

A vitorlás maximális sebessége

A vitorlás mozgását a hajó hossztengeleyével párhuzamos két erő határozza meg:

$$F_1 = 2A \cdot \rho \cdot c^2 \sin\beta \cdot \sin^2(\alpha - \beta)$$

a szél által kifejlesztett mozgatóerő és $F_k = C_h \cdot \rho_v \cdot v^2$ közegellenállási erő, amely négyzetesen függ a mozgás v sebességétől, függ a hajó vízbe merülő részének alakjaitól (C_h) és arányos a víz ρ_v sűrűségével. A hajó haladó mozgása Newton II. törvényének megfelelően megy végbe: $2A \cdot \rho \cdot c^2 \sin\beta \cdot \sin^2(\alpha - \beta) - C_h \cdot \rho_v \cdot v^2 = m \cdot a$, ahol m a hajó tömege és a gyorsulása. Amikor az álló hajóban alkalmas módon beállítjuk a vitorlákat - „szelet fogunk” - hajónk a vitorlán ébredő húzóerő hatására gyorsulni kezd. A növekvő sebesség két következménnyel jár: egyrészt egyre gyorsuló ütemben nő a víz ellenállása, másrészt a hajóban észlelhető szélirány és sebesség is változik. A vitorlázónak tehát a változó körülményeknek megfelelően egyre beljebb kell húznia a vitorlát, ha a szél erejét optimálisan kívánja hasznosítani. A hajó mindaddig gyorsul, míg a vitorlán ébredő húzóerő nagyobb a menetellenállásból származó fékezőerőnél. Amikor ez a két erő egyenlővé válik, a gyorsulás nulla lesz, ekkor éri el a hajó a maximális sebességet:

$$2A \cdot \rho \cdot c^2 \sin\beta \cdot \sin^2(\alpha - \beta) - C_h \cdot \rho_v \cdot v_{\max}^2 = 0 \Rightarrow v_{\max} = \sqrt{\frac{2A \cdot \rho \cdot \sin\beta}{C_h \cdot \rho_v}} \cdot c \cdot \sin(\alpha - \beta).$$

A felhasznált forrásmunkák

- 1) Bokor Péter, Teknős Péter: Felfedezők és hódítók, Móra Ferenc Könyvkiadó, Budapest, 1961
- 2) Horváth Gábor, Juhász András, Tasnádi Péter: Mindennapok fizikája, ELTE TTK Továbbképzési Csoportjának kiadványa, Budapest, 1989
- 3) Révai Nagy Lexikona, IX. Kötet, Hasonmás kiadás, Babits Kiadó, 1993
- 4) <https://hu.Wikipedia.org/wiki/Sportvitorlás>

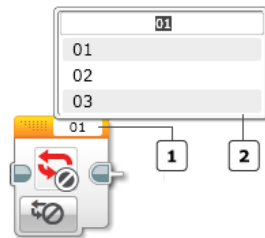
Ferenczi János, Nagybánya

LEGO robotok

XI. rész

III.1.20. A ciklusbefejező blokk

A ciklusbefejező blokk (Loop Interrupt Block) a megadott szimbolikus nevű ciklust fejezi be. Egyszerűen arra kényszeríti a vezérlést, hogy azonnal lépjen ki a ciklusból, és a program a ciklus utáni blokkal folytatódjon. A ciklusbefejező blokk a normális befejezésnél hamarabb, vagy más feltétel beteljesedésekor fejezi be a ciklust akár a cikluson belülről, akár bármilyen más, párhuzamosan futó programszekvenciából.

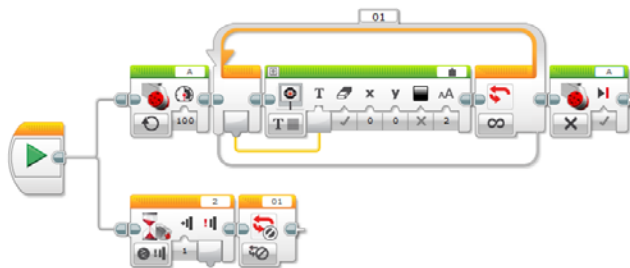


77. ábra: A ciklusbefejező blokk

Az 1. gombon állíthatjuk be a ciklus szimbolikus nevét. Ezt kiválaszthatjuk a 2-es listából is, amelyben az összes program által használt ciklus megjelenik.

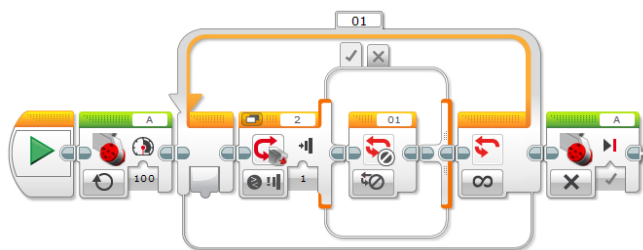
A 78. ábrán látható program két párhuzamosan futó szekvenciára bomlik. Az első beindítja a nagy motort, majd egy végtelen ciklusban kiírja a ciklusváltozó értékeit a téglaképernyőjére. Elméletileg ez a ciklus végtelen, tehát az elem lemerüléséig forogna a nagy motor.

A második szekvencia az érintésérzékelő lenyomására vár. Ha ez az esemény megtörténik, akkor életbe lép a ciklusbefejező blokk, amely leállítja a 01-es végtelen ciklust, vagyis a vezérlés a ciklus utáni utasítással folytatódik, amely megállítja a nagy motort.



78. ábra: Végtelen ciklus leállítása kívülről

A 79. ábra azt mutatja be, hogy hogyan használhatjuk a ciklusbefejező blokkot a ciklus belsejében.



79. ábra: Végtelen ciklus leállítása belülről

III.1.21. Az elágazás

A 79. ábrán látható program akkor fejezi be a ciklust, ha megnyomjuk az érintésérzékelőt. Ennek a tesztelésére egy logikai elágazást használtunk, amely beolvassa az érintésérzékelő állapotát, és ha ez le volt nyomva, akkor átadja a vezérlést a ciklusbefejező blokknak. Ha nem volt lenyomva, akkor nem történik semmi.

Az elágazás utasítások valósították meg először a futás pillanatában történő döntést bizonyos feltételek függvényében. Ennek a megvalósításnak köszönhető, hogy ugyanaz az algoritmus különböző bemeneti értékek, illetve részeredmények alapján, önmagából, más-más lineáris utasítássorozatot hajt végre. Ettől az újítástól vált a lineárisan programozható algoritmust végrehajtó gép számítógéppé. Az elágazás megvalósítása Neumann Jánosnak tulajdonítható.

Az egyszerű elágazás egy logikai kifejezés igaz vagy hamis értékének függvényében vagy az egyik ágon, vagy a másik ágon hajtja végre a programban található blokkokat.

A többszintű elágazást megvalósító utasítást Wirth és Hoare vezették be 1966-ban. Szemantikai szerepe: több alternatíva közül egynek a kiválasztása.

A végrehajtandó alternatíva kiválasztása egy *szelektornak* nevezett kifejezés alapján történik, és a szelektor-kifejezés megfelelő értéke alapján történik az elágazás. A szelektor-kifejezés megfelelő értékeit *eseteknek* (Case) nevezzük.

Az *elágazás blokk* (Switch block) komplex utasítás-blokkja a LEGO MINDSTORMS EV3-nak.

A blokk úgy működik, hogy belépéskor elvégzi a kiválasztott tesztet, majd az eredmény függvényében végrehajtja a megfelelő esetet. Mindig egy és csakis egy eset fog végrehajtódni, ezután a program az elágazás blokk utáni blokkal fog folytatódni.

Ha a blokkban egy esetet üresen hagyunk, akkor azon az ágon a program nem fog csinálni semmit.

A blokkban egy esetet megjelölhetünk *alapértelmezettnek* (Default Case) is, ez akkor fog végrehajtódni, ha a szelektor értéke nem talál egyetlen más esettel sem.

A 80. ábrán az elágazás blokk egyik változatát látjuk. Hasonló jelenik meg, amikor a palettáról behúzzuk a blokkot a program felületére.

Az 1. gomb a *mód szelektor*, a 2. a *port szelektor*, a 3. pedig a megfelelő bemeneti értékek megadására szolgál.

Az 1-es módszelektor segítségével ki tudjuk választani, hogy a blokk milyen tesztet hajtson végre az elején.

Tesztelni tudunk egyszerűen egy szöveges, numerikus vagy logikai értéket, vagy tesztelni tudunk egy érzékelőn mért értéket.

Az érzékelőket *mérés* (Measure) vagy *összehasonlítás* (Compare) módokban tudjuk használni.

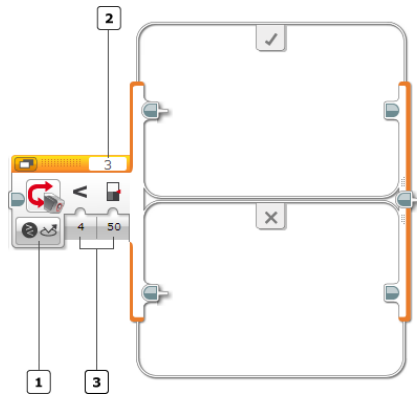
Mérés módban a blokk a következőket tudja tesztelni:

- téglagombok;
- színérzékelő;
- infravörös érzékelő.

Összehasonlítás módban a blokk a következőket tudja tesztelni:

- téglagombok;
- színérzékelő;

- infravörös érzékelő;
- motorforgás;
- időzítő;
- érintésérzékelő;
- üzenet.




80. ábra: Az elágazás blokk

Megjegyezzük, hogy a várj blokkal ellentétben az elágazás blokkal nem vár addig, ameddig a szenzoron megjelenik a megfelelő érték. A blokk csak a tesztet hajtja végre, a program végrehajtása azonnal tovább is megy. Ha tehát azt szeretnénk, hogy például a 81. ábrán látható program a „Lenyomva” szöveget jelentesse meg a téglá képernyőjén, akkor jobb, ha az érintésérzékelőt már a program indítása előtt lenyomjuk és nyomva is tartjuk.

Mivel az elágazás blokk grafikailag is nagyon komplex lehet (sok más blokk kerülhet bele), a jobb átláthatóság kedvéért kétfajta megjelenítési mód közül választhatunk.

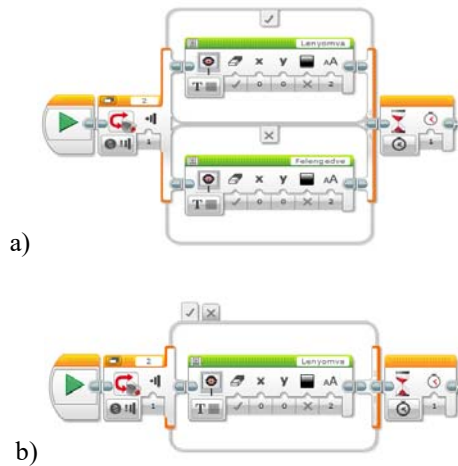
A *nyílt nézet* (Flat View) a 80. ábrának megfelelően egymás alatt jeleníti meg az összes esetet. Nyilvánvaló, hogy sok esetben nem fog kiférni a képernyőre a blokk, viszont könnyen áttekinthető.

A *füles nézet* (Tabbed View) minden esetnek egy fülecskét hoz létre, és egyszerre csak egy eset látható a képernyőn, ha át akarunk térni egy másik esetre, akkor fülecskét kell váltani. Így a blokk könnyen kifér a képernyőre, csak nem annyira áttekinthető.

A két nézet között a blokk bal felső sarkában található  gombbal lehet váltani (Switch to Tabbed View / Switch to Flat View).

Nyilvánvaló, a program működését nem befolyásolja a két nézet, csak a grafikáját. Annak ellenére, hogy a füles nézetben egyszerre csak egy eset látszik, a többi is ugyanúgy része a programnak.

Szintén a grafikus kinézethez tartozik, hogy a blokkot tetszés szerint át lehet méretezni a területén lévő jelölő körök és négyzetek segítségével (82. ábra).

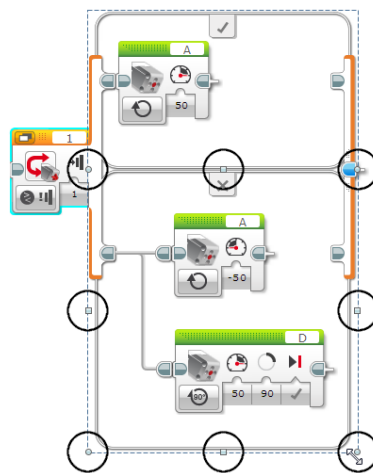


81. ábra: a) nyílt és b) füles nézet

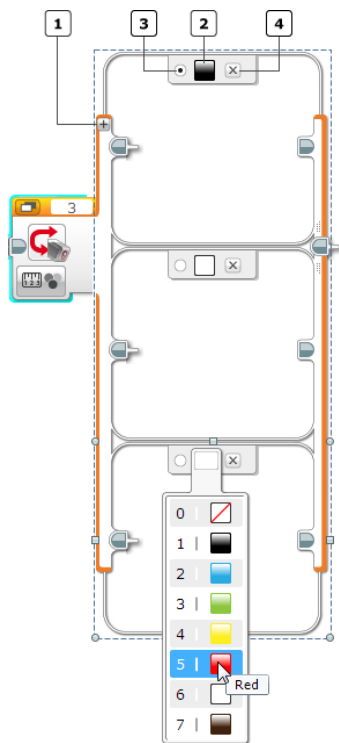
A logikai teszt Igaz ágát a „pipa” , a Hamis ágát az „X” jelöli.

A 79. és 81. ábrákon megfigyelhettük, hogyan működik az elágazás egyszerű logikai esetekre. Nézzük most meg, hogyan működik numerikus értékekre.

Alapértelmezetten, ha ráhúzzuk a blokkot a programozási felületre, két eset jelenik csak meg. Például, ha numerikus értékek esetén több esetet szeretnénk ágaztatni, akkor a blokkhoz újabb eseteket kell hozzáadnunk. Amint a 83. ábrán is látjuk, a blokk grafikus elemei lehetőséget biztosítanak újabb esetek hozzáadására, esetek kitörlésére, alapértelmezett eset beállítására stb.



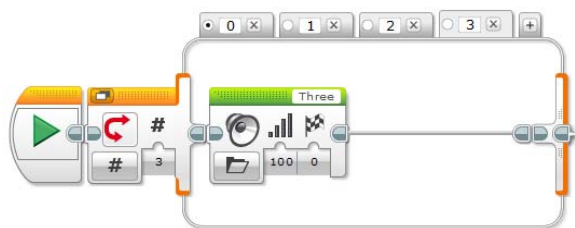
82. ábra: A blokk átméretezése



83. ábra: *Esetek hozzáadása, törlése, kezelése*

Az 1-es gomb segítségével új esetet adhatunk hozzá, a 2-es gombbal beállíthatjuk az esetet, a 3-as gomb segítségével mondhatjuk meg, hogy melyik eset legyen az alapértelmezett, a 4-essel pedig kitörölhetjük az esetet.

A 84. ábrán látható blokk a numerikus bemenet függvényében angolul kimondja a számjegyeket 0-ás, 1-es, 2-es, vagy 3-as számjegyek esetén. Mivel a 0-ást alapértelmezett esetnek állítottuk be, ezért ha 3-nál nagyobb számjegyet adunk meg, ez egyik esetre sem talál, csak a nullást fogja angolul kimondani.

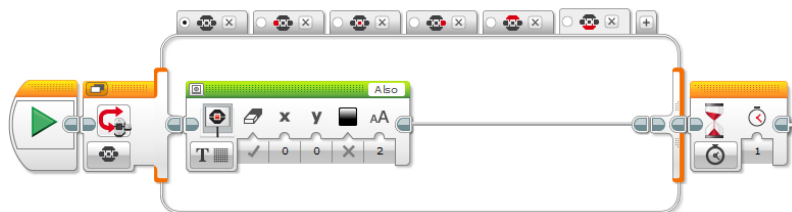


84. ábra: *Numerikus esetek*

Ugyanez történik, ha a blokkot szöveges üzemmódban használjuk. Ekkor a beérkező szöveget hasonlítja össze, s találhat esetén lefut a megfelelő esethez tartozó kód.

Mérés módban a blokk tesztelni tudja a téglagombok, színérzékelő, vagy az infravörös érzékelő szolgáltatott értékeket.

A 85. ábrán látható program a téglagombokat teszteli le, és kiírja, hogy a bal, a jobb, a középső, a felső, vagy az alsó gomb volt-e lenyomva. Alapértelmezett esetben a képernyőn a „Nincs gomb” felirat jelenik meg.



85. ábra: Téglagombok tesztelése

Hasonlóan tudunk eljárni a színérzékelő vagy az infravörös érzékelő által visszaszolgáltattott értékekkel is. Mérés módban az infravörös érzékelő csupán a távirányító gombjainak értékeit tudja visszaszolgáltatni, ezeket lehet tesztelni.

A színeket a 83. ábrán látható módon tudjuk beállítani.

Legösszetettebb tesztek az összehasonlítás módban elérhető tesztek. Ezek a tesztek összehasonlításokon alapulnak, és igaz vagy hamis értékeket visszatérítő kétesetes tesztek. Itt a blokk tesztelni tudja, ha egy megadott téglagomb volt-e megnyomva, vagy lenyomtuk több megadott gombot együttesen. Tesztelni tudja, hogy a színérzékelő adott szint, vagy színeket érzékelt-e, a háttér vagy a visszavert fényerősség elért, meghaladott-e egy adott értéket, netán egyenlő, vagy nem egyenlő-e egy adott értékkel. Bemenetként megadhatjuk az összehasonlítási műveletet (0 – egyenlő, 1 – nem egyenlő, 2 – nagyobb, 3 – nagyobb, vagy egyenlő, 4 – kisebb, 5 – kisebb vagy egyenlő), valamint a küszöbértéket, amihez hasonlítjuk az érzékelő által visszaszolgáltattott értéket.

Az infravörös érzékelő esetén is megadhatjuk most már az erre vonatkozó összes mód (*közelségi mód*, *irányjeladó mód* és *távirányító mód*) szerinti küszöbértéket és összehasonlítási műveletet.

Ha a motor forgását szeretnénk tesztelni, beállíthatjuk a küszöbértéket, a relációs műveletet, valamint azt, hogy szög, fordulatszám, vagy erősség értéket tesztelünk.

Ha egy időzítő értékét akarjuk tesztelni, akkor beállíthatjuk az időzítőt (ez az érték 1-től 8-ig változhat; 8 időzítőt tud kezelni a LEGO Mindstorms), a relációs műveletet, valamint a küszöbértéket.

Ha az érintésérzékelőt teszteljük, beállíthatjuk az érintés módját, hogy *benyomott* (0 – Pressed), *felengedett* (1 – Released) és *ütközött* (2 – Bumped) legyen az állapota.

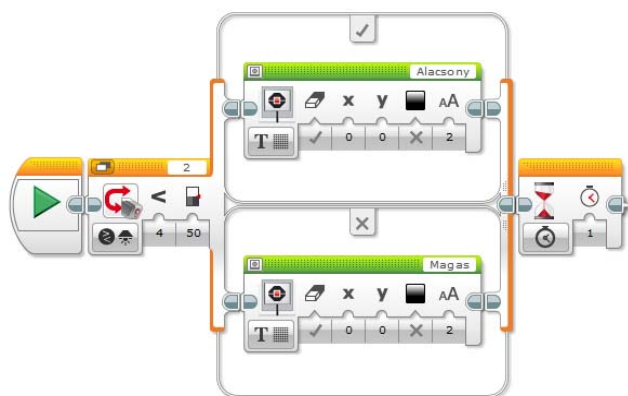
A fenti esetek mindegyikében, ha valamilyen érzékelőt tesztelünk, akkor a blokkon megjelenik a *portszelektor* is, ahol beállíthatjuk az érzékelő csatlakozási portját.

Ha üzenetet tesztelünk, akkor a beérkezett üzenet szöveges, numerikus vagy logikai lehet. Az üzenet Bluetooth csatornán érkezik. Szöveges üzemmódban azt állíthatjuk be,

hogy a beérkezett üzenet legyen egyenlő vagy nem egyenlő egy megadott szöveggel. Numerikus üzenet esetében a relációs műveletek (0 – egyenlő, 1 – nem egyenlő, 2 – nagyobb, 3 – nagyobb vagy egyenlő, 4 – kisebb, 5 – kisebb vagy egyenlő) valamelyikét állíthatjuk be, valamint a küszöbértéket.

Logikai üzenet esetén közvetlenül a tesztesetek valamelyike jön be az igaz vagy a hamis ágon.

A 86. ábrán látható elágazás blokk a színérzékelő által mért háttérvilágítást hasonlítja össze az 50-es értékkel. Ha ennél kisebb, kiírja, hogy „Alacsony” a háttérvilágítás, ha nagyobb vagy egyenlő, akkor pedig kiírja, hogy „Magas” a háttérvilágítás.



86. ábra: Háttérvilágítás tesztelése

Kovács Lehel István

Érdekes informatika feladatok

XLII. rész

A maja naptár

A 2016/2017-es Nemes Tihamér Országos Informatikai Tanulmányi Verseny programozás kategóriájának 1. korcsoportjában (5–8. osztály) szerepelt az alábbi feladat:

Maja naptár

A maják többféle naptárt is használtak történelmük során. A **tzolkin** naptár évei 260 naposak, amelyek 13 darab húsznapos hónapból állnak. A **haab** naptár évei 18 húsznapos hónapból és az év végén hozzátett 5 további napból állnak. 52 **haab** év pontosan megegyezik 73 **tzolkin** évvel.