

OpenCPI FSK App Getting Started Guide

Version 1.3

Revision History

Revision	Description of Change	Date
v1.1	Initial Release	3/2017
v1.2	Updated for OpenCPI Release 1.2	8/2017
v1.3	Updated for OpenCPI Release 1.3	1/2018

Table of Contents

1	References	4
2	Overview	5
3	Prerequisites	5
4	Build the Core Project	5
5	Build the Assets Project	5
6	Build the FSK Application Executable	5
7	Running the Application	5
7.1	Setting Up the Execution Platform	5
7.1.1	Networked Mode	5
7.1.2	Standalone Mode	6
7.2	Setting OCPI_LIBRARY_PATH Environment Variable	6
7.2.1	Networked Mode	6
7.2.2	Standalone Mode	6
7.3	Running the Application	6
7.4	View the Results	6

1 References

This document assumes a basic understanding of the Linux command line (or “shell”) environment. The reference(s) in Table 1 can be used as an overview of OpenCPI and may prove useful.

Table 1: References

Title	Published By	Link
Getting Started	ANGRYVIPER Team	Getting_Started.pdf
Installation Guide	ANGRYVIPER Team	RPM_Installation_Guide.pdf
Acronyms and Definitions	ANGRYVIPER Team	Acronyms_and_Definitions.pdf
Overview	OpenCPI	https://goo.gl/RskxiV
FSK App ¹	OpenCPI	FSK_app.pdf

¹Provides details of the “FSK App” reference application

2 Overview

This purpose of this document is to provide a compact set of instructions to build, run, and verify the OpenCPI FSK App reference application.

3 Prerequisites

This document assumes that the OpenCPI framework has been installed. The application is supported on all of the OpenCPI platforms, but all of the examples shown here are for the Matchstiq-Z1 platform.

4 Build the Core Project

If the Core Project has not been created yet, follow the instructions in the OpenCPI Getting Started Guide. Once the Core project has been created, the following `ocpidev` command can be used to build the primitives and workers required by the FSK app. Navigate to the Core project directory and run the command:

```
ocpidev build --rcc --rcc-platform xilinx13_3 --hdl --hdl-platform matchstiq_z1 --no-assemblies
```

Note: The `--no-assemblies` argument excludes the creation of executable bitstreams.
This step takes approximately 20 minutes to complete.

5 Build the Assets Project

If the Assets Project has not been created yet, follow the instructions in the OpenCPI Getting Started Guide. Once the Assets project has been created, the following `ocpidev` command can be used to build the primitives and workers required by the FSK app. Navigate to the Core project directory and run the command:

```
ocpidev build --rcc --rcc-platform xilinx13_3 --hdl --hdl-platform matchstiq_z1 --hdl-assembly fsk_modem
```

This step takes approximately 60 minutes to complete.

6 Build the FSK Application Executable

Next, the executable for the FSK Application must be built. Navigate to the `applications/FSK` of the Assets project and run the command:

```
ocpidev build --rcc-platform xilinx13_3
```

If successful, a new directory name `target-linux-x13_3-arm` will contain the executable.

7 Running the Application

The following steps can also be viewed by navigating to the `applications/FSK` directory and typing `make show`. Ensure that the RX and TX ports of the platform are connected together via a coax cable, RF attenuation pads are not necessary for this test.

7.1 Setting Up the Execution Platform

For embedded platforms (Matchstiq-Z1, Zedboard), connect to the platform via a serial port. Example syntax for establishing a connection can be seen below:

```
screen /dev/ttyUSB0 115200
```

7.1.1 Networked Mode

Networked Mode is one of two modes typically used for running the application. It involves creating NFS mounts between the Development Platform and the Execution Platform to enable Application deployment. A setup script is used to automate the required steps for this mode. Example syntax for running this script can be seen below:

```
source /mnt/card/opencpi/mynetsetup.sh <IP Address of Development Host>
```

7.1.2 Standalone Mode

Standalone Mode is the other mode used for running the application. It involves deploying the application using files only on local storage and not over a network connection. Similar to Networked Mode, the setup for Standalone Mode also involves running a setup script. Example syntax for running this script can be seen below:

```
source /mnt/card/opencpi/mysetup.sh
```

7.2 Setting OCPI_LIBRARY_PATH Environment Variable

The OCPI_LIBRARY_PATH environment variable is a colon-separated list of files/directories which is searched for executable artifacts during Application Deployment. For proper execution of the FSK App, all of the FSK App artifacts (detailed in the FSK App document) must be included in the OCPI_LIBRARY_PATH environment variable. Example syntax of the setting can be seen below:

7.2.1 Networked Mode

In networked mode, artifact files can be accessed via NFS mounts.

```
export OCPI_LIBRARY_PATH=/mnt/net/cdk/../../projects/core/exports/lib/components/rcc:\
/mnt/ocpi_assets/applications/FSK/../../hdl/assemblies/fsk_modem:\
/mnt/ocpi_assets/applications/FSK/../../exports/lib
```

7.2.2 Standalone Mode

In standalone mode, artifact files can be copied to the SD card.

```
export OCPI_LIBRARY_PATH=/mnt/card/opencpi/artifacts
```

7.3 Running the Application

To run the application, navigate to the `applications/FSK` directory and run the executable in the `target-*` directory of your Execution Platform. Example syntax can be seen below.

```
./target-linux-x13_3-arm/FSK txrx
```

Reference the *FSK_app.pdf* for executing the various modes (filerw, tx, rx, txrx, bbloopback) of the FSK application. Ensure that OCPI_LIBRARY_PATH is configured, as provided, for the respective mode and platform.

7.4 View the Results

After the application completes, the output can be found in the `applications/FSK/odata` directory. To view the results on the Development Host, navigate to the `applications/FSK` directory and execute the command:

```
eog odata/out_app_fsk_txrx.bin
```