- 2. A program legkártékonyabb hibájára csak akkor derül fény, ha a programot már legalább hat hónapig használták.
- 3. Azokat a kontrollkártyákat, amelyeket semmiképpen nem szabad helytelen sorrendben tárolni, helytelen sorrendben tárolják.
- 4. A tetszőleges sorrendben tárolható lyukszalagokat nem tárolják helytelen sorrendben.
- 5. Ha az inputellenőrzést úgy tervzték meg, hogy kiszűrje a téves inputokat, egy leleményes idióta biztosan kidolgozza azt a módszert, melynek segítségével a téves inputok megkerülik az ellenőrzést.
- 6. A programozók legjobban a káromkodás nyelvét ismerik.

Gilb megbízhatatlansági törvényei:

- 1. A számítógép megbízhatatlan, de az ember méginkább.
- 2. Az emberi megbízhatóságra alapozott rendszerek megbízhatatlanok.
- 3. A felderíthetetlen hibák végtelenül változatosak, szemben a felderíthető hibákkal, amelyeknek száma a dolog természetéből következöen: korlátozott.
- 4. A megbízhatóság fokozására eszközölt befektetések addig fokozódnak, amíg túl nem haladják a hibák valószínű költségét, illetve amíg valaki el nem éri, hogy a munka is folyjék.

A számítógépek világának törvényei Golub szerint:

- 1. A homályos célkitűzések azt szolgálják, hogy senkit ne feszélyezzen a költségek előzetes bemérhetősége.
- 2. A hanyagul megtervezett munka a vártnál háromszor több időt vesz igénybe, míg a gondosan megtervezett csak kétszer többet.
- 3. A folyamatmódosításhoz szükséges erőfeszítés az idő előrehaladtával mértani haladványban növekszik.
- 4. A munkacsoportok azért rühellik a heti teljesítménybeszámolókat, mert ezek ékesen bizonyítják, hogy teljesítményől szó sincs.

A számítógép-javítás Smith-féle szabálya: A forrasztónyílások egy mérettel szűkebbek a kelleténél. Folyamánya: A megfelelő méretű forrasztó nyíllások viszont rossz helyen vannak.

Jaruk törvénye: Ha olcsóbb volna új berendezést vásárolni, a vállalat csak azért is a régit fogja javítattni. Folyamánya: Ha a régi berendezés javítása volna olcsóbb, a vállalat csak azért is vadonatújat fog vásárolni.

LEGO robotok

III. rész

III.1.3. Eszközök

A LEGO MINDSTORMS EV3 Home Edition szoftver *Eszközök* (Tools) menüjében számos eszköz található, amelyek extra funkcionalitást és támogatást nyújtanak az EV3-tégla és szoftver használatához.

A *Hangszerkesztő* (Sound Editor) segítségével ki tudjuk alakítani saját hangeffektusainkat, majd használni tudjuk a szerkesztett hangokat a robotunk programozásában.

A *Képszerkesztő* (Image Editor) segítségével képeket, grafikákat tervezhetünk az EV3-tégla kijelzője számára.

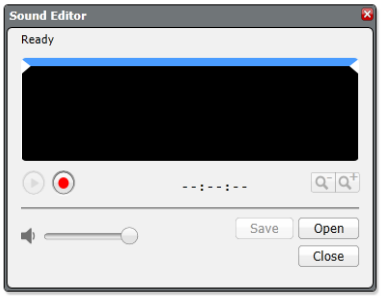
A ceruzával szabálytalan alakzatokat rajzolhatunk, ezen kívül vonalat húzhatunk, köröket, téglalapokat rajzolhatunk, festhetünk, törölhetünk, szöveget írhatunk, vagy kiválaszthatunk részeket a rajzból.

Az alakzatok vonalainak háromféle vastagsága lehet, és kétféle betűtípus közül választhatunk.

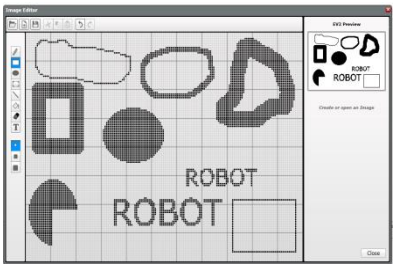
Egy előnézet ablakban megtekinthetjük, hogy az ábránk hogyan fog kinézni az EV3-tégla kijelzőjén.

A *Saját blokk építő* (My Block Builder) segítségével alprogramokat, saját blokkokat hozhatunk létre és szerkeszthetünk. Elnevezhetjük, ikonnal láthatjuk el és hozzárendelhetünk olyan paramétereket, amelyek nekünk fontosak. A saját blokkok automatikusan tárolódnak a Saját blokkok palettán.

Hasznos eszköz a *Firmware frissítő* (Firmware Update). Időnként frissített firmware jelenik meg az EV3-téglához javasolt ezeknek az új verzióknak a telepítése, amint azok elérhetővé válnak.



13. ábra: *Hangszerkesztő*



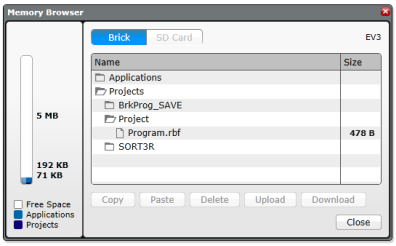
14. ábra: *Képszerkesztő*

A *Vezeték nélküli beállítás* (Wireless Setup) segít abban, hogy vezeték nélküli kapcsolatot állítsunk fel a téglával. Ehhez be kell szereznünk egy Wi-Fi USB adaptert az EV3-téglához, és engedélyeznünk kell a Wi-Fi kommunikációt a téglán.

A *Blokk importálása* (Block Import) menüpont, eszköz segítségével új blokkot adhatunk a Programfejlesztő palettához. Ez lehet egy új LEGO blokk, vagy más gyártók által fejlesztett blokk is, például egy harmadik fél által gyártott érzékelőhöz. Ezeket a blokkokat először le kell töltenünk a számítógépünkre, majd ezt követően importálhatjuk őket a szoftverbe.

A *Letöltés appként* (Download as App) segítségével úgy tölthetjük le programjainkat az EV3-téglára, hogy az a *Tégla appok képernyőn* jelenjen meg az alapértelmezett alkalmazások mellett.

A *Memóriaböngésző* (Memory Browser) áttekintést ad a téglán történő memóriahasználatról (az SD kártyát is beleérve, ha behelyeztünk egyet). Fel lehet használni programok, hang, grafikus és egyéb fájlok áthelyezésére az EV3-téglára, és minden olyan állomány másolására és törlésére, amelyek már a téglán vannak.



15. ábra: *Memóriaböngésző*

A *Tégla program importálása* (Import Brick Program) eszköz lehetővé teszi, hogy az EV3-tégla *Tégla programozás app*-ban készült

programot beimportálhassuk az EV3 szoftverbe. Programunkat így tovább finomíthatjuk a LEGO MINDSTORMS EV3 Home Edition szoftver teljes funkcionalitásának felhasználásával.

III.1.4. Program blokkok

A LEGO robotok programozásához *program blokk*okat (Program Blocks) használunk. Ezek a programfejlesztői vászon (Programming Canvas) alatti palettán (Programming Palettes) vannak elhelyezve.

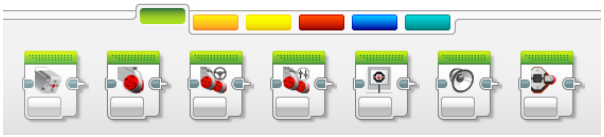
A blokkokat típus és jelleg alapján kategóriákba sorolták, így könnyebben megtalálhatjuk a szükséges blokkot.

A megfelelő palettáról kiválasztott blokkot az egér segítségével a programfejlesztői vászonra húzzuk, összekötjük őket a megfelelő más blokkokkal, beállítjuk a bemeneti adatait, és máris futtatható programot kapunk.

A palettán a blokkok csoportjait színek jelölik: zöld, narancs, sárga, vörös, kék, türkiz. A különböző csoportok a következő blokkokat tartalmazzák:

Zöld – Cselekvő blokkok (Action Blocks):

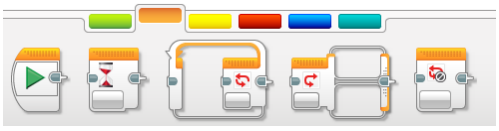
- Közepes motor (Medium Motor)
- Nagy motor (Large Motor)
- Kormányozás – mozgásvezérlés (Move Steering)
- Tank – mozgástank (Move Tank)
- Kijelző (Display)
- Hang (Sound)
- Tégla állapotjelző fény (Brick Status Light)



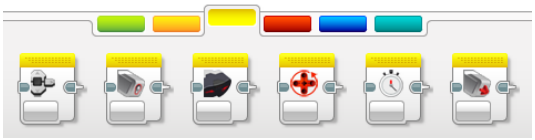
16. ábra: *Cselekvő blokkok*

Narancs – Folyamat blokkok (Flow Blocks)

- Start (Start)
- Várj (Wait)
- Hurok – ciklus (Loop)
- Kapcsoló – elágazás (Switch)
- Hurok, ciklus megszakítás (Loop Interrupt)



17. ábra: *Folyamat blokkok*



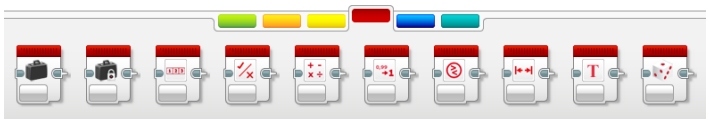
18. ábra: *Érzékelő blokkok*

Sárga – Érzékelő blokkok (Sensor Blocks)

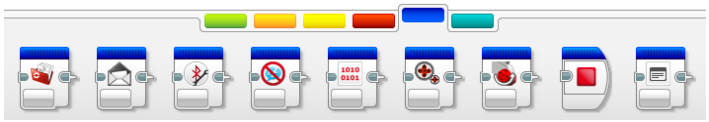
- Tégla gombok (Brick Buttons)
- Színérzékelő (Color Sensor)
- Infravörös érzékelő (Infrared Sensor)
- Motor forgás (Motor Rotation)
- Időzítő (Timer)
- Érintés érzékelő (Touch Sensor)

Vörös – Adatblokkok (Data Blocks)

- Változó (Variable)
- Állandó (Constant)
- Műveletek tömbökkel (Array Operations)
- Logikai műveletek (Logic Operations)
- Matematika (Math)
- Kerekítés (Round)
- Összehasonlítás (Compare)
- Tartomány (Range)
- Szöveg (Text)
- Véletlenszerű (Random)



19. ábra: *Adatblokkok*



20. ábra: *Speciális blokkok*

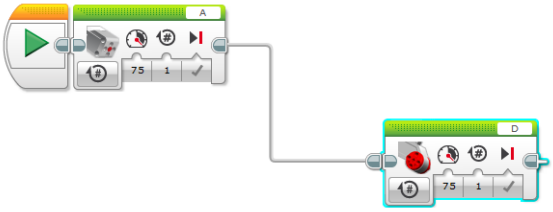
Kék – Speciális blokkok (Advanced Blocks)

- File Access (Fájlhozzáférés)
- Messaging (Üzenetek)
- Bluetooth Connection (Kapcsolat)
- Keep Awake (Virrasztás)
- Raw Sensor Value (Nyers érzékelő érték)

- Unregulated Motor (Szabályozatlan motor)
- Invert Motor (Motor invertálás)
- Stop Program (Program leállítás)
- Megjegyzés (Comment)

Türkiz – Saját blokkok (My Blocks)

- Kezdetben ez a paletta üres. Ha egy program valamilyen részletét sok más programban fel szeretnénk használni, akkor létrehozhatunk egy saját blokkot. Ez olyan, mint az eljárás vagy függvény imperatív nyelvek esetén. A létrehozott saját blokkok erre a palettára kerülnek, azután ezeket egyszerűen beszúrhatjuk a későbbi programjainkba, ugyanazon a projekten belül.

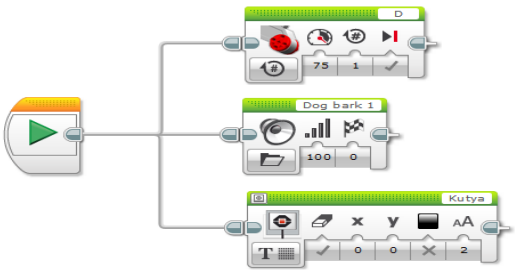


21. ábra: *Összeragasztott és összekötött blokkok*

A LEGO robotok programozása úgy történik, hogy a programfejlesztői vászonra előbb felteszünk egy Start blokkot a Folyamat blokkok palettáról, majd a kívánt program létrehozása érdekében a többi blokkot. Ha egy blokkot megfogunk az egérrel és azt a palettáról a vászonra húzzuk, közel egy már meglévő blokkhoz, akkor a két ellentétes oldalon lévő fülecskék révén ezek egymáshoz ragadnak, és a második blokk az első programbeli folytatása lesz. Így egymás mellé több blokkot is feltehetünk, amikor futtatjuk a programot, a blokkok egymásután kapják meg a vezérlést úgy, ahogy a vászonra fel voltak helyezve, balról jobbra. Ez a végrehajtási sorrend.

Ha a blokkot valamivel távolabb helyezzük el az előző blokktól, akkor ezeken nem ragadnak össze, az összekötést a programozó kell megoldja úgy, hogy az első blokk jobboldali fülecskéjéből egy drótot húz ki az egérrel, és ezt a drótot a második blokk baloldali fülecskéjével összeköti.

Az összekötő drótokat egyszerűen letörölhetjük úgy, hogy a drót jobb oldali fülecskéjére kattintunk az egérrel.



22. ábra: *Párhuzamos blokkok*

Az egér segítségével kiválaszthatunk egy adott blokkot (vagy a SHIFT gomb lenyomásával egyszerre többet is), ekkor a blokk körül egy világoskék keret jelenik meg. A kiválasztott blokkot áthelyezhetjük, vagy akár le is törölhetjük.

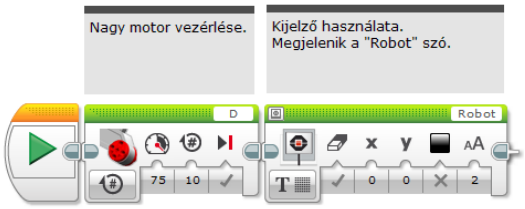
A folyamatvezérlő blokkok (ciklus, elágazás) négy sarkában négy kis köröcske, az oldalak közepén pedig négy kis négyzet jelenik meg, ezek segítségével tetszőlegesen át tudjuk méretezni a blokkot.

A robotok programozása párhuzamosan is történhet. Ha egy blokk után két vagy több blokkot teszünk fel egymás alá, és ezeket az elsővel úgy kötjük össze, hogy az elsőből kihúzott drótból ágazik el a vezérlés, ezek az utóbbi blokkok párhuzamosan fognak végrehajtódni.

A 22. ábrán látható példában egy nagy motort vezérlünk, közben a hangfalon egy kutyaugatást játszunk le, valamint a kijelzőre kiírjuk a „Kutya” feliratot. E három egymás alatti blokk párhuzamosan fog végrehajtódni.

A párhuzamos programozásnál vigyázzunk az erőforrások megfelelő használatára, hisz könnyű értelmetlen parancsokat kiadni a robotnak! Például, ha párhuzamosra állított két blokk segítségével ugyanazt a motort próbáljuk irányítani úgy, hogy az egyik blokkon 10-szer jobbra forgatjuk, a másik párhuzamos blokkon pedig 10-szer balra forgatjuk, akkor nyilvánvaló, hogy a motor működtetésében komoly konfliktushelyzet áll elő.

A vászont, s így a rajta lévő blokkokat nagyíthatjuk, kicsinyíthetjük a vászon fölötti eszközsáv jobb oldali gombjaival, mozgathatjuk ezeket a kéz ikonú gomb segítségével, illetve megjegyzéseket is írhatunk a vászonra. A megjegyzések szövegdobozát tetszőlegesen át lehet méretezni. Ezek nagyon hasznosak lehetnek a program működésének leírására, megértésére.



23. ábra: *Megjegyzések*

A programozás során használhatjuk a LEGO MINDSTORMS EV3 Home Edition szoftver gyorsbillentyűit is. Ezeket a gyorsbillentyűket a 9. táblázat foglalja össze.

Windows	**Mac**	**Eredmény**
CTRL+A	Command-A	Mindent kiválaszt
CTRL+B	Command-B	Leállítja az EV3-at
CTRL+C	Command-C	Másolás
CTRL+D	Command-D	Letöltés az EV3-ra
CTRL+H	Command-H	Kontextus függő súgó
CTRL+F	Command-F	Képernyő lementése

Windows	Mac	Eredmény
CTRL+I	Command-I	EV3 memória navigátor
CTRL+M	Command-M	Hardver oldal ki/be-kapcsolása
CTRL+N	Command-N	Új program
CTRL+E	Command-E	Új kísérlet
CTRL+O	Command-O	Megnyitás
CTRL+P	Command-P	Nyomtatás
CTRL+Q	Command-Q	Kilépés
CTRL+R	Command-R	Letöltés és futtatás
CTRL+S	Command-S	Mentés
CTRL+Shift+S	Command-Shift-S	Mentés másként
CTRL+T	Command-T	Előrejelzés
CTRL+U	Command-U	Letöltés az EV3-ról
CTRL+V	Command-V	Beillesztés
CTRL+W	Command-W	Fül bezárása
CTRL+Shift+W	Command-Shift-W	Projekt bezárása
CTRL+X	Command-X	Kivágás
CTRL+Y	Command-Y	Helyrehoz
CTRL+Z	Command-Z	Visszavonás
CTRL+G	Command-G	Eszközök közötti váltás
CTRL+Shift+H	Command-Shift-H	Tevékenység elrejtése/megjelenítése
CTRL+Shift+P	Command-Shift-P	Pont-elemzés
CTRL+Shift+A	Command-Shift-A	Szekció-elemzés
F1	Command-Option-?	Súgó
1	1	Cselekvő paletta
2	2	Folyamat paletta
3	3	Érzékelő paletta
4	4	Adat paletta
5	5	Speciális paletta
6	6	Saját blokkok paletta
Nyíl (balra)	Left arrow	Balra vivés
Nyíl (jobbra)	Right arrow	Jobbra vivés
Alt+Drag	Alt-húzás	A program mozgatása, átméretezése
CTRL+J	Command-J	Új megjegyzés

9. táblázat: *Gyorsbillentyűk*

III.1.5. Adattípusok

A LEGO MINDSTORMS EV3 Home Edition grafikus programozási nyelv adattípusokat használ a blokkok adatainak ábrázolásához. Ezek a következők:

- Numerikus (Numeric)
- Logikai (Logic)
- Szöveg (Text)
- Numerikus tömb (Numeric Array)
- Logikai tömb (Logic Array)

A *numerikus* adattípus negatív vagy pozitív egész, illetve valós számokat fed. Például: –2, –1.54, 0, 56, 516.2356.

A *logikai* típus egy Igaz (True) vagy Hamis (False) értéket ábrázol.

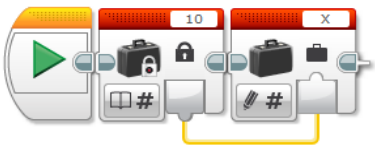
24. ábra: *A Simple Text karakterei*

A *szöveg* típus egy karaktersorozatot (karakterláncot) ábrázol. Az egyes karakterek az úgynevezett *egyszerűsített szöveg* (Simple Text) karakterei lehetnek. Ezeket a 24. ábrán mutatjuk be. Más karaktereket nem tud ábrázolni az EV3-tégla. A karakterek segítségével angol, orosz, egyszerűsített kínai, illetve japán szövegeket tud megjeleníteni, ábrázolni. Természetesen a szöveg szóközöket is tartalmazhat, így nemcsak szavakat, hanem mondatokat is képezhetünk. Például: „Udvozollek a Sapientian!”

A *numerikus tömb* egy negatív vagy pozitív egész, illetve valós számokból álló listát jelent. A listának meghatározott hossza van, amelynek csak az EV3-tégla memóriája szab határt. A lista minden egyes eleme egy numerikus érték, amelyet a megadott sorrendben tárol a rendszer. Mivel a lista nem halmaz, ezért egy érték többször is szerepelhet benne. A lista elemeit pontosvesszővel („;”) választjuk el egymástól, és az egész listát szögletes zárójelek közé tesszük („[]”). Például: [0; –0.25; 345.25; 7; 7; 7]. Az üres tömböt []-el jelöljük, ennek a hossza: 0.

A *logikai tömb* a numerikus tömbhöz hasonló adattípus, azzal a különbséggel, hogy ennek az elemei csak az Igaz (True) vagy Hamis (False) értékek lehetnek.

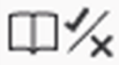
Az adattípusokhoz szorosan kötődnek az *adatdrót*ok (Data Wire), amelyek segítségével a blokkok kimeneteleit köthetjük a bemenetelekhez, így átadva egymásnak a megfelelő adatokat. A blokkok között így interakció jöhet létre, az adatfolyamok révén pedig összetettebb programok valósíthatók meg, így a robot viselkedése is komplexebb lehet.



25. ábra: *Adatdrót*

A 25. ábra egy numerikus adatdrótot ábrázol. Az X változó (bemeneti adat) felveszi a konstans 10-es értékét (kimeneti adat). Ez megfelel az X = 10 értékadásnak imperatív programozási nyelvek esetén. A kimeneti adatot tartalmazó blokk meg kell, hogy előzze a bemenetet tartalmazó blokkot.

A 10. táblázat a különböző típusú adatdrótokat és az adattípusok grafikai szimbólumait mutatja be.

Adattípus	Be	Ki	Drót	Jel írásra (bemenet)	Jel olvasásra (kimenet)
Numerikus					
Logikai					
Szöveg					
Numerikus tömb					
Logikai tömb					

10. táblázat: *Adatdrótok és az adattípusok szimbólumai*

Az adatdrótokat egyszerű „fogd és vidd” (drag and drop) technikával lehet a grafikus felületen kialakítani. Ha az egérrel valamely blokk kimeneti adata fölé megyünk, akkor az egérmutató (kurzor) átvált egy dróttekercset ábrázoló kurzorrá, majd kattintva és megfogva, áthúzhatjuk a drótot egy másik blokk bemenetére.

Egy kimenet egyszerre több blokk bemenete is lehet.

Az adatdrótokat szintén „fogd és vidd” (drag and drop) technikával lehet letörölni: a bemenetről kell lehúzni az adatdrót végét.

A 11. táblázat alapján az egyes adattípusok között automatikus és adatveszteség nélküli konverzió hajtódik végre.

Típusról	Típusra	Eredmény
Logikai	Numerikus	Hamis (False) = 0 Igaz (True) = 1
Logikai	Szöveg	Hamis (False) = „0” Igaz (True) = „1”
Logikai	Logikai tömb	Egy egyelemű tömb
Logikai	Numerikus tömb	Egy egyelemű tömb (0 vagy 1)
Numerikus	Szöveg	A szám szöveges ábrázolása (Például: „23.65”)
Numerikus	Numerikus tömb	Egy egyelemű tömb
Logikai tömb	Numerikus tömb	Ugyanolyan hosszúságú numerikus tömb 0 vagy 1 elemekkel

11. táblázat: *Adatkonverziók*

Ha az EV3-téglát a számítógéphez csatlakoztatjuk (USB, Bluetooth vagy Wi-Fi), és futtatjuk a programot, az aktívan futó blokk valamely drótja fölé húzva az egeret, egy ablakban megjelenik a dróton lévő érték, vagyis a ki/bemeneti adat. Így könnyen nyomon követhetjük és ellenőrizhetjük adatainkat, s ezáltal a teljes program adatfolyamát, működését.

Ha a portokat adatdrót segítségével adjuk meg, akkor ezek numerikus értékként fognak szerepelni a következőképp:

- A port értéke: 1
- B port értéke: 2
- C port értéke: 3
- D port értéke: 4
- 1 port értéke: 1
- 2 port értéke: 2
- 3 port értéke: 3
- 4 port értéke: 4

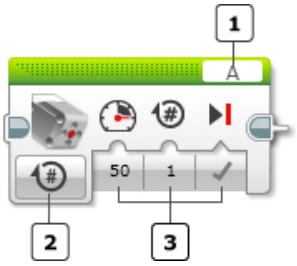
Ha a kormányzási vagy tank üzemmódot választjuk, akkor a két port numerikus értékei:

- B, C: 23
- C, B: 32
- A, B: 12
- A, D: 14

Ha lánckapcsolással kötjük össze az EV3-téglákat, akkor az első tégla portjainak numerikus értékéhez 100-at kell hozzáadni (pl. 101, 102, 103, 104, 123, 132, 112, 114), a második tégla portjainak numerikus értékeihez 200-at, a harmadiknak 300-at, a negyediknek pedig 400-at (pl. 403 a negyedik tégla C portja).

III.1.6. A közepes motor programozása

A közepes motor programozására egy blokk van fenntartva a kezelőfelület zöld eszköztárán. A 26. ábrán látható blokk beállításai a következők:



26. ábra
A közepes motor blokkja

1: Az 1-es gomb segítségével a portot választhatjuk ki (port selector). Ezen a porton keresztül fog kommunikálni az EV3-tégla a motorral, itt küldi át a parancsokat. A port az A, B, C vagy D valamelyike lehet.

2: A 2-es gomb segítségével egy legördülő menüből kiválaszthatjuk a motor működési módját (mode selector): *Off* (kikapcsol), *On* (bekapcsol), *On for Seconds* (bekapcsol időre), *On for Degrees* (bekapcsol fokokra), *On for Rotations* (bekapcsol fordulatszámra).

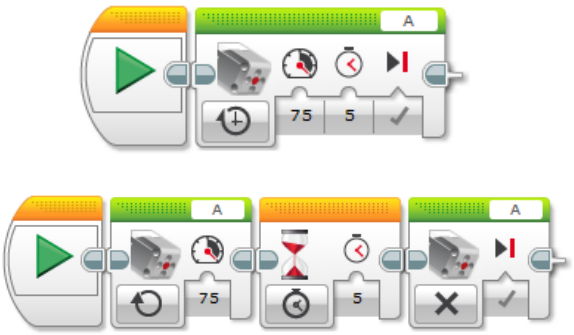
3: A 3-as gomb vagy gombok segítségével a bemeneti adatokat adhatjuk meg. Ezek száma, mértékegysége és mérete a módoktól függ.

Az *On* mód bekapcsolja a motort és ez addig fog működni, míg egy *Off* módú blokk ki nem kapcsolja. A vezérlés a bekapcsolás után azonnal átadódik a következő blokknak. Az egyedüli beállítható bemeneti adat a motor sebessége/ereje, amely egy –100 és

100 közötti érték. Az előjel a forgás irányát jelenti (hátra/előre). A pozitív irány az óramutató járásával megegyező, a negatív irány pedig az óramutató járásával ellentétes.

Az *Off* mód leállít egy előzőleg bekapcsolt motort. Az egyedüli beállítható bemeneti adat a *Brake at End* (leállás a végén). Ez egy logikai érték: *Igaz* (*True*) vagy *Hamis* (*False*) lehet. Az *Igaz* érték azt jelenti, hogy a motor azonnal leáll, és úgy marad abban a pozícióban (*Brake*), a *Hamis* érték pedig azt jelenti, hogy a motor áramellátása kikapcsol, és az addig forog tehetetlenségből szabadon, amíg meg nem áll (*Coast*).

Az *On for Seconds* (bekapcsol időre) mód a megadott ideig működteti a motort. Be lehet állítani a motor erejét, másodpercekben kifejezve (lehet valós szám is) a működés időtartamát, valamint a *Brake at End*-et.



27. ábra: *a) Közepes motor vezérlése egy blokkal; b) Közepes motor vezérlése több blokkal*

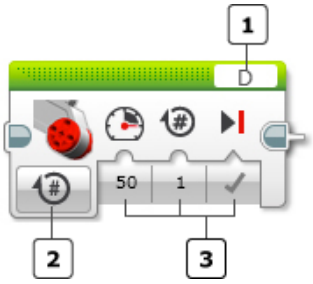
Az *On for Degrees* (bekapcsol fokokra) mód a megadott fokig forgatja a motort. Be lehet állítani a motor erejét, fokokban kifejezve (lehet valós szám is) a fordulatot (egy teljes fordulat 360º), valamint a *Brake at End*-et. A motor belső érzékelője pontosan méri a fokokat, és a motor végrehajtja a megfelelő fordulatot. Ez azt jelenti például, ha működés közben egy akadály nem engedi továbbforogni a motort, ez addig fog várni, míg az akadály elhárul, és pontosan végrehajtódik a megadott foknyi fordulat. Ameddig akadályozva van, a robot programja leáll, a vezérlés nem adódik át a következő blokknak.

Az *On for Rotations* (bekapcsol fordulatszámra) mód a megadott fordulatszámig forgatja a motort. Be lehet állítani a motor erejét, numerikus értékben kifejezve (lehet valós szám is) a fordulatszámot (egy teljes fordulat 1-es, két és fél fordulat: 2,5 stb.), valamint a *Brake at End*-et. Ez a mód teljesen megfelel az *On for Degrees* módnak, annyi különbséggel, hogy más a mértékegység: 1 fordulatszám 360º, 2 fordulatszám 720º, fél fordulatszám 180º, 1,25 fordulatszám 450º stb.

A 27. ábra a) és b) programjai ugyanazt csinálják, csak különféleképpen oldják meg a közepes motor vezérlését. Az a) ábrán egy motorvezérlő blokkot láthatunk, amely *On for Seconds* (bekapcsol időre) módban 5 másodpercig működteti 75%-os erővel az A portra kapcsolt motort mégpedig úgy, hogy azonnal leáll és megtartja a pozícióját. A b) ábrán pedig elindítjuk az A portra kapcsolt motort 75%-os erővel az *On* mód segítségével, majd egy *Wait* (várakozás) blokk segítségével 5 másodpercet várunk (ez idő alatt működik a motor), és végül az *Off* mód segítségével leállítjuk a motort úgy, hogy az megtartja pozícióját.

III.1.7. A nagy motor programozása

Blokk szintjén a nagy motor programozása teljesen megegyezik a közepes motor programozásával. Az ott leírtak érvényesek a nagy motor blokkjára is.

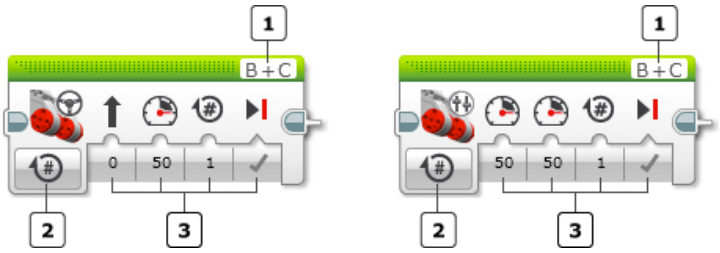


28. ábra: *A nagy motor blokkja*

Mindazonáltal a nagy motorok annyiban különböznek a közepes motortól, hogy ezeket párba is lehet kötni (például a B + C portokat használva). A párban működtetésre két új blokkot használhatunk: a *Move Steering* (kormányozás blokk) illetve a *Move Tank* (tank blokk) blokkokat.

Az 1-es, 2-es gombok használata teljesen megegyezik a közepes motornál bemutatottaknál, a 3-as gombok (bemeneti adatok) változnak csak némiképp, egész pontosan mindkét blokk esetén bejön egy plusz bemeneti adat.

A kormányozás blokk segítségével a robot előre, hátra, fordulásra, vagy megállásra programozható. Beállíthatjuk a kormányozás milyenségét, hogy a robot egyenesen, hajló ívek, vagy szűk fordulások mentén kanyarodjon.



29. ábra: *a) Move Steering; b) Move Tank*

A kormányozás blokk azt feltételezi, hogy a robot két nagy motorral van felszerelve, egyik motor a jármű (robot) bal oldalán, a másik a jobb oldalán. A kormányozás blokk egyszerre vezérli mindkét motort, s így tudjuk vezetni (kanyarítani) a járművet abba az irányba, amit választottunk.

Az irány miatt vigyázzunk mindig, hogy a bal oldali motor portja legyen beállítva először. Például, ha B + C van kiválasztva a portnál, akkor a bal oldali motor legyen a B portra, a jobb oldali motor pedig a C portra kötve.

A kormányozás blokk esetén az első bemeneti adat a kanyarodás mértéke és iránya egy –100 és 100 közötti értékkel megadva, amely százalékban fejezi ki a kanyarodás mértékét: 0 azt jelenti, hogy egyenesen előre halad, 50 azt, hogy 90°-ban kanyarodik jobbra, –50 azt, hogy hogy 90°-ban kanyarodik balra stb.

Figyeljünk arra, hogy az *On for Degrees* (bekapcsol fokokra) vagy az *On for Rotations* (bekapcsol fordulatszámra) mód esetén a robot által megtett távolság most már nemcsak a beállított foktól vagy fordulatszámtól függ, hanem a kanyarodás ívétől is. Ebben az esetben a beállított értékek mindig arra a motorra értendők, amelyik gyorsabban fordul.

A tank blokk nagyon hasonlít a kormányozás blokkhoz, csak azt feltételezi, hogy a robot lánctalpakon jár, így a fordulatot nem a kanyarodás mértékével lehet megadni, hanem a két motor erejével. Ha ez egyik motornak nagyobb az ereje, mint a másiknak, a jármű elfordul. Így a blokkon két erő-beállítási lehetőség (gomb) van, az egyik a bal oldali motor, a másik

pedig a jobb oldali motor erejét állítja be. Például, ha a bal oldali motor ereje 100, a jobb oldalié pedig 50, a jármű egy ívben jobbra fog fordulni. A jármű azonnal megfordul, ha például a bal oldali motor erejét 50-re, a jobboldaliét pedig –50-re állítjuk.

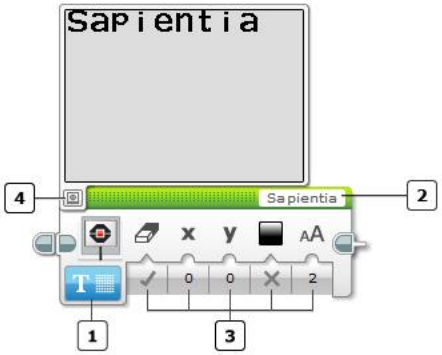
Természetesen a fordulás, kanyarodás íve mindkét esetben függ a kerekek méretétől, vagy a kerekek közötti távolságtól, és más faktoroktól is.

III.1.8. A kijelző programozása

A kijelző programozása a *kijelző blokk* (Display Block) segítségével történik. Így fehér-fekete grafikát vagy szöveget jelentethetünk meg az EV3-tégla kijelzőjén.

A 30. ábrán látható kijelző blokk részei:

- 1. A blokk módjának kiválasztó gombja (mode selector)
- 2. Szövegdoboz
- 3. Bemeneti adatok
- 4. Kijelző megjelenítő gomb

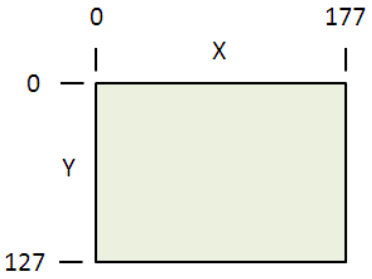


30. ábra: *A kijelző blokk*

Az 1-es gomb segítségével választhatjuk ki, hogy a kijelzőn szöveget (pixeles vagy rácsos), alakzatot (vonal, kör, téglalap, pont) vagy valamilyen beolvasott képet kívánunk megjeleníteni, illetve itt lehet eredeti állapotába visszaállítani a kijelzőt (Reset Screen).

Ha az 1-es gomb segítségével szöveget választunk, akkor a 2-es szövegdobozba írhatjuk be a megjelenítésre szánt szöveget, vagy itt kiválaszthatjuk azt is, hogy a szöveg egy adatdrót segítségével legyen megadva bemenetként (Wired). Ha az 1-es gomb által egy állományt választunk ki, akkor szintén itt, a 2-es szövegdobozban adhatjuk meg az állomány nevét, vagy választhatunk a LEGO által eleve megadott képek közül.

Érdekes a 4-es kijelző megjelenítő gomb. Amennyiben elkészítettük a grafikánkat vagy szövegünket, ezzel a gombbal egy ablakot hívhatunk elő, amely ugyanúgy tartalmazza a megjelenítendő grafikát vagy szöveget, mint az EV3-as tégla kijelzője. Itt tehát előre láthatunk mindent.



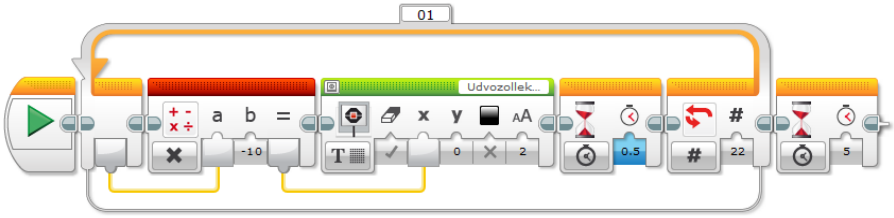
31. ábra: *A kijelző koordinátarendszere*

A 3-as gombok segítségével a bemeneti adatokat adhatjuk meg. Természetesen ezek a kiválasztott módtól függnek.

Ha a pixeles szöveg (Pixel) módot választjuk ki, akkor bemeneti adatként a következőket adhatjuk meg:

- A képernyő letörlése (Clear Screen): logikai érték, Igaz vagy Hamis lehet. Ha Igaz, a szöveg megjelenítése előtt a rendszer letörli a képernyőt.
- X és Y: a szöveg kezdetének (bal felső sarok) koordinátái.
- Szín (Color): logikai érték. Ha Igaz, akkor a szöveg fekete alapon fehérrel jelenik meg, ha Hamis, akkor a szöveg fehér alapon feketével jelenik meg.
- Betűtípus (Font): 0, 1 vagy 2 lehet. A 0 normál (Normal) betűtípust, az 1-es félkövér (Bold), a 2-es nagy (Large) betűket jelent.

Ha a rácsos szöveg (Grid) módot választjuk ki, akkor a fentiek annyiban módosulnak, hogy az X és az Y rácspontokat kell megadni, és a rendszer ezekhez a rácspontokhoz igazítja a szöveget. Így a szöveget sorokban és oszlopokban jeleníthetjük meg. Egy oszlop szélessége megegyezik egy karakter szélességével normál és félkövér betűtípus esetén, ez 8 pixelt jelent. A nagy betűtípus pedig kétszer akkora, mint a félkövér (16×16 pixel). Mivel a normál betűtípus 9 pixel magas, a félkövér pedig 8, a sorok mérete 10 pixel. A kijelzőn tehát 12 (0-tól 11-ig) sor és 22 (0-tól 21-ig) oszlop található.



32. ábra: *A mozgó „Udvozollek a Sapientian!" felirat*

Ha az alakzatok – vonal módot választjuk, akkor bemenetként, a képernyőtörlés és szín mellett megadhatjuk a vonal bal felső és jobb alsó pontjainak a koordinátáit: X1, Y1, X2, Y2. Egy fehér vonal csak akkor látható, ha nem töröljük le a képernyőt, és az előzőleg feketére volt festve.

Az alakzatok – kör esetén a kör középpontjának X és Y koordinátáit, illetve a kör sugarát kell megadjuk, valamint azt is, hogy a kör legyen-e kitöltve vagy sem.

Az alakzatok – téglalap esetén a téglalap bal felső sarkának az X, Y koordinátáit, valamint a téglalap magasságát és szélességét kell megadjuk bemeneti adatként. Ha a kijelzőnek csak egy részét szeretnénk letörölni, akkor arra a területre rajzolhatunk egy fehér téglalapot.

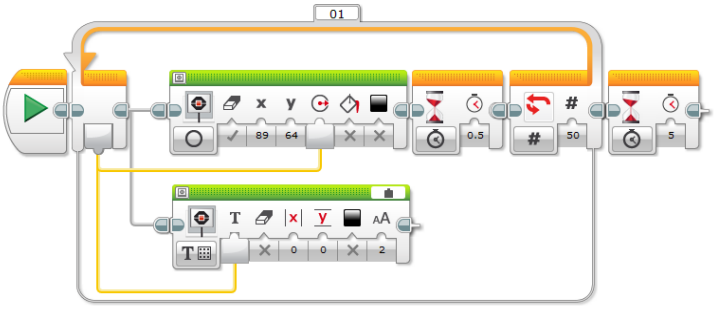
Az alakzatok – pont egy pixelt rajzol ki a kijelzőre a megadott X, Y koordinátákra.

A kép mód segítségével egy fehér-fekete képállományt jelentethetünk meg. A kép bal felső sarkának koordinátái a megadott X, Y pontokban lesznek.

A kijelző visszaállítása mód visszaállítja az EV3-tégla kijelzőjét a normál állapotba, vagyis a program futásáról jelennek meg az információk.

Hogy a megjelenített grafika vagy szöveg ne tűnjön el hamar a képernyőről, a megjelenítés végén használjunk egy várakozás blokkot (Wait).

Ha bármilyen numerikus adatot akarunk megjeleníteni, például egy érzékelő kimeneti értékét, egyszerűen az adatdrót segítségével kössük össze a kimenetet a kijelző blokk szöveges bemenetével. A program automatikusan átalakítja a numerikus értéket szöveggé, és ez megjelenik a kijelzőn.



33. ábra: *Növekvő kör és a sugár megjelenítése*

A 33. ábrán lévő program egy kört rajzol ki a kijelző középpontjába és animál úgy, hogy a sugara növekszik 0-tól 50-ig. A kör kirajzolása előtt letörli a rendszer a képernyőt. Ezzel párhuzamosan a kör sugarát kiírja a kijelző bal felső sarkába (ekkor már nincs képernyőtörlés). Minden körrajzolás után a rendszer fél másodpercet várakozik, a végén pedig 5 másodperc múlva állítja vissza az eredeti képernyőt.

Könyvészet

http://botbench.com/blog/2013/01/08/comparing-the-nxt-and-ev3-bricks/
http://education.lego.com/es-es/products
http://en.wikipedia.org/wiki/ARM9
http://en.wikipedia.org/wiki/Lego_Mindstorms
http://en.wikipedia.org/wiki/Linux_kernel
http://hu.wikipedia.org/wiki/ARM_architekt%C3%BAra
http://hu.wikipedia.org/wiki/MOS_Technology_6502
http://hu.wikipedia.org/wiki/Robot
http://mindstorms.lego.com/en-us/Default.aspx?domainredir=lego.com
http://www.ev-3.net/en/archives/850
http://www.geeks.hu/blog/ces_2013/130108_lego_mindstorms_ev3
http://www.hdidakt.hu/mindstorms.php?csoport=50
http://www.lego.com/en-us/mindstorms/support/faq/
http://www.lego.com/hu-hu/mindstorms/downloads/software/ddsoftwaredownload/download-software/
http://www.legomindstormsrobots.com/lego-mindstorms-ev3/programming-ev3-c-bricxcc/
http://www.leg-technic.hu/blog/38/31313-mindstorms-ev3-az-itelet-elso-napja
http://www.leg-technic.hu/blog/39/31313-mindstorms-ev3-az-itelet-masodik-napja
http://www.philohome.com/sort3r/sort3r.htm
LEGO Mindstorms EV3 Felhasználói útmutató (www.lego.com)
LEGO MINDSTORMS EV3 Home Edition súgó

Kovács Lehel István