EVB Pin		Port Bit	Bit Addresses & Labels
1	2	1.	
		2.	
3	4	3.	
		4.	
5	6	5.	
		6.	
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. 0XA0	SS
		30.	
31	32	31. 0XB6	LED0
		32. 0XB7	BUZZER
33	34	33. 0XB4	B1LED1
		34.	
35	36	35.	
		36. 0XB3	B1LED0
37	38	37. 0XB1	PB2
		38. 0XB0	PB1
39	40	39.	
		40.	

41 ⟷ 60

Software Initializations

A) Port I/0

```
P3MDOUT |= 0xD8;
P3MDOUT &= 0xFC;
P3 |= ~0xFC;

P2MDOUT &= 0xFE;
P2 |= ~0xFE;
```

B) Timers

C) Interrupts

D) A/D

E) PCA

F) XBAR

G) I2C

Yilu Zhou

compiler directives
```
#include <C8051_SDCC.h>
#include <stdio.h>
```
function prototypes
```
void Port_Init (void);
void Set_Outputs (void);
int Slide_Switch_Status (void);
int Pushbutton1_Status (void);
int Pushbutton2_Status (void);
void Status_Report (void);
```
declare global variables
```
sbit LED0, BILED0, BILED1, BUZZER, SS, PB1, PB2.
```
main function
declare local variables (NONE)
initialization functions.
```
Sys_Init ();
putchar (' ');
Port_Init ();
```
Begin infinite loop
execute Set_Outputs (void) function to read sbit inputs and set sbit outputs.
End infinite loop
End main function.

Functions
```
void Port_Init (void) Initialize the input/output device.
int Slide_Switch_Status (void) return 1 if Slide switch is ON.
int Pushbutton1_Status (void) return 1 if button 1 is pushed, 0 if not.
int Pushbutton2_Status (void) return 1 if button2 is pushed, 0 if not.
void Status_Report (void)
    Report the status (ON/OFF) of the inputs SS, PB1, PB2
void Set_outputs
    Report the status of input using Status_Report.
    If Slide switch is OFF, turn off all outputs.
    If Slide switch is ON, turn on LED
    If Both button is pushed, turn on Buzzer.
    If only PB2 is pushed, turn the BILED to green.
    If only PB1 is pushed, turn the BILED to red.
```

Compiler directives
#include <C8051_CPU.h>
#include <stdio.h>
function prototypes
void Port_Init(void);
void Set_Outputs(void);
int Slide_Switch_Status(void);
int Pushbutton1_Status(void);
int Pushbutton2_Status(void);
Void Status_Report(void);
declare global variables
bit LED0, BLLED0, BZED1, BUZZER, SS,
PB1, PB2.
main function
declare local variables (NONE)
initialization functions
Sys_Init();
putchar(' ');
Port_Init();
Begin infinite loop
execute Set_Outputs() function to read shift
input and set shift outputs.
End infinite loop
End main function
Functions

port_Init - Initializes Ports 2 and 3 in the desired modes for input and output

Void Port_Init (void)
{ Set Port 3 output pins to push-pull mode.
Set Port 3 input pins to open drain mode.
Set Port 3 input pin to high impedance state.
Set Port 2 input pins to open drain mode
Set Port 2 input pins to high impedence

P2MOUT &= 0xFE; // Set port 2 input pins to open drain mode.
P2 |= ~0xFE; // Set Port 2 input pins to high impedance state.

Void Set_outputs (void)
1. Slide switch "off" (input Hi) BiLED is green
all other input "off"
2. Slide switch is "on" and no Push buttons are pushed
BiLED is off
3. when slide switch is "on" and only Push button 1 is pushed. the Buzzer is turned on.
4. When the Slide switch is "on" and only Push button 2 is pushed the LED is on
5. When the slide switch is "on" and both Push buttons are pushed. the BiLED is red.
Return 1 if slide switch is ON
Return 0 if slide switch is OFF
Return 0 if push button is pushed
Return 1 if push button is not pushed

Report the status of input

Compiler directives;

#include <C8051_SDCC.h>

#include <stdio.h>

function protytpes

void Port-Init (void);

void Set-Outputs(void);

declare global variables

sbit SS, LED10, BL0, BL1, BUZZER, PB1, PB2

main function

declare local variables (NONE)

initialization functions.

Sys_Init();

putchar(' ');

port_Init ();

Infinite loop

execute Set_outputs (void) function to read inputs and set sbit outputs.

End infinite loop

End Main function.

Functions

void Port-Init (void) Initialize the input / output device

void Set_outputs(void)

If slide switch is on

turn on the LED 0

If the push button 1 is pushed

BL is not red

BL is green

If the push button 2 is pushed

BL is red

BL is not green

If both button is pushed

Buzzer is on

If Slide Switch is off

Turn off LED 0

Laboratory Worksheet #01
Number Systems Exercise

Convert the following decimal numbers to binary:

1) 14 — 1110

2) 189 — 1011 1101

3) 257 — 0001 0000 0001

4) 472 — 0001 1101 1000

Convert the following decimal numbers to hex:

5) 14 — 0xE

6) 189 — 0xBD

7) 257 — 0x101

8) 472 — 0x1D8

Convert the following hex numbers to decimal:

9) 0x37 — 55

10) 0xAB — 171

11) 0x0147 — 327

12) 0x2AE1 — 10977

Convert the following hex numbers to binary:

13) 0x37 — 0011 0111

14) 0xAB — 1010 1011

15) 0x0147 — 0000 0001 0100 0111

16) 0x2AE1 — 0010 1010 1110 0001

Convert the following binary numbers to hex:

17) 0010 1101 — 0x2D

18) 1010 1010 — 0xAA

19) 1110 0011 — 0xE3

20) 0010 1001 1011 0101 — 0x29B5

Convert the following binary numbers to decimal:

21) 0010 1101 — 45

22) 1010 1010 — 170

23) 1110 0011 — 227

24) 0010 1001 1011 0101 — 10677

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

Laboratory Worksheet #02
Logic Exercise

Answer the following questions given:

char a,b,c,d;

a = 0x00;

b = 0x0F;

c = 0x09;

0000 0000

0000 1111

0000 1001

0000 1111

0000 0100

0100

0x04

0011 0011

0000 1111

What is the value of **d** after execution of the following lines?

1) **d = a & c;** 0x00

2) **d = b & c;** 0x09

3) **d = b & 0x04;** 0x04

4) **d = b & 0x33;** 0x00

5) **d = b & !c;** 0x00

Are the following TRUE or FALSE?

6) **(a && c)** FALSE

7) **(b && c)** TRUE

8) **(b && 0x04)** TRUE

What is the value of **d** after execution of the following lines?

9) **d = a | c;** 0x09

10) **d = b | 0x10;** 0x1F

11) **d = b | a;** 0x FF

12) **d = b | !a;** 0x0F

Convert the following hex numbers to binary:

13) **(a || c)** TRUE

14) **(a || (c & 0x10))** FALSE

15) **(a || !b)** FALSE

0001 0000

0000 1001

0000 0000

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

Laboratory Worksheet #03
Hardware: Digital Input and Output Exercise

When developing hardware circuits, it is recommended to build and test small circuits that will later be expanded upon. This first project involves the use of a couple key components (74365 chips, LEDs, BILEDs, Buzzers, Resistors) that play an important role in both digital input and output.

Construct the circuit shown below. Debugging the hardware circuit is performed using the Logic Probe available in the Toolbox. Directions on using the logic probe can be found not only in the LITEC Multimedia Tutorials, but also in Chapter 2 of your lab manual. Also, please refer to Appendix B, Figure B.1 in the lab manual concerning the connections of **+5V** and **Ground** on the Smart Car connection board.

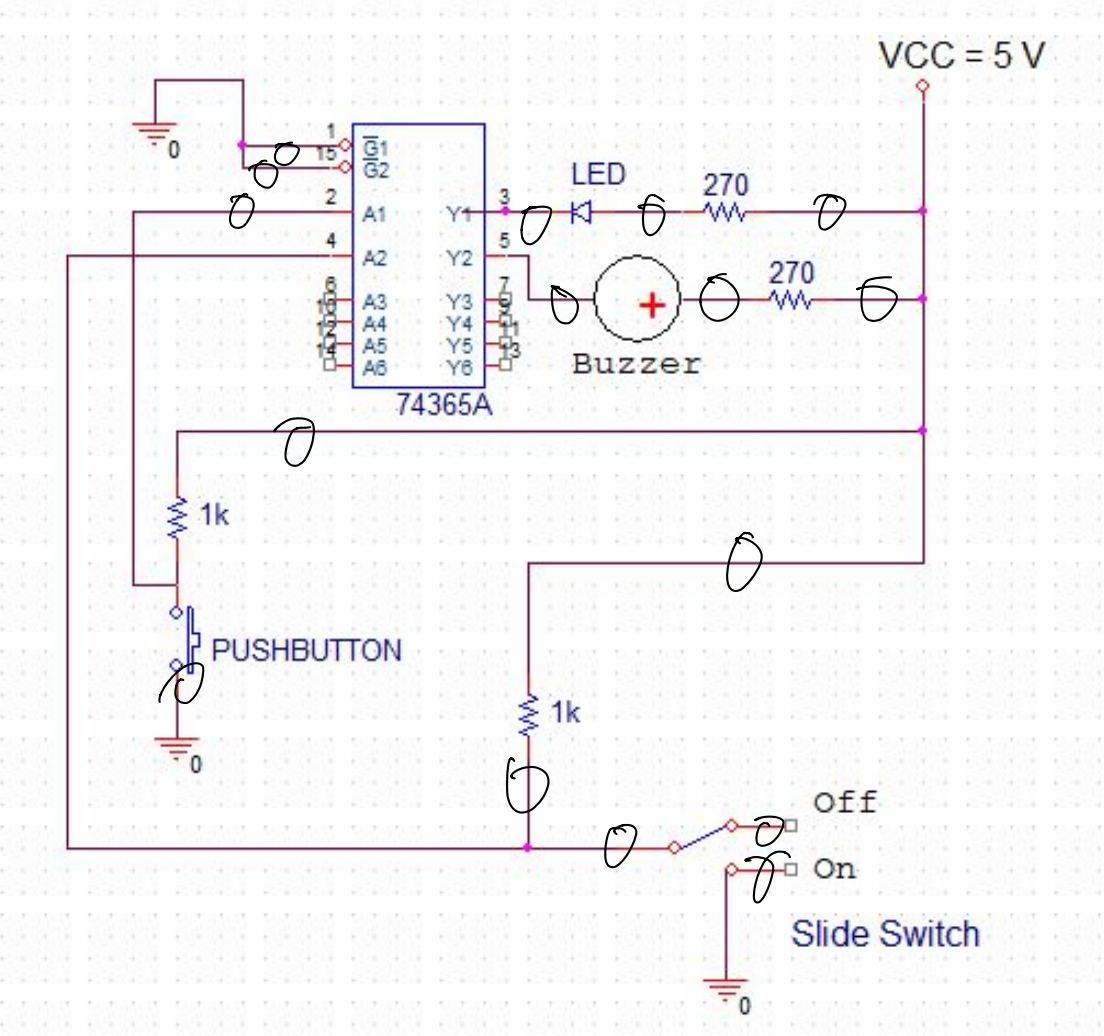

Figure 1: Worksheet 3 Schematic

Note: In the above circuit schematic, the power connections for the 74365 chip are not shown. This format is common to circuit schematics, where the implementer is expected to know the connections for power and ground.

1) Using the on-line data sheets, determine all pins of the 74F365 that need to be connected toa high voltage (VCCor +5V) and those that need to be connected to a low voltage (GND or 0V).

16 high voltage

1, 15, 8 low voltage

2) Using the Logic Probe, what value (high/low) do you get when you test pin #16 on the 74F365 buffer? What color is Logic Probe LED?

HI (Red)

3) What about when you test pin#1 on the same chip?

LO (Green)

4) What voltage values are the buffer gate outputs connected to the LED and Buzzer (pin 3 and pin 5 of the Hex buffer, respectively) when the slide switch is ON and the button is pushed? Are the LED and Buzzer on (lit/noisy) or off (unlit/no sound)?

5.5V for pin 3

0.03V for pin 5

5) Connect pins 1 and 15 to power (5V) instead of ground. What happens when you push the button or move the slide switch?

Nothing happened

6) Disconnect pins 1 and 15 completely (so they are not connected to anything). The pins are now considered 'floating', which means their voltage level is uncertain. What happens when you push the button or move the slide switch?

Nothing happened

When complete, include Worksheet 3 with your Laboratory 1-1 Pre-lab submission.

Laboratory Worksheet #04
Hardware: Digital Input and Output Exercise

When complete, insert Worksheet 04 in your laboratory notebook. Worksheets are required when the notebooks are graded. Perform any necessary calculations on the left page of the notebook where the worksheet is placed. Keep individual copies of the worksheet for your own records. This worksheet is a pre-lab exercise to be done before starting Lab 1-1 and should not be confused with the lab.

One of the important aspects of the software is initializing Special Function Registers (SFRs). In Laboratory 1, you will create initialization functions for Port I/O, which involves setting the correct SFR bits to 0 or 1, as needed. The logic assignment operations developed in Worksheet 2 are used to set the appropriate bits without changing the other bits.

Additionally, using the sbit command to assign a variable name to a single bit in the SFR can make programming and code execution much simpler. You will read from or write to individual bits when performing Input/Output operations on the Port pins. These read and write operations will be performed using the sbit labels assigned to the specific Port pins.

As an example problem, Port 2 will be configured for both input and output. Note, this is an example problem and is not to confused with Laboratory 1. The followed Port bits will be assigned as inputs or outputs:

P*n.m* (Port *n*, bit *m*)	Description	Bit Label
P2.1	Input bit for doorbell	**DB**
P2.3	Input bit for an alarm clock	**AC**
P2.5	Output bit for a porch light	**Porch**
P2.6	Input bit for a garage door	**GD**

All other bits are considered previously assigned and should not be changed.

Refer to the manual section *Input/Output Ports on the C8051* or the course slides to determine the syntax. The memory locations for Port 2 bit 0 is 0xA0, Port 2 bit 1 is 0xA1, through Port 2 bit 7 at 0xA7. Complete the following four lines of code to assign the labels to the appropriate bit using the sbit command.

__sbit __at 0x A1 DB ; // remember to include the double "__"

__sbit __at 0xA3 AC;

__sbit __at 0xA5 Porch;

__sbit __at 0xA6 GD;

In the following SFR data tables, indicate whether the bit should be set high (1), low (0), or undetermined/unchanged (X).

P2MDOUT (input bits are set to 0, output bits are set to 1, unchanged bits are indicated with an X)

bit	7	6	5	4	3	2	1	0
	X	0	1	X	0	X	0	X

Determine the bit mask for setting the appropriate bits high (logic 1) P2MDOUT | = 0x20;

Determine the bit mask for setting the appropriate bits low (logic 0) P2MDOUT & = 0xB5;

P2 (input bits are set to 1 which is a high impedance state, all other bits are unchanged X)

bit	7	6	5	4	3	2	1	0
	X	1	X	X	1	X	1	X

Determine the bit mask for setting the appropriate bits high (logic 1) P2 | = 0x4A;

In summary, using the above definitions, complete the Port_Init() function for this example.

```
Port_Init()
{
    P2MDOUT &= 0xB5;    //configure Port 2 bits as inputs
    P2MDOUT |= 0x20;    //configure Port 2 bits as outputs
    P2 |= 0x4A;         //set Port 2 input bits to a high impedance state
}
```

When complete, include Worksheet 4 with your Laboratory 1-1 Pre-lab submission.