

# Tolatókamera és radar fejlesztése Raspberry Pi 3 segítségével

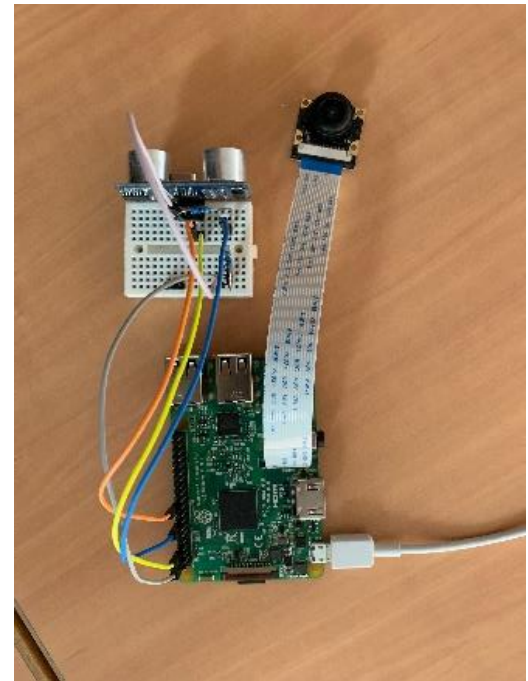
Varga Tamás  
Konzulens: Dr. Orosz György



Méréstechnika és  
Információs Rendszerek  
Tanszék

# Előzmény

- Önálló labor 1 téma
- Cím: Tolatókamera és radar fejlesztése Raspberry pi 3 segítségével
  - Feladat: Működőképes tolatást (vezetést) támogató rendszer összeállítása
  - Saját feladat:
    - Kamerakép streamelése Android applikációra
  - Két irányra vált szét:
    - Képfeldolgozás (Kristóf)
    - Távolságmérés (Tamás)

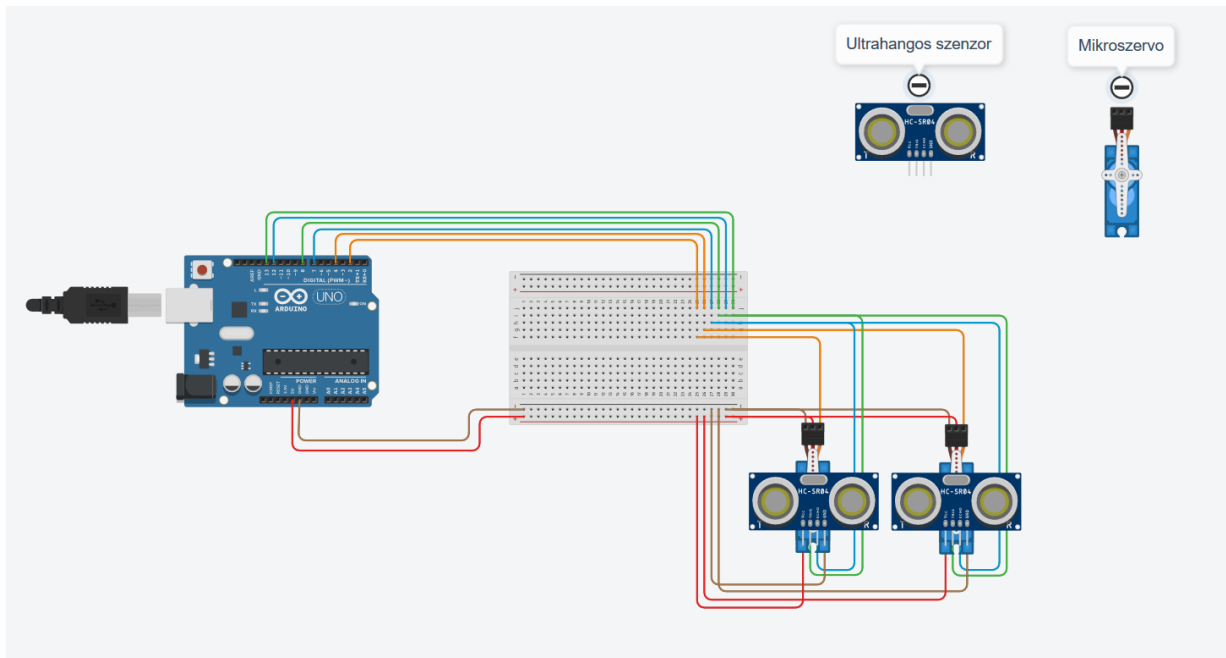


# Feladatok

- A hallgató feladata a már előző témakörben érintett tolatóradar mélyebb szintű megértése, tesztelése és továbbfejlesztése egy mikrokontrolleres környezetben. A feladat során a használt szenzorok összehangolása és a mért eredmények kiértékelése, illetve egy real-time grafikus megjelenítése a cél.
  - 1. Feladat: Ultrahangos szenzor pontosságának a mérése és tesztelése
  - 2. Feladat: Készítsen egy valószínűségi információt tartalmazó adatsort
  - 3. Feladat: Tesztelje a kidolgozott eljárásokat egy prototípus rendszeren
  - 4. Feladat: Több szenzor összehangolásának a lehetősége

# Összeállítás

- Mikrokontroller: Arduino uno
- 2 db Servo motor
- Ultrahangos szenzorok: 2 db HC-SR04
- Kiegészítők: jumper kábelek, breadboard



# Ultrahangos szenzor

## ■ HC-SR04

## ■ Adatlap alapján:

- Range: 3 cm – 400 cm
- Pontosság: 3 mm
- 40 kHz impulzus
- Mérési szög: 15°

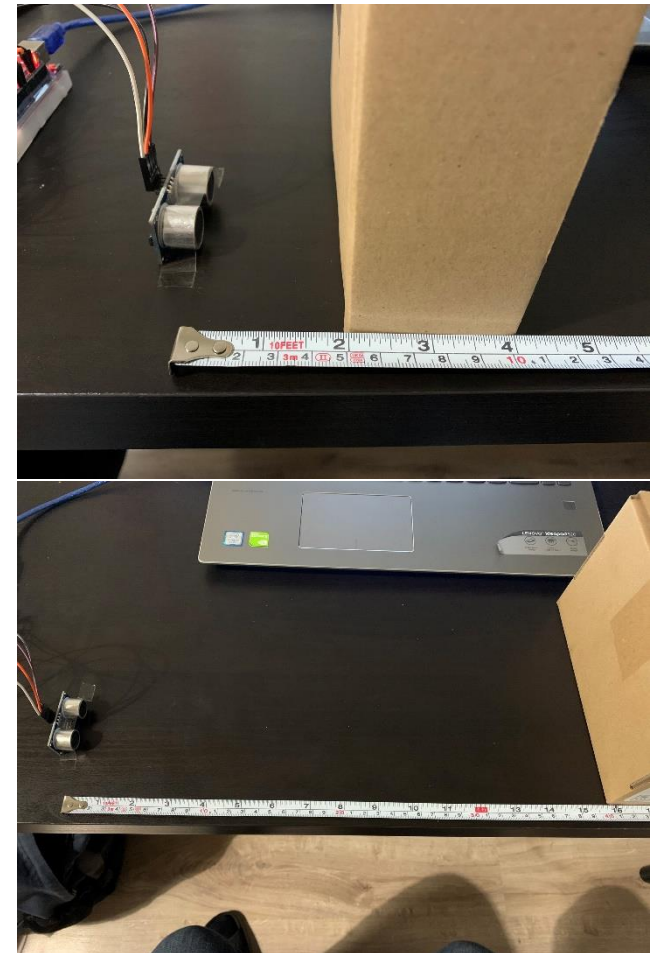
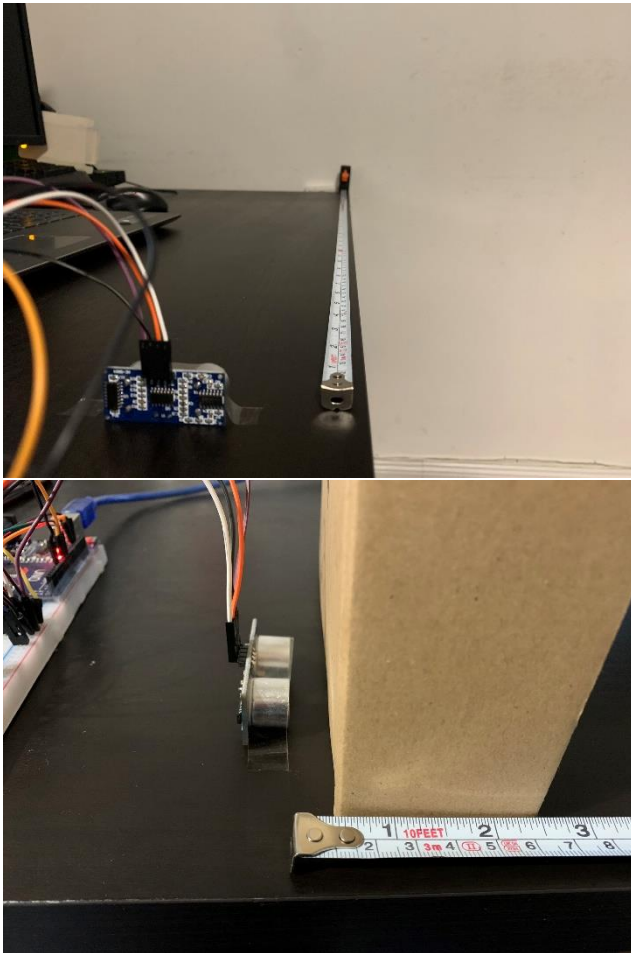


## ■ Kérdés: Igazak az adatok?

- Range: igaz, akár több is (~4,7 m)
- Pontosság: változó, 1 – 5 mm a szórása, de volt akár 1,5 cm is
- Mérési szög: kicsit kevesebb, 13° - 14°
- Ezen kívül további kérdések is vannak

# 1. Feladat

- Ultrahangos szenzor pontosságának a mérése és tesztelése



# 1. Feladat

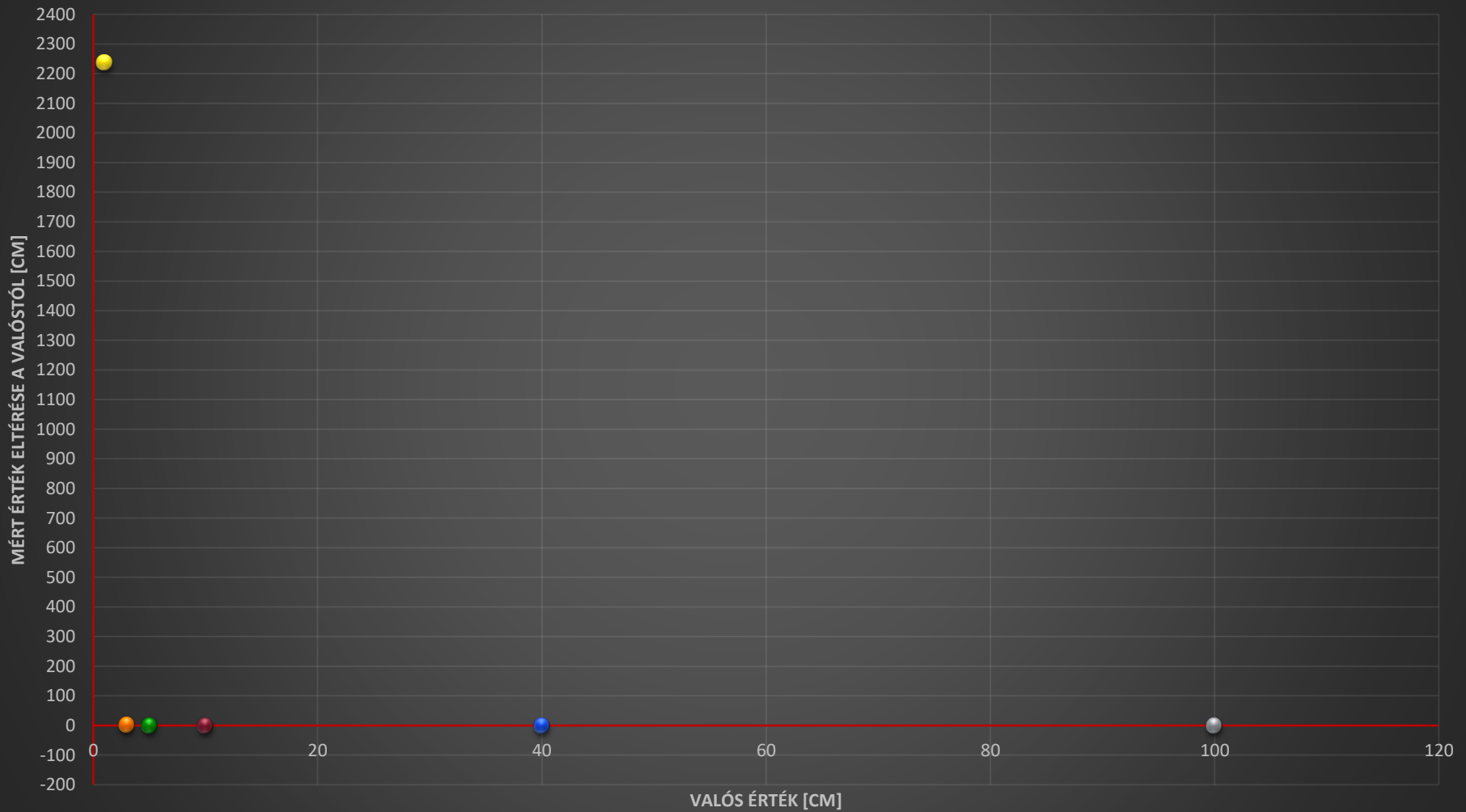
1. mérés:					
mérési adat\AT [ms]:	1	10	50	100	200
1.	2239,06	2239,11	2238,85	2238,90	2239,66
2.	2239,06	2238,90	2238,82	2238,99	2239,50
3.	2238,99	2239,14	2238,83	2238,97	2239,48
4.	2239,00	2239,09	2238,75	2239,40	2239,06
5.	2239,14	2238,83	2238,82	2239,12	2239,02
6.	2238,83	2238,82	2238,85	2239,36	2238,95
7.	2239,02	2238,90	2238,66	2239,55	2239,07
8.	2238,94	2238,73	2238,87	2239,48	2238,88
9.	2239,00	2238,76	2238,85	2239,30	2238,82
10.	2239,18	2238,70	2238,68	2239,24	2238,88
átlag mérés [cm]:	2239,02	2238,90	2238,80	2239,23	2239,13
valós [cm]:	1				
Eltérés [cm]:	2238,02	2237,90	2237,80	2238,23	2238,13

2. mérés:					
mérési adat\AT [ms]:	1	10	50	100	200
1.	5,31	4,80	4,90	5,21	4,80
2.	5,31	5,31	4,90	5,31	4,90
3.	6,65	4,90	4,90	4,80	4,90
4.	5,21	4,90	4,80	5,33	4,90
5.	5,21	4,90	4,90	4,80	4,90
6.	5,31	4,80	4,90	5,31	4,90
7.	5,31	4,90	4,80	4,92	4,90
8.	5,31	4,80	5,31	4,90	4,89
9.	5,21	4,90	4,90	4,90	4,80
10.	5,21	4,92	4,80	4,90	5,21
átlag mérés [cm]:	5,40	4,91	4,91	5,04	4,91
valós [cm]:	3				
Eltérés [cm]:	2,40	1,91	1,91	2,04	1,91

- Több mérési pont vizsgálata
- Mérési adatok táblázata (összesen 7 mérés)

# 1. Feladat

## Hiba mértéke



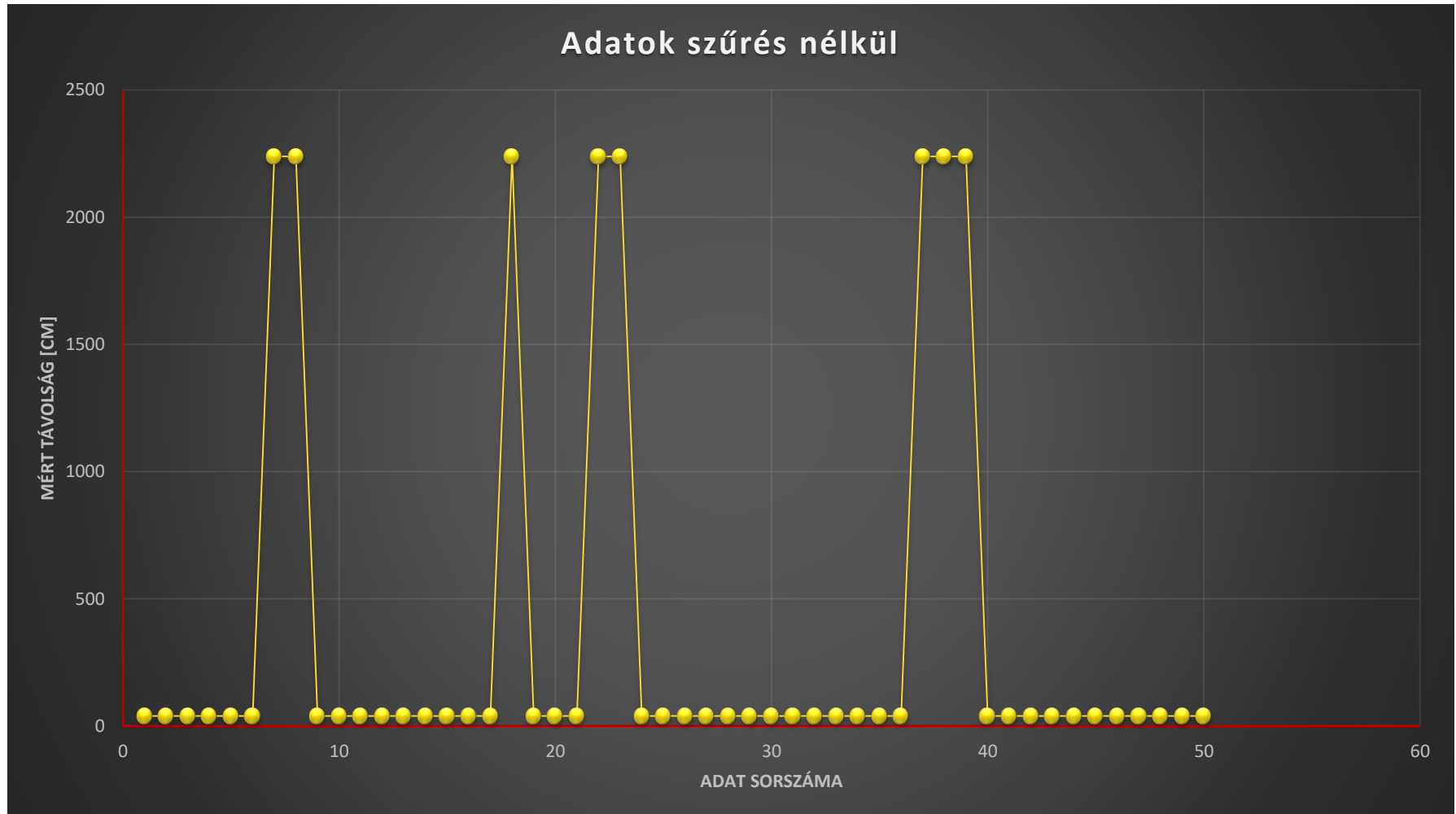


# 1. Feladat



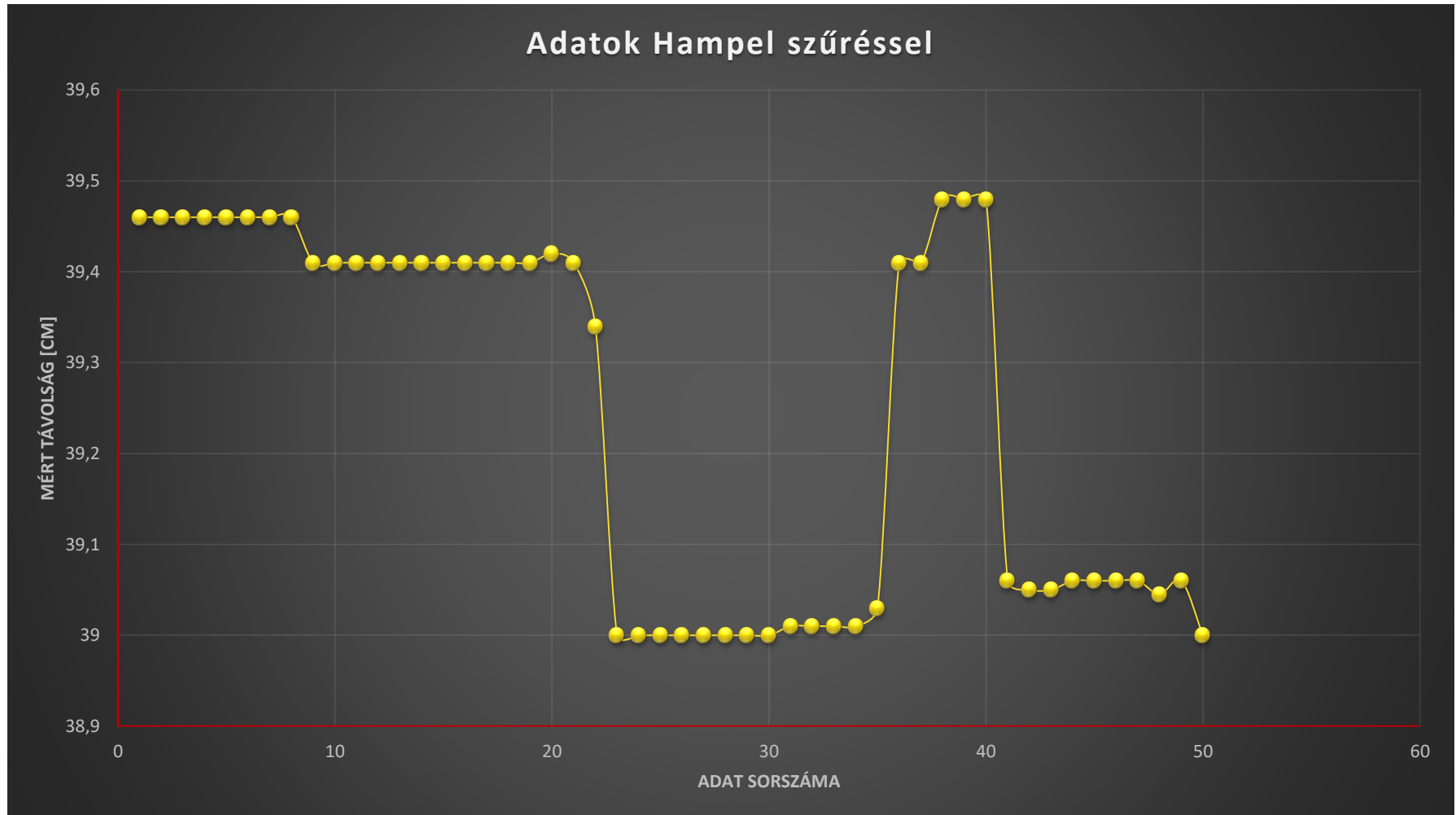
# 1. Feladat

- Hibás adatok szűrése → Hampel Filter



# 1. Feladat

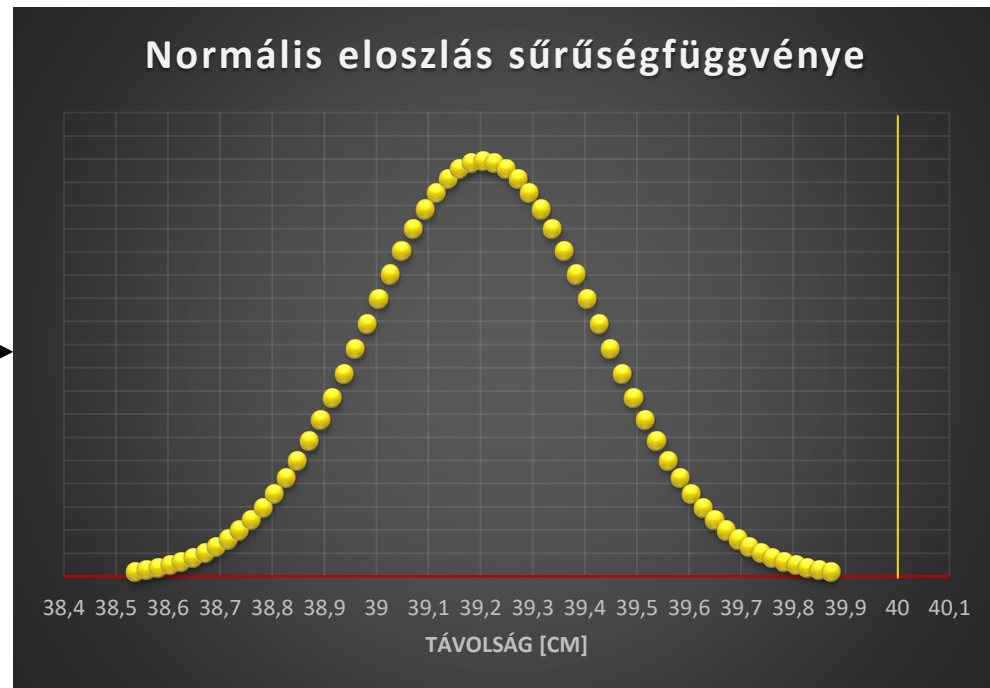
- Hibás adatok szűrése → Hampel Filter



## 2. feladat

- 40 cm-es mérés kicsit jobban elüt, mint a többi
- Készítsen egy valószínűségi információt tartalmazó adatsort

5. mérés:					
mérési adat\AT [ms]:	1	10	50	100	200
1.	39,36	39,41	39,00	39,01	39,48
2.	39,46	39,41	39,41	38,91	39,05
3.	39,46	39,41	39,01	39,01	39,05
4.	39,46	39,46	38,91	39,01	39,06
5.	39,48	39,41	39,00	39,03	39,03
6.	39,36	39,30	38,89	39,41	39,48
7.	39,48	39,42	39,00	39,30	39,06
8.	39,46	38,91	39,01	39,41	39,48
9.	39,46	39,41	39,00	39,01	38,94
10.	38,94	39,34	39,00	39,03	38,94
átlag mérés [cm]:	39,39	39,35	39,02	39,11	39,16
valós [cm]:	40				
Eltérés [cm]:	-0,61	-0,65	-0,98	-0,89	-0,84

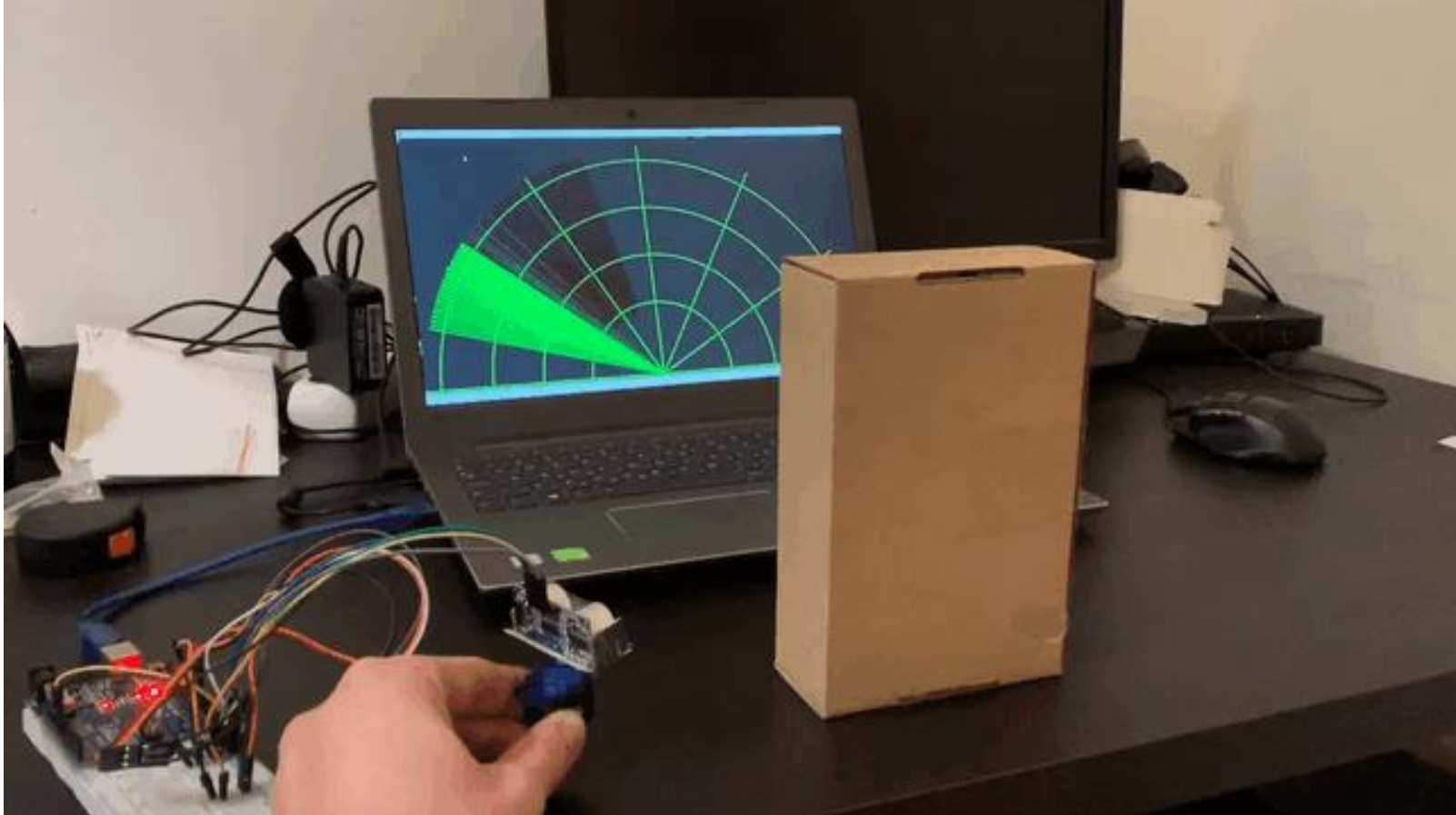


# 3. Feladat

Tesztelje a kidolgozott eljárásokat egy prototípus rendszeren

- Eddig is igaz volt, hogy meg kellett őket építeni + programozni
- Az első 2 feladathoz elég volt csak az Arduinot
- 3.-4. feladathoz szükség volt a Processing 3-ra
- Java alapú programozás → „könnyű” benne grafikus dolgokat lekódolni és megjeleníteni
- Processing 3 jól ötvözi a soros porton érkező adatok fogadását/feldolgozását és ezek adatokhoz grafikát készíteni

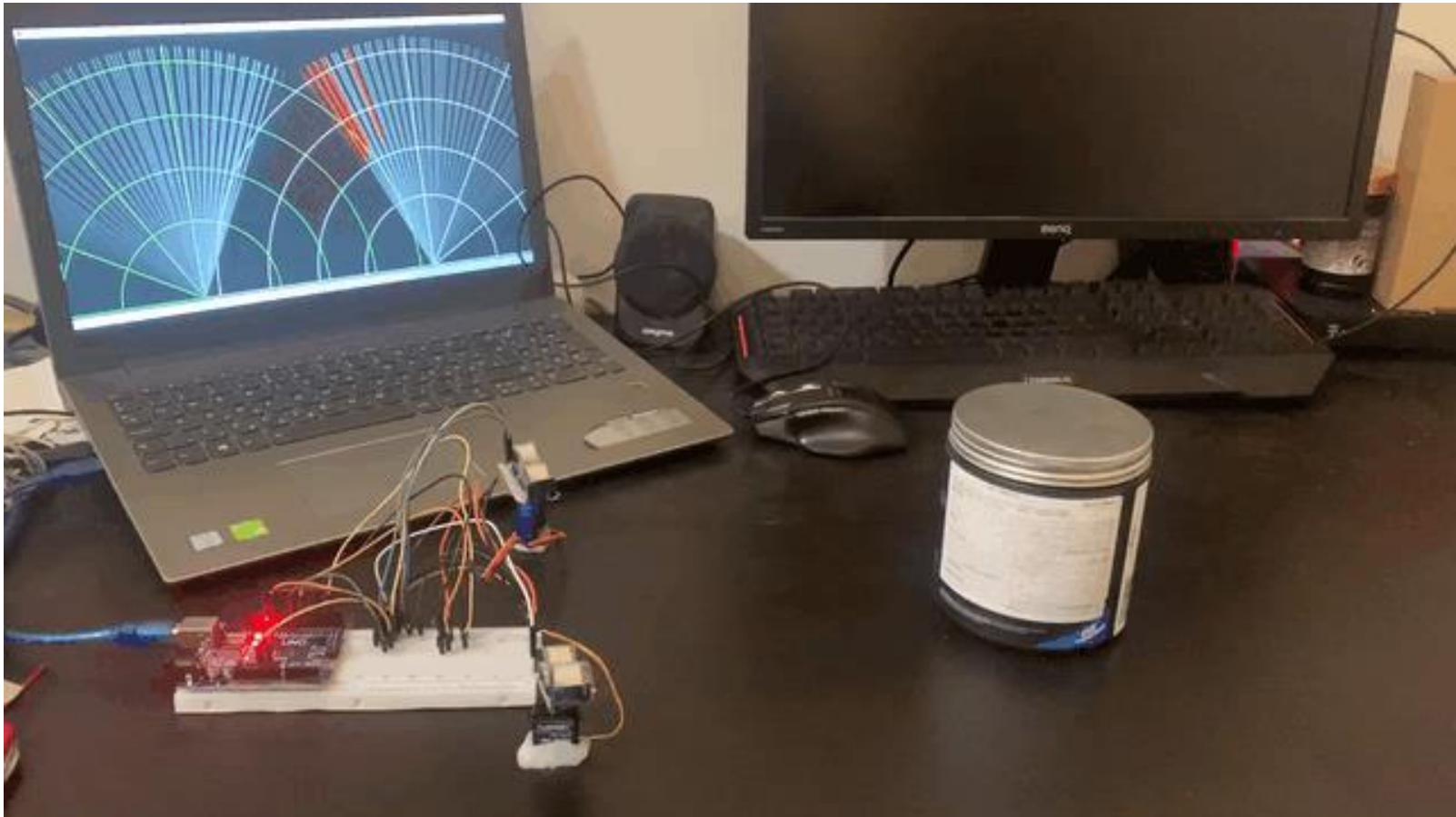
# 3. Feladat



# 4. Feladat

- Működik ez több szenzorral is?
- Kicsit nehezkesebb a feldolgozás
- Lassabb lett a rendszer
- Viszont:
  - Egy szenzor csak relatíve sík felületet „lát”
  - Több szenzorral akár kedvezőtlenebb felületekről is jobb képet kaphatunk → javítja a detektálást.

# 4. Feladat





# Összefoglalás

- Az előző féléves téma folytatása azért több kérdést is felvet:
  - Hogyan integrálható össze a másik projekttel?
  - Hogyan hozhatunk létre egy tápegységtől és pc/laptoptól független rendszert mindezek működtetésére?
  - Hogyan adhatjuk tudtára a sofőrnek az adatokat?
    - Grafikusan?
    - Hangjelzéssel?
    - Stb.

Köszönöm a figyelmet!