

# **MC1490**

---

---

**BLE 开发文档**

**Revision 0.72**

2018.11

## COPYRIGHT

THE INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF FOHEART CO., LTD. TRANSMITTING, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF THE CONTENT ARE FORBIDDEN WITHOUT PERMISSION. OFFENDERS WILL BE HELD LIABLE FOR PAYMENT OF DAMAGES. ALL RIGHTS ARE RESERVED IN THE EVENT OF A PATENT GRANT OR REGISTRATION OF A UTILITY MODEL OR DESIGN.

## GENERAL NOTES

FOHEART OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. FOHEART MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. FOHEART DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

**For technical support, please visit:**

<http://www.foheart.com/support.html> or

<http://www.foheart.com/question.html>

北京总部

Tel: (+86)010-56106165

Email: [contact@foheart.com](mailto:contact@foheart.com)

地址: 北京市海淀区黑山扈路红山口 8 号 D2-南-3 号

邮编: 100091



*Copyright © FOHEART Co., Ltd. 2015-2018. All rights reserved.*

## Revision History

Rev. No.	Date	Description	Author(s)
0.10	Sep. 9, 2017	<ul style="list-style-type: none"> <li>● First Draft</li> </ul>	HMX
0.20	Sep. 12, 2017	<ul style="list-style-type: none"> <li>● 修正了节点中发送 STARTRECORD 指令回复不正确的问题。</li> <li>● 修正了节点中发送 STOPRECORD 指令回复不正确的问题。</li> <li>● 修改了 CLRMAG 指令执行正确返回 0x01。</li> <li>● 2.1.1 增加记录频率参数。</li> <li>● 增加 2.7 节。</li> <li>● 2.1.2 节增加返回量。</li> <li>● 【硬件】增加了正在记录姿态时, LED 闪烁 1Hz 频率。</li> <li>● 2.3.2 增加 status 状态。</li> <li>● 2.4.2 增加 status 状态。</li> <li>● 增加 1.5 节, 电池剩余电量查询指令。</li> <li>● 2.5.1 增加上传频率参数。</li> <li>● 增加 2.8 2.9 2.10 节。</li> </ul>	HMX
0.30	Sep. 19, 2017	<ul style="list-style-type: none"> <li>● 修改了 2.8.1 节的数据结构。</li> <li>● 修改了 2.10 节的数据结构。</li> <li>● 添加了连接后的密码认证功能。</li> </ul>	HMX
0.40	Sep. 22, 2017	<ul style="list-style-type: none"> <li>● 修改了 2.9.2 的数据结构, 区分了姿态与位移。</li> </ul>	HMX
0.50	Sep. 27, 2017	<ul style="list-style-type: none"> <li>● 修改了底层硬件 bug。</li> </ul>	HMX
0.66	Oct., 2018	<ul style="list-style-type: none"> <li>● 修改了实时传输数据结构。</li> </ul>	HMX
0.68	Nov., 2018	<ul style="list-style-type: none"> <li>● 添加了第三章, 添加了每个指令的结构体参考。</li> </ul>	HMX
0.70	Nov., 2018	<ul style="list-style-type: none"> <li>● 添加了第四章、第五章、订货信息</li> </ul>	HMX
0.72	Nov., 2018	<ul style="list-style-type: none"> <li>● 添加了磁干扰指示</li> </ul>	HMX

## 名词解释：

**节点：**指的是硬件端，包括硬件及硬件上运行的嵌入式驱动程序。

**APP 或手机端：**指的是运行在 iOS 或者 Android 系统的手机客户端。

**BLE：**指的是低功耗蓝牙，为节点或者手机的前端射频部分。

**SPIFLASH：**指的是节点端的存储器，即使断电保存在上面的内容也不会丢失，用来存储运行过程中的姿态数据，存储为连续存储，当达到最大存储量时，会自动循环覆盖最早产生的数据。

**充电过程指示灯：**充电过程中呼吸灯闪烁，充满自动关机。

**四元数：**模值为 1，发送顺序按照 WXYZ 发送，坐标系定义见附录。

**欧拉角：**单位为 degree，发送顺序按照 XYZ 发送，坐标系定义见附录。

**加速度值：**单位为 g，发送顺序按照 XYZ 发送，坐标系定义见附录。

**角速度值：**单位为 degree/s，发送顺序按照 XYZ 发送，坐标系定义见附录。

**磁力计值：**单位为 mGauss，发送顺序按照 XYZ 发送，坐标系定义见附录。

**数据记录长度：**

内存容量版本	存储时长（按最快纪录频率 200Hz 计算）	说明
8Mb	10min	当数据量达到最大存储容量时，会自动覆盖最早产生的数据。
16Mb	21min	同上
<b>32Mb（默认）</b>	<b>43.5min</b>	<b>同上</b>
64Mb	84min	同上
128Mb	168min	同上
256Mb	6h	同上
1Gb	24h	同上
2Gb	48h	同上

名词解释:	3
一、基本命令	6
1.1 查询固件信息	6
1.1.1 发送端数据帧格式	6
1.1.2 接收端返回数据帧格式	6
1.2 控制 LED 闪灯	7
1.2.1 发送端数据帧格式	7
1.2.2 接收端返回数据帧格式	8
1.3 查询磁场校准状态	8
1.3.1 发送端数据帧格式	8
1.3.2 接收端返回数据帧格式	9
1.4 清除磁场校准状态	9
1.4.1 发送端数据帧格式	10
1.4.2 接收端返回数据帧格式	10
1.5 查询节点电量	11
1.5.1 发送端数据帧格式	11
1.5.2 接收端返回数据帧格式	11
二、控制命令	12
2.1 开始记录姿态	12
2.1.1 发送端数据帧格式	13
2.1.2 接收端返回数据帧格式	13
2.2 停止记录姿态	15
2.2.1 发送端数据帧格式	15
2.2.2 接收端返回数据帧格式	15
2.3 查询记录个数	16
2.3.1 发送端数据帧格式	16
2.3.2 接收端返回数据帧格式	16
2.4 查询第 N 条记录的信息	17
2.4.1 发送端数据帧格式	17
2.4.2 接收端返回数据帧格式	17
2.5 请求传输第 N 条记录	18
2.5.1 发送端数据帧格式	18
2.5.2 接收端返回数据帧格式	19
2.6 请求记录传输	20
2.6.1 发送端数据帧格式	20

2.6.2 接收端返回数据帧格式.....	20
2.7 停止记录传输.....	21
2.7.1 发送端数据帧格式.....	21
2.7.2 接收端返回数据帧格式.....	21
2.8 请求开始实时传输.....	22
2.8.1 发送端数据帧格式.....	22
2.8.2 接收端返回数据帧格式.....	24
2.9 实时传输.....	25
2.9.1 发送端数据帧格式.....	25
2.9.2 接收端返回数据帧格式.....	25
2.10 停止实时传输.....	29
2.7.1 发送端数据帧格式.....	29
2.7.2 接收端返回数据帧格式.....	29
三、 使用流程.....	31
3.1 请求实时数据流程.....	31
3.2 如何校准节点.....	31
3.2.1 校准方法一.....	32
3.2.2 校准方法二.....	34
四、 数据通信说明.....	35
4.1 接收数据.....	38
4.2 发送数据.....	39
五、 硬件参数.....	40
六、 订货信息.....	41
七、 工作流程图.....	41
附录 1 帧格式约定.....	42
附录 2 CRC8 校验源码.....	42
附录 3 坐标系定义.....	43

# 一、基本命令

## 1.1 查询固件信息

查询节点的序列号、固件版本号等信息。

命令字	指令	描述
0x01 (Write)	REQFW	查询固件版本
0x02 (Read)		

### 1.1.1 发送端数据帧格式

数据帧格式
<0x01>+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t    code;     uint8_t    crc8; }REQFW_RMT2BLE;</pre>

### 1.1.2 接收端返回数据帧格式

数据帧格式
<0x02>+[4Byte 出厂固件版本号]+[4Byte 序列号]+[1Byte 内存容量]+[4Byte 蓝牙固件版本号]+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t    code;     uint32_t  fwver;     uint32_t  sn;     uint8_t    memory;     uint32_t  blever;     uint8_t    crc8;</pre>

```
}REQFW_BLE2RMT;
```

出厂固件版本号：例如收到 0x01000304，则代表版本号是 V1.0.3.4，每一个字节代表一个单独的含义，无需再转换成 10 进制。

序列号：例如收到 0x34DF89CC，则节点的蓝牙广播名称为 SKI-DF89CC，APP 连接该蓝牙即可，节点的序列号会印制在包装盒上。

内存容量以 1MByte 为单位，如 0x01 即代表内存为 1MByte，默认返回 4，

注：如果不存在固件版本号、序列号其中的一个或多个，对应位置返回 0xFFFFFFFF。

蓝牙固件版本号格式说明：把收到 4Byte 蓝牙固件用 32 位 16 进制表示，例如收到的是 0x00010007 则蓝牙固件版本号是 V1.0.7，最高 8 位一般为 0。

## 1.2 控制 LED 闪灯

用来在周围很多相同 SKIIN2 的情况下，通过手动控制闪灯，找到自己的 SKIIN2，如果上一次的闪灯时间尚未结束，新时间大于等于上一次的时间，则时间被覆盖，如果小于上一次的时间，则无效。闪灯频率一般选择 4Hz，识别性较强。

命令字	指令	描述	重启生效
0x03 (Write)	LEDBLINK	控制 LED 闪灯	N
0x04 (Read)			

### 1.2.1 发送端数据帧格式

#### 数据帧格式

<0x03>+<1Byte 闪灯频率：1 或 2 或 4Hz>+<2Bytes 闪灯时间：毫秒为单位，最大 65535ms>+<CRC8>

#### C/C++结构体参考

```
typedef struct {
    uint8_t code;
    uint8_t freq;
    uint16_t blinkTimeMs;
    uint8_t crc8;
}LEDBLINK_RMT2BLE;
```



## 1.2.2 接收端返回数据帧格式

数据帧格式
<0x04>+[OK:0x01]+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t code;     uint8_t blinkok;     uint8_t crc8; }LEDBLINK_BLE2RMT;</pre>

用例		
发送	返回	说明
0x0304B80BFF	0x0401B2	以 4Hz 频率闪灯 3000 毫秒

## 1.3 查询磁场校准状态

APP 端查询节点硬件的磁场校准状态和磁场干扰状态。

命令字	指令	描述	重启生效
0x05 (Write)	REQMAG	查询磁场校准状态	N
0x06 (Read)			

### 1.3.1 发送端数据帧格式

数据帧格式
<0x05>+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t code;     uint8_t crc8; }REQMAG_RMT2BLE;</pre>

### 1.3.2 接收端返回数据帧格式

数据帧格式
<0x06>+<1Byte:状态>+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t code;     uint8_t status;     uint8_t crc8; }REQMAG_BLE2RMT;</pre>

状态：低四位为磁场校正准确度，0~3 四个等级。  
高四位为磁场干扰状态，0 为无干扰，1 为有干扰。

用例		
发送	返回	说明
0x05FF	0x060309	节点校准完成，且准确度为 3 最高
0x05FF	06005A	节点校准未完成

### 1.4 清除磁场校准状态

清除保留在模块外置存储器中的磁场校准参数，这在以下情况必须使用：

- (1) 首次使用。
- (2) 更换使用地点（例如更换城市）。
- (3) 受到强磁干扰，姿态异常（表现为即使节点未动，姿态仍在摆动）。
- (4) 其它问题造成的姿态异常。

发送完此条指令，节点会重启姿态解算算法，至少等待 1000ms 后，节点返回数据包指示是否清除成功（清除成功会返回 0x01 状态）。

命令字	指令	描述	重启生效
0x07 (Write)	CLRMAG	清除磁场校准状态	N
0x08 (Read)			

## 1.4.1 发送端数据帧格式

数据帧格式
<0x07>+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t code;     uint8_t crc8; }CLRMAG_RMT2BLE;</pre>

## 1.4.2 接收端返回数据帧格式

数据帧格式
<0x08>+<1Byte:Status>+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t code;     uint8_t status;     uint8_t crc8; }CLRMAG_BLE2RMT;</pre>

1Byte:Status		
代码	状态	备注
0x01	成功清除磁场校准状态	
0xFF	未能清除磁场	

用例		
发送	返回	说明
0x07FF	0x080106	成功清除磁场校准状态

## 1.5 查询节点电量

查询节点的电池剩余电量，0x00~0x64 分别表示剩余电量 0%~100%，0x65 表示节点正在充电，没有充电进度表示，充电满后，LED 灯常亮。

命令字	指令	描述	重启生效
0x09 (Write)	REQBATT	查询节点电量	N
0x0A (Read)			

### 1.5.1 发送端数据帧格式

数据帧格式
<0x09>+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t code;     uint8_t crc8; }REQBATT_RMT2BLE;</pre>

### 1.5.2 接收端返回数据帧格式

数据帧格式
<0x0A>+<1Byte:电量>+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t code;     uint8_t status;     uint8_t crc8; }REQBATT_BLE2RMT;</pre>

## 二、控制命令

### 名词解释：

(1) 记录：一条记录代表一段时间内的姿态数据，包含几百到上万条姿态数据。

(2) 姿态数据：一条姿态数据使用一个四元数表示，每条四元数由四个 int16 类型表示，顺序为 WXYZ。

例如：原始小数表示的四元数为 (0.99,0.707,0,-0.707)，所有数全部乘以 (1<<13) 后，得到整数表示的四元数 (8110,5792,0,-5792)，然后经过蓝牙发送到手机 APP。所以手机端收到的是 8110\5792\0\ -5292 这四个连续的数，然后 APP 再将每个数都除以 (1<<13) 就可以得到小数表示的四元数，这个四元数 APP 可以用来 3D 显示或计算。

底层的数据记录表示：

第 0 条记录，名字为 170920124309（假如：共 1000 条姿态数据）	最大 15 条记录
第 1 条记录，名字为 170923124509（假如：共 3000 条姿态数据）	
第 2 条记录，名字为 170920133455（假如：共 1000 条姿态数据）	
.....	
第 14 条记录，名字为 170920090855（假如：共 5000 条姿态数据）	

注：所有名字在这里都是假设的，APP 端在点击开始按键后，下发 2.1.1 节规定格式的文件名。

基本流程是，APP 端首先查询底层数据的个数，然后查询每一个数据的详细信息，然后选择性的上传一个记录，一个记录通常包含数十到上万条姿态数据，所以底层需要分包上传，最大可能有数十万个数据包，需要持续传输数十分钟。

### 2.1 开始记录姿态

APP 端发送时间戳当做名字，节点记录这个名字并且开始存储姿态信息。

命令字	指令	描述
0x10 (Write)	STARTRECORD	开始记录

0x11 (Read)

## 2.1.1 发送端数据帧格式

数据帧格式
<0x10>+<12Bytes:NAME>+<1Bytes:记录频率>+<1Byte: 数据存储类型>+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t code;     uint8_t name[12];     uint8_t rcdReqFreq;     uint8_t rcdContent;     uint8_t crc8; }STARTRECORD_RMT2BLE;</pre>

NAME: 我们在这里约定文件名固定为 12 字节,

例如: 170920124309

含义: 代表 2017 年 9 月 20 日 12 点 43 分 09 秒创建的文件。

记录频率: 代表底层一秒钟记录多少个姿态值, 取值范围 **1, 2, 4, 5, 10, 20, 25, 50, 100, 200**。

1Byte: 数据发送及存储类型							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
保留位, 默认为 0	保留位, 默认为 0	保留位, 默认为 0	磁力值	角速度	加速度	欧拉角	四元数

注: 为了鉴别记录姿态是否开始, 所以在记录过程中, LED 灯以 1Hz 频率闪烁。

## 2.1.2 接收端返回数据帧格式

数据帧格式
<0x11>+<1Byte: 存储器记录执行状态>+<CRC8>
C/C++结构体参考

```
typedef struct {
    uint8_t code;
    uint8_t status;
    uint8_t crc8;
}STARTRECORD_BLE2RMT;
```

**1Byte: 存储器记录执行状态**

代码	状态	备注
0x01	执行成功	
0x02	节点已经处于记录状态	如果节点已经处于存储状态： 不可以再次开启记录。 不支持实时更改记录频率。 需要：先发送 STOPRECORD 指令结束 上一次的记录。
0xFF	参数错误	存储器记录频率不在取值范围内。

**用例**

发送	返回	说明
0x103137303932303132343330396401FF	0x1101A B	2017年9月20日12点43分09秒创建文件，并以100Hz记录数据，记录四元数
0x10313730393230313234333039C801FF	0x1101A B	2017年9月20日12点43分09秒创建文件，并以200Hz记录数据，记录四元数
0x10313730393230313234333039C801FF	0x1102F8	（假设节点正处于记录状态）创建记录失败，节点正处于记录状态

## 2.2 停止记录姿态

APP 端发送停止记录姿态指令，节点停止记录。

命令字	指令	描述
0x12 (Write)	STOPRECORD	停止记录
0x13 (Read)		

### 2.2.1 发送端数据帧格式

数据帧格式
<0x12>+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t code;     uint8_t crc8; }STOPRECORD_RMT2RMT;</pre>

### 2.2.2 接收端返回数据帧格式

数据帧格式
<0x13>+<1Byte:Status>+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t code;     uint8_t status;     uint8_t crc8; }STOPRECORD_BLE2RMT;</pre>

1Byte:Status		
代码	状态	备注
0x01	停止成功	
0xFF	其它错误导致未能停止	

### 用例



发送	返回	说明
0x12FF	0x130172	成功停止记录

## 2.3 查询记录个数

APP 端查询存储在节点端的记录个数，最大 15 个。

命令字	指令	描述
0x14 (Write)	REQRCDNUM	查询姿态数据的记录个数
0x15 (Read)		

### 2.3.1 发送端数据帧格式

数据帧格式
<0x14>+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t code;     uint8_t crc8; }REQRCDNUM_RMT2BLE;</pre>

### 2.3.2 接收端返回数据帧格式

数据帧格式
<0x15>+<1Byte: status>+[1Byte: RECORDNUM: 记录个数, 为 0~0x0F]+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t code;     uint8_t status;     uint8_t recordNum;     uint8_t crc8; }REQRCDNUM_BLE2RMT;</pre>

status: 0x01-查询成功。 0x02-查询错误，底层正处于记录姿态状态（应先发送停止记录命令 STOPRECORD 然后再次查询）。 0xFF-其它错误。

RECORDNUM 为 0 是代表底层没有任何记录。这时 APP 仍然可以查询每条记录，但是每条记录都是空。

样例，仅供参考：

发送	返回	说明
14FF	1501010E	查询成功，且底层有一条记录
14FF	15020012	查询失败，由于底层正处于记录状态
14FF	1501000D	查询成功，且底层没有记录

## 2.4 查询第 N 条记录的信息

APP 端查询存储在节点端的第 N 条记录的信息。

命令字	指令	描述
0x16 (Write)	REQRCDN	查询第 N 段姿态数据的信息
0x17 (Read)		

### 2.4.1 发送端数据帧格式

数据帧格式
<0x16>+<RecordIndex: 0~0x0F>+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t code;     uint8_t index;     uint8_t crc8; }REQRCDN_RMT2BLE;</pre>

注意：RecordIndex 为第 N 条（N 从 0 开始而不是 1）记录的索引，如果想要查询第 0 条记录，应设置其值为 0。

### 2.4.2 接收端返回数据帧格式

数据帧格式
<0x17>+<1Byte: status>+<EXIST: 存在 0x01, 不存在 0x00>+<12 字节名字>+<帧率: 最大 200>+<四元数个数: 四字节>+<CRC8>

**C/C++结构体参考**

```
typedef struct {
    uint8_t    code;
    uint8_t    status;
    uint8_t    exist;
    uint8_t    name[12];
    uint8_t    freq;
    uint32_t   quatNum;
    uint8_t    crc8;
}REQRCDN_BLE2RMT;
```

status: 0x01-查询成功。 0x02-查询错误，底层正处于记录姿态状态（应先发送停止记录命令 STOPRECORD 然后再次查询）。 0xFF-其它错误。

如果 EXIST 部分为 0，则说明查询的记录不存在，后面的名字帧率等信息默认都是 0，也就不需要解析了。

举例：

发送 1600FF 查询第 0 条记录，返回

17010131 37303932 30313234 333039c8 d62e0000 4e

则代表查询成功、存在此条记录、2017 年 9 月 20 日 12 点 43 分 09 秒创建文件，记录频率为 200Hz。

## 2.5 请求传输第 N 条记录

APP 端开始请求节点传输第 N 条记录，这时要确保底层的数据记录已经停止，发送完回复指令后，节点使用 TRANSRCDN 指令逐条向 APP 端发送四元数。

命令字	指令	描述
0x18 (Write)	TRANSRCDN	传输第 N 段姿态数据
0x19 (Read)		

### 2.5.1 发送端数据帧格式

发送端请求传输时，默认传输所有的姿态数据，但是这样姿态数据个数有可能太多，有时只需要传输其中一段就可以了，例如，第 0 号记录共有 10000 条数据，

但 APP 端只需要第 2000~3000 个数据，所以这时，StartIndex 为 1999 (0x07CF)，结束 Index 为 2999 (0x0BB7)。

这条指令还可以用来获取某条记录的缩略图，例如每隔 10 条数据申请一次。

#### 数据帧格式

<0x18>+<1 Byte RecordIndex: 0~0x0F>+<StartIndex: 4Bytes 表示传输的起始索引>+<EndIndex: 4Bytes 表示传输的结束索引>+<发送频率: 2Bytes>+<CRC8>

#### C/C++结构体参考

```
typedef struct {
    uint8_t    code;
    uint8_t    recordIdx;
    uint32_t   startIdx;
    uint32_t   endIdx;
    uint16_t   freq;
    uint8_t    crc8;
}TRANSRCDN_RMT2BLE;
```

发送频率：代表节点每秒上传的数据包个数，取值范围 1~1000，一般取值 1~100 内发送是比较稳定的，当取值大于 100 时，数据有可能会发生丢包（视使用环境而定）。该值还取决于手机端或 PC 端蓝牙的传输能力，节点的最大传输速率为大约 20KB/s，一般手机端要低于此值。另外，由于蓝牙 BLE 的带宽有上限，所以在使用节点时，若手机端还同时连接了蓝牙耳机、智能手表等外设，节点的数据包发送频率应相应减小以保证数据的连续。

## 2.5.2 接收端返回数据帧格式

#### 数据帧格式

<0x19>+<1Byte Status>+<StartIndex Exist: 索引存在 0x01，不存在或出错，则为 0x00>+<EndIndex Exist: 索引存在 0x01，不存在或出错，则为 0x00>+<CRC8>

#### C/C++结构体参考

```
typedef struct {
    uint8_t code;
    uint8_t status;
    uint8_t startIdxExist;
```

```

uint8_t endIdxExist;
uint8_t crc8;
}TRANSRCDN_BLE2RMT;

```

1Byte Status	
命令	描述
0x00	记录不存在
0x01	记录存在
0x02	节点正处于记录状态，需要先发送 STOPRECORD 指令停止记录。
0x03	节点正处于实时传输状态，需要先发送 REQRTTRANS 指令停止发送。
0x04	节点正处于记录传输状态，需要先发送 STOPTRANSRCD 指令停止发送。

## 2.6 请求记录传输

当节点收到有效的 TRANSRCDN 指令后，则开始从外置存储器 SPIFLASH 中逐条读出请求的姿态数据。

命令字	指令	描述	重启生效
0xXX (No Write)	TRANSRCD	逐条传输姿态数	N
0x2A (Read)		据	

### 2.6.1 发送端数据帧格式

无

### 2.6.2 接收端返回数据帧格式

一次发送两个姿态的数据，由于每个数据包包含两个姿态，所以假如申请奇数个姿态（例如 3 个），则 APP 端会收到两个包共 4 个姿态，忽略最后一个姿态即可，因为最后一个姿态数据是随机读取没有实际含义。

数据帧格式
<0x2A>+<3Bytes: 索引>+<8 Bytes: 四元数 1>+<8 Bytes: 四元数 2>
C/C++结构体参考
<pre> typedef struct {     uint8_t code; </pre>

```
uint8_t index[3];
uint8_t quat[2][8];
}TRANSRCD_BLE2RMT;
```

注：此包没有 CRC8 校验位。

## 2.7 停止记录传输

当需要打断发送流程 TRANSRCD 时，发送该条指令，只有在节点正在发送数据时，该条指令才能发挥作用，否则无效。

命令字	指令	描述	重启生效
0x2B (Write)	STOPTRANSRCD	停止传输姿态数据	N
0x2C (Read)			

### 2.7.1 发送端数据帧格式

数据帧格式
<0x2B>+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t code;     uint8_t crc8; }STOPTRANSRCDN_RMT2BLE;</pre>

### 2.7.2 接收端返回数据帧格式

数据帧格式
<0x2C>+<1Bytes: Status>+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t code;     uint8_t status;     uint8_t crc8;</pre>

```
}STOPTRANSRCDN_BLE2RMT;
```

**1Byte:Status**

代码	状态	备注
0x01	停止成功	
0xFF	其它错误导致未能停止	

## 2.8 请求开始实时传输

要求节点实时传输数据，并同时规定请求的数据内容，等同于同时发送 STARTRECORD 指令，可以人为控制同时板载外置存储器存储/不存储姿态的数据。

命令字	指令	描述
0x2D (Write)	REQRTTRANS	请求实时传输姿态/位置
0x2E (Read)		

### 2.8.1 发送端数据帧格式

**数据帧格式**

<0x2D>+<1Byte: 数据发送频率>+<1Byte: 数据发送及存储类型>+<1Byte: 存储器记录频率>+<12Bytes: NAME>+<CRC8>

**C/C++结构体参考（编译器需支持位域）**

```
typedef struct {
    uint8_t code;
    uint8_t dataReqFreq;
    union {
        struct {
            uint8_t quatRequest : 1;          /*Bit:0*/
            uint8_t eulerAngleRequest : 1; /*Bit:1*/
            uint8_t accelRequest : 1;        /*Bit:2*/
            uint8_t gyroRequest : 1;        /*Bit:3*/
            uint8_t magRequest : 1;         /*Bit:4*/
            uint8_t resv : 3;                /*Bit:5~7*/
        };
    };
    uint8_t dataRequest;
```

<pre>}; uint8_t rcdReqFreq; uint8_t name[12]; uint8_t crc8; }REQRTTRANS_RMT2BLE;</pre>
或（无位域定义）
<pre>typedef struct {     uint8_t code;     uint8_t dataReqFreq;     uint8_t contentRequest;     uint8_t rcdReqFreq;     uint8_t name[12];     uint8_t crc8; }REQRTTRANS_RMT2BLE;</pre>

1Byte: 数据发送频率	
取值范围	说明
1, 2, 4, 5, 10, 20, 25, 50, 100, 200	代表底层一秒钟发送多少个数据包

数据发送及存储类型:

1Byte: 数据发送及存储类型							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
保留位, 默认为 0	保留位, 默认为 0	保留位, 默认为 0	磁力值	角速度	加速度	欧拉角	四元数

存储器记录频率: 代表节点同时进行的数据记录频率, 取值范围 1, 2, 4, 5, 10, 20, 25, 50, 100, 200。

名字命名规范, 所以我们在这里约定文件名固定为 12 字节。

注: 建议发送频率与存储器记录频率相同以保持一致性。当蓝牙传输过程中出现丢包时, 能够后期通过索引从节点获取丢失的帧。

用例		
发送	返回	说明



0x2D320100313730393230313234333039FF	0x2e01ffd6	50Hz 传输，发送四元数，底层不记录
0x2D640100313730393230313234333039FF	0x2e01ffd6	100Hz 传输，发送四元数，底层不记录
0x2D640300313730393230313234333039FF	0x2e01ffd6	100Hz 传输，发送四元数、欧拉角，底层不记录
0x2D640200313730393230313234333039FF	0x2e01ffd6	100Hz 传输，发送欧拉角，底层不记录
0x2D640600313730393230313234333039FF	0x2e01ffd6	100Hz 传输，发送欧拉角、加速度，底层不记录
0x2D640E00313730393230313234333039FF	0x2e01ffd6	100Hz 传输，发送欧拉角、加速度、角速度，底层不记录
0x2D641E00313730393230313234333039FF	0x2e01ffd6	100Hz 传输，发送欧拉角、加速度、角速度、磁场值，底层不记录

## 2.8.2 接收端返回数据帧格式

<b>数据帧格式</b>
<0x2E>+<1Byte: 数据发送执行状态>+<1Byte: 存储器记录执行状态>+<CRC8>
<b>C/C++结构体参考</b>
<pre>typedef struct {     uint8_t code;     uint8_t quatStatus;     uint8_t storStatus;     uint8_t crc8; }REQRTTRANS_BLE2RMT;</pre>

数据发送执行状态：

1Byte: 数据发送执行状态		
代码	状态	备注
0x01	执行成功	

0xFF	参数错误	存储器记录频率不在取值范围内
------	------	----------------

存储器记录执行状态：

1Byte: 存储器记录执行状态		
代码	状态	备注
0x01	执行成功	
0x02	节点已经处于记录状态	如果节点已经处于存储状态： (1) 不可以再次开启记录。 (2) 不支持实时更改记录频率。 需要：先发送 STOPRECORD 指令结束上一次的记录。
0xFF	参数错误	存储器记录频率不在取值范围内。

## 2.9 实时传输

节点收到 REQRTRANS 并且返回 0x01-开启成功状态后，按照请求的发送频率开始上传四元数姿态。

命令字	指令	描述
0xFF (No Write)	RTRANS	实时传输姿态
0x2F (Read)		

### 2.9.1 发送端数据帧格式

无

### 2.9.2 接收端返回数据帧格式

一个数据包发送一个姿态的数据，索引共 3 字节，最大为 16777215。若以最大速率 200Hz 发送，包序号每 23 小时溢出归零一次。

数据包的长度是动态的，由发送内容决定，由于手机或者电脑端接收 Notify 数据每次只有 20 字节，所以会出现粘包问题，在拥堵网络环境下还有可能出现丢包问题，所以建议在任何情况下都使用缓冲区的方法手动处理数据包粘包丢包问

题。

发送数据内容	数据长度
无（只发送 0x2F+数据索引+数据类型标志 0x00，一般用于网络测试）	5
四元数	5+8=13
四元数+欧拉角	5+8+6=19
四元数+欧拉角+加速度	5+8+6+6=25
四元数+欧拉角+加速度+角速度	5+8+6+6+6=31
四元数+欧拉角+加速度+角速度+磁力值	5+8+6+6+6+6=37
欧拉角	5+6=11
欧拉角+加速度	5+6+6=17
欧拉角+加速度+角速度	5+6+6+6=23
欧拉角+加速度+角速度+磁力值	5+6+6+6+6=29

#### 数据帧格式

<0x2F>+<3Bytes: 索引>+<1Byte: 数据标志>+[8 Bytes: 四元数]+[6Bytes: 欧拉角]+[6Bytes: 加速度]+[6Bytes: 角速度]+[6Bytes: 磁力值]

#### C/C++结构体参考

```
typedef struct {
    uint8_t code;
    uint8_t index[3];
    union {
        struct {
            uint8_t quatRequest : 1;           /*Bit:0*/
            uint8_t eulerAngleRequest : 1;    /*Bit:1*/
            uint8_t accelRequest : 1;         /*Bit:2*/
            uint8_t gyroRequest : 1;          /*Bit:3*/
            uint8_t magRequest : 1;           /*Bit:4*/
            uint8_t resv : 2;                  /*Bit:5~6*/
            uint8_t magDisturb : 1;           /*Bit:7*/
        };
        uint8_t dataFlag;
    };
    uint8_t data[64];
}RTTRANS_BLE2RMT;
```

数据类型标志：

标志为 8 位，某一位为 1 则代表之后有此数据，否则无此数据。

1Byte: 数据标志							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
磁干扰指示, 0 代表无磁干扰, 1 代表有磁干扰	保留位, 默认为 0	保留位, 默认为 0	磁力值	角速度	加速度	欧拉角	四元数

8Bytes: 四元数							
Byte7	Byte6	Byte5	Byte4	Byte3	Byte2	Byte1	Byte0
z 高字节	z 低字节	y 高字节	y 低字节	x 高字节	x 低字节	w 高字节	w 低字节
两字节合为一个 int16_t 类型数据		两字节合为一个 int16_t 类型数据		两字节合为一个 int16_t 类型数据		两字节合为一个 int16_t 类型数据	
将上述所得数据除以 $(1 \ll 13)$ 获得浮点数值表示的 z		将上述所得数据除以 $(1 \ll 13)$ 获得浮点数值表示的 y		将上述所得数据除以 $(1 \ll 13)$ 获得浮点数值表示的 x		将上述所得数据除以 $(1 \ll 13)$ 获得浮点数值表示的 w	
取值范围为 [-1.0,+1.0]		取值范围为 [-1.0,+1.0]		取值范围为 [-1.0,+1.0]		取值范围为 [-1.0,+1.0]	

6Bytes: 欧拉角					
Byte5	Byte4	Byte3	Byte2	Byte1	Byte0
z 高字节	z 低字节	y 高字节	y 低字节	x 高字节	x 低字节
两字节合为一个 int16_t 类型数据		两字节合为一个 int16_t 类型数据		两字节合为一个 int16_t 类型数据	
将上述所得数据除以 $(1 \ll 7)$ 获得浮点数值表示的 z		将上述所得数据除以 $(1 \ll 7)$ 获得浮点数值表示的 y		将上述所得数据除以 $(1 \ll 7)$ 获得浮点数值表示的 x	

取值范围为[0,360.0]	取值范围为 [-180.0,+180.0]	取值范围为 [-180.0,+180.0]
----------------	--------------------------	--------------------------

**6Bytes: 加速度**

Byte5	Byte4	Byte3	Byte2	Byte1	Byte0
z 高字节	z 低字节	y 高字节	y 低字节	x 高字节	x 低字节
两字节合为一个 int16_t 类型数据		两字节合为一个 int16_t 类型数据		两字节合为一个 int16_t 类型数据	
将上述所得数据除以 2048.0 获得浮点数值表示的 z		将上述所得数据除以 2048.0 获得浮点数值表示的 y		将上述所得数据除以 2048.0 获得浮点数值表示的 x	
取值范围为[-16.0,+16.0]		取值范围为[-16.0,+16.0]		取值范围为[-16.0,+16.0]	

**6Bytes: 角速度**

Byte5	Byte4	Byte3	Byte2	Byte1	Byte0
z 高字节	z 低字节	y 高字节	y 低字节	x 高字节	x 低字节
两字节合为一个 int16_t 类型数据		两字节合为一个 int16_t 类型数据		两字节合为一个 int16_t 类型数据	
将上述所得数据除以 16.4 获得浮点数值表示的 z		将上述所得数据除以 16.4 获得浮点数值表示的 y		将上述所得数据除以 16.4 获得浮点数值表示的 x	
取值范围为 [-2000.0,+2000.0]		取值范围为 [-2000.0,+2000.0]		取值范围为 [-2000.0,+2000.0]	

**6Bytes: 磁力值**

Byte5	Byte4	Byte3	Byte2	Byte1	Byte0
z 高字节	z 低字节	y 高字节	y 低字节	x 高字节	x 低字节
两字节合为一个 int16_t 类型数据		两字节合为一个 int16_t 类型数据		两字节合为一个 int16_t 类型数据	
将上述所得数据除以 1.0 获得浮点数值表示的 z		将上述所得数据除以 1.0 获得浮点数值表示的 y		将上述所得数据除以 1.0 获得浮点数值表示的 x	

注：此包没有 CRC8 校验位。

## 2.10 停止实时传输

当需要打断发送流程 REQRTTRANS 时，发送该条指令，只有在节点正在发送实时数据时，该条指令才能发挥作用，否则无效。

命令字	指令	描述
0x30 (Write)	STOPTRANSRCD	停止传输姿态数据
0x31 (Read)		

### 2.7.1 发送端数据帧格式

数据帧格式
<0x30>+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t code;     uint8_t crc8; }STOPRTTRANS_RMT2BLE;</pre>

### 2.7.2 接收端返回数据帧格式

数据帧格式
<0x31>+<1Bytes: Status>+<CRC8>
C/C++结构体参考
<pre>typedef struct {     uint8_t code;     uint8_t status;     uint8_t crc8; }STOPRTTRANS_BLE2RMT;</pre>

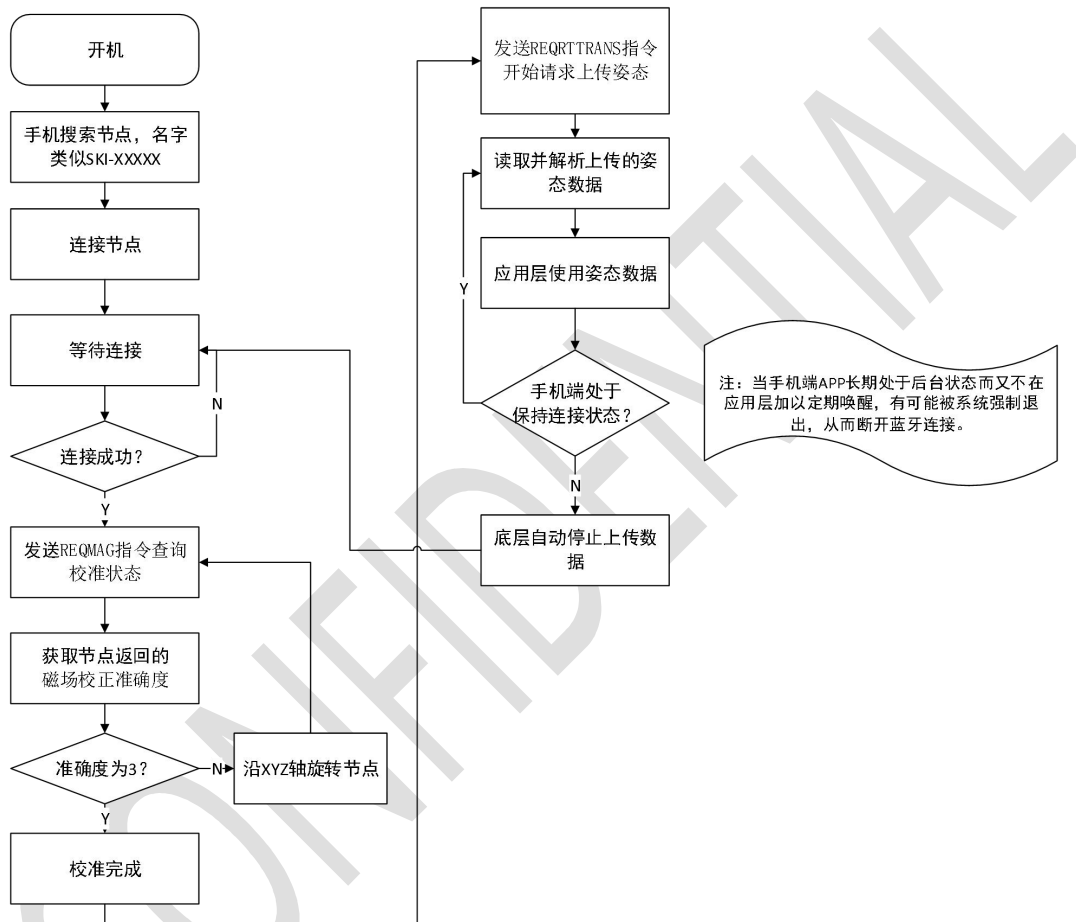
1Byte:Status		
代码	状态	备注
0x01	停止成功	
0xFF	其它错误导致未能停止	

用例		
发送	返回	说明
0x30FF	0x310177	停止传输，执行成功

## 三、使用流程

本章介绍节点的一般使用流程和方法，可以按照流程发送指令进行操作。

### 3.1 请求实时数据流程



需要特别注意的是，当节点处于上传数据过程中时，APP 退至后台有可能被系统强制退出，此时蓝牙连接会断开，节点检测到蓝牙连接断开之后，会自动停止数据上传。当手机端 APP 重新打开并连接节点后，需要再次发送 REQRTTRANS 指令重启上传。

### 3.2 如何校准节点

校准节点是为了校准地磁场，其目的是使节点能够使用 9-DoF 追踪，如果不



校准磁场，航向角会随时间漂移，也就是常说的 6-DoF 追踪。

当传感器的磁场不受周围环境干扰影响的情况下，每次开机使用不必重新校准传感器。当更换使用地点（比如将传感器拿到另外一个城市使用）时，请务必重新校准传感器。当传感器使用环境磁场干扰特别严重时，节点可能无法完成磁场校准，请远离周围强磁干扰。

### 什么是强磁干扰？

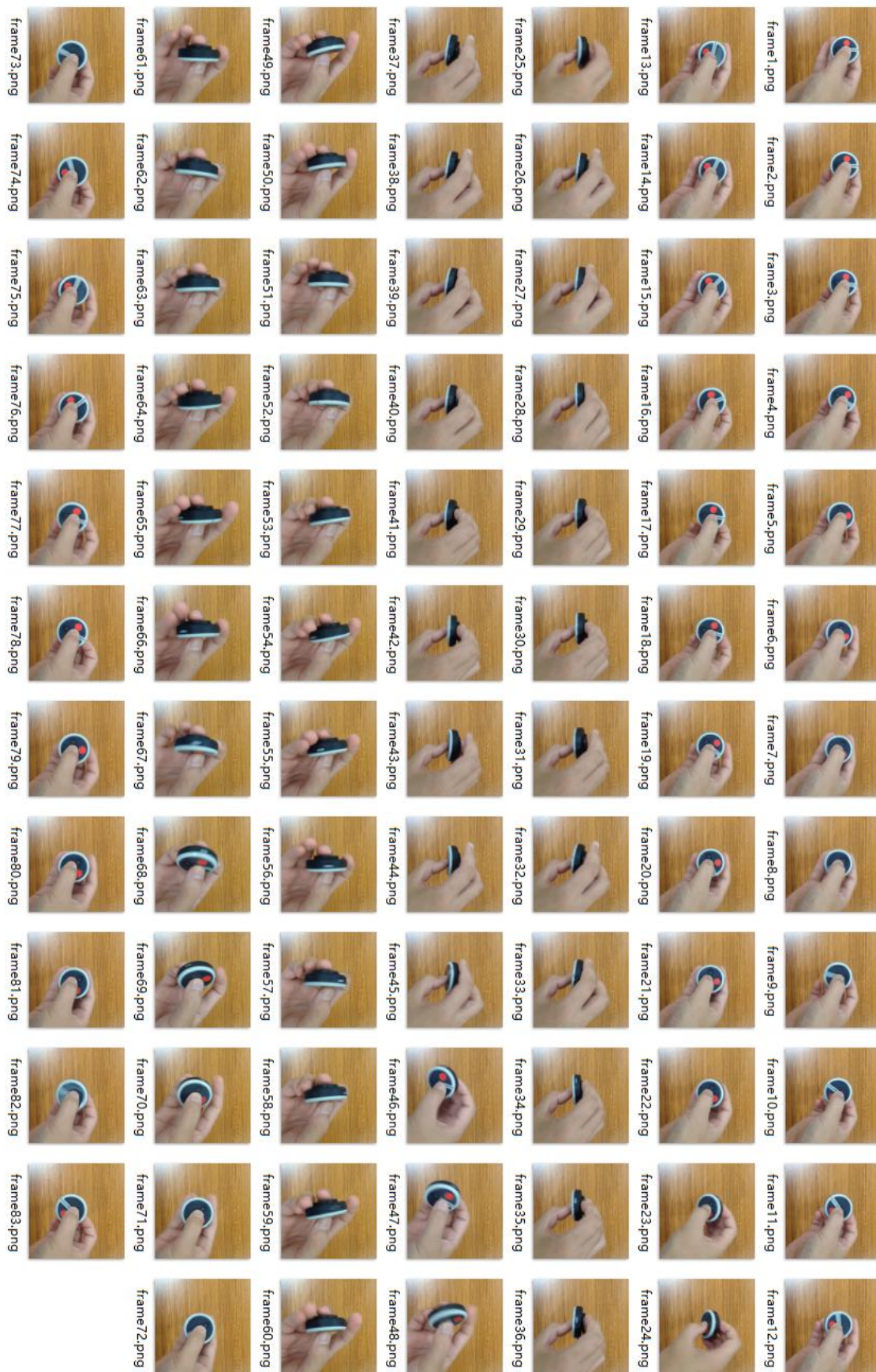
- (1) 手机、手表等电子设备及钥匙硬币；
- (2) 铁、钴、镍以及它们组成的合金等磁体；
- (3) 大功率变电站、无线基站、空调机柜、或大功率电机等工作时周围数米之内；

### 如何避免磁干扰？

对于上述所述 (1) (2) 类干扰需要距离 20cm 以上，对于 (3) 类干扰视干扰强度需要距离至少 1m。

## 3.2.1 校准方法一

使节点分别按照 XYZ 轴旋转三圈，如下图所示（横屏观看）：



### 3.2.2 校准方法二

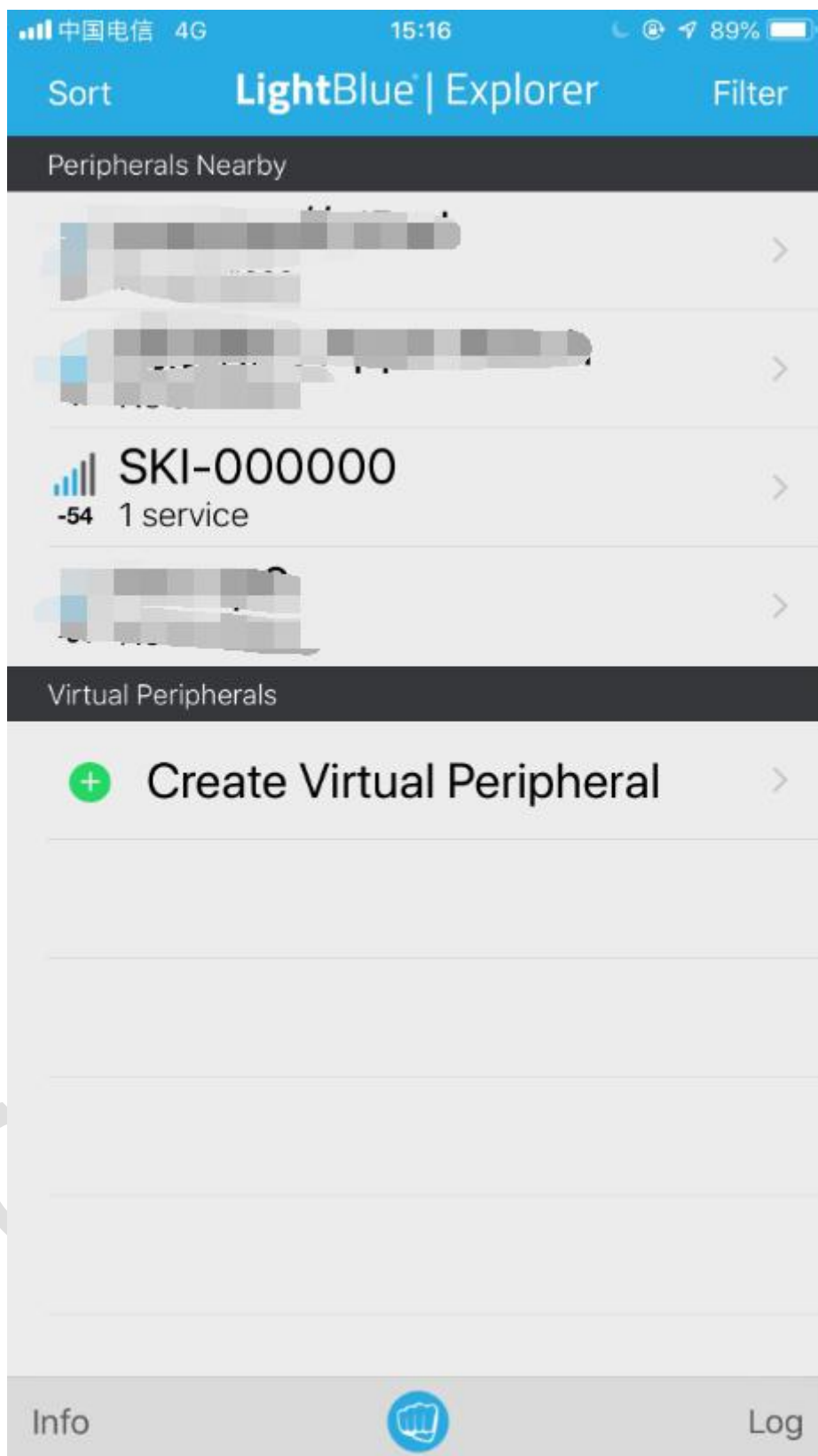
手持节点在空中绘制 8 字形。反复绘制 3~4 次即可。

CONFIDENTIAL

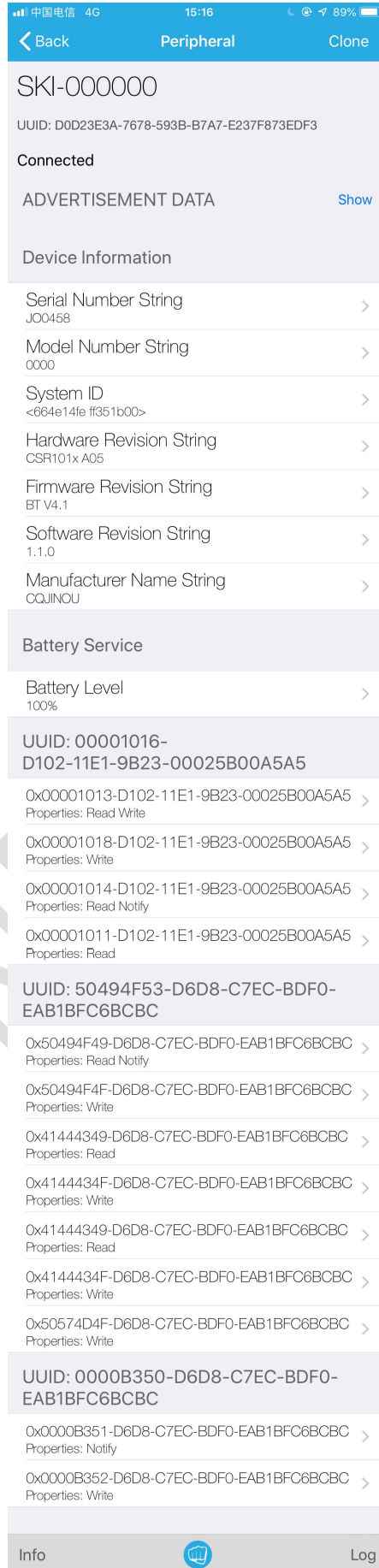
## 四、数据通信说明

使用 IOS LightBlue 软件搜索周围蓝牙设备，会搜索到以 SKI 开头，名字为 SKI-XXXXXX 的蓝牙设备，点击设备，LightBlue 将会与其建立连接。

CONFIDENTIAL



连接建立后，将会显示节点的广播数据、设备信息、PIO 控制服务和数据传输服务以及特征值。

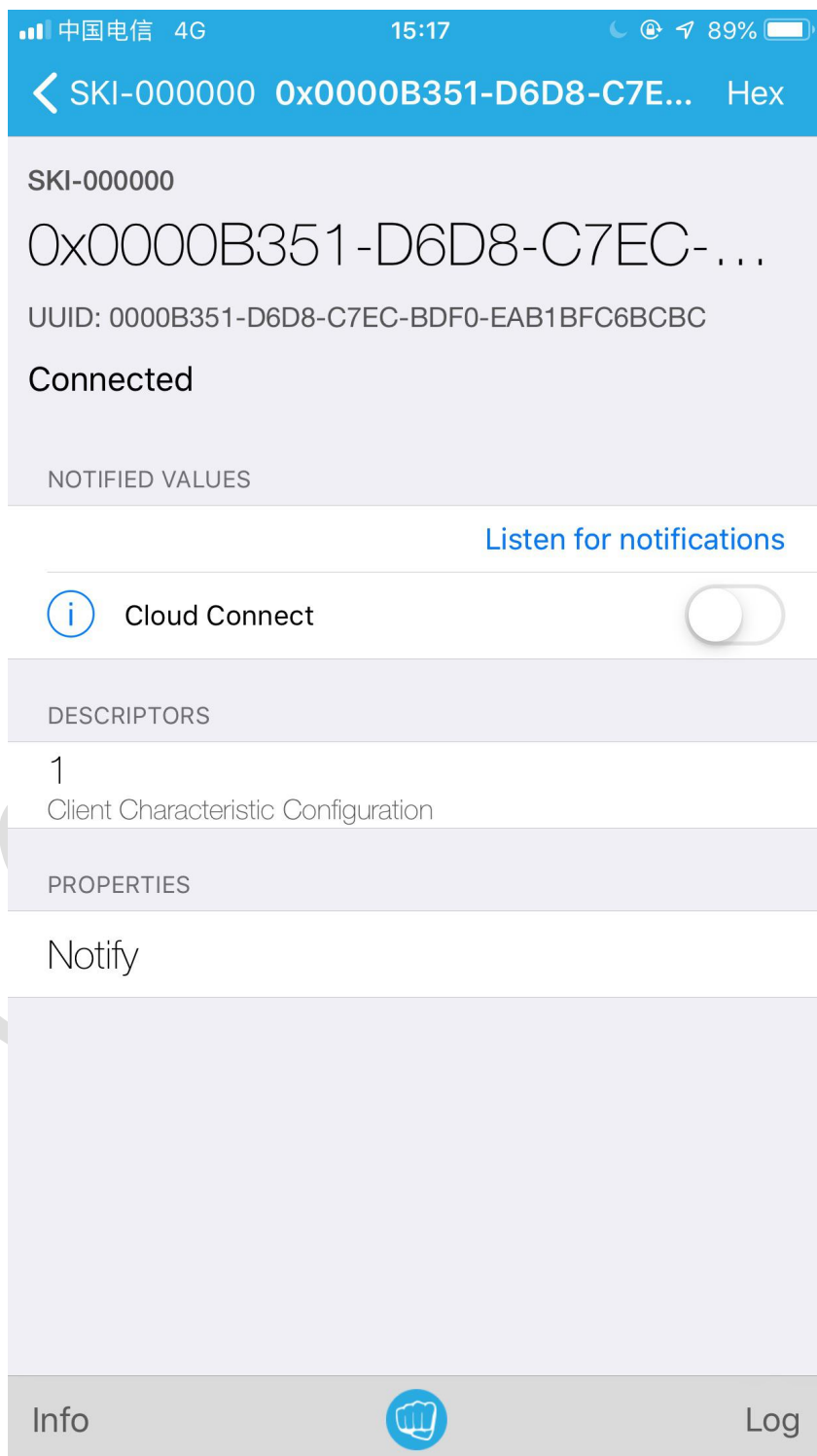


节点的数据传输服务和特征值为使用配置指令进行修改。

默认的服务 UUID 为 0x0000B350-D6D8-C7EC-BDF0-EAB1BFC6BCBC;

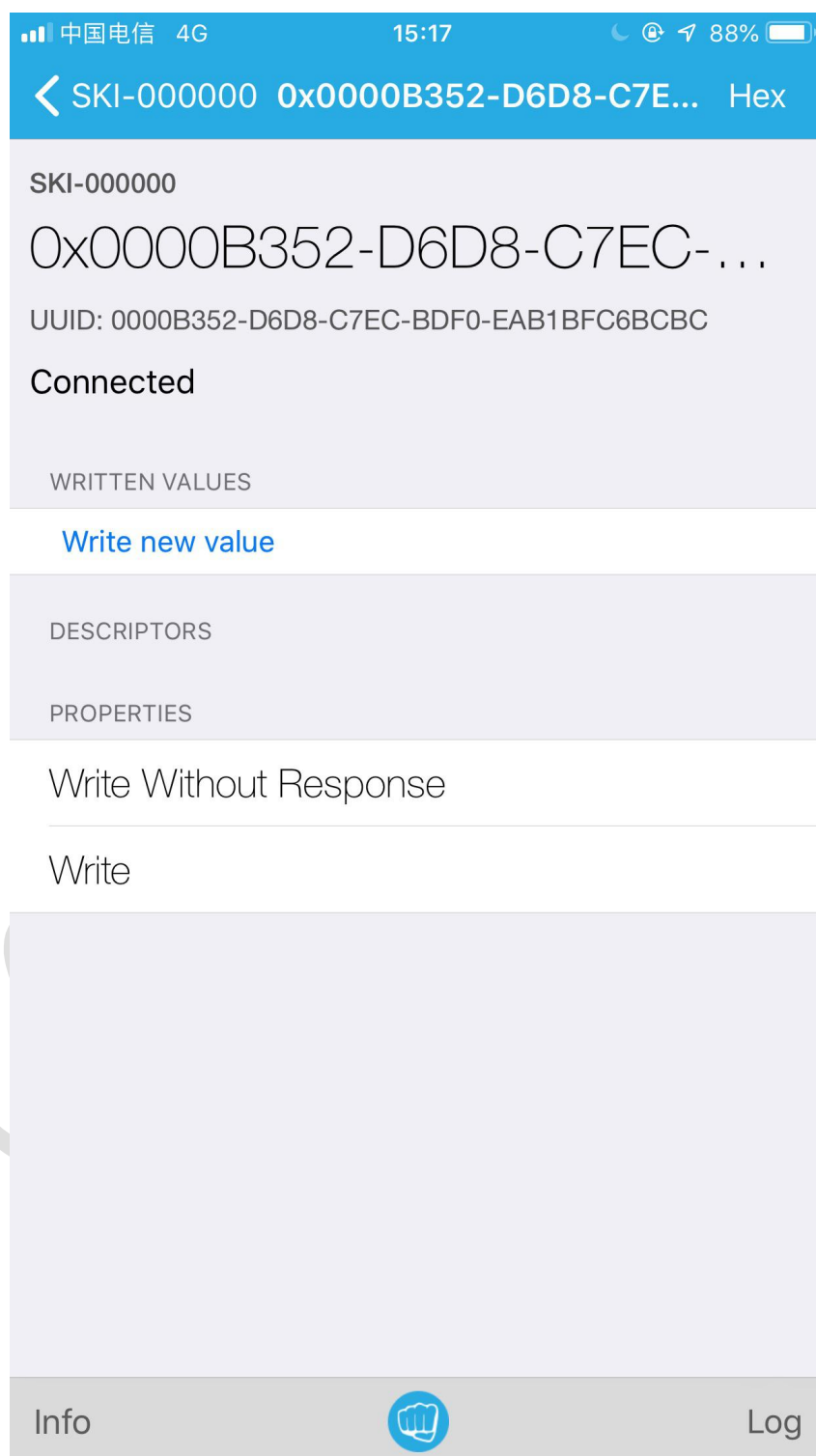
## 4.1 接收数据

默认接收特征值 UUID 为 0x0000B351-D6D8-C7EC-BDF0-EAB1BFC6BCBC，该通道为节点向主机发送的通道；



## 4.2 发送数据

默认发送特征值 UUID 为 0x0000B352-D6D8-C7EC-BDF0-EAB1BFC6BCBC，该通道为主机向节点发送的通道。





## 五、硬件参数

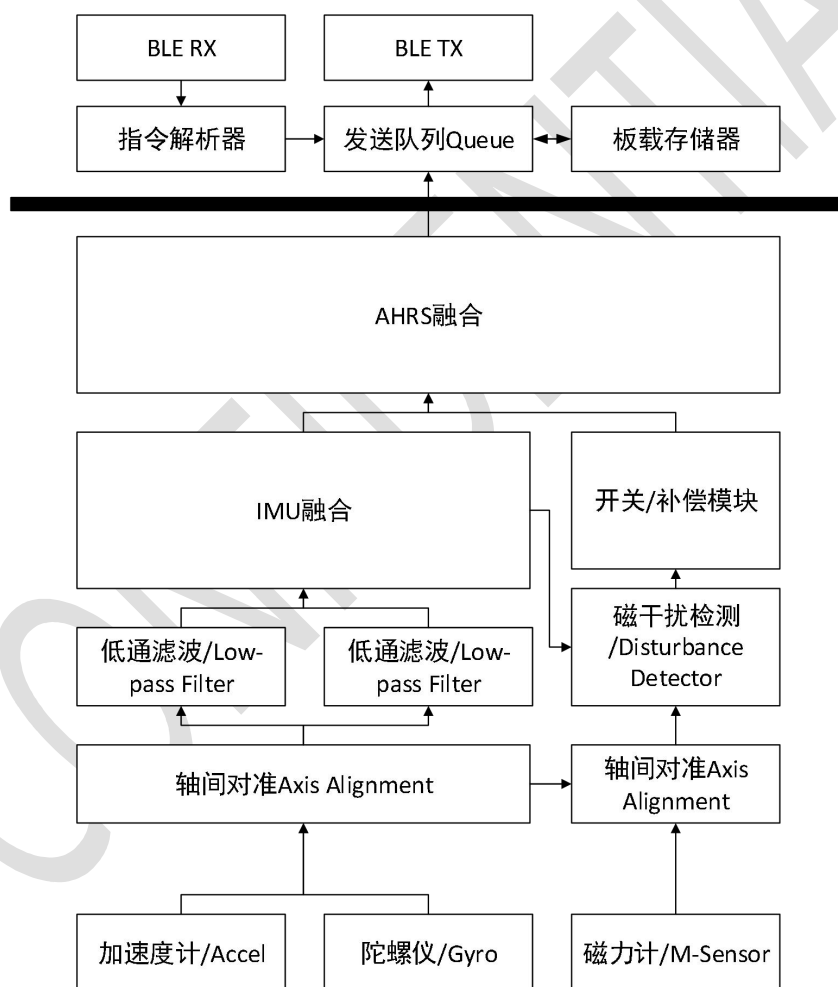
蓝牙规范	Bluetooth V4.1 BLE
工作频段	ISM 频段 2.402~2.480GHz
天线	内置陶瓷天线
发射功率级别	Class2
有效距离	30 米以上
灵敏度	9 dBm transmit power and -93 dBm receive sensitivity
安全特性	SMP、L2CAP、ATT、GAP、GATT
充电电源	5V 1A DC

## 六、订货信息

型号	品牌
MC1490	FOHEART

## 七、工作流程图

本产品可实现 100~200hz 的 9-DoF 追踪，其工作流程如下：



## 附录 1 帧格式约定

帧格式说明：

< >：表示必须包含的部分。

[ ]：表示可选的部分。

一般格式为

<命令字>+[ ].....[ ]+<CRC8>，

其中：

- (1) 命令字：标识当前包的功能。
- (2) CRC8：对整个帧的 CRC8 校验码。

## 附录 2 CRC8 校验源码

由于数据传输需要，所以数据校验必不可少；在这里约定一种统一的 CRC8 校验方法，可以移植到任何支持 ANSIC 标准编译器的平台使用。

```
#include <stdint.h>
static const unsigned char crc_table[] =
{
0x00,0x31,0x62,0x53,0xc4,0xf5,0xa6,0x97,0xb9,0x88,0xdb,0xea,0x7d,0x4c,0x1f,0x2e,
0x43,0x72,0x21,0x10,0x87,0xb6,0xe5,0xd4,0xfa,0xcb,0x98,0xa9,0x3e,0x0f,0x5c,0x6d,
0x86,0xb7,0xe4,0xd5,0x42,0x73,0x20,0x11,0x3f,0x0e,0x5d,0x6c,0xfb,0xca,0x99,0xa8,
0xc5,0xf4,0xa7,0x96,0x01,0x30,0x63,0x52,0x7c,0x4d,0x1e,0x2f,0xb8,0x89,0xda,0xeb,
0x3d,0x0c,0x5f,0x6e,0xf9,0xc8,0x9b,0xaa,0x84,0xb5,0xe6,0xd7,0x40,0x71,0x22,0x13,
0x7e,0x4f,0x1c,0x2d,0xba,0x8b,0xd8,0xe9,0xc7,0xf6,0xa5,0x94,0x03,0x32,0x61,0x50,
0xbb,0x8a,0xd9,0xe8,0x7f,0x4e,0x1d,0x2c,0x02,0x33,0x60,0x51,0xc6,0xf7,0xa4,0x95,
0xf8,0xc9,0x9a,0xab,0x3c,0x0d,0x5e,0x6f,0x41,0x70,0x23,0x12,0x85,0xb4,0xe7,0xd6,
0x7a,0x4b,0x18,0x29,0xbe,0x8f,0xdc,0xed,0xc3,0xf2,0xa1,0x90,0x07,0x36,0x65,0x54,
0x39,0x08,0x5b,0x6a,0xfd,0xcc,0x9f,0xae,0x80,0xb1,0xe2,0xd3,0x44,0x75,0x26,0x17,
0xfc,0xcd,0x9e,0xaf,0x38,0x09,0x5a,0x6b,0x45,0x74,0x27,0x16,0x81,0xb0,0xe3,0xd2,
0xbf,0x8e,0xdd,0xec,0x7b,0x4a,0x19,0x28,0x06,0x37,0x64,0x55,0xc2,0xf3,0xa0,0x91,
0x47,0x76,0x25,0x14,0x83,0xb2,0xe1,0xd0,0xfe,0xcf,0x9c,0xad,0x3a,0x0b,0x58,0x69,
0x04,0x35,0x66,0x57,0xc0,0xf1,0xa2,0x93,0xbd,0x8c,0xdf,0xee,0x79,0x48,0x1b,0x2a,
0xc1,0xf0,0xa3,0x92,0x05,0x34,0x67,0x56,0x78,0x49,0x1a,0x2b,0xbc,0x8d,0xde,0xef,
0x82,0xb3,0xe0,0xd1,0x46,0x77,0x24,0x15,0x3b,0x0a,0x59,0x68,0xff,0xce,0x9d,0xac
```

```

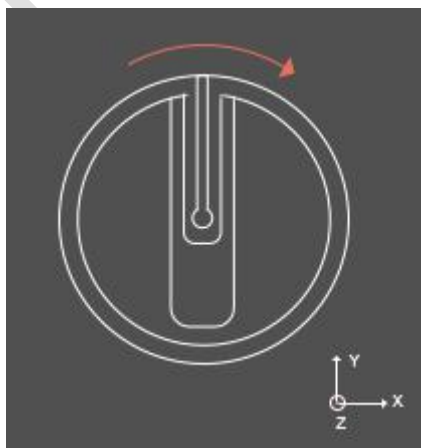
};

/*****
* Function:      CRC8
* Description:   Return the CRC8 result of data stream
* Input:        ptr:Data stream header address
* Input:        len: Total length of data stream
* Output:
* Return:       CRC8 Results
* Others:       Other Description.
*****/
unsigned char cal_crc_table(unsigned char *ptr, unsigned char len)
{
    unsigned char  crc = 0x00;
    while (len--)
    {
        crc = crc_table[crc ^ *ptr++];
    }
    return (crc);
}

```

## 附录 3 坐标系定义

惯性捕捉节点（IMU Sensor），其坐标系定义如下：



导光柱指向 y 轴，右侧指向 x 轴，z 轴指向 Sensor 正上方。

符合该坐标系的输出值：

Sensor 加速度值、角速度值、磁力值。

四元数、欧拉角坐标系符合东(X)北(Y)天(Z)坐标系定义。

CONFIDENTIAL



☎ (+86)010-56106165

✉ [contact@foheart.com](mailto:contact@foheart.com)

🌐 [www.foheart.com](http://www.foheart.com)

📍 北京市海淀区黑山扈路红山口8号D2-南-3号

