# A Tinkercad Arduino-szimulátor alkalmazása az online programozásoktatásban

Somogyi Anikó[1], Makan Gergely[2], Kelemen András[3], Mingesz Róbert[4]

{[1]somogyia, [2]makan, [4]mingesz}@inf.u-szeged.hu;
[1,2,4] SZTE TTIK Műszaki Informatika Tanszék
[3]kelemen.andras.felix@szte.hu
[3] SZTE JGYPK Informatika Alkalmazásai Tanszék
[1,3] Szegedi Radnóti Miklós Kísérleti Gimnázium

**Absztrakt.** A COVID-19 világjárvány miatti digitális átállás az informatikaoktatást is közvetlenül érintette. Az online munkaforma a műszaki informatikai jellegű tárgyak oktatásában különleges kihívást jelent, hiszen az intelligens elektronikai eszközök programozásának tanítása alapvetően laboratóriumi környezetben valósul meg. Azonban a rendelkezésre álló szimulátorprogramok lehetőséget nyújtanak, hogy a képzés során a megkövetelt elektronikai és informatikai ismereteket a hallgató vagy diák elsajátítsa akár online oktatás során is. A tanulmányban bemutatjuk a regisztráció után ingyenesen használható online szimulációs programcsomagot, a Tinkercadet. A felületen elektronikai áramkörök valósághű modellezése, a virtuális Arduino UNO-hoz áramköri elemek, szenzorok, aktuátorok csatlakoztatása lehetséges, és az eszköz szöveges kódolással (C/C++) vagy blokk alapú környezetben is programozható böngészőben. A platform a frontális oktatás alternatívájaként lehetővé teszi az egyéni és kiscsoportos tanulói aktivitást is a köz- és felsőoktatásban egyaránt. Munkánkban megosztjuk oktatási tapasztalatainkat a programozók és informatikatanárok képzésében, továbbá ismertetünk néhány konkrét gyakorlatot, amelyet a hallgatóink megvalósítottak a digitális oktatás során.

**Kulcsszavak:** műszaki informatika, Arduino, online oktatás

## 1. Bevezetés

A Szegedi Tudományegyetem Természettudományi és Informatikai Karának Műszaki Informatika Tanszékén több olyan kurzust is oktatunk, amelyen intelligens elektronikai eszközök programozása témakörben az Arduino-programozás hangsúlyos szerepet kap. Tanárszakos hallgatóink számára akkreditáltunk egy kurzust, Az informatika műszaki alkalmazásai címmel, melynek keretén belül a hallgatók többek között interdiszciplináris laborgyakorlatokat hajtanak végre. Korábban publikáltuk a gyakorlatokon előforduló feladatok tartalmát, céljait és tapasztalatainkat [1,2]. Az utóbbi években több alapozó kurzusunk tananyagában (Műszaki alapismeretek, Elektronika alapjai programozóknak) is szerepelnek az Arduino platformmal kapcsolatos alapismeretek, sőt villamosmérnök hallgatóink tehetséggondozó óráin is hangsúlyos szerepet kapnak.

A 2019/20-as és 2020/21-es tanévek új kihívásokat támasztottak a tárgy oktatói elé. A SARS-COV19 vírus terjedése miatti korlátozások a megszokott oktatási formákat is felülírták. Mivel ezek a gyakorlatok alapvetően laboratóriumi környezetben, mérőeszközökkel, folyamatos tanári konzultációval zajlottak korábban, így nem kis kihívást jelentett a műszaki tartalmú gyakorlati anyag áthelyezése az online térbe. A távoktatás az Arduino esetében otthon elvégzett gyakorlatokkal és párhuzamos online konzultációval egészen jól megoldható, ha a hallgató beszerzi a szükséges eszközkészletet, ugyanakkor erre nem kötelezhető [3]. A gyakorlatainkon ezért a fizikai eszközöket virtuális műszerekre cseréltük, és a laborfeladatokat átalakítottuk szimulációs feladatokká. A különböző szimulációs programok programozásoktatásban való hasznosítására jó példákat találhatunk a szakirodalomban is [4], sőt a pandémia miatti lezárások alatt más felsőoktatási intézmények is alkalmazták ezeket műszaki tartalmú gyakorlataik oktatása során [5]. Fontos, hogy a szimulációs feladatok segítségével

ugyanazokat a tudáselemeket lehessen elsajátítani, hiszen a hallgatónak később ezeket valódi eszközökön is kell tudnia alkalmazni. Hasznos továbbá, ha a használt környezet valósághű, tartalmazza a gyakorlatok elvégzéséhez szükséges szenzorok jó részét, vagy legalábbis alternatív eszközökkel megoldhatók.

Az Arduino népszerűségének és széles körben való elterjedésének köszönhetően különböző szintű Arduino-szimulátorokból szerencsére nincs hiány [6,7]. A szimulátorok jó része nem oktatási célra, hanem prototípuskészítésre, tesztelésre készültek. A mi választásunk az Autodesk nagyvállalat Circuits nevű alkalmazására esett, amely Tinkercad honlapon lévő szoftvercsomag egyik alkalmazása. Ezt a programot komoly oktatási portálok [8,9] és az Arduinót népszerűsítő honlapok is ajánlják, bár említik a korlátait is [10,11].

A szoftver oktatásban való alkalmazásával kapcsolatban a tanulmány szerzőinek is volt korábbi tapasztalata, hiszen a MTA-SZTE Műszaki Informatika Szakmódszertani Kutatócsoport (MISZAK) által fejlesztett oktatóanyagok, példaalkalmazások és videóleckék egy része ezzel a programmal készültek [12,13].

# 2. A Tinkercad Circuits szimulátor

## 2.1. A Tinkercad platform általános bemutatása

A Tinkercad [14] platform egy 3D-modellezésre alkalmas, egyszerűen kezelhető, ingyenes, online CAD-szoftver 2010 óta érhető el a felhasználók számára. 2013-ban a későbbi tulajdonos, az Autodesk bővítette a Circuits ('áramkörök') alkalmazással, míg napjainkban a CodeBlocks nevű, programozható, LEGO-elemeket is tartalmazó 3D-tervezőprogrammal [15]. A magyar nyelven is elérhető szoftverek alkalmasak iskolai konstruktív geometria (pl. STEM projektekben 3D-nyomtatáshoz exportálható tervek készítése) elektronika és programozás oktatásban történő felhasználásra [16].

## 2.2. A Tinkercad biztosította lehetőségek az oktatásban

A Tinkercad fejlesztői számos, kifejezetten oktatási célú eszközt építettek be a platformba. A kidolgozott adatvédelmi protokollal rendelkező honlapon elkülönülnek a tanár/oktató és a diák/hallgató felhasználók. A diákokat osztályokként (Classroom) fejleszthetik a kollégák, melyhez kész, jól kidolgozott óraterveket biztosít a honlap. A Tinkercad Circuits jelen van az Arduino-projektekben is bővelkedő Instructables honlapon, YouTube-csatornájukon rendszeresen tesznek közzé oktatóvideókat, de jól reprezentálják magukat a közösségi média felületein is, ahol folyamatosan értesülhetnek a felhasználók az elérhető újdonságokról [17,18].

## 2.3. Fájlkezelés a Tinkercadben

A Tinkercad Circuits alkalmazásra kattintva automatikusan létrejön és mentésre kerül egy új fájl (ún. terv) egy fantázianévvel, és bekerül a saját gyűjteményünkbe. A főoldalra visszalépve csempeszerű elrendezésben látjuk valamennyi korábbi fájlunkat, a legfrissebbel kezdve időrendi sorrendben. A csempe jobb felső sarkára kattintva beállíthatók bizonyos tulajdonságok, mint például a terv neve, láthatósága (privát vagy nyilvános), de itt lehet megkettőzéssel klónt készíteni a tervről.

Lehetőség van a fájlok projektekbe rendezésére is. Egy projekt létrehozása után visszatérve a Circuits gyűjteményünkbe, a terv beállításai között található Ugrás projektre címszónál hozzárendelhetjük a kívánt projekthez. Hasznos, hogy a 3D tervek és a Kódblokkokban készült tervek is közös projektbe rendezhetők a Circuits-tervekkel, így egy helyen kezelhetők összetett, nagyobb oktatási projektek tervei.

A létrehozott fájlok alapértelmezésben mindig privát típusúak, viszont a fájl megnyitása után a Megosztás opciót választva van lehetőség Emberek meghívására: a link ismeretében más is megnyithatja, módosíthatja a tervet. Van lehetőség a terv nyilvánossá tételére is, ez esetben egy a Tinkercad-

közösség számára elérhető, kereshető fájl jön létre. Az ilyen publikus tervek remixelhetők, továbbfejleszthetők mások által, és természetesen mi is felhasználhatjuk más felhasználók nyilvános munkáit.

## 2.4. Tinkercad által támogatott oktatási formák

Az itt leírtak alapján többféle munkaformában alkalmazhatók ezek a Tinkercad-tervek az oktatásban. Egyrészt lehetséges frontális (akár online) előadásokon „élőben" szimulálni egy-egy kísérletet sokkal egyszerűbben, mint fizikai eszközt kamerázva. Előre elkészített online tananyagokhoz pedig egy képernyőrögzítésre alkalmas programmal meglehetősen élethű, szép szimulációk készíthetők.

Egy másik természetesen adódó munkaforma az önálló, egyéni munka, amelyre online gyakorlati órákat is lehet felépíteni. Itt a hallgatónak adhatunk olyan feladatot, amelyben teljes mértékben neki kell elhelyezni a szükséges eszközöket, áramköri elemeket a tervben, majd összeállítani az áramkört és azt később programozni. Az is lehetséges, hogy előre elkészített áramköri részeket, esetleg kódrészletet hozunk létre, és ezt nyilvánossá téve a hallgató remixelheti a tervet, és azt saját terveként továbbfejlesztheti.

Egy harmadik, az online oktatás során szintén hasznos munkaforma, hogy a hallgatók tudnak távolról ugyanazon a terven dolgozni, így kooperatív, kiscsoportban végzett feladatmegoldásra is van lehetőség.

## 2.5. A Tinkercad áramkörmodellező felülete

A Tinkercad áramkörépítő grafikus felületén „fogd és vidd" (drag & drop) típusú szerkesztési lehetőséget kap a felhasználó: az eszköztárból kiválasztott absztrakt objektumot (áramköri elem) a számítógép egere segítségével behúzhatunk a grafikus felület megfelelő helyére. Az áramköri elemeket (pl. ellenállások, diódák, tranzisztorok, áramforrások) szintén az egér segítségével köthetjük össze virtuális vezetékekkel így hozhatunk létre zárt áramköröket. Szerelőlap néven implementálták a próbapanelt (breadboard), amely segítségével a terveink átláthatóan összeállíthatók, vezetékelhetők.

Az eszközök között találhatunk Arduino UNO modellt, és hozzá számos szenzort, kiejelzőket, motorokat, továbbá jó néhány integrált áramkört, drivert is implementáltak. Bizonyos alap elektronikai laboreszközök (multiméter, labortápegység, oszcilloszkóp, tesztgenerátor) modelljei is elérhetők.

## 2.6. A Tinkercad fejlesztőkörnyezete

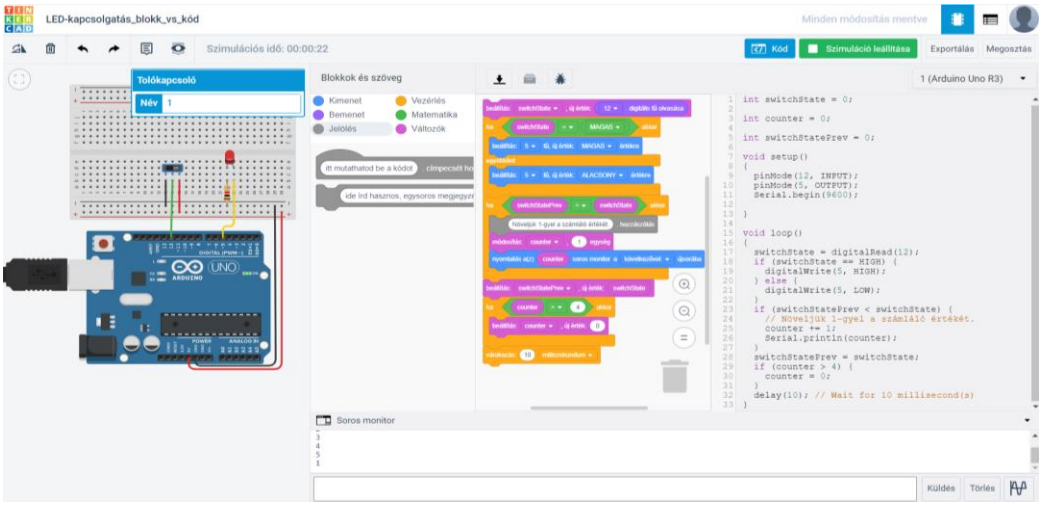
A Tinkercad jelenlegi verziójában több programozható eszköz található, az Arduino UNO R3 modellje, az ATtiny és a BBC micro:bit. Miután az eszközt elhelyezi a felhasználó a CAD-felületen, elérhetővé válik számára a Kód felület, ahol megkezdhető az eszköz programozása. Háromféle módban történhet a kódolás: a kezdő programozók számára a Scratchre [19] emlékeztető blokk alapú programozási felület nyújthat sikerélményt, míg a haladók a C/C++ alapú Arduino nyelven szöveges kódot írhatnak. A kettő közötti átmenetet és a tanulási folyamatot segíti a „Blokkok és szöveg" lehetőség, amelynél a szöveges kód felülete inaktív és nem szerkeszthető, viszont a blokkok elhelyezésekor automatikusan kiírja a program a blokkoknak megfelelő szöveges forráskódot. A BBC micro:bit esetén egyelőre csak a blokkprogramozás elérhető.

Az 1. ábrán látható áramkörben a Tinkercad felületen egy Arduino UNO 12. számú digitális bemenetére egy kapcsolót, míg az 5. számú digitális kimenetére egy LED-et csatlakoztattunk egy áramkorlátozó ellenálláson keresztül. A példa szimulációnkban megvalósítjuk, hogy amikor a kapcsolót jobbra vagy balra csúsztatjuk, a LED rendre be- és kikapcsol. A bekapcsolásokat számláljuk, de 5-nél visszaállítjuk a számlálót a kezdőértékre.

A megvalósításhoz három integer típusú változót hoztunk létre (magenta), elágazások programozásához és várakoztatáshoz vezérlési struktúrákat használtunk (narancssárga), melyek logikai feltételeinél a Matematika blokkjait építettük be (zöld). A kapcsoló állapotát a bemenet értékéből

olvassuk ki (lila), míg a LED állapotát, illetve a virtuálisan a számítógépre „nyomtatott” számlálóértéket a Kimenet (kék) blokkjaival írattuk ki. Kommentek beiktatására is van lehetőség (szürke). Megjegyezzük, hogy a setup() és a loop() függvények nem szerepelnek a blokkok között. A megfelelő blokk elhelyezésekor és beállításakor a program automatikusan beírja a beállításokat a setup()-ba, míg az egymás után fűzött blokkokkal valójában a loop() ciklusmagját kódoljuk.

Az ablak jobb szélső területén megjelenő szöveges kód másolással kiexportálható, és a fejlesztőkörnyezetben feltölthető a valós eszközre. A nemzetközi szakirodalom is egyre gyakrabban ajánlja ezt a lehetőséget általános iskolás korúakat tanító kollégák figyelmébe, bevezető szintű programozás tanításához [20,21].




**1. ábra:** Blokk alapú és szöveges programozás a Tinkercadben

A szöveges programozást is több hasznos funkció segíti. A Tinkercadben is elérhető néhány olyan függvénykönyvtár, amely nagyban megkönnyíti az Arduinóhoz csatlakoztatott aktív eszközök programozását (pl. Servo a szervomotorokhoz, NeoPixel a LED-szalagokhoz, stb.) Ezek listája és dokumentációja a kódoló felületen Elemtárak címszónál érhető el.

Vannak olyan összetett eszközök (Indítók), amelyeknél a programozható eszközhöz már eleve csatlakoztattak bizonyos alkatrészeket, és ezeknek a felületre húzásával a vezérlő vagy mérést végző kód is bekerül a szövegszerkesztőbe. Több programozható eszköz is elhelyezhető a grafikus felületen, ez esetben külön-külön programozhatók az eszközök.

Egy további funkció segíti a kódolást: a hibakereső lehetővé teszi a kód szakaszos futtatását, sőt a szünetekben a változók aktuális értékét megjeleníti a program, ha a kurzort a változó fölé visszük.

## 3. A Tinkercad alkalmazása a műszaki informatika online oktatásában

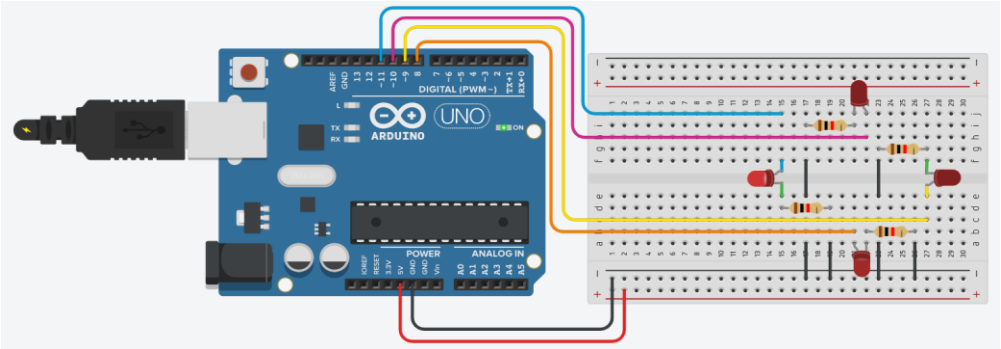
A következőkben bemutatunk néhány Tinkercadben készült Arduino-szimulációt, amelyeket különböző kurzusainkon a hallgatóink is elkészítettek. Nem titkolt célunk ugyanakkor az Arduino informatikatanárok körében történő népszerűsítése is: az itt bemutatott alkalmazásokat úgy igyekeztünk kiválasztani, hogy elsajátításukkal egy fokozatosan felépülő tudásanyagot nyújthassanak az Arduino-programozás kezdőinek, ugyanakkor a haladók is találjanak benne kihívást, tanulságos megoldásokat. A feladatok megoldásait és a forráskódokat tartalmazó Tinkercad-szimulációkat elérhetővé

tettük a honlapunkon [22]. Bár ezeket a gyakorlatokat itt szimulációként tárgyaljuk, hangsúlyozzuk, hogy az itt leírt kísérletek egy az egyben megvalósíthatók valódi eszközökkel is.

## 3.1. Digitális kimenet alkalmazása: futófény

Az Arduino-világban a LED-villogtatás programozása számít a „Hello world"-szintű bevezető programnak. Itt egyetlen LED-re kapcsolunk logikai magas (5V), illetve logikai alacsony (0V) jelszintet egyetlen digitális kimeneten keresztül. Ha az áramkörben elhelyezünk 4 db LED-et, amelyeket különböző digitális kimeneteken keresztül vezérelhetünk, ciklikus, illetve oszcilláló futófényeket is programozhatunk egyszerű vezérlési szerkezetek (pl. for ciklus) segítségével.

Egy másik alkalmazás a léptetőmotor (pl. 28BYJ-48) driveren (pl. ULN2003A) keresztül történő vezérlése [23], amely szerepel a tananyagban, viszont a Tinkercad nem tartalmazza ezt az aktuátort. Az online szimuláció [24] alapján jól látszik, hogy a ciklikus futófény programozása valójában – a sebességtől eltekintve - a léptetőmotor egy irányban történő mozgatásához szükséges logikai jelszintek beállításával egyenértékű. Így az online oktatás alatt a hallgatóknak a 2. ábrán szereplő áramkört kellett összeállítani, majd szimulálni 5 periódusnyi, óramutató járásával ellentétes irányú, majd rövid szünet után 5 periódusnyi óramutató járásával megegyező irányú ciklikus futófényt. Az ismétlésszámok növelésével és a lépések közötti időtartam csökkentésével a léptetőmotor oda-vissza mozgatása pl. kapunyitáshoz is megvalósítható, így a hallgatók által készült programokat később lehet léptetőmotoron is tesztelni [25].




**2. ábra:** Áramkör a léptető motort modellező futófény szimulációjához

## 3.2. Digitális bemenet alkalmazása: kapuvezérlő rendszer modellje

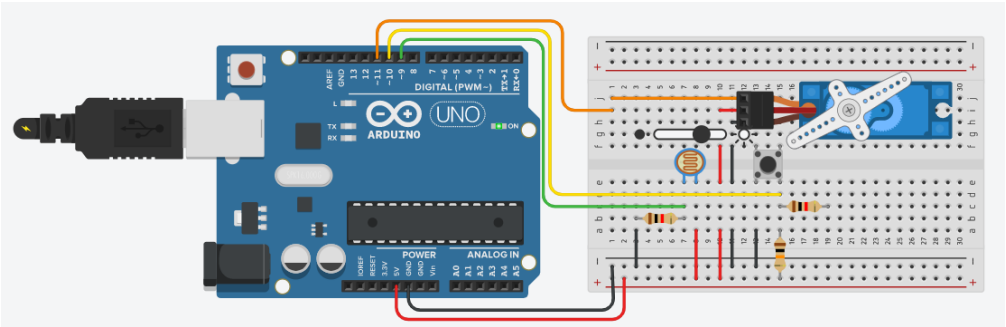
A digitális bemenetek alkalmazására fontos példák a két logikai szint váltására alkalmas kapcsoló és nyomógomb aktuális jelszintjének beolvasása. Ugyanakkor analóg szenzorokat is kapcsolhatunk digitális bemenetre: Az Arduino UNO esetében a digitális bemenetre kapcsolt 0 és 5V közötti jelek esetében 3V fölötti érték esetén a műszer logikai magas, míg alatta logikai alacsony jelet mér. (Az Arduino UNO-ban alkalmazott ATmega328 adatlapja alapján a bemeneti I/O portokon egy Schmitt trigger valamint egy szinkronizáló áramkör is gondoskodik arról, hogy a folytonosan változó jeleket is helyesen tudja feldolgozni a mikrovezérlő.)

Feladatként kitűzhető, hogy valósítsa meg a hallgató vagy diák egy kapuvezérlő rendszer modelljét a Tinkercadben. A kapu forgatását végző mechanikai szerkezetet ezúttal egy szervomotor reprezentálja. A szervomotor működtetéséhez elegendő egyetlen digitális kimenet alkalmazása, és programozásához elérhető a Servo függvénykönyvtár Tinkercadben is. A modellben a szervomotort 90°-kal egyik irányba elforgatjuk, majd rövid szünet után az eredeti helyzetbe visszaforgatjuk, ezzel reprezentálva a kapu nyitását és zárását.

A kapu forgásának elindítását a felhasználó vezérli, egy nyomógomb segítségével, melynek jelét az Arduino UNO digitális bemenetén olvassuk be. A nyomógomb lenyomása a szimuláció elindítása után egérrel kattintással lehetséges.

Ha a kapu útjába kerül egy akadály, amelyet valamilyen optoszenzor-rendszer (fotokapu) érzékel, vészleállításra van szükség. Ennek egy lehetséges megoldása, hogy a kapu zárt állásának vonalában helyeznek el egy fotokaput, illetve a kapu előtt 1-2 méterre egy másikat. Az akadályt figyelő mozgásérzékelőt a tényleges kísérletben és ebben a szimulációs feladatban is egyetlen fotoellenállás (3. ábra) segítségével egyszerűsítettük le, amelyet a környezeti fény világít meg, de ha kezünkkel eltakarjuk, akkor megváltozik a mért logikai szint. (Ha a hallgatók két fényérzékelőt használnak, a modell még jobban közelíthető egy tényleges fotokapu-rendszerhez.) A fotoellenállás analóg jelét ezúttal egy digitális bemenetre kapcsoltuk. Ezt egy valós tantermi kísérletben megtehetjük, hiszen a motor mozgását leállító akadályt a hallgató keze jelenti, amely kitakarja a terem környezeti fényét. Mivel a választott szenzort és a vele sorba kapcsolt előtétellenállást az Arduino UNO digitális bemenetére kapcsoltuk, megfelelő kísérleti körülmények esetén az tapasztalható, hogy a jel tulajdonképpen bináris és megfelelő, hogy változására a kapu megálljon. Ennek oka, hogy a kísérletben a fotorezisztor ellenállása oly mértékben változik, hogy az analóg jel egyértelműen olvasható digitális bemeneten. [1]. A kísérlet szimulációjában a fényviszony-változás megfelelő szimulálására úgy van lehetőség, hogy indítás után a szenzorra kattintás hatására megjelenő csúszkán manuálisan változtatjuk a „megvilágító fény erősségét”.

A hallgatók feladata, hogy olyan szimulációt hozzanak létre, melyben a kapunyitás (motormozgás) nem indul el addig, amíg be nem kapcsolják (felhasználó lenyomja a nyomógombot), vagy a mozgásérzékelő (fotoellenállás) valamit érzékel a fényútban. Indítás után, ha a fényútba kerül valami (felhasználó a sötétedés irányába mozgatja a csúszkát), akkor a veszély elhárításáig megáll a kapu mozgása, ami az akadály eltávolítása után automatikusan újraindul. A kapu bezáródása után újabb engedélyre (nyomógomb lenyomására) vár a program [26].

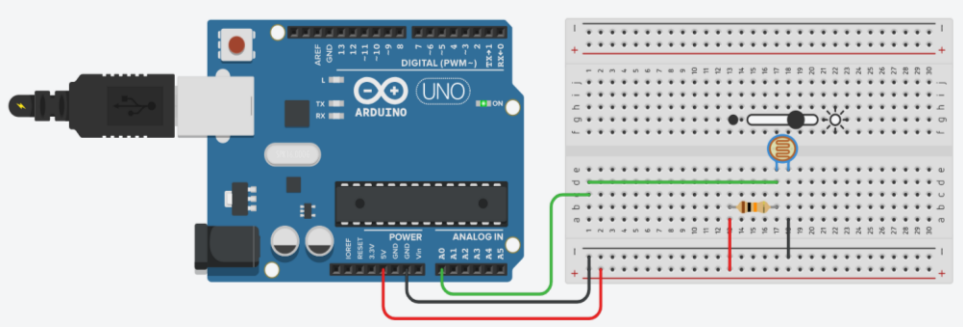



**3. ábra:** Áramkör a szervomotorral megvalósított kapuvezérlőrendszer-modell szimulációjához

## 3.3. Analóg bemenet és soros monitor alkalmazása: ellenállásmérés

Az Arduino analóg bemenetét feszültség- és ellenállásmérésre is alkalmazhatjuk [1]. Míg a laborgyakorlaton egy ismeretlen értékű ellenállás értékének a meghatározása a cél, addig a Tinkercadben a furatszerelt ellenállások ohmban mért ellenállása ismert a hallgató számára. Ha azonban a változtatható értékű fotoellenállást kapcsoljuk a feszültségosztóba, az ellenállásméréssel a félvezető eszköz elektromost tulajdonságainak vizsgálatára is lehetőség van. A szimuláció indítása után a csúszka mozgatása mellett megmérhető, hogy az erősebb megvilágítás esetén a szenzor ellenállása drasztikusan lecsökken. Itt egy említésre méltó önellenőrzési lehetőséget is ad a felhasználó számára a szoftver: található benne beépített multiméter is, amellyel szintén megvizsgálható a fényérzékelő ellenállása az áramkörből kivéve.

A Tinkercadben implementálták a számítógép és az Arduino közötti USB-porton történő soros kommunikációt is. A Soros monitor felnyitása után a grafikon ikonra kattintva a hagyományos fejlesztőkörnyezet (Arduino IDE) Soros plotteréhez hasonló diagramon lehet az idő függvényében megfigyelni az ellenállás értékét. [27]



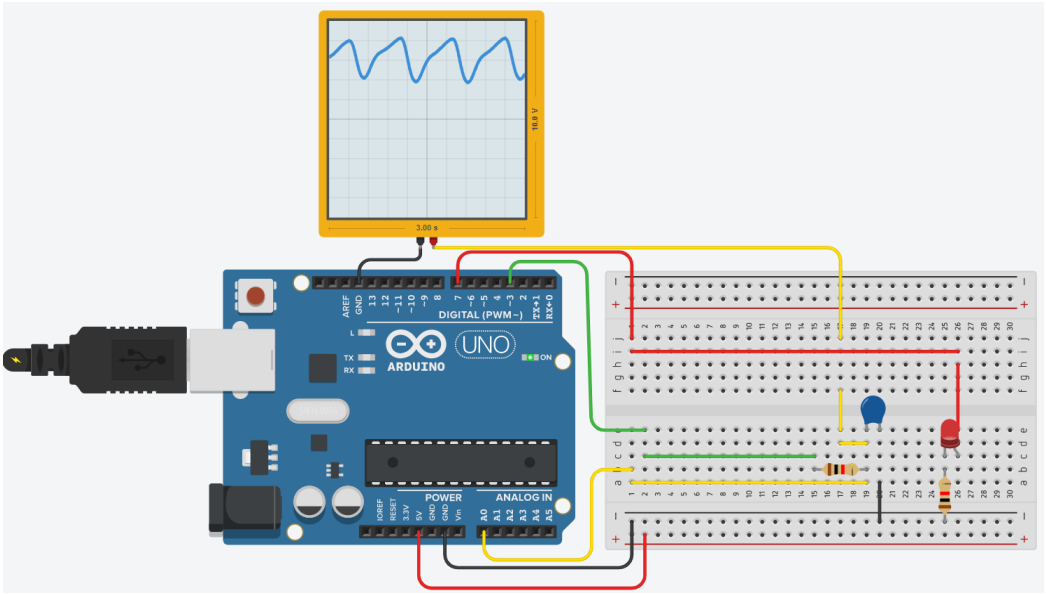

**4. ábra:** Áramkör a fotoellenállás különböző fényviszonyok között történő ellenállásmérésének szimulációjához

## 3.4. PWM jelgenerálás alkalmazása: fotopletizmográfiai mérésen alapuló pulzusmérés

Az Arduinóval végzendő fotopletizmográfiai méréshez a kutatócsoport által fejlesztett EDAQuino szenzor interfész használatát javasoljuk [28]. Ha azonban online alternatívát keresünk, nincs egyszerű dolgunk, ugyanis az ujjbegyben történő nyomásváltozás szimulálására nem alkalmas a Tinkercad.

Egyszerű periodikus jelek (négyszög-, szinusz-, háromszögjel) generálására ugyan lehetőséget nyújt a beépített Tesztgenerátor eszköze, viszont ezen jelek frekvenciáját a felhasználó állíthatja be, így a jel frekvenciájának mérése kevéssé tűnt megfelelő alternatívának, noha magának a szintmetszés algoritmusának, és a periódusidő, frekvencia meghatározásának programozását kiválóan el lehet sajátítani egy leegyszerűsített jelalakkal pl. háromszögjellel.

Egy megoldást szeretnénk arra mutatni, hogy hogy lehet mérésen alapuló jelalakot megvalósítani a szimulátorban. Az EDAQuino segítségével valós mérésként felvettük egy személy fotopletizmogram-jelét 100 Hz-es mintavételi frekvenciával. A mért jel néhány periódusnyi hosszát kivágtuk úgy, hogy azt ismételgetve folytonos jelet lehessen szimulálni, és eltároltuk egy tömbben. Mivel az Arduino UNO-n nem található analóg kimenet, ezért pulzusszélesség-modulációt alkalmaztunk: a PWM-típusú digitális jel kitöltési tényezőjét ciklikusan olvastuk ki a tömbből 10 ms-onként, majd egy aluláteresztő RC szűrő (passzív integrátor) segítségével átalakítottuk analóg jellé. Megjegyezzük, hogy mind PWM-jel, mind pedig az analóg jel alakjának grafikus megjelenítése is megoldható az Oszcilloszkóp eszköz segítségével az 5. ábrán látható módon. Ezt a jelet visszamértük az egyik analóg bemeneten.



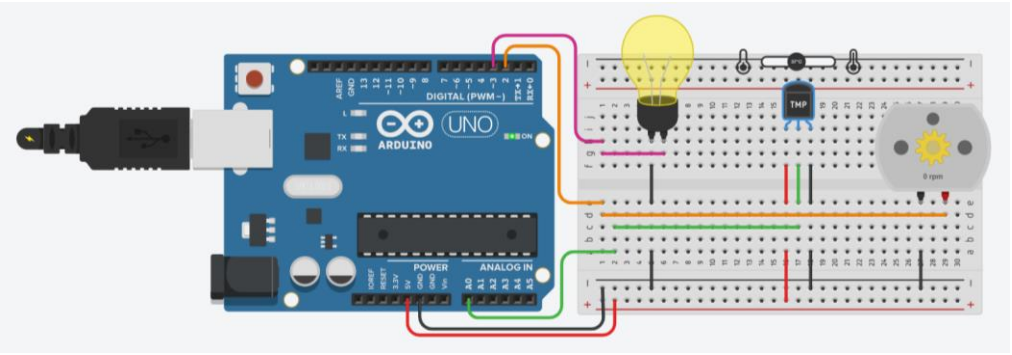

**5. ábra:** Áramkör a mért adatokból generált fotopletizmogramjel alapján történő pulzusmérés szimulációjához

A mért jelen egy lineáris értéktranszformáció (tükrözés) is végrehajtható szoftveresen, hogy a fotopletizmogramra emlékeztető jelet megkaphassuk, és a Tinkercad Soros plotterén megfigyelhessük. Ezután lehetővé válik az így kapott jel felszálló ágain történő szintmetszések időpontjai között eltelt időtartamok (periódusidő) mérése, illetve a reciprok értékét véve a pulzus meghatározása [29].

## 3.5. Hőmérsékletszabályozás

A hallgatók a gyakorlat során megismerkednek az on-off szabályzás elvével: feladatuk egy kétállásos hőmérsékletszabályozó rendszer modellezése. A hőmérséklet folyamatos monitorozása mellett, ha a hőmérséklet lecsökken egy bizonyos 1. számú szint alá, akkor bekapcsol a fűtés, amelynek hatására a hőmérséklet értelemszerűen megnövekszik, és egy 2. szint elérésénél a fűtés kikapcsol. Azonban, ha a rendszer túlmelegszik azaz a 2. számú szint fölé emelkedik, egy ventilátor is bekapcsol, és addig működik, amíg a hőmérséklet újra el nem éri a 1. számú szintet.

A Tinkercadben „látványos” fűtőellenállás nincs implementálva, így fűtőelemként egy izzót használunk. A hőmérséklet növekedésére nemlineáris módon csökkenő ellenállású félvezető analóg szenzor, a termisztor sajnos nem szerepel az elérhető eszközök között, viszont a TMP36 szenzor igen, így a mért analóg feszültség hőmérsékletté történő konverziója konverzió könnyen megvalósítható. A Tinkercad itt is ad egy ellenőrzési lehetőséget, hiszen a szimuláció futtatása közben megjeleníthető szabályozó csúszkán a program kiírja a beállított (és mérendő) hőmérséklet értéket, ami összevethető a mért értékkel. A ventilátort egy egyszerű egyenáramú motorral szemléltetjük. A fűtő-, illetve hűtő elemek hatása a hőmérsékletre sajnos nem érhető tetten a szimulációban, viszont még így, manuálisan „reagálva” és változtatva a hőmérséklet értékét is jól szimulálható és megérthető a szabályozórendszer működése [30].



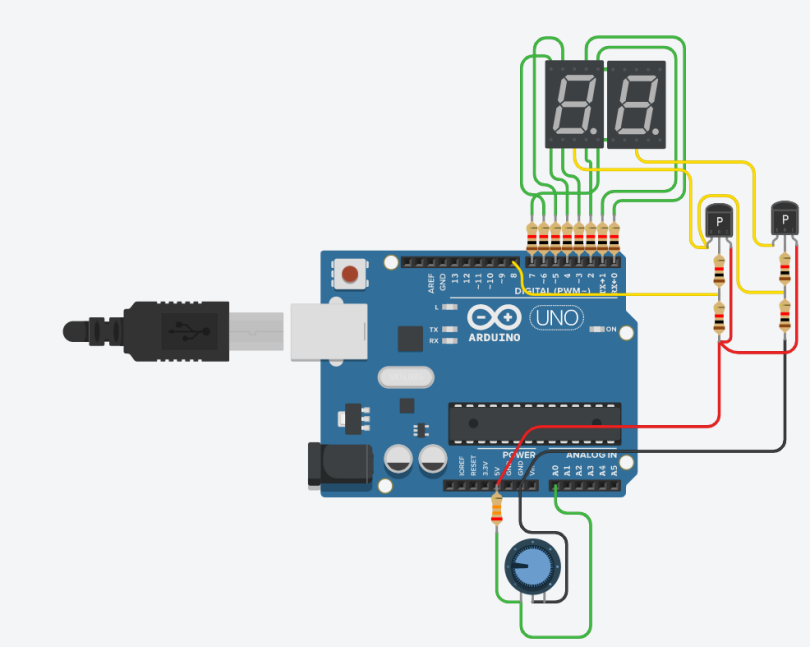

**6. ábra:** Áramkör a hőmérsékletszabályozó rendszer szimulációjához

## 3.6. Ellenállásmérés, 7-szegmenses kijelző vezérlése

Az ellenállásmérési feladatot kiegészíthetjük azzal, hogy egy a potenciométer egyik végpontja és a csúszóérintkezője közötti ellenállását két 7-szegmenses kijelzőn jelenítjük meg. A potenciométert egy referencia ellenállással sorba kötve megkapjuk a potenciométer ellenállását, ha a rajta eső feszültséget mérve az ADC kódot feszültségbe konvertáljuk és a feszültségosztó egyenletéből kifejezzük a mérendő ellenállást.

A 7-szegmenses kijelző vezérlését nagyban megkönnyíti, hogy a teljes portot, azaz mind a 8 bitet egy utasítással, a *PORTD* makróval tudjuk írni. Teljes port csak egy érhető el az Arduino UNO-n, a D port. Ennél a portnál viszont arra kell figyelni, hogy a 0 és az 1-es kivezetések a soros kommunikációval is osztoznak, így ha szükségünk van ezekre a bitekre, akkor nem ajánlott soros kommunikációt is használni mellette. A kivezetések módját digitális kimenetnek konfigurálhatjuk a DDRD makró segítségével, így nem kell egyesével beállítani a kimeneteket a pinMode() függvénnyel [31]. Mivel két kijelző van, de csak egy teljes port, ezért mindkét kijelző ugyanarra a portra van kötve és két tranzisztor segítségével van megoldva, hogy egyszerre csak az egyik legyen bekapcsolva. A két tranzisztort egy digitális kimenettel tudjuk vezérelni és amikor az egyik kinyit, akkor a másik zár. Ezt a kijelzőválasztó bitet viszonylag nagy frekvenciával (100 Hz) negáljuk, így a villogás nem látszik, de mindkét kijelző egyedileg is használható lesz. A Tinkercadben elérhető kijelzővezérlő IC is, de a hallgatók feladata lehet ennek a funkciónak a megvalósítása is.

Az ellenállást 0,1 kΩ-os felbontással tudjuk kiírni a kijelzőkre. Amikor a kijelzőválasztó bit logikai magas, akkor az első digittel, amikor logikai alacsony, akkor a második digittel kell kiindexelni karaktereket tartalmazó konstans tömböt. Ennek a módszernek az előnye, hogy nem kell a számkarakterek kezeléséhez egy hosszú, 10 esetet tartalmazó switch-case struktúrát használni, amiben minden számjegynek megfelelne egy-egy eset [32].




**7. ábra:** Áramkör a potenciométer ellenállásmérésén alapuló 7-szegmenseskijelző-vezérlés szimulációjához

# 4. A Tinkercad előnyei és hátrányai

Sorra vesszük a Tinkercad – oktatásban és tananyagfejlesztésben történő - használata során összegyűlt tapasztalatainkat két csoportra bontva: először bemutatjuk a program előnyeit és a benne rejlő lehetőségeket, majd a hátrányokat és a szimulátor alkalmazhatóságának korlátait is [11].

## 4.1. Előnyök

A távoktatás során a legfőbb előnye a programnak, hogy a laborgyakorlatok esetén legalább a gyakorlati jellegét meg tudtuk őrizni a kurzusoknak, és továbbra is megvalósítható volt a hallgatók aktív tanulása, a gyakorlás a frontálisan közölt vagy kiadott tananyagok elsajátítása helyett.

A tantárgyak közötti kapcsolatot lehet építeni komolyabb eszközbeszerzés nélkül. Az informatikaoktatás során be lehet vonni a fizika (elektronika) tudáselemeit, ugyanakkor az áramkörépítő alkalmazás a közoktatás szintjén alkalmas az egyszerű elektronikai mérések szimulálására is [33]. Az áramkörök breadboardon való megvalósítása nagyon szépen megtervezhető a szimulátorban: a vezetékek derékszögben hajlíthatók, így áttekinthetővé tehetők a „kísérleti elrendezések”.

Ha még hozzávesszük a lehetőségekhez a 3D geometriai tervező programokat, akkor könnyen belátható, hogy meglehetősen komplex STEAM-projektek támogatására is alkalmas a programcsomag.[1]

[1] A mozaikszó ’Science, Technology, Engineering, Art & Math’, azaz természettudományok, technológia, mérnöktudományok, művészet és matematikatudományterületek ötvözésére utal.

Internetelérés esetén bárhol, bármikor folytatható egy projekt, ami a programozás időszakát nagyon kényelmessé teszi. Később, amikor az Arduino és a hozzá csatlakoztatott egyéb eszközök rendelkezésre állnak, ki lehet próbálni a programot, de a kísérletet így már megelőzi egy hosszabb tervezés és szimulátorral segített tesztfolyamat, ami a tényleges kísérlet sikerét szinte garantálja. Osztálytermi környezetben, ha nincs elegendő Arduino Kit, akkor is mindenki haladhat a programozással, folyamatos visszajelzés mellett, később váltva kipróbálhatják a kész programot valós eszközökön. Nem csupán a tanulási folyamatban, de saját projektek készítésekor a tesztelés során elkerülhetetlen a hibák lokalizálása, melyhez a korábban részletezett hibakereső nagy segítséget nyújt.

## 4.2. Hátrányok

A következőkben ismertetjük a Tinkercad jelenlegi verziójának általunk felismert korlátait is.

Jó lenne, ha mobil eszközökön is elérhető lenne az applikáció. Ugyan létezik már iOS-változat (az Android-verzió jelenleg nem elérhető) a szoftverhez, viszont jelenleg a Circuits csak megnyitást enged, sem a tervek szerkesztése, sem pedig a szimuláció nem megengedett. [34]

Oktatási szempontból mindenképp hangsúlyozni kell, hogy nagyon meggyorsítja a tesztelési, hibakeresési folyamatot egy előzetes szimuláció készítése. Ugyanakkor – online oktatás során - nem fog olyan hibákat produkálni, mint a valóság, ahol hibás lehet egy csatlakozás, vezeték vagy komponens. Ez előny is, kevesebb hibakeresésre van szükség, ugyanakkor fontos készségeket nem sajátítanak el.

Elektronikai szempontból elmondható, hogy bizonyos alkatrészek fizikai tulajdonságai nem, vagy nem helyesen vannak megvalósítva. A LED-eknek a nyitófeszültsége a LED színével van összefüggésben, viszont itt színtől függetlenül ugyanannál a feszültségértéknél nyitnak ki a világító diódák. Bizonyos eszközöknél a multifunkcionalitást hiányoljuk: a multiméter és a függvénygenerátor csupán a legalapvetőbb funkcióval rendelkezik. A szoftvert viszonylag könnyű úgy terhelni (pl. túl gyakori mintavételezéssel), hogy lelassuljon a szimuláció közben, és ne valós időben mutassa a kísérletet.

A hobbielektronika világa olyan rohamosan fejlődik, ezzel a Tinkercad nem veszi fel a versenyt. Viszont eléggé alapvetőnek számító alkatrészek is hiányoznak (pl. termisztor, stepper motor). Amit még hiányolunk, hogy nincsen lehetőség saját eszköz tervezésére, bevitelére. Jó lenne, ha további programozható eszközök (pl. Arduino Nano vagy Arduino Mega) elérhetők lennének. Korábban elérhető volt az ESP8266 wifi modul, de sajnos biztonsági okokra hivatkozva már nem elérhető.

Ami a programozási környezetet illeti, jó lenne, ha bővítenék a blokkprogramozásban elérhető blokkokat, továbbá, ha a felhasználó tudná bővíteni a saját függvénykönyvtárát, esetleg saját blokkokat létrehozni. Ezen kívül nem túl felhasználóbarát a szövegszerkesztő (nincs vonalzó, automatikus behúzások, automatikus zárójelbezárás vagy épp gépelési hiba jelzése (kisbetű, nagybetű), javítása. Természetesen külső szövegszerkesztőben elért külalakot megőrzi a Tinkercad is.

# 5. Összefoglalás

Tanulmányunkban oktatási tapasztalataink alapján bemutattuk a Tinkercad Circuits ingyenes, online Arduino-szimulátort, melyet az online programozásoktatásban alkalmaztunk. Bemutattuk a Tinkercad oktatást támogató funkcióit, az áramkörmodellezést lehetővé tévő grafikus objektumait és az intelligens elektronikai eszközök (Arduino UNO, ATtiny, BBC micro:bit) programozását lehetővé tévő fejlesztőkörnyezetet. Ezt követően ismertettünk konkrét szimulációkat, amelyeket az eredetileg laborgyakorlatnak meghirdetett műszaki informatikai kurzusaink online oktatásra történő átállásakor alkalmaztunk a gyakorlati feladataink alternatívájaként. Végül sorra vettük, milyen lehetőségeket rejt magában, ugyanakkor milyen korlátokkal rendelkezik a Tinkercad.

Bízunk benne, hogy az általunk bemutatott platform felkelti az informatikatanárok, oktatók érdeklődését, és saját oktatói munkájuk során is hasznosnak találják majd az általunk bemutatott ötleteket, megoldásokat.

## Köszönetnyilvánítás

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

## Irodalom

1. Makan G., Dóra A., Mingesz R., Gingl Z., Kopasz K., Mellár J.Z., Vadai G., *Interdiszciplináris műszaki gyakorlatok az informatikatanár szakon*. (2018). https://doi.org/10.6084/m9.figshare.7359203.v1
2. G. Makan, D. Antal, R. Mingesz, Z. Gingl, J. Mellár, G. Vadai, K. Kopasz, *Interdisciplinary Technical Exercises for Informatics Teacher Students*. Central-European Journal of New Technologies in Research, Education and Practice, (2019) 1. 21–34. https://doi.org/10.36427/CEJNTREP.1.1.382
3. *Arduino Remote Learning*. https://www.arduino.cc/remotelearning (Utoljára megtekintve: 2020. 10. 28.)
4. M.G. Jamil, S.O. Isiaq, *Teaching technology with technology: approaches to bridging learning and teaching gaps in simulation-based programming education*. International Journal of Educational Technology in Higher Education, (2019) 16. 25. https://doi.org/10.1186/s41239-019-0159-9
5. P.L. Rocca, F. Riggi, C. Pinto, *Remotely teaching Arduino by means of an online simulator*. Physics Education, (2020) 55. 063003. https://doi.org/10.1088/1361-6552/abaa21
6. *A Beágyazott rendszerek alapjai tantárgyhoz: taneszközkészlet, ajánlható irodalom, szoftverek. | Villamos Csoport*. http://vill.elmki.hu/taneszkozkeszlet-a-beagyazott-rendszerek-alapjai-tantargyhoz/#more-2270 (Utoljára megtekintve: 2020. 11. 2.)
7. *Top 10 Best Simulators for Arduino*. Projectiot123 Technology Information Website Worldwide, (2019). https://projectiot123.com/2019/03/15/top-10-best-simulators-for-arduino/ (Utoljára megtekintve: 2020. 10. 28.)
8. *Teacher Resources for Computer Engineering*. TryEngineering.Org Powered by IEEE,. https://tryengineering.org/category/teacher-resources/ (Utoljára megtekintve: 2020. 10. 28.)
9. *Electronics Online - FUSE - Department of Education & Training*. https://fuse.education.vic.gov.au/Resource/LandingPage?objectId=2c138337-0fcf-4425-92d3-a8e39e9015d4 (Utoljára megtekintve: 2020. 10. 28.)
10. *The Arduino Simulator you've been looking for!* Programming Electronics Academy, (2019). https://www.programmingelectronics.com/arduino-simulator-tinkercad/ (Utoljára megtekintve: 2020. 11. 5.)
11. *Arduino Simulator Q & A*. Programming Electronics Academy, (2019). https://www.programmingelectronics.com/tinkercad-part-2/ (Utoljára megtekintve: 2020. 10. 28.)

12. *Arduino alkalmazása a fizika és a digitális kultúra oktatásában | MISZAK.* http://www.inf.u-szeged.hu/miszak/arduino-alkalmazasa-a-fizika-es-az-informatika-oktatasaban/ (Utoljára megtekintve: 2020. 11. 2.)
13. *miszak-mta-szte | A MISZAK YouTube-oldala.* https://www.youtube.com/channel/UCak16GKllLDG1w1ZspKd3Ww (Utoljára megtekintve: 2020. 11. 2.)
14. *Tinkercad | Create 3D digital designs with online CAD.* https://www.tinkercad.com (Utoljára megtekintve: 2020. 10. 28.)
15. *Tinkercad.* Wikipedia, (2020). https://en.wikipedia.org/w/index.php?title=Tinkercad&oldid=985779734 (Utoljára megtekintve: 2020. 10. 28.)
16. *Autodesk | 3D Design, Engineering & Construction Software.* https://www.autodesk.com/ (Utoljára megtekintve: 2020. 10. 28.)
17. *Tinkercad tanárok számára.* https://www.tinkercad.com/teach (Utoljára megtekintve: 2020. 10. 28.)
18. *Instructables - Tinkercad circuits.* https://www.instructables.com/member/circuits/ (Utoljára megtekintve: 2020. 10. 28.)
19. M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, Y. Kafai, *Scratch: programming for all.* Communications of the ACM, (2009) 52. 60–67. https://doi.org/10.1145/1592761.1592779
20. B.N. Mohapatra, R.K. Mohapatra, J. Joshi, S. Zagade, *Easy Performance Based Learning Of Arduino And Sensors through Tinkercad.* International Journal of Open Information Technologies, (2020) 8. 4.
21. C. Vidal-Silva, J. Serrano-Malebran, F. Pereira, *Scratch and Arduino for Effectively Developing Programming and Computing-Electronic Competences in Primary School Children*, in: 2019 38th International Conference of the Chilean Computer Science Society (SCCC), 2019 1–7. https://doi.org/10.1109/SCCC49216.2019.8966401
22. *A Tinkercad Arduino-szimulátor alkalmazása az online programozásoktatásban | MISZAK.* http://www.inf.u-szeged.hu/miszak/infodidact2020/ (Utoljára megtekintve: 2020. 11. 5.)
23. A. Somogyi, A. Kelemen, R. Mingesz, *Motion tracking by an Arduino Due and Excel*, in: Proceedings of XXXIII. DidMatTech 2020 Conference. New Methods and Technologies in Education, Research and Practice, Eötvös Loránd University in Budapest, Trnava University in Trnava
24. *Stepper motor.* Wikipedia, (2020). https://en.wikipedia.org/w/index.php?title=Stepper_motor&oldid=986069649 (Utoljára megtekintve: 2020. 11. 2.)
25. *Ismételt futófény számlálóval.* https://www.tinkercad.com/things/5Bq0GhUsLDt (Utoljára megtekintve: 2020. 11. 1.)
26. *Kapuvezérlő rendszer szervomotorral.* https://www.tinkercad.com/things/bitdgKX3BAe (Utoljára megtekintve: 2020. 11. 1.)
27. *Ellenállásmérés.* https://www.tinkercad.com/things/cysTcfu2OOT (Utoljára megtekintve: 2020. 11. 1.)

28. Z. Gingl, J. Mellár, T. Szepe, G. Makan, R. Mingesz, G. Vadai, K. Kopasz, *Universal Arduino-based experimenting system to support teaching of natural sciences*. Journal of Physics: Conference Series, (2019) 1287. 012052. https://doi.org/10.1088/1742-6596/1287/1/012052
29. *Fotopletizmográfia pulzusszélesség-modulációval - pulzusmérés*. https://www.tinkercad.com/things/0YSWThiW8rJ (Utoljára megtekintve: 2020. 11. 1.)
30. *Hőmérsékletszabályozás*. https://www.tinkercad.com/things/lCLayymNgrW (Utoljára megtekintve: 2020. 11. 2.)
31. *Arduino - PortManipulation*. https://www.arduino.cc/en/Reference/PortManipulation (Utoljára megtekintve: 2020. 11. 3.)
32. *Ellenállásmérés, 7-szegmenses kijelző vezérlése*. https://www.tinkercad.com/things/gMqp1WRamdd (Utoljára megtekintve: 2020. 11. 3.)
33. *Elektromosság KIT*. https://sites.google.com/view/elektromossag-kit (Utoljára megtekintve: 2020. 11. 3.)
34. *Tinkercad for iOS*. https://apps.apple.com/us/app/tinkercad/id1469440830 (Utoljára megtekintve: 2020. 10. 28.)