UNIT – V: Real Word Interfacing With 18FXXXX

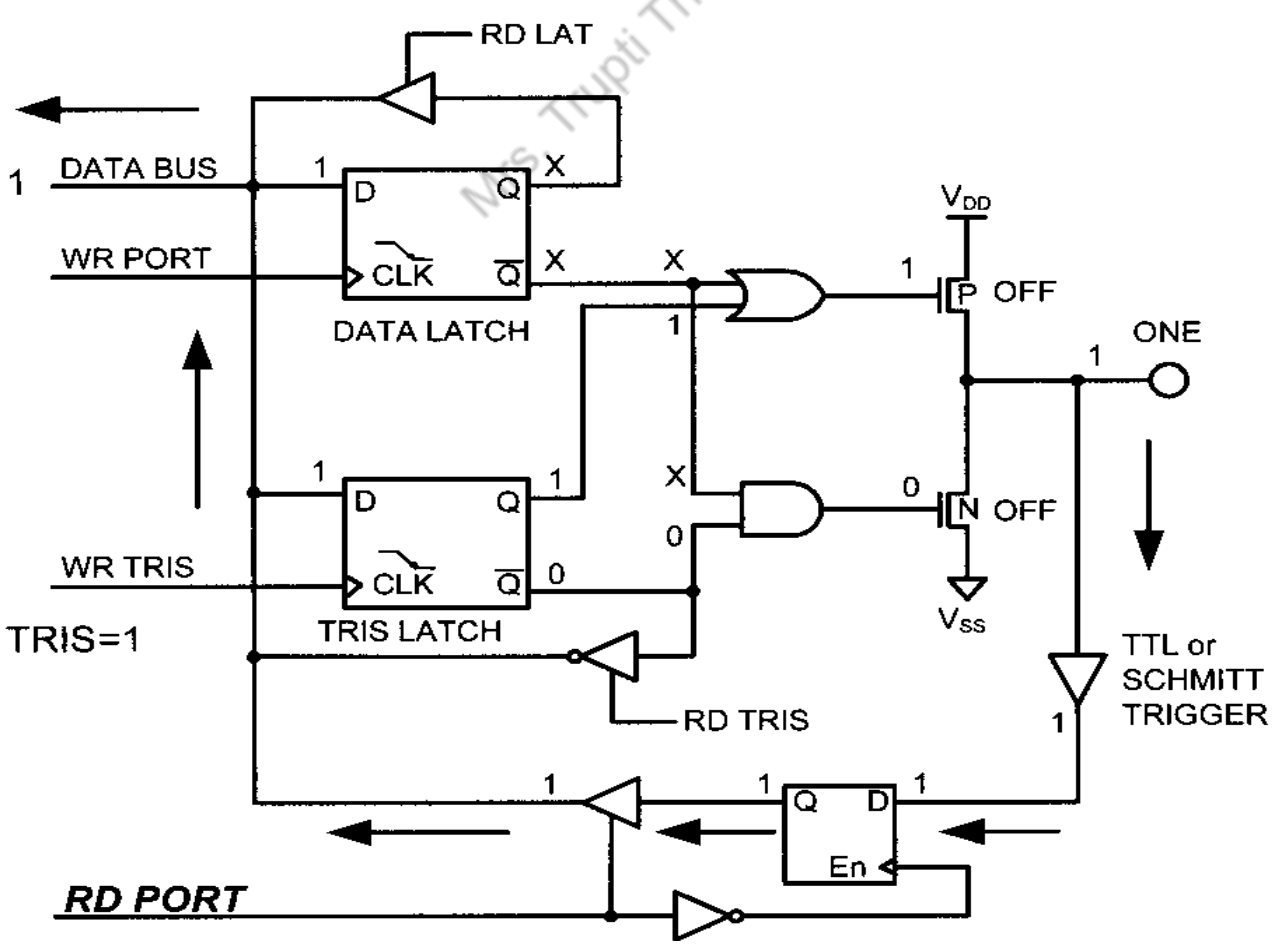
Q. 1 Explain PIC 18F4XX Port Structure

- PIC18FXX microcontroller has a number of input/output pins which are used for connection with external devices. Called as ports.
- PORTS denoted by A, B, C, D and E.
    - Port A consists of seven pins named as RA0-RA6
    - Port B consists of eight pins named as RB0-RA7
    - Port C consists of eight pins named as RC0-RC7
    - Port D consists of eight pins named as RD0-RD7
    - Port E consists of three pins named as RE0-RE2

Each port has three registers for its operation. These registers are

- TRIS register (data direction register)
- PORT register (reads the levels on the pins of the device)
- LAT register (output latch)

Internal structure of Port is as shown below



Above port structure consist of following components,

1. Data latch
2. Output driver
3. Input buffer
4. TRIS latch

There are 4 cases of Operation,

Writing LOW on pin	Reading LOW on pin
Writing High on pin	Reading High on pin

Two cases are explained below

Case1 : Writing "LOW" on Pin Px.x	Case : Reading "High" on Pin Px.x
Internal CPU Bus(DATA Latch input)= 0 Output of DATA latch Q = 0 , Q = 1 Internal (TRIS Latch input)= 0 Output of TRIS latch Q = 0 , Q = 1 **OR-gate** A input=1 B input=0 Y output=1 P-type MOSFET=OFF **AND -gate** C input= 1 D input =1 Z output=1 N-type MOSFET=ON The pin is pulled down by the lower FET. Hence at Px.x = logic '0' on pin. Hence the output becomes zero.	Internal (TRIS Latch input)= 1 Output of TRIS latch Q = 1 , Q = 0 **OR-gate** A input=X (Don't care) B input=1 Y output=1 P-type MOSFET=OFF **AND -gate** C input= X (Don't care) D input =0 Z output=0 N-type MOSFET=OFF Both FET's remains OFF and Hence Logic 1 which at pin Px.x will be read **Read Latch** Input =1 Q=1 **Internal CPU Bus will receive = 1**

Q. 2 Explain PIC 18F4XX Interfacing of LED

Or

Write code to flash LED after 1 second

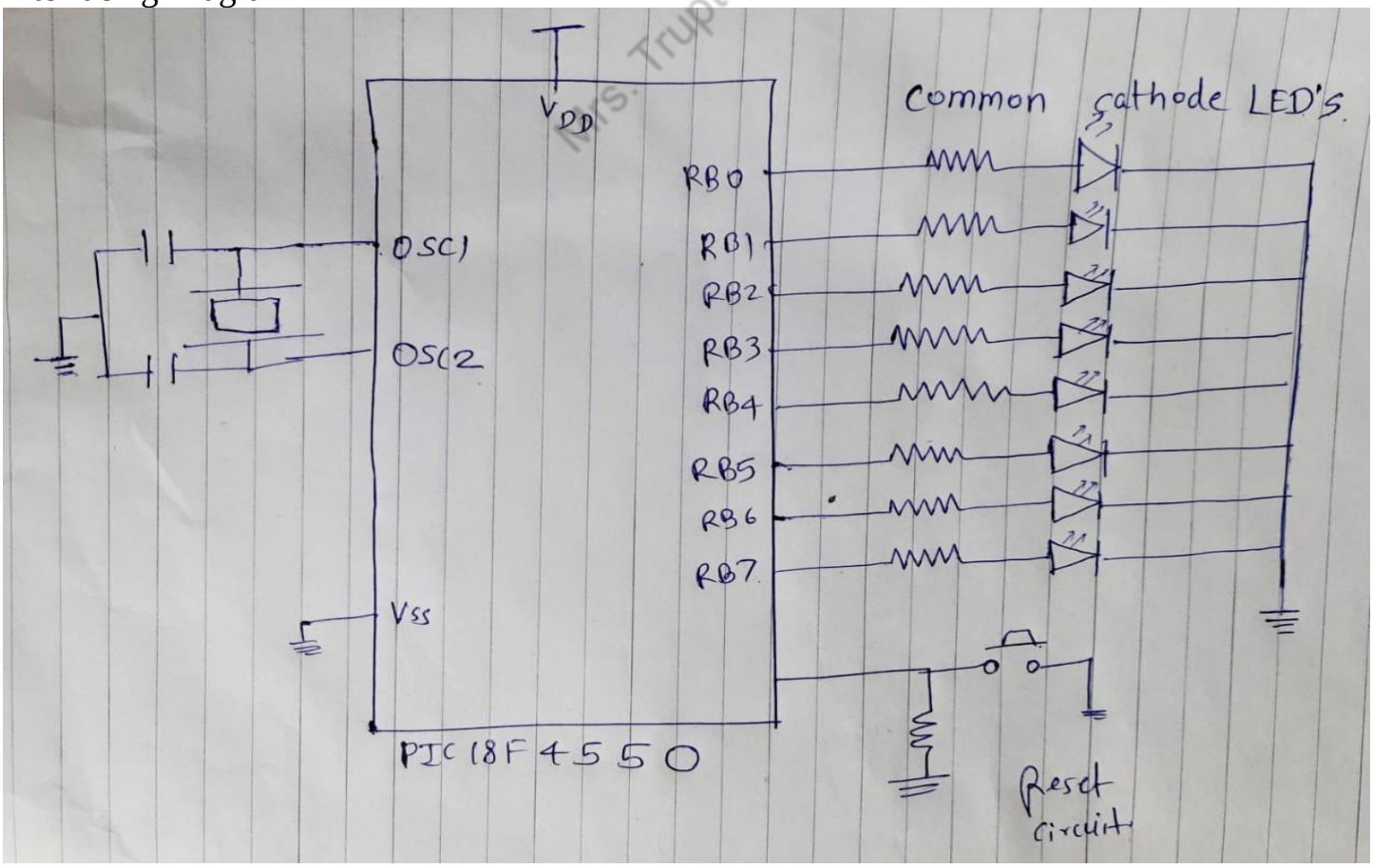
LED (Light emitting diode) is output device having two terminals anode and cathode. More than one LED can be interfaced with any microcontroller using either common cathode or common anode configuration

Forward voltage required by a diode to turn it on for red/orange/yellow LED = 1.8 – 2.2V, and for green/blue/white LED= 3.4V.)

Algorithm

1. Configure the TRISB register to make PortB as output port.
2. Set all the bits of PORTB register High (1) to glow all LEDs.
3. Provide some delay.
4. Set all the bits of PORTB register Low (0) to turn off the LEDs.
5. Provide some delay.
6. Repeat the process from step 2.

Interfacing Diagram



Program for 1 second delay

```
#include <PIC18F4550.h>
void Delay(void); //Delay Function

Void  main()
{
TRISB=0x00; //Port-B pin 4 as output
while(1)
 {
PORTB=0X00; //Toggle Output
Delay();
PORTB=0X00; //Toggle Output
Delay(); //Delay
   }
}
void Delay(void)
{ //10ms delay
T1CON=0x01; //Timer-1 16-bit mode Prescaler 1:4
TMR1H=0x30; //Count Hight Byte
TMR1L=0xD4; //Count Low Byte
//Runing  for loop for 100 times produces 1 second 10ms x 100 = 1 second
for(int i=1;i< =100;i++)
  {
T1CONbits.TMR1ON=1; //Run timer
while(INTCONbits.TMR0IF==0); //Wait for flag to over flow
T1CONbits.TMR1ON=0; //Switch off timer
INTCONbits.TMR0IF=0; //Clear Interrupt
   }
}
```


Q. 3 Explain PIC 18F Interfacing to LCD

Or

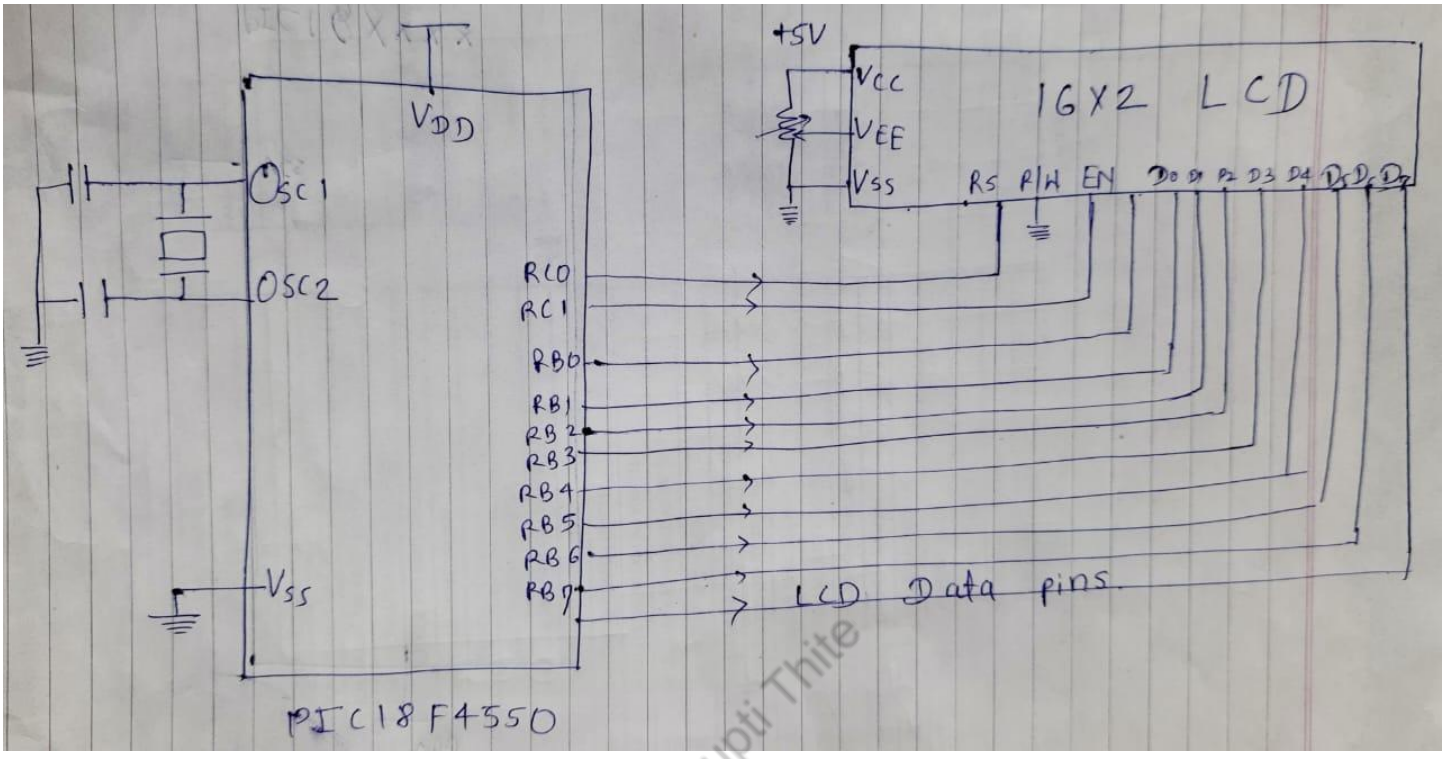
Write embedded C code to display “Welcome” on LCD

LCD stands for Liquid crystal display. 16×2 LCD is named so because; it has 16 Columns and 2 Rows. There are a lot of combinations available like 8×1, 8×2, 10×2, 16×1, etc. but the most used one is the 16×2 LCD. So, it will have 16×2 = 32 characters in total and each character will be made of 5×8 Pixel Dots.

It can be used in two modes. In 4 bit mode we send the data nibble by nibble.

Whereas in 8 bit mode we can send the 8-bit data directly in one stroke since we use all the 8 data lines.

Interfacing Diagram for 8 bit mode



Algorithm

1. Make respective port as output using TRIS register
2. Send First commands to LCD
3. Send characters / Message /data to display on LCD

Steps for Sending Command:

step1: Send the I/P command to LCD.
step2: Select the Control Register by making RS low.
step4: Send a High-to-Low pulse on Enable PIN with some delay

Steps for Sending Data:

step1: Send the character to LCD.
step2: Select the Data Register by making RS high.
step4: Send a High-to-Low pulse on Enable PIN with some delay

Program:

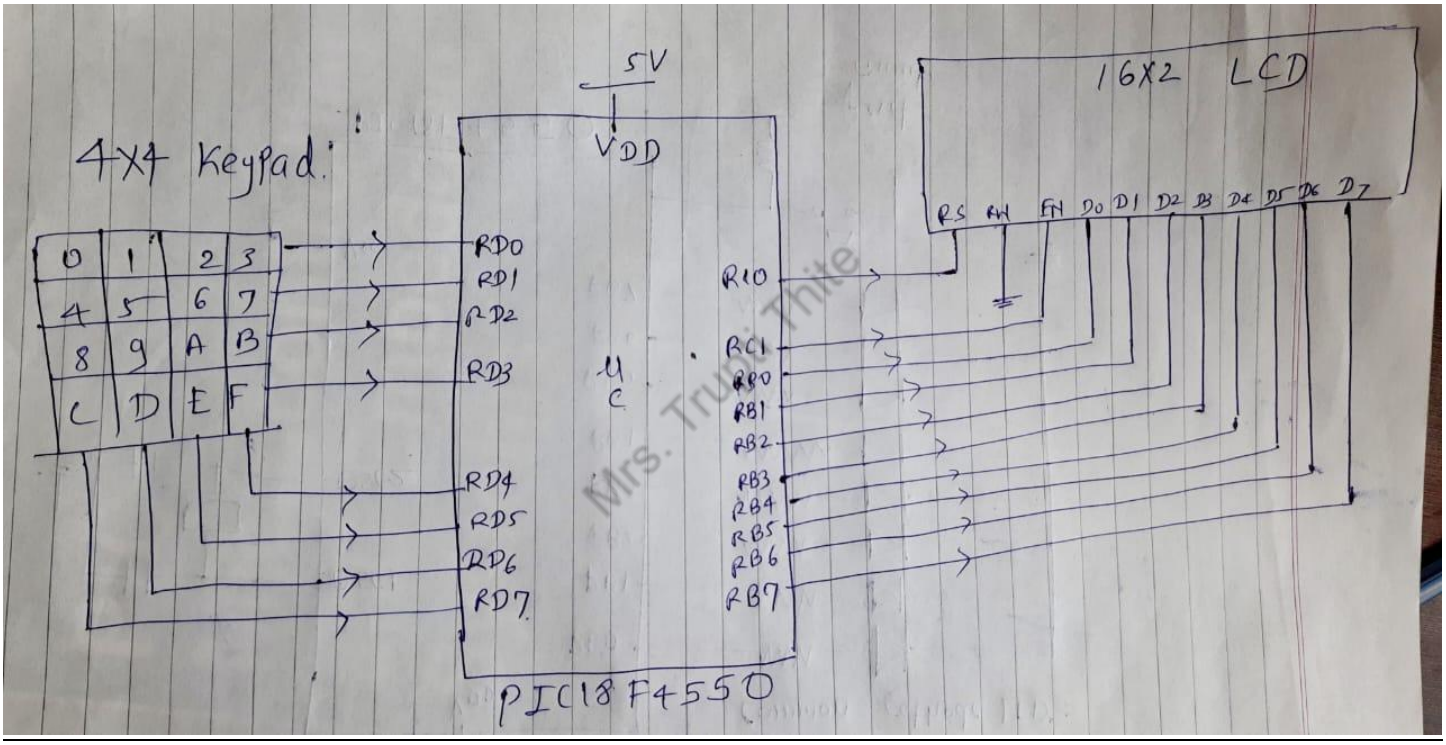
```
#include<P18f4550.h
#define ldata PORTB              //define name ldata to PORTB
#define rs PORTCbits.RC0         //define name rs(egister select) for RC0 pin
#define en PORTCbits.RC1         //define name en (enable) for RC1 pin

void delay(unsigned int);
void lcdcmd(unsigned char);
void lcdval(unsigned char);
void main()
{
unsigned char xc;
   TRISB=0;          //make PORTB as output
   TRISC=0;          //make PORTC as output

   lcdcmd(0x38);     //command to initialise LCD.
   lcdcmd(0x0E);     //command  to make Display on cursor blinking
   lcdcmd(0x01);     //command to clear display screen
   lcdcmd(0x06);     //command to increment cursor
   lcdcmd(0x86);     //command to set cursor at 1st line 6th position

   lcdval('W');
   lcdval('e');
   lcdval('l');
   lcdval('c);
   lcdval('o');
   lcdval('m);
   lcdval('e');

  while(1);
}

// lcdcmd fuction used to select COMMAND REGISTER of LCD by making,
// rs=0 and en=1 to 0 edge AND sends 8 bit command to PORTB
```

```
void lcdcmd(unsigned char y)
{
   rs=0;
   ldata=y;
   en=1;
   delay(10);
   en=0;
   delay(10);
}

// lcdval fuction used to select DATA REGISTER of LCD by making,
// rs=1 and en=1 to 0 edge AND sends 8 bit command to PORTB

void lcdval(unsigned char y)
{
   rs=1;
   ldata=y;
   en=1;
   delay(10);
   en=0;
   delay(10);
}


 void delay(unsigned int X)
   {
     unsigned int i,j;
     for(i=0;i<=X;i++)
        {  for(j=0;j<20;j++)
             {}
        }
    }
```

Q. 4 Explain PIC 18F Interfacing to Keypad

- The keypad is used as an input device to read the key pressed by the user and to process it.
- 4x4 keypad consists of 4 rows and 4 columns. Switches are placed between the rows and columns. A keypress establishes a connection between the corresponding row and column between which the switch is placed.
- In order to read the key press, we need to configure the rows as outputs and columns as inputs.
- Columns are read after applying signals to the rows in order to determine whether or not a key is pressed and if pressed, which key is pressed.

Interfacing Diagram



Algorithm:

1. Connect four column pins and four-row pins to the microcontroller port.
2. Define column pins as input and row pins as output

TRISD = 0XF0; /*TRISD register decides the direction of the PORT D pins.

1 for input and 0 for output */

3. Wait for a key press (initial scanning)
4. Make all the row pins high, then keep scanning the rows till a keypress get detected.

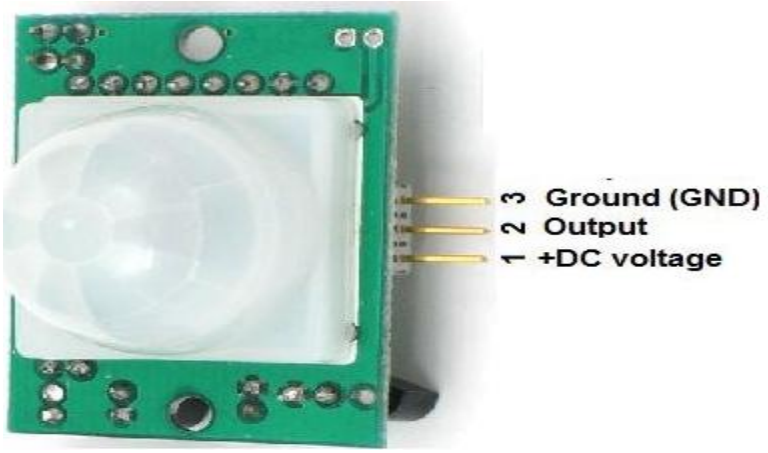
```
do {
   PORTD = 0X0F;          // Make Row pins high
   column = PORTD;        // Read column pin status: Read the port
   column &= 0xF0;        // Read column pin status: Extract column pin status from PortD
   }while(column==0x00);
```

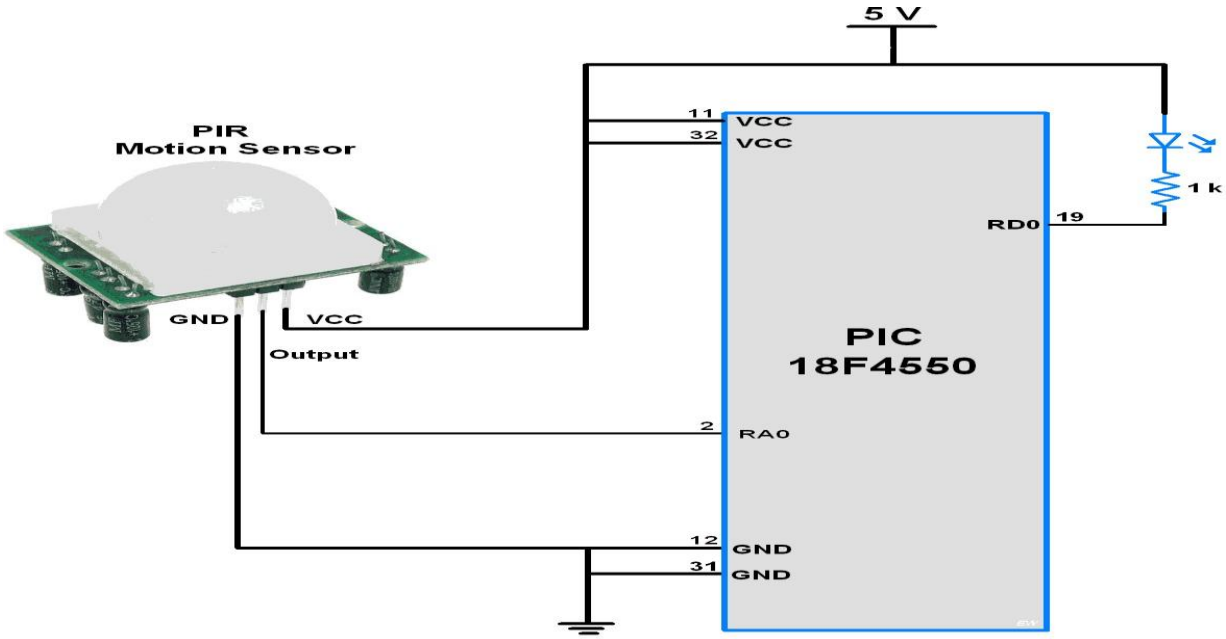
5. Repeat this procedure till a keypress event got detected
6. Detect and identify the pressed key and display on LCD

Q 5 Explain PIR Motion Sensor Interface with PIC18F4550

- PIR sensor detects infrared heat radiations.
- It can be used to detect the presence of living objects that emit infrared heat radiation.
- The PIR sensor is split into two slots.
- Whenever a stationary object is in front of the sensor, the two slots receive the same amount of radiation and the output is zero.
- Whenever a moving object is in front of the sensor, one of the slots receives more radiation than the other slot.
- This makes the output swing high or low.
- This change in output voltage is the result of the detection of motion.

PIR Sensor pin diagram






```
 * PIR Motion sensor interface with PIC18F4550
 * http://www.electronicwings.com
 */


#include <pic18f4550.h>
#include "Configuration_Header_File.h"

#define Motion_detection PORTAbits.RA0  /* Read PIR sensor's data on this pi
#define PORT_Dir TRISAbits.RA0          /* define for setting direction */
#define LED LATD0                       /* connect LED to the PORT pin */
#define LED_Dir TRISDbits.RD0           /* define for setting direction */

void MSdelay(unsigned int val);

void main(void)
{
    ADCON1=0x0F;        /* this makes all pins as a digital I/O pins */
    PORT_Dir = 1;       /* set as input port */
    LED_Dir = 0;        /* set as output port */
    LED = 0;            /* initially turned OFF LED */
    OSCCON = 0x72;
    while(1)
    {
        while(Motion_detection)
            LED = 1;    /* LED turn ON if any Human motion is detected */

            LED = 0;    /* LED turn OFF */
    }
}
void MSdelay(unsigned int val)
{
     unsigned int i,j;
        for(i=0;i<val;i++)
            for(j=0;j<165;j++);  /*This count Provide delay of 1 ms for 8MH
 }
```

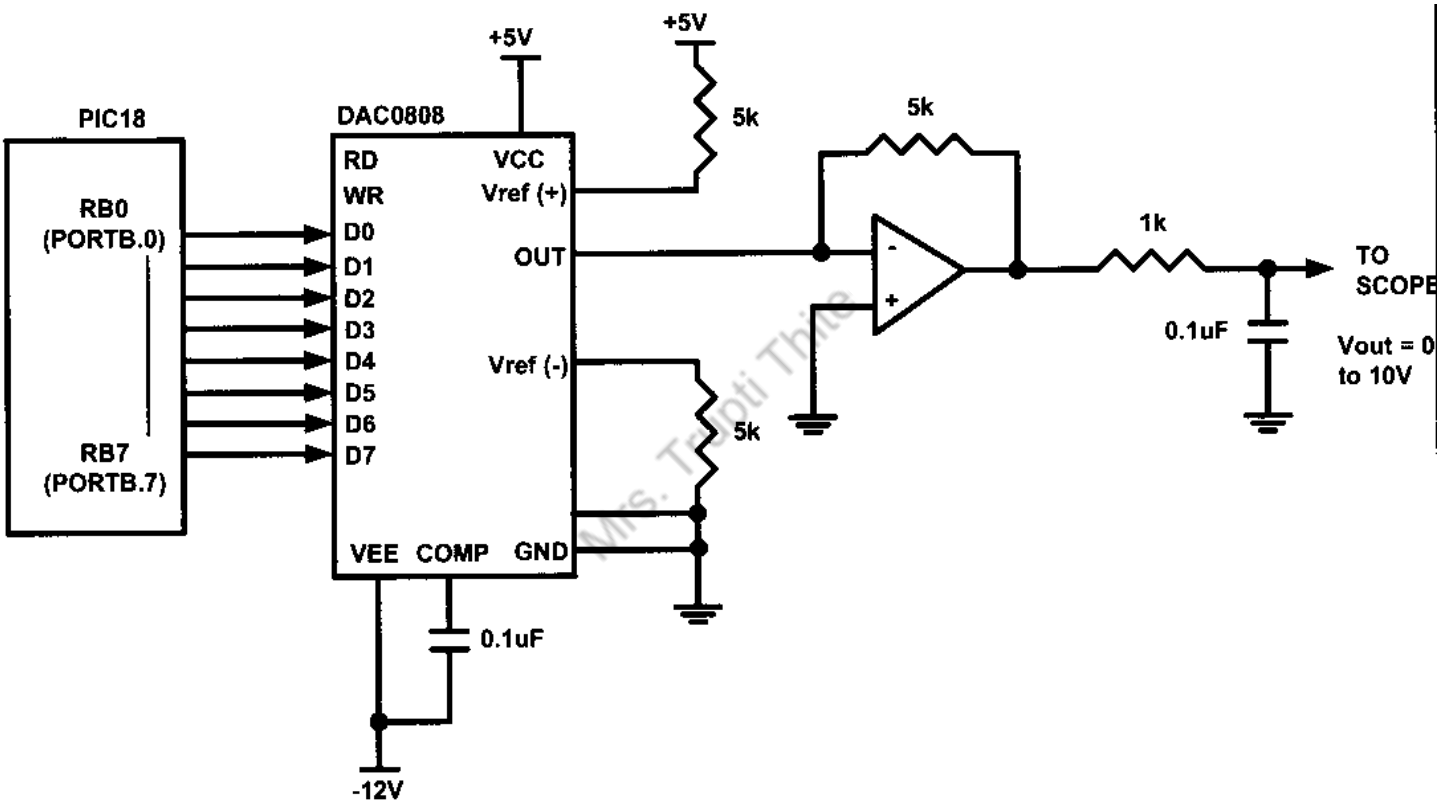
Q 6 Explain DAC with PIC18F4550

OR

Write C code to Generate sine waveform using DAC with PIC18F4550

Digital to analog Converter is widely used to convert digital pulses into analog signal. Interfacing diagram for DAC and PIC 18 for sine wave generation is as shown below.

DAC 0808 COVERTS 8 BIT Digital value into current output, This current output is converted into voltage by V-I Converter using Op-Amp.



Formula for generating sine wave using DAC is as follows,

$$V_{out} = 5\ V + (5 \times \sin\theta)$$

Using Above formula we can create table as follows for the values to send to DAC from PIC

Angle θ (degrees)	Sin θ	V_{out} (Voltage Magnitude) 5 V + (5 V × sin θ)	Values Sent to DAC (decimal) (Voltage Mag. × 25.6)
0	0	5	128
30	0.5	7.5	192
60	0.866	9.33	238
90	1.0	10	255
120	0.866	9.33	238
150	0.5	7.5	192
180	0	5	128
210	–0.5	2.5	64
240	–0.866	0.669	17
270	–1.0	0	0
300	–0.866	0.669	17
330	–0.5	2.5	64
360	0	5	128

Program:


```
#include<PIC18F4550.h>

            unsigned char WAVEVALUE[12] ={128,192,238,255,
                    238,192,128,64,
                    17,0,17,64};
void main()
  {
    unsigned char x;
    TRISB=0;
    while(1)
       {
       for(x=0;x<12;x++)
        PORTB = WAVEVALUE[x];
       }
  }
```

Design of Home Protection System

Problem Statement- Design a Home protection system to for indicating various parameters like temperature, door open /closed, internal apparatus on, which will give alert by indicator, display and sounding alarm if exceed the set point. Also make provision to store few current records in the serial memory for analysis.

Answer

Step 1-Design of Power Supply

The microcontroller selected is PIC 18F4550 which works on the frequency of oscillator ranging from 0 to 20 MHz and requires power supply of ± 5 or ±12 V. A simple circuit design for +5 V is shown in Fig.

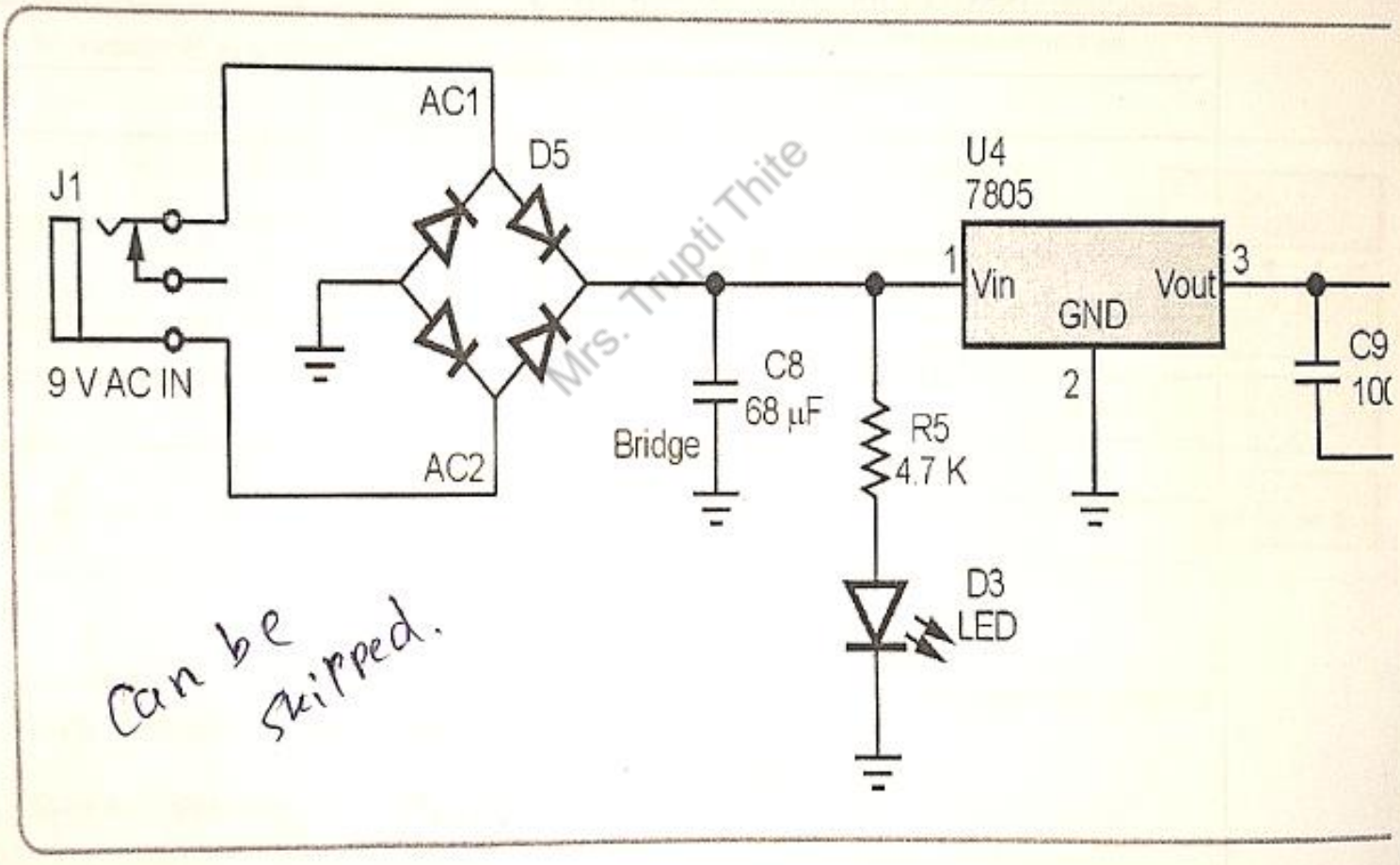


Fig. Design of +5V Power Supply

Step 2-Design of Clock Circuit

The Quartz crystal is connected to OSC1 and OSC2 pin in order to synchronize the operation of all components connected with internal and external means. The values for C1 and C2 are

selected according to the crystal frequency for stabilizing the oscillator pulses. In general with quartz crystal 22-33 µF is preferred.

Step 3-Design of Reset Circuit

The RC high pass filter with C=0.1µF along with 20 kΩ register is connected to MCLR pin. When high pulse appear on it, resets the contents of inter registers and SFRS to initial value.

PIC has 33 I/O lines which can be configured as input and out using TRISX register.

Step 4-General Block Diagram

The general block diagram of any security system without in build ADC is shown in Fig. Some of the modern processor like PIC has the in-build ADC and require only signal conditioning circuit. The sample signal conditioning circuit s shown in Fig.7.23 The signal conditioning circuit for any analog signal varies from signal to signal. For any low level signals an Instrumentation amplifier is best choice.

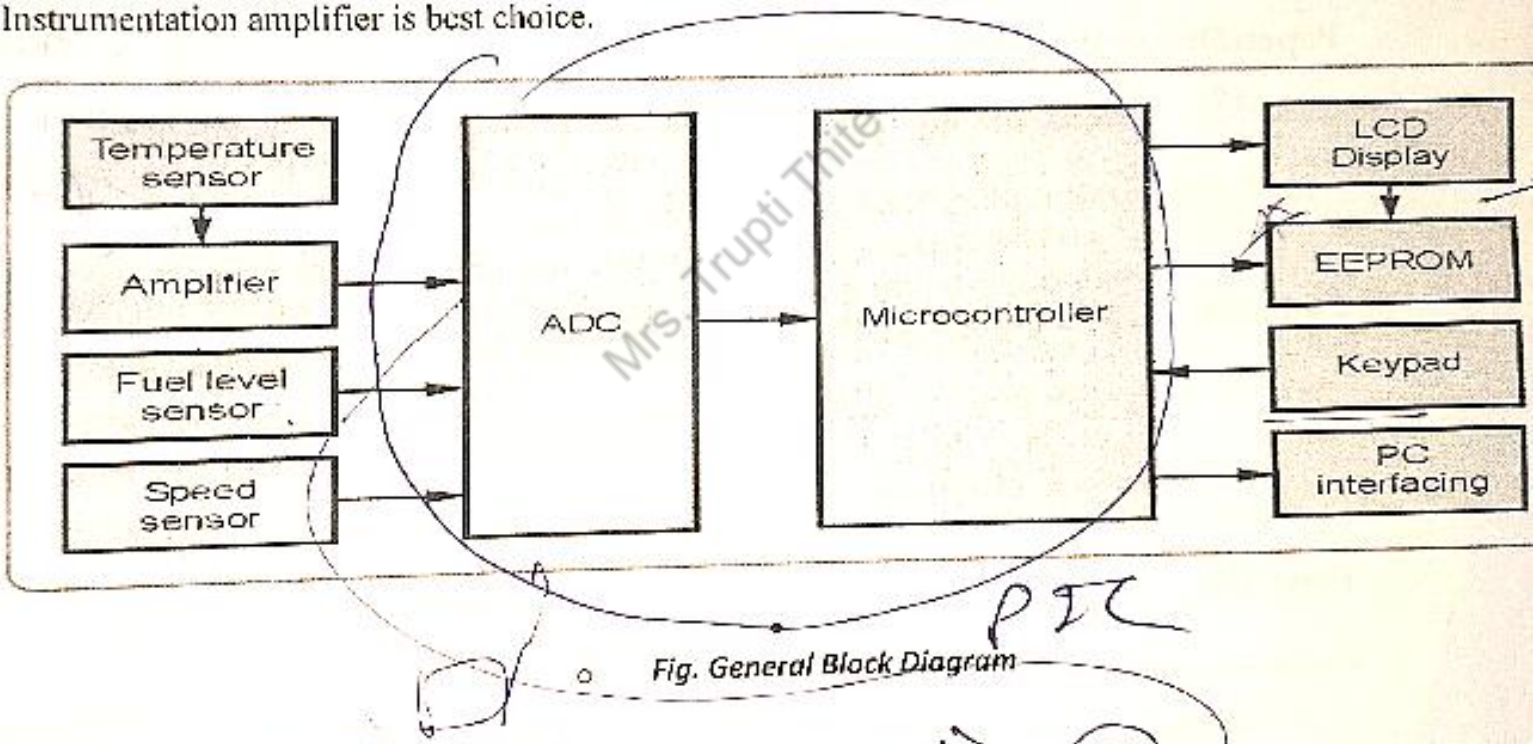


Fig. General Block Diagram

Step 5-Signal Conditioning Circuit

Step 6-Algorithm

1. Initialize the PIC ports, LCD etc.
2. Provide set points for various controlling actions
3. Accept the signals (Analog or digital) on ADC pins.
4. Check the set conditions
5. If not met or exceed get the indication by beep, display on LCD, and making devices on and Off
6. Store the current records
7. If everything is set right, continues from step 3.

Design of PIC Test Board.

Step1: Design of Power Supply:

The microcontroller selected is PIC 18F4550 which works on the frequency of oscillator ranging from 0 to 20 MHz and requires power supply of + 5 or ±12 V. A simple circuit design for +5 V is shown in Fig.

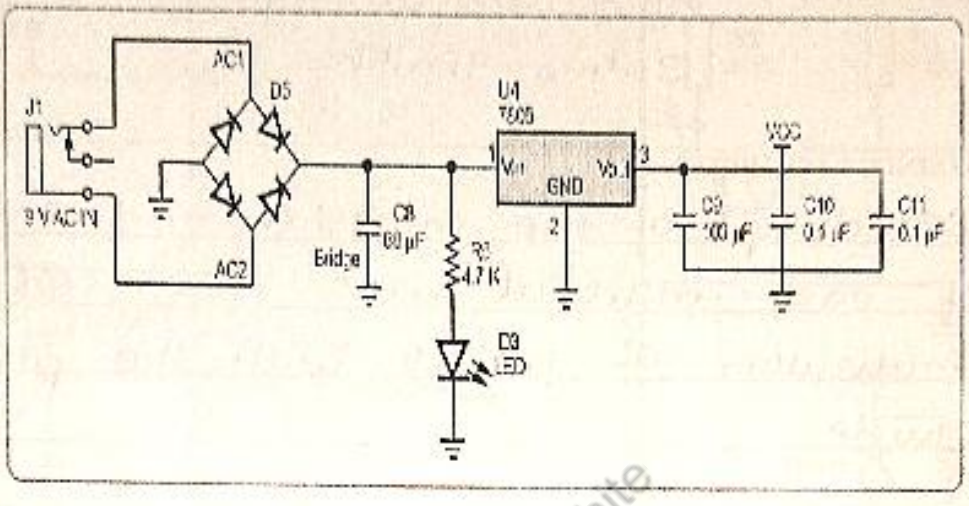


Fig. Design of +5V Power Supply

Step2: Design of Clock Circuit:

The Quartz crystal is connected to OSC1 and OSC2 pin in order to synchronize the operation of all components connected with internal and external means. The values for C1 and C2 are selected according to the crystal frequency for stabilizing the oscillator pulses. In general with quartz crystal 22-33 µF is preferred.

Step3: Design of Reset Circuit:

The RC high pass filter with C=0.1µF along with 20 kΩ register is connected to MCLR pin. When high pulse appear on it, resets the contents of inter registers and SFRS to initial value.

Step4: Configuration of Port

PIC has 33 I/O lines which can be configured as input and output using TRISX register as

- If TRISX = 0 —Ports (A-E) are configured as output ports
- If TRISX = 1 —Ports (A-E) are configured as input ports.

Step 5: Connection Diagram:

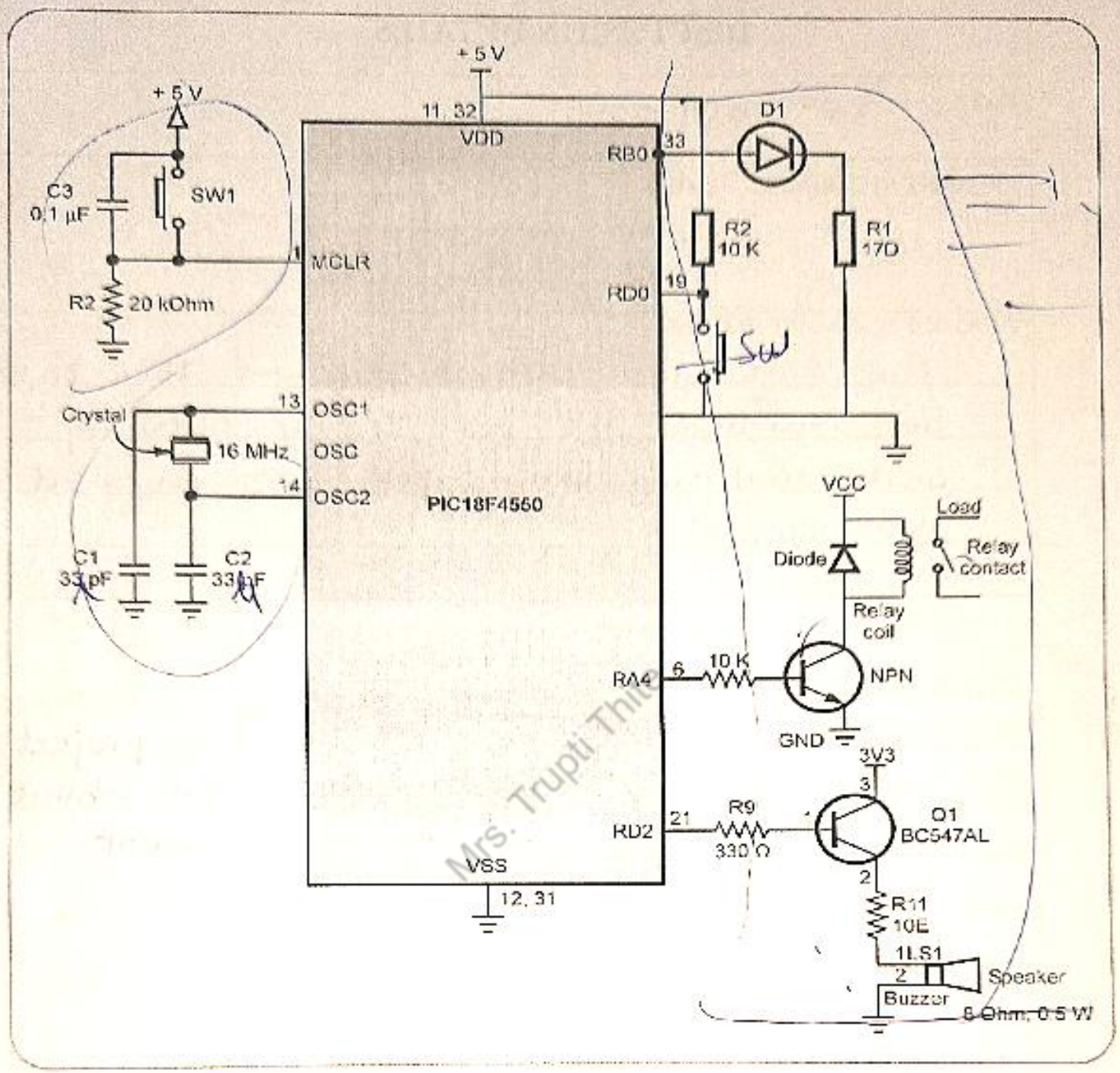


Fig.Minimum connection Diagram PIC Test board

Step 6 : Algorithm

Step 1 : Initialize TRIS SFR for direction control
Step 2 :Check status of switch
Step 3 : If closed, load bit pattern to glow LED, switch on buzzer and energizes the relay.
Step 4 :Otherwise wait till closer
Step 5 :Transfer content to Port register
Step 6 :Wait for some time i.e. delay
Step 7 :Load same or different data sequence of LED glowing
Step 8 :Continue Go to step 2