

# OpenCPI

## FSK Digital Radio Controller App Guide

Version 1.5

### *Revision History*

Revision	Description of Change	Date
v1.5	Initial release.	4/2019

## Table of Contents

<b>1</b>	<b>Description</b>	<b>2</b>
1.1	Application Modes . . . . .	2
1.1.1	Non-Supported Modes / Hardware-Specific Testing . . . . .	2
<b>2</b>	<b>Hardware Portability</b>	<b>3</b>
2.1	Supported Hardware . . . . .	3
2.2	Expanding Hardware Support . . . . .	3
<b>3</b>	<b>Execution</b>	<b>3</b>
3.1	Prerequisites . . . . .	3
3.2	Command(s) . . . . .	4
<b>4</b>	<b>Verification</b>	<b>4</b>
4.1	filerw, txrx Modes . . . . .	4
4.2	txrx Mode Extensive Testing . . . . .	4
<b>5</b>	<b>Troubleshooting / Known Issues</b>	<b>5</b>
5.1	Limited RX Carrier Recovery Ability . . . . .	5
5.2	Error Message Explanation . . . . .	5

## List of Figures

1	Application block diagram. . . . .	2
2	Expected output file's image for filerw and txrx modes. . . . .	4

# 1 Description

This application is a 2-level FSK modem that is based on the `dig_radio_ctrlr` component spec which provides a fully RF transceiver-agnostic command/control mechanism for RF transceivers. The modulation algorithm reads a JPEG file from disk which has been modified to include a bit synchronization pattern that the demodulation algorithm's `real_digitizer` worker is hardcoded to recognize.

## 1.1 Application Modes

The application may be run in either the *filerw* or *txrx* mode. The application XML filename is sent as the last argument to the application executable. This filename is in the format `<description>_<mode>.xml`, e.g. `foo_txrx.xml` would force the executable to use the *txrx* mode. The *filerw* mode uses `file_read` and `file_write` workers to process the input using only application workers (platform-agnostic) in a purely digital fashion. The *txrx* mode is the full transceiver mode of the application. This mode transmits input file data as the radio RF TX output and inputs RF RX radio input that is written to file.

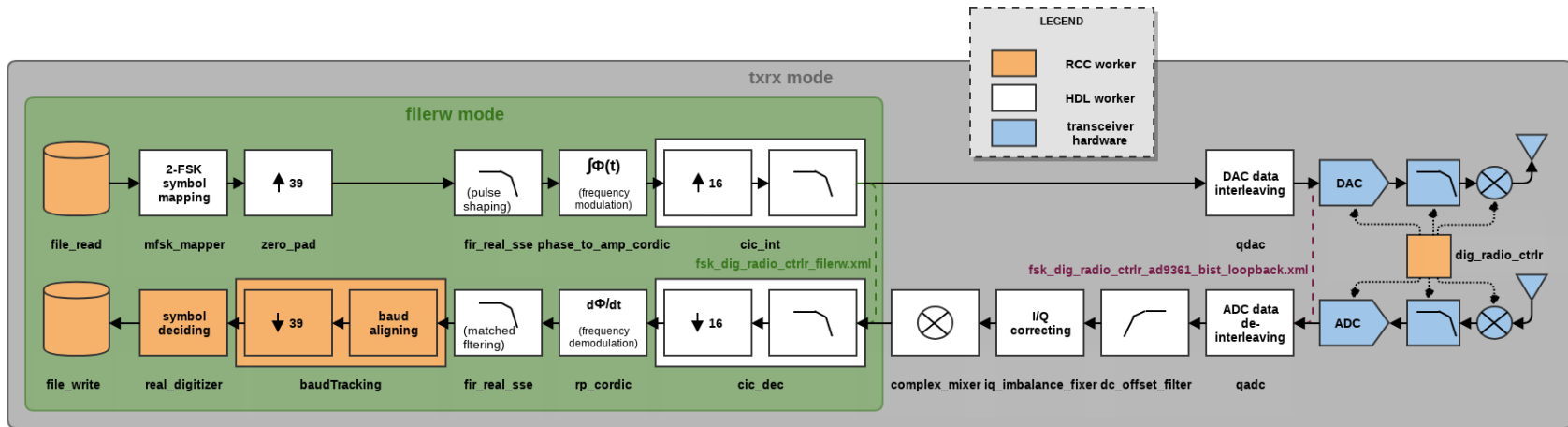


Figure 1: Application block diagram.

### 1.1.1 Non-Supported Modes / Hardware-Specific Testing

For purposes of debugging hardware-specific test settings which the hardware-agnostic ACI cannot support, application XMLs may exist for use with `ocpirun`. For example, `fsk_dig_radio_ctrlr_ad9361_bist_loopback.xml` is used to exercise the AD9361 BIST loopback, which is a digital loopback of the DAC data to the ADC data. This is similar to `filerw` mode but the loopback is performed on the AD9361 itself instead of the FPGA. This is useful for testing the AD9361 FPGA-to-DAC and ADC-to-FPGA interface data fidelity as it pertains to this application. See file contents header example usage with `ocpirun`.

## 2 Hardware Portability

The application executable and ACI source code are fully hardware-portable. Each application XML which does not end in `filerw.xml` is currently specific to an RF transceiver.

### 2.1 Supported Hardware

This application provides support for any one of the following hardware configurations:

- CentOS6/7 x86 machine, Xilinx ML605 PCIE card, FMCOMMS2 FMC transceiver card
- CentOS6/7 x86 machine, Xilinx ML605 PCIE card, FMCOMMS3 FMC transceiver card
- Digilent Zedboard running OpenCPI xilinx13.3 OS, FMCOMMS2 FMC transceiver card
- Digilent Zedboard running OpenCPI xilinx13.3 OS, FMCOMMS3 FMC transceiver card
- Any other RF transceiver platform/card which has a project containing a `dig_radio_ctrlr` worker implementation and has an application XML for this application's txrx mode.

### 2.2 Expanding Hardware Support

To add support to this application for a new RF transceiver, implement a `dig_radio_ctrlr` worker for the new transceiver and create a new application XML for this application's txrx mode. Note that the application filename will be used as an argument to the executable.

## 3 Execution

**WARNING:** While this application may be run by using `ocpirun` directly on any of the application XMLs, it is highly recommended to use the application executable to avoid misuse.

### 3.1 Prerequisites

The following must be true before application execution:

- Either
  - a `zed` or `ml605` platform is available with an FMCOMMS2/3 card in any available FMC slot, or
  - another RF transceiver platform is available whose Board Support Package exists in another project.
- The following assets are built and their build artifacts (FPGA bitstream file/shared object file) are contained within the directory list of the `OCPI_LIBRARY_PATH` environment variable.
  - `fsk_filerw` assembly
  - `fsk_modem` assembly (from assets project, or same assembly XML w/ another BSP-specific container XML in another BSP-specific project)
  - `file_read.rcc` (from core project)
  - `mfsk_mapper.rcc`
  - `zero_pad.rcc`
  - `Baudtracking_simple.rcc`
  - `real_digitizer.rcc`
  - `file_write.rcc` (from core project)
- If using a platform with multiple slots available, the intended slot-specific bitstream must occur first in the `OCPI_LIBRARY_PATH`
- The current directory is the `applications/fsk_dig_radio_ctrlr` directory.

## 3.2 Command(s)

See demo scripts in the application's scripts sub-directory[2] for example usage. The executable's `--help` option is also useful.

# 4 Verification

## 4.1 filerw, txrx Modes

Upon completion of a successful run, the application returns an exit status of 0 and a human-visible Orioles logo is saved in an output file. The output file can be viewed on x86 Centos6/7 machines by running the following command. The ACI prints out the filename of the output file.

```
./scripts/view.sh <output-filename>
```

The output file should appear as follows.



Figure 2: Expected output file's image for filerw and txrx modes.

## 4.2 txrx Mode Extensive Testing

In the assets project, there also exists platform-specific test scripts for SMA loopback using the txrx mode in the scripts/test subdirectory. These test a range of executable arguments for the intended platform and/or card. These scripts return an exit status of 0 and print PASSED to the screen upon success, and return with a non-zero exit status and print FAILED to the screen upon failure.

## 5 Troubleshooting / Known Issues

### 5.1 Limited RX Carrier Recovery Ability

The demodulation algorithm currently suffers from limited carrier recovery ability which can cause the output image to be corrupted or non-existent when using the *txrx* mode. If using an SMA loopback for *txrx* mode on an RF transceiver where the RX and TX data stream's LOs are sourced by the same clock (which is the case on FMCOMMS2/3), carrier recovery is not expected to be an issue.

### 5.2 Error Message Explanation

```
ERROR: exception caught: Worker dig_radio_ctrlr produced error during execution: config lock \
request was unsuccessful, set OCPI_LOG_LEVEL to 8 (or higher) for more info
```

Usually this means a config setting was requested that was outside of the valid ranges of the RF transceiver. Note that this application currently has fixed upsampling/downsampling rates which will limit the maximum baud rate/-transceiver sampling rate (see 1). Set the log level to 8 to see more info about the `dig_radio_ctrlr` worker's actions. The `dig_radio_ctrlr` logging is implementation-dependant, but is intended to be as follows.

- Enumerates the actual on-hardware values, usually with high precision, that were applied to the RF transceiver.
- Upon a request for a configuration value that it outside the current possible range, range of currently valid values for said configuration is given. Note that the range of valid values for any given configuration may depend on the requested values for other configurations, even for seemingly unrelated data streams, e.g. RX dependent upon TX. Log level 10 is intended to provide even more information about currently valid values for the transceiver, including enumerating the valid values for all configurations for all data streams at any moment that one of the configurations changes.

```
Calibration TIMEOUT (0x16, 0x10)
```

This error message can occur when using one of the FMCOMMS2/3 cards and indicates a baseband RX analog filter tune calibration failure for the AD9361 RF transceiver IC. For more info on this calibration failure see [1]. This failure is a known intermittent issue with the AD9361 hardware and/or the `dig_radio_ctrl_fmcomms_2.3.rcc` worker's underlying No-OS library which can occur when setting low sampling rates. While simply running the application again with the same settings might remedy the problem, the recommended remedy is to run the application at a baud rate greater than or equal to 5699 baud. Note also that when this error occurs, it will likely be followed by a config lock request failure.

```
ad9361_init : Unsupported PRODUCT_ID 0xC0ad9361_init : AD936x initialization error
```

This error message can occur when using one of the FMCOMMS2/3 cards and indicates a hardware communication error between the FPGA and the AD9361 RF transceiver IC. This message would occur, for example, if the FMCOMMS2/3 card was not plugged in to the PCB containing the FPGA.

## References

- [1] AD9361 Reference Manual UG-570  
AD9361\_Reference\_Manual\_UG-570.pdf
- [2] `opencpi/projects/assets/applications/fsk_dig_radio_ctrlr/scripts` at develop · `opencpi/opencpi` · GitHub  
[https://github.com/opencpi/opencpi/blob/develop/projects/assets/applications/fsk\\_dig\\_radio\\_ctrlr/scripts/](https://github.com/opencpi/opencpi/blob/develop/projects/assets/applications/fsk_dig_radio_ctrlr/scripts/)