

**Iowa State University**  
**Electrical and Computer Engineering**  
**E E 452. Electric Machines and Power Electronic Drives**

**Laboratory #2**  
**Introduction to the TI F28035 Microcontroller**

### Summary

This lab provides an introduction to the F28035 Experimenter Kit, Code Composer Studio, and MATLAB/Simulink Embedded Coder, by executing a simple program to blink an LED. The TI F28035 Experimenter Kit will be used (later on) to control the power electronic devices in the lab. The Code Composer Studio IDE is used to generate an executable file and program the micro-controller unit (MCU). Embedded Coder is a toolbox in MATLAB/Simulink, which provides a graphical method of generating C code for the MCU.

### Learning objectives

- Learn general features and capabilities of the F28035 MCU.
- Generate, build, and debug a project in Code Composer Studio (CCS).
- Use Embedded Coder to generate a program for the F28035 MCU.

### Background material (Read before coming to the lab)

- *Piccolo Experimenter Kit Quick Start Guide (QSG-ExperimentersKit.pdf)*
- TMS320F28035 Datasheet (tms320f28035.pdf) – *Identify main sections of this document and be able to navigate it. No need to read everything in detail.*
- *Getting Started with Code Composer Studio v5:*  
**[http://processors.wiki.ti.com/index.php/C2000\\_Getting\\_Started\\_with\\_Code\\_Compiler\\_Studio\\_v5](http://processors.wiki.ti.com/index.php/C2000_Getting_Started_with_Code_Compiler_Studio_v5)**; mainly study section 6.
- *Embedded Coder User's Guide (Embedded\_Coder\_User\_Guides.pdf)* – Study section 36, *Working with Texas Instruments Code Composer Studio 4 IDE.*

### Exercises and Questions

*Instructions: every student should deliver his/her own report at the end of the lab session, even though the experiments are conducted in groups. You may want to answer the questions as you go along the exercises. Time yourselves according to the recommendations below.*

## 1. Pre-lab assignment

Study the material provided with this lab to familiarize yourself with the hardware configuration, and answer the following questions:

DELIVERABLE 1: What is the system clock frequency of the Piccolo F28035 MCU?

DELIVERABLE 2: Is the processor 16, 32, or 64-Bit?

DELIVERABLE 3: What system control voltage is used for the F28035? 5V or 3.3V?

DELIVERABLE 4: What pin on the TMS320F28035 device and what GPIO pin on the ControlCARD are connected to the ControlCARD's Red LED (LD3)?

DELIVERABLE 5: How many PWM channels are on the F28035? What is the GPIO pin designation of these outputs on the Docking Station? (The answer is in *F28035controlCARD-PinOutTable.pdf*)

For information regarding applications, visit [www.ti.com/apps](http://www.ti.com/apps).

## 2. Piccolo F28035 Experimenter Board. [30 minutes]

*This section requires use of the Experimenter Kit schematics (USBDockingStation\_Schematic [R3].pdf), and the TMDSF28035 datasheet (tms320f28035.pdf). Use this lab time to learn the capabilities and features of this control platform; we will use it to control power electronic devices in future labs.*

Open ControlSUITE from the Start Menu, or navigate to C:\TI\controlSUITE\controlSUITE.exe. ControlSUITE is your one-stop-shop for everything C2000. Here, you can find the Piccolo F2803x Datasheet, as well as many other useful documents. Take some time to browse ControlSUITE; there are many useful resources here.

Navigate through ControlSUITE to find the *Experimenter's Kit Quick Start Guide (QSG) DIM100*, for your F28035 ControlCARD. Go through this document to setup the hardware (section 1). Skip the software setup for Control Suite (section 2) at this time; it has already been installed. Skip the installation of drivers (section 3). *Running Your First Program* (section 4) will be performed in the next section of this lab.

### 3. Code Composer Studio v5 [60 minutes]

*Go through the download process several times to become proficient. You will do this a lot in the future. For a more detailed guide, see “How to use Code Composer Studio v5 to program a TI device” in the supporting documents for this lab, or consult the CCS Help menu.*

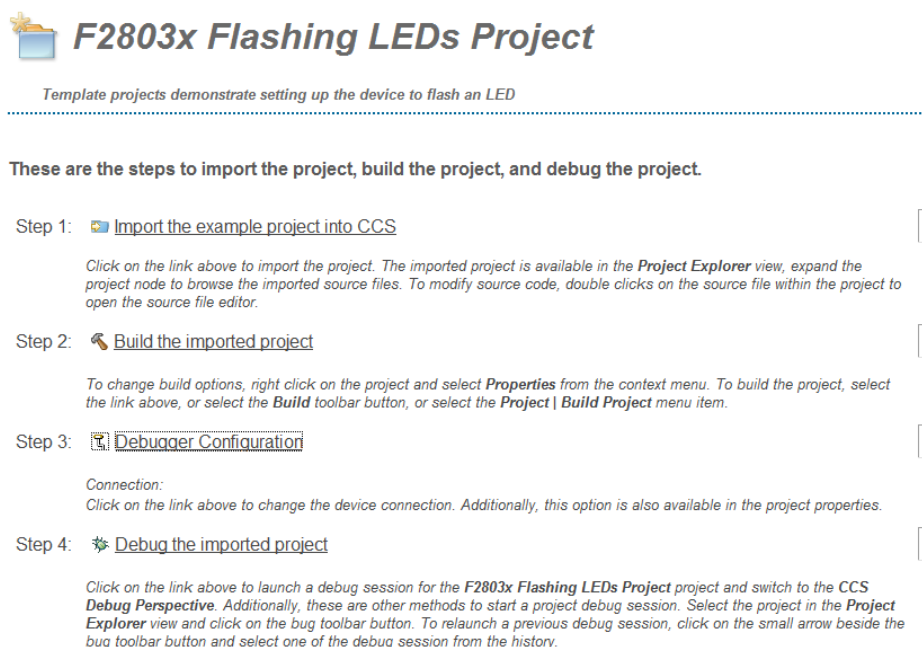
ControlSUITE provides an example project, which simply blinks an LED. We will access this project through CCSv5. Alternatively, project files can be copied via ControlSUITE.

Navigate to C:\ticcsv5 and open Code Composer Studio v5. When prompted, set the Workspace path to C:\WorkspaceCCSv5\yourfolder. This is a shared location on the local machine, with Read/Write privileges enabled for any user; it should be used for all of your CCS work. When you are finished with this lab, transfer the contents of this folder to a location of your choice for safe keeping. DO NOT check the box to make this the default location. If prompted to specify a license, click “Activate” and enter:

License Server Location: io.ece.iastate.edu  
Port: 27003

Code Composer Studio should read “Licensed” in the bottom left corner.

Go to Help/Welcome to CCS. On the Welcome screen, click on “Examples”. This will open a navigation bar on the left side of the screen. Scroll down to find the Flashing LED example for the F28035 ControlCARD. Follow the steps outlined there to import, build, and download the example project. You should see the following set of instructions:




**F2803x Flashing LEDs Project**


Template projects demonstrate setting up the device to flash an LED

---


These are the steps to import the project, build the project, and debug the project.

Step 1:  [Import the example project into CCS](#)


Click on the link above to import the project. The imported project is available in the **Project Explorer** view, expand the project node to browse the imported source files. To modify source code, double clicks on the source file within the project to open the source file editor.

Step 2:  [Build the imported project](#)

To change build options, right click on the project and select **Properties** from the context menu. To build the project, select the link above, or select the **Build** toolbar button, or select the **Project | Build Project** menu item.

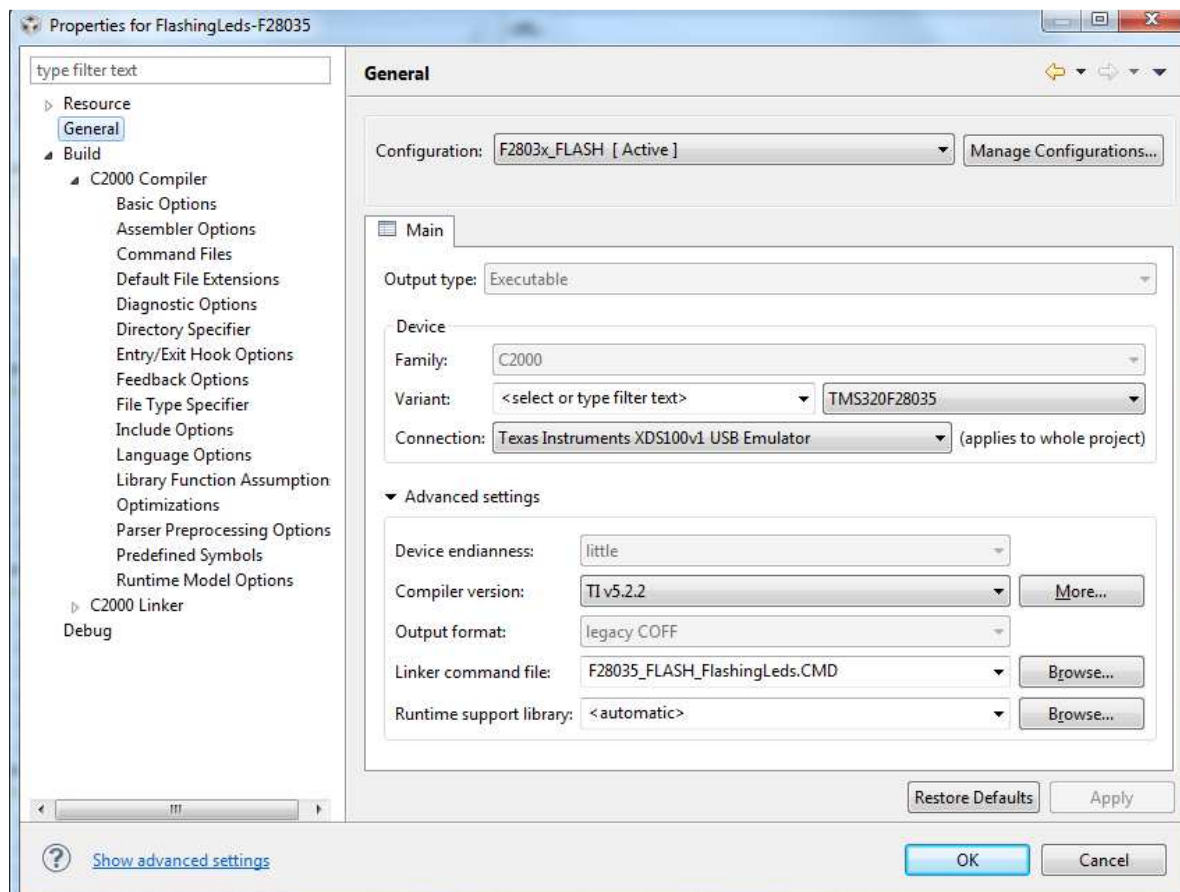
Step 3:  [Debugger Configuration](#)

Connection:  
Click on the link above to change the device connection. Additionally, this option is also available in the project properties.

Step 4:  [Debug the imported project](#)

Click on the link above to launch a debug session for the **F2803x Flashing LEDs Project** project and switch to the **CCS Debug Perspective**. Additionally, these are other methods to start a project debug session. Select the project in the **Project Explorer** view and click on the bug toolbar button. To relaunch a previous debug session, click on the small arrow beside the bug toolbar button and select one of the debug session from the history.

Before building the project, highlight the FlashingLED project in the Project Explorer, then go to Project/Properties, which is shown below, to adjust the project options. On the General tab, select your *Device Variant* as *Experimenter's Kit - Piccolo F28035*, and set the *Connection* to *Texas Instruments XDS100v1 USB Emulator*. Also, ensure that the *Compiler Version* is set to *TI v5.2.2*. The *Linker command file* should be set as *F28035\_FLASH\_FlashingLeds.CMD*. If it is not, browse to the example project location *C:\TI\controlSUITE\development\_kits\TemplateProjects\FlashingLeds-F28035\_v1.0*, and select the FLASH command file. Also set the *Runtime support library* to *<automatic>*. Click "OK".



Build the project according to Step 2 of the example project menu.. Note the build progress shown in the Console. When done, you should read "*Finished building target: FlashingLeds.out*".

Configure the debugger according to Step 3 of the example project menu.

Connect the target to the computer via the supplied USB cable. Turn on the board. Debug the project according to Step 4 of the example project menu. CCS will connect to the target, erase, program, and verify; be patient. Once the program is loaded, your

device will appear in the Debug window, and will show a status of “Suspended”. Click the “Resume” button, or go to Run/Continue. The red LED on the ControlCARD should be blinking, and the status should be “Running”. Roam around the various options and familiarize yourself with the debug environment. Go through the download and reload processes multiple times.

Before disconnecting the target, or terminating the debug session, be sure to reset the CPU. DO NOT PUT A PASSWORD ON THE PROGRAM YOU ARE LOADING; YOU COULD PERMANENTLY LOCK THE DEVICE!

Additionally, you may go to Help/Code Composer Help/Getting Started Quickly. Go through the subsections to further familiarize yourself with the CCS environment.

#### **4. Embedded Coder and the TI F28035 Toolbox. [75 minutes]**

*Explore the Embedded Coder Toolbox. Go through the project build process several times to become proficient; we will be doing this a lot in future labs. For a more detailed guide, see “Installation, Setup, and Program Generation using ControlSUITE, CCSv5, and Embedded Coder” provided in the supporting documents for this lab.*

Open MATLAB. Set the *Current Folder*, on the top toolbar, to C:\WorkspaceMATLAB\yourfolder. If you have the folder set as a network location, you will encounter errors during the build process. Read through the Help information for Embedded Coder. In Particular, review the sections in Help/Embedded Coder/User’s Guide/Embedded IDEs and Embedded Targets/Working with Texas Instruments Code Composer Studio 4 IDE.

Explore other topics within Embedded IDEs and Embedded Targets/ Project and Build Configurations for Embedded Targets, and answer the following question:

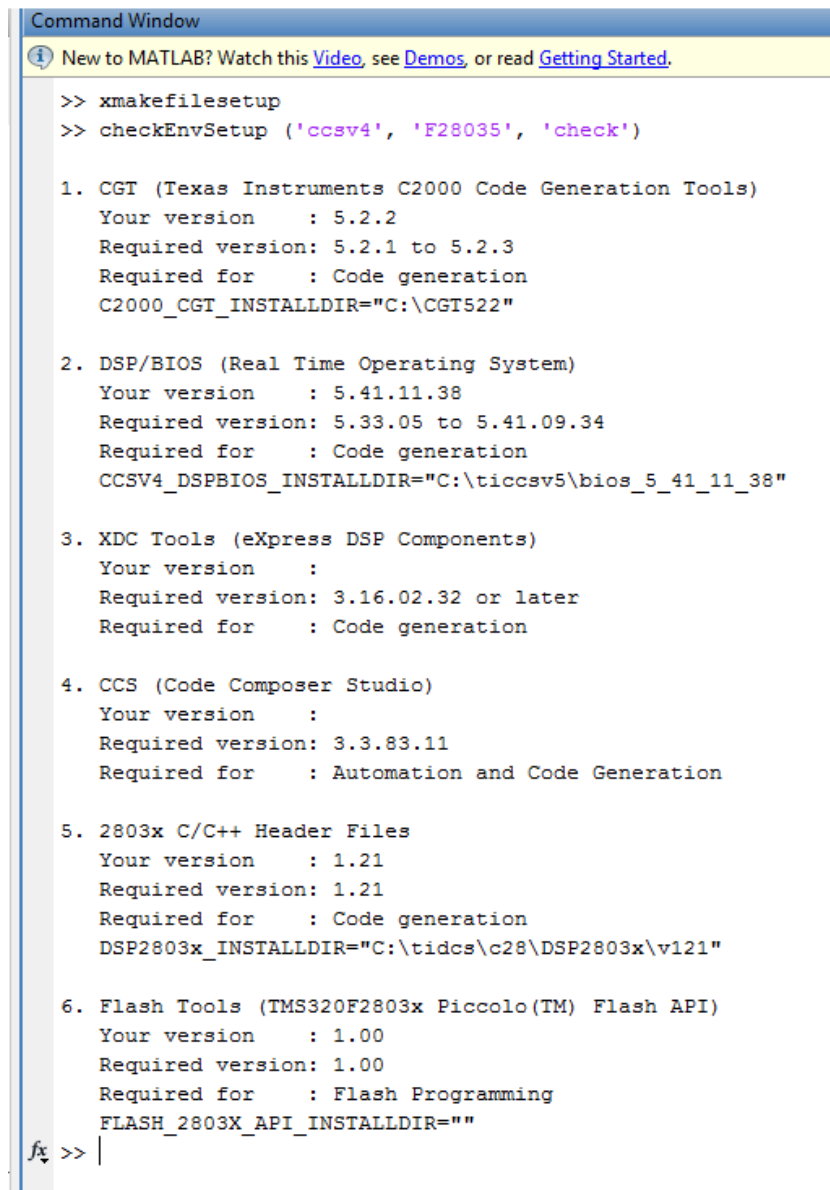
DELIVERABLE 6: What Simulink blocks cannot be used to generate code for Embedded Systems?

For more information on code generation capabilities of specific blocks, see Help/Simulink Coder/User’s Guide/Model Architecture and Design/Modeling/Supported Products and Block Usage.

After reviewing the *Getting Started* section and browsing through *Tutorial: Using Makefiles with Code Composer Studio 4.x* perform the following:

a) Verify the Code Composer Installation:

Type the command “checkEnvSetup (‘ccsv4’, ‘f28035’, ‘check’)”. Ensure that the Code Generation Tool (compiler) is within the range specified, and that the Header Files match those specified. MATLAB should return the following screen:



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> xmakefilesetup
>> checkEnvSetup ('ccsv4', 'F28035', 'check')

1. CGT (Texas Instruments C2000 Code Generation Tools)
   Your version      : 5.2.2
   Required version: 5.2.1 to 5.2.3
   Required for      : Code generation
   C2000_CGT_INSTALLDIR="C:\CGT522"

2. DSP/BIOS (Real Time Operating System)
   Your version      : 5.41.11.38
   Required version: 5.33.05 to 5.41.09.34
   Required for      : Code generation
   CCSV4_DSPBIOS_INSTALLDIR="C:\ticcsv5\bios_5_41_11_38"

3. XDC Tools (eXpress DSP Components)
   Your version      :
   Required version: 3.16.02.32 or later
   Required for      : Code generation

4. CCS (Code Composer Studio)
   Your version      :
   Required version: 3.3.83.11
   Required for      : Automation and Code Generation

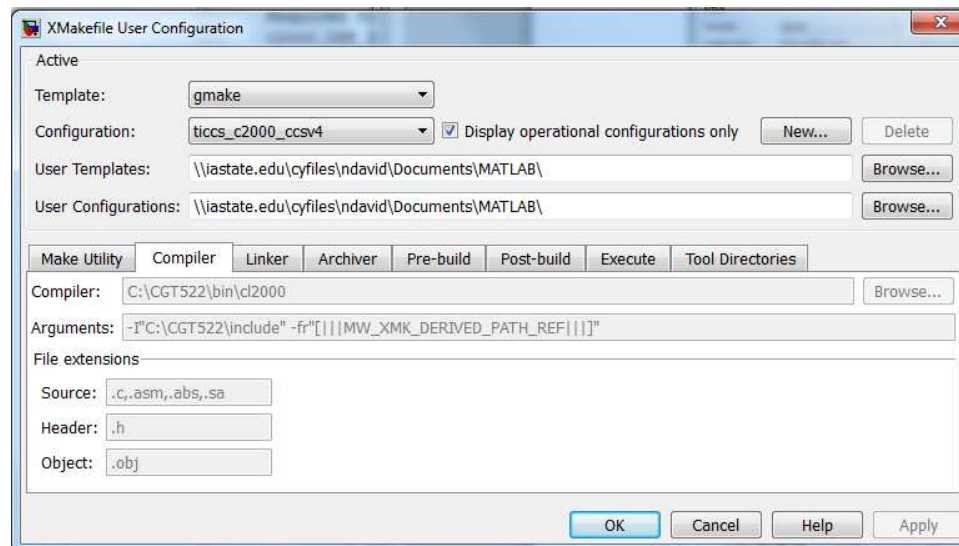
5. 2803x C/C++ Header Files
   Your version      : 1.21
   Required version: 1.21
   Required for      : Code generation
   DSP2803x_INSTALLDIR="C:\tidcs\c28\DSP2803x\v121"

6. Flash Tools (TMS320F2803x Piccolo(TM) Flash API)
   Your version      : 1.00
   Required version: 1.00
   Required for      : Flash Programming
   FLASH_2803X_API_INSTALLDIR=""

fx >> |
```

b) Setup the XMakeFile Utility:

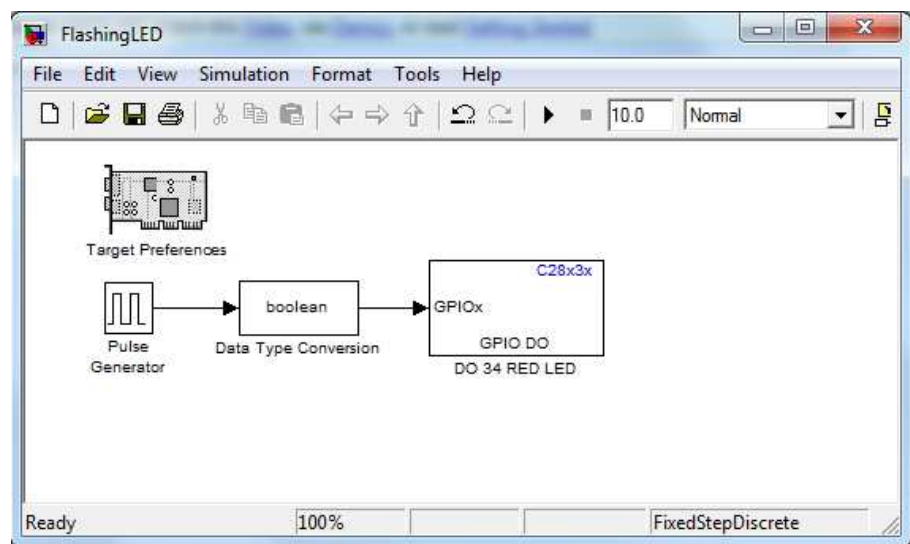
Enter the command “xmakefilesetup”. The xMakefile Utility will open after a few moments, as shown below.



Select the *Configuration* to *ticcs\_C2000\_ccsv4*. Click “Apply”. If prompted to specify a Code Composer Installation directory, point to the location of C:\ticcsv5\ccsv5. Click on the *Tool Directories* tab, and set the *Code Generation Tool* by browsing to C:\CGT522. Click “Apply”. Check that the compiler, linker, and archiver tool directories are specified and correct. They should be set to use tools located within the folder for compiler version 5.2.2.

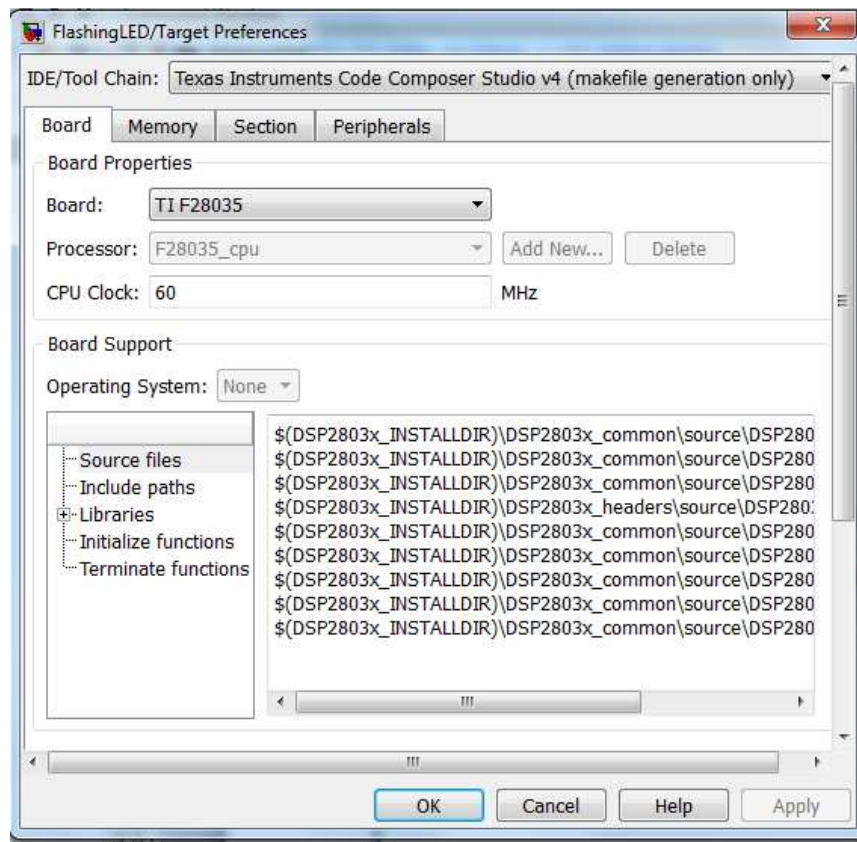
c) Build a Simulink Model to Generate Code:

Build the Simulink block diagram shown below



Inside the *Target Preferences* block, select the *IDE/ Tool Chain* as *CCSv4*, and select the proper board, as shown here:





Configure the Digital Output for the GPIO of the red LED on the controlCARD; you found this pin location earlier.

Set the Source Pulse Generator to output a square wave with period 1s, and duty cycle of 75%.

Go to Simulation/Configuration Parameters. On the Solver tab, verify that the solver is set to *discrete*. On the Hardware Implementation tab, verify that *Texas Instruments*, *C2000*, and *Little Endian* are selected. On the Code Generation tab, ensure the *System target file* is "idmlink\_ert.tlc". Code Generation objectives, such as *Execution efficiency* and *ROM efficiency* can also be specified on this tab.

Within Code Generation, go to the Report tab, and check to *Create a code generation report*. In the IDE Link tab, change the *Build action* to *Build*. Also, verify the *Configuration* is *Custom*. Apply the changes and click OK.

In your model, go to Tools/Code Generation/Build Model to build the project. Watch the command window for progress. Examine the code generation report.

DELIVERABLE 7: How many lines of code are in the file FlashingLED\_main.c? How about the model file FlashingLED.c?



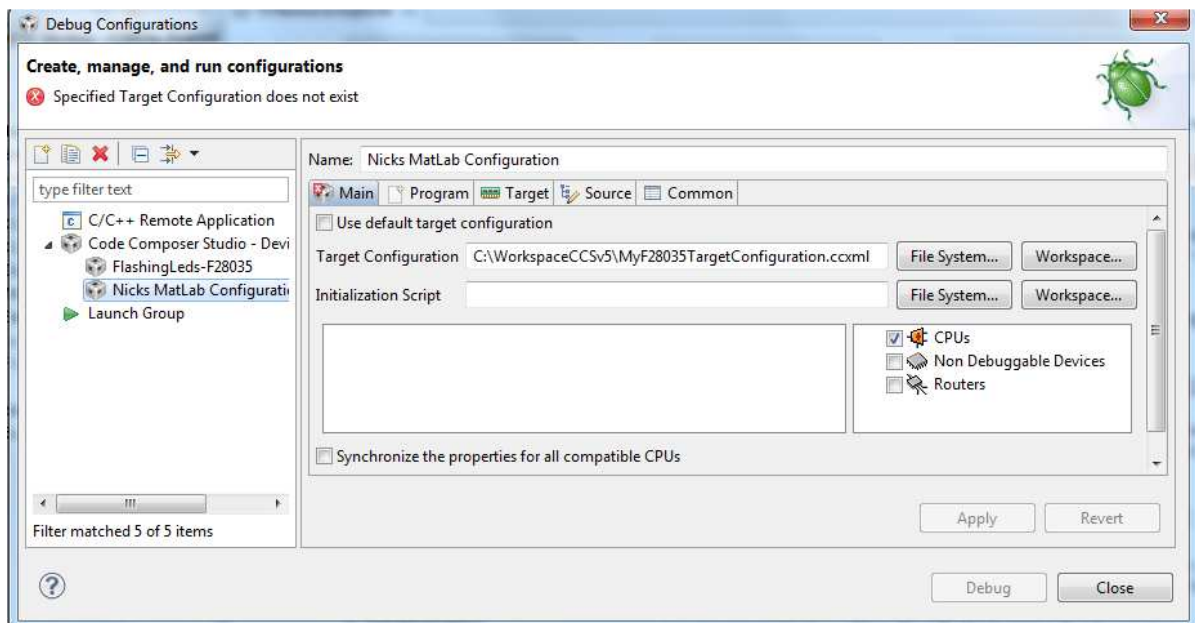
When the build is complete, you will see a new folder created in the location specified in the *Current Folder* called "projectName\_ticcs". This folder has all files created for use in Code Composer, including a "projectName.out" executable which will be downloaded to the board. Verify this file exists.

d) Load the Project to the Target Device:

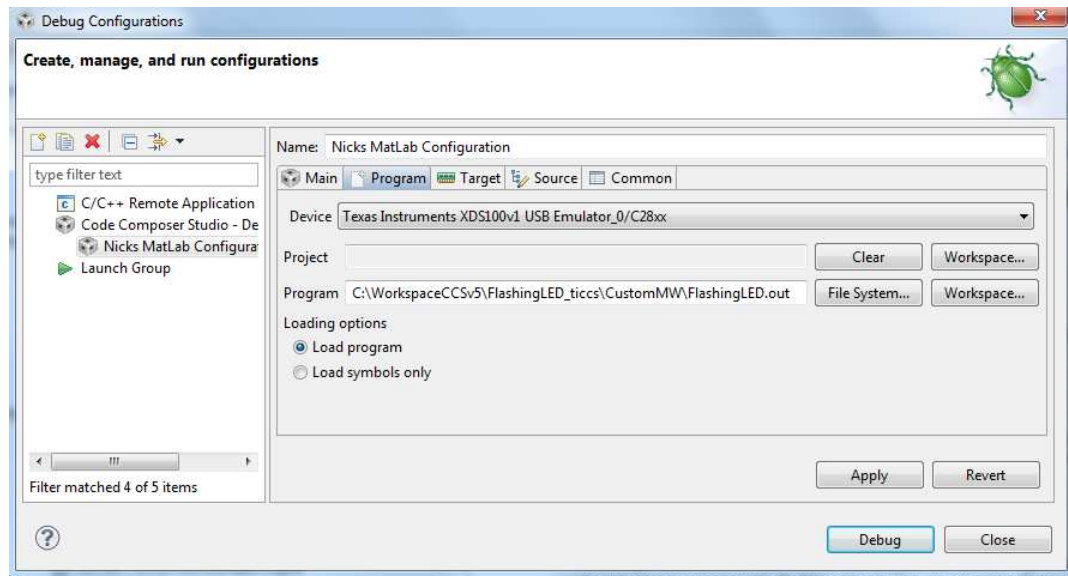
Open Code Composer Studio and select your workspace.

Create a new Target Configuration File, and save it to C:\WorkspaceCCSv5\yourfolder. Select the appropriate device and connection type (Experimenter's Kit - Piccolo F28035, XDS100v1 USB). Turn on the board and click to *Test Connection*.

Go to Run/Debug Configurations, shown below. Make a new Launch Configuration under the *Code Composer Studio – Device Debugging* section. Search the *File System* to select the target configuration file you just created.



On the *Program* tab, browse to the location of the \*.out file created during the Embedded Coder build process.



On the Target tab, scroll down and check *Connect to Target on Debugger Startup*. Click “Apply” and verify there are no warnings. Then click “Debug”. The device is erased and programmed with the specified file. When the download is done, the device will be “Suspended”. Click “Resume” (play button) to run the program.

DELIVERABLE 8: What duty cycle do you observe for the ON state of the LED? Why?

e) Reset the CPU and Disconnect the Target:

**WARNING:** When you “Suspend” (pause button) the target, the program stops at the command it is currently executing; the current state of any outputs will be latched. **DO NOT “SUSPEND/PAUSE” THE TARGET WHILE DRIVING ENERGIZED POWER SWITCHES!** Use “Reset” to halt the device; outputs will latch to their default state, and the program will stop running.

## 5. Conclusion [15 minutes]

Write about one or two things you learned in this lab that you think are important or interesting, and why.