**字节说明**

Uint8 无符号字符 1字节；

Uint16 无符号短整型， 2字节

Uint32 无符号整型 4字节；

String 字符串 字节长度 = uint16 + 字符串内容 = 两字节 + 字符串长度

总共三个发送协议,每个协议包含协议头，协议体两部分。每次读包时，首先读取协议头最前面的两个字节的包的长度，然后再读取协议头其他部分，再读取协议体。读取字符串时，首先读取两字节字符串长度，在读取字符串本身。

分别如下：

1. **心跳协议 协议ID(0x01)**

Client(机器人) -> Server(全通处理服务器)

发送协议头：

uint16\_t len; // 整个包的长度

uint16\_t cmd; // 协议ID

uint32\_t code; // 校验码

uint32\_t seq; // 包的序列号

发送协议体：

Uint8\_t ServerType;//服务器类型

Uint8\_t LoadLevel;//服务器优先级

Uint16\_t ServerNo://服务器ID

Server(全通处理服务器)->Client(机器人)

接收协议头：

uint16\_t len; // 整个包的长度

uint16\_t cmd; // 协议ID

uint32\_t code; // 校验码

uint32\_t seq; // 包的序列号

接收协议体：

Uint8\_t result;//0：正常，其他：不正常

1. **会话开启关闭协议 协议ID(0x02)**

Client(机器人) -> Server(全通处理服务器)

发送协议头：

uint16\_t len; // 整个包的长度

uint16\_t cmd; // 协议ID

uint32\_t code; // 校验码

uint32\_t seq; // 包的序列号

发送协议体：

Uint8\_t cmd;//0：请求包，1：返回包

Uint8\_t opt://操作类型， 0：打开会话，1：关闭会话

Uint32\_t conversionID;//用户会话id

Uint32\_t channelid;//渠道id

Uint32\_t cityid;//地市id

String opened;//用户id

String sendTime;//发送时间：公元 1970 年1 月1 日的UTC 时间从0 时0 分0 秒算起到现在所经过的秒数

Server(全通处理服务器)->Client(机器人)

接收协议头：

uint16\_t len; // 整个包的长度

uint16\_t cmd; // 协议ID

uint32\_t code; // 校验码

uint32\_t seq; // 包的序列号

接收协议体：

Uint8\_t result;//0：正常，其他：不正常

String opened;//用户id

1. **内容推送协议 协议ID(0x03)**

Client(机器人) -> Server(全通处理服务器)

发送协议头：

uint16\_t len; // 整个包的长度

uint16\_t cmd; // 协议ID

uint32\_t code; // 校验码

uint32\_t seq; // 包的序列号

发送协议体：

uint8\_t Cmd; // 0为请求包，1为返回包

string OpenID; //用户id

uint8\_t UserType; // 用户类型

string SendTime; // 发送时间 ：公元 1970 年1 月1 日的UTC 时间从0 时0 分0 秒算起到现在所经过的秒数

string Content; // 对话内容

Server(全通处理服务器)->Client(机器人)

接收协议头：

uint16\_t len; // 整个包的长度

uint16\_t cmd; // 协议ID

uint32\_t code; // 校验码

uint32\_t seq; // 包的序列号

接收协议体：

Uint8\_t result;//0：正常，其他：不正常

string OpenID; //用户id