

LEGO és micro:bit kéz a kézben

I. rész

„Egyben hiszünk, kéz a kézben indulunk.”
(John Paul)

1. A Ring:bit Bricks Pack bemutatása

Az 1. ábrán látható Elecfreaks Ring:bit Bricks Pack egy igen hasznos micro:bit kiterjesztés, amely lehetővé teszi, hogy LEGO TECHNIC® elemekhez csatlakoztassuk a micro:bitet, és ezáltal egyszerű robotokat építsünk.

A csomag tartalmaz egy nyomtatott huzalozású lemezt (NYÁK, PCB) a micro:bit számára. Ez kibővíti a micro:bit három általános rendeltetésű ki/bemeneti portját (GPIO), és átalakítja a P0 / P1 / P2 portokat egy közös, a szervomotorokhoz szükséges GVS (Ground / Voltage / Signal – Földelés / Feszültség / Jel) terminállá. A NYÁK 3 darab AAA elemmel működik, és 5 csavarral kell felszerelni a micro:bit 0, 1, 2, 3V, GND pinjeire. A NYÁK kimenete 3 darab GVS port: 0VG, 1VG és 2VG. Ezekhez csatlakoztathatók a szervomotorok. A NYÁK LEGO-elemekhez rögzíthető.

A csomag két, LEGO-elemekhez építhető szervomotort tartalmaz. A szervomotor olyan motor, amely lehetővé teszi a motortengely pontos pozíciójának, valamint a fordulatszámnak és/vagy a gyorsulásnak az irányítását. A szervomotor bemenőjele villamos feszültség vagy áram, kimenő jele szögelfordulás vagy mechanikai elmozdulás. Jellemzője a rendkívül gyors indulás és forgásirányváltás, valamint egy adott pozícióba történő pontos beállítás. A szervomotor tehát mindig ismeri az aktuális pozícióját és a célpozíciót. A szervomotorok vezérlése 3 huzalon történik a GVS szabvány szerint. Itt barna vagy fekete a föld (G), piros a feszültség (V), sárga a jel (S). A szervomotor vezérlése egy impulzusszélesség-modulált (PWM) jel küldésével történik a vezérlőhuzalon keresztül.

A Ring:bit Bricks Pack egy 10 LED-es öntapadós RGB színváltós LED szalagot is tartalmaz. A LED fénykibocsátó dióda (Light Emitting Diode), a rákapcsolt egyenfeszültség hatására fényt bocsát ki. A LED-et 1955-ben találták fel, ekkor vizsgálták a gallium-arszenid félvezetők által elektromos áram gerjesztés hatására kibocsátott infravörös fényt. 1962-ben fejlesztették ki a gyakorlatban is használható látható fényű diódát. A színes LED-világítás speciális lehetősége az RGB világítás. Ekkor egy piros (R), egy zöld (G) és egy kék (B) LED és egy speciális vezérlő segítségével a szívrányt összes színe előállítható. A három alapszínű LED-et különböző fényerővel működtetjük, így jön létre a kellő fényhatás. A szintén GVS szabvány szerint vezérelhető LED szalag szegmensekből épül fel. A szegmensek egymással sorba kötött LED-ekből állnak, a szegmenseket viszont párhuzamosan vagy akár sorosan tovább lehet kötni, így a LED szalag bővíthető és a szegmensek vágópontjainál vágható is.



1. ábra: A Ring:bit Bricks Pack

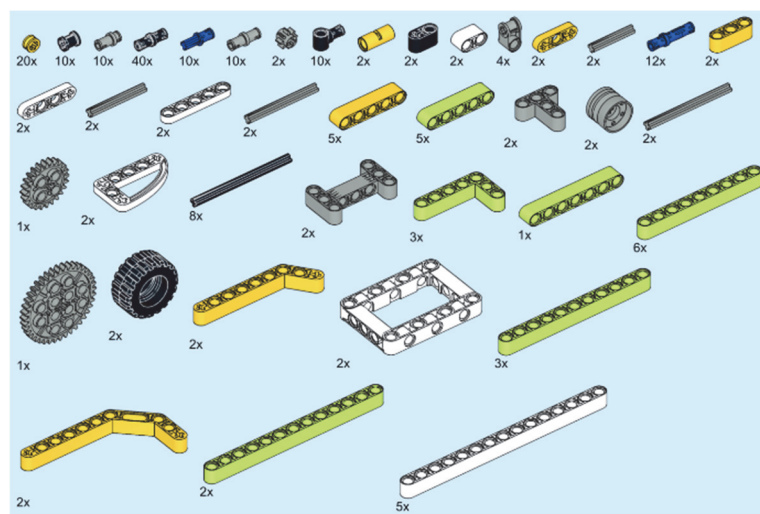
A fentiekén kívül a csomag tartalmaz 6 befőttesgumit (ezek például a katapulthoz kellene) és egy kis csavarhúzó is.

Természetesen a Ring:bit Bricks Pack 206 LEGO TECHNIC-szerű elemeket is tartalmaz, amelyek segítségével különböző robotokat, alakzatokat építhetünk, köztük közlekedési lámpát, intelligens autót, hőmérséklet-szabályozott ventilátort, intelligens asztali lámpát, ablaktörlőt, katapultot. A https://www.electfreaks.com/learn-en/microbitKit/ring-bit_bricks_pack/index.html honlapon szerelési útmutatókat, kódolási segédleteket találunk.

2. LEGO-elemek

Sajnos sem a dobozban, sem a honlapon nem található meg a LEGO TECHNIC kompatibilis elemek leltára, így ezt elkészítettük (2. ábra).

A szett 2 kereket, 3 típusú fogaskereket, 9 típusú csatlakozóelemet és pint, 5 típusú tengelyt, 20-féle lapos TECHNIC elemet tartalmaz.



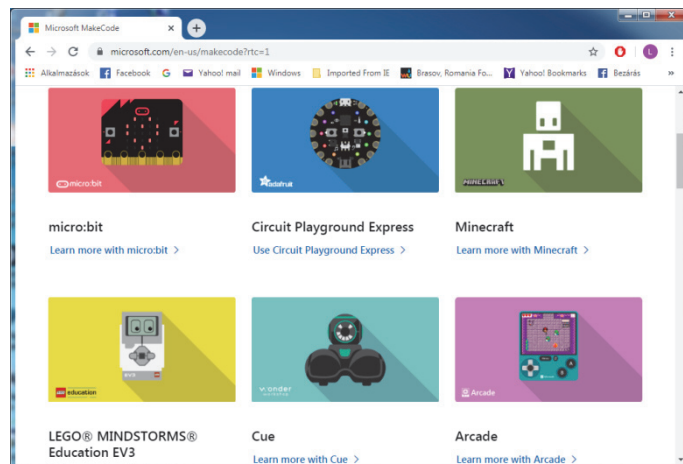
2. ábra: LEGO TECHNIC kompatibilis elemek leltára

3. A Ring:bit Bricks Pack programozása

A Ring:bit Bricks Pack esetén a micro:bit programozása is Microsoft MakeCode segítségével történik. A Microsoft MakeCode egy olyan webes felület, amely a számítógépek és külső eszközök (pl. micro:bit, LEGO® MINDSTORMS® Education EV3, más áramkört és robotikai hardverek stb.) kódolását, programozását segíti elő. Az oktatási céloknak tökéletesen megfelel, a változatos alkalmazások robotika klubok működését, tudományos órátartást, gyakorlatok levezetését teszi lehetővé nagyon egyszerűen, egységes keretben.

A felület érdekessége, hogy vizuális paradigmára épülve grafikusan tudjuk összerakni a programokat, ám a háttérben JavaScript kódot is generál, tehát meglesz a teljes imperatív program is, így a felület két legfontosabb része a Blocks és a JavaScript.

A MakeCode felület akármilyen böngészőből egyszerűen elérhető a <https://www.microsoft.com/makecode> honlapon (3. ábra).



3. ábra: A MakeCode honlapja

A MakeCode felületen a programozás azt jelenti, hogy a működés szerint csoportosított *Eszköztár*ból blokkokat választunk ki, majd azokat a *Munkaterület*en összerakjuk más blokkokkal. Így építjük fel a programot.

A MakeCode felület a micro:bit programozását eseményorientáltan, eseményvezérléssel oldja meg.

Az eseményvezérelt programozás lényege, hogy a teljes program vagy ennek részei, ágai nem szekvenciálisan, előre meghatározott sorrendben futnak le, hanem a vezérlést bizonyos külső vagy belső események határozzák meg, indítják el.

A program így nem más, mint az események bekövetkezésére válaszul végrehajtott eljárások (úgynevezett *eseménykezelők*) halmaza, amelyek nagyrészt egymástól függetlenül dolgoznak.

Az eseményvezérelt architektúrák ez utóbbi okból kifolyólag rendkívül robusztusak és könnyen átláthatók.

Programozás során természetesen nem minden eseményre kell reagálnunk, csak azokra, amelyek a program szempontjából elengedhetetlenek. Így a programozó mondja meg, hogy melyek azok az események, amelyekre egy forrásobjektumon a programnak reagálnia kell, vagyis a programozó készíti el az egyes eseményekhez az eseménykezelőket.

Először tehát kiválasztjuk a programunkhoz szükséges eseménykezelőket, ezeket felhelyezzük a *Munkaterület*re, majd beletesszük a szükséges utasításokat. Ha *Blokk* nézetben vagyunk, akkor egyszerűen az egérrel behúzzuk, ha *JavaScript* nézetben vagyunk, vagy beírjuk, vagy behúzzuk az egérrel. Ekkor vigyázzunk a zárójelekre!

3.1. A motorok programozása

A szervomotorok programozásához először be kell töltenünk a *Kiterjesztések* közül a *servo* és/vagy a *RingbitCar* bővítményt.

A *servo* bővítmény (4. ábra) a szervomotorok vezérlését látja el.

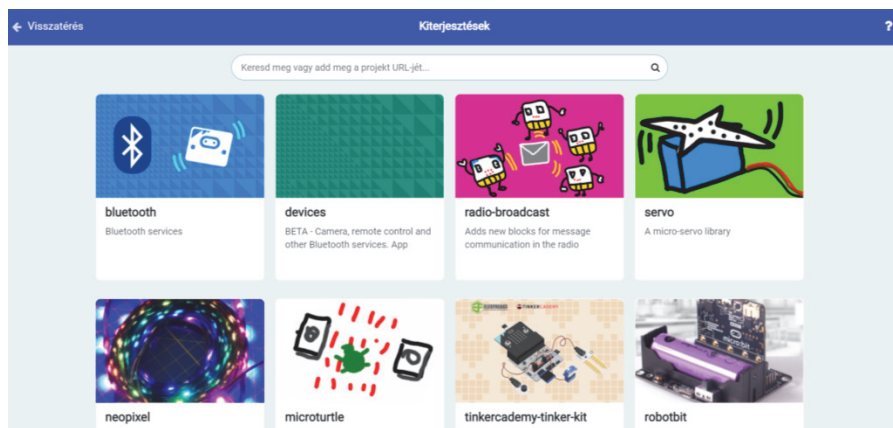
A *Ringbit Car* csomagot és a *RingbitCar* bővítményt (6. ábra) az Elecfreaks cég fejlesztette a Tinkercademy kisebb segítségével. Így kétkerekű intelligens autót valósít meg, amely

könnyen programozható önálló vagy távirányítós működésre. Az autó vonalakat, fényeket tud követni, akadályokat kerülhet el, rajzolhat, és számos egyéb feladatot is meg tud valósítani.

Mindkét bővítmény lehetővé teszi a motorok nagyon könnyű programozását. Tekintsük át a bővítmények blokkjait.

A *servo* bővítménynek (*Servos* menü – 5. ábra) egy pozicionáló, két működési és három konfigurációs blokkja van, ezek a következők:

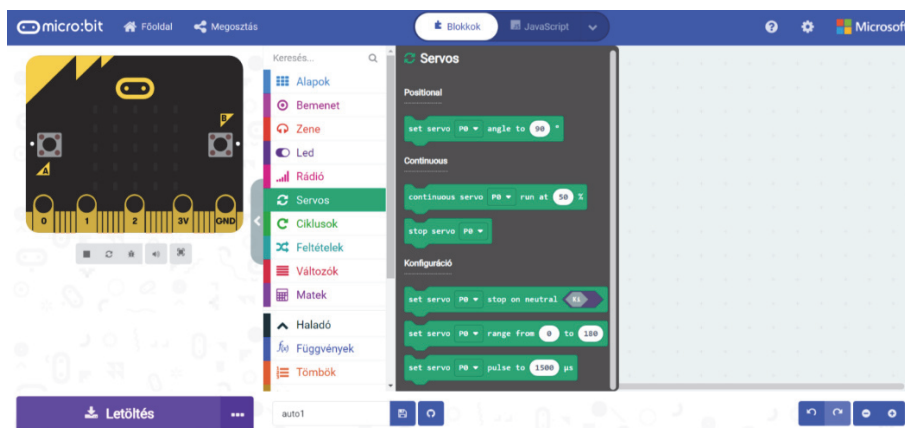
- **set servo PIN angle to SZÖG°:** Az adott SZÖG-be fordítja a szervomotort. A 90 fokos helyzet az alap vagy semleges helyzet. A szervotengely balra forgatásához (az óramutató járásával ellentétes irányba) 0 és 90 közötti fokszámot használhatunk, a jobbra forgatásához (az óramutató járásával megegyező irányba) pedig 90 és 180 közötti fokszámot. A PIN a P0, P1 vagy P2 lehet, és azt a portot jelöli, amelyre a szervomotor csatlakoztatva van.



4. ábra: A *servo* és a *neopixel* bővítmény

- **continuous servo PIN run at SEBESSÉG%:** A megadott SEBESSÉG-gel folyamatosan forgatni kezdi a PIN-re csatlakoztatott szervomotort. A folyamatos forgatású szervomotor mind az óramutató járásával megegyező irányba, mind az óramutató járásával ellentétes irányba foroghat. Az óramutató járásával megegyező irányú forgatáshoz pozitív számot adjunk meg, vagyis $SEBESSÉG > 0$, az óramutató járásával ellentétes irányú forgatáshoz pedig negatív százalékos értéket használjunk.
- **stop servo PIN:** Megállítja a PIN-re kötött szervomotort az aktuális helyzetben.
- **set servo PIN stop on neutral KI/BE:** Megálláskor a szervomotort a semleges helyzetbe (90°) hozza, ha be van kapcsolva.
- **set servo PIN range from SZÖG_MIN to SZÖG_MAX:** Beállítja a szervomotor minimális és maximális forgási szögét. Ha korlátozni szeretnénk a szervomotor forgási tartományát, beállíthatjuk a tartomány minimális és maximális forgási szögét. A minimális szöghatárt 0 és 90 fok között határozták meg, a maximálist pedig 90 és 180 között.

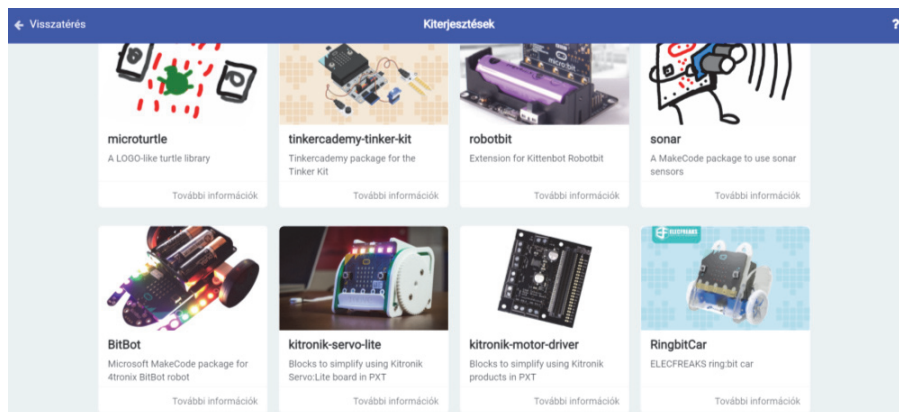
- set servo PIN pulse to SZÁM μ s: A blokk segítségével beállíthatjuk a szervomotor impulzusidejét. A szervomotor a forgatás szögét vagy sebességét az impulzusjelből határozza meg. A digitális impulzusjelnek csak két értéke lehet: magas és alacsony. Éppen ezért az időtartamot (az impulzus bekapcsolási idejét) használjuk arra, hogy megmondjuk a szervomotornak, hogy mennyit forogjon. A legtöbb hobbiszervó 1 milliszekundum és 2 milliszekundum közötti impulzust használ a forgatás vezérlésére. Az 1 milliszekundumos impulzus azt jelenti, hogy a szervomotor tengelye teljesen balra fordul, ezt általában 0 fokos helyzetnek hívják. A 2 milliszekundumos impulzus arra utasítja a szervót, hogy forduljon teljesen jobbra, vagyis a 180 fokos helyzetbe. A 0,5 milliszekundumos impulzus az az időtartam, amely arra utasítja a szervót, hogy semleges helyzetben maradjon. Így mikroszekundumokat használva (μ s) a semleges helyzet 1500 mikroszekundum, a szervomotor a tengelyt egészen balra forgatja 1000 mikroszekundumnyi impulzussal, jobbra pedig az impulzus 2000 mikroszekundum. Folyamatos forgatású szervomotor esetén a tengely nem áll meg egy adott helyzetben. Ebben az esetben a szervónak küldött impulzus nem pozíciót jelent, hanem sebességszámot. A tengely folyamatosan forog, ha 1500 milliszekundumtól eltérő impulzusszámot küldünk a szervóra. Az 1000 mikroszekundum teljes sebességgel való forgást jelent balra, a 2000 mikroszekundum pedig teljes sebességgel való forgást jobbra. Az 1500 mikroszekundum itt is semleges.



5. ábra: A Servos menü

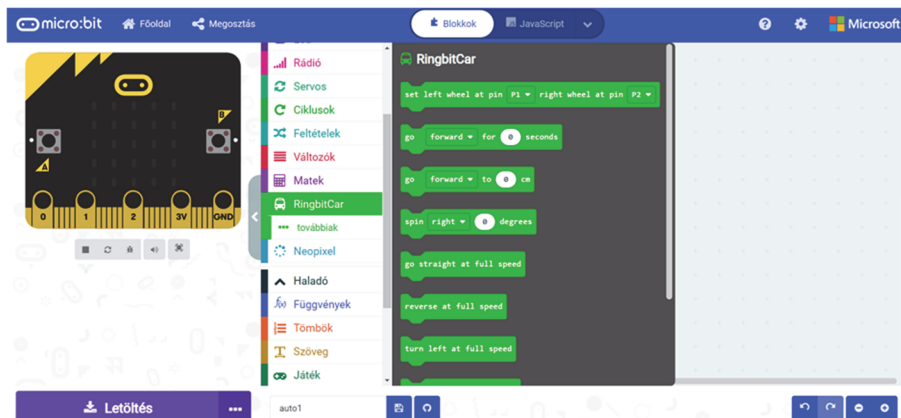
A RingbitCar bővítmény (RingbitCar menü – 7. ábra) 10 blokkot, valamint 2 további blokkot tartalmaz, ezek a következők:

- set left wheel at pin PIN1 right wheel at PIN2: A bal kerék szervomotorját a PIN1-re, a jobb kerékét pedig a PIN2-re állítja be. Itt a listában P0-tól P16-ig vannak a pinek felsorolva.
- go FORWARD/BACKWARD for MÁSODPERC seconds: A megadott MÁSODPERC-ig előre (FORWARD) vagy hátra (BACKWARD) megy.



6. ábra: A RingbitCar bővítmény

- go FORWARD/BACKWARD for CENTIMÉTER cm: A megadott CENTIMÉTER-ig előre (FORWARD) vagy hátra (BACKWARD) megy. Figyelem! A CENTIMÉTER nem a kerék által megtett távolságot jelenti (ez függ a kerék méretétől), hanem csak a motor tengelyének a forgáshosszát.
- spin RIGHT/LEFT FOK degrees: A megadott FOK-ig jobbra (RIGHT) vagy balra (LEFT) fordul. Figyelem! A FOK nem a kerék által megtett szöveget jelenti (ez függ a kerék méretétől), hanem csak a motor tengelyének a forgásszögét.
- go straight at full speed: „Teljes gőzzel előre!”, vagyis a motor menjen egyenesen teljes sebességgel.
- reverse at full speed: Hátramenet teljes sebességgel.
- turn left at full speed: A motor forduljon balra teljes sebességgel.
- turn right at full speed: A motor forduljon jobbra teljes sebességgel.
- brake: „Állj!”, vagyis a motor álljon le.
- set left wheel speed at SEBESSÉG1 right wheel speed at SEBESSÉG2: A bal kerék sebességét SEBESSÉG1-re, a jobb kerék sebességét pedig SEBESSÉG2-re állítja.
- tracking state is JELEK: Speciális logikai feltétel, amely a vonalkövetés állapotát vizsgálja. Ez külső-külső, külső-belső, belső-külső vagy belső-belső lehet. Figyelem! Ide szükséges a vonalkövető érzékelő!
- ultrasonic distance in unit EGYSEG: A megadott EGYSEG-ben (mm, cm, inch) visszatéríti az ultrahangos érzékelő által mért távolságot. Figyelem! Ide szükséges az ultrahangos érzékelő!



7. ábra: A RingbitCar menü

3.2. A LED szalag programozása

A LED szalag programozásához először be kell töltenünk a *Kiterjesztések* közül a *neopixel* bővítményt (4. ábra). Ez a bővítmény telepíti a rendszerbe a *Neopixel* (8. ábra) menüt, amellyel lehetőségünk lesz a 20 blokk használatára. Ezek a következők:

- STRIP legyen NeoPixel at pin PIN with DARAB leds as FORMAT: Egy értékadás-blokk a STRIP Neopixel változóra (ez testesíti meg a LED szalagot, ha több LED szalagunk van, akkor több változónk lesz). Beállítja a LED szalag csatlakozási portját, ez a P0 – P16 pin valamelyike lehet, a LED-ek DARAB-számát, valamint a LED-ek formátumát. A FORMAT RGB (GRB format), RGB+W, valamint RGB (RGB format) lehet. Az RGB szintér egy olyan additív színmodell, ami a vörös, zöld és kék fény különböző mértékű keverésével határozza meg a különböző színeket. Az elnevezése ezen három alapszín angol megfelelőinek első betűiből ered: Red (piros), Green (zöld), Blue (kék). A GRB formátum az RGB fordítottja, erre azért van szükség, mert a memóriában fordítva ábrázolják a számokat, az RGB+W pedig az RGB-hez hozzáveszi a fehér (White) értéket is.
- RANGE legyen STRIP range from SZÁM with DARAB leds: Egy értékadás-blokk a RANGE Neopixel változóra. A LED szalagról a SZÁM pozíciótól levászt DARAB számú LED-et, és ez lesz az új szalag.
- STRIP show rainbow from SZÁM1 to SZÁM2: A SZÁM1 és a SZÁM2 által meghatározott szivárványt jelentet meg a STRIP LED szalagon. A két paraméter a színek intervallumát határozza meg.
- STRIP show color SZÍN: A teljes STRIP LED szalagot a megadott SZÍN-re állítja. A SZÍN red (piros), orange (narancs), yellow (sárga), green (zöld), blue (kék), indigo (indigó), violet (ibolya), purple (lila), white (fehér), black (fekete), vagy saját RGB (HSL) kódok szerint beállított szín lehet. Feketére állítva töröljük a LED szalagot, vagyis az összes LED kialszik.

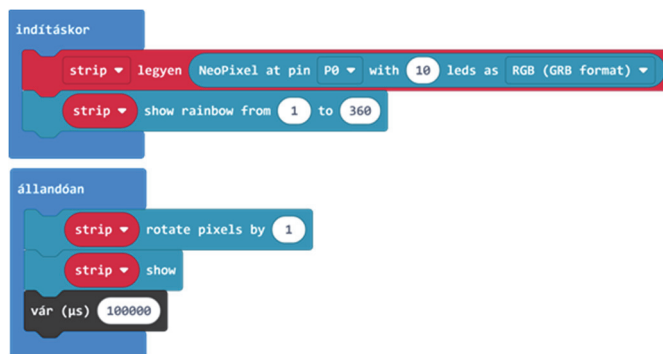


8. ábra: A Neopixel menü

- STRIP show bar graph of SZÁM1 up to SZÁM2: A STRIP LED mátrixon megjelenít egy SZÁM1, SZÁM2 paraméterekkel rendelkező oszlopdiagramot.
- STRIP show: Megjeleníti a STRIP LED szalagon a módosításokat.
- STRIP clear: Törli a STRIP LED szalagot.
- hue SZÍN saturation TELÍTETTSÉG luminosity FÉNYERŐ: A HSL színtérben a színárnyalat (SZÍN), TELÍTETTSÉG, FÉNYERŐ színekoordináták segítségével lehet egy színt megadni. Ez a színtér az RGB színmodell alternatív reprezentációja, amelyet az 1970-es években terveztek a számítógépes grafika kutatói, hogy jobban illeszkedjen az emberi látás színalkotó tulajdonságaihoz.
- STRIP shift pixels by SZÁM: A STRIP LED szalagon SZÁM lépésközzel eltolja a LED-ek fényeit. Például, ha az első LED piros, és a SZÁM 1, akkor a második LED lesz piros.
- STRIP rotate pixels by SZÁM: A STRIP LED szalagon SZÁM lépésközzel elforgatja a LED-ek fényeit. Az elforgatás azt jelenti, hogy az utolsóból első lesz, és az egész folyamat folytatódik.
- STRIP set pixel white LED at SZÁM1 to SZÁM2: A STRIP LED szalagon fehér LED-eket állít be a két SZÁM között.
- STRIP set pixel color at SZÁM to SZÍN: A STRIP LED szalag SZÁM indexű LED-jét beállítja az adott SZÍN-re. A SZÍN red (piros), orange (narancs), yellow (sárga), green (zöld), blue (kék), indigo (indigó), violet (ibolya), purple (lila), white (fehér), black (fekete), vagy saját RGB (HSL) kódok szerint beállított szín lehet.
- STRIP length: Visszatéríti a STRIP LED szalag hosszát.
- STRIP set brightness ÉRTÉK: Beállítja a STRIP LED szalag fényerejét (fényességét). Az ÉRTÉK egy 0–255 közötti szám lehet.
- STRIP ease brightness: Könnyed fényerőre vált.
- STRIP power (mA): Visszatéríti a STRIP LED szalag áramerősségét milliampereben.

- STRIP set matrix width SZÁM: A STRIP LED mátrix szélességét állítja be.
- STRIP set matrix color at x X y Y to SZÍN: A STRIP LED mátrix X, Y koordinátájú LED-jét beállítja a megadott SZÍN-re. A SZÍN red (piros), orange (narancs), yellow (sárga), green (zöld), blue (kék), indigo (indigó), violet (ibolya), purple (lila), white (fehér), black (fekete), vagy saját RGB (HSL) kódok szerint beállított szín lehet.
- SZÍN: Előre definiált színek konstansok. A SZÍN red (piros), orange (narancs), yellow (sárga), green (zöld), blue (kék), indigo (indigó), violet (ibolya), purple (lila), white (fehér), black (fekete) lehet.
- red R green G blue B: A megadott RGB értékek szerint határozza meg a színt. Az R, G, B 0–255 között vehet fel értékeket. Az RGB skálán egy színt az határoz meg, hogy milyen intenzitású a három komponense. Ha mindhárom 0, akkor az eredő szín fekete lesz, ha 255, akkor fehér, az összes köztes érték eredményezi a különböző árnyalatokat.

Például, ha a LED szalagot a szírvárvány színeiben szeretnénk kigyújtani, majd azt megvalósítani, hogy a színek körbe forogjanak, a 9. ábrán látható programot kell összeállítanunk.

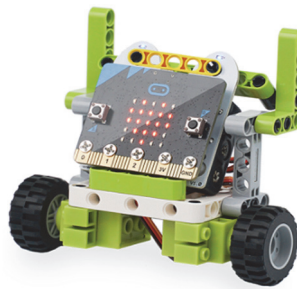


9. ábra: Forgó szírvárvány a LED szalagon

4. Egy kis mozgó robotautó

A https://www.electronics.com/learn-en/microbitKit/ringbit_bricks_pack/ringbit_bricks_pack_case_02.html útmutatásai alapján készítsük el a kis robotautót (10. ábra)!

1769-ben a francia N.J. Guinou megépítette a világ első gőzhajtású háromkerekű autóját, amelynek neve Capo Oré volt. A világ első autóját Carl Benz (Karlsruhe, 1844. november 25. – Mannheim, 1929. április 4.), német mérnök, a modern gépkocsitechnika egyik úttörője fejlesztette ki 1885 októberében. Ez is egy háromkerekű, ám gázmotoros jármű volt.



10. ábra: Egy kis robotautó

Az általunk is használt robottechnika annyiban különbözik a hagyományos autóktól, hogy a két főkerék meghajtására két különböző motort használunk, a harmadik, úgynevezett bolygókerék (amely egy egyszerű támasz is lehet) csak arra jó, hogy az autót egyensúlyban tartsa. Tehát nincs külön kormány szerkezet, amely a jobbra, illetve balra fordulást valósítja meg, hanem a két motor működtetésével kormányozzuk az autót. Az irányítás a harckocsik, tankok vezérléséhez hasonló: a két motor eltérő sebességű forgatása eredményezi a kanyarodást. Ez pont olyan, mint ahogy egy kétevezős csónakot irányítunk kormány nélkül. Például, az M5 Stuart (1941) harckocsik két Cadillac V8-as motort használtak.

Az előbbi útmutató alapján összeszerelt robotautót többféleképpen vezérelhetjük. A programoknak csak a fantázia szab határt.

**Kovács András Apor,
Kovács Árpád Apold,
Kovács Lehel István**

LEGO robotok

XXVI. rész



12. Feladat

Egy színérzékelővel ellátott robot menjen előre addig, ameddig egy fekete sávot nem érzékel. Menés közben a robot LED-jei pirosan villogjanak, megállás után váltsanak zöldre!

A feladat megoldása érdekében csatlakoztassuk a színérzékelőt a téglá 1-es portjára, és a robotunkat lássuk el két motorral.

ROBOTC-ben lehetőségünk van arra, hogy a motoroknak saját nevet adjunk, így nem a `motorA`, `motorB` konstansokkal hivatkozhatunk rájuk, hanem az általunk adott nevekkel.

Legyen az A portra kötött motor neve `jobb`, a D portra kötött motor neve pedig `bal`.

Az átnevezés úgy történik, hogy a Robot menü Motors and Sensors Setup menüsorát választjuk ki, s az innen előjövő párbeszédablak Motors fülében beállítjuk a `motorA` és a `motor D` nevét, a 177. ábra szerint.