

AMIC110/AM335x EtherCAT Slave Application User guide

----刘靖伟 Steven Liu

EtherCAT 的应用在国内比较流行，使用 AM335x/AMIC110/AM437x/AMIC120 可以支持 EtherCAT slave 的应用，所以这里从零开始，以 AM335x/AMIC110 为例，从零开始，把如何在 AM 系列上玩转这一套方案的流程进行了描述，希望对大家有帮助。本文涉及到的安装和编译，最后 demo 搭建部分，直接参考相关的 user guide 即可。

软件版本选择及安装、CCS 升级：

在跑 AM335x/AMIC110 的 EtherCAT slave demo 的时候，首先要安装两个部分：

一个是 **PRU-ICSS-ETHERCAT-SLAVE** 开发包，当前版本为：01_00_03_01

下载地址：http://software-dl.ti.com/processor-industrial-sw/esd/PRU_ICSS_EtherCAT_Slave/latest/index_FDS.html

另一个就是我们的 **PROCESS SDK RTOS**，建议版本为：03_03_00_04

下载地址：http://software-dl.ti.com/processor-sdk-rtos/esd/AM335X/03_03_00_04/index_FDS.html

注意：后续的软件版本可能会有更新，但一定要注意版本的匹配问题。比如当前的 PRU-ICSS-ETHERCAT-SLAVE 开发包中，安装完成后在 docs 文件夹下 PRU_ICSS_EtherCAT_User_Guide.pdf 文件，中会有对当前支持版本的描述，如下：

Component Version

The following is a list of tools bundled into the CCS installer. If you need to install CCS, see the CCS installation section

Component	Version
CCS	7.0.0
Linaro GCC ARM Compiler	gcc-arm-none-eabi-4_9-2015q3

This release requires Processor SDK RTOS (<http://www.ti.com/tool/ti-rtos-proc>), see Processor SDK Getting Started Guide

Component	Version
Processor SDK RTOS for AM57xx	3.3.0
Processor SDK RTOS for AM437x	3.3.0
Processor SDK RTOS for AM335x	3.3.0
Processor SDK RTOS for K2G	3.3.0

据此，我们选择 PROCESSOR RTOS 的版本为 3.3.0.4。

下载安装完成以上两个开发包后，就可以开始 RTOS 的编译和 ETHERCAT SLAVE 的例程编译工作了。

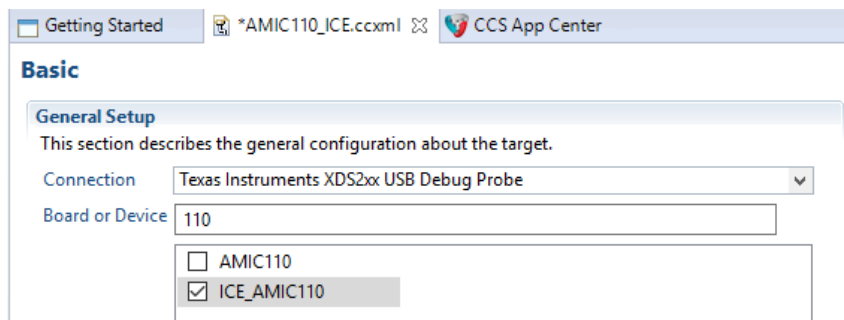
这里，因为大部分用户都是使用 windows 的，所以我目前用的是 windows 10 64-bit 作为环境。

这里简单介绍一下 AMIC110 的开发板：

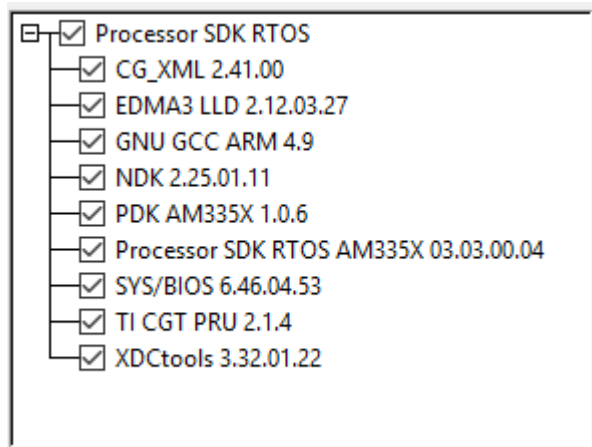
AMIC110 的开发板上，主要是使用 SPI 启动的方式，因此编译的 image，都是要基于 SPI flash 来进行操作。PS: 对于 AMIC110 ICE EVM 版 bootloader 的支持，不建议使用当前的建议使用 3.3.0.4 版本，并不完善。建议使用 processor_sdk_rtos_am335x_4_00_00_04 版本。对于 AM335x ICE 及其他 EVM，3.3.0.4 足以。

CCS 一般建议 7.0 及以上版本。特别需要注意的是：在建立 target configuration 的时候，如果没有 AMIC110 和 ICE_AMIC110 的选项，你需要升级当前的 CCS 组件，方法如下：

选择 CCS 的 Help 选项卡，然后选择 Check for Updates，在弹出的窗体中选择 Sitara device support，然后升级这个部分，升级完成之后，就可以在新配置中看到新发布的产品选项，比如 ICE_AMIC110。



RTOS 和相关软件的安装：安装的时候，RTOS 的相关组件一般建议完整安装，使用默认，不要乱取消即可，列表如下。相关软件建议就安装在默认位置，C 盘的 TI 目录下。若是要改变安装路径，后续脚本编译过程中，要记得改路径名。



软件编译：

安装完成之后，就准备开始编译工作了。

Step 1: 确认路径以及环境变量的设置

若是 CCS 或者 SDK 没有安装在默认路径，需要按照下面的方法来设置你的安装路径，在设置完后，即可使用脚本编译。

注意：你的 CCS 的安装路径和 SDK 的安装路径可能是分开的，所以要根据对应的条目来确认你的安装方法。例如，CCS 安装在自定义路径，但是 SDK 安装在默认路径时，就看链接中的“CCS in Custom Path and SDK RTOS in Default Path”。链接中描述的很详细，注意找对相应的条目。

http://processors.wiki.ti.com/index.php/Processor_SDK_RTOS_Install_In_Custom_Path

如果是安装在默认路径，可以忽略这个步骤。

Step2: 编译 SDK

2.1 设置环境变量 -- 运行 setupenv.bat

Windows 打开 cmd，然后进入到 PDK 的根目录下，例如：

C:\ti\processor_sdk_rtos_am335x_3_03_00_04，然后运行 setupenv.bat 文件。

```
cd c:/ti
```

```
cd processor_sdk_rtos_am335x_3_03_00_04
```

```
setupenv.bat
```

即可，正常情况下的情况如下所示。

如果发生任何问题：

1. 请先检查一下上面 step1 过程中的路径设置是否正确，这其中包括 CCS 的路径，PDK 的路径，PRU-ETHERCAT 的路径。
2. 然后再确认你在 CMD 命令行下的操作，是否是在正确的文件夹下了。
3. 再有就是权限的控制，因为有些公司是有相关的安全要求，所以会有加密和权限等设置，目前接触的 case 中是有类似情况发生的，建议找台相关干净一些的电脑进行相关操作。

cmd Select Processor-SDK RTOS Build Environment

```
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\ao220224>cd c:/ti

c:\ti>cd processor_sdk_rtos_am335x_3_03_00_04

c:\ti\processor_sdk_rtos_am335x_3_03_00_04>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          : am335x
LIMIT_BOARDS        : evmAM335x icev2AM335x bbbAM335x skAM335x
PDK_INSTALL_PATH    : /ti/PDK_AM335x/packages
C6X_GEN_INSTALL_PATH : C:/ti/ti-cgt-c6000_8.1.3
TOOLCHAIN_PATH_A15   : C:/ti/gcc-arm-none-eabi-4_9-2015q3
TOOLCHAIN_PATH_A8     : C:/ti/gcc-arm-none-eabi-4_9-2015q3
TOOLCHAIN_PATH_A9     : C:/ti/gcc-arm-none-eabi-4_9-2015q3
TOOLCHAIN_PATH_M4     : C:/ti/ti-cgt-arm_16.9.1.LTS
TOOLCHAIN_PATH_Arm9   : C:/ti/ti-cgt-arm_16.9.1.LTS
TOOLCHAIN_PATH_EVE    : C:/ti/ccsv7/tools/compiler/arp32_1.0.8
CL_PRU_INSTALL_PATH  : C:/ti/ti-cgt-pru_2.1.4
UTILS_INSTALL_DIR    : C:/ti/xdctools_3_32_01_22_core/bin
FPULIB_PATH          : C:/ti/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_01_22_core
BIOS_INSTALL_PATH    : C:/ti/bios_6_46_04_53
IPC_INSTALL_PATH      : C:/ti/ipc_3_44_01_01
EDMA3LLD_BIOS6_INSTALLDIR : C:/ti/edma3_lld_2_12_03_27
NDK_INSTALL_PATH      : C:/ti/ndk_2_25_01_11
IMGLIB_INSTALL_PATH   : C:/ti/imglib_c66x_3_1_1_0
DSPLIB_INSTALL_PATH   : C:/ti/dsplib_c66x_3_4_0_0
MATHLIB_INSTALL_PATH  : C:/ti/mathlib_c66x_3_1_1_0
UIA_INSTALL_PATH      : C:/ti/uia_2_00_06_52
IPC_PLATFORM: UNKNOWN
IPC_ALT_PLATFORM:
PROC_SDK_INSTALL_PATH : C:/ti/processor_sdk_rtos_am335x_3_03_00_04
*****
Changing to short name to support directory names containing spaces
current directory: C:/ti/processor_sdk_rtos_am335x_3_03_00_04
PROCESSOR SDK BUILD ENVIRONMENT CONFIGURED
*****

c:\ti\processor_sdk_rtos_am335x_3_03_00_04>
```

2.2 使用 gmake 进行编译:

在命令行中，敲入“gmake”，此后，脚本便开始编译工作，下面是正常开始编译的截图。完整编完的时间会比较长(> 40min)。

编译开始:

```
c:\ti\processor_sdk_rtos_am335x_3_03_00_04>gmake
gmake -C /ti/PDK_AM~3/packages all
gmake[1]: Entering directory `c:/ti/PDK_AM~3/packages'
gmake -C ./ti/csl all
gmake[2]: Entering directory `c:/ti/PDK_AM~3/packages/ti/csl'
gmake libcores SOC=am335x
gmake[3]: Entering directory `c:/ti/PDK_AM~3/packages/ti/csl'
gmake lib_allendians CORE=a8host BUILD_PROFILE=a8host-release
gmake[4]: Entering directory `c:/ti/PDK_AM~3/packages/ti/csl'
gmake comp_libs ENDIAN=little
gmake[5]: Entering directory `c:/ti/PDK_AM~3/packages/ti/csl'
Nothing to be done for am335x csl_init
Nothing to be done for am335x csl_initc
gmake -C /ti/PDK_AM~3/packages/ti/csl -fmakefile.mk
gmake[6]: Entering directory `c:/ti/PDK_AM~3/packages/ti/csl'
C:/ti/xdctools_3_32_01_22_core/bin/mkdir -p /ti/PDK_AM~3/packages/ti/binary/ti/csl/obj/am335x/a8/release
C:/ti/xdctools_3_32_01_22_core/bin/mkdir -p /ti/PDK_AM~3/packages/ti/binary/ti/csl/obj/am335x/a8/release/.deps
# Compiling am335x:a8host:release:csl: src/ip/elm/V0/priv/elm.c
C:/ti/gcc-arm-none-eabi-4.9-2015q3/bin/arm-none-eabi-gcc -MD -MF /ti/PDK_AM~3/packages/ti/binary/ti/csl/obj/am335x/a8/release/.deps/elm.P -DMAKEFILE_BUILD -c -mcpu=cortex-a8 -mtune=cortex-a8 -march=armv7-a -marm -mfpu=neon -mfloat=abi-hard -mabi=aapcs -mapcs-frame -D_ARMV7 -Wimplicit -Wall -Wunused -Wunknown-pragmas -ffunction-sections -fdata-sections -Wall -WMD -MP -Werror -O2 -s -DNDEBUG -DBUILD_A8 -DSOC_AM335x -IC:/ti/gcc-arm-none-eabi-4.9-2015q3/arm-none-eabi/include -I. -I. -Isrc/ip/elm/V0 -I. -Isrc/ip/gpio/V1 -I. -Isrc/ip/gpmc/V1 -I. -Isrc/ip/i2c/V2 -I. -Isrc/ip/mcasp/V1 -I. -Isrc/ip/mcspi/V0 -I. -Isrc/ip/mmc/V0 -I. -Isrc/ip/qspi/V1 -I. -Isrc/ip/timer/V1 -I. -Isrc/ip/uart/V1 -I/ti/PDK_AM~3/packages -I/ti/PDK_AM~3/packages/ti/csl -I/ti/PDK_AM~3/packages/ti/csl -I/apps/apps/nonbaw/inc -I/ -I/common -MF /ti/PDK_AM~3/packages/ti/binary/ti/csl/obj/am335x/a8/release/elm.d -MT /ti/PDK_AM~3/packages/ti/binary/ti/csl/obj/am335x/a8/release/elm.oa8fg -o /ti/PDK_AM~3/packages/ti/binary/ti/csl/obj/am335x/a8/release/elm.oa8fg src/ip/elm/V0/priv/elm.c
# Compiling am335x:a8host:release:csl: src/ip/gpio/V1/priv/gpio_v2.c
C:/ti/gcc-arm-none-eabi-4.9-2015q3/bin/arm-none-eabi-gcc -MD -MF /ti/PDK_AM~3/packages/ti/binary/ti/csl/obj/am335x/a8/release/.deps/gpio_v2.P -DMAKEFILE_BUILD -c -mcpu=cortex-a8 -mtune=cortex-a8 -march=armv7-a -marm -mfpu=neon -mfloat=abi-hard -mabi=aapcs -mapcs-frame -D_ARMV7 -Wimplicit -Wall -Wunused -Wunknown-pragmas -ffunction-sections -fdata-sections -Wall -WMD -MP -Werror -O2 -s -DNDEBUG -DBUILD_A8 -DSOC_AM335x -IC:/ti/gcc-arm-none-eabi-4.9-2015q3/arm-none-eabi/include -I. -I. -Isrc/ip/elm/V0 -I. -Isrc/ip/gpio/V1 -I. -Isrc/ip/gpmc/V1 -I. -Isrc/ip/i2c/V2 -I. -Isrc/ip/mcasp/V1 -I. -Isrc/ip/mcspi/V0 -I. -Isrc/ip/mmc/V0 -I. -Isrc/ip/qspi/V1 -I. -Isrc/ip/timer/V1 -I. -Isrc/ip/uart/V1 -I/ti/PDK_AM~3/packages -I/ti/PDK_AM~3/packages/ti/csl -I/ti/PDK_AM~3/packages/ti/csl -I/apps/apps/nonbaw/inc -I/ -I/common -MF /ti/PDK_AM~3/packages/ti/binary/ti/csl/obj/am335x/a8/release/gpio_v2.d -MT /ti/PDK_AM~3/packages/ti/binary/ti/csl/obj/am335x/a8/release/gpio_v2.oa8fg -o /ti/PDK_AM~3/packages/ti/binary/ti/csl/obj/am335x/a8/release/gpio_v2.oa8fg src/ip/gpio/V1/priv/gpio_v2.c
```

编译完成:

```
gmake[3]: Leaving directory `c:/ti/PDK_AM~3/packages/ti/starterware/examples/lcdc/raster'
gmake[2]: Leaving directory `c:/ti/PDK_AM~3/packages/ti/starterware'
gmake -C ./ti/starterware mcspi_app_flash PROFILE=debug PLATFORM=am335x-evm -s KW_BUILD=no
gmake[2]: Entering directory `c:/ti/PDK_AM~3/packages/ti/starterware'
gmake[3]: Entering directory `c:/ti/PDK_AM~3/packages/ti/starterware/examples/mcspi/flash'
# Making am335x-evm:a8host:debug:board...
# Making am335x-evm:a8host:debug:utils...
# Making am335x-evm:a8host:debug:soc...
# Making am335x-evm:a8host:debug:dal...
# Compiling am335x-evm:a8host:debug:mcspi_app_flash: ../mcspi_app.c
# Compiling am335x-evm:a8host:debug:mcspi_app_flash: flash_app_main.c
# Linking into c:/ti/PDK_AM~3/packages/ti/starterware/binary/mcspi_app_flash/bin/am335x-evm/gcc/mcspi_app_flash_a8host_debug.out...
#
# c:/ti/PDK_AM~3/packages/ti/starterware/binary/mcspi_app_flash/bin/am335x-evm/gcc/mcspi_app_flash_a8host_debug.out created.
#
# Generating c:/ti/PDK_AM~3/packages/ti/starterware/binary/mcspi_app_flash/bin/am335x-evm/gcc/mcspi_app_flash_a8host_debug.bin...
# c:/ti/PDK_AM~3/packages/ti/starterware/binary/mcspi_app_flash/bin/am335x-evm/gcc/mcspi_app_flash_a8host_debug.bin created.
#
# Generating c:/ti/PDK_AM~3/packages/ti/starterware/binary/mcspi_app_flash/bin/am335x-evm/gcc/mcspi_app_flash_a8host_debug_ti.bin...
# c:/ti/PDK_AM~3/packages/ti/starterware/binary/mcspi_app_flash/bin/am335x-evm/gcc/mcspi_app_flash_a8host_debug_ti.bin created.
#
gmake[3]: Leaving directory `c:/ti/PDK_AM~3/packages/ti/starterware/examples/mcspi/flash'
gmake[2]: Leaving directory `c:/ti/PDK_AM~3/packages/ti/starterware'
gmake -C ./ti/starterware mcspi_app_flash PROFILE=release PLATFORM=am335x-evm -s KW_BUILD=no
gmake[2]: Entering directory `c:/ti/PDK_AM~3/packages/ti/starterware'
gmake[3]: Entering directory `c:/ti/PDK_AM~3/packages/ti/starterware/examples/mcspi/flash'
# Making am335x-evm:a8host:release:board...
# Making am335x-evm:a8host:release:utils...
# Making am335x-evm:a8host:release:soc...
# Making am335x-evm:a8host:release:dal...
# Compiling am335x-evm:a8host:release:mcspi_app_flash: ../mcspi_app.c
# Compiling am335x-evm:a8host:release:mcspi_app_flash: flash_app_main.c
# Linking into c:/ti/PDK_AM~3/packages/ti/starterware/binary/mcspi_app_flash/bin/am335x-evm/gcc/mcspi_app_flash_a8host_release.out...
#
# c:/ti/PDK_AM~3/packages/ti/starterware/binary/mcspi_app_flash/bin/am335x-evm/gcc/mcspi_app_flash_a8host_release.out created.
#
# Generating c:/ti/PDK_AM~3/packages/ti/starterware/binary/mcspi_app_flash/bin/am335x-evm/gcc/mcspi_app_flash_a8host_release.bin...
# c:/ti/PDK_AM~3/packages/ti/starterware/binary/mcspi_app_flash/bin/am335x-evm/gcc/mcspi_app_flash_a8host_release.bin created.
#
# Generating c:/ti/PDK_AM~3/packages/ti/starterware/binary/mcspi_app_flash/bin/am335x-evm/gcc/mcspi_app_flash_a8host_release_ti.bin...
# c:/ti/PDK_AM~3/packages/ti/starterware/binary/mcspi_app_flash/bin/am335x-evm/gcc/mcspi_app_flash_a8host_release_ti.bin created.
#
gmake[3]: Leaving directory `c:/ti/PDK_AM~3/packages/ti/starterware/examples/mcspi/flash'
gmake[2]: Leaving directory `c:/ti/PDK_AM~3/packages/ti/starterware'
gmake[1]: Leaving directory `c:/ti/PDK_AM~3/packages'
c:\ti\processor_sdk_rtos_am335x_3_03_00_04>
```

参考: http://processors.wiki.ti.com/index.php/Processor_SDK_RTOS_Building_The_SDK

至此, 编译工作已经完成。将 MLO (bootloader) 和 app 烧录到 AMIC110 ICE SPI 中, 即可使其工作。

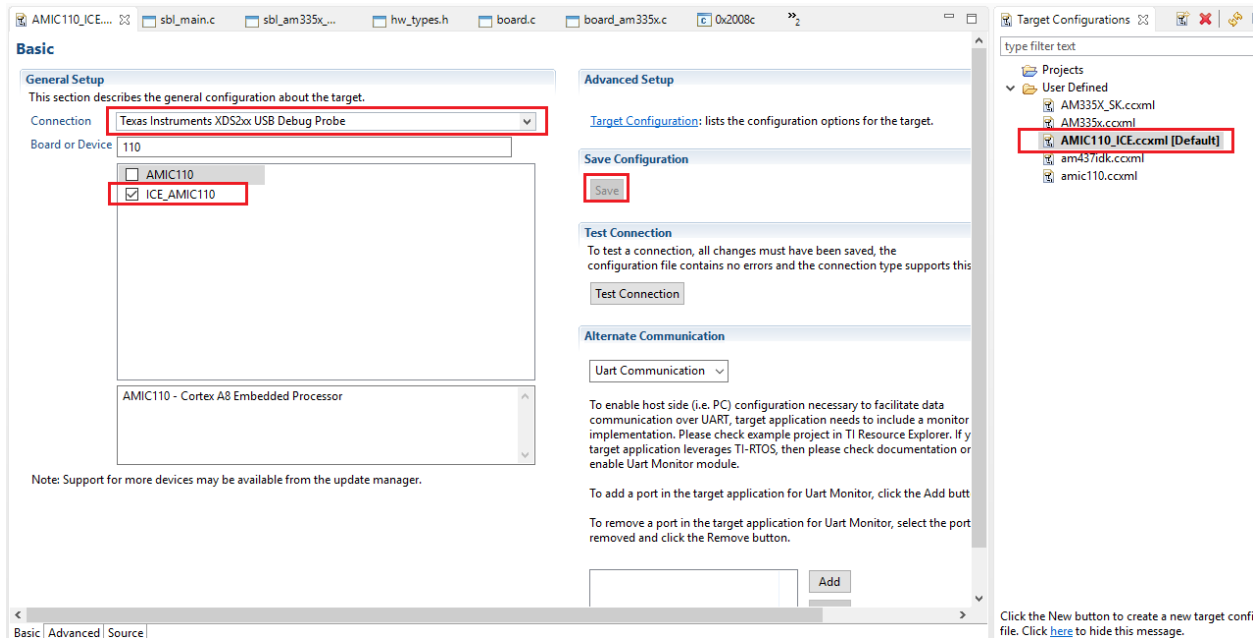
PS: 编译成功的 bootloader image 文件在以下文件夹下:

C:\ti\pdk_am335x_1_0_6\packages\ti\starterware\binary\bootloader\bin\am335x-evm\gcc

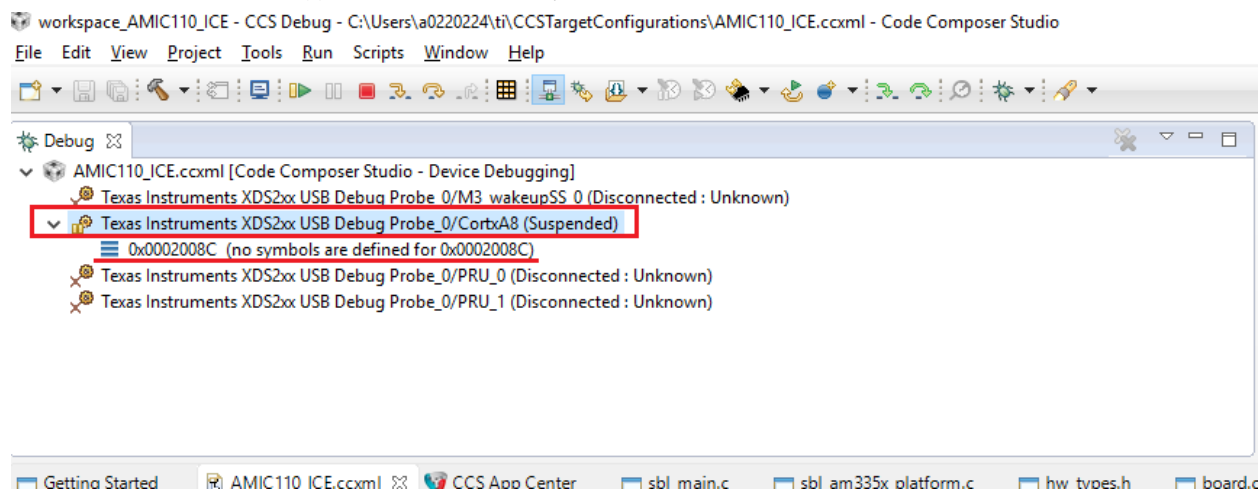
Step 3 烧录到 SPI flash 中的需要使用 CCS 运行 flash write。

使用步骤:

1. 打开 CCS，从 View->Target Configuration 中，右键点击 User Defined 文件夹，新建一个 ccxml 文件，并从中选择你的仿真器型号和板卡型号。比如我建立的是 AMIC110_ICE.ccxml，使用的是 XDS200 的仿真器，板子是 AMIC 110 ICE 板卡，选择如下：



2. Save 保存后，接好仿真器，给板子通上电后，即可右键点击 AMIC110_ICE.ccxml，选择 Launch Selected Configuration。这时候就会进入到 debug 界面，然后在 Cortex A8 那里右键点击选择 Connect Target。这时候，如果正常连接，就能看到地址信息（如下图的 0x0002008C，这是当前指针停的位置），同时 GEL 文件就会自动运行，GEL 中会初始化 ARM 和 DDR，这样后续就可以正常的 load 文件从而运行了。参考如下：



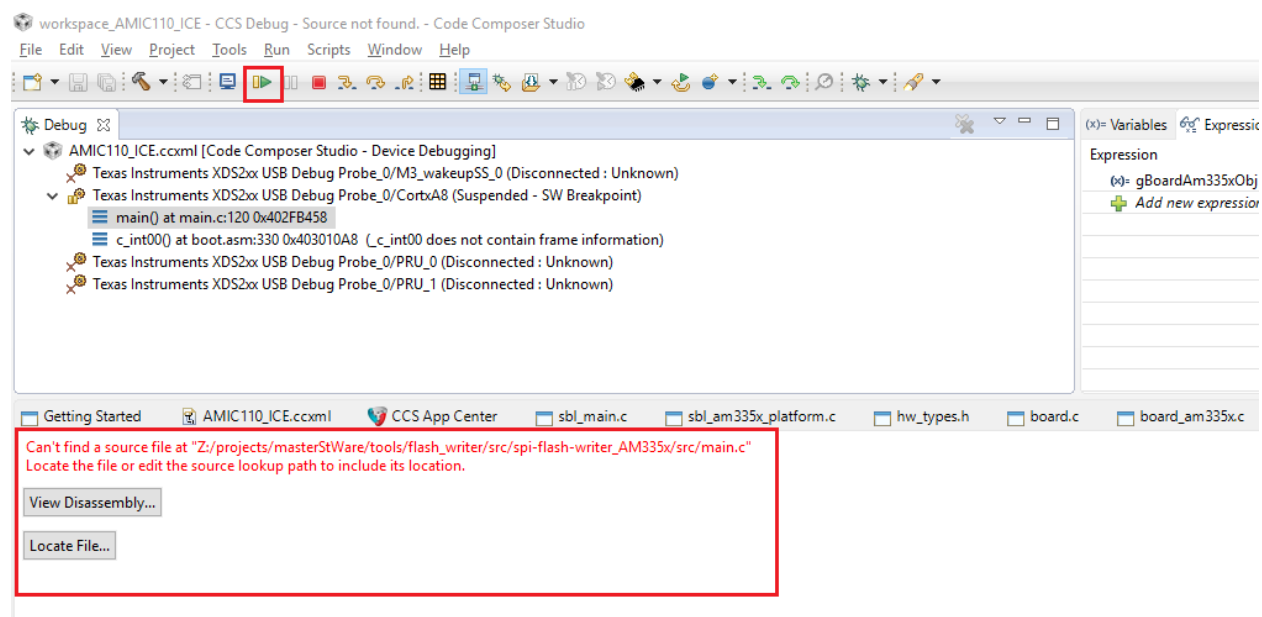
3. Load 烧写 flash 的文件，下列文件即为可以进行 SPI flash 烧录的工具文件：

C:\ti\pdk_am335x_1_0_6\packages\ti\starterware\tools\flash_writer\sbi_flash_writer_AM335X.out

这个是已经编译好的，load 进去执行即可。

操作：在 debug 界面下，选择 Run->Load->Load Program，在弹出的选项卡中选择 browse，选择上面的 out 文件，点击 OK 即可进行 load，出现如下界面的时候点击绿色播放按钮，即可正常执行。

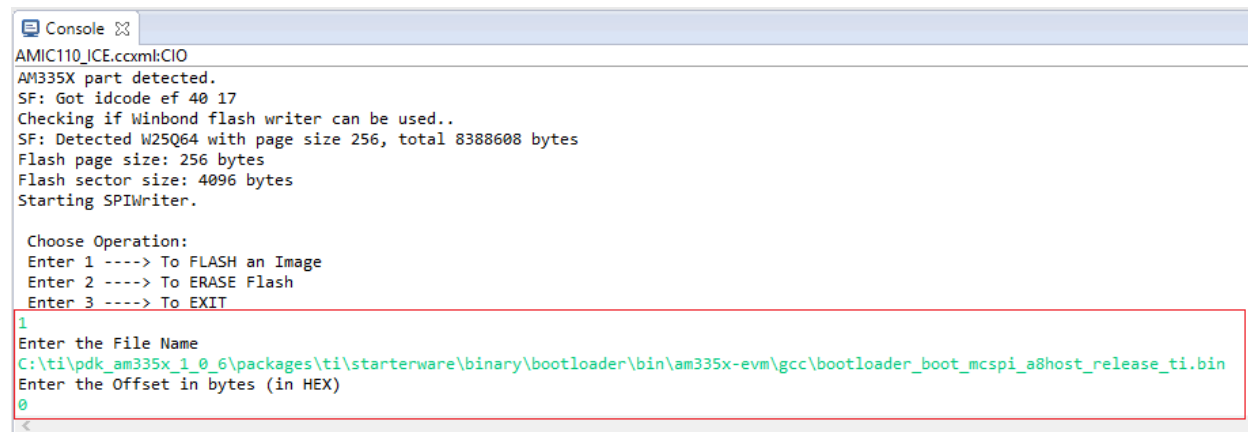
PS: 红框部分，只是因为没有找到源文件关联才出现的提示，所以不用管它，正常执行即可。



4. 烧录 bootloader 到地址 0x00000000 区间上。

在点击运行之后，在 console 中即会出现相关的打印信息，按照提示操作即可进行文件的烧录。

Bootloader 的文件名是 bootloader_boot_mcspi_a8host_release_ti.bin，注意在输入文件名的时候，需要输入完整的路径，如下所示：



正常烧录完成后，即可看到如下提示：

```
Console
AMIC110_ICE.ccxml:CIO
Erasing flash at byte offset: 0, byte length: 36864
SE: cmd = { 0x20 0x000000 }
SE: cmd = { 0x20 0x001000 }
SE: cmd = { 0x20 0x002000 }
SE: cmd = { 0x20 0x003000 }
SE: cmd = { 0x20 0x004000 }
SE: cmd = { 0x20 0x005000 }
SE: cmd = { 0x20 0x006000 }
SE: cmd = { 0x20 0x007000 }
SE: cmd = { 0x20 0x008000 }
SF: Successfully erased 36864 bytes @ 0x9000
Writing flash at page offset: 0, number of pages: 134

Flashing is in progress...
Verifying... Success.
```

此时 bootloader 烧录完成。

5. 烧录 app 到地址 0x20000 区间上。

App 即为 EtherCAT slave 的应用程序。

这里可以直接 download EtherCAT slave 编译好的 prebuilt binaries。下载后解压，适用于 AMIC110 ICE 开发板的即为：C:\ti\PRU-ICSS-

EtherCAT_Slave_01.00.03.01\pru_icss_ethercat_slave_prebuilt_01.00.03.01\applications\ethercat_slave_demo_arm\AMIC11x_release 目录下的 app 文件

下载链接：

http://software-dl.ti.com/processor-industrial-sw/esd/PRU_ICSS_EtherCAT_Slave/latest/index_FDS.html

编译 EtherCAT slave 的 app

安装完 PRU-ETHERCAT 的软件后，可以在 C:\ti\PRU-ICSS-EtherCAT_Slave_01.00.03.01\docs 文件夹下查看 PRU_ICSS_EtherCAT_User_Guide.pdf，里面包含了所有的编译流程，描述如下：

1. 打开 windows 命令行，进入到如下文件夹：

C:\ti\PRU-ICSS-EtherCAT_Slave_01.00.03.01\protocols\ethercat_slave\projects

使用如下命令进行工程文件的创建：

projectCreate.bat AMIC11x arm ethercat_slave_demo

创建完成后，在 C:\ti\PRU-ICSS-EtherCAT_Slave_01.00.03.01\protocols\ethercat_slave\projects 文件夹下便会存在一个新的 ethercat_slave_demo_AMIC11x_arm 文件夹。该工程即为 CCS 工程。


```
C:\windows\system32\CMD.exe
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\ao220224>cd C:\ti\PRU-ICSS-EtherCAT_Slave_01.00.03.01\protocols\ethercat_slave\projects
C:\ti\PRU-ICSS-EtherCAT_Slave_01.00.03.01\protocols\ethercat_slave\projects>projectCreate.bat AMIC11x arm ethercat_slave_demo
*****
Environment Configuration Summary:
  CCS Install Directory      : C:\ti\ccsv7
  CCS Workspace Directory   : C:\Users\ao220224\ccs_workspace
  SDK Install Directory     : C:\ti\PRU-ICSS-EtherCAT_Slave_01.00.03.01
  PDK Install Directory    : C:\ti\pdk_am335x_1_0_6\packages
  Projects create Directory : C:\ti\PRU-ICSS-EtherCAT_Slave_01.00.03.01\protocols\ethercat_slave\projects
  Project create args       : C:\ti\PRU-ICSS-EtherCAT_Slave_01.00.03.01\protocols\ethercat_slave\projects\ccsproject_args
*****

Creating project 'ethercat_slave_demo_AMIC11x_arm' for 'AMIC11x' platform in directory 'C:\ti\PRU-ICSS-EtherCAT_Slave_01.00.03.01\protocols\ethercat_slave\projects\ethercat_slave_demo_AMIC11x_arm' by overwriting the project if it exists already...

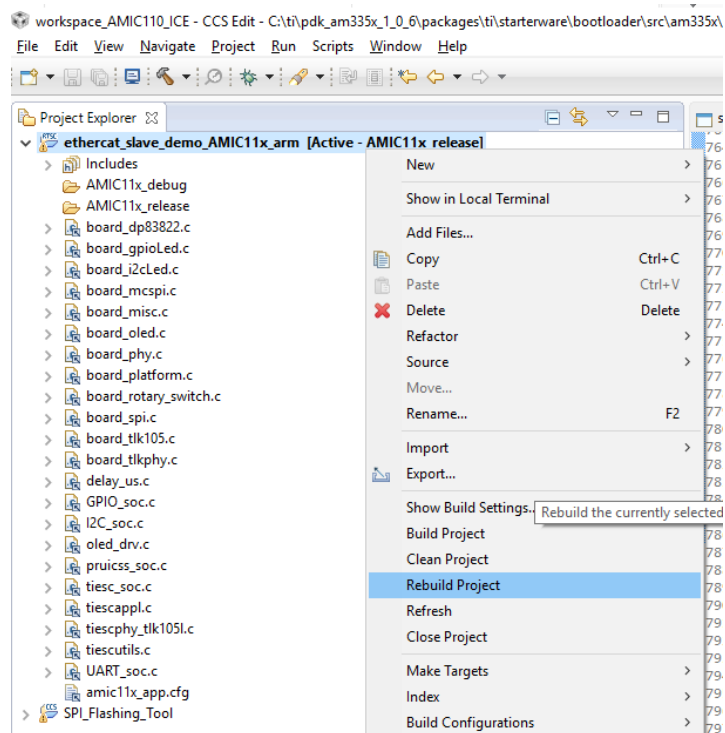
-----
Creating project 'ethercat_slave_demo_AMIC11x_arm'...

Done!

C:\ti\PRU-ICSS-EtherCAT_Slave_01.00.03.01\protocols\ethercat_slave\projects>
```

2. 使用 CCS 进行文件的导入编译。

在 CCS 中，选择 file->import，然后在 Code Composer Studio 下面选择 CCS Projects，然后选择到上述文件目录，即可完成导入。导入后，邮件选择该工程，进行工程的重建，rebuild project。正常情况下，即可编译生成 EtherCAT slave 的 demo 例程以及相关 app 文件。



PS: 对于 EtherCAT 的应用，一般建议在我们提供的 app 中直接进行修改。若是自建工程进行 RTOS 的学习、测试，一定要注意版本的对应关系，可以参考相关的 release notes

http://processors.wiki.ti.com/index.php?title=Processor_SDK_RTOS_Release_Notes&oldid=227097#Component_Version

Component	Version	AM335x
PDK	Per platform	1.0.6
SYS/BIOS	6.46.4.53	X
XDCTOOLS	3.32.1.22	X