### 南方电网对讲机传输通路协议定义

[文档副标题]

徐勇

2016

目录

[南方电网对讲机传输通路协议定义 0](file:///C:\\Users\\steve\\Documents\\南方电网对讲机传输通路协议定义.docx" \l "_Toc452820021)

[系统组成 2](#_Toc452820022)

[通信方案： 2](#_Toc452820023)

[通信流程 2](#_Toc452820024)

[通信协议 4](#_Toc452820025)

## 系统组成



图1

通信方案：

1. 蓝牙

手台蓝牙版本4.0。需要智能手机的蓝牙能够支持4.0版本。手台蓝牙和智能手机自动配对。

1. 控制中心和网关之间的通信
2. 网关到手台之间的通信
3. 手机和手台之间的通信

通信流程

1. 手机和手台之间的通信

手机和通过蓝牙的Socket链接到手台的蓝牙上



1. 手台和网关之间的通信

手台和网关之间通过专业通信网络传递数据



1. 控制中心和网关之间的通信

控制中心和网关之间建立TCP长链接，控制中心作为服务器，网关连接到控制中心。



通信协议

注意：协议中需要说明协议格式是小端模式还是大端模式。小端模式即低字节在前，高字节在后；

明确地址数据的内容含义。

对讲机系统提供手机和控制中心之间的透明传输协议。所有控制中心和手机之间传递的数据通过封装在数据单元进行传输。手机与手台之间、网关和控制中心之间都使用该协议

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **报头** | **地址** | **命令** | **长度** | **数据** | **CRC** |
| **0XFFFE** | 2 Bytes | 1 Byte | 1 Byte | 最大60 Bytes | 2 Bytes |

* 报头:2字节。固定为0xfffe
* 地址：2字节。网关发送到控制中心，这个地址设置为0x0000；控制中心发送到网关，这个地址为对讲机的ID号（在组网的时候设置的对讲机唯一标识）
* 命令：
  + 0x01：数据传递，表示该报文有数据传递
  + 0x00：ACK，表示对方已正确收到该报文
  + 0x02：keep alive
  + 0xB1：通知APP，手台发送短信结果，数据中携带结果详情

数据长度：1字节；

数据内容如下：

0x00:发送成功

0x01:失败（未知原因，可能设备忙，或者不在信号覆盖范围内。建议尝试重发，并设置最大重发次数）

0x02:手台未注册

0x03:手台电量过低

0x04:目标ID无效

0x05:无效的状态码

* + 0xFF：NACK，表示对方收到的报文有误
* 长度：1字节。数据单元的长度
* 数据：协议负载数据
* CRC：校验和。校验数据包括：地址，命令，长度和数据

CRC算法示例代码:

unsigned int CRC16\_2(unsigned char \*buf, int len)

{

unsigned int crc = 0xFFFF;

for (int pos = 0; pos < len; pos++)

{

crc ^= (unsigned int)buf[pos]; // XOR byte into least sig. byte of crc

for (int i = 8; i != 0; i--) { // Loop over each bit

if ((crc & 0x0001) != 0) { // If the LSB is set

crc >>= 1; // Shift right and XOR 0xA001

crc ^= 0xA001;

}

else // Else LSB is not set

crc >>= 1; // Just shift right

}

}

return crc;

}

可供参考的校验算法：

unsigned short crc16(unsigned char\* p, int len)

{

#define CRC\_SEED 0xFFFF

#define POLY16 0x1021

unsigned short shift, data, val;

int i;

shift = CRC\_SEED;

for (i = 0; i < len; i++) {

if ((i % 8) == 0){

data = (\*p++) << 8;

}

val = shift ^ data;

shift = shift << 1;

data = data << 1;

if (val & 0x8000){

shift = shift ^ POLY16;

}

}

return shift;

}