

Anonymous MIT student

Professor Ariza

21M.380

2 December 2009

A Breath-Controlled Arduino Theremin: The Airemino

The Theremin (or Thereminovox) is *traditionally* a continuously sounding, monophonic instrument controlled by moving one's hands through free space in proximity to two antennae – one that controls the frequency of the generated sound and another that controls its amplitude. It was originally designed in the 1920s by Russian electrical engineer Lev Sergeyevich Termen as an extension of his work on a security system that involved sensing the presence of humans by detecting fluctuations in radio signals. The Theremin is one of the few instruments that receives input from the musician without requiring any physical contact, in addition to the modern laser harp.

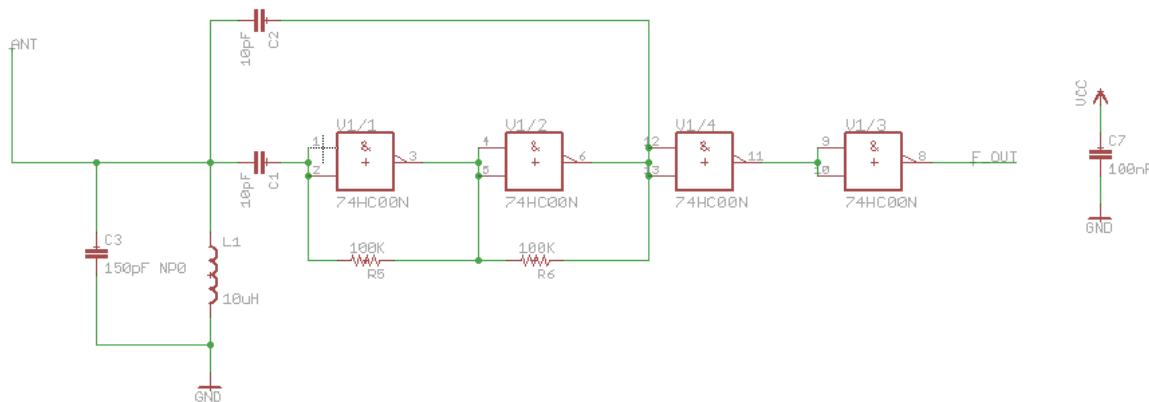
In order to control the pitch of the Theremin, the musician adjusts the position of his fingers in free space to change the apparent capacitance of the frequency sense antenna, which determines the natural frequency at which an inductor-capacitor (LC) oscillator will resonate. The output signal of this variable frequency LC oscillator is then mixed (multiplied) with the output signal of a fixed frequency LC oscillator, whose resonant frequency is equal to the resonant frequency of the variable oscillator when the antenna is in isolation (no human hand present). The mixing operation outputs two signals, one with frequency equal to the sum of the two input frequencies and one with frequency equal to the difference. A simple low-pass filter will eliminate the sum frequency and leave only the difference, or “beat” frequency.

The signal processing technique described above is called heterodyning and is of great value to the original analog Theremin because it transforms minute fluctuations in capacitance (due to hand motion) into audible frequencies. Consider the case where the fixed oscillator rings at 1 MHz and the variable oscillator fluctuates around 1 MHz, by only 0.1%, due to hand motion. If we subtract 1 MHz from 1.001 MHz, we are left with 1 kHz, which is indeed in the audible range. To sweep the entire audible range, the frequency of the variable oscillator must fluctuate by only 0.2%. Although the original analog Theremin design delivers the classic Theremin sound in the most pure sense, building one can make for an entire semester long project. Building an oscillator that is stable enough for the small frequency fluctuations to be significant (frequency drift must be less than 0.001% for 10 Hz resolution in the example described above) requires very low-tolerance components with little temperature dependence. Additionally, the heterodyning process is difficult to implement in analog electronics and would be error-prone for the novice. However, despite the difficulty inherent to building an analog Theremin, it is possible to transform the underlying principles on which the Theremin operates into the digital domain where they can be dealt with easily. This is exactly the approach that Martin Nawrath took in his article “Arduino + Theremin = Theremino”¹ in the November 2009 issue of Elektor Magazine.

Nawrath preserves the capacitive sense pitch control aspect of the Theremin, but instead of using an analog oscillator with heterodyning circuitry, he uses a digital oscillator and a frequency counter implemented in software on the Arduino platform (AVR microcontroller) to detect the small changes in frequency due to changes in antenna capacitance. The digital oscillator is built using an LC tank oscillator with resonant frequency 4.1 MHz, as well as four

¹ “Arduino + Theremin = Theremino”, by Martin Nawrath (Academy of Media Arts, Cologne, Germany), Elektor Magazine, Volume 1, Issue 11, pp. 32-35 (2009).

NAND gates with the appropriate resistive/capacitive feedback to allow for oscillation. An antenna is added in parallel with the LC tank oscillator just like in the original analog Theremin design. A square wave of frequency $4.1 \text{ MHz} \pm \Delta f$ (in which Δf is the small frequency variation due to hand position) appears at the output of the oscillator, and can be connected directly to one of the Arduino's digital inputs since it is a digital signal. A schematic of Nawrath's variable frequency square wave oscillator is shown below.



The software on the Arduino board performs a few simple interrupt driven tasks to determine the minute fluctuations in oscillator frequency. Two hardware counter/timers are utilized, 16-bit Timer/Counter1 and 8-bit Timer/Counter2. Timer/Counter2 functions as a simple 100 ms timer, and during each 100 ms period Timer/Counter1 counts the number of transitions on the digital input pin connected to the square wave oscillator. The number of transitions observed in a 100 ms time window can be used to infer relative fluctuations in oscillator frequency, by first measuring the number of transitions in the 100 ms time window after the Theremin is turned on (assuming that a human hand is not present initially) and using this value as a baseline. Bringing an object into range of the antenna will change its capacitance slightly, leading to a variation in oscillator frequency and a different transition count. The difference

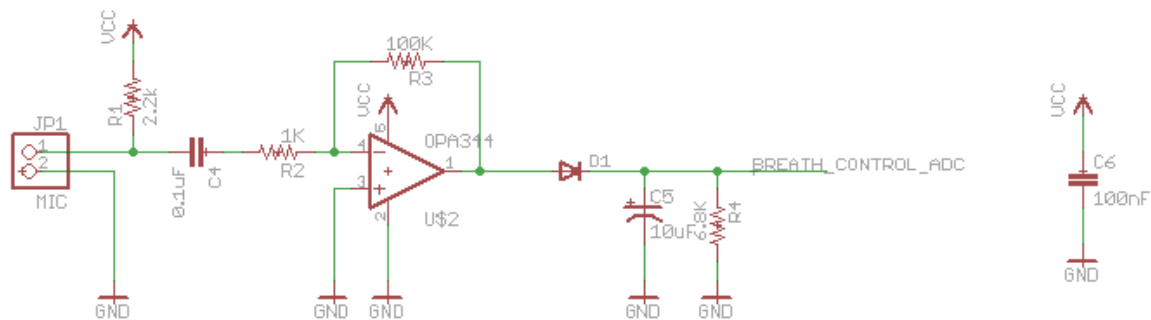
between this transition count and the baseline can be used to adjust the frequency of a simple square wave tone generated on one of the Arduino's digital outputs. In the end, the software is responsible for mapping a small frequency fluctuation in the oscillator's digital output to an audible pitch.

Nawrath's implementation of the digital Theremin works considerably well for discretely controlling the pitch of a generated square wave, but it lacks the functionality of amplitude control that was available in the original analog Theremin. Volume control in an analog Theremin works exactly the same way as frequency control, except rather than being played directly, a band-pass filter is applied to the beat frequency of the fixed and variable oscillators, outputting a signal whose amplitude varies depending on hand position. By passing this signal through an envelope tracker and using the envelope as the control voltage for a VCA (voltage controlled amplifier), it is possible to control the volume of the Theremin's audio output.

Since it would require some type of digital to analog converter to adjust the amplitude of the generated square wave on the Arduino (a clumsy R2R ladder perhaps), we can instead sacrifice the ability to control dynamics for simplicity in hardware and settle for a binary means of controlling the amplitude of the audio output by simply turning it on and off based on some form of user input (be it an additional oscillator/frequency counter or something else). A simple button would work for this purpose and would allow the musician to articulate rhythms by depressing and releasing the button while still controlling the pitch of the generated tone with hand position. However, in the spirit of making the Theremin resemblant of actual musical instruments, we can add an exciting twist to the Theremino's musical interface by switching on and off the audio output by means of a breath controller, thereby transforming it into an

Airemino.

The simple breath controller in the Airemino works as follows. An active high-pass filter (with gain of 2 dB at high frequencies²) is applied to the output of an electret microphone. This active high-pass filter is identical to the inverting amplifier op-amp configuration, but with the addition of an AC-coupling capacitor in between the electret microphone's output and the inverting amplifier. The AC-coupling capacitor is necessary to eliminate the electret microphone's DC bias voltage and pass only the time-varying component of its output signal. This time-varying signal is then amplified by the inverting amplifier and fed into an envelope tracker, composed of a diode and a parallel RC circuit. The voltage across the resistor and the capacitor follows the envelope of the audio signal coming from the microphone, which is then used as a control voltage for the Arduino's audio output. A schematic of the breath controller is shown below.



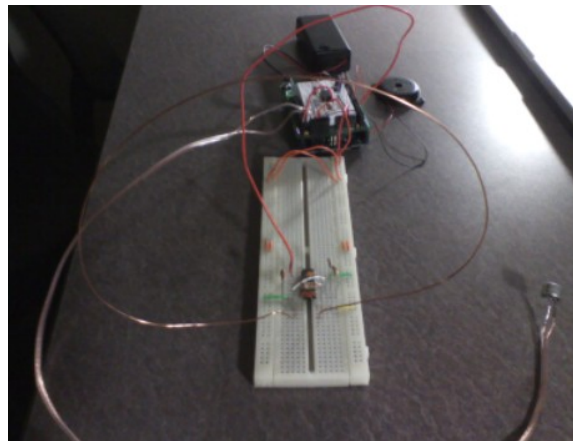
The active high-pass filter must have a gain of 2 dB at high frequencies since the electret microphone tends to output audio signals with maximum amplitude around 50 mV (about the DC bias). The 2 dB gain boosts this maximum amplitude to around 5 V, enabling the envelope

² In this case we are not concerned with the low frequency components of the electret microphone's audio output signal since the musician will be blowing into the microphone. Attenuating the low frequencies in this white noise input will not affect the final result; however, if we were instead interested in sampling an acoustic waveform (like speech or some other sound) the cut-off frequency of the high-pass filter would need to be tuned to our range of interest.

6

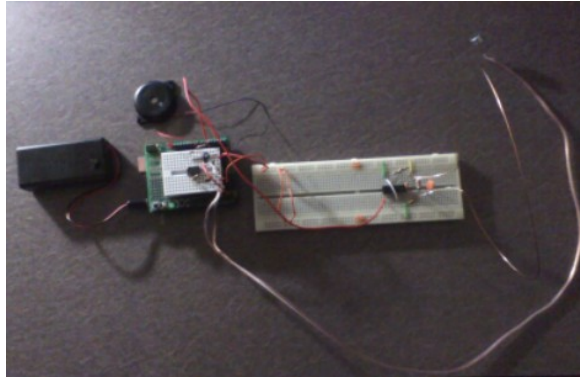
tracker to produce a control voltage between GND (0 V) and VCC (5 V) that can be read into the Arduino's ADC (analog to digital converter) and thresholded to determine whether to output audio or not.

All in all, the addition of the breath controller allows for the articulation of rhythms in a way that is familiar to woodwind musicians, while the capacitive sense pitch controller provides pitch selection in a way that is familiar to string musicians. When combined, these two musical interfaces allow for real unique opportunity in shaping sounds that is unavailable in most conventional instruments. Below are several pictures of the final implementation of the Airemino.

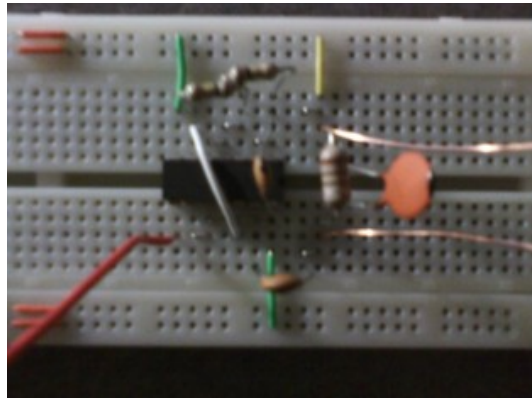


Seen from the front.

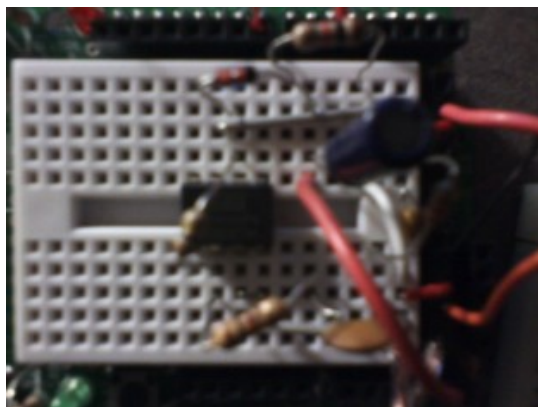
7



Seen from the top.



Nawrath's Variable Frequency Digital Oscillator



Homebrew Breath Controller

MIT OpenCourseWare
<http://ocw.mit.edu>

21M.380 Music and Technology (Contemporary History and Aesthetics)
Fall 2009

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.