uCHobby

Making things with Micro-controllers

Arduino Sound – Part 1

This is the first in a series of articles about generating sound with an Arduino. The goals are to generate good quality sound which can be used to play simple tones, stored music, sampled sound, and even act as a MIDI synthesizer. I will cover the basic design, including code and hardware that anyone can use to add sound to Arduino microcontroller projects.

The first step is to look what has already been done. We want to explore many possible options for sound generation and resolve which is best for given situations. I hope to spark some discussion so please comment with information links, advice, suggestions and general opinions.

Background
Sound is an important part of perception and an excellent way for a microcontroller to interact with humans (and other animals). It can be simple and inexpensive to implement. A microcontroller could do anything from generate tones, play music, or even output sampled sounds such as a dog bark. Next Halloween you could have a microcontroller project that yells "BOO" when someone walks up your sidewalk.

If you don't understand how sound works or what frequency means try reading up on the subject at Wikipedia. Here are a few links to get you started. Sound, Frequency, and Pitch.

Prior Work
I am not the first to work on this problem and it has been solved by others. I should point out that I am not a musician. I don't know how to read music or play any instrument. I just latched onto sound generation as an interesting project that would force me to learn.

There are a lot of really good results that come up with a simple search. I am sure that everything I will do has already been done better by others. I do want to reinvent this wheel, to learn and provide a useful resource for other hobbyist as I go. Here is a list of links I have been exploring. Feel free to comment with more related links.

Links to learn from
The shear volume of projects where people are making sound devices is daunting. I find so much to study that I dont know where to start. It also makes you wonder if there is room for one more article on the subject. These links dont even come close to covering all the information you will find with simple searches. Look over the projects at the Make Blog Music archive linked below. It will take me days to go though all that absorbing information.

- Discussion on Arduino Synth in the Arduino forum.
- Music Archive at the Make Mag site.
  Large collection of how-to articles related to music. Most are electronics related.
- Narbotic
  This is a great article covering Arduino sound generation. The author makes a keyboard controlled instrument with simple sound generation and an Arduino.
- Sound search at the Arduino site.
  This link does a Google search on the Arduino web site for sound.
- Quick and Dirty D to A on the AVR: A timer tutorial
  For many of my sound experiments I use an internal timer on the Arduino. This article gives some detail on how the timers work and is mostly targeted to PWM generation on the AVR. I will likely use this information to help me with the PWM audio experiment.
- Generating polyphonic sound with Arduino
  This article covers using external oscillators and really does solve the problem of audio generation with an Arduino. Multi voice sound for $5 or so without loading down the microcontroller.
- Synth Code from Paul Badger of Modern Device Company.
  Paul created a neat synthesizer instrument and has shared his design and code. This is direct link to Paul's source code described in the discussion linked above.
- Freqout function from Paul Badger
  This is part of the code from Paul's instrument linked above. By far the simplest way I have seen to get sound going on your Arduino. Just include this code and assign a pin for sound.
- The AvrX synthesizer
  This is a complete synthesizer using the Arduino. You can purchase the PCBs to build your own. And the software is open source. I did not see the schematic so I'm guessing the hardware is not open.
  Update: Jack reports that the hardware design is documented along with the source in the zip file.

- Midi-In for the Arduino
  Another great forum thread. The author provides a simple MIDI circuit and code for an Arduino to control LEDs based on MIDI note commands.
- Fun with sea moss
  Great article about generating sound using logic gates, not microcontroller based but a good read anyway.
- Fee, Fi, Fo, FM: Explore the World of FM Synthesis
  Explains FM Synthesis From Oreilly.
- Make a mellotron out of walkmans
  Bree Pettis and Eric Beug build a synthesizer like device from a bunch of Sony Walkmans. They also add an Arduino to control radio static to simulate drums.

Sound Generation Methods
Here is a list of everything I could think of for microcontroller based sound generation. Each method has a brief description and links to more information on the web. If I missed a method or you have a link suggestion let us know with a comment.

- **Bit-Banging**
  Toggle an I/O pin at the desired frequency. This will generate square wave tones which don't sound very pure. They can be filtered a bit to make them sound good at a specific frequencies.
    - The Good
        - This method is great for simple generation of tones.
        - Is almost free in terms of cost.
        - If a timer is used to do the pin toggling then the processor is not loaded and has time for other task.
    - The Bad
        - If the processor uses delay loops to toggle the pin at the desired frequency it will be very busy while the tone is active.
        - No amplitude control.
        - Square wave output does not sound pure. Its really the sum of all the odd harmonics at the base frequency.
        - Broad frequency coverage with simple output filters can result in significant amplitude variations across frequency.

- **Hardware PWM**
  Use a low pass filter on a timer controlled PWM output. Voltage of the final output signal is set by the PWM duty cycle percentage. Microcontroller updates the duty cycle to build the waveform. Basically the PWM function would act as a digital to analog converter (D/A or DAC).
    - The Good
        - Can generate quality audio and would be very inexpensive in terms of hardware.
        - Should allow for simple power amplification as the amp can be class C with the speaker and output circuit doing the filtering.
    - The Bad
        - The default Arduino analogWrite functions are not adequate for sound generation because the base frequency is about 500Hz.
        - We need to take over the PWM timer output and increase the frequency. This might result in reduced resolution.
        - The microcontroller would need to update the PWM output very quickly to generate a waveform for the sound. For example, if we wanted a sin wave output the microcontroller would have to update the PWM pin many times for each cycle of the desired frequency.
        - Significant amounts of memory are required to store waveforms.
    - The unknown
        - I have not done the experiment yet. I plan to hijack a timer and do PWM control without the built in Arduino functions. The rate needs to be increased to well above audio, say 50KHz. I have not checked into what resolution would remain for duty cycle control. If all

goes to plan this could be the best solution for high quality sound output but I don't see it being used much so there must be some problems.

- **R2R DAC**
  Use a simple resistor array to build up the voltage based on high or low I/O pins on the uC. You need two resistors and an I/O pin for each bit of resolution. I have used a 6 bit (12 resistors, 6 I/0) R2R for most of my experiments so far.
    - The Good
        - Simple design
        - Low cost, just two resistors per bit of resolution
        - High quality waveforms
        - Software can do internal mixing of sounds to create multi-voice outputs.
        - Can do sampled sound such as "Boo". Possibly even playback at different frequencies so that the "boo" sound can be played like musical notes.
    - The Bad
        - Requires one I/O bit per bit of resolution. A six bit converter uses 6-digital I/O lines.
        - To generate frequencies the processor must update the output value many times per cycle of the desired waveform. This keeps the processor busy
        - To be accurate a timer interrupt is needed
        - Significant amounts of memory are required to store waveforms.

- **Hardware DAC**
  This is basically the same as the R2R DAC except a chip is used that contains the R2R network.
    - The Good
        - All the goods from the R2R DAC
        - A serial DAC chip would reduce the I/O pin count but might hurt in terms of processor resources. It could take significant instruction time to send the DAC updates.
        - Since the hardware DAC has a built in register, you can use the I/O pins for other multiplexed purposes.
        - By adding one more I/O pin, you can have another DAC. The extra I/O pin is used to select which DAC you update. It is simple to have Dual outputs, just use a dual DAC.
        - Most DAC chips have an analog buffer on their output which makes it much easier to do a good output filter and dive other inputs.
        - Can do analog multiplication which is nice for mixing signals.
    - The Bad
        - Same as with R2R DAC
        - Cost more then an R2R network if you don't need multiple outputs and output buffers.

- **Voltage Controlled Oscillators**
  Use the PWM outputs or some other method to generate a voltage which controls the frequency of an external oscillator. This is the method used in some of the other synths linked in the resources above.
    - The Good
        - Frees the processor from updating the waveform shape as required by the DAC methods.
        - Generates smooth waveforms which sound great.
    - The Bad
        - Not as versatile as the DAC methods.
        - Requires external parts to make the oscillators and the control circuits such as serial controlled potentiometers.
        - Tone accuracy could be a problem.

- **Synthesizes Chip**
  Chips designed for generation of sound. These chips are used on sound cards and PC mother boards. They provide multiple voices and have very sophisticated sound generation features.
  - The Good
    - Lots of voices
    - Ability to generate different instrument sounds
    - Keep the processor load down
  - The Bad
    - May not support playback of sampled sounds.
    - Could be complex to control from software.
    - May require lots of parts to support the chip.
    - Probably cost more then DAC methods.
    - Probably only available in surface mount packages.
  - The Unknown
    - I have not done much research on these chips. I don't know what is available or how much they cost.
    - I don't know how hard they are to control.
    - I hope to receive some comments with suggestions for chips to try.
  - Some Devices to research
    I would like to get my hands on these devices to see how well they work. Write a comment to suggest any others.

    - Soundgin – sound chip in a PIC
      This is apparently a PIC uC with firmware to make a synth chip. It's serial controlled and can even do speech. Check out the sound samples at the link above.
    - Yamaha YM2612
      I found this chip while looking for an image on Google. The specs look good and the picture shows a leaded device. I have not dug up the datasheet or a possible supplier.
    - SARA-001
      This is not really a chip but would interface to the uC.
    - SAA1099
      Chip used on older Sound Blaster cards, might be able to salvage one. The SARA-001 uses this chip so they may still be available somewhere.

- **MP3 Decoder Chip**
  Basically make a MP3 player that plays multiple files based on program control. The Make Daisy MP3 player is an example. There are lots of MP3 player projects described on the net.

  - The Good
    - Playback any MP3 file.
    - Sound processing done by external device
  - The Bad
    - Could cost more then other methods
    - The IC would likely be a surface mount component
    - The microcontroller would have to spend time sending data to the device
    - May not support generation of sounds based on inputs. You could do things like play a MP3 file when an event was detected but it would be hard to make something like a piano keyboard or to vary frequency as a function of some sensor input.

- **Use external MIDI device**
  Use an external device or a PC with a MIDI control input. When the microcontroller detects an event it could trigger a drum beat or an electric guitar strum.
  - The Good
    - All sound processing is done external of the uC.

        - Broad range of sound. Everything form a realistic piano to a drum set would be easily available.
        - Easy to record the data stream for playback later.
        - Could use your PC or an old spare PC for sound generation.
    - The Bad
        - Need a MIDI interface on the uC.
        - Need an external device that can handle MIDI
        - Would cost more if you don't have a free MIDI device you don't mind dedicating to your project.
    - The Unknown
        - I have not played with MIDI so I know very little about it. I like to learn though
        - Don't know how to build, send or receive MIDI. Initial research shows that there is a lot of information on the net so my ignorance should be easily curable.
        - Interface parts and code for the Arduino.

- **Control an external device**
  Use the uC to activate or control an external MP3 player or even a cassette player. The maker crew did this with a bunch of tape players controlled from an Arduino, check this out here. Circuit bending under uC control would be another form of this type of sound generation. Might be easy to send serial messages to your PC and have it generate the sounds for example.

    - The Good
        - Easy to play back sampled sounds. You could easily have a tape loop that greets visitors for example. The uC could detect that someone was at the welcome matt and activate the loop.
        - If the external device was a recorder, it would be easy to customize sounds/messages.
        - The external device might have built in speakers.
    - The Bad
        - You have to have a sound device that you can hack.
        - Depending on what you need, you might have to have several external units. To act like a piano for example, you would need a bunch of tape players, one for each note.
        - Interfacing could be difficult. You have to simulate button presses and sometimes those require mechanical action.

- **Simulate a human using a real instrument**
  Use the uC to emulate a human playing the instrument. For example, you could rig up some electro mechanical solenoids to press the keys on a piano or to beat a drum. This has been done for many instruments and is one of the coolest things you can do.
    - The Good
        - Usually perfect reproduction of the instrument sound. After all what better way to make a drum sound then to play a real drum
        - Very cool so it has great presentation value.
        - Great for simple uses, would be fairly easy to hit a large bell for effect.
    - The Bad
        - You have to rig up a real instrument
        - Electro mechanical components can be expensive and difficult to deal with.
        - Your project will require much more space.
        - Might be hard to contain in a single package.

What's Next
For the next installment of Arduino Sound we cover getting the sound out of your microcontroller. We do a "Sound Hello World" and get our breadboard connected to a speaker so we can hear the sounds. Future articles will cover the methods described above and will introduce some advanced programming methods. I plan to abstract the hardware DAC in software and possibly the Synth code as well. More about this later.

Comments Please
I would like to hear your thoughts on this article and sound generation in general.

- Have I missed a method that should be covered?
- Have any of you used software that turns the sound card input into an O-Scope?
- Sound editor programs?
- Ever done MIDI with an Arduino, where do I start?
- What program should I use for MIDI on a PC to control from the uC or to receive commands from the uC?

Posted in Discovering, Microcontroller, Projects.

26 comments

By uCHobby – November 11, 2007

---

26 Responses

Stay in touch with the conversation, subscribe to the RSS feed for comments on this post.

1. *biojae says*

   What about a hybrid solution?
   like PWM to a 4-6 bit dac?
   Source: http://www.k9spud.com/traxmod/pwmdac.php

   ```
   pwm0 -----[10k]----.
   pwm1 -----[220]---.|
   pwm2 -----[120]--.||
   pwm3 -----[50]--.|||
                   ||||
                  +++++---(spkr)-|
                                 ---
                                  -
   ```

   April 5, 2008, 5:53 PM
2. *DrLucas says*

   Hey all, i was wondering if anyone has used an arduino to play piano key notes(the whole piano) and if so how did you do it? If not, did you make something else that could be rigged the same way, fyi, looking for a low cost solution.

   I have an old piano that is non-electric and want to rig it to play electrical notes when a key is pressed. I want each key to complete a circuit when pressed and then the arduino send a signal to a speaker to play the sound, or send a signal to something else i might need to then send the sound… i read most of the things in this and have not really seen anything that i thought would work the way i needed, but then again, some stuff i just did not understand 🙁

   Thanks in Advance,
   Daniel
   DrLucas17@yahoo.com

   April 3, 2008, 9:39 PM
3. *dfowler says*

   Hui,

   That sounds like a great idea. You could scrounge a connector from an old MB. It should be fairly easy to interface to the old ISA standard. I wonder how many old sound blaster cards could be recycled into

MIDI synths using an AVR to work the magic between the MIDI and the ISA bus…

March 12, 2008, 1:43 AM

4. *Hui Hai says*

I'm thinking about use old ISA soundblaster awe32 to generate sounds. It has goor wavetable synthesizer mpu8000 that can generate 32 channels. AVR(orPIC) controls the ISA bus. Gravis card is better documented but hard to find?

March 11, 2008, 8:20 PM

5. *manibabu says*

i realy like the project u r submitted….one important think is ur software details r not enought to create complete project…so i want more information on creating different sound wave using PIC16F785 with the help of inbuild PWM.
thanks…

February 25, 2008, 7:06 AM

6. *dfowler says*

Mike,

I loaded up your code, it sounds great! Very hypotonic.

I modified the code so that it plays continuously with a phase increment control. This way I can effectively resample the waveform like a DDS does to control output frequency.

Your sound does not loop well; I did adjust the length to get a better loop sound. It would be neat to use the Futurama Hypno-Toad sound for this. Maybe make the sound vary based on some input like light level.

I will send the code if your interested.

December 31, 2007, 2:35 AM

7. *Mike says*

Hey, I had some luck with the hardware PWM trick you talked about.

http://www.arduino.cc/playground/Code/PCMAudio

December 30, 2007, 4:16 PM

8. *Gian Pablo says*

I've been using a hybrid approach, using serially controlled digital potentiometers and voltage-controlled oscillators.

It gives me 6 quite good-sounding voices, with very little processor load.

I'm planning to package up the hardware as an Arduino shield, or as a dedicated Arduino-compatible audio package.

More info here: http://itp.nyu.edu/~gpv206/2007_fall/arduino_audio/

November 21, 2007, 2:26 PM

9. *Craig says*

What about the WinBond ChipCorder? Used them in a talking appliance project, and they worked quite well. Not really a solution for generating sound, but you can record onto them and playback, all controlled by I2C or SPI protocols. Pretty much plug and play, even has features to manage the memory addresses of your messages. Seems to have been designed for telephone answering machines.

http://www.winbond-usa.com/en/content/view/36/140/

November 14, 2007, 2:37 PM

10. *dfowler says*

    Wow, so much good information to explore. Thank you Trumpetto, I will study OSC.

    November 14, 2007, 12:27 PM

11. *Trumpetto says*

    The MIDI program I would use is PD as you can very easy create small monitoring apps and test apps. I don't know any free and good sequencers for the PC (I us a mac).

    At the Open Sound Control site I gave, under "OSC Implementations" you can find many source files for using

    November 14, 2007, 8:39 AM

12. *dfowler says*

    follower,

    Yes it does… kinda steals the steam from my project LOL

    November 14, 2007, 3:13 AM

13. *follower says*

    Wow, the example from the comment on the Make blog post sounds incredible:

    http://elm-chan.org/works/mxb/report_e.html

    –Phil.

    November 14, 2007, 3:09 AM

14. *pK - BricoGeek.com DIY in Spanish version says*

    wow, im surprised to see what arduino can actually do. I will try this soon for sure!

    November 13, 2007, 10:54 AM

15. *Steve Dickie says*

    # Have any of you used software that turns the sound card input into an O-Scope?

    I use a free program on windows called Visual Analyser.
    http://www.sillanumsoft.org/

    November 13, 2007, 5:02 AM

16. *Jack says*

    The AVRX Software and Hardware are both open. Download the source code zip, and look in the "/AVRX/doc" folder. It's all in there.

    Now, why they don't make that more clear on their website is beyond me. Most people would think a zip file labeled "source code" would contain only code, not drawings and schematics as well.

    November 13, 2007, 2:01 AM

17. *Alan Parekh says*

    Fantastic topic, love all the juicy links. 🙂 I will be back when I have a bit more time to read the great collection you put together!

November 13, 2007, 1:42 AM

18. *dfowler says*

    Trumpetto,

    Thanks, I revised the “Control an external device” based on your “using a PC” comments. This is a good idea and I will give it a try.

    I do want some kind of MIDI synth program to run on a PC. I want to control it from the Arduino. I also want a control program for the PC that can control an Arduino synth over MIDI.

    I had looked at OSC in the past, when I was playing with the Make Controller Kit. Do you know of some source code to get started with the string decoding necessary for that?

    November 12, 2007, 6:23 PM

19. *dfowler says*

    Vic,
    The bit shifting sounds like a good way to simulate multi-voice with the Bit-Bang method. Is that what you were thinking? I would like to give this a try. Maybe pre-calculate some multi-voice sounds into 32 bit values then shift them out to the I/O pin.

    Or maybe you were thinking that one of the serial output functions on the uC could be used. Hummm Or maybe even just use an external shift register.

    Wow, there really are a lot of possibilities for sound.

    November 12, 2007, 6:17 PM

20. *Trumpetto says*

    Q: Have I missed a method that should be covered?
    A: Yes, you can very well control sound generating programs on your pc from the arduino. Think of pd, Max or DIY in Python for example. You don’t need MIDI to do that.
    http://www.arduino.cc/playground/Interfacing/PD
    http://www.arduino.cc/playground/Interfacing/SuperCollider

    Q: Sound editor programs?
    A: Audacity. It’s very good, free and cross platform. I guess you don’t mean the kind of programs that are used for commercial synths…

    Q: Ever done MIDI with an Arduino, where do I start?
    A: Start with a simple MIDI OUT to your PC or a synth.
    http://itp.nyu.edu/physcomp/Labs/MIDIOutput
    MIDI is an old and slow protocol but that also means that it is easy and gives quick succes.
    Don’t forget to have a look at OSC (Open Sound Control).
    http://www.cnmat.berkeley.edu/OpenSoundControl/

    Q: What program should I use for MIDI on a PC to control from the uC or to receive commands from the uC?
    There are many options; Pd, Keykit, any sequencer.

    November 12, 2007, 3:52 PM

21. *vic says*

    Regarding the Bit-Banging option, a trick can be to set the series of bits on a byte, and then shift the byte on one pin. It can reduce greatly the time spent in loops. It only works for simple sounds, though.

    November 12, 2007, 2:39 PM

22. *tateu says*

hey, great start. you have gathered all the great resources available that i know of. i am working on AVR sound generation myself. so far, i'm most interested in the R2R approach with timer interupt. i've been able to get a sound as analogIn and reproduce it on the R2R. the quality isn't great (4 bits, very choppy and saturated output). the timer is running at about 7khz at the moment. more to be done on my end. looking forward to your next posts. cheers.

November 12, 2007, 12:40 PM

23. *Nigel Crawley says*

oops – I see you've already covered that idea 🙂

November 12, 2007, 12:32 PM

24. *Nigel Crawley says*

What about using the Arduino to control a circuit bent device – e.g. a circuit bent speak and spell toy?

cheers

Nigel

November 12, 2007, 12:27 PM

25. *follower says*

> Have any of you used software that turns the sound card
> input into an O-Scope?

Yes, see here:

http://code.rancidbacon.com/LearningAboutXoscope

–Phil.

November 12, 2007, 9:58 AM

Continuing the Discussion

1. *buy hgh 1000* linked to this post on March 25, 2008

    buy hgh 1000

    buy hgh 1000

« Arduino AVR In System Programmer (ISP) Arduino Sound Part 2: Hello World »

Subscribe

Site RSS Feed

About uCHobby

uCHobby's how-tos, making things, hacking, and experimenting with microcontrollers. Featuring the Arduino, and other microcontrollers.

Archives

- August 2016

- February 2016
- January 2016
- September 2015
- June 2015
- May 2015
- February 2015
- October 2014
- July 2014
- June 2014
- May 2014
- June 2013
- January 2012
- December 2010
- August 2010
- May 2010
- March 2010
- February 2010
- January 2010
- October 2009
- September 2009
- August 2009
- July 2009
- June 2009
- May 2009
- April 2009
- March 2009
- February 2009
- January 2009
- December 2008
- November 2008
- September 2008
- August 2008
- July 2008
- June 2008
- May 2008
- April 2008
- March 2008
- February 2008
- January 2008
- December 2007
- November 2007
- October 2007
- September 2007
- August 2007
- July 2007
- June 2007
- May 2007
- April 2007
- March 2007
- February 2007
- January 2007
- December 2006
- November 2006
- October 2006
- September 2006
