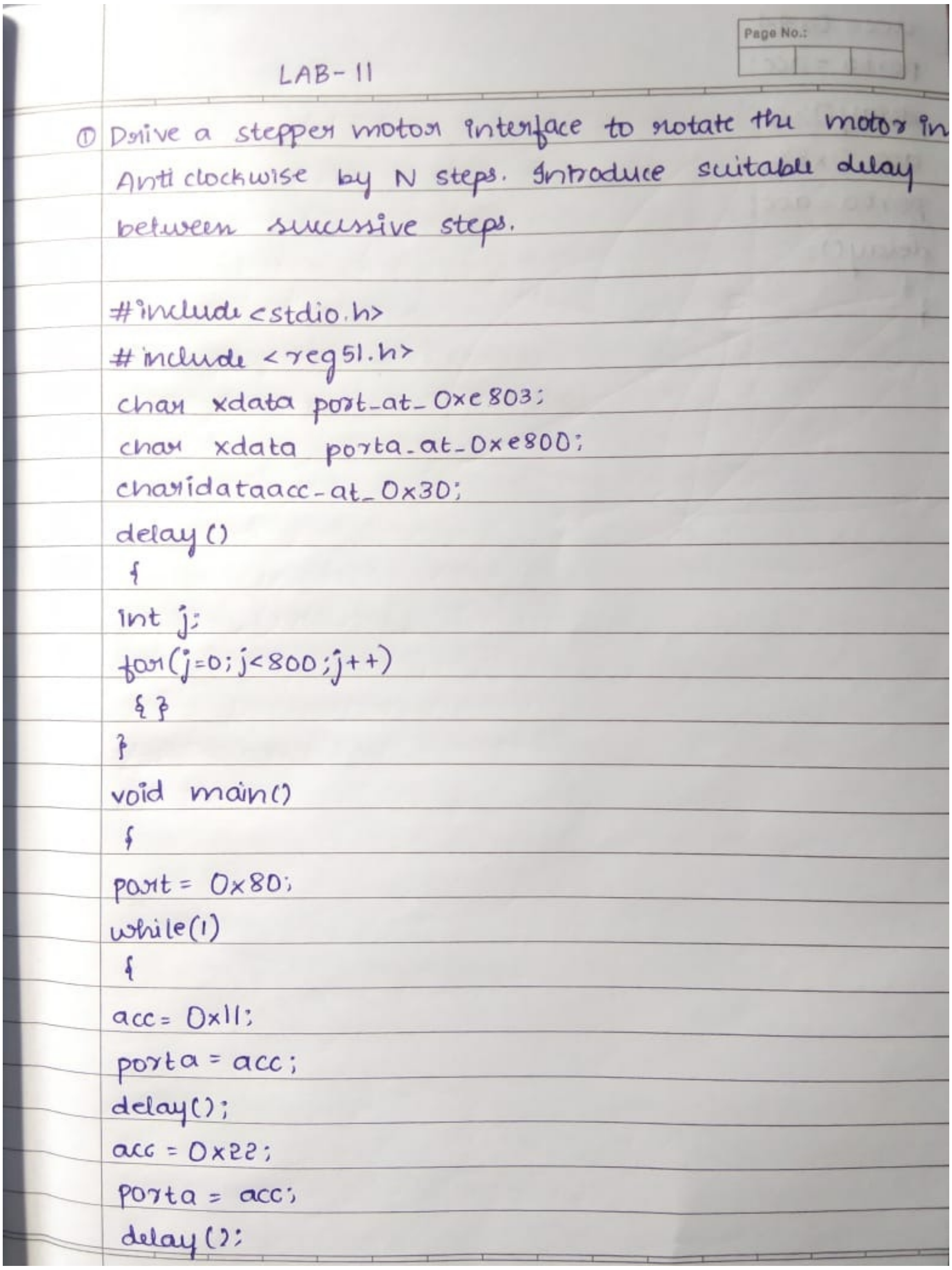
INTERFACING PROGRAMS USING 8051:

11. Drive a steper motor interface to rotate the motor in Anti clockwise by N steps .Introduce suitable delay between successive steps .

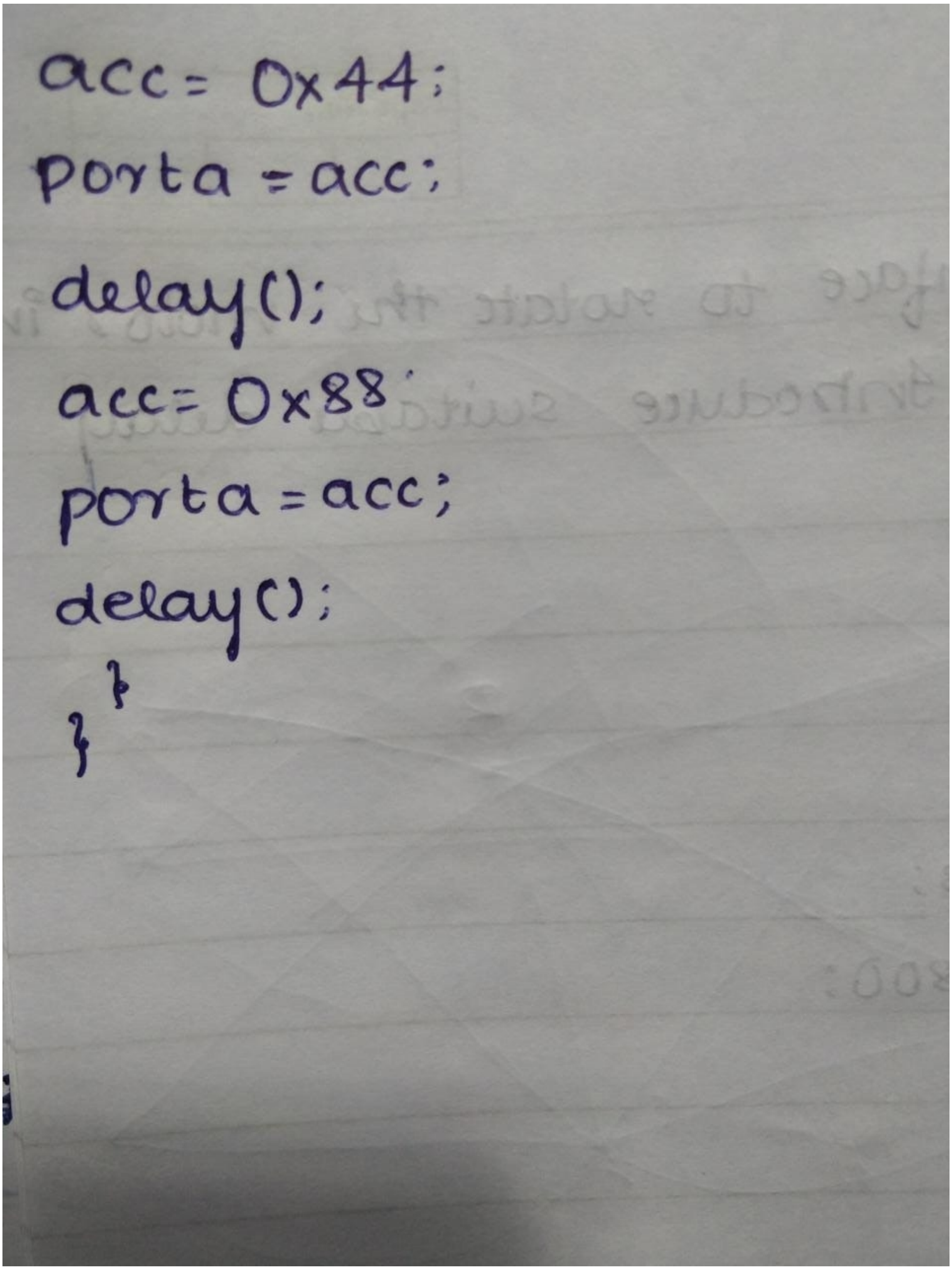


Page No.:

LAB- 11

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 porta_at_0xe800;
charidataacc-at_0x30;
delay ()
 {
int j;
for(j=0;j<800;j++)
 {}
}
void main()
 {
port = 0x80;
while(1)
 {
acc= 0x11;
porta = acc;
delay();
acc = 0x22;
porta = acc;
delay();
```



```
acc= 0x44;
porta =acc;

delay();
acc= 0x88;
porta =acc;
delay();
 }
}
```

12. Drive a Stepper Motor interface to rotate the motor in clockwise by N steps.

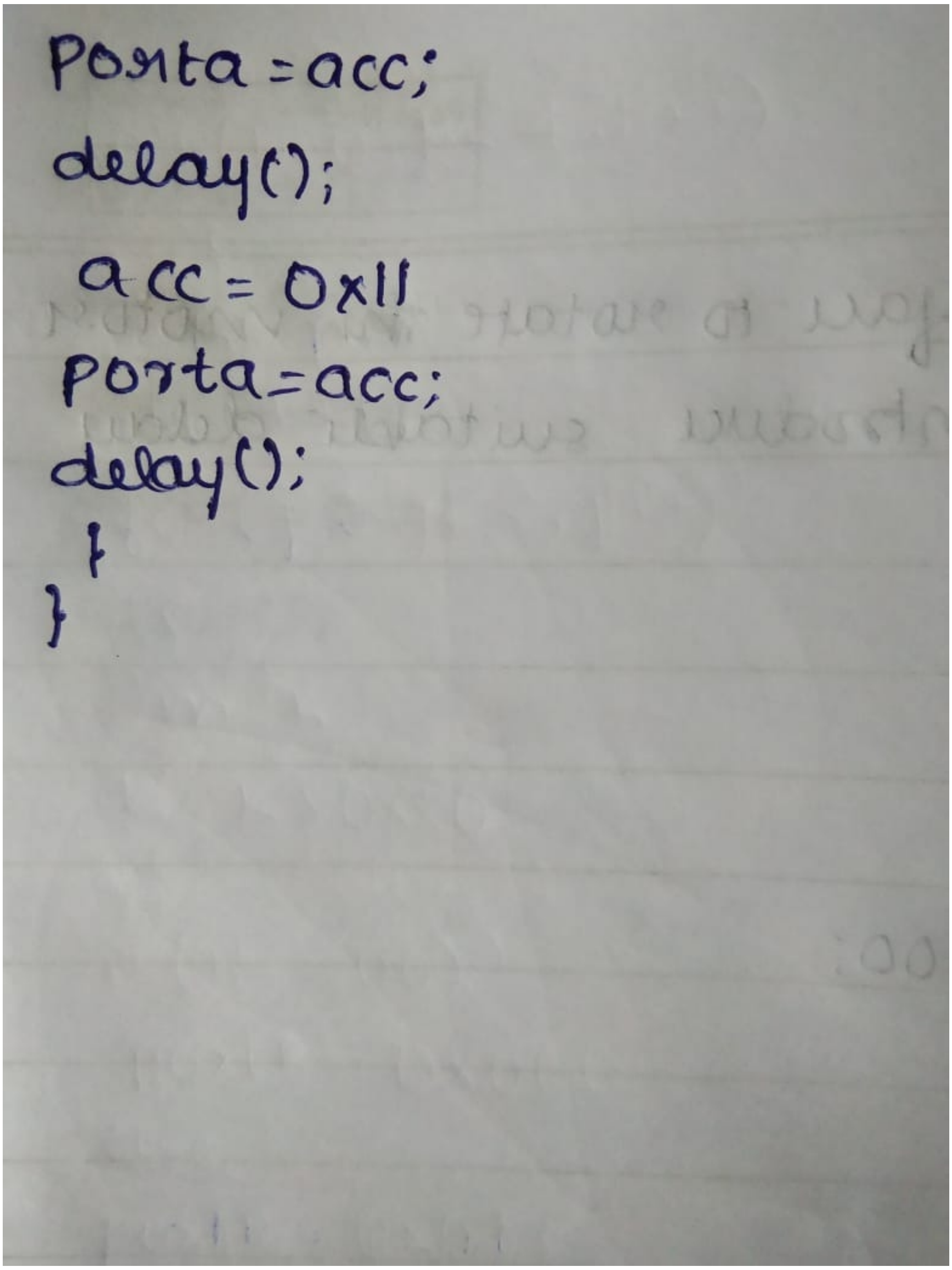
Introduce suitable delay between successive steps

Page No.:

LAB-12.

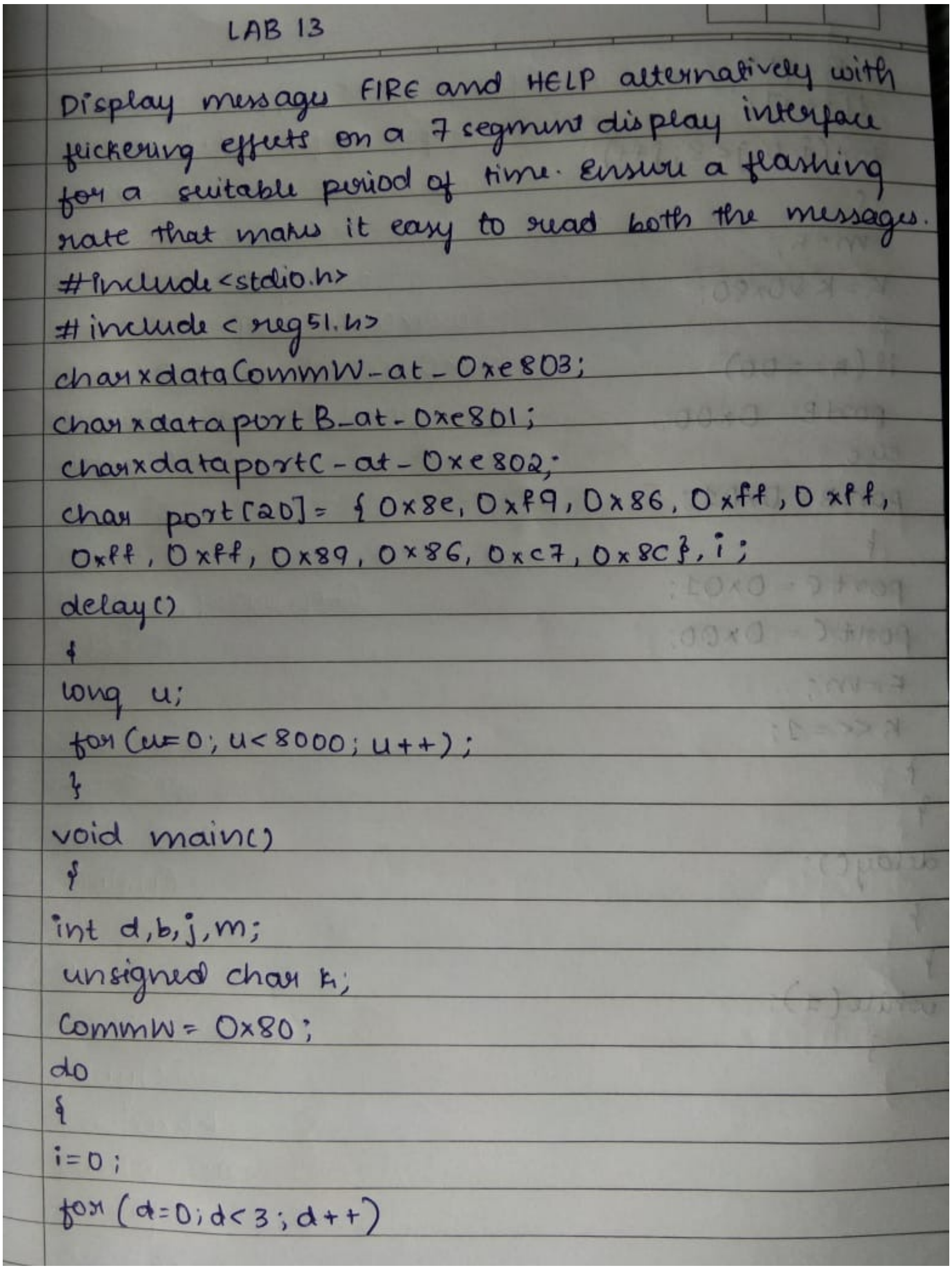
Drive a stepper Motor interface to rotate the mot in clockwise by N steps. Introduce suitable del between successive steps.

```
#include <stdio.h>
#include <reg51.h>
charxdata port-at-0xe803;
charxdata porta-at-0xe800;
charidataacc-at-0x30;
delay()
{
int j;
for (j=0;j<800;j++)
  {}
}
void main()
{
port = 0x80;
while(1)
{
acc = 0x88;
porta = acc;
delay();
acc = 0x44;
porta = acc;
delay();
acc = 0x22;
```



```
Porta =acc;
delay();
 acc = 0x11
 porta=acc;
delay();
  }
}
```

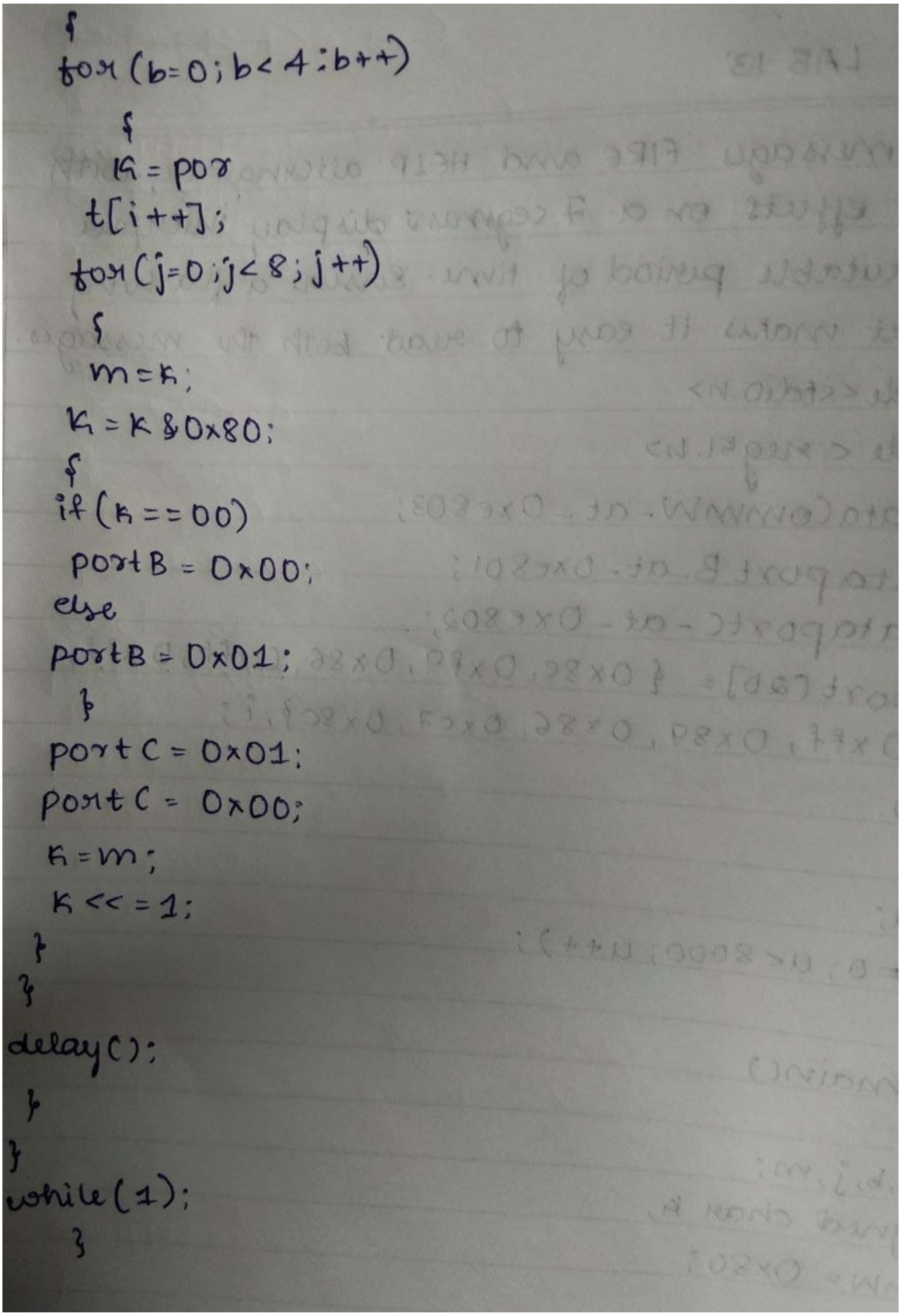
13. Display messages FIRE and HELP alternately with flickering effects on a 7-segment display interface for a suitable period of time. Ensure a flashing rate that makes it easy to read both the messages



LAB 13

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

```
#include<stdio.h>
#include <reg51.h>
charxdataCommW-at-0xe803;
charxdataportB-at-0xe801;
charxdataportC-at-0xe802;
char port[20]= {0x8e,0xf9,0x86,0xff,0xff,
0xff,0xff,0x89,0x86,0xc7,0x8c},i;
delay()
{
long u;
for(u=0;u<8000;u++);
}
void main()
{
int d,b,j,m;
unsigned char k;
CommW= 0x80;
do
{
i=0;
for(d=0;d<3;d++)
```



```
 {
 for(b=0;b<4:b++)
   {
    k = por
   t[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();
 }
}
while(1);
  }
```

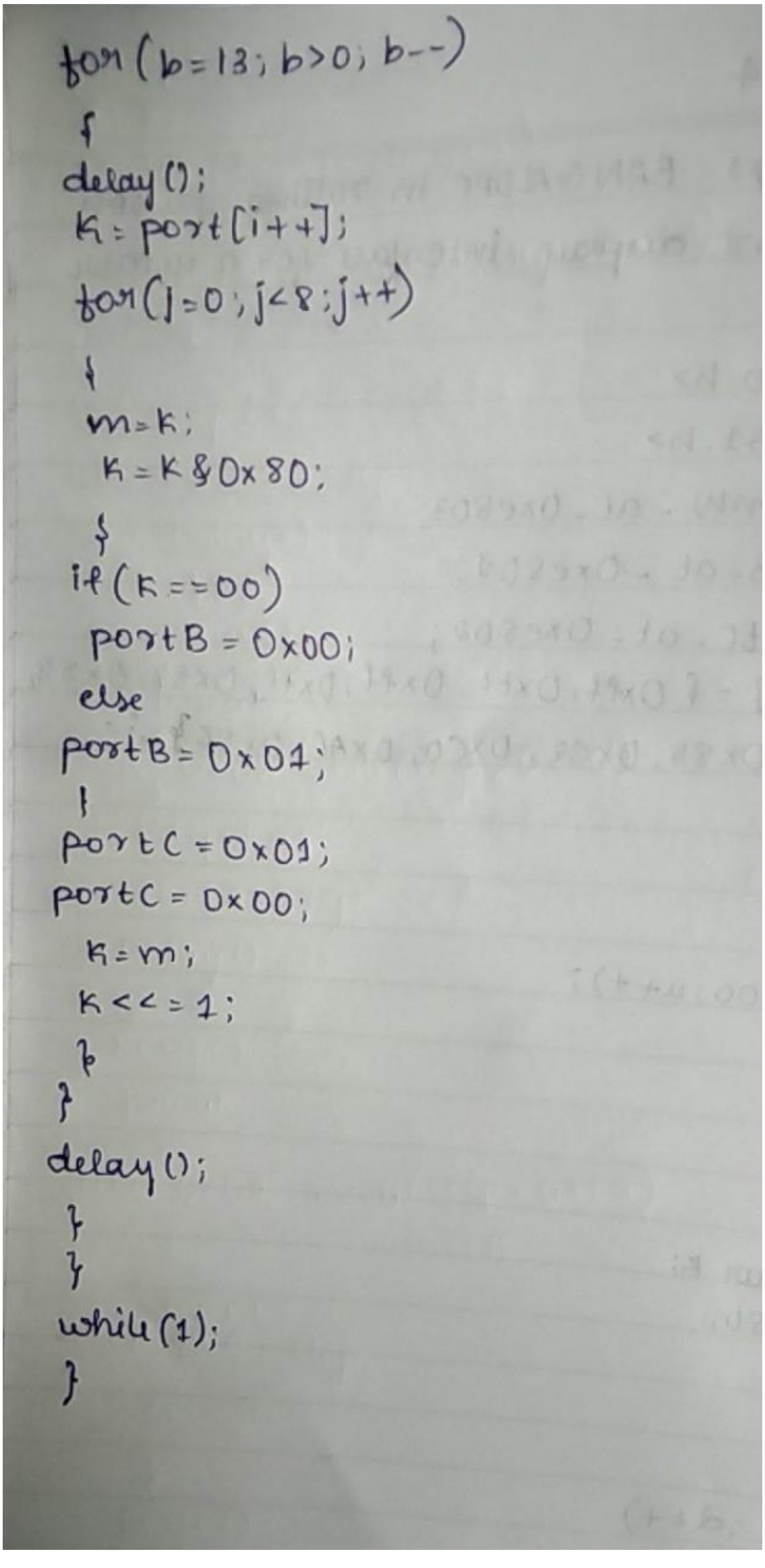
14. Display messages BANGALORE in rolling fasion on a 7-segment display interface

for a suitable period of time.

LAB 14

Display messages BANGALORE in rolling fashion on a 7-segment display interface for a suitable period of time.

```
#include <stdio.h>
#include <reg51.h>
char xdata CommW _at_ 0xe803;
char xdata portB _at_ 0xe801;
char xdata portC _at_ 0xe802;
char port[20] = { 0xff, 0xff, 0xff, 0xff, 0x83, 0x88,
0xC8, 0x82, 0x88, 0xC7, 0xC0, 0xAF, 0x86 }, i;
delay()
{
long u;
for(u=0;u<4000;u++);
}
void main()
{
int d,b,j,m;
unsigned char k;
CommW = 0x80;
do
{
i=0;
for(d=0;d<1;d++)
{
```



```
for(b=13;b>0;b--)
 {
delay();
 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();
 }
 }
while(1);
}
```

15. Program to demo the elevator interface

Lab 15

Program to demo the elevator interface.

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

unsigned char xdataCommandWord-at_0xe803;
unsigned char xdataPortA-at-0xe800;
unsigned char xdata PortB-at-0xe801;
unsigned char xdataPresentFloor, RequestedFloor,
Step = 0xf0;
unsigned long xdataCount,i;

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

Reset()
{
Step= Step & 0x0f;
PortA= Step;
Step = Step | 0xf0;
Port A = Step;
}
```

```
GoUp()
{
switch(RequestedFloor)
 {
 case 0x0d : while(step <0xf3)
    {
    step++;
    PortA = Step;
    Delay();
    }
    Reset();
    break;
case 0x0b: while (step < 0xf6)
   {
   step++;
   PortA = step;
   Delay();
   }
  Reset();
  break;
case 0x07: while (step<0xf9)
   {
   step++;
   PortA = step;
   Delay();
   }
  Reset();
  break;
}
}
```

```
GoDown()
{
switch(RequestedFloor)
 {
case 0x0d : while (step > 0xf3)
             {
              step--;
             PortA = step;
             Delay();
            }
  Reset();
  break;
case 0x0b: while(step>0xf6)
   {
    step--;
    PortA=step;
    Delay();
    }
    Reset();
     break;
case 0x0e: while (step>0xf0)
    {
     Step --;
     PortA = step;
     Delay();
     }
    Reset();
    break;
  }
}
void main()
 {
CommandWord = 0x82;
 PortA = 0xf0;
 PresentFloor = 0x0e;
 while(1){
  RequestedFloor = PortB;
  RequestedFloor = RequestedFloor & 0x0f;
  if(RequestedFloor != 0x0f && RequestedFloor != PresentFlo
   if(RequestedFloor < PresentFloor)
        GoUp();
   else
       GoDown();
    PresentFloor = RequestedFloor;
    }
  RequestedFloor = PortB;
    }
```