Goal
This lab introduces you to C programming for an important embedded control skill: Capturing and recording an event

Project Summary
You implement a system which captures and counts the number of 'button pressing (=pushing down)' event. The result is displayed in both LEDs and LCD as follows.

No LED is ON when no button is pressed.
ED1 is ON when only S1 is pressed.
LED2 is ON when only S2 is pressed.
LED3 is ON when both S1 and S2 are pressed together.

At the same time, the number of pressing of each button is counted and displayed on the LCD screen using the following format.

Button Counter
B1=xxxx B2=yyyy

where, xxxx and yyyy are the number of pressings of S1 and S2, respectively, in hex format.

Caution: Make sure that the value displayed in LCD increases only once per each pressing event. It should not increase until the next pressing event occurs (i.e., no increment while holding the button down).

SetUp
1. Open PSoC Designer. Click on File -> New Project. [New Project] window opens. Select Chip-level Project and Type your project name in the Name box.

2. Click Browse to select the directory where you are storing your Lab assignments. Click the check box to create directory for workspace. Click OK.

3. In the next [Select Project Type] window, select part CY8C24894 through the View Catalog… button and choose to Generate Main file using C. Click OK when done.

Adding User Module
4. We’ll now add three LEDs. In the User Module Catalog, expand Misc Digital, Right Click on LED and select Place. Now you should see one user modules added in the Workspace Explorer as “LED_1”.

Ports and Pins
5. So now that we’ve added the hardware we need for this project, we need to connect the hardware to specific pin to give the microcontroller a way to ‘talk’ to the LED to tell it what to do. We will connect the LED_1 to P3[0]. In the Workspace explorer click on LED_1. Its property window should appear. In the drop down menu next to Port select Port 3. In the drop down menu next to Pin select Port3_0.

6. Repeat the similar process to connect LED_2/3 to P3[1]/P3[2], respectively. Do the similar process to add LCD to your design connecting it to Port_4(since it is prewired).

7. Now that we have finished making pin connection for our user module, we still need to make pin connections for our push buttons. Let’s assign S1 to P3[5], S2 to P3[7]. In order to do this double click on Project_name[Pinout] in the Workspace Explorer.

8. In the Pinout diagram that appears, locate P3[5] and click on the blue bar. A window appears that allows you to specify the name and other parameters for this pin. Name this pin ‘PB1’ and change the Drive of this pin to ‘PullDown’. PullDown just means that the button will have a ‘0’ value when not pressed and a ‘1’ value when pressed. Click ‘OK’ in the window to make the change. Do the same thing for S2 with naming P3[7] with ‘PB2’.
Adding Your Embedded C-code

9. Now that we have the hardware hooked up, we can write an embedded C program. As we all know by this time, in the Workspace Explorer, expand the Project_name Folder and Source Files Tab and double click on Main.c.

10. For embedded C-programming tip: All necessary variables and functions were provided except internal part of WHILE LOOP. You need to develop your logic and fill the blank part.

11. Your embedded coding

```c
#include <m8c.h>        // part specific constants and macros
#include "PSoCAPI.h"    // PSOC API definitions for all User Modules

void main()
{
    unsigned char PB1, PB2; //Push Button(S1,S2) inputs, 0 for not pressed, Positive for pressed
    int     cnt1=0, cnt2=0; //number of pressings of S1 and S2
    int    PB1_pressed=0, PB2_pressed=0;  //PB state variable, 0 for not pressed, 1 for pressed

    LED_1_Start();    LED_2_Start();    LED_3_Start();  // LED initialization
    LED_1_Switch(0);  LED_2_Switch(0);  LED_3_Switch(0);// let's turn all LEDs OFF at start

    LCD_1_Start();    LCD_1_Init();                     // LCD Initialization- common
    LCD_1_Position(0,0); LCD_1_PrCString("Button Counter");  // LCD screen format
    LCD_1_Position(1,0); LCD_1_PrCString("B1=0000");        // initial value
    LCD_1_Position(1,8); LCD_1_PrCString("B2=0000");        // initial value

    while (1) {
        //-------------------- Beginning of your code -------------------------
        Your code here...
        //-------------------- The END of Your code ---------------------------
    } // end while
} // end main
```

Building and Programming Your Project on PSOC board

12. See the previous LAB documents for this step.

Wiring information

13. PB1(s1-p35), PB2(s2-p37), LED1/2/3 - p30/31/32/33, respectively