

1 节点 3 端口 1394B 总线仿真板卡 PXIe/CPCI/PCIe/PCI/PMC 总线

用

户

手

册

目 录

第一章 概述	1
1.1 关于本手册	2
1.2 产品描述	2
1.2.1 特性	2
1.2.2 一般规格	3
1.3 产品安装	3
1.3.1 安装之前的准备	3
1.3.2 硬件安装	4
1.3.3 驱动安装	6
第二章 硬件说明	9
2.1 连接器和信号定义	10
2.1.1 连接器型号	10
2.1.2 接器点号定义	10
第三章 驱动程序编程接口	13
3.1 系统架构	14
3.1.1 节点数据缓冲区	15
3.1.2 内存接收 FIFO 缓冲区	15
3.2 软件使用总体流程	15
3.2.1 软件使用流程图	17
3.3 软件详细接口说明	18
3.3.1 上电初始化函数	18
3.3.2 周期任务的数据发送接收函数	25
3.3.3 其它辅助函数	31
3.4 文件内容	33
第四章 功能演示软件	34
4.1 使用环境	35
4.1.1 硬件	35
4.1.2 操作系统	35
4.1.3 开发工具	35
4.2 界面展示	35
4.3 使用说明	36
4.3.1 基本设置	36
4.3.2 总线模式	36
4.3.3 数据显示	37

第一章 概述

1.1 关于本手册

本手册适用于下列产品型号：

- ZXB-CPCI-2130 CPCI 总线 1 节点，3 端口，1394B 总线仿真板卡
- ZXB-PCI-2130 PCI 总线 1 节点，3 端口，1394B 总线仿真板卡
- ZXB-PCIe-2130 PCIe 总线 1 节点，3 端口，1394B 总线仿真板卡
- ZXB-PXIe-2130 PXIe 总线 1 节点，3 端口，1394B 总线仿真板卡
- ZXB-PMC-2130 PMC 总线 1 节点，3 端口，1394B 总线仿真板卡
- ZXB-ETH-2130 RJ45 总线 1 节点，3 端口，1394B 总线仿真板卡

MIL-1394B 总线产品基于处理 AS5643 数据流内容的监控和分析，支持 MIL-1394B 协议和 AS5643 协议标准，可满足故障注入和总线检测等功能。

MIL-1394B 总线产品架构提供硬件层面支持 MIL-1394B (SAE AS5643)，在包封装、数据提取、收发传输 STOF 过程中极大降低了对计算机硬件的要求。MIL-1394B 总线通讯定时完全由产品硬件实现，极大的提高了定时准确性，最大程度的保证了验证的可靠性。

注意

在使用该产品之前，请您详细阅读本手册各章节的内容。

1.2 产品描述

MIL-1394B 总线产品是我公司自主研发的军用航电总线产品，本系列产品包含 1 个独立的 1394B 节点，每个节点具有 3 个端口，每个节点可以根据用户配置工作在 CC、RN 或 BM 模式。其强大的功能能够满足不同用户的测量和自动化控制需求，良好的兼容性适用于各类系统配置。

1.2.1 特性

- 标准 PXIe/CPCI/PCIe/PCI 总线 3U 板卡
- 符合 IEEE 1394b-2008 规范，满足 SAE AS5643 协议；
- 支持 1 节点，每个节点扩展 3 个 Port。
- 支持 S100、S200、S400 传输速率，Beta 工作模式，自主可配置。
- 端口模式默认为 S400，支持更改为 S200、S100；
- 支持 CC、RN 仿真和 BM 监控模式，模式自主可配置；
- 支持按配置表进行消息调度，配置表可由软件加载；

- 监控的数据具有高精度时间戳，分辨率 200ns；
- 基于硬件控制的高精度总线同步及消息调度，周期和偏移分辨率为 1us；
- 支持发送异步流包数据故障注入，包括 VPC、HCRC、DCRC 错误，以及总线复位等，支持软件精确控制输入的故障数量；
- CC 模式下，支持多种 STOF 发送模式；
- STOF 包可注入故障，包括 VPC、HCRC、DCRC、周期错误等；
- BM 下支持对总线所有数据和行为、事件进行长时间监控记录；
- 支持总线全监控或选择性监控记录；
- RN 模式工作时，RN 发送偏移支持有两种方式：可以通过本地配置获取；可以从 CC 至 RN 的异步流包中的发送偏移；
- 支持总线拓扑显示、根节点检测、端口状态获取等；
- 发送异步流包心跳支持应用更新或自动更新，自动更新频率可配置；
- 支持周期发送和软件触发发送两种模式，以及两种模式的混合应用；
- 支持用户定义异步流包负载区二次校验算法；
- 接收数据包支持硬件层正确性校验；
- 支持收发 128 条消息配置。

1.2.2 一般规格

- 物理尺寸：标准 PXI/PXIe/CPCI 3U 卡尺寸，160mm×100mm×20mm，公差小于 0.2mm
标准 PCI/PCIe 尺寸 175mm×106mm，公差小于 0.2mm
- 连接器：SCSI68 母座
- 工作功耗：5V/1.5A
- 相对湿度：5~95%，无凝结

1.3 产品安装

1.3.1 安装之前的准备

1. 在您安装产品之前请检查包装是否完好，以确定产品在运输的过程中没有遭到损坏。
如果包装发现有破损，请马上与运输商联系。
1. 在打开包装后请检查产品以及配件的完整性。打开产品外包装后，您应该发现如下产品：

- 板卡
- 产品合格证
- 产品配套光盘
- 标配连接器

如有规格不符,请您立刻联系我们,我们将负责维修或者更换。

2. 如果有可能,请您准备防静电工作台并佩戴防静电腕带。如果不具备以上静电防护装备,请接触计算机设备的导地部分,例如机箱壳金属部分,以释放身体上的静电。

现在您可以准备安装板卡了。

1.3.2 硬件安装

第一步: 打开板卡的防静电包装袋,取出板卡。

① 注意

手持板卡时,请尽量只接触板卡的边缘。在板卡安装到计算机设备之前,请将板卡平放,置于防静电包装袋中,这样有利于保护板卡不受静电损伤。取出板卡后,请保留产品的防静电和防震包装,以便在不使用时,产品可以妥善存放。



图 1



图 2

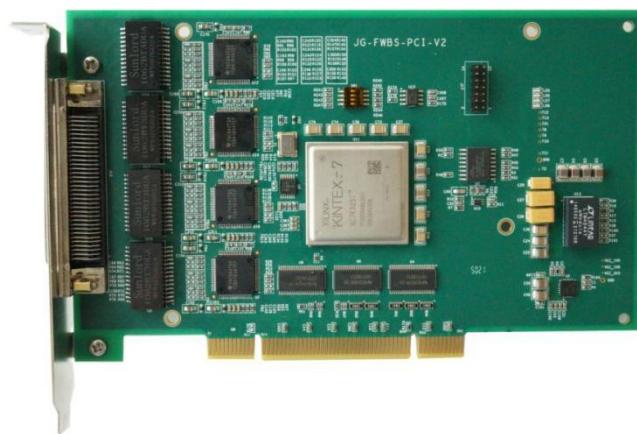


图 3



图 4



图 5

第二步： 关闭计算机设备的电源，将板卡安装到您的计算机机箱内。

第三步： 将配套的连接器或连接线缆插到板卡的连接器接口上。

关于连接电缆的制作请参照接口部分的内容。

开启计算机，系统提示发现新硬件，安装产品的驱动。

1. 3. 3 驱动安装

在产品配套光盘的“驱动安装”目录中，可以找到该设备的驱动程序。按如下步骤安装产品的驱动程序：

1) 设备初次接入工控机时，需对该设备进行驱动程序安装；



图 1 PCI 设备检测

2) 鼠标右键“我的电脑”，选择“管理”，选择“设备管理器”，在“其他设备”下选择“PCI 设备”，右键选择“更新驱动程序”，驱动搜索路径指定为配套光盘文件存放路径；



图 2 驱动程序搜索路径指定

- 3) 路径指定完成后，选择“下一步”，出现如下页面，勾选“始终信任来自……”，点击“安装”；

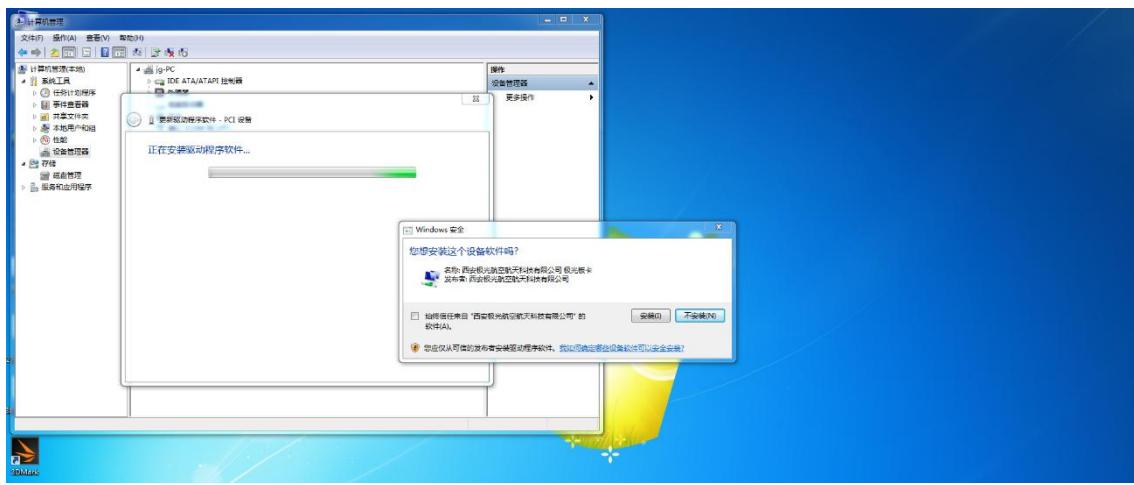


图 3 设备驱动程序安装

- 4) 稍等片刻后，出现如下页面，该设备的驱动程序便安装成功了。

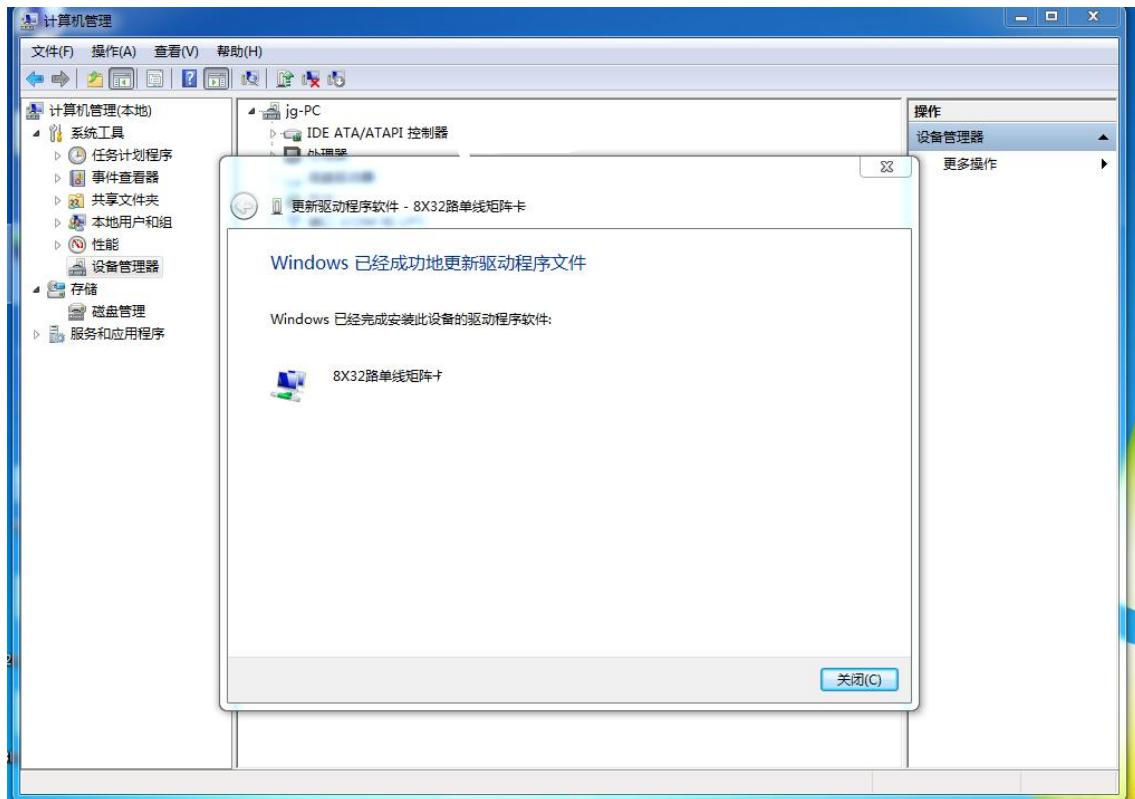


图 4 设备驱动安装成功

在完成板卡驱动安装后，可以通过计算机系统的“设备管理器”来确认设备驱动是否正确安装。访问“设备管理器”可以通过“此电脑” / “管理” / “设备管理器”。如果板卡驱动正确安装，可以在“设备管理器”的设备列表中看到板卡设备项，如图所示。

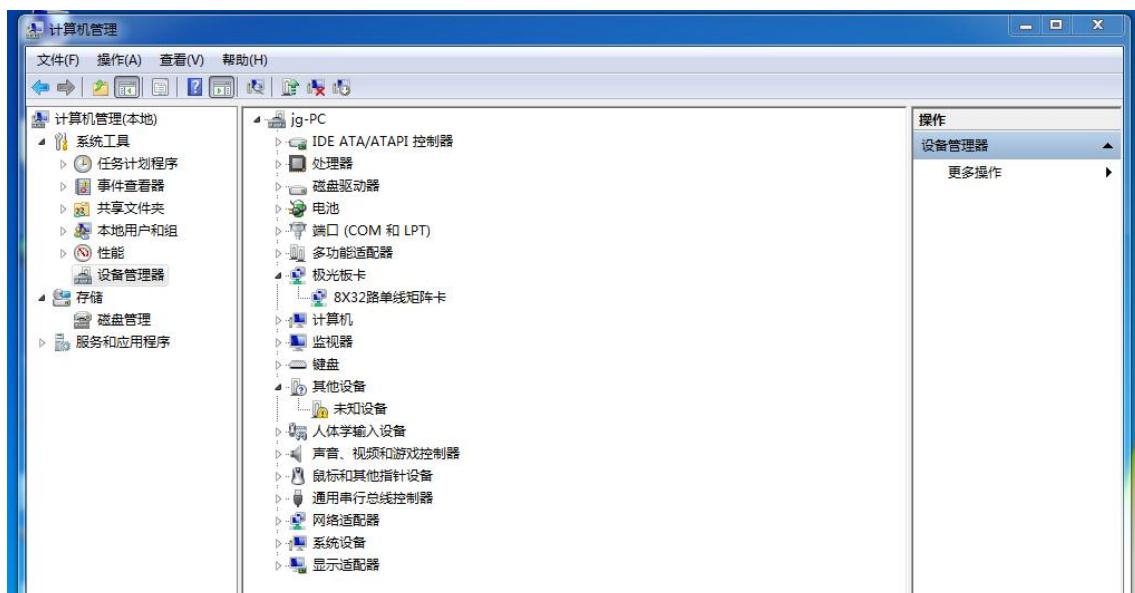


图 5 设备驱动检查

第二章 硬件说明

本章描述了该板卡的硬件信息，包括硬件设置、连接器和信号定义等。

2.1 连接器和信号定义

2.1.1 连接器型号

本产品采用 SCSI68 母座标准接头。



2.1.2 连接器点号定义

从板卡连接一端看过去，连接器的定义如下：



按照管脚排序如下表所示：

管脚	信号定义	说明
1	GND	
2	N/A	
3	N/A	
4	GND	
5	N/A	
6	N/A	
7	GND	
8	N/A	
9	N/A	
10	GND	
11	N/A	
12	N/A	
13	GND	
14	N/A	
15	N/A	
16	GND	
17	N/A	
18	N/A	

管脚	信号定义	说明
35	GND	
36	N/A	
37	N/A	
38	GND	
39	N/A	
40	N/A	
41	GND	
42	N/A	
43	N/A	
44	GND	
45	N/A	
46	N/A	
47	GND	
48	GND	
49	N/A	
50	N/A	
51	GND	
52	N/A	

19	GND		53	GND	
20	N/A		54	N/A	
21	N/A		55	N/A	
22	N/A		56	N/A	
23	N/A		57	N/A	
24	N/A		58	N/A	
25	N/A		59	N/A	
26	NO-P2A+	接收+	60	NO-P2B+	发送+
27	NO-P2A-	接收-	61	NO-P2B-	发送-
28	GND		62	GND	
29	NO-P1A+	接收+	63	NO-P1B+	发送+
30	NO-P1A-	接收-	64	NO-P1B-	发送-
31	GND		65	GND	
32	NO-POA+	接收+	66	NO-POB+	发送+
33	NO-POA-	接收-	67	NO-POB-	发送-
34	GND		68	GND	

按照信号分组排序如下表所示：

信号定义	管脚	说明
NO-POB+	66	发送+
NO-POB-	67	发送-
NO-POA+	32	接收+
NO-POA-	33	接收-
GND	31	
NO-P1B+	63	发送+
NO-P1B-	64	发送-
NO-P1A+	29	接收+
NO-P1A-	30	接收-
GND	28	
NO-P2B+	60	发送+
NO-P2B-	61	发送-
NO-P2A+	26	接收+
NO-P2A-	27	接收-
GND	62	

按照信号名称排序如下表所示：

节点号	P0					P1					P2				
	TX+	TX-	RX+	RX-	GND	TX+	TX-	RX+	RX-	GND	TX+	TX-	RX+	RX-	GND
0	66	67	32	33	31	63	64	29	30	28	60	61	26	27	62

注：N/A 不可用信号，保持悬空

2.1.3 LEMO 型连接器

从板卡插座连接一端看过去，连接器的定义如下：

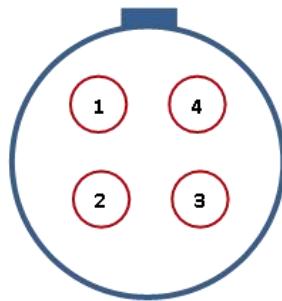


图 6 LMEO 型连接器示意图

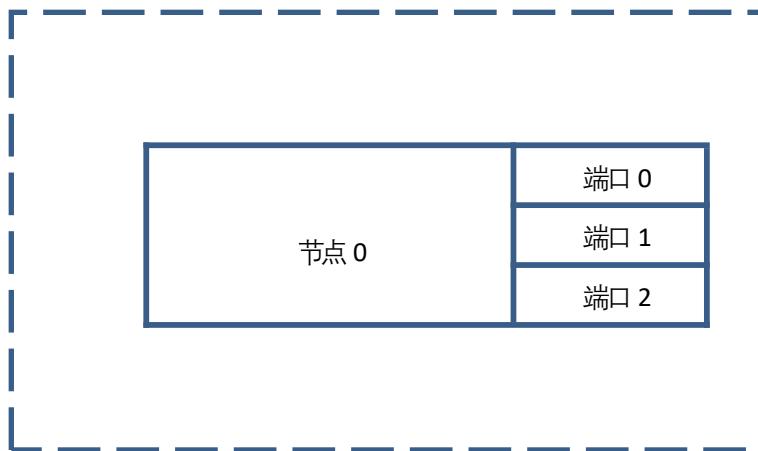
管脚号	信号定义
1	PA-
2	PB-
3	PA+
4	PB+

第三章 驱动程序编程接口

本章主要讲述了如何使用 1394B 板卡的驱动程序接口，为用户编程提供参考。1394B 驱动程序提供了丰富的接口函数，能满足用户对板卡的操作需求；具有良好的兼容性，能适用于多种编程环境；操作简单方便，可以大大缩短用户的开发周期。

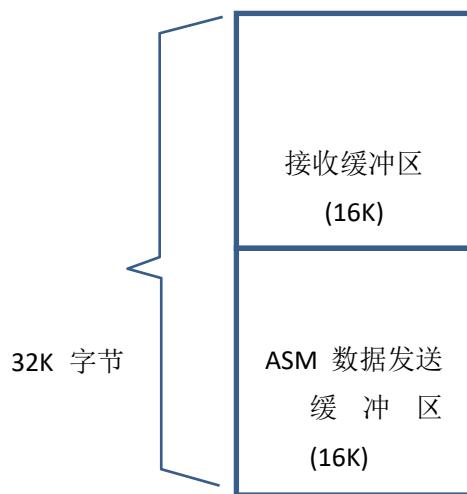
3.1 系统架构

本仿真卡内实现了 1 个独立的 1394 节点，每个节点具有 3 个端口，每个节点可以根据用户配置工作在 CC、RN 或 BM 模式，总体结构如下图所示：



3.1.1 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 32K 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 32K 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 16K 字节, ASM 数据发送缓冲区 16K 字节。



3.1.2 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 `m1394_get_fifo_len` 得到当前 FIFO 缓冲区中还剩余的数据包个数。

3.2 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 m1394_brd_open(int brd);`
- `void m1394_brd_close(int brd);`
- `void m1394_brd_reset(int brd)`
- `int m1394_config_brd(int brd, int speed, int node_en);`
- `int m1394_config_node_mode(int brd, int node, unsigned int mode);`
- `int m1394_config_node_mem(int brd, int node, unsigned int rdbuf_len, unsignedint tdbuf_asm_len, unsigned int bak);`
- `int m1394_config_node_ctrl(int brd, int node, unsigned int cc_ctrl, unsigned int frame_cycle, unsigned int b2b_mode, unsigned int rdbuf_use_cc_offset, unsigned int min_tx_time);`
- `int m1394_config_node_chan_filter(int brd, int node, unsigned int filter_en, unsigned int filter_hi32, unsigned int filter_lo32);`
- `int m1394_config_node_msg_filter(int brd, int node, unsigned int filter_en, unsigned int filter_num, unsigned int filter_id[]);`
- `int 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 m1394_start_init(int brd);`
- `unsigned int m1394_end_init(int brd);`
- `int m1394_node_start(int brd, int node, int rx_en, int tx_en);`
- `int 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 m1394_write_tx_stof_data(int brd, int node, int auto_cword, void *pbuf, unsigned int len);`
- `int m1394_write_tx_asm_data(int brd, int node, int auto_cword, unsigned int index, void *pbuf, unsigned int len);`
- `int m1394_node_tx(int brd, int node, int tx_stof, int tx_asm, int bak);`
- `int m1394_read_rx_data(int brd, int len, void * pbuf);`

3. 其它辅助函数

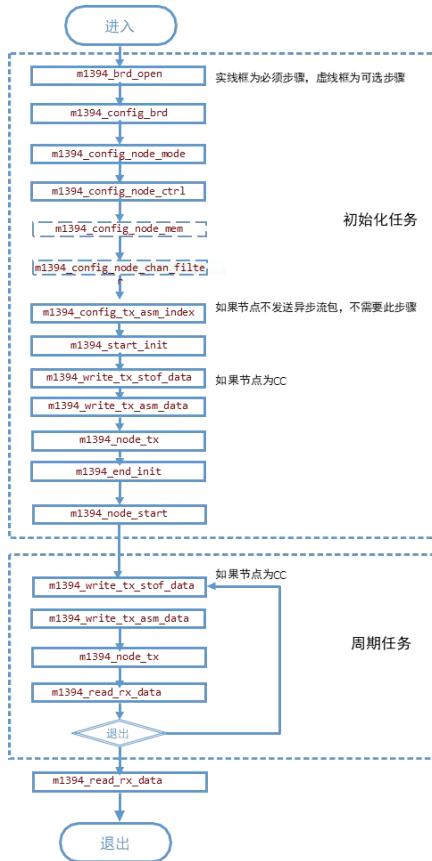
- `int m1394_node_stop(int brd, int node, int rx_en, int tx_en);`

- `int m1394_rx_buf_index_num(void * pbuf);`
- `m1394_rx_buf_index_info(void* pbuf,intindex,unsignedint*plen,unsignedint*pspd, unsigned int *perr_code, unsigned int *perr_vpc, unsigned int *ptime_tag_hi, unsigned int *ptime_tag_lo);`
- `int m1394_rx_buf_index_data(void * pbuf, int index);`
- `int m1394_rx_buf_chan_data(void * pbuf, unsigned int chan);`
- `int m1394_get_fifo_len(int brd, unsigned int *plen)`
- `int m1394_rx_buf_len(void * pbuf)`
- `int m1394_manual_tx_stof_data(int brd, int node, int auto_cword, void *pbuf, unsigned int len);`
- `int m1394_rx_buf_stof_addr(void * pbuf)`
- `int m1394_rx_buf_stof_info(void* pbuf,intindex,unsignedintdw_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。

3.2.1 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。



3.3 软件详细接口说明

3.3.1 上电初始化函数

- `int 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 m1394_brd_rst(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 m1394_config_brd(int brd, int speed, int node_en)` 功能说明：设置板卡的总线
西安市高新区沣惠南路 36 号橡树街区 B 座 1503 室 联系电话：029-88867449/13474137544 18

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:** bit3、bit2、bit1、bit0 分别表示 4 个节点的使能状态，1 表示使能，0 表示不使能

● `int m1394_config_node_mode(int brd, int node, unsigned int mode)`

功能说明：设置 4 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 ~ 3 分别表示节点 0 到节点 3
- **mode:** 工作模式，1 表示 CC 模式，2 表示 RN 模式

● `int 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 ~ 3 分别表示节点 0 到节点 3
 - **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 `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_time` 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 (VPCY) ”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 (VPCY) ”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 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 ~ 3 分别表示节点 0 到节点 3
- **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 m1394_config_node_msg_filter(int brd, int node, unsigned int filter_en, unsignedint 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 ~ 3 分别表示节点 0 到节点 3
- `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 m1394_config_node_mem(int brd, int node, unsigned int rdbuf_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 0x4000 字节，ASM 发送缓冲区缺省为 0x4000 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 ~ 3 分别表示节点 0 到节点 3
- `rdbuf_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 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 ~ 3 分别表示节点 0 到节点 3
- **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 rn1394_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)

Bit 位置	说明
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 <code>m1394_set_node_heartbeat_cycle</code> 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 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 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

30 位为错误代码，错误代码的详细信息如表 3 所示。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 初始检查错误代码说明

- `int 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 ~ 3 分别表示节点 0 到节点 3
- `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 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 ~ 3 分别表示节点 0 到节点 3
- `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 为启动发送功能，0 为保持当前发送状态不变

- `int 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 ~ 3 分别表示节点 0 到节点 3
- `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。

3.3.2 周期任务的数据发送接收函数

- `int m1394_write_tx_stof_data(int brd, int node, int auto_cword, void *pbuf, unsignedint 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 ~ 3 分别表示节点 0 到节点 3
- **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

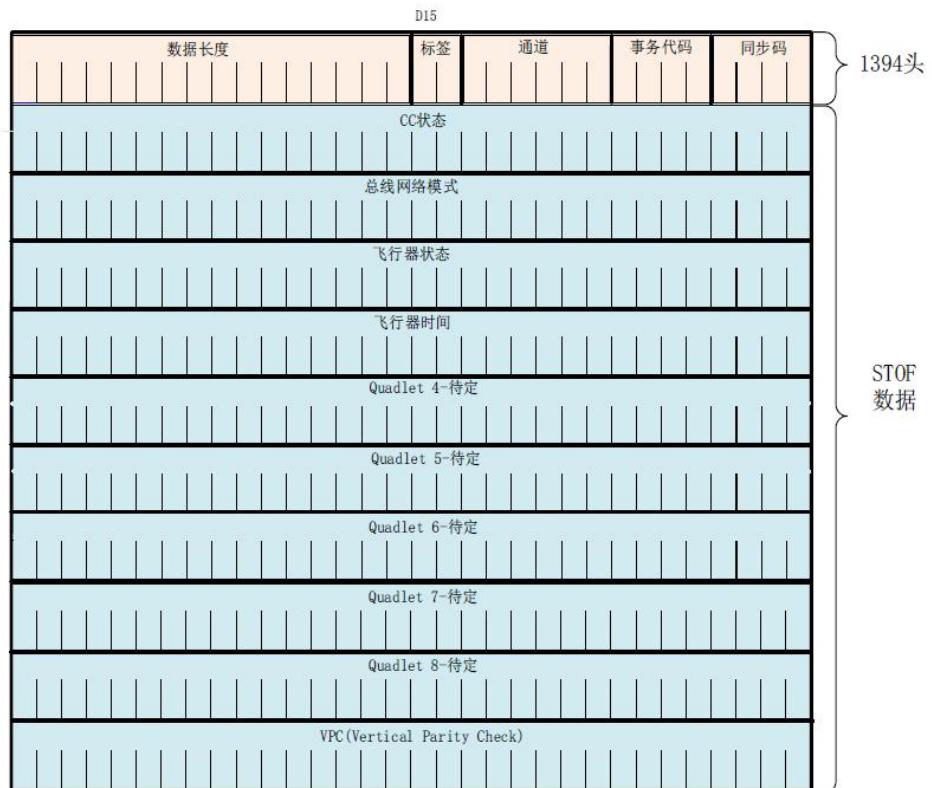


图 1 发送 STOF 数据缓冲区结构

- `int 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 ~ 3 分别表示节点 0 到节点 3
- `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

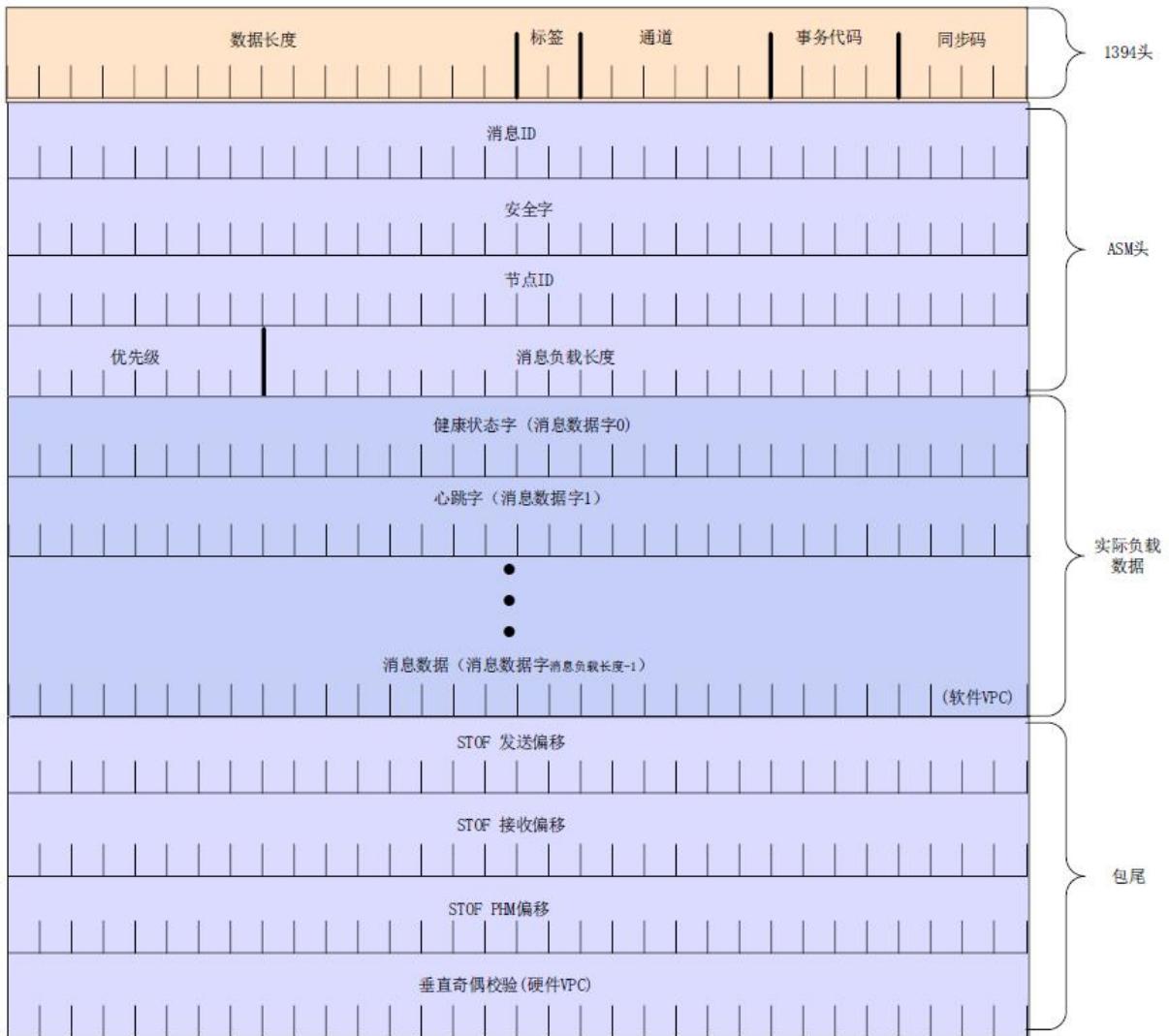


图 2 发送数据区格式

- `int 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 ~ 3 分别表示节点 0 到节点 3
- `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 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 3 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。

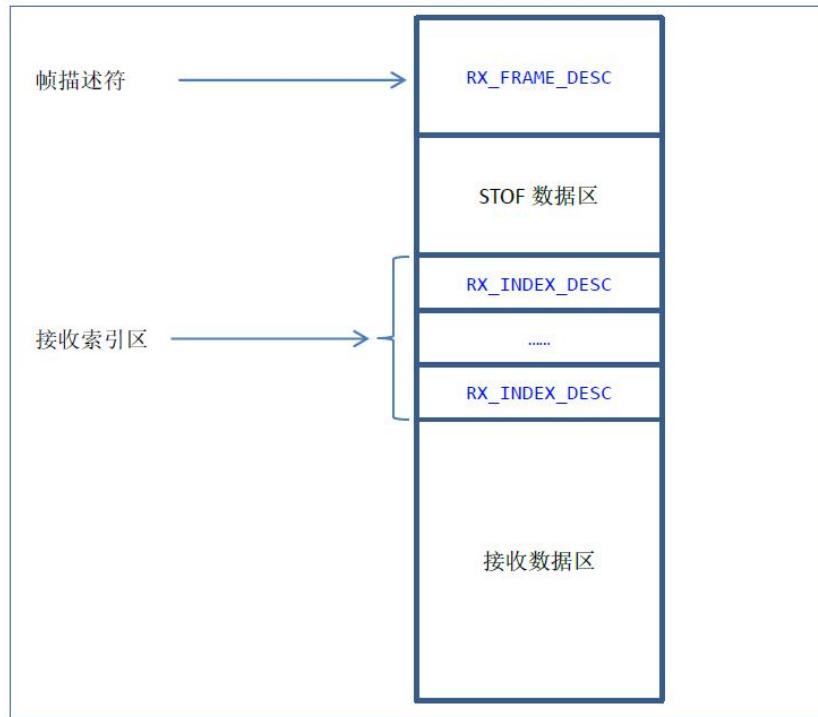


图 3 接收缓冲区结构组成

```
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
} RX_INDEX_DESC, * PRX_INDEX_DESC;
```

图 4 接收缓冲区的帧结构及索引结构

● `int 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 `m1394_read_rx_data` 获取

- `int 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 `m1394_read_rx_data` 获取

- `intm1394_rx_buf_index_info(void*pbuff,intindex,unsignedint*plen,unsignedint`

```
*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：

- **pbuff**: 接收缓冲区的首地址, 其中的数据一般由函数 `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 `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 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 `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 `m1394_rx_buf_index_num` 获取
- `int 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 `m1394_read_rx_data` 获取
- **chan:** 数据包对应的通道号

- `int 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 `m1394_read_rx_data` 获取

- `int 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 `m1394_read_rx_data` 获取
- **dw_num:** 本包数据中包含的 32 位数据字的个数
- **stat:** bit1~bit0: 传输速率

bit2: 备用

bit7~bit3: 接收确认码, 含义如下:

- 1: 确认完成
- 2: 确认挂起
- 4: 确认忙 X
- 5: 确认忙 A

- 6: 确认忙 B
- 11: 确认延迟
- 12: 确认冲突错误
- 13: 确认数据错误
- 14: 确认类型错误
- 15: 确认地址错误
- 16: 未收到确认
- 17: 确认信息过长, 超过 8 个bit
- 18: 确认信息过短, 少于 8 个bit
- 其它码: 保留

bit8: 0 表示 VPC 校验正确, 1 表示 VPC 校验错误

bit9: 1 表示本包数据传输正确, 0 表示本包数据传输错误

bit12~bit10: 备用

bit13: 0 表示飞行时间正确, 1 表示飞行时间错误

bit15~bit14: 备用

- o [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
- o [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

3.3.3 其它辅助函数

- [int 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:

- o **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
- o **node**: 节点号, 0 ~ 3 分别表示节点 0 到节点 3
- o **index**: 发送 ASM 数据的索引号, 从 0 开始, 依次增加

- [int 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 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 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, 0~3 分别表示节点 0 到节点 3
- **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 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, 0~3 分别表示节点 0 到节点 3
- **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 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 ~ 3 分别表示节点 0 到节点 3
 - **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 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 ~ 3 分别表示节点 0 到节点 3
- **index:** 发送异步流包的索引号, 参见函数 **m1394_config_tx_asm_index** 的说明。
- **ctrl_hi16:** 发送异步流包的控制字, 参见函数 **m1394_config_tx_asm_index** 的说明。

3.4 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/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/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/m1394_api.h	64 位	Linux	系统 API 接口头文件

第四章 功能演示软件

板卡功能演示软件可用于实现和演示本公司 1394B 总线仿真板卡通讯功能。利用本软件，可轻松地对板卡进行操作。

应用程序主要包含以下 5 个图形化界面：

- 板卡选择界面：选择需要打开的板卡
- 主界面：程序主界面
- 配置界面：工作模式、加载配置
- 查询功能：发送包数、接收包数、STOF 包数、复位次数
- 复位操作：Node 复位、Logic 复位 LLC 复位、PHY 复位

4.1 使用环境

4.1.1 硬件

- 内存：128M 以上
- CPU：200MHz 以上
- 显示分辨率：800×600 以上

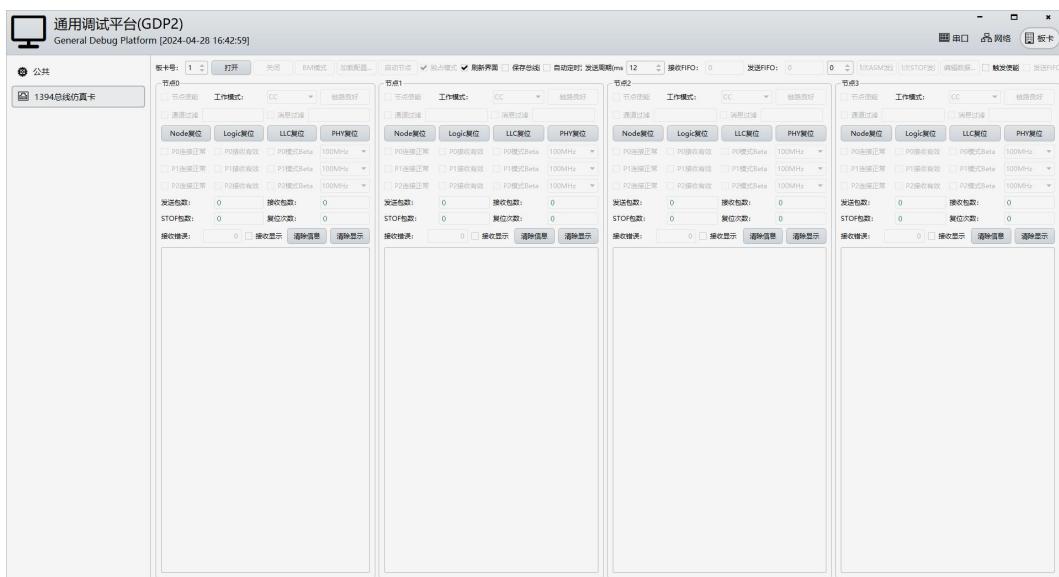
4.1.2 操作系统

Windows 98/2000/2003/xp/Win7 操作系统

4.1.3 开发工具

Microsoft Visual Studio C++ 7.1

4.2 界面展示



4.3 使用说明

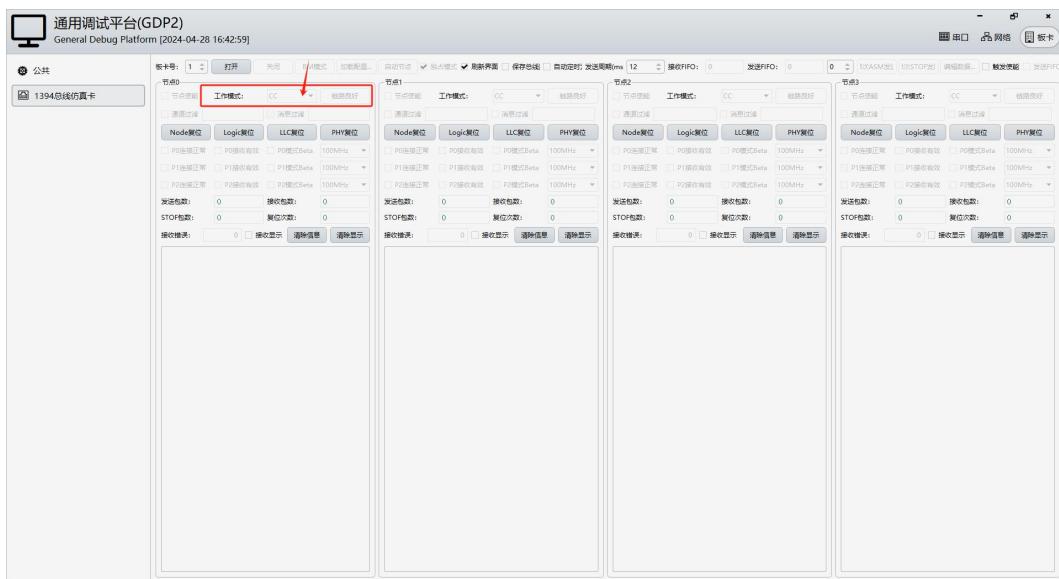
运行演示程序，进入主界面，点击“1394 总线仿真卡”。

4.3.1 基本设置

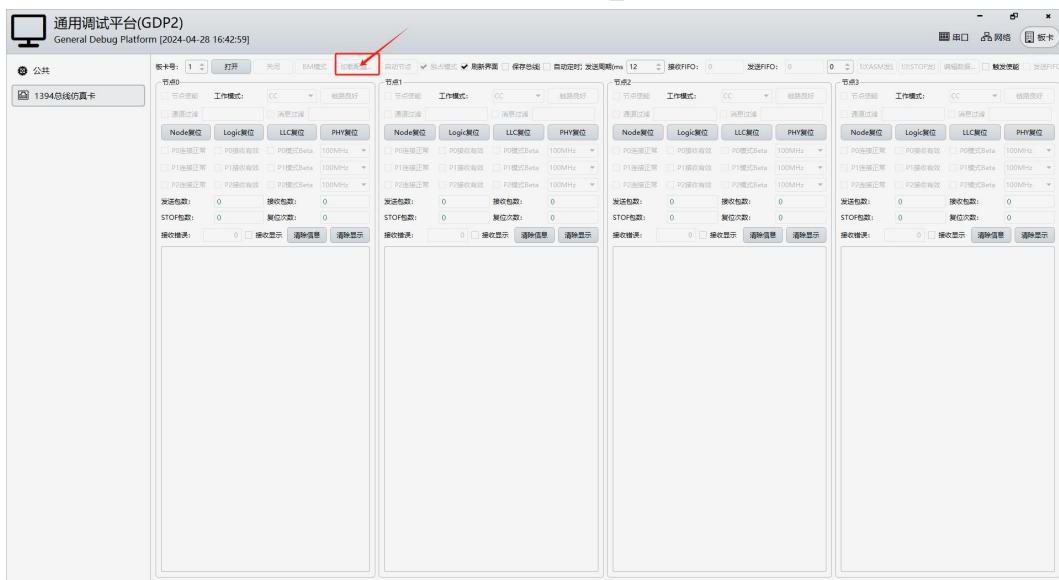
- 1、 打开主程序在    中选择板卡号，默认为“1”；
- 2、 单击    中的“打开”按钮。

4.3.2 总线模式

在如图位置，选择总线工作模式。



在如图位置，进行加载配置，1 节点板卡建议选择：1cc_3rn。



4.3.3 数据显示

1、选择对应通道，勾选“使能”按钮；

2、检查数据区显示情况；

