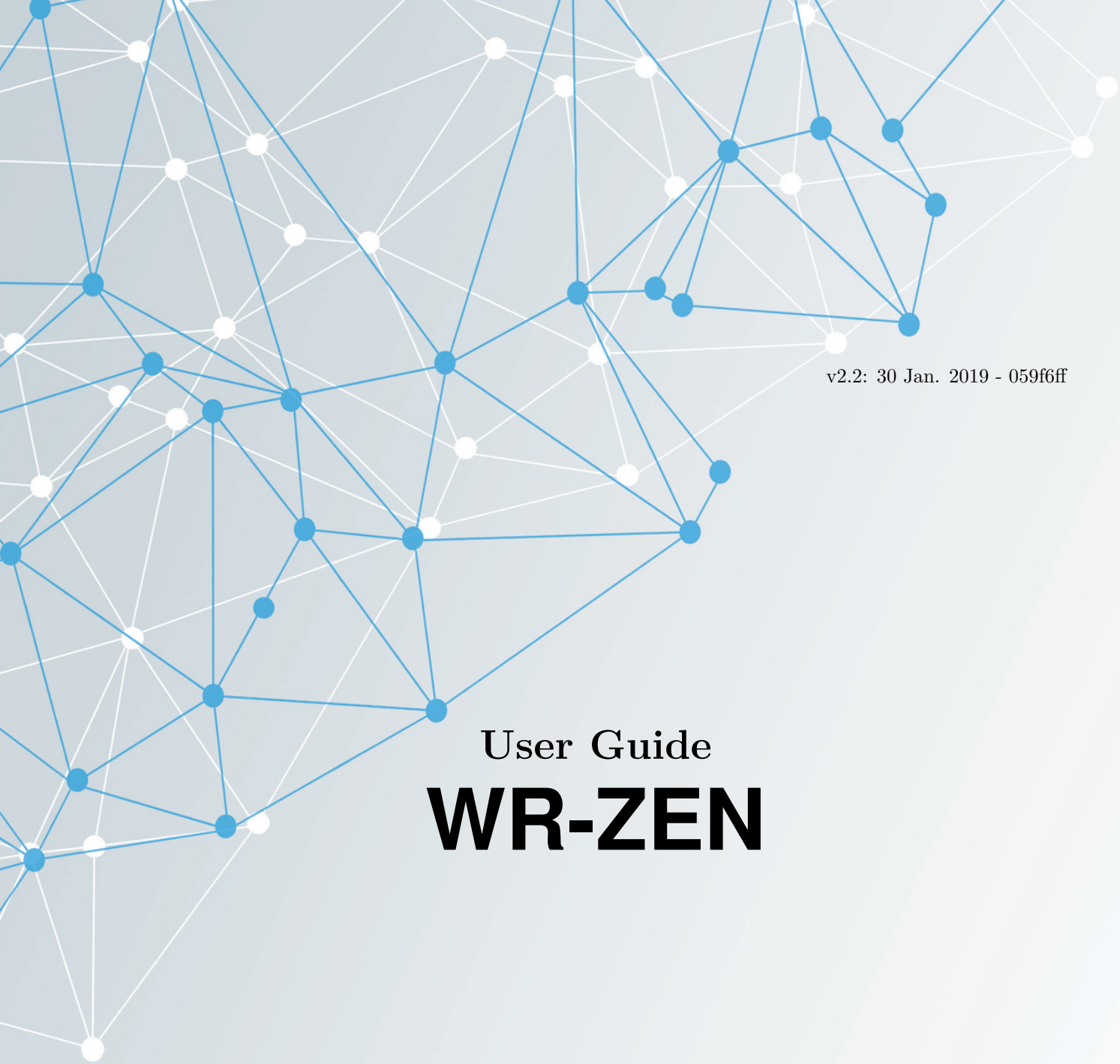


v2.2: 30 Jan. 2019 - 059f6ff

User Guide

WR-ZEN

Seven Solutions S.L
www.sevensols.com

SEVEN Solutions
When every nanosecond counts

Contents

Revision table

Rev	Date	Author	Comments
1.0b1	11/04/2016	Benoit Rat	First draft published
1.0b2	15/04/2016	Pablo Marin	Correct spelling
2.0	02/10/2017	Pablo Marin	Final v2 software support
2.1	21/11/2017	Pablo Marin	v2.1 software support
2.1.1	10/07/2017	Francisco Girela	Grammar correction
2.2	10/07/2017	Pablo Marin	v2.2 software support
2.2.2	18/10/2018	Fidel Rodríguez	Update and corrections
2.2.4	20/12/2018	Benoit Rat	Timing outputs & recovery mode

Glossary

CLI Command Line Interface (uart or ssh based), also known as terminal or command line.
DHCP The Dynamic Host Configuration Protocol to obtain network configuration.
FMC FPGA Mezzanine Card, an ANSI standard for mezzanine card form factor.
HDL Hardware description language.
LM32 LatticeMico32 is a 32-bit microprocessor soft core optimized for field-programmable gate arrays (FPGAs).
NAND NAND Flash Memory, a type of reprogrammable non-volatile computer memory.
PCIe Peripheral Component Interconnect Express, a high-speed serial computer expansion bus standard.
PTP Precise Time Protocol, a time synchronization protocol.
SFP Small form-factor pluggable transceiver, a hot-pluggable transceiver for optical fiber.
SMC SubMiniature version C, coaxial connector used in radio-frequency circuits.
SNMP Simple Network Management Protocol, for management of network devices.
SPEC Simple PCIe FMC carrier.
SVEC Simple VME FMC carrier.
UART Universal Asynchronous Receiver/Transmitter.
WR White Rabbit.
WRS White Rabbit Switch.
WMI Web Management Interface

1 Introduction.

1.1 About the White Rabbit technology.

White Rabbit (WR) is an extension to Ethernet network with **ultra-accurate synchronization** and **Gigabit data transfer capability**. It has been conceived to fulfill the following goals:

- ***Time Precision:*** The WR technology provides a common clock for physical layer in the entire network, allowing synchronization at **subnanosecond** level with picoseconds precision.
- ***Scalability:*** The WR network is designed to be highly scalable to support **thousands of nodes**. It also intends to be as modular as possible and compatible with non-WR devices.
- ***Distance Range:*** Taking into account the size and ranges of the majority industrial and scientific facilities, the WR network specifications have been designed to support distances up to several tens of kilometers between nodes.

1.2 About this document.

This document describes the essential information to enable the user the first contact with the WR-ZEN TP.

If you are a partner and want to develop your own application you can find more information in the `doc/AdvancedGuide.md` file of the wr-zynq-os source code. This tutorial will help you in re-building your own OS with you custom applications.

2 The WR-ZEN TP family.

The White Rabbit ZEN Time Providers (aka. WR-ZEN TP) is a high performance timing nodes family whose design is based on the **Xilinx Serie-7 FPGAs** that combines an Artix-FPGA with a Dual-Core ARM.

It is composed of the next products:

- WR-ZEN TP
- WR-ZEN TP-32BNC
- WR-ZEN TP-FL

They are the versatile standalone nodes that provide WR features (i.e., sub-nanosecond time synchronization in a network with thousands of nodes) controlled by a user friendly Linux OS which runs on a dual ARM processor.

The WR-ZEN TP **easily distributes** time and frequency to other equipments by implementing standard timing protocols such as PTP, NTP and IRIG-B. The WR-ZEN TP combines **ultra stable clocks** with low jitter and temperature compensated clock resources to enhance its synchronization accuracy. In addition, it allows integrating FMC cards.

Its **robustness** to failure is one of its key feature thanks to **redundant** dual power suply and a fanless design.

It is also worth mentioning its ability to support **daisy chain configurations** in which each WR-ZEN TP device synchronizes with another WR-ZEN TP belonging to a higher order in the hierarchy level. At the same time, it provides synchronization to a lower level.

2.1 Front panel (Legend)

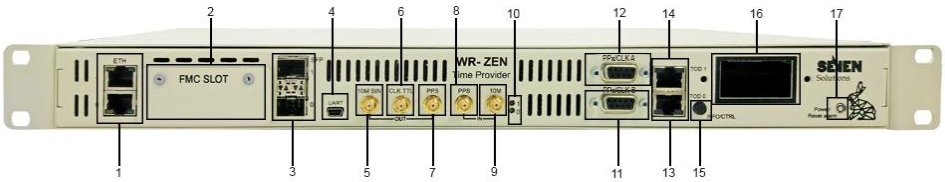


Figure 1: Front Panel of the WR-ZEN TP

1. Dual ethernet ports (eth0 & eth1)
2. FMC expansion port
3. Dual SFP (WR) ports (wr0 & wr1)
4. ARM Mini-USB (B) UART
5. 10MHz sinusoidal reference clock output
   - SMA connector (F)
   - 50 Ω termination
   - 2.5V (LVTTL), Output from PLL
6. CLK TTL: Configurable clock output
   - SMA connector (F)
   - 50 Ω termination
   - High level out: 2.5V +/-0.2V
   - By default it is configured to output 10MHz (customizable).
7. PPS output
   - SMA connector (F)
   - 50 Ω termination
   - Digital output
   - High level output: 2.5V +/-0.2V
   - Timing: PPS OUT is output with a fixed (tuneable) delay to the rising edge of 10MHz OUT and active for 10ms
8. PPS input
   - SMA connector (F)
   - 50 Ω termination
   - TTL input (3.3V)
9. 10 MHz input
   - SMA connector (F)
   - 50 Ω termination
   - 1Veff (+/-30%) sine wave
10. LED
    - red: error LED
    - green: status LED
11. DB9: PPx/CLK A
    - With 50 Ω termination: 1.7V +/-0.2V
12. DB9: PPx/CLK B (with selectable 50 Ω termination)
    - With 50 Ω termination: 3V +/-0.2V
    - Without 50 Ω adaptation: 3.3V +/-0.2V
13. TOD0: RJ45 RS422 (Time of Day / NMEA)
14. TOD1: RJ45 RS422 (Time of Day / NMEA)
15. INFO/CTRL Button
16. LCD Display: panel for alert & configuration information
17. Power/Reset Alarm button

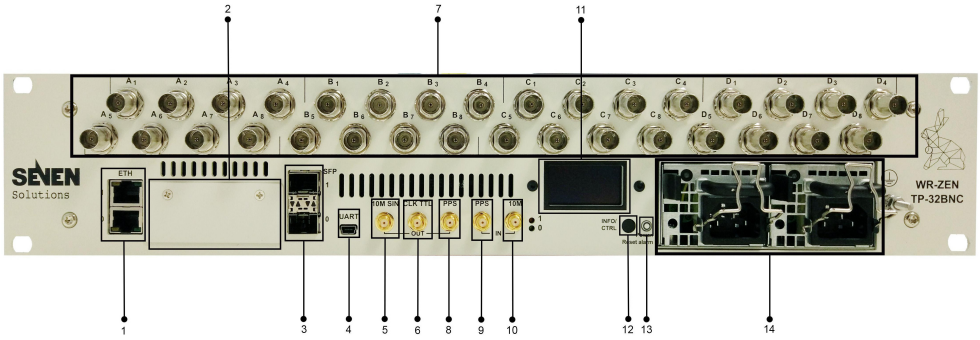


Figure 2: Front Panel of the WR-ZEN-TP 32-BNC

1. Dual ethernet ports (eth0 & eth1)
2. FMC expansion port
3. Dual SFP (WR) ports (wr0 & wr1)
4. ARM Mini-USB (B) UART
5. 10MHz sinusoidal reference clock output
   - SMA connector (F)
   - 50 Ω termination
   - 2.5V (LVTTL), Output from PLL
6. CLK TTL: Configurable clock output
   - SMA connector (F)
   - 50 Ω termination
   - High level out: 2.5V +/-0.2V
   - By default it is configured to output 10MHz (customizable).
7. 32 configurable PPS/10 MHz
   - BNC connector (F)
   - Digital output
   - High level output for groups A & B:
     - With 50 Ω termination: 1.7V +/-0.2V
   - High level output for groups C & D (Selectable 50 Ω termination):
     - With 50 Ω termination: 3V +/-0.2V
     - Without 50 Ω adaptation: 3.3V +/-0.2V
8. PPS output
   - SMA connector (F)
   - 50 Ω termination
   - Digital output
   - High level output: 2.5V +/-0.2V
   - Timing: PPS OUT is output with a fixed (tuneable) delay to the rising edge of 10MHz OUT and active for 10ms
9. PPS input
   - SMA connector (F)
   - 50 Ω termination
   - TTL input (3.3V)
10. 10 MHz input
    - SMA connector (F)
    - 50 Ω termination
    - 1Veff (+/-30%) sine wave
11. LCD Display: panel for alert & configuration information
12. INFO/CTRL Button
13. Power/Reset Alarm button
14. Power Supply 0 & 1

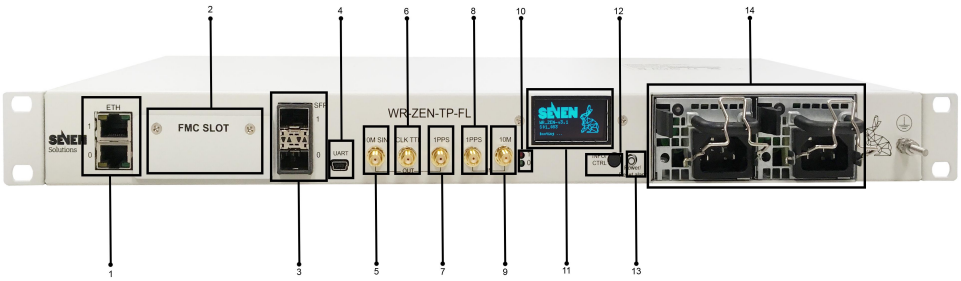


Figure 3: Front Panel of the WR-ZEN TP-FL

1. Dual ethernet ports (eth0 & eth1)
2. FMC expansion port
3. Dual SFP (WR) ports (wr0 & wr1)
4. ARM Mini-USB (B) UART
5. 10MHz sinusoidal reference clock output
   - SMA connector (F)
   - 50 Ω termination
   - 2.5V (LVTTL), Output from PLL
6. CLK TTL: Configurable clock output
   - SMA connector (F)
   - 50 Ω termination
   - High level out: 2.5V +/-0.2V
   - By default it is configured to output 10MHz (customizable).
7. PPS output
   - SMA connector (F)
   - 50 Ω termination
   - Digital output
   - High level output: 2.5V +/-0.2V
   - Timing: PPS OUT is output with a fixed (tuneable) delay to the rising edge of 10MHz OUT and active for 10ms
8. PPS input
   - SMA connector (F)
   - 50 Ω termination
   - TTL input (3.3V)
9. 10 MHz input
   - SMA connector (F)
   - 50 Ω termination
   - 1Veff (+/-30%) sine wave
10. LED
    - red: error LED
    - green: status LED
11. LCD Display: panel for alert & configuration information
12. INFO/CTRL Button
13. Power/Reset Alarm button
14. Power Supply 0 & 1

2.2 Back panel (Legend)

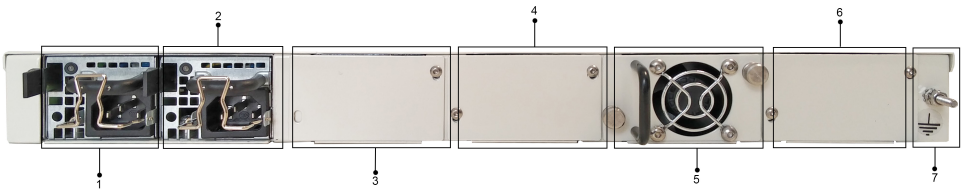


Figure 4: Back Panel of the WR-ZEN TP

1. Power Supply 0
2. Power Supply 1
3. Plug'n'play module slot #J1
4. Plug'n'play module slot #J2
5. Plug'n'play module slot #J3
    - By default we provide the WR-ZEN-TP with one FAN on Slot #J3
6. Plug'n'play module slot #J4
7. Ground Connector

3 Quick-Start

3.1 Default Settings

The device is factory configured with the following default settings:

- Ethernet ports
  - eth0 get its IP from a **DHCP** server
  - eth1 is configured with the static IP 192.168.77.100
- SFP WR ports:
  - wr0: By default configured as SLAVE
  - wr1: By default configured as MASTER
- SSH/WEB credentials: **root**/**root**

3.2 Obtaining the IP address

In order to login to the WR-ZEN TP using SSH or through the web interface you need to obtain the IP of the Ethernet management port.

If you have configured your network interface with a static IP (i.e, 192.168.77.10) in the same LAN as the eth1 port, you already know the IP of the eth1 port so you can connect using:

- `http://192.168.77.100` (in a web browser)
- `ssh root@192.168.77.100` (in a terminal)

However if you have a DHCP server and you want to use it to configure both your PC network interface and the WR-ZEN TP eth0 interface, you need to know which IP has been assigned to the eth0 port. You can use several ways to retrieve the eth0 network interface IP of the WR-ZEN TP:

3.2.1 Using the LCD screen

The IP of both `ETH` interfaces will be displayed in one of the menus in the LCD screen. Once the network menu is displayed, press the `CTRL/Info` button, until reaching the corresponding menu.

***Notes**: The LCD menu with network interface is only available starting on the `wr-zynq-os-v1.2` release.*

3.2.2 Using the USB-UART

Login through USB-UART and execute the command `ifconfig eth0` to show detailed information about the eth0 interface including the IP address.

This is an example of ifconfig output:

```
root@wrztp-xxx:~# ifconfig eth0
eth0 Link encap:Ethernet HWaddr 64:FB:81:20:10:17
         inet addr:192.168.1.115 Bcast:192.168.7.255 Mask:255.255.255.0
         inet6 addr: fe80::66fb:81ff:fe20:1017/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
         RX packets:12 errors:0 dropped:0 overruns:0 frame:0
         TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:2996 (2.9 KiB) TX bytes:1806 (1.7 KiB)
         Interrupt:147 Base address:0xb000
```

3.3 Login through SSH

In order to connect to the WR-ZEN TP using the SSH protocol you need to know the IP of your device. Please read the previous section to know which IP is the one you can use to connect to your device. Once you have the correct IP:

Notes**: Note that the default port used for the SSH service is **22

3.3.1 SSH Ubuntu LTS

Connecting to your device using the SSH protocol is straighforward in Ubuntu (and other Linux distributions). Just open a console and execute:

```
ssh root@<ip_of_your_device>
```

Then you will be connected directly to your device using the SSH protocol.

3.3.2 SSH Windows

Similarly to the section USB-UART under Windows, you can connect to the WR-ZEN TP using the SSH protocol with the Putty application.

You should not forget to replace the IP of your WR-ZEN TP (yellow) by the one in your subnetwork as shown in Fig. 5.

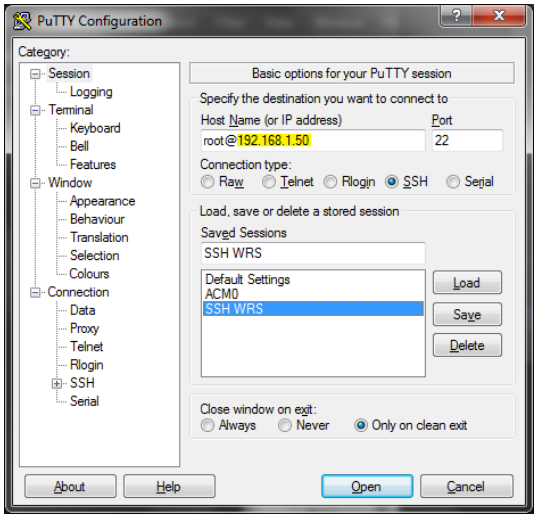


Figure 5: Putty - SSH connection.

3.4 Login through the Mini-USB UART

The Mini-USB UART is one of the fastest and simplest way to connect locally to the WR-ZEN TP.

3.4.1 UART Ubuntu LTS

The connection with an UART using Ubuntu LTS (and other distributions) can be done with the `picocom` tools.

First you must install the `picocom` package by executing

```
sudo apt-get install picocom
```

Then you need to plug a USB cable from your PC to the mini-USB UART port of the WR-ZEN TP. Once the connection is done you should see a new device in your system. Below an example of two WR-ZEN TP connected to a PC with Ubuntu LTS 14.04 is shown.

```
#### List of UART devices connected through USB
>$ lsusb | grep UART
Bus 002 Device 068: ID 10c4:ea60 Cygnal Integrated Products, Inc. CP210x UART Bridge / myAVR
    mySmartUSB light
Bus 002 Device 069: ID 10c4:ea60 Cygnal Integrated Products, Inc. CP210x UART Bridge / myAVR
    mySmartUSB light

#### Get the device name
>$ ls /dev/ttyUSB*
/dev/ttyUSB0
/dev/ttyUSB1
```

If you only have one device connected to your PC, you should see only the device `ttyUSB0`, but if you have more devices the order of ID of the ttyUSB device will be given by the order of the connection.

Finally, when you know which ttyUSB is the one that corresponds to your device (i.e, ttyUSB0) you can execute the following command in order to open the console to this device.

```
#### Connecting to the Management USB port
>$ picocom /dev/ttyUSB0 -b 115200
```

Once the console is opened, you can connect to the device using your user and password (root/root). Below an example of the login to the WR-ZEN TP with the serial number #023.

```
Welcome to WR-Zen board
wrztp-020 login: root
Password:
root@wrztp-020:~#
```

***Notes**: Previously we have suggested to use `minicom` tool but it is not compatible with the color scheme of `wrz_config` (strange characters appear) so it is why we now recommend to install `picocom`.*

3.4.2 UART Windows

The process to connect to the UART using Windows (XP,Vista,7,8,10) is explained below:

1. You first need to download the Putty Tool and install it.
2. Then you need to list and find out which serial port in Windows corresponds to which interface. A simple way to perform this is to plug only one USB cable at a time, and go to `Device Manager > Ports (COM & LPT)` to check the name of the `COM<X>` port.
3. Finally to connect through the USB you just need to open the Putty tool and configure it as indicated in Fig.6. Do not forget to replace the `COM9` port name by the one that corresponds to the USB management.

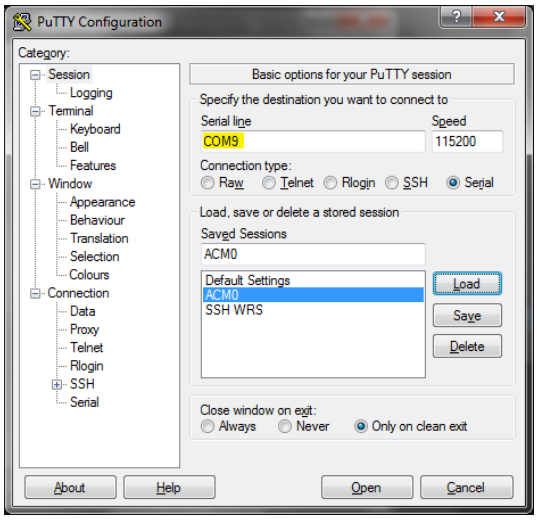


Figure 6: Putty - USB connection.

Compatibility with `wrz_config`

In order to have putty compatible with the `wrz_config` color scheme and avoid strange characters we suggest the user to try the following option under `Window:Translation:`

- Change remote character set to ISO-8859-1.
- Uncheck “Override with UTF-8 if locale says so”.
- Select “Use Unicode line drawing code points”. (This is the default.)

4 WR Operation

White Rabbit (WR) is an ultra-accurate time transfer protocol derived from PTPv2. Like PTP it needs a master device and one or more slaves. The slaves discipline their clock frequency and phase based on the one received from the master. This requires an specific configuration per link in which one end will be the master and the other end of the link will be the slave. Every device allows only one port acting as slave because the internal clock cannot be disciplined from more than one source at the same time. By default, the port acting as slave will be the one with the lowest number. The rest of them will act as master. White Rabbit requires a fiber connection so it is only supported on the SFP ports. The WR protocol acts only on layer 2 so there is no need to assign an IP address to the ports.

You can change this default configuration changing the *mode* on the *WR/PPSI* options. The different configuration options are:

- Grandmaster: Both SFP ports are configured as master. It requires a valid 10 MHz and PPS input signals. To know more check section grandmaster mode
- Master: Both SFP ports are configured as master.
- Slave on WR1: WR0 is configured as master, WR1 is configured as slave.
- Slave on WR0: WR1 is configured as master, WR0 is configured as slave.

All changes that are not saved from the web interface or the wrz_config tool, will be erased after a reboot. To test the actual configuration without rebooting, the next command can be used:

```
gpa_ctrl hald calib/force_refresh 1
```

The devices are factory calibrated to use with the following SFPs:

Vendor	model	Tx/Rx Wavelength (nm)	Distance
Axcen	AXGE-1254-0531	1310/1490	10 Km
Axcen	AXGE-3454-0531	1490/1310	10 Km
Axcen	AXGE-3254-0591	1310/1490	40 Km
Axcen	AXGE-3454-0591	1490/1310	40 Km
Axcen	AXGD-3854-05Cx	1490/1550	80 Km
Axcen	AXGD-3954-05Cx	1550/1490	80 Km

There could be more SFPs (for distances up to 120 Km) added by request. Please, if you are interested, contact the Seven Solutions support.

5 PTP Operation

IEEE 1588-2008 protocol is supported on the management interfaces eth0 and eth1 (by default on eth1). The devices can act as slave and as master. This allows the devices to *translate* between WR and PTP.

The supported profiles are:

- Default: multicast, end to end, IPv4/IPv6 (layer 3).
- Telecom (G.8265.1): unicast, end to end, IPv4/IPv6 (layer 3).
- Telecom (G.8275.1): multicast, end to end, IPv4/IPv6 (layer 3).
- Power (C37.238-2011): multicast, point to point, Ethernet (layer 2). *BETA*.

You can change the configuration by using the web interface in section Timing and then PTP 1588 web interface, or by command line by using the wrz_config. Note that any change on the configuration require a reboot of the ptp daemon running. You can reboot the device for that or use the next command:

```
/etc/init.d/ptpd restart
```

All changes on the configuration of the device are saved on the file *.config* located on the /root/ directory. Here are two examples of the default and telecom profiles:

Default:

```
CONFIG_PTPD_CONFIG_INTERFACE_ETH1=y
CONFIG_PTPD_CONFIG_PRESET_MASTERONLY=y
CONFIG_PTPD_PROFILE=default
```

Telecom (G.8265.1):

```
CONFIG_PTPD_CONFIG_INTERFACE_ETH1=y
CONFIG_PTPD_PROFILE=t-8265.1
CONFIG_PTPD_IPV4_UNICAST="192.168.7.5"
```

Additional PTP options can be changed editing the ptp configuration file located in `/etc/ptpd/ptpd_default.conf`. If you need to restart the ptp daemon you can use the following command:

```
/etc/init.d/ptpd restart
```

6 Running Modules and Parameters

Here you will find an explanation of all the different parameters available to the user in each module. In the section tools it will be explained how to read/write them.

Each parameters have a specific access mode given by a combination of the flags {R,W,L,X}:

- R: The parameter can be Read. *(almost all parameters are R except the one similar to "reset/refresh")*
- W: The parameter can be Written *(this means that it can be directly applied without reboot).*
- L: The parameter can be Loaded at startup *(this means that it can be saved in a .config and then loaded at startup of its module/service).*
- X: The parameter is only shown in eXpert mode.

6.1 HALD

This is the list of parameters of the Hardware Abstraction Layer Daemon (HALD). You can control the Ethernet interfaces as well as the calibration values for the White Rabbit protocol. Some of the parameters related to timing outputs are specific to TP/TP-32BNC, please read the timing outputs section for more information.

The list of all available parameters is below:

prefix/key	unit	access	description
ver/hw/base/board_manucturer		R	
ver/hw/base/product_name		R	
ver/hw/base/part_number		R	
ver/hw/base/serial_number		R	
ver/hw/base/eth0.ethaddr		R	
ver/hw/base/eth1.ethaddr		R	
ver/hw/base/version		R	
ver/hw/base/product		R	
ver/hw/base/manufacturer		R	
ver/hw/base/hw.serial-number		R	
ver/hw/base/hw.batch		R	
ver/hw/base/hw.fpga		R	
ver/hw/base/hw.full		R	
ver/hw/base/hw.date		R	
ver/hw/exp/board_manucturer		R	
ver/hw/exp/product_name		R	
ver/hw/exp/part_number		R	
ver/hw/exp/serial_number		R	
ver/hw/mod0/board_manucturer		R	
ver/hw/mod0/product_name		R	
ver/hw/mod0/part_number		R	
ver/hw/mod0/serial_number		R	
ver/gw/id		R	ID of the current Gateware
ver/gw/major		R	Major version of the Gateware
ver/gw/minor		R	Minor version of the Gateware
ver/gw/build		R	This field includes the build number indexed with the GW version (optional)
ver/gw/synth_date		R R	Synthesis date that corresponds of the current gateware
ver/sw/tag		R	Software version (release tag)
ver/sw/compiled		R	Software compilation date
ver/sw/libgpa		R	Software version of libgpa (release tag)
ver/sw/uboot		R	Software version of uboot (release tag)
device/code_id		R	Device code ID
device/name		R	Name of the device
device/bitfname		R	Bitstream of the device

prefix/key	unit	access	description
device/hostname		RL	This name is used by many of the networking programs to identify the machine
eth0/link_up		R	Active when the link on eth0 is up
eth0/ethaddr		R	MAC address of eth0
eth0/ipv4.addr		RL	IP address of eth0
eth0/ipv4.netmask		RL	Subnet mask (32-bits) for eth0
eth0/gw_address		RL	Default gateway for eth0
eth0/bc_address		RL	Broadcast address for eth0
eth0/is_dhcp		RL R	Activate the dynamic IP address resolution on eth0 using DHCP
eth0/speed		R	Auto-negotiated speed of the eth0 interface
eth0/tx_packets		R	Transmitted packets on eth0
eth0/rx_packets		R	Received packets on eth0
eth0/tx_bytes		R	Transmitted bytes on eth0
eth0/rx_bytes		R	Received bytes on eth0
eth0/tx_errors		R	Transmission errors on eth0
eth0/rx_errors		R	Reception errors on eth0
eth1/link_up		R	Active when the link on eth1 is up
eth1/ethaddr		R	MAC address of eth1
eth1/ipv4.addr		RL	IP address of eth1
eth1/ipv4.netmask		RL	Subnet mask (32-bits) for eth1
eth1/gw_address		RL	Default gateway for eth1
eth1/bc_address		RL	Broadcast address for eth1
eth1/is_dhcp		RL	Activate the dynamic IP address resolution on eth1 using DHCP
eth1/speed		R	Auto-negotiated speed of the eth1 interface
eth1/tx_packets		R	Transmitted packets on eth1
eth1/rx_packets		R	Received packets on eth1
eth1/tx_bytes		R	Transmitted bytes on eth1
eth1/rx_bytes		R	Received bytes on eth1
eth1/tx_errors		R	Transmission errors on eth1
eth1/rx_errors		R	Reception errors on eth1
sync/pps_coffset	ns	RWL	Change the coarse offset of the PPS (multiple of 8ns)
sync/pps_width	ns	RWL	Change the width of the PPS (multiple of 8ns)
sync/pps_xtimes		RWL	xtimes to repeat PPS (1PPS; 10PPS, 100PPS, 1KPPS)
sync/sync_ad9516	ns	RWL	Resync the PLL (AD9516-1) with respect to the internal PPS (multiple of 8ns)
sync/sync_lmk	ns	RWL	Resync the PLL LMK with respect to the internal PPS (multiple of 8ns)
sync/AB_xpps_en		RWL	Enable the xPPS mode: 0:freq from the PLL 1: xPPS latched
sync/A_mode		RWL	xPPS mode of Ax BNC outputs: (only when AB_xpps_en=1) 0: PPS 1: IRIG-B
sync/B_mode		RWL	xPPS mode of Bx BNC (only when AB_xpps_en=1): 0: PPS 1: IRIG-B
sync/C_mode		RWL	xPPS mode of Cx BNC outputs: 0: PPS 1: IRIG-B
sync/D_mode		RWL	xPPS mode of Dx BNC outputs: 0: PPS 1: IRIG-B
sync/C_term_adapt		RWL	Adapt Cx outputs for 50Ohm termination
sync/D_term_adapt		RWL	Adapt Dx outputs for 50Ohm termination
sync/A_fdelay	ps	RWL	Finely delay the output A
sync/B_fdelay	ps	RWL	Finely delay the output B
sync/C_fdelay	ps	RWL	Finely delay the output C
sync/D_fdelay	ps	RWL	Finely delay the output D

prefix/key	unit	access	description
calib/force_refresh		RW	Force a refresh to apply the (modified) calibration values
calib/force_usr		RWL	When true the user value are not override by GW calibration file
calib/wr0_dtx	ps	RWL	Fixed Tx delay of wr0 (wri1)
calib/wr0_drx	ps	RWL	Fixed Rx delay of wr0 (wri1)
calib/wr0_fiber_idx		RWL	Associated index of fiber for wr0 (wri1)
calib/wr1_dtx	ps	RWL	Fixed Tx delay of wr1 (wri2)
calib/wr1_drx	ps	RWL	Fixed Rx delay of wr1 (wri2)
calib/wr1_fiber_idx		RWL	Associated index of fiber for wr1 (wri2)
calib/n_sfps		RL	Number of type of SFPs in the database with an associated calibration
calib/sfp00_vn		RWL	Vendor-name for SFP database entry #00
calib/sfp00_pn		RWL	Part-number for SFP database entry #00
calib/sfp00_dtx	ps	RWL	Fixed Tx delay for SFP database entry #00
calib/sfp00_drx	ps	RWL	Fixed Rx delay for SFP database entry #00
calib/sfp00_tx_wl	nm	RWL	Tx wavelength associated to SFP #00
calib/sfp00_fiber_idx		RWL	Index of fiber associated to SFP #00
calib/sfp01_vn		RWL	Vendor-name for SFP database entry #01
calib/sfp01_pn		RWL	Part-number for SFP database entry #01
calib/sfp01_dtx	ps	RWL	Fixed Tx delay for SFP database entry #01
calib/sfp01_drx	ps	RWL	Fixed Rx delay for SFP database entry #01
calib/sfp01_tx_wl	nm	RWL	Tx wavelength associated to SFP #01
calib/sfp01_fiber_idx		RWL	Index of fiber associated to SFP #01
calib/sfp02_vn		RWL	Vendor-name for SFP database entry #02
calib/sfp02_pn		RWL	Part-number for SFP database entry #02
calib/sfp02_dtx	ps	RWL	Fixed Tx delay for SFP database entry #02
calib/sfp02_drx	ps	RWL	Fixed Rx delay for SFP database entry #02
calib/sfp02_tx_wl	nm	RWL	Tx wavelength associated to SFP #02
calib/sfp02_fiber_idx		RWL	Index of fiber associated to SFP #02
calib/sfp03_vn		RWL	Vendor-name for SFP database entry #03
calib/sfp03_pn		RWL	Part-number for SFP database entry #03
calib/sfp03_dtx	ps	RWL	Fixed Tx delay for SFP database entry #03
calib/sfp03_drx	ps	RWL	Fixed Rx delay for SFP database entry #03
calib/sfp03_tx_wl	nm	RWL	Tx wavelength associated to SFP #03
calib/sfp03_fiber_idx		RWL	Index of fiber associated to SFP #03
calib/sfp04_vn		RWL	Vendor-name for SFP database entry #04
calib/sfp04_pn		RWL	Part-number for SFP database entry #04
calib/sfp04_dtx	ps	RWL	Fixed Tx delay for SFP database entry #04
calib/sfp04_drx	ps	RWL	Fixed Rx delay for SFP database entry #04
calib/sfp04_tx_wl	nm	RWL	Tx wavelength associated to SFP #04
calib/sfp04_fiber_idx		RWL	Index of fiber associated to SFP #04
calib/sfp05_vn		RWL	Vendor-name for SFP database entry #05
calib/sfp05_pn		RWL	Part-number for SFP database entry #05
calib/sfp05_dtx	ps	RWL	Fixed Tx delay for SFP database entry #05
calib/sfp05_drx	ps	RWL	Fixed Rx delay for SFP database entry #05
calib/sfp05_tx_wl	nm	RWL	Tx wavelength associated to SFP #05
calib/sfp05_fiber_idx		RWL	Index of fiber associated to SFP #05
calib/n_fibers		RL	Number of fiber type in the database with a calibrated asymmetry
calib/fiber00_wltx	nm	RWL	TX wavelength associated with asymmetry of fiber DB entry #00 (if 0 or negative no check)
calib/fiber00_wlrx	nm	RWL	RX wavelength associated with asymmetry of fiber DB entry #00 (if 0 or negative no check)
calib/fiber00_alpha		RWL	Alpha asymmetry for fiber DB entry #00
calib/fiber00_desc		RWL	Description of fiber DB #00 (i.e, type G652-D)
calib/fiber01_wltx	nm	RWL	TX wavelength associated with asymmetry of fiber DB entry #01 (if 0 or negative no check)
calib/fiber01_wlrx	nm	RWL	RX wavelength associated with asymmetry of fiber DB entry #01 (if 0 or negative no check)
calib/fiber01_alpha		RWL	Alpha asymmetry for fiber DB entry #01
calib/fiber01_desc		RWL	Description of fiber DB #01 (i.e, type G652-D)
calib/fiber02_wltx	nm	RWL	TX wavelength associated with asymmetry of fiber DB entry #02 (if 0 or negative no check)

prefix/key	unit	access	description
calib/fiber02_wlrx	nm	RWL	RX wavelength associated with asymmetry of fiber DB entry #02 (if 0 or negative no check)
calib/fiber02_alpha		RWL	Alpha asymmetry for fiber DB entry #02
calib/fiber02_desc		RWL	Description of fiber DB #02 (i.e, type G652-D)
calib/fiber03_wltx	nm	RWL	TX wavelength associated with asymmetry of fiber DB entry #03 (if 0 or negative no check)
calib/fiber03_wlrx	nm	RWL	RX wavelength associated with asymmetry of fiber DB entry #03 (if 0 or negative no check)
calib/fiber03_alpha		RWL	Alpha asymmetry for fiber DB entry #03
calib/fiber03_desc		RWL	Description of fiber DB #03 (i.e, type G652-D)
calib/fiber04_wltx	nm	RWL	TX wavelength associated with asymmetry of fiber DB entry #04 (if 0 or negative no check)
calib/fiber04_wlrx	nm	RWL	RX wavelength associated with asymmetry of fiber DB entry #04 (if 0 or negative no check)
calib/fiber04_alpha		RWL	Alpha asymmetry for fiber DB entry #04
calib/fiber04_desc		RWL	Description of fiber DB #04 (i.e, type G652-D)
calib/fiber05_wltx	nm	RWL	TX wavelength associated with asymmetry of fiber DB entry #05 (if 0 or negative no check)
calib/fiber05_wlrx	nm	RWL	RX wavelength associated with asymmetry of fiber DB entry #05 (if 0 or negative no check)
calib/fiber05_alpha		RWL	Alpha asymmetry for fiber DB entry #05
calib/fiber05_desc		RWL	Description of fiber DB #05 (i.e, type G652-D)

6.2 PPSi

This module contains the parameters to configure the White Rabbit protocol and its statistics. The list of all available parameters is below:

prefix/key	unit	access	description
spll/mode		R	0: IDDLE 1: GM 2: FREE-RUNING 3: SLAVE 4: DISABLED
spll/seq_state		R	State machine for the soft PLL
spll/ext.enabled		R	
spll/ext.align_state		R	
spll/ext.align_timer		R	
spll/ext.align_target		R	
spll/ext.align_step		R	
spll/ext.align_shift		R	
spll/mpll.state		R	
spll/mpll.pi.ki		R	
spll/mpll.pi.kp		R	
spll/mpll.pi.x		R	
spll/mpll.pi.y		R	
spll/mpll.adder_ref		R	
spll/mpll.adder_out		R	
spll/mpll.tag_ref		R	
spll/mpll.tag_out		R	
spll/mpll.tag_ref_d		R	
spll/mpll.tag_out_d		R	
spll/mpll.seq_ref		R	
spll/mpll.seq_out		R	
spll/mpll.match_state		R	
spll/mpll.match_seq		R	
spll/mpll.phase_shift_target	ps	RW	Adjustment need for the main PLL PI loop
spll/mpll.phase_shift_current		R	
spll/mpll.id_ref		R	
spll/mpll.id_out		R	

prefix/key	unit	access	description
spll/mpll.sample_n		R	
spll/mpll.delock_count		R	
spll/mpll.dac_index		R	
spll/mpll.enabled		R	
servo/sync_source		R	Display the port that is synchronized to WR
servo/tracking_enabled		R	
servo/rtt	ps	R	Total time for a packet to go out and back on the fiber line (measured value)
servo/delay_ms	ps	R	Estimated time to go from master to slave (taking into accout fiber asymmetry and fixed delay)
servo/dtxm	ps	R	Fixed transmission delay on master (PCB + GW +SFP) provided with PTP
servo/drxm	ps	R	Fixed transmission delay on slave (PCB + GW + SFP) retrieved from local calibration
servo/dtxs	ps	R	Fixed reception delay on master (PCB + GW + SFP) provided by PTP + master bitslide
servo/drxs	ps	R	Fixed reception delay on slave (PCB + GW + SFP) retrieved from local calibration + slave bitslide
servo/fiber_asymmetry	ps	R	Total asymmetry minus the fixed delay
servo/total_asymmetry	ps	R	Time difference between going from master to slave and going back from slave to master
servo/cur_offset	ps	R	Slave to Master offset calculated by ptp/ppsi daemon
servo/cur_setpoint	ps	R	Current slave's clock phase shift value
servo/cur_skew	ps	R	The difference between current and previous estimated one-way link delay
servo/update_count		R	Number of update since WR start to lock to master
servo/mode		RWL	0: UNKNOWN 1: GM 2: MASTER 3: SLAVE_WR1 4: SLAVE_WR0
servo/sec	s	RW	TAI time in seconds
servo/nsec	ns	R	TAI time (nanoseconds parts)
servo/temp	°C	R	Temperature of the PLL
servo/leap_sec		R	
servo/gm_offset	ps	RWL	Correct time offset between PPS input and PPS output (in GM)
servo/fiber_length	m	R	Estimation of fiber length
servo/state		R	0: Uninitialized 1: SYNC_NSEC 2: SYNC_SEC 3: SYNC_PHASE 4: TRACK_PHASE 5: SYNC_IDLE 6: WAIT_OFF.STA. 7: INVALID 8: UNDEFINED
wr0/mode		R	0: UNKNOWN 1: WR_MASTER 2: WR_SLAVE 3: NON_WR 4: WR_M&S 4: WR_LISTENER
wr0/link_up		R	
wr0/is_locked		R	The GTx link has its phase locked with PLL
wr0/t2p	ps	R	
wr0/t4p	ps	R	
wr0/hw_addr		R	MAC address of the network interface
wr0/fiber_alpha		R	Fiber asymmetry
wr0/tx_count		R	Number of packets received

prefix/key	unit	access	description
wr0/rx_count		R	Number of packets transmitted
wr0/bitslide	ps	R	Delay due to bit alignment of received signal and internal clock when phase is locked (multiple of 800 ps)
wr0/sfp.vn		R	SFP: Vendor Name
wr0/sfp.pn		R	SFP: Part Number
wr0/sfp.sn		R	SFP: Serial Number
wr0/sfp.flags		R	Bit flags [7...0]
wr0/sfp.tx_wl	nm	R	SFP: Tx Wavelength
wr0/sfp.temp	°C	R	SFP: Internal temperature
wr0/sfp.vcc	V	R	SFP: Internal supply voltage
wr0/sfp.tx_bias	uA	R	SFP: TX bias current
wr0/sfp.tx_pow	dBm	R	SFP: Optical Transmission (Tx) Power
wr0/sfp.rx_pow	dBm	R	SFP: Measured Optical Reception (Rx) Power
wr1/mode		R	0: UNKNOWN 1: WR_MASTER 2: WR_SLAVE 3: NON_WR 4: WR_M&S 4: WR_LISTENER
wr1/link_up		R	
wr1/is_locked		R	The GTx link has its phase locked with PLL
wr1/t2p	ps	R	
wr1/t4p	ps	R	
wr1/hw_addr		R	MAC address of the network interface
wr1/fiber_alpha		R	Fiber asymmetry
wr1/tx_count		R	Number of packets received
wr1/rx_count		R	Number of packets transmitted
wr1/bitslide	ps	R	Delay due to bit alignment of received signal and internal clock when phase is locked (multiple of 800 ps)
wr1/sfp.vn		R	SFP: Vendor Name
wr1/sfp.pn		R	SFP: Part Number
wr1/sfp.sn		R	SFP: Serial Number
wr1/sfp.flags		R	Bit flags [7...0]
wr1/sfp.tx_wl	nm	R	SFP: Tx Wavelength
wr1/sfp.temp	°C	R	SFP: Internal temperature
wr1/sfp.vcc	V	R	SFP: Internal supply voltage
wr1/sfp.tx_bias	uA	R	SFP: TX bias current
wr1/sfp.tx_pow	dBm	R	SFP: Optical Transmission (Tx) Power
wr1/sfp.rx_pow	dBm	R	SFP: Measured Optical Reception (Rx) Power
wr1/sfp.vcc	V	R	SFP: Internal supply voltage
wr1/sfp.tx_bias	uA	R	SFP: TX bias current
wr1/sfp.tx_pow	dBm	R	SFP: Optical Transmission (Tx) Power
wr1/sfp.rx_pow	dBm	R	SFP: Measured Optical Reception (Rx) Power

6.3 Healthing

This module contains system wide configuration parameters and statistics. The list of all available parameters is below:

prefix/key	unit	access	description
pws0/status		R	Global Status. 0:ok; otherwise:error.
pws0/temperature	C	R	Temperature of PWS in ºC
pws0/v_in	V	R	Power Supply: Volts IN
pws0/v_out	V	R	Power Supply: Volts Out
pws0/power_in	W	R	Power consumed from Line in Watts
pws0/power_out	W	R	Power given from Power Supply
pws0/disable_alert		RWL	Enable/Disable the critical alert when the power supply is not plugged
pws1/status		R	Global Status. 0:ok; otherwise:error.
pws1/temperature	C	R	Temperature of PWS in ºC

prefix/key	unit	access	description
pws1/v_in	V	R	Power Supply: Volts IN
pws1/v_out	V	R	Power Supply: Volts Out
pws1/power_in	W	R	Power consumed from Line in Watts
pws1/power_out	W	R	Power given from Power Supply
pws1/disable_alert		RWL	Enable/Disable the critical alert when the power supply is not plugged
sys/uptime		R	Days:HH:MM:SS
sys/local_time		R	Current system date-time with time zone
sys/ram_total		R	Total RAM in system
sys/ram_free		R	Free RAM in system
sys/cpus		R	Number of Processors
sys/cpu_load1		R	CPU Average Load last minute
sys/cpu_load5		R	CPU Average Load last 5 minutes
sys/cpu_load15		R	CPU Average Load last 15 minutes
sys/cpu_usage	%	R	Average CPU usage in all cores
sys/hdd1_size	KB	R	Hard Disk Total (BOOT partition)
sys/hdd1_free	KB	R	Hard Disk Free (BOOT partition)
sys/hdd2_size	KB	R	Hard Disk Total (DATA partition)
sys/hdd2_free	KB	R	Hard Disk Free (DATA partition)
sys/fpga_temp	C	R	FPGA Temperature
cfg/screen_saver		RWL	0: Switch off display. 1: Show screen saver
cfg/screen_saver_delay	s	RWL	0: No screen saver. Otherwise: Time to screen saver in seconds
cfg/screen_contrast		RWL	Contrast of the screen in normal mode [0-255]
cfg/screen_contrast		RWL	Contrast of the screen in normal mode [0-255]

6.4 Misc

prefix/key	access	description
time/tz	RL	Local timezone
time/ntp_server	RL	IP of the NTP server to retrieve absolute sec (needed in GM mode)
logging/server_ip	RL	Logger server ip address
logging/server_port	RL	Logger server port
logging/protocol	RWL	Logging (rsyslogd) protocol (UDP/TCP)

6.5 Security

prefix/key	access	description
https/mode	RL	HTTP mode

6.6 wptpd

prefix/key	unit	access	description |
eth0/cfg/mode		RWL	Operation mode (master/slave/disabled)
eth0/cfg/transfer_mode		RWL	PTP transfer mode (unicast/multicast)
eth0/cfg/unicast_dest_addr		RWL	Unicast Address
eth0/cfg/unicast_negotiation		RWL	Unicast negotiation
eth0/cfg/clock_class		RWL	Clock class number (0…255)
eth0/cfg/log_announce_interval		RWL	logAnnounceInterval (log2(n))
eth0/cfg/log_sync_interval		RWL	logSyncInterval (log2(n))
eth0/cfg/log_delayreq_interval		RWL	logDelayReqInterval (log2(n))
eth0/cfg/delay_mechanism		RWL	Delay mechanism calc (E2P/P2P)
eth0/cfg/network_mode		RWL	Delay mechanism calc (E2P/P2P)
eth0/cfg/domain		RWL	Domain Number (0…255)
eth0/cfg/timescale_mode		RWL	Timescale (PTP/ARB)
eth0/cfg/priority1		RWL	Priority 1

prefix/key	unit	access	description |
eth0/cfg/priority2		RWL	Priority 2
eth0/cfg/time_traceable		RWL	TimeTraceable
eth0/cfg/frequency_traceable		RWL	FrequencyTraceable
eth0/cfg/offset_correction	ns	RWL	offset_correction
eth0/cfg/disable_bmca		RWL	Disable Best Master Clock Algorithm
eth0/cfg/timesource		RWL	Timesource (PTP/NTP/FREE-RUNNING/GPS)
eth0/cfg/clock_adjust		RWL	Hardware clock adjustments
eth0/cfg/servo_kp		RWL	Servo Kp constant
eth0/cfg/servo_ki		RWL	Servo Ki constant
eth0/cfg/www_force_refresh		RW	Force a refresh to apply the (modified) values
eth0/cfg/daemon_pid		R	Daemon_PID
eth0/stat/daemon_status		R	Daemon status (Running/Stopped/Unknown
eth0/stat/ptp_state		R	PTP state
eth0/stat/license		R	license
eth0/stat/servo_state		R	Servo state
eth0/stat/sync_mode		R	Sync Mode (one or two step)
eth0/stat/resets		R	Number of reset at LISTENING state
eth0/stat/state_transitions		R	State transitions (i.e: listening->tracking..)
eth0/stat/bm_id		R	Best Master ID
eth0/stat/bmp2		R	Best Master Priority 2
eth0/stat/bmp1		R	Best Master Priority 1
eth0/stat/bmcc		R	Best Master Clock-Class
eth0/stat/rx_announce		R	Receive announce packets
eth0/stat/rx_sync		R	Receive Sync packets
eth0/stat/rx_fu		R	Receive Follow_Up packets
eth0/stat/rx_dreq		R	Receive Delay_Req packets
eth0/stat/rx_dresp		R	Receive Delay_Rep packets
eth0/stat/tx_announce		R	Transmitted announce packets
eth0/stat/tx_sync		R	Transmitted Sync packets
eth0/stat/tx_fu		R	Transmitted Follow_Up packets
eth0/stat/tx_dreq		R	Transmitted Delay_Req packets
eth0/stat/tx_dresp		R	Transmitted Delay_Resp packets
eth0/stat/current_offset	s	R	Current offset with master
eth0/stat/mean_offset	s	R	Mean offset with master
eth0/stat/stdev_offset	s	R	Stdev offset with master
eth0/stat/current_pathdelay	s	R	Current path delay
eth0/stat/mean_pathdelay	s	R	Mean path delay
eth0/stat/stdev_pathdelay	s	R	Stdev path delay
eth1/cfg/mode		RWL	Operation mode (master/slave/disabled)
eth1/cfg/transfer_mode		RWL	PTP transfer mode (unicast/multicast)
eth1/cfg/unicast_dest_addr		RWL	Unicast Address
eth1/cfg/unicast_negotiation		RWL	Unicast negotiation
eth1/cfg/clock_class		RWL	Clock class number (0…255)
eth1/cfg/log_announce_interval		RWL	logAnnounceInterval (log2(n))
eth1/cfg/log_sync_interval		RWL	logSyncInterval (log2(n))
eth1/cfg/log_delayreq_interval		RWL	logDelayReqInterval (log2(n))
eth1/cfg/delay_mechanism		RWL	Delay mechanism calc (E2P/P2P)
eth1/cfg/network_mode		RWL	Delay mechanism calc (E2P/P2P)
eth1/cfg/domain		RWL	Domain Number (0…255)
eth1/cfg/timescale_mode		RWL	Timescale (PTP/ARB)
eth1/cfg/priority1		RWL	Priority 1
eth1/cfg/priority2		RWL	Priority 2
eth1/cfg/time_traceable		RWL	TimeTraceable
eth1/cfg/frequency_traceable		RWL	FrequencyTraceable
eth1/cfg/offset_correction	ns	RWL	offset_correction
eth1/cfg/disable_bmca		RWL	Disable Best Master Clock Algorithm
eth1/cfg/timesource		RWL	Timesource (PTP/NTP/FREE-RUNNING/GPS)
eth1/cfg/clock_adjust		RWL	Hardware clock adjustments
eth1/cfg/servo_kp		RWL	Servo Kp constant
eth1/cfg/servo_ki		RWL	Servo Ki constant
eth1/cfg/www_force_refresh		RW	Force a refresh to apply the (modified) values
eth1/cfg/daemon_pid		R	Daemon_PID
eth1/stat/daemon_status		R	Daemon status (Running/Stopped/Unknown
eth1/stat/ptp_state		R	PTP state

prefix/key	unit	access	description |
eth1/stat/license		R	license
eth1/stat/servo_state		R	Servo state
eth1/stat/sync_mode		R	Sync Mode (one or two step)
eth1/stat/resets		R	Number of reset at LISTENING state
eth1/stat/state_transitions		R	State transitions (i.e: listening->tracking..)
eth1/stat/bm_id		R	Best Master ID
eth1/stat/bmp2		R	Best Master Priority 2
eth1/stat/bmp1		R	Best Master Priority 1
eth1/stat/bmcc		R	Best Master Clock-Class
eth1/stat/rx_announce		R	Receive announce packets
eth1/stat/rx_sync		R	Receive Sync packets
eth1/stat/rx_fu		R	Receive Follow_Up packets
eth1/stat/rx_dreq		R	Receive Delay_Req packets
eth1/stat/rx_dresp		R	Receive Delay_Rep packets
eth1/stat/tx_announce		R	Transmitted announce packets
eth1/stat/tx_sync		R	Transmitted Sync packets
eth1/stat/tx_fu		R	Transmitted Follow_Up packets
eth1/stat/tx_dreq		R	Transmitted Delay_Req packets
eth1/stat/tx_dresp		R	Transmitted Delay_Resp packets
eth1/stat/current_offset	s	R	Current offset with master
eth1/stat/mean_offset	s	R	Mean offset with master
eth1/stat/stdev_offset	s	R	Stdev offset with master
eth1/stat/current_pathdelay	s	R	Current path delay
eth1/stat/mean_pathdelay	s	R	Mean path delay
eth1/stat/stdev_pathdelay	s	R	Stdev path delay

7 LCD screen menu

The WR-ZEN TP family includes a LCD screen where you can see some configuration values and the state of the device.

7.1 Screen Saver

Figure 7: Screensaver.

This is the screen you usually see when not interacting with the device. This is a screen saver screen, showing the IPs of each interface to connect to the device.

When you press INFO/CTRL button the main menu will appear.

You can configure the screen saver options with these parameters:

- **cfg/screen_saver**:
  - 1: screen saver configured.
  - 0: no showing screen saver (the screen turns off when screen_saver_dealy is reached).
- **cfg/screen_saver_delay**: time without pressing INFO/CTRL button before active screen_saver.

You can change these values through gpa_ctrl or snmp.

7.2 Main Screen

When you press INFO/CTRL button, or if the screen saver is not active, you can see the main screen.

The main screen has the next structure:

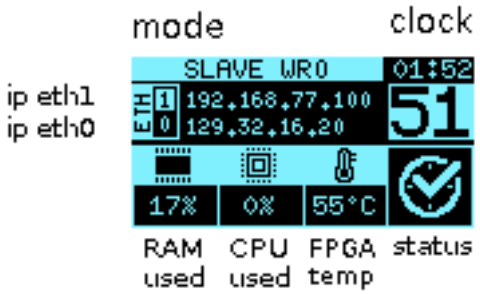


Figure 8: LCD Main screen.

This screen shows the general status of your device. Status can be:

- Ok: All is ok.
- Alert: There is at least a parameters with warning values.
- Critical: There is a problem with at least one parameter value or module.

If you press INFO/CTRL button, the main menu will show.

7.3 Main Menu

Figure 9: LCD Main menu.

To select an option, long press over INFO/CTRL button. To go to the next option, short press over INFO/CTRL button.

The different options in this menu are:

- `REVIEW PARAMETERS`: Select this submenu to review the status and values of all modules and parameters in the device.
- `NETWORK`: This submenu allows you to review the `eth0` and `eth1` info and white rabbit network status.
- `WR STATS`: It shows info about white rabbit statistics.
- `MANAGEMENT`: It allows to execute simple prefedined actions like rebooting the device or restore the default configuration.

1. REVIEW PARAMETERS

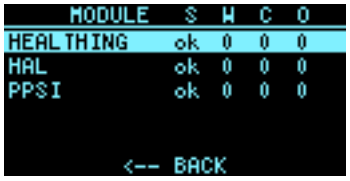


Figure 10: LCD modules menu.

This screen shows the global status of the modules running in the device.

- **S**: Status:
  - OK: all is ok.
  - ! (warning): at least one parameter in warning range.
  - X (critical): at least one parameter in critical range, or there is a problem with the module.
- **W**: All parameters in warning range.
- **C**: All parameters in critical range.
- **O**: Out of sync.

You can view details of each module by pressing long INFO/CTRL button over the module you want to review.

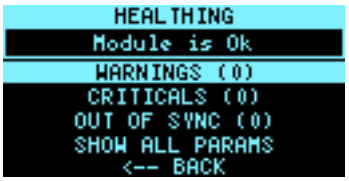


Figure 11: LCD module details.

The module status description is shown. The different options to select in this screen are:

- `WARNINGS(n)`: select this option to review parameters in warning range.
- `CRITICALS(n)`: select this option to review parameters in critical range.
- `OUT OF SYNC(n)`: select this option to review parameters which are out of sync.
- `SHOW ALL PARAMS`: select this option to review all parameters in the module.

Where n is the number of parameters in that condition.

If you select any of the above options the screen will show information about the parameters that caused the special status. If there is 0 parameters with a special status, it will no enter the sub-menu.

Figure 12: LCD parameters details.

To exit this screen, long press over INFO/CTRL button. To go to next parameter, short press over INFO/CTRL button. When the last parameter was reached, a short press over INFO/CTRL exit this screen, returning to the previous menu.

2. NETWORK

In this section you can review the parameters regarding the network interfaces of the device.

Figure 13: Network options.

The options are:

- `ETH0`: information about copper interface Ethernet 0.
- `ETH1`: information about copper interface Ethernet 1.
- `WR0`: information about fiber interface White Rabbit 0.
- `WR1`: information about fiber interface White Rabbit 1.

Figure 14: Network menu.

When you select `ETH0` or `ETH1` option the details of the Ethernet interface will appear:

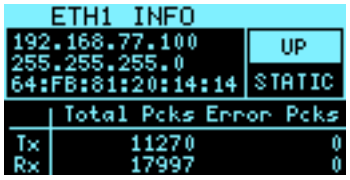


Figure 15: LCD Ethernet details.

When you select WR0 or WR1 option the details of the White Rabbit interface will appear.

Figure 16: LCD White Rabbit details.

3. WR STATS

Figure 17: LCD White Rabbit menu option

In this sub-menu you can review the information about white rabbit:

Figure 18: LCD White Rabbit statistics

4. MANAGEMENT

In this section you can perform simple prefedined actions like rebooting the device or restore the default configuration. –>

The options are:

- POWER OFF: Power off the device in a safe way.
- REBOOT: Reboot the device.
- REBOOT WR: Reboot white rabbit.
- TOGGLE M/GM: Activate/Deactivate Grand-Master mode.
- FACTORY RESET: Reset the value of all parameters to factory configuration.

Once you select an option by long pressing INFO/CTRL button, a count down is shown:

Figure 19: LCD counting down

By press INFO/CTRL button you can cancel the execution of the option selected. When countdown reaches 0, the option will be executed.

8 Tools

8.1 Web Interface

The web interface can be accesed through the IP address of any of the management ports. The dashboard is shown as main page showing information relative to the device like the IP adresses, uptime, version, serial number, temperature, ...

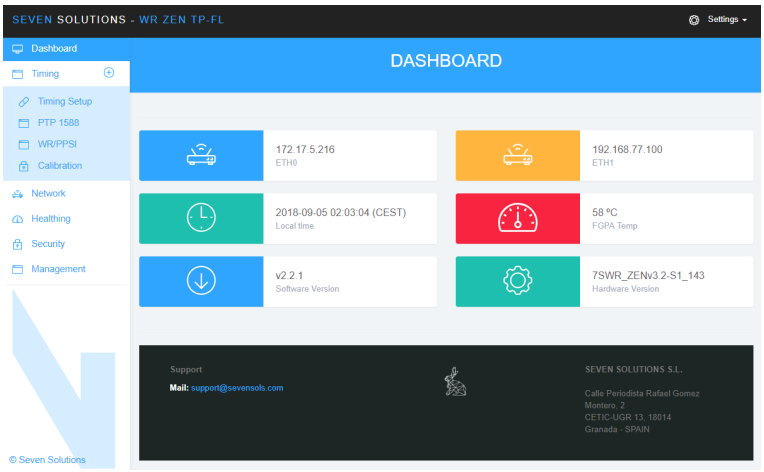


Figure 20: Web interface dashboard.

To access to the configuration and extended parameters information you need to log in. The default user is *root* and default password is also *root*.

From the top right side you can find the *Settings* where you can change the password to have administration rights and an expert mode with additional configuration options for advanced users.

A full menu on the left side menu is shown with the next options:

1. Dashboard
2. Network
3. WR/PPSI
4. PTP
5. Calibration
6. Outputs/Sync
7. Healthing
8. Misc
9. Update Firmware

When any parameter is modified in any section, at the end there are usually two buttons: *Save* and *Apply*. Save stores the values introduced on memory but it does not apply them. *Apply* does not store the values on memory and are only applied on the fly. Because some parameters cannot be changed on the fly, sometimes a reboot is needed to apply them.

8.1.1 Dashboard

The dashboard is the main page accesible even without credentials. It shows basic information of the device like the IP address of the management ports, serial number, version and basic system parameters.

8.1.2 Network

It shows the configuration and statistics of the management network interfaces. IP address and netmask can be configured and DHCP activated or deactivated.

8.1.3 WR/PPSI

It shows the configuration and statistics of White Rabbit protocol controlled by the PPSi module. Timing performance information and configuration can be found. From the Servo tab, the general mode of operation can be changed. The options are:

- *GM*: Grand-Master. Selecting this option will make the device lock to an external reference. 10 MHz and PPS inputs are required.
- *MASTER*: Both SFP ports act as master ports.
- *SLAVE_WR0*: This is the default mode. The SFP port 0 is configured as slave and port 1 as master.
- *SLAVE_WR1*: The SFP port 1 is configured as slave and port 0 as master.
- *UNKNOWN*: This value is set if the device is unable to determine the mode of operation.

There is also tabs for WR0 and WR1 that refer to the SFP 0 and SFP 1 ports. These tabs show information related to each port like the SFP connected, the transmission coefficient of the fiber, hardware address, etc.

8.1.4 PTP

It shows the configuration and statistics from PTP module. It gives information regarding the PTP protocol and the parameters associated with its timing performance. In the *Configuration* tab, the PTP protocol interface can be selected. The configuration options are:

- *masteronly*: It will act as a PTP master on the selected interface.
- *slaveonly*: It will act as a PTP slave on the selected interface. If the WR/PPSI configuration is also set as *slave_wr0* or *slave_wr1* it will give an error as a device cannot be slave of two different instances. It needs the WR/PPSI configured as *GM* or *MASTER*.
- *masterslave*: The device will look for a master in its network to get its timing from it (acting as slave), but if none is found, it will act as master.

8.1.5 Calibration

All calibration values regarding the White Rabbit protocol ar shown here. There is support for storing up to 5 different SFPs per device and 5 different fibers in the database. If you prefer to use specific calibration values for a port, not taking into account which SFP transceiver you are using, you can select the “WR” box.

Setting to *1* the *Force Usr* field, will force that port to use the values specified below at boot.

8.1.6 Outputs/Sync

Information and advanced configuration regarding the timing outputs. The information on this menu will depend on the specific device on the WR-ZEN TP family. The number of options available are incremental depending on the device.

On all devices you can configure the PPS timing options and select higher ratio versions of it.

WR-ZEN TP

On this device you can find two modules named *A* and *B*. The modules are relative to the corresponding ports *PPx/CLK A* and *PPx/CLK B*. Enabling the *AB xPPS Mode* gives the possibility to select 10MHz, 5MHz, PPS and IRIG-B. Disable it and only 5MHz and 10MHz can be selected.

If *AB xPPS Mode* is disabled the signals come directly from a PLL being much more stable in terms of jitter.

WR-ZEN TP-32BNC

On this device you can find four modules named *A*,*B*, *C* and *D*. The modules are relative to the corresponding BNC connectors. Each module manages 8 BNC connectors. Enabling the *AB xPPS Mode* gives the possibility to select 10MHz, 5MHz, PPS and IRIG-B on modules *A* and *B*. Disable it and only 5MHz and 10MHz can be selected. Modules *C* and *D* have enable the xPPS mode by default and cannot be changed.

If *AB xPPS Mode* is disabled the signals come directly from a PLL being much more stable in terms of jitter.

In modules *C* and *D* the termination can be selected between 50Ω and 1MΩ. On modules *A* and *B* is 50Ω by hardware.

8.1.7 Healthing

It shows information of the current status of the system, power supply and fans. On the two power supply tabs (PWS0 and PWS1) you can disable the unplugged cable alert. Otherwise, if you have only one power supply connected, there will be a permanent critical alert and the error led will be on.

8.1.8 Misc

There are three tabs: *Time*, *Info* and *Management*

On *Time* the timezone and the NTP server address can be selected. The WR-ZEN TP can obtain the date and time information through the NTP server.

On the *Info* tab there is information regarding the WR-ZEN TP product. This information would be very important when contacting Seven Solutions Team for support.

On the *Management* tab there are two buttons to restart and reset the device to the factory default configuration.

8.1.9 Update firmware

This menu has already been covered on the section Update Procedure.

8.2 wrz_config

The `wrz_config` is a tool that allows to make changes on the configuration of the device and to save them so they can be applied on boot. The user can store values permanently and they will be loaded on boot. You can change the value of any parameter defined in the section Running Modules and Parameters with the access type `L`.

In order to use this configuration tool you need to run the command `wrz_config`. A interface like the next figure will appear.

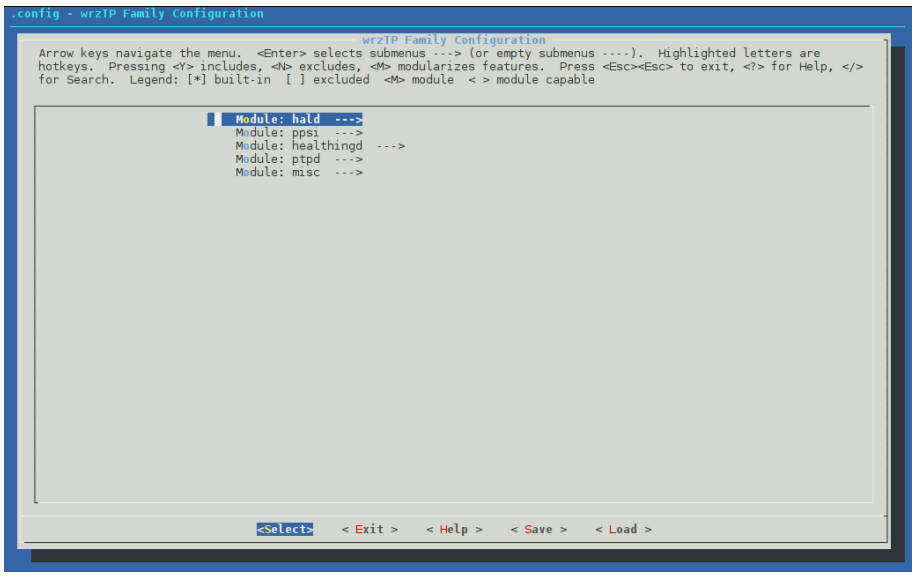


Figure 21: Configuration interface

Here you have access to each module. Inside each module you will find sub-sections and on each sub-section you will find its parameters. You can change the value of these parameters like in the next figure.

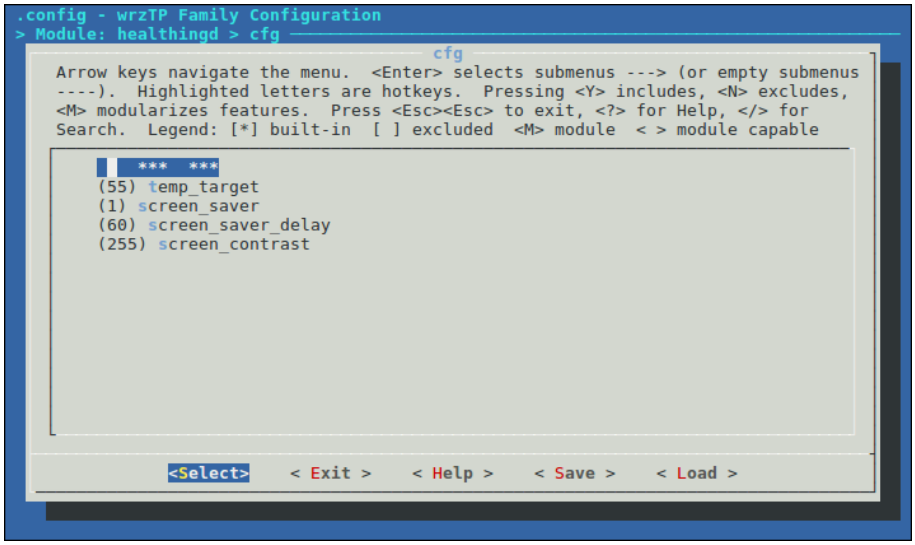


Figure 22: Configuration parameters

If you encounter graphical issues with `wrz_config` (i.e, when using `minicom` instead of `picocom`), try the alternative color scheme with `wrz_config -n`.

8.3 gpa_ctrl

`gpa_ctrl` is a tool to query for the values of the module parameters running in a WR-ZEN device.

```
root@wrztp-020:/# gpa_ctrl -h
usage: gpa_ctrl [OPTIONS] [<mod_key> [<prm_key> [<write_val>]]]
where OPTIONS are:
    -h Show this little help message
    -q quiet mode
    -v verbose mode
    -V very verbose
    -f force syncing
```

- `-f` force syncing: force connection to the module and refresh the read data. If the module is down you can review the values before it went down. But if you select `-f` the module must be running to show you the actual parameters.

If you run `gpa_ctrl` without any argument you will get a list of all parameters in all modules as defined in the section Running Modules and Parameters.

If you would like to show only the parameters of a module in particular run `gpa_ctrl <module_name>`. Where `<module_name>` can be:

- `hald`
- `ppsi`
- `healthingd`

If you wish to know the value of a parameter, run `gpa_ctrl <module_name> <param_name>`. In this case, `<param_name>` can be any parameter defined in the section Running Modules and Parameters. For example, if we would like to know the target temperate of the FPGA:

```
root@wrztp-020:/# gpa_ctrl healthingd cfg/temp_target
55
```

If a parameter is writeable you can change the value by doing `gpa_ctrl <module_name> <param_name> <new_value>`. For example, if we want to change the value of `temp_target` in `cfg/healthingd` to 60:

```
root@wrztp-020:/# gpa_ctrl healthingd cfg/temp_target 60
```

Now if you read the value you will get:

```
root@wrztp-020:/# gpa_ctrl healthingd cfg/temp_target
60
```

The values written using this tool will be lost after rebooting the device.

Verbose options -v and -V are useful if you want additional parameter info:

```
root@wrztp-020:/# gpa_ctrl -v healthingd sys/fpga_temp
sys/fpga_temp: 56
```

```
root@wrztp-020:/# gpa_ctrl -V healthingd sys/fpga_temp
gpa_ctrl#D: main:144: gpa-shmem-2: ID=2 argc=3
gpa_ctrl#D: gpa_mod_create_user:281: healthingd @[0x36e49040:0x36eeb700]
sys/fpga_temp : 56 C ; s:GEN, v:i16-x11, a:x01, o:x01
                   (FPGA Temperature)
```

8.4 wr_mon

This tool allows you to check for the basic `ppsi` parameters dynamically. Once executed, it runs continuously refreshing the parameters every second. The information shown has the next format:

```
time
 * sec : 13004 (s)
 * nsec : 705285968 (ns)

wr0 : WR_SLAVE locked TX:2324 RX:8105 (SFP:AXGE-1254-0531 @1310nm)
wr1 : WR_MASTER locked TX:5854 RX:0

servo
 * Servo state : TRACK_PHASE
 * Synchronization source : wr0
 * Phase tracking : 1
 * Round-trip time (mu) : 49876103 (ps)
 * Master-slave delay : 24949222 (ps)
 * Master-PHY TX delays : 170 (ps)
 * Master-PHY RX delays : 12681 (ps)
 * Slave-PHY TX delays : 202429 (ps)
```

```
 * Slave-PHY RX delays : 230663 (ps)
 * Total link asymmetry : -22341 (ps)
 * fiber_asymmetry : -38064 (ps)
 * Clock Offset : 2 (ps)
 * Update counter : 1592

temp
 * temp : 44 (°C)
 * sfp.temp : 0.000000 (°C)
 * sfp.temp : 0.000000 (°C)
```

8.5 .config file

In order to save parameters that will be applied after rebooting the device, you should create a file `.config` on the `/root/` folder of the device. Then, write in it the parameters and values to be applied with the next format:

```
CONFIG_<MODULE_NAME>_<PARAM_PREFIX>_<PARAM_NAME>="<VALUE>"
```

Where the modules and parameters are the ones defined in the section Running Modules and Parameters. An example to activate the Grand-Master mode would be:

```
CONFIG_PPSI_SERVO_MODE_GM=y
```

All parameters defined on the file must have the access type `L` as they will be loaded on boot.

9 The Timing interfaces

9.1 Grandmaster mode

In a White Rabbit network, we can distinguish 3 types of configuration (Fig. 23):

- The WR nodes (WR slaves): these elements get the timing information from a WR master.
- The WR master: this element receive the timing from another WR master and forward this information to a WR slave.
- The WR grandmaster: The top element of the network that follows the clock from external 1PPS and 10 MHz signal. It also receive the ToD from another protocol.

The WR-ZEN TP can be configured in each three types: As a simple WR slave, as a WR master when used in daisy chain configuration, and finally as a GrandMaster.

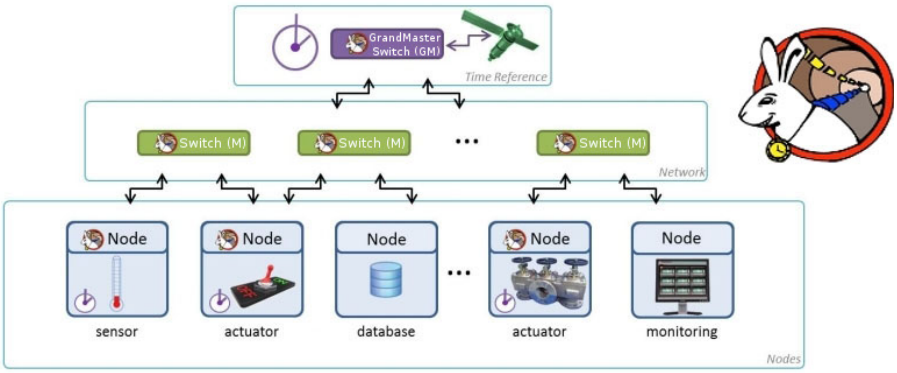


Figure 23: WR network scheme.

9.1.1 The PPS/10MHz

In order to setup the WR in grandmaster you first need to connect the `10MHz In` and `PPS In` SMA inputs to an external reference such as a GPS, Atomic Clock, etc. These signals must meet the following electrical requirements:

- PPS input: 50Ω, LVTTL or TTL levels
- 10 MHz input: 50Ω, TTL/LVTTL/sine ([1 – 5]V rms)
- t_{SETUP} (PPS-to-10MHz) > 20 ns (see Fig. 24)

The 10 MHz is used to produce the internal frecuency of WR (62.5MHz/125MHz) and the PPS is used only to ensure the alignment of the edges of both clocks at inter-second boundary.

The actual inter-second boundary is the 10 MHz rising edge after the rising edge of the PPS pulse (see Fig. 24).

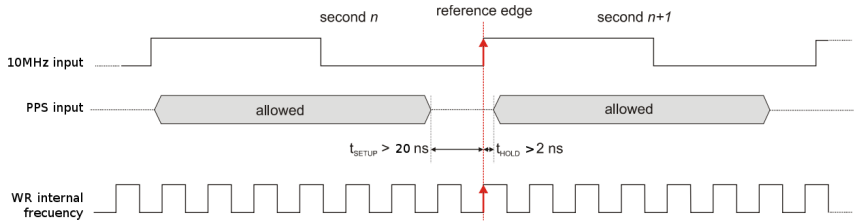


Figure 24: External reference requirements.

***Notes**: PPS input is sampled only once at boot (when the PLL is locking), then it will not be used at all.*

***Warning:** The 10 MHz input must not "jump" (i.e. have temporary period different than 100 ns). Some GPS receivers were reported to produce such 10 MHz with such artifacts – these devices cannot be used with WR*

9.1.2 Checking the PPS/10Mhz input

When GrandMaster mode is set, the device will try to lock to the external reference, althought it does not exist. This action will start in the state `WR_FREERUNNING`. If both required inputs (PPS and 10MHz inputs) are fine, the servo_state will step to `GM_LOCKED`. **It means that PPS and 10Mhz are correct**. However, sometimes the system will end at `GM_ERROR`, which in the most of the cases the inputs are wrong (bad frequency or the PPS is not correct).

9.1.3 Supplying Time of Day

Once the grandmaster mode is locked to the external reference the WR-ZEN TP will have a relative notion of time, but it still needs to retrieve the time of day (ToD) information from another source. The protocols actually supported for this operation are the following:

9.2 NTP

When the WR-ZEN TP is in the GrandMaster mode, it is recommended to retrieve the ToD from the Network Time Protocol (NTP). For this purpose, the ntp client can be configured to get the time from the server. The server ip can be specified in a two different ways:

- Using the web interface (**www**). Accessing to the web interface, section `Timing>Timing Setup` and editing the NTP server entry. Finally, save the configuration and reboot.
- Using the CLI, via SSH or serial port. Editing the `/root/.config` file and adding(or editing) the entry: `CONFIG_MISC_TIME_NTP_SERVER=xxx.xxx.xxx.xxx`

After that, you will need to reboot the system to make the changes take effect.

9.3 Clock TTL configuration

By default the SMA connector `CLK TTL` outputs a 10MHz square digital signal. Please, contact Seven Solutions if you want to output any other specific frequency synchronized with White Rabbit.

9.4 Timing outputs (BNC/DB9)

All devices from the WR-ZEN family can output one 10MHz/PPS from the SMA connectors. In order to get more outputs the WR-ZEN TP and/or the WR-ZEN TP32BNC can be use to distribute xPPS/CLK signal to many equipment in a rack cabinets.

9.4.1 xPPS module

By default the xPPS module from the FPGA output 1PPS, but this module also allows to output:

- 1 PPS
- 10 PPS
- 1000 PPS
- IRIG-B 004

It also allows to modify the coarse offset of the signal and the width of the xPPS pulse.

***NOTE**: any change will modify the configuration of all blocks. It is not possible to change independently a block of output.*

9.4.2 A,B,C,D blocks for WR-ZEN TP-32BNC

The WR-ZEN TP-32BNC as four blocks (A,B,C,D) with 8 BNCs outputs each blocks.

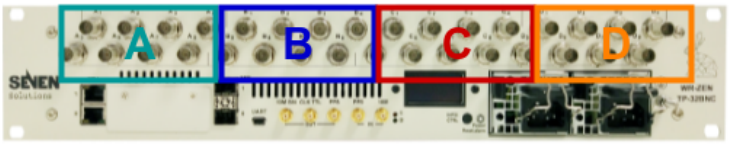


Figure 25: A,B,C,D blocks

The corresponding functional diagram:

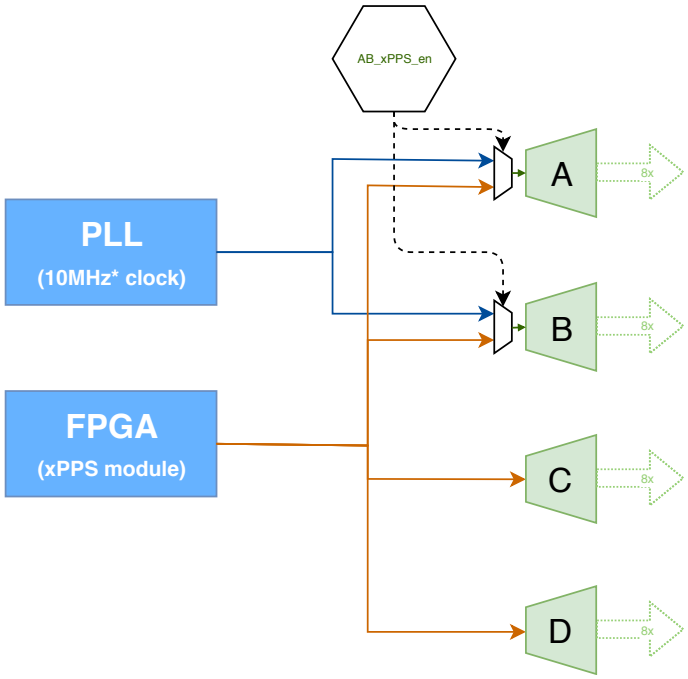


Figure 26: 32-BNC blocks diagram

By default, A and B blocks are configured to output a very clean, very low jitter 10MHz[1] signal from the PLL. Then, C and D are configured to output 1PPS coming from the FPGA (xPPS module).

If you need to output xPPS signal for A or B blocks you should enable the `AB_xpps_en` parameter. Please be aware that this latest parameter act on both blocks (A & B) and thus you will not be able to output a clean frequency from any of A or B block.

***WARNING**: Until v2.x, the user could select to output 5MHz/10MHz from the xPPS module (FPGA) but this operational mode should **NOT** be used anymore as its behavior vary between equipments.*

The recommended configurations for A,B,C,D block are to provide:

- 1PPS x 16 & 10MHz x 16 (default)
- 1PSS x 32 (on all BNCs blocks)

1PPS x 16 & 10MHz x 16

This is the default configuration and it should be similar on your web interface (with the delay already calibrated for your hardware):

[1] The PLL is configured to output 10MHz. You might contact support@sevensols.com to check if your custom requirement are compatible with our device.

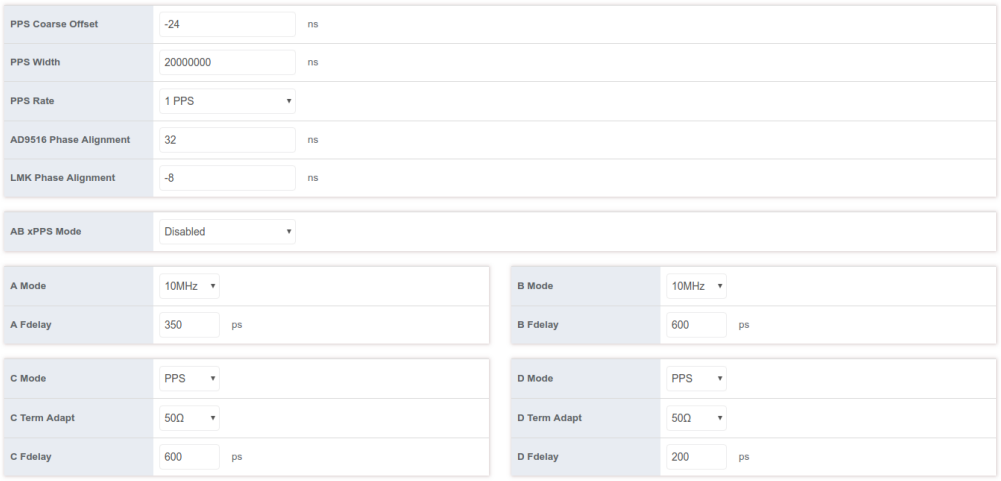


Figure 27: TP32 with 10MHz on A & B and 1PSS C & D

Or if you prefer to force this configuration from command line you can use `gpa_ctrl` tool:

```
##Force to use 10MHz CLK PLL on A, B (instead of xPPS signal from FPGA)
gpa_ctrl hald sync/AB_xpps_en 0

##Force to use 1PPS on C & D blocks
gpa_ctrl hald sync/pps_xtimes 1PPS
gpa_ctrl hald sync/C_mode PPS
gpa_ctrl hald sync/D_mode PPS
```

***NOTE**: As AB_xpps_en is enabled, the value of A_mode and B_mode is useless and could be set to anything.*
***WARNING**: Do not output frequency on C,D blocks*

1PPS x 32

In another case if the user only want to have 32x PPS he need to configure as:

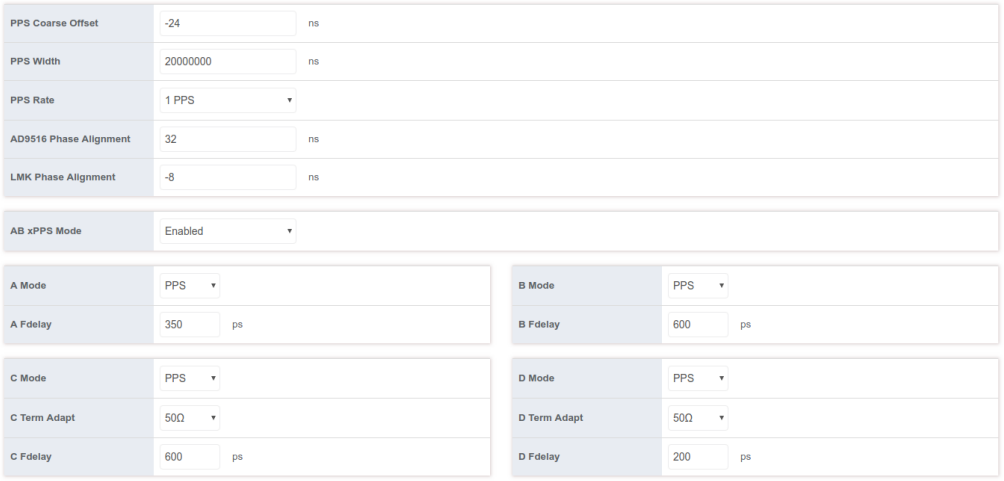


Figure 28: TP32 with 1PSS on A,B,C,D

or run the following command:

```
##Set xPPS to output 1PPS of 20ms
gpa_ctrl hald sync/pps_xtimes 1PPS
gpa_ctrl hald sync/pps_width 20000000

## Enable xPPS on A & B blocks
gpa_ctrl hald sync/AB_xpps_en 1
```

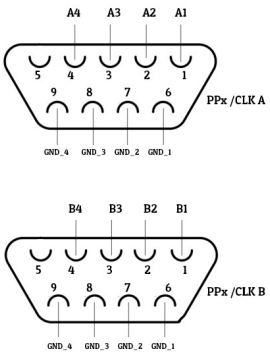
```
gpa_ctrl hald sync/A_mode PPS
gpa_ctrl hald sync/B_mode PPS

## Enable xPPS on C & D blocks
gpa_ctrl hald sync/C_mode PPS
gpa_ctrl hald sync/D_mode PPS
```

9.4.3 A,B blocks for WR-ZEN TP

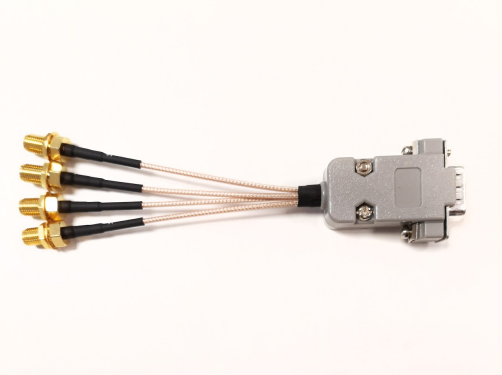
Similary as the WR-ZEN TP-32BNC, the WR-ZEN TP is able to output xPPS/CLK signals through the DB9 connector.

Figure 29: TP with A & B blocks

If you then need SMA outputs, a DB9-4xSMA adapter can be ordered from Seven Solutions.



(a) Pinout of the A & B DB9 blocks



(b) DB9 (M) - 4xSMA (F) adapter

Figure 30: WR-ZEN TP PPx/CLK DB9 connectors

10MHz x 4 & 1PPS x 4

This is the default configuration but if you need to come back to this state you could set-it up from the web interface as follow:

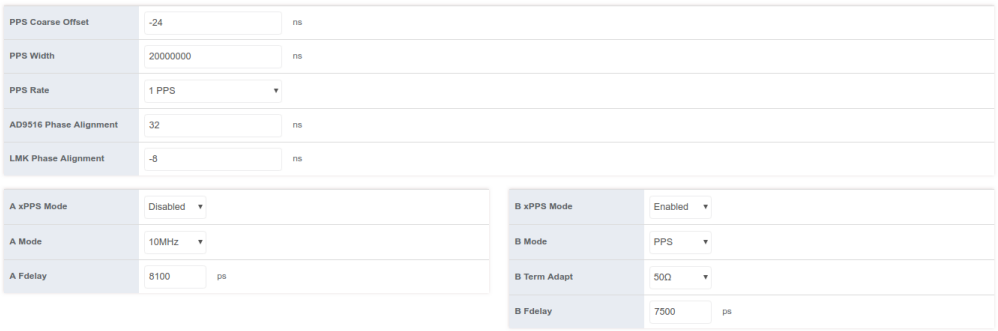


Figure 31: TP with 10MHz on A & 1PSS on B

or if you only have access through the terminal you should run:

```
##Set xPPS to output 1PPS of 20ms
gpa_ctrl hald sync/pps_xtimes 1PPS
gpa_ctrl hald sync/pps_width 20000000

## Disable xPPS on A block, use CLK output
gpa_ctrl hald sync/A_xpps_en 0
gpa_ctrl hald sync/A_mode 10M

## Enable xPPS on B block
gpa_ctrl hald sync/B_xpps_en 1
gpa_ctrl hald sync/B_mode PPS
```

1PPS x 8 (2 x 4)

If you want to ouput 1PPS on both A & B blocks in order to get up to 1PPS on 8 additional SMA you need to configure the web as follow:

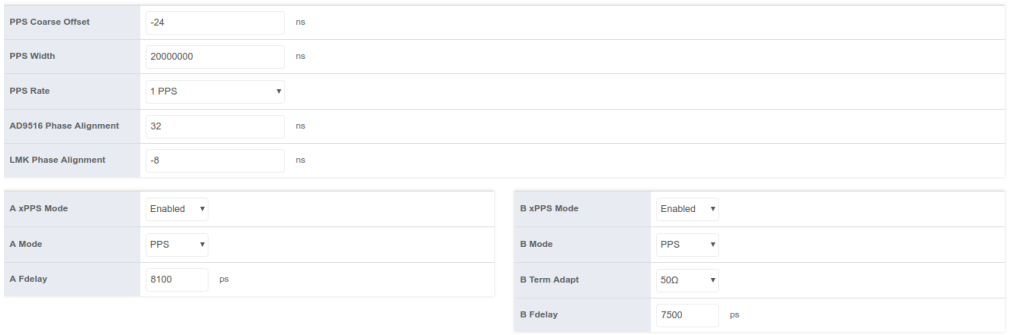


Figure 32: TP with 1PSS on A & B

Or you need to execute the following command:

```
##Set xPPS to output 1PPS of 20ms
gpa_ctrl hald sync/pps_xtimes 1PPS
gpa_ctrl hald sync/pps_width 20000000

## Enable xPPS on A block
gpa_ctrl hald sync/A_xpps_en 1
gpa_ctrl hald sync/A_mode PPS

## Enable xPPS on B block
gpa_ctrl hald sync/B_xpps_en 1
gpa_ctrl hald sync/B_mode PPS
```

9.5 WR-ZEN TP: TOD0 & TOD1

In order to recover ToD (Time Of Day) from the WR-ZEN TP and plug it into your own device you might need to check that the pinout (Figure 33) of the RJ45 cable is compatible with your device. If it is not the case you can contact support@sevensols.com in order to get your customize cable for your device.

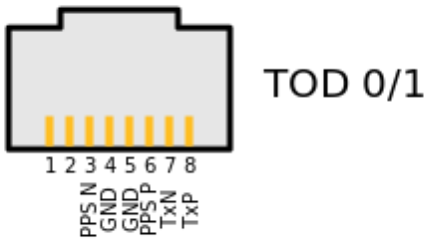


Figure 33: Pinout of the TODx ports (RS422-NMEA)

Then the NMEA protocol is sent over the Tx{P/N} pins and to retrieve it you should configure the input to read the UART with the following configuration: `57600 bps, 8N1`. By default we have configured the WR-ZEN TP to provide *TOD Cisco Message Frame Structure* which is detailed below:

Position	Name	Description
1	Satellite availability	`*` = valid, `!` = not valid
2	Revision	Fixed to `A`
3	Modified Julian Date	Number of days since Nov 17, 1858
4	Date	Format: `+"%y/%m/%d"`
5	Time	Format: `+"%H:%M:%S"`
6	Time zone offset	`+/-` hours from UTC0 or `0`
7	Leap Seconds	`0` or TAI-UTC offset
8	Latitude	`0` or given by GNSS module
9	Longitude	`0` or given by GNSS module
10	Altitude	`0` or given by GNSS module
11	Alarm (severity src cause)	severity is given by the first word (`EV` = event, `MN` = minor, `MJ` = major, `CL` = critical, `--` = not implemented), then the second word is the source and finally the last word is the cause

An example of ToD output obtained using `picocom -b 57600 /dev/ttyXXXX`

```
!.A.58492.19/01/09.13:16:54.+1.37.0.0.0.0.-- -- --
!.A.58492.19/01/09.13:16:55.+1.37.0.0.0.0.-- -- --
!.A.58492.19/01/09.13:16:56.+1.37.0.0.0.0.-- -- --
!.A.58492.19/01/09.13:16:57.+1.37.0.0.0.0.-- -- --
```

10 SNMP

10.1 Introduction

10.1.1 Naming and legend

- *"module"*: A virtual component that connects parameters to their corresponding resources (e.g, daemon, command tools, etc.) and provides the way to read/write the parameters.
- *"parameter"*: A generic structure to declare all kind of parameters in a common way. With this common structure we ease the sharing process.

10.1.2 OS support

This guide has been written using Ubuntu 16.04 but should be valid for other distribution with few changes. Please be aware that we will only provide support for Ubuntu 16.04 (and other LTS version).

10.1.3 Install snmp package

To install the snmp packages needed by your platform you should execute the following commands

```
sudo apt-get install snmp snmp-mibs-downloader libsnmp-dev
```

10.2 Download and install the MIB

10.2.1 Get the MIB

To query and set information for any WR-ZEN TP you can download the MIB from http://sevensols.com/downloads/SEVEN-PRODUCT-MIB.txt.

10.2.2 Store the MIB

The snmp tools like net-snmp translate numeric object identifies (OIDs) into textual object identifiers using the MIB description files.

Each object that is managed by snmp has a unique OID.

If you are in a Linux Environment the default paths to store MIBs are:

1. `$HOME/.snmp/mibs`
2. `/usr/local/share/snmp/mibs`

If you use net-snmp distribution, you can find out which directories are used to look for the MIBs by running the next command:

```
net-snmp-config --default-mibdirs
```

If that does not work it is probably because your distribution did not repackage net-snmp-config. You can use this instead:

```
snmptranslate -Dinit_mib .1.3 2>&1 | grep MIBDIR
```

10.2.3 Add SEVEN-PRODUCT-MIB to SNMP

Before using any snmp tools you have to tell SNMP to load our MIB. You can do it in two different ways:

Using configuration file

You can put the following lines in a `snmp.conf` file. This file can be placed in the system-wide configuration location (e.g. `/usr/local/share/snmp.conf`) or in a personal file (e.g. , `$HOME/.snmp/snmp.conf`). The system-wide configuration file location will depend on how Net-SNMP was built on your system. Run `net-snmp-config -snmpconfpath` to display the list of paths. Once you get the correct file just append the following line:

```
mibs +SEVEN-PRODUCT-MIB
```

If you want to load all mibs in your system you can put:

```
mibs +ALL
```

Using the MIBS environment variable:

```
MIBS=+SEVEN-PRODUCT-MIB
export MIBS
```

Now we are ready to run snmp commands in our wrzTimeProvider.

10.2.4 Check if installed

After you copy the SEVEN-PRODUCT-MIB.txt to the appropiate directory, you can test if your system recognizes it by executing the command:

```
snmptranslate -m +SEVEN-PRODUCT-MIB -IR -On wrzGeneralStatus
```

If it is ok you should get the OID for the wrzGeneralStatus:

```
.1.3.6.1.4.1.49029.1.10.1
```

You can also check it by executing:

```
snmptranslate -On SEVEN-PRODUCT-MIB::wrzTimeProvider
```

If you want a complete description of wrzTimeProvider subtree you can run:

```
snmptranslate -Tp -Td SEVEN-PRODUCT-MIB::wrzTimeProvider
```

10.3 MIB description and examples of use

In this section we are going to explain the main parts of our MIB definition.

The IANA assigns to *Seven Solutions* company the following subtree: `.1.3.6.1.4.1.49029`. Then we have created the subtree `xxx.49029.1` for all our WR products.

Finally, we have assigned number 10 to wrzTimeProvider. So the MIB subtree for this product is: `.1.3.6.1.4.1.49029.1.10`

In this subtree you can find:

- wrzGeneralStatus (.1.3.6.1.4.1.49029.1.10.1)
- wrzModuleTable (.1.3.6.1.4.1.49029.1.10.2)
- wrzParamTable (.1.3.6.1.4.1.49029.1.10.3)
- wrzParamXTable (.1.3.6.1.4.1.49029.1.10.4)

10.3.1 wrzGeneralStatus

(.1.3.6.1.4.1.49029.1.10.1)

This is a scalar object. This object has the general status of the wrzTimeProvider. We query all the modules and check their states. This object shows the current status. All columns are read-only.

Status	Name	Status Description
1	**ok**	All modules is up and all parameters of the modules are ok.
200	**modNotShMem**	At least one module share memory file is not found and there aren't modules down.
201	**modDown**	At least one module is not running
202	**modUnknownStatus**	At least one module is in unknown status and there aren't modules down or without share mem.
203	**modParamsNotSync**	At least one parameter in a module is out of sync and there aren't modules in previous states.
204	**modParamsCritical**	At least one parameter in a module is critical and there aren't modules in previous states.
205	**modParamsWarning**	At least one parameter in a module is in warning range and there aren't modules in previous states.

If you want to get the general status you have to run:

```
snmpget -v 3 ${USER_CREDENTIALS} ${HOST} SEVEN-PRODUCT-MIB::wrzGeneralStatus.0
```

For example (from the WR-ZEN device):

```
snmpget -v 3 -u adminSNMP -l authPriv -a MD5 -x DES -A adminSNMPpass -X adminSNMPpass localhost
    SEVEN-PRODUCT-MIB::wrzGeneralStatus.0
```

You can get something like this:

```
SEVEN-PRODUCT-MIB::wrzGeneralStatus.0 = INTEGER: modParamsWarning(205)
```

10.3.2 wrzModuleTable

(.1.3.6.1.4.1.49029.1.10.2)

This table has a list of all modules running in the device. With this table you can review the general status of the WR-ZEN TP. If one of the parameters in the module has a warning or error you can review it watching column wrzModuleStatus. The rows of this table are preloaded. One row per module.

Before explaining the different objects in this table we are going to retrieve the table.

To get this table you can do:

```
snmptable -v 3 ${USER_CREDENTIALS} ${HOST} SEVEN-PRODUCT-MIB::wrzModuleTable -Ci -Cb
```

Notes:

- *Ci show the index of each entry in the table.*
- *Cb remove prefix in column name (wrzParam).*

You should get an output similar to this:

```
index Name Status TotalNotSync TotalWarnings TotalCriticals LastBoot LastSync
    0 hald ok 0 0 0 1970-01-01 04:18:07 1970-01-02 01:38:48
    1 ppsi ok 0 0 0 1970-01-01 03:46:56 1970-01-02 01:38:48
    2 healthing ok 0 0 0 1970-01-02 01:34:39 1970-01-02 01:38:48
```

Another way to retrieve all this node is using snmpwalk:

```
snmpwalk -v 3 ${USER_CREDENTIALS} ${HOST} SEVEN-PRODUCT-MIB::wrzModuleTable
```

You'd get something like this:

```
SEVEN-PRODUCT-MIB::wrzModuleName.0 = STRING: hald
SEVEN-PRODUCT-MIB::wrzModuleName.1 = STRING: ppsi
SEVEN-PRODUCT-MIB::wrzModuleName.2 = STRING: healthing
SEVEN-PRODUCT-MIB::wrzModuleStatus.0 = INTEGER: ok(1)
SEVEN-PRODUCT-MIB::wrzModuleStatus.1 = INTEGER: ok(1)
SEVEN-PRODUCT-MIB::wrzModuleStatus.2 = INTEGER: ok(1)
SEVEN-PRODUCT-MIB::wrzModuleTotalNotSync.0 = INTEGER: 0
SEVEN-PRODUCT-MIB::wrzModuleTotalNotSync.1 = INTEGER: 0
SEVEN-PRODUCT-MIB::wrzModuleTotalNotSync.2 = INTEGER: 0
SEVEN-PRODUCT-MIB::wrzModuleTotalWarnings.0 = INTEGER: 0
SEVEN-PRODUCT-MIB::wrzModuleTotalWarnings.1 = INTEGER: 0
SEVEN-PRODUCT-MIB::wrzModuleTotalWarnings.2 = INTEGER: 0
SEVEN-PRODUCT-MIB::wrzModuleTotalCriticals.0 = INTEGER: 0
SEVEN-PRODUCT-MIB::wrzModuleTotalCriticals.1 = INTEGER: 0
SEVEN-PRODUCT-MIB::wrzModuleTotalCriticals.2 = INTEGER: 0
SEVEN-PRODUCT-MIB::wrzModuleLastBoot.0 = STRING: 1970-01-01 04:18:07
SEVEN-PRODUCT-MIB::wrzModuleLastBoot.1 = STRING: 1970-01-01 03:46:56
SEVEN-PRODUCT-MIB::wrzModuleLastBoot.2 = STRING: 1970-01-02 01:34:39
SEVEN-PRODUCT-MIB::wrzModuleLastSync.0 = STRING: 1970-01-02 01:38:24
SEVEN-PRODUCT-MIB::wrzModuleLastSync.1 = STRING: 1970-01-02 01:38:24
SEVEN-PRODUCT-MIB::wrzModuleLastSync.2 = STRING: 1970-01-02 01:38:25
```

If you want to get a value of a cell you can tell snmp the name of the column you get followed by a dot (.) and the row index. For example if you want to get the Status of soft_daemon the command is:

```
snmpget ${USER_CREDENTIALS} ${HOST} SEVEN-PRODUCT-MIB::wrzModuleStatus.128
```

Now we are going to describe the different columns in this table:

- **index**: The id of the module.
- **wrzModuleName**: The name of the module.
- **wrzModuleStatus**: The current status of the module. A table with the diferrent possible status is shown below.
- **wrzModuleTotalNotSync**: Number of parameters that belong to the module that are unavailable to sync with its resource. A parameter not synchronized could be a warning about a hardware hardware resource that is not responding or is out of order.
- **wrzModuleTotalWarnings**: Number of parameters that belong to the module in a warning range.
- **wrzModuleTotalCriticals**: Number of parameters that belong to the module in a critical range.
- **wrzModuleLastBoot**: Time of last start of the module.
- **wrzModuleLastSync**: Time of last sync of the parameters in the module.

Index	Name	Status Description
1	**ok**	Module is up and all parameters of the module are ok.
200	**modNotShMem**	Unable to find shared memory file.
201	**modDown**	Module is not running.
202	**modUnknownStatus**	Bug state, if not matching with other conditions.
203	**modParamsNotSync**	At least one parameter is out of sync.
204	**modParamsCritical**	At least one parameter of the module is in critical range and there aren't parameters out of sync.
205	**modParamsWarning**	At least one parameter of the module is in warning range, and there aren't parameters out of sync or in critical range.

10.3.3 wrzParamTable

(.1.3.6.1.4.1.49029.1.10.3)

This table has a list of parameters running in the WR-ZEN TP. This table is related with wrzModuleTable. Each parameter has an index related to its parent module. The rows of this table are preloaded. The values

of the parameters can be setting through SNMP if this parameter allows write operations. Each parameter is identified by its module, prefix and parameter index. This table preloads the relationship between indexes and prefixName and paramName. Once a parameter is assigned to an index in this table it belongs to this parameter forever. All columns are read-only excepting wrzParamVal.

Before explaining the different objects in this table we are going to retrieve the table.

To get this table you can do:

```
snmptable -v 3 ${USER_CREDENTIALS} ${HOST} SEVEN-PRODUCT-MIB::wrzParamTable -Ci -Cb
```

For example with default password and running from wrzTimeProvider:

```
snmptable -u adminSNMP -l authPriv -a MD5 -x DES -A adminSNMPpass -X adminSNMPpass localhost SEVEN-
    PRODUCT-MIB::wrzParamTable -Ci -Cb
```

Notes:

- *Ci show the index of each entry in the table.*
- *Cb remove prefix in column name (wrzParam).*

You should get an output similar to this:

```
 index ModuleName PrefixName Name Type Val Unit Status LastSync
 0.1.1 hald eth0 link_up u8 0 ok 1970-01-02 01:40:00
 0.1.2 hald eth0 ethaddr v_ 64:FB:81:20:10:14 ok 1970-01-01 04:18:07
 0.1.3 hald eth0 ipv4.addr v_ 129.32.16.20 ok 1970-01-01 04:18:07
 0.1.4 hald eth0 ipv4.netmask v_ 129.32.16.20 ok 1970-01-01 04:18:07
 0.1.5 hald eth0 is_dhcp u8 0 ok 1970-01-02 01:40:00
 0.1.6 hald eth0 speed str none ok 1970-01-02 01:40:00
 0.1.7 hald eth0 tx_packets u32 0 ok 1970-01-02 01:40:00
 0.1.8 hald eth0 rx_packets u32 0 ok 1970-01-02 01:40:00
 0.1.9 hald eth0 tx_bytes u32 0 ok 1970-01-02 01:40:00
0.1.10 hald eth0 rx_bytes u32 0 ok 1970-01-02 01:40:00
0.1.11 hald eth0 tx_errors u32 0 ok 1970-01-02 01:40:00
0.1.12 hald eth0 rx_errors u32 0 ok 1970-01-02 01:40:00
 0.2.1 hald eth1 link_up u8 1 ok 1970-01-02 01:40:00
...
```

Another way to retrieve all this node is using snmpwalk:

```
snmpwalk -v 3 ${USER_CREDENTIALS} ${HOST} SEVEN-PRODUCT-MIB::wrzParamTable
```

If you want to get a value from a cell you can tell snmp the name of the column you get followed by a . and the row index. For example if you want to get the Val of link_up in prefix 'eth0' and in the module hald the command is:

```
snmpget -v 3 ${USER_CREDENTIALS} ${HOST} SEVEN-PRODUCT-MIB::wrzParamVal.0.1.1
```

With default credentials and running from wrzTimeProvider:

```
snmpget -v 3 -u adminSNMP -l authPriv -a MD5 -x DES -A adminSNMPpass -X adminSNMPpass localhost
    SEVEN-PRODUCT-MIB::wrzParamVal.0.1.1
```

To set a value of a parameter:

```
snmpset -v 3 ${USER_CREDENTIALS} ${HOST} SEVEN-PRODUCT-MIB::wrzParamVal.0.1.5 s "1"
```

With default credentials and running from wrzTimeProvider:

```
snmpset -v 3 -u adminSNMP -l authPriv -a MD5 -x DES -A adminSNMPpass -X adminSNMPpass localhost
    SEVEN-PRODUCT-MIB::wrzParamVal.0.1.5 s "1"
```

IMPORTANT

All parameters are set as STRINGs regardless of the the internal type. The agent converts it to the proper type

Now we are going to describe the different columns in this table:

- **index**: The index has 3 fields:
  - module_index: related to the owner module of this parameter.
  - prefix_index: related to the prefix of this parameter.
  - parameter_index: a parameter in a module and in a list.
- **wrzParamModuleName**: The name of the owner module of this parameter.
- **wrzParamPrefixName**: The name of the prefix of this parameter.
- **wrzParamName**: The name of the parameter.
- **wrzParamType**: The internal type of the parameter.
- **wrzParamVal**: The value of the parameter.
- **wrzParamUnit**: The unit of the parameter.
- **wrzParamStatus**: The status of the parameter. A table with the diferrent possible status is shown below.
- **wrzParamLastSync**: Time of last sync of the parameter.

Status	Name	Status Description
1	**ok**	Parameter is ok.
200	**modNotShMem**	Cannot find share memory file.
201	**modDown**	Module is not running.
202	**modUnknownStatus**	Bug state, if not matching with other conditions.
301	**paramNotFound**	Parameter is not found in the module. This error happens if an parameter is load for SNMP but module is not supporting this parameter now.
302	**paramUnknown**	Bug state, if not matching with other conditions.
303	**paramNotSync**	Parameter is not synced with its resource.
304	**paramCritical**	Parameter is in critical range and sync.
305	**paramWarning**	Parameter is in warning range and sync, and not in critical range.

10.3.4 wrzParamXTable

(.1.3.6.1.4.1.49029.1.10.4)

This table is an augmented table with extra columns describing the parameters. This table allows change config values, which are loaded when a module starts.

```
snmptable -v 3 ${USER_CREDENTIALS} ${HOST} SEVEN-PRODUCT-MIB::wrzParamXTable -Ci -Cb
```

With default credentials and running from the device:

```
snmptable -v 3 -u adminSNMP -l authPriv -a MD5 -x DES -A adminSNMPpass -X adminSNMPpass localhost
    SEVEN-PRODUCT-MIB::wrzParamXTable -Ci -Cb
```

Notes: ** Ci show the index of each entry in the table. * Cb remove prefix in column name (wrzParamX).*

```
 index ModuleName PrefixName Name Access ConfigVal SetRange WarningRange CriticalRange Description
128.1.1 soft_daemon . linemsg RW
128.1.2 soft_daemon . verbosity RW
128.1.3 soft_daemon . iteration R
128.1.4 soft_daemon . coeff_a RWL -1 Coefficient a in r=a*r+b
128.1.5 soft_daemon . coeff_b RWL 2 [-10.000000,10.000000] [-20.000000,-15.000000]
128.1.6 soft_daemon . result R
128.2.1 soft_daemon ONE_DIR msleep RL 4000 [300,30000] [4000,7000] [100,3999] Sleep time between
    each iteration
```

To modify a Config Value:

```
snmpset -v 3 -u adminSNMP -l authPriv -a MD5 -x DES -A adminSNMPpass -X adminSNMPpass localhost
    SEVEN-PRODUCT-MIB::wrzParamXConfigVal.128.1.4 s "5"
```

Now we are going to describe the different columns in this table:

- **index**: The index has 3 fields:
  - module_index: related to the owner module of this parameter
  - prefix_index: related to the prefix of this parameter
  - parameter_index: a parameter in a module and in a list.
- **wrzParamXModuleName**: The name of the owner module of this parameter.
- **wrzParamXPrefixName**: The name of the prefix of this parameter.
- **wrzParamXName**: The name of the parameter
- **wrzParamXAccess**: The allowed access.
  - `R (1)` - The value of the parameter is read-only
  - `RW (2)` - The value of the parameter is read-write
  - `RL (3)` - The value of the parameter is read-only but it's possible to change config value
  - `RWL (4)` - The value of the parameter is read-write and its value is loaded from dot-config file
- **wrzParamXConfigVal**: The config value of a parameter. This value is loading from .config file to a parameter when modules boot.
  - When you change a value, you can watch the valued marked with an `*`. This means that the Config Value is not been saved yet to its dot-config file.
  - To save all the values marked with `*` use wrzCommands. Review this parameter to understand how to save and apply.

Example if you change the config value to ./coeef_a and ONE_DIR/msleep

```
 index ModuleName PrefixName Name Access ConfigVal SetRange WarningRange CriticalRange Description
128.1.1 soft_daemon . linemsg RW
128.1.2 soft_daemon . verbosity RW
128.1.3 soft_daemon . iteration R
128.1.4 soft_daemon . coeff_a RWL 5 * Coefficient a in r=a*r+b
128.1.5 soft_daemon . coeff_b RWL 2 [-10.000000,10.000000] [-20.000000,-15.000000]
128.1.6 soft_daemon . result R
128.2.1 soft_daemon ONE_DIR msleep RL 2000 * [300,30000] [4000,7000] [100,3999] Sleep time between
    each iteration
```

- **wrzParamXSetRange**: Range to set values. If blank, it is the range of the type.
- **wrzParamXWarningRange**: Range of values defined as warning.
- **wrzParamXCriticalRange**: Range of values defined as critical
- **wrzParamXDescription**: Description of parameter.

11 Security and firewall

The WRZ family implements from the `wr-zynq-os-v2.2.2` the netfilter/iptables set of tools (with the respective kernel support) for adding rules to the firewall.

By default, the configuration is empty, so no traffic is filtered, but the user is able to create a new set of rules according to the needings.

1. Check that iptables are empty (by default):

```
root@wrztpfl-xxx:~# iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination

Chain FORWARD (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination
```

2. Create a new rule. As example, we will configure the firewall to drop all input traffic which protocol is icmp (pings) to our device, by the interface eth0

```
root@wrztpfl-xxx:~# iptables -A INPUT -i eth0 -p icmp -j DROP
```

If the rule has been added successfully, you will see the following output:

```
root@wrztpfl-xxx:~# iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination
DROP icmp -- anywhere anywhere

Chain FORWARD (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination
```

3. Export the current configuration to a file to make the changes permanent.

The current configuration file is saved at `/media/data/usr/local/etc/iptables` and it will be placed at `/etc/iptables` at start.

```
root@wrztpfl-xxx:~# mkdir -p /media/data/usr/local/etc/ && iptables-save > /media/data/usr/local/
    etc/iptables
```

4. Reboot and repeat the step 1 to ensure that all configuration has been loaded correctly.

From this point, no one could do ping to the interface `eth0`. All incoming traffic will be rejected and dropped by the kernel.

12 Misc

12.1 Logging

It could happens that the system reports a fail, warning or just an info message. These kind of messages are sent from the different daemons to the system (linux logger subsystem). All the logs will be located at `/var/log/systemlog`. This log can be easily configurable to be sent remotely (specifing ip,port and protocol), using the `rsyslog` daemon to an external rsyslog server. All of these parameters are part of the `misc` daemon, so you can configure it by changing the following entry in the `/root/.config` file (or creating it if it does not exist):

```
CONFIG_MISC_LOGGING_SERVER_IP="195.195.195.195"
CONFIG_MISC_LOGGING_SERVER_PORT="777"
CONFIG_MISC_LOGGING_PROTOCOL_UDP=y
CONFIG_MISC_LOGGING_PROTOCOL_TCP=y
```

Save it and then perform a reboot of the system.

You can also modify the ip/port and protocol using the web interface.

12.1.1 WRZ_LOGDUMP

The logdump is composed by `/var/log/systemlog` and another set of important logs in order to debug a problem by our technical service. You can generate a logdump with all neccesary information from the command line interface by using the wrz_logdump tool:

```
root@wrztpfl-xxx:~# wrz_logdump
Removing older log...
Saving log... please wait

Log saved successfully: /var/www/downloads/logdump-tpfl-xxx-20180917-054931.tar.gz
```

And then, you can transfer it to your computer using, for example, the `scp` tool:

```
user@user-PC:~/$ scp root@ip_address:/var/www/downloads/logdump-tpfl-xxx-20180917-054931.tar.gz .
logdump-tpfl-xxx-20180917-054931.tar.gz 100% 98KB 97.5KB/s 00:00
```

Alternatively, you are able to generate the wrz_logdump from the web interface, going to `Management > Misc > Dump Log`.

This logdump contains the following list of information:

- The content of `/boot` and `/media` partitions.
- The main configuration from the `/root/.config`.
- Info about the interfaces, ip, netmask, packets, status...
- Information about interrupts (two measurements).
- Information related with memory status.
- The systemlog `/var/log/systemlog`.
- Information about uboot, versions...

Providing this log to us, we will know more about your system and will be easier to help you.

13 Update Procedure

There are two different ways to update the software and firmware of the device. From the web interface and from SSH.

13.1 From web interface

First, you will need to access the management portal by accessing the IP of the device from a browser. Once you log in (default user: root, default password: root), you are redirected to the dashboard. Navigate to the *Update Firmware* menu. From it, the update file can be drag and dropped or browse your files to select it. Then, press *Flash* to start the process.

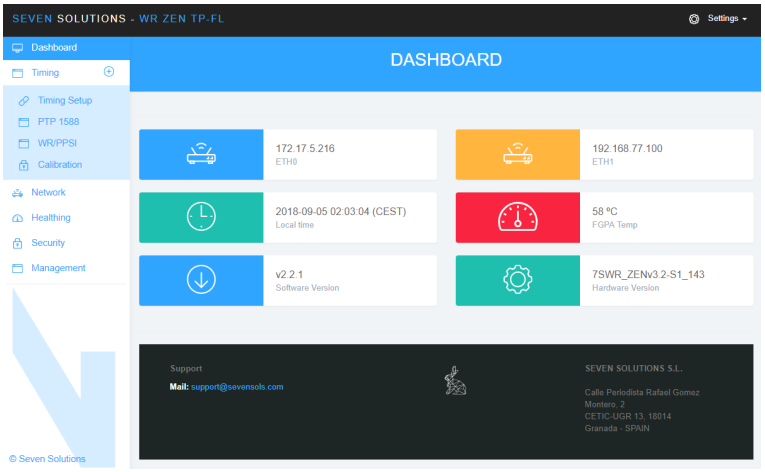


Figure 34: Web interface dashboard.

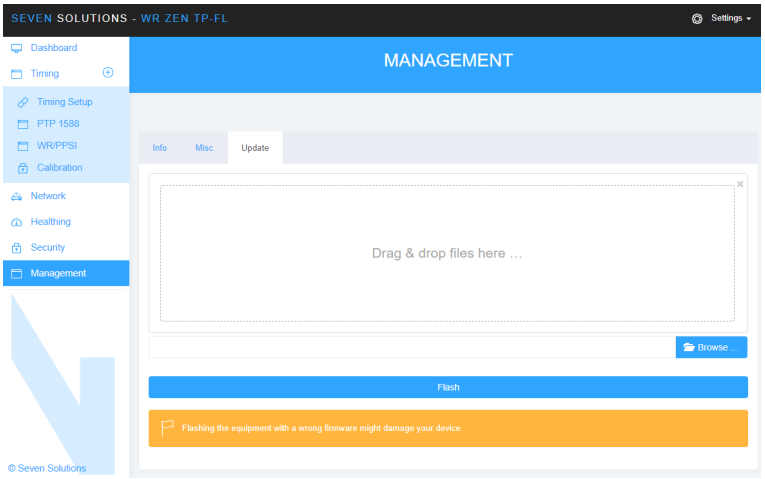


Figure 35: Web interface update menu.

13.2 From SSH or command line

First you need to place the update file provided by *Seven Solutions* on the `/update/` folder of the device and rename it as `update.tar`. Then, erase the file `updated_ok` and perform a reboot with the command `reboot`.

14 References

- wrz-tp.pdf : Datasheet for the WR-ZEN Timeprovider
- wrs-quick-startup.pdf : User manual documentation of the tools.
- whiterabbitsolution : White Rabbit as a complete timing solutions
- WR Wiki : White Rabbit Wiki on ohwr.org
- wr-calibration : Wiki page about the WR calibration.
- SFPs Wiki : Type of SFP supported by the WR-ZEN TP
- OHWR : Link to the Open Hardware Repository, where the WR project is hosted.
- FMC cards : List of the FMC cards developed in the WR framework.
- wrc : White Rabbit PTP Core User's Manual.
- wrc-shell : List of wrc shell commands.
- PPSI : PPSi Manual.
- wrc.pdf : White Rabbit PTP Core User's Manual.
- RMA Instructions : Explain the steps to follow to return a product.

A Appendix: Advanced features

A.1 The "custom mount" directories

In case you want to keep the "official" firmware but add some custom configurations or put some new tools you can use the second ext4 partition on the SD drive mounted as /media/data.

Below the list of the directories to create and their mount points in the operating system after a reboot:

Directory in SD drive	Mount points	Comments
`/media/data/update`	`/update`	Always created, used to update the FW
`/media/data/root`	`/root`	Root files where we can store the configuration
`/media/data/usr/local/bin`	`/usr/local/bin`	For custom binaries tools
`/media/data/usr/local/sbin`	`/usr/local/sbin`	For custom script
`/media/data/usr/local/lib`	`/usr/local/lib`	For custom libraries
`/media/data/usr/local/etc`	`/etc/`	Create symbolic links into the /etc dir

NOTES: *These directories are mounted only during the boot procedure so you might want to reboot your system after creating them into the /media/data/.*

A.2 Recovery Mode

If you have "bricked" the device during an update/customization procedure it should enter by itself in a recovery mode so that you have a access to a minimal linux firmware that allows to:

- Reflash the device with another firmware
- Recover configuration (if possible)
- Clean/format SD remotely

Once the device has been booted in recovery mode, it should apply the network configuration previously saved in the `.config` file. However it might be possible that the `.config` is corrupted/unreadable, in this case the device will be accessible using the default network parameters defined at the beggining of the document.

In order to repair the firmware you could try to:

- In case you have customized your environment, you might want to delete all the files in the `/media/data` partition except the update folder.
- Login to the web interface (WMI) and follow the update procedure with the firmware of your choice.

But you can also contact `support@sevensols.com` to get more help.

A.2.1 Manual recovery mode

In case you need to manually enter the recovery mode please execute the following steps:

1. Reboot the WR-ZEN TP
2. Press the `CTRL/Info` button around 5s while Uboot is loading.
3. Wait until the factory failsafe image is loaded from QSPI dataflash (This can take more than 1 minute).

You can also try to execute the command from Uboot console (USB-UART)

1. Press any key when seeing

```
Loading wr7shw preboot...

U-Boot 201X.xx-wr7s-vX.X (Jun 25 2018 - 16:07:12) ZENv3
WR_ZEN-vx.x-Sxx_xxx
Hit any key to stop autoboot: 0
```

and execute

```
wr7s-uboot> env run recoveryboot
```

until you get access to a linux terminal.

A.3 Failsafe Mode

In case a miss configuration of the device deny the access to the device, one can manually reset the configuration to default factory value by:

1. Reboot the WR-ZEN TP
2. Press the `CTRL/Info` button more than 15s while Uboot is loading.
3. Hold until reset factory message appears on LCD or error LED is switched off
4. Wait until the device reboot with default factory parameters.

A.4 Timing outputs properties

In order to provide a good overview of the timing & electrical properties of the different outputs provided by all the device in the WR-ZEN family, a serie of oscillocope screenshots have been captured.

A.4.1 SMA outputs

The connection to the oscillocope are:

- C1 (yellow): CLK TTL (10MHz by default)
- C2 (cyan): 10M SIN
- C3 (magenta): 1PPS

The level of the signals are provided in the next figure (with `10M SIN` centered in 0V):

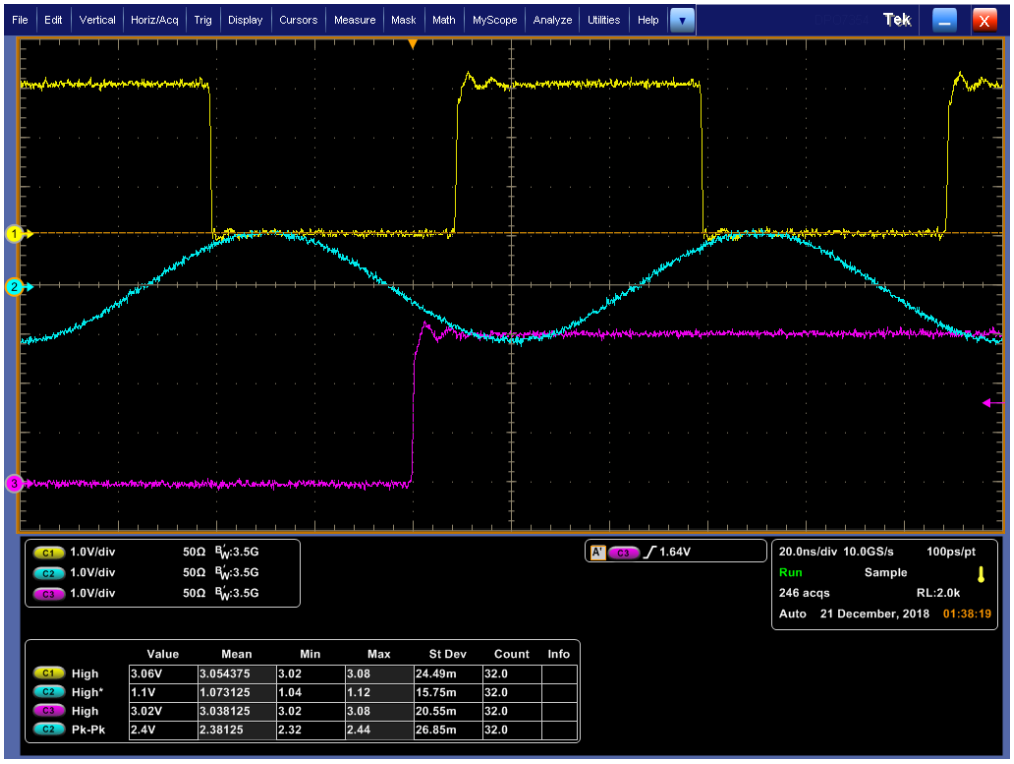


Figure 36: Levels of SMA outputs

And the timing properties (delay and duty cycle) should be:

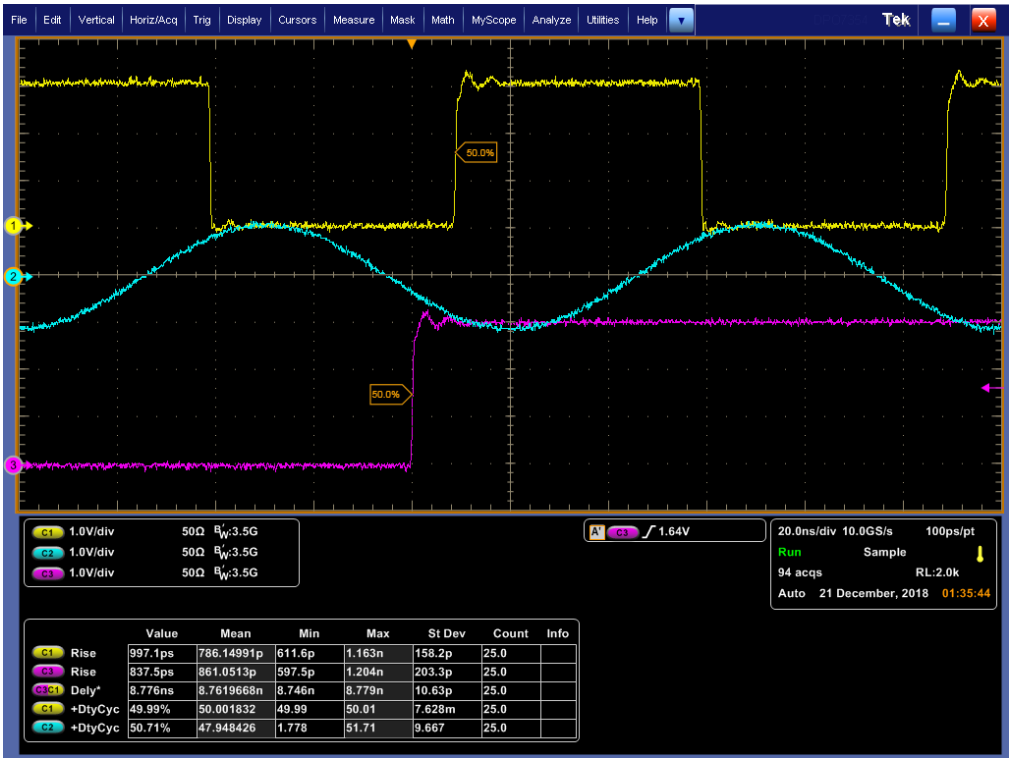


Figure 37: Delay & Duty cycle of SMA outputs

A.4.2 WR-ZEN TP-32 BNC

- C1 (yellow): 10MHz (`AB_xpps_en=0`)
- C2 (cyan): 10MHz (`AB_xpps_en=0`)
- C3 (magenta): 1PPS with 50Ω enabled
- C4 (green): 1PPS with 50Ω disabled

Electrical properties:

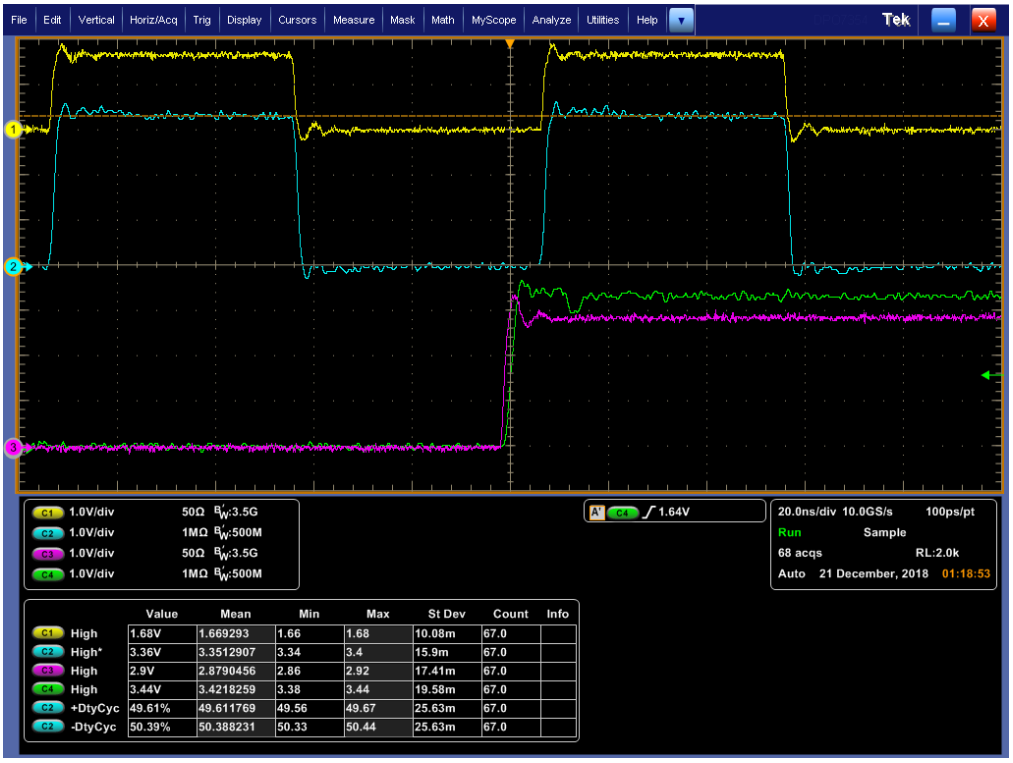


Figure 38: TP32 with 1PSS on A,B,C,D

Timing properties:

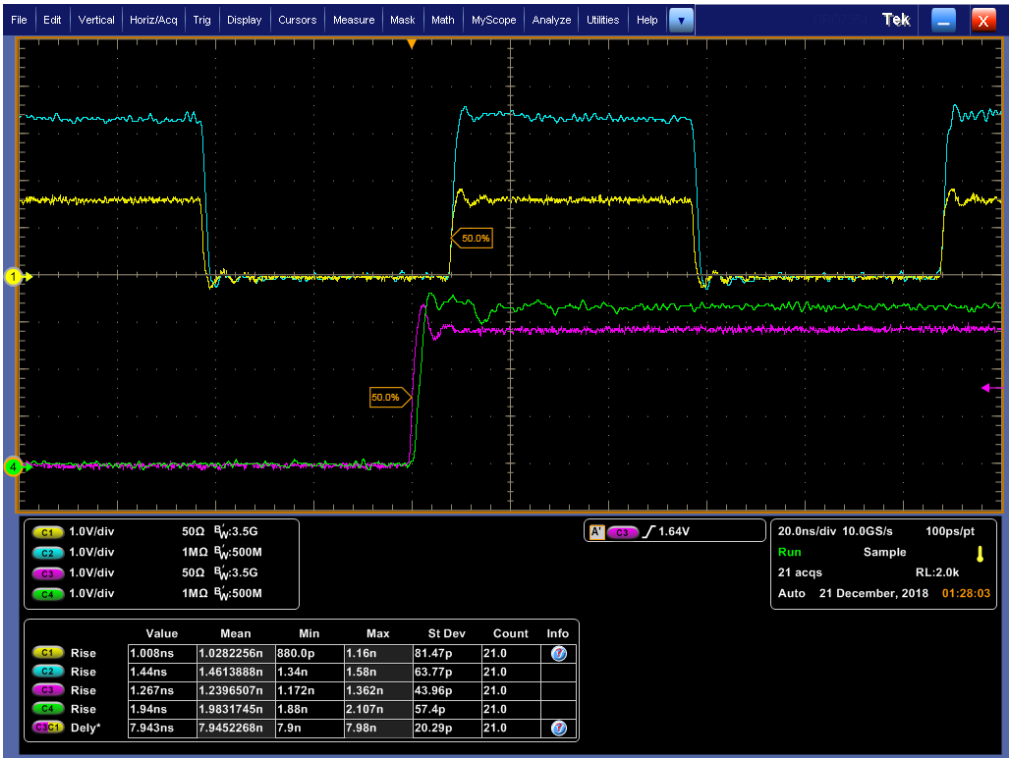


Figure 39: TP32 with 1PSS on A,B,C,D

A.4.3 WR-ZEN TP: DB9

First with "50Ω adapted" enabled on B block and a termination of 50Ω on the oscilloscope we get:

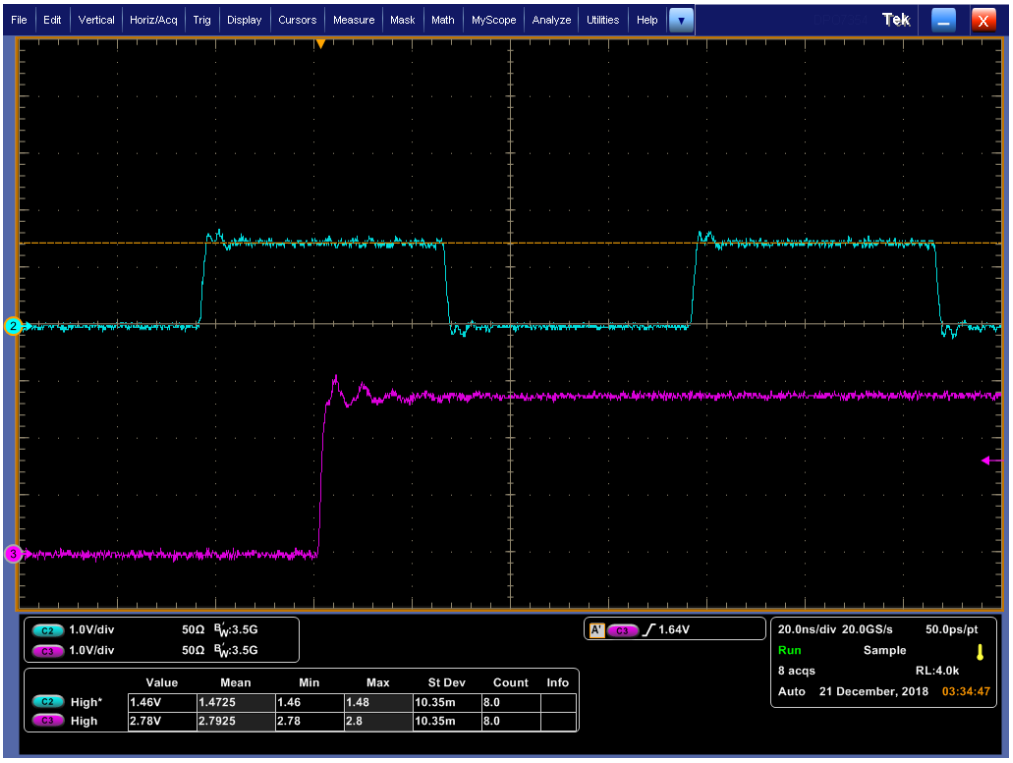


Figure 40: TP A & B levels with 50Ω

Then as we select a 1MΩ termination (no termination) on the oscilloscope we also disable the parameter "50Ω adapted" on the B block. The output signals will thus look like:

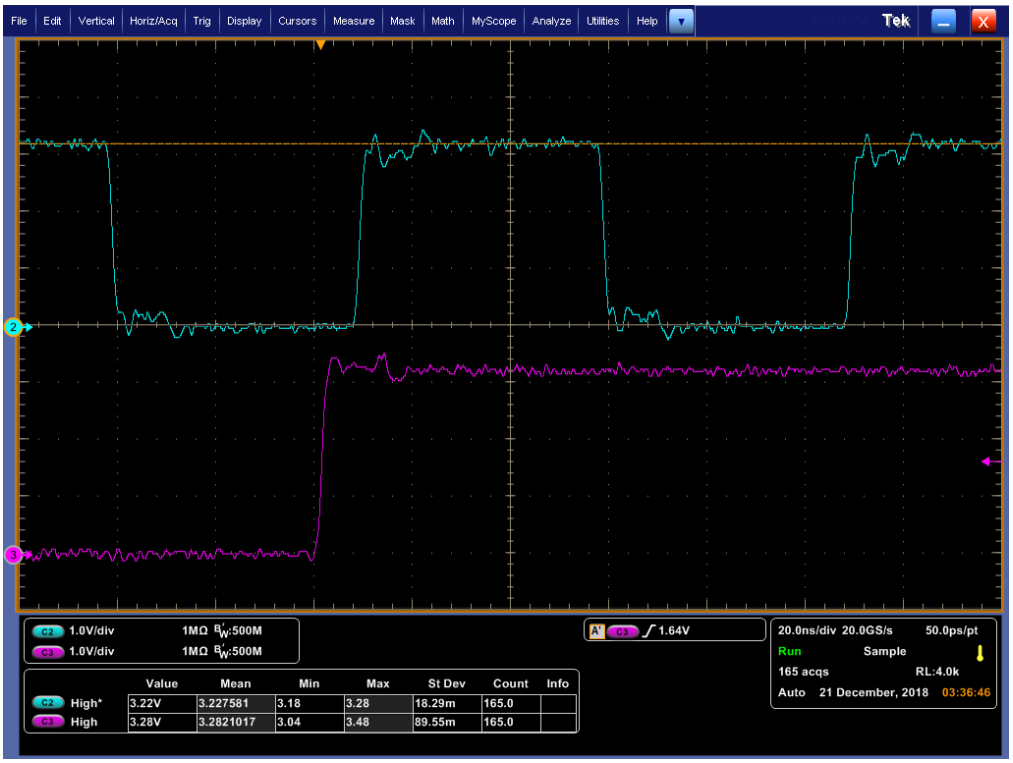


Figure 41: TP A & B levels with 1MΩ

And the timing properties (delay and duty cycle) should be:

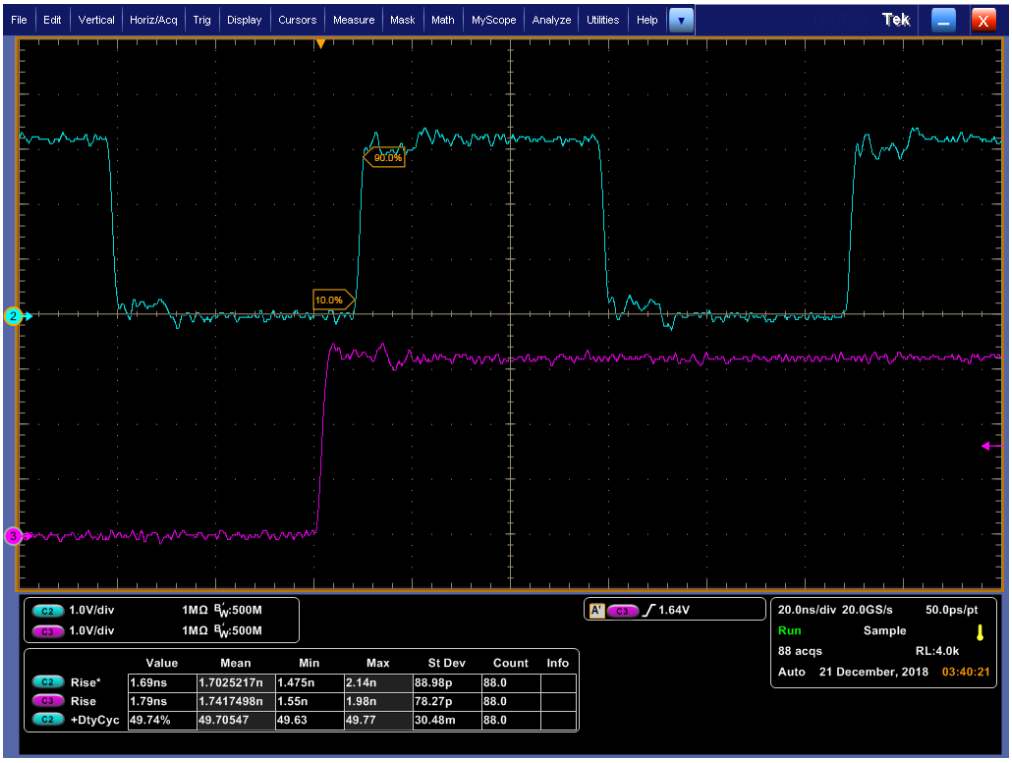


Figure 42: TP A & B delay & rising time

B Appendix: Misc.

B.1 Safety Notes

Warning Do not block the air vents which are located on the front panel of the WR-ZEN TP, the internal temperature might increase and damage your device.

Warning To increase the lifetime of your device it is recommended to use it in a controlled temperature environment and limit to the ambient condition.

Warning The standard power source for this equipment is designed to work in the range of 110-240V with 50-60Hz.

Warning This equipment is intended to be grounded using the *Grounded Connector*. Ensure that the host is connected to earth ground during normal use.

B.2 Warranty

The WR-ZEN TP is fully factory tested and warranted against manufacturing defects for a period of one year. Failure of the WR-ZEN TP due to installation problems caused by the the circumstances under the WR-ZEN TP is installed can not be warranted. This includes misuse, miswiring, overheating, operation under loads beyond the design range of the WR-ZEN TP.

For warranty or non-warranty replacement please follow the RMA Instructions.