

Alternatívák a LEGO EV3 helyettesítésére, az újgenerációs Mindstorms és a Stem:Bit készletek használata az oktatásban

Gaál Bence¹, Solymos Dóra²

{¹gaalbence, ²solymos.dora}@inf.elte.hu
ELTE IK

Absztrakt. Aki foglalkozott már robotika oktatásával, biztosan találkozott a LEGO Mindstorms EV3 modelljével, amely 2013 óta meghatározó szerepet tölt be ezen a területen. Ezen robotok gyártása azonban hivatalosan is megszűnt és támogatásuk is megszűnik a közeljövőben. Cikkünkben az új generációs LEGO Mindstorms Robot Inventor bemutatása mellett egy olcsóbb alternatívát is be kívánunk mutatni a Stem:Bit képében, amelyeket az EV3 helyettesítésére ajánlunk. Az eszközök összehasonlításának fontos eleme, hogy megvizsgáljuk azt is, milyen funkciókat biztosítanak az adott eszközök az EV3-hoz viszonyítva és melyeknek milyen előnyei és hátrányai vannak a robotika oktatásában.

Kulcsszavak: robotika, programozás, Mindstorms EV3, Robot Inventor, Stem:Bit

1. Bevezetés

2020 szeptemberében bevezetésre került NAT 2020-hoz illeszkedő kerettantervekben már az általános iskola harmadik osztályában [1] megjelennek a robotika alapfogalmai. Alsó tagozatban általában a népszerű padlórobotokkal kezdik a gyerekek a robotika megismerését, azonban felsőbe átlépve már komolyabb, nagyobb tudással és több programozási lehetőséggel rendelkező robotokat ismerhetnek meg [2]. Ebbe a csoportba tartozik a LEGO Education által fejlesztett LEGO Mindstorms EV3 is, amely éppen a NAT 2020 bevezetésével egy időben került kivételre a forgalomból. Az eszköz oktatásban betöltött szerepéről és felhasználhatóságáról számos cikkben olvashatunk [3][4][5] azonban, ha valaki, most szeretné beszerezni, akkor sajnos alternatívát kell keresnie, mert ezt már nem forgalmazzák az üzletekben.

Ehhez szeretnénk segítséget nyújtani két eszköz bemutatásával. Az első eszköz az EV3 utódja, amelynek a hivatalos neve LEGO Mindstorms Robot Inventor, magyarul pedig a Robot Feltaláló elnevezést kapta, azonban a köznyelvben EV4-ként is hivatkoznak rá. A másik eszköz a Stem:Bit, ami egy LEGO kompatibilis elemeket tartalmazó, BBC Micro:bit eszközzel vezérelhető robot.

Cikkünkben bemutatjuk a változásokat és a feladatok felhasználásához szükséges módosításokat. Tapasztalataink alapján mindkét eszköz használható az oktatás során, a vizsgált feladatok kisebb módosításokkal megoldhatók, így a korábbi verzióhoz készült tananyagok használhatóvá válnak.

2. A LEGO Mindstorms Robot Inventor (51515) bemutatása

A legújabb LEGO Mindstorms eszköz a Robot Inventor (51515) 2020-ban jelent meg, leváltva elődjét, az EV3-t. Az új eszköz papírdobozban érkezett, mely az 1. ábrán látható. A doboz 47,5 cm széles, 37 cm magas és 6,7 cm vastag, melyet felnyitva rengeteg LEGO elemet találunk, valamint egy matricákat tartalmazó kis ívet és egy kis füzetet, amiben megtalálható az elemlista. A doboz kinyitása egyszerűsödött az EV3 otthoni (31313) változatához képest, ugyanis cipősdobozhoz hasonlóan egymásba illeszthetőek a részei, azonban, ennél jobb megoldás volt az EV3 oktatási (45544) verziójának doboza. Ugyanis, az oktatási verzióban volt belső rendszerező, ebben azonban nem találunk ilyet.

Ajánlott korcsoport	10 éves kortól
Programozási nyelv típusa:	blokkos/Python
Építőelemek száma	949
Motorok száma	4
Érzékelők típusa	távolság-, fény- és színérzékelő, iránytű, giroszkóp
Egyéb tartozékok	-
Jelenlegi ár	129 990 Ft ¹

1. táblázat: A LEGO Mindstorms Robot Inventor 51515 általános ismertetője

1. ábra: A LEGO Mindstorms Robot Inventor (51515) készlet doboza².

A dobozban összesen 949 elem található, ami összehasonlításképpen, az otthoni verzió elemszámának majdnem 1,5-szerese, míg az oktatási verzióknak a kétszerese. Az elemek között megtaláljuk az EV3 készletekben található elemek jelentős részét, újdonság viszont a több nagyméretű elem a dobozban, köztük egy tábla szerű elem³, amely alkalmas arra, hogy rá, illetve bele illesszünk érzékelőket, motorokat és egyéb elemeket. (A legjobban a BBC Micro:bittel használható próbapanelkekhez lehetne hasonlítani az új LEGO táblát.) Az EV3-ban található fém golyót kivették a készletből, ami autószerű felépítéseknél okoz(hat) problémákat. Az elemek megjelenése modernizálódott, színesebb lett, emiatt inkább a LEGO Education SPIKE™ Prime (45678) szetthez hasonlítható.

2.1. Modellek

A fejlesztők 5 modellt terveztek a készlethez, melyek építési útmutatóit a programozáshoz használt alkalmazásban találhatjuk meg. (Erről a későbbiekben írunk [részletesebben](#).) A modellek az 1. ábra bal oldalán is láthatóak. Neveik balról jobbra a következők: M.V.P, GELO, BLAST, CHARLIE és TRICKY. A modellek nagy részéhez tartozik módosítási lehetőség építési útmutatóval. Például, a TRICKY alapmodellt át lehet úgy építeni, hogy tudjon kosárlabdázni (egy emelő karral mozgatja a

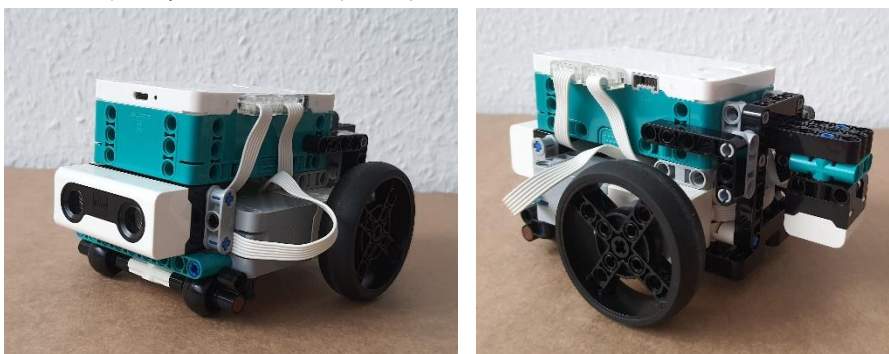
¹ Forrás: <https://www.lego.com/hu-hu/product/robot-inventor-51515>

² Forrás: <https://www.lego.com/hu-hu/product/robot-inventor-51515>

³ Megtekinthető a következő honlapon: <https://www.brickowl.com/catalog/lego-dark-turquoise-panel-11-x-19-39369>

labdát), focizni (lábszerű karokkal elrúg egy labdát), bowlingozni (forgatva kilő egy labdát) vagy rajzolni (felemelhető toll állványt lehet rátenni).

Az eszköz teszteléséhez többféle modellt is kipróbáltunk, azonban oktatási célra a készítőik által ajánlott TRICKY felépítés tűnt a legpraktikusabbnak (2. ábra). A modellt kis mértékben módosítottuk, a díszítő elemeket nem helyeztük fel az esetleges későbbi módosítások miatt, ami hosszabb távon jó döntésnek bizonyult. Az alapmodellbe az útmutató szerint csak a távolságérzékelő kerül beépítésre, azonban mi szerettünk volna minél több érzékelőt kipróbálni, így a robot hátsó részére illesztettük a színérzékelőt, majd a nyomásérzékelőt (3. ábra).



2. ábra: A TRICKY nevű modell látható a képen, a ráadásként felszerelt színérzékelővel⁴.

2.2. Motorok és érzékelők

A motorok számát tekintve növekedés történt, ugyanis négy egyforma méretű, úgymond nagy motort kaptunk a korábbi két nagy és egy közepeshez képest. Ezen kívül sokkal kisebbek lettek és több illesztő felülettel rendelkeznek, mint a korábbi Mindstorms motorok. Ezzel a megjelenéssel és működéssel a SPIKE szettben találkozhattunk korábban más színben. A motorok rendszere is frissült, ugyanis időről-időre az alkalmazás szól, hogy megjelent egy új verzió és frissíteni kellene őket, amivel korábban nem találkozhattunk.

Az érzékelők jelentősen megváltoztak a megjelenésükben, ezek is a SPIKE szett elemeire hasonlítanak, mint a motorok. A csomagban egy ultrahangos távolságérzékelőt találunk, mint az EV3 oktatási verziójában, azonban ez kisebb hatótávolsággal rendelkezik, csupán 200 cm-re lát el. Az eszköz pontosságát tekintve tapasztalataim alapján javulás fedezhető fel. További extraként elhelyeztek az érzékelő körül négy darab lámpát, amelyeket külön-külön programozhatunk.

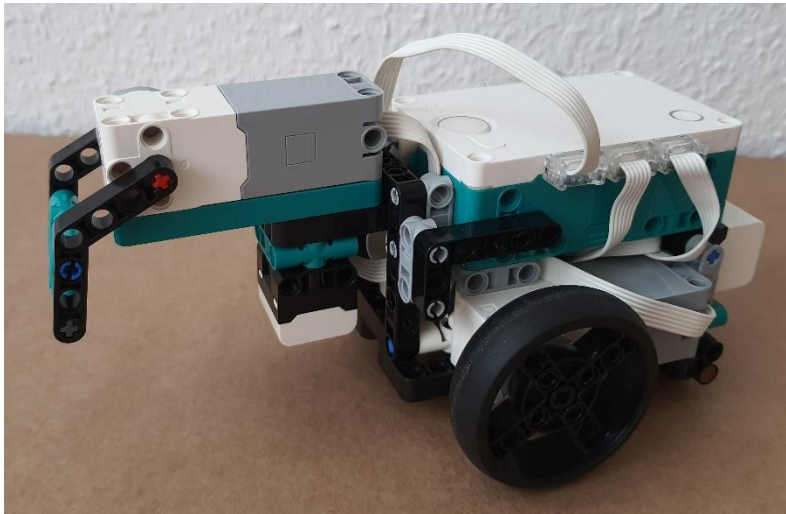
A fény- és színérzékelő nyolc színt és a semleges színt tudja megkülönböztetni. A korábbi listából kivették a barnát, helyette bekerült a rózsaszín, és kiegészítették a szett cián kék színével. Tehát a következő színeket érzékeli az eszköz: fekete, rózsaszín, kék, cián, zöld, citromsárga, piros, fehér és semleges. Az EV3-ban megismert fényvisszaverődés itt is használható, sőt, mindkét változata (*reflected light* és *ambient light*) elérhető a programozó felületen. Az érzékelő ezeken kívül képes megmérni, hogy az általa látott szín milyen mértékben tartalmaz pirosat, kéket vagy zöldet. Tehát, le tudnánk kérdezni egy szín RGB kódját, amit a színérzékelős feladatokhoz fel tudnánk használni. Így tudnánk olyan pályát vagy nyomkövető színes csíkokat nyomtatni, amit nagyobb valószínűséggel érzékel az eszköz, mint korábbi változata. Azonban a színérzékelő továbbra is gyenge pontja az eszköznek, ugyanis a saját LEGO elemeiből sem ismeri fel minden esetben a zöldet, a kéket és az új ciánt, melyek az EV3-nál is problémás színek voltak. Bizonyos szögben, bizonyos magasságban nagyobb az esély a sikerre

⁴ Forrás: Solymos Dóra saját fotói

ezeknél a színeknél is, azonban, ha biztosra szeretnénk menni, inkább maradjunk a fekete, fehér, piros vagy sárga színeknél.

Az iránytű és a giroszkóp beépítésre került a Hub nevet viselő vezérlőbe. Ezeknek köszönhetően lehet például erőteljes érintésre megjeleníteni egy-egy smiley-t a Hubon, vagy a BBC Micro:bithez hasonlóan a különböző irányú mozgásokra beállítani különböző funkciókat.

Az EV3-hoz képest hiányérzetünk lehet a szenzorokat tekintve, ugyanis a szettben nincs nyomásérzékelő. Azonban a SPIKE-ban van, ami kompatibilis ezzel a szettel, így ha be tudjuk szerezni, akkor könnyedén tudjuk használni. Ha szeretnénk valamivel helyettesíteni, akkor a meglévő elemeinkből kreativitásunkat felhasználva építhetünk valamit, ami elfogadható egy nyomásérzékelőnek. Az interneten⁵ a legtöbb motor felhasználásával készítenek nyomásérzékelőt, mi is ebből inspirálódtunk és a 3. ábrán látható érzékelőt készítettük el, ami a színérzékelő felett kapott helyet a tesztautón. Ez az összeállítás működőképes, helyettesítheti a nyomásérzékelőt, de pontosságát tekintve gyenge.



3. ábra: A tesztautó nyomásérzékelővel kiegészített változata⁶.

2.3. A vezérlő, vagyis a Hub

Megjelenésében jelentős változást tapasztalhatunk az elődhez képest. Az EV3-on megtalálható kijelző eltűnt, így számos eddig megszokott műveletet nem vagy máshol tudunk elérni. A Hubon egy nagy bekapcsoló, két kisebb programozható, és egy bluetooth gombot találunk. A bekapcsoló gomb körül lévő LED lámpa programozható, mint elődjénél, csak a színek megváltoztak és nőtt a számuk. A kijelző, mint rajzoló felület bizonyos szempontból egyszerűsödött és inkább a Micro:bit ledjeihez hasonlóan vált programozhatóvá, tehát koordinátánként tudjuk őket kapcsolni és állítani a fényerősségüket. Továbbá, meg tudunk jeleníteni rajta animációkat és a képernyőnél hosszabb szövegeket, amiket fényszalaghoz hasonló módon jelenít meg. Ezen kívül még hang lejátszására is alkalmas.

A vezérlőn hat darab port található, amelyek az eszköz hosszanti oldalán helyezkednek el egyenlő arányban és A–F betűkkel azonosíthatók. A portok között nincs különbség, bármelyikbe csatlakoztatható motor és érzékelő is. Ezen kívül még találunk egy micro USB csatlakozási lehetőséget is a

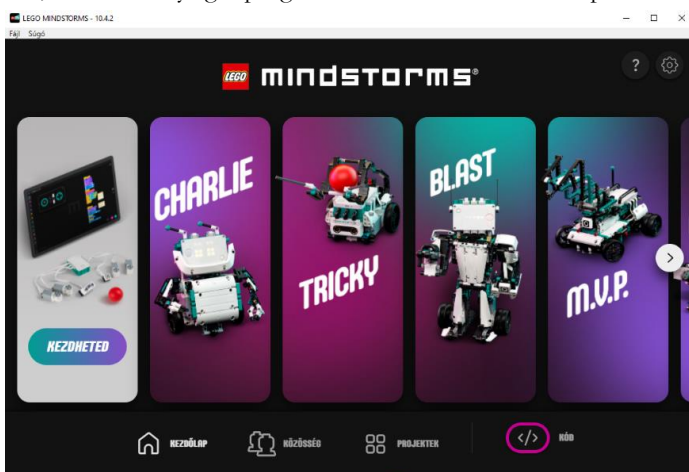
⁵ <https://www.eurobricks.com/forum/index.php?/forums/topic/183443-fix-no-force-sensortouch-sensor-in-51515-robot-inventor/>

⁶ Forrás: Solymos Dóra saját fotója

Hubon. Ezen keresztül, a csomagban található micro USB – USB kábel segítségével csatlakozhatunk a számítógéphez. A kábel nemcsak töltésre alkalmas, hanem adatátvitelre is.

2.4. A hivatalos programozói felület bemutatása

A Robot Inventor programozásához egy alkalmazást kell letöltenünk, ami elérhető a Microsoft Store-ban, a Play Áruházban és az App Store-ban is. Az alkalmazás neve magyarul LEGO® MINDSTORMS® Feltaláló, angolul pedig LEGO® MINDSTORMS® Robot Inventor. Több nyelv közül lehet választani, de először a telepített eszköz alapértelmezett nyelvén fog elindulni az alkalmazás. Az app megnyitásakor a 4. ábrán látható felület fog megjelenni, ami az app kezdőlapja. A *Kezddbeted* gombot választva az eszköz használatához kapunk segítséget az alapoktól indulva. A modellek képeire kattintva pedig elérhetjük az építési útmutatókat, majd az építés után, kapunk egy kódot is, amivel kipróbálhatjuk a felépített robotunkat. A *Közösség* lapra kattintva elérjük a felhasználók által készített modellek építési útmutatóit. A felület hátránya, hogy csak akkor tudjuk megnézni bármelyik útmutatót, ha letöltjük az összeset. A főmenü harmadik gombja a *Projektek*, amivel a korábbi kódjainkat tudjuk előkeresni. Itt egy helyen találjuk meg a blokkos és a Pythonban megírt kódjainkat is. A kezdőlap utolsó gombja a *Kód*, amivel a tényleges programozói felületre tudunk átlépn.



4. ábra: A programozói felület kezdőlapja⁷.

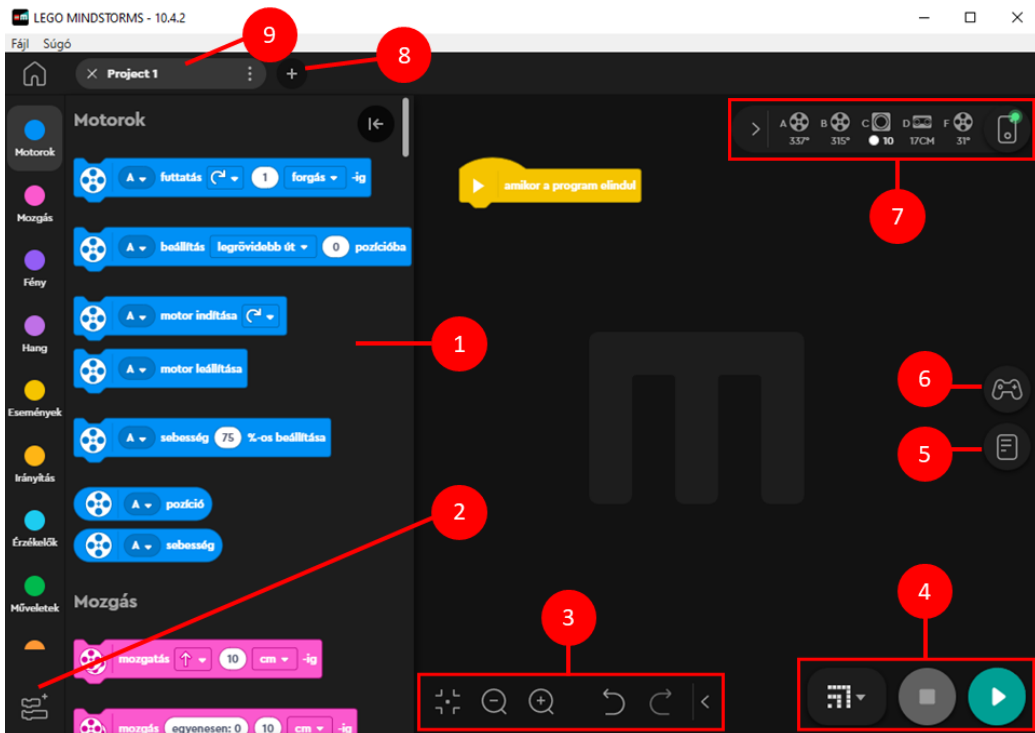
Az EV3-t a hivatalos szoftverjében ikon alapú blokkokkal tudtuk programozni, ahol vízszintesen kellett elhelyezni egymás mellett a blokkokat. Később MicroPythonban is lehetett programozni, de szöveg alapú blokkokkal is programozhattuk a Makecode felületén, ahol függőlegesen kellett egymásba illeszteni a blokkokat és egy szimulátor is rendelkezésünkre állt. Az új Mindstorms blokkos felületén szöveg alapú blokkokat használhatunk, amiket függőlegesen kell egymásba illeszteni. A környezet maga Scratch alapú, így a kinézet és néhány blokk ismerős lehet. Ezen kívül programozhatjuk az eszközt Python nyelven is a MicroPython segítségével. Ehhez készítettek egy dokumentációt is, amely angol nyelven itt érhető el: <https://lego.github.io/MINDSTORMS-Robot-Inventor-hub-API/index.html>

A programozói felület (5. ábra) letisztult, csak a programozáshoz szükséges gombok láthatóak, azonban ezek közül is elrejtethetünk néhányat, ha éppen nem kellenek. A felület részei a következők:

⁷ Forrás: Solymos Dóra saját fotója

1. Blokkpaletta
2. Bővítmények hozzáadása
3. Nagyítás, visszavonás és ismétlés
4. Letöltés/Streaming, leállítás és lejátszás
5. Súgó és figyelő
6. Távirányító
7. Hub kapcsolat
8. Új projekt létrehozása
9. Aktuális projekt

A felületen elrejtethető az 1. számmal jelölt blokkpaletta, a 3. számú nagyítás, visszavonás és ismétlés, valamint a 7. számú Hub kapcsolat.



5. ábra: A blokkprogramozáshoz használható felület kezdőlapja⁸

A Bővítmények hozzáadása gombra kattintva felugrik egy ablak, ahol számos bővítmény közül választhatunk, melyek a következők: modellblokkok, több motor, több mozgás, zene, időjárás, LEGO powered up, további érzékelők. Ezen kívül kísérleti státuszban vannak, de ettől függetlenül kipróbálhatók a következő bővítmények: gépi tanulás, XBOX ONE controller, DUALSHOCK 4 controller és Hub-Hub kommunikáció.

Az 5. ábrán 4-es számmal jelölt gombokkal van lehetőségünk a program letöltésére és bluetooth kapcsolaton keresztül futtatására. Az utóbbi mód a *Streaming* mód kiválasztásával érhető el, amivel

⁸ Forrás: Solymos Dóra saját fotója

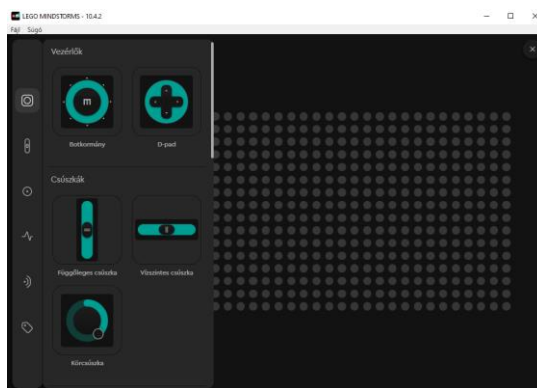
letöltés nélkül, azonnal futtatjuk a programunkat. A letöltési mód annyiban változott a korábbi verzióhoz képest, hogy itt maximalizálták a letölthető programok számát, mégpedig 19-ben, és a letöltés előtt kiválaszthatjuk, hogy hanyadik helyre szeretnénk letölteni a programot. A letöltés után rögtön futtatja a Hub a programot, azonban, ha az eszköztől szeretnénk elindítani azt, akkor ki kell keresnünk a sorszámát a Hubon a balra-jobbra gombokkal és a bekapcsoló gombbal kell elindítanunk.

Az EV3-nál magunk adhattunk nevet a programnak, és az alapján kellett megkeresnünk az eddigi összes letöltött fájl között. Az EV3-nál a fájlok karbantartása nehézkes volt, hiszen egyesével kellett törölni a fájlokat, azonban itt nagymértékben leegyszerűsödött. Az alkalmazásból könnyedén, egy gomb megnyomására törölhetjük a szükségtelen fájlokat.

A Robot Inventoron lévő két módot úgy tudjuk megkülönböztetni az eszközön, hogy más-más mintában kapcsolódnak fel a lámpák, ami a letöltésnél a letöltési hely sorszámát, míg a streamnél egy wifi jel.

A felületen (5. ábra 4-es számú gombcsoport) a mód választó gomb mellett találjuk a Leállítás és a Lejátszás gombokat. Ezek az első programok elkészítésénél nem tűnnek annyira fontosnak, azonban kicsit több próbálkozás után már elengedhetetlenek. Ugyanis az eszközön nincs olyan gomb, amivel le lehetne állítani az aktuálisan futó programot. (Ez az EV3-nál a képernyő bal alsó sarkában elhelyezkedő szürke gomb volt.) A program a blokkok végrehajtása után nem lép ki a programból, mint elődje, hanem tovább fut, még akkor is, ha már egyébként nincs mit tennie. Ekkor a Lejátszás gombra kattintva újra el tudjuk indítani a programot, míg a Leállítás gombbal tudjuk befejezni a futtatást. Másik módszer a programból való kilépésre az, hogy betesszük a kódunk végére az *állj és kilépés a programból* nevű narancssárga blokkot. Ekkor nincs szükség arra, hogy a Leállítás gombbal befejezzük a futtatást.

Az 5. ábrán a 6-os számú, controller alakú gombbal, egy kis táblát tudunk előhívni (6. ábra), amire különböző irányításhoz kapcsolódó gombokat, csúszkákat, érzékelőket helyezhetünk el, majd formázhatjuk őket és nevet adhatunk nekik. Ezt követően be is programozhatjuk őket és a futtatás során távirányítóként használhatjuk a felületet. Ezzel a funkcióval készíthetünk egy olyan programot, amely az EV3-hoz volt elérhető alkalmazás formájában és az eszköz telefonról történő irányítására lehetett használni.



6. ábra: A távirányító készítésére szolgáló felület⁹

A felületen található ságó egyfajta tudástárként is szolgál, hiszen a blokkok leírásán kívül videókat, útmutatókat és kódrészleteket is kapunk az egyes funkciók használatához. Azonban, a magyar felületen a ságó nem minden esetben érthető, ugyanis a fordítás néhol értelmetlen vagy nem azt adja vissza, mint például az angol nyelvű változata.

⁹ Forrás: Solymos Dóra saját fotója

A 7-es számú, Hub kapcsolat nevű gombra kattintva tudunk csatlakozni az eszközhöz. Csatlakozás után itt tudjuk módosítani az eszköz nevét, kezelni tudjuk a Hubon lévő programokat, megváltoztathatjuk a motorok és szenzorok módját, illetve frissíthetjük a motorokat. Ezen kívül itt nézhetjük meg a beépített szenzorok aktuális értékeit is. A programozás során a Hub gombja mellett egy kis sávban láthatjuk, hogy melyik portba milyen eszköz van csatlakoztatva, sőt az aktuális értéket is kiírja a szenzoroknak. (A motoroknál megjeleníthető az aktuális elfordulási szög vagy sebesség.)

2.4.1. Blokkcsoportok bemutatása

A felület maga könnyen használható, különösen, ha már van blokkos környezetben tapasztalatunk, azonban az EV3-hoz képest bizonyos feladatokat máshogy kell megoldanunk. A motornál külön blokkban kapott helyet szinte minden olyan beállítás, ami eddig egy blokkban volt, például a motorok nevei, sebessége, irányítás módja (*degree, rotation, seconds*). A *Motorok* nevű (kék) blokkoknál minden egyes bloknál beállíthatjuk a használni kívánt motort, azonban a sebességet, irányt és a mozgás hosszát már külön blokkokban kell megadnunk. Az EV3-ban *rotation*nek nevezett módot magyarra forgásként fordították le, ami azért nem szerencsés, mert közvetlenül alatta találjuk a fok (korábbi *degree*) módot, amivel a gyerekek könnyen összekeverhetik.

Ha egy blokkban szeretnénk beállítani a motor mozgásának hosszát és a sebességét, akkor a bővítmények közül aktiválnunk kell a *Több motor* címűt, majd a blokkcsoportok legalján fogjuk megtalálni az erre alkalmas blokkokat. Ezek között találhatunk olyan blokkokat, amivel a motort elindíthatjuk általunk megadott értékű sebességgel vagy teljesítménnyel. A két blokk ugyanúgy néz ki, csupán az utolsó kifejezésben különböznek, a működésükben viszont nem találtunk különbséget. Ez a jelenség fennáll a *Több mozgás* című bővítménynél is.

A *Mozgás* című blokkok között a korábbi, két motort egyszerre működtető blokkokhoz hasonló blokkokat találunk. Azonban, itt csak az EV3-nál megismert *Move Steering* blokkhoz hasonló, irány-meghatározással működő blokkokat láthatjuk. Ha különböző sebességgel szeretnénk elindítani a két motorunkat, akkor ahhoz aktiválnunk kell a *Több mozgás* című bővítményt. Itt találhatunk elég összetett blokkokat is, hiszen egy-egy blokkban akár 2-3 értéket is módosíthatunk.

A *Fény* (lila) blokkok között találjuk a Hub „képernyőjét” használó blokkokat. Újdonság, hogy lehet animációkat lejátszani, amihez rendelkezésünkre áll egy könyvtár, ahol a modellekhez készült animációkat találjuk. Azonban készíthetünk mi magunk is animációt. Ezen kívül megjeleníthetünk egy-egy képet is a kijelzőn, ehhez viszont nem áll rendelkezésünkre könyvtár. Továbbá, megjeleníthetünk szöveget is, amiben előrelépés történt, hiszen nem csak a képernyő méretével megegyező hosszúságú szöveget lehet megjeleníteni, ugyanis a szöveget fényszalagként jeleníti meg az eszköz. Ez a funkció a *micro:bit*-nél megismert szöveg megjelenítésre hasonlít, mint ahogy a koordinátánkénti pont megjelenítés is. Ezt itt a *képpontok beállítása 1-en, 1-on 100%-ra* blokkal tudjuk elérni. Ezen kívül, ebbe a csoportba került a bekapcsoló gomb körüli ledek programozására szolgáló blokk is, ill. az újdonságként megjelenő, a távolságérzékelő körüli lámpákat állító blokk is. Itt kapott helyet az eszköz tájolását beállító két blokk is.

A *Hang* blokkcsoportban az EV3-ból megismert blokkokkal találkozhatunk. Újdonságként jelent meg, hogy hangot nemcsak a Hubon tudunk lejátszani, hanem a programozó eszközön is. Emiatt a hangokat két csoportra osztották. A *Hangok a Hubon* című részben azok a hangok szerepelnek, amiket a Hubon tudunk lejátszani, míg a *Könyvtár* részben azok, amiket a programozó eszközön. Új funkcióként szerkeszthetjük a könyvtárban szereplő hangokat és egyszerűbbé vált a hangfelvétel készítés is. A zongora hangok lejátszására szolgáló blokkok nevei megváltoztak, angolul a *play beep* kezdetű blokkokkal, míg magyarul a *sípszó lejátszása* kezdetű blokkokkal tudunk ilyen hangokat lejátszani. Az EV3-ban ezt a *Play Note* mód választásával tudtuk megtenni.

Az *Események* csoportban találhatunk olyan indító blokkokat, amelyekben egy-egy érzékelő hatására, a Hub mozgatására vagy gombnyomásra kezdhetjük a programunkat. Ezen kívül, megtaláljuk a Scratchből ismert futtatást indító blokkokat is.

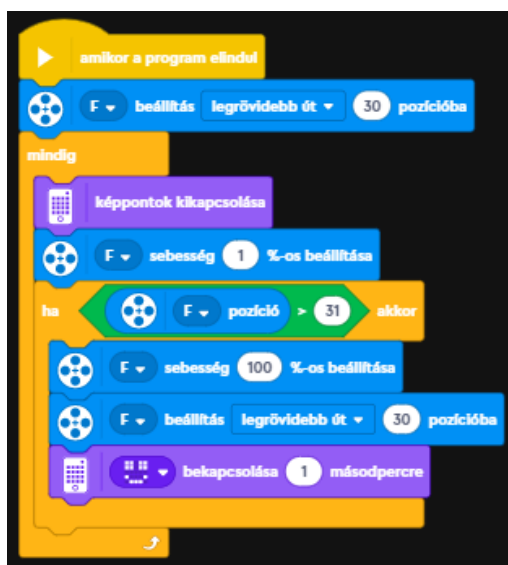
Az *Érzékelők* című blokkok között találhatunk a színek érzékeléséhez szükséges blokkokat, ill. a fényvisszaverődéshez is, azonban ebben a részben csak a korábbi *reflected light* módban működő blokkokat érjük el, amit *tükröző fényként* fordítottak le. Ha az *ambient light* módot szeretnénk használni, akkor a *További érzékelők* csoport aktiválása után, a *Környezeti fény* című blokkokkal tudjuk azt megtenni. Az érzékelők között új funkcióként jelent meg a gesztusok érzékelése és a tájolás, ami szintén a Micro:bit-hez hasonló módon működik. A *További érzékelők* között találhatjuk a vörös, kék és zöld színek lekérdezésére szolgáló blokkot, ami a színértéket adja eredményül egy 0 és 255 közötti számként. Továbbá, itt találjuk a Spike szetthez tartozó nyomásérzékelő blokkjait is.

2.4.2. Milyen feladatokat oldhatunk meg a Robot Inventorral?

A Robot Inventorhoz angolul megjelent néhány könyv, azonban magyar tananyag még nem érhető el, így az EV3-hoz készült tananyagok közül a [6]-ben szereplő feladatok közül válogattunk néhányat, amit igyekeztünk a Robot Inventor blokkos környezetében megoldani. A tesztünk során a feladatok jelentős részét meg lehetett oldani az eszközzel, azonban bizonyos esetekben csak a feladatok átfogalmazásával tudtuk ezt elérni.

A színes ledet használatánál olyan problémába ütköztünk, hogy a ledet nem tudjuk kikapcsolni, ill. nem tartozik villogó funkció a blokkhoz, így az ehhez kapcsolódó feladatokat módosítanunk kellett. A kikapcsolást a fekete szín beállításával oldottuk meg, míg a villogást ciklus és várakozó blokkok kombinációjával.

Az érintésérzékelő hiánya miatt az ehhez kapcsolódó feladatokat egy motorral kellett megoldanunk. Ez a csere, a feladatok teljes átfogalmazását igényelte és néhány kihagyását. Ugyanis, így a *pressed*, *released* és *bumped* funkciók tesztelése értelmetlenné vált, hiszen nem egy gombunk van, hanem egy karunk, ami a kis változásokra nagyon érzékeny, viszont, ha el akarjuk fordítani, akkor azt csak kézzel tudjuk megtenni. Tesztünk alapján, ha túl gyorsan megy neki egy tárgynak a robot, akkor nem fordul el a nyomásérzékelő motorja, de az eszköz felborul. Ha lassan megy a robot, akkor érintésérzékelőként használható, viszont ekkor minimális mozgás is elég az érzékeléshez. Ez látható a 7. ábrán látható kódban.



7. **ábra:** A következő feladatot megvalósító program kódja látható a képen: az érintésérzékelőként működő motor ha „benyomódik”, akkor mosolygó fej jelenik meg a kijelzőn¹⁰.

A tananyagban szereplő párhuzamos program készítésével kapcsolatos feladatokat ([6] 27. oldal 1–4. feladatok) meg lehet oldani ebben a környezetben is, azonban nem lesznek párhuzamos programok, hanem csak egyszálúak. Ezen kívül, nem lehet megoldani vagy csak a feladat jelentős mértékű egyszerűsítésével a házikó kirajzolásával kapcsolatos feladatot ([6] 45. oldal 5. feladat). A kijelző 5×5 pixeles mérete eléggé lekorlátozza a megjeleníthető képeket. Továbbá, a vonalkövetési feladatot ([6] 33. oldal 3. feladat) a szín- és fényérzékelő pontatlansága miatt egy matematikai képlet megadásával tudjuk csak megoldani. A képletről és annak működéséről ebből a videóból lehet többet megtudni: <https://youtu.be/Z8Cv60f75JJ>.

A hangok között nem szerepelnek a színek nevei, azonban mivel elég egyszerűvé vált a hangfelvétel készítése, valamint akár telefonról is programozható az eszköz, így a gyerekek felvehetik magyarul a színek neveit a telefonjukkal közvetlenül az appban, amiket fel is használhatnak a feladat megoldásánál. Így picit hangszerkészítést is tanulhatnak, továbbá, saját nyelvükön tesztelhetik a *Hangos színek* című feladatot ([6] 36. oldal 3. feladat) működését, ami németül tanuló gyerekek esetében előnyt jelenthet.

Az említésre nem kerülő feladatoknál különösebb problémát nem tapasztaltunk, a rendelkezésünkre álló blokkokkal meg tudtuk oldani őket.

3. A Stem:Bit bemutatása

3.1. Az eszköz specifikációja [7][8]

Ajánlott korcsoport	5. osztálytól felfele
Programozási nyelv típusa:	blokk alapú/Python/JavaScript/Scratch
Építőelemek száma	302
Motorok száma	2
Érzékelők típusa	akadályérzékelő, vonalkövető, ultrahang csatlakoztatási lehetőség
Egyéb tartozékok	rezgőmotor, 2×10 mm RGB LED, $3 \times$ kicsi RGB LED, infravörös vevő egység, távirányító
Ár (micro:bittel)	117,95€/csomag

2. táblázat: A Stem:Bit általános ismertetője

¹⁰ Forrás: Solymos Dóra saját fotója



8. ábra: A Stem:Bit csomag¹¹

A Stem:bit csomag két fő részre bontható, az egyik az építőelemek, a másik pedig maga a kiterjesztést szolgáló expansion board. Ehhez tudjuk csatlakoztatni a micro:bitet, amely vezérelni fogja a robotunkat. A táblázatban (2. táblázat) látható egyéb tartozékok és érzékelők mind integrálva vannak erre a boardra, amely az építés során esetlegesen korlátozhatja a kreativitást. A csomag építőelemei teljes mértékben kompatibilisek a LEGO termékekkel. Ugyan az EV3 csomagja csak öt hivatalos építési lehetőséget tartalmaz, de kapunk hitelesített felhasználói útmutatókat is. A tananyagok tekintetében is az EV3 javára billen a mérleg, ugyanis sajnos nem érhető el kidolgozott tananyag a Stem:Bithez.

A micro:bit általi vezérlés lehetővé teszi azt, hogy egy széles életkori spektrumon használható eszköz legyen a Stem:Bit. Emiatt az eszköz rendelkezik minden olyan előnyös tulajdonsággal is, amely a micro:bitre jellemző, mint például a költséghatékonyság, a széleskörben történt alkalmazás és sikeresség, valamint a kiterjeszthetőség elég magas volumene[9]. A micro:bitről számos leírás és cikk elérhető már, ezért a cikkben külön nem foglalkoztunk azzal, hogy mint vezérlőegységet bemutassuk azt.

3.2. Modellek



9. ábra: Az IR Car modellje

¹¹ Forrás: <https://shop.sb-components.co.uk/products/stem-bit-the-programmable-blocks-kit-for-micro-bit?variant=31064155455571>

A csomagból kilenc hivatalos modellt van lehetőség megépíteni, amelyhez a kézikönyv segítséget is nyújt az építési útmutatók által. Ezen modellek megépítése során nem ütköztünk nehézségekbe. Fontos megjegyezni, hogy vannak csak olyan modellek, amelyek mozgásra nem képesek. A leginkább EV3 feladatok megoldására képes modell az IR Car (9. ábra¹²). A nyomvonallal és objektumok elkerülésével kapcsolatos feladatok terén hasonlóan szerepel a két robot, azonban fontos figyelembe venni, hogy nem tudunk egy darab Stem:Bittel olyan robotot építeni, amely mozog és képes objektumokat is mozgatni, tehát ez egy nagy hátrány lehet. Az akadályok érzékelése és elkerülése terén mindkét robot hasonlóan teljesít akárcsak a vonalkövetésben. A WRO verseny szabályzata miatt versenyzésre ez a csomag nem alkalmas, hiszen csak és kizárólag LEGO robotokkal lehet ebben részt venni[10].

Lehetőségünk van két lánctalpas jármű elkészítésére is, amelyek jó alapok lehetnek a terepi mérőjárművek készítésére, amit tudunk távirányítóval is működtetni. A micro:bit nyújtotta komplex szenzorcsomag pedig lehetőséget ad arra, hogy a digitális kultúra órákon kívül, komolyabban számításba lehessen venni az eszközt a természettudományok oktatásában is.

A fentebb említett eszközökön túl, tudunk egy darut, egy csúzlít, egy tuk-tuk kocsit, egy terepen mászni képes hatlábú robotot, egy úgynevezett robot autót (vonalkövetésre nem alkalmas), és egy tárgyak megfogására alkalmas eszközt építeni a csomagból.

3.3. Motorok és érzékelők

A Stem:Bit csomagban összesen két darab motort kapunk. Nem csak a darabszámban marad el a csomag kvalitása az EV3-hoz képest, de a motorok fajtáját tekintve itt nem szervomotorokat kapunk. Emiatt a technográfia padlón való megvalósításánál nehézségekbe ütközhetünk, mivel így csak a motorok működésének idejét és sebességét adhatjuk meg, utóbbit pedig nagyban befolyásolni fogja az akkumulátor töltöttsége. A kiterjeszhetősége szempontjából azonban három helyet kapunk szervomotorok számára, így maximálisan 5 motort tudunk csatlakoztatni az extension boardra.

Szenzorok terén egy rendkívül sokrétű eszközzel beszélhetünk. Ennek egyik oka az, hogy a micro:bit által nyújtott szenzorkészletet is használhatjuk a munka folyamán, így kapunk: gyorsulásmérőt, iránytűt, fényérzékelőt, hőmérőt, mikrofont és érintő szenzort (utóbbi kettőt csak v2-es micro:bit esetén). Ezekon felül pedig a board is el van látva egy infravörös akadályérzékelő és egy vonalkövető szenzorral, valamint lehetőség van ultrahang szenzor csatlakoztatására is. Az akadály elkerülő és a vonalkövető a tesztek során viszonylag kevés alkalommal hibázott, azt lehet mondani, hogy a szenzorok működése megfelelő.

A színérzékelő hiánya miatt szintén nem fog tudni minden EV3 által végzett feladatot megvalósítani, például nem lesz képes az eszköz a színeket megkülönböztetni. Ezen felül van egy infravörös vevőegység is az eszközön, ami lehetővé teszi, hogy irányítsuk a robotot a csomaghoz kapott távirányító segítségével is. Kiterjeszhetőség szempontjából egy jóval komplexebb eszközt kapunk, ha a szenzorokat nézzük. A csomag építőszámát tekintve alulmarad a LEGO termékekhez képest, de a kompatibilitás miatt, itt is lehetőség van a bővítésre.

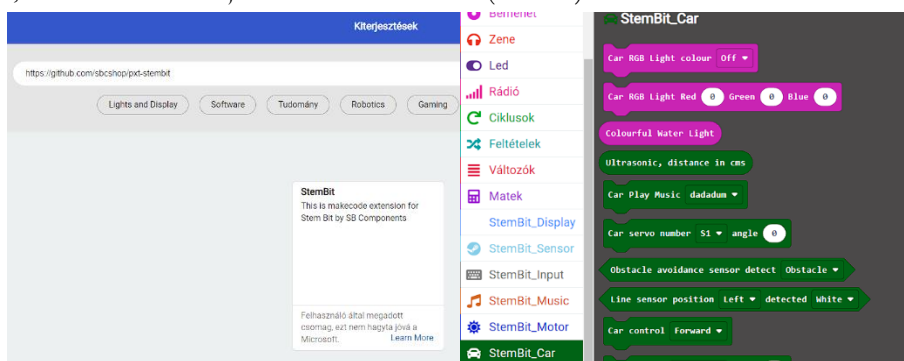
Amit még érdemes megemlíteni és hasznos lehet a gyerekek számára egy ilyen foglalkozáson, az a kimeneti perifériák. Itt kapunk egy rezgőmotort, valamint 3 kicsi RGB led izzót, amit programozhatunk, valamint további két 10 mm-es ledet, amely a board elején helyezkedik el, mintha egy autónak a fényszórója lenne. Amennyiben a micro:bit v2-es verzióját használjuk, úgy egy hangszóróval is bővíthetjük a kimeneti eszközök tárházát.

¹² Forrás: <https://shop.sb-components.co.uk/products/stem-bit-the-programmable-blocks-kit-for-micro-bit?variant=31064155455571>

3.4. A hivatalos programozói felület bemutatása

A micro:bit elterjedése miatt, a hivatalos felület részletes bemutatása helyett, az adott eszközhöz kapcsolódó új blokkok és azok integrálására szeretnénk fókuszálni a makecode felületén. A blokk alapú nyelv tökéletes lehet arra, hogy bármilyen életkorban használjuk az eszközt és azzal bevezessük a programozást vagy nagyobb motivációt gyakoroljunk a diákokra. A blokkprogramozás hazánkban már a harmadik osztályban megjelenik [1], de van olyan nemzetközi kutatás is, amely bizonyítja, hogy a blokkprogramozásnak számos előnye van akár középiskolában is [11], hiszen egy könnyen olvasható, egyszerű programozási megoldásról beszélhetünk, így remek lehetőséget biztosít a programozás megszerettetéséhez, a motiváció növelésére. A board irányításáért felelős blokkok nem csak felhasználásra elérhetőek, hanem akár lehetőségünk van azok szerkesztésére is, hiszen a Stem:Bithez tartozó programkódok és források elérhetőek egy GitHub repositoryban¹³. A blokkok elnevezései egyértelműek és minimális angol nyelvtudással könnyen megérthető, hogy az egyes blokkokat mire is fogjuk tudni hasznosítani.

Az új blokkokat a kiterjesztésekre kattintva tudjuk elérni. Itt szükséges a repository linkjének beillesztése, ami után már ki tudjuk választani az eszközt (13. ábra).



10. ábra: A kiterjesztés csatolása és az új blokkok¹⁴

A kiterjesztés hozzáadása után megkapjuk a kódokat, amelyek angol nyelvűek, ezért kisgyermekek esetében ez nehézséget okozhat és szükség lehet tanári magyarázatra. A blokkok felépítésükben nem térnek el a már megszokott micro:bit környezettől, ezért pozitívum lehet az, hogy egy ismerős környezettel találják magukat szemben azok a gyerekek, akik már korábban programoztak micro:bitet vagy használtak blokkos nyelveket.

A felületről egy rövid ismertető megtalálható a csomaggal érkező kézikönyvben, a már korábban említett kilenc építési útmutatón túl, valamint 1-1 képet a szükséges kódsorról, amellyel működésre lehet bírni az eszközt, illetve az útmutatók elején egy listát leírással azokról a blokkokról, amelyet használni fogunk.

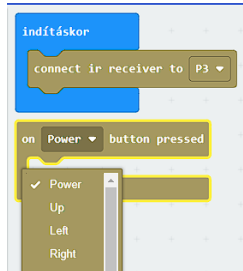
A korábban említett távirányítós vezérléshez is szükségünk van a kiterjesztésre és ahogy az Inventor esetében, itt is nekünk kell a különböző funkciókat beprogramoznunk. Erről a következő fejezetben lesz szó bővebben.

¹³ <https://github.com/sbcshop/pxt-stembit>

¹⁴ Forrás: Gaál Bence saját képe

3.4.1. Blokkcsoportok bemutatása¹⁵

Ebben az alfejezetben főként azokra a blokkokra fogjuk a hangsúlyt fektetni, amelyek nem igényelnek egyéb kiegészítőt és a csomagban lévő eszközöket tudjuk velük irányítani. Ilyen blokkcsoportok a távirányítóhoz és az „autó” vezérlő blokkok.

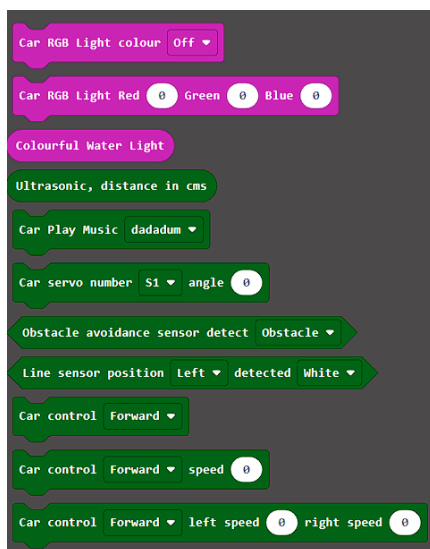


11. ábra: A távirányítót vezérlő blokkok

Az első bemutatni kívánt blokkcsoportunk a távirányítóhoz köthető. Itt mindösszesen kettő új blokk szerepel, amik közül az egyikkel gyakorlatilag aktiváljuk a vevőegységet. Tudni kell, hogy a vevő egység a P3-as pinhez van kötve, ezért nekünk is azt kell beállítani a programozás folyamán. A másik blokknál pedig lehetőségünk van megadni, hogy melyik gomb megnyomásakor mi történjen. Ennek működési elve hasonló a micro:biten lévő gombok vezérléséhez, tehát adott gomb lenyomásakor a blokk belsejében lévő kód fog lefutni (11. ábra).

A *StemBit_Car* blokkcsoport már jóval több elemből épül fel, amelyeket három alcsoportra oszthatunk. Az első a képzeletbeli autónk fényszóróit hivatott vezérelni, ezek bíborvörös színnel vannak jelölve. Ezekkel előre beállított, vagy tetszésünk szerint létrehozott (rgb kód által) színeket tudunk megjeleníteni az eszköz két nagy ledjén. A második csoportba az érzékelőket soroltuk, ugyanis itt van lehetőségünk lekérdezni azt, hogy van-e objektum az akadályérzékelő szenzor előtt, hogy a vonalkövető érzékel-e vagy sem fehér/fekete vonalat, illetve az ultrahangos távolságmérő értékét centiméterben. A harmadik egységgel vezérelhetjük a motorok forgását, akár külön külön is, így lehetőségünk van a kanyarodásra. Itt kapott még helyet egy előre beépített zenét lejátszó blokk, valamint lehetőség nyílik itt is állítani azt, hogy hány fokot forduljanak a servomotorok, amit az ultrahangos érzékelőhöz hasonlóan külsőleg kell csatlakoztatni az eszközhöz (12. ábra).

¹⁵ Képek forrása: Gaál Bence saját képei



12. ábra: A motorvezérlő és érzékelő blokkok

A *StemBit_Music* blokkban tudjuk bekapcsolni a rezgőmotort, míg a *StemBit_Motor* blokkok segítségével a külsőleg csatlakoztatott servomotorok sebeségét, illetve azok fordulati fokát állíthatjuk be. Az *input* blokkok külsőleg csatlakoztatott eszközök kezelésére, míg a *sensor* blokkcsoport az eszközön található infra mellett, a mikrofon és az ultrahang szenzor kezeléseire alkalmas. A *neopixel* blokkok segítségével a boardon található 3 ledet tudjuk működtetésre bírni. Erre és a nagyobb ledet vezérlésre szintén lehetőségünk van a display blokkok használatával. Ezen blokkok tanórai használatát nem feltétlen javasoljuk a programozás bevezetésénél. Ennek egyik oka, hogy ezek a blokkok komplexitásukat nézve nem feltétlen ajánlottak kezdők számára, illetve az adott eszköz ki és bekapcsolása helyett a pineket vezérelve tudjuk a különböző beállításokat elvégezni.

3.4.2. Milyen feladatokat oldhatunk meg a Stem:Bittel?

A vonalkövető feladatok alapvetően megvalósíthatók az eszközzel, akár csak az objektumok kikerüléséhez kapcsolódó problémák is megoldhatók vele. A párhuzamos programok, pedig az Inventorhoz hasonlóan oldhatók meg ebben a környezetben. A feladatok megvalósításában a szenzorok fajtái, illetve a motorok típusai korlátozzák az eszközt, ahogy ezt a korábbi fejezetekben kifejtettük. Szintén felmerült itt és a kijelző mérete így a rajzolós feladatok egyszerűsítésére szolgálnak, hiszen a micro:biten is csak 5x5-ös led panel található.

Versenykörnyezetben történő használatban pedig nem lehet összehasonlítani sem a korábban tárgyalt Inventorral, sem pedig az EV3 robottal, mivel a WRO Robo Mission és Robo Sports versenyeken csak és kizárólag a LEGO termékeit használhatják a résztvevők [10].

Főbb felhasználására inkább az osztálytermi órákat javasoljuk, ahol a programozás érdekesebb tételében, illetve a micro:bit kibővítésében tudja az eszköz igazán megállni a helyét. A blokkos környezetnek hála pedig akár már kisiskoláskorban is használható, ahogy a tesztelés folyamán mi is hasznosítottuk ezt a tulajdonságát, negyedik osztályban mint figyelemfelkeltő és motivációt erősítő eszköz használtuk, míg ötödik osztályban az informatikaóra mellett a természettudomány tantárgyban játszott szerepet mint modellező eszköz.

Számos olyan érdekes projektet lehet vele megvalósítani, ami nem csak a digitális kultúra órán történő használatra korlátozódik. Lánctalpas kialakítása miatt hatásos lehet egy kutatórobot modellezésére, amely méréseket végez, majd adatokat továbbít egy másik eszközre. A micro:bitnek segítségével

akár olyan járműveket is építhetünk, amelyeket egy kézre erősített másik micro:bittel kommunikálva irányíthatóvá válnak a kézmozdulataink által.

Az eszköz bővíthetőségének komplexitása előnyös lehet a nagyobb korosztályok számára, hiszen a szenzorcsatlakozások és azok pinekkel történő vezérlése az elektronikai és villamosági területekhez kapcsolódó készségeket is erősíteni tudja. Ennek hátránya lehet az, hogy időigényesebb a bővítés, mint egy moduláris robot esetében.

4. Összegzés

Az EV3 kivezetése miatt szükséges, hogy alternatívák után nézzünk. Nyilvánvalóan az egyes robotkészletek felépítése és ára nagyban befolyásolja a választást, de érdemes lehet egyéb szempontok szerint is átgondolni, hogy melyiket is akarjuk használni. A fentebb taglalt két robotsomag specifikációit (3. táblázat) nézve egy jó lehetőség lehet az EV3 kiváltására, de egyik sem tudja teljeskörűen helyettesíteni azt, így vegyük figyelembe az egyéb tényezőket is.

Szerettünk volna a közismert és már bejáratott LEGO márka mellett egy kevésbé ismert olcsóbb alternatívát is nyújtani. Az oktatásban való hasznosítása mindkét eszköznek nagyban függ attól, hogy milyen céllal is vesszük be azt a tanterembe. Ugyan a LEGO Robot Inventor egy drágább lehetőséget biztosít, de mindenféleképpen egy jobban támogatott eszközzel beszélhetünk és a márkának megfelelően egy magasminőségű eszközt kapunk. Hátránya, hogy a megvalósítások nem feltétlenül tükrözik azt, hogy ez az eszköz az EV3 utódja lenne. Erősen érződik, hogy a tervezés folyamán nagy behatást gyakorolt az eszközre a Spike termékcsalád, amely a közeljövőben teljes mértékben átveszi a szerepet a LEGO oktatási robot szortimentjében és a gyártót csak erre a termékre fog fókuszálni.

A táblázatból látszik, hogy a Stem:Bit fele annyi építőelemből áll, azonban szenzoros kiterjeszthetőségét nézve sokkal nagyobb szabadságot biztosít. A kevesebb motorszám és a kevesebb építőelem akadályt jelenthet a komplexebb robotok építésében, és mint korábban említettük, gátat szabhat a kreatív alkotásban. A robot ára viszont kevesebb, mint a fele az Inventornak és az oktatásban ez is egy meghatározó tényező.

Ennek ellenére úgy gondoljuk, hogy az osztálytermi feladatokhoz, a programozás megszerettetésére egy kifejezetten jó eszközt kapunk. Érdekes, látványos mérőeszközöket, távirányítású autókat hozhatunk velük létre és jó megoldás lehet különböző interdiszciplináris STEM projektek költséghatékony megvalósítására.

Összeségében tehát, mindkét robot kifejezetten jó oktatási célokra, csak azon belül más-más területet vettek vele célba. Az Inventor egy nagyobb támogatottsággal és szélesebb körben ismert eszköz, amely alkalmazható akár nemzetközi versenyeken is, míg az olcsóbb alternatívájú Stem:Bit a sokoldalú micro:bitet kiaknázva nyújt teljes élményt minamellét, hogy mindkét eszköz lehetőséget biztosít a tanmenetek által előírt robotikaoktatás lefedésére.

	Robot Inventor (51515)	Stem:Bit	EV3 Home (31313)	EV3 Education (45544)
motorok száma	4	2	2 + 1	2 + 1
motorcsatlakoztatósi lehetőségek (portok) száma	6	2 + 3 szervó	4	4
szenzorok száma	2 (beépített) + 2 (moduláris)	6 (beépített)	3 (moduláris)	5 (moduláris)
szenzorcsatlakoztatási lehetőség	6	beépített+4	4	4
vezérlőegység	Hub	micro:bit	Brick/Tégla	Brick/Tégla
áramellátás	akkumulátor	akkumulátor	6 db AA elem	akkumulátor
elemszám összesen	949	307	601	541
ár	(359,99 € ¹⁶)	117,95 € ¹⁷	legutolsó ismert ár: ¹⁸ (570 €)	legutolsó ismert ár: ¹⁹ (745 €)

3. táblázat: A specifikációk összehasonlítása.

Irodalom

1. Kerettanterv az általános iskola 1–4. évfolyama számára- Digitális kultúra 3–4. évfolyam https://www.oktatas.hu/pub_bin/dload/kozoktat/kerettanterv/Digitalis_kultura_A.docx (utoljára megtekintve: 2022.10.20.)
2. Kerettanterv az általános iskola 5–8. évfolyama számára- Digitális kultúra 5–8. évfolyam https://www.oktatas.hu/pub_bin/dload/kozoktat/kerettanterv/Digitalis_kultura_F.docx (utoljára megtekintve: 2022.10.20.)
3. Çam, E. & Kuyucu, M. (2022). The impact of robotics assisted programming education on academic success, problem solving skills and motivation . Journal of Educational Technology and Online Learning , 5 (1) , 47-65 . DOI: 10.31681/jetol.1028825
4. Bradley S. Barker & John Ansoorge (2007) Robotics as Means to Increase Achievement Scores in an Informal Learning Environment, Journal of Research on Technology in Education, 39:3, 229-243, DOI: 10.1080/15391523.2007.10782481
5. Solymos Dóra, „LEGO robotok felhasználási lehetőségei az oktatásban,” *InfoDidact 2019*, pp. 265-275, 2019.
6. Barbalics Dóra Krisztina, Solymos Dóra, Lego Mindstorms EV3 robotok programozása, ELTE Informatikai Kar, Budapest, 2018. Elérhető: http://tet.inf.elte.hu/tetkucko/wp-content/uploads/2018/12/legomindstorms_szakkoranyag.pdf (utoljára megtekintve: 2022.11.5.)

¹⁶ Forrás: <https://www.lego.com/en-de/product/robot-inventor-51515>

¹⁷ Hazai forgalmazótól nem kapható, ezért csak euróban tudtuk megadni az árát.

¹⁸ Forrás: <https://kockashop.hu/lego/mindstorms-ev3-31313>

¹⁹ Forrás: <https://kockashop.hu/lego/ev3-lego-mindstorms-education-bazis-szett-oktatasi-robotkeszlet-45544>

7. *A Stem:Bit hivatalos értékesítőjének honlapja*
<https://shop.sb-components.co.uk/products/stem-bit-the-programmable-blocks-kit-for-micro-bit?variant=31064155455571> (utoljára megtekintve: 2022.10.20.)
8. *A Stem:Bit kickstarter honlapja*
<https://www.kickstarter.com/projects/sb-gajendra/stem-bit/description> (utoljára megtekintve: 2022.10.20.)
9. Gaál, B. (2022). Robotics-Enhanced Natural Science in Primary Schools. In: Bollin, A., Futschek, G. (eds) Informatics in Schools. A Step Beyond Digital Education. ISSEP 2022. Lecture Notes in Computer Science, vol 13488. Springer, Cham. https://doi.org/10.1007/978-3-031-15851-3_8 (utoljára megtekintve: 2022.11.08.)
10. General Rules WRO-2022. <https://wro-association.org/competition/2022-season/> (utoljára megtekintve: 2022.11.08.)
11. Weintrop, David & Wilensky, Uri. (2015). To Block or not to Block, That is the Question: Students' Perceptions of Blocks-based Programming. <https://doi.org/10.1145/2771839.2771860> (utoljára megtekintve: 2022.11.08.)