```
//-------------------------------------------------------------------------------------------------//
//B.V.Bhoomaradi College of Engineering and Technology,
//Department of Electronics and Communication.
//Hubli-32
//Project Team members: Ajay Bhat, Sagar Karur, Sagar Medikeri, Suyash Ugare.
//Project Guides: Prof. B.L.Desai and Sunita Shirahatti.
//Date: 15/03/09.

#include<reg52.h>

//sbit dav=P1^0;       //input-data available pin.
//sbit fb=P3^2;        //input-feed enable button.
sbit mux=P3^0;         //read foot step through 16:1 mux.
sbit enter=P3^1;       //input-press to enter mode.
sbit latch=P3^2;       //use to latch data in basic mode(mode 1).
sbit mod=P3^3;         //interrupt pin.
sbit apr_en=P3^4;
sbit busy1=P3^6;       // APR1 busy indicator - high during playback
sbit busy2=P3^5;       // APR2 busy indicator - high during playback
//sbit te1=P3^6;
//sbit te2=P3^7;
//unsigned char arr[16];          //variable to store mode pattern.
//unsigned char N;                //number of steps entered.
//unsigned char m3[];             //mode 3.

unsigned char i,k,k1,once;              //
unsigned char mode=0;              //indicates the mode in which it is functioning.
void delay(unsigned char time);    // delay is common for both APR & LCD........
void mode1(void);                  //basic mode.
void mode2(void);                  //pattern 1.
void mode3(void);                  //pattern 2.
//void mode4(void);                //feed mode.
//void mode4(void);
void stepread(void);
void send(unsigned char);
void lcd(unsigned char);
void delay_propogation(void);
void apr1(void);
void apr2(void);
//void timecalculation(unsigned char,unsigned char);
//void startcounter(void);
//void keypadread(void);

//-----------------------------------------MAIN-------------------------------------//


void main(void)
{
      P0=0x00; P1=0x00; P2=0x00; P3=0x00;//initialising all ports.
      mux=1; enter=1; mod=1; busy1=1; busy2=1;
//    fb=1;
//    P2=0x0F;                          //inputs made high for key pad.
      TCON=0x04;                        //interrupt 1 is low-edge triggered
```

```
      IE=0x84;                          //enable interrupt 1;
      i=1;
      lcd('0');                         //welcome. please select the mode.
      while(1)
      {
            switch (mode)
            {
                  case 1:        if(enter==1)
                                    {
                                    mode1();
                                    }
                                    break;
                  case 2:        if(enter==1)
                                    {
                                    mode2();
                                    }
                                    break;
                  case 3:           if(enter==1)
                                    {
                                    mode3();
                                    }
                                    break;
                  default:    break;
            }
      }

}

//---------------------------------------------MODE 1-------------------------------//


void mode1(void)                              //basic mode of operation.
{
      lcd('4');
      k1=20;                                  // set k1 equal to a value which cannot be
taken by k, i.e. k1>15
      while(1)
      {
            for(k=0;k<16;k++)
            {
                  P1=k;
                  delay_propogation();        //compensation for propogation delay.
                  if(mux==0)
                  {
                        if(k1!=k)                       // actually, apr wont play
immediately when stepped on a different tile, it play in the immediate second looping
due to tile press.
                        {
                              i=1;
                        }
                        k1=k;                                         //this has to be
modified when APR data is to be sent.
                     latch=0;
                       delay_propogation();
```

```
                        latch=1;
                        delay_propogation();        //latch the data to the decoder.
                        latch=0;
                        delay_propogation();        //the stepped tile glows.
                        if (i<=2)
                        {
                              if(k<8)
                              {
                                    apr1();
                              }
                              else
                              {
                                    apr2();
                              }
                        }
                  }
            }
      }
}

//-------------------------------------------------MODE 2----------------------//


void mode2(void) //pre fetched pattern 1.
{
      unsigned char
m1[16]={0x00,0x02,0x05,0x06,0x09,0x0A,0x0D,0x0E,0x0F,0x0C,0x0B,0x08,0x07,0x04,0x03,0x01
};
      lcd('5');              //mode 1 selected. time:
repeat2:for(k=0;k<16;k++)
            {
                  send(m1[k]);
            }
            if(k==16)
            goto repeat2;
}

//-------------------------------------------------MODE 3-----------------------------
----------//


void mode3(void)                  //pre fetched pattern 2.
{
      unsigned char
m2[16]={0x00,0x03,0x05,0x07,0x09,0x0B,0x0D,0x0F,0x0E,0x0C,0x0A,0x08,0x06,0x04,0x02,0x01
};
      lcd('6');                           //mode 2 selected. time:
repeat3: for(k=0;k<16;k++)
            {
                  send(m2[k]);
            }
            if(k==16)
            goto repeat3;
}
```

```
//-----------------------------------------INT1.-----------------------------------
--------//


void extint1(void) interrupt 2//increment mode.
{
            mode++;
            if(mode==1)
            {lcd('1');}
            else if(mode==2)
            {lcd('2');}
            else if(mode==3)
            {lcd('3');}
            else if(mode>3)
            {
                  mode=1;
                  lcd('1');
            }

}

//-------------------------------------------------SEND-----------------------------
--------//


void send(unsigned char arr)
{
            P1=arr;                         //this has to be modified when APR data is
to be sent.
            latch=0;
            delay_propogation();                  //latch the data to the decoder.
            latch=1;
            delay_propogation();
            latch=0;
            delay_propogation();
            delay_propogation();
            stepread();
}
//--------------------------------STEPREAD-----------------------------------------
------//


void stepread(void)
{
      while(mux==1);      //take care of complementary output of the mux.
      apr();
      //timecalculation(TH0,TL0);
      //stepdelay()
}
//--------------------------------PROPOGATION DELAY ----------------------------------
-----------//
```

```
void delay_propogation(void)
{
      unsigned int d;
      for(d=0;d<5000;d++);
}


//-----------------------------------------------------apr 1-----------------------------
----------//
void apr1(void)
{
      i++;
      apr_en=0;                           // latch data
      delay(250);                         // give sufficient delay for APR's decoder to
latch data to APR
      apr_en=1;
      while(busy1==0);
}

//-----------------------------------------------------apr 2-----------------------------
----------//
void apr2(void)
{
      i++;
      apr_en=0;                           // latch data
      delay(250);                         // give sufficient delay for APR's decoder to
latch data to APR
      apr_en=1;
      while(busy2==0);
}
//---------------------------------------LCD PROGRAM-------------------------------------
-----//


sfr ldata=0x80;
sbit rs=P2^0;
sbit rw=P2^1;
sbit en=P2^2;
void lcdpress(void);
void lcdcmd(unsigned char value);
void lcddata(unsigned char value);
void lcdbasic(void);
void lcdmode(void);

void lcd(unsigned char z)
{
switch(z)
{
case '0':
{
lcdbasic();
lcdcmd(0x84);
delay(5);
lcddata('W');
```

```
delay(5);
lcddata('E');
delay(5);
lcddata('L');
delay(5);
lcddata('C');
delay(5);
lcddata('O');
delay(5);
lcddata('M');
delay(5);
lcddata('E');
delay(5);
break;
}
case '1':
{
lcdbasic();
lcdcmd(0x84);
delay(5);
lcdmode();
lcddata('1');
delay(5);
lcdcmd(0xC0);
delay(5);
lcdpress();
delay(5);
break;
}
case '2':
{
lcdbasic();
lcdcmd(0x84);
delay(5);
lcdmode();
delay(5);
lcddata('2');
delay(5);
lcdcmd(0xC0);
delay(5);
lcdpress();
break;
}
case '3':
{
lcdbasic();
lcdcmd(0x84);
delay(5);
lcdmode();
delay(5);
lcddata('3');
delay(5);
lcdcmd(0xC0);
delay(5);
```

```
lcdpress();
break;
}
case '4':
{
lcdbasic();
lcdcmd(0x84);
delay(5);
lcdmode();
lcddata('1');
delay(5);
break;
}
case '5':
{
lcdbasic();
lcdcmd(0x84);
delay(5);
lcdmode();
delay(5);
lcddata('2');
break;
}
case '6':
{
lcdbasic();
lcdcmd(0x84);
delay(5);
lcdmode();
delay(5);
lcddata('3');
break;
}
}
}

void lcdcmd(unsigned char value)
{
ldata=value;
rs=0;
rw=0;
en=1;
delay(50);
en=0;
return;
}

void lcddata(unsigned char value)
{
ldata=value;
rs=1;
rw=0;
en=1;
delay(50);
```

```
en=0;
return;
}

void delay(unsigned char time)
{
      unsigned char i,j;
      for(i=0;i<time;i++)
      {
            for(j=0;j<time;j++);
      }
}

void lcdbasic(void)
{
lcdcmd(0x38);
delay(5);
lcdcmd(0x0E);
delay(5);
lcdcmd(0x01);
delay(5);
lcdcmd(0x06);
delay(5);
return;
}

void lcdmode(void)
{
lcddata('M');
delay(5);
lcddata('O');
delay(5);
lcddata('D');
delay(5);
lcddata('E');
delay(5);
lcddata(':');
delay(5);
return;
}

void lcdpress(void)
{
lcddata('E');
delay(5);
lcddata('N');
delay(5);
lcddata('T');
delay(5);
lcddata('E');
delay(5);
lcddata('R');
delay(5);
lcddata(' ');
```

```
delay(5);
lcddata('-');
delay(5);
lcddata(' ');
delay(5);
lcddata('T');
delay(5);
lcddata('O');
delay(5);
lcddata(' ');
delay(5);
lcddata('B');
delay(5);
lcddata('E');
delay(5);
lcddata('G');
delay(5);
lcddata('I');
delay(5);
lcddata('N');
delay(5);
return;
}



/*void mode4(void)
{
//    lcd();                  //mode 3 selected. time:
//    keypadread();
repeat: for(k=0;k<N;k++)
            {
                  send(m3[k]);
            }
            if(k==N)
            {
                  goto repeat;
            }
} */

/*void extint0(void) interrupt 0
{
} */


/*void keypadread(void)
{
      unsigned char i=0;
      while(fb==1)            //aliter - enter can also be used to do the job of fb.
      {
            if(dav==1)
            {
                  m3[i]=P2;//masking is done while sending data.
                  i++;
```

```
            }

      }
      N=i;
} */

/*void startcounter(void)
{
      T0=1;
      TMOD=0x05;
      TL0=0;
      TH0=0;
      TR0=1;
}

void timecalculation(unsigned char hB,unsigned char lB)//hB-higher byte,lB-lower byte.
{

}*/
```