# 独立按键（查询）

本章主要讲解独立按键的检测、消抖及其简单的使用。

## 1、硬件及原理

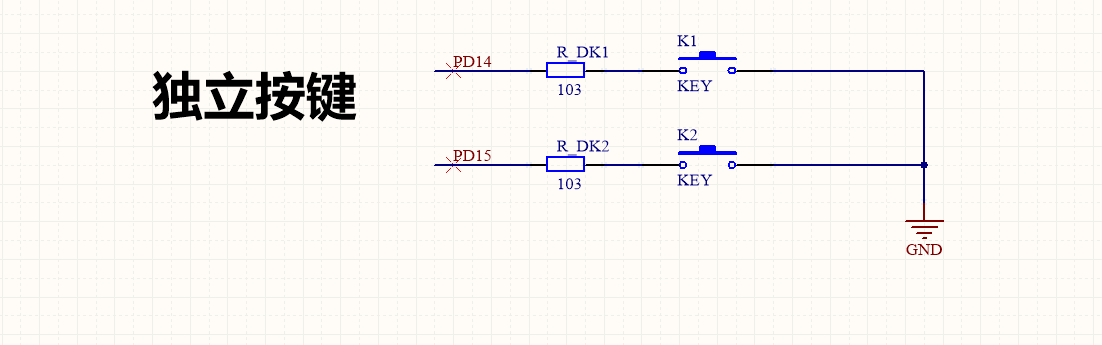


图 1

718stm32开发板独立按键原理图如图一所示，由原理图可知，当按键按下时，单片机对应的GPIO引脚被接地，GPIO引脚的输入状态为低电平；当按键弹起时，我们需要GPIO引脚的输入为高电平，所以我们将GPIO引脚设置为上拉输入，设置完成后内部原理图大致图2所示

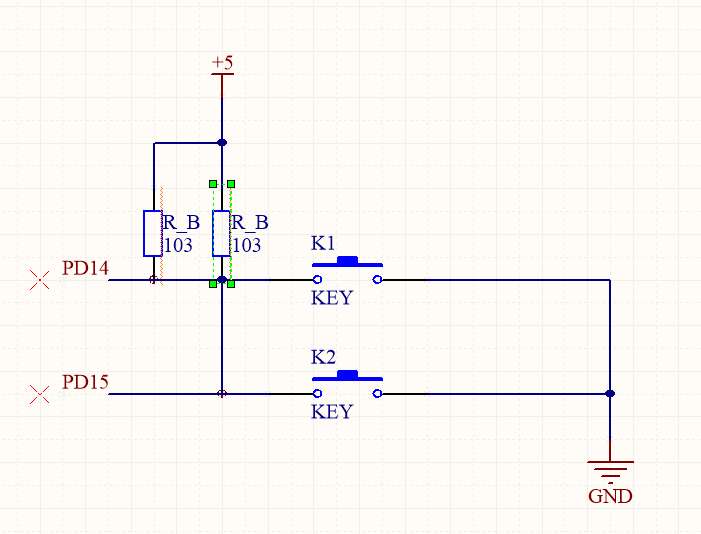


图 2

C:\Users\admin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\5028e3d9ac79c9cf6fd7e7af549fd0a1ea6de6b0.jpg2、软件消抖简介

图 3理想波形

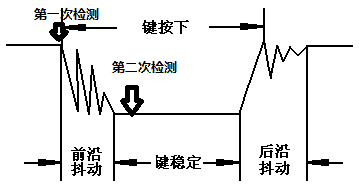


图 4实际波形

如图所示，在按键机械触点断开，闭合时，由于触点的弹性作用，按键开关并不会马上稳定接通或一下子断开，使用时会产生图中所示的带纹波的信号，如果不处理，一次按键按下可能会判断为多次按下。所以需要用到消抖处理，具体方法为在检测到按键第一次按下时，进行一个短暂的延时（10ms左右），然后再进行检测，只有当第一第二次检测都是按键按下时，才判定为按键按下。

## 3、软件设计

### 3.1编程要点

1、使能GPIO端口时钟；

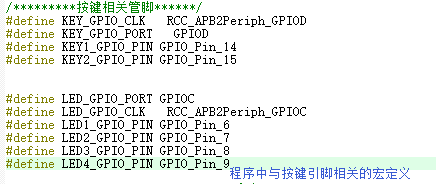
2、初始化GPIO目标引脚为输入模式（上拉输入）；

3、编写简单的测试程序，检测按键的状态，实现按键对LED的控制。

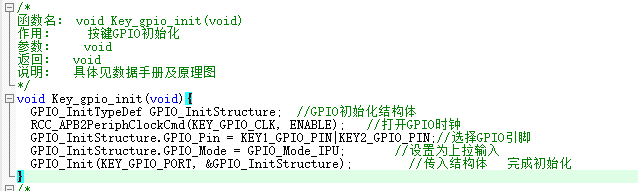
### 3.2代码分析

**1、按键引脚宏定义**

考虑到硬件情况可能发生改变，我们把按键检测的相关引脚进行了宏定义，修改时直接修改宏定义即可。



**2、按键GPIO初始化函数**



1）使用GPIO\_InitTypeDef定义GPIO初始化结构体变量，以便线面用于存储GPIO配置。

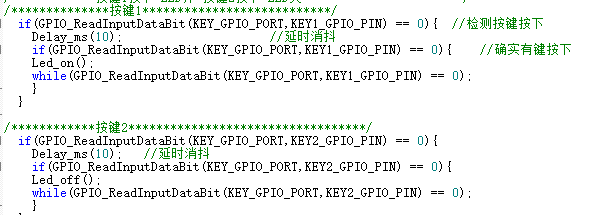
2）调用库函数RCC\_APB2PeriphClockCmd（）来使能GPIO的端口时钟，调用时可用‘|’同时配置多个时钟。

3）更改结构体变量GPIO\_InitStructure.GPIO\_Pin时也可使用‘|’来同时操作多个引脚。

4）完成所有对于初始化结构体的配置以后，调用GPIO\_Init函数向寄存器写入参数，完成对GPIO的初始化。

**3、检测按键的状态**

初始化完按键之后，就可以通过检测对应引脚的电平来判断按键的状态了。程序如下图：



GPIO\_ReadInputDataBit(GPIO\_TypeDef\* GPIOx, uint16\_t GPIO\_Pin)；

该函数为检测对应引脚电平，函数有两个参数：第一个为GPIOx（GPIOA、GPIOB……），第二个为GPIO\_Pin\_x（x=1,2,3……）。

由原理图可知，检测到低电平为按键按下，检测到高电平为按键弹起。Delay\_ms（10）为延时消抖处理。

### 3.3下载验证

把编译好的程序下载到开发板并复位，按键一和二分别控制LED的亮和灭。