EVB Pin

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	38
39	40
41 ⟷ 60	

	Port Bit	Bit Addresses & Labels
1.		
2.		
3.		
4.		
5.		
6.		
7.		
8.		
9.		
10.		
11.		
12.		
13.		
14.		
15.		
16.		
17.		
18.		
19.		
20.		
21.		
22.		
23.		
24.		
25.		
26.		
27.		
28.		
29.	2.0	0xA0 SS
30.		
31.	3.6	0xB6 LED0
32.	3.7	0xB7 Buzzer
33.	3.4	0xB4 BILED2
34.	3.5	0xB5 LED1
35.		
36.	3.3	0xB3 BILED1
37.	3.1	0xB1 PB1
38.	3.0	0xB0 PB0
39.		
40.		

Software Initializations

A) Port I/0

P3MDOUT &= ~0x03

P3MDOUT |= 0xF8

P3 = 0x03

P2MDOUT &= ~0x01

P2 |= 0x01

B) Timers

CKCON |= 0x08

TMOD &= 0xF0

TMOD |= 0x01

TR0 = 0

TMR0 = 0

C) Interrupts

IE |= 0x82

D) A/D

E) PCA

F) XBAR

G) I2C

Laboratory Worksheet #05
Timer Overflow Interrupts Exercise

The following is an exercise on Timer Overflow interrupts and will serve as a good starting point for Lab 1-2. The hardware used for this activity should already be ready on your board from Lab 1-1.

Exercise 1:

1) On what page of the LITEC manual does the discussion of Timer Functions start? 34

Utilizing the timer on the microcontroller requires initializing and interacting with a number of SFRs. Based on the following descriptions, identify which SFR is indicated.

2) The high byte of Timer0 TH0

3) The SFR which contains a bit to turn the Timer0 on/off TCON

4) The SFR that sets Timer0 to 16 bit mode TMOD

5) What is the frequency of the system clock (to 6 decimal places)? 22.1184

Exercise 2:

The program configures Timer0 to use SYSCLK/12 as its source and in a 13-bit mode.

1) Complete the initialization code that follows. Note: this example code is only part of the full initialization routine. The indicated initial settings of the SFRs are here for example.

```
//current state of TMOD is xxxx 1111
//current state of CKCON is xxxx xxxx
TMOD &= 0xF0
CKCON &= 0xF7
```

2) How much time (in seconds) is required for a timer overflow interrupt (assume the Timer is initialized to 0)? .00889

3) How many overflows will occur in 1 second? 112

4) Assume a variable counts keeps track of the number of overflows. After 2.5 seconds, what is the value of counts? (assume counts is zero a 0 seconds) 281

Exercise 3:

The sample code, *Worksheet_05.c*, is available on the LMS website under the "Laboratories" section, under Lab 1, part 2. You need to complete the initialization routines for the Port I/O, the Timer, and Interrupts. Once you have done that, compile, link, download and run this program. This program counts the number of timer overflows occurring while the slide switch is in the Off position.

Exercise 4:

In the *Worksheet_05.c* code, the variable *counts* keeps track of the number of timer overflows and this value is printed on the terminal.

1) Modify the printf statement to also print out the corresponding time period in seconds.
2) Using a handy watch or online clock, turn the switch to the 'count' position for 10 seconds and compare the accuracy of the counter to the time you measure.

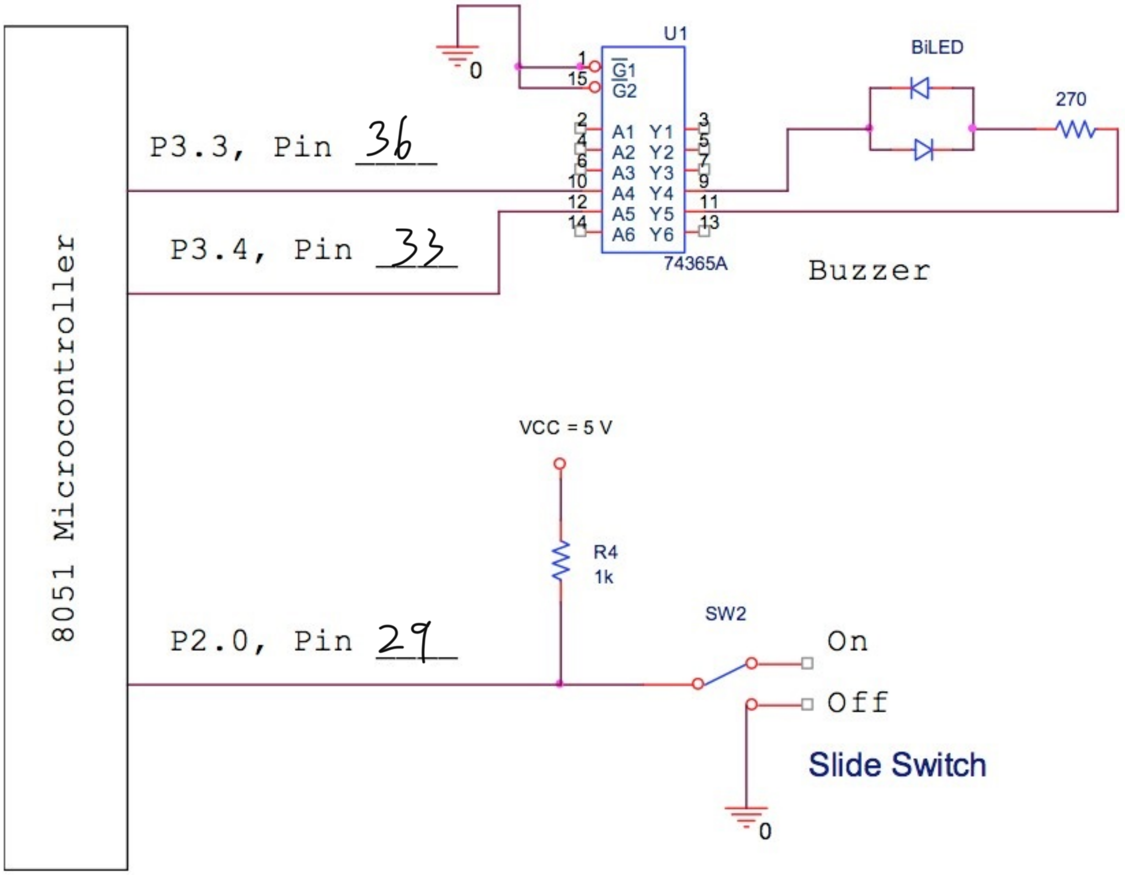

Figure 1: Worksheet 3 Schematic

When complete, include Worksheet 5 with your Laboratory 1-2 Pre-lab submission.

```
/* This program demonstrates the use of T0 interrupt. The code will count the
number of T0 timer overflows that occur while a slide switch is in Off position
Some editing is requited prior to running the code. Fill in the indicated blanks.
*/

#include <c8051_SDCC.h> // include files. You need to include stdio and c8051_SCDD.
#include <stdio.h>      // Add lines as needed

//-----------------------------------------------------------
// Function PROTOTYPES
//-----------------------------------------------------------
void T0_ISR (void) __interrupt 1; // Function Prototype for Interrupt Service Routine
void Port_Init(void);  // Initialize ports for input and output
void Timer_Init(void); // Initialize Timer 0
void Interrupt_Init(void);
void Counter_off(void);
void Counter_on(void); //

//-----------------------------------------------------------
// Global variables
//-----------------------------------------------------------

// one end of bicolor LED0 is associated with Port 3 Pin 3
__sbit __at 0xB3 Biled1;
// other end of bicolor LED0 is associated with Port 3 Pin 4
__sbit __at 0xB4 Biled2;
__sbit __at 0xA0 SW; // Slide Switch associated with Port 2 Pin 0

unsigned int Counts = 0;
```

```
//***************
void main(void)
{
  Sys_Init();   // System Initialization Always do this first.
  putchar(' ');  // line added to allow printf statements
  Port_Init();  // Initialize port 2 and 3
  Timer_Init();  // Initialize Timer 0
  Interrupt_Init();

  printf("Start\r\n");
  while (1)
  {
    Counter_off();
    Counter_on();
  }
}


void Port_Init(void)
{

  // Port 3
  P3MDOUT |= 0x18; // set output pins P3.3 and P3.4 in push-pull mode

  // Port 2
  P2MDOUT &= ~0x01; // set input pin P2.0 in open drain mode
  P2 |= 0x01; // set input pin P2.0 to high impedance state
}
```

```
void Interrupt_Init(void)
{
   IE |= 0x82; //enable Timer0 interrupts by setting the appropriate bit in the SFR
   EA = 1;  //enable all interrupts using an existing sbit label
}


void Timer_Init(void)
{
   CKCON &= ~0x08;  // Make T1 intact and T0 use SYSCLK/12
   TMOD &= 0xF0;   // Clear the 4 least significant bits
   TMOD |= 0x00;   // Leave T1 intact and set T0 mode 13bit
   TR0 = 0;    // Stop Timer0
// 2 ways to clear 16-bit T0 counter: use a single command for all 16 bits
   TMR0 = 0;    // Clear both bytes of T0
// or use 2 commands for low and high bytes separately
   //TL0 _____________;    // Clear low byte of register T0
   //TH0 _____________;    // Clear high byte of register T0
}

void T0_ISR ( void ) __interrupt 1 //Interrupt service routine
{
   TF0 = 0;  // clear interrupt request (not required - cleared automatically by hardware)
   Counts++; // increment overflow counter
}

void Counter_off(void) // turn the BILED off and stop the counter
{
   TR0 = 0;    // turn off the counter
   Counts = 0; // reset counts to 0
```

```
    Biled1 = 0; //on
    Biled2 = 0; //on
    TL0 = 0x00;
    TH0 = 0x00; // initialize the Timer to a 0 start value
    while(SW);  // while the switch is off, wait
}

void Counter_on(void) // turn the BILED on and count how long it the switch is on
{
    Biled1 = 1;
    Biled2 = 0;
    TR0 = 1;    // start the counter
    while(!SW); // while the switch is on, wait
    printf("\rNumber of Overflows = %d\n", Counts);
}
```

LAB 1.2

```
Compiler directives
    #include <c8051_SDCC.h>
    #include <stdio.h>
    #include <stdlib.h>
Function Prototypes
    Void Port_Init(void)
    Void Timer_Init(void)
    Void Interrrupt_Init(void)
    Void Timer0_ISR(void) __interrupt 1
    Unsigned char random(void)
Declare global variables
    Sbit PB0, PB1, SS, LED0, LED1, BILED1, BILED2, BUZZER
Main function
    Declare local variables
        (none)
    Initialization functions
        Sys_Init()
        putchar(‘ ‘)
        Port_init()
        Interrupt_Init()
        Timer_Init()

    Begin infinite loop
        Int temp=-1;
        While(TRUE)
            Turn off LEDs and buzzer
            if(SS is ON)
                While (less than 10 times)
                    Enable timer1
                    Int ran;
                    ran=random(0,2);

                    While (ran==temp)
                        an=random(0,2)

                    temp=ran;

                    If (ran==0)
                        LED0 is lit

                    Else If (ran==1)
                        LED1 is lit
                    Else If (ran==2)
                        Both LED0 and LED are lit

                    Reset counts and TIMR0

                    If (LED0 lit && PB0 pushed)
                        Green LED lit and score++
                    Else if (LED1 lit && PB1 pushed)
                        Green LED lit and score++
                    Else if (LED0 lit && LED1 lit && PB0 pushed && PB1
pushed )

                        Green LED lit and score++
                    Else
                        Red LED lit
                    Reprat until 10 times
    End infinite loop
End main function

Functions
    Void Port_Init(void)
        Set SFRs P2, P3, and P2MDOUT and P3MD
        So that P2.0, P3.0, P3.1 are inputs,
            P3.3, P3.4, P3.6, P3.7 are outputs
    End Port_Init

    Void Set_outputs(void)
        If SS is off
            BILED is green, and all others are off
            Print “slide switch is off”
        Else
            Print “slide switch is on”
             if only PB1 is on
                BUZZER is on
                Print “push button 1 is activated”
            Else if only PB2 is on
                LED0 is on
                Print “push button 2 is activated”
            Else if both PB1 and PB2 is on
                BILED is red
                Print “ push button 1 and 2 are both activated”
            Else ( PB1 and PB2 are both not pushed)
```

```
                BILED and all others are off
                Print nothing

End Set_Outputs
```