Build yourself a Digital Theremin using Arduino

By BAO SMITH

Silicon Chip has described quite a number of Theremins over the years but this is something new: an Arduino-based Theremin with hand sensing via an acoustic distance sensor.

THE THEREMIN is one of the first electronic musical instruments, and the first to be played without physical contact.

It was patented by the Russian inventor Lev Termen in 1928 and is played by waving your hands near two metal plates or antennas. The proximity of your hands is used to alter the instrument's pitch and volume and it has been used many times in movies and by pop bands.

Part of its appeal comes from its ethereal sound and you can see why if you take a look at a few videos of it being played. To find them, just do a Google search for "Theremin". For a list of previous Theremin projects described in Silicon Chip, see the panel at the end of this article.

This Arduino version of a Theremin was devised by the technical staff at Jaycar Electronics and instead of using analog circuitry to sense hand proximity, it uses a standard Arduino shield, an ultrasonic sensor module, to sense your hand movement.

It only varies the pitch and is quite effective at that, but volume can only be varied by using the volume control on a small amplifier module.

We have a separate article describing the circuitry of the HC-SR04 ultrasonic module elsewhere in this issue.

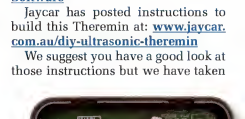
Jaycar sells a kit for this Arduino Theremin and it consists of a Arduino Uno (XC-4410; which is based on an ATmega328P microcontroller), an Arduino prototyping shield (XC-4482), an ultrasonic sensor module (XC-4422), an amplifier module (AA-0373) and a 40mm plastic cone loudspeaker (AS-3004). The kit is available for just $65.30, or less if you have one of Jaycar's "Nerd Perks" cards.

Putting this project together is quite simple but you will need a PC that can run the Arduino IDE, which can be found at www.arduino.cc/en/Main/Software

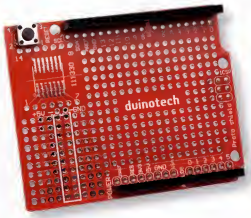
Jaycar has posted instructions to build this Theremin at: www.jaycar.com.au/diy-ultrasonic-theremin

We suggest you have a good look at those instructions but we have taken

The mono amplifier module based on the Champ (Silicon Chip, February 1994) doesn't come with a knob for volume control, but you can easily add one yourself.



In this amplifier module, the leads were soldered and glued to the PCB, making removing them quite difficult.



The prototyping board provides headers to connect with the main Arduino board plus through-hole solder pads, along with a reset switch.

Step 1: the Speaker lead from the amplifier module needs to be soldered to the underside of the speaker. It doesn't matter which lead goes to which solder pad.

a slightly different approach here, which you may prefer.

Putting it together

The prototyping shield can be plugged on top of the Uno board first. Next, you connect the amplifier module to the speaker and then to the prototyping board. This module is actually a built-up version of our very popular CHAMP amplifier which was published in the February 1994 issue of SILICON CHIP. This version is on a slightly smaller PCB and housed in a neat plastic case.

Jaycar recommends removing the PCB from the amplifier module, extracting all leads from it and mounting it on the prototyping shield using wire soldered to the underside of the PCB.

This would be advisable if you want to add an external volume control which can be easily manipulated while you are playing the instrument.

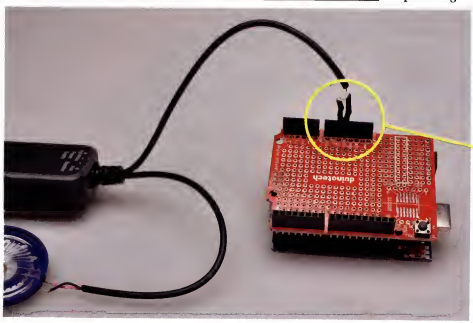
As supplied, the module comes with a tiny preset volume potentiometer on the PCB and that is a bit tricky to adjust. However, it's much easier to leave the amplifier in its case and just wire it up to the board. The reason we did this is to let us easily swap around components on the board, making adjustments easier, and allows us to reuse each part for different projects.

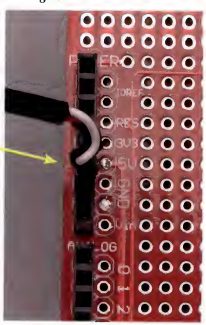
Step 1: solder the speaker positive (pink) and negative (black) leads from the amplifier module to the speaker's terminals (either way around).

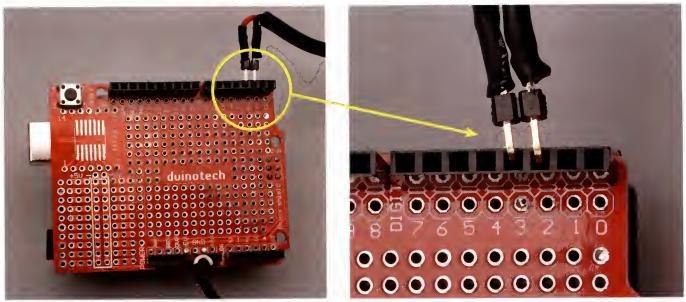
Step 2: make the connections to

You may want to solder a two-pin (or two one-pin) header(s) to the input and power leads from the amplifier. Keep in mind for the amplifier Audio In, right-angle header(s) need to be used if connecting to the female header. This is needed for clearance from the sensor module.

Step 2: the Power Supply positive lead (red) from the amplifier goes to the 5V connection on the prototyping board. The negative lead (white) can then go to either of the two GND connections nearby. We've used a two-pin male header and added heatshrink tubing over the solder joints to provide greater strength.







Step 3: the amplifier Audio In positive lead (red) needs to be plugged into DIGITAL pin 3. The negative lead (black) is not needed and so can go to any unused pin. Here we have the negative lead plugged into pin 2 directly next to the positive lead, but you can choose what works most comfortably for yourself.

power the amplifier module. We soldered the red and black supply leads to a 2-pin male header that can then be plugged directly into the 5V and GND pins on the prototyping shield, which are indicated on the silkscreen printing.

Step 3: solder the red and black input leads of the amplifier module to a 2-pin right-angle header and then plug it into pin 2 (for the black wire) and pin 3 (for the red wire), on the opposite side of the prototyping shield from the power supply connection.

Step 4: make sure the proto shield is correctly plugged into the Uno board.

Step 5: straighten the pins on the ultrasonic sensor and plug it into the prototyping board header next to the amplifier audio input leads. Its four pins are labelled V_{CC}, Trig, Echo & GND. These are plugged into the DIGITAL shield pins with V_{CC} to pin 8, Trig to pin 9, Echo to pin 10 and GND to pin 11. These are default pin locations set by the software but you could modify the software to change them, as explained later in this article.

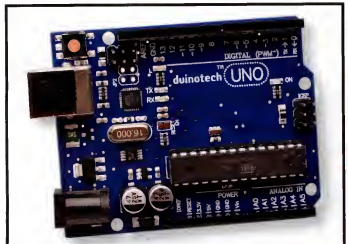
All the above connections are listed in the table entitled "Lead Connections" later in this article, so refer to that if you're unsure.

With all parts connected, the next step is loading the software on the ATmega328P chip via the Arduino IDE. The original software can be found at: www.jaycar.com.au/diy-ultrasonic-theremin#sketchfiles

There will be two files, **Ultrasonic_Theremin.ino** and **sample.c** and these should be downloaded to a folder on your PC named "Ultrasonic_Theremin". The Arduino must be connected to your computer using a USB Type-A to Type-B cable (as commonly used for printers) so that the software can be loaded onto it.

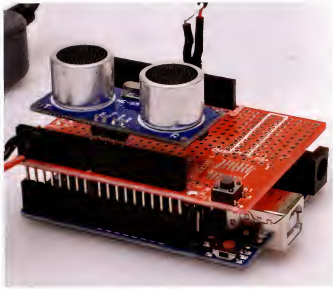
Steps 6 & 7: once the Arduino IDE

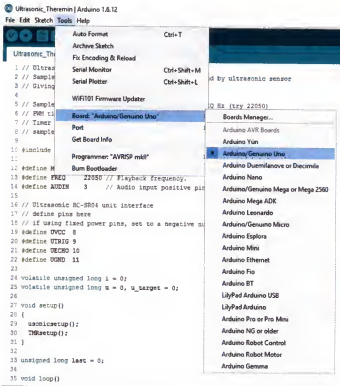
Step 4: if you haven't already done it, now is the time to plug the proto shield into the Uno board. The orientation is simple as both reset switches should be in the same location.



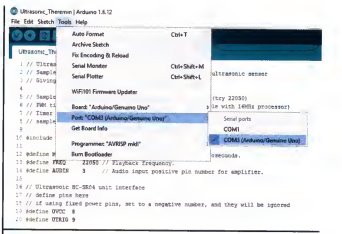
The Uno development board. We use the 5V and GND pins on the POWER header, one PWM~ and four DIGITAL pins for the Theremin.

Step 5: the ultrasonic sensor needs to have its pins straightened and then it can be plugged into the female header with V_{CC} on DIGITAL pin 8 and Trig on pin 9.

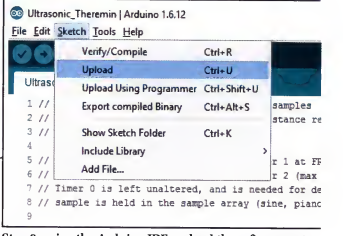




Step 6 (above): check that the Board type selected is "Arduino/ Genuino Uno" in the Arduino IDE, before uploading the software to the board.



Step 7 (upper right): While the Uno is plugged into the computer, check that Port is correctly set to the one that the device is connected to. In this case, ours is on serial port COM3.



Step 8: using the Arduino IDE, upload the software onto the board. Assuming there have been no changes, it should compile and run correctly.

has been installed, open **Ultrasonic_ Theremin.ino** in it and on the menu bar, go to **Tools** and check that the board is set to "Arduino/Genuine Uno" and that **Port** is "COMX (Arduino/Genuine Uno)", where X is whatever port number it has been assigned to on your PC. If your board does not show up in this list, you may need to manually install the drivers for it. Instructions on how to do this can be found at: www.arduino.cc/en/Guide/ArduinoUno

The Arduino uses Virtual COM Port (VCP) drivers to emulate a COM port over a serial connection. If you're interested, it will be explained in greater detail in the next Low-Cost Asian Electronic Modules article on the CP2102 USB-UART bridge in next month's SILICON CHIP magazine.

Step 8: if everything is in working order, on the menu bar go to **Sketch → Upload**. This will compile and upload the software onto the Arduino. The device can then be tested by holding your hand over the ultrasonic sensor. It should produce a sound with a pitch which increases as your hand gets closer to the sensor and conversely, lowers as your hand moves away from the sensor.

If you don't get any sound, check that the amplifier is wired to the correct pins on the prototyping shield and that the compilation and uploading proceeded with no errors, which

The HC-SR04 ultrasonic sensor module, described in greater detail in the article on page 82.

Table 1: Lead Connections

Component	Lead	To Header/Part	To Pin
Amp Power Supply (4-12V)	+ (red)	POWER	5V
	- (white)	POWER	GND
Speaker	± (pink/white)	Mono speaker	
Amplifier Audio Input	+ (red)	DIGITAL	3
	- (black)		any unused
Ultrasonic Sensor	V_{CC}	DIGITAL	8
	Trig		9
	Echo		10
	GND		11

would be displayed at the bottom of the Arduino IDE window.

Besides communication issues, the most likely problem would be if the two provided files are not in the same directory.

Once the software has been uploaded to the device, rather than plugging it into your PC, it can be powered via a 7-12V DC power supply or battery, via the DC barrel socket located next to the USB connector.

Making some improvements

Once you have it running, you will probably find that the Theremin sound is not particularly good and not like the Theremins that you will have seen on videos on YouTube.

With that in mind, we modified the software to give a more realistic Theremin sound, more like that which could be produced by one of the previous Silicon Chip Theremins listed elsewhere in this article.

The important change is the addition of a sinewave look-up table which is substantially smaller (256 bytes compared to 16 kilobytes) than playing back a larger digital sample at a varying rate to control the pitch. This is especially important when considering that memory on the micro is quite sparse, at 31.5KB of usable flash and 2KB of SRAM.

The Silicon Chip version of the software is available for free on our website (www.siliconchip.com.au). Download the two files which are labelled **SC_Ultrasonic_Theremin.ino** and **SC_sample.c**. They should unzip into a suitably named directory ("SC_Ultrasonic_Theremin"). You then upload them to the Uno using the same procedure as described above. We hope you find the resultant sound more satisfying.

One of the advantages of using an Arduino to build this Theremin is that you can easily modify the software if you want to. For example, you could change the linearity of the pitch control or change the waveform. If you do want to modify the software, it would be a good idea to familiarise yourself with a programming language like C or Java. However, even inexperienced readers may have some luck making simple changes.

For example, at the top of the .ino file, some macros are defined which allow you to easily change certain properties of the Theremin:

- MAX_DIST sets the maximum distance for the range sensor, with a value of 5700 approximately equal to 1m. The sensor has an effective range of 2-400cm, although in practice it will barely work beyond 3m, which gives MAX_DIST at most being 300cm ÷ 0.0175cm/µs ≈ 17000 (µs). Note that 0.0175cm/µs is half the speed of sound at 25°C and 100kPa.
- FREQ sets the default playback frequency for the sample. Generally, anything within 22050 ± 5000 (ie, half the sampling rate of a CD) sounds best, but depending on the sample used, your results may vary.
- UVCC, UTRIG, etc define the pin location for the ultrasonic sensor. These can be changed if you want to move the sensor to a different location on the board.

Also, there are four different samples you can use with the Theremin (in sample.c): sine, piano, theremin and sine256. The first three can be easily selected to by changing the name referenced on line 67 of the .ino file, ie:

```
OCR2B = pgm_read_byte
(&theremin[i >> 1]);
```

Here, you can change "theremin" to "piano" or "sine". If you want to use sine256 instead you need to comment out this line (by prefixing it with two slashes, ie, "//"), and un-comment the one above.

Without difficult changes you cannot use the other PWM-enabled pins for the amplifier audio in. Since any DIGITAL pin that the amplifier audio input is on needs to be matched with corresponding OCR register settings, if the pin location is changed the OCR referenced in the code needs to be changed too. For example, by default we use the OCR2 (pins 3 and 11) register which is an 8-bit register, while the other 8-bit register OCR0 (on pins 5 & 6) could also be used. However, OCR1 (pins 9 & 10) is a 16-bit register making working with them quite different.

If you know what you're doing you can alter this, otherwise it's best not to. You can find the pin mapping for the chip here: https://www.arduino.cc/en/Hacking/PinMapping168 and the ATmega328P documentation here: http://www.siliconchip.com.au/l/aaai

Where to get it

All the components for the kit can be purchased from Jaycar as a kit for $65.30, or $52 if you have a Nerd Perks card. It is available from their retail stores and their website (www.jaycar.com.au).

The speaker and amplifier module can easily be substituted to obtain more power and better bass and that should make it considerably more satisfying to play.

Finally, for a detailed Arduino installation guide, see: www.arduino.cc/en/Guide/HomePage

Next month, we hope to publish details on how to add a second ultrasonic module to control the volume of the Theremin. This will be accompanied with changes to the software. SC

Parts List

Ultrasonic Theremin Project Kit

1 Uno Main Board (Jaycar XC4410)
1 Arduino Prototyping Shield (Jaycar XC4482)
1 Ultrasonic Sensor Module (Jaycar XC4442)
1 Mono Amplifier Module (Jaycar AA0373 or equivalent)
1 8Ω 1/4W 40MM Speaker (Jaycar AS3004 or equivalent)

Additional Items

1 USB Type-B to Type-A male-male connector (e.g, printer cable)
1 7-12V DC plugpack (if you want to run it without USB)
1 2-pin male header
1 2-pin male right-angle header

Silicon Chip Theremin Projects

1. Opto-Theremin*, September & October 2014
2. The Theremin Mk.2 with improved voicing, March 2009
3. Mini-Theremin, July & August 2006
4. MIDI Theremin, April & May 2005
5. The Theremin, August 2005

* Note: PCBs and key parts for the Opto-Theremin project are available from the Silicon Chip shop – see our website for more details (www.siliconchip.com.au/Shop).