

# LEGO ROBOTOK PROGRAMOZÁSA

**Tompa Tamás**

tanársegéd

Általános Informatikai Intézeti Tanszék

Miskolc, 2018. október 20. ; november 09.

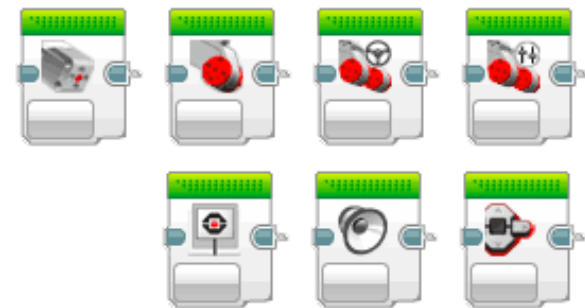
# MENETREND

- Gyakorlat I.
  - 1. Programok írása, futtatása; saját eljárásblokk készítése
  - 2. Motorvezérlés
  - 3. Szenzorok
  - 4. Kijelző használata
  - Szünet (5 perc)
- Gyakorlat II.
  - 1. Komplex (egyéni) feladatmegoldás
  - 2. Robotok közötti kommunikáció
- *Felhasznált és ajánlott irodalom:*
  - Kiss Róbert: A MINDSTORMS® EV3 robotok programozásának alapjai (2014)



# ELŐKÉSZÜLETEK

- Lego EV3 programozása
  - szoftver, algoritmus segítségével
  - utasítások sorozata
- LEGO MINDSTORMS EV3 szoftver
  - Home Edition
  - Education Edition
  - blokkok egymáshoz fűzése
- Programkörnyezetek
  - C, Java, Python programozási nyelvek
  - Matlab
  - LabView
  - RobotC
  - leJOS
  - ROS (Robot Operating System)
  - stb.



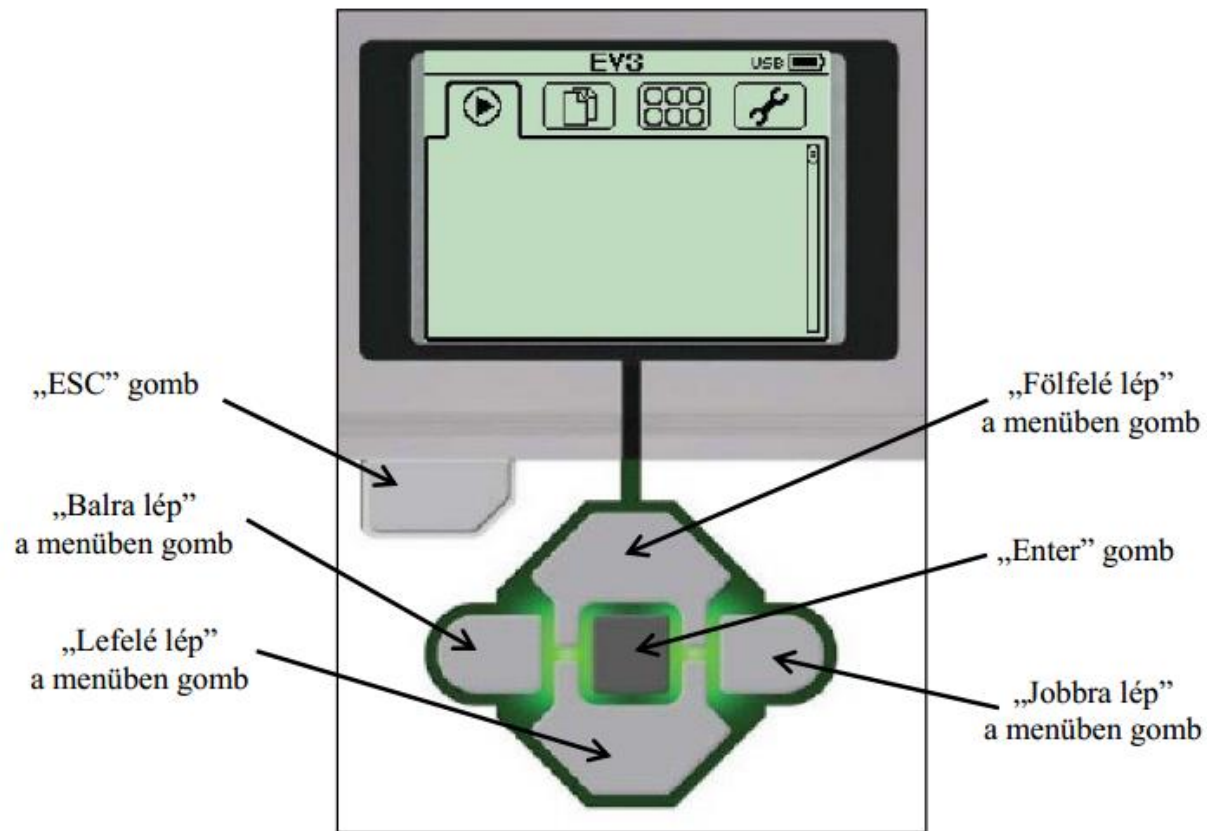
# ELŐKÉSZÜLETEK

- A programozás során használt tesztrobot



# ELŐKÉSZÜLETEK

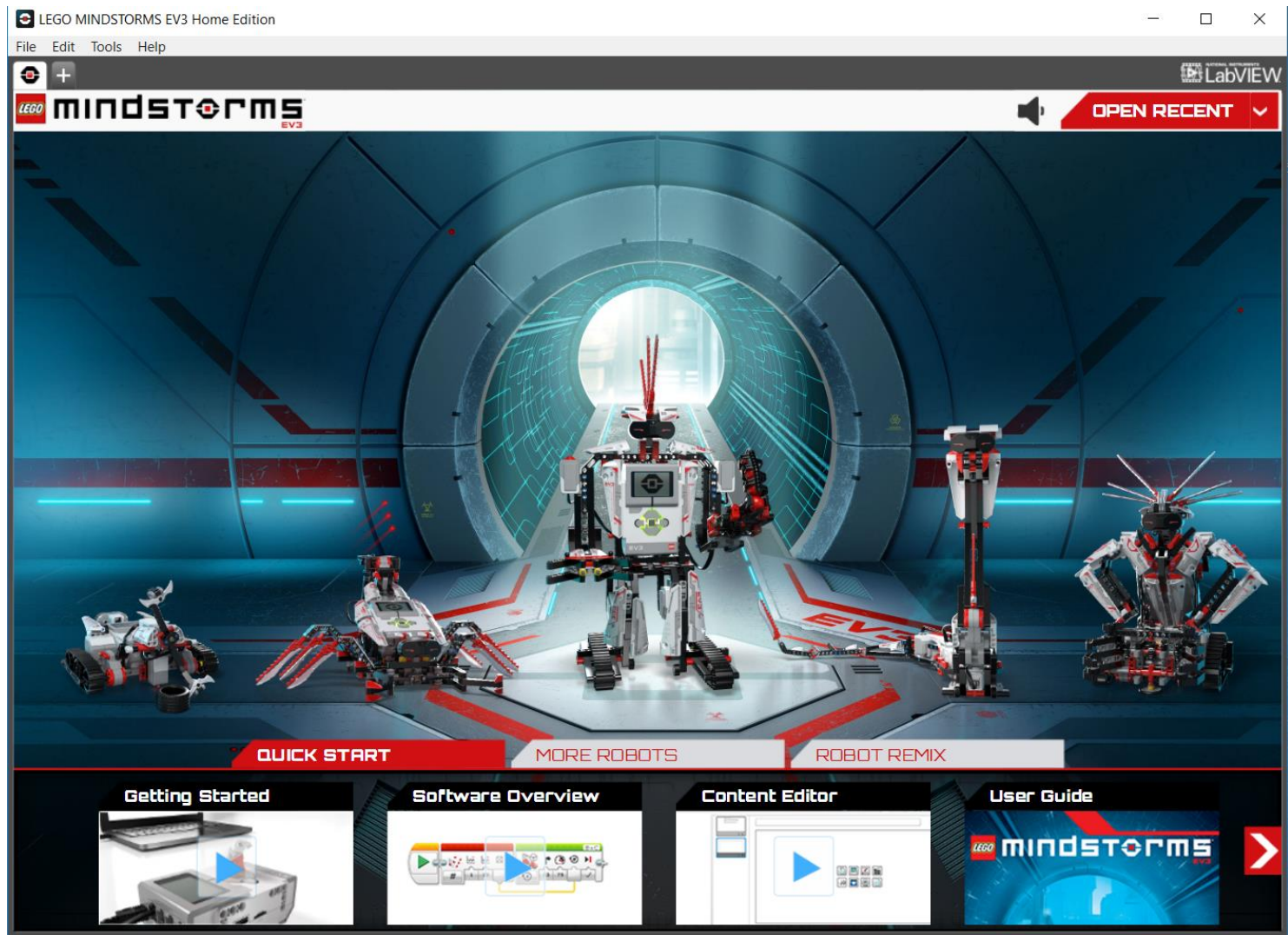
- A robotvezérlő modul felépítése (brick - tégl)



# GYAKORLAT I. - 1.

## LEGO MINDSTORMS EV3 HOME EDITION

- EV3 szoftver nyitó képernyő

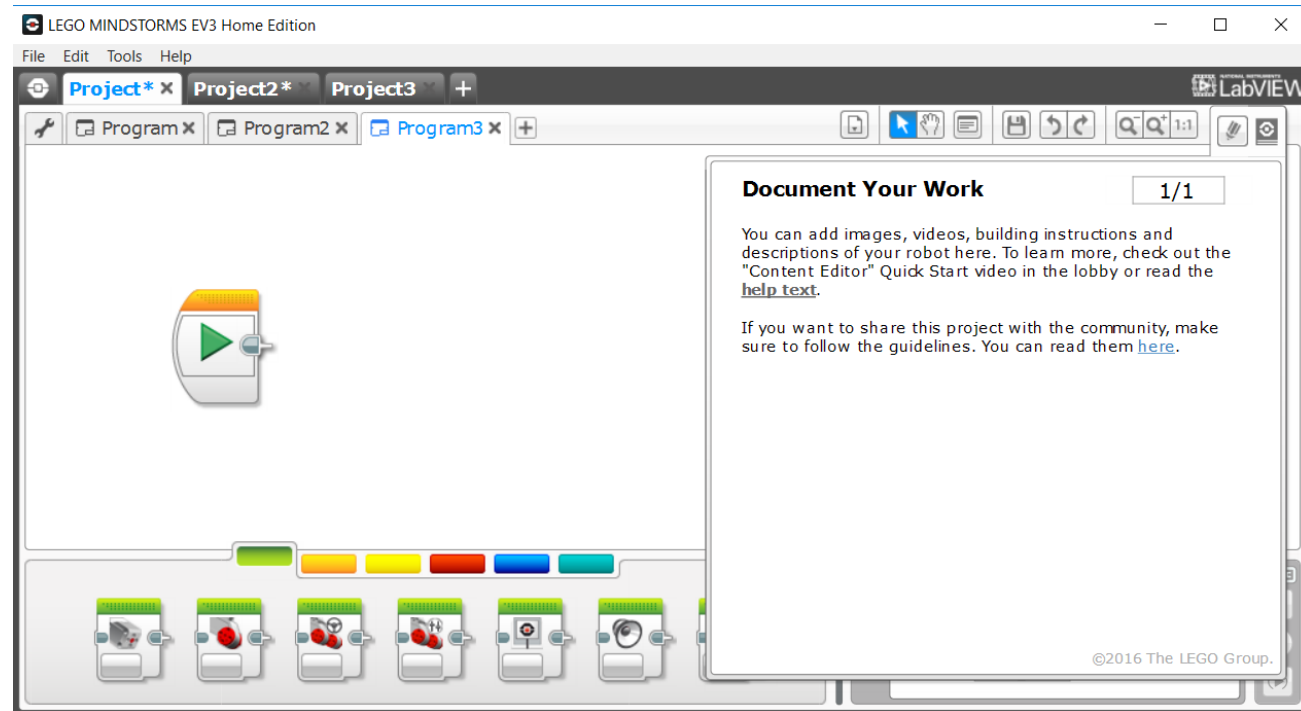




# GYAKORLAT I. - 1.

## EV3 PROJEKT LÉTREHOZÁS

- **Létrehozás**
  - File -> New Project
- **Megnyitás**
  - File -> Open Project
- **Az EV3 szoftveren belül több project is létrehozható**
  - megadott nevekkal ellátva
- **Egy projekten belül több program is lehetséges**
- **\*.ev3 kiterjesztésű állomány (ev3 project file)**



# GYAKORLAT I. - 1.

## EV3 PROJEKT LÉTREHOZÁS

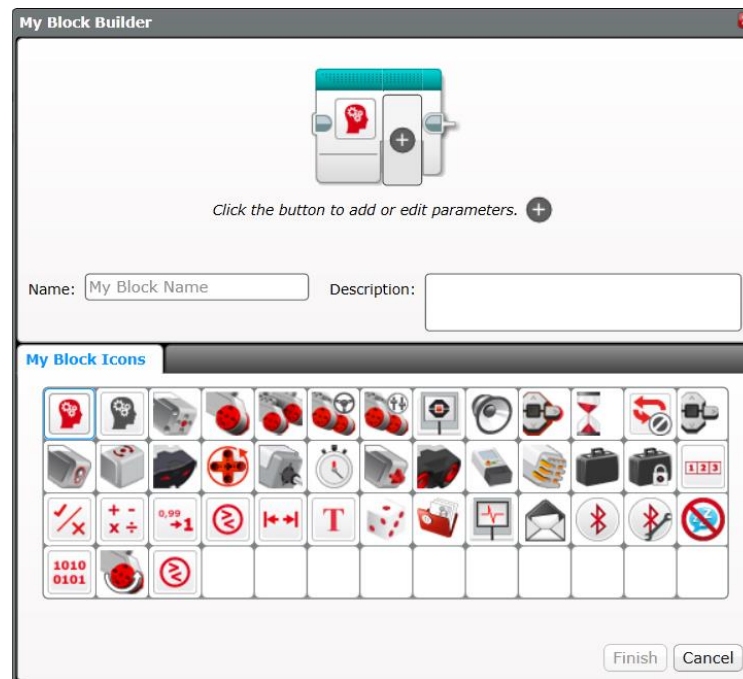
- **Különböző funkcióblokkok, csoportok különböző színekkel jelölve**

- Zöld
- Narancs
- Sárga
- Piros
- Kék
- Cián



- **Sajátblokk létrehozása**

- blokkok kijelölése
- Tool -> My Block Builder
- blokk elnevezése
- ikon megadás
- cián színű csoportban a saját blokk





# GYAKORLAT I. - 1.

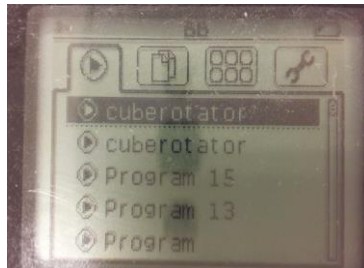
## EV3 PROJEKT MENTÉS, FUTTATÁS

- **Mentés**

- File -> Save Project
- a \*-al jelölt projektek nincsennek elmentve! -> mindig mentsetek!

- **Futtatás**

- a program robotra való áttöltése után
  - USB
  - Bluetooth
  - Wifi
- Az áttöltött programok a projekttel azonos nevű mappába kerülnek a robot belső memóriájába, a futtatás innen is lehetséges
- de futtatható az EV3 szoftverből is



Az elkészült program feltöltése a robotra.

Az elkészült program feltöltése a robotra és a program futtatása.

Kijelölt programrészlet feltöltése a robotra és a program futtatása.

# GYAKORLAT I. - 2.

## MOTORVEZÉRLÉS



Közepes motor

Nagy motor

- **A robothoz 4 motor csatlakoztatható összesen**

- A, B, C, D jelölések a téglán
- Motorvezérlő blokkok



Nagy motor blokk  
(Large Motor)

1 motor



Kormányvezérelt motor blokk  
(Move Steering)

2 motor (1 sebesség)



Sebességvezérelt motor blokk  
(Move Tank)

2 motor (2 sebesség)

# GYAKORLAT I. - 2.

## MOTORVEZÉRLÉS

1. Írjon programot, amelyet végrehajtva a robot 50-es sebességgel előre halad 500°-os tengelyfordulásig!

Az ábra a feladat megoldásának a programkódját mutatja be.



- kormányvezérelt (*Steering Motor*) blokk
- a motorok leállítása nem szükséges
  - 500 fokos tengelyelfordulás után illetve a program végeztével leállnak
  - a motor tengelye fordul 500 fokot!

# GYAKORLAT I. - 2.

## MOTORVEZÉRLÉS

2. Írjon programot, amelyet végrehajtva a robot 2 mp-ig tolat, majd balra fordul kb. 90°-ot, végül előre megy a tengely háromszoros körbefordulásig!

A program forráskódja három kormányvezérelt motor ikonból áll.



- Az első blokk a B és C motorokat –50-es sebességgel 2 mp-ig forgatja. A negatív sebességérték a hátrafelé mozgást jelenti
- A második blokk valósítja meg a kb. 90°-os balra fordulást. Ehhez az összes „nyomatékot” a C (port függő) motorra adjuk és a motorokat 50-es sebességgel 0,5 mp-ig működtetjük (nem a robot 90°-kal történő elfordulása, hanem a motor tengelyének 90°-os elfordulása)
- A harmadik modul a háromszoros tengelyfordulásig történő előre mozgást valósítja meg.

# GYAKORLAT I. - 2.

## MOTORVEZÉRLÉS

3. Írjon programot, amelyet végrehajtva a robot 50-es sebességgel körbe forog 2 mp-ig!

A feladat megoldásának a programkódja:



- kormányvezérelt (*Steering Motor*) blokk
- motorokat 50-es sebességgel, de különböző irányban forgatja 2 mp-ig
- a helyben forgást úgy érjük el, hogy a *Steering* paraméter csúszkáját a C motor irányába toltuk el.
- a *Seconds* paraméter beállításával adtuk meg a mozgás idejét.

# GYAKORLAT I. - 2.

## MOTORVEZÉRLÉS

4. Írjon programot, amelyet végrehajtva a robot 50-es sebességgel körbe forog 2 mp-ig de sebességvezérelt blokkal!



*Mit lehet észre venni ebben a sebességvezérelt módban?*

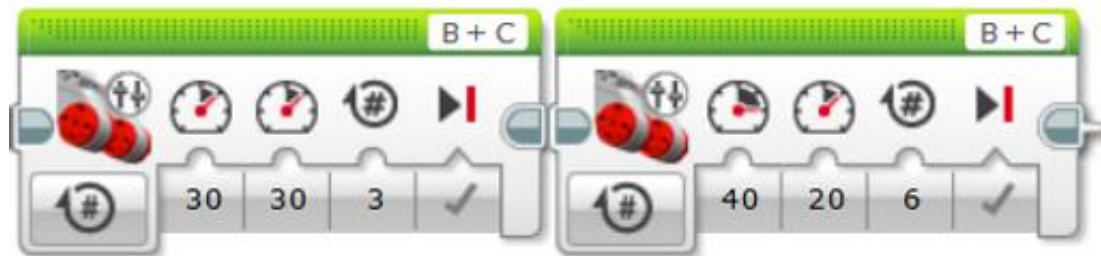


# GYAKORLAT I. - 2.

## MOTORVEZÉRLÉS

5. Írjon programot, amelyet végrehajtva a robot szinusz pályán mozog (saját eljárás készítése)

- Szinusz blokk (Tools -> My Block Builder)



- Program



# GYAKORLAT I. - 3.

## SZENZOROK HASZNÁLATA

### Szenzorok

- 3 alap szenzor

- ütközésérzékelő (*touch sensor*)



- színérzékelő/fényérzékelő (*color sensor/light sensor*)



- távolságérzékelő (*ultrasonic sensor*)



# GYAKORLAT I. - 3.

## SZENZOROK HASZNÁLATA

- Giroszkóp (*gyro sensor*)



- robot előjeles elfordulásának mérése fokban
- előjeles elfordulás értékének viszonyítási pontja a szenzor Reset állapota
- Gyro Sensor blokk beillesztése majd Reset mód  
lenullázás  
innenntől kezdve a két irányú elfordulást előjeles  
értékét kaphatjuk vissza a szenzortól
- Vezérlés  
elfordulási **szög**: *Angle* (szög),  
elfordulási **arány**: *Rate*, (a fordulás szögének időbeli változása szög/másodperc mértékegységben)

# GYAKORLAT I. - 3.

## SZENZOROK HASZNÁLATA

1. Írjon programot, amelyet végrehajtva a robot egyenesen halad előre mindaddig, amíg a fényérzékelője az alapszíntől eltérő színt nem észlel, ekkor álljon meg!

### Előkészületek

- A feladat végrehajtása során homogén fehér felületen mozog a robot.
- Az eltérő színű csíkot pl. fekete színű szigetelő szalag felragasztásával állíthatjuk elő.
- A program megírása előtt a képernyő *View* funkciójával a különböző színű felületekről visszavert fény intenzitását megmértük, fehér: 62; fekete: 26.
- $62 + 26 = 44$  (határérték: átlag)
- Tehát ha a fényérzékelő 44-nél kisebb értéket mér, akkor a robot elérte a fekete csíkot.

# GYAKORLAT I. - 3.

## SZENZOROK HASZNÁLATA



- összehasonlítás módot és a fényszenzort saját fénnel (Reflected Light)
- az **előre mozgás**nál a motorerőt 50-re, a működési módot On-ra állítottuk, mivel nem tudjuk előre, hogy mikor ér a robot a fekete színű vonalhoz.
- a második ikon addig várakoztatja (**wait** blokk!) a program utasításainak végrehajtását, amíg a **fényérzékelő** által mért **érték kisebb** nem lesz **44-nél**
- így a robot a fekete csík eléréséig **folyamatosan halad előre**.
- elérve a fekete csíkot a mért **érték kisebb lesz 44-nél**, ezért a program a Wait-et követő utasítással folytatódik, és a robot **megáll**.

# GYAKORLAT I. - 3.

## SZENZOROK HASZNÁLATA

2. Írjon programot, amelyet végrehajtva a robot egyenesen halad mindaddig, amíg a távolságérzékelője 15 cm-nél kisebb távolságot nem mér! Ekkor álljon meg!

A következő ábra a feladat megoldásának programkódját mutatja:



- a távolságérzékelő a 4. portra van kötve (**wait blokk!**).
- a mértékegységet centiméterre, a továbblépési feltételként mért távolságot 15 cm-nél kisebbre állítsuk



# GYAKORLAT I. - 3.

## SZENZOROK HASZNÁLATA

3. Írjon programot, amellyel ha a robot 10 cm-en belül mozgást érzékel akkor hangot ad ki.

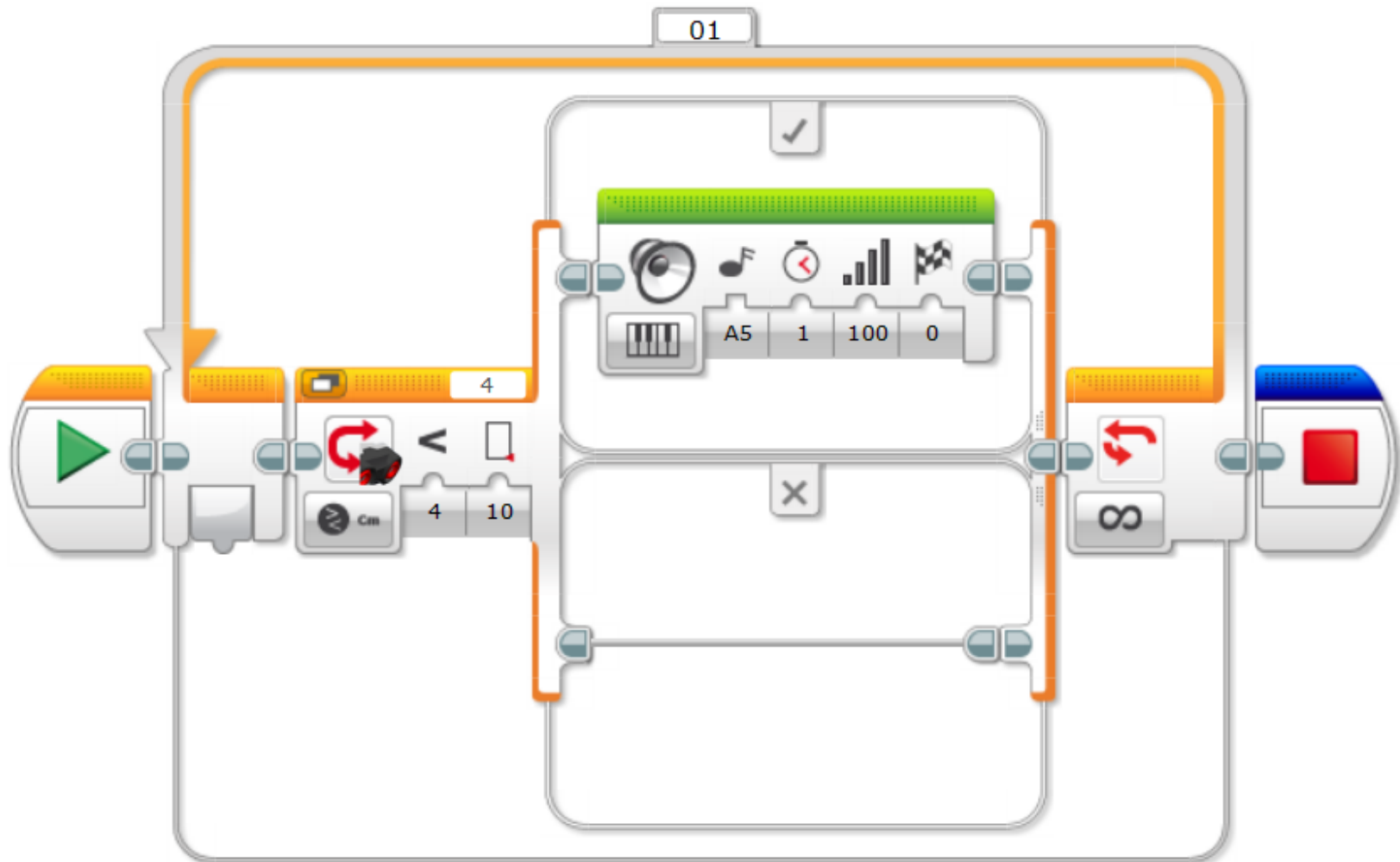
### Algoritmus:

- távolságmérés
- a mért távolság **kisebb-e** mint **10** cm
- Előbbi feltétel **ellenőrzése**
  - **igaz**: hang kiadás
  - **hamis**: **semmi** sem történik
- folyamat **ismétlése** előlről

# GYAKORLAT I. - 3.

## SZENZOROK HASZNÁLATA

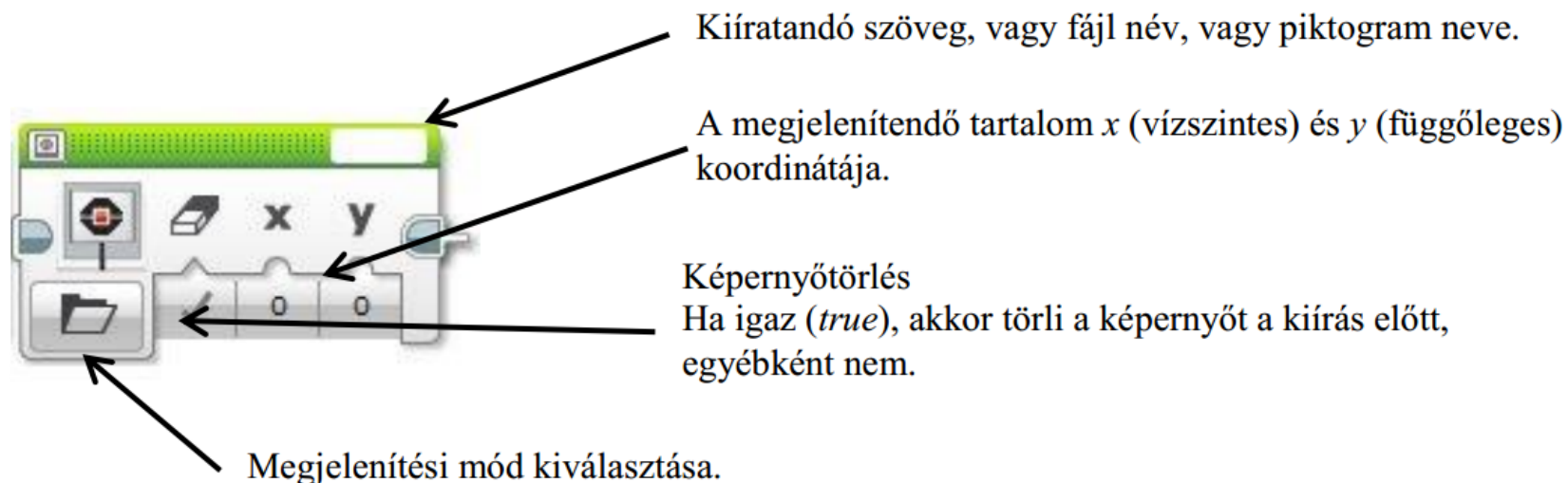
Megoldás:



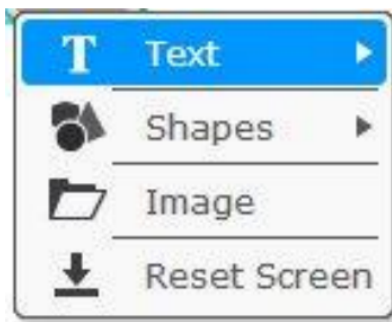
# GYAKORLAT I. - 4.

## KIJELZŐ HASZNÁLATA

- képernyő használata az *Action* csoporton belüli *Display* modul programba illesztésével valósítható meg



A lehetséges megjelenítési módok:



Szöveg

*Pixels* vagy *Grid* beállítással szöveget jelenít meg a képernyőn.

Alakzat

Egyenes (*Line*), kör (*Circle*), téglalap (*Rectangle*) vagy pont (*Point*)

Kép

*rbf* kiterjesztésű, kétszínű piktogram jeleníthető meg a képernyőn.

A képernyő alaphelyzetbe állítása.

# GYAKORLAT I. - 4.

## KIJELZŐ HASZNÁLATA

1. Írjon programot, amelyet végrehajtva a robot a képernyőjére folyamatosan kiírja az ultrahangos távolságérzékelője által mért értéket! A robot a programot kikapcsolásig ismétlje!

### Algoritmus:

- távolságmérés (ultrahangos, 4es port)
- a mért **érték kiírása** (Display blokk), 10,10 koordinátára
- Minden kiírás előtt a **képernyő törlése**
- **kiírás** fekete színnel
- folyamat **ismétlése** előlről

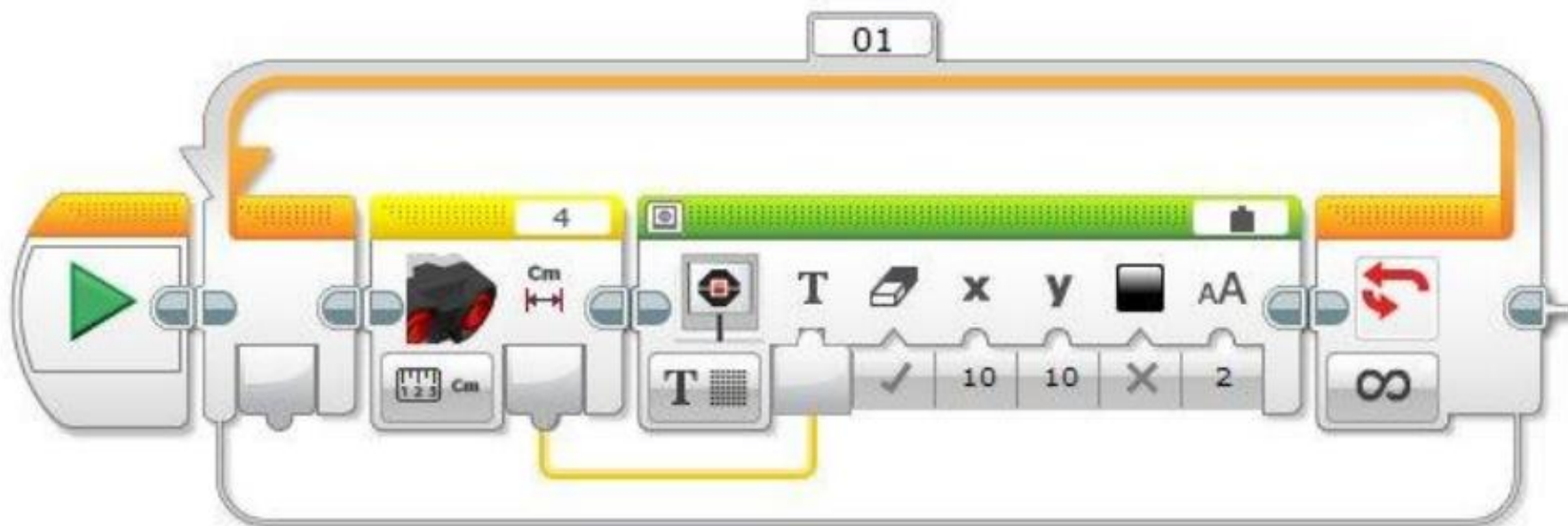
A 4-es portra csatlakoztatott ultrahangos távolságérzékelő által mért értéket kapja meg az Display blokk és a nagy méretben írja a képernyő (10;10) koordinátájú helyétől kezdődően (a kiírt érték bal felső sarka kerül az adott koordinátára). Minden kiírás előtt törlődik a képernyő és a kiíratást fekete színnel végezzük.

# GYAKORLAT I. - 4.

## KIJELZŐ HASZNÁLATA

1. Írjon programot, amelyet végrehajtva a robot a képernyőjére folyamatosan kiírja az ultrahangos távolságérzékelője által mért értéket! A robot a programot kikapcsolásig ismétlje!

Megoldás:



# Szünet (5 perc)



# GYAKORLAT II. - 1.

## EGYÉNI KOMPLEX FELADATMEGOLDÁS

1. **Asztalon marad:** A robot az asztal szélén megáll, megfordul, majd ismét az asztal széléig megy. Figyelem! A fényszenzor elhelyezésétől függően a robot lebillenhet az asztalról.

### Algoritmus:

- a robot **halad előre** valamekkora sebességgel
- **fénymérés**, a visszavert fény intenzitása 0-e, azaz nincs visszavert fény -> asztal széle
- **megáll**
- **tolat** 1s-ig valamekkora sebességgel
- **elfordul** valamelyik irányba valamekkora sebességgel
- győro szenzorral **elfordulás figyelése**
- majd a teljes folyamat **ismétlése**



# GYAKORLAT II. - 1.

## EGYÉNI KOMPLEX FELADATMEGOLDÁS

### 1. Asztalon marad:

Megoldás



# GYAKORLAT II. - 1.

## EGYÉNI KOMPLEX FELADATMEGOLDÁS

### 2. Mozgó tárgy követése távolságérzékelővel:

Működése: 250 cm-en belül a legközelebbi akadályhoz képest 50 cm-es távolságot tart.

Algoritmus:

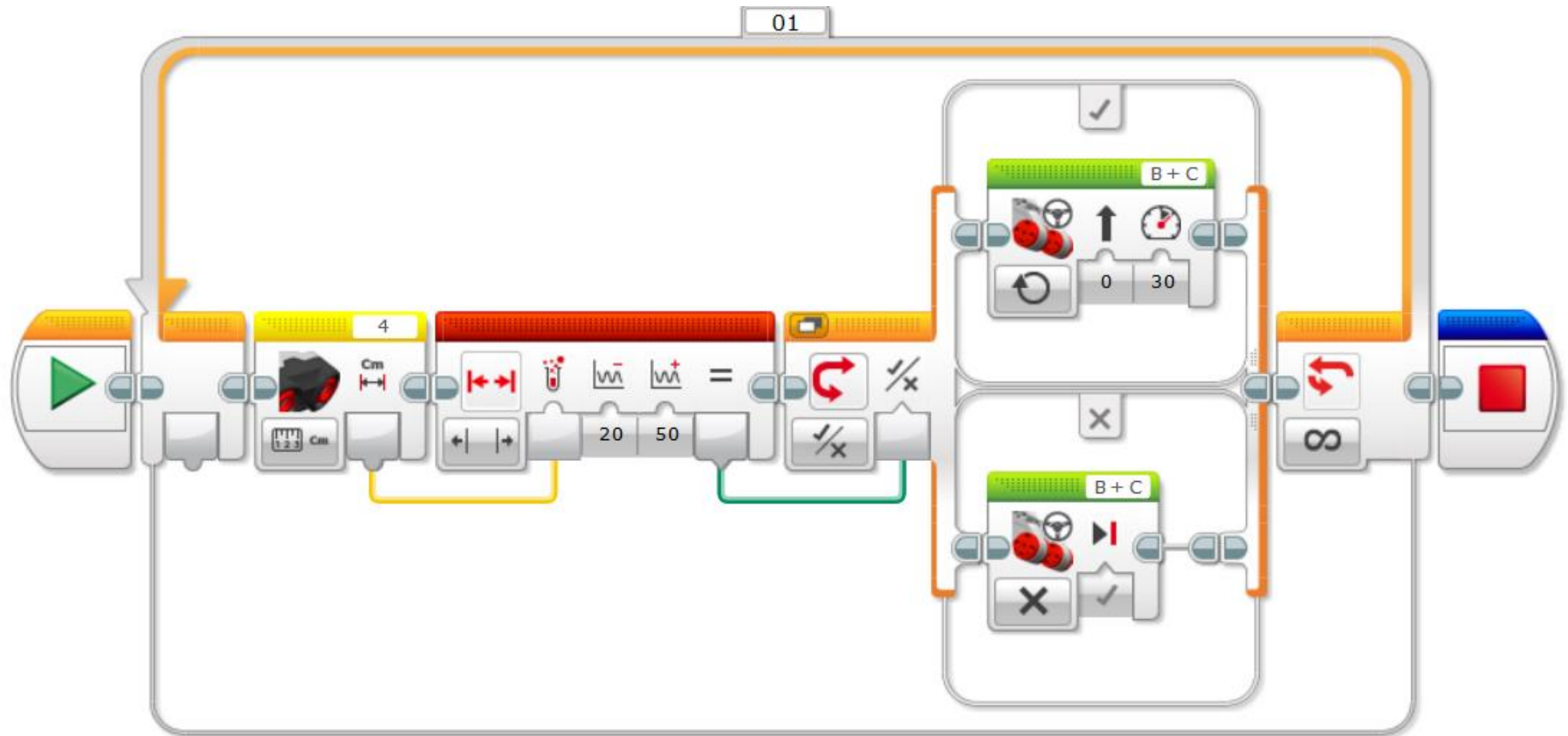
- távolságmérés
- a mért érték **figyelése**, **tartományon kívül** esik-e
- az előbbi **feltétel** ellenőrzése, igaz-e vagy hamis
  - **igaz**: a robot **halad előre** valamekkora sebességgel
  - **hamis**: **megáll**
- majd a teljes folyamat **ismétlése**

# GYAKORLAT II. - 1.

## EGYÉNI KOMPLEX FELADATMEGOLDÁS

### 2. Mozgó tárgy követése távolságérzékelővel:

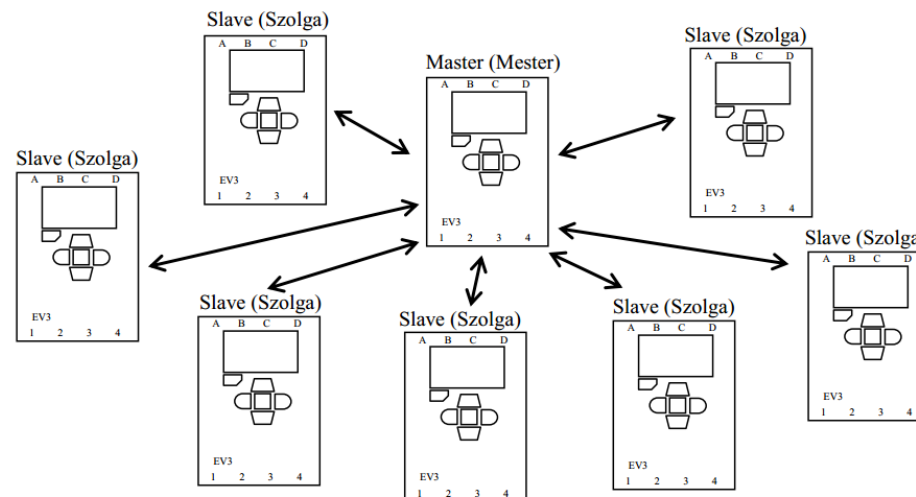
Megoldás



# GYAKORLAT II. - 2.

## ROBOTOK KÖZÖTTI KOMMUNIKÁCIÓ

- Célja: több robot együttműködése valamilyen feladat megoldása érdekében
- kommunikáció bluetooth kapcsolaton keresztül
- Master-slave (mester-szolga) alapú kommunikáció
  - az egymással kommunikációs kapcsolatban álló robotok között alá-fölé rendeltségi viszony
  - Master a kitüntetett szerepű, amelyen keresztül az adatok továbbítása történik
  - 1 master – max 7 slave -> 8 elemű BT alapú hálózat
  - a slave robotok kommunikációja csak a master-en keresztül történhet
  - egy robot vagy master vagy slave szerepkörű lehet



# GYAKORLAT II. - 2.

## ROBOTOK KÖZÖTTI KOMMUNIKÁCIÓ

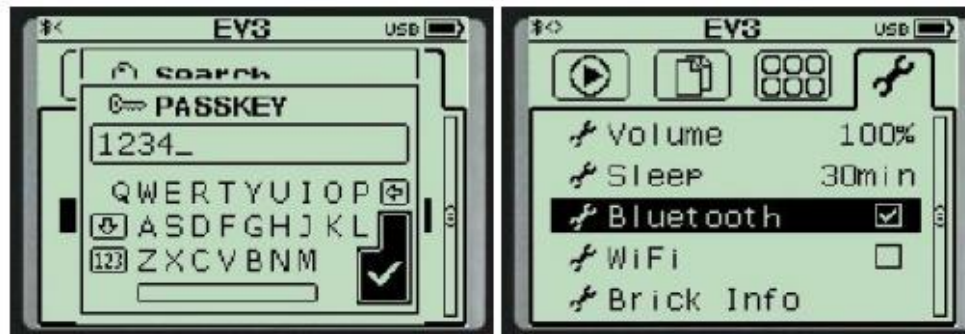
- Lépések:
  - **bluetooth kapcsolat felépítése**
    - a mesternek kijelölt roboton, a villáskulccsal jelölt (beállítások) menüben a Bluetooth menüpontban (vagy a programból). Bekapcsolt állapot jele: „<” a képernyő bal felső sarkában.
  - kommunikációra képes **eszközök keresése**
    - a mester robot képernyőmenüjének **Beállítások – Bluetooth – Connections – Search** menüpontjában.
    - szolgák kiválasztása a listából



# GYAKORLAT II. - 2.

## ROBOTOK KÖZÖTTI KOMMUNIKÁCIÓ

- az első kapcsolatépítésnél kódcserre történik, amely minden, a kapcsolatban résztvevő eszköz képernyőjén megjelenő kód beállítását és elfogadását jelenti.
- a felépült kapcsolatot a bal felső sarokban lévő ikon jelzi („<>”).



- Ha a kapcsolat felépült, akkor a rendszer megjegyzi a kódokat, és a továbbiakban a keresés helyett a Connections menüpontban is megjelenik az eszköz
- kapcsolatot programból is lehet kezdeményezni: Advanced -> Bluetooth Connections



# GYAKORLAT II. - 2.

## ROBOTOK KÖZÖTTI KOMMUNIKÁCIÓ

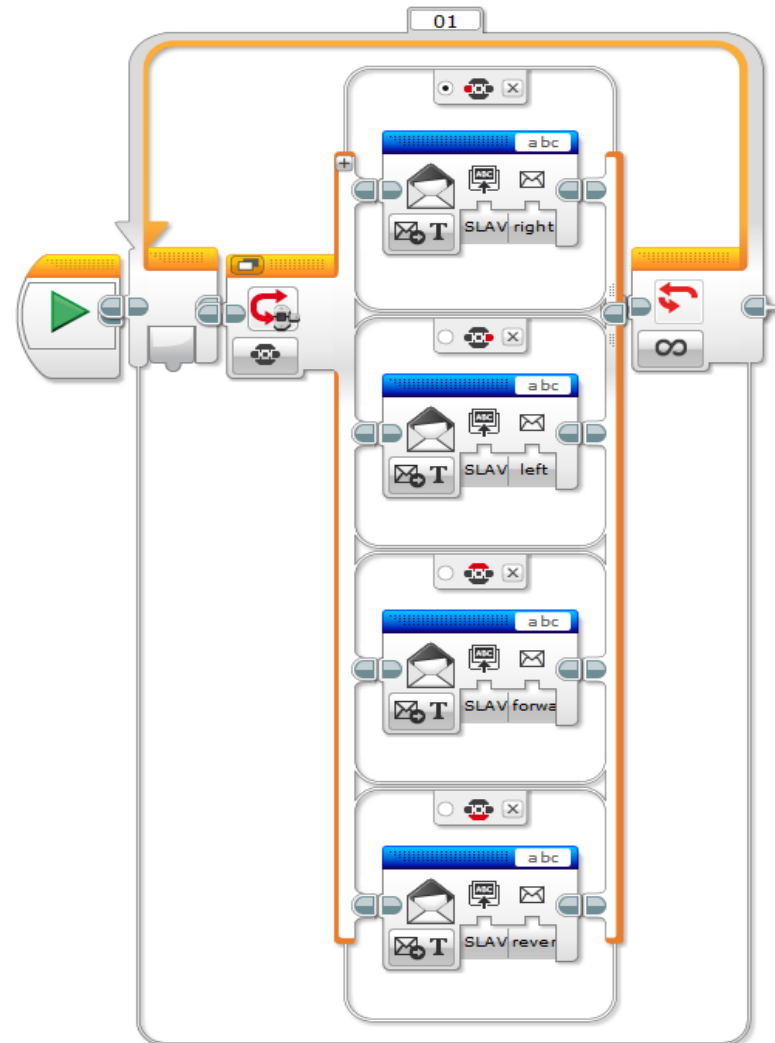
### 1. Készítsünk távirányítható robotot. Az irányító robot lesz a mester, az irányított a szolga.

#### Megvalósítás lépései:

1. A mester robot képernyőmenüjéből felépítjük a bluetooth kapcsolatot.
2. Megírjuk a mester robot programját és áttöltjük a robotra.
3. Megírjuk a szolga robot programját és áttöltjük a robotra.
4. Elindítjuk a mester robot programját, majd a szolga robot programját.

#### **A mester programja:**

- A mester téglá egy nyomógombjának lenyomása esetén a szolga az adott irányba mozduljon.

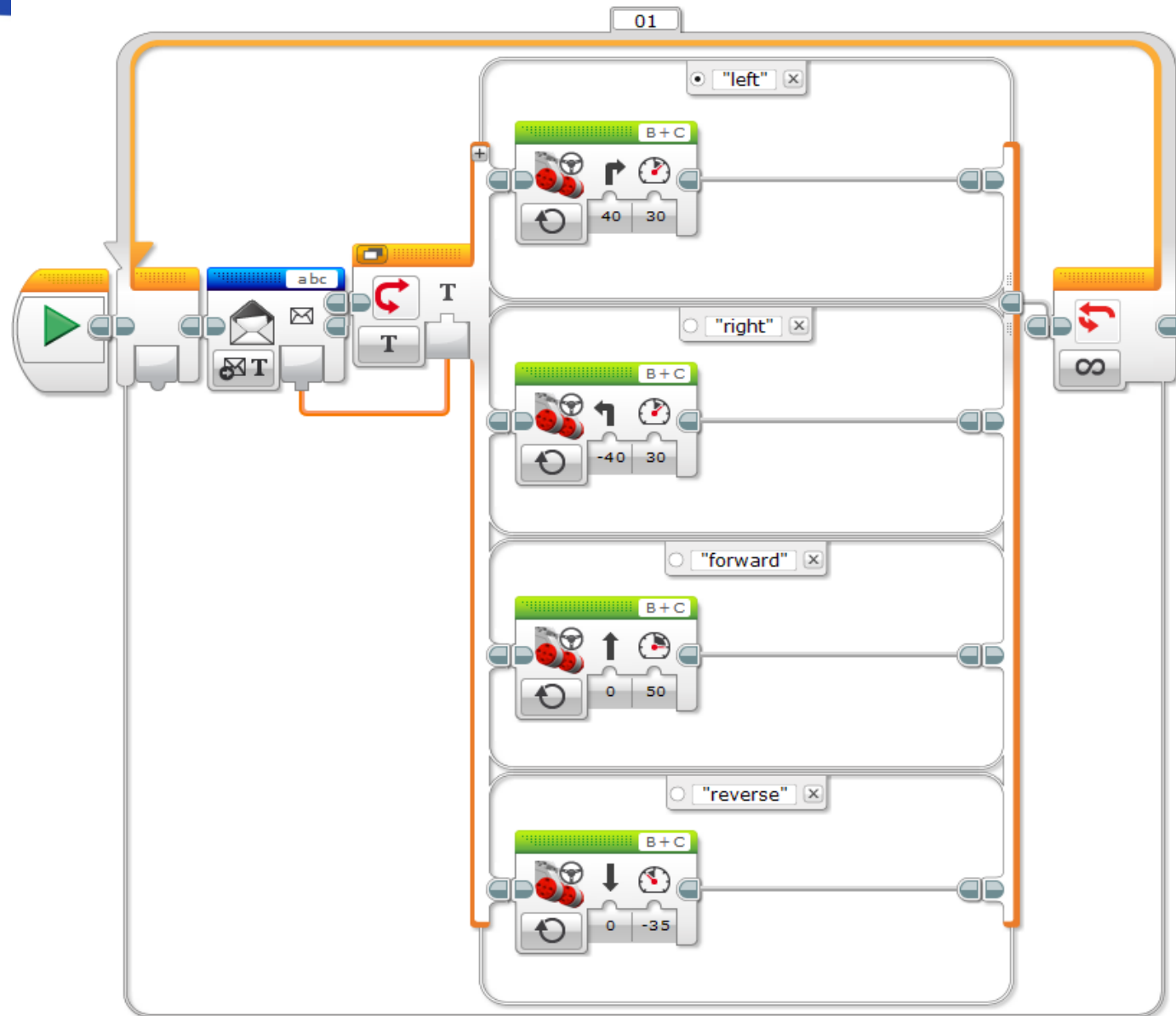


# GYAKORLAT II. - 2.

## ROBOTOK KÖZÖTTI KOMMUNIKÁCIÓ

**A távirányítással működtetett (szolga) robot programja:**

- a mestertől kapott üzenet alapján végrehajtja a megfelelő irányú mozgást





# GYAKORLAT II. - 2.

## ROBOTOK KÖZÖTTI KOMMUNIKÁCIÓ

2. **Írjon programot, amely két robot közötti kommunikációt valósít meg! A mester robot folyamatosan küldi a szolgának az ultrahangszenzora által mért értékeket, a szolga robot pedig megjeleníti ezt a képernyőjén. A programok kikapcsolásig fussanak!**

### Működés:

A mester robot programja egy végtelen ciklusba helyezett két utasításból áll.

Az ultrahang szenzor által mért adatot küldjük át a „Slave” nevű szolgának.

Mindezt folyamatosan, tehát végtelen ciklusba illesztve.

Az adatküldés modul (*Messaging*) paraméterénél *Number* típust kell beállítani.

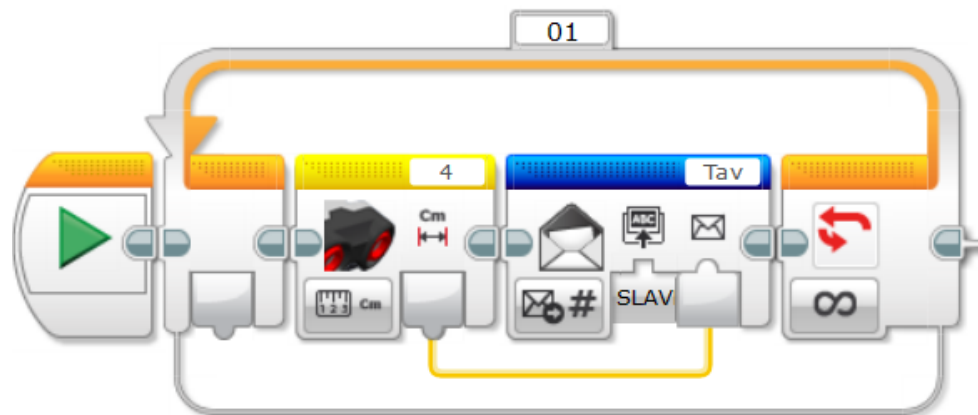
Az üzenet azonosítója „Tav”.

# GYAKORLAT II. - 2.

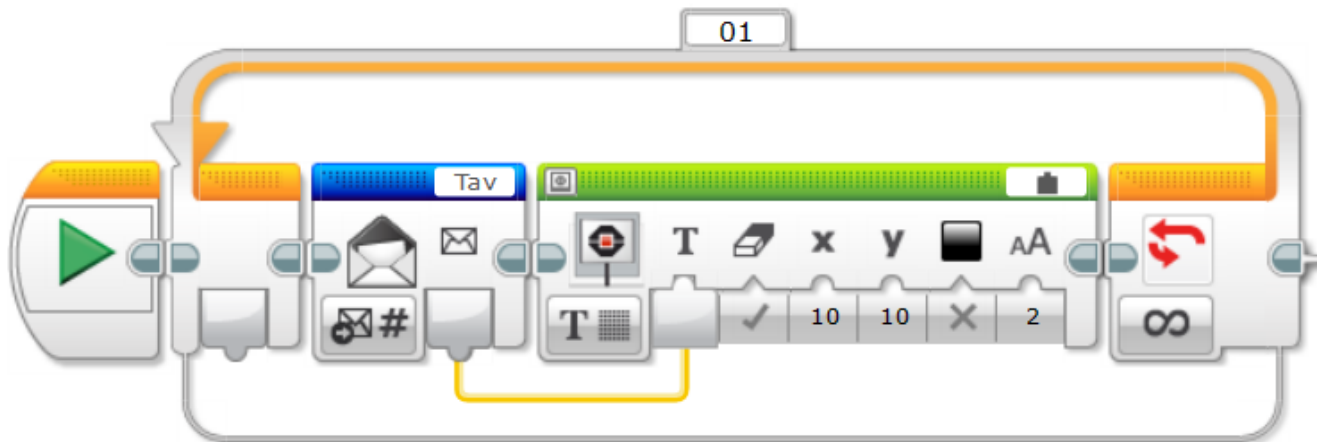
## ROBOTOK KÖZÖTTI KOMMUNIKÁCIÓ

Megoldás:

**Master:**



**Slave:**



# KÖSZÖNÖM A FIGYELMET!



**SZÉCHENYI** 



MAGYARORSZÁG  
KORMÁNYA

Európai Unió  
Európai Szociális  
Alap



**BEFEKTETÉS A JÖVŐBE**