

**创新创业大作业(答辩报告)**

**空中飞鼠**

**Flying Mouse**

|  |  |
| --- | --- |
| **学　　院** | 光电信息与计算机工程学院 |
| **专　　业** | XXXXXX |
| **学生姓名** | XXXXX |
| **学　　号** | XXXXXXXX |
| **指导教师** | XXXXX |
| **完成日期** | 2018.12 |

**目录**

[1. 课题创意 1](#_Toc530401713)

[1.1 课题背景 1](#_Toc530401714)

[1.2 创作意图 1](#_Toc530401715)

[2. 方案论证 2](#_Toc530401716)

[2.1 方案选择 2](#_Toc530401717)

[2.1.1 2.4GHz 2](#_Toc530401718)

[2.1.2 蓝牙无线 2](#_Toc530401719)

[2.2 方案比较 2](#_Toc530401720)

[3. 课题设计与实现 3](#_Toc530401721)

[3.1 硬件设计 3](#_Toc530401722)

[3.1.1 硬件构成 3](#_Toc530401723)

[3.1.2 PCB板设计 5](#_Toc530401724)

[3.1.3 关键部分电路设计 5](#_Toc530401725)

[3.2 软件部分 9](#_Toc530401726)

[3.2.1软件架构描述 9](#_Toc530401727)

[3.2.2软件执行流程 9](#_Toc530401728)

[3.2.3 软件关键技术及详细功能说明 11](#_Toc530401729)

[4. 调试 14](#_Toc530401730)

[5. 心得体会 15](#_Toc530401731)

[附录 16](#_Toc530401732)

[附录1 实物图 16](#_Toc530401733)

# 1. 课题创意

1.1 课题背景

空中飞鼠是一种概念性的鼠标，顾名思义就是可以在悬空中使用的鼠标。又名空中鼠标，空鼠，飞鼠。空中飞鼠内置了陀螺仪传感器，可以感知方向和速度变化，只要通过在空中挥动鼠标，就可以控制电脑的光标移动，不需要传统的鼠标，大大的解放了双手，增加了使用乐趣。

市场上最新技术的空中飞鼠接近于光学鼠的精度，3档DPI可调（低中高）高速移动控制，可以用于对速度要求极高的应用，例如：水果忍者。智能消抖动算法能根据用户的操作自动输出适当坐标。让用户无法体会到手抖对坐标控制的严重影响。支持市面上流行的所有操作系统，用户只需插上USB接收器就可以开始使用A2。与传统鼠标完全兼容的左右键及仿真滚轮键，可以模拟标准滚轮的上下滚动。还提供音量加减、静音、关闭程序、关机、切回桌面及打开浏览器主页的多媒体及上网应用快捷键。

1.2 创作意图

由于技术与资金原因，我们打算实现一个实现基本操作功能的空中鼠标，我们的空中飞鼠设计采用 STM32F103RCT6 作为主控芯片，以陀螺仪（MPU6050）作为位置传感器，通过 NRF24l01无线模块进行数据传输。飞鼠整体分为发射模块和接收模块两部分，发射模块集成 NRF 和 MPU6050，两个微动开关模拟鼠标左右键，并且添加了鼠标编码器和一个微动开关以模拟鼠标的滚轮和滚轮的按压效果，另有一个按键实现键盘 Enter 键效果，发射板底部集成两节5号锂电池座以支持手持。接收端由一块主控芯片和 NRF 组成，通过一个A型 USB 接口和 PC端连接，接收端经 NRF 获得发射端传来的位置信息，将其转化为坐标变换量，通过 USB 的 HID 类协议将信息发给PC。为了前期调试方便，我们使用 J-link 仿真器的 SWD 接口对 STM32 进行下载和调试。在板子设计方面，考虑到经济性，我们发射和接收模块的印刷电路板（PCB）设计相同，这样只需向工厂发一套PCB就能实现两种功能。

# 2. 方案论证

2.1 方案选择

目前市场上无线鼠标的通信方式主要有 27MHz、2.4GHz 和蓝牙无线等，在飞鼠制作期间，我们主要对比了以上 2.4GHz 和蓝牙无线方式，并分别对其优缺点进行论证如下。

2.1.1 2.4GHz

2.4GHz无线模块（2.4Ghz RF transceiver / receiver module）工作在全球免申请 ISM 频道 2400M-2483M 范围内，具有实现开机自动扫频功能，共有50个工作信道，可以同时供50个用户在同一场合同时工作，无需使用者人工协调、配置信道。同时，可以根据成本考虑，选择50米内、150米、600 米多种类型无线模块。接收单元和遥控器单元具有1键自动对码功能，数字地址编码，容量大，避免地址重复。

2.1.2 蓝牙无线

蓝牙模块（英文：Bluetooth）无线蓝牙的工作方式和2.4GHz相同，采用的也是2.4G RF一致的频率段：在大多数国家免费、无授权的 2.4-2.485 GHz ISM。但其最大的特点是，只需电脑本身内置有蓝牙模块即可使用，比普通的2.4GHz要节省一个接收器，而且还多了自适应调频技术，不必像2.4GHz 鼠标那样需要对码，在便携性上更加出色，当然价格也更高，但是无线蓝牙模块的续航能力与2.4GHz相比较弱。

2.2 方案比较

2.4GHz 无线模块传输能力强，配置简单，可选择不同价位。蓝牙模块虽然比2.4GHz 无线模块少一个外置接收器，但价格更高，工作范围和续航也不如2.4GHz 无线模块。最后我们出于设计，价格，实用性，续航方面的综合考虑选择2.4GHz无线模块。

# 3. 课题设计与实现

3.1 硬件设计

3.1.1 硬件构成

1) 电源模块

该模块采用 AMS1117-3.3稳压单元，输入端VCC=5v由电池供电，输出端3.3v 供给MCU、MPU6050、NRF 等。

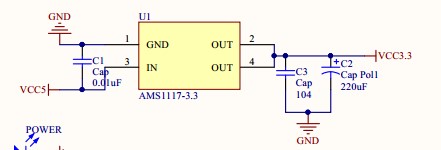


图3 电源模块

2）MCU模块

主控芯片选择 STM32F103RCT6,使用该 MCU 的 SPI1、IIC1、TIM4 等模块，外接 8MHz 晶振，提供系统时钟，此外使用 PC6-9 普通 IO 口作为按键接口。

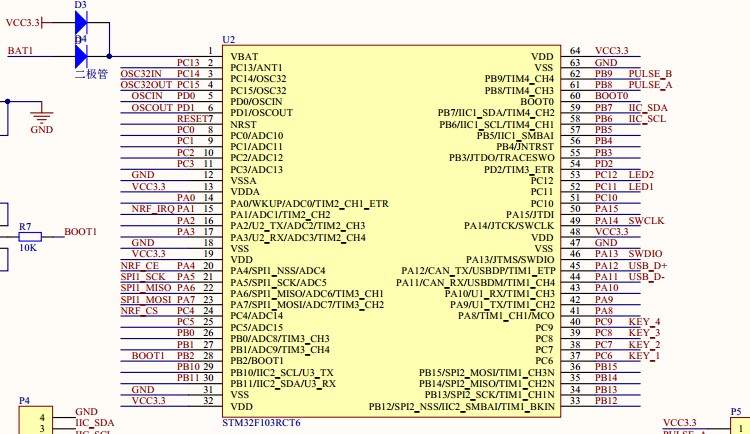


图4 MCU模块

3）仿真下载接口

仿真调试接口我们选用了占用引脚和空间最小的SWD接口，这种接口只需2根信号线即完成下载调试，并且效果稳定，对于开发空间有限的设计来说无疑是最好的选择。 接收端采用将发射端的模块上添加了 USB 接口电路，这样只需在发射板上焊接USB 接口即可，无需单独再制作发射板，这样设计是出于经济考虑。

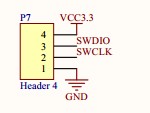


图5 仿真下载接口

4）USB接口电路

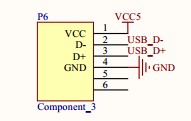


图6 USB接口电路

3.1.2 PCB板设计

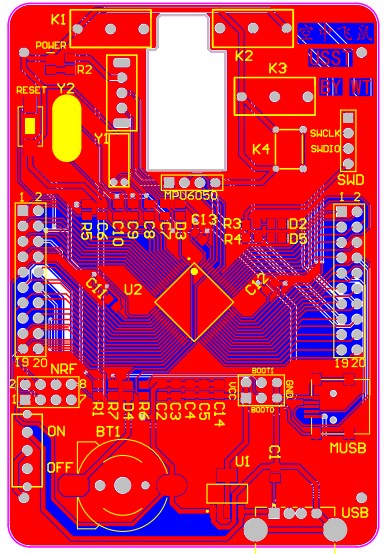


图7 PCB板设计

在 PCB 板的设计方面除了元器件的合理布局，还要考虑 PCB 板的手持舒适性，最终我们采用的板型如上，整体长方形6x9cm，四周圆弧倒角以减少手握时的硌手感，上部开了一个长方形的凹槽，以方便鼠标滚轮安装，USB 接口设计在板子最边缘，以便不影响发射板的美观，而且在接收板上也更方便插进PC接口，电池座设计在板子下方，符合外观设计。

3.1.3 关键部分电路设计

1）USB电路设计

在电路设计中，USB 电路连接并不是直接连接在 MCU 的 D+和 D-，主机检测设备接入时，是通过判断 D+和 D-信号线压差实现的，D+上拉至高电平时，主机识别设备为全速模式，D-上拉高电平时，主机识别设备为低速模式。由于我们的飞鼠工作在全速模式，所以选择 D+上拉。在此可以选择将 D+直接通过 1.5K 电阻上拉至 3.3v，也可以通过一个三极管，经MCU一个普通 IO 口控制将 D+上拉至高电平。显然，后者的实现更可控和灵活，但是出于电路简洁和避免不必要的麻烦，我们还是选择了前者。

2) 程序与硬件接口

（1）MPU6050 ：MPU6050 模块是通过 IIC 接口与 MCU 相连。其内部整合了 3 轴陀螺仪和 3 轴加速度传感器，并且含有一个 IIC接口，可用于连接外部磁力传感器，并利用自带的数字运动处理器（ DMP: Digital Motion Processor）硬件加速引擎，通过主 IIC接口，向应用端输出完整的 9 轴融合演算数据。有了 DMP，我们可以使用 InvenSense 公司提供的运动处理资料库，非常方便的实现姿态解算，降低了运动处理运算对操作系统的负荷，同时大大降低了开发难度。

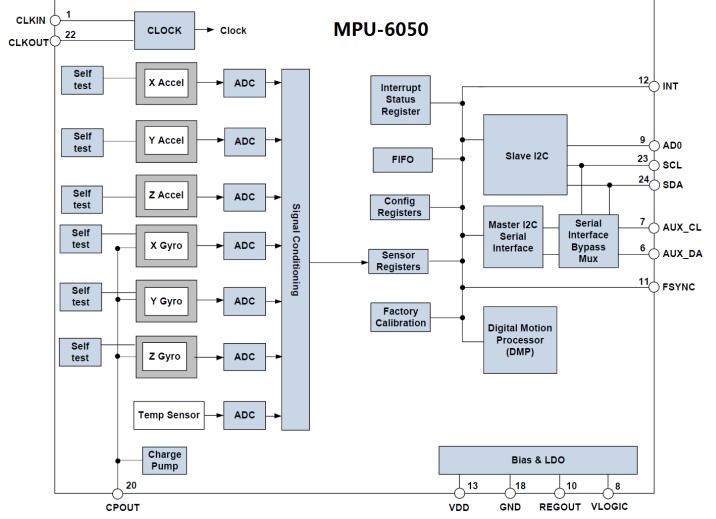


图8 MPU6050

（2）初始化 IIC 接口：MPU6050 采用 IIC 与 STM32F1 通信，所以我们需要先初始化与 MPU6050 连接的 SDA 和 SCL 串行数据线。

（3）复位 MPU6050：这一步让 MPU6050 内部所有寄存器恢复默认值，通过对电源管理寄存器 1（0X6B）的 bit7 。写 1 实现。 复位后， 电源管理寄存器 1 恢复默认值(0X40)，然后必须设置该寄存器 0X00，以唤醒 MPU6050，进入正常工作状态。

（4）设置角速度传感器（陀螺仪）和加速度传感器的满量程范围：这一步，我们设置两个传感器的满量程范围(FSR)，分别通过陀螺仪配置寄存器（0X1B）和加速度传感器配置寄存器（0X1C）设置。我们一般设置陀螺仪的满量程范围±2000dps。速度传感器的满量程范围为±2g。

（5）SPI传输方式：NRF24L01 通过 SPI 方式与 MCU 进行数据传输，其管脚图如下图9，根据管脚图了解各管脚传输时序图如图10：

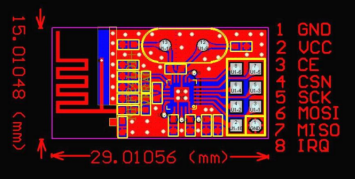


图9 NRF24L01管脚图

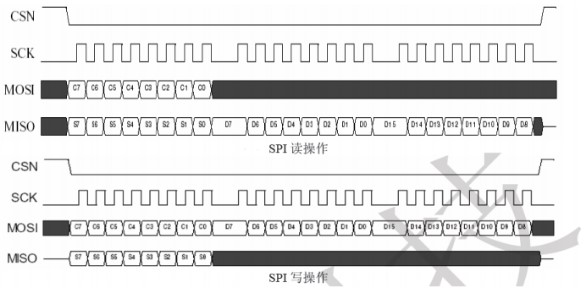


图10 NRF24L01读写操作时序

根据时序图进行程序配置，上图中 Cn 代表指令位，Sn 代表状态寄存器位，Dn 代表数据位。从图中可以看出，SCK 空闲的时候是低电平的，而数据在 SCK 的上升沿被读写。所以，我们需要设置 SPI 的 CPOL 和 CPHA 均为0，来满足 NRF24L01 对 SPI 操作的要求。

3） USB模块

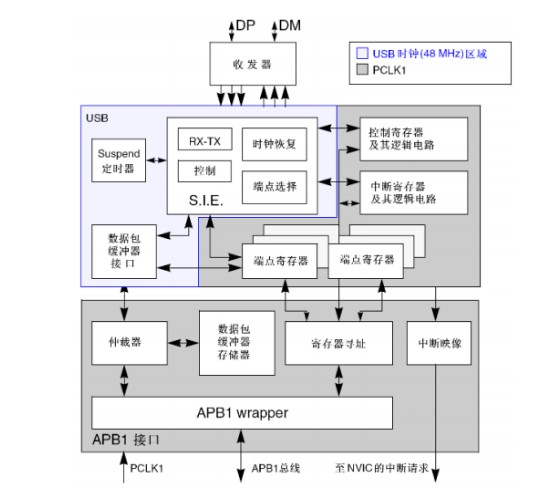


图11 USB 设备框图

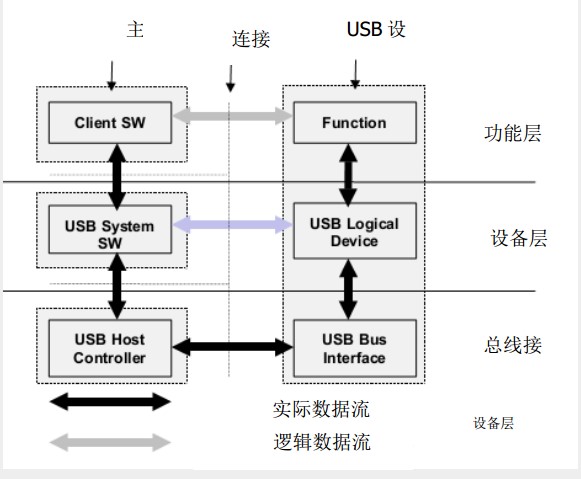


图12 USB系统架构

STM32F103 系列的 USB 控制器外设(不含 OTG 功能)可用于制作 USB 设备。其主要特性如下：

（1）符合 USB2.0 全速设备的技术规范，即速度为 12Mb/s 。

（2）可配置1到8个 USB 端点，其中的8个端点均为双向端点，若配置的都是单向端点，则可配置 16 个。

（3）CRC(循环冗余校验)生成/ 校验，反向不归零(NRZI)编码/ 解码和位填充。

支持同步传输。

（4）支持批量/ 同步端点的双缓冲区机制 ，为实现 USB 存储器提供了良好的持。

（5）支持 USB 挂起/ 恢复操作 ，这有利于降低功耗。

3.2 软件部分

3.2.1软件架构描述

1）发射端

程序初始化完成后，以IIC通讯方式获取MPU6050中的姿态数据(x,y),按键检测模块实时检测按键状态，姿态数据和按键状态组合成一个数据包通过SPI通讯方式经NRF24L01发射至接收端。

2）接收端

接收端系统初始化完成之后，进行USB初始化，配置USB为HID类，接着微控制单元MCU通过SPI通讯方式经NRF24L01接收来自发射端的数据包，并将其提取出坐标信息和按键信息，通过USB发送函数将坐标信息发送至PC。

3.2.2软件执行流程

1）发送端流程图

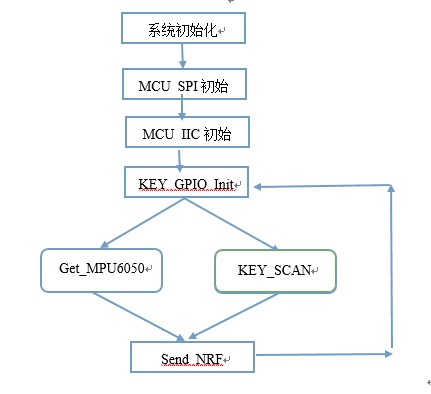


图1 发送端流程图

2）接收端流程图

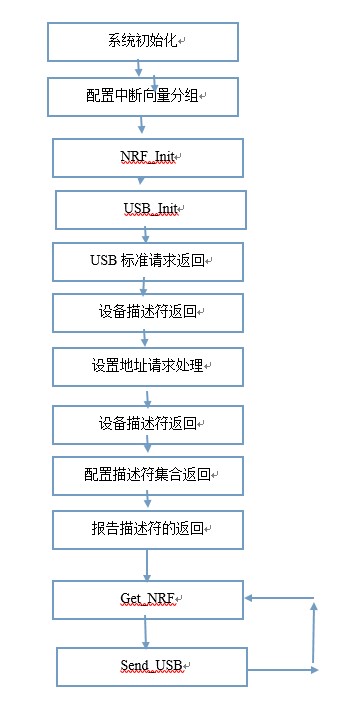


图2 接收端流程图

3.2.3 软件关键技术及详细功能说明

1）主机集线器检测到新设备

主机集线器监视着每个端口的信号电压，当有新设备接入时便可察觉。（集线器端口的两根信号线的每一根都有15kΩ的下拉电阻，而每一个设备在D+都有一个1.5kΩ 的上拉电阻。当用USB线将PC和设备接通后，设备的上拉电阻使信号线的电位升高，因此被主机集线器检测到。

2）主机发送 Get\_Status 请求

每个集线器用中断传输来报告在集线器上的事件。当主机知道了这个事件，它给集线器发送一个 Get\_Status 请求来了解更多的消息。返回的消息告诉主机一个设备是什么时候连接的。

3）主机发送 Set\_Feature 请求，集线器重启端口

当主机知道有一个新的设备时，主机给集线器发送一个 Set\_Feature 请求，请求集线器来重启端口。集线器使得设备的USB数据线处于重启（RESET）状态至少10ms。

4）集线器在设备和主机之间建立一个信号通路

主机发送一个 Get\_Status 请求来验证设备是否激起重启状态。返回的数据有一位表示设备仍然处于重启状态。当集线器释放了重启状态，设备就处于默认状态了，设备已经准备好通过 Endpoint 0 的默认流程响应控制传输，即设备现在使用默认地址0x0与主机通信。

5）集线器检测设备速度

集线器通过测定哪根信号线（D+或D-）在空闲时有更高的电压来检测设备是低速设备还是全速设备。（全速和高速设备D+有上拉电阻，低速设备D-有上拉电阻）。

6）获取最大数据包长度

PC 向 address 0发送USB协议规定的 Get\_Device\_Descriptor 命令，以取得缺省控制管道所支持的最大数据包长度，并在有限的时间内等待 USB 设备的响应。该长度包含在设备描述符的 bMaxPacketSize0 字段中，其地址偏移量为 7，所以这时主机只需读取该描述符的前8个字节。注意，主机一次只能枚举一个USB设备，所以同一时刻只能有一个USB设备使用缺省地址0。

7）主机分配一个新的地址给设备

主机通过发送一个 Set\_Address 请求来分配一个唯一的地址给设备。设备读取这个请求，返回一个确认，并保存新的地址。从此开始所有通信都使用这个新地址。

8）主机重新发送 Get\_Device\_Descriptor 命令，读取完整设备描述符

主机向新地址重新发送 Get\_Device\_Descriptor 命令，此次读取其设备描述符的全部字段，以了解该设备的总体信息，如VID，PID。

9）主机发送 Get\_Device\_Configuration 命令，获取完整配置信息

主机以上述方式向设备循环发送 Get\_Device\_Configuration 命令，要求USB设备回答，以读取全部配置信息。

10）主机发送 Get\_Device\_String 命令，获得描述字符集（unicode）

描述字符集包括了产商、产品描述、型号等信息。

11）主机展示新设备信息

此时主机将会弹出窗口，展示发现新设备的信息，产商、产品描述、型号等。

12）PC判断能否提供该类USB的驱动

根据 Device\_Descriptor 和 Device\_Configuration 应答，PC 判断是否能够提供USB 的Driver，一般win2k能提供几大类的设备，如游戏操作杆、存储、打印机、扫描仪等，操作就在后台运行。但是Win98却不可以，所以在此时将会弹出对话框，索要USB 的 Driver。

13）主机发送 Set\_Configuration（x）命令，请求为设备选择一个配置

加载了USB设备驱动以后，主机发送 Set\_Configuration（x）命令请求为该设备选择一个合适的配置(x代表非0的配置值)。如果配置成功，USB设备进入“配置”状态，并可以和客户软件进行数据传输。

通过STM89C51检测红外对管的反馈

# 4. 调试

由于 PCB 发板需要一定周期，在制作前期我们采用在 STM32 开发板上调试的方式，在制作过程中我们觉得最困难的部分是 USB 与 PC 通信的建立过程，所以我们最先选择了调试接收板。在程序中添加串口发送函数，这样我们便能将程序执行过程中的关键数据通过串口发送到上位机进行查看。例如，在 USB 的枚举过程中，我们可以将 PC 主机发送的描述符请求转发到上位机，以此得知主机正在请求哪种描述符，并根据发送的状态判断枚举错误出现在哪一步，方便后期修改。

在 MPU6050 数据读取方面我们也是添加串口发送程序，将采集的坐标数据发送到上位机，这样可以方便修改姿态处理函数的参数。在调试 NRF 模块时，我们从发射板经 NRF 发送已知的数据包，在接收板上经 NRF 接收数据包，并将数据包发送到上位机，如果数据包无误则说明 NRF 无线通信正常，便可进行后面的程序整合。

在线调试：由于我们采用了 SWD 调试接口，所以可以很方便的进行在线调试，下面介绍采用在线调试发现并解决问题的实例。在我们把 PCB 板焊接完成之后，发现同一个程序在开发板上可以正确运行，而在飞鼠板上就是会出问题，PC 一直提示无法识别的设备，这说明枚举过程失败了。我们在线调试发现程序在进入 USB 发送数据程序是进入了硬件异常中断，卡在中断死循环中。经过一段时间分析，我们怀疑是硬件晶振问题，于是在检查过程发现晶振的接地端电容太大，更换为 22pF 之后一切才恢复正常。

# 5. 心得体会

经过本次的创新设计课程，我学习到了很多以前未曾接触过的知识，比如陀螺仪MPU6050这一集成芯片的通信传感，STM32单片机芯片的各种优势，明白以前所接触的51单片机的很多局限性，更接触到了USB通信协议，了解到原来USB并非简单的一个插口，而是有很复杂的通信协议，程序等等。从刚开始了解VR所用的手持式鼠标，开始有做空中飞鼠的想法，到跟着组长一步步实践下来。看着两块PCB逐渐成型，内容日益丰富，我们经过不断查错，改进，终于得到目前的空中飞鼠。虽然它仍有缺陷，比如设计舒适性不高，即使有受持支撑，但由于没有合适的设计外壳，应用起来影响流畅度，另外由于设计PCB时考虑欠佳，后来用了很久才发现上拉电阻未能安置，因此利用飞插焊接，影响了美观，预计还会影响稳定性和使用寿命。

总之，通过这次课程，让我学到了很多知识，与同学之间的团队合作也尤为重要，查找资料整合信息的能力也有所提高，更重要的是让我认识到STM32，陀螺仪，USB等的魅力所在，激起了我想要深入学习的欲望，受益匪浅。

# 附录

附录1 实物图

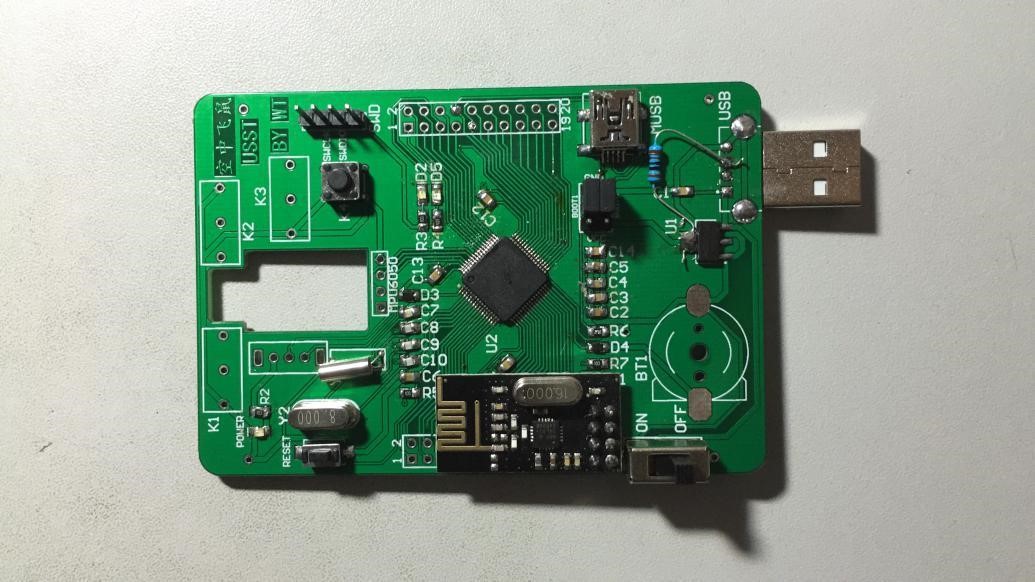


图12 接收端（正）

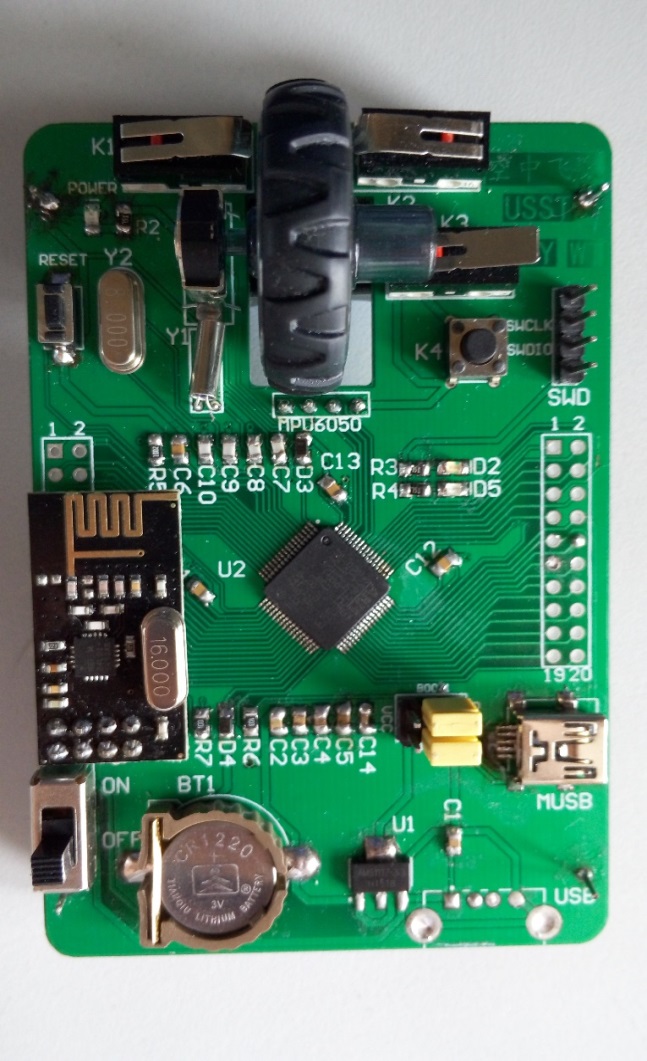


图14发射端（正）



图15 发射端（反）