

# LAB 1: Get started and pulse generation

---

## Goal

1. Get familiar with development environment (Code Composer Studio for MSP430)
2. Understand the register manipulation, including value assignment, addition, subtraction and bitwise logic.
3. Learn to manipulate IO port of the microcontroller, including the port direction configuration, port value assignment and port interrupt handling.

## Introduction

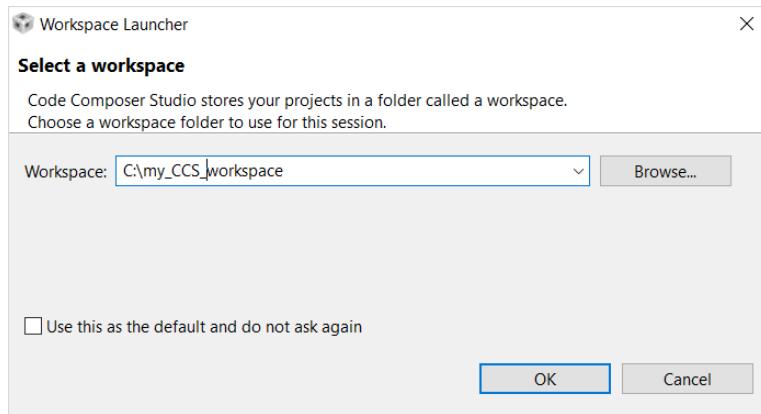
This is the first lab of the course. This lab is separated into mainly two parts. In the first part, the lab focuses on studying the development environment – Code Composer Studio. The lab shows a step-by-step introduction based on one simple task. There are several tasks in this part. The second part of the lab is to use one of the IO pins in the MSP430 board to generate a pulse signal. The pulse frequency is depend on the number of times of the push button is clicked. (More detailed description is given in the second part).

Before the lab, one should hand in the lab preparation which contains the preliminary answer of the questions listed at the end of this lab PM. The lab preparation should be handed in before the lab and the review will be given during the lab individually. For the lab, normally it is 2 people in one group.

## Part 1 Get Started!

### 1. Open Code Composer Studio 7.00 (CCSv7)

Find the program named “Code Composer Studio 7.00” (or maybe some version that earlier or later). When CCSv7 is opened, you should first select a workspace in the following window.

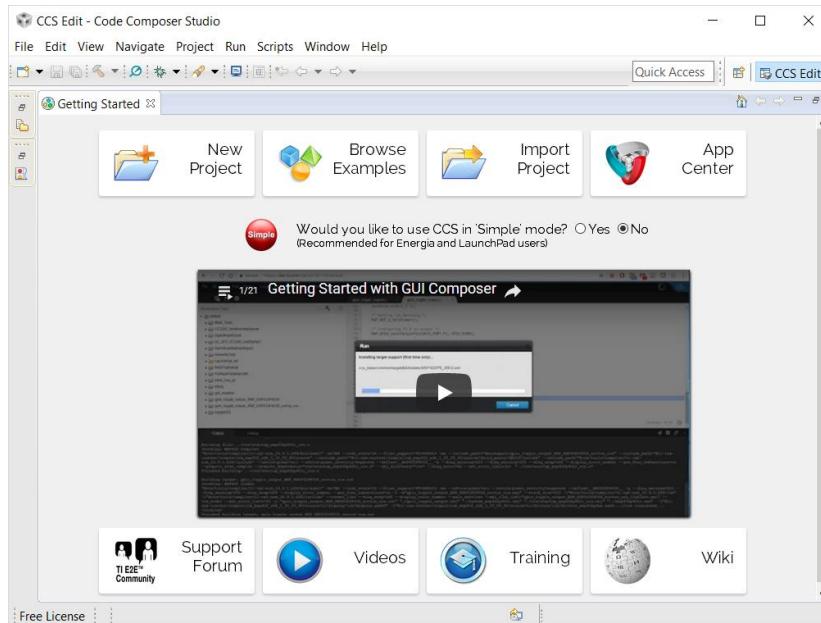


### 2. Setting the Workspace Directory

The Workspace directory you select will save your projects with the project files, configuration files, source files and compiler output files you may want to have access in the future.

Do **NOT** check the “Use this as the default and do not ask again” box.

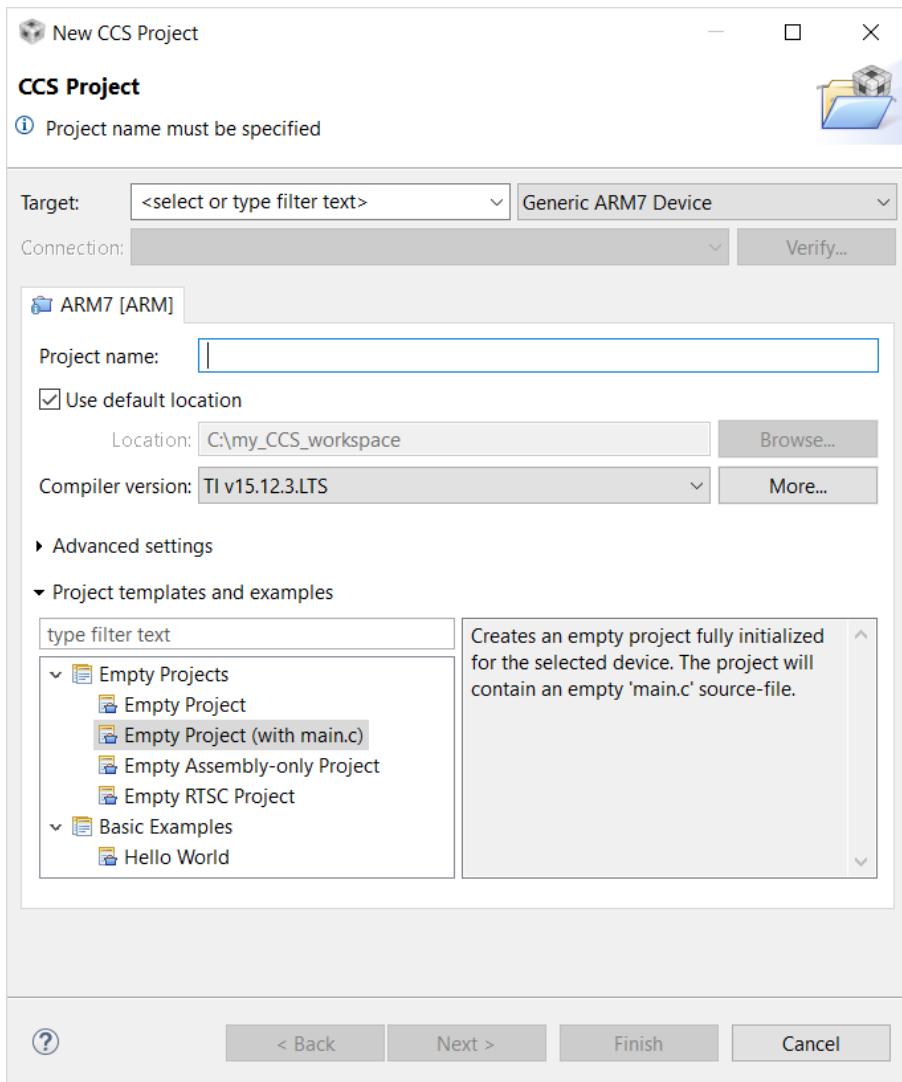
After clicking “OK”, the following window is shown.



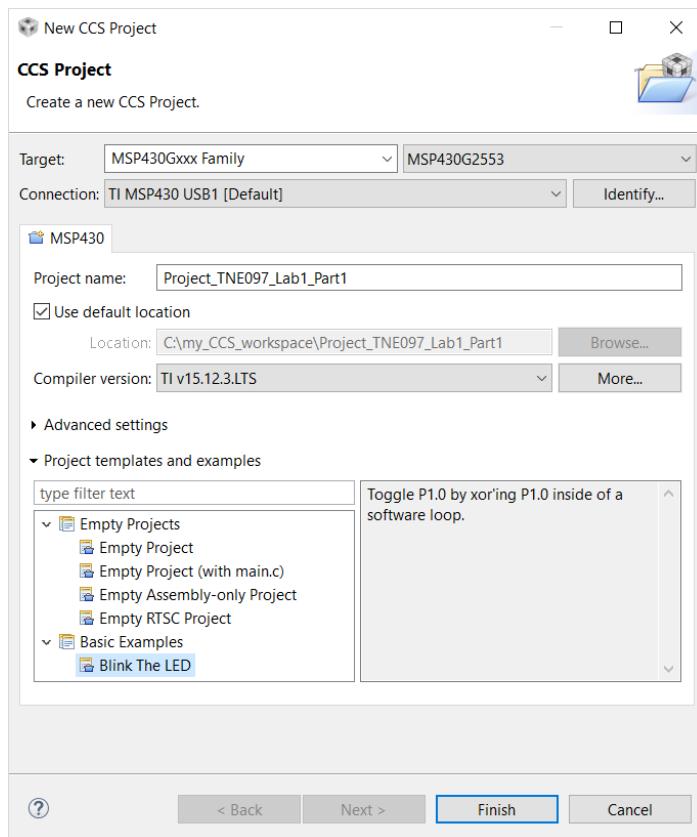
In this page, you can find pretty much useful information about CCS.

### 3. Create a new project

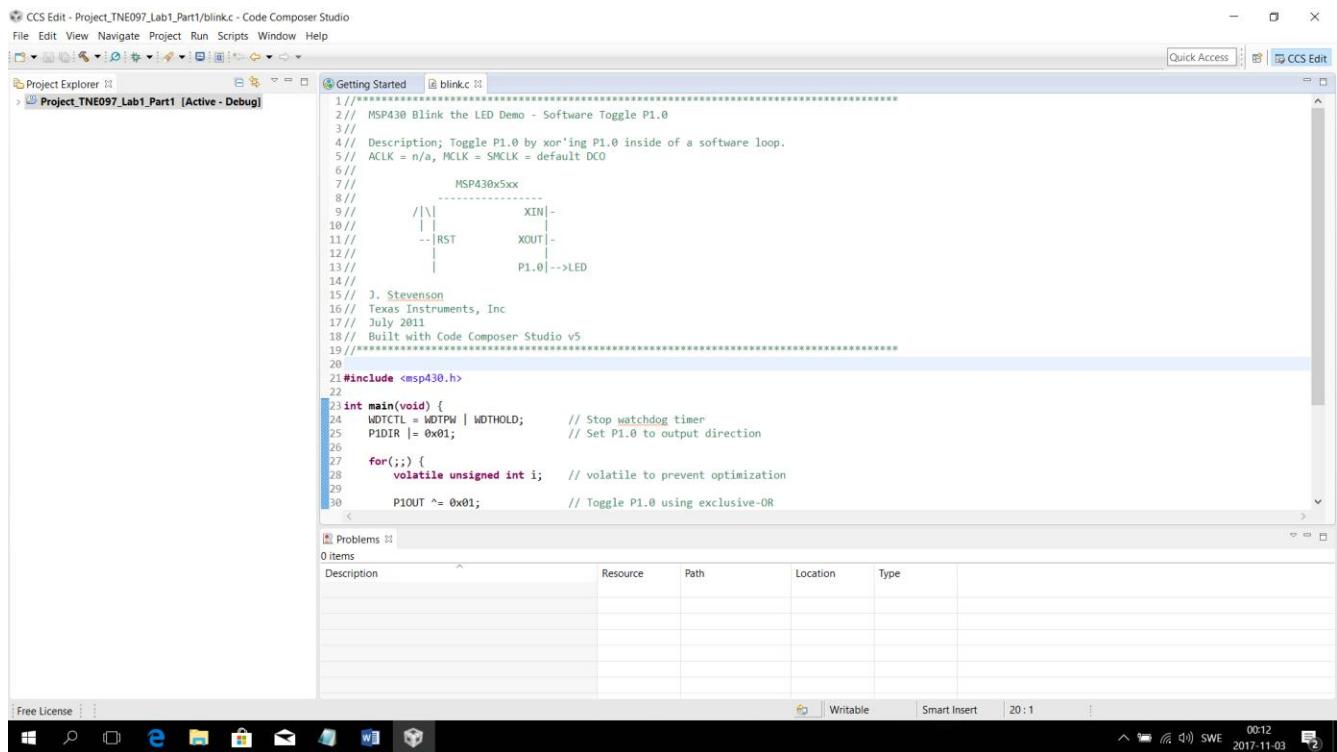
Click “New Project” icon, the following window is shown.



In the “Target” box, select filter text “MSP430Gxxx Family” from a pulldown list and then select “MSP430G2553” from the pulldown list of microcontrollers of the MSP430Gxxx family. Type your project name in the “Project name” box. In the “Project templates and examples” box, expand “Empty Projects” and “Basic Examples”. You can either select “Blink The LED” or “Empty Project (with main.c)”. The description of each selected item is shown in the neighboring box. You can select “Blink The LED” to test the LaunchPad as follows.



After clicking “Finish”, CCS Edit Perspectives is shown.



Now you can also create a new project by clicking “Project -> New CCS Project...” in the menu.

## 4. Connect LaunchPad to PC

Since CCSv6 and CCSv7 do NOT have any simulator for MSP430, we should connect the LaunchPad to the PC to debug programs. Texas Instruments is moving away from providing simulators and instead is focusing on providing low cost development boards. We are going to download our software image into MSP430 development board and do debugging in the board.

Use a USB cable to connect the LaunchPad and the PC. The green LED (LED0) should be ON.

## 5. Compile and Link the programs

In the “Project Explorer” pane, click on “>” beside “Project\_TNE097\_Lab1\_Part1” to view the files of the project. Double click on “blink.c” to view the program.

Click “Project-> Build All” in the menu to compile and link.

The screenshot shows the Code Composer Studio interface. The top menu bar includes File, Edit, View, Navigate, Project, Run, Scripts, Window, Help, and Quick Access. The left sidebar has a Project Explorer with 'Project\_TNE097\_Lab1\_Part1 [Active - Debug]' expanded, showing subfolders like \$, Includes, Debug, targetConfig, and blink.c. Below it is a 'Link\_msp430g2553.cmd' file. A 'Problems' section shows 0 errors, 1 warning, and 0 others. The main workspace contains an 'Getting Started' tab with the file 'blink.c'. The code in 'blink.c' is as follows:

```
1 //***** MSP430 Blink the LED Demo - Software Toggle P1.0
2 //
3 // Description: Toggle P1.0 by xor-ing P1.0 inside of a software loop.
4 //              ACKX = nRA, RCLK = SMCLK = default DCO
5 //              ALMX = nRA, RCLK = SMCLK = default DCO
6 //
7 // MSP430g255x
8 //
9 //          |----- XIN |
10 //          |----- RST   XOUT |
11 //          |----- |
12 //          |----- P1.0 |--> LED
13 //
14 //
15 // 3. Stevenson
16 // Texas Instruments, Inc
17 // July 2011
18 // Built with Code Composer Studio v5
19 //*****
```

The code includes comments explaining the hardware connections and the software logic. The 'Resource' tab below the editor is visible. At the bottom, the 'Console' tab shows the build log:

```
*Building Target: Project_TNE097_Lab1_Part1.out
*Invoking: MSP430 Linker
"C:\ti\ccsv6\tool\compiler\msp430_15.12.3.LTS\bin\cl430" -wmsp --use_hw_psynone --advice:power=off -g --define=_MSP430G2553_ --diag_warning=225 --diag_wrap=off --display_error_number
--print_supported_mems --m"Project_TNE097_Lab1_Part1.map" --strip_size=4k --map_size=8k -l"C:\ti\ccsv6\ccs_base\msp430\include" -l"C:\ti\ccsv6\tools\compiler\msp430_15.12.3.LTS\lib"
[TC] C:\ti\ccsv6\ccs_base\msp430\include\iom430.h:188:1: warning: #include "iom430.h" is never used [-WUnusedIncludeDirective]
[TC] C:\ti\ccsv6\ccs_base\msp430\include\iom430.h:188:1: warning: #include "iom430.h" is never used [-WUnusedIncludeDirective]
[TC] C:\ti\ccsv6\ccs_base\msp430\include\iom430.h:188:1: warning: #include "iom430.h" is never used [-WUnusedIncludeDirective]
linking
remark #10372-0: (ULP 4.1) Detected no uses of low power mode state changing instructions
remark #10372-0: (ULP 4.1) Detected uninitialized Port 2 in this project. Recommend initializing all unused ports to eliminate wasted current consumption on unused pins.
remark #10372-0: (ULP 4.1) Detected uninitialized Port 3 in this project. Recommend initializing all unused ports to eliminate wasted current consumption on unused pins.
*finished building target: Project_TNE097_Lab1_Part1.out"
."
**** Build Finished ****
```

## 6. Debug the program

Click “Run -> Debug” in the menu. Now CCS Debug Perspectives is shown.

The screenshot shows the Code Composer Studio interface with the following details:

- Title Bar:** CCS Debug - Project\_TN8097\_Lab1\_Part1 [Code Composer Studio]
- Menu Bar:** File Edit View Project Tools Run Scripts Window Help
- Toolbar:** Includes icons for Open, Save, Build, Run, Stop, and others.
- Left Panel:** Shows the code editor with the file "main.c" containing C code for a MSP430 application. The code includes comments about blinking P1.0 and setting up the timer. It also includes a section for the ULP Advisor.
- Center Panel:** A modal dialog box titled "ULP Advisor".
  - Content:** Describes the Ultra-Low-Power Advisor (ULP Advisor) which checks for ultra-low-power best practices. It lists a reminder about power consumption and provides links to change default settings, review advice, and view options.
  - Buttons:** "Proceed" and "Cancel".
- Bottom Panel:** Shows the build console output for the project.
- System Tray:** Shows icons for battery level, signal strength, and system status.

A pop-up window “ULP Advisor” might be shown. Click “Proceed” to continue.

Click “Run” in the menu for CCS Debug Perspectives. Choose a command to debug the program. You can also click a command symbol on the menu bar.

The screenshot shows the CCS Debug interface for the Project\_TNE097\_Lab1.Part1\blink.c project. The assembly code window displays the following code:

```
Getting Started blink.c
4// Description: Toggle
5// ACLK = n/a, MCLK = 5
6// FLL = 16MHz
7// MSP430
8// 
9// 
10// 
11// -- RST
12// 
13// 
14// J. Stevenson
15// Texas Instruments, Inc.
16// July 2011
17// 
18// Built with Code Composer Studio 6.2.1
19// 
20// #include <msp430.h>
21// 
22int main(void) {
23    WDTCTL = WDT_PWDN | WDT_TMR0;
24    P1DIR |= 0x01;
25    P1OUT |= 0x01;
26    for(;;) {
27        volatile unsigned int i; // volatile to prevent optimization
28        P1OUT ^= 0x01; // Toggle P1.0 using exclusive-OR
29        i = 100000;
30        do i--; // Six Delay
31        while(i > 0);
32    }
33}
```

The Registers window shows the following register values:

Name	Value
Core Registers	
Special Function	
ADC10	
System_Clock	
Comparator_A	
Flash	
Mem1_1,2	
Port_3,4	
Timer0_A3	
Timer1_A3	
USCI_A0_UART Mode	

The Stack Dump window shows the following memory dump:

Address	Value
00000000	00000000
00000001	00000000
00000002	00000000
00000003	00000000
00000004	00000000
00000005	00000000
00000006	00000000
00000007	00000000
00000008	00000000
00000009	00000000
0000000A	00000000
0000000B	00000000
0000000C	00000000
0000000D	00000000
0000000E	00000000
0000000F	00000000
00000010	00000000
00000011	00000000
00000012	00000000
00000013	00000000
00000014	00000000
00000015	00000000
00000016	00000000
00000017	00000000
00000018	00000000
00000019	00000000
0000001A	00000000
0000001B	00000000
0000001C	00000000
0000001D	00000000
0000001E	00000000
0000001F	00000000
00000020	00000000
00000021	00000000
00000022	00000000
00000023	00000000
00000024	00000000
00000025	00000000
00000026	00000000
00000027	00000000
00000028	00000000
00000029	00000000
0000002A	00000000
0000002B	00000000
0000002C	00000000
0000002D	00000000
0000002E	00000000
0000002F	00000000
00000030	00000000
00000031	00000000
00000032	00000000
00000033	00000000
00000034	00000000
00000035	00000000
00000036	00000000
00000037	00000000
00000038	00000000
00000039	00000000
0000003A	00000000
0000003B	00000000
0000003C	00000000
0000003D	00000000
0000003E	00000000
0000003F	00000000
00000040	00000000
00000041	00000000
00000042	00000000
00000043	00000000
00000044	00000000
00000045	00000000
00000046	00000000
00000047	00000000
00000048	00000000
00000049	00000000
0000004A	00000000
0000004B	00000000
0000004C	00000000
0000004D	00000000
0000004E	00000000
0000004F	00000000
00000050	00000000
00000051	00000000
00000052	00000000
00000053	00000000
00000054	00000000
00000055	00000000
00000056	00000000
00000057	00000000
00000058	00000000
00000059	00000000
0000005A	00000000
0000005B	00000000
0000005C	00000000
0000005D	00000000
0000005E	00000000
0000005F	00000000
00000060	00000000
00000061	00000000
00000062	00000000
00000063	00000000
00000064	00000000
00000065	00000000
00000066	00000000
00000067	00000000
00000068	00000000
00000069	00000000
0000006A	00000000
0000006B	00000000
0000006C	00000000
0000006D	00000000
0000006E	00000000
0000006F	00000000
00000070	00000000
00000071	00000000
00000072	00000000
00000073	00000000
00000074	00000000
00000075	00000000
00000076	00000000
00000077	00000000
00000078	00000000
00000079	00000000
0000007A	00000000
0000007B	00000000
0000007C	00000000
0000007D	00000000
0000007E	00000000
0000007F	00000000
00000080	00000000
00000081	00000000
00000082	00000000
00000083	00000000
00000084	00000000
00000085	00000000
00000086	00000000
00000087	00000000
00000088	00000000
00000089	00000000
0000008A	00000000
0000008B	00000000
0000008C	00000000
0000008D	00000000
0000008E	00000000
0000008F	00000000
00000090	00000000
00000091	00000000
00000092	00000000
00000093	00000000
00000094	00000000
00000095	00000000
00000096	00000000
00000097	00000000
00000098	00000000
00000099	00000000
0000009A	00000000
0000009B	00000000
0000009C	00000000
0000009D	00000000
0000009E	00000000
0000009F	00000000
000000A0	00000000
000000A1	00000000
000000A2	00000000
000000A3	00000000
000000A4	00000000
000000A5	00000000
000000A6	00000000
000000A7	00000000
000000A8	00000000
000000A9	00000000
000000AA	00000000
000000AB	00000000
000000AC	00000000
000000AD	00000000
000000AE	00000000
000000AF	00000000
000000B0	00000000
000000B1	00000000
000000B2	00000000
000000B3	00000000
000000B4	00000000
000000B5	00000000
000000B6	00000000
000000B7	00000000
000000B8	00000000
000000B9	00000000
000000BA	00000000
000000BB	00000000
000000BC	00000000
000000BD	00000000
000000BE	00000000
000000BF	00000000
000000C0	00000000
000000C1	00000000
000000C2	00000000
000000C3	00000000
000000C4	00000000
000000C5	00000000
000000C6	00000000
000000C7	00000000
000000C8	00000000
000000C9	00000000
000000CA	00000000
000000CB	00000000
000000CC	00000000
000000CD	00000000
000000CE	00000000
000000CF	00000000
000000D0	00000000
000000D1	00000000
000000D2	00000000
000000D3	00000000
000000D4	00000000
000000D5	00000000
000000D6	00000000
000000D7	00000000
000000D8	00000000
000000D9	00000000
000000DA	00000000
000000DB	00000000
000000DC	00000000
000000DD	00000000
000000DE	00000000
000000DF	00000000
000000E0	00000000
000000E1	00000000
000000E2	00000000
000000E3	00000000
000000E4	00000000
000000E5	00000000
000000E6	00000000
000000E7	00000000
000000E8	00000000
000000E9	00000000
000000EA	00000000
000000EB	00000000
000000EC	00000000
000000ED	00000000
000000EE	00000000
000000EF	00000000
000000F0	00000000
000000F1	00000000
000000F2	00000000
000000F3	00000000
000000F4	00000000
000000F5	00000000
000000F6	00000000
000000F7	00000000
000000F8	00000000
000000F9	00000000
000000FA	00000000
000000FB	00000000
000000FC	00000000
000000FD	00000000
000000FE	00000000
000000FF	00000000

If you choose “Resume”, the red LED (LED1) will blink.

You can view Variables, Expressions and Registers in the “watch” pane.

If you change the code in the program, you should click “Run -> Suspend” and terminate the debugging by clicking “Run -> Terminate”. In CCS Edit Perspectives, save the modified program and click “Run -> Debug”. This will compile, link and start CCS Debug Perspectives.