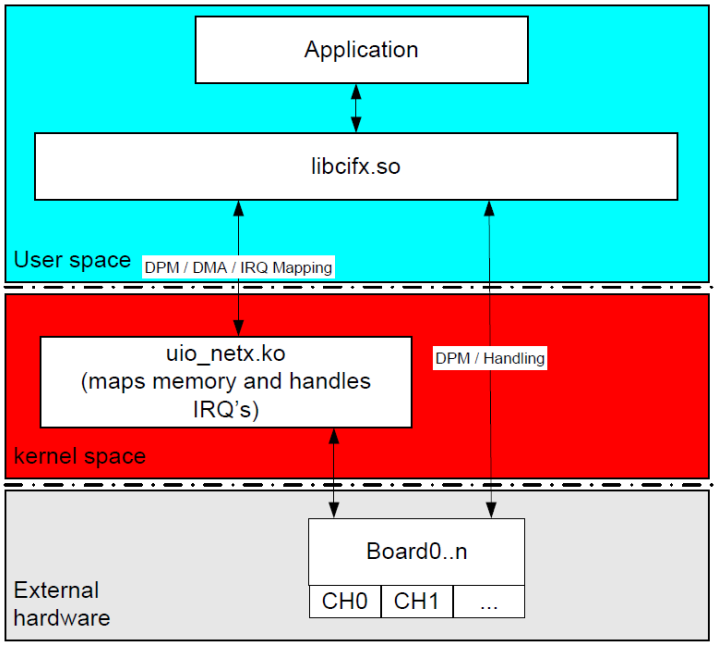
HK-CIFX PC 板卡 Centos 环境下驱动安装及通讯配置操作指南

——万彬，技术支持工程师，2021.07.29

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 Centos 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 Centos 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。Centos 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. Centos 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 Centos 组件更新

安装驱动之前，根据 HK/CIFX PC 板卡在 Centos 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 Centos 系统联网情况下（注意修改/etc/sysconfig/network-scripts/路径对于网络配置文件中 ONBOOT = yes），可通过 sudo yum install 命令更新 libpciaccess 库（yum install libpciaccess-devel），pkg-config 可参考如下说明安装：

1、下载 pkg-config：wget https://pkg-config.freedesktop.org/releases/pkg-config-0.29.tar.gz

2、解压：tar -zxvf pkg-config-0.29.tar.gz

3、cd 进入解压文件夹

4、运行配置文件进行系统配置 : ./configure --with-internal-glib

5、编译 pkgconfig : make

6、安装包自检测 ： make check

7、安装 ： make install

安装完成后，建议先确认下 libpciaccess 库所在路径及其 pciaccess.pc 文件所包含路径，与 PKG_CONFIG_PATH 环境变量路径是否一致，不一致通过 export 命令额外添加上，避免编译和安装 CIFX Linux 驱动时出“libpciaccess not found via pkg-config”。

查看 PKG_CONFIG_PATH 当前路径：echo $PKG_CONFIG_PATH

加上某库的 pkg 路径：export PKG_CONFIG_PATH=/usr/lib/pkgconfig/

2.3 解压驱动包

复制驱动压缩包到 Centos 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，这里通过 mv 命令重新命名为 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。

2.4 安装驱动

（1、进入驱动文件夹/home/project/driver，参考 ReadMe 中安装 uio_netx 说明按步骤执行操作：

```
Open the console and change in the directory containing this document.

1) Build and install the kernel modules
1.1) Enter folder script (e.g. cd ./script)
1.2) To build the modules run "./install_uio_netx build"     (root required)
1.3) To install the modules run "./install_uio_netx install" (root required)
1.4) To load the modules run "./install_uio_netx load"       (root required)

2) Install the bootloader
2.1) Enter folder script (e.g. cd ./script)
2.2) run the script "./install_firmware install" (root required)

3) Build and install the user space library libcifx
3.1) Enter folder libcifx (e.g. cd ./libcifx)
3.2) Run "./configure"
3.3) Run "make"
3.4) Run "make install" (root required)
```

之后执行驱动安装脚本 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 Centos 组件更新一节说明，安装必要的组件。

（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/usr/local/lib 目录中可以查看到已安装的 libcifx 相关的库文件，头文件在/usr/local/include/cifx 中。

（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 Centos 模块自启动内，可以直接在/etc 目录下 modules 文件中添加 uio_netx。

以及若例程编译后执行时时可能出现无法链接到 libcifx 库的情况，提示找不到相关的.so 库报错。也就是说，/usr/local/lib 目录不在系统默认的库搜索目录中，需要将目录加进去：

1、首先打开/etc/ld.so.conf 文件；

2、加入动态库文件所在的目录：执行 vi /etc/ld.so.conf，在"include ld.so.conf.d/*.conf"下方增加"/usr/local/lib";

3、保存后，在命令行终端执行：/sbin/ldconfig -v；其作用是将文件/etc/ld.so.conf 列出的路径下的库文件缓存。

或者出现：Linux make && make install 时出现 missing aclocal-1.14 -I m4 make: *** [aclocal.m4] 错误 127。这时：先执行 autoreconf -ivf 加载缺失文件 后 再执行 make 就可以。

以上操作完成。重启 Centos 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 Centos 下使用。

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 Centos 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 *single directory* is used, the search path is set to <**BASEDIR**>/**deviceconfig/FW** Contains the *device.conf* which holds the device specific settings **Note:** This directory must contain the rcX base firmware if loadable modules are used.
channel<#>	Channel specific files ▪ 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 Centos 设备配置目录中，single directory 方式保存

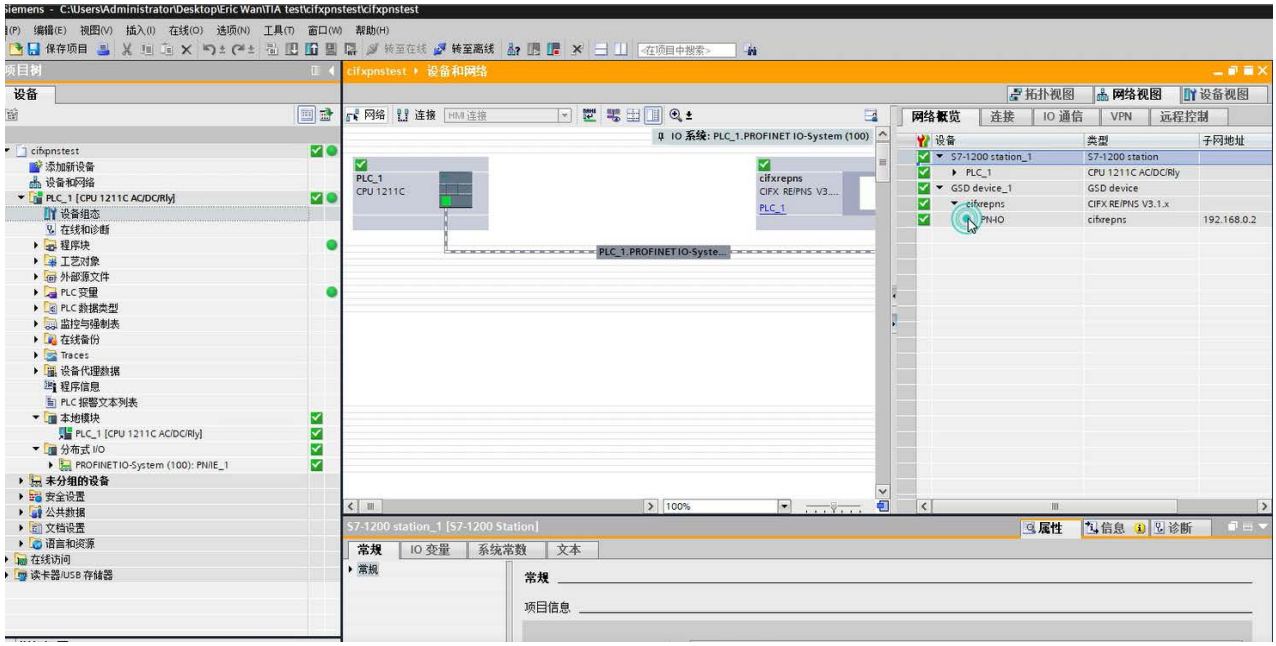
/opt/cifx/deviceconfig/FW/channel0 中。

需要重启工控机才能使得固件和配置加载到 cifx 板卡。

```
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，注意在博途中导入的 gsd 文件版本以及 cifx 板卡所添加的输入输出模块需与 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 Centos 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。

```
*******************************************************************************
*** 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
```

根据需要执行相关测试。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。

```
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
```