
AdafruitTPA2016 Library Documentation

Release 1.0

Kattni Rembor

Jul 08, 2020

Contents

1	Dependencies	3
1.1	Installing from PyPI	3
2	Usage Example	5
3	Contributing	7
4	Documentation	9
5	Table of Contents	11
5.1	Simple test	11
5.2	adafruit_tpa2016	11
5.2.1	Implementation Notes	11
6	Indices and tables	15
	Python Module Index	17
	Index	19

CircuitPython driver for TPA2016 Class D Amplifier.

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-tpa2016
```

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

```
sudo pip3 install adafruit-circuitpython-tpa2016
```

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-tpa2016
```


CHAPTER 2

Usage Example

```
import adafruit_tpa2016
import busio
import board

i2c = busio.I2C(board.SCL, board.SDA)
tpa = adafruit_tpa2016.TPA2016(i2c)

tpa.fixed_gain = -16
```


CHAPTER 3

Contributing

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

CHAPTER 4

Documentation

For information on building library documentation, please check out [this guide](#).

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/tpa2016_simpletest.py

```
1 import busio
2 import board
3 import adafruit_tpa2016
4
5 i2c = busio.I2C(board.SCL, board.SDA)
6 tpa = adafruit_tpa2016.TPA2016(i2c)
7
8 tpa.fixed_gain = -16
```

5.2 adafruit_tpa2016

CircuitPython driver for TPA2016 Class D Amplifier.

- Author(s): Kattni Rembor

5.2.1 Implementation Notes

Hardware:

- Adafruit TPA2016 - I2C Control AGC

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_tpa2016.TPA2016 (*i2c_bus*)

Driver for the TPA2016 class D amplifier.

Parameters *i2c_bus* (*busio.I2C*) – The I2C bus the TPA2016 is connected to.

amplifier_shutdown

Amplifier shutdown. Amplifier is disabled if `True`. Defaults to `False`. If `True`, device is in software shutdown, e.g. control, bias and oscillator are inactive.

attack_time

The attack time. This is the minimum time between gain decreases. Set to 1 - 63 where 1 = 0.1067ms and the time increases 0.1067ms with each step, for a maximum of 6.722ms. Defaults to 5, or 0.5335ms.

This example sets the attack time to 1, or 0.1067ms.

```
import adafruit_tpa2016
import busio
import board

i2c = busio.I2C(board.SCL, board.SDA)
tpa = adafruit_tpa2016.TPA2016(i2c)

tpa.attack_time = 1
```

compression_ratio

The compression ratio.

Ratio settings are: 1:1, 2:1, 4:1, 8:1. Settings options are: `COMPRESSION_1_1`, `COMPRESSION_2_1`, `COMPRESSION_4_1`, `COMPRESSION_8_1`. Defaults to 4:1.

This example sets the compression ratio to 2:1.

```
import adafruit_tpa2016
import busio
import board

i2c = busio.I2C(board.SCL, board.SDA)
tpa = adafruit_tpa2016.TPA2016(i2c)

tpa.compression_ratio = tpa.COMPRESSION_2_1
```

fixed_gain

The fixed gain of the amplifier in dB. If compression is enabled, fixed gain is adjustable from -28 to 30. If compression is disabled, fixed gain is adjustable from 0 to 30.

The following example sets the fixed gain to -16dB.

```
import adafruit_tpa2016
import busio
import board

i2c = busio.I2C(board.SCL, board.SDA)
tpa = adafruit_tpa2016.TPA2016(i2c)

tpa.fixed_gain = -16
```

hold_time

The hold time. This is the minimum time between attack and release. Set to 0 - 63 where 0 = disabled, and the time increases 0.0137ms with each step, for a maximum of 0.8631ms. Defaults to 0, or disabled.

This example sets hold time to 1, or 0.0137ms.

```
import adafruit_tpa2016
import busio
import board

i2c = busio.I2C(board.SCL, board.SDA)
tpa = adafruit_tpa2016.TPA2016(i2c)

tpa.hold_time = 1
```

max_gain

The max gain in dB. Must be between 18 and 30.

noise_gate_enable

NoiseGate function enable. Enabled by default. Can only be enabled when compression ratio is NOT 1:1. To disable, set to `False`.

noise_gate_threshold

Noise Gate threshold in mV.

Noise gate settings are 1mV, 4mV, 10mV, and 20mV. Settings options are `NOISE_GATE_1`, `NOISE_GATE_4`, `NOISE_GATE_10`, `NOISE_GATE_20`. Only functional when compression ratio is NOT 1:1. Defaults to 4mV.

This example sets the noise gate threshold to 10mV.

```
import adafruit_tpa2016
import busio
import board

i2c = busio.I2C(board.SCL, board.SDA)
tpa = adafruit_tpa2016.TPA2016(i2c)

tpa.noise_gate_threshold = tpa.NOISE_GATE_10
```

output_limiter_disable

Output limiter disable.

Enabled by default when compression ratio is NOT 1:1. Can only be disabled if compression ratio is 1:1. To disable, set to `True`.

output_limiter_level

The output limiter level in dBV. Must be between -6.5 and 9, set in increments of 0.5.

release_time

The release time. This is the minimum time between gain increases. Set to 1 - 63 where 1 = 0.0137ms, and the time increases 0.0137ms with each step, for a maximum of 0.8631ms. Defaults to 11, or 0.1507ms.

This example sets release time to 1, or 0.0137ms.

```
import adafruit_tpa2016
import busio
import board

i2c = busio.I2C(board.SCL, board.SDA)
tpa = adafruit_tpa2016.TPA2016(i2c)

tpa.release_time = 1
```

reset_Fault_l

Over-current event on left channel indicated by returning `True`. Reset by setting to `False`.

reset_fault_r

Over-current event on right channel indicated by returning `True`. Reset by setting to `False`.

reset_thermal

Thermal software shutdown indicated by returning `True`. Reset by setting to `False`.

speaker_enable_l

Enables left speaker. Defaults to enabled. Set to `False` to disable.

speaker_enable_r

Enables right speaker. Defaults to enabled. Set to `False` to disable.

CHAPTER 6

Indices and tables

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

a

`adafruit_tpa2016, 11`

A

adafruit_tpa2016 (module), 11
amplifier_shutdown (adafruit_tpa2016.TPA2016 attribute), 12
attack_time (adafruit_tpa2016.TPA2016 attribute), 12

C

compression_ratio (adafruit_tpa2016.TPA2016 attribute), 12

F

fixed_gain (adafruit_tpa2016.TPA2016 attribute), 12

H

hold_time (adafruit_tpa2016.TPA2016 attribute), 12

M

max_gain (adafruit_tpa2016.TPA2016 attribute), 13

N

noise_gate_enable (adafruit_tpa2016.TPA2016 attribute), 13
noise_gate_threshold (adafruit_tpa2016.TPA2016 attribute), 13

O

output_limiter_disable (adafruit_tpa2016.TPA2016 attribute), 13
output_limiter_level (adafruit_tpa2016.TPA2016 attribute), 13

R

release_time (adafruit_tpa2016.TPA2016 attribute), 13
reset_fault_l (adafruit_tpa2016.TPA2016 attribute), 13
reset_fault_r (adafruit_tpa2016.TPA2016 attribute), 14

reset_thermal (adafruit_tpa2016.TPA2016 attribute), 14

S

speaker_enable_l (adafruit_tpa2016.TPA2016 attribute), 14
speaker_enable_r (adafruit_tpa2016.TPA2016 attribute), 14

T

TPA2016 (class in adafruit_tpa2016), 12