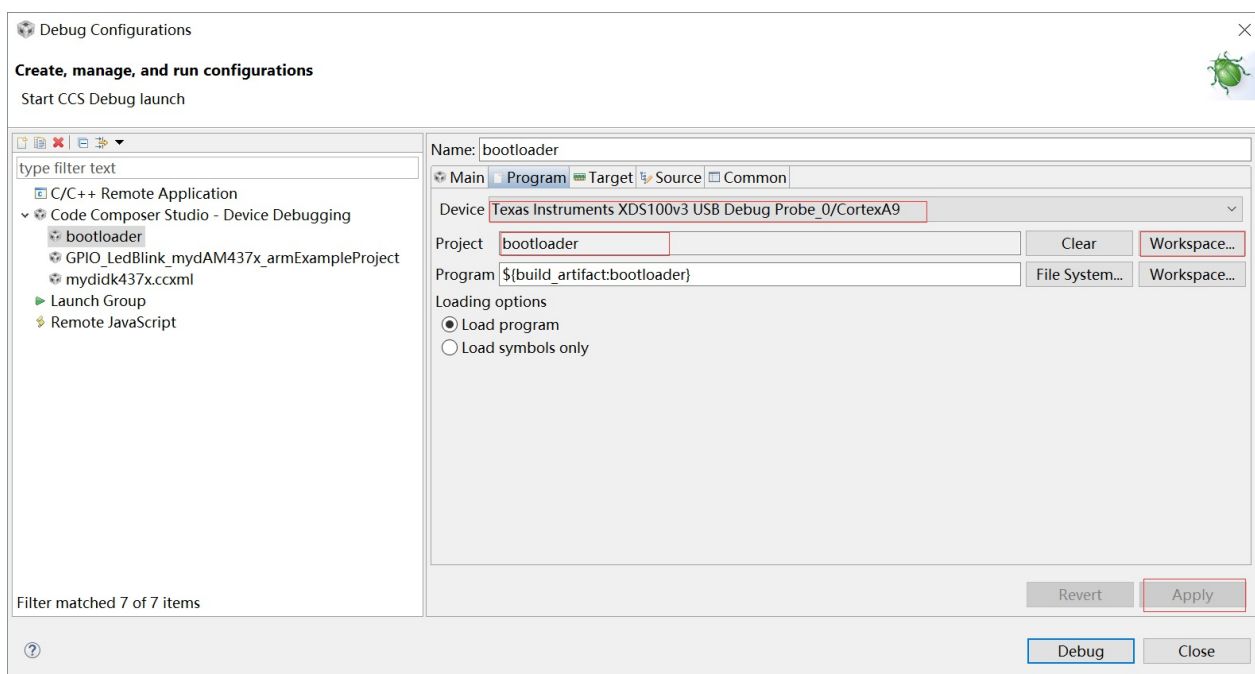


图3-2-3 bootloader项目调试设置二

3.3 生成RTOS应用

本节介绍RTOS应用的创建，编译以及调试和运行的过程。

PDK介绍：

processor_sdk_rtos_am437x_3_01_00_06中另外一个重要的目录就是C:\ti\pdk_am437x_1_0_4, 其中包含了一些CCS插件，以及RTOS应用开发的支持包，样例程序等。

我们从sysbios_ind_sdk_02.01.02.02开发包中提取的实时工业通信例程也置于该目录之

下。C:\ti\pdk_am437x_1_0_4\packages目录下还有几个重要的脚本文件，下面分别加以说明。

```
pdksetupenv      -- pdk环境变量设置脚本
pdkProjectCreate -- pdk工程创建脚本，创建CCS工程的RTOS应用
pdkAppImageCreate -- pdk应用程序创建脚本，在CCS工程的RTOS应用编译生成的过程中进行格式转换，
                    并最终生成app文件
```

创建PDK项目：

我们可以使用pdkProjectCreate脚本针对不同的模块创建CCS工程。在Windows操作系统下使用方法如下：

```
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 / nimu / 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
```

目前，PDK中提供了针对MYD-C437X-PRU的GPIO控制LED例程，可以使用下面的命令创建该例程的CCS工程文件。

```

C:\ti\SDK_AM437x\packages>pdkProjectCreate.bat AM437x mydAM437x
=====
Configuration:
SOC           : AM437x
BOARD         : mydAM437x
ENDIAN        : little
MODULE        : all
PROCESSOR     : arm
PDK_SHORT_NAME : C:\ti\SDK_AM437x\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\SDK_AM437x\packages\MyExampleProjects
Detected Test Project: GPIO_LedBlink_mydAM437x_armExampleProject
-----
Creating project 'GPIO_LedBlink_mydAM437x_armExampleProject'...
Done!
Copying macro.ini
已复制          1 个文件。
No projects detected
Project generation complete
*****

```

至此，针对MYD-C437X-PRU的GPIO控制LED例程创建成功，对应的CCS工程位于
C:\ti\SDK_AM437x\packages\MyExampleProjects\GPIO_LedBlink_mydAM437x_armExampleProject目录。

编译和运行PDK项目：

- 导入项目，具体步骤如下：

1. 打开CCS
2. 选择File->Import，出现项目导入的对话框，如下图3-3-1所示：
3. 在导入项目对话框中选择需要导入的工程所在文件夹C:\ti\SDK_AM437x\packages\MyExampleProjects\GPIO_LedBlink_mydAM437x_armExampleProject
4. 选择Finish完成项目导入。

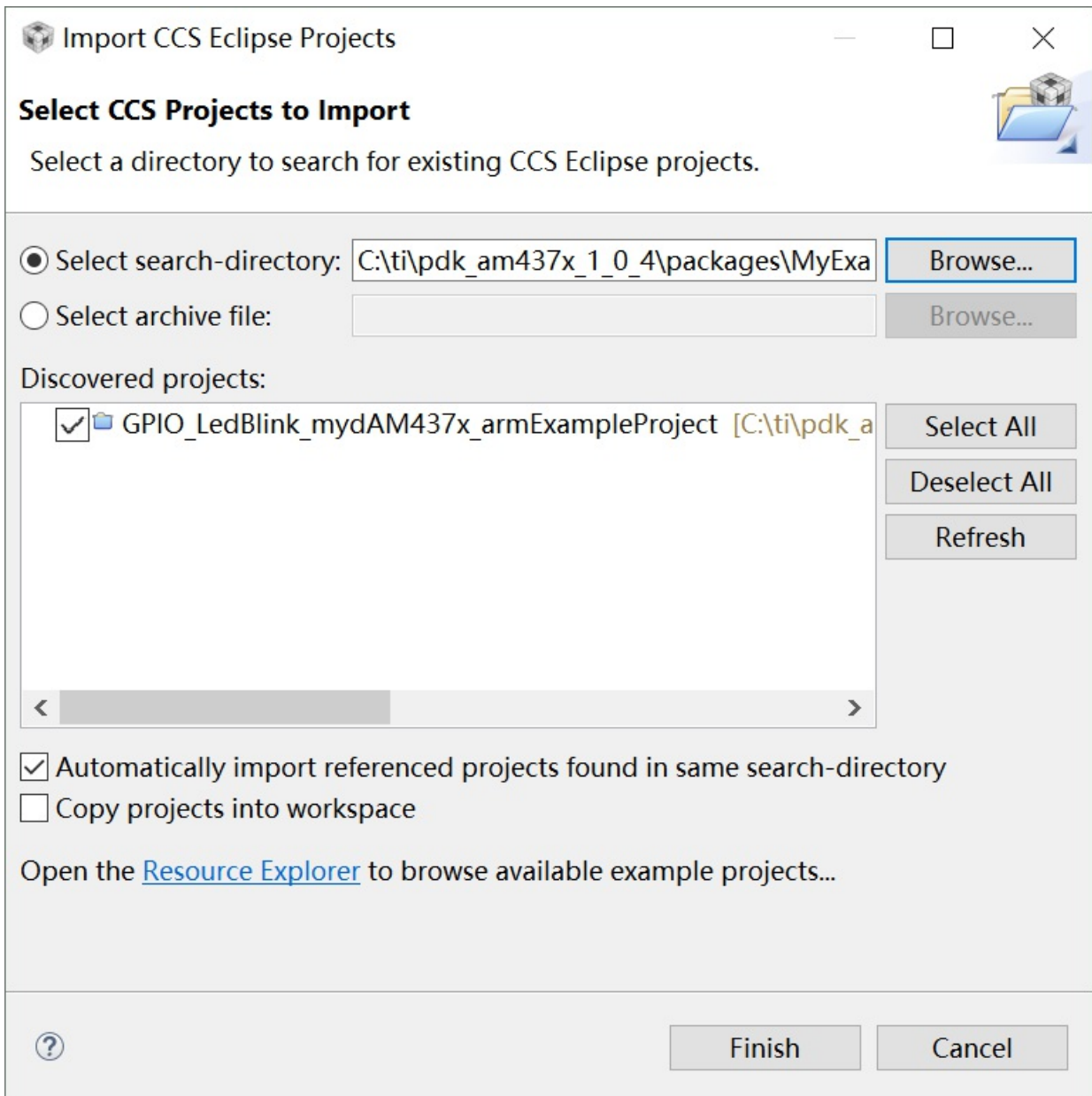


图3-3-1 gpio_ledblik工程导入

- 编译生成项目

右键选择已导入的工程，在弹出菜单中选择Build Project。编译过程中如果出现下面的错误, 可以右键选择该工程，在弹出菜单中选择Properties,在Properties设置对话框中选择

Build->GNU Compiler->Runtime下修改-fmpu参数为neon, 如下图3-3-2所示:

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

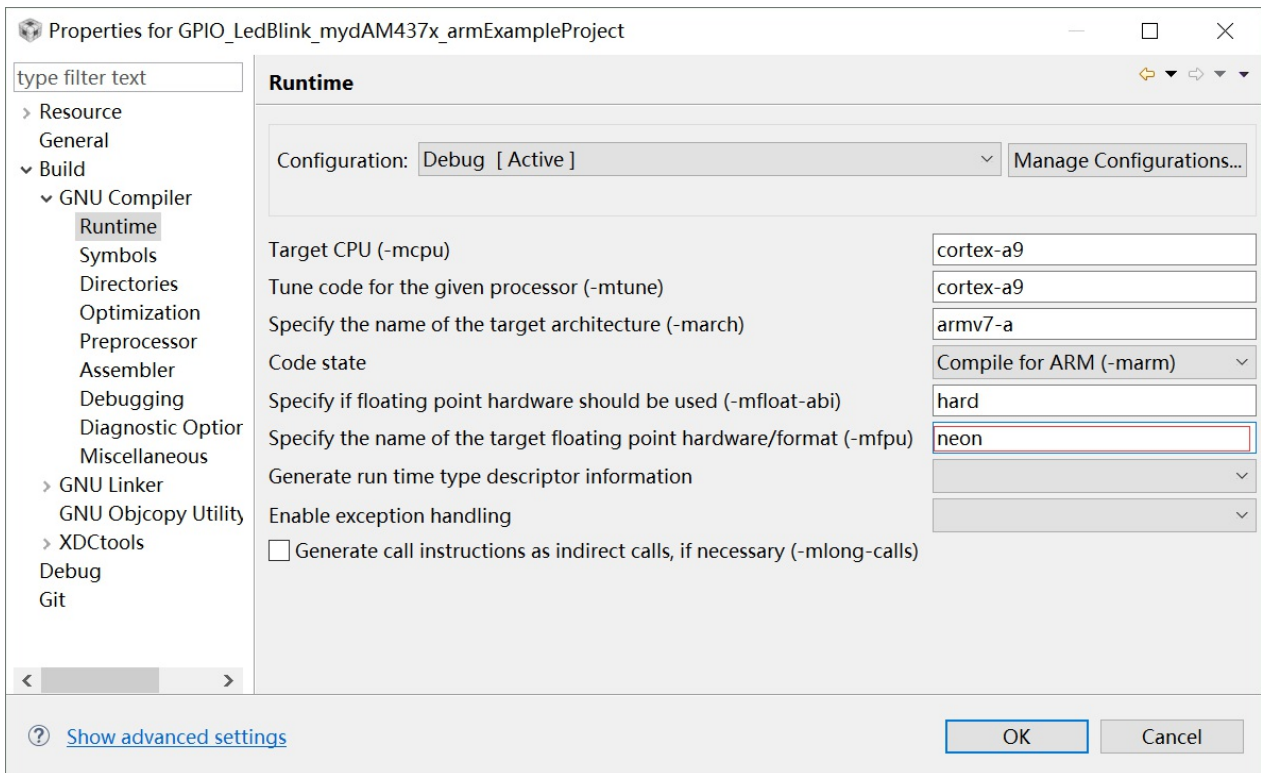


图3-3-2 gpio_ledblik工程属性设置

- 调试项目

项目编译成功之后，可以对其进行调试。调试之前如要进行一些调试设置。具体流程如下：

1. 右键选择已编译完成的工程，在弹出菜单中选择Debug As->Debug Configurations...,此时会弹出一个对话框。
2. 右键选择Code Composer Studio-Device Debugging, 在弹出菜单下选择New
3. 配置命名为GPIO_LedBlink_mydAM437x_armExampleProject, 并对其进行下图3-3-3和图3-3-4所示的参数配置。
4. 选择Apply进行配置保存。
5. 选择Debug进入调试视图界面, 如图3-3-5所示, 接下来就可以进行单步调试了。

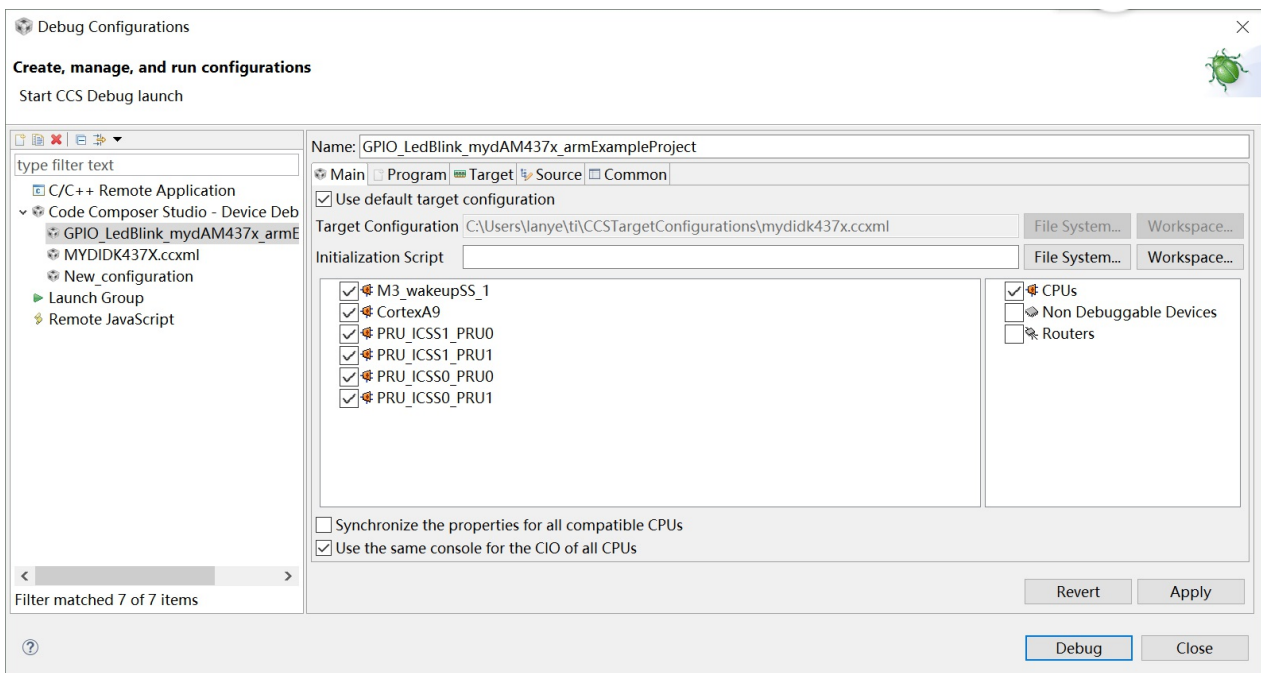


图3-3-3 gpio_ledblik项目调试设置一

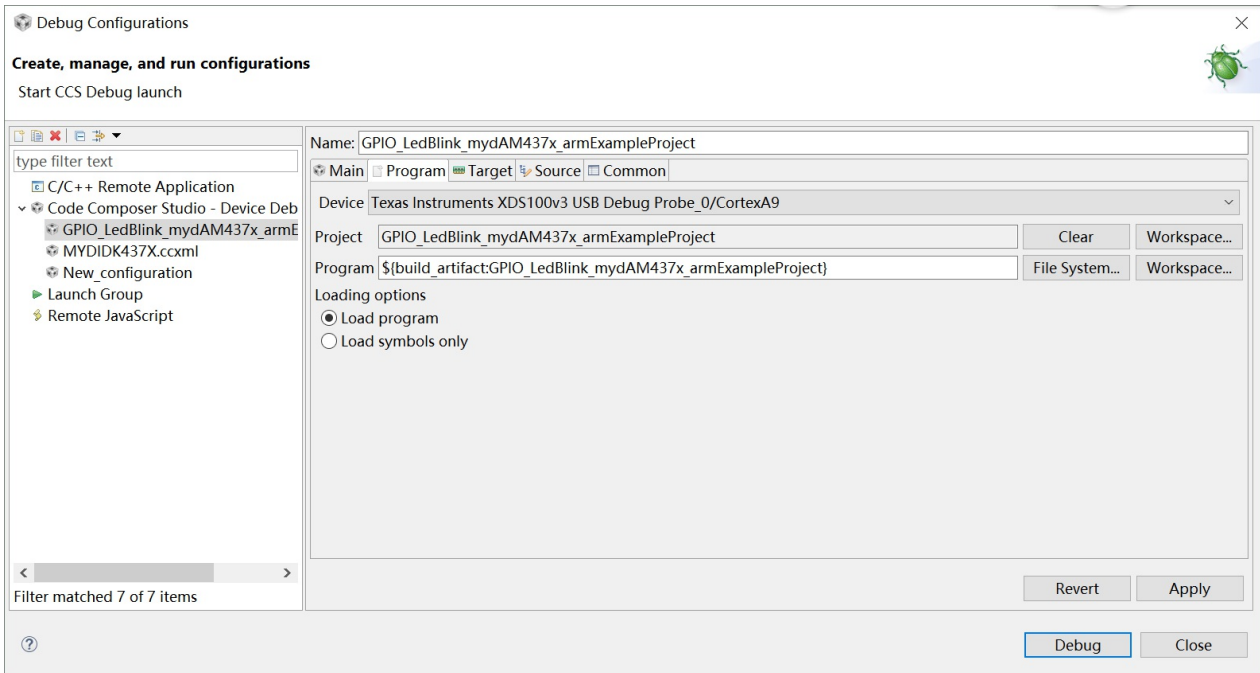


图3-3-4 gpio_ledblink项目调试设置二

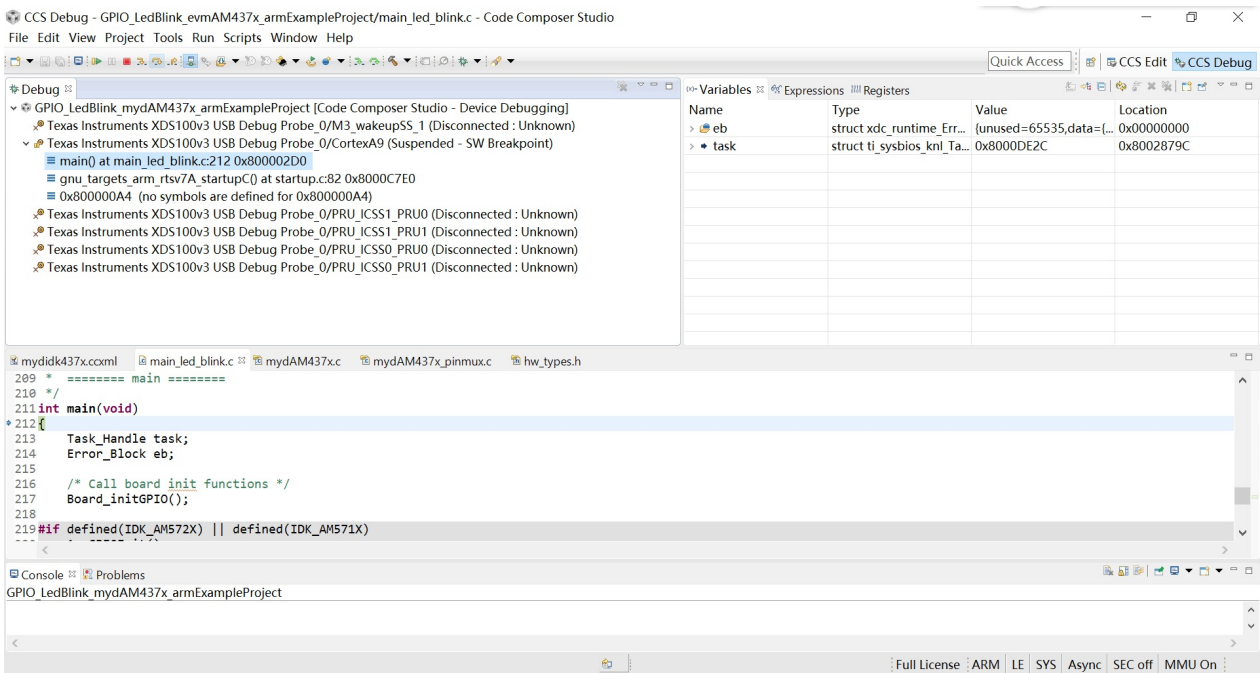


图3-3-5 gpio_ledblink工程调试视图

- 运行项目

想要运行项目，在调试阶段可以直接选择F8全速执行。

编译生成的程序镜像位于

C:\ti\pdk_am437x_1_0_4\packages\MyExampleProjects\GPIO_LedBlink_mydAM437x_armExampleProject\Debug 目录

```

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

```

其中app文件是可以直接用于烧写到QSPI Flash和TF卡中运行的，这个文件是通过GPIO_LedBlink_mydAM437x_armExampleProject.bin文件然后添加文件头信息生成的，生成过程如下：

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

tiimage.exe位于C:\ti\pdk_am437x_1_0_4\packages\ti\starterware\tools\ti_image目录，例程在编译的时候已经自动完成app生成步骤了。

拷贝app至TF卡，设置开发板为TF卡启动模式，重新开机。通过调试串口的输出信息(如下所示)和开发板上LED的变化观察RTOS应用程序的执行情况。

```
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
```

反复按下释放底板上的S3按键，核心板上蓝色LED灯随之点亮和熄灭。

3.4 启动方式

MYD-C437X-PRU开发板出厂镜像文件位于光盘02-Images\rtos-images目录下，QSPI目录下为QSPI烧写和启动的ethercat_slave应用镜像，TF目录下为TF卡启动的ethercat_slave应用镜像。

TF卡启动

TF卡启动需要两个文件，一个命名为MLO的boot文件，一个命名app的应用程序文件。

boot文件和应用程序文件都是以*_ti.bin结尾的文件。

boot文件位于C:\ti\pdk_am437x_1_0_4\packages\ti\starterware\binary\bootloader\bin\am43xx-evm\gcc目录

```
bootloader_boot_mmc_ssd_a9host_debug_ti.bin
bootloader_boot_mmc_ssd_a9host_release_ti.bin
```

debug版本和release版本功能是一样的，只是debug版本多了用于CCS仿真调试的信息。拷贝其中一个文件至SD卡然后重新命名为MLO。

应用程序文件以QSPI烧写程序为例说明，文件位于

C:\ti\pdk_am437x_1_0_4\packages\ti\starterware\binary\qspi_app_flash_writer\bin\am43xx-evm\gcc目录下

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

debug版本和release版本功能是一样的，只是debug版本多了用于CCS仿真调试的信息。拷贝其中一个文件至SD卡然后重新命名为app。

QSPI启动

QSPI启动分为QSPI烧写和QSPI程序运行。

QSPI烧写也是借助TF卡启动来完成的，TF卡中包含以下文件。

```
app
boot
config
ethercat
MLO
```

MLO为TF启动的boot文件，app为TF卡启动的QSPI烧写应用程序文件。

MLO文件和app文件的具体来源在上述TF卡启动中已说明。

config为QSPI烧写的配置文件，里面包含QSPI启动的boot文件和QSPI应用程序的烧写信息，配置文件内容如下：

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

boot文件是位于C:\ti\pdk_am437x_1_0_4\packages\ti\starterware\binary\bootloader\bin\am43xx-evm\gcc目录下：

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

拷贝其中一个文件至SD卡然后重新命名为boot。

ethercat应用文件位于C:\ti\pdk_am437x_1_0_4\packages\ti\examples\ethercat_slave\am437x_release目录下：

```
ethercat_ti.bin
```

拷贝文件至SD卡然后重新命名为ethercat。

上述文件准备完成后，将TF卡插入开发板，设置开发为TF卡启动，给开发板上电，从串口中看到以下信息说明烧写完成。

```
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
```

烧写完成后，设置开发板的启动模式为QSPI模式，拔掉SD卡，断电重新上电，从串口中看到以下信息说明QSPI启动成功。

```
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. 工业协议应用

MYD-C437X-PRU开发板为用户提供一个实时工业通信协议的开发平台。TI官方发布的sysbios_ind_sdk_02.01.02.02开发包提供了多个实时工业通讯例程。本节选取比较典型的EtherCAT Slave工业通信协议应用移植到MYD-C437X-PRU平台，为用户的实时工业通信协议开发提供参考。

测试硬件环境：

- 带以太网口的PC主机一台，2GB以上内存。
- MYD-C437X-PRU 开发板一块。
- CAT5交叉网线一根，连接PC主机和开发板。
- USB转TTL调试串口一根，连接开发板的J25, PC端波特率设置115200-8-n-1。

测试软件环境：

- Windows 7/10 PC主机
- Beckhoff TWinCAT V3.1.4020.28
- CCS 6.2.0
- ethercat slave 应用程序

运行Beckhoff TwinCAT V3.1.4020.28的PC主机作为EtherCAT的主站，MYD-C437X-PRU开发板运行ethercat_slave应用作为从站。PC端的以太网卡需要支持ethercat协议。支持EtherCAT协议的以太网卡请查看beckhoff官方网站的支持列表，如下：

https://infosys.beckhoff.com/english.php?content=../content/1033/tcsystemmanager/reference/ethercat/html/ethercat_supnetworkcontroller.htm&id=

生成EtherCAT Slave应用：

EtherCAT应用由CCS编译生成。打开CSS，选择File->Import，出现项目导入的对话框。然后再对话框中选择C:\ti\pdk_am437x_1_0_4\packages\ti\examples\ethercat_slave目录导入到CSS中。接下来根据上一节中生成RTOS应用的方法进行编译，调试和运行。

注意：编译RTOS应用依赖pdk中的一些库文件，所以需要先使用gmake对pdk进行编译。

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

或者

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

编译之后，C:\ti\pdk_am437x_1_0_4\packages\ti\examples\ethercat_slave\am437x_debug目录下生成ethercat_ti.bin文件，重新命名为app之后，可以跟前一节中一样和MLO文件一起拷贝到预先准备好的TF卡中，从TF卡启动运行。

测试过程：

本应用程序主要演示EtherCAT的接口通信，LED控制。具体过程如下。

- 安装TwinCAT V3.1.4020.28。

Beckhoff公司提供了TwinCAT的免费评估版本，用户可以从MYD-C437X-PRU出厂附带资料03-Tools中获取TC31-Full-Setup.3.1.4020.28压缩包解压后安装，也可以直接从 Beckhoff公司的网站[下载](#)。安装时注意选择[eXtended Automation Engineering (XAE)]模式。

- 使用TwinCAT主站和开发板通信，对LED进行控制，具体步骤如下：

1. 拷贝 `C:\ti\pdk_am437x_1_0_4\packages\ti\examples\ethercat_slave\TiEtherCATLib.xml` 文件到TwinCAT安装目录:`\TwinCAT\3.1\Config\lo\EtherCAT\`之下。
拷贝 `C:\ti\pdk_am437x_1_0_4\packages\ti\protocols\ethercat_slave\ecat_app\les\TI_ESC_CTT.xml`文件到TwinCAT安装目录:`\TwinCAT\3.1\Config\lo\EtherCAT\`之下。
2. 打开TwinCAT XAE(VS2013)
3. 创建一个新的TwinCAT XAE工程: File->New Project->TwinCAT Project
4. 安装TwinCAT RT Ethernet intermediate驱动: TwinCAT->Show Real Time Ethernet Compatible Devices;选中所使用的网卡, 然后选择Install进行安装
5. 使用CAT5网线连接PC上安装了RT Ethernet intermediate驱动的网卡和开发板的PRUETH0网卡(J26)。
6. 扫描设备: 在Solutin Explorer下, 右键选择刚创建的工程->I/O->右键选择Devices->Scan, 如果Scan Boxes是灰色则选择TwinCAT->Restart TwinCAT(Config Mode)
7. TI Boxn(ti-esc)将会被检测到, 然后选择Device1(EtherCAT),在选择TwinCAT菜单->Reload Devices
8. 在弹出对话框中选择yes, 接下来弹出对话框Activate Free Run中同样选择yes. TI ESC将会进入OP模式, 用户可以通过选择Boxn(TI-ESC)->RxPDO->32Bit Output, 写入相应的数字对LED进行控制, 如下图4-1所示

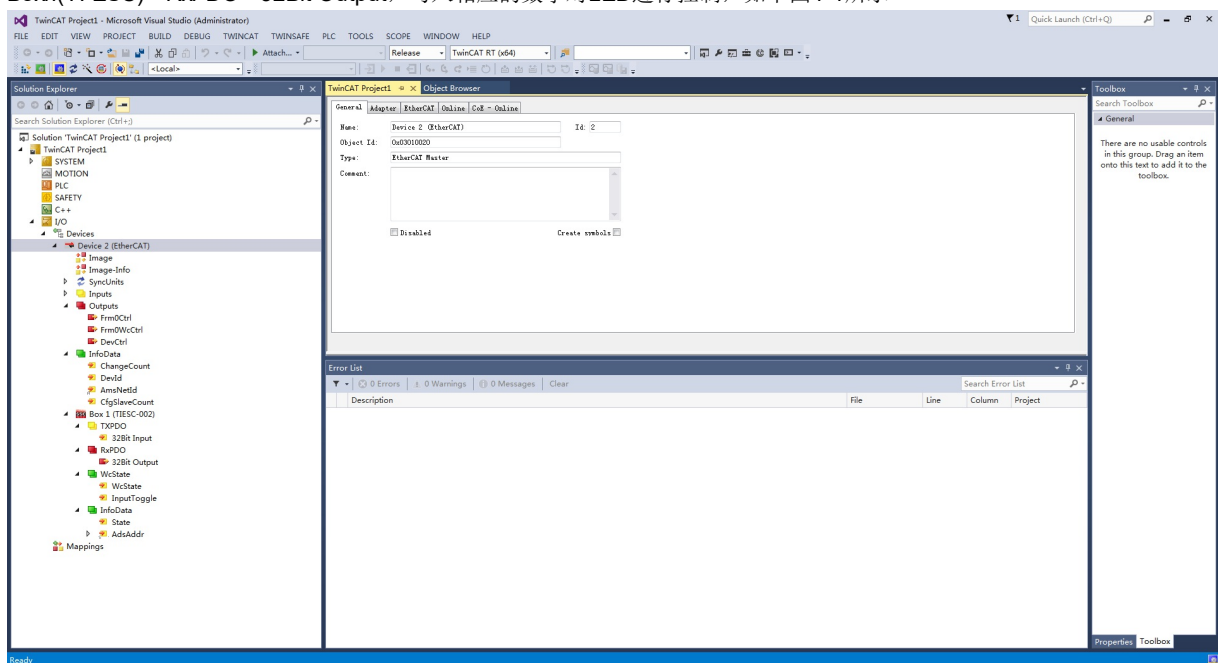


图4-1 TwinCAT主站检测到从站设备

9. 选择Boxn(TI-ESC)->RxPDO->32Bit, 在右边窗口选择Online->Write 依次写入一个0~15之间的数字, 观察开发板上LED的变化。如下图4-2所示, 输入数字15, 开发板上四个蓝色LED灯全部点亮。

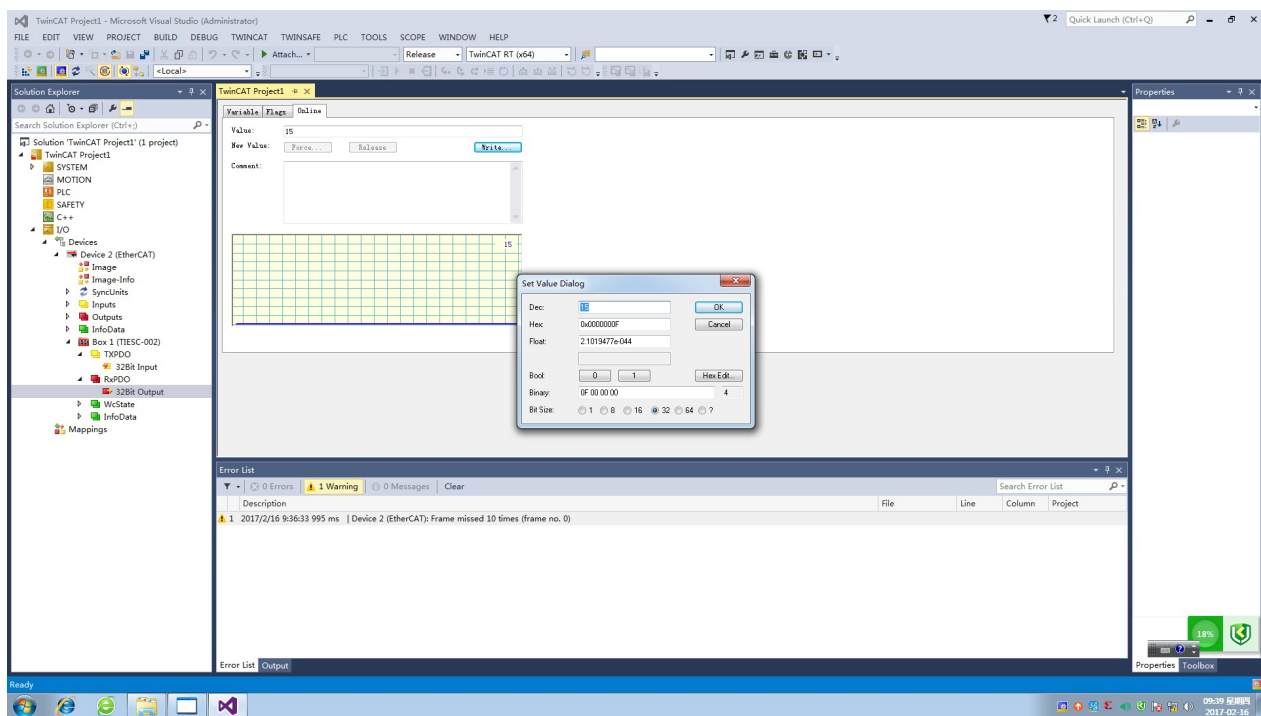


图4-2 TwinCAT主站操作控制开发板上LED

附录一 联系方式

销售联系方式

- 网址: www.myir-tech.com
- 邮箱: sales.cn@myirtech.com

深圳总部

- 负责区域: 广东 / 四川 / 重庆 / 湖南 / 广西 / 云南 / 贵州 / 海南 / 香港 / 澳门
- 电话: 0755-25622735 0755-22929657
- 传真: 0755-25532724
- 邮编: 518020
- 地址: 深圳市龙岗区坂田街道发达路云里智能园2栋6楼04室

上海办事处

- 负责区域: 上海 / 湖北 / 江苏 / 浙江 / 安徽 / 福建 / 江西
- 电话: 021-60317628 15901764611
- 传真: 021-60317630
- 邮编: 200062
- 地址: 上海市普陀区中江路106号北岸长风I座1402

北京办事处

- 负责区域: 北京 / 天津 / 陕西 / 辽宁 / 山东 / 河南 / 河北 / 黑龙江 / 吉林 / 山西 / 甘肃 / 内蒙古 / 宁夏
- 电话: 010-84675491 13269791724
- 传真: 010-84675491
- 邮编: 102218
- 地址: 北京市昌平区东小口镇中滩村润枫欣尚2号楼1009

技术支持联系方式

- 电话: 027-59621648
- 邮箱: support.cn@myirtech.com

如果您通过邮件获取帮助时, 请使用以下格式书写邮件标题:

[公司名称/个人--开发板型号]问题概述

这样可以使我们更快速跟进您的问题, 以便相应开发组可以处理您的问题。

附录二 售后服务与技术支持

凡是通过米尔科技直接购买或经米尔科技授权的正规代理商处购买的米尔科技全系列产品，均可享受以下权益：

- 1、6个月免费保修服务周期
- 2、终身免费技术支持服务
- 3、终身维修服务
- 4、免费享有所购买产品配套的软件升级服务
- 5、免费享有所购买产品配套的软件源代码，以及米尔科技开发的部分软件源代码
- 6、可直接从米尔科技购买主要芯片样品，简单、方便、快速；免去从代理商处购买时，漫长的等待周期
- 7、自购买之日起，即成为米尔科技永久客户，享有再次购买米尔科技任何一款软硬件产品的优惠政策
- 8、OEM/ODM服务

如有以下情况之一，则不享有免费保修服务：

- 1、超过免费保修服务周期
- 2、无产品序列号或无产品有效购买单据
- 3、进液、受潮、发霉或腐蚀
- 4、受撞击、挤压、摔落、刮伤等非产品本身质量问题引起的故障和损坏
- 5、擅自改造硬件、错误上电、错误操作造成的故障和损坏
- 6、由不可抗拒自然因素引起的故障和损坏

产品返修：用户在使用过程中由于产品故障、损坏或其他异常现象，在寄回维修之前，请先致电米尔科技客服部，与工程师进行沟通以确认问题，避免故障判断错误造成不必要的运费损失及周期的耽误。

维修周期：收到返修产品后，我们将即日安排工程师进行检测，我们将在最短的时间内维修或更换并寄回。一般的故障维修周期为3个工作日（自我司收到物品之日起，不计运输过程时间），由于特殊故障导致无法短期内维修的产品，我们会与用户另行沟通并确认维修周期。

维修费用：在免费保修期内的产品，由于产品质量问题引起的故障，不收任何维修费用；不属于免费保修范围内的故障或损坏，在检测确认问题后，我们将与客户沟通并确认维修费用，我们仅收取元器件材料费，不收取维修服务费；超过保修期限的产品，根据实际损坏的程度来确定收取的元器件材料费和维修服务费。

运输费用：产品正常保修时，用户寄回的运费由用户承担，维修后寄回给用户的费用由我司承担。非正常保修产品来回运费均由用户承担。