

# PTP Scenarios based on S32G PFE

本文对S32G中PTP基于PFE几种典型场景进行demo的搭建并进行说明。

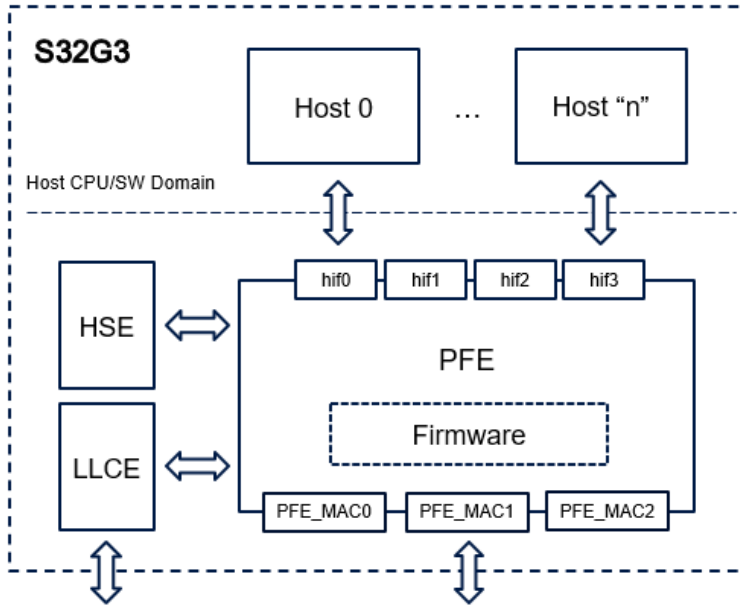
历史	说明	作者
V1	● 创建本文	● Jerry Huang

## 目录

1	PFE中PTP的支持 .....	2
1.1	参考资料.....	2
1.2	版本匹配说明.....	2
1.3	需要的硬件 .....	3
2	需要的Image文件.....	3
3	PTP时钟 .....	3
4	S32G（PFE）中PTP工作场景 .....	4
4.1	S32G399ARDB3工作为Transparent Clock .....	5
4.2	S32G399ARDB3工作为Boundary Clock .....	10
4.3	S32G399ARDB3工作为Ordinary Clock.....	13
4.4	PFE eMAC配置 .....	14

# 1 PFE 中 PTP 的支持

PFE 是 S32G 系列芯片一个用于网络通信的模块，通过三个 EMAC 接口和外部网络相连，同时有四个 HIF，可以和内部不同 host 进行数据交换，比如 A 核和 M 核可以通过某一个 EMAC 同时和外部建立起网络连接。



PFE 支持 802.11AS-Rev 和 IEEE 1588 精确时钟同步协议，支持单步时钟模式和两步时钟模式。

## 1.1 参考资料

序号	资料	说明	如何获取
1	S32G3 BSP39 Linux Release. ● S32G3_LinuxBSP_39.0_User_Manual.pdf ● S32G2_LinuxBSP_39.0_User_Manual.pdf ● binaries_auto_linux_bsp39.0_s32g2_pfe.tgz binaries_auto_linux_bsp39.0_s32g3_pfe_and_s32g3_scprt.tgz	根据文档构建 Yocto Linux 编译环境	<a href="http://www.nxp.com/s32g">www.nxp.com/s32g</a> 通过 bundle 下载 Linux 相关文档

## 1.2 版本匹配说明

在本文档中，使用标准的预编译好的支持 PFE 的 image，使用默认的文件系统环境。

## 1.3 需要的硬件

在本文档的使用场景中，需要用到一下硬件设备：

- 1) 1 台 PC 带有支持硬件 Timestamp 的网卡
- 2) 2 块 S32G399ARDB3 或者 2 块 S32G274ARDB2，或者各一块

## 2 需要的 Image 文件

本文档使用 BSP39 预编译好的 image，所以不需要再单独编译需要的组件。

使用了 S32G274ARDB2 和 S32G399ARDB3 两块开发板，所以将下载好的文件解压后，分别烧写在两块不同的 TF 卡，分别启动 S32G274ARDB2 和 S32G399ARDB3。

使用到的两个 SD card 文件：

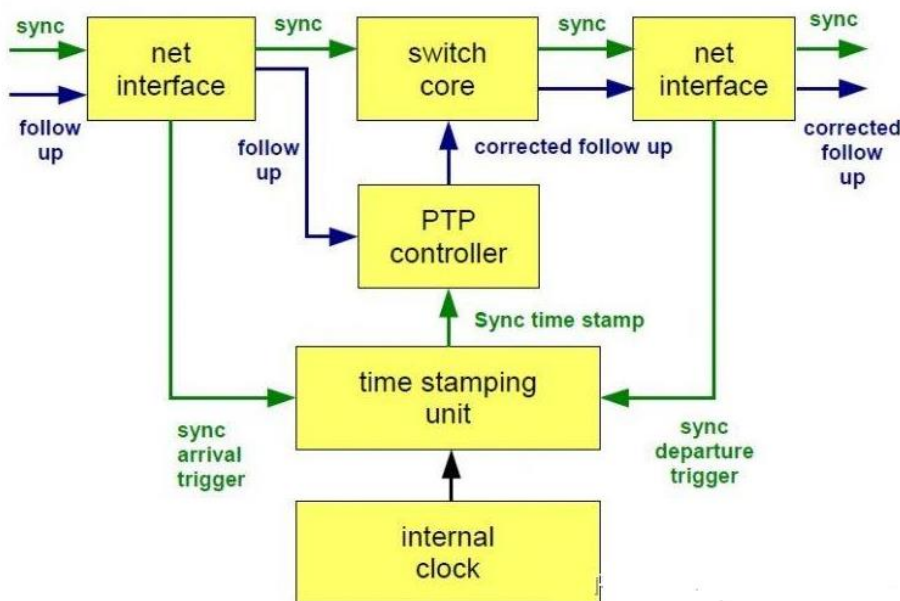
fsl-image-auto-s32g399ardb3.sdcard

fsl-image-auto-s32g274ardb2.sdcard

## 3 PTP 时钟

对于支持 PTP 的终端设备来说，可以工作为 PTP master，或者工作为 PTP slave。

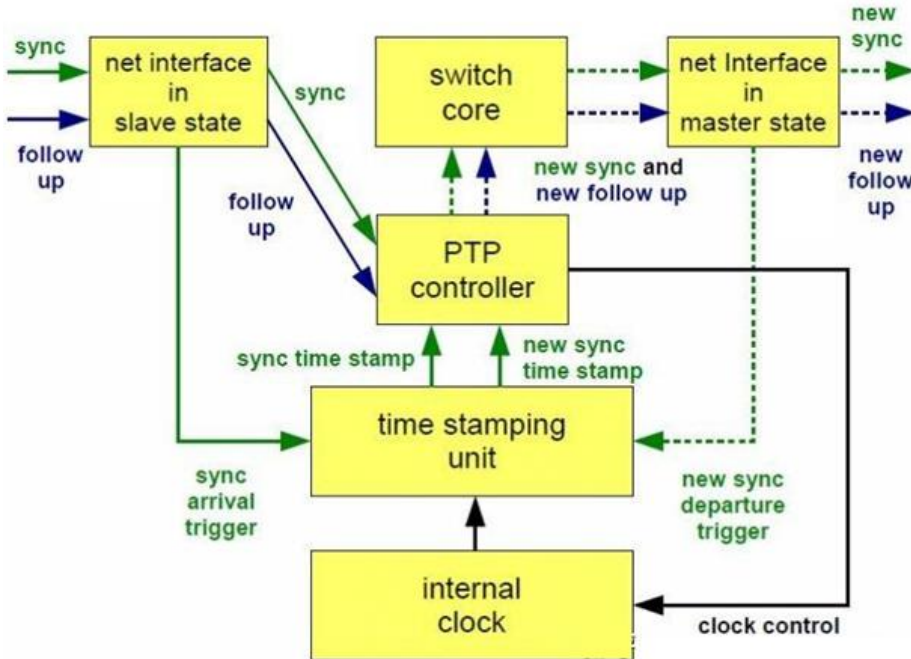
对于支持 PTP Transparent Clock (TC) 的交换机来说，其工作流程如下图所以 (E2E)。



PTP Based on S32G PFE

- 1) 从 grand master 来的 sync 报文到达时，交换机获取到达的 timestamp，当 sync 报文从交换机发出时，交换机也将获取发出的 timestamp，根据进入和离开的 timestamp，计算出 sync 报文在交换机中的驻留时间（Resident time）。
- 2) 当交换机接收到 follow up 报文时，将驻留时间修改到 follow up 的时间域中，然后把 follow up 报文（corrected follow up）重新发出去。

交换机工作为边界时钟（Boundary Clock, BC），如下图所示：



对于工作为 BC 的交换机，其一个端口将为 slave，其余的 1 个或多个端口为 master。Slave 端口从 grand master 同步时钟到本地；而 master 端口将本地时钟发布到相连的其余 PTP 设备。

## 4 S32G（PFE）中 PTP 工作场景

S32G 中的 PFE 可以工作为交换机模式。但是 PFE 中 EMAC 仅仅支持硬件 timestamp，不支持 Transparent Clock，如果要支持 Transparent Clock，需要 S32G CPU 的参与。

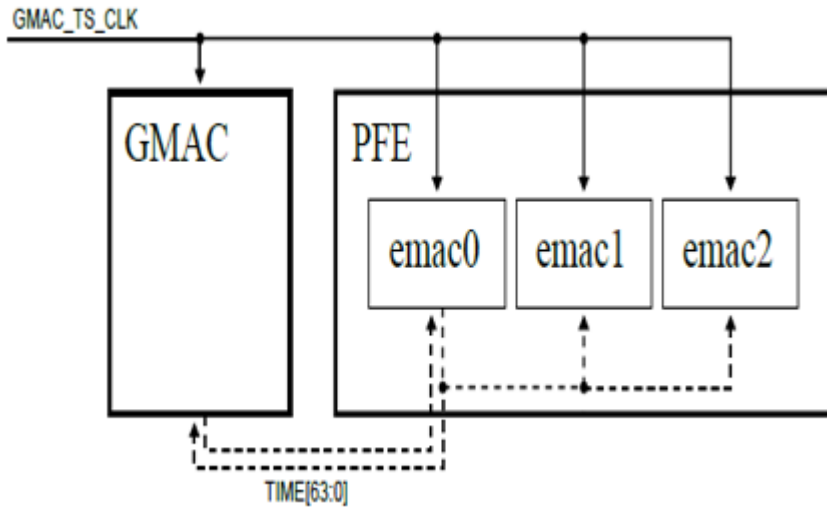
本文档详细描述如何使 S32G 工作为 PTP Boundary Clock 和 PTP Transparent Clock。

由于 PTP 终端节点（master, slave）是很标准的配置，本文中只做简单的描述。

在 S32G 中，GMAC，PFE eMAC0，eMAC1 和 eMAC2 的 Time stamp 使用同一个时钟源。对于各个 MAC 来说，可以使用自己内部的 time stamp，也可以统一的外部 time stamp，如 eMAC0 可以使用 GMAC time stamp，同时 GMAC，eMAC1 和 eMAC2 可以使用 eMAC0 的 time stamp。

### PTP Based on S32G PFE

在本文中，我们统一使用 eMAC0 的时钟，因而需要将 eMAC1 和 eMAC2 的 ESTI 使能，下图为 S32G 中 Time stamp 时钟源：



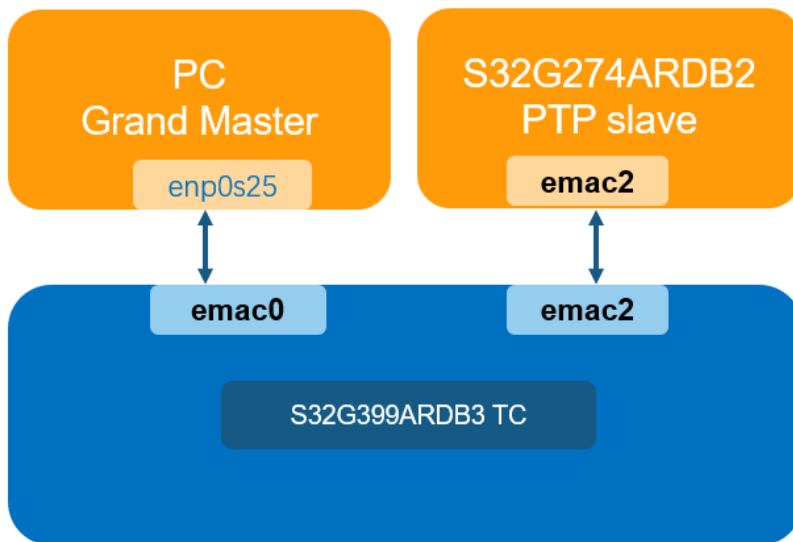
#### 4.1 S32G399ARDB3 工作为 Transparent Clock

如下图所示，S32G399ARDB3（PFE）工作为PTP Transparent Clock，S32G274ARDB2工作为 PTP slave 节点，一台PC工作为PTP grand master clock。

S32G399ARDB3使用eMAC0和PC网口相连，同时eMAC2和S32G274ARDB2的eMAC2相连（也可以使用其余的EMAC接口）。

注意，当S32G399ARDB3工作为Transparent Clock时，S32G399ARDB3的本地Timestamp不会更新，只有作为PTP slave的S32G274ARDB2的Timestamp会更新。

PFE使用默认配置



下面为执行命令：

**PTP Based on S32G PFE**

- 1) Execute following command to make PC as grandmaster clock
  - `ptp4l -i enp0s25 -f master.cfg -m`
- 2) Execute following command to make S32G399ARDB3 as relay
  - `#set pfe_mac1 ESTI bit`  
`devmem2 0x460a4b00 w 0x113f03`
  - `#set pfe_mac2 ESTI bit`  
`devmem2 0x460a8b00 w 0x113f03`
  - `ptp4l -i pfe2 -i pfe0 -f tc.cfg -m`
- 3) Execute following command to make S32G299ARDB2 as slave
  - `ptp4l -i pfe2 -f slave.cfg -m`
  - `phc2sys -s pfe2 -c CLOCK_REALTIME -O0 &`
  - `phc_ctl pfe2 get`

在此demo中，使用P2P TC时钟，并使用P2P 2步进行通信。

#### Master.cfg内容:

```
[global]
gmCapable          1
priority1          248
priority2          248
logSyncInterval    -3
syncReceiptTimeout 3
neighborPropDelayThresh 800
min_neighbor_prop_delay -20000000
assume_two_step    1
path_trace_enabled 1
follow_up_info     1
transportSpecific  0x1
ptp_dst_mac        01:80:C2:00:00:0E
network_transport  L2
delay_mechanism     P2P
BMCA                noop
serverOnly         1
inhibit_announce   1
asCapable           true
inhibit_delay_req  1
```

#### Slave.cfg内容:

### PTP Based on S32G PFE

```
[global]
gmCapable          1
priority1          248
priority2          248
logSyncInterval    -3
syncReceiptTimeout 3
neighborPropDelayThresh 800
min_neighbor_prop_delay -20000000
assume_two_step    1
path_trace_enabled 1
follow_up_info     1
transportSpecific  0x1
ptp_dst_mac        01:80:C2:00:00:0E
network_transport  L2
delay_mechanism    P2P
BMCA               noop
clientOnly         1
inhibit_announce  1
asCapable          true
ignore_source_id   1
step_threshold     1
operLogSyncInterval 0
operLogPdelayReqInterval 2
msg_interval_request 1
servo_offset_threshold 30
servo_num_offset_values 10
```

Tc.cfg内容:

```
[global]
priority1          254
free_running      1
freq_est_interval 3
summary_interval  1
clock_type        P2P_TC
network_transport L2
delay_mechanism   P2P
assume_two_step   1
path_trace_enabled 1
follow_up_info    1
transportSpecific 0x1
ptp_dst_mac       01:80:C2:00:00:0E
BMCA              noop
clientOnly        1
inhibit_announce  1
asCapable         true
ignore_source_id  1
step_threshold    1
operLogSyncInterval 0
operLogPdelayReqInterval 2
msg_interval_request 1
servo_offset_threshold 30
servo_num_offset_values 10
```

运行时的log如下:



PC Master:

```
huangcm@Ares-hcm:~/images/ptp-tc$ sudo ptp4l -i enp0s25 -f master.cfg -m
ptp4l[1819598.484]: selected /dev/ptp0 as PTP clock
ptp4l[1819598.524]: port 1 (enp0s25): INITIALIZING to MASTER on INIT_COMPLETE
ptp4l[1819598.524]: port 0 (/var/run/ptp4l): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1819598.524]: port 0 (/var/run/ptp4lro): INITIALIZING to LISTENING on INIT_COMPLETE
```

S32G399ARDB3 TC:

```
root@s32g399ardb3:~# ptp4l -i pfe2 -i pfe0 -f tc.cfg -m
ptp4l[1966.648]: port 1 (pfe2): INITIALIZING to SLAVE on INIT_COMPLETE
ptp4l[1966.680]: port 2 (pfe0): INITIALIZING to SLAVE on INIT_COMPLETE
ptp4l[1966.680]: port 0 (/var/run/ptp4l): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1966.680]: port 0 (/var/run/ptp4lro): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1973.581]: selected local clock 00049f.ffff.beef02 as best master
ptp4l[1981.605]: master offset      296352 s0 freq   +5463 path delay    10882
ptp4l[1989.611]: master offset      339687 s0 freq   +5472 path delay    11353
```

S32G274ARDB2 Slave:

```
root@s32g274ardb2:~# ptp4l -i pfe2 -f slave.cfg -m
ptp4l[1644.474]: selected /dev/ptp2 as PTP clock
ptp4l[1644.508]: port 1 (pfe2): INITIALIZING to SLAVE on INIT_COMPLETE
ptp4l[1644.508]: port 0 (/var/run/ptp4l): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1644.508]: port 0 (/var/run/ptp4lro): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1646.411]: rms 5300 max 6989 freq -1161 +/- 3608 delay 635 +/- 0
ptp4l[1647.412]: rms 893 max 1338 freq +1668 +/- 1257 delay 632 +/- 0
ptp4l[1648.413]: rms 1502 max 1591 freq +4571 +/- 450 delay 630 +/- 0
ptp4l[1649.414]: rms 994 max 1358 freq +5188 +/- 85 delay 627 +/- 0
ptp4l[1650.414]: rms 278 max 512 freq +4687 +/- 250 delay 625 +/- 0
ptp4l[1651.415]: rms 152 max 257 freq +4272 +/- 153 delay 626 +/- 0
ptp4l[1652.416]: rms 80 max 133 freq +4308 +/- 104 delay 625 +/- 0
```

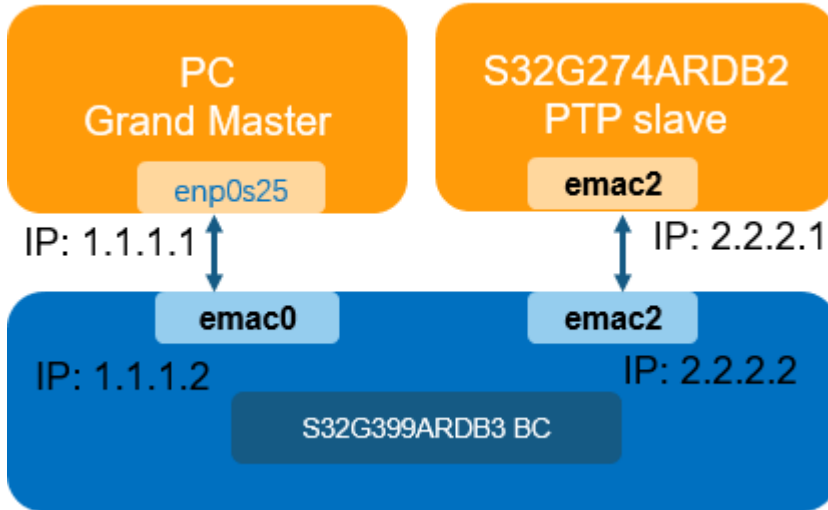
Slave端输出结果说明:

```
pr_info("rms %4.0f max %4.0f "
        "freq %+6.0f +/- %3.0f "
        "delay %5.0f +/- %3.0f",
        offset_stats.rms, offset_stats.max_abs,
        freq_stats.mean, freq_stats.stddev,
        delay_stats.mean, delay_stats.stddev);
stats->mean = old_mean + (value - old_mean) / stats->num;
stats->sum_sqr += value * value;
stats->sum_diff_sqr += (value - old_mean) * (value - stats->mean);
result->max_abs = stats->max > -stats->min ? stats->max : -stats->min;
result->mean = stats->mean;
result->rms = sqrt(stats->sum_sqr / stats->num);
result->stddev = sqrt(stats->sum_diff_sqr / stats->num);
ratio = tmv_dbl(tmv_sub(origin, f->origin1)) /
        | tmv_dbl(tmv_sub(ingress, f->ingress1));
freq = (1.0 - ratio) * 1e9;
```

## PTP Based on S32G PFE

rms: offset的均方根  
max: offset最大绝对值  
freq: 根据后续的值和初始值的比例计算而来

## 4.2 S32G399ARDB3 工作为 Boundary Clock



如上图所示，S32G399ARDB3（PFE）工作为PTP Boundary Clock，S32G274ARDB2工作为PTP slave节点，一台PC工作为PTP grand master clock。

S32G399ARDB3使用eMAC0和PC网口相连，同时eMAC2和S32G274ARDB2的eMAC2相连（也可以使用其余的EMAC接口）。

注意，当S32G399ARDB3工作为Boundary Clock时，S32G399ARDB3的本地Timestamp会更新，同时作为PTP slave的S32G274ARDB2的Timestamp也会更新。

PTP可以运行在L2，UDP4和UDP6，使用参数network\_transport进行设置，在下面的命令中使用默认的UDP4。

下面为执行命令：

- 1) Execute following command to make PC as grandmaster clock
  - `ifconfig enp0s25 1.1.1.1`
  - `ptp4l -i enp0s25 --serverOnly 1 -m`
- 2) Execute following command to make S32G399ARDB3 as relay
  - `#set pfe_mac1 ESTI bit`  
`devmem2 0x460a4b00 w 0x113f03`
  - `#set pfe_mac2 ESTI bit`  
`devmem2 0x460a8b00 w 0x113f03`
  - `#set IP address`  
`ifconfig pfe0 1.1.1.2`  
`ifconfig pfe2 2.2.2.2`
  - `ptp4l -f bc.cfg -m`
- 3) Execute following command to make S32G299ARDB2 as slave
  - `ifconfig pfe2 2.2.2.1`
  - `ptp4l -i pfe2 --clientOnly 1 --priority1 248 -m`
  - `phc2sys -s pfe2 -c CLOCK_REALTIME -O0 &`
  - `phc_ctl pfe2 get`

#### Bc.cfg内容:

```
[global]
tx_timestamp_timeout 10
logMinPdelayReqInterval -3
logSyncInterval -3
twoStepFlag 1
summary_interval 0
priority1 200
priority2 200
[pfe0]
boundary_clock_jbod 1
[pfe2]
boundary_clock_jbod 1
```

运行log如下:

```

PC master:
huangcm@Ares-hcm:~/images/ptp-tc/bc$ sudo ptp4l -i enp0s25 --serverOnly 1 -m
ptp4l[1818109.375]: selected /dev/ptp0 as PTP clock
ptp4l[1818109.375]: port 1 (enp0s25): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1818109.376]: port 0 (/var/run/ptp4l): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1818109.376]: port 0 (/var/run/ptp4lro): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1818115.912]: port 1 (enp0s25): LISTENING to MASTER on
ANNOUNCE_RECEIPT_TIMEOUT_EXPIRES
ptp4l[1818115.912]: selected local clock 64006a.ffffe.59874a as best master
ptp4l[1818115.912]: port 1 (enp0s25): assuming the grand master role

S32G399ARDB3 BC:
root@s32g399ardb3:~# ptp4l -f bc.cfg -m
ptp4l[1123.891]: selected /dev/ptp0 as PTP clock
ptp4l[1123.900]: port 2 (pfe2): just a bunch of devices
ptp4l[1123.901]: port 1 (pfe0): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1123.902]: port 2 (pfe2): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1123.902]: port 0 (/var/run/ptp4l): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1123.902]: port 0 (/var/run/ptp4lro): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1125.668]: port 1 (pfe0): new foreign master 64006a.ffffe.59874a-1
ptp4l[1127.668]: selected best master clock 64006a.ffffe.59874a
ptp4l[1127.668]: port 1 (pfe0): LISTENING to UNCALIBRATED on RS_SLAVE
ptp4l[1127.668]: Switched to /dev/ptp0 as PTP clock
ptp4l[1129.692]: master offset      94603 s0 freq  -11737 path delay      20470
ptp4l[1129.712]: master offset      94788 s0 freq  -11737 path delay      20470
ptp4l[1130.693]: master offset      98314 s1 freq   -8026 path delay      20470
ptp4l[1130.712]: master offset         45 s2 freq   -7981 path delay      20470
ptp4l[1130.712]: port 1 (pfe0): UNCALIBRATED to SLAVE on MASTER_CLOCK_SELECTED
ptp4l[1131.691]: port 2 (pfe2): LISTENING to MASTER on ANNOUNCE_RECEIPT_TIMEOUT_EXPIRES
ptp4l[1131.692]: master offset         51 s2 freq   -7962 path delay      20470
ptp4l[1131.712]: master offset      -33 s2 freq   -8031 path delay      20470

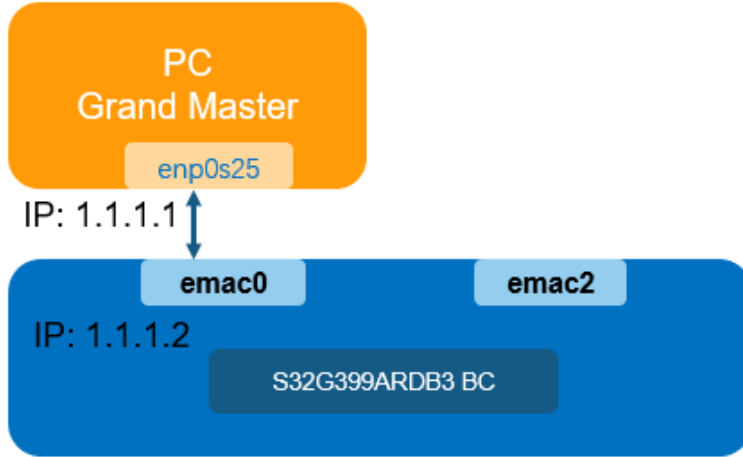
S32G274ARDB2 Slave:
root@s32g274ardb2:~# ptp4l -i pfe2 --clientOnly 1 --priority1 248 -m
ptp4l[254.488]: selected /dev/ptp2 as PTP clock
ptp4l[254.497]: port 1 (pfe2): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[254.498]: port 0 (/var/run/ptp4l): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[254.498]: port 0 (/var/run/ptp4lro): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[255.735]: port 1 (pfe2): new foreign master 00049f.ffffe.beef00-2
ptp4l[259.372]: selected best master clock 64006a.ffffe.59874a
ptp4l[259.372]: port 1 (pfe2): LISTENING to UNCALIBRATED on RS_SLAVE
ptp4l[261.614]: port 1 (pfe2): UNCALIBRATED to SLAVE on MASTER_CLOCK_SELECTED
ptp4l[262.094]: rms 1046546834270655232 max 1709003823925451008 freq  +394 +/- 586 delay
168 +/-0
ptp4l[263.061]: rms 2123 max 2478 freq  +4208 +/- 1054
ptp4l[263.998]: rms 2254 max 2495 freq  +6829 +/- 378 delay  176 +/-  2
ptp4l[264.946]: rms  993 max 1536 freq  +6717 +/- 212 delay  198 +/-  7
ptp4l[265.900]: rms  928 max  991 freq  +7557 +/- 382

```

## PTP Based on S32G PFE

### 4.3 S32G399ARDB3 工作为 Ordinary Clock

下图为 S32G399ARDB3 工作为 ordinary clock 模式，将同步 PC 的授时。



下面为执行命令：

- 1) Execute following command to make PC as grandmaster clock
  - `ptp4l -i enp0s25 --serverOnly 1 --network_transport L2 -m`
- 4) Execute following command to make S32G399ARDB3 as slave
  - `ptp4l -i pfe0 --clientOnly 1 --priority1 248 --network_transport L2 -m`
  - `phc2sys -s pfe0 -c CLOCK_REALTIME -O0 &`
  - `phc_ctl pfe0 get`

运行log如下：

```

PC master:
huangcm@Ares-hcm:~/images/ptp-tc/bc$ sudo ptp4l -i enp0s25 --serverOnly 1
--network_transport L2 -m
ptp4l[1911446.149]: selected /dev/ptp0 as PTP clock
ptp4l[1911446.184]: port 1 (enp0s25): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1911446.184]: port 0 (/var/run/ptp4l): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1911446.184]: port 0 (/var/run/ptp4lro): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[1911453.000]: port 1 (enp0s25): LISTENING to MASTER on
ANNOUNCE_RECEIPT_TIMEOUT_EXPIRES
ptp4l[1911453.000]: selected local clock 64006a.ffff.59874a as best master
ptp4l[1911453.000]: port 1 (enp0s25): assuming the grand master role

S32G399ARDB3 Slave:
root@s32g399ardb3:~# ptp4l -i pfe0 --clientOnly 1 --priority1 248 --network_transport L2 -m
ptp4l[18502.859]: selected /dev/ptp0 as PTP clock
ptp4l[18502.900]: port 1 (pfe0): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[18502.900]: port 0 (/var/run/ptp4l): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[18502.900]: port 0 (/var/run/ptp4lro): INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[18504.759]: port 1 (pfe0): new foreign master 64006a.ffff.59874a-1
ptp4l[18508.759]: selected best master clock 64006a.ffff.59874a
ptp4l[18508.759]: port 1 (pfe0): LISTENING to UNCALIBRATED on RS_SLAVE
ptp4l[18511.393]: master offset      -3133 s0 freq   -8085 path delay   18900
ptp4l[18512.393]: master offset      -3152 s2 freq   -8104 path delay   18900
ptp4l[18512.393]: port 1 (pfe0): UNCALIBRATED to SLAVE on MASTER_CLOCK_SELECTED
ptp4l[18513.393]: master offset      -3641 s2 freq  -11745 path delay   19411
ptp4l[18514.393]: master offset         402 s2 freq   -8794 path delay   18900

```

## 4.4 PFE eMAC 配置

```
root@s32g399ardb3:~# libfci_cli phyif-print -i emac0
```

DISCLAIMER: This is a DEMO application. It is not part of the production code deliverables.

```

0: emac0
  <ENABLED>
  <promisc:OFF, mode:DEFAULT, block-state:NORMAL>
  <vlan-conf:OFF, ptp-conf:OFF, ptp-promisc:OFF, q-in-q:ON>
  <discard-if-ttl-below-2:OFF>
  ingress: 0 egress: 355 discarded: 0 malformed: 0
  ptp-mgmt-if: ---
  MAC:
    00:04:9f:be:ef:00
    33:33:00:00:00:01
    01:00:5e:00:00:01
    01:80:c2:00:00:00

```

**PTP Based on S32G PFE**

01:80:c2:00:00:03

01:80:c2:00:00:0e

33:33:ff:be:ef:00

33:33:00:01:00:03

mirrors:

rxmirr0: ---

rxmirr1: ---

txmirr0: ---

txmirr1: ---

logical interfaces:

0: pfe0

<ENABLED>

<promisc:ON, match-mode:AND, discard-on-match:OFF, loopback:OFF>

accepted: 0 rejected: 0 discarded: 0 processed: 0

egress: hif0

match-rules: ---

Command successfully executed.

root@s32g399ardb3:~# libfci\_cli phyif-print -i emac2

DISCLAIMER: This is a DEMO application. It is not part of the production code deliverables.

2: emac2

<ENABLED>

<promisc:OFF, mode:DEFAULT, block-state:NORMAL>

<vlan-conf:OFF, ptp-conf:OFF, ptp-promisc:OFF, q-in-q:ON>

<discard-if-ttl-below-2:OFF>

ingress: 0 egress: 0 discarded: 0 malformed: 0

ptp-mgmt-if: ---

MAC:

00:04:9f:be:ef:02

33:33:00:00:00:01

01:00:5e:00:00:01

mirrors:

rxmirr0: ---

rxmirr1: ---

txmirr0: ---

**PTP Based on S32G PFE**

```
txmirr1: ---
logical interfaces:
4: pfe2
  <ENABLED>
  <promisc:ON, match-mode:AND, discard-on-match:OFF, loopback:OFF>
  accepted: 0 rejected: 0 discarded: 0 processed: 0
  egress: hif2
  match-rules: ---
Command successfully executed.
```



