

14/12/20 Microcontroller

Program - 1

Q Write a program to rotate stepper motor in clock wise direction

```
→ #include <stdio.h>
#include <conio.h>
#include <reg51.h>
char xdata port_a = 0xe803;
char xdata port_a = 0xe800;
char xdata acc = 0xe30;
void delay
```

```
{
    for (int i=0; i<800; i++)
    {
        {
```

```
void main ()
```

```
{
    port = 0x800;
    while (1)
    {
```

```
        acc = 0x88;
        port_a = acc;
        delay ();
```

```
        acc = 0x44;
        port_a = acc;
        delay ();
```

```
acc = 0x44;
portA = acc;
delay ( );
```

```
acc = 0x11;
port = acc;
delay ( );
}
```

- Q. Derive a stepper motor Interface to rotate the motor in Anti-clockwise direction by N steps.

```
#include <stdio.h>
#include <reg51.h>
char xdata a port - at - 0x803;
char xdata port - at - 0xe800;
char i data acc - at - 0x30;
delay ( )
{
    int j;
    for (j = 0; j < 800; j++)
    {
        }
    }
}
```

```
void main ( )
{
```

```
port = 0x80
```

Configure all ports
in 8288255
as output port

```
while (1)
```

```
{
```

```
    acc = 0x11;
```

```
    port a = acc;
```

```
    delay (1);
```

```
    acc = 0x22;
```

```
    port a = acc;
```

```
    delay (1);
```

```
    acc = 0x44;
```

```
    port a = acc;
```

```
    delay (1);
```

```
    acc = 0x88;
```

```
    port a = acc;
```

```
    delay (1);
```

```
}
```

```
}
```

Program - 3 Microcontroller

To display Bangalore in rolling fashion

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

```
#include <stdlib.h>
```

```
char x data - at 0xe803
```

```
char x data port B - at - 0xe801;
```

```
char x data port C - at - 0xe802;
```

```
void delay ()
```

```
{
```

```
    for (int i=0; i<4000; i++)
```

```
    {
```

```
    }
```

```
}
```

```
void main ()
```

```
{
```

```
    int i=0; m
```

```
    unsigned char k, m;
```

```
    coordMN = 0x80
```

```
    for (int d=0; d<1; d++)
```

```
    {
```

```
        i=0;
```

```
        for (int b=13; b>0; b--)
```

```
        {
```

```
            delay ();
```

```
            k = port [i++];
```

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

```
            {
```

```
                m = k
```

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

```
if (k == 0)
```

```
    port B = 0x00;
```

```
else
```

```
    port B = 0x01;
```

```
    port C = 0x01
```

```
    port C = 0x00;
```

```
    k = m
```

```
    k++;
```

```
}
```

```
    delay (1)
```

```
}
```

```
& } write (1)
```

```
}
```


Program - 4 Microcontroller Fire! & Help

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

```
#include <reg 51.h>
```

```
char xdata CommN_at = 0xe803;
```

```
char xdata portB_at = 0xe801;
```

```
char xdata portC_at = 0xe302;
```

```
char port[20] = { 0x8e, 0xf9, 0xde, 0x86,  
0xff, 0xff, 0xff, 0xff, 0x89, 0x86, 0xc7, 0x86,  
};
```

```
void delay()
```

```
{  
for (i=0; i<8000; i++)  
{  
}
```

```
{  
void main()
```

```
{  
int d, b, v; m;
```

```
unsigned char k;
```

```
CommN = 0x80;
```

```
de  
}
```

```
i = 0;
```

llusing band c as
input.

Program 5 Microcontroller

Program to demo the elevator interface

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

```
#include <reg 51.h>
```

```
unsigned char xdata CommandWord - at - 0xe802
```

```
unsigned char xdata PortA - at - 0xe800;
```

```
unsigned char xdata portB - at - 0xe801;
```

```
unsigned char xdata present floor, requested floor;
```

```
slcr = 0xf0;
```

```
unsigned long xdata Count;;
```

```
delay ()
```

```
{
```

```
for (count = 0; count <= 4500; count++)
```

```
{
```

```
}
```

```
}
```

```
Reset ()
```

```
{
```

```
step = step & 0xf
```

```
portA = step
```

```
step = step | 0xf0
```

```
portA = step
```

```
}
```

```
Group 6)
```

```
{
```

```
See.
```

Go down ()

{

switch (Requested Floor)

{

case 0x0d: while (step > 0xf3)

{

Step --;

port A = Step;

Delay ();

}

Reset ();

break;

Case 0x0b: while (step > 0xf6)

{

Step --;

port A = step;

Delay ();

}

Reset ();

break;

case 0x0e: while (step > 0xf0)

{

Step --

port A = step;

Delay ();

}

Reset ();

break;

}

return 0;

}

void main ()

{

Command word = 0x82;

Port A = 0xf0;

Present Floor = 0x0e;

while (1) {

Request Floor = Port B;

Requested Floor = Requested Floor + 0x0f;

else

Go down ();

Present floor = Requested floor ;

}
}

Requested Floor = Port B;

}

}