# 红外遥控-鼠标键盘-使用说明&测试报告

## 项目要求

1. USB\_HID设备
2. 支持hid keyboard设备
3. GPIO按键功能开发
4. 支持单按键发送，例如A,B,C,…,1,2,3…
5. 支持组合键发送，例如Ctrl+F4
6. window能够正常响应对于的keyboard键值
7. 支持hid mouse设备，使用操控杠实现上、下、左、右移动功能
8. 红外遥控USB\_HID设备
9. 支持识别红外遥控NEC协议；
10. 抗干扰纠错处理；
11. 识别遥控指令，通过USB控制PC相关功能：数字0-9、F1-F12、Left、Right、Down、Up、Enter、Window、Alt+F4、Alt+Tab；

## 开发环境介绍

1. 开发板使用的是dia的PORT-205R
2. 编译器KEIL版本：5.0
3. 代码语言：C语言
4. Git仓库https地址：

<https://gitdojo.gz.cvte.cn/WelcomeOs/stm32_hid_keyboard_mouse>

## 构建方法

1. 下载代码：$ git clone

<https://gitdojo.gz.cvte.cn/WelcomeOs/stm32_hid_keyboard_mouse>

(2) 使用KEIL5.0编译代码生成hex文件

(3) 将hex文件烧录进微雪PORT-205R开发板

## 功能说明

此项目分为两个程序，第一个是USB\_HID设备，第二个是红外遥控USB\_HID设备，下面分别介绍这两个程序的功能。

### USB\_HID设备

USB\_HID设备连上PC后，会被识别为USB键盘-鼠标复合设备。KEY1、KEY2、KEY3可以作为键盘按键使用，初始设置为KEY1-b，KEY2-TAB，KEY3-Alt，按键可以通过KEIL在app.c文件中进行修改。键盘按键支持单按键、组合按键、长按。摇杆可以作为鼠标使用，上下左右可以控制鼠标在四个方向的移动，按下遥控代表使用鼠标左键。

### 红外遥控USB\_HID设备

设备连上PC后，会被识别为USB键盘。此时通过遥控器可以控制PC上的键盘，支持单按键、组合按键、长按。遥控键盘按键包括0~9，F1~F12，Win键、空格键，上下左右箭头按键，page-up、page-down键，Alt+TAB组合键、Alt+F4组合键。

## 功能测试

### USB\_HID设备

测试方案：

1. 设备连上电脑，测试是否能够识别USB-键盘鼠标复合设备；
2. 识别成功后，键盘三个按键的单按键、组合按键、长按测试；
3. 鼠标的四个方向摇杆控制，鼠标左键灵敏度测试。

测试结果：

1. 当设备连上电脑时，打开设备管理器，可以看到电脑成功识别连上的时USB-键盘鼠标复合设备。
2. 单按KEY1按键，文档里出现b，长按KEY1按键，文档里出现连续的b，表示b键被长按；单按KEY2按键，文档出现空格，表示TAB按键被按下，长按KEY2按键，文档里出现连续的空格，表示TAB按键被长按；单按KEY3按键，文档功能栏出现字母标识，表示Alt按键被按下，长按KEY2按键，文档无反应，表示Alt按键被长按。
3. 按住KEY3按键，再按住KEY2按键，模拟实际键盘先按Alt再按TAB按键进行窗口的切换功能，得到了与实际键盘一样的结果，表示组合键没有异常。
4. 使用摇杆进行四个方向的移动，鼠标会根据摇杆的方向进行移动，按下摇杆，鼠标左键被点击，表示鼠标功能正常。

### 红外遥控USB\_HID设备测试

测试方案：

1. 设备连上电脑，测试是否能够识别USB-键盘设备；
2. 遥控器单按键、长按、组合键测试。

测试结果：

1. 当设备连上电脑时，打开设备管理器，可以看到电脑成功识别连上的时USB-键盘设备。
2. 按下遥控器的0~9时，文档里会出现对应的字符，长按时会持续出现对应的字符，说明单按键与长按功能正常；按下遥控器上的Alt+TAB按键时，电脑会切换窗口，说明组合按键功能正常。

## 方案设计

## 1 系统功能的设计

### 1.1功能需求

USB\_KEYBOARD\_MOUSE设备要求支持HID KEYBOARD、HID MOUSE设备，支持用户定制按键，支持单按键、组合按键、长按，支持红外遥控控制，Windows能够正常响应对于的KEYBOARD键值。

### 1.2系统流程分析

系统流程就是开发板烧录该程序后上电之后的工作过程，其功能如1-1所示。

开始

键值打包发送

检测按键

图6-1系统流程图

这套方案工作的基本流程是检测按键，按键检测得到的有键盘数据、鼠标数据、红外遥控数据三种数据类型，当检测得到的是键盘数据时，进行键盘数据的打包并发送Windows能识别的键盘数据；当检测得到的是鼠标数据时，进行键盘数据的打包并发送Windows能识别的鼠标数据；当检测得到的是红外遥控数据时，对红外遥控数据进行解析并与键盘数据配对，打包成键盘数据并发送Windows能识别的键盘数据。

### 1.3状态转换图

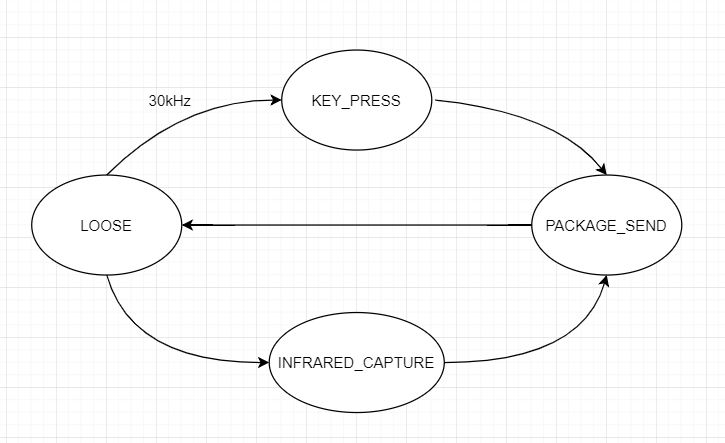


图6-2 状态转换图

程序的设计采用了状态机的思想，主体分为四个状态，一共有两种运行情况：第一种是初始状态为LOOSE状态，不会进行任何动作，当检测到有按键被按下时，进入KEY\_PRESS状态，此时进行按键的解析与打包，根据哪个按键被按下选择打包成键盘数据或者鼠标数据，之后进入PACKAGE\_SEND状态，进行打包后键值的发送，发送完毕后回归到LOOSE状态。第二种是初始状态为LOOSE状态，不会进行任何动作，当接受到红外遥控数据时，进入INFRARED\_CAPTURE状态，对红外数据进行解析，再进入PACKAGE\_SEND状态对解析完成的数据进行打包，打包成键盘数据进行发送，发送完毕后回归到LOOSE状态。

## 2 系统方案的设计

2.1 按键检测的解决方案

按键检测采用定时器轮询方式检测，使用定时器设定30kHz的频率对按键的接口进行轮询，获取接口当前的状态用以判断按键是否被按下。

2.2 按键防抖的解决方案

按键防抖是在按键检测的基础上，定时器以30kHz的频率轮询按键接口当前的状态，当连续三次检测到按键被按下的情况后才会进入按键的打包发送程序，以此来避开电平抖动带来的按键误触发问题，实现按键防抖。



图6-3 按键防抖代码示例

2.3 组合按键的解决方案

组合按键主要对应的是键盘和红外遥控，采用的方案是检测按键时判断是否多个按键一起被按下，当多按键被按下时打包程序会进行组合按键的打包方式，从而实现多按键功能。

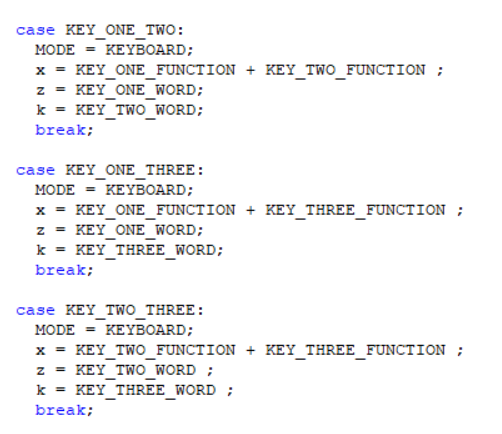


图6-4 组合按键代码示例

2.4 红外遥控数据接收的解决方案

由于红外遥控数据信息装载在传送的信号上，所以对于红外遥控数据的接收采用了定时器捕获触发中断的方式，当捕获到红外遥控数据时进入中断，识别到同步码时则进入遥控数据解析、打包、发送程序。

2.5 按键抬起的解决方案

由于USB键盘发送一个按键的数据包后Windows会识别成该按键一直被按下的状态，因此发送完一个按键的数据包后应该再发送一个空的按键数据包，Windows才能完成一个完整的按键按下、抬起动作。这里采用定时器定时发送空键值数据包的方案实现按键抬起的动作。

2.6 鼠标、键盘复合设备识别的解决方案

通过CubeMX搭建USB—HID设备的底层，通过修改HID三个描述符：设备描述符、报告描述符、配置描述符使得USB设备能被正确识别为鼠标、键盘复合设备。

2.7 用户定制的解决方案

通过在app.c这个文件中的宏定义，用户可以很方便的修改自己需要的键盘和遥控器。