# ECE9407– PSoC 4 BLE Lab #1
# Blinking LED

**Name:**     **An He, Amr Gaballah**

**Number:**   **250847871, 250776065**

**Data:**       **2016-06-06**

**Instructor: Dr. Ahmed Hussein**

## 1. Lab1 Task Description

Implement and getting start with PSoC BLE platform. This target of this lab will make us have a quick overview and experience on how to develop and program on PSoC Creator IDE software and PSoC 4 BLE hardware.

## 2. Equipment and Objectives

### Equipment:

Hardware: BLE Pioneer Kit (CY8CKIT-042-BLE)

Software : PSoC Creator 3.3 SP1

### Objectives:

Learn how to get start with PSoC hardware and software

Implement blinking LED API and C program(Firmware)

### Prepare Tasks:

Look the training videos from Cypress Website

http://www.cypress.com/training/free-online-video-training-and-tutorials-cypress

Scan and research cypress products documents

## 3. Overview and Diagram
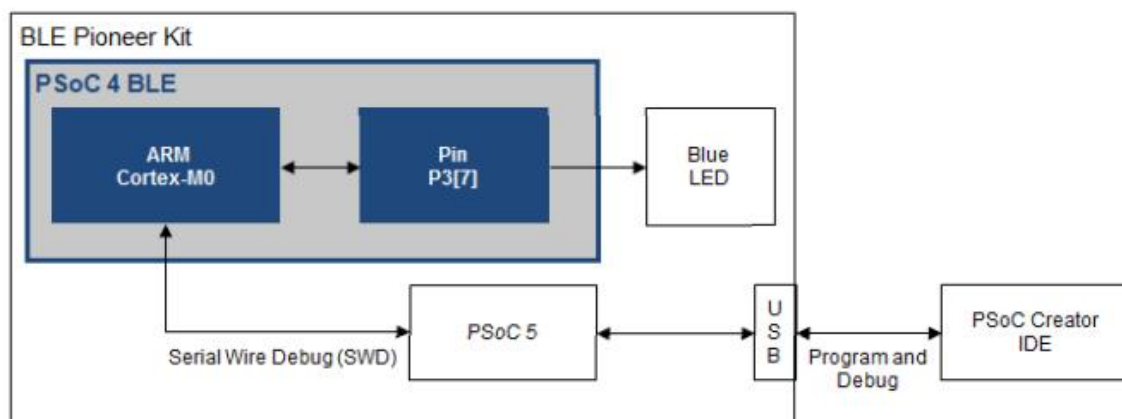


Figure 1: Lab #1 Block Diagram Overview

## 4. Steps and Results

## 4.1 Step1. Create Progect
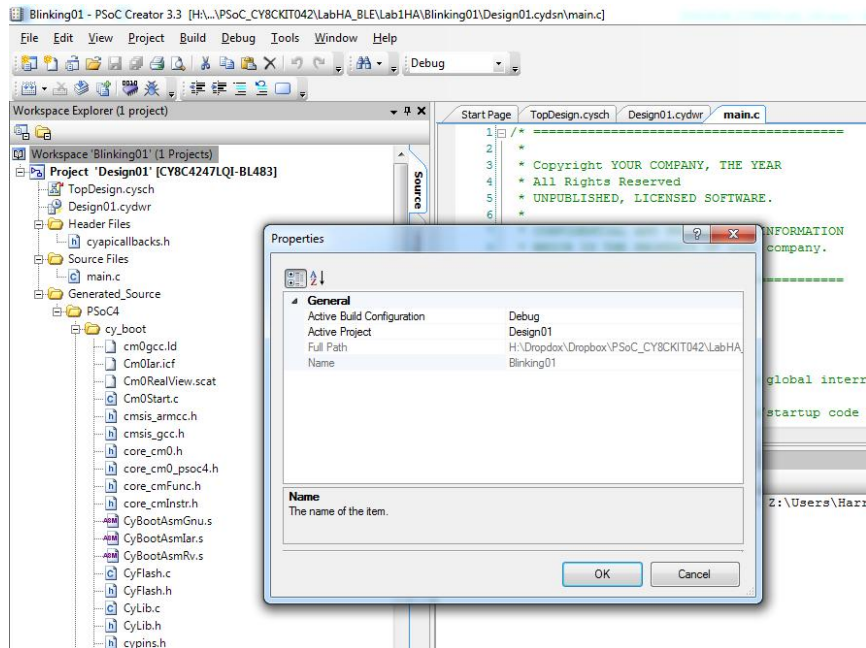


## 4.2 Step2. Locate Digital Output Pin in "Top Design Chart" and Setting the parameter in Component Configuration Tool.
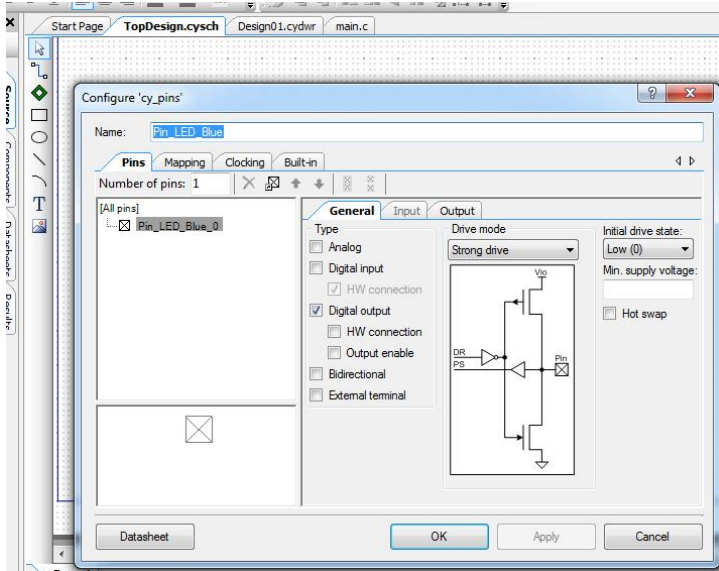
## 4.3 Step3. Configure the Design Wide Resources (DWR) through *.cydwr file
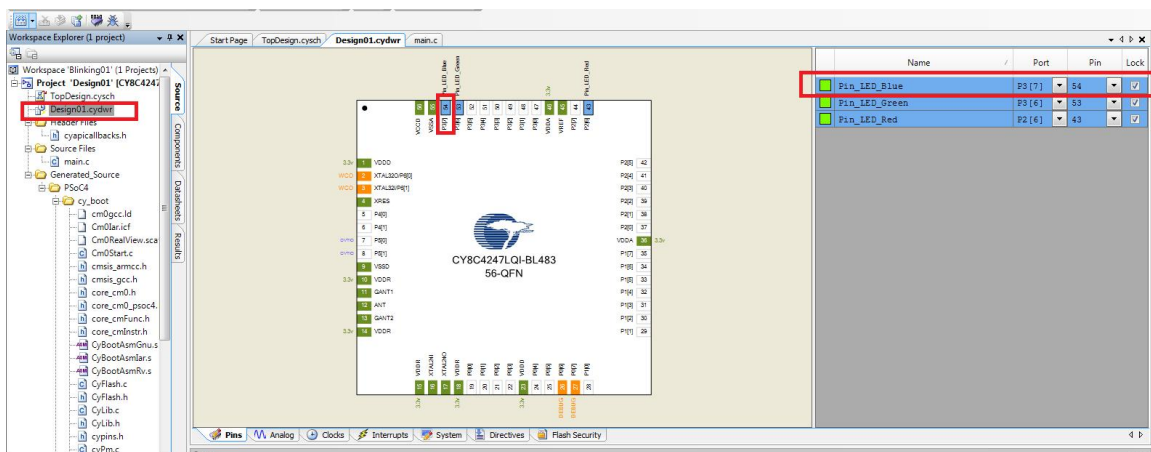


## 4.4 Step4. Write code as firmware

Add the code inside the for(;;) infinite loop to toggle the LED pin by every second using Pin_'name'_Read/Write and CyDelay() command.

The following screenshot includes the additional exercises code, which can execute blink and constant white color from the RGB LED.

```
Pin_LED_Red_Write(255);
Pin_LED_Green_Write(255);
Pin_LED_Blue_Write(255);
```
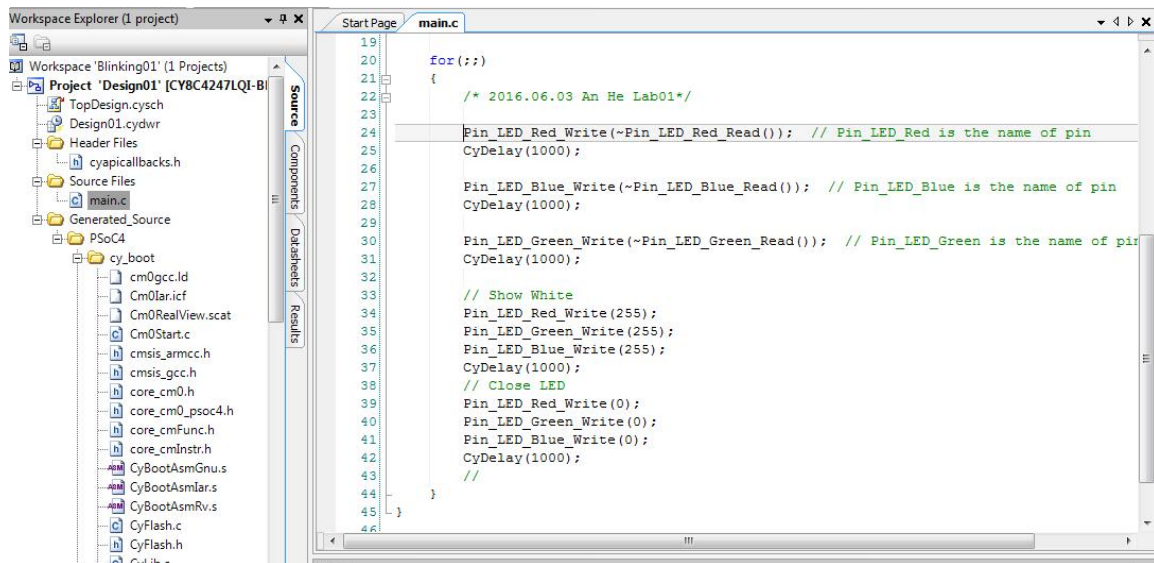
4.5 Step5. Compile, Program and Debug the project

Though item Build -> Build "Project Name", the whole project will be compiled.

Though item Debug->Program, the complied project will be downloaded into hardware

Though item Debug->Debug, the program, PSoc hardware will start as the debug status.

The 'step over' function is very useful when we check program problem and commissioning.

Top screenshot menu and code:

```
Build Design01        Shift+F6          Debug
Clean Design01
Clean and Build Design01
Cancel Build          Ctrl+Break
Compile File          Ctrl+F6
Generate Application
Generate Project Datasheet
```

main.c

```
/* 2016.06.03 An He Lab01*/

Pin_LED_Red_Write(~Pin_LED_Red_Read());  // Pin_LED_Red is the name of pin
CyDelay(1000);
```

Show output from: All

```
arm-none-eabi-gcc.exe -mcpu=cortex-m0 -mthumb -Wno-main -I. -IGenerated_Source\PSoC4 -Wa,-alh=.
arm-none-eabi-gcc.exe -mcpu=cortex-m0 -mthumb -Wno-main -I. -IGenerated_Source\PSoC4 -Wa,-alh=.
arm-none-eabi-gcc.exe -mcpu=cortex-m0 -mthumb -Wno-main -I. -IGenerated_Source\PSoC4 -Wa,-alh=.
arm-none-eabi-as.exe -mcpu=cortex-m0 -mthumb -I. -IGenerated_Source\PSoC4 -alh=.\CortexM0\ARM_G
arm-none-eabi-gcc.exe -mcpu=cortex-m0 -mthumb -Wno-main -I. -IGenerated_Source\PSoC4 -Wa,-alh=.
arm-none-eabi-gcc.exe -mcpu=cortex-m0 -mthumb -Wno-main -I. -IGenerated_Source\PSoC4 -Wa,-alh=.
arm-none-eabi-gcc.exe -mcpu=cortex-m0 -mthumb -Wno-main -I. -IGenerated_Source\PSoC4 -Wa,-alh=.
arm-none-eabi-gcc.exe -mcpu=cortex-m0 -mthumb -Wno-main -I. -IGenerated_Source\PSoC4 -Wa,-alh=.
arm-none-eabi-gcc.exe -mcpu=cortex-m0 -mthumb -Wno-main -I. -IGenerated_Source\PSoC4 -Wa,-alh=.
arm-none-eabi-gcc.exe -mcpu=cortex-m0 -mthumb -Wno-main -I. -IGenerated_Source\PSoC4 -Wa,-alh=.
arm-none-eabi-gcc.exe -mcpu=cortex-m0 -mthumb -Wno-main -I. -IGenerated_Source\PSoC4 -Wa,-alh=.
arm-none-eabi-gcc.exe -mcpu=cortex-m0 -mthumb -Wno-main -I. -IGenerated_Source\PSoC4 -Wa,-alh=.
arm-none-eabi-gcc.exe -mcpu=cortex-m0 -mthumb -Wno-main -I. -IGenerated_Source\PSoC4 -Wa,-alh=.
arm-none-eabi-ar.exe -rs .\CortexM0\ARM_GCC_493\Debug\Design01.a .\CortexM0\ARM_GCC_493\Debug\cy
arm-none-eabi-ar.exe: creating .\CortexM0\ARM_GCC_493\Debug\Design01.a
arm-none-eabi-gcc.exe -Wl,--start-group -o .\CortexM0\ARM_GCC_493\Debug\Design01.elf .\CortexM0
cyelftool.exe -C E:\DropboxHA\Dropbox\PSoC_CY8CKIT042\LabHA_BLE\Lab1HA\Blinking01\Design01.cydsr
cyelftool.exe -S E:\DropboxHA\Dropbox\PSoC_CY8CKIT042\LabHA_BLE\Lab1HA\Blinking01\Design01.cydsr
Flash used: 2048 of 131072 bytes (1.6 %).
SRAM used: 2476 of 16384 bytes (15.1 %). Stack: 2048 bytes. Heap: 128 bytes.
--------------- Build Succeeded: 06/06/2016 23:05:09 ---------------
```
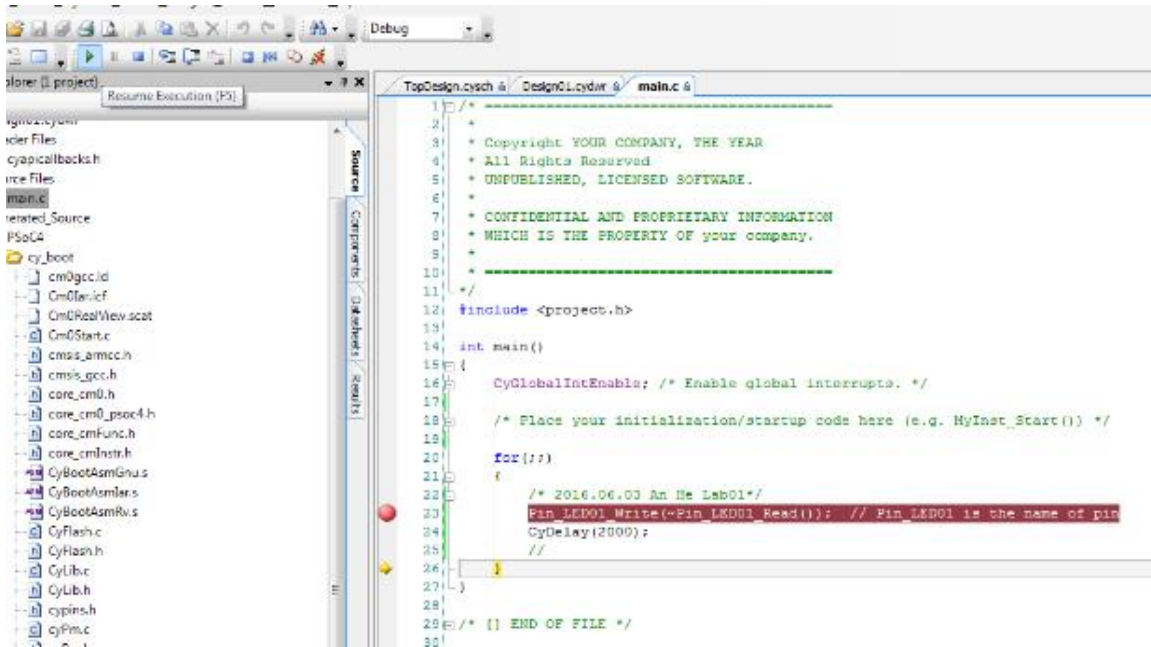
Workspace Explorer (1 project)

```
Workspace 'Blinking01' (1 P
Project 'Design01' [CY
  TopDesign.cysch
  Design01.cydwr
  Header Files
    cyapicallbacks.h
  Source Files
    main.c
  Generated_Source
    PSoC4
      cy_boot
        cm0gcc.ld
        Cm0Iar.icf
        Cm0RealView.scat
        Cm0Start.c
        cmsis_armcc.h
        cmsis_gcc.h
        core_cm0.h
        core_cm0_psoc4.h
        core_cmFunc.h
        core_cmInstr.h
        CyBootAsmGnu.s
        CyBootAsmIar.s
        CyBootAsmRv.s
        CyFlash.c
        CyFlash.h
        CyLib.c
        CyLib.h
        cypins.h
        cyPm.c
```



Bottom screenshot:

Blinking01 - PSoC Creator 3.3  [E:\...\PSoC_CY8CKIT042\LabHA_BLE\Lab1HA\Blinking01\Design01.cydsn\main.c]

File  Edit  View  Project  Build  Debug  Tools  Window  Help

Debug menu:
```
Windows                                    ►
Program                        Ctrl+F5
Select Debug Target...
Debug                          F5
Debug without Programming      Alt+F5
Attach to Running Target...
Toggle Breakpoint              F9
New Breakpoint                             ►
Delete All Breakpoints         Ctrl+Shift+F9
Enable All Breakpoints
```

Workspace Explorer (1 project)

```
Workspace 'Blinking01' (1 Projects)
Project 'Design01' [CY8C4247
  TopDesign.cysch
  Design01.cydwr
  Header Files
    cyapicallbacks.h
  Source Files
    main.c
  Generated_Source
    PSoC4
      cy_boot
        cm0gcc.ld
        Cm0Iar.icf
        Cm0RealView.scat
        Cm0Start.c
        cmsis_armcc.h
        cmsis_gcc.h
        core_cm0.h
        core_cm0_psoc4.h
        core_cmFunc.h
        core_cmInstr.h
        CyBootAsmGnu.s
        CyBootAsmIar.s
        CyBootAsmRv.s
        CyFlash.c
        CyFlash.h
        CyLib.c
        CyLib.h
        cypins.h
```

Code area:
```
6.06.03 An He Lab
_Red_Write(~Pin_
v(1000):
```

Output:
```
-mcpu=cortex-m0
-mcpu=cortex-m0
arm-none-eabi-gcc.exe -mcpu=cortex-m0
arm-none-eabi-as.exe -mcpu=cortex-m0 -m
arm-none-eabi-gcc.exe -mcpu=cortex-m0
arm-none-eabi-gcc.exe -mcpu=cortex-m0
arm-none-eabi-gcc.exe -mcpu=cortex-m0
arm-none-eabi-gcc.exe -mcpu=cortex-m0
arm-none-eabi-gcc.exe -mcpu=cortex-m0
arm-none-eabi-gcc.exe -mcpu=cortex-m0
arm-none-eabi-gcc.exe -mcpu=cortex-m0
arm-none-eabi-gcc.exe -mcpu=cortex-m0
arm-none-eabi-gcc.exe -mcpu=cortex-m0
arm-none-eabi-ar.exe -rs .\CortexM0\ARM
arm-none-eabi-ar.exe: creating .\Cortex
arm-none-eabi-gcc.exe -Wl,--start-group
cyelftool.exe -C E:\DropboxHA\Dropbox\B
cyelftool.exe -S E:\DropboxHA\Dropbox\B
Flash used: 2048 of 131072 bytes (1.6 %
SRAM used: 2476 of 16384 bytes (15.1 %)
--------------- Build Succeeded: 06/06/
```

## 5. Summary

The Lab1 gave us a good opportunity to learn how to use I/O, API, C code from PSoC Creator. Though online commissioning, we implement our program into the hardware. Although this lab is easy, we learn the whole procedure. That is very important for the following test and final project.

The issue about LED turns ON when the pin is driven low, and OFF when the pin is driven high should be paid attention. The part setting was in Component Configuration Tool and Component API. The understanding the output circuit logic will make us commissioning more easier.