

LEGO® Education SPIKE™ Prime segédlet

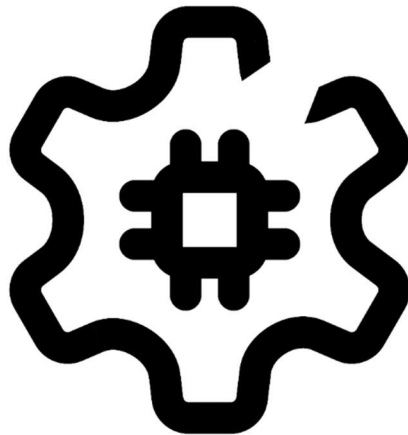
LEGO® Education SPIKE™ 3.x.x verziószámú programhoz.

Szenzorok használata – alapok

szín-, távolság-, erő (nyomás) és gyro (elfordulás, gyorsulás) szenzor.

Dokumentum verzió: v_01

Utolsó módosítás: 2026. május 17.



STEM Kuckó

Schlepp Péter 2026.

A LEGO® és SPIKE™ a LEGO Csoport vállalatainak védjegye, amely nem szponzorálja, és nem hagyja jóvá ezt a dokumentumot!


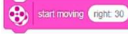
Felelősség kizárása: a dokumentumban leírt információk pontosságáért, az esetleges elírásokért, valamint a tartalom használatából eredő közvetlen vagy közvetett károkért a készítő semmilyen felelősséget nem vállal!



Bevezetés


A motorvezérléssel megvalósított összetett, különösen hosszabb, több részből álló útvonal esetén tapasztalhatjuk, hogy az adott program futtatása során sok esetben teljesen más utat jár be a robot, mint az általunk várt illetve amelyet az előző futtatás(ok) során a robot megtett. (Erről több a **02_spike-v3_basic_move.pdf** segédletben olvasható.)

Ezeket a hibákat – amennyiben a robot pályája, környezete erre lehetőséget ad – szenzorok, elfordulásakor a gyro (elfordulás) szenzor – használatával lehet elviselhető mértékűre csökkenteni.

A szenzorok használatának elmélete röviden:

A robotot folyamatosan mozgatjuk előre/hátra  vagy forgatjuk  a **start moving** blokkal. Mivel a mozgás/forgás folyamatos, gondoskodni kell annak megállításáról. Ehhez használjuk az adott szenzort, úgy, hogy várunk arra, hogy a szenzor által mért/érezékelt érték elérje az általunk beállított értéket. Amint ez megtörtént leállítjuk/kikapcsoljuk a motorokat.

A várakozást a  **wait until..** blokkal tudjuk elérni. Ez a blokk egy olyan paramétert vár, amely igaz vagy hamis értéket ad vissza. Ameddig az érték hamis, a program vár és a  **wait until..** blokk előtti blokk hajtódik végre. Amikor a paraméter igaz értéket ad vissza, a program futása a **wait until..** blokk utáni blokkra ugorva folytatódik.

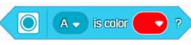
A  **wait until..** által várt paraméter értéke színszenzor esetén lehet adott szín vagy a felületről visszaverődött fény értéke, távolságérzékelő szenzor esetén az akadály távolsága, erő (nyomásérzékelő) szenzor esetén a megnyomás ténye vagy erőssége, gyro (elfordulás, gyorsulás) szenzor esetén az elfordulás mértéke.

A szenzorok által mért, érzékelt értékekről több a **01_spike-v3_block_guide_hu.pdf** című segédlet **Szenzor (érzékelő) blokkok – BASIC** részében található!

FONTOS! Minél kisebb sebességgel forgatjuk, mozgatjuk a robotot, annál pontosabban lesz az érzékelés, annál közelebb áll meg a robot az általunk beállított/kívánt pozícióhoz, értékhez! Teszteléssel kell meghatározni azt a sebesség értéket, ami a megfelel az általunk támasztott elvárásoknak! (gyorsaság vagy pontosság)



Színérzékelő használata

Robot mozgatása adott szín vagy visszavert fényérték érzékeléséig.

Színérzékelő szenzor használata esetén a  **wait until..** blokk paramétereként használhatjuk az **is color..**  blokkot, amely igaz értéket ad vissza, ha a színérzékelő a beállított színt érzékeli.



The image shows a Scratch script on the left and a console output on the right. The script starts with 'when program starts', followed by 'set movement motors to C+D', 'set movement speed to 20 %', 'start moving' (up arrow), 'wait until' (with 'is color' block set to red), 'stop moving', and 'stop and exit program'. The console output reads: 'A robot előre haladva piros szín érzékelésekor megáll!', 'Blokkok:', 'Mozgató motorok beállítása C+D', 'Motor sebesség beállítás 20%-ra.', 'Folyamatos mozgás előre', 'Várakozás, amíg a színérzékelő szenzor piros színt érzékel.', 'Motorok/mozgás leállítása', 'Program leállítás és kilépés a programból'.

Ha nem konkrét színt, hanem a felületről visszavert fény értékét szeretnénk használni a robot vezérléséhez, akkor a  **wait until..** blokk paramétereként használhatjuk az **reflection..**  blokkot, amely igaz értéket ad vissza, ha a színérzékelő a beállított értéknél **nagyobb, egyenlő vagy kisebb**, a felületről visszavert fény intenzitást érzékel.



The image shows a Scratch script on the left and a console output on the right. The script starts with 'when program starts', followed by 'set movement motors to C+D', 'set movement speed to 20 %', 'start moving' (up arrow), 'wait until' (with 'reflection' block set to '>' and '50 %'), 'stop moving', and 'stop and exit program'. The console output reads: 'A robot előre haladva 50%-nál nagyobb visszavert fényérték érzékelésekor megáll!', 'Blokkok:', 'Mozgató motorok beállítása C+D', 'Motor sebesség beállítás 20%-ra.', 'Folyamatos mozgás előre', 'Várakozás, amíg a színérzékelő 50%-nál nagyobb a felületről visszavert fényértéket érzékel.', 'Motorok/mozgás leállítása', 'Program leállítás és kilépés a programból'.

A színérzékelő szenzorról több a [01_spike-v3_block_guide_hu.pdf](#) című segédlet

Szenzor (érzékelő) blokkok – BASIC részében található!

A példa alapján – gyakorlással és teszteléssel – minden színszenzort használó mozgásos feladat elvégezhető! 😊



Távolságérzékelő szenzor használata

Robot mozgatása adott távolságban érzékelt akadályig.

Távolságérzékelő szenzor használata esetén a  **wait until..** blokk paramétereként használhatjuk az **is ..**  blokkot, amely igaz értéket ad vissza, ha a távolságérzékelő a beállított távolságnál **kevesebbet**, **többet** vagy **pontosan ugyanakkora** értéket érzékel mint %-ban, centiméterben vagy inch-ben megadott érték.





The image shows a Scratch script on the left and a console window on the right. The script consists of the following blocks: 'when program starts', 'set movement motors to C+D', 'set movement speed to 20 %', 'start moving' (up arrow), 'wait until' (with 'E' selected, 'is exactly at', '15', 'cm', and a question mark), 'stop moving', and 'stop and exit program'. The console window displays the following text: 'A robot előre haladva az akadály előtt 15 cm-rel megáll!', 'Blokkok:', 'Mozgató motorok beállítása C+D', 'Motor sebesség beállítás 20%-ra.', 'Folyamatos mozgás előre', 'Várakozás, amíg a távolságérzékelő szenzor 15 cm-es távolságban akadályt érzékel.', 'Motorok/mozgás leállítása', and 'Program leállítás és kilépés a programból'.

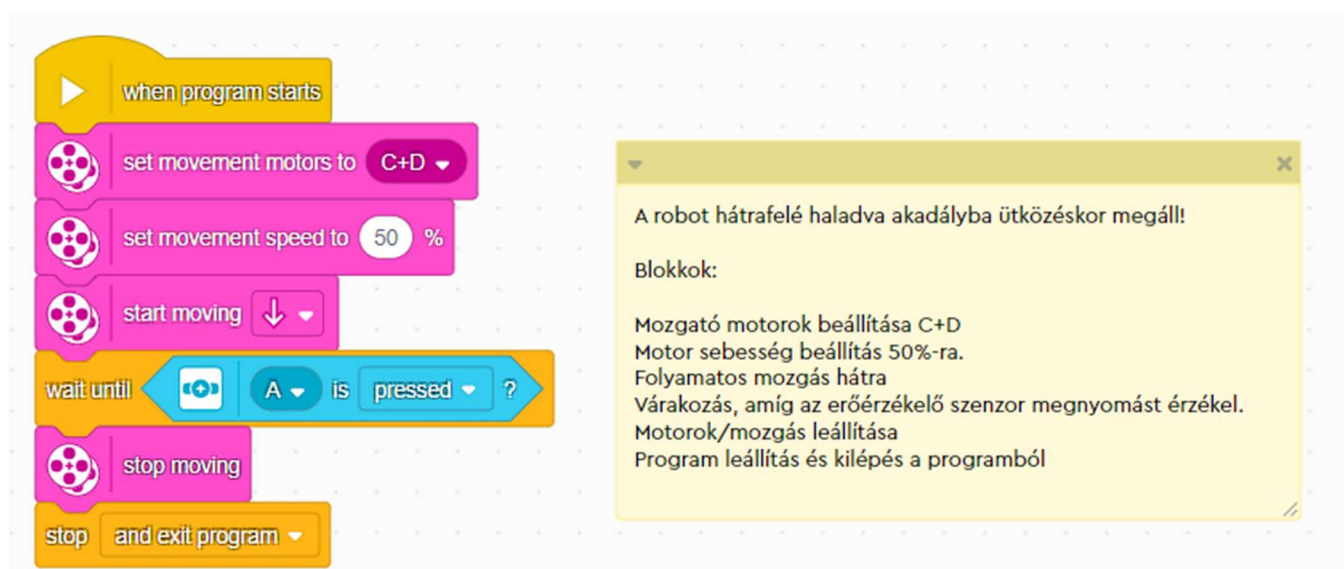
A távolságérzékelő szenzorról több a [01_spike-v3_block_guide_hu.pdf](#) című segédlet **Szenzor (érzékelő) blokkok – BASIC** részében található!

A példa alapján – gyakorlással és teszteléssel – minden távolság érzékeléshez kapcsolódó mozgásos feladat elvégezhető! 😊

Nyomásérzékelő használata

Robot mozgatása az erőszenzor megnyomásáig.

Erő (nyomás) érzékelő szenzor használata esetén a  **wait until..** blokk paramétereként használhatjuk az **is ..**  blokkot, amely igaz értéket ad vissza, ha az erő érzékelő a beállított nyomást érzékeli.



The image shows a Scratch script on the left and a text box on the right. The script consists of the following blocks: 'when program starts', 'set movement motors to C+D', 'set movement speed to 50 %', 'start moving' (with a downward arrow), 'wait until' (with 'A is pressed' selected), 'stop moving', and 'stop and exit program'. The text box on the right contains the following text: 'A robot hátrafelé haladva akadályba ütközéskor megáll!', 'Blokkok:', 'Mozgató motorok beállítása C+D', 'Motor sebesség beállítás 50%-ra.', 'Folyamatos mozgás hátra', 'Várakozás, amíg az erőérzékelő szenzor megnyomást érzékel.', 'Motorok/mozgás leállítása', and 'Program leállítás és kilépés a programból'.

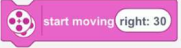

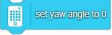
Az erő (nyomás) érzékelő szenzorról több a [01_spike-v3_block_guide_hu.pdf](#) című segédlet **Szenzor (érzékelő) blokkok – BASIC** részében található!





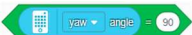

A példa alapján – gyakorlással és teszteléssel – minden nyomásérzékeléshez kapcsolódó mozgásos feladat elvégezhető! 😊

Gyro szenzor használata

Robot forgatása gyro (elfordulás) szenzor segítségével.

Pontosabb robot fordulás programozható elfordulás szenzor (érzékelő) használatával.

A megvalósítás elmélete röviden: a robotot valamelyik irányba adott fordulási móddal (egyik kerék áll, a másik forog vagy a két kerék ellentétes irányba forog) folyamatosan forgatjuk. Ehhez a **start moving**  blokkot használjuk. Mivel a forgás (mozgás) folyamatos, gondoskodni kell a forgás megállításáról. Ehhez a már megismert **wait until..**  blokkot használjuk, amelynek paramétereként az elfordulás szenzor által mért értéket hasonlítjuk össze egy általunk megadott értékkel, azaz várunk arra, hogy a fordulás mértéke elérje az általunk beállított értéket. Amint ez megtörtént leállítjuk/kikapcsoljuk a motorokat. (Javaslom a forgatás előtt a gyro szenzort „resetelni” azaz **yaw** értéket (az irányt) nullára állítani a **set yaw angle to 0**  blokkal!)

Mint tudjuk, a **wait until..**  blokk igaz értéke esetén lép a program a következő blokkra, esetünkben a forgás megállítására, azért a **wait until..**  paraméterét úgy kell meghatározni, hogy az általunk kívánt fordulási mértékét elérve igaz legyen. Ehhez szükségünk van a gyro szenzor által visszaadott értékre illetve egy olyan blokkra, amelyben a szenzor által mért fordulás értéket össze tudjuk hasonlítani az általunk kívánt értékkel. A gyro szenzor által mért aktuális elfordulás értékét a  **angle** blokk tartalmazza, míg az összehasonlításhoz az **operators** blokkcsoportból egy egyenlőséget vizsgáló blokkot  használunk. Ennek a blokknak a bal oldalára kerül az **angle** blokk, a jobb oldalára az általunk kívánt elfordulás fokban megadott értéke. Például így:  Ebben az esetben ez a blokk akkor ad igaz értéket, ha robot elfordulás **yaw** (vízszintes) irányba **90 fokos** lesz. Ezt a blokkot felhasználva a **wait until..**  blokk paramétereként tudjuk a robot fordulását tetszőleges mértékű elfordulásra programozni.

Az elfordulás fok értéke lehet negatív vagy pozitív a fordulás irányától és a motor(ok) forgási irányától függően! **A robot (hub) óramutatóval megegyező irányú fordulása a gyro szenzor pozitív értéktartománya, míg az óramutatóval ellentétes fordulása a negatív értéktartománya.**

FONTOS! Minél kisebb sebességgel forgatjuk a robotot, annál pontosabban lesz az érzékelés, annál közelebb áll meg az elfordulás az általunk beállított/kívánt értékhez! Teszteléssel kell meghatározni azt a sebesség értéket, ami a megfelel az általunk támasztott elvárásoknak! (gyorsaság vagy pontosság)

A gyro (elfordulás, gyorsulás) szenzorról több a **01_spike-v3_block_guide_hu.pdf** című segédlet **Szenzor (érzékelő) blokkok – BASIC** részében található!



A fordulás, kanyarodás megvalósításához a **Movement** csoportban található blokkokkal valósítható meg. Részletes leírás a **Movement** csoportban található blokkokról a **02_spike-v3_basic_move_hu.pdf** és a **01_spike-v3_block_guide_hu.pdf** dokumentumokban található. Itt csak az elforduláshoz, kanyarodáshoz szükséges blokkokat mutatom be részletesen.

	<p>Mozgató motorok beállítása.</p> <p>Itt adjuk meg, hogy melyik két csatlakozóra (portra) vannak kötve a robotot mozgató motorok</p>
	<p>A beállított mozgató motorok sebességét állítja be a maximális érték százalékára. (100 és -100 közötti érték)</p> <p>Negatív érték ellentétes motormozgást jelent!</p>
	<p>A beállított motorok vezérlésével forgatja a robotot jobbra vagy balra az iránynál megadott számhoz tartozó fordulási mód szerint.</p> <p>Ennél a blokknál a forgást megállításáról gondoskodni kell!</p>
	<p>A beállított motorok vezérlését lekapcsolja, a motorok megállnak!</p>

Fordulás, kanyarodás



A robot mozgatását a fixen beépített motorok biztosítják. Ennél a megoldásnál nincs kormányzott kerék, így a robot fordulásának, kanyarodásának megoldása a lánctalpas járművekével (pl. tank) azonos módon történik. Szenzorokkal vezérelt fordulás/kanyarodás megvalósításhoz a **start moving** blokkot használjuk.

Ahogy a **02_spike-v3_basic_move_hu.pdf** dokumentumban részletesen olvasható, az fordulás/kanyarodás megvalósítására technikailag három megoldás létezik:

1. Az egyik kerék (motor) áll, a másik forog. Ebben az esetben a fordulás az álló kerék körül történik.
2. A két kerék (motor) ellentétes irányba forog. Ebben az esetben a fordulás helyben, a két kereket összekötő képzeletbeli vonal középpontja körül történik.
3. Az egyik kerék (motor) lassabban forog, mint a másik. Ebben az esetben a robot ívesen fordul, kanyarodik a lassabban forgó kerék körül.

Ebben a dokumentumban az első két megoldás minden lehetséges beállítására, a harmadik, íves fordulásra előre, a **move ... to ..** blokk, **move** érték **right 20** és **left -20** beállításra található példa.

Fordulás egyik kerék körül – az egyik kerék (motor) áll, a másik forog.

A **start moving**  blokknál a **move** értéke **right 50** vagy **left -50**. Ez a beállítás jelenti azt, hogy az egyik kerék áll, a másik forog. **Set movement speed to:**  blokknál ha a % értéke pozitív, akkor a mozgó kerék előre forog, ha negatív, akkor a mozgó kerék hátra forog

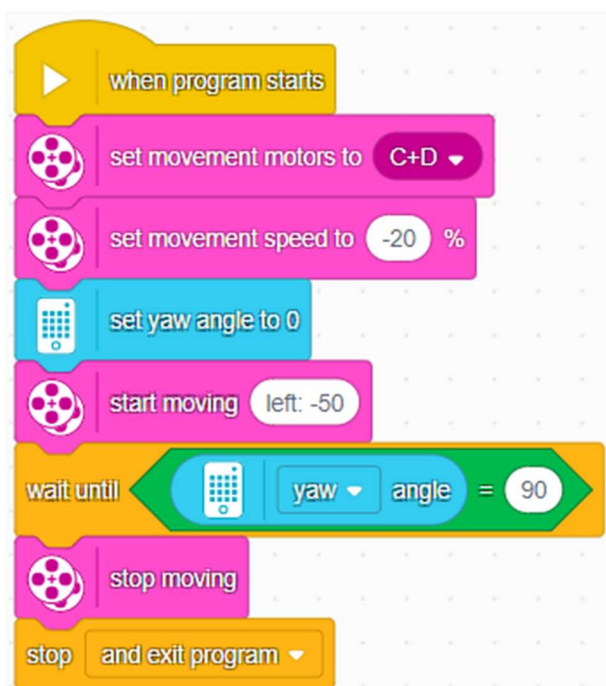
Jobbra előre fordulás 90 fokig:



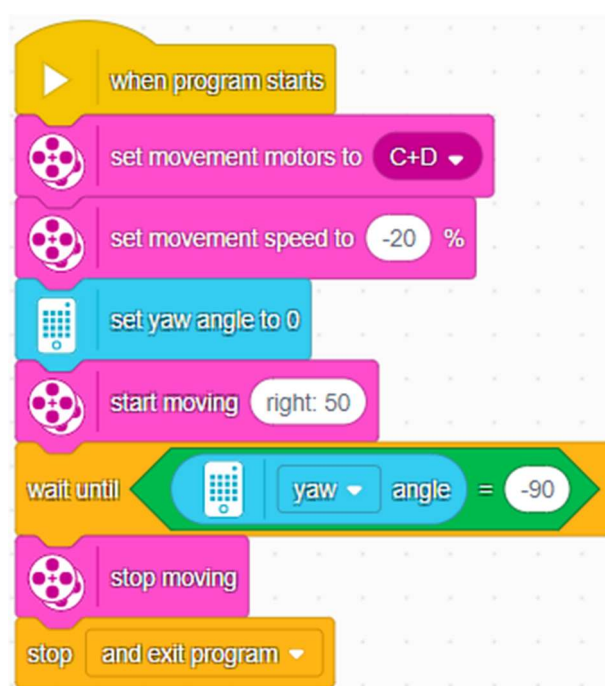
Balra előre fordulás 90 fokig:




Jobbra hátra fordulás 90 fokig:



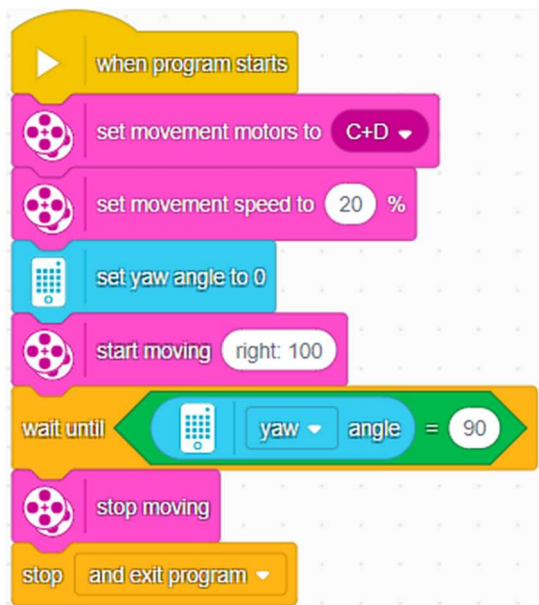
Balra hátra fordulás 90 fokig:



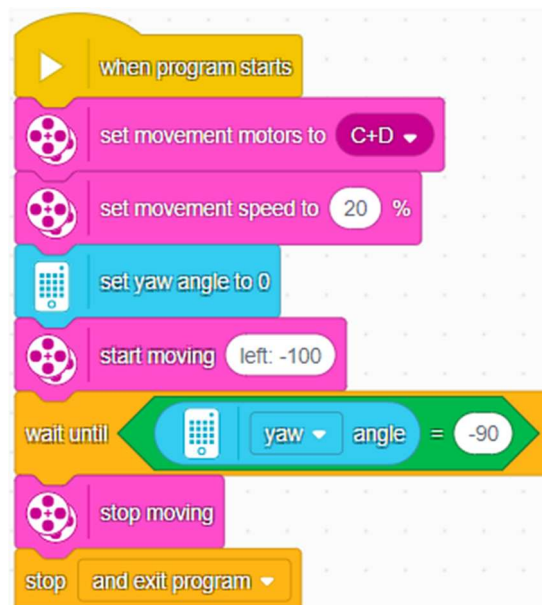
Fordulás helyben – az egyik kerék (motor) „előre”, a másik „hátra” forog.

A **start moving**  blokk értéke **right 100** vagy **left -100**. Ez a beállítás jelenti azt, hogy a két kerék ellentétes irányba forog. **Set movement speed to:**  blokknál ha a % értéke pozitív, akkor a kerekek előre forognak, ha negatív, akkor hátra.

Jobbra előre fordulás 90 fokig:



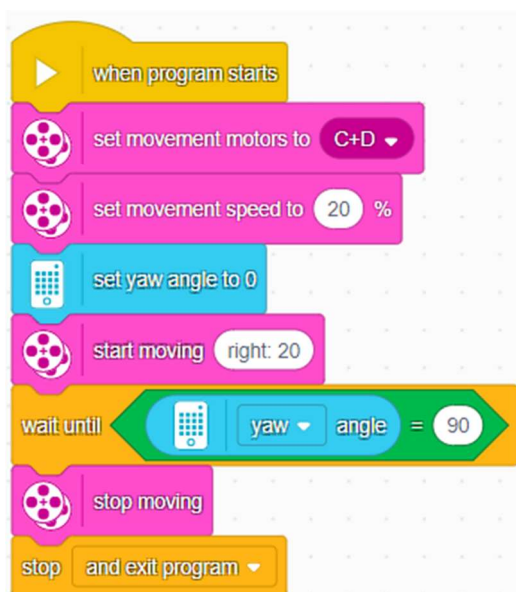
Balra előre fordulás 90 fokig:



Íves kanyarodás – az egyik kerék (motor) lassabban forog, mint a másik.

Ha a robotot ívesen/kanyarodva akarjuk mozgatni, akkor a **move ... for..** blokknál a **move** értékét 1 és 49 vagy 51 és 99 közötti tartományból kell kiválasztanunk. Az alábbi két példában **start moving right 20** és **left -20** beállítással előre (**set movement speed to ..** pozitív érték) kanyarodásra látható példa.

Jobbra előre kanyarodás 90 fokig:



Balra elő

