

Microcontrollers

1) Drive a stepper motor interface to rotate the motor in Anti-clockwise by N-steps. Introduce suitable delay between successive steps

```
#include <stdio.h>
```

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

```
char xdata port_at_0xe803;
```

```
char xdata port_at_0xe800;
```

```
char idata acc_at_0x30;
```

```
delay();
```

```
{
```

```
int j;
```

```
for (j=0; j<800; j++)
```

```
{
```

```
{
```

```
void main() {
```

```
port = 0x80;
```

```
while (1)
```

```
{
```

```
acc = 0x11;
```

```
port = acc
```

```
delay();
```

```
acc = 0x22;
```

```
port = acc;
```

```
delay();
```

```
acc = 0x44;
```

```
port = acc;
```

```
delay();
```

```
acc = 0x88;
```

```

    portA = 0x0C;
    delay(1);
}
}

```

2) Clockwise rotation :-

```

#include <reg51.h>
#include <stdio.h>
void main()
{

```

```

    void delay();
    while(1)
    {

```

```

        P0 = 0x06;

```

```

        delay(1);

```

```

        P0 = 0x0C;

```

```

        delay(1);

```

```

        P0 = 0x09;

```

```

        delay(1);

```

```

        P0 = 0x03;

```

```

        delay(1);
    }

```

```

void delay()
{

```

```

    unsigned char cnt, cnt1;

```

```

    for (cnt = 0; cnt <= 254; cnt++)

```

```

    for (cnt1 = 0; cnt1 <= 254; cnt1++)

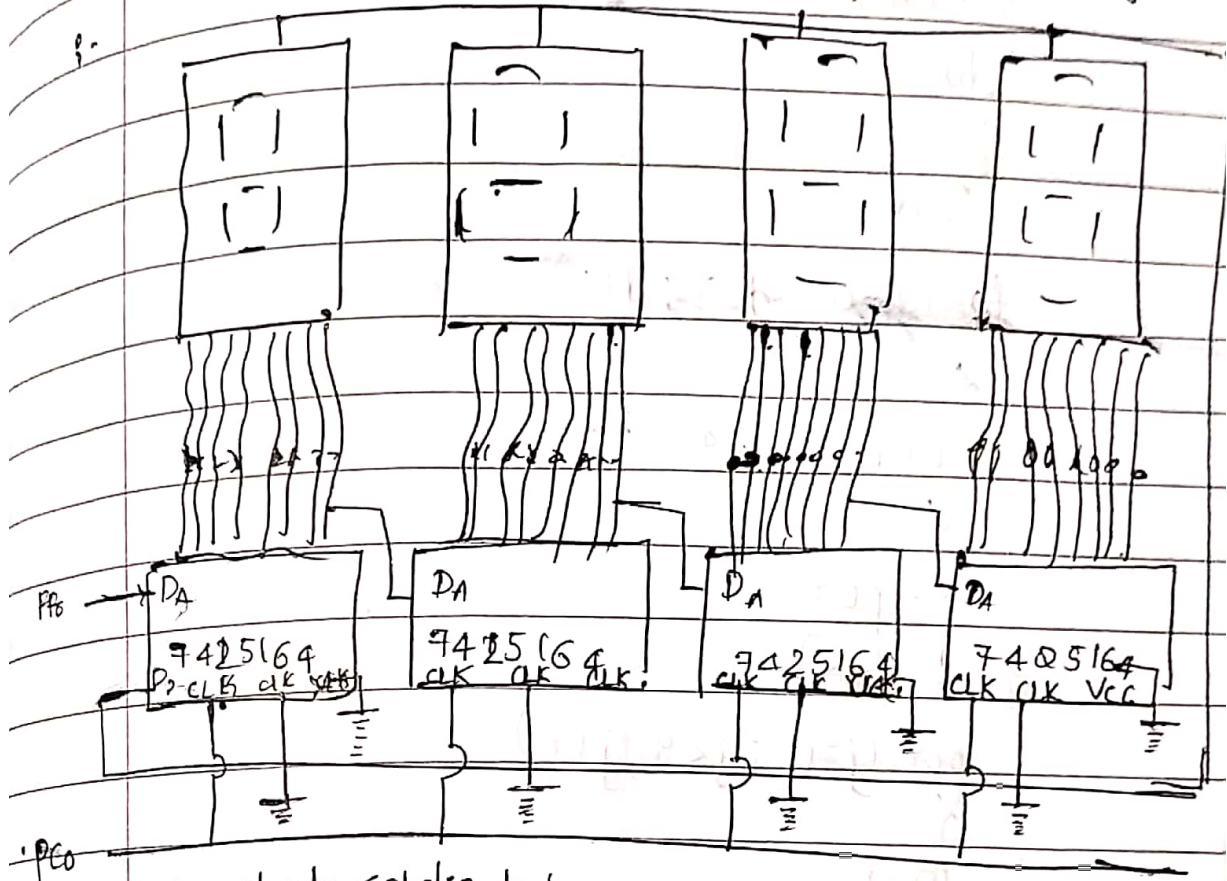
```

```

    }
}

```

3) Display message FIRE and HELP alternatively with flickering effects on a 7-segment display interface period of time. Ensure a flashing rate that makes it easy to read both the messages



```
#include <stdio.h>
#include <reg51.h>
char xdata comm1 _at_ 0xe803;
char xdata portB _at_ 0xe801;
char xdata portC _at_ 0xe802;
char port[20] = {0x8e, 0xf9, 0xde, 0x86, 0xf4, 0xf4, 0xf4,
                 0x89, 0x86, 0xc7, 0x8c, 0, 0, 0, 0, 0, 0, 0, 0, 0};

delay()
{
    long u;
    for (u = 0; u < 800; u++)
    {
    }
}
```



```
void main()
```

```
{
```

```
int d, b, j, m;
```

```
unsigned char k;
```

```
commhl = 0x86;
```

```
do
```

```
{
```

```
    i = 0;
```

```
    for (d = 0; d < 3; d++)
```

```
    {
```

```
        for (b = 0; b < 4; b++)
```

```
        {
```

```
            k = port
```

```
                + [i++];
```

```
            for (j = 0; j < 8; j++)
```

```
            {
```

```
                m = k
```

```
                k = k & 0x80;
```

```
            }
```

```
            if (k == 00)
```

```
                portB = 0x00;
```

```
            else
```

```
                portB = 0x01;
```

```
            }
```

```
            portC = 0x01;
```

```
            portC = 0x00;
```

```
            k = m;
```

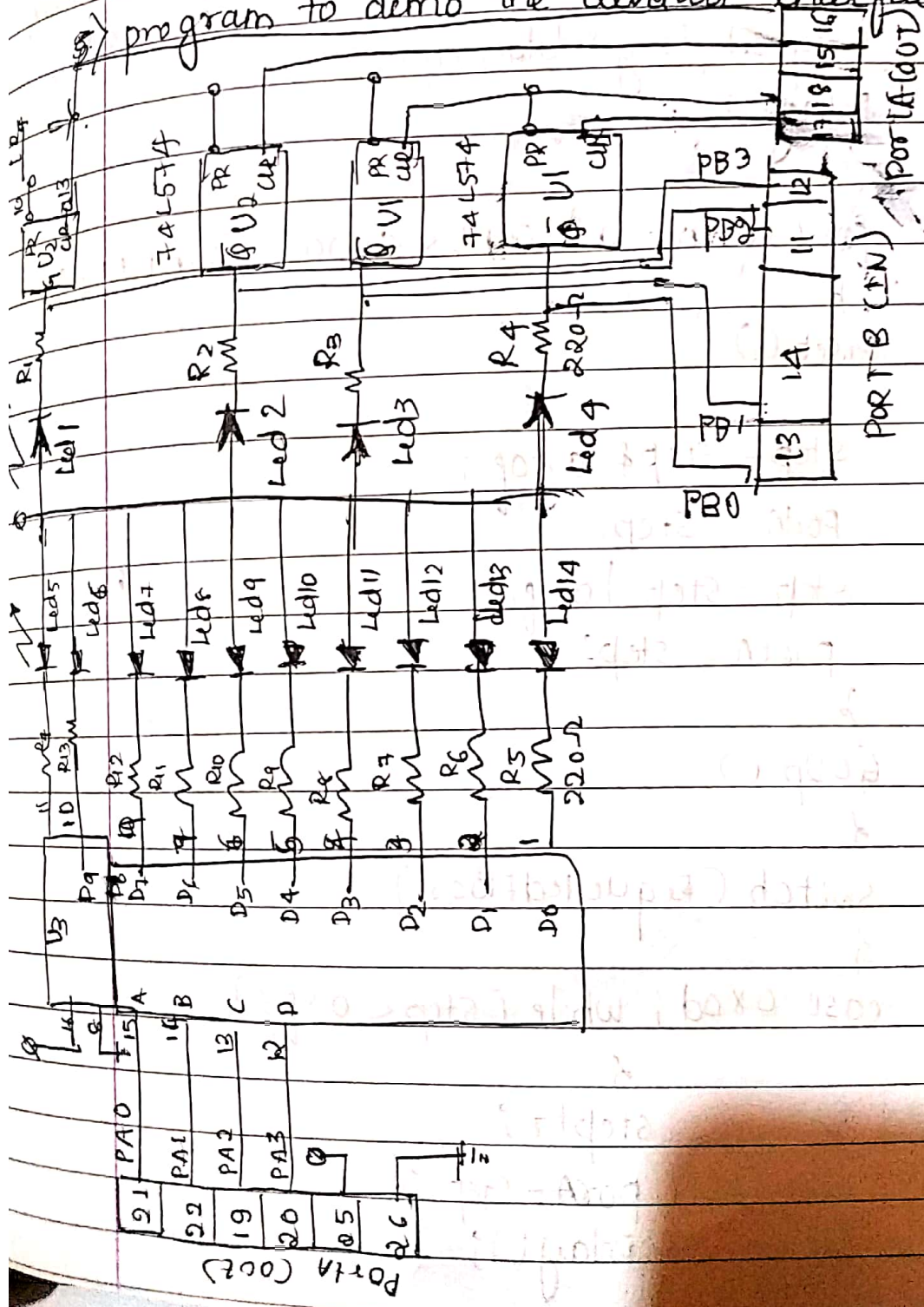
```
            K <<= 1;
```

```
}
```

delay (2)

```
while (1);
```

§) program to demo the elevator interface




```
#include <stdio.h>
#include <reg51.h>
unsigned char xdata CommandWord at 0xe800;
portA
unsigned char xdata CommandWord at 0xe801;
unsigned char xdata portB at 0xe801;
unsigned char xdata presentFloor, RequestedFloor;
Step = 0xf0;

unsigned long xdata Count, i;

Delay()
{
    for (count = 0; count <= 4500; count++);
    Reset();
}

Step = Step & 0xf0;
PortA = Step;
Step = Step | 0xf0;
portA = Step;

Group()
{
    switch (RequestedFloor)
    {
        case 0x0d; while (Step < 0xf3)
        {
            Step++;
            portA = Step;
            Delay();
        }
    }
}
```

Reset ();

break;

case 0x0b: while (step < 0xfg)

{

Step ++;

port A = step;

Delay ();

}

Reset ();

Break ;

case 0x07: while (step < 0xfg)

{

Step ++;

port A = step;

Delay

}

Reset ();

break;

}

}

GoDown ()

{

Switch (RequestedFloor)

{

case 0x0d: while (step > 0x3)

{

Step ++;

port A = step;

Delay ();

}


```
Reset();  
break;  
case 0x0b: while (step > 0x06)
```

```
{  
    step--;  
    portA = step;  
    delay(1);  
}
```

```
Reset();
```

```
break;
```

```
case 0x0e: while (step > 0x00)
```

```
{  
    step++;  
    portA = step;  
    delay(1);  
}
```

```
Reset();
```

```
break;
```

```
}
```

```
}
```

```
void main()
```

```
{
```

```
    Commandword = 0x82;
```

```
    portA = 0x00;
```

```
    presentFloor = 0x0e;
```

```
    while (1)
```

```
    {
```

```
        RequestedFloor = portB;
```


RequestedFloor = RequestedFloor & 0x0f;
 if (RequestedFloor != 0x0f & RequestedFloor != presentFloor)

{
 if (RequestedFloor < presentFloor)
 GoUp();

else

GoDown();

presentFloor = RequestedFloor;

}
 RequestedFloor = Port B;
 }