Animated Skeleton in a Coffin for Halloween

https://github.com/jmoonware/jaws

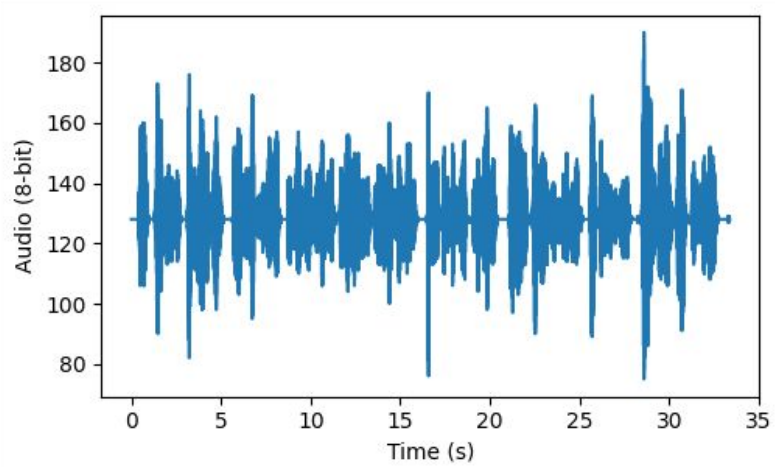
The basic issue: How do you animate the jaw motion of a plastic skull, given an audio input?

Possible solutions:

(1) Feed audio into low-pass circuit and use amplitude to drive motor
- A simple solution that doesn't work very well… X

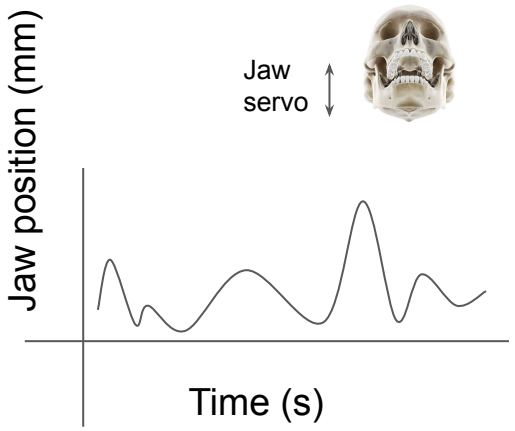
(2) Get a pile of videos, detect jaw motion, and feed results (with audio) into a neural net
- Allows arbitrary audio input, but is a lot of work! X

(3) "In between" solution:
- Record video of desired speech, detect jaw motion, and use synchronized jaw motion and audio as playback on a microprocessor ⇐



Audio signal

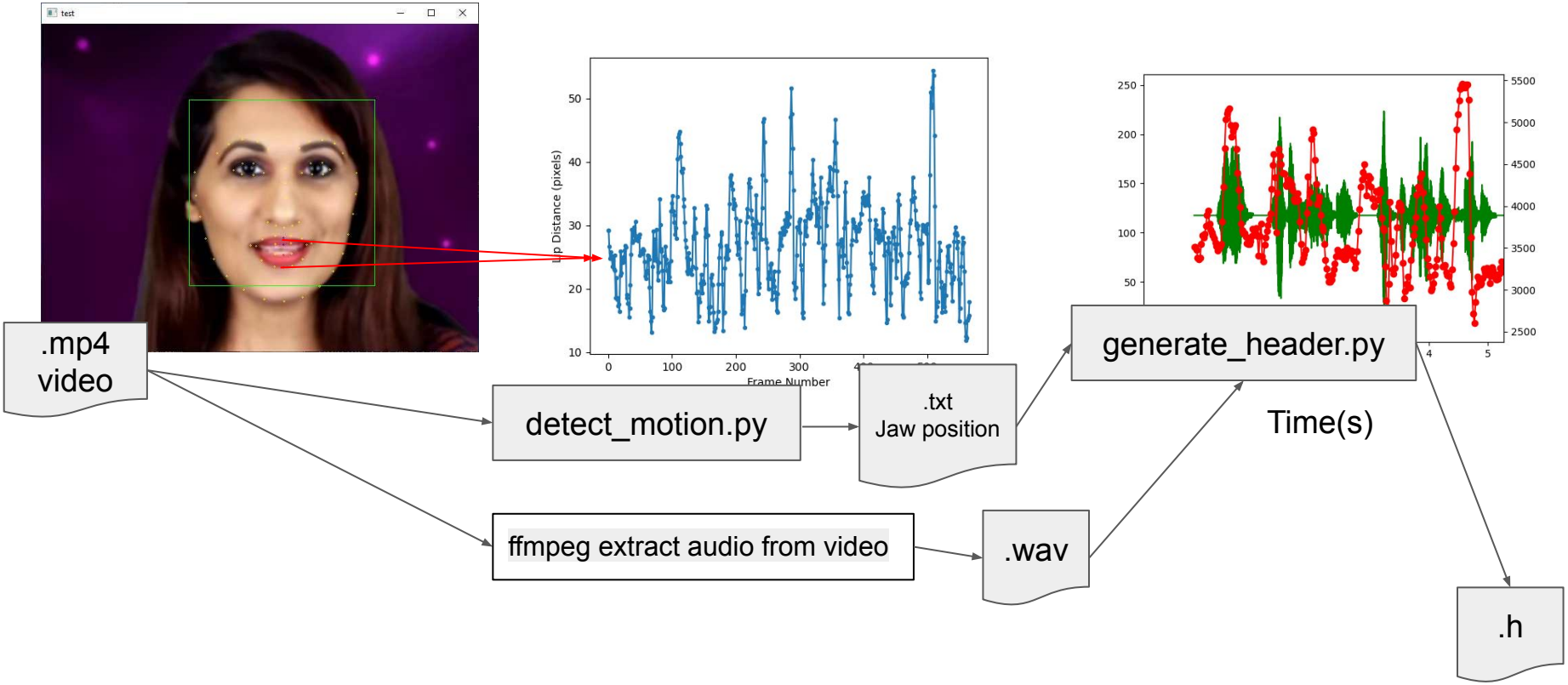
?



Jaw motion

Also, want to turn other things on/off, open/close lids, etc. so some DIO is needed

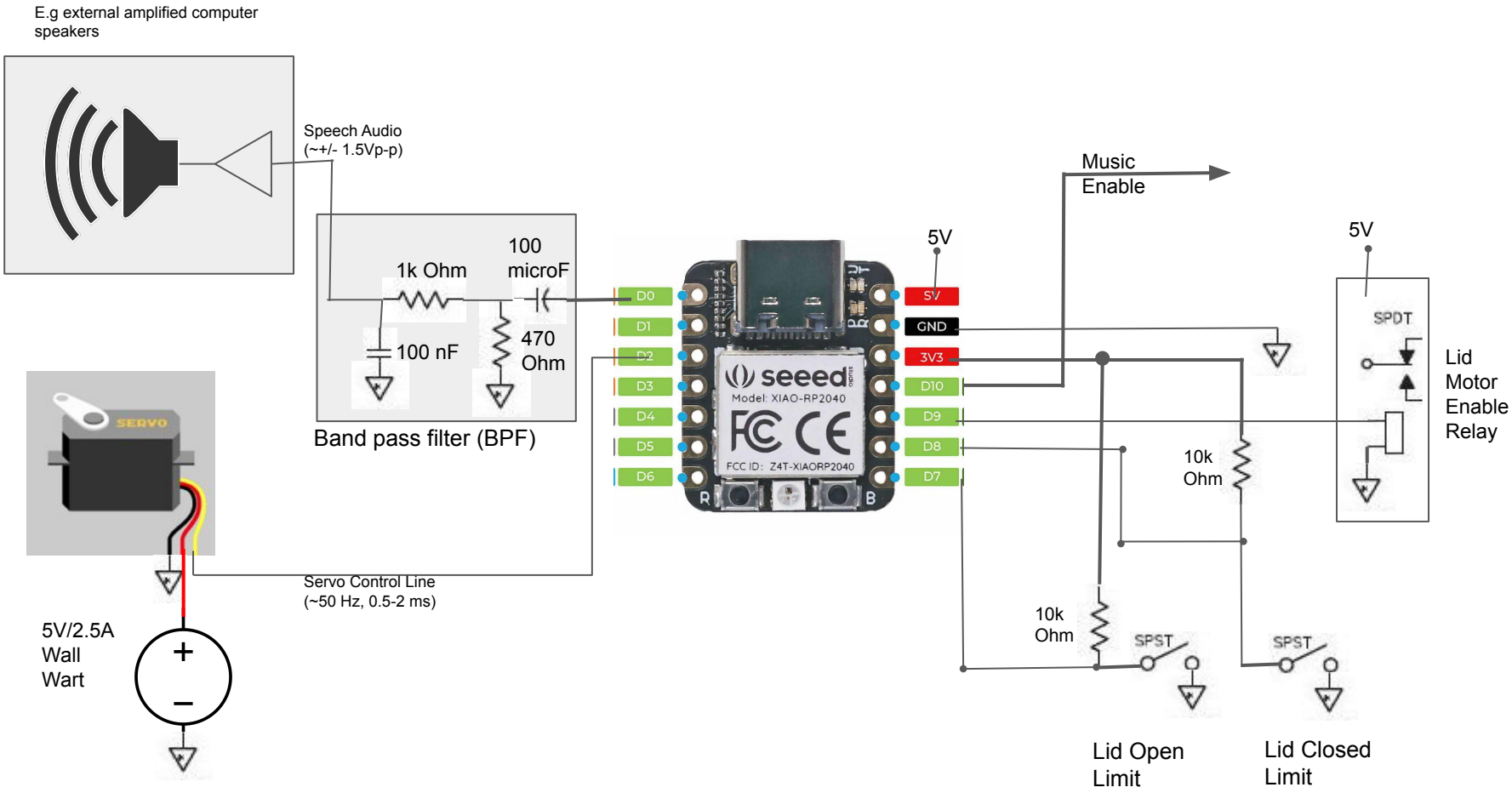
How the servo/speech audio is synchronized



The generated .h file has const buffers of audio, motion PWM set values. This is included in the Arduino project, compiled, then loaded onto the Seeeduino Xiao chip

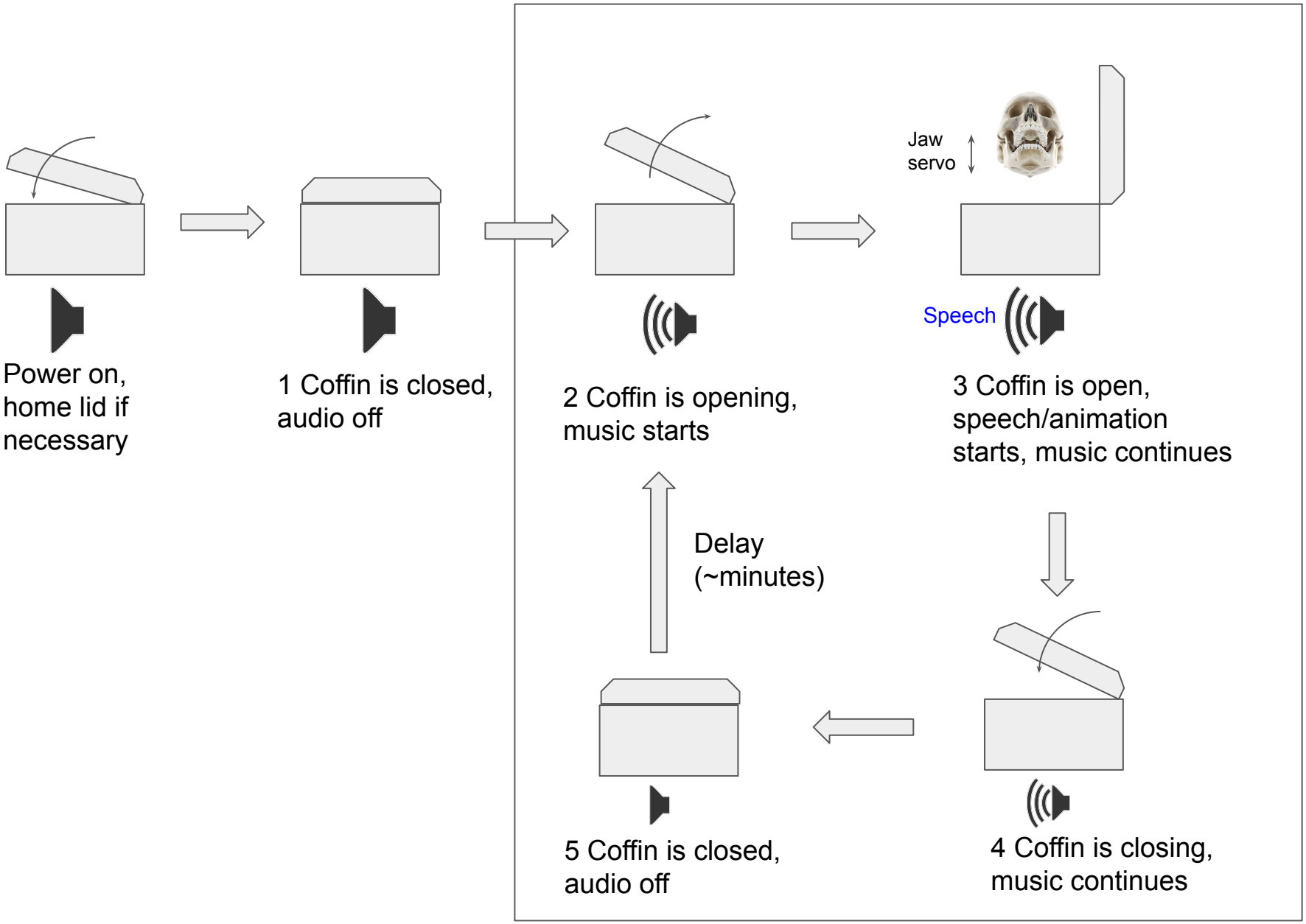
See https://github.com/jmoonware/jaws

Circuit diagram for overall system: RP2040-based microcontroller

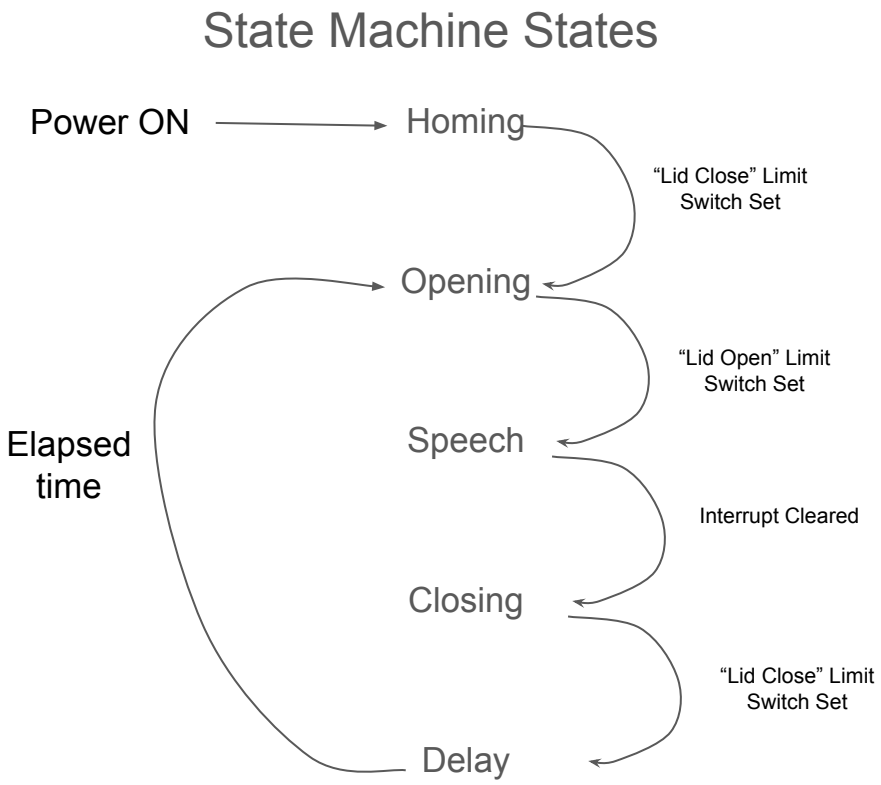


Note: could easily add second, independent stereo channel (and BPF) on D1

Overview of performance - animated skeleton in coffin



Arduino Software State Machine



See https://github.com/jmoonware/jaws_arduino

Extracting audio from video: ffmpeg usage

Example:

ffmpeg.exe -i test_raw.mp4 -map 0:a -ar 20000 -c:a pcm_u8 -ac 1 output20_u8_1.wav

Options explanations:

-i ⇒ input mp4 video file

-map 0:a ⇒ Select input 0, map all (‘a’) audio streams

-ar 20000 ⇒ audio sampling frequency (here, 20,000 Hz)

-c:a pcm_u8 ⇒ set codec to 8 bits, unsigned (so audio value is 0-255)

-ac 1 ⇒ Set to (here, 1) output audio channels (i.e. mono)

How the Pulse Width Modulation (PWM) Counters are set

“IRQ PWM” - generates an interrupt on each audio sample (~20kHz)

“Motor PWM” - generates (~50 Hz update, ~0.5-2 ms width) servo control pulses

“Audio PWM” - generates PWM signal (~500 kHz), then band-pass filtered into ~20-10kHz analog signal