Hi3510 keypad 应用

Application Notes

文档版本	01
发布日期	2006-09-10
BOM编码	N/A

深圳市海思半导体有限公司

深圳市海思半导体有限公司为客户提供全方位的技术支持，用户可与就近的海思办事处联系，也可直接与公司总部联系。

深圳市海思半导体有限公司

地址： 深圳市龙岗区坂田华为基地华为电气生产中心 邮编：518129

网址： http://www.hisilicon.com

客户服务电话： 0755-28788858

客户服务传真： 0755-28788838

客户服务邮箱： support@hisilicon.com.

版权所有 © 深圳市海思半导体有限公司 2006。 保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明

 、Hisilicon、海思，均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

前 言

概述

本节介绍本文档的内容、对应的产品版本、适用的读者对象、行文表达约定、历史修订记录等。

产品版本

与本文档相对应的产品版本如下所示。

产品名称	产品版本
Hi3510 通信媒体处理器芯片（简称 Hi3510）	Hi3510 V100
	Hi3510 V101
	Hi3510 V110
Hi3510 DVS 方案	Hi3510 DMS V100R001
Hi3510 VPhone 方案	Hi3510 DMS V200R001

读者对象

本文档主要适用于软件开发人员。

内容简介

本文档首先介绍 GPIO 引脚与按键的连接方式和键盘扫描策略；然后说明键盘驱动模块的基本功能，模块提供的用户接口函数和模块应用示例。

章节	内容
1 概述	简要介绍 GPIO 实现键盘 IO 接入的主要内容。
2 GPIO 实现键盘 IO 接入	详细介绍 GPIO 实现键盘 IO 接入的连接方式和键盘扫描策略。

章节	内容
3　键盘模块	详细介绍键盘模块的基本功能、用户接口函数和应用示例。

约定

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	以本标志开始的文本表示有高度潜在危险，如果不能避免，会导致人员死亡或严重伤害。
 警告	以本标志开始的文本表示有中度或低度潜在危险，如果不能避免，可能导致人员轻微或中等伤害。
 注意	以本标志开始的文本表示有潜在风险，如果忽视这些文本，可能导致设备损坏、数据丢失、设备性能降低或不可预知的结果。
 窍门	以本标志开始的文本能帮助您解决某个问题或节省您的时间。
 说明	以本标志开始的文本是正文的附加信息，是对正文的强调和补充。

通用格式约定

格式	说明
宋体	正文采用宋体表示。
黑体	一级、二级、三级标题采用黑体。
楷体	警告、提示等内容一律用楷体，并且在内容前后增加线条与正文隔离。
“Terminal Display”格式	“Terminal Display”格式表示屏幕输出信息。此外，屏幕输出信息中夹杂的用户从终端输入的信息采用加粗字体表示。

修改记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

文档版本 01 (2006-09-10)

第一次版本。

1 概述

本文主要介绍 GPIO 实现键盘的 I/O 接入和键盘驱动模块的工作原理。主要分为以下两个方面进行介绍：

- GPIO 引脚与按键的连接方式及键盘扫描策略。
- 键盘驱动模块的基本功能和应用示例。

2 GPIO 实现键盘 I/O 接入

2.1 键盘硬件连接

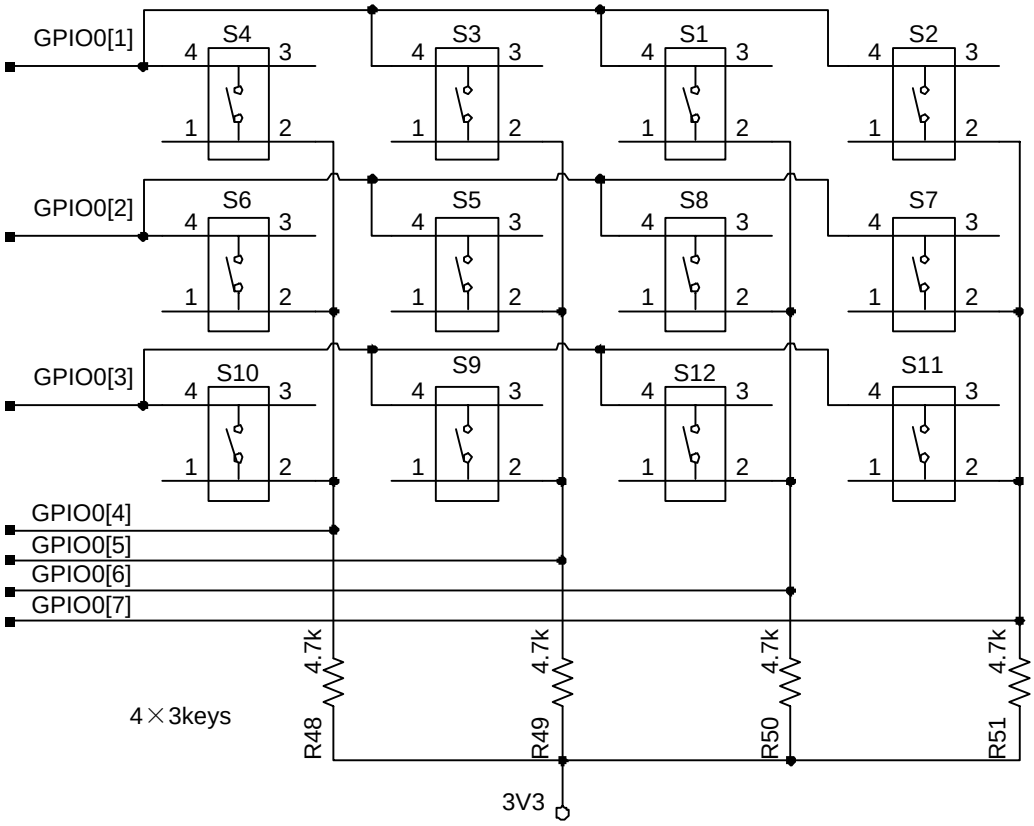
在视频评估板（VSEVB）中，键盘驱动模块共支持 12 个按键，分 3 行 4 列。每 1 行或每 1 列都对应 1 个 GPIO 引脚。可通过适当的键盘扫描策略实现对按键操作的检测。

说明

用于连接键盘行和列的 GPIO 引脚和键盘的布局都可以根据需要更改。如果更换 GPIO 引脚或更改键盘布局，需要修改驱动模块的 GPIO 初始化函数和键盘扫描函数。

键盘的硬件连接原理图如图 2-1 所示。

图2-1 键盘硬件连接原理图



2.2 键盘扫描策略

在 VSEVB 中，键盘利用 GPIO0[3:1]的 3 个引脚和 GPIO0[7:4]的 4 个引脚分别作为键盘接入的行和列，具体情况如表 2-1 所示。

表2-1 GPIO0[3:1]和 GPIO0[7:4]的引脚说明

引脚	行选择/列选择	输入/输出	初始化电平
GPIO0[3:1]的 3 个引脚	行	输出	低电平
GPIO0[7:4]的 4 个引脚	列	输入	高电平

键盘驱动模块将 GPIO0[7:4]的中断触发方式设置为双沿触发，即当按键按下或按键弹起时，都有中断发生。

当按键被按下时，行选择和列选择连通。确定按键的方法如下：

- 确定按键的列

当 GPIO0[7:4]中的 1 个引脚由高电平变为低电平后会触发中断，可以确定 GPIO0[7:4]中的某一引脚连接的按键被按下，则可确定按键的列，即与电平变低引脚连接的列。

- 确定按键的行

  GPIO0[3:1]的 3 个引脚轮流输出低电平，若扫描到已确定按键的列为低电平则可确定按键的行，即与输出低电平的引脚连接的行。

按键的行和列的确定就可以唯一地确定 1 个按键。

3 键盘驱动模块

3.1 基本功能

键盘驱动模块的基本功能是响应按键操作。对按键进行按下和弹起操作，可以返回以下 3 个值：

- 按键按下的键值

  小写字母表示按键按下。
- 按键弹起的键值

  大写字母表示按键弹起。
- 按键的持续时间值

  在按键弹起时，返回按键操作的持续时间。

  时间单位为时钟滴答。在当前系统中，100 个滴答表示 1 秒。

用不同的字母区分不同的按键。

3.2 注意事项

对键盘模块操作时，要注意去抖动和低电平恢复。

去抖动

当按键按下时，电平不会平滑地由高变低。在电平变低的过程中会出现毛刺和抖动，然后才变为低电平。当按键弹起时，也同样会出现抖动。

如果在抖动的过程中读数，会有以下影响：

- 造成读数的不准确。
- 频繁的触发中断。

因此，软件采用延时一段时间的方法进行去抖动。在延时的时间内，电平抖动引起的所有中断将被屏蔽，延时时间结束后，再进行读数和其它操作。

低电平恢复

在扫描键盘结束后，要将 GPIO0[3:1]的电平恢复为低电平。

3.3 接口说明

用户只能通过标准的 I/O 接口函数与键盘驱动模块进行通信，用到的接口函数如下：

```
int open(const char *pathname, int flags);
int close(int fd);
ssize_t read(int fd, const void *buf, size_t count);
int select(int n, fd_set *readfds, fd_set *writefds, fd_set *exceptfds,
           struct timeval *timeout);
```

3.4 应用示例

下面的代码完成读取键值和按键持续时间值。当有按键操作时，可依次打印与按键对应的大写字母、小写字母和按键持续时间。

```
#include <fcntl.h>

static int fd = -1;
int  main(void)
{
    int result;
    int counter = 1,tmp;
    int buf[1];

    /* open keypad in O_NONBLOCK mode */
    fd=open("/dev/misc/keypad",O_NONBLOCK);
    if(fd<0)
    {
        printf("Can't open\n");
        return -1;
    }
    while(1)
    {
        /* read the keyvalue_buf of kernel,return immediately */
        result = read(fd,buf,sizeof(buf));
        if(result>0)
        {
            if(buf[0]<0)
            {
                printf("An int leaked!reset!***********\n\n\n");
```

```
                counter = 1;
                continue;
            }
            tmp=counter%3;
            if(tmp)
            printf("the result is  %c\n",buf[0]);
            else
            printf("the durtion is %d\n\n\n\n",buf[0]);
            counter++;
        }

    }
    return 0;
}
```