

TINKER

tinkertit

Open source releases from TinkerLondon and Tinker.it

Search projects

[Project Home](#)

[Downloads](#)

[Wiki](#)

[Issues](#)

[Source](#)

[Export to GitHub](#)

Search

Current pages ▼

for

Search

Auduino
Details of the Auduino synthesiser
[auduino](#), [arduino](#), [Featured](#)

Updated May 1, 2010 by [cathed...@gmail.com](#)



Introduction

The Auduino is a sound synthesiser based on the [Arduino platform](#). It works on all Arduinos running at 16MHz - everything from the original Arduino serial to the Arduino Mega. It uses granular synthesis techniques to generate a distinctive filter-sweep sound that had much more character than boring square waves.

Have a look and a listen

- [The first Auduino](#)
- [Aiden's Auduino](#)
- [xndr77's beautifully cased Auduino](#)
- [Medaispoca](#)
- [Berit Greinke](#)
- [Adam Franchino's Ant Farm](#)

Synthesis model

Sound is generated by playing the same noise ('grain') repeatedly at very high speed. This merges into a tone that is an audible hybrid of the repetition rate and the original grain. It sounds quite similar to an oscillator with two resonating bandpass filters, although the different architecture means there are lots of additional interesting noises at parameter extremes.

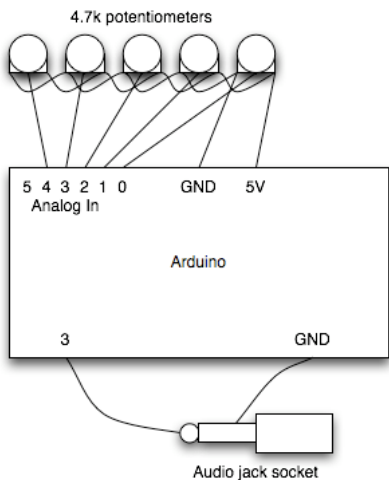
The grain consists of two triangular waves of adjustable frequency, and adjustable decay rate. This is based on FOF synthesis model, but using triangle waves instead of sine and using a rectangular window.

The repetition rate is set by another control.

Programming the Arduino

Download the source code from the the [Tinker.it Google Code site](#). Load it into the Arduino environment, set up your board type and serial port, then hit the Upload button. Done. If you've never used Arduino before, you'll need to download the development software from [the Arduino site](#).

Construction



Auduino uses 5 controls. You can use anything that generates a 0-5V analogue signal, but the prototype uses five 4.7Kohm linear potentiometers. Connect one side of each potentiometer to GND. Connect the other side to the 5V pin on Arduino. Connect the middle (wiper) pins to Analog inputs 0 to 4 on the Arduino.

The audio comes out of Digital pin 3 (or pin 11 on ATmega8 Arduinos). The prototype uses a 1/4" jack socket, with the tip connected to pin 3 (or 11 on ATmega8's) and the shield connected to GND. Plug the other end into an audio amplifier, and you're good to go.

The Arduino can drive a small piezo, speaker or headphones directly. Strictly speaking it outputs at 5V rather than the 1V line level, but most amplifiers don't seem to mind.

Auduino community

We now have a group on [Google Groups](#). Show off your hacks, or get help with your Auduino projects.

Auduino controller hacks

The Auduino can take any analogue signal and make it audible. Add a [Light Dependent Resistor](#), you have an instant theremin. Add a [linear softpot](#), instant keyboard or ribbon controller. The [Arduino Playground](#) is a great place to get new ideas.

Auduino software hacks

Have a look at the loop() function. The controllers are mapped to the synthesiser parameters there. You can really customise the controllers there - adding different musical scales, vibrato, envelopes - whatever takes your fancy.

Start by playing with the pitch mapping. Three mappings are available to start with. Uncomment the one you prefer:

```
// Smooth frequency mapping
//syncPhaseInc = mapPhaseInc(analogRead(SYNC_CONTROL)) / 4;

// Stepped mapping to MIDI notes: C, Db, D, Eb, E, F...
//syncPhaseInc = mapMidi(analogRead(SYNC_CONTROL));

// Stepped pentatonic mapping: D, E, G, A, B
syncPhaseInc = mapPentatonic(analogRead(SYNC_CONTROL));
```

syncPhaseInc is proportional to frequency, so the /4 above drops the pitch to a quarter, or two octaves.

Auduino hardcore hacking

If you're really into algorithmic synthesis, have a poke around in the interrupt routine at the bottom of the source code. You'll have to be careful to keep the routine fast, but there is plenty of spare CPU to implement other synthesis techniques.

Comment by [brian.de...@gmail.com](#), Nov 18, 2008

Just making sure, are the Pots Linear? Cheers Brian Degger

Comment by project member [cathed...@gmail.com](#), Nov 18, 2008

Very good question. Yes they are - and I've updated the text. Thanks Brian!

Comment by 3vl.0ve...@googlemail.com, Nov 18, 2008

Just attach a few buttons over some resistors instead of the pot that controls the freq and you got a keyboard ;-)

Comment by [casai...@gmail.com](#), Nov 18, 2008

Congratulations for this project!! Looks very good :-)

Comment by [jesse.fr...@gmail.com](#), Nov 22, 2008

Just breadboarded this and am playing around with it. With a little experimentation on the controls, it sounds awesome! Can't wait for my softpot to arrive and then I can get some keyboard action going...thanks for sharing this project!

Jesse

Comment by [xnd...@gmail.com](#), Dec 7, 2008

Very nice project! I built one myself to demonstrate the protomodule / arduino enclosure project I've been working on:
<http://www.flickr.com/photos/machinecollective/sets/72157610797368431/>

I think I spent almost 2 hours playing with this baby.. It's amazing! Thanks and keep up the good work!

Comment by project member [cathed...@gmail.com](#), Dec 8, 2008

Love it xndr77! Great video, and love the enclosure. Linked up the video for all to see.

Comment by bas682...@gmail.com, Dec 9, 2008

i love the sounds that come out of this, and would like to explore the use of this as a keyboard driven synth. i noticed that one person talked about buttons over resistors to make it a keyboard...

i have seen another arduino keyboard before, but, the sounds that come out of the auduino are much more pleasing to my ear... i also thought about the use of a softpot... are you able to set the pitches to a range of resistances in the software? thanks!

Comment by project member [cathed...@gmail.com](#), Dec 11, 2008

@bas682009: One thing to watch with a keyboard - it controls two things - the pitch of the note, and also if a note is playing. Auduino right now allows pitch control, but not note gating. Of course, you've got the complete source code so there's nothing stopping you implementing that. You might want to consider [ADSR](#). As for mapping pitches to resistance: you need to map the keyboard to voltage. I'd look at using a potential divider for that, as a

chain of same value resistors. Depending on how you organise the chain, you may need to modify the ADC reading to pitch mapping in the loop() routine. Have a look at the source code.

Comment by p00...@gmail.com, Dec 12, 2008

Awesome project! I have been playing with this code and hooked it up so it is controlled from windows through serial rather than the pots. Not as usable as physical pots but it was just something to play with.

once again, great work!

@bas682009: I have changed it so that the values that are used to generate the sound are stored in variables and those variable are updated via software through serial, so yea it can handle the storing the values. I dont even notice any delay when it plays back even though it is monitoring the serial and doing some basic logic from it.

Comment by project member cathed...@gmail.com, Dec 13, 2008

@p00b0x: Glad you're having fun.

This comments section is getting a bit long, so I've created a [Google Group](#)

That should make things easier when sharing source code, diagrams and such.

Comment by lifeisno...@gmail.com, Feb 7, 2009

Very cool project, I just got mine working. My kid loves it too. <http://noiseislife.tumblr.com/>

Comment by goatboy...@gmail.com, Mar 14, 2009

"although any Arduino running at 16MHz will work fine" Does this mean that you can't operate Auduino if your using an Arduino mini pro that operates at 3.3v/8MHz? A newby getting to grips with the Arduino Universe!!! Brilliant project by the way!!, Robbie

Comment by project member cathed...@gmail.com, Mar 14, 2009

@goatboyrobbie: It could work with 8MHz devices, but... It would run an octave lower. Also, the PWM would run at half the rate, so the 'whistle' of PWM would be much more obvious. Try it - it will work as is - but I think you'll prefer running it on a 16MHz device.

Comment by goatboy...@gmail.com, Apr 10, 2009

Thanks Cathedrow! I have a couple of mini pro's running at 5v now so it won't matter but on another issue I can't seem to get any real decent sound out of this Auduino! I've checked the wiring and everything's loaded perfectly but it seems very tricky adjusting the pots to hold on to the melody and certainly can't get anything close to that fantastic video demo!! I just wonder where I'm going wrong? Any body else finding that tricky or is it just me? Appreciate any advice, Goatboy

Comment by goatboy...@gmail.com, Apr 11, 2009

All sorted!! Finally got a grip on this little baby!! Between the pentatonic scales and the filter sweeps it'll put you right up there with the old Mongolian polyphonics!!!

Comment by dsi...@gmail.com, May 11, 2009

Thanks Peter K. for your help at openhacklondon. We rushed managed to assemble your circuit, in the end because it was "passive" we had to add some more resistors but got a working prototype in time. Cheers The pedal steel guitar tone control hackers

Comment by baadspe...@gmail.com, Aug 15, 2009

dude where can i find the code for "Synthesis model" i searched the site and nothing :S

some one pease link me ? / email me @ baadspella (@) gmail.com

Comment by jesse.j...@gmail.com, Aug 18, 2009

I made one too!

Here's a video of the build: <http://www.youtube.com/watch?v=hwV-SKpQAak>

Next step is trying to build and add a ribbon controller.

Comment by marcelij...@gmail.com, Aug 24, 2009

Great that you share this, easy to understand. Thanks

Comment by deadastr...@yahoo.co.uk, Nov 10, 2009

could this have a guitar input!...from a pot or something!..to drive the sound?

Comment by project member cathed...@gmail.com, Nov 10, 2009

@deadastronaut Sounds like you'd better get building. Post it up once you've got it working!

Comment by jwhagenb...@gmail.com, Nov 23, 2009

can i hang it on a baby10 sequencer? i am a little afraid to put an extra 5V to the arduino. so i mean an external cv to control the unit...

Comment by farmergl...@gmail.com, Dec 11, 2009

Absolutly excellent sounding device - just finished mine - would love to know where to start if I want to add some effect routines (distortion maybe) to the code - perhaps change one of the knobs to overload the sound - any ideas ?

Comment by farmergl...@gmail.com, Dec 15, 2009

Having just banged together a simple contact mic - replacing a couple of POTS with these might be worth a simple hack !!

Comment by sonom...@gmail.com, Dec 18, 2009

Here's mine little auduino makin glitchy soundz))

<http://soundcloud.com/kuyanov/dds-test1>

thanx!

Comment by animazon...@gmail.com, Dec 30, 2009

hey sonomute really nice glitches how did you made them????

Comment by farmergl...@gmail.com, Jan 24, 2010

Working on syncing a Baby10 to control all/none of the pots - and maybe have a LDR hooked in also - this project has thrown up many ideas - thanks !!

Comment by ryan.p.c...@gmail.com, Feb 1, 2010

I had a question about the potentiometers. Do I have to use 4.7k ohm pots or could I use 10k or even 100k pots? I priced out some 4.7k ohm pots and they were much more expensive than 100k pots.

Comment by mark.kiz...@gmail.com, Mar 6, 2010

Hi All! I just finished building my Auduino. It was a great experience! Check out my project: <http://turtlethink.com/2010/03/auduino-diy-arduino-synthesizer/>

Comment by joseph.p...@gmail.com, May 9, 2010

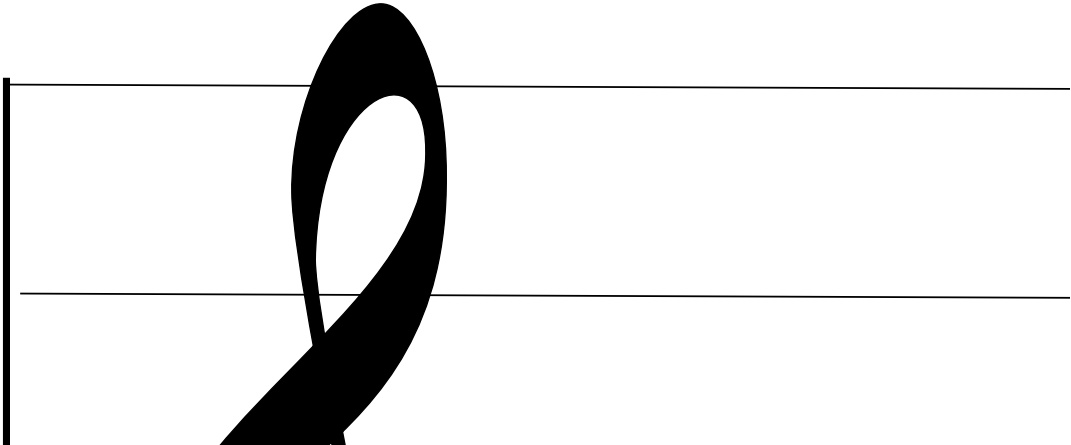
This is a ton of fun. Is there a way to change the key it plays? Or can you tell me how to come up with the equation for cminor? This is a wonderful thing for arduino, and people should realize the power you have unlocked..

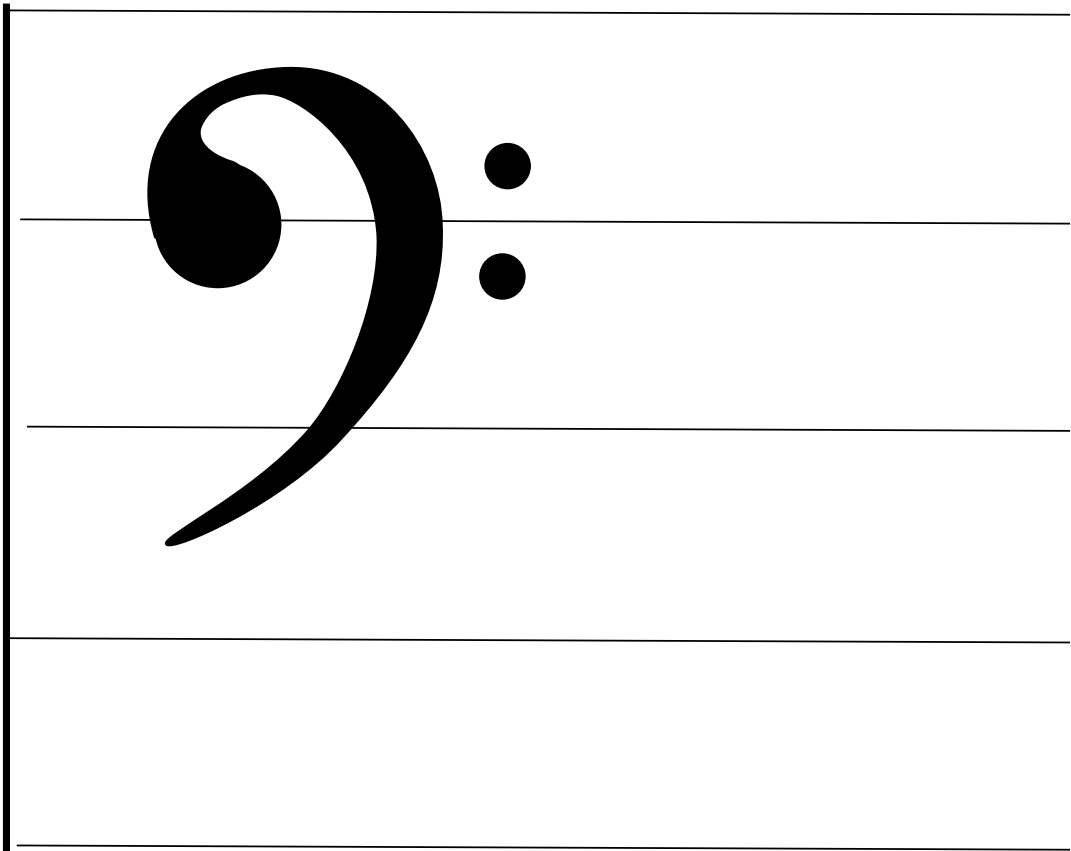
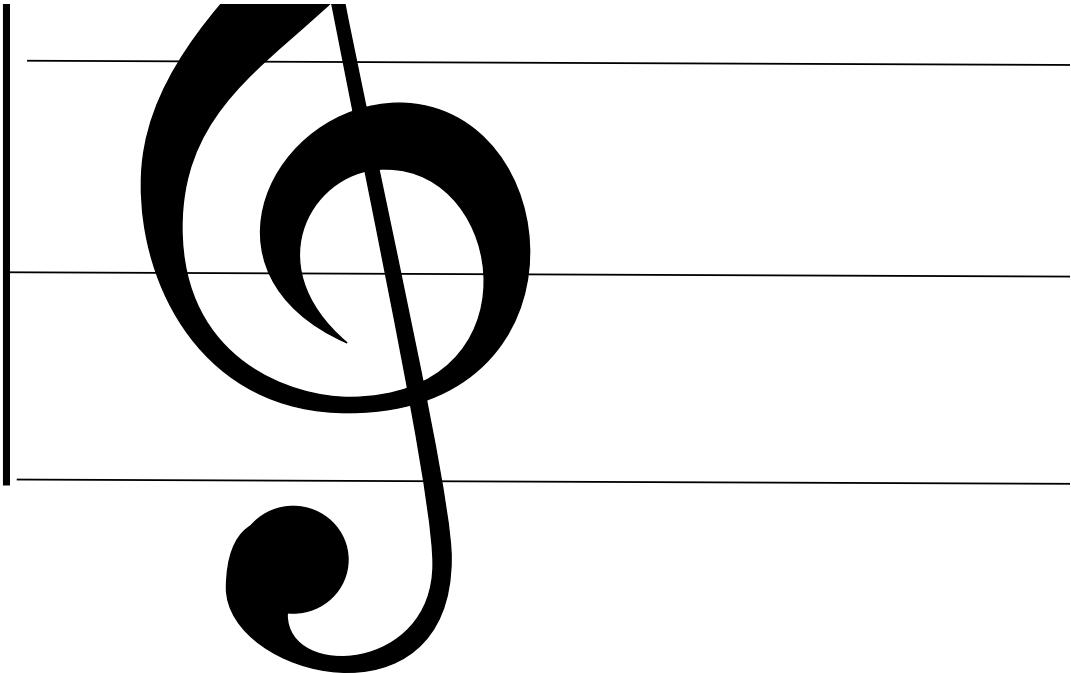
Comment by opensource@till.name, Aug 23, 2010

The value in the array for a tone with freq should be something like: $65536 / ((62500/\text{freq})/2)$

62500 comes from the PWM frequency, i.e. it is the frequency with which the PWM_INTERRUPT is called and 65536 is the amount of values for syncPhaseInc. The Arduino Physical Computing book by Manuel Odendahl, Julian Finn & Alex Wenger helped me to understand this. But it does not really explain this.

You can also use the values from the midi table, a mapping from the midi note number to the actual note is available here:





E.g. the 69th value (c' in midi) in the table is 923. But I did not yet fully understand the code. It might be, that the values are actually off by an octave.

Comment by project member cathed...@gmail.com, Aug 26, 2010

@michael.burzycki Arduino is basically C with some extra libraries. (There is a little C++ in there too, but Auduino doesn't use that). `#include`, `>>` and `-=` are all standard C constructs. I urge you to get a copy of "The C Programming Language" by Kenighan and Ritchie. It's a slim book that not only teaches C - it effectively defines it. With that by your side things should make a lot more sense.

Comment by kog...@gmail.com, Nov 30, 2010

hey @cathedrow,

i'm building something like this for my electronics class, i was thinking of substituting the potentiometers (or some of them) with force sensors or piezo elements. my idea is to make an enclosure with the sensors on the walls, and when you squeeze the enclosure wherever the sensors are taped, cool noises are made. the only problem is that force sensors are a little more expensive. so i wanted to know if you think this could work? thanks so much for your help! i look forward to a response!

Comment by brett...@gmail.com, Dec 10, 2010

Has anyone gotten this to work successfully on a MEGA 2560? I followed everything exactly and get no output at all. I verified all components to be good as well as the MEGA 2560 and all associated ports. Thanks in advance!

Comment by matstev...@gmail.com, Dec 16, 2010

I'm having the same issue as brett... The LED blinks twice but no output. And I've double checked everything.. twice. Any ideas? Thanks!

Comment by bruno.s....@gmail.com, Jan 11, 2011

Hey guys check mine @ <http://www.youtube.com/watch?v=DY-8CubNmio>

Tks for the project

Comment by OrrinFra...@gmail.com, Jan 19, 2011

WE NEED VOLUME CONTROL!!

Comment by TannerTe...@gmail.com, Feb 20, 2011

The only output coming from it is from Analog pin 4, witch makes the pitch high or low. Help please!

Comment by parafern...@gmail.com, Jun 8, 2011

Hi!!!thank you to share it!!! I built the synthesize, but now i would like to connect the potentiometer to the mouse, to interact with my screen, any idea????? thank you!!!!

Comment by lemore...@gmail.com, Oct 3, 2011

Hi, I added 2 LDR parallel to pot, funny!

<http://midisurfing.blogspot.com/>

Comment by msowa2...@gmail.com, Oct 11, 2011

Hello, great project.

Please see my demo video of it;

<http://www.youtube.com/watch?v=6-NODXBWIBc>

Comment by juanpa...@gmail.com, Nov 12, 2011

in the construction of any side is, how to connect the SoftPot?? Membrane Potentiometer - 100m? as would be the connection

<http://www.sparkfun.com/products/8607>

http://www.youtube.com/watch?v=q9tmbrG7D-o&list=FL8zXRt0tGap8EAVUai6qY4A&index=16&feature=plpp_video

Comment by jepZy...@gmail.com, Jan 18, 2012

Sensors are transforming the way engineers think about position sensing

Distance Sensor

Comment by mr.dat...@gmail.com, Feb 20, 2012

Is this page still open?

I am trying to get auduino v5 to work with a uno.

I can only get pot in analogue 4 to work. The others have no real effect or too small to notice.

I have checked my pot wirings (in serial monitor) and they all work correctly.

The tone from pin3 is noisy. If I plug into pin 13 I get a pure tone. No other pins give me a tone.

I have a 1W 8ohm speaker with inline 100ohm resistor wired up and have tested it on its own using the Tone() lib.

Is this because it is incompatible with the uno?

Comment by EMS5...@gmail.com, Mar 5, 2012

I'm having the same problems as the above poster with a Mega... any solutions?

Comment by Dr.Brian...@gmail.com, Mar 6, 2012

Does this software work with a Arduino Mega 2560 REV3? It is the only one available locally..Thanks in advance. Here's hoping it does...

Comment by DuaneSBa...@gmail.com, Apr 8, 2012

Hi,

I to those having problems, I have tested the code in Arduino 1.0 and it requires the volatile keyword to be added to the variable declarations in order to work. My guess is that a compiler optimisation is replacing the variables with constants inside the ISRs, volatile fixes this. I am only breadboarding the circuit at the moment but will post my suggest for the fix once I have a permanent circuit up and running. Note - I am not part of the Audino team, nor have I contributed in anyway, I just think its an amazing little project and hope this suggestion can help some others.

Duane B

rcarduino.blogspot.com

Comment by benw...@gmail.com, Apr 23, 2012

@Duane, Do you have any update on the fix?

Thanks! Ben

Comment by [DuaneSBa...@gmail.com](#), Jun 16, 2012

Hi, Yes I can confirm that the fix is to simply to change the first few lines from this -

```
uint16_t syncPhaseAcc; uint16_t syncPhaseInc; uint16_t grainPhaseAcc; uint16_t grainPhaseInc; uint16_t grainAmp; uint8_t grainDecay; uint16_t grain2PhaseAcc; uint16_t grain2PhaseInc; uint16_t grain2Amp; uint8_t grain2Decay;
```

to this -

```
uint16_t syncPhaseAcc; volatile uint16_t syncPhaseInc; uint16_t grainPhaseAcc; volatile uint16_t grainPhaseInc; uint16_t grainAmp; volatile uint8_t grainDecay; uint16_t grain2PhaseAcc; volatile uint16_t grain2PhaseInc; uint16_t grain2Amp; volatile uint8_t grain2Decay;
```

All that we are doing is adding the volatile keyword to the variables which are accessed by both the interrupt service routine and loop. This prevents the compiler from optimising them. Without this keyword, the compiler will optimise the variables inside the interrupt service routine so that they are no longer updated by loop.

Tested in Arduino 1.0

Duane B

rcarduino.blogspot.com

Comment by [Aidan.Te...@gmail.com](#), Jul 6, 2012

could you use a atmega328 or atmega168 to do this?

Comment by [pascal.a...@gmail.com](#), Jul 22, 2012

I got it to work on the Mega 2560 by using PWN pin 9 instead of 3. Just had to change the #define in the source code.

Comment by [DuaneSBa...@gmail.com](#), Aug 20, 2012

There is a short clip of my Auduino in this post about using the LM386 Amplifier chip to add reasonably quality audio to arduino projects using six simple components - <http://rcarduino.blogspot.com/2012/08/adding-audio-to-arduino-projects.html>

Duane B rcarduino.blogspot.com

Comment by [PMEEnri...@gmail.com](#), Nov 1, 2012

I think the google group is no longer accepting new posts, so I'll post this here.

What I wanted to add to the Auduino was the ability to change the key the instrument plays in. Having very little programming experience I wasn't sure I was going to be able to figure it out, but this seems to be working out well.

<http://pastebin.com/WVuaGA2e>

A new pot is connected to the analog 5 pin, and it cycles from C major to F major around the circle of fifths. I decided to use the circle of fifths instead of A, A#, B, C, etc because key changes will more reliably follow the circle of fifths.

All keys are oriented to a major scale, but obviously minor keys are available by selecting the relative major scale. I wanted to add augmented and diminished scales as well as a few more by adding another pot, or a switch, however adding any more key maps won't be possible due to Arduino memory limitations (as discovered by google group member goatboy). If someone knows a way around that issue please let me know.

Comment by [DuaneSBa...@gmail.com](#), Nov 15, 2012

As the group does not seem to be in use anymore, I have posted a copy of the Auduino code including the fix for Arduino 1.0 and an switchable delay/echo effect here -

<http://rcarduino.blogspot.com/2012/11/auduino-with-delay.html>

Duane B

Comment by [7ke...@gmail.com](#), Dec 6, 2012

Instead of using a pot could you use a line level signal to produce a tone e.g output from a keyboard?

Comment by [DuaneSBa...@gmail.com](#), Jan 4, 2013

Hi, Lots of options for input, I am currently building a version which uses a soft pot similar to a Korg Monotron. You could also build an interface similar to the stylophone/nebulophone or follow the approach I have taken with the five dollar keyboard project and hack a kids keyboard -

<http://rcarduino.blogspot.com/2012/10/five-dollar-synthesiser.html>

I am sure there are others that have implemented midi and line level interfaces.

Duane B

Comment by [Doctor.S...@gmail.com](#), Mar 20, 2013

I have done mine ! awesome project , thank you ! I used a marlboro box to do it , it's quite fun [https://vine.co/v/bp9w3Fe71wg?](https://vine.co/v/bp9w3Fe71wg?fb_action_ids=10200291204775534&fb_action_types=vine-app%3Apost&fb_source=aggregation&fb_aggregation_id=288381481237582)
[fb_action_ids=10200291204775534&fb_action_types=vine-app%3Apost&fb_source=aggregation&fb_aggregation_id=288381481237582](https://vine.co/v/bp9w3Fe71wg?fb_action_ids=10200291204775534&fb_action_types=vine-app%3Apost&fb_source=aggregation&fb_aggregation_id=288381481237582)

Comment by [brigh...@gmail.com](#), Mar 26, 2013

I'm looking at transferring this to a breadboard/ atmega chip to free up my arduino, is there anything is should know before doing this?

Comment by nielsh...@gmail.com, Jun 21, 2013

heeya Im trying to hook up rotary encoders instead of pot's .. does anyone have any idea how to do this ?. thx

Comment by etmoo...@gmail.com, Aug 21, 2013

Still a useful project. I'm working out midi control via a second microcontroller and DAC chips. Considering building four of these on daughter-boards and making a 4 voice polyphonic unit.

As for breadboarding one, no problem. Make sure to connect AGND to GND or you'll smoke your chip! I built one on an Evil Mad Scientist ATMegax8 target board, and added a LC filter to smooth out the output.

Glad to share results, will put up a website about the whole project.

Comment by etmoo...@gmail.com, Aug 22, 2013

If I might post this here: a first test of the outboard-controlled Auduino. I'm using a 12 bit DAC for the repetition freq. and 8 bit dacs for the other lines. With each "pitch" I'm doing a "split sweep" of the grains that is a fast sweep of grain 1 with a slower sweep of grain 2 in the middle of it. The "pitch" values (actual numbers being sent to the 12 bit DAC are on the display) are very arbitrary and obviously with a small amount of tweaking I can match the chromatic mapping. Again, will share whole project on webpage.

https://www.youtube.com/watch?v=gP0_2guP2MY

Comment by Taartmet...@hotmail.com, Sep 21, 2013

I've made my own little Auduino with five 10k pots. It works, but the volume is pretty dam low. I've got some headphones that I power with a Fiio E07K amp but I can't get past what the average person would deem 'reasonably loud'. I also have a big guitar amplifier but the sound that it produces from my Auduino is way, WAY less louder than when I plug my guitar in.

Is it the 10k potentiometers, something in the code, or should I leave amps behind and buy a crappy speaker? I used the original code with the fix by Duane B.

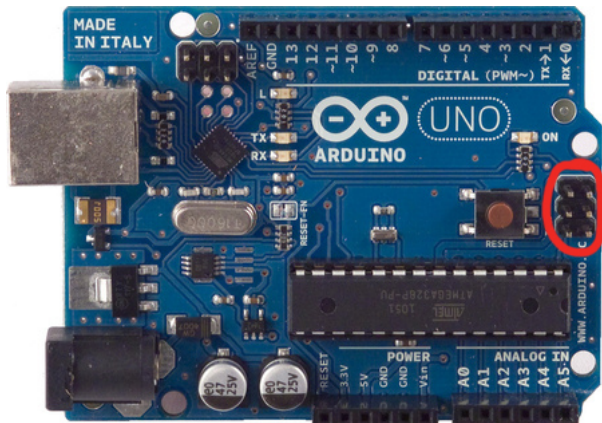
Comment by etmoo...@gmail.com, Sep 21, 2013

The 10k pots shouldn't affect volume, only the curve of the control. (I made the same mistake!) As for volume, check you connection to the output pin, I have had to add a series resistor to reduce the amplitude.

Another update on my adaptation <http://youtu.be/h3144hMPZHI>

Comment by Taartmet...@hotmail.com, Sep 22, 2013

I've found something funny. I put my finger on the 6 pins labeled ICSP by accident, and the bloody volume increased tenfold. Is this damaging? Can I exploit this? I've marked the pins in the picture below



Comment by Nigel.Wh...@gmail.com, Nov 28, 2013

YEAH Dude Try Sticking your fingers in the mains Im shure the volume will get much louder than that """"

Comment by mschausp...@gmail.com, Jan 11, 2014

Thanks for this project -- I built it when I first got my Arduino and loved it. I wrote about it along with some tricks I figured out -- volume control, on/off, and a super cheap case: <http://mschausprojects.blogspot.com/2013/07/an-arduino-based-synthesizer.html>

Comment by colellig...@gmail.com, Mar 6, 2014

I'm sorry if I do not speak English well and if you are not expert in the field. To realize the project arduino Should I Buy?

Comment by juliakim...@gmail.com, Oct 23, 2014

Do you have documentation in France language?

bugatti picture <http://www.carasports.com/category/bugatti/> picture

Comment by leeseib...@gmail.com, Jan 3, 2015

Check out my MIDI controlled Arduino build! <https://www.youtube.com/watch?v=ZcXmspfqsyw>

Comment by violinte...@gmail.com, Mar 3, 2015

Hi and thanks for the Auduino project. My name is Steve. I am an audio systems post-grad and an audio electronics/embedded/microcontroller nut/experimenter. I have studied the code for the Auduino and understand how it works but would love to know some more about the maths behind it. Would it be possible for you to explain the maths and the numbers used in the accumulators??

Oh, also I have got it working on the DUE. :) It is only using 1 channel but it would be simple to add another. I've posted the code below:

```
#include <DueTimer2.h>
```

```
uint16_t syncPhaseAcc; volatile uint16_t syncPhaseInc; uint16_t grainPhaseAcc; volatile uint16_t grainPhaseInc; uint16_t grainAmp; volatile uint8_t grainDecay; uint16_t grain2PhaseAcc; volatile uint16_t grain2PhaseInc; uint16_t grain2Amp; volatile uint8_t grain2Decay;
```

```
// Map Analogue channels #define SYNC_CONTROL (4) #define GRAIN_FREQ_CONTROL (0) #define GRAIN_DECAY_CONTROL (2) #define GRAIN2_FREQ_CONTROL (3) #define GRAIN2_DECAY_CONTROL (1)
```

```
// DB #define SMOOTH_PIN 8
```

```
#define MAX_DELAY 8192 unsigned char sDelayBufferMAX_DELAY; unsigned int nDelayCounter = 0; unsigned char bDelay;
```

```
#define DELAY_BUTTON 4
```

```
// Smooth logarithmic mapping // uint16_t antilogTable = {
```

```
64830,64132,63441,62757,62081,61413,60751,60097,59449,58809,58176,57549,56929,56316,55709,55109,
54515,53928,53347,52773,52204,51642,51085,50535,49991,49452,48920,48393,47871,47356,46846,46341,
45842,45348,44859,44376,43898,43425,42958,42495,42037,41584,41136,40693,40255,39821,39392,38968,
38548,38133,37722,37316,36914,36516,36123,35734,35349,34968,34591,34219,33850,33486,33125,32768
```

```
}; uint16_t mapPhaseInc(uint16_t input) {
```

```
return (antilogTable & 0x3f) >> (input >> 6);
```

```
}
```

```
// Stepped chromatic mapping // uint16_t midiTable = {
```

```
1. ,18,19,20,22,23,24,26,27,29,31,32,34,36,38,41,43,46,48,51,54,58,61,65,69,73,
```

```
77,82,86,92,97,103,109,115,122,129,137,145,154,163,173,183,194,206,218,231,
244,259,274,291,308,326,346,366,388,411,435,461,489,518,549,581,616,652,691,
732,776,822,871,923,978,1036,1097,1163,1232,1305,1383,1465,1552,1644,1742,
```

```
1. 45,1955,2071,2195,2325,2463,2610,2765,2930,3104,3288,3484,3691,3910,4143,
```

```
4389,4650,4927,5220,5530,5859,6207,6577,6968,7382,7821,8286,8779,9301,9854,
```

```
1. 440,11060,11718,12415,13153,13935,14764,15642,16572,17557,18601,19708,20879,
```

```
22121,23436,24830,26306
```

```
}; uint16_t mapMidi(uint16_t input) {
```

```
return (midiTable[(1023-input) >> 3]);
```

```
}
```

```
// Stepped Pentatonic mapping // uint16_t pentatonicTable54 = {
```

```
0,19,22,26,29,32,38,43,51,58,65,77,86,103,115,129,154,173,206,231,259,308,346,
411,461,518,616,691,822,923,1036,1232,1383,1644,1845,2071,2463,2765,3288,
3691,4143,4927,5530,6577,7382,8286,9854,11060,13153,14764,16572,19708,22121,26306
```

```
};
```

```
uint16_t mapPentatonic(uint16_t input) {
```

```
uint8_t value = (1023-input) / (1024/53); return (pentatonicTablevalue);
```

```
}
```

```
void setup() {
```

```
analogWrite(DAC0, 1); Timer3.attachInterrupt(audioLoop);
```

```
Timer3.start(32); // 31250 KHz sampling rate
```

```
pinMode(DELAY_BUTTON, INPUT);
```

```
// set pin mode and turn on pull up so that default mode // is PENTATONIC, pull the pin low to switch to smooth
pinMode(SMOOTH_PIN, INPUT); digitalWrite(SMOOTH_PIN, HIGH);
```

```
}
```

```
void loop() {
```

```

// The loop is pretty simple - it just updates the parameters for the oscillators. // Avoid using any functions that make extensive use of
interrupts, or turn interrupts off. // They will cause clicks and poops in the audio.

// defaults to pentatonic stepped tones, pull pin low for smooth frequency without distinct tones // syncPhaseInc =
mapPhaseInc(analogRead(SYNC_CONTROL)) / 4;

syncPhaseInc = mapPentatonic(analogRead(SYNC_CONTROL));

// updated 29/01/2013 // pull the DELAY_BUTTON pin high for delay, low for no delay // use either a pull up/pull down resistor // or a pull up
resistor with a toggle switch between the pin and ground bDelay = digitalRead(DELAY_BUTTON);

// Stepped mapping to MIDI notes: C, Db, D, Eb, E, F... //syncPhaseInc = mapMidi(analogRead(SYNC_CONTROL));

// Stepped pentatonic mapping: D, E, G, A, B

grainPhaseInc = mapPhaseInc(analogRead(GRAIN_FREQ_CONTROL)) / 2; grainDecay = analogRead(GRAIN_DECAY_CONTROL) / 8;
grain2PhaseInc = mapPhaseInc(analogRead(GRAIN2_FREQ_CONTROL)) / 2; grain2Decay = analogRead(GRAIN2_DECAY_CONTROL) / 4;
}

void audioLoop() {

  uint8_t value;

  uint16_t output;

  syncPhaseAcc += syncPhaseInc; if (syncPhaseAcc > syncPhaseInc) {

    // Time to start the next grain grainPhaseAcc = 0; grainAmp = 0x7fff; grain2PhaseAcc = 0; grain2Amp = 0x7fff;

  // LED_PORT ^= 1 << LED_BIT; // Faster than using digitalWrite

  }

  // Increment the phase of the grain oscillators grainPhaseAcc += grainPhaseInc; grain2PhaseAcc += grain2PhaseInc;

  // Convert phase into a triangle wave value = (grainPhaseAcc >> 7) & 0xff; if (grainPhaseAcc & 0x8000) value = ~value; // Multiply by current
grain amplitude to get sample output = value (grainAmp >> 8);

  // Repeat for second grain value = (grain2PhaseAcc >> 7) & 0xff; if (grain2PhaseAcc & 0x8000) value = ~value; output += value
(grain2Amp >> 8);

  // Make the grain amplitudes decay by a factor every sample (exponential decay) grainAmp -= (grainAmp >> 8) grainDecay; grain2Amp -=
(grain2Amp >> 8) grain2Decay;

  // Scale output to the available range, clipping if necessary output >= 9; if (output > 255) output = 255;

  // Duane B // rcarduino.blogspot.com // 15/11/2012 // add a button to set bDelay true or false to turn delay on and off if(bDelay) {

    // Output to PWM (this is faster than using analogWrite) // Here we add the delay buffer to the output value, this produces // an subtle
echo effect, the delay buffer is effectively replaying the sound from // 1/8th of a second ago.

  // LED_PORT |= 1 << LED_BIT; // Faster than using digitalWrite dacc_set_channel_selection(DACC_INTERFACE, 0); //select DAC channel 0

  dacc_write_conversion_data(DACC_INTERFACE, (output + (sDelayBufferDelayCounter2)>>1)); //write on DAC

  // PWM_VALUE = (output + (sDelayBufferDelayCounter2)>>1);

  // add the new output to the buffer so we can use it when the buffer next wraps around sDelayBufferDelayCounter2 = output;
nDelayCounter++; if(nDelayCounter == MAX_DELAY) {

    nDelayCounter = 0;

  }

} else { // LED_PORT &= ~(1 << LED_BIT); // Faster than using digitalWrite

  dacc_set_channel_selection(DACC_INTERFACE, 0); //select DAC channel 0

  dacc_write_conversion_data(DACC_INTERFACE, (output)); //write on DAC

  // PWM_VALUE = output;

}

}

//dacc_set_channel_selection(DACC_INTERFACE, 0); //select DAC channel 0 // dacc_write_conversion_data(DACC_INTERFACE, rd); //write on DAC
// dacc_set_channel_selection(DACC_INTERFACE, 1); //select DAC channel 1 // dacc_write_conversion_data(DACC_INTERFACE, rd2); //write on
DAC

```

Cheers! Steve.S

Comment by violinte...@gmail.com, Mar 3, 2015

oh, for the above comment my gmail is defacato@gmail.com. Thanks. Steve.S

► [Sign in](#) to add a comment

[Terms](#) - [Privacy](#) - [Project Hosting Help](#)

Powered by [Google Project Hosting](#)