

cedargrove_punkconsole

A CircuitPython-based Atari Punk Console emulation helper class based on the "Stepped Tone Generator" circuit, "Engineer's Mini-Notebook: 555 Circuits", Forrest M. Mims III (1984).

The CedarGrove Punk Console emulates an astable square-wave oscillator and synchronized non-retriggerable one-shot monostable multivibrator to create the classic stepped-tone generator sound of the Atari Punk Console. As with the original circuit, the oscillator frequency and one-shot pulse width are the input parameters. Instantiation of the Punk Console class will start the output waveform based on the input parameters and enable the output signal if ``mute=False``. If no input parameters are provided, the output signal will be disabled regardless of the mute value. Once instantiated, the class is controlled by the ``frequency``, ``pulse_width_ms``, and ``mute`` properties.

This version of the emulator works only with PWM-capable output pins.

Depending on the timer and PWM capabilities of the host MPU board, the emulator can easily outperform the original analog circuit. Oscillator frequency is only limited by the MPU's PWM duty cycle and frequency parameters, which may create output signals well above the practical audio hearing range. Therefore, it is recommended that one-shot pulse width input be limited to the range of 0.5ms and 5ms and that the oscillator frequency input range be between 3Hz and 3kHz -- although experimentation is encouraged!

The repo contains three examples, a simple single-channel console, an annoying stereo noisemaker, and a note table driven sequencer. For the first two examples, input is provided by potentiometers attached to two analog input pins. The sequencer is controlled by an internal list of notes that select the oscillator frequency; pulse width is potentiometer controlled.

Minimum and maximum input ranges (may be further limited by the MPU):

pulse_width: 0.05ms to 5000ms
frequency: 1Hz to >4MHz

Practical input ranges for audio (empirically determined):

pulse_width: 0.5ms to 5ms
frequency: 3Hz to 3kHz

The CedarGrove Punk Console algorithm uses PWM frequency and duty cycle parameters to build the output waveform. The PWM output frequency is an integer multiple of the oscillator frequency input compared to the one-shot pulse width input:

```
`pwm_freq = freq_in / (int((pulse_width) * freq_in) + 1)`
```

The PWM output duty cycle is calculated after the PWM output frequency is determined. The PWM output duty cycle is the ratio of the one-shot pulse width and the wavelength of the PWM output frequency:

```
`pwm_duty_cycle = pulse_width * pwm_freq`
```

- Author(s): JG for Cedar Grove Maker Studios

Implementation Notes

Hardware:

Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: <https://circuitpython.org/downloads>

```
class cedargrove_punkconsole.PunkConsole(*, pin, frequency=1, pulse_width_ms=0, mute=True)
```

Class representing a CircuitPython helper class to emulate the Atari Punk Console.

- Parameters:**
- **pin** – The PWM-capable output pin. No default.
 - **frequency** – The oscillator frequency integer value setting in Hertz. Defaults to 1 Hertz to squelch output noise.
 - **pulse_width_ms** – The non-retriggerable one-shot monostable multivibrator delay (pulse_width) floating point value setting in milliseconds. Defaults to zero milliseconds to squelch output noise.
 - **mute** – The boolean output signal state. `True` to mute; `False` to unmute. Defaults to muted.

pin

A class getter and setter property.

frequency

A class getter and setter property.

pulse_width_ms

A class getter and setter property.

mute

A class getter and setter property.