

# OpenTSN 网络控制器使用手册

## （版本 1.1）

OpenTSN 开源项目组

2020 年 04 月

## 版本历史

版本	修订时间	修订内容	文件标识
1.0	2021.01.28	完成 TSN 集中控制器使用手册的初步版本	OpenTSN3.0 工程使用
1.1	2021.04.02	修改文件格式，按照统一的文件标准重新修改文本	

## 目录

一、引言.....	4
二、运行环境.....	4
三、文件说明 .....	4
3.1 ARP 应用 .....	5
3.2 通用函数库 .....	5
3.3 状态监测应用.....	7
3.4 PTP 时间同步应用 .....	8
四、编译和运行步骤 .....	9
4.1 编译.....	9
4.2 运行.....	10
附录一： Libxml2 库安装教程.....	11
附录二： 问题记录.....	12

## 一、引言

本文档为TSN集中控制器使用手册，主要描述控制器运行环境、文件说明、编译和运行步骤、组网示例，用户可以参考该文档使用网络控制器。

网络控制器主要功能包含 ARP 代理、网络初始配置、状态监测和 PTP 时间同步，所有功能需要与硬件配合才能实现。

## 二、运行环境

网络控制器的运行环境为 Linux 系统，需要安装以下库支撑程序运行。

- 需要安装 libpcap 和 libnet 库，用于收包和发包。
- 需要安装 libxml2 库，用于进行 xml 文件解析。
- Linux 设备需要把网络接口开启混杂模式

## 三、文件说明

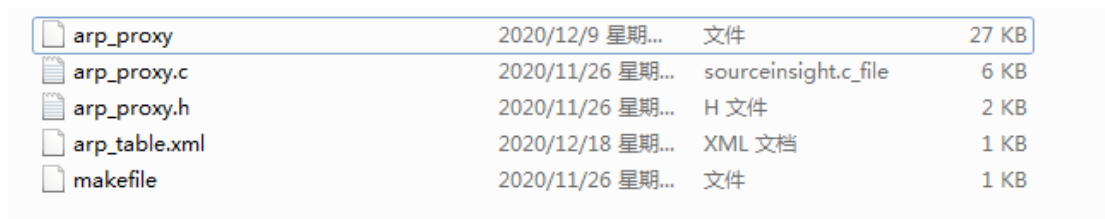
网络控制器包含的源文件如下图所示，arp 文件夹下存放的是 arp 应用程序，cnc\_api 文件下存放通用基础库，cnc\_ptp 文件下存放 PTP 时间同步应用程序，net\_init 存放网络初始化应用程序，state\_monitor 文件夹存放状态监测应用程序。

名称	修改日期	类型	大小
arp	2020/12/19 星期...	文件夹	
cnc_api	2020/12/19 星期...	文件夹	
cnc_ptp	2020/12/22 星期...	文件夹	
net_init	2020/12/19 星期...	文件夹	
state_monitor	2020/12/19 星期...	文件夹	

图 3-1 网络控制器源文件目录

### 3.1 ARP 应用

arp 文件夹存放 ARP 代理应用程序，用于响应 ARP 请求报文，包含的文件如下图所示。



arp_proxy	2020/12/9 星期...	文件	27 KB
arp_proxy.c	2020/11/26 星期...	sourceinsight.c_file	6 KB
arp_proxy.h	2020/11/26 星期...	H 文件	2 KB
arp_table.xml	2020/12/18 星期...	XML 文档	1 KB
makefile	2020/11/26 星期...	文件	1 KB

图 3-2 ARP 应用源文件目录

各个文件的具体含义：

- arp\_proxy: 进行编译后生成的可执行文件
- arp\_proxy.c: arp 代理的源文件，所有的函数在该文件中实现
- arp\_proxy.h: arp 代理的头文件，arp 数据结构定义在该文件里面。
- arp\_table.xml: 用于存放 arp 表项
- makefile: 可以进行编译的文件

### 3.2 通用函数库

cnc\_api 文件夹为基础函数库，包含 include 文件夹（主要存放通用函数的 api 头文件）和 src 文件夹，包含的文件如下图所示。

名称	修改日期	类型	大小
include	2020/12/19 星期...	文件夹	
src	2020/12/19 星期...	文件夹	
libcnc_api.a	2020/12/17 星期...	A 文件	58 KB
makefile	2020/11/26 星期...	文件	1 KB

图 3-3 通用函数库源文件目录

各个文件的具体含义：

- include：主要存放通用函数的 API 头文件和通用的数据结构定义
- src：通用 API 的实现
- libcnc\_api.a：编译基础函数生成的动态库文件
- makefile：可以进行编译的文件

include 文件夹中包含的文件：

beacon_report.h	2020/12/15 星期...	H 文件	13 KB
cnc_api.h	2020/12/10 星期...	H 文件	4 KB
reg_cfg.h	2020/12/8 星期...	H 文件	5 KB
tsmp_protocol.h	2020/11/24 星期...	H 文件	3 KB

图 3-4 include 源文件目录

- beacon\_report.h：芯片上报报文的数据结构定义
- cnc\_api.h：通用 API 的头文件
- reg\_cfg.h：芯片配置和 HCP 配置报文的数据结构定义
- tsmp\_protocol.h：tsmp 协议的数据结构定义

src 文件夹中包含的文件：

beacon_report.c	2020/12/2 星期...	sourceinsight.c_f...	1 KB
beacon_report.o	2020/12/17 星期...	O 文件	4 KB
data_rec_engine.c	2020/12/10 星期...	sourceinsight.c_f...	3 KB
data_rec_engine.o	2020/12/17 星期...	O 文件	9 KB
data_send_engine.c	2020/11/24 星期...	sourceinsight.c_f...	3 KB
data_send_engine.o	2020/12/17 星期...	O 文件	9 KB
reg_cfg.c	2020/12/17 星期...	sourceinsight.c_f...	21 KB
reg_cfg.o	2020/12/17 星期...	O 文件	27 KB
tsmp_protocol.c	2020/11/24 星期...	sourceinsight.c_f...	4 KB
tsmp_protocol.o	2020/12/17 星期...	O 文件	9 KB

图 3-5 cnc\_api 中 src 源文件目录

- beacon\_report.c: 解析芯片上报报文和 HCP 上报报文的通用函数实现
- beacon\_report.o: 编译 beacon\_report.c 生成的目标文件
- data\_rec\_engine.c: 数据接收通用函数实现
- data\_rec\_engine.o: 编译数据接收文件生成的目标文件
- data\_send\_engine.c 数据发送通用函数实现
- data\_send\_engine.o: 编译数据发送文件生成的目标文件
- reg\_cfg.c: 寄存器配置通用函数实现
- reg\_cfg.o: 编译寄存器配置文件生成的目标文件
- tsmp\_protocol.c: tsmp 协议通用函数实现
- tsmp\_protocol.o: 编译 tsmp 协议文件生成的目标文件

### 3.3 状态监测应用

state\_monitor 文件夹存放状态监测应用程序的源文件，用于对网络进行状态检测，具体报文源文件如下图所示：


	makefile	2020/12/2 星期...	文件	1 KB
	monitor	2020/12/15 星期...	文件	51 KB
	state_monitor.c	2020/12/15 星期...	sourceinsight.c_f...	16 KB
	state_monitor.h	2020/12/2 星期...	H 文件	5 KB
	topology_info.xml.xml	2020/12/11 星期...	XML 文档	1 KB

图 3-6 状态监测源文件目录

各个文件的具体含义：

- **makefile**: make 该文件生成可执行文件 **monitor**
- **monitor**: 生成的可执行文件
- **state\_monitor.c**: 状态监测模块的主函数和关键函数的实现
- **state\_monitor.h**: 状态监测的头文件
- **topology\_info.xml.xml**: 拓扑信息的 xml 文本

### 3.4 PTP 时间同步应用

**cnc\_ptp** 文件夹存放时间同步 PTP 应用程序的源文件，用于实现网络时间同步的功能，具体报文源文件如下图所示：






	libptp	2020/12/21 16:22	文件夹	
	cnc_ptp	2020/12/18 11:12	文件	59 KB
	libptpl.a	2020/12/18 11:12	A 文件	40 KB
	main	2020/12/18 11:12	C source file	2 KB
	makefile	2020/11/26 8:56	文件	1 KB

图 3-7 时间同步应用 PTP 源文件目录

各个文件的具体含义：

- **libptp**: 主要存放时间同步相关函数的 API 头文件和数据结构定义以及时间同步相关函数 API 的实现
- **cnc\_ptp**: 生成的可执行文件
- **Libptpl.a**: 编译 PTP 时间同步函数生成的动态库文件



- main: 时间同步主函数
- makefile: make 该文件生成可执行文件 cnc\_ptp

libptp 文件夹中包含的文件:







	ptp	2020/12/18 10:49	C source file	18 KB
	ptp	2020/12/18 10:44	Header file	3 KB
	ptp.o	2020/12/18 11:12	O 文件	28 KB
	timer	2020/12/18 10:41	C source file	3 KB
	timer	2020/11/25 22:39	Header file	1 KB
	timer.o	2020/12/18 11:12	O 文件	11 KB

图 3-8 cnc\_ptp 中 libptp 源文件目录

- ptp.c: PTP 同步处理模块的关键函数实现
- ptp.h: PTP 同步处理的头文件
- ptp.o: 编译时间同步处理文件生成的目标文件
- timer.c: 定时器模块的关键函数实现
- timer.h: 定时器处理的头文件
- timer.o: 编译定时器处理文件生成的目标文件

## 四、编译和运行步骤

本章主要讲述程序运行之前对程序的编译, 和程序运行的主要步骤。

### 4.1 编译

在编译时按以下步骤进行操作:

(1) 首先编译 cnc\_api 文件夹下的库文件, 在 cnc\_api 文件夹下执行以下命令

- make clean

- make

(2) 编译 net\_init 文件夹下的网络初始化进程

- make clean

- make

(3) 编译 state\_monitor 文件夹下的状态监测进程

- make clean

- make

(4) 编译 arp 文件夹下的 arp 代理进程

- make clean

- make

(5) 编译 cnc\_ptp 文件夹下的 ptp 时间同步代理进程

- make clean

- make

## 4.2 运行

在运行时按以下步骤进行操作（注：在 root 权限下执行）：

(1) 运行网络初始化进程

- 在 net\_init 目录下执行：./init 网络接口名

本设备的网络接口名为 enp0s17，因此在执行时输入为 ./init

enp0s17

```
root@ubuntu:/mnt/hgfs/cnc/cnc_api# ifconfig
enp0s17 Link encap:以太网 硬件地址 08:00:27:b9:68:3c
        inet 地址:192.168.1.30 广播:192.168.1.255 掩码:255.255.255.0
        inet6 地址: fe80::bdd0:b73f:bc4:ad1/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 跃点数:1
        接收数据包:86635107 错误:0 丢弃:2 过载:0 帧数:0
        发送数据包:1128917 错误:0 丢弃:0 过载:0 载波:0
        碰撞:0 发送队列长度:1000
        接收字节:13623726052 (13.6 GB) 发送字节:609972357 (609.9 MB)

lo      Link encap:本地环回
        inet 地址:127.0.0.1 掩码:255.0.0.0
        inet6 地址: ::1/128 Scope:Host
        UP LOOPBACK RUNNING MTU:65536 跃点数:1
        接收数据包:4434 错误:0 丢弃:0 过载:0 帧数:0
        发送数据包:4434 错误:0 丢弃:0 过载:0 载波:0
        碰撞:0 发送队列长度:1
        接收字节:372046 (372.0 KB) 发送字节:372046 (372.0 KB)
```

图 4-1 网络接口名查看结果

## (2) 运行状态监测进程

- 在 state\_monitor 目录下执行: ./monitor 网络接口名

## (3) 运行 arp 代理进程

- 在 arp 目录下执行: ./arp\_proxy 网络接口名

## (4) 运行 PTP 时间同步进程

- 在 cnc\_ptp 目录下执行: ./cnc\_ptp

## 附录一: Libxml2 库安装教程

- 拷贝 Libxml2 库文件夹到 Linux 机器中
- 进入 libxml2-2.6.2 文件夹目录下, 在终端中打开
- 拷贝库文件夹到 Linux 机器中
- 进入 libxml2-2.6.2 文件夹目录下, 在终端中打开
- 使用 root 权限执行以下命令
  - make distclean
  - ./configure
  - make
  - make install

## 附录二：问题记录

- 执行 `./init enp0s17` 时，出现没有可执行权限，需要执行 `chmod 777 init`，赋予 `init` 文件可执行权限