Admin

Wrap of assignments 🎁

Bug Fix Party 🎉 today 2-5pm

Final deadline for re-test fixes and full system **5pm Sun Nov 24**

Onto … projects!

Proposals due now, add to Amazon group order by Monday

Today: Output

Use gpio to drive outputs

Communicate, make sounds, control servos/motors, …

Digital to analog conversion

GPIO output

GPIO pin is for digital signal (very little power)

3.3V logic level

Pi can supply max current ~50mA across all pins

Max total power 0.165W

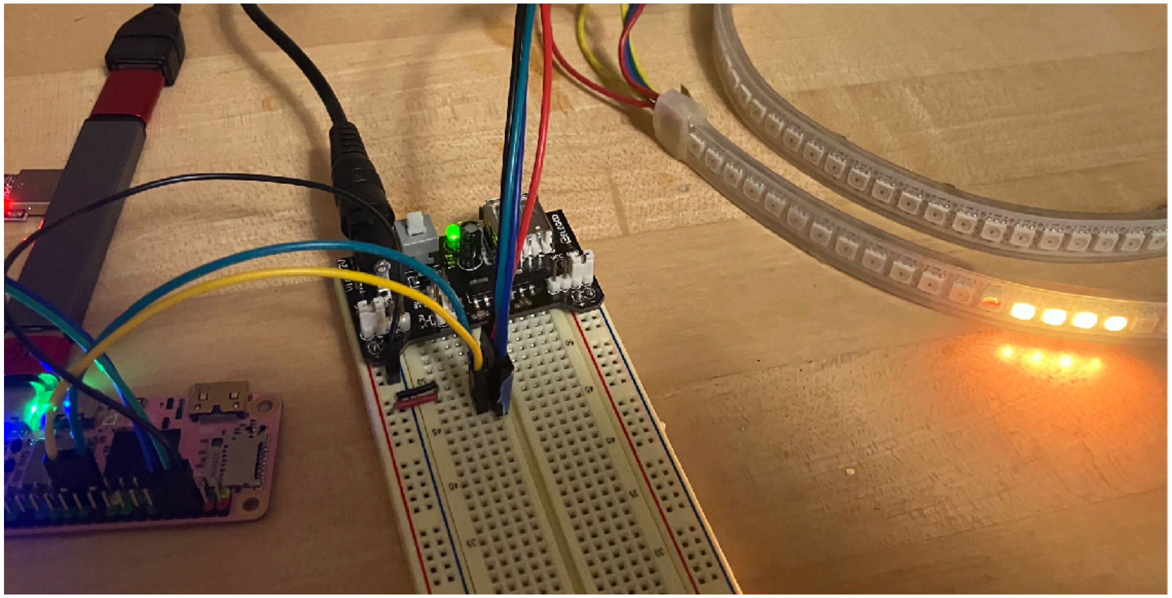
Output devices often need more power

High-power LEDs, long LED strips, motors, speakers, etc.

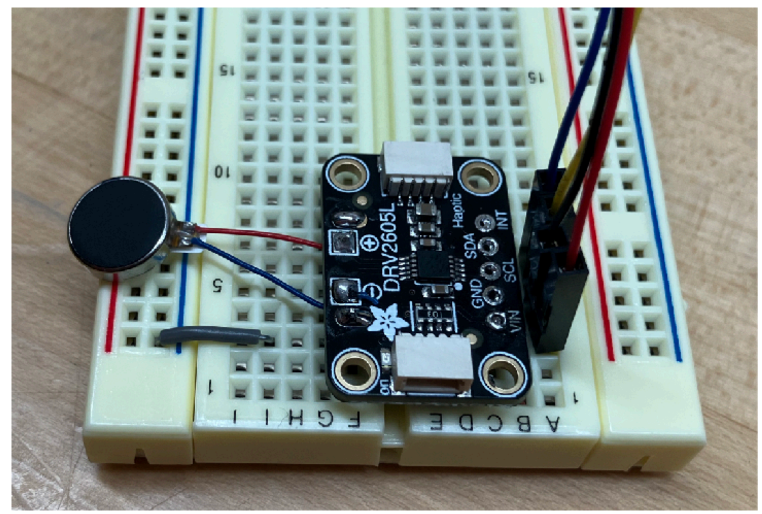
Attempt to drive via gpio can fail and/or damage Pi

Calculate power requirements and select appropriate power source (DC wall adapter, battery park)

Demo: SPI & I2C

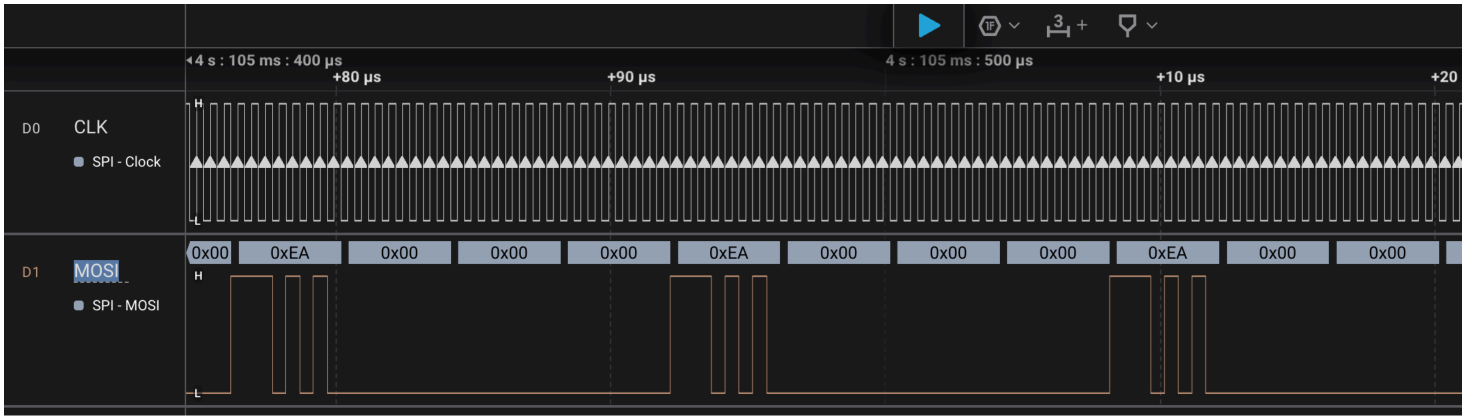


SPI: APA102 DotStar RGB led strip

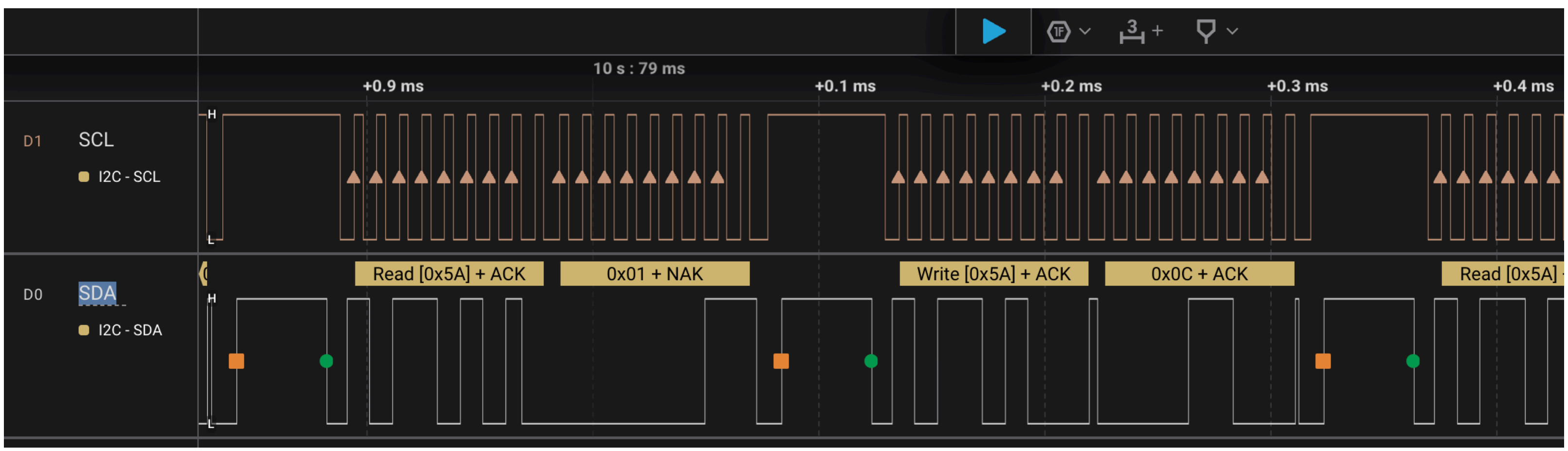


I2C: DRV2605 haptic vibration controller

Logic Analyzer FTW



SPI: APA102 DotStar RGB led strip



I2C: DRV2605 haptic vibration controller

GPIO output is digital

Configure pin as output, write 1 or 0

What can can a high/low voltage signal?

- Turn on/off: led, transistor, relay
- Single bit in comm protocol such as UART or PS/2

How to turn voltage into motion, sound, radio, …

Electromagnetism in action!

Piezoelectric disk

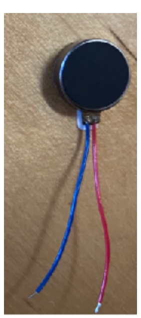
Passive buzzer

Speaker

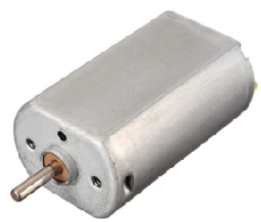
Linear actuator

Electromagnet

Solenoids



Vibrating disk



DC motor

Stepper motor

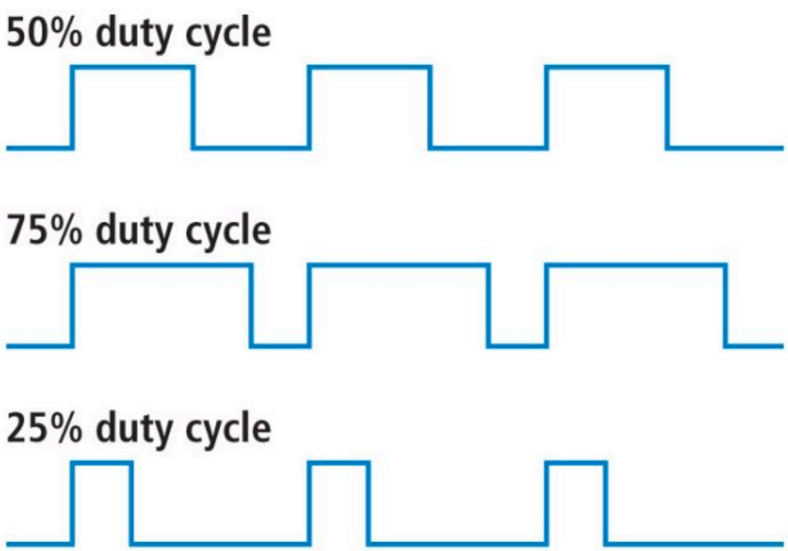
Servo

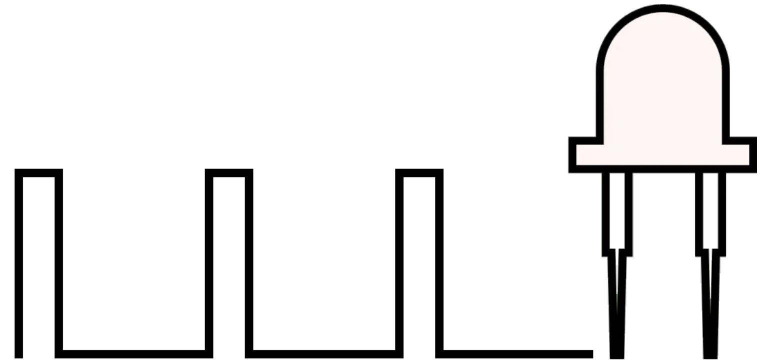
Demo: electromagnet, solenoid

How to synthesize analog signal from digital hardware?

Digital-to-analog conversion (DAC)

Pulse-width modulation (PWM)





Generate square wave of given frequency

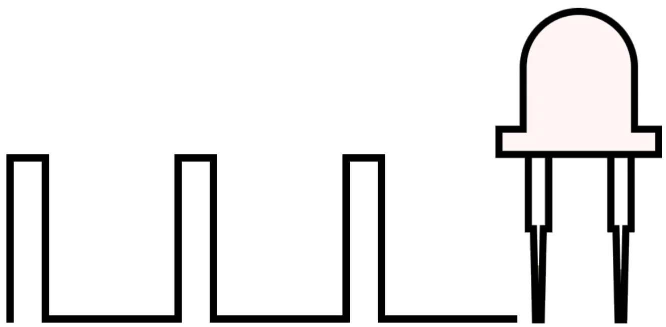
Duty cycle = ratio on/off in one period

Higher/lower frequency — what is effect?

Smaller/larger duty cycle — what is effect?

Demo: Software PWM

Code to "bit-bang" gpio pin



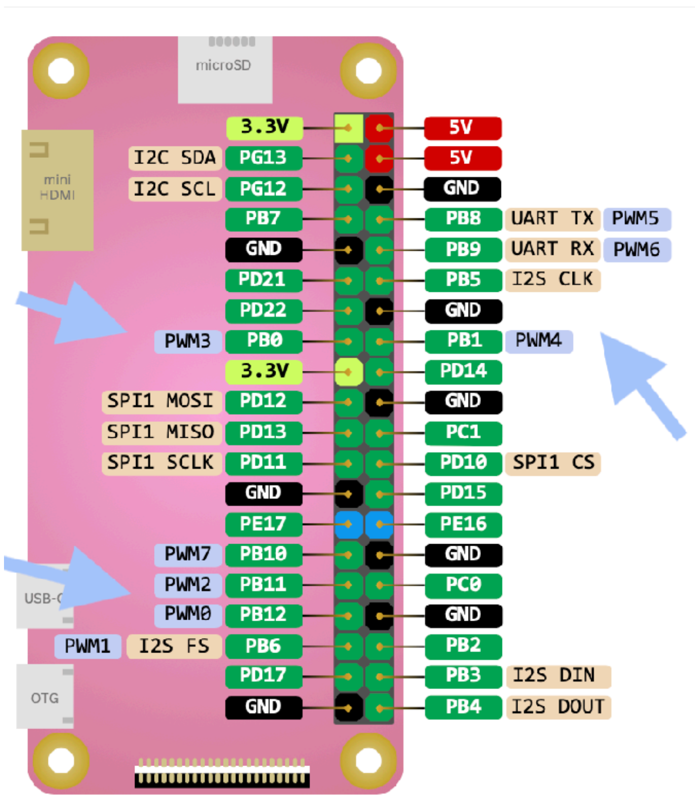
Ramp led on/off

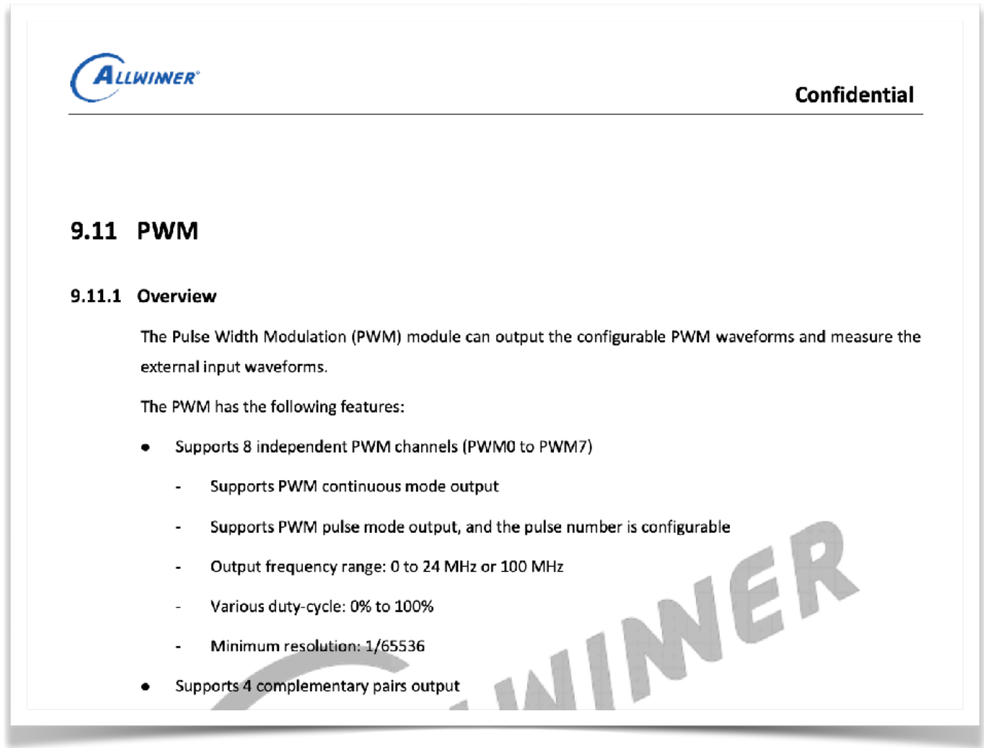
```
for (int i = 0; i < n; i++) {
    gpio_write(pin, 1);
    timer_delay_us(on_time);
    gpio_write(pin, 0);
    timer_delay_us(off_time);
}
```

(Q:What happens if substitute motor or buzzer for led?)

Works, but … CPU consumed with timing-sensitive task

Hardware PWM





ALLWINNER

Confidential

9.11 PWM

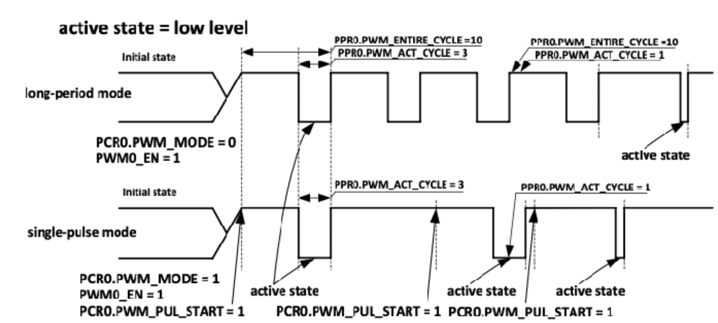
9.11.1 Overview

The Pulse Width Modulation (PWM) module can output the configurable PWM waveforms and measure the external input waveforms.

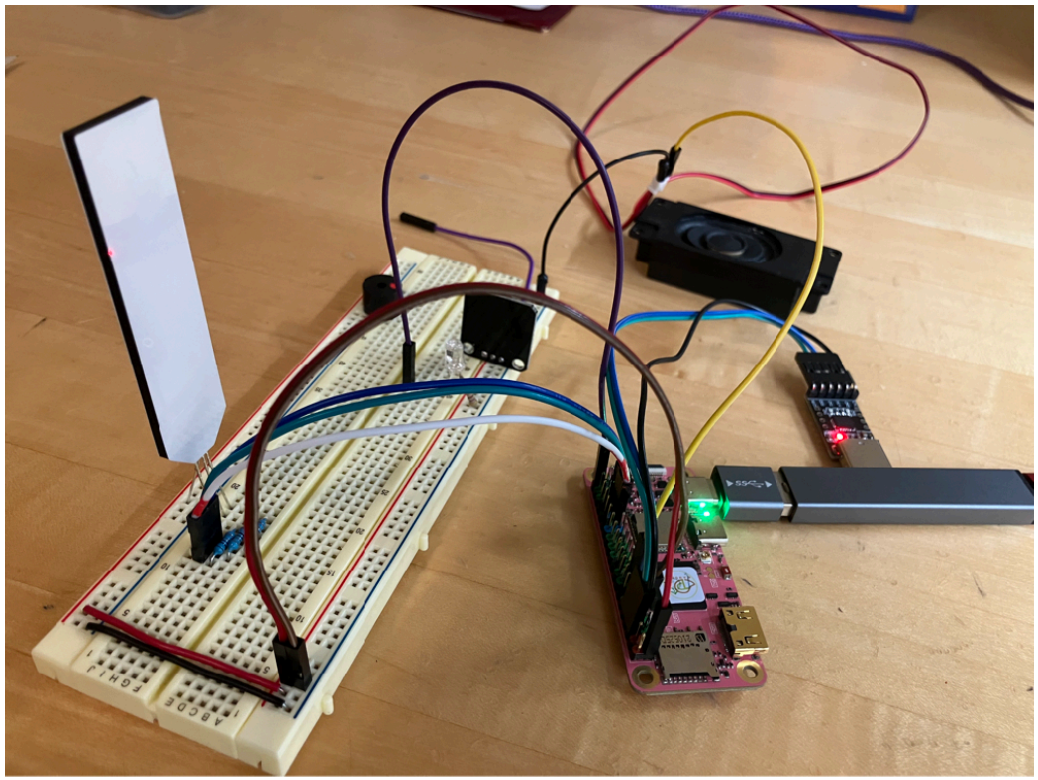
The PWM has the following features:

- Supports 8 independent PWM channels (PWM0 to PWM7)
  - Supports PWM continuous mode output
  - Supports PWM pulse mode output, and the pulse number is configurable
  - Output frequency range: 0 to 24 MHz or 100 MHz
  - Various duty-cycle: 0% to 100%
  - Minimum resolution: 1/65536
- Supports 4 complementary pairs output

Figure 9-89 PWM0 Output Waveform in Pulse Mode and Cycle Mode

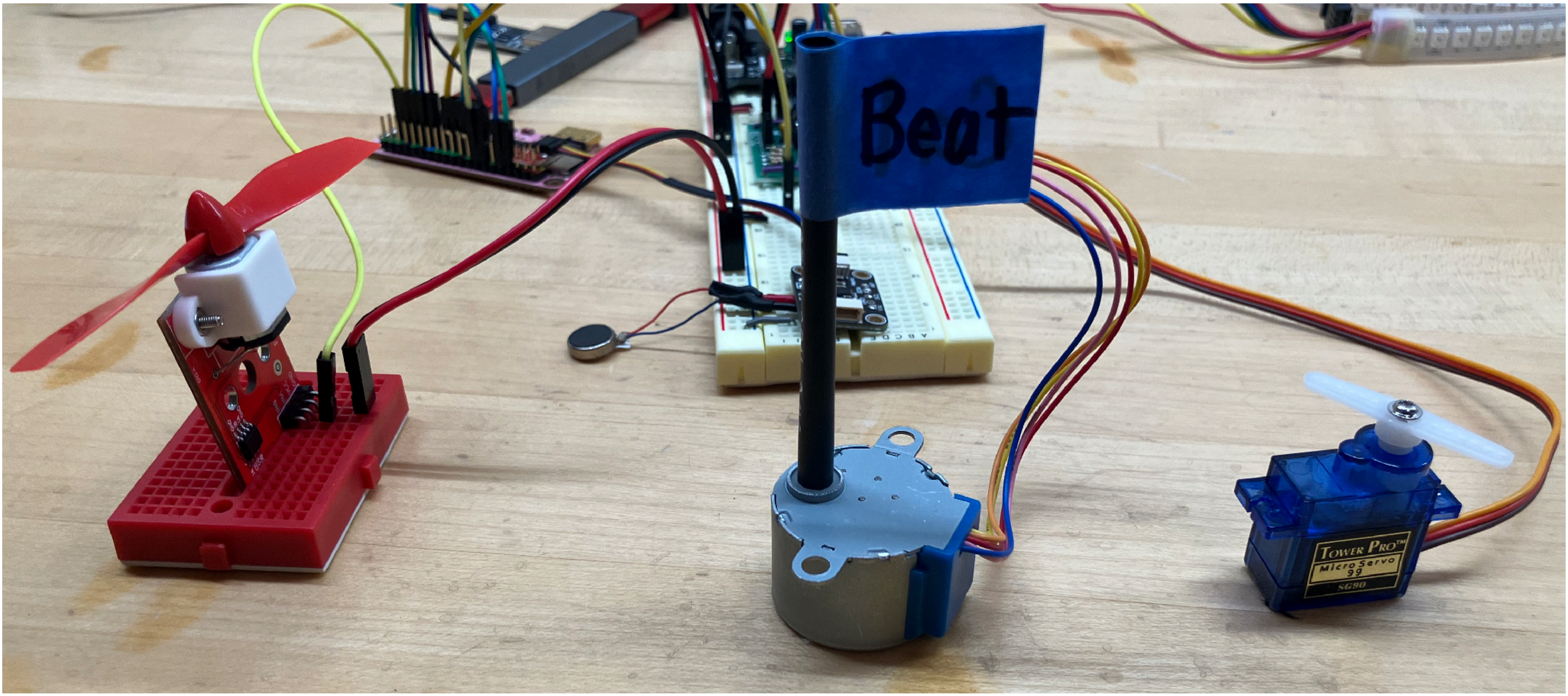


Demo: Hardware PWM



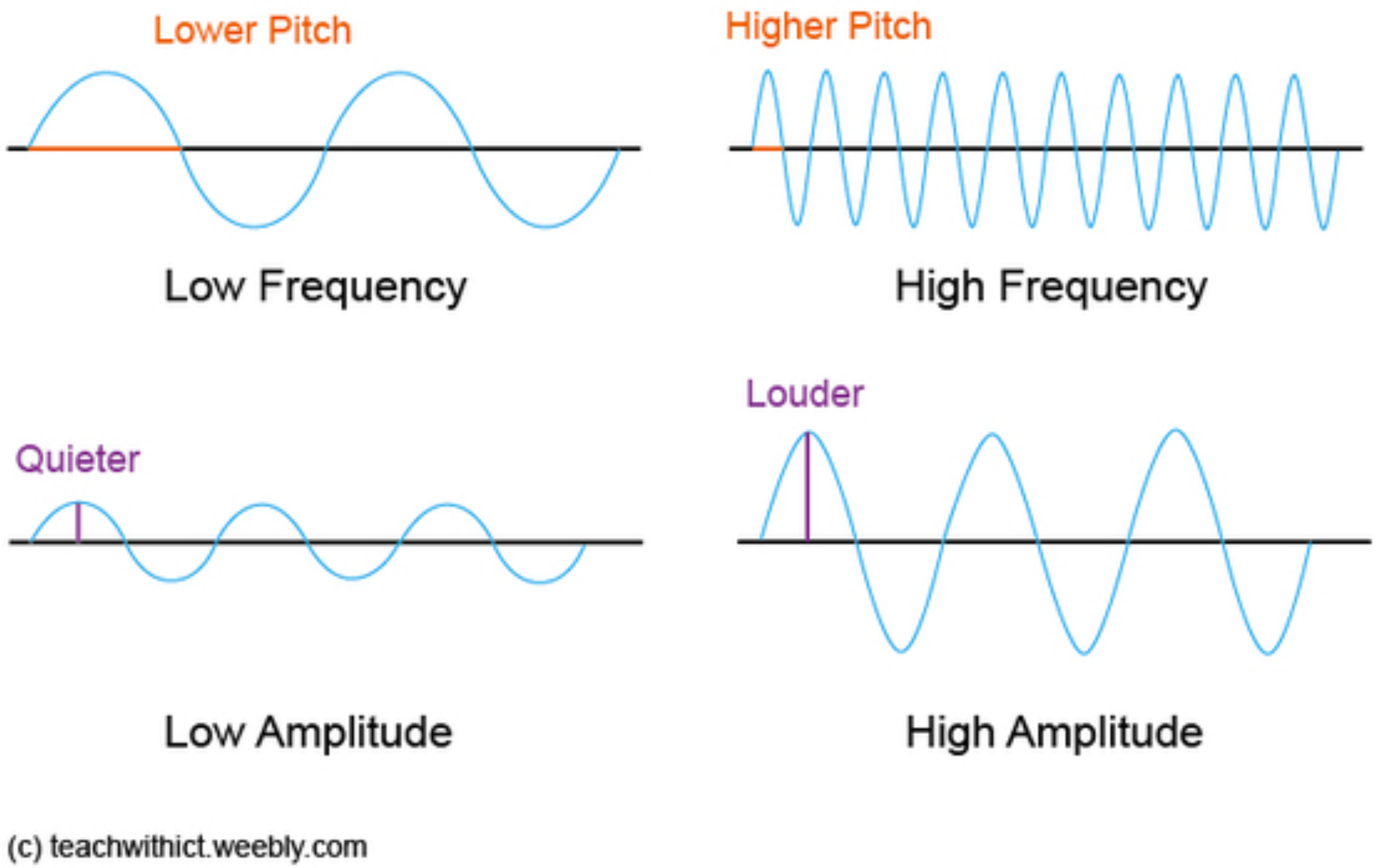
RGB led
Square wave tones and sound effects

Demo: Motors



DC motor, stepper, servo

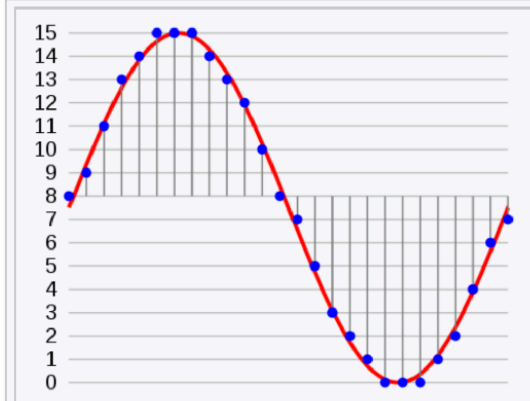
Sound Waves


Lower Pitch
Higher Pitch
Low Frequency
High Frequency
Quieter
Louder
Low Amplitude
High Amplitude
(c) teachwithict.weebly.com

Pulse-Code Modulation

I2S/PCM standard for digital sound
Quantization of sound wave into array of samples
".wav" file is raw PCM data (MP3 is compressed form)

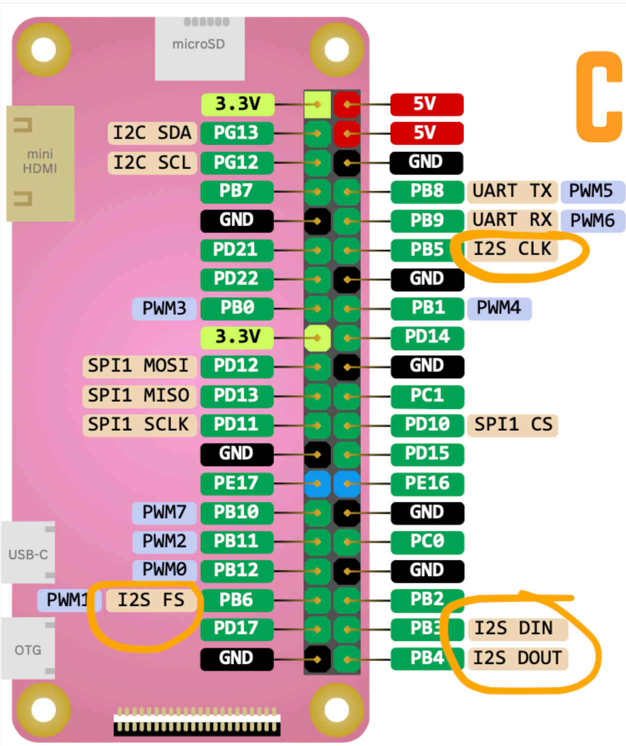
Playback needs sampling rate, bit depth, num channels
CDs are PCM, sample rate 44.1 kHz, 16-bit, stereo

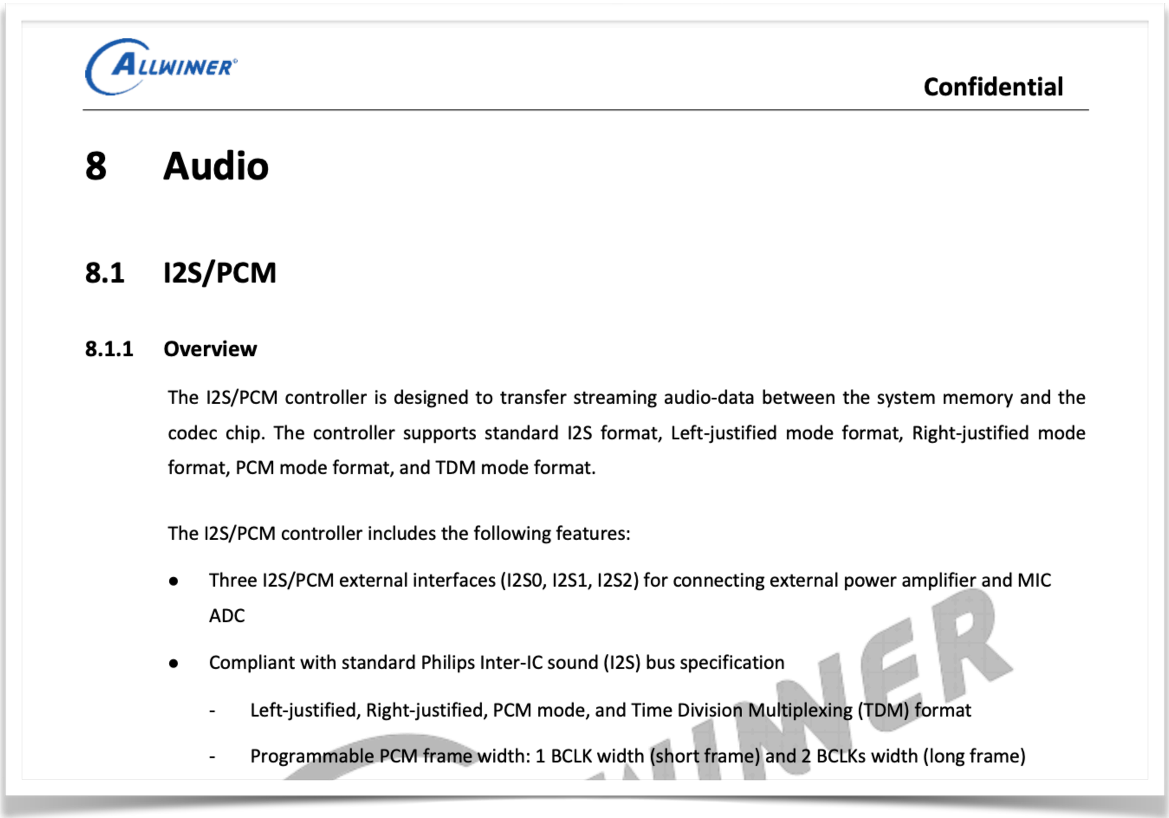


Sampling and quantization of a signal (red) for 4-bit LPCM over a time domain at specific frequency

Source: Wikipedia

I2S





ALLWINNER

Confidential

8 Audio

8.1 I2S/PCM

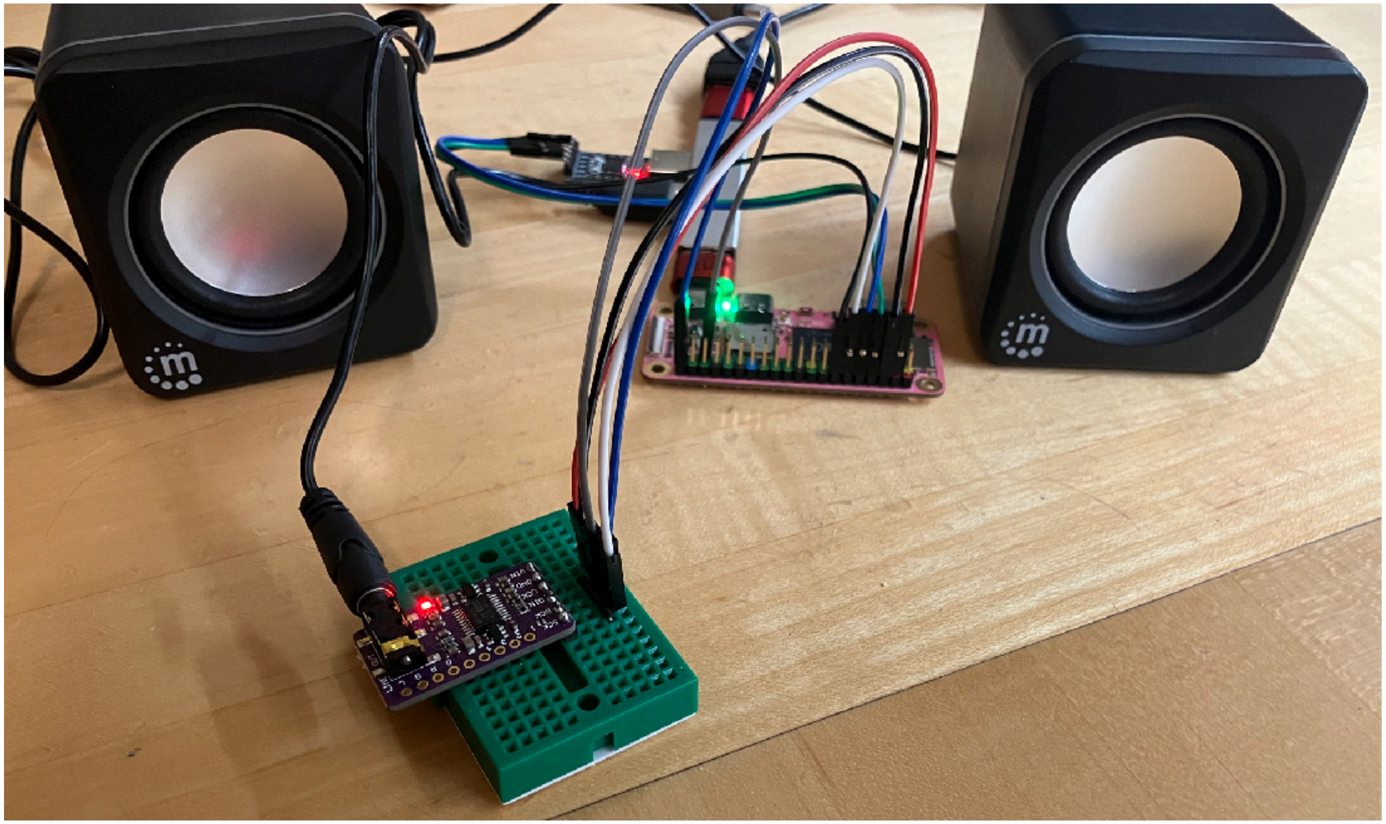
8.1.1 Overview

The I2S/PCM controller is designed to transfer streaming audio-data between the system memory and the codec chip. The controller supports standard I2S format, Left-justified mode format, Right-justified mode format, PCM mode format, and TDM mode format.

The I2S/PCM controller includes the following features:

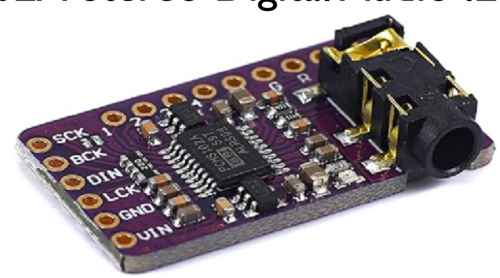
- Three I2S/PCM external interfaces (I2S0, I2S1, I2S2) for connecting external power amplifier and MIC ADC
- Compliant with standard Philips Inter-IC sound (I2S) bus specification
  - Left-justified, Right-justified, PCM mode, and Time Division Multiplexing (TDM) format
  - Programmable PCM frame width: 1 BCLK width (short frame) and 2 BCLKs width (long frame)

Demo: I2S



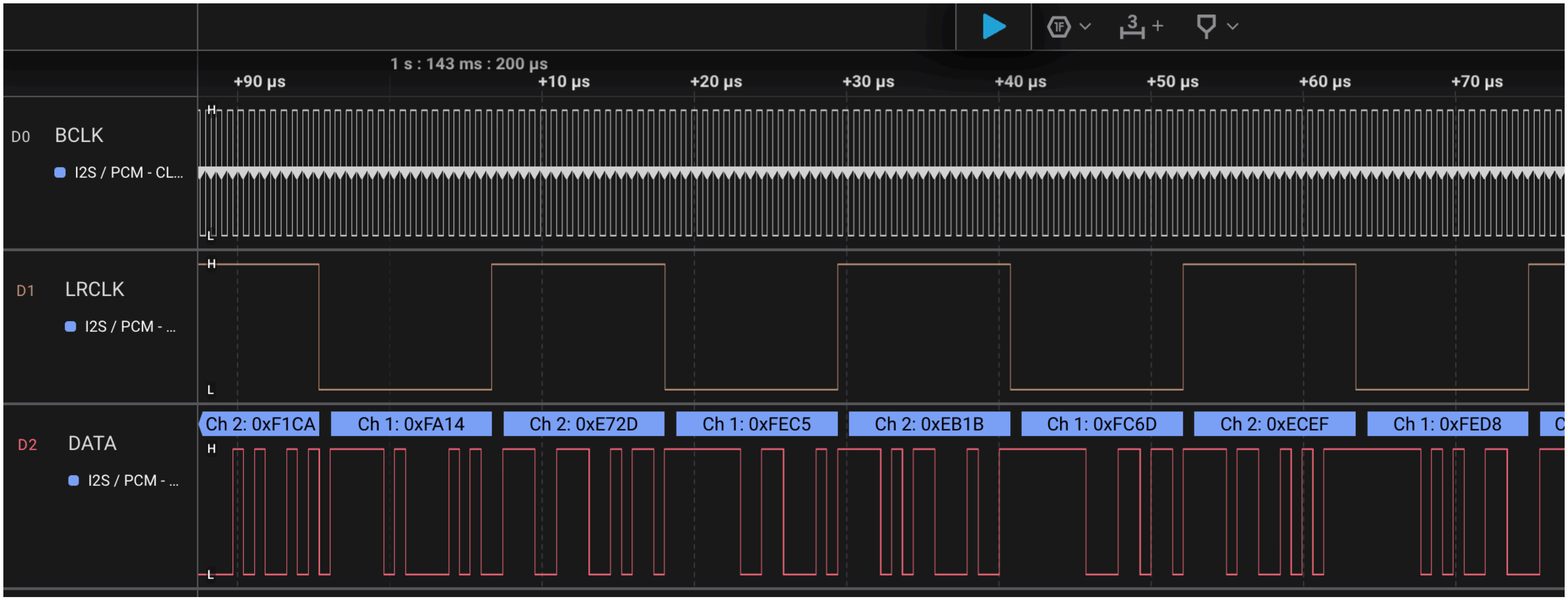
Play wav file

DAC: PCM5102A Stereo Digital Audio I2S



python script convert samples into C array compiled into program

I2S protocol capture



Bit clock
LR clock (frame)
Data (sample value)

For "real" music: MIDI

Musical Instrument Digital Interface

Simple protocol to control musical instruments

Emerged from electronic music and instruments in 1970s

First version described in Keyboard magazine in 1982

MIDI protocol

31.25 kbps 8-N-1 serial protocol

Commands are 1 byte, with variable parameters (c=channel, k=key, v=velocity, l=low bits, m=high bits)

Command	Code	Param	Param
Note on	`1001cccc`	`0kkkkkkk`	`0vvvvvvv`
Note off	`1000cccc`	`0kkkkkkk`	`0vvvvvvv`
Pitch bender	`1110cccc`	`0lllllll`	`0mmmmmmm`