

---

# **ptttl Documentation**

***Release v1.0.0***

**Erik Nyquist**

**Aug 29, 2023**



**CONTENTS:**

<b>1</b>	<b>ptttl</b>	<b>1</b>
1.1	ptttl package . . . . .	1
<b>2</b>	<b>Indices and tables</b>	<b>5</b>
	<b>Python Module Index</b>	<b>7</b>
	<b>Index</b>	<b>9</b>



## 1.1 ptttl package

### 1.1.1 Submodules

### 1.1.2 ptttl.audio module

`ptttl.audio.ptttl_to_mp3(ptttl_data, mp3_filename, amplitude=0.5, wavetype=0)`

Convert a PTTTLData object to audio data and write it to an .mp3 file (requires the LAME audio mp3 encoder to be installed and in your system path).

**Parameters**

- **ptttl\_data** (*str*) – PTTTL/RTTTL source text
- **mp3\_filename** (*str*) – Filename for output .mp3 file
- **amplitude** (*float*) – Output signal amplitude, between 0.0 and 1.0.
- **wavetype** (*int*) – Waveform type for output signal. Must be one of `tones.SINE_WAVE`, `tones.SQUARE_WAVE`, `tones.TRIANGLE_WAVE`, or `tones.SAWTOOTH_WAVE`.

`ptttl.audio.ptttl_to_samples(ptttl_data, amplitude=0.5, wavetype=0)`

Convert a PTTTLData object to a list of audio samples.

**Parameters**

- **ptttl\_data** (`PTTTLData`) – PTTTL/RTTTL source text
- **amplitude** (*float*) – Output signal amplitude, between 0.0 and 1.0.
- **wavetype** (*int*) – Waveform type for output signal. Must be one of `tones.SINE_WAVE`, `tones.SQUARE_WAVE`, `tones.TRIANGLE_WAVE`, or `tones.SAWTOOTH_WAVE`.

**Returns**

list of audio samples

**Return type**

`tones.tone.Samples`

`ptttl.audio.ptttl_to_wav(ptttl_data, wav_filename, amplitude=0.5, wavetype=0)`

Convert a PTTTLData object to audio data and write it to a .wav file.

**Parameters**

- **ptttl\_data** (*str*) – PTTTL/RTTTL source text
- **wav\_filename** (*str*) – Filename for output .wav file

- **amplitude** (*float*) – Output signal amplitude, between 0.0 and 1.0.
- **wavetype** (*int*) – Waveform type for output signal. Must be one of `tones.SINE_WAVE`, `tones.SQUARE_WAVE`, `tones.TRIANGLE_WAVE`, or `tones.SAWTOOTH_WAVE`.

`ptttl.audio.ptttl_to_wav_samples(ptttl_data, amplitude=0.5, wavetype=0)`

Convert a PTTTLData object to a list of audio samples, packed into string and ready for writing to .wav files.

#### Parameters

- **ptttl\_data** (*str*) – PTTTL/RTTTL source text
- **amplitude** (*float*) – Output signal amplitude, between 0.0 and 1.0.
- **wavetype** (*int*) – Waveform type for output signal. Must be one of `tones.SINE_WAVE`, `tones.SQUARE_WAVE`, `tones.TRIANGLE_WAVE`, or `tones.SAWTOOTH_WAVE`.

#### Returns

list of audio samples

#### Return type

str

### 1.1.3 ptttl.parser module

**class** `ptttl.parser.PTTTLData(bpm=123, default_octave=4, default_duration=8, default_vibrato_freq=7.0, default_vibrato_var=20.0)`

Bases: `object`

Represents song data extracted from a PTTTL/RTTTL file. May contain multiple tracks, where each track is a list of PTTTLNote objects.

#### Variables

- **tracks** (`[ [PTTTLNote] ]`) – List of tracks. Each track is a list of PTTTLNote objects.
- **bpm** (*float*) – playback speed in BPM (beats per minute).
- **default\_octave** (*int*) – Default octave to use when none is specified
- **default\_duration** (*int*) – Default note duration to use when none is specified
- **default\_vibrato\_freq** (*float*) – Default vibrato frequency when none is specified, in Hz
- **default\_vibrato\_var** (*float*) – Default vibrato variance when none is specified, in Hz

**add\_track**(*notes*)

**class** `ptttl.parser.PTTTLNote(pitch, duration, vfreq=None, vvar=None)`

Bases: `object`

Represents a single musical note, with a pitch and duration

#### Variables

- **pitch** (*float*) – Note pitch in Hz
- **duration** (*float*) – Note duration in seconds
- **vfreq** (*float*) – Vibrato frequency in Hz
- **vvar** (*float*) – Vibrato variance from main pitch in Hz

**has\_vibrato()**

Returns True if vibrato frequency and variance are non-zero

**Returns**

True if vibrato is non-zero

**Return type**

bool

**class** ptttl.parser.PTTTLParser

Bases: object

Converts PTTTL/RTTTL source text to a PTTTLData object.

**parse**(*ptttl\_string*)

Extracts song data from ptttl/rtttl source data.

**Parameters**

**ptttl\_string** (*str*) – PTTTL/RTTTL source text.

**Returns**

Song data extracted from source text.

**Return type**

*PTTTLData*

**exception** ptttl.parser.PTTTLSyntaxError

Bases: Exception

Raised by PTTTLParser when ptttl data is malformed and cannot be parsed

**exception** ptttl.parser.PTTTLValueError

Bases: Exception

Raised by PTTTLParser when ptttl data parsing completes, but an invalid configuration value or note value was seen.

### 1.1.4 Module contents





## INDICES AND TABLES

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



## PYTHON MODULE INDEX

### p

`ptttl`, [3](#)

`ptttl.audio`, [1](#)

`ptttl.parser`, [2](#)



## INDEX

### A

`add_track()` (*ptttl.parser.PTTTLData method*), 2

### H

`has_vibrato()` (*ptttl.parser.PTTTLNote method*), 2

### M

module

`ptttl`, 3

`ptttl.audio`, 1

`ptttl.parser`, 2

### P

`parse()` (*ptttl.parser.PTTTLParser method*), 3

`ptttl`

    module, 3

`ptttl.audio`

    module, 1

`ptttl.parser`

    module, 2

`ptttl_to_mp3()` (*in module ptttl.audio*), 1

`ptttl_to_samples()` (*in module ptttl.audio*), 1

`ptttl_to_wav()` (*in module ptttl.audio*), 1

`ptttl_to_wav_samples()` (*in module ptttl.audio*), 2

`PTTTLData` (*class in ptttl.parser*), 2

`PTTTLNote` (*class in ptttl.parser*), 2

`PTTTLParser` (*class in ptttl.parser*), 3

`PTTTLSyntaxError`, 3

`PTTTLValueError`, 3