

## LEGO robotok

XIII. rész

### III.1.25. Állományok

Az informatikában *adatállománynak*, *állománynak* vagy *fájl*nak nevezzük a logikailag összefüggő adatok halmazát, tömbjét. Tárolásuk bármilyen adathordozón történhet.

Minden állomány rendelkezik azonosítóval, mely alapján megkülönböztetjük őket. Ez az azonosító egy névből, egy kiterjesztésből és a kettőt elválasztó karakterből (pont) áll. Az azonosításnak különböző operációs rendszerekben eltérő szabályai vannak.

A *szövegállomány* vagy *szöveges állomány* sorokba rendezett, a sorokat a CR/LF karakterek zárják, az állományt pedig Ctrl-Z. A hozzáférés szekvenciálisan történik, az írás és az olvasás csak külön-külön történhet. Minden állomány, megnyitása után rendelkezik egy állomány mutatóval. Ez az aktuális pozíciót mutatja az állományban, írás vagy olvasás után ez elmozdul.

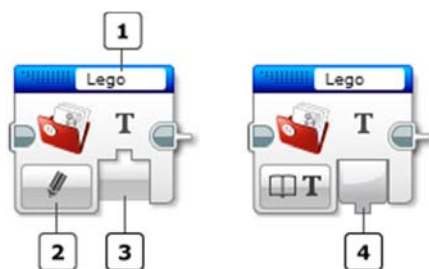
Az EV3 tégla esetén az állomány-hozzáférés blokk teszi lehetővé az adatok állományokból való olvasását és írását.

Használat után az állományt be kell zárni.

Ha egyszer létrehoztunk egy állományt, azután akárhányszor elérhető lesz, és az állomány-hozzáférési blokk, vagy a memóriaböngésző segítségével használni is tudjuk.

Az EV3 tégla automatikusan .rtf kiterjesztést ad az állományoknak, így ezeknek más kiterjesztésük nem lehet. Ha kiterjesztést is megadunk az állománynévnél (106. ábra 1-es gomb), akkor a tégla vagy végtelen ciklusba kerül, vagy *FILE NAME ERROR! (Állománynév hiba!)* hibaüzenettel leáll. Állománynévnek lehetőleg csak az angol ábécé betűit, számokat, illetve aláhúzás jelt („\_”) adjunk meg. Ha más karaktert adunk, a tégla jobb esetben *FILE NAME ERROR! (Állománynév hiba!)* hibaüzenettel leáll, rosszabb esetben valamilyen átkódolt nevű állományt hoz létre, amit le sem tudunk törölni, csak úgy, ha letöröljük a teljes projekt mappát, amiben a program és az állomány is van.

Ha nem létező nevű állományból akarunk olvasni, vagy törlést, bezárást hajtunk végre, az eredmény egy hibaüzenet lesz.



106. ábra: *Állományok*

A 106. ábrán látható blokk esetében az 1-es gombon az állomány nevét állíthatjuk be. A 2-es a *módszelektor*, itt az írás, olvasás, törlés, bezárás módokat állíthatjuk be. A 3-as gomb az input, a bemenetel, a 4-es gomb pedig az output, a kimenetel.

A *törlés mód* végérvényesen letörli a háttértárolóról a megadott nevű állományt. Ha egy állományt újra akarunk írni, teljesen új adatokat akarunk felvezetni, akkor először töröljük ki a meglévő állományt, majd írjuk be az új adatokat.

A *bezárás mód* segítségével tudjuk bezárni a megnyitott állományt, ha már nincs szükségünk rá, megtörtént az adatok olvasása vagy írása.

Az *olvasás mód* segítségével szöveges vagy numerikus értéket olvashatunk az állományból. Vigyázzunk, mert a numerikus értéket ki tudjuk bármikor olvasni szöveggként, de ha szöveget akarunk kiolvasni numerikus értéként, akkor az operációs rendszer és a processzor kódolásának megfelelő ábrázolási módban kapunk vissza értékeket, legjobb esetben 0-át, rossz esetben pedig az EV3 téglá nem várt módon viselkedhet.

*Írás mód*ban szöveges adatokat írhatunk az állományba. Ha a megadott nevű állomány nem létezik, a blokk létrehozza ezt. Ha az állomány létezik, akkor a végére fogja beírni az adatokat. Az írás mód soha nem törli ki az állományt.



107. ábra: Példaprogram – állományok

A 107. ábrán látható példaprogram létrehozza az *all01* nevű állományt, amelynek a neve a Memory Browserben *all01.rtf* lesz, majd beírja (szöveggként) a 15-ös értéket. Az állomány bezárása után szintén szöveggként kiolvassa ezt az értéket, majd ismét bezárja az állományt. A kiolvasott 15-ös értéket kiírja a téglá képernyőjére, és 5 másodperc múlva befejezi futását.

### III.1.26. Kommunikáció

Az EV3 téglák közötti, illetve a számítógép – EV3 téglá, vagy mobil telefon, tablett – EV3 téglá közötti drótnélküli (wireless) kommunikáció Bluetooth vagy WiFi segítségével valósulhat meg.

A WiFi, az IEEE által kifejlesztett vezeték nélküli mikrohullámú kommunikációt (WLAN) megvalósító, széleskörűen elterjedt szabvány népszerű neve. A WiFi, az elterjedt nézetekkel szemben, nem az angol Wireless Fidelity kifejezésnek a rövidítése. Az elnevezést egy marketingcég találta ki játékosan utalva a HiFi szóra, csak később igyekeztek rövidítésként aposztrofálni és úgy reklámozni.

Az EV3 téglá alpból nem tud WiFi kommunikációt megvalósítani, kell hozzá egy USB csatlakozású NetGear WNA1100 típusú külső WiFi modul.

Az EV3 téglá alpból Bluetooth kapcsolatot tud létesíteni.

A Bluetooth rövid hatótávolságú, adatcseréhez használt, nyílt, vezeték nélküli szabvány. Alkalmazásával számítógépek, mobiltelefonok és egyéb készülékek között automatikusan létesíthetünk kis hatótávolságú rádiós kapcsolatot.

A név Harald Blåtand (I. Harald dán király), dán király nevének angol változata, aki 958-tól, illetve 976-tól 986-ig volt Dánia és Norvégia uralkodója, és nagyon szerette az áfonyát, ezért kékek voltak a fogai. Harald arról volt nevezetes, hogy egyesítette a lá-

zongó dán, norvég és svéd törzseket. Ehhez hasonlóan a Bluetooth-ot is arra szánták, hogy egyesítsen és összekössön olyan különböző eszközöket, mint a számítógép vagy a mobiltelefon. A Bluetooth logója a H és B betűknek megfelelő skandináv rúnákat, a *Hagla*zt és a *Berkanant* idézi.

A Bluetooth kapcsolat árnyoldala, hogy személyes adatainkhoz olyanok is hozzáférhetnek, akiknek nem akartuk ezt megengedni.

Néhány könnyen betartható biztonsági tanács:

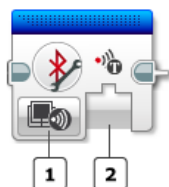
Csak akkor engedélyezzük a Bluetooth használatát az eszközben (például telefonban), amikor használni akarjuk, és használat után tiltsuk azt le.

- Használjunk hosszú, nehezen kitalálható számkódot az eszközök párosításához (használjunk 8 jegyű vagy még hosszabb számot – az „1234” kód nem jó).
- Párosítás után az eszköz legyen „rejtett” (hidden) állapotban, így is működni fog a már párosított másik eszközzel.
- Utasítsunk vissza minden ismeretlen kapcsolódási kísérletet.
- Engedélyezzük a titkosítást (encryption).
- Időnként nézzük meg a párosított eszközök listáját, nincs-e köztünk olyan, amit nem mi állítottunk be.
- Frissítsük a mobilunk firmware-szoftverét a legújabb verzióra a gyártó honlapjáról.

A Bluetooth-kapcsolat blokk segítségével kapcsolhatjuk be vagy ki a Bluetooth-ot, csatlakozhatunk egy másik Bluetooth-eszközhöz, vagy lezárhatjuk a kapcsolatot. A Bluetooth eszközök közé tartoznak az EV3 téglák, mobiltelefonok és számítógépek is, noha nem minden Bluetooth eszköz támogatja az EV3 téglával való kapcsolatot. Ha már létrehoztuk a Bluetooth-kapcsolatot az EV3 téglagombos menüje segítségével, akkor nem kell Bluetooth-kapcsolat blokkot használnunk a programban.

Az EV3 rendszer Bluetooth protokollja Master/Slave (mester/szolga) elven működik. Kiválasztjuk a Master EV3 téglát, és ezt használva csatlakozunk a Slave EV3 téglához. Egy Master EV3 téglá akár 7 Slave EV3 téglához is csatlakozhat. A Master EV3 téglá üzenetet küldhet minden Slave téglának, a Slave téglák azonban csak a Master téglának küldhetnek üzenetet, egymás között a Slave téglák nem tudnak közvetlenül kommunikálni.

Ha létrejött a Bluetooth-kapcsolat, akkor üzenetekkel tudunk kommunikálni.



108. ábra: A Bluetooth-kapcsolat blokk

A 108. ábrán látható Bluetooth-kapcsolat blokk 1-es gombja a *módszelektor*, 2-es gombja pedig a *bemenet*.

A módszelektor segítségével On (be), Off (ki), Initiate (kezdeményezés), illetve Clear (törlés) módokat tudunk beállítani.

A *be mód* bekapcsolja a Bluetooth kommunikációt.

A *ki mód* kikapcsolja a Bluetooth kommunikációt.

A *kezdeményezés mód* segítségével kapcsolatot kezdeményezhetünk egy meghatározott Bluetooth eszközzel. Az eszköz nevét a bemenet gombon kell megadni. A Bluetooth kapcsolat mindaddig fennmarad, amíg be nem zárjuk a törlés móddal.

A *törlés móddal* kapcsolatot zárhatunk le egy megadott nevű Bluetooth eszközzel.

### Üzenetküldés

A 109. ábrán látható üzenetek blokk segítségével Bluetooth üzeneteket küldhetünk az EV3 téglák között. Üzenet küldéséhez vagy fogadásához először az EV3 téglákat kell csatlakoztatni, akár a téglá Bluetooth menüje, akár a Bluetooth-kapcsolat blokk segítségével.

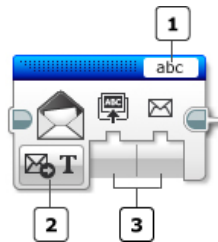
Az 1-es gomb tartalmazza az *üzenet azonosítóját*, a 2-es gomb a *módszelektor*, a 3-as gomb a *bemenetek*.

Az *üzenet azonosítója* egy szöveges címke, amely egyértelműen azonosítja az üzenetet.

Minden üzenetet három összetevő jellemez:

1. Azok a téglák, amelyek között az üzenet átmegy.
2. Az üzenet azonosítója.
3. Az üzenet értéke. Ez lehet szöveges, numerikus vagy logikai.

Az üzenet azonosítója lehetővé teszi, hogy egyszerre több adatot küldjünk az EV3 téglák között. Például egy téglá „sebesseg” és a „szog” azonosítójú üzeneteket (adatokat) egyszerre küld ugyanabban a programban.



109. ábra: Az üzenetek blokk

A *módszelektor* segítségével a küldés, fogadás, valamint összehasonlítás üzemmódokat állíthatjuk be. Az üzemmód kiválasztása után kiválaszthatjuk a bemenetek értékeit. A rendelkezésre álló bemenetek az üzemmódtól függően változnak.

*Küldés* üzemmódban szöveges, numerikus vagy logikai adatot küldhetünk a megadott téglának. Ebben az esetben bemenetként meg kell adni a fogadó téglá nevét, valamint az üzenet értékét (az adatot).

*Fogadás* üzemmódban a téglá szöveges, numerikus vagy logikai üzenetet kaphat egy Bluetooth segítségével csatlakoztatott EV3 téglától. Ekkor a blokknak egyetlen kimenete van (a bemenet kimenetévé válik), mégpedig a kapott üzenet. A kimenet False (hamis) mindaddig, amíg meg nem érkezik az üzenet.

Az *összehasonlítás* üzemmód összehasonlítja a kapott üzenetet (szöveges, numerikus vagy logikai) egy meglévő értékkel. Ha a feltételek teljesülnek, akkor a kimenet igaz (True) lesz, ha a feltételek nem teljesülnek, a kimenet hamis (False) lesz.

A szöveges és numerikus összehasonlítás esetében a blokknak két-két bemenete és két-két kimenete van.

Az egyik bemenet az összehasonlító művelet.

Szövegek összehasonlítására csak az *egyenlő* (0), illetve a *nem egyenlő* (1) relációs műveleteket használhatjuk, numerikus értékek összehasonlítására a 23. táblázatban is szereplő *egyenlő* (0), *nem egyenlő* (1), *nagyobb* (2), *nagyobb vagy egyenlő* (3), *kisebb* (4), *kisebb vagy egyenlő* (5) relációs műveletek állnak a rendelkezésünkre.

A másik bemenet az összehasonlítandó érték.

Az egyik kimenet az összehasonlítás eredménye (igaz vagy hamis), a másik kimenet pedig maga az üzenet.

Logikai összehasonlítás esetében a blokknak csak két kimenete van: az összehasonlítás eredménye, illetve maga az üzenet.

### III.1.27. Sajátos motorblokkok

A kék fülben két sajátos motorblokk található, a motor invertálás, illetve a szabályozatlan motor.

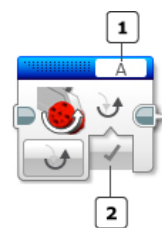
A 110. ábrán látható motor invertálás blokk egyszerűen megváltoztatja a motor forgásirányát. Ha a motor eddig az óramutatóval megegyező forgásirányban haladt, ezentúl az óramutatóval ellentétes lesz a forgásiránya, és fordítva.

Az 1-es gomb segítségével a *portot* állíthatjuk be (A, B, C, D valamelyike), a 2-es gomb pedig a *bemenet*, ahol megadhatjuk a forgásirányt. Ha az inverz bemenet igaz (True), akkor a kiválasztott motor normál „előre” és „hátra” irányt vált. A motor invertálás blokk után minden olyan programblokk, amely a motort az óramutató járásával megegyező irányba fordítaná, a motort az óramutató járásával ellentétes irányba fogja fordítani, és fordítva.

Ha egy motor irányát megváltoztattuk, az megváltozva marad mindaddig, amíg egy másik motor invertálás blokk vissza nem állítja a hamis (False) bemeneti értékkel.

A 111. ábrán látható szabályozatlan motor blokk a közepes motort és a nagy motort is vezérelni tudja. Segítségével bekapcsolhatjuk a megadott porton lévő motort, és ellenőrizni tudjuk a motor teljesítményszintjét.

A szabályozatlan motorblokk nem tartalmazza a motor automatikus vezérlését. A megadott teljesítmény bemenet csak a motor vezérlésére szolgál. A motor sebességét és irányát a tápfeszültség bemenet segítségével szabályozhatjuk. A motor mindaddig forog, amíg meg nem állítja egy másik motorblokk, vagy amíg a program véget nem ér.



110. ábra:

*A motor invertálás blokk*



111. ábra:

*Szabályozatlan motor blokk*

A teljesítményvezérlés kompenzálja a motor által tapasztalt bármilyen ellenállást vagy csúszást, az energiagazdálkodás pedig lehetőség szerint megpróbálja kompenzálni az akkumulátor szintjét.

A 112. ábrán látható program bekapcsolja a motort, 5 másodpercig forgatja, majd a nagy motor blokk segítségével megállítja.



112. ábra: A szabályozatlan motor blokk használata

### III.1.28. További lehetőségek

A kék fül még három további sajátos lehetőséget tartalmaz: a virrasztás blokkot, a nyers érzékelő érték blokkot, valamint a programleállítás blokkot.

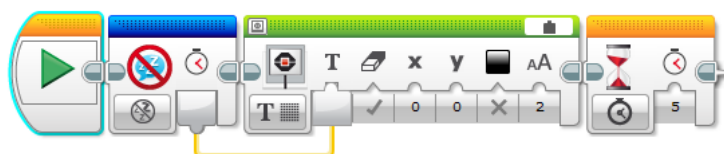
A 113. ábrán látható virrasztás blokk visszaadja az EV3 tégla várakozási idejét. Akkor használjuk ezt a blokkot, ha a programnak hosszabb ideig kell várnia mint az EV3 tégla várakozási ideje, amelyet a tégla interfésze segítségével tudunk beállítani.

A blokknak egyetlen kimenete van, amely azt mutatja, hogy hány milliszekundum marad az EV3 tégla leállásáig (az alvásig hátralévő idő). Mivel a virrasztás blokk azonnal elindítja a tégla alvási időzítőjét, az alvási idő meg fog egyezni az alvás beállítással.



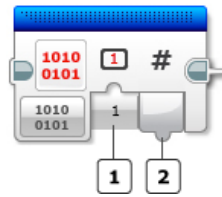
113. ábra: A virrasztás blokk

A 114. ábrán látható program kiírja, hogy az EV3 tégla hány milliszekundum múlva fog leállni.



114. ábra: A virrasztás blokk használata

A 115. ábrán látható nyers érzékelő érték blokk megadja a fel nem dolgozott érzékelő-értéket, ami egy 0 és 1023 közötti szám.



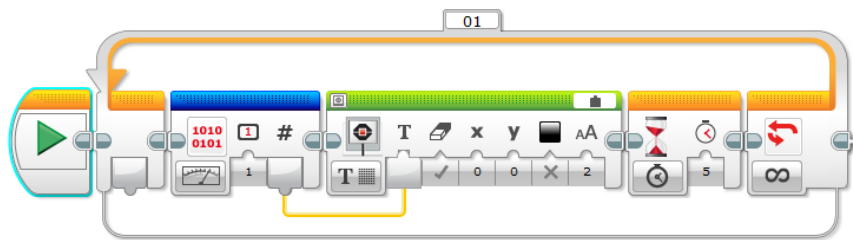
115. ábra: A *nyers érzékelő blokk*

A bloknak egy bemenete és egy kimenete van. A bemenet a *port* (1-es gomb), a kimenet pedig egy *érték* (2-es gomb).

Minden érzékelő nyers értéket szolgáltat vissza. A programozási blokkok ezeket az értékeket veszik át, és alakítják át őket több információt szolgáltató számmá.

Például az érintésérzékelő, ha nincs benyomva, egy 160 körüli nyers értéket térít vissza, ha pedig be van nyomva, egy 3360 körüli értéket. Ezt a nyers értéket dolgozza fel az érintésérzékelő programozási blokkja, és adja vissza azt, hogy az érzékelő be volt-e nyomva, vagy sem.

A nyers érzékelő érték blokk akkor hasznos, ha egy olyan érzékelőt szeretnénk használni, amelyik nem rendelkezik még programozási blokkal, például egy harmadik fél, vagy akár egy általunk gyártott érzékelőt. Így használni tudjuk például a LEGO MINDSTORMS EV3 Home Edition szoftvert anélkül, hogy az érzékelőnek programozási blokkja lenne.



116. ábra: A *nyers érzékelő blokk használata*

A 116. ábra egy olyan programot mutat be, amely az 1-es porton lévő érintésérzékelő nyers adatait írja ki a tégla képernyőjére.

A 117. ábra a programleállítás blokkot mutatja be.

A programleállítás blokk azonnal lezárja az összes programozási blokkot, és befejezi a programot.



117. ábra: A *programleállítás blokk*

Programleállítási blokkot bárhova tehetünk, elágazásokba, ciklusokba is. Ha a vezérlés elérte ezt a blokkot, a program azonnal leáll, így egy tetszőleges program végére nem érdemes ilyen blokkot tenni, mert ott a futás így is, úgy is leállna.

Kovács Lehel István