Adafruit RTTTL Library Documentation

Release 1.0

Scott Shawcroft

Dec 21, 2018

Contents

1 Dependencies **3**

2 Usage Example **5**

3 CPX Usage Example **7**

4 Contributing **9**

5 Building locally **11**
5.1 Sphinx documentation 11

6 Table of Contents **13**
6.1 Simple test 13
6.2 `adafruit_rtttl` 14

7 Indices and tables **15**

Python Module Index **17**

This plays RTTTL melodies.

CHAPTER 1

Dependencies

This driver depends on:

- Adafruit CircuitPython
- Adafruit CircuitPython Waveform

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 plays Frosty the Snowman:

```
import board
import adafruit_rtttl

adafruit_rtttl.play(board.A0, "Snowman:d=8,o=5,b=200:2g,4e.,f,4g,2c6,b,c6,4d6,4c6,4b,
↪a,2g.,b,c6,4d6,4c6,4b,a,a,g,4c6,4e.,g,a,4g,4f,4e,4d,2c.,4c,4a,4a,4c6,4c6,4b,4a,4g,
↪4e,4f,4a,4g,4f,2e.,4e,4d,4d,4g,4g,4b,4b,4d6,d6,b,4d6,4c6,4b,4a,4g,4p,2g")
```

CHAPTER 3

CPX Usage Example

This plays Frosty the Snowman on a Circuit Playground Express (we must enable onboard speaker):

```
import board
from digitalio import DigitalInOut, Direction
import adafruit_rtttl
spkrenable = DigitalInOut(board.SPEAKER_ENABLE)
spkrenable.direction = Direction.OUTPUT
spkrenable.value = True

adafruit_rtttl.play(board.A0, "Snowman:d=8,o=5,b=200:2g,4e.,f,4g,2c6,b,c6,4d6,4c6,4b,
↪a,2g.,b,c6,4d6,4c6,4b,a,a,g,4c6,4e.,g,a,4g,4f,4e,4d,2c.,4c,4a,4a,4c6,4c6,4b,4a,4g,
↪4e,4f,4a,4g,4f,2e.,4e,4d,4d,4g,4g,4b,4b,4d6,d6,b,4d6,4c6,4b,4a,4g,4p,2g")
```

CHAPTER 4

Contributing

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

CHAPTER 5

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-rtttl --library_
↪location .
```

5.1 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 6

Table of Contents

6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/rtttl_simpletest.py

```
# The MIT License (MIT)
#
# Copyright (c) 2017 Scott Shawcroft for Adafruit Industries
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to deal
# in the Software without restriction, including without limitation the rights
# to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in
# all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
# THE SOFTWARE.

# This is tested on the CircuitPlayground Express

import digitalio
import board
import adafruit_rtttl
```

(continues on next page)

(continued from previous page)

```

enable = digitalio.DigitalInOut(board.SPEAKER_ENABLE)
enable.switch_to_output(value=True)

adafruit_rtttl.play(board.SPEAKER, "itchy:d=8,o=6,b=160:c,a5,4p,c,a,4p,c,a5,c,a5," + \
    "c,a,4p,p,c,d,e,p,e,f,g,4p,d,c,4d,f,4a#,4a,2c7")
adafruit_rtttl.play(board.SPEAKER, "Phantom:d=4,o=5,b=140:c,f,c,d#.,8c#,2c#,a#4," + \
    "d#,8a#4,2c,c,f,c,d#.,8c#,2c#,a#4,d#.,8a#4,2c,p,c,f,g#,c.6,8a#,2a#,a#,d#.6,8a#,"␣
↪+ \
    "2c6,p,c6,2f.6,8d#6,8c#6,8c6,8a#,8g#,8g,8f,2e,c#,c#.,8c,2c")
```

6.2 adafruit_rtttl

Play notes to a digialio pin using ring tone text transfer language (rtttl).

- Author(s): Scott Shawcroft

`adafruit_rtttl.play`(*pin*, *rtttl*, *octave=None*, *duration=None*, *tempo=None*)

Play notes to a digialio pin using ring tone text transfer language (rtttl). :param ~digitalio.DigitalInOut pin: the speaker pin :param rtttl: string containing rtttl :param int octave: represents octave number (default 6 starts at middle c) :param int duration: length of notes (default 4 quarter note) :param int tempo: how fast (default 63 beats per minute)

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

adafruit_rtttl, 14

Index

A

adafruit_rtttl (module), 14

P

play() (in module adafruit_rtttl), 14