## Contents

1. What’s the Local Management Tool? ..... 2
2. Current Status ..... 2
3. LMT Development ..... 2
4. LMT Features ..... 3
   4.1 Switch Back-end ..... 5
5. How to use it ..... 5
   5.1 Main Menu ..... 6
   5.2 Advanced Menu ..... 7
6. Screenshots ..... 7
1 What’s the Local Management Tool?

The Local Management Tool (LMT) is a web interface that allows users to manage the White-Rabbit (WR) switch from a web browser. It displays the main configuration and status of the services and components that are available for the switch, such as endpoints’ mode and calibration status, SFP calibration, PTP, etc. It acts as an abstraction layer between the back-end scripts and programs in /wr/bin/ folder, making the WR switch management easier for the user.

![Local Management Tool - Main Dashboard Screen](image)

Figure 1: Local Management Tool - Main Dashboard Screen

2 Current Status

The preliminary version of LMT is already finished and ready to use for the v3.3 software version of the WR switch. It belongs to the wr-switch-sw project but it is not available on master branch yet. At the moment of writing this document the LMT is being merged from ugr-181104 UGR branch to rubi-140204 where it is getting ready for the next White-Rabbit software release.

3 LMT Development

LMT has been developed as a HTML web server using PHP as the main scripting language for the front-end implementation.

At the very beginning three web server for embedded systems were tested for the development of this tool: Boa, thttpd and Lighttpd. All of them are Open Source and are available on the WR switch buildroot.

After testing the stability, performance and resource consumption of them, we decided to integrate Lighttpd+PHP webservice as the base for the LMT because of its results in terms of stability, characterization possibilities and resources administration.

As previously said, the main front-end’s programming language is PHP. On the other hand, this development includes the utilization of several configuration scripts and C programs playing the role of back-end that perform the necessary reading and writing operations in the gateware registers to make possible the switch configuration.
4 LMT Features

The main purpose of the LMT development is to configure and monitor different switch elements, such as endpoint configured mode, calibration parameters for SFPs and endpoints, PTP configuration, and the possibility of rebooting the switch. These main features can be grouped by as it follows:

- **System information**: the web interface shows several information regarding the state of different switch services and components: operating system, IP configuration, firmware, endpoint calibration, PTP status, max. transfer filesize and the WR timing.

![](image1.png)

Figure 2: Local Management Tool - Endpoint Display Info (status, calibration and lock info)

- **Administration**: Administration of PHP file transfer size, configuration of WR switch as GrandMaster and Master, and the possibility of rebooting the switch.

![](image2.png)

Figure 3: Local Management Tool - Switch Management Menu

- **Tools**: The front-end consists in a web interface that interacts with WR switch scripts and C programs. These scripts and programs carry out the endpoint configuration tasks, loading binary files into both the LM32 processor and the FPGA, PTP configuration, and terminal access.
Figure 4: Local Management Tool - Loading ln32 and FPGA binary files

Figure 5: Local Management Tool - Endpoint configuration (Master/Slave)
The following list displays all the actions that can be performed by this tool:

- Modify switch operation mode (Master/GrandMaster).
- Check whether an endpoint is wr_master/wr_slave, locked/unlocked and calibrated or un-calibrated.
- Modify endpoint wr_master/wr_slave mode.
- Display IP switch configuration.
- Software and hardware versions.
- White-Rabbit timing.
- Modify maximum filesize of upload files to the switch.
- PTP daemon configuration: interface, unicast, clock accuracy, etc.
- Endpoint configuration: registers configuration.
- Terminal simulation avoiding SSH connections.
- Login system.
- Load lm32 and FPGA binaries into the switch.
- Switch reboot.

### 4.1 Switch Back-end

Nowadays, the back-end is composed by a group of scripts and programs that are implemented in the switch. The user will only interact with them from the web interface, which will execute the necessary operations and functions that perform the different features that the switch provides transparently for the user. This saves non-expert users to work at low-level since WR switch requires form a certain knowledge to handle configuration files and scripts.

The scripts and programs that are used from the web interface are the following:

- `/wr/bin/load-lm32`
- `/wr/bin/load-virtex`
- `/wr/bin/ptpd`
- `/wr/bin/shw__ver`
- `/wr/bin/wr_date`
- `/wr/bin/wr_management`
- `/wr/bin/wrsw__hal`
- `/wr/bin/wrfs_mnt.sh`
- `/etc/init.d/S80lighttpd.sh`

These programs are run from the web interface by using the PHP function `shell_exec()` and adding the necessary parameters correctly. After execution them, the front-end gets the output which is later formatted and shown to the user.

### 5 How to use it

As every web server, the WR switch must be configured with a single IP. The method that you use for this is up to you, but the easiest one is to connect the switch Ethernet configuration port to a router that offers a DHCP service.
The LMT runs from the start-up thanks to the S80lighttpd.sh script in `/etc/init.d` so that you only need to open a web browser (tested on Google Chrome and Mozilla Firefox) and access the tool on `http://ip_of_your_switch`.

![Local Management Tool - Before logged in](image1)

The first time you access the LMT you will probably not be logged in, so you are only able to see the dashboard information and the endpoint display. In order to perform any configuration you would have to log in. At this moment there is just one user `root` which password is `blank` (no password). This will be changed in the future.

![Local Management Tool - Main Dashboard Screen](image2)

### 5.1 Main Menu

- **Dashboard**: Displays the main switch info: IP configuration, hostname, firmware version, FPGA, WR time, PTP status and PHP maximum filesize.
- **PTP Configuration**: Configures and relaunches the PTP daemon.
• **Endpoint Mode**: Changes endpoint `wr_slave/wr_master` mode.
• **LM32 & FPGA**: Loads lm32 and FPGA files into the switch `/wr/lib/firmware` folder.
• **Switch Management**: Changes switch mode (Master/GrandMaster) and PHP maximum filesize. It is also possible to reboot the switch.

### 5.2 Advanced Menu

In order to enter the advanced menu please click on the **Advanced Mode** on the left Menu bar. This will display the SFP calibration, endpoint tool, endpoint calibration and virtual console pages.

• **SFP Calibration**: Adds SFP’s configuration to the database in `/etc/sfp__database.conf`.
• **Endpoint Tool**: Tools for endpoints: enable/disable calibration transmission (not yet available in back-end), endpoint registers configuration, lock, etc.
• **Endpoint Calibration**: Adds and modifies calibration values for all endpoints. It modifies `/etc/wrsw_hal.conf` file.
• **Virtual Console**: implements a virtual terminal from the web interface.

### 6 Screenshots

![White-Rabbit Switch Manager](image)

Figure 8: Local Management Tool - Header: Endpoint info from `wr_mon`
Figure 9: Local Management Tool - Main Dashboard Screen

Figure 10: Local Management Tool - PTP Configuration
Figure 11: Local Management Tool - Endpoint Mode Configuration

Figure 12: Local Management Tool - Loading Binary Files
Figure 13: Local Management Tool - Switch Management

Figure 14: Local Management Tool - Dashboard & Advance Menu (left side)
Figure 15: Local Management Tool - Adding SFP Database

Figure 16: Local Management Tool - Endpoint Tool (wr_phytool)
Figure 17: Local Management Tool - Endpoint Tool (wr_phytool)

Figure 18: Local Management Tool - Endpoint Tx/Rx Calibration
Figure 19: Local Management Tool - Terminal Emulator