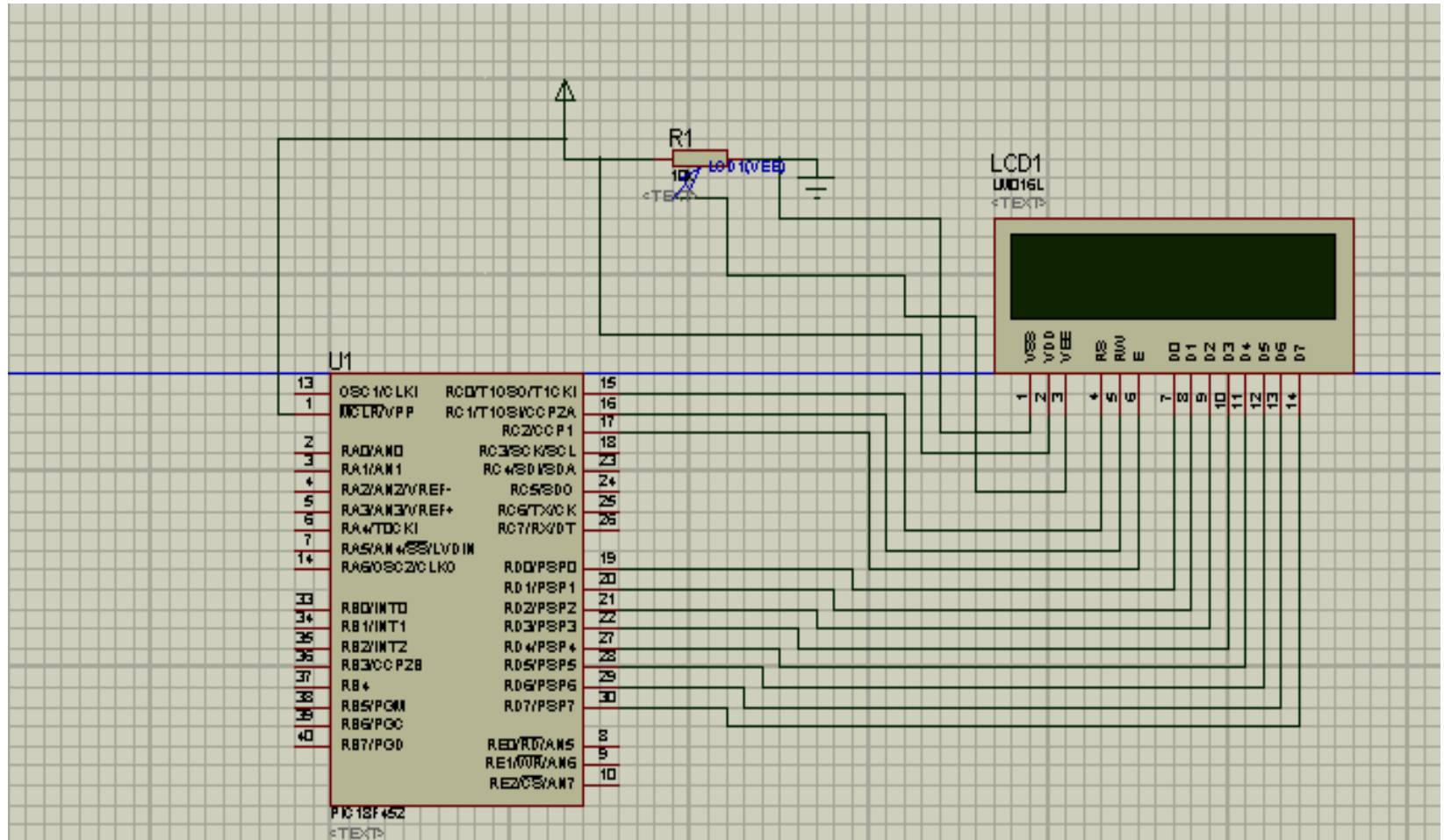


Interfacing Diagram LCD



Interfacing of LCD to PIC18F4520

- `#include<P18f4520.h>`
- `#define rs PORTCbits.RC0;`
- `#define rw PORTCbits.RC1;`
- `#define en PORTCbits.RC2;`
- `#define LDATA PORTD;`
- `void command(unsigned char);`
- `void data(unsigned char);`
- `void delay(void);`
- `void main(void)`

- {
- TRISCbits.TRISC0=0;
- TRISCbits.TRISC1=0;
- TRISCbits.TRISC2=0;
- TRISD=0x00;
- while(1)
- {

- `command(0x38);`
- `delay();`
- `command(0x0e);`
- `delay();`
- `command(0x01);`
- `delay();`
- `command(0x80);`
- `delay();`

```
data('A');
```

```
delay();
```

```
data('V');
```

```
delay();
```

```
data('C');
```

```
delay();
```

```
data('O');
```

```
delay();
```

```
data('E');
```

```
delay();
```

```
data('E');
```

```
delay();
```

```
data('E');
```

```
delay();
```

```
data('E');
```

```
delay();
```

```
}
```

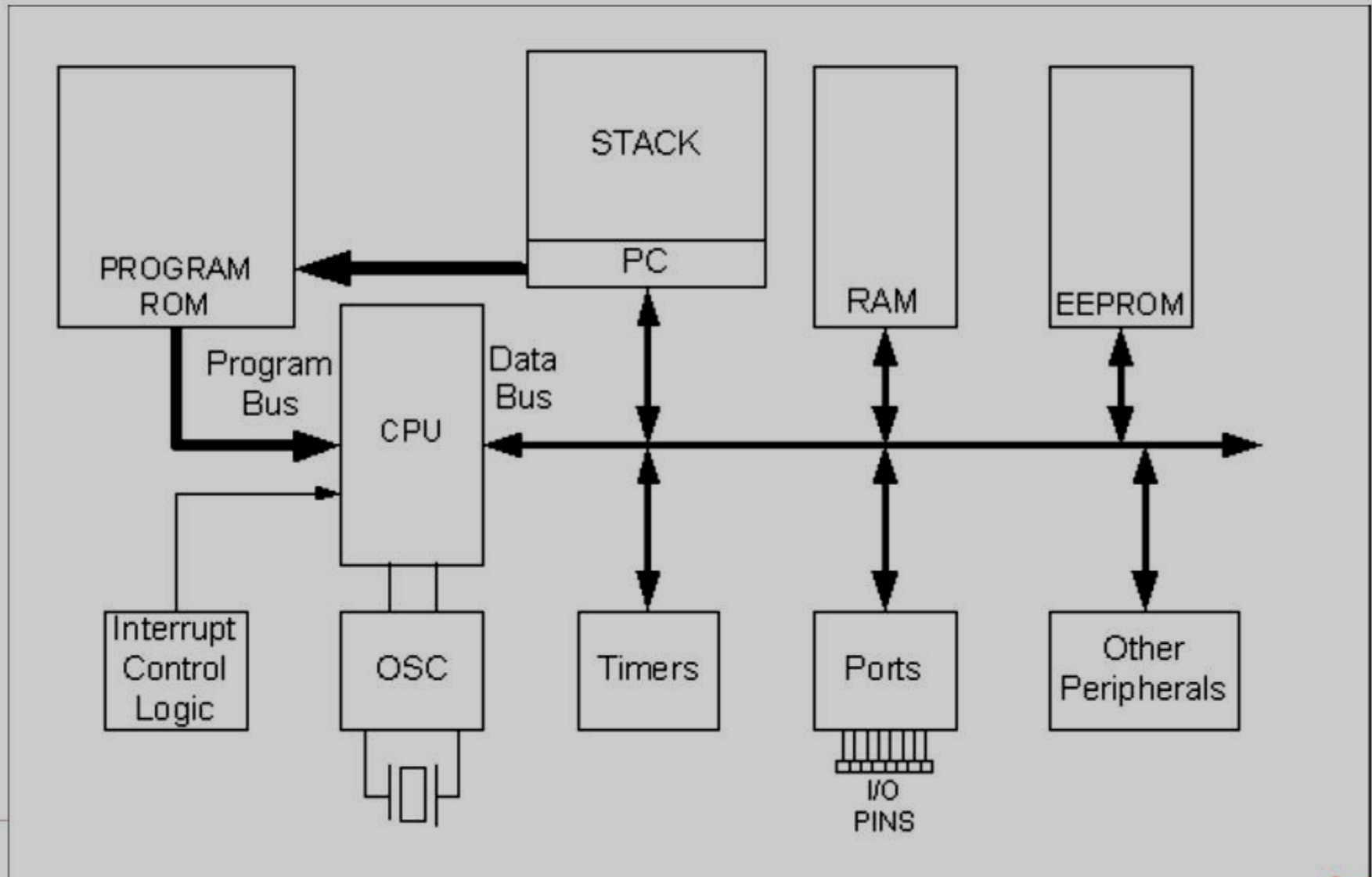
```
}
```

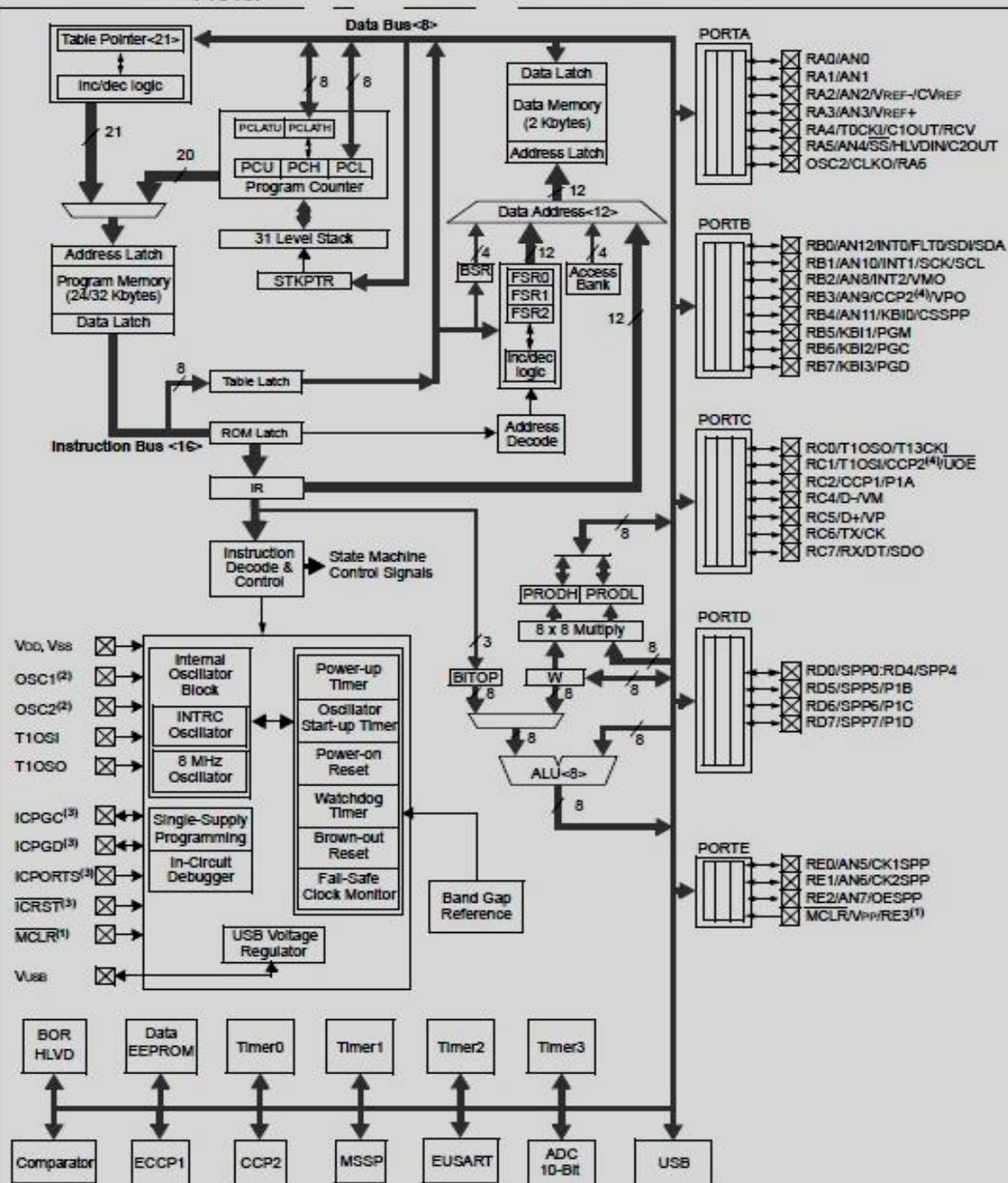
- void command(unsigned char value)
- {
- LDATA PORTD=value;
- rs PORTCbits.RC0=0;
- rw PORTCbits.RC1=0;
- en PORTCbits.RC2=1;
- delay();
- en PORTCbits.RC2=0;
- }

```
void data(unsigned char value)
{
    LDATA PORTD=value;
    rs PORTCbits.RC0=1;
    rw PORTCbits.RC1=0;
    en PORTCbits.RC2=1;
    delay();
    en PORTCbits.RC2=0;
}

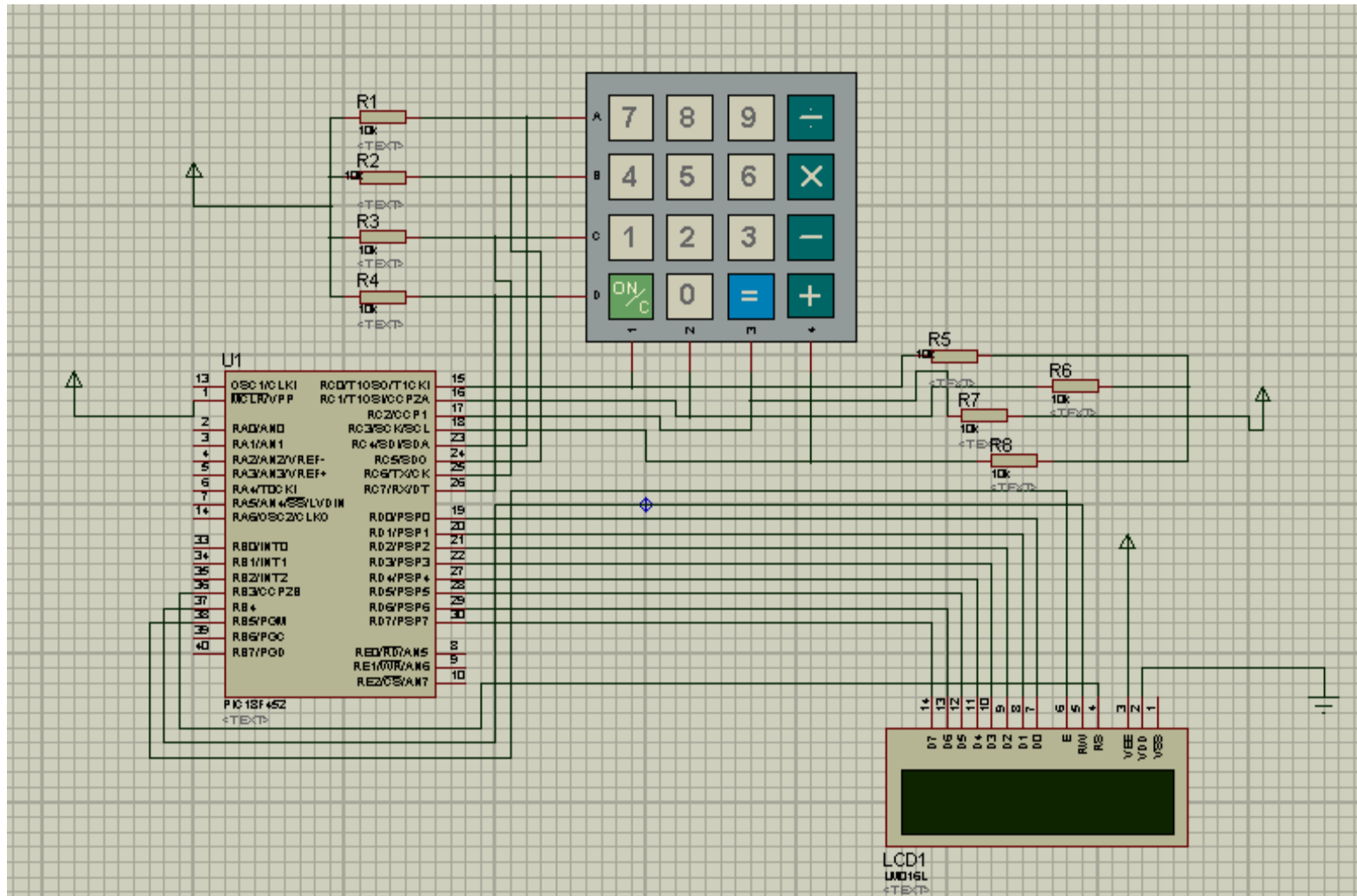
void delay(void)
{
    unsigned int i;
    for(i=0;i<5000;i++);
}
```

Simplified View of a PIC Microcontroller





Interfacing Diagram of keypad



Interfacing of Keypad to PIC

```
#include<p18f452.h>
#define display PORTD
#define rs PORTBbits.RB3
#define rw PORTBbits.RB4
#define en PORTBbits.RB5
#define C1 PORTCbits.RC0
#define C2 PORTCbits.RC1
#define C3 PORTCbits.RC2
#define C4 PORTCbits.RC3
#define R1 PORTCbits.RC4
#define R2 PORTCbits.RC5
#define R3 PORTCbits.RC6
#define R4 PORTCbits.RC7
```

```
void delay(void);  
void cmd(unsigned char);  
void data(unsigned char);  
void main(void)  
{  
    TRISB=0X00;  
    TRISD=0X00;  
    TRISCbits.TRISC0=1;  
    TRISCbits.TRISC1=1;  
    TRISCbits.TRISC2=1;  
    TRISCbits.TRISC3=1;  
    TRISCbits.TRISC4=0;  
    TRISCbits.TRISC5=0;  
    TRISCbits.TRISC6=0;  
    TRISCbits.TRISC7=0;
```

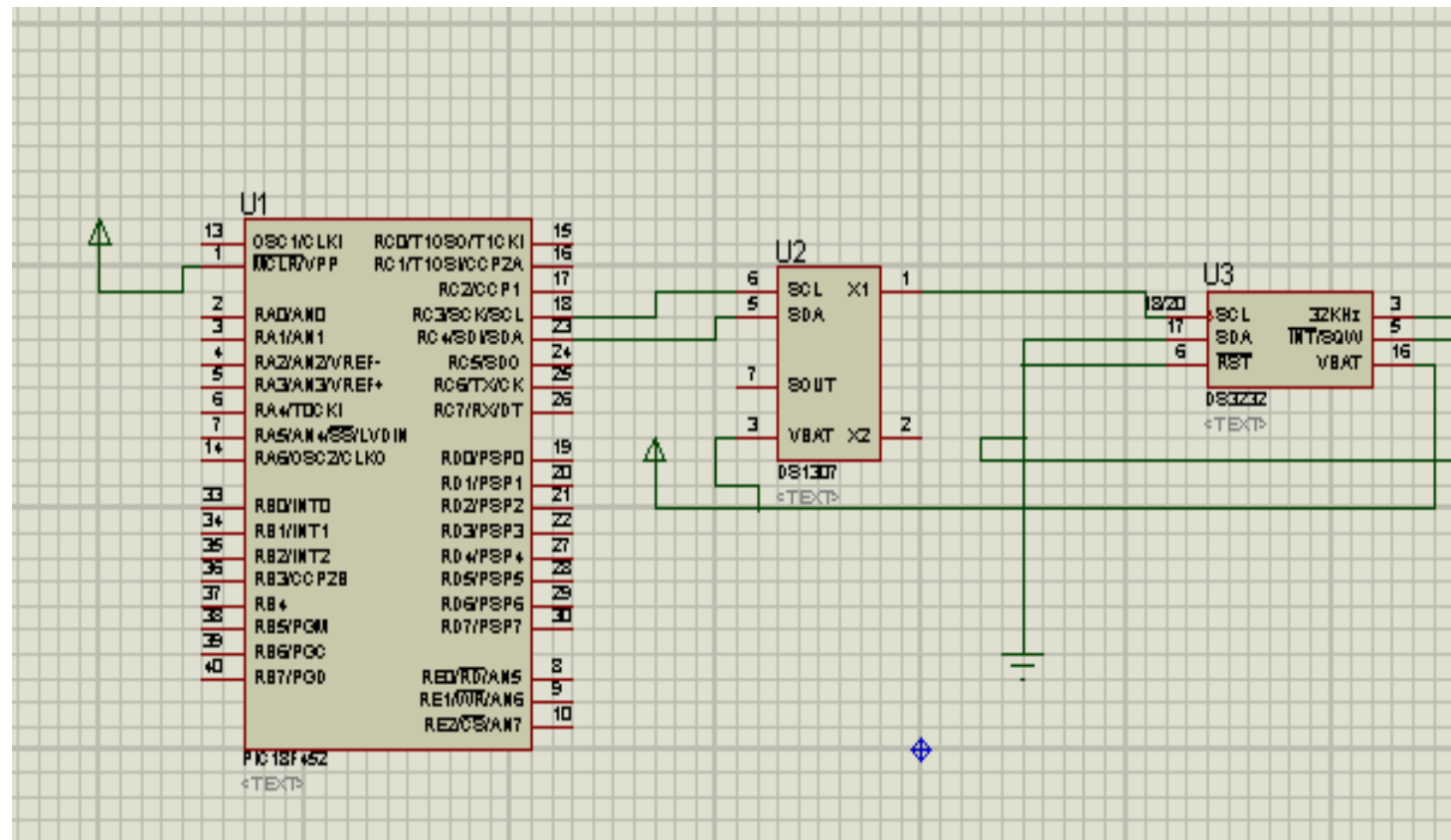
```
cmd(0x38);  
delay();  
cmd(0x0e);  
delay();  
cmd(0x01);  
delay();  
cmd(0x06);  
delay();  
cmd(0x80);  
delay();  
while(1)
```

```
{  
R1=0;R2=1;R3=1;R4=1;  
delay();  
if(C1==0){ data('7');}  
if(C2==0){ data('8');}  
if(C3==0){ data('9');}  
if(C4==0){ data('/');}  
R2=0;R1=R3=R4=1;  
delay();  
if(C1==0){ data('4');}  
if(C2==0){ data('5');}  
if(C3==0){ data('6');}  
if(C4==0){ data('*');}
```

```
R3=0;R1=R2=R4=1;
delay();
if(C1==0){ data('1');}
if(C2==0){ data('2');}
if(C3==0){ data('3');}
if(C4==0){ data('-');}
R4=0;R1=R2=R3=1;
delay();
if(C1==0){ data('C');}
if(C2==0){ data('0');}
if(C3==0){ data('=');}
if(C4==0){ data('+');}
}
}
```

```
void cmd(unsigned char value)
{
display=value;
rs=0;
rw=0;
en=1;
delay();
en=0;
}
void data(unsigned char value)
{
display=value;
rs=1;
rw=0;
en=1;
delay();
en=0;
}
void delay(void)
{
unsigned int i;
for(i=0;i<5000;i++);
}
```


Interfacing Diagram of RTC to SPI



Interfacing of RTC to SPI/PIC18f452

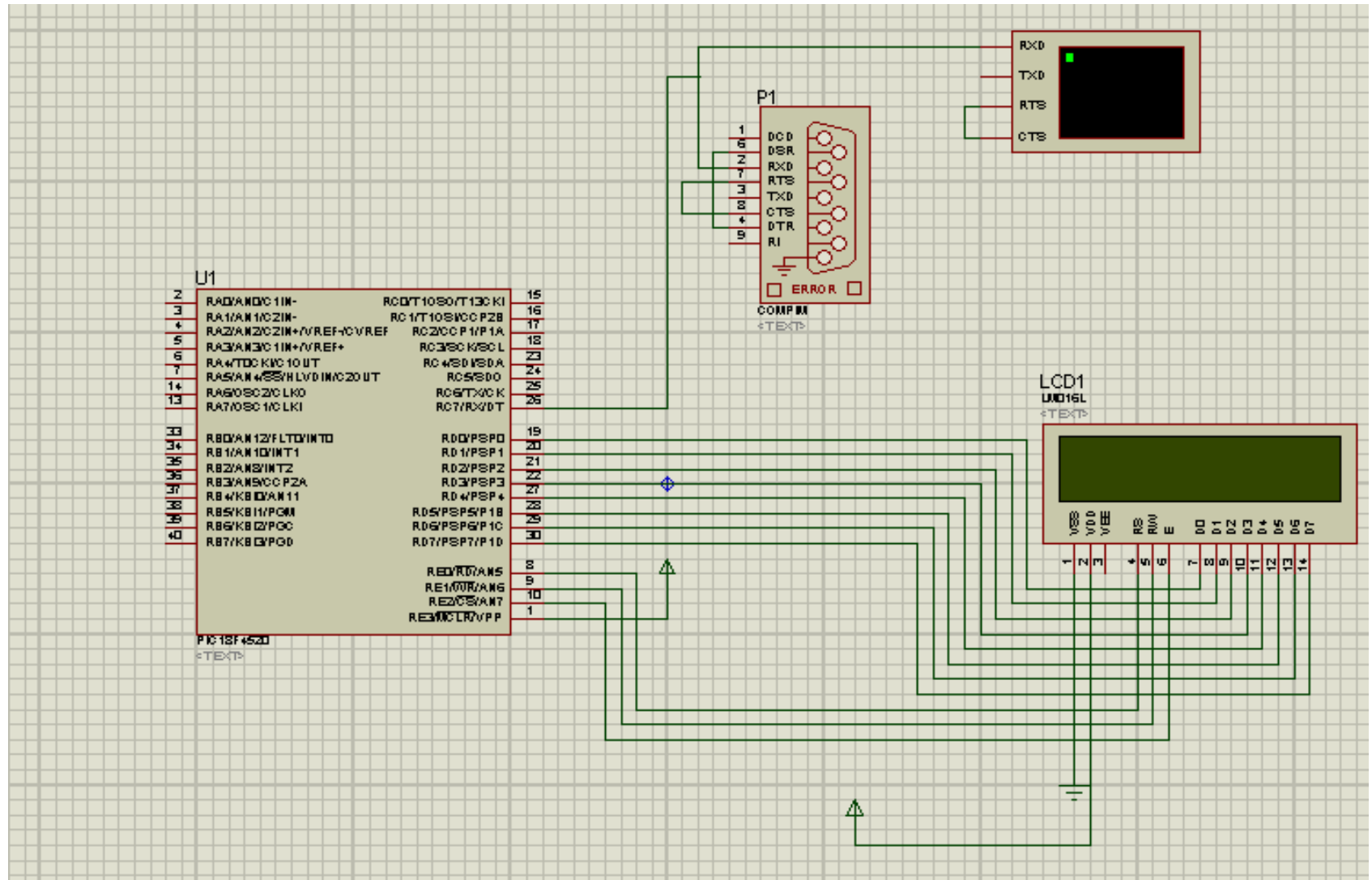
```
#include<p18f452.h>
unsigned char SPI (unsigned char);
void DELAY(int ms);
void main()
{
    SSPSTAT=0;
    SSPCON1=0X22;
    TRISC=0;
    TRISCbits.TRISC4=1;
    TRISCbits.TRISC7=1;
    PORTCbits.RC2=1;
    DELAY(1);
```

```
SPI(0X8F);  
SPI(0X00);  
PORTCbits.RC2=0;  
DELAY(1);  
PORTCbits.RC2=1;  
SPI(0X55);  
SPI(0X58);  
SPI(0X16);  
SPI(0X3);  
SPI(0X19);  
SPI(0X10);  
SPI(0X04);  
PORTCbits.RC2=0;  
DELAY(1);  
}
```

```
unsigned char SPI(unsigned char mybyte)
{
    SSPBUF=mybyte;
    while(!SSPSTATbits.BF);
    return SSPBUF;
}

void DELAY(int ms)
{
    unsigned int i;
    for(i=0;i<5000;i++);
}
```

Interfacing Diagram to Receive Data serially



Program to Receive data Serially:

```
#include <P18f4520.h>
#define ldata PORTD;
#define rs PORTEbits.RE0;
#define rw PORTEbits.RE1;
#define en PORTEbits.RE2;
void delay(void);
void lcmdcmd(unsigned char value);
void lcddata(unsigned char value);
void main()
```

```
{  
  unsigned char data;  
  TRISD = 0x00;  
  TRISE=0x00;  
  TRISCbits.TRISC7=1;  
  RCSTA=0X90;  
  SPBRG=12;  
  lcdcmd(0x38);  
  delay();  
  lcdcmd(0x0E);  
  delay();  
  lcdcmd(0x01);  
  delay();  
  lcdcmd(0x06);  
  delay();  
}
```

```
{  
  while(PIR1bits.RCIF==0);  
    {  
      data=RCREG;  
      lcdcmd(0x80);  
      lcddata(data);  
    }  
}  
}  
void delay(void)  
{  
    unsigned int i;  
    for(i=0;i<10;i++);  
}
```



```
void lcdcmd (unsigned char value)
```

```
{
```

```
    ldata=value;
```

```
    rs=0;
```

```
    rw=0;
```

```
    en=1;
```

```
    delay();
```

```
    en=0;
```

```
}
```

```
void lcddata (unsigned char value)
```

```
{
```

```
    ldata=value;
```

```
    rs=1;
```

```
    rw=0;
```

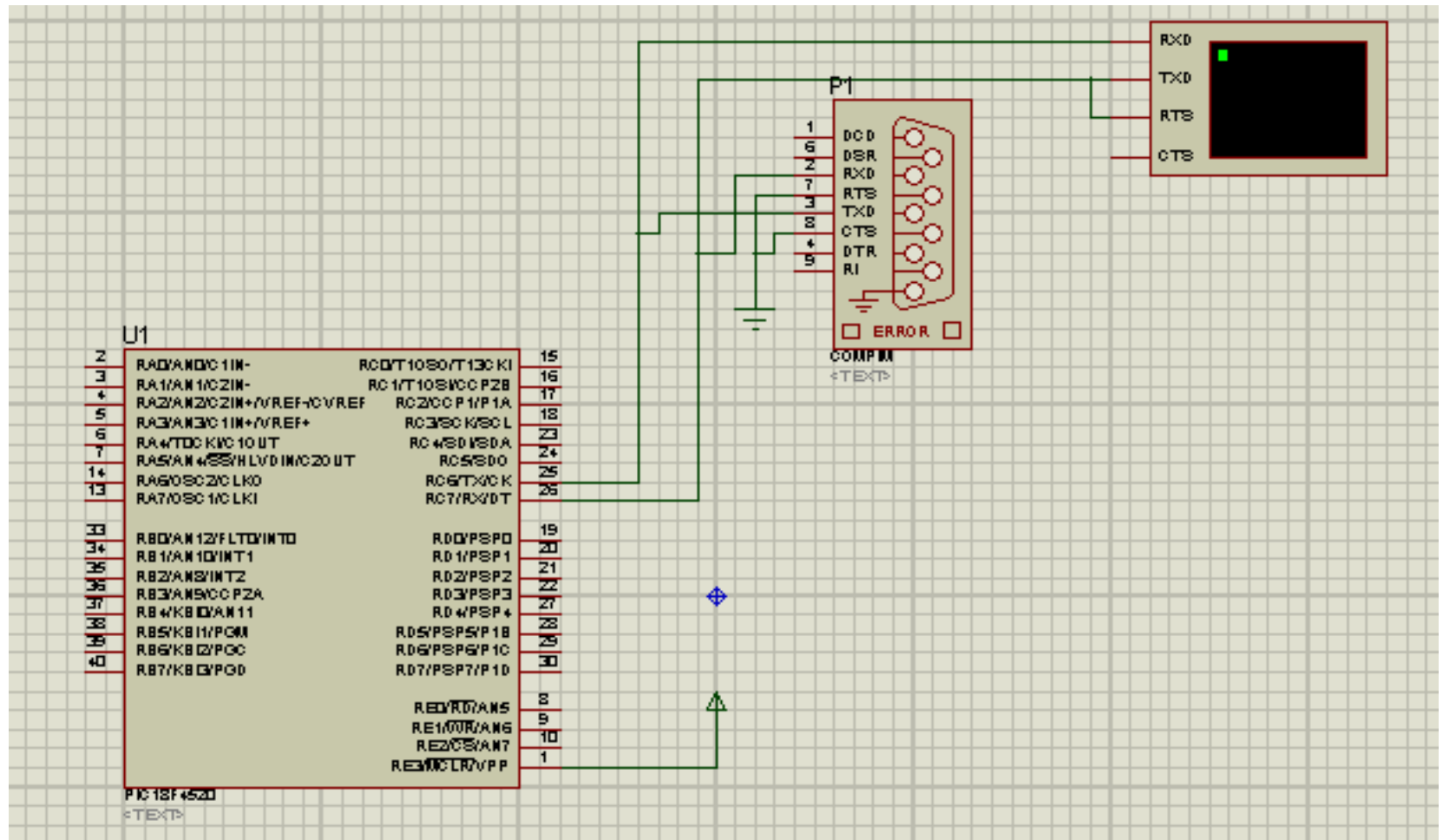
```
    en=1;
```

```
    delay();
```

```
    en=0;
```

```
}
```

Interfacing Diagram to Transmit Data Serially

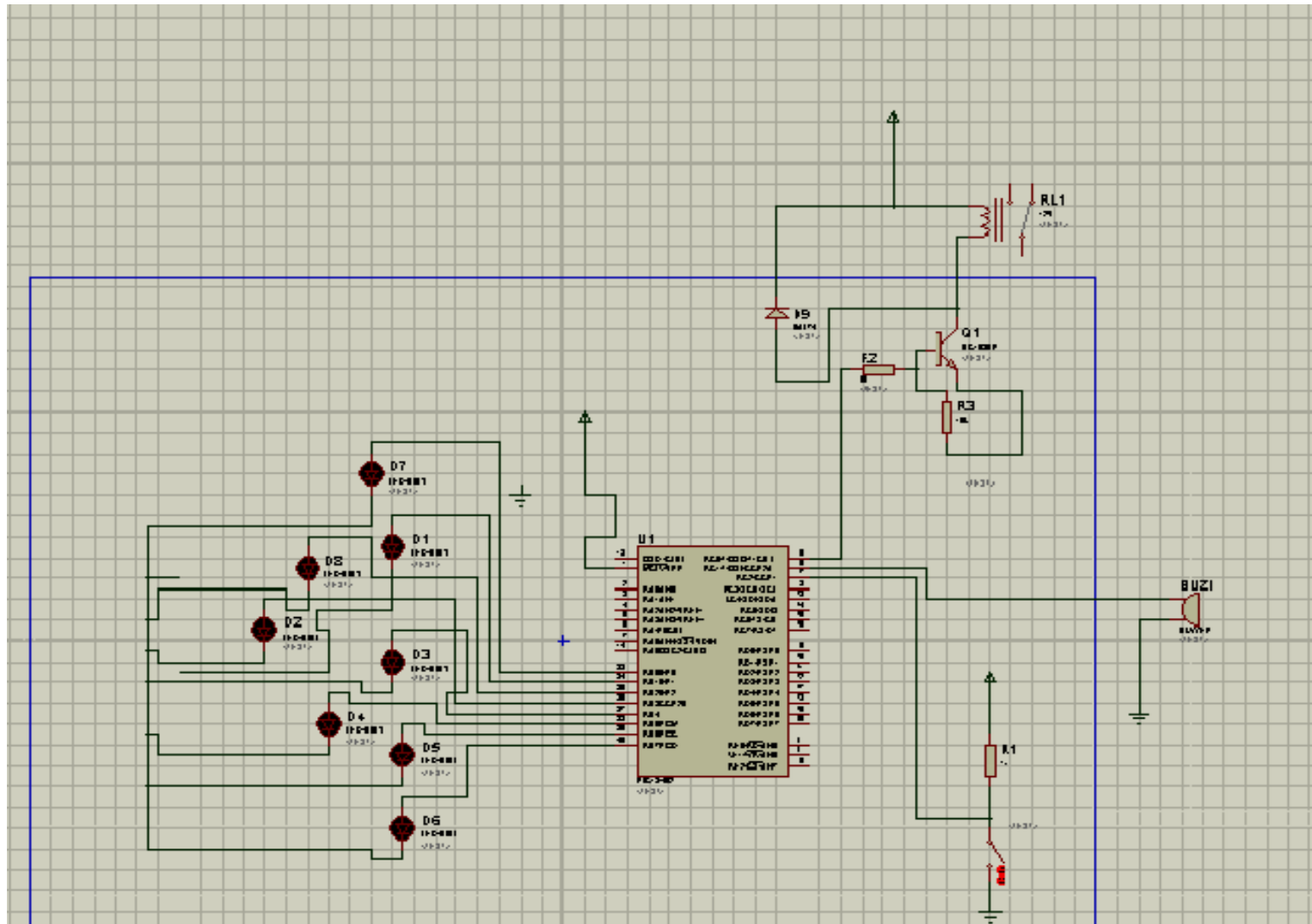


Program to Transmit data Serially:

```
#include<p18f4520.h>
void delay(void);
void sendUSART (void);
unsigned char text[]="\n\rSAMARTH";
unsigned int i=0;
void main(void)
{
    TXSTA=0X20;
    SPBRG=12;
    TXSTAbits.TXEN=1;
    RCSTAbits.SPEN=1;
    sendUSART ();
    while(1);
}
```

```
void delay( void)
{
  unsigned int i;
  for(i=0;i<=1000;i++);
}
void sendUSART (void)
{
  while(text[i]!='\0')
  {
    while(PIR1bits.TXIF==0);
    TXREG=text[i];
    i++;
  }
}
```

Interfacing Diagram of LED,Relay Switch Buzzer



Program for LED, Relay, Switch and Buzzer

```
#include<p18f452.h>
#define relay PORTCbits.RC0
#define buzzer PORTCbits.RC1
#define sw PORTCbits.RC2
#define LED0 PORTBbits.RB0
#define LED1 PORTBbits.RB1
#define LED2 PORTBbits.RB2
#define LED3 PORTBbits.RB3
#define LED4 PORTBbits.RB4
#define LED5 PORTBbits.RB5
#define LED6 PORTBbits.RB6
#define LED7 PORTBbits.RB7
void delay(void);
void main(void)
```

```
{  
TRISCBits.TRISC0=0;  
TRISCBits.TRISC1=0;  
TRISCBits.TRISC2=1;  
TRISB=0X00;  
while(1)  
{  
if (sw==0)  
{  
relay=1;  
buzzer=1;  
LED0=1;  
delay();  
LED0=0;  
delay();  
LED1=1;  
delay();  
LED1=0;  
delay();  
LED2=1;  
delay();  
LED2=0;  
delay();  
}}
```

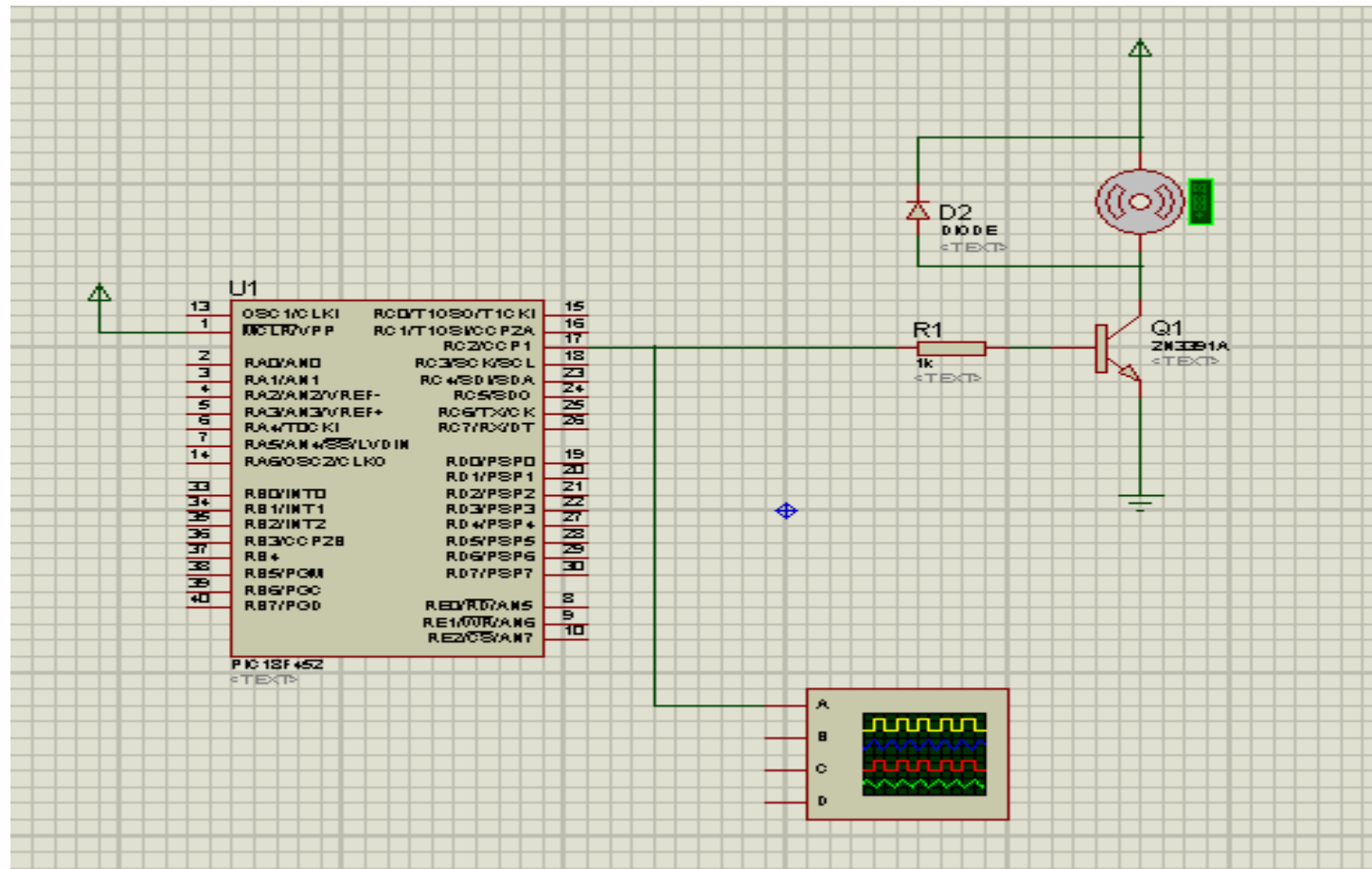
```
LED3=1;  
delay();  
LED3=0;  
delay();  
LED4=1;  
delay();  
LED4=0;  
delay();  
LED5=1;  
delay();  
LED5=0;  
delay();  
LED6=1;  
delay();  
LED6=0;  
delay();  
LED7=1;  
delay();  
LED7=0;  
delay();  
}
```



```
else
{
relay=0;
delay();
buzzer=0;
delay();
LED7=1;
delay();
LED7=0;
delay();
LED6=1;
delay();
LED6=0;
delay();
LED5=1;
delay();
LED5=0;
delay();
LED4=1;
delay();
LED4=0;
delay();
```

```
LED3=1;
delay();
LED3=0;
delay();
LED2=1;
delay();
LED2=0;
delay();
LED1=1;
delay();
LED1=0;
delay();
LED0=1;
delay();
LED0=0;
delay();
}
}
}
void delay(void)
{
  unsigned int i;
  for(i=0;i<5000;i++);
}
```

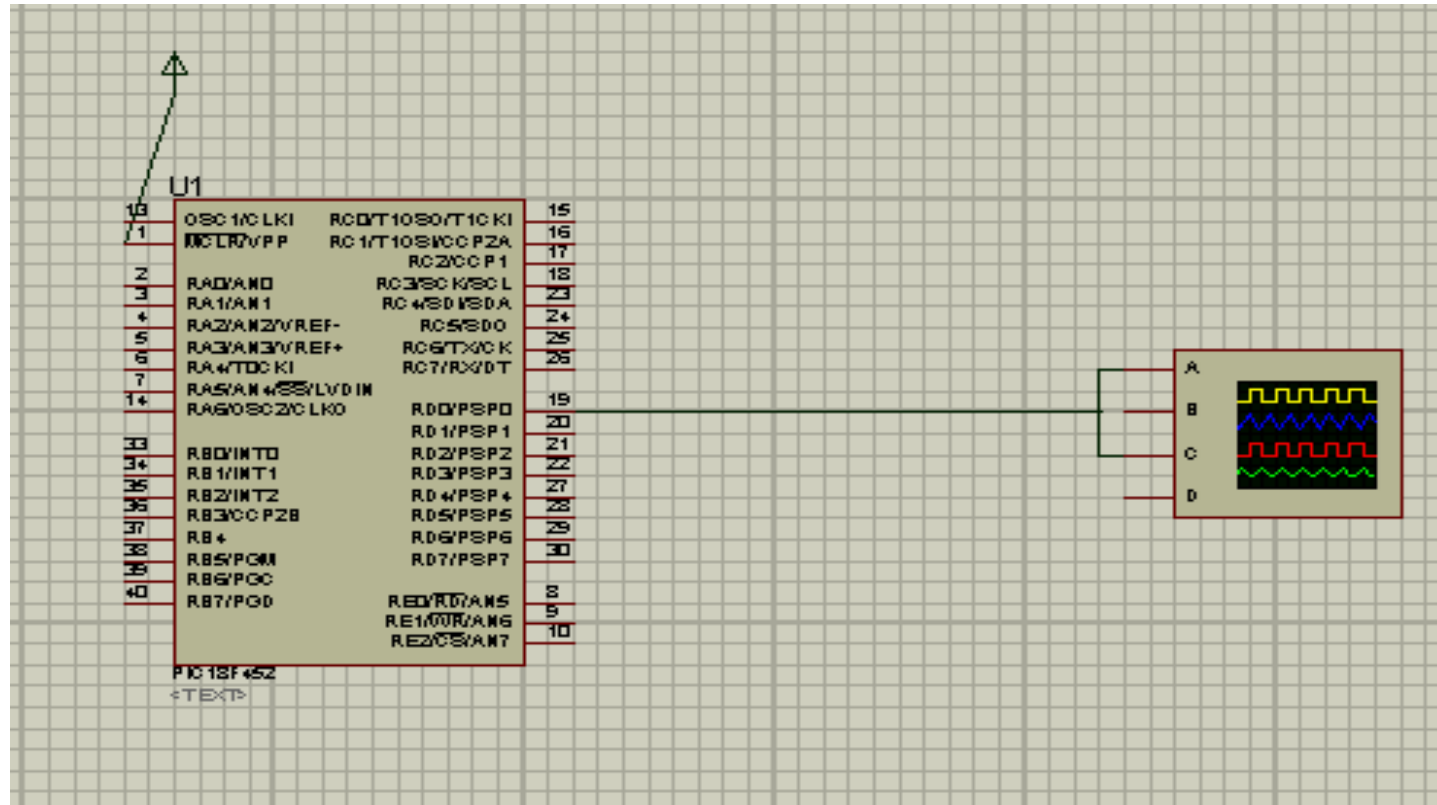
Interfacing Diagram for DC Motor



Program for DC Motor

```
#include<p18f452.h>
void main()
{
  CCP1CON=0;
  PR2=249;
  CCPR1L=186;
  TRISCbits.TRISC2=0;
  T2CON=0X01;
  CCP1CON=0X3C;
  TMR2=0;
  T2CONbits.TMR2ON=1;
  while(1)
  {
    PIR1bits.TMR2IF=0;
    while(PIR1bits.TMR2IF=0);
  }
}
```

Interfacing Diagram for Square Wave



Program for Square Wave

```
#include<p18f452.h>
void timer(void);
void main(void)
{
    TRISDbits.TRISD0=0;
    while(1)
    {
        PORTDbits.RD0=1;
        timer();
        PORTDbits.RD0=0;
        timer();
    }
}
```

```
void timer(void)
{
    TOCON=0X08;
    TMR0H=0X35;
    TMR0L=0X00;
    TOCONbits.TMR0ON=1;
    while(INTCONbits.TMR0IF==0);
    TOCONbits.TMR0ON=0;
    INTCONbits.TMR0IF=0;
}
```

Interfacing of ADC to PIC18F4520

```
#include <P18f4520.h>
#define display PORTD
#define rs PORTBbits.RB0
#define rw PORTBbits.RB1
#define enable PORTBbits.RB2
void delay(void);
void bcd(unsigned char adout);
void command(unsigned char);
void data(unsigned int);
void main(void)
```



```
{  
unsigned char adout, L, H,d1,d2,d3;  
TRISD=0x00;  
TRISB=0x00;  
TRISAbits.TRISA0=1;  
ADCON1 = 0XCE;  
ADCON0=0x81;  
    command(0x38);  
    delay();  
    command(0x01);  
    delay();  
    command(0x0E);  
    delay();  
    command(0x80);  
    delay();  
}
```

```
while(1)
{
    delay();
    ADCON0bits.GO = 1;
    while(ADCON0bits.DONE ==1);
    H = ADRESH;
    L = ADRESL;
    L>>=2;
    L =0x3F;
    H<<=6;
    H =0xC0;
    adout=L | H;
    PORTD=adout;
    d1=(adout/100)%10;
    d2=((adout/10)%10);
    d3=(adout%10);
```

```
command(0x80);  
data('A');  
command(0x81);  
data('D');  
command(0x82);  
data('C');
```

```
command(0x84);  
bcd(d3);  
command(0x85);  
bcd(d2);  
command(0x86);  
bcd(d1);  
delay();  
}  
}
```

```
void delay(void)
{
    unsigned int i;
    for(i=0;i<5000;i++);
}
```

```
void command(unsigned char value)
{
    display=value;
    rs=0;
    rw=0;
    enable=1;
    delay();
    enable=0;
}
```

```
void data(unsigned int value)
{
    display=value;
    rs=1;
    rw=0;
    enable=1;
    delay();
    enable=0;
}

void bcd(unsigned char adout)
{
    unsigned char x;
    x=adout|0x30;
    data(x);
}
```

Interfacing diagram for ADC

