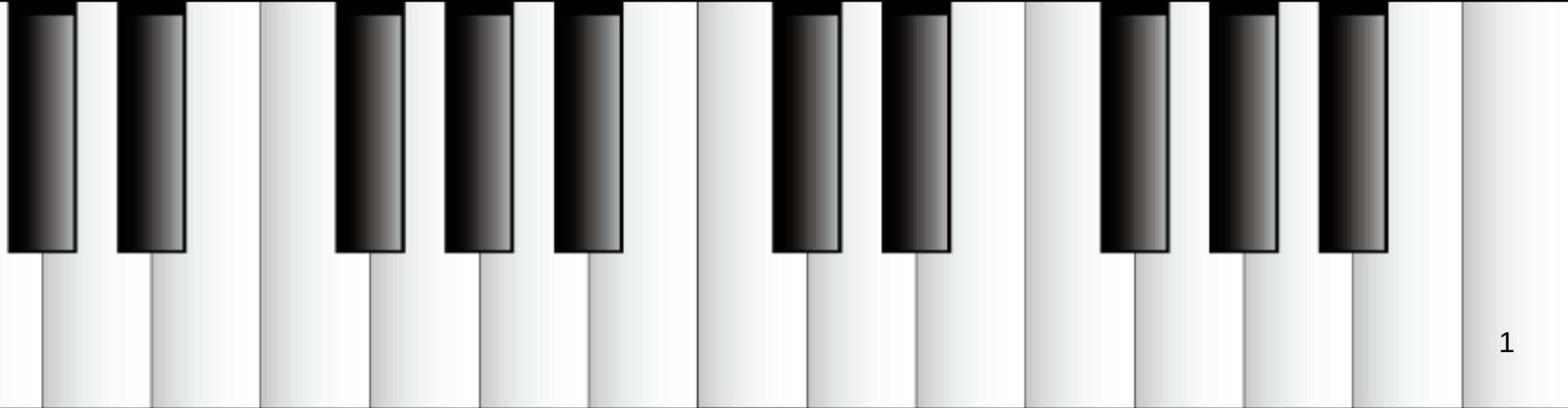


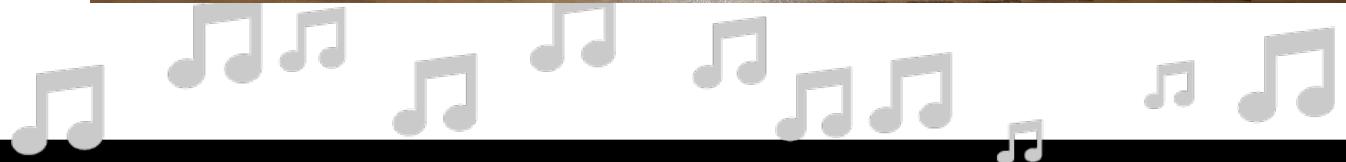
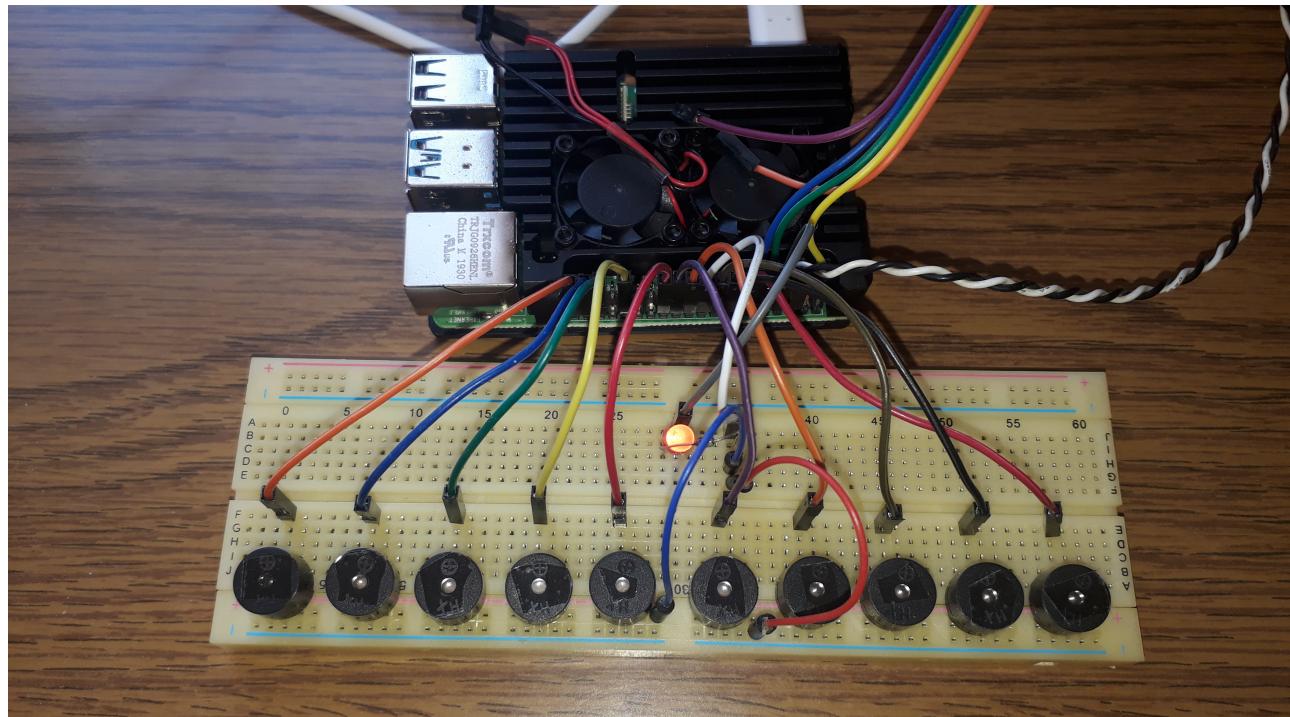
Zene lejátszása Floppy meghajtókon és Piezo csipogókon

Mérnökinformatikus MSc
Szenzorok rendszerteknikája (BMEVIETMA17) Otthoni projekt



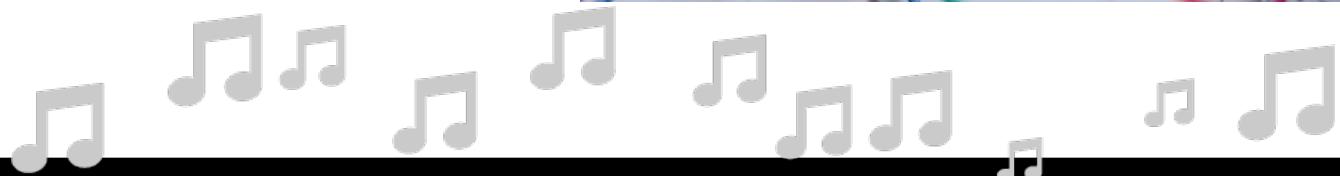
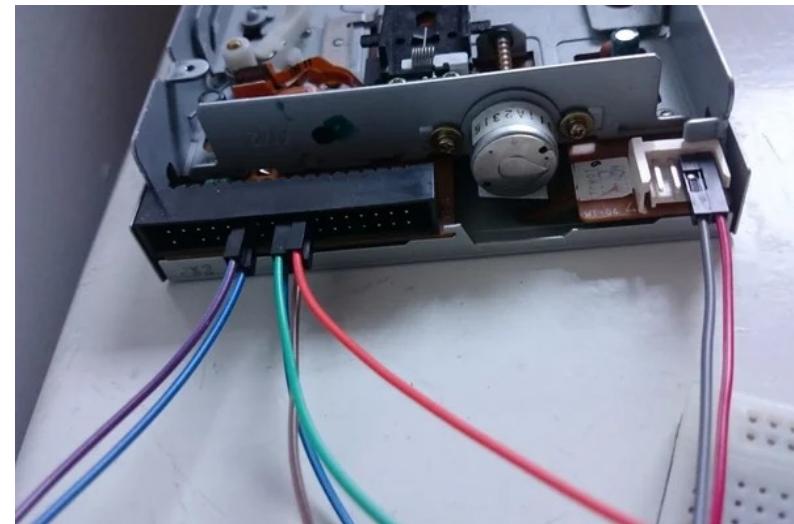
Előzmény

- Raspberry Pi 4B
- Többszólamú Piezo csipogós projekt
- Ötlet: kibővíteni többhangszeressé:
+3 Floppy meghajtó



Hardver bekötése

- Forrás:
<https://www.instructables.com/Floppy-Drive-music-w-Raspberry-Pi/>
- 2 pin rövidre zárása
- 2 pin földelése
- DIR és MOTOR pinek rákötése a RPI kimenetére
- áram ATX táppal



Szoftver

- Ezután a Floppy meghajtók léptetőmotorja a piezo csipogóhoz nagyon hasonlóan vezérelhető
- Piezo csipogó: 1111000011110000...
- Wiringpi könyvtár SoftToneCreate+SoftToneWrite függvénye megoldja külön szállal, hívható pythonból is: egyszerre sok hang szólhat
- Szebb hangzásért, és egyszerűségért a DIR is folyamatosan változik:
- DIR: 1111000011110000...
MOTOR: 1100110011001100...
- 2 SoftTone meghajtónként



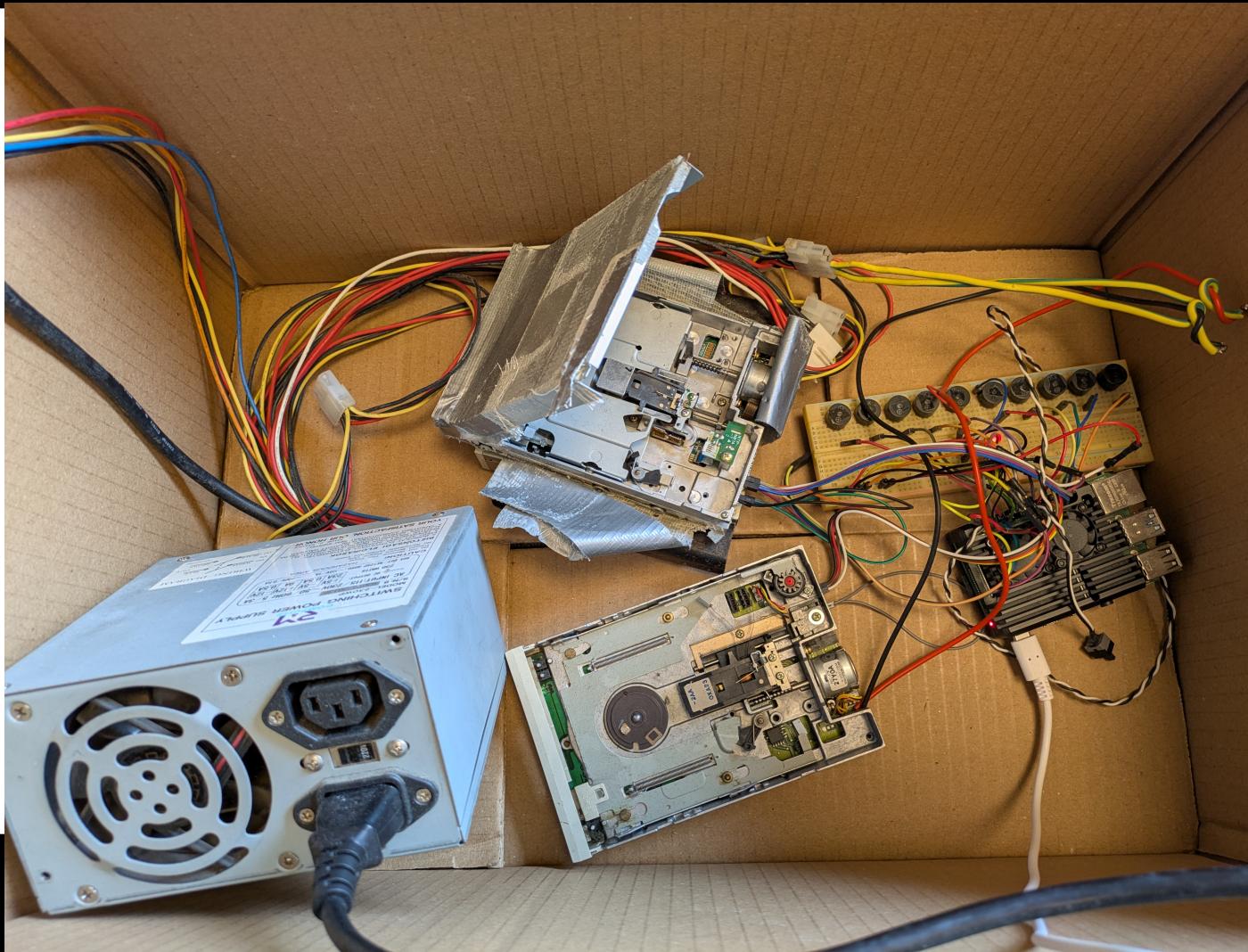
Szoftver #2

- MIDI fájlformátum: nem hullámos, hanem hang kezdet és -vég
- szólamok szétválasztása: track vs channel
(bal kéz-jobb kéz? más hangszer? nem egyértelmű)
- nehézségek a hangmagasságánál: csipogó ne legyen túl mély, floppy meghajtó ne legyen túl magas
- py_midicsv könyvtár (vs Mido)
- piezo csipogós kód:
https://github.com/4321ba/midi_player/tree/master



Eredmény:

- Videó:
- <https://youtu.be/kjZNSoIBvME>



Példakód: hangoló script

```
#!/usr/bin/python3
import wiringpi
import time
wiringpi.wiringPiSetupGpio()

MOTOR_PIN_1 = 11
DIR_PIN_1 = 10
MOTOR_PIN_2 = 19
DIR_PIN_2 = 13
MOTOR_PIN_3 = 6
DIR_PIN_3 = 5

motor_pin = MOTOR_PIN_3
dir_pin = DIR_PIN_3

wiringpi.softToneCreate(motor_pin)
wiringpi.softToneCreate(dir_pin)

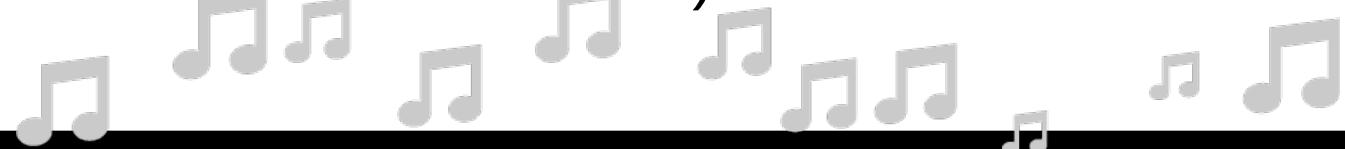
hz = 440
x = True
while x:
    tex = input("d, u, q vagy hz: ")
    try:
        hz = int(tex)
    except:
        if tex == "d":
            hz -= 1
        elif tex == "u":
            hz += 1
        elif tex == "q":
            x = False
    wiringpi.softToneWrite(motor_pin, hz*2)
    wiringpi.softToneWrite(dir_pin, hz)
```



Mérés

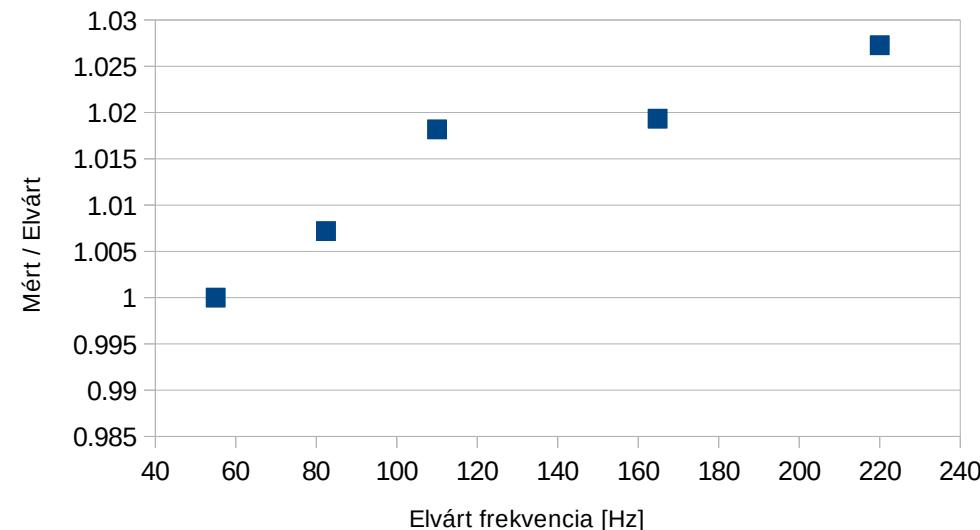
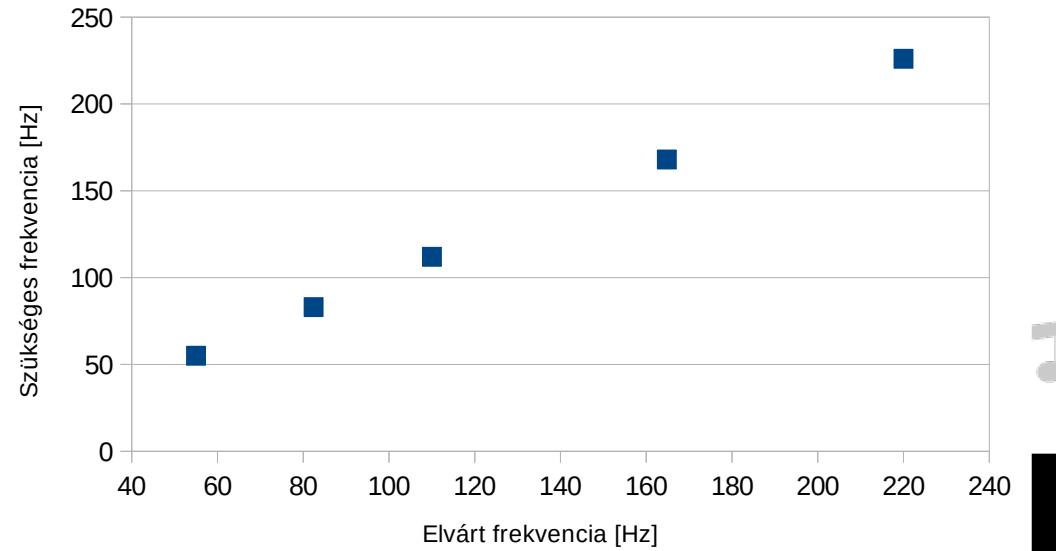
Hang	a1	e2	a2	e3	a3
Midiszám	33	40	45	52	57
Frekvencia	55	82.4	110	164.8	220
Mért	55	83	112	168	226

- Mért sor: mekkora frekvenciát kell kiadni, hogy ténylegesen a megfelelő frekvenciájú hang szóljon
- Alacsony frekvenciák: magasabbat nem tud kiadni, emiatt viszont kicsi az eltérés, tulajdonképpen nem zavaró
- Mindhárom meghajtónál ugyanolyan mért értékek (kivéve: a 3. már nem tudta kiadni a 220Hz-et)



Mérés ábrázolása

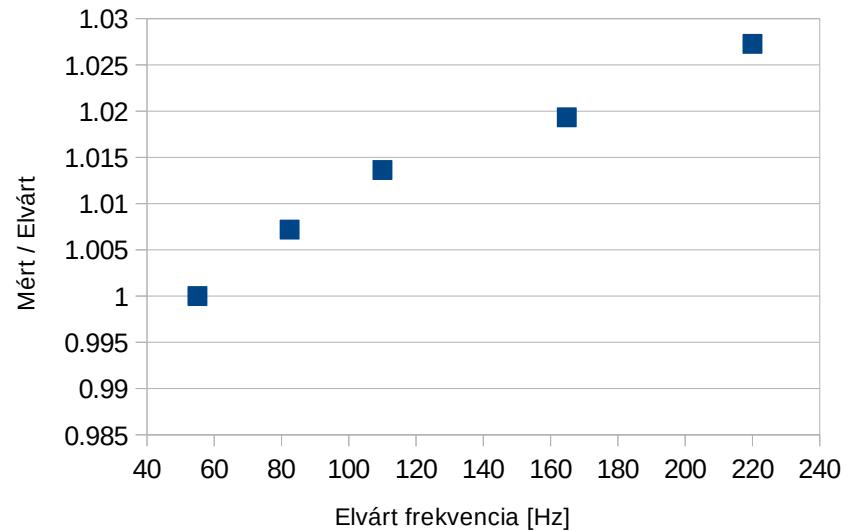
- Az elvárt, és a szükséges (=mért) értékek egyszerű ábrázolása szinte teljesen lineáris összefüggésnek látszik (bal)
- Másfajta ábrázolás (jobb): a szükséges/elvárt hányados ábrázolása az elvárt frekvenciákon



Görbeillesztés

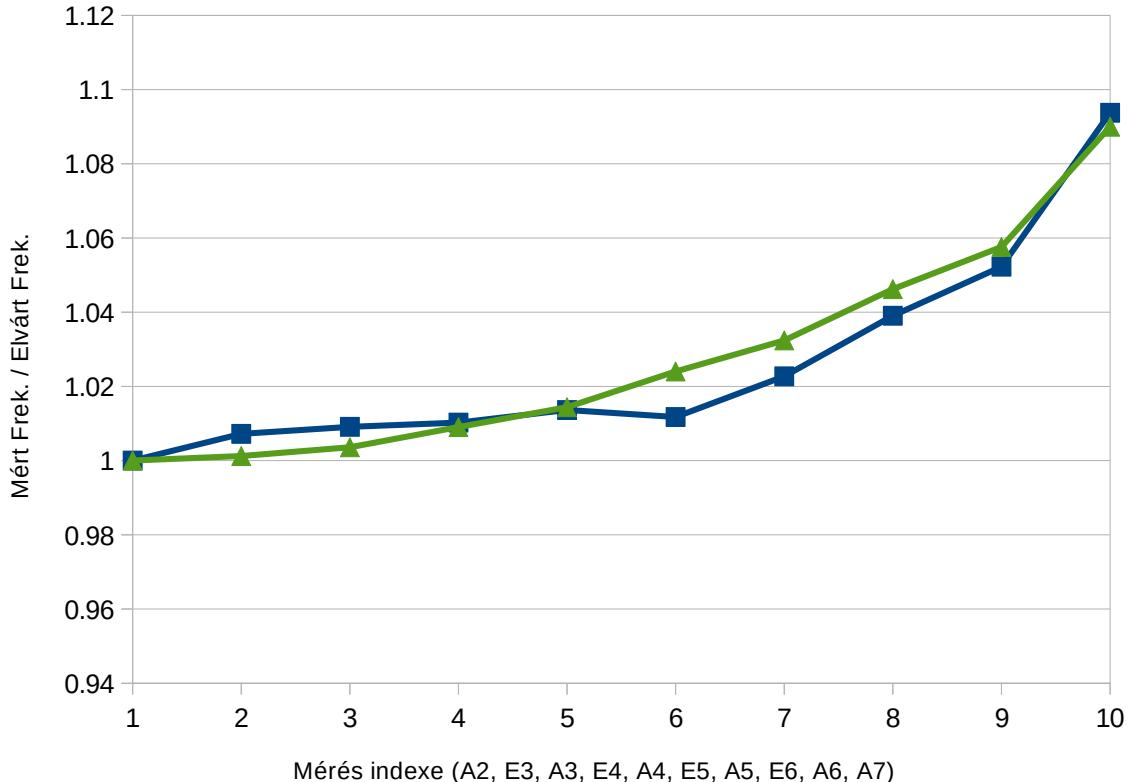
- 110Hz-es kiugró érték feltételezhetőleg azért, mert 112Hz mért érték helyett 111.5Hz lenne helyes, a könyvtár viszont nem tud tört frekvenciákat kezelni
- Ez így már esetlegesen közelíthető egy lineáris függvénnnyel
- Pl: tipp = $(0.000157*f + 0.994)*f$
- További ötletek, másfajta görbe?

Elvárt	55	82.4	110	164.8
Mért	55	83	111.5	168
Tippelt	55.12	82.95	111.20	168.03



Piezo csipogók esete

- Kék: mért hánnyados
- Zöld: illesztett másodfokú görbe
- 110Hz-től (A2)
3520Hz-ig (A7)



Köszönöm a figyelmet!

- Kérdések?

