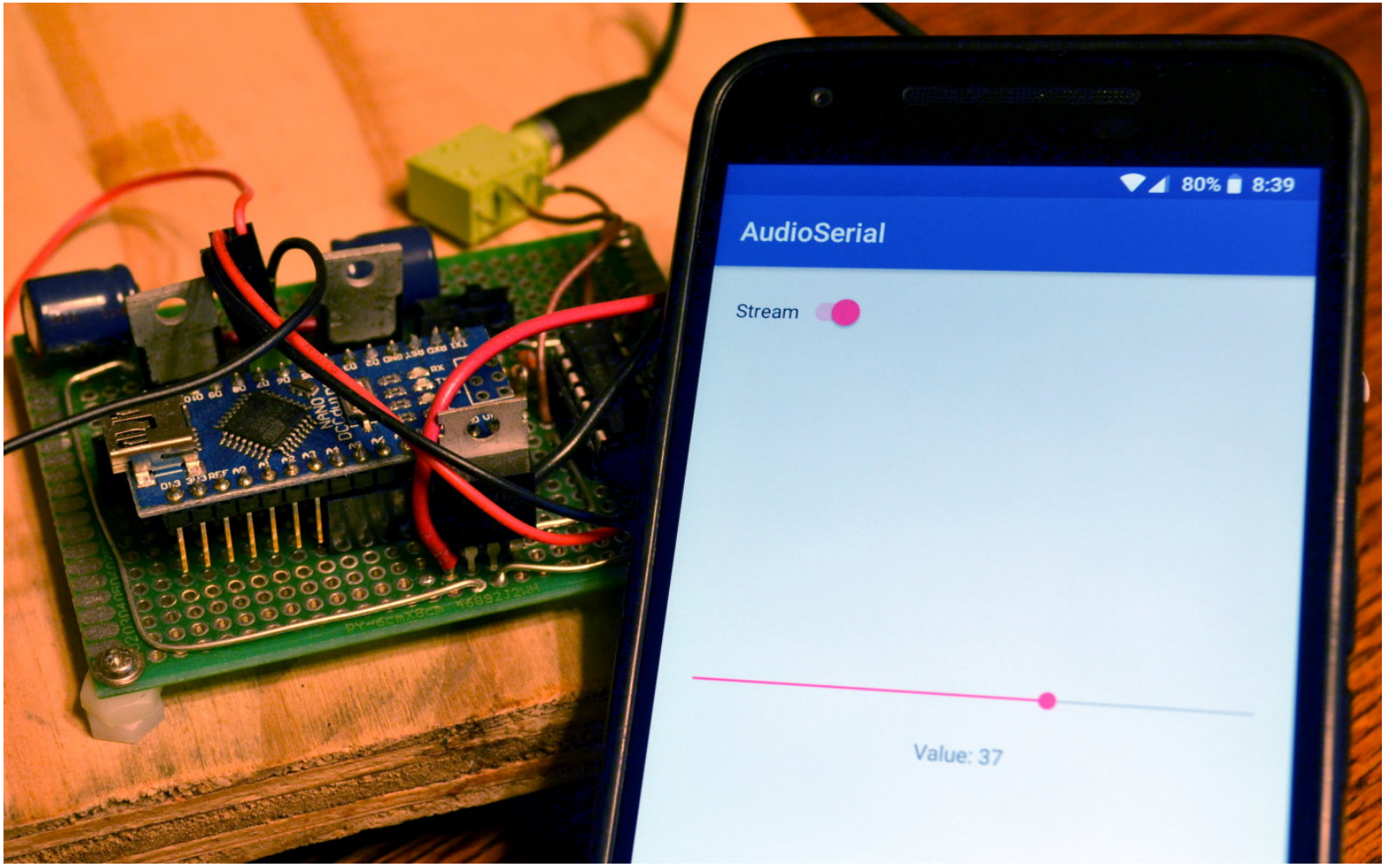
MAY 26, 2018 / BY DAVID WEHR / IN HOW-TO / 4 COMMETNS

SERIAL COMMUNICATION VIA AUDIO ON ANDROID

In this blog post, we'll go over how to send serial data from an Android phone using just the audio port.



When working on AndroWobble, we needed a low-latency way to send data from an Android phone to an Arduino microcontroller. Revisiting this over a year later, I'm not sure why we didn't simply buy a USB On-The-Go (OTG) cable and communicate that way, so I'll pretend that we had a good reason for doing this. Regardless, many microcontrollers don't have a USB to TTL chip like the Arduino has, so being able to send raw serial data from a phone is still useful.

~~All~~ Nearly all phones have a headphone jack, which can create arbitrary waveforms, and what is a digital signal but a specific waveform? So, let's see if can send a serial signal out the audio port.

SERIAL COMMUNICATION

Here's a refresher on the asynchronous serial protocol. It runs at a pre-specified baud rate (commonly 9600, 57600, or 115200), which determines how wide each pulse is. Data is sent in "frames", which include

- A start bit (low logic level)
- 5-9 data bits
- An optional parity bit
- 1-2 stop bits (high logic level)

It's asynchronous, so when nothing is being sent, the line is held high until the next data frame. Sparkfun has a good article with more details, if you're not familiar.

To send the waveform, we make use of the phone's digital to analog converter (DAC). These can be guaranteed to process at least 44 100 samples per second (44.1 kHz), which ensures that they can recreate the entire audible frequency range (See the Nyquist rate). In practice, they typically run at 48 kHz.

AUDIO OUTPUT

To create the correct waveform, we can use the Android NDK and the Oboe library for high-performance audio.

With 8-bit data chunks, there are only 256 unique frames that we might send, so they can be pre-generated and stored in a table. Below is code for creating the wave table. With the comments, it should be clear enough; for each audio sample, we figure out which bit of the frame is being output, and set the audio to ± `MAXIMUM_AMPLITUDE_VALUE`. For 16-bit PCM, this is ± 32 768.

```
1   // Number of bits in each byte frame (start bits, stop bits, data)
2   int byte_frame_bits = DATA_BITS + START_BITS + STOP_BITS;
3   // Number of samples per UART bit
4   float bit_width = sample_rate / BAUD_RATE;
5   // Number of samples per byte frame
6   frame_width = std::ceil(byte_frame_bits * bit_width);
```

```
7
8    // Fill out bytes wavetable
9    for (int byte = 0; byte < bytes_wavetable.size(); byte++) {
10       auto& wave = bytes_wavetable[byte];
11       wave.resize(frame_width);
12
13       for (int i = 0; i < frame_width; i++) {
14           int bit_id = std::floor((float) i / bit_width - START_BITS);
15           int16_t to_send;
16           // start bits
17           if (bit_id < 0) {
18               to_send = -MAXIMUM_AMPLITUDE_VALUE;
19           }
20           // data bits
21           else if (bit_id < DATA_BITS && bit_id >= 0) {
22               // 0/1 bit to send
23               uint8_t bit_send = (byte >> bit_id) & (0x01);
24               // 0/1 expanded to minimum or maximum amplitude
25               to_send = ((2 * bit_send) - 1) * MAXIMUM_AMPLITUDE_VALUE;
26           }
27           // stop bit and holding line high to indicate no data
28           else {
29               to_send = MAXIMUM_AMPLITUDE_VALUE;
30           }
31           wave[i] = to_send;
32       }
33   }
```

Oboe allows you to set a callback whenever new audio data is needed (`AudioStreamCallback::onAudioReady()`), so we can copy the frames from the table into the output audio buffer, or if there is no data to send, fill the buffer with line high values.

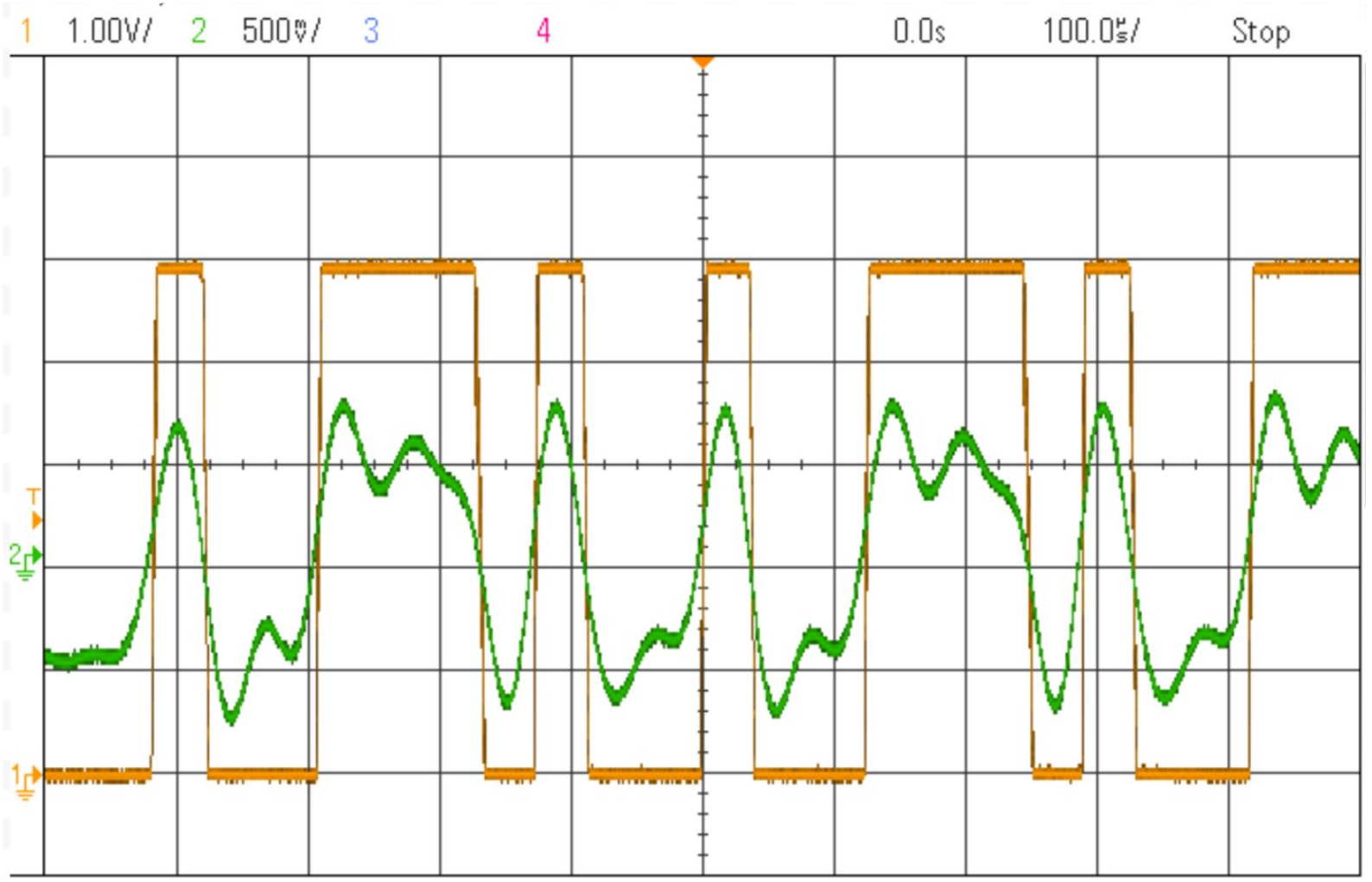
SHAPING THE SIGNAL

Great, so we can just send our waveform out through the audio port at 24 kilobaud (two samples are needed to make a square wave)! Not so fast.

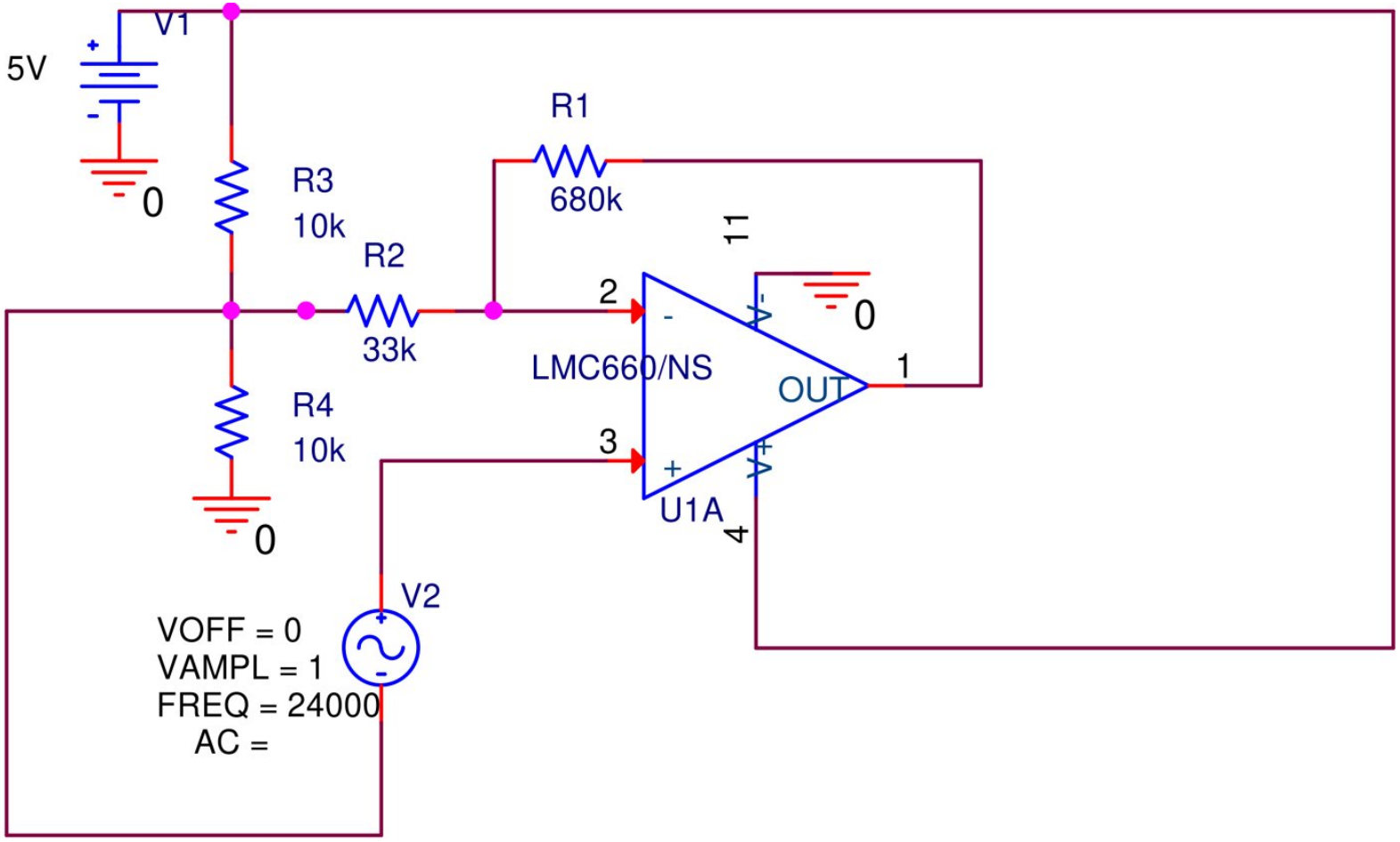
There are two problems to handle:

1. Firstly, the output is in the range of ± 1.5V; we need 0V-5V.
2. Secondly, an audio DAC doesn't actually output the samples directly as given. It's trying to recreate a sampled audio signal, so it will do fancy interpolation. (Read about the Nyquist-Shannon sampling theorem and the sinc function to learn more.)

Below is an oscilloscope probe showing what the raw signal out of the DAC looks like (green), and what we'd like it to look like (orange). Note that the two signals have different vertical scaling and offset.



With an op-amp, we can design a comparator circuit that will output 0V when the signal is negative, and +5V when the signal is positive. Here is a good article on ElectronicsTutorials with more info about op-amp comparators.

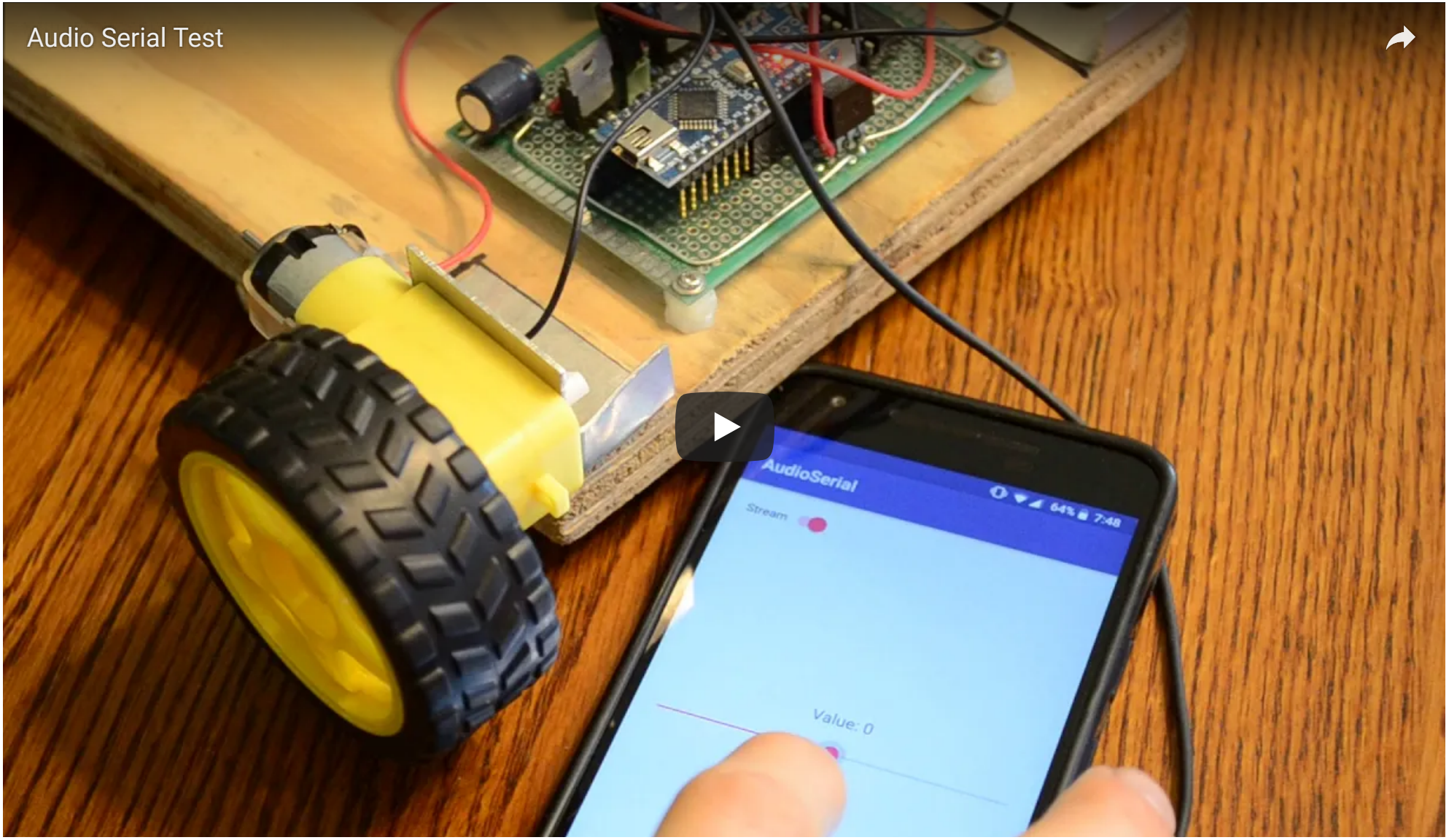


The input audio signal is represented by V2, and the output appears at the op-amp OUT terminal. The circuit is a non-inverting comparator with positive feedback (resulting in hysteresis for stability near the zero crossover point). Resistors R3 and R4 form a voltage divider to make the reference voltage of 2.5V. Then the incoming audio signal is biased by +2.5V by connecting audio ground to the reference 2.5V. Therefore, comparing the biased input to 2.5V is equivalent to comparing to 0V. The output goes from 0V-5V because the op-amp is powered by a 5V supply, and its output is rail-to-rail.

Don't forget to connect the power supply ground to the ground of whatever is reading the serial signal. You wouldn't believe how many times I've forgetten to do that!

EXAMPLE

We used this technique when making AndroWobble, but it used the now deprecated Howie library. So I made a simple app using Oboe to demonstrate a complete example. The full code is on GitHub, and a video of controlling a motor by sending data to an Arduino is below.



CATEGORIES

How-to

RECENT POSTS

Serial communication via audio on Android

Running CUDA on Google's Project Tango Tablet

TAGS

CUDA

Android

Hardware

DAVID WEHR

RECENT PROJECTS

Parallel kd-Tree Construction on the GPU

Structural Analysis App

Melody Magic

Androwobble

Classifying Amazon Food Reviews

RECENT POSTS

Serial communication via audio on Android

Running CUDA on Google's Project Tango Tablet

CONTACT

Contact Page