

一种基于 USB 的无线触摸鼠标板的设计与实现

摘要：针对一种基于 USB 的无线触摸鼠标板的设计与实现方法,给出了较为详细的设计和实现过程。该设计结合了 USB 和无线设备的优点,将手持端的鼠标触摸板控制信息通过无线发送到接收端并通过 USB 接口与主机 PC 实现交互通信。整个系统主要由 PIC16F877A 单片机,PDIUSB12 芯片,CC2420 无线模块,PS/2 鼠标触摸板实现。

随着 IT 技术的发展,多媒体教学已经在学校得到广泛的应用。但传统的多媒体教学存在互动性不强等缺点,不利于师生的交流和沟通,基于这种情况,我们设计和实现了一种基于 USB 的无线触摸鼠标板,使得教师在学生座位旁依然可以像站在讲台那样自由控制电脑。

1 系统组成

本系统可以大致分为两个部分,其中一端是手持设备终端,提供 PS/2 接口供鼠标触摸板接入使用;另一端是 PC 接入端,通过 USB 接口与主机相连,二者之间通过无线传输协议联系在一起。整体构架图如图 1 所示。

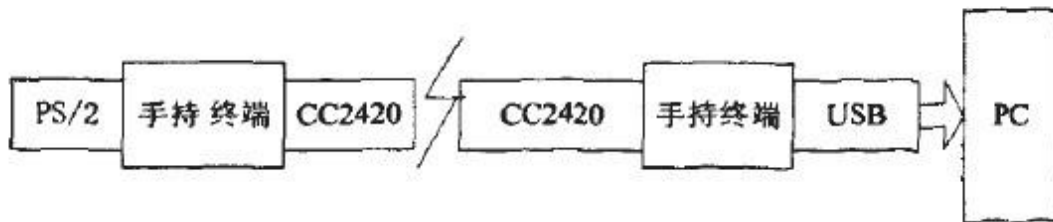


图 1 整体构架图

因为设计所选用的 CC2420 集成了发送和接收功能在同一芯片中,故在手持终端和接入设备部分的无线收发器可复用同一模块,无需重新设计。而在 PC 端,为了实现设计的普遍适用性以及即插即用的特性,对接入设备采用了 HID 类设计从而使 PC 端无需增加任何软件,即可使用操作系统提供的设备驱动程序。

本设计的实验开发板由笔者自行设计实现,不仅包括了本设计所需要的功能,还有调试功能以及扩展功能,供日后有需要时使用。

2 硬件设计与实现

2.1 手持端硬件设计与实现

在手持端,主要包括的模块有 MCU 控制模块,无线发送模块,PS/2 接口模块和调试模块等。各模块的功能示意图如图 2 所示:

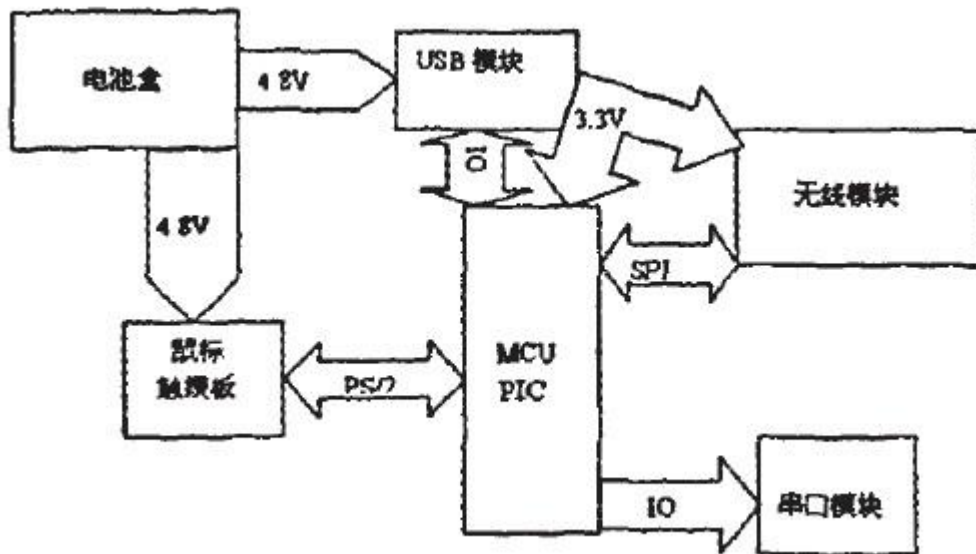


图 2 手持端模块图

2.1.1 MCU PIC 整体控制模块

MCU 控制着整个系统的运行，包括软件的触发、运行，以及各个部分的协调工作。整个系统的软件都是存储并运行在 MCU 当中，在触摸板事件响应方面采用的是中断触发。而 PS/2 接口实际上只需两条 IO 线，也就是说其只占用了 MCU 两个管脚资源，其中一个为普通 IO 口，另一个则是中断口，考虑到中断 12 的可能冲突，可以通过跳线方式自行选择。在 CC2420 方面，亦采用电平跳变中断检测其 SFD 及 FIFOP 的中断请求，而其数据通信则是使用芯片集成的 SPI 接口中的 RCA (SO)、RC5 (si)、rico (CS-N)、RC3 (SCK) 等 4 条线来实现。

2.1.2 串口模块

由于 USB 的数据信息在其未被主机正确识别之前无法得到，并且在无线传输方面也需将数据及控制信息等传回主机以供调试，所以在开发板上另外设计一个串口调试模块，方便设计过程中的调试需要。

RS232 规范使用 12V 供电，而开发板上使用的电压为 3.3V,这样如果直接将数据线直接连到 RS232 接口上会将实验板上的元件烧毁。为了实现不同电压之间的连接，采用了 RS232 电平转换芯片 MAX3232.利用这个芯片我们可将 12V 电压的数据传输转换为 3V,这样就可以达到将数据与主机进行交互的目的。在数据传输线路上，由于 PIC16F877A 内置了串口接口，所以只需使用 RC6、RC7 这两根线就可以实现数据的传输，经过 MAX3232 的电平变换到达主机。

2.1.3 无线传输模块

考虑到工艺问题，靠手工无法完成，所以采用模块购买的方式来获得 CC2420 的使用，购买的模块仅有非常简单的外围器件，以及 12 根控制引脚外引（包括电源和地），这样我们

就可以通过 MCU 利用这 12 根引脚控制 CC2420 的工作了。根据该模块的外接引脚，在实际的电路设计中设计了一个两排 12 根线的接口，可以直接接到 MCU 控制接口，因为 MCU 和 CC2420 都是采用 3.3V 的电压，所以不会有电压过高的问题。

在 CC2420 的电路设计过程中发现，对于控制引脚 SFD、FIFOP 这两个用于检测中断接受数据的接口尤其重要，如果在这两条线路出现了干扰的话，在软件的实现方面将产生很多不必要的软件开销，例如去除干扰等。所以在 PCB 设计的时候要对这两条线加宽，不要横跨电源和地，这样会产生很大的干扰。

2.2 PC 接入端硬件设计与实现

此部分将介绍接入设备的硬件设计与实现，在串口调试部分采用和手持端同样的硬件设计，而在无线接收模块，因为 CC2420 集合接收发送功能在同一芯片中，而应用电路也采用同样电路，故在接入设备部分可以使用相同模块。所以在这里将主要介绍 USB 模块的设计与实现。

在 USB 接 IZI 方面，为了使用上的方便，将设备设计成标准的 HID 类鼠标，这样可以直接使用操作系统自带的驱动程序而无需安装任何的驱动软件。在供电方面使用标准 USB 接口提供的 5V 电源为 USB 芯片供电，接入端的主要硬件架构可用图 3 所示：

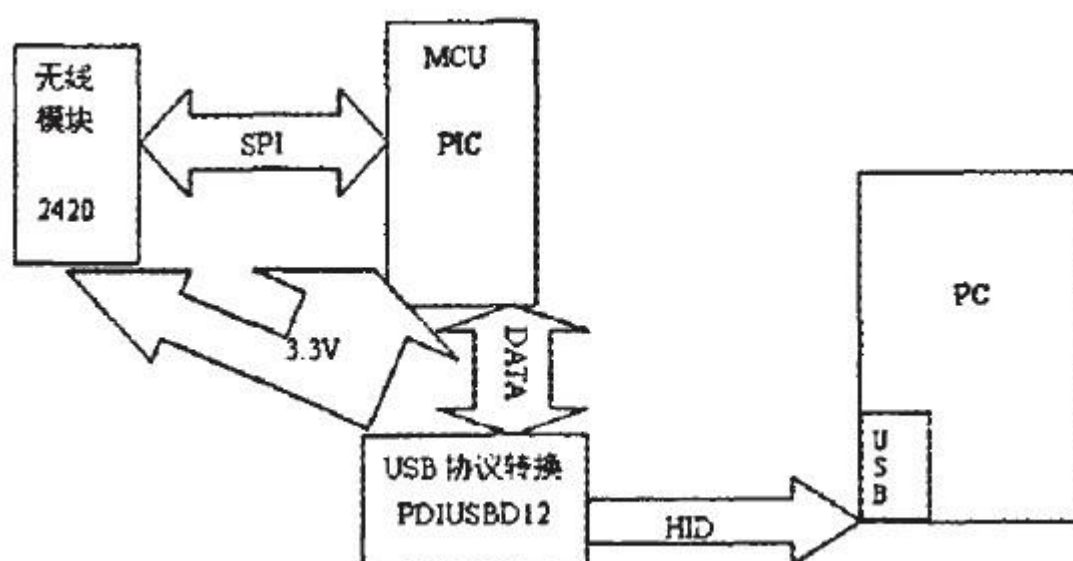


图 3 接入端硬件构架图

在 USB 接口芯片的连接上，采用了与 MCU 控制器的 INTO 外部中断的方式连接，这样可以保证 USB 数据传输的实时性，而其他的控制接 EI 使用一般的数据 IO 口，其 8 位数据传输口使用 PIC16F877A 的 RD 口连接。

3 系统软件设计与实现

在本部分，主要将介绍本设计的软件实现，其中包括接入端和手持端两大部分。

3.1 整体设计方案的设计与实现环境

本次设计采用的控制芯片为 PIC16F877A。

由于程序是在 PC 上设计，需要将程序下载到单片机上运行和测试，所以需要有一个程序下载工具，本次设计所采用的程序烧录器为 TOP WIN2004,该烧录器自身提供有一个下载软件，使用也非常方便，安装按照软件目录下的 README 一步步操作即可。

本设计中的软件均是在单片机内实现，运行在 MCU 内的软件控制着整个设计所要实现的模块。

但是由于 MCU 与传统 Pc 有着巨大的差别，必须考虑存储空间、运行速度、外界干扰等因素。在实现此次设计的过程中，除了首先要保证硬件环境的设计正确和优质之外，软件方面的严谨设计也是必不可少的一个重要部分。

本设计的软件总体框架图可如图 4 所示，为了使两端达到速度及任务的均衡，一些软件处理将在手持端完成，然后以接收端可以识别的格式打包发送过去，由图 4 也可以看出两端的任务处理基本处于均衡状态。

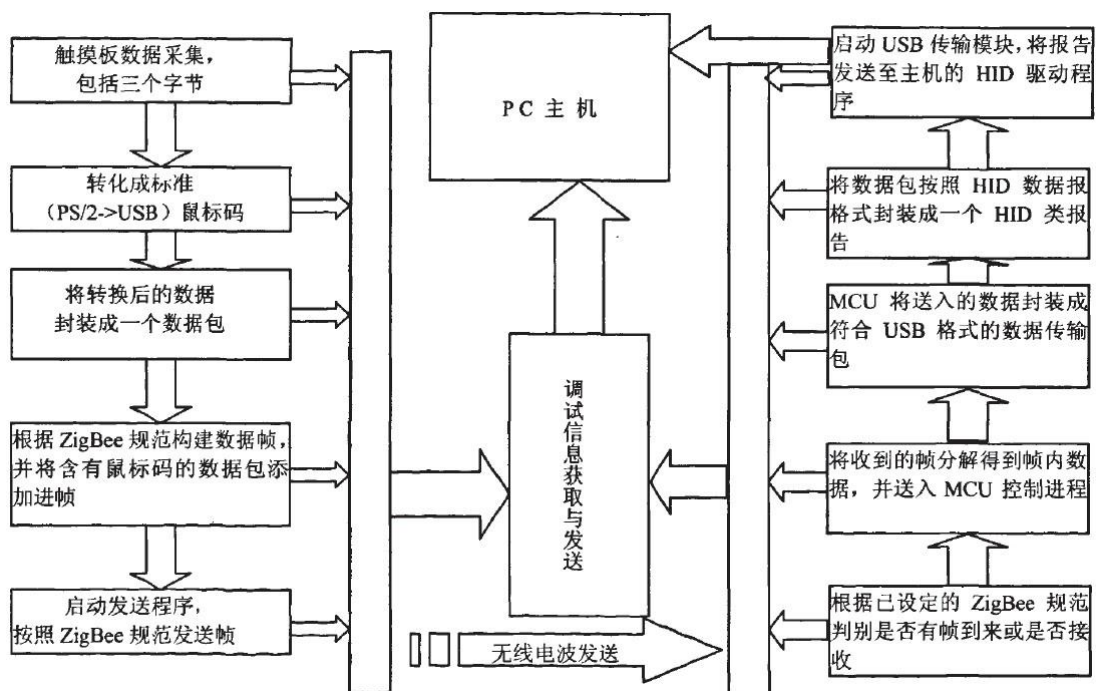


图 4 软件总体框架图

3.2 手持端软件的设计与实现

在本部分将介绍手持终端的软件设计过程，以及各个模块部分的具体实现。软件的运行流程图如图 5 所示。

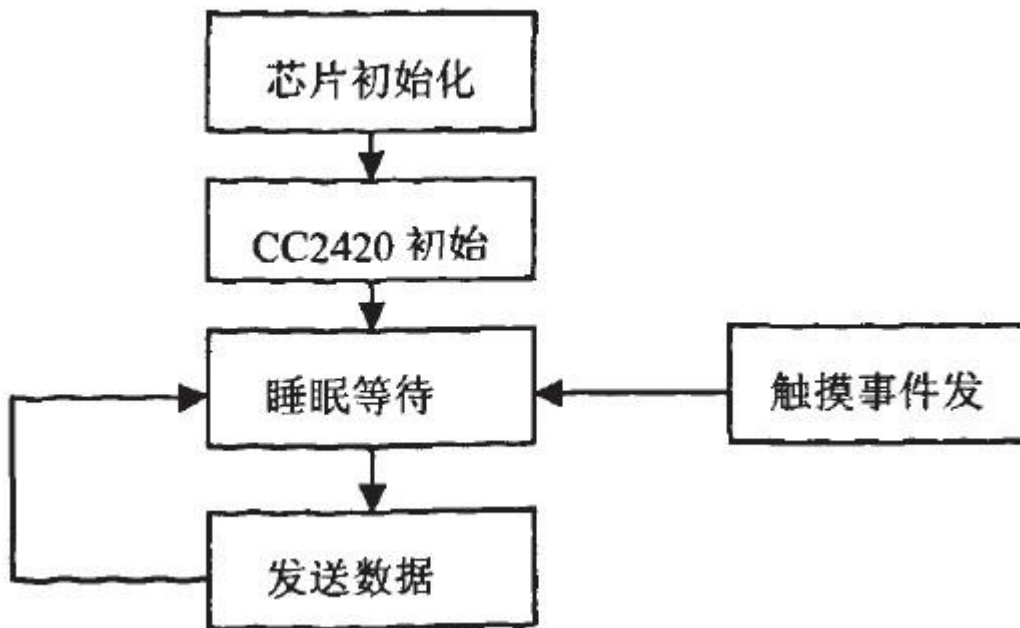


图 5 手持端软件流程

对于事件发生的获取是采用中断模式，也就是在有鼠标触摸板事件发生的时候，将唤醒 MCU 进行数据的采集处理，然后将采集到的 PS/2 数据进行一定规则的转换，然后将其打包发送。若没有数据需要处理，则进入睡眠状态以节省功耗。

3.2.1 芯片初始化

在芯片初始化阶段，主要是芯片的内部工作的时钟的设置、IO 口的初始化设置、以及 CC2420 的初始化。

在芯片的初始化阶段需要注意对每个 IO 口的数据方向进行设置，并且要准确设置时钟以及串口的传输数率，否则在调试阶段无法获取数据。不用的 IO 口全部设置成输出口，这样可减少端口不必要的干扰。

3.2.2 鼠标触摸板事件的信息采集

首先要对触摸板进行初始化，其初始化序列请参照 PS/2 协议的命令序列。初始化成功后，该触摸板将正常工作，只要有触摸事件发生，将产生连续三帧的数据发送，而按键事件发生，则在按下产生三帧数据发送，弹起产生三帧数据发送。我们需要做的工作就是在触摸板发送数据的时候对其进行采集。采集过程也非常简单，在熟悉 PS/2 协议的基础上，中断产生后，按照时序对数据进行采集，一定要在读完一帧的基础上再读下一帧，读取完三帧数

据后就马上进行数据转换和打包发送，然后再次等待中断事件发生。所选择的 MCU 在速度处理上完全可以满足一般键盘鼠标数据包发送的间隔时间，也就是说在读取完一个数据包后，就处理该数据包而不需要完全读取完一次事件产生的数据堆。

3.2.3 CC2420 数据发送

在 CC2420 的数据发送阶段，MCU 将数据发送命令传送给 CC2420 之后，可以通过 SFD 引脚来判断其发送过程。该引脚会有一个由低电平向高电平跳变的过程，其过程如图 6 所示。

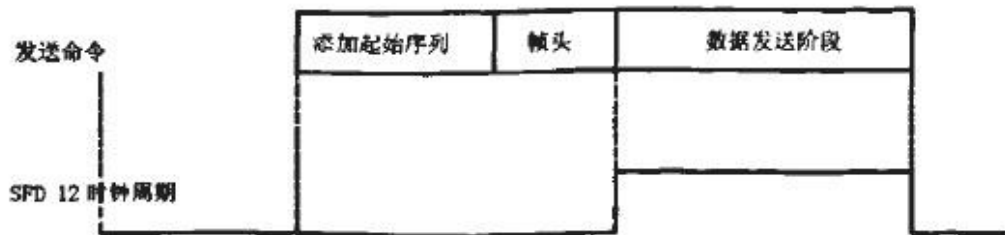


图 6 发送过程示意图

控制器向 CC2420 发送了发送数据的命令之后，并不能马上把数据发送出去，而是要经过 12 个时钟周期之后发送器才启动，之后将数据添加了起始序列和帧头之后，才能开始数据发送进程。在数据发送完毕之前，SFD 引脚将一直为高电平，这样在 MCU 就可以根据此引脚的变化来跟踪发送过程，发送过程具体的软件实现，可用图 7 表示。

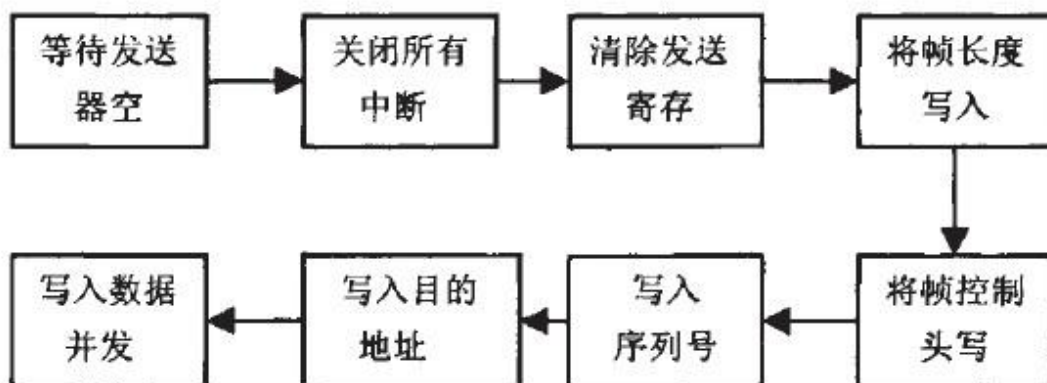


图 7 CC2420 发送实现过程

在 CC2420 的数据发送阶段，发射校正和添加起始序列和帧头可以由硬件自动完成，但是帧的长度需要在软件中计算得到再写入帧中，并且需要将数据写入发送器的发送缓冲区之后向 CC2420 发送发送启动命令。

3.3 接入端软件的设计与实现

接入设备部分的软件设计，主要是分为两部分：CC2420 的接收和 USB 数据传输。CC2420 由于是发送和接收集成在同一模块中，所以其初始化可采用发送端的设计。而在 USB 数据传输部分，则采用 HID 的协议与主机进行通信，这样在软件设计上可以减少复杂度。

3.3.1 无线数据接收

无线数据接收的软件实现，使用 RB5 端口的电平中断触发接收进程。当 CC2420 发现有帧传输过来时其引脚会产生相应的变化，MCU 通过检测这些引脚的变化可以得知 CC2420 所在的接收阶段与状态。在接收过程中，有 3 根引脚可以检测出 CC2420 的状态，其数据接收的变化示意图限于篇幅，在此略去。

当 CC2420 收到有符合其起始序列的时候，其自动接收帧头和帧的长度，并且 SFD、FIFO、FIFOP 这 3 根输出引脚分别输出上图所示电平。当接收到帧的目的地址时候，CC2420 自动进行地址匹配，若地址匹配成功，则接收进程继续进行，如果地址匹配不成功，则马上停止接收过程，并把 3 根输出引脚都置为低电平。MCU 在控制时可以通过检测这 3 根输出引脚的电平来判断接收过程的状态。

如果地址匹配成功且接收完整个帧，则可以启动数据读取进程。

在数据读取阶段，CC2420 提供了帧的正确接收的初始判断，由此可以知道该帧是否被正确接收，以决定是读取该帧还是放弃该帧。按照 CC2420 的规定，当一个帧接收完毕之后，如果出现 FIFOP=1 并且 FIFO=0 的情况，则说明该帧错误接收，应当向其连续写入两个 CC2420_SFLUSHRX 命令将数据冲刷。如果没有出现这种情况，则按照帧发送的相反过程从接收缓冲中读取数据。

3.3.2 USB 设备 HID 类设计与实现

USB 数据传输的软件设计，主要是集中在 USB 芯片的初始化以及协议的初始化、以及之后的枚举过程，在成功枚举并被主机正确识别之后，只需将数据写入发送缓冲即可将数据发送到主机。在协议的初始化时，需要将协议中规定的各个参数设定都按照设计的要求设定，并且将各种描述符，包括配置描述符、端点描述符、接口描述符、报告描述符等初始完毕，以供枚举过程使用。

USB 芯片的初始化过程的最后一步是将其 SoftConnect 功能打开，从这时起，主机才能开始检测到设备的接入，也从这个时候起，USB 芯片才真正开始与主机通信并开始设备枚举过程。在枚举过程中，主机首先向设备发送启动信号，设备在接收到这个信号之后向主机发送回送的协议数据以供主机识别，当识别到这个信号之后，主机则开始向设备获取描述符等信息，进而完成设备枚举过程。

初始化完成之后，主循环实际上不再参与 USB 通信。每次 PDIUSB D12 接口芯片检测到 USB 总线上主机发给这个设备的信号时候就触发一次单片机的外部中断。中断处理子程序要做的工作就是首先 D12 读取中断信息，查看 D12 收到了什么样的封包，然后根据主机的不同要求做出响应，中断处理程序流程如图 8 所示。

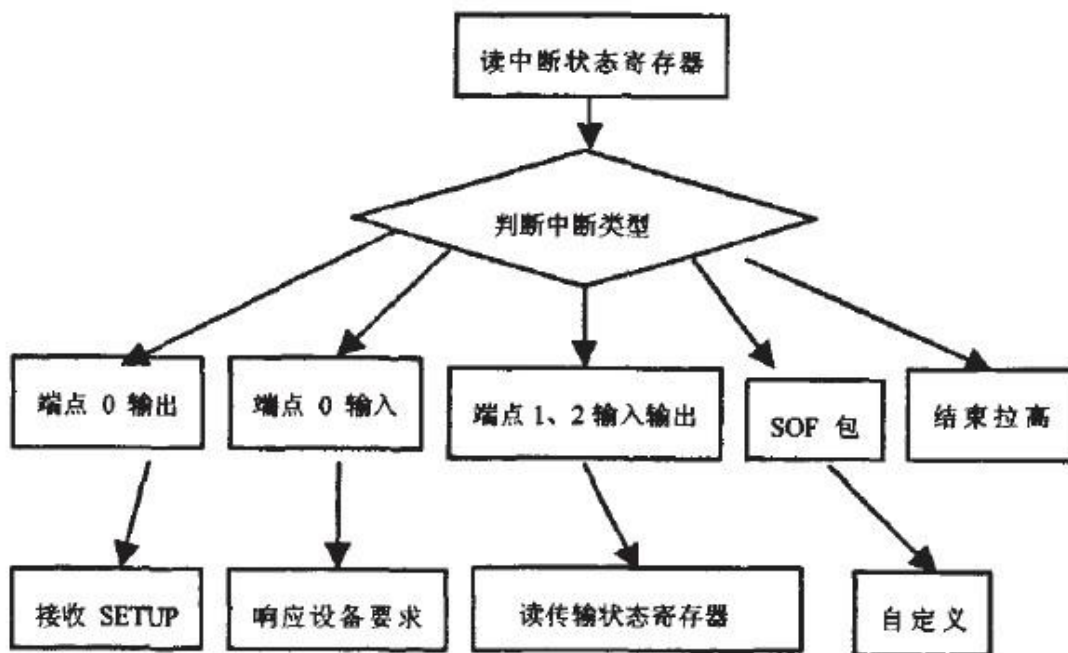


图 8 中断处理程序流程

本系统设备中，固件程序只须处理端点 0 的输出和输入要求就可以完成设备的正常配置和基本工作。

在图 8 中，SOF 包处理程序是自定义的部分，是为了更好地配合设备工作而添加的代码，与设备的实质没有太大关系，而结束状态改变引起的中断在这里不需要做处理。由于 USB 的数据传输都是由 D12 自动完成，端点 1 和 2 的输出/输入中断处理就没有什么实质内容。

4 结束语

本设计所实现的基于 USB 的无线触摸鼠标板除了具有无线空中鼠标的所以功能外，且成本低廉，容易实现，无线传输距离远（20 m），无指向性，适应性强，能广泛应用于现有的多媒体教室。