

LEGO robotok és micro:bitek programozása MATLAB környezetben

Programming LEGO robots and micro:bits in the MATLAB environment

Dr. KOVÁCS Lehel István

Sapientia Erdélyi Magyar Tudományegyetem, Marosvásárhely
klehel@ms.sapientia.ro

Abstract

MATLAB is a modern, powerful programming environment that can be used to program LEGO robots, micro:bits and other microcontrollers. Being an interpreter, we can also write interactive programs, not just applications. In our study, we present the system settings and the M-code and Simulink level programming of the LEGO MINDSTORMS EV3 brick and micro:bit.

Keywords: MATLAB®, Simulink®, LEGO® MINDSTORMS® EV3, micro:bit, communication

Kivonat

A MATLAB korszerű, hatékony programozási környezet, amellyel LEGO robotokat, micro:biteket és más mikrovezérlőket is lehet programozni. Interpreter lévén interaktív programokat is írhatunk, nemcsak letölthető alkalmazásokat. Tanulmányunkban a rendszer beállításait, valamint a LEGO MINDSTORMS EV3 téglát és a micro:bit M-code és Simulink szintű programozását mutatjuk be.

Kulcsszavak: MATLAB®, Simulink®, LEGO® MINDSTORMS® EV3, micro:bit, kommunikáció

BEVEZETŐ

A MATLAB numerikus számítások elvégzésére alkalmas speciális programrendszer és egyben programozási nyelv, amelyet a MathWorks fejleszt. A programrendszer képes mátrix számítások elvégzésére, függvények és adatok ábrázolására, algoritmusok implementációjára és felhasználói interfészek kialakítására.

A MATLAB-ot (jelentése: „**matrix laboratory**”) az 1970-es évek elején Cleve Moler kezdte el fejleszteni, az akkori Új-Mexikói Egyetem Számítástudományi Intézetének elnöke.

A MATLAB a MuPAD csomag hozzáadásával képes matematikai kifejezéseket grafikusán is megjeleníteni. [1]

A MATLAB, tehát, egy hatékony interpreter, amelyben előre megírt függvények és eszköztárak segítik a felhasználót a gyors algoritmusfejlesztésben és az adatok megjelenítésében.

A Simulink egy grafikus környezet több tartományú dinamikus és beágyazott rendszerek szimulációjához és modell alapú tervezéséhez. A MathWorks közel 100 további terméket fejlesztett ki olyan speciális feladatokhoz, mint az adatelemzés, képfeldolgozás és mesterséges intelligencia.

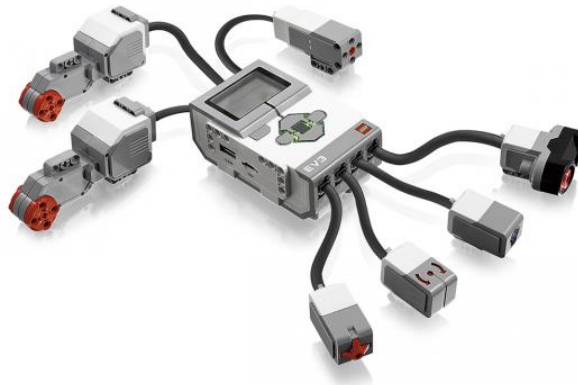
A LEGO Mindstorms EV3 harmadik generációs LEGO robot. 2013 szeptemberében e termékcsalád megjelenítésével ünnepelte a népszerű Mindstorms játék- és oktatóeszköz tizenötödik születésnapját a LEGO.

A Mindstorms EV3 a korábbi modellnél gyorsabb processzort (egy ARM9 alapú 32 bites RISC processzort, amin *Linux* fut) és nagyobb memóriát kapott, így a rendszer lelkét képező *téglára* (brick) előre megírt programok segítségével komolyabb feladatok is bízhatók.

A LEGO Mindstorms EV3 robot önálló életet élhet, elszakadhat a programozásához használt számítógéptől, így fontos szerephez jut a mobil kommunikációs eszközökkel való szorosabb együttműködés.

A *Linux* alapú firmware-nek, az SD kártyaolvasónak és az USB-portnak köszönhetően a LEGO Mindstorms EV3 téglát tetszőlegesen újra programozható, így a bővíthetőségnek és az alakíthatóságnak szinte csak a fantázia szab határt.

A robotok programozására az érzékelők szolgáltatnak bemenetet (*infravörös érzékelő, ultrahangos érzékelő, giroszkópikus érzékelő, színérzékelő, érintésérzékelő*), és a kimenet vagy a motorokat (*nagy motor, közepes motor*) vezényli, vagy valamilyen más funkcióját a téglának (*kijelző, gombok, hangfal, portok és csatlakozók stb.*). [2]

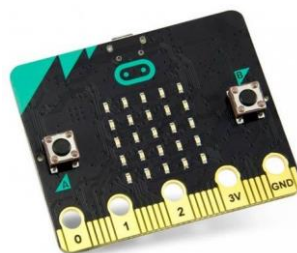


2. ábra. Az EV3-tégla motorokkal és érzékelőkkel

A BBC micro:bit (3. ábra) egy kifejezetten oktatási célra létrehozott, egylapkás mikrovezérlő, amely 4×5 cm-es méretével, 5×5-ös LED-mátrix kijelzőjével, gyorsulásérzékelő, hőmérséklet érzékelő, fény-érzékelő, irány érzékelő szenzoraival, be- és kimeneti csatlakozóival, 2 gombjával, bluetooth/rádió kapcsolódási lehetőségével igen sokrétű alkalmazást tesz lehetővé, legyen az (akár többfelhasználós) játék fejlesztése, viselhető eszközök (pl. okosóra, lépésszámláló, okosruha) tervezése és megvalósítása, kísérletezés a szenzorok által mért adatok felhasználásával, vagy éppen külső eszközök vezérlése/irányítása. Mivel az eszköz egy mikrovezérlő, ezért a programozásához szükséges egy számítógép (asztali, notebook, vagy akár tablet és okostelefon), amelyhez vagy USB kábellel, vagy Bluetooth kapcsolaton keresztül kapcsolódhatunk. PC-ről vagy mobilról is elérhető web-es felületen [3] írhatunk programokat, amelyek USB-n vagy akár bluetooth-on keresztül tölthetők fel az eszközre. A programot egy-szerűen fel kell másolni a micro:bit virtuális meghajtójára és már működni is kezd [4].

A micro:bit 25 külső csatlakozóval rendelkezik a kártya élcsatlakozóján (peremcsatlakozóján), amelyeket *pin*eknek nevezünk.

Első (V1) és második (V2) verziója ismert, a V2 saját hangjelzővel, mikrofonnal, még egy gombbal, nyolcszor nagyobb memóriával hoz többet régebbi társánál.



3. ábra. BBC micro:bit

1. MATLAB ÉS LEGO ROBOTOK

1.1. A rendszer beállítása

Legegyszerűbb, ha a MathWorks termékek, például a MATLAB, a Simulink WiFi-n keresztül kapcsolódik az EV3 téglához, így maximális mobilitást érhetünk el a robotunk számára.

Szükségünk lesz tehát egy WiFi routerre, amelyen a titkosítás WPA2-re legyen beállítva, az IP-címek pedig legyenek dinamikus kiosztásúak (DHCP).

Ellenőrizzük le az EV3-as tégla firmware verzióját! Ahhoz, hogy a rendszer működjön az EV3 firmware V1.03E vagy későbbi szükséges.

A beállítások fülecskén válasszuk ki a Brick Info menüpontot, majd nézzük meg a Brick FW-t. Ha ez kisebb, mint az említett verzió, akkor cseréljük le a firmware-t.

A firmware cseréjére látogassunk el a <https://education.lego.com/en-us/product-resources/mindstorms-ev3/downloads/firmware-update> honlapra, kapcsoljuk össze a téglát USB kábellel a számítógéppel, majd nyissuk meg az EV3 Device manager honlapot. A rendszer felszólít, hogy töltsük le és telepítsük a Device Managert.

A „H” végződésű firmware a Home, míg az „E” végződésű az Education kiadás firmware-je. Bármelyiket feltehetjük a téglára, de nem mindegy, hogy melyik szoftverrel. A LEGO MINDSTORMS Education EV3 mindegyiket tudja kezelni, a LEGO MINDSTORMS EV3 Home Edition csak a „H” végződésűeket.

Az EV3 firmware V1.09E-ből és a EV3 firmware V1.09H-ből kivették a telnet-et, így maximum a V1.08-at használjuk! Létezik egy speciális „D” végződésű firmware is, amely a telnet használatát jelszóhoz köti. Ha telnettel jelentkezünk rá a téglára (PuTTY, SSH), akkor a felhasználónév root, a jelszó pedig üres.

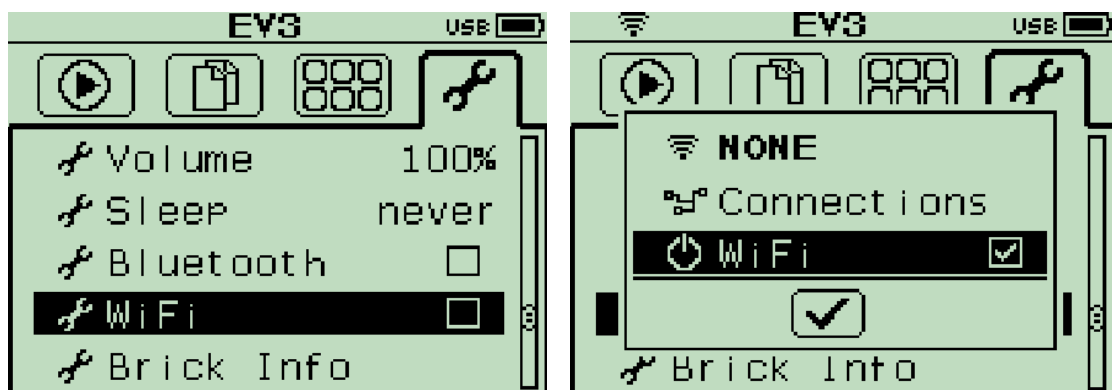
Az EV3-as téglá WiFi beüzemeltetéséhez szükségünk van egy USB-s WiFi adapterre. A LEGO a Netgear WNA1100 – N150 Wireless USB adaptert ajánlja, amely meglehetősen régi modell, így napjainkban elég nehezen szerezhető be (4. ábra).



4. ábra. Netgear WNA1100 – N150 Wireless USB adapter

Tegyük be az adaptert a téglá USB portjába, majd kapcsoljuk be a LEGO MINDSTORM EV3-as téglát. Menjünk a beállítások földre (utolsó fül), és válasszuk ki a WiFi-t (5. ábra).

Először engedélyezzük a WiFi-t (5. ábra), a folyamat eltarthat pár másodpercig, majd a Connections menüponttal keressünk rá a WiFi-hálózatokra (ez is eltarthat pár másodpercig).



5. ábra. WiFi beállítása az Ev3 téglán

Válasszuk ki a WiFi-hálózatunkat, a titkosítási mód legyen WPA2, majd írjuk be a jelszót.

Pár másodperc múlva visszakapjuk a kezdőképernyőt, a csatlakoztatott hálózat ki lesz pipálva. Zárjuk be az ablakokat, a téglánk a WiFi-hálózaton van.

Ezután a Brick Info menüpontban keressük meg az IP Address-t, és írjuk le, mert erre az IP-címre szükségünk lesz. Ez itt most: 192.168.0.198. Hasonlóan a téglá azonosítójára is, ez itt most: 0026F247E3C6.

A MATLAB-bal (R2021a vagy későbbi kiadás) való összekapcsolásra szükségünk van a Simulink Support Package for LEGO MINDSTORMS EV3 Hardware csomagra, illetve egy MathWorks fordítóprogramra is. [5]

Indítsuk el a MATLAB-ot, majd a főmenü alatti eszköztárból válasszuk ki az Add-Onst, majd a Get Hardware Support Package-est.

Megjelenik az Add-On Explorer, innen pedig hardverelemeket választhatunk ki (az Arduinohoz, drónokhoz, Kinecthez is itt találhatóak a szükséges szoftverek). Telepítsük az említett csomagot, ehhez meg kell adnunk az e-mail címünket és a jelszavunkat (be kell jelentkeznünk a rendszerbe) a licenz miatt.

A telepítés eltarthat pár percig, hisz pár komponens fog felmásolni. A PixyMon V2-vel LEGO blokkokat és példákat is telepíthetünk, ezeket külön kell letölteni.

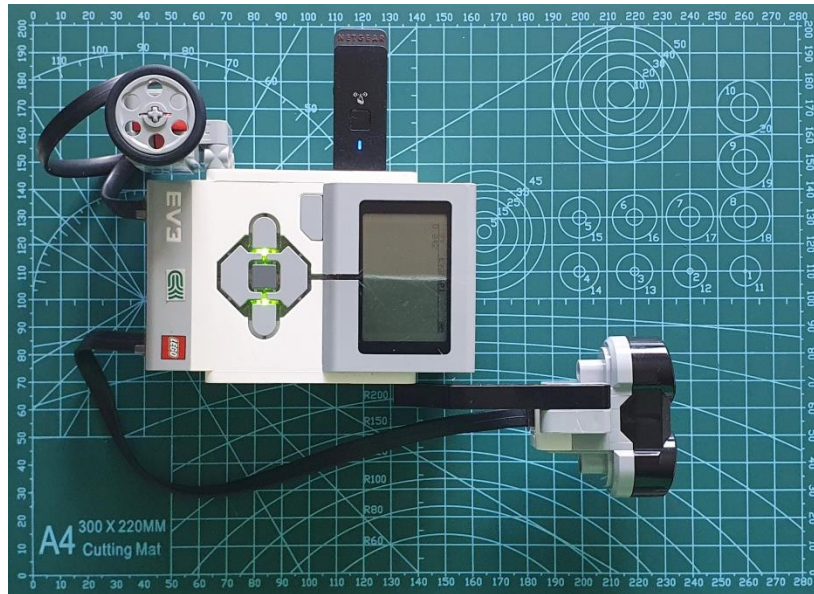
Fordítóprogramként a MinGW-t telepítettük ugyaninnen.

Ha ezzel megvagyunk, próbáljuk ki a fordítóprogram működését, írjuk be a parancssorba a mex -setup parancsot, ekkor megjelenik, hogy a MinGW64 az alapértelmezett fordító.

1.2. LEGO robotok programozása MATLAB-ban

Ha az előző alfejezetben szereplő beállításokat kieszköztük, akkor megírhatjuk LEGO EV3 robotra az első MATLAB programunkat, vagyis M-kódunkat (M-code).

Az EV3 téglára tegyük egy ütközésérzékelőt, a 4-es portjára pedig egy ultrahangos érzékelőt. Így létre tudjuk hozni a 6. ábrán szereplő távolságmérőt, amelyet nemcsak távolság, hanem például sebesség mérésre is használhatunk.



6. ábra. Távolságmérő

A MATLAB erőssége nemcsak az, hogy nagyon egyszerűen kapcsolódhatunk az EV3 téglához, hanem az is, hogy az érzékelőkről származó adatokat valós időben és folyamatosan tudjuk lekérdezni a LEGO EV3 téglától, így azokat meg is tudjuk jeleníteni úgy, hogy különösen nem kell törődnünk a kommunikáció megvalósításával egy külön kommunikációs modul írásával vagy azzal, hogy a programunkat letöltsük az EV3 téglára és az onnan küldözgesse valamilyen protokoll szerint az adatokat, mint azt tennék más EV3 nyelvek esetében. Nem kell tehát külön program a számítógépre, külön program az EV3 téglára.

Az első utasításunk a kapcsolat felvétele az EV3 téglával, ide szükséges az IP cím és az említett azonosító. Az utasítás eredménye egy objektum, amely a téglát jelképezi.

```
ml = legoev3('wifi', '192.168.0.198', '0026F247E3C6');
```

Ugyanezzel az utasítással lehet beállítani a kapcsolat típusát is. Ha nem WiFi-n keresztül szeretnénk kommunikálni, akkor lehetőségünk van Bluetooth vagy USB kapcsolat megteremtésére is a `ml = legoev3('bluetooth', <Serial_Port>)` vagy `ml = legoev3('usb')` parancsok segítségével.

Az `ml` objektumon keresztül érhetjük el a téglák funkcióit, például a képernyőjét, az érzékelőket stb., amelyekre külön objektumokat hozhatunk létre és így hivatkozhatunk rájuk:

```
ms = sonicSensor(ml, 4);
mt = touchSensor(ml, 1);
```

A zárójelben szereplő második paraméter (4 és 1) a csatlakozó portok azonosító száma.

A LEGO EV3 MATLAB függvényeit megtalálhatjuk a [8] oldalon. Ez a MATLAB hivatalos súgója, példákat is találhatunk mind M-code, mind pedig Simulink alkalmazásokra.

Legyen a feladat az, hogy folyamatosan, 0,1 szekundumos időközönként leolvassuk a távolságot, és ezt megjelenítjük egy grafikonon. Mivel az időt is mérjük, nagyon könnyen sebességet is mérhetünk, ábrázolhatunk a sebesség = megtett út / idő képlet szerint.

A teljes MATLAB program a következő:

```

ml = legoev3('wifi', '192.168.0.198', '0026F247E3C6');
ml.beep;
ml.clearLCD;
ml.writeLCD('ready...' , 1, 1);
ms = sonicSensor(ml, 4);
mt = touchSensor(ml, 1);

while(readTouch(mt) == 0)
end

Tmax = 100; % sec
stepsize = 0.1; % sec
figure
xlim([0 Tmax])
H = plot(0, 0);

tstart = clock;

while(etime(clock, tstart) <= Tmax)
    ml.clearLCD;
    newData = ms.readDistance;
    H.YData = [H.YData newData];
    H.XData = [H.XData H.XData(end) + stepsize];
    ml.writeLCD(num2str(newData), 1, 1);
    pause(stepsize);
    drawnow;
end

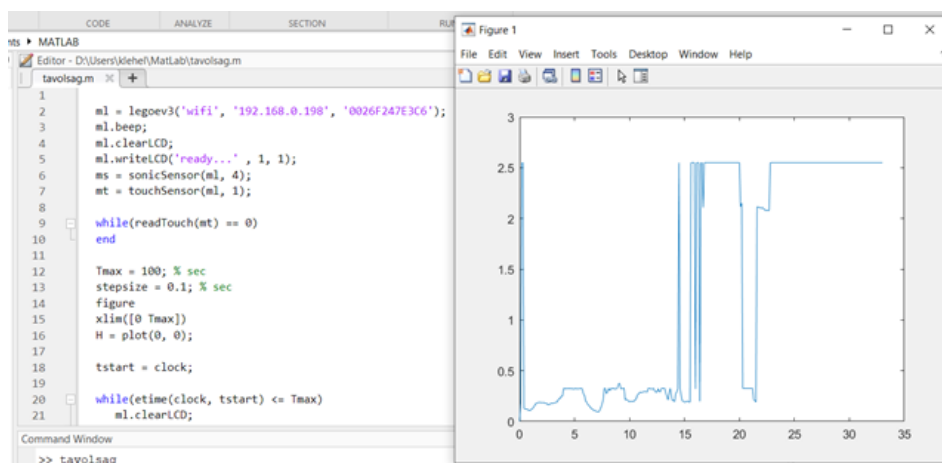
ml.beep;
ml.clearLCD;

delete(ms);
delete(mt);
delete(ml);

```

Amint észrevehetjük, a programban felváltva kezelhetjük interaktívan a számítógépet és az EV3 téglát, mindkettőnek adhatunk utasításokat, olvashatunk be adatokat. A MATLAB nagyon megkönnyíti a kommunikációt, az adatfolyamot.

A 7. ábrán a program működését láthatjuk MATLAB környezetben.



7. ábra. A program működése

1.3. LEGO robotok és a Simulink

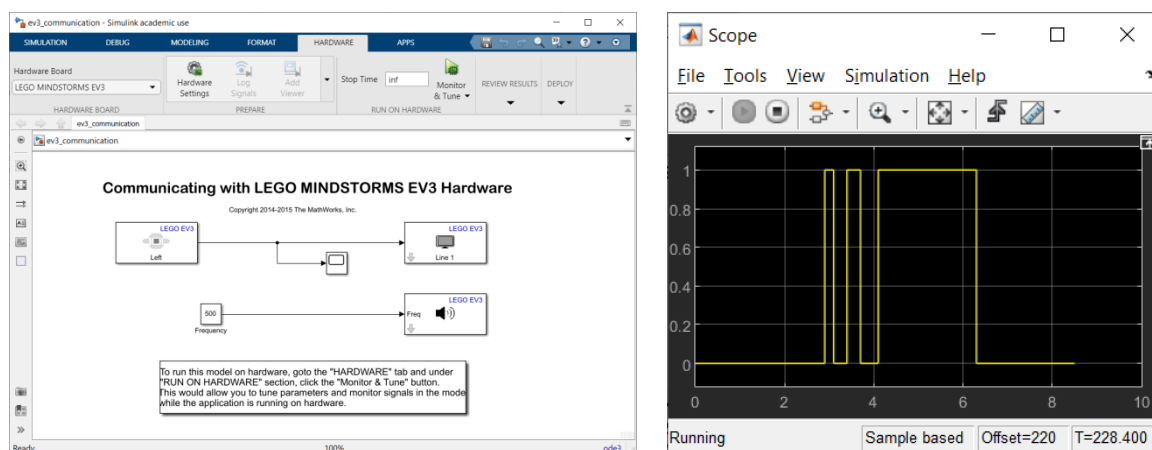
A Simulink egy MATLAB alapú grafikus programozási környezet többtartományú dinamikus rendszerek modellezésére, szimulálására és elemzésére. Elsődleges felülete egy grafikus blokkdiagram-készítő eszköz és egy testre szabható blokkkönyvtár-készlet.

A LEGO EV3-as könyvtárat elérhetjük a legoev3lib parancs segítségével.

Az 1.1. alfejezetben bemutatott beállítások szerint, első lépésben indítsuk el a Simulink kommunikációs modellt, vagyis adjuk ki az ev3_communication parancsot. Természetesen lehetőségünk van új kommunikációs modell létrehozására is.

Gyakorlatilag a Simulink abban segít, hogy grafikus felületen programozhatjuk az EV3 téglánkat, blokkok, diagramok segítségével.

A 8. ábrán látható beépített alap kommunikációs modell szerint az EV3 téglá bal nyomógombjának használatát, valamint egy hanggenerátort célzunk meg. A scope grafikon mutatja, hogy megnyomtuk-e, illetve mennyit tartottuk lenyomva a bal nyomógombot, illetve az EV3 téglá hangfala egy 500 Hz-es hangot szólaltat meg.



8. ábra. Simulink és LEGO EV3

2. MATLAB ÉS MICRO:BIT

2.1. A rendszer beállítása

Amint a LEGO EV3 téglá számára, úgy a micro:bit részére is be kell állítani a rendszert, bővítményeket kell letölteni, aktualizálni kell a micro:bit firmware-jét.

A MATLAB környezetben kattintsunk az Add-Ons menügombra, majd a megjelent ablak keresőjébe írjuk be hogy „micro:bit”. A pár megjelent bővítmény közül válasszuk ki a Simulink Coder Support Package for BBC micro:bit Board-ot és telepítsük. A telepítéshez meg kell adnunk felhasználó nevünket és jelszavunkat, hogy a licenz érvényes legyen.

A telepítés után a rendszer tutorial videókat is a rendelkezésünkre bocsát.

A micro:bit firmware cseréje érdekében látogassunk el a <https://os.mbed.com/daplink-on-kl26z/> honlapra.

Tudni kell, hogy ha USB-kábellel csatlakoztattuk a micro:bitet a számítógépünkhöz, akkor egy új lemezegységként jelenik meg, amelynek MICROBIT a neve. Ha áthúzzuk erre a lemezegységre a lefordított micro:bit programunkat (.HEX állomány), akkor az már futni kezd.

A firmware cseréhez a micro:bitet karbantartás üzemmódba kell helyezni, amit úgy érhetünk el, hogy kihúzzuk az USB-kábelt, lenyomjuk a micro:biten lévő RESET gombot, és azt lenyomva tartva tesszük vissza a számítógép USB-portjába, majd felengedjük a RESET gombot. Ekkor a MICROBIT nevű meghajtó helyett egy MAINTENANCE nevű meghajtó jelenik meg, és erre kell másolnunk a letöltött firmware-t.

A karbantartási módból való kilépéshez húzzuk ki a micro:bit kártyát az USB-csatlakozóból, győződjünk meg arról, hogy nincs megnyomva a RESET gomb, majd dugjuk vissza az USB-kábelt. [7]

Kiválasztva a micro:bitünk verziószámát (V1 vagy V2), töltsük le a legutolsó firmware verziót, jelen esetben nekünk ez a 0249_kl26z_microbit_0x8000.hex állomány.

Menjünk át karbantartási üzemmódba és másoljuk az állományt a micro:bitre. Ezzel a rendszer üzemképesé vált.

Az 1.1. alfejezetben említettek szerint itt is hasznos a fordítóprogram ott leírt telepítése, hisz a rendszer .HEX állományt állít elő, amelyet felmásol a micro:bitre. Ez a bináris állomány tartalmazza a micro:bit által futtatható kódot. Ez mind a MATLAB, mind pedig a Simulink számára fontos.

Megjegyzendő, hogy a MATLAB, a Simulink más mikrovezérlőket is tud kezelni, mint például az Arduino, Raspberry Pi, BeagleBone Black, BeagleBone Blue, STMicroelectronics Nucleo, NXP, VEX EDR V5 stb. Ezek programozására is számos függvénykönyvtár, modul létezik.

2.2. A micro:bit programozása MATLAB-ban

Ha az előző alfejezetben szereplő beállításokat kieszközöltük, akkor megírhatjuk a micro:bitre az első MATLAB programunkat.

Az első utasításunk a kapcsolat felvétele a micro:bittel. A számítógépünk eszközezőrlőjében (Device Manager) nézzük meg, hogy az operációs rendszer melyik kommunikációs portot osztotta ki az USB-kábellel összekapcsolt micro:bitnek. Jelen esetben ez a COM3, amelyet megadunk paraméterként. Az utasítás eredménye egy objektum, amely a micro:bitet jelképezi.

```
ml = microbit('com3');
```

A micro:bit MATLAB függvényeit megtalálhatjuk a [9] oldalon. Ez a MATLAB hivatalos súgója, példákat is találhatunk mind M-code, mind pedig Simulink alkalmazásokra.

Itt már nincs annyi lehetőség, mint a LEGO EV3 téglá esetében, a micro:bithez sokkal több érzékelőt kapcsolhatunk, de ezekhez alapszinten, digitális vagy analóg pinek írásával olvasásával vagy SPI, illetve I²C buszok írásával, olvasásával tudunk hozzáférni. Így a micro:bit MATLAB függvényeinek nagy része ezekről szól. A függvénykönyvtár 19 függvényt tartalmaz, Simulink blokkokban jóval több a lehetőség.

Alapérzékelőkként csak a gyorsulásmérő (Accelerometer), illetve a mágnesesség érzékelő (Magnetometer) szerepel függvények szintjén.

Legyen a feladat az, hogy felvesszük a kapcsolatot a micro:bittel a COM3 porton, töröljük a LED mátrixot, felgyűjtjük a második oszlop második LED-jét, kirajzoltatjuk a micro:bit pin-kiosztását, majd folyamatosan, 0,1 szekundumos időközönként leolvassuk az „A” gomb állapotát (5. pin), és ezt megjelenítjük egy grafikonon.

A teljes MATLAB program a következő:

```
ml = microbit('com3');

clearLED(ml);
writeLED(ml, 7);
showPins(ml);

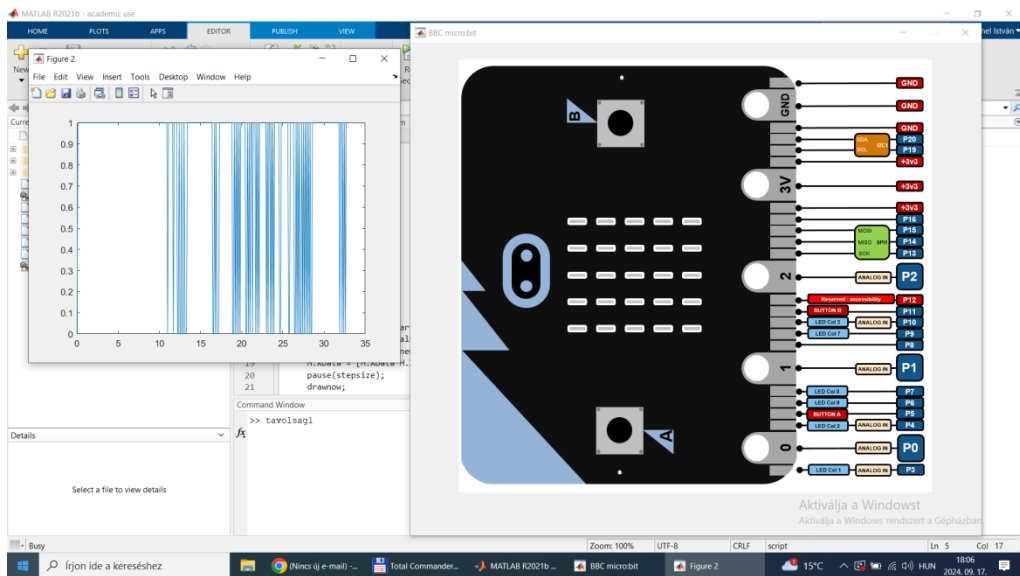
Tmax = 100; % sec
stepsize = 0.1; % sec
figure
xlim([0 Tmax])
H = plot(0, 0);

tstart = clock;

while(etime(clock, tstart) <= Tmax)
    newData = readDigitalPin(ml, 'P5');
    H.YData = [H.YData newData];
    H.XData = [H.XData H.XData(end) + stepsize];
    pause(stepsize);
    drawnow;
end

delete(ml);
```

A 9. ábrán a program működését láthatjuk MATLAB környezetben. A diagram az „A” gomb lenyomását mutatja digitális értéként (0 vagy 1). A másik ablakban a pinek kiosztása jelenik meg, ahogy a MATLAB kirajzolja.



9. ábra. Simulink és micro:bit

2.3. A micro:bit és a Simulink

Simulinkben is készíthetünk egy modellt a micro:bit számára, itt most egy nagyon egyszerűt mutatunk be, hogy ismertessük a lépéseket.

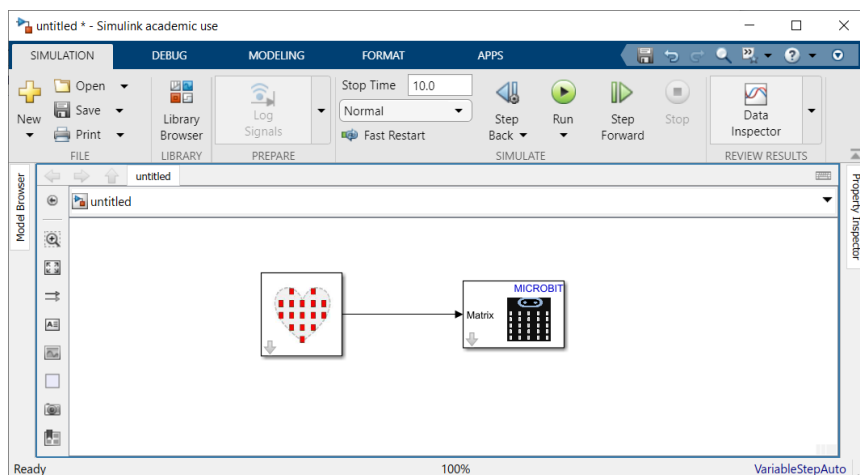
Ha a 2.1. alfejezetben leírtak szerint telepítettük és beállítottuk a rendszert, akkor nem is maradt más hátra, mint hogy a MATLAB környezetből, a Simulink menügombra kattintva elindítsuk a Simulink felületet. Válasszunk egy üres modellt (Blank Model), így egy tiszta lapú felületet kapunk.

A Library Browser segítségével megtekinthetjük a telepített könyvtárakat. Számunkra most a Simulink Coder Support Package for BBC micro:bit a fontos, amelyet a 2.1. alfejezetben leírtakkal összhangban telepítettünk. Itt található az összes micro:bit számára írt modul, funkcionalitásuk szerint csoportosítva. Nem kell mást tennünk, mint kiválasztani a kívánt blokkot és ráhúzni a felületre, majd a funkcionalitás függvényében összekötni ezeket.

Minket most a LED könyvtár érdekel. Húzzuk a felületre innen az 5x5 LED Matrix blokkot, majd keressük ki a megjeleníteni kívánt LED-ikont. Legyen ez most a szív. Ezt is húzzuk a felületre, majd kössük össze a két blokkot a 10. ábra szerint! A szív ikon a kimenet, a LED Matrix a bemenet, de ez kiderül a nyilak irányából is.

Ahhoz, hogy a vizuálisan létrehozott programot futtatni tudjuk a micro:biten, a Modeling menüben be kell állítanunk a Model settings ablakban a Hardware Implementation lehetőségeket. A Hardware boardnál válasszuk ki a BBC micro:bit beállítást, majd várjuk meg, míg a rendszer áttér erre a kártyára.

Nem maradt más hátra, mint elindítani a fordítóprogramot, hogy elkészítse a .HEX futtatható állományt, hogy ezt a micro:bitre másolva futtatni tudja a rendszer. Ezt a Monitor & Tune menügombbal tehetjük meg, és az eredmény a micro:biten megjelent szívecske lesz.



10. ábra. Simulink és micro:bit

3. KÖVETKEZTETÉSEK

A matematikai pontosság mellett nem hanyagolhatjuk el a MATLAB didaktikai, módszertani szerepét sem. Egy teljesen másfajta paradigma bontakozik ki előttünk, mégpedig a könnyedén megvalósítható interaktív kommunikáció paradigmája. Természetesen továbbra is lehetőség van arra, hogy az EV3 vagy más mikrovezérlőkkel ellátott robotokra letölthető, önállóan futtatható alkalmazásokat készítsünk, de a MATLAB az interaktív, gyors, más segédeszköz nélküli adatforgalmat, kommunikálást is biztosítani tudja, mi több, az adatokat precíz grafikonokon, ábrákon tudjuk megjelteni, így a tudományos aspektus mellett a szemléletességre is hangsúlyt fektethetünk.

A MATLAB nagyon hasznos eszköz azon diákok és oktatók számára, akik a tudományos számítások, a mérnöki tervezés, az adatelemzés, a szimulációk és a modellezés terén dolgoznak.

A MATLAB lehetővé teszi az adatelemzési folyamat egyszerűsítését és automatizálását, amely magában foglalja az adatok előfeldolgozását, a statisztikai elemzéseket és a modellezést is.

A MATLAB grafikus funkciói segítségével különféle diagramokat, táblázatokat és grafikonokat hozhatunk létre az eredmények megjelenítéséhez.

A MATLAB egy nagyon rugalmas programozási nyelv (M-code), amely lehetővé teszi a felhasználók számára a különféle alkalmazások megvalósítását, akár robotokra is. A MATLAB támogatja a strukturált programozást, az objektumorientált programozást és az egyéb fejlett programozási technikákat.

A MATLAB nagy előnye az is, hogy rengeteg dokumentáció és online forrás áll rendelkezésünkre, amelyek segítségével fejleszthetjük ismereteinket és készségeinket. [6]

IRODALMI HIVATKOZÁSOK

- [1] <https://hu.wikipedia.org/wiki/MATLAB>
- [2] <http://education.lego.com/es-es/products>
- [3] <https://makecode.microbit.org/>.
- [4] <https://github.com/JorgePe/microbit>.
- [5] <https://www.mathworks.com/hardware-support/lego-mindstorms-ev3-matlab.html>
- [6] <https://uni-obuda.hu/2023/05/09/matlab-a-tanulashoz-kutatashoz-oktatashoz/>
- [7] <https://os.mbed.com/daplink-on-kl26z/>
- [8] https://www.mathworks.com/help/matlab/referencelist.html?type=function&listtype=cat&category=legomindstormsev3io&blocktype=all&capability=&startrelease=&endrelease=&s_tid=CRUX_lftnav
- [9] https://www.mathworks.com/help/rtw/working-with-matlab-functions-microbit.html?s_tid=CRUX_lftnav