

```

#include <p32xxxx.h>
#pragma config FPLLMUL = MUL_20, FPLLIDIV = DIV_2, FPLLODIV = DIV_1, FWDTEN = OFF
#pragma config POSCMOD = HS, FNOSC = PRIPLL, FPBDIV = DIV_8
#define TRUE 1
#define MOVE_LEFT 0
#define MOVE_RIGHT 1
#define STEP 2
#define DELAY_1us 80
#define START_FROM_FIRST_PAGE 0b10111000
#define START_FROM_FIRST_LINE 0b11111111
#define DISPLAY_ON 0b00111111
#define WHITE 0x00
void initPortD(void);
void initPortB(void);
void initPortE(void);
void initPortF(void);
void delay(int);
void enable_pulse(void);
void initLCD(void);
void set_curr_segment(int,int,int);
void write_image(unsigned int,int);
void fill_white(void);

int main()
{
    unsigned int i=0,j=0;
    unsigned int direction=MOVE_RIGHT;
    initPortB();
    initPortD();
    initPortE();
    initPortF();
    initLCD();
    fill_color(WHITE);
    while(TRUE)
    {
        for(j = 0;j < 128;j++) //columns-> x coordinate
        {
            set_curr_segment(j%64,3,((j&0x00000040)>>6)+1);
            // j between 0-63 then lcd_cs=1 and left side is activated
            // moving square 2 rows (16 pixel) height and 16 segments width
            if ((j>=(i*2))&& (j<((i*2)+16)))
                write_image(0xff,((j&0x00000040)>>6)+1);
            else
                write_image(0x00,((j&0x00000040)>>6)+1);
            set_curr_segment(j%64,4,((j&0x00000040)>>6)+1);
            if ((j>=(i*2))&& (j<((i*2)+16)))
                write_image(0xff,((j&0x00000040)>>6)+1);
            else
                write_image(0x00,((j&0x00000040)>>6)+1);
        }
        if(direction==MOVE_RIGHT)
        {
            i+=STEP;
            if(i==56)
                direction=MOVE_LEFT;
        }
        else // MOVE_LEFT

```

```
    {
        i-=STEP;
        if(i==0)
            direction=MOVE_RIGHT;
    }
    delay(DELAY_1us*1000);
}
return 1;
}

void enable_pulse(void)
{
    PORTDbits.RD4 = 1;//enable=1
    PORTDbits.RD4 = 0;//enable=0
}

void initPortD(void)
{
    TRISD &= 0xFFFFF4F;
    PORTDbits.RD4 = 0;
    PORTDbits.RD5 = 0;
    PORTDbits.RD7 = 0;
}

void initPortB(void)
{
    TRISB &= 0xFFFF7FFF;
    PORTBbits.RB15 = 0;
}

void initPortE(void)
{
    TRISE &= 0xFFFFF00;
    PORTE = 0x00;
}

void initPortF(void)
{
    TRISF &= 0xFFFFFEF8;
    PORTFbits.RF8 = 1;
}

void delay(int delay_value)
{
    unsigned int i;
    for(i=0;i<delay_value;i++);
}

void initLCD(void)
{
    int CONTROL[3] = {START_FROM_FIRST_PAGE,START_FROM_FIRST_LINE,DISPLAY_ON};
    int i;
    PORTDbits.RD5 = 0; // write mode
    PORTBbits.RB15 = 0; // write instruction
    PORTF = 0x01; // GLCD CS=1 (left side rectangele 64x64 pixel) and GLCD enable
    PORTDbits.RD7 = 0; // GLCD enable (operation)
```

```

PORTDbits.RD7 = 1;    // enable rise
PORTF = 0x02;        // GLCD CS=2 (right side rectangele 64x64 pixel) and GLCD enable
PORTDbits.RD7 = 0;    // GLCD enable (operation)
PORTDbits.RD7 = 1;    // enable rise
PORTFbits.RF8 = 1;
for(i = 0;i < 3;i++)
{
    PORTE = CONTROL[i];
    PORTF = 0x01;
    enable_pulse();
    delay(DELAY_1us*10);
    PORTF = 0x02;
    enable_pulse();
    delay(DELAY_1us*10);
}
PORTFbits.RF8 = 1;    // Disable GLCD
}

void set_curr_segment(int column, int row, int lcd_cs)
{
    PORTDbits.RD5 = 0;    // RD/nWR=0 write mode
    PORTBbits.RB15 = 0;    // RS=0 instruction mode
    PORTF = lcd_cs;        // b"01" - left rectangle is enabled, b"10" - right rectangle is
    enabled
    PORTE = 0x40 + column; // set column counter command: 0x40 + column index
    enable_pulse();
    delay(DELAY_1us/10);
    PORTE = 0xB8 + row;    // set page command: 0xb8 + page (row) number between 0-7
    enable_pulse();
    delay(DELAY_1us/10);
    PORTFbits.RF8 = 1;    // Disable GLCD
}

void write_image(unsigned int segment_data,int lcd_cs)
{
    int i;
    PORTDbits.RD5 = 0;    // RD/nWR=0 write mode
    PORTBbits.RB15 = 1;    // RS=1 data mode
    PORTF = lcd_cs;        // b"01" - left rectangle is enabled, b"10" - right rectangle is
    enabled
    PORTE = segment_data; // segment of 8 bit, each '1' into data means active pixel of
    segment
    enable_pulse();
    delay(DELAY_1us/10);
    PORTFbits.RF8 = 1;    // Disable GLCD
}

void fill_color(int color)
{
    int i,j;
    for(i = 0;i < 8;i++)
        for(j = 0;j < 128;j++) //columns-> x coordinate
        {
            set_curr_segment(j%64,i,((j&0x00000040)>>6)+1);
            write_image(color,((j&0x00000040)>>6)+1);
        }
    delay(DELAY_1us/4);
}

```

}