并口数据采集通讯协议

第一节 链路数据包结构（Base Package）

- 本节只定义链路数据包结构
- 本节不关心双方链接状态，该功能由第二节或者其他文档支持

Base Package结构

```
-------------------------------------------------------------
| HEAD(3 B) | VER（1 B）| LEN(2 ) | TYPE(1 ) |    DATA | CRC16 |
-------------------------------------------------------------
```

- HEAD：三个字节，固定值 0x55 0xFF 0x55
- VER：一个字节，版本号，当前版本号0x01
- LEN ：两个字节，数据包长度（包括 TYPE，DATA 但不包含CRC16）
- TYPE：一个字节，数据类型，由第二节或者其他文档规定

- DATA：不定长，格式有TYPE决定
- CRC16: 对DATA字段内容校验

第二节 数据采集设备传输数据和协议（**TYPE=0x01**）

- 本节定义数据采集设备和盒子的数据模型以及传输协议
- 本节定义的传输协议运行在第一节（UART Package）之上
- 本节定义的Package在第一节（UART Package）中的TYPE类型为0x01

数据定义

数据定义：

```
SYNC：=0x01
DATA:= 0x02
ACK := 0x03
EOF := 0x04
RESULT := 0x00 | 0x0F （0x00 ok， 0x0F fail ）
VER := 0x01
```

Package基本结构：

```
-----------------------------------------
| TYPE VERSION (1) | SEQ (1 BYTE) | DATA |
-----------------------------------------
```

- TYPE：四个bit，数据类型，取值SYNC | DATA| EOF| ACK
- VERSION：四个bit，版本号，当前版本号0x01
- SEQ：一个字节，用于判别包一致性
- DATA：传输给对方的数据，根据TYPE 不同取值不同

TYPE = SYNC | EOF

```
 ----------------------------------
 | TYPE VERSION (1) | SEQ (1 BYTE) |
 ----------------------------------
 TYPE :=SYNC | EOF
 VERSION := VER
```

TYPE = DATA

```
-----------------------------------------
| TYPE VERSION (1) | SEQ (1 BYTE) | DATA |
-----------------------------------------
TYPE :=DATA
VERSION := ver
DATA  := OCTET
```

TYPE = ACK

```
-----------------------------------------
| TYPE VERSION (1) | SEQ (1 BYTE) | DATA |
-----------------------------------------
TYPE :=ACK
VERSION := VER
DATA  := RESULT
```

SYNC包：

用于初始化一次会话，同步双方seq 需要对方ACK包响应，且SEQ相同

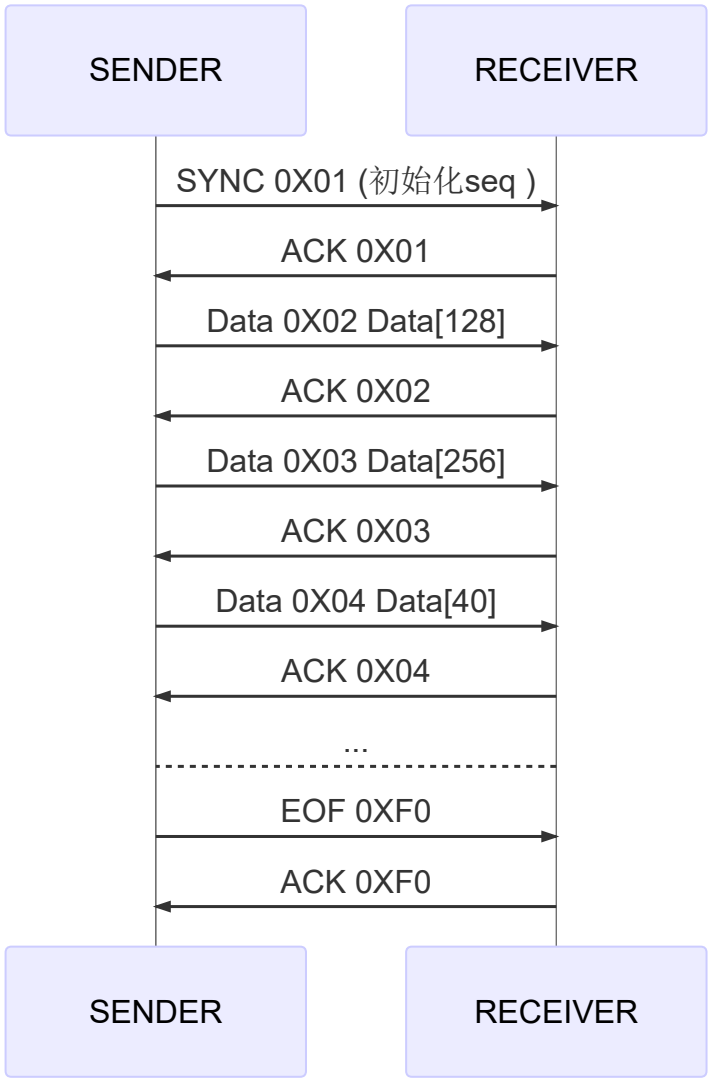
EOF 包：

用于结束一次会话， 需要对方ACK包响应，且SEQ相同

DATA包：

用于会话中传输数据，SEQ递增 需要对方ACK包响应，且SEQ相同

交互示例



超时处理：

- 发送端

> 当发送端，接收到ACK fail，或者 在20ms内没有接收到ACK，选择重发一次， 如果重发再一次表示会话失败，结束会话

- 接收端

> 如果在20ms 内没有数据，认为数据发送完毕，结束会话

SEQ字段：

> SYNC 包初始化seq， DATA 和 EOF 包采用循环递增方式产生该字段

CRC16-CCITT

```
static unsigned short ccitt_table[256] = {
0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50A5, 0x60C6, 0x70E7,
0x8108, 0x9129, 0xA14A, 0xB16B, 0xC18C, 0xD1AD, 0xE1CE, 0xF1EF,
0x1231, 0x0210, 0x3273, 0x2252, 0x52B5, 0x4294, 0x72F7, 0x62D6,
0x9339, 0x8318, 0xB37B, 0xA35A, 0xD3BD, 0xC39C, 0xF3FF, 0xE3DE,
0x2462, 0x3443, 0x0420, 0x1401, 0x64E6, 0x74C7, 0x44A4, 0x5485,
0xA56A, 0xB54B, 0x8528, 0x9509, 0xE5EE, 0xF5CF, 0xC5AC, 0xD58D,
0x3653, 0x2672, 0x1611, 0x0630, 0x76D7, 0x66F6, 0x5695, 0x46B4,
0xB75B, 0xA77A, 0x9719, 0x8738, 0xF7DF, 0xE7FE, 0xD79D, 0xC7BC,
0x48C4, 0x58E5, 0x6886, 0x78A7, 0x0840, 0x1861, 0x2802, 0x3823,
0xC9CC, 0xD9ED, 0xE98E, 0xF9AF, 0x8948, 0x9969, 0xA90A, 0xB92B,
0x5AF5, 0x4AD4, 0x7AB7, 0x6A96, 0x1A71, 0x0A50, 0x3A33, 0x2A12,
0xDBFD, 0xCBDC, 0xFBBF, 0xEB9E, 0x9B79, 0x8B58, 0xBB3B, 0xAB1A,
0x6CA6, 0x7C87, 0x4CE4, 0x5CC5, 0x2C22, 0x3C03, 0x0C60, 0x1C41,
0xEDAE, 0xFD8F, 0xCDEC, 0xDDCD, 0xAD2A, 0xBD0B, 0x8D68, 0x9D49,
0x7E97, 0x6EB6, 0x5ED5, 0x4EF4, 0x3E13, 0x2E32, 0x1E51, 0x0E70,
0xFF9F, 0xEFBE, 0xDFDD, 0xCFFC, 0xBF1B, 0xAF3A, 0x9F59, 0x8F78,
0x9188, 0x81A9, 0xB1CA, 0xA1EB, 0xD10C, 0xC12D, 0xF14E, 0xE16F,
0x1080, 0x00A1, 0x30C2, 0x20E3, 0x5004, 0x4025, 0x7046, 0x6067,
0x83B9, 0x9398, 0xA3FB, 0xB3DA, 0xC33D, 0xD31C, 0xE37F, 0xF35E,
0x02B1, 0x1290, 0x22F3, 0x32D2, 0x4235, 0x5214, 0x6277, 0x7256,
0xB5EA, 0xA5CB, 0x95A8, 0x8589, 0xF56E, 0xE54F, 0xD52C, 0xC50D,
0x34E2, 0x24C3, 0x14A0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,
0xA7DB, 0xB7FA, 0x8799, 0x97B8, 0xE75F, 0xF77E, 0xC71D, 0xD73C,
0x26D3, 0x36F2, 0x0691, 0x16B0, 0x6657, 0x7676, 0x4615, 0x5634,
0xD94C, 0xC96D, 0xF90E, 0xE92F, 0x99C8, 0x89E9, 0xB98A, 0xA9AB,
0x5844, 0x4865, 0x7806, 0x6827, 0x18C0, 0x08E1, 0x3882, 0x28A3,
0xCB7D, 0xDB5C, 0xEB3F, 0xFB1E, 0x8BF9, 0x9BD8, 0xABBB, 0xBB9A,
0x4A75, 0x5A54, 0x6A37, 0x7A16, 0x0AF1, 0x1AD0, 0x2AB3, 0x3A92,
0xFD2E, 0xED0F, 0xDD6C, 0xCD4D, 0xBDAA, 0xAD8B, 0x9DE8, 0x8DC9,
0x7C26, 0x6C07, 0x5C64, 0x4C45, 0x3CA2, 0x2C83, 0x1CE0, 0x0CC1,
0xEF1F, 0xFF3E, 0xCF5D, 0xDF7C, 0xAF9B, 0xBFBA, 0x8FD9, 0x9FF8,
0x6E17, 0x7E36, 0x4E55, 0x5E74, 0x2E93, 0x3EB2, 0x0ED1, 0x1EF0
};
unsigned short crc_ccitt(unsigned char *q, int len)
{
unsigned short crc = 0;

while (len-- > 0)
crc = ccitt_table[(crc >> 8 ^ *q++) & 0xff] ^ (crc << 8);
return ~crc

}
```