

---

# **Adafruit WAVEFORM Library Documentation**

***Release 1.0***

**Scott Shawcroft**

**Feb 10, 2021**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Installing from PyPI</b>	<b>5</b>
<b>3</b>	<b>Usage Example</b>	<b>7</b>
<b>4</b>	<b>Contributing</b>	<b>9</b>
<b>5</b>	<b>Documentation</b>	<b>11</b>
<b>6</b>	<b>Table of Contents</b>	<b>13</b>
6.1	Simple tests . . . . .	13
6.2	adafruit_waveform.sine . . . . .	14
6.3	adafruit_waveform.square . . . . .	14
<b>7</b>	<b>Indices and tables</b>	<b>15</b>
	<b>Python Module Index</b>	<b>17</b>
	<b>Index</b>	<b>19</b>



This library generates simple waveforms that can be used to generate different type of audio signals.



# CHAPTER 1

---

## Dependencies

---

This driver depends on:

- [Adafruit CircuitPython](#)

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





## CHAPTER 2

---

### 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-waveform
```

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

```
sudo pip3 install adafruit-circuitpython-waveform
```

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-waveform
```



## CHAPTER 3

---

### Usage Example

---

This example generates one wavelength of a 440hz sine wave when played at 16 kilosamples per second:

```
from adafruit_waveform import sine
wave = sine.sine_wave(16000, 440)
```



## CHAPTER 4

---

### Contributing

---

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



## CHAPTER 5

---

### Documentation

---

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





## 6.1 Simple tests

Ensure your device works with these simple tests.

Listing 1: examples/waveform\_sine\_simpletest.py

```
1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  """
5  'sine_demo.py'.
6
7  =====
8  toggles the builtin LED using a sine wave
9  """
10 import time
11 import board
12 import digitalio
13 from adafruit_waveform import sine
14
15 LED = digitalio.DigitalInOut(board.D13)
16 LED.switch_to_output()
17
18 SINE_SAMPLE = sine.sine_wave(150, 50)
19
20 while True:
21     for i in range(len(SINE_SAMPLE)):
22         LED.value = i
23         print(LED.value)
24         time.sleep(0.50)
```

Listing 2: examples/waveform\_square\_simpletest.py

```

1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  """
5  'square_demo.py'.
6
7  =====
8  toggles the builtin LED using a square wave
9  """
10 import time
11 import digitalio
12 import board
13 from adafruit_waveform import square
14
15 LED = digitalio.DigitalInOut(board.D13)
16 LED.switch_to_output()
17 SAMPLE_SQUARE = square.square_wave(2)
18
19 while True:
20     for i in range(len(SAMPLE_SQUARE)):
21         LED.value = i
22         print(LED.value)
23         time.sleep(0.5)

```

## 6.2 adafruit\_waveform.sine

This library generates sine waveforms that can be used to generate sine audio signals.

- Author(s): Scott Shawcroft

`adafruit_waveform.sine.sine_wave(sample_frequency, pitch)`

Generate a single sine wav cycle at the given sampling frequency and pitch.

## 6.3 adafruit\_waveform.square

This library generates squard waveforms that can be used to generate squard audio signals.

- Author(s): Scott Shawcroft, BrentRu

`adafruit_waveform.square.square_wave(sample_length=2)`

Generate a single square wave of sample\_length size

## CHAPTER 7

---

### Indices and tables

---

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



### **a**

`adafruit_waveform.sine`, [14](#)

`adafruit_waveform.square`, [14](#)



## A

`adafruit_waveform.sine` (*module*), [14](#)  
`adafruit_waveform.square` (*module*), [14](#)

## S

`sine_wave()` (*in module adafruit\_waveform.sine*), [14](#)  
`square_wave()` (*in module adafruit\_waveform.square*), [14](#)