

1.BLE透传参数

透传服务 UUID

0000fff0-0000-1000-8000-00805f9b34fb ▼

透传TX特征的UUID (模块-> 方向)

0000fff1-0000-1000-8000-00805f9b34fb ▼

透传RX特征的UUID (->模块 方向)

0000fff2-0000-1000-8000-00805f9b34fb ▼

由于不同公司推出的BLE蓝牙串口模块实现透明传输的策略有所差异，因此如果出现通信失败的情况时，请耐心等待查看蓝牙模块对应的说明书，调整通信的参数。

2.发送程序

CRC-8(x^8+x^2+x+1)的Java和Python实现

注:使用的十六进制参数的字母均为小写

```
public class CRC8Calculator {
    public static int calCRC(int data) {
        int crc = data;
        int poly = 0x07;
        for (int i = 8; i > 0; i--) {
            if ((crc & 0x80) >> 7 == 1) {
                crc = (crc << 1) ^ poly;
            } else {
                crc = (crc << 1);
            }
        }
        return crc & 0xFF;
    }
    public static int calculateCRC8(byte[] data) {
        int crc = 0x00;
        int length = data.length;

        for (int i = 0; i < length; i++) {
            if (i == 0) {
                crc = calCRC(data[0] & 0xFF);
            } else {
                crc = (crc ^ (data[i] & 0xFF)) & 0xFF;
                crc = calCRC(crc);
            }
        }
    }
}
```

```

    }
}
return crc & 0xFF;
}
}

```

```

def cal_crc(data):
    crc = data
    poly = 0x07
    for i in range(8, 0, -1):
        if ((crc & 0x80) >> 7) == 1:
            crc = (crc << 1) ^ poly
        else:
            crc = (crc << 1)
    return crc & 0xFF
def crc8(datas):
    list = [int(byte) for byte in datas]
    length = len(list)
    crc = 0x00
    for i in range(length):
        if i == 0:
            crc = cal_crc(list[0])
        else:
            crc = (crc ^ list[i]) & 0xFF
            crc = cal_crc(crc)
    return crc & 0xFF

```

以下内容以Python的形式叙述

文件发送

发送段需要先发送 `b'\xff\xff'` + 接收段保存文件名 + 分卷大小 + 所发送文件的CRC-8

注:

1. 分卷大小 及 所发送文件的CRC-8 均为十六进制且字母均为小写
2. 接收段保存文件名 需要小于等于10个字节,不足10字节需于前面用 `b'\xff'` 补齐
3. 分卷大小 需要小于等于6个字节,不足6字节需于前面用 `b'0'` 补齐,每18个字节为一个分卷
4. 所发送文件的CRC-8 需要小于等于2个字节,不足2字节需于前面用 `b'0'` 补齐

接收段回应应有三种情况

1. 接收端存在所发送文件,且存在的文件分卷大小,CRC-8与接收到的相同
将接收到的内容去除 接收段保存文件名 返回
2. 接收端存在所发送文件,且存在的文件分卷大小小于接收到的文件分卷大小
返回 `b'\xff\xff'` + 存在的文件分卷大小 + 存在的文件最后的分卷CRC-8
3. 其他,例如不存在文件,存在的文件分卷大小大于接收到的文件分卷大小等
返回 `b'\xff\xff00000000'`

发送端需根据回应发送数据

1. 发送 `b'\xff\xff\xff'` 结束文件发送

2.根据 存在的文件分卷大小 计算出所发送文件对应分卷的CRC-8与 存在的文件最后的分卷CRC-8 进行比较

若相同则发送 `b'\xff\xff\x01'` ,从相应分卷发送分卷内容,发送结束后发送 `b'\xff\xff\xff'` 结束文件发送

不相同发送 `b'\xff\xff\x00'` ,从头发送分卷内容,发送结束后发送 `b'\xff\xff\xff'` 结束文件发送

3.发送 `b'\xff\xff\x00'` ,从头发送分卷内容,发送结束后发送 `b'\xff\xff\xff'` 结束文件发送

注:发送分卷内容前建议延时70ms防止分卷内容与回应内容"黏"在一起,在发送分卷内容时无需延时

3.运行程序

发送端发送 `b'\xff\xff\x10' + 所需运行的程序名`

注:

1.所需运行的程序名 为接收端运行目录下所想运行的程序名,未限制字节长度,但建议使用20字节以内,过多字节可能会导致分包的情况,即发送 `b'\xff\xff\x10example'` 接收到 `b'\xff\xff\x10exa'` 和 `b'mple'` 的情况

2.接收端运行过程的输出会通过蓝牙模块直接返回,接收端只需接收蓝牙返回的信息即可

4.结束程序

发送端发送 `b'\xff\xff\x11'` 即可结束 3.运行程序 中接收端所运行的程序