AS6802 时间同步控制软件

使用手册

（版本 1.0）

OpenTSN 开源项目组

2022 年 05 月

版本历史

版本	修订时间	修订内容	文件标识
1.0	2022.5	初版编制	OpenTSN3.4

目录

1. 概述

本手册主要从 TTE 同步控制软件的运行环境、文件目录和操作步骤三个部分对 TSNLight3.4 的使用进行说明。

2. 运行环境

本文档是介绍基于 OpenSync 实现时间触发以太网 TTE 中的时钟同步协议 AS6802，文档分为缩略语，项目概述以及总体设计三个部分。

基于 OpenSync 的时钟同步控制软件的运行环境包括两部分：硬件环境和软件环境。硬件环境指对 OpenTSN 硬件工程版本提出要求，软件环境指对操作系统和依赖库提出要求。

2.1. 软件环境

TTE 时钟同步控制软件的运行环境与 TSNLight3.4 的相同，Linux 操作系统，且依赖以下库：

（1） libpcap 库，用于接收数据报文。需要网卡开启混杂模式（开启方式参考附录 3）；

（2） libnet 库，用于发送报文；

（3） libxml2 库，用于进行 xml 文件解析。要求 libxml2 库支持连续解析多个 xml 文本，建议使用 OpenTSN 项目组所提供的 libxml2 库压缩包进行安装和使用。

因此，运行 TTE 时钟同步控制程序的控制主机，需要安装 Linux 操作系统或 Linux 操作系统虚拟机，且还要安装 libnet，libpcap 和 libxml2 库，网卡开启混杂模式。

关于 libpcap 和 libxml2 库的安装教程，请参考《TSN 网络控制器使用手册》附录 2 所述；Linux 操作系统虚拟机的安装和虚拟机如何开启混杂模式，请参考该文档附录 3 所述。

2.2. 硬件环境

硬件环境基于 OpenTSN3.4 版本的硬件。

3. 文件目录

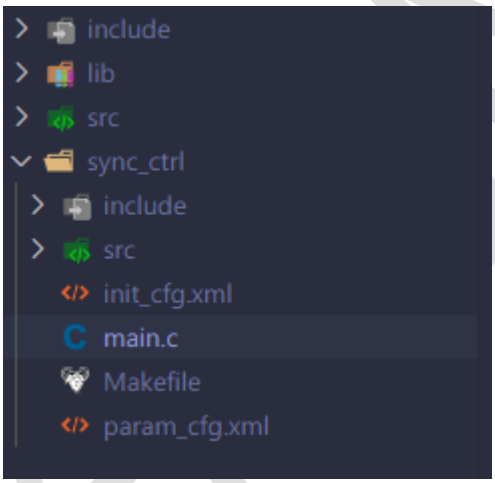

图 3-1 文件目录

一级目录包含四个文件夹，其中 include, src 是 OpenSync 编程接口的库源文件，库文件编译完成后静态库文件在 lib 文件夹中。

sync_ctrl 文件夹中是 TTE 时钟同步控制软件的文件夹，该目录下中的 include 是需要的头文件、src 是 tte 同步逻辑处理函数的源文件。此外，还包含两个 xml 配置文件、main 源文件和 makefile。其中，init_cfg.xml 是用于初始化配置设备的信息，例如时钟同步角色，设备的 MID 等，具体的内容见附录 A；param_cfg.xml 用于配置时钟同步过程中的参数，例如集成周期、最大传输延时等，具体的内容见附录 B。

4. 操作步骤

在 TSN 网络控制器运行之后，完成了网络组网，各个报文通路能够正常的接收和发送报文。然后便可以编译、运行时钟同步控制软件。

4.1. 编译步骤

（1）首先，编译 OpenSync 静态库文件。

进入 tte_sync/src，使用 makefile 进行编译 生成静态库文件 libopensync.a

```
→  src git:(master) make
gcc -g -o pkt_rec_api.o -c pkt_rec_api.c -lpcap -lnet -lsim -I ../include
gcc -g -o pkt_snd_api.o -c pkt_snd_api.c -lpcap -lnet -lsim -I ../include
gcc -g -o shadow_clock.o -c shadow_clock.c
gcc -g -o tools.o -c tools.c -lpcap -lnet -lsim -I ../include
gcc -g -o clock_config.o -c clock_config.c
gcc -g -o header_generate.o -c header_generate.c
gcc -g -o header_parse.o -c header_parse.c
ar -rc ../lib/libopensync.a pkt_snd_api.o pkt_rec_api.o shadow_clock.o to→
```

图 4-1 编译 opensync 静态库

（2）然后，编译 tte 时钟同步控制软件文件。

进入 tte_sync/sync_ctrl，使用 makefile 进行编译，生成可执行文件 sync_ctrl.

```
→  sync_ctrl git:(master) ✗ make
gcc -g -fno-stack-protector -o ./src/basic_func.o -c ./src/basic_func.c
gcc -g -fno-stack-protector -o ./src/cm_pkt_proc.o -c ./src/cm_pkt_proc.c -L
 ../lib -lopensync -lpcap -lnet -lsim -lxml2 -I ./include -I ../include
gcc -g -fno-stack-protector -o ./src/cm_timeout_handle.o -c ./src/cm_timeout
_handle.c -L ../lib -lopensync -lpcap -lnet -lsim -lxml2 -I ./include -I ../
include
gcc -g -fno-stack-protector -o ./src/pcf_pkt_generate.o -c ./src/pcf_pkt_gen
erate.c
gcc -g -fno-stack-protector -o ./src/sm_pkt_proc.o -c ./src/sm_pkt_proc.c  -
L ../lib -lopensync -lpcap -lnet -lsim -lxml2 -I ./include -I ../include
gcc -g -fno-stack-protector -o ./src/sm_timeout_handle.o -c ./src/sm_timeout
_handle.c -L ../lib -lopensync -lpcap -lnet -lsim -lxml2 -I ./include -I ../
include
gcc -g -fno-stack-protector -o ./src/timerlist.o -c ./src/timerlist.c
gcc -g -fno-stack-protector -o ./src/pkt_process.o -c ./src/pkt_process.c
ar -rc ./src/libttesync.a ./src/basic_func.o ./src/cm_pkt_proc.o ./src/cm_ti
meout_handle.o ./src/pcf_pkt_generate.o \
./src/sm_pkt_proc.o ./src/sm_timeout_handle.o ./src/timerlist.o ./src/pkt_pr
ocess.o
gcc -g -fno-stack-protector -o sync_ctrl main.c ./src/libttesync.a -L ./src
-lttesync -L ../lib -lopensync -lpcap -lnet -lsim -lxml2 -I ./include -I ../
include
→  sync_ctrl git:(master) ✗ ls
Makefile  include  init_cfg.xml  main.c  param_cfg.xml  src  sync_ctrl
→  sync_ctrl git:(master) ✗
```

4.2. 运行步骤

编译完成之后，修改 init_cfg.xml 中的 net_interface 为程序运行主机的接口，该接口可以使用 ifconfig 指令查看。

```
→  sync_ctrl git:(master) ✗ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.26.184.99  netmask 255.255.240.0  broadcast 172.26.191.255
        inet6 fe80::215:5dff:feb4:8450  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:b4:84:50  txqueuelen 1000  (Ethernet)
        RX packets 483  bytes 175276 (175.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 262  bytes 41450 (41.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 2462  bytes 26522259 (26.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2462  bytes 26522259 (26.5 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

→  sync_ctrl git:(master) ✗
```

使用管理员权限运行控制程序

```
→  sync_ctrl git:(master) ✗ sudo ./sync_ctrl
```

附录A. 初始化配置 init_cfg.xml

首先是网络中同步设备的数量，控制程序运行主机的网络接口名称。sync ctrl 标签下是描述控制程序的 MID 和组播的地址。device 标签下是描述设备的 ID， 时钟同步角色，MID 以及该节点在同步过程中的静态延时。

```
<?xml version="1.0" encoding="utf-8"?>
<init_cfg>
    <device_num>6</device_num>
    <net_interface>enp44s0</net_interface>
    <sync_ctrl> <!-- 控制程序运行的设备地址 -->
        <mid>6</mid>
        <multi_mac>9</multi_mac>
    </sync_ctrl>

    <device>
        <id>0</id>
        <clock_role>17</clock_role>          <!-- CM = 0x11 ; SM = 0x12 -->
        <mid>0</mid>
        <static_delay>0</static_delay>
    </device>

    <device>
        <id>1</id>
        <clock_role>18</clock_role>          <!-- CM = 0x11 ; SM = 0x12 -->
        <mid>1</mid>
```

```
        <static_delay>800</static_delay>
    </device>

    <device>
        <id>2</id>
        <clock_role>18</clock_role>          <!-- CM = 0x11 ; SM
= 0x12 -->
        <mid>2</mid>
        <static_delay>800</static_delay>
    </device>

    <device>
        <id>3</id>
        <clock_role>18</clock_role>          <!-- CM = 0x11 ; SM
= 0x12 -->
        <mid>3</mid>
        <static_delay>800</static_delay>
    </device>

    <device>
        <id>4</id>
        <clock_role>18</clock_role>          <!-- CM = 0x11 ; SM
= 0x12 -->
        <mid>4</mid>
        <static_delay>1500</static_delay>
    </device>

    <device>
```

```
            <id>5</id>
            <clock_role>18</clock_role>              <!-- CM = 0x11 ; SM
= 0x12 -->
            <mid>5</mid>
            <static_delay>1500</static_delay>
      </device>

</init_cfg>
```

附录B. 同步参数配置 param_cfg.xml

参数配置是按照纳秒为单位的十进制数进行配置。

参数的含义见《基于 OpenSync 的 TTE 时钟同步控制软件设计文档》，或者参照 SAE AS6802 标准。

```
<?xml version="1.0" encoding="utf-8"?>
<param_cfg>
      <global_param>
<integration_cycle_duration>100000000</integration_cycle_duration>
<max_transmission_delay>20000000</max_transmission_delay>
<max_integration_cycle>200</max_integration_cycle>
<accuracy>400</accuracy>
      </global_param>

      <cm_param>
            <cm_listen_timeout>100000000</cm_listen_timeout>
<cm_ca_enable_timeout>100000000</cm_ca_enable_timeout>
<cm_wait_4_in_timeout>100000000</cm_wait_4_in_timeout>
            <cm_restart_timeout>100000000</cm_restart_timeout>
            <cm_dispatch_delay>40000</cm_dispatch_delay>
```

```
<cm_caculation_overhead>20000</cm_caculation_overhead>

<cm_integrate_to_sync_thrld>8</cm_integrate_to_sync_thrld>
        <cm_unsync_to_sync_thrld>1</cm_unsync_to_sync_thrld>
        <cm_sync_threshold_sync>1</cm_sync_threshold_sync>
        <cm_sync_threshold_async>2</cm_sync_threshold_async>

<cm_sync_listen_timeout>100000000</cm_sync_listen_timeout>
        <cm_ca_listen_timeout>80000000</cm_ca_listen_timeout>

<cm_wait4in_listen_timeout>80000000</cm_wait4in_listen_timeout>
    </cm_param>

    <sm_param>
        <sm_listen_timeout>200000000</sm_listen_timeout>
        <sm_coldstart_timeout>500000000</sm_coldstart_timeout>
        <sm_restart_timeout>200000000</sm_restart_timeout>
        <cs_offset>40000000</cs_offset>
        <ca_offset>40000000</ca_offset>
        <ca_receive_timeout>200000000</ca_receive_timeout>
        <ca_acceptance_window>200</ca_acceptance_window>

<sm_integrate_to_sync_thrld>1</sm_integrate_to_sync_thrld>
        <sm_unsync_to_sync_thrld>1</sm_unsync_to_sync_thrld>

<sm_unsync_to_tentative_thrld>2</sm_unsync_to_tentative_thrld>
```

```
<sm_tentative_sync_threshold_sync>1</sm_tentative_sync_threshold_
sync>

<sm_tentative_to_sync_thrld>1</sm_tentative_to_sync_thrld>
        <sm_sync_threshold_sync>1</sm_sync_threshold_sync>
        <num_stable_cycles>8</num_stable_cycles>
        <num_unstable_cycles>2</num_unstable_cycles>
        <sm_stable_threshold_sync>1</sm_stable_threshold_sync>
    </sm_param>
</param_cfg>
```