

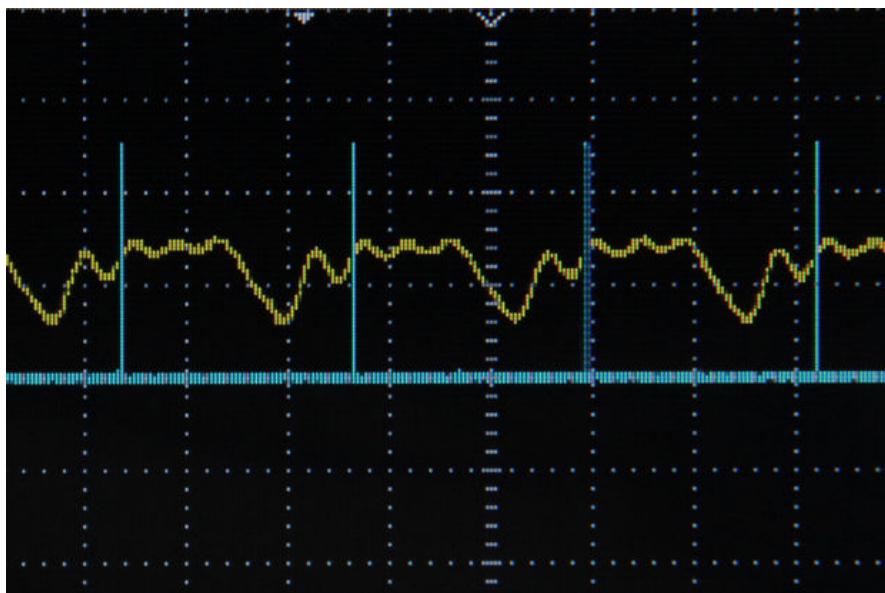


share what you make &gt;

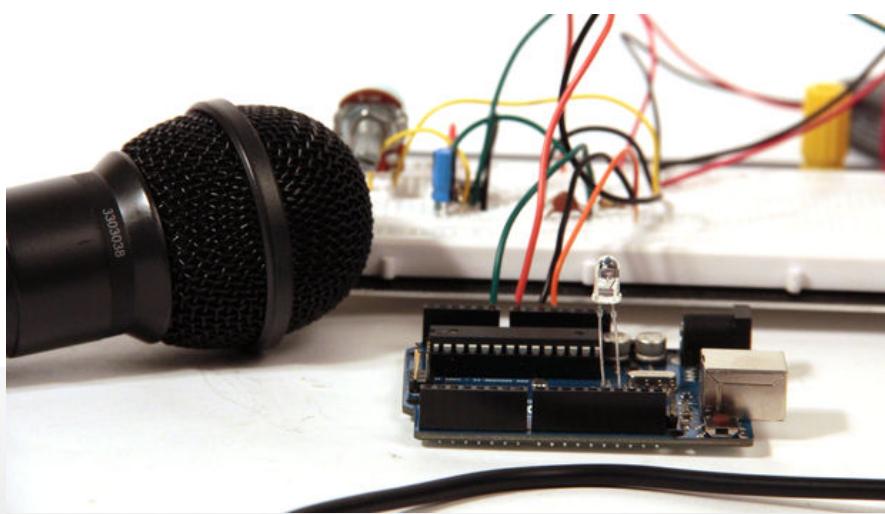
(Create account)

# GUERILLA DESIGN CONTEST

Prizes provided by Will Holman, author of Guerilla Furniture Design® & Storey Publishing

[VIEW CONTEST ▶](#)

(http://cdn.instructables.com/F2WX3D6/H7W4SMJY/F2WX3D6H7W4SMJY.LARGE.jpg)



(http://cdn.instructables.com/FK4/T20G/H742PPP8/FK4T20GH742PPP8.LARGE.jpg)

[VIEW CONTEST ▶](#)

## About This Instructable

145,427 views

License:  

249 favorites



amandaghassaei  
(/member/amandaghassaei)  
uh-man-duh-guss-eye-dot-com  
(http://www.amandaghassaei.co)

(/member/amandaghassaei)

Follow 2333

**Bio:** I'm a grad student at the Center for Bits and Atoms at MIT Media Lab. Before that I used to work at Instructables, writing code for the website and iOS apps and making stuff in our shop.

As a follow up to the Arduino Audio Input tutorial

(http://www.instructables.com/id/Arduino-Audio-Input/) that I posted last week, I wrote a sketch which analyzes a signal coming into the Arduino's analog input and determines the frequency. The code uses a sampling rate of 38.5kHz and is generalized for arbitrary waveshapes. I've also turned the LED attached to pin 13 into a clipping indicator, so you know if you need to adjust your signal's amplitude

as you send it into the Arduino.

Some project ideas for the code presented here include:

**pitch reactive projects**- change the color of RGB LEDs with pitch, or make a lock that only opens when you sing a certain pitch or melody

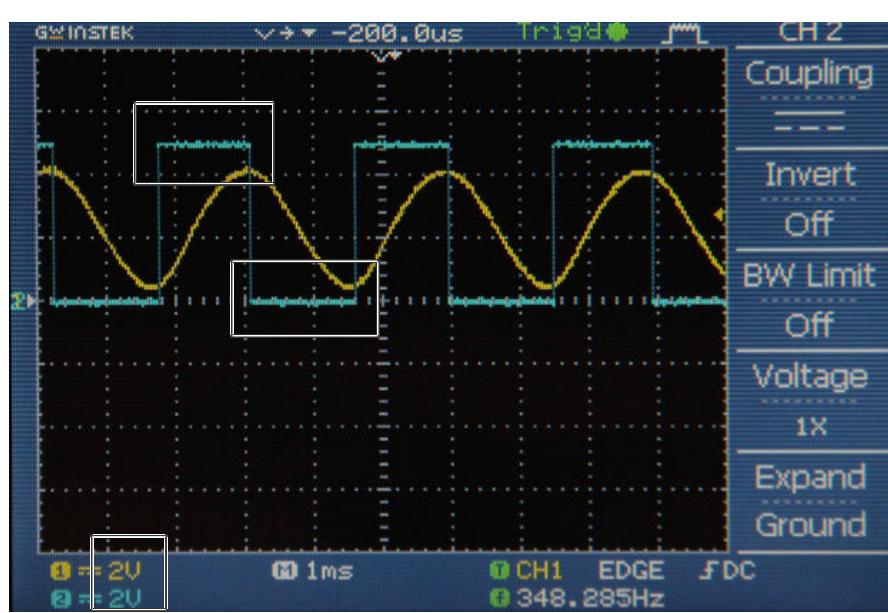
**audio to MIDI conversion**- get the Arduino to translate an incoming signal into a series of MIDI (<http://www.instructables.com/id/What-is-MIDI/>) messages. See my instructable about getting the Arduino to send and receive MIDI (<http://www.instructables.com/id/Send-and-Receive-MIDI-with-Arduino/>) for lots of example code to get started

**audio effects**- use the frequency information to reconstruct an audio signal from the tone() (<http://code.google.com/p/rogue-code/wiki/ToneLibraryDocumentation>) library or with some stored samples to make a cool effects box/synthesizer

The first step of this project is to set up the audio input circuit. I wrote a detailed Instructable about that here (<http://www.instructables.com/id/Arduino-Audio-Input/>).

Remove these ads by **Signing Up** (/account/gopro?  
source=removeads&nxtPgName=Arduino+Frequency+Detection&nxtPg=/id/Arduino-Frequency-Detection/?ALLSTEPS)

## Step 1: Detection of Signal Slope



(<http://cdn.instructables.com/F39/VIP6/H74320EP/F39VIP6H74320EP.LARGE.jpg>)

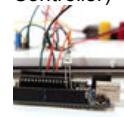
More by amandaghassaei



(/id/9-Degrees-of-Freedom-IMU)



(/id/Sugarcube-MIDI-Controller)



(/id/Beginner-Arduino)

Tags:

arduino (/tag/type-id/category-technology/keywords/arduino/)

frequency (/tag/type-id/category-technology/keywords/frequency/)

pitch (/tag/type-id/category-technology/keywords/pitch/)

note (/tag/type-id/category-technology/keywords/note/)

counter (/tag/type-id/category-technology/keywords/counter/)

detector (/tag/type-id/category-technology/keywords/detector/)

measurement (/tag/type-id/category-technology/keywords/measurement/)

meter (/tag/type-id/category-technology/keywords/meter/)

freq (/tag/type-id/category-technology/keywords/freq/)

audio (/tag/type-id/category-technology/keywords/audio/)

Related



Arduino play (micro)music without shield  
(<http://www.instructables.com/play-micromusic-without-shield/>)



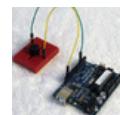
Arduino - Play The Piezo  
(<http://www.instructables.com/Play-The-Piezo/>)  
utm\_source=base&utm\_medium=related-instructables&utm\_campaign=related\_test



Arduino Audio Input  
(<http://www.instructables.com/Arduino-Audio-Input/>)  
utm\_source=base&utm\_medium=related-instructables&utm\_campaign=related\_test



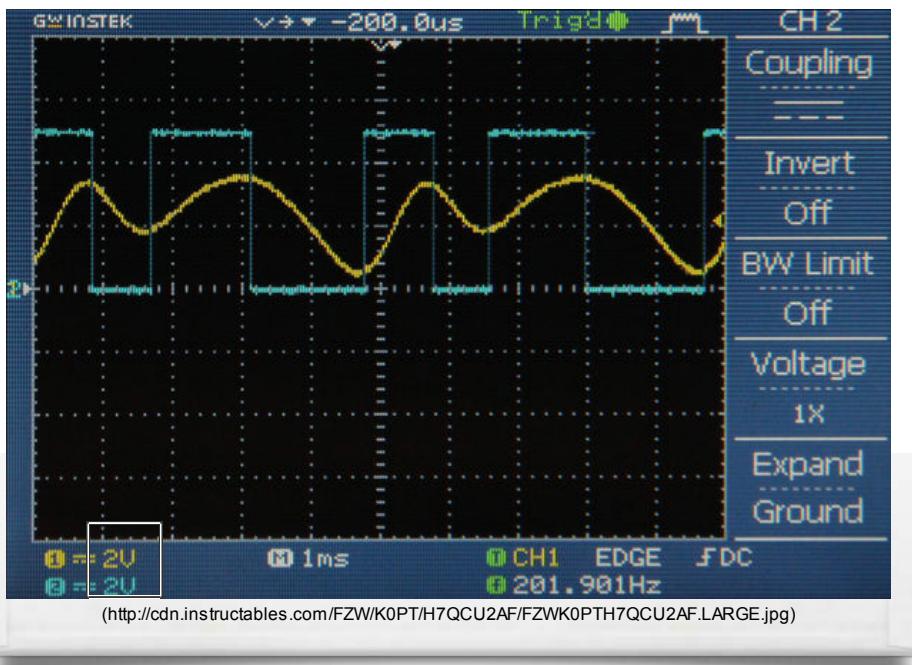
Arduino Vocal Effects Box  
(<http://www.instructables.com/Vocal-Effects-Box/>)  
utm\_source=base&utm\_medium=related-instructables&utm\_campaign=related\_test



Play the French Can Can Using an Arduino and Buzzer  
(<http://www.instructables.com/Play-the-French-Can-Can-Using-an-Arduino-and-Buzzer/>)

See More

([http://www.instructables.com/tag/type-id/?q=&utm\\_source=base&utm\\_medium=related-instructables&utm\\_campaign=related\\_test](http://www.instructables.com/tag/type-id/?q=&utm_source=base&utm_medium=related-instructables&utm_campaign=related_test))



First I wanted to experiment with peak detection, so I wrote a piece of code (below) that outputs a high signal when the incoming audio signal has a positive slope, and outputs a low signal when the incoming audio signal has a negative slope. For a simple sine wave, this will generate a pulse signal with the same frequency as the sine wave and a duty cycle ([http://en.wikipedia.org/wiki/Duty\\_cycle](http://en.wikipedia.org/wiki/Duty_cycle)) of 50% (a square wave). This way, the peaks are always located where the pulse wave toggles between its high and low states.

The important portion of the code is reproduced below. All of this code takes place in the ADC interrupt (interrupts and runs each time a new analog in value is ready from A0, more info about what interrupts are and why we use them can be found here (<http://www.instructables.com/id/Arduino-Timer-Interrupts/>))

```
prevData = newData;//store previous value
newData = ADCH;//get value from A0
if (newData > prevData){//if positive slope
    PORTB |= B00010000;//set pin 12 high
}
else if (newData < prevData){if negative slope
    PORTB &= B11101111;//set pin 12 low
}
```

I should note here that in this tutorial I use direct port manipulation (<http://www.arduino.cc/en/Reference/PortManipulation>) to turn off and on the output pin (pin 12) of the Arduino. I did this because port manipulation is a much faster way of addressing the Arduino's pins than the digitalWrite() command. Since I had to put all the code above inside an interrupt routine (<http://www.instructables.com/id/Arduino-Timer-Interrupts/>) that was going off at 38.5kHz, I needed the code to be as efficient as possible. You can read more about port manipulation on the Arduino website (<http://www.arduino.cc/en/Reference/PortManipulation>), or see the comments I've written above to understand what each line does. You'll also notice in the code below that I used some unfamiliar commands in the setup() function so that I could get the Arduino's analog input to sample at a high frequency. More info on that can be found in my Arduino Audio Input (<http://www.instructables.com/id/Arduino-Audio-Input/step6/Sampling-rate-of-40kHz/>) tutorial.

Fig 1 shows the pulse output in blue and the sine wave in yellow on an oscilloscope (<http://www.instructables.com/id/Oscilloscope-How-To/>). Notice how the pulse output toggles each time the sine wave reaches a maximum or minimum. Fig 2 shows the pulse output in blue for an arbitrary waveshape in yellow. Notice here how pulse wave takes on an irregular duty cycle because the incoming signal (yellow) is much more complicated than a sine wave.

```

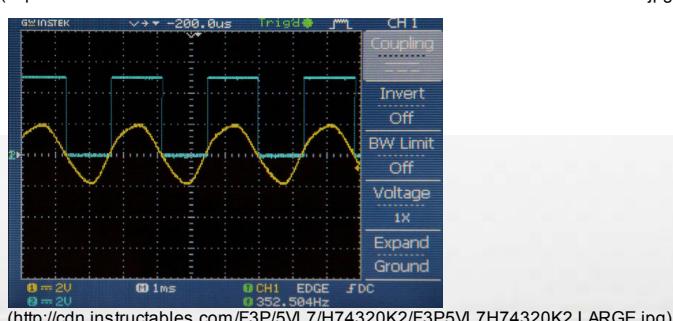
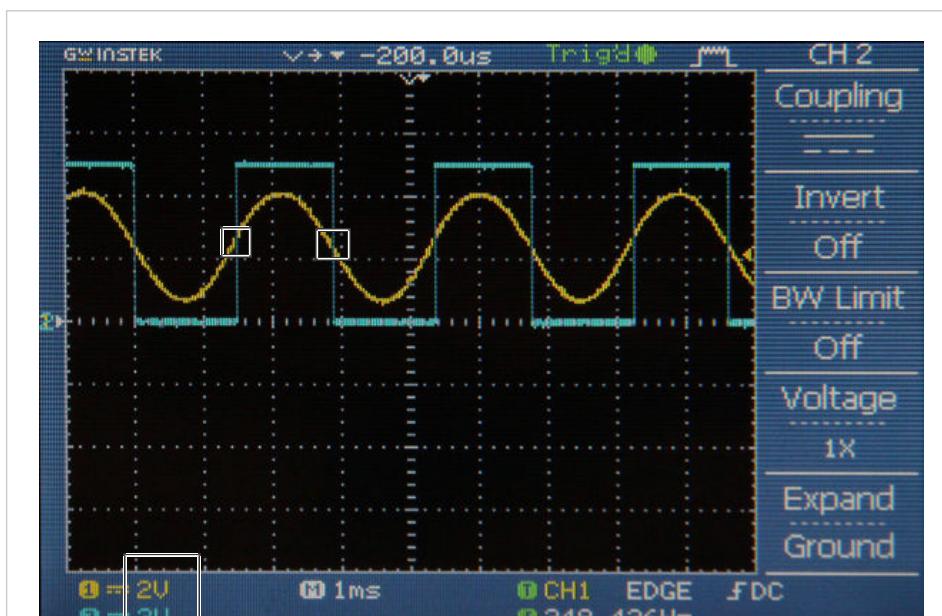
//Detection of signal slope with 38.5kHz sampling rate and interrupts
//by Amanda Ghassaei
//http://www.instructables.com/id/Arduino-Frequency-Detection/
//Sept 2012

/*
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 3 of the License, or
 * (at your option) any later version.
 *
 */

//clipping indicator variables
boolean clipping = 0;

```

## Step 2: Mid Point Detection



I decided to select the frequency of a wave by keeping instead of counting peaks. In the last step, I measured the wave where the slope = 0 instead of the wave where the slope = 0. However, when the slope = 0, noise skewed my results. When the slope was larger than 0, noise skewed my results even more.

The important changes to the code are reproduced below. Since I am measuring the incoming signal from A0 with 8 bit precision (0-255), the midpoint (2.5V) will give a value of 127. All of the following code takes place in the ADC interrupt (interrupts each time a new analog in value is ready from A0)

(http://cdn.instructables.com/FUMY38S/H74320K3/FUMY38SH74320K3.LARGE.jpg)

```

prevData = newData;//store previous value
newData = ADCH;//get value from A0
if (prevData < 127 && newData >= 127){//if increasing and crossing midpoint
    PORTB |= B00010000;//set pin 12 high
}
else if (prevData > 127 && newData <= 127){//if decreasing and crossing
midpoint
    PORTB &= B11101111;//set pin 12 low
}

```

Fig 1 shows the pulse output in blue and the incoming signal to A0 in yellow.

Notice how each time the signal crosses 2.5V, the pulse output toggles.

Specifically, the output goes high when the signal crosses 2.5V with a positive slope and the signal goes low when the signal crosses 2.5V with a negative slope.

Fig 2 shows the pulse output in blue and the audio signal before it has been +2.5V DC offset in yellow. Remember, this DC offset was necessary to get the audio signal in the 0-5V range for the Arduino's analog input pin, but normally audio signal oscillate around 0V. In fig 2 you can see how the pulse outputs toggle corresponds to the time when the audio signal crosses 0V. Fig 3 shows an arbitrary waveform in yellow (again before DC offset) and the pulse output in blue. Again, the pulse toggles each time the yellow signal crosses 0V, notice how the behavior of the pulse output with the arbitrary waveform is more complex than with the sine wave.

```

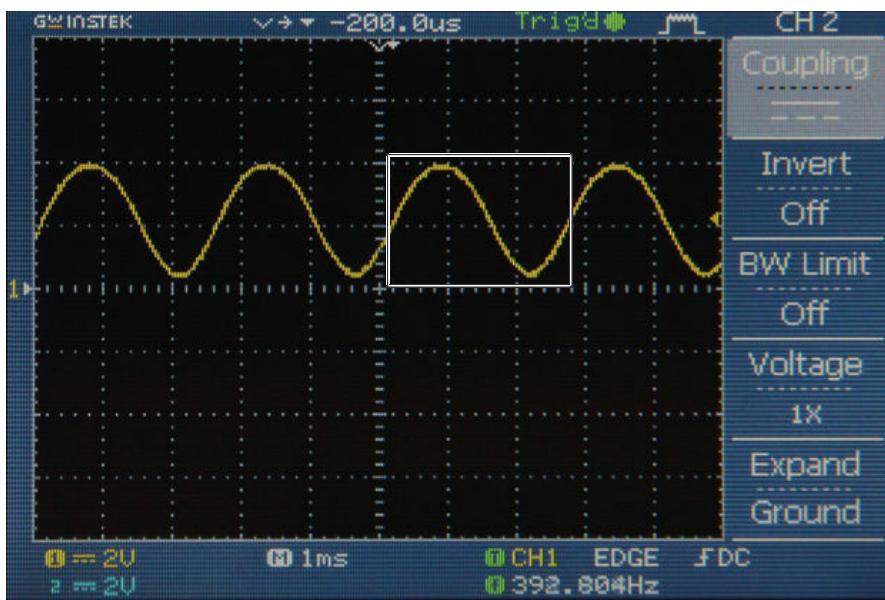
//Detection of midpoint crossing with 38.5kHz sampling rate and interrupts
//by Amanda Ghassaei
//http://www.instructables.com/id/Arduino-Frequency-Detection/
//Sept 2012

/*
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 3 of the License, or
 * (at your option) any later version.
 *
 */

//clipping indicator variables
boolean clipping = 0;

```

### Step 3: Sine Wave Frequency Detection



(http://cdn.instructables.com/F6H/HKY9/H7QCU2C8/F6HHKY9H7QCU2C8.LARGE.jpg)



(http://cdn.instructables.com/FFA/BUMG/H7QCU277/FFABUMGH7QCU277.LARGE.jpg)

Next I measured the period of an incoming sine wave, calculated the frequency, and printed the frequency. To do this I set up a timer in the ADC interrupt that increments each time the interrupt executes (a rate of 38462Hz). Each time the incoming signal crosses 2.5V with a rising slope I sent the current value of the timer to a variable called "period" and reset the timer to 0. That code is reproduced below (all takes place within the ADC interrupt).

```

prevData = newData;//store previous value
newData = ADCH;//get value from A0
if (prevData < 127 && newData >= 127){//if increasing and crossing midpoint
    period = timer;//get period from current timer value
    timer = 0;//reset timer
}

timer++;//increment timer

```

Then in the main loop() function, I calculated the frequency by dividing the timer rate by the period. I used Serial.print to print these results in the Arduino serial monitor.

```

frequency = 38462/period;//timer rate/period
//print results
Serial.print(frequency);
Serial.println(" hz");

```

Fig 1 shows the signal coming into A0. The start and end of one cycle measured by timer is indicated by the image note. Fig 2 shows the output from the serial monitor (command/ctrl+shift+m). This technique works great for sine waves, but when wave become more complicated (and cross 2.5V more than twice in one cycle) this technique breaks down.

```
//sine wave freq detection with 38.5kHz sampling rate and interrupts
//by Amanda Ghassaei
//http://www.instructables.com/id/Arduino-Frequency-Detection/
//July 2012
```

```
/*
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 3 of the License, or
 * (at your option) any later version.
 *
 */
```

## Arduino Frequency Detection

by amandaghassaei (/member/amandaghassaei/)



boolean clipping = 0;

[Download \(/id/Arduino-Frequency-Detection/?download=pdf\)](#)

[\(/id/Arduino-Frequency-Detection/\)](#)

4 Steps

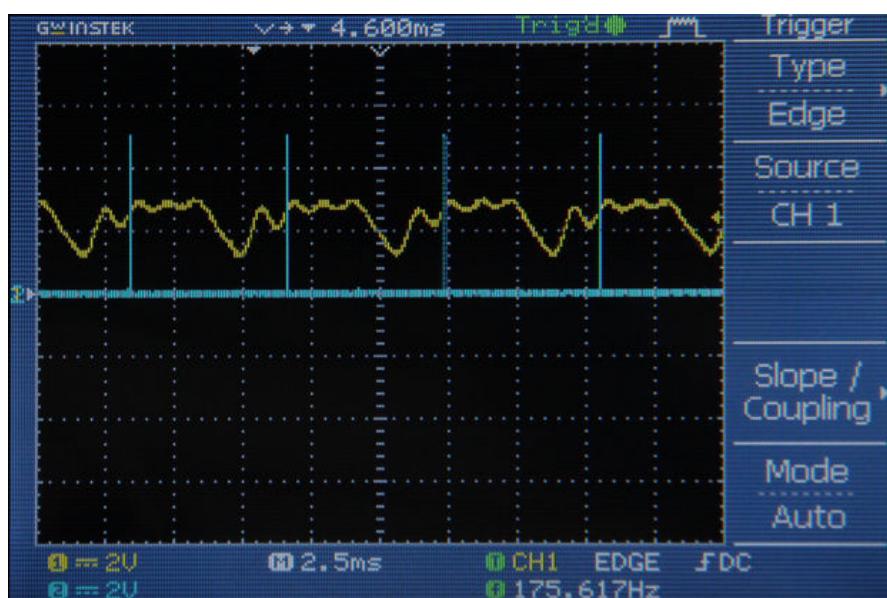
Collection

I Made it!

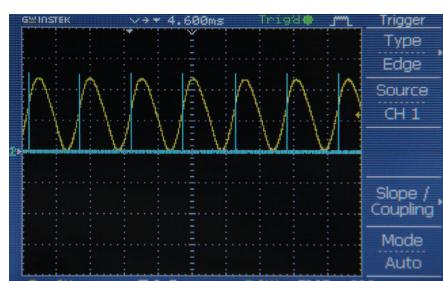
Favorite

Share ▾

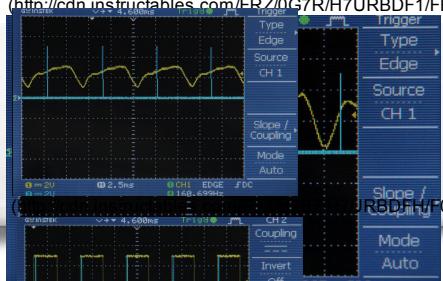
### Step 4: Generalized pitch detection



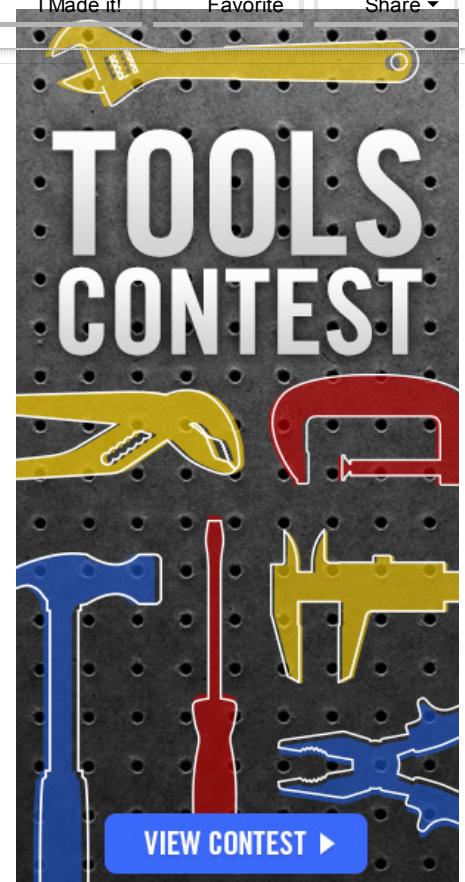
(<http://cdn.instructables.com/FBQ/LQS0/H7UQYP8D/FBQLQS0H7UQYP8D.LARGE.jpg>)



(<http://cdn.instructables.com/FRZ0G7R/H7URBDF1/FRZ0G7RH7URBDF1.LARGE.jpg>)



(<http://cdn.instructables.com/F0XH787H7URBDFH.F0XH787H7URBDFH.LARGE.jpg>)



In this code I wanted to measure the frequency of waves of different amplitudes. When writing this code I wanted to stick with the point I wanted to measure the frequency of the wave using the peaks and valleys as markers (to minimize error due to noise). I also wanted to write something that was as simple as possible (needs to execute at 38.5kHz) while

(<http://cdn.instructables.com/F7D/1KYUH7W5YUB3/F7D/1KYUH7W5YUB3.LARGE.jpg>)

still being I  
technique  
(<http://www>



Basically within the bounds of with an upward slope, let's call these "threshold events". If this happened multiple times in one cycle I chose the threshold event with the largest slope to be the beginning of my cycle. Similar to the last step, I used a variable called "time" (incremented at a rate of 38.5kHz) to measure the time between threshold events and stored this in an array called timer[]]. I also recorded the slope at each of the threshold events in an array called slope[]]. Then I compared the elements of timer[] and slope[] to figure out where there was a match. Once a match was found, I added up the elements of timer[] to determine the duration of the cycle and sent this value to a global variable called "period." Then in the main loop() function (all of the steps I've just described happen in the ADC interrupt routine) I used the value of period to calculate the frequency and print it. I should also add that I put a variable in the code called "noMatch" which helped me to decide that it had been too long since I had a good match and that I should just rerecord the elements of timer[] and slope[]].

When writing this I thought about a lot of possible scenarios that might break the algorithm. The trickiest wave in my mind is one which passes the 2.5V threshold many times in one cycle at similar slopes and spaced out along the cycle similarly. If you have a wave like this, you should keep slopeTol very low (0-3) and you might find that lowering timerTol (to 5 maybe) helps track the wave correctly. Also, if you want to measure waves with very steep slopes (like pulse waves) you should set the value of slopeTol up to 100 or even all the way up to 256 to track them better.

Generally this piece of code seems to handle lots of shapes very well, you can see some of my results in the images above. The incoming signal is shown in yellow and the threshold event that the Arduino is tracking is indicated by a pulse of pin 12 (blue).

```
//generalized wave freq detection with 38.5kHz sampling rate and interrupts
//by Amanda Ghassaei
//http://www.instructables.com/id/Arduino-Frequency-Detection/
//Sept 2012

/*
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 3 of the License, or
 * (at your option) any later version.
 *
 */

//clipping indicator variables
boolean clipping = 0;
```

I also added a bit of code to stop calculating and print frequency data when the amplitude of the wave falls below a certain level. (If there is little or no signal then the code above sometimes spits out a bunch of garbage). Here it is:

```
//generalized wave freq detection with 38.5kHz sampling rate and interrupts
//by Amanda Ghassaei
//http://www.instructables.com/id/Arduino-Frequency-Detection/
//Sept 2012

/*
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 3 of the License, or
 * (at your option) any later version.
 *
 */

//clipping indicator variables
```



We have a **be nice** comment policy.  
Please be positive and constructive.

I Made it! Add Images Make Comment

1-40 of  
140

Next » (<http://www.instructables.com/id/Arduino-Frequency-Detection/?&sort=ACTIVE&limit=40&offset=40#DISCUSS>)



성범강 (/member/%EC%84%B1%EB%B2%94%EA%B0%95/)

5 days ago

Reply

Hi Amanda, thank you very much for this source!! very amazing!!!  
(/member/%EC%84%B1%EB%B2%94%EA%B0%95/)  
but... little problem my arduino...

your source is excellent !!

but.. my arduino is not working

my arduino version...is...

8-bit

Microcontroller

with 256K Bytes

In-System

Programmable

Flash

maybe...

version is problem is different???



byondo (/member/byondo/)

1 month ago

Reply

Hi Amanda, thank you very much for this instructable! Very useful!  
(/member/byondo/)  
I prefer this simpler code version (integrated with amp threshold), because I want to make a sound analyzer , where you whistle or sing a "oooh" note and then you have the frequency. Making a portable built with a 4 digit display and there it is a perfect gadget for any sound engineer or musician, who wants to know which frequency causes microphones feedbacks.

Does it make any sense collecting a number of measurements and make a mathematical average? as I don't need a fast process, but a slower and more solid (without "garbage" measurements); I tried some "for" cycle with a simple array, but it doesn't seem to add any precision (maybe I don't know where to put it)..



rien.brand (/member/rien.brand/)

1 month ago

[Reply](#)

Hi Amanda,

(/member/rien.brand/)  
Thanks for the very usefull work you have done, nou I can measure the frquency from my agriculture application that use a exciter ring.

I have one qustion how can I change to another analog pin in the ide?

Thanks,

Rien Brand

Netherlands



renaissa (/member/renaissa/)

2 months ago

[Reply](#)

(/member/renaissa/)  
Hi amanda...your instructables on arduino are great..however i want to know if it is possible to measure a amplitude of sine wave on arduino

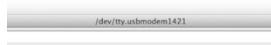


shinew (/member/shinew/)

3 months ago

[Reply](#)

(/member/shinew/)  
Thanks for the instruction! I just put together a circuit with an amplified electret mic signal using opamp feeding into it. However, when I'm looking at the serial output, it only seems to work for the first second or 2, then it stops working with the last line looks like attached image. Any idea what could be the cause? thanks!



(<http://cdn.instructables.com/FWI/AWOJ/I2QUADWB/FWIAWOJI2QUADWB.LARGE.jpg>)



weichi.chien (/member/weichi.chien/)

3 months ago

[Reply](#)

(/member/weichi.chien/)  
Hi, this is an excellent project. It works fine on my arduino uno. However, I need it to work on a arduino mega with ethernet shield. When adding the codes to my ethernet project, the connection to internet is brocken (I'm uploading data onto a xively server). Any suggestion?



buddika123 (/member/buddika123/)

4 months ago

[Reply](#)

Hiii

(/member/buddika123/)  
However, I measured the frequency ....I want to detect frequnacy range 60Hz to 100Hz....can I use this progtame thanks and quick reply from you



Derpancakes (/member/Derpancakes/)

4 months ago

[Reply](#)

(/member/Derpancakes/)  
This is awesome! I'm looking in to a guitar-MIDI pedal, but a DIY solution is cheaper and way cooler. Props to you for this awesome bit of code!



LucasP2 (/member/LucasP2/)

5 months ago

[Reply](#)

(/member/LucasP2/)  
So with this arduino code I would be able to control a the light's frequency with the frequency of the sound? I want to build a lighting system for my room and car that will react with the sound, but in a way so that bass="cool colors", treble= "warm colors"(or even vice versa). I feel like that would give a better experience to the music

[novaninjas \(/member/novaninjas/\)](#)



5 months ago Reply  
Hi, has anybody completed this project (possibly, with a diff sort of mic) and  
(/member/amanda/)  
would mind sharing about it? Am a beginner and would really like to try this so..  
slightly more detailed steps would be most helpful :D



pierattilio (/member/pierattilio/) 5 months ago Reply  
Hi amanda, i want to congratulate with you on this instructable, really powerful :D I  
(/member/pierattilio/)  
have a question, that is is it possible to check the frequencies of a playing song in  
this way, through an aux signal from i.e. an MP3 player or the notebook audio  
headphones output (after the right managment of the signal like in your  
instructable on arduino audio input)?

Thanks a lot  
;D



aoss (/member/aoss/) 6 months ago Reply  
Has anyone made this project with ChipKit Uno32. I am really looking for projects  
(/member/aoss/)  
like that for my Uno32 developement board because I need to read analog signal  
frequency coming from optical rotation sensor.



Bokononestly (/member/Bokononestly/) 1 year ago Reply  
Great work! I'm trying to use this code for an automatic laser oscilloscope I am  
(/member/Bokononestly/)

I've found that the code stops measuring frequencies as soon as I try  
analogRead() from a different analog pin. At the moment I don't fully understand  
the lower level programming of the ADC that you've done. Is there any way to  
read from the other analog pins without interfering with the pitch measurement?



amandaghassaei (/member/amandaghassaei/) (author) Bokononestly  
1 year ago Reply  
yes, the way this is set up, all of the other analog pins  
(/member/amandaghassaei/)  
are deactivated. you can read an analog value from a  
digital pin by using RCtime:  
<http://arduino.cc/en/Tutorial/RCtime>  
<http://www.instructables.com/id/Arduino-Basics-RCtime/>  
hope that works for what you're doing.



MirzaR1 (/member/MirzaR1/) amandaghassaei 7 months ago Reply

Thank you for the great article. I was just wondering can I use a  
(/member/MirzaR1/)  
Microphone with a preamp to make the detection.

I have :

<http://www.freetronics.com/products/microphone-sou...>  
(<http://www.freetronics.com/products/microphone-sound-input-module>)

Thank you very much!



arimika (/member/arimika/) 10 months ago Reply  
I tried to build guitar tuner <http://www.instructables.com/id/Arduino-Guitar-Tun...>  
(<http://www.instructables.com/id/Arduino-Guitar-Tuner/?ALLSTEPS>)

but arduino can't read the input from my guitar.  
can you help me to solve this problem ?



amandaghassaei (/member/amandaghassaei/) (author) arimika  
10 months ago Reply  
(/member/amandaghassaei/)  
do you have an oscilloscope?



arimika (/member/arimika/) amandaghassaei

9 months ago

[Reply](#)

i don't have oscilloscope.

(/member/arimika/) does become a problem if I use 100nF Cap on the DC offset ?

i see your project use 47nF Cap.



amandaghassaei (/member/amandaghassaei/) (author) arimika

9 months ago

[Reply](#)

no that should be fine. You will need an

(/member/amandaghassaei/) oscilloscope to debug this, maybe see if there's a

hackerspace

([http://hackerspaces.org/wiki>List\\_of\\_Hacker\\_Space](http://hackerspaces.org/wiki>List_of_Hacker_Space)

or school near you where you can use one for free.



arimika (/member/arimika/) amandaghassaei

9 months ago

[Reply](#)

what is the main problem that the input from guitar can not reach the

(/member/amandaghassaei/) Arduino?

whether the project could be completed if there is oscilloscope ?



amandaghassaei (/member/amandaghassaei/) (author) arimika

9 months ago

[Reply](#)

you could try using a computer running audacity as

(/member/amandaghassaei/) an oscilloscope, to find out where the

signal is getting lost. Do you have an audio input on

your computer?



arimika (/member/arimika/) amandaghassaei

9 months ago

[Reply](#)

yes, i have.

(/member/amandaghassaei/) how to make audio output from amplifier and DC offset circuit into my computer ? there are 3 outputs "A0 , 5v and GND"



andrew95434 (/member/andrew95434/)

10 months ago

[Reply](#)

I'm new in Arduino, I want to make this but with an Electret Microphone, I think it's

(/member/andrew95434/) very different than the microphone you used. How can I make the circuit with an

Electret Microphone?



amandaghassaei (/member/amandaghassaei/) (author) andrew95434

10 months ago

[Reply](#)

it shouldn't be too different, but you might need to

(/member/amandaghassaei/) change the gain on the amplifier going into the

Arduino's analog pin. You will probably want an

oscilloscope handy so you can see what you're doing.



andrew95434 (/member/andrew95434/) amandaghassaei

9 months ago

[Reply](#)

Do you know if there's a virtual oscilloscope which I

(/member/andrew95434/) could use? Like a PC program.

About the microphone, I'm using an Elecret Microphone with Breakout Board (amplifier included), I connect it directly to the Arduino, is everything alright?

amandaghassaei (/member/amandaghassaei/) (author) andrew95434



9 months ago [Reply](#)  
i guess you could use the audio input on your comp and record using  
(/member/amandaghassaei)

MY  
FACE  
HERE

**sweller** (/member/sweller/)

10 months ago

[Reply](#)

I'm so happy I stumbled onto your Instructable! This is \*exactly\* what I'm  
(/member/sweller) attempting to do. Unfortunately, I'm trying to do it with a Picaxe. :-)



**EthnoChris** (/member/EthnoChris/)

10 months ago

[Reply](#)

Hey Amanda! Awesome Instructable. This has been a great learning experience.  
(/member/EthnoChris) I managed to get the code ( and circuit from your Arduino Audio Input Instructable  
) working fine on an Uno clone, but when uploading the same sketch to an  
Arduino Micro it doesn't seem to update the output values when I print the  
variable ( I just get "inf" ).

Is there a difference in the Micro that would prevent this sketch from working? I'm  
using the same sketch and circuit for both versions ( I've only swapped out the  
Uno to a Micro ), but have I potentially done something incorrectly?



**amandaghassaei** (/member/amandaghassaei/) (author)

EthnoChris

10 months ago

[Reply](#)

yeah the micro uses an atmega32 (as opposed to  
(/member/amandaghassaei) a 328). I think the code will have to be adjusted  
for the chip. You can take a shot using the info  
from the datasheet  
([http://www.atmel.com/dyn/resources/prod\\_documents/](http://www.atmel.com/dyn/resources/prod_documents/))  
I've never used a micro unfortunately.



**EthnoChris** (/member/EthnoChris/)

amandaghassaei

10 months ago

[Reply](#)

Thanks for the quick response. Converting it over  
(/member/EthnoChris) might be a bit over my head, so I'll stick with the  
Uno for now. Cheers for the information!



**amandaghassaei** (/member/amandaghassaei/) (author)

amandaghassaei

10 months ago

[Reply](#)

bad link, here it is:  
(/member/amandaghassaei)  
<http://www.atmel.com/devices/atmega32u4.aspx>



**raptorofaxys** (/member/raptorofaxys/)

1 year ago

[Reply](#)

Hey there! This was a great read! I'm a fan of yours - your 'ibles are always  
(/member/raptorofaxys) interesting and very well put together.

I just wanted to mention that a few years ago I built an Arduino-based guitar tuner  
and had also tried this kind of frequency detection approach, but, as you mention  
yourself, I ultimately found it to be unruly when fed arbitrary waveforms. After some  
research, I implemented a modified YIN  
([http://audition.ens.fr/adc/pdf/2002\\_JASA\\_YIN.pdf](http://audition.ens.fr/adc/pdf/2002_JASA_YIN.pdf)  
([http://audition.ens.fr/adc/pdf/2002\\_JASA\\_YIN.pdf](http://audition.ens.fr/adc/pdf/2002_JASA_YIN.pdf))) on Arduino, and this worked  
extremely well while keeping the code simple.

The project writeup is here, and the code is freely downloadable from that page:  
<http://deambulatorymatrix.blogspot.ca/2010/11/digital-chromatic-guitar-tuner-2008.html> (<http://deambulatorymatrix.blogspot.ca/2010/11/digital-chromatic-guitar-tuner-2008.html>)

All the best, and keep up the great work! :D

**gtoal** (/member/gtoal/)

raptorofaxys



1 year ago Reply  
I think I may use your code - thanks for posting it. I want to build a system to adjust the tension on bicycle spokes by pinging the spokes and determining the resonant frequency, then adjusting all the spokes to match the mean - an extremely similar process to guitar tuning (if a guitar had 6 identical strings :-), which is actually what gave me the idea. The frequency of the spokes on my bike is centered around 350Hz, so amandaghassaei's code probably won't do, as she says that it starts to become inaccurate at 350Hz. Of course I could use something faster than an arduino, maybe one of the arm-based arduino-alikes, but the thought of building this with an atmega is very tempting.

Thanks,

Graham

PS I expect most people's bike spokes will resonate at a lower frequency, but I have shorter spokes as it's an e-bike with a hub motor.



rinksrides ([/member/rinksrides/](#)) gtoal 11 months ago Reply

Quick FYI, the whole point of adjusting spokes tension is to true the wheel (~~make it~~ straight as possible). In my experience (mostly Free ride style), tension on the spokes varies according to any warp in the rim itself.. and in the case of my bikes, after doing a few 6ft drops the rims get tweaked and some of the spokes need to really be cranked up to get the rim trued again. hope this helps in any troubleshooting you might be having problems with.



raptorofaxys ([/member/raptorofaxys/](#)) gtoal 1 year ago Reply

Wow, that sounds like quite an original use for this kind of setup! I know my dad tunes his own spokes using a piano, but using electronics would probably be more accurate :) Best of luck, and keep us posted!



RU4Realz ([/member/RU4Realz/](#)) gtoal 1 year ago Reply

sounds like a cool project!  
([/member/RU4Realz/](#))



amandaghassaei ([/member/amandaghassaei/](#)) (author) raptorofaxys 1 year ago Reply

thanks! I like your project a lot! looks like the YIN algorithm works great. you should enter the microcontroller contest:  
<http://www.instructables.com/contest/>



raptorofaxys ([/member/raptorofaxys/](#)) amandaghassaei 1 year ago Reply

Hahaha, thanks for the suggestion! I knew about the contests, but did not realize you had so many going in parallel. I would have loved to enter, but I live in good ol' Quebec, which makes me ineligible. :-(



RobinCh ([/member/RobinCh/](#)) 11 months ago Reply

Hi miss Amanda!

([/member/RobinCh/](#))

It's really great what you have made! I tried this on my Arduino Uno and it works perfect! But now, I have a problem. I want to use this code on my other Arduino, the Arduino Due. For this, I had to download the new beta-software 1.5 (Instead of 1.0 for the other Arduino's) because the standard software doesn't support the Due (he's too recent I guess). After trying a lot of things, I could finally connect him to the software! But now I uploaded your code again with the new software to the new Arduino, and I have a lot of errors, but I changed nothing... This is the list the software gave me:

\_1e\_versie\_Arduino\_sketch.ino: In function 'void setup()':

\_1e\_versie\_Arduino\_sketch:34: error: 'cli' cannot be used as a function

```
_1e_versie_Arduino_sketch:39: error: 'ADCSRA' was not declared in this scope  
_1e_versie_Arduino_sketch:40: error: 'ADCSRB' was not declared in this scope  
_1e_versie_Arduino_sketch:42: error: 'ADMUX' was not declared in this scope  
_1e_versie_Arduino_sketch:42: error: 'REFS0' was not declared in this scope  
_1e_versie_Arduino_sketch:43: error: 'ADLAR' was not declared in this scope  
_1e_versie_Arduino_sketch:45: error: 'ADPS2' was not declared in this scope  
_1e_versie_Arduino_sketch:45: error: 'ADPS0' was not declared in this scope  
_1e_versie_Arduino_sketch:46: error: 'ADATE' was not declared in this scope  
_1e_versie_Arduino_sketch:47: error: 'ADIE' was not declared in this scope  
_1e_versie_Arduino_sketch:48: error: 'ADEN' was not declared in this scope  
_1e_versie_Arduino_sketch:49: error: 'ADSC' was not declared in this scope  
_1e_versie_Arduino_sketch:51: error: 'sei' was not declared in this scope  
_1e_versie_Arduino_sketch.ino: In function 'void ADC_vect()':  
_1e_versie_Arduino_sketch:57: error: 'ADCH' was not declared in this scope  
_1e_versie_Arduino_sketch:65: error: 'PORTB' was not declared in this scope  
_1e_versie_Arduino_sketch.ino: In function 'void loop()':  
_1e_versie_Arduino_sketch:74: error: 'PORTB' was not declared in this scope
```

What do I have to do about it? I don't understand why it doesn't work in this software...

Thank you so much for the help! I really appreciate it!

Greetings,  
Robin C,  
Belgium



amandaghassaei (/member/amandaghassaei/) (author) RobinCh

11 months ago

[Reply](#)

this code will have to be rewritten for the due,  
(/member/amandaghassaei/ have deleted one, sorry!)

1-40 of  
140

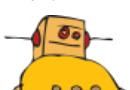
[Next »](#) (<http://www.instructables.com/id/Arduino-Frequency-Detection/?&sort=ACTIVE&limit=40&offset=40#DISCUSS>)

We have a **be nice** comment policy.  
Please be positive and constructive.

[I Made it!](#) [Add Images](#) [Make Comment](#)

## About Us

[Who We Are](#) (/about/)  
[Advertise](#) (/advertise/)  
[Contact](#) (/about/contact.jsp)  
[Jobs](#) (/community/Positions-available-at-Instructables/)  
[Help](#) (/id/how-to-write-a-great-instructable/)



## Find Us

[Facebook](#) (<http://www.facebook.com/instructables>)  
[Youtube](#) (<http://www.youtube.com/user/instructablestv>)  
[Twitter](#) (<http://www.twitter.com/instructables>)  
[Pinterest](#) (<http://www.pinterest.com/instructables>)  
[Google+](#) (<https://plus.google.com/+instructables>)  
[Tumblr](#) (<http://instructables.tumblr.com>)



For Teachers (/teachers/) [Join our newsletter:](#)  Join!

Artist-in-Residence (<http://www.autodesk.com/artist-in-residence/home>)

Gift Pro Account (/account/give?sourcea=footer)

Forums (/community/)

Terms of Service (<http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=21959721>)

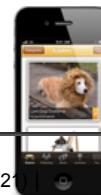
Answers (/tag/type/question/2601-RECENT) | Privacy Statement (<http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=21292079>) |

Site map (/site-map/)

Legal Notices & Trademarks (<http://usa.autodesk.com/legal-notices-trademarks/>) | Mobile Site (<http://m.instructables.com>)



© 2014 Autodesk, Inc.



## Mobile

Download our new apps for iOS,

Android and Windows 8!

English

Android

(<https://play.google.com/store/apps/details?id=com.adsk.instructables>)

iOS

(<https://itunes.apple.com/app/instructables/id586765571>)

Windows

(<http://apps.microsoft.com/windows/en-us/app/7afc8194-c771-441a-9590-54250d6a8300>)



Go Pro Today » (/account/gopro?sourcea=footer)



We're Hiring! » (/community/Positions-available-at-Instructables/)