

```

#include<reg51.h>
#define display_port P1    //Data pins connected to port 2 on microcontroller
sbit rs = P3^2; //RS pin connected to pin 2 of port 3
sbit rw = P3^3; // RW pin connected to pin 3 of port 3
sbit en = P3^4; //E pin connected to pin 4 of port 3
sbit cen = P3^7;    //switch connection for dino movement
code unsigned char a[5]={0x02,0x38,0x0c,0x01,0x06};
code unsigned char
mov[16]={0xc0,0xc1,0xc2,0xc3,0xc4,0xc5,0xc6,0xc7,0xc8,0xc9,0xca,0xcb,0xcc,0xcd,0xce,0xcf}
;
code unsigned char
moc[16]={0x80,0x81,0x82,0x83,0x84,0x85,0x86,0x87,0x88,0x89,0x8a,0x8b,0x8c,0x8d,0x8e,0x
8f};
code unsigned char intro[]="WELCOME TO DINO";
code unsigned char intro2[]=" RUN....!!";
code unsigned char intro3[]="ENJOY...!!";
code unsigned char end[]="GAME OVER...";
int score[]={0xfc,0x60,0xda,0xf2,0x66,0xb6,0xbe,0xe0,0xfe,0xf6};

```

```

#define SegOne  0x01
#define SegTwo   0x02
#define SegThree 0x04
#define SegFour  0x08

```

```

void scores(void);
int dinojump(int);
unsigned int i,j,k,l,count;
int cnt, num, temp,temp1,temp2,temp3,h,zeo;
void delay(unsigned int time)
{
unsigned int i,j;
for(i=0;i< time;i++)
for(j=0;j<5;j++);
}
void cmdwrite()
{
rs=0;
rw=0;
en=1;
delay(1);
en=0;
}
void datawrite()

```

```

{
    rs=1;
    rw=0;
    en=1;
    delay(1);
    en=0;
}

void main()
{

    unsigned char cactii[8]={

0x04,
0x05,
0x15,
0x15,
0x0E,
0x04,
0x04,
0x04

};

    unsigned char dino[8] = {

0x0F,
0x0B,
0x0F,
0x0C,
0x0F,
0x0C,
0x1C,
0x06

};

    unsigned char blank[8]={

0x00,
0x00,
0x00,
0x00,
0x00,
0x00,
0x00,
0x00

};

```

```

int i;
    P1 = (0x40+8);
    cmdwrite();
    for(i=0;i<8;i++)
        {
            P1 = cactii[i];
            datawrite();
        }
        delay(50);
CGRAM                                     //making cactii and storing in 0x01

```

```

    P1 = (0x40+16);
    cmdwrite();
    for(i=0;i<8;i++)
        {
            P1 = dino[i];
            datawrite();
        }
        delay(50);
CGRAM                                     //making dino and storing in 0x02

```

```

    P1 = (0x40+24);
    cmdwrite();
    for(i=0;i<8;i++)
        {
            P1 = blank[i];
            datawrite();
        }
        delay(50);

```

```

for(i=0;i<=5;i++)
{
P1=a[i];
cmdwrite();
delay(100);
}
//*****INTRO*****//
P1=0x80;
cmdwrite();
i=0;
for(i=0;i<15;i++)
{

```

```

        P1=intro[i];
        datawrite();
        delay(1000);
    }
    P1=0xc3;
    cmdwrite();
    i=0;
    for(i=0;i<10;i++)
    {
        P1=intro2[i];
        datawrite();
        delay(1000);
    }
    P1=0X01;
    cmdwrite();
    delay(100);
    P1=0x82;
    cmdwrite();
    i=0;
    for(i=0;i<10;i++)
    {
        P1=intro3[i];
        datawrite();
        delay(1000);
    }
    delay(1000);
//*****//
cnt= 0;
while(1)
{
    cen = 1;
    zeo=0;
//*****process of moving the cactus while keeping the dino constant by erasing the
dinos created behind every original dino*****//
    for(i=0;i<14;i++)
    {
        cnt=++cnt;

        P1=mov[i+1];
        cmdwrite();
        scores();
    }
    P1=0x02;
    datawrite();
    scores();

```

```

P1=0xCF;
cmdwrite();
        scores();
//delay(100);

P1=0x01;        //display cactii
        datawrite();


P1=0x18;
cmdwrite();
        scores();

delay(1000);
        if(cen==0)
{
i=dinojump(i+1);

}


        for(l=0;l<i+1;l++)
{
P1=mov[i+1];
        cmdwrite();
        P1=0x03;
        datawrite();

}


if(i==13 && cen==1)        //game over
{
        P1=0x01;
        cmdwrite();
        P1=0x82;
        cmdwrite();
        for(i=0;i<12;i++)
        {
                P1=end[i];

```

```

        scores();
        datawrite();
        scores();
    }

    while(1)
    {
        P1=0x08;
        cmdwrite();
        delay(100);
        P1=0x0c;
        cmdwrite();
        delay(50);
        scores();
    }
}

//*****//

    }
    P1=0x01;
    cmdwrite();
    scores();
}
}

int dinojump(k)
{

    count=0;

        P1=mov[k];
        cmdwrite();
        P1=0x03;
        datawrite();

        P1=moc[k+2];
        cmdwrite();
P1=0x02;
datawrite();
        delay(10000);

        P1=moc[k+2];
        cmdwrite();
        P1=0x03;
        datawrite();

```

```
        P1=mov[k+3];
        cmdwrite();
P1=0x02;
datawrite();
delay(10000);
```

```
        P1=mov[k+3];
        cmdwrite();
P1=0x03;
datawrite();
```

```
return(k);
```

```
}
```

```
void scores()
{
```

```
//*****//
```

```
        for (h = 0; h < 300; h++)
```

```
        {
```

```
                                num = cnt;
```

```
                temp = num / 1000;
                num = num % 1000;
                P0 = SegOne;
                P2 = score[temp];
                //delay(10);
```

```
                temp1 = num / 100;
                num = num % 100;
                P0 = SegTwo;
                P2 = score[temp1];
                //delay(10);
```

```
                temp2 = num / 10;
                P0 = SegThree;
                P2 = score[temp2];
                //delay(10);
```

```
                temp3 = num % 10;
                P0 = SegFour;
```

```
P2 = score[temp3];  
//delay(10);  
}  
//*****//
```

```
}
```

```
//*****FULLY WORKING DINO RUN GAME ON 8051*****//  
//*****CLEAN VERSION*****//
```