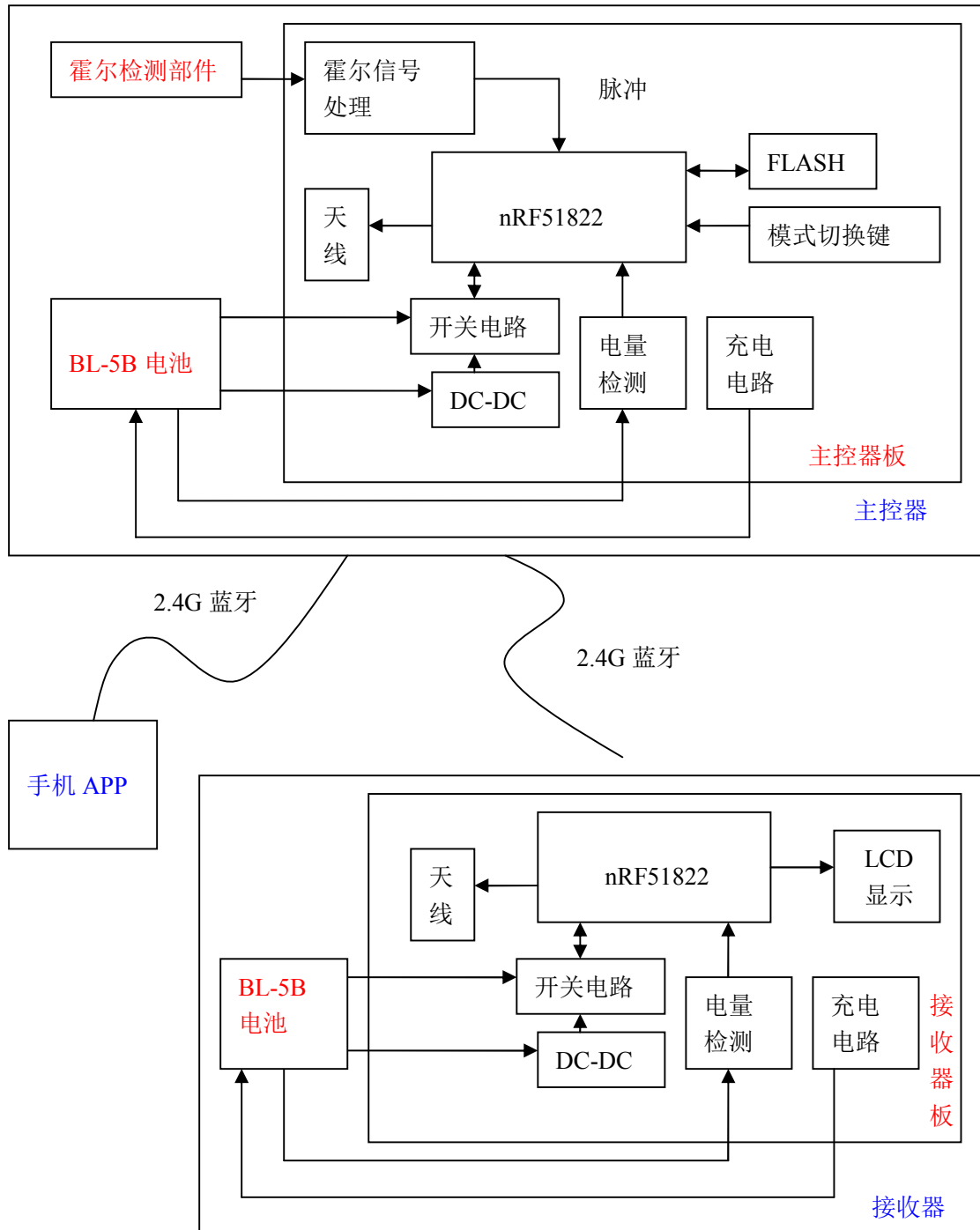


功能需求书

1、硬件设计

1.1 硬件架构框图



1.2 细节要求

霍尔检测部件： 霍尔检测电路板上有霍尔检测元件和耳机

座。耳机座作为霍尔检测部件的电源和信号接口。

霍尔信号处理：从霍尔部件输入模拟信号到主控器电路板上的霍尔信号处理电路，经过霍尔处理电路处理后输出脉冲信号到 nFR51822。每次脉冲促使 nFR51822 产生一次外部中断。最高脉冲频率为 1khz。

主控器充电电路：对 BL-5B 电池充电；Micro usb 接口；5V 充电输入。

接收器充电电路：对 BL-5B 锂电池充电。

电量检测：检测电池的剩余电量。通 nRF51822 的 ADC 转换实现。

开关电路：实现轻触按钮开关机。不采用开/断路方式开关机。

Flash：采用 32Mbit(4Mbyte)的 Flash。

模式切换键：主控器正常情况下做为 BLE 主机，手机 APP 和接收器作为从机；当使用者需要用手机连接主控器且主控器没有记录使用者手机的 uuid 的时候，有必要让主控器工作于从机状态、手机处于主机状态；

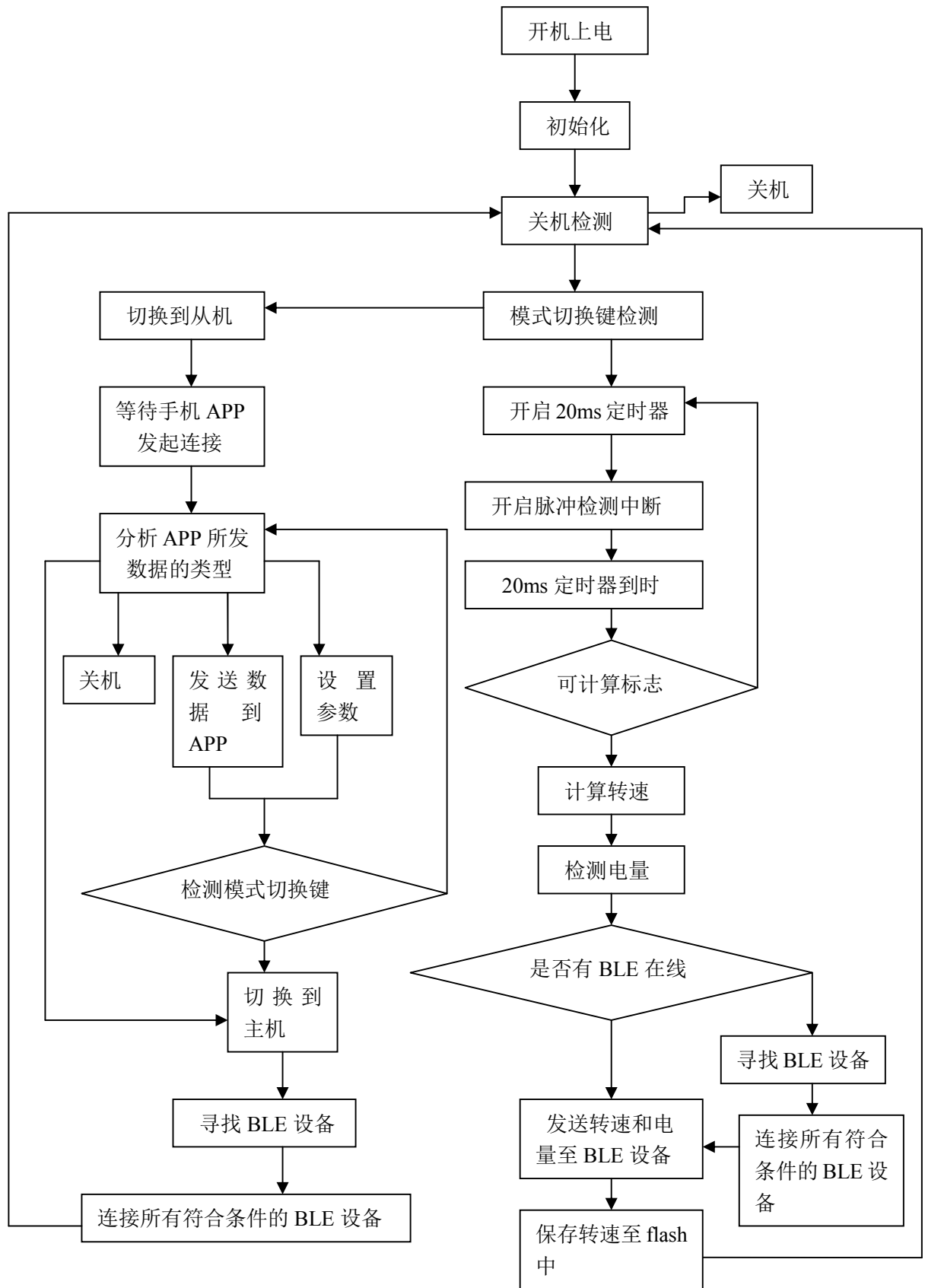
其他：板载天线；最高功耗<20ma；外形尺寸见附图。

2、软件设计

2.1 主控器软件

主控器软件的功能包括初始化、累计脉冲、计算转速和速度、发送速度到手机 APP 和接收器、将数据保存至 Flash、模式切换、设置参数和读取数据、电量检测、关机。

2.1.1 主控器软件流程参考图



描述数据结构： 涉及到的参数有转盘周长、已注册的 BLE

设备、在线 BLE 设备、每次中断的时间、转速、电量。

1、用一个结构体来描述圆盘周长、已经注册的 BLE 设备、在线 BLE 设备、转速和电量。

Typedef struct

```
{  
    Double perimeter;//周长  
    Double speed;//转速  
    Int Power_Remain;//剩余电量  
    Int OnlineBle_num;//在线设备的数量  
    Int RegisterBle_num;//已注册 BLE 设备的数量  
    Int isReady2caculate;//可以计算速度 标志  
} MainCtrl_Des
```

MainCtrl_Des Main_Ctrl;

Type struct

```
{  
    Unsigned char name[30];//BLE 名称  
    Unsigned char uuid[100];//uuid  
} Ble_Des;
```

Ble_Des Regsiter_BLE_Dev[4];//已注册的 BLE 设备，最多四个 BLE 设备，其中一个接收器，最多三个手机与主控器连接。

Ble_Des Online_BLE_Dev[4];//在线 BLE 设备，最多四个 BLE 设备，其中一个接收器，最多三个手机与主控器连接。

2、随着转盘转动，霍尔检测元件输出不规整的模拟脉冲信号，经过霍尔信号处理电路后产生边沿陡峭的脉冲信号。每次脉冲促使 nFR51822 产生一次中断。在中断服务程序中，将产生中断的时间存入栈。

Typedef struct

```
{  
    Int counter;//定时器溢出的次数  
    Int timer_data;//定时器中的数据  
} time;
```

Time 数据结构由两部分组成，一部分是 counter，另一部分是 timer_data。采用这种数据结构，主要为了解决 16 位定时器所定时间太短。

```
Int timer_count;//定时器溢出的次数
```

```
Time time_stack[50];//保存一定时间内没冲发生的时间
```

3、根据一定的算法(见可计算标志)判断是否可以对 time_stack 堆栈中的数据进行分析和计算速度，分析和计算速度的算法见计算速度。将计算好的速度发送到在线的 BLE 设备(手机 APP 和接收器)显示，同时保存速度至 Flash 中。保存在 Flash 中速度的格式为：

```
Typedef Struct
```

```
{  
    Double speed;
```

Int time;//当前时间

} time2flash;

Time2flash[20];在 ram 中设 20 个缓存，当缓存满 20 个时保存至 flash 中，并清空缓存重新使用缓存记录。

此数据结构用来按时间记录速度。

Flash 存储器分配：

Double perimeter;//1-4Bytes

Unsigned char RegisterBle_num;//5th Byte 已注册 BLE 设备的数量

Ble_Des Regsiter_BLE_Dev[4];//6-525

Time2flash; 存储速度 526 byte---END

初始化：1、完成板子配置(UART、GPIO 等); 2、读取 flash 中保存的参数，填充 Main_Ctrl 和 Regsiter_BLE_Dev[4];

关机检测：判断对应开关机按键的相关的 GPIO 脚是否有关机信号。

关机：检测到关机按键信号或者接受到手机 APP 传来的关机命令后进行关机。关机的流程如下：1、保存参数；2、保存速度；3、通知在线的 BLE 设备关机；4、设置相关 IO 脚来断电。

模式切换检测：判断对应模式切换按键的相关的 GPIO 脚是否有信号。

切换到从机、等待手机 APP 连接：对于已经注册的手机，需

要实现的是——当手机靠近主控器时，主控器与手机自动连接。注册是指主控器记录和保存手机 BLE 的 uuid 和名称。由于各个手机 BLE 的名称和 uuid 没有规律，所以主控器不能主动连接一台没有注册过的手机 BLE。需要将主控器切换到从机，通过手机 APP 寻找并主动发起与主控器的连接，连接成功后，手机 APP 发送此手机的 BLE 名称和 uuid 给主控器进行注册。手机在主控器注册成功以后，只要手机接近主控器，主控器主动寻找已注册的手机 BLE 并主动对手机 BLE 发起连接。

分析手机 APP 所发送的数据类型：共有 5 条指令，包括关机指令、模式切换指令、读取数据、设置参数、删除 Flash 中的速度值。

- 1、 关机指令数据格式 0x01(1bytes);
- 2、 模式切换指令数据格式 0x02(1bytes);
- 3、 设置参数指令 0x03(1byte) perimeter(2-5bytes);
- 4、 查询速度值的存储情况; 0x04(1bytes);
- 5、 读取速度值指令 0x05(1byte);
- 6、 删除 Flash 中的速度值 0x06(1byte)

发送数据到手机 APP：手机 APP 向主控器发送查询速度值存储情况指令后，主控器返回给手机 APP 的数据格式：

Num, term_1,term_2,term_s,……,term_Num。Num 为所有项数，term 的结构为 struct{

Int Start_time;//起始时间

Int End_time;//结束时间

}。将时间上连续的一串速度值视作一项。若转盘停止转动 10 秒钟以上，启动时视作新的一项速度值开始记录。

当接收到读取数据指令后，主控器将保存在 flash 中的速度值发送给手机 APP。手机 APP 以此数据画曲线，横轴为时间，纵轴为速度。

设置参数：接收到参数设置指令时候，更新 Main_Ctrl.Perimeter，并保存至 flash 中。

切换到主机：

寻找所有 BLE 设备：寻找是否有已注册的 BLE 设备。

连接符合条件的 BLE 设备：若发现附近有已注册的 BLE 设备，则发起连接。

20ms 定时器：过完 20ms 后，检查 time_stack 栈；若 time_stack 栈内有至少两个数据则 iSready=1。若 time_stack 栈中没有至少两个数据，则继续 20ms 定时。当进行 10 次 20ms 定时后，仍没有至少两个数据，则认为转盘停止转动。此时保存一项速度。

脉冲中断：保存数据点的格式。

可计算标志：iSready 变量标识是否可以计算速度。

计算速度：计算转盘的速度，计算公式： $[(\text{Main_Ctrl.perimeter}/4)*\text{num}]/(\text{start_time}-\text{end_time})$ 。其中 num 为 20ms 内脉冲的

总个数，start_time 为产生第一个脉冲的时间，end_time 为 20ms 内产生最后一个脉冲的时间。

计算电量：直接用 ADC 采集到的数据，无需计算。

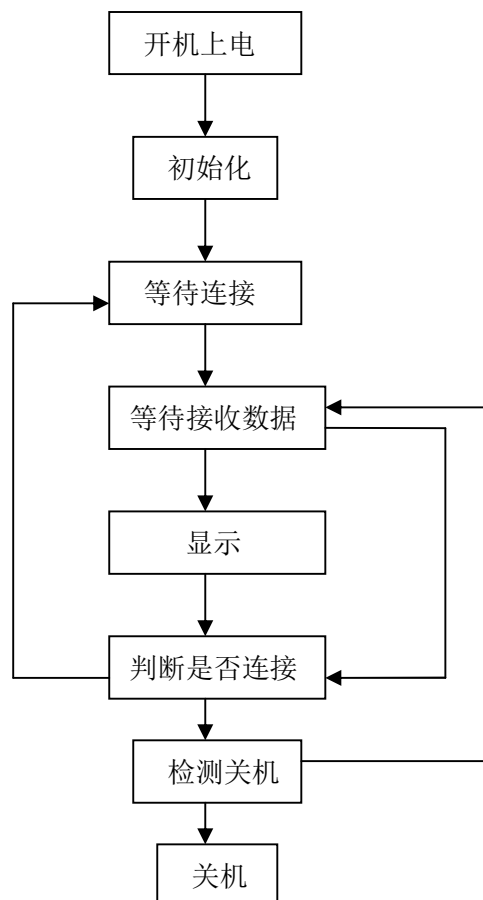
发送速度和电量至 BLE 设备：数据格式 0x07(1Byte), power_remain(2-3byte),speed(4-7byte)

保存转速至 FLASH 中：

2.2 接收器软件

接收器主要的功能是接收主控器发来的速度值和电量值、检测自身电量；显示速度和电量。

2.2.1 接收器参考流程图



初始化：

等待连接：

等待接收数据：

显示：

判断是否已连接：

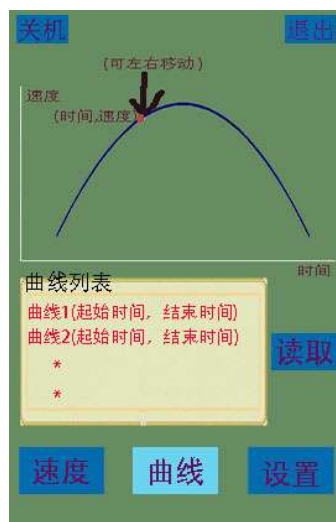
检测关机：

2.3 手机 APP

手机 APP 主要有几个功能：设置参数、显示速度、速度曲线、软件版本自动升级。设置参数、显示速度和速度曲线对应三个界面。



速度显示界面

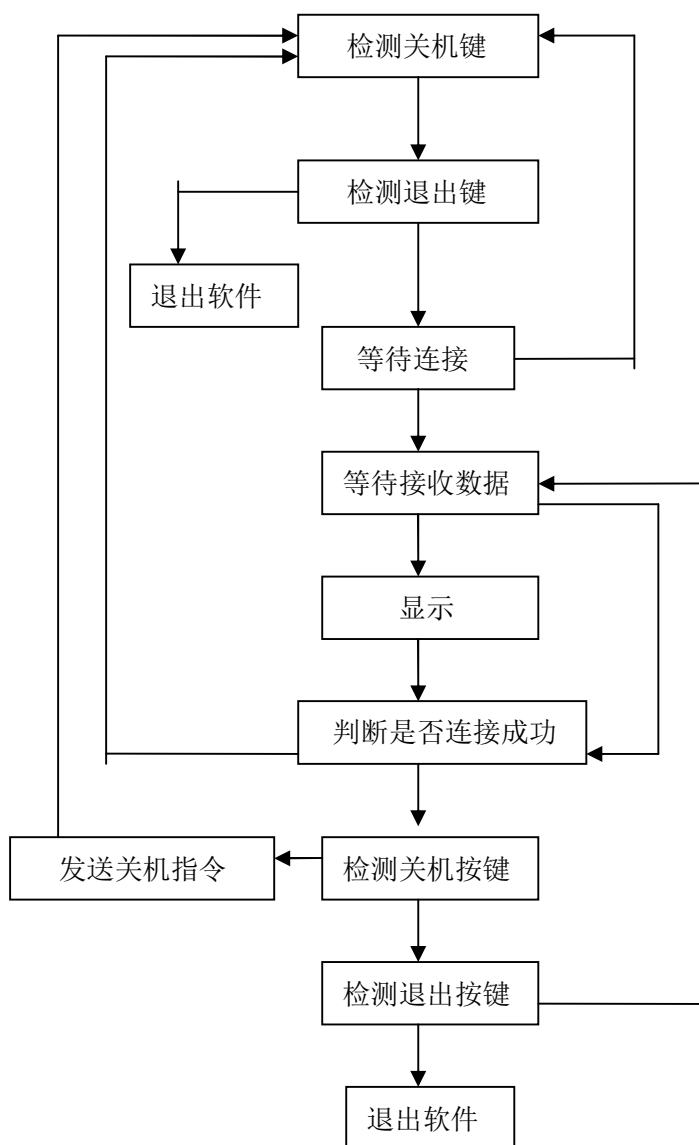


速度曲线界面



注册与设置参数

速度显示：按“速度”按钮切换至速度显示界面。当手机接近主控器时，主控器主动对手机 BLE 发起连接；手机 BLE 与主控器连接成功后，主控器不断向手机 APP 发送速度值。主控器接收并显示速度值。流程参考图如下：



速度曲线界面：1、切换到“速度曲线界面”后，点击“读取”按钮向主控器发送读取“速度曲线”请求。主控器接收到读取速度曲线指令后，将所有保存在 Flash 中的速度曲线数据发送给手机 APP。手机 APP 接收、保存所有曲线的数据。

2、手机 APP 通过曲线列表窗口显示所有的曲线的名称和时间信息，每一行显示一条曲线的信息(起始时间，结束时间)。使用者点击一条速度曲线信息时，直角坐标系绘制对应的速

度曲线。3、在控制器和手机 BLE 没有连接的情况下，点击“读取按钮”后弹出提示对话框——“与控制器失去连接，无法读取数据”。4、控制器与手机 BLE 有链接的情况下，读取超时后对话框提示——“读取超时”。

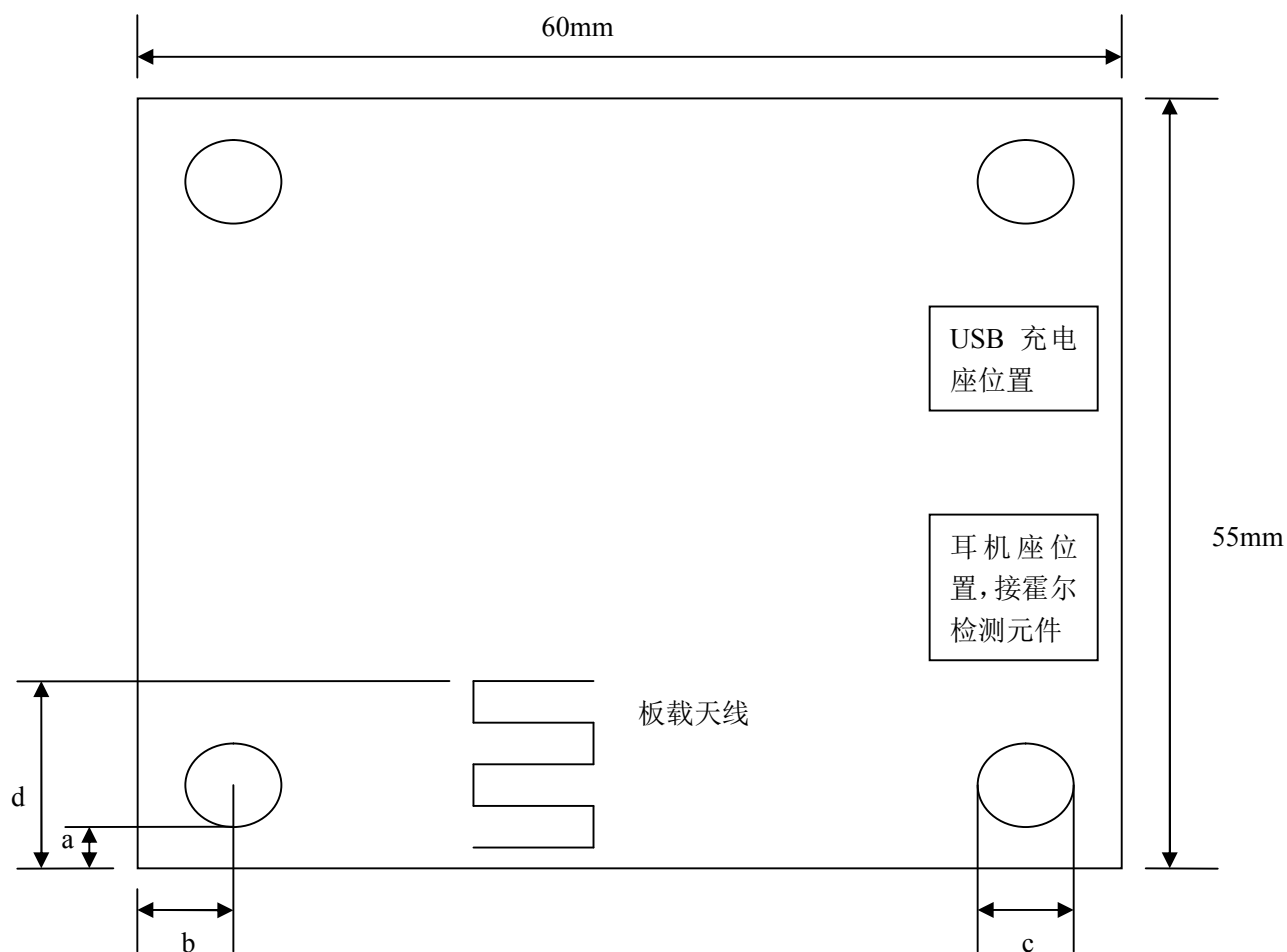
注册：1、注册是指控制器记录手机 BLE 的信息(BLE 名称和 uuid)。若控制器没有记录手机 BLE 的信息，当手机靠近控制器时，控制器无法主动与手机 BLE 连接。2、注册操作过程：第一步，打开手机 APP；第二步，按控制器模式切换键，使控制器切换至从机状态；第三步，点击“设置—搜索”，搜索到附近的 BLE 设备；第四步，在附近 BLE 设备列表中选择控制器的 BLE，点击“注册”。 点击注册后，手机 APP 向控制器发起连接，并把手机 BLE 的信息发给控制器。通讯协议由接包方设计。

参数设置：1、通过手机 APP 输入转盘的周长。手机工作在主机状态或者从机状态。2、点击“读取”按钮来读取保存在 Flash 中的周长值。若手机与控制器未连接，点击读取按钮后对话框提示——“手机与控制器未连接”。若读取成功，在显示框中显示周长。3、显示框中输入周长值，点击“设定”来设置周长参数。点击设定后，手机 APP 将参数发送给控制器，控制器将数据保存到 Flash 中。若手机与控制器未连接，则对话框提示——“手机与控制器未连接”。

软件升级功能：考虑到后续会发布新版本的软件，所以得有

软件升级功能。打开手机 APP 时，手机 APP 与服务器软件通讯。服务器软件同时手机 APP 是否有更新版本软件。若有更新版本软件，从服务器下载软件更新安装。

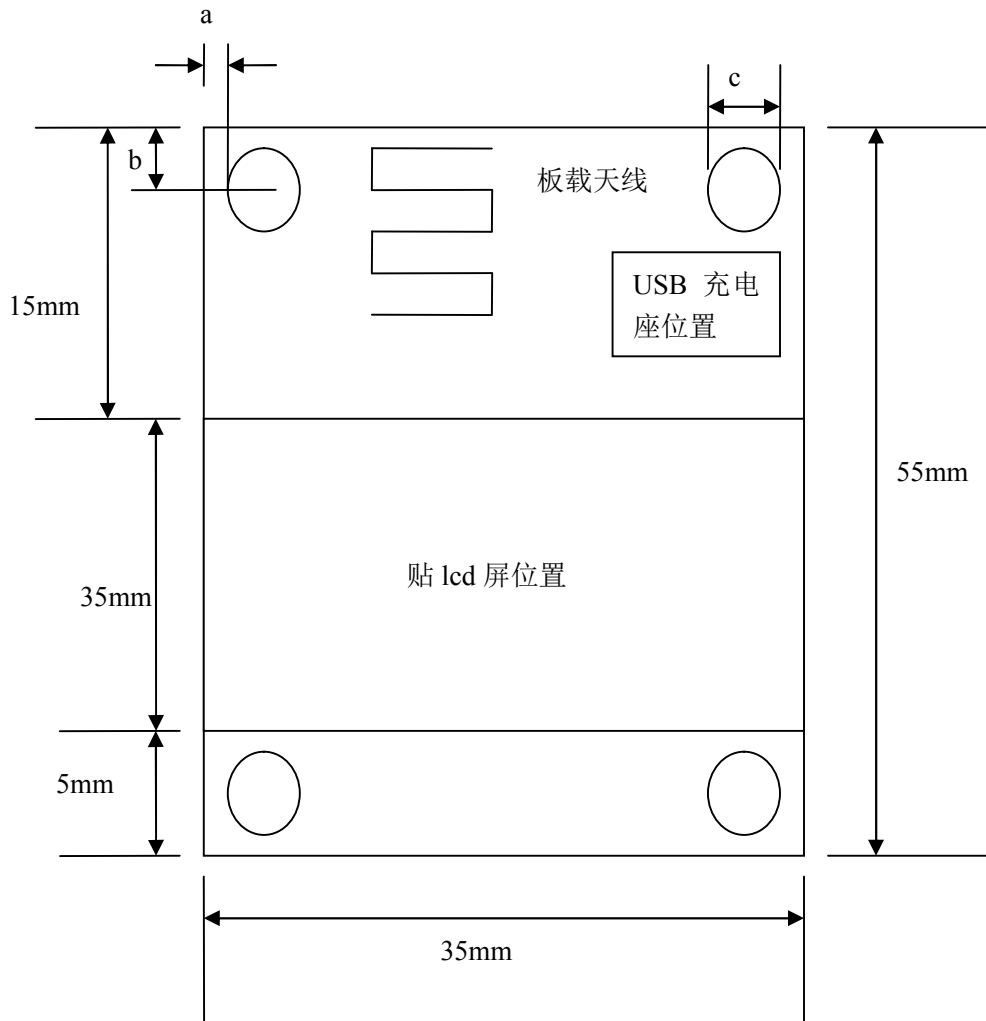
附图 1：主控器电路板尺寸



a:1mm b: 2.5mm c:3mm d:20mm

四个钻孔的边距相同(1mm)，直径相同(3mm)。

附图 2: 接收器电路板尺寸



a:1mm b: 2.5mm c:3mm

四个钻孔的边距相同(1mm)，直径相同(3mm)。

采用四层板。Lcd 屏位置区域的表层只走 lcd 显示器的信号线。

Lcd 屏位置区域的底层可以贴片和走线。