

A homályosságot úgy küszöböljük ki, hogy sok kockát exponálunk ugyanarról az objektumról, minél többet, annál jobb, és a képfeldolgozás során ezeket egy speciális asztroszoft segítségével kiátlagoljuk, csak a konstans módon megjelenő képelemeket hagyjuk meg. Ily módon a végső fotón szép éles csillagok és tiszta kontúrok jelennek meg. Ha jó minőségű, részletgazdag képet akarunk elérni, minimum 20-25 képkockára van szükségünk, ez természetesen függ az ábrázolt objektumtól is – minél halványabb, annál hosszabb expókra és annál több képkockára lesz szükségünk. Leghosszabban rögzített képemhez 160 darab 10 perces képkockát készítettem, ez több mint 27 órát tesz ki összesen, amit nyolc éjszakán át gyűjtöttem be.

A szenzorban keletkező képzet kiküszöbölésére ún. sötétképeket készítünk. Ez azt jelenti, hogy teljesen lezárt fényutakkal (távcsőre felhelyezett sapka, objektívra felhelyezett kupak) készítünk ugyanolyan ISO-val ugyanolyan hosszú expókat – általában 4-5 darabot – és ezeket a sötétképeket digitálisan kivonjuk mindenik képkockából, még átlagolásuk és összeadásuk előtt.

A képeket mindig RAW-formátumban rögzítjük, hogy végül a teljes információ mennyiség rendelkezésünkre álljon, ne veszítsünk el semmit belőle a rögzítés során.

#### *A képek digitális feldolgozása*

A képek feldolgozásának elmélete és gyakorlata roppant komplex számítástechnikai feladat, aminek a taglalása messze meghaladja jelen írásom kereteit. Itt csak nagy vonalakban utalok a lényegre. Aki nekivág ennek a feladatnak, a szoftver betanulása során majd lépésről lépésre megérti az elméletet is, Így jártam jómagam is.

A használt programok száma nap mint nap növekszik, egyre-másra jelennek meg jobbnál jobb asztroszoftok, mégis mindig egy régebbi programot ajánlok mindenkinek: nekem az IRIS névre keresztelt nyílt forráskódú, ingyen letölthető és ingyen használható program vált be a legjobban, manapság is ezt használom, a fenti asztrofotóim mindegyikét ezzel a programmal dolgoztam fel.

Az asztrofotós szoftverek mindegyike elvben ugyanazokat a lépéseket követi: az egyes nyers fotók zajszűrése, esetleg egyéb fajta korrigálása, a képek illesztése (csillagonként megfeleltetik az összes képet az első képhez viszonyítva), a képek átlagolása és összeadása (stacking), a végső kép feldolgozása (részletek láthatóságának beállítása, méretre vágás, fehéregyensúly beállítása, háttérgradiensek korrigálása).

**Dr. Münzlinger Attila**

## **LEGO robotok**

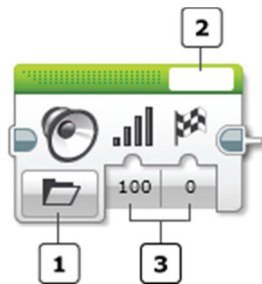
IV. rész

### **III.1.9. A hangfal programozása**

A *hangfal blokk* (Sound Block) segítségével szólaltathatjuk meg az EV3-tégla beépített hangfalát.

A 34. ábrán látható hangfal blokk részei:

- 1. A blokk módjának kiválasztó gombja (mode selector)
- 2. Szövegdoboz az állománynevek számára
- 3. Bemeneti adatok



34. ábra: A hangfal blokk

Az 1-es gomb segítségével választhatjuk ki, hogy a hangfalon a rendszer egy állományt játsszon le, egy hangot szólaltasson meg vagy egy hangjegyet, illetve leállítsa a hangfalat.

Amennyiben a kiválasztott mód egy hangállomány lejátszása, akkor a 2-es szövegdobozban adhatjuk meg az állomány nevét, vagy választhatunk a LEGO által eleve megadott hangok közül. Szintén itt választhatjuk ki azt is, hogy a név egy adatdrót segítségével legyen megadva bemenetként (Wired).

A 3-as gombok segítségével a bemeneti adatokat adhatjuk meg mindegyik mód esetén.

Állomány lejátszása esetén a hangállományt a projekthez kell adni, majd az állomány nevének kívül bemeneti adatként a hangerősséget adhatjuk meg egy 0 és 100 közötti szám segítségével, illetve a lejátszási módot, amely 0, 1 vagy 2 lehet. A 0 azt jelenti, hogy a rendszer megvárja a hangállomány lejátszását, és csak azután adódik át a vezérlés a következő blokknak (Wait for Completion), az 1-es esetben a hangállomány lejátszása megkezdésének pillanatában a vezérlés már át is adódik a következő blokknak, és a rendszer egyszer játssza le az állományt (Play Once). A 2-es esetben a hangállományt addig ismétli a rendszer, míg egy másik hangfal blokk le nem állítja (Repeat). A vezérlés ebben az esetben is azonnal átadódik a következő blokknak.

A hang lejátszása esetén Hz-ben adhatjuk meg a hang frekvenciáját 250-től 10 000-ig, vagy kiválaszthatjuk ezt a szabványos hangok listájából: C 261.63, D 293.67, A 440 stb. A második bemeneti adat a hang hosszát jelenti másodpercekben. Harmadik bemeneti adat a hangerő, negyedik pedig az előbb ismertetett lejátszási mód.

A hangjegy lejátszása nagyon hasonlít a hang lejátszására, azzal a különbséggel, hogy itt első paraméterként egy háromoktávós zongora szabványos hangjegyeit adhatjuk meg. A paraméter nélküli Stop mód leállít bármiféle hanglejátszást.

A 12. táblázat a hangjegyeket és ezek frekvenciáját foglalja össze.

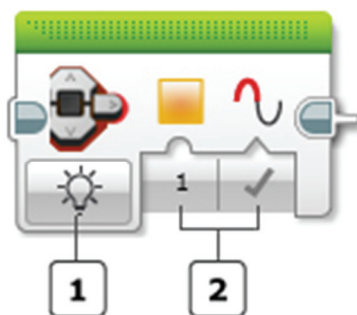
Betűjel	Hangjegy	Frekvencia / oktáv			
		4	5	6	7
C	Dó	261.63	523.25	1046.5	2093
C#	Di	277.18	554.37	1108.73	
D	Ré	293.67	587.33	1174.66	
D#	Ri	311.13	622.25	1244.51	
E	Mi	329.63	659.26	1318.51	
F	Fá	349.23	698.46	1396.91	
F#	Fi	369.99	739.99	1479.88	

Betűjel	Hangjegy	Frekvencia / oktáv			
G	Szó	392	783.99	1567.98	
G#	Szi	415.31	830.61	1661.22	
A	Lá	440	880	1760	
A#	Li	466.16	932.33	1864.66	
H (B)	Ti	493.88	987.77	1975.53	

12. táblázat: Hangjegyek

### III.1.10. A téglá állapotát jelző fények programozása

Az EV3-téglá állapotát zöld, narancsszínű és vörös villogó vagy folytonos fények jelezhetik. Ezeket a 35. ábrán látható blokkal programozhatjuk, ahol 1. a módváltó, 2. pedig a bemeneti adatok gombjai.



35. ábra: Az állapotjelző fények

Az 1-es mód Off (kikapcsolt), On (bekapcsolt) vagy Reset (Visszaállított) lehet. A mód függvényében változnak a bemeneti adatok.

A bekapcsolt mód két adatot kíván: a fények színét, ez 0, 1 vagy 2 lehet. A 0 a zöldet, az 1 a narancsszínűt, a 2 a vöröset jelenti, valamint a villogást jelentő logikai Igaz vagy Hamis értéket. Az állapotjelző fények bekapcsolt állapotban maradnak mindaddig, ameddig nem kapcsoljuk ki őket egy adatokat nem igénylő Off móddal, vagy amíg a program be nem fejeződik.

A visszaállítás mód szintén nem igényel bemenő adatokat, és az állapotjelző fényeket visszaállítja a szabványos program futását jelző villogó zöld állapotjelzésre. Ez a villogás valamivel másabb, mint a programozott zöld villogás.

### III.1.11. Az érintésérzékelő programozása

Az érzékelők bemeneti adatokat szolgáltatnak. Az érintésérzékelő a legegyszerűbb érzékelő, tulajdonképpen egy nyomógomb, amelynek három állapota lehet: *benyomott* (Pressed), *felengedett* (Released) és *ütközött* (Bumped).

Az érintésérzékelőtől tudhatjuk meg, ha a robotunk valamivel ütközött, de a gomb benyomásával vagy felengedésével különböző cselekvéseket is kiválthatunk. Az érintésérzékelő nem tud adatokat szolgáltatni arról, hogy például a gomb mennyi ideig volt benyomva, vagy milyen erősen volt megnyomva, csak arról, hogy be van-e nyoma, vagy sem, illetve az ütközött állapot azt jelzi, hogy a gomb be volt-e nyomva és fel volt-e en-

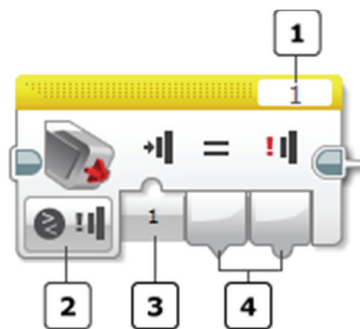
gedve a közelmúltban, tehát a robot ütközött-e már valamivel. Az ütközött állapot egyszerűen megmondja azt, hogy az érintésérzékelőt nyomógombként használtuk-e vagy sem, vagyis nem kell lekérdezni egymás után azt, hogy le van-e nyomva a gomb, majd fel van-e engedve.

A 13. táblázat azt mutatja be, hogy 7 lépésen keresztül milyen tevékenységekre milyen logikai értéket szolgáltat vissza az érintésérzékelő. Az érintésérzékelő felengedett állapotban lévő gombját kétszer egymás után lenyomjuk, és minden lépés után a program kiolvassa az érintésérzékelő által visszatérített logikai értéket.

Lépés	Tevékenység	Benyomott	Felengedett	Ütközött
1	Alapállapot: az érintésérzékelő gombja nincs benyomva	Hamis	Igaz	Hamis
2	Benyomjuk a gombot (nyomva van, míg a program beolvassa az adatokat)	Igaz	Hamis	Hamis
3	Felengedjük a gombot	Hamis	Igaz	Igaz
4	A gomb fel van engedve, és a program olvassa az érzékelő adatait	Hamis	Igaz	Hamis
5	Másodszorra is benyomjuk a gombot	Igaz	Hamis	Hamis
6	Felengedjük a gombot	Hamis	Igaz	Igaz
7	A gomb fel van engedve, és a program olvassa az érzékelő adatait	Hamis	Igaz	Hamis

13. táblázat: Az érintésérzékelő állapotainak változása

Az érintésérzékelő blokk segítségével leolvashatjuk az érintésérzékelőről kapott adatokat, amelyek az érintkező fizikai állapotát tükrözik.



36. ábra: Érintésérzékelő

A 36. ábrán látható blokkon az 1-es gomb segítségével a portot választhatjuk ki (port selector). Ezen a porton keresztül fog kommunikálni az EV3-tégla az érzékelővel, innen olvassa be az adatokat. A port az 1, 2, 3 vagy 4 valamelyike lehet.

A 2-es gomb segítségével egy legördülő menüből kiválaszthatjuk az érzékelő működési módját (mode selector), ez a *Measure* (mérés), vagy *Compare* (összehasonlítás) lehet.

A mérés üzemmódban a visszatérítési értéke egy logikai érték, amely Igaz (True) vagy Hamis (False) lehet. Ezt a 4-es gombon szolgáltatott visszatérítési értéket adatdrót segítségével adhatjuk át más blokknak.

A visszatérési érték Igaz, ha az érintésérzékelő be van nyomva, különben Hamis.

Az összehasonlítás üzemmódban a 3-as gomb segítségével bemeneti adatként megadhatjuk, hogy a *benyomott* (Pressed), *felengedett* (Released) vagy *ütközött* (Bumped) állapotot szeretnénk-e lemérni. A menü segítségével egy numerikus értéket állíthatunk be: 0 – felengedett, 1 – benyomott, 2 – ütközött.

A blokk két értéket térít vissza. Az első egy logikai érték, amely megmutatja, hogy az elvárt állapot következett-e be. A második pedig maga a szenzor állapota az előbbi kódolás szerint, numerikus értéként.

A 14. táblázat tevékenységi lépésként mutatja be, hogy az érintésérzékelő összehasonlítás üzemmódban milyen bemeneti értékre, milyen visszatérési értékeket szolgáltat.

<i>Lépés</i>	<i>Tevékenység</i>	<i>Bemenet</i>	<i>Első visszatérési érték</i>	<i>Második visszatérési érték</i>
1	Alapállapot: az érintésérzékelő gombja nincs benyomva	0	Igaz	0
2	Benyomjuk a gombot (nyomva van, míg a program beolvassa az adatokat)	0	Hamis	1
3	Felengedjük a gombot	0	Hamis	2
4	A gomb fel van engedve, és a program olvassa az érzékelő adatait	0	Igaz	0
5	Gyorsan benyomjuk és felengedjük a gombot (közben a program nem olvas, csak a felengedés után)	0	Hamis	2
6	Alapállapot: az érintésérzékelő gombja nincs benyomva	1	Hamis	0
7	Benyomjuk a gombot (nyomva van, míg a program beolvassa az adatokat)	1	Igaz	1
8	Felengedjük a gombot	1	Hamis	2
9	A gomb fel van engedve, és a program olvassa az érzékelő adatait	1	Hamis	0
10	Gyorsan benyomjuk és felengedjük a gombot (közben a program nem olvas, csak a felengedés után)	1	Hamis	2
11	Alapállapot: az érintésérzékelő gombja nincs benyomva	2	Hamis	0

Lépés	Tevékenység	Bemenet	Első visszatérési érték	Második visszatérési érték
12	Benyomjuk a gombot (nyomva van, míg a program beolvassa az adatokat)	2	Hamis	1
13	Felengedjük a gombot	2	Igaz	2
14	A gomb fel van engedve, és a program olvassa az érzékelő adatait	2	Hamis	0
15	Gyorsan benyomjuk és felengedjük a gombot (közben a program nem olvas, csak a felengedés után)	2	Igaz	2

14. táblázat: Az érintésérzékelő összehasonlítás üzemmódban

Az érzékelők programozása azért bonyolultabb, mint például a motoroké, mert a LEGO MINDSTORMS EV3 Home Edition szoftver vezérlőszerkezetei (ciklus, elágazás, várj-blokk stb.) is már eleve felhasználhatják az érzékelőkről nyert adatokat. Például egy ciklus tarthat addig, amíg az érintésérzékelő be van nyomva stb. Ezeket a funkciókat a megfelelő blokkoknál fogjuk letárgyalni.

### 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/Lego_Mindstorms)
- [http://en.wikipedia.org/wiki/Linux\\_kernel](http://en.wikipedia.org/wiki/Linux_kernel)
- [http://hu.wikipedia.org/wiki/ARM\\_architekt%C3%BAra](http://hu.wikipedia.org/wiki/ARM_architekt%C3%BAra)
- [http://hu.wikipedia.org/wiki/MOS\\_Technology\\_6502](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.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