



Abstract

This application note describes the process of running the Precision Time Protocol (PTP) as defined in IEEE 1588-2008 on a PC or server using a syn1588® PCIe NIC. It highlights the basic steps required to install and use the syn1588® PCIe NIC as well as references to the syn1588® User Guide (UG001) and other Application Notes.



Prerequisites

- A syn1588® PCIe NIC
- A PC with a free PCIe slot
- The wooden USB-Stick bundled with your syn1588® PCIe NIC containing the syn1588® PTP software and all accompanying drivers and documents within a syn1588® Linux Live System.
- In case the software is to be installed on an existing operating systems rather than using the syn1588® Linux Live System administrator rights are required.

What about the syn1588® Dual NIC

The basic installation procedure for the syn1588® Dual NIC is identical to the one for the syn1588® PCIe NIC. The main difference is that the syn1588® Dual NIC currently just supports Linux OS.

What about the syn1588® PCIe NIC Revision 2.3

Starting with November 2022 the new Revision 2.3 of the syn1588® PCIe NIC is available. Although the card is functionally fully compatible with its predecessor Revision 2.1 there are some differences. Please check the Application Note AN041 “What's new with syn1588® PCIe NIC Revision 2.3” which describes all details.

Basic Installation of the syn1588® PCIe NIC

To set up the syn1588® PCIe NIC please start with the following steps:

- Power off your PC.
- Please be sure to thoroughly ground yourself by means of a grounding strap or by touching a grounded object prior to unpacking the syn1588® PCIe NIC

- Insert the syn1588® PCIe NIC into an empty PCIe slot. Use an appropriate slot that supports PCIe x1 cards.
- Connect the syn1588® PCIe NIC to a network or to another PC using a standard Ethernet cable (or optical fiber in case of the SFP version of the syn1588® PCIe NIC).

Caution

Please note that some mainboards or BIOS versions do not allow to insert a 1-lane PCIe card into an 8-lane or 16-lane slot. Please consult your respective mainboard/BIOS manual in case of troubles.

There are two different ways of continuing: Either use the bootable syn1588® Linux Live System contained on the USB Stick provided with the syn1588® PCIe NIC or you could install the software together with the drivers onto an existing Windows/Linux PC.

Using the syn1588® Linux Live System

The syn1588® Linux Live System starts a dedicated Linux (Ubuntu 20.04, 64 bit) with all drivers and utilities pre-installed. Please make sure, that the PC you are using is able to boot from an USB stick; you might have to check the BIOS settings for this.

Insert the USB stick into your computer and select the appropriate boot option. After the boot procedure has been completed successfully you will see an Ubuntu desktop. There are several icons on the desktop of the syn1588® Linux Live System. Refer to the Application Note (AN038) for more information about the syn1588® Linux Live System.

Installing the driver on your PC

The first step which has to be completed after powering up your PC is the installation of the device driver. Please note, that this step usually needs Administrator-Rights on Windows, or root-rights on Linux respectively.



There are tools to verify the proper operation of the syn1588® PCIe NIC hardware. This allows to deal with systems that behave somewhat different from standard systems. These tools can be found on the provided USB-Stick or other storage medium. If you are able to see an additional network adapter in the system you can proceed.

If you are experiencing problems during installation please double check the support operating system versions as outlined in the document “syn1588_release_OS_support” that is also part of the syn1588 Linux Live System.

Installing the device driver in Linux

Debian/Ubuntu .deb

For a Debian type Linux operating system open a root shell, browse to the Linux driver-directory “/syn1588/sw” on the provided syn1588® USB Stick and execute:

```
> dpkg -i syn1588-dkms_<version>_all.deb
> apt install -f # for installing needed dependencies
```

Note the <version> field holds the actual version of the current driver package.

E.g., for Release v2.17 the full file name is syn1588-dkms_2.17.0_all.deb.

Installing the device driver in Linux

RedHat/CentOS/SUSE .rpm

If your Linux supports RPM packages please use this command sequence. One has to enable the “EPEL” package repository on the RedHat/CentOS PC first prior installing the module.

```
# v7 (yum)
> yum install --enablerepo=extras epel-release
> yum install kmod-syn1588nic*.rpm
# v8 (dnf)
> dnf install epel-release
> dnf install kmod-syn1588nic*.rpm
```

Loading the device driver

After successful installation, the device driver will be loaded after a restart or it can be loaded using the following command:

```
> sudo modprobe syn1588_pci
```

Unloading the device driver manually

If the device driver is updated, the old modules have to be removed from the memory.

```
> sudo modprobe -r syn1588_syncd
> sudo modprobe -r syn1588_pci
> sudo modprobe -r oregano_spi
```

After this step is completed load the new driver manually (as explained above) or restart the system.

To verify if the device driver was successfully installed and loaded check [Chapter Driver Check](#).

Installing the device driver in Windows (NDIS 5.0)

If not already prompted to install a new device, simply browse to the Control Panel and open the Device Manager. Select the new device, which may be listed as “unknown device” and update the drivers. Manually select “syn1588.inf” in the device driver folder “/syn1588/sw/windows_drv/ndis5.0” on the provided syn1588® USB Stick. After successful installation, the syn1588® PCIe NIC should be listed at the section “network cards” in the Windows device manager or System Control. For troubleshooting or further information refer to (UG001) Chapter 4.4.2.

Installing the device driver in Windows (NDIS 6.5 BETA)

!Please read the instructions carefully before you install the driver!

Choosing the right driver

For the syn1588 NIC revision 2.3 there are currently two supported versions of the Windows driver: NDIS 6.3 (old firmware) and NDIS 6.5 (new firmware).

It is recommended that you attempt to install NDIS 6.5 driver first (instructions below). If the chosen driver is not “suitable” with the firmware on your NIC, the installation will not start. In this case proceed to install the NDIS 6.3 driver.

If your syn1588 hardware is older than revision 2.3 you might need a different driver version (e.g.: NDIS 5.0 for Rev. 2.1). If you are not sure which driver to install refer to ‘syn1588_release_OS_support.docx’ for further information.

If you wish to update the firmware of your syn1588 NIC contact our support support@oregano.at.

Installing the driver

Preparation (not required for NDIS 5.0)

As the NDIS 6.x driver versions are in BETA stage, they are not digitally signed. Therefore you will need to enable the loading of test-signed drivers on your PC.

To do this firstly disable Secure Boot in the BIOS settings. Then proceed to enabling the TESTSIGNING mode:

Execute the following BCDEdit command line with administrative rights in the Command Prompt:

```
Bcdedit.exe -set TESTSIGNING ON
```

You will see a watermark with the text “Test Mode” on the lower right corner of the Desktop after you restart the PC. Now you can proceed with the installation of the driver on your system.

Installation

Browse to the Control Panel and open the Device Manager. There should be a new network interface card listed for the syn1588® PCIe NIC. In the case that no matching driver is found or not installed yet the card shows up as a newly added ethernet device under “Other devices”.

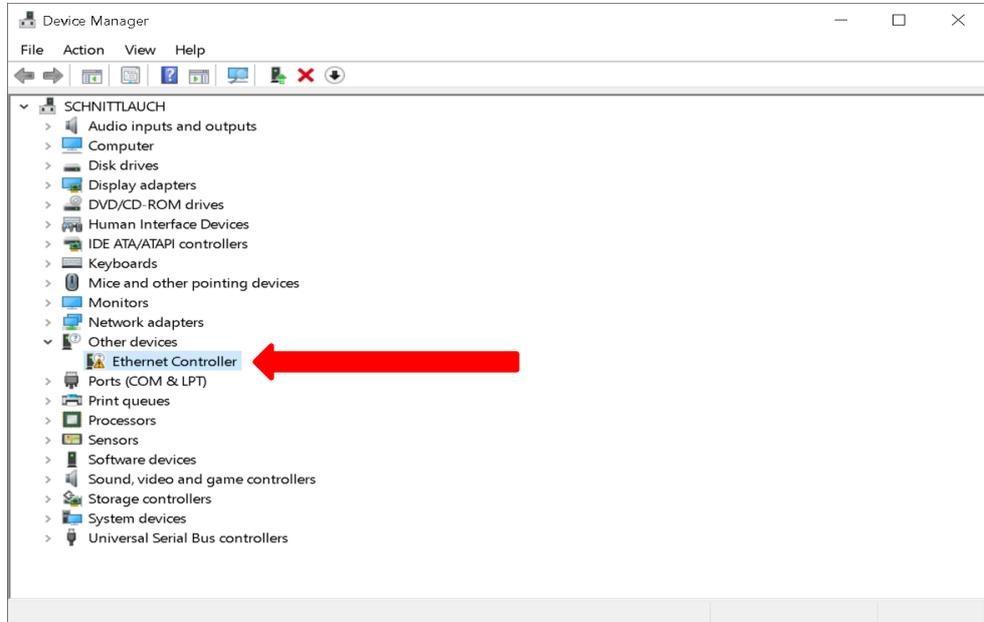


Figure 1: Device manager: Other Devices

In case the device is not visible first right click on network adapters and scan for hardware changes. A new network adapter shall appear.

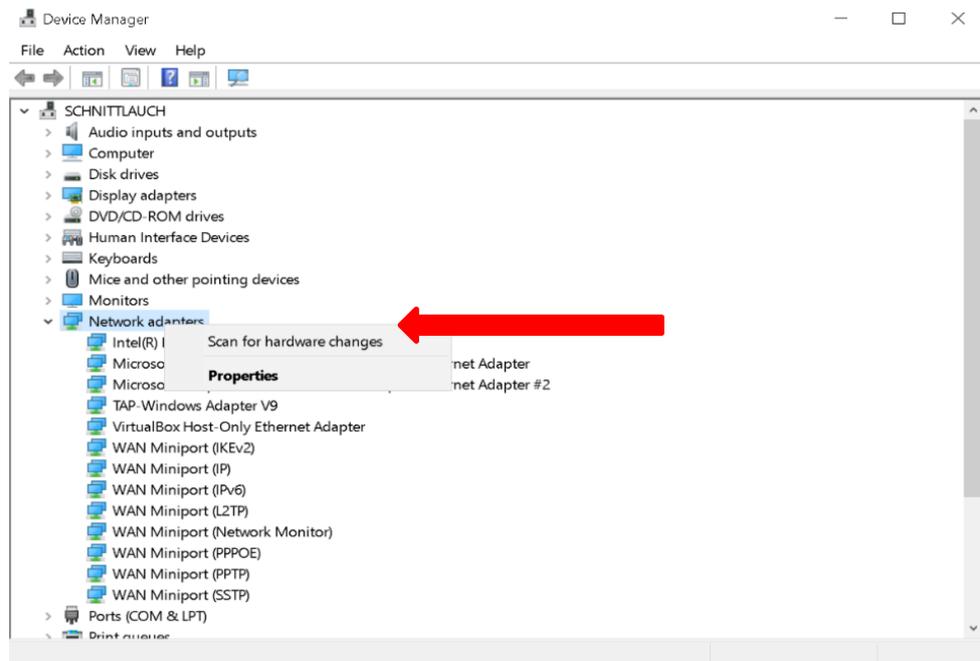


Figure 2: Device manager: scan for network adapters

After the ethernet controller appears, right click on it and choose “Update Driver Software...”.

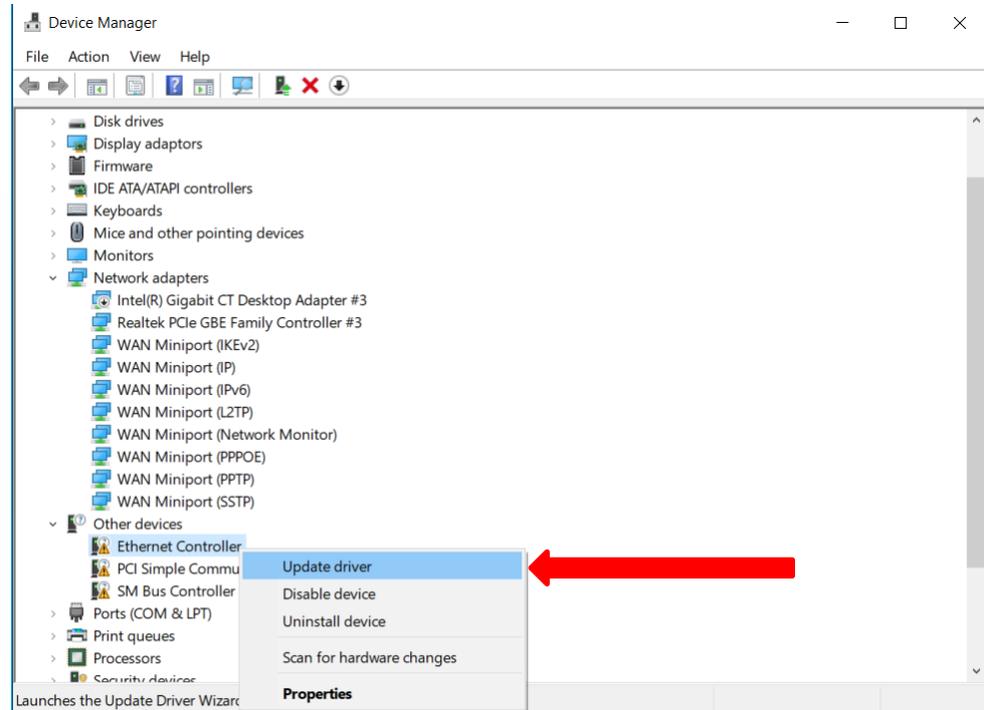


Figure 3: Device manager: Update driver

Choose “Browse my computer for driver software” in the next window and select the device driver folder “\syn1588\sw\windows_drv\ndis6.x(5.0)” on the provided syn1588® USB Stick.

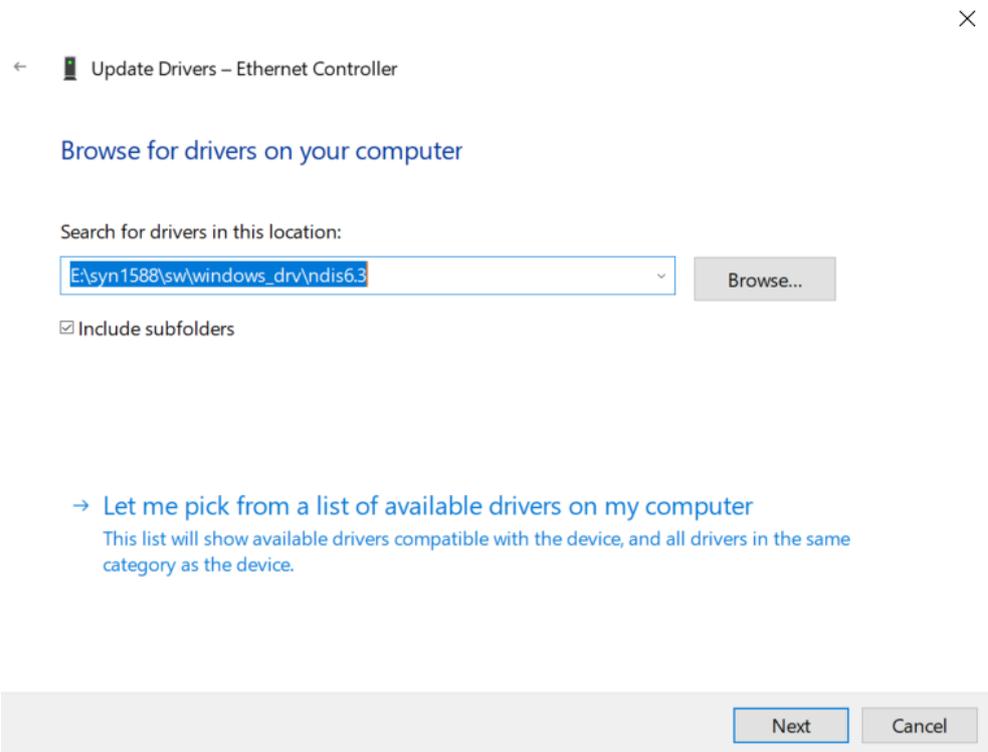


Figure 4: Browse for device driver folder

In the case of NDIS 6.x a Windows Security warning will appear and inform that Windows can't verify the publisher of this driver software (due to the BETA version of the software). Choose "Install this driver software anyway" to proceed with the installation.

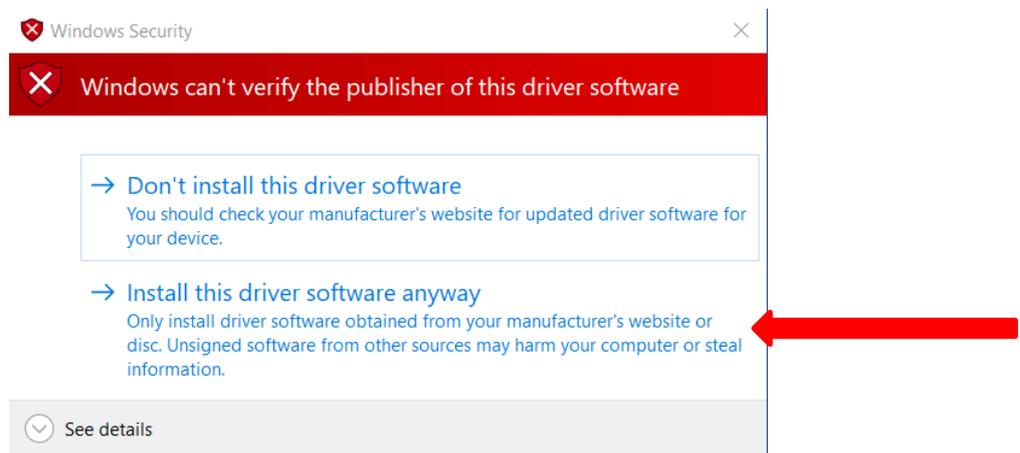


Figure 5: Windows Security install warning

After successful installation, the syn1588® PCIe NIC should be listed at the section "network cards" in the Windows device manager. For other driver installation issues see ReadMe file in the driver folder.

In case of installation issues

In case the installation procedure of the device driver software fails please check this chapter that describes how to fix frequently known issues.

Basic Presence Check of the syn1588[®] PCIe NIC

First step is to check for the presence of the syn1588[®] PCIe NIC in the device list of the computer. If this check fails the device is not recognized by the PC's BIOS. Reasons for this are either an inappropriate PCIe slot or the card is not correctly inserted. Please power-off your PC prior checking the proper connection. Note, the Revision 2.1 and the Revision 2.3 use a different PCI device identifier.

Revision 2.1 on Linux

Run the command "lspci | grep Altera" from any terminal. The command shall return a line like the following:

```
> lspci | grep Altera
03:00.0 Ethernet controller: Altera Corporation Device 1000 (rev 03)
```

If the command returns no entry the BIOS had not recognized the syn1588[®] PCIe NIC. Check that the syn1588[®] PCIe NIC had been properly inserted in an appropriate PCIe slot.

Revision 2.3 on Linux

Run the command "lspci | grep Meinberg" from any terminal. The command shall return a line like the following:

```
> lspci | grep Meinberg
03:00.0 Ethernet controller: Meinberg Funkuhren Device 0a01
```

If the command returns no entry the BIOS had not recognized the syn1588[®] PCIe NIC. Check that the syn1588[®] PCIe NIC had been properly inserted in an appropriate PCIe slot.

Windows

Browse to the Control Panel and open the Device Manager. There should be a new network interface card listed for the syn1588[®] PCIe NIC. In rare cases the card shows up as an unknown device. If no such device is present the BIOS had not recognized the syn1588[®] PCIe NIC.

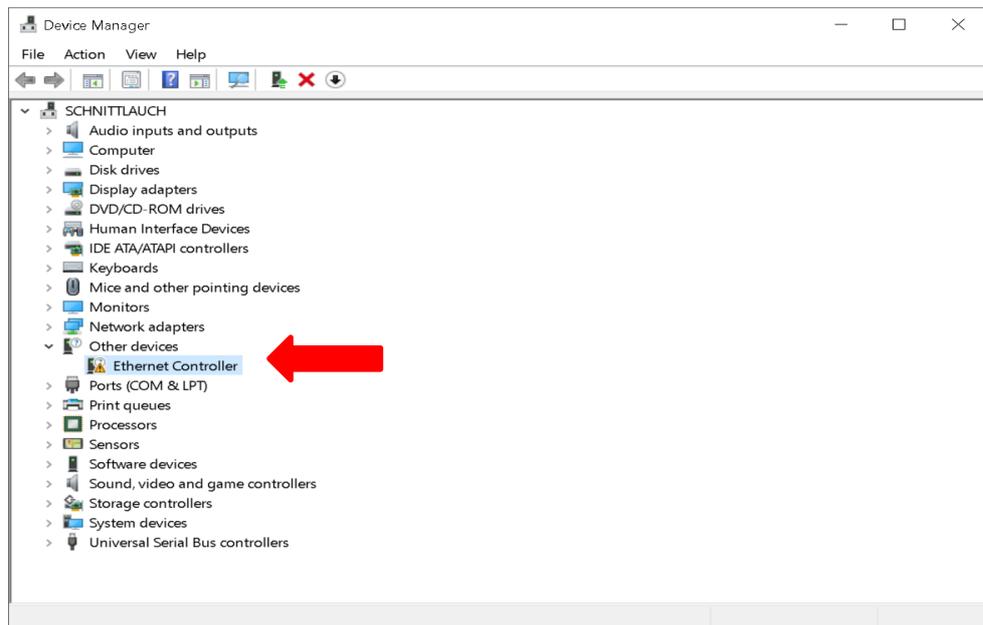


Figure 1: Device manager: Other Devices

Driver Check

If the syn1588® PCIe NIC had been successfully recognized by the PC's BIOS one has to check whether the device driver had been successfully installed.

Linux

One can use the "lsmod" command to check for the presence of the driver:

```
>lsmod | grep syn1588
syn1588_syncd          40960  2
oregano_mac            73728  1 syn1588_syncd
oregano_clock          65536  2 syn1588_syncd,oregano_mac
oregano_updater_altera  16384  1 syn1588_syncd
syn1588_pci            20480  0
si_time_pll            24576  1 syn1588_pci

>lsmod | grep oregano
oregano_mac            73728  1 syn1588_syncd
altera_pci_bridge      20480  1 oregano_mac
oregano_clock          65536  2 syn1588_syncd,oregano_mac
oregano_updater_altera  16384  1 syn1588_syncd
oregano_spi            20480  0
oregano_i2c            20480  1 oregano_mac
```

You can check the current version via the following command (assuming the syn1588 NIC is named "eth0"):

```
>ethtool -i eth0
driver: oregano-mac
version: v2.17-rc1-12-g86eadb1
firmware-version: MAC: 3180
expansion-rom-version:
bus-info: eth0
supports-statistics: yes
supports-test: yes
supports-eeprom-access: no
supports-register-dump: yes
supports-priv-flags: no
```

Installation from source

Extract the sources from the `syn1588-dkms_<version>.tar.gz` archive and navigate to the `linux` sub-directory. There you can find a Makefile providing the necessary build steps.

Kernel log example

```
[346549.202267] syn1588_pci 0000:02:00.0: load Oregano Syn1588 Nic pci driver
version: v2.17-rc1-12-g86eadb1
[346549.202456] syn1588_pci 0000:02:00.0: product name syn1588nic_rev2_3_a
[346549.202460] syn1588_pci 0000:02:00.0: build id 6299
[346549.202739] syn1588_pci 0000:02:00.0: finish init syn1588_pci
[346549.214980] oregano-updater-altera oregano-updater-altera.3.auto: probe
oregano updater altera driver
[346549.217927] oregano-spi oregano-spi.2.auto: probe oregano spi driver
[346549.219273] sit95147 spi0.0: driver loaded
[346549.219563] oregano-spi oregano-spi.2.auto: spi_oregano_probe(): SPI init OK
[346549.248038] oregano-clock oregano-clock.4.auto: probe oregano clock driver
[346549.248074] oregano-clock oregano-clock.4.auto: detected syn1588(R) Clock
version M237.
[346549.248080] oregano_clock: clock_m miits0: 0
[346549.248083] oregano_clock: clock_m miits1: 1
[346549.248085] oregano_clock: clock_m lsync_sync_status: 1
[346549.248086] oregano_clock: clock_m measurement: 0
[346549.248088] oregano_clock: clock_m event0: 1
[346549.248089] oregano_clock: clock_m event1: 1
[346549.248091] oregano_clock: clock_m trigger0: 1
[346549.248092] oregano_clock: clock_m trigger1: 1
[346549.248094] oregano_clock: clock_m period0: 1
[346549.248095] oregano_clock: clock_m period1: 1
[346549.248097] oregano_clock: clock_m time2: 1
[346549.248098] oregano_clock: clock_m irogb_output: 1
[346549.248100] oregano_clock: clock_m onestep: 1
[346549.248101] oregano_clock: clock_m period0cfg: 1
[346549.248103] oregano_clock: clock_m ntp: 1
[346549.248104] oregano_clock: clock_m leapset: 1
[346549.248106] oregano_clock: clock_m dutyperiod: 1
[346549.248107] oregano_clock: clock_m packetgenerator: 0
[346549.248109] oregano_clock: clock_m packetslaveformat: 0
[346549.248110] oregano_clock: clock_m precisionperiod: 1
[346549.248112] oregano_clock: clock_m irigb_input: 1
[346549.248113] oregano_clock: clock_m ext_vlan_pgkgen: 0
[346549.248114] oregano_clock: clock_m stop_clock: 1
[346549.248116] oregano_clock: clock_m hsnetworkindirect: 0
[346549.248117] oregano_clock: clock_m period2: 1
[346549.248119] oregano_clock: clock_m period3: 1
[346549.248120] oregano_clock: clock_m extime: 0
[346549.248122] oregano_clock: clock_m ptp_sync_status: 1
[346549.248123] oregano_clock: clock_m sdi_extended_support: 0
[346549.248125] oregano_clock: clock_m audio_counter: 0
[346549.248126] oregano_clock: clock_m sdi_ltc_timecode: 0
[346549.248128] oregano_clock: clock_m miits2_1g: 0
[346549.248129] oregano_clock: clock_m miits3_1g: 0
[346549.248131] oregano_clock: clock_m miits4_1g: 0
[346549.248132] oregano_clock: clock_m miits5_1g: 0
[346549.248133] oregano_clock: clock_m miits2_10g: 0
[346549.248135] oregano_clock: clock_m miits3_10g: 0
[346549.248136] oregano_clock: clock_m miits4_10g: 0
[346549.248138] oregano_clock: clock_m miits5_10g: 0
[346549.248139] oregano_clock: clock_m miits2_25g: 0
[346549.248141] oregano_clock: clock_m miits3_25g: 0
[346549.248142] oregano_clock: clock_m miits4_25g: 0
```

```

[346549.248144] oregano_clock: clock_m miits5_25g: 0
[346549.248145] oregano_clock: clock_m miits2_40g: 0
[346549.248147] oregano_clock: clock_m miits3_40g: 0
[346549.248148] oregano_clock: clock_m miits4_40g: 0
[346549.248150] oregano_clock: clock_m miits5_40g: 0
[346549.248151] oregano_clock: clock_m miits2_100g: 0
[346549.248152] oregano_clock: clock_m miits3_100g: 0
[346549.248154] oregano_clock: clock_m miits4_100g: 0
[346549.248155] oregano_clock: clock_m miits5_100g: 0
[346549.248159] oregano-clock oregano-clock.4.auto: preparing device file.
[346549.248161] oregano-clock oregano-clock.4.auto: configured clock frequency:
125000 kHz.
[346549.248165] oregano-clock oregano-clock.4.auto: setting initial clock step
size to 8.000 ns.
[346549.289952] oregano-mac oregano-mac.5.auto: probe oregano mac driver
...
<capability information cropped intentionally>
...
[346549.290185] oregano_mac: read permanent HW MAC address 8c:a5:a1:00:14:10
[346549.290199] oregano_mac: NIC_DEV_INIT: Initializing 64 TX DMA buffers of size
2048 and startoff 0.
[346549.319349] oregano-syncd oregano-syncd.6.auto: probe oregano clock driver
[346549.319359] oregano-syncd oregano-syncd.6.auto: invalid resource
[346549.319366] oregano-syncd oregano-syncd.6.auto: invalid resource
[346549.319370] oregano-syncd oregano-syncd.6.auto: invalid resource
[346549.319373] oregano-syncd oregano-syncd.6.auto: preparing device file.
[346549.360911] net eth0: NIC_DEV_OPEN:Configuring MAC for interface eth0.
[346549.360922] net eth0: no phy link, disable mac
[346549.361094] 8021q: adding VLAN 0 to HW filter on device eth0
[346549.361101] Set MAC_VLAN_GRP_ADR to 00000001
[346549.361106] MAC_MODE 40900
[346550.610549] net eth0: sfp module inserted
[346550.635542] Marvell 88E1111 i2c:eth0:16: attached PHY driver
(mii_bus:phy_addr=i2c:eth0:16, irq=POLL)
[346550.768922] oregano-mac oregano-mac.5.auto eth0: Link is Up - 1Gbps/Full -
flow control off
[346550.788975] oregano_mac: set phy delay for GMII copper
[346550.788984] net eth0: MAC PHY delay TX setting failed!
[346550.788991] net eth0: MAC PHY delay RX setting failed!
[346551.796183] oregano-mac oregano-mac.5.auto eth0: Link is Down
[346551.816235] oregano_mac: could not determit link delay
[346555.893091] oregano-mac oregano-mac.5.auto eth0: Link is Up - 1Gbps/Full -
flow control off
[346555.913150] oregano_mac: set phy delay for GMII copper
[346555.913160] net eth0: MAC PHY delay TX setting failed!
[346555.913167] net eth0: MAC PHY delay RX setting failed!

```

The marked output can be ignored for current Rev 2.3 NIC firmware as it does apply values for asymmetry correction automatically.

Windows

In case the device is not visible first right click on network adapters and scan for hardware changes. A new network adapter named “SYN1588 PCIe Ethernet Adapter” shall appear.

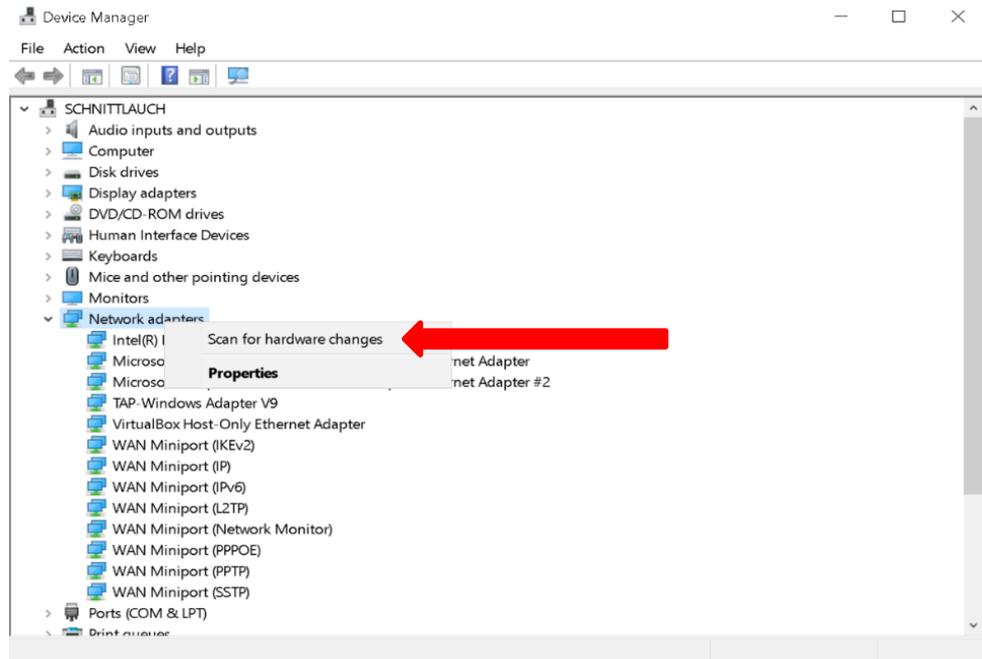


Figure 2: Device manager: scan for network adapters

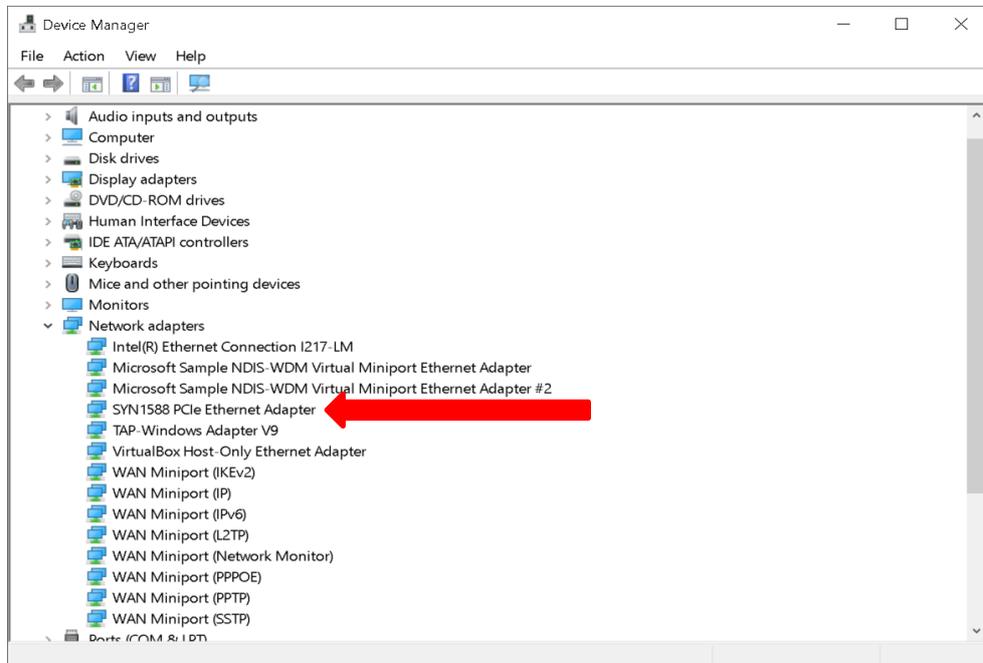


Figure 2.1: Device manager: new network adapter

To access more information or update the device driver travel to the menu “Properties” and in the dialog box select the “Driver” tab. Under this section one can select “Driver Details” to find out more details about the installed driver files or to “Update Driver” to update the installed device driver.

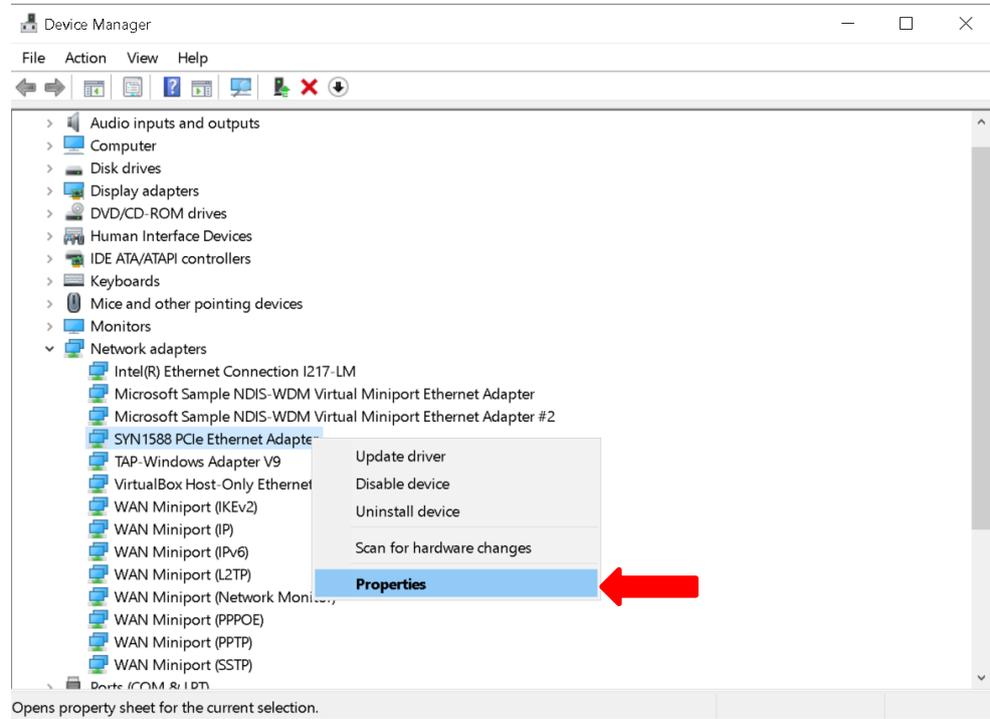


Figure 2.2: Device manager: syn1588® PCIe NIC

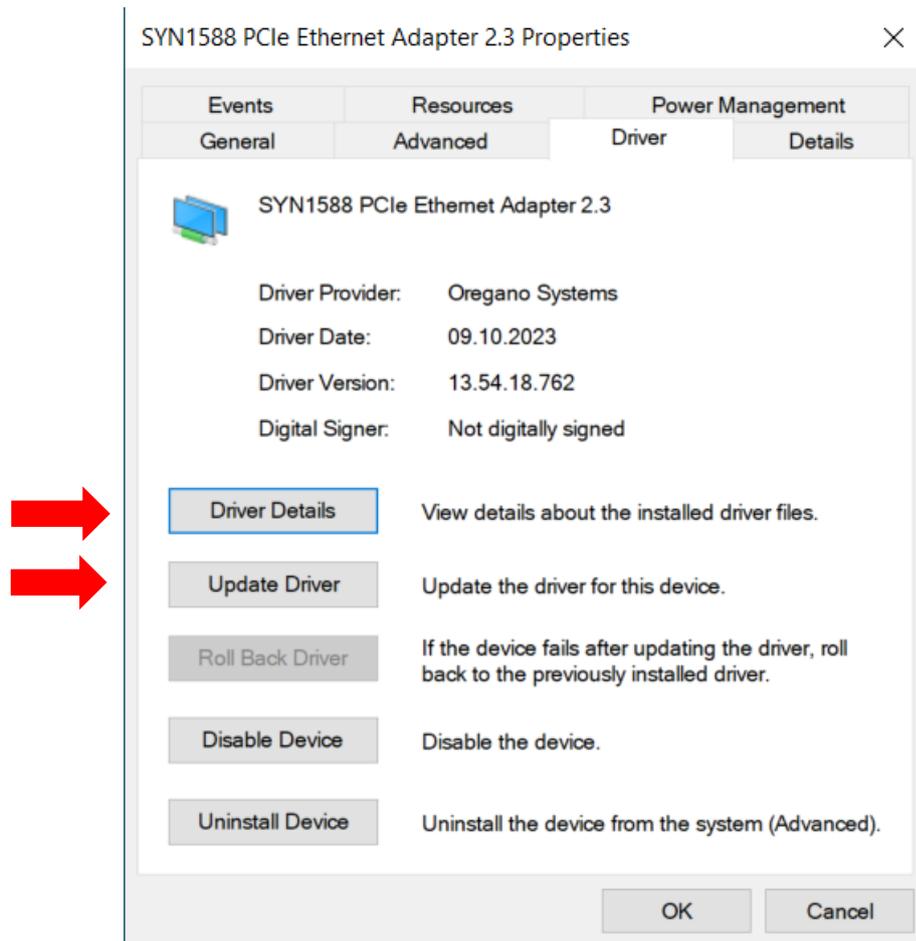


Figure 2.3: syn1588® PCIe NIC: Properties

An alternate way to check if the driver had been installed, is to enter in the command prompt the command “driverquery | findstr SYN1588”. No administrator rights are needed. This should bring up the syn1588® driver installed on the system.

```
> driverquery | findstr SYN1588
Module Name  Display Name  Driver Type  Link Date
=====
Syn1588Mini6 SYN1588 PCIe Ethernet  Kernel      09.10.2023 13:54:23
```

If this output message is not received, the installation of the driver was not successful. Please re-install the driver as shown in the previous steps or refer to the tutorial [How to install a driver in Windows for a syn1588® PCIe NIC](#).

Uninstalling the driver

Uninstalling the device driver in Linux – Debian/Ubuntu .deb

To remove the syn1588 driver from your system you can also use the dpkg packet manager. Open a root shell and execute the following command:

```
> apt autoremove syn1588-dkms
```

When asked for confirmation type 'y'. This will also remove all installed dependencies. To verify if the device driver was successfully uninstalled check [Chapter Driver Check](#).

Uninstalling the device driver in Linux – RedHat/CentOS/SUSE .rpm

If you installed the rpm driver package proceed as described below: Open a root shell and execute the following command:

```
# v7 (yum)
> yum autoremove syn1588-dkms
# v8 (dnf)
> dnf autoremove syn1588-dkms
```

When asked for confirmation type 'y'. This will also remove all installed dependencies. To verify if the device driver was successfully uninstalled check [Chapter Driver Check](#).

Uninstalling the device driver on Windows

Browse to the Control Panel and open the Device Manager. There should be a new network interface card listed for the syn1588[®] PCIe NIC. Right-click on the syn1588 device and choose Uninstall device. A warning will appear for your confirmation. To completely delete the driver off the system select the checkbox below and click 'Uninstall'.

First Steps with your installed syn1588® PCIe NIC

The easiest way to use the syn1588® PTP Stack and other tools is the command line or shell. Please ensure your shell-session has sufficient rights. For Windows an administrator-command line, and for Linux a root-shell is recommended.

As a first step you need to ensure that the syn1588® PCIe NIC has a valid IP-address assigned to it either via DHCP or manual configuration. For Windows write down this IP-address (i.e. 192.168.0.123) for further reference. For Linux you require the adapter name (i.e. eth3).

Starting the syn1588 Utility

The syn1588 utility is used to control and check several functions of the syn1588® PCIe NIC. For more information about the syn1588 Utility please refer to the syn1588® User Guide (UG001) Chapter 7.1.

syn1588® Linux Live System

```
> cd /opt/oregano/bin
> sudo ./syn1588
```

The output of the syn1588 utility on Linux will look like this example output:

```
2022-01-05 14:20:16.942478 [INFO ] [ ] syn1588(R) Driver Interface
2022-01-05 14:20:16.942508 [INFO ] [ ] Build date: 2022-01-
05T12:43:16 - v1.14-9-g7cbf9e73
2022-01-05 14:20:16.942514 [INFO ] [ ] Copyright (c) Oregano
Systems - Design & Consulting GesmbH 2005-2022
2022-01-05 14:20:16.942519 [INFO ] [ ] Confidential unpublished
data - All rights reserved
2022-01-05 14:20:16.942539 [INFO ] [syn1588 ] Syn1588Ifc requires at
least:
2022-01-05 14:20:16.942546 [INFO ] [syn1588 ] - linux driver version 1.4-
15-g05b7283
2022-01-05 14:20:16.942550 [INFO ] [syn1588 ] - windows driver version
10/05/2017, 10.9.16.182
2022-01-05 14:20:16.942565 [INFO ] [syn1588 ] Device /dev/syncD0 found
2022-01-05 14:20:16.942582 [INFO ] [syn1588 ] syn1588(R) Hardware Clock M
2.3.5 f=125000000 Hz
2022-01-05 14:20:16.942593 [INFO ] [syn1588 ] Found stop clock support
2022-01-05 14:20:16.942610 [INFO ] [syn1588 ] Using MAC TS Version 3160
2022-01-05 14:20:16.942624 [INFO ] [syn1588 ] Using programmable 1-step
TS
Selected syn1588(R) card 0>
```

Version and status information about the tool and your syn1588® PCIe NIC is shown, if everything was set up correctly. Additionally, one may enter the command “build” on the prompt of the syn1588 utility to get even more information on the syn1588® PCIe NIC.

If any problems occur, please ensure the previous steps were completed successfully.

The syn1588 Software Suite is delivered in ready-to-use installers for Linux and Windows systems. For more information about custom installation please refer to the (AN038) Chapter “Installer”.

Linux Custom Installation

As default the syn1588 Software Suite is installed in “/opt/oregano/”. To start the syn1588 utility go to the “bin” folder and start the application as shown.

```
> cd /opt/oregano/bin
> ./syn1588
```

The output of the syn1588 utility on Linux will look like the example above.

Windows Custom Installation

To start the syn1588 utility go to the chosen location where the syn1588® Software Suite is installed. The “bin” folder contains the application binaries including the syn1588 utility.

```
> cd <chosen location>\bin
> syn1588.exe
```

The output of the syn1588 utility on Windows will look like this example output:

```
2022-01-05 14:20:16.942478 [INFO ] [ ] syn1588(R) Driver Interface
2022-01-05 14:20:16.942508 [INFO ] [ ] Build date: 2022-01-
05T12:43:16 - v1.14-9-g7cbf9e73
2022-01-05 14:20:16.942514 [INFO ] [ ] Copyright (c) Oregano
Systems - Design & Consulting GesmbH 2005-2022
2022-01-05 14:20:16.942519 [INFO ] [ ] Confidential unpublished
data - All rights reserved
2022-01-05 14:20:16.942539 [INFO ] [syn1588 ] Syn1588Ifc requires at
least:
2022-01-05 14:20:16.942546 [INFO ] [syn1588 ] - linux driver version 1.4-
15-g05b7283
2022-01-05 14:20:16.942550 [INFO ] [syn1588 ] - windows driver version
10/05/2017, 10.9.16.182
2022-01-05 14:20:16.942582 [INFO ] [syn1588 ] syn1588(R) Hardware Clock M
2.3.5 f=125000000 Hz
2022-01-05 14:20:16.942593 [INFO ] [syn1588 ] Found stop clock support
2022-01-05 14:20:16.942610 [INFO ] [syn1588 ] Using MAC TS Version 3160
2022-01-05 14:20:16.942624 [INFO ] [syn1588 ] Using programmable 1-step
TS
Selected syn1588(R) card 0
```

Starting the syn1588® PTP Stack

The syn1588® PTP Stack is a lightweight Precision Time Protocol stack which enables highly accurate clock synchronization in local area networks. For more information about the syn1588® PTP Stack please refer to the syn1588® User Guide (UG001) Chapter 6.

To start your PC as a PTP node, run the syn1588® PTP Stack as follows. Please change the IP-address or interface name accordingly to your system settings.

Linux Live System

```
> cd /opt/oregano/bin
> sudo ./ptp -i enp2s0 -C S -v info -r 0
```

The first part of the output on Linux will look like this example output:

```
2022-01-05 15:48:48.845172 [INFO ] [ ] syn1588(R) PTP Stack -
IEEE1588-2008 Engine
2022-01-05 15:48:48.845206 [INFO ] [ ] Build date: 2022-01-
05T12:43:16 - v1.14-9-g7cbf9e73
2022-01-05 15:48:48.845212 [INFO ] [ ] Copyright (c) Oregano
Systems - Design & Consulting GesmbH 2005-2022
2022-01-05 15:48:48.845217 [INFO ] [ ] Confidential unpublished
data - All rights reserved
2022-01-05 15:48:48.845224 [INFO ] [ ] syn1588(R) PTP Stack
started: 2022-01-05 14:48:48.845222 (UTC)
2022-01-05 15:48:48.845317 [INFO ] [p1 ] Port 1: adding config "i" =
"enp2s0"
2022-01-05 15:48:48.845327 [INFO ] [p1 ] Port 1: adding config "C" =
"S"
2022-01-05 15:48:48.845333 [INFO ] [p1 ] Port 1: adding config "v" =
"info"
2022-01-05 15:48:48.845338 [INFO ] [p1 ] Port 1: adding config "r" =
"0"
2022-01-05 15:48:48.845344 [INFO ] [p1 ] PTP version 2.0
2022-01-05 15:48:48.845348 [INFO ] [ ] Command line: ./ptp -i
enp2s0 -C S -v info -r 0
2022-01-05 15:48:48.845358 [WARNING ] [ ] Failed to set high priority
for the PTP thread!
2022-01-05 15:48:48.845365 [INFO ] [p1 ] Found Configuration for 1
ports
2022-01-05 15:48:48.845563 [INFO ] [syn1588 ] Syn1588Ifc requires at
least:
2022-01-05 15:48:48.845571 [INFO ] [syn1588 ] - linux driver version 1.4-
15-g05b7283
2022-01-05 15:48:48.845576 [INFO ] [syn1588 ] - windows driver version
10/05/2017, 10.9.16.182
2022-01-05 15:48:48.845590 [INFO ] [syn1588 ] Device /dev/syncD0 found
2022-01-05 15:48:48.845608 [INFO ] [syn1588 ] syn1588(R) Hardware Clock M
2.3.5 f=125000000 Hz
2022-01-05 15:48:48.845617 [INFO ] [syn1588 ] Found stop clock support
2022-01-05 15:48:48.845630 [INFO ] [syn1588 ] Using MAC TS Version 3160
2022-01-05 15:48:48.845635 [INFO ] [syn1588 ] Using programmable 1-step
TS
2022-01-05 15:48:48.845653 [INFO ] [syn1588 ] syn1588(R) PCIe NIC
Revision 2, Build 876
2022-01-05 15:48:48.845669 [INFO ] [p1 ] Using syn1588 mode
2022-01-05 15:48:48.845699 [INFO ] [p1.clock ] Spike M2S: Init with ival
0, buffer size 16
```

```

2022-01-05 15:48:48.845762 [INFO    ] [pl.clock    ] Spike Path: Init with ival
0, buffer size 16
2022-01-05 15:48:48.845778 [INFO    ] [pl.io       ] Init shared mem
2022-01-05 15:48:48.848090 [INFO    ] [syn1588    ] syn1588HwClk: clearing leap
second jump
2022-01-05 15:48:48.848103 [INFO    ] [pl.engine   ] Settings: ClockId
00:1e:c0:ff:fe:85:de:2b
2022-01-05 15:48:48.848112 [INFO    ] [pl.engine   ] Settings: Prio1 128
ClkClass 255 clkAccuracy 39 clkVariance 65535
2022-01-05 15:48:48.848119 [INFO    ] [pl.engine   ] Settings: Prio2 128 Domain
0
2022-01-05 15:48:48.848235 [INFO    ] [pl.network.mcport] SIOCShWTSTAMP: tx_type
1 requested, got 1; rx_filter 0 requested, got 12
2022-01-05 15:48:48.848245 [INFO    ] [pl.network.mcport] Activated
SO_TIMESTAMPING hardware
2022-01-05 15:48:48.848340 [INFO    ] [pl.network.mcport] SIOCShWTSTAMP: tx_type
1 requested, got 1; rx_filter 0 requested, got 12
2022-01-05 15:48:48.848349 [INFO    ] [pl.network.mcport] Activated
SO_TIMESTAMPING hardware
2022-01-05 15:48:48.848599 [INFO    ] [pl.clock    ] Clk: Using Oregano Systems;
syn1588(R) PCIe NIC Revision 2; 00:1E:C0:85:DE:2B
2022-01-05 15:48:48.848608 [INFO    ] [pl.clock    ] with ClockId
00:1e:c0:ff:fe:85:de:2b
2022-01-05 15:48:48.848742 [INFO    ] [pl.clock    ] Clk: Resetting servos
2022-01-05 15:48:48.848748 [INFO    ] [pl.clock    ] Clk: Resetting filters

```

Linux Custom Installation

As default the syn1588 Software Suite is installed in “/opt/oregano/”. To start the syn1588 PTP Stack go to the “bin” folder and start the application as shown.

```

> cd /opt/oregano/bin
> sudo ./ptp -i eth3 -v2

```

Windows Custom Installation

To start the syn1588 PTP Stack go to the chosen location where the Syn1588 Software Suite is installed. The “bin” folder contains the application binaries including the syn1588 PTP Stack.

```

> cd <chosen location>\bin
> ptp.exe -i 192.168.0.123 -v2

```

FAQ, Caveats, and Common Traps and Pitfalls

- For a list of supported operating systems please consult the “syn1588_release_OS_support” document found in the syn1588 Linux Live System together with this document.
- Please ensure, that your syn1588[®] PCIe NIC has an IP address assigned. Check also for correct setting of the subnet mask, gateway IP address and DNS server.
- Check, if the IP address set does not interfere with other IP addresses on the same machine or in the network. Ideally a separated subnet should be assigned to all PTP devices at least during initial testing.
- On all operating systems the syn1588[®] PTP Stack requires root-privileges to run correctly.
- Check, if the firewall blocks any PTP traffic. PTP (in layer 3 mode) communicates via IP/UDP port 319 and 320. Please make sure that these ports are not blocked by your firewall.
- Simple way to check the configured IP address and functioning of hardware and/or software is to ping another host IP address. Also Wireshark can be used to check if the packets are received/transmitted at the configured network interface.
- If there are difficulties in using the syn1588[®] hardware and/or software, generate a system information report as specified in this Application Note (AN005). Collect the syn1588[®] System Information Report and send it to us for further support.

How-To

This chapter briefly describes commonly used scenarios using the syn1588[®] PCIe NIC.

How-To Generate a 1 PPS Output Signal?

As soon as the driver is properly installed for the syn1588[®] PCIe NIC a 1 PPS output signal is generated automatically. If there is not master in the network available, the syn1588[®] PCIe NIC acts as a master and thus the 1 PPS signal is free running (based on the local oscillator). One may use the IOMATRIX register (please refer to the application note [an_register_map.pdf](#) for a detailed register description) to define the SMA connector that outputs the 1 PPS signal. Note, that one may select more than one SMA output for driving the 1 PPS signal.

How-To Generate a Frequency Output Signal?

There are two PERIOD outputs of the syn1588[®] Clock_M IP Core that may be programmed to generate a symmetrical frequency signal. One has to define half of the desired clock period in the registers PERIODTIME0/1_L/H respectively (please refer to the application note an_register_map.pdf for a detailed register description). Note that PERIODTIME0/1 are 64 bit values and thus require two 32 bit registers _L (low) and _H (high).

One may use the IOMATRIX register to define the SMA connector that outputs the generated PERIOD signal. One may directly generate frequencies in the range of mHz (correct that are milli Hertz) to some MHz. The upper frequency limit is given by the syn1588[®] clock frequency (at least 4 clock cycles per half-period are needed) and the digital jitter. Note that the generated frequency signal has not to be a multiple of the base clock. This digital jitter is negligible at frequencies up to 1 MHz and becomes dominant above 10 MHz.

There is the syn1588 utility which allows you to specify the function on command line. Additionally there is the syn1588[®] API that offers a programmatically interface.

Please check the application note AN008 which describes using the on-board jitter cleaner PLL and AN007 which describes generating and timestamping signals

How-To Synchronize the System Clock?

The syn1588[®] PTP Stack running on a node equipped with a syn1588[®] NIC synchronizes the local hardware clock. If one would like to have the local system clock synchronized as well, one has to use the ISync utility (see chapter 6.3 of the syn1588[®] User Guide).

Note that IEEE 1588 uses a different time notation as your system clock or NTP! Thus one has to deal on how to translate leap seconds present in your system time that are unknown to IEEE 1588 (TAI notation). Caution: This translation may cause “jumps” in your time which can cause problems with software affected by the system time! There is no better time representation. Both solutions offer pros and cons – everything else is a religious discussion.

How-To Synchronize to a Local GPS Receiver?

The syn1588[®] PTP Stack running on a node equipped with a syn1588[®] NIC may be synchronized to an external GPS receiver using the eSync utility (see chapter 6.2 of the syn1588[®] User Guide or AN009). One just needs to connect the 1 PPS signal from the GPS receiver to the respective input on the compatible data stream.

How-To Timestamp an External (Hardware) Event?

There are two EVENT inputs (please refer to the application note application note AN007) to the syn1588® Clock_M IP Core. Every event (i.e. rising edge of this signal) will draw a timestamp without any software interaction. Optionally an interrupt may be issued in your system. EVENT0 input offers a FIFO allowing to capture and timestamp even dense (approx. 150 ns) events. EVENT1 offers a register interface and thus is suited for capturing slow events.

How-To Timestamp an Application Software Event?

Capturing a timestamp based on events in your software “just” requires reading the TIME2 registers (please refer to the application note an_register_map.pdf for a detailed register description). These three registers hold the current time. The application may skip reading the TIME2_E register frequently since this – slowly changing – time information may be handled on application level.

While reading the two/three 32 bit registers is a quite simple and fast task from the hardware perspective it can be quite difficult on a node running a standard operating system (OS) that is not capable to handle real-time events. Note, an OS like Linux or Windows may introduce varying latencies from your request to read the hardware register to the actual read operation in the range of microseconds to even milliseconds.

Of course, the fastest method is to read the system clock since dedicated fast function calls are present for this task event in standard OS. We recommend synchronizing your local system clock to the IEEE 1588 hardware clock and reading the system clock instead.

The error introduced by using the inaccurate system clock is smaller than the error introduced by the variation of latencies for hardware calls.

Caution:

Please make sure that the hardware clock uses a reasonable timescale to avoid problems with your system. Please do also note that reading the TIME registers will interfere with the syn1588® PTP Stack running on your node and cause malfunctions!

How-To Download Updated Driver or Software?

If you would like to receive updated driver or software for your syn1588® products please check our web site where we provide the latest driver software.

How-To Issue and Handle an Interrupt on an Event?

There are several conditions that may flag an interrupt to the host CPU. There is the Interrupt Enable register IREN (please refer to the application

note an_register_map.pdf for a detailed register description) that enables or disables the interrupt conditions and the Interrupt Status register IRSRC (please refer to the application note an_register_map.pdf for a detailed register description) that holds the status of the enabled interrupt conditions.

The following table lists the possible conditions that may be selected to create an interrupt.

Interrupt Events
Ethernet timestamping unit 0 RX
Timestamp FIFO unit 0 RX full
Ethernet timestamping unit 0 TX
Timestamp FIFO unit 0 TX full
Ethernet timestamping unit 1 RX
Timestamp FIFO unit 1 RX full
Ethernet timestamping unit 1 TX
Timestamp FIFO unit 1 TX full
Event on EVENT0 input
Event on EVENT1 input
TRIGGER0 elapsed
TRIGGER1 elapsed
PERIOD0 elapsed
PERIOD1 elapsed
TIME overflow

Table 1 syn1588® PCIe NIC interrupt events

Literature

AN005. *Application Note: "syn1588® System Information Report"*. Oregano Systems, Version 1.13.0 - July 2021.

AN038. *Application Note: "syn1588® Linux Life System Quickstart Guide"*. Oregano Systems, Version 1.13.2- September 2021.

AN041. *Application Note: " What's new with syn1588® PCIe NIC Revision 2.3"*. Oregano Systems, Version 1.16- November 2022

UG001. *User Guide: "syn1588® User Guide"*. Oregano Systems, Version 1.13.1 - July 2021.

 <p>Oregano Systems A Meinberg Company</p> <p>Franzosengraben 8 A-1030 Vienna Austria http://oregano.at contact@oregano.at</p>	<p>Copyright © 2024 Oregano Systems – Design & Consulting GmbH ALL RIGHTS RESERVED.</p> <p>Oregano Systems does not assume any liability arising out of the application or use of any product described or shown herein nor does it convey any license under its patents, copyrights, or any rights of others.</p> <p>Licenses or any other rights such as, but not limited to, patents, utility models, trademarks or tradenames, are neither granted nor conveyed by this document, nor does this document constitute any obligation of the disclosing party to grant or convey such rights to the receiving party.</p> <p>Oregano Systems reserves the right to make changes, at any time without notice, in order to improve reliability, function or design. Oregano Systems will not assume responsibility for the use of any circuitry described herein.</p> <p>All trademarks used in this document are the property of their respective owners.</p>
--	--