

GINOP-6.2.3-17-2017-00038

Lemorzsolódás csökkentése a Szekszárdi Szakképzési Centrum tagintézményeiben

Kompetencia alapú képzésekhez kidolgozott új tananyag, módszertan leírása

Szakmai megvalósító (tagintézmény) megnevezése, címe:	Szekszárdi SZC Ady Endre Szakképző Iskolája és Kollégiuma 7100, Szekszárd, Széchenyi u. 2-14
A fejlesztés megnevezése:	Lego Mindstorms EV3 robotok programozása
A fejlesztés típusa:	<u>Tananyagfejlesztés</u> / Módszertani fejlesztés
A fejlesztés kiemelt céljai:	matematikai, a digitális és a szakmatanuláshoz szükséges kompetenciák fejlesztése
A fejlesztésben érintett célcsoport:	9. évfolyamos tanulók
A fejlesztés alkalmazási területe:	informatika tanóra, fejlesztő foglalkozások
A fejlesztés kapcsolódik-e a NAT, helyi szakmai, képzési program valamelyik kiemelt fejlesztési feladatának – nevelési-oktatási-képzési céljának a megvalósulásához? Ha igen, hogyan?	A NAT kompetenciaterületei közül a digitális és a matematikai kompetencia fejlesztését szolgálja a robotprogramozás.
A fejlesztés részletes bemutatása (előzetes ismeretek, időkeret, felépítés, tananyagtartalom, munkamódszerek, eszközök, tárgyi és személyi feltételrendszer, költségvonzat, pedagógiai és módszertani lépései, résztvevők száma, kivitelezés szabályai, felhasznált források stb.)	
Előzetes ismeretek: alapszintű számítógép-kezelői ismeretek, amellyel 9. évfolyamban már rendelkeznek tanulóink Időkeret: 8-12 óra Tananyagtartalom: robotprogramozás a fejlesztett tananyag alapján Munkamódszerek: csoportmunka, projektmunka, szemléltetés, munkáltatás Tárgyi eszközök: Legalább egy Lego Mindstorms EV3 robotkészlet és 1 számítógép/ 6 fő Személyi feltételek: 1 fő pedagógus (informatika szakos, vagy Lego robotprogramozási továbbképzést végzett) Résztevők száma: legfeljebb 18 fő. Felhasznált források: <i>Kiss Róbert-A MINDSTORMS EV3 robotok programozásának alapjai.</i>	
A fejlesztés megvalósulása, eredményei és tapasztalatai:	

A tananyagot eddig 9. évfolyamos informatika órákon teszteltük. A legnagyobb sikere informatikai ágazaton volt, de a többi szakköznevelési osztályban is jól hasznosítható. Az informatikai ágazaton az összetettebb, az egyéb ágazatokon az egyszerűbb feladatok megoldása a célravezetőbb.

Ellenőrzés, visszacsatolás bemutatása:

Javasolt ellenőrzés: A csoportok egymás programjait a pedagógussal közösen ellenőrzik és értékelik.

Nyilvánosság, tudásmegosztás lehetősége:

a tananyag feltölthető az iskola weblapjára PDF-formátumban

Balesetvédelmi megfontolások:

Számítógép-termi rend ismerete és betartása, általános- tűz és balesetvédelmi szabályok.

Egyéb megjegyzés:

-

Kapcsolódó segédanyagok, dokumentumok száma, megnevezése (ld. csatolmányok):

Egy darab 18 oldalas elektronikus tananyag a LEGO Mindstorms EV3 robotok programozása címmel

GINOP-6.2.3-17-2017-00038

Lemorzsolódás csökkentése a Szekszárdi Szakképzési Centrum tagintézményeiben

Kompetencia alapú képzésekhez kidolgozott új tananyag megnevezése:

Lego Mindstorms EV3 robotok programozása

A kidolgozott tananyag, szakmai tartalma (rövid leírás):

Szakképző intézményként kiemelt feladatunk matematikai, a digitális és a szakmatanuláshoz szükséges kompetenciák fejlesztése. Erre nyújt kitűnő alkalmat a robotok megépítése és programozása. A programozás a logikai-matematikai, a robotépítés a térbeli-vizuális, míg a csoportmunka az interperszonális tehetségterület erősíti. Ezen a kompetenciák fejlesztése a lemorzsolódás csökkentésében is nagy szerepet játszik.

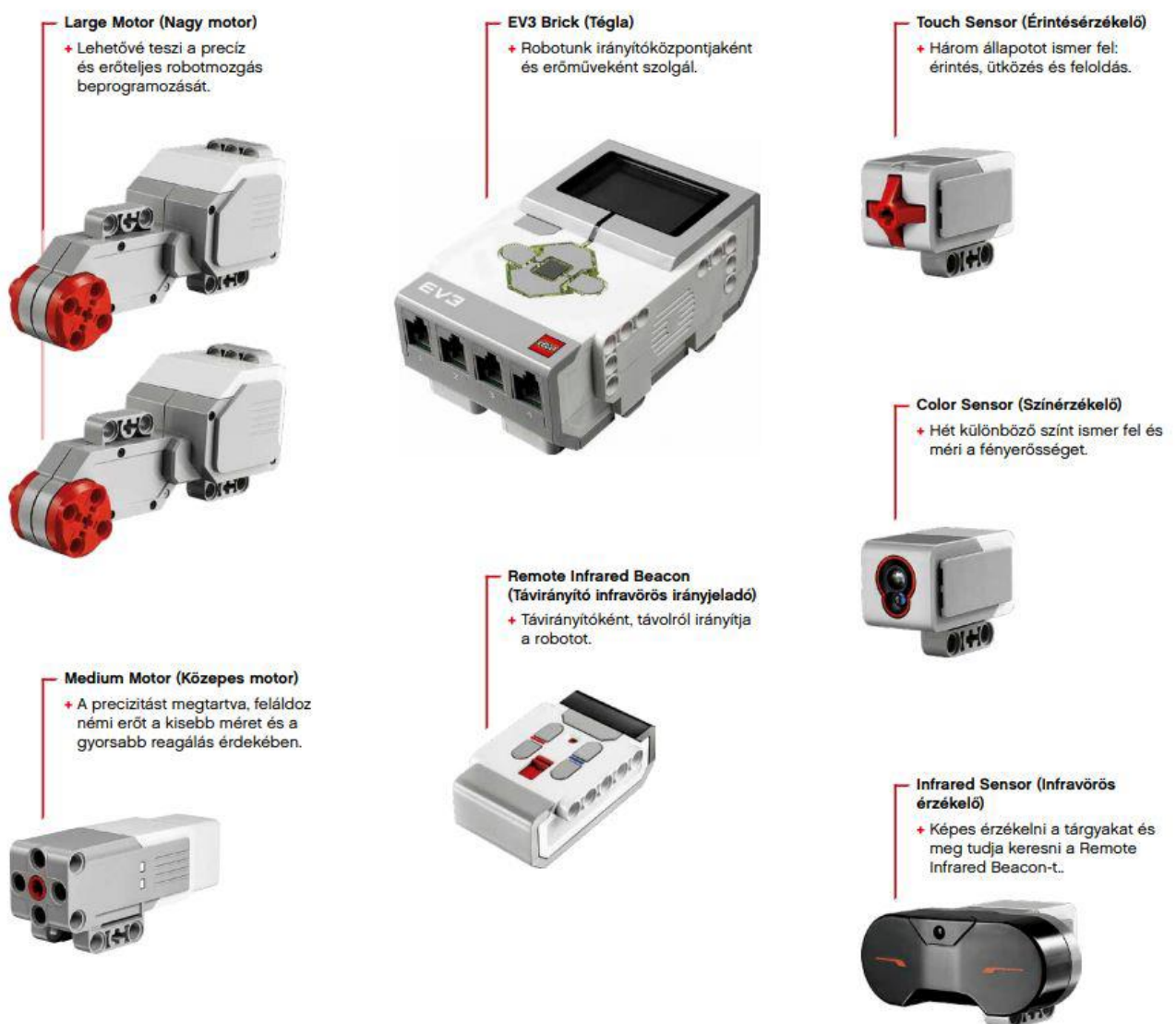
A robotprogramozási környezet nem igényel gépelési tudást, egérhasználattal lehet a legbonyolultabb algoritmusokat is elkészíteni, így a diákok által megszokott informatikai eszközhasználati rutinok működnek és nem szabnak határt a tanulói kreativitásnak.

Tartalom

1. A Lego Mindstorms EV3 felépítése az „intelligens” téglák és egyéb hardver elemek.....	3
2. A programozási környezet bemutatása, a programozás alapjai.....	4
3. A téglák képernyőjének programozása.....	8
4. Mozgások.....	10
5. Szenzorok használata	12
6. Track3R robot felépítése	14
7. Feladatok a Track3R robotra	15

1. A Lego Mindstorms EV3 felépítése az „intelligens” téglák és egyéb hardver elemek

A robotkészlet agya egy 300 MHz-es, Linux alapú ARM9-es mikrokontroller. A központi memóriája 64 MB RAM-ot, és 16 MB Flash memóriát tartalmaz, amely a beépített mikro SDHC kártyahelynek köszönhetően tovább bővíthető. Kijelzőként egy 178x128 pixel felbontású monokróm grafikus képernyő tartozik az eszközhöz. Négy bementi és négy kimeneti csatlakozási port található a téglán (*brick*), amely a motorok és a szenzorok illesztésére alkalmas.



1. ábra. A Mindstorms EV3 robotkészlet fő részei

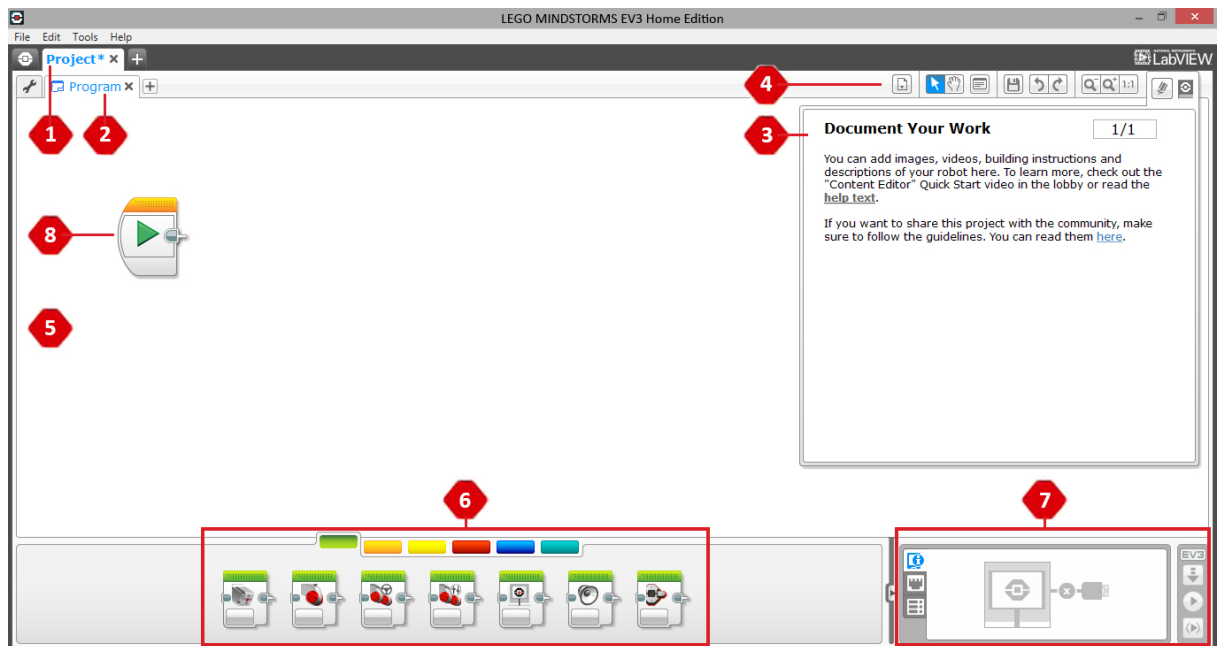
A bemeneti portokat 1-től 4-ig számozták és jelölték a tégla alsó részén. A három kimeneti portra elsősorban szervomotorok kapcsolhatók, esetleg ledek. A kimeneti portokat A-tól D-ig betűzték a tégla felső részén. Itt kapott helyet a mikro USB csatlakozási pont is, amelynek segítségével számítógéphez csatlakoztatható és a programok feltöltésére használható. A tégla oldalán találjuk az SDHC kártya bővítő helyet és egy USB csatlakozási pontot.

A működés elve az, hogy a megépített robotkonstrukció tartalmazza a téglát, valamint az ahhoz csatlakoztatott szenzorokat és motorokat. Számítógépen elkészítjük a célnak megfelelő programot, amely a szenzorok által érzékelt adatok alapján döntéseket hoz a szükséges tevékenységről, amelyet a robot a motorjai segítségével végrehajt. A programot USB kábelen, bluetooth-on vagy wifi-n keresztül a robotra töltve az már önállóan viselkedik a programja utasításai alapján. A bluetooth-os, illetve wireless kommunikáció miatt a robotok egymással is kommunikálhatnak, és programjuk alapján csoportos viselkedésre is képesek.

2. A programozási környezet bemutatása, a programozás alapjai

A programot a következő honlapról lehet letölteni: <https://www.lego.com/hu-hu/themes/mindstorms/downloads>.

A szoftver a LEGO Group és a National Instruments közös fejlesztése, amely a LabVIEW alapokhoz illeszkedően különösebb előképzettség nélkül lehetővé teszi a programozást. A szoftverben egyszerű, zömében egérhasználattal megoldható programírás az ikonok egymás után illesztését és a megfelelő paraméterek beállítását jelenti. A grafikus EV3-G programnyelv használata egyszerű, és vizualitása miatt igen szemléletes. Az egyes hardver elemeket és a legfontosabb programozástechnikai eszközöket egy-egy ikon reprezentálja. Ezeknek az objektumoknak az egymás után fűzéséből vagy akár elágazásokat tartalmazó láncából épül fel a program.

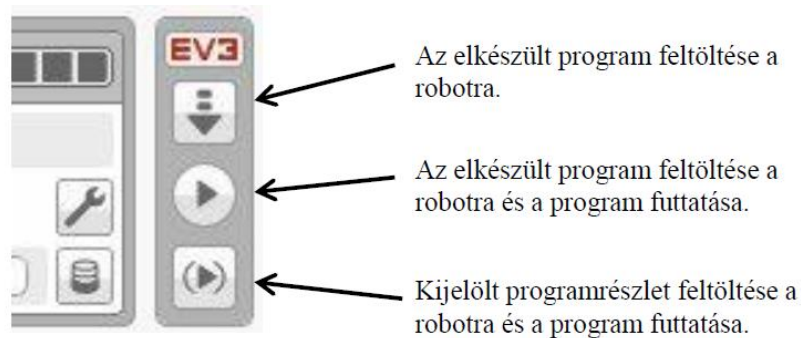


2. ábra. A programozói felület

A 2. ábrán látható programozói felület részei a következők:

1. Projekt címe
2. Program címe
3. Dokumentáció készítő
4. Gyorsgombok
5. Programozói munkaablak
6. Program blokkok
7. Hardveroldal
8. Program kezdőpontja

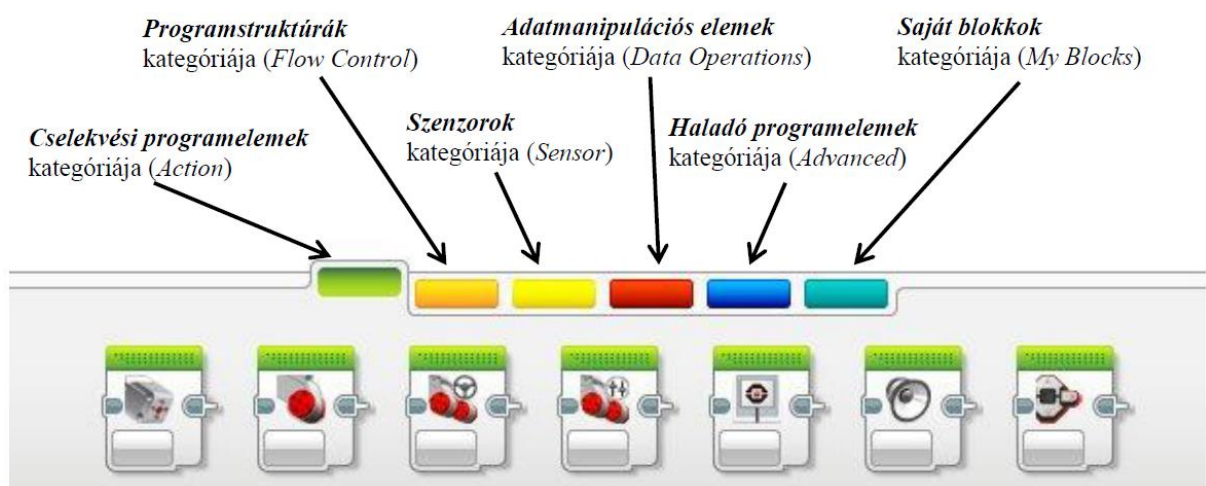
A robot és a számítógép közötti adatcserét megvalósító funkciók a programozói felület jobb alsó sarkában található hardveroldal ikoncsoporton keresztül érhetők el.



3. ábra. Hardveroldal ikoncsoport

A programírás az egyes utasításokat szimbolizáló ikonok egymás után illesztését jelenti. A programikonok funkcióik szerint csoportokba rendezve érhetők el a szoftverkörnyezetben. A képernyő alsó részén tudunk az egyes programcsoportok között navigálni. A megfelelő programcsoportot választva a benne található programikonok, blokkok elérhetők és egérrel a programozási területre húzhatók.

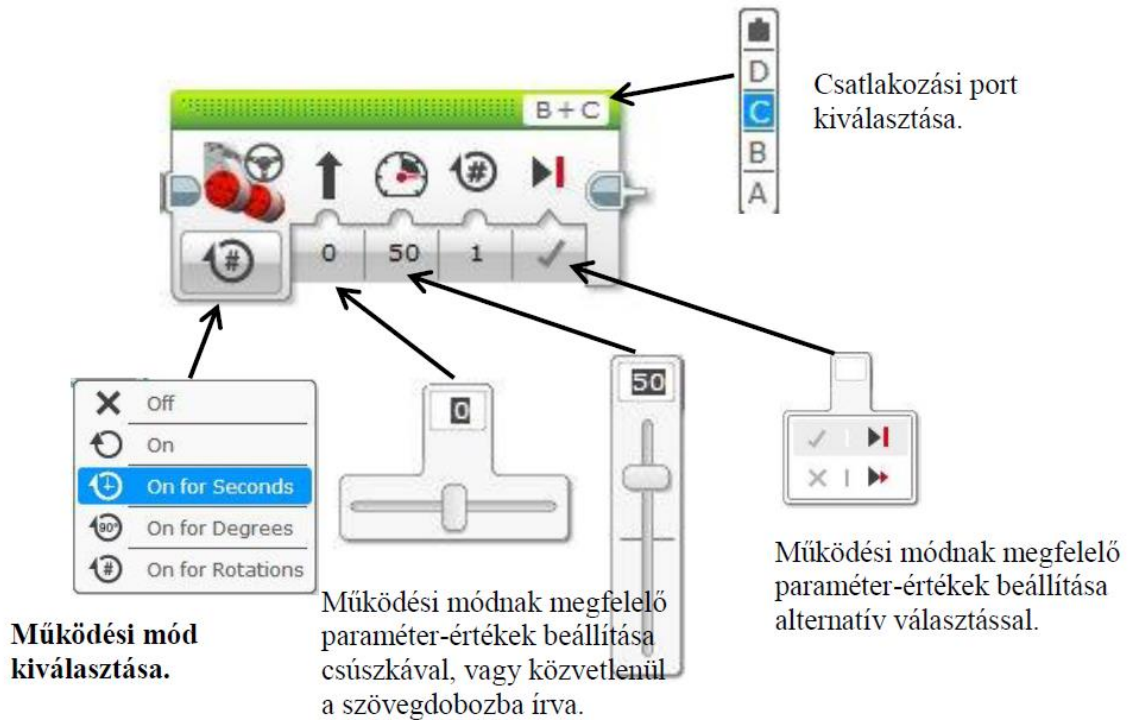
Az egyes programcsoportokat színekkel különböztetik meg. A csoportot jelentő megfelelő színű téglalapra kattintva megjelennek csoportba tartozó ikonok, amelyek közül választani tudunk. Egy-egy ilyen ikon reprezentálja a program utasításait. A programcsoporton belüli ikonokat sokszor moduloknak vagy blokkoknak hívják.



4. ábra. A programcsoportok

Az egyes hardver eszközök a téglaportjaira csatlakoznak. A-tól D-ig a motorok, míg 1-től 4-ig a szenzorok. A helyes működés szempontjából fontos megadni, hogy melyik portra milyen eszközt csatlakoztattunk. Ezt a szoftver automatikusan felismeri, ha a robot csatlakoztatva van a számítógéphez. Előfordulhat azonban, hogy meg kell változtatni a port beállítását. Ezt, az utasításblokk jobb felső részén található szimbólumra kattintva tehetjük meg.

Valamennyi blokk esetén használhatók a felsorolt funkciók, de természetesen az utasításnak megfelelően különböző tartalommal. Az alábbi ábra a motorvezérlő utasításblokk esetén mutatja be a lehetőségeket.



5. ábra. Motorvezérlő utasításblokk főbb beállítási lehetőségei

A programírás tehát azt jelenti, hogy:

- ✓ A kitűzött feladatot (a robottól várt viselkedést) lebontjuk egyszerű utasítások sorozatára, amelyek eredményeként a robot a kívánt módon fog viselkedni.
- ✓ Az utasításoknak megfelelő ikonokat egymás után „felfűzzük” a programszálra,
- ✓ Minden utasításnál beállítjuk a megfelelő paraméterértékeket.
- ✓ Ha elkészültünk, a programunkat feltöltjük a robotra és ott elindítjuk.

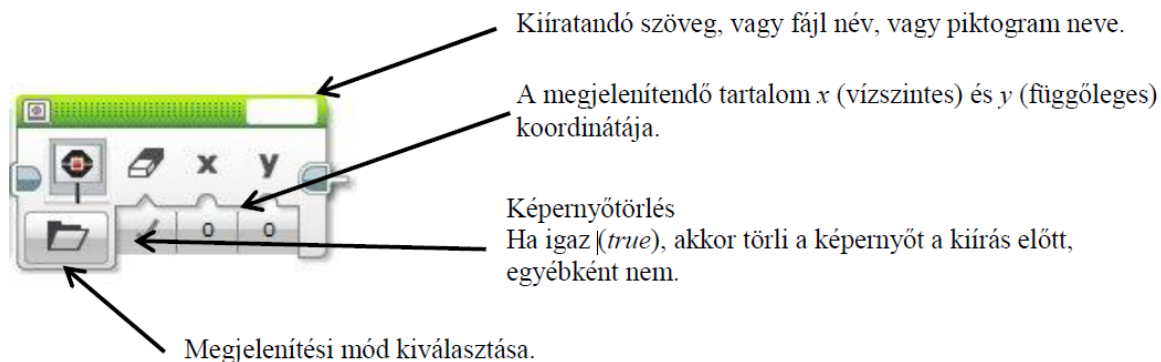
Az alábbi 1. példaprogram a kiírja a téglá képernyőjének közepére 10 másodpercnyi időre, hogy Ady Endre SZKI. (A program elkészítéséhez szükséges Display modul használatának magyarázata „A téglá képernyőjének programozása” című 3. fejezetben történik.)







1. példaprogram megoldása

3. A téglá képernyőjének programozása

A képernyő használata az Action csoporton belüli Display modul programba illesztésével valósítható meg. Alaphelyzetben az ábrán látható funkciók jelennek meg. Beállítva a működési módot a további lehetőségek felkerülnek a blokk ikonjára.



A lehetséges megjelenítési módok:

 Text	Szöveg <i>Pixels</i> vagy <i>Grid</i> beállítással szöveget jelenít meg a képernyőn.
 Shapes	Alakzat Egyenes (<i>Line</i>), kör (<i>Circle</i>), téglalap (<i>Rectangle</i>) vagy pont (<i>Point</i>)
 Image	Kép <i>rbf</i> kiterjesztésű, kétszínű piktogram jeleníthető meg a képernyőn.
 Reset Screen	A képernyő alaphelyzetbe állítása.

6. ábra. Display modul beállítási lehetőségei

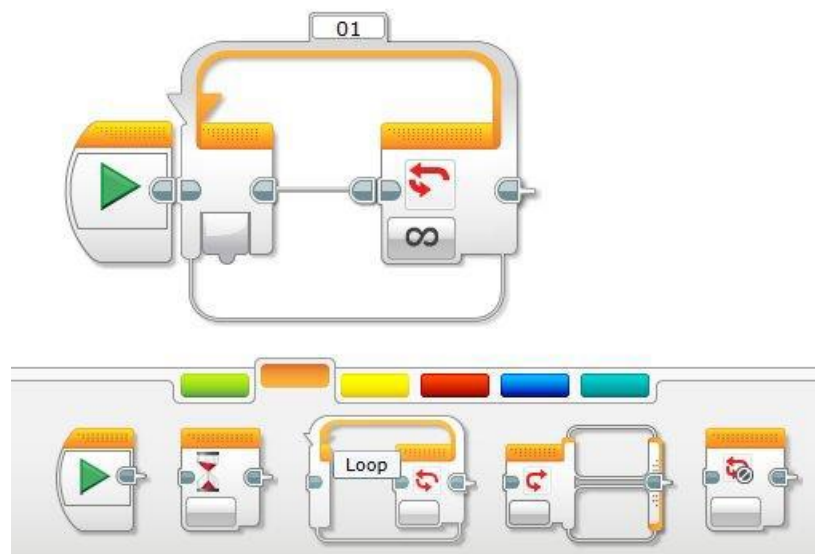
Az 1. példaprogramban az „Ady Endre SZKI” szöveg kiírása történik Text megjelenítési módban a 40 oszlop 63 sortól kezdődően 10 másodperc hosszan. A 10 másodpercet a Wait modul segítségével lehet beállítani.

2. példaprogram: Írjunk programot, amely egymás után szavanként jeleníti meg a képernyőn az Ady Endre SZKI szöveget, mindegyik szó 3 másodperc után kövesse a következőt majd a teljes felírat 10 mp-ig látszódjon és ez ismétlődjön meg egymás után 10-szer! Az ismétlődéshez ciklusra van szükség. Ciklusokat akkor használunk, ha egy utasítást vagy utasítássorozatot többször meg kell ismételni. Az ismétlődő utasításokat ciklusmagnak vagy ciklustörzsnek nevezzük. A ciklusok ikonja a Loop ikon, amely a Flow Control kategóriában található.

Abban az esetben, ha előre tudjuk, hogy hányszor kell megismételni a ciklusmagot, növekményes ciklusról beszélünk. Ilyenkor a ciklusmag mindig egy előre megadott

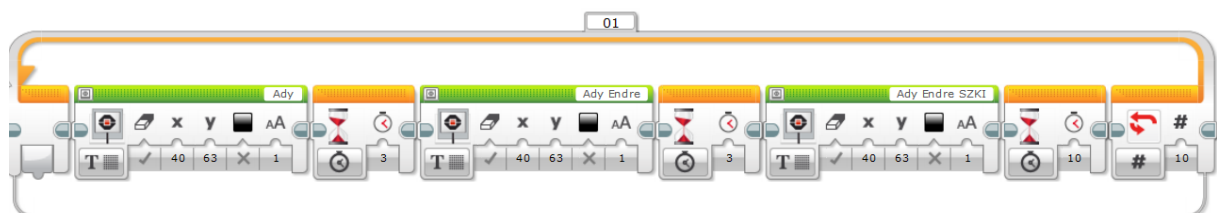
darabszámra hajtódik végre. Ha nem ismert előre az ismétlések száma, akkor a ciklusmag végrehajtását valamilyen feltétel teljesüléséhez, vagy nem teljesüléséhez kötjük. Az ilyen ciklusokat feltételes ciklusoknak nevezzük. A feltételes ciklusoknak két fajtáját különböztetjük meg. Az előfeltételes ciklus esetén a feltétel vizsgálata a ciklusmag végrehajtása előtt történik. Utófeltételes ciklusnál a feltételt a ciklusmag végrehajtása után vizsgáljuk.

Az EV3 grafikus nyelvben mind a növekményes, mind a feltételes ciklus megvalósítható. A feltételes ciklusok közül a nyelv csak az utófeltételes ciklust tartalmazza.



7. ábra Ciklusok

A 2. példaprogram megoldásban tehát növekményes ciklust kell alkalmaznunk, mert előre tudjuk, hogy 10-szer kell elvégeznünk a megfelelő szavak megfelelő ideig történő kiíratását. Lássuk a 2. példaprogram megoldását:



2. példaprogram megoldása

Gyakorló feladatok:

1. Írj programot, amely kiírja először a nevedet, majd az osztályodat a téglák képernyőjére!
2. Rajzoltass ki a téglák képernyőjére egy céltáblát! (Segítség: Azonos középpontú körök megrajzolása)
3. Írj programot, amely egy pattogó labdát jelenít meg a robot képernyőjén!

4. Mozgások

A robot a vezérlőegységhez kapcsolt motorok segítségével valósítja meg a különböző mozgásokat. A robothoz négy motor csatlakoztatható, melyek csatlakozási helyeit A, B, C és D betűkkel jelölték a téglán.

A készletben két különböző felépítésű motor található. A közepes motor forgási tengelye párhuzamos a motor hosszanti tengelyével (*Medium Motor*), míg a nagy motor forgási tengelye merőleges a hosszanti tengelyre (*Large Motor*).

A motorok vezérlésére négy modul is alkalmas. Ezek a *Action* kategóriában található *Large Motor*, *Move Steering* és a *Move Tank* modulok, valamint a speciális közepes motorhoz tartozó *Medium Motor*.



8. ábra. A leggyakoribb motorvezérlő modulok

A motorok irányítása, és ezeken keresztül a robotkonstrukció mozgatása különböző paraméterek beállítását jelenti. A motorok esetén beállítható működési mód: Azt jelenti, hogy a motor működésének időtartamát mi szabályozza. Ez lehet idő (másodperc), elfordulási szög (a tengely fordulási szöge fokban), tengelyfordulatok száma, illetve a motort lehet be illetve kikapcsolni.

A három motor modul közötti különbségek:

- A *Large Motor* ikonnal egyetlen motor irányítható, amely portját a jobb felső sarokban lehet beállítani.
- A *Move Steering* ikonnal két motor irányítható. Egyetlen sebességparamétert lehet megadni, így a fordulást a két motor közötti sebesség elosztásával lehet szabályozni.
- A *Move Tank* esetén az irányítás a tankok vezérléséhez hasonló: a két motor eltérő sebességű forgatása okozza a kanyarodást. Mintha két botkormányval vezérelnénk a motorokat. Mindkét motornak külön-külön állatható a sebessége, és ezáltal finomabban hangolható a mozgása és gyorsabb mozgást, forgást is eredményezhet.

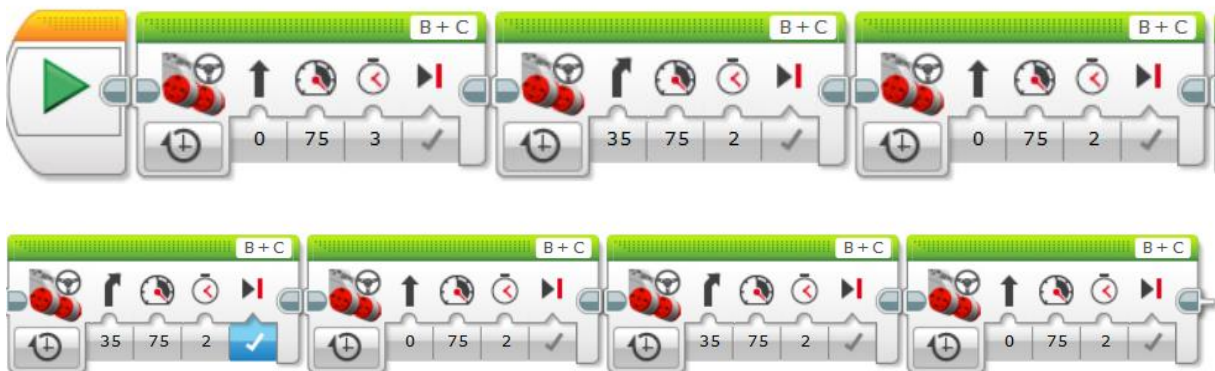
A példaprogramjainkban a Track3R (leírása és építési útmutató a 6. fejezetben) robotot irányítjuk.

3. *példaprogram*: Írjunk programot, amelyet végrehajtva a robot 75-es sebességgel előre halad 5 másodpercig!



3. példaprogram megoldása

4. *példaprogram*: Irányítsuk a robotot úgy, hogy bejárjon egy téglalap alakú útvonalat!



4. példaprogram megoldása

Magyarázat: A programban a B és C portokra kötött nagy motorokat vezéreljük a Move Steering modul segítségével beállított másodperc értékekkel (3 illetve 2 másodperc) és 75-ös forgási sebességgel. A jobbra fordulást a Steering 35-ös értékre állításával érjük el. (Természetesen más megoldás is lehetséges, a robot akár helyben is foroghat, az idő vezérlés helyett is használható más működési mód stb.)

Gyakorló feladatok:

1. Írj programot, amely a robotot egy kör alakú pályán vezeti végig!
2. Írj programot, amely a robotot egy háromszög alakú pályán vezeti végig!
3. Írj programot, amely a robotot 2 percig működteti kígyózó mozgással!

5. Szenzorok használata

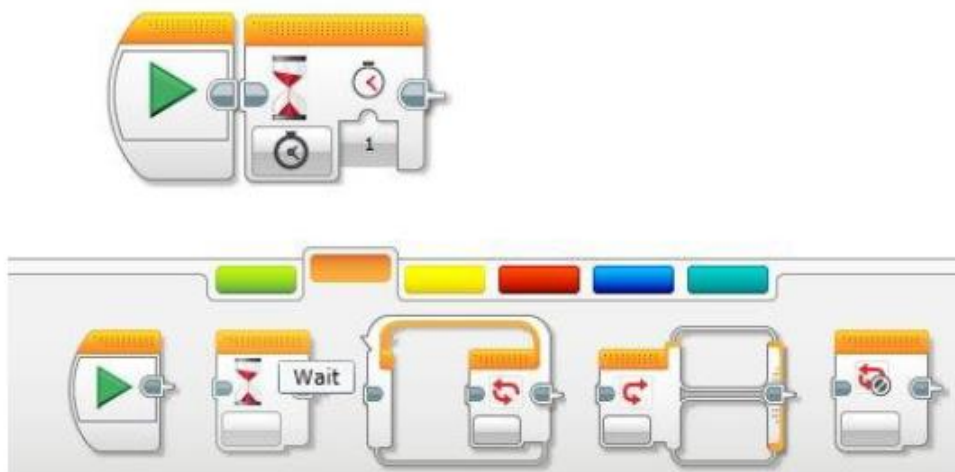
A robot külső érzékelői (szenzorai) segítségével képes érzékelni a „külvilág” jeleit. A szenzorok a téglá 1-4. sorszámú portjaira csatlakoztathatók.

Leggyakrabban a három „alapszenzort” használjuk, amelyek az EV3 oktatási készletben is megtalálhatók. Ezek a következők:

- ütközésérzékelő (*touch sensor*);
- színérzékelő/fényérzékelő (*colour sensor/light sensor*);
- távolságérzékelő (*ultrasonic sensor*)

A robot érzékelőit kétféle módon használhatjuk, ezért a legtöbb szenzorhoz két ikon áll a rendelkezésünkre.

A szenzorok használatának egyik módja, amikor a megfelelő ikon beillesztése a programszálra, felfüggeszti a program további utasításainak végrehajtását, és csak a szenzoron bekövetkezett valamilyen esemény, vagy az érzékelő által visszaadott megfelelő érték hatására folytatódik a programszál utasításainak végrehajtása.



9.ábra. Szenzorok

Ennél a módnál valamennyi szenzor egy ikonnal a *Flow Control* kategóriában található *Wait*-tel vezérelhető (narancssárga szegélyű ikon). A homokóra jelölés a blokkon utal az utasítások várakoztatására.

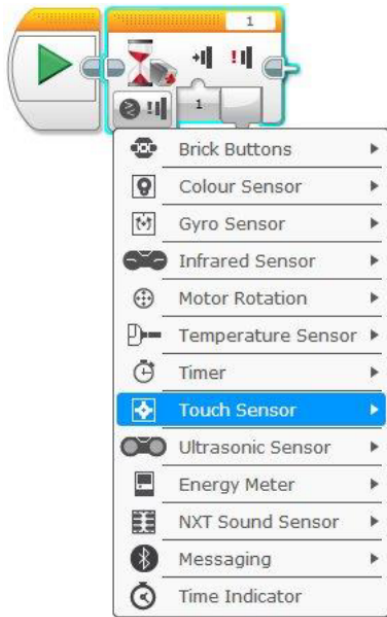
A szenzorhasználat másik módja a Szenzor (*Sensor*) programcsoporton belül található ikonok programszálra illesztésével történik. Az ikonok sárga szegéllyel rendelkeznek.

A továbbiakban a *Wait* ikon működését részletezem.

A *Wait* ikon lényegében addig várakoztatja a programot, amíg a szenzoron beállított esemény be nem következik, vagy amíg a szenzor a paraméterként megadott értéknek megfelelő adatot nem mér. Ha a beállított feltétel teljesül, akkor a vezérlés a program végrehajtását a *Wait*-et követő ikonnal folytatja.

Az ikon bal alsó részén lévő szimbólumra kattintva a működési módot tudjuk beállítani.

A legördülő lista sok elemet tartalmaz:



- A téglán szereplő nyomógombok (benyomott/felengedett állapot)
- Szín/fény szenzor (%-os fényintenzitás érték vagy szín)
- Giroszkóp (elfordulási szög vagy elfordulási arány)
- Infravörös szenzor (távolság érték)
- Motor elfordulási szög (érték fokban vagy tengelyelfordulásban)
- Hőmérséklet szenzor (érték Celsiusban vagy Fahrenheitben)
- Időzítő, stopper (eltelt idő másodpercben)
- Ütközésérzékelő (benyomott/felengedett állapot)
- Ultraszong szenzor (távolság érték cm-ben vagy inchesben)
- Multiméter (feszültség, áramerősség érték, teljesítmény)
- Hang szenzor (decibel érték)
- Bluetooth-on kapott jel (szám, szöveg vagy logikai érték)
- Idő (másodperc)

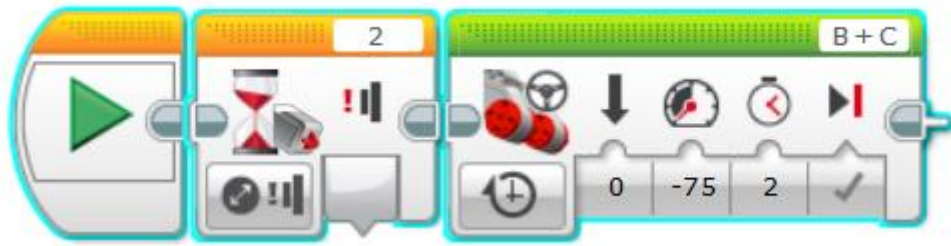
10. ábra. Szenzorok használata a Wait ikonnal

5. példaprogram: Írjunk programot, amelyet végrehajtva a Track3R robot elindul és 3 másodpercig halad egyenesen, ha rávilágítunk a fényszenzorára! (A fényszenzort az 1-es portra kötjük.)



5. példaprogram megoldása

6. példaprogram: Írjunk programot, amelyet végrehajtva a Track3R robot tolatni kezd 2 másodpercig, ha ütközésérzékelőt benyomjuk! (Az ütközésérzékelőt a 2-es portra kötjük.)



6. példaprogram megoldása

6. Track3R robot felépítése

A tananyagban a 7. fejezet feladatai a LEGO Track3R nevű robotjára készültek, amely egy nagyon népszerű robot.



11. ábra. A megépített Track3R robot

Alapesetben 3 motort és egy érzékelőt tartalmaz. A két Nagy motor a B és C portra van kötve és a lánctalpas kerekeket mozgatja, a közepes motor az A portra van kötve és a forgó kart mozgatja. Az alaprobot egy infravörös szenzort tartalmaz a 4-es portra kötve.

A Track3R építési útmutatóját a következő címről lehet letölteni: <https://www.lego.com/biassets/bi/6124045.pdf>. A robot megépítésének lépései a 8.-41. oldalig találhatóak. A következő fejezet feladataiban ezt a robotot fogjuk programozni

7. Feladatok a Track3R robotra

1. feladat: Írjunk programot, amely a robotot 3 másodpercig előre mozgatja, ezután 3 másodpercig megforgatja a robotkart, majd a robot visszatol a kiindulási helyzetébe és végezetül megszólal a fanfár (ünnepélyes jeladás).

Megoldás:



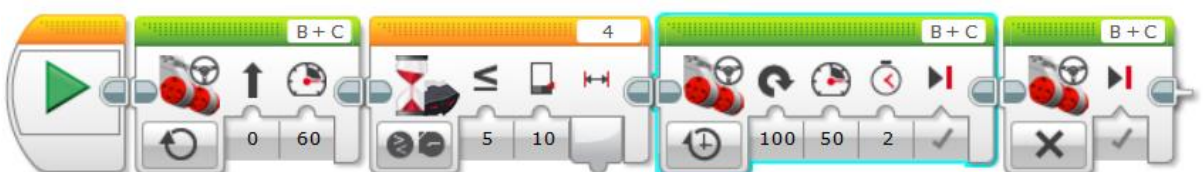
2. feladat: Írjunk programot, amely a robotot mindaddig mozgatja 10-es sebességgel, amíg neki nem ütközik bármilyen objektumnak, ezután visszatol és megáll! (Az ütközésérzékelőt a 2-es portra kötjük.)

Megoldás:



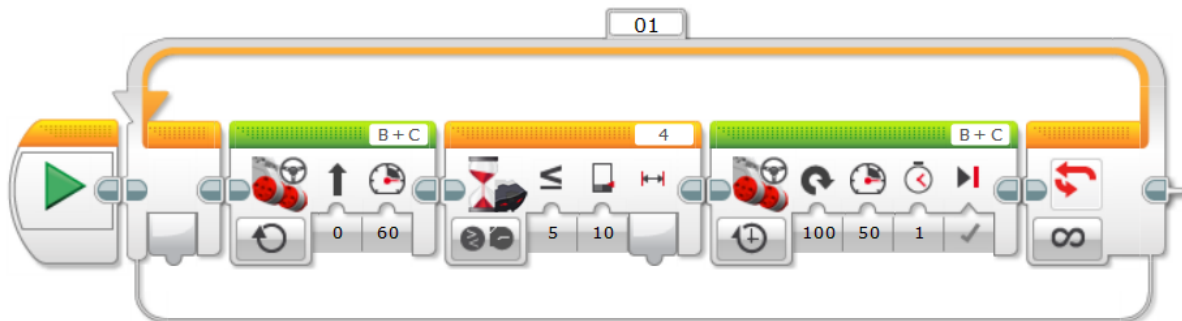
3. feladat: Írjunk programot, amely a robotot előre mozgatja 60-as forgási sebességgel mindaddig, amíg 10 cm-re meg nem közelíti a falat, ezután helyben megfordul és megáll! (A távolságérzékelőt a 4-es portra kötjük.)

Megoldás:



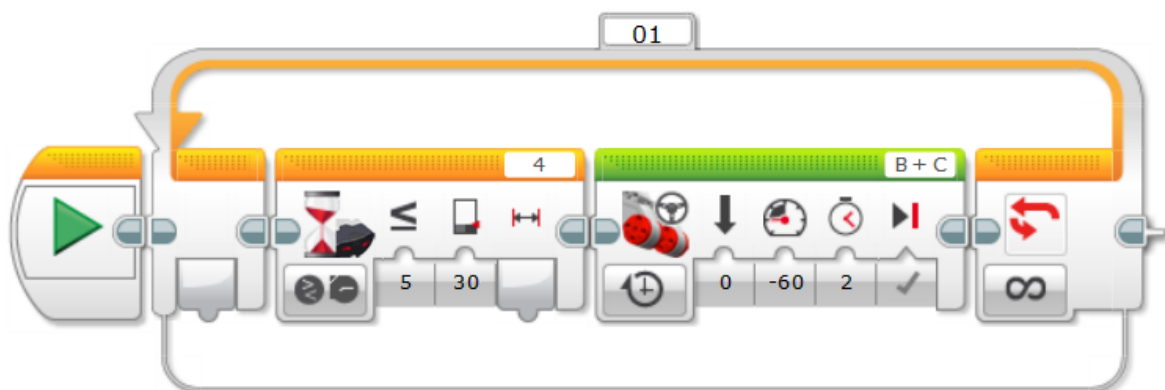
4. feladat: Írjunk programot, amely a robotot előre mozgatja mindaddig, amíg 10 cm-re meg nem közelíti a falat, ezután elfordul 90°-ot és folytatja útját folyamatosan a következő falig és ott ismét elfordul! (A távolságérzékelőt a 4-es portra kötjük.)

Megoldás:



5. feladat: Írjunk programot, amelyben a robot csak akkor indul el 2 másodpercig tolatva, ha távolságérzékelőjét 30 cm-re megközelítjük előlről és ezt folyamatosan ismétli! (A távolságérzékelőt a 4-es portra kötjük.)

Megoldás:

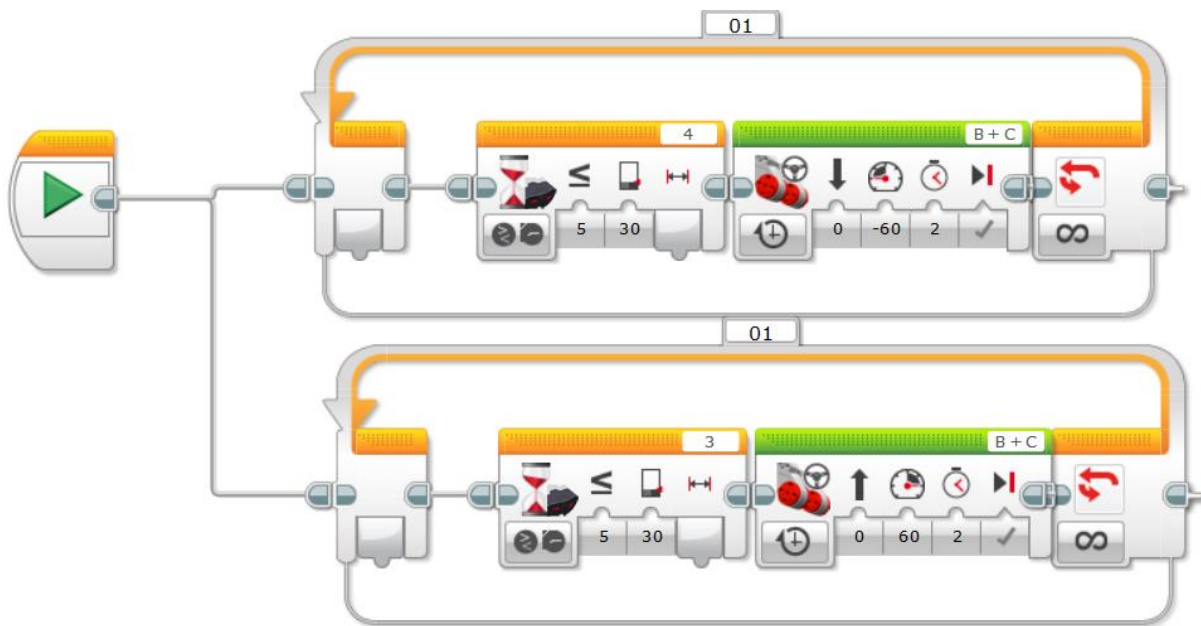


6. feladat: Egészítsük ki az előző programot úgy, hogy ha előlről közelítjük meg 30 cm-re akkor tolasson, ha hátulról közelítjük meg, akkor menjen előre két másodpercig!

Segítség: Két távolságérzékelőre van szükség, robot elejére szereltet a 4-es portra, a hátuljára szereltet a 3-as portra kötjük. Emellett új elemként megjelenik többszálú programozás, hiszen a szenzorok értékeit egyszerre kell figyelni és attól függően, hogy honnan közelítjük meg a robotot, eltérően kell viselkednie.

A programunkban tehát létrehozhatunk egymással látszólag párhuzamosan működő, futó taszkokat (szálakat), így egy időben tudjuk a motorokat vezérelni és a szenzorokkal mért adatokat kezelni.

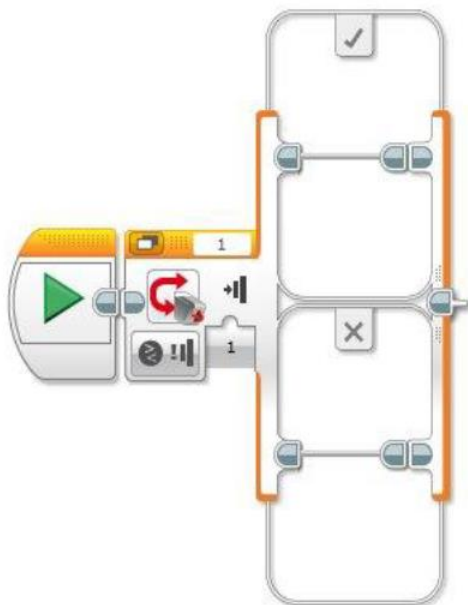
Megoldás:



7. feladat: Írjunk programot, amellyel a robot egy fekete szigetelőszalaggal kijelölt útvonalat követ! (A színszenzort az 1-es portra kötjük.)

Segítség: A megoldáshoz szükség lesz új elemként az elágazásra.

Az elágazásokat a *Flow Control* kategóriában található *Switch* ikonnal hozhatunk létre.



Igaz ág.

Ha a feltétel igaz, akkor kerülnek végrehajtásra a programszálon lévő utasítások.

Hamis ág.

Ha a feltétel hamis, akkor kerülnek végrehajtásra a programszálon lévő utasítások.

12. ábra. Elágazás

Az útvonal követése úgy történik, hogy ha a fény szenzor 50-nél nagyobb értéket mér, akkor a szenzor nem az útvonal felett van. Ilyenkor az egyik motorjával előre halad, a másik motor kikapcsolt állapotban van. Ennek hatására a robot ráfordul az útvonalra. Ekkor azonban 50-nél

kisebbs értéket mér. Ebben az esetben a robot a másik motort forgatja előre, az előzőleg működőt leállítja, és lefordul az útvonalról (szenzora nem az útvonal fölött lesz). Ezt egy végtelen ciklusban ismételve a robot kígyózó mozgással követi az útvonalat. A motorokat nem célszerű túl nagy sebességgel működtetni, mivel ekkor a robot nagyokat fordul, és elveszítheti a követendő útvonalat. A feladat megoldásánál 20-as erővel működtettük a motorokat. A program működése során a robot tulajdonképpen nem az útvonalat követi, hanem a fekete sáv határvonalát.

Megoldás:

