

Application notes

ECE480 Design Team3

Yuan Mei



Content of the Application notes

1. Introduction
2. Power Set-up
3. Quick-start Application
4. Code Composer Studio (CCS)
5. Example
6. Conclusion

• Introduction

Stellaris LM3S3748 Evaluation Kit

The Stellaris LM3S3748 Evaluation Kit provides a low-cost way to start designing applications with Stellaris microcontrollers on a compact and versatile evaluation platform.

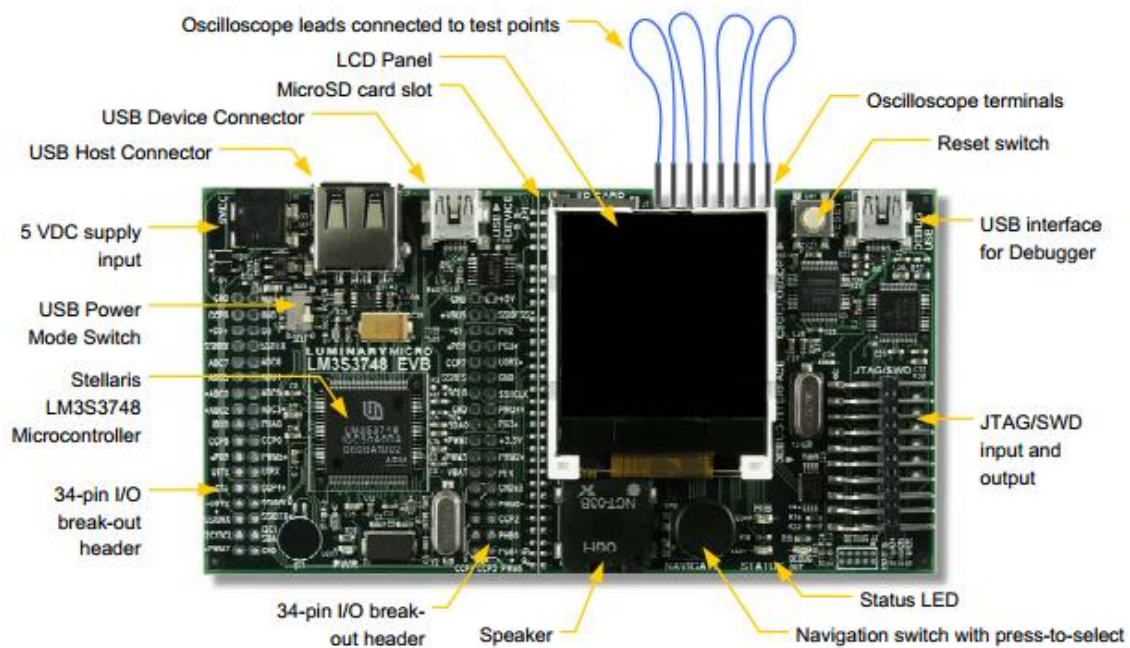


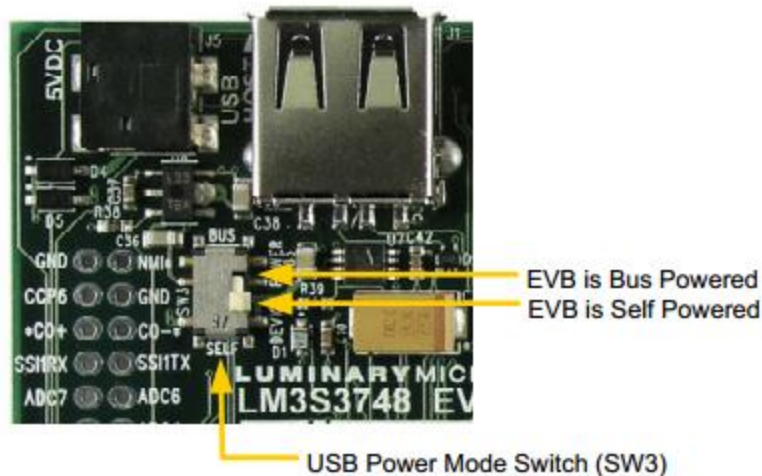
Figure1: Stellaris LM3S3748 Evaluation Kit

• Board Set-up

1. Powering the EVB

The first step for the initial board set-up is to power the EVB

- 1) Move the USB power mode switch (SW3) to the “SELF” position, like show below



2) Using one of the USB cables provided in the kit, connect the mini-b(smaller) end of the USB cable to the USB debug interface connector labeled “DEBUG USB” on the EVB.

3) Connect the other end (Type A) of the USB cable to a free USB port on your PC

2. Installing the FTDI Drivers

The next step is to install the FTDI drivers for the board.

When you power the EVB using the “DEBUG USB” connector for the first time, windows starts the Found New Hardware Wizard and asks if Windows can connect to Window Update to search for software. Select” No, not this time”and then click Next.



2) Next, select “Install from a list or specific location(Advanced)” and click Next.



- 3) Select “Search for the best driver in these locations,” and check the “Search removable media (floppy, CD-ROM...)” option. Click Next.

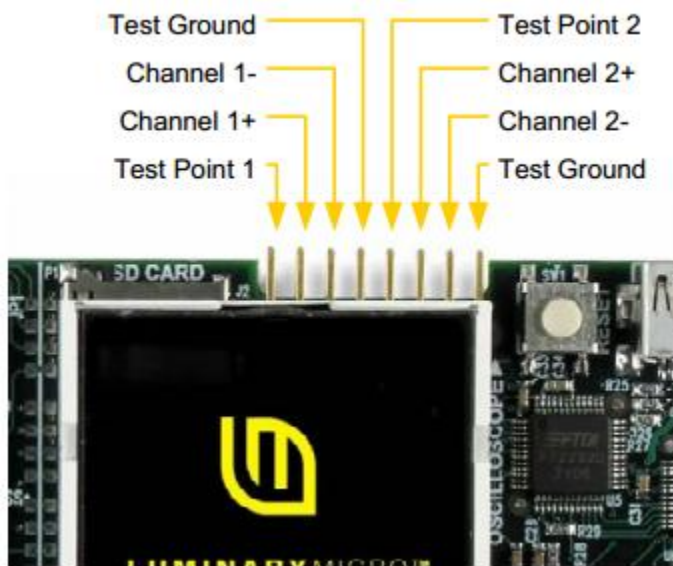


- 4) Windows finishes installing the drivers for “Stellaris Evaluation Board A.” When the driver install is finished, the Found New Hardware Wizard window appears like the one below. Click Finish to close the dialog box.

• Quick Start Application

The LM3S3748 Evaluation Board comes preprogrammed with a quick start application. Once you have powered the board, this application runs automatically. You have probably already noticed this running as you installed the drivers. A splash screen appears on the LCD for a few seconds before the application begins.

The quickstart application provides a simple two channel oscilloscope sampling at up to 1M samples per second. The two oscilloscope channels are differential measurement channels which provide waveform acquisition using the LM3S3748 microcontroller's Analog-to-Digital Converter (ADC). The evaluation board includes an oscilloscope header that contains the two channel differential inputs, two test point pins, and two test ground pins.



Test Point 1 is connected to the speaker input on the EVB and allows the signal for the keyboard click to be viewed. Note that waveform capture is typically not taking place while the keyboard is being serviced so the click may not be seen on the waveform display for every keypress.

Test Point 2 is not used in our project.

After set-up the Stellaris, we need to focus on the software part. Stellaris Development and Evaluation Kits provide a low-cost way to start designing with Stellaris microcontrollers using Texas Instruments' Code Composer Studio development tools. The evaluation boards can function as either a complete evaluation target or as a debugger interface to any external Stellaris device.

• Code Composer Studio

Step 1: Install Code Composer Studio

Follow the instructions in the Code Composer Studio installation program. Select to install the Platinum Edition or the Microcontroller Edition with the Stellaris CortexM3 MCUs ISA selected at a minimum.

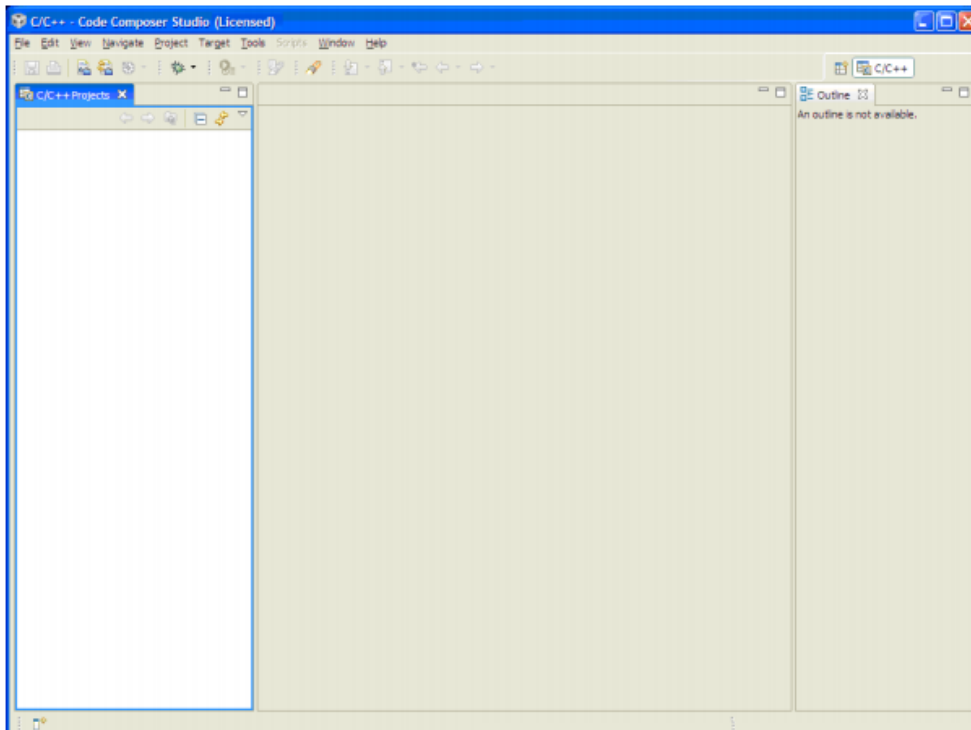
Step 2: Install the StellarisWare Package

A full set of C-based peripheral drivers is provided, covering all peripherals and functionality of the Stellaris devices. The StellarisWare package includes various example applications with project files for all major tool vendors that support Stellaris, including Code Composer Studio

Click on the 'Install' link next in the StellarisWare section of the DVD and run the StellarisWare installer

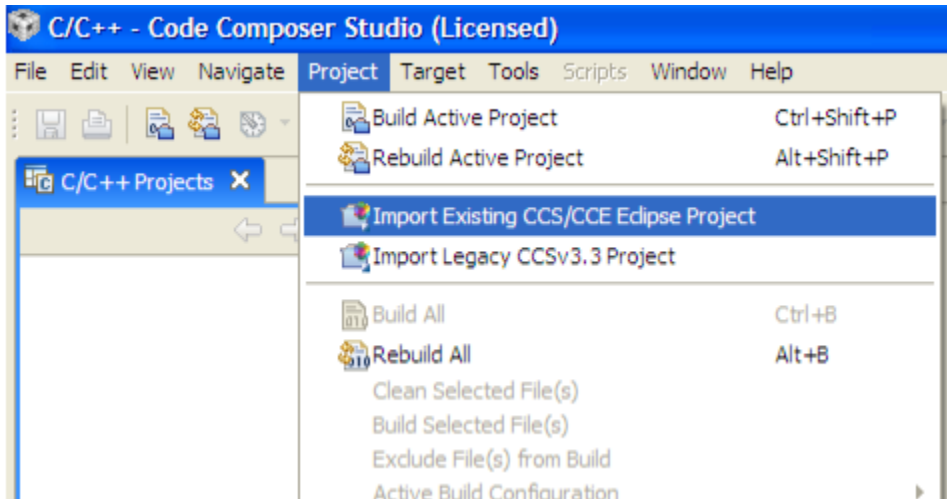
Step 3: Start Code Composer Studio and Open a workspace

- 1) Start the Code Composer Studio IDE by selecting it from the Windows Start menu or double-clicking the icon installed on your desktop.
- 2) Click OK to use this default workspace location if the Workspace launcher asks you to set the path
- 3) The Code Composer Studio IDE may now open with the welcome page. If so, close out the welcome page by clicking the 'X' on the tab. You should now have an empty workspace.

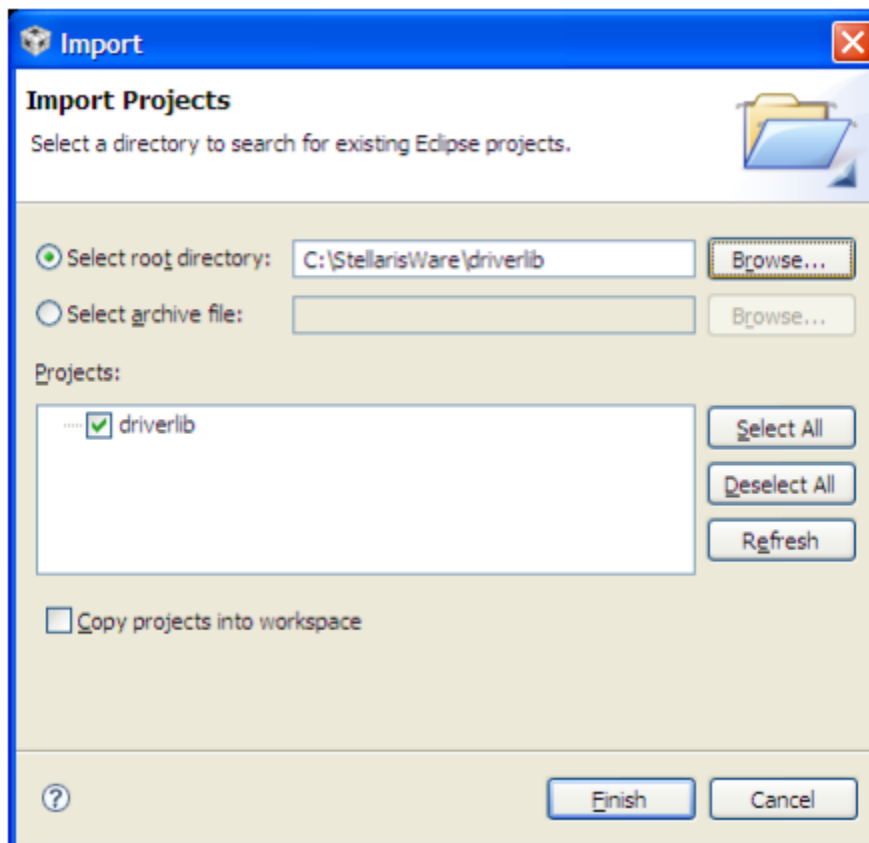


Step 4: Import Libraries

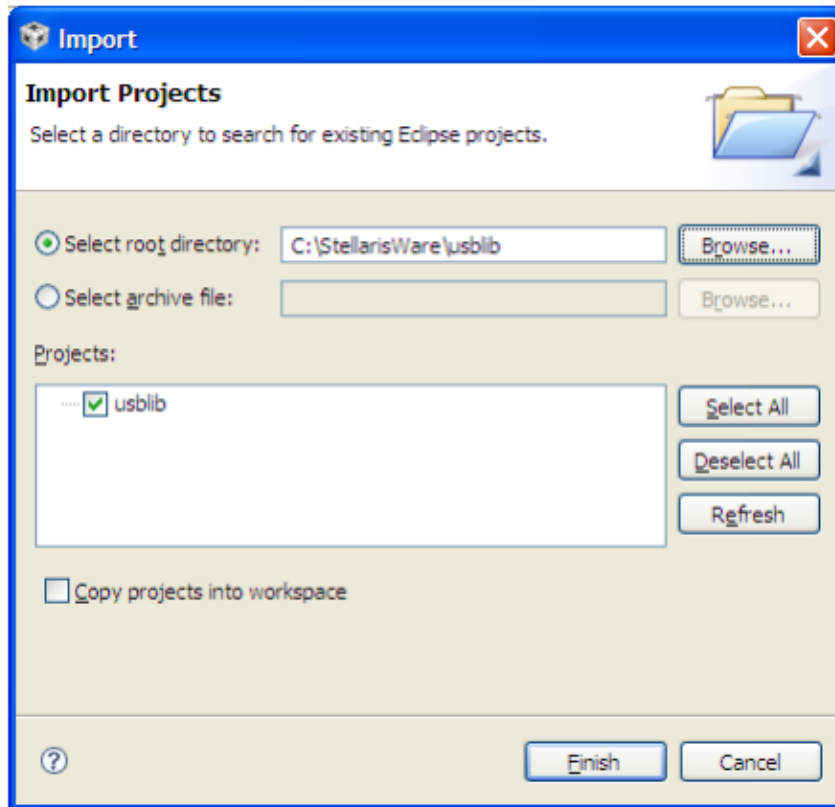
- 1) Select the “Import Existing CCS/CCE Eclipse Project” option from the Project menu in the IDE.



- 2) The “Import” dialog box appears. Browse to the root directory of driver library (StellarisWare\driverlib). Be sure that “driverlib” is selected in the listed projects and click “Finish.”



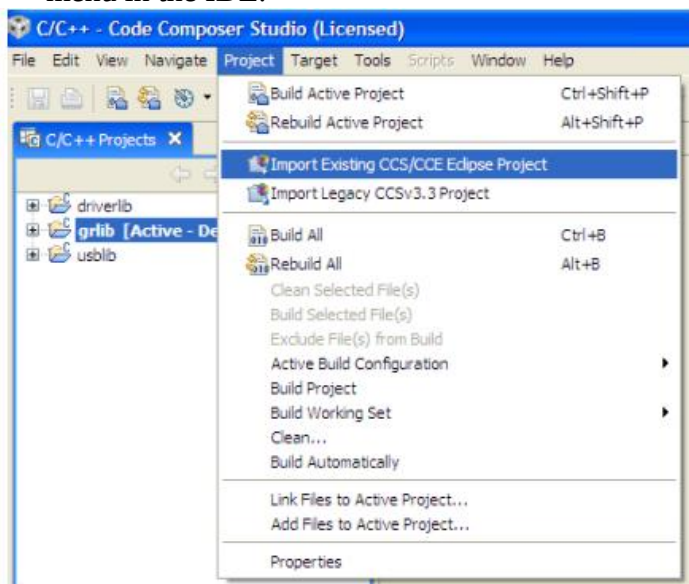
- 3) Select the “Import Existing CCS/CCE Eclipse Project” option from the Project menu in the IDE again. Browse to the root directory of USB library (StellarisWare\usblib). Be sure that “usblib” is selected in the projects list and click “Finish.”



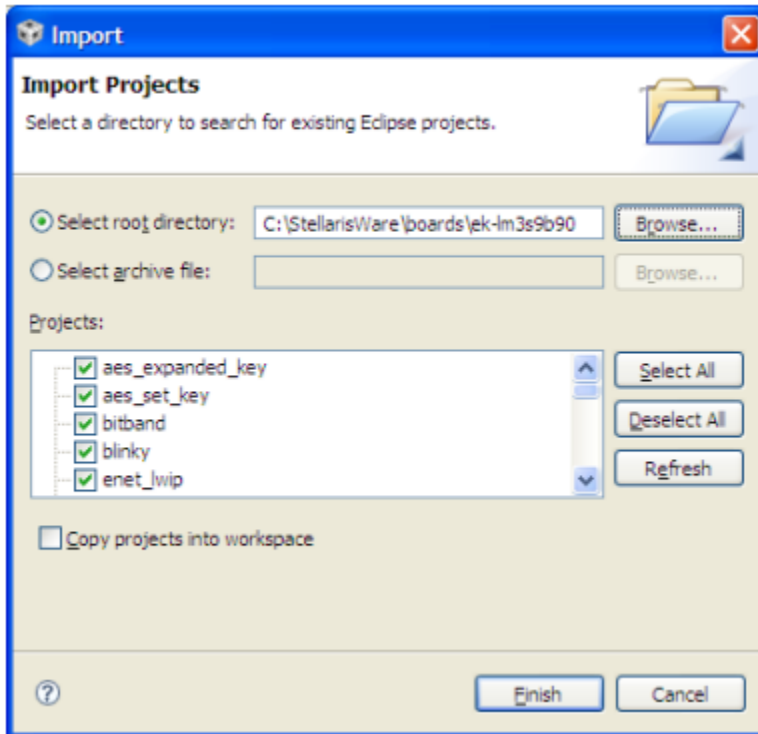
- 4) Select the “Import Existing CCS/CCE Eclipse Project” option from the Project menu in the IDE again. Browse to the root directory of the graphics library (StellarisWare\grlib). Be sure that “grlib” is selected in the projects list and click “Finish.”

Step 5: Import Board Examples

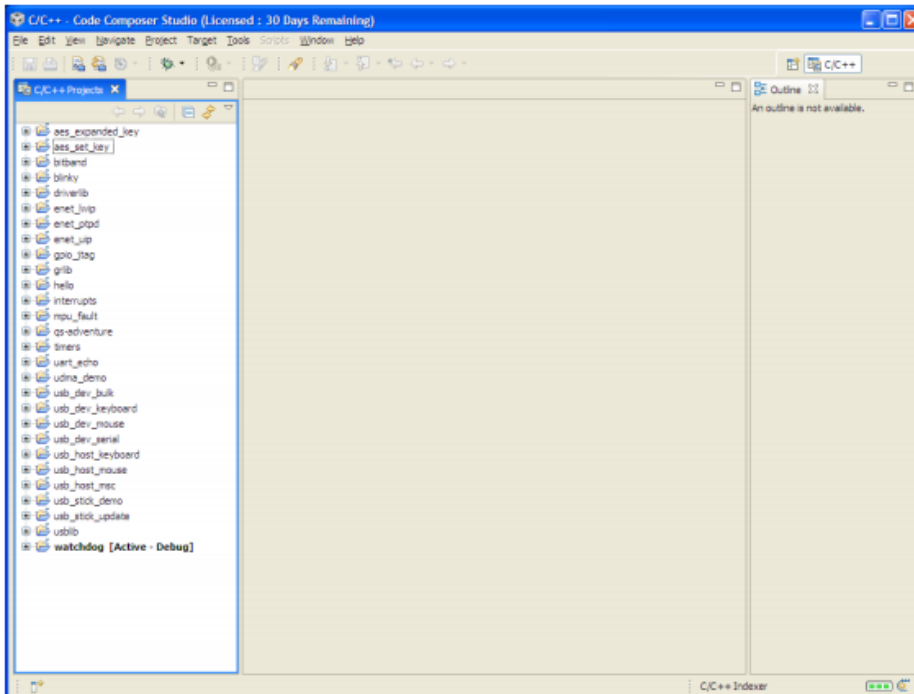
- 1) Select the “Import Existing CCS/CCE Eclipse Project” option from the Project menu in the IDE.



2) 2. The “Import” dialog box appears. Browse to the root directory for your chosen board. The example screen shot below has the EK-LM3S9B90 board as the chosen board. (StellarisWare\boards\ek-lm3s9b90). Be sure that the examples are selected in the listed projects and click “Finish.”

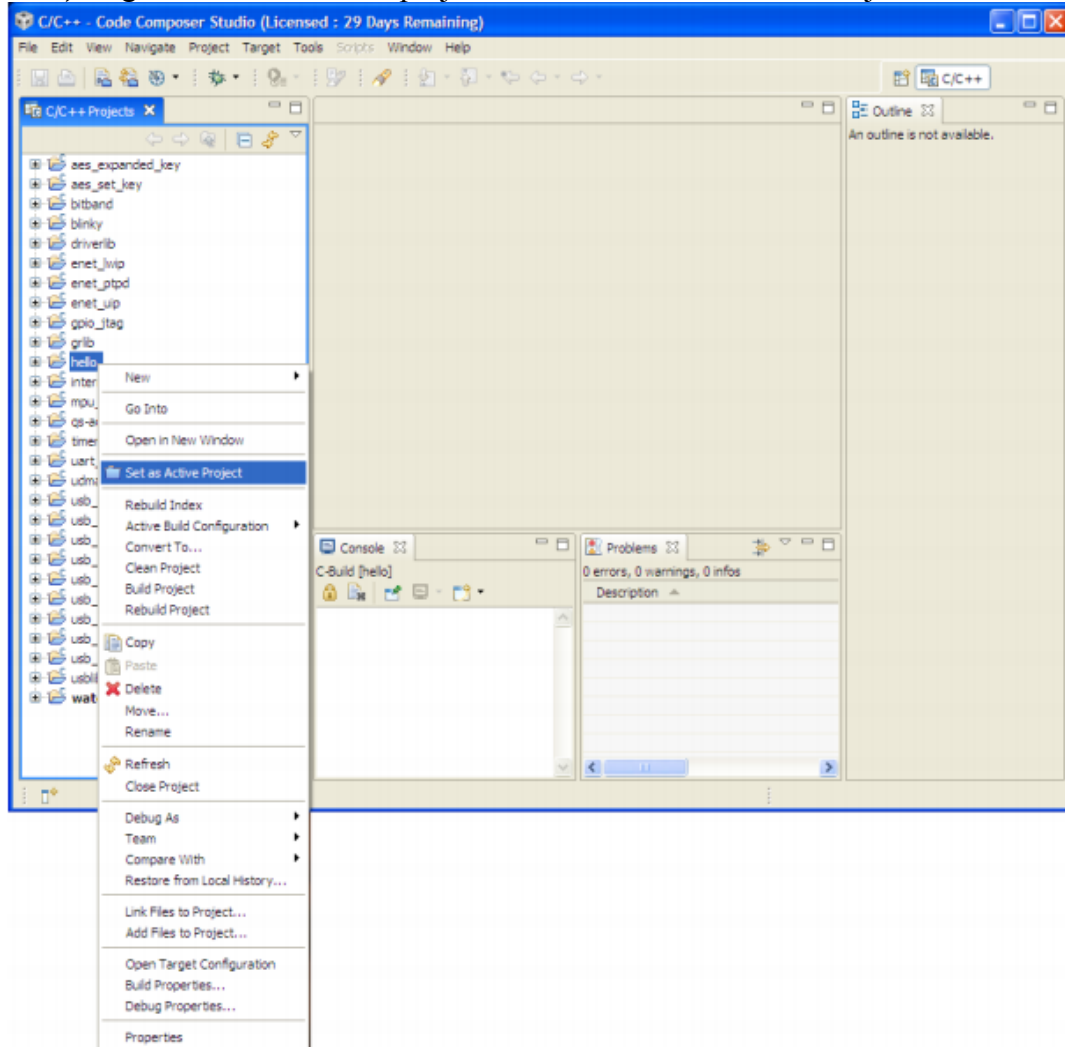


3) All of the projects now show in the Projects Explorer Window.

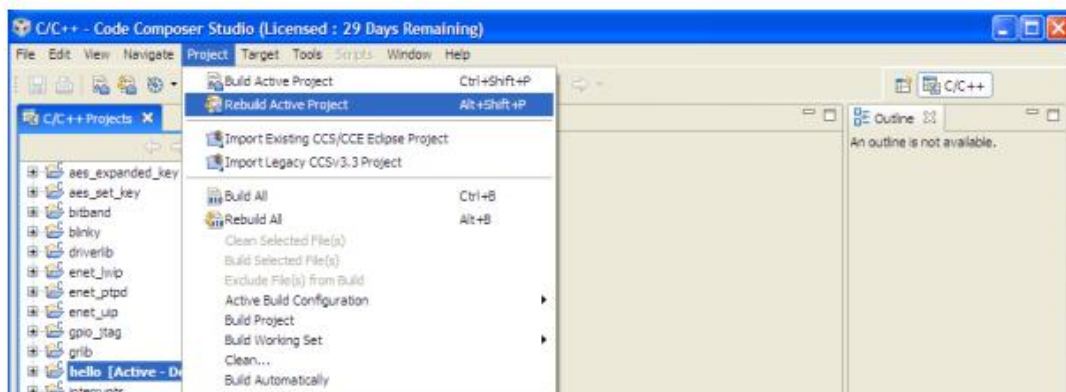


Step 6: Building and debugging a project

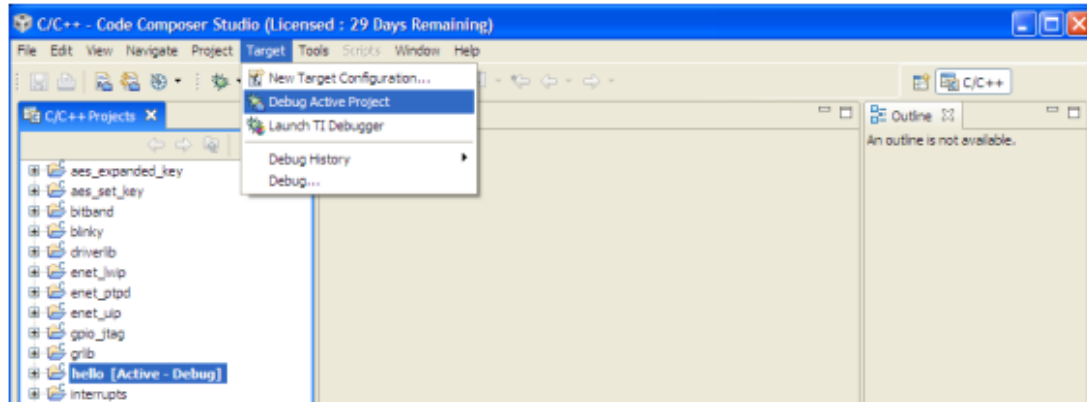
1) Right click on the hello project and select “Set as Active Project.”



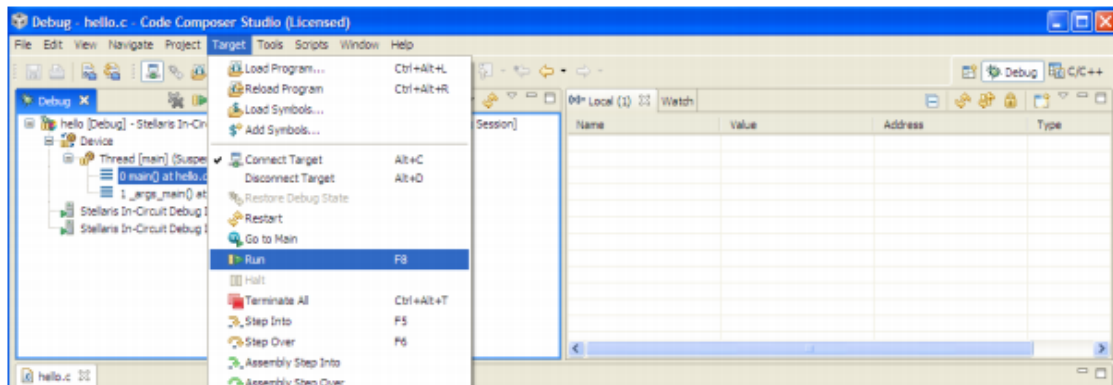
2) Click on the Project pull-down menu and select “Rebuild Active Project.” The build may take a few moments. As the project builds, messages scroll by in the console window. When the build is complete, the words “Build complete for project hello” appear in the console window.



3) Click on the Target pull-down menu and select “Debug Active Project.” Alternatively, you can simply click the “Debug Launch” icon on the toolbar.



4) The Code Composer Studio debugger automatically connects to your evaluation board, programs the Flash memory, and runs to the beginning of the main() function. From here, you can examine and modify memory, program variables and processor registers, set breakpoints, step, and perform other typical debugging activities. To run the program, select “Run” from the Target pull-down menu.



And you can see the output on the LED



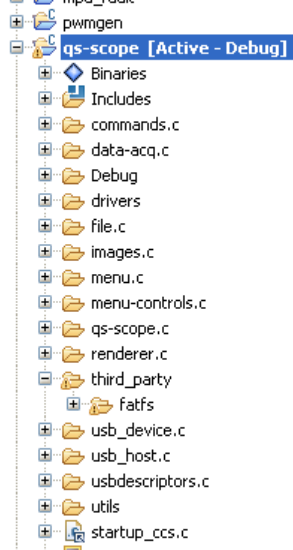
One of the functions of Stellaris LM3S3748 in our Project is the Oscilloscope. Besides the build in function of source code that it brings, we also need the microcontroller helps us do the calculation and displaying the Beat per minutes (BPM).

Here is one example how we add the features of the Stellaris. First, our group figure the time base that it original has is not sufficient, the maximum time base is only 250 ms, we expect to have a time base at 1s, so it can close to our heart beat per seconds, which can provide higher resolution of the ECG signal on the LED.

The first step is same, we set the qs_scope.c file as the active project, rebuild, debug and run it.



Next step is to find the C file that contains function of the timebase.



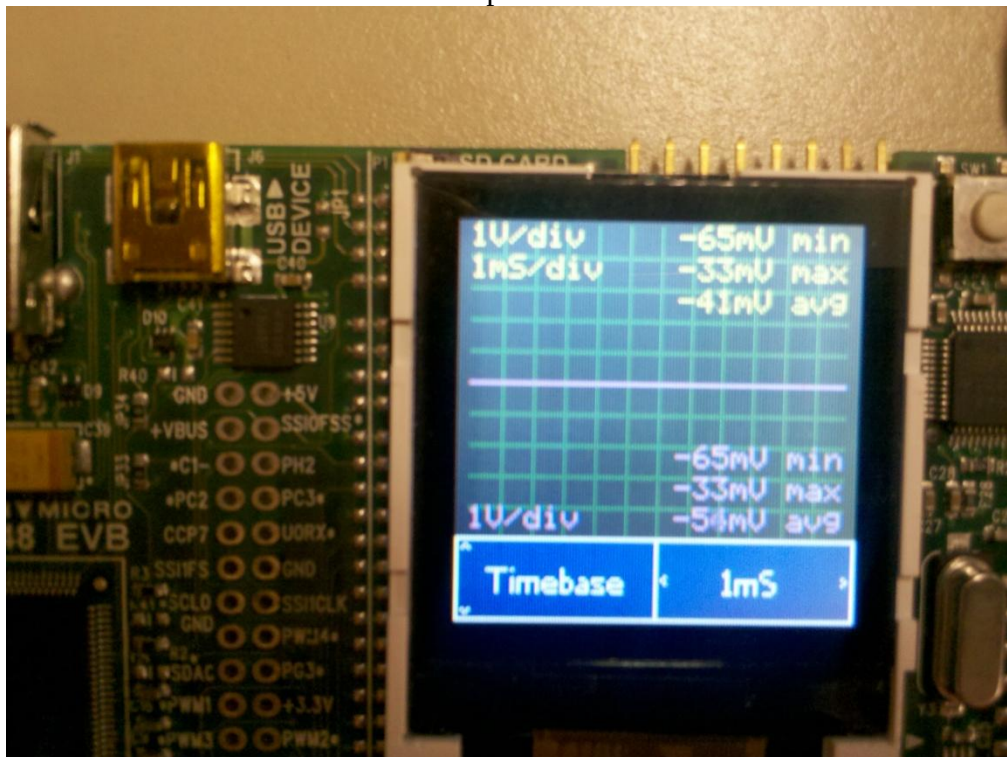
Choose the menu_control.c and open it.

```

tControlChoice g_psTimebaseChoices[] =
{
    { "2uS", 2 },
    { "5uS", 5 },
    { "10uS", 10 },
    { "25uS", 25 },
    { "50uS", 50 },
    { "100uS", 100 },
    { "250uS", 250 },
    { "500uS", 500 },
    { "1mS", 1000 },
    { "2.5mS", 2500 },
    { "5mS", 5000 },
    { "10mS", 10000 },
    { "25mS", 25000 },
    { "50mS", 50000 },
    { "100mS", 100000 },
    { "200mS", 200000 },
    { "500mS", 500000 },
    { "1S", 1000000 }
}

```

And 500ms and 1s in the format as it provided.



- **Conclusion:**

We have now installed the Code Composer Studio development tools and used them to build and load an example application on Stellaris Evaluation Board. From here, we can experiment with the debugger or start creating our own application using the example projects as a reference. Changing the time base is a simple example that how our group add features to the Stellaris. To have the BPM display on the LED, we need to implement our signal processing algorithm and write in C into the Stellaris.