ToD Architecture Time Synchronization Test

# Test Objective and Expected Results

Using three AX88279 devices to simulate the GNSS-based time synchronization process:

**Device\_1** simulates the GNSS (ToD Master), providing the standard time (ToD and PPS).

**Device\_2** simulates receiving the ToD and PPS from the GNSS, using GNSS as the upstream time source and acting as the PTP Grandmaster to provide time for all nodes in the network.

**Device\_3** simulates a downstream device that synchronizes with the PTP Grandmaster and aligns its time with the GNSS reference.

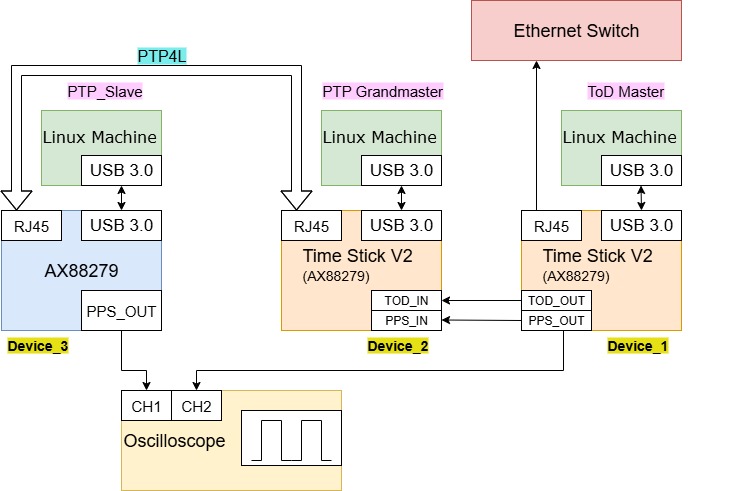
# Test Equipment

Linux Platform \* 3

AX88279 \*3

Ethernet switch \*1

Oscilloscope \*1



# Environment Configuration

## 3.1 ToD Master setting

Build The Driver

1. Open the asix\_usb\_nic\_linux\_driver\_v3.5.16.
2. Extract the source code from the compressed file.

**$ tar -xf asix\_usb\_nic\_linux\_driver\_v3.5.16.zip**

1. Edit the Makefile and enable PTP

**ENABLE\_PTP\_FUNC = y** // The PTP\_FUNC flag must be enabled.

1. In a terminal, navigate to the driver source code folder.
2. From within the driver source code folder, build the driver

**$ make**

Running the Driver

1. Install the driver if you’d like to use the modprobe command to mount the driver.

**$ sudo make install**

1. To load the driver by modprobe command

**$ sudo modprobe ax\_usb\_nic**

1. To unload the driver by modprobe command

**$ sudo modprobe -r ax\_usb\_nic**

1. To check driver information

**$ modinfo ax\_usb\_nic**

Remove the driver and cdc\_ncm

**$ sudo modprobe -r ax\_usb\_nic**

**$ sudo rmmod cdc\_mbim**

**$ sudo rmmod cdc\_ncm**

**$ sudo rmmod ax88179\_178a**

Run the driver

**$ sudo modprobe ax\_usb\_nic**

Check Device\_1 interface name

**$ ifconfig**

Configure the DUT as the ToD Master. //these commands must be run as root.

**$ sudo -s**

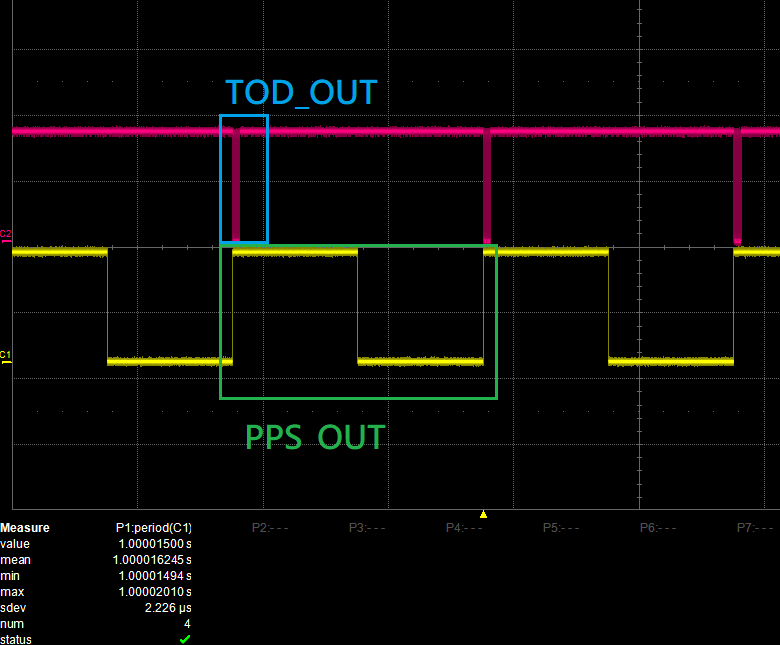
**# echo "ENABLE\_TOD\_MODE 1" > /sys/class/net/[interface name]/device/gpio\_vfs**

**# echo "CONFIG\_TOD\_GPIO 0 7" > /sys/class/net/[interface name]/device/gpio\_vfs**

**# echo "CONFIG\_PPS\_GPIO 0 14" > /sys/class/net/[interface name]/device/gpio\_vfs**

// After the parameters are configured, connect the network cable and verify the network connectivity.

Check output waveform



## 3.2 PTP Grandmaster setting

The driver configuration is the same as in **Section 3.1**.

Configure the DUT as the PTP Grandmaster. //these commands must be run as root.

**$ sudo -s**

**# echo "ENABLE\_TOD\_MODE 0" > /sys/class/net/[interface name]/device/gpio\_vfs**

**# echo "CONFIG\_TOD\_GPIO 1 6" > /sys/class/net/[interface name]/device/gpio\_vfs**

**# echo "CONFIG\_PPS\_GPIO 1 14" > /sys/class/net/[interface name]/device/gpio\_vfs**

// After the parameters are configured, connect the network cable and verify the network connectivity.

The DUT runs PTP4L and configured as the PTP master.

**$ sudo ip addr add 192.168.200.2/24 dev [interface name]**

**$ sudo ip link set [interface name] up**

**$ ip addr show [interface name]**

**// Add an option to specify the interface!**

**$ ping 192.168.200.1**

**$ sudo ptp4l -i [interface name] -m -H --masterOnly 1 --priority1 100**

## 3.3 PTP slave

The driver configuration is the same as in **Section 3.1**.

The DUT runs PTP4L and configured as the PTP Salve.

**$ sudo ip addr add 192.168.200.1/24 dev [interface name]**

**$ sudo ip link set [interface name]up**

**$ ip addr show [interface name]**

**// Add an option to specify the interface!**

**$ ping 192.168.200.2**

**$ sudo ptp4l -i [interface name]-m -H --slaveOnly 1**

Check whether the PPS out of Device\_1 and the PPS out of Device\_3 are synchronized.