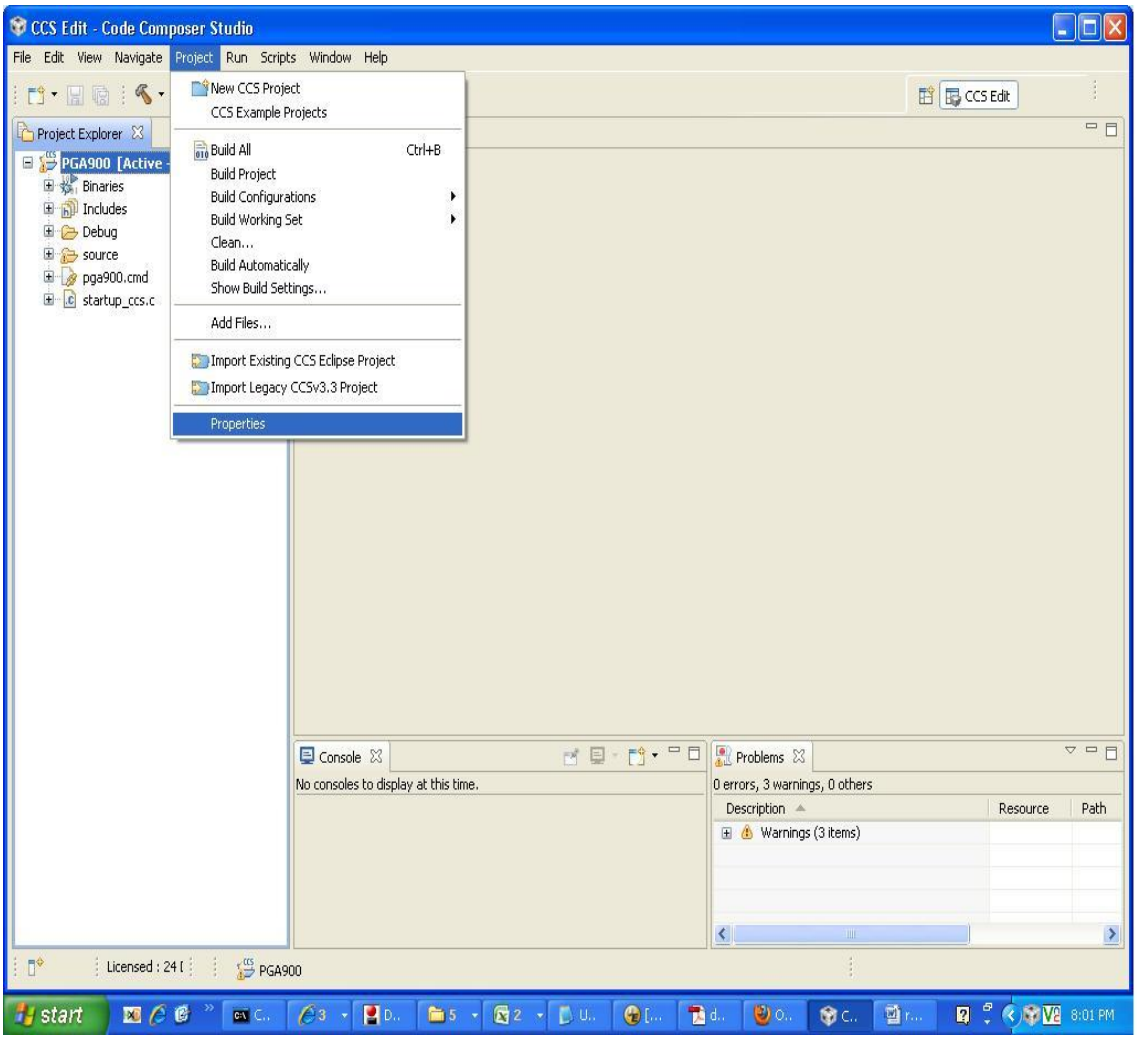
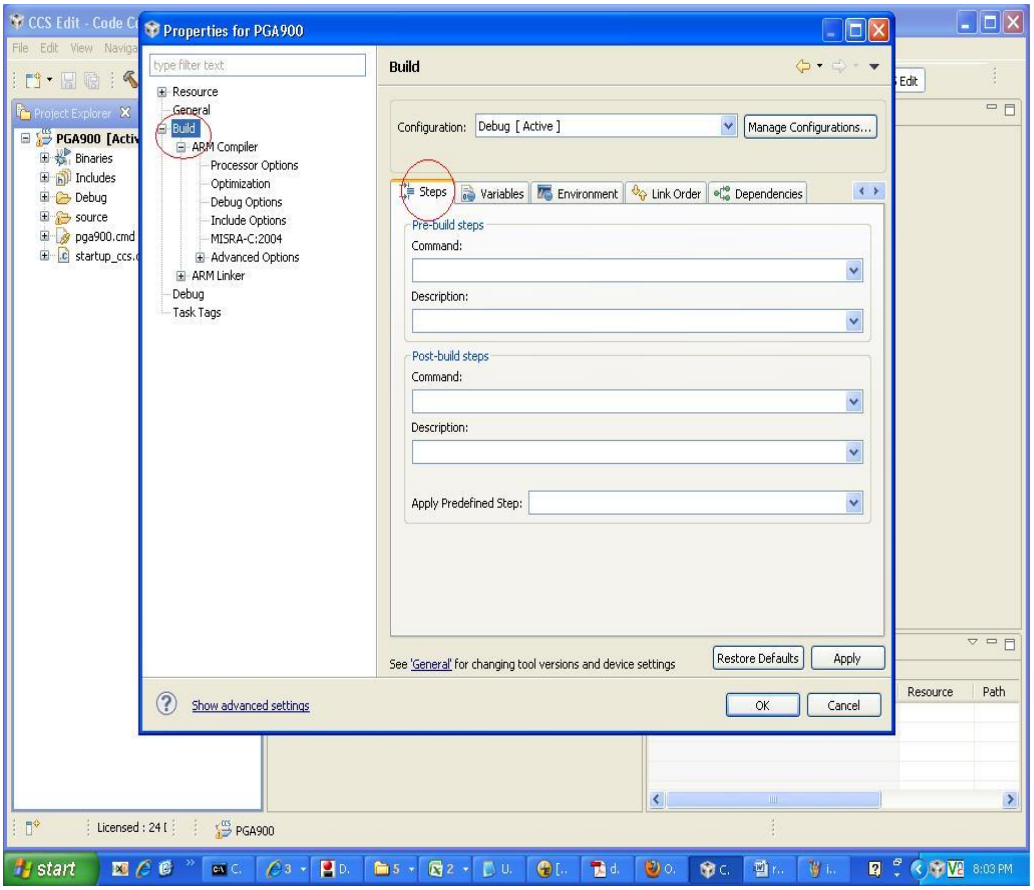
1 Procedure to Generate Intel Hex File

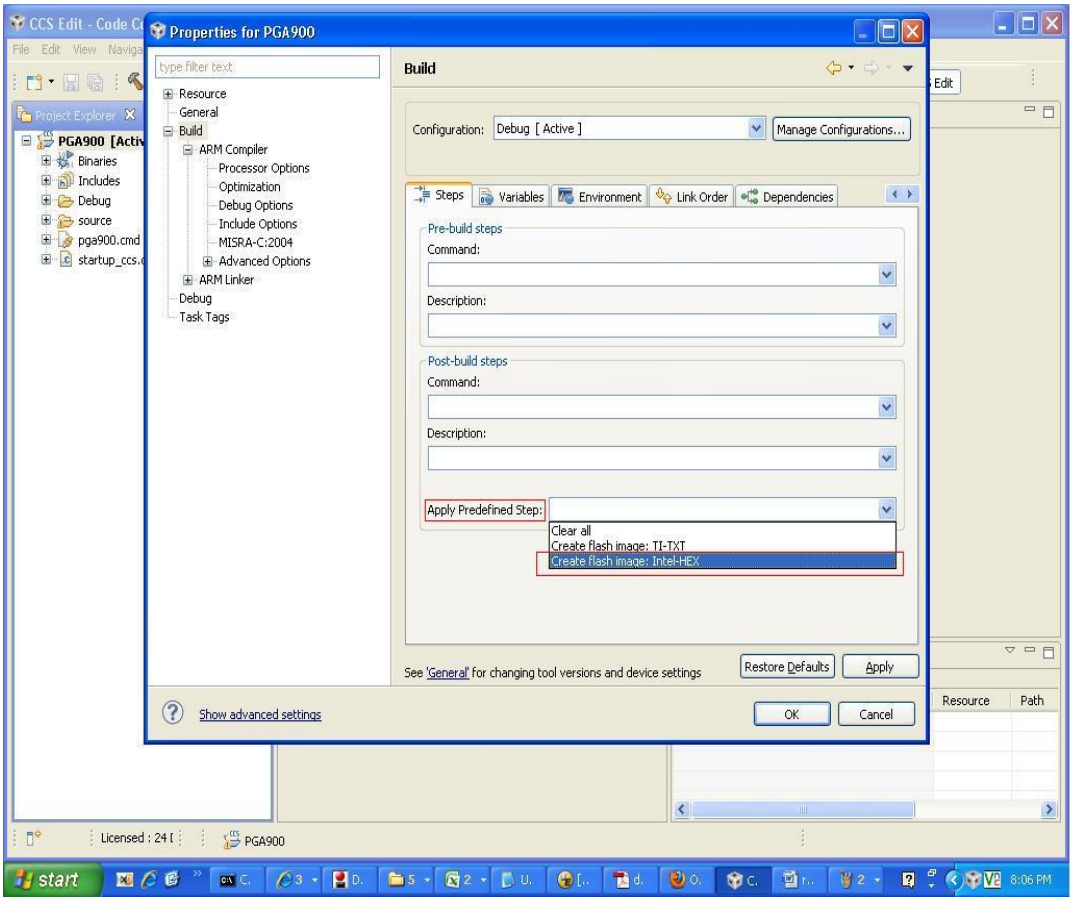
1. Go to menu Project->Properties



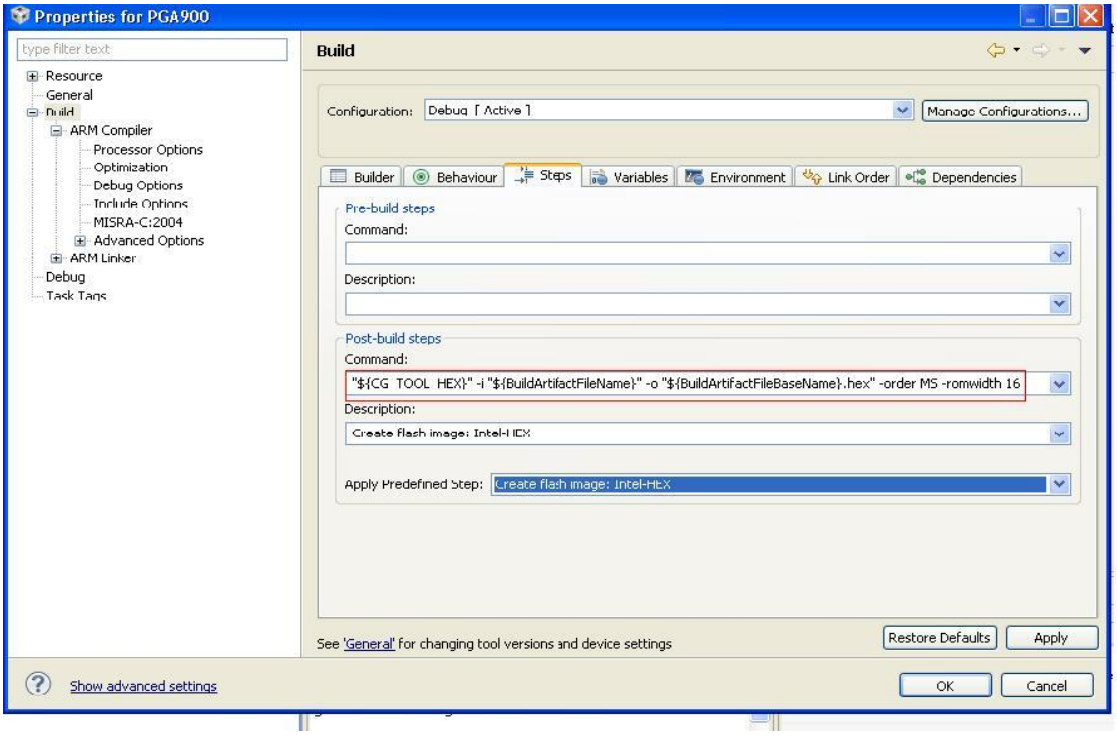
2. Select ‘Build’ and user can see ‘Steps’ as shown below:



3. ‘Apply Predefined Step:’ contains drop down box and select ‘Create flash image:Intel-Hex’ as shown below:



4. Selection of ‘Create flash image:Intel-Hex’ shows



Command:

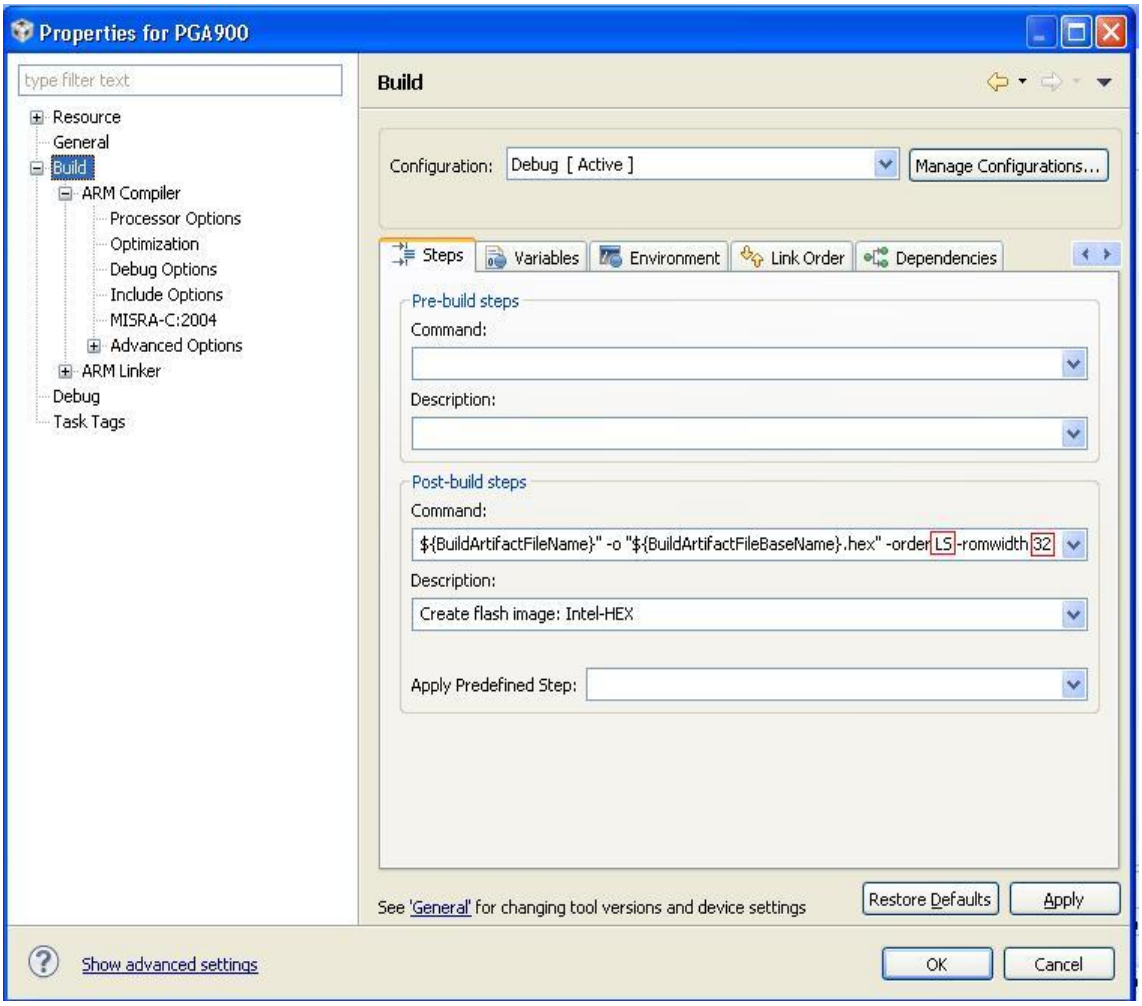
It shows default options as

"${CG_TOOL_HEX}" -i "${BuildArtifactFileName}" -o "${BuildArtifactFileBaseName}.hex" -order MS -romwidth 16

-order MS – indicates big-endian and -romwidth 16 – indicates hex conversion formats width.

5. PGA900 is little-endian (LS) and Intel hex conversion format width is 32bits.

   Modify these two options as shown below:



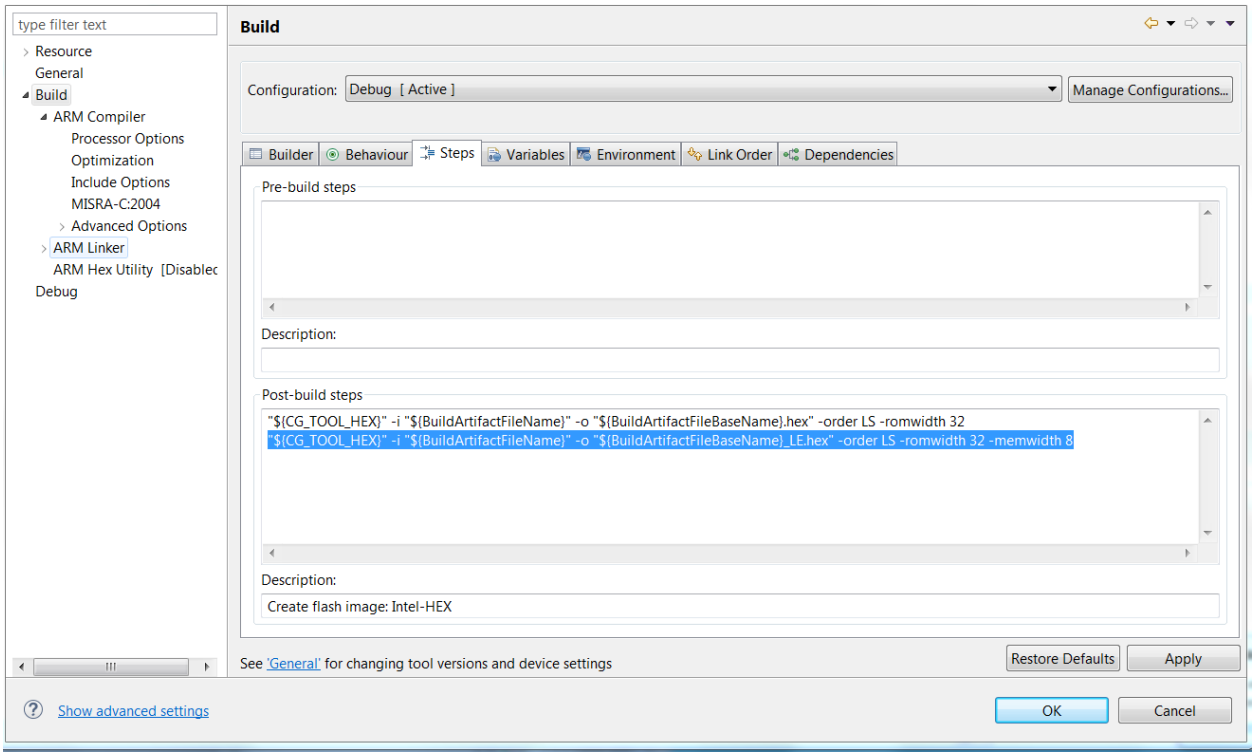
6. Click on Apply button.

7. Add the following command to post-build steps

   "${CG_TOOL_HEX}" -i "${BuildArtifactFileName}" -o "${BuildArtifactFileBaseName}_LE.hex" -order LS -romwidth 32 -memwidth 8

8. Click on Apply button

9. Screenshot is shown below:

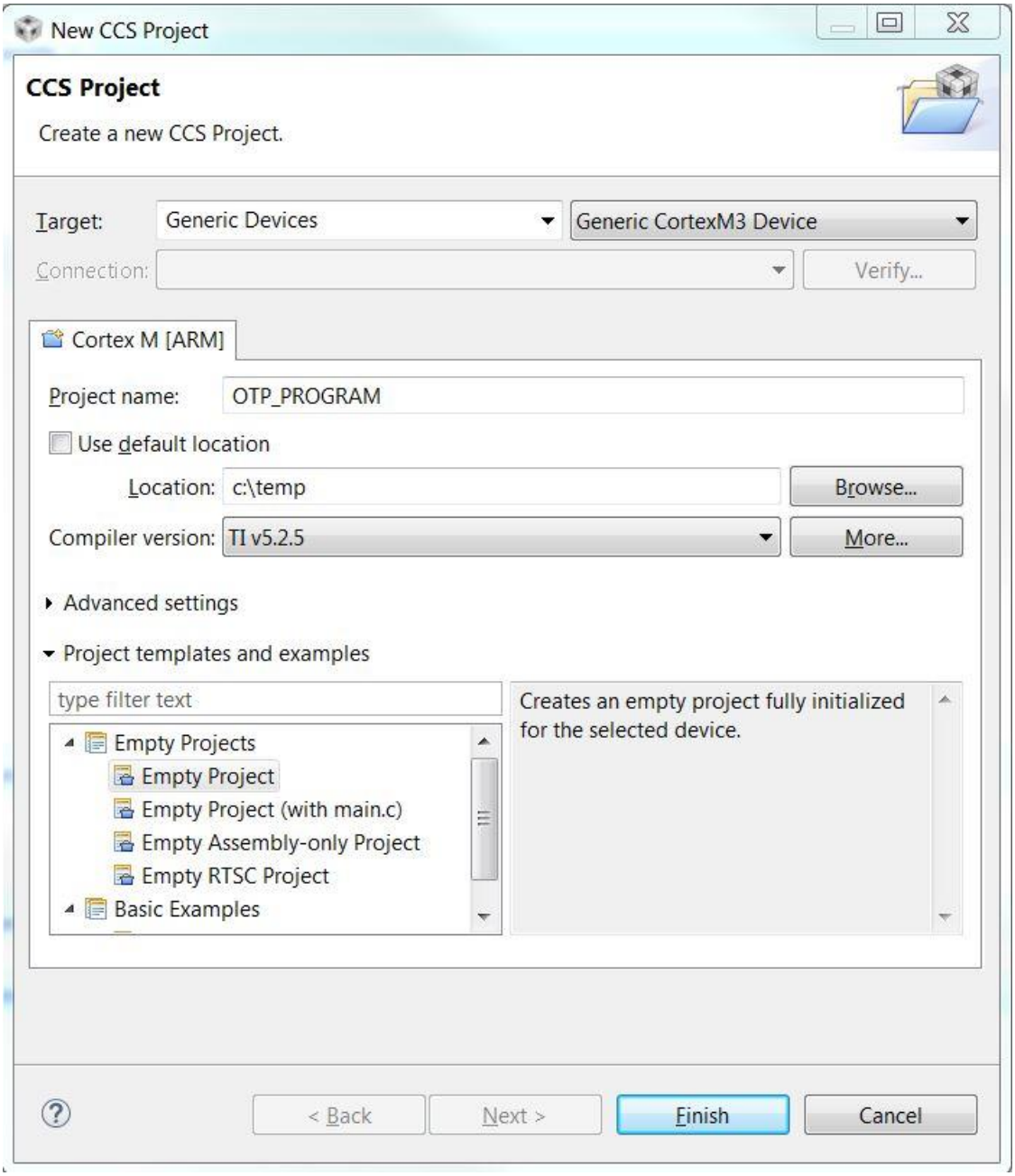


10. Click on OK button.
11. Compile PGA900 code and Intel hex file (PGA900.hex and PGA900_LE.hex) will get generated in Debug folder.

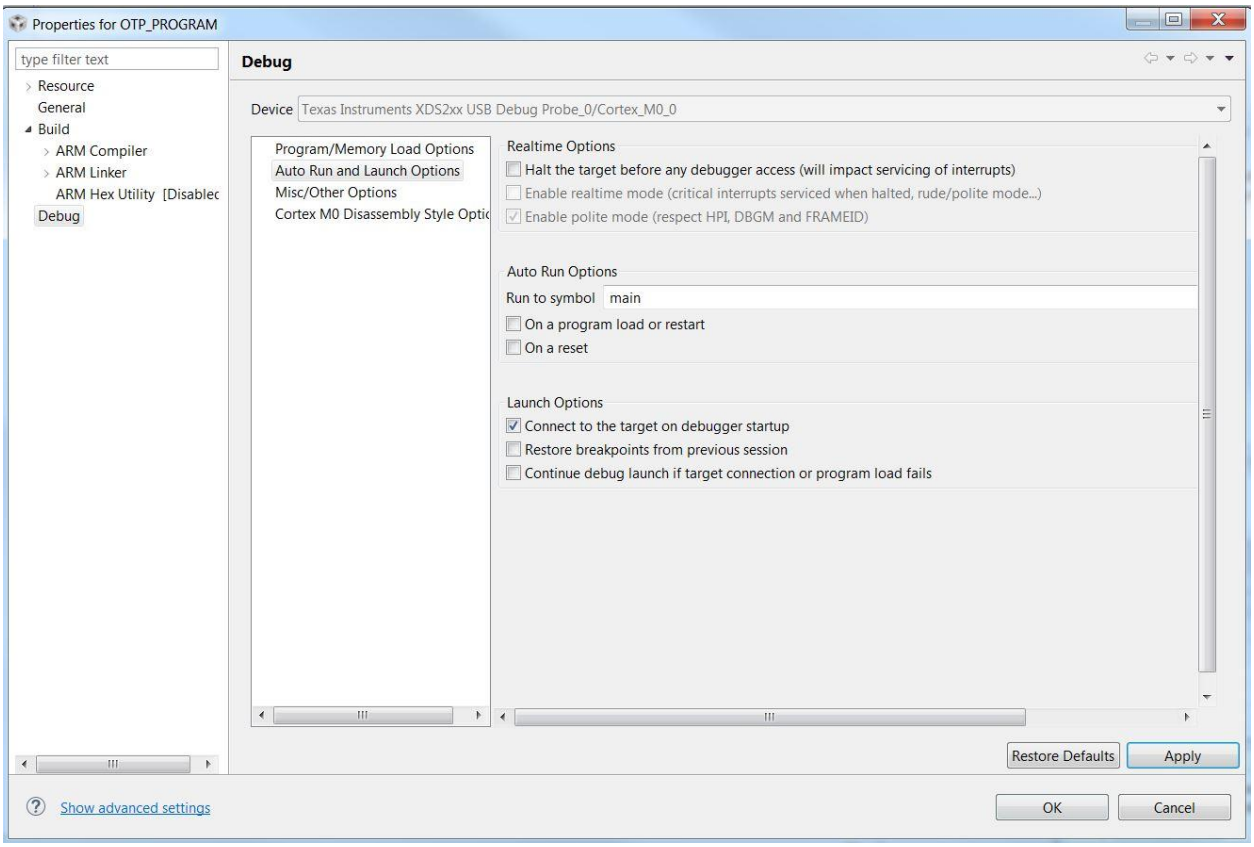
2 Programming OTP using XDS200 USB JTAG emulator

2.1.1 Creating and Configuring CCS project for OTP Programming

1. Launch CCS and create an Empty Project (See screenshot below)

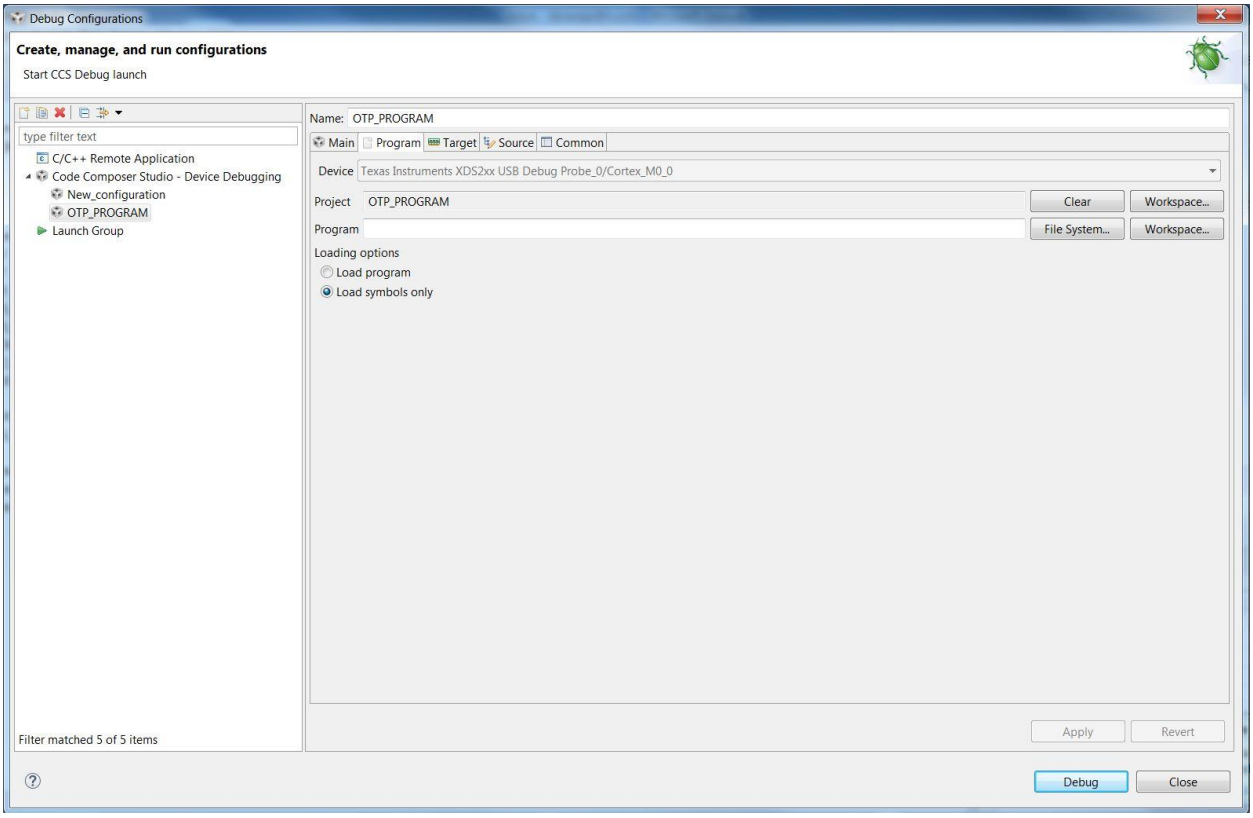


2. Configure the project to disable autorun and to restore breakpoints from previous session (See screenshot below)

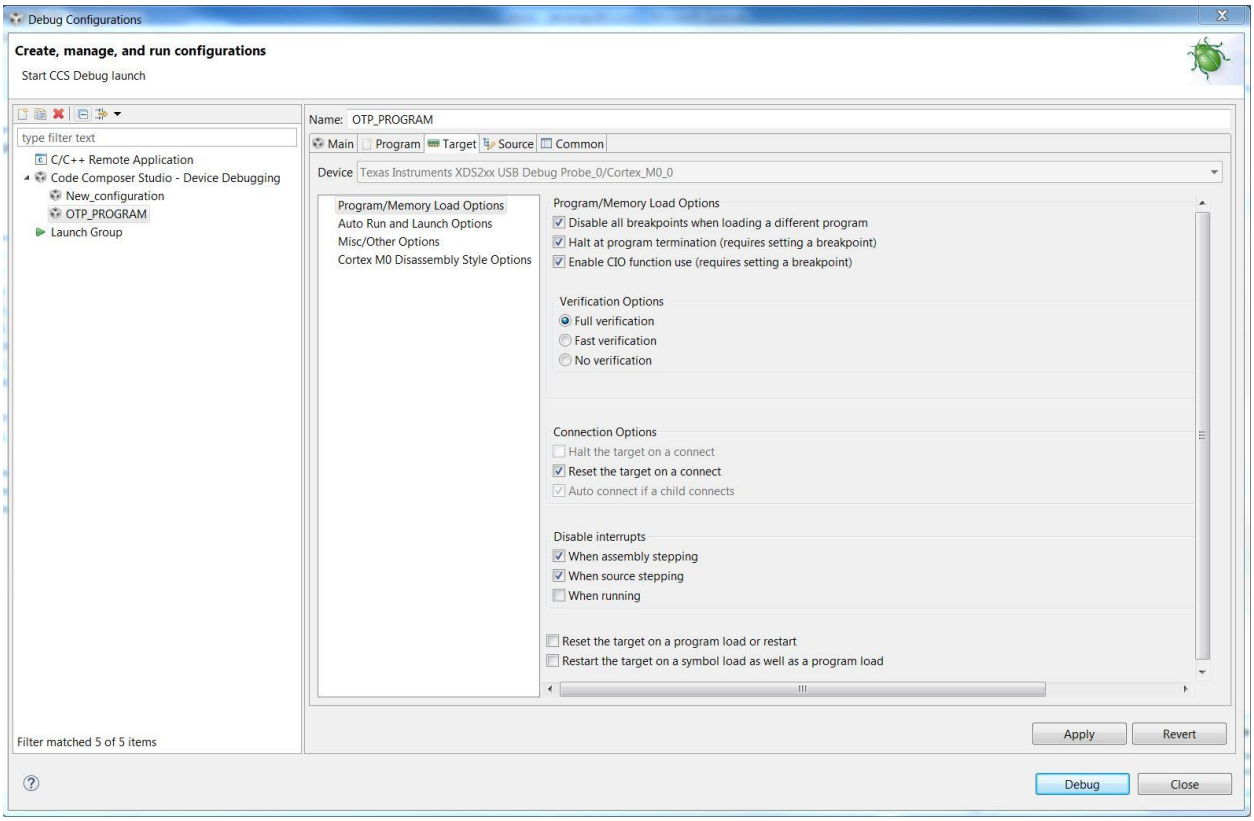


3. Update the debug configuration for the project by selecting the debug dropdown menu and selecting debug configuration (See screenshot below)

   a. Configure the Program section to load symbols only and delete the entry in the Program Field (See screenshot below):



b. Configure the Target section for Full Verification of program



2.1.2 Creating GEL file for OTP Programming

a. Copy pga900.gel from C:\ti\ccsv5\ccs_base\emulation\gel\
(if CCS is installed on C:\ drive otherwise select appropriate drive) to
C:\ti\ccsv5\ccs_base\emulation\gel\pga900_otp_program.gel
(if CCS is installed on C:\ drive otherwise select appropriate drive)

b. Make the following changes to pga900_otp_program.gel

   a. Change:

   Config_Remap()

   {

   WR_MEM_08(REMAP_ADDR,0x01);

   }

   To

   Config_Remap()

{

WR_MEM_08(REMAP_ADDR,0x00);

}

b. Change:

GEL_MapAddStr(0x00000000, 0, 0x00002000, "R|W", 0); // OTP

To

GEL_MapAddStr(0x00000000, 0, 0x00002000, "RAM|AS4", 0); // OTP

c. Add following function:

OnPreFileLoaded()

{

GEL_TextOut("**** PGA900 OnPreFileLoaded ..... \n","Output",1,1,1);

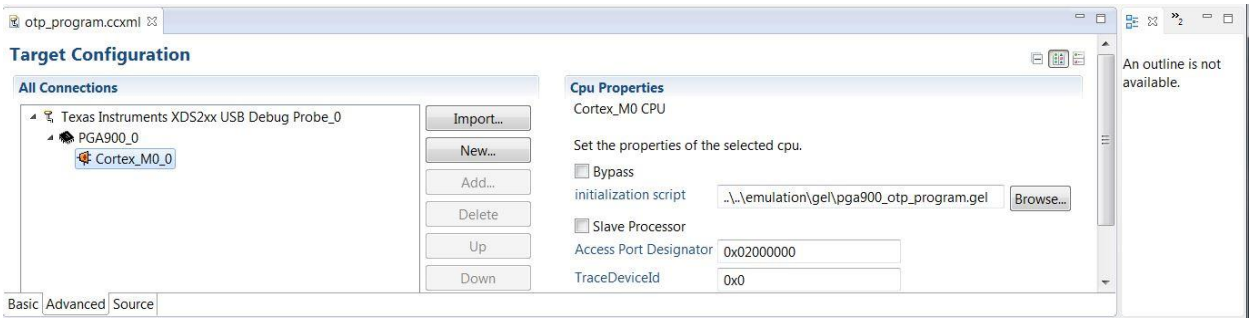
Config_Remap();

GEL_TextOut("**** PGA900 OnPreFileLoaded is done **** \n","Output",1,1,1);

}

2.1.3 Creating Target Configuration file

Setup the Target Configuration Options as described in Section 4.2). Select pga900_otp_program.gel as the initialization script instead of pga900.gel (Screen shot shown below).

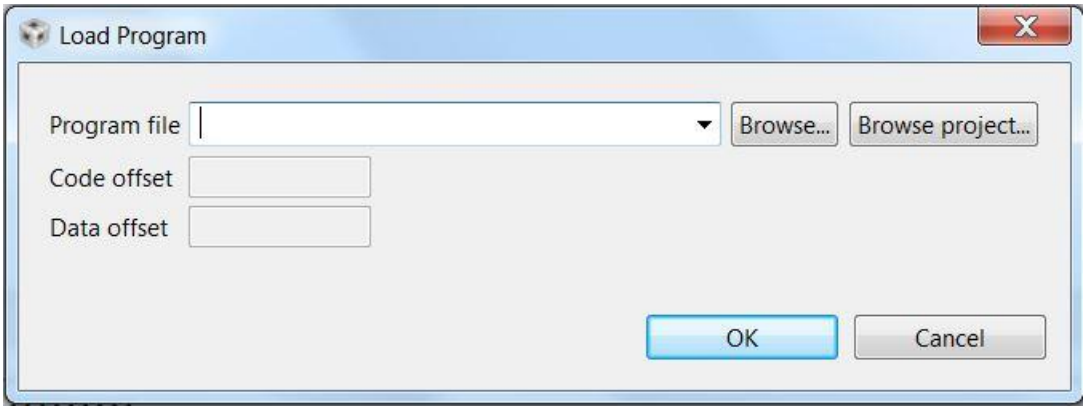


2.1.4 Programming OTP memory

1. Connect to the PGA900 device using the XDS200 USB JTAG emulator

2. Proceed to load program by clicking the load program icon (circled in RED below):

3. Select hex file PGA900_LE.hex to be programmed into OTP



4. Apply programming power to OTP

5. Click OK

6. Following messages will be displayed:

```
Cortex_M0_0: Output: ****  PGA900 OnPreFileLoaded ..........
Cortex_M0_0: Output: ****  PGA900 OnPreFileLoaded is done ****
Cortex_M0_0: Output: ****  PGA900 OnRestart ..........
Cortex_M0_0: Output: ****  PGA900 OnRestart is done ****
Cortex_M0_0: Output: ****  PGA900 OnFileLoaded ..........
Cortex_M0_0: Output: ****  PGA900 OnFileLoaded is done ****
```

7. OTP is now programmed

8. Remove programming power to OTP

9. Disconnect XDS200 USB JTAG emulator from the PGA900 by clicking on the disconnect icon (circled in RED below)