

Blink your first LED

From Texas Instruments Wiki

Blink your first LED

Translate this page to cs - Česky Translate

Contents

- 1 My First MSP430 LaunchPad Project
- 2 Things you'll need
- 3 Hardware Setup
- 4 Creating a new Code Composer Studio Project
- 5 Writing Code
- 6 Other exercises!

My First MSP430 LaunchPad Project

In this project, we will learn a few things:

- How to create a new project with Code Composer Studio
- Learn how to blink the on-board Red LED on the MSP430 LaunchPad
- Change the speed of the blinking Red LED
- Learn how to toggle between the Red and Green LED

Things you'll need

1. MSP430 LaunchPad Evaluation Kit (MSP-EXP430G2) -- Get one here! (<https://estore.ti.com/MSP-EXP430G2-MSP430-LaunchPad-Value-Line-Development-kit-P2031.aspx>)
2. Code Composer Studio (Software Development Environment) -- Download & install it
3. 10 minutes

Hardware Setup

1. The MSP430 LaunchPad kit includes everything you need out of the box. To start programming your MSP430 LaunchPad, you'll first have to install Code Composer Studio onto your computer. This will install all of the required drivers for your new MSP430 LaunchPad kit.
2. Next, simply plug in your LaunchPad with the included USB cable to your Windows PC. The green power (PWR) LED should glow.
3. If prompted, let Windows automatically install the software.

Creating a new Code Composer Studio Project

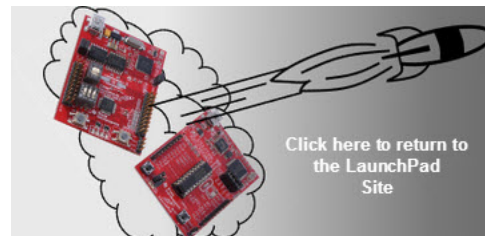
1. Now that our hardware is setup, we will open up Code Composer Studio. As it is opening, CCS will ask you to select a workspace.
2. Since this is our first project, we'll create a new one called "LaunchPad_Projects". A workspace is where all of your Code Composer Studio projects will live. Once created, *press OK*

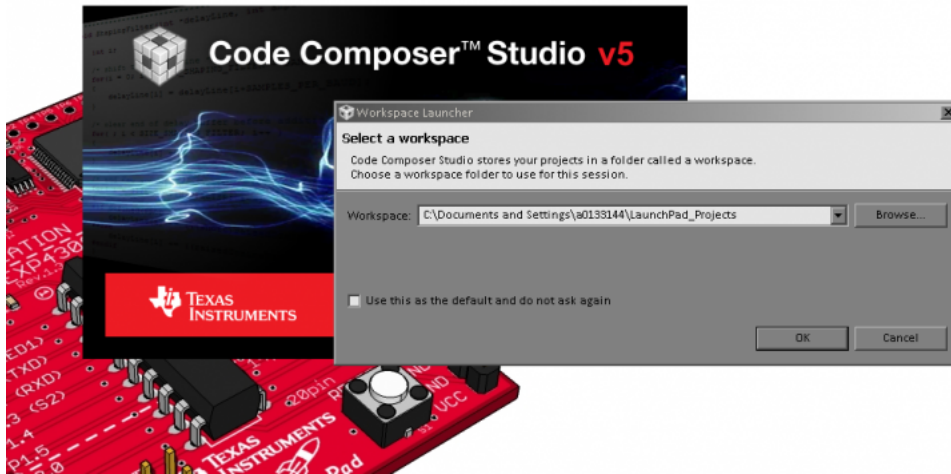
LaunchPad Resource Portal

This wiki is open and can be edited by all!

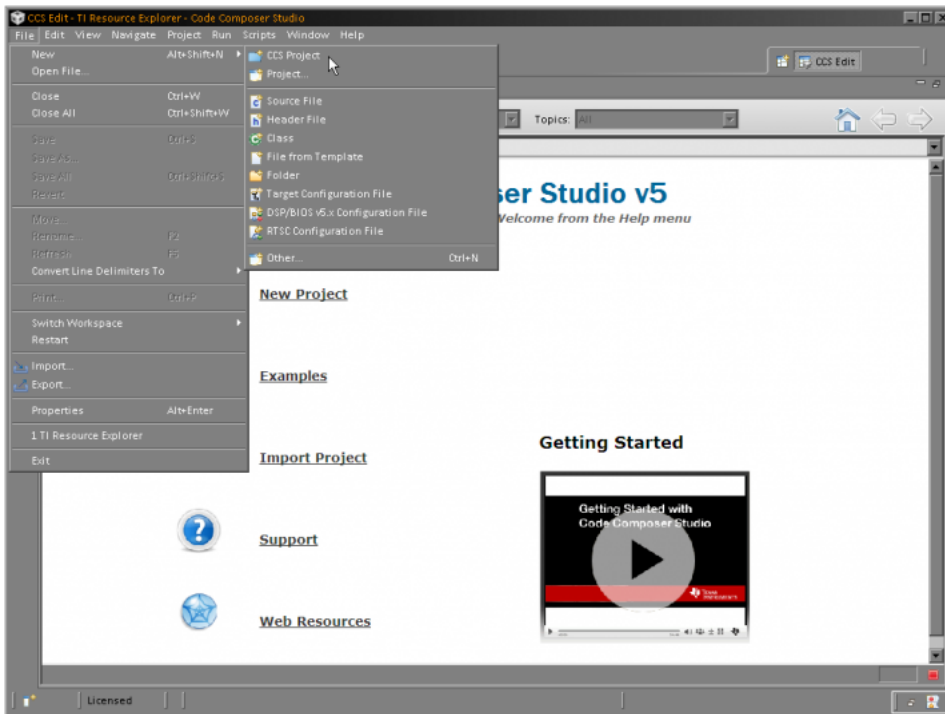
More information available @ www.ti.com/launchpad (<http://www.ti.com/launchpad>)

- **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)
 - TivaWare for C Series (<http://www.ti.com/tivaware>)
 - Other MSP430 Software Tools TivaWare for C Series
 - Other C2000 Software Tools (<http://www.ti.com/tool/controlsuite>)
- **Resources (Tutorials, Code examples & Projects)**
 - MSP430 LaunchPad
 - C2000 LaunchPad
 - Tiva C Series LaunchPad
 - Hercules LaunchPad
 - Share your LaunchPad-based projects here! (<http://e2e.ti.com/group/msp430launchpad/m/project/default.aspx>)
- **Build Your Own BoosterPack (BYOB)**
 - BoosterPack Baseline Standard
 - BoosterPack Design Guide
 - BoosterPack Standards and Design Guide

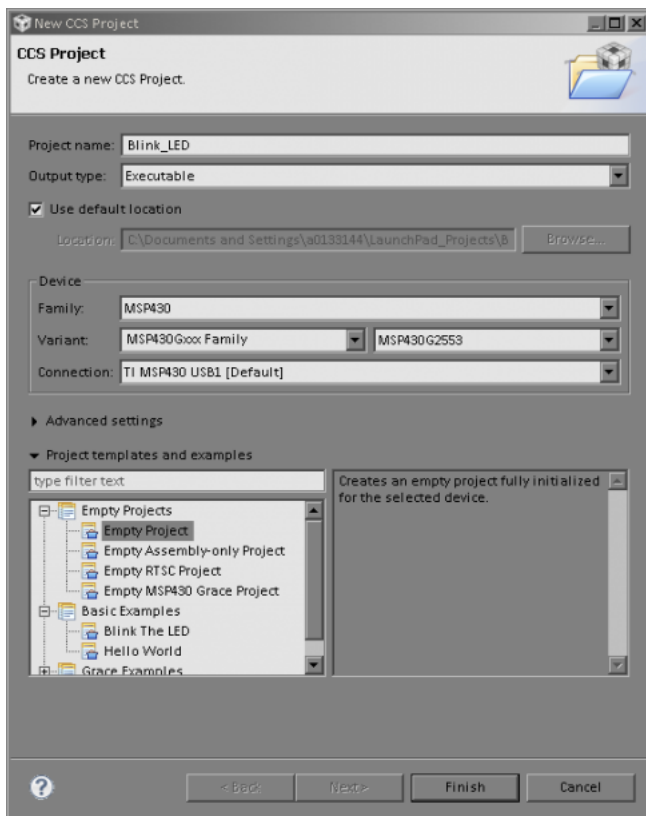




3. Once CCS is opened, we can create a new project by going to **File > New > CCS Project**



4. This will open up the "New CCS Project" Window. Within this window, we need to do 2 things. Name our project & choose our *Device Variant*.
 Let's name our project "Blink_LED"
 We also need to choose the appropriate MSP430 device. For this tutorial, we will program the MSP430G2553 device that comes pre-populated on the MSP430 LaunchPad.
 (Due to the simplicity of this particular tutorial, any of the MSP430G2xx devices will work for this example!)
 Then, *click "Finish"*



Writing Code

1. Now, we have our blank canvas! We can finally start writing code!

```
#include <msp430g2553.h>

unsigned int i = 0;                // Initialize variables. This will keep count of how many cycles between LED toggles

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD;      // Stop watchdog timer. This line of code is needed at the beginning of most MSP430 projects.
    // This line of code turns off the watchdog timer, which can reset the device after a certain period of time.

    P1DIR |= 0x01;                // P1DIR is a register that configures the direction (DIR) of a port pin as an output or an input.

    // To set a specific pin as output or input, we write a '1' or '0' on the appropriate bit of the register.

    // P1DIR = <PIN7><PIN6><PIN5><PIN4><PIN3><PIN2><PIN1><PIN0>

    // Since we want to blink the on-board red LED, we want to set the direction of Port 1, Pin 0 (P1.0) as an output
    // We do that by writing a 1 on the PIN0 bit of the P1DIR register
    // P1DIR = <PIN7><PIN6><PIN5><PIN4><PIN3><PIN2><PIN1><PIN0>
    // P1DIR = 0000 0001
    // P1DIR = 0x01    <-- this is the hexadecimal conversion of 0000 0001

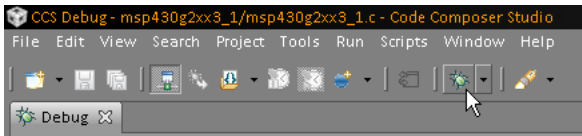
    for (;;)                      // This empty for-loop will cause the lines of code within to loop infinitely
    {

        P1OUT ^= 0x01;           // Toggle P1.0 using exclusive-OR operation (^=)

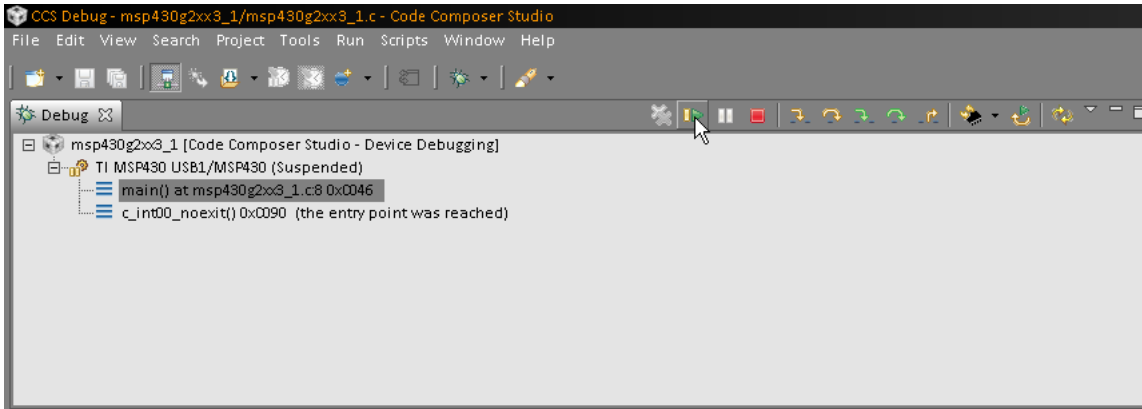
        // P1OUT is another register which holds the status of the LED.
        // '1' specifies that it's ON or HIGH, while '0' specifies that it's OFF or LOW
        // Since our LED is tied to P1.0, we will toggle the 0 bit of the P1OUT register

        for(i=0; i< 20000; i++); // Delay between LED toggles. This for-loop will run until the condition is met.
        //In this case, it will loop until the variable i increments to 20000.
    }
}
```

1. Now that we have written our code, we can download it to our MSP430 LaunchPad that is plugged into the USB port! We can do this by *clicking the debug button*



2. Clicking the Debug button will take us to the CCS Debug View. Here, we can *press the Run button* to start running the code we just wrote.



3. At this point, your Red LED should be blinkiing! **Congratulations!**

Other exercises!

1. Now that we have our LED blinking, play around with the number inside of the for-loop to change the speed of the blinking LED. The smaller the number, the shorter the delay between LED toggles. Alternatively, the larger the number, the longer the delay. Try values like 5000, 40000, etc.
2. Another exercise is getting the green LED to blink as well. The green LED is tied to Port P1.6. Using the P1DIR and P1OUT registers we used above, see if you can get both LEDs to blink. Can you make them blink in unison? Can you make them blink alternatively?

Be sure to check out more resources for the MSP-EXP430G2 LaunchPad [here!](#)



For technical support please post your questions at <http://e2e.ti.com>. Please post only comments about the article **Blink your first LED** here.

Links



Amplifiers & Linear
(http://www.ti.com/lscs/ti/analog/amplifier_and_linear.page)
Audio (http://www.ti.com/lscs/ti/analog/audio/audio_overview.page)
Broadband RF/IF & Digital Radio
(<http://www.ti.com/lscs/ti/analog/rfif.page>)
Clocks & Timers
(http://www.ti.com/lscs/ti/analog/clocksandtimers/clocks_and_timers.page)
Data Converters
(http://www.ti.com/lscs/ti/analog/dataconverters/data_converter.page)

DLP & MEMS (<http://www.ti.com/lscs/ti/analog/mems/mems.page>)
High-Reliability (http://www.ti.com/lscs/ti/analog/high_reliability.page)
Interface (<http://www.ti.com/lscs/ti/analog/interface/interface.page>)
Logic (http://www.ti.com/lscs/ti/logic/home_overview.page)
Power Management
(http://www.ti.com/lscs/ti/analog/powermanagement/power_portal.page)

Processors
(http://www.ti.com/lscs/ti/processors/processors_overview.page)
■ AR!
(http://www.ti.com/lscs/ti/processors/arm_processors.page)
■ Digi
(http://www.ti.com/lscs/ti/processors/digital_processors.page)
■ Mic
(<http://www.ti.com/lscs/ti/processors/microprocessors.page>)
■ OM
(http://www.ti.com/lscs/ti/processors/om_processors.page)
proc

Retrieved from "http://processors.wiki.ti.com/index.php/Blink_your_first_LED"

Comments

Comments on Blink your first LED
