

doi: 10.7690/bgzd.2013.04.026

显示模块 GU140×32F-7050 的接口设计与应用

孔文艳, 李坚利, 朱卯成

(上海电控研究所综合电子技术研究室, 上海 200092)

摘要: 为实现某控制装置文字和图形的显示, 设计显示模块 GU140×32F-7050 的接口。硬件接口采用 MC9S12DP512 作为 MPU, 通过并行接口与显示模块连接; 软件接口包括数据写入、初始化、显示、窗口定义和窗口选择。根据点阵图获取汉字、数字、字母或图形的图像数据, 以 2 种方式调用显示接口函数得到相同的显示效果, 并给出相应的程序代码。目前该显示模块已批量应用于某控制装置, 实际应用表明, 显示模块 GU140×32F-7050 工作稳定、可靠性高。

关键词: 显示模块; 接口; GU140×32F-7050; MC9S12DP512

中图分类号: TP334.7 **文献标志码:** A

Interface Design and Application of Display Module GU140×32F-7050

Kong Wenyan, Li Jianli, Zhu Maocheng

(Integrated Electronic Technology Laboratory of Shanghai Electronic Control Institute, Shanghai 200092, China)

Abstract: The interface of display module GU140 × 32F-7050 is designed to realize the text and graphics display of certain type control device. It adopts MC9S12DP512 as it's MPU, they connect with display module by parallel interface. The software interface includes writing data, initialization, display, window definition and window selection. The image data of Chinese characters, numbers, letters or graphics is acquired according to the dot matrix diagram. And the same effect is gotten by two ways of calling display interface functions with corresponding code. Now the display module is widely used for some control device. The practical application shows that: the display module GU140 × 32F-7050 is stable and reliable when working.

Key words: display module; interface; GU140×32F-7050; MC9S12DP512

0 引言

GU140×32F-7050 是日本 Noritake Itron 公司生产的一款荧光显示模块, 具有自发光、高亮度、高可视性的特点。分辨率为 140×32, 只需 5 V 供电, 工作温度 -40~85 °C。该显示模块由 140×32 点阵图形 BD-VFD(built-in drive vacuum fluorescent display)、RAM、字形发生器、DC/DC 转换器、显示控制器和控制逻辑等组成, 能够同时显示图形和字符^[1]。笔者应用该显示模块, 设计接口电路和软件接口。

1 硬件接口设计

本接口电路使用 Freescale 公司 HCS12 系列单片机 MC9S12DP512^[2], 具有 512 KB 片内 Flash 和 14 KB 片内数据 RAM, 可供存放执行程序 and 字库。

显示模块提供并行、串行 2 种接口, 本设计采用并行接口(显示模块的 CN1 插口)与 MPU(微处理器)连接。其连接电路如图 1, 将显示模块的接口作为存储器挂接在 MPU 总线上, MPU 以访问存储器的方式对显示模块进行操作^[3]。图中 MC9S12DP512

的 PA 口为数据口, PB0 为选择信号, PB1 为写信号, PB2 为读信号, PJ0 为忙/闲状态信号。此外, 还需将显示模块 JRB 插口的 C 与 B 短接, 将 PBUSY 信号连接到 CN1 插口的 3 号脚。

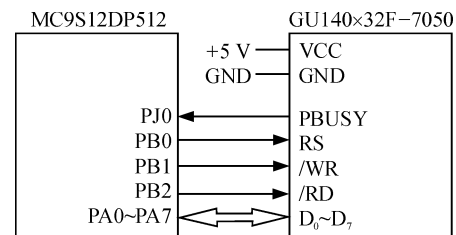


图 1 显示模块接口电路

2 软件接口设计

软件接口包括数据写入、初始化、显示、窗口定义和窗口选择。根据点阵图获取汉字、数字、字母或图形的图像数据。

2.1 接口函数的设计

对显示模块操作需写入不同的数据, 以下是数据写入函数的代码, 先查询显示模块的 PBUSY 信号(1“忙”, 0“不忙”), 遇“忙”等待, “不忙”

收稿日期: 2012-10-27; 修回日期: 2012-11-19

作者简介: 孔文艳(1979—), 女, 江苏人, 工程师, 从事嵌入式计算机系统的研究。

时再按“写”的时序写入数据。

```
void Write_Byte (unsigned char Bdata)
{
    while(PTIJ_PTIJ0==1) /* PBUSY=1, “忙” */
        { ; } /* 等待 */
    PORTB_BIT1=1; /* /WR 置 1 */
    PORTB_BIT1=0; /* /WR 置 0 */
    PORTB_BIT2=1; /* /RD 置 1 */
    PORTB_BIT0=0; /* RS 置 0 */
    PORTB_BIT0=1; /* RS 置 1 */
    PORTA = Bdata; /* 数据 D0~D7 */
    PORTB_BIT1=1; /* /WR 置 1 */
    PORTB_BIT0=0; /* RS 置 0 */
}
```

软件接口程序分为初始化和显示 2 部分。

初始化函数包括清屏和亮度设置，清屏需写入命令数据 0x0C，设置亮度时先写入命令串 0x1F、0x58，紧接着写入亮度数据设置亮度水平（1：亮度 12.5%；2：亮度 25%；3：亮度 37.5%；4：亮度 50%；5：亮度 62.5%；6：亮度 75%；7：亮度 87.5%；8：亮度 100%）。初始化程序为：

```
void Display_Init(unsigned char b)
{
    Write_Byte(0x0C); /* 清屏 */
    Write_Byte(0x1F); /* 亮度设置命令 */
    Write_Byte(0x58); /* 亮度设置命令 */
    Write_Byte(b); /* 设置亮度 */
}
```

显示函数包括指针设置和位图像显示。指针设置时先写入命令串 0x1F、0x24，紧接着写入指针位置。位图像显示时先写入命令串 0x1F、0x28、0x66、0x11，紧接着写入图像尺寸和图像数据。显示函数代码为：

```
void Display(int Cx,int Cy,int Sx,int Sy,unsigned char *Idata)
{
    int i;
    Write_Byte(0x1F); /* 指针设置命令 */
    Write_Byte(0x24); /* 指针设置命令 */
    Write_Byte(Cx%256); /* 设置指针位置 x 低 8 位 */
    Write_Byte(Cx/256); /* 设置指针位置 x 高 8 位 */
    Write_Byte(Cy%256); /* 设置指针位置 y 低 8 位 */
    Write_Byte(Cy/256); /* 设置指针位置 y 高 8 位 */
    Write_Byte(0x1F); /* 位图像显示命令 */
    Write_Byte(0x28); /* 位图像显示命令 */
    Write_Byte(0x66); /* 位图像显示命令 */
```

```
Write_Byte(0x11); /* 位图像显示命令 */
Write_Byte(Sx%256); /* 设置图像尺寸 x 低 8 位 */
Write_Byte(Sx/256); /* 设置图像尺寸 x 高 8 位 */
Write_Byte(Sy%256); /* 设置图像尺寸 y 低 8 位 */
Write_Byte(Sy/256); /* 设置图像尺寸 y 高 8 位 */
Write_Byte(1);
for(i = 0; i < Sx*Sy; i++) {
    Write_Byte(Idata[i]); /* 写入图像数据 */
}
}
```

显示模块具有基础窗口和用户窗口 2 种窗口，最多可定义 4 个用户窗口。定义用户窗口时先写入命令串 0x1F、0x28、0x77、0x02，紧接着写入窗口位置和窗口尺寸。用户窗口定义函数为：

```
void Window_Define(int xP,int yP,int xS,int yS,int a)
{
    Write_Byte(0x1F); /* 用户窗口定义命令 */
    Write_Byte(0x28); /* 用户窗口定义命令 */
    Write_Byte(0x77); /* 用户窗口定义命令 */
    Write_Byte(0x02); /* 用户窗口定义命令 */
    Write_Byte(a); /* 设置用户窗口号 */
    Write_Byte(1); /* 定义用户窗口 */
    Write_Byte(xP%256); /* 设置窗口位置 x 低 8 位 */
    Write_Byte(xP/256); /* 设置窗口位置 x 高 8 位 */
    Write_Byte(yP%256); /* 设置窗口位置 y 低 8 位 */
    Write_Byte(yP/256); /* 设置窗口位置 y 高 8 位 */
    Write_Byte(xS%256); /* 设置窗口尺寸 x 低 8 位 */
    Write_Byte(xS/256); /* 设置窗口尺寸 x 高 8 位 */
    Write_Byte(yS%256); /* 设置窗口尺寸 y 低 8 位 */
    Write_Byte(yS/256); /* 设置窗口尺寸 y 高 8 位 */
}
```

定义了用户窗口后，在对窗口进行操作前必须先选定窗口，选择窗口时先写入命令串 0x1F、0x28、0x77、0x01，紧接着写入窗口号（0：基础窗口；1~4：用户窗口）。窗口选择函数为：

```
void Window_Select(int w)
{
    Write_Byte(0x1F); /* 窗口选择命令 */
    Write_Byte(0x28); /* 窗口选择命令 */
    Write_Byte(0x77); /* 窗口选择命令 */
    Write_Byte(0x01); /* 窗口选择命令 */
    Write_Byte(w); /* 设置窗口号 */
}
```

2.2 图像数据的获取方法

先画出所需显示的汉字、数字、字母或图形等的点阵图，再根据点阵图从左到右按列取数据（“1”、

“0”分别表示显示屏上一个点的亮、暗), 最后把取得的一组数据放入数组。

例 1: 汉字“值”的 16×16 点阵图如图 2。

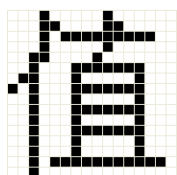


图 2 汉字“值”的点阵图

根据图 2, 取得汉字“值”的图像数据为:

```
ImageData1[32]={0x01,0x00,0x02,0x00,0x0F,
0xFF,0xF8,0x00,0x40,0x02,0x20,0x02,0x27,0xFE,
0x25,0x52,0x2D,0x52,0xF5,0x52,0x65,0x52,0x25,
0x52,0x27,0xFE,0x20,0x02,0x00,0x02,0x00,0x00}。
```

例 2: 数字“3”的 8×16 点阵图如图 3。



图 3 数字“3”的点阵图

根据图 3, 取得数字“3”的图像数据为:

```
ImageData2[16]={0x00,0x00,0x0C,0x18,0x10,
0x04,0x11,0x04,0x11,0x04,0x12,0x88,0x0C,0x70,
0x00,0x00}。
```

例 3: 字母“H”的 8×16 点阵图如图 4。



图 4 字母“H”的点阵图

根据图 4, 取得字母“H”的图像数据为:

```
ImageData3[16]={0x10,0x04,0x1F,0xFC,0x10,
0x84,0x00,0x80,0x00,0x80,0x10,0x84,0x1F,0xFC,
0x10,0x04}。
```

例 4: 自定义图形 16×16 点阵图如图 5。



图 5 自定义图形的点阵图

根据图 5, 取得自定义图形的图像数据为:

```
ImageData4[32]={0x00,0x04,0x00,0x02,0x1C,
0x12,0x22,0x21,0xC1,0xC1,0x00,0x01,0x00,0x02,0x
1C,0x12,0x22,0x22,0xC1,0xC2,0x00,0x01,0x00,0x01,
0x1C,0x11,0x22,0x22,0xC1,0xC2,0x00,0x04}。
```

3 应用

得到所要显示的汉字、数字、字母或图形等的

图像数据后, 只要调用显示接口函数即可在显示屏上显示内容。函数调用程序代码为:

```
Display_Init(4); /*显示模块初始化, 亮度为 50%*/
Window_Define( 90,1,20,2,1);
/*定义用户窗口 1, 位置 x=90,y=1, 大小 20×16 */
Window_Select(0); /*选择基础窗口*/
Display(30,1,16,2,ImageData4);
/*在 x=30, y=1 的位置显示自定义图形*/
Display(50,1,16,2,ImageData1);
/*在 x=50, y=1 的位置显示汉字“值”*/
Window_Select(1); /*选择用户窗口 1*/
Display( 0,0, 8,2,ImageData2);
/*在 x=0, y=1 的位置显示数字“3”*/
Display(10,0, 8,2,ImageData3);
/*在 x=10, y=1 的位置显示字母“H”*/
```

也可以不定义用户窗口, 直接在基础窗口显示, 函数调用程序代码为:

```
Display_Init(4); /*显示模块初始化, 亮度为 50%*/
Display( 30,1,16,2,ImageData4);
/*在 x=30, y=1 的位置显示自定义图形*/
Display( 50,1,16,2,ImageData1);
/*在 x=50, y=1 的位置显示汉字“值”*/
Display( 90,1, 8,2,ImageData2);
/*在 x=90, y=1 的位置显示数字“3”*/
Display(100,1, 8,2,ImageData3);
/*在 x=100, y=1 的位置显示字母“H”*/
```

以上 2 种操作方式的显示结果是一样的, 实际显示效果如图 6 所示。



图 6 显示实图

4 结语

采用显示模块 GU140×32F-7050 作为控制装置的显示界面, 接口电路简单, 软件控制方便, 可移植性强。该显示模块已批量应用于某控制装置, 实际应用结果表明, 显示模块 GU140×32F-7050 工作稳定、可靠性高。

参考文献:

- [1] Vacuum Fluorescent Display Module Specification (Model: GU140X32F-7050). Noritake Itron CORP., 2005.
- [2] 孙同景, 陈桂友. Freescale 9S12 十六位单片机原理及嵌入式开发技术[M]. 北京: 机械工业出版社, 2008: 11.
- [3] 章百宝, 姚毅. OLED 显示屏接口电路的设计[J]. 兵工自动化, 2006, 25(9): 86-92.