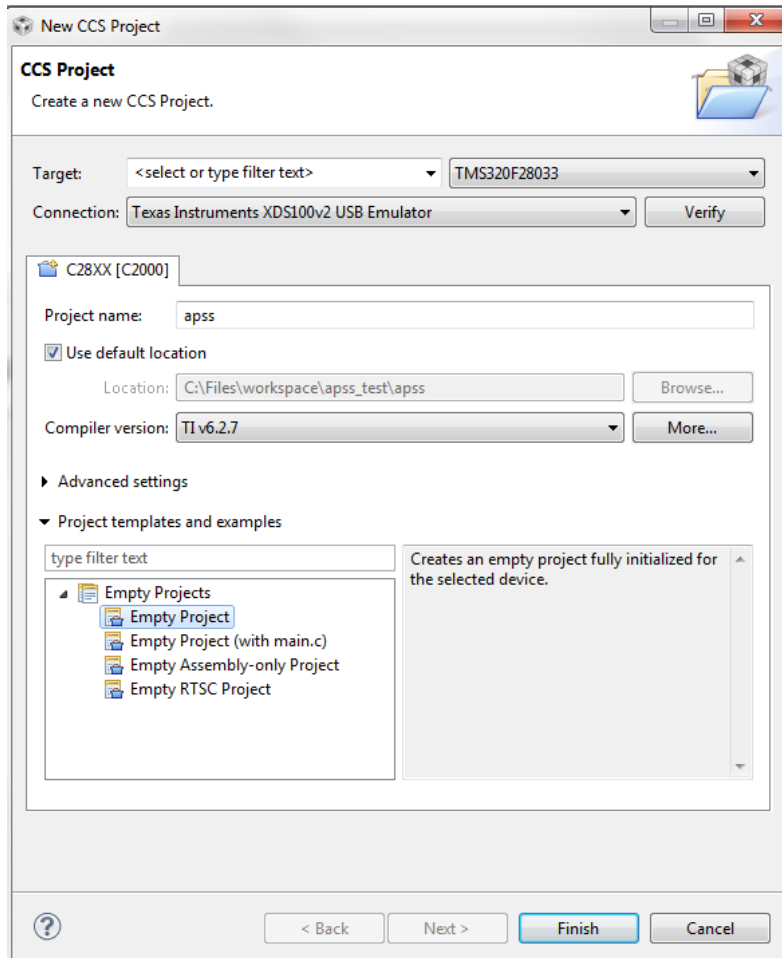
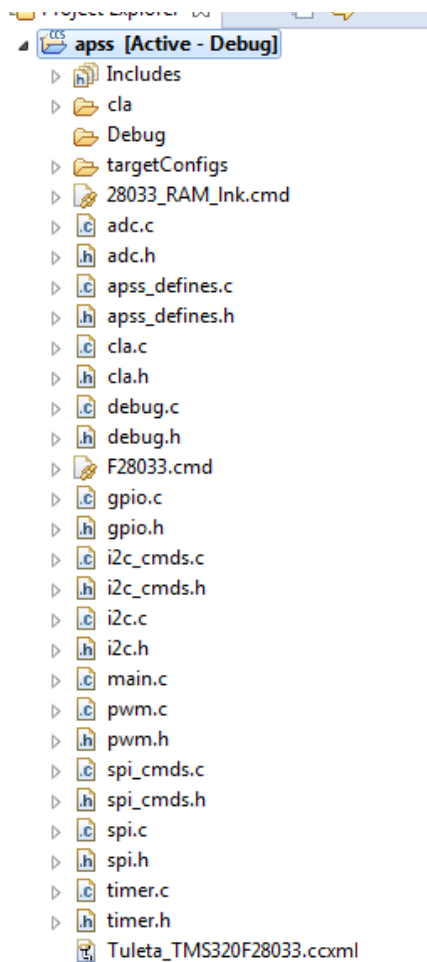


APSS build instructions:

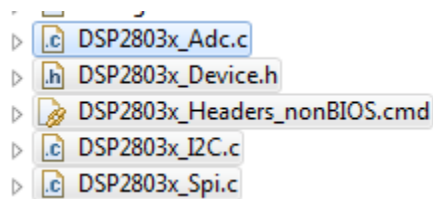
- Download and install ControlSuite software from TI : <http://www.ti.com/tool/ControlSUITE> , C2000 is the family of parts and TMS320F28033PAGT is the specific device of interest (<http://focus.ti.com/lit/ds/symlink/tms320f28033.pdf>)
- Download and install code composer studio from TI: <http://www.ti.com/tool/ccstudio> . The free version is sufficient for building this application.
- Start the new project wizard, use settings shown here:



- Clone the APSS repository to your newly created workspace project folder, <http://github.com/open-power/apss>



- Right click 28033_RAM_Lnk.cmd and 'exclude from build'
- The following files will need to be added to the project from the ControlSuite installation directory:



When prompted, select "copy files" to your workspace folder. Some of these files will need to be edited from the defaults. See below for the edits required on these files:

DSP2803x_Adc.c :

```

AdcRegs.ADCCTL1.bit.ADCPWDN    = 1;           // Power ADC
AdcRegs.ADCCTL1.bit.ADCENABLE = 1;           // Enable ADC
AdcRegs.ADCCTL1.bit.ADCREFSEL = 1;           // Select external BG
EDIS;

```

```
#define DSP28_28033PN TARGET
#define DSP28_28034PAG 0
#define DSP28_28034PN 0
#define DSP28_28035PAG 0
#define DSP28_28035PN 0
```

```
// GpioCtrlRegs.GPAPUD.bit.GPIO28 = 0;    // Enable pull-up for GPIO28 (SDAA)
// GpioCtrlRegs.GPAPUD.bit.GPIO29 = 0;    // Enable pull-up for GPIO29 (SCLA)

GpioCtrlRegs.GPBPUD.bit.GPIO32 = 0;    // Enable pull-up for GPIO32 (SDAA)
GpioCtrlRegs.GPBPUD.bit.GPIO33 = 0;    // Enable pull-up for GPIO33 (SCLA)

/* Set qualification for selected pins to asynch only */
// This will select asynch (no qualification) for the selected pins.
// Comment out other unwanted lines.

// GpioCtrlRegs.GPAQSEL2.bit.GPIO28 = 3; // Asynch input GPIO28 (SDAA)
// GpioCtrlRegs.GPAQSEL2.bit.GPIO29 = 3; // Asynch input GPIO29 (SCLA)

GpioCtrlRegs.GPBQSEL1.bit.GPIO32 = 3; // Asynch input GPIO32 (SDAA)
GpioCtrlRegs.GPBQSEL1.bit.GPIO33 = 3; // Asynch input GPIO33 (SCLA)

/* Configure I2C pins using GPIO regs*/
// This specifies which of the possible GPIO pins will be I2C functional pins.
// Comment out other unwanted lines.

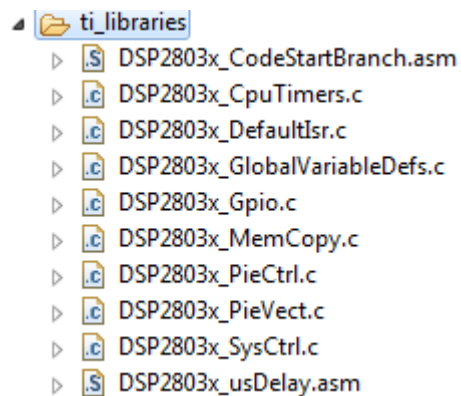
// GpioCtrlRegs.GPAMUX2.bit.GPIO28 = 2;    // Configure GPIO28 for SDAA operation
// GpioCtrlRegs.GPAMUX2.bit.GPIO29 = 2;    // Configure GPIO29 for SCLA operation

GpioCtrlRegs.GPBMUX1.bit.GPIO32 = 1;    // Configure GPIO32 for SDAA operation
GpioCtrlRegs.GPBMUX1.bit.GPIO33 = 1;    // Configure GPIO33 for SCLA operation
```

- Link to ControlSuite directories:
 - o Go to workspace properties
 - o Select Build -> C2000 Compiler -> Include Options
 - o Add search paths as shown below (use your own ControlSuite installation path). Also include a workspace path to the 'cla' subdirectory.

```
Add dir to #include search path (--include_path, -I)
"{CG_TOOL_ROOT}/include"
"C:\Files\workspace\workspace_ti\v126\DSP2803x_common\include"
"C:\Files\workspace\workspace_ti\v126\DSP2803x_headers\include"
"${workspace_loc}/${ProjName}/cl"}
```

- Create a 'ti_libraries' folder underneath the apss project. Add the files as shown below (these are located in ControlSuite installation directory, 'copy files' to local space):



- Your workspace project explorer should look something like this:

- Includes
 - apss/cla
 - C:/Files/workspace/workspace_ti/v12
 - C:/Files/workspace/workspace_ti/v12
 - C:/ti/ccsv6/tools/compiler/c2000_6.2
 - cla
 - Debug
 - targetConfigs
 - ti_libraries
 - DSP2803x_CodeStartBranch.asm
 - DSP2803x_CpuTimers.c
 - DSP2803x_DefaultIsr.c
 - DSP2803x_GlobalVariableDefs.c
 - DSP2803x_Gpio.c
 - DSP2803x_MemCopy.c
 - DSP2803x_PieCtrl.c
 - DSP2803x_PieVect.c
 - DSP2803x_SysCtrl.c
 - DSP2803x_usDelay.asm
 - adc.c
 - adc.h
 - apss_defines.c
 - apss_defines.h
 - cla.c
 - cla.h
 - debug.c
 - debug.h
 - DSP2803x_Adc.c
 - DSP2803x_Device.h
 - DSP2803x_Headers_nonBIOS.cmd
 - DSP2803x_I2C.c
 - DSP2803x_Spi.c
 - DSP28x_Project.h
 - F28033.cmd
 - gpio.c
 - gpio.h
 - i2c_cmds.c
 - i2c_cmds.h
 - i2c.c
 - i2c.h
 - main.c
 - pwm.c
 - pwm.h
 - spi_cmds.c
 - spi_cmds.h
 - spi.c
 - spi.h
 - timer.c
 - timer.h
 - 28033_RAM_Ink.cmd
 - Tuleta_TMS320F28033.ccxml

- A build should now be run with no errors. Less than 20 warnings is expected. A “release” build should be run before deploying to a production build. “Debug” builds will not perform as well.
- The .out file that is generated in the “Debug” or “Release” folder is used to load new code to a target machine via the JTAG interface. An XDS100V2 dongle is recommended for this. The .ccxml file is used for setting up the JTAG chain. The example one called “Tuleta_TMS320F28033.ccxml” is included but that will most likely need to be modified. The TI software tool called “UniFlash” can be downloaded for programming a target machine if that is preferred but the .ccxml file generated in code composer will need to be used with that tool.