EVB Pin		Port Bit	Bit Addresses & Labels
1	2	1.	
		2.	
3	4	3.	
		4.	
5	6	5.	
		6.	1.4	0x94 Potentiometer
7	8	7.	
		8.	
9	10	9.	
		10.	
11	12	11.	
		12.	
13	14	13.	
		14.	
15	16	15.	
		16.	
17	18	17.	
		18.	
19	20	19.	
		20.	
21	22	21.	
		22.	
23	24	23.	
		24.	
25	26	25.	
		26.	
27	28	27.	
		28.	
29	30	29.	2.0	0xA0 SS
		30.	
31	32	31.	3.6	0xB6 LED0
		32.	3.7	0xB7 Buzzer
33	34	33.	3.4	0xB4 BILED2
		34.	3.5	0xB5 LED1
35	36	35.	
		36.	3.3	0xB3 BILED1
37	38	37.	3.1	0xB1 PB1
		38.	3.0	0xB0 PB0
39	40	39.	
		40.	
41 ⟷ 60			

Software Initializations

A) Port I/0

```
P3MDOUT &= ~0x03
P3MDOUT |= 0xF8
P3 = 0x03
P2MDOUT &= ~0x01
P2 |= 0x01
P1MDIN &= ~0x10
P1MDOUT &= ~0x10
P1 |= 0x10
```

B) Timers

```
CKCON |= 0x08
TMOD &= 0xF0
TMOD |= 0x01
TR0 = 0
TMR0 = 0
```

C) Interrupts

```
IE |= 0x82
```

D) A/D

```
REF0CN = 0x03
ADC1CN = 0x80
ADC1CF |= 0x01
```

E) PCA

F) XBAR

G) I2C

Lab2

Complier directives

```
#include<c8051_SDCC.h>
#include <stdio.h>
#include<stdlib.h>
```

Function Prototypes

```
Void Port_Init(void);
Void Timer_Init(void);
Void Interrupt_Init(void);
Void Timer0_ISR(void) __interrupt 1;
Void ADC_Init(void)
Unsigned char read_AD_input(unsigned char n)
Unsigned char random(unsigned char speed)
Void mode0(void)
Void mode1(void)
Void mode2 (void)
Void mode3 (void)
```

Global variables

```
Sbit PB0 PB1 PB2 PB3 SS0 SS1 LED0 LED1 LED2 LED3 BILED0 BILED1 BUZZER
```

Main function

```
Declare local variables
        (none)
Initialization functions
        Sys_Init()
        Putchar(‘ ’)
        Port_Init()
        Interrupt_Init()
        Timer_Init()
        ADC_Init()
        Begin infinite loop
            If(ss0 is off and ss1 is off)
                Mode0()
            Else If (ss0 is off and ss1 is on)
                Mode1()
            Else if (ss0 is on and ss1 is off)
                Mode2()
            Else if (ss0 is on and ss1 is on)
                Mode3()
```

End main function

```
Void Mode0 (void )
    Self design
Void Mode1 (void)
    Int randomLED, randomTimes;
    For (int sequence=1 ; sequence<9 ; sequence++)
            Use random function Create the step
```

```
 Get sequence
Use push button to response
If correct
                Success Score ++ and green light on
Else
                Fial Redlight on
Display final score and option to replay
End mode1


Void mode2(void)
Use push button indicate led
use potentiometer indicate blinks
create sequence
response
if correct
                success
else
                fail
end mode2

void mode3(void)

set blink rate
read potentiometer
use function to get delay period use LED0 to indicate
save the data globally
set sequence step period
Read potentiometer
Save step period globally
Play the game with new blink rate and step period
End
```

Laboratory Worksheet #05
Timer Overflow Interrupts Exercise

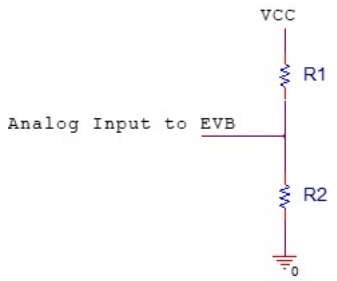

Figure 1: Voltage Divider

The above resistor network has two resistors, R1 and R2 (neither of these are potentiometers), connected in series between 5V and ground (0V). Recall, the total resistance for a series network is $R_{Tot} = R1 + R2$. The voltage across resistors in series is found by using voltage divider concepts. For the circuit shown, the voltage across each resistor is:

$$V_{R1} = \frac{R1}{R1 + R2}(Vcc) \qquad V_{R2} = \frac{R2}{R1 + R2}(Vcc)$$

In general, for N resistors in series, the voltage across the ith resistor is given by

$$V_{Ri} = \frac{Ri}{\sum_{n=1}^{N} R_n}(Vcc)$$

When answering the following questions, the current flowing through the two resistors is 2.5mA and the voltage input to the EVB is 1.25V when VCC = 5V.

Exercise 1:

1)	What is the total resistance of the two resistors?	2000ohm
2)	What is the voltage across R1?	3.75V
3)	What is the voltage across R2?	1.25V
4)	What is the resistance value of R1?	1500ohm
5)	What is the resistance value of R2?	500ohm

6) Using the closest resistor components you can find, build the small circuit and measure the voltage at the EVB input using the voltmeter. How close is it to the spec? (Put your resistors back when finished.)

it was 1.14 when tested. 0.16V smaller than expected.

Exercise 2: Software Initialization and A/D conversion

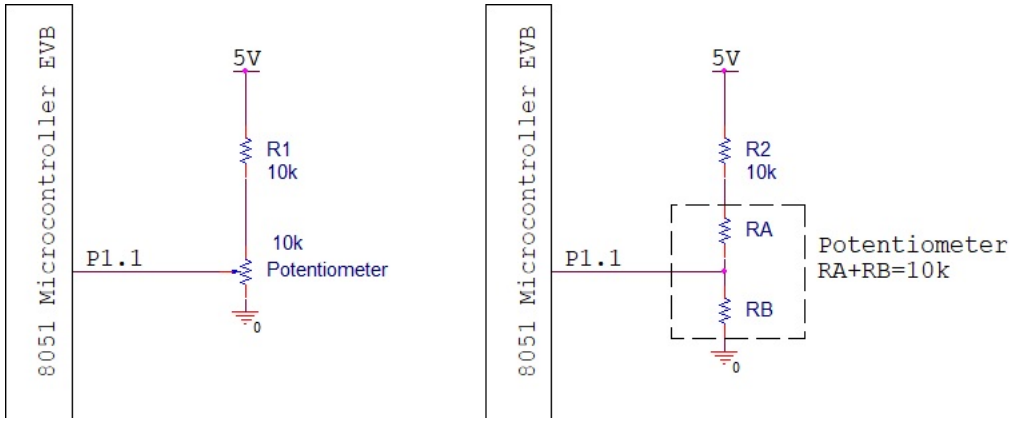

Figure 2: Potentiometer Circuit

In the above circuit, a resistor in series with a potentiometer is shown on the left. The equivalent circuit is shown on the right, with the two variable resistors inside the potentiometer. When determining voltage input to P1.1, you should treat the potentiometer as two resistors in series, making the effective circuit three resistors in series. The voltage at P1.1 is equal to the voltage across the 'bottom' resistor of the potentiometer, RB. The general voltage divider expression from Exercise 1 applies to the circuit. Answer the following questions based on the schematic shown above.

1) Write a function emphvoid Port_Init(void) to configure P1.1 as an analog input. Your function should leave the other bits (0, 2-7) unchanged.

```
P1MDIN &= ~0x02;          //Set P1.1 as an analog input
P1MDOUT & = ~0x02;        // Set P1.1 as a input port bit
P1 |= 0x02;               // Set P1.1 to a high impedance state
```

2) Write a function emphvoid ADC_Init(void) to set VREF to use the internal voltage reference of 2.4 V, set the ADC gain to 1, and enable ADC1.

```
REF0CN |= 0x03;           // Configure ADC1 to use VREF
REF0CN &= ~0x08;          //
ADC1CF |= 0x01;           // Set a gain of 1
ADC1CN = 0x80;            // Enable ADC1
```

3) Write a function *void ADC_Test(void)* that performs an ADC (analog to digital conversion) on P1.1 and stores the result in variable *P1_1_Result*.

```
AMX1SL = 1;                    // Set the Port pin number
ADC1CN &= ~0x20;               // Clear the conversion complete flag
ADC1CN |= 0x10;                // Start and A/D conversion
while(!(ADC1CN & 0x20));       // Wait for the conversion to be complete
P1_1_Result = ADC1;            // Assign the A/D conversion result
```

4) When the potentiometer is adjust such that RA = 6k and RB = 4k, what is the voltage at P1.1? (Apply voltage divider concepts.)

1V

5) Considering parts 3 and 4, what value (unsigned char) will be stored in the variable P1_1_Result?

106 or 0x6A;

6) If the voltage at P1.1 is now changed to 3.53 V, what value will be stored in P1_1_Result?

255 or 0xFF;

When complete, include Worksheet 6 with your Laboratory 2 Pre-lab submission.