syn1588 ®

Version 1.16 - November 2022

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 v1.13 the full file name is syn1588-dkms_1.13.1_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
```

After successful installation, the device driver has to be loaded using the following command:

```
> modprobe SyncDriver
```

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

```
> rmmod SyncDriver.ko
> rmmod syn1588nic.ko
```

After this step is completed load the new driver:

```
> modprobe SyncDriver
```

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 BETA device driver in Windows (NDIS 6.3)

As the current BETA driver is not digitally signed you will need to disable the Driver signature enforcement in Windows.

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/ndis6.3" 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.

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.
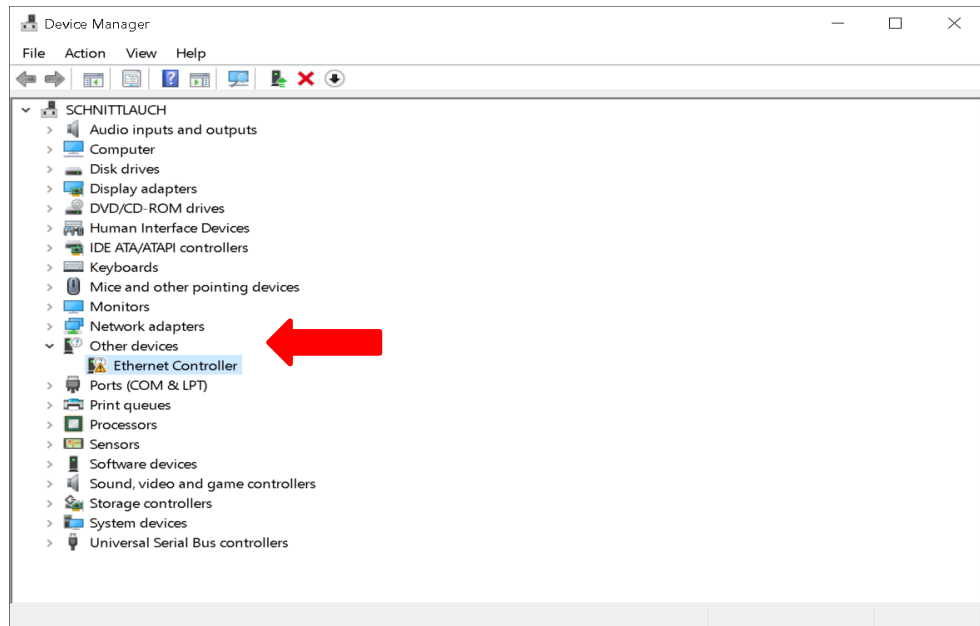

Figure 1: Device manager: Other Devices

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

For Linux there is one device drivers that has to be loaded:

- syn1588

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

```
> lsmod |grep syn1588
syn1588                86016  0
mii                    20480  1 syn1588
```

You can check the current version via the following command:

```
> modinfo syn1588
filename:       /lib/modules/5.4.0-113-generic/updates/dkms/syn1588.ko
alias:          SyncDriver
alias:          syn1588nic
alias:          pci:v00001172d00001000sv00000000sd00000000bc02sc00i00
version:        v1.15-2-g7c21a0e
license:        GPL v2
description:    syn1588(R) NIC device driver
author:         Norbert Summer <summer@oregano.at>
srcversion:     96D9E1EB9D96C9BB307ACA5
depends:        mii
retpoline:      Y
name:           syn1588
vermagic:       5.4.0-113-generic SMP mod_unload modversions
sig_id:         PKCS#7
signer:         syn1588 Secure Boot Module Signature key
sig_key:        7A:96:0F:29:70:2E:F0:87:DF:DF:D7:EF:F2:E4:AA:76:ED:4F:11:1B
sig_hashalgo:   sha512
signature:      0F:9F:34:F4:98:15:95:FD:20:10:DD:BD:24:2F:BF:BB:EB:64:0A:0B:
                AB:73:BB:C2:FA:04:2A:9F:B2:8A:1F:C4:E9:B4:9A:C1:D6:3F:E0:58:
                94:08:41:55:8A:87:F9:01:78:74:70:83:07:6D:3B:47:F0:99:44:26:
                17:45:3E:16:78:CE:D1:0C:A5:76:58:0E:E6:3F:C0:13:E5:B1:18:C1:
                6F:DC:5E:5A:E9:1F:D8:2B:2E:3C:34:7A:EC:7A:F2:89:18:5E:30:D5:
                F8:6B:32:BE:21:D3:25:2A:79:2D:B9:9C:E6:BD:2D:EE:B2:40:18:F4:
                F8:C5:8E:36:03:6D:39:54:56:43:AA:0C:71:4C:74:45:65:C1:AF:F4:
                06:86:A9:9E:5E:FF:0E:2C:5C:7F:B6:E2:47:26:85:7E:92:75:7E:E6:
                E1:5D:2E:32:C5:6C:BC:B0:E4:3C:D0:19:BE:29:C8:E5:25:DF:AE:2C:
                E8:0A:BF:2F:46:AF:47:87:B5:BE:5F:37:1C:DD:E2:F0:41:54:B6:A0:
                D2:9C:D0:15:EA:38:A7:4C:46:52:FB:DB:A0:6E:48:5F:B7:5B:5C:75:
                11:63:FC:F7:F4:25:14:27:62:F0:AE:EA:2F:15:1A:C1:F6:DA:47:CB:
                13:21:5A:E6:23:0B:27:47:17:C7:91:9F:91:CF:D4:39
parm:           clock_m_frq:syn1588(R) clock frequency [KHz] (int)
parm:           clock_m_dev:Set to anything but -1 to enable the clock_mX device
(int)
parm:           syn1588_extime_enable:use only one clock in a dualnic (bool)
parm:           syn1588_disabled_output:all outputs are disabled by default (bool)
```

Manual driver installation from source

Extract the sources from the syn1588-dkms_<version>.tar.gz archive and navigate to the linux sub-directory.

```
> make modules_install
```

You may see the following or a similar output:

```
bash scripts/write_version.sh
tying to build driver with kernel source /lib/modules/5.4.0-77-generic/build
  CC [M]  /opt/oregano/syn1588_drv/syn1588-dkms_1.13.1.tar/linux/SyncDriver.o
  CC [M]  /opt/oregano /syn1588_drv/syn1588-
dkms_1.13.1.tar/linux/syn1588nic_main.o
  CC [M]  /opt/oregano/syn1588_drv/syn1588-dkms_1.13.1.tar/linux/syn1588nic_pci.o
  CC [M]  /opt/oregano/syn1588_drv/syn1588-dkms_1.13.1.tar/linux/syn1588_debugfs.o
  LD [M]  /opt/oregano/syn1588_drv/syn1588-dkms_1.13.1.tar/linux/syn1588nic.o
  Building modules, stage 2.
  MODPOST 2 modules
  CC [M]  /opt/oregano/syn1588_drv/syn1588-dkms_1.13.1.tar/linux/SyncDriver.mod.o
  LD [M]  /opt/oregano/syn1588_drv/syn1588-dkms_1.13.1.tar/linux/SyncDriver.ko
  CC [M]  /opt/oregano/syn1588_drv/syn1588-dkms_1.13.1.tar/linux/syn1588nic.mod.o
  LD [M]  /opt/oregano/syn1588_drv/syn1588-dkms_1.13.1.tar/linux/syn1588nic.ko
  INSTALL /mnt/user_puhm/puhm/design/syn1588_drv/syn1588-
dkms_1.13.1.tar/linux/SyncDriver.ko
At main.c:160:
- SSL error:02001002:system library:fopen:No such file or directory:
../crypto/bio/bss_file.c:69
- SSL error:2006D080:BIO routines:BIO_new_file:no such file:
../crypto/bio/bss_file.c:76
sign-file: certs/signing_key.pem: No such file or directory
  INSTALL /mnt/user_puhm/puhm/design/syn1588_drv/syn1588-
dkms_1.13.1.tar/linux/syn1588nic.ko
At main.c:160:
- SSL error:02001002:system library:fopen:No such file or directory:
../crypto/bio/bss_file.c:69
- SSL error:2006D080:BIO routines:BIO_new_file:no such file:
../crypto/bio/bss_file.c:76
sign-file: certs/signing_key.pem: No such file or directory
  DEPMOD  5.4.0-77-generic
Warning: modules_install: missing 'System.map' file. Skipping depmod.
depmod -b "/" -a 5.4.0-77-generic
```

Note, some Linux versions require a signed device driver. The "SSL errors" messages marked with red color in the example output above are caused by the fact that the driver is not signed or properly signed by the Linux OS which requires a signed driver. Follow the step-by-step instructions of the "README" file included in the driver directory to generate such a signed driver.

If the drivers had been successfully installed one can check the respective boot messages in the system log. For this enter the command "dmesg | grep syn1588".

```
[    3.124202] syn1588: loading out-of-tree module taints kernel.
[    3.124245] syn1588: module verification failed: signature and/or required key
missing - tainting kernel
[    3.124531] Oregano Systems syn1588(R) PCIe/Dual NIC Device Driver (syn1588nic)
v1.14-2-g3068ead
[    3.124679] syn1588nic 0000:02:00.0: enabling device (0000 -> 0002)
[    3.124725] syn1588nic: allocating device resources.
[    3.124754] syn1588: system_cap: eth_build: 876
[    3.124755] syn1588: system_cap: has_tse_mac: 1
[    3.124755] syn1588: system_cap: has_clockm: 1
[    3.124755] syn1588: system_cap: has_pll: 1
[    3.124756] syn1588: system_cap: has_oxco: 1
[    3.124756] syn1588: system_cap: is_sfp: 0
[    3.124756] syn1588: system_cap: has_spi: 1
[    3.124756] syn1588: system_cap: has_ext_clk_input: 1
[    3.124757] syn1588: system_cap: is_rev21: 1
[    3.124757] syn1588: system_cap: has_i2c_oscillator: 0
[    3.124757] syn1588: system_cap: is_dual_nic: 0
[    3.124758] syn1588: system_cap: has_extime: 0
[    3.124758] syn1588: system_cap: has_mac_serial: 0
[    3.124758] syn1588: system_cap: has_high_speed_nic: 0
[    3.124761] syn1588: device resources (IRQ #17, I/O 0xF7100000 (65536))
detected.
[    3.124795] syn1588: detected syn1588(R) Clock version M235.
[    3.124796] syn1588: ClockFreq: req_version 11, value 35
[    3.124799] syn1588: preparing device file.
[    3.125626] syn1588: device file clockm0 setup (minor #58) complete.
[    3.125627] syn1588: configured clock frequency: 125000 kHz.
[    3.125627] syn1588: setting initial clock step size to 8.0 ns.
[    3.125670] syn1588: nic_cap rx_timestamper_implemented: 1
[    3.125670] syn1588: nic_cap tx_timestamper_implemented: 1
[    3.125670] syn1588: nic_cap one_step_implemented: 1
[    3.125671] syn1588: nic_cap statistic_counter_32bit: 1
[    3.125671] syn1588: nic_cap vlan_4_selectable: 1
[    3.125671] syn1588: nic_cap vlan_group_implemented: 1
[    3.125672] syn1588: nic_cap statistic_counter_additional_implemented: 1
[    3.125672] syn1588: nic_cap base_1g: 1
[    3.125672] syn1588: nic_cap base_10g: 0
[    3.125672] syn1588: nic_cap base_25g: 0
[    3.125673] syn1588: nic_cap fec_available: 0
[    3.125673] syn1588: nic_cap jumbo_paket_support: 1
[    3.125673] syn1588: nic_cap axi_interface: 1
[    3.125673] syn1588: nic_cap second_timestamper: 0
[    3.125676] syn1588: Found MAC version 3160, build 876
[    3.125676] syn1588: Found MAC with timestamper
[    3.125678] syn1588nic: Using burst lenght: 16
[    3.125682] syn1588: read permanent HW MAC address 00:1e:c0:85:de:2b
[    3.125684] syn1588: using MII managment data clock 500 kHz (div.: 50).
[    3.126477] syn1588: detected PHY (0x00221622) with ID 0x00.
[    3.127281] syn1588: NIC_DEV_INIT: Initializing 64 TX DMA buffers of size 2048
and startoff 0.
[    3.127392] syn1588nic: detected syn1588(R) PCIe/Dual NIC revision 2.x
(eth0/clockm0).
[    3.127393] syn1588: pci rev id: 3
[    3.127393] syn1588nic: preparing device file.
[    3.133239] syn1588nic: device file syncD0 setup (minor #57) for device handle
0x00000000d2073898 complete.
[    3.133240] syn1588nic: PCI_DEV_INIT: This is a syn1588(R) PCIe/Dual NIC Rev 2,
programming DAC
[    3.530267] syn1588nic 0000:02:00.0 enp2s0: renamed from eth0
[  109.004254] syn1588nic: file pointer already initialized! 0000000091022846
[  109.004256] syn1588nic: created new virtual device for PID 1638.
[  109.007948] syn1588nic: unlinked virtual device for PID 1638.
```

```
[  109.007949] syn1588nic: disabling all IRQs.
[  109.007949] syn1588nic: unregistered IRQ callback function.
[  123.297460] syn1588nic: file pointer already initialized! 0000000091022846
[  123.297461] syn1588nic: created new virtual device for PID 1647.
[  139.615453] syn1588nic: unlinked virtual device for PID 1647.
[  139.615455] syn1588nic: disabling all IRQs.
[  139.615457] syn1588nic: unregistered IRQ callback function.
[  334.576659] syn1588nic: file pointer already initialized! 0000000091022846
[  334.576661] syn1588nic: created new virtual device for PID 1743.
[  352.017306] syn1588nic: unlinked virtual device for PID 1743.
[  352.017309] syn1588nic: disabling all IRQs.
[  352.017310] syn1588nic: unregistered IRQ callback function.
```

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.
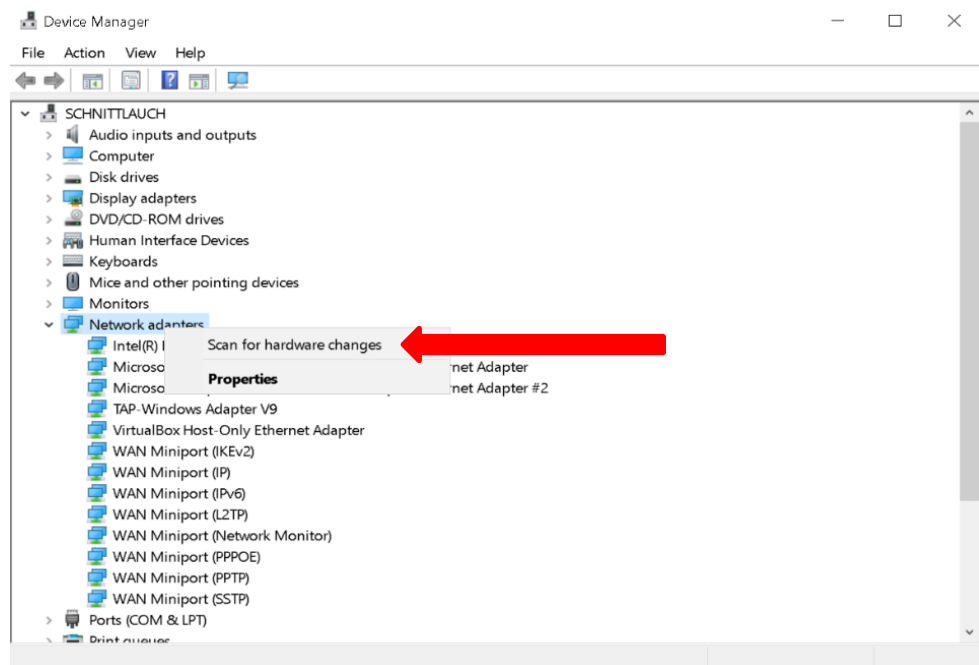

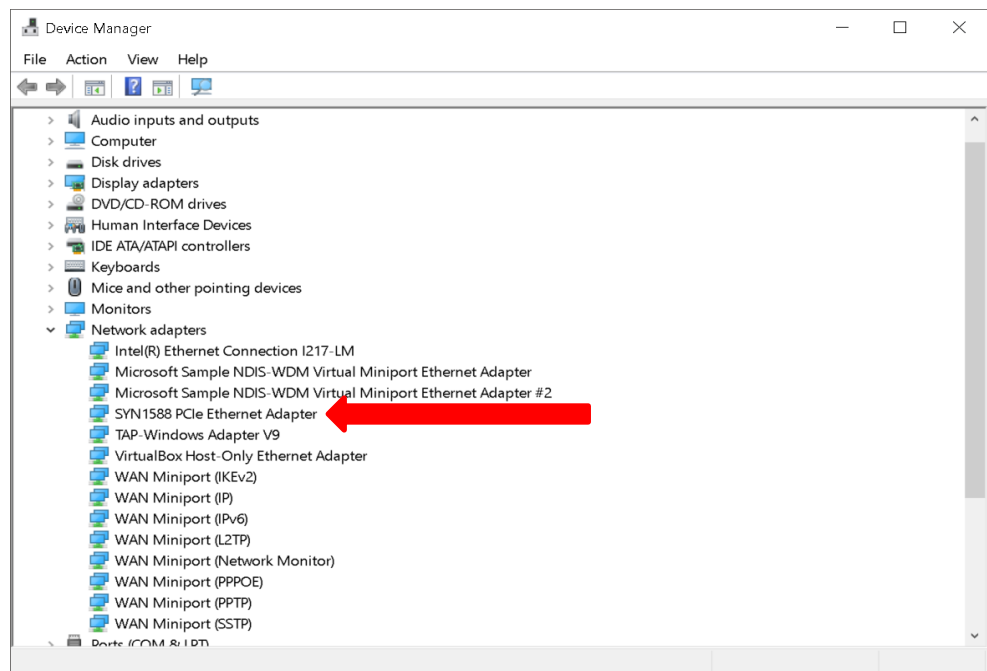**Figure 2: Device manager: scan for network adapters**

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.
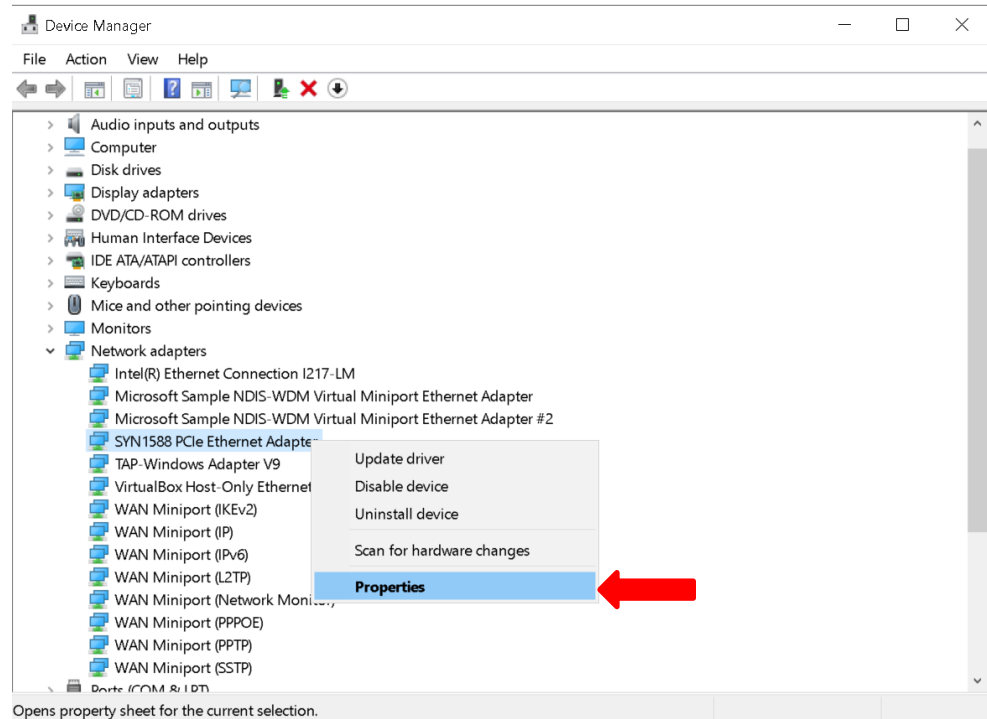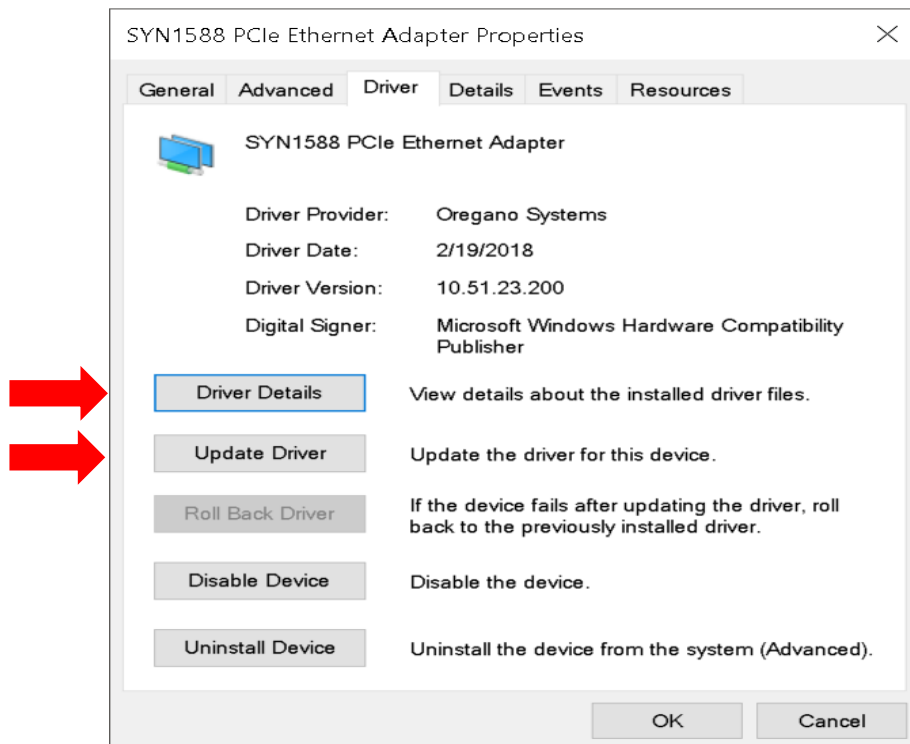

Figure 2.2: Device manager: syn1588® PCIe NIC

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
============  ======================  ============  ==========
syn1588       SYN1588 PCIe Ethernet  Kernel        2/19/2018 10:51:12 AM
```

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.

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    ] [p1.clock     ]  Spike Path: Init with ival
0, buffer size 16
2022-01-05 15:48:48.845778 [INFO    ] [p1.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    ] [p1.engine    ]  Settings: ClockId
00:1e:c0:ff:fe:85:de:2b
2022-01-05 15:48:48.848112 [INFO    ] [p1.engine    ]  Settings: Prio1 128
ClkClass 255 clkAccuracy 39 clkVariance 65535
2022-01-05 15:48:48.848119 [INFO    ] [p1.engine    ]  Settings: Prio2 128 Domain
0
2022-01-05 15:48:48.848235 [INFO    ] [p1.network.mcport]  SIOCSHWTSTAMP: tx_type
1 requested, got 1; rx_filter 0 requested, got 12
2022-01-05 15:48:48.848245 [INFO    ] [p1.network.mcport]  Activated
SO_TIMESTAMPING hardware
2022-01-05 15:48:48.848340 [INFO    ] [p1.network.mcport]  SIOCSHWTSTAMP: tx_type
1 requested, got 1; rx_filter 0 requested, got 12
2022-01-05 15:48:48.848349 [INFO    ] [p1.network.mcport]  Activated
SO_TIMESTAMPING hardware
2022-01-05 15:48:48.848599 [INFO    ] [p1.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    ] [p1.clock     ]    with ClockId
00:1e:c0:ff:fe:85:de:2b
2022-01-05 15:48:48.848742 [INFO    ] [p1.clock     ]  Clk: Resetting servos
2022-01-05 15:48:48.848748 [INFO    ] [p1.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 lSync 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

One does not need to write software in order to set actions in case of an active interrupt. One may simply use the "wait" command of the syn1588 utility to wait for such an interrupt event. Of course, one can write its own interrupt service routine. Oregano Systems will provide you the required C header files. In the following listing you'll find a brief example of such an interrupt routine.

```c
#include "syn1588_ifc.h"
#include "stdio.h"

int main() {

  Syn1588Ifc syn;
  // register Event0 input for SMA port 0
  // set IO matrix to allow Event0 input being processed
  syn.writeReg(Syn1588Ifc::IOMATRIX, (uint32_t) 0x9 << 16);

  // activate interrupt for Event0 (bit 10)
  syn.registerEvent( (uint32_t) 1 << 10);
  // activate event input
  syn.setBit(Syn1588Ifc::EVENTCTRL, 1);

  // wait for interrupt, timeout of 3 sec
  uint32_t src = syn.waitForEvent(3000);

  if (src) {
    printf("Event timestamp: %d nsec\n", syn.readReg(Syn1588Ifc::EVENTTIME0_M));
    printf("Event timestamp: %d sec\n", syn.readReg(Syn1588Ifc::EVENTTIME0_H));
  }
  else printf("Timeout\n");
  return 0;
}
```

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.

Franzosengraben 8
A-1030 Vienna
Austria
http://oregano.at
contact@oregano.at