Universal Configuration Manager

Quickstart Guide

Product Info	
Product Manager	Sven Meier
Author(s)	Sven Meier
Reviewer(s)	-
Version	1.1
Date	07.02.2020

Copyright Notice

Disclaimer

The information available to you in this document/code may contain errors and is subject to periods of interruption. While NetTimeLogic GmbH does its best to maintain the information it offers in the document/code, it cannot be held responsible for any errors, defects, lost profits, or other consequential damages arising from the use of this document/code.

NETTIMELOGIC GMBH PROVIDES THE INFORMATION, SERVICES AND PRODUCTS AVAILABLE IN THIS DOCUMENT/CODE "AS IS," WITH NO WARRANTIES WHATSOEVER. ALL EXPRESS WARRANTIES AND ALL IMPLIED WARRANTIES, INCLUDING WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF PROPRIETARY RIGHTS ARE HEREBY DISCLAIMED TO THE FULLEST EXTENT PERMITTED BY LAW. IN NO EVENT SHALL NETTIMELOGIC GMBH BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL AND EXEMPLARY DAMAGES, OR ANY DAMAGES WHATSOEVER, ARISING FROM THE USE OR PERFORMANCE OF THIS DOCUMENT/CODE OR FROM ANY INFORMATION, SERVICES OR PRODUCTS PROVIDED THROUGH THIS DOCUMENT/CODE, EVEN IF NETTIMELOGIC GMBH HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

IF YOU ARE DISSATISFIED WITH THIS DOCUMENT/CODE, OR ANY PORTION THEREOF, YOUR EXCLUSIVE REMEDY SHALL BE TO CEASE USING THE DOCUMENT/CODE.

Overview

NetTimeLogic's Universal Configuration Manager is an open source solution for configuring and supervising all NetTimeLogic's IP cores. It allows to configure the configuration registers of the individual cores and allows to supervise the status of the cores. Some cores allow real-time monitoring of status information and can show this in a graph (e.g. PTP). The connection between the host and the target is done via UART (often USB USART) or Ethernet and has its own protocol running on it. The GUI can detect all instantiated cores in the systems and their AXI base addresses at runtime and will provide tabs for the individual cores. The solution consists of two parts, an FPGA part and a GUI part. The FPGA part allows the access to the registers, provides information about the cores in the system and makes a protocol and interface conversion between UART or Ethernet and AXI. The GUI part is the frontend for the user, it abstracts the UART or Ethernet interface and the individual registers and does the data presentation. Multiple instances of the tool can run in parallel and allow configuration and monitoring of multiple systems. Multiple instances of the same core in a system are handled and can be configured individually.

Key Features:

- Open Source GUI
- HW/SW co-solution
- Configuration of the cores via UART or Ethernet
- Status monitoring of the cores via UART or Ethernet
- Register access to all AXI addresses in the system (also 3rd party)
- Auto detection of available cores and base addresses
- Proprietary protocol for the UART connection, can also be done from a terminal
- Multiple systems and multiple cores in a system support
- Loading of configurations from a file (plain ASCII)
- Logging of all accesses
- QT based

Revision History

This table shows the revision history of this document.

Version	Date	Revision
1.0	08.08.2018	First release
1.1	07.02.2020	Added note to install WinPcap or Npcap in WinPcap API

Table 1: Revision History

Content

Definitions

Definitions	
Config	A set of parameters that can be stored in a file
Tap	A part of the GUI which shows some parts of the design

Table 2: Definitions

Abbreviations

Abbreviations	
AXI	AMBA4 Specification (Stream and Memory Mapped)
FPGA	Field Programmable Gate Array
IRQ	Interrupt, Signaling to e.g. a CPU
PPS	Pulse Per Second
TS	Timestamp
CLK	Clock
CC	Counter Clock
UCM	Universal Configuration Manager
VHDL	Hardware description Language for FPGA's

Table 3: Abbreviations

1 Introduction

1.1 Context Overview

NetTimeLogic's Universal Configuration Manager is meant as a solution for configuring and supervising all NetTimeLogic's IP cores. It allows to configure the configuration registers of the individual cores and allows to supervise the status of the cores. The connection between the host and the target is done via UART (often USB UART) or Ethernet (depending on the hardware setup) and has its own protocol running on it. The solution consists of two parts, an FPGA part and a GUI part. The FPGA part allows the access to the registers, provides information about the cores in the system and makes a protocol and interface conversion between UART and AXI. The GUI part is the frontend for the user, it abstracts the UART interface and the individual registers and does the data presentation. Multiple instances of the tool can run in parallel and allow configuration and monitoring of multiple systems. Multiple instances of the same core in a system are handled and can be configured individually.

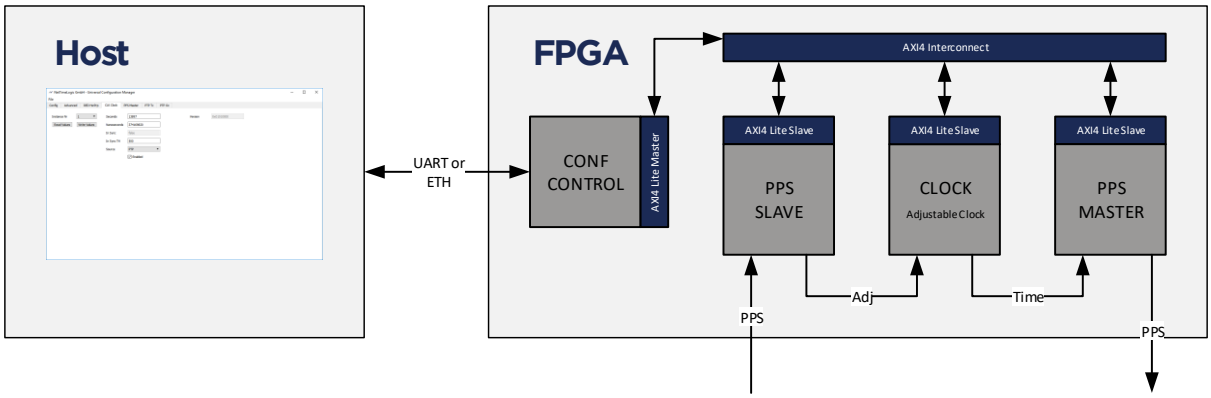

Figure 1: Context Block Diagram

1.2 Function

The Universal Configuration Manager allows to read and write registers via an FPGA configuration block which converts between a proprietary UART protocol and AXI. It first tries to connect to the configuration core and asks for a specific acknowledge. If it received the expected acknowledge it reads the configuration ROM in the configuration core to get the information about the instantiated cores like base address and instance number. This register map is then shown and the

individual tabs of the instantiated cores are shown. Then in the individual tabs the registers can be written and read. The registers are shown as fields with a meaningful value and therefore are abstracted from the individual addresses and bits.
For some of the cores also an auto refresh functionality is available which polls the registers in a fixed interval (1 s)

2 Getting Started

Note: Make sure, that you installed WinPcap (https://www.winpcap.org/) or Npcap (https://nmap.org/npcap/). When using Npcap make sure you install Npcap in the WinPcap Api mode.

2.1 Connect Device

2.1.1 UART

1. Given that design contain a Conf IP Core from NetTimeLogic in UART mode, connect your device via UART (or in most cases USB to UART) to the PC where you run the Universal Configuration Manager
2. Check the UART Port the device has enumerated (COMx)
3. Start the Universal Configuration Manager

2.1.2 Ethernet

1. Given that design contain a Conf IP Core from NetTimeLogic in Ethernet mode, connect your device via an Ethernet cable to the PC where you run the Universal Configuration Manager (can also go over a Switch but not router)
2. Remember the configured IP of the device
3. Start the Universal Configuration Manager

2.2 Config Tap

2.2.1 Connect to device via UART

1. When started the Universal Configuration Manager checks for all available COM ports and lists them

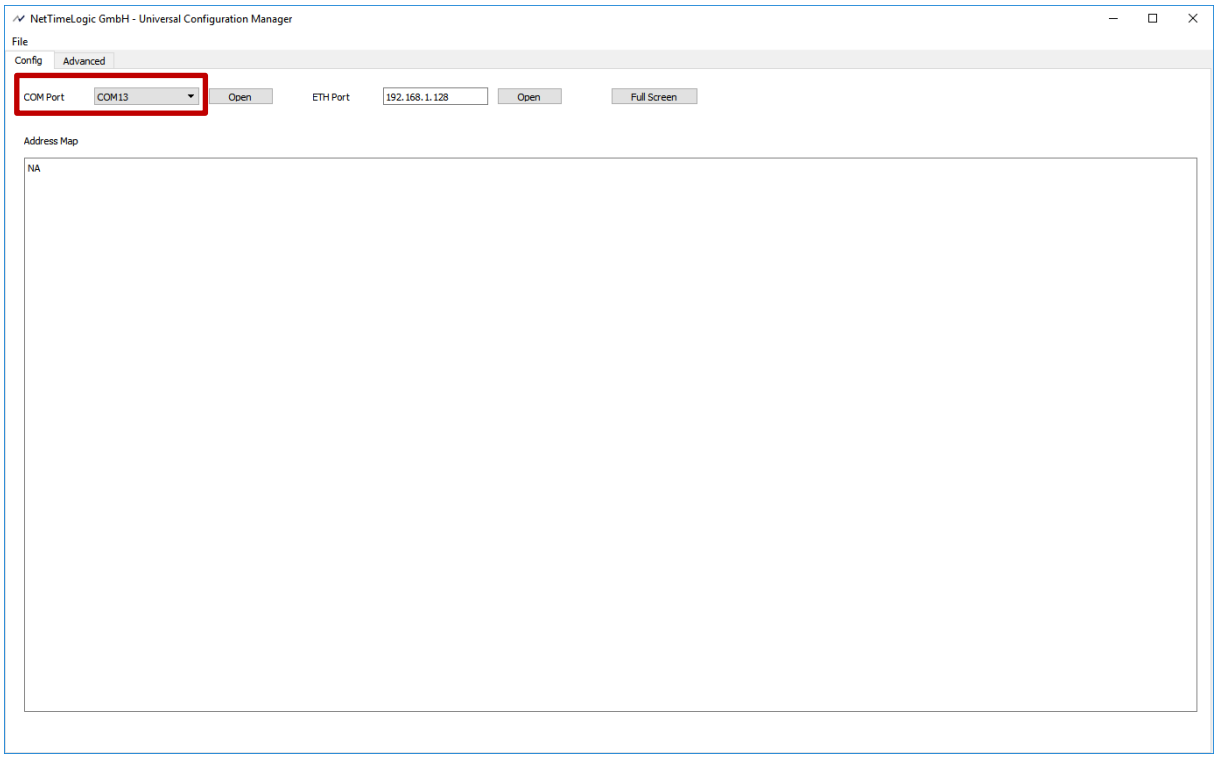

Figure 2: UART Ports

2. Select the correct COM port from the list and press “Open” (this will start the connection with the device and checks if a known device is connected).

2.2.2 Connect to device via Ethernet

1. The Universal Configuration Manager can also connect to a device via Ethernet. For this on the device side the IP has be known

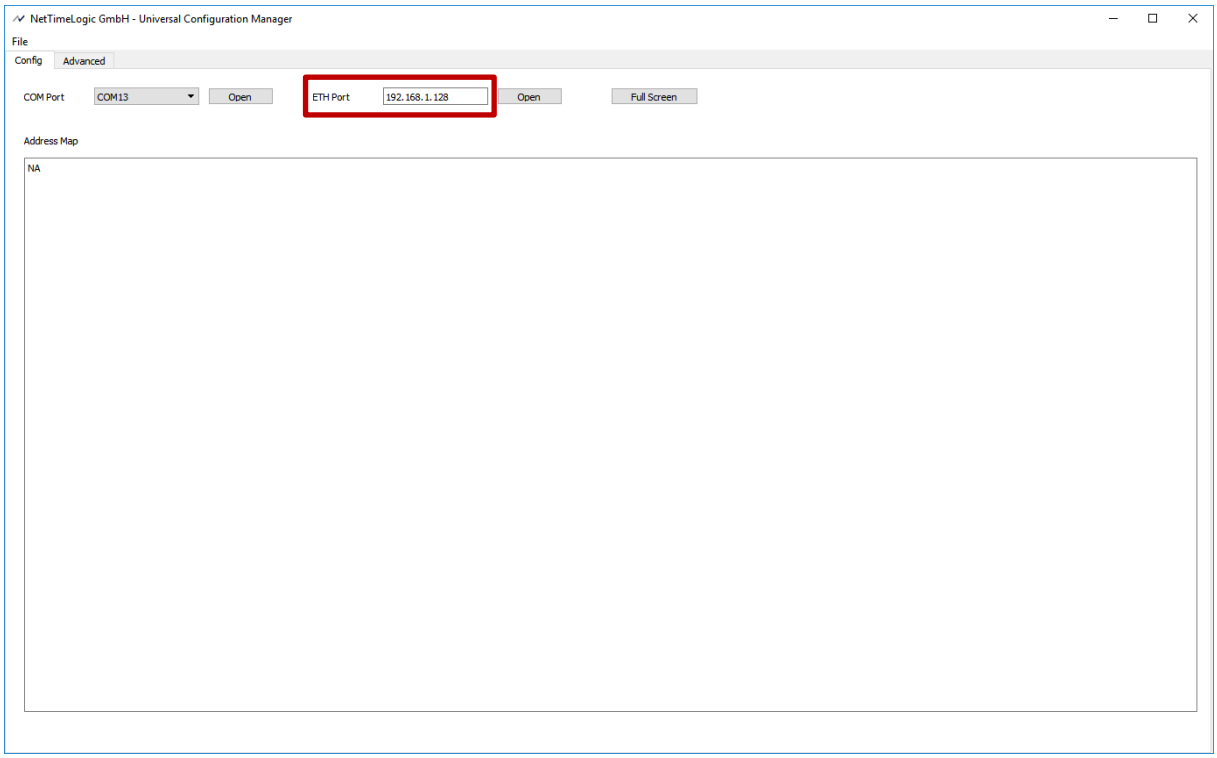

Figure 3: Ethernet Ports

2. Enter the IP of your device and press “Open” (this will start the connection with the device and checks if a known device is connected).

2.2.3 Opened Device and Taps

1. During opening, the Universal Configuration Manager will check for all available NetTimeLogic IP cores in the design and will list them together with their base addresses and types (the list depends on your cores, the figure below is just an example you may have less, more or different cores instantiated and different base addresses)

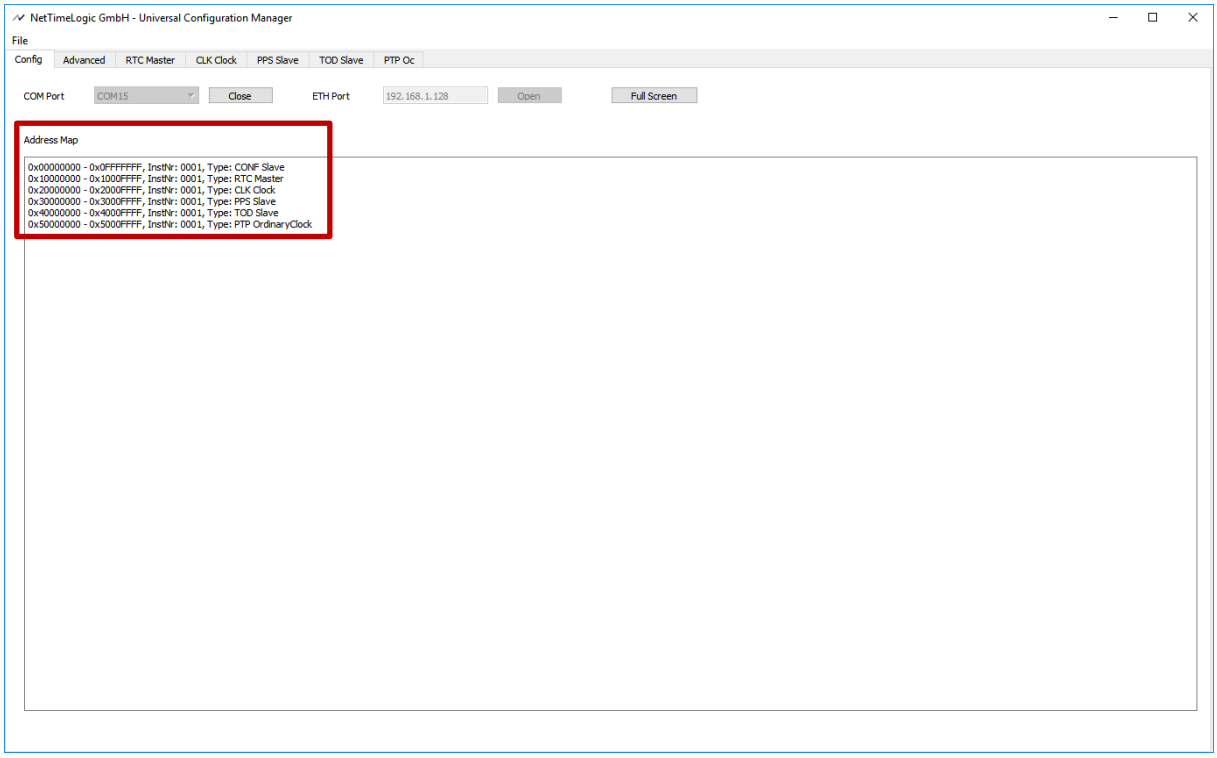

Figure 4: Available Cores

2. Now the individual cores can be configured in their respective Taps (the list depends on your cores, the figure below is just an example you may have less, more or different cores instantiated)

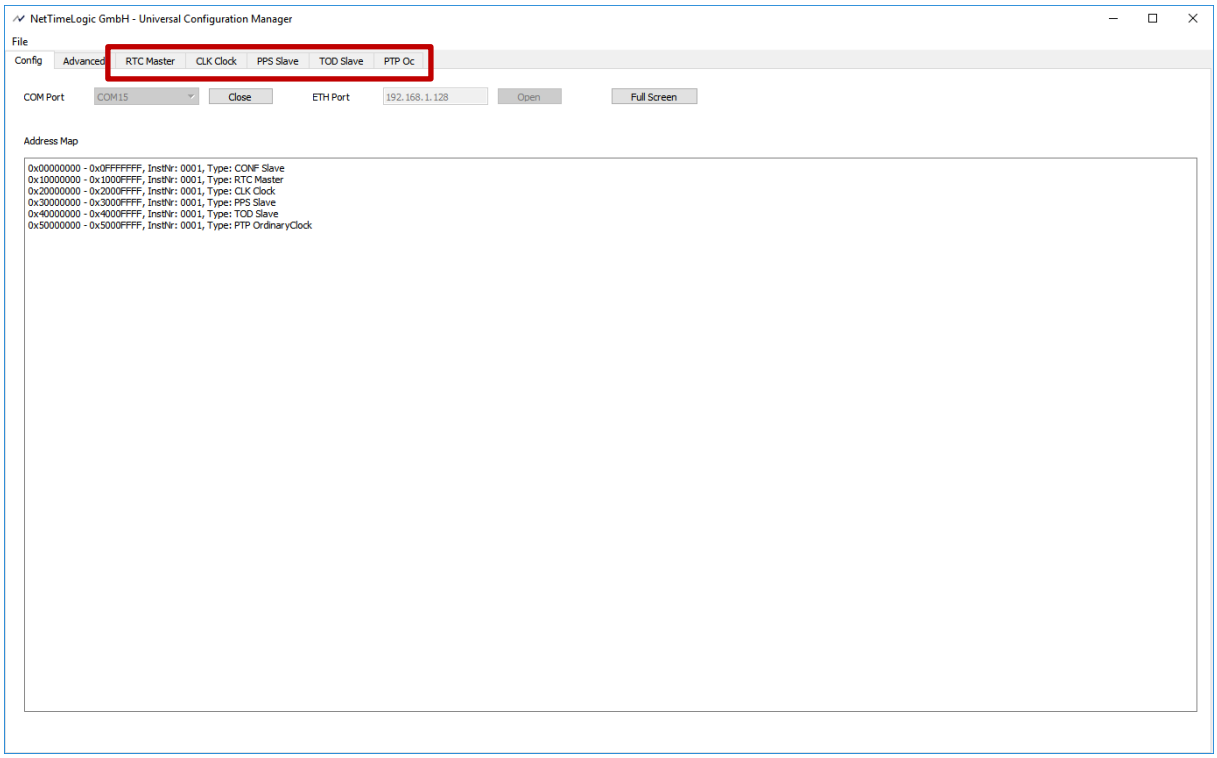

Figure 5: Available Taps

2.3 Advanced Tap

2.3.1 Log

1. When connected a log is shown of all commands and events that happen (some errors might appear when some registers are not available due to IP core configurations)
2. The log can be cleared or be saved to a file, for that a file name has to be entered in the “Save Log File” field.

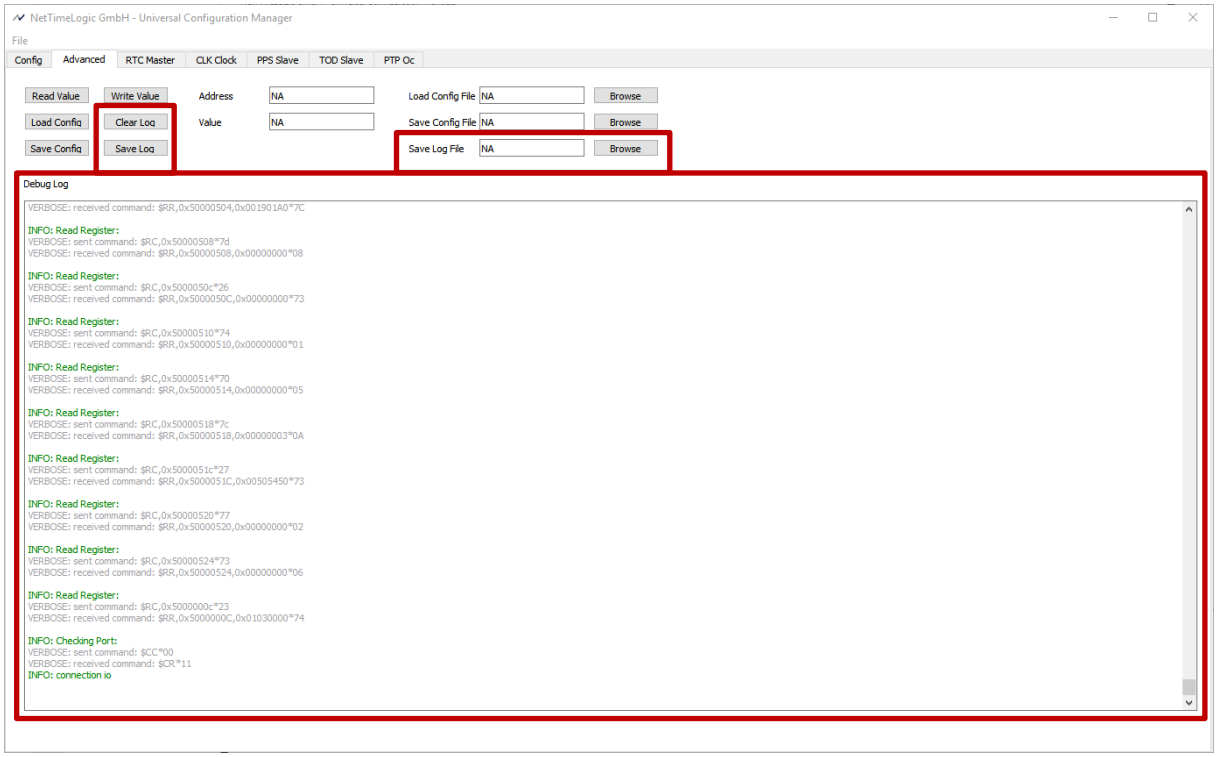

Figure 6: Debug Logs

2.3.2 Access to individual registers

1. It allows to access also registers individually (also to thirdparty cores not listed in the Address Map) by first entering the register address with base address in hex into the “Address” field and the pressing “Read Value” or “Write Value” buttons. For a write a value in hex has to be entered to the “Value” field as well. Check the IP cores register map to see which registers are available.
   WARINING if you enter an address range which is not available it will stall and the FPGA needs a reset as well as a reconnection from the Universal Configuration Manager. This is because AXI has per definition no timeout and will wait for the access to complete forever (which will never happen).

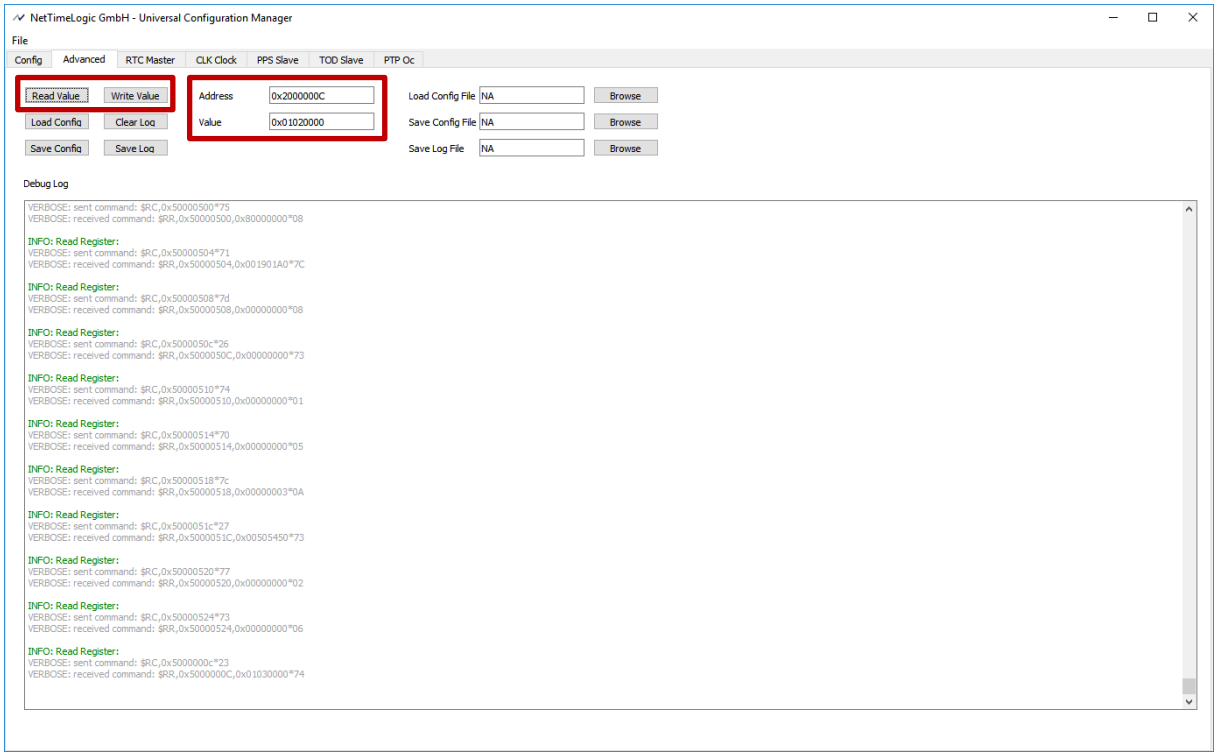

Figure 7: Access individual registers

2.3.3 Save and Open Config

1. You can load a configuration from a file rather than configuring each core again individually, for this enter a configuration file in the “Load Config File” field first and then press “Load Config”.
2. Once configured a popup will occur that it has completed, now the cores are configured.

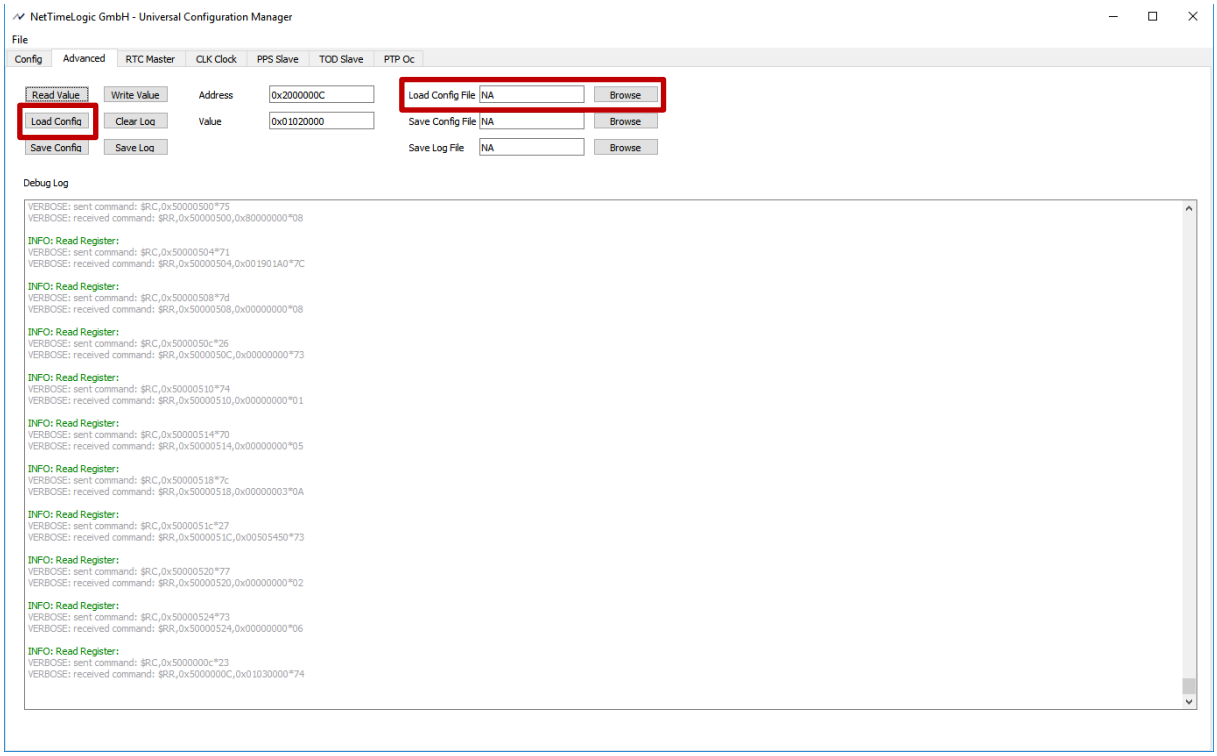

Figure 8: Load configuration

3. To save a configuration, first enter a configuration file in the “Save Config File” field
4. Then press “Clear Log”. This is important since it will just extract the write commands from the log.

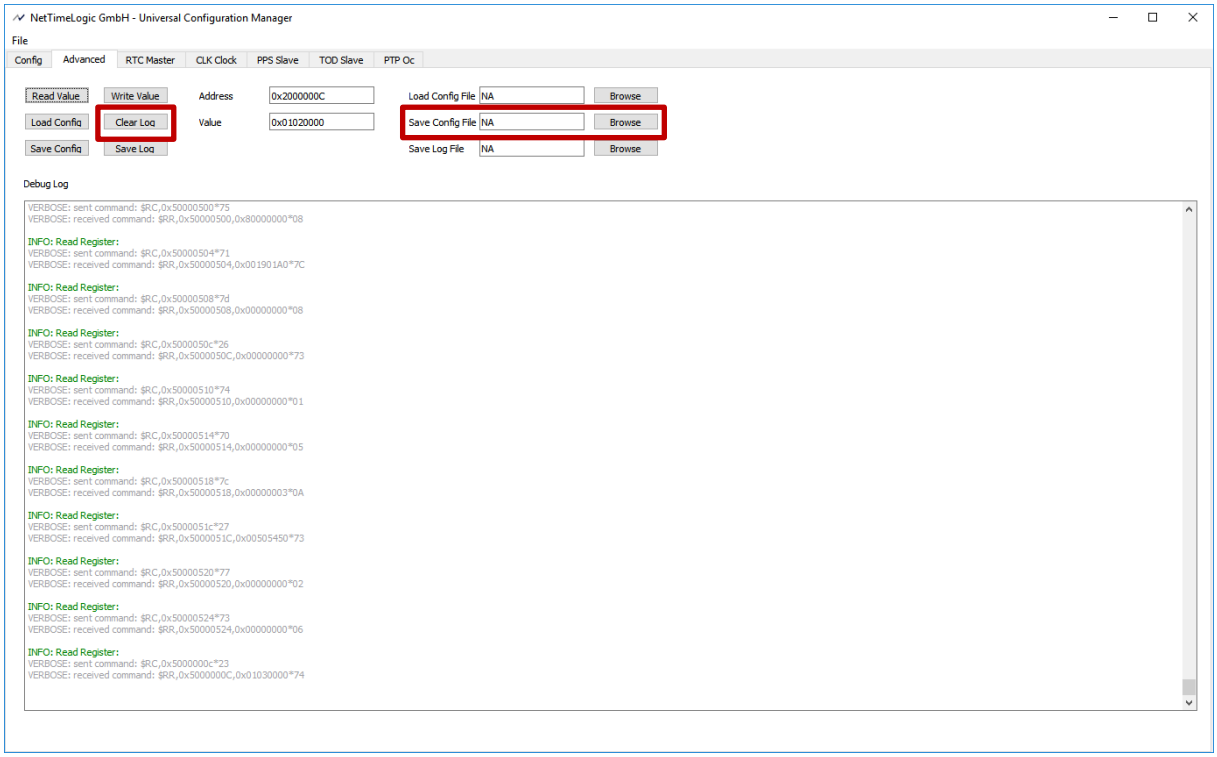

Figure 9: Save configuration file

5. Go to the individual Core taps change the configurations you want and press “Write Values”, do this for all cores you want to configure.

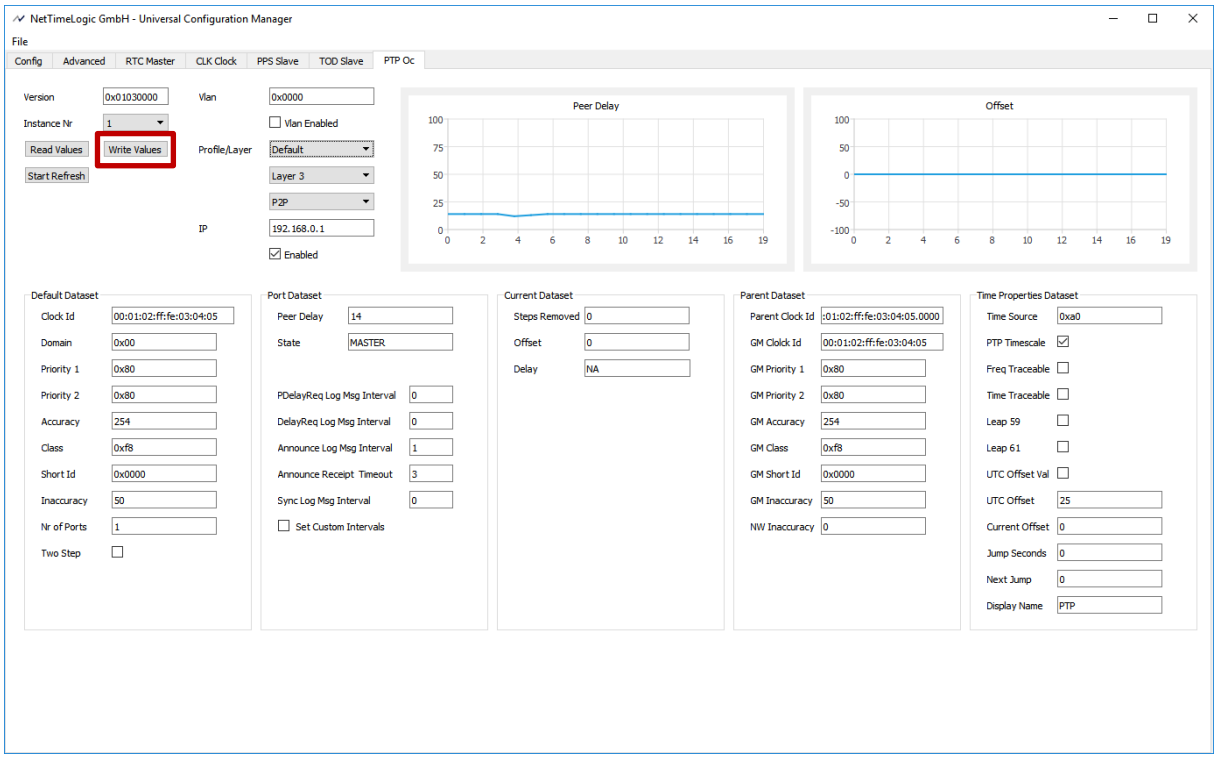

Figure 10: Write configuration

6. Go back to the “Advanced” tap and press “Save Config”

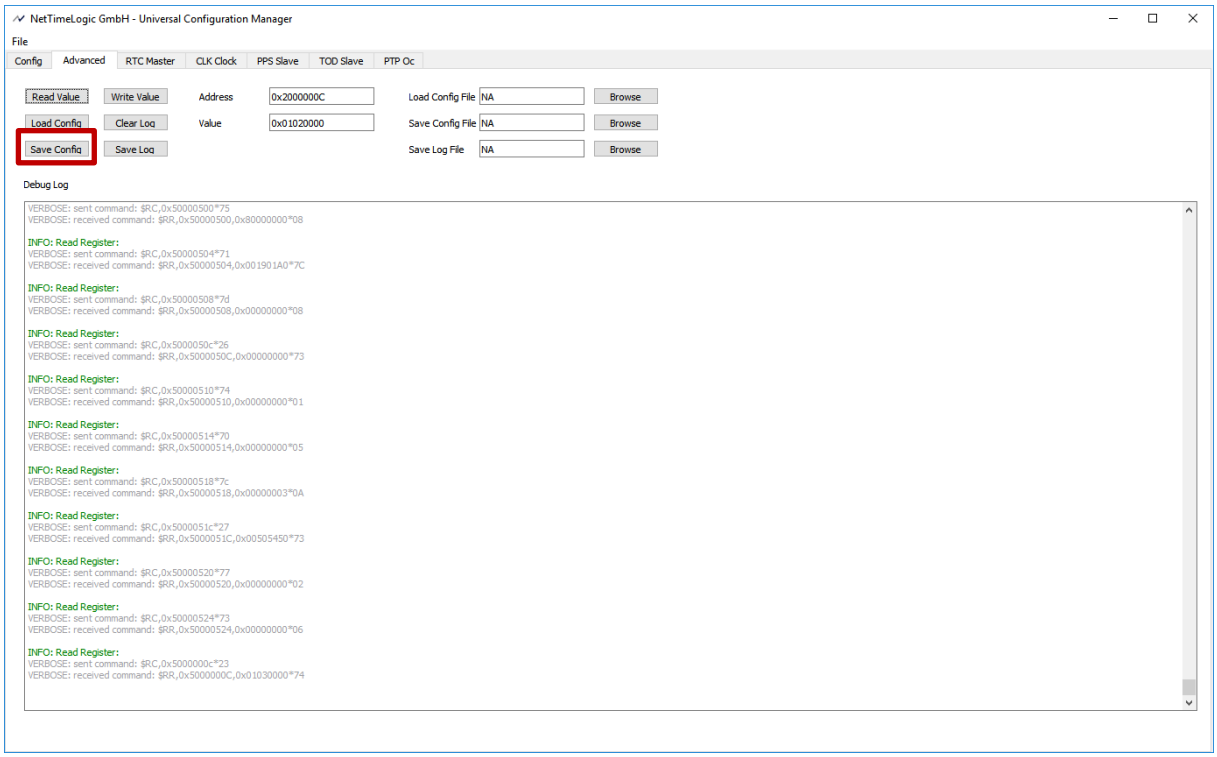

Figure 11: Save configuration

7. This will create a configuration file, which can be opened in a text editor as well and modified there since the config is in ASCII text:
   E.g.
   $WC,0x20000008,0x00000004

   ⇨ $WC: this is the write command
   ⇨ 0x20000008: is the address
   ⇨ 0x00000004: is the value

2.4 Core Taps

In the individual core taps the configurable fields are shown, some of them are read only in the core and a write will not have an effect and will be overwritten by the next read in the field.

2.4.1 Choose Core Instance

Each core can be instantiated multiple time in the design therefore the instance has to be chosen.

1. Choose the instance of the core to work on. Per default instance 1 is chose which is the normal case if you only have one core instantiated

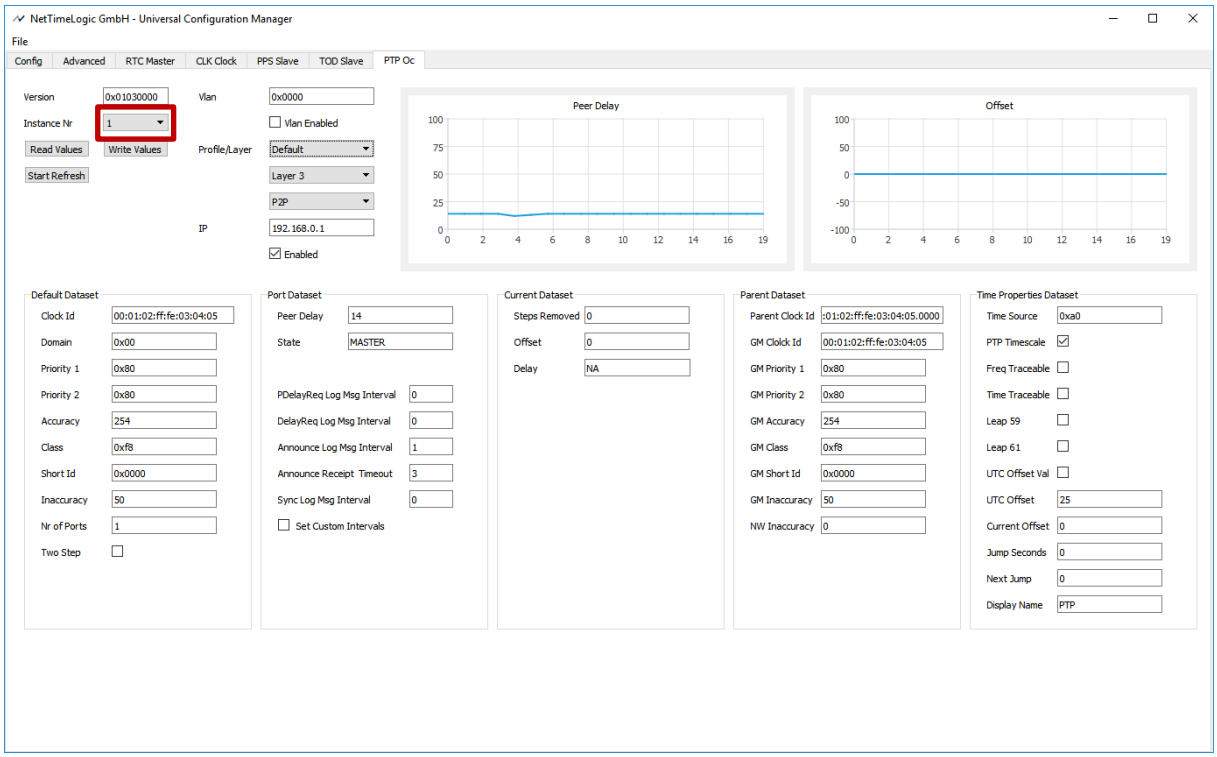

Figure 12: Load configuration

2.4.2 Read Configuration

2. Always do a read first to get the current configuration by pressing the “Read Value” button

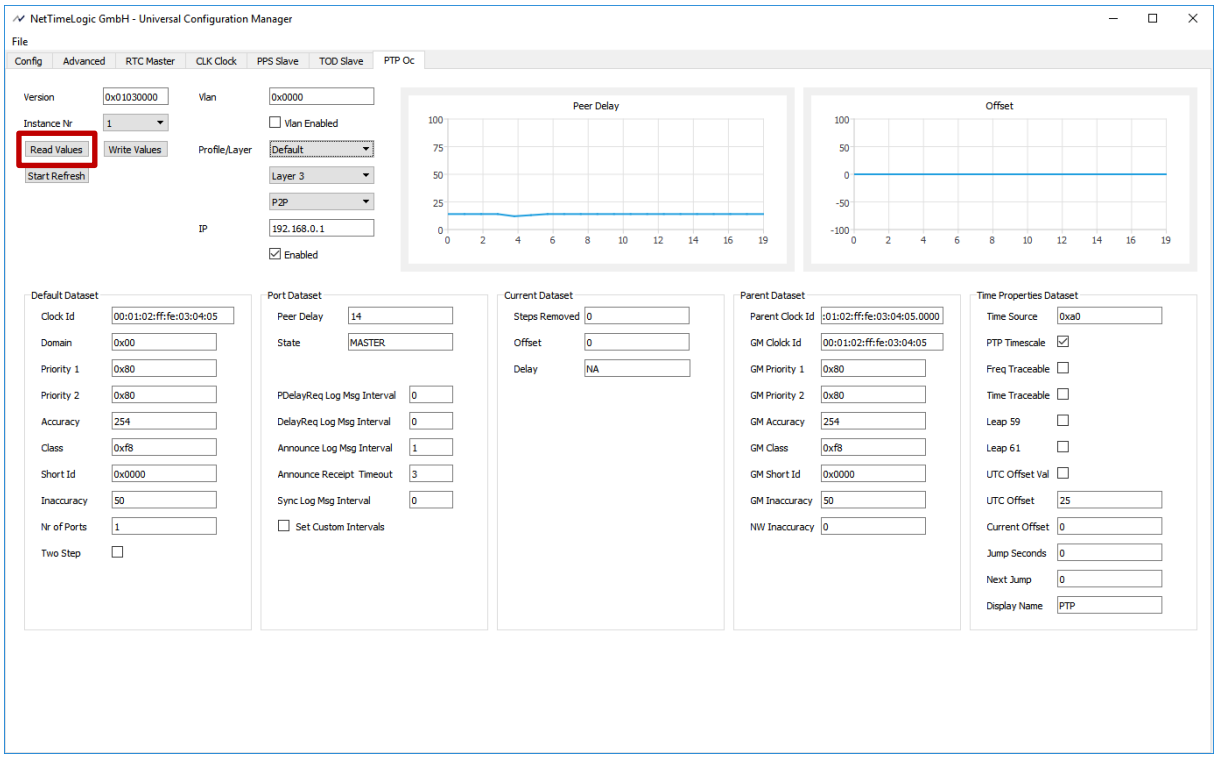

Figure 13: Read configuration

2.4.3 Write Configuration

1. Change the configuration and press the “Write Value” button when done. The write will be done and followed by an immediate read

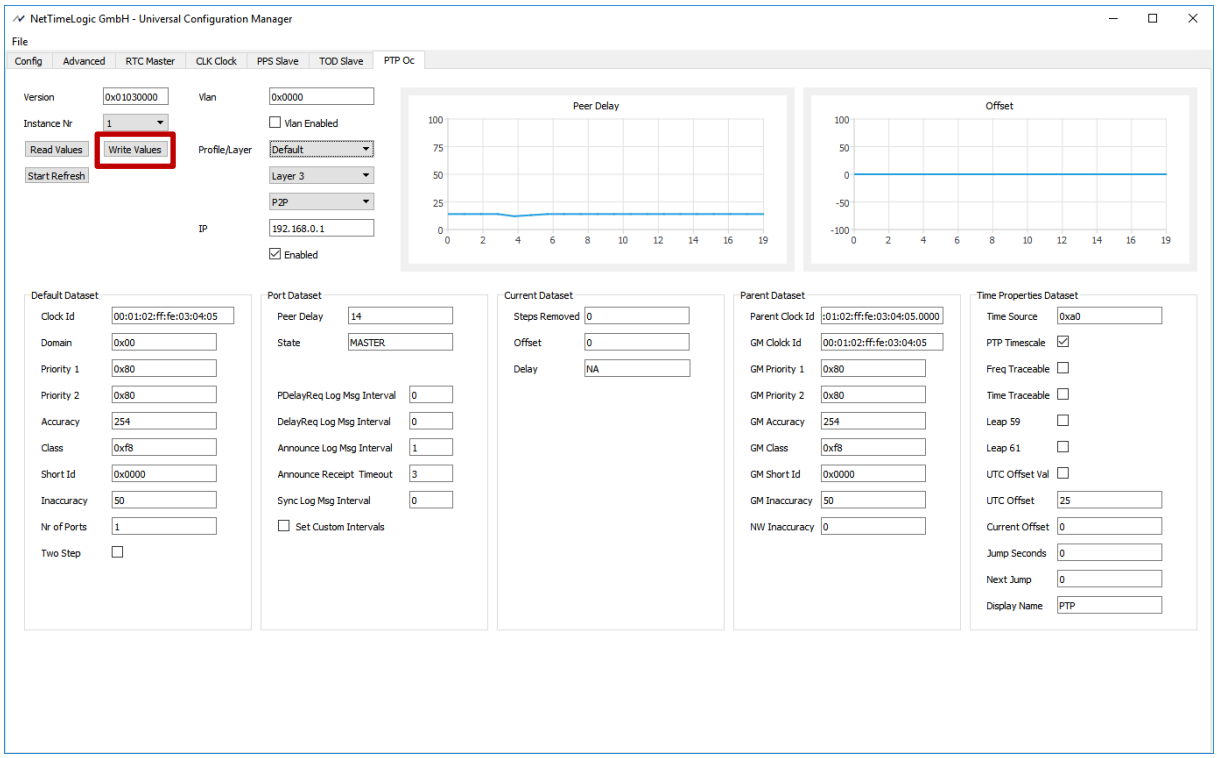

Figure 14: Write configuration

2.4.4 Auto Refresh

1. Some of the Core taps have also an auto refresh functionality which reads all the registers of the core every second and will also start to draw graphs on specific values. Whenever the “Start Refresh” button is pressed the graphs are cleared and will start again.

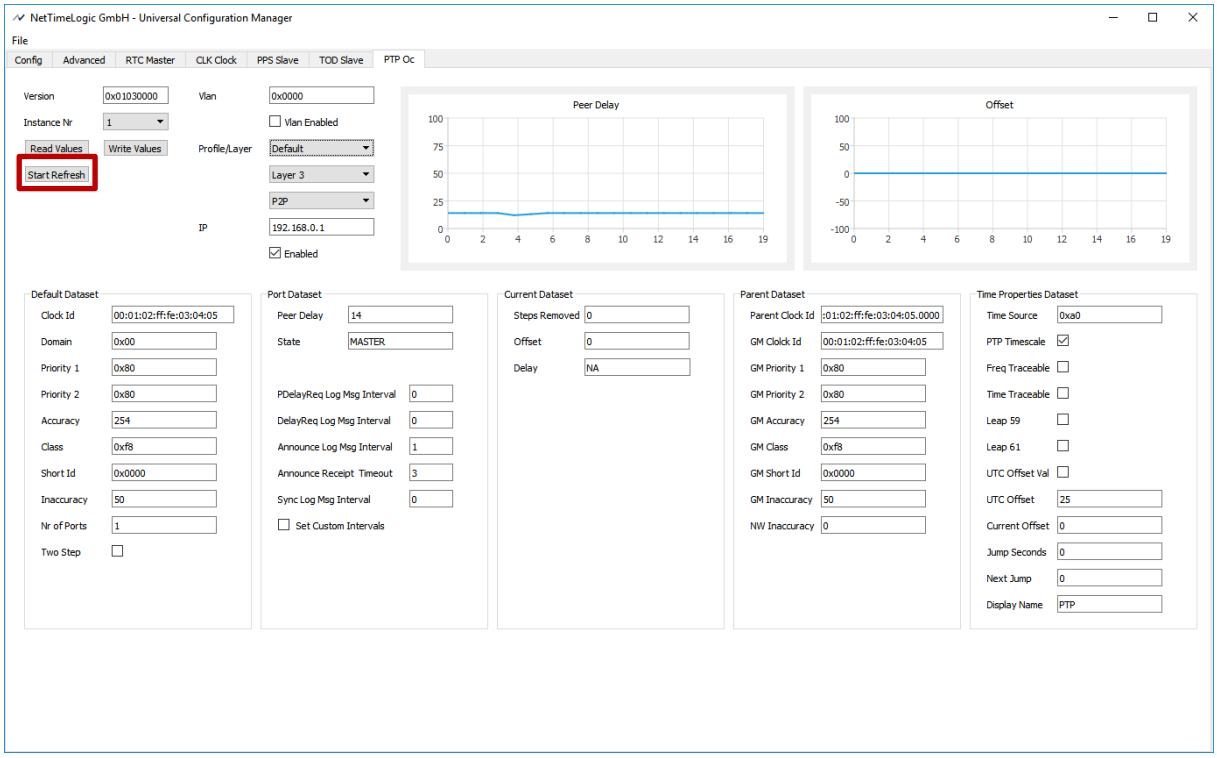

Figure 15: Start Refresh

2. When pressed “Start Refresh” is pressed the button will turn into “Stop Refresh” which can be pressed to stop auto refreshing. During an auto refresh reading and writing is disabled.

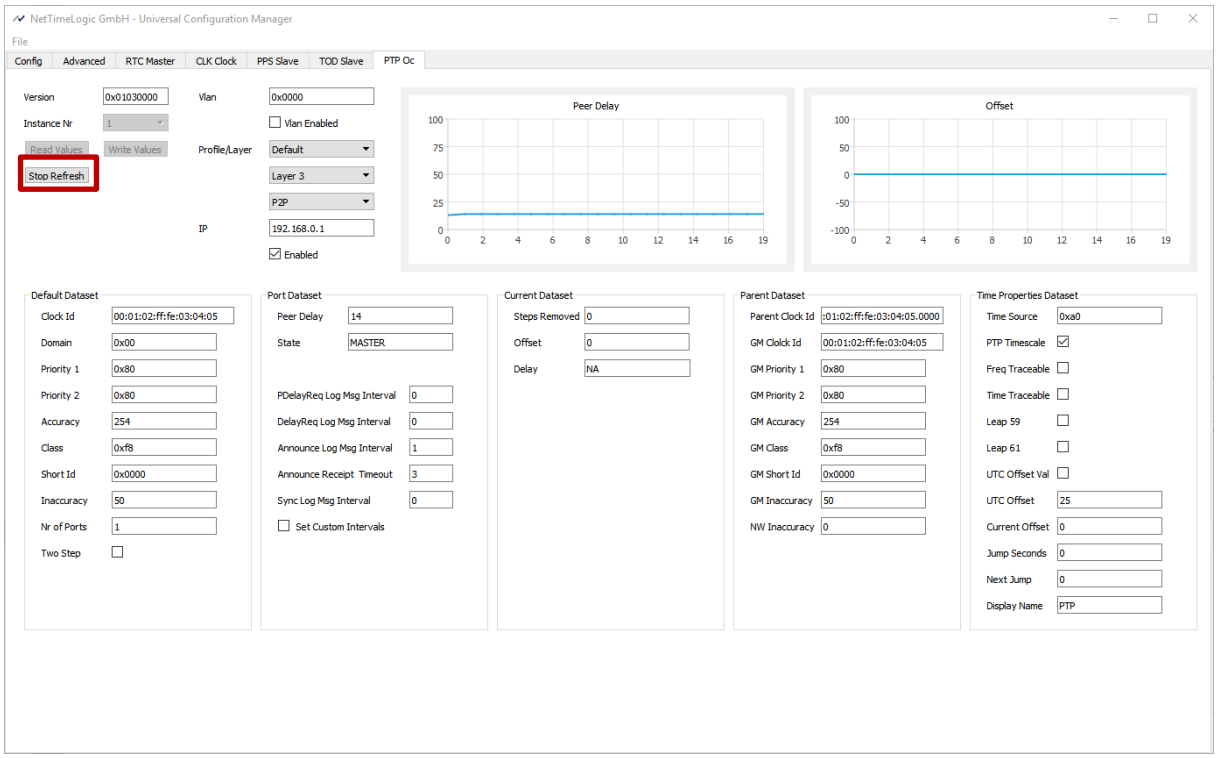

Figure 16: Stop Refresh

A List of tables

B List of figures