

文章编号:1007-2780(2009)06-0901-06

基于 ARM 的图形用户界面软件开发平台的搭建

古容江,张 珣

(杭州电子科技大学 电子信息学院,浙江 杭州 310018,E-mail:yzgurongjiang@163.com)

摘 要: 设计了基于三星 S3C2410 微控制器的 TFT-LCD 驱动控制系统,并将其与 Micrium 公司的 μ C/GUI 图形用户界面整合在一起,形成了一个方便小巧的嵌入式系统应用软件开发平台。详细分析了如何利用 ARM9 的 LCD 控制器进行 LCD 的驱动,并介绍了将该驱动程序移植到图形用户界面 μ C/GUI 的过程。

关 键 词: TFT-LCD 驱动; μ C/GUI 移植; ARM; 图形用户界面
中图分类号: TP311;TN141.9 **文献标识码:** A

1 引 言

现代 IT 技术的不断创新和进步促进了嵌入式系统的应用和发展。现在,利用嵌入式系统生产的产品已经融入了人类生活的各个方面,如手机、MP3、MP4、GPS 和 PDA 等。在一个高端电子系统的开发中,除了需要一个功能强大、工作稳定、接口丰富的硬件开发平台外,还需要一个完善的软件开发平台,这个软件开发平台一般包括各类接口的底层驱动中间件和操作系统^[1]。

本文基于三星公司 S3C2410 硬件平台^[2],将 TFT-LCD 的驱动和图形用户界面 μ C/GUI^[3,4] 设计整合到一起,搭建了一个较为完善的图形用户界面软件开发平台,方便程序员在不考虑硬件的环境下设计自己的视窗应用程序。

2 TFT-LCD 的驱动

2.1 TFT-LCD 的硬件接口电路

设计中采用的液晶显示模块是日本 NEC 公司的 NL2432DR22 TFT-LCD^[5],显示区对角线为 8.8 cm(3.5 in),色彩为 262 144 色,分辨率为 320×240。

图 1 为系统的硬件电路框图,其中主要有 32 位微控制器(内含 LCD 控制器)、TFT-LCD 模块、LED 背光和工作电源模块等。

微控制器选用 S3C2410,内部集成有 LCD 控制器,这使 LCD 的驱动变得非常简便。NL2432

DR22 液晶显示模块 LED 背光的电压指标由专用芯片 MAX1664 提供,工作电压从核心板引入。

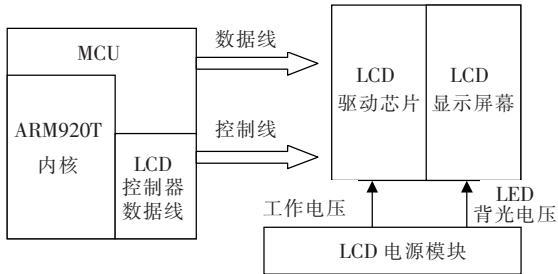


图 1 系统硬件电路框图

Fig. 1 System hardware configuration

2.2 LCD 的驱动控制^[6,7]

S3C2410 的 LCD 控制器由 REG BANK、LCD CDMA、VID PRCS、LPC3600 和 TIME GEN 组成^[2]。REG BANK 用来设置 LCD 控制器;LCD CDMA 是个专用 DMA,用来存放视频数据;VID PRCS 用来将数据转化成合适的数据格式,如 8 位单扫描格式;LPC3600 和 TIME GEN 用来产生 LCD 所需要的各种控制信号的时序。LCD 控制器内部结构如图 2 所示。

其中以 TIME GEN 最为关键,它产生 LCD 驱动器所需要的控制信号,如 VFRAME, VLINE, VCLK, VM 等。实现 LCD 驱动的关键就是要根据 LCD 驱动器的要求,合理配置 LCD 控制寄存器。

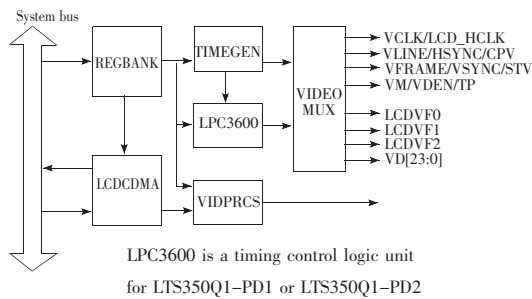


图 2 LCD 控制器内部结构框图
Fig. 2 LCD controller configuration

LCD 驱动器用 VM 信号改变行、列的电压极性来开关像素。VM 频率有两种模式，其中当 LCDCON1 寄存器中的 MMODE 位为 0 时，VM 频率与 Frame Rate 一致；MMODE 位为 1 时，VM 频率与 LCDCON4 寄存器的 MVAL[7:0] 相关，计算公式如下：

$$\text{VM Rate} = \text{VLINE Rate} / (2 \times \text{MVAL}) \quad (1)$$

LCD 的各种控制信号都与时钟信号 VCLK 相关，因此 VCLK 的设置尤为重要，一般 8.8 cm (3.5 in) TFT-LCD 都要求 VCLK 大于 6.5 MHz。VCLK 的设置与 LCDCON1 寄存器中的 CLKVAL 和 HCLK 有关，关系如式(2)所示：

$$\text{VCLK (Hz)} = \text{HCLK} / (2 \times \text{CLKVAL}) \quad (2)$$

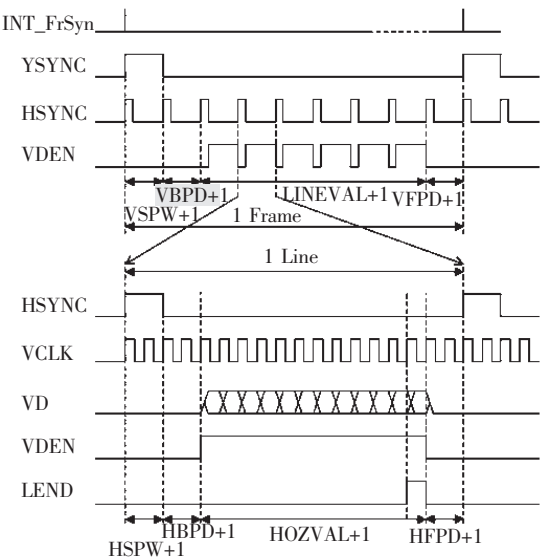


图 3 TFT- LCD 时序图样例
Fig. 3 Example of TFT -LCD timing

行场扫描信号 VFRAME 和 VLINE 比较复杂，它们的时序图如图 3 所示。

从图 3 可以看出，VFRAME (VSYSNC) 和 VLINE (HSYNC) 的信号与 LINEVAL、HOZVAL、VBPD、VFDP、VSPW、HBPDP、HFPDP、HSPW 有关。

LINEVAL 为垂直显示长度字域，与 LCD 的垂直显示尺寸和扫描模式有关：

单扫描模式：
$$\text{LINEVAL} = (\text{垂直显示尺寸}) - 1 \quad (3)$$

双扫描模式：
$$\text{LINEVAL} = (\text{垂直显示尺寸} / 2) - 1 \quad (4)$$

HOZVAL 为水平显示长度字域，与 LCD 的水平显示尺寸和有效 VD 数据线有关：

$$\text{HOZVAL} = (\text{水平显示尺寸} / \text{有效 VD 数据线}) - 1 \quad (5)$$

VBPD 是垂直同步信号的后肩，确定帧同步信号和帧数据传输前的一段延迟时间。

VFDP 是垂直同步信号的前肩，确定帧数据传输完成后到下一帧同步信号到来的一段延迟时间。

VSPW 是垂直同步信号的脉宽，确定帧同步时钟脉冲宽度。

同样的水平同步信号的各个字域有 HBPDP、HFPDP、HSPW，这里不再赘述。这 6 个字域对显示区域的偏移有影响，一般 LCD 的数据手册都会提供这 6 个字域的最小值、典型值和最大值，也可以自己手动调整这些字域值，直到显示符合要求。全部设置好后，根据式(6)可以得出 VFRAM 的信号频率：

$$\begin{aligned} \text{Frame Rate} = & 1 / [\{ (\text{VSPW} + 1) + (\text{VBPD} + 1) + (\text{LINEVAL} + 1) + (\text{VFDP} + 1) \} \times \{ (\text{HSPW} + 1) + (\text{HBPDP} + 1) + (\text{HFPDP} + 1) + (\text{HOZVAL} + 1) \} \times \\ & \{ 2(\text{CLKVAL} + 1) / (\text{HCLK}) \}] \end{aligned} \quad (6)$$

2.3 NL2432DR22 液晶显示模块的驱动代码

根据以上原理，要驱动 NL2432DR22 液晶显示模块，首先要设置 LCD 的相关参数：

```
#define LCD_XSIZE_TFT    (240)           //LCD 的尺寸,NL2432DR22 是 320×240 像素
#define LCD_YSIZE_TFT    (320)
```

```
#define HOZVAL_TFT      (LCD_XSIZE_TFT-1)    //根据式(5)
#define LINEVAL_TFT     (LCD_YSIZE_TFT-1)    //根据式(3)
#define CLKVAL_TFT      (1)                  //根据式(2)
```

若 FCLK 经过 PLL 设置为 200 MHz,且 FCLK:HCLK:PCLK=1:2:4,那么 HCLK 为 100 MHz。根据 NL2432DR22 的数据手册,可以设置以下参数:

```
#define VBPD ((9-1)&0xff)
//查 LCD 的 datasheet
#define VFPD ((9-1)&0xff) //手动微调
```

```
#define VSPW ((9-1)&0xff)
#define HBPD ((18-1)&0x7f)
#define HFPD ((16-1)&0xff)
#define HSPW ((16-1)&0xff)
```

接下来设置 LCD 数据在内存的缓冲区。初始化 LCD 相关的 GPIO,就可以配置 LCD 控制器的各个寄存器如下:

```
rGPGDAT |= (1 << 4); // 点亮 LCD 背光
rLCDCON1=(7<<8)|(MVAL_USED<<7)|(3<<5)|(12<<1)|0;
//设置 vclk=hclk/7*2, 选择场同, mmode=1, 选择 TFT, 16bpp, 禁止 viedio, 16bpp TFT,
ENVID=off
rLCDCON2=(VBPD<<24)|(LINEVAL_TFT<<14)|(VFPD<<6)|(VSPW);
rLCDCON3=(HBPD<<19)|(HOZVAL_TFT<<8)|(HFPD);
rLCDCON4=(MVAL<<8)|(HSPW);
rLCDCON5=(1<<11)|(1<<9)|(1<<8)|(1<<3)|(1<<0);
//FRM5:6:5,HSYNC 和 VSYNC 设置极性反转 ,初始化 TFT
rLCDSADDR1=((FRAMEBUF_DMA_BASE>>22)<<21)|M5D(FRAMEBUF_DMA_BASE>>1); //lcdbank=0x180,lcdbaseu=0x80000,设置上部扫描地址
rLCDSADDR2=M5D((FRAMEBUF_DMA_BASE+(LCD_XSIZE_TFT* LCD_YSIZE_TFT*2))>>1); //设置下部扫描地址
rLCDSADDR3=((((LCD_XSIZE_TFT-LCD_XSIZE_TFT)/1)<<11)|(LCD_XSIZE_TFT/1);
//偏移大小,虚拟屏幕宽,这里与原屏幕一样
rLCDINTMSK|=(3); //屏蔽 LCD 中断
rLPCSEL&=~(7); // 禁止 LPC3600
rTPAL=0;
rLCDCON1 |= 1;
```

3 图形用户界面 μ C/GUI 的移植

μ C/GUI 是 Micrium 公司开发的通用嵌入式用户图形界面软件,它可以给任何使用图形 LCD 的应用程序提供独立于处理器和 LCD 控制器之外的有效的图形用户接口,可以应用于单一任务环境或多任务环境中。

μ C/GUI 的特点如下:

- (1)适用于任何 8 位/16 位/32 位 CPU,可允许用于支持 ANSI C 的任何编译器;
- (2)适用于任何控制器驱动任何 LCD(单色,

- 灰度,或者彩色);
- (3)通过配置宏,可支持任何接口;
- (4)可在 LCD 的任何一点上显示字符和画位图;
- (5)支持虚拟显示,虚拟显示的尺寸比实际显示大;
- (6)与 Micrium 公司的 μ C/OS 操作系统有着完美的兼容性;
- (7)可定制附加的窗口管理,PC 窗口控件,触摸屏鼠标设备支持和内存管理等功能。

3.1 μC/GUI 的结构说明

μC/GUI的代码是用移植性很高的 ANSI C 语言编写的,可以方便地移植到各种嵌入式显示平台上,它的文件结构如表 1 所示。

表 1 μC/GUI 的结构
Table 1 Structure of μC/GUI

目 录	说 明
Config	配置文件,包括 LCD 的配置与 μC/GUI 的定制
Gui\AntiAlias	抗锯齿的支持,使图像更圆滑
Gui\ConvertColor	用于支持彩色 LCD 的程序
Gui\ConvertMono	用于支持黑白 LCD 的程序
Gui\Core	核心文件,主要包括初始化,2D 图形绘制,字库绘制
Gui\Font	字体文件都放在其下
Gui\LCDDriver	LCD 驱动程序的接口
Gui\MemDev	存储器件的支持,主要防止画面闪烁
Gui\Touch	触摸屏以及鼠标的支持
Gui\Widget	视窗控件库,包含按钮,对话框,进度条等控件
Gui\WM	视窗管理器,主要创建,缩放,移动一个窗口

μC/GUI是典型的分层软件,上层的 API 与硬件驱动层隔离开来,十分适合各种平台下的移植工作,只需要修改硬件驱动层就可以方便地使用 μC/GUI 提供的丰富功能,设计自己的用户界面,如图 4 所示。

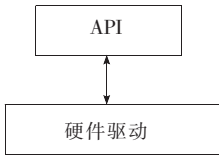


图 4 μC/GUI 的层次结构

Fig. 4 Layer construction of μC/GUI

3.2 μC/GUI 的移植

要移植 μC/GUI,首先需要修改它的硬件驱动层,包括 Config 和 Gui\LCDDriver 下的相关文件。

首先修改 Config 下的 LCDConf. h 文件,设置 LCD 的 Xsize 和 Ysize,这里要与 LCD 的实际尺寸相一致。修改如下:

```
#define LCD_XSIZE (320) //X-resolution of LCD
#define LCD_YSIZE (240) //Y-resolution of LCD
```

tion of LCD

然后修改 Config 下的 GUIConf. h 来定制 μC/GUI 的功能,如下:

```
#define GUI_OS (1) //结合 μC/OS
#define GUI_WINSUPPORT (0) //不使用窗口
#define GUI_SUPPORT_MEMDEV (1) //支持存储,防止闪烁
#define GUI_SUPPORT_TOUCH (0) //没有触摸屏
#define GUI_SUPPORT_UNICODE (1) //支持 UNICODE
```

最后,填充 LCDDriver 下的 LCDSLin. c 文件,完成 μC/GUI 要求的各个接口函数。如下:

```
LCD_L0_SetPixelIndex //画点函数
LCD_L0_GetPixelIndex //捕捉点函数
LCD_L0_XorPixel //重叠反色画点函数
LCD_L0_DrawHLine //水平画线函数
LCD_L0_DrawVLine //垂直画线函数
LCD_L0_FillRect //矩形填充函数
LCD_L0_DrawBitmap //画位图函数
LCD_L0_SetOrg //设置焦点函数
LCD_L0_Init //初始化函数
LCD_On //显示器开
LCD_Off //显示器关
```

下面挑选具有代表性的函数进行分析。首先把第 2 节完成的 LCD 驱动程序写进函数 LCD_L0_Init 中,驱动 LCD。然后写出画点的函数如下:

```
void LCD_L0_SetPixelIndex(int x, int y, int PixelIndex)
{
    if ( ( x < SCR_XSIZE_TFT_240320 ) &&
        ( y < SCR_YSIZE_TFT_240320 ) ) //在显示范围内
        LCD_BUFER[(y)][(x)] = PixelIndex;
    //给 x,y 坐标的点显示颜色
}
```

其他函数的编程思路是:画线函数就是调用画点函数,依次画点;填充函数调用画线函数,依次画线,这样就可以完成 μC/GUI 的接口函数了。

3.3 μC/GUI 的仿真与调试

在 ADS+AXD 的集成环境中进行代码的编

写和调试,仿真器使用了 HJTAG 软件支持的简
易的 JTAG 仿真器。用户调试函数如下:

```
GUI_DrawLine(10,10,200,200,0xff00);  
//画线  
GUI_DrawLine(10,200,200,10,0xff00);  
GUI_FillRect(200,280,240,320); //画  
填充矩形  
GUI_SetDrawMode(GUI_DRAWMODE_
```

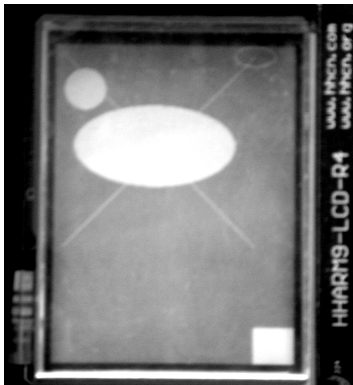


图 5 LCD 仿真效果
Fig. 5 Interactive simulation

```
NORMAL); //显示模式为正常  
GUI_FillCircle(30,44,20); //画填充圆  
GUI_SetDrawMode(GUI_DRAWMODE_  
NORMAL);  
GUI_DrawEllipse(200,10,20,10); //画  
椭圆  
GUI_SetDrawMode(GUI_DRAWMODE_  
NORMAL);  
GUI_FillEllipse(100,100,80,40); //画  
填充椭圆
```

仿真的效果如图 5 所示。

4 结 论

实现了基于三星 S3C2410 微控制器的 TFT-
LCD 驱动控制,并将其与 Micrium 公司的 μ C/
GUI 的图形用户界面整合在一起,形成了一个方
便小巧的嵌入式系统应用软件开发平台,为以后
进行界面程序设计提供了环境支撑,使程序员可
以在并不熟悉硬件工作方式下,专心开发自己的
界面应用程序。

参 考 文 献:

[1] 田泽. 嵌入式系统开发与应用 [M]. 北京:北京航空航天大学出版社,2005:466-477.
[2] Samsung 公司. S3C44B0X date sheet [R]//Samsung 公司技术资料. 韩国:Samsung ,2004.
[3] 拉伯罗斯(美). μ C/ OS-II: 源码公开的实时嵌入式操作系统 [M]. 邵贝贝,译. 北京:中国电力出版社,2001:
186-240.
[4] Micrium 公司. μ C/GUI user manual[R]//Micrium 公司用户手册,US: Micrium,2002:325 - 347.
[5] NEC 公司. NL2432DR22 date sheet [R]//NEC 公司用户手册,Tokyo:NEC,2001.
[6] 谢昭莉,蒋涛,刘亮,等. 基于 ARM 嵌入式系统的 VGA 接口的研究与设计 [J]. 液晶与显示,2007,22(6):761-764.
[7] 纪宁宁,孙灵燕. S3C2440A 驱动 RGB 接口 TFT LCD 的研究 [J]. 液晶与显示,2008,23(1):96-101.

Establishment of GUI Software Development Platform Based on ARM

GU Rong-jiang, ZHANG Xun

(College of Electronics Information, Hangzhou Dianzi University, Hangzhou 310018, China, E-mail: yzgurongjiang@163.com)

Abstract

Driving and controlling of TFT-LCD was realized based on S3C2410 microcontroller. Then, it
was integrated with Micrium's μ C/GUI, graphical user interface, together to form a compact embed-
ded software system. GUI hardware design, GUI and LCD module's work mechanisms were

载体类型	标志代码
磁带(magnetic tape)	MT
磁盘(disk)	DK
光盘(CD-ROM)	CD
联机网络(online)	OL