

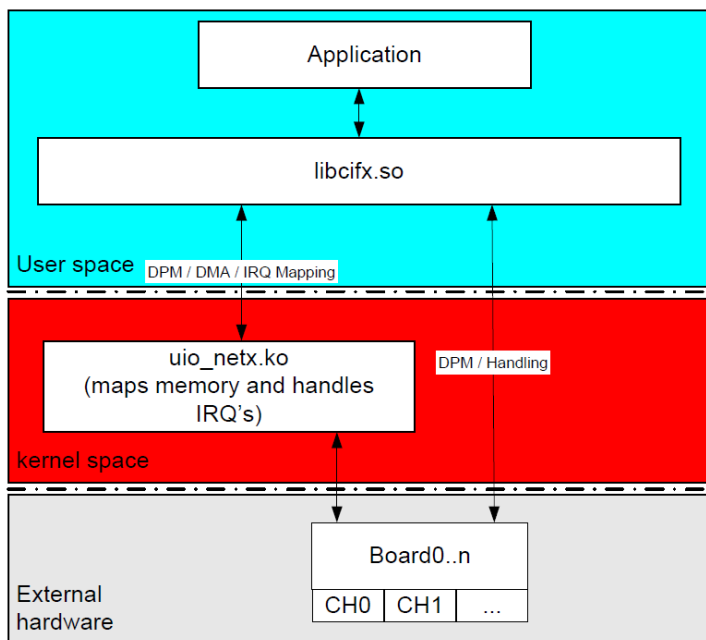
HK-CIFX 操作指南

HK-CIFX PC 板卡 Linux 环境下驱动安装及通讯配置操作指南



1.应用简介

本文档的用意在于让初次接触HK/CIFX PC板卡的使用者了解该板卡在Linux系统中驱动的安装，配置，调试，以及二次开发包的使用。通过该文档的引导，使用者可以让HK/CIFX PC板卡在Linux环境下正常运行起来，并与其它设备或PLC进行基本的通信测试。Linux下，CIFX Linux驱动作为一个库在用户空间运行，并通过IO内核模块访问该板卡(用户空间I/O)。HK/CIFX PC板卡可作为工业实时以太网或现场总线协议的主站或从站，如Profinet主从站，Ethernet/IP主从站，EtherCAT主从站，详细的介绍请查看板卡的简介资料与说明手册。



2. Linux 驱动安装

2.1 环境要求

Mandatory

- Linux Kernel Source (for PCI cards or other devices which rely on uio_netx)
- libpthread, librt
- CMake (min 2.8.9)

Optional

- Linux standard libraries *libpciaccess* (tested with V0.10.2 / V0.13.1-2)
- always needed for cifX PCI cards, support can be disabled by defining *CIFX_TOOLKIT_DISABLEPCI*
- *Optional: pkg-config* utility for automatic finding/configuring needed libraries

2.2 Linux 组件更新

安装驱动之前，根据 HK/CIFX PC板卡在Linux系统下运行环境的基本要求，因此建议先更新和安装必要的功能组件，主要是安装下libpciaccess（PCI板卡）标准库以及pkg-config工具，在Linux系统联网情况下，可通过sudo apt-get 命令更新。

```
root@eric-Default-string:/home/eric# sudo apt-get update
命中:1 http://mirrors.aliyun.com/ubuntu bionic InRelease
获取:2 http://mirrors.aliyun.com/ubuntu bionic-security InRelease [88.7 kB]
```

```
root@eric-Default-string:/home/eric# sudo apt-get install libpciaccess-dev
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
将会同时安装下列软件：
  libpciaccess0
下列软件包将被升级：
  libpciaccess-dev libpciaccess0
```

```
root@eric-Default-string:/home/project/driver# sudo apt-get install pkg-config
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列【新】软件包将被安装：
  pkg-config
```

2.3 解压驱动包

复制驱动压缩包到Linux操作系统环境中，通过tar -zxvf命令解压CIFX PC板卡Linux驱动压缩包，便于演示，这里重新命名为project，注意解压路径中不要包含空格。解压后文件中包含/driver和/examples文件中，分别包含CIFX PC板卡驱动以及板卡程序示例。

```
root@ubuntu:/home# ls
eric V1.1.4.0.tar.gz
root@ubuntu:/home# tar -zxvf V1.1.4.0.tar.gz
V1.1.4.0/
V1.1.4.0/driver/
```

```
root@ubuntu:/home# mv V1.1.4.0 project
root@ubuntu:/home# ls
eric project V1.1.4.0.tar.gz
```

```
root@ubuntu:/home# ls
eric project V1.1.4.0.tar.gz
root@ubuntu:/home# cd project/
root@ubuntu:/home/project# ls
driver examples
root@ubuntu:/home/project# cd driver/
root@ubuntu:/home/project/driver# ls
BSL driver_history.txt plugins scripts uio_netx
build_install_driver libcifx readme templates
root@ubuntu:/home/project/driver# cd ../examples/
root@ubuntu:/home/project/examples# ls
basedir cifXTCPServer CMakeLists.txt LoadModules
cifxsample cifXTestConsole ISASample SPISample
root@ubuntu:/home/project/examples#
```

2.4 安装驱动

(1、进入驱动文件夹/home/project/driver，执行驱动安装脚本build_install_driver，根据安装过程中出现的提示选择必要的操作，驱动正确安装打印消息最后会提示“cifx driver successfully installed”。如果驱动安装过程中出现报错“configure:error:libpciaccess not found via pkg-config,or wrong version!”，那么参考Linux组件更新一节说明，安装必要的组件。

```
root@eric-Default-string:/home# ls
eric lost+found project V1.1.4.0.tar.gz
root@eric-Default-string:/home# cd project/
root@eric-Default-string:/home/project# ls
driver examples
root@eric-Default-string:/home/project# cd driver/
root@eric-Default-string:/home/project/driver# ls
BSL driver_history.txt plugins scripts uio_netx
build_install_driver libcifx readme templates
root@eric-Default-string:/home/project/driver# ./build_install_driver
*****
**** Start building kernel module uio_netx ****
*****
```

```
make[2]: 离开目录“/home/project/driver/libcifx”
make[1]: 离开目录“/home/project/driver/libcifx”
cifx driver successfully installed!
root@eric-Default-string:/home/project/driver#
```

(2、驱动正确安装后，可以查看到通过驱动安装脚本创建的/opt/cifx文件中，其中包括板卡 boot loader文件， device.conf配置文件以及设备识别所需的文件树结构（以下示例中为：Device identification via single directory方式文件树结构）。另外在用户本地库目录中可以看到已安装的libcifx相关的库文件，头文件在/usr/local/include/cifx中。

```

root@ubuntu:/opt# tree
├── cifx
│   ├── bsl_history.txt
│   ├── deviceconfig
│   │   └── FW
│   │       ├── channel0
│   │       └── device.conf
│   ├── NETX100-BSL.bin
│   ├── NETX10-BSL.bin
│   ├── NETX50-BSL.bin
│   ├── NETX51-BSL.bin
│   └── NETX52-BSL.bin

```

```

root@ubuntu:/usr/local/lib# ls
libcifx.a      libcifxpre.so.0      pkgconfig
libcifx.la    libcifxpre.so.0.0.0 python2.7
libcifxpre.a  libcifx.so           python3.4
libcifxpre.la libcifx.so.1
libcifxpre.so libcifx.so.1.1.4
root@ubuntu:/usr/local/lib# cd ../include/cifx/
root@ubuntu:/usr/local/include/cifx# ls
cifXEndianess.h  cifxlinux.h  netx_tap.h  rcX_User.h
cifXErrors.h    cifxUser.h   rcX_Public.h  TLR_Types.h

```

(3、通常情况下驱动安装后uio_netx需要手动再安装 (modprobe uio_netx) ，为避免反复操作，建议将uio_netx添加到Linux模块自启动内，可以直接在/etc目录下modules文件中添加uio_netx。以及例程编译后执行时时可能出现无法链接到libcifx库的情况，可通过ldconfig在库目录下注册库并创建链接。

以上操作完成。重启Linux系统可使得修改的配置生效。驱动正确安装与配置后，CIFX PC板卡可在Linux下使用。

```

root@ubuntu:/home/project/examples/cifXTCPserver# ./cifXTCPserver
./cifXTCPserver: error while loading shared libraries: libcifx.so.1: cannot open
shared object file: No such file or directory
root@ubuntu:/home/project/examples/cifXTCPserver# cd /usr/local/lib/
root@ubuntu:/usr/local/lib# ls
libcifx.a      libcifxpre.la    libcifxpre.so.0.0.0  libcifx.so.1.1.4  python3.4
libcifx.la    libcifxpre.so    libcifx.so           pkgconfig
libcifxpre.a  libcifxpre.so.0  libcifx.so.1        python2.7
root@ubuntu:/usr/local/lib# ldconfig
root@ubuntu:/usr/local/lib# cd /home/project/examples/cifXTCPserver/
root@ubuntu:/home/project/examples/cifXTCPserver# ./cifXTCPserver
cifXDriverInit...
Press ctrl+'c' to quit!

*****
The following card(s) are now accessible via TCP/IP remote connection:

The server is reachable under:
Interface Name : IP
-----
eth0           : 192.168.182.129

```

3. 板卡配置与通讯测试

3.1 板卡配置

HK/CIFX PC板卡主要应用是可作为工业实时以太网或现场总线协议的主站或从站，如Profinet主从站，Ethernet/IP主从站，EtherCAT主从站。HK/CIFX PC板卡可通过加载不同的协议固件，使得板卡可以充当相应的主从站，对于cifX设备，固件及其配置文件不存储在硬件上，因此还需要给Linux环境中提供必要的文件，工控机每次上电时会将boot loader、firmware、configuration file加载到硬件板卡中。

(1、创建特定的文件结构

为了允许特定设备的配置，板卡识别，固件需要被存储在主机上的特定文件夹结构中，以建立板卡和固件之间的唯一关系。默认情况下，所有相关的文件位于"/opt/cifx"目录中。驱动程序支持如下四种不同类型的板卡的配置，每一种都有其特定的文件夹结构。

slotnumber (depends on the hardware, requires slotnumber switch on the hardware)

device and serial number

card name

single directory

一种比较简单的设置配置文件存储的方式是通过位于驱动/driver/scripts/中的安装脚本

install_firmware来创建相应的文件结构。

- First install the second stage boot loader by calling (root privileges are required)
./install_firmware install
This creates the folder '/opt/cifx/deviceconfig' and copies the second stage boot loader to '/opt/cifx'
- Depending on the chosen configuration file storage method, execute one of the following commands (root privileges are required)
Device identification via device and serial number:
./install_firmware add_device [device no] [serial no]
Device identification via slotnumber:
./install_firmware add_slot_dir [slot no]
Device identification via single directory:
./install_firmware create_single_dir

本文章示例中采用single directory实现文件存储，建立板卡与固件之间关系，实现设备识别与固件下载。默认执行cifx板卡驱动安装脚本后，所创建好的文件目录与single directory方式一致，因此如果采用此方式可省略此步。

Subdirectory	Description
<BASEDIR>	Base directory Default: '/opt/cifx' Can be changed during userspace library initialization. This directory must contain the second stage PCI boot loader (e.g. NETX100-BSL.BIN).
deviceconfig	Device specific configuration files
FW	If <i>single directory</i> is used, the search path is set to - BASEDIR ~/deviceconfig/FW Contains the <i>device.conf</i> which holds the device specific settings Note: This directory must contain the rcX base firmware if loadable modules are used.
channel-<#>	Channel specific files <ul style="list-style-type: none"> ▪ firmware file (*.nxf - e.g. cifxdpm.nxf) ▪ fieldbus configuration file (*.nxd - e.g. config.nxd) ▪ firmware loadable module file (*.nxo) Note: Currently only channel 0 is supported

(2、板卡组态配置

在Windows系统中安装板卡配置软件SYCON.net，在SYCON.net中对板卡参数进行网络组态，如输入输出数据量、通讯周期、IP地址等，并将通讯正常的配置保存，具体配置可参考相关案例。从SYCON.net导出.nxd格式的数据库文件或者从Windows板卡默认配置文件存储目录中拷贝出来（通常建议后者）。并将.nxd数据库文件和.nxf协议栈固件文件拷贝到Linux设备配置目录中，single directory方式保存/opt/cifx/deviceconfig/FW/channel0中。

需要重启工控机才能使得固件和配置加载到cifx板卡。

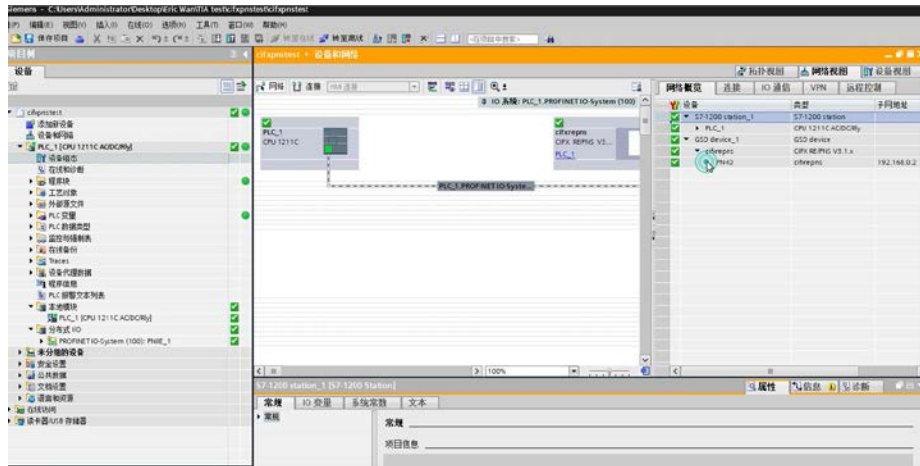
```

root@ubuntu:/opt# tree
├── cifx
│   ├── bsl_history.txt
│   ├── deviceconfig
│   │   └── FW
│   │       ├── channel0
│   │       │   ├── cifxpns.nxf
│   │       │   ├── CONFIG.nxd
│   │       │   └── NWID.nxd
│   │       └── device.conf
│   ├── NETX100-BSL.bin
│   ├── NETX10-BSL.bin
│   ├── NETX50-BSL.bin
│   ├── NETX51-BSL.bin
│   └── NETX52-BSL.bin

```

3.2 通讯测试

本文章以HK/CIFX PC板卡作为PROFINET从站为例：首先在西门子PLC博途软件中进行PROFINET的网络组态与配置，注意cifx板卡所添加的输入输出模块与SYCON.net软件中配置的一致，并下载给PLC，建立PLC与cifx板卡之间的PROFINET链接。



然后在Linux环境下运行程序，这里是以运行examples中cifXTestConsole为例，进入文件夹 /examples/cifXTestConsole,先执行configure脚本（./configure），再make编译程序，最后执行生成的cifXTestConsole（./ cifXTestConsole），执行程序后可以查看到板卡基本信息。


```

root@eric-Default-string:/home/project/examples/cifXTestConsole# ls
aclocal.m4          config.h            depcomp            NEWS
AUTHORS            config.h.in        INSTALL           OS_Includes.h
autom4te.cache     config.log         install-sh        OS_Specific.c
ChangeLog          config.status      Main.cpp          README
cifXTestConsole   configure          Makefile         shared
cifXTestConsole-Main.o  configure.ac      Makefile.am      stamp-h1
cifXTestConsole-OS_Specific.o  COPYING          Makefile.in      VersionHistory
compile           cross_build_driver  missing

root@eric-Default-string:/home/project/examples/cifXTestConsole# ./cifXTestConsole
*****
*** cifX Driver Test Program
*****

cifXDriverInit...
*****
*** CIFx Driver Test Program
*****

Driver Version: cifX Toolkit 1.4.0.0

Board0 Information:
Name : cifX0
Alias: MyAlias
DevNr: 1251100
SN   : 34807

Channel0 Information:
Channel Error      : 0xC0000145
Board Name        : cifX0
Alias Name        : MyAlias
Device Nr.        : 1251100
Serial Nr.        : 34807
MBX Size          : 1596
Firmware Name     : PROFINET IO Device
Firmware Version  : 3.4.0 Build 47
Open Counter      : 0
Put Packet Counter: 1
Get Packet Counter: 1
Number of IO Input Areas : 2
Number of IO Output Areas: 2
Size of handshake cells : 2
Actual netX Flags : 0x0000003A
Actual host Flags  : 0x00000038

Channel1 Information:
Channel Error      : 0x00000000
Board Name        : cifX0
Alias Name        : MyAlias
Device Nr.        : 1251100
Serial Nr.        : 34807
MBX Size          : 1596
Firmware Name     : Ethernet Interface
Firmware Version  : 4.2.0 Build 0
Open Counter      : 0
Put Packet Counter: 1
Get Packet Counter: 1

```

执行cifXTestConsole例程中IO数据交互，可实现与PLC之间的周期性数据收发。

```

root@eric-Default-string:/home/project/examples/cifXTestConsole
IN byte 0,1 = 0xFF, 0xFF, OUT byte 0,1 = 0x55, 0xFF
IN byte 0,1 = 0xFF, 0xFF, OUT byte 0,1 = 0x56, 0xFF
IN byte 0,1 = 0xFF, 0xFF, OUT byte 0,1 = 0x57, 0xFF
IN byte 0,1 = 0xFF, 0xFF, OUT byte 0,1 = 0x58, 0xFF
IN byte 0,1 = 0xFF, 0xFF, OUT byte 0,1 = 0x59, 0xFF
IN byte 0,1 = 0xFF, 0xFF, OUT byte 0,1 = 0x5A, 0xFF
IN byte 0,1 = 0xFF, 0xFF, OUT byte 0,1 = 0x5B, 0xFF
IN byte 0,1 = 0xFF, 0xFF, OUT byte 0,1 = 0x5C, 0xFF
IN byte 0,1 = 0xFF, 0xFF, OUT byte 0,1 = 0x5D, 0xFF
IN byte 0,1 = 0xFF, 0xFF, OUT byte 0,1 = 0x5E, 0xFF
IN byte 0,1 = 0xFF, 0xFF, OUT byte 0,1 = 0x5F, 0xFF
IN byte 0,1 = 0xFF, 0xFF, OUT byte 0,1 = 0x60, 0xFF
IN byte 0,1 = 0xFF, 0xFF, OUT byte 0,1 = 0x61, 0xFF
IN byte 0,1 = 0xFF, 0xFF, OUT byte 0,1 = 0x62, 0xFF

```

虹科云课堂

HongKe Online Academy

2020年2月21日,虹科云课堂首次与大家见面,带来的第一节《CAN总线基础之物理层篇》课程,就得到了各位工程师朋友们的热情支持与参与,当晚观看人数4900+。我们非常感恩,愿不负支持与鼓励,致力将虹科云课堂打造成干货知识共享平台。

目前虹科云课堂的全部课程已经超过200节,如下表格是我们汽车相关的部分课程列表,大家通过微信扫描二维码关注公众号,点击免费课程直接进入观看,全部免费。

汽车以太网课程

智能网联下车载以太网的解决方案
SOME/IP协议介绍
基于CanEasy浅谈XCP
TSN/AVB 基于信用点的整形

TSN技术课程

基于TSN的汽车实时数据传输网络解决方案
TSN时间敏感型网络技术综述
以太网流量模型和仿真
基于TSN的智能驾驶汽车E/E架构设计案例分享
IEEE 802.1AS 时间同步机制
TSN技术如何提高下一代汽车以太网的服务质量?

CAN、CAN FD、CAN XL总线课程

CAN总线基础之物理层篇
CAN数据链路层详解篇
CAN FD协议基础
CAN总线一致性测试基本方法
CAN测试软件(PCAN-Explorer6)基本使用方法
CAN测试软件(PCAN-Explorer6)高级功能使用
浅谈CAN总线的最新发展: CAN FD与CAN XL
CAN线的各种故障模式波形分析

LIN总线相关课程

汽车LIN总线基本协议概述
汽车LIN总线诊断及节点配置规范
LIN总线一致性测试基本方法
LIN自动化测试软件(LINWorks)基本使用方法
LIN自动化测试软件(LINWorks)高级功能使用
基于CAN/LIN总线的汽车零部件测试方案

CAN高级应用课程

UDS诊断基础
UDS诊断及ISO27145
基于UDS的ECU刷写
基于PCAN的二次开发方法
CCP标定技术
J1939及国六排放
OBD诊断及应用(GB3847)
BMS电池组仿真测试方案
总线开发的流程及注意事项
车用总线深入解析

汽车测修诊断相关课程

汽车维修诊断大师系列-如何选择示波器
汽车维修诊断大师系列-巧用示波器
汽车维修诊断-振动异响(NVH)诊断方案

工业通讯协议基础课程

PROFINET协议基础知识
初识EtherCAT协议
初识CANopen协议
EtherNet/IP协议基础知识
IO-Link: 工业物联网的现场基础
新兴工业级无线技术IO-Link Wireless



关注获取最新课程



汽车电子bilibili主页



工业智能互联
bilibili主页

智能通讯领域专业的 资源整合及技术服务落地供应商

关于虹科

虹科电子科技有限公司（前身是宏科）成立于1995年，总部位于中国南方经济和文化中心-广州；还在上海、北京、成都、西安、苏州、台湾、香港，韩国和日本设有分公司。

我们是一家高新技术公司，是广东省特批的两高四新、三个一批、专精特新和瞪羚企业，并与全球顶尖公司有多领域的深度技术合作，业务包括工业自动化和数字化、汽车研发测试、自动驾驶等领域；医药和风电行业等的环境监测；半导体、轨道交通、航空航天等测试测量方案。

虹科工程师团队致力于为行业客户提供创新产品和解决方案，全力帮助客户成功。

智能互联事业部

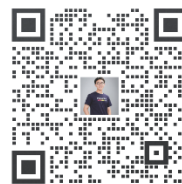
虹科是一家在通讯领域，尤其是汽车电子和智能自动化领域拥有超过 15 年经验的高科技公司，致力于为客户提供全方位的一站式智能互联解决方案。多年来，我们与全球行业专家深度合作，成为了行业内领先的通讯技术服务商。我们提供全面的软硬件解决方案，包括【CAN/CAN FD、LIN、车载以太网、TSN、IO-Link/IO-Link wireless、OPC UA、CANopen、PROFINET、EtherNet/IP、EtherCAT】等各类通讯协议的解决方案、测试方案、培训和开发服务等。

我们以满足客户需求为导向，以技术能力为基础，为国内外企业提供最适合的产品和最满意的服务。目前我们服务的客户已经超过 5000 家，我们自主研发的 EOL 测试系统、CCP/XCP 标定和 UDS 诊断服务开发服务以及 TSN 网络验证测试系统等也已经在业内完成超过 1000 次安装和测试。我们的方案覆盖了各行业知名企业，得到了包括蔚来，比亚迪，长城，联影，东芝三菱，安川等多个用户的一致好评。



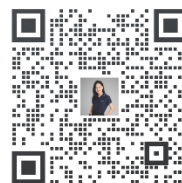
华东区（上海）销售
高印祺

电话/微信: 136 6024 4187
邮箱: gao.yinqi@intelnect.com



华东区（非上海）销售
林燕芬

电话/微信: 135 1276 7172
邮箱: lin.yanfen@intelnect.com



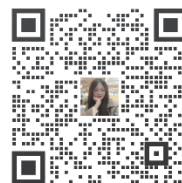
华南区销售
董欢

电话/微信: 189 2224 3009
邮箱: dong.huan@intelnect.com



华北区销售
张瑞婕

电话/微信: 181 3875 8797
邮箱: zhang.ruijie@intelnect.com



协议开发方案（全国）
郭泽明

电话/微信: 189 2224 2268
邮箱: guo.zeming@intelnect.com



HongKe
虹科

虹科电子科技有限公司

www.intelnect.com
info@intelnect.com

广州市黄埔区开泰大道30号佳都PCI科技园6号楼

T (+86)400-999-3848

各分部: 广州 | 成都 | 上海 | 苏州 | 西安 |
北京 | 台湾 | 香港 | 日本 | 韩国

版本: V1.0 - 22/11/14



获取工业行业资料



获取汽车行业资料