**Gate内部协议说明**

文档修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| **文 档 历 史 记 录** | | | |
| 版本号定义规则：  使用阿拉伯数字，并由小数点分割成两部分。  第一部分：整体升级或改造时使用。  第二部分：本文档重大修改时使用，通常需要修改当前生产使用的应用程序。  第三部分：BUG修复时使用 | | | |
| **日 期** | **姓 名** | **版 本** | **更 新 记 录** |
| 2016.05.30 | 白汉升 | 1.0.0 | 基础版本 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 协议约定

1. 上行数据：Client -> Server
2. 下行数据：Server -> Client
3. 长度字段，从“开始符”到校验码的长度，报含开始符和校验码。
4. CRC校验，从“开始符”到数据体,开始符到CRC16的前1个字节。

# 上下行格式

|  |  |  |
| --- | --- | --- |
| **字段** | **长度** | **描述** |
| 开始符 | 2 | 上传：0x40 0x40（＇@＇＇@＇） |
| 目的地址 | 1 | Hex |
| 源地址 | 1 | Hex |
| 包长度 | 2 | 高位在前，地位在后 |
| 数据体 | N |  |
| 校验码 | 2 | CRC16 （附录） |

# 报文类型

## 识别包与回应

识别包由Client到Server，回应包由Server到Client；

识别包与回应包完全相同；

目的地址与源地址同为Client地址，数据体为空，长度为8；

若改Client与Server在3秒钟内无数据通信，则发送该数据包。

## 数据包

目的地址为数据发送目的Client的ID，源地址为数据来源Client的ID。

# 参考文档

1. 《Gate概要设计.docx》
2. 《Gate详细设计.docx》
3. 《Gate相关图.vsdx》

# 附录

|  |
| --- |
| unsigned int code CRC16table[256] =  {  0x0000,0xC0C1,0xC181,0x0140,0xC301,0x03C0,0x0280,0xC241,  0xC601,0x06C0,0x0780,0xC741,0x0500,0xC5C1,0xC481,0x0440,  0xCC01,0x0CC0,0x0D80,0xCD41,0x0F00,0xCFC1,0xCE81,0x0E40,  0x0A00,0xCAC1,0xCB81,0x0B40,0xC901,0x09C0,0x0880,0xC841,  0xD801,0x18C0,0x1980,0xD941,0x1B00,0xDBC1,0xDA81,0x1A40,  0x1E00,0xDEC1,0xDF81,0x1F40,0xDD01,0x1DC0,0x1C80,0xDC41,  0x1400,0xD4C1,0xD581,0x1540,0xD701,0x17C0,0x1680,0xD641,  0xD201,0x12C0,0x1380,0xD341,0x1100,0xD1C1,0xD081,0x1040,  0xF001,0x30C0,0x3180,0xF141,0x3300,0xF3C1,0xF281,0x3240,  0x3600,0xF6C1,0xF781,0x3740,0xF501,0x35C0,0x3480,0xF441,  0x3C00,0xFCC1,0xFD81,0x3D40,0xFF01,0x3FC0,0x3E80,0xFE41,  0xFA01,0x3AC0,0x3B80,0xFB41,0x3900,0xF9C1,0xF881,0x3840,  0x2800,0xE8C1,0xE981,0x2940,0xEB01,0x2BC0,0x2A80,0xEA41,  0xEE01,0x2EC0,0x2F80,0xEF41,0x2D00,0xEDC1,0xEC81,0x2C40,  0xE401,0x24C0,0x2580,0xE541,0x2700,0xE7C1,0xE681,0x2640,  0x2200,0xE2C1,0xE381,0x2340,0xE101,0x21C0,0x2080,0xE041,  0xA001,0x60C0,0x6180,0xA141,0x6300,0xA3C1,0xA281,0x6240,  0x6600,0xA6C1,0xA781,0x6740,0xA501,0x65C0,0x6480,0xA441,  0x6C00,0xACC1,0xAD81,0x6D40,0xAF01,0x6FC0,0x6E80,0xAE41,  0xAA01,0x6AC0,0x6B80,0xAB41,0x6900,0xA9C1,0xA881,0x6840,  0x7800,0xB8C1,0xB981,0x7940,0xBB01,0x7BC0,0x7A80,0xBA41,  0xBE01,0x7EC0,0x7F80,0xBF41,0x7D00,0xBDC1,0xBC81,0x7C40,  0xB401,0x74C0,0x7580,0xB541,0x7700,0xB7C1,0xB681,0x7640,  0x7200,0xB2C1,0xB381,0x7340,0xB101,0x71C0,0x7080,0xB041,  0x5000,0x90C1,0x9181,0x5140,0x9301,0x53C0,0x5280,0x9241,  0x9601,0x56C0,0x5780,0x9741,0x5500,0x95C1,0x9481,0x5440,  0x9C01,0x5CC0,0x5D80,0x9D41,0x5F00,0x9FC1,0x9E81,0x5E40,  0x5A00,0x9AC1,0x9B81,0x5B40,0x9901,0x59C0,0x5880,0x9841,  0x8801,0x48C0,0x4980,0x8941,0x4B00,0x8BC1,0x8A81,0x4A40,  0x4E00,0x8EC1,0x8F81,0x4F40,0x8D01,0x4DC0,0x4C80,0x8C41,  0x4400,0x84C1,0x8581,0x4540,0x8701,0x47C0,0x4680,0x8641,  0x8201,0x42C0,0x4380,0x8341,0x4100,0x81C1,0x8081,0x4040  };  unsigned int CRC16\_isr(unsigned char \*Dat, unsigned int len)  {  unsigned int i,TxCRC16;  for (i=0; i<len; i++)  {  if (Dat[i]!=0)  {  TxCRC16=0;  for (i=0; i<len; i++)  {  TxCRC16=(CRC16table[Dat[i]^(TxCRC16&0xFF)]^(TxCRC16/0x100));  }  }  }  if(TxCRC16==0x0000)  TxCRC16=0xFFFF;  return TxCRC16;  } |