

SG-PCIE-ECAT二次开发文档

SG-PCIE-ECAT接口说明

版本记录

时间	版本	说明
2026.6.29	1.0	添加EtherCAT从站功能

功能简介

SG-PCIE-PN-200S 是一款高性能的PCIe接口卡，使PC机通过PCIe接口连接至EtherCAT网络，板卡支持EtherCAT从站。支持Windows、Linux下进行二次开发，以动态连接库的方式驱动。

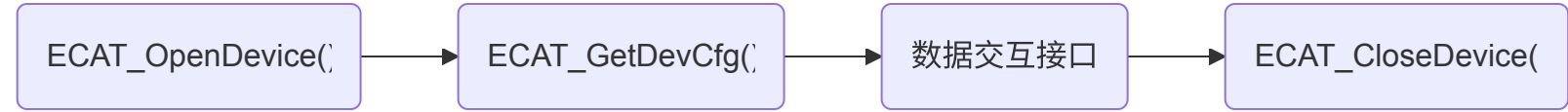
适用平台

平台	架构	编译器	支持状态
Linux	x86_64	GCC 9+	待支持
Windows	x86_64	MSVC 2019+	支持

Windows 端

开发编程时，直接加载 **PN-DEV.dll** 即可，接口描述文件位于 **PNDEV.h** ,dll文件放置在可执行文件同级目录下即可，根据自身实际使用场景，选择对应章节开发说明。

库函数调用流程



通用部分

板卡支持PROFINET主、从站，部分数据结构跟库接口是通用的，数据交互接口是有差异的，可根据自身使用场景，查看对应章节。

数据结构说明

ECAT_CFG_T

```
typedef struct {
    uint8_t Cmd;
    uint8_t Status;
    uint8_t nc[2];
    uint16_t InSize;
    uint16_t OutSize;
    uint8_t app_Version[8];
    uint8_t fw_Version[8];
} ECAT_CFG_T, * PECAT_CFG_T;
```

C

Cmd: 0 - 获取库版本号；1 - 获取板卡固件版本号；

Status: 1 - OP

nc: 无意义

app_Version: 库版本号

fw_Version: 固件版本号

库接口说明

打开设备

```
int ECAT_OpenDevice(int Index);
```

说明: 用于打开设备，连接板卡
Index: 设备索引号，有一个设备时索引号为0，有两个可以是0或1
返回值: 0为成功

关闭设备

```
BOOL ECAT_CloseDevice(int Index);
```

说明: 用于关闭 `device_open()` 打开的资源
Index: 设备索引号
返回值: 0为成功

获取参数信息

```
int ECAT_GetDevCfg(int Index, PECAT_CFG_T p);
```

说明: 用于获取板卡IP等网络参数
Index: 设备索引号
p: 结构体指针
返回值: 0为成功

读输入缓冲区数据

```
int ECAT_GetInputBuff(int Index, int Addr, int Size, uint8_t *Data);
```

说明: 用于设置板卡IP等网络参数
Index: 设备索引号
Addr: 数据地址
Size: 数据大小
Data: 存放数据指针
返回值: 0为成功

写输出缓冲区数据

```
int ECAT_SetOutputBuff(int Index, int Addr, int Size, uint8_t *Data);
```

说明: 用于设置板卡IP等网络参数
Index: 设备索引号
Addr: 数据地址
Size: 数据大小
Data: 存放数据指针
返回值: 0为成功

接口库使用说明

将库函数文件都放置在工作目录下即可，库函数文件有三个：`ECATDEV.h`、`ECATDEV.lib`、`ECATDEV.dll`。

VC调用动态库方法

- 在扩展名为.cpp的文件中包含PNDEV.h头文件。如：`#include "ECATDEV.h"`
- 在工程的链接器设置中链接到 `ECATDEV.lib` 文件。如：在VS环境下，在项目属性页里的配置属性-->连接器-->输入-->附加依赖项 中 添加 `ECATDEV.lib`。

Linux 端

本目录包含库和示例程序，库函数调用流程及接口库参考Windows端即可。

驱动编译

```
cd Linux/Driver
make
```

bash

安装驱动

检查是否存在 `ch36x.ko` 的驱动程序，驱动编译的路径不能包含中文，否则会导致编译失败。

```
sudo insmod ch36x.ko
```

bash

卸载驱动

由于测试原因，需卸载驱动，执行以下命令卸载。

```
sudo rmmod ch36x.ko
```

bash

永久安装&卸载

永久安装后可避免每次开机手动执行 `sudo insmod ch36x.ko` 加载驱动。在Driver目录下执行以下命令。

```
sudo make install      #安装
sudo make uninstall    #卸载
```

bash

测试程序编译

```
cd Linux/Demo/example
mkdir build && cd build
cmake ..
make
```

bash

编译完成后会在build目录下生成 `test_demo` 的测试程序。

运行测试程序

```
./test_demo -d 0
```

bash

注：测试程序需root用户运行，或者给相应的设备节点读写权限，例如：`sudo chmod 777 /dev/ch36xpci0`