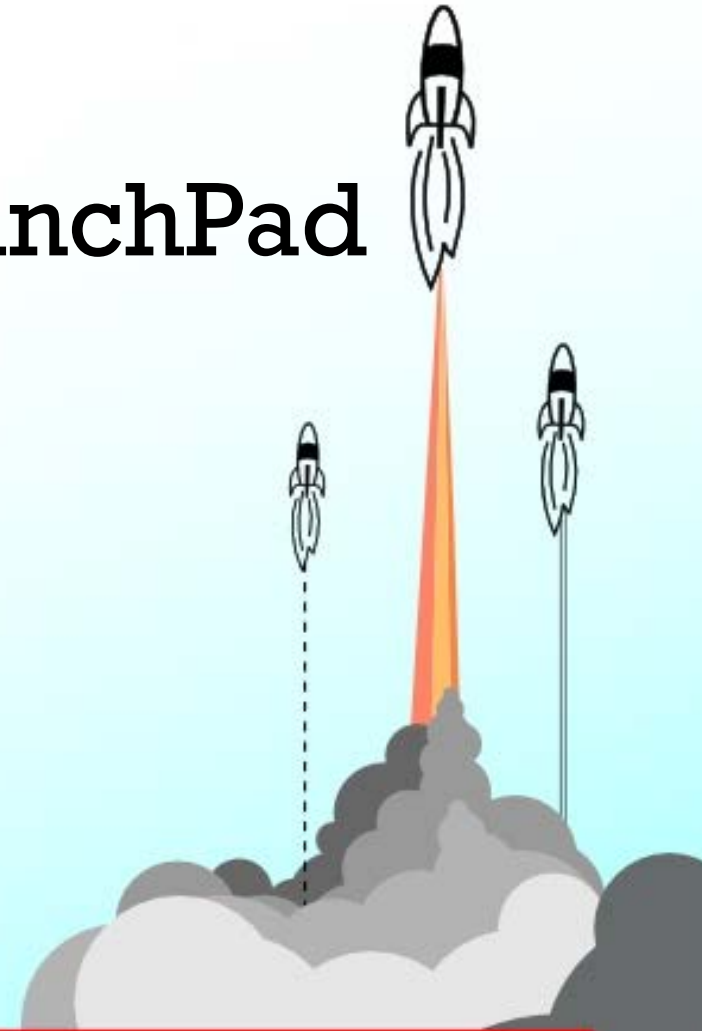


# MSP430F5529 LaunchPad

## Project0



# Overview

## In this exercise you will learn:

- How to create a new project with Code Composer Studio (CCS)
- 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 will need:

- MSP430F5529 Launchpad Evaluation Kit
  - MSP-EXP430F5529LP
- [Code Composer Studio](#) (Software Development Environment)
- 10-15 minutes

# Hardware Setup

- The MSP430F5529 LaunchPad kit includes everything you need out of the box
- To start programming, you'll first have to install CCS which contains the required drivers for your new MSP430F5529 LaunchPad kit
- Plug your F5529 LaunchPad into the PC with the included USB cable
- If prompted, let Windows automatically install the software.

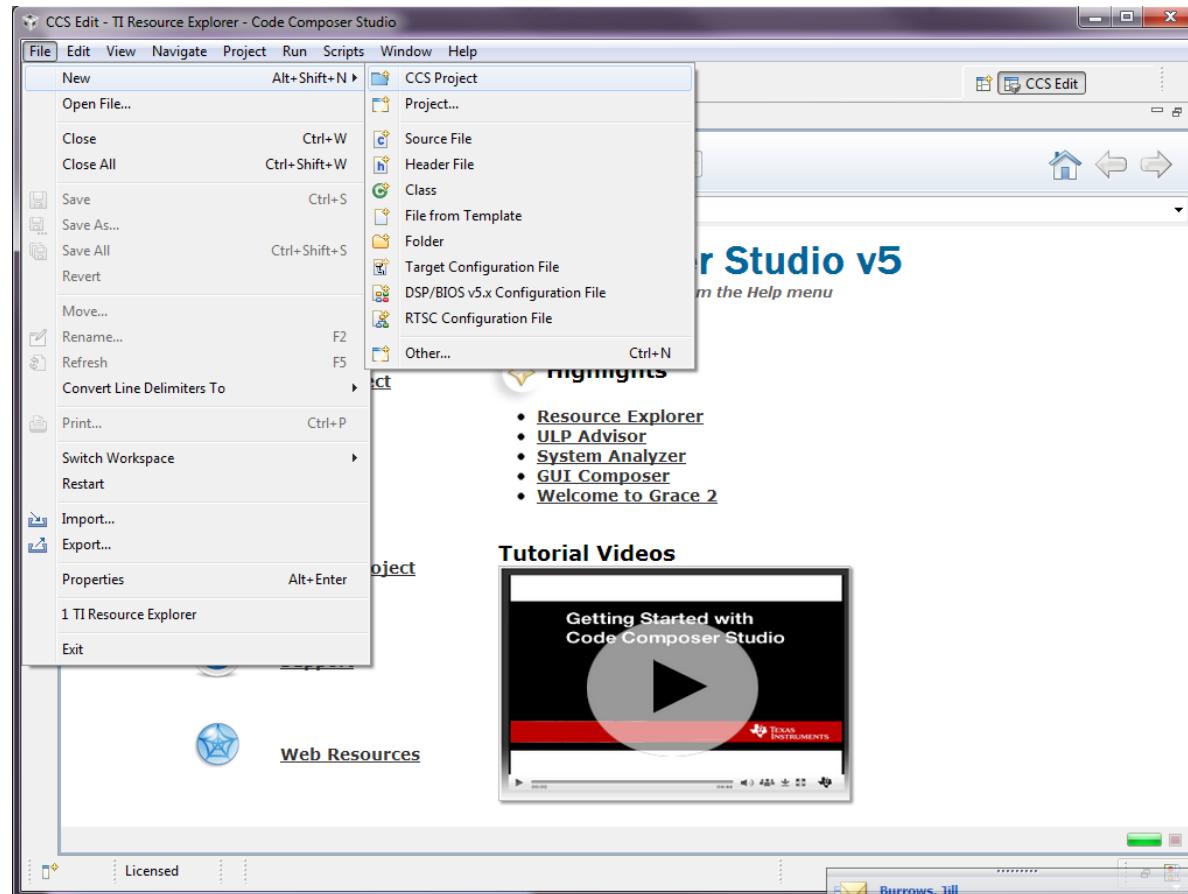
# Create a New CCS Workspace

- Upon opening CCS, it will ask you to select a workspace
- A workspace is where all of your CCS projects will be stored. Create a workspace folder, then *press OK*



# Create a New CCS Project

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

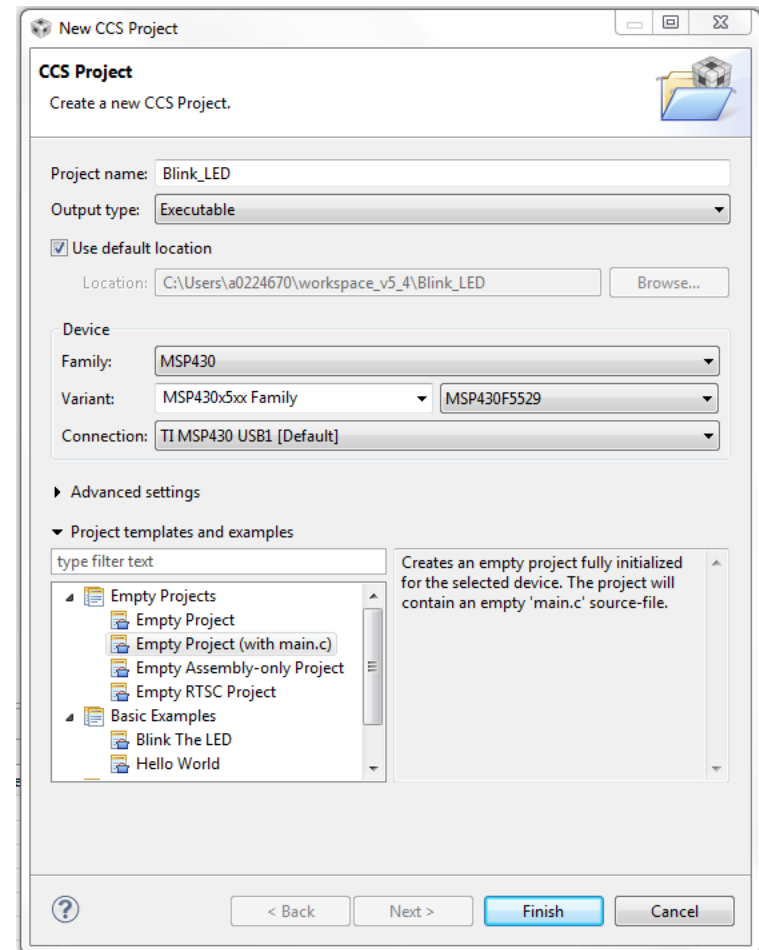


# Choose CCS Project Settings

This will open up the "New CCS Project" window. Within this window, we need to do 2 things.

1. Name our project
2. Choose our *Device Variant*.

1. Let's name our project "Blink\_LED"
2. We also need to choose the appropriate MSP430 device. For this tutorial, we will program the MSP430F5529 device that comes pre-populated on the MSP430F5529 LaunchPad



# Write Code

In the main.c file we have a blank canvas to write our code!

```
#include <msp430.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

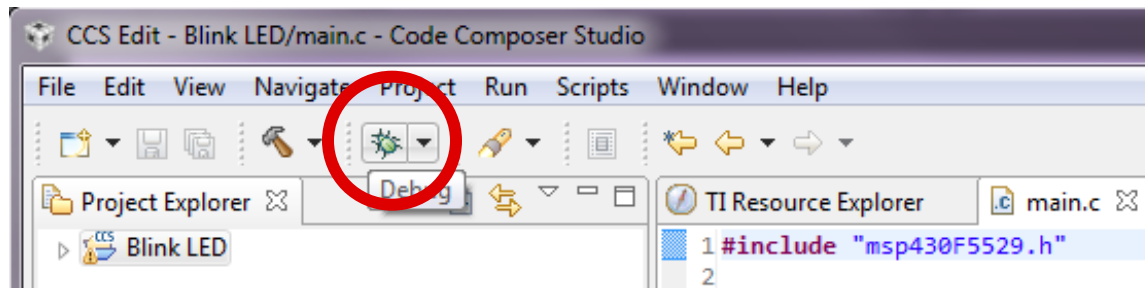
    for (;;)                 // This empty for-loop will cause the lines of code within to loop infinitely
    {
        P1OUT ^= 0x01;       // Toggle P1.0 using exclusive-OR operation (^=)

                               // 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.
    }
}
```

# Download Code

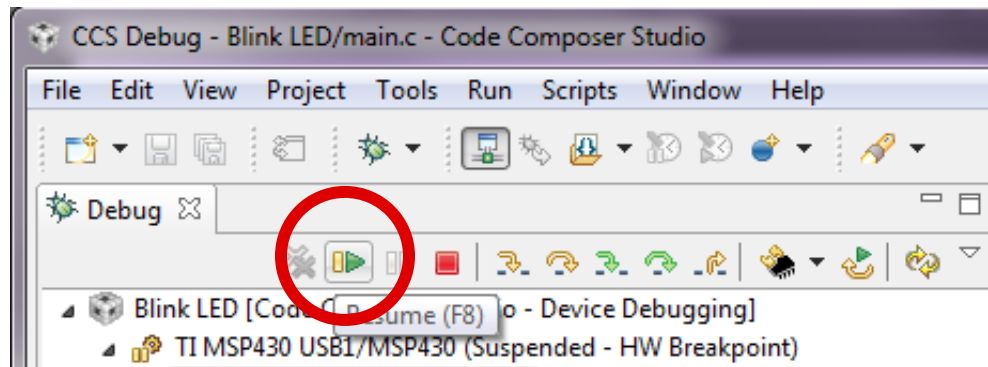
- Now that we have written our code, we can download it to our MSP430 LaunchPad that is plugged into the USB port!
- Code is downloaded by clicking the *debug* button





# Run Code

- 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.



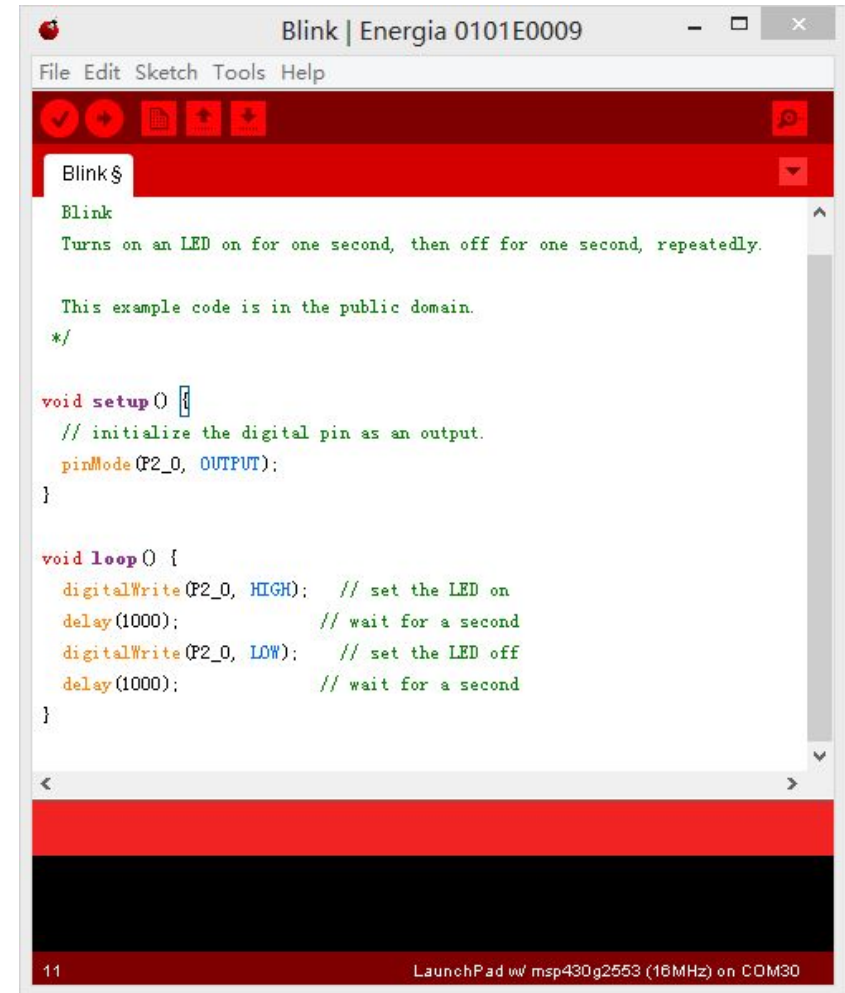
- At this point, your red LED should be blinking.  
***Congratulations!***

# Additional Exercises

- 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, or the faster the blinking. Alternatively, the larger the number, the longer the delay. Try values like 5000, 40000, etc.
- Another exercise is getting the green LED to blink as well. The green LED is tied to Port P4.7. Using the P4DIR and P4OUT 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?

# Other ideas!

- Congratulations! You've just developed your first MSP-EXP430F5529LP LaunchPad project using TI's Code Composer Studio!
- There are other development tools out there. One example is a community-driven code editing tool called **Energia**.
- You can get Energia @ [www.energia.nu](http://www.energia.nu)



```
Blink | Energia 0101E0009
File Edit Sketch Tools Help
Blink $
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.
*/

void setup() {
  // initialize the digital pin as an output.
  pinMode(P2_0, OUTPUT);
}

void loop() {
  digitalWrite(P2_0, HIGH); // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(P2_0, LOW);  // set the LED off
  delay(1000);              // wait for a second
}

11 LaunchPad w/ msp430g2553 (16MHz) on COM30
```

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have **not** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

### Products

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DLP® Products	<a href="http://www.dlp.com">www.dlp.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>
OMAP Applications Processors	<a href="http://www.ti.com/omap">www.ti.com/omap</a>
Wireless Connectivity	<a href="http://www.ti.com/wirelessconnectivity">www.ti.com/wirelessconnectivity</a>

### Applications

Automotive and Transportation	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Communications and Telecom	<a href="http://www.ti.com/communications">www.ti.com/communications</a>
Computers and Peripherals	<a href="http://www.ti.com/computers">www.ti.com/computers</a>
Consumer Electronics	<a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a>
Energy and Lighting	<a href="http://www.ti.com/energy">www.ti.com/energy</a>
Industrial	<a href="http://www.ti.com/industrial">www.ti.com/industrial</a>
Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Space, Avionics and Defense	<a href="http://www.ti.com/space-avionics-defense">www.ti.com/space-avionics-defense</a>
Video and Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>

### TI E2E Community

[e2e.ti.com](http://e2e.ti.com)