8 节点 1394 仿真卡使用说明书

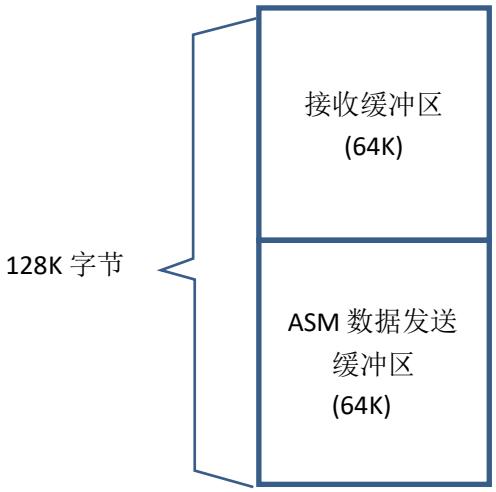
1 系统架构

本仿真卡内实现了 8 个独立的 1394 节点，每个节点具有 1 个端口，每个节点可以根据用户配置工作在 CC、RN 或 BM 模式。

使用板卡开发应用程序时，对于 WIN7 及以上的操作系统，应用程序运行时，需要使用管理员权限。

1.1 节点数据缓冲区

每个节点内的数据接收发送采用缓冲区的方式，缓冲区的大小为 128K 字节，由数据接收缓冲区、发送 ASM 数据缓冲区，用户可以根据自己的需要设置各个缓冲区的大小，但缓冲区总和不能够超过128K字节大小。缺省情况下，如果用户不配置缓冲区大小，则数据接收缓冲区大小为 64K 字节，ASM数据发送缓冲区 64K 字节。



1.2 内存接收 FIFO 缓冲区

板卡上每个节点内的接收缓冲区作为接收数据的一级缓冲区，用户应用程序无法直接访问到该接收缓冲区。相反，板卡硬件会自动将每个节点内的接收缓冲区中的数据发送到主机内存中开辟的 4M 字节大小的 FIFO 缓冲区中，用户应用程序可以通过 API 函数，每次能够读取一个节点的一个 STOF 周期中的所有接收数据。由于采用了该 FIFO 结构，可以保证应用程序在定时器不准的情况下，不会丢

失总线上的数据，但同时，也要求用户应用程序的读取速度应该足够快，至少需要大于总线的 STOF 周期，以保证内存中的 FIFO 缓冲区不会溢出。用户程序可以通过调用函数 b9019_m1394_get_fifo_len 得到当前 FIFO 缓冲区中还剩余的数据包个数。

2 软件使用总体流程

软件的 API 接口从功能上划分为 3 大部分，上电初始化、周期任务的数据发送接收、其它辅助功能。相关函数如下：

1. 上电初始化函数
   - int b9019_m1394_brd_open(int brd);
   - void b9019_m1394_brd_close(int brd);
   - void b9019_m1394_brd_reset(int brd)
   - int b9019_m1394_config_brd(int brd, int speed, int node_en);
   - int b9019_m1394_config_node_mode(int brd, int node, unsigned int mode);
   - int b9019_m1394_config_node_mem(int brd, int node, unsigned int rxbuf_len, unsigned int txbuf_asm_len, unsigned int bak);
   - int b9019_m1394_config_node_ctrl(int brd, int node, unsigned int cc_ctrl, unsigned int frame_cycle , unsigned int b2b_mode, unsigned int rn_use_cc_offset, unsigned int min_tx_time);
   - int b9019_m1394_config_node_chan_filter(int brd, int node, unsigned int filter_en, unsigned int filter_hi32, unsigned int filter_lo32);
   - int b9019_m1394_config_node_msg_filter(int brd, int node, unsigned int filter_en, unsigned int filter_num, unsigned int filter_id[]);
   - int b9019_m1394_config_tx_asm_index(int brd, int node, unsigned int index, unsigned int ctrl_hi16, unsigned int len, unsigned int tx_time, unsigned int dest_chan_id);
   - void b9019_m1394_start_init(int brd);
   - unsigned int b9019_m1394_end_init(int brd);
   - int b9019_m1394_node_start(int brd, int node, int rx_en, int tx_en);
   - int b9019_m1394_set_node_stof_heartbeat_en(int brd, int node, unsigned int cfg);
2. 周期任务的数据发送接收函数
   - int b9019_m1394_write_tx_stof_data(int brd, int node, int auto_cword, void *pbuf, unsigned int len);
   - int b9019_m1394_write_tx_asm_data(int brd, int node, int auto_cword, unsigned

```
int index, void *pbuf, unsigned int len);
```

- `int b9019_m1394_node_tx(int brd, int node, int tx_stof, int tx_asm, int bak);`
- `int b9019_m1394_read_rx_data(int brd, int len, void * pbuf);`

3. 其它辅助函数

- `int b9019_m1394_node_stop(int brd, int node, int rx_en, int tx_en);`
- `int b9019_m1394_rx_buf_index_num(void * pbuf);`
- `b9019_m1394_rx_buf_index_info(void * pbuf, int index, unsigned int *plen, unsigned int *pspd, unsigned int *perr_code, unsigned int *perr_vpc, unsigned int *ptime_tag_hi, unsigned int *ptime_tag_lo);`
- `int b9019_m1394_rx_buf_index_data(void * pbuf, int index);`
- `int b9019_m1394_rx_buf_chan_data(void * pbuf, unsigned int chan);`
- `int b9019_m1394_get_fifo_len(int brd, unsigned int *plen)`
- `int b9019_m1394_rx_buf_len(void * pbuf)`
- `int b9019_m1394_manual_tx_stof_data(int brd, int node, int auto_cword, void *pbuf, unsigned int len);`
- `int b9019_m1394_rx_buf_stof_addr(void * pbuf)`
- `int b9019_m1394_rx_buf_stof_info(void * pbuf, int index, unsigned int dw_num, unsigned int *stat, unsigned int *ptime_tag_hi, unsigned int *ptime_tag_lo);`

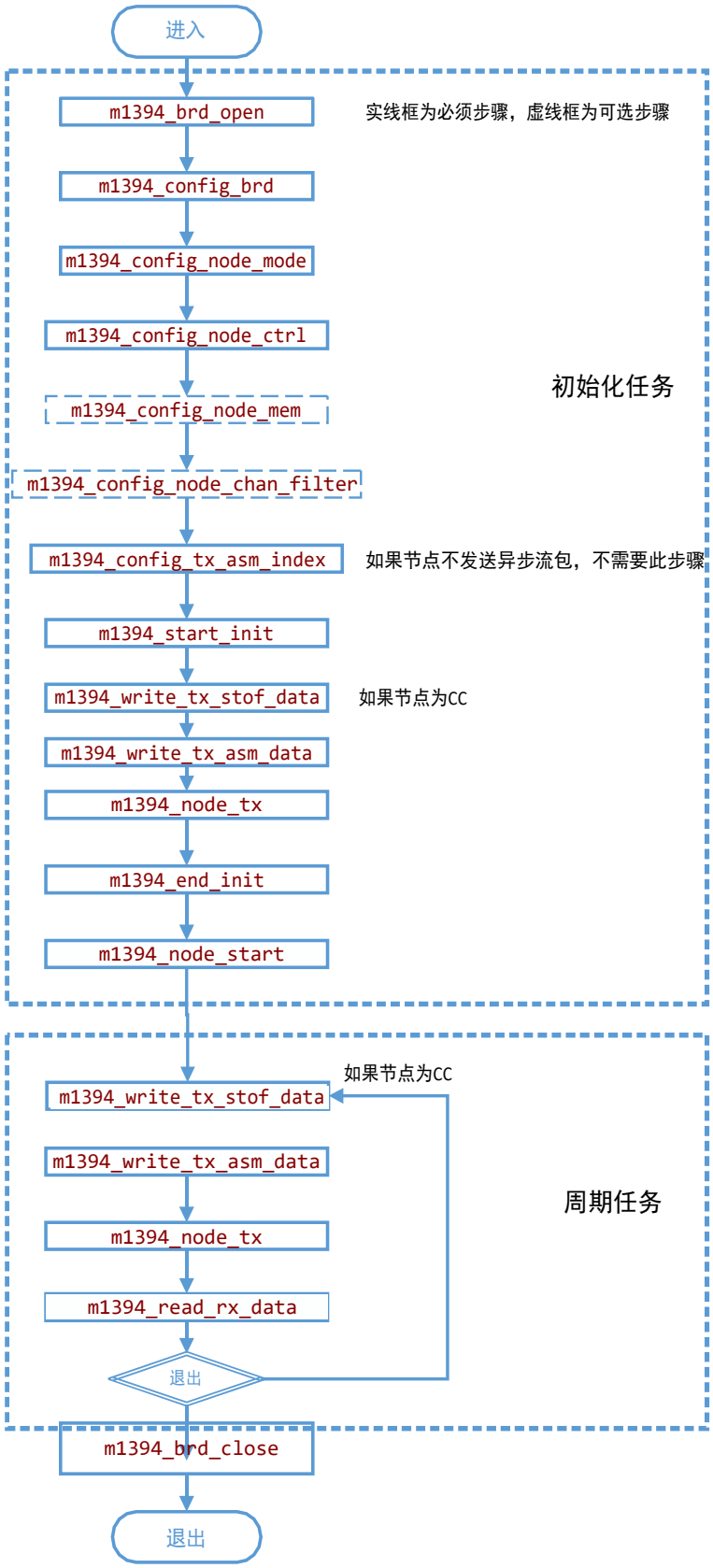
由于需要对硬件板卡进行操作，因此，用户应用程序需要使用“管理员权限”运行，否则将无法打开板卡。

2.1 软件使用流程图

应用软件从使用上基本分为上电初始化及周期任务，详细的函数调用过程如下图所示：

其中虚线框内的初始化函数的调用是可选的，其它是必须调用的。

注，图中的所有函数均应加 b9019_前缀。


进入
m1394_brd_open
实线框为必须步骤，虚线框为可选步骤
m1394_config_brd
m1394_config_node_mode
m1394_config_node_ctrl
初始化任务
m1394_config_node_mem
m1394_config_node_chan_filter
m1394_config_tx_asm_index
如果节点不发送异步流包，不需要此步骤
m1394_start_init
m1394_write_tx_stof_data
如果节点为CC
m1394_write_tx_asm_data
m1394_node_tx
m1394_end_init
m1394_node_start
如果节点为CC
m1394_write_tx_stof_data
m1394_write_tx_asm_data
m1394_node_tx
周期任务
m1394_read_rx_data
退出
m1394_brd_close
退出

3 软件详细接口说明

3.1 上电初始化函数

- `int b9019_m1394_brd_open(int brd)`

  功能说明：打开板卡。

  返回值：成功打开板卡返回值大于 0，失败时返回值小于 0

  输入参数：

  - brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3

- `int b9019_m1394_brd_rsset(int brd)`

  功能说明：复位板卡并清空接收缓冲区。

  返回值：无

  输入参数：

  - brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3

- `int b9019_m1394_config_brd(int brd, int speed, int node_en)`

  功能说明：设置板卡的总线速率，并设置各个节点的工作使能状态

  返回值：配置成功返回值大于 0，失败时返回值小于 0

  输入参数：

  - brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3
  - speed：总线、速率，0 表示 100MHz，1 表示 200MHz，2 表示 400MHz
  - node_en：bit7、bit6、bit5、bit4、bit3、bit2、bit1、bit0 分别表示 8 个节点的使能状态，1 表示使能，0 表示不使能

- `int b9019_m1394_config_node_mode(int brd, int node, unsigned int mode)`

  功能说明：设置 8 个节点的工作模式

  返回值：设置成功时返回值大于 0，失败时返回值小于 0

  输入参数：

  - brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3
  - node：节点号，0~7 分别表示节点 0 到节点 7
  - mode：工作模式，1 表示 CC 模式，2 表示 RN 模式

- `int b9019_m1394_config_node_ctrl(int brd, int node, unsigned int cc_ctrl, unsigned int frame_cycle , unsigned int bak1, unsigned int rn_use_cc_offset, unsigned int bak2)`

  功能说明：设置节点的控制信息

返回值：设置成功时返回值大于 0，失败时返回值小于 0

输入参数：

- brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3
- node：节点号，0~7 分别表示节点 0 到节点 7
- cc_ctrl：如果该节点为 CC 模式，则可以设置一些 CC 有关的控制信息，具体如表 1 所示。当D0配置为 0 即手动发送 STOF 数据包时，可以调用函数 b9019_m1394_manual_tx_stof_data进行一次手动发送操作
- frame_cycle：设置帧周期，时间单位为 us
- bak1：保留参数，可以输入 0
- rn_use_cc_offset：RN 是否使用 CC 第 1 个 ASM 包中的发送偏移时间，1 为使用，0 为不使用，此时，RN 将使用最后一个参数 min_tx_tim 设置的时间作为偏移时间
- bak2：保留参数，可以输入 10

bit 位置	说明
D31:D6	备用
D5	为 1 表示发送的 STOF 包中“**数据 CRC**”由硬件自动产生**错误**数据，该字应用层软件不可见，只影响链路层传输
D4	为 1 表示发送的 STOF 包中“**包头 CRC**”由硬件自动产生**错误**数据，该字应用层软件不可见，只影响链路层传输
D3	为 1 表示发送的 STOF 包中第 10 个字“**垂直校验数据（VPC）**”由硬件自动产生**错误**数值
D2	为 1 表示发送的 STOF 包中第 10 个字“**垂直校验数据（VPC）**”由硬件自动产生**正确**数值，为 0 表示该字由软件写入
D1	为 1 表示发送的 STOF 包中第 4 个字“**飞行器时间**”由硬件自动产生
D0	为 1 表示 CC 节点将自动发送 STOF 数据包，即每到帧周期时间，自动发送 STOF 包

表 1 CC 模式控制信息

- int b9019_m1394_config_node_chan_filter(int brd, int node, unsigned int filter_en, unsigned int filter_hi32, unsigned int filter_lo32)

  功能说明：设置节点的按通道号过滤接收功能

  返回值：设置成功时返回值大于 0，失败时返回值小于 0

  输入参数：

  - brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3
  - node：节点号，0~7 分别表示节点 0 到节点 7
  - filter_en：是否开启按通道号过滤接收功能，1 表示允许过滤，0 表示禁止过滤
  - filter_hi32：该参数为掩码形式，每一个 bit 为 1 表示允许接收相应通道号的数据，为 0 表示不接收相应通道号数据，因此，bit31 到 bit0 可以表示通道 63 到通道 32 共 32 个通道
  - filter_lo32：该参数为掩码形式，每一个 bit 为 1 表示允许接收相应通道号的数据，为 0 表示不接收相应通道号数据，因此，bit31 到 bit0 可以表示通道 31 到通道 0 共 32 个通道

- int b9019_m1394_config_node_msg_filter(int brd, int node, unsigned int filter_en, unsigned int filter_num, unsigned int filter_id[])

功能说明：设置节点的按消息号过滤接收功能

返回值：设置成功时返回值大于 0，失败时返回值小于 0

输入参数：

- brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3
- node：节点号，0～7 分别表示节点 0 到节点 7
- filter_en：是否开启按消息号过滤接收功能，1 表示允许过滤，0 表示禁止过滤
- filter_num：待过滤消息号的个数，取值范围为 0 到 8
- filter_id[]：待过滤消息的消息号数组，超过 8 个消息号时，只有前 8 个消息号有效

- int b9019_m1394_config_node_mem(int brd, int node, unsigned int rxbuf_len, unsigned int txbuf_asm_len, unsigned int bak)

功能说明：设置节点内各个缓冲区的大小，板卡初始化后，接收缓冲区缺省为 64K 字节，ASM 发送缓冲区缺省为 64K 字节。

返回值：设置成功时返回值大于 0，失败时返回值小于 0

输入参数：

- brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3
- node：节点号，0～7 分别表示节点 0 到节点 7
- rxbuf_len：数据接收缓冲区大小，单位为字节
- txbuf_asm_len：ASM 数据发送缓冲区大小，单位为字节
- bak：备份，固定传入 0

- int b9019_m1394_config_tx_asm_index(int brd, int node, unsigned int index, unsigned int ctrl_hi16, unsigned int len, unsigned int tx_time, unsigned int dest_chan_id)

功能说明：设置 ASM 数据发送索引信息

返回值：设置成功时返回值大于 0，失败时返回值小于 0

输入参数：

- brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3
- node：节点号，0～7 分别表示节点 0 到节点 7
- index：发送 ASM 数据的索引号，从 0 开始，依次增加
- ctrl_hi16：发送 ASM 数据区的控制信息，该控制信息为 16 位长度，位于 32 位无符号型整数的高 16 位，其具体含义如表 2 所示：
- len：发送 ASM 数据缓冲区的长度，该长度为包括 4 字节 1394 头、16 字节 ASM 头，数据区，以及 16 字节 ASM 包尾的全部长度
- tx_time:该 ASM 数据缓冲区相对于 STOF 包的发送偏移时间，如果是非背靠背模式发送，则发送偏移时间至少距离上一包的发送时间大于 30us，如果上一包的发送长度大，该距离时间还应再加上上一包数据所占用的总线时间。
- dest_chan_id：为本 ASM 数据包的目标通道地址

bit 位置	说明
D31	0 表示该索引为最后一个索引项，1 表示后续还有索引
D30	1 表示允许本包发送
D29	1 表示背靠背模式，0 表示非背靠背。背靠背模式是指本包数据的发送偏移时间紧挨着前一包数据的发送，不需要用户专门配置发送时间，非背靠背模式是指本包数据的发送偏移时间使用参数 tx_time 所指令的发送偏移时间。 第 1 包数据该位必须设置为 0，即非背靠背模式。 如果节点工作在 RN 模式，第 1 包数据的发送时间还受 b9019_m1394_config_node_ctrl 函数中 rn_use_cc_offset 参数的影响，该参数为 1 时，则第 1 包数据的发送时间使用收到的第 1 包 CC 发送的异步流包中包尾所配置的发送时间。
D28	设为 0
D27	设为 0
D26	1 表示本索引为单次发送，发送后本标志位会自动清 0。单次发送功能只有在 D30 为0 时才有效。
D25	1 表示使用软件产生“**软件 VPC 值**”，0 表示由硬件自动产生软件 VPC 值。“软件 VPC 值”即数据负载区中最后一个字
D24	1 表示发送**错误**“**软件 VPC 值**”(VPC 的值是相应数据的相加和)，0 表示发送**正确**“**软件 VPC 值**”(VPC 的值是相应数据的相加和然后取反)
D23	1 表示发送**错误**“**数据 CRC**”值
D22	1 表示发送**错误**“**包头 CRC**”值
D21	1 表示发送**错误**“**硬件 VPC 值**”
D20	1 表示硬件自动发送“**健康状态字**”信息（数据为硬件读入的物理层芯片信息）
D19	1 表示硬件自动发送 1394 包尾信息，0 表示软件自主填写 1394 包尾信息
D18	1 表示硬件自动发送数据包头中的“**节点 ID**”，信息来自链路层芯片读入的节点 ID
D17	1 表示硬件自动计算并发送数据包头中的“**心跳字**”，如果设为硬件计算“心跳字”，可以调用 *b9019_m1394_set_node_heartbeat_cycle* 函数设置心跳字的更新周期
D16	1 表示硬件自动计算并发送“**硬件 VPC 值**”。“硬件 VPC 值”即数据包尾的最后一个字

表 2　发送索引区控制信息说明

- `void b9019_m1394_start_init(int brd)`

  功能说明：开始启动发送缓冲区的数据初始化

  返回值：无

  输入参数：

  - brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3

- `unsigned int b9019_m1394_end_init(int brd)`

  功能说明：结束启动发送缓冲区的数据初始化

  返回值：如果所有初始化信息正确，则返回 0，否则返回无符号 32 位整数，其中高 2 位为节点号，低

输入参数：

- brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3

错误代码	错误描述
1	发送空间接收空间的总和超出最大值
2	所有发送 ASM 包的长度和大于配置的发送空间
3	CC 节点的周期时间为 0
4	CC 节点的STOF 包头配置错误，正确应该为 0x00281FA0

表 3　初始化检查错误代码说明

- `int b9019_m1394_node_start(int brd, int node, int rx_en, int tx_en);`

  功能说明：启动节点周期性任务

  返回值：成功时返回值大于 0，失败时返回值小于 0

  输入参数：

  - brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3
  - node：节点号，0～7 分别表示节点 0 到节点 7
  - rx_en：1 为启动接收功能，0 为保持当前接收状态不变
  - tx_en：1 为启动发送功能，0 为保持当前发送状态不变

- `int b9019_m1394_node_stop(int brd, int node, int rx_en, int tx_en);`

  功能说明：停止节点周期性任务

  返回值：成功时返回值大于 0，失败时返回值小于 0

  输入参数：

  - brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3
  - node：节点号，0～7 分别表示节点 0 到节点 7
  - rx_en：1 为停止接收功能，0 为保持当前接收状态不变
  - tx_en：1 为停止发送功能，0 为保持当前发送状态不变

- `int b9019_m1394_set_node_stof_heartbeat_en(int brd, int node, unsigned int cfg);`

  功能说明：设置STOF 数据包的心跳字配置

  返回值：成功时返回值大于 0，失败时返回值小于 0

  输入参数：

  - brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3
  - node：节点号，0～7 分别表示节点 0 到节点 7
  - cfg：bit0 到 bit8 中某1 位为1，表示STOF 数据包中相应的第1 个字（CC 状态）到第9 个字（Quadlet 8）为心跳字，该字的值会由板卡在每个STOF 周期自动加 1，同时软件将无法对该字进行写操作。

3.2 周期任务的数据发送接收函数

- `int b9019_m1394_write_tx_stof_data(int brd, int node, int auto_cword, void *pbuf, unsigned int len)`

  功能说明：更新STOF 发送缓冲区数据

  返回值：设置成功时返回值大于 0，失败时返回值小于 0

  输入参数：

  - brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3
  - node：节点号，0~7 分别表示节点 0 到节点 7
  - auto_cword：是否自动计算 1394 头控制字，1 为自动计算，0 为使用用户缓冲区提供的控制字
  - pbuf：待更新的 STOF 数据缓冲区地址，其格式如图 1 所示
  - len：待更新的 STOF 数据缓冲区的长度，单位为字节

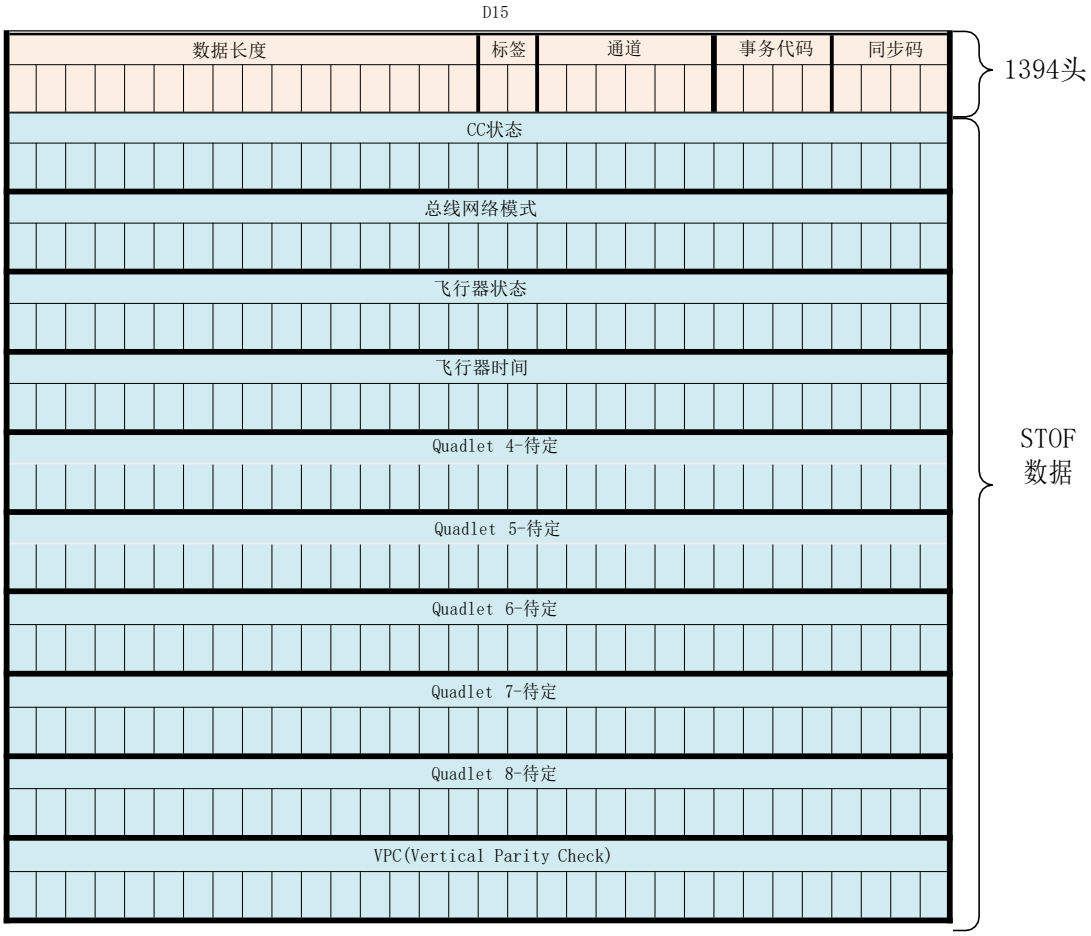


图 1　发送 STOF 数据缓冲区结构

- `int b9019_m1394_write_tx_asm_data(int brd, int node, int auto_cword, unsigned int index, void *pbuf, unsigned int len)`

  功能说明：更新 ASM 发送缓冲区数据

  返回值：设置成功时返回值大于 0，失败时返回值小于 0

  输入参数：

  - brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3
  - node：节点号，0～7 分别表示节点 0 到节点 7
  - auto_cword：是否自动计算 1394 头控制字，1 为自动计算，0 为使用用户缓冲区提供的控制字
  - pbuf：待更新的 ASM 数据缓冲区地址，其格式如图 2 所示
  - len：待更新的 ASM 数据缓冲区的长度，单位为字节

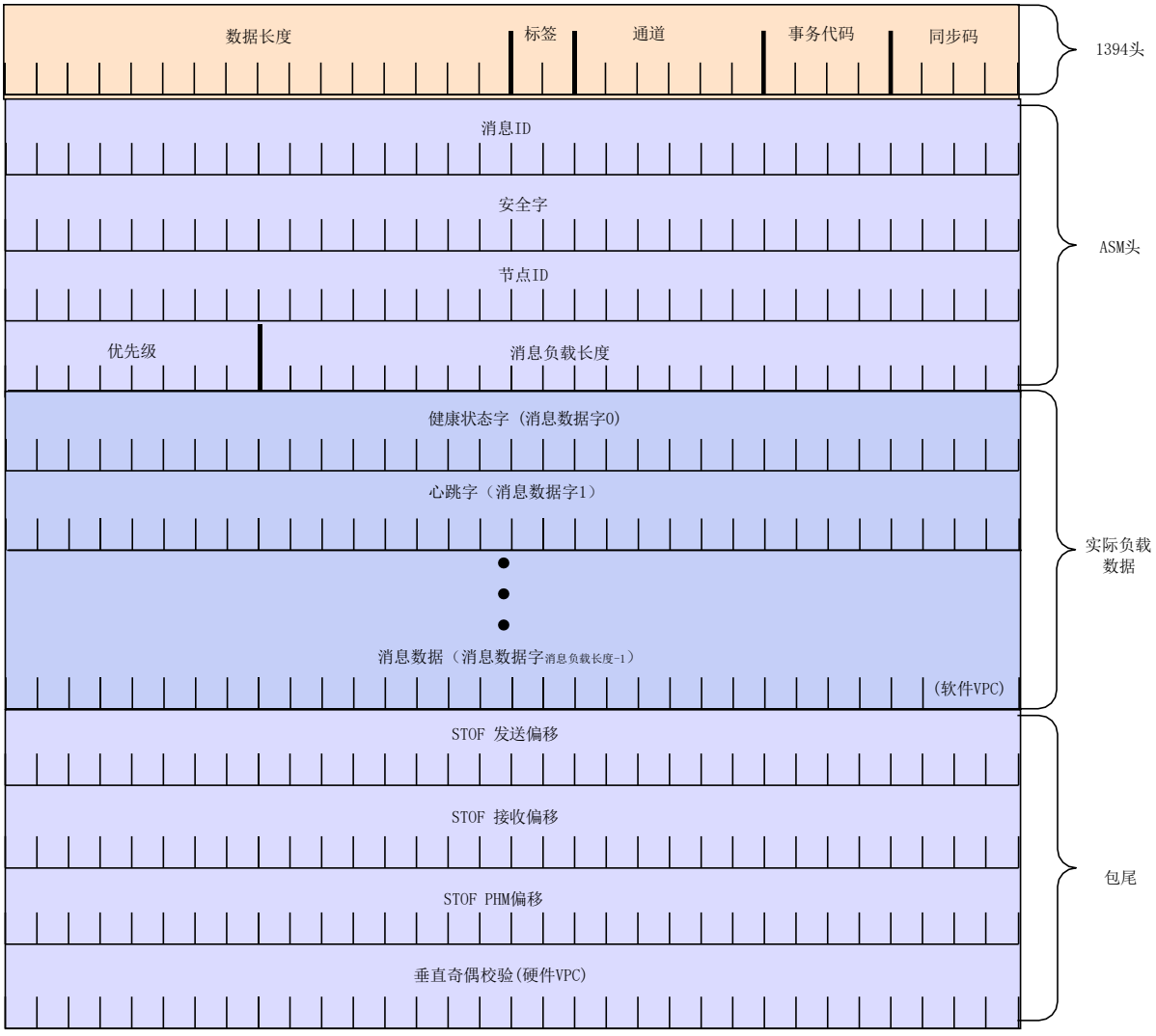


图 2　发送数据区格式

int b9019_m1394_node_tx(int brd, int node, int tx_stof, int tx_asm, int bak)功能说明：启动一次节点的数据发送功能

返回值：设置成功时返回值大于 0，失败时返回值小于 0

输入参数：

- brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3
- node：节点号，0~7 分别表示节点 0 到节点 7
- tx_stof：是否发送 STOF 数据，1 为发送，0 为不发送
- tx_asm：是否发送 CM 数据，1 为发送，0 为不发送
- bak：备份，固定传入 0

- int b9019_m1394_read_rx_data(int brd, int len, void * pbuf)

功能说明：读取节点接收 FIFO 中的数据，该数据以一个STOF 周期内的所有数据为单位

返回值：成功时返回值大于 0，范围为 1 到 4，分别表示该数据为节点 0 到节点 7 中的某个节点数据。

返回值为 0 表示没有读取到数据，返回值-1 表示读取错误

输入参数：

- brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3
- len：期望获取的数据的最大长度，这个长度的建议值参考 pbuf 参数的说明
- pbuf：读取到的数据存放缓冲区，若读取到数据，则该缓冲区中的数据结构如图 3 所示。其头部第 1 部分为帧描述符区，第 2 部分为 STOF 数据区，第 3 部分为接收数据索引区，第 4 部分为真正的数据区。帧描述符与接收索引的数据结构如图 4 所示。
  实际使用时，用户需要尽可能给一个较大的缓冲区（推荐 32K 字节）以保证缓冲区不会溢出。用户可以不用关心该缓冲区的具体结构，可以使用后面其它的辅助类 API 函数对此缓冲区进行解析。

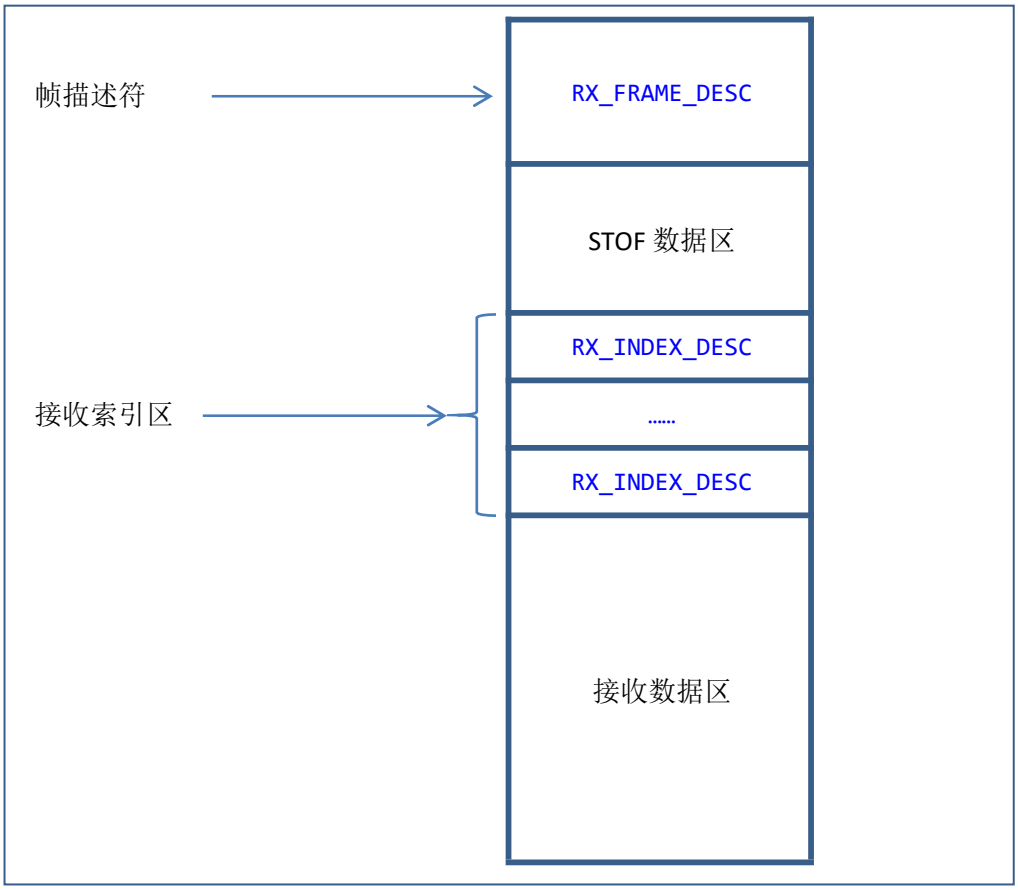


图 3　接收缓冲区结构组成

```
typedef   struct   _RX_FRAME_DESC_   {
    unsigned int      node              ;  // 0, 0x000，表示该数据为哪个节点的数据
    unsigned int      stof_addr         ;  // 1, 0x004，STOF数据相对于本缓冲区的偏移地址
    unsigned int      stof_len          ;  // 2, 0x008，STOF数据的长度
    unsigned int      asm_index_addr    ;  // 3, 0x00C，ASM索引区相对于本缓冲区的偏移地址
    unsigned int      asm_index_len     ;  // 4, 0x010，ASM索引区的长度
    unsigned int      asm_data_addr     ;  // 5, 0x014，ASM数据区相对于本缓冲区的偏移地址
    unsigned int      asm_data_len      ;  // 6, 0x018，ASM数据区的长度
    unsigned int      bak               ;  // 7, 0x01C，备份，当前未使用
} RX_FRAME_DESC, * PRX_FRAME_DESC;
```

```
typedef   struct   _RX_INDEX_DESC_   {
    unsigned int      ctrl_len          ;  // 0, 0x000，低16位表示数据包的长度，高16位表示控制信息
    unsigned int      delay_time        ;  // 1, 0x004，数据包的传输延时
    unsigned int      time_addr         ;  // 2, 0x008，低16位表示数据包在缓冲区中的起始地址，高16位表示数
据接收的本地时间的低位，单位为0.02微秒
    unsigned int      time_tag          ;  // 3, 0x00C，数据接收的本地时间的高位，单位为125微秒
```

图 4　接收缓冲区的帧结构及索引结构

- `int b9019_m1394_rx_buf_index_num(void * pbuf)`

  功能说明：获取接收缓冲区中的索引个数

返回值：当前缓冲区中的索引个数，其值大于等于 0

输入参数：

- pbuf：接收缓冲区的首地址，其中的数据一般由函数 b9019_m1394_read_rx_data 获取

- int b9019_m1394_rx_buf_stof_addr(void * pbuf)

功能说明：获取接收缓冲区中 STOF 数据包的偏移地址，STOF 数据包的数据格式参见图 1 发送 STOF 数据缓冲区结构。

返回值：当前缓冲区中的 STOF 数据包的偏移地址

输入参数：

- pbuf：接收缓冲区的首地址，其中的数据一般由函数 b9019_m1394_read_rx_data 获取

- int b9019_m1394_rx_buf_index_info(void * pbuf, int index, unsigned int *plen, unsigned int *pspd, unsigned int *perr_code, unsigned int *perr_vpc, unsigned int *ptime_tag_hi, unsigned int *ptime_tag_lo)

功能说明：获取接收缓冲区中的索引的详细信息

返回值：成功时返回值大于 0

输入参数：

- pbuf：接收缓冲区的首地址，其中的数据一般由函数 b9019_m1394_read_rx_data 获取
- index：缓冲区中索引的序号，最小值为 0，最大值由函数 b9019_m1394_rx_buf_index_num 获取
- plen：返回该索引的数据区的字节大小
- pspd：返回该索引数据包的总线速率，0 为 100MHz，1 为 200MHz，2 为 400MHz
- perr_code：返回数据包的错误代码，当前未使用
- perr_vpc：返回数据的 VPC 校验状态，1 表示校验出错
- ptime_tag_hi：返回本包数据接收时的本地时间的高 32 位，其单位为 125us
- ptime_tag_lo：返回本包数据接收时的本地时间的低 16 位，其单位为 20ns

- int b9019_m1394_rx_buf_index_data(void * pbuf, int index)

功能说明：获取接收缓冲区中的索引数据本身的地址，其值为相对于本缓冲区首地址的偏移地址，缓冲区中数据的格式参见图 2 发送数据区格式。

返回值：索引数据在当前缓冲区的偏移地址

输入参数：

- pbuf：接收缓冲区的首地址，其中的数据一般由函数 b9019_m1394_read_rx_data 获取
- index：缓冲区中索引的序号，最小值为 0，最大值由函数 b9019_m1394_rx_buf_index_num 获取

- int b9019_m1394_rx_buf_chan_data(void * pbuf, unsigned int chan)

功能说明：获取接收缓冲区中的相应通道号的数据包的偏移地址，其值为相对于本缓冲区首地址的偏移地址

返回值：数据包在当前缓冲区的偏移地址，如果数据包不存在，则偏移地址为 0

输入参数：

- pbuf：接收缓冲区的首地址，其中的数据一般由函数 b9019_m1394_read_rx_data 获取
- chan：数据包对应的通道号

- int b9019_m1394_rx_buf_stof_addr(void * pbuf)

功能说明：获取接收缓冲区中的STOF 包的偏移地址，其值为相对于本缓冲区首地址的偏移地址

返回值：数据包在当前缓冲区的偏移地址，如果数据包不存在，则偏移地址为 0

输入参数：

- pbuf：接收缓冲区的首地址，其中的数据一般由函数 b9019_m1394_read_rx_data 获取

- int b9019_m1394_rx_buf_stof_info(void * pbuf, int index, unsigned int dw_num, unsigned int *stat, unsigned int *ptime_tag_hi, unsigned int *ptime_tag_lo)

功能说明：获取接收缓冲区中的STOF 包的附加信息

返回值：成功时返回值大于 0

输入参数：

- pbuf：接收缓冲区的首地址，其中的数据一般由函数 b9019_m1394_read_rx_data 获取
- dw_num：备用
- stat：备用
- ptime_tag_hi：返回本包数据接收时的本地时间的高 32 位，其单位为 125us
- ptime_tag_lo：返回本包数据接收时的本地时间的低 16 位，其单位为 20ns

3.3 其它辅助函数

- int b9019_m1394_tx_asm_single(int brd, int node, unsigned int index)

功能说明：启动一次单次发送型ASM 包的发送

返回值：成功时返回值大于 0，失败时返回值小于 0

输入参数：

- brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3
- node：节点号，0～7 分别表示节点 0 到节点 7
- index：发送 ASM 数据的索引号，从 0 开始，依次增加

- int b9019_m1394_get_fifo_len(int brd, unsigned int *plen)

功能说明：读取板卡接收 FIFO 中当前数据帧的个数

返回值：成功时返回值大于 0，失败时返回值小于 0输入参数：

brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3

plen：存放返回数据帧个数的变量地址

- int b9019_m1394_rx_buf_len(void * pbuf)

  功能说明：读取接收缓冲区的有效长度

  返回值：成功时返回值大于 0，失败时返回值小于 0

  输入参数：

  - pbuf：接收缓冲区的首地址

- int b9019_m1394_manual_tx_stof_data(int brd, int node, int auto_cword, void *pbuf, unsigned int len)

  功能说明：手动发送STOF 数据包

  返回值：发送成功时返回值大于 0，失败时返回值小于 0

  输入参数：

  - brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3
  - node：节点号，0～7 分别表示节点 0 到节点 7
  - auto_cword：是否自动计算 1394 头控制字，1 为自动计算，0 为使用用户缓冲区提供的控制字
  - pbuf：待更新的 STOF 数据缓冲区地址，其格式如图 1 所示
  - len：待更新的 STOF 数据缓冲区的长度，单位为字节

- int b9019_m1394_set_node_heartbeat_cycle(int brd, int node, unsigned int stof_num)

  功能说明：在硬件自动计算心跳字的情况下配置心跳字的更新周期，单位为 STOF 个数。

  返回值：成功时返回值大于 0，失败时返回值小于 0

  输入参数：

  - brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3
  - node：节点号，0～7 分别表示节点 0 到节点 7
  - stof_num：心跳字更新的 STOF 周期，例如如果设为 3，表示心跳字每 3 个 STOF 周期加 1

- int b9019_m1394_set_node_stof_heartbeat_en(int brd, int node, unsigned int cfg)

  功能说明：将STOF 数据包里某个地址的数据设置为自动心跳字。

  返回值：成功时返回值大于 0，失败时返回值小于 0

  输入参数：

  - brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3

node：节点号，0~7 分别表示节点 0 到节点 7

- cfg：有效 bit 为bit0 到 bit8，当 bit0 位设置为 1 时，表示 STOF 数据包内第一个字（图 1 中的“CC 状态”字）为心跳字，并且该心跳字将由硬件自动进行加 1。

- void b9019_m1394_node_update_asm_index(int brd, int node, int index, unsigned int ctrl_hi16)

  功能说明：修改异步流包发送索引的控制字。

  返回值：无

  输入参数：

  - brd：板卡序号，第 1 块板卡为 1，第 2 块板卡为 2，第 3 块板卡为 3
  - node：节点号，0~7 分别表示节点 0 到节点 7
  - index：发送异步流包的索引号，参见函数 b9019_m1394_config_tx_asm_index 的说明。
  - ctrl_hi16：发送异步流包的控制字，参见函数b9019_m1394_config_tx_asm_index 的说明。

4 文件内容

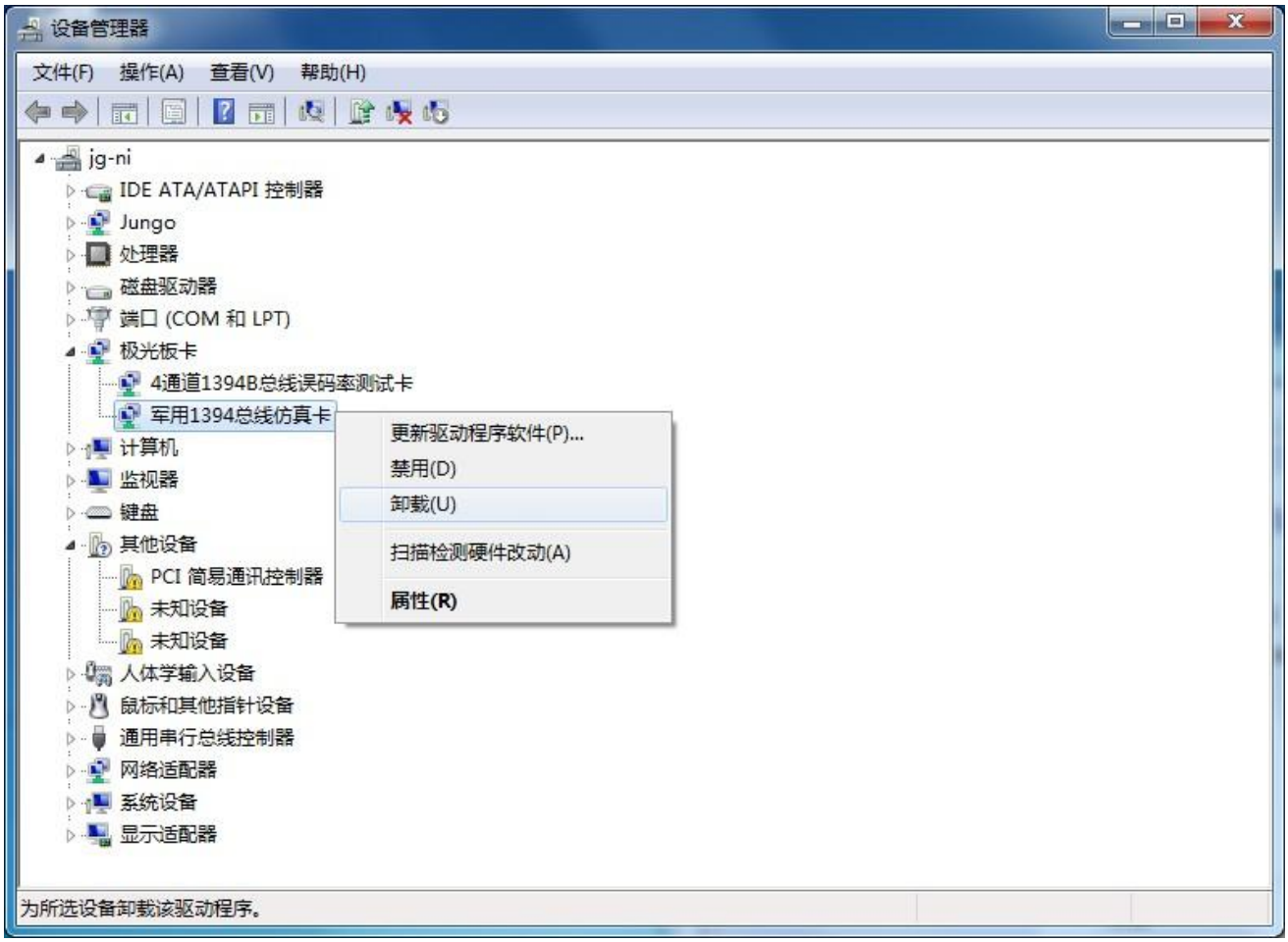
本卡以动态库的方式为用户提供编程接口，包括以下文件：

1) win/x86/dll_1394d.dll 32 位 Windows 系统动态库运行文件
2) win/x86/dll_1394d.lib 32 位 Windows 系统动态库链接文件
3) win/x86/b9019_m1394_api.h 32 位 Windows 系统 API 接口头文件
4) win/x64/dll_1394d.dll 64 位 Windows 系统动态库运行文件
5) win/x64/dll_1394d.lib 64 位 Windows 系统动态库链接文件
6) win/x64/b9019_m1394_api.h 64 位 Windows 系统 API 接口头文件
7) lin/x64/libfwbs.so 64 位 Linux 系统动态库运行文件
8) lin/x64/libfwbs.a 64 位 Linux 系统静态库链接文件
9) lin/x64/b9019_m1394_api.h 64 位 Linux 系统 API 接口头文件

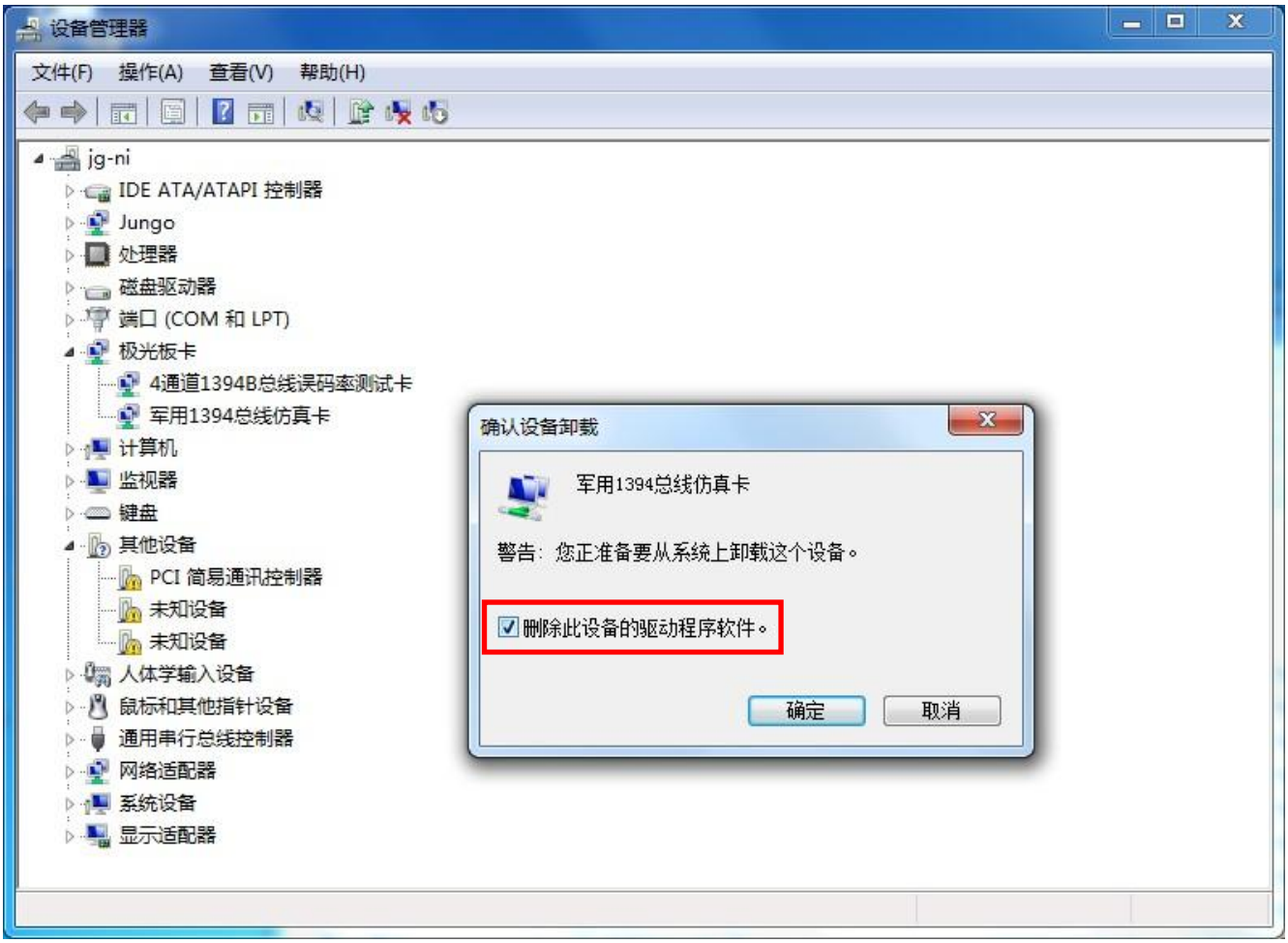
5 驱动更新

如果有新的驱动需要升级时，按照以下步骤进行操作：

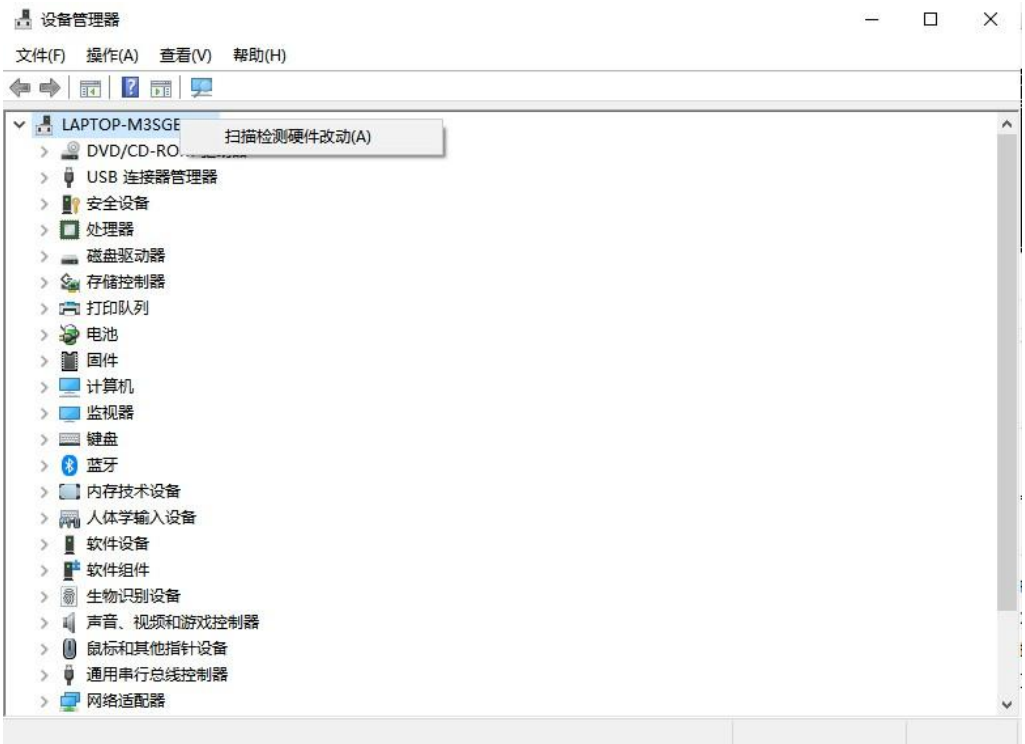
1. 在操作系统的设备管理器对话框中，选中需要操作的板卡，然后点击右键，选择“卸载设备”，如下图所示：



2. 在新出现的对话框中选中“删除此设备的驱动程序软件”，然后点击“卸载”按钮卸载原驱动程序，如下图所示：



3. 在操作系统的设备管理器对话框中，右击最顶层的项，然后选择“扫描检测硬件改动”，如下图所示：



经过以上步骤后，在设备管理器对话框中会出现未安装驱动程序的板卡图标，然后按照正常的驱动程序安装过程安装新的驱动程序即可。

6 板卡连接器定义

6.1 SCSI-68 型连接器

从板卡连接器看过去，连接器的定义如下：

1接口SCSI-68

SCSI-68 管脚号	信号定义		SCSI-68 管脚号	信号定义
1	GND		35	GND
2	N2-P2A+		36	N2-P2B+
3	N2-P2A-		37	N2-P2B-
4	GND		38	GND
5	N2-P1A+		39	N2-P1B+
6	N2-P1A-		40	N2-P1B-
7	GND		41	GND
8	N2-P0A+		42	N2-P0B+
9	N2-P0A-		43	N2-P0B-
10	GND		44	GND
11	N3-P2A+		45	N3-P2B+
12	N3-P2A-		46	N3-P2B-
13	GND		47	GND
14	N1-P2A+		48	N1-P2B+
15	N1-P2A-		49	N1-P2B-

16	GND		50	GND
17	N1-P1A+		51	N1-P1B+

SCSI-68 管脚号	信号定义		SCSI-68 管脚号	信号定义
18	N1-P1A-		52	N1-P1B-
19	GND		53	GND
20	N1-P0A+		54	N1-P0B+
21	N1-P0A-		55	N1-P0B-
22	N3-P1A+		56	N3-P1B+
23	N3-P1A-		57	N3-P1B-
24	N3-P0A+		58	N3-P0B+
25	N3-P0A-		59	N3-P0B-
26	N0-P2A+		60	N0-P2B+
27	N0-P2A-		61	N0-P2B-
28	GND		62	GND
29	N0-P1A+		63	N0-P1B+
30	N0-P1A-		64	N0-P1B-
31	GND		65	GND
32	N0-P0A+		66	N0-P0B+
33	N0-P0A-		67	N0-P0B-
34	GND		68	GND

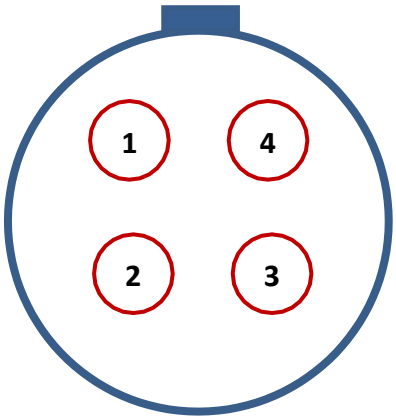
2接口SCSI-68

SCSI-68 管脚号	信号定义		SCSI-68 管脚号	信号定义
1	GND		35	GND
2	N6-P2A+		36	N6-P2B+
3	N6-P2A-		37	N6-P2B-
4	GND		38	GND
5	N6-P1A+		39	N6-P1B+
6	N6-P1A-		40	N6-P1B-
7	GND		41	GND
8	N6-P0A+		42	N6-P0B+
9	N6-P0A-		43	N6-P0B-
10	GND		44	GND
11	N7-P2A+		45	N7-P2B+
12	N7-P2A-		46	N7-P2B-
13	GND		47	GND
14	N5-P2A+		48	N5-P2B+
15	N5-P2A-		49	N5-P2B-
16	GND		50	GND
17	N5-P1A+		51	N5-P1B+

SCSI-68 管脚号	信号定义		SCSI-68 管脚号	信号定义
18	N5-P1A-		52	N5-P1B-
19	GND		53	GND
20	N5-P0A+		54	N5P0B+
21	N5-P0A-		55	N5P0B-
22	N7-P1A+		56	N7-P1B+
23	N7-P1A-		57	N7-P1B-
24	N7-P0A+		58	N7-P0B+
25	N7-P0A-		59	N7-P0B-
26	N4-P2A+		60	N4-P2B+
27	N4-P2A-		61	N4-P2B-
28	GND		62	GND
29	N4-P1A+		63	N4-P1B+
30	N4-P1A-		64	N4-P1B-
31	GND		65	GND
32	N4-P0A+		66	N4-P0B+
33	N4-P0A-		67	N4-P0B-
34	GND		68	GND

6.2 单端口 LEMO 型连接器

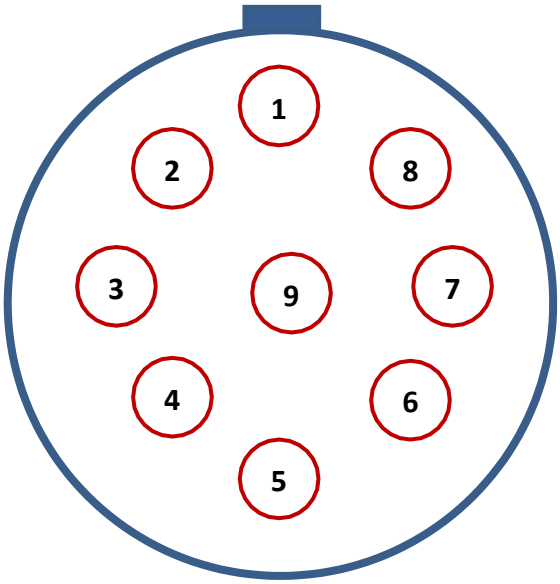
从板卡插座的方向看进去：



管脚号	信号定义
1	PA-
2	PB-
3	PA+
4	PB+

6.3 双端口 LEMO 型连接器

从板卡插座的方向看进去：



管脚号	信号定义
1	P1A+
2	P1A-
3	P1B+
4	P1B-
5	P2A+
6	P2A-
7	P2B+
8	P2B-
9	GND

7 版本历史

- 2019/12/08 Rev 1.0 正式发布
- 2020/04/01 Rev 1.1
  - 重新整理并明确了表 2 中的一些信息；
  - 添加了 LEMO 型连接器定义。
- 2020/05/02 Rev 1.2
  - 修改了 b9019_m1394_read_rx_data 函数说明中的参数定义，去掉了一个多余的参数

- 添加了 b9019_m1394_rx_buf_len 函数的说明。

- 2020/07/01 Rev 1.3
  - 增加了 b9019_m1394_manual_tx_stof_data 函数说明
- 2020/11/26 Rev 1.4
  - 更正了表 2 中 D31 位的说明，D31 位 0 表示该索引为最后一个索引，V1.3 版中将此位错写成了 1 表示该索引为最后一个索引。
- 2020/12/15 Rev 1.5
  - 添加了表 2 中 D26 位单次发送功能的说明及函数 b9019_m1394_config_tx_asm_index 的说明。
- 2021/02/28 Rev 1.6
  - 添加主机接收 FIFO 缓冲区的说明。
- 2021/03/01 Rev 1.7
  - 修改了 b9019_m1394_config_node_ctrl 函数中参数的说明，去掉了背靠背参数；
  - 增加了表 2 中 D29 位发送索引背靠背模式的说明。
- 2021/05/25 Rev 1.8
  - 修改了 b9019_m1394_config_tx_asm_index 函数中背靠背控制位的说明；
  - 增加了 b9019_m1394_set_node_stof_heartbeat_en 函数说明。
- 2021/08/26 Rev 1.9
  - 修改了 STOF 数据包图中最后一个字，由 CRC 改为 VPC；
  - 将节点内部缓冲区大小由 32K 字节修改为 128K 字节。
- 2022/02/09 Rev 1.10
  - 增加了 b9019_m1394_node_update_asm_index 函数说明。