
AdafruitBD3491FS Library Documentation

Release 1.0

Bryan Siepert

May 28, 2019

Contents

1	Dependencies	3
1.1	Installing from PyPI	3
2	Usage Example	5
3	Contributing	7
4	Building locally	9
4.1	Zip release files	9
4.2	Sphinx documentation	9
5	Table of Contents	11
5.1	Simple test	11
5.2	adafruit_bd3491fs	11
5.2.1	Implementation Notes	11
6	Indices and tables	15
	Python Module Index	17

CircuitPython library for the Rohm BD3491FS Audio Processor

This driver depends on:

- [Adafruit CircuitPython](#)
- [Bus Device](#)
- [Register](#)

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the Adafruit library and driver bundle.

1.1 Installing from PyPI

On supported GNU/Linux systems like the Raspberry Pi, you can install the driver locally [from PyPI](#). To install for current user:

```
pip3 install adafruit-circuitpython-bd3491fs
```

To install system-wide (this may be required in some cases):

```
sudo pip3 install adafruit-circuitpython-bd3491fs
```

To install in a virtual environment in your current project:

```
mkdir project-name && cd project-name
python3 -m venv .env
source .env/bin/activate
pip3 install adafruit-circuitpython-bd3491fs
```


CHAPTER 2

Usage Example

```
import board
import adafruit_bd3491fs
import busio

i2c = busio.I2C(board.SCL, board.SDA)
bd3491fs = adafruit_bd3491fs.BD3491FS(i2c)

bd3491fs.active_input = adafruit_bd3491fs.Input.A
bd3491fs.input_gain = adafruit_bd3491fs.Level.LEVEL_20DB
bd3491fs.channel_1_attenuation = 0
bd3491fs.channel_2_attenuation = 0
```


CHAPTER 3

Contributing

Contributions are welcome! Please read our [Code of Conduct](#) before contributing to help this project stay welcoming.

4.1 Zip release files

To build this library locally you'll need to install the `circuitpython-build-tools` package.

```
python3 -m venv .env
source .env/bin/activate
pip install circuitpython-build-tools
```

Once installed, make sure you are in the virtual environment:

```
source .env/bin/activate
```

Then run the build:

```
circuitpython-build-bundles --filename_prefix adafruit-circuitpython-bd3491fs --
↳library_location .
```

4.2 Sphinx documentation

Sphinx is used to build the documentation based on rST files and comments in the code. First, install dependencies (feel free to reuse the virtual environment from above):

```
python3 -m venv .env
source .env/bin/activate
pip install Sphinx sphinx-rtd-theme
```

Now, once you have the virtual environment activated:

```
cd docs
sphinx-build -E -W -b html . _build/html
```

This will output the documentation to `docs/_build/html`. Open the `index.html` in your browser to view them. It will also (due to `-W`) error out on any warning like Travis will. This is a good way to locally verify it will pass.

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/bd3491fs_simpletest.py

```
1 import board
2 import busio
3 import adafruit_bd3491fs
4 i2c = busio.I2C(board.SCL, board.SDA)
5 bd3491fs = adafruit_bd3491fs.BD3491FS(i2c)
6
7 bd3491fs.active_input = adafruit_bd3491fs.Input.A
8 bd3491fs.input_gain = adafruit_bd3491fs.Level.LEVEL_20DB
9 bd3491fs.channel_1_attenuation = 0
10 bd3491fs.channel_2_attenuation = 0
```

5.2 adafruit_bd3491fs

CircuitPython library for the Rohm BD3491FS Audio Processor

- Author(s): Bryan Siepert

5.2.1 Implementation Notes

Hardware:

- [Adafruit BD3491FS Breakout](#)

Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: <https://github.com/adafruit/circuitpython/releases>
- Adafruit’s Bus Device library: https://github.com/adafruit/Adafruit_CircuitPython_BusDevice
- Adafruit’s Register library: https://github.com/adafruit/Adafruit_CircuitPython_Register

class `adafruit_bd3491fs.BD3491FS` (*i2c_bus*)

Driver for the Rohm BD3491FS audio processor

Parameters `i2c_bus` (*I2C*) – The I2C bus the BD3491FS is connected to.

active_input

The currently selected input. Must be an `Input`

This example sets A1 and A2 to the active input pair. .. code-block:: python
`bd3491fs.active_input = adafruit_bd3491fs.Input.A`

channel_1_attenuation

The attenuation applied to channel 1 of the currently selected input pair in -dB. Maximum is -87dB. To mute set to 255 This example sets the attenuation for input channel 1 to -10dB. .. code-block:: python
`bd3491fs.channel_1_attenuation = 10`”

channel_2_attenuation

The attenuation applied to channel 2 of the currently selected input pair in -dB. Maximum is -87dB. To mute set to 255 This example sets the attenuation for input channel 2 to -10dB. .. code-block:: python
`bd3491fs.channel_2_attenuation = 10`”

input_gain

The gain applied to all inputs equally” This example sets the input gain to 10dB. .. code-block:: python
`bd3491fs.input_gain = adafruit_bd3491fs.Level.10_DB`”

reset ()

Reset the sensor, muting the input, reducing input gain to 0dB, and the output channel attenuation to maximum

class `adafruit_bd3491fs.Input`

Options for `active_input`

Input	Input Pair
<code>Input.A</code>	Inputs A1 and A2
<code>Input.B</code>	Inputs B1 and B2
<code>Input.C</code>	Inputs C1 and C2
<code>Input.D</code>	Inputs D1 and D2
<code>Input.E</code>	Inputs E1 and E2
<code>Input.F</code>	Inputs F1 and F2
<code>Input.SHORT</code>	Short inputs
<code>Input.MUTE</code>	Mute all

class `adafruit_bd3491fs.Level`

Options for `input_gain`

Level	Value
Level.LEVEL_0DB	0dB
Level.LEVEL_2DB	2dB
Level.LEVEL_4DB	4dB
Level.LEVEL_6DB	6dB
Level.LEVEL_8DB	8dB
Level.LEVEL_10DB	10dB
Level.LEVEL_12DB	12dB
Level.LEVEL_14DB	14dB
Level.LEVEL_16DB	16dB
Level.LEVEL_20DB	20dB

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`

a

`adafruit_bd3491fs`, [11](#)

A

`active_input` (*adafruit_bd3491fs.BD3491FS* attribute), 12

`adafruit_bd3491fs` (*module*), 11

B

`BD3491FS` (*class in adafruit_bd3491fs*), 12

C

`channel_1_attenuation`
(*adafruit_bd3491fs.BD3491FS* attribute),
12

`channel_2_attenuation`
(*adafruit_bd3491fs.BD3491FS* attribute),
12

I

`Input` (*class in adafruit_bd3491fs*), 12

`input_gain` (*adafruit_bd3491fs.BD3491FS* attribute),
12

L

`Level` (*class in adafruit_bd3491fs*), 12

R

`reset ()` (*adafruit_bd3491fs.BD3491FS* method), 12