

```

#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 DELAY_1us 80
#define START_FROM_FIRST_PAGE 0xB8
#define START_FROM_FIRST_LINE 0xFF
#define DISPLAY_ON 0x3F
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 char image [8][128] = {
{0x7F, 0x7F, 0x7F, 0x7F, 0x7F, 0x7F, 0x7F, 0x7F, 0x7F, 0x7F, 0xF0, 0xC0, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x80, 0x40,
0x00,
0x00, 0x00, 0x08, 0x04, 0x04, 0x00, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02,
0x02,
0x00, 0x02, 0x00, 0x04, 0x08, 0x08, 0x10, 0x20, 0x20, 0x90, 0x08, 0x08, 0x04, 0x00, 0x02,
0x02,
0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x00, 0x04, 0x07, 0x0F, 0x17, 0x07,
0x67,
0x8F, 0x8F, 0x9F, 0x3F, 0x3F, 0xFF, 0x7F, 0x3F, 0x1F, 0x3F, 0x7F, 0xFF, 0x7F, 0x1F, 0x1F,
0x0F,
0x1F, 0x1F, 0x3F, 0x7B, 0x71, 0x18, 0x0C, 0x04, 0x0C, 0x18, 0x10, 0x30, 0x30, 0x1C, 0x0C,
0x04,
0x06, 0x0E, 0x0F, 0x1F, 0x37, 0x34, 0x0C, 0x0C, 0x0E, 0x3F, 0x3F, 0x3F, 0x3F, 0x3F, 0x1F,
0x1F},
{0xF0, 0xE0, 0xE0, 0xE0, 0xE0, 0xC0, 0xC0, 0x80, 0x80, 0x01, 0x07, 0x1F, 0xF0, 0xC0, 0x00,
0x00,
0x80, 0xF0, 0xF0, 0x60, 0xC0, 0x80, 0x00, 0x00, 0x00, 0x00, 0xF0, 0xFC, 0x01, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0xE0, 0xE0, 0xE0, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x8E, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0xE0, 0xF0, 0xF0, 0xE0, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0xC0, 0xFE, 0x00, 0x03, 0x03, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00},
{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0xFC, 0x1B, 0x1F, 0x7E,
0xF0,

```

```
0x3F, 0x3F, 0x01, 0x00, 0x00, 0x03, 0x1E, 0x38, 0x30, 0x00, 0x00, 0x02, 0x0C, 0x10, 0x20,
0x40,
0x80, 0x80, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x18, 0x0C, 0x07, 0x00, 0x00, 0x04,
0x04,
0x04, 0x04, 0x04, 0x04, 0x00, 0x00, 0x08, 0x00, 0x00, 0x08, 0x18, 0xF0, 0xF0, 0x10, 0x08,
0x04,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00},
{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFB, 0x1C, 0x04, 0x26,
0x27,
0xE2, 0x62, 0x22, 0x66, 0x66, 0xC4, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x01, 0x01, 0x02, 0x82, 0x84, 0xC4, 0x44, 0x44, 0x24, 0x24, 0x24, 0x24, 0x24,
0x24,
0x20, 0x22, 0x22, 0x20, 0x31, 0x31, 0x30, 0x30, 0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x10,
0x10,
0x10, 0x10, 0x10, 0x10, 0x10, 0x18, 0x08, 0x08, 0x08, 0x0C, 0x04, 0x07, 0x19, 0x30, 0x20,
0x60,
0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00},
{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFC, 0xFC, 0xF8,
0xF8,
0xF0, 0xF0, 0xF0, 0x10, 0x18, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xE0,
0x38, 0x18, 0x0C, 0x03, 0x03, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x01, 0x07, 0x1E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x10, 0x08, 0x08,
0x00,
0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x00, 0x08, 0x08, 0x10, 0x10,
0x00,
0x40, 0xE0, 0x20, 0x00, 0x08, 0x00, 0x00, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04,
0x04},
{0xFF, 0xFF, 0xFF, 0xFF, 0x7F, 0x7F, 0x3F, 0x1F, 0x1F, 0x0F, 0x07, 0x07, 0x03, 0x03, 0x03,
0x01,
0xE1, 0xFF, 0x3F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,
0xFF,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x80,
0x80,
0xC0, 0x40, 0x20, 0x1E, 0x00, 0x00, 0x68, 0x86, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```

0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0xC0, 0xC0, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x01, 0x88, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80,
0xC0},
{0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0,
0x78,
0x07, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03,
0x1F,
0xF0, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01, 0x01, 0x03, 0x02, 0x02, 0x02,
0x02,
0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04,
0x04,
0x84, 0xC4, 0x44, 0x34, 0x34, 0x14, 0x0C, 0x0C, 0x06, 0x02, 0x02, 0x02, 0x3F, 0xF9, 0xC1,
0x01,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x0C, 0x20, 0x00, 0x40, 0x80, 0x80, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x01, 0x03, 0x03, 0x03, 0x03, 0x00, 0x00, 0x00, 0x80, 0x80, 0x00,
0x40,
0x20, 0x10, 0x0C, 0x09, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x00, 0x00, 0x08,
0x09},
{0x03, 0x03, 0x03, 0x06, 0x38, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1E, 0x3F, 0x23,
0x40,
0xC0, 0xC0, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x01, 0x06, 0x04, 0x0C, 0x18, 0x10, 0x10, 0x30, 0x20, 0x60, 0x60, 0x40, 0x40,
0x40,
0x40, 0x40, 0x40, 0x40, 0x40, 0x40, 0x40, 0xC0, 0xC0, 0x60, 0x20, 0x30, 0x18, 0x0C, 0x0E,
0x03,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x60, 0x10, 0x11,
0x8F,
0xC8, 0xC8, 0x48, 0x48, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01,
0x01,
0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x00, 0x01, 0x01, 0x00, 0x00, 0x00,
0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x10, 0x10, 0x10, 0x10, 0x08, 0x08, 0x08, 0x08,
0x0C}
};

```

```

initPortB();
initPortD();
initPortE();
initPortF();
initLCD();
fill_white();
for(i = 0;i < 8;i++) //rows-> y coordinate
    for(j = 0;j < 128;j++) //columns-> x coordinate
    {
        set_curr_segment(j%64,i,((j&0x00000040)>>6)+1);
        // j between 0-63 then lcd_cs=1 and left side is activated
        write_image(image[i][j],((j&0x00000040)>>6)+1);
    }
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; // Disable GLCD

    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_white(void)
{
    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);
            // j between 0-63 then lcd_cs=1 and left side is activated
            write_image(0x00,((j&0x00000040)>>6)+1);
        }
    delay(DELAY_1us/4);
}

```