



SERENITY

SERENITY Board Setup Guide

Release v1.1

SERENITY Collaboration

Rev2.0: May 14th, 2019

TABLE OF CONTENTS:

1	First Steps	2
1.1	Accessing the Daughter Card Sites:	3
1.2	Installation (SMASH & EMP)	3
1.3	Power up!	4
1.4	Uploading firmware to the daughter cards	5
1.5	Checking communication with the loaded firmware	7
1.6	Next Steps:	8
2	Artix TTC	9
2.1	Configuring the TTC firmware	9
2.2	Checking the firmware version	10
3	Cheatsheet for test stands (serenitybutler etc.)	11
3.1	Version checks	11
3.2	Power control	11
3.3	Reference clock and firefly configuration	12
3.4	Programming the daughter card FPGAs	12
3.5	Crosspoint switch	12
3.6	XVC server	13
3.7	Artix TTC	13
3.8	Test suite	13
3.9	Running docker containers	14
3.10	Changing the CPU frequency governor	14
4	Troubleshooting and FAQ	16
4.1	Advanced Troubleshooting	16
4.2	Uncommon Issues	17
5	Advanced procedures	18
5.1	Memory and Artix service firmware	18
5.2	On-board computer: Adding CERN accounts	19
6	Current software and firmware versions	22
6.1	Recommended versions	22
6.2	Checking versions: Recommended method	22
6.3	Checking versions: Alternate method	23
7	Status of this Document	25

SERENITY

Scope of this Document

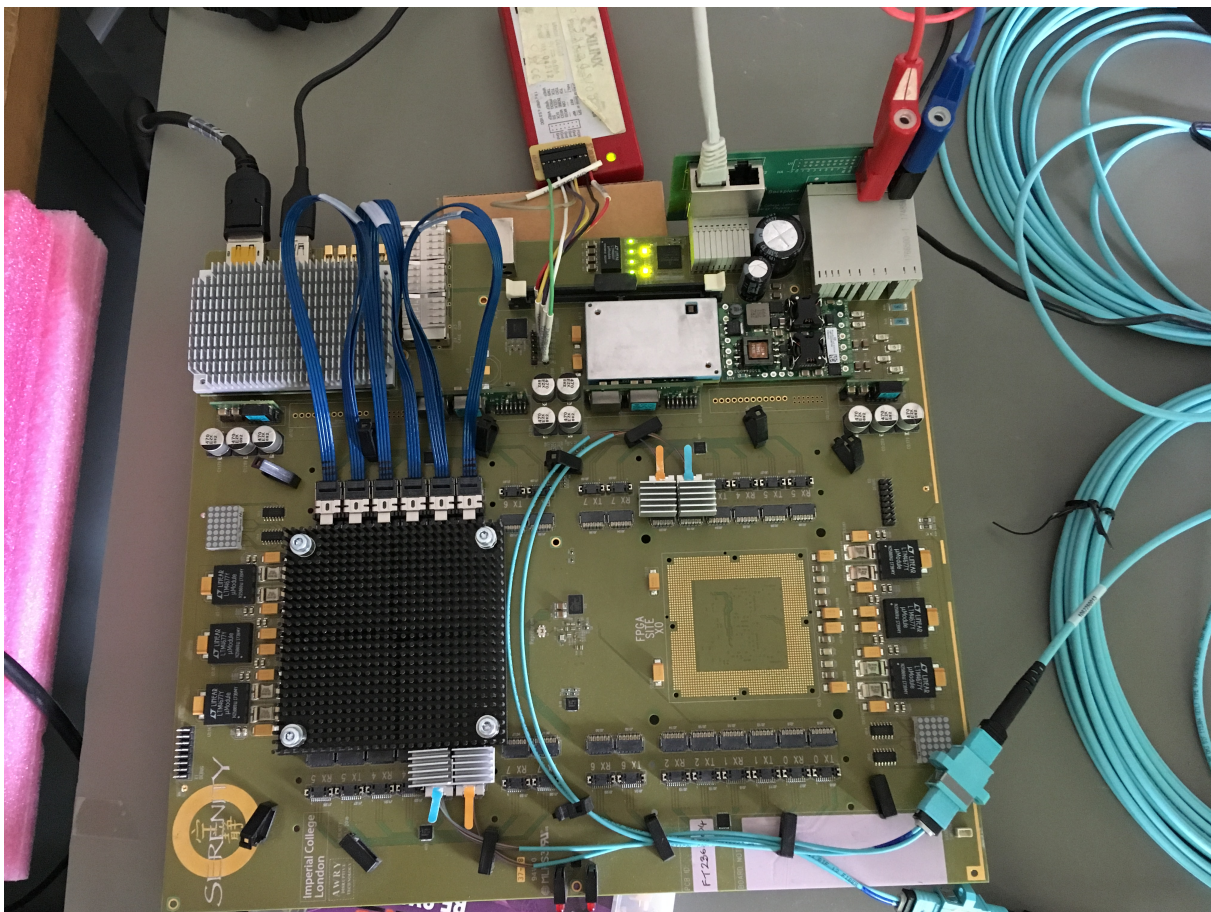
This document describes the first steps to getting started to use the SERENITY prototype system. More details about the SERENITY board can be found here: [SERENITY overview](#) and [technical specs](#).

All content is covered by the Serenity Collaboration Licence ([CC BY-NC-ND 4.0](#)).

FIRST STEPS

It is assumed that you have a pre-configured SERENITY prototype board and access to Vivado software with a connection to the JTAG cable connection on the SERENITY board. Delivered systems will have a “cmx” user account with sudo access, therefore the following commands should work on that given system. Hostnames and specific commands are given as an example.

The SERENITY Prototype board is shown below. One daughter card (DC) site is populated (X1) with a the black heat-sink mounted on the top. The JTAG interface with connections is visible to the top. The ETHERNET connector and ATCA backplane adaptor (power) is mounted on the top right.



1.1 Accessing the Daughter Card Sites:

1. There are three methods of accessing the daughter card sites via JTAG:
 - directly from the COM Express (`serenitybutler program...`) - *fastest* method
 - via a cable connected to the jtag header (labelled “IPMC”)
 - or remotely through the COM Express (XVC)

Below we describe installation on a standard SERENITY board with embedded system as delivered. There is also a [troubleshooting guide](#) which will be expanded as the prototype systems are deployed amongst users.

Most of the configuration is handled by the `serenitybutler` which is explained in more extensive detail in a [cheatsheet](#) reference. For more complete and advanced usage see the [SMASH](#) documentation.

1.2 Installation (SMASH & EMP)

1. (optional) open a vivado session (e.g. on greg-special)
 - navigate to Flow -> Open Hardware Manager specific carrier card (jtag end-point)
 - Daughter Card (DC) FPGA should be powered off, therefore no DC connection is visible
2. Log into the SERENITY host (COM Express)
 - e.g. `ssh -X cmx@serenity-2368-04` (also need sudo permission)
3. Remove existing (legacy) packages in case they exist:

```
sudo yum remove "cactuscore-uhal*" "cactusboards-mp7*" "smash*" "cactusboards-emp*"
↳ "cactusboards-serenity*"
sudo yum groups mark remove smash
sudo yum groups mark remove emp
sudo yum groups mark remove uhal
```

4. Download the appropriate YUM repo files, and copy under `/etc/yum.repos.d` (Requires sudo permission for the account on the SERENITY board and first time needs password):

```
sudo curl https://serenity.web.cern.ch/serenity/board-setup/_downloads/ipbus-sw.
↳ centos7.repo -o /etc/yum.repos.d/ipbus-sw.repo
sudo curl https://serenity.web.cern.ch/serenity/board-setup/_downloads/smash.repo -
↳ o /etc/yum.repos.d/smash.repo
sudo curl https://serenity.web.cern.ch/serenity/board-setup/_downloads/emp.repo -o /
↳ etc/yum.repos.d/emp.repo
sudo curl https://serenity.web.cern.ch/serenity/board-setup/_downloads/serenity.
↳ repo -o /etc/yum.repos.d/serenity.repo
sudo curl http://serenity.web.cern.ch/serenity/emp-fwk/_downloads/cactus-build-
↳ utils.repo -o /etc/yum.repos.d/cactus-build-utils.repo
```

5. Install the software (again with sudo rights):

```
sudo python3.6 -m pip install --upgrade click==8.0.4 click_didyoumean==0.3.0
↳ pexpect==4.8.0 pytest==7.0.1 pexpect==4.8.0 pyyaml==6.0 python-dateutil==2.8.2
↳ colorama==0.4.5
sudo yum clean all
```

(continues on next page)

(continued from previous page)

```
sudo yum groupinstall smash emp uhal
sudo yum install cactuscore-uhal-python36 smash-python36 "cactusboards-serenity*"
↪python36-rpm cactuscore-build-utils-0.2.9 prometheus-node_exporter
```

The yum install command will check for dependencies so more than three packages will be installed. After checking a confirmation is required (y) to proceed with the installation.

The output looks like this (and choose y if asked):

```
120.141.223.37 ~ sudo yum groupinstall smash mp7 emp
Loaded plugins: fastestmirror, langpacks
Determining fastest mirrors
epel/x86_64/metalink | 30 kB 00:00:00
 * base: linuxsoft.cern.ch
 * epel: fr2.rpmfind.net
 * extras: linuxsoft.cern.ch
 * updates: linuxsoft.cern.ch
base | 3.6 kB 00:00:00
cactus-build-utils | 2.9 kB 00:00:00
cactus-mp7-base | 3.6 kB 00:00:00
cactus-mp7-updates | 2.9 kB 00:00:00
emp-sw-autobuild | 3.6 kB 00:00:00
epel | 4.7 kB 00:00:00
extras | 3.4 kB 00:00:00
ipbus-sw-base | 3.6 kB 00:00:00
ipbus-sw-updates | 2.9 kB 00:00:00
smash-sw-autobuild | 3.6 kB 00:00:00
subline-text | 2.9 kB 00:00:00
updates | 3.4 kB 00:00:00
(1/19): base/7/x86_64/group_gz | 166 kB 00:00:00
(2/19): cactus-build-utils/primary_db | 1.8 kB 00:00:00
(3/19): cactus-mp7-base/group_gz | 244 B 00:00:00
(4/19): cactus-mp7-base/primary_db | 5.8 kB 00:00:00
(5/19): base/7/x86_64/primary_db | 6.0 MB 00:00:00
(6/19): cactus-mp7-updates/primary_db | 1.2 kB 00:00:00
(7/19): emp-sw-autobuild/group_gz | 235 B 00:00:00
(8/19): emp-sw-autobuild/primary_db | 3.2 kB 00:00:00
(9/19): epel/x86_64/group_gz | 88 kB 00:00:00
(10/19): epel/x86_64/updateinfo | 988 kB 00:00:00
(11/19): ipbus-sw-base/group_gz | 294 B 00:00:00
(12/19): extras/7/x86_64/primary_db | 188 kB 00:00:00
(13/19): ipbus-sw-base/primary_db | 33 kB 00:00:00
(14/19): epel/x86_64/primary_db | 6.7 MB 00:00:01
(15/19): ipbus-sw-updates/primary_db | 37 kB 00:00:00
(16/19): smash-sw-autobuild/group_gz | 227 B 00:00:00
(17/19): updates/7/x86_64/primary_db | 3.4 kB 00:00:00
(18/19): smash-sw-autobuild/primary_db | 3.8 kB 00:00:00
(19/19): subline-text/primary_db | 2.0 kB 00:00:01
Resolving Dependencies
--> Running transaction check
--> Package cactusboards-emp-core.x86_64 0:0.1.0-0.autobuild.1e59cfa.centos7 will be updated
--> Package cactusboards-emp-core.x86_64 0:0.2.6-0.centos7 will be an update
--> Package cactusboards-emp-python.x86_64 0:0.1.0-0.autobuild.1e59cfa.centos7.python2.7 will be updated
--> Package cactusboards-emp-python.x86_64 0:0.2.6-0.centos7.python2.7 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Updating:
cactusboards-emp-core x86_64 0.2.6-0.centos7 emp-sw-autobuild 282 k
cactusboards-emp-python x86_64 0.2.6-0.centos7.python2.7 emp-sw-autobuild 1.3 M
Transaction Summary
-----
Upgrade 2 Packages

Total download size: 1.5 M
Is this ok [y/n]:
```

Ends with the satisfying Complete!

You can compare against the latest versions by following the reference [Current software and firmware versions](#), which also describes way to fix any problems.

On the SERENITY baseboard ComX there is typically a cmx shared account and it is important that you create your own filesystem under this area. Therefore we suggest a directory using your name or your CERN login (MY_USERNAME):

```
cd /home/cmx
mkdir MY_USERNAME
cd MY_USERNAME
```

1.3 Power up!

This procedure will switch on a daughter card site.

1. Environment

```
source /opt/cactus/bin/serenity/serenity_env.sh
```

2. Power on Daughter Card sites using SMASH:

- If the default SMASH config file variable hasn't already been set:

```
export SMASH_DEFAULT_CONFIG=/path/to/someBoardConfig.smash
```

- For X0 DC:

```
serenitybutler power on x0
```

- For X1 DC:

```
serenitybutler power on x1
```

Here the relevant SMASH commands are issued by the butler script internally. Note that in order for SMASH to function correctly `someBoardConfig.smash` should point to a configuration file that actually represents the hardware configuration you have on your board. It is suggested to look at the packaged configuration files within `/opt/smash/etc/serenity/configuration` and adapt accordingly. More details on SMASH are available within the [documentation](#).

1.4 Uploading firmware to the daughter cards

If you're loading a bitfile onto a Serenity daughter card FPGA for the first time, we recommend that you use one of the EMP null algo builds. The EMP framework provides common infrastructural firmware - as well as corresponding control and monitoring software - to support arbitrary user-created payload firmware, e.g. the reconstruction algorithm in a level-1 track finder or level-1 trigger board. In the null algo builds, the data received on each input channel is fed into the corresponding output channel, after several clock cycles. The null algo bitfiles for the latest EMP release can be found [here](#).

The recommended method for uploading firmware to the daughter card FPGAs is to load the bitfile directly from an application running on the COM express, via the Artix-7 FPGA.

Alternately, the daughter card FPGAs can be programmed via a SMASH XVC (Xilinx Virtual Cable) server, which creates a network-based interface for Vivado to access the JTAG connection via the COM express. Finally, if a JTAG programmer cable is connected to the board, that can be used to load bitfiles instead of the 'direct' and 'XVC server' methods.

Each of these procedures are described below.

1.4.1 Direct programming from the COM express

After the bitfile has been copied onto the board, it can be loaded onto the FPGA using the following Serenity butler command:

```
export SMASH_CONFIG=/path/to/someBoardConfig.smash
serenitybutler program SITE /path/to/top.bit
```

where `SITE` should be changed to `x0` for the FPGA on the X0 site, `x1` for X1, or `both` to program both sites

1.4.2 XVC server

- With the FPGA powered start smash and instantiate the XVC:

```
smash.exe -c /path/to/someBoardConfig.smash "DC0:FPGA Decorate XVC 5000" -b
```

This should then start the virtual cable and print some lines of code (with the correct addresses) which need to be run within the TCL command (or as a script):

- In the vivado session lines like the following should be entered:

```
open_hw
connect_hw_server
open_hw_target -xvc_url 128.141.YYY.XXX:5000
set current_hw_jtag [get_property hw_jtag [get_hw_target localhost:3121/xilinx_tcf/
↪Xilinx/128.141.YYY.XXX:5000]]
```

where XXX and YYY are your own unique subnet address. The 3121 is a Vivado default and the 5000 is the port which needs to be open for communication to work.

- If the XVC fails to connect then there is a chance that the firewall on the ComExpress is somehow blocking it. In which case first check that 5000 is open:

```
sudo firewall-cmd --zone=public --list-ports
```

If port 5000 is not listed then you can first try adding it:

```
sudo firewall-cmd --zone=public --add-port=5000/tcp --permanent
```

But if connection still fails (timeouts) then it is suggested to disable the firewall completely:

```
sudo systemctl stop firewalld
```

Then restart `smash.exe` with the appropriate `decorate` command and re-run the TCL script within Vivado.

- The FPGA should now be visible for programming within Vivado.

More details about SMASH and the associated commands for the virtual JTAG can be found [here](#)

1.4.3 JTAG cable method

Finally, it is also possible to connect directly to the FPGA using a JTAG interface cable according to the following procedure:

- Connect the FGPA to the JTAG header:

- For X0 DC:

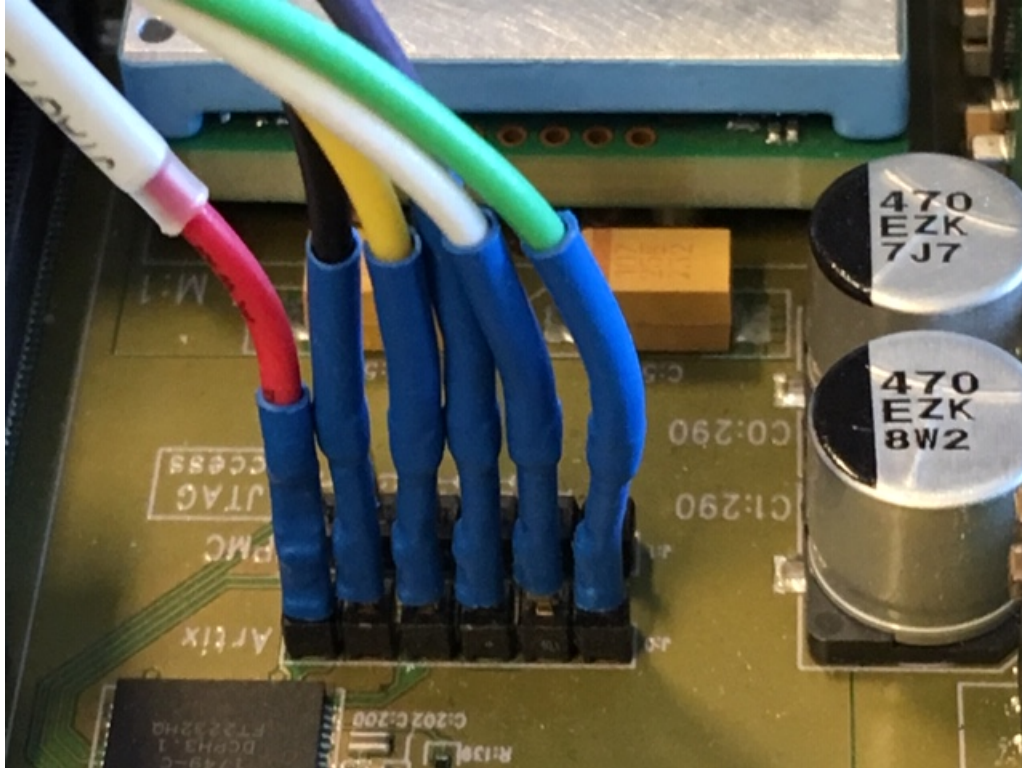
```
smash.exe -c /path/to/someBoardConfig.smash -c /opt/smash/etc/serenity/examples/
↪jtag_header_x0.smash
```

- For X1 DC site:

```
smash.exe -c /path/to/someBoardConfig.smash -c /opt/smash/etc/serenity/examples/
↪jtag_header_x1.smash
```


This tells the ARTIX FPGA to direct the jtag header to that daughter card site.

- Connect the JTAG to the header marked IPMC on the carrier card. Prior to issuing the second “power-up” command the header would be directed to the IPMC, hence the label. But if the above power-up instructions were completed successfully the correct daughter end-point should be visible from Vivado.



In the above a path should be provided for the configuration file for SMASH. Examples of pre-existing configurations can be found under `/opt/smash/etc/serenity/configuration/` however it is important to have the correct configuration for your actual hardware implementation (e.g. FPGA voltages). In addition it is possible to remove the `-c config_file` construction from `smash.exe` commands if you have already set the `SMASH_DEFAULT_CONFIG` environment variable.

1.5 Checking communication with the loaded firmware

In order to check that the firmware was loaded onto the daughter cards successfully, you can run the `serenitybutler info` command which will print the EMP firmware version registers for each daughter card site, along with version information for the software and the Artix.

If you were loading firmware onto the X1 site, but the X0 site is empty, the output of `serenitybutler info` should look like:

```
Software versions
uHAL:      2.7.9
EMP:       0.7.2
SMASH:     0.6.0
Serenity:  0.3.16
```

Artix

(continues on next page)

(continued from previous page)

```
FW version: 0.3.2
```

```
X0
```

```
No daughter card present  
Power: Disabled
```

```
X1
```

```
Power: Enabled  
FPGA: Programmed  
Framework version: 0.3.2 (design 0x42)  
Payload version: 0x12345678
```

1.6 Next Steps:

You can now proceed to understanding the EMP framework and developing an algorithm within the firmware:

- [EMP Framework](#) and the [walkthroughs](#) described therein.
- In addition if you are using a facility test stand then we have created a helpful [cheatsheet](#) which includes concise commands for common operations - e.g. power cycling, configuring clock synthesisers and fireflies.

ARTIX TTC

The TTC firmware for the Artix-7 FPGA on the Serenity can receive a legacy TTC signal from the backplane and retransmit the LHC clock and TTC commands to each of the daughter card sites via LVDS. It can also be used to locally generate a 40MHz clock and TTC commands, allowing for synchronised data transfer between the North and South daughter cards without any external clock/TTC source.

The TTC logic is controlled and monitored using the `serenitybutler` script (in particular, the `artix` subcommands).

Note: Before running the commands in the sections below, you need to create a Serenity butler config file (YAML format) that defines which address table should be used for the Artix.

The address tables for the latest Artix firmware are installed by the SMASH RPMs under `/opt/smash/etc/serenity/base/uHAL/v0.3/`, with the top-file named `top_artix.xml`. The Artix address table is specified by the `artix` field in the Serenity butler config file's address map, and so a minimal config file - sufficient for all of the commands in this page - is:

```
address:  
  artix: /opt/smash/etc/serenity/base/uHAL/v0.3/top_artix.xml
```

The location of this config file can be specified either through the `-c` argument of `serenitybutler`, or by setting the `SERENITY_CONFIG` environment variable - e.g:

```
export SERENITY_CONFIG=/path/to/serenity.xml
```

2.1 Configuring the TTC firmware

The TTC logic can be reset and configured to distribute the clock/TTC signals received from the backplane (i.e. use the external TTC source) by running the following command:

```
serenitybutler artix reset external
```

Or, the TTC logic can be reset and configured to distribute internally-generated clock/TTC signals as follows:

```
serenitybutler artix reset internal
```

2.2 Checking the firmware version

In case you're unsure which version of the Artix firmware is loaded, you can check this by running:

```
serenitybutler artix info
```

CHEATSHEET FOR TEST STANDS (SERENITYBUTLER ETC.)

The Serenity butler can be used to perform the configuration/monitoring procedures that are required in test stands most often, as outlined in the sections below. For some more advanced procedures that don't commonly take place in test stands you may need to use other tools than `serenitybutler`. Internally the butler invokes the required sequences of `SMASH` functions and/or measurements, and it supports a minimal set of command-line options that should cover most of the typical test stand procedures.

Note: Before running any of the commands listed on this page, you should set up your environment variables as follows:

```
source /opt/cactus/bin/serenity/serenity_env.sh
```

You should also either set `SMASH_DEFAULT_CONFIG` to the path of the `SMASH` config file that describes your board's daughter card & firefly configuration, or reference that config file from the `smash` entry in your Serenity butler config file (whose path should be specified using the `SERENITY_CONFIG` variable).

3.1 Version checks

To check the versions of software packages are installed, and what versions of firmware are loaded onto the FPGAs:

```
serenitybutler info
```

3.2 Power control

To turn on power to X0, X1, or to both sites:

```
serenitybutler power on (x0|x1|both)
```

To power cycle the X0 site, X1 site, or both of them:

```
serenitybutler power cycle (x0|x1|both)
```

To turn off power to X0, X1 or to both sites:

```
serenitybutler power off (x0|x1|both)
```

To check the status of the power modules:

```
serenitybutler power status (x0|x1|both)
```

3.3 Reference clock and firefly configuration

Note: All firefly modules and clock synthesisers associated with a site need to be reconfigured after powering on that site.

To configure the clock synthesisers that provide the MGT reference clocks:

```
serenitybutler clocks configure (x0|x1|both)
```

By default, the clock synthesisers will be configured to output 320MHz clocks, which is suitable for operating both 16 and 25Gbps links in EMP-based firmware. However, if you need to have a different reference clock frequency, you can specify a different clock configuration file using the `--config` option.

To configure all Firefly optical modules on the board:

```
serenitybutler fireflys configure
```

To then check the status of the Firefly modules:

```
serenitybutler fireflys status
```

3.4 Programming the daughter card FPGAs

To program the daughter card FPGA(s) at X0, X1 or both sites:

```
serenitybutler program (x0|x1|both) <path_to_bitfile/bitfile_name.bit>
```

You can also add the `-r` flag to the end of this command, in order to power cycle the site(s) and reconfigure the clock synthesisers before loading the bitfile.

3.5 Crosspoint switch

On the Serenity Z1.2 board, there is a high-speed switch that routes the TCDS signals between backplane as the daughter card that is currently in use. The crosspoint switch can be configured to route the TCDS signals appropriately as follows:

```
serenitybutler xpoint-switch configure (x0|x1)
```

3.6 XVC server

To run an XVC server (port 5000) connected to the FPGA at either one or both of the sites:

```
serenitybutler jtag xvc (x0|x1|both)
```

Just leave this command running while you want the XVC server, and stop the script with Ctrl+C whenever you want to stop the server. To change the port number to a non-default value, just add one extra argument - e.g. to run the XVC server on port 9999:

```
serenitybutler jtag xvc (x0|x1|both) 9999
```

Or to run XVC servers for both FPGAs, but on separate ports, e.g. 5000 for X0 and 5001 for X1:

```
serenitybutler jtag xvc both 5000 5001
```

3.7 Artix TTC

Note: In order for the commands in this section to work, the Artix address table must be specified in the Serenity butler config file. The format of that file is discussed in the [Artix TTC](#) page, along with a more in-depth summary of the functionality of the Artix TTC firmware.

To configure the TTC logic to distribute internally-generated clock and TTC signals:

```
serenitybutler artix reset internal
```

To configure the TTC logic to use an external TTC source instead:

```
serenitybutler artix reset external
```

3.8 Test suite

The serenitybutler test suite (implemented in Python, using pytest) validates that the firmware and software are functioning as expected. The tests are organised into many test cases, each of which tests a specific workflow or area of functionality with a specific set of parameters. These test cases are grouped into three test modules:

- `test_i2c` - Power cycle daughter card and configure clock synths
- `test_jtag` - Program daughter card with valid and invalid bitfiles
- `test_ttc` - Reset Artix TTC controller using both internal and external TTC clock and data sources

The test suite can be run as follows:

```
python -m serenity.tests --site [site] --repeat=[N] --config [serenitybutler config_
↪file] --tarball [firmware tarball]
```

For example:

```
python -m serenity.tests --site x0 --repeat=2 --config serenity.yml --tarball bitfiles/
↪serenity_vu13p_so2.tgz --external-ttc-clock-only
```

Warning: If external TTC sources are present, use flag `--external-ttc` (both clock and data sources) or `--external-ttc-clock-only` (clock source only).

If the board PROM is programmed with a bitfile, use flag `--prom-programmed`.

Correctly setting these flags is important so that the test suite knows which tests should be run and which ones should pass and fail.

Note: If you want to only run a subset of tests, you can specify the name(s) or part of the name(s) of the test module or test cases using the `-k` flag. For example, to only run the I2C and TTC tests:

```
python -m serenity.tests -k 'test_i2c or test_ttc' ...
```

Or to skip the firefly tests:

```
python -m serenity.tests -k 'not firefly' ...
```

3.9 Running docker containers

If you are running docker containers that need access to the board's FPGAs, we strongly recommend starting those containers using the `/opt/cactus/bin/serenity/docker-run.sh` script rather than invoking `docker run` directly. It is a wrapper script that invokes `docker run` internally, passes all user-supplied arguments onto that command, and automatically adds other arguments that are required in order to allow applications running in the containers to communicate with the FPGAs.

For example, with the wrapper script, you can run the 'Serenity watch' container in the background (listening on TCP port 8000, and bind mounting a SMASH config to `/board.smash`) as follows:

```
/opt/cactus/bin/serenity/docker-run.sh -d -p 8000:8000 -v /path/to/myBoard.smash:/board.  
↪smash gitlab-registry.cern.ch/p2-xware/software/serenity-watch/serenity-watch:master
```

Note: If there are problems in accessing files in certain circumstances (in particular, PCIe device files after programming the FPGA), then make sure that the PCP SELinux RPM is updated to `pcp-selinux-4.3.2-13.el7_9.x86_64` (e.g. by running `sudo yum update pcp-selinux-4.3.2-13.el7_9.x86_64`) - there appears to be a bug in `pcp-selinux-4.3.2-12.el7`.

3.10 Changing the CPU frequency governor

On Intel Atom COM express modules, it's possible to select from one of two CPU frequency 'governor' settings, that provide a different balance between power usage and performance:

- **powersave:** This governor forces the CPU to the lowest possible frequency
- **performance:** Forces the CPU to use the highest possible frequency

We typically recommend running with the **performance** governor as this setting doesn't pose any problems with cooling or power usage on the Serenity.

The governor can be set at runtime with the following command:


```
sudo cpupower frequency-set --governor GOVERNOR_NAME
```

And you can find out which governor is currently active as follows:

```
sudo cpupower frequency-info --policy
```

TROUBLESHOOTING AND FAQ

- During the yum install phase ([first steps](#)) there is a chance that conflicts or dependencies are not resolved. Particularly if the COM express operating system has been modified. Therefore, any errors should be noted, reported and corrected if necessary.
- All SERENITY prototype boards should have a cmx account, however, all needed components should be located in in /opt (or /opt/emp).
- The environment should be taken care of by the RPM installation and the above instructions. If errors or unfound paths are identified these should also be reported.
- Linux (specifically centos7) is installed on the SERENITY boards and is used for configuring and managing the Daughter Cards, slow control and other infrastructure.
- most packages are installed using RPMs installed via yum. rpm used to mean redhat package manager, but now is used more widely as an rpm package manner. Yum refers to the tool used to manage the rpm installation. more info: [rpm](#) and [yum](#)

4.1 Advanced Troubleshooting

- With direct connection to the display port you see a blank screen:

```
sudo emacs /etc/default/grub
```

then find the following line and make sure it has the nomodeset at the end:

```
GRUB_CMDLINE_LINUX="rd.lvm.lv=cl_dyn20-032-64/root rd.lvm.lv=cl_dyn20-032-64/swap_
↵rhgb quiet nomodeset"
```

then save it and do:

```
sudo grub2-mkconfig --output=/boot/grub2/grub.cfg
```

Rebooting should then fix the issue.

If you switch from a DVI connector to only display port then you actually need to remove the “nomodeset” and repeat, otherwise you will get the same problem reversed.

- A **powered** USB hub is required if you have a keyboard/mouse connected to the USB port on the Serenity.

4.2 Uncommon Issues

If you are using a **Mac** to connect to the board there is a chance that the locale environment is unrecognised. This will give the following error (or something similar):

```
RuntimeError: locale::facet::_S_create_c_locale name not valid
```

The option `set locale environment variables on startup` in the Mac OSX Terminal needs to be **unchecked**.

ADVANCED PROCEDURES

5.1 Memory and Artix service firmware

5.1.1 Memory Devices

The Artix PROM is the following:

- Master SPI (x4)
- Spansion S25FL256SAGBHBA0

The X0/X1 PROMs are the following:

- Master SPI (dual x4 - i.e. x8)
- Spansion S25FL256SAGBHBA0 (S25FL512S in future)
- Appears as S25FL256Sxxxxx0 in vivado (Spansion part with a 256 Mb size)

5.1.2 Create memory file

To make a memory (.mcs) file from a bitfile in Vivado there is a tool (under tools->create memory file) that should be configured with the above devices. This can then be loaded using program memory device after attaching it to the relevant FPGA.

5.1.3 Programming the Artix

In order to program the Artix (either PROM or FPGA directly) the PCIe connection should be removed from the ComExpress otherwise a reboot will be required to recover the CPU.

The `serenitybutler` script's `pcie disconnect` command should be explicitly run prior to programming the Artix or it's associated PROM:

```
serenitybutler pcie disconnect artix
```

If you program only the Artix (bit-file) then `serenitybutler pcie reconnect artix` can be run in order to bring back the PCIe connection to the Artix and then program the DC FPGAs if required. However, if you only program the PROM then the PCIe will remain disconnected until you reboot the card.

5.1.4 Bitfiles for Artix firmware

Currently two different versions of the ARTIX FPGA have been deployed on SERENITY base boards: the a35 and a50 variants. This requires different bit files and associated MCS memory configurations.

First to check which firmware you have loaded on a SERENITY board, access the ComX and then run the following command (NB: you need to have a `serenity.xml` config file set as described here - [Artix TTC](#)):

```
serenitybutler artix info
```

The release notes for the Artix firmware can be found here, and the latest (v0.3. 1) release can be downloaded using the following links:

Artix mcs memory files

- [a35 Memory file](#)
- [a50 Memory file](#)

Artix bit files

- [a35 compressed bit file](#)
- [a50 compressed bit file](#)

Note that loading the bitfiles into the artix FPGA will only be persistent until the next power cycle. Therefore it is recommended that you also program the memory part. Also, before programming the Artix or its PROM via the JTAG header, remember to disconnect the PCIe bus by running:

```
serenitybutler pcie disconnect artix
```

Otherwise, if this command is not run, the operating system will crash, and the board will have to be power cycled to restore access to the ComX.

5.2 On-board computer: Adding CERN accounts

5.2.1 CERN accounts, Kerberos & AFS

In order to allow people to log into a board with their CERN accounts, install Kerberos, and mount AFS, run the following command:

```
sudo /usr/bin/locmap --enable afs
sudo /usr/bin/locmap --configure all
```

To check that the relevant modules have been enabled & configured, then run `sudo /usr/bin/locmap --list` - you should see the following output:

```
[Available Modules]
afs          [ enabled]
cernbox      [disabled]
cvmfs        [disabled]
eosclient    [disabled]
kerberos     [ enabled]
lpadmin      [ enabled]
nscd         [ enabled]
ntp          [ enabled]
```

(continues on next page)

(continued from previous page)

```
sendmail      [ enabled]
ssh           [ enabled]
sudo          [ enabled]
```

Note: You will only be able to use a Kerberos token for authentication with SSH, the board will need to have a fixed IP address. If the output of `host $(hostname)` includes `dyndns.cern.ch`, e.g:

```
BOARD-HOSTNAME.cern.ch is an alias for BOARD-HOSTNAME.dyndns.cern.ch.
BOARD-HOSTNAME.dyndns.cern.ch has address 128.141.143.116
```

then your board has a dynamic IP address and so it will not generate the `/etc/krb5.keytab` file that's required to use Kerberos tokens with SSH.

CERN accounts can then be added to the computer as follows:

```
sudo addusercern SOME_USERNAME
```

If people logged in from CERN accounts will need to be able to read files under `/home/cmx`, run the following command from the `cmx` account:

```
chmod -R a+r /home/cmx
chmod a+x $(find /home/cmx -type d)
```

5.2.2 SSH access for large groups of users

In order to give several CERN accounts SSH access to multiple boards at CERN, you can centrally manage the list of accounts with an e-group, and configure each board to read the membership of that egroup from the central LDAP service.

1. To set this up, you'll first need to create the corresponding egroup. Notably, you'll need to wait until that e-group has been approved. You will also need to wait until the e-group membership has been synchronised - see the "Sync state" field in the "Settings" tab after finding your egroup in <https://e-groups.cern.ch/e-groups/EgroupsSearchForm.do>. You can then verify that the e-group membership is being correctly published by the central LDAP service, by running the following command (make sure that you replace `NAME_OF_MY_EGROUP` with the appropriate value):

```
ldapsearch -LLL -x -h xldap.cern.ch -b "DC=cern,DC=ch" "(&
↳(objectClass=user)(memberOf=CN=NAME_OF_MY_EGROUP,OU=e-groups,OU=Workgroups,DC=cern,
↳DC=ch))" objectGUID userPrincipalName
```

2. Download the SSSD configuration file to your board

```
sudo curl https://serenity.web.cern.ch/serenity/board-setup/_downloads/sssds/sssds.
↳conf -o /etc/sssds/sssds.conf
```

Then open the downloaded file and change `NAME_OF_MY_EGROUP` to the name of the egroup you want to use.

3. Update the SSSD config file's settings

```
sudo chown root:root /etc/sssds/sssds.conf
sudo chmod 0600 /etc/sssds/sssds.conf
sudo restorecon /etc/sssds/sssds.conf
```

4. Enable SSSD

```
sudo authconfig --enablesssd --enablesssdauth --update
```

5. Run SSSD daemon, and ensure that it starts on boot

```
sudo systemctl enable sssd
sudo systemctl stop sssd
sudo systemctl start sssd
```

6. Verify that you can access the information for a CERN account using the following command:

```
getent passwd SOME_CERN_USERNAME
```

Note: These instructions are based off the information in <https://linux.web.cern.ch/docs/account-mgmt/> . However, if the above commands do not work without modifications, please get in touch with Tom Williams to correct this page.

CURRENT SOFTWARE AND FIRMWARE VERSIONS

This page lists the current versions of the software components that are required to use the SERENITY carrier card.

6.1 Recommended versions

The latest versions of software are:

- uHAL: 2.8.1
- EMP: 0.7.2
- SMASH: 0.6.0
- Serenity toolbox: 0.3.16

The current version of the Artix firmware is v0.3.2.

6.2 Checking versions: Recommended method

You should have already followed the instructions in the *First Steps* section and therefore have up to date versions of the RPMs. If you last followed the instructions from that section after Thursday 19th December, then the easiest way to check that you have the latest software and firmware versions is using the `serenitybutler` script - specifically, by running:

```
serenitybutler info
```

This command prints out the currently installed version of each software component, and the versions of firmware currently running on each FPGA. The software and Artix sections of the output should look like:

```
Software versions
uHAL:      2.8.1
EMP:       0.7.2
SMASH:     0.6.0
Serenity:  0.3.16

Artix
FW version: 0.3.2
```

If your software versions differ then the first step should be to issue an update command to packages for which you have repository access:


```
yum update
```

Then re-run `serenitybutler info` to check the software versions again. If you still don't have the latest versions or missing packages then it is recommended that you follow the package installation instructions described under *Installation (SMASH & EMP)* and then run `serenitybutler info` again.

6.3 Checking versions: Alternate method

If you last installed the software on or before 19th December, then you won't have the `serenitybutler` script mentioned above. In that case, you should consider re-running the instructions in the *First Steps* section to get the `serenitybutler`. But if you cannot do that straight away, you can instead verify which versions of software you have on your COM Express by running the following YUM commands:

```
yum grouplist | sed '/^Installed Groups:/, $!d;/^Available Groups:/, $d;/^
↳ Installed Groups:/d;s/^[[:space:]]*//'; yum list installed | grep -E
↳ 'smash|emp-|uhal'
```

The output should look something like this:

```
emp
smash
uhal
cactusboards-emp-core.x86_64      0.3.5-0.centos7      @emp-sw-updates
cactusboards-emp-logger.x86_64    0.3.5-0.centos7      @emp-sw-updates
cactusboards-emp-python.x86_64    0.3.5-0.centos7.python2.7 @emp-sw-updates
cactuscore-uhal-grammars.x86_64    2.6.6-1.centos7.gcc4_8_5 @ipbus-sw-updates
cactuscore-uhal-gui.x86_64         2.6.6-1.centos7.gcc4_8_5.python2.7
cactuscore-uhal-log.x86_64         2.6.6-1.centos7.gcc4_8_5 @ipbus-sw-updates
cactuscore-uhal-pycohal.x86_64     2.6.6-1.centos7.gcc4_8_5.python2.7
cactuscore-uhal-tests.x86_64       2.6.6-1.centos7.gcc4_8_5 @ipbus-sw-updates
cactuscore-uhal-tools.noarch       2.6.6-1.centos7      @ipbus-sw-updates
cactuscore-uhal-uhal.x86_64        2.6.6-1.centos7.gcc4_8_5 @ipbus-sw-updates
smash-components.x86_64            0.2.5-0.centos7.cc4_8_5 @smash-sw-updates
smash-core.x86_64                  0.2.5-0.centos7.cc4_8_5 @smash-sw-updates
smash-logger.x86_64                0.2.5-0.centos7.cc4_8_5 @smash-sw-updates
smash-pcieutils.x86_64             0.2.5-0.centos7.cc4_8_5 @smash-sw-updates
smash-python.x86_64                0.2.5-0.centos7.python @smash-sw-updates
smash-serenity.x86_64              0.2.5-0.centos7.cc4_8_5 @smash-sw-updates
```

Fig. 6.1: Current software versions on the ComX

If your software versions differ then the first step should be to issue an update command to packages for which you have repository access:

```
yum update
```

Then repeat the verification step above. If you still don't have the latest versions or missing packages then it is recommended that you following the package installation instructions described under *Installation (SMASH & EMP)* and then running the above command again.

If the group is marked as installed but the package has not been updated then you can force its removal first:

```
sudo yum groupremove smash emp
```

If the RPM has been removed (individually), but the group is marked at installed then you can force it:

```
sudo yum group mark remove smash emp
```

An installation of the latest software versions should then be possible with:

```
sudo yum clean all && sudo yum groupinstall smash emp
```

Or:

```
yum groupupdate smash emp
```

Lastly to remove all components and start afresh:

```
sudo yum group remove smash emp
```

You can then proceed according to *Installation (SMASH & EMP)* and you should be able to reproduce the latest package list displayed in *Current software versions on the ComX*.

STATUS OF THIS DOCUMENT

First steps to using the SERENITY prototype system.

- Rev 1.0: First version - Alex Howard, 3rd April 2019.
- Rev 1.1: Second version - Alex Howard, 9th April 2019.
- Rev 1.2: Added latest smash functionality and XVC - Alex Howard, 28th June 2019.