OpenCPI Standalone Training

Version 1.4.0

Revision History

Revision	Description of Change	Date
v1.2	Initial creation	9/2017
v1.3	Minor tweaks for OSS Release and features	2/2018
v1.4	Moderate tweaks for OpenCPI 1.4	2/2019

Contents

1 Overview

This document describes how to set up a CentOS 7 machine to allow a user to perform the Labs found in the ANGRYVIPER Team's OpenCPI training. The training is originally designed to be an **on-site four-day course**, but a self-paced review of most of the course material is possible. We will enumerate and elaborate the required tasks that the *Lab Work Environment* slide deck "hand waves."

This document assumes the user has `root` access on the machine and a Matchstiq-Z1 is available. If using a different target platform, the synthesis instructions *will* be different and the main OpenCPI documentation and the platform's *Getting Started Guide* should be consulted.

The authors intend this document to be printed out to provide a physical checklist for reference.

This process will make *permanent* changes to the security profile of your system. These procedures should only be performed with IT and/or Security approval.

1.1 Additional Resources

If you were provided *this* document in hard copy or on media, you can find the latest version on `github.io` at the bottom of `https://opencpi.github.io/`. This page also includes the official documentation for the project, a useful "*Acronyms and Definitions*" sheet, and the various documents noted elsewhere below, *e.g.* "the *Matchstiq-Z1 Getting Started Guide*."

Since the in-person training includes a walk-through of the GUI that is missing here, the "*ANGRYVIPER IDE User Guide*" might also be a useful reference to consult.

2 OS and Tool Installation

Unless explicitly noted otherwise, every command-line entry within this section assumes it is being executed by the `root` user or using `sudo`.

- ☐ All RPMs found in the RPM-based OpenCPI distribution should be installed. This will ensure all prerequisites are installed. Consult the ANGRYVIPER Team's *Installation* and *Getting Started Guides* for specifics.
- ☐ Additional software installation:
  - ☐ `yum install -y epel-release`
  - ☐ `yum install -y scipy python-matplotlib git autoconf automake make screen gcc-c++`
- ☐ Xilinx's Vivado software should be installed into `/opt/Xilinx/` (the "main" software as well as the 2013.4 SDK). Consult the *FPGA Vendor Tools Installation Guide* for details, noting requirements for hardware and software development targeting the Zynq-ARM build environment. *This installation will include the* `xsim` *simulator, so no other software from that document is required.*
- ☐ A Xilinx WebPACK license should be acquired and copied to `/opt/Xilinx/Xilinx.lic`; again consult the *FPGA Vendor Tools Installation Guide* for details.
- ☐ A user named `training` should be created.
- ☐ The user `training` must be added to the `opencpi` group:
  `sudo usermod -aG opencpi training`
- ☐ A file should be created at `/opt/opencpi/cdk/env.d/xilinx.sh` (see `xilinx.sh.example`) containing:
  `export OCPI_XILINX_LICENSE_FILE=/opt/Xilinx/Xilinx.lic`
- ☐ Log out the `training` user and log back in. Ensure `OCPI_XILINX_LICENSE_FILE` has been set.
- ☐ Ensure the GUI can be launched using the command "`ocpigui`" as the `training` user.

3 Building the Core and Assets Project

The instructions given in the *Matchstiq-Z1 Getting Started Guide* should be followed in conjunction with this checklist to copy and build the `core` and `assets` projects. **The use of the IDE to do these builds is not advised at this time** because the training will guide its usage later.

> Unless explicitly noted otherwise, every command-line entry within this section assumes it is being executed by `training` from their home directory.

☐ Install local copies of Core and Assets Projects:
```
ocpi-copy-projects /home/training/
```

☐ Build Core Project (following instructions in *Matchstiq-Z1 Getting Started Guide*)

Append "`--hdl-platform xsim`" to any `ocpidev` commands whenever you specify HDL platform, *e.g.*
```
ocpidev build --rcc --rcc-platform centos7 --rcc-platform xilinx13_3 \
--hdl --hdl-platform matchstiq_z1 --hdl-platform xsim --no-assemblies
```

☐ Build Assets Project (following instructions in *Matchstiq-Z1 Getting Started Guide*)

Append "`--hdl-platform xsim`" to any `ocpidev` commands whenever you specify HDL platform.

☐ Build the required assemblies from Assets (from `/home/training/assets`):
```
ocpidev build --rcc --rcc-platform centos7 --rcc-platform xilinx13_3 --hdl \
 --hdl-platform xsim --hdl-platform matchstiq_z1 --hdl-assembly fsk_filerw \
 --hdl-assembly fsk_modem
```

☐ Build the test application (from `/home/training/assets/applications/FSK`):
```
ocpidev build --rcc-platform centos7 --rcc-platform xilinx13_3
```

4 NFS Exporting

This section assumes CentOS 7's `firewalld` is in use and is treating the current connection as the "public zone." This can be queried using "`firewall-cmd --get-active-zones`". Any other configuration will require changes to allow NFS connections as performed below; consult your local IT if `firewalld` is not in use. **Some of these steps may have already been performed, *e.g.* if the user already followed the instructions given in the *Matchstiq-Z1 Getting Started Guide*.**

> Unless explicitly noted otherwise, every command-line entry within this section assumes it is being executed by the `root` user or using `sudo`.

☐ Have SELinux allow NFS mounting read/write from home directories:
```
setsebool -P nfs_export_all_rw 1
setsebool -P use_nfs_home_dirs 1
```

☐ Allow NFS and related connections through the machine's firewall
```
firewall-cmd --permanent --zone=public --add-service=nfs
firewall-cmd --permanent --zone=public --add-port=2049/udp
firewall-cmd --permanent --zone=public --add-service=mountd
firewall-cmd --permanent --zone=public --add-service=rpc-bind
firewall-cmd --reload
```

☐ Configure NFS exporting by creating at `/etc/exports.d/training.exports` containing[1]:
```
/home/training *(rw,sync,no_root_squash,crossmnt,fsid=33)
/opt/opencpi *(rw,sync,no_root_squash,crossmnt,fsid=34)
```

☐ Enable NFS sharing[2]:

[1] If other NFS exports exist in `/etc/exports.d/`, ensure the `fsid` values are unique across *all* shares.

[2] Some commands may report errors based on installed packages and point release. If everything works otherwise, they can likely be ignored.

```
exportfs -r
exportfs -v
systemctl enable rpcbind
systemctl enable nfs-server
systemctl enable nfs-lock
systemctl enable nfs-idmap
systemctl restart rpcbind
systemctl restart nfs-server
systemctl restart nfs-lock
systemctl restart nfs-idmap
rpcinfo -p | grep nfs
```

The second command should show the two directories being exported (along with additional flags the OS may add, *e.g.* "`wdelay`"). The last command should report that the RPC daemon is aware of "nfs" and "nfc_acl" on port 2049 for both TCP and UDP.

5 Installing the Training

Within the training repo or tarball, you will find the following directory structure:

Directory	Usage
`PDFs`	Documents (including this document)
`PDFs/datasheets`	Datasheets for the Components used in training
`PDFs/presentations`	Training presentation slides
`provided`	The "provided" files for each individual lab, referenced by lab documentation
`training`	The "completed" lab solution project; this is *not* normally provided to the student and will be "hidden" below

Some of these directories are expected to be in certain locations to match the on-site training locations, *e.g.* the script that resumes Labs in Section 9. After extraction, please `mv` these directories to match:

Directory	Target location
`PDFs`	*(N/A; they can be anywhere)*
`provided`	`/home/training/provided`
`training`	`/home/training/.dist/training`

6 Matchstiq SD Card Modifications

This section assumes you start with an SD image and Matchstiq-Z1 that are a result of the setup instructions given in the *Matchstiq-Z1 Getting Started Guide* **for a "Network Mode" SD image with the appropriate scripts configured**.

> Unless explicitly noted otherwise, every command-line entry within this section assumes it is being executed by the `root` user or using `sudo`.

☐ Mount the SD image's *first* partition (vfat) to a directory[3], *e.g.* `/mnt/sd/p1`

☐ Change to the `opencpi` subdirectory on the SD

☐ Modify `mynetsetup.sh` to add the following[4] after "add any commands to be run only the first time":

```
# Training mounts
mkdir -p /mnt/ocpi_core
mount -t nfs -o udp,nolock,soft,intr $1:/home/training/core /mnt/ocpi_core
mkdir -p /mnt/ocpi_assets
mount -t nfs -o udp,nolock,soft,intr $1:/home/training/assets /mnt/ocpi_assets
mkdir -p /mnt/training_project
mount -t nfs -o udp,nolock,soft,intr $1:/home/training/training_project /mnt/training_project
```

[3] Your OS may mount this automatically, *e.g.* `/run/media/training/ATLAS`.

[4] Other mounts may have already been defined by following the *Matchstiq-Z1 Getting Started Guide*. Those should be commented out for our purposes.

7 Slide Presentation Order

The usual order of the in-person training is as follows:

1. ☐ Overview (`Course_01_OpenCPI_Overview.pdf`)
2. ☐ General Concepts (`Course_02_Concepts.pdf`)
3. ☐ Demo of the FSK App[5] (`Course_03_FSK_App_Demo.pdf`)
4. ☐ Lab Environment Overview (`Course_04_Lab_Environment.pdf`)
5. ☐ IDE Overview and Presentation (*No Slide Deck*)
6. ☐ Lab 1 (`Lab1_AppDevelopment.pdf`)
7. ☐ Lab 2 (`Lab2_AppDevelopment.pdf`)
8. ☐ Component / Worker Overview (`Course_05_Component_and_Worker_Overview.pdf`)
9. ☐ Application Control Interface (ACI) (`Course_06_ACI_Overview.pdf`)
10. ☐ Overview of Unit Tests (`Course_07_Automated_Unit_Test.pdf`)
11. ☐ RCC Application Workers (`Course_08_RCC_Development.pdf`)
12. ☐ Lab 3[6] (`Lab3_Peak_Detector.pdf`)
13. ☐ Unit Test Deeper Look (`Course_09_Automated_Unit_Test_Details.pdf`)
14. ☐ Lab 4 (`Lab4_ComplexMixer.pdf`)
15. ☐ Lab 5 (`Lab5_TimeDemux.pdf`)
16. ☐ HDL Application Workers (`Course_10_HDL_App_Workers.pdf`)
17. ☐ Lab 6 (`Lab6_PeakDetector_HDL.pdf`)
18. ☐ HDL Assemblies (`Course_11_HDL_Assemblies.pdf`)
19. ☐ HDL Primitives (`Course_12_HDL_Primitives.pdf`)
20. ☐ Lab 7 (`Lab7_AgcComplex_HDL.pdf`)
21. ☐ Lab 8 (`Lab8_ComplexMixer_HDL.pdf`)
22. ☐ Tools / Debugging (`Course_13_Tools_Debugging.pdf`)
23. ☐ Lab 9[7] (`Lab9_Counter_Debugging_HDL.pdf`)

Advanced Topics

These are slide decks that begin to explain how to develop Board Support Packages (BSPs) for OpenCPI. They present topics and information that is more advanced than any Application Developer would need.

1. ☐ Device Workers (`Extra_Device_Support_Development.pdf`)
2. ☐ Intro to Platform Development (`Course_14_Intro_Platform_Development.pdf`)
3. ☐ (Continued) Platform Development (`Extra_Platform_Development.pdf`)

[5] This is an instructor-led demonstration and included here only for completeness.
[6] Lab 3 is performed within the middle of the RCC Development Lecture.
[7] Has problems, is being actively reworked *now* (2019-02)

8 Specific Lab Notes

8.1 Lab 4

This uses a copy of the Liquid DSP libraries that we've provided; this concept can be extended to any third-party library that you need to use. The OpenCPI build system has hooks to handle cross-compilation for you.

8.2 Lab 5

There are more advanced C++11 templated examples available in the file "`time_demux_advanced_reference.cc`" to show possible approaches to copying data across different Protocols.

9 Catching Up on Failed Labs

> Unless explicitly noted otherwise, every command-line entry within this section assumes it is being executed by `training` from their home directory.

There is a script that allows you to "catch up" and complete a set of Labs to use if a previous Lab failed[8]. It assumes that Section 3 has been completed. To use it:

```
rm -rf training_project # Or rename it
./provided/scripts/complete_lab.sh 4 # Replace 4 with desired lab to COMPLETE
# Or, if you want the HDL built and xsim tests run, set BUILD_HDL. This will take much much longer to complete:
BUILD_HDL=1 ./provided/scripts/complete_lab.sh 6
```

Note: Running this script will cause `training_project` to be recreated. This can result in "stale" NFS mounts; rebooting the radio is recommended whenever this script is used.

10 "Missing" Items

Upon completion of all steps within this document, the following differences still exist between the on-site training setup and the newly-configured system:

- `ocpi.assets` datasheets are not on the `training` user's desktop
- PDF copies of the presentation slides and labs are not on the `training` user's desktop

[8] Lab 1 is *not* supported, and Lab 2 is discarded if you request Lab 3 or beyond.