大连佳显电子有限公司

GOOD DISPLAY

YM12864-G35BSW(Y/B/G)

SPECIFICATION

DOC.REVISION A00

Customer Approval:

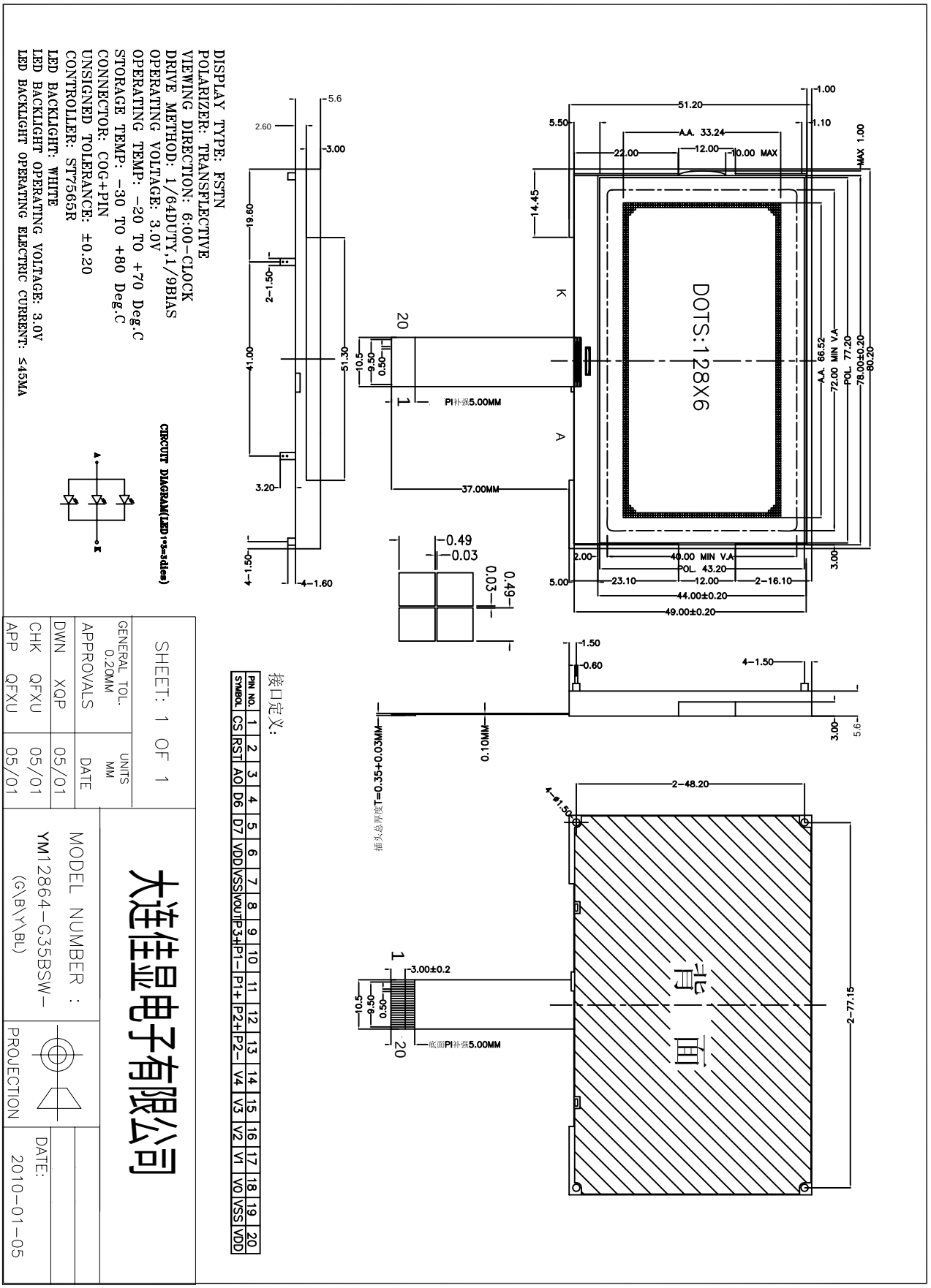
	NAME	SIGNATURE	DATE
PREPARED BY			30th Mar 2010
APPROVED BY			

DOCUMENT REVISION HISTORY

Version	DATE	DESCRIPTION	CHANGED BY
A00	16-Mar-2010	First issue	

CONTENTS

1.DIMENSIONAL OUTLINE



2.FUNCTIONS & FEATURES

2-1. Format	: 128 *64dots
2-2. LCD mode	: FSTN, Positive Mode
2-3. Viewing direction	: 6 o'clock
2-4. Driving scheme	: 1/64Duty , 1/9 Bias
2-5. Driver IC	: ST7565R

3.MECHANICAL SPECIFICATIONS

3-1. Module size	80.20mm(L)*51.20mm(W) *5.6mm(H)
3-2. Viewing area	66.52mm(L)*33.24mm(W)

4. PIN DESCRIPTION

Pin no.	Symbol	Function(parallel)
20	IRS	This terminal selects the resistors for the V0 voltage level IRS="H" use the internal resistor IRS="L" not use the internal
19	P/S	This pin configures the interface to be parallel mode or serial mode.
18	V0	This is a multi-level power supply for the liquid crystal drive.
17	V1	
16	V2	
15	V3	
14	V4	
13	CAP2-	DC/DC voltage converter. Connect a capacitor between this terminal and the CAP2P terminal.
12	CAP2+	DC/DC voltage converter. Connect a capacitor between this terminal and the CAP2N terminal.
11	CAP1+	DC/DC voltage converter. Connect a capacitor between this terminal and the CAP1N terminal.
10	CAP1-	DC/DC voltage converter. Connect a capacitor between this terminal and the CAP1P terminal.
9	CAP3+	DC/DC voltage converter. Connect a capacitor between this terminal and the CAP1N terminal.
8	VOUT	DC/DC voltage converter output.
7	VSS	Ground
6	VDD	Power supply
5	D7/SDA	Serial data input
4	D6/SCL	Serial clock input
3	A0	This is connect to the least significant bit of the normal MPU address bus, and it determines whether the data bits are data or command.
2	RST	When/RES is set to "L", the settings are initialized
1	CS	This is the chip select signal.

BACKLIGHT SPECIFIATIONS

Item	Symbol	Min	Type	Max	Unit
Forward voltage	Vf		3.0		V
Forward current	Ir		45		mA

5.MAXIMUM ABSOLUTE LIMIT (T=25°C)

Unless otherwise noted, V_{SS} = 0V

Table 17

Parameter		Symbol	Conditions	Unit
Power Supply Voltage		VDD	-0.3 ~ 3.6	V
Power supply voltage (V_{DD} standard)		VDD2	-0.3 ~ 3.6	V
Power supply voltage (V_{DD} standard)		V_0, V_{OUT}	-0.3 ~ 13.5	V
Power supply voltage (V_{DD} standard)		V_1, V_2, V_3, V_4	-0.3 to V_0	V
Operating temperature		T_{OPR}	–30 to +85	°C
Storage temperature	Bare chip	T_{STR}	–65 to +150	°C

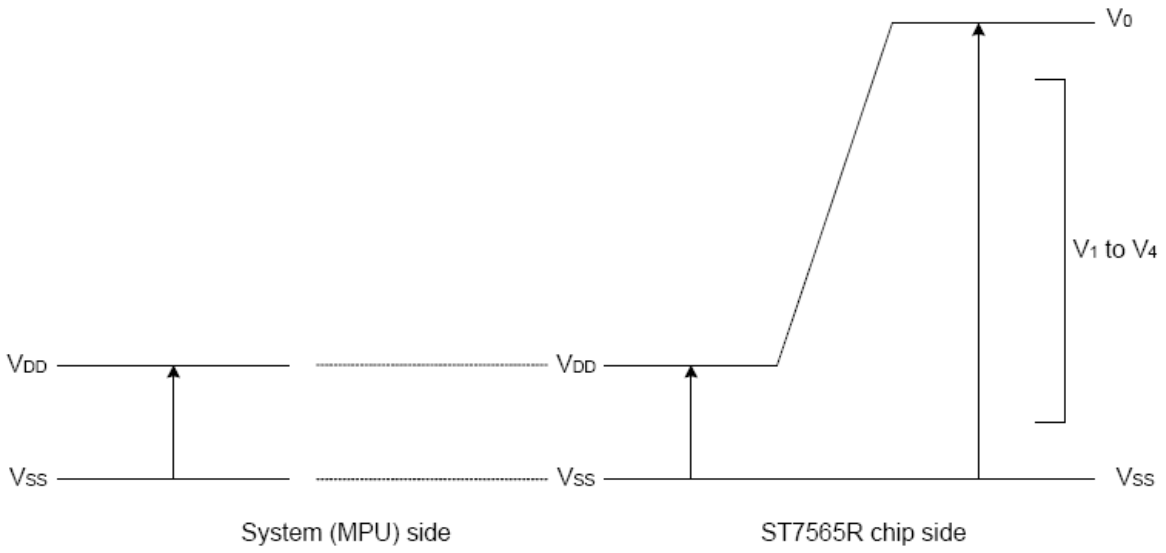


Figure 30

Notes and Cautions

1. The V_{DD2}, V_0 to V_4 and V_{OUT} are relative to the V_{SS} = 0V reference.
2. Insure that the voltage levels of V_1, V_2, V_3, and V_4 are always such that $V_{OUT} \geqq V_0 \geqq V_1 \geqq V_2 \geqq V_3 \geqq V_4$.
3. Permanent damage to the LSI may result if the LSI is used outside of the absolute maximum ratings. Moreover, it is recommended that in normal operation the chip be used at the electrical characteristic conditions, and use of the LSI outside of these conditions may not only result in malfunctions of the LSI, but may have a negative impact on the LSI reliability as well.

6.ELECTRICAL CHARACTERISTICS

Unless otherwise specified, V_{SS} = 0 V, V_{DD} = 3.0 V, Ta = –30 to 85°C

Table 18

Item		Symbol	Condition		Rating Min.	Rating Typ.	Rating Max.	Units	Applicable Pin
Operating Voltage (1)		V_{DD}			1.8	—	3.3	V	V_{DD}*1
Operating Voltage (2)		V_{DD2}	(Relative to V_{SS})		2.4	—	3.3	V	V_{DD}
High-level Input Voltage		V_{IHC}			0.8 x V_{DD}	—	V_{DD}	V	*3
Low-level Input Voltage		V_{ILC}			V_{SS}	—	0.2 x V_{DD}	V	*3
High-level Output Voltage		V_{OHC}	I_{OH} = –0.5 mA		0.8 x V_{DD}	—	V_{DD}	V	*4
Low-level Output Voltage		V_{OLC}	I_{OL} = 0.5 mA		V_{SS}	—	0.2 x V_{DD}	V	*4
Input leakage current		I_{LI}	V_{IN} = V_{DD} or V_{SS}		–1.0	—	1.0	µA	*5
Output leakage current		I_{LO}	V_{IN} = V_{DD} or V_{SS}		–3.0	—	3.0	µA	*6
Liquid Crystal Driver ON Resistance		R_{ON}	Ta = 25°C (Relative to V_{SS})	V_0 = 13.0 V	—	2.0	3.5	KΩ	SEGn COMn *7
				V_0 = 8.0 V	—	3.2	5.4		
Static Consumption Current		I_{SSQ}	V_0 = 13.0 V (Relative To V_{SS})		—	0.01	2	µA	V_{DD}, V_{DD2}
Output Leakage Current		I_{OQ}			—	0.01	10	µA	V_0
Input Terminal Capacitance		C_{IN}	Ta = 25°C, f = 1 MHz		—	5.0	8.0	pF	
Oscillator Frequency	Internal Oscillator	f_{OSC}	1/65 duty 1/33 duty	Ta = 25°C	17	20	24	kHz	*8
	External Input	f_{CL}			17	20	24	kHz	CL
	Internal Oscillator	f_{OSC}	1/49 duty 1/53 duty 1/55 duty	Ta = 25°C	25	30	35	kHz	*8
	External Input	f_{CL}			25	30	35	kHz	CL

Table 19

Item		Symbol	Condition	Rating Min.	Rating Typ.	Rating Max.	Units	Applicable Pin
Internal Power	Input voltage	V_{DD2}	(Relative To V_{SS})	2.4	—	3.3	V	V_{DD}
	Supply Step-up output voltage Circuit	V_{OUT}	(Relative To V_{SS})	—	—	13.5	V	V_{OUT}
	Voltage regulator Circuit Operating Voltage	V_{OUT}	(Relative To V_{SS})	6.0	—	13.5	V	V_{OUT}
	Voltage Follower Circuit Operating Voltage	V_0	(Relative To V_{SS})	4.0	—	13.5	V	V_0 * 9
	Base Voltage	VRS	Ta = 25°C, (Relative To V_{SS}) –0.05%/°C	2.07	2.10	2.13	V	*10

7.AC CHARACTERISTICS

The 4-line SPI Interface

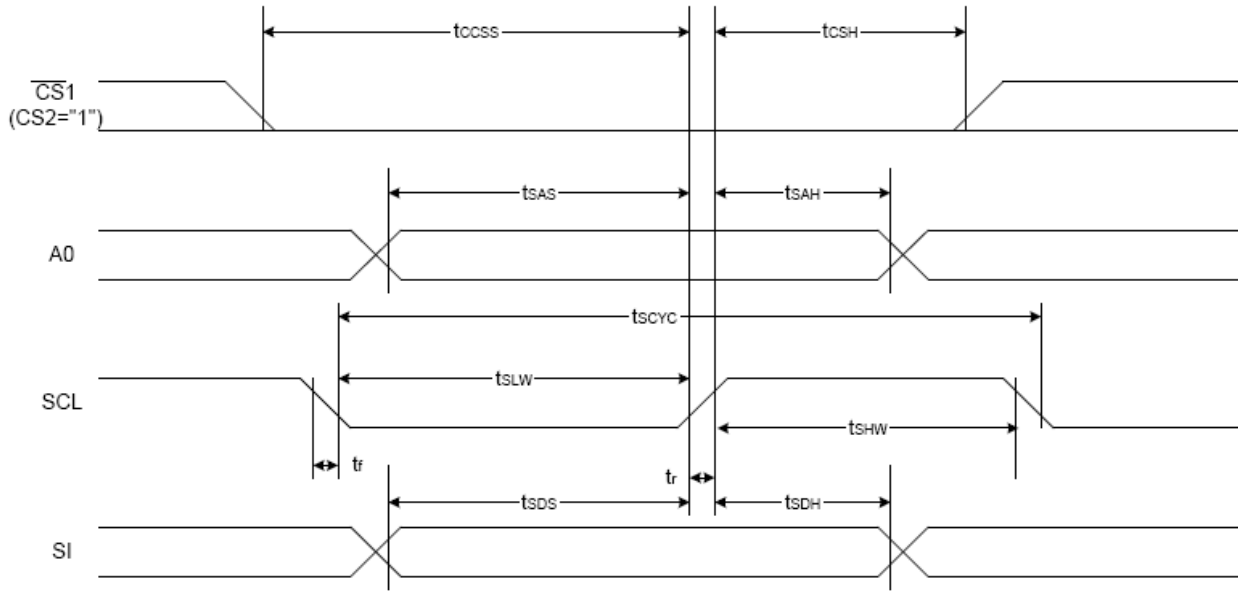


Figure 39

Table 30

(VDD = 3.3V,Ta = –30 to 85°C)

Item	Signal	Symbol	Condition	Rating		Units
				Min.	Max.	
4-line SPI Clock Period	SCL	Tscyc		50	—	ns
SCL "H" pulse width		Tshw		25	—	
SCL "L" pulse width		TSLW		25	—	
Address setup time	A0	TSAS		20	—	
Address hold time		Tsah		10	—	
Data setup time	SI	Tsds		20	—	
Data hold time		TSDH		10	—	
CS-SCL time	CS	Tcss		20	—	
CS-SCL time		Tcsh		40	—	

Table 31

(VDD = 2.7V,Ta = –30 to 85°C)

Item	Signal	Symbol	Condition	Rating		Units
				Min.	Max.	
4-line SPI Clock Period	SCL	Tscyc		100	—	ns
SCL "H" pulse width		TSHW		50	—	
SCL "L" pulse width		TSLW		50	—	
Address setup time	A0	TSAS		30	—	
Address hold time		TSAH		20	—	
Data setup time	SI	TSDS		30	—	
Data hold time		TSDH		20	—	
CS-SCL time	CS	TCSS		30	—	
CS-SCL time		TCSH		60	—	

8.REFERENCE APPLICATIONS

The ST7565R Series can be connected to either 80X86 Series MPUs or to 6800 Series MPUs. Moreover, using the 4-line SPI interface it is possible to operate the ST7565R series chips with fewer signal lines.
The display area can be enlarged by using multiple ST7565R Series chips. When this is done, the chip select signal can be used to select the individual ICs to access.

(1) 8080 Series MPUs

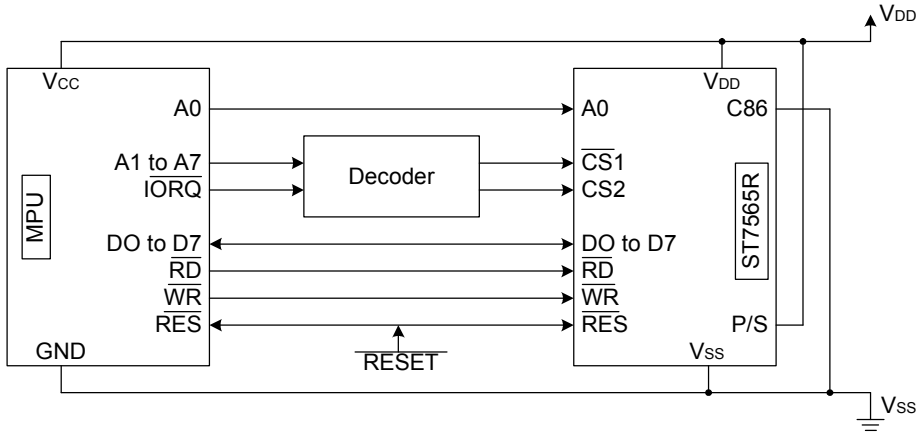


Figure 42-1

(2) 6800 Series MPUs

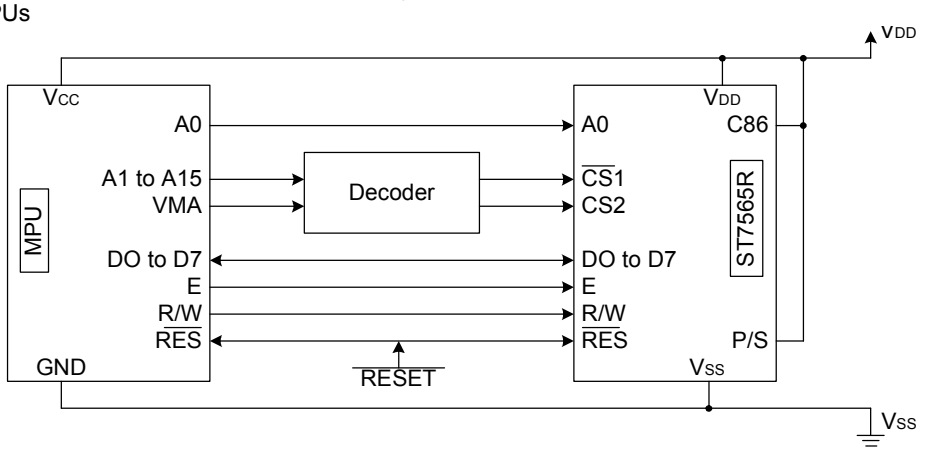


Figure 42-2

(3) Using the 4-line SPI Interface

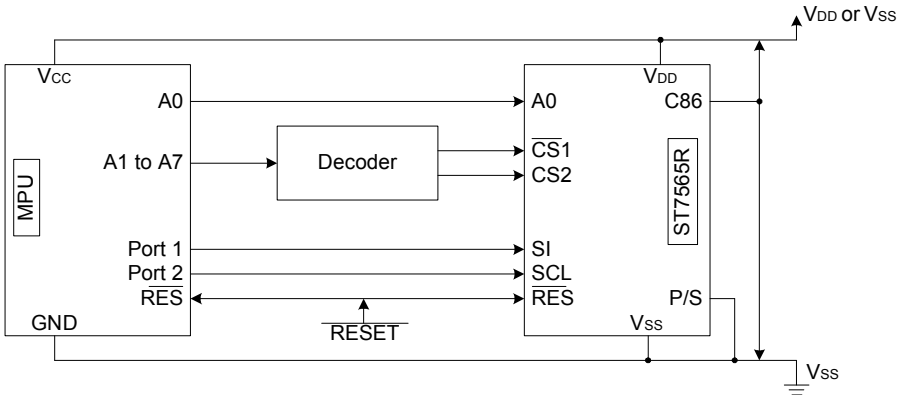
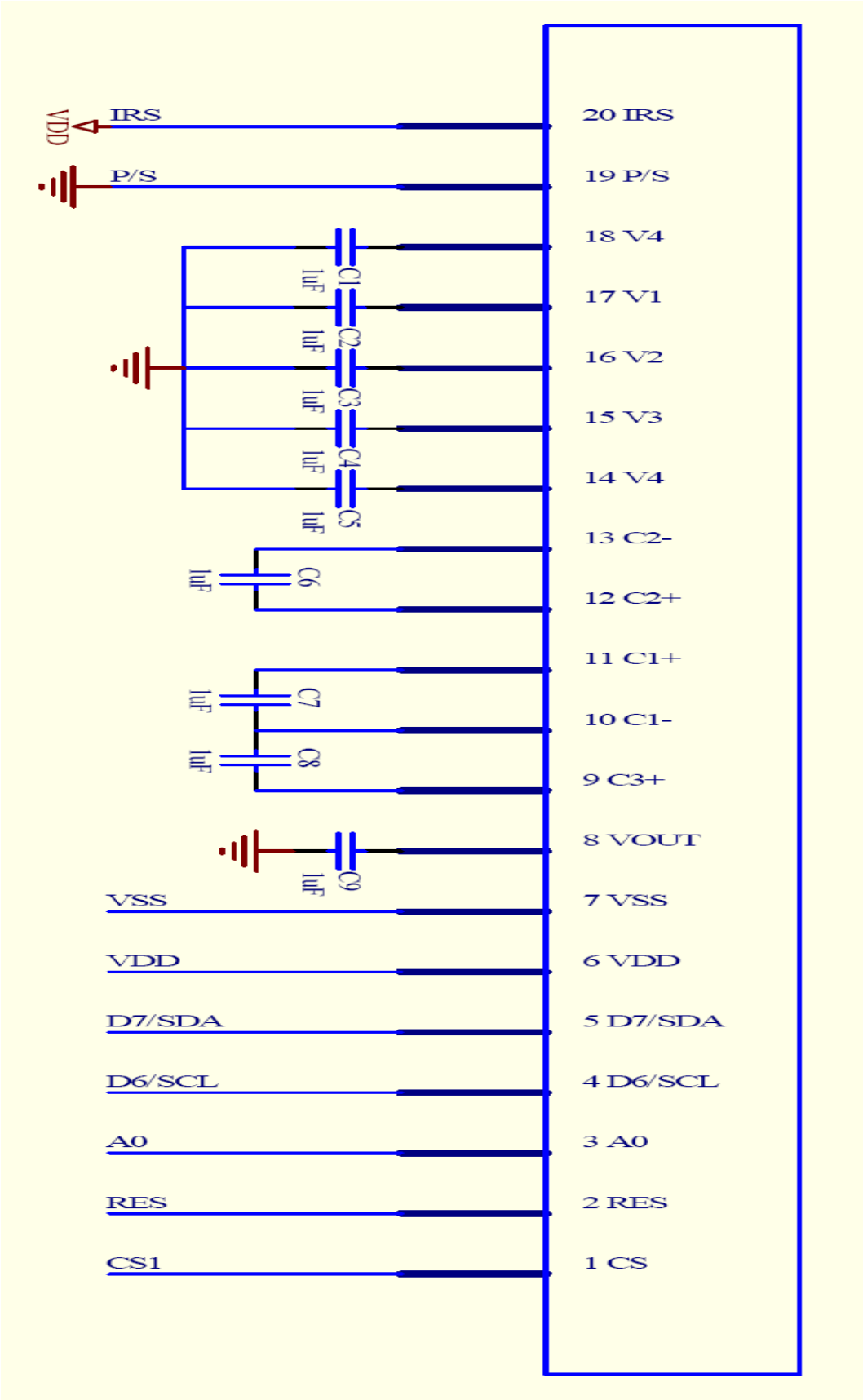


Figure 42-3


VDD
IRS
P/S
C1 1uF
C2 1uF
C3 1uF
C4 1uF
C5 1uF
C6 1uF
C7 1uF
C8 1uF
C9 1uF
VSS
VDD
D7/SDA
D6/SCL
A0
RES
CS1
20 IRS
19 P/S
18 V4
17 V1
16 V2
15 V3
14 V4
13 C2-
12 C2+
11 C1+
10 C1-
9 C3+
8 VOUT
7 VSS
6 VDD
5 D7/SDA
4 D6/SCL
3 A0
2 RES
1 CS

```
/************************************/
/*          IC:ST7565R              */
/*          Date:10-03-23           */
/************************************/
/************************************/
/*          Head File Define        */
/************************************/
#include <reg51.h>
#include <E:\driver_prg\image\char.h>
//#include <math.h>
//#include <intrins.h>

/************************************/
/*          Interface Define        */
/************************************/
sbit        RESET=P3^0;
sbit        CD=P3^4;
sbit        CS=P3^2;
sbit        SCL=P1^6;
sbit        SDA=P1^7;
sbit        KEY_PRESS=P3^3;
sbit        DEC_KEY=P3^1;
sbit        EXIT_KEY=P3^7;
/************************************/
/*          Parameter Define        */
/************************************/
#define    CONTRAST=0x0C
#define    DATA0=0x40
#define    DATA1=0x41
unsigned char    page_width;
unsigned char    seg_width;

unsigned char Buf_1,Buf_2,Buf_3,conversion;
unsigned char *char_point;
int    contrast;
/************************************/
/*          Picture data            */
/************************************/
unsigned char code row_table[]={0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};

unsigned char code char_0[]={62,65,65,62,0 };
unsigned char code char_1[]={0,66,127,64,0 };
```

```
unsigned char code char_2[]={98,81,73,70,0};
unsigned char code char_3[]={34,73,73,54,0 };
unsigned char code char_4[]={28,18,127,16,0 };
unsigned char code char_5[]={39,69,69,57,0};
unsigned char code char_6[]={62,73,73,50,0};
unsigned char code char_7[]={1,121,7,1,0};
unsigned char code char_8[]={54,73,73,54,0 };
unsigned char code char_9[]={6,73,73,62,0};
unsigned char code orise_tech[]={


/************************************/
/*          Time Delay                  */
/************************************/
void Delay(long i)
{
        while(i!=0)
             {
               i--;
              }
}


/************************************/
/*          Transfer Commmand           */
/************************************/
void Write_Command(unsigned char command)
{
        int i,j;
        j=0x80;
        CS=0;
        CD=0;
        for(i=0;i<8;i++)
        {
             SCL=0;
             if(command&j)SDA=1;
             else SDA=0;
             SCL=1;
             j=j>>1;
        }
        CS=1;
}
```

```
/***********************************/
/*          Transfer Data                  */
/***********************************/
void Write_Data(unsigned char data1)
{
        int i,j;
        j=0x80;
        CS=0;
        CD=1;
        for(i=0;i<8;i++)
        {
             SCL=0;
             if(data1&j)SDA=1;
             else SDA=0;
             SCL=1;
             j=j>>1;
        }
        CS=1;
}
//-------------------------------------------------------------------
void Set_column_addr(unsigned char add)
{
        unsigned char temp;
        temp=add;
        add=add>>4;
        add=add&0x0f;
        add=add|0x10;
        Write_Command(add);  //Set upper addr;
        add=temp;
        add=add&0x0F;
        Write_Command(add);  //Set lower addr;
}
//-------------------------------------------------------------------
void Set_row_addr(unsigned char row)
{
        row=row&0x0F;
        row=row|0x0B0;
        Write_Command(row);        //page addr set
}
/***********************************/
/*          LCD Initial Code               */
```

```
/***********************************/
void Lcd_Set()
{
        RESET=0;
        Delay(1000);
        RESET=1;
        Delay(1000);
        Write_Command(0xA2);            //Set Bias          0xA2
        Write_Command(0xA0);            //Segment Direction Select ,bit0=1,reverse;=0,normal;
        Write_Command(0xC8);            //Common Direction Select,bit3=1,reverse
direction;=0,normal;
        Write_Command(0xAC);
        Write_Command(0xA7);            //bit0=0,Normal/bit0=1,Reverse Display
        Write_Command(0xA4);
        Write_Command(0x2C);            //Power Control Set
        Delay(10);
        Write_Command(0x2E);            //Power Control Set
        Delay(10);
        Write_Command(0x2F);            //Power Control Set
        Delay(10);
        Write_Command(0x24);            //set ra/rb 0x25
        Write_Command(0x81);            //Set Contrast
        Write_Command(0x27);            //29
        Write_Command(0xD5);
        Write_Command(0x00);
        Write_Command(0xD2);
        Write_Command(0x00);
        Write_Command(0x60);            //Set Display Start Line      40
        Write_Command(0xb0);
        Write_Command(0x10);
        Write_Command(0x00);
        Write_Command(0xAF);            //Display ON

}

/***********************************/
/*          Clear Display                  */
/***********************************/
void Display_Clear(data1,data2)
{
        int i,j,m;
        m=0xb0;
```

```
        for(i=0;i<9;i++)
            {
                Write_Command(m);
                Write_Command(0x10);
                Write_Command(0x00);
                for(j=0;j<96;j++)
                    {
                        Write_Data(data1);
                        Write_Data(data2);
                     }
                m++;
             }
}
/***********************************/
void show_three_h_bar(int row)
{
        int i,page,row_data;
        page=row/8;
        i=row%8;
        row_data=row_table[i];

        Set_row_addr(page-1);       //清除前一条横线
        Set_column_addr(00);
        for(i=0;i<192;i++)
            {
                Write_Data(0x00);
             }
        Set_row_addr(page);         //清除前一条横线
        Set_column_addr(00);
        for(i=0;i<192;i++)
            {
                Write_Data(0x00);
             }

        Set_row_addr(page);
        Set_column_addr(00);
        for(i=0;i<192;i++)
            {
                Write_Data(row_data);
             }
}
/***********************************/
```

```
/**********************************/
/*          three vertical bar                */
/**********************************/
void show_three_v_bar(int bar)
{
        int i;
        Set_column_addr(bar-1);
        for(i=0;i<9;i++)
             {
                  Set_row_addr(i);
                  Set_column_addr(bar-1);
                  Write_Data(0x00);
              }
        Set_column_addr(bar);
        for(i=0;i<9;i++)
             {
                  Set_row_addr(i);
                  Set_column_addr(bar);
                  Write_Data(0x0ff);
              }
        for(i=0;i<9;i++)
             {
                  Set_row_addr(i);
                  Set_column_addr(bar+1);
                  Write_Data(0x0ff);
              }
        for(i=0;i<9;i++)
             {
                  Set_row_addr(i);
                  Set_column_addr(bar+2);
                  Write_Data(0x0ff);
              }
}
/**********************************/
/**********************************/
/*          one vertical bar                */
/**********************************/
void show_one_v_bar(int bar)
{
        int i;
        Set_column_addr(bar-1);
        for(i=0;i<9;i++)
```

```
            {
                Set_row_addr(i);
                Set_column_addr(bar-1);
                Write_Data(0x00);
             }
        Set_column_addr(bar);
        for(i=0;i<9;i++)
            {
                Set_row_addr(i);
                Set_column_addr(bar);
                Write_Data(0x0ff);
             }
}
/************************************/
/*          one horizontal bar                */
/************************************/
void show_one_h_bar(int row)
{
        int i,page,row_data;
        page=row/8;
        i=row%8;
        row_data=row_table[i];

        Set_row_addr(page-1);       //清除前一条横线
        Set_column_addr(00);
        for(i=0;i<132;i++)
            {
                Write_Data(0x00);
             }
        Set_row_addr(page);         //清除前一条横线
        Set_column_addr(00);
        for(i=0;i<192;i++)
            {
                Write_Data(0x00);
             }

        Set_row_addr(page);
        Set_column_addr(00);
        for(i=0;i<192;i++)
            {
                Write_Data(row_data);
             }
```

```
}
/************************************/
/*          Display Font                        */
/************************************/
void Display_Font(unsigned char page0,seg0,unsigned char *p)
{
        int i,m,n;
        m=((seg0&0xf0)>>4)|0x10;
        n=seg0&0x0f;
        Write_Command(page0);
        Write_Command(m);
        Write_Command(n);
        for(i=0;i<5;i++)
            {
                Write_Data(*p++);
             }
}

/************************************/
/*          Display Hanzi                       */
/************************************/
void Display_Hanzi(unsigned char page0,seg0,unsigned char *p)
{
        int i,m,n;
        m=((seg0&0xf0)>>4)|0x10;
        n=seg0&0x0f;
        Write_Command(page0);
        Write_Command(m);
        Write_Command(n);
        for(i=0;i<16;i++)
            {
                Write_Data(*p++);
             }
        m=((seg0&0xf0)>>4)|0x10;
        n=seg0&0x0f;
        Write_Command(page0+1);
        Write_Command(m);
        Write_Command(n);
        for(i=0;i<16;i++)
            {
                Write_Data(*p++);
             }
```

```
}

/**********************************/
/*          Display Character                */
/**********************************/
void Display_Character(unsigned char page0,seg0,unsigned char *p0)
{
                int step0=10;
                int step1=5;
                Display_Font(page0,seg0,p0);
                seg0=seg0+step0;
                p0=p0+step1;
                Display_Font(page0,seg0,p0);
                seg0=seg0+step0;
                p0=p0+step1;
                Display_Font(page0,seg0,p0);
                seg0=seg0+step0;
                p0=p0+step1;
                Display_Font(page0,seg0,p0);
                seg0=seg0+step0;
                p0=p0+step1;
                Display_Font(page0,seg0,p0);
                seg0=seg0+step0;
                p0=p0+step1;
                Display_Font(page0,seg0,p0);
                seg0=seg0+step0;
                p0=p0+step1;
                Display_Font(page0,seg0,p0);
                seg0=seg0+step0;
                p0=p0+step1;
                Display_Font(page0,seg0,p0);
                seg0=seg0+step0;
                p0=p0+step1;
                Display_Font(page0,seg0,p0);
                seg0=seg0+step0;
                p0=p0+step1;
                Display_Font(page0,seg0,p0);
                seg0=seg0+step0;
                p0=p0+step1;
                Display_Font(page0,seg0,p0);
                seg0=seg0+step0;
                p0=p0+step1;
```

```
Display_Font(page0,seg0,p0);
seg0=seg0+step0;
p0=p0+step1;
Display_Font(page0,seg0,p0);
/*
seg0=seg0+step0;
p0=p0+step1;
Display_Font(page0,seg0,p0);
seg0=seg0+step0;
p0=p0+step1;
Display_Font(page0,seg0,p0);
seg0=seg0+step0;
p0=p0+step1;
Display_Font(page0,seg0,p0);
seg0=seg0+step0;
p0=p0+step1;
Display_Font(page0,seg0,p0);
seg0=seg0+step0;
p0=p0+step1;
Display_Font(page0,seg0,p0);
seg0=seg0+step0;
p0=p0+step1;
Display_Font(page0,seg0,p0);
seg0=seg0+step0;
p0=p0+step1;
Display_Font(page0,seg0,p0);
seg0=seg0+step0;
p0=p0+step1;
Display_Font(page0,seg0,p0);
seg0=seg0+step0;
p0=p0+step1;
Display_Font(page0,seg0,p0);
seg0=seg0+step0;
p0=p0+step1;
Display_Font(page0,seg0,p0);
seg0=seg0+step0;
p0=p0+step1;
Display_Font(page0,seg0,p0);
seg0=seg0+step0;
p0=p0+step1;
Display_Font(page0,seg0,p0);
```

```
                seg0=seg0+step0;
                p0=p0+step1;
                Display_Font(page0,seg0,_00);
                */

}
//---------------------------------------------------------------------
void Show_one_char(unsigned char xs,unsigned char page,unsigned char *p)
{
        int i;
        Set_column_addr(xs);          //column addr set
        Set_row_addr(page);
        for(i=0;i<5;i++)            //5X8 font
            {
                Write_Data(*(p++));
                //Write_Data(*(p++));
             }
}
//---------------------------------------------------------------------

//---------------------------------------------------------------------
void Conversion_T0_char(int number)
{
          Buf_1=number/100;
          conversion=number%100;
          Buf_2=conversion/10;
          Buf_3=conversion%10;
}
//---------------------------------------------------------------------
char *Get_addr(int num)
{
                unsigned char *pt;
                switch(num)
                {
                    case 0: pt=char_0;break;
                    case 1: pt=char_1;break;
                    case 2: pt=char_2;break;
                    case 3: pt=char_3;break;
                    case 4: pt=char_4;break;
                    case 5: pt=char_5;break;
                    case 6: pt=char_6;break;
                    case 7: pt=char_7;break;
```

```
                    case 8: pt=char_8;break;
                    case 9: pt=char_9;break;
                }
                return pt;
}
//-------------------------------------------------------------
void show_three_number(unsigned char x,y,int n)
{
       Conversion_T0_char(n);
       char_point=Get_addr(Buf_1);
       Show_one_char(x,y,char_point);
       char_point=Get_addr(Buf_2);
       x=x+5;
       Show_one_char(x,y,char_point);
       char_point=Get_addr(Buf_3);
       x=x+5;
       Show_one_char(x,y,char_point);
}
//**********************************
/**********************************/
void Wait_Press()
{
     while(KEY_PRESS);
     Delay(0x5000);
}
/**********************************/
/*          Draw a image                        */
/**********************************/
void Display_Image(unsigned char page0,seg0,pagew,segw,unsigned char *p)
{
       int i,j,m,n;
       m=((seg0&0xf0)>>4)|0x10;
       n=seg0&0x0f;
       for(i=0;i<pagew;i++)
            {
                Write_Command(page0);
                Write_Command(m);
                Write_Command(n);
                for(j=0;j<segw;j++)
                    {
                        Write_Data(*(p++));
                     }
```

```
                page0++;
            }
}
/************************************/
//REGULATE CONTRAST
void Regulate_contrast()
{

    while(EXIT_KEY&&01)
    {
        if(KEY_PRESS==0)
        {
            contrast=contrast+1;
            Write_Command(0x81);            //Set Contrast
            Write_Command(contrast);
            Delay(0x5000);
        }
        if(DEC_KEY==0)
        {
            contrast=contrast-1;
            Write_Command(0x81);            //Set Contrast
            Write_Command(contrast);
          Delay(0x5000);
        }
    }
}
/************************************/
/*          Main                    */
/************************************/
void main()
{
      int i;
      int j=0x40;
      contrast=0x0C;
      Lcd_Set();
      while(1)
          {


            Display_Image(0xb0,0x01,8,128,description1);
            Wait_Press();     //Delay(0x5000);
           Display_Image(0xb0,0x01,8,128,description2);
```

```
        Wait_Press();     //Delay(0x5000);
        Display_Image(0xb0,0x01,8,128,description3);
        Wait_Press();     //Delay(0x5000);
      Display_Image(0xb0,0x01,8,128,description4);
        Wait_Press();     //Delay(0x5000);

      Display_Image(0xb0,0x01,8,128,description5);
        Wait_Press();     //Delay(0x5000);


    Display_Clear(0x00,0x00);
    Display_Clear(0x55,0xaa);              //SNOW
    Wait_Press();
    Display_Clear(0x00,0x00);
    Display_Clear(0xFF,0xFF);              //BLACK
    Wait_Press();

    Display_Clear(0x00,0x00);
    Display_Clear(0xff,0x00);              //V_BAR
    Wait_Press();

    Display_Clear(0x00,0x00);
    Display_Clear(0x55,0x55);              //H_BAR
    Wait_Press();

    Display_Clear(0x00,0x00);
    //Display_Clear(0xFF,0xFF);            //BLACK
    for(i=0;i<65;i++)
    {
        show_one_h_bar(i);
        show_three_number(4,3,i);
        Wait_Press();
    }
      }
}

→
```