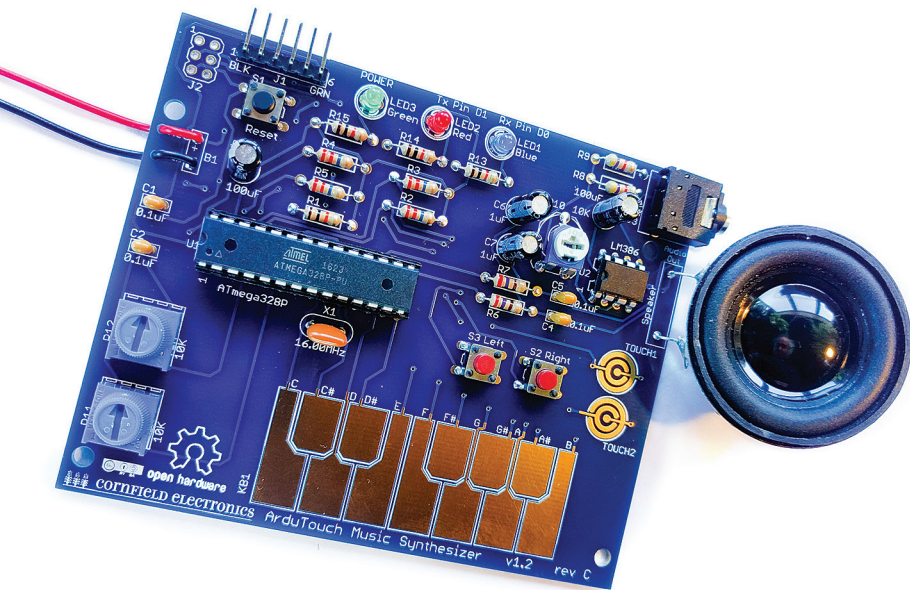


# ARDUTOUCH: AN ARDUINO-COMPATIBLE SYNTHESIZER

## DIGITAL SIGNAL PROCESSING SQUEEZED INTO AN EASY-TO-BUILD KIT



so I used pulse-width modulation (PWM) to encode the stereo audio channels coming out of the ATmega328P. A low-pass filter for each channel, made up of one resistor and one capacitor apiece, converts the PWM signal into audio.

Experienced designers will recognize that the ArduTouch hardware is simple. It is the sophisticated audio-synthesis software library used to program the ATmega328P that is the core of the project.

It's actually pretty easy to get a microcontroller to produce basic musical tones. Just connecting a loudspeaker to an I/O pin, and toggling it on and off to send out a square wave at different frequencies will do (a technique used by no less than Alan Turing with the Manchester Mark II computer). But the result is a harsh sound that's far removed from the richness of a Moog synthesizer.

To get better sound, I needed to turn to digital music synthesis. This is the act of creating binary-number representations of audio waveforms, mucking around with them, and then converting those numbers into analog signals. There are a number of different ways to do this, and I'm pleased to say that the ArduTouch supports quite a few techniques, including the use of wavetables (using digital audio samples stored in memory) and generative methods (calculating audio on the fly).

Once the basic binary representation of a sound has been created, the "mucking around" stage is critically important. Unless you mess around with the sound, it's usually thin and sterile. Dynamic manipulation is what produces cool sounds. The ArduTouch can digitally apply many dynamic functions and effects, including tremolo, vibrato, portamento, and low-pass and high-pass filters.

Even though all of the above is conceptually not too difficult to study and understand, coding it all gets quite complicated. The ArduTouch library, created by my friend Bill Alessi and me, does all the low-level calculations required and took two years to write.

The library is designed for flexibility, allowing even inexperienced users to install new synthesizers that make different sounds. A synthesizer called Thick that combines four sawtooth waveforms comes preinstalled, but five other "prerolled" synthesizers are

A

**As a kid with a lust for music, I was rocked by the Moog synthesizer sounds**

of 1968's *Switched-On Bach*. I needed to learn how to make those sounds!

Thus began a lifetime of learning and synthesizer making while I made my way in the tech industry, where I ultimately created the TV-B-Gone, a gadget that lets you turn off almost any model of remote-controlled television. Since the popular success of the TV-B-Gone, I've created many fun, open-source, hackable hardware kits for the maker workshops I give around the world. In these workshops, newbies learn to solder, tinkering their way into electronics and microcontrollers. Remembering my own youth, I wanted to provide them with a kit that was simple to assemble and use but still a fully fledged music synthesizer. • The result was the US \$30 ArduTouch. This project incorporates, on a single board, a touch keyboard, an ATmega328P (the same processor used in the Arduino Uno), and an audio amp with a speaker. It also has a software library that can serve as an entry point into the world of digital signal processing. • The biggest challenge in designing the board was the ATmega328P's limited number of input/output (I/O) pins. I used 12 I/O pins for the synthesizer's touch keyboard to provide a complete chromatic musical scale. The keyboard is laid out like an old Stylophone—one of my favorite analog synthesizers from the late 1960s—and senses touch capacitively. Two more I/O pins are used for outputting stereo sound, and two pins are used for serial communications (the ArduTouch can be programmed with the standard Arduino development environment, although you'll need an FTDI cable to connect it to a host computer). The remaining I/O pins are just enough for two buttons and two potentiometers that control the synthesizer. • So that builders can get audible results immediately after soldering their kit together, I put an LM386 amplifier chip and speaker on the board. (The amp is bypassed when you plug into the audio output jack.) Digital-to-analog converter chips are expensive,

# HARWIN

**NEW**  
high performance  
connectors  
for demanding  
applications

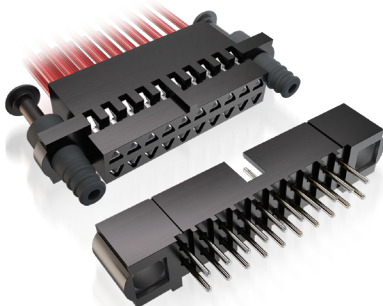


**Vivern**  
circular  
connector

- Break away collar
- Up to +150°C
- 100g shock

[www.harwin.com/vivern](http://www.harwin.com/vivern)

find out  
more

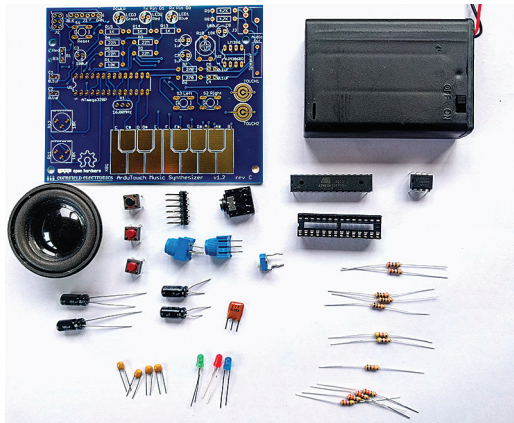


**M225**  
industrial  
connector

- 2mm Pitch
- Fully shrouded  
& polarised
- 100g shock

[www.harwin.com/m225](http://www.harwin.com/m225)

find out  
more



**SOLDERING SOUND:** The ArduTouch requires relatively few components [middle], and it uses a capacitive keyboard to play notes [top]. It's simple enough to assemble in an introductory workshop [bottom].

available for download. More advanced users can learn to create their own synthesizers from scratch by calling a few high-level functions. I'm hoping that people who like the project will create new synths and share their code with me, so that I can share them with others.

But even using the default synthesizer, the sound is surprisingly good for such a low-powered microcontroller (which allows an audio-sample rate of only 15 kilohertz) and such low-cost supporting hardware. You can see and hear many video demos on my YouTube channel.

For the future of the project, we'll keep improving the ArduTouch Arduino library and making more prerolled synths. I'm also working on a more powerful version of ArduTouch that will be suitable for live public performances. This one won't be a kit, but it will still be Arduino compatible. It will have a much more powerful, 32-bit microcontroller with

lots of I/O pins, a digital-to-analog converter, lots of memory (which will allow for MIDI), a small LCD, several buttons and potentiometers and indicator LEDs—and, of course, even better sound. —MITCH ALTMAN

POST YOUR COMMENTS at <http://spectrum.ieee.org/ardutouch1218>