

| CAN发送帧数据格式 | | | | | | | | | | | | | | | | | | | | | | | |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------------|--------|------|
| 帧头 | | 帧长 | 命令 | 发送次数 | | | | 时间间隔 | | | | ID类型 | CAN ID | | | | 帧类型 | len | idAcc | dataAcc | data[len] | CRC | |
| Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 | Data8 | Data9 | Data10 | Data11 | Data12 | Data13 | Data14 | Data15 | Data16 | Data17 | Data18 | Data19 | Data20 | Data21~Data28 | Data29 | |
| 16bit | | 8bit | 8bit | 32bit | | | | 16bit | | | | 8bit | 32bit | | | | 8bit | | | 8bit | | 8bit*8 | 8bit |
| 55 | AA | 1e | 01 | 01 | 00 | 00 | 00 | 0a | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 08 | 00 | 00 | data[8] | crc | |

注：其余格式为 串口转发数据

命令
0x01 转发CAN数据帧
0x02 PC 与设备握手，设备反馈OK
0x03 非反馈CAN转发，不反馈发送状态

ID类型：00 标准帧； 01扩展帧；

帧类型：00 数据帧； 01远程帧

CRC 底层暂时未解析 可以任意数字填充

eg: 55 aa 1e 01 01 00 00 00 0a 00 00 00 00 00 00 00 00 08 00 00 12 23 34 45 56 67 78 89 88

| 串口波特率设置命令 | | | | | | | | | | | |
|-----------|-------|----------|-------|-------|-------|---------|---------|--------|-------|--------|--|
| 帧头 | | baudrate | | | | databit | stopbit | parity | 帧尾 | | |
| Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 | Data8 | Data9 | Data10 | |
| 16bit | | 32bit | | | | 8bit | 8bit | 8bit | 16bit | | |
| 0x55 | 0xAA | | | | | | | | 0xAA | 0x55 | |

```
typedef struct {
    uint32_t baudrate; //波特率
    uint8_t databit; //数据位
    uint8_t stopbit; //格式停止位
    uint8_t parity; //校验
}Baudrate_set_data;
```

```
//数据长度设定位
switch(setdata_t->databit) //0: 8bit 1: 9bit
{
    case 0u: stcNnitCfg.enDataLength = UsartDataBits8; break;
    case 1u: stcNnitCfg.enDataLength = UsartDataBits9; break;
    default: stcNnitCfg.enDataLength = UsartDataBits8; break;
}
//奇偶校验设置
switch(setdata_t->parity) //
{
    case 0u: stcNnitCfg.enParity = UsartParityNone; break;
    case 1u: stcNnitCfg.enParity = UsartParityEven; break;
    case 2u: stcNnitCfg.enParity = UsartParityOdd; break;
    default: stcNnitCfg.enParity = UsartParityNone; break;
}

//奇偶校验设置
switch(setdata_t->stopbit) //
{
    case 0u: stcNnitCfg.enStopBit = UsartOneStopBit; break;
    case 1u: stcNnitCfg.enStopBit = UsartTwoStopBit; break;
    default: stcNnitCfg.enStopBit = UsartTwoStopBit; break;
}
```

| 设置CAN 波特率命令 | | | | |
|-------------|-------|-------|-------|-------|
| 帧头 | | 索引 | 帧尾 | |
| Data0 | Data1 | Data2 | Data3 | Data4 |
| 16bit | | 8bit | | |
| 0x55 | 0x05 | | 0xAA | 0x55 |

| 索引 | 波特率 | 理想CIA | 实际CIA |
|--------------|----------|----------|----------|
| //index0---- | 1000kbps | -- 75% | -- 75 % |
| //index1---- | 800kbps | -- 80% | -- 80 % |
| //index2---- | 666kbps | -- 80% | -- 83.3% |
| //index3---- | 500kbps | -- 87.5% | -- 87.5% |
| //index4---- | 400kbps | -- 87.5% | -- 85 % |
| //index5---- | 250kbps | -- 87.5% | -- 87.5% |
| //index6---- | 200kbps | -- 87.5% | -- 85 % |
| //index7---- | 125kbps | -- 87.5% | -- 87.5% |
| //index8---- | 100kbps | -- 87.5% | -- 87.5% |
| //index9---- | 80kbps | -- 87.5% | -- 865% |
| //index10--- | 50kbps | -- 87.5% | -- 87.5% |
| //index11--- | 40kbps | -- 87.5% | -- 85 % |
| //index12--- | 20kbps | -- 87.5% | -- 85 % |
| //index13--- | 10kbps | -- 87.5% | -- 85 % |
| //index14--- | 5kbps | -- 87.5% | -- 87.5% |

| PC与设备心跳 | | | | |
|---------|-------|-------|--|-------|
| 帧头 | | 帧尾 | | |
| Data0 | Data1 | Data3 | | Data4 |
| 16bit | | | | |
| 0x55 | 0x04 | 0xAA | | 0x55 |

注： CAN返回数据 CMD为00 其余can数据无效

| CAN停止发送（连续发送中） | | | | |
|----------------|-------|-------|--|-------|
| 帧头 | | 帧尾 | | |
| Data0 | Data1 | Data3 | | Data4 |
| 16bit | | | | |
| 0x55 | 0x03 | 0xAA | | 0x55 |

注： 停止现有连续发送

| CAN接收数据帧格式 | | | | | | | | | | | | | | |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|
| 帧头 | 命令 | 格式 | CANID | | | | 数据 | | | | | | | 帧尾 |
| Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 | Data8 | Data9 | Data10 | Data11 | Data12 | Data13 | Data14 |
| AA | 0x00 | | | | | | | | | | | | | 55 |

CAN 命令 //00 心跳 0x01 接收失败 0x11 接收成功 0x02 发送失败 0x12 发送成功 0x03 波特率设置失败 0x13 波特率设置成功 0xEE 通讯错误信息

当命令字段为0xEE时 格式字段为错误代码

格式 包含数据长度 数据帧类型 详见如下结构体

```
CAN 返回数据格式
typedef struct //can发送功能相关结构体 16bytes
{
    uint8_t freamHeader; //发送标志位 0xAA
    uint8_t CMD; //CAN 命令 //00 心跳 0x01 接收失败 0x11 接收成功 0x02 发送失败 0x12 发送成功
    uint8_t canDataLen; //数据长度
    uint8_t canIde; //ide:0,标准帧;1,扩展帧
    uint8_t canRtr; //rtr:0,数据帧;1,远程帧
    uint32_t CANID; //can ID
    uint8_t canData[8]; //Can 数据
    uint8_t freamEnd; //结尾 0x55
} CAN_Fream;
```

错误代码
0x00 无错误
0x01 总线关闭 请求复位

eg: AA 11 08 00 00 00 00 00 01 02 03 04 05 06 07 55

| CAN特殊发送帧数据格式 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|-------|-------|-------|-----------|-------|--------------|--------------|--------|---------------|-----------|--------|---------------|--------------|--------|---------------|-----------|--------|---------------|--------------|--------|---------------|-----------|--|--|--------------|------|------|-----------|--|--|--------------|------|------|-----------|--|--|--------------|--|--|
| 帧头 | | 帧长 | 命令 | CAN ID[0] | | | data[0][len] | | | CAN ID[1] | | | data[1][len] | | | CAN ID[2] | | | data[2][len] | | | CAN ID[3] | | | data[3][len] | | | CAN ID[4] | | | data[4][len] | | | CAN ID[5] | | | data[5][len] | | |
| Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6~Data13 | Data14 | Data15 | Data16~Data23 | Data24 | Data25 | Data26~Data33 | Data34 | Data35 | Data36~Data43 | Data44 | Data45 | Data46~Data53 | Data54 | Data55 | Data56~Data63 | | | | | | | | | | | | | | | | | | |
| 16bit | | 8bit | 8bit | 16bit | | | 8bit | 8bit | 8bit | 16bit | | | 8bit | 8bit | 8bit | 16bit | | | 8bit | 8bit | 8bit | 16bit | | | 8bit | 8bit | 8bit | 16bit | | | 8bit | 8bit | 8bit | | | | | | |
| 55 | AA | 40 | FF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

默认标准帧 ID占16位 数据8bytes

命令 FF 无反馈状态发送 0x5F 反馈发送状态