

Creating a CCS Project for SBL on AM572x GP EVM

Overview

This page details how to port the makefile-based Secondary Boot Loader (SBL) to a Code Composer Studio (CCS) project. The SBL sets-up the PLL clocks, powers on the I/O Peripherals, initializes the DDR, loads the application image into DDR, and brings the slave cores for applicable SOCs out of reset. This guide uses the AM572x GP EVM as an example, but the steps provided here can be applied to any board.

Create a CCS Project

1. In Code Composer Studio, navigate to File->New->CCS Project.
2. In the New CCS Project window:
 - Select the Target Board. For this example, we select the GPEVM_AM572X_SiRevA.
 - Select the Cortex A [ARM] tab.
 - Enter a project name.
 - Select a GNU Compiler.
 - Select 'Empty Project.'
 - Click Finish.

Add Include Directories

1. In Project Properties, navigate to CCS Build->GNU Compiler->Directories.
2. Add the include directories listed in the makefile. The makefile can be found in the directory:
C:/ti/pdk_x_x_x/packages/ti/boot/sbl/board/{BOARD}/build

For the evmAM572x, add the following paths:

```
{PDK_INSTALL_PATH}
{PDK_INSTALL_PATH}/ti/csl
{PDK_INSTALL_PATH}/ti/board
{SBL_SRC_DIR}/board/src
{SBL_SRC_DIR}/src/tpc
{SBL_SRC_DIR}/soc/am57xx
```

3. If the boot mode is set to MMCSDB, the directory '{SBL_SRC_DIR}/src/mmcsdb' also needs to be added.

PDK_INSTALL_PATH refers to the directory 'C:/ti/pdk_x_x_x/packages'

SBL_SRC_DIR refers to the directory 'C:/ti/pdk_x_x_x/packages/ti/boot/sbl'

Add Libraries

1. In Project Properties, navigate to CCS Build->GNU Linker->Libraries.

2. Add the libraries listed in the makefile by doing the following:

- In the Libraries window, add the libraries in the following format:

```
:ti.board.aa15fg
```

```
:ti.drv.uart.aa15fg
```

```
:ti.drv.i2c.aa15fg
```

```
:ti.drv.mmcsd.aa15fg
```

```
:ti.fs.fatfs.aa15fg
```

```
:ti.csl.aa15fg
```

```
:ti.csl.init.aa15fg
```

```
:ti.osal.aa15fg
```

```
:pm_hal.aa15fg
```

- In the Library search path window, add the corresponding library paths:

```
${PDK_INSTALL_PATH}/ti/board/lib/evmAM572x/a15/release/
```

```
${PDK_INSTALL_PATH}/ti/drv/uart/lib/a15/release/
```

```
${PDK_INSTALL_PATH}/ti/drv/i2c/lib/a15/release/
```

```
${PDK_INSTALL_PATH}/ti/drv/mmcsd/lib/a15/release/
```

```
${PDK_INSTALL_PATH}/ti/fs/fatfs/lib/a15/release/
```

```
${PDK_INSTALL_PATH}/ti/csl/lib/am572x/a15/release/
```

```
${PDK_INSTALL_PATH}/ti/csl/lib/am572x/a15/release/
```

```
${PDK_INSTALL_PATH}/ti/osal/lib/nonos/am572x/a15/release/
```

```
${PDK_INSTALL_PATH}/ti/drv/pm/lib/am572x/a15/release/
```

Add Compiler Flags

1. In Project Properties, navigate to CCS Build->GNU Compiler and click 'Edit Flags...'

2. Add the compiler flags listed in the makefile.

For the evmAM572x makefile, this consists of the following:

```
-g -gdwarf-3 -gstrict-dwarf -Wall -D__ARMv7 -DSOC_AM572x  
-mtune=cortex-a15 -march=armv7-a -marm -mfloat-abi=hard -mfpu=neon  
-DAM572x_BUILD -DevmAM572x -DOPP_NOM
```

3. If the boot mode is set to MMCSd, the flag '-DBOOT_MMCSd' also needs to be added.

Add Linker Flags

1. In Project Properties, navigate to CCS Build->GNU Linker and click 'Edit Flags...'
2. Add the linker flags listed in the makefile.

For the evmAM572x makefile, this consists of the following:

```
-mfloat-abi=hard -Wl,--undefined,__aeabi_uidiv
-Wl,--undefined,__aeabi_idiv --entry Entry -nostartfiles -static
-Wl,--gc-sections -Wl,-T
${SBL_SRC_DIR}/board/evmAM572x/build/linker.cmd -lgcc -lc -lrdimon
```

Add Source Files to Project Folder

1. Add the source files listed in the makefile.

For the evmAM572x, this includes the following:

sbl_main.c in the directory \${PDK_INSTALL_PATH}\ti\boot\sbl\board\evmAM572x

sbl_rprc.c in the directory \${PDK_INSTALL_PATH}\ti\boot\sbl\src\rprc

sbl_prcm.c in the directory \${PDK_INSTALL_PATH}\ti\boot\sbl\soc\am57xx

sbl_slave_core_boot.c in the directory \${PDK_INSTALL_PATH}\ti\boot\sbl\soc\am57xx

sbl_init.S in the directory \${PDK_INSTALL_PATH}\ti\boot\sbl\soc\am57xx

sbl_startup.c in the directory \${PDK_INSTALL_PATH}\ti\boot\sbl\soc\am57xx

sbl_avs_config.c in the directory \${PDK_INSTALL_PATH}\ti\boot\sbl\board\src

UART_soc.c in the directory \${PDK_INSTALL_PATH}\ti\drv\uart\soc\am572x

2. If the boot mode is set to MMCSD, the following source files also need to be added:

sbl_mmcSD.c in the directory \${PDK_INSTALL_PATH}\ti\boot\sbl\src\mmcSD

MMCSD_soc.c in the directory \${PDK_INSTALL_PATH}\ti\drv\mmcSD\soc\am57xx

3. Remove the files 'startup_ARMCA15.S' and 'AM572x.lsd' from the project folder.

Configure the Link Order

In Project Properties, navigate to CCS Build -> Link Order and set the link order as shown below.

Placing sbl_init.S at the top of the link order ensures that the Entry symbol is placed at the correct location.

Test the Project



To test the project, we need a bootable SD card. To create a bootable SD card, follow the guides posted here http://processors.wiki.ti.com/index.php/Processor_SDK_RTOS_How_To_Guides.

1. Build the project.
2. Launch the target configuration.
3. Remove the SD card from the EVM.
4. Connect to the Cortex A15 core on the EVM.
5. Insert the SD card.
6. Load the .out file created by the SBL project onto the A15 core and run it.
7. The terminal window should now show the SBL booting the application from the SD card and the application running.

Article Sources and Contributors

Creating a CCS Project for SBL on AM572x GP EVM *Source:* <http://processors.wiki.ti.com/index.php?oldid=232041> *Contributors:* A0226755