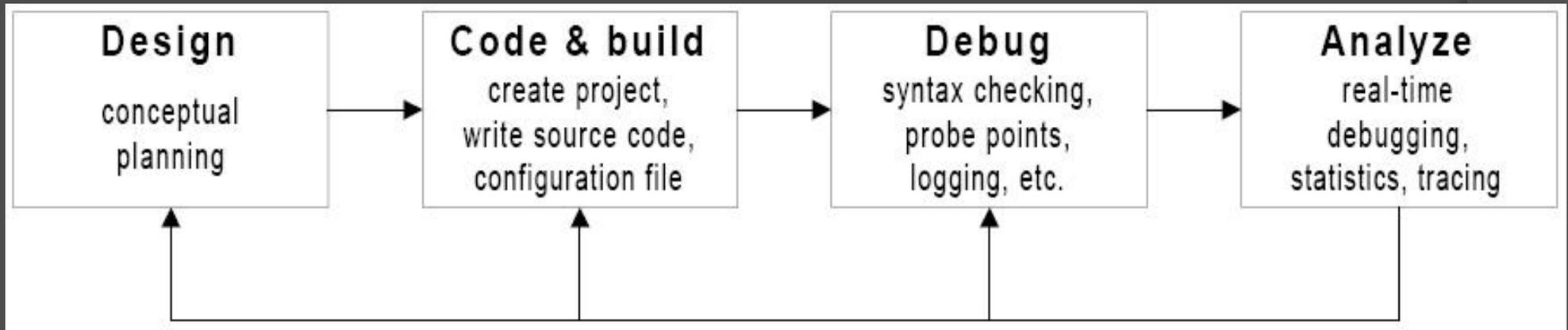


D. Richard Brown III
Associate Professor
Worcester Polytechnic Institute
Electrical and Computer Engineering Department
drb@ece.wpi.edu

ECE4703 REAL-TIME DSP LAB SOFTWARE OVERVIEW



Code Composer Studio IDE



Note that we will be using CCS v5.

CCS v4/v5 is based on the Eclipse IDE

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

Code Composer Studio IDE

- Connect USB cable from PC to DSK
- Connect power supply to DSK
- Wait for POST to complete
 - If this is the first time connecting the DSK, Windows may install a driver. This should happen automatically.
- Launch Code Composer Studio v5
- CCS will load and wait for your input



Code Composer Studio IDE

The screenshot displays the Code Composer Studio (CCS) IDE interface. The main window shows a C source file named `hello.c` with the following code:

```
1 #include <stdio.h>
2
3 /*
4  * hello.c
5  */
6 int main(void) {
7     printf("Hello World!\n");
8
9     return 0;
10 }
11
```

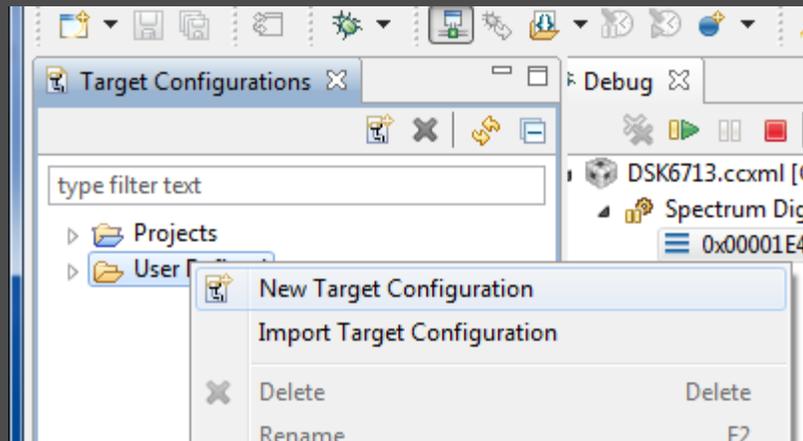
The interface includes a Project Explorer on the left showing the project structure with `helloworld` (Active - Debug) and `stereoloopi`. The Target Configuration window on the right shows a filter for "type filter text" and a list of projects, including "User Defined" and "DSK6713.ccxml". The Console window at the bottom left shows the output: "TMS320C671X: GEL Output: GEL StartUp Complete." The Problems window at the bottom right shows "0 items". The status bar at the bottom indicates the IDE is "Licensed", "LE", "Writable", "Smart Insert", and "1:1".

CCS Workspace

- You should probably not keep your CCS files on the local computer
- Recommended workspace path:
M:\ECE4703\labN
where N is the current lab number
- Each part of the project will then be in a subpath like
M:\ECE4703\lab I \part I \
- **Note: CCS may not work with long drive names, e.g. <\\ece-homes.ece.wpi.edu>. You should use M:**

CCS v5 Initial Configuration

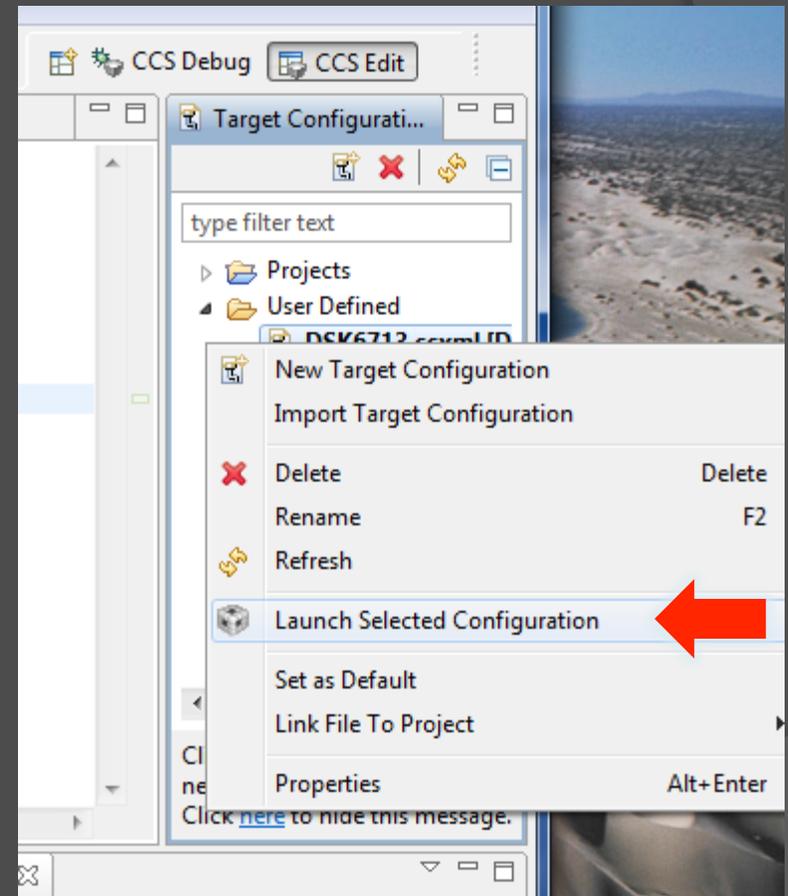
- If this is the first time CCS v5 is run, you will need to set up target configuration for C6713DSK
- Window -> Show View -> Target Configurations
- Create new target configuration (right click):



- Details for setting up a DSK6713 target here:
[http://spinlab.wpi.edu/courses/ece4703/
configureccsv5.html](http://spinlab.wpi.edu/courses/ece4703/configureccsv5.html)

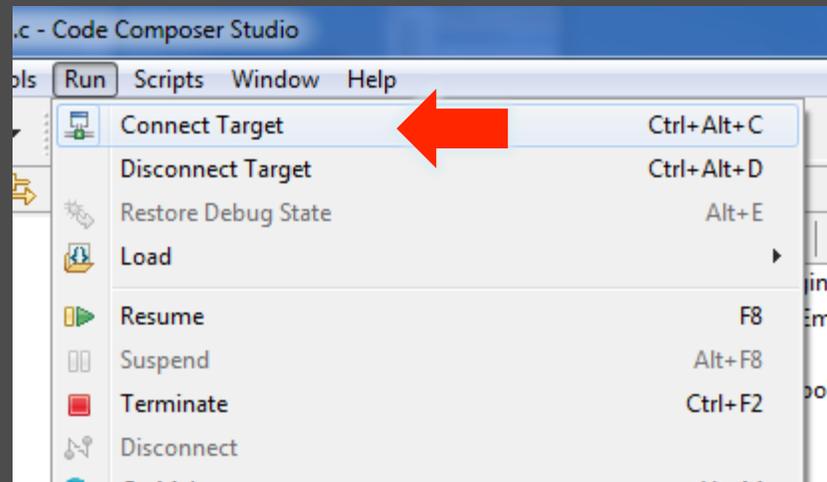
Launching Target Configuration

- Window -> Show View -> Target Configurations
- Right click on your DSK6713.ccxml target configuration
- Launch selected configuration

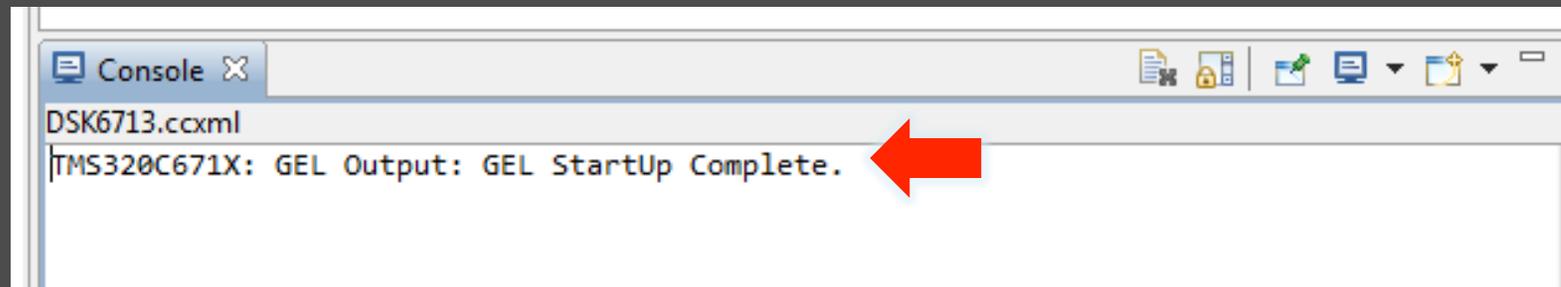


Connecting to the C6713 DSK

- Run -> Connect Target (or Ctrl+Alt+C)



- If successful, you should see this in the console



Goals for Today

- Get familiar with DSK and lab hardware
- Get familiar with CCS v5
- Get Helloworld project working (Lab 1, part 1)
<http://spinlab.wpi.edu/courses/ece4703/helloworld.html>
- Get Stereoloop project working (Lab 1, part 2)
<http://spinlab.wpi.edu/courses/ece4703/stereoloop.html>
 - This will probably require downloading some chip support and board support libraries: <http://spinlab.wpi.edu/courses/ece4703/cslbsl.html>
- Start part 3 of the lab 1 assignment

Does your Stereoloop project work?

- ⦿ Try playing some music into the line input of the DSK.
- ⦿ Plug headphones in the headphone output of the DSK.
- ⦿ If your code is running correctly, you should hear the music in the headphones.
- ⦿ Suspend your code. The music should stop.
- ⦿ This code simply reads in samples from the line input jack and outputs them (unmodified) to the line output and headphone jacks.
- ⦿ This code doesn't actually do any signal processing, but it will serve as a template for most of the DSP programs you will write in ECE4703.