

# TSN 集中控制器使用手册

## （单进程版本 1.0）

OpenTSN 开源项目组

2021 年 5 月

## 版本历史

| 版本  | 修订时间      | 修订内容   | 修订人        | 文件标识                                 |
|-----|-----------|--|------------|--------------------------------------|
| 1.0 | 2021.5.18 | 按照实际测试使用的情<br>况撰写初稿。<br><br>填充网络初始化、配置、<br>cnc_api 的说明 | 孙寅涵<br>李军帅 | TSNLight3.0<br>单进程<br>集中式控制器<br>使用手册 |
|     |           |  |            |                                      |
|     |           |  |            |                                      |
|     |           |  |            |                                      |
|     |           |  |            |                                      |
|     |           |  |            |                                      |
|     |           |  |            |                                      |
|     |           |  |            |                                      |

## 目录

|                           |    |
|---------------------------|----|
| 1、引言 .....                | 4  |
| 2、详细说明 .....              | 4  |
| 1. 文件说明 .....             | 4  |
| (1) arp_proxy .....       | 5  |
| (2) basic_cfg .....       | 5  |
| (3) cnc_api .....         | 6  |
| (4) local_cfg .....       | 6  |
| (5) net_init.....         | 7  |
| (6) ptp.....              | 7  |
| (7) remote_cfg.....       | 8  |
| (8) state_monitor.....    | 8  |
| (9) main.c .....          | 8  |
| (10) makefile.....        | 8  |
| 2. 程序运行 .....             | 9  |
| 附录 A: Libxml2 库安装教程 ..... | 10 |
| 附录 B: 问题记录 .....          | 10 |

## 1、引言

本文档为 OpenTSN3.0 版本的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。

此版本控制器与历史版本最显著的修改是实现方式从多进程到单进程，从而能够提高在单核处理器嵌入式设备等资源受限场景控制器的性能。同时，控制器还新增了与 NETCONF 代理通信，获取控制信息并配置网络等功能。

当前版本的控制器主要功能包含网络初始化、本地配置、远程配置、ARP 代理、网络基础配置、状态监测和 PTP 时间同步，所有功能需要与硬件配合才能正常运行。

## 2、详细说明

### 1. 文件说明

按照不同功能分类，工程文件夹主目录共包含 8 个子文件夹和 3 个文件。如下图所示。

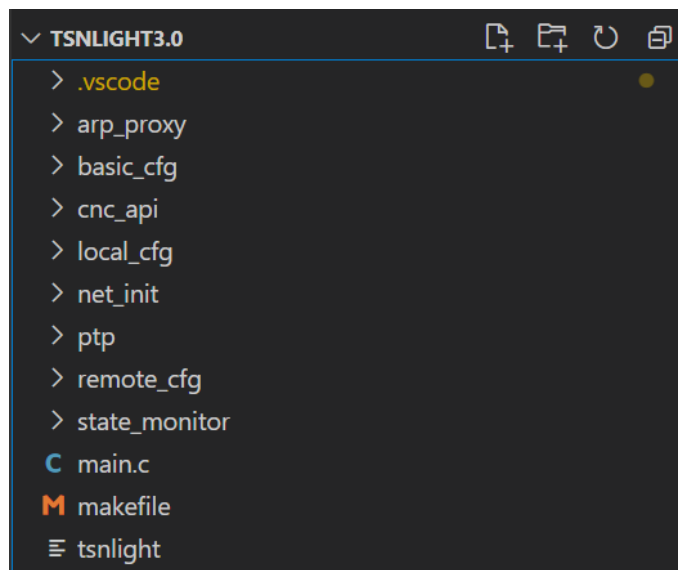


图 1-1 工程文件目录

### (1) arp\_proxy

该文件夹包含的是与 ARP 代理相关的代码文件。

表 2-1 ARP 代理代码文件说明

| 文件名称          | 内容                            |
|---------------|-------------------------------|
| arp.h         | 声明 ARP 代理功能主要的数据结构和编程接口       |
| arp.c         | 编程接口的具体实现，全局变量声明              |
| arp_table.xml | 代码运行时会读取该文件，获取 ARP 表存到对应的数据结构 |

### (2) basic\_cfg

该文件夹包含的是与基础配置相关的代码文件。

表 2-2 基础配置代码文件说明

| 文件名称        | 内容                          |
|-------------|-----------------------------|
| basic_cfg.h | 基础配置的头文件，声明基础配置的主要数据结构和编程接口 |
| basic_cfg.c | 基础配置的源文件，主要实现基础配置的功能        |

### (3) cnc\_api

该文件夹包含的是通用 API 相关的代码文件，该文件夹下包含 include 文件夹和 src 文件夹。

include 文件下主要包含通用函数库的头文件

表 2-3 通用 api 头文件说明

| 文件名称            | 内容  |
|-----------------|---|
| beacon_report.h | 声明上报信息的数据结构，包含上报寄存器的格式，上报统计信息的格式和状态信息的数据格式                      |
| cnc_api.h       | 包含通用头文件，各个状态的定义以及通用 API 的声明                                     |
| reg_cfg.h       | 定义各个配置寄存器和表项的地址，配置内容的数据格式，配置报文的数据格式                             |
| tsmp_protocol.h | 主要声明关于 tsmp 协议，包含有符号和无符号的重定义，TSMP 子类型的定义、TSNTag 的定义以及 TSMP 头的定义 |

src 文件下主要包含通用函数库的源文件

表 2-4 通用 api 源文件说明

| 文件名称               | 内容   |
|--------------------|--|
| beacon_report.c    | Beacon 上报报文的源程序，目前只有获取上报类型的函数                |
| data_rec_engine.c  | 数据接收的源文件，主要包含数据接收初始化函数、数据接收函数                |
| data_send_engine.c | 数据发送的源文件，主要包含数据发送初始化函数、数据发送处理函数              |
| reg_cfg.c          | 网络配置的源文件，包含寄存器配置函数、表项配置函数、hcp 配置函数，以及大小端转换函数 |
| tsmp_protocol.c    | tsmp 协议源文件，包含 tsmp 报文构造和解析函数                 |

### (4) local\_cfg

该文件夹包含的是与本地配置相关的代码文件。

表 2-5 本地配置文件说明

| 文件名称              | 内容   |
|-------------------|--|
| local_cfg.h       | 本地配置的头文件，声明本地配置的数据结构以及函数声明                   |
| local_cfg.c       | 本地配置的源文件，实现本地配置的功能                           |
| local_cfg_xml.xml | 代码运行时会读取该文件，获取离线规划的本地配置信息，然后解析该文本中的内容，配置到硬件中 |

#### (5) net\_init

该文件夹包含的是网络初始化相关的代码文件。

表 2-6 网络初始化文件说明

| 文件名称             | 内容   |
|------------------|--|
| net_init.h       | 声明网络初始化主要的数据结构和编程接口  |
| net_init.c       | 网络初始化编程接口的具体实现，全局变量声明  |
| init_cfg_xml.xml | 代码运行时会读取该文件，获取网络基础配置的内容，该内容具体包含节点寄存器参数、转发表。（该部分的目的是使配置报文能够到达各个节点，上报报文可以到达控制器）。 |

#### (6) ptp

该文件夹包含的是与 ptp 时间同步相关的代码文件。

表 2-7 PTP 时间同步文件说明

| 文件名称                 | 内容                   |
|----------------------|----------------------|
| ptp_single_process.h | 声明时间同步功能主要的数据结构和编程接口 |
| ptp_single_process.h | 编程接口的具体实现，全局变量声明     |

### (7) remote\_cfg

该文件夹包含的是与远程配置相关的代码文件。

表 2-8 远程配置文件说明

| 文件名称         | 内容                   |
|--------------|----------------------|
| remote_cfg.h | 声明远程配置主要的数据结构和编程接口   |
| remote_cfg.c | 远程配置编程接口的具体实现，全局变量声明 |

### (8) state\_monitor

该文件夹包含的是与状态监测相关的代码文件。

表 2-9 状态监测文件说明

| 文件名称                  | 内容                            |
|-----------------------|-------------------------------|
| state_monitor.h       | 声明状态监测功能主要的数据结构和编程接口          |
| state_monitor.c       | 编程接口的具体实现，全局变量声明              |
| topology_info_xml.xml | 代码运行时会读取该文件，获取网络拓扑信息存到对应的数据结构 |

### (9) main.c

该文件中包含了控制器的主函数，实现了整个控制器运行时候的状态跳转和流程。

### (10) makefile

该文件是整个工程的编译文件 makefile，在主文件夹下执行 make 指令可以编译生成可执行文件 tsnlight。当需要清空编译的中间文件



和最终文件时，在主文件夹下执行 `make clean` 指令。

## 2. 程序运行

首先，在主文件目录下执行指令 `make`，会生成可执行文件 `tsnlight`;

```
make
```

然后执行 `sudo ./tsnlight [interface]`, 注意这一步需要使用管理员权限，参数是用于接收和发送报文的网卡名称。

```
sudo ./tsnlight enp1s0
```

## 附录 A: Libxml2 库安装教程

- 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
- 执行 ./configure
- make
- make install

## 附录 B: 问题记录

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