

# 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**

Copyright © 2025 NetTimeLogic GmbH, Switzerland. All rights reserved. Unauthorized duplication of this document, in whole or in part, by any means, is prohibited without the prior written permission of NetTimeLogic GmbH, Switzerland.

All referenced registered marks and trademarks are the property of their respective owners

## 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 PROD-UCTS 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 PARTIC-ULAR 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 DOCU-MENT/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 3<sup>rd</sup> 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

1 I	INTRODUCTION	7
1.1 (	Context Overview	7
1.2	Function	7
2 (	GETTING STARTED	8
2.1 (	Connect Device	8
2.1.1	UART	8
2.1.2	Ethernet	8
2.2 (	Config Tap	9
2.2.1	Connect to device via UART	9
2.2.2	Connect to device via Ethernet	10
2.2.3	Opened Device and Taps	11
2.3	Advanced Tap	13
2.3.1	Log	13
2.3.2	Access to individual registers	14
2.3.3	Save and Open Config	15
2.4 (	Core Taps	18
2.4.1	Choose Core Instance	18
2.4.2	Read Configuration	19
2.4.3	Write Configuration	19
2.4.4	Auto Refresh	21



# Definitions

Definitions	
Config	A set of parameters that can be stored in a file
Тар	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
СС	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 meaning-ful 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 (<u>https://www.winpcap.org/</u>) or Npcap (<u>https://nmap.org/npcap/</u>). When using Npcap make sure you install Npcap in the WinPcap Api mode.

## 2.1 Connect Device

#### 2.1.1 UART

- 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

✓ NetTimeLo	gic GmbH - Universal Con	figuration Manager		-	×
File					
Config Adv	anced				
COM Port	COM13 •	Open	ETH Port	192.168.1.128 Open Full Screen	
Address Map					
NA					
·					_

#### 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

✓ NetTim	neLogic GmbH - Universal Configuration Manager	-	×
File			
Config	Advanced		
COM Port	t COM13  Open ETH Port 192.168.1.128 Open Full Screen Map		
NA			

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)

✓ NetTimeLogic GmbH - Universal Configuration Manager	-	×
File		
Config Advanced RTC Master CLK Clock PPS Slave TOD Slave PTP Oc		
COM Port COM15  Close ETH Port 192.168.1.128 Open Pull Screen		
Address Map		 _
0x0000000-0x0FFFFFF, Institiv:0001, Type: ICUM Save 0x1000000-0x200FFFF, Institiv:0001, Type: ICU Clock 0x2000000-0x200FFFF, Institiv:0001, Type: ICU Clock 0x2000000-0x200FFFF, Institiv:0001, Type: ICU Clock 0x5000000-0x500FFFF, Institiv:0001, Type: ICU Clock		

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)



🛷 NetTi	meLogic GmbH - Universal Configuration Manager	-	$\times$
File			
Config	Advanced RTC Master CLK Clock PPS Slave TOD Slave PTP Oc		
COM Po	rt COM15 🝸 Close ETH Port 192.168.1.128 Open Full Screen		
Address	Map		
0x000	0000 - 0x0FFFFFF, InstWr: 0001, Type: CONF Slave		
0x100 0x200	0000 - 0x1000FFFF, InstW: 0001, Type: RTC Master 0000 - 0x200FFFF, InstW: 0001, Type: RTC Master		
0x300	1000 - 0.03000FFFF, Isstw: 0001, Type: PPS Slave		
0x400	0000 - 0x5000FFFF, Instity: 0001, Type: FTD OrdinaryGlock		

Figure 5: Available Taps



## 2.3 Advanced Tap

#### 2.3.1 Log

- 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.

NetTimeLogic GmbH - Universal Configuration Manager	>
Read Value     Write Value     Address     NA       Load Config     Clear Log     Value     NA       Save Config     Save Log     Save Log     Save Log	P OC Load Config File NA Browse Save Config File NA Browse Save Log File NA Browse
Debug Log	
VERBOSE: received command: \$RR,0x5000504,0x001391A0*7C <b>INFO: Read Register:</b> VERBOSE: service command: \$RR,0x5000508*7d VERBOSE: received command: \$R0,0x5000500,0x0000000*08 <b>INFO: Read Register:</b> <b>INFO: Read Register:</b> VERBOSE:-received command: \$R0,0x5000500-73 VERBOSE:-received command: \$R0,0x5000500-75 VERBOSE:-received command: \$R0,0x50005000-75 VERBOSE:-received command: \$R0,0x50000000000000000000000000000000000	^
<b>INFO: Read Register:</b> VERBOGS: sent command: \$RC,0x5000510*74 VERBOGS: read command: \$R5,0x5000510,0x00000000*01 <b>VERDOS:</b> read Register: VERBORS: end register:	
VERBOSE: received command: \$RR,0x50000514,0x0000000°05 INFO: Read Register: VERBOSE: service command: \$RC,0x50000518/%C VERBOSE: received command: \$RR,0x50000518,0x0000003°0A	
INFO: Read Register: VERBOSE: sent command: \$RC,0x5000051c*27 VERBOSE: received command: \$RR,0x5000051C,0x00505450*73	
INFO: Read Register: VERBOSE: sent command: \$RC,0x50000520*77 VERBOSE: received command: \$RR,0x50000520,0x00000000*02	
INFO: Read Register: VERBOSE: sent command: \$RC,0x50000524*73 VERBOSE: received command: \$RR,0x50000524,0x000000000000	
INFO: Read Register: VERBOSE: sent command: \$RC,0x5000000c*23 VERBOSE: received command: \$RR,0x5000000C,0x01030000*74	
INFO: Checking Port: VERDOCE: service momend: \$CC*00 VERDOCE: received command: \$CR*11 INFO: connection io	

Figure 6: Debug Logs



#### 2.3.2 Access to individual registers

 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).

🚧 NetTimeLogic GmbH - Universal Configuration Manager	-	×
File		
Config Advanced RTC Master CLK Clock PPS Save TOD Slave PTP Oc		
Read Value Write Value Address 0x2000000C Load Config File INA Browse		
Load Comg Vaue UXUI22000 Save Comg He VA browse		
Save Config Save Log Save Log File NA Browse		
Debug Log		
VERBOSE: sent command: 4RC,0x50000500*75 VERBOSE: received command: 4RR,0x50000500,0x80000000*08		 ^
B#O: Read Register: VRB03E: text command: \$RC,0x5000059471 VRB03E: reviewd command: \$RV,0x50000594,0x001901A0*7C		
BF0 <sup>-0</sup> Read Register: VERBOSE: tracelyed command: \$RC,0x50005508,0x00000000°08 VERBOSE: received command: \$RR,0x50005508,0x00000000°08		
IPPO: Read Register:           VERBOSE: enclowed command: \$RC,0x5000050c,"a6         VERBOSE: received command: \$RR,0x5000050c,0x0000000*73         VERBOSE: received command: \$RR,0x5000050c,0x0000000*73		
UPC::         Read Register:           VERDOSE::         received command: \$RC,0x50000510,"74           VERBOSE::         received command: \$RR,0x50000510,0x0000000*01		
IP#O: Read Register:           VERBOSE: encolved command: \$RC,0x50000514,0x0000000°05           VERBOSE: received command: \$RR,0x50000514,0x0000000°05		
IDFO:         Read Register:           VERIBORS::         received command: \$RC,0x50000518,9x0000003*0A           VERBORS::         received command: \$RR,0x50000518,0x0000003*0A		
IP-0: Read Register:           VERBOSE: encoded command: \$RC,0x5000051cf*27           VERBOSE: received command: \$R2,0x5000051c,0x00505450*73		
IPF0: Read Register:           VERIBOSE: enclosed command: \$RC,0x50000520*77           VERIBOSE: received command: \$R2,0x50000520,0x0000000*02		
BPCP:         Read Register:           VERBOSE:         received command: \$RC,0x5000524,9x30000000005           VERBOSE:         received command: \$RR,0x5000524,0x000000006		
IPPC: Read Register:           VERBOSE: encoved command: \$RC,0x5000000;#23           VERBOSE: received command: \$R,0x5000000;,0x01030000*74		
		~

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.

✓ NetTimeLogic GmbH - Universal Configuration Manager File	- 0	×
Config Advanced RTC Master CLK Clock PPS Slave PTP Oc		
Read Value Write Value Address 0x2000000C Load Config File NA Browse		
Load Config Clear Log Value 0x01020000 Save Config File NA Browse		
Save Config Save Log File NA Browse		
Debug Log		
VERBOSE: sent command: \$RC,0x50000500*75 VERBOSE: received command: \$RC,0x50000500,0x80000000*08		^
IVEC: Read Register:           VERROSC: sent command: \$RC,0x5000050+71           VERROSC: reveal command: \$RC,0x5000050+0x00 190 1A0*7C		
INFO: Read Register:           VERBOSE: sent command: \$RC,0x50000508.7%1           VERBOSE: reveal command: \$RC,0x50000508,0x00000000*08		
INFO: Read Register: 1/EBR050: sent command: \$RC,0x5000050:*26 VERB050: reveal command: \$R2,0x5000050C,0x00000000*73		
INFO: Read Register: VERBOSE: sent command: \$RC,0x50000510"74 VERBOSE: reveal command: \$RS,0x50000510,0x00000000"01		
INFO: Read Register: 1/EBR050: sent command: \$RC,0x50000514*70 1/EBR050: reveal command: \$R2,0x50000514,0x00000000*05		
1VFO: Read Register: VEBIOSE: sent command: \$RC,0x50000518*7c VEBIOSE: reveal command: \$RR,0x50000518,0x00000003*0A		
10F0: Read Register: 1VERIOSE: sent command: §RC,0x5000051c <sup>+2</sup> 27 1VERIOSE: reveal command: §RC,0x5000051c,0x00505450 <sup>+73</sup>		
IVF0: Read Register: VERD05: sent command: \$RC,0x50000520*77 VERD05: reveal command: \$R2,0x50000520,0x00000000*02		
1940: Read Register: 1958/052: sent command: \$RC,0x50000524*73 1958/052: reveal command: \$RC,0x50000524,0x00000000*06		
10FO: Read Register: VERBOSE: sent command: \$RC,0x5000000c*23 VERBOSE: received command: \$RC,0x5000000c,0x01030000*74		
		~

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.

Advanced RTC Master CLK Clock PPS Slave TOD Slave	PTP Oc	
ad Value Address 0x2000000C	Load Config File NA Browse	
ad Config Clear Log Value 0x01020000	Save Config File NA Browse	
ve Config Save Log	Save Log File NA Browse	
ug Log		
BOSE: sent command: \$RC,0x50000500*75 BOSE: received command: \$RE.0x50000500.0x80000000*08		
O: Read Register:		
RBOSE: sent command: \$RC,0x50000504*71 RBOSE: received command: \$RR,0x50000504,0x001901A0*7C		
O: Read Register:		
BOSE: received command: \$RC,0X50000508-70 BOSE: received command: \$RR,0X50000508,0X00000000°08		
0: Read Register: BOSE: sent command: \$RC.0x5000050c*26		
BOSE: received command: \$RR,0x5000050C,0x00000000*73		
O: Read Register: RBOSE: sent command: \$RC,0x50000510*74		
RECSE: received command: \$KK,0x50000510,0x00000000001		
UNCED REGIST. IBOSE: sent command: \$RC,0x50000514*70 IBOSE: received command: \$RR.0x50000514.0x00000000*05		
=O: Read Register:		
RBOSE: sent command: \$RC,0x50000518*7c RBOSE: received command: \$RR,0x50000518,0x00000003*0A		
C: Read Register: 2RCSE: sent command: \$R.C.0x5000051c*27		
BOSE: received command: \$RR,0x5000051C,0x00505450*73		
F <b>O: Read Register:</b> RBOSE: sent command: \$RC,0x50000520*77		
lBOSE: received command: \$RR,0x50000520,0x00000000°02		
BOSE: sent command: \$RC,0x50000524*73 BOSE: received command: \$RR.0x50000524.0x00000000*06		
O: Read Register:		
REOSE: sent command: \$RC,0x5000000c*23 REOSE: received command: \$RR,0x5000000C,0x01030000*74		

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.



			rodate to	io siave in														
rsion	0x01030000	Vlan	0x0000					Pee	r Delay						Offset			
stance Nr	1 -		Vian Enable	ed	100								100					
ead Values	Write Values	Profile/Layer	Default	•	75								50					
art Refresh			Layer 3	•	50								0					
			P2P	•	25						_		-50					
		IP	192.168.0.1		0	2	4 6	. 8	10	12	14	16 19	-100	2 4	6 8 10	12	14 1	6 10
			C Enabled			-							, in the second s			**		
efault Dataset			Port Dataset				Current Da'	aset				Parent Dataset			Time Properties I	Dataset		
Clock Id	00:01:02:ff:fe:03	:04:05	Peer Delay	14			Steps Rer	moved 0	)			Parent Clock Id	:01:02:ff:fe	:03:04:05.0000	Time Source	0xa0		
Domain	0x00		State	MASTER			Offset	C	)			GM Cloick Id	00:01:02:ff	f:fe:03:04:05	PTP Timescale	$\checkmark$		
Priority 1	0x80						Delay	ľ	NA			GM Priority 1	0x80		Freq Traceable	•		
Priority 2	0x80		PDelayReq Lo	og Msg Interv	al O le							GM Priority 2	0x80		Time Traceable	•		
Accuracy	254		DelayReq Log	g Msg Interval	0							GM Accuracy	254		Leap 59			
Class	0xf8		Announce Lo	ig Msg Interva	1							GM Class	0xf8		Leap 61			
Short Id	0x0000		Announce Re	eceipt Timeou	3							GM Short Id	0x0000		UTC Offset Va			
Inaccuracy	50		Sync Log Msg	g Interval	0							GM Inaccuracy	50		UTC Offset	25		
Nr of Ports	1		Set Cust	tom Intervals								NW Inaccuracy	0		Current Offse	0		
Two Step															Jump Seconds	0		
															Next Jump	0		
															Display Name	PTP		

Figure 10: Write configuration

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

ng	Advanced	RTC Master	CLK Clock	PPS Slave	TOD Slave	PTP Oc					
Rea	d Value	Write Value	Address	0x20000	00C	Load Config	File NA	Browse			
Load	Config	Clear Log	Value	0x01020	000	Save Config	I File NA	Browse			
Save	Config	Save Log				Save Log Fil	e NA	Browse			
ebug	Log										
ERB	DSE: sent com	mand: \$RC,0x50	000500*75						 	 	
NEO	Read Registr	•r:									
VERB VERB	DSE: sent com DSE: received	mand: \$RC,0x50 command: \$RR,0	000504*71 x50000504,0>	001901A0*7C							
INFO	Read Registe	er:									
VERB	JSE: sent com DSE: received	mand: \$RC,0x50 command: \$RR,0	000508*7d 0x50000508,0>	000000008							
	Read Registe	er: mand: \$RC.0x50	00050c*26								
VERB	DSE: received	command: \$RR,0	x5000050C,0	<00000000*73							
(NFO VERB	Read Registe	ar: mand: \$RC,0x50	000510*74								
VERB	DSE: received	command: \$RR,(	x50000510,0x	000000000*01							
INFO VERB	DSE: sent com	mand: \$RC,0x50	000514*70								
VERD	Dood Dooloty	command: şkek,u	X50000514,03	0000000-05							
VERB	DSE: sent com DSE: received	mand: \$RC,0x50 command: \$RR.(	000518*7c x50000518.0x	00000003*0A							
INFO	Read Registe	er:									
VERB VERB	JSE: sent com JSE: received	mand: \$RC,0x50 command: \$RR,0	00051c*27 x5000051C,0	<00505450*73							
INFO	Read Registe	ar:	000520*77								
VERB	DSE: received	command: \$RR,(	x50000520,0	:00000000°02							
INFO VERB	Read Registe	er: mand: \$RC,0x50	000524*73								
VERB	DSE: received	command: \$RR,0	x50000524,0>	00000000*06							
INFO VERB	Read Registe	er: mand: \$RC,0x50	00000c*23								
	occurrence and and	command: ¢DD (	×5000000C.01	01030000*74							

Figure 11: Save configuration



 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,0x2000008,0x0000004

- ⇒ \$WC: this is the write command
- ⇒ 0x2000008: is the address
- ⇒ 0x0000004: 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

ig Advance	d RTC Master	CLK Clock	PPS Slave TOD Slave PTP Oc													
sion	0x01030000	Vlan	0x0000			Peer Delay							Offset			
tance Nr	1 💌		Vian Enabled	100						100						
ead Values	Write Values	Profile/Layer	Default 🔻	75						50						
art Refresh			Layer 3 🔹	50						0		-		_		
			P2P •	25						-50						
		IP	192.168.0.1	0						-100				12		
			Enabled	0 2	4 6	8 10	12 14	16	19	U	2 4	ь	8 10	12 1	4 1	) 19
fault Dataset			Port Dataset		Current Dataset			Р	arent Dataset				lime Properties D	ataset		
Clock Id	00:01:02:ff:fe:03	:04:05	Peer Delay 14		Steps Remove	d 0			Parent Clock Id	:01:02:ff:fe:	03:04:05.0000		Time Source	0xa0		
Domain	0x00		State MASTER		Offset	0			GM Cloick Id	00:01:02:ff:	fe:03:04:05		PTP Timescale	$\checkmark$		
Priority 1	0x80				Delay	NA			GM Priority 1	0x80			Freq Traceable			
Priority 2	0x80		PDelayReq Log Msg Interval	)					GM Priority 2	0x80			Time Traceable			
Accuracy	254		DelayReq Log Msg Interval	)					GM Accuracy	254			Leap 59			
Class	0xf8		Announce Log Msg Interval	1					GM Class	0xf8			Leap 61			
Short Id	0x0000		Announce Receipt Timeout	3					GM Short Id	0x0000			UTC Offset Val			
Inaccuracy	50		Sync Log Msg Interval	)					GM Inaccuracy	50			UTC Offset	25		
Nr of Ports	1		Set Custom Intervals						NW Inaccuracy	0			Current Offset	0		
Two Step													Jump Seconds	0		
													Next Jump	0		
													Display Name	PTP		



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

fig Advanced	d RTC Master	CLK Clock	PPS Slave TO	OD Slave	PTP Oc															
rsion	0x01030000	Vlan	0x0000						Peer Delay							Offset				
stance Nr	1 •		Vian Enabl	led		100							100							
Read Values	Write Values	Profile/Layer	Default	•		75							50							
tart Refresh			Layer 3	•		50							0		_		-	-		
			P2P	-		25							-50							
		IP	192.168.0.1			0							-100							
			Enabled			U	2 4	0	8 1	J 12	14 1	16 19	0	2 4	b	8	10	12 1	14 1	5 19
efault Dataset			Port Dataset				Curr	ent Datase	et			Parent Dataset				Time Pro	perties D	ataset		
Clock Id	00:01:02:ff:fe:03	3:04:05	Peer Delay	14			St	eps Remov	ved 0			Parent Clock I	d :01:02:f	f:fe:03:04:05.00	00	Time S	ource	0xa0		
Domain	0x00		State	MASTER	ι.		of	ffset	0			GM Cloick Id	00:01:0	2:ff:fe:03:04:05		PTP Tr	nescale	$\checkmark$		
Priority 1	0x80						De	elay	NA			GM Priority 1	0x80			Freq T	raceable			
Priority 2	0x80		PDelayReq L	Log Msg Inter	val 0							GM Priority 2	0x80			Time T	raceable			
Accuracy	254		DelayReq Lo	og Msg Interv	al 0							GM Accuracy	254			Leap 5	9			
Class	0xf8		Announce Le	.og Msg Interv	val 1							GM Class	0xf8			Leap 6	1			
Short Id	0x0000		Announce R	leceipt Timeo	ut 3							GM Short Id	0x0000			итс о	ffset Val			
Inaccuracy	50		Sync Log Ms	sg Interval	0							GM Inaccurac	y 50			итс о	ffset	25		
Nr of Ports	1		Set Cus	stom Intervals	,							NW Inaccurac	y 0			Currer	t Offset	0		
Two Step																Jump !	Seconds	0		
																Next J	ump	0		
																Displa	Name	РТР		

Figure 13: Read configuration

#### 2.4.3 Write Configuration

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



0x01030000	Vlan	0x0000					Peer	Delay							Offse	t			
1 🔻		Vian Enable	ed .	100								100							
Write Values	Profile/Layer	Default	•	75								50							
		Layer 3	•	50								0				-		-	
		P2P	-	25								-50							
	IP	192.168.0.1		0	2	4 6	8	10	12	14	16 19	-100	2	4 6	8	10	12	14	6 1
		Enabled			-		0	10	16		10 15		-			10	16		
		Dent Detreet									Descrit Defende				T				
00:01:02:ff:fe:03:0	4:05	Peer Delay	14			Steps Rem	noved 0				Parent Clock	Id :01:02:ff	:fe:03:04:05.	0000	Time	Source	0xa0		
0x00		State	MASTER			Offset	0			-	GM Cloick Id	00:01:02	:ff:fe:03:04:	)5	PTP T	imescale			
0x80						Delay	N	A		7	GM Priority 1	0x80		1	Freq	Traceable			
0x80		PDelayReq Lo	og Msg Interval	0						_	GM Priority 2	0x80		1	Time	Traceable			
254		DelayReq Log	g Msg Interval	0							GM Accuracy	254		7	Leap	59			
0xf8		Announce Lo	g Msg Interval	1							GM Class	0xf8		7	Leap	61			
0x0000		Announce Re	ceipt Timeout	3							GM Short Id	0x0000		1	итс о	Offset Val			
50		Sync Log Msg	g Interval	0							GM Inaccurac	y 50		7	итс о	Offset	25		
1		Set Cust	om Intervals								NW Inaccura	y 0		1	Curre	nt Offset	0		_
															Jump	Seconds	0		
															Next	Jump	0		
															Displa	y Name	PTP		_
	1 Vitte Values	1         V           Write Values         Profile/Layer           00:01:02:ff:fe:03:04:05         0x00           0x80         0x80           0x80         0x80           0x48         0x000           0x000         0x10           1         0x10	1         Van Enable           Write Values         Profile,Layer         Default           Layer 3         p2P           IP         192.168.0.1           Ø0:01:02:ff:fe:03:04:05         Port Dataset           Ø0:00         Port Dataset           Ø0:00         PoelayReq Lo           Øx80         PoelayReq Lo           Øx60         Announce Lo           Øx000         Sync Log Mg           1         Set Cust	I     Van Enabled       Write Values     Profile/Layer       Default     Van Enabled       Uayer 3     Van Enabled       IP     192.168.0.1       IP     192.168.0.1       Ox00     Port Dataset       Ox00     Poel Delay Req Log Msg Interval       0x80     PoelayReq Log Msg Interval       0x86     Amounce Log Msg Interval       1     Set Custom Intervals	Image: state	I         Van Enabled         100           Write Values         Profie/Layer         Default         Image: Construction of the second sec	Image: state stat	I         Van Enabled         100         75         55         25         0         2         4         6         8           Write Values         Profile/Layer         Izyer 3         V         50         25         0         2         4         6         8           IP         192.158.0.1         0         0         2         4         6         8           00:01:02:ff:ff:03:00+005         Port Dataset         Pere Delay         14         1         0         0         2         4         6         8           00:01:02:ff:ff:03:00+005         Port Dataset         Pere Delay         14         1         0         0         2         4         6         8           00:01:02:ff:ff:03:00+005         Port Dataset         Pere Delay         14         1         0         0         0         1         0         0         14         0         0         0         14         0         0         0         14         0         0         0         14         0         0         0         14         14         14         14         14         14         14         14         14         14         14         14	I         Van Enabled         100         75         50         25           IP         192.168.0.1         24         6         8         10           IP         192.168.0.1         24         6         8         10           Ox00         O         2         4         6         8         10           Ox00         O         2         4         6         8         10           Ox80         Poel Dataset         Current Dataset         Steps Removed 0         Offset         0           Ox80         PoelayReq Log Msg Interval         0         0         Announce Log Msg Interval         0           State         MASTER         0         0         0         0         0         0           State         Master         0	Image: state         Per Delay         Image: state         Image: state <td>Image: constraint of the second sec</td> <td>Image: state       Maximum (state)         Poil       100         Image: state       Maximum (state)         Image:</td> <td>Image: Period and a construction of the state o</td> <td>Image: Period and set in the value         Page         Image: Period and set in the value         Page: Page         Page         Image: Page         Page: Page         Page: Page         Page: Page         Page: Page         Page: Page: Page         Page: Pa</td> <td>Image: Second Second</td> <td>Image: state masked with values       Period balaset masked masked</td> <td>Image: State       Pare Delay       Image: State       I</td> <td>Image: Second Second</td> <td>Image: Signed State       Period Dataset       Parent Dataset       Parent Dataset       Parent Dataset         Image: Signed State       Parent Dataset       Parent Dataset       Parent Dataset       Parent Dataset         Image: Signed State       Parent Dataset       Offeet 0       Offeet 0       Offeet 0         Image: Signed State       Parent Dataset       Parent Dataset       Parent Dataset       Parent Dataset         Image: Signed State       Parent Dataset       Offeet 0       Offeet 0       Offeet 0       Offeet 0         Image: Signed State       MASTER       Offeet 0       Offeet 0       Offeet 0       Offeet 0         Image: Signed State       MASTER       Offeet 0       Offeet 0       Offeet 0       Offeet 0         Image: Signed State       Master Master       Offeet 0       Offeet 0       Offeet 0       Offeet 0         Image: Signed State       Master Master       Offeet 0       Offeet 0       Offeet 0       Offeet 0         Image: Signed State       Master Master       Offeet 0       Offeet 0       Offeet 0       Offeet 0         Image: Signed State       Master Master       Offeet 0       Offeet 0       Offeet 0       Offeet 0         Image: Signed State       Signe State       Image: Signe State       I</td>	Image: constraint of the second sec	Image: state       Maximum (state)         Poil       100         Image: state       Maximum (state)         Image:	Image: Period and a construction of the state o	Image: Period and set in the value         Page         Image: Period and set in the value         Page: Page         Page         Image: Page         Page: Page         Page: Page         Page: Page         Page: Page         Page: Page: Page         Page: Pa	Image: Second	Image: state masked with values       Period balaset masked	Image: State       Pare Delay       Image: State       I	Image: Second	Image: Signed State       Period Dataset       Parent Dataset       Parent Dataset       Parent Dataset         Image: Signed State       Parent Dataset       Parent Dataset       Parent Dataset       Parent Dataset         Image: Signed State       Parent Dataset       Offeet 0       Offeet 0       Offeet 0         Image: Signed State       Parent Dataset       Parent Dataset       Parent Dataset       Parent Dataset         Image: Signed State       Parent Dataset       Offeet 0       Offeet 0       Offeet 0       Offeet 0         Image: Signed State       MASTER       Offeet 0       Offeet 0       Offeet 0       Offeet 0         Image: Signed State       MASTER       Offeet 0       Offeet 0       Offeet 0       Offeet 0         Image: Signed State       Master Master       Offeet 0       Offeet 0       Offeet 0       Offeet 0         Image: Signed State       Master Master       Offeet 0       Offeet 0       Offeet 0       Offeet 0         Image: Signed State       Master Master       Offeet 0       Offeet 0       Offeet 0       Offeet 0         Image: Signed State       Master Master       Offeet 0       Offeet 0       Offeet 0       Offeet 0         Image: Signed State       Signe State       Image: Signe State       I

Figure 14: Write configuration



#### 2.4.4 Auto Refresh

 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.

		ULK CIOCK	PPS slave TOD Slave PTF	·u									
rsion	0x01030000	Vlan	0x0000	100		eer Delay			100	Offset			
tance Nr	1 🔻		Vian Enabled	100					100				
ead Values	Write Values	Profile/Layer	Default 🔻	/5					50				
tart Refresh			Layer 3 💌	50					0				_
			P2P ▼	25					-50				
		IP	192.168.0.1	0 2	4 6	8 10	12 14	16 19	-100 0 2 4	6 8 10	12 14	16	19
			✓ Enabled										
Jefault Dataset			Port Dataset		Current Dataset			Parent Dataset		Time Properties I	Dataset		
Clock Id	00:01:02:ff:fe:03	8:04:05	Peer Delay 14		Steps Remove	1 0		Parent Clock Id	:01:02:ff:fe:03:04:05.000	0 Time Source	0xa0		
Domain	0x00		State MASTER		Offset	0		GM Cloick Id	00:01:02:ff:fe:03:04:05	PTP Timescale	$\checkmark$		
Priority 1	0x80				Delay	NA		GM Priority 1	0x80	Freq Traceable	•		
Priority 2	0x80		PDelayReq Log Msg Interval	0				GM Priority 2	0x80	Time Traceable	•		
Accuracy	254		DelayReq Log Msg Interval	0				GM Accuracy	254	Leap 59			
Class	0xf8		Announce Log Msg Interval	1				GM Class	0xf8	Leap 61			
Short Id	0x0000		Announce Receipt Timeout	3				GM Short Id	0x0000	UTC Offset Va			
Inaccuracy	50		Sync Log Msg Interval	0				GM Inaccuracy	50	UTC Offset	25		
Nr of Ports	1		Set Custom Intervals					NW Inaccuracy	0	Current Offse	t 0		
Two Step										Jump Seconds	0		
										Next Jump	0		
										Display Name	PTP		

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.



ersion	0x01030000	Vlan	0x0000		10				Peer Delay	<i>(</i>				100			Offs	et				
stance Nr	1 *		Vian Enable	d	10	0								100								
tead Values	Write Values	Profile/Layer	Default	*	7	5								50								
top Refresh			Layer 3	•	5	0								0								1
			P2P	•	2	5								-50								
		IP	192.168.0.1			0 2	4	6	8	10 1	2 14	16 1	,	-100	2	4 6	8	10	12	14	16	19
			Enabled																			
efault Datase			Port Dataset				Current	t Dataset				Parer	t Dataset				Time P	operties D	ataset			
Clock Id	00:01:02:ff:fe:0	8:04:05	Peer Delay	14			Steps	s Remove	d O b			Par	ent Clock Id	:01:02:ff:fi	:03:04:05.0	0000	Time	Source	0xa0			]
Domain	0x00		State	MASTER	L		Offse	et	0			GM	Cloick Id	00:01:02:f	f:fe:03:04:0	5	PTP	Timescale	$\checkmark$			
Priority 1	0x80						Delay	y	NA			GM	Priority 1	0x80			Freq	Traceable				
Priority 2	0x80		PDelayReq L	og Msg Inter	val 0							GM	Priority 2	0x80			Time	Traceable				
Accuracy	254		DelayReq Lo	g Msg Interv	al 0							GM	Accuracy	254			Leap	59				
Class	0xf8		Announce Lo	g Msg Inter	al 1							GM	Class	0xf8			Leap	61				
Short Id	0x0000		Announce Re	ceipt Timed	ut 3							GM	Short Id	0x0000			υтс	Offset Val				
Inaccuracy	50		Sync Log Ms	g Interval	0							GM	Inaccuracy	50			υтс	Offset	25			]
Nr of Ports	1		Set Cust	om Intervals								NW	Inaccuracy	0			Curr	ent Offset	0			1
Two Step																	Jumj	Seconds	0			]
																	Next	Jump	0			]
																	Disp	ay Name	PTP			1
																	Disp	ay Name	РТР			

Figure 16: Stop Refresh



# A List of tables

Table 1:	Revision History	.4
Table 2:	Definitions	6
Table 3:	Abbreviations	.6

# **B** List of figures

Context Block Diagram	7
UART Ports	9
Ethernet Ports	10
Available Cores	11
Available Taps	12
Debug Logs	13
Access individual registers	14
Load configuration	15
Save configuration file	16
Write configuration	17
Save configuration	17
Load configuration	19
Read configuration	19
Write configuration	20
Start Refresh	
Stop Refresh	22
	Context Block Diagram