

基于 ARM 的嵌入式系统 GUI 设计分析及实现

向前, 罗大庸

(中南大学 信息科学与工程学院, 湖南 长沙 410075)

摘要:随着嵌入式产品在多领域的广泛使用, 具有图形界面的嵌入式系统正被越来越广泛地应用, 而许多诸如 $\mu\text{C}/\text{OS}$ 之类的优秀操作系统本身不具备图形界面模块, 需要进行外部移植。文中详细论述了 S3C44B0X 微处理器的 GUI 硬件设计及 LCD 模块的驱动机理, 在 $\mu\text{C}/\text{GUI}$ 的基础上, 着重介绍了 GUI 工作机理, 详细论述了基于 ARM 的 GUI 的移植方法。

关键词: LCD; $\mu\text{C}/\text{GUI}$; 嵌入式图形界面

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2006)12-0218-02

GUI Design and Realization of Embedded System Based on ARM

XIANG Qian, LUO Da-yong

(Information Science and Engineering College, Central South University, Changsha 410075, China)

Abstract: With wide application of many embedded products in many fields, the embedded system with GUI is more and more widely applied. But many perfect systems have no graphical interface module, it must be transplanted. Describes GUI hardware design and LCD module's work mechanism. On the basis of $\mu\text{C}/\text{GUI}$, mainly introduces GUI's work mechanism, particularly discusses GUI's transplantation methods based on ARM.

Key words: LCD; $\mu\text{C}/\text{GUI}$; graphical interfaces for embedded system

0 引言

在嵌入式系统运行时, 动态显示运行数据和各种参数成为嵌入式系统设计中的一个基本要求, 目前在许多应用设计中都使用 LCD 做为信息交互的显示平台。文中重点介绍一种基于三星公司生产的 ARM 微处理器 S3C44B0X 的 GUI 的设计方法。

1 LCD 模块的驱动机理及 GUI 硬件设计

在 S3C44B0X44B0 中内置一个 LCD 控制器。LCD 控制器支持在 LCD 上的单色、4 级、16 级灰度、256 色的显示^[1], 具有很强的通用性。S3C44B0XLCD 控制器的作用是将缓存中数据传送到外部的 LCD 驱动器^[2], 并产生必须的 LCD 控制信号。

(1) LCD 控制器的结构。

LCD 控制器的基本结构如图 1 所示, 它包括 REG-BANK、LCD DMA、VIDPRCS 和 TIMEGEN。REG-BANK 有 18 个控制寄存器, 通过对它们的配置, 可以完成对整个 LCD 控制器的控制。LCD DMA 是 LCD 控制器专用 DMA, 在没有 CPU 介入的情况下完成自动传输帧存储器的数据到 LCD 驱动器进行显示。VIDPRCS 负责从 LCD-

DMA 接受数据, 并在进行适当的数据格式转换之后通过 VD[7:0] 把数据传送到 LCD 控制器。在模块 TIMEGEN 中包含了可编程逻辑器件, 针对不同 LCD 对接口、速率和时钟的不同要求, 可生成不同的 VFRAME, VLINE, VCLK 和 VM 等外部控制信号。

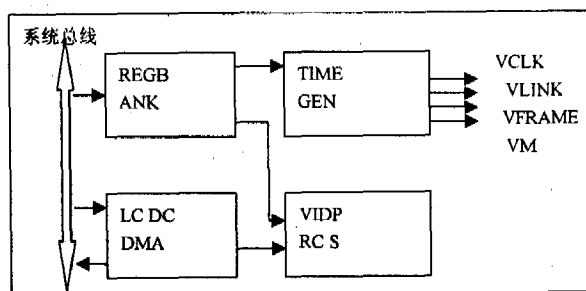


图 1 内置 LCD 控制器结构

VFRAME: LCD 控制器和驱动器之间的帧同步信号, 当控制器完成一帧的显示后通知 LCD 刷屏, 从而使 LCD 的显示指针指向第一个显示位置。

VLINE: 它是 LCD 控制器和驱动器之间的行同步信号。在控制器把一行数据传送到驱动器之后, 控制器发出这个信号, 通知驱动器把数据送到液晶屏显示出来。

VCLK: 它是 LCD 控制器和驱动器之间的像素传送脉冲, 在它的上升沿 LCD 传送数据, 而在下降沿驱动数据。

VM: 它是 LCD 驱动器用来改变行和列的交流电压, 从而决定开关 LCD 上对应像素点的信号。它的值由显示

收稿日期: 2006-04-03

作者简介: 向前(1979-), 男, 湖南岳阳人, 硕士研究生, 研究方向为嵌入式信息系统; 罗大庸, 博士生导师, 研究方向为信息融合和综合自动化系统。

模式、LCDCON1 和 LCDCON2 寄存器里的值设定。

VFRAME 和 VM 脉冲信号的产生是由 LCD 控制寄存器里的水平和垂直参数所控制的。而这两个参数是由 LCD 的尺寸和显示模式所决定的。也就是说不同的 LCD 决定了不同的 VFRAME 和 VM 脉冲信号。

(2) GUI 硬件设计。

LM057QC1TO1 是夏普公司生产的图形点阵 256 色 STN 液晶模块,它的分辨率为 320×240 。S3C44B0X 微处理器外围扩展 LM057QC1TO1 液晶模块的电路原理如图 2 所示。

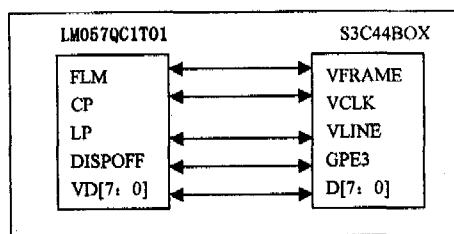


图 2 S3C44B0X 与 LM057QC1TO1 接口

2 μ c/GUI 的移植

2.1 数据结构

μ c/GUI 是 μ c/OS 开发小组开发的图形界面系统,工作机制和 WINDOWS 窗口图形界面相似,可以很方便地移植到目标板。以窗口作为内存管理的基本单位,对所有窗口的管理是通过定义一个堆结构实现的。每个窗口在创建时根据其结构类型的不同为其在堆空间里分配特定大小的连续内存块,并用一个块结构体数组中的一个元素标识^[3],块结构体的定义如下:

```
typedef struct
```

```
{
```

```
tALLOCINT Off; /* 在堆中块开始的位置 */
```

```
tALLOCINT Size; /* 在堆中所占连续内存块的大小 */
```

```
HANDLE Next; /* 指向后一窗口指针 */
```

```
HANDLE Prev; /* 指向前一窗口指针 */
```

```
}tBlock;
```

```
static tBlock aBlock [GUI_MAXBLOCKS];
```

2.2 工作机制

窗口的创建、显示及删除都离不开消息机制和回调函数。在 μ c/GUI 中定义了数十种用于各类基本操作的消息宏^[4],对一个窗口的操作基本上都是通过向其对应的回调函数传送消息参数完成的。

2.3 移植配置

2.3.1 底层配置

在目标板移植使用 μ c/GUI 之前,需要对其进行初始化配置,亦称底层配置。 μ c/GUI 在 LCDCONF.H 文件中定义了宏^[3],主要包括 LCD 宏和 LCD 控制宏。

LCD 宏定义了 LCD 的显示尺寸和可选功能,修改 LCDConf.h 的内容如下:

```
/* LCDConf.h */
```

```
#ifndef LCDCONF_H
```

```
#define LCDCONF_H
```

```
#define LCDG4 //lcd 颜色数,必须定义,LCDMONO(单色),  
LCDG4(四级灰度),LCDG16(16 级灰度)
```

```
#define LCD_XSIZE (320) /* LCD 水平分辨率 */
```

```
#define LCD_YSIZE (240) /* LCD 垂直分辨率 */
```

```
#define LCD_BITSPERPIXEL (8)
```

```
#endif /* LCDCONF_H */
```

LCD 控制宏定义了如何对 LCD 控制器进行访问,修改控制器的配置如下:

```
/* 以下是 S3C44B0X LCD 控制器的配置 */
```

```
#define SCR_XSIZE (320) //视窗屏幕大小
```

```
#define SCR_YSIZE (240)
```

```
#define LCD_XSIZE (320) //液晶屏幕大小
```

```
#define LCD_YSIZE (240)
```

```
#define MSD(n) ((n) & 0x1ffff)
```

```
#define ARRAY_SIZE_G4 (SCR_XSIZE/4 * SCR_YSIZE)
```

```
#define HOZVAL (LCD_XSIZE/4 - 1)
```

```
#define HOZVAL_COLOR (LCD_XSIZE * 3/8 - 1)
```

```
#define LINEVAL (LCD_YSIZE - 1)
```

```
#define MVAL (13)
```

```
#define CLKVAL_G4 (10)
```

```
#define MVAL_USED 0
```

```
#endif /* LCDCONF_H */
```

2.3.2 驱动程序

ARM(S3C44B0X)内置 LCD 控制器^[5],相关驱动写在 lcd44b0.c 文件中,主要进行相关的寄存器配置以及和 GUI 的接口程序。

1)定义显示缓冲区时使用的 char 数据类型,它是 8bit 的:

```
unsigned char Bmp[ARRAY_SIZE_G16]; //液晶显示缓冲数组
```

2)定义读写缓冲区时使用的数据类型,也是 8bit 的 U8:

```
#define LCD_READ_MEM(Off) * ((U8 *) (Bmp + (((U32) (Off))))))
```

```
#define LCD_WRITE_MEM(Off, data) * ((U8 *) (Bmp + (((U32) (Off)))))) = data
```

//#define LCD_READ_REG(Off) //这个函数可以不用定义,这里没有用到

#define LCD_WRITE_REG(Off, data) //有些地方用到了,定义为空,避免做大的改动

3)定义液晶总线宽度定义位 8bit 的:

```
#ifndef LCD_BUSWIDTH
```

```
#define LCD_BUSWIDTH (8)
```

```
#endif
```

4)定义字节顺序:

```
#define LCD_SWAP_BYTE_ORDER (0) //8bit 时不需要交换
```

2.3.3 应用程序

通过对 API 函数的调用即可在 MAIN()中实现应用

(下转第 229 页)

判断,实现了系统信息的安全性。

2 两种方法的比较

(1)两种方法都实现了用户角色权限的验证,保证了系统信息的安全性。

(2)第一种方法没有将模式和视图分开,随着系统的增大,将会导致系统更新的复杂化,以及系统的可扩展性较差。

(3)第二种方法将逻辑和视图分开。通过对 Action 的覆写,实际上实现了对 JSP 页面的定义,将单个的 JSP 页面映射称为一个权限,从而通过对应的权限实现了对相应页面的访问。这样提高了系统的可扩展性。对于新增的 JSP 页面只要新增对应的 Action 以及系统权限,即可保证系统权限验证的完整性。

(4)第二种方法通过禁止对“*.jsp”页面的直接请求,隐藏了系统的文件信息,即隐藏了系统的文件结构,无形中也提高了系统安全性。

3 结论

在 B/S 结构中,系统的权限验证比 C/S 中更显重要,C/S 结构由于具有特定的客户端,因此用户的权限验证可

以通过客户端实现或通过客户端+服务器检测实现,而在 B/S 结构中,用户通过浏览器进行系统信息的访问,如果没有一个全面的权限验证,那么“非法用户”很可能通过浏览器访问到 B/S 结构系统中非授权功能。因此 B/S 结构中用户角色的权限验证将是保证系统信息安全性的一个核心部分。文中构建的基于 Struts 框架的 Web 系统从各方面对用户提交的请求进行验证,防止了用户提交不合法的参数而获取非法信息的可能性,并隐藏了系统的文件结构,最大程度上保证了系统信息的安全性。

参考文献:

- [1] 张桂元,贾燕枫. Struts 开发入门与项目实践[M]. 北京:人民邮电出版社,2005:2-8.
- [2] 暴志刚,胡艳军,顾新建. 基于 Web 的系统权限管理实现方法[J]. 计算机工程,2006,32(1):169-170.
- [3] 张祖平,王 磊. 基于多种模式的权限控制技术研究[J]. 计算机工程,2006,32(1):177-178.
- [4] 周 杰. 对 Struts 应用开发框架的研究和改进[J]. 计算机工程,2004,30(增刊):144-145.
- [5] 孙卫琴. 基于 MVC 的 JavaWeb 设计与开发[M]. 北京:电子工业出版社,2004:5-10.

(上接第 219 页)

程序的编写,有这方面编程经验者可参考 $\mu\text{C}/\text{GUI}$ 用户手册 API 部分。

3 总结

以上详细论述了 S3C44B0X 微处理器的 GUI 硬件设计及 LCD 模块的驱动机理,在 $\mu\text{C}/\text{GUI}$ 的基础上,着重介绍了 GUI 工作机理,详细论述了基于 ARM 的 GUI 的移植方法。其中很多部分仅是提出了概要的描述,实际工作中还有很多细节工作需要完善。

(上接第 226 页)

这里添加了 key_refresh 字段,表示加密的粒度为 $2^{\text{key_refresh}}$ 个字节。协商了加密的粒度之后,现在对原来的密钥导出函数进行修改。密钥分组的计算过程如下:

$\text{key_block} = \text{PRF}(\text{master_secret}, \text{"key expansion"}, \text{server_random} + \text{client_random} + \text{Seq_Num})$

其中 Seq_Num 取值依次为 $0, 2^{\text{key_refresh}}, 2 \times 2^{\text{key_refresh}}, 3 \times 2^{\text{key_refresh}}$ 。协议实现中记录已传输的字节数,当传输的数据达到 $2^{\text{key_refresh}}$ 个字节的时候,SSL 连接两端重新计算密钥分组。

3 结束语

通过对 SSL 协议本身安全性的分析和几种针对 SSL 协议攻击手段的讨论,发现虽然上述攻击手段可以通过协议实现中进行特殊处理而避免,但是从安全协议设计的角

参考文献:

- [1] 王田苗. 嵌入式系统设计与实例开发[M]. 北京:清华大学出版社,2002:123-126.
- [2] Samsung 公司. S3C44B0X data sheet. Samsung 公司技术资料[M]. 韩国:Samsung,2004.
- [3] Micrium 公司. $\mu\text{C}/\text{GUI}$ user manual. Micrium 公司用户手册[M]. [s.l.]:Micrium,2002:325-347.
- [4] 陈冰峰,姜 卓,王剑钢. 基于 ARM7 S3C44B0X 嵌入式系统 GUI 设计[J]. 仪器仪表用户,2004(4):30-31.
- [5] 杜春雷. ARM 体系结构与编程[M]. 北京:清华大学出版社,2001:377-379.

度看,协议自身应当避免安全隐患,而不要依赖于协议实现者的谨慎。

参考文献:

- [1] Frier A, Karlton P, Kocher P. The SSL 3.0 Protocol[EB/OL]. Netscape Communications Corp. 1996-11. <http://wp.netscape.com/eng/ssl3/ssl-toc.html>.
- [2] 戴英侠. SSL 协议的安全缺陷与改进[J]. 中国科学院研究生院学报,2000,17(1):86-92.
- [3] Wagner D, Schneier B. Analyse of the SSL 3.0 Protocol[C]// Proceedings of the 2nd Usenix workshop on electronic commerce. [s.l.]:USENIX Press, 1996:29-40.
- [4] Rescorla E. SSL 与 TLS[M]. 崔 凯译. 北京:中国电力出版社,2002.
- [5] 孙红林. 传输层安全协议的安全性分析及改进[J]. 软件学报,2003,14(3):518-523.