
STW-PCIE-B 板卡

产品使用说明书

共 34 页

成都同相科技有限公司

声明

本文档包含了成都同相科技有限公司生产的光纤双向频率传递设备的主要功能、技术指标、工作原理、使用及操作说明、产品齐套等内容。成都同相科技有限公司保留在未提前声明的情况下，更改产品和本文档内容的权利，请随时关注 <http://www.sync-tech.com> 网站得到产品最新信息。

成都同相科技有限公司是一家致力于高精度时间频率设备研发和生产的高新技术企业。公司不断将最新的研究成果转化为高性能产品，始终为用户提供优质的产品和服务。主要研发的产品：

频率基准：铯原子钟、铷原子钟、高性能晶振，驯服铷钟基准等

钟组配套：隔离放大器，脉冲分配器，高性能微跃器，无缝切换器等

时频模组：高稳定度 DDS 模块，分配器模块，倍频器模块，分频器模块等

时频传输：卫星时频终端、光纤双向时频传递、卫星共视、微波双向比对

测试仪器：时频综测仪、频率稳定度测试仪，相噪测试仪，多通道比相仪等

科研信号源：铯原子微波源，铷原子微波源，低相噪频率源

时频产品已广泛应用于移动通信、电力、交通、计量测试、天文观测、航天测控、国防军工等部门。

成都同相科技有限公司

公司地址：四川省成都市高新西区天辰路 88 号 4 栋 3 单元

北京办公室：北京市海淀区上地十街辉煌国际大厦 4 号楼

销售电话：186 1177 6749（孙经理） 185 1140 1798（郑经理）

技术支持：186 1168 5224（任工）

<http://www.sync-tech.com>

Email: slh@sync-tech.com

Copyright © 2021, 成都同相科技有限公司

本文档中涉及的其他相关产品版权归相应公司所有。

目 录

1.产品介绍.....	4
2.功能指标.....	4
2.1 功能说明.....	4
2.2 指标说明.....	4
3.结构及外观.....	5
4.工作原理.....	6
5.使用与操作说明.....	7
5.1 安装与测试.....	7
5.2 软件开发说明.....	9
5.3 板卡硬件接口说明.....	27
5.4.板卡测试方法.....	28
5.5 板卡协议.....	33
6.常见故障排除及维修.....	34
7.运输和贮存.....	34
8.产品齐套.....	34

成都同相科技有限公司

1.产品介绍

STW-PCIE-B 板卡是高精度时间输入与输出的 PCIE 半高尺寸授时板卡，具备接收 GPS/北斗，B(DC)，1588 PTP，NTP，CDMA，1PPS+TOD 等功能。通过 PCIE 总线输出高精度时间信息，地理位置等信息。参考无效后，自动进入守时模式。STW-PCIE-B 板卡还具有输出频率可配置的 PCIE 中断到计算机的功能。

2.功能指标

2.1 功能说明

1. 可输入 GPS/北斗，B(DC)，1588V2 PTP，1PPS+TOD(可选)，1PPS 等时间源。
2. 具备接收 CDMA 授时功能(可选)。
3. 可输入外 10M 作为板卡时钟源。
4. 可输出 1588V2 PTP，NTP，BDC, 1PPS+TOD(可选)。
5. 可通过 PCIE 驱动读取板卡状态，时间信息，地理位置等信息，通过驱动进行输入信号选择，中断频率 1-1000hz 可配置，网络参数以及 1588V2 PTP /NTP 相关参数配置。
6. 自动守时功能。
7. PCIE 驱动支持 windows，linux 等常用操作系统。

2.2 指标说明

GPS/北斗：优于 50ns

B(DC)：优于 50ns

1588 PTP：优于 50ns（背靠背）

NTP：优于 20us（背靠背）

CDMA：优于 1ms（选配）

3.结构及外观

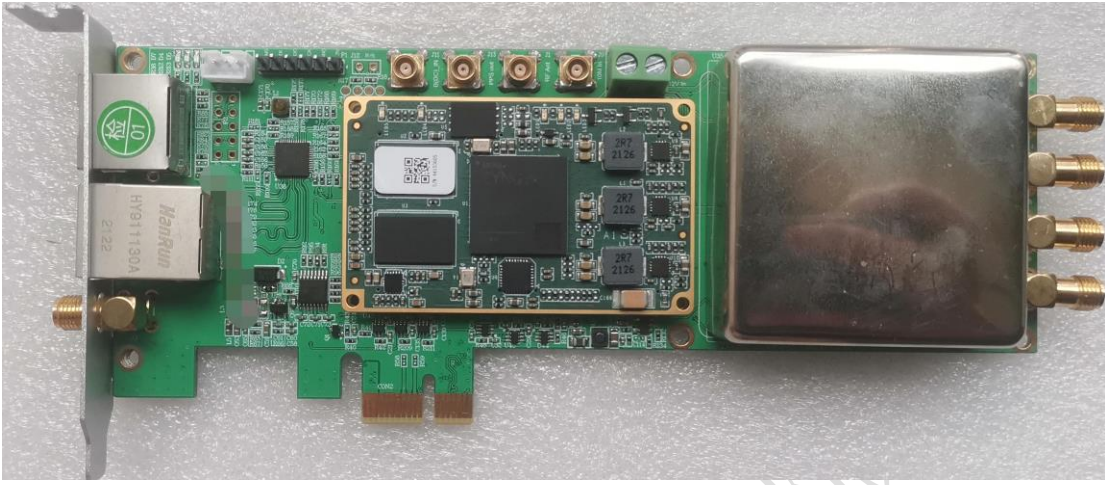


图 3-1-1 正面图



图 3-1-2 背面图



图 3-1-3 面板图

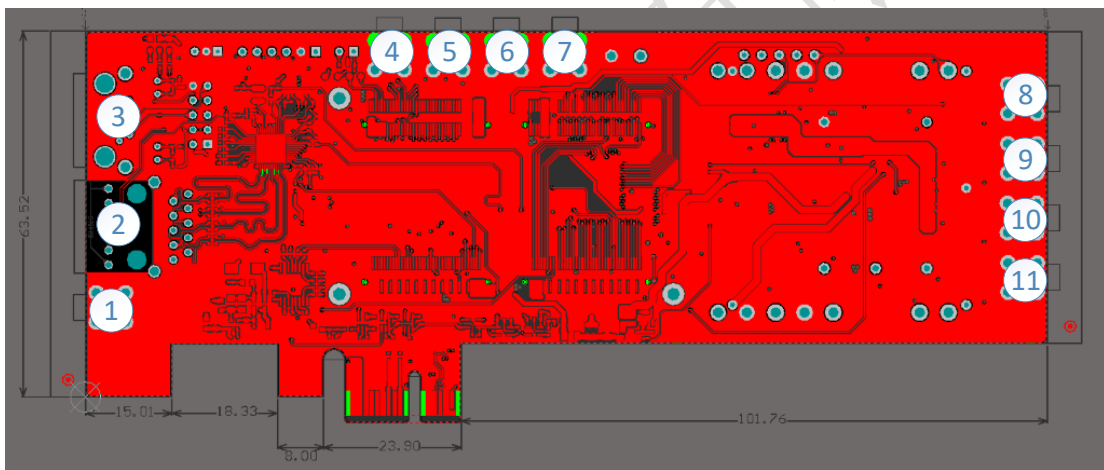


图 3-1-4 结构尺寸图

4.工作原理

STW-PCIE-B 板卡使用高稳晶振(可选配铷钟)作为时钟源, 具备接收 GPS/北斗, B(DC), 1588 PTP, NTP, CDMA, 1PPS+TOD 等时间信息, 然后通过 PCIE 总线输出板卡状态以及时间信息到操作系统, 为操作系统提供可靠稳定的时间信息。

5.使用与操作说明

5.1 安装与测试

5.1.1 linux 驱动安装

1. 驱动安装测试。

打开 terminal，进入 DRIVER 目录。

输入 make 编译驱动，若无法编译，一般是缺失编译驱动环境，根据错误提示进行安装。

输入 `sudo insmod tx_pciDrv.ko` 安装驱动。

输入 `lsmod |grep tx_pciDrv` 查看驱动是否安装成功。

进入 DEMO 目录，

输入 `sh build.sh` 编译测试程序

输入 `sudo ./demo` 运行测试程序，运行成功后可输入 help 查看测试程序支持命令。

2. 开机自启动安装。

打开 terminal，进入 DRIVER 目录，

输入 `sudo sh installcmd.sh` 进行驱动安装，安装成功后，重启电脑。

打开 terminal，输入 `lsmod |grep tx_pciDrv` 查看驱动是否自动启动。

注意：

1. 由于 linux 系统版本较多，本开机自启动方法不保证能正常工作，若不能正常使用，可在操作系统自启动脚本中增加“`insmod tx_pciDrv.ko`”命令实现开机自动加载驱动。

2. 若需要普通用户权限访问驱动，请使用管理员权限输入 `chmod 666 /dev/txpci`，若需要开机自动输入，需要把此脚本放到操作系统的开机自启动文件里。

3. 自动校正操作系统时间。

进入 DEMO 目录，输入 `sudo ./demo tmS 30` 可自动校正操作系统时间，校正周期为 30 秒。若需要开机自启动，可写脚本运行此程序即可。

5.1.2 windows 驱动安装

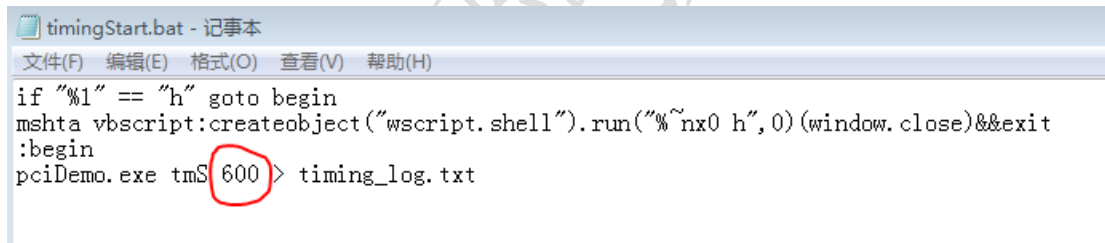
1. 驱动安装

解压 Driver installation.zip 文件，然后直接点击 CH367DRV.EXE 文件进行驱动安装。

2. 可使用 DEMO/Release 中的测试程序 pciDemo.exe 进行测试。

3. 自动校正操作系统时间。

运行 DEMO/Release/timingStart.bat 脚本，可进行后台自动校正操作系统时间。若需开自启动可创建 timingStart.bat 快捷方式，然后快捷方式复制到“开始”/“启动”文件夹里，然后重启即可开机自动启动校时，第一次启动校时程序需要时间源有效才会进入校时流程。若需要修改校时频率，可用记事本代开 timingStart.bat 文件，然后进行如下位置修改。



```
timingStart.bat - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
if "%1" == "h" goto begin
mshta vbscript:createobject("wscript.shell").run("%~nx0 h",0)(window.close)&&exit
:begin
pciDemo.exe tmS 600 > timing_log.txt
```

如上图所示，600 表示 600 秒校正一次操作系统时间。

5.1.3 QNX 驱动安装

1. 驱动安装测试。

QNX 驱动以 .so 库的形式存在，把库放在程序同目录，可进行动态库直接调用使用，编译时使用 libtx_pci_drvS.a 库文件。

注意:QNX 操作系统驱动程序为定制化开发，需在需求中明确才会开发。

5.2 软件开发说明

此驱动使用 C 语言开发，不同平台开发方式不同，为方便客户二次开发，确保不同平台的接口参数一致。

5.2.1 linux 系统开发说明

1). 打开驱动

安装板卡驱动后，会在 dev 目录下生成 tx_pci 文件，用户程序在初始化时需要先 open 该文件。打开方法如下所示：

```
3 int fd;
4 fd = open("/dev/tx_pci", O_RDWR);
5 if(fd < 0)
6 {
7     printf("open tx_pci failed\n");
8 }
9
```

程序运行过程中，请不要重复打开。

2). 关闭驱动

用户程序在运行过程中不需要关闭该文件，待用户程序退出时再进行关闭。关闭方法如下所示：

```
2
3 close(fd);
4
```

3). 读写驱动

open 之后，用户程序使用句柄 fd 通过 ioctl 来访问驱动，获取或修改板卡的时间，状态等信息。读写驱动时使用 ioctl 来实现，其函数原型为 int ioctl(int fd, unsigned long cmd, int arg...); 根据不同的 cmd 来进行不同参数的读写。可参考 tx_pcidemo.c 测试程序。

其 cmd 类型宏定义以及 arg 类型如下所示：

序号	cmd 宏定义	arg 参数说明	功能含义
1	#define TX_PCI_R_VER _IOR('Z', 1, long)	TX_PCI_VER 结构体	获取板卡版本
2	#define TX_PCI_RESET _IOW('Z', 2, int)	int	复位板卡
3	#define TX_PCI_R_TM _IOR('Z', 3, long)	TX_PCI_TIME 结构体	获取当前时间
4	#define TX_PCI_W_TM _IOW('Z', 4, long)	TX_PCI_TIME 结构体	设置当前时间
5	#define TX_PCI_R_CARD_INFO _IOR('Z', 5, long)	TX_PCI_INFO 结构体	获取板卡信息
6	#define TX_PCI_W_SRTM _IOW('Z', 6, int)	int 类型	设置板卡时间源
7	#define TX_PCI_R_SATE_INFO _IOR('Z', 7, char)	SATE_INFO 结构体	获取板卡卫星信息
8	#define TX_PCI_R_NET_INFO _IOR('Z', 8, long)	NET_PARAM 结构体	获取网络信息

9	#define TX_PCI_W_NET_INFO _IOW('Z', 9, long)	NET_PARAM 结构体	设置网 络信息
10	#define TX_PCI_R_PTP_INFO _IOR('Z', 10, long)	PTP_PARAM 结构体	获 取 PTP 参 数
11	#define TX_PCI_W_PTP_INFO _IOW('Z', 11, long)	PTP_PARAM 结构体	设 置 PTP 参 数
12	#define TX_PCI_W_INT_FREQ _IOW('Z', 12, int)	int	PCI 频 率设置
13	#define TX_PCI_R_INT_FREQ _IOR('Z', 13, int)	int	PCI 频 率获取
14	#define TX_PCI_INT_ENABLE _IOR('Z', 14, int)	int	PCI 中 断使能
15	#define TX_PCI_W_SRTMMODE _IOR('Z', 15, int)	int	时间源 切换方 式

注意：各个 arg 参数结构体内容定义请参考 5.2.4 章节<软件开发参数说明>。

传入的 arg 参数应该是相应变量的地址。

无接收机版本不支持上述接收机相关参数读取与设置。

调用代码例程如下：

读取板卡时间所示：

```
5 typedef struct
6 {
7     int num; /*获取第几路时间信息, 最大是4路*/
8     int year;
9     int month;
10    int day;
11    int hour;
12    int minute;
13    int second;
14    int ns;
15 }TX_PCI_TIME;
16
17 TX_PCI_TIME cardTime;
18 cardTime.num = 1;
19 if(ioctl(fd, TX_PCI_R_TM, &cardTime))
20 {
21     printf("ioctl TX_PCI_R_TM failed\n");
22     return -1;
23 }
24
```

设置板卡时间如下所示:

```
27 TX_PCI_TIME cardTime;
28 cardTime.year = 2020;
29 cardTime.month = 01;
30 cardTime.day = 01;
31 cardTime.hour = 18;
32 cardTime.minute = 02;
33 cardTime.second = 01;
34 if(ioctl(fd, TX_PCI_W_TM, &cardTime))
35 {
36     printf("ioctl TX_PCI_W_TM failed\n");
37     return -1;
38 }
39
```

4). 驱动中断

板卡支持PCIE中断功能,用户程序使用read函数阻塞的方式来实现中断响应,当中断到来时,read函数立即返回,并返回中断计数。代码例程如下:

```

39 unsigned int interruptCount;
40 while (1)
41 {
42     if (read(fd, &interruptCount, sizeof(unsigned int)) == 1)
43     {
44         if (TX_PCIIntCallback)
45         {
46             (*TX_PCIIntCallback) ();
47         }
48     }
49 }
50

```

注意：RTLINUX 系暂不支持中断功能。

5.2.2 windows 系统开发说明

windows 驱动使用 dll 动态库的形式实现用户程序与驱动程序的访问。可通过隐式链接或显式链接来实现 dll 库的调用。Dll 库中提供对应函数给用户程序直接调用。

用户安装完驱动后，直接拷贝 txpci.lib.dll，txpci.lib.lib 库文件到用户开发工程。

1). 打开驱动

接口函数：int tx_pci_open(int num)

返回值：int 类型句柄，num 固定填 0。

用户程序在初始化时需要先 open 该文件。打开方法如下所示：

```

3 {
4     {
5         int fd=0;
6         fd = tx_pci_open(0);
7         if(fd<0)
8         {
9             printf("open tx_pci failed\n");
10            getchar();
11            return 0;
12        }
13    }
14 }

```

程序运行过程中，请不要重复打开。

2). 关闭驱动

接口函数: void tx_pci_exit(int fd, int num)

参数: fd 为 tx_pci_open 返回时句柄, num 固定填 0。

用户程序在运行过程中不需要关闭该文件, 待用户程序退出时再进行关闭。关闭方法如下所示:

```
tx_pci_exit(fd, 0);
```

3). 读写驱动

open 之后, 用户程序使用句柄 fd 通过调用函数访问驱动, 获取或修改板卡的时间, 状态等信息。读写驱动时使用调用函数来实现, 根据不同的函数调用来进行不同参数的读写。其函数声明如下所示:

序号	接口函数	函数声明 第二个参数说明	功能含义
1	int tx_pci_r_ver(int fd, TX_PCI_VER *ver)	TX_PCI_VER 结构体	获取板卡版本
2	void tx_pci_reset(int fd, int flag)	int	复位板卡
3	int tx_pci_r_tm(int fd, TX_PCI_TIME *cardTime)	TX_PCI_TIME 结构体	获取当前时间
4	int tx_pci_w_tm(int fd, TX_PCI_TIME *cardTime)	TX_PCI_TIME 结构体	设置当前时间

		体	
5	int tx_pci_r_card_info(int fd, TX_PCI_INFO *info)	TX_PCI_INFO 结构体	获取板卡信息
6	int tx_pci_w_srTm(int fd, int sr)	int 类型	设置板卡时间源
7	int tx_pci_r_sate_info(int fd, SATE_INFO *info)	SATE_INFO 结构体	获取板卡卫星信息
8	int tx_pci_r_net_info(int fd, NET_PARAM *info)	NET_PARAM 结构体	获取网络信息
9	int tx_pci_w_net_info(int fd, NET_PARAM *info)	NET_PARAM 结构体	设置网络信息
10	int tx_pci_r_ptp_info(int fd, PTP_PARAM *info)	PTP_PARAM 结构体	获取 PTP 参数
11	int tx_pci_w_ptp_info(int fd, PTP_PARAM *info)	PTP_PARAM 结构体	设置 PTP 参数
12	int tx_pci_w_int_freq(int fd, int freq)	int	PCI 频

			率设置
13	int tx_pci_r_int_freq(int fd, int *freq)	int	PCI 频率获取
14	int tx_pci_int_enable(int fd, int flag)	int	PCI 中断使能
15	int tx_pci_w_srTmMode(int fd, int mode)	int	时间源切换方式

注意：函数返回值 0：表示成功，-1 表示失败。

函数声明中第一个参数是 open 时返回的句柄，第二个参数内容定义请参考 5.2.4 章节<软件开发参数说明>。

无接收机版本不支持上述接收机相关参数读取与设置。

代码如下读取板卡时间所示：

```

77 typedef struct
78 {
79     int num;
80     int year;
81     int month;
82     int day;
83     int hour;
84     int minute;
85     int second;
86     int ns;
87 }TX_PCI_TIME;
88
89 TX_PCI_TIME cardTime;
90 cardTime.num = 0;
91 if(tx_pci_r_tm(fd, &cardTime))
92 {
93     printf("ioctl tx_pci_r_tm failed\n");
94     return -1;
95 }

```

设置板卡时间如下所示：


```
98 TX_PCI_TIME cardTime;
99 cardTime.year = 2020;
100 cardTime.month = 01;
101 cardTime.day = 01;
102 cardTime.hour = 18;
103 cardTime.minute = 02;
104 cardTime.second = 01;
105 if(tx_pci_w_tm(fd, &cardTime))
106 {
107     printf("ioctl tx_pci_w_tm failed\n");
108     return -1;
109 }
```

4). 驱动中断

板卡支持 PCIE 中断功能，使用注册中断的方式注册中断，中断注册函数为：

```
typedef void (*TX_INT_CALLBACK)(unsigned int flag);
int tx_pci_int_request(TX_INT_CALLBACK fun)
```

代码例程如下：

注册中断回调函数：

```
tx_pci_int_request(interrupt_callback);
tx_enable_int(g_fd, 1);
```

中断回调函数：

```
TX_INT_CALLBACK interrupt_callback(unsigned int flag)
{
    //do something
}
```

5.2.3 QNX 系统开发说明

1). 打开驱动

接口函数：int tx_pci_open(int num)

返回值：int 类型句柄，num 固定填 0。

用户程序在初始化时需要先 open 该文件。打开方法如下所示：

```

3   {
   int fd=0;
   fd = tx_pci_open(0);
   if(fd<0)
3   {
   printf("open tx_pci failed\n");
   getchar();
   return 0;
   }
-   }
-

```

程序运行过程中，请不要重复打开。

2). 关闭驱动

接口函数: void tx_pci_exit (int fd,int num)

用户程序在运行过程中不需要关闭该文件，待用户程序退出时再进行关闭。num 为预留参数，固定填 0。

3). 读写驱动

open 之后，用户程序使用句柄 fd 通过调用函数访问驱动，获取或修改板卡的时间，状态等信息。读写驱动时使用调用函数来实现，根据不同的函数调用来进行不同参数的读写。其函数声明如下所示：

序号	接口函数	函数声明 第二个 参数说明	功能含义
1	int tx_pci_r_ver(int fd, TX_PCI_VER *ver)	TX_PCI_ VER 结构	获取板卡版本

		体	
2	void tx_pci_reset(int fd,int flag)	int	复位板卡
3	int tx_pci_r_tm(int fd,TX_PCI_TIME *cardTime)	TX_PCI_TIME 结构体	获取当前时间
4	int tx_pci_w_tm(int fd,TX_PCI_TIME *cardTime)	TX_PCI_TIME 结构体	设置当前时间
5	int tx_pci_r_card_info(int fd,TX_PCI_INFO *info)	TX_PCI_INFO 结构体	获取板卡信息
6	int tx_pci_w_srTm(int fd,int sr)	int 类型	设置板卡时间源
7	int tx_pci_r_sate_info(int fd, SATE_INFO *info)	SATE_INFO 结构体	获取板卡卫星信息
8	int tx_pci_r_net_info(int fd, NET_PARAM *info)	NET_PARAM 结构体	获取网络信息

9	<code>int tx_pci_w_net_info(int fd, NET_PARAM *info)</code>	NET_PAR AM 结构 体	设置网 络信息
10	<code>int tx_pci_r_ptp_info(int fd, PTP_PARAM *info)</code>	PTP_PAR AM 结构 体	获取 PTP 参数
11	<code>int tx_pci_w_ptp_info(int fd, PTP_PARAM *info)</code>	PTP_PAR AM 结构 体	设置 PTP 参数
12	<code>int tx_pci_w_int_freq(int fd, int freq)</code>	int	PCI 频率 设置
13	<code>int tx_pci_r_int_freq(int fd, int *freq)</code>	int	PCI 频率 获取
14	<code>int tx_pci_enable_int (int fd, int flag)</code>	int	PCI 中断 使能

注意：函数返回值 0；表示成功，-1 表示失败。

函数声明中第一个参数是 open 时返回的句柄，第二个参数内容定义请参考 5.2.4 章节<软件开发参数说明>。

无接收机版本不支持上述接收机相关参数读取与设置。

代码如下读取板卡时间所示：

```
77 typedef struct
78 {
79     int num;
80     int year;
81     int month;
82     int day;
83     int hour;
84     int minute;
85     int second;
86     int ns;
87 }TX_PCI_TIME;
88
89 TX_PCI_TIME cardTime;
90 cardTime.num = 0;
91 if(tx_pci_r_tm(fd,&cardTime))
92 {
93     printf("ioctl tx_pci_r_tm failed\n");
94     return -1;
95 }
```

设置板卡时间如下所示：

```
98 TX_PCI_TIME cardTime;
99 cardTime.year = 2020;
100 cardTime.month = 01;
101 cardTime.day = 01;
102 cardTime.hour = 18;
103 cardTime.minute = 02;
104 cardTime.second = 01;
105 if(tx_pci_w_tm(fd,&cardTime))
106 {
107     printf("ioctl tx_pci_w_tm failed\n");
108     return -1;
109 }
```

4). 驱动中断

驱动在初始化后会自动挂接中断，用户程序可直接使用 InterruptWait 等待中断，实现方式如下如所示：

```

void *interrupt_thread(void)
{
    int count = 0;

    printf("thread success\n");

    setprio(0, 63); /*设置线程优先级*/
    tx_enable_int(0,1);
    while(1)
    {
        InterruptWait(0, NULL); /*等待中断*/
        //do something
    }
    tx_enable_int(0,0);
}

```

注意：如上代码所示，tx_enable_int 跟等待中断应该在一个线程里执行。

5.2.4 软件开发参数说明

5.2.4.1 获取板卡版本(TX_PCI_VER)

```

typedef struct
{
    int hwVer; /*硬件版本*/
    int fpgaVer; /*fpga 版本*/
    int softVer; /*软件版本*/
    int drvVer; /*驱动版本*/
}TX_PCI_VER;

```

说明：获取版本信息，非必须使用此功能。

5.2.4.2 复位板卡(int)

功能说明：复位板卡。一般没有复位需求可不进行复位操作。

此参数为预留参数，可固定传 0；

5.2.4.3 获取/设置当前时间(TX_PCI_TIME)

```
typedef struct
{
    int num; /*固定填 0 */
    int year; /*年*/
    int month; /*月*/
    int day; /*日*/
    int hour; /*时*/
    int minute; /*分*/
    int second; /*秒*/
    int ns; /*当前秒以下信息，单位为纳秒*/
}TX_PCI_TIME;
```

功能说明：获取或设置板卡时间。

num 表示获取第几通道(0-5)时间，一般默认填 0。当多线程频繁获取板卡时间时，不同线程可获取不同通道时间，避免多线程获取时间等待太久。

5.2.4.4 获取板卡状态(TX_PCI_INFO)

```
typedef struct
{
    int cur_srTm; /*当前时间源：1:GPS 2:BEIDOU 3:GNSS 4:PTP 8:1PPS+TOD
                9:1PPS 11:CDMA 12:BDC*/
    int srTmValid; /*当前时间源有效性:bit 表示, 为 1 表示有效。
                bit1:GPS bit2:BEIDOU bit3:GNSS bit4:PTP
                bit8:1PPS+TOD bit9:1PPS bit11:CDMA bit12:BDC*/
    int card_status; /*当前板卡状态:1:正常, 2: 不正常*/
```

```

int sync_status; /*同步状态:1:未同步, 2:同步, 3:守时*/
int syncAccuracy; /*同步精度, 单位纳秒*/
int ext10MStatus; /*外 10M:1:存在, 0: 不存在*/
int srTm_mode; /*时间源切换方式: 1: 自动 2: 手动*/
int res[10]; /*预留*/
}TX_PCI_INFO;

```

功能说明：获取板卡状态信息。

5.2.4.5 设置板卡时间源(int)

功能说明：选择板卡的同步时间源。选择后，板卡会跟被选择的时间源进行时间同步。使用 int 类型。其值如下所示：

1:GPS 2:BEIDOU 3:GNSS 4:PTP 8:1PPS+TOD 9:1PPS 11:CDMA 12:BDC

5.2.4.6 获取卫星信息(SATE_INFO)

```

typedef struct
{
    int gps_sat_num; /*GPS 卫星数*/
    int bd_sat_num; /*北斗卫星数*/
    int gps_signal; /*GPS 最大信噪比, 单位 dB*/
    int bd_signal; /*北斗最大信噪比, 单位 dB*/
    char antennaStatus; /*天线状态: 0: 未连接 1: 连接正常*/
    int latitude_d; /*纬度: 度*/
    char latitude_m; /*纬度: 分*/
    char latitude_s; /*纬度: 秒*/
    char latitude_sign; /*方向: ASSIC 表示, 'N' 或 'S' */
    int longitude_d; /*经度: 度*/
}

```

```
char longitude_m; /*经度：分*/
char longitude_s; /*经度：秒*/
char longitude_sign; /*方向：ASSIC 表示，'E' 或'W' */
int altitude; /*海拔高度，单位米*100*/
}SATE_INFO;
```

功能说明：获取当前卫星信息，需要接天线，搜星后才能获取到卫星信息。

5.2.4.7 获取/设置网络信息(NET_PARAM)

```
typedef struct
{
    int num; /*网口号*/
    int ipaddr; /*ip 地址*/
    int subnet; /*子网掩码*/
    int gateWay; /*网关*/
    int link_speed; /*0:未 link, 1:10M, 2:100M, 3:1000M*/
}NET_PARAM;
```

功能说明：获取网口 IP 地址等信息。当获取或设置时，num 网口号固定填 0，当设置 IP 地址等信息时，link_speed 可以不赋值。

5.2.4.8 获取/设置 PTP 参数(PTP_PARAM)

```
typedef struct
{
    int role; /*1:主, 2: 从*/
    int delay_mode; /*0:E2E, 1:P2P*/
    int domain_no; /*域号，一般固定 0*/
    int msg_mode; /*0:组播, 1: 单播*/
    int uc_addr; /*主机 IP 地址，单播模式可用*/
```

}PTP_PARAM;

功能说明：获取或配置 PTP 运行参数。一般可使用出厂配置，即 PTP 从模式，E2E，域号为 0，组播模式。

5.2.4.9 PCI 频率设置或查询(int)

功能说明：设置或查询中断频率，如设置为 1hz 时表示 1 秒产生一个 PCIE 中断。

int 类型。单位为 hz，范围是 1-1000

5.2.4.10 PCI 中断使能(int)

功能说明：使能 PCIE 中断。

int 类型。1：开中断，0：关闭中断。

5.2.4.11 时间源切换方式设置(int)

功能说明：设置时间源切换方式。

int 类型：1：自动 2：手动

自动模式下，会优先使用设置的时间源，当该时间源无效后，会自动切换别的时间源(由于 CDMA 精度比较差，而且长期有效，为防止自动切换到 CDMA 造成把时间拉偏，所以当所设置时间源无效后，不会自动切换 CDMA)。

手动模式下，只会使用设置的时间源，当设置的时间源无效后，会进入守时模式。

5.3 板卡硬件接口说明

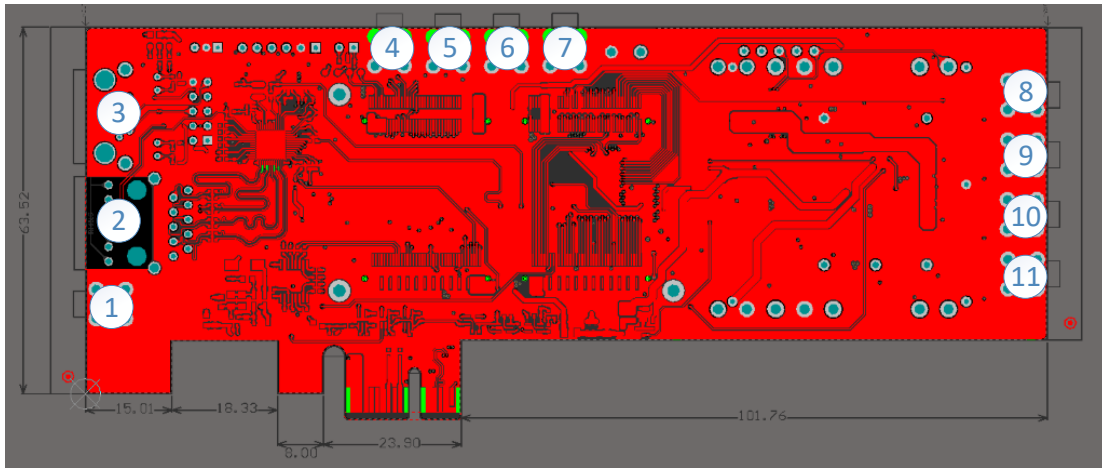


图 5-3-1 板卡接口图

序号	信号名称	接插件	信号参数	备注
1	卫星天线	SMA	---	GPS/北斗
2	1000M/100M 网口	RJ45	网络信号	NTP、1588V2 PTP 信号
3	其它信号	RJ45	----	1PPS (TTL) +TOD (232) 输入 (定义见下图)
4	BDC 输入信号	mcx	TTL	第二路 BDC 输入 (TTL)
5	PPS 输出信号	mcx	TTL	PPS 输出信号
6	CDMA 天线	mcx	----	----
7	10M 输入	mcx	----	----
8	10M 输出	mcx	TTL	10M 输出
9	BDC 输出	mcx	TTL	BDC 输出
10	BDC 输出	mcx	TTL	BDC 输出
11	BDC 输出	mcx	TTL	BDC 输出

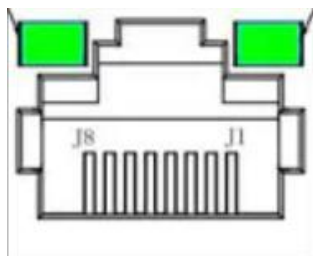


图 5-3-2 其它信号定义

J1:TOD_232 输出 (232)

J2:1PPS 输出 (TTL)

J3:1PPS 输入 (422 正)

J4:BDC 输入 (TTL)

J5:GND

J6:1PPS 输入 (422 负)

J7:预留

J8:预留

左灯：秒灯，每秒闪烁。

右灯：状态灯，时间源有效后常亮。

5.4.板卡测试方法

5.4.1 驱动 demo 使用方式

本驱动软件可使用 demo 软件辅助进行板卡测试。demo 软件主要用于测试板卡使用，该软件提供了源码，以便帮助用户能快速进行二次开发。

不同平台 demo 软件打开方式不一样，打开方式请查看《3. 安装与测试》。demo 软件打开后，可输入 help 查看支持的测试命令。如下图所示：

```
please input cmd,like help
input:help
input:help
usage:
  l: show all info
  int: open interrupt
  intc:close interrupt
  tms [year] [month] [day] [hour] [minute] [second]: set card time
  ips [ip] [snm] [gw]: set card ip(0xc0a80a01)
  ptps [1:master 2:slave] [0:E2E 1:P2P] [domain] [0:multicast;1:Unicast] [Unicast Addr]: set master ip
  srtm [1:GPS 2:BEIDOU 3:GNSS 4:PTP 8:1PPS+TOD 9:1PPS 11:CDMA 12:BDC]: set time source
  srtm_m [1:AUTO 2:MANUAL]: set srtm change mode
  freqs [1-1000hz]: set interrupt frequency
  timings: adjust system time once
  reset: reset card
  exit :exit program
please input cmd,like help
input:
```

图 5-4-1 demo 软件帮助界面

如上图所示，命令分为两类，一类是不带参数的命令，如“l”，“int”等，另外一类是带参数的命令，如“tms”，“ips”等。

“l”：显示当前板卡信息。

“int”：打开 PCIE 中断。

“intc”：关闭 PCIE 中断。

“tms”：设置板卡时间。

“ips”：设置板卡网口 IP 地址，如输入 ips 0xc0a80a02 0xffffffff00 0xc0a80a01，表示设置板卡 IP 地址为 192.168.10.2，网关设置为 192.168.10.1。

“ptps”：表示设置 ptp 参数，master 表示设置 ptp 为主模式，slave 表示 ptp 从模式；E2E 表示 E2E 模式，P2P 表示 P2P 模式，ptp 主从之间需要此模式保持一致，一般默认 E2E。domain 表示时间域号，主从之间需要保证时间域号一致，一般默认 0。multicast 表示 PTP 组播模式，Unicast 表示 PTP 单播模式。Unicast Addr 表示单播地址，即当 PTP 设置成单播从模式的时候，需要设置单播地址，单播地址即为 PTP 主的 IP 地址。一般板卡出厂时 PTP 会设置为 E2E，组播，从模式。这是通用设置，一般可直接做从使用。

如 ptps 2 0 0 1 0xc0a80a02 表示设置 PTP 从，E2E 模式，时间域号 0，单播，单播地址为 192.168.10.2。

“srtm”：表示设置时间源。如输入：srtm 1 表示设置时间源为 GPS。

“srtm_m”：表示设置时间源切换自动，手动模式。AUTO 表示自动切换，当所选时间源无效后，会自动切换别的时间源，manual 表示手动切换，当所选时间源无效后，不会自动切换成别的时间源。由于 CDMA 模式一般都会有效，但 CDMA 精度较低，毫秒级别，为防止把秒拉偏，所以自动模式下不会自动切换到 CDMA，如需要同步 CDMA 时间，需要设置时间源为 CDMA 模式。

“freqs”：表示设置 PCIE 中断频率。如输入 freqs 10。表示设置 PCIE 中断频率为 10Hz。

“timings”：表示校正操作系统时间一次。

“reset”：表示复位板卡，复位板卡后需要等待一段时间才能使用。

输入“1”即可查看当前板卡的所有信息，如下图所示：

```
input:1
*****VER*****
hwver:1
fpgaver:20220125
softver:20220125
drvrver:20211115
*****INFO*****
srTm:1 [1:GPS 2:BEIDOU 3:GNSS 4:PTP 8:1PPS+TOD 9:1PPS 11:CDMA 12:BDC]
srTmValid:GPS<Y> BEIDOU<Y> GNSS<Y> PTP<Y> 1PPSTOD<N> 1PPS<N> CDMA<Y> BDC<N>
cardStatus:NORMAL
syncStatus:SYNC
10M:none
srTmMode:AUTO
syncAccuracy:8
*****SATE*****
gps:10,beidou:13
gps snr:44,beidou snr:44
latitude:30'44'24 N
longitude:103'57'45 E
altitude:546.999988
*****NET*****
ipaddr:192.168.1.10
subnet:255.255.255.0
gateway:192.168.1.1
link_speed:1000M
*****PTP*****
role:slave[2]
delay:E2E[0]
domain:0
msg:multicast[0]
Unicast Addr:0.0.0.0
*****FREQ*****
int:1hz
card time[0]:2022-01-28 09:55:33 947 004 470
please input cmd,like help
input:
```

图 5-4-2 demo 软件显示板卡信息

“VER”显示板卡内部调试信息，

“INFO”中 srTm 表示当前使用的时间源，srTmValid 显示时间源有效状态，Y 表示有效，N 表示无效。cardStatus 表示板卡状态，NORMAL 表示正常。syncStatus 表示板卡同步状态，同步后表示当前时间可用。10M 表示外 10M 输入状态，当有外 10M 输入后，板卡会自动切换为外 10M，并显示 exist。srTmMode 表示时间源切换方式。“syncAccuracy”表示板卡与当前时间源的同步精度，即与当前时间源的秒相位偏差，单位为纳秒。

“SATE”表示卫星定位信息。包括 gps，北斗可用卫星数，最大信噪比，以及经纬度信息。当卫星数较少，信噪比较低时表示当前信号不好，需要查看天线是否没有接好或者天线有遮挡。

“NET”表示板卡网口的IP地址，子网掩码，网关，当前连接状态。

“PTP”表示PTP可配置参数。role表示PTP主从模式，delay表示PTP延时机制，包括E2E, P2P模式。msg表示PTP组播或单播模式。Unicast表示PTP单播地址。

“FREQ”表示PCIE中断频率。

“card time”表示当前板卡时间。分为年月日，时分秒，毫米，微妙，纳秒。

带参数的输入方式如下所示：

```
please input cmd,like help
input:tms 2022 01 01 01 02 30
input:tms 2022 01 01 01 02 30
set time:2022-01-01 01:02:30
please input cmd,like help
input:
```

图 5-4-3 demo 软件设置板卡时间

如上图所示，设置时间输入 tms 2022 01 02 03 04 05。表示设置时间为 2022 年 1 月 2 日 3 时 4 分 5 秒。

5.4.2 板卡测试环境搭建

板卡测试环境可如下如所示搭建。板卡插在电脑 PCIE 插槽上，使用时频综测仪对板卡进行授时以及测试。

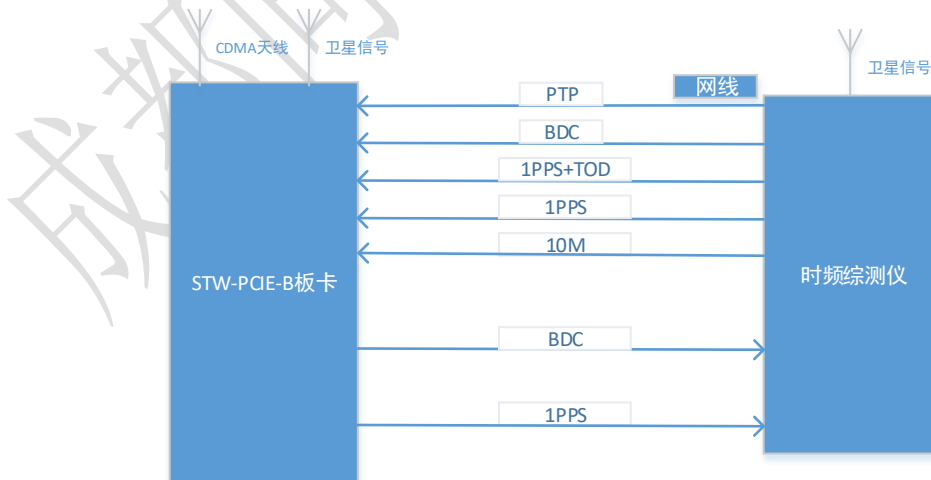


图 5-4-4 板卡测试环境框图

1) .测量时间源输入

如上图所示分别给板卡输入 CDMA 信号(使用天线), 卫星信号(使用天线), PTP(信号来自时频综测仪), BDC(信号来自时频综测仪), 1PPS+TOD(信号来自时频综测仪), 1PPS (信号来自时频综测仪), 10M (信号来自时频综测仪)。当测量某种信号输入时, 可用驱动 demo 程序切换到该时间源, 同时查看板卡与该时间源的有效以及同步状态。板卡同步之后输出 1PPS 到时频综测仪器进行测试。

时频综测仪器通过卫星给自身授时, 通过测量板卡输出的 1PPS 信号来测量统计板卡的同步精度。也可以使用标准时间设备, 然后使用示波器来测量标准时间设备输出的秒与板卡输出的秒的相差。

配置需求: PTP 出厂时会设置为 E2E, 组播, 从模式。这是通用设置, 一般可直接做从使用。若无法正常授时请检查网络是否连通, PTP 主设备的模式是否与从一致, 不一致则需要进行修改。

1PPS+TOD 时间源 TOD 使用 NMEA 协议。

BDC 按照 GJB2991A-2008 标准协议。

其它时间源输入不需要针对时间源进行特殊配置。

2) .测量板卡输出

STW-PCIE-B 板卡与时频综测仪分别接天线授时, 授时成功后, 使用板卡输出 BDC 到时频综测仪进行测试, 然后查看时频综测仪的测试结果, 查看输出板卡输出 BDC 输出的时间是否正确。

5.5 板卡协议

5.5.1 TOD 协议说明

TOD 使用波特率 115200, 使用 NMEA 协议中的 ZDA 格式, 格式内容如下所示:

```
$BDZDA,hhmmss.ss,day,month,year,,*cs<CR><LF>
```

Example:

```
$GPZDA,082710.00,16,09,2002,00,00*64
```

hhmmss.ss:UTC 时间, 时分秒。

day: UTC 天

month: UTC 月

year: UTC 年

cs: \$之后, *之前的 16 进制异或校验。

6. 常见故障排除及维修

故障现象	故障原因	排查方法
系统无法识别板卡	1. 板卡未插好 2. 金手指氧化	1. 重新插拔板卡, 确定插对是 PCIE 槽位, 并确定槽位没问题 2. 此问题一般出现在长久使用后的板卡, 可用酒精或者橡皮擦拭金手指, 把金手指氧化层擦掉

7. 运输和贮存

产品采用专用包装箱运输, 箱内有防振软泡沫衬垫, 确保产品在运输过程中不被损坏, 产品在运输时应水平摆放, 严禁淋雨。

贮存产品的位置应具有良好的通风环境, 并保持环境干燥, 室内无酸碱及腐蚀性气体, 无强力的振动或撞击。

8. 产品齐套

序号	名称	规格型号	数量	备注
1	STW-PCIE-B 板卡		1 张	
2	STW-PCIE-B 板卡 配套资料光盘		1 张	
3	测试报告		1 份	

4	合格证		1份	
---	-----	--	----	--

成都同相科技有限公司