Adafruit WAVEFORM Library Documentation

Release 1.0

Scott Shawcroft

May 26, 2019

Contents

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

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 3

Contributing

Contributions are welcome! Please read our Code of Conduct before contributing to help this project stay welcoming.

CHAPTER 4

Building locally

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-waveform --
↪library_location .
```

4.1 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.

CHAPTER 5

Table of Contents

5.1 Simple tests

Ensure your device works with these simple tests.

Listing 1: examples/sine_demo.py

```
"""
'sine_demo.py'.

================================================
toggles the builtin LED using a sine wave
"""
import time
import board
import digitalio
from adafruit_waveform import sine

LED = digitalio.DigitalInOut(board.D13)
LED.switch_to_output()

SINE_SAMPLE = sine.sine_wave(150, 50)

while True:
    for i in range(len(SINE_SAMPLE)):
        LED.value = i
        print(LED.value)
        time.sleep(0.50)
```

Listing 2: examples/square_demo.py

```
"""
'square_demo.py'.

```

(continues on next page)

(continued from previous page)

```
================================================
toggles the builtin LED using a square wave
"""
import time
import digitalio
import board
from adafruit_waveform import square

LED = digitalio.DigitalInOut(board.D13)
LED.switch_to_output()
SAMPLE_SQUARE = square.square_wave(2)

while True:
    for i in range(len(SAMPLE_SQUARE)):
        LED.value = i
        print(LED.value)
        time.sleep(0.5)
```

5.2 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.

5.3 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 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index

A

S