

WR Node (WR PTP Core)

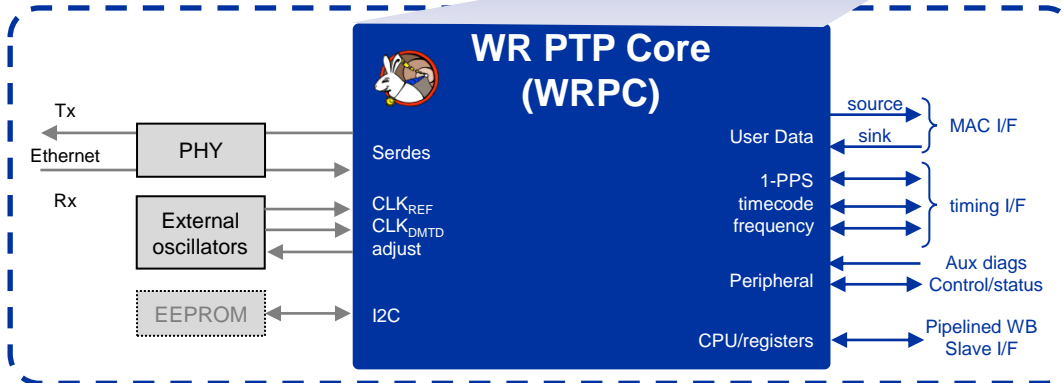
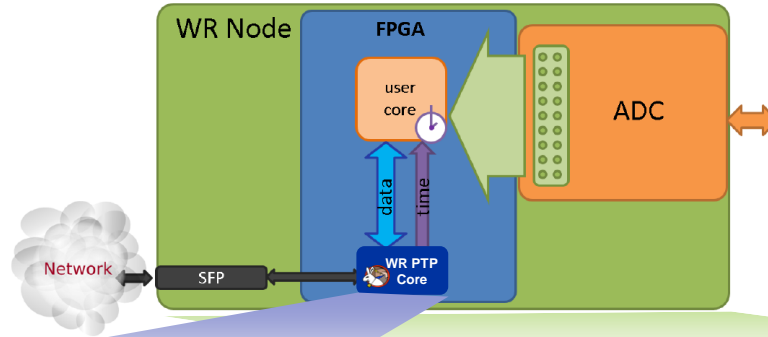


White Rabbit
COLLABORATION

Training material

Maciej Lipinski
WR Collaboration / CERN

WR Node = WR PTP Core + hardware



Hardware with WR support



[Simple PCIe FMC carrier \(SPEC\)](#)

[Simple VME FMC carrier \(SVEC\)](#)

[PCI eXtensions for Instr. \(PXI\) module](#)

[FPGA Mezzanine Card \(FMC\) module](#)

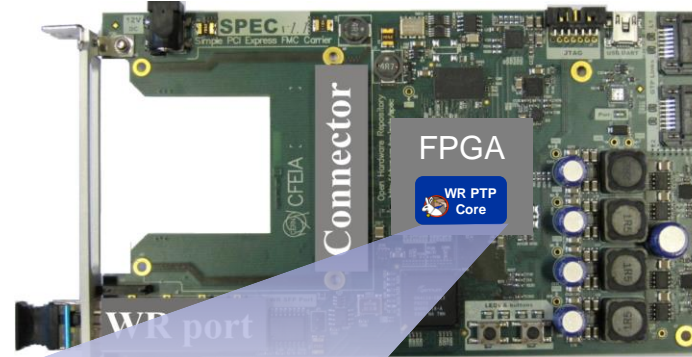
[CompactRIO White Rabbit](#)

[Mini-WR to WR-enable carrier](#)







GW: <https://ohwr.org/project/wr-cores>
SW: <https://ohwr.org/project/wrpc-sw>

Wiki: <https://ohwr.org/project/wr-cores/-/wikis/wrpc-core>
Release: <https://ohwr.org/project/wr-cores/-/wikis/Current-release>

WR Node = WR PTP Core + hardware



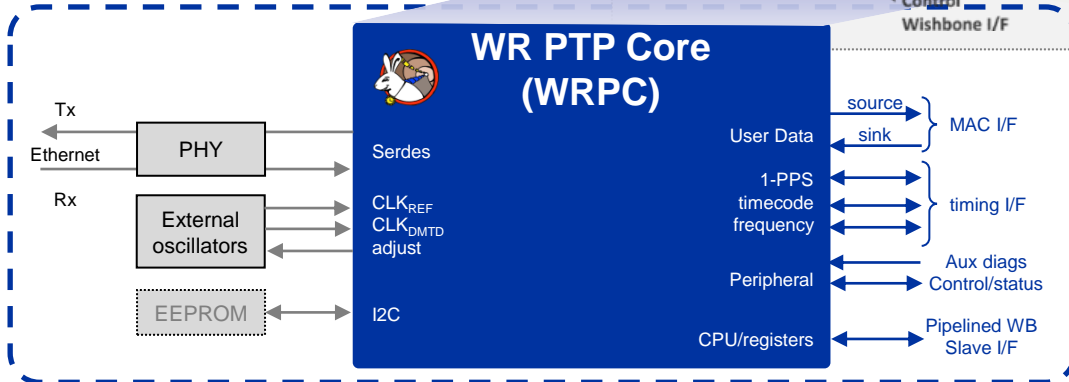
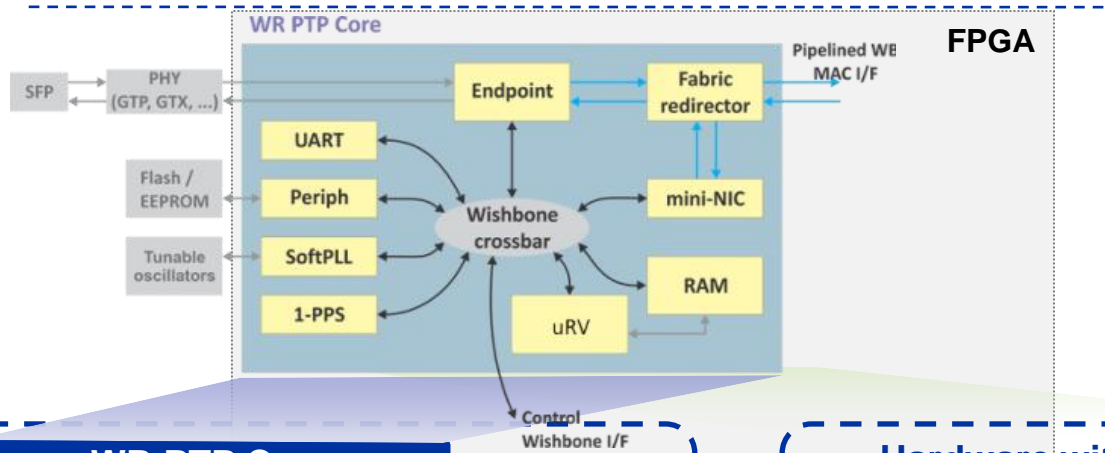
Hardware with WR support

-  [Simple PCIe FMC carrier \(SPEC\)](#)
-  [Simple VME FMC carrier \(SVEC\)](#)
-  [PCI eXtensions for Instr. \(PXI\) module](#)
-  [FPGA Mezzanine Card \(FMC\) module](#)
-  [CompactRIO White Rabbit](#)
-  [Mini-WR to WR-enable carrier](#)







GW: <https://ohwr.org/project/wr-cores>
SW: <https://ohwr.org/project/wrpc-sw>

Wiki: <https://ohwr.org/project/wr-cores/-/wikis/wrpc-core>
Release: <https://ohwr.org/project/wr-cores/-/wikis/Current-release>

WR Node = WR PTP Core + hardware



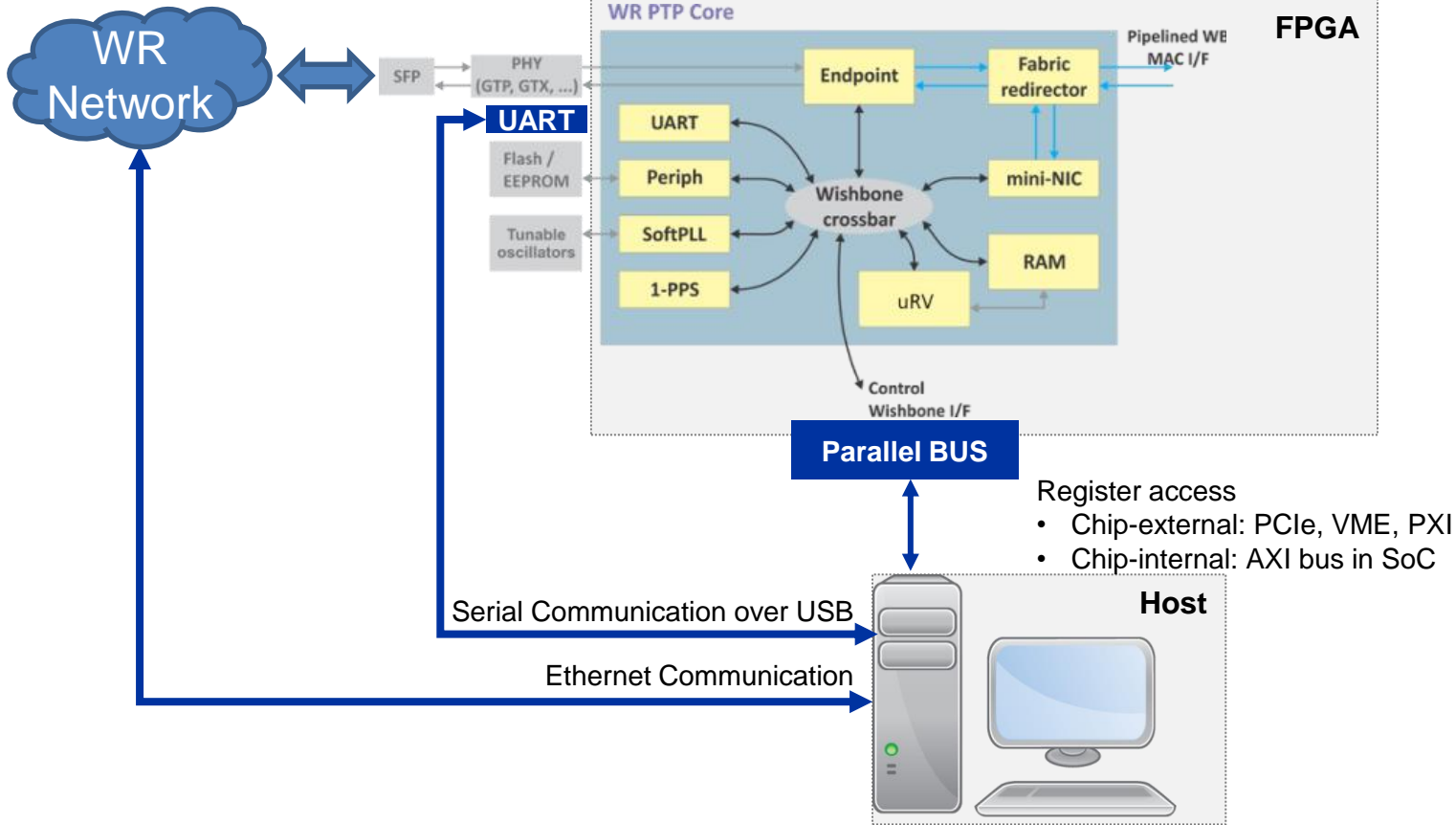
Hardware with WR support

-  [Simple PCIe FMC carrier \(SPEC\)](#)
-  [Simple VME FMC carrier \(SVEC\)](#)
-  [PCI eXtensions for Instr. \(PXI\) module](#)
-  [FPGA Mezzanine Card \(FMC\) module](#)
-  [CompactRIO White Rabbit](#)
-  [Mini-WR to WR-enable carrier](#)

GW: <https://ohwr.org/project/wr-cores>
SW: <https://ohwr.org/project/wrpc-sw>

Wiki: <https://ohwr.org/project/wr-cores/-/wikis/wrpc-core>
Release: <https://ohwr.org/project/wr-cores/-/wikis/Current-release>

Accessing WR PTP Core



Accessing WR PTP Core

- Register access over **parallel bus**
(access to all registers, using dedicated tools or direct memory mapping)

- Configuration and diagnostics
- Firmware loading
- Debugging (GDB, SoftPLL)
- Access to WRC shell via virtual UART

- Serial communication over **UART**

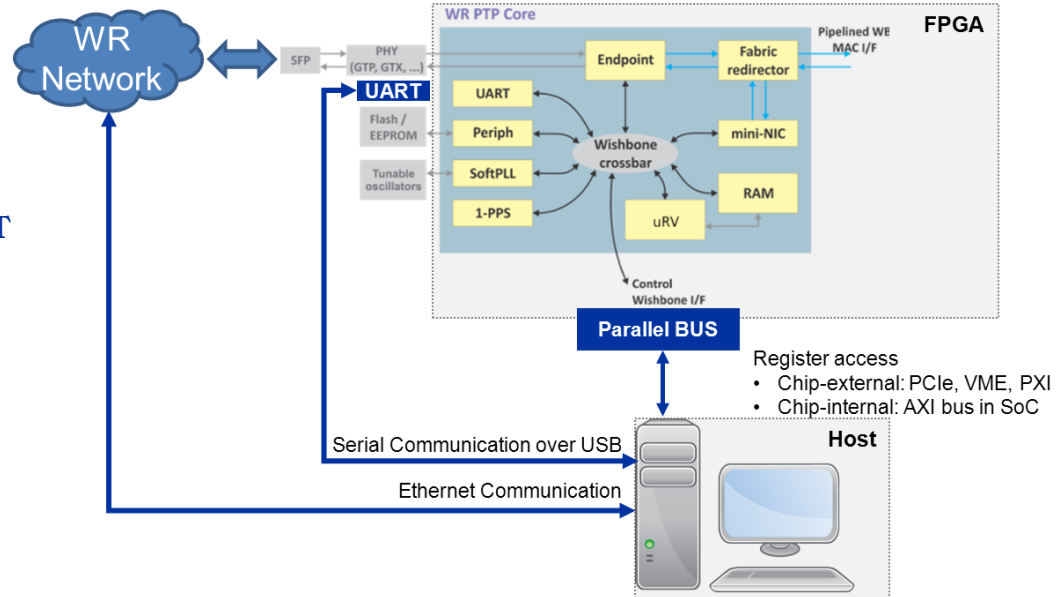
- Access to WRC shell via physical UART

- Ethernet communication over **WR Network**

- Access to WRC shell via netconsole
- SNMP get/set
- Syslog

- **Host**

- Machine with OS (ideally Linux), e.g., PC, PL of SoC
- Hosts tools used to access WRPC
- Tools available in **tools** folder of <https://ohwr.org/project/wrpc-sw>

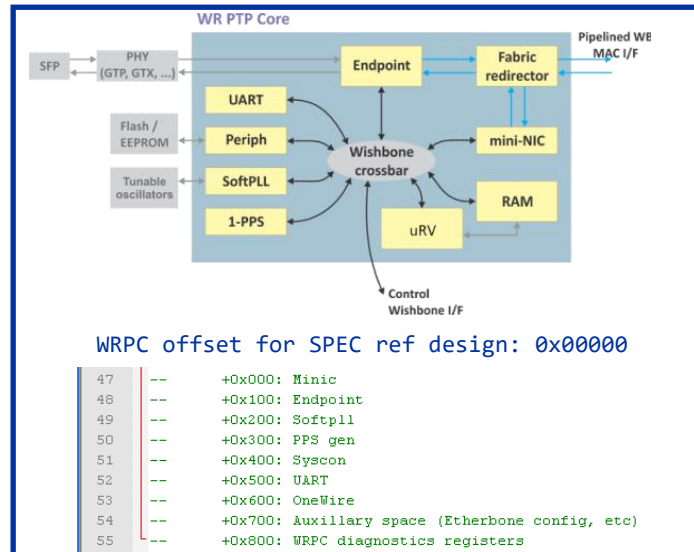


WRC shell: ways of accessing


- Register access over parallel bus: **vuart**
 - Tools available in tool **folder** of <https://ohwr.org/project/wrpc-sw>
 - **wrpc-vuart** -f <resource> -o <UART module offset> - [depricated]
 - **wrpc vuart -b <board> -f <resource> -o <WRPC offset>** - new “Swisstool”
 - Example execution for SPEC reference design over PCIe)
host# sudo ./wrpc-vuart -f /sys/bus/pci/devices/0000:01:00.0/resource0 -o 0x00500
host# sudo ./wrpc vuart -b pci -f /sys/bus/pci/devices/0000:01:00.0/resource0 -o 0x00000

- Serial communication over UART: use your favorite tool
 - Baudrate: 115200
 - Example tools: putty, picocom, minicom
 - Example command:
host# sudo picocom -b 115200 /dev/ttyUSB0

- Ethernet communication over WR Network: **nc**
 - netconsole must be enabled and configured on WRPC (see later slides)
 - Example command:
host# nc -u <IP of WRPC> 55



WRC shell: commands

- **WRC shell** can be accessed using
 - wrpc-vuart over parallel bus,
 - serial communication over UART
 - netconsole over WR Network
- **Commands available** in WRC shell
 - Basic commands: available by default in WRPC software binary available with release 
 - Advanced commands that need to be enabled before compiling new WRPC software binary - info later in this presentation

```
wrc# help
Available commands:
calibration
diag
gui
help
init
ip
mac
mode
pll
ps
ptp
ptrack
sdb
sfp
sleep
stat
temp
time
uptime
ver
verbose
vlan
w1
```


WRC shell: gui

wrc# gui

- The best tool to start with
- Equivalent of `wr_mon` available on WR Switch (see detailed description in “WR Switch Basics” presentation)
- Redesigned for WRPC v5
- Gives many important information about WRPC status and configuration:
 - Build version
 - Link status
 - PTP/WR configuration
 - MAC of the peer
 - PLL locking status
 - Timing mode
 - WR time
 - Servo status (if in slave mode)

```
SPEC WRPC Monitor wrpc-v5.0-pre1-1074-g176b78b8 | Esc/q = exit; r = redraw
TAI Time: 2024-11-28-22:02:18 UTC offset: 37 PLL mode: BC state: Locked
-----+-----+-----+-----+-----+-----+-----+
# | MAC | IP (source) | RX | TX | VLAN
-----+-----+-----+-----+-----+-----+
0 | 22:33:03:6c:56:70 | 192.168.5.10(BOOTP) | 356122 | 319300 | 0

--- HAL ---|----- PPSI -----
Itf | Frq | Config | MAC of peer port | PTP/EXT/PDETECT States | Pro
-----+-----+-----+-----+-----+-----+
wr0 | Lck | auto | 7c:bc:84:a0:02:a5 | SLAVE /IDLE /EXT_ON | R-W
Pro(protocol): R-RawEth, V-VLAN, U-UDP

----- Synchronization status -----
Servo state: White-Rabbit: TRACK_PHASE

--- Timing parameters ---
meanDelay : 25.003 ns err state: 21
delayMS : 25.000 ns err offset: 20
delayMM : 902.443 ns err delta: 3
delayAsymmetry : -0.003 ns
delayCoefficient : -0.0002678699999999987 fpa 18445508741375795472
ingressLatency : 148.326 ns
egressLatency : 180.750 ns
semistaticLatency: 6.400 ns
offsetFromMaster : 0.001 ns
Phase setpoint : 1.817 ns
Skew : 0.003 ns
Update counter : 57 times
Master PHY delays TX: 239.491 ns RX: 277.469 ns
Slave PHY delays TX: 180.750 ns RX: 154.726 ns
```

WRC shell: initial configuration

wrc# sdb fs 0

- SDB = Self-Describing Bus ([link](#))
- Creates SDB FS image in Flash memory
Base address and sector size taken from HDL
- SDBFS stores
 - MAC address
 - SFP database
 - Init script
 - t24p calibration

```
wrc# sdb fs 0
Formatting using location 0x0
filename: . ; first: 0; last: 37f
filename: mac-address ; first: 10000; last: 10005
filename: sfp-database ; first: 20000; last: 2007f
filename: wr-init ; first: 30000; last: 300ff
filename: calibration ; first: 40000; last: 4007f
Formatting SDBFS (base 0x00000000, size 0x00050000)...
Verification...
OK.
```

wrc# sfp add <PN> <egLat> <inLat> <relCoefH> <relCoefL>

- Add SFP parameters to database:
 - <PN> - SFP's part number
 - <egLat> - egress latency (deltaTx) [ps]
 - <inLat> - ingress latency (deltaRx) [ps]
 - <relCoefH> <relCoefL> - most & least significant 9 digits of relative delay coefficient (alpha)
- Database organized differently than on Switch

```
wrc# sfp add AXGE-1254-0531 180750 148326 1235332 333756144
Adding new SFP entry
1 SFPs in DB
wrc# sfp add AXGE-3454-0531 180750 148326 -1235332 333756144
Adding new SFP entry
2 SFPs in DB
wrc# sfp match
AXGE-3454-0531
SFP matched, dTx=180750 dRx=148326 alpha=-1235332333756144
```

wrc# sfp match

- Match inserted SFP with entry in database

wrc# sfp show

- Show SFP database

```
wrc# sfp show
1: PN:AXGE-1254-0531 dTx: 180750 dRx: 148326 alpha: 1235332333756144
2: PN:AXGE-3454-0531 dTx: 180750 dRx: 148326 alpha: -1235332333756144
```

WRC shell: initial configuration

wrc# mode <PTP Mode>

- Set PTP mode
 - gm – for Grandmaster
 - master for Free-running master
 - slave for Slave

```
wrc# mode gm
Locking PLL.....
wrc# mode master
Locking PLL...
wrc# mode slave
```

wrc# mac get

- Report MAC address of WRPC

wrc# mac <cmd> xx:xx:xx:xx:xx:xx

- Set MAC address to the value indicates with xx:xx:xx:xx:xx:xx in HEX
- cmd:
 - set – set for runtime
 - setp – set permanently

```
wrc# mac get
MAC-address: 22:33:03:6c:56:70
wrc# mac set xx:xx:xx:xx:xx:xx
```

wrc# ip

- Report IP address of WRPC which can be either retrieved from DHCP server using bootp, or set manually

```
wrc# ip
IP-address: 192.168.5.10 (bootp)
wrc# ip set 192.168.1.100
IP-address: 192.168.1.100 (static)
```

wrc# ip set

- Set manually IP v4 address of WRPC

WRC shell: init script

- WRPC allows to load user-defined initialization script on every startup
- This mechanism allows custom configuration
- Commands

wrc# init add <cmd>

- Adds command to the “user-defined” part
- Any configuration command can be added

wrc# init show

- Shows content of the script

wrc# init boot

- Restart WRPC

```
wrc# init add ip set 192.168.1.1
OK.
wrc# init show
-- built-in script --
vlan off
ptp stop
sfp match
mode slave
ptp start
-- user-defined script --
ip set 192.168.1.1
wrc# init boot
executing: vlan off
current vlan: 0 (0x0)
executing: ptp stop
executing: sfp match
AXGE-3454-0531
SFP matched, dTx=180750 dRx=148326 alpha=-1235332333756144
executing: mode slave
executing: ptp start
executing: ip set 192.168.1.1
IP-address: 192.168.1.1 (static)
```

WRC shell: monitoring

wrc# stat <on | off>

- Reports repeatedly (until off) loggable statistics
- Without parameter, toggles reporting

wrc# stat bts

- Prints bitslide value for established WR Link

wrc# ps

- [Manual](#) – section 5.4.5
- Prints:
 - Task list
 - Total time spent on task
 - Maximum time spent on task
- Possible to reset counters
- If any task is executed longer than 250-500ms might cause problems with synchronization. This tool can help detect this

```
Lnk:1 rx:22736 tx:16384 lock:1 ptp:uncalibrated sv:1 ss:'SYNC_SEC' sec:1633478659 nsec:458263232 mu:959160 dms:251919
Lnk:1 rx:22755 tx:16396 lock:1 ptp:uncalibrated sv:1 ss:'SYNC_SEC' sec:1633478663 nsec:538085464 mu:946843 dms:245759
Lnk:1 rx:22771 tx:16409 lock:1 ptp:slave sv:1 ss:'SYNC_NSEC' sec:1633478666 nsec:254039800 mu:957527 dms:251902
Lnk:1 rx:22785 tx:16419 lock:1 ptp:slave sv:1 ss:'SYNC_PHASE' sec:1633478669 nsec:457566360 mu:957530 dms:251904
Lnk:1 rx:22796 tx:16428 lock:1 ptp:slave sv:1 ss:'WAIT_OFFSET STABLE' sec:1633478672 nsec:025936904 mu:957538 dms:251908
Lnk:1 rx:22832 tx:16454 lock:1 ptp:slave sv:1 ss:'WAIT_OFFSET STABLE' sec:1633478680 nsec:133774424 mu:957540 dms:251909
Lnk:1 rx:22837 tx:16457 lock:1 ptp:slave sv:1 ss:'WAIT_OFFSET STABLE' sec:1633478680 nsec:912643576 mu:957541 dms:251909
Lnk:1 rx:22842 tx:16461 lock:1 ptp:slave sv:1 ss:'SYNC_PHASE' sec:1633478681 nsec:935006632 mu:957546 dms:251912
Lnk:1 rx:22846 tx:16463 lock:1 ptp:slave sv:1 ss:'WAIT_OFFSET STABLE' sec:1633478682 nsec:820976872 mu:958390 dms:252334
Lnk:1 rx:22851 tx:16466 lock:1 ptp:slave sv:1 ss:'WAIT_OFFSET STABLE' sec:1633478683 nsec:750801640 mu:957533 dms:251905
Lnk:1 rx:22854 tx:16469 lock:1 ptp:slave sv:1 ss:'TRACK_PHASE' sec:1633478684 nsec:742779880 mu:957530 dms:251904
Lnk:1 rx:22859 tx:16472 lock:1 ptp:slave sv:1 ss:'TRACK_PHASE' sec:1633478685 nsec:887992712 mu:957546 dms:251912

dtxm:224500 drxm:226090 dtxs:0 drxs:4800 asym:34 crtt:503770 cko:16 setp:7611 ucnt:0 bsld:3200 hd:33082 md:22558 ad:0 temp:48.8125 C
dtxm:224500 drxm:226090 dtxs:0 drxs:4800 asym:33 crtt:491453 cko:-69775784 setp:7611 ucnt:1 bsld:3200 hd:33086 md:22563 ad:0 temp:48.8125 C
dtxm:224500 drxm:226090 dtxs:0 drxs:3200 asym:34 crtt:503737 cko:-69779495 setp:7611 ucnt:2 bsld:3200 hd:33084 md:22568 ad:0 temp:48.8125 C
dtxm:224500 drxm:226090 dtxs:0 drxs:3200 asym:34 crtt:503740 cko:-3492 setp:4119 ucnt:3 bsld:3200 hd:33084 md:22564 ad:0 temp:48.8125 C
dtxm:224500 drxm:226090 dtxs:0 drxs:3200 asym:34 crtt:503748 cko:1089 setp:4119 ucnt:4 bsld:3200 hd:33085 md:22606 ad:0 temp:48.8125 C
dtxm:224500 drxm:226090 dtxs:0 drxs:3200 asym:34 crtt:503750 cko:1089 setp:4119 ucnt:12 bsld:3200 hd:33085 md:22622 ad:0 temp:48.8125 C
dtxm:224500 drxm:226090 dtxs:0 drxs:3200 asym:34 crtt:503751 cko:1089 setp:4119 ucnt:13 bsld:3200 hd:33084 md:22575 ad:0 temp:48.8125 C
dtxm:224500 drxm:226090 dtxs:0 drxs:3200 asym:34 crtt:503756 cko:1899 setp:6008 ucnt:14 bsld:3200 hd:33081 md:22599 ad:0 temp:48.8125 C
dtxm:224500 drxm:226090 dtxs:0 drxs:3200 asym:34 crtt:504600 cko:438 setp:6008 ucnt:15 bsld:3200 hd:33084 md:22624 ad:0 temp:48.8125 C
dtxm:224500 drxm:226090 dtxs:0 drxs:3200 asym:34 crtt:503743 cko:-2 setp:6008 ucnt:16 bsld:3200 hd:33083 md:22554 ad:0 temp:48.8125 C
dtxm:224500 drxm:226090 dtxs:0 drxs:3200 asym:34 crtt:503740 cko:3 setp:6008 ucnt:17 bsld:3200 hd:33083 md:22607 ad:0 temp:48.7500 C
dtxm:224500 drxm:226090 dtxs:0 drxs:3200 asym:34 crtt:503756 cko:-1 setp:6008 ucnt:18 bsld:3200 hd:33084 md:22565 ad:0 temp:48.7500 C
```

wrc# ps

iterations	seconds.micros	max_ms	name
250796	76.848777	1	idle
0	0.000000	0	check-link
79	0.000853	0	uptime
731	0.393709	2	ptp
39	0.027316	1	ptp bmc
13	0.130972	128	shell+gui
0	0.000000	0	spil-bh
12	0.123003	13	temperature
360	0.055148	1	net-bh
0	0.000000	0	dac-logger
0	0.000000	0	arp
79	0.070932	2	ipv4
0	0.000000	0	latency-probe
16	0.013606	1	lldp
0	0.000000	0	snmp
0	0.000000	0	stats
79	0.008827	1	diags
0	0.000000	0	netconsole
79	0.754349	10	sfp dom
0	0.000000	0	netif

WRC shell: debugging

wrc# verbose bbbbbb

- Sets level of verbosity of PTP daemon (PPSi)
- Verbosity levels

verbose 00000000

```
|||||||
|||||||\config
|||||||\ext
|||||\bmc
|||\servo
||\frames
|\time
\fsm
```

```
wrc# verbose 0010000
PPSi verbosity: 00100000
wrc# diag-frames-1-wr0: RECV 46 bytes at 1732836018.600560467.213 (type 0, sync)
diag-frames-1-wr0: RECV 46 bytes at 1732836018.601159443.213 (type 8, follow_up)
diag-frames-1-wr0: SENT 44 bytes at 1732836018.884843800.000 (delay_req)
diag-frames-1-wr0: RECV 54 bytes at 1732836018.885714291.212 (type 9, delay_resp)
diag-frames-1-wr0: RECV 78 bytes at 1732836019.477038675.215 (type b, announce)
diag-frames-1-wr0: SENT 44 bytes at 1732836019.518811224.000 (delay_req)
diag-frames-1-wr0: RECV 54 bytes at 1732836019.520059331.215 (type 9, delay_resp)
diag-frames-1-wr0: RECV 46 bytes at 1732836019.654518835.212 (type 0, sync)
diag-frames-1-wr0: RECV 46 bytes at 1732836019.655151859.212 (type 8, follow_up)
vdiag-frames-1-wr0: SENT 44 bytes at 1732836019.844816888.000 (delay_req)
diag-frames-1-wr0: RECV 54 bytes at 1732836019.845683459.214 (type 9, delay_resp)
erbodiag-frames-1-wr0: SENT 44 bytes at 1732836020.409851960.000 (delay_req)
diag-frames-1-wr0: RECV 54 bytes at 1732836020.410732307.212 (type 9, delay_resp)
sediag-frames-1-wr0: RECV 46 bytes at 1732836020.486281171.209 (type 0, sync)
diag-frames-1-wr0: RECV 46 bytes at 1732836020.486962035.209 (type 8, follow_up)
00diag-frames-1-wr0: SENT 44 bytes at 1732836021.145815960.000 (delay_req)
diag-frames-1-wr0: RECV 54 bytes at 1732836021.146530867.213 (type 9, delay_resp)
0diag-frames-1-wr0: SENT 44 bytes at 1732836021.293810776.000 (delay_req)
diag-frames-1-wr0: RECV 54 bytes at 1732836021.294675843.214 (type 9, delay_resp)
0
PPSi verbosity: 00000000
wrc#
```

WRC shell: advanced commands

wrc# sfp info

- Limited equivalent of wrs_sfp_dump
- Prints basic information about SFP
- Compilation options:
 - CONFIG_CMD_SFP_INFO=y

wrc# netconsole

- [Manual](#) – section 4.4
- Enable netconsole, ready to accept connections from remote host
- Compilation options
 - CONFIG_CMD_NETCONSOLE=y
 - NETCONSOLE_DEF_WAIT=y

host#nc -u <IP> 55

```
wrc# sfp info
Nominal Bit Rate: 1300 Mbits/s
Vendor Name: FS
Vendor PN: GE-LC-1310
Vendor serial: C2003129746
TX Wavelength: 1310
Temperature: 36.97 C
Voltage: 3.3488 V
Bias Current: 22.88 mA
TX power: 0.2265 mW
RX power: 0.2856 mW
```

```
wrc# netconsole
netconsole is waiting for peer
```

```
$ nc -u 192.168.1.20 55
```

```
wrc#temp
temp
pcb:46.3750
wrc#mode
mode
running; e2e slave
```

WRC compilation configuration

- WRC software: <https://ohwr.org/project/wrpc-sw>
- Needs to be compiled with suitable configuration
 - Predefined for existing boards in **configs**
 - Using predefined configuration:
host# make <configuration file name>
 - Access configuration menu:
host# make menuconfig
 - Configuration saved to file: **.config**
- Important parameters which must match HW/GW
 - Target Platform
 - Generic WR Node 8-bit PCS/PHY
 - Generic WR Node 16-bit PCS/PHY
 - WR Switch
 - AFCZ
 - eRTM14/15
 - Size of RAM in FPGA (in Advanced config)
 - Should match RAM in FPGA
- Many advanced commands can be added, e.g.,
 - netconsole
 - sfp info
 - freqmon

Predefined SW configs:

```
wrpc-sw$ ls -l configs/
afcz_defconfig
afczv2_defconfig
all_defconfig
babywr_defconfig
devel_build_test_defconfig
ertm14_defconfig
gsi_defconfig
gsi_delay_defconfig
pxie_fmc_defconfig
spec_defconfig
spec_delay_defconfig
spec_riscv_comp_defconfig
spec_silabs_defconfig
target_generic_phy_16bit_defconfig
target_sis8300ku_defconfig
wr2rf_defconfig
wren_defconfig
wrnic_defconfig
wrpc_sim_defconfig
wr_switch_defconfig
```

```
WR PTP Core software configuration
Arrow keys navigate the menu. <Enter> selects submenus --> (or empty submenu ----).
Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes, <h> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]
excluded <M> module < > module capable

[*] architecture (riscv) -->
[ ] Use compressed instructions for RISCv
(A) Data row size to support cpu.raset
Target Platform (Generic WR Node with 8-bit PCS/PHY) -->
Build as a host process to develop/debug network (NEW)
Build size for the temporary output string of 'pp' priority (NEW)
LPDC (Low-phase drift calibration) transceiver (No lpdc transceiver) ---->
*** features ***
[*] Include minimal IP support in wrpc-sw
[ ] Include syslog client support
[ ] Ethernet support
[*] Filter and rx/tx frames in a VLAN (as opposed to untagged)
(1) Use this VLAN number for filter/tx/rx
(10) Route this VLAN to fabric class 7 (Ethernet)
(11) Route this VLAN too to fabric class 7 (Ethernet)
(20) Route this VLAN too to fabric class 6 (Streamer/NIC)
[*] Mini SNMP responder
[*] Add SET support to the Mini SNMP responder
[*] Add support of changing init script/commands via SNMP
[*] Add support of generation of SDB via SNMP
[ ] Allow to run any shell command via SNMP
[*] Add support of auxiliary diagnostics registers to SNMP
[*] Add support for auxiliary diagnostics registers
[*] Write of WRCP diagnostics to dedicated WB registers
[*] Support absolute calibration
[ ] Include LLDp protocol transmit-only
[ ] Include netconsole via UDP
[*] Add support for temperature sensors
[ ] Include netconsole via UDP
[*] Add support for temperature sensors
[ ] Add support for generic sensors
[ ] Add support for W1 bus
[*] Add support for W1 temperature sensor
[ ] Add support for W1 eeprom
[ ] Add support for fractional PLL (experimental)
*** commands ***
[ ] Add 'config' command to display this configuration
[ ] Add 'init' command in the binary (build-time)
(vlan off;ptp stop;sfp match;node slave;ptp start) Enter the init command, use ';'
[ ] Add 'calibration show' command to print all calibration values
[ ] Add 'refresh' command
[ ] Add 'pps' command
[ ] Add 'sfp info' command to print info about SFP
Extend 'sfp info' command with SFP monitoring info
[ ] Add 'leapsec' command to control leapsecond value
[ ] Extend 'ptp' command with extra subcommands
[*] Extend 'gul' and 'stat' commands with stats for servo errors
[ ] Add 'freqmon' command for built-on clock frequency monitor
[ ] Add 'ep' command to diagnose endpoints
*** wrpc-sw is tainted if you change the following options ***
[*] Advanced configurations, only for developers
[ ] Control of trace/debug messages verbosity
[ ] Build as a host process to develop/debug network (NEW)
(151072) Size of the RAM in the FPGA for this program (NEW)
(2048) Size of the stack area needed by this program (NEW)
(238) Size for the temporary output string of 'pp' priority (NEW)
(15) Poll interval in seconds, for temperature sensors (NEW)
```


WRC host tools

- Host tools are in **tools** folder of <https://ohwr.org/project/wrpc-sw>

- Old tools (to be deprecated, still available):

host# wrpc-vuart -f <resource> -o <UART module offset> - access virtual UART

host# wrpc-diags -f <resource> -o <UART module offset> - access diagnostics registers

host# wrpc-load -f <resource> -o <UART module offset> - load software binary

- New “swiss tool”:

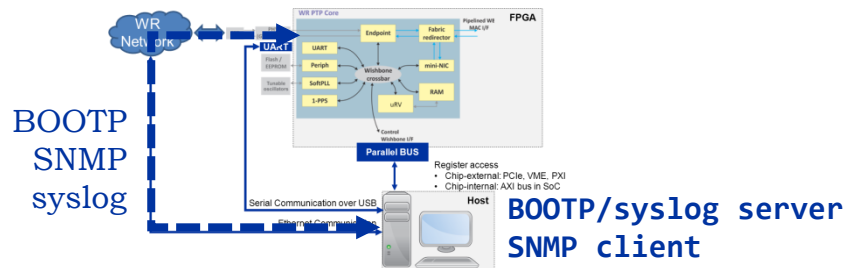
host# wrpc <cmd> -b <board> <board-specific access to resource> <cmd-specific>

- Replaces existing tools above
- Provides new functionalities:
 - Reading board and wrpc info
 - Recording data from SoftPLL
 - Providing gdb server for Risk-V
 - Displaying diagnostics from auxiliary input (user-specific)
- More info on presentation by Tristan

```
m lipinsk@cfc-774-cb1m:wrpc help
usage: wrpc [command] [-b BOARD] [OPTIONS...]
command is one of:
  help          - display list of commands (this help), or help for a command
  version       - display tool version
  board         - display list of supported boards, or help for a board
  load          - load wrpc firmware and restart
  vuart        - virtual uart, connect to wrpc cli
  info          - display wrpc info and check board
  spll-recorder - SoftPLL log recorder
  gdbserver     - risc-v gdb-server
  wdiags        - WR diags dumper
  aux-logger    - display wdiag AUX0 value for logging
```

Standard services (1)

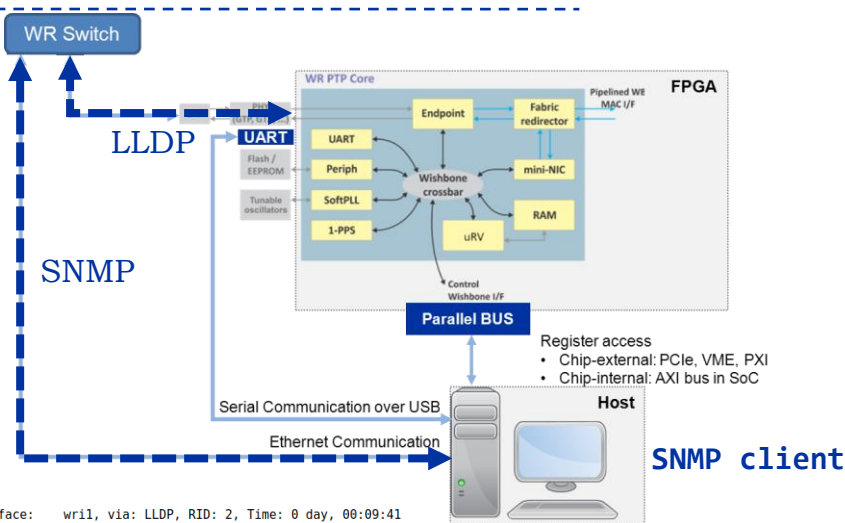
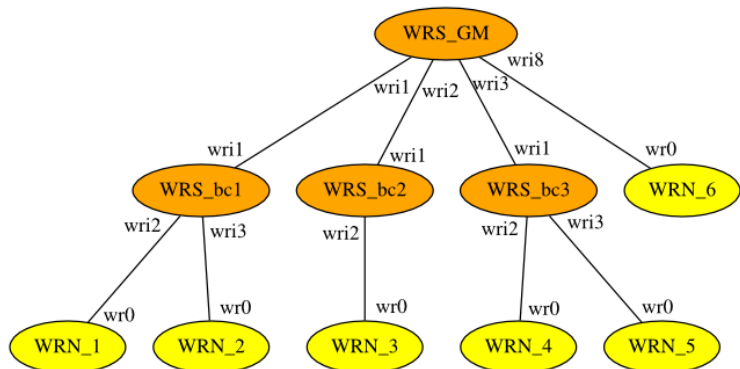
- BOOTP ([Manual](#) – section 6)
 - Allows to obtain IPv4 address from BOOTP server
- SNMP ([Manual](#) – section 5.1)
 - Single Network Management Protocol
 - Diagnostics and configuration over WR Network
 - MIB: [WR-WRPC-MIB.txt](#)
 - Allows:
 - Reading the monitoring values
 - Managing SFP database
 - Setting up init script
 - Generating SDBFS
 - Configuring netconsole
 - Running shell commands
 - Useful commands:
host# snmpwalk
host# snmpget
host# snmpset
- Syslog ([Manual](#) – section 5.2)
 - Send to server: boot time, links/sync status, temp
 - Command to enable/configure
wrp# syslog <IP> <MAC>



```
$ SNMP_OPT="-c public -v 2c -m WR-WRPC-MIB -M +/var/lib/mibs/ietf:lib 192.168.1.20"
$ snmpwalk $SNMP_OPT wrpcCore
WR-WRPC-MIB::wrpcVersionHwType.0 = STRING: spec
WR-WRPC-MIB::wrpcVersionSwVersion.0 = STRING: wrpc-v3.0-251-g14e952e
WR-WRPC-MIB::wrpcVersionSwBuildBy.0 = STRING: Adam Wujek
WR-WRPC-MIB::wrpcVersionSwBuildDate.0 = STRING: Jun 7 2016 18:12:24
WR-WRPC-MIB::wrpcTimeTAI.0 = Counter64: 1465375022
WR-WRPC-MIB::wrpcTimeTAIString.0 = STRING: 2016-06-08-08:37:02
WR-WRPC-MIB::wrpcTimeSystemUptime.0 = Timeticks: (18186) 0:03:01.86
WR-WRPC-MIB::wrpcTemperatureName.1 = STRING: pcb
WR-WRPC-MIB::wrpcTemperatureValue.1 = STRING: 38.5625
WR-WRPC-MIB::wrpcSp1lMode.0 = INTEGER: slave(3)
WR-WRPC-MIB::wrpcSp1lIrqCnt.0 = Counter32: 1259605
[...]
WR-WRPC-MIB::wrpcPortSfpInDB.0 = INTEGER: inDataBase(2)
WR-WRPC-MIB::wrpcPortInternalTx.0 = Counter32: 452
WR-WRPC-MIB::wrpcPortInternalRx.0 = Counter32: 869
WR-WRPC-MIB::wrpcSfpPn.1 = STRING: AXGE-1254-0531
WR-WRPC-MIB::wrpcSfpDeltaTx.1 = INTEGER: 180750
WR-WRPC-MIB::wrpcSfpDeltaRx.1 = INTEGER: 148326
WR-WRPC-MIB::wrpcSfpAlpha.1 = INTEGER: 72169888
End of MIB
```

Standard services (2)

- LLDP
 - Link Layer Discovery Protocol
 - Allows to retrieve information about device on the other side of a link
 - WRPC: only reports its presence to a peer
 - WR Switch uses LLDP to discover devices connected to it. WR Switch reports this information over SNMP to SNMP server
 - Used to draw network tree:



```

Interface: wr11, via: LLDP, RID: 2, Time: 0 day, 00:09:41
ChassisID: mac [redacted]:02:61
SysName: wrch2
SysDescr: WR-SWITCH: SCB HW:3.5, FPGA:LX240T; version: WP3a-wrpc_fixes-20-g46be1a82 (Adam Wujek); compiled at Sep 1 2021 03:05:45
MgmtIP: 192.168.1.112
MgmtIface: 2
Capability: Bridge, off
Capability: Router, off
Capability: Wlan, off
Capability: Station, on
Port:
  PortID: mac [redacted]:61:03
  PortDescr: wr13
  TTL: 20
-----
Interface: wr12, via: LLDP, RID: 4, Time: 0 day, 00:00:39
ChassisID: mac [redacted]:66:77
SysName: 0.0.0.0
SysDescr: SPEC: WP3a-wrpc_fixes-42-gb668299e-dir
Port:
  PortID: mac [redacted]:66:77
  PortDescr: wr0
  TTL: 40
    
```

Summary

- Method of accessing WR PTP Core
- WRPC Shell
 - Ways of accessing
 - Ways of enabling command
 - Commands
- Compilation configuration
- Host tools
- Standard services

Thank you