DELAY PROGRAM

```
#include "delay.h"
void delayus(unsigned char delay)
{
  while(delay--);
  }
  void delayms(unsigned char delay)
  {
    while(delay--)
    delayus(149);
    }
    Keypad program:
    #include "keypad.h"
    #include "delay.h"
    bit keystatus = FALSE;
    void keypad_init()
    {
      keyport &=0x0F;
      }
      unsigned char getkey ()
      { unsigned char i,j,k,key=0,temp;
      k=1;
      for(i=0;i<4;i++)
      {
        keyport &=~(0x80>>i);
        temp = keyport;
        temp &= 0x07;
        if(7-temp)
        {
          if(!col1)
          {
            key = k+0;
            while(!col1);
            return key; }
            if(!col2)
            {
              key = k+1; 24
              while(!col2);
              return key;
              }
              if(!col3)
              {
                key = k+2;
                while(!col3);
                return key;
                }
                j++;
                }
                k+=3;
                keyport |= 0x80>>i;
                delayms(10);
                }
                return FALSE;
                }
                unsigned char translate(unsigned char keyval)
                {
                  if(keyval<10)
```

```
return keyval+'
else if(keyval==10)
return 'x';
else if(keyval==11)
return '0';
else if(keyval==12)
return 'e';
}
```

LCD program:

```
#include "lcd.h"
#include "delay.h"
#include <REG2051.H>
unsigned char codelockicon[]={0xe, 0xa, 0x1f, 0x1f, 0x1b, 0x1b, 0xe, 0x0};
unsigned char codeunlockicon[]={0xe, 0x2, 0x1f, 0x1f, 0x1b, 0x1b, 0xe, 0x0};
unsigned char codeex[]={0x1f, 0x1b, 0x1b, 0x1b, 0x1b, 0x1f, 0x1b, 0x1f};
unsigned char codeok[]={0x0, 0x1, 0x3, 0x16, 0x1c, 0x8, 0x0, 0x0};
void lcd_reset()
{
  lcd_port= 0xFF; delayms(20);
  lcd_port= 0x03+LCD_EN;


  lcd_port = 0x03;
  delayms(10);
  lcd_port= 0x03+LCD_EN;
  lcd_port= 0x03; delayms(1);
  lcd_port= 0x03+LCD_EN;
  lcd_port= 0x03; delayms(1);
  lcd_port= 0x02+LCD_EN;
  lcd_port= 0x02;
  delayms(1);
  }
  void lcd_init ()
  { unsigned char i;
  lcd_reset();
  lcd_cmd(LCD_SETFUNCTION); // 4-bit mode - 1 line - 5x7 font.
  lcd_cmd(LCD_SETVISIBLE+0x04); // Display no cursor - no blink.
  lcd_cmd(LCD_SETMODE+0x02); // Automatic Increment - No Display shift.
  lcd_cmd(LCD_SETCGADDR);
  for(i=0;i<8;i++)
  lcd_data(lockicon[i]); 27
  for(i=0;i<8;i++)
  lcd_data(unlockicon[i]);
  for(i=0;i<8;i++)
  lcd_data(ex[i]);
  for(i=0;i<8;i++)
  lcd_data(for(i=0;i<8;i++)
  lcd_data(unlockicon[i]);
  for(i=0;i<8;i++)
  lcd_data(ex[i]);
```

```
for(i=0;i<8;i++)
lcd_data(ok[i]);
lcd_cmd(LCD_SETDDADDR); // Address DDRAM with 0 offset 80h.
}
void lcd_cmd (char cmd)
{
  lcd_port= ((cmd >> 4) & 0x0F)|LCD_EN;
  lcd_port= ((cmd >> 4) & 0x0F);
  lcd_port= (cmd & 0x0F)|LCD_EN;
  lcd_port= (cmd & 0x0F);
  delayus(200);
  delayus(200);
  }
  void lcd_data (unsigned char dat)
  {
    lcd_port= (((dat >> 4) & 0x0F)|LCD_EN|LCD_RS);
    lcd_port= (((dat >> 4) & 0x0F)|LCD_RS);
    lcd_port= ((dat & 0x0F)|LCD_EN|LCD_RS);
    lcd_port= ((dat & 0x0F)|LCD_RS); delayus(200);
    delayus(200);
    }
    void lcd_str(unsigned char *str)
    { while(*str){ lcd_data(*str++);
    }
    }
```

Lock program:

```
#include "keypad.h"
#include "lcd.h"
#include "delay.h"
#include "lock.h"
unsigned char codemasterlock[10]="1234567890", defaultulock[5]="54321";
unsigned char userlock[5], input[10];
extern bit newlock;
bit check(unsigned char *first, unsigned char *second, unsigned char len)
{
  unsigned char i=0;
  lcd_str("Enter new code");
  lcd_cmd(0xC0);
  lcd_data(LOCK);
  lcd_data(':');
  status = getinput(5);
  if(status == TRUE)
  {
    lcd_cmd(LCD_CLS);
    lcd_data(OK);
    lcd_str("lockcodesaved!");
    newlock = TRUE;
    store_code();
    delayms(250);
    delayms(250);
    delayms(250);
    delayms(250);
    goto exit;
    }
    else if(status == RETRY)
    goto retry1;
    else if(status == EXIT
    goto exit;
```

```
}
else{ lcd_cmd(LCD_CLS);
lcd_str("WRONG CODE!");
delayms(250);
delayms(250);
delayms(250);
delayms(250);
goto exit;
}
}
else if(status == RETRY)
goto retry;
else if(status == EXIT)
goto exit;
exit:;
}
char getinput(unsigned char max)
{ unsigned char i,key; i=0;
while(1){
  while(!(key=getkey()));
  key = translate(key);
  input[i]=key;
  if(key=='x')
  { if(i==0)
  return EXIT;
  i--;
  lcd_cmd(0xC2+i);
  lcd_data(' ');
  lcd_cmd(0xC2+i);
  }
  else if(key=='e')
  {
    return TRUE;
    }
    else{ i++;
    if(i>max)
    {
      lcd_cmd(LCD_CLS);
      lcd_data(EX);
      lcd_str(" Codetoo Long...");
      delayms(250);
      delayms(250);
      delayms(250);
      delayms(250);
      return RETRY;
      }
      lcd_data('*');
      } } }
      void store_code(){
        unsigned char i;
        for(i=0;i<5;i++)
        userlock[i]=input[i];
        }
```

Main program:

```
#include "lcd.h"
#include "keypad.h"
#include "lock.h"
#include “delay.h"
 extern unsigned char input[10], userlock[5];
extern unsigned char codedefaultulock[5],masterlock[10];
bit newlock=FALSE;
unsigned char retrycount=3;
void main(){
  unsigned char status,i=0;
  bit lockstatus;
  lcd_init();
  keypad_init();
  while(1){
    lcd_cmd(LCD_CLS);
    lcd_str("Enter lock code");
    lcd_cmd(0xC0);
    lcd_data(LOCK);
    lcd_data(':');
    status = getinput(5);
    if(check(input,"12345",5)){
      setulock();
      if(check(input,masterlock,10)){
        retrycount=3;
        lcd_cmd(LCD_CLS);
        lcd_data(EX);
        lcd_str("UNBLOCKED");
        lcd_data(EX);
        delayms(250);
        delayms(250);
        delayms(250);
        delayms(250);
        }
        else{
          lcd_cmd(LCD_CLS);
          lcd_data(EX);
          lcd_str("WRONG CODE");
          lcd_data(EX);
          delayms(250);
          delayms(250);
          delayms(250);
          delayms(goto blocked;
          }
          }
          }
          }
          done:;
          }
```

}

0r
use it only if upper code is not working
Try to make all packages

```
#include<reg51.h>

#include"delay.h"
#include"lcd.h"
#include"keypad.h"
/******************
motor PIN
******************/
sbit a1=P2^0;
sbit a2=P2^1;
sbit a3=P2^2;
sbit a4=P2^3;

void motor();

unsigned char ar[5];

unsigned char com[5]={"77777"};

unsigned int f,m=1,p;

/******************
main program
******************/
void main()
{
	unsigned int i;
```

```
	lcd_ini();

	while(1)
	{
		cmd(0x01);
		lcd_str(" ENTER YOUR");
		cmd(0xc0);
		lcd_str(" PASSWORD");

		for(i=0;i<5;i++)
		{
			keypad1();

			ar[i]=c;

			if(i==0)
			{
			  cmd(0x01);
				lcd_display(' ');
			}
			lcd_display('*');
		}

			//compare();
		if(ar[0]==com[0] && ar[1]==com[1] && ar[2]==com[2] &&
ar[3]==com[3] && ar[4]==com[4])
		{
			m=1;
		}
		else
		{
			m=0;
		}
		if(m==1)
		{
			cmd(0x01);
			lcd_str(" PASSWORD MATCHED");
			cmd(0xc0);
			lcd_str(" ACCESS GRANTED");
```

```
                motor();
                p=0;
                delay_fv(1000,100);
        }
        else if(m==0)
        {
                p++;
                cmd(0x01);
                lcd_str(" WRONG PASSWORD");
                cmd(0xc0);
                lcd_str(" ACCESS DENIED");
                delay_fv(1000,100);
                delay_fv(1000,100);
                m=1;
                if(p==3)
                {
                        cmd(0x01);
                        lcd_str(" PLEASE CONTACT");
                        cmd(0xc0);
                        lcd_str(" YOUR HOD....");
                        while(p==3);
                }
        }
    }
}



void motor()
{
    unsigned int i;

    for(i=0;i<10;i++)
    {
        a2=a3=a4=0;
        a1=1;
        delay_ff();
        a1=a3=a4=0;
        a2=1;
        delay_ff();
```

```
            a1=a2=a4=0;
            a3=1;
            delay_ff();
            a1=a2=a3=0;
            a4=1;
            delay_ff();
  }
      delay_fv(1000,100);
      for(i=0;i<10;i++)
      {
            a2=a3=a1=0;
            a4=1;
            delay_ff();
            a1=a2=a4=0;
            a3=1;
            delay_ff();
            a1=a3=a4=0;
            a2=1;
            delay_ff();
            a4=a2=a3=0;
            a1=1;
            delay_ff();
      }
}
```

lcd.h

```
sbit rs=P3^0;

sbit rw=P3^1;

sbit en=P3^2;

sfr lcd=0x90;


void lcd_display(unsigned int x)  // lcd display fuction
```

```
{
        unsigned int i;
        lcd=x;
        rs=1;
        rw=0;
        en=1;
        for(i=0;i<100;i++);
        en=0;

}

void cmd(unsigned char m)  // lcd command fuction
{
        unsigned int i;
        lcd=m;
        rs=0;
        rw=0;
        en=1;
        for(i=0;i<10;i++);
        en=0;
}

void lcd_ini()   // lcd initilize
```

```
{
        cmd(0x38);
        cmd(0x0e);
        cmd(0x01);
        cmd(0x06);
        cmd(0x90);
}
void lcd_str(unsigned char *str)   // display string on lcd
{
        while(*str!='\0')
        {
          lcd_display(*str);
          str++;
        }
}
```

Keypad.h

```
sbit r1=P0^0; // row 1
sbit r2=P0^1; // row 2
sbit r3=P0^2; // row 3
```

```
sbit r4=P0^3; // row 4
sbit c1=P0^4; // column 1
sbit c2=P0^5; // column 2
sbit c3=P0^6; // column 3
sbit c4=P0^7; // column 4

unsigned int c;

char keypad1()
{
	P2=0xff;
	while(1)
	{
		r1=0;
		r4=1;
		if(c1==0)
		{
			c='7';
			delay_pf(500);
			return c;
		}
		else if(c2==0)
		{
```

```
        c='8';
        delay_pf(500);
        return c;
}
else if(c3==0)
{
        c='9';
        delay_pf(500);
        return c;
}
else if(c4==0)
{
        c='/';
        delay_pf(500);
        return c;
}

r1=1;
r2=0;
if(c1==0)
{
        c='4';
        delay_pf(500);
```

```
        return c;
}
else if(c2==0)
{

        c='5';
        delay_pf(500);
        return c;
}
else if(c3==0)
{
        c='6';
        delay_pf(500);
        return c;
}
else if(c4==0)
{
        c='*';
        delay_pf(500);
        return c;
}

r2=1;
```

```
r3=0;
if(c1==0)
{
        c='1';
        delay_pf(500);
        return c;
}
else if(c2==0)
{

        c='2';
        delay_pf(500);
        return c;
}
else if(c3==0)
{
        c='3';
        delay_pf(500);
        return c;
}
else if(c4==0)
{
        c='-';
```

```
        delay_pf(500);
        return c;
}

r3=1;
r4=0;
if(c1==0)
{
        delay_pf(500);
        cmd(0x01);
}
else if(c2==0)
{

        c='0';
        delay_pf(500);
        return c;
}
else if(c3==0)
{
        c='=';
        delay_pf(500);
        return c;
```

```
            }
            else if(c4==0)
            {
                c='+';
                delay_pf(500);
                return c;
            }
        }
}
```

Delay.h

```
void delay_ff()  // fully fixed delay
{
    unsigned int i,j;

    for(i=0;i<80;i++)
    for(j=0;j<120;j++);
}

void delay_pf(unsigned int x)  // partial variable delay
{
        unsigned int i,j;
```

```
        for(i=0;i<x;i++)
        for(j=0;j<50;j++);
}

void delay_fv(unsigned int x,y)  // fully variable delay
{
        unsigned int i,j;

        for(i=0;i<x;i++)
        for(j=0;j<y;j++);
}
```