

Tiva TM4C123G LaunchPad Blink the RGB

Contents [\[hide\]](#)

1 Project0 Your First Project

1.1 Introduction

1.2 Download and Install Software You Will Need

1.2.1 Single Download

1.2.2 Individual Downloads

1.2.3 Optional Files

1.3 Hardware Setup and Driver Installation

1.4 Import Project0 files into Code Composer Studio

1.4.1 Test Project0 Example

1.5 Modify Project0

1.5.1 Download and Debug your Project0

1.5.2 Experiment

1.6 Toast Success

1.7 Mac OSX

Project0 Your First Project

Introduction

Note that the video below uses the Stellaris LM4F120 LaunchPad, but the content of the video is equally applicable to the [Tiva C Series TM4C123G LaunchPad](#).

LaunchPad Resource Portal

This wiki is open and can be edited by all!

More information available @ www.ti.com/launchpad

• Hardware Tools, Documentation, Schematics, etc

- [List of available LaunchPad hardware](#)
- [List of available BoosterPack plug-in modules](#)

• Software Tools

- Software Development Environments
 - [Code Composer Studio \(CCS\)](#)
 - [Energia Project](#)
 - [TivaWare for C Series](#)
 - [Other MSP430 Software Tools](#)
 - [Other C2000 Software Tools](#)

• Resources (Tutorials, Code examples & Projects)

- [MSP430 LaunchPad](#)
- [C2000 LaunchPad](#)
- [Tiva C Series LaunchPads](#)
- [Hercules LaunchPads](#)
- [Share your LaunchPad-based projects here!](#)

• Build Your Own BoosterPack (BYOB)

- [BoosterPack Baseline Standard](#)



Click here to return to the LaunchPad Site

The `#evp` parser function was deprecated in EmbedVideo 2.0. Please convert your parser function tag to `#ev`.

In this project we will walk you through the process of importing your first project and modifying the project behavior. This Lesson will verify that your tools are properly installed and that you can edit, compile, download and debug your code on the Tiva™ TM4C123G LaunchPad.

We will cover:

- Installing Code Composer Studio
- Installing the debug drivers
- Installing TivaWare™ for C Series
- Importing a project into Code Composer Workspace

- Edit, Compile, Download and Debug the Project0 starter example

Download and Install Software You Will Need

There are two ways to get all the tools you will need. One is to download a single package that contains all of the installers and documentation. The alternate if you already have one or more of the components needed is to download them separately. Use the separate method to obtain Code Composer Studio (CCS) for Linux.

Single Download

Use this if you don't have CCS or have not used Tiva C Series devices before.

1. Download the EK-TM4C123GXL-CCS package from [here](#) .
You can also download a package for one of the other supported compilers. This tutorial covers only CCS. The process should be similar.
2. Unzip the package.
3. Find the CCS Installer in the Software\CCS directory of the download package.
4. Find the TivaWare installer in the Software\TivaWare directory.
5. Note the location of the Software\ICDI directory. You will use this in a later step to install device drivers.

Individual Downloads

Use this if you already have CCS or one of the other components.

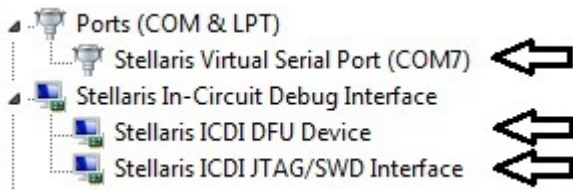
1. Download the [ICDI Drivers](#) . Extract the zip file and note its location for later use.
2. Download the SW-EK-TM4C123GXL package from [here](#) .
3. Download [Code Composer Studio](#) . This tutorial assumes you use the Windows version.
Linux support for this LaunchPad's ICDI interface in CCS is considered Beta at this time.
See [this](#) forum thread for more information

Optional Files

- [LMFlashProgrammer](#) is a handy windows tool that can be used to directly download binary images to the LaunchPad and other Tiva C Series kits.
 - It also can easily perform many flash program, erase and verify operations.
 - It can be used to program the USER0 and USER1 flash registers of the device which are often used for serial numbers or unique identification.
 - Perform a "Debug Mass Erase" which can unlock and restore a device that a user has locked themselves out of. Restores the device to a factory like condition.
 - Can perform an ICDI update of the firmware on the LaunchPad debug interface.

Hardware Setup and Driver Installation

1. Extract the [Stellaris ICDI Drivers](#) .
2. Plug the LaunchPad into your PC's USB port
3. Follow the [driver installation guide](#) .
4. Your device manager should show three Stellaris devices like the image below.



Import Project0 files into Code Composer Studio

- Use the [Tiva C Series QuickStart Guide for Code Composer Studio](#) to complete CCS and TivaWare installation.
- Use the quickstart guide to import the TivaWare libraries and ek-tm4c123gxl (Tiva TM4C123G LaunchPad) program examples, including the project0 example.

Test Project0 Example

Before moving on be sure to test the project0 example prior to making any modifications.

If you run into trouble double check device manager and the three Stellaris devices are listed as mentioned above. Occasionally you must re-install the drivers even though the devices are listed as working correctly. If you suspect the drivers are working and the problem is CCS related try LMFlashProgrammer to download one of the binary files that come with StellarisWare. All of these binaries are tested and known to work. Make sure you select an ek-tm4c123gxl binary that you have not rebuilt or modified.

The project0 example should be blinking the LED alternating between RED and BLUE.

Modify Project0

If you are still in a debug session terminate the session and return to the CCS Edit perspective.

Note that all three LED colors are already defined on lines 36-38 of project0.c

```
#define RED_LED    GPIO_PIN_1
#define BLUE_LED   GPIO_PIN_2
#define GREEN_LED  GPIO_PIN_3
```

Note on line 75 of project0.c that all three LED GPIO pins have been configured for output mode

```
GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, RED_LED|BLUE_LED|GREEN_LED);
```

Copy Lines 82 through 90 of project0.c. Note that the 2nd parameter serves as a bit mask for which pins to write.

```
//
// Turn on the LED
//
GPIOPinWrite(GPIO_PORTF_BASE, RED_LED|BLUE_LED|GREEN_LED, RED_LED);

//
// Delay for a bit
```

```
//  
SysCtlDelay(2000000);
```

Paste the copied lines below line 100. The 'while(1)' should now look like this:

```
while(1)  
{  
    //  
    // Turn on the LED  
    //  
    GPIOPinWrite(GPIO_PORTF_BASE, RED_LED|BLUE_LED|GREEN_LED, RED_LED);  
  
    //  
    // Delay for a bit  
    //  
    SysCtlDelay(2000000);  
  
    //  
    // Turn on the LED  
    //  
    GPIOPinWrite(GPIO_PORTF_BASE, RED_LED|BLUE_LED|GREEN_LED, BLUE_LED);  
  
    //  
    // Delay for a bit  
    //  
    SysCtlDelay(2000000);  
  
    //  
    // Turn on the LED  
    //  
    GPIOPinWrite(GPIO_PORTF_BASE, RED_LED|BLUE_LED|GREEN_LED, RED_LED);  
  
    //  
    // Delay for a bit  
    //  
    SysCtlDelay(2000000);  
}
```

Change line 105 from RED_LED to GREEN LED.

```
GPIOPinWrite(GPIO_PORTF_BASE, RED_LED|BLUE_LED|GREEN_LED, GREEN_LED);
```

Download and Debug your Project0

Click the bug icon to build, download and debug your changes. CCS will change back to the Debug Perspective. The code will run until the first line of main and then stop. Click the run icon to resume operation. Observe red, blue and green blinking pattern.

Play with CCS to pause, set breakpoints and get to know the environment.

Experiment

- Try turning on more than one color.

- Experiment with the bit mask parameter of GPIOPinWrite.
- Have fun, be creative.
- Use this as a starting point for your own creations.
- Look at the qs-rgb to see a set of RGB driver API's that use the timer PWM functionality to control brightness and mix colors.

Toast Success

Congratulations. You have just imported, modified, downloaded and debugged your first TivaWare Project.