

A fidget spinner forgómozgásának vizsgálata a műszaki informatika eszközeivel

Somogyi Anikó¹, Kelemen András², Mellár János³, Mingesz Róbert⁴

{¹somogyia, ³mellar, ⁴mingesz}@inf.u-szeged.hu

^{1,3,4}SZTE TTIK Műszaki Informatika Tanszék

²kelemen@jgypk.szte.hu

²SZTE JGYPK Informatika Alkalmazásai Tanszék

^{1,2}Szegedi Radnóti Miklós Kísérleti Gimnázium

Absztrakt. A 21. században az informatikai, műszaki tudományok és a természettudományok közötti határvonalak elmosódni látszanak. Az informatika oktatásában szerepet kap a mikrokontrollerek programozása, ugyanakkor a modern fizikatanítás integrálja a számítógépes szimulációkban, számítógép-vezérelt mérési kísérletekben rejlő lehetőségeket. Utóbbi demonstrálására egy kinematikai jelenség dinamikai okainak precíz feltárását célzó, komplex műszaki informatikai megoldásokat alkalmazó kísérletet mutatunk be. A közegellenállási erő szerepének vizsgálatához egy vákuumkamrát fejlesztettünk, amelyben elhelyezett szenzorok jeleit a multifunkciós NI myDAQ és a népszerű Arduino Uno segítségével detektáltuk. Különböző informatikai eljárásokat prezentálunk az adatgyűjtés vezérlésére, a fotokapu négyszögjelének numerikus analízisére, továbbá a származtatott időfüggő jelek grafikus megjelenítésére és kiértékelésére. A kétféle eszköz sajátosságai miatt az eszközök vezérlése és a mért adatok számítógéppel történő fogadása is eltérő programozási technikákat követelnek meg a felhasználatól (LabVIEW, C/C++). A vizsgálat eredményei, módszerei alkalmazhatók a középiskolai fizika- és informatikaoktatásban egyaránt.

Kulcsszavak: interdiszciplináris oktatás, műszaki informatika, Arduino, LabVIEW, Excel Data Streamer

1. Bevezetés

A 21. században a természettudományok és az informatikai, műszaki tudományok közötti határvonalak elmosódni látszanak.

Az informatika oktatásában szerepet kap a mikrokontrollerek programozása. Jó példa erre Nagy-Britanniában a Pythonban, illetve JavaScriptben programozható micro:bit elterjedése [1], [2], vagy az olcsón elérhető, C/C++ alapú Arduino, amelyről szóló publikációk száma az elmúlt tíz évben drasztikusan megnövekedett [3], [4].

Ugyanakkor a fizikatanítás modernizációját célzó szakemberek integrálják a számítógépes szimulációkban, számítógép-vezérelt mérési kísérletekben rejlő lehetőségeket. Egyre nagyobb népszerűségnek örvend az iskolai mérésekre fejlesztett, LabVIEW-ban programozható NI myDAQ többfunkciós mérőeszköz is.

A hazai informatikai és a középiskolákban a műszaki képzés alapját is jelentő természettudományos oktatás megújításáért tevékenykednek az MTA-SZTE Műszaki Informatika Szakmódszertani Kutatócsoport tagjai is, köztük a jelen tanulmány szerzői is [5].

Általánosan elmondható, hogy a kutatók a szoftver-vezérelt mérésekben lehetőséget látnak a természet- és műszaki tudományok népszerűségének növelésére, a munkaerőpiacon jelentkező műszaki tudományokhoz kapcsolódó *hiányszakmák utánpótlás-nevelésére* [6].

2. Informatika a természettudományos megismerés szolgálatában

A természettudományos megismerés alapvetően mindig az emberi tapasztalatszerzésből, *információgyűjtésből* (megismerés, kísérlet, mérés) indul ki. Ezt követi a kiértékelés, összefüggések keresése, majd ezek rendszerbe foglalása. Ezek alapján fogalmazhatunk meg általános törvényszerűségeket, készíthetünk matematikai modellt (pl. aránypárok, egyenletek, képletek felállítása). A modell alapján lehetőség van új állítások levezetésére, azonban ezeket mindig össze kell vetni a tapasztalattal. A mérőeszközök fejlődése az emberiség történelme során mindig a természettudományok fejlődését vontta magával [7]. Természetesen a számítástechnika térhódítása újabb és újabb ajtókat nyit meg a mai napig, hiszen segítségével a részletezett megismerési folyamatban mind a mérés, a kiértékelés, a rendszerezés, a modellalkotás precízebbé, gyorsabbá és hatékonyabbá tehető.

Fontosnak tartjuk, hogy az informatikának ezt a modern szerepét már korán mutassuk meg diákjainknak, adjuk kezükbe azokat az eszközöket, amelyekkel később maguk is képesek lesznek önálló, kutatás alapú tanulásra. Az informatika és a természettudományok ötvözésének *motivációs szerepe* is lehet: az informatikába beemelt kísérleteknek, illetve az egyébként kevésbé népszerű természettudományok informatikai tartalmakkal történő színesítésének egyaránt motivációs, figyelemfelkeltő hatása van. Általános elvek, módszerek és eljárások helyett egy konkrét fizikai kísérlet számítógéppel segített vizsgálatát szeretnénk bemutatni, mellyel azt kívánjuk szemléltetni, milyen *szorosan kapcsolható össze a fizika és az informatika tanítása* a műszaki informatika eszközeivel [8].

Problémafelvetés:

A fizika tanításával kapcsolatos nemzetközi szakirodalomban egyre gyakrabban jelenik meg a fidget spinner. Ha megpörgetjük a mindenki által jól ismert játékot, nagy sebességű forgásának szögsebessége egy idő után csökken a fékező hatások miatt [9], [10]. Több internetes fórumon, videómegosztó portálon is fogalmaznak meg sejtéseket (illetve foglalnak állást), hogy vajon a közegellenállás vagy a súrlódás hatása dominál [11]. Ha ezt a könnyen érthető fizikai jelenséget a modern digitális méréses technika precíz nagyítóján keresztül vizsgáljuk meg, hamarosan kiderül, mégsem olyan egyszerű a feltett kérdésre választ adni, a számítógép-vezérelt kísérletünk azonban segít feltárni az ok-okozati összefüggéseket a jelenség hátterében.

Célkitűzés:

A vizsgálat bemutatása során igyekszünk reflektorfénybe állítani az alkalmazott informatikai módszereket, ötleteket, amelyek reményeink szerint az oktatás legkülönbözőbb szintjein alkalmazhatónak bizonyulnak majd.

3. A mérésben alkalmazott „hardver”

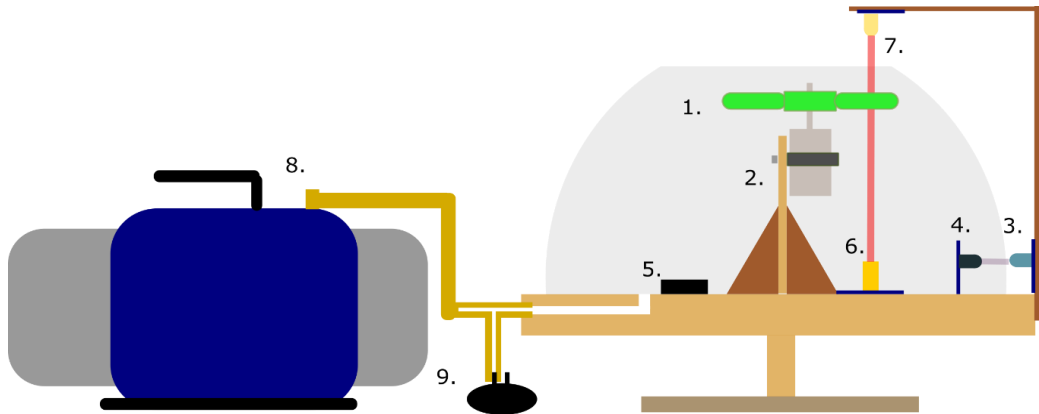
Ha a forgómozgás kinematikai jellemzőit (pl. szögelfordulás, szögsebesség) vizsgáljuk, azaz a „*Hogyan mozog?*” kérdésre keressük a választ, elegendő egy egyszerű fotokaput csatlakoztatni egy hangkártyához [12]. Ha azonban a „*Miért úgy mozog, ahogy?*” kérdésre adott feleletet célozzuk meg, azaz a dinamikai okok feltárását, akkor a mérés során el kell különíteni a pörgettyűt érő kétféle fékező hatást. Természetesen adódott az ötlet, hogy a légellenállást gyakorlatilag meg lehet szüntetni egy vákuumban végzett kontrollkísérlettel.

3.1. A mérési elrendezés

Olyan kísérleti elrendezést kellett megvalósítani, amelyben normál légköri nyomáson is vizsgálható a fidget spinner mozgása, ugyanakkor megoldható, hogy kiszivattyúzzuk a körülötte lévő levegőt, és a megváltozott körülmény mellett megismételjük a kísérletünket.

Egy salátástálból, fadeszkából készült vákuumkamrába helyeztük el a fidget spinnert. Nyilván vákuumban a játék kézzel történő bepörgetése nem lehetséges, ezért egy infravörös adó-vevővel

távolról vezérelhető egyenáramú hobbi motort (1-3 V, max. $6600 \pm 10\%$ RPM) motort használtunk erre a célra. A mozgásérzékelést végző *fotokaput* úgy készítettük el, hogy a megvilágító lézert diódát a fidget spinner alá, a fényt, valamint annak megszakítását érzékelő fototranzisztor a búra fölé helyeztük el. A búra alól a levegőt egy nagyteljesítményű vákuumszivattyúval szívattuk ki, a bent uralkodó légnyomás értékét egy *nyomásmérő* segítségével figyeltük a mérés során. Ennek a saját tervezésű vákuumkamrának a sematikus vázlatát szemléltetjük az 1. ábrán.



1. ábra: mérési elrendezés felépítése. Jelmagyarázat: 1. Fidget spinner, 2. DC motor állványra rögzítve, 3. infravörös adó és az áramköre, 4. infravörös vevő és az áramköre, 5. elemek 6. lézert dióda modul 7. fototranzisztor és az áramköre (6-7. együttesen fotokapu), 8. vákuumszivattyú, 9. nyomásmérő szenzor¹

3.2. A mérési feladatok elektronikai megvalósítása

Mérési elrendezésünkhöz egyedi elektronikát is terveztünk. A megoldandó feladatok a

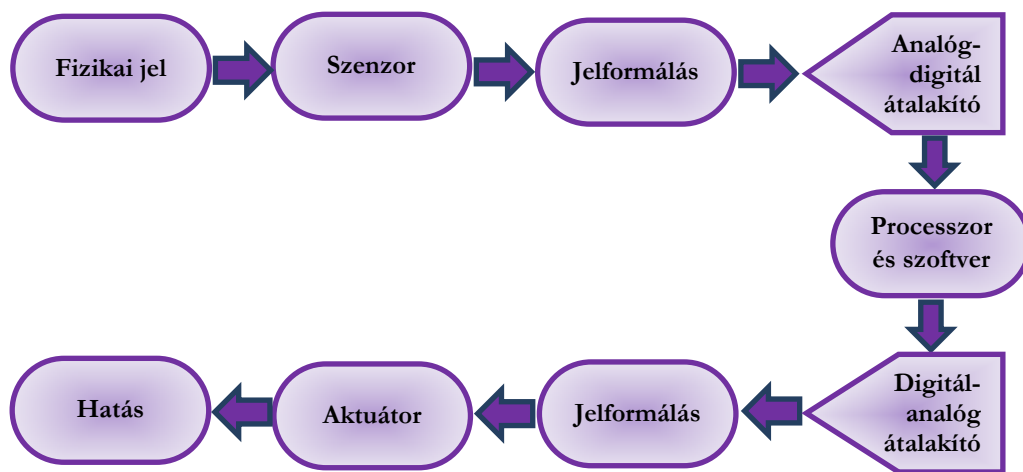
- A fidget spinneret meghajtó motor tápellátása
- A motor bekapcsolását végző tranzisztor és a lézert dióda tápellátása
- A tranzisztor nyitó IR adó-vevő vezérlése
- A fotokapu jelének detektálása
- A nyomás mérése

voltak.

3.2.1. Szenzorok és átalakítók

A *digitális jelfeldolgozás* folyamatában elengedhetetlenek a *szenzorok*, amelyek a fizikai jeleket alakítják elektromos jelekké, illetve a mérőcsatló eszközök, amelyek az analóg elektromos jeleket átalakítják – a számítógép által értelmezhető – digitális jelekké. Szerepüket szemléltetjük a 2. ábrán [13].

¹ Az ábra az Inkscape programban készült.



2. ábra: A digitális jelfeldolgozás folyamata

A mérésünk során a vákuumszivattyúhoz csatlakoztatva egy MPX 4250DP típusú nyomásmérő szenzort használtunk. Az érzékelő egy ún. differenciális, kalibrált szenzor, mely a külső légnyomáshoz hasonlítva követte a búra alatt uralkodó nyomást [14].

A mozgás érzékelésére a feljebb említett saját összeállítású fotokaput használtuk, melybe egy KY-008 típusú lézermódult és egy fototranzisztort építettünk be [15], [16].

A fent felsorolt feladatokat ellátó szenzorokat és az áramköröket kétféle mérőeszközhöz (myDAQ, Arduino UNO) csatlakoztattuk, amelyek többek között az *analóg-digitál átalakító* szerepét töltik be a mérési folyamatunkban.

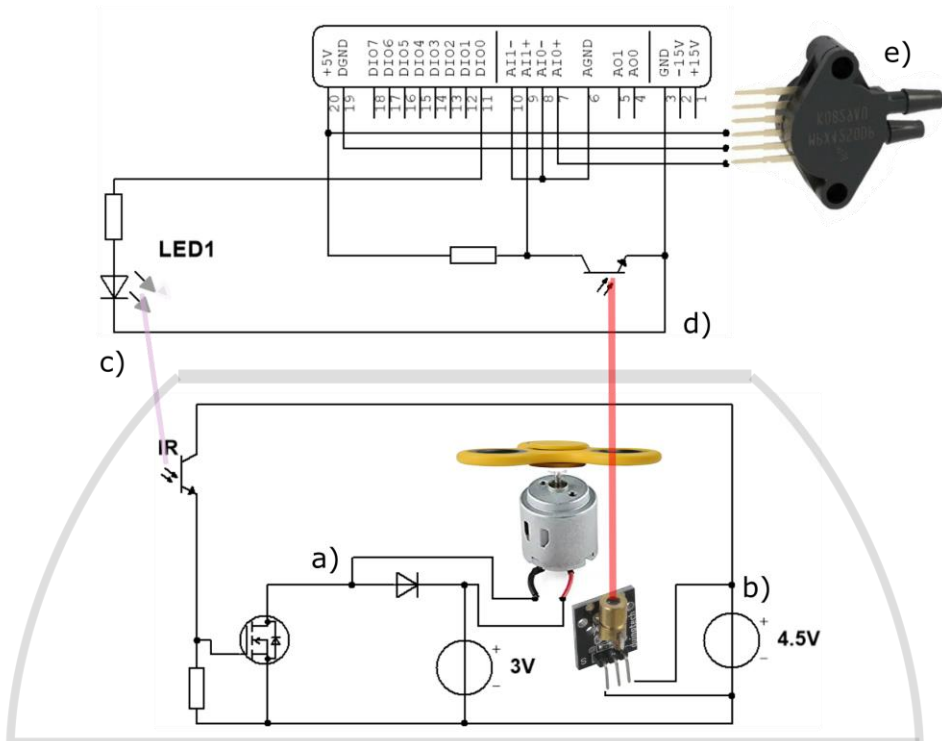
3.2.2. Jel mérése myDAQ-kal

A mérést elvégeztük a National Instruments *myDAQ* nevű *multifunkcionális eszközzel*, amelyet direkt iskolai laboratóriumi mérésekre fejlesztettek ki. Az eszközön 2 db analóg input (AI) és 2 db analóg output (AO) csatlakozó, 8 db digitális input/output (DIO) csatlakozó, 1 db audio input és 1 db audio output csatlakozó, továbbá digitális multiméter (DMM) csatlakozók találhatóak. A műszer tápellátása, a műszert vezérlő és a mért adatokat a számítógépbe juttató szoftveres kommunikáció USB-porton keresztül történik [17].

Mérésünk első változatában 100 000 minta/s-os mintavételi frekvenciával mértük a fotokapu jelet az AI1 analóg csatornán. A mérés eredményeképp egy növekvő szélességű *analóg négyzetjel*et kapunk, hiszen ahogy a játék valamelyik karja áthalad a fotokapun a műszer által mért feszültség $\sim 5V$ -ról $\sim 0 V$ értékre változik. Ha a motorral felpörgetett fidget spinner szögsebessége $110 \frac{\text{rad}}{\text{s}}$, 19 ms-onként következik be a feszültségugrás, a lassulás miatt egyre hosszabb időtartamot mérhetünk.

A nyomás mérését a myDAQ AI0 analóg csatornáján mértük, 1000 minta/s-os mintavételi frekvenciával. Az analóg jel mérésének az az előnye, hogy látjuk a jelalakot, folytonosan mintavételezünk, a hardver végzi a megbízható *ütemezést*.

A 3. ábrán kapcsolási rajzokon mutatjuk be a fent felsorolt mérési feladatok (a-e) megvalósítására megépített, a myDAQ-hoz csatlakoztatott áramköröket.



3. ábra: Az NI myDAQ mérőcsatolóhoz kapcsolt elektronika kapcsolási rajza. Az ábra alsó felében a vákuumkamrán belül, a felső részében a kívül elhelyezett áramköröket ábrázoltuk²

3.2.3. Jel mérése Arduinóval

A National Instruments myDAQ eszközt kifejezetten tanulói kísérletezésre fejlesztették, több szempontból igen kényelmes a felhasználó számára, ugyanakkor beszerzése viszonylag költséges. Bár értesüléseink szerint a myDAQ egyre több magyarországi gimnáziumban elérhető különböző pályázatok révén, mérésünket átalakítottuk úgy, hogy *Arduinóval* is elvégezhető legyen. A népszerű, elérhető áron és könnyen beszerezhető eszközzel elvileg teljesen hasonló módon lehetne mérni, mint a myDAQ-kal. Azonban a mintavételi frekvenciája jóval elmarad a már ismertetett eszköztől, kevesebb mint 10 000 minta/s a maximális érték. Éppen ezért az Arduinóval végzett mérésünkben a fotokapu jelét *digitális csatornára* kapcsoltuk, és azokat az eseményeket figyeltük, amikor a jel logikai alacsony értékről logikai magas értékre ugrott, azaz, amikor a fidget spinner karja metszi a fénytut. A 4. ábrán szemléltettjük, hogyan lehet az Arduinóhoz csatlakoztatott áramköröket összeállítani

² Az ábra az NI Multisim próbaverziójával, valamint az Inkscape programmal készült. A felhasznált képek forrásai (Utolsó megtekintés: 2019. október 30).

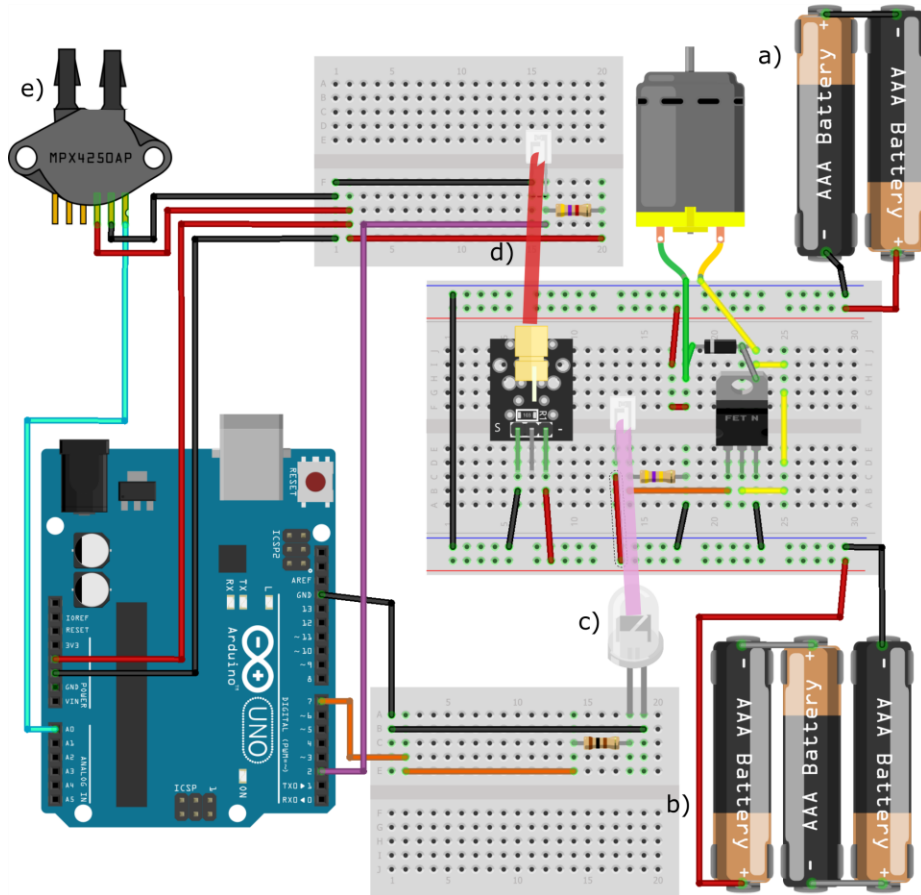
https://images-na.ssl-images-amazon.com/images/I/71aDevQIaZL._SL1500_.jpg

https://images-na.ssl-images-amazon.com/images/I/41V-87Kw50L._SX342_.jpg

https://bestcustomflashdrives.com/wp-content/uploads/2017/08/Custom_Fidget_Spinners_sideview.jpg

<https://i.ebayimg.com/images/g/FyoAAOSw60NbQwca/s-l1600.jpg>

dugaszolós próbapaneleken (breadboardokon), pl. tesztelés céljából. (A mérési feladatokat a 3. ábrával megegyezően jelöltük.)



4. ábra: Az Arduino UNO-hoz csatlakoztatott áramkörök megépítése dugaszolós próbapaneleken (breadboardokon)³

4. A mérésben alkalmazott „szoftver”

A fent ismertetett két eszköz programozása meglehetősen különböző. A myDAQ-ot alapvetően a *LabVIEW grafikus programozási nyelven*, míg az Arduino-t *C/C++-alapú környezetben*, esetleg Pythonban szokás programozni. Az általunk használt szoftveres részek a két eszközre azért is különböznek gyökeresen, hiszen az előzőekben kifejtett érvek alapján a myDAQ-kal analóg, az Arduinoval digitális jelet mértünk.

³ Az ábra a Fritzing nevű ingyenesen letölthető programban készült.

4.1. A myDAQ-kal történő mérés programozása LabVIEW-ban

4.1.1. A LabVIEW grafikus programozási környezet

A LabVIEW-t főként ipari számítógépes mérések szabályozására és vezérlésére, a mérési eredmények feldolgozására fejlesztették ki. A LabVIEW-ban megírt programokat *virtuális műszereknek* (.vi)-oknak szokás nevezni, melyeknek két fő része a műszerfalra emlékeztető *Front Panel* és a programkódot tartalmazó *Block Panel*. A Front Panelen a felhasználó által használható kontrolmezőket (pl. gombok, legördülő mezők) és adatok megjelenítésére szolgáló indikátormezőket (pl. számadat- vagy stringmezőket, grafikonokat) helyezünk el. A Block Panelen pedig beépített függvényeket reprezentáló grafikus objektumokat helyezünk el, kapcsolunk össze adatvezetékekkel, és rendezünk különböző struktúrákba. A nyelv fontos elve az ún. *Data Flow*: egy parancs a programban akkor hajtódik végre, amikor az összes bemenetére megkapta a szükséges jelet. Így a parancsokat vezetékekkel „láncra fűzve” határozhatjuk meg a végrehajtási sorrendjüket [18].

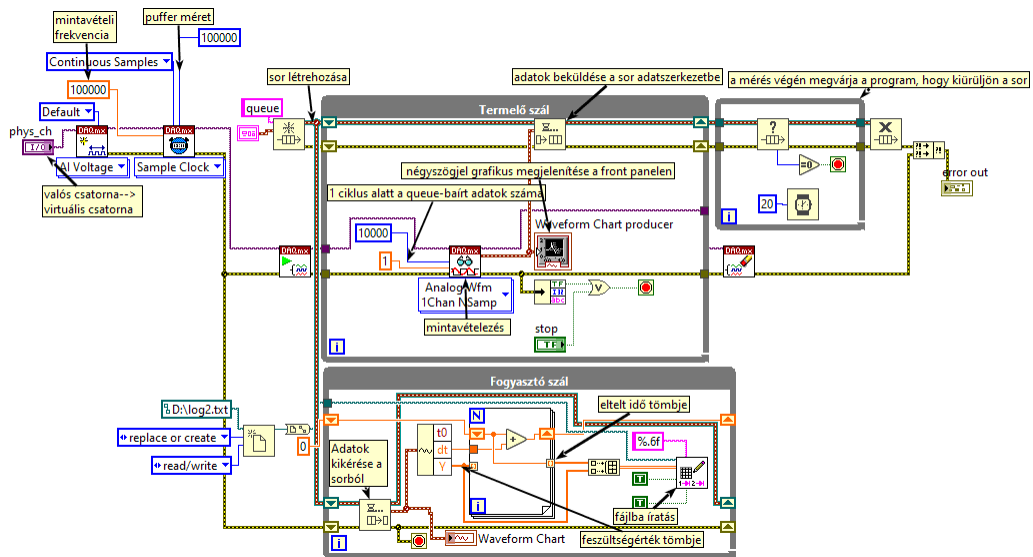
A National Instruments mérőeszközeinek programozásához általában az *NI-DAQmx drivert* használjuk, amellyel a konfigurációs, adatmérési és -generálási teendőket az igényeinknek megfelelően beprogramozni. A driver függvénytárában szerepel többek között virtuális csatorna létrehozása, csatorna olvasása, csatorna írása, feladat indítása, leállítás, időzítése, stb. [19].

4.1.2. A termelő-fogyasztó probléma megoldása LabVIEW-ban

A klasszikusnak mondható *termelő-fogyasztó szinkronizációs probléma* lényege, hogy a program két szálát, az objektumokat létrehozó és a *pufferba* helyező termelőt, valamint az adatokat a pufferből kiemelők és felhasználó fogyasztót egymástól függetleníteni kell. Ha a pufferben nincsen adat, vagy megtelik adattal, akkor valamelyik szálnak várakoznia kell.

A myDAQ-kal történő analóg mérésünk során a termelő szál a mintavételezést végzi, meghatározott időközönként (100 000 minta/s esetén 1 μ s-onként), és ezeket az adatokat helyezi a 100 000 minta tárolására szolgáló pufferba. *First in first out* (FIFO) elvet valósítunk meg, tehát amely adat először jön, az először távozik, erre a sor (Queue) adatszerkezetet használjuk. A termelői szál futtatása előtt létrehozunk egy ilyen sort, a szál maga pedig folyamatosan tölti adatokkal, ciklusonként 10 000 db feszültségérték kerül a sorba.

Virtuális műszerünk fogyasztó szálának a feladata, hogy kikérje a mért feszültségadatokat a pufferből, a mérések közti időközt, majd tömbökbe rendezze az adatsorokat, végül kiírja egy adatfájlba. Ezzel a megoldással az adatok tárolása és feldolgozása nem szakítja meg a *valós időben*, egyenlő időközönként történő mérést.



5. ábra: A myDAQ mérőeszköz virtuális műszere a fotokapu jelének mérésére

4.1.3. Utólagos vagy párhuzamos kiértékelés LabVIEW-ban

Az előbbieken ismertetett virtuális műszer segítségével végrehajtott mérés végeztével egy .txt fájlban található a mért adataink tabulátorral elválasztva. Ezután az adatok egy egyszerű táblázatkezelőben, vagy ugyanitt, LabVIEW-ban feldolgozhatók.

Az első feladat, hogy a négyzetjelben *színtmérés-detektálást* végezzünk, azaz megkeressük azokat az időpontokat, ahol a jelben ugrás van, és ezeket egy tömbben összegyűjtjük. Ezek között az időtartamok között a fidget spinner 120°-ot haladt, növekvő szögelfordulásából egy for ciklussal szintén készítünk egy tömböt. A klaszterbe rendezett tömpárt, a szögsebességet az idő függvényében egy XY grafikonon ábrázolhatjuk. Ezzel a módszerrel arra is van lehetőség, hogy akár több mérés eredményét egy közös grafikonon ábrázoljuk.

A fizikai változásokról többet árul el, ha a változás sebeségét figyeljük. Jelölje α_i és t_i az i -edik elemet rendre a szögelfordulás- és időtömbben. Ekkor a t_i pillanatban az ω_i pillanatnyi szögsebességet közelíthetjük:

$$\omega_i \approx \frac{\alpha_{i+1} - \alpha_i}{t_{i+1} - t_i}$$

Itt valójában a differenciahányados értékét számoljuk ki, ezt előre mutató *numerikus deriválnak* is nevezik. A tömbök elemein elvégezve a műveleteket egy shift regiszter segítségével a szögsebesség-értékekből is egy tömböt képezhetünk, és azt is ábrázolhatjuk az idő függvényében, mint ahogy tettük a 10. ábrán.

A LabVIEW-ban történő kiértékelésnek nagy előnye, hogy a grafikus kódunkat át tudtuk emelni a fogyasztó szálba, így közel valós időben tudjuk grafikusán ábrázolni a mért adatainkat. Az ilyen „mozi” típusú megjelenítés kiváló tanórai demonstráció megvalósítására.

4.2. Az Arduinoval történő adatgyűjtés megvalósítása

Arduino használata esetén már két szoftverről is beszélnünk kell. Egyrészt van az Arduinón futó *beágyazott szoftver*, mely C/C++ nyelven van megírva. Az adatok továbbítása a PC-re az USB soros porton keresztül történik egy egyszerű protokoll segítségével: az Arduino folyamatosan detektálja a

fidget spinner elhaladását a fototranzisztor előtt, majd elküldi ezen *események időbélyegeit* egész számként. Minden üzenetet egy vesszővel majd egy sortöréssel zár le.

A számítógép felől több lehetőségünk is van az adatok fogadására, a fejezet második részében három egyszerű opciót mutatunk be.

4.2.1. Az Arduinón futó beágyazott szoftver megvalósítása

Míg a myDAQ esetén a mérés vezérlését a számítógépen futó LabVIEW szoftver végzi, addig az Arduino alkalmazása esetén nekünk kell megvalósítani az Arduinón futó beágyazott szoftvert. Ennek a szoftvernek a feladata a kívánt jelek mérése, majd a kapott adatok továbbítása a számítógépre a *soros porton* keresztül. A megvalósítás során olyan szoftvereszközöket kerestünk, melyek jól ismertek, és minden Arduino kártyával kompatibilisek (így nem használtuk a regiszterek közvetlen írását, olvasását). Arduino esetén nincs lehetőség az analóg jelek gyors mintavételezésére, mint a myDAQ esetén. Ugyanakkor a jel, amit mérni szeretnénk tulajdonképpen digitális: a fidget spinner vagy kitarja a fény útját, vagy nem. E miatt, egy digitális bemenettel közvetlenül is mérhető. Arra azonban figyelni kell, hogy a fototranzisztor kimenetén mérhető jelszintek az Arduino által érzékelt feszültségtartományba kerüljenek.

Ha pontosan szeretnénk időt mérni, akkor a legjobb módszer a *megszakítások* használata [20]. Az Arduino képes külső megszakítások fogadására, ahol beállíthatjuk, hogy felfutó élre (0-ról 1-re váltásnál) hívódjon meg a megszakítást kiszolgáló rutin. A rutinon belül az idő lekérdezésére a `micros()` függvényt használhatjuk. A függvény mikroszekundumban (μs) adja meg a mikrovezérlő elindulása eltelt időt egy 32 bites számként. Ezt fogjuk időbélyegzőként (timestamp) használni az események követésére. Az időbélyegző valódi felbontása függhet a konkrét platformtól, az aktuálisan felhasznált Arduino UNO esetén 4 μs . Ezen felül némi bizonytalanságot okozhat az, hogy pontosan mikor hajtódik végre a megszakítást kezelő rutin, összességében azonban 10 μs alatt marad a hiba, ezzel pedig jobb tud lenni a korábban alkalmazott myDAQ-nál.

A megszakítások használata miatt, bár csak egymagos rendszerről beszélhetünk, de egyfajta időosztásos rendszerként, több száunk is fut. Ezért az adatok biztonságos átadása a szálak között kritikus fontosságú. Egyrészt, az érintett változókat *volatile*-nak kell deklarálni, így azok biztosan nem lesznek valamely regiszterben gyorsítótárazva. Mivel az időbélyegző 32 bit-es, vagyis 4 bájtos, annak másolása nem atomi művelet. Ezért a főprogramban (*loop*), a másolás idejére letiltjuk a megszakításokat. Az adatok továbbítása a PC-re az USB soros porton keresztül történik, a *kommunikációs sebessége* 115200 bit/s.

A teljes kód:

```
// Digitális bemenet időmérésre
#define INT_PIN 2
// Detektált él: felfutó él
#define TYPE RISING

volatile unsigned long timeisr; // ideiglenes változó timestamp értékének biztonságos átadására
unsigned long timetmp; // ideiglenes változó a timestamp értékének ideiglenes tárolására
unsigned long timeprev = 0; // korábbi timestamp érték
unsigned int timeoutcounter = 0; // timeout számláló

/* Inicializálás, a mikrovezérlő indításakor fut le egyszer. */
void setup() {
  pinMode(INT_PIN, INPUT); // Időmérésre használt bemenet konfigurálása
  attachInterrupt(digitalPinToInterrupt(INT_PIN), isr, TYPE); // Megszakítás hozzárendelése digitális bemenethez
  Serial.begin(115200); // Soros porti kommunikáció sebességének beállítása
  while (!Serial) { ; } // Várakozás a csatlakozásra (USB eszközök esetén)
```

```

}

/* Fő ciklus, végtelen ciklusként ismétlődik. */
void loop() {
  noInterrupts(); // Megszakítások tiltása timestamp értékének biztonságos át-
adására
  timetmp = timeisr; // Timestamp átmásolása ideiglenes változóba
  interrupts(); // Megszakítások újra engedélyezése
  if (timeprev != timetmp) { // Ha a korábban eltárolt timestamp nem egyezik az
újjal: esemény van
    timeprev = timetmp; // Eltárolt timestamp frissítése
    Serial.print(timetmp); // Timestamp elküldése
    Serial.println(","); // Lezáró karakterek elküldése
  }

  delay(1); // Kis mértékű várakozás
}

/* Külső megszakítást kezelő rutin. */
void isr() {
  timeisr = micros(); // timestamp beolvasása
}

```

A fenti kód viszonylag egyszerű, ugyanakkor csak az időmérést végzi el. Hogyha szükséges a nyomás valós idejű mérése, akkor ki kell egészíteni az analóg jel mérésével és továbbításával. Ha akkor is szeretnénk nyomást mérni, amikor a fidget spinner nem forog, akkor az adatküldés protokolljának összetettebbnek kell lenni. Az Arduino alkalmas lehet a motor vezérlésére is, ekkor a számítógéptől kapott vezérlés függvényében kapcsol be vagy ki egy digitális kimenetet. E két feladat megvalósításának leírása már túl esik a cikk célján, de a kódok elérhetők a cikk mellékleteiben [21].

4.2.2. Az Arduino adatainak fogadása Excel Data Streamer segítségével

A Microsoft Office Garázs Programjának fejlesztői csapata készített egy ingyenes *bővítményt*, amelyet direkt oktatási célra, tanulói mérésekre fejlesztettek ki. A *Data Streamer* a Microsoft honlapjáról letölthetjük, illetve az Office 365-ben elegendő a COM-bővítmények között engedélyoznünk. Előbbi esetén angol nyelven, utóbbi esetén magyar nyelven, Adatgyűjtő néven jelenik meg az Excel fejlécében a bővítmény [22].

Ezután testre kell szabni az alkalmazást a mérésünknek megfelelően. A *Device Connection* fülnél ki kell választani a megfelelő kommunikációs portot, illetve hozzá csatlakoztatott eszközt, esetünkben az Arduinót. Ekkor automatikusan megjelennek a *Data in* (bejövő adatok megjelenítésére), a *Data Out* (kimenő paraméterek küldésére) és a *Settings* (beállítások változtatására szolgáló) Excel-munkalapok. Utóbbiban többek között beállíthatjuk a bejövő adatok közötti időintervallumot 10 ms-10 s-ig, a csatornák számát, és hogy a legfrissebb adat a megjelenített adatok oszlopában az utolsó vagy első cellában szerepeljen. A megjeleníthető adatok alapértelmezett száma 25, ez az érték legfeljebb 500-ra növelhető. Ezután az *Advanced* beállításoknál be kell állítani a kommunikációs sebességet (*Baud Rate-et*) a mérésben alkalmazottal megegyezően, amennyiben ez az érték eltér az alapértelmezett, 9600-as értéktől.

A mérést a *Start Data* funkcióval indíthatjuk, ekkor elindul az adatok megjelenítése a Data In munkalapon. Az adatok tárolására a *Start Recording* gomb megnyomásával nyílik lehetőség, a *Stop Recording* gomb megnyomása után a felugró ablakban elmenthetjük az adatainkat .csv formátumban. Ezek később beolvashatók a táblázatkezelőbe, és elvégezhető rajtuk a kiértékelés. Hasznos lehet pl. a mért adatokat a tanulóknak kiadni, és önálló munkában elvégezni a numerikus deriválást (a változási sebesség számítását).

Ha demonstrációs célból valós időben szeretnénk kiértékelni a diákjainkkal a mért adatokat, természetesen egy új munkalapon a szokásos cellahivatkozással megtehetjük. Problémát okoz azonban, hogy nem növelhető 500 fölé a megjeleníthető adatok száma, így ha ennél több hasznos adat jön be a csatornákon, akkor módot kell találni az adatok összegyűjtésére.

Erre egy kézenfekvő megoldást adtunk, egy *Excel VBA makró* segítségével. A néhány soros program figyeli azt a cellát, amelyben a legfrissebb adat szokott megjelenni, és amennyiben változik az értéke, akkor a legelső sor alá írja. Így az összegyűjtött időpontok időrendi sorrendben megjelennek a kiértékelésre használt extra munkalapunk megfelelő (A) oszlopában.

```
Private old As Long
Private Sub Worksheet_Change(ByVal Target As Range)
    Dim KeyCells As Range

    Set KeyCells = Range("B5")

    If Not Application.Intersect(KeyCells, Range(Target.Address)) _
        Is Nothing Then

        v = Worksheets("Data In").Range("B5").Value
        If old <> v Then
            LstRw = Worksheets("Munkal").Cells(Rows.Count, "A").End(xlUp).Row + 1
            Worksheets("Munkal").Cells(LstRw, "A") = v
            old = v
        End If
    End If
End Sub
```

A korábban bemutatott Arduino-mérőprogramunk mikroszekundumban olvassa ki az események időpontjait. Célszerű egyrészt másodpercben dolgozni, másrészt pedig érdemes a $t = 0$ -t az első adat megérkezésének időpontjához rögzíteni (az A oszlop első eleme) a

```
HA(A3=0;" ";(A3- $\$A\$2$ )*10-6)
```

képlettel.

Előre beállíthatjuk, hogy a kiértékelő munkalap egy másik oszlopában számítsa ki az Excel a szögelfordulás mértékét a

```
HA(A1=0;" ";(SOR(D1)-1)*2*PI()/3)
```

képlettel. Amennyiben az adott sorban megjelent már egy időpont, akkor az előző sorbeli értékhez hozzáadunk $2\pi/3$ -at (120° -ot.)

A numerikus deriválásnak, a szögsebesség számításának is szentelünk egy oszlopot. A

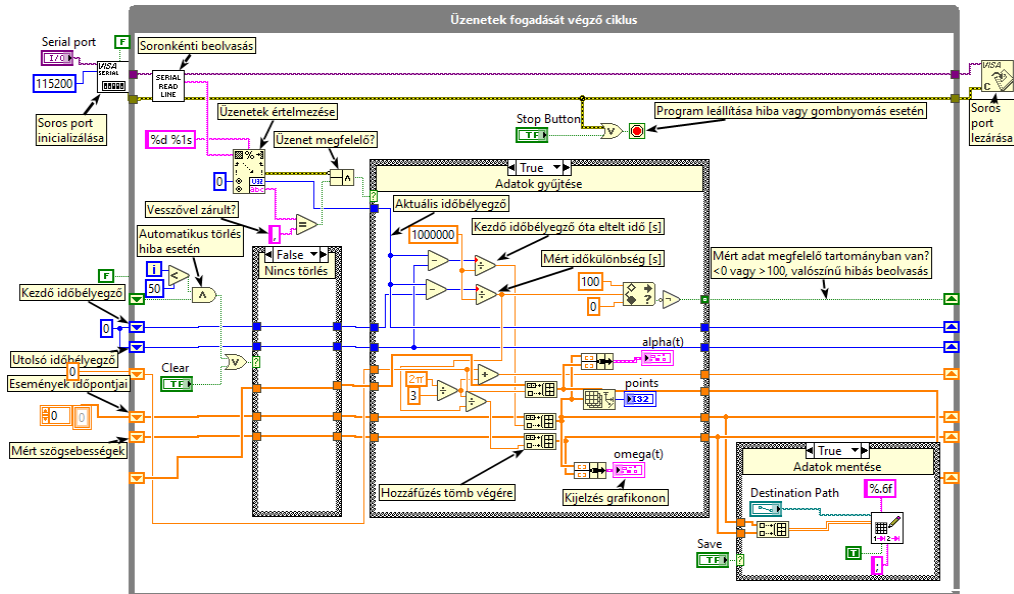
```
HA( $\$A1=0$ ;" ";2*PI()/3/($B2- $\$B1$ ))
```

képlettel számíthatjuk az első elemet. Nyilván itt az aktuálisan beérkező időpont mellett egy értelmezhetetlen mező jelenik majd meg a harmadik oszlopban, amely majd csak a következő adat beérkezésekor válik értelmezetté, ilyen módon a szögsebesség pillanatnyi értékét mindig lépéshatárnyban derítjük ki.

Ezek után a megfelelő oszlopok kijelölésével pontdiagramot készíthetünk, amelyen frissülni fognak a rendelkezésre álló adatok, így – gyakorlatilag – valós időben monitorozhatjuk a szögelfordulás és a szögsebesség időbeli változását.

4.2.3. Arduino adatainak fogadása LabVIEW program segítségével

Az Arduino által küldött, soros porton érkező adatokat *LabVIEW segítségével* is tudjuk fogadni, illetve feldolgozni. A LabVIEW program előnye, hogy azt egyszerűen ki tudjuk egészíteni további adatfeldolgozással. Az elkészített LabVIEW program viszonylag egyszerű, a 6. ábrán látható részletes magyarázatokkal.

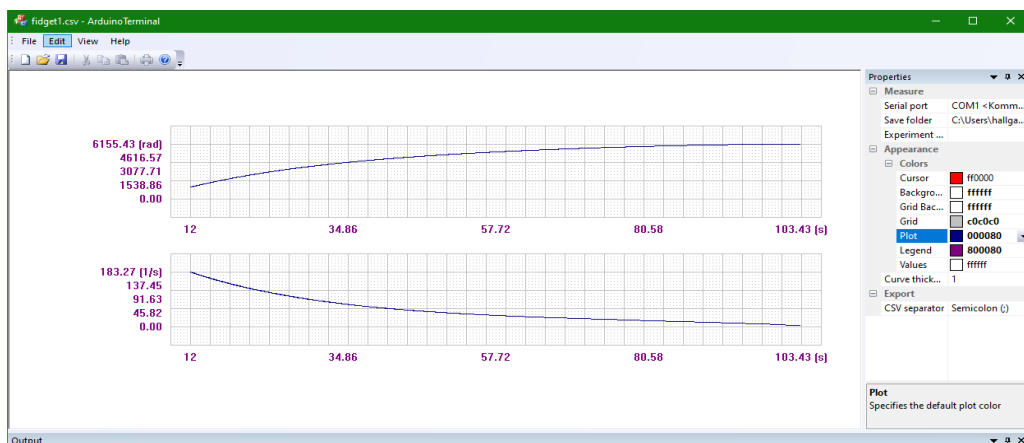


6. ábra: Az adatok beolvasása soros portról LabVIEW-ban

A program folyamatosan *monitorozza a soros portot*, és ha érkezik egy üzenet, akkor azt megpróbálja értelmezni. Sikeres üzenetformátum esetén eltárolja az adatokat egy tömbben (az események idejét, valamint az események között mért időközönbséget). Az eltárolt adatokat kijelzheti egy grafikonon, de el is lehet menteni egy fájlba. Az elmentett adatokat később bármilyen programmal fel lehet dolgozni. A program elérhető LabVIEW forrásként is, de LabVIEW nélkül is futtatható állományként. A korábban említett extra funkciók (folyamatos nyomásmérés, motorvezérlés) szintén megvalósíthatók LabVIEW-ban, ezek forrását is mellékeljük.

4.2.4. Kiértékelés saját szoftverben

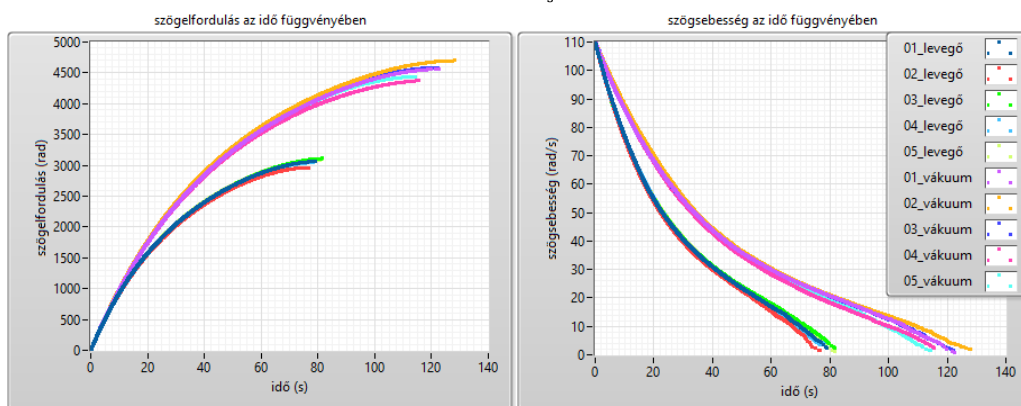
Az arduinós mérési adatok PC-n történő feldolgozására kifejlesztettünk *egy saját szoftvert*. Ez az Arduinóról érkező adatokat valós időben fogadja, feldolgozza, valamint CSV formátumú állományban tárolja. Offline üzemmódban képes a jelenség szimulálására, valamint időpontokénti elemzésre.



7. ábra: A saját fejlesztésű szoftver által megjelenített adatok

5. Eredmények

A problémafelvetésben szereplő kérdésünkre a mérési eredményeink alapján végre választ adhatunk. A 10. ábra grafikonjain szemléltetjük a fidget spinner szögelfordulását, majd a szögsebességét az idő függvényében. A grafikonokon 5 darab normál légköri nyomáson, illetve 5 darab 17%-os vákuumban (17 kPa nyomáson) történt mérés eredményeit vetettük össze. Úgy illesztettük az adatsorokat, hogy a $t = 0$ s-os kezdőidőponttól ábrázoltuk a $110 \frac{\text{rad}}{\text{s}}$ -os szögsebességről történő csökkenést.



8. ábra: Mérési eredményeink 17%-os vákuumban és normál légköri nyomáson

A grafikonokat összehasonlítva láthatjuk, hogy ugyanakkora sebességről vákuumban 40 másodperccel tovább tart a szabad forgás. Ez arra utal, hogy a fékező hatásban volt szerepe a légellenállásnak. Ugyanakkor vákuumban is jelentős a fékező hatás, ezt már a csapágyunk időfüggő súrlódása okozza. Az, hogy nemlineáris módon csökken a szögsebesség még vákuumban is, utalhat arra, hogy a súrlódás is sebességfüggő, ami a középiskolai fizikát ismerők számára meglepő állítás lehet. Ahhoz, hogy egészen korrekten körbejárjuk a problémát, további mérésekre van szükség (lásd 6. bekezdés).

A fidget spinner tehetetlenségi nyomatékának mérése, valamint a szöggyorsulás időbeli változásának vizsgálata után lehetőség nyílna a súrlódási nyomaték feltételezett sebességfüggésének feltárására, ugyanakkor egyéb tényezők is állhatnak a jelenség hátterében, mint például az üzemi hőmérséklet esetleges időbeli változása.

6. Tovább lépés

Tanulmányunkban bemutattunk egy lehetséges mérési eljárást arra, hogyan lehet optikai jellel megoldani a vezeték nélküli jelátadást, akár a vákuumban lévő áramkörök vezérlése, illetve az ott mért jelek mérése céljából. A célunk, hogy úgy fejlesszük tovább a rendszerünket, hogy iskolai használatra alkalmas legyen, így figyelembe kell vennünk esetlegesen felmerülő problémákat.

Az első ilyen, hogy bizonyos iskolában eleve rendelkezésre állnak vákuumkamrák és szivattyúk, így náluk a *nyomá szenzor csatlakoztatását* kellene megoldani. Ha a csatlakozók azonban nem a ma elterjedt szabványoknak felelnek meg, a feladat komplikálttá válik. Célszerű lehet ezért egy abszolút nyomásmérőt használni, és a vákuumkamrán belül elhelyezni.

Egy másik lehetséges nehezítő körülmény az *edény falának hatása az optikai jelre*. A vastag üveganyag részben elnyeli a lézertényt, gyakran fényszóródás is tapasztalható, sőt ha nem merőleges a fényt az üveg határfelületére, az edény megtöri a fény sugarat. Ezek mind-mind megnehezítik a lézertény külső detektálását, a fototranzisztor megfelelő beállítását. Ez a probléma is megoldódik, ha egy fotokaput a búra alatt helyezünk el.

Ha azonban így járunk el, mindkét szenzor esetében meg kell oldani, hogy a jelét a vákuumkamrán kívülre továbbítsuk. Az elektromos vezetékeken történő jelátadás nem célszerű, hiszen így még több helyen kellene megakadályozni a levegő-visszáramlást. Felmerül, hogy helyezzünk egy mikrovezérlőt is a vákuumkamrába, így lokálisan mérhetjük a nyomá szenzor és a fotokapu jelét, sőt kapcsolhatjuk közvetlenül a motor áramkörét. Nyilván olyan eszközt kell választani, amely alkalmas a *mért adatok vezeték nélküli továbbítására*, például a Bluetooth- és Wifi-kommunikációra képes, Arduino-kompatibilis ESP32 chipet [23].

Az eredmények validálását is tervezzük a továbbiakban. A játék csapágóját egy ismert katalógus-adatokkal rendelkező, pl. SKF gyártmányú csapággal helyettesíthetjük, és meghatározhatjuk a mozgás közben a csapágyra ható radiális és axiális erőket. Ezek ismeretében az SKF által alkotott modell [24] és az adott típusú csapágy paraméterei alapján kiszámítható a csapágyra ható súrlódási nyomaték a forgás sebességének függvényében. Ha az így átszerelt fidget spinner újramért kinematikai jellemzőiből meghatározzuk a β szöggyorsulást (a szögforgulás második deriváltját), illetve független módszerrel megmérjük a tömegeloszlást megadó tehetetlenségi nyomatékokat (θ), és ezek szorzatát megadjuk a sebesség függvényében, akkor gyakorlatilag megadható a teljes fékező nyomaték (ΣM) a sebesség függvényében a forgómozgás alapegyenlete ($\Sigma M = \theta \cdot \beta$) szerint. Így lehetőség lesz az elméleti és mért értékek összevetésére.

7. Oktatási tapasztalatok

A Szegedi Radnóti Miklós Kísérleti Gimnáziumban 3 pályamunka készült az Eötvös Loránd Fizikai Társulat és a National Instruments által támogatott fizikatanároknak és tanítványaiknak szóló, a myDAQ mérőcsatlakozó eszköz használatát és LabVIEW-ban történő programozását propagáló pályázatra [25]. A jelen tanulmány magját a Somogyi Anikó, Kovács Kornél és Balog Katalin által írt Ismétlés a tanulás anyja - Periodikus jelek generálása és kiértékelése pályamunkában is szerepeltettük. Az iskolában heti rendszerességgel myDAQ-szakkört, illetve hetente kétszer programozási szakkört tartanak e tanulmány szerzői. A bemutatott mérést egyszerűbb kísérleti elrendezéssel megvalósítottuk kisebb (3 fő), majd nagyobb létszámú (10 fő) szakkörökön, illetve osztálytermi demonstrációs kísérletként is felhasználtuk.

A Műszaki Informatika Kutatócsoport tagjai közül többen maguk is középiskolai tanárok. A munkacsoport számos, Edaq530-ra, illetve Arduinóra fejlesztett, tanulók között kipróbált mérést publikált az elmúlt évtizedben.

8. Összefoglalás

A műszaki informatika az informatika és az informatika között egy természetes kapcsolatot teremthet. Tapasztalataink szerint érdemes a méréseket meghonosítani az osztályteremben is, hiszen motivációs szerepén túlmutatva hasznosul a programozás tanításában is. Tanulmányunkban egy konkrét mérés kivitelezésére és az adatok kiértékelésére adtunk néhány hatékony megoldást, melyek más mérések kapcsán is hasznosíthatók. Ennek érdekében a forráskódokat mellékeljük a tanulmányhoz [21].

Köszönetnyilvánítás

Az Innovációs és Technológiai Minisztérium ÚNKP-19-3-SZTE-257 kódszámú Új Nemzeti Kiválóság Programjának szakmai támogatásával készült.”



A tanulmány elkészítését a Magyar Tudományos Akadémia Tantárgypedagógiai Kutatási Programja támogatta.

Az előadó konferencián való részvételét az EFOP-3.6.2-16-2017-00015 (2019/10) pályázat programja támogatta.

Irodalom

1. *A Micro:bit Foundation honlapja.*
<https://microbit.org/hu/> (Utolsó megtekintés dátuma: 2019. november 4.)
2. Y. Rogers, V. Shum, N. Marquardt, S. Lechelt, R. Johnson és H. Baker: *From the BBC Micro to micro:bit and Beyond: A British Innovation.* Interactions, 24. 74-77. (2017)
3. *Arduino - Home.*
<https://www.arduino.cc/> (Utolsó megtekintés dátuma: 2019. november 4.)
4. M. El-Abd: *A Review of Embedded Systems Education in the Arduino Age: Lessons Learned and Future Directions.* Int. J. Eng. Ped., 7. köt., 2. sz. (2017)
5. MISZAK.
<http://www.inf.u-szeged.hu/miszak/> (Utolsó megtekintés dátuma: 2019. november 4.)
6. Gingl Z. és mtsai: *Műszaki informatikai megoldások a modern középiskolai oktatásban.* Előadás INFODIDACT 2017, Zamárdi (2017)
7. *A megismerés folyamata* | Fizika - 9. évfolyam | Sulinet Tudásbázis.
<https://tudasbazis.sulinet.hu/hu/termesztudomanyok/fizika/fizika-9-efolyam/a-megismeres-folyamata/a-megismeres-folyamata> (Utolsó megtekintés dátuma: 2019. november 4.)
8. H. Zieris, H. Gerstberger és W. Müller: *Using Arduino-Based Experiments to Integrate Computer Science Education and Natural Science.* Potsdam Publication Portal, (2013)
9. L. Mesquita, G. Brockington, P. A. de Almeida, M. E. Truyol, L. A. Testoni és P. F. F. Sousa: *Using a fidget spinner to teach physics.* Phys. Educ., 53. köt., 4. sz. (2018)
10. R. Tanel, M. Büyükdede és A. Çoban: *Using fidget spinners in teaching some physics concepts.* Phys. Educ., 54. köt., 3. sz. (2019)
11. D. Macisaac: *Fidget spinner physics sites.* The Physics Teacher, 55. köt. (2017)
12. Z. Gingl, R. Mingesz, P. Makra és J. Mellár: *Review of sound card photogates.* Eur. J. Phys., 32. köt., 4. sz. (2011)

13. Kelemen A.: Digitális jelfeldolgozás | Digitális Tankönyvtár.
https://www.tankonyvtar.hu/hu/tartalom/tamop412A/2011-0013_kelemen_digitalis_jelfeldolgozas/adatok.html (Utolsó megtekintés dátuma: 2019. november 4.)
14. *MPX4250DP20 Datasheet, PDF* - Alldatasheet.
<https://www.alldatasheet.com/>. (Utolsó megtekintés dátuma: 2019. november 4.)
15. *ArduinoModules, KY-008 Laser Transmitter Module*, ArduinoModulesInfo.
<https://arduinomodels.info/ky-008-laser-transmitter-module/>
16. *How the Phototransistor Circuit Works*,
<http://learn.parallax.com/tutorials/robot/shield-bot/robotics-board-education-shield-arduino/chapter-6-light-sensitive-11> (Utolsó megtekintés dátuma: 2019. november 4.)
17. *NI myDAQ User Guide* - National Instruments.
<http://www.ni.com/pdf/manuals/373060g.pdf>
18. *Learn LabVIEW: intro to graphical programming in NI LabVIEW* - National Instruments.
<https://www.ni.com/getting-started/labview-basics/> (Utolsó megtekintés dátuma: 2019. november 4.)
19. Mingesz R. és Gingl Z.: *Mérés és adatgyűjtés - laboratóriumi jegyzet*, Digitális Tankönyvtár.
https://www.tankonyvtar.hu/hu/tartalom/tamop412A/2011_0104_SZTE-3_Meres_es_adatgyujtes/adatok.html. (Utolsó megtekintés dátuma: 2019. november 4.)
20. Z. Gingl, G. Makan, J. Mellár, G. Vadai, és R. Mingesz: *Phonocardiography and Photoplethysmography With Simple Arduino Setups to Support Interdisciplinary STEM Education*. IEEE Access, 7. köt. (2019)
21. *Fidget spinner – Infodidact 2019 | MISZAK*.
<http://www.inf.u-szeged.hu/miszak/fidget-spinner-infodidact-2019/> (Utolsó megtekintés dátuma: 2019. november 4.)
22. Piláth K.: *Adatgyűjtés Excelben – Fizika kísérletek*.
<https://pilath.wordpress.com/adatgyujtes-excelben/> (Utolsó megtekintés dátuma: 2019. november 4.)
23. *ESP32*, Wikipedia
<https://en.wikipedia.org/wiki/ESP32>
24. *Using a friction model as engineering tool*. Evolution SKF 2006
<http://evolution.skf.com/us/using-a-friction-model-as-an-engineering-tool-2/> (Utolsó megtekintés dátuma: 2019. november 4.)
25. *MyDAQ Pályázat Fizikatanároknak*.
<http://sukjaro.eu/ELFT-NI-palyazat/> (Utolsó megtekintés dátuma: 2019. november 4.)