Electronics for IoT

Electronic Filters

Bernhard E. Boser
University of California, Berkeley
boser@eecs.berkeley.edu
Labs 6 & 7

- Capacitors, phasors in the real world
- Audio synthesizer (Lab 6)
- Capacitive touch sensor (Lab 7)
Fourier Series
Square Wave

![Square Wave Diagram](image)

The upper graph shows sinusoids, and the lower graph shows the sum of sinusoids over time.
Active RC Low-Pass Filter
Low-Pass Characteristic

![Graph showing the low-pass characteristic. The graph plots the magnitude in dB against frequency in Hz. The blue line represents the calculated characteristic, while the orange line represents the rounded characteristic. The x-axis is labeled "Frequency [Hz]" ranging from $10^2$ to $10^4$, and the y-axis is labeled "Magnitude [dB]" ranging from 0 to -40.]
Time Domain
Setup

[Diagram showing the connection of MCU with PWM outputs to a Low-pass Filter, followed by a Summing Amplifier and then an Audio Amplifier with a speaker image.]
Hook it up …
Layout
CAD
Printed Circuit Board

IoT49 Synthesizer
UC Berkeley EECS

B. E. Boser

IoT49: Filters
Solder ... listen!
PWM

50% duty cycle

75% duty cycle

25% duty cycle
Docs and Example

https://github.com/loboris/MicroPython_ESP32_psRAM_LoBo/wiki/pwm

Create the pwm instance object

```python
pwm = machine.PWM(pin [, freq=f] [, duty=d] [, timer=tm])
```

<table>
<thead>
<tr>
<th>Arg</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pin</td>
<td>esp32 GPIO number to be used as pwm output can be given as integer value or machine.Pin object</td>
</tr>
<tr>
<td>freq</td>
<td>optional, default 5 kHz; pwm frequency in Hz (1 - 40000000)</td>
</tr>
<tr>
<td>duty</td>
<td>optional, default 50% kHz; pwm duty cycle in % (0 - 100)</td>
</tr>
<tr>
<td>timer</td>
<td>optional, default 0; pwm timer (0 - 3)</td>
</tr>
</tbody>
</table>

PWM channel is selected automatically from 8 available pwm channels.

```python
pwm.init([ freq=f] [, duty=d] [, timer=tm])
```

Reinitialize the pwm channel

<table>
<thead>
<tr>
<th>Arg</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>freq</td>
<td>optional, if not given, the frequency is not changed</td>
</tr>
<tr>
<td>duty</td>
<td>optional, if not given, the duty cycle is not changed</td>
</tr>
<tr>
<td>timer</td>
<td>optional, if not given, the pwm timer is not changed</td>
</tr>
</tbody>
</table>

Changing the frequency or timer will affect all pwm channels using the same timer.

Maybe it works
If not → see documentation!!!