## BÀI 1: 8 LED sáng dần tắt dần

```
#include <at89x51.h>
#define LED0 P2_0
#define LED1 P2_1
#define LED2 P2_2
#define LED3 P2_3
#define LED4 P2_4
#define LED5 P2_5
#define LED6 P2_6
#define LED7 P2_7

void delay(unsigned int
ms)
{
    unsigned int i,j;
    for (i=0;i<ms;i++)
        for (j=0;j<120;j++)
        {}
}

main(void)
{
    P2 = 0xff;
    delay(300);
    while(1)
    {
        LED0=0;
      delay(300);
        LED1=0;
      delay(300);
        LED2=0;
      delay(300);
        LED3=0;
      delay(300);
       LED4=0;
      delay(300);
      LED5=0;
      delay(300);
        LED6=0;
      delay(300);
        LED7=0;
      delay(300);
        LED0=1;
      delay(300);
        LED1=1;
      delay(300);
       LED2=1;
      delay(300);
        LED3=1;
      delay(300);
        LED4=1;
      delay(300);
        LED5=1;
      delay(300);
        LED6=1;
      delay(300);
        LED7=1;
      delay(300);
    }
}
```

## BÀI 2: Button control LED

```
#include <at89x51.h>

void delay(unsigned int
ms)
{
    unsigned int i,j;
    for (i=0;i<ms;i++)
        for (j=0;j<120;j++)
        {}
}
main(void)
{
    P2_0 = P2_1 = P2_2= P2_3 = 1;
    while(1)
    {
        if(P3_0==0)
        {
            P2_0 = ~P2_0;
            delay(1000);
        }
        else if(P3_1==0)
        {
            P2_1 = ~P2_1;
            delay(1000);
        }
        else if(P3_2==0)
        {
            P2_2 = ~P2_2;
            delay(1000);
        }
        else if(P3_3==0)
        {
            P2_3 = ~P2_3;
            delay(1000);
        }
    }
}
```

## BÀI 3: Hiển thị LED 7 đoạn các số từ 0-99

```
#include <at89x52.h>
unsigned char led[10]={0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f,
0x6f};
int i,j,n;
void delay(unsigned int
ms)
{
    unsigned int p,k;
    for (p=0;p<ms;p++)
        for (k=0;k<120;k++)
        {}
}
void main()
{
    while(1)
    {
        for(j=0; j<=9; j++)
        {
        for(i=0; i<=9; i++)
            {
                for(n=0; n<20; n++)
                {
                P2=0x04;
                P0= led[j];
                delay(5);
                P2=0x00;
                P0= led[i];
                delay(5);
                }
            }
        }
    }
}
```

## BÀI 4: Nút đếm lên, đếm xuống

```
#include <at89x52.h>
unsigned char led[10]={0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f,
0x6f};
int n;

void delay(unsigned int
ms)
{
    unsigned int p,k;
    for (p=0;p<ms;p++)
        for (k=0;k<120;k++)
        {}
}

void main()
{
    P2=0x00;
    n=0;
    P0=led[n];
    while(1)
    {
        if(P3_3==0 && n<9)
        {
            n++;
            P0=led[n];
            delay(300);
        }
        if(P3_2==0 && n>0)
        {
            n--;
            P0=led[n];
            delay(300);
        }
    }
}
```

## BÀI 5: Đồng hồ hiển thị led 7 đoạn

```
#include <at89x52.h>
unsigned char led[10]={0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f,
0x6f};

void delay(unsigned int
ms)
{
    unsigned int p,k;
    for (p=0;p<ms;p++)
        for (k=0;k<240;k++)
        {}
}

void hienthi(unsigned char hour, unsigned char minute, unsigned char second){
    int n;
        for(n=0; n<60; n++)
                {
                P2=0x00;
                P0= led[second%10];
                delay(1);
                P2=0x04;
                P0= led[second / 10];
                delay(1);
                    P2=0x08;
                    P0= 0x40;
                delay(1);
                    P2=0xC;
                    P0= led[minute % 10];
                delay(1);
                    P2=0x10;
                    P0= led[minute / 10];
                delay(1);
                    P2=0x14;
                    P0= 0x40;
                delay(1);
                    P2=0x18;
                    P0= led[hour % 10];
                delay(1);
                P2=0x1C;
                    P0= led[hour / 10];
                delay(1);
                }
}

void main() {
    int i,j,k;
```

```
   unsigned char hour;
     unsigned char minute;
     unsigned char second;
    i=4;
    j=35;
    k=0;
    while(1)
    {
    k++;
        if(k==60)
        {
            k=0;
            j++;
        }
        if(j==60)
        {
            j=0;
            i++;
        }
        if(i==24)
        {
            i=0;
        }
        hienthi(i,j,k);
    }
}
```

## BÀI 6: RTC

```
#include <at89x52.h>
#ifndef __DS1302__
#define __DS1302__

#define low   0
#define high  1

#define DS1302_SCK P3_6
#define DS1302_IO  P3_4
#define DS1302_RST P3_5

#define DS1302_ADDR_YEAR     0x8C
#define DS1302_ADDR_WEEK     0x8A
#define DS1302_ADDR_MONTH    0x88
#define DS1302_ADDR_DAY      0x86
#define DS1302_ADDR_HOUR     0x84
#define DS1302_ADDR_MINUTE   0x82
#define DS1302_ADDR_SECOND   0x80
unsigned char led[11]={0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 
0x90, 0x7f};
unsigned int count = 0;
unsigned int check = 1, it=1, j=0, k=0, m=0, n=0, q=0;

typedef struct __DS1302_DATE_TIME__
{
  unsigned char year;
  unsigned char week;
  unsigned char month;
  unsigned char day;
  unsigned char hour;
  unsigned char minute;
  unsigned char second;
} ds1302_Datetime;
 ds1302_Datetime myDatetime;

unsigned char to_bcd(unsigned char val)
{
  return ((val / 10) << 4 | (val % 10));
}

unsigned char from_bcd(unsigned char bcd)
{
  return ((bcd&0x70)>>4)*10 + (bcd&0x0F);
}

void ds1302_write(unsigned char dat)
```

```
{
  unsigned char i;
  for(i=0; i<8; i++)
  {
    DS1302_SCK = low;
    DS1302_IO = dat & 0x01;
    dat >>= 1;
    DS1302_SCK = high;
  }
}

void ds1302_write_to_addr(unsigned char addr, unsigned char dat)
{
  DS1302_RST = low;
  DS1302_SCK = low;
  DS1302_RST = high;
  ds1302_write(addr);
  ds1302_write(dat);
  DS1302_RST = low;
  DS1302_IO = high;
  DS1302_SCK = high;
}

unsigned char ds1302_read_from_addr(unsigned char addr)
{
  unsigned char i, value = 0;
  DS1302_RST = low;
  DS1302_SCK = low;
  DS1302_RST = high;
  ds1302_write(addr);
  for(i=0; i<8; i++)
  {
    DS1302_SCK = low;
    value >>= 1;
    if(DS1302_IO)
      value |= 0x80;
    DS1302_SCK = high;
  }
  DS1302_RST = low;
  DS1302_IO = high;
  DS1302_SCK = high;
  return value;
}

void ds1302_set_datetime(ds1302_Datetime *datetime)
{
  ds1302_write_to_addr(0x8E, 0x00);
  ds1302_write_to_addr(DS1302_ADDR_YEAR, to_bcd(datetime->year));
  ds1302_write_to_addr(DS1302_ADDR_WEEK, to_bcd(datetime->week));
  ds1302_write_to_addr(DS1302_ADDR_MONTH, to_bcd(datetime->month));
  ds1302_write_to_addr(DS1302_ADDR_DAY, to_bcd(datetime->day));
  ds1302_write_to_addr(DS1302_ADDR_HOUR, to_bcd(datetime->hour));
  ds1302_write_to_addr(DS1302_ADDR_MINUTE, to_bcd(datetime->minute));
  ds1302_write_to_addr(DS1302_ADDR_SECOND, to_bcd(datetime->second));
  ds1302_write_to_addr(0x8E, 0x10);
}

void ds1302_get_datetime(ds1302_Datetime *datetime)
{
  datetime->year   = from_bcd(ds1302_read_from_addr(DS1302_ADDR_YEAR |
0x01));
  datetime->week   = from_bcd(ds1302_read_from_addr(DS1302_ADDR_WEEK |
0x01));
  datetime->month  = from_bcd(ds1302_read_from_addr(DS1302_ADDR_MONTH |
0x01));
  datetime->day    = from_bcd(ds1302_read_from_addr(DS1302_ADDR_DAY | 0x01));
  datetime->hour   = from_bcd(ds1302_read_from_addr(DS1302_ADDR_HOUR |
0x01));
  datetime->minute = from_bcd(ds1302_read_from_addr(DS1302_ADDR_MINUTE |
0x01));
  datetime->second = from_bcd(ds1302_read_from_addr(DS1302_ADDR_SECOND |
0x01));
}
void delay(unsigned int ms)
{
      unsigned int p,k;
      for (p=0;p<ms;p++)
            for (k=0;k<10;k++)
            {}
}

void hienthi(unsigned char hour, unsigned char minute, unsigned char second){
      int n;
            for(n=0; n<10; n++)
    {
    P2=0x00;
    P0= led[second%10];
    delay(1);
    P2=0x04;
    P0= led[second / 10];
    delay(1);
      P2=0x08;
      P0= 0x3f;
    delay(1);
      P2=0xC;
      P0= led[minute % 10];
    delay(1);
      P2=0x10;
      P0= led[minute / 10];
    delay(1);
```

```
      P2=0x14;
      P0= 0x3f;
    delay(1);
      P2=0x18;
      P0= led[hour % 10];
    delay(1);
    P2=0x1C;
      P0= led[hour / 10];
    delay(1);
    }
}
void timer0_isr(void) interrupt 1
{
    count++; // Tang bi?n d?m khi ng?t Timer 0 x?y ra

    if (count >= 61) // 2 ng?t x?y ra m?i giây (1 giây / 250ms = 4)
    {
        count = 0; // Đ?t l?i bi?n d?m
                  check=1-check;
    }
}

void external_interrupt() interrupt 0 // X? lý ng?t ngo?i INT0
{
    hienthi(myDatetime.hour, myDatetime.minute, myDatetime.second);
            m=1; n=0; q=0;
}
void external_interrupt1() interrupt 2 // X? lý ng?t ngo?i INT0
{
    hienthi(myDatetime.day, myDatetime.month, myDatetime.year);
            n=1; m=0; q=0;
}

void main()
{
   TMOD = 0x01; // Ch? d? 16-bit cho Timer 0
   TH0 = 0 ;      // Giá tr? cho TH0
   TL0 = 0 ;

   ET0 = 1;     // Cho phép ng?t Timer 0
   EA = 1;      // Cho phép ng?t t?ng quát

   TR0 = 1;     // Kh?i d?ng Timer 0
   EA = 1;     // B?t bit EA (Interrupt Enable All)
   EX0 = 1;
   IT0 = 0;
   EX1 = 1;
   IT1 = 0;
      while(1)
      {
            ds1302_get_datetime(&myDatetime);
            if(P3_1==0)
            {q=1;
                  m=0; n=0;
            }
            if(m==1)
            {
                  q=0;
                  ET0 = 0;
                  hienthi(myDatetime.hour, myDatetime.minute, 
myDatetime.second);
            }
                  if(n==1)
            {
                  q=0;
                  ET0 = 0;
                  hienthi(myDatetime.day, myDatetime.month, myDatetime.year);
            }
            if(q==1)
            {
                  ET0 = 1;
                  if(check) hienthi(myDatetime.hour, myDatetime.minute, 
myDatetime.second);
                  else hienthi(myDatetime.day, myDatetime.month, 
myDatetime.year);
            }
      }
}
```

## BÀI 7: Hiển thị led matrix

```
#include<reg51.h>
#include<intrins.h>

sbit SRCLK=P3^6;
sbit RCLK=P3^5;
sbit SER=P3^4;
#define COMMONPORTS           P0
unsigned char code TAB[8]  = {0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};
unsigned char code CHARCODE[18][8]=
{

{0x00,0x00,0x3e,0x41,0x41,0x41,0x3e,0x00},

{0x00,0x00,0x00,0x00,0x21,0x7f,0x01,0x00},

{0x00,0x00,0x27,0x45,0x45,0x45,0x39,0x00},

{0x00,0x00,0x22,0x49,0x49,0x49,0x36,0x00},

{0x00,0x00,0x0c,0x14,0x24,0x7f,0x04,0x00},

{0x00,0x00,0x72,0x51,0x51,0x51,0x4e,0x00},

{0x00,0x00,0x3e,0x49,0x49,0x49,0x26,0x00},

{0x00,0x00,0x40,0x40,0x40,0x4f,0x70,0x00},

{0x00,0x00,0x36,0x49,0x49,0x49,0x36,0x00},

{0x00,0x00,0x32,0x49,0x49,0x49,0x3e,0x00},

{0x00,0x00,0x7F,0x48,0x48,0x30,0x00,0x00},

{0x00,0x00,0x7F,0x48,0x4C,0x73,0x00,0x00},

{0x00,0x00,0x7F,0x49,0x49,0x49,0x00,0x00},

{0x00,0x00,0x3E,0x41,0x41,0x62,0x00,0x00},

{0x00,0x00,0x7F,0x08,0x08,0x7F,0x00,0x00},

{0x00,0x00,0x00,0xFF,0xFF,0x00,0x00,0x00},
{0x00,0x7F,0x10,0x08,0x04,0x7F,0x00,0x00},

{0x7C,0x48,0x48,0xFF,0x48,0x48,0x7C,0x00}

};
void delay(unsigned int time)
{
  unsigned int i,j;
  for(i=0;i<time;i++)
    for(j=0;j<121;j++);
}

void Hc595SendByte(unsigned char dat)
{
      unsigned char a;
      SRCLK=0;
      RCLK=0;
      for(a=0;a<8;a++)
      {
            SER=dat>>7;
            dat<<=1;

            SRCLK=1;
            _nop_();
            _nop_();
            SRCLK=0;
      }

      RCLK=1;
      _nop_();
      _nop_();
      RCLK=0;
}

void main()
{
      unsigned char tab, j;
      unsigned int  i;

      while(1)
      {
            for(i= 0; i<50; i++ )
            {
                  for(tab=0;tab<8;tab++)
                  {

    Hc595SendByte(0x00);
    COMMONPORTS   = TAB[tab];
    Hc595SendByte(CHARCODE[j][tab]);
    delay(2);
```

```
            }

        }
        j++;
        if(j == 18)
        {
            j = 0;
        }
    }
}
```

## BÀI 8: IR REMOTE

```
#include <at89x52.h>

unsigned long int bitMask=0,newKey=0;
unsigned char tVal;
unsigned char ticks=0;
char pulseCnt=0;


sbit LED1 = P2^0;
sbit LED2 = P2^1;
sbit LED3 = P2^2;

#define POWER     0xFFA25D
#define RESET     0xFF629D
#define MODE            0xFFE21D
#define CH_MIN    0xFF22DD
#define CH_PLS    0xFF02FD
#define PREV            0xFFE01F
#define NEXT            0xFFA857
#define EQ              0xFFC23D
#define VOL_M           0xFF6897
#define VOL_P           0xFF9867
#define PLAY            0xFF906F
#define ZERO            0xFFB04F
#define ONE             0xFF30CF
#define TWO             0xFF18E7
#define THREE           0xFF7A85
#define FOUR            0xFF10EF
#define FIVE            0xFF38C7
#define SIX             0xFF5AA5
#define SEVEN           0xFF42BD
#define EIGHT           0xFF4AB5
#define NINE            0xFF52AD


void timer0_ISR() interrupt 1
{
    if(ticks<50){
        ticks++;
            }

    TH0 = 0xFC;
    TL0 = 0x67;
}
```

```
void INT0_ISR()  interrupt 0
{
    tVal = ticks;
    ticks = 0;

    TH0 = 0xFC;
    TL0 = 0x67;

    pulseCnt++;

    if((tVal>=50))
    {
        pulseCnt = -2;
        bitMask = 0;
    }
    else if((pulseCnt>=0) && (pulseCnt<32))
    {

        if(tVal>=2)
        {
            bitMask |=(unsigned long int)1<<(31-pulseCnt);
        }

    }
    else if(pulseCnt>=32)          /*End of Frame*/
    {
        newKey = bitMask;
        pulseCnt = 0;
    }
}


void timerInit (void)
{
        TMOD |= 0x01;  /* Timer 0 Mode 1 */
    TH0 = 0xFC;    /* Reload Value for 1ms for AT89S52*/
    TL0 = 0x67;
    TR0 = 1;       /*Start Timer*/
    ET0 = 1;       /*Enable Timer Interrupt*/

    IT0 = 1;       /*Int0 Falling Edge Interrupt*/
    EX0 = 1;       /*Enable INT0 Interrupt*/

    EA  = 1;       /*Enable Global Interrupts*/
}

void main()
{
      timerInit();


    while(1)
    {
        if(newKey!=0)
        {
                if(newKey == ONE){
                  P2=0x1c;
                  P0=0x06;
            }
            else if(newKey == TWO){
                  P2=0x18;
                  P0=0x5b;
            }
            else if(newKey == THREE)
            {
                  P2=0x14;
                  P0=0x4f;
            }
            else if(newKey == FOUR)
            {
                  P2=0x10;
                  P0=0x66;
            }
            else if(newKey == FIVE)
            {
                  P2=0x0c;
        P0=0x6D;
            }
            else if(newKey == SIX)
            {
                  P2=0x08;
        P0=0x7d;
            }
            else if(newKey == SEVEN)
            {
                  P2=0x04;
        P0=0x07;
            }
            else if(newKey == EIGHT)
            {
                  P2=0x00;
        P0=0x7f;
            }
            newKey = 1;
        }
    }
}
```

## BÀI 9: Calculator LED 7 đoạn

```
#include <at89x52.h>
int led[14]={0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f,
0x39,0x70,0x40,0x40};
int m,s,o,r,key;
int keypad[4][4]={{0,1100,1200,1300},
                  {1,2,3,10},
                  {4,5,6,10},
                  {7,8,9,10},
                  };

void delay(unsigned int
ms)
{
      unsigned int p,k;
      for (p=0;p<ms;p++)
            for (k=0;k<120;k++)
            {}
}

int quetphim()
{
int c, r;
P1=0x0F;
delay(2);
      if(P1!=0x0F)
      {
            for(r=0;r<4;r++)
            {
                  P1=~(0x01<<(4+r));
                  delay(1);
                  for(c=0;c<4;c++)
                  {
                        if((P1&(0x01<<c))==0)
                        {
                              while((P1&(0x01<<c))==0) {};
                                    P1=0x0F;
                              return keypad[r][3-c];
                  }
                  }
            }
      }
P1=0x0F;
      return 1400;
}

void hienthi(int j){
      if(j<100){
            P2=0x00;
```

```
        P0= led[j%10];
        delay(5);
        if(j>9)
        {
        P2=0x04;
        P0= led[j/10];
        delay(5);
        }
    }
    else if(j==1100)
    {
        P2=0x00;
        P0= led[11];
        delay(5);
        P2=0x04;
        P0= led[12];
        delay(5);
    }
    else if(j==1200)
    {
        P2=0x00;
        P0= led[12];
        delay(5);
    }
}

void main(){
    int z=0;
     m=0;s=0; o=0;
    while(1){
        key=quetphim();
            if (key<1400)
            {
                if(key == 1300)
                {
                    if(z==0)
                        s=s+o;
                    else
                        s=s-o;
                    while(P1==0x0F)
                    {
                        hienthi(s);
                    }
                    o=0;
                }
                else if(key<10)
                {
                    o=o*10+key;
                    while(P1==0x0F)
                    {
                        hienthi(o);
                    }
                }
                else if (key==1100)
                {
                    while(P1==0x0F)
                    {
                        hienthi(key);
                    }
                    s=s+o;
                    z=0;
                    o=0;
                }
                else if (key==1200)
                {
                        while(P1==0x0F)
                    {
                        hienthi(key);
                    }
                    s=s+o;
                    z=1;
                    o=0;
                }
            }
    }
}
```

## BÀI 10: Calculator led matrix

```
#include <REG51.h>
#include<intrins.h>
#define COMMONPORTS           P0
sbit SRCLK=P3^6;
sbit RCLK=P3^5;
sbit SER=P3^4;
unsigned char code TAB[8]  = {0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};
unsigned char code CHARCODE[10][4]=
{
{0x00,0x7E,0x42,0x7E},
{0x00,0x00,0x00,0x7E},
{0x00,0x5E,0x52,0x72},
{0x00,0x52,0x52,0x7E},
{0x00,0x70,0x10,0x7E},
{0x00,0x72,0x52,0x5E},
{0x00,0x7E,0x52,0x5E},
{0x00,0x40,0x40,0x7E},
{0x00,0x7E,0x52,0x7E},
{0x00,0x72,0x52,0x7E},
};
unsigned char code CHARCODE1[3][8]=
{
      {0x00,0x00,0x08,0x08,0x3E,0x08,0x08,0x00},
      {0x00,0x00,0x08,0x08,0x08,0x08,0x00,0x00},
      {0x00,0x00,0x14,0x14,0x14,0x14,0x00,0x00},
};
int m,s,o,r,key;
int keypad[4][4]={{0,1100,1200,1300},
                                                      {1,2,3,10},
                                                      {4,5,6,10},
                                                      {7,8,9,10},
                                                      };

void delay(unsigned int
ms)
{
      unsigned int p,k;
      for (p=0;p<ms;p++)
            for (k=0;k<121;k++)
            {}
}

int quetphim()
{
int c, r;
P1=0x0F;
delay(2);
```

```
      if(P1!=0x0F)
      {
            for(r=0;r<4;r++)
            {
                  P1=~(0x01<<(4+r));
                  delay(1);
                  for(c=0;c<4;c++)
                  {
                        if((P1&(0x01<<c))==0)
                        {
                              while((P1&(0x01<<c))==0) {};
                                    P1=0x0F;
                              return keypad[r][3-c];
                  }
                  }
            }
      }
P1=0x0F;
      return 1400;
}

void Hc595SendByte(unsigned char dat)
{
      unsigned char a;
      SRCLK=0;
      RCLK=0;
      for(a=0;a<8;a++)
      {
            SER=dat>>7;
            dat<<=1;

            SRCLK=1;
            _nop_();
            _nop_();
            SRCLK=0;
      }

      RCLK=1;
      _nop_();
      _nop_();
      RCLK=0;
}

void hienthi(int j){
      unsigned char tab;
      unsigned int  i;
      if(j<100)
      {
            for(i= 0; i<10; i++ )
            {
                  for(tab=0;tab<4;tab++)
                  {
                        Hc595SendByte(0x00);

                        COMMONPORTS = TAB[tab];
                        Hc595SendByte(CHARCODE[j/10][tab]);
                        delay(1);
                  }
                  for(tab=0;tab<4;tab++)
                  {

                        Hc595SendByte(0x00);

                        COMMONPORTS = TAB[tab+4];
                        Hc595SendByte(CHARCODE[j%10][tab]);
                        delay(1);
                  }
      }
}
      else if(j>1000)
      {
            j=j/100-11;
            for(i= 0; i<30; i++ )
            {
                  for(tab=0;tab<8;tab++)
                  {

                        Hc595SendByte(0x00);

                        COMMONPORTS = TAB[7-tab];
                        Hc595SendByte(CHARCODE1[j][tab]);
                        delay(1);
                  }
            }
      }
}

void main(){
      int z=0;
       m=0;s=0; o=0;
      while(1){
            key=quetphim();
                  if (key<1400)
                  {
                        if(key == 1300)
                        {
                              if(z==0)
                                    s=s+o;
                              else
                                    s=s-o;
```

```
                    while(P1==0x0F)
                    {
                        hienthi(s);
                    }
                    o=0;
                }
                else if(key<10)
                {
                    o=o*10+key;
                    while(P1==0x0F)
                    {
                        hienthi(o);
                    }
                }
                else if (key==1100)
                {
                    while(P1==0x0F)
                    {
                        hienthi(key);
                    }
                    s=s+o;
                    z=0;
                    o=0;
                }
                else if (key==1200)
                {
                        while(P1==0x0F)
                    {
                        hienthi(key);
                    }
                    s=s+o;
                    z=1;
                    o=0;
                }
            }
    }
}
```

## BÀI 11: Smart home

```
#include <at89x52.h>
#include <XPT2046.c>


unsigned long int bitMask=0,newKey=0;
unsigned char tVal;
unsigned char ticks=0;
char pulseCnt=0;
int key=0, t=0;
float light = 0;
unsigned int ain2 = 0x0000;

sbit LED1 = P2^0;
sbit LED2 = P2^1;
sbit LED3 = P2^2;

#define RL                                            100.0
#define LDR_Constant                                500000.0
#define VDD_in_mV                                   5000.0
#define ADC_count_max                               4095.0

#define POWER     0xFFA25D
#define RESET     0xFF629D
#define MODE            0xFFE21D
#define CH_MIN    0xFF22DD
#define CH_PLS    0xFF02FD
#define PREV            0xFFE01F
#define NEXT            0xFFA857
#define EQ              0xFFC23D
#define VOL_M           0xFF6897
#define VOL_P           0xFF9867
#define PLAY            0xFF906F
#define ZERO            0xFFB04F
#define ONE             0xFF30CF
#define TWO             0xFF18E7
#define THREE           0xFF7A85
#define FOUR            0xFF10EF
#define FIVE            0xFF38C7
#define SIX             0xFF5AA5
#define SEVEN           0xFF42BD
#define EIGHT           0xFF4AB5
#define NINE            0xFF52AD

int keypad[2][4]={{1,2,3,4},
                                                      {5,6,7,8}
                                                      };
```

```
void delay(unsigned int ms)
{
      unsigned int p,k;
      for (p=0;p<ms;p++)
            for (k=0;k<120;k++)
            {}
}

int quetphim()
{
int c, r;
P1=0x0F;
delay(2);
      if(P1!=0x0F)
      {
            for(r=0;r<2;r++)
            {
                  P1=~(0x01<<(4+r));
                  delay(1);
                  for(c=0;c<4;c++)
                  {
                        if((P1&(0x01<<c))==0)
                        {
                              while((P1&(0x01<<c))==0) {};
                                    P1=0x0F;
                              return keypad[r][3-c];
                  }
                  }
            }
      }
P1=0x0F;
      return 0;
}

void timer0_ISR() interrupt 1
{
    if(ticks<50){
        ticks++;
            }

    TH0 = 0xFC;   /*Reload Timer Values for 1ms*/
    TL0 = 0x67;
}


void INT0_ISR()  interrupt 0
{
    tVal = ticks;
    ticks = 0;
    TH0 = 0xFC;  /*Reload Timer Values for 1ms*/
    TL0 = 0x67;

    pulseCnt++;

    if((tVal>=50)) /*Pulse width greater than 50ms signifies Start of Frame*/
    {
        pulseCnt = -2; /*Skip first 2 counts*/
        bitMask = 0;
    }
    else if((pulseCnt>=0) && (pulseCnt<32))
    {
                  /* Accumulate 32 bit data */
        if(tVal>=2)
        {
            bitMask |=(unsigned long int)1<<(31-pulseCnt);
        }

    }
    else if(pulseCnt>=32)          /*End of Frame*/
    {
        newKey = bitMask;
        pulseCnt = 0;
    }
}


void timerInit (void)
{
        TMOD |= 0x01;  /* Timer 0 Mode 1 */
    TH0 = 0xFC;    /* Reload Value for 1ms for AT89S52*/
    TL0 = 0x67;
    TR0 = 1;       /*Start Timer*/
    ET0 = 1;       /*Enable Timer Interrupt*/

    IT0 = 1;       /*Int0 Falling Edge Interrupt*/
    EX0 = 1;       /*Enable INT0 Interrupt*/

    EA  = 1;       /*Enable Global Interrupts*/
}


void main()
{
      timerInit();
    while(1)
    {
                  key = quetphim();
        if(newKey!=0 || key!=0 )
        {
```

```
                                    if(newKey == ONE || key==1){
                                          P2=0x1c;
                                          P0=0x06;
                                    }
                                    else if(newKey == TWO || key==2){
                                          P2=0x18;
                                          P0=0x5b;
                                    }
                                    else if(newKey == THREE || key==3)
                                    {
                                          P2=0x14;
                                          P0=0x4f;
                                    }
                                    else if(newKey == FOUR || key==4)
                                    {
                                          P2=0x10;
                                          P0=0x66;
                                    }
                                    else if(newKey == FIVE || key==5)
                                    {
                                          P2=0x0c;
                                                P0=0x6D;
                                    }
                                    else if(newKey == SIX || key==6)
                                    {
                                          P2=0x08;
                                                P0=0x7d;
                                    }
                                    else if(newKey == SEVEN || key==7)
                                    {
                                          P2=0x04;
                                                P0=0x07;
                                    }
                                    else if(newKey == EIGHT || key==8)
                                    {
                                          P2=0x00;
                                                P0=0x7f;
                                    }
            newKey = 0;
                                    key=0;
                                    t=1;
        }
                        ain2 = XPT2046_read_ADC(ch2, (ADC_res_12 |
single_ended_ADC | power_down_between_conversions_IRQ_enabled));
                        light = ain2;
         if(light<50)
                               {
                                     P2=0x00;
                                     t=0;
                               }
```

```
                                                 else if(t==0)
                                                {
                                                      P2=0xff;
                                                }
                    }
               }
```