Auduino - Lo-Fi Arduino Synthesizer

by **DIY Circuits** on June 27, 2016

Table of Contents

Intro: Auduino - Lo-Fi Arduino Synthesizer

In this instructable I will show you how I made the **Auduino synthesizer**. It is meant to be listened on headphones but if you want to listen on speaker you will have to use an amplifier which I will show in other instructable.

The circuit is relatively simple. For me the hardest part was to burn the bootloader on a blank atmega328 and uploading the program.

Step 1: Demonstration

I **did not** make the video.

In this video you can hear some of the sounds the synthesizer can produce. The video itself is not only a demonstration of what you can do with this circuit but also a great tutorial on how to make it. In part2 he also shows how to make cool front panel for not only this project!

The differences are that I have etched a custom PCB for this circuit and aslo modified it so it could run on batteries.

Step 2: Parts List

Here are the components you will need:

1x - Auduino PCB

1x - Enclosure box

1x - Battery holder for 3AA - If you buy one with 2f22 (9v) connector you wil also need a 9v connector clip

5x - Potentiometer 5k? linear

5x - Potentiometer knob

1x - DC power jack

1x - 2pole 2throw ON-OFF-ON switch

1x - Audio Jack 6.3mm/3.5mm mono/stereo - I used 6.3mm stereo because it holds the cable tighter but if you want to listen with headphones you will also need a reduction to 3.5mm

1x - 7805 voltage regulator

3x - 100µF electrolytic capacitor (at least 16V)

1x - 33nF foil capacitor (can be also ceramic but foil is advised for audio circuits)

2x - 22pF ceramic capacitor

1x - 2.2 k? resistor

1x - 1 k? resistor (or 2.2k? if the led is very bright)

1x - 10k

1x - micro push button (for resetting the arduino if something goes wrong during testing)

1x - 28 pin socket

1x - Atmega328

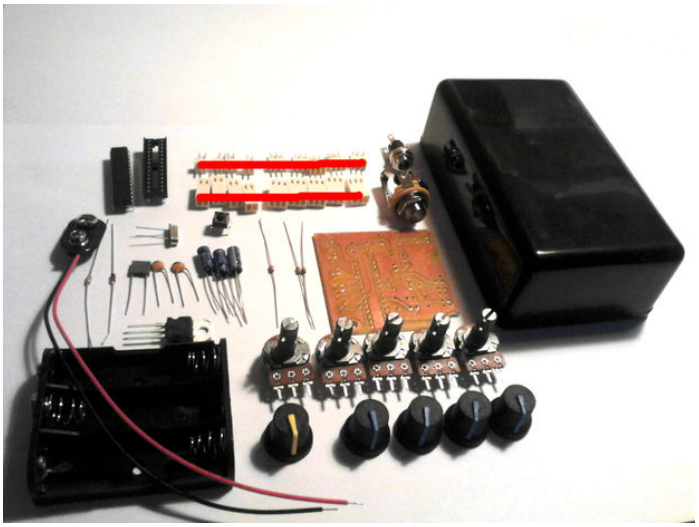
1x - 16MHz crystal

2x - 1N4148 or other general purpose diode (nearly any will work)

1x - 3mm led

1x - 3mm led holder

A few meters of hookup wire for wiring the potentiometers power jack and switch - I used 24AWG red, black and blue



Step 3: Schematic and Board

To make the schematic and board I used a free program CadSoft Eagle which I would recommend to anyone making schematics or boards.

link: Download CadSoft Eagle

The Eagle files and PDF of board can be downloaded on the bottom of this step.

EXPLANATION OF SCHEMATIC

The schematic is fairly easy to understand as all the difficult things are done by software in microcontroller.

The button S1 if pressed pulls the pin 1 to ground which resets the Atmega328 but when the button is not pressed the R1 pull up resistor pulls the pin 1 to 5 Volts which tells the microcontroller not to reset.

The two 22pF ceramics and 16 MHz crystall form a clock signal for the microcntroller.

The LED1 is status led which signalizes if the circuit is on. The R3 can be changed for 2.2K resistor

On the bottom is located the power supply. It's made with 7805 voltage regulator(output 5V). It was designed so that the battery voltage is not regulated by 7805 because it is not needed, we would be only wasting power. The battery voltage goes to D2 which protects the batteries from charging when the device is powered by adaptor.

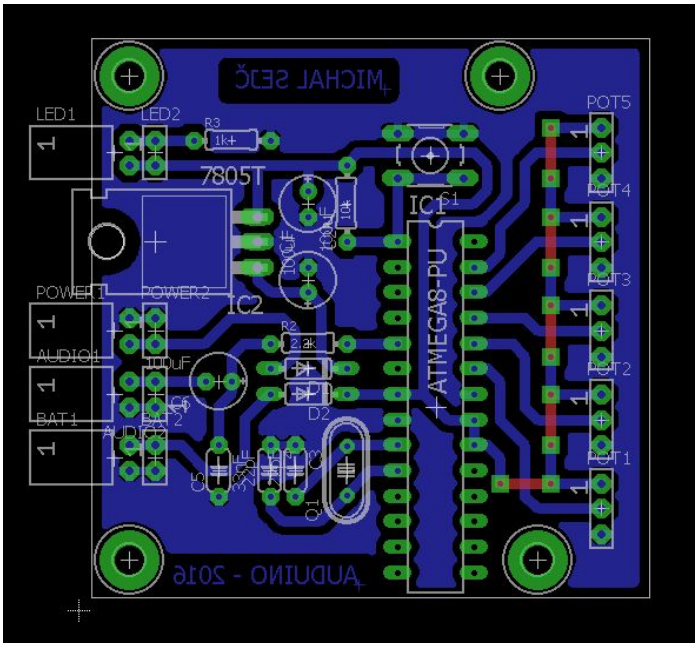
The left side connects to DC power jack and the right side connects to the batteries which **must not have higher voltage than 5 volts** othervise they will damage the microcontroller.

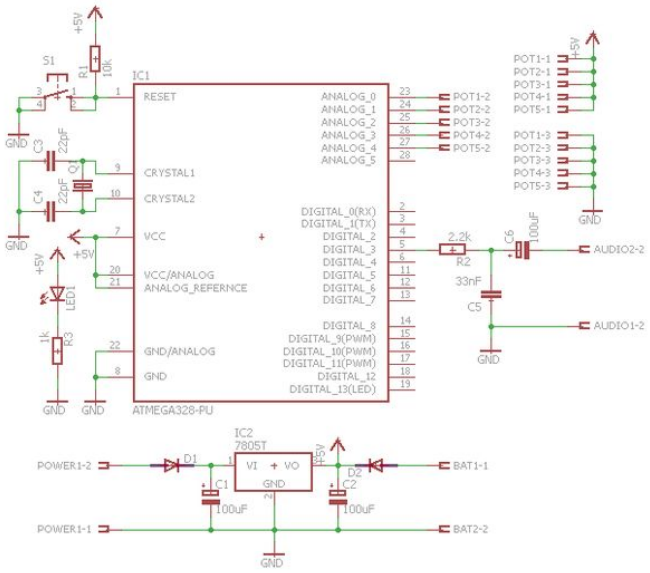
BTW: I would recommend to use BAT42 shottky diode in place of D2 instead
of 1N4148 because the 4,5V from batteries minus 0,7V(the voltage drop of 1N4148 diode) equals 3.6V which can get even lower when the batteries will discharge.

The 2.2K? resistor(R2) and 33nF capacitor(C5) are forming an RC network which filters out high frequency noises that somewhat get to the output and also lowers the output voltage to about 775mV which is standard voltage for headphones and audio equipment. Without the network the output sound would be quite messy and a bit louder.

C6 - 100µF this capacitor isolates the speaker or headphones from the output in case of some failure.(DC voltage or some other bullshit)

And finally the potentiometers are wired well like potentiometers are used to be wired. The first pin goes to V+ the middle to microcontroller and the third is grounded.





File Downloads

Arduino Synthesizer - Auduino.pdf (37 KB)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'Arduino Synthesizer - Auduino.pdf']

Arduino Synthesizer - Auduino.sch (780 KB)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'Arduino Synthesizer - Auduino.sch']

Arduino Synthesizer - Auduino.brd (86 KB)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'Arduino Synthesizer - Auduino.brd']

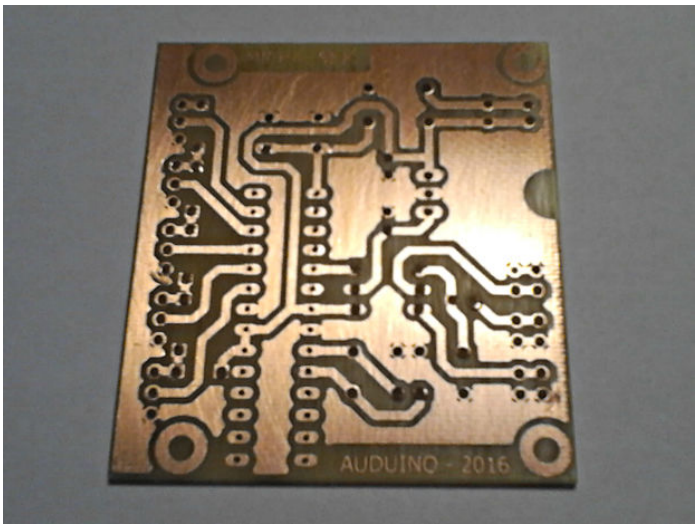
Step 4: Etching the PCB

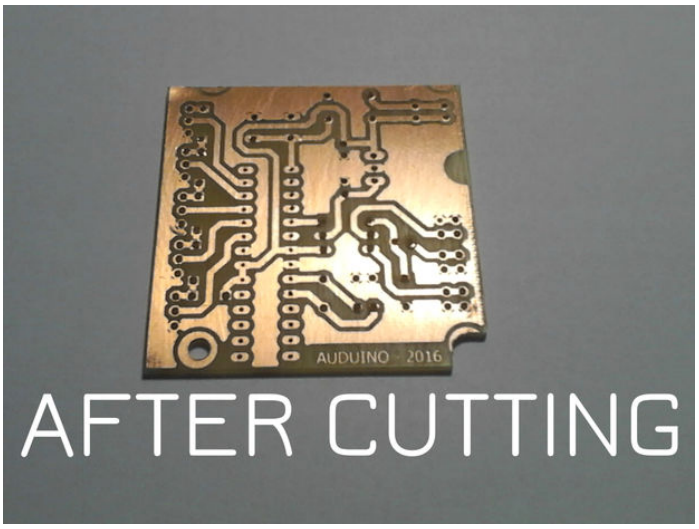
I **did not** make the video.

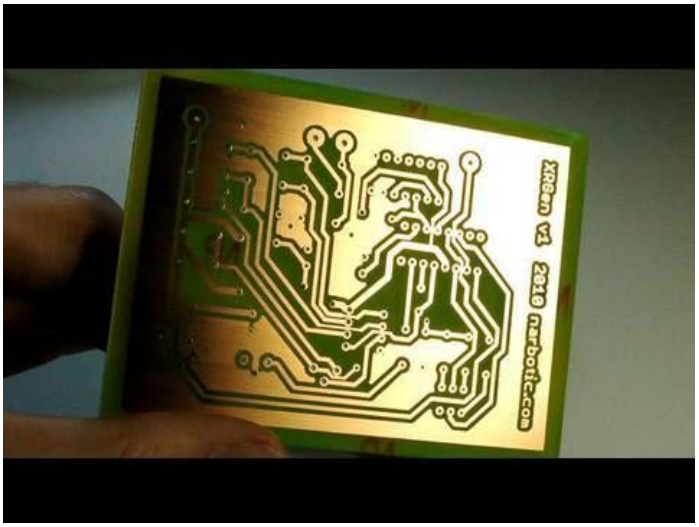
I etched the PCB with i guess "photographic method" - that would be the direct translation from my language. Well i printed out the PDF file from previous step on transparent foil and then shone through it on photosensitve PCB, then developed it in solution of sodium hydroxide and then etched it in ferric chloride. If you find it confusing just watch the video, It's very well explained.

Also **don't forget** to orient the transparent foil on the pcb so thaht you can read the words written on it.

When I tried to fit the pcb into the enclosure i realised that it's too big so I had to cut some parts that were not crucial, in this case I cut the space where were the mounting holes because I was not going to screw the PCB to box anyway.







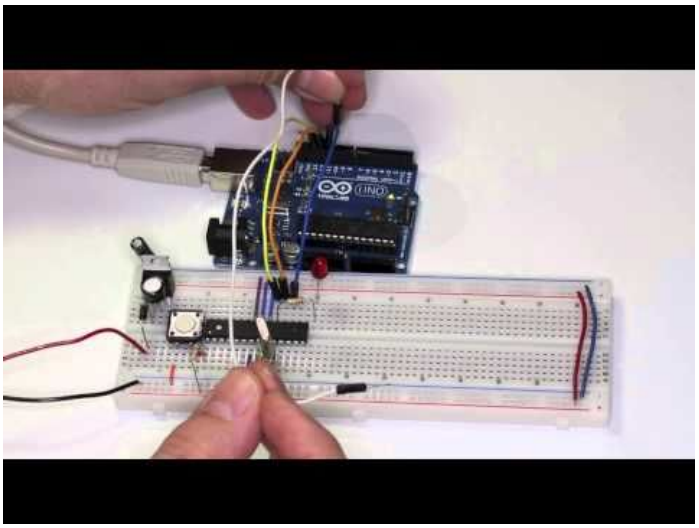
Step 5: Burn the Bootloader and upload the Program

I **did not** make the video.

What is a bootloader you ask. Well it's something like an operating system for arduino which allows you to do all the awesome stuff with arduino IDE. Without it you would not be able to use all the fancy libraries but would have towrite programs in pure C.

There are many ways to burn the bootloader. In the video above you can see using arduino as ISP to program a blank atmega328 chip though this method did not work for me so I used USBASP programmer which is basically a preprogrammed small arduino chip solely used to program other avr microcontrollers.

Once you have succesfully burned the bootloader you can plug your chip directly into arduino(replacing the original chip) and program it as usual. Then take out the chip and insert it into your homemade pcb.

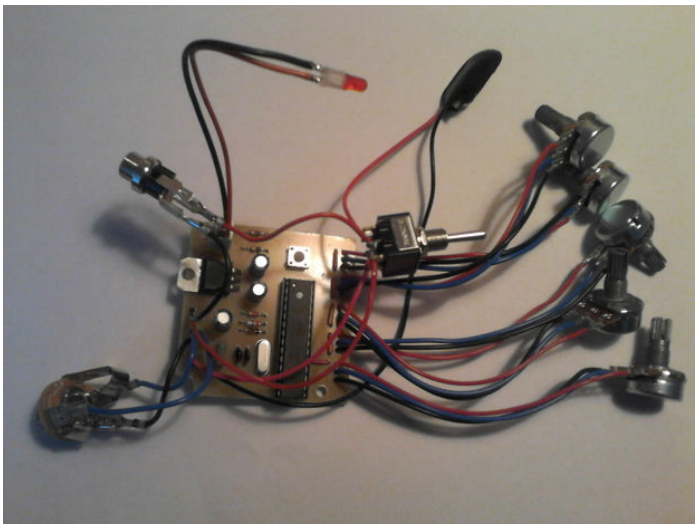


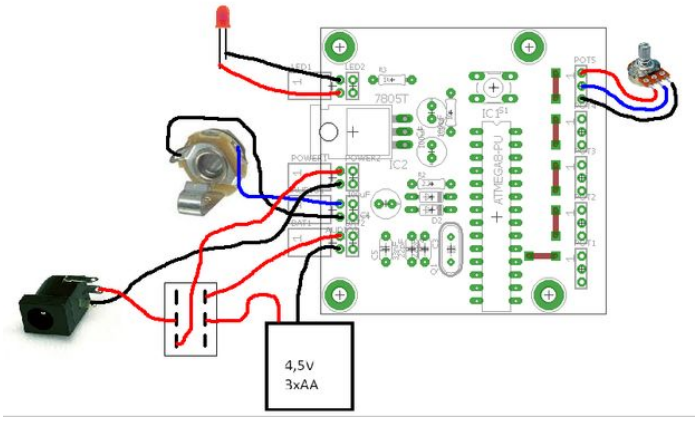
File Downloads

auduino.txt (6 KB)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'auduino.txt']

Step 6: Working Circuit

If you come this far by now you should have your PCB soldered. Now comes the fun part of wiring all the components that will be attached to the front panel. The image should clarify the wiring. I drawed only one potentiometer as all five are wired the same way. The rectangle with 6 pins is our 2 throw 2 pole switch.





Step 7: Drilling the Enclosure and fitting everything in

For me the drilling is always the fun part because I often don't have the proper drill bits but when I have them I can not put them into my small hand dril anyway so I drilled most of the holes by turning the drillbits with pliers.

When you're finished drilling you can just screw everything in there and enjoy your awesome DIY synthesizer!

Related Instructables

Adding CV inputs to the Auduino granular synth by Chuck Stephens

Auduino (Lo-fi Synth for arduino) by TobaTobias

The Arduino Synthesizer by audreyobscura

Intel Edison and Ridemakerz RC Chassis Robotics Platform by Chuck Stephens

Adapting electronic percussion projects to interact with my Lunetta open patch CMOS noise machine by Chuck Stephens

BUGGO Arpeggio Synth by rczarnik

Comments

1 comments **Add Comment**

seamster says:
Nicely done! Great first instructable too. Can't wait to see what you make next :)

Jun 27, 2016. 2:31 PM **REPLY**