**通信协议**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 协议头 | 消息体 | 分隔符 | 校验位 | 结束符 |
| $ | 消息体 | \* | HL | \r\n |

说明：

协议头：以$作为起始位

消息体：是a~z,A~Z,0~9 等字母组成

分隔符：使用“星号”作为消息体和校验位分隔符

校验位：为消息体的校验位 H为高位，L为低位

结束符：以“回车换行 ”作为结束符

校验位生成算法

#define uint8 unsigned char

/\*

输入：

\*p :消息体

len:消息体字节数

输出：

\*L:校验位低位

\*H：校验位高位

\*/

void gen\_ascii\_crc(char \*p, uint8 len, char \*L, char \*H)

{

uint8 crc = 0;

uint8 i;

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

crc ^=p[i];

}

\*H = crc / 16;

if (\*H > 9) {

\*H = (\*H - 10) + 'A';

}

else {

\*H = (\*H - 9) + '0';

}

\*L = crc % 16;

if (\*L > 9) {

\*L = (\*L - 10) + 'A';

}

else {

\*L = (\*L - 9) + '0';

}

}

实例：

$ABCDEFG,04\r\n

$ABCDEFG1234567890,14\r\n

服务器返回数据

变量名称 类型

00 String 成功

11 String 失败

备注：之前服务器使用OK 或者NO作为返回值。但是这个值和AT指令中的有冲突，导致解析数据会遇到问题，因此选择00 和11 作为服务器的返回值。