

Electronics for IoT

Electronic Filters

Bernhard E. Boser

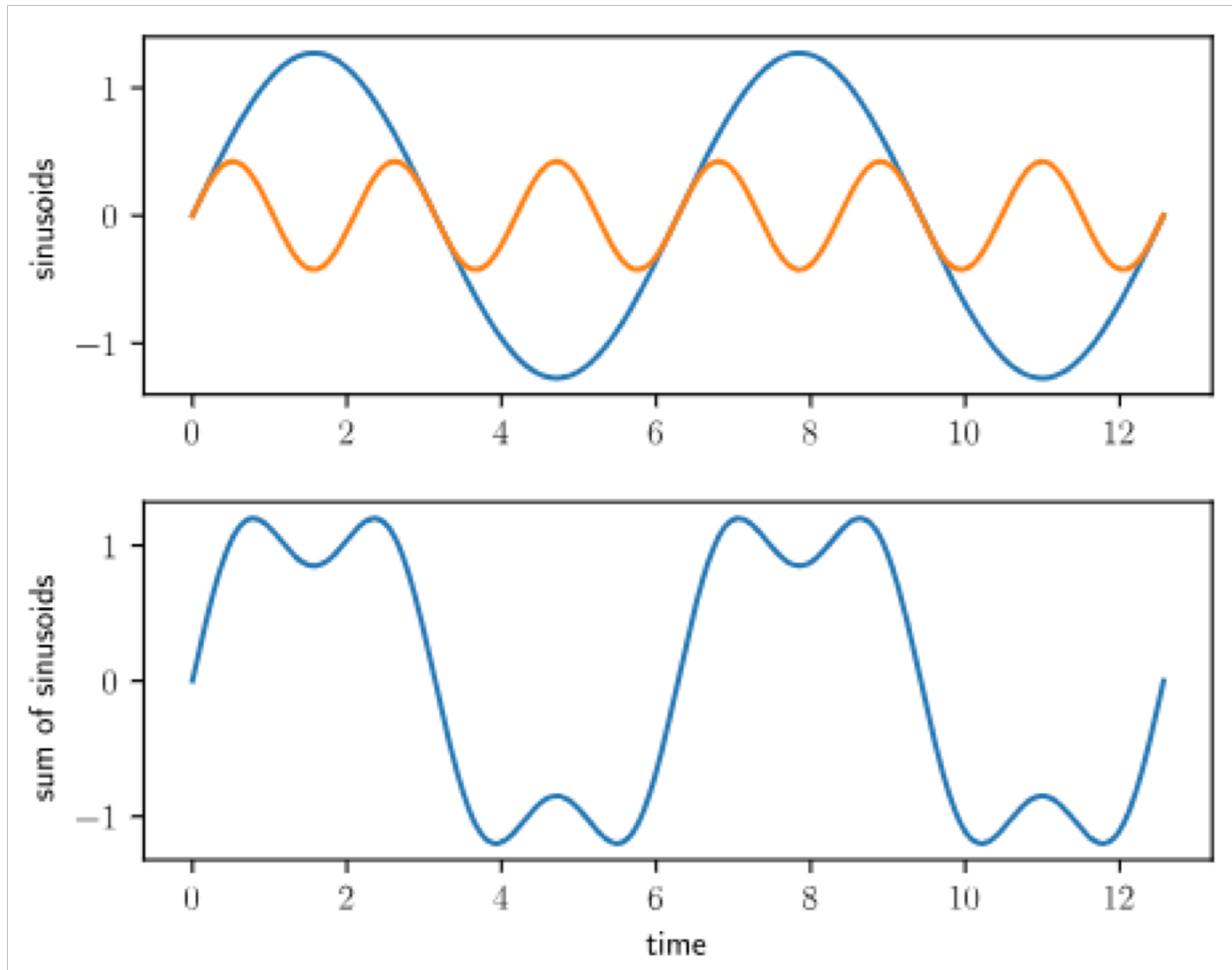
University of California, Berkeley

boser@eecs.berkeley.edu

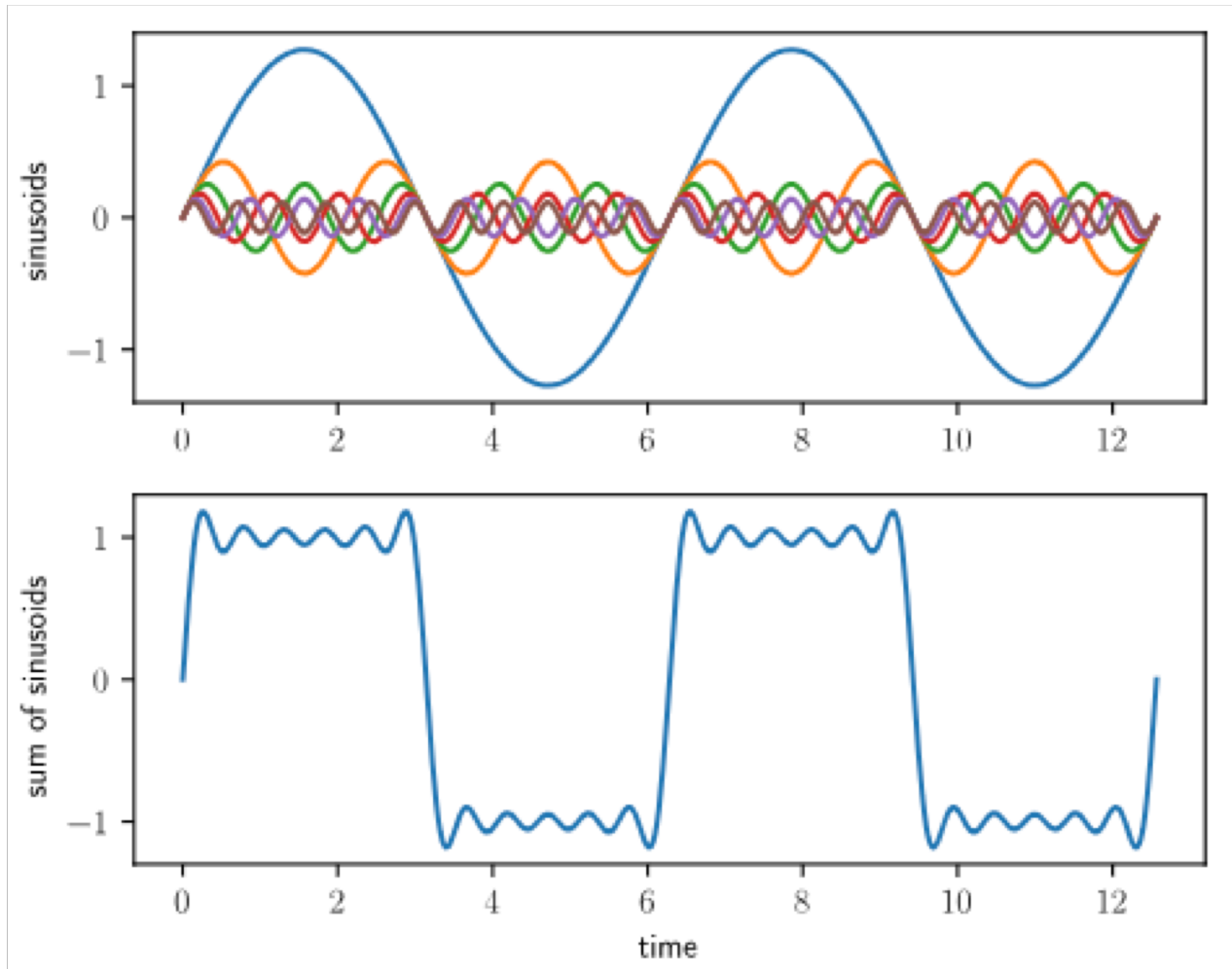
Labs 6 & 7

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

Fourier Series

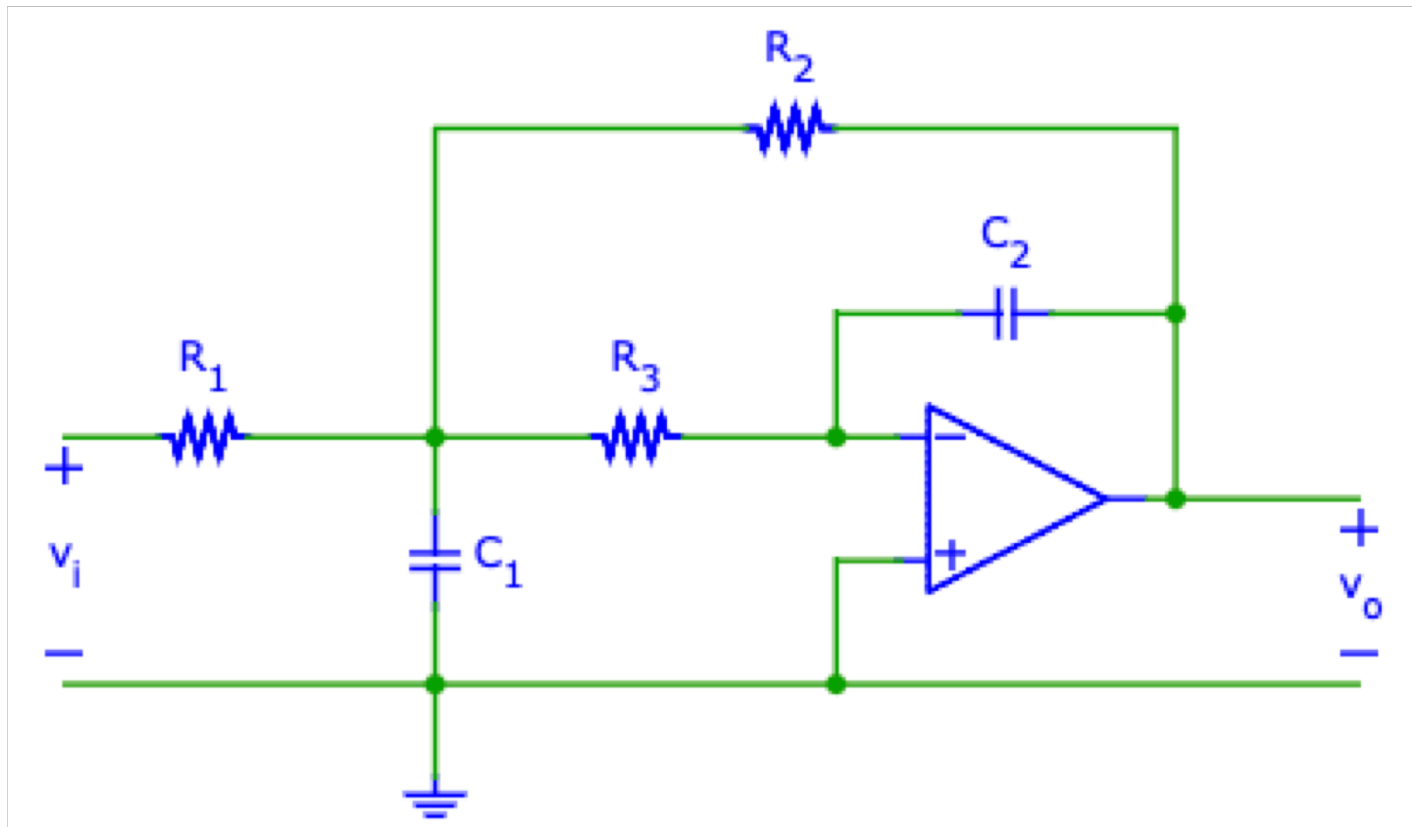


Square Wave

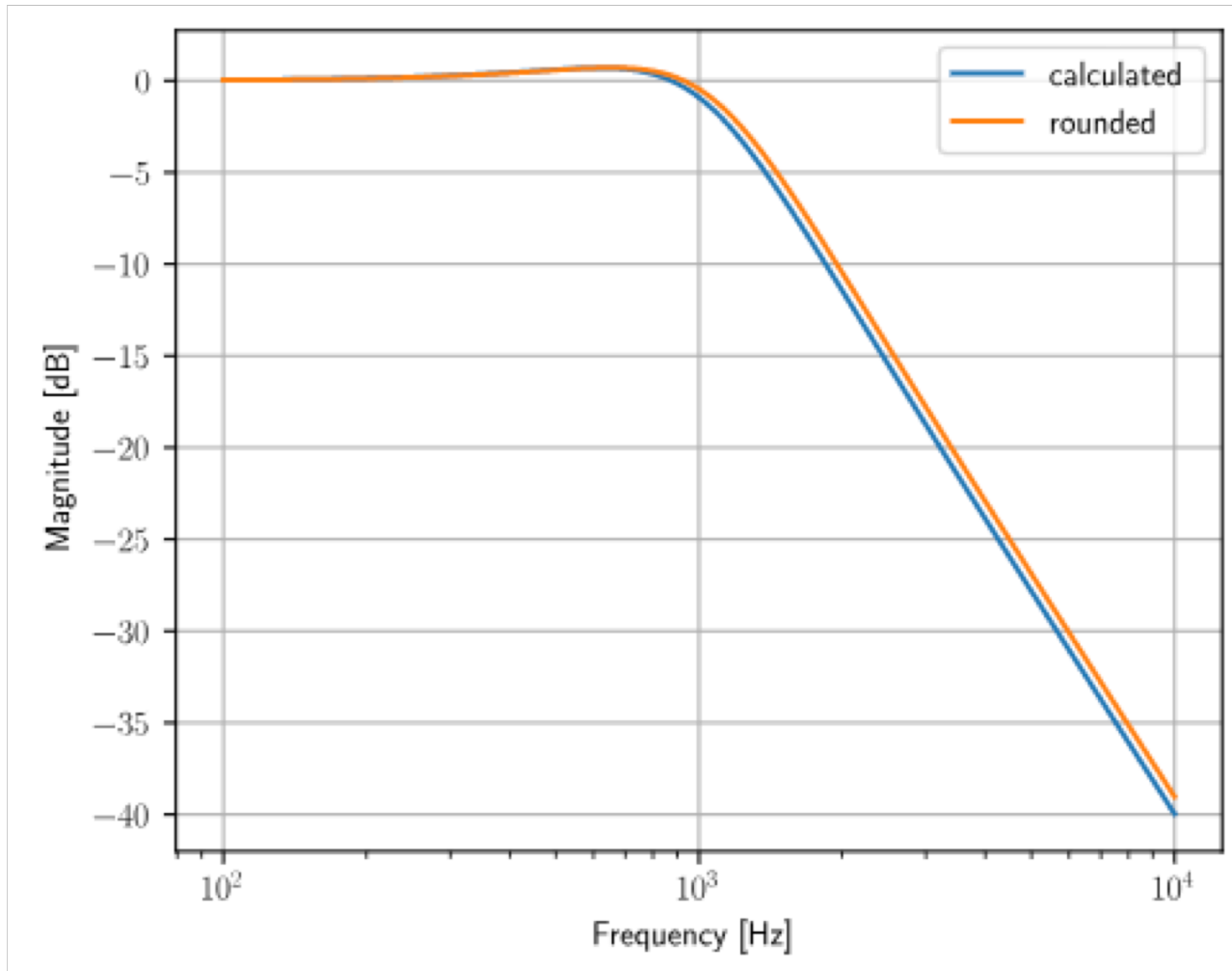


Audio Synthesizer

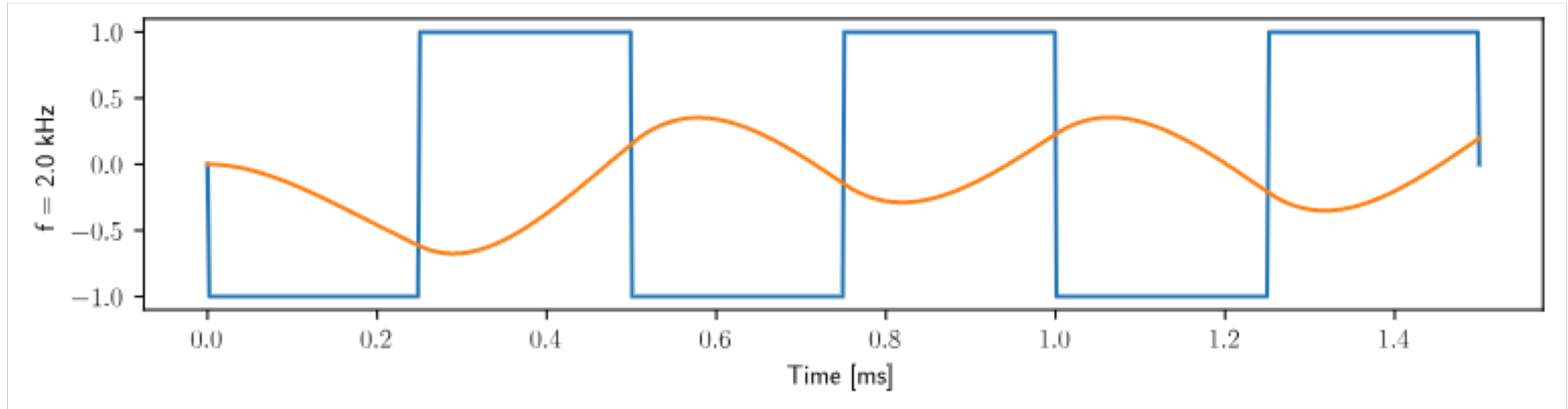
Active RC Low-Pass Filter



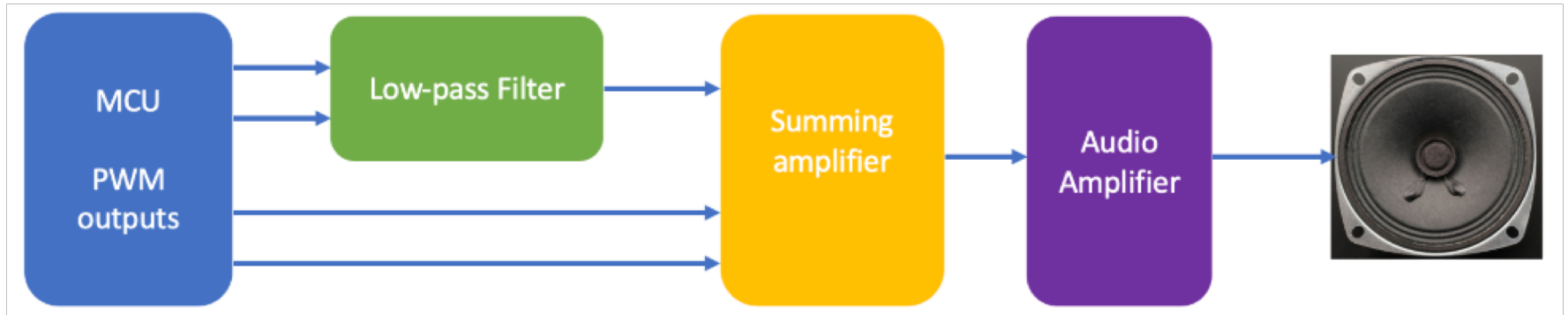
Low-Pass Characteristic



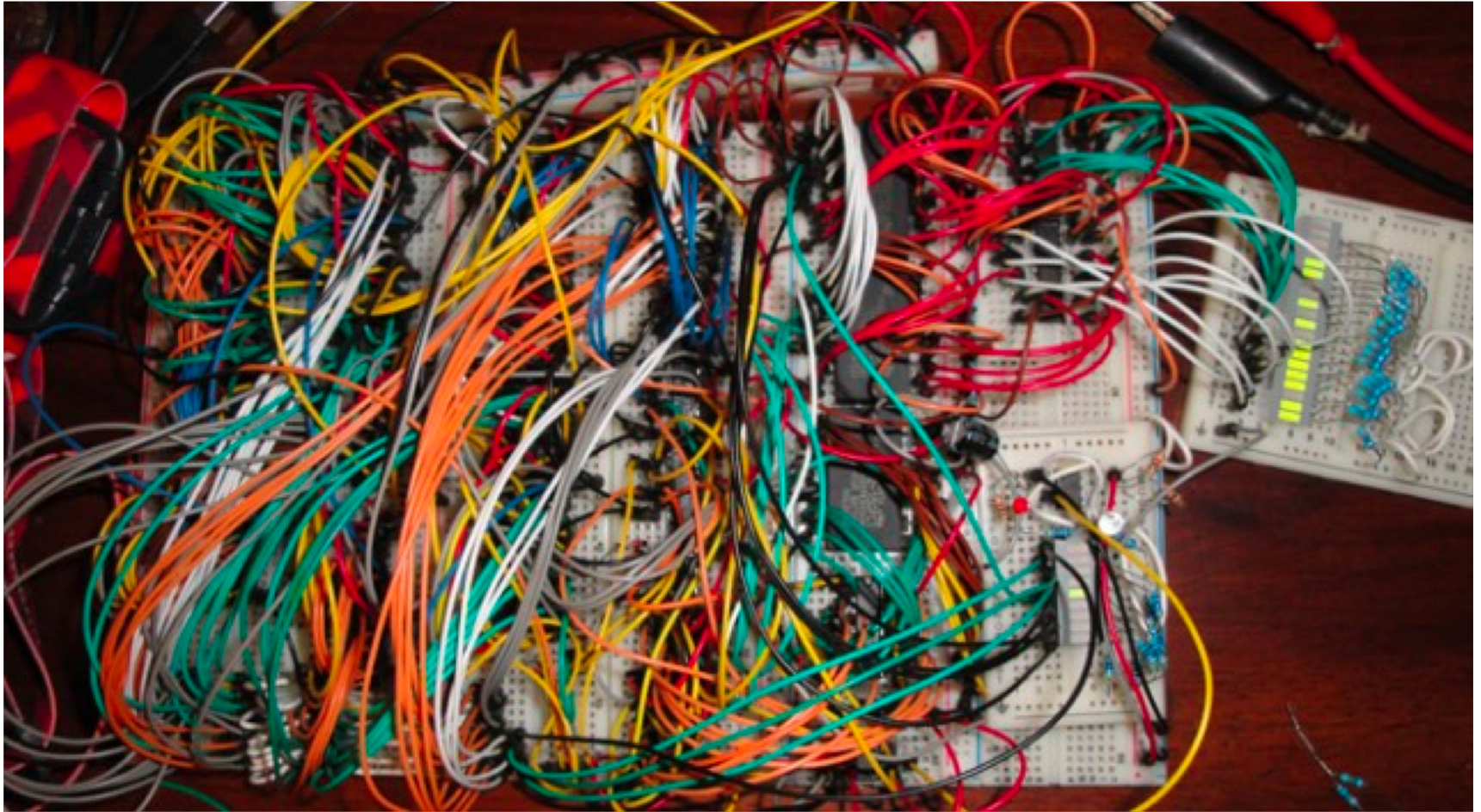
Time Domain



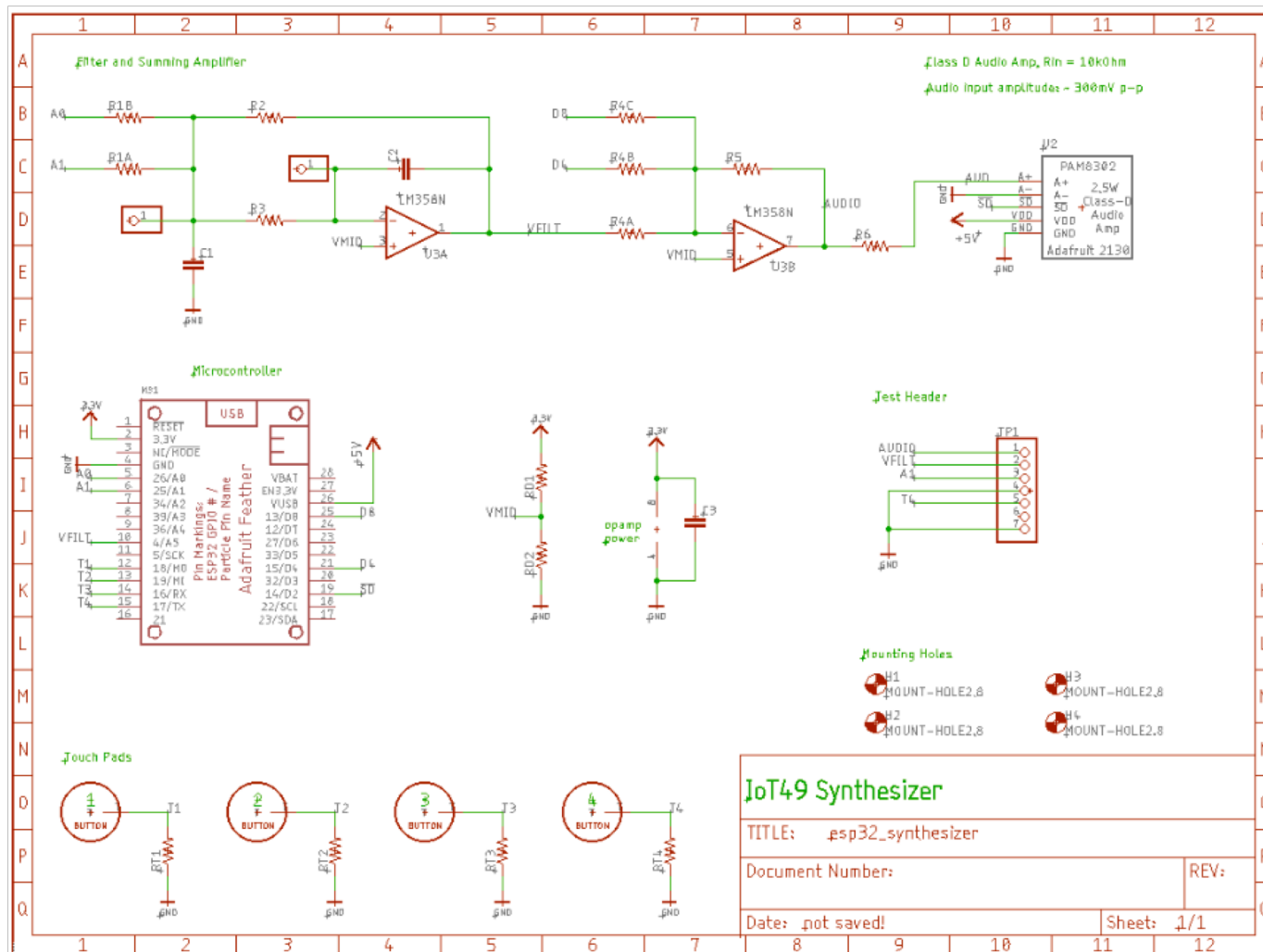
Setup



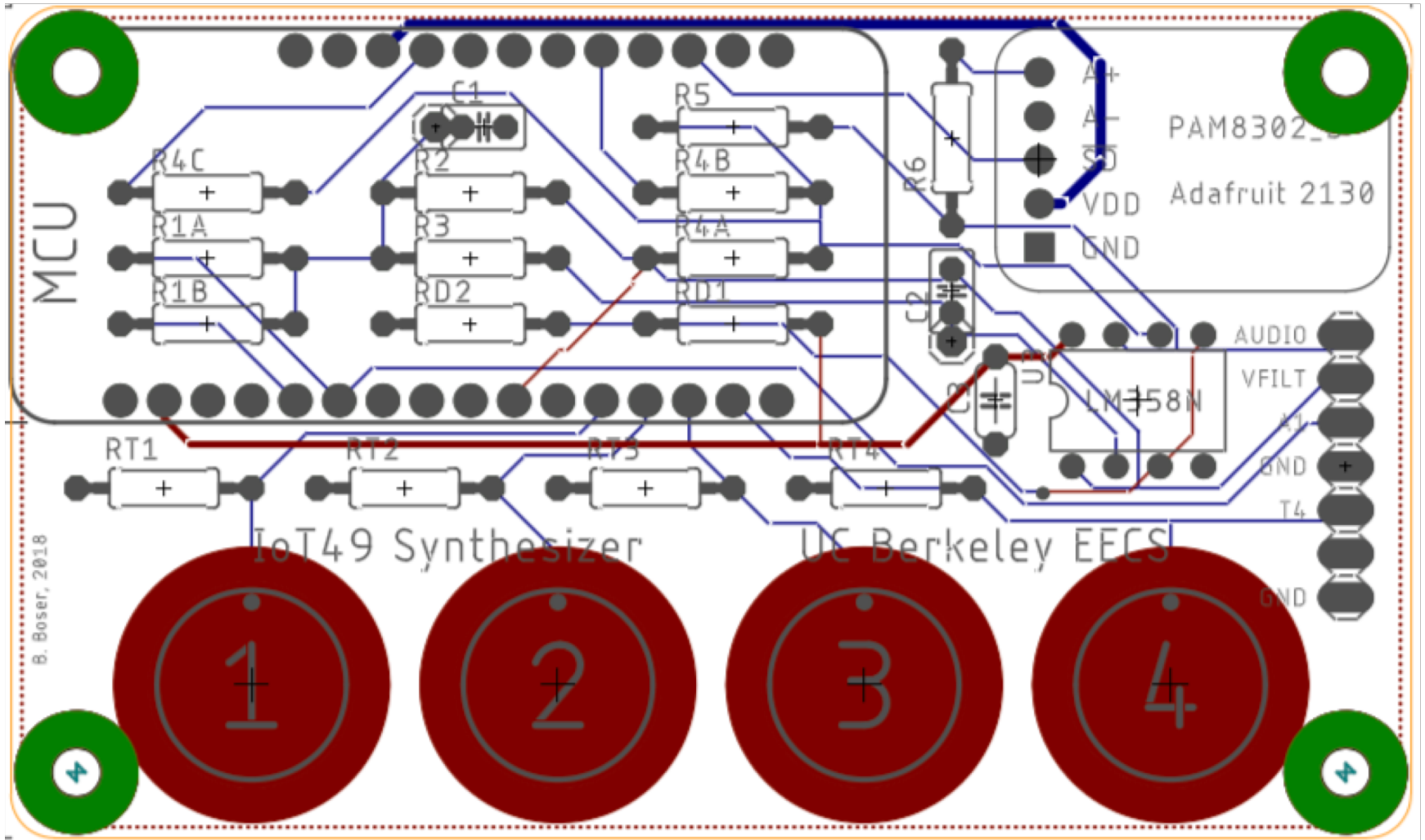
Hook it up ...



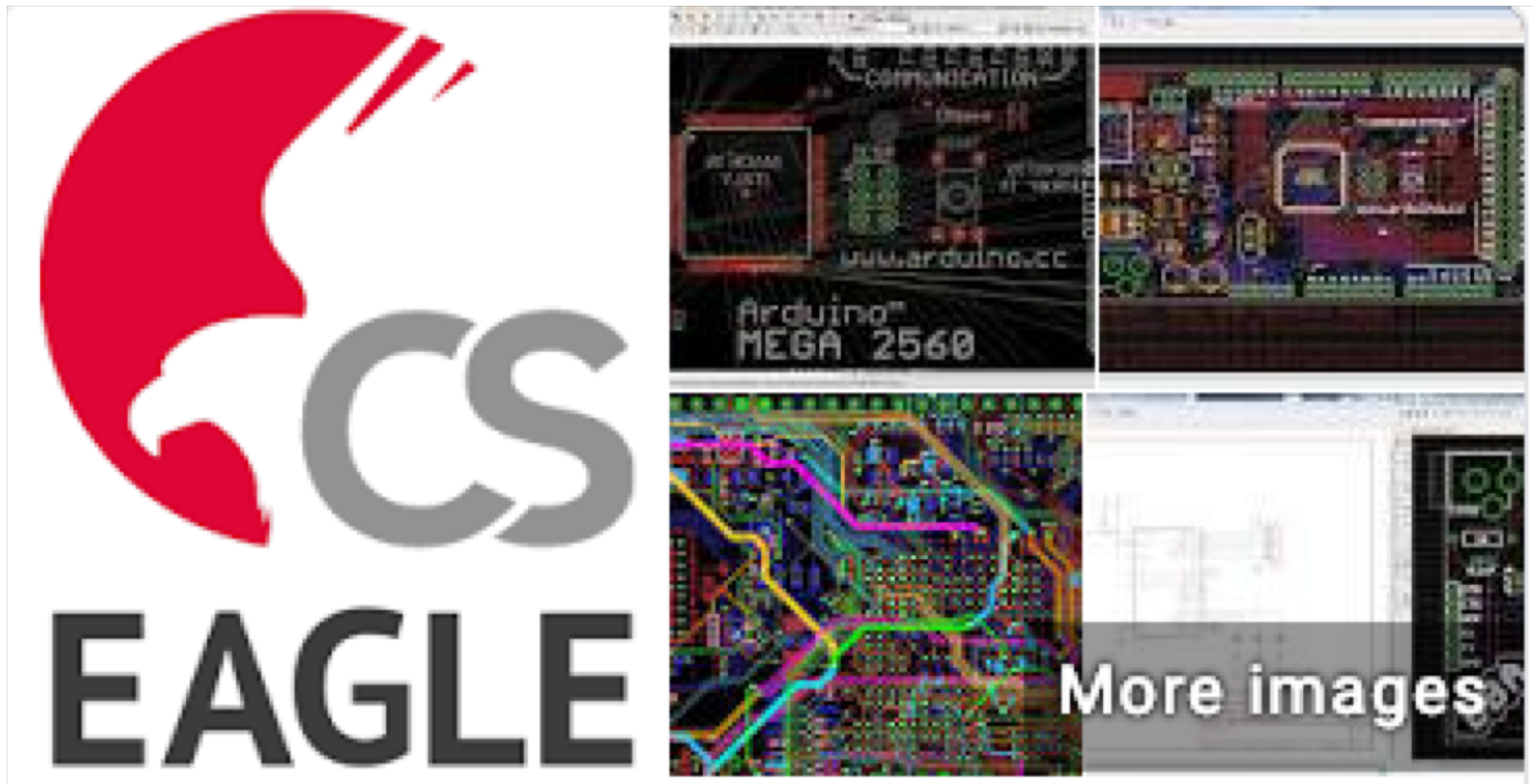
Circuit Diagram



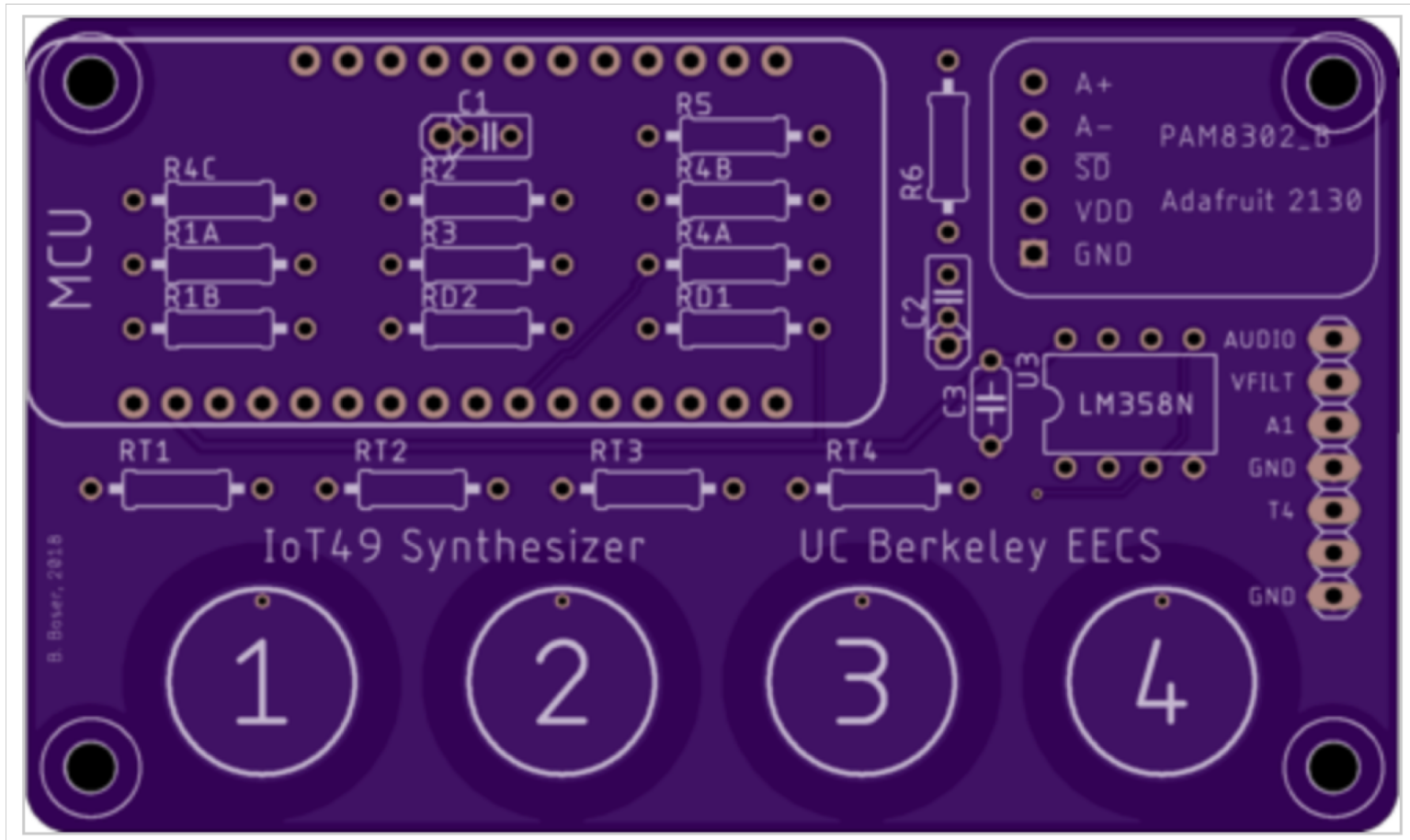
Layout



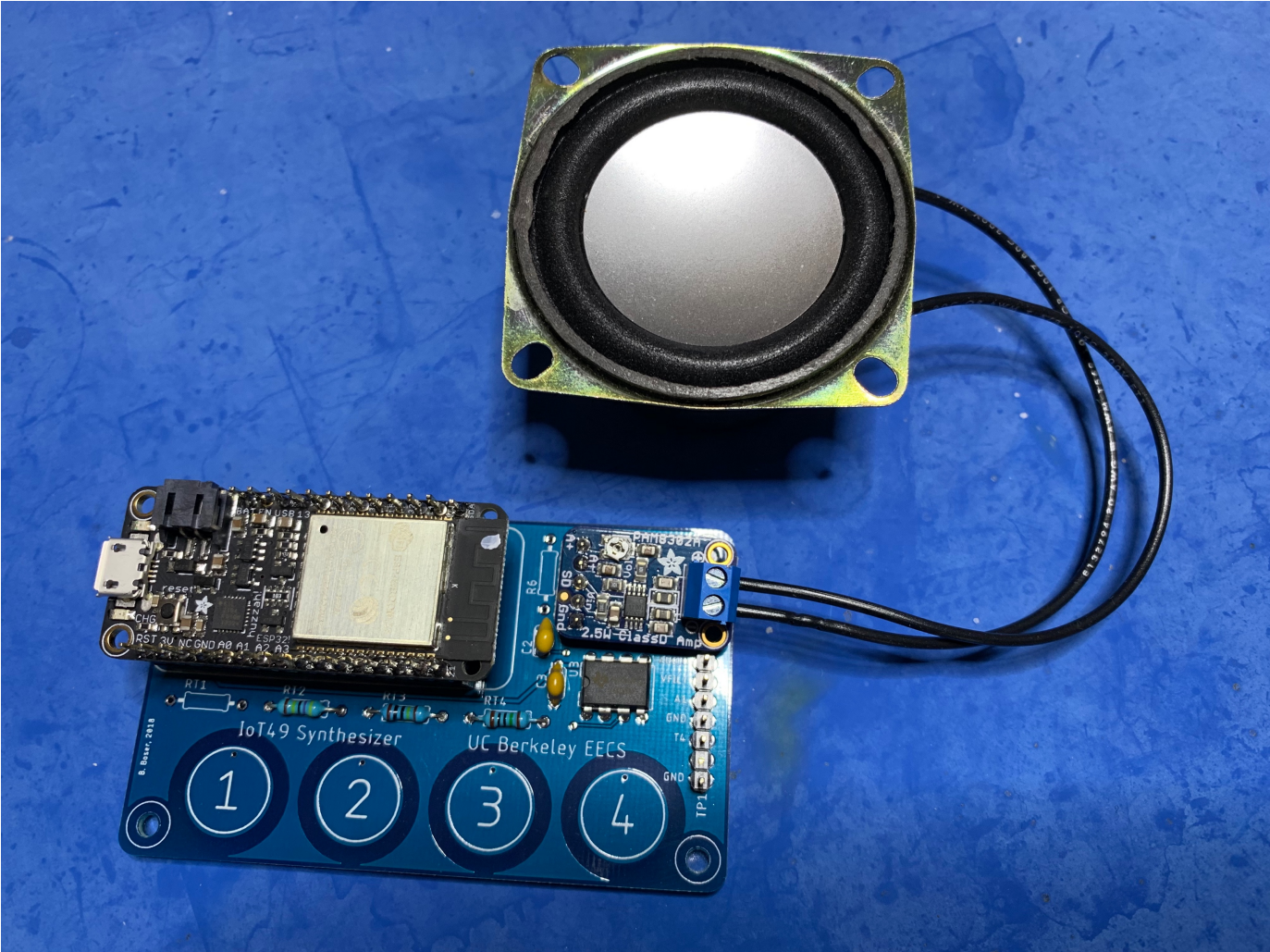
CAD



Printed Circuit Board



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

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

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

PWM channel is selected automatically from 8 available pwm channels.

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

Reinitialize the pwm channel

Arg	Description
freq	optional , if not given, the frequency is not changed
duty	optional , if not given, the duty cycle is not changed
timer	optional , if not given, the pwm timer is not changed

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

```
from machine import Pin, PWM
from board import A0
```

```
p = PWM(Pin(A0), freq=20e3, duty=30, timer=2)
```

```
p.freq(30e3) ①
p.duty(80)    ②
```

Maybe it works

If not → **see documentation!!!**