

嵌入式 Linux 和 Qtopia 下矩阵键盘输入的实现^①

李 翠 屈玉贵 (中国科学技术大学 电子工程与信息科学系 安徽 合肥 230027)

摘 要: 详细介绍了基于嵌入式 Linux 和 Qtopia 的矩阵键盘输入实现方法。通过重载 Qt 的事件过滤器, 结合信号和槽机制, 利用两级状态处理及有效输入的界定, 来简化输入过程, 减少按键次数, 提高输入效率。同时利用 Qtopia 的图形显示, 使输入界面清晰, 更适用于嵌入式终端设备。

关键字: 嵌入式终端; 矩阵键盘; Qtopia; 事件过滤器; 信号与槽机制

Implementation of Matrix Keyboard Input in Embedded Linux and Qtopia

LI Cui, QU Yu-Gui (Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China)

Abstract: This paper introduces an input method in detail based on embedded Linux and Qtopia matrix keyboard. By reimplementing the Qt event filter, combined with signal and slot mechanism to deal with the use of two-tier status, it is possible to use the definition of the effective input to simplify the input process, reduce the number of pressing button, and enhance the efficiency of input. At the same time, with the use of Qtopia graphical display, the input interface is clearer and more applicable to the embedded device.

Key words: embedded device; matrix keyboard; Qtopia; event filter; signal and slot mechanism

随着电子技术的不断发展, 嵌入式系统的应用越来越广泛, 对其人机交互界面的要求也越来越高。Qtopia 是 Trolltech 公司为采用嵌入式 Linux 操作系统的电子设备开发的应用平台^[1], Qtopia 包含完整的应用层、灵活的用户界面、窗口操作系统、应用程序启动程序以及开发框架, 可根据实际需要定制编译, 易于增加新的插件, 在嵌入式设备中被广泛应用, Qtopia 的实现结构如图 1 所示。对于嵌入式系统, 键盘是主要的输入设备。针对键盘输入, 在文献[2]中实现的键盘驱动为我们实现 Qtopia 下的输入提供了思路, 但它在文本输入时状态切换频繁, 而且可输入符号不全、界面不清晰。本文开发基于 S3C2410 开发板的矩阵键盘输入法来实现 Qtopia 界面的操作与中英文以及数字符号的输入。

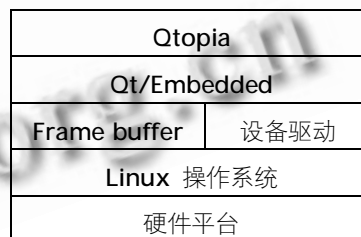


图 1 Qtopia 结构 1

1 键盘的硬件电路及驱动

1.1 硬件平台

基于 S3C2410 开发板的矩阵键盘硬件电路由 4 根行线和 4 根列线组成, 使用 S3C2410 的 GPIO 作为键盘的行线和列线, 共占用 8 根信号线。当键盘按下时, 利用行线导通后电压的变化来产生中断, 以此

① 基金项目: 安徽高校省级自然科学基金重点项目计划(JK2009A13, JK2009A25)

收稿时间: 2010-01-13; 收到修改稿时间: 2010-05-04

来通知按键事件的发生，进而再通过列线的电平设置来判断具体是哪个按键按下。

1.2 键盘的驱动

使用 Linux 输入子系统来实现输入设备的驱动，输入子系统由驱动层，输入子系统核心层和事件处理层三部份组成^[3]。事件处理层对键盘类型的事件已有相应的处理函数，我们只需在驱动层注册相应的设备，并报告设备的事件类型。驱动层通过调用 `Input_register_device` 函数和 `Input_unregister_device` 函数来向输入子系统中注册和注销输入设备。`kbd.dev.evbit[0] = BIT(EV_KEY)`，设置 `evbit` 字段用来定义该输入设备可以支持的事件类型为按键型。

驱动采用中断处理的方法，在中断处理函数中，首先判断出按键所在的行和列，并将对应位置按键的键值上报给核心层，交由核心层将此次按键事件进行处理，`input_report_key(&kbd.dev, button->keycode, 0);input_sync(&kbd.dev);`

定义键盘对应的键值如下：`static int s3c2410kbd_buttons[16] = {KEY_A, KEY_D, KEY_G, KEY_J, KEY_M, KEY_P, KEY_T, KEY_W, KEY_LEFTCTRL, KEY_UP, KEY_DOWN, KEY_BACKSPACE, KEY_CAPSLOCK, KEY_LEFT, KEY_RIGHT, KEY_ENTER};`

1.3 键盘与 Qt 应用程序通信机制

Qt/Embedded 作为服务器，将初始化键盘设备，服务器打开输入设备后，将设备文件描述符与 `socket` 连接起来，然后通过 `socket` 接收输入设备的数据，将其发送给客户应用程序^[1]。Qt/Embedded 中对于键盘响应的实现函数位于 `src/kernel/qkeyboard_qws.cpp` 中。经过上述步骤，开发板按键按下后，其键码就会发送到 Qtopia 的有效输入法插件中进行处理，进而再发送至 Qtopia 应用程序中。下面的工作就是要在 Qtopia 的环境中实现输入法。

2 键盘的整体功能规划

根据嵌入式设备的应用需求和文本输入的特点，一般情况下数字和符号均会出现在中英文状态文本的输入过程中，所以我们将数字符号的输入状态置于中英文的输入状态下，这种组合可以减少状态的切换，例如在中文状态下的输入过程中，在需要输入数字的情况下，切换符号/数字键到数字状态，此时键盘将发

送按键所代表的数字键值，在数字输入结束后，再将二级状态切换回符号状态，便可继续进行中文输入，此时是一级状态处于中文状态，二级状态处于符号状态，既可以输入汉字，又可以输入符号，不需要状态的切换。由此我们将状态分为两级：一级状态：中文、英文大写、英文小写状态；二级状态：数字、符号、无，状态组合的图表表示如表 1。

表 1 状态组合

中文			英文大写			英文小写		
数	符	无	数	符	无	数	符	无
字	号		字	号		字	号	

最终规划利用 16 个按键来输入中英文、33 个符号、以及 10 个数字，因此，将 26 个字母分布在 8 个按键上，一个按键可根据连接情况来决定输入第几个字母，同时为了输入中文，英文大写，英文小写，需要一个中英文切换按键；为了输入数字、符号，需要一个输入数字、符号的状态切换键；对于汉字的选择，以及数字、符号的选择我们需要用到上下左右键；回车与删除键也是必不可少的，对于 Qtopia 应用程序的操作，还需要一个 ESC 键和 Tab 键。键盘布局如图 2 所示。其中：键（中/英大/英小）用来切换输入的状态，中文或英文大写，英文小写；键（符号/数字）用来切换到数字输入状态，或符号图标模式下的输入状态；回车键作为输入确认键和 Qtopia 中应用程序的确认键、换行键。对于“左/ESC”键和“右/Tab”键，做如下规定：在英文大写状态下，分别是“ESC”键值和“Tab”键值有效；其他状态则分别是“左”键值和“右”键值有效。特别说明的一点是，由于符号字符数量多，不方便直接将其置于按键上，所以这里通过虚拟键的方式来选择输入，下面的部分将详细讲述。通过上述按键的布置，实现基本的 Qtopia 应用程序操作和文本输入。



图 2 键盘布局

3 矩阵键盘输入在Qtopia下的实现

3.1 输入法插件的建立

为在 Qtopia 中实现矩阵键盘,需要在 Qtopia 中建立输入法插件,它属于服务器插件,在 Qtopia 启动时由 Qtopia 服务器进行插件装载。输入法插件建立的步骤如下:

(1) 创建插件的接口,采用 Qtopia 已有接口

`InputMethodInterface`。

(2) 插件的功能函数

```
class QInputFrame : public QFrame, public
QWSServer::KeyboardFilter
```

(3) 为接口定义子类:

```
class QInputImpl : public InputMethod
Interface
```

(4) 创建输入法插件的实例

```
Q_EXPORT_INTERFACE()
{Q_CREATE_INSTANCE( QInputImpl )}
```

Qt/Embedded 提供了事件过滤器的接口,在插件的功能函数中安装键盘事件过滤器^[4]: `qwsServer->setKeyboardFilter (this);`

重载事件过滤器 `filter()`,所有的输入法的操作完全是在 `filter()`函数中实现。

3.2 输入法实现过程

键盘的设置逻辑如第二部分所述,是对应于我们输入过程中应用的思维,但在实现流程中对于键值的判断逻辑又有不同,首先如果要所按键的对应的键值为字母(1.2 部分所设),需要先去判断当前是否处于数字输入状态,如果是则直接将按键对应的数字发送到焦点程序,否则进入键值选择流程。然后再进入一级状态的处理中。

实现过程如图 3 所示,当有按键按下时,首先判断是否是状态切换键(中/英大/英小键和符号/数字键),是则切换相应的状态,否则查看当前状态,并进入当前状态下的按键事件的处理。

中英文切换的处理,用全局变量来表示,每次按下中英文切换键都对此变量的值进行处理:

$$m_bEnglishMode = (m_bEnglishMode + 1) \% 3;$$

`m_bEnglishMode` 值为 0、1、2 依次对应状态为中文状态、英文小写状态、英文大写状态。

对于数字/符号键的处理,用全局变量 `m_bShowSymbol` 来表示,每次按下中英文切换键都对此变量

的值进行处理: `m_bShowSymbol = (m_bShowSymbol + 1) \% 3;`

`m_bShowSymbol` 值为 0、1、2 依次对应状态为无二级状态、符号输入状态、数字输入状态。

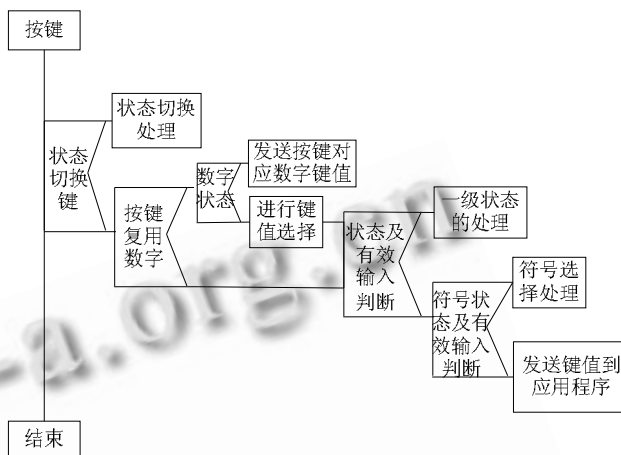


图 3 实现过程的 PAD 图

关于有效输入的说明,有效输入的定义为:在当前状态下,可以用来组成、选择、确认键值的输入。在中文输入状态下,有效输入为字母、上、下、左、右、删除、回车键,其中上下左右、删除、回车键有效的前提是拼音输入框内有拼音时,否则为无效键值;在英文大小写状态下只有字母为有效键值;在数字状态下有效键值为数字;在符号输入状态下,有效输入为上、下、左、右、回车键。对于在一级状态下的无效键值,将发送到二级状态进行处理,如果二级状态仍是无效键值,则直接将其发送到 Qtopia 当前的应用程序。通过以上分析,我们就可以实现在中英文的输入过程中输入符号,而不需要状态的切换。

下面关于一级状态的处理主要是中文状态下的处理,对于英文大小写状态下的处理比较简单不多做介绍。

3.3 按键值选择

由 8 个按键来实现 26 个英文字母的输入,参照手机的输入方式,利用按键连续按下的次数来切换不同的字母,即在一定时间内按键再次按下就使按键发送它所代表的下一个字母的 unicode。

键值选择处理过程如图 4 所示,当字母键按下时,首先判断与上次所按键是否相同,如果相同,再判断是否在限定时间内,如果在限定时间内则先发送删除键,再发送当前按键所代表的下一个字母,并记录按

键和发送的键值。

过程处理中的关键部分：首先，创建定时器，对于按键间隔的限制使用定时器来实现，使用定时器有两种方法：一种是使用 QTimer 类，一种是使用 QObject 类的定时器。在这里，采用 QTimer 类，
doublekey= new QTimer(this);

信号与槽机制是 Qt 的一个重要特征，它使函数的实现更加简便灵活，这里就利用信号与槽的机制来实现按键连接情况下键值选择的处理，将定时器的超时信号与特定槽连接起来，这个槽主要是实现一个全局变量 (timer_in) 的值的改变，用来表示此次按键是否在一定时间内，这里我们限制为 1s，在按键弹起时开启定时器：doublekey->start(1000)；计算到按键再次被按下之间的间隔，判断是否在 1s 内。

第二，每次按键按下时要记录两个值：所按的按键和所按按键发送的 unicode，所以定义静态变量 static int unicode (用来保存键盘驱动程序中所给出的按键的 unicode，即记录所按的按键) _key,unicode_press(用来保存上次按键发送给程序的 unicond)。

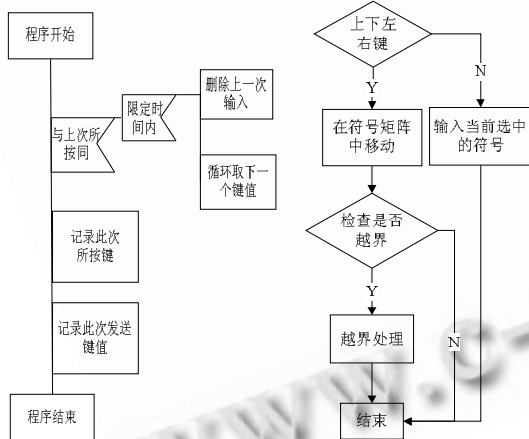


图 4 键值选择 PAD 图 图 5 符号选择处理流程图

第三，键值的选取，对于除 p 和 w 键以外，其它按键都代表三个字母，p 和 w 按键代表四个字母，所以连续按键次数是在这几个字母中循环选择。键值的选择如下：

```
if (unicode==0x70||unicode==0x77) //所按键为 p 或 w
    unicode=(unicode_press+1-unicode_key)%
    4+unicode_key;
```

```
else
    unicode=(unicode_press+1-unicode_key)%
    3+unicode_key;
    flag=1; //用于表示在限定时间内连续按同一按键事件发生
    当 flag=1 时，需要发送一个删除键，以删除上一次按键所发送的字母，SendKey(0x8,0x1003)。
```

3.4 符号选择的处理

针对符号字符的输入，由于其数量多，不方便直接将其置于按键，结合 Qtopia 的图形显示来产生虚拟键，并通过实际的上下左右键在 33 个虚拟按键中选择，通过回车键来确认输入。首先创建储存虚拟键信息的结构体：

```
typedef struct Symbolmap
{const char *label; //在 Qtopia 界面显示的字符串
  int unicode; //字符在英文状态下的 unicode
  int zhongunicode; } //字符在中文状态下的 unicode
```

然后定义 33 个字符的数组来储存 33 个虚拟键信息。利用 Qt 绘图类 QPainter 将数组所表示的图标绘制出来，并通过颜色的不同来显示当前选中的符号字符。实现界面如图 6 所示。



图 6 符号图标界面

针对符号按键的处理过程如下：当点击 (符号/数字) 键，出现符号虚拟键界面 (包含空格在内的 33 个符号)。对于符号，有区分中英文状态，如果一级状态为中文则输入中文状态符号，否则输入英文状态符号。符号选择处理流程如图 5。上下键的移动相当于数组行的选择，左右键的移动相当于数组列的选择，回车键代表确认输入数组当前行列的符号。定义全局变量 S_h_counter 和 S_v_counter、分别代表符号数组的行下标和列下标，上下左右键的按下代表这两个变量值的变化。针对越界情况，对两个变量的处理如下：

```
S_v_counter=S_v_counter+S_h_counter/11;
```

```
S_h_counter=S_h_counter%11;
```

```
S_v_counter=S_v_counter%3;
```

本输入的设计中比较突出的一点是：当拼音输入框中无拼音输入时，上下左右回车键值将作为一级状态的无效输入被发送到符号选择状态下进行处理，从而输入符号，不需要切换，可以继续输入拼音。因此在文本输入过程中不需要一直切换状态键，减少键盘的敲击次数，提高了输入效率。

3.5 中文状态的处理

对于中文状态的处理，我们采用开源拼音输入法 **MurphyPinyin**^[5] 输入汉字，同时由于它是针对普通键盘编写的输入法，对于嵌入式的小键盘，受键盘个数的限制，要完成其功能，需要做一些改进。

在定制的嵌入式矩阵键盘中，数字键与字母键复用，不能通过数字键来选择汉字，只能通过上下左右键来选择汉字，这就要求我们要将当前选中的汉字与其它汉字有所区别，这里将左右键作为翻页键，用来切换到不同的汉字选择页，上下键用来在一页中选择不同的汉字，同时为了将选中汉字凸显，需要记录上下左右键按下的次数，使在重新绘制汉字显示框时将相应的汉字凸显出来，我们定义了一个全局变量：**charchoicetotal**，表示当前选中的汉字是同音汉字中的第几个，用来表示要发送的汉字的序号，与发送键值相关，同时还与画图相关，用来决定凸显的条件。上下左右键决定着这个全局变量的值，同时关键的一点



图 7 中文输入

是在除上下左右键以外的其它键被按下时这两个变量的值都要被置零，以清除上次操作对下次拼音输入的影响，回车键完成确认输入的功能，实现界面如图 7 所示。

4 结论

本文描述了基于 **Linux** 和 **Otopia** 的嵌入式设备键盘输入的整个实现过程，通过两级状态处理与有效输入界定，并结合 **Otopia** 图形显示实现了中文、26 个英文字母、10 个数字、包括空格在内的 33 个符号的输入，涵盖了文本输入的所有字符，而且结合 **Otopia** 图形界面通过颜色的不同来凸显出当前选中的符号，使输入界面清晰。在 **Otopia** 界面下，回车键与 **ESC** 键可完成对文件的打开与关闭，所以基于 **4*4** 的键盘基本上实现了对裁剪后的 **Otopia** 界面的操作，相对于软键盘，它不需要对触摸屏的支持。对于手机键盘的符号键是每使用一次就要调用一次，而我们的键盘是符号图标可以随时处于显示和输入状态，对于符号的输入可以减少敲击次数。

参考文献

- 1 倪继利. Qt 及 Linux 操作系统窗口设计. 北京:电子工业出版社, 2006.212-270.
- 2 张萍, 徐晶. Qt/Embedded 环境下嵌入式键盘驱动的实现. 计算机工程, 2007,33(11):254-255.
- 3 冯国进. 嵌入式 Linux 驱动程序设计从入门到精通. 北京:清华大学出版社, 2008.
- 4 Trolltech company. Documentation on Qt, [2009-9-1]. <http://doc.trolltech.com/4.5/index.html>
- 5 MurphyPinyin 输入法. [2009-10-10]. <http://galen.lupa-world.com/action-download-itemid-2839.html>