1 Setup

1. you need to setup a DHCP+TFTP server on your PC.
   In some distribution, you can use the package `dnsmasq`\(^1\) to ease the installation

2. Then you need to setup a NFS server in `/tftpboot`

3. You need to compile binaries from the lastest branch of `wrs-switch-sw`
   Below a small exemple on how to look it but you should look at the official documentation\(^2\).

\(^{1}\)A small tutorial written for Ubuntu 12.04, can be found here
\(^{2}\)Read the wrs-build.pdf inside the `wrs-switch-sw/doc/` folder to build the switch binaries.
wr-switch-sw/build/wrs_build-all  
wr-switch-sw/build/wrs_build-all --pack

4. You need to put in the TFTP’s root directory (/tftpboot) the firmware you want to flash at the end of the test by copying

```bash
cp ${WRS_OUTPUT_DIR}/images/at91bootstrap.bin /tftpboot
cp ${WRS_OUTPUT_DIR}/images/barebox.bin /tftpboot
cp ${WRS_OUTPUT_DIR}/images/zImage /tftpboot
cp ${WRS_OUTPUT_DIR}/images/wrs-image.jffs2.img /tftpboot
```

5. Then you need to extract the filesystem on your /tftpboot directory.

```bash
mkdir -p /tftpboot/rootfs-test
sudo tar -xzf ${WRS_OUTPUT_DIR}/images/wrs-image.tar.gz -C /tftpboot/rootfs-test
```

6. And finally use the install scripts to send the files to tftpboot

```bash
sudo wr-switch-testing/sw/install.sh /tftpboot/rootfs-test/
```

## 2 Flashing

Once the compilation is done, you need to check the DDR and install the bootloaders into the dataflash for each board to test. Do do this you need to connect the left-USB port to your machine and execute:

```bash
wr-switch-sw/build/flash-wrs -c --build
```

## 3 Usage

Once the board is flashed, boot your board by pressing reset button. Connect to USB serial (Test)

```bash
minicom -D /dev/ttyUSB0 -b 115200
```
and select

**boot from nfs (test)**

---

3 You can use the `--clean` flag to make sure to rebuild everything.
4 Read the wrs-build.pdf inside the wrs-switch-sw/doc/ folder to understand better how to flash the board.
during barebox menu. Then if everything go well (DHCP, TFTP & NFS) you should obtain the following message:

Starting up Test...

We do not recommend to run alpha-pts from serial port, because it is shared with FPGA UART.
To call the test you should try:

* USB Gadget: minicom -D /dev/ttyACM0 -b 115200
* SSH: ssh root@192.168.7.51

And then execute the testing script

/alpha-pts/testing.sh

This message show the basic steps to follow that are:

1. Open a USB Gadget connexion (left USB port)
2. Run /alpha-pts/testing.sh
3. Enter the serial of the board.
4. Wait until all the test are completed
5. Look at the log in your /tftpboot/rootfs-test/alpha-pts/logs/

Below, a log of a test with failing NAND (test 007) on sub test 0 & sub test 1:

Thu Jan  1 00:00:41 UTC 1970

001 > OK (00’06)
002 > OK (00’00)
005 > OK (00’53)
006 > OK (06’34)
007 > ERROR ( Sub0:13% Sub1:75% Sub2:OK Sub3:OK) (08’21)
008 > OK (06’29)
009 > OK (00’30)
010 > OK (09’16)

You can also find a full log for each test and one that keep an history of all test execution with the following nomenclatures

output-<SN>_<TESTID>.log
output-<SN>_history.log
3.1 Single test

Finally, by executing /alpha-pts/testing.sh --help you can obtain an help message where you can find out that test can be run separately.

Usage: /alpha-pts/testing.sh [OPTION]... [TESTNUM]

Testing script to check various components of the White Rabbit Switch

if TESTNUM is not defined all tests will be executed in ascending order

Options:
- h --help                         Show this little message
- f --force                        Do not ask to continue
- u --update                       Update binary files
- i --inlog                        Create an independant log file for each test
- l --list                         List the different building steps
- s --serial                       Serial number

For example by calling

$ /alpha-pts/testing.sh -s 310010 -f 07

you will run test 007 (NAND) on board with S/N=310010 without prompting user if he wants to run it or skip it.

3.2 Description of the Tests

You can find a better description of the test by reading the header of each test in the source code.

1. 001-MD5_Clrk.sh: Check MD5 of needed files
2. 002-LED_Clrk.sh: Check the LED of the switch (Not implemented)
3. 003-FAN_Clrk.sh: Check the FANs (Not implemented)
4. 004-USB.sh: Check the USB (Not implemented)
5. 005-FPGA_EBI1Bridge.sh: Check the the EB1 bridge (FPGA<->CPU)
6. 006-FPGA_QDDRStress.sh: Stress the QDDR test.
7. 006-FPGA_TempRetrieving.sh: Retrieve the temperature (It should be at the end of the test, when the QDDRStress can auto shutdown when temperature reach a maximum)
8. 007-NF_MT29F4G16.sh: Test the NAND flash (This test has a bug, see [Troubleshoots])

9. 008-DF_AT45DB642.sh: Check the dataflash.

10. 009-TFTP_Flashing.sh: Reflash the DF and NF using firmware in TFTP directory.

4 Troubleshoots

4.1 NAND

It seems that the NAND test does not performs well on every board when it reach the address 0x08040000

> Checking Block: 1026 @ 0x08040000 ..........ERROR (26 %)

There is also an error when we try to erase a bad block, it is why we only use `flash_eraseall -j` to erase NAND flash at the beginning of the test.

4.2 Dataflash

In case the dataflash is failling you can load the bootloaders directly to the ddr by using the following command:

`wr-switch-sw/build/flash-wrs --test`

This document is written in markdown syntax, and can be compiled by executing `pandoc --toc --number-section README.mkd -o README.pdf`