

Egyszerű animáció micro:bit segítségével

Simple animation using micro:bit

Dr. KOVÁCS Lehel István

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

Abstract

Micro: bit is a popular tool in the education of informatics and computer science, with the help of which it is possible to introduce algorithmic thinking, the structure and construction of programs, IT thinking and last but not least the use of electronic and hardware components in schools. In this study, we show what kind of animations we can create with the help of micro:bit. Even without knowledge of computer graphics, students can master the essence of simple keyframe animation, pay attention to planning key positions, and analyze the movement. They can also expand their knowledge at the level of hardware elements, use LED matrices, OLED displays, SPI and I²C data transfer standards.

Keywords: micro:bit, animation, OLED display, keyframe, LED-matrix

Kivonat

A micro:bit az informatika, számítástechnika oktatásának egyik közkedvelt eszköze, amelynek segítségével már kiskorban meg lehet ismertetni az algoritmikus gondolkodást, a programok szerkezetét és felépítését, az informatikai gondolkodást és nem utolsósorban az elektronikai és hardverelemek használatát. Ebben a tanulmányban azt mutatjuk be, hogy a micro:bit segítségével milyen animációkat készíthetünk. A tanulók, diákok a számítógépes grafika ismerete nélkül is el tudják így sajátítani az egyszerű kulcskocka animáció lényegét, oda tudnak figyelni a kulcspozíciók megtervezésére, ki tudják elemezni a mozgást. Hardverelemek szintjén is bővíthetik ismereteiket, használhatják a LED-mátrixokat, OLED kijelzőket, az SPI, valamint az I²C adatátviteli szabványokat.

Kulcsszavak: micro:bit, animáció, OLED kijelző, kulcskocka, LED-mátrix

1. BEVEZETŐ

Az animáció a mozgás művészete. Egy olyan filmkészítési technika, amely élettelen tárgyak, rajzok vagy ábrák „kockázásával” olyan illúziót kelt a nézőben, mintha az egymástól kissé eltérő képkockák sorozatából összeálló történetben, a szereplők megelevenednének vagy élnének [1]. Számítógép segítségével az animáció egyre közelebb jut a valós, életszerű lények reális mozgásához [5], [6].

A kulcskocka animáció (keyframe animation) egy egyszerű animációs technika, amely arról szól, hogy egymástól alig eltérően megrajzolt képkockákat vetítünk egymásután (másodpercenként legalább 24-et), és az emberi szem ezt folytonos mozgásnak látja. A kulcskocka tehát egy rajz vagy felvétel, amely meghatározza egy sima átmenet kezdő- és végpontját [5], [6].

A kulcskockák sorozata határozza meg, hogy a néző milyen mozgást fog látni, míg a kulcskockák helyzete a filmen, videón vagy animáción meghatározza a mozgás időzítését. Mivel csak két-három kulcskocka egy másodperc alatt nem kelti a mozgás illúzióját, a többi képkockát „köztes” elemekkel kell kitölteni (1. ábra).



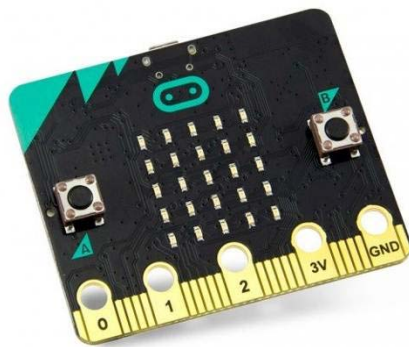
1. ábra. Kulcskocka animáció. A barnák a kulcskockák, a kékek a „köztes” elemek

A BBC micro:bit egy kifejezetten oktatási célra létrehozott, egy lapkás mikrovezérlő, amely 4×5 cm-es méretével, 5×5-ös LED-mátrix kijelzőjével, gyorsulásérzékelő, hőmérséklet érzékelő, fény-érzékelő, irány érzékelő szenzorokkal, 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 [2] 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 [3].

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.

Mindazonáltal sem a V1, sem a V2 nem rendelkezik a saját LED-mátrixukon kívül semmiféle kijelzővel.

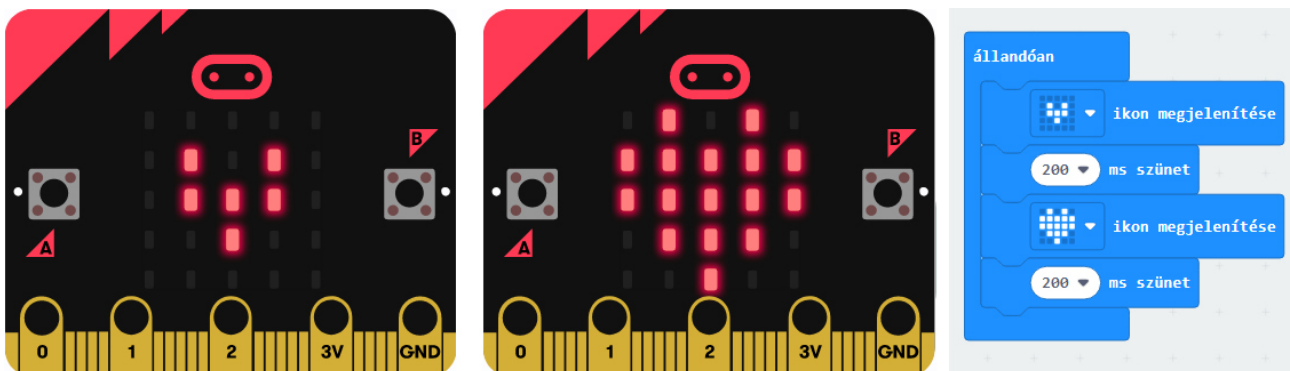


2. ábra. BBC micro:bit

2. ANIMÁCIÓS LEHETŐSÉGEK

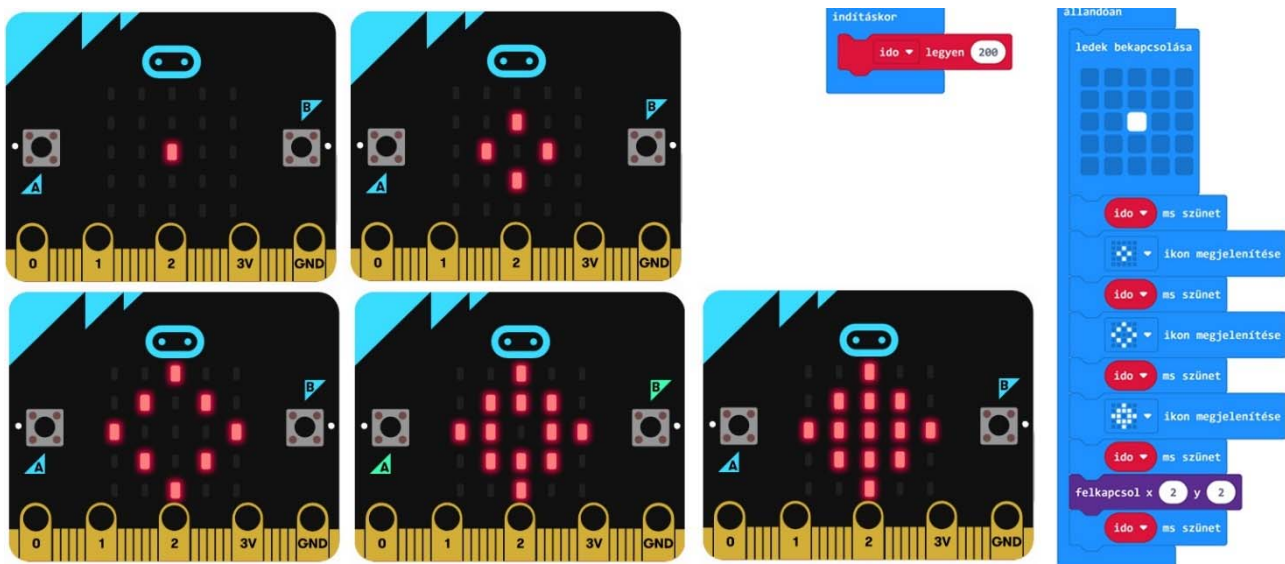
A micro:bit saját 5×5-ös LED-mátrix kijelzőjével rendelkezik. Már ez elegendő ahhoz, hogy kezdetleget animációkat készítsünk, például a 3. ábrán látható szívdobogást vagy a 4. ábrán bemutatott gyémánt-növekedést.

Mindkét animációnak az a lényege, hogy kulcskockák sorozatát adjuk meg, és ezeket bizonyos időközönként kirajzoljuk, megjelenítjük.



3. ábra. Szívdobogás

Azt, hogy gyorsabban vagy lassabban működjön az animáció, egy változó segítségével tudjuk befolyásolni. A változóban adjuk meg a szünet időegységét, amint azt a 4. ábrán láthatjuk. Ez az ábra azt is bemutatja, hogy a micro:bit kijelzőjét többféleképpen is programozhatjuk: ikonok segítségével, saját rajzzal vagy akár az egyes LED-eket külön-külön is elérhetjük. A LED-ek indexelése a mátrixban 0, 0-val kezdődik.



4. ábra. Gyémánt-növekedés

Mindazonáltal egy 5×5-ös LED-mátrix kijelzőn igencsak korlátozottak az animációs lehetőségeink, nézzük meg tehát, hogyan csatlakoztathatunk a micro:bit-hez más kijelzőket!

3. EGY 8×8-AS LED-MÁTRIX LEHETŐSÉGEI

Egy egyszínű 8×8-as LED-mátrix kijelző 64 LED-et tartalmaz, amelyek külön-külön történő vezérléséhez 65 vezetékre lenne szükség (64 I/O láb és a közös anód vagy a katód).

A 8×8-as LED-mátrixok esetében is egyszerűbb, ha a kereskedelemben kapható, az 5. ábrán látható MAX7219-es modellt használjuk, hisz ennek a vezérlése 5 pin segítségével megoldható (SPI szinkron soros kommunikációs interfész által), sőt megvan az az előnye, hogy hasonló LED-mátrixokkal azonnal sorba felfűzhető.



5. ábra. MAX7219-es LED-mátrix

Ez a modell egy soros meghajtású 8×8-as LED-mátrix, amelyet egy integrált soros MAX7219 kijelző-meghajtó vezérel. Ez a meghajtó nemcsak 8×8-as LED-mátrixot tud vezérelni, hanem akár 8 darab hétszögletes digitális LED-kijelzőt is.

Felépítésében BCD-kódoló, többcsatornás (multiplex) letapogató áramkör, belső 16 bites shift regiszter, de egy statikus RAM is található az egyes adatok tárolására.

A LED-mátrix programozásához töltsük le és telepítsük a 6. ábrán látható MAX7219_8x8 bővítményt, amely a MAX7219_8x8 nevű menüt adja a menüsorhoz.



6. ábra. A MAX7219_8x8 kiterjesztés

A kiterjesztés 25 blokkot tartalmaz, ezek a következők:

- Setup MAX7219: Number of matrixs **NR** CS(Load) = **PIN1** MOSI(DIN) = **PIN2** MISO(not used) = **PIN3** SCK(CLK) = **PIN4**: beállítja a LED-mátrix vezérlését, inicializálja a mátrixot. A **NR** a sorba fűzött 8×8-as LED-mátrixok száma, ahol 0 a micro:bit-től legtávolabb eső mátrix indexe, a micro:bitre pedig a **NR – 1** indexű LED-mátrix van kötve. A **PIN1**, **PIN2**, **PIN3**, **PIN4** a megfelelő lábak pinjei. Alapértelmezett pinek: CS: P16, MOSI: P15, MISO: P14, SCK: P13.
- Display text (align right) **SZÖVEG** clear screen first **ÉRTÉK**: jobbra igazítva megjeleníti a **SZÖVEG** szöveget úgy, hogy előtte letörli a kijelzőt, ha az **ÉRTÉK** igaz vagy nem törli le, ha az **ÉRTÉK** hamis.
- Display text (align left) **SZÖVEG** clear screen first **ÉRTÉK**: balra igazítva megjeleníti a **SZÖVEG** szöveget úgy, hogy előtte letörli a kijelzőt, ha az **ÉRTÉK** igaz vagy nem törli le, ha az **ÉRTÉK** hamis.
- Scroll text **SZÖVEG** delay (ms) **KÉSLELTETÉS** at the end wait (ms) **VÁRAKOZÁS**: a kijelzőn végiggörgeti a **SZÖVEG** szöveget **KÉSLELTETÉS** sebességgel úgy, hogy a végén **VÁRAKOZÁS** ms időt várakozik.
- Randomize all LEDs: véletlenszerűen felkapcsol LED-eket.
- Clear all LEDs: letörli a kijelzőt, minden LED-et lekapcsol.
- Fill all LEDs: minden LED-et felkapcsol.
- Set all brightness level **FÉNYERŐ**: **FÉNYERŐ** erősségre állít minden LED-et. A fényerősséget egy 0–15 skálán lehet beállítani.
- Set 8x8 pattern **MINTA** x = **X** y = **Y** value to **ÉRTÉK**: a LED-mátrixon látható kép egyik megadási módja egy 8×8-as **MINTA** segítségével. Egy változónak átadunk egy üres mintát, majd ebben beállítjuk az **X**, **Y** koordinátájú LED-et a megadott **ÉRTÉKRE**.
- Empty 8x8 pattern: az üres 8×8-as mintát jelenti.
- Display 8x8 pattern **MINTA** on matrix index = **NR**: a **NR** indexű LED-mátrixon megjeleníti a **MINTA** mintát, képet.
- Rotate matrix display **IRÁNY** reverse printing order **ÉRTÉK**: a megadott **IRÁNY** irányban (none – nem forog, clockwise – az óramutató forgásával megegyező irányban, counter-clockwise – az óramutató forgásával ellentétes irányban, 180-degree – 180 fokkal) elforgatja a képet szöveget. Be lehet állítani az írás irányát is balról jobbra (hamis / false), jobbról balra (igaz / true) **ÉRTÉK** szerint.
- Display all fonts at delay **KÉSLELTETÉS**: **KÉSLELTETÉS** sebességgel megjeleníti a teljes karakterkészletet.
- Add custom character **KARAKTER** number array **TÖMB** to the extension font library: hozzáad egy **KARAKTER** azonosítójú karaktert a készlethez. A karaktert a **TÖMB** 8 elemű számtömbbel lehet megadni. Minden szám egy sort kódol binárisan, ezt kell 10-es számrendszerbe átalakítva betenni a tömbbe.

- Get custom character number array from byte-array string **SZÖVEG**: visszatérít egy karaktert, képet a **SZÖVEG** 8 elemű bináris tömbből. Ez a képek megadásának harmadik módja, ekkor minden sort egy 8 hosszúságú bináris kódként kell megadni, ezeket a kódokat „B” előtaggal, vesszővel elválasztva kell felsorolni (lásd 7. ábra).
- Display custom character from number array **TÖMB** offset **ELTOLÁS** clear screen first **ÉRTÉK**: megjelentet egy felhasználó által definiált karaktert **ELTOLÁS** eltolással úgy, hogy előtte letörli a kijelzőt, ha az **ÉRTÉK** igaz vagy nem törli le, ha az **ÉRTÉK** hamis. A karaktert kétféleképpen lehet megadni: vagy az előbb említett bináris szöveggént, vagy a **TÖMB** 8 elemű számtömbbel. Ekkor minden szám egy sort kódol binárisan, ezt kell 10-es számrendszerbe átalakítva betenni a tömbbe (lásd 66. ábra). Például ha a mátrix főátlóján szeretnénk felkapcsolni a LED-eket, akkor a 8 elemű tömb 2 hatványait tartalmazza: 1, 2, 4, 8, 16, 32, 64, 128.
- Randomize LEDs on matrix index = **INDEX**: véletlenszerűen felkapcsol LED-eket az **INDEX** indexű LED-mátrixon.
- Clear LEDs on matrix index = **INDEX**: kikapcsolja a LED-eket az **INDEX** indexű LED-mátrixon.
- Fill LEDs on matrix index = **INDEX**: bekapcsolja a LED-eket az **INDEX** indexű LED-mátrixon.
- Set brightness level **FÉNYERŐ** on matrix index = **INDEX**: **FÉNYERŐ** erősségre állít minden LED-et az **INDEX** indexű LED-mátrixon. A fényerősséget egy 0–15 skálán lehet beállítani.
- Turn on all matrixs **ÉRTÉK**: bekapcsolja vagy kikapcsolj az **ÉRTÉK**-nek megfelelően (igaz vagy hamis) a LED-mátrixokat.
- Toggle value in 8x8 pattern **MINTA** x = X y = Y: a 8×8-as **MINTA** X, Y koordinátájú értékét megcseréli (ha 0-s volt 1-es lesz, ha 1-es volt 0-s lesz).
- Get value from 8x8 pattern **MINTA** x = X y = Y: visszatéríti a 8×8-as **MINTA** X, Y koordinátájú értékét.
- Full 8x8 pattern: a tele 8×8-as mintát jelenti.
- Display 8x8 pattern **MINTA** on all matrixs: minden LED-mátrixon megjelenteti ugyanazt a 8×8-as **MINTA**-t.

	0	1	2	3	4	5	6	7	Bináris szöveg	Szám
0	○	○	○	●	○	○	○	○	▶ B00010000 ▶	16
1	○	○	●	○	●	○	○	○	▶ B00101000 ▶	40
2	○	●	○	○	○	○	●	○	▶ B01000100 ▶	68
3	○	●	○	○	○	○	○	●	▶ B01000100 ▶	68
4	○	●	●	●	●	●	●	○	▶ B01111110 ▶	126
5	○	●	○	○	○	○	○	○	▶ B01000100 ▶	68
6	○	●	○	○	○	○	○	○	▶ B01000100 ▶	68
7	○	●	●	○	○	○	○	○	▶ B01100110 ▶	102

7. ábra. Karakterek megadása

A 8. ábrán látható program inicializálja a LED-mátrixot, majd megjelenteti a bináris szöveg segítségével definiált „A” karaktert.

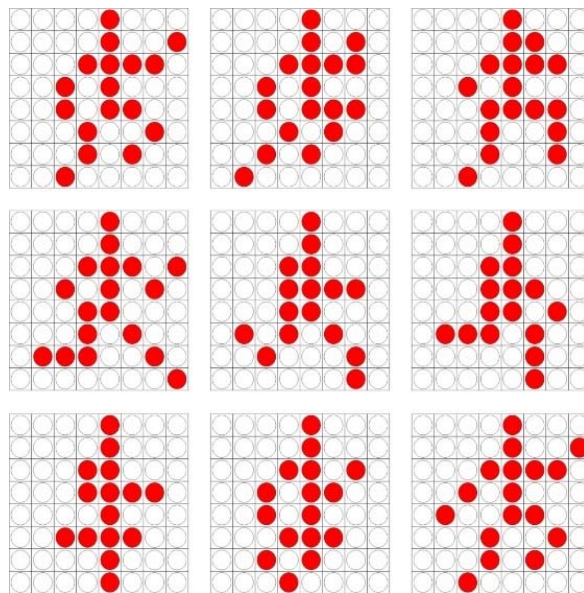


8. ábra. „A” karakter megjelenítése

A következő feladat nagyobb kihívás: jelenítsünk meg egy futó pálcikaemberkét! A figura megjelenítéséhez kulcskocka animációt használunk. Ez azt jelenti, hogy a megjelenítendő figurát több lényeges pozícióban megrajzoljuk, majd ezeket egymásután levetítve kapjuk meg a kívánt mozgást.

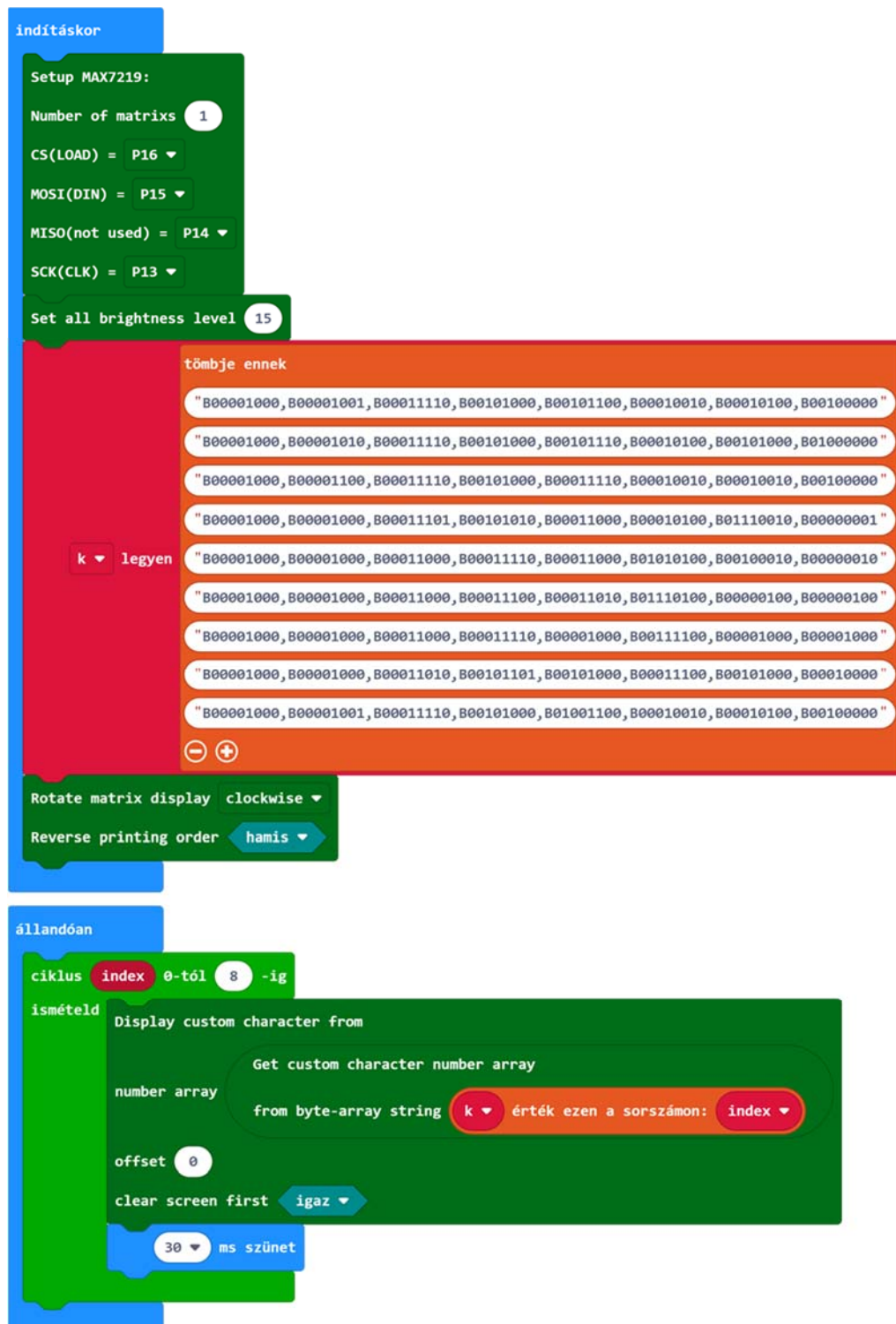
Ha a kulcskockák megadásához a mozgást akarjuk tanulmányozni, akkor javasoljuk, hogy használjuk a *Pivot Animator* szoftvert [4].

Amint azt a 9. ábrán is láthatjuk, a futó alakot 9 kulcspozícióban rajzoltuk meg, majd ezeket kódoltuk bináris szöveggé, és adtuk át a 10. ábrán látható programnak.



9. ábra. Futó alak kulcspozíciói

A teljes program a 10. ábrán látható: az inicializálás után adjuk meg a tömböt, elforgatjuk a kijelzőt, majd 30 ms-os szünettel ciklusban megjelenítjük a képeket. A kulcspozíciók váltakozásai folyamatos mozgást eredményeznek: az alak futni kezd a LED-mátrixon.



10. ábra. Animáció LED-mátrixon

4. MAGASABB SZINTŰ LEHETŐSÉGEK

A micro:bitet I²C szabvány szerint kommunikáló OLED kijelzővel is el lehet látni.

Az OLED megjelenítők kifejlesztéséhez az ötletet szentjánosbogarak szolgáltatták. Az állatok fénykibocsátása azt jelzi, hogy a természetben olyan szerves anyag is előfordul, amely az elektromos feszültség hatására fényt bocsát ki.

Ezt, az *elektrolumineszcencia* névre hallgató fizikai jelenséget használják fel az OLED (Organic Light-Emitting Diode – *szerves fénykibocsátó dióda*) kijelzőkben. Az anód és a katód között szerves szén alapú félvezető anyag található, és ez az anyag az elektromosság hatására fényt bocsát ki magából.

Az OLED megoldás nagyon kevés helyet foglal, és lehetséges, hogy egy képernyő minden egyes képpontját, pixelét külön-külön vezérelhessük.



11. ábra. *OLED kijelző*

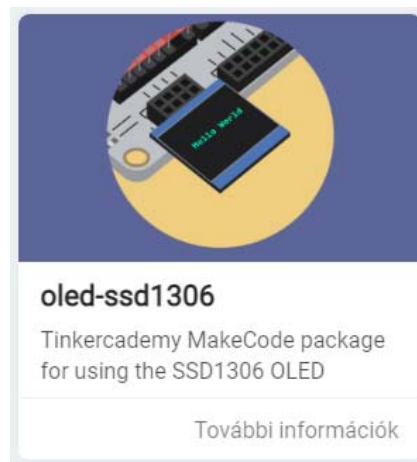
A 11. ábrán látható OLED kijelző 0,96"-es, felbontása 128×64 pixel. Ha karaktereket írunk ki, ez azt jelenti, hogy 8 sor, soronként pedig 21 karakternyi hely áll a rendelkezésünkre.

A kijelzőre 160°-os szögből tökéletes rálátás nyílik, 0,08 W-ot fogyaszt, és jól működik 3 V feszültséggel, tehát a micro:bittel is. A kijelzőt egy SSD1306 chip vezérli.

A micro:bittel I²C szabvány szerint kommunikál, ez azt jelenti, hogy csak négy pinre van szüksége: 3 V, GND, SCL, SDA. Amint már láttuk, az SCL az I²C órajel, az SDA az I²C adatjel.

Bekötése tehát nagyon egyszerű, a 3 V (VCC – Voltage Common Collector) a micro:bit 3 V pinjére, a GND a micro:bit GND pinjére kerül. Az SCL-t a micro:bit standard SCL pinjére kell kötni, ez a P19-es, az SDA-t pedig a micro:bit standard SDA pinjére, ez a P20-as. A bekötéseket apa-anya típusú Dupont huzalokkal is elvégezhetjük a szerelőlapon, vagy anya-anya típusú Dupont huzalokkal az élcsatlakozón:

Az OLED kijelző programozása is egyszerű. Használunk kell a 12. ábrán látható oled-ssd1306 bővítmény nyújtotta lehetőségeket.



12. ábra. *OLED bővítmény*

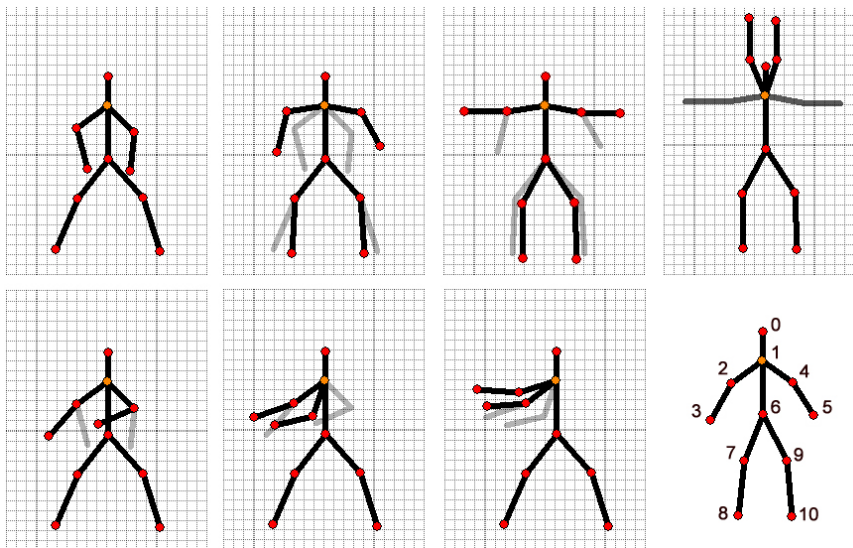
Ez a bővítmény a 10 blokkot tartalmazó *OLED* menüt telepíti. A blokkok a következők:

- initialize OLED with width **SZÉLESSÉG** height **MAGASSÁG**: inicializálja az OLED kijelzőt, és felkészíti a micro:bittel való használatra. Beállítja a kijelző szélességét és magasságát. Ez itt 128×64 pixel.
- show string **SZÖVEG**: kiírja a megadott szöveget.
- show number **SZÁM**: kiírja a megadott számot.
- show (without newline) string **SZÖVEG**: kiírja a megadott szöveget úgy, hogy azután nem megy át új sorba.
- show (without newline) number **SZÁM**: kiírja a megadott számot úgy, hogy azután nem megy át új sorba.

- insert newline: sortörést hajt végre, vagyis átugrik a kurzor egy új sorba.
- clear OLED display: törli az OLED kijelzőt.
- draw loading bar at **SZÁZALÉK** percent: megjelenít egy folyamatjelző sávot a kijelző közepén a haladás megadott százalékával. A sáv fölé kiírja, hogy „Loading” és ott is megjelenik a megadott százalék.
- draw line from x: **X1** y: **Y1** to x: **X2** y: **Y2**: a megadott koordináták közé vonalat húz.
- draw rectangle from x: **X1** y: **Y1** to x: **X2** y: **Y2**: a megadott koordináták közé kirajzol egy téglalapot.

Minket az előbbiekből a grafikus vonalrajzoló parancs érdekel.

Az animációhoz itt is a *Pivot Animator* szoftver [4] segítségével tervezünk meg egy egyszerű pálcikaemberkét. Tulajdonképpen egy 10 csontból és 11 ízületből álló csontvázunk, csont/ízület rendszerünk lesz, a 7 alap kulcskockát és a kockák közötti átmeneteket is, amelyek segítségével a pálcikaemberke tornagyakorlatokat fog tudni végezni a 13. ábrán láthatjuk.



13. ábra. Pálcikaemberke tornagyakorlata

A 13. ábrán látható ízületpozíciókat átkódoljuk x, y koordinátákká, majd a micro:biten megadjuk ezeket két tömb adatszerkezet segítségével. Ezután történik meg a rajzolás, valamint az animáció, a mozgás összerakása.

Az inicializálás:

```
OLED.init(128, 64)
let x1 = [10, 10, 7, 8, 13, 12, 10, 7, 5, 14, 15]
let y1 = [7, 10, 12, 16, 13, 17, 15, 20, 24, 20, 24]
let x2 = [10, 10, 5, 4, 13, 15, 10, 6, 6, 14, 14]
let y2 = [7, 10, 11, 15, 11, 14, 15, 19, 25, 19, 25]
let x3 = [10, 10, 6, 2, 14, 18, 10, 8, 8, 13, 13]
let y3 = [7, 10, 11, 11, 11, 11, 15, 20, 25, 20, 25]
let x4 = [10, 10, 8, 8, 11, 11, 10, 8, 8, 13, 13]
let y4 = [7, 10, 6, 1, 6, 1, 15, 20, 25, 20, 25]
let x5 = [10, 10, 7, 4, 13, 9, 10, 7, 5, 14, 15]
let y5 = [7, 10, 12, 15, 13, 14, 15, 19, 25, 20, 25]
let x6 = [10, 10, 6, 2, 8, 4, 10, 7, 5, 14, 15]
let y6 = [7, 10, 12, 14, 15, 16, 15, 20, 25, 20, 25]
let x7 = [10, 10, 5, 1, 6, 2, 10, 7, 5, 14, 15]
let y7 = [7, 10, 9, 7, 11, 12, 15, 20, 25, 20, 25]
```

Ahhoz, hogy az OLED képernyő 128×64 -es felbontásába jól illeszkedjenek bele a fenti koordináták, először mindegyiket skálázzuk 2-vel (duplájára növeljük), majd az x koordinátákat 45-tel, az y koordinátákat pedig 5-tel toljuk el. Így a képernyő közepére kerül a mozgó alak és jól látható is lesz a kijelzőn.

A kirajzolást egy függvénnyel oldjuk meg:

```
function rajzol(list: number[], list2: number[]) {
```

```

OLED.drawLine(list[1], list2[1], list[2], list2[2])
OLED.drawLine(list[2], list2[2], list[3], list2[3])
OLED.drawLine(list[1], list2[1], list[4], list2[4])
OLED.drawLine(list[4], list2[4], list[5], list2[5])
OLED.drawLine(list[6], list2[6], list[7], list2[7])
OLED.drawLine(list[7], list2[7], list[8], list2[8])
OLED.drawLine(list[6], list2[6], list[9], list2[9])
OLED.drawLine(list[9], list2[9], list[10], list2[10])
OLED.drawLine(list[0], list2[0], list[1], list2[1])
OLED.drawLine(list[1], list2[1], list[6], list2[6])
}

```

A tornagyakorlatot, vagyis a kulcskockák sorozatát az állandóan végrehajtott blokkban rajzoljuk ki. Sajnos az OLED törlése és maga a grafikus kirajzolás is eléggé időigényes, így az animáció villogni fog.

```

basic.forever(function() {
  OLED.clear()
  rajzol(x1, y1)
  OLED.clear()
  rajzol(x2, y2)
  OLED.clear()
  rajzol(x3, y3)
  OLED.clear()
  rajzol(x4, y4)
  OLED.clear()
  rajzol(x3, y3)
  OLED.clear()
  rajzol(x2, y2)
  OLED.clear()
  rajzol(x1, y1)
  OLED.clear()
  rajzol(x5, y5)
  OLED.clear()
  rajzol(x6, y6)
  OLED.clear()
  rajzol(x7, y7)
  OLED.clear()
  rajzol(x6, y6)
  OLED.clear()
  rajzol(x5, y5)
})

```

A 14. ábrán egy pár micro:bit segítségével OLED kijelzőre kirajzolt kulcskocka látható.



14. ábra. Pálcikaemberke kulcskockái OLED kijelzőn

IRODALMI HIVATKOZÁSOK

- [1] https://hu.wikipedia.org/wiki/Anim%C3%A1ci%C3%B3s_film
- [2] <https://makecode.microbit.org/>
- [3] <https://github.com/JorgePe/microbit>
- [4] <https://pivotanimator.net/>
- [5] Szirmay-Kalos L., Antal Gy., Csonka F. *Háromdimenziós grafika, animáció és játékfejlesztés*, ComputerBooks, Budapest, 2006.
- [6] Kovács L. I. *Számítógépes grafika*, Scientia Kiadó, Kolozsvár, 2009.