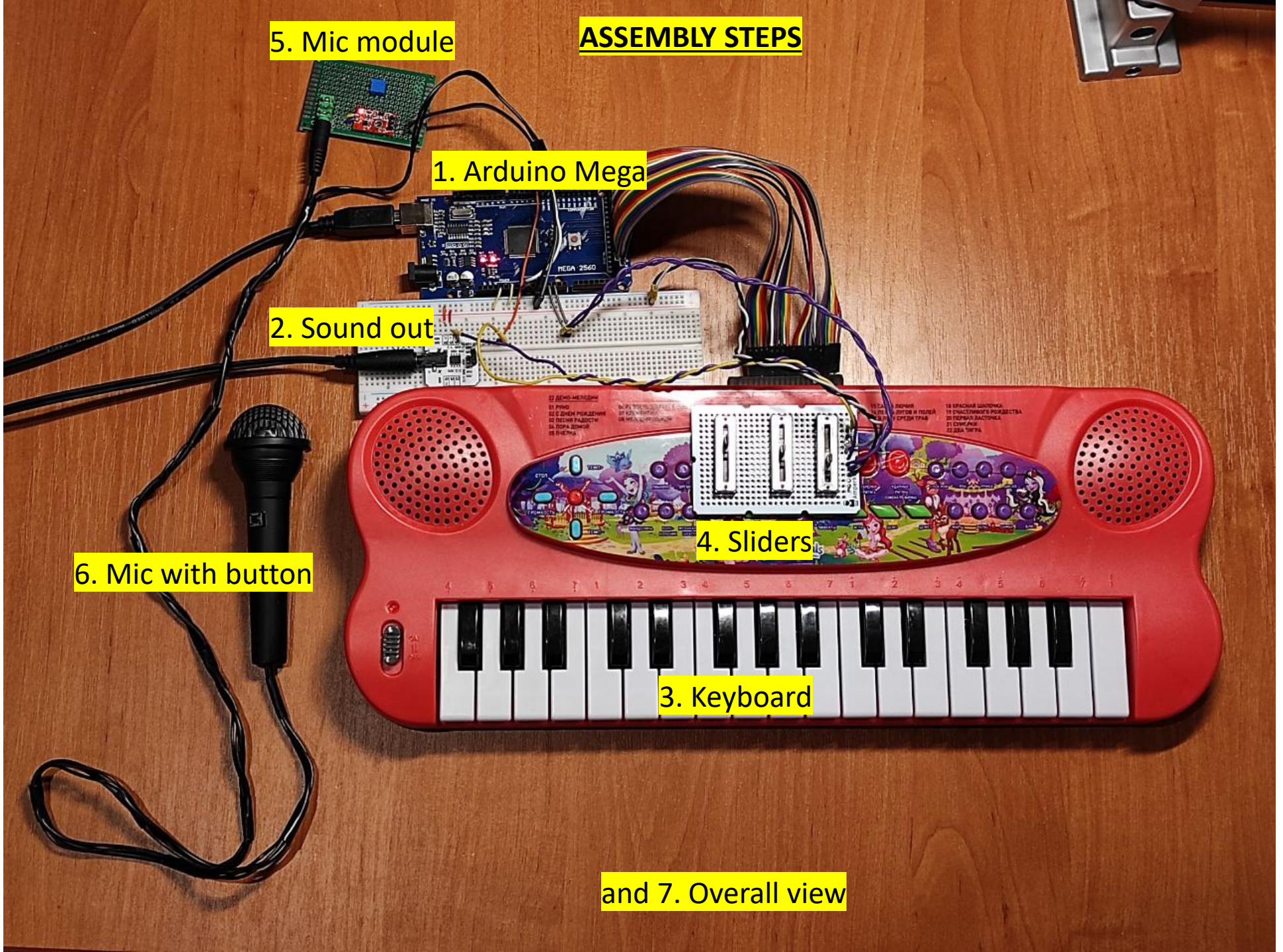
Assembly Instructions for **EndlessSynth v.04 DIY Synthesizer**

version *04_SineSynthArpegiatorMega_Sliders_Boombox*

13 Feb. 2021

Project info: https://github.com/EndlessBits/EndlessSynth
Project is created by Endless Attractions museum https://eamusem.com


ASSEMBLY STEPS
5. Mic module
1. Arduino Mega
2. Sound out
4. Sliders
6. Mic with button
3. Keyboard
and 7. Overall view

The **EndlessSynth** v.04 is a 1-bit DIY synthesizer intended to sound experimenting free of obligations to create “pure” sound. Instead, it’s based on two ideas: use “1-bit sound” and “creat it as simple as possible“. As a result, the resulted synthesizer is made using very simple components, have short and clean code and generate harsh 1-bit “old-game sounds”.

04_SineSynthArpegiatorMega_Sliders_Boombox

Synthesizer for Arduino Mega and toy keyboard with two sliders controlling 1-bit dithering algorithm. Output goes through buzzer or "Troika" audio amplifier to minijack. Also to A0 connected toy microphone coupled with potentiometer to output to A0 512 in silence. Also connected button from microphone. When button is pressed is sends signal instead synthesizing. So synthesizer works as boombox 1-bit engine too.

The second slider is kind of mic sensitivity. Hint: setting both 1-bit sliders to zero and just presing button generates glitch sound! But to normal mic sound - increase both 1-bit sliders.

- sine wave synth, without attack and release
- two sliders connected to A4 and A5 controls 1-bit algorithm.
  - slider 1: "step" - kind of "level of sound generation", the higher - the more dithering
  - slider 2: "duration" - duration of dithering, the lower - the more harshness
- arpeggiator ("note" and "string" keys)
- harsh sound
- switch octaves
- switch sample rate
- microphone with button for boombox mode

This list of features is part of README file at https://github.com/EndlessBits/EndlessSynth/tree/main/synths

Before you start

At each stage we list required components.

Additionally, you will need
- hooking wires,
- male pin connectors (need about 50),
- heat shrink,
- wires with plugs for arduino,
- soldering iron and accessories.

1. Arduino Mega

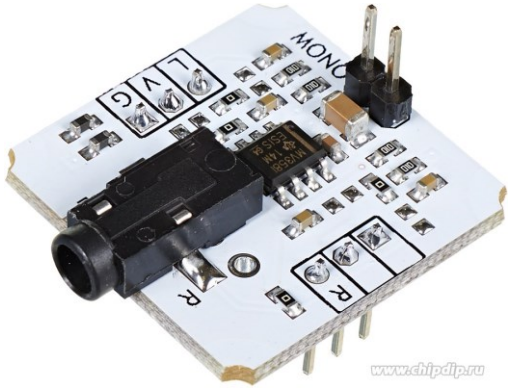
You need:
1) PC
2) Arduino Mega with USB cable

Steps:
1. Install Arduino IDE to PC
2. Download code
   https://github.com/EndlessBits/EndlessSynth/tree/main/synths/04_SineSynthArpegiatorMega_Sliders_Boombox
3. Please see comments in main sketch file with information which libraries you need to install.
4. Upload code to Mega.
5. Try to compile and upload – that's all we need to now.
   Nothing will work yet, we just test all is compiling.
   Later we will test parts of the synthesizer by running test sketches.

2. Sound out

To output sound from synth you need to choose from three possibilities, depending on synth usage:
- Buzzer is “must have” for simple immediate testing.
- Out Mini jack is required for performances and sound recording.
- Compact amplifier and speaker allows to create ready-to-use synth in one case.

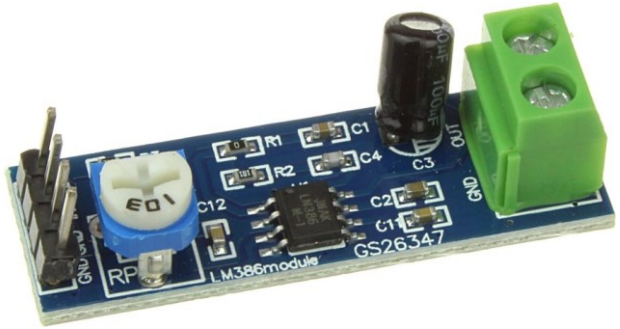
Now more details about it:



Simplest: use buzzer
You need passive buzzer, 5V. Also, choose biggest one to get better sound.

Pro: Use Troyka-Line Out Mini Jack
or similar module with mini-jack output
to connect it to your audio amplifiers or speakers.

Note: I have some problems with unwanted sound distortions from here into mixer, though speakers and sound card process this normally.



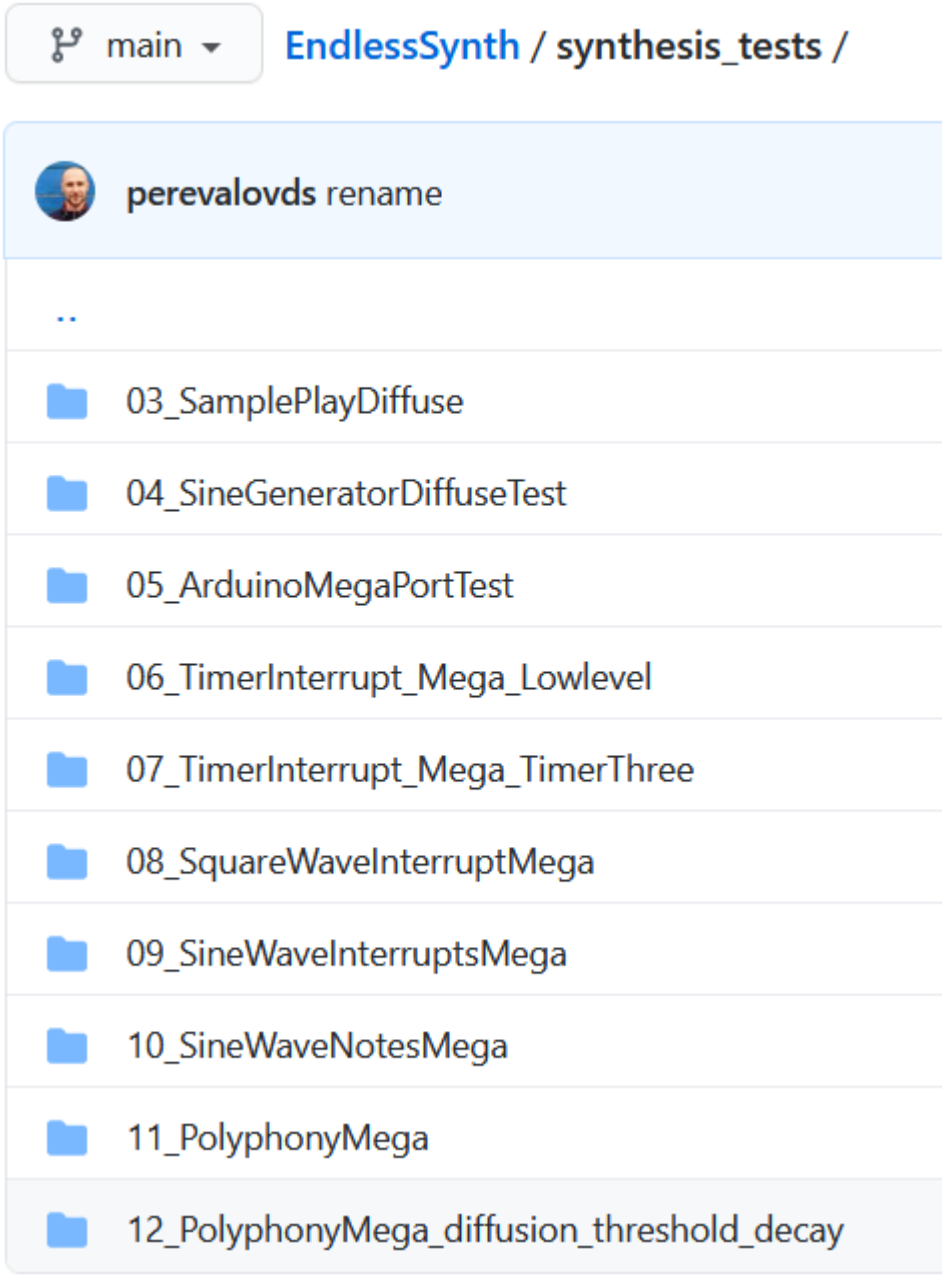
Middle: Use compact amplifier and speaker
(need amplifier, speaker, power supply).

So, choosing and connecting some sound output,
you may explore 1-bit sound synthesis!

For that check the codes:
https://github.com/EndlessBits/EndlessSynth/tree/main/synthesis_tests

This small projects starts from very simple approach
and goes to polyphonic sine wave synthesis using dithering,
Implemented on Arduio timer interrupts.

You may just run it to check the sound,
or explore and learn if you are programmer.



2. Keyboard

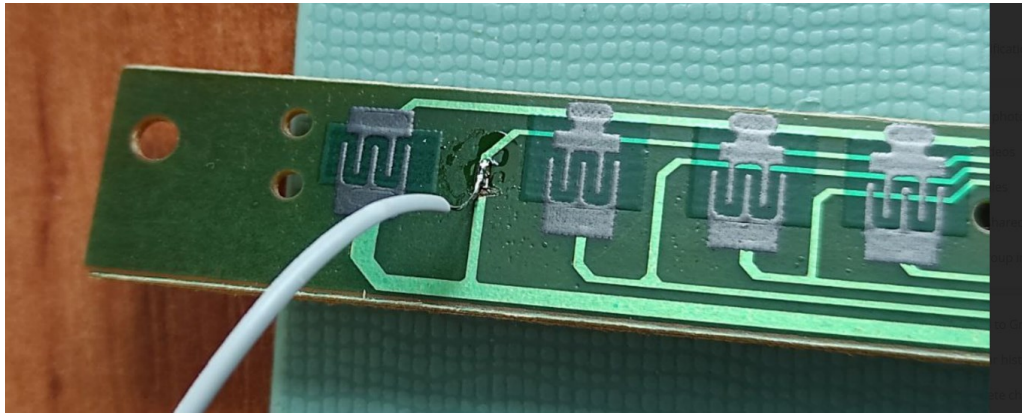
You need:

1) Cheap toy keyboard piano with 32 keys.

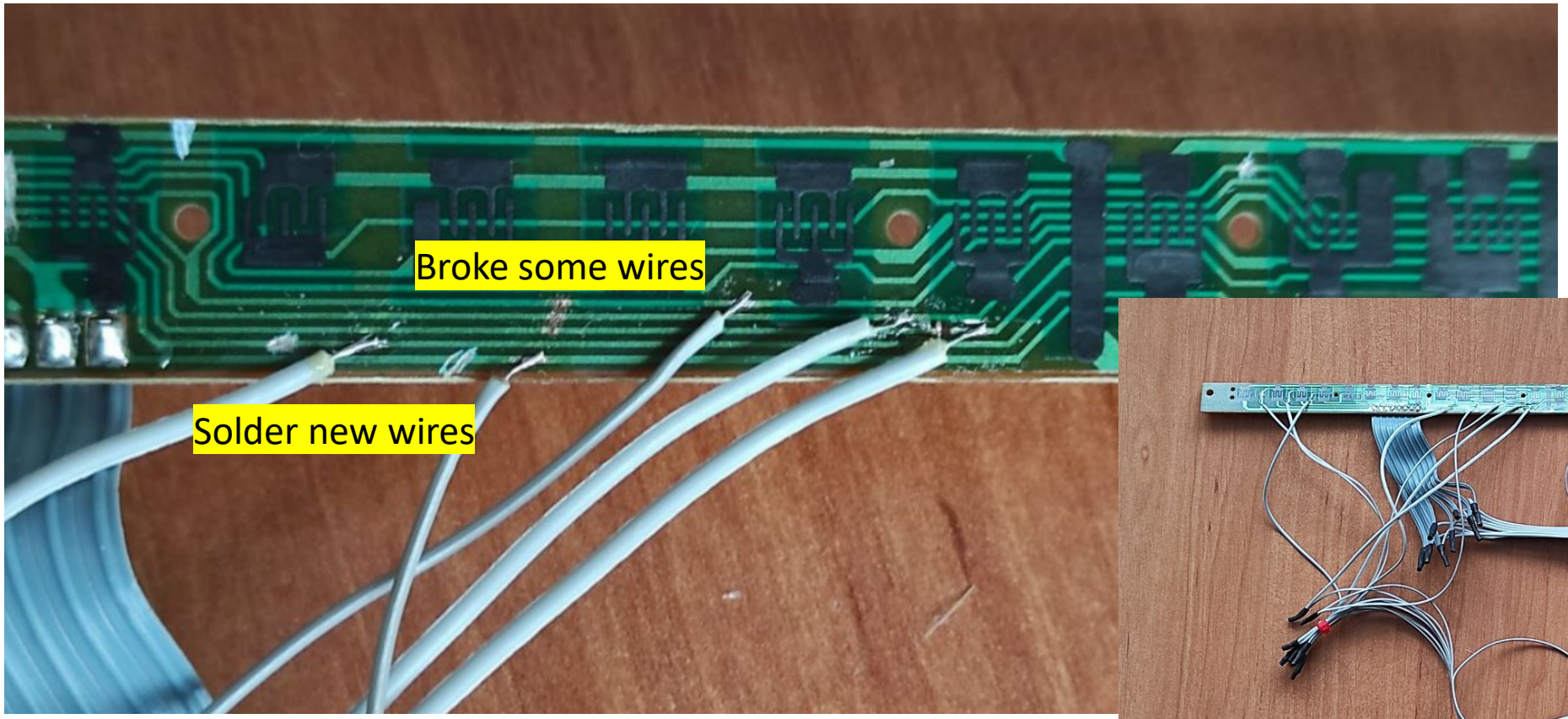
2) Ribbon cable with 40 pins
“Raspberry Pi GPIO Ribbon Cable - 40-pin, 6" (RPi 3, RPi2, B+)”

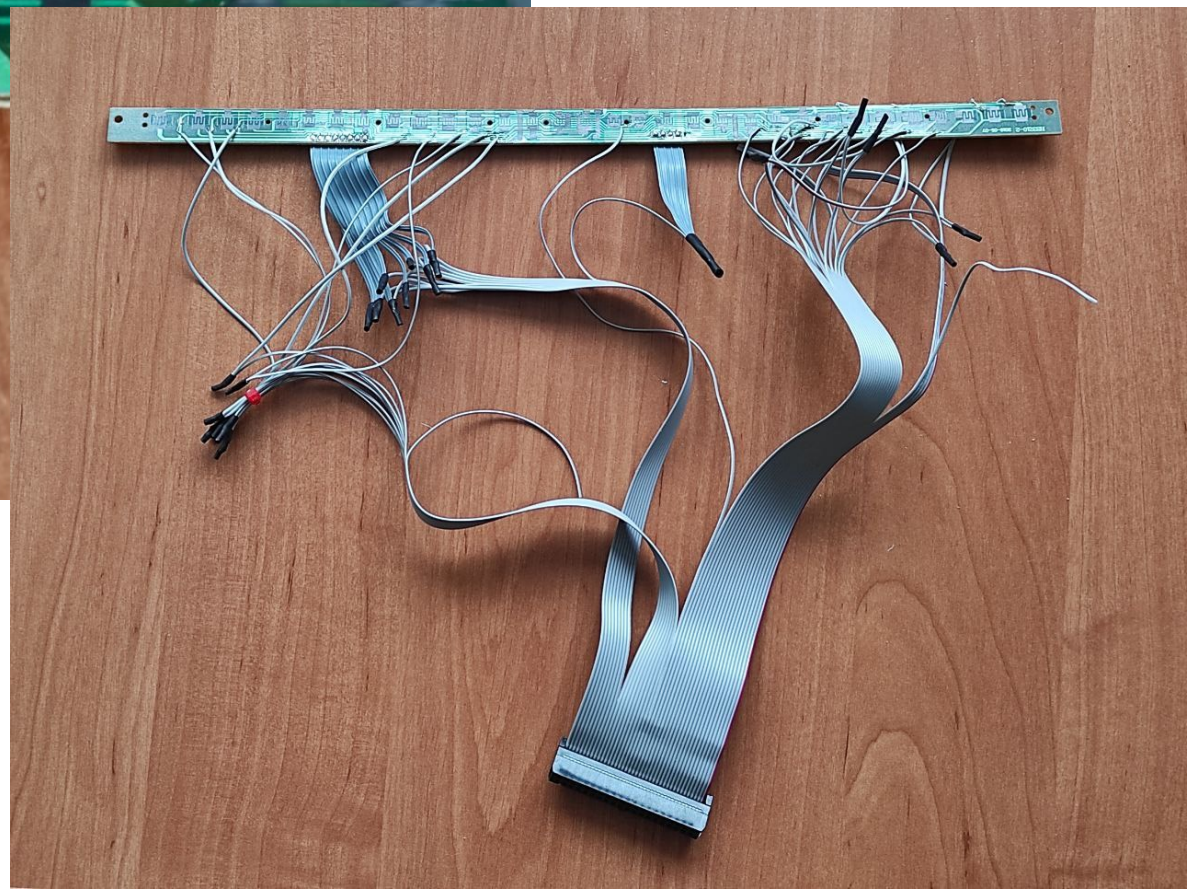
Steps:

1. Disassembly toy keyboard
2. Explore the keyboard’s sensor structure. Cheap keyboards use 4 blocks by 8 keys, in each block only one key may be requested properly. We need “truth” polyphony, so we need to solder tiny wires and break some wires on the sensor.

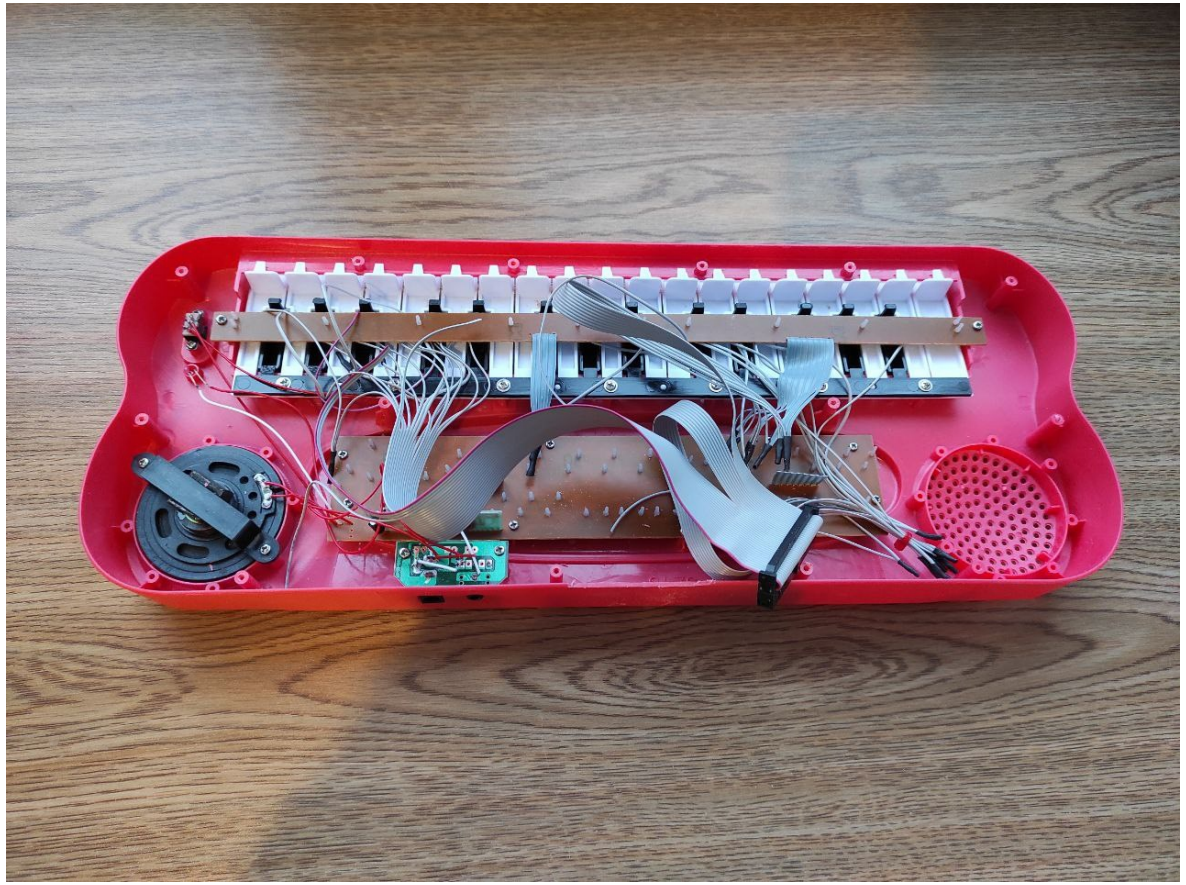


Normally we can use 8 keys from “original” keyboard scheme, and add 24 keys by soldering.





Soldering is finished



Put inside keyboard

Keyboard is ready to be connected to Arduino!

So, connect keyboard to Arduino and check it

Connect keyboard using 33 wires: 32 keys + Gnd.
This is the reason we use Arduino Mega, not Uno.

Download code and upload to Mega for testing keyboard:

https://github.com/EndlessBits/EndlessSynth/tree/main/keyboard_tests/keyboard_polyphony/01_SynthPinsTest_Polyphony

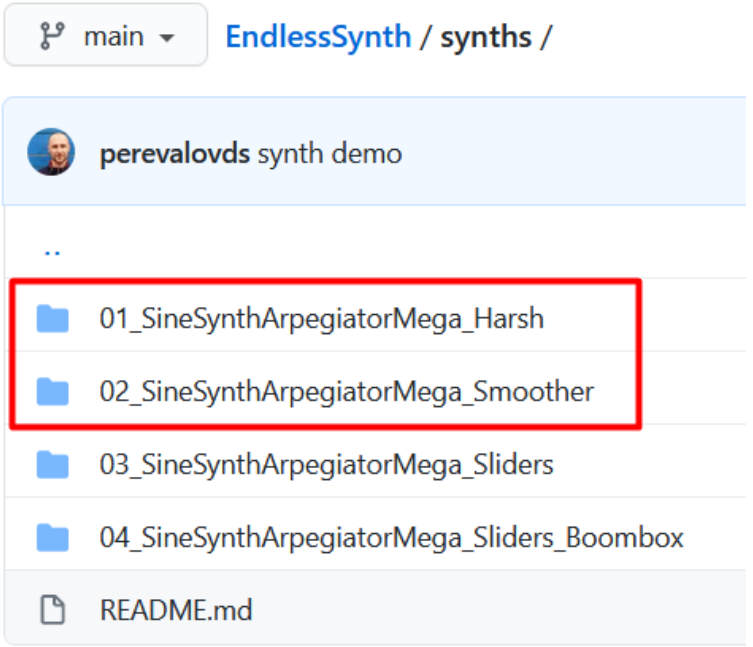
Next you may upload and play synthesizer!

It will use only keyboard and sound output:

https://github.com/EndlessBits/EndlessSynth/tree/main/synths

Two first projects – is that you need on this step.

Files contains detailed instructions how to connect
and play all – please read them carefully.



The main point there is that, well, synthesizer works as a guitar:
to hear anythong, and must press 1,2 or 3 “notes” at the left part of keyboard, and hit “strings” – three right most keys.

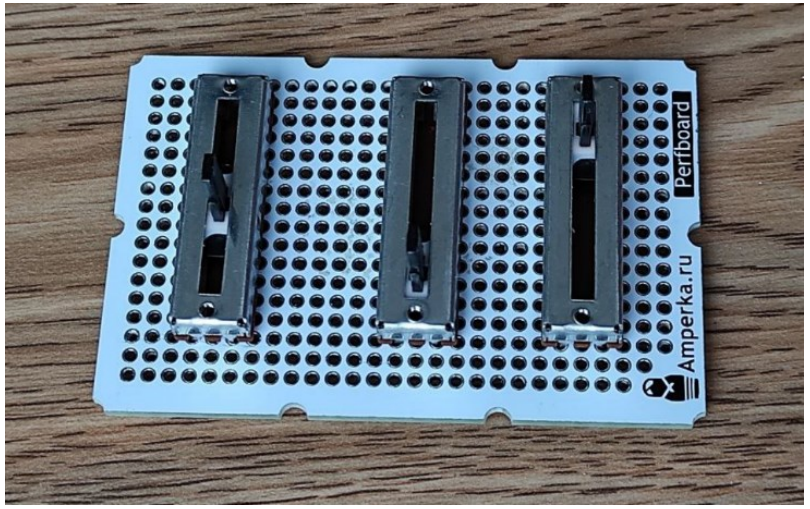
Circles on image is an example what to press to hear sound.

4. Sliders

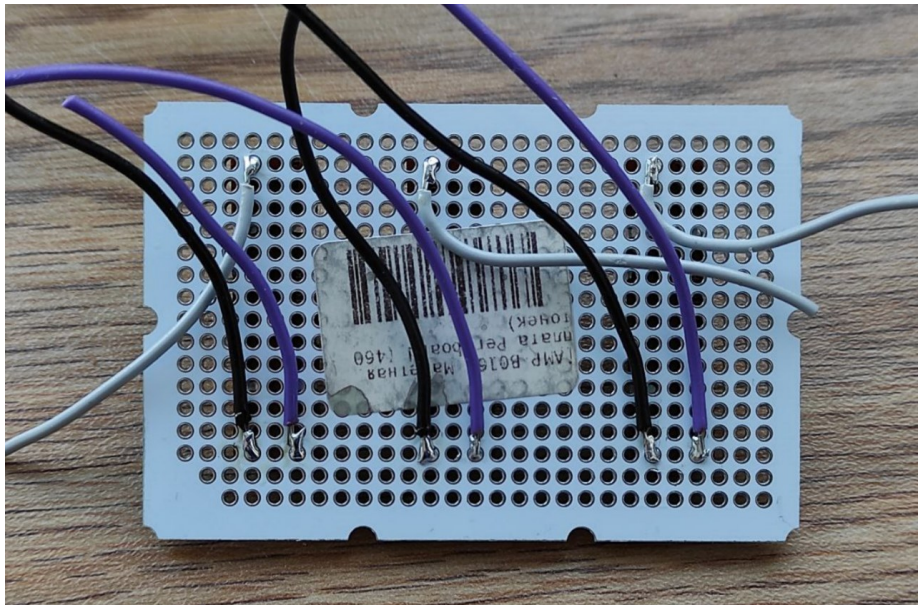
You need:
1) Three 10 kOhm potentiometers.
2) Perforated bread board.

The first pot will be used to control output sound volume, the second and third will go to analog inputs of Arduino.

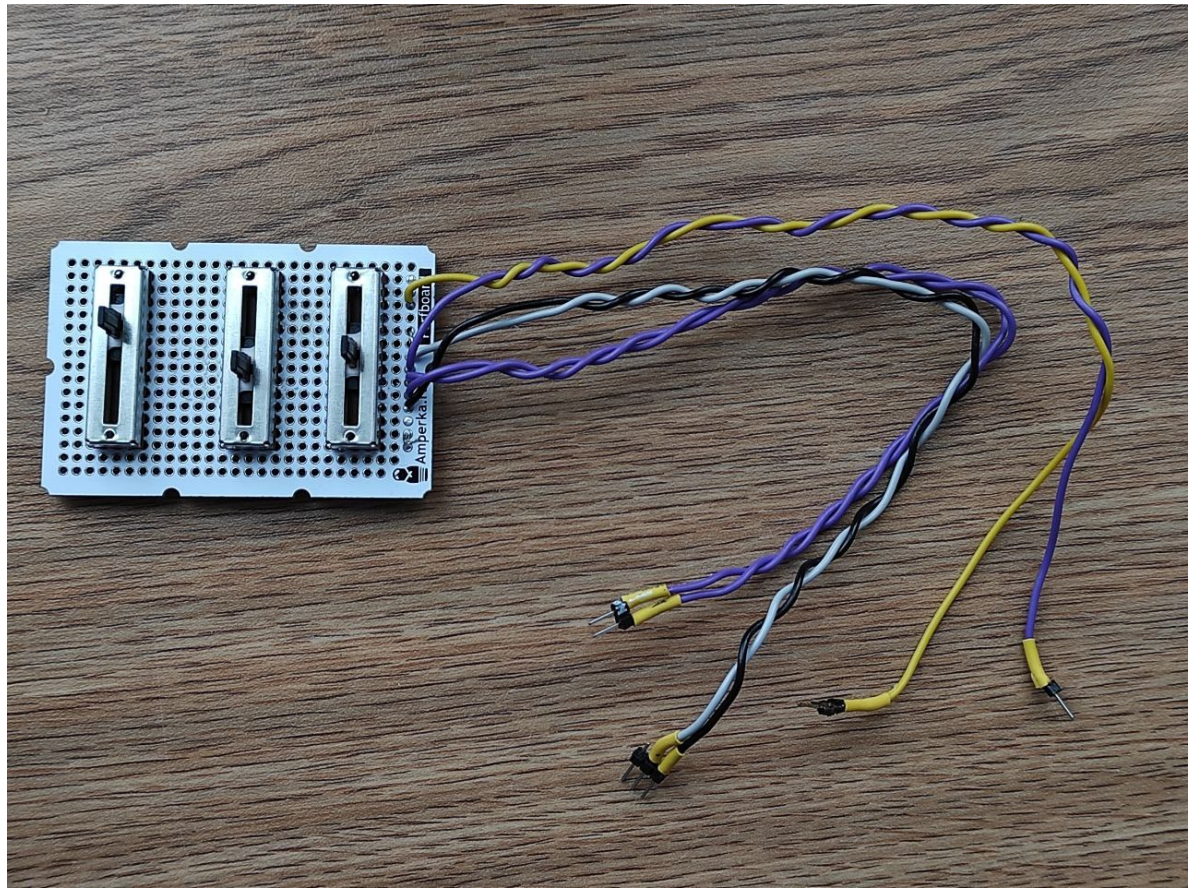
Steps:
1) Put pots to bread board
2) Solder wires

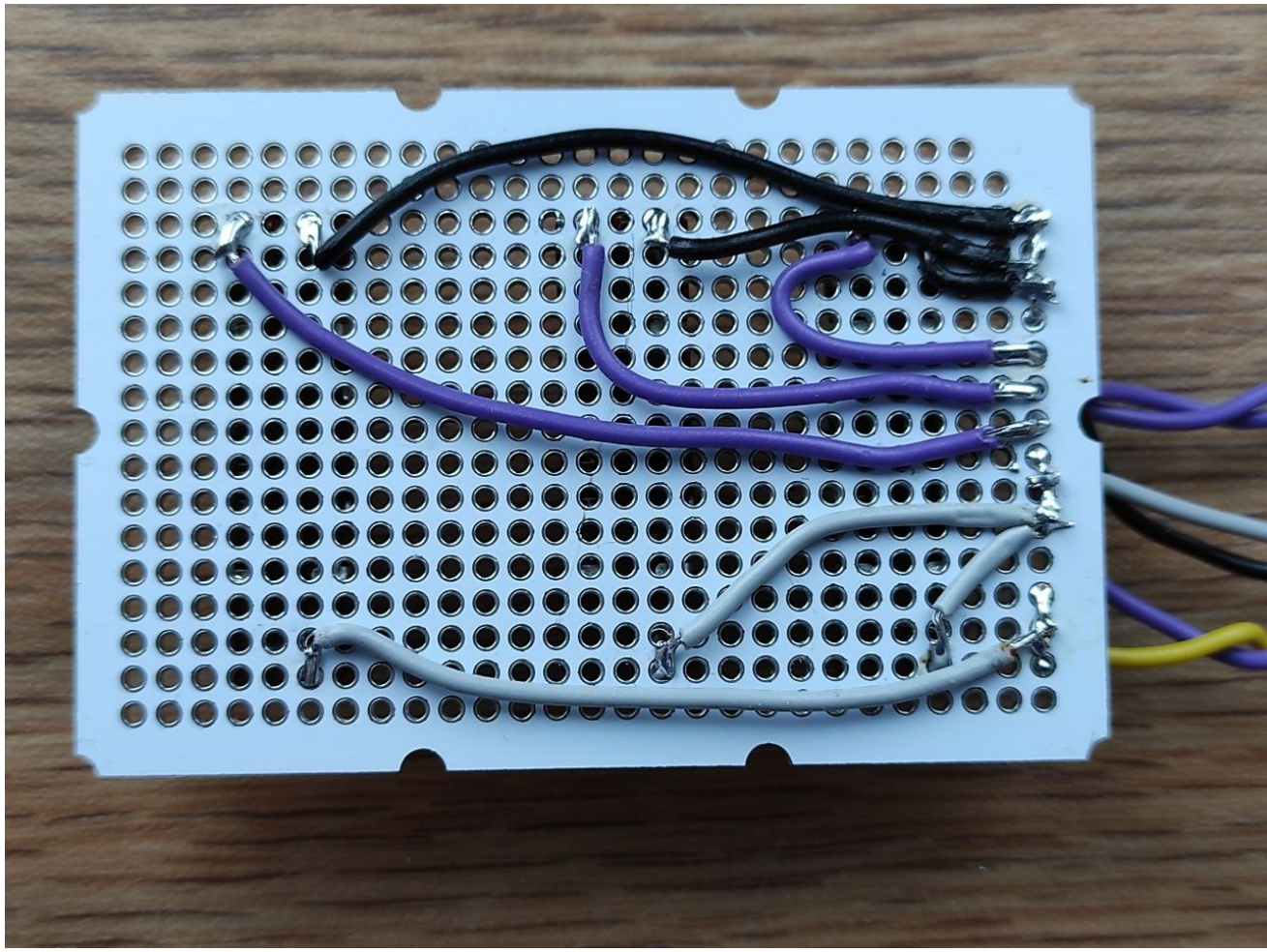


Put pots



Begin soldering wires



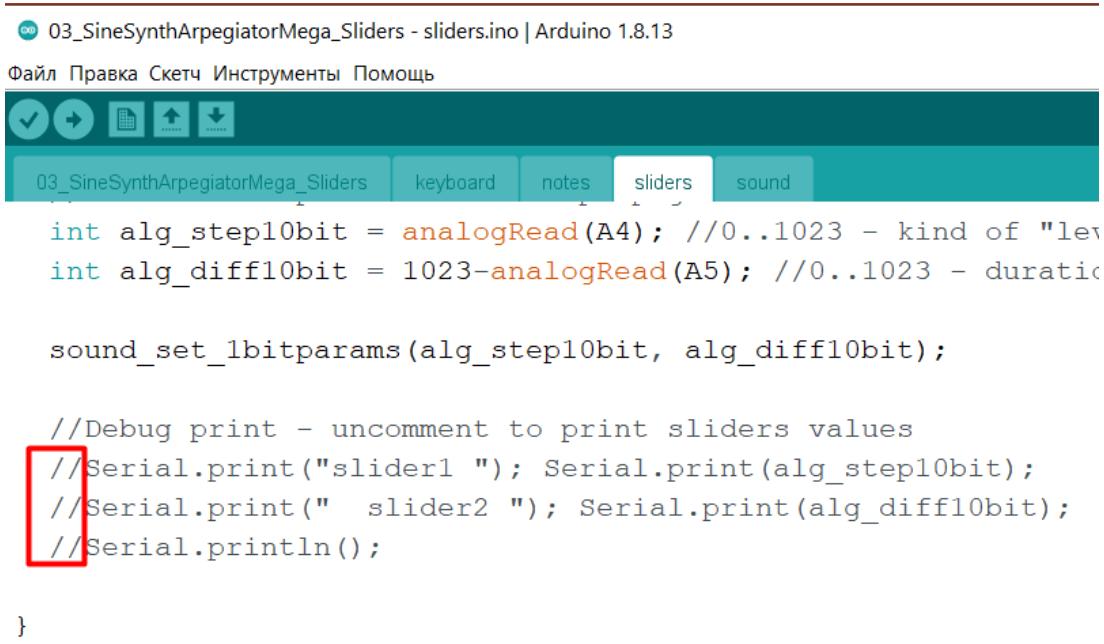


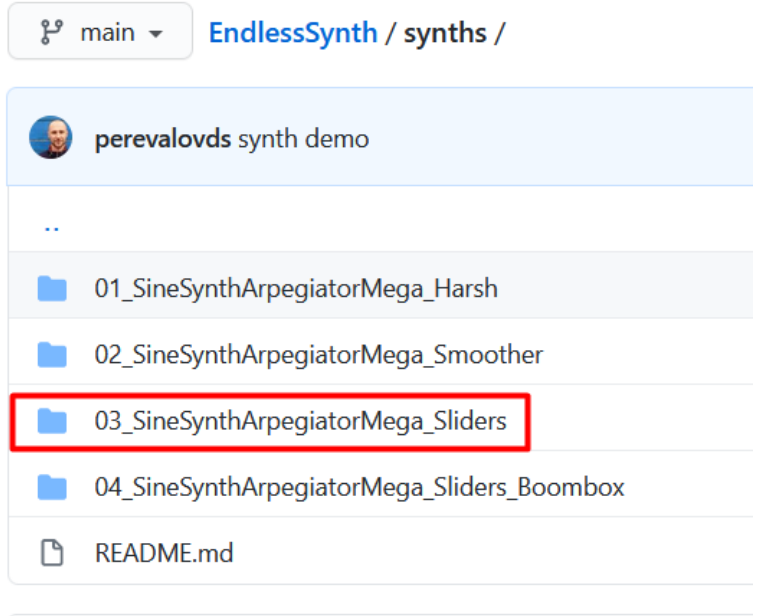
Black and white wires will be connected to Gnd and 5V
Two magenta wires will be connected to A4, A5.
Yellow and Magenta – connected to pin 2 (sound out) and to your sound output device input.

Test sliders!

1) Connect them as pointed at the previous page
2) Check that volume now regulated
3) Upload third synth. Now sound must change when you move sliders 2 and 3.

To be sure that sliders connected properly, uncomment the following three lines in *sliders* file then sliders values will be printed to Arduino Serial Monitor:





5. Mic module

You need:

1) “Standard” Arduino microphone module.
2) Trimmer resistor 10 kOhm.
3) Jack 3.5 socket adapter (mono or stereo, mono is preferable).
4) Perforated bread board.

Discussion:

Microphone unit gives -2.5V...2.5V output,
so to digitize signal carefully we need to pull up it to 0..5V range for A0.

As a solution, we connect trimmer resistor’s left and right pins to Gnd, 5V, and output to A0 too.
Next, we potentiometer to obtain 2.5V when Mic is in a silence (or, equally, 512 on A0).
Now microphone will output 0..5V to A0!

Microphone module FC 109

Trimmer resistor 3266P-1-103LF, 10 kOhm

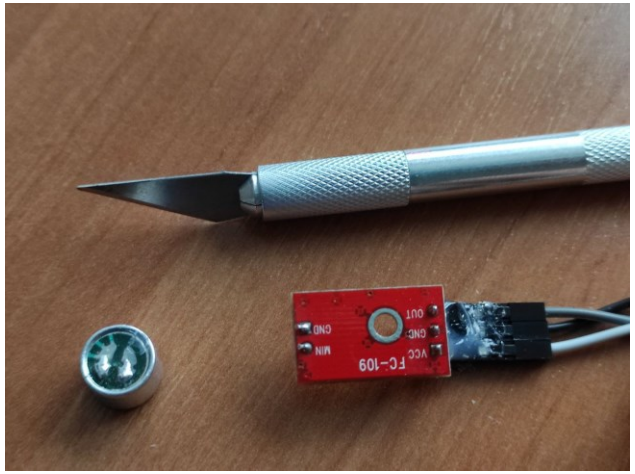
Jack 3.5 socket adapter

Steps:

Simple way: use this microphone.
Connect mic module connect to Gnd, 5V, and output signal connect to A0, and connect trimmer resistor contact 1 to Gnd, contact 2 to A0 and contact 3 to 5V (so they connected in parallel).

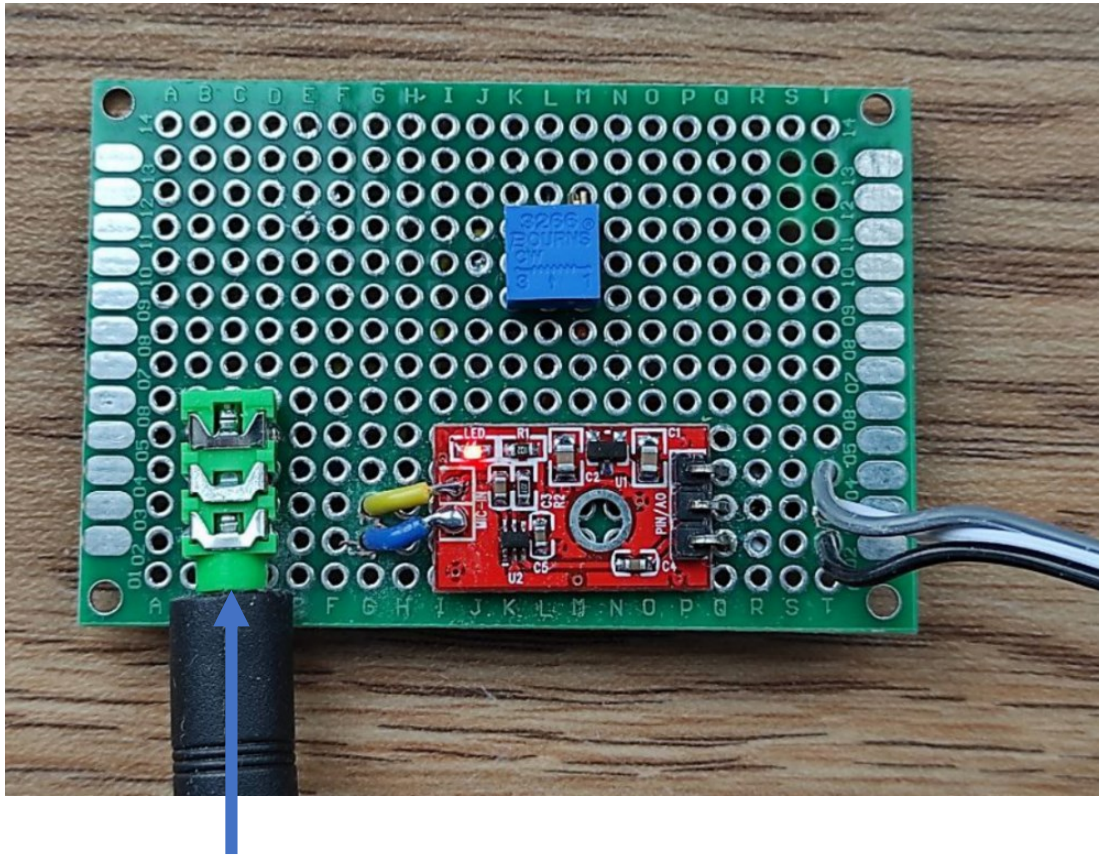
Harder way: use toy microphone
Such microphones often goes together with toy synths, so may be you already have it.
In this case,

1. Cut off microphone from module

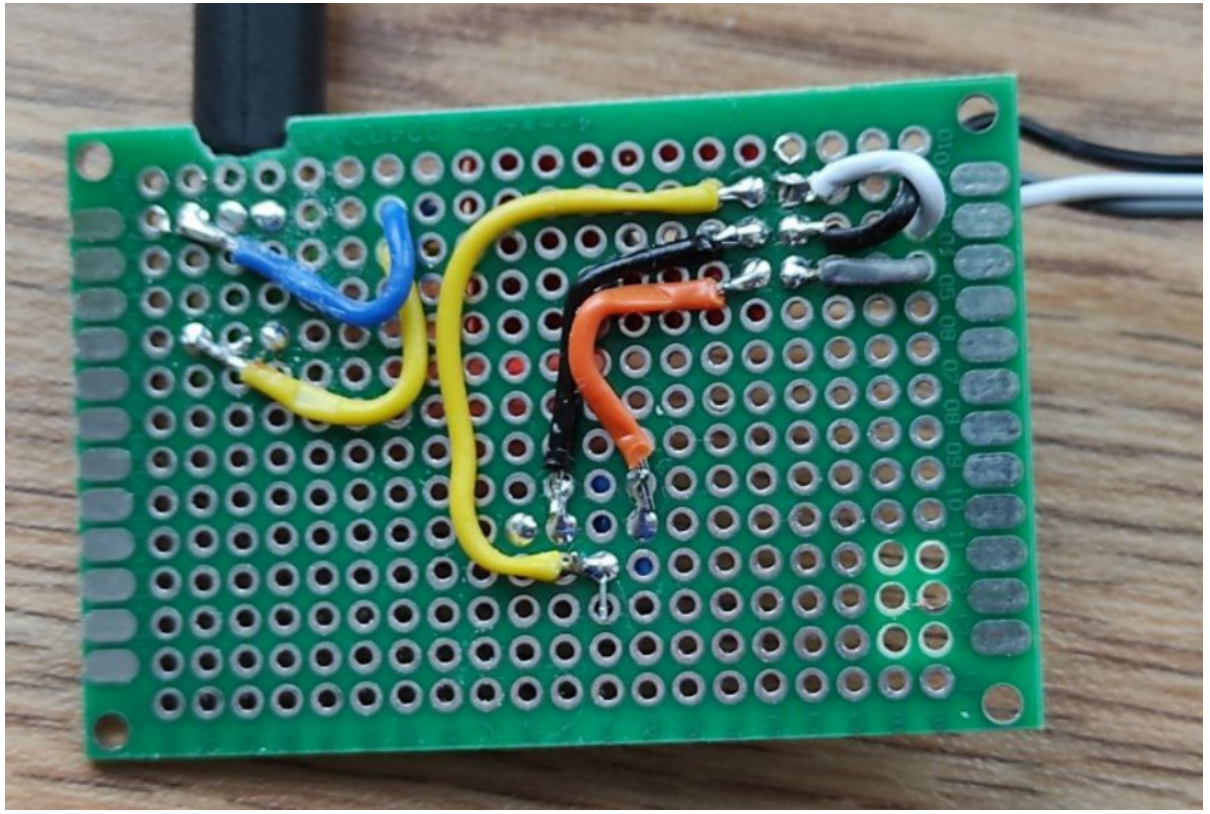


Toy microphone

2. Place module, trimmer resistor and Jack 3.5 socket adapter on bread board and solder connections
   (see the next page)



Place for connecting microphone



Wires connection: black to Gnd, Gray to 5V, White to A0.

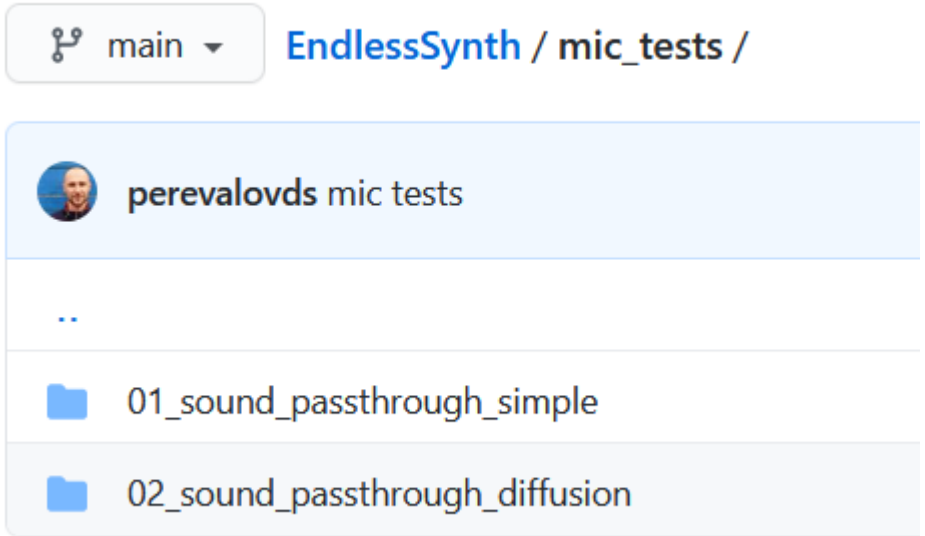
Having connected microphone module with microphone (toy or original) to Arduino, now *adjust trimmer resistor*.

To do it, upload to Arduino the following sketch: https://github.com/EndlessBits/EndlessSynth/tree/main/mic_tests/01_sound_passthrough_simple

It gets input from A0 and converts it to 1-bit sound to pin 2 using simple comparison algorithm.

For us is important that it prints input values to Arduino Serial Monitor. So see on this values and adjust tune resistor using screwdriver so it outputs about 512 at silence.

The adjustment is ready.
Run 02_sound_passthrough_diffusion sketch for more advanced sound processing using dithering. Tell something to microhone! May be you need to do it louder to hear clear the result.



6. Mic with button

Now we can get sound from microphone. To use it in the synthesizer we need to add a button for switching “boombox mode” on and off during performing on synth.

We describe how to add a button to microphone,
as depicted on the photo, but of course you may use any other button
and not to place it to microphone.

You need:

1) Toy microphone
2) Compact button

Toy microphone with added button

Steps:

1) Disassemble the mic
2) Make holes for button’s contacts
3) Put button and solder wires
4) Assembly mic
5) Solder male pin connectors

Disassembled mic

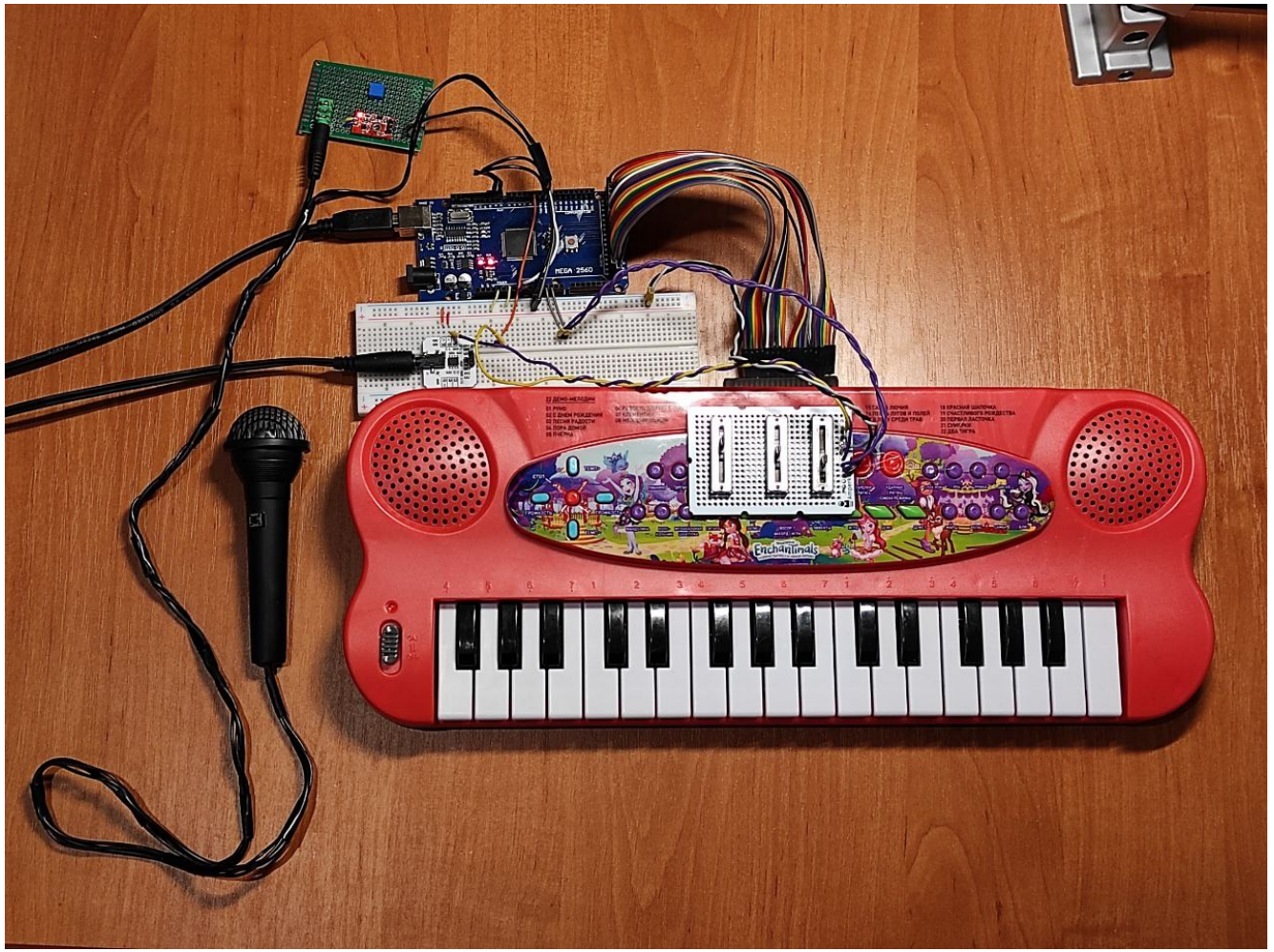
Button with soldered wires inside mic

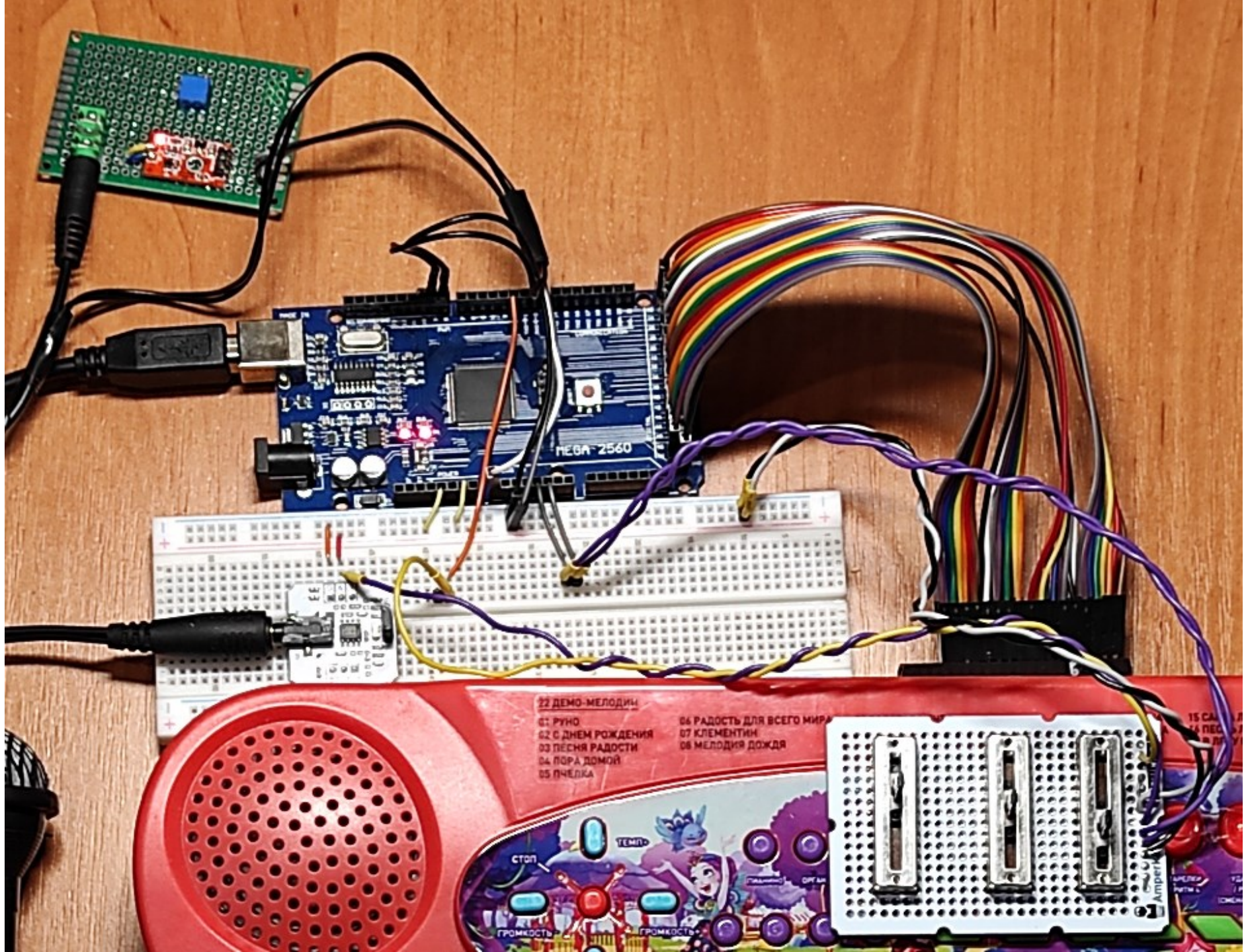
Mic with button is ready to be connected to synth

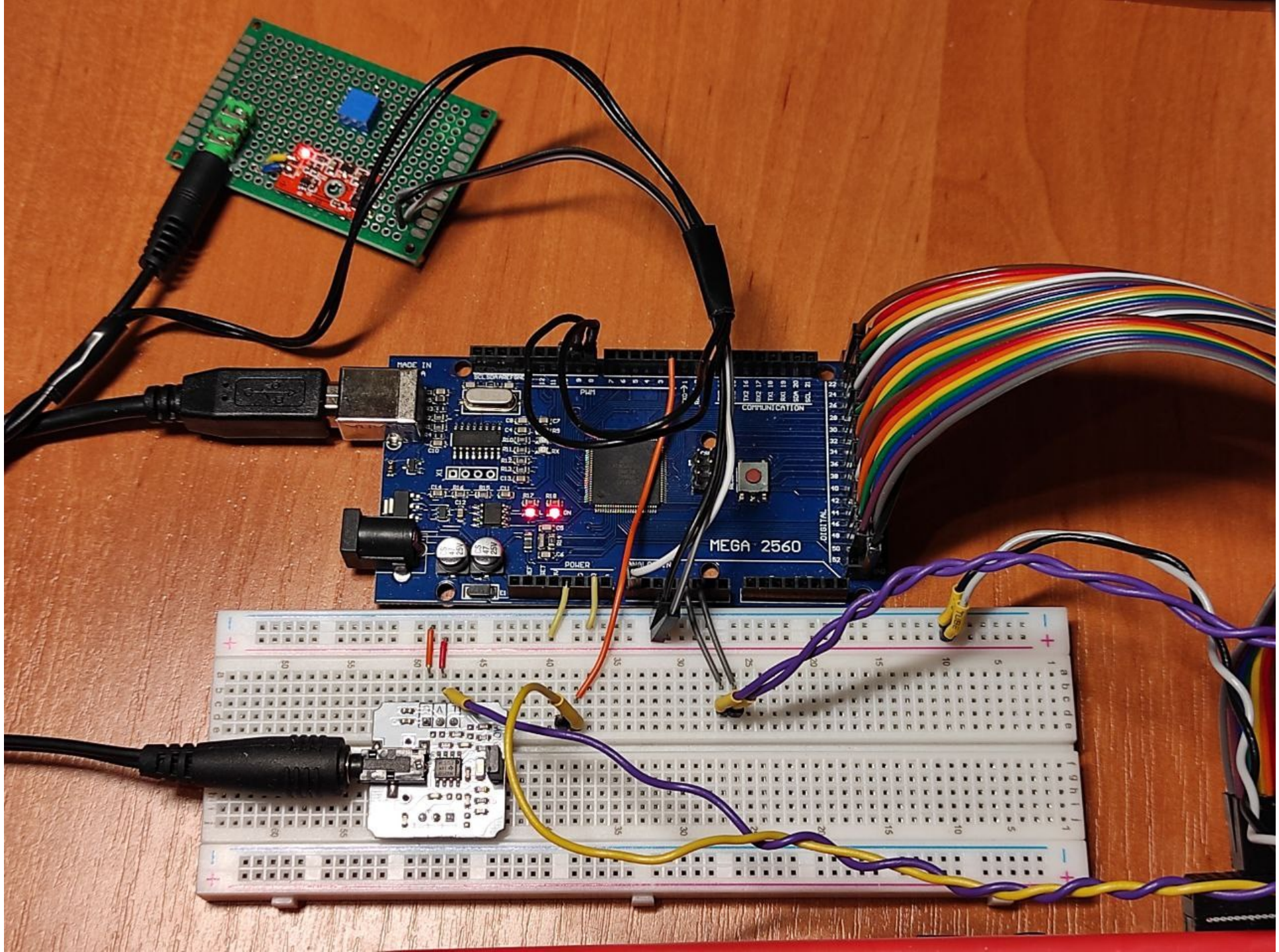
Now connect button’s outputs to pins 8 and 9. If you made all the previous steps, **the synth is ready!**

7. Overall view

The synth is ready, so let’s see once again the synth’s part connections at a once.




MEGA 2560
22 ДЕМО-МЕЛОДИИ
01 РУНО
02 С ДНЕМ РОЖДЕНИЯ
03 ПЕСНЯ РАДОСТИ
04 ПОРА ДОМОЙ
05 ПЧЁЛКА
06 РАДОСТЬ ДЛЯ ВСЕГО МИРА
07 КЛЕМЕНТИН
08 МЕЛОДИЯ ДОЖДЯ
СТОП
ТЕМП
ГРОМКОСТЬ
ГРОМКОСТЬ


MEGA 2560

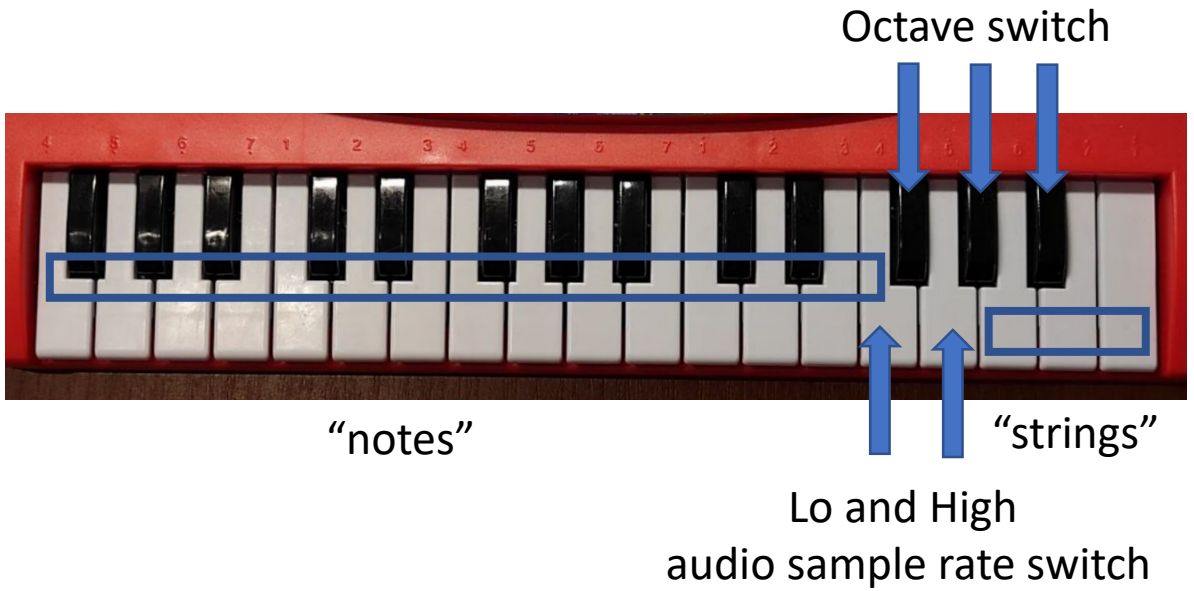
Synth hardware is completed. The final step:

Upload main synthesizer sketch to Arduino:

https://github.com/EndlessBits/EndlessSynth/tree/main/synths/04_SineSynthArpegiatorMega_Sliders_Boombox

Now start to play on keyboard and also press mic button to boomboxing!

Also, this sketch uses two more keys for switching audio sample rate – by default synth uses High, and Low makes sound “harsh”:



That’s all for now. Have fun!

Credits

We want to say thanks somebody:

Thanks Alexey Shulgin for involving us to the worls of 1-bit projects.

Thanks our colleagues in Rodchenko school online for their ideas and critics.

Thanks Michael (https://irrlichtproject.de) for his help and fruitful discussions about 1-bit music programming.