***//Expt.5: Interfacing LEDs, Switches, Buzzer and Relay***

***//Includes***

#include <p18f4550.h> //Include Controller specific .h

#include "vector\_relocate.h" //Vector Remapping for USB

//HID Bootloader

***//Declarations***

#define lrbit PORTBbits.RB1 //SW0 interfaced to RB1

#define rlbit PORTBbits.RB0 //SW1 interfaced to RB0

#define buzzer PORTCbits.RC2 //Buzzer interfaced to RC2

#define relay PORTCbits.RC1 //Relay interfaced to RC1

***//Function Prototypes***

void msdelay (unsigned int time); //Function for delay

//Start of Program Code

void main() //Main Program

{

unsigned char val=0; //Variable to latch the switch condition INTCON2bits.RBPU=0; //To Activate the internal pull on PORTB

ADCON1 = 0x0F; //To disable the all analog inputs

TRISBbits.RB0=1; //To configure RB0 as input for sensing SW1

TRISBbits.RB1=1; //To configure RB1 as input for sensing SW0

TRISCbits.TRISC1=0; //To configure RC1 (relay) as output

TRISCbits.TRISC2=0; //To configure RC2 (buzzer) as output

TRISD = 0x00; //To configure PORTD (LED) as output

PORTD = 0x00; //Initial Value for LED

buzzer = 0; //Initial Value for Buzzer

relay = 0; //Initial Value for Relay

while (1) //While loop for repeated operation

{

if (!(lrbit)) //To check whether SW0 is pressed

val = 1; // Latch the status of switch SW0

if (!(rlbit)) //To check whether SW1 is pressed

val = 2; // Latch the status of switch SW1

if (val == 1)

{

buzzer = 1;

relay = 1;

PORTD = 0x80; // Make the MSB bit equal to 1

msdelay(250);

PORTD = PORTD >>1; //Shift left by 1 bit

}

if (val == 2)

{

buzzer = 0;

relay = 0;

PORTD = 0x01; // Make the LSB bit eqaul to 1

msdelay(250);

PORTD = PORTD<<1; //Shift right by 1 bit

}

} //End of the Program

***//Function Definitions***

void msdelay (unsigned int time) //Function for delay

{

unsigned int i, j;

for (i = 0; i < time; i++)

for (j = 0; j < 710; j++);//Calibrated for a 1 ms delay in MPLAB

}

***// EXP:6 Generate 10Hz frequency by interrrupt method and Timer 0***

#include <p18f4550.h>

void timer\_isr(void);

void delay\_ms(unsigned int);

extern void \_startup (void);

#pragma code RESET\_INTERRUPT\_VECTOR = 0x1000

void reset (void)

{

\_asm

goto \_startup

\_endasm

}

#pragma code

#pragma code HIGH\_INTERRUPT\_VECTOR = 0x1008

void high\_ISR (void)

{

\_asm

goto timer\_isr

\_endasm //The program is relocated to execute the interrupt routine timer\_iser

}

#pragma code

// This function is executed as soon as the timer interrupt is generated due to timer overflow

#pragma interrupt timer\_isr

void timer\_isr(void)

{

TMR0H = 0X6D; // Reloading the timer values after overflow

TMR0L = 0X82;

PORTDbits.RD0 = ~PORTDbits.RD0; //Toggle the PORTB led outputs RB0 - RB3

INTCONbits.TMR0IF = 0; //Resetting the timer overflow interrupt flag

}

void main()

{

TRISD = 0;

PORTD=0;

T0CON = 0x03; //Set the timer to 16-bit mode, internal

//instruction cycle clock,1:256 prescaler

TMR0H = 0x6D; // Reset Timer0 to 0x48E5 TO MAKE DELAY OF 1 SECOND

TMR0L = 0x82;

INTCONbits.GIE = 1; // Global interrupt enabled

INTCONbits.TMR0IE = 1; // TMR0 interrupt enabled

T0CONbits.TMR0ON = 1; // Start timer0

while(1);

}

void delay\_ms(unsigned int time)

{

unsigned int i,j;

for (i=0;i<time;i++)

for (j=0; j<710;j++);

}

***//Expt.7: LCD Interfacing***

***//Includes***

#include <p18f4550.h>

#include "vector\_relocate.h"

***//Declarations***

#define LCD\_DATA PORTD //LCD data port to PORTD

#define ctrl PORTE //LCD control port to PORTE

#define rs PORTEbits.RE0 //register select signal to RE0

#define rw PORTEbits.RE1 //read/write signal to RE1

#define en PORTEbits.RE2 //enable signal to RE2

***//Function Prototypes***

void init\_LCD(void); //Function to initialise the LCD

void LCD\_command(unsigned char cmd); //Function to

//pass command to the LCD

void LCD\_data(unsigned char data); //Function to

//write character to the LCD

void LCD\_write\_string(static char \*str); //Function to write

//string to the LCD

void msdelay (unsigned int time); //Function to generate delay

***//Start of Main Program***

void main(void)

{

char var1[] = " E & TC DEPT."; //Declare message to be displayed

char var2[] = "PICT";

ADCON1 = 0x0F; //Configuring the PORTE pins as digital I/O

TRISD = 0x00; //Configuring PORTD as output

TRISE = 0x00; //Configuring PORTE as output

init\_LCD(); // call function to initialise of LCD

msdelay(50); // delay of 50 mili seconds

LCD\_write\_string(var1); //Display message on first line

msdelay(15);

LCD\_command(0xC0); // initiate cursor to second line

LCD\_write\_string(var2); //Display message on second line

while (1); //Loop here

} //End of Main

***//Function Definitions***

void msdelay (unsigned int time) //Function to generate delay

{

unsigned int i, j;

for (i = 0; i < time; i++)

for (j = 0; j < 710; j++); //Calibrated for a 1 ms delay in MPLAB

}

***// Function to initialise the LCD***

void init\_LCD(void)

{

LCD\_command(0x38); // initialization of 16X2 LCD in 8bit mode

msdelay(15);

LCD\_command(0x01); // clear LCD

msdelay(15);

LCD\_command(0x0C); // cursor off

msdelay(15);

LCD\_command(0x80); // go to first line and 0th position

msdelay(15);

}

***//Function to pass command to the LCD***

void LCD\_command(unsigned char cmd)

{

LCD\_DATA = cmd; //Send data on LCD data bus

rs = 0; //RS = 0 since command to LCD

rw = 0; //RW = 0 since writing to LCD

en = 1; //Generate High to low pulse on EN

msdelay(15);

en = 0;

}

***//Function to write data to the LCD***

void LCD\_data(unsigned char data)

{

LCD\_DATA = data; //Send data on LCD data bus

rs = 1; //RS = 1 since data to LCD

rw = 0; //RW = 0 since writing to LCD

en = 1; //Generate High to low pulse on EN

msdelay(15);

en = 0;

}

***//Function to write string to LCD***

void LCD\_write\_string(static char \*str)

{

int i = 0;

while (str[i] != ‘\0’)

{

LCD\_data(str[i]); // sending data on LCD byte by byte

msdelay(15);

i++;

}}