PSoC 4 BLE Lab #1: Blinking LED

Group: Gloria Bauman and Kervins Nicolas

Due: September 27th, 2016

Description

This Lab introduces you to the PSoC Creator IDE for developing and programming applications on the PSoC 4 BLE chip. It also introduces you to the Bluetooth Low Energy (BLE) Pioneer Kit.

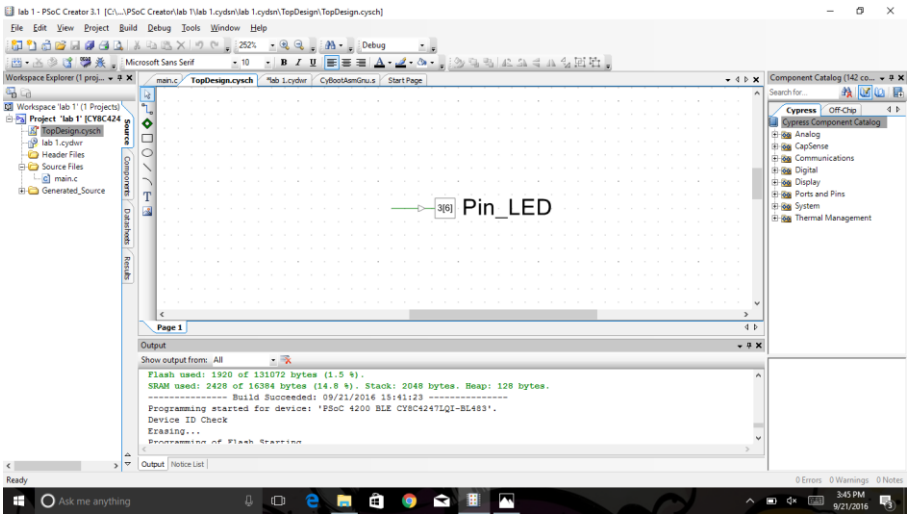
Objectives

1. Learn how to use PSoC Creator to implement and debug PSoC designs
2. Implement a simple blinking LED design

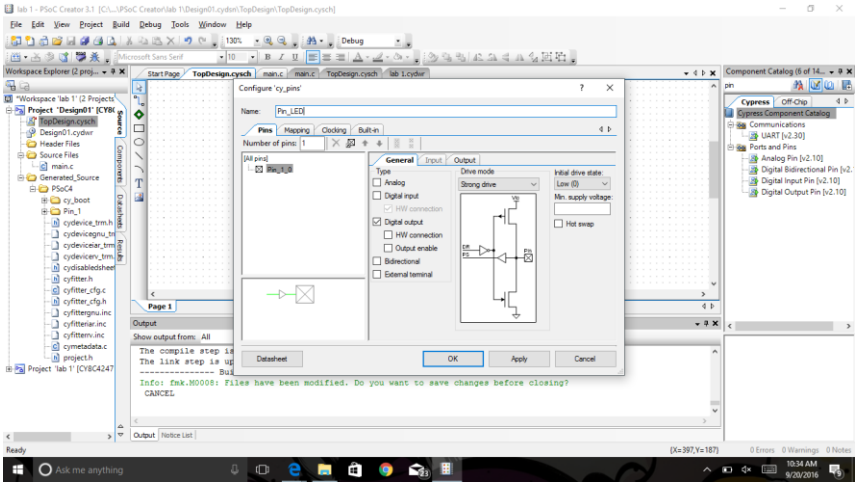
GitHub - https://github.com/kervins/Lab-1

Process

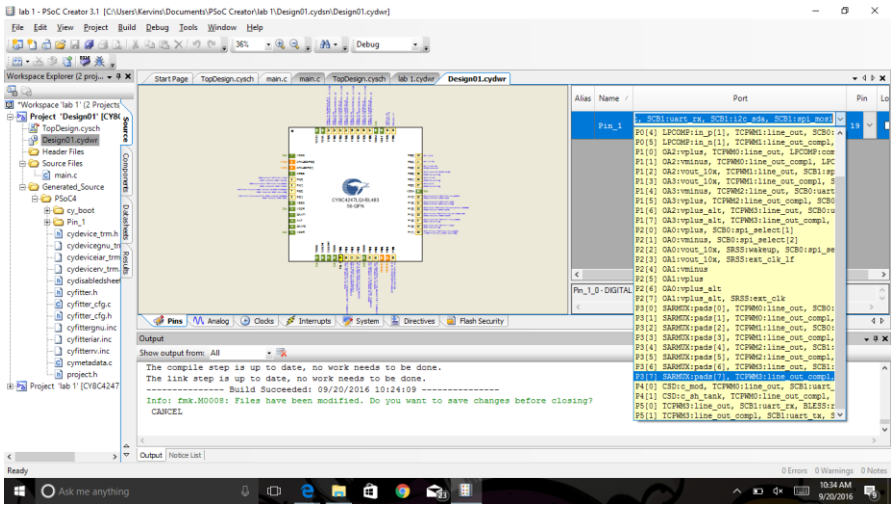
1) Opened a “New Project” and placed an “output_pin” on the schematic diagram. Renamed the “Pin_0” to “Pin_LED.”



2) Configured “Pin_LED” so that there is no hardware attached to it.



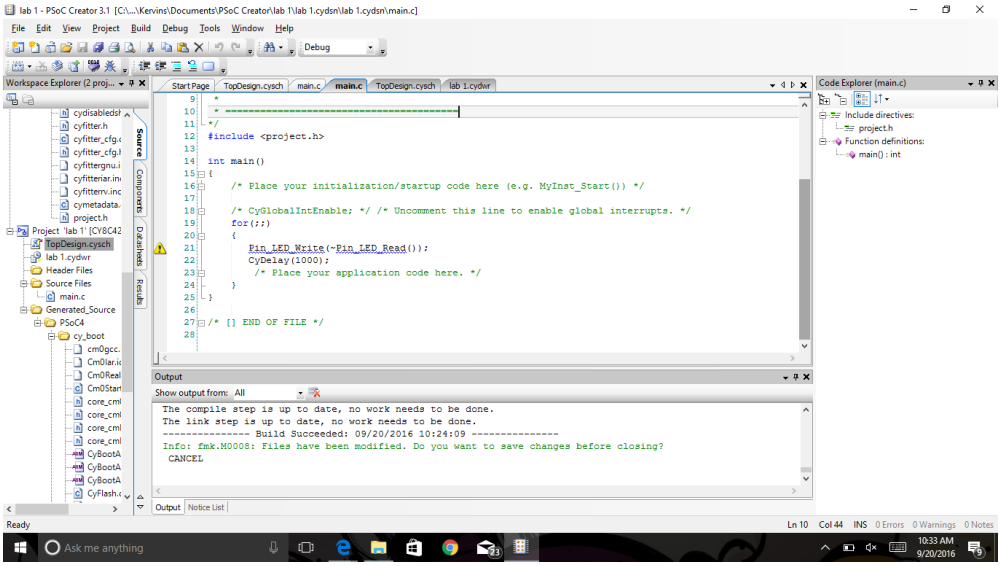
3) Assign “Pin_LED” to ‘P3[7]” in the “Pins” tab.



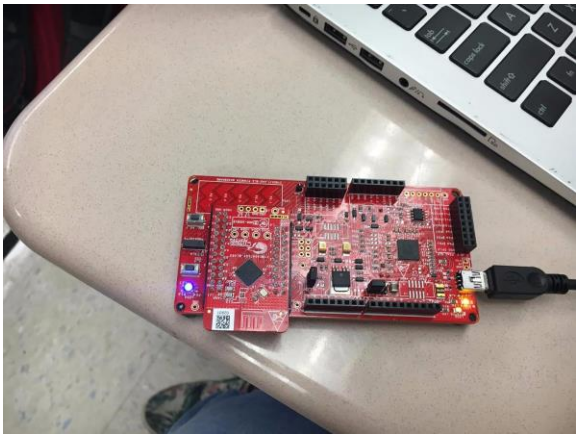
4) Open “main.c” to access the source file in the code editor. Then add in the code inside the “for(::)”

“Pin_LED_Write(~Pin_LED_Read()); //Toggle Pin State”

“CyDelay(1000); //System Delay of 1 second”

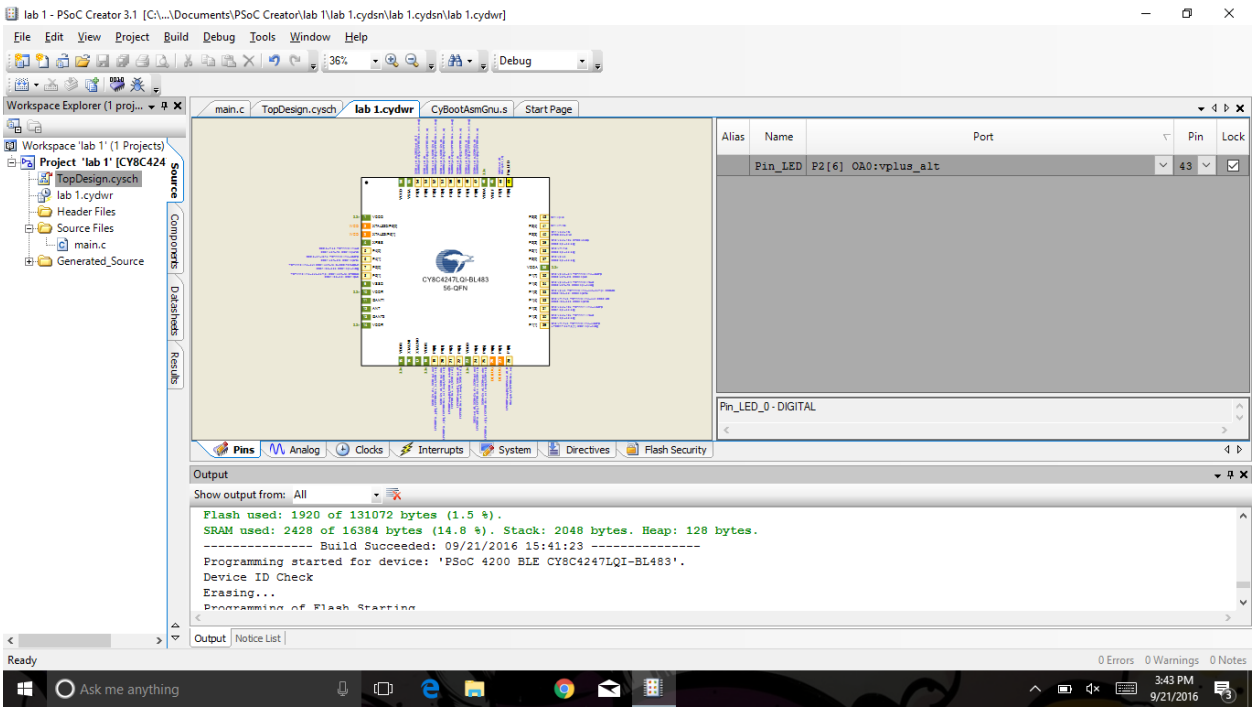


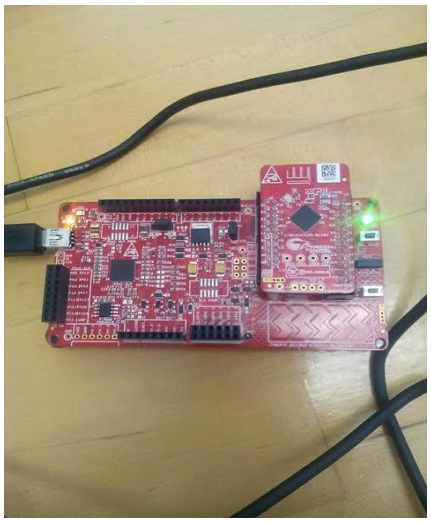
5) Build and Debug your program. After the program is debugged the blue LED on the kit blinks on and off every second.

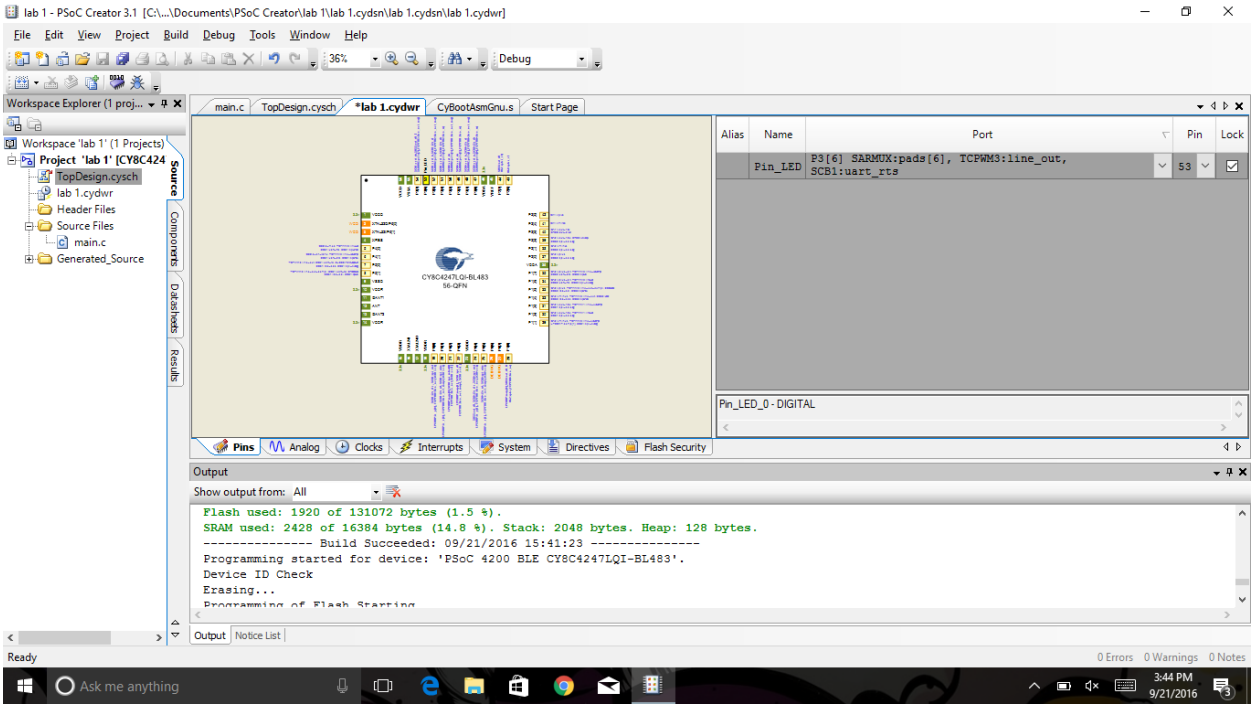


Additional Exercises

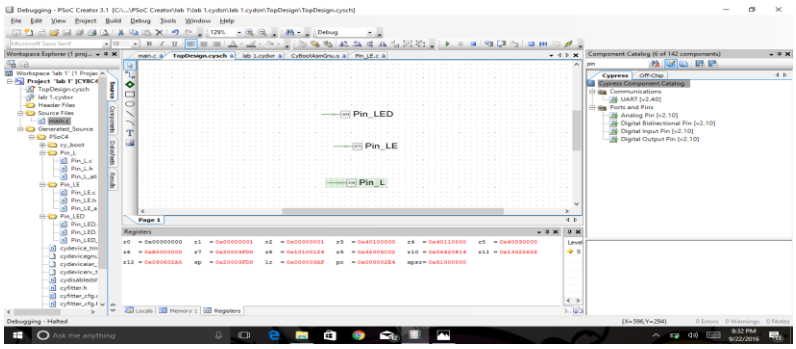
1) Instead of Blue LED, blink Red or Green LED.
    a) Hint: the LED pin-outs are printed on the board.
    - Change the pin LED outputs

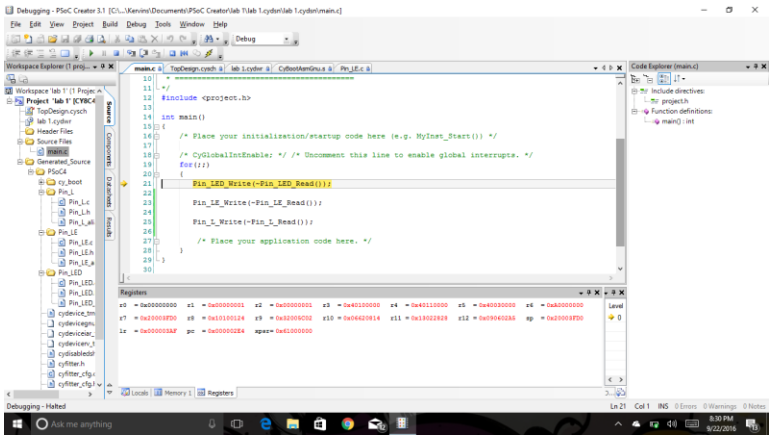


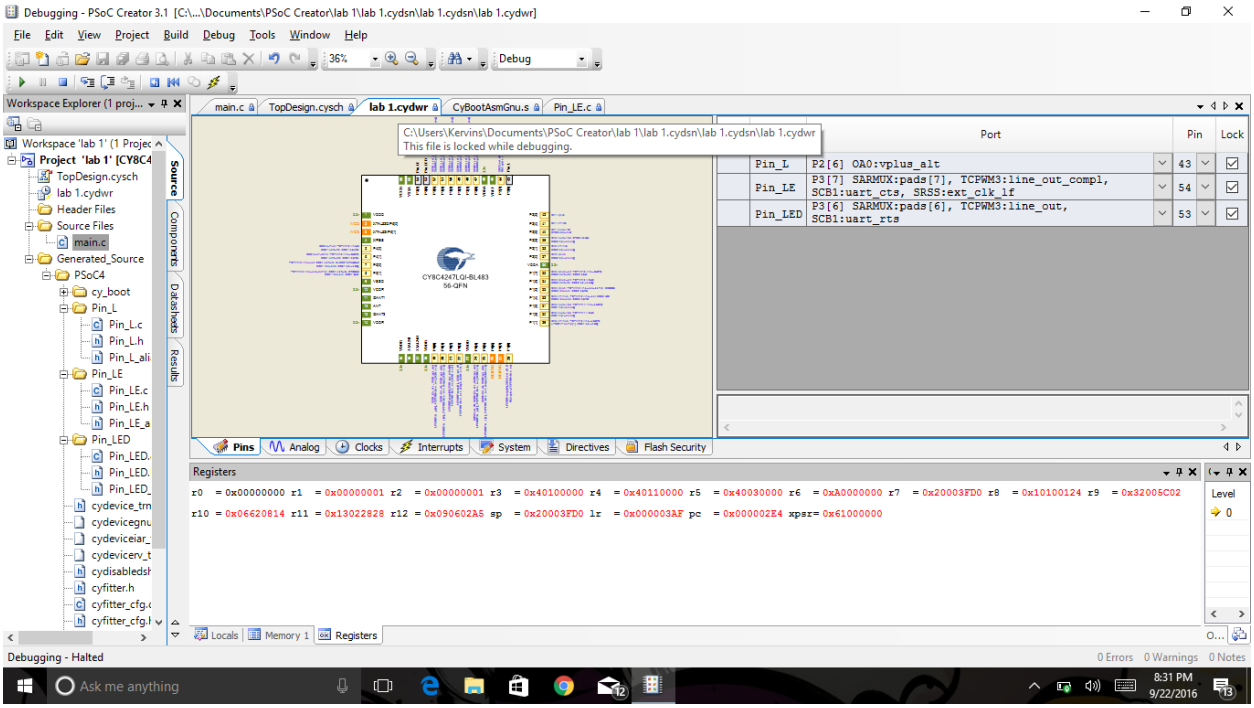


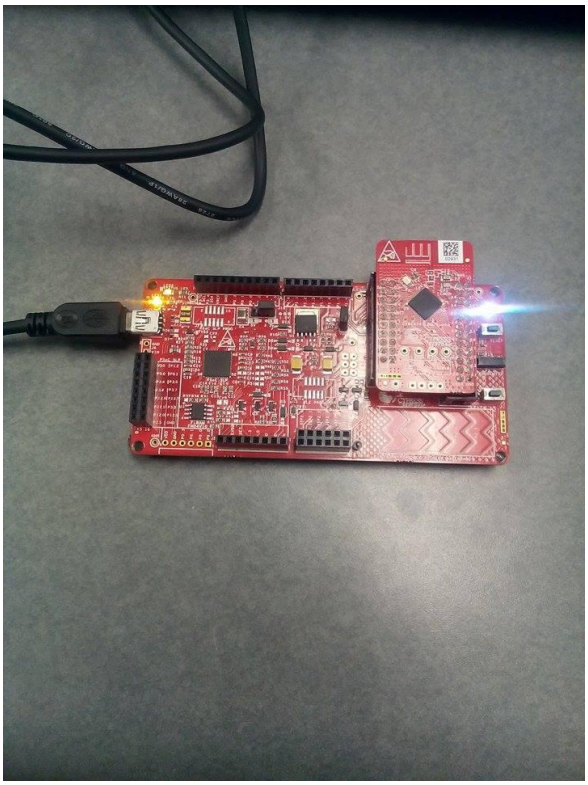

lab 1 - PSoC Creator 3.1 [C:\...\Documents\PSoC Creator\lab 1\lab 1.cydsn\lab 1.cydsn\lab 1.cydwr]
File Edit View Project Build Debug Tools Window Help
Debug
Workspace Explorer (1 proj...
Workspace 'lab 1' (1 Projects)
Project 'lab 1' [CY8C424
TopDesign.cysch
lab 1.cydwr
Header Files
Source Files
main.c
Generated_Source
main.c
TopDesign.cysch
*lab 1.cydwr
CyBootAsmGnu.s
Start Page
CY8C4247LQI-BL483
56-QFN
Alias
Name
Port
Pin
Lock
Pin_LED
P3[6] SARMUX:pads[6], TCPWM3:line_out, SCB1:uart_rts
53
Pin_LED_0 - DIGITAL
Pins
Analog
Clocks
Interrupts
System
Directives
Flash Security
Output
Show output from: All
Flash used: 1920 of 131072 bytes (1.5 %).
SRAM used: 2428 of 16384 bytes (14.8 %). Stack: 2048 bytes. Heap: 128 bytes.
---------- Build Succeeded: 09/21/2016 15:41:23 ----------
Programming started for device: 'PSoC 4200 BLE CY8C4247LQI-BL483'.
Device ID Check
Erasing...
Output
Notice List
Ready
0 Errors 0 Warnings 0 Notes
Ask me anything
3:44 PM
9/21/2016

2) Instead of blinking the LED, generate a constant white color from the RGB LED
   - Combine all of the pins together
   - Created 3 pin: Pin_LED
     Pin_LE
     Pin_L
   - Assigned the pins to a specific output
   - Wrote code in main.c
     Pin_"name"_Write(~Pin_"name"_Read());



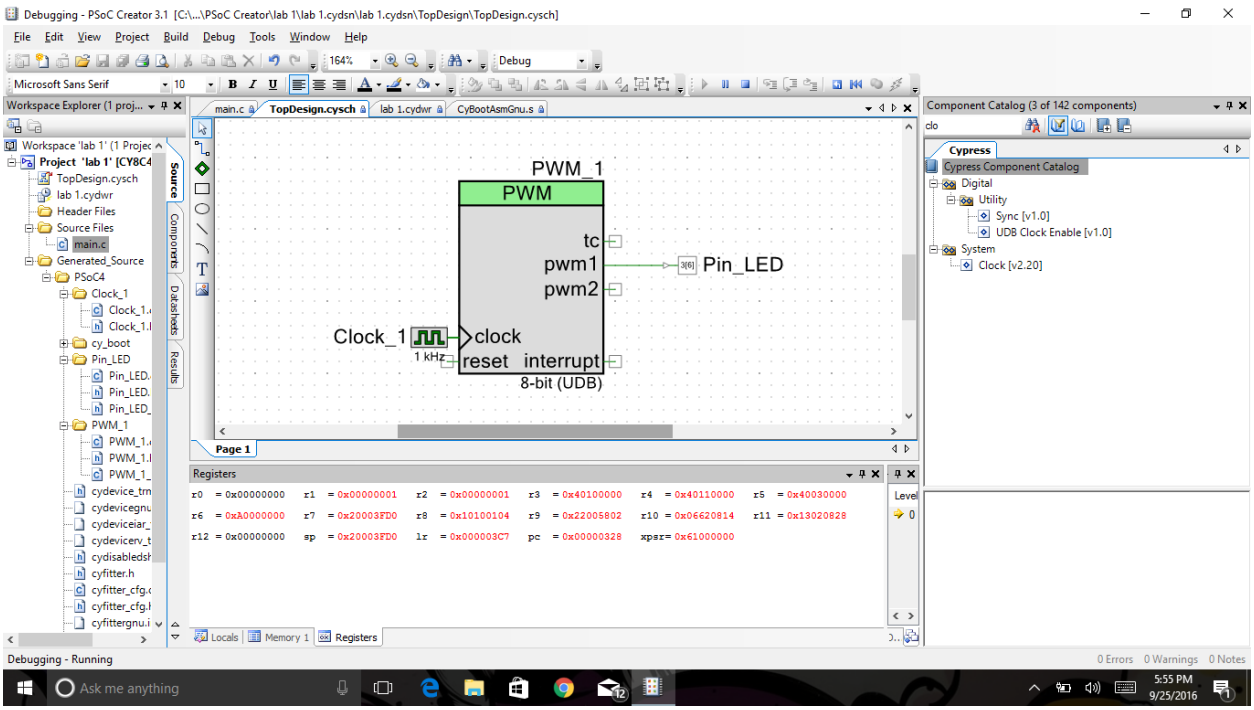



Debugging - PSoC Creator 3.1 [C:\...\Documents\PSoC Creator\lab 1\lab 1.cydsn\lab 1.cydsn\lab 1.cydwr]
File Edit View Project Build Debug Tools Window Help
Workspace Explorer (1 proj...
Workspace 'lab 1' (1 Projec
Project 'lab 1' [CY8C4
TopDesign.cysch
lab 1.cydwr
Header Files
Source Files
main.c
Generated_Source
PSoC4
cy_boot
Pin_L
Pin_L.c
Pin_L.h
Pin_L_ali
Pin_LE
Pin_LE.c
Pin_LE.h
Pin_LE_a
Pin_LED
Pin_LED.
Pin_LED.
Pin_LED_
cydevice_trm
cydevicegnu
cydeviceiar_
cydevicerv_t
cydisabledsh
cyfitter.h
cyfitter_cfg.c
cyfitter_cfg.h
Source
Components
Datasheets
Results
main.c
TopDesign.cysch
lab 1.cydwr
CyBootAsmGnu.s
Pin_LE.c
C:\Users\Kervins\Documents\PSoC Creator\lab 1\lab 1.cydsn\lab 1.cydsn\lab 1.cydwr
This file is locked while debugging.
CY8C4247LQI-BL483
56-QFN
Port
Pin
Lock
Pin_L
P2[6] OA0:vplus_alt
43
Pin_LE
P3[7] SARMUX:pads[7], TCPWM3:line_out_compl, SCB1:uart_cts, SRSS:ext_clk_lf
54
Pin_LED
P3[6] SARMUX:pads[6], TCPWM3:line_out, SCB1:uart_rts
53
Pins
Analog
Clocks
Interrupts
System
Directives
Flash Security
Registers
r0 = 0x00000000 r1 = 0x00000001 r2 = 0x00000001 r3 = 0x40100000 r4 = 0x40110000 r5 = 0x40030000 r6 = 0xA0000000 r7 = 0x20003FD0 r8 = 0x10100124 r9 = 0x32005C02
r10 = 0x06620814 r11 = 0x13022828 r12 = 0x090602A5 sp = 0x20003FD0 lr = 0x000003AF pc = 0x000002E4 xpsr= 0x61000000
Level
0
Locals
Memory 1
Registers
Debugging - Halted
0 Errors 0 Warnings 0 Notes
Ask me anything
8:31 PM
9/22/2016

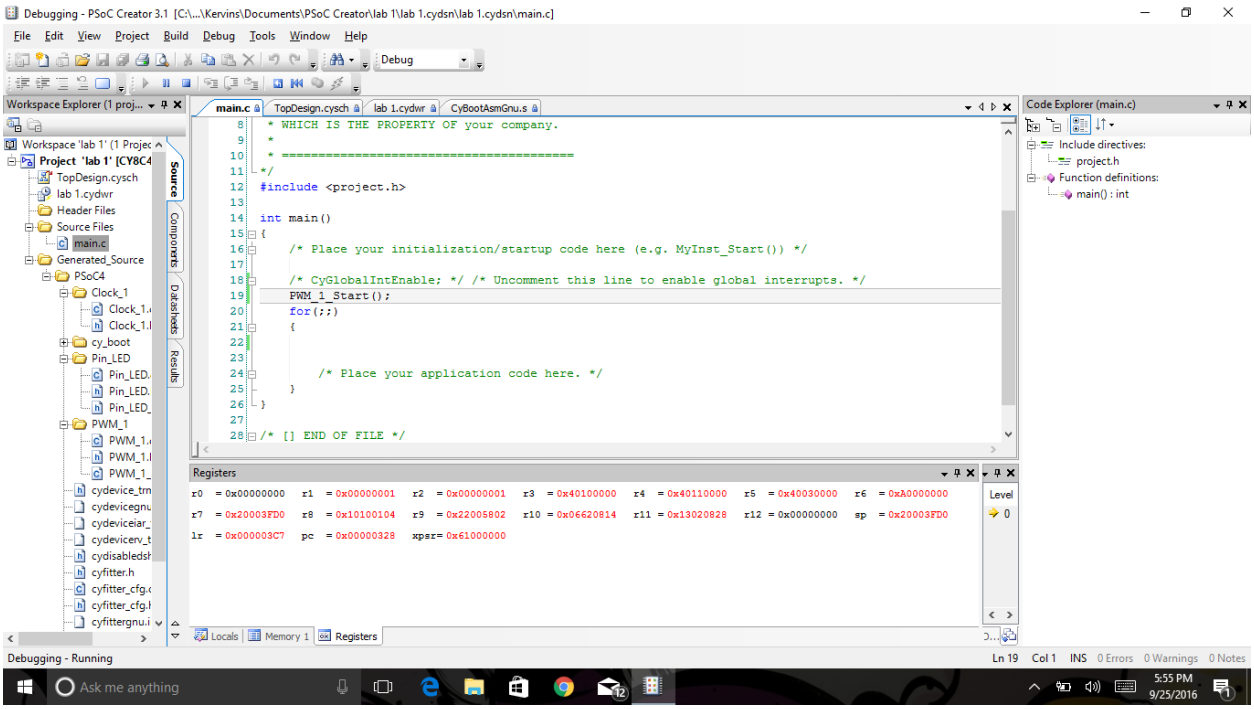


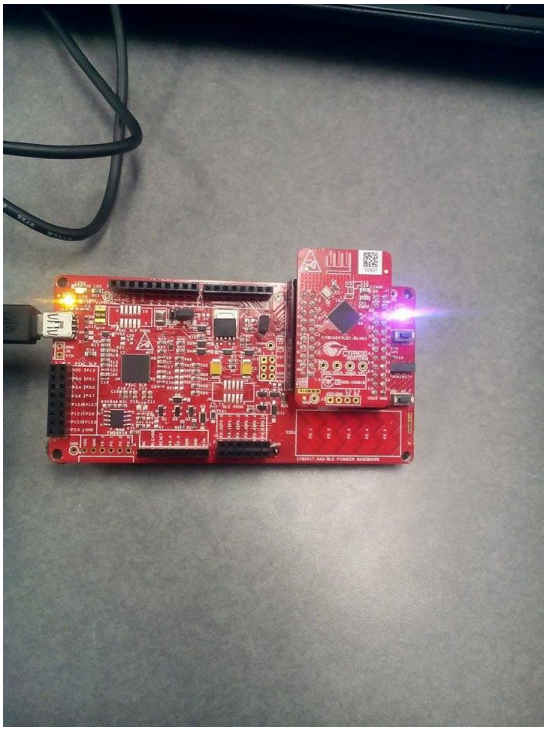
3) Instead of using firmware to blink the LED, use a PWM Component (hardware) to achieve the same result.
    a) Hint: Pick a PWM component from the Cypress Component Catalog. Read the Component data sheet for details on how to configure its parameters. Remember to start the PWM Component in main.c by using the <Component_name>_Start() API. Remember to enable the HW Connection on the pin Component so you can control it via the PWM instead of firmware.

- On the search PWM
- Place it on TopDesign.cysch
- Place a clock of 1 kHz and a Pin output
- In this case it was Pin_LED



- Then finished by implementing the code in the main.c





Code:

```
/* ========================================
 *
 * Copyright YOUR COMPANY, THE YEAR
 * All Rights Reserved
 * UNPUBLISHED, LICENSED SOFTWARE.
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION
 * WHICH IS THE PROPERTY OF your company.
 *
 * ========================================
*/
#include <project.h>
int main()
{
    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    /* CyGlobalIntEnable; */ /* Uncomment this line to enable global interrupts. */
    PWM_1_Start();
    for(;;)
    {


    /* Place your application code here. */
    }
}
/* [] END OF FILE */
```

Conclusion:

This is a basic project for beginner Psoc users to get familiar with the system. By learning how to implement pins into this project, we were able to configure the LED's on the board into what we wanted. Basic coding for the project, debugging and building are all skills that we learned to use for upcoming laboratory work.