文章编号:1008-0570(2008)09-2-0027-03

基于 ARMlinux 平台的 QT/E 键盘实现

An Implementation of Keyboard Based on QT/E on ARMlinux

(1.西华大学;2.成都纺织高等专科学校) 李军民[1] 祝红军[2]
LI Jun-min ZHU Hong-jun

摘要: 目前,随着仪器仪表要求日益增长,对用户图形界面(GUI)的需求越来越高。面向嵌入式系统的 Qt/E 被广泛应用于其中,而底层设备的实现成为我们设计系统最关键的一步。本文深入分析 QT/E 事件驱动原理,并在 ARMlinux 平台下成功实现了自行设计键盘的输入,详细介绍了软件设计,特别阐述了基于 QT/E 事件驱动原理的键盘驱动设计要点。最后进行了实验,结果表明方案可行、实用。

关键词: ARMlinux; Qt/E; Qtopia 键盘驱动
中图分类号: TP393　　**文献标识码**: A

Abstract: With the demand increase of instrument and meter,GUI is required more and more hightly. Qt/E which is oriented embedded system is applied in instrument and meter.However,the implement of bottom equipment is the key step in system design.This paper analyses event driven principel of Qt/E deeply and realizes input of self-design keybord.Meanwhile, this paper introduces software design of the system in detail,and especially describes the points of keyboard driver design based on event driven principel of Qt/E.At last, the testing result shows the scheme is reliable and practical.

Key words: ARMlinux; Qt/E; Qtopia keyboard driver

技术创新

前言

Qt/E 是著名的 Qt 库开发商 Tmlhech 公司开发的面向嵌入式系统的 Qt 版本。Qt/E 是 Server/Client 结构,延续了 Qt 在 X 上的强大功能,在底层摒弃了 XIib,仅采用帧缓冲作为底层图形接口。同时,将外部输入设备抽象为 keyboard 和 mouse 输入事件,底层接口支持键盘、GPM 鼠标、触摸屏以及用户自定义的设备输入设备是各种嵌入式系统的重要组成部分,如鼠标、小键盘、触摸屏,要让 QT/E 支持特定的嵌入式输入设备,需要对设备驱动和 QT/E 的事件驱动非常熟悉,有很多的资料已经介绍了 QT/E 的事件驱动的相关知识,但都没有给出详细的分析和具体的实现代码。本文主要以自己设计的键盘出发深入分析了 QT/E 的事件驱动的原理和及其实现过程,给出了详细的源代码分析,对基于嵌入式 QT 的工程开发一定的参考价值。

1 Qt/E 的实现结构

Qt/E 巧妙地利用了 C++独有的机制,如继承、多态、模板等,具体实现非常灵活。但其底层代码由于追求与多种系统、多种硬件的兼容,代码补丁较多,风格稍显混乱 Qt/E 的实现结构如图 1:

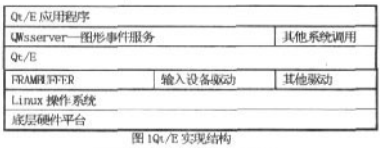

图 1Qt/E 实现结构
Fig1 Realization structure of Qt/E

李军民: 讲师 硕士
基金项目:四川省教育厅青年基金;项目名称:基于网络的机器设备故障智能诊断监测系统研究(2006B043);
四川省科技厅重点攻关项目;项目名称:数控机床运动精度动态测试技术及装置研究(0032785)

2 Qt/E 的事件驱动

Qt/E 中与用户输入事件相关的信号,是建立在对底层输入设备的接口调用之上的。Qt/E 中的输入设备,分为鼠标类与键盘类。以 3.x 版本系列为例,其中鼠标设备的抽象基类为 QWSMouse Handler,键盘设备的抽象基类为 QWSKeyboaord Handler,从该类又重新派生出一些具体的实现类。与图形发生引擎加载方式类似的,在系统加载构造 QWSServer 时,调用 QWSServer::openMouse 与 QWSServer::openKeyboard 函数。这时 qwindowsystem_qws.cpp 里有个 getenv("QWS_MOUSE_PROTO") 和 getenv("QWS_KEYBOARD")这两个函数获取环境变量,它会根据 Linux 系统的环境变量 QWS_MOUSE_PROTO 与 QWS_KEYBOARD 获得鼠标类设备和键盘类设备的设备类型和设备节点。打开相应设备并返回相应设备的基类句柄指针给系统,获得对具体鼠标类设备和键盘类设备的调用操作。图 2 详细介绍了键盘的事件驱动流程。

图 2 事件驱动流程
Fig2 Event driven flow

3 基于 s3c2410 下的键盘驱动程序设计

我们设计的是 4×4 的矩阵键盘，采用的 ZLG7289，当有键按下时产生一个中断，然后通过串行传送数据，下面主要针对 QT/E 的事件驱动原理介绍特定的键盘驱动程序的结构，本系统增加了 poll 阻塞函数，具体结构如下：

```
struct file_operations KBC_INT_ops ={
open:    KBC_INT_open,
read:    KBC_INT_read,
write:   KBC_INT_write,
poll:    skb_poll,
release: KBC_INT_release,
};
```

3.1 模块的插入和卸载

本设计的驱动程序是以模块的形式表现出来的，模块在加载的时候注册设备操作句柄，如下：

```
int init_module()
{s3c2410_KBC_INT_init();}
```

当模块从当前系统注销的时候则调用下面的函数，其作用是释放键盘驱动程序占用的中断号，注销在 kernel 中占用的资源,以及在/dev 下设备不可见。

当使用 insmod 命令的时候系统自动调用 init_module( ) 函数来注册设备，

当使用 rmmod 的时候系统自动调用 cleanup_module()函数来释放设备。

3.2 中断函数和读操作函数的实现

当完成系统的初始化操作以后，系统等待按键中断的到来，一旦时间发生，系统自动调用当前中断处理函数 KBC_INT_interrupt( )。其实现如下：

```
static void KBC_INT_interrupt (int nr, void *devid, struct pt_regs *regs)
{disable_irq(IRQ_KBC_INT);//关中断
{读数据}
wake_up_interruptible(&KBC_wq);//唤醒阻塞程序
enable_irq(IRQ_KBC_INT);//开中断
}
```

而 KBC_INT_read( ) 函数主要调用 copy_to_user(buf, &data, sizeof(data))是把数据从内核拷贝到应用层。

3.3 skb_poll 函数实现

skb_poll 函数是根据 QT/E 事件驱动原理设计的，poll 结构在 kernel 中是通过来 poll_wait(filp, &KBC_wq, wait)来达到阻塞的目的，当没按键时，上层自动进入阻塞状态，当有按键时，在中断服务程序里把全局变量 havedata 置 1 并唤醒等待进程，进程被唤醒后 sys_poll->do_poll 会再一次的调用驱动程序的 poll 函数，返回 POLLIN | POLLRDNORM，就触发信号与槽。其键盘的响应及其在 QT/E 中实现过程如图 3。

```
static unsigned int skb_poll(struct file *filp, poll_table *wait)
{ poll_wait(filp, &KBC_wq, wait);
if(havedata==1)
{ return POLLIN | POLLRDNORM; }
return 0;
}
```

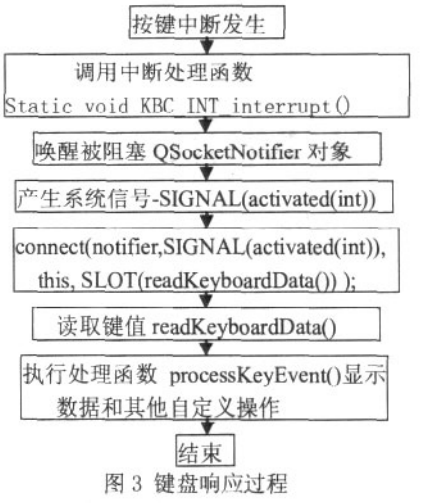

图 3 键盘响应过程

Fig3 Keyboard respond process

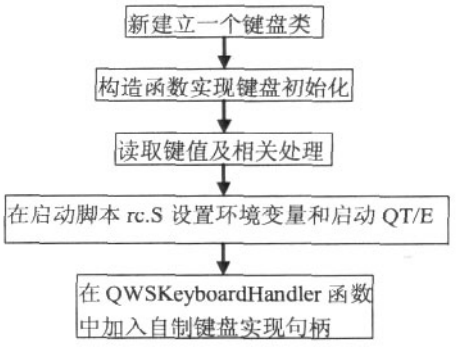

图 4 程序实现流程

Fig3 Program implement flow

4 程序实现流程及源代码

前面分析了 QT/E 的事件驱动原理及键盘响应过程，其程序实现流程如图 4。

下面主要介绍程序通过我们查看/qt-2.3.7-emb/src/kernel /qkeyboard_qws.cpp 中对 USB, TTY 等键盘的处理可以得出，键盘驱动程序是某个键盘上层接口的继承。所以在本驱动程序以其参考自我实现了一个基于 QWSPC101KeyboardHandler 类的接口类。代码如下：

```
class QWSSKBKeyboardHandler : public QWSPC101KeyboardHandler
{
Q_OBJECT
public:
QWSSKBKeyboardHandler(const QString& device);
virtual ~QWSSKBKeyboardHandler();
private slots:
void readKeyboardData();
private:
int fd;
};
```

当我们上层在设置完启动脚本以后，当系统使用 qpe 启动

Qtopia 时，系统自动调用 QWSSKBKeyboardHandler 构造函数来初始化 qte 的键盘接口。

```
QWSSKBKeyboardHandler::QWSSKBKeyboardHandler (const QString& device)
{ fd = open(device.isEmpty()?"/dev/KBC_INT":device.latin1(),
O_RDONLY|O_NONBLOCK, 0);
qDebug("fd = %d\n",fd);
if ( fd >= 0 ) {
QSocketNotifier *notifier;
notifier = new QSocketNotifier( fd, QSocketNotifier::Read, this );
connect( notifier, SIGNAL(activated(int)),this,SLOT(readKeyboardData()));
}
```

通过 QWSSKBKeyboardHandler 实现可以看到，完成设备的打开以后，系统会注册一个信号槽。信号的发送者是 QSocketNotifier，QSocketNotifier 类是基于阻塞模式实现的。它的实现原理是在初始化过程中调用 poll 函数来达到阻塞，通过前面对 poll 的分析可以看到，唤醒 poll 等待队列的时机是当中断处理程序将当前数据读取到缓冲中之后，系统会自动将当前阻塞队列唤醒，接收此信号的处理函数为 readKeyboardData()，此函数用来实现从设备上读取扫描码，通过唯一的扫描码来调用 processKeyEvent()通知 Qt/E 上层处理接口，processKeyEvent()根据传递的参数来通知上层函数所需要显示的数据或需要触发的功能键。当按下某个键以后，我们则可以在基于 QT/E 的应用程序或 Qtopia 中看到输入的数据。程序实现如下：

```
void QWSSKBKeyboardHandler::readKeyboardData()
{ unsigned char portdata[2];
int n = read(fd, &portdata, sizeof(portdata) );
{ 根据 portdata 的值调用 processKeyEvent()进行显示 }
}
```

为了让系统正常运行，建立启动脚本/etc/rc.S 尤为关键，其设置为：

```
insmod /s_kbc/s_kbc.o
export QTDIR=/Qtopia
export QPEDIR=/Qtopia
export LD_LIBRARY_PATH=$QTDIR/lib:$QPEDIR/lib:/usr/lib
export QWS_KEYBOARD="SKB:/dev/KBC_INT"
export PATH=$QPEDIR/bin:$PATH
qpe - qws&
```

5 测试及结果

本设计 CPU 为 S3C2410,bootload 为 VIVI,操作系统为嵌入式 ARMlinux，由于编译后的 QT/E 和 Qtopia 的文件系统较大，直接烧写在 FLASH 中，很浪费时间，通过制作 nfs 根文件系统来调试非常方便，把文件系统放在宿主机上，首先建立 NFS 的调试环境，在内核设置中在 File Systems 里选中“NFS file system support”和“Root File system on NFS”，bootload 设置，本调试系统宿主机 ip 为 192.168.2.122，目标板 ip 为 192.168.2.120，具体设置如下：

```
#vivi>param set linux_cmd_line "noinitrd root=/dev/nfs nfsroot=192.168.2.122:/nfs/root ip=192.168.2.120 init=/sbin/init console=ttyS0"
```

在本系统设计硬件平台基础上，按键反应快、稳定、效果好，已成功应用在嵌入式数控精度测试装置上。

6 结论

本文作者创新点是通过深入分析 QT/E 的事件驱动原理，把 QT 的驱动原理和 Linux 驱动程序设计相结合，通过 poll_wait 阻塞函数成功在 QT/E 实现自己设计的键盘，用同样的方法我们实现了 PS2 键盘、USB 键盘、USB 鼠标、触摸屏，并应用在实际工程中，对 QT/E 的输入设备的实现有一定的应用价值和参考价值。

参考文献

[1]张方辉，王建群. Qt/E 在嵌入式 Linux 上的移植[J]. 计算机技术与发展，2006.7:64-65.

[2]Alessandro Rubini&Jonathan Corbet.Linux 设备驱动程序[M]. O'Reilly Media，Inc.2002.

[3]罗贤全，尚朝轩，高勤.用 GTK 开发 Linux 嵌入式 GUI 应用程序[J].微计算机信息，2004. 6:67-68

作者简介：李军民（1975-），男（汉），四川营山人，西华大学讲师，硕士，主要从事嵌入式系统、网络测控研究。

Biography:LI Jun-min (1975-),Male (the Han nationality), Sichuan Yingshan,Xihua university,Master, Mainly undertaking the research of embedded system and network measure-control.

(610039 西华大学数学与计算机学院) 李军民

(610039 成都纺织高等专科学校教务处 成都) 祝红军

(School of Mathematical&Computer Sicnce XiHua University 610039) LI Jun-min

(Chengdu Textile Institute Educational Administration Department Chengdu 610039.) ZHU Hong-jun

通讯地址：(610039 成都 西华大学研究生部)李军民

(收稿日期:2008.07.23)(修稿日期:2008.09.05)

(上接第 41 页)

[6]MAX7219 使用手册

作者简介：蒋国峰（1975-），男（汉族），河南睢县人，空军第一航空学院航空电子工程系讲师，硕士，主要从事综合航空电子系统的教学和机载通信设备测控系统的研制。

Biography: JIANG Guo-feng (1975-), Male (Han nationality), Henan Province, Department of Aeronautic Electronic Engineering, The First Aeronautic College of Airforce, Lecturer, Master, Unified aircraft electronic system instruction and airborne communication equipment MACSYM development.

(464000 空军第一航空学院航空电子工程系通信教研室 河南信阳)蒋国峰

(Department of Aeronautic Electronic Engineering, The First Aeronautic College of Airforce, Henan 464000, China)

JIANG Guo-feng

通讯地址：(464000 空军第一航空学院航空电子工程系通信教研室 河南信阳)蒋国峰

(收稿日期:2008.07.23)(修稿日期:2008.09.05)