

- i) Design a Stepper motor interface to rotate the motor in Anticlockwise by N steps. Introduce suitable delay between successive steps

#include <8051.h> // Stepper motor code in C language (is)

#include <delay.h> // gives better results than sleep

#include <reg51.h> // gives better results than delay

char xdata port_at_0xe803;

char xdata porta_at_0xe800; // port_a = 800H

char datacc_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();

main() { acc = 0x88; } //initialization

 porta = acc;

 delay();

}

3. Now in start or stop, still return address to main.

result push address - subtract - result in port selection

4. result stored

- 2) Devise a stepper motor interface to rotate the motor in clockwise and anticlockwise by N steps. Introduce suitable delay between successive steps.

#include <stdio.h> //for delay function

#include <uart.h> //for serial communication

char xdata porta_at_0xe803;

char xdata porta_at_0xe800;

char data acc_at_0x30;

delay();

{ int j;

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

 {

}

void main()

{

 port = 0x80;

 while(1)

{

 acc = 0x88;

 porta = acc;

 delay();

acc = 0x44)

(029x0 - 20 - 61 to update acc)

porta = acc;

(029x0 - 20 - 3 to update porta)

delay();

if (8>0) acc = 0x22; 22x0 (1100 1000)

porta = acc;

delay();

acc = 0x11)

porta = acc;

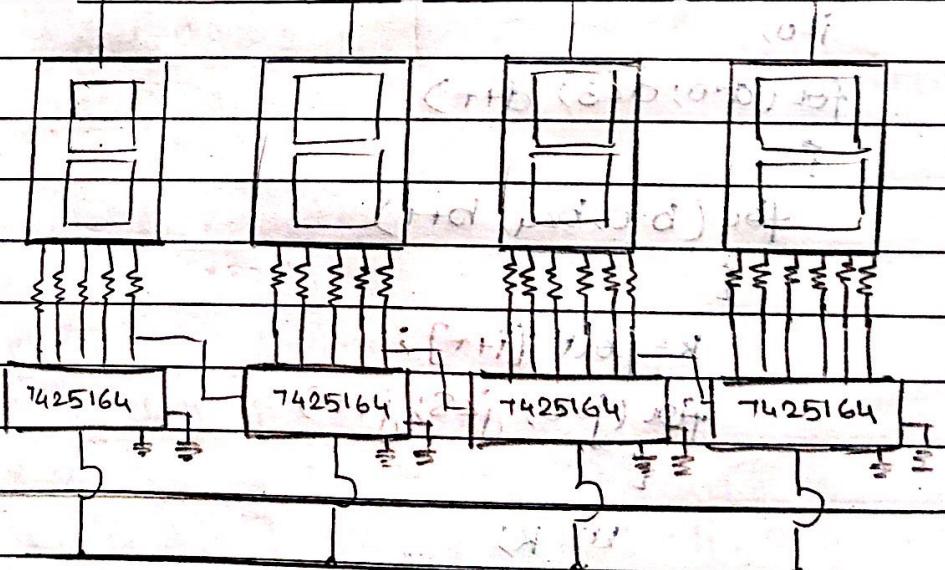
delay();

(+HLL0008 IN LO=H) 10)

3

3

- 3) Display messages FIRE and HELP alternately with flicking 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>

char xdata CommW = 0xC803;

char * dataportB - at - 0xe801; : portB = 0x0
char * dataportC - at - 0xe802; : portC = 0x0
char port [20] = { 0x8e, 0xf9, 0x0d, 0x86, 0xff,
 0xff, 0xff, 0x89, 0x89, 0xc7, 0x8c }, i;

delay()

{

 long u;
 for (u=0; u<8000; u++)

}

void main()

{

 init(); wait d,b,j,m; get free disk address from phys. disk
 unsigned char k; portA = dataportB + 0x8000
 portB = 0x801; portC = dataportC + 0x8000
 commW = 0x801; portB = 0x0 and portC = 0x0
 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 80 x 80;

{

 if (k==000)

 portB = 0x00;

else

portB = 0x01;

}

portC = 0x01;

portC = 0x00;

K=M.

R<=1>;

3

3

delay(2);

3

3

while(1);

3

- 4) Display message, BANGALORE in sailing fashion on a 7 segment display interface for a suitable period of time

#include <stdio.h>

#include <regst.h>

char *data commW at 0x803;

char *data portB - at 0x801;

char *data portE - at 0x802;

char port[20] = {0xff, 0xff, 0xff, 0xff, 0x83, 0x88, 0x08, 0x82, 0x88, 0xc1, 0xc0, 0xAF,

0x86, 3, i};

delay()

{

long u;

for (u=0; u<4000; u++)

}

void main()

{

int d, b, j, m;

unsigned char k;

comm0 = 0x80;

do

{

i=0;

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

{

for (b=15; b>0; b--)

{

Das nächste Problem delay() auf einer anderen Maschine (A)

mit (p, b), Datiere K = portB[i++]; portB zwisch.

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

{

m=k;

K = K & 0x80;

{

if (K==0)

portB = 0x00;

else portB = 0xFF;

else portB = 0x01;

}

PORT C = 0x01;

PORTC = 0x00;

K=m;

K<=1;

3

3

delay();

3

3

while(1);

3

(initialiser) main

5) Program to demo elevator interface

#include <stdio.h>

#include <sys/types.h>

unsigned char *data CommandWord - at - 0xe803;

unsigned char *data FOLTA - at - 0xe800;

unsigned char *data PORTB - at - 0xe801;

unsigned char idata PresentFloor, RequestedFloor,

&tpc = 0xf0;

unsigned long data count; i;

delay()

{

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

}

Reset()

{

Step = Step2 0x0f,

PortA = Step;

Step = Step1 0xf0,

PortA = Step;

}

GOUp()

{

switch (requested floor)

{

case 0x0d : while (step < 0xf3)

{ /* move towards current or request */ }

Step++;

PortA = Step;

Delay();

}

:0089x0 - 10 - Reset(); /* move back now before

:0089x0 - break; /* after back now before

:1089x0 - 10 - either :0089x0 now before

main case 0x0b : while (step > 0xf6) /* back now before

{

Step++; /* move back now before */

PortA = Step;

Delay();

}

(Reset()); /* now :0089x0 now before */

break;

Case 0x07 : while (step < 0xf9)

{

Step++;

PoutA = Step;

Delay();

}

Reset();

break;

}

}

GoDown();

{

Switch (requested floor);

{

case 0x0d : while (step > 0xf3);

{

Step--;

PoutA = Step;

Delay();

Reset();

break;

(condition - until superset) {

case 0x0b : while (step > 0xf6)

{

Step--;

PoutA = Step;

Delay();

} Reset();

break;

case 0x06 : while (&step > 0xfo)

{

Step --; // -get

PotfA = Step;

Delay(); // 10ms

}

C

Rwet(); // 10ms

break;

}

C

void main()

{

commonword = 0x82, 0x1f, 0x00, 0x00; // init

PotfA = 0xfo;

PresentFloor = 0x0c; // start = 0x0c + 0x00

while (1)

{

RequestedFloor = PotfB; // -get

RequestedFloor = RequestedFloor & 0x0f;

if (RequestedFloor != 0x0f && RequestedFloor != PresentFloor)

{

if (RequestedFloor < PresentFloor)

(0x0c - Goup()); // -get

else

Godewn(); // -get

PresentFloor = RequestedFloor;

}

RequestedFloor = PotfB; // -get

}