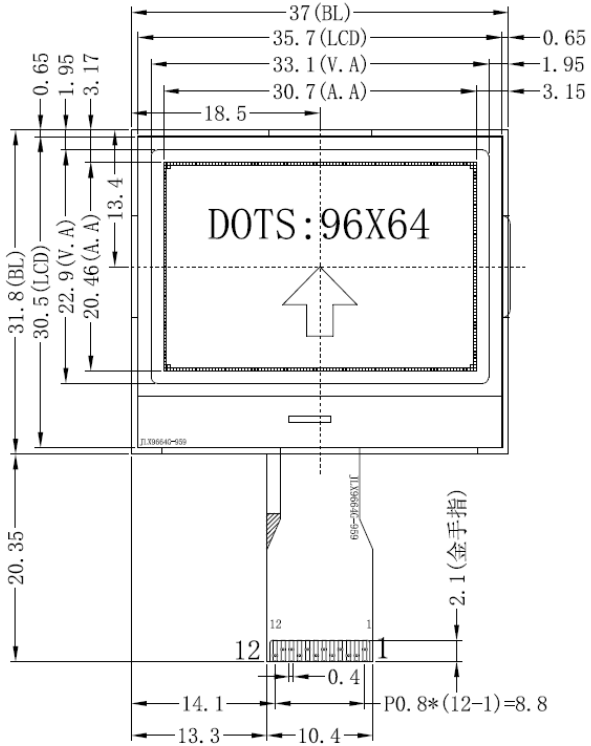
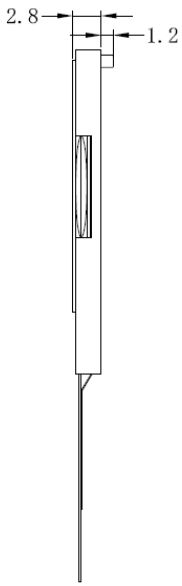
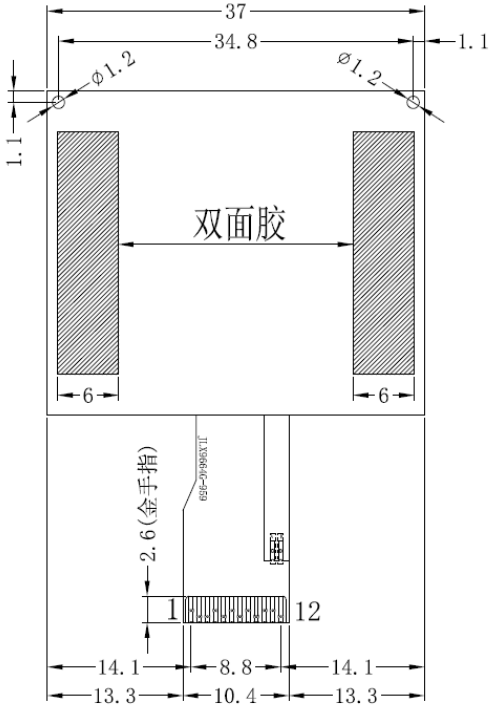
正面图　　侧面图　　背面图

接口定义：

1	CSB
2	RST
3	A0
4	SCK
5	SDA
6	VDD
7	VSS
8	V0
9	XV0
10	VG
11	LEDA
12	LEDK



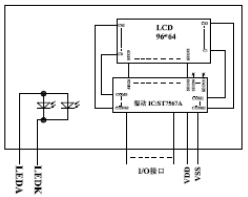


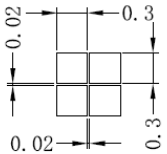


说明：

1.LCM包括LCD、BACKLIGHT、FPC、IC；
2.LCM工作电压VDD=3.3V；
3.工作温度-20~+70摄氏度；
4.储存温度-30~+80摄氏度；
5.视角为6点钟；
6.LCD驱动条件为1/64Duty，1/9Bias,Vop=8.2V；
7.背光为白色2颗LED侧背光，3.0V，16~40mA；
8.LCD底色白底黑字或蓝底白字。
9.IC型号：ST7567A
10.连接方式：COG(Chip On Glass)

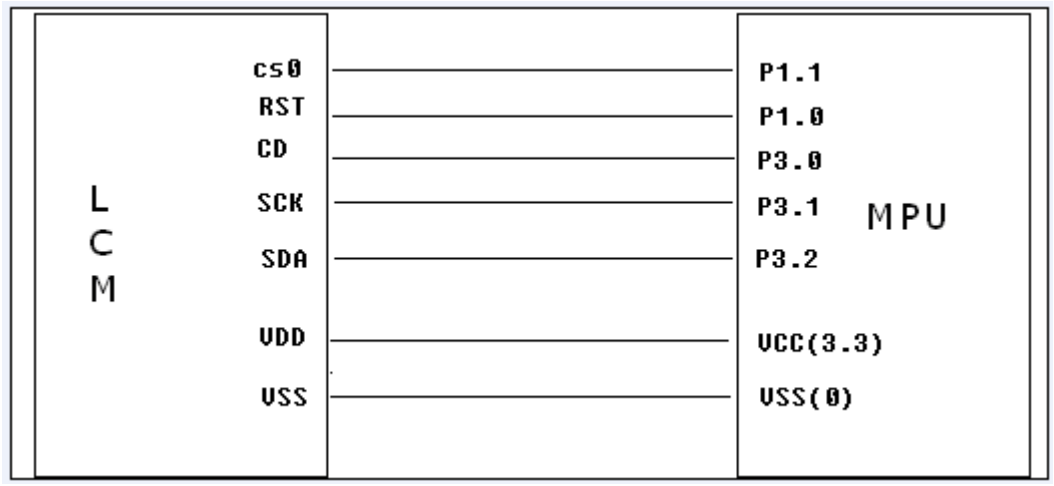
电路框图





SCALE:10:1

REVISION RECORD							
A	FIRST.					未注公差为：±0.2	
B		Model No:JLX9664G-959-BN				PEG(3)	
C		Part No:FOG+BL				VER:	A
D		DRAWN	SHEN	DATE	2021/07/03	SHEET	1/2
E		CHECKED		DATE		UNIT:	mm
F		APPROVED		DATE		SCALE	1:1




```
*/
#include <reg51.H>
#include <chinese.h>  //取模数据存放处

sbit key=P2^0;   //对应我司主板按键接口
sbit cs0=P1^1;   //对应 LCD 的 CS 引脚
sbit RST=P1^0; //对应 LCD 的 RST 引脚
sbit CD=P3^0;    //对应 LCD 的 RS 引脚
sbit SCK=P3^1;  //对应 LCD 的 SCK 引脚
sbit SDA=P3^2;   //对应 LCD 的 SDA 引脚

/*延时*/
void delay(int i)
{
      int j,k;
      for(j=0;j<i;j++)
            for(k=0;k<110;k++);
}

/*短延时*/
void delay_us(int i)
{
      int j,k;
      for(j=0;j<i;j++)
            for(k=0;k<1;k++);
}

void waitkey()
{
//repeat:  if(key==1) goto repeat;
```

```
//          else
                  delay(2000);
}

/*写指令到LCD模块*/
void transfer_command(int data1)
{
      char i;
      cs0=0;
      CD=0;
      for(i=0;i<8;i++)
      {
            SCK=0;
            if(data1&0x80) SDA=1;
            else SDA=0;
            SCK=1;
            delay_us(1);
            data1=data1<<=1;
      }
      cs0=1;
}

/*写数据到LCD模块*/
void transfer_data(int data1)
{
      char i;
      cs0=0;
      CD=1;
      for(i=0;i<8;i++)
      {
            SCK=0;
            if(data1&0x80) SDA=1;
            else SDA=0;
            SCK=1;
            delay_us(1);
            data1=data1<<=1;
      }
      cs0=1;
}

/*LCD模块初始化*/
//------对比度值设置,粗度0x24,微调0x1e------//
void initial_lcd()
{
      RST=0;         /*低电平复位*/
      delay(100);      //100ms
      RST=1;        /*复位完毕*/
```

```
    delay(200);  //200ms
    transfer_command(0xe2);      /*软复位*/
    delay(5);  //5ms
    transfer_command(0x2c);  /*升压步聚 1*/
    delay(5);   //5ms
    transfer_command(0x2e);  /*升压步聚 2*/
    delay(5);  //5ms
    transfer_command(0x2f);  /*升压步聚 3*/
    delay(5);  //5ms
    transfer_command(0x24);  /*粗调对比度，可设置范围 0x20～0x27*/
    transfer_command(0x81);  /*微调对比度*/
    transfer_command(0x21);  /*微调对比度的值，可设置范围 0x00～0x3f*/
    transfer_command(0xa2);  /*1/9 偏压比（bias）*/

    transfer_command(0xc8);  /*行扫描顺序：从上到下*/
    transfer_command(0xa0);  /*列扫描顺序：从左到右*/

    transfer_command(0x40);  /*起始行：第一行开始*/
    transfer_command(0xaf);  /*开显示*/
}

//==========设置 LCD 地址函数==========//
void lcd_address(uchar page,uchar column)
{
    column=column-1;                                    //我们平常所说的第 1 列，在 LCD 驱动 IC 里是第 0 列。所以在这里减去 1.
    page=page-1;
    transfer_command(0xb0+page);                        //设置页地址。每页是 8 行。一个画面的 64 行被分成 8 个页。我们平常所说的第 1 页，在 LCD
驱动 IC 里是第 0 页，所以在这里减去 1
    transfer_command(((column>>4)&0x0f)+0x10);   //设置列地址的高 4 位
    transfer_command(column&0x0f);                      //设置列地址的低 4 位
}

/*全屏清屏*/
void clear_screen()
{
    unsigned char i,j;
    for(i=0;i<9;i++)
    {
        lcd_address(1+i,1);
        for(j=0;j<132;j++)
        {
            transfer_data(0x00);
        }
    }
}

//=======显示 96*64 点阵图像=========//
```

```
void display_graphic(uchar *dp)
{
      uchar i,j;
      for(j=0;j<8;j++)
      {
            lcd_address(j+1,1);
            for(i=0;i<96;i++)
            {
                  transfer_data(*dp);
                  dp++;
            }
      }
}

/*显示 32x32 点阵图像、汉字、生僻字或 32x32 点阵的其他图标*/
void display_graphic_32x32(uchar page,uchar column,uchar *dp)
{
      uchar i,j;
      for(j=0;j<4;j++)
      {
            lcd_address(page+j,column);
            for (i=0;i<31;i++)
            {
                  transfer_data(*dp);          /*写数据到 LCD,每写完一个 8 位的数据后列地址自动加 1*/
                  dp++;
            }
      }
}

/*显示 16x16 点阵图像、汉字、生僻字或 16x16 点阵的其他图标*/
void display_graphic_16x16(uchar page,uchar column,uchar reverse,uchar *dp)
{
      uchar i,j;
      for(j=0;j<2;j++)
      {
            lcd_address(page+j,column);
            for (i=0;i<16;i++)
            {
                  if(reverse==1)
                  {
                        transfer_data(~*dp);         /*写数据到 LCD,每写完一个 8 位的数据后列地址自动加 1*/
                  }
                  else
                        transfer_data(*dp);
                  dp++;
            }
      }
```

```
}

/*显示 8x16 点阵图像、ASCII, 或 8x16 点阵的自造字符、其他图标*/
void display_graphic_8x16(uchar page,uchar column,uchar *dp)
{
      uchar i,j;
      for(j=0;j<2;j++)
      {
            lcd_address(page+j,column);
            for (i=0;i<8;i++)
            {
                  transfer_data(*dp);                              /*写数据到 LCD,每写完一个 8 位的数据后列地址自动加 1*/
                  dp++;
            }
      }
}

//显示一串 8x16 点阵的字符串
//括号里的参数分别为（页，列，是否反显，数据指针）
void display_string_8x16(uint page,uint column,uchar reverse,uchar *text)
{
      uint i=0,j,k,n,data1;
      while(text[i]>0x00)
      {
            if((text[i]>=0x20)&&(text[i]<=0x7e))
            {
                  j=text[i]-0x20;
                  for(n=0;n<2;n++)
                  {
                        lcd_address(page+n,column);
                        for(k=0;k<8;k++)
                        {
                              if(reverse==1)   data1=~ascii_table_8x16[j][k+8*n];
                              else  data1=ascii_table_8x16[j][k+8*n];
                              transfer_data(data1);
                        }
                  }
                  i++;
                  column+=8;
            }
            else
            i++;
      }
}

//显示一串 5x8 点阵的字符串
//括号里的参数分别为（页，列，是否反显，数据指针）
```

```
void display_string_5x8(uint page,uint column,uchar reverse,uchar *text)
{
      uchar i=0,j,k,data1;
      while(text[i]>0x00)
      {
            if((text[i]>=0x20)&&(text[i]<=0x7e))
            {
                  j=text[i]-0x20;
                  lcd_address(page,column);
                  for(k=0;k<5;k++)
                  {
                        if(reverse==1)   data1=~ascii_table_5x8[j][k];
                        else  data1=ascii_table_5x8[j][k];
                        transfer_data(data1);
                  }
                  if(reverse==1)   transfer_data(0xff);
                  else  transfer_data(0x00);
                  i++;
                  column+=6;
            }
            else
            i++;
      }
}

//睡眠模式
void sleep()  //进入睡眠
{
      transfer_command(0xac);/*静态图标关闭*/
      transfer_command(0x00);/*静态图标寄存器设置：关闭。此指令与上述指令一起完成静态图标关闭*/
      transfer_command(0xae);/*显示：关*/
      transfer_command(0xa5);/*全屏显示：开*/
}
void wake()  //退出睡眠
{
      transfer_command(0xa4);/*全屏显示：关。进入正常模式*/
      transfer_command(0xad);/*静态图标开启*/
      transfer_command(0x03);/*静态图标寄存器设置：开。此指令与上述指令一起完成静态图标开启*/
      transfer_command(0xaf);/*显示：开*/
}

//主程序
void main(void)
{
      initial_lcd();  //LCD 初始化
      while(1)
      {
```

```
        clear_screen();   //clear all dots
        display_graphic(bmp1);  //显示 96*64 的图片
        waitkey();  //按键可用延时代替
        clear_screen();   //clear all dots
        display_string_5x8(1,1,1,"      MENU      "); //显示 5x8 点阵的字符串，括号里的参数分别为（页，列，是否反显，数据指针）
        display_string_5x8(3,1,0,"Select>");          //同上
        display_string_5x8(3,43,1,"1.Graphic");       //同上
        display_string_5x8(4,43,0,"2.Chinese");       //同上
        display_string_5x8(5,43,0,"3.Movie  ");       //同上
        display_string_5x8(6,43,0,"4.Contrast");      //同上
        display_string_5x8(7,43,0,"5.Mirror ");       //同上
        display_string_5x8(8,1,0,"PRE USER DEL NEW"); //同上
        waitkey();  //按键可用延时代替
        clear_screen();   //clear all dots
        display_graphic_32x32(1,17,cheng1);          //在第 1 页，第 17 列显示单个汉字"成"*/
        display_graphic_32x32(1,49,gong);            //在第 1 页，第 49 列显示单个汉字"功"*/
        display_graphic_16x16(6,1,1,zhuang1);        //在第 6 页，第 1 列显示单个汉字"状"
        display_graphic_16x16(6,(1+16),1,tai1);      //在第 6 页，第 17 列显示单个汉字"态"
        display_string_8x16(6,33,0,":");             //显示":"
        display_graphic_16x16(6,41,0,shi1);          //在第 6 页，第 41 列显示单个汉字"使"
        display_graphic_16x16(6,(1+16*3+8),0,yong1);//在第 6 页，第 57 列显示单个汉字"用"
        display_string_8x16(6,79,0,"00");            //显示 8x16 点阵的字符串，括号里的参数分别为（页，列，是否反显，数据指针）
        waitkey(); //按键可用延时代替
        clear_screen();   //clear all dots
        display_string_8x16(1,1,0,"0123456789abcdef");     //显示 8x16 点阵的字符串，括号里的参数分别为（页，列，是否反显，数据指针）
        display_string_8x16(3,1,0,"~`!@#$%^&*()_-+=");     //同上
        display_string_5x8(5,1,1," ! #$%&'()*+,-./01234");//显示 5x8 点阵的字符串，括号里的参数分别为（页，列，是否反显，数据指针）
        display_string_5x8(6,1,0,"56789:;<=>?@ABCDEFGHI");//同上
        display_string_5x8(7,1,0,"JKLMNOPQRSTUVWXYZ[\]^");//同上
        display_string_5x8(8,1,0,"_`abcdefghijklmnopqrs");//同上
        waitkey();  //按键可用延时代替
    }
}
```

-END-