

Workshop di Robotica – IPSIA G. Ceconi

Benvenuti!

22 maggio 2021

Robot

# Robot

Dal ceco **robota**, lavoro pesante.

# Robot

Dal ceco **robota**, lavoro pesante.

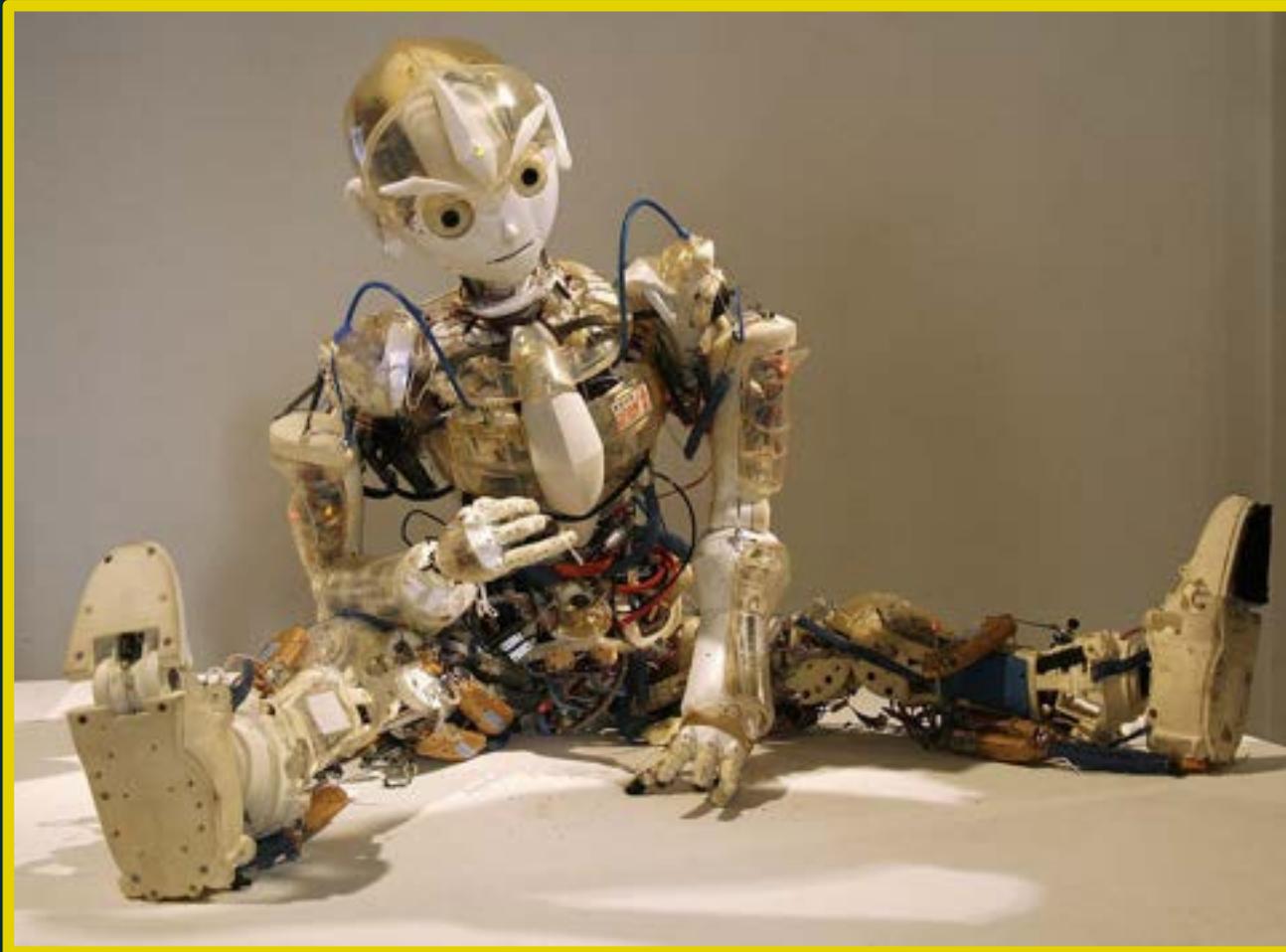
Usato dallo scrittore ceco Karel Čapek in un dramma teatrale del 1920 ad indicare degli umanoidi creati per svolgere i lavori più faticosi...

# Definizione

Apparato meccanico ed elettronico programmabile, impiegato nell'industria, in sostituzione dell'uomo, per eseguire automaticamente e autonomamente lavorazioni e operazioni ripetitive, o complesse, pesanti e pericolose.

Esempi

# Esempi

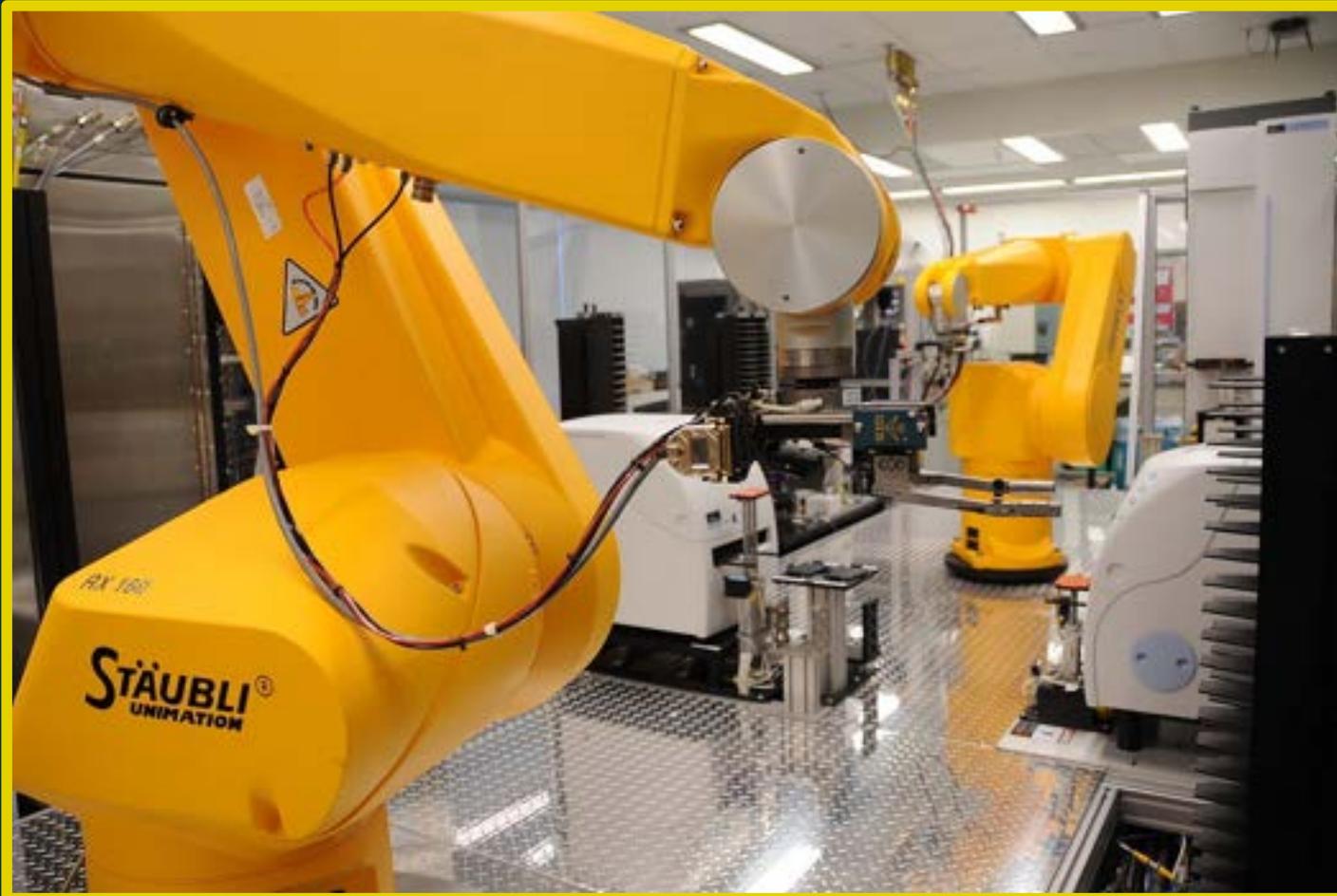


By Manfred Werner - Tsui - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=4762533>

# Esempi



# Esempi



By Maggie Bartlett, National Human Genome Research Institute - <http://www.genome.gov/dmd/img.cfm?node=Photos/Technology/Research%20laboratory&id=79299>,  
Public Domain, <https://commons.wikimedia.org/w/index.php?curid=37410189>

# Esempi



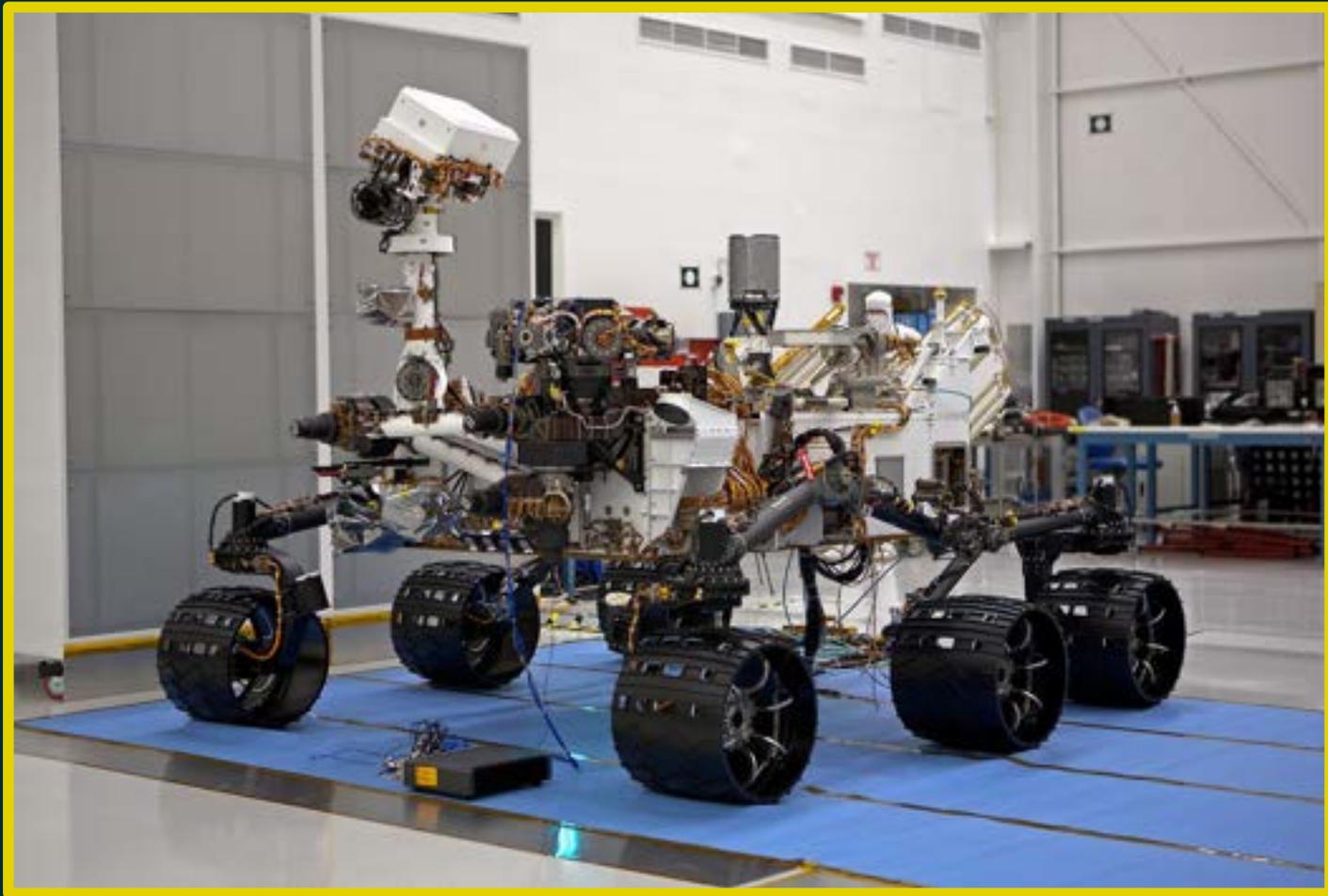
By Robotics - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=56206814>

# Esempi



By [Franciscan Health](https://www.franiscanhealth.org/health-care-services/robotic-assisted-surgery-334) - <https://www.franiscanhealth.org/health-care-services/robotic-assisted-surgery-334>, CC BY-SA 4.0,  
<https://commons.wikimedia.org/w/index.php?curid=70874369>

# Esempi

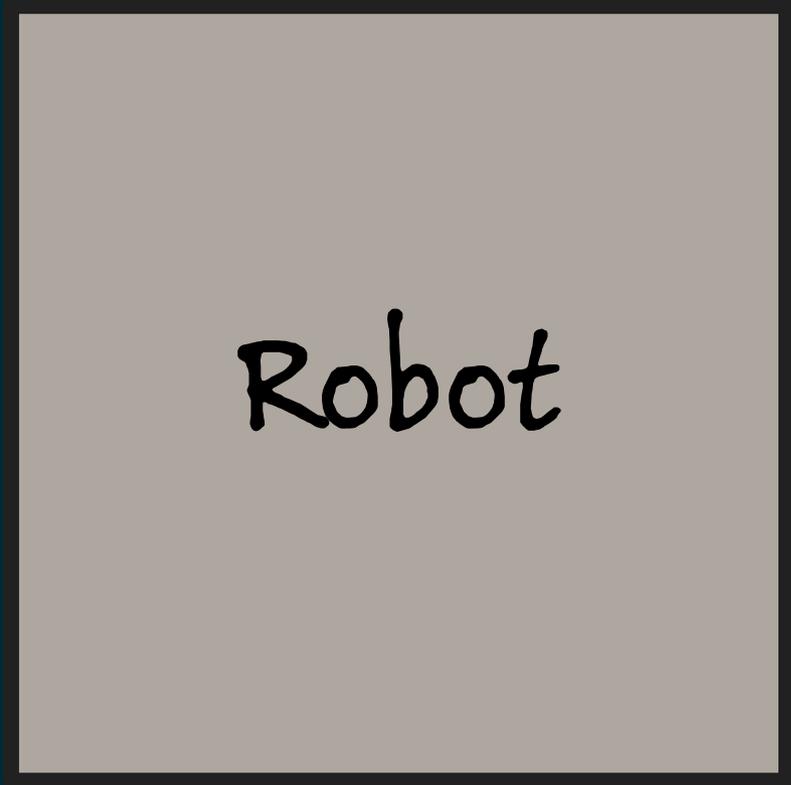


By NASA - <http://photojournal.jpl.nasa.gov/catalog/PIA14309>, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=17465432>

# Funzionamento

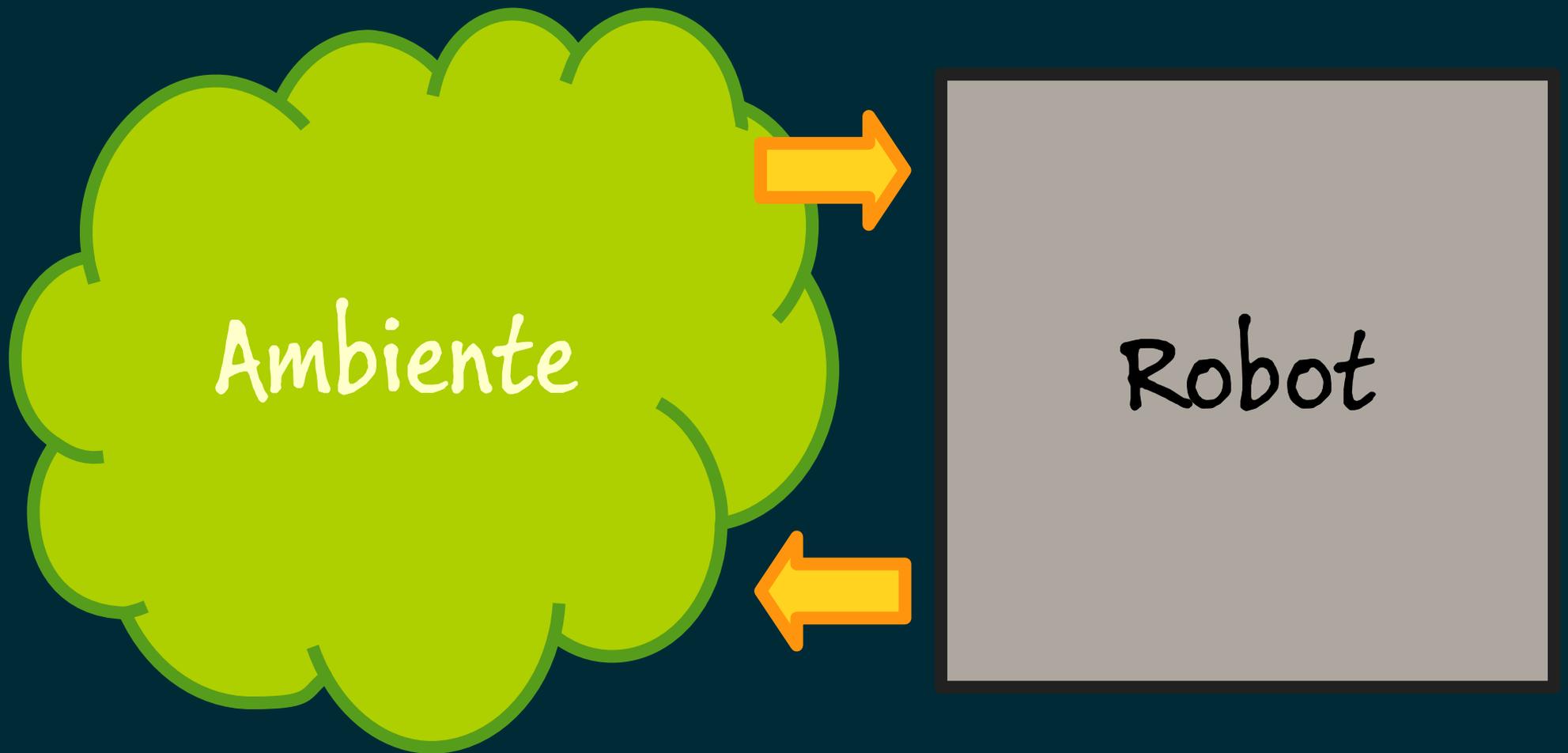


Ambiente

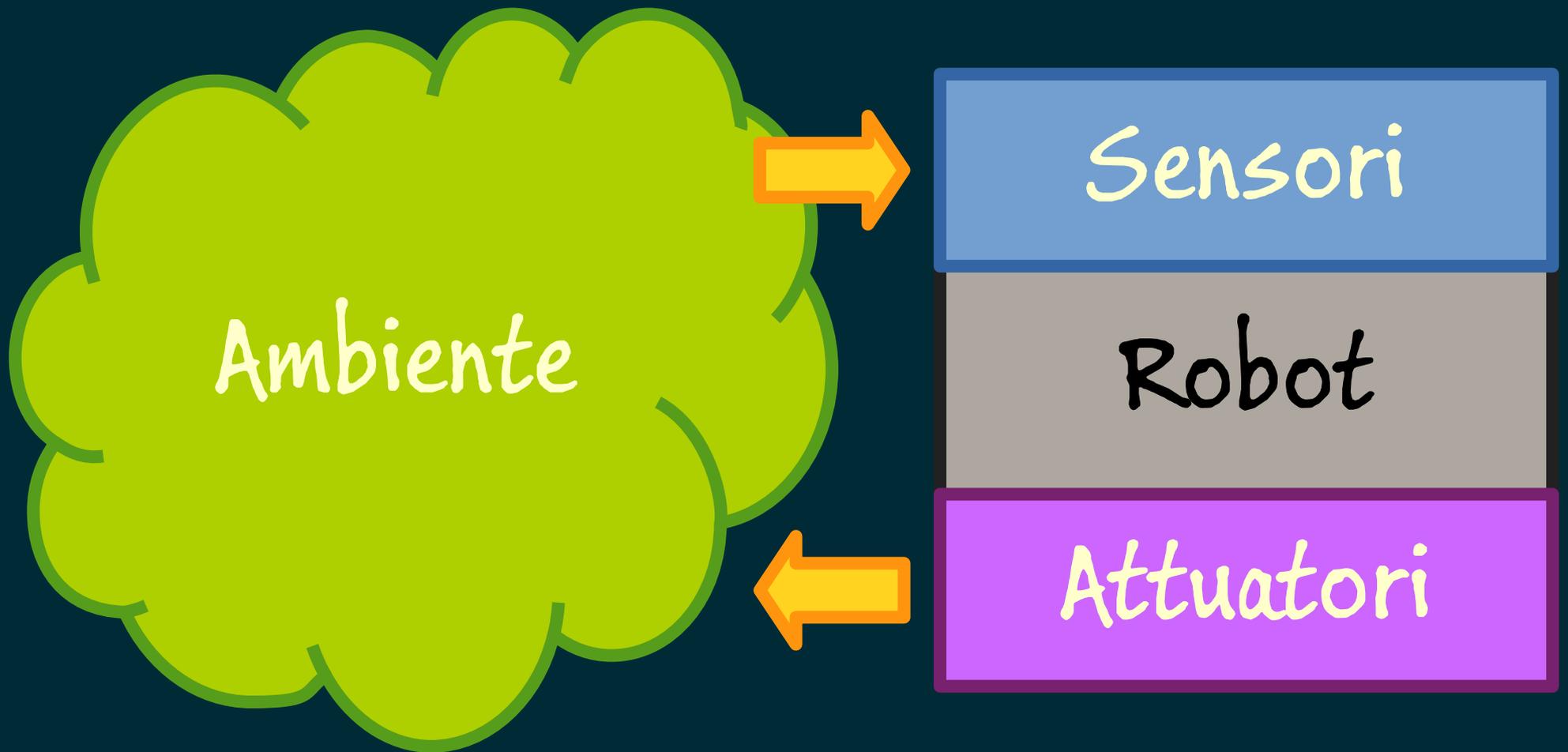


Robot

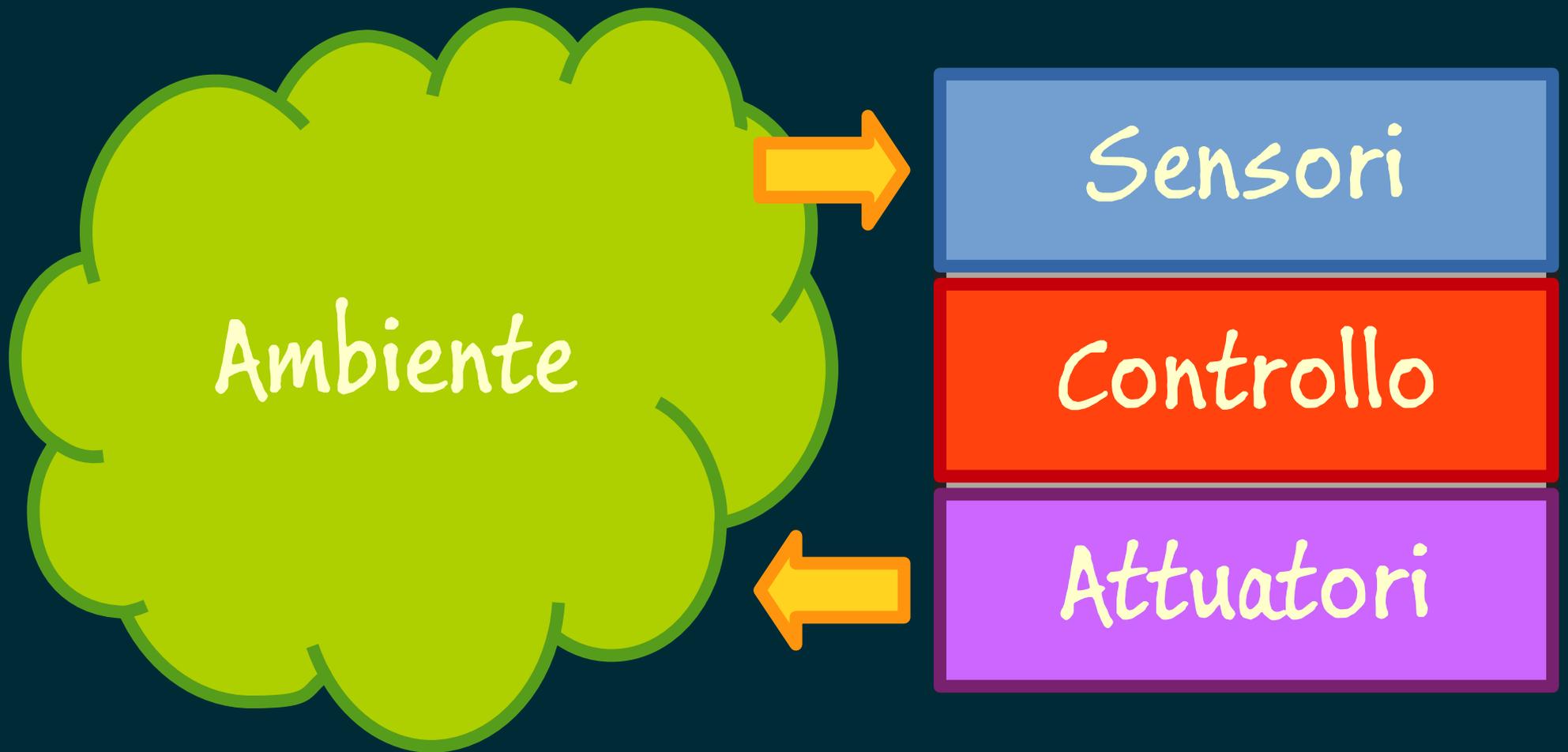
# Funzionamento



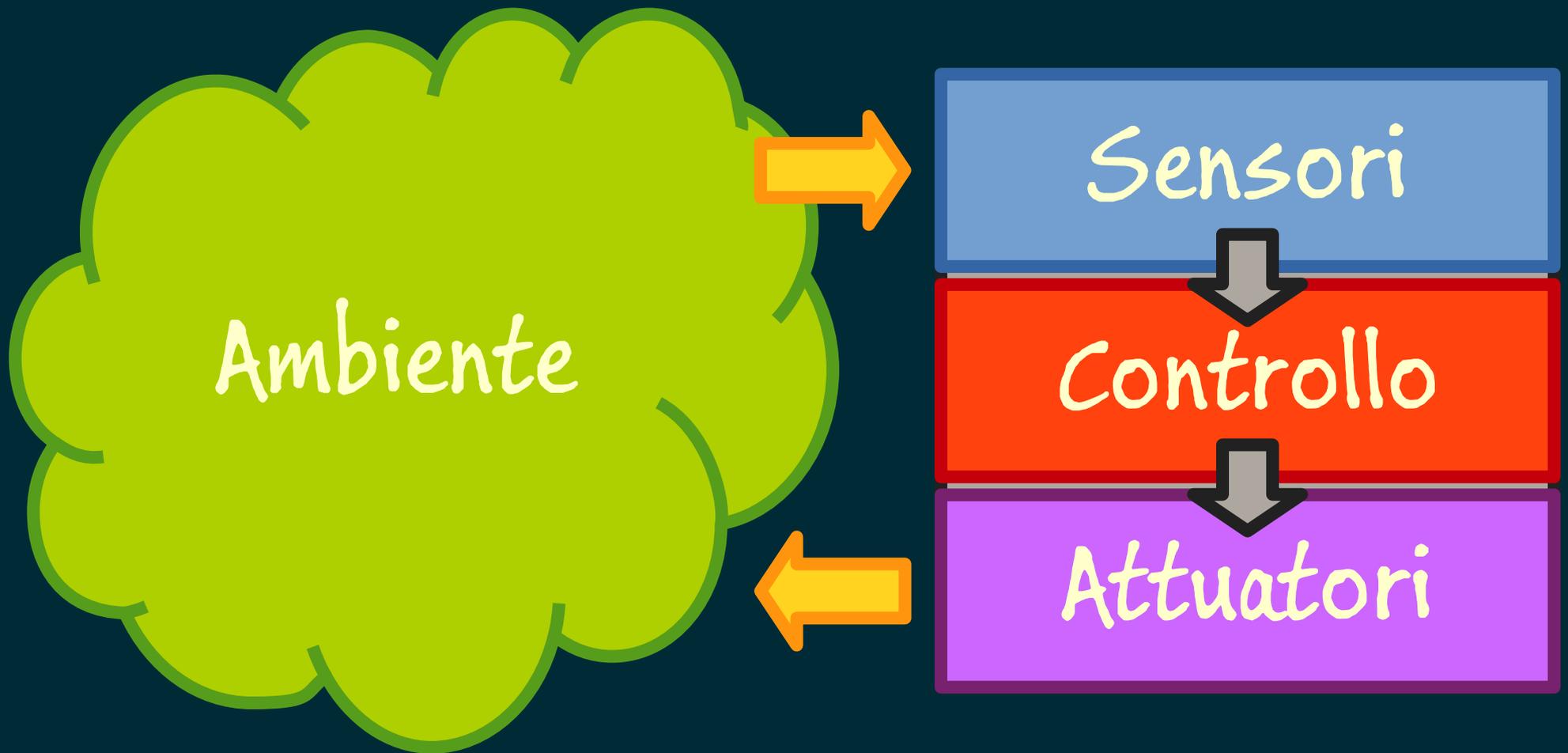
# Funzionamento



# Funzionamento



# Funzionamento



# mBot



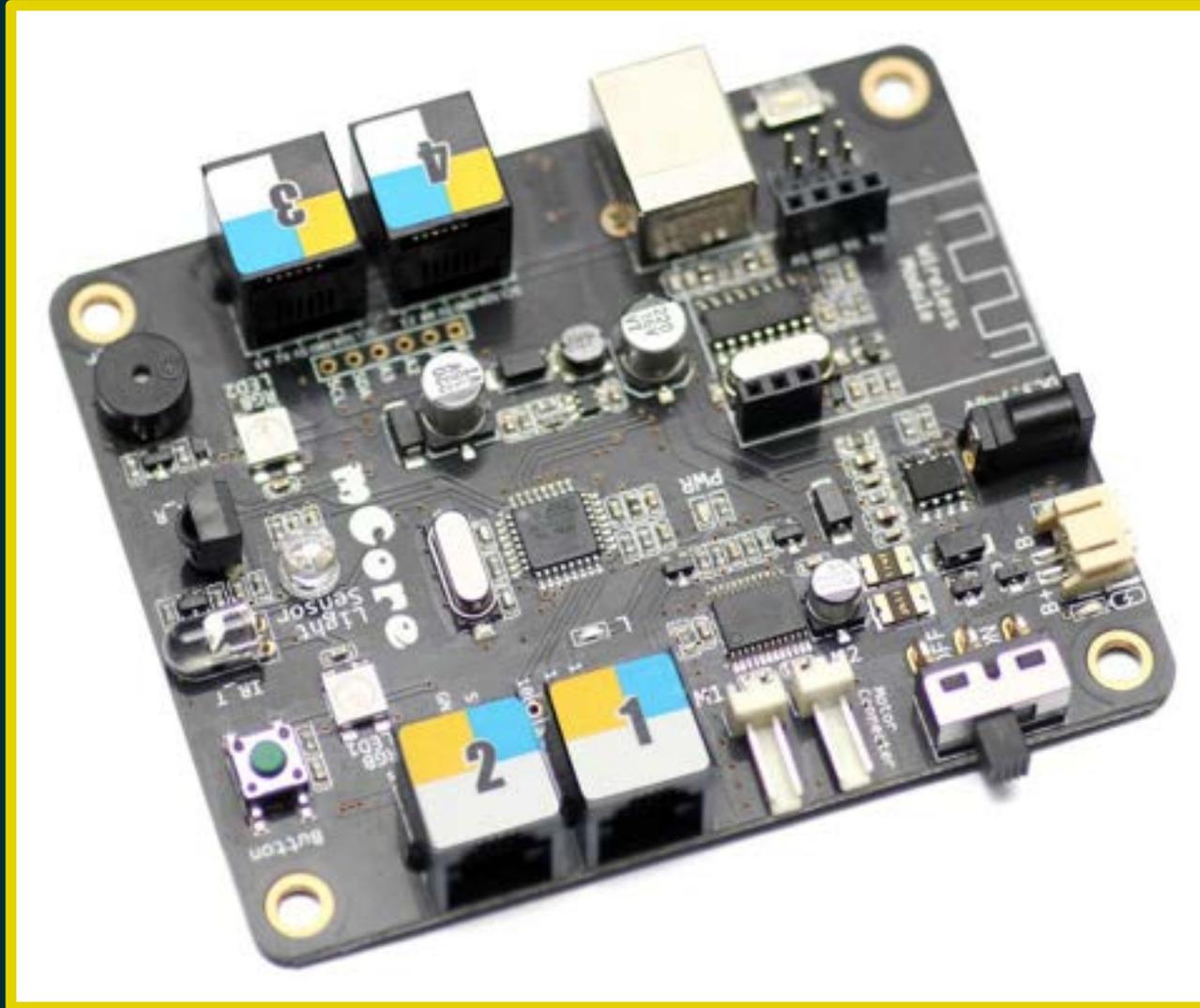
# Sensori



# Attuatori



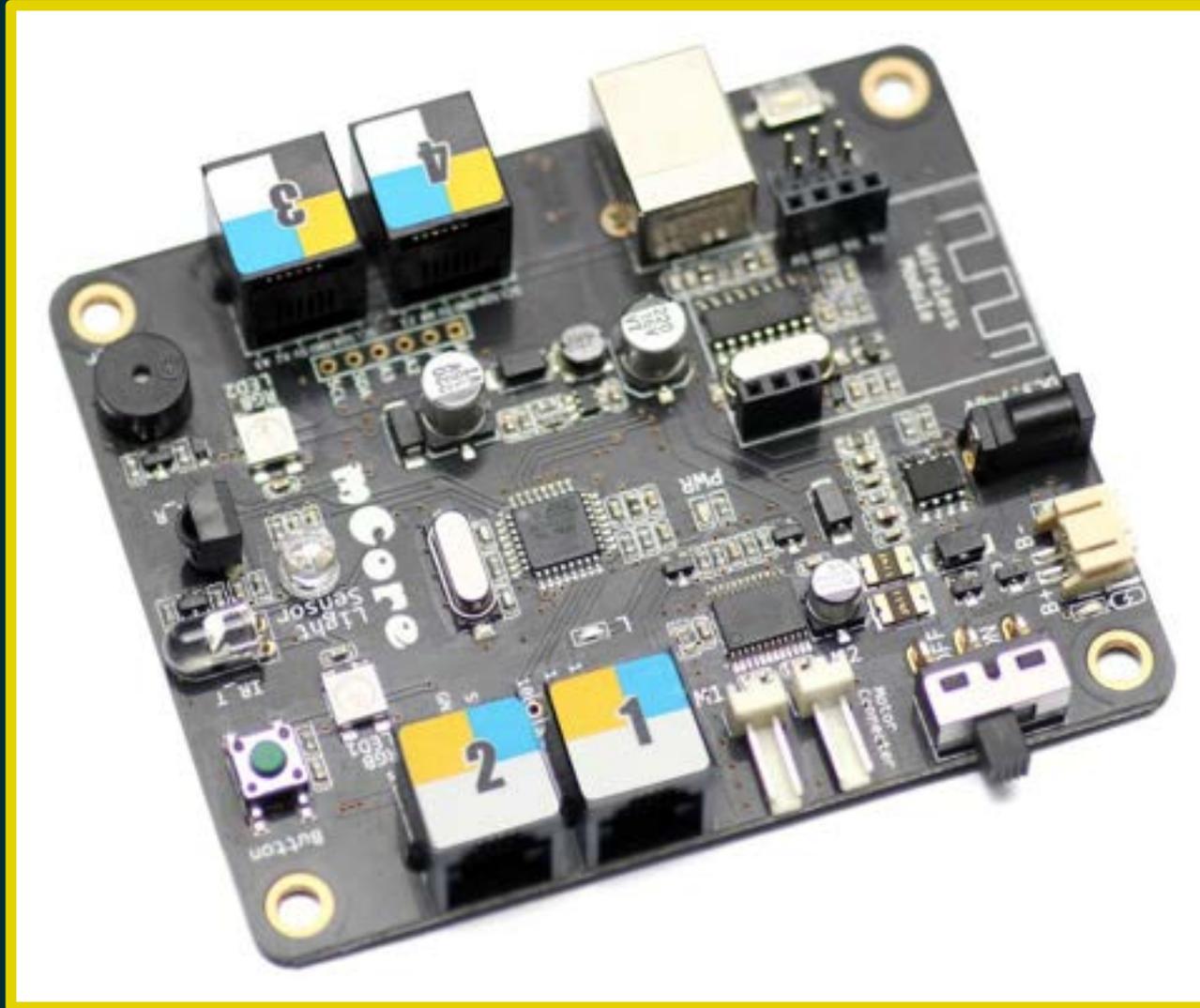
# Controllo (mCore)



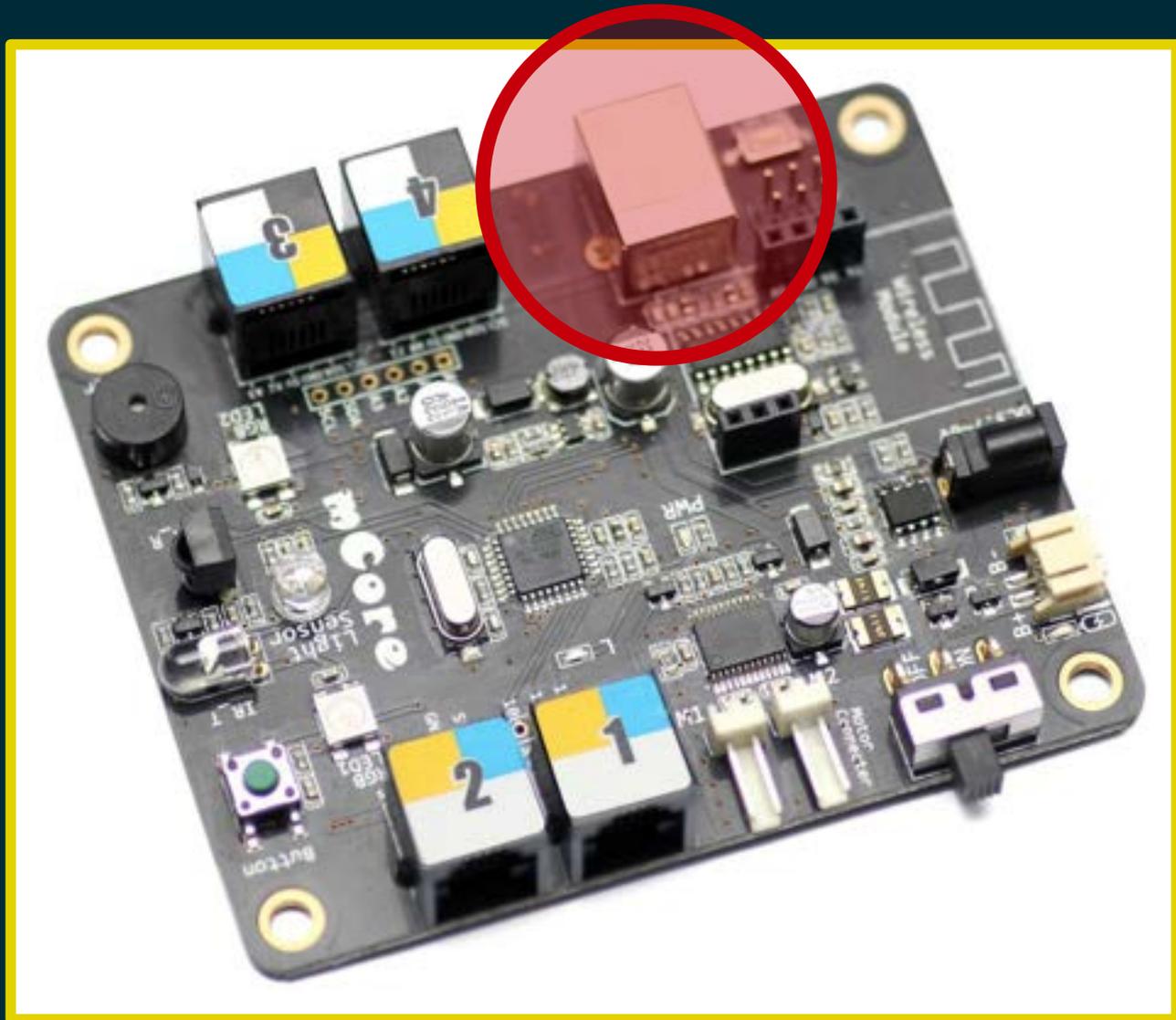
# Scheda Arduino



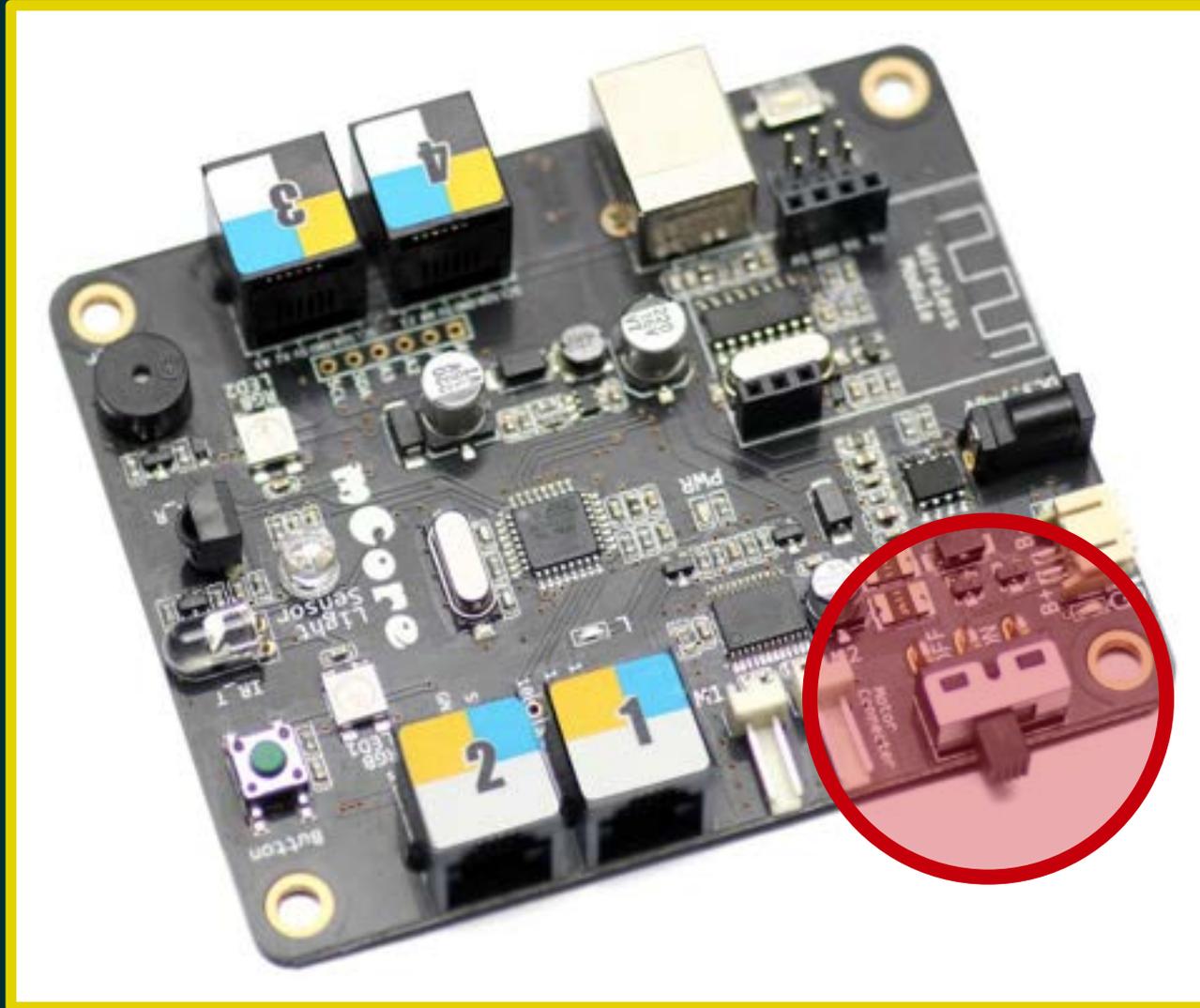
# Scheda mCore



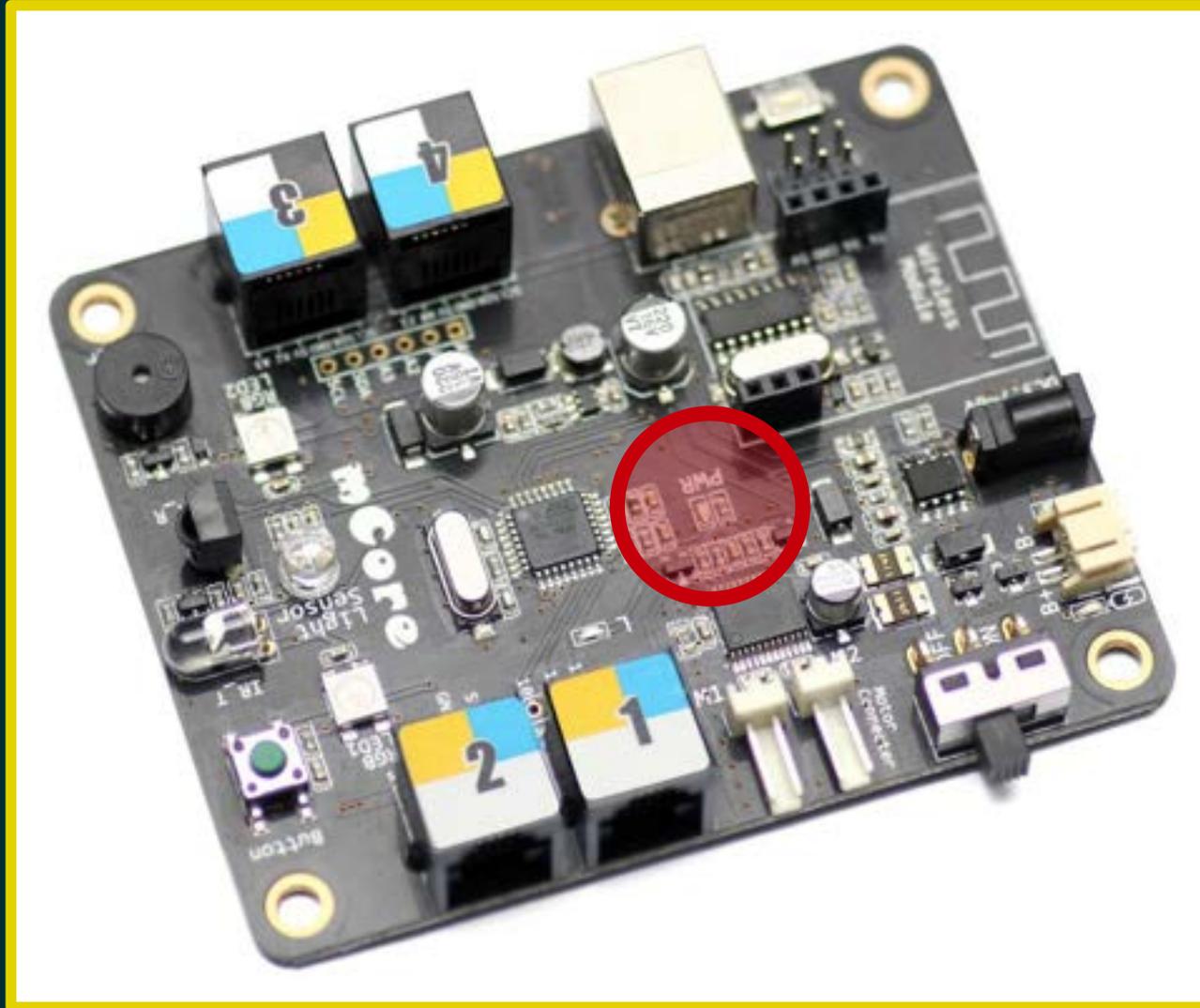
# Presă USB



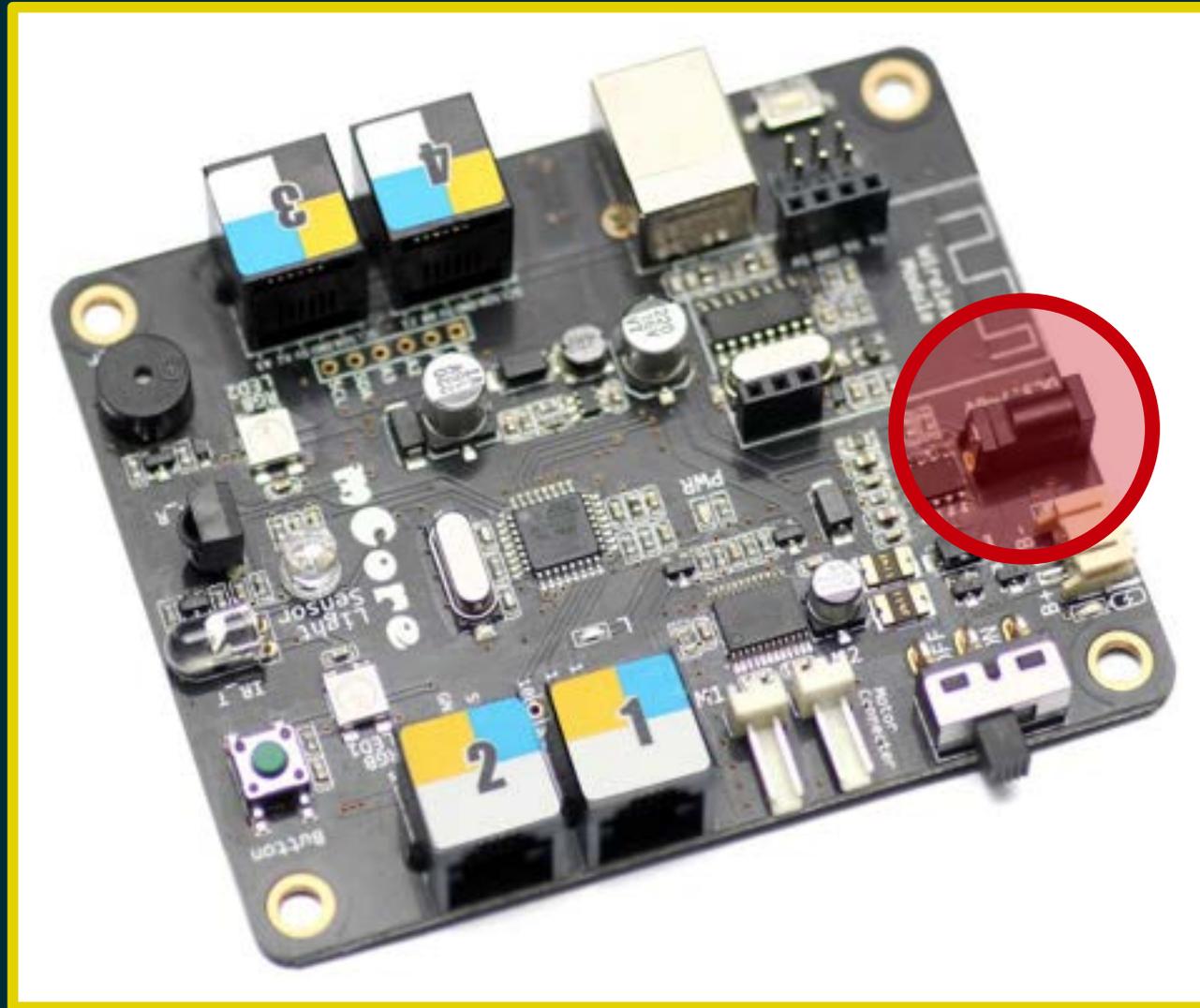
# Interruttore



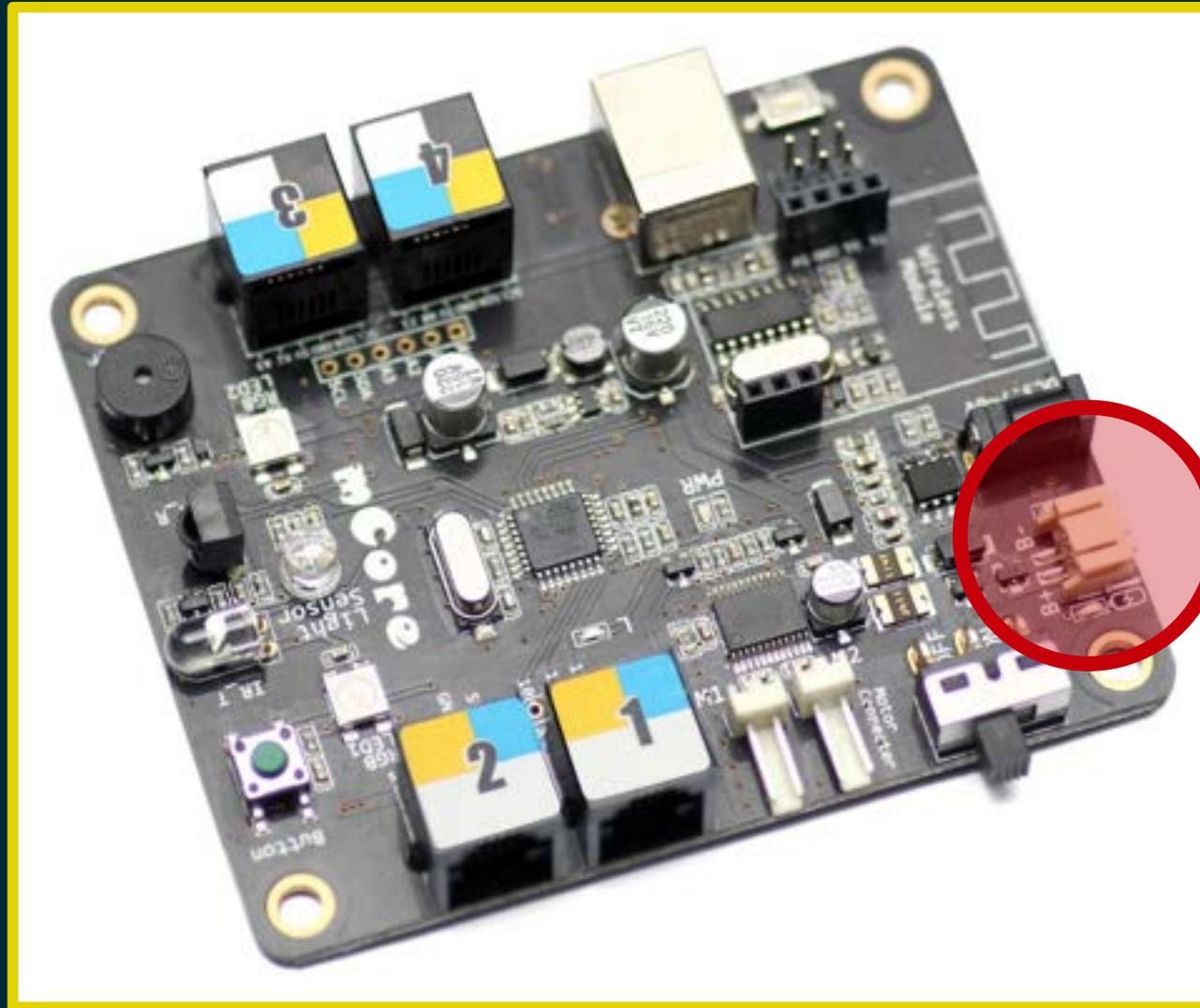
# Spia d'accensione



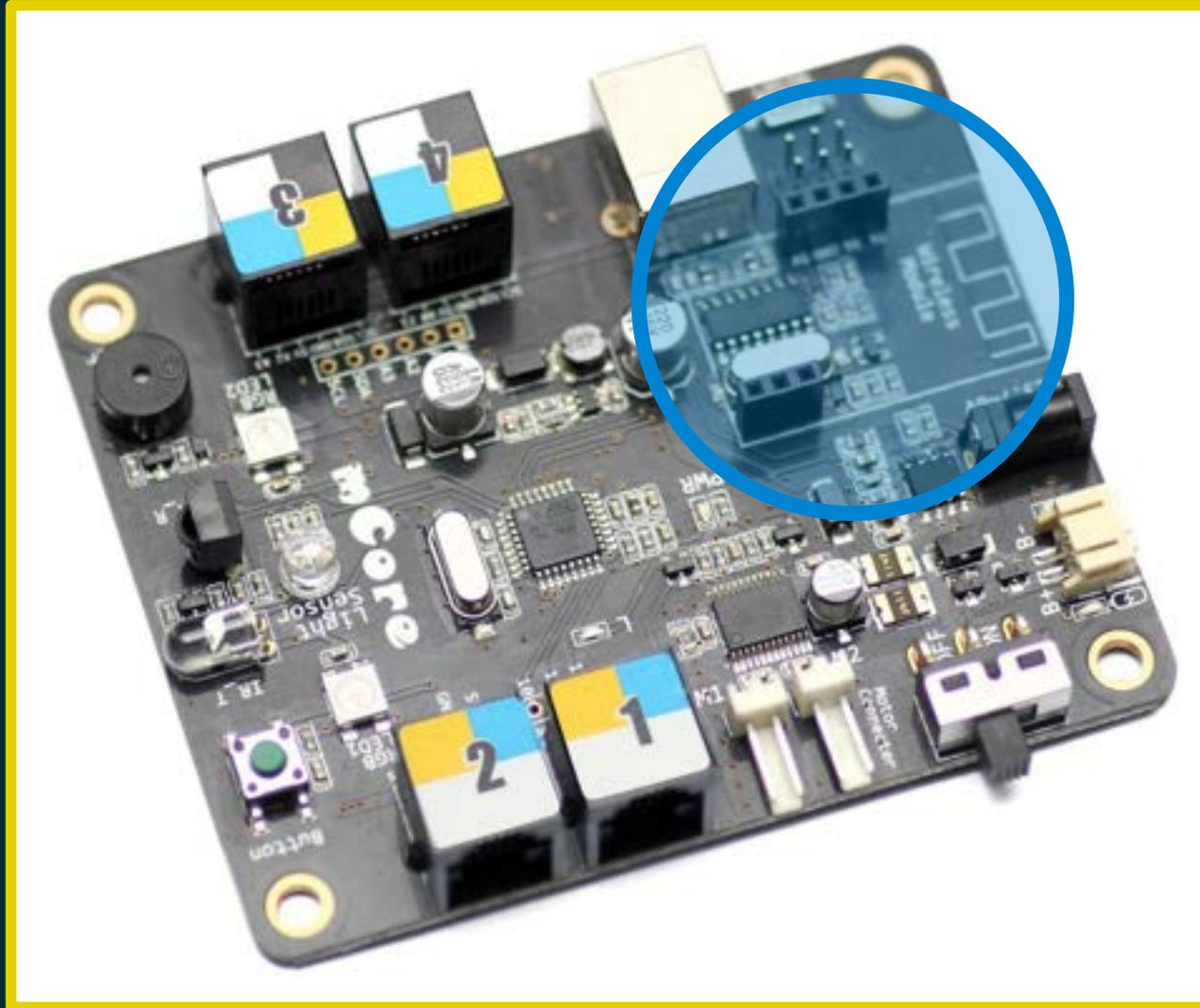
# Presse pacco batterie



# Presca batteria al litio



# Modulo Bluetooth/WiFi



Prova

# Prova

Accendo il computer...

# Prova

Accendo il computer

Collego il cavo USB al computer...

# Prova

Accendo il computer

Collego il cavo USB al computer

Collego il cavo USB a mCore...

# Prova

Accendo il computer

Collego il cavo USB al computer

Collego il cavo USB a mCore

Porto l'interruttore su ON...

# Prova

Accendo il computer

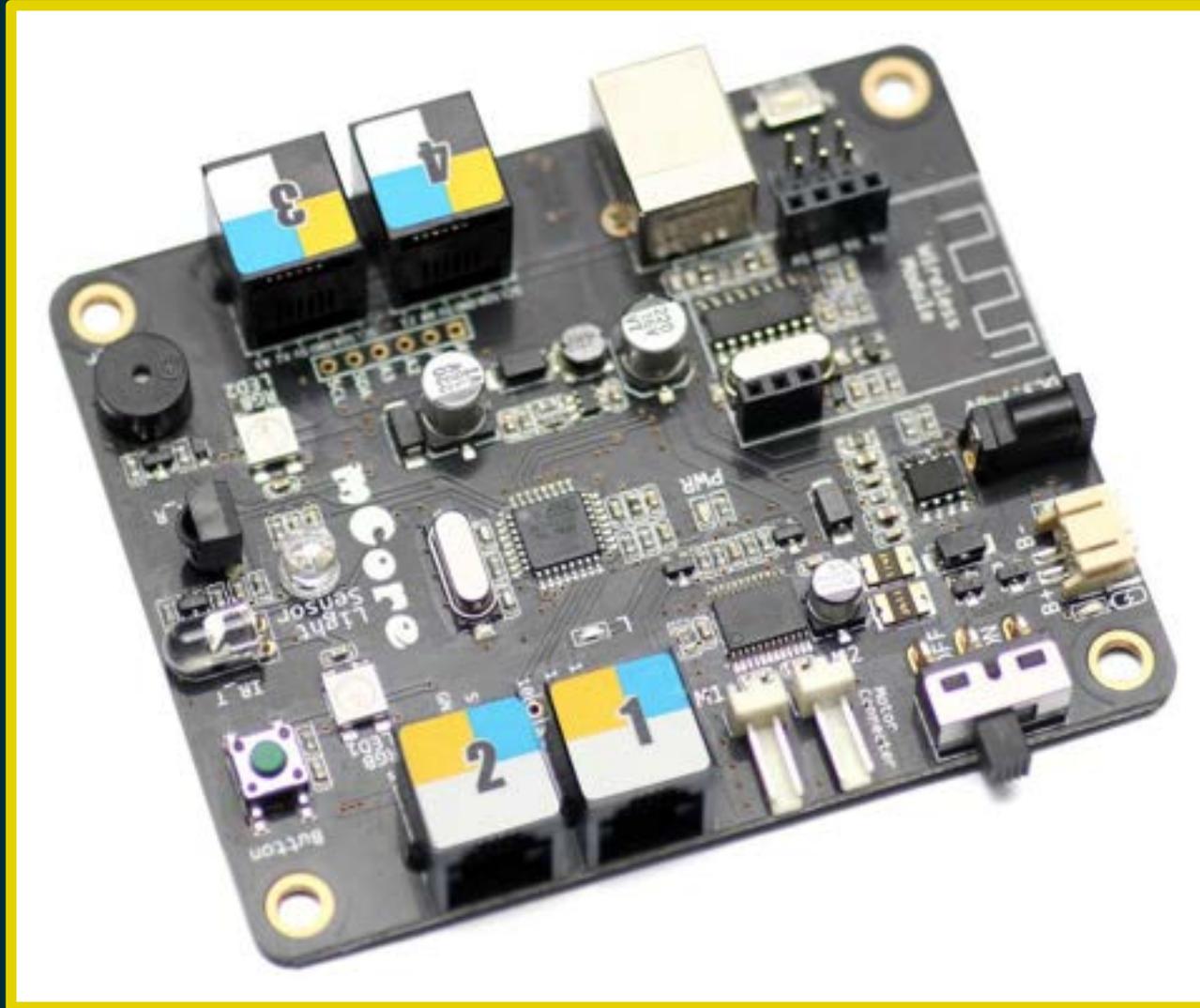
Collego il cavo USB al computer

Collego il cavo USB a mCore

Porto l'interruttore su ON

... la spia di accensione si accende!

# Programmazione



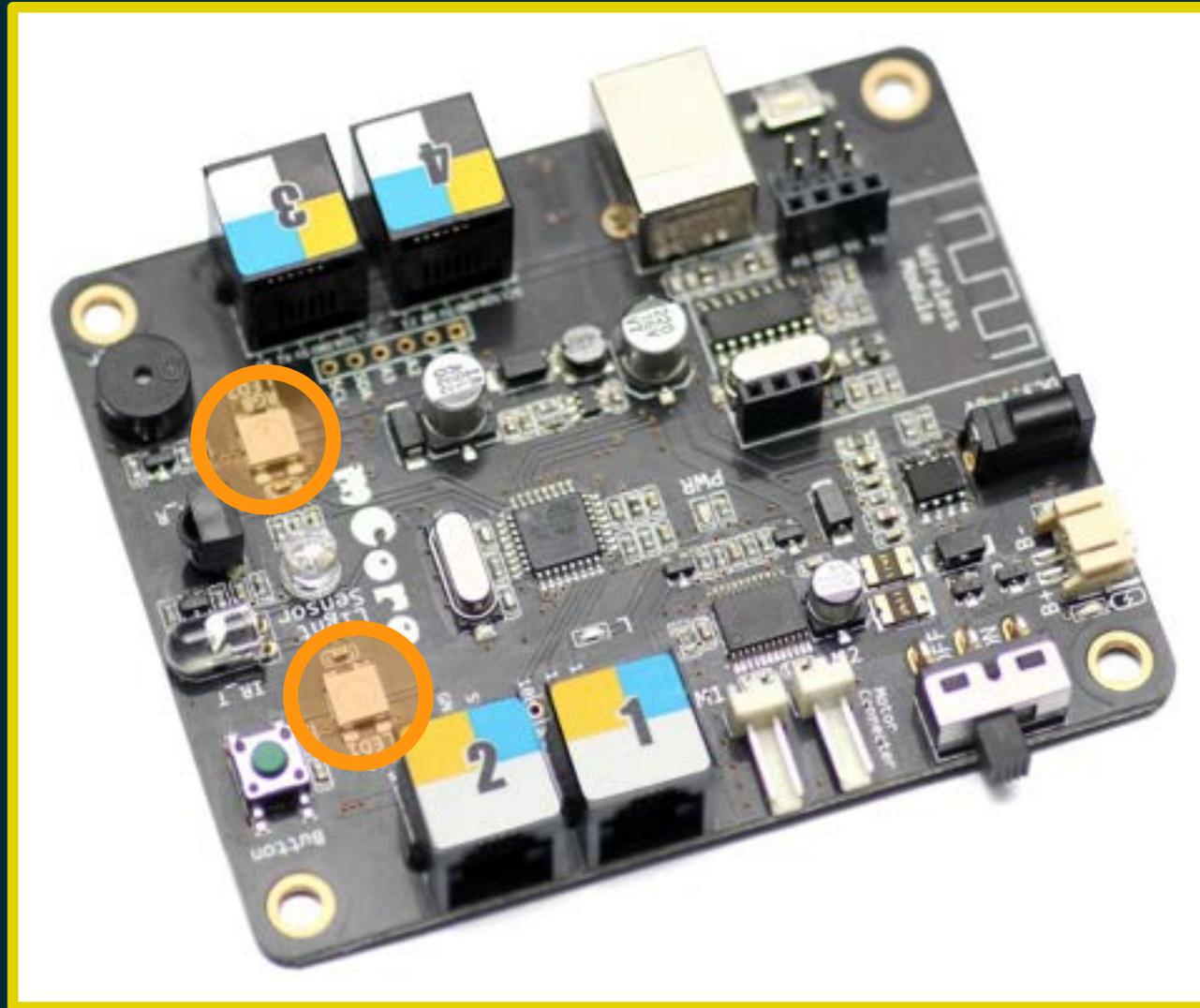
# mBlock



# Programma n. 1

Accendere i due led di rosso

# Light-Emitting Diode



# Connessione con mBlock

Collego mCore al computer..

# Connessione con mBlock

Collego mCore al computer

Avvio il programma mBlock...

# Connessione con mBlock

Collego mCore al computer

Avvio il programma mBlock

Connetto la scheda al programma..

# Connessione con mBlock

Demo!

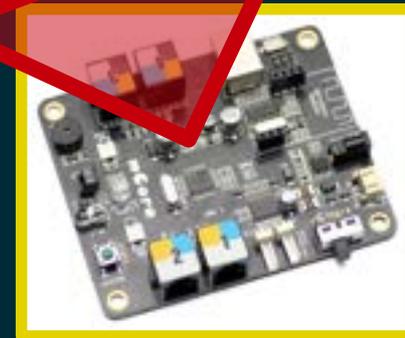
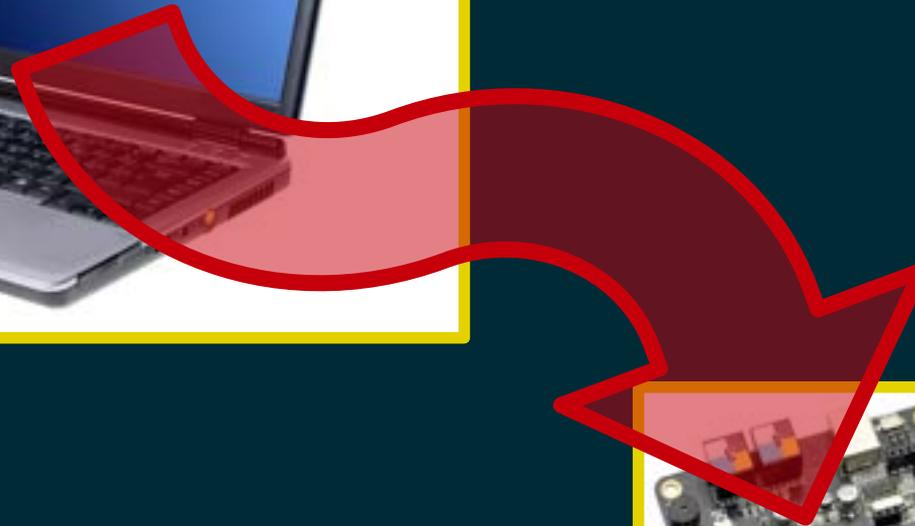
# Programma n. 1

Accendere i due led di rosso

# Programma n. 1

Demo!

# Trasferimento su mCore



# Trasferimento su mCore

Demo!

# Trasferimento su mCore

Fase I: compilazione

# Trasferimento su mCore

Fase 1: compilazione

Fase 2: trasmissione

Prova

# Prova

Spengo mCore...

# Prova

Spengo mCore

Riaccendo mCore...

# Prova

Spengo mCore

Riaccendo mCore

... i LED si accendono!

# Prova

Spengo mCore

Riaccendo mCore

... i LED si accendono!

Il programma è salvato in mCore!

# Attenzione!

Quando spengo (o scollego) mCore...

# Attenzione!

Quando spengo (o scollego) mCore...

... devo riconnettere mCore a mBlock!

# Esercitazione

Riduco l'intensità luminosa dei LED

# Esercitazione

Riduco l'intensità luminosa dei LED  
Spengo i LED dopo qualche secondo

# Esercitazione

Riduco l'intensità luminosa dei LED  
Spengo i LED dopo qualche secondo  
Faccio lampeggiare i LED

# Esercitazione

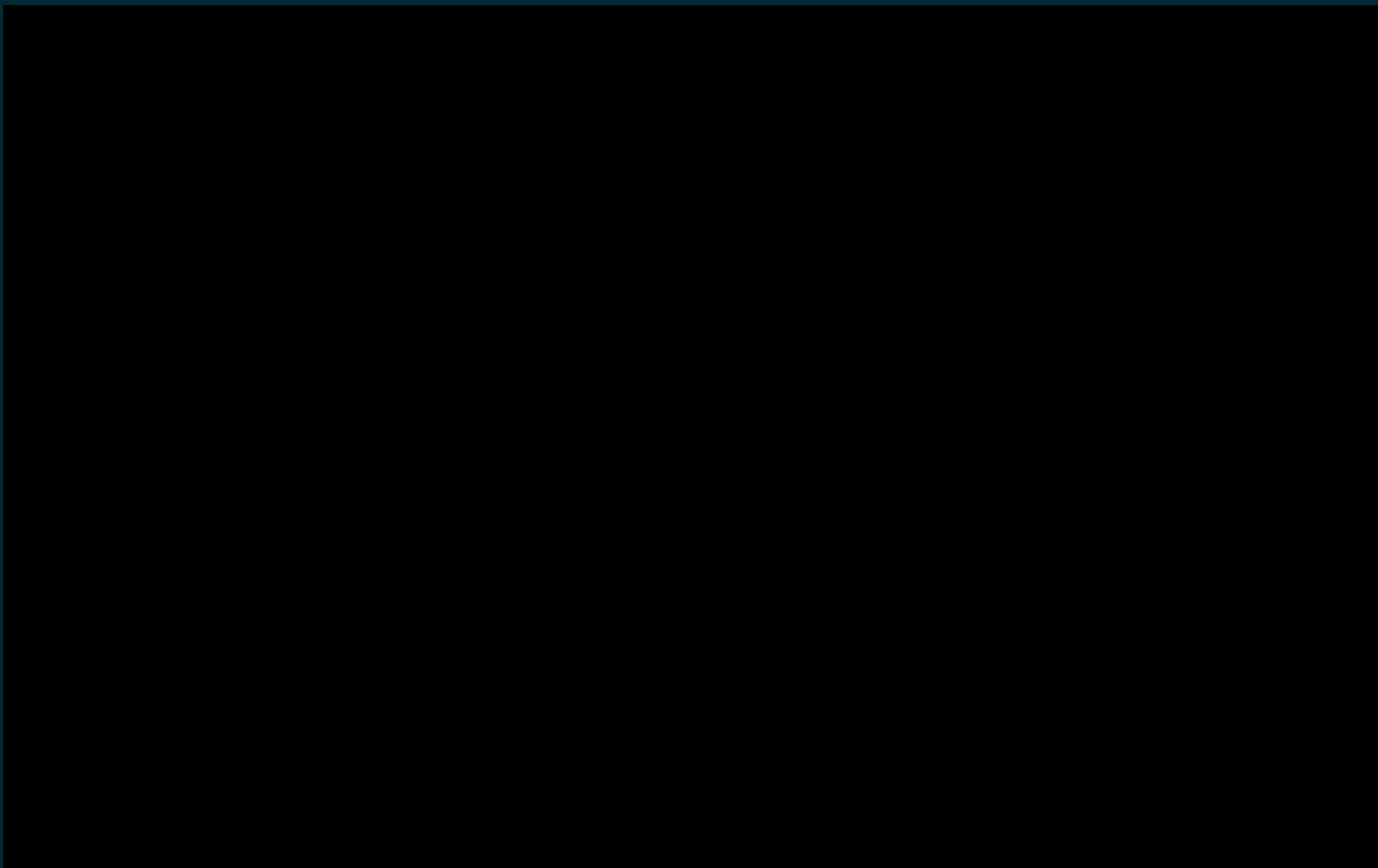
Riduco l'intensità luminosa dei LED

Spengo i LED dopo qualche secondo

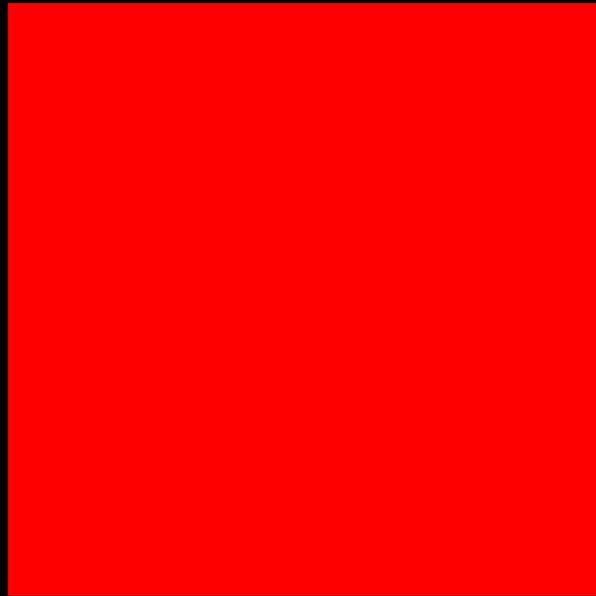
Faccio lampeggiare i LED

Faccio lampeggiare i LED alternativamente

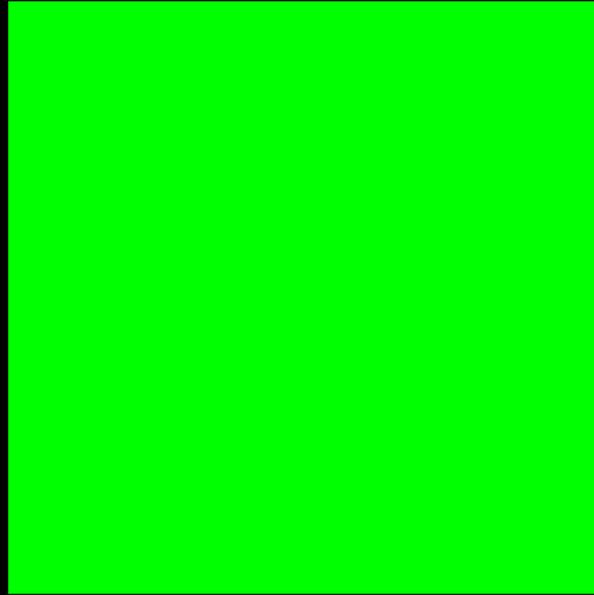
# Colori RGB



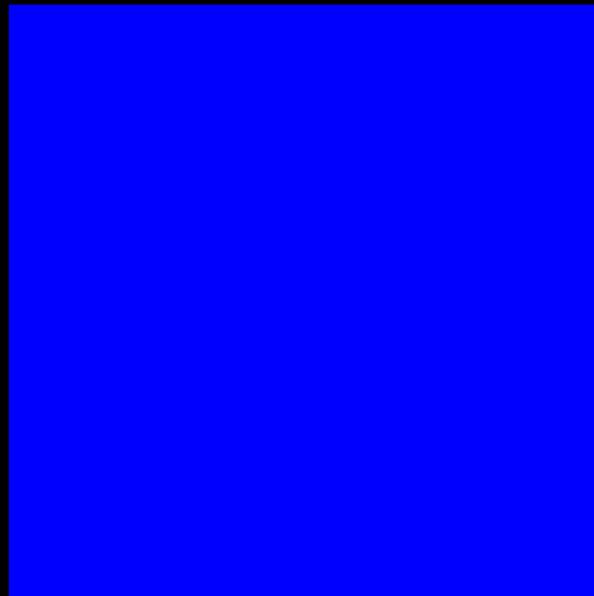
Rosso (Red)



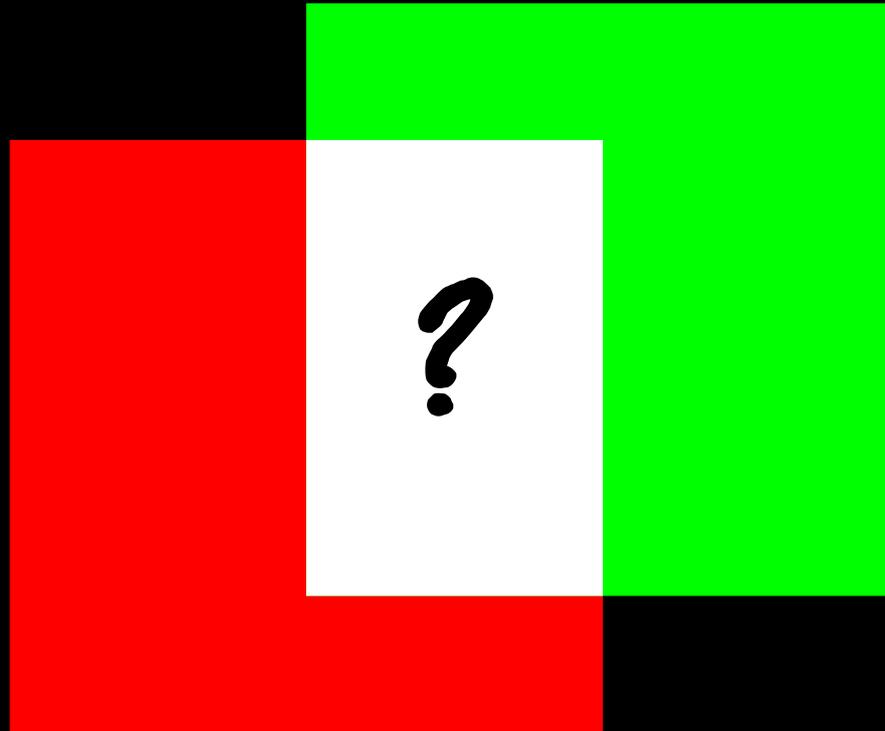
Verde (Green)



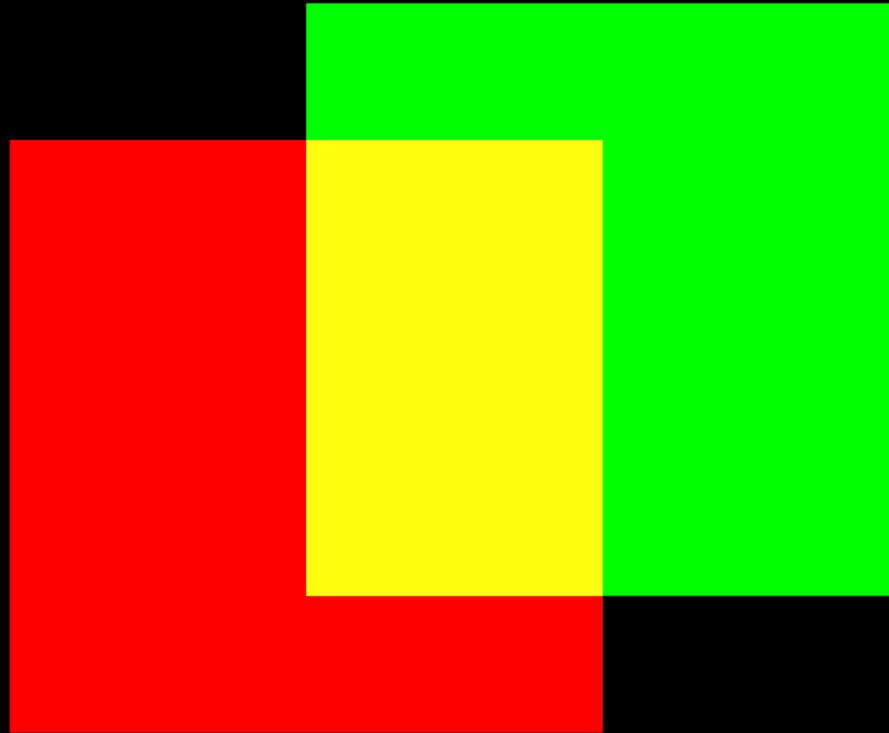
Blu (Blue)



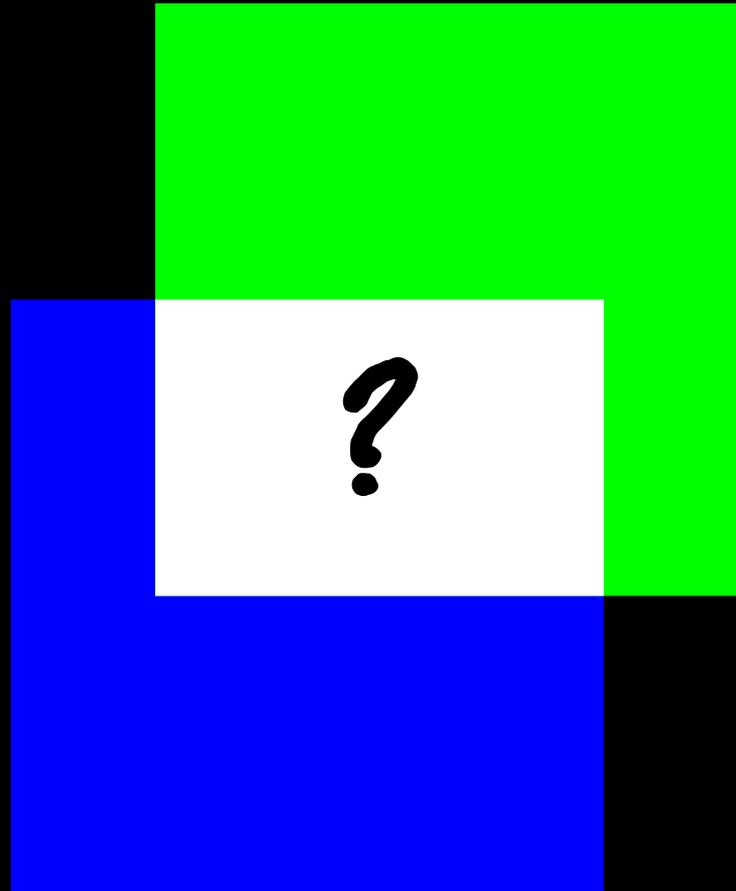
Rosso + Verde



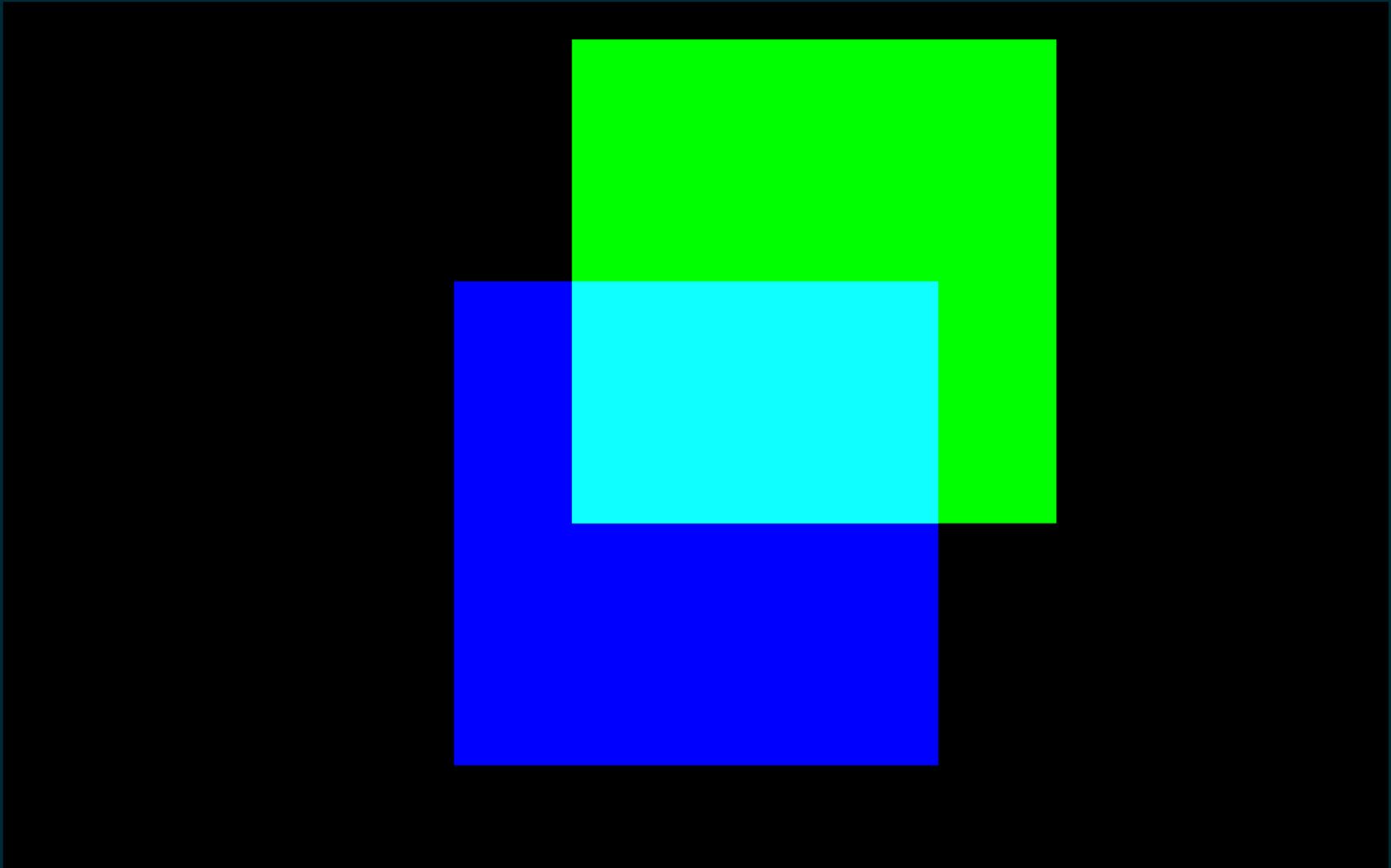
Giallo



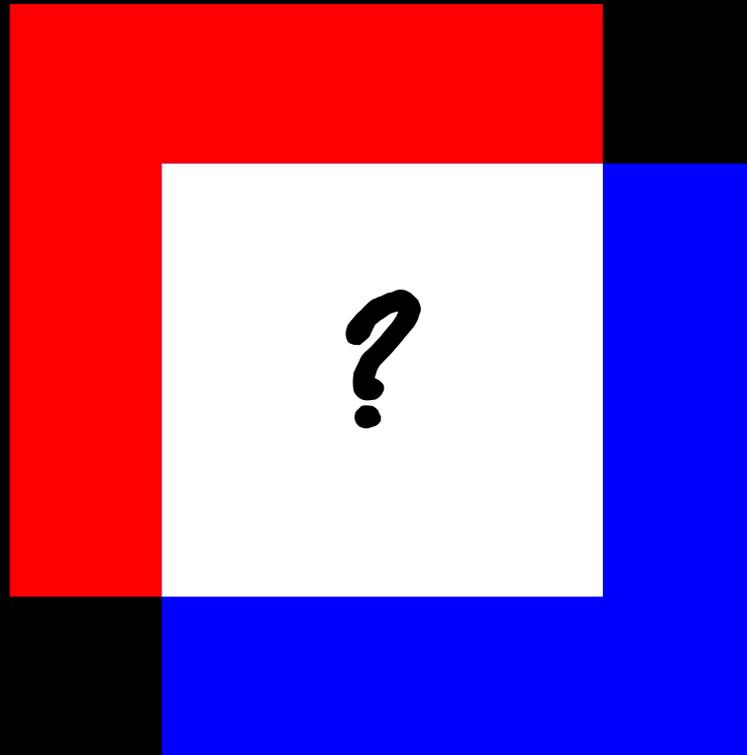
Verde + Blu



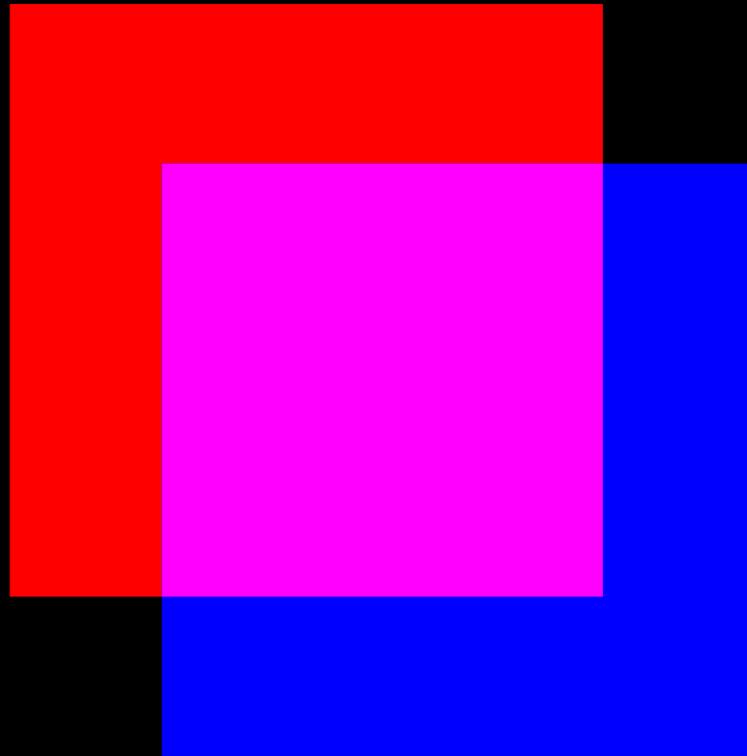
Ciano



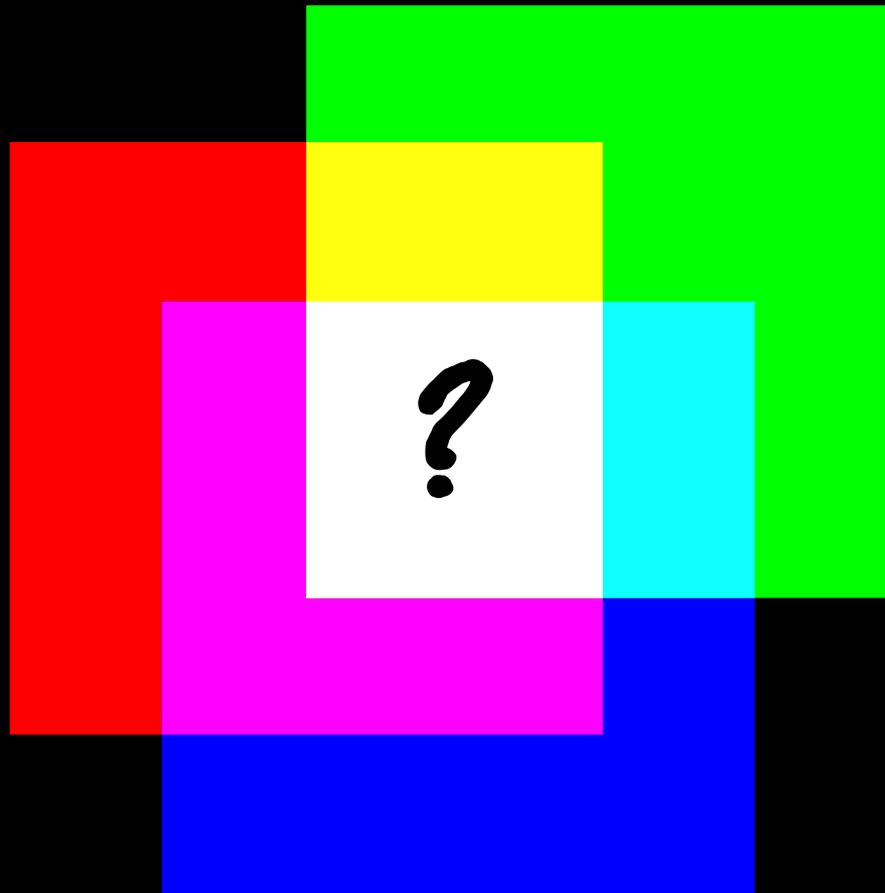
Rosso + Blu



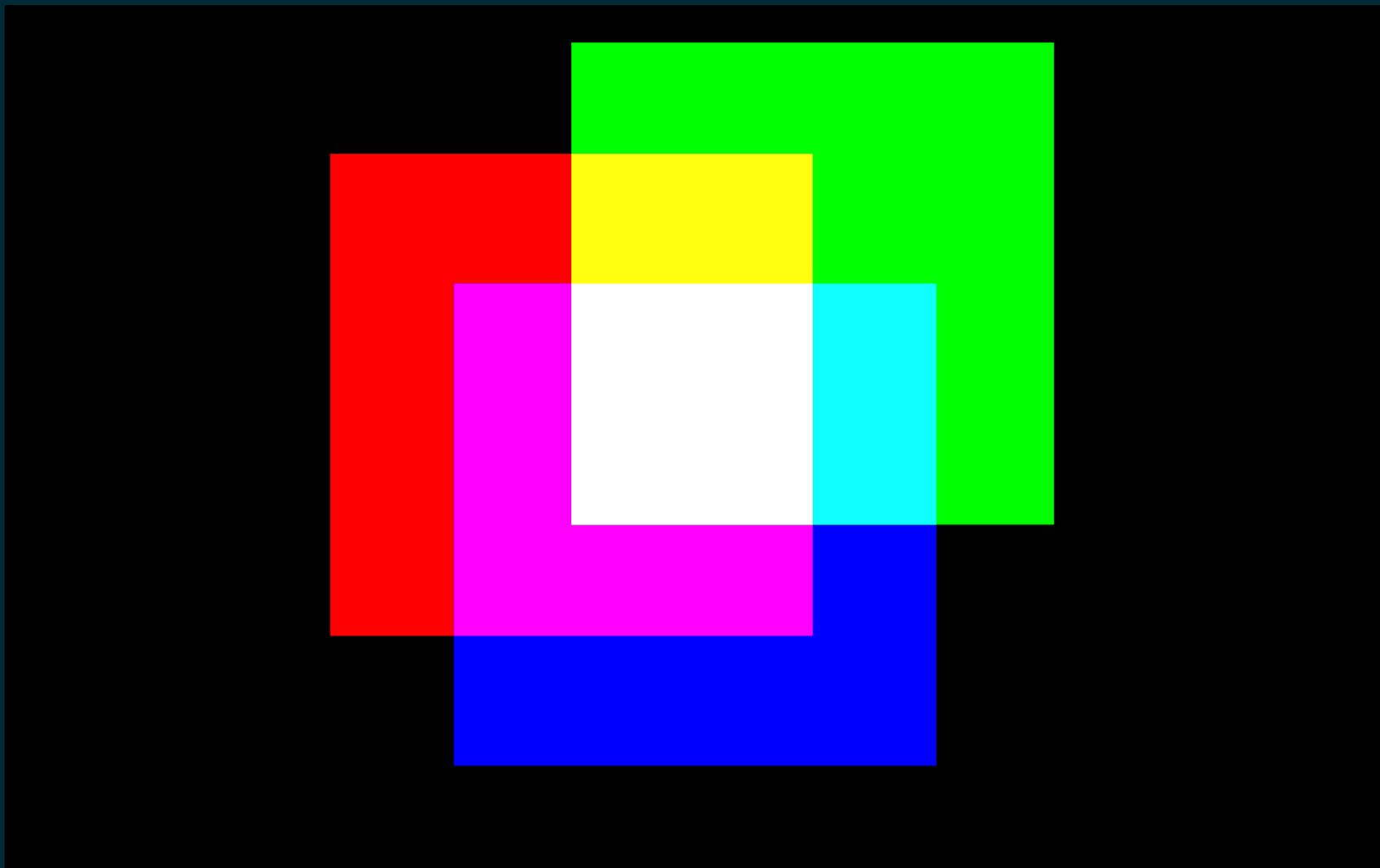
Magenta



Rosso + Verde + Blu



Bianco



# Esercitazione

# Esercitazione

Simulo un semaforo

# Esercitazione

Simulo un semaforo

il verde dura 5 secondi

il giallo dura 2 secondi

il rosso dura 4 secondi

# Esercitazione

Simulo un semaforo

Cambio il colore dei LED

# Esercitazione

Simulo un semaforo

Cambio il colore dei LED

a caso, ogni secondo un colore diverso

# Esercitazione

Simulo un semaforo

Cambio il colore dei LED ogni secondo

# Esercitazione

Simulo un semaforo

Cambio il colore dei LED ogni secondo

Replico il lampeggiante della polizia

# Esercitazione

Simulo un semaforo

Cambio il colore dei LED ogni secondo

Replico il lampeggiante della polizia

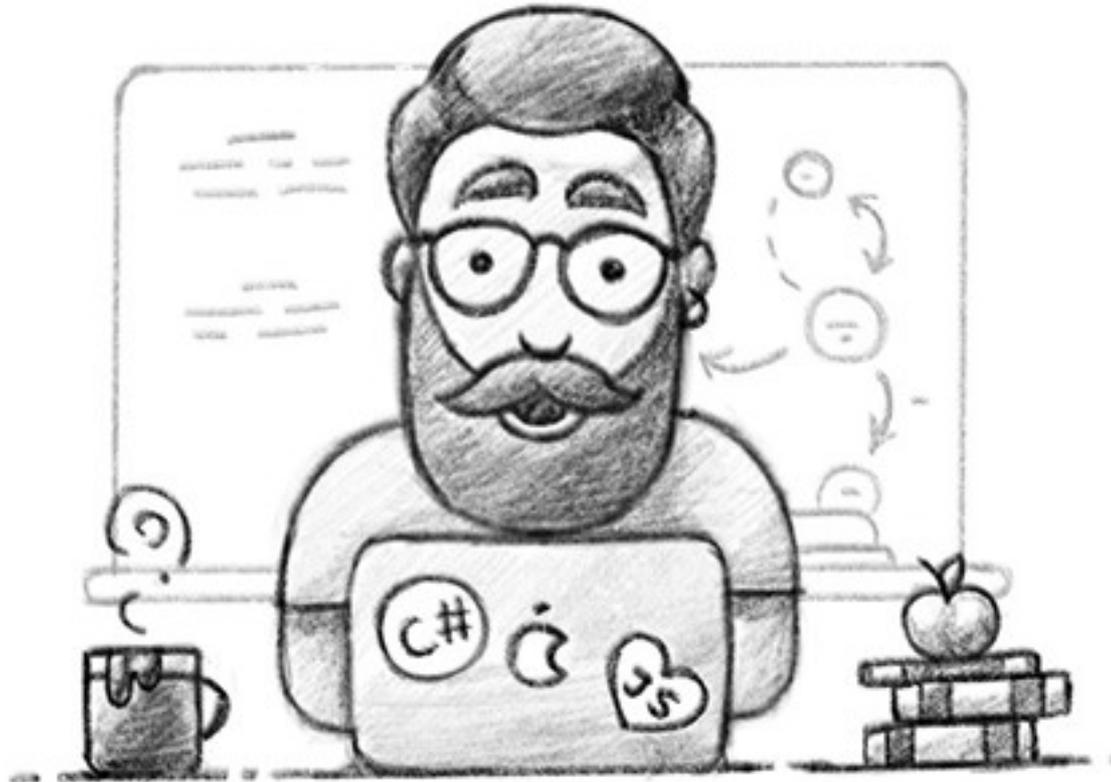
4 lampeggi blu "stroboscopici"

(LED accesi per 20ms, spenti per 80ms)

alternati sui due LED sinistro/destro

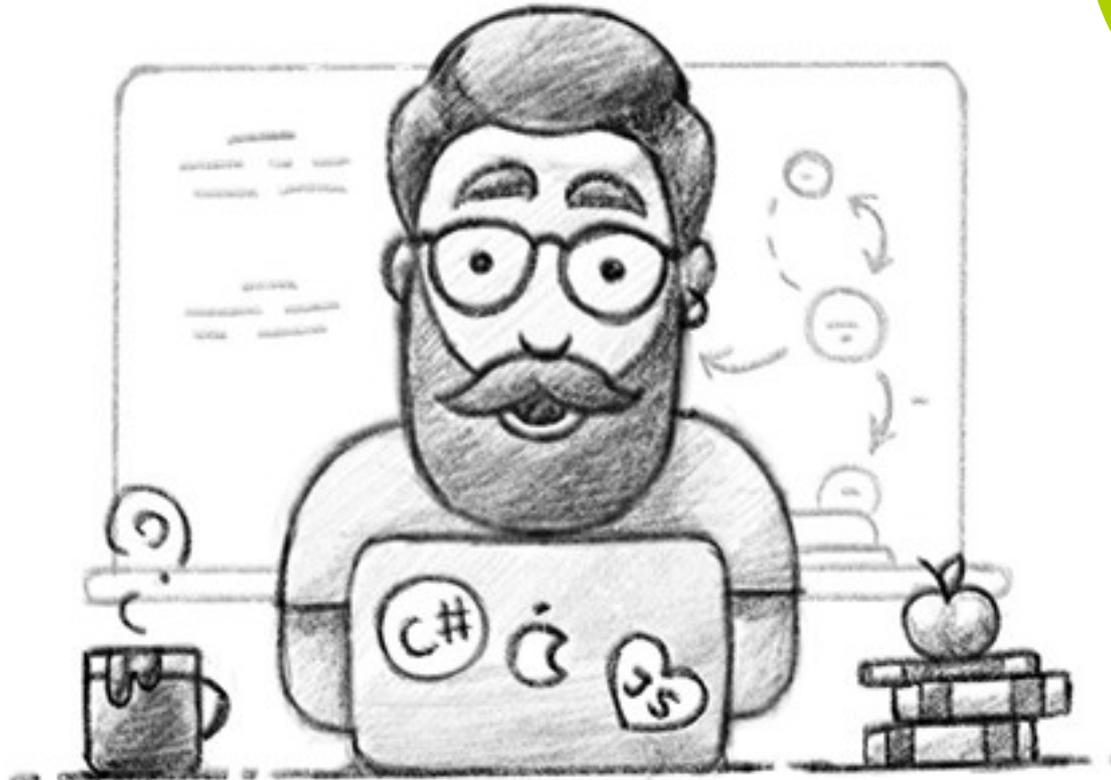
Programmazione

# Il programmatore

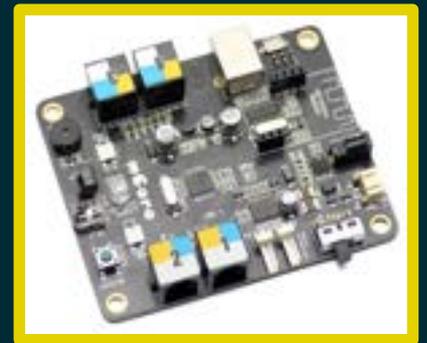
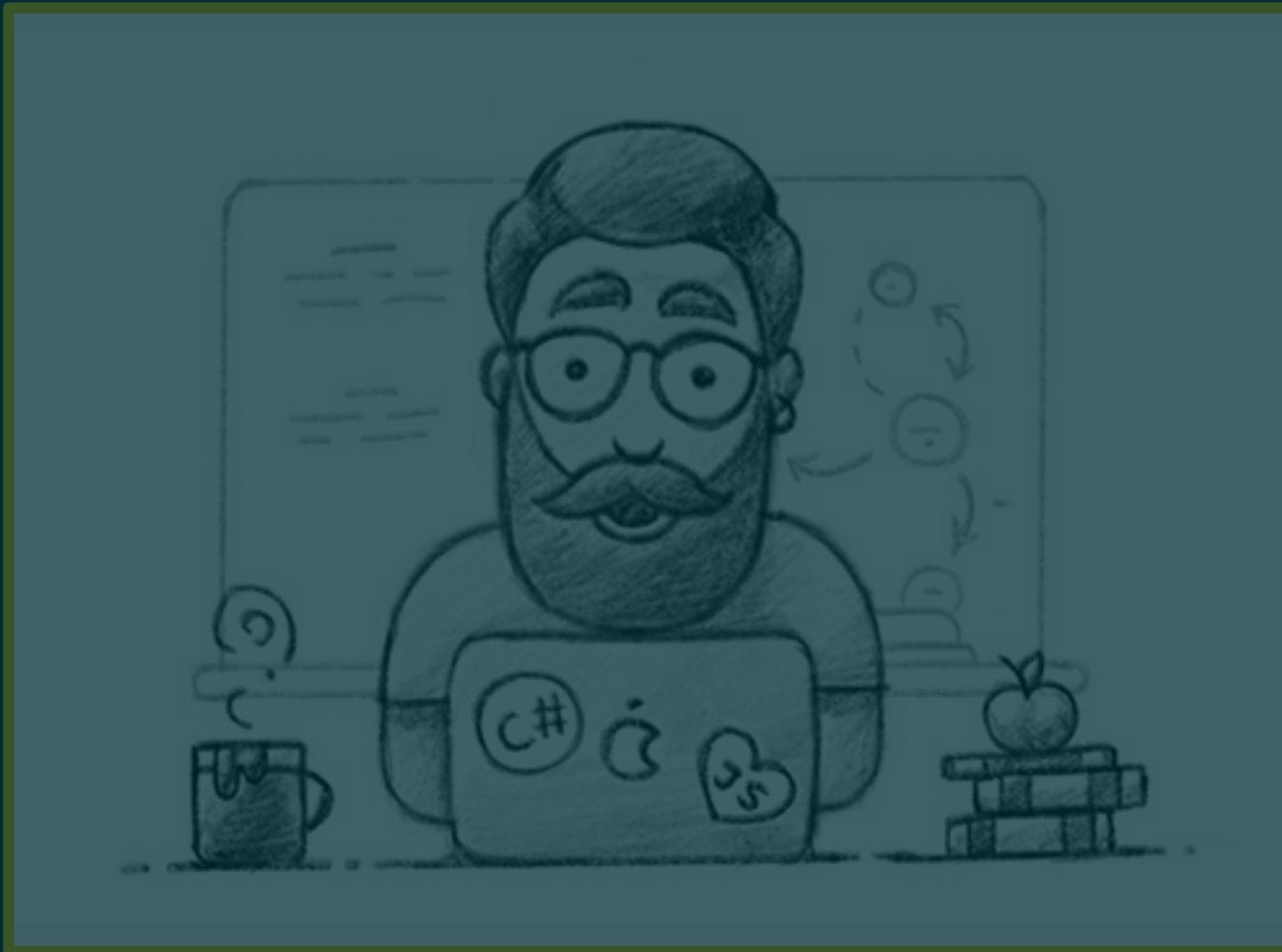


# Linguaggio naturale

Accendi i LED  
di rosso!

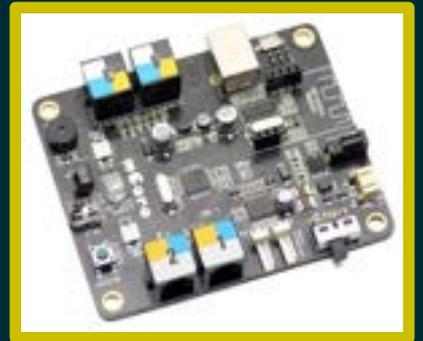
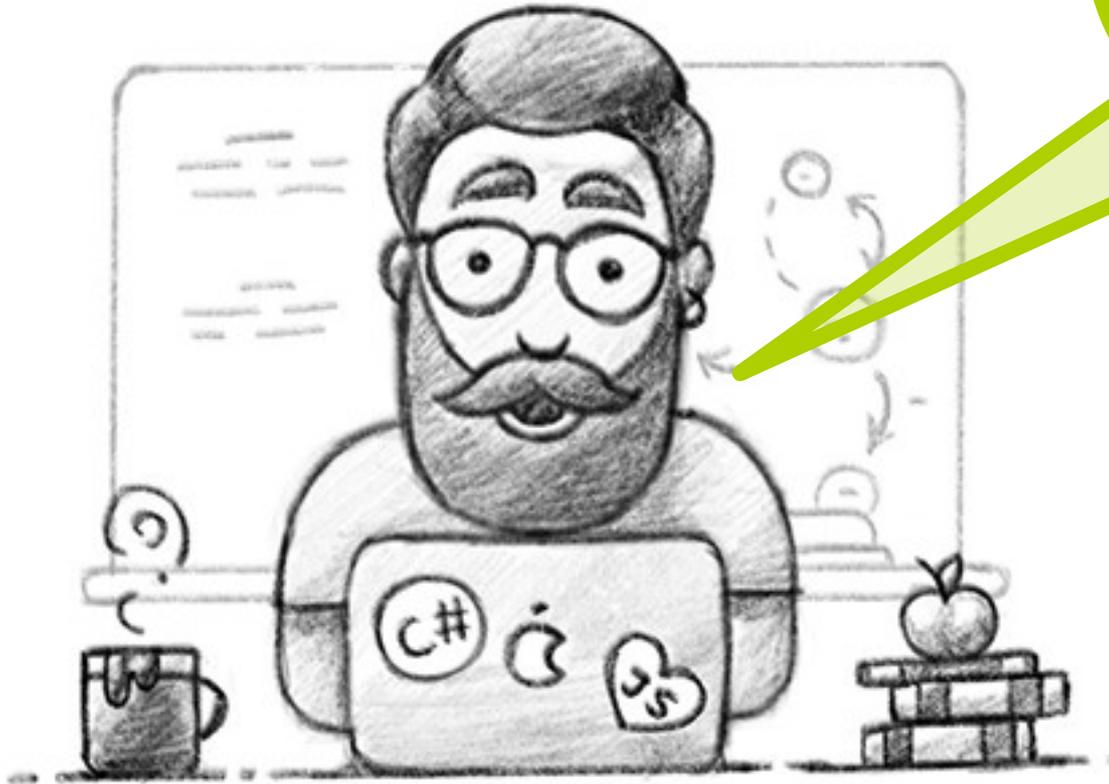


# Il circuito da programmare

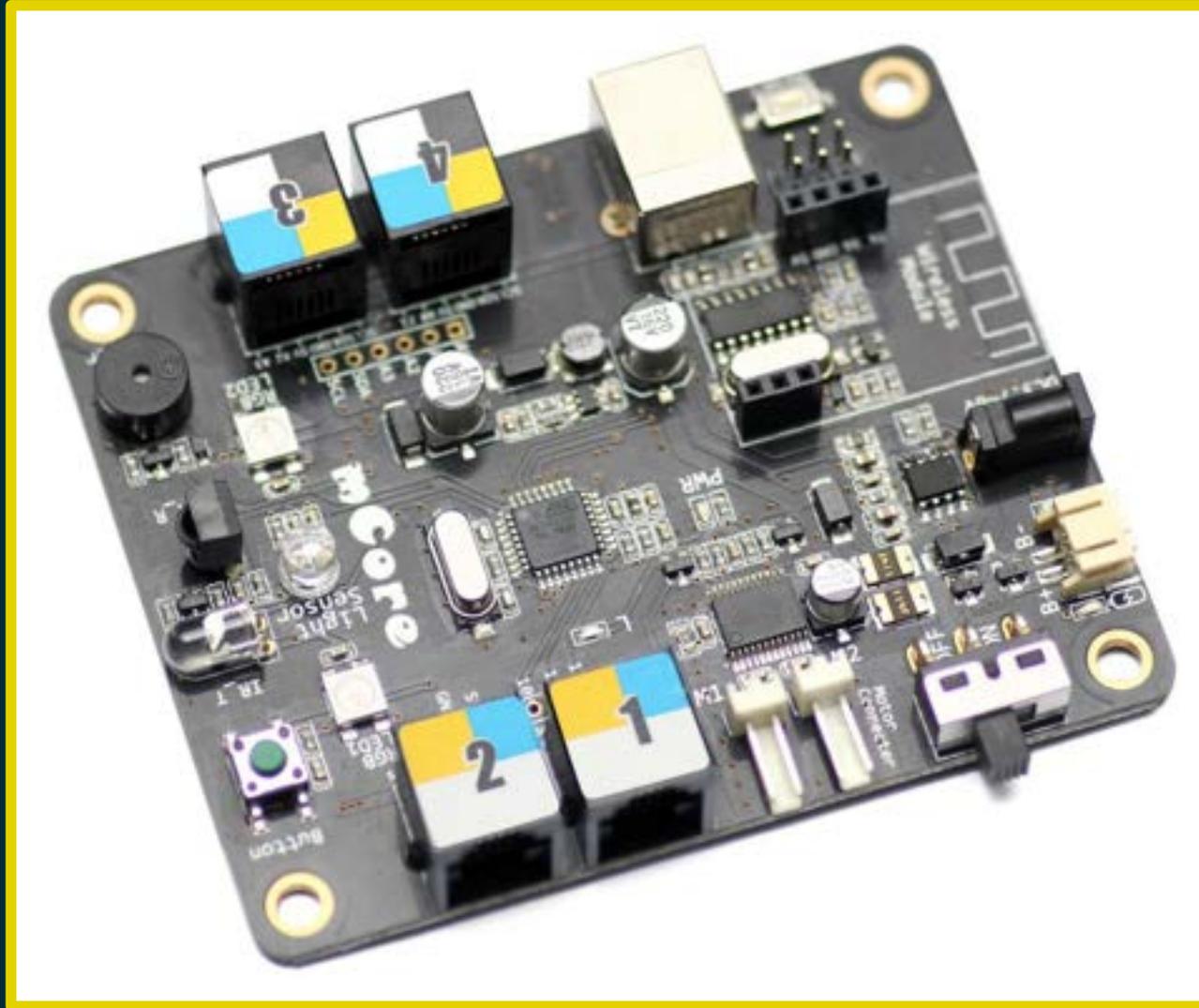


# Programmazione

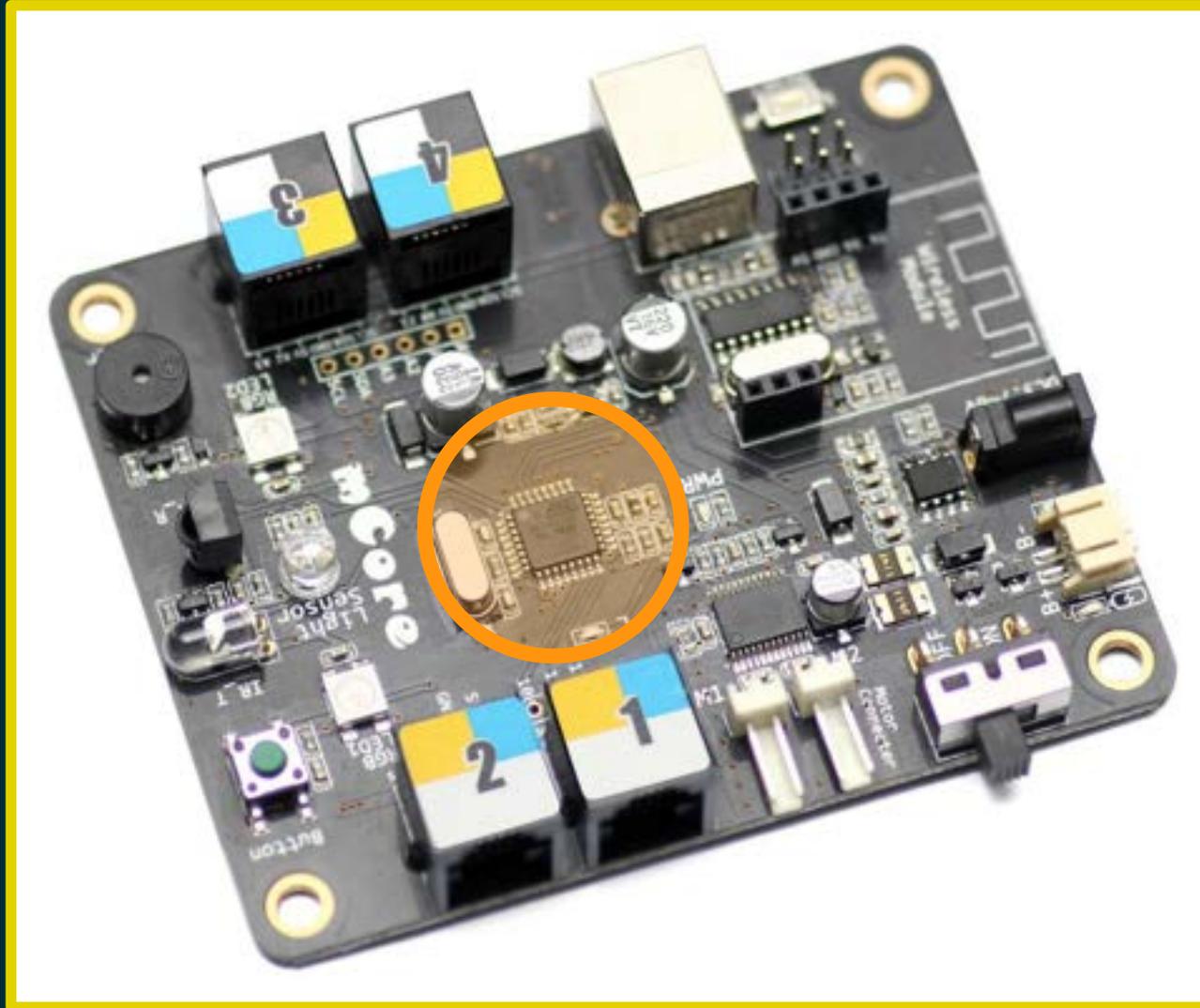
Accendi i LED  
di rosso?!



# Il circuito da programmare



# Il microcontrollore



# Linguaggio binario

# Linguaggio binario

000011001001010000110001100000000000  
00011001001010010001011000000000...

# Linguaggio binario

00001100100101000011000011000000000000  
00011001001010001000101100000000000000  
00110010010100010000101100000000000000  
01100100101000011010100000000100...

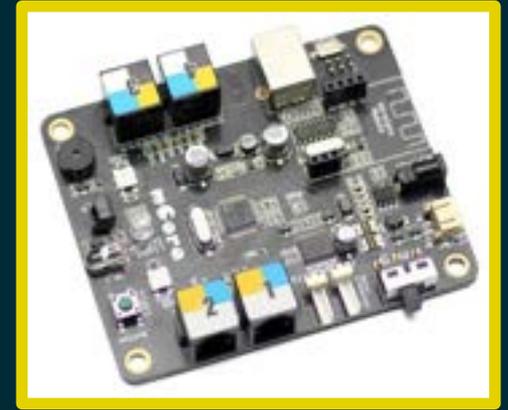
# Linguaggio binario

00001100100101000011000011000000000000  
00011001001010001000101100000000000000  
00110010010100010000101100000000000000  
0110010010100001101010000000010000000011  
001001010000110101000000001000000001100  
100101000011010100000000100...

# Linguaggio binario

36656 bit!

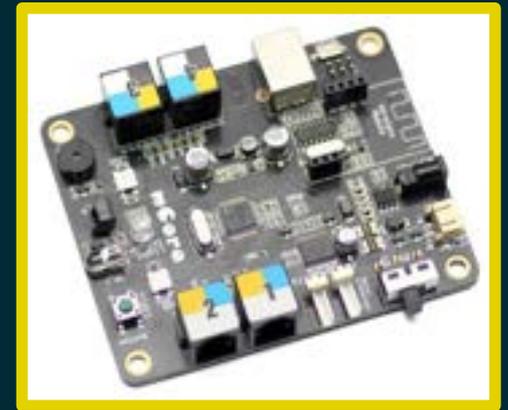
# Un dialogo difficile!



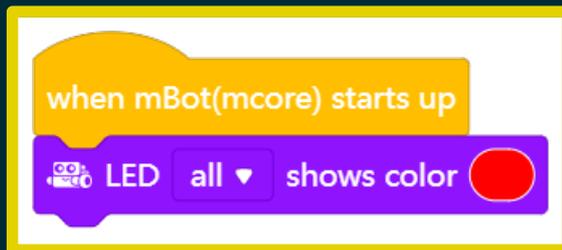
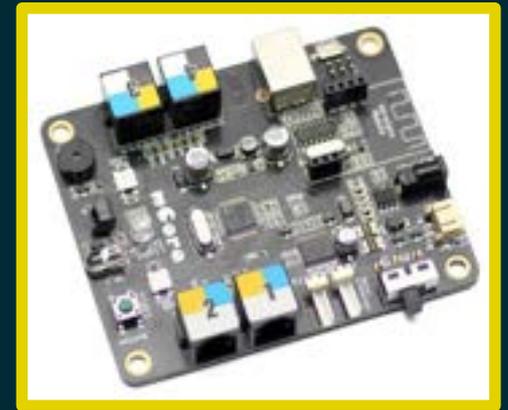
Accendi i LED di rosso!

000011001001010001100011  
00000000000001100100101  
0010001011000000000000...

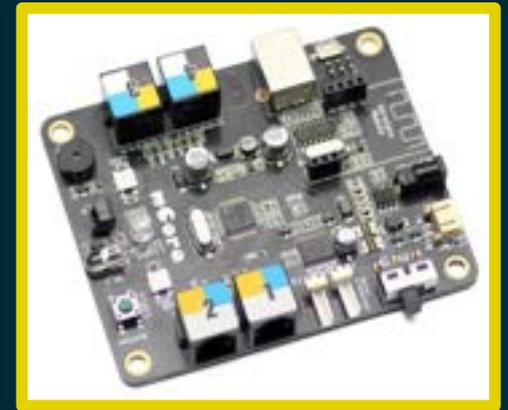
# Un interprete in aiuto



# Linguaggio di alto livello



# Linguaggio di basso livello



```
000011001001010001100011  
00000000000001100100101  
00100010110000000000...
```

Un passaggio intermedio

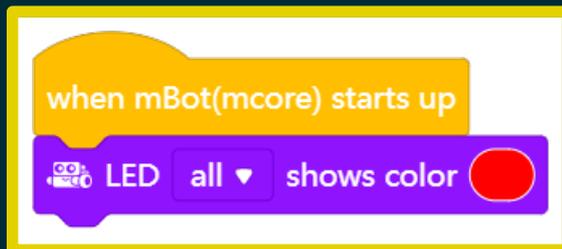
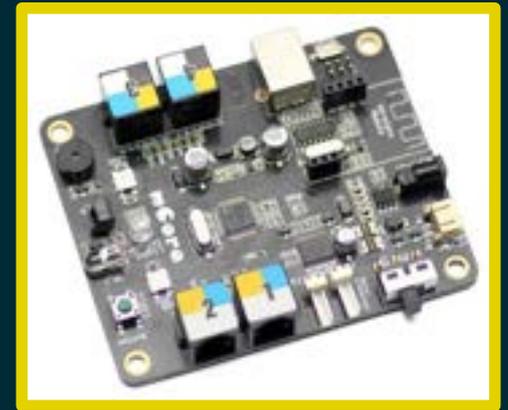
Un passaggio intermedio

Arduino C

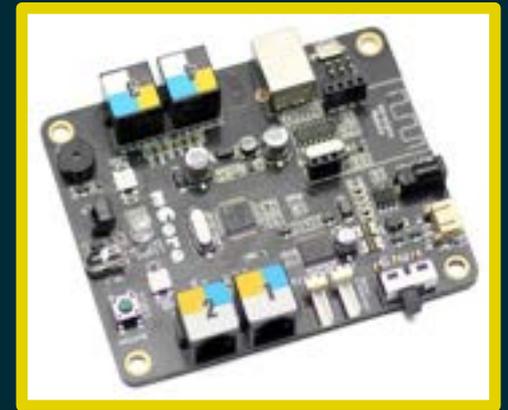
# Linguaggio intermedio

```
1 // generated by mBlock5 for mBot
2 // codes make you happy
3
4 #include <MeMCore.h>
5 #include <Arduino.h>
6 #include <Wire.h>
7 #include <SoftwareSerial.h>
8
9 MeRGBLed rgbled_7(7, 2);
10
11 void _delay(float seconds) {
12     long endTime = millis() + seconds * 1000;
13     while(millis() < endTime) _loop();
14 }
15
16 void setup() {
17     rgbled_7.fillPixelsBak(0, 2, 1);
18
19     rgbled_7.setColor(0, 255, 0, 0);
20     rgbled_7.show();
21 }
22
23
24 void _loop() {
25 }
26
27 void loop() {
28     _loop();
29 }
30
```

# Scrittura del programma

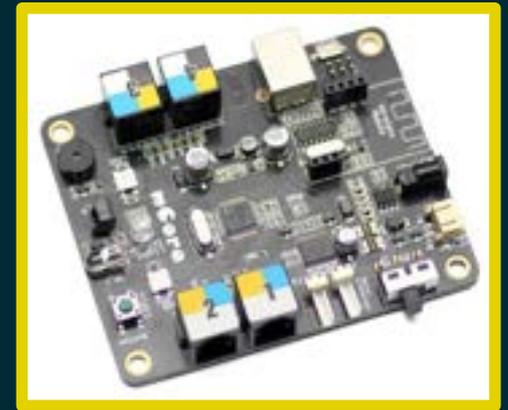


# Traduzione "al volo"



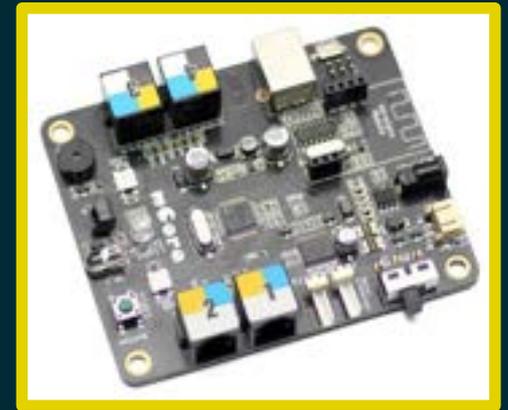
```
16 void setup() {  
17     rgbled_7.fillPixelsBak(0, 2, 1);  
18  
19     rgbled_7.setColor(0, 255, 0, 0);  
20     rgbled_7.show();  
21  
22 }
```

# Compilazione



```
000011001001010001100011000000000000  
0110010010100100010110000000000000110  
01001010010001011000000000000011001...
```

# Trasmissione



000011001001010001100011000000000000  
0110010010100100010110000000000000110  
010010100100010110000000000000011001...

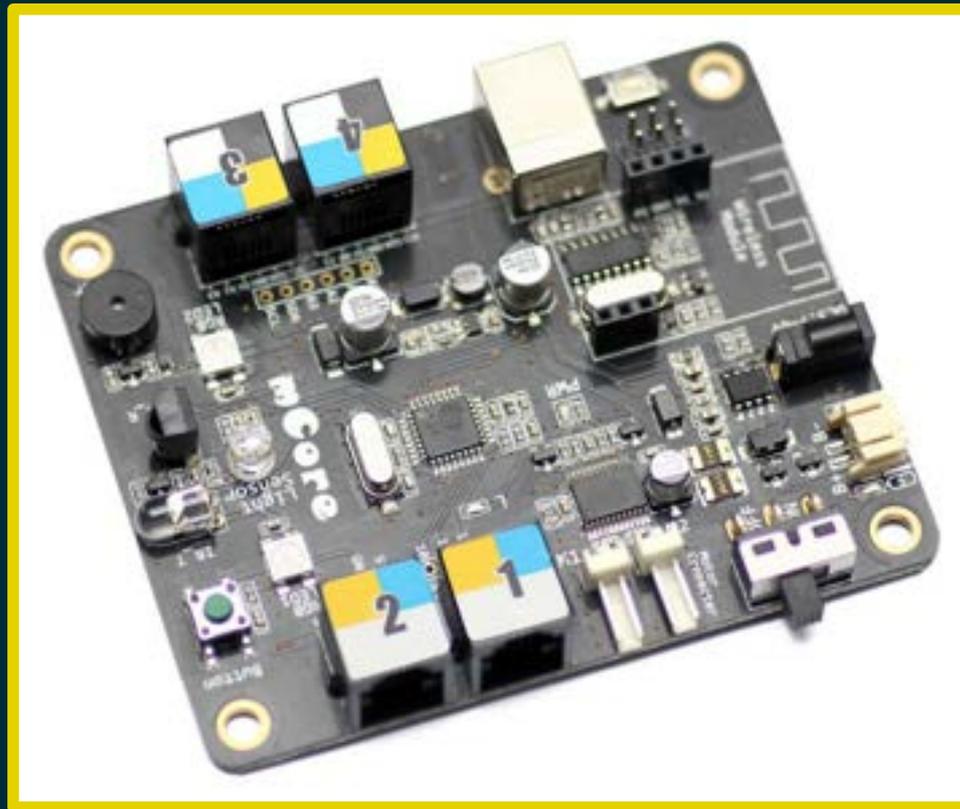
# Hardware vs Software

# Hardware vs Software

Hardware: la parte tangibile del sistema.

# Hardware vs Software

Hardware: la parte tangibile del sistema.



# Hardware vs Software

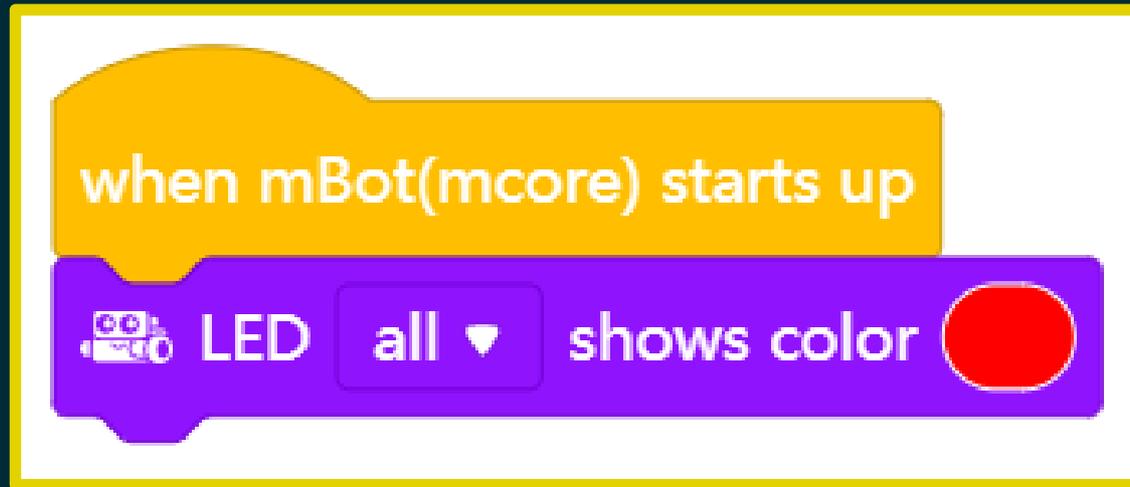
Hardware: la parte tangibile del sistema.

Software: i programmi eseguiti dall'H/W.

# Hardware vs Software

Hardware: la parte tangibile del sistema.

Software: i programmi eseguiti dall'H/W.



# Hardware vs Software

Hardware: la parte tangibile del sistema.

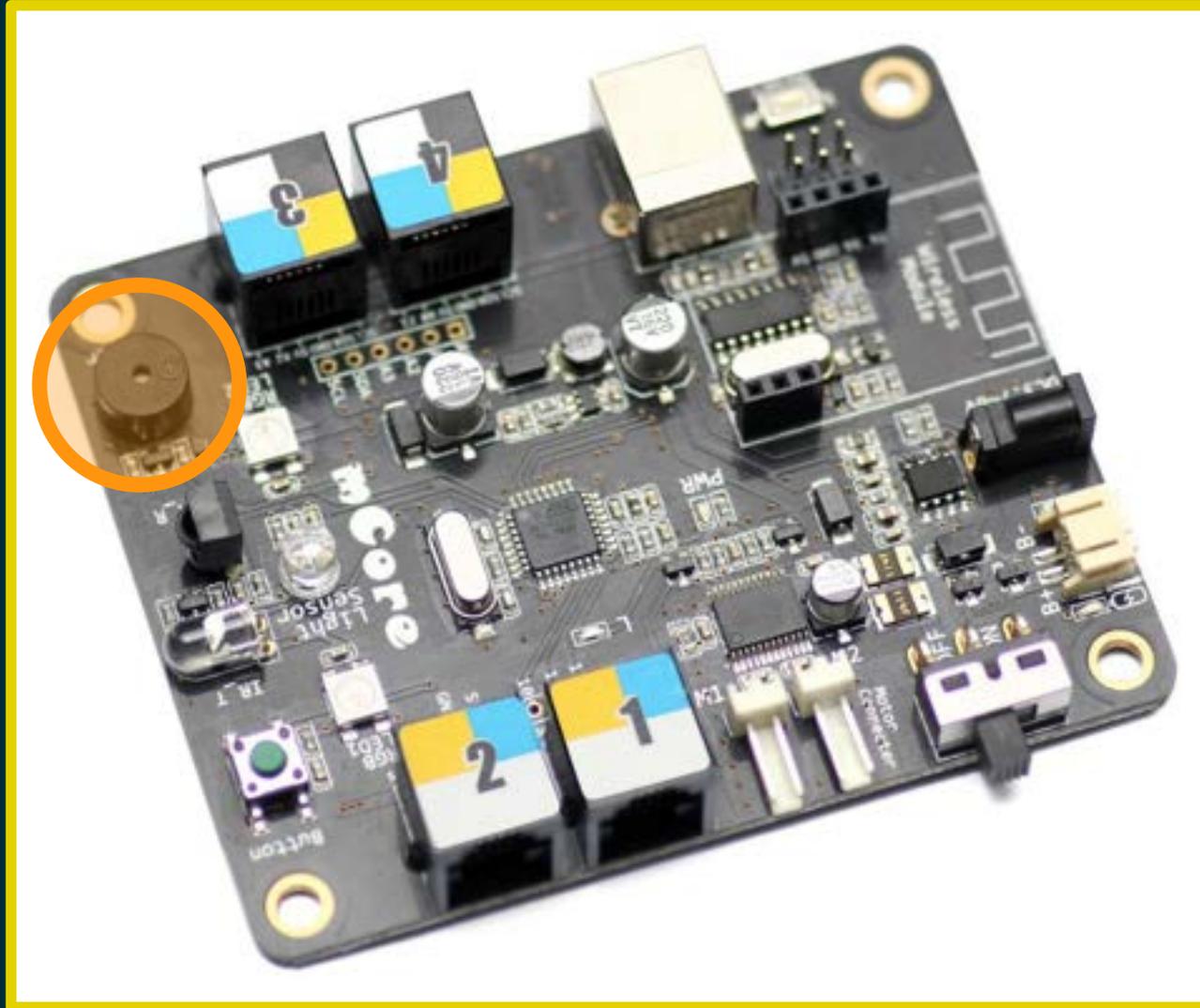
Software: i programmi eseguiti dall'H/W.

Firmware: software integrato  
permanentemente in un componente  
elettronico.

# Programma n. 2

Far suonare il cicalino

# Cicalino



# Programma n. 2

Far suonare il cicalino

# Programma n. 2

Demo!

# Esercitazione

Suono la scala diatonica

# Esercitazione

Suono la scala diatonica

C4 D4 E4 F4 G4 A4 B4 C5

# Esercitazione

Suono la scala diatonica

# Esercitazione

Suono la scala diatonica

Simulo la sirena della polizia

# Esercitazione

Suono la scala diatonica

Simulo la sirena della polizia

G4 per un secondo

E5 per un sesto di secondo

G4 per un sesto di secondo

E5 per un sesto di secondo

# Esercitazione

Suono la scala diatonica

Simulo la sirena della polizia

G4 per un secondo

E5 per un sesto di secondo

G4 per un sesto di secondo

E5 per un sesto di secondo

**RIPETERE!**

# Esercitazione

Suono la scala diatonica

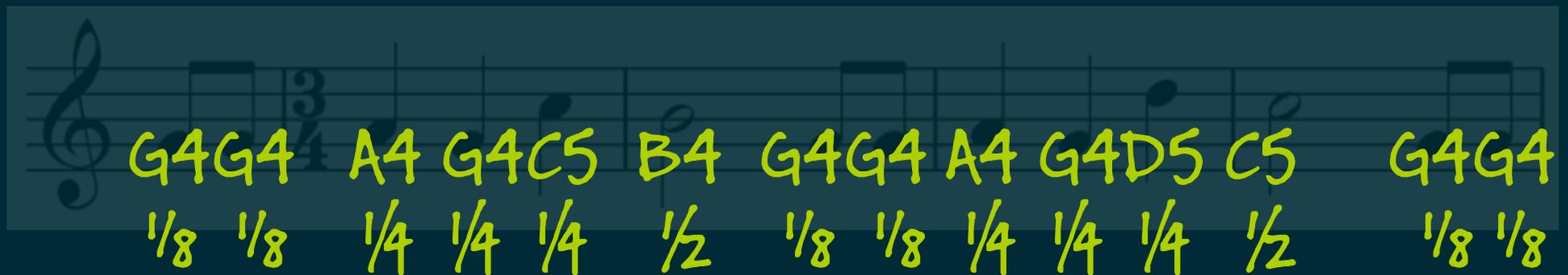
Simulo la sirena della polizia

Suono la melodia di "Tanti auguri"



# Esercitazione

## Melodia di "Tanti auguri"



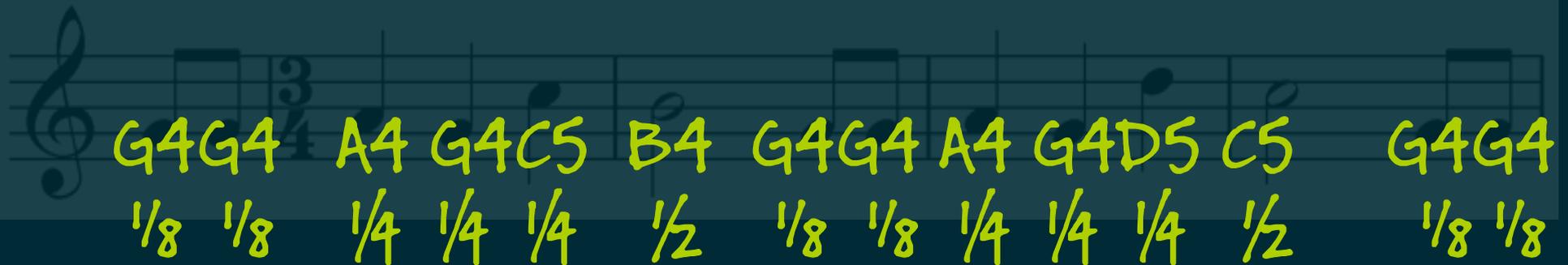
G4G4 A4 G4C5 B4 G4G4 A4 G4D5 C5 G4G4  
1/8 1/8 1/4 1/4 1/4 1/2 1/8 1/8 1/4 1/4 1/4 1/2 1/8 1/8



Musical staff showing the melody of "Tanti auguri" in treble clef, 3/4 time signature. The notes are: G4 (quarter), A4 (quarter), G4 (quarter), C5 (quarter), B4 (quarter), G4 (quarter), A4 (quarter), G4 (quarter), D5 (quarter), C5 (quarter), G4 (quarter), G4 (quarter).

# Esercitazione

## Melodia di "Tanti auguri"



Musical staff showing the first line of the melody for "Tanti auguri". The staff is in treble clef and 3/4 time. The notes are G4, G4, A4, G4, C5, B4, G4, G4, A4, G4, D5, C5, G4, G4. The chord symbols are G4, G4, A4, G4, C5, B4, G4, G4, A4, G4, D5, C5, G4, G4. The durations are 1/8, 1/8, 1/4, 1/4, 1/4, 1/2, 1/8, 1/8, 1/4, 1/4, 1/4, 1/2, 1/8, 1/8.



Musical staff showing the second line of the melody for "Tanti auguri". The staff is in treble clef and 3/4 time. The notes are G5, E5, C5, B4, A4, F5, F5, E5, C5, D5, C5. The chord symbols are G5, E5, C5, B4, A4, F5, F5, E5, C5, D5, C5. The durations are 1/4, 1/4, 1/4, 1/4, 1/4, 1/8, 1/8, 1/4, 1/4, 1/4, 1/2.

Concorrenza

# Concorrenza

Riesco a far lampeggiare i LED...

# Concorrenza

Riesco a far lampeggiare i LED  
e suonare la sirena della polizia..

# Concorrenza

Riesco a far lampeggiare i LED  
e suonare la sirena della polizia..

**CONTEMPORANEAMENTE!?**

# Concorrenza

Riesco a far lampeggiare i LED  
e suonare la sirena della polizia..

**CONTEMPORANEAMENTE!?**

In generale no, il microcontrollore non lo  
consente - è troppo limitato.

# mCore vs iPhone 7

	mCore	iPhone 7

# mCore vs iPhone 7

	mCore	iPhone 7
core	1	2

# mCore vs iPhone 7

	mCore	iPhone 7
core	1	2
parola (bit)	8	64

# mCore vs iPhone 7

	mCore	iPhone 7
core	1	2
parola (bit)	8	64
frequenza	16Mhz	2.34Ghz

# mCore vs iPhone 7

	mCore	iPhone 7
core	1	2
parola (bit)	8	64
frequenza	1.46GHz	2.34GHz

x 146!

# mCore vs iPhone 7

	mCore	iPhone 7
core	1	2
parola (bit)	8	64
frequenza	16Mhz	2.34Ghz

# mCore vs iPhone 7

	mCore	iPhone 7
core	1	2
parola (bit)	8	64
frequenza	16Mhz	2.34Ghz
memoria RAM	2KB	2GB

# mCore vs iPhone 7

	mCore	iPhone 7
core	1	2
parola (bit)	8	64
frequenza	1 Mb	2 Ghz
memoria RAM	2KB	2GB

x 1.000.000!

# mCore vs iPhone 7

	mCore	iPhone 7
core	1	2
parola (bit)	8	64
frequenza	16Mhz	2.34Ghz
memoria RAM	2KB	2GB

# mCore vs iPhone 7

	mCore	iPhone 7
core	1	2
parola (bit)	8	64
frequenza	16Mhz	2.34Ghz
memoria RAM	2KB	2GB
capacità	32KB	32GB

# mCore vs iPhone 7

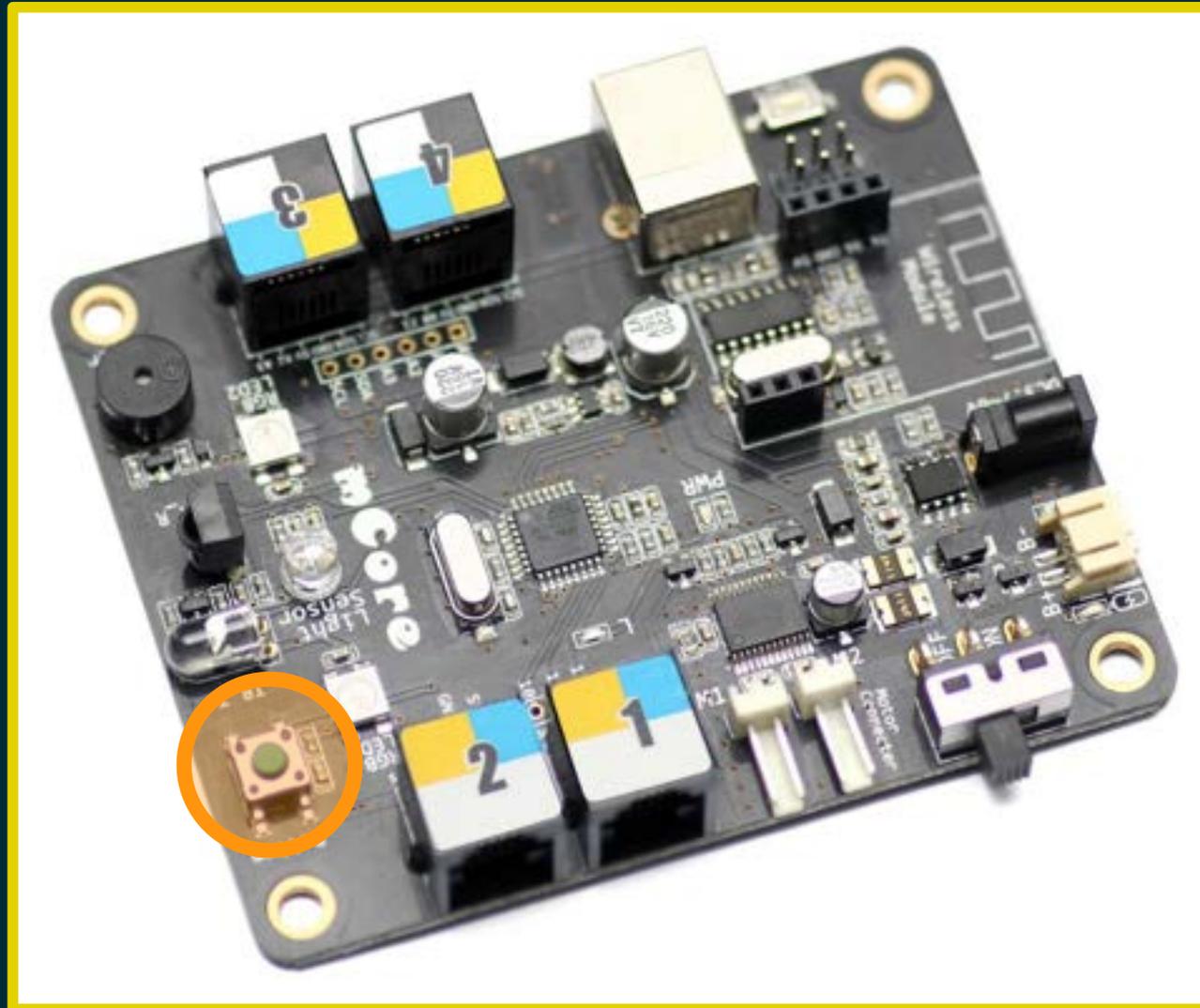
	mCore	iPhone 7
core	1	2
parola (bit)	8	64
frequenza	1 Mb	2 Gh
memoria RAM	2KB	2GB
capacità	32KB	32GB

x 1.000.000!

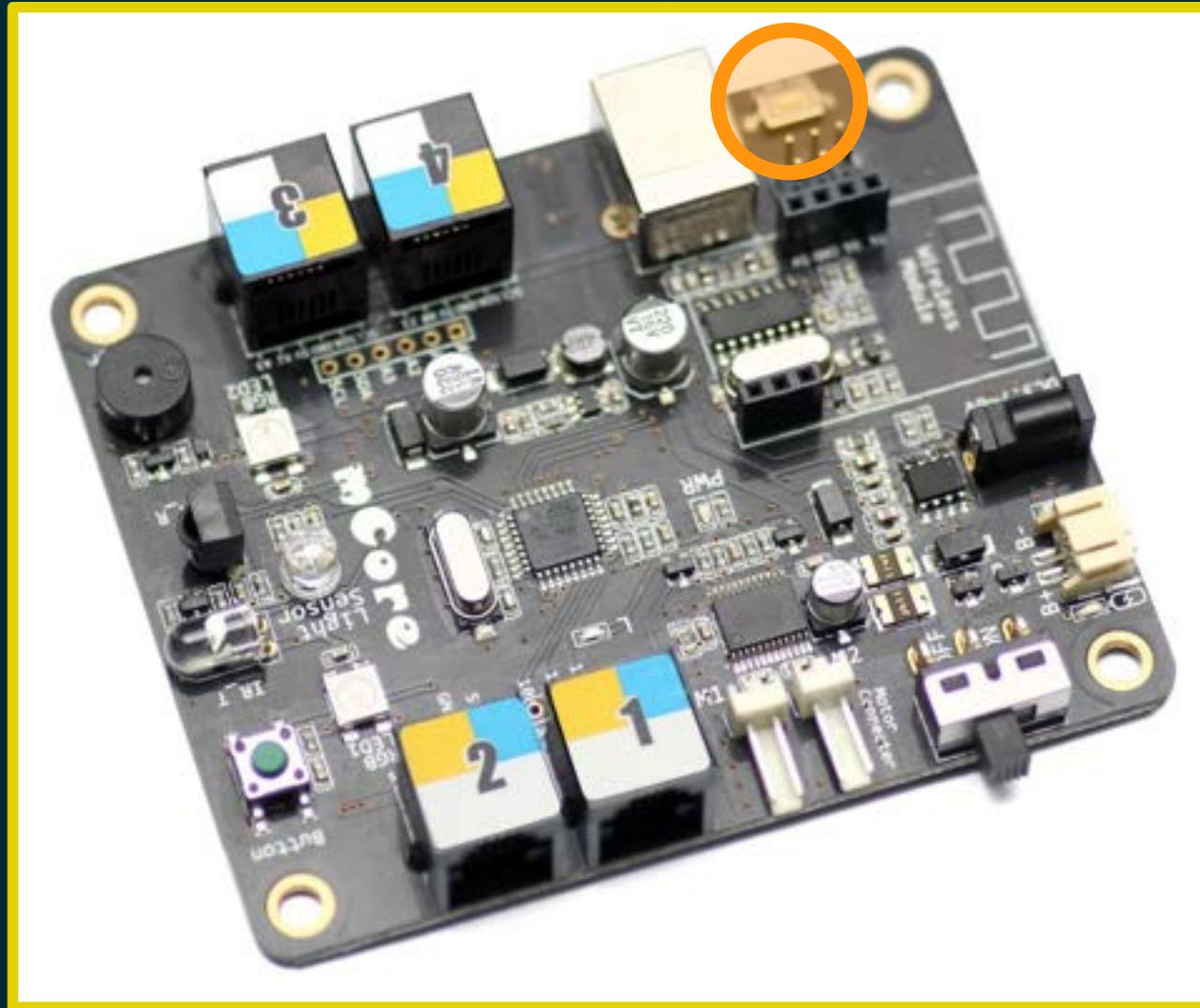
# Programma n. 3

Controllare un pulsante

# Pulsante



# Pulsante di azzeramento



# Programma n. 3

Controllare un pulsante

# Programma n. 3

Demo!

# Esercitazione

Quando il pulsante è premuto...

# Esercitazione

Quando il pulsante è premuto...

... emetto un suono di sirena

# Esercitazione

Quando il pulsante è premuto...

... emetto un suono di sirena

... cambio colore ai LED

# Esercitazione

Quando il pulsante è premuto...

... emetto un suono di sirena

... cambio colore ai LED

... accendo i LED...

# Esercitazione

Quando il pulsante è premuto...

... emetto un suono di sirena

... cambio colore ai LED

... accendo i LED, **che si devono spegnere**  
**quando rilascio il pulsante!**

Via con il montaggio!

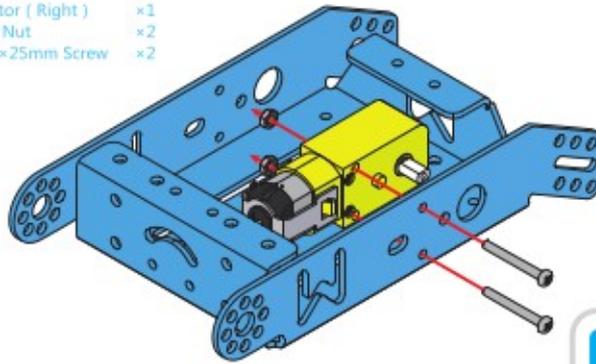




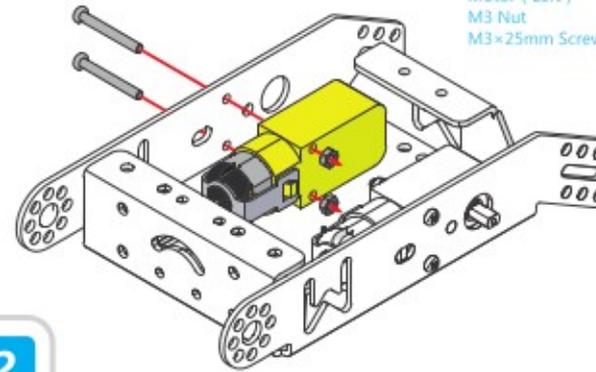
# Motori e ruote

## Assembly Instructions

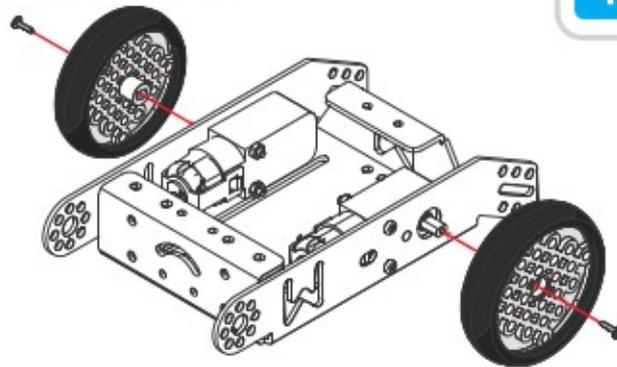
Chassis ×1  
Motor ( Right ) ×1  
M3 Nut ×2  
M3×25mm Screw ×2



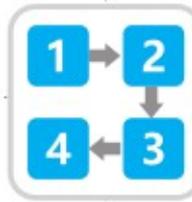
Motor ( Left ) ×1  
M3 Nut ×2  
M3×25mm Screw ×2



M2.2×9mm Self-drilling Screw ×2



Wheel ×2



# Viti e bulloni



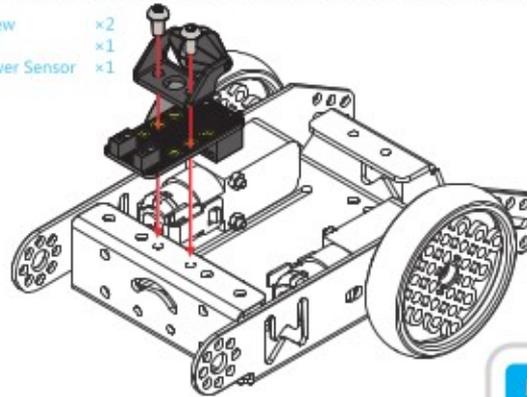
Fissaggio  
dei motori

Fissaggio  
delle ruote

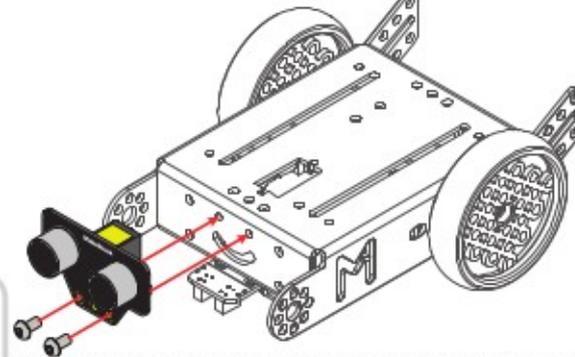
Fissaggio  
dei sensori

# Sensori esterni

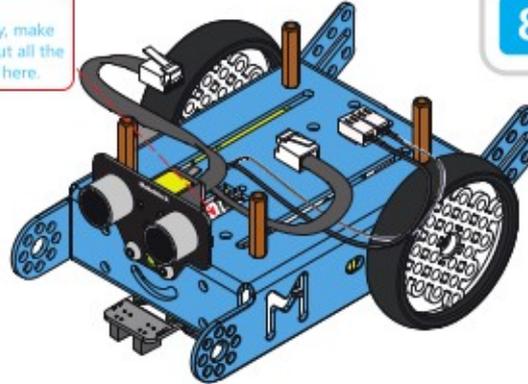
M4×8mm Screw ×2  
Mini Wheel ×1  
Me Line-follower Sensor ×1



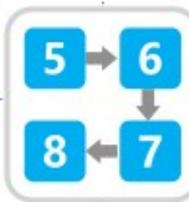
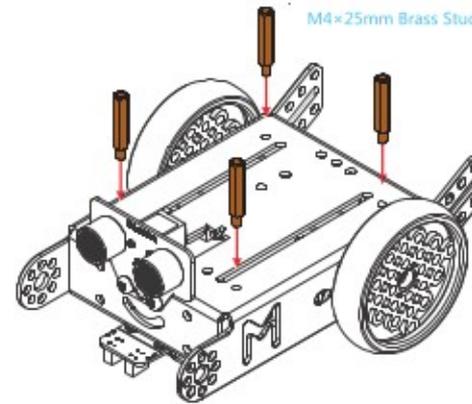
M4×8mm Screw ×2  
Me Ultrasonic Sensor ×1



**Tips:**  
After assembly, make sure to take out all the wires through here.

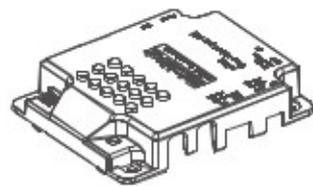


M4×25mm Brass Stud ×4

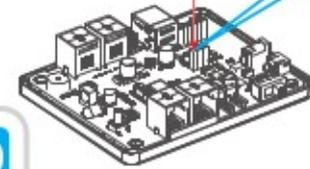
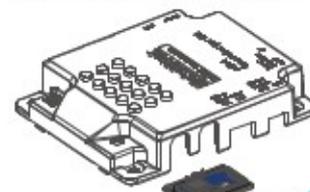


# Guscio mCore

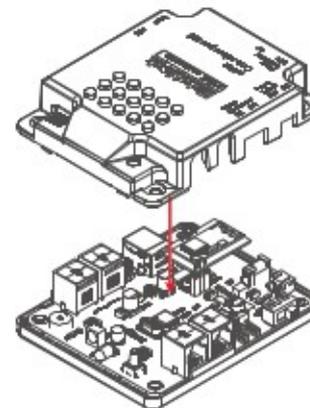
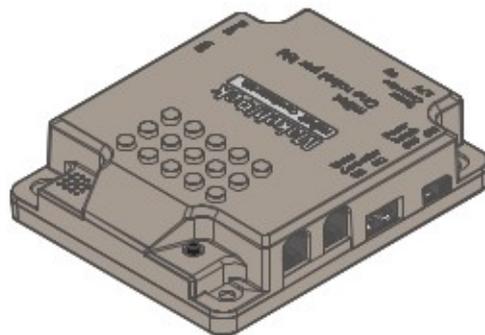
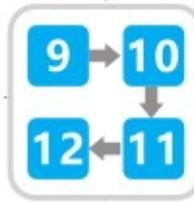
mCore Case ×1  
mCore ×1



Bluetooth/2.4G Module ×1



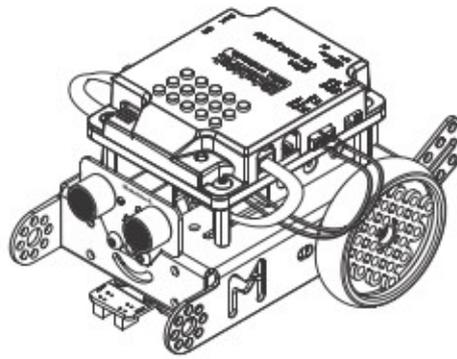
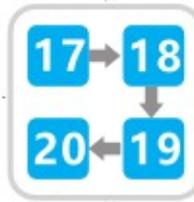
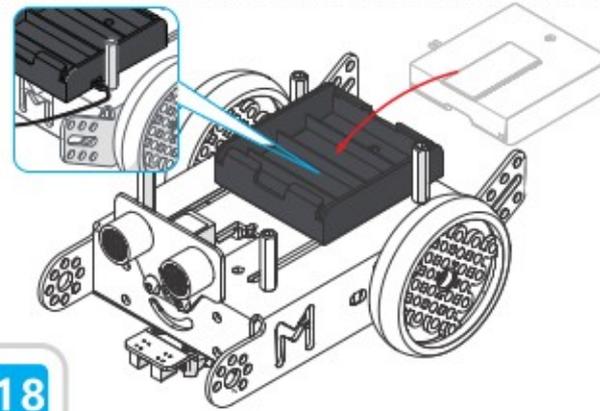
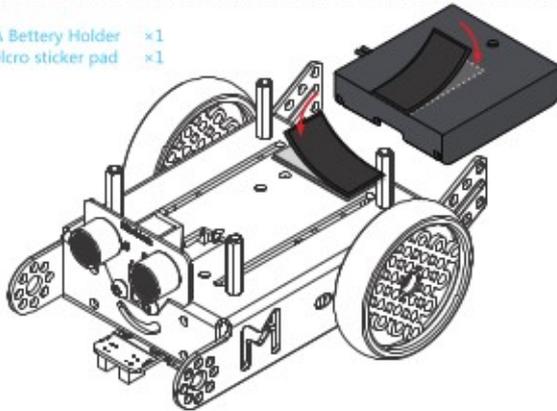
Tips:  
Please check carefully if the pins of the Bluetooth or 2.4G Module are well inserted, otherwise the chip may be damaged upon power-on.



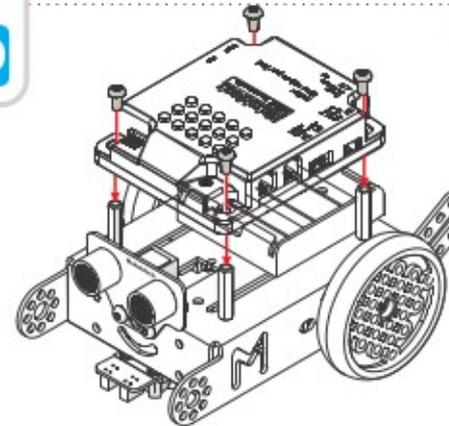
# Pacco batterie

## Powered by AA Battery (we provide two alternative power supply methods, Lithium battery or AA battery.)

AA Battery Holder ×1  
Velcro sticker pad ×1



M4×8mm Screw ×4

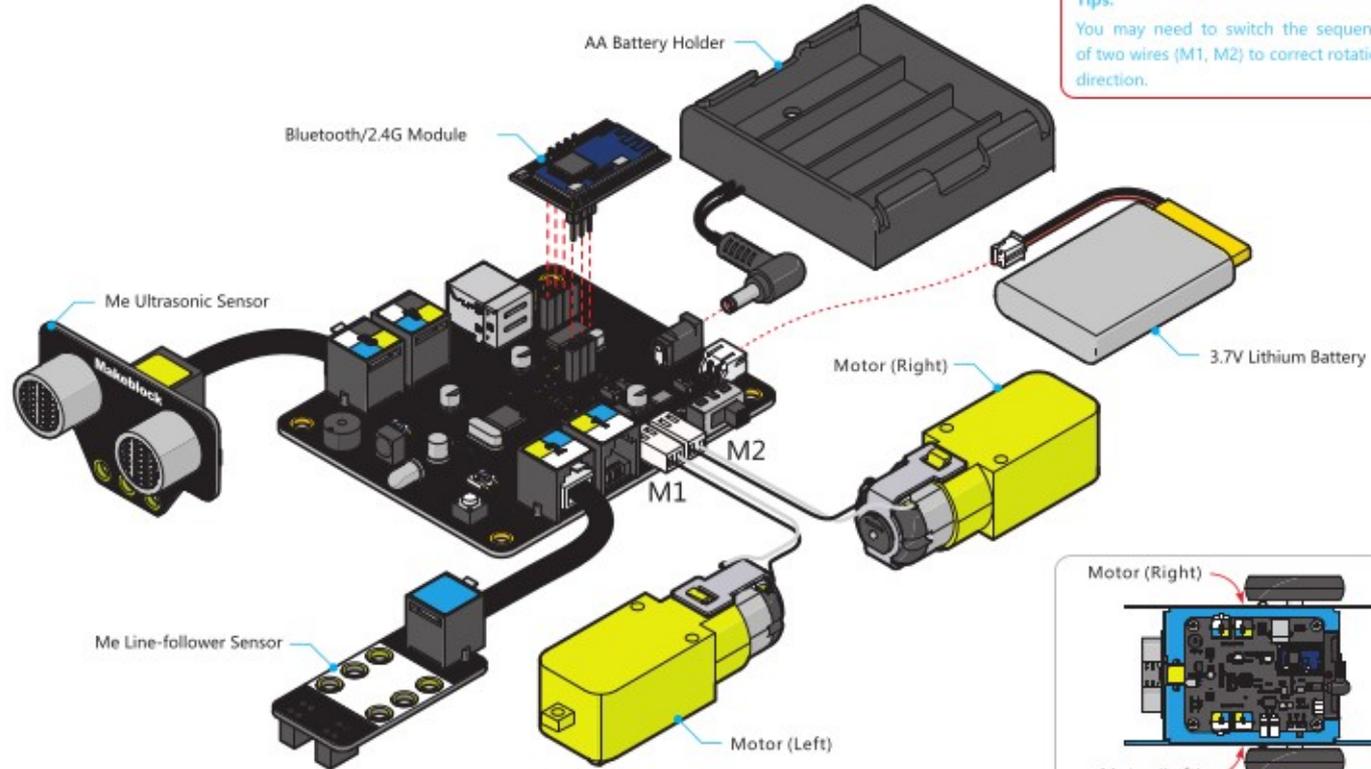


Please wire the electronic modules after assembly.



# Verifica delle connessioni

## Wiring Instructions



# Programma n.4

Far muovere mBot

# Programma n.4

Demo!?

# Prudenza!

mBot comincerà a muoversi  
non appena il trasferimento del  
programma sarà completato.

# Prudenza!

mBot comincerà a muoversi  
non appena il trasferimento del  
programma sarà completato.

Attenzione a non farsi  
prendere di sorpresa!

# Prudenza!

mBot continuerà a muoversi  
finché non lo spegneremo  
(o si esauriranno le batterie).

# Programma 4 bis

Far muovere mBot...

# Programma 4 bis

Far muovere mBot...

- quando si preme il pulsante

# Programma 4 bis

Far muovere mBot...

- quando si preme il pulsante
- per tre secondi, dopodiché si ferma

# Programma 4 bis

The image shows a sequence of code blocks for a mBot program. The blocks are as follows:

- when mBot(mcore) starts up** (Yellow block)
- forever** (Orange loop block)
- wait until** (Orange block) with a light blue trigger block: **when on-board button pressed** (dropdown) and **?** (dropdown)
- move forward** (Blue block) with a dropdown arrow and **at power 100 %** (input field)
- wait 3 seconds** (Orange block)
- stop moving** (Blue block)
- A small white arrow icon at the bottom of the loop block.

# Esercitazione

Muoversi a caso cambiando direzione ogni secondo...

# Esercitazione

Muoversi a caso cambiando direzione ogni secondo, usando i LED come indicatori:

entrambi verdi per "avanti"

entrambi rossi per "indietro"

giallo (solo il LED interno) per "gira".

# Programma n.5

Scansare gli ostacoli

# Sensore di prossimità



# Applicazione

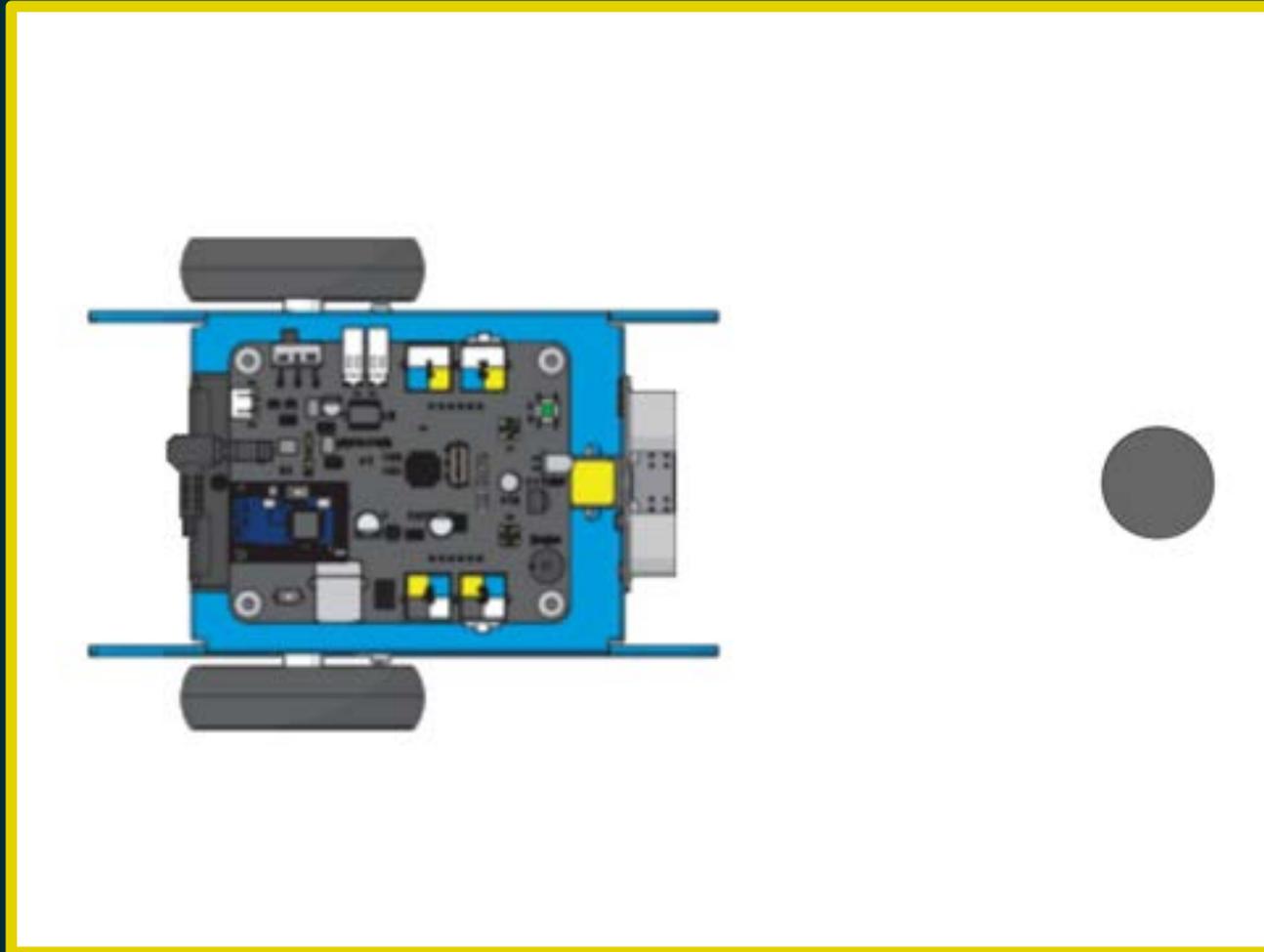


By Powerrethdd - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=15009646>

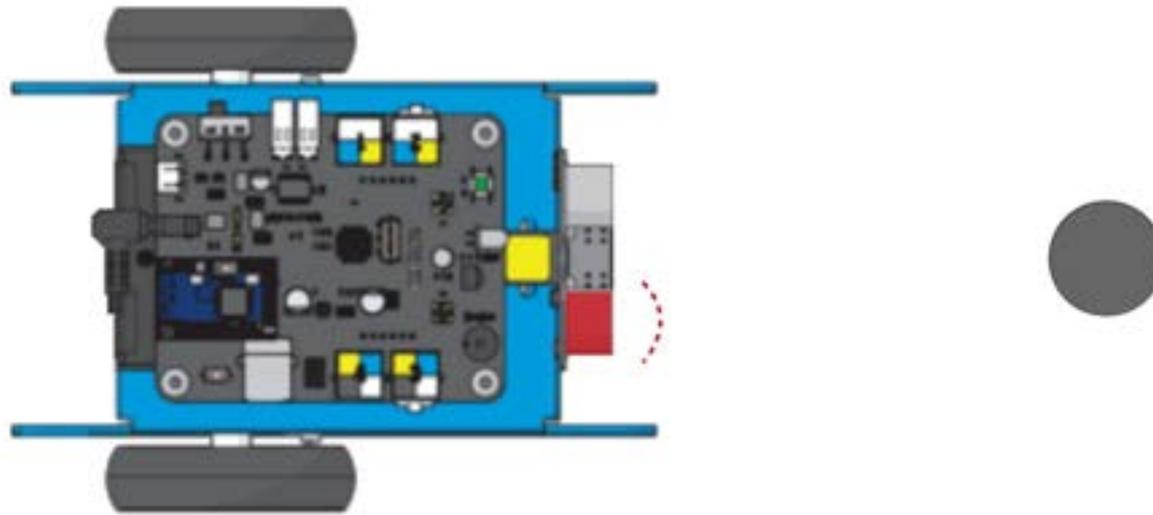
# Applicazione



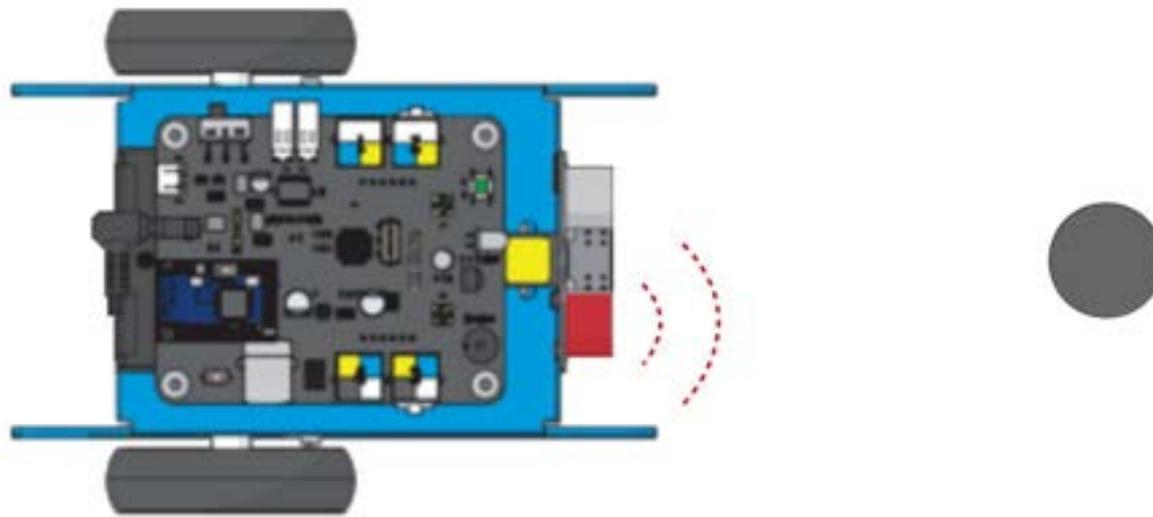
# Principio di funzionamento



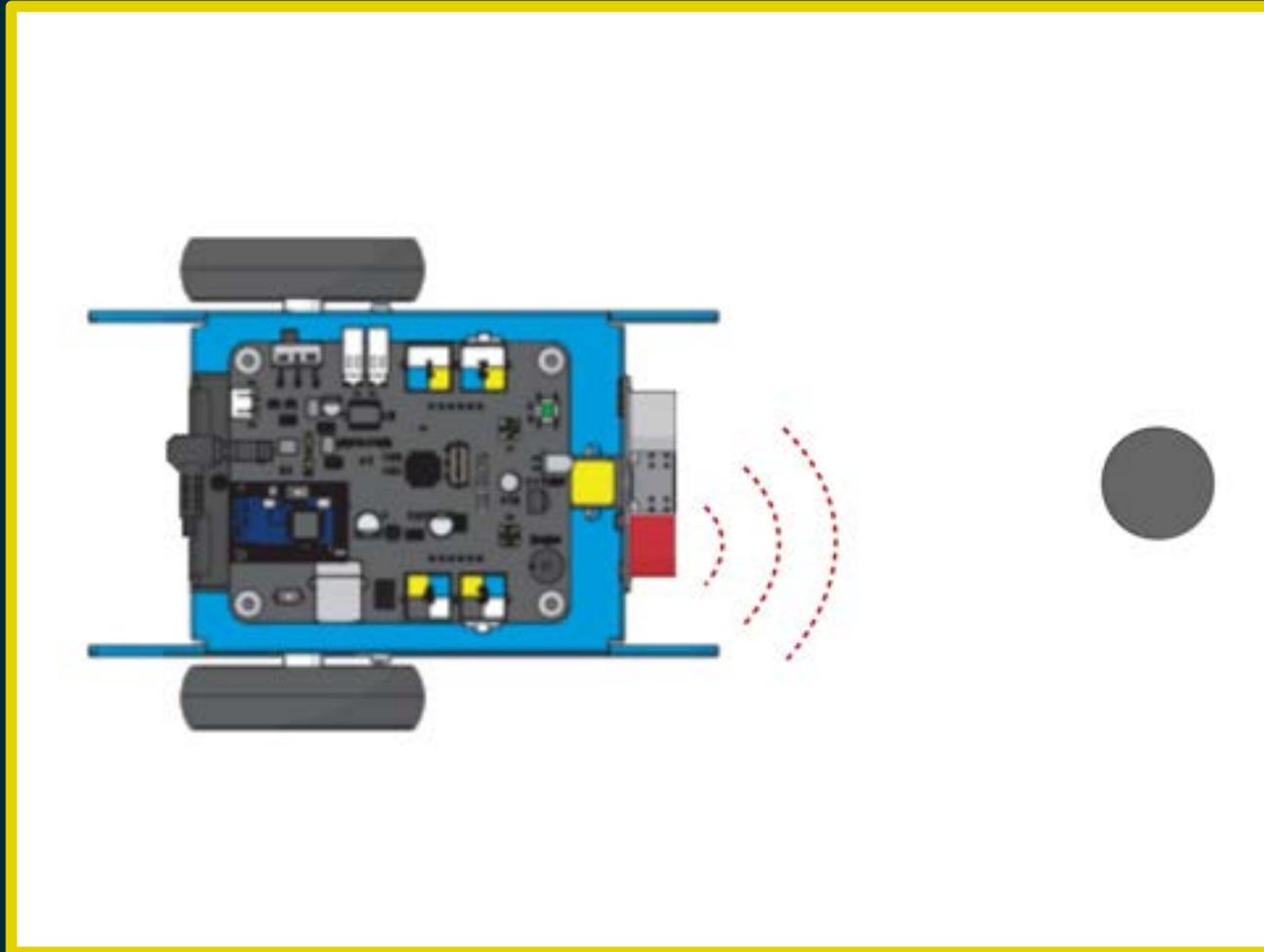
# Principio di funzionamento



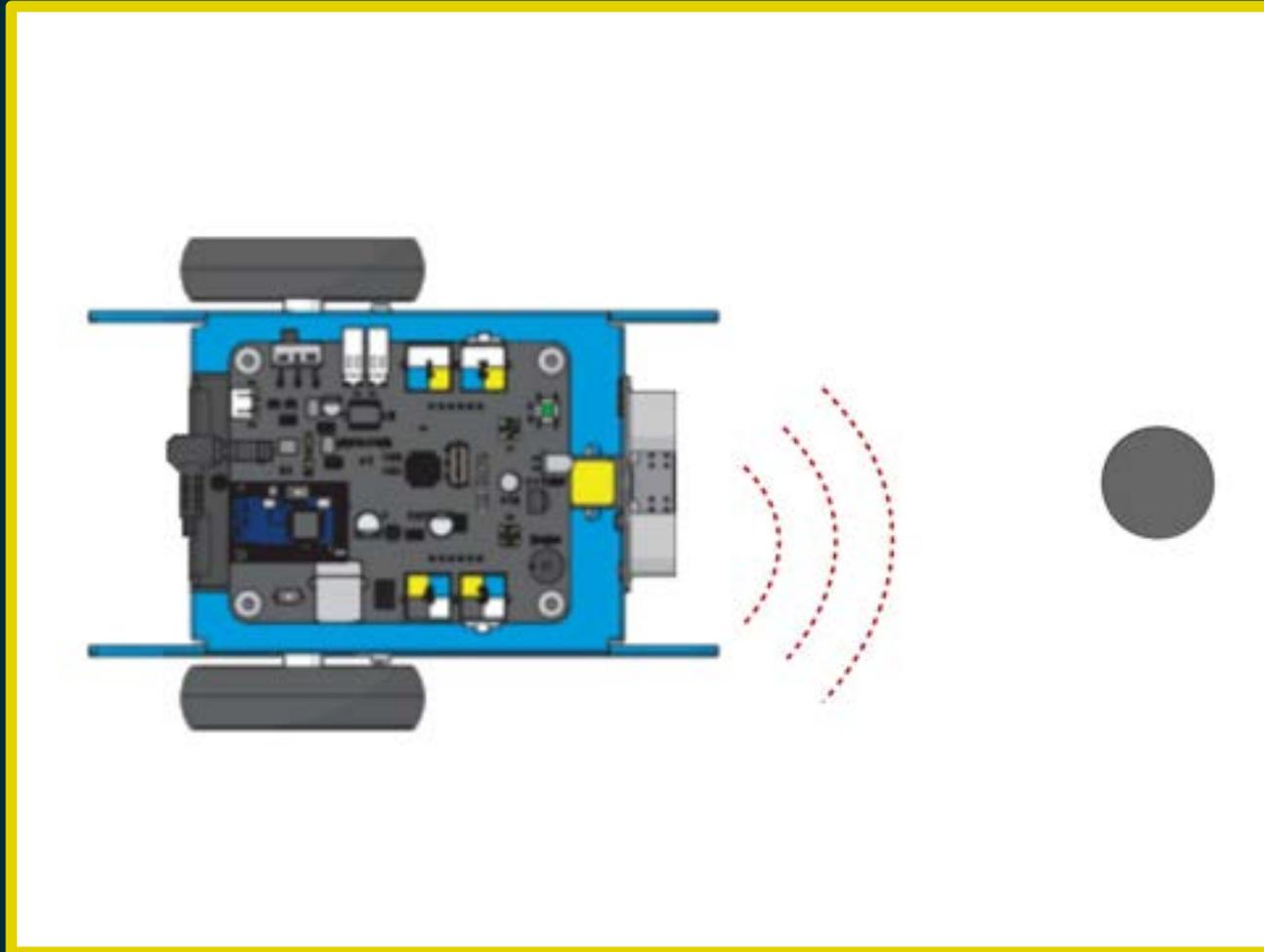
# Principio di funzionamento



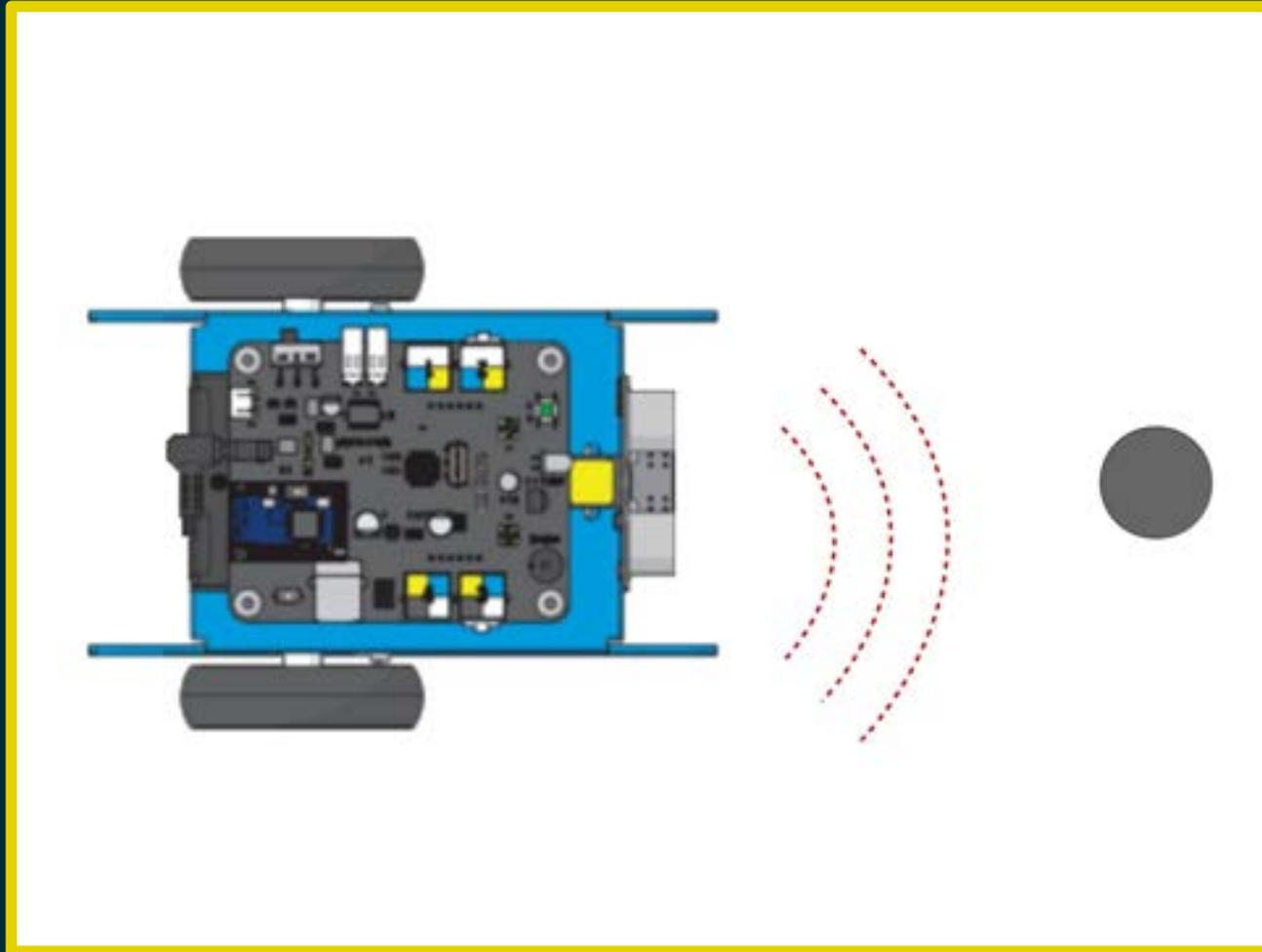
# Principio di funzionamento



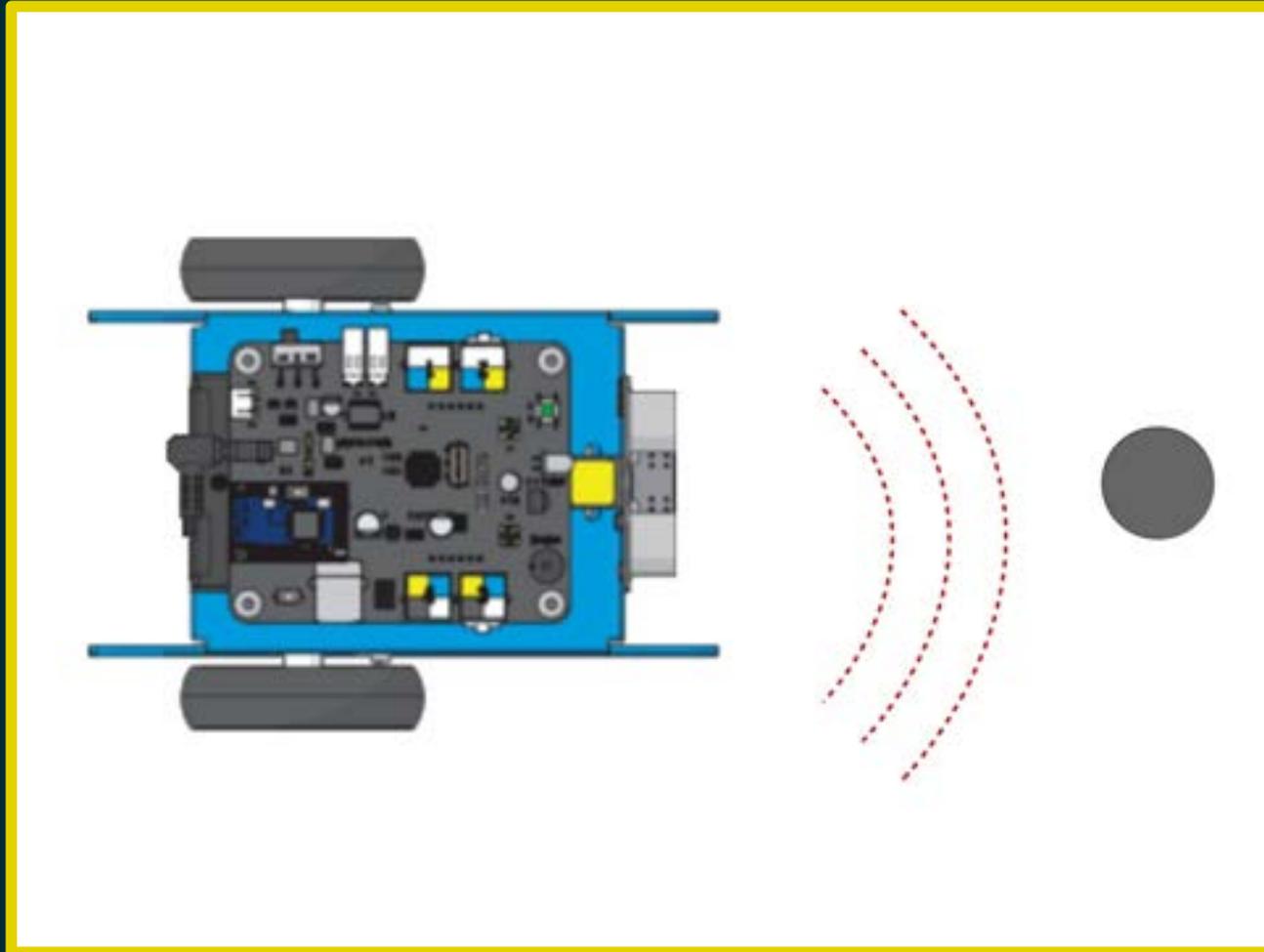
# Principio di funzionamento



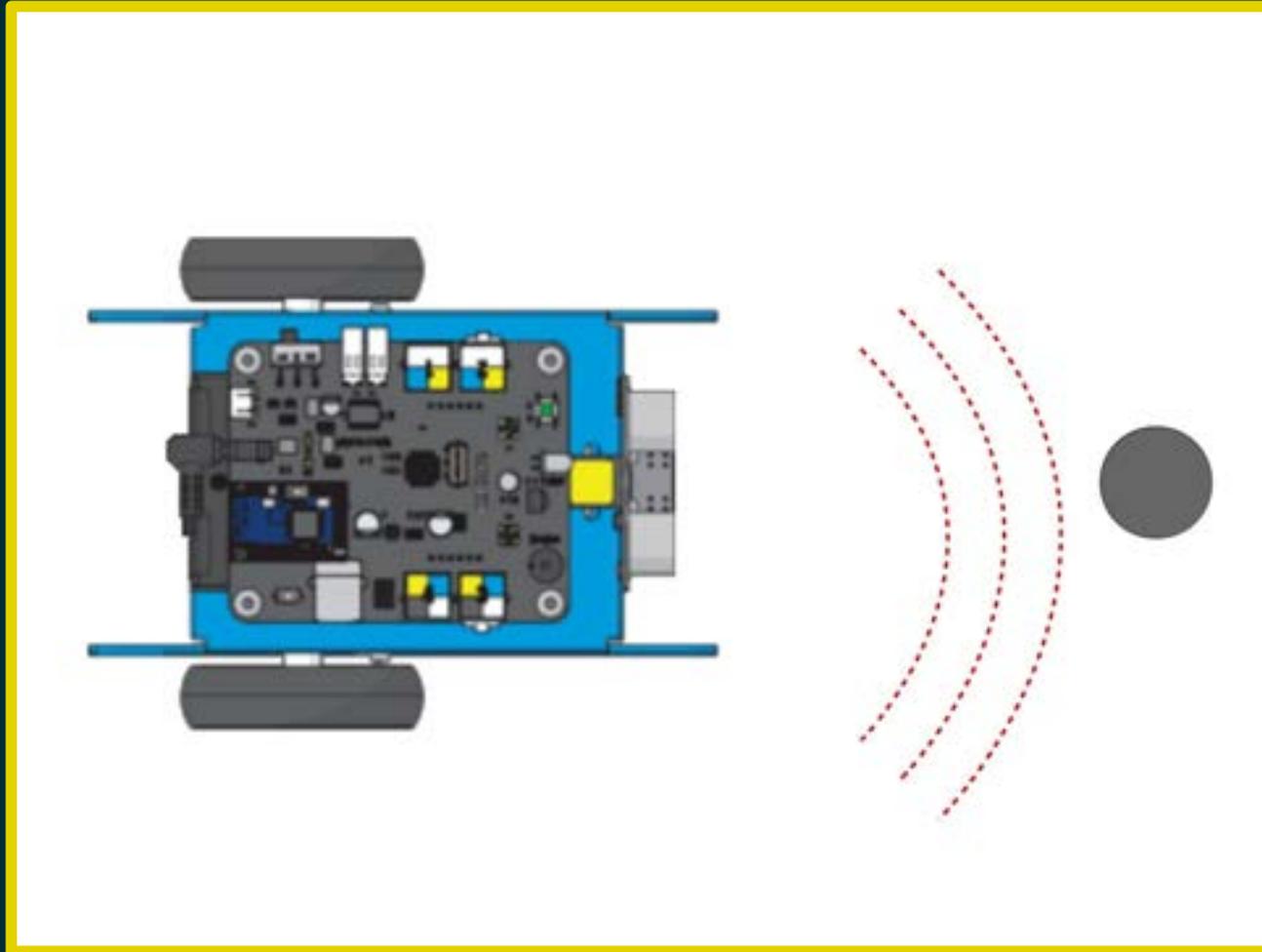
# Principio di funzionamento



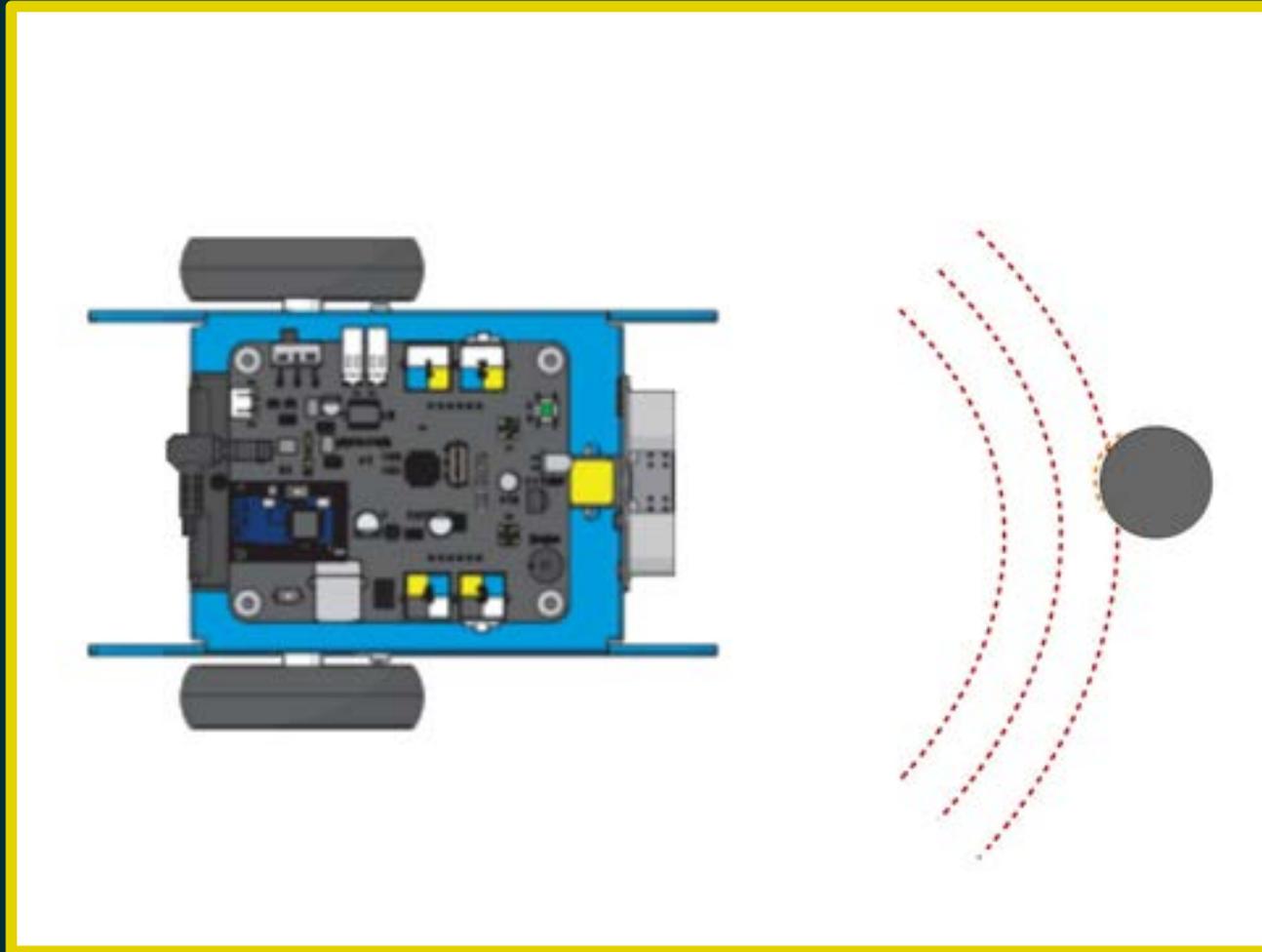
# Principio di funzionamento



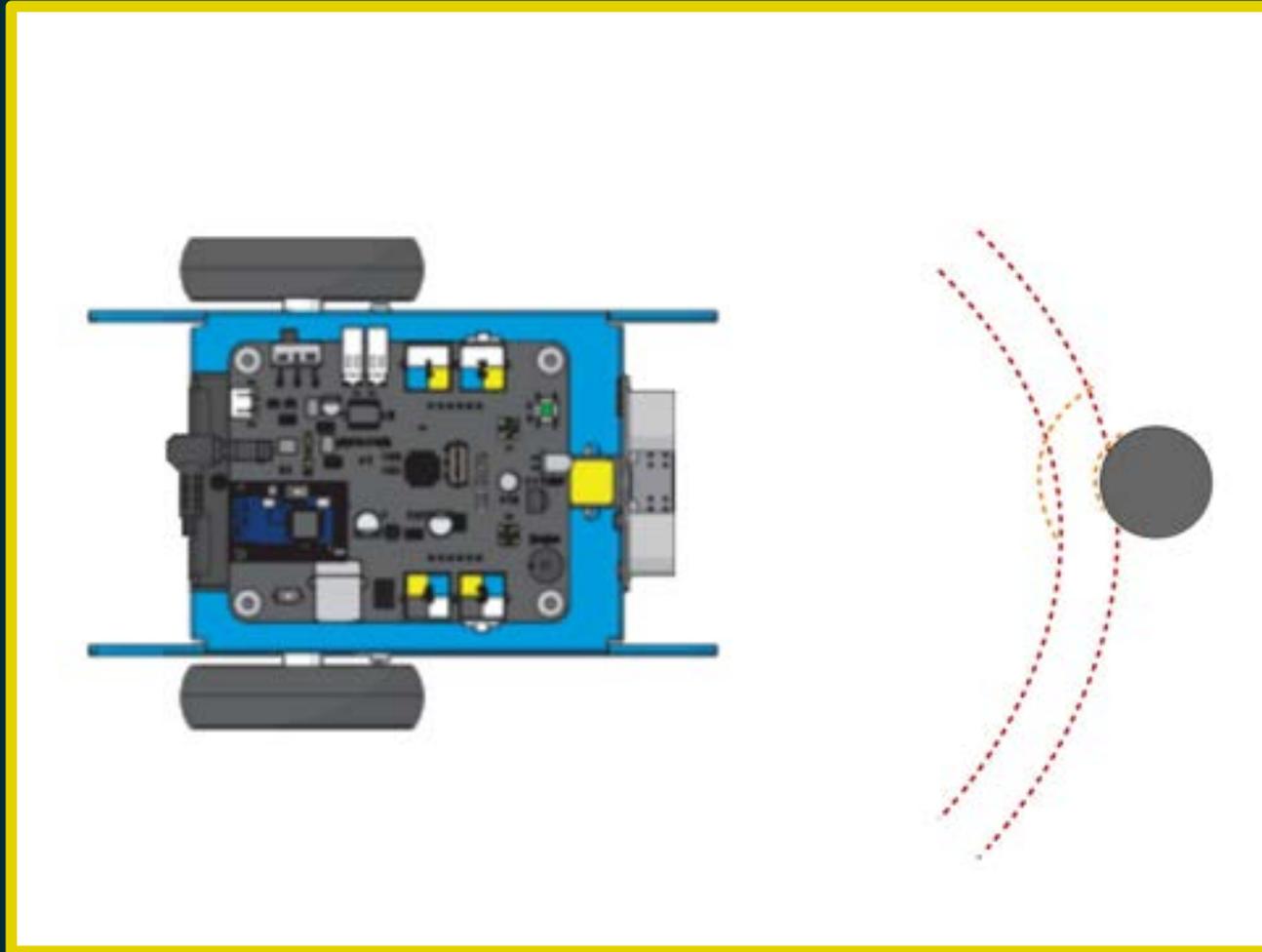
# Principio di funzionamento



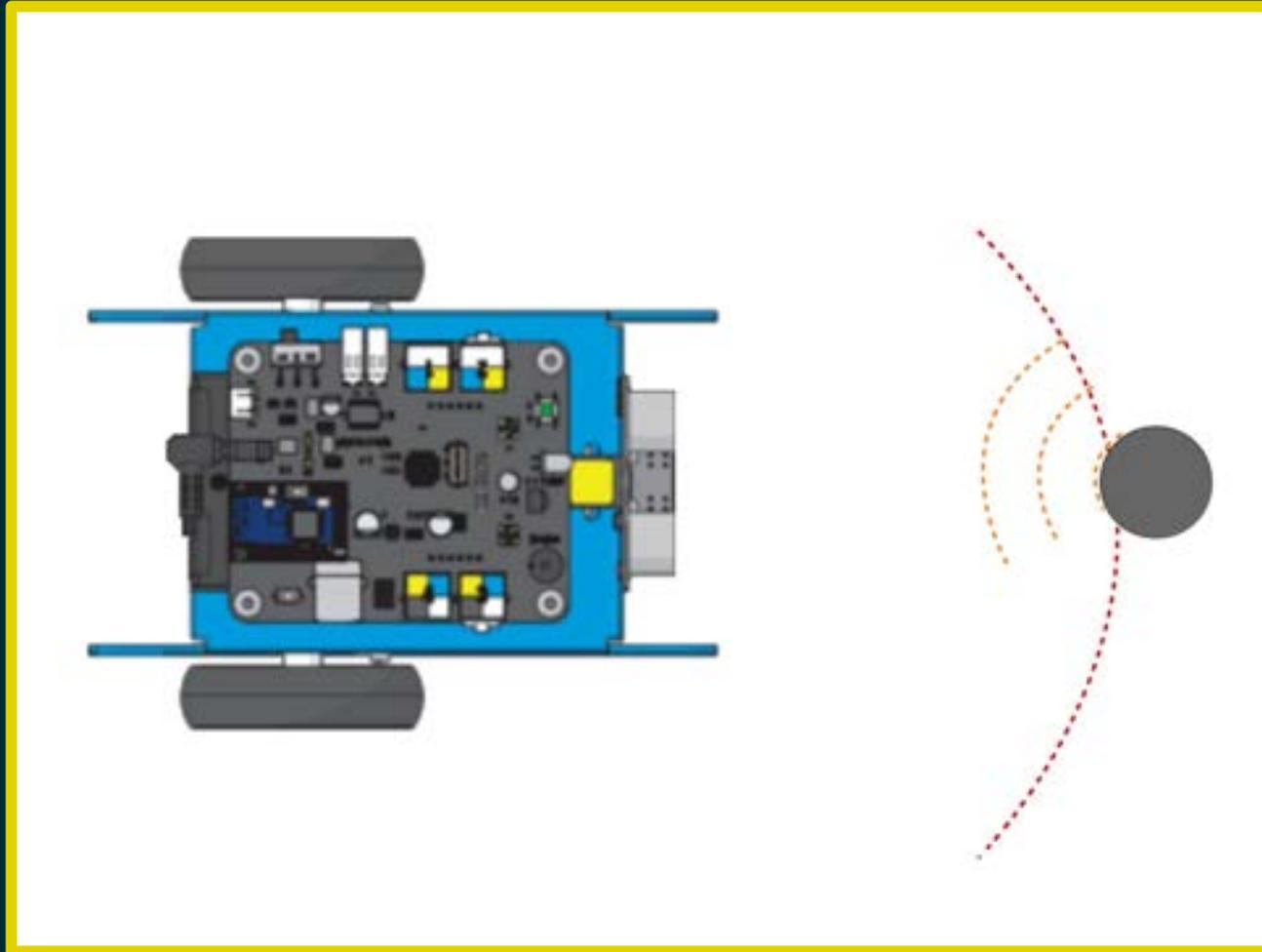
# Principio di funzionamento



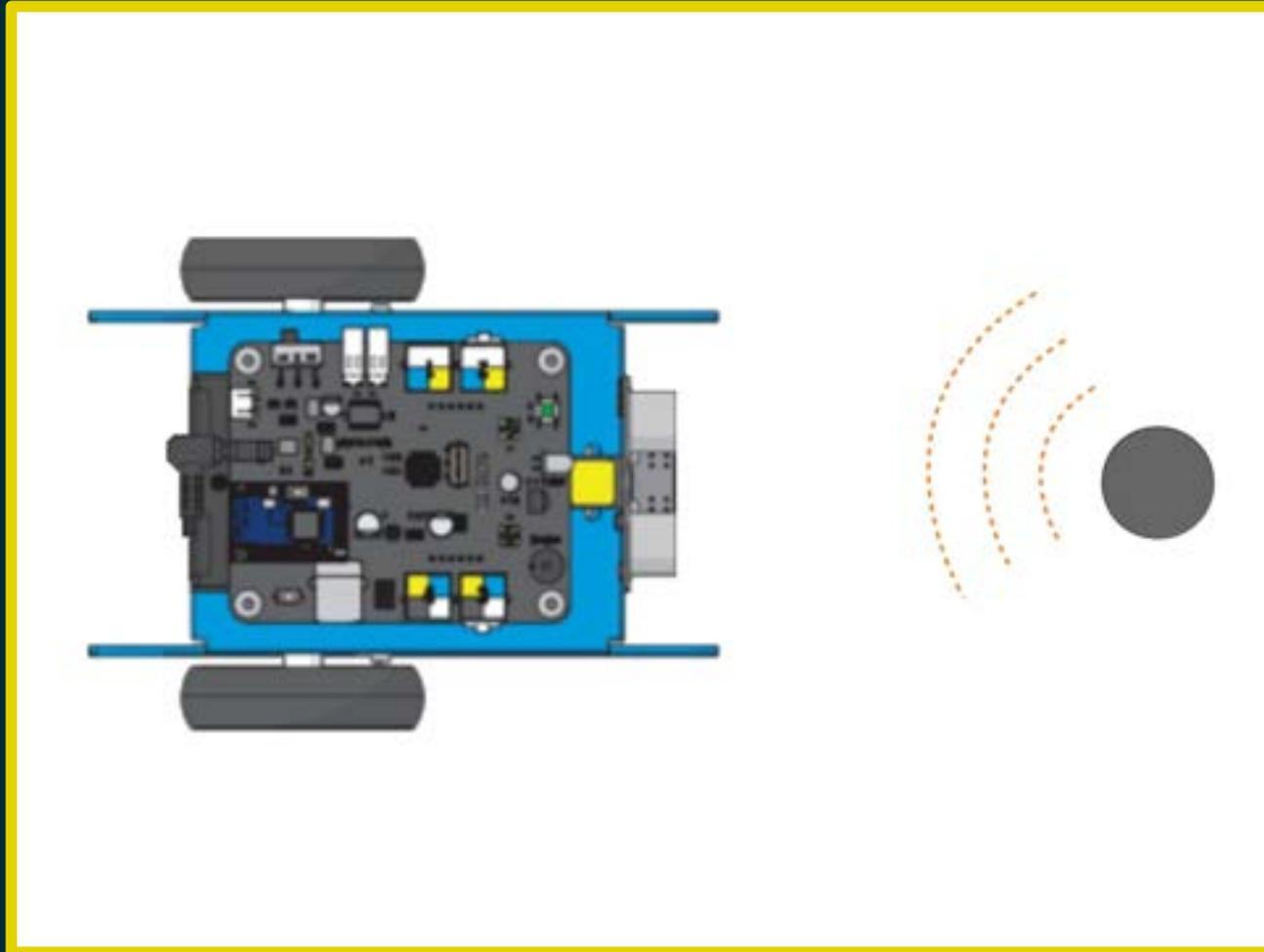
# Principio di funzionamento



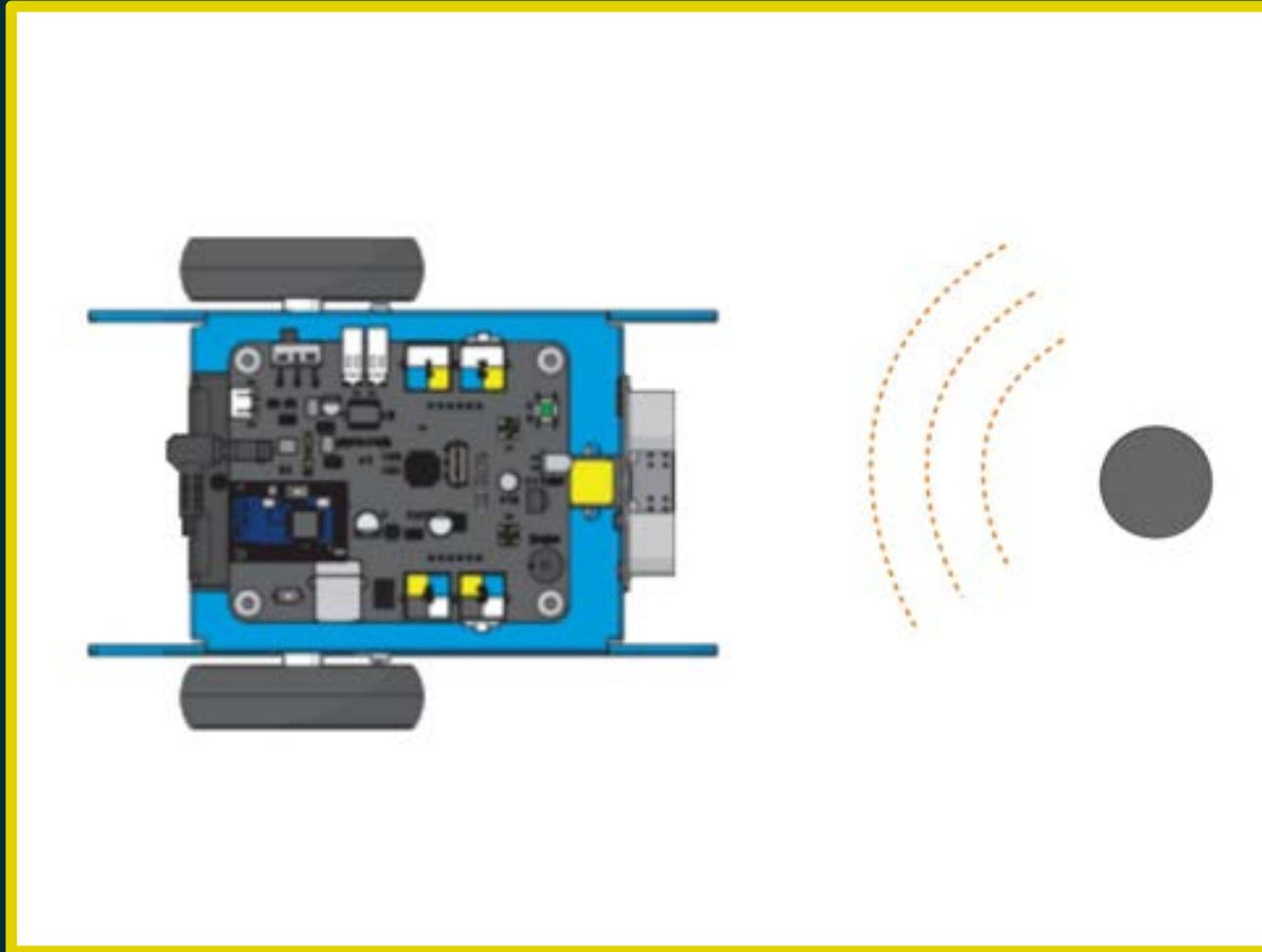
# Principio di funzionamento



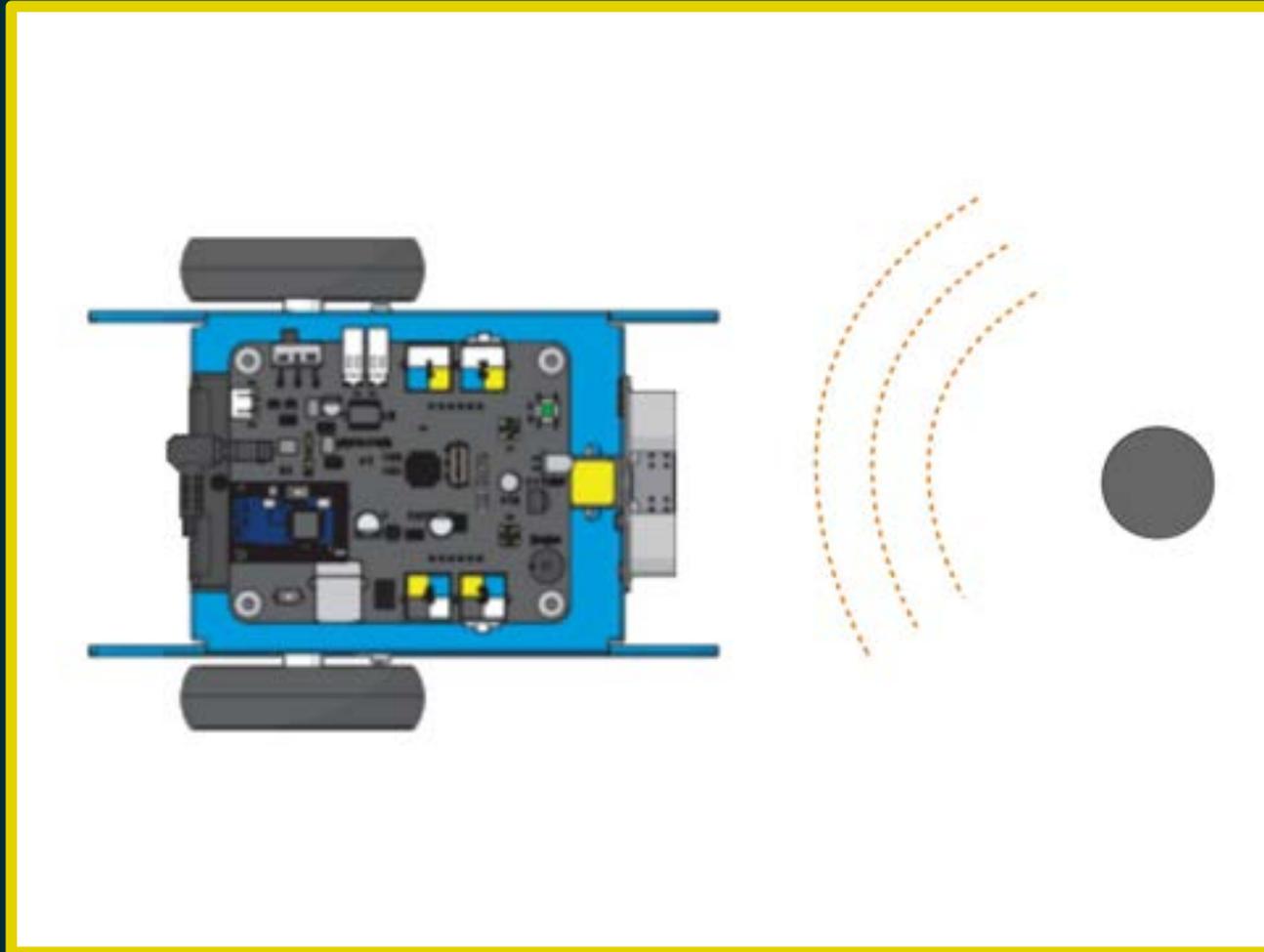
# Principio di funzionamento



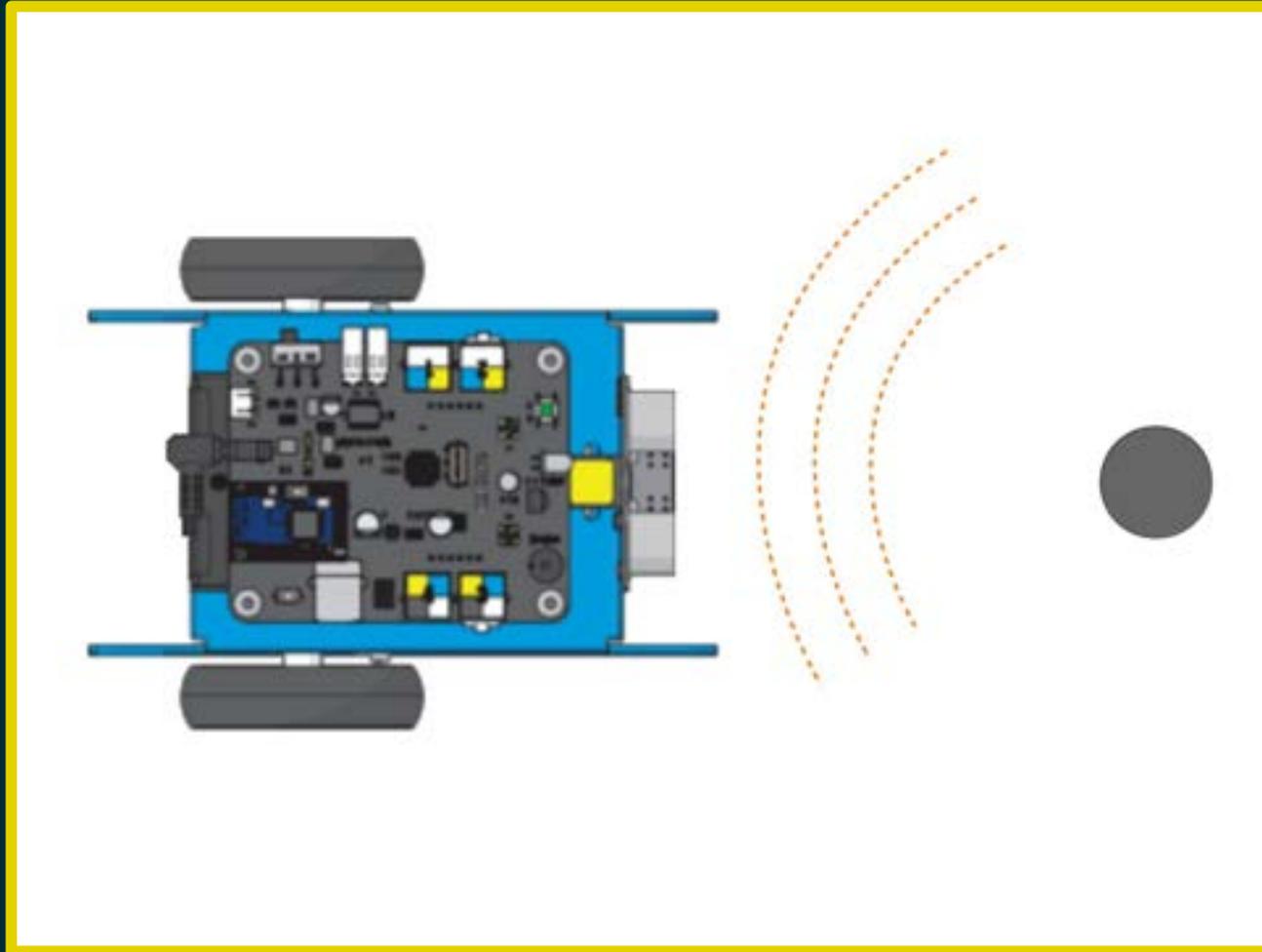
# Principio di funzionamento



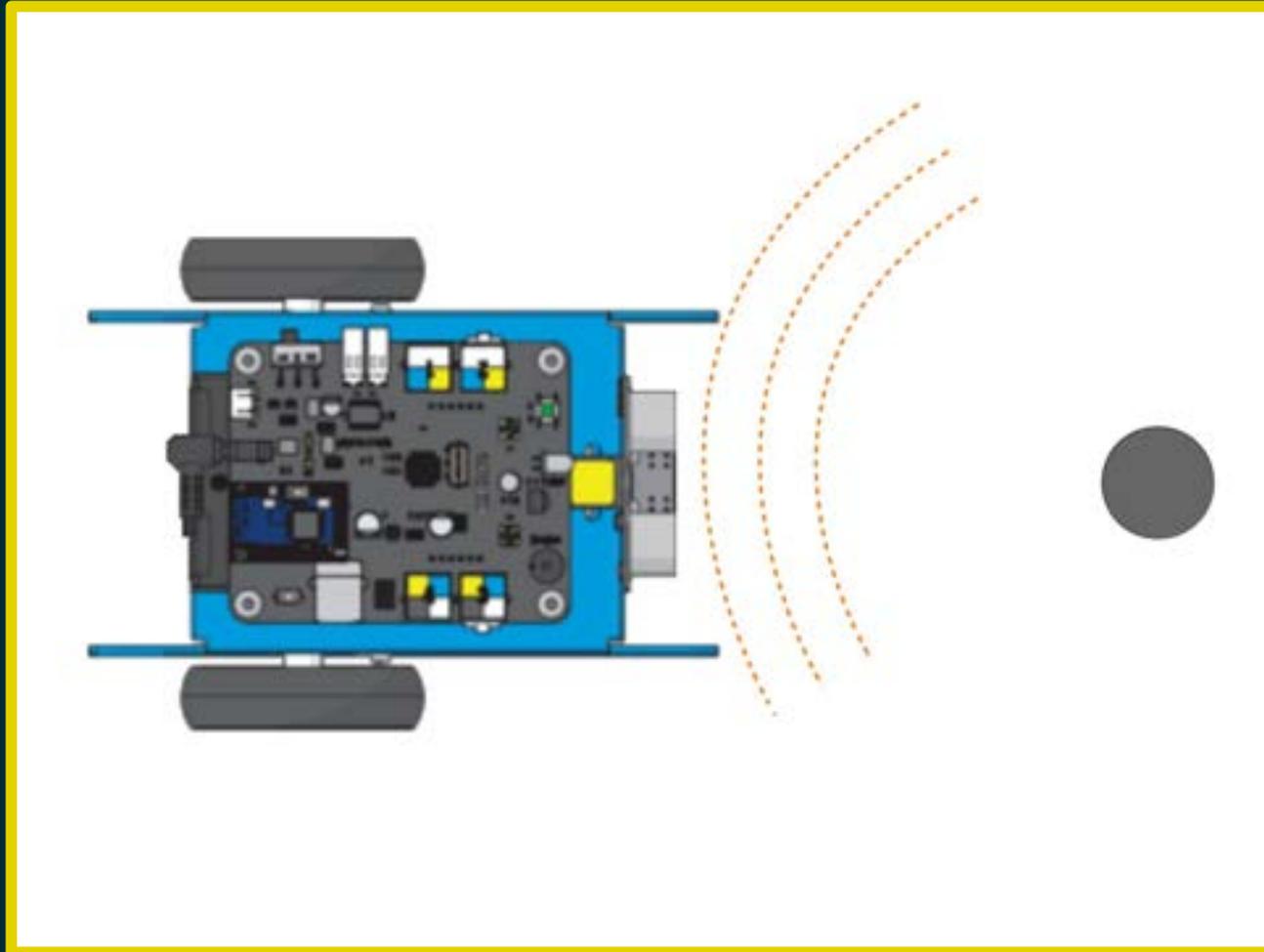
# Principio di funzionamento



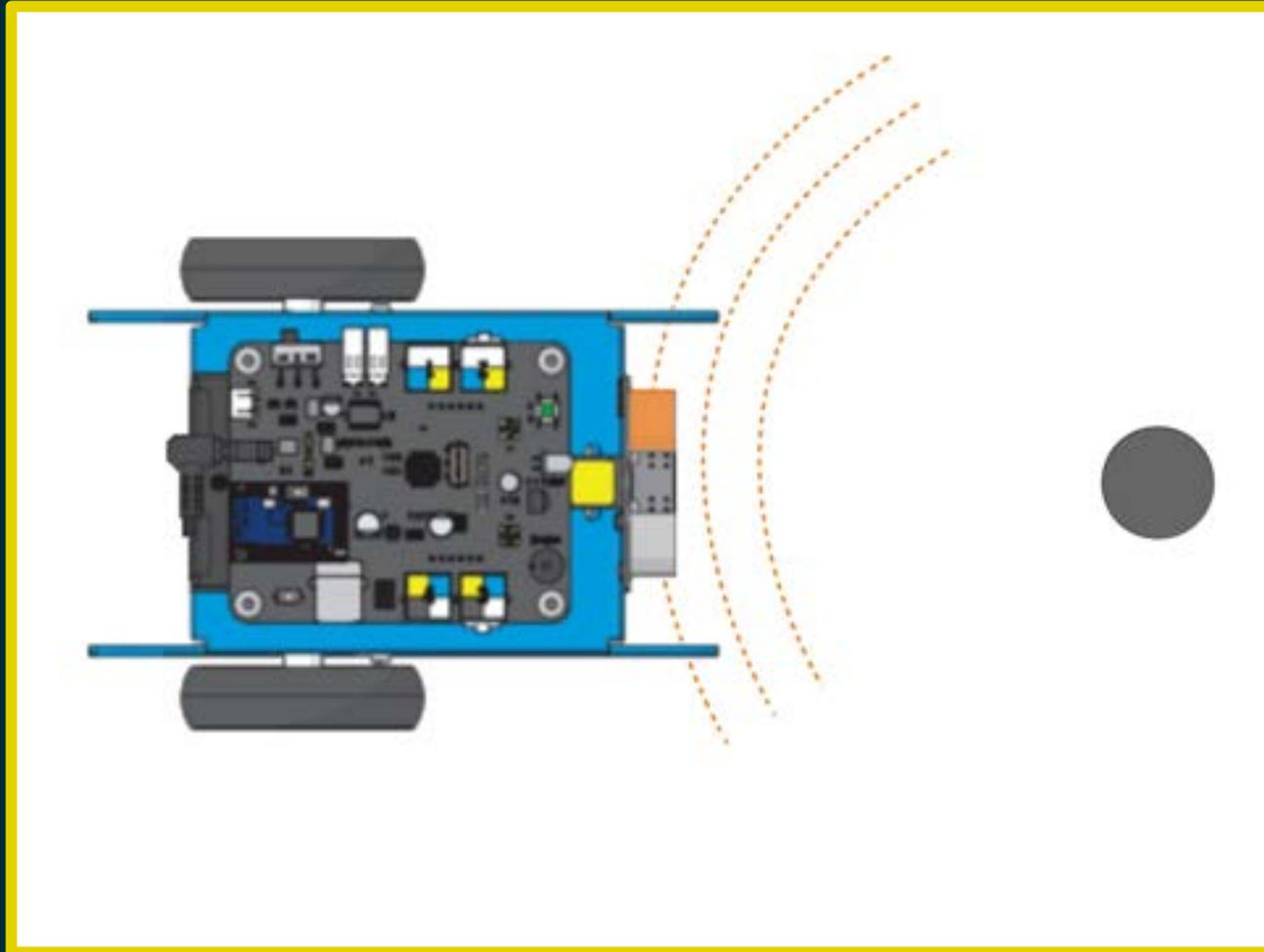
# Principio di funzionamento



# Principio di funzionamento



# Principio di funzionamento



# Caratteristiche del sensore

# Caratteristiche del sensore

Campo di misura: da 3 a 400cm

# Caratteristiche del sensore

Campo di misura: da 3 a 400cm

Risoluzione: 1 cm

# Caratteristiche del sensore

Campo di misura: da 3 a 400cm

Risoluzione: 1 cm

Non particolarmente stabile.

# Caratteristiche del sensore

Campo di misura: da 3 a 400cm

Risoluzione: 1 cm

Non particolarmente stabile.

Soggetto a interferenze: attendere almeno 50ms tra una lettura e l'altra.

Test del sensore /1

# Test del sensore /1

Indicare la distanza con il colore dei LED.

# Test del sensore /1

Indicare la distanza con il colore dei LED.

meno di 15 cm: LED rossi

tra 15 e 45 cm: LED gialli

oltre 45 cm: LED verdi

# Test del sensore /2

Indicare la distanza con il cicalino: più l'oggetto è vicino, più ravvicinati i "beep".

# Test del sensore /2

Indicare la distanza con il cicalino: più l'oggetto è vicino, più ravvicinati i "beep".

tra 100 e 75 cm: 1 beep al secondo

tra 75 e 50 cm: 2 beep al secondo

tra 50 e 25 cm: 4 beep al secondo

meno di 25 cm: beep continuo

Scansa ostacoli

# Scansa ostacoli

Leggiamo il valore registrato dal sensore.

# Scansa ostacoli

Leggiamo il valore registrato dal sensore.  
Se l'ostacolo si trova a più di 40cm...

# Scansa ostacoli

Leggiamo il valore registrato dal sensore.  
Se l'ostacolo si trova a più di 40cm  
allora proseguiamo dritti;

# Scansa ostacoli

Leggiamo il valore registrato dal sensore.

Se l'ostacolo si trova a più di 40cm

allora proseguiamo dritti;

altrimenti...

# Scansa ostacoli

Leggiamo il valore registrato dal sensore.

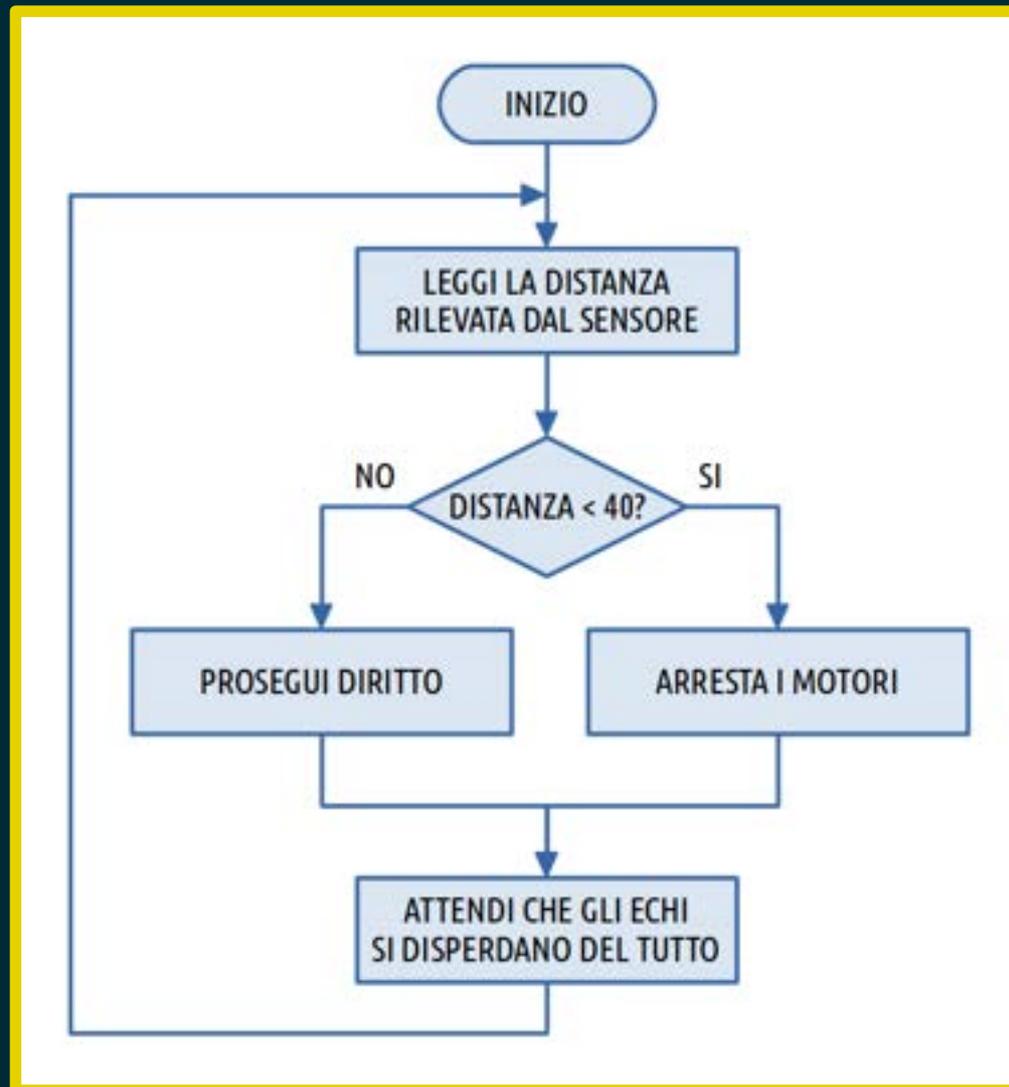
Se l'ostacolo si trova a più di 40cm

allora proseguiamo dritti;

altrimenti

ci fermiamo.

# Diagramma di flusso



# Scansa ostacoli

Accendere i LED:

di giallo durante l'attesa iniziale,  
di verde quando il robot avanza,  
di rosso quando il robot è fermo.

# Sfida n.1

Poniamo mBot di fronte a un muro, a un metro di distanza; alla pressione del pulsante mBot avanza verso il muro; mBot deve fermarsi il più vicino possibile al muro, senza toccarlo. Vince chi va più vicino e, a parità di distanza, il più veloce.

# Scansa ostacoli /2

Leggiamo il valore registrato dal sensore.

Se l'ostacolo si trova a più di 40cm

allora proseguiamo dritti;

altrimenti...

# Scansa ostacoli /2

Leggiamo il valore registrato dal sensore.

Se l'ostacolo si trova a più di 40cm

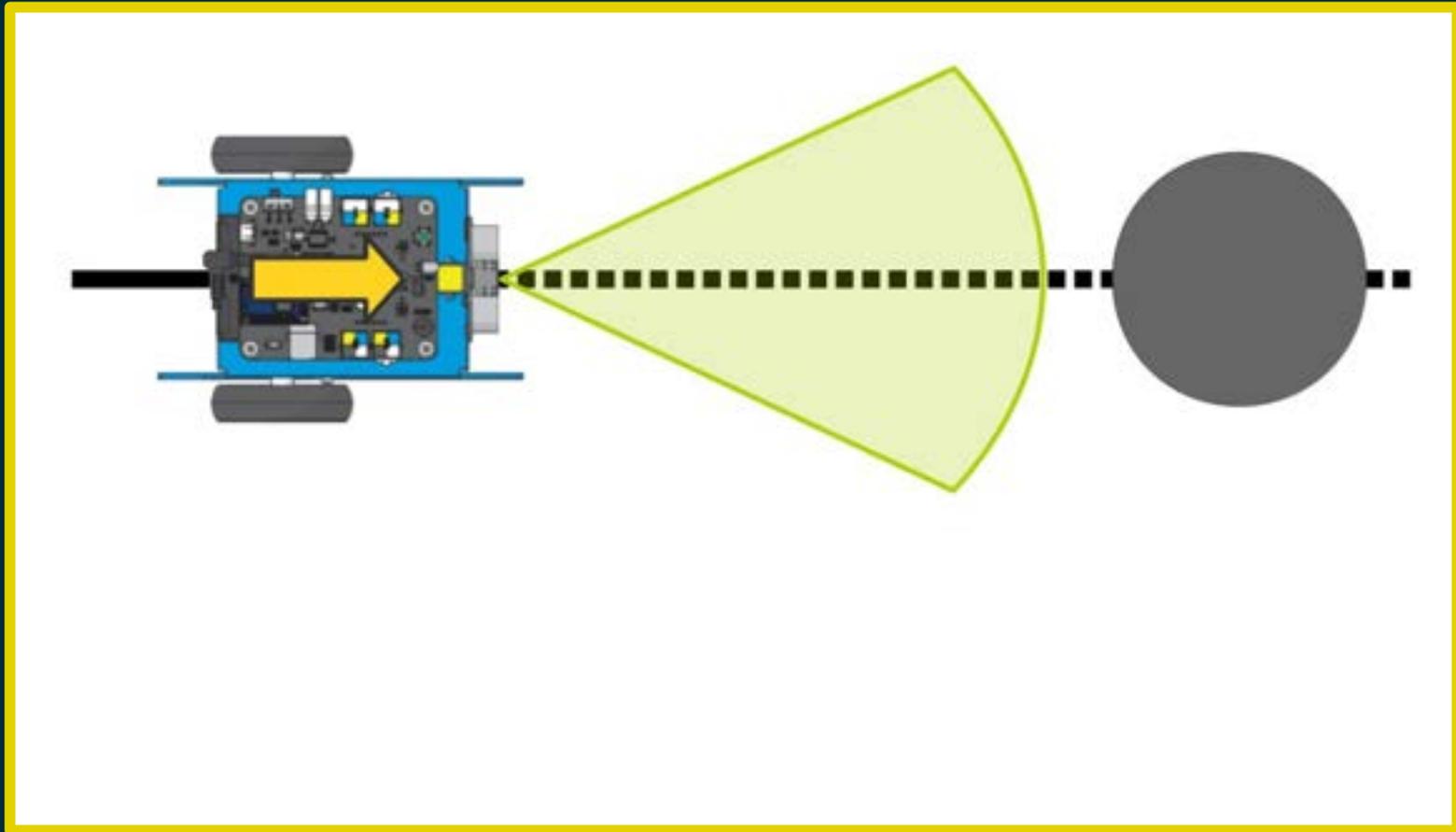
allora proseguiamo dritti;

altrimenti

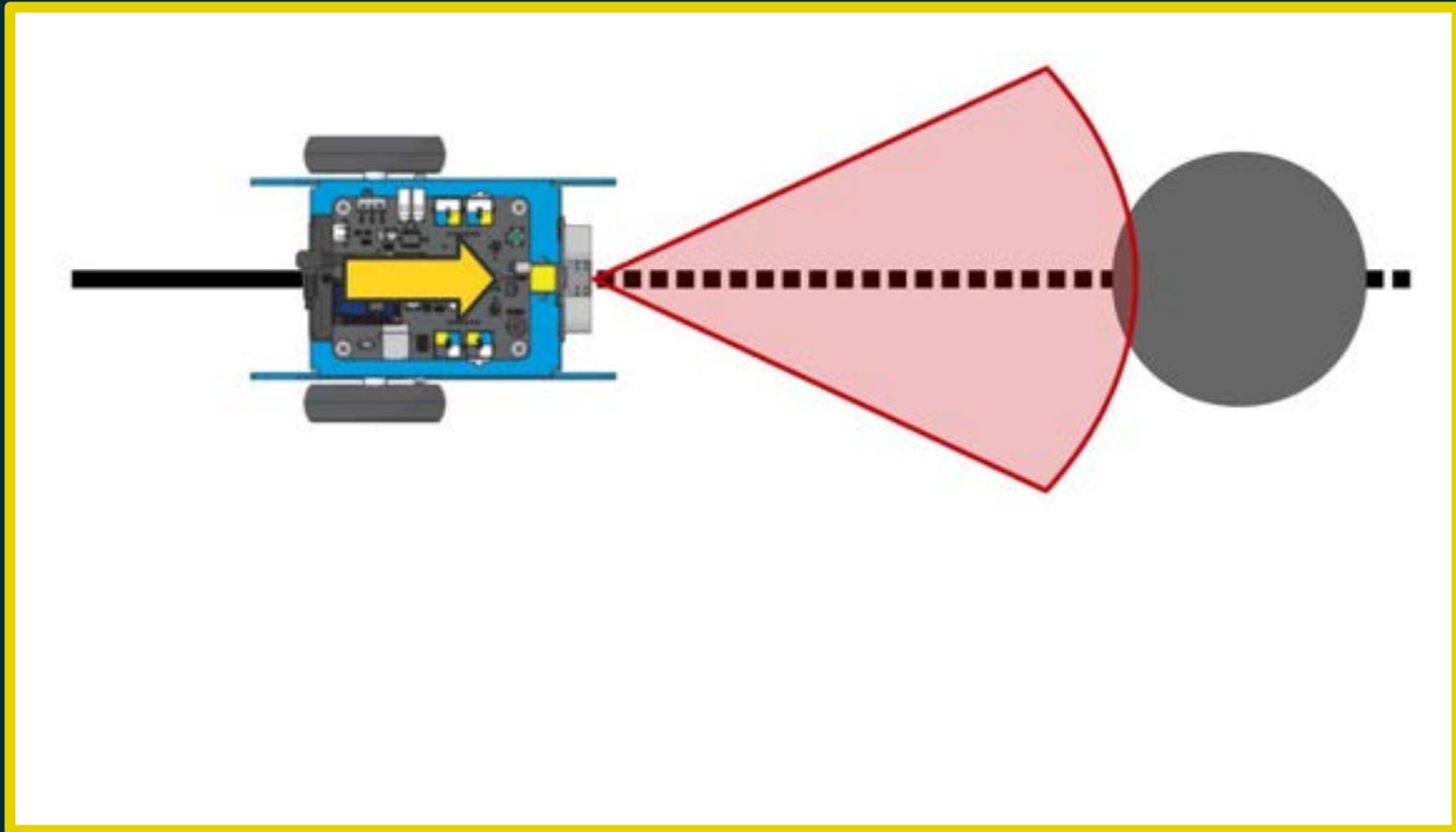
ruotiamo sul posto verso una

direzione scelta a caso.

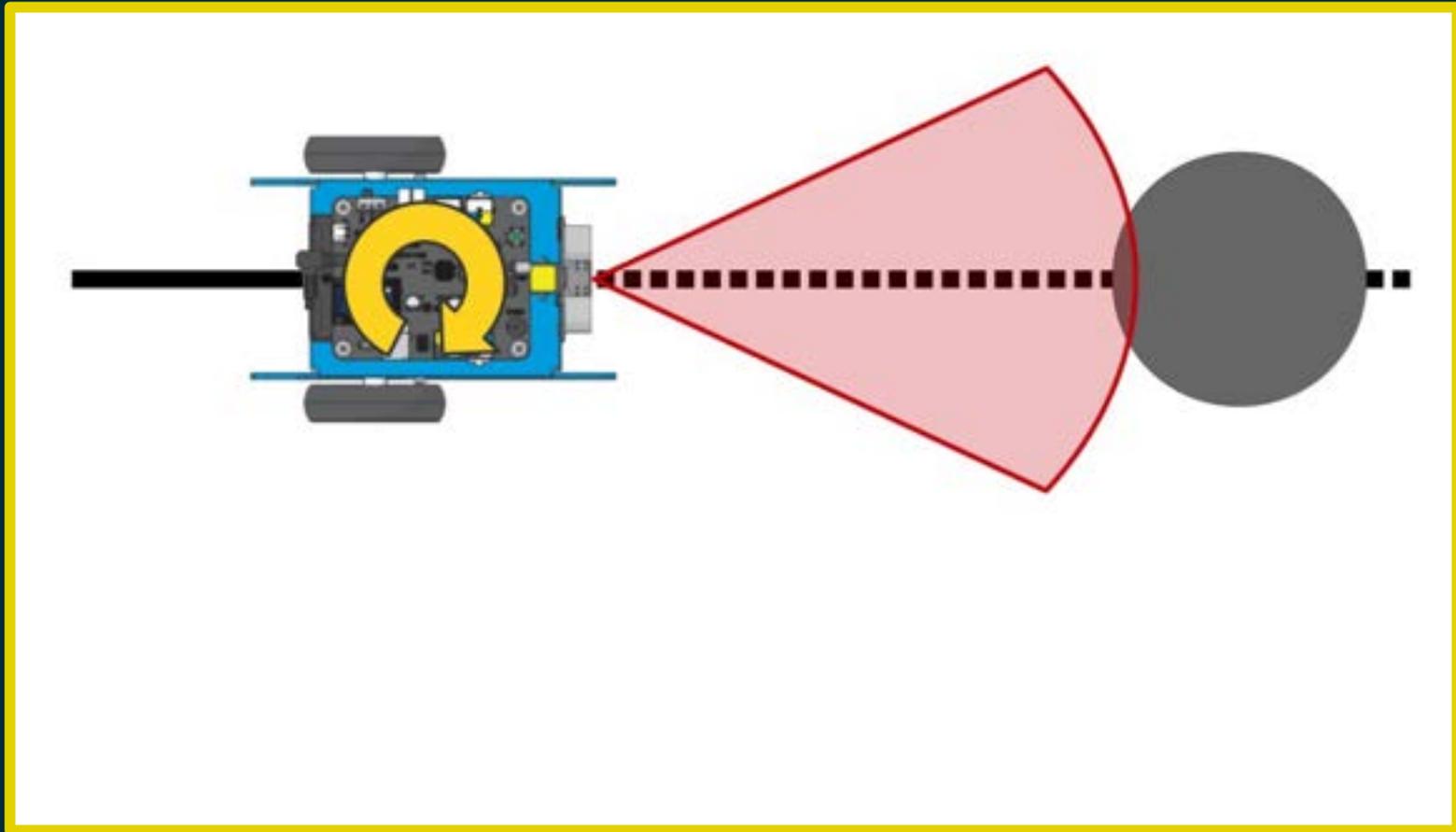
# Esemplificazione



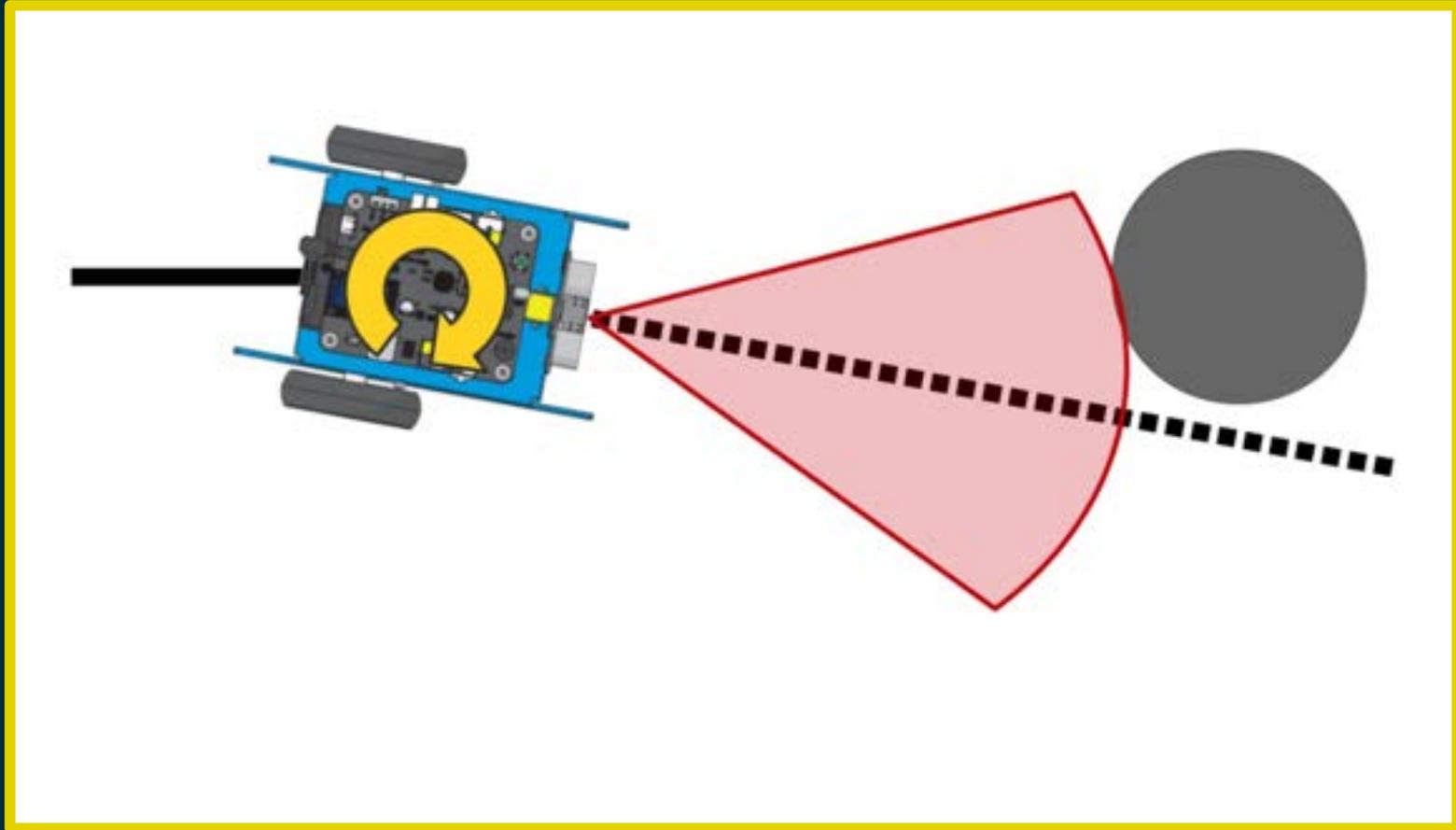
# Esemplificazione



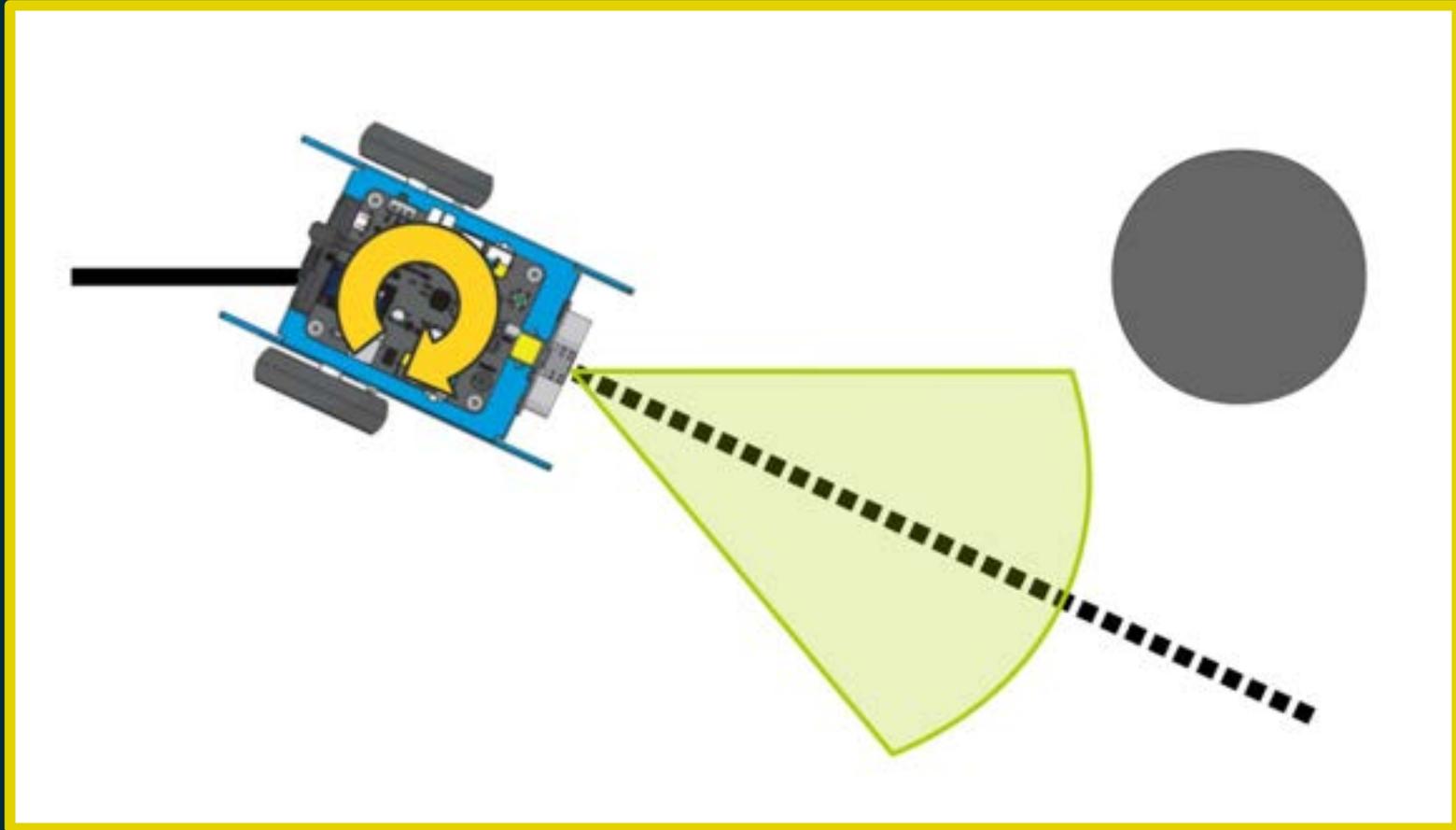
# Esemplificazione



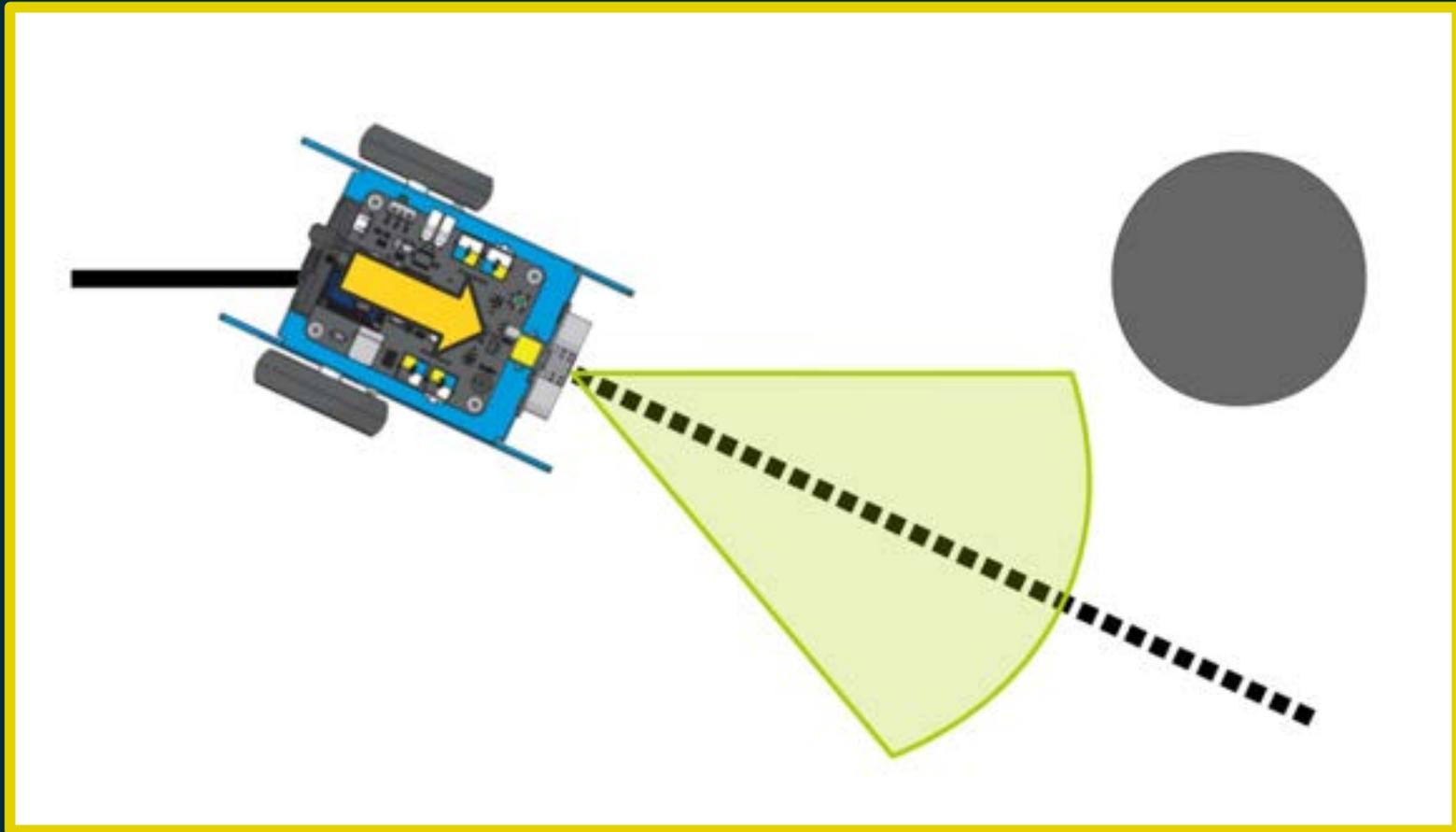
# Esemplificazione



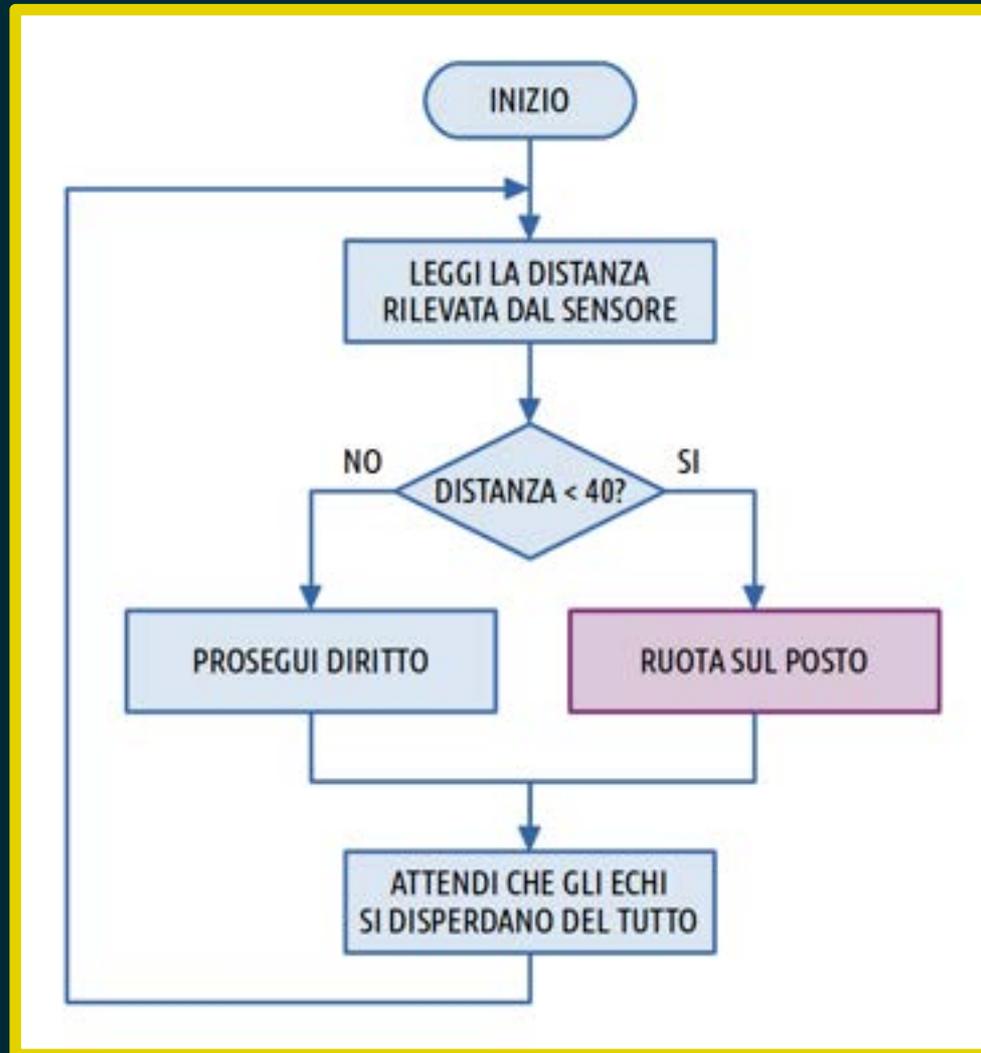
# Esemplificazione



# Esemplificazione



# Diagramma di flusso



# Scansa Ostacoli /3

Il programma funziona alla perfezione con gli ostacoli frontali immobili.

# Scansa Ostacoli /3

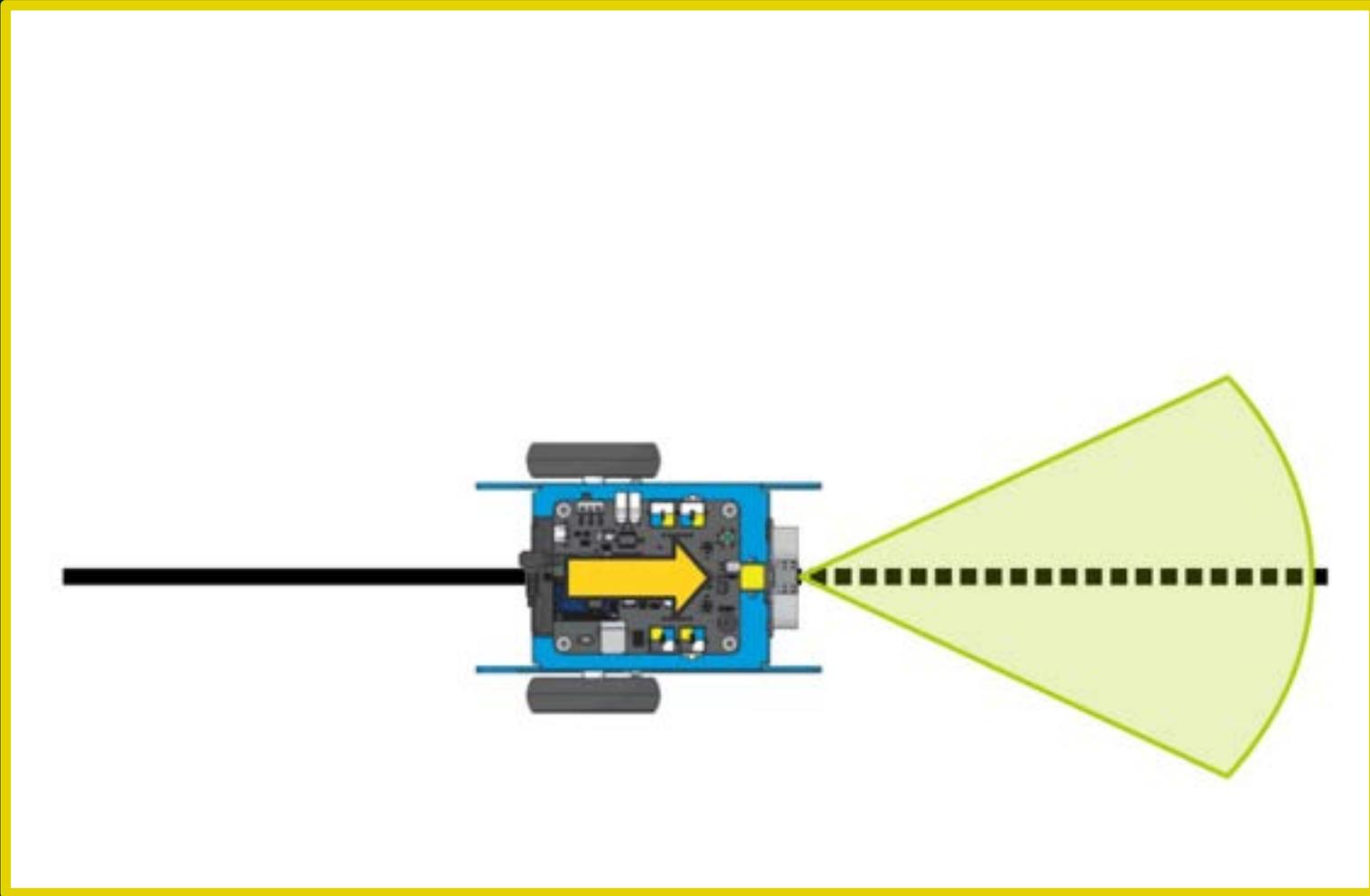
Il programma funziona alla perfezione con gli ostacoli frontali immobili. Cosa accade se un ostacolo appare all'improvviso davanti al robot?

# Scansa Ostacoli /3

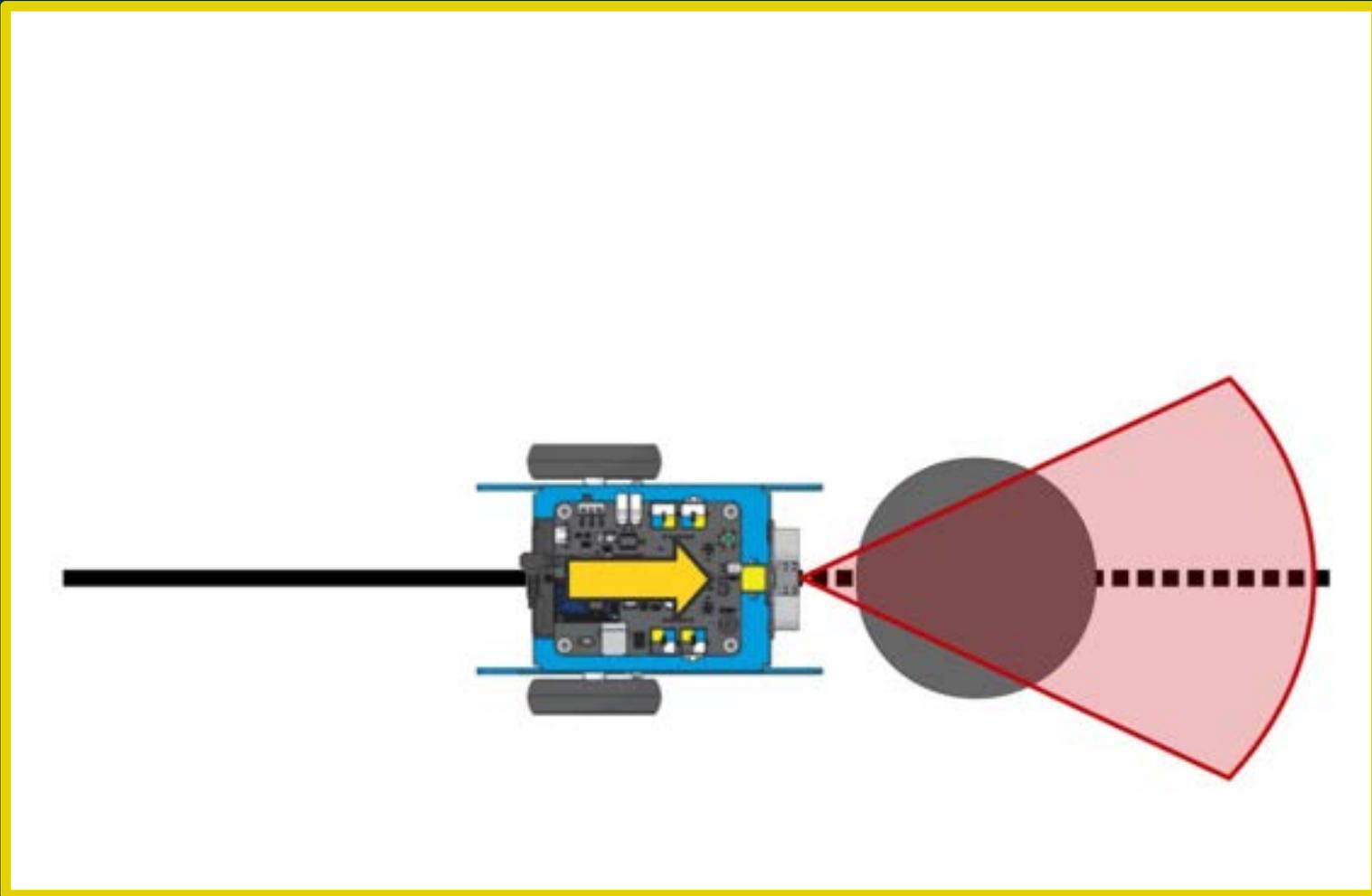
Il programma funziona alla perfezione con gli ostacoli frontali immobili. Cosa accade se un ostacolo appare all'improvviso davanti al robot?

La rotazione sul posto potrebbe non bastare ad evitare l'urto!

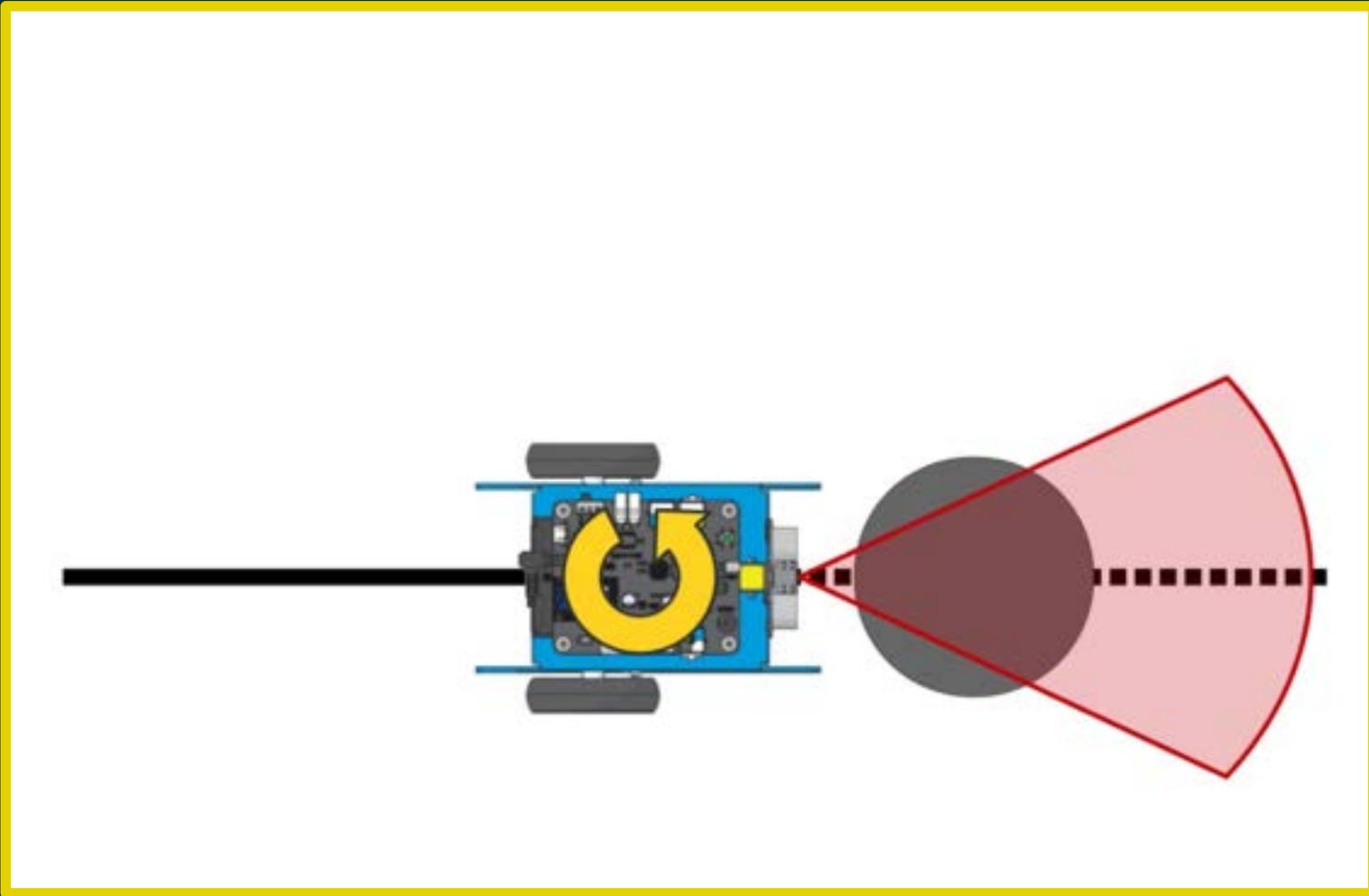
# Esemplificazione



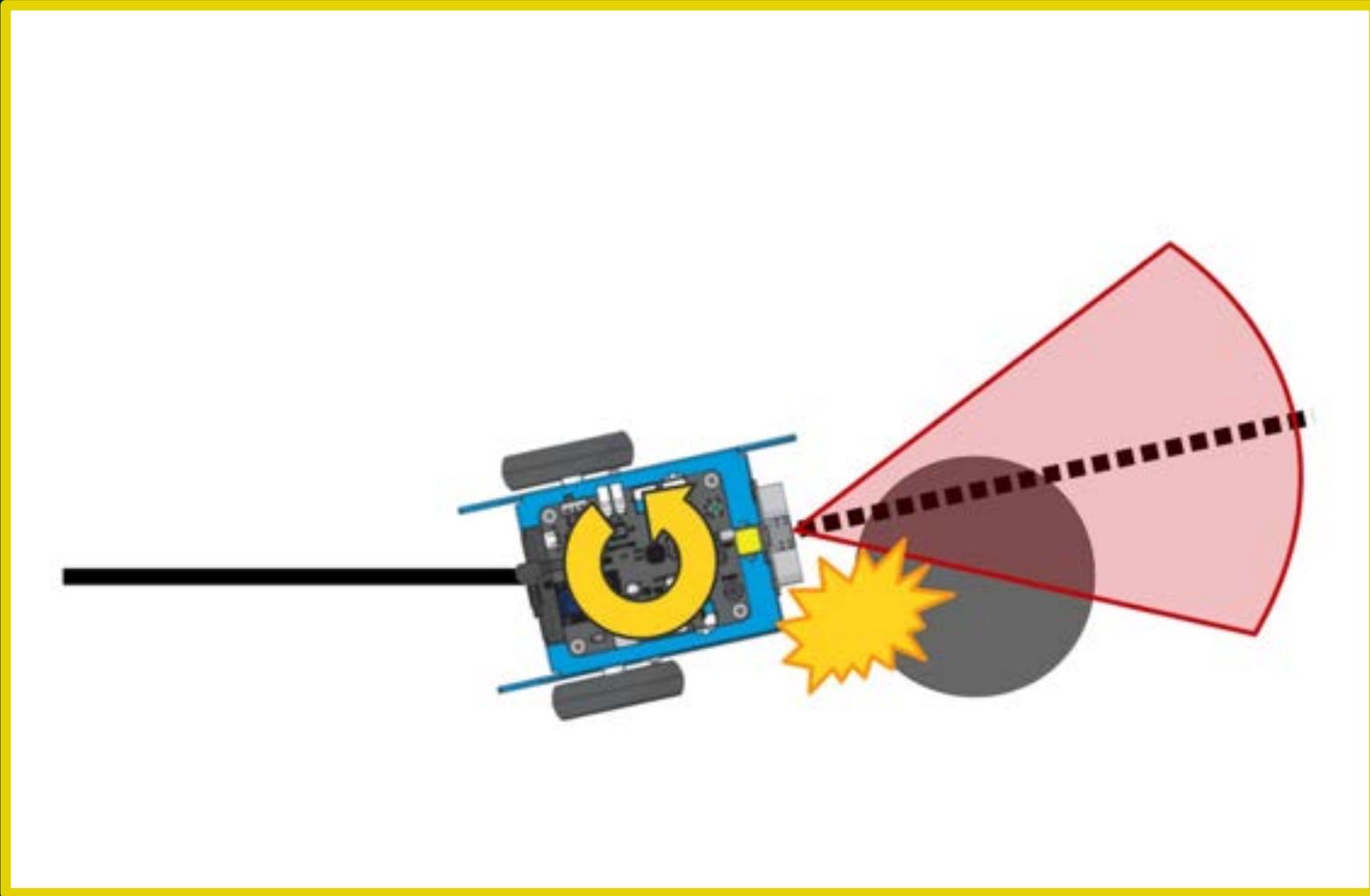
# Esemplificazione



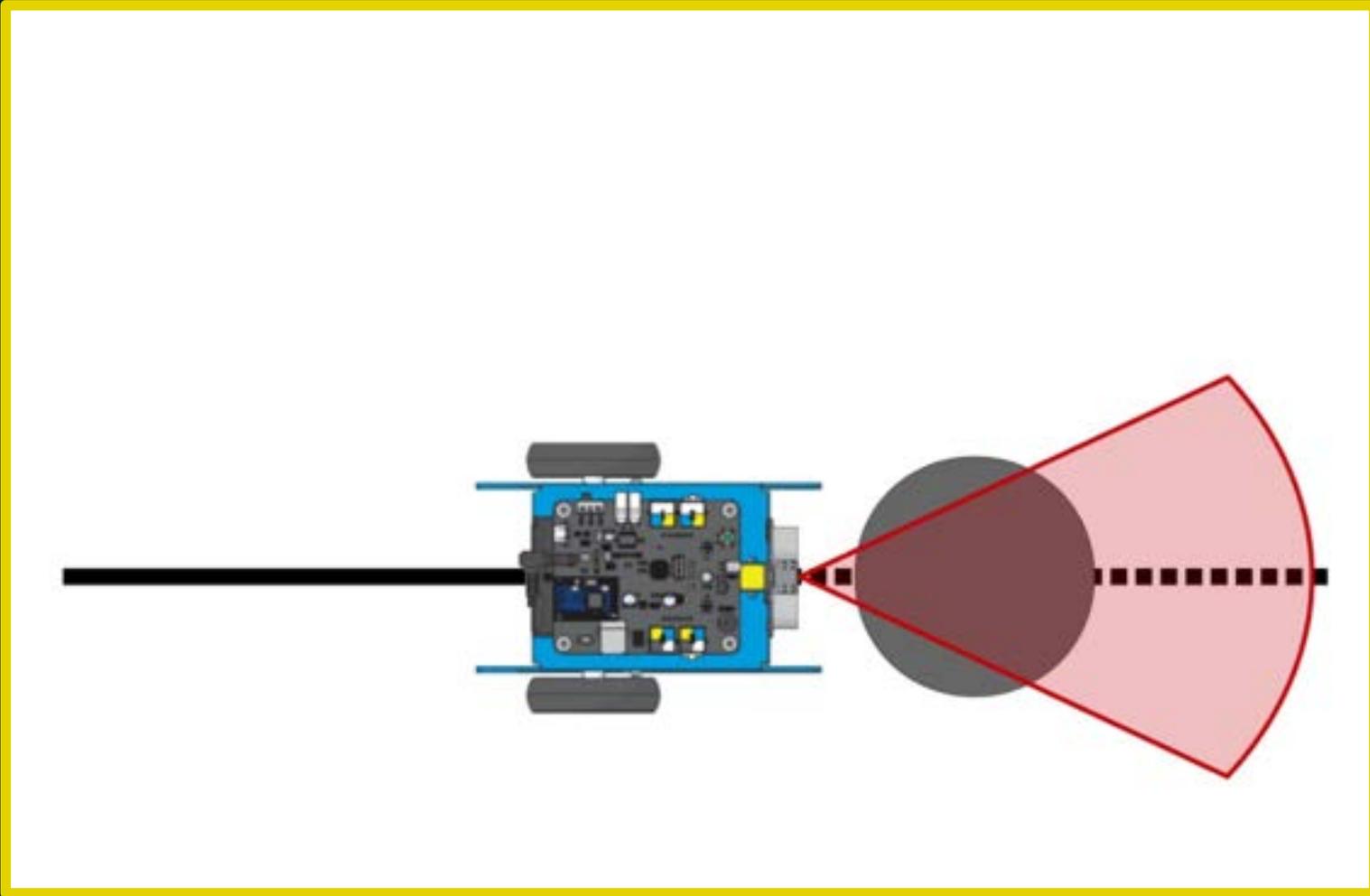
# Esemplificazione



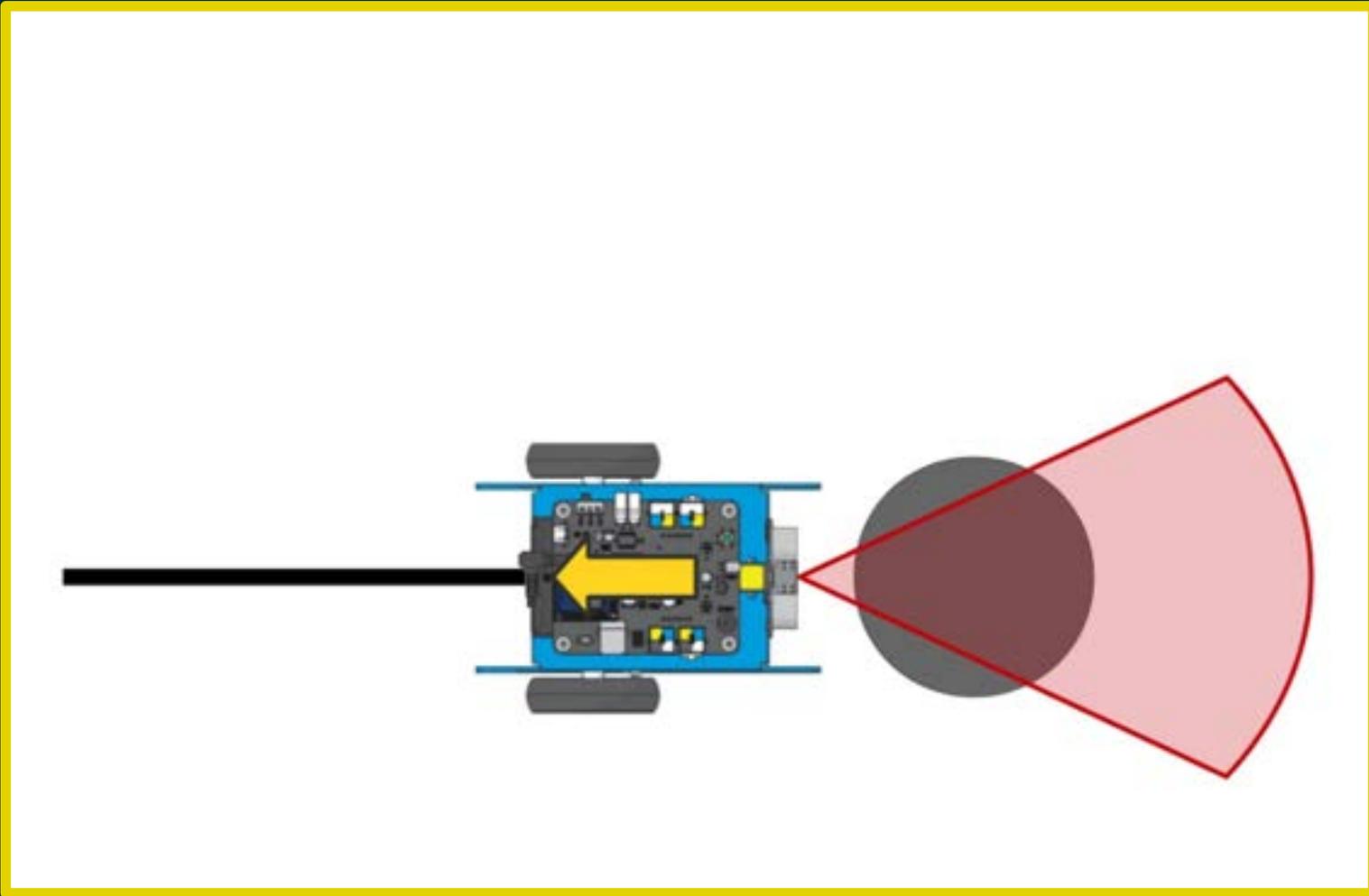
# Esemplificazione



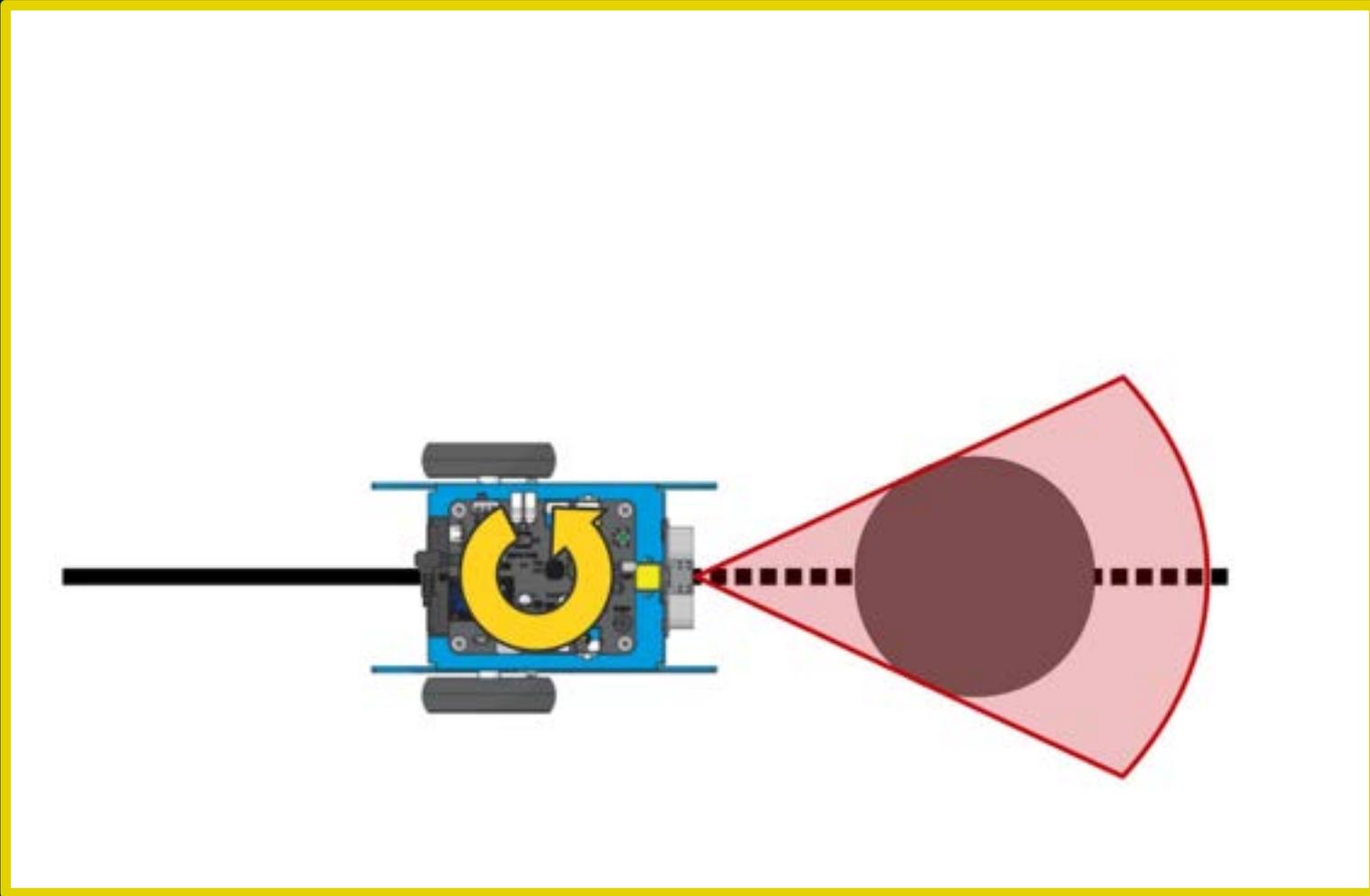
# Esemplificazione



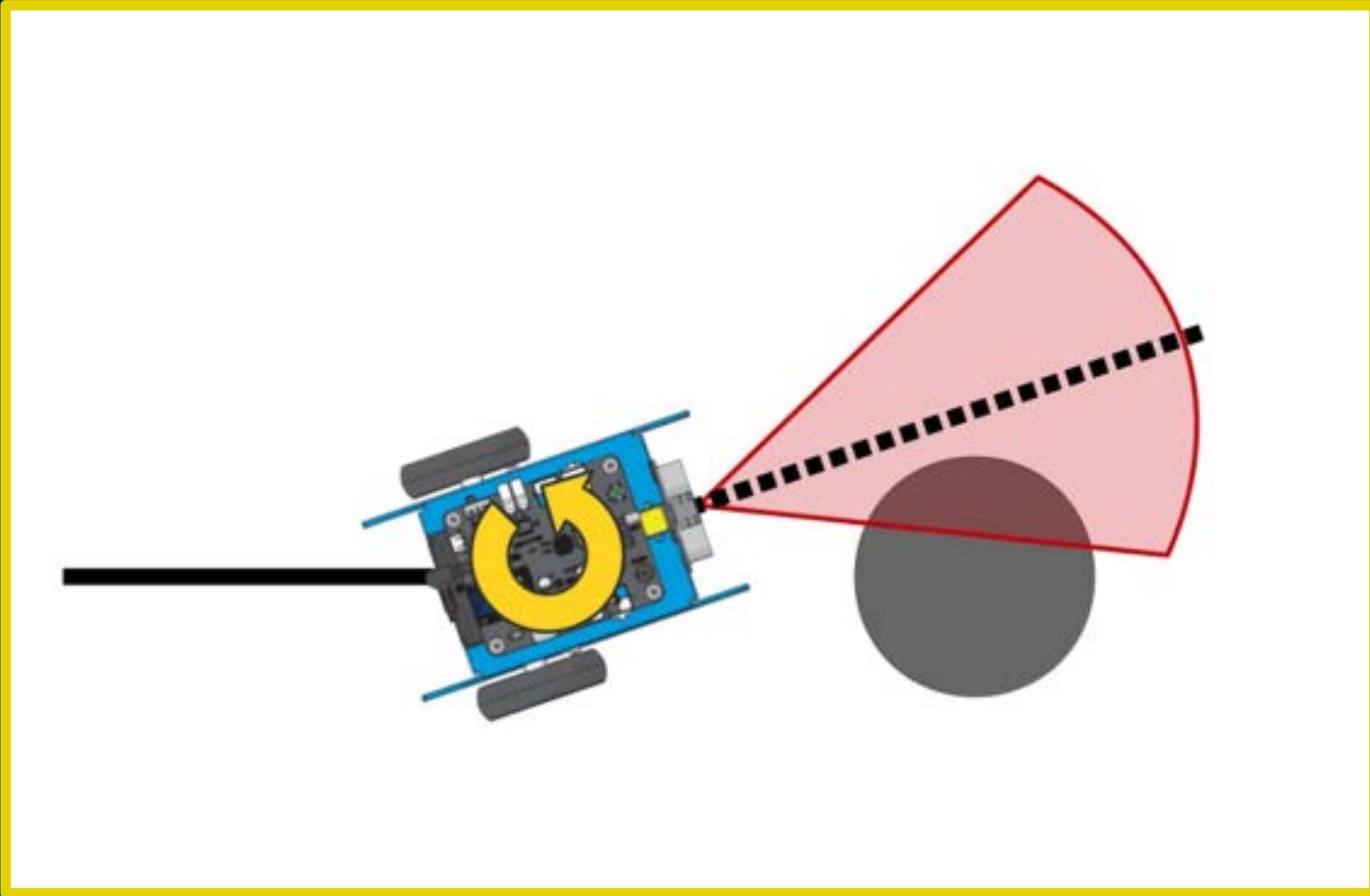
# Esemplificazione



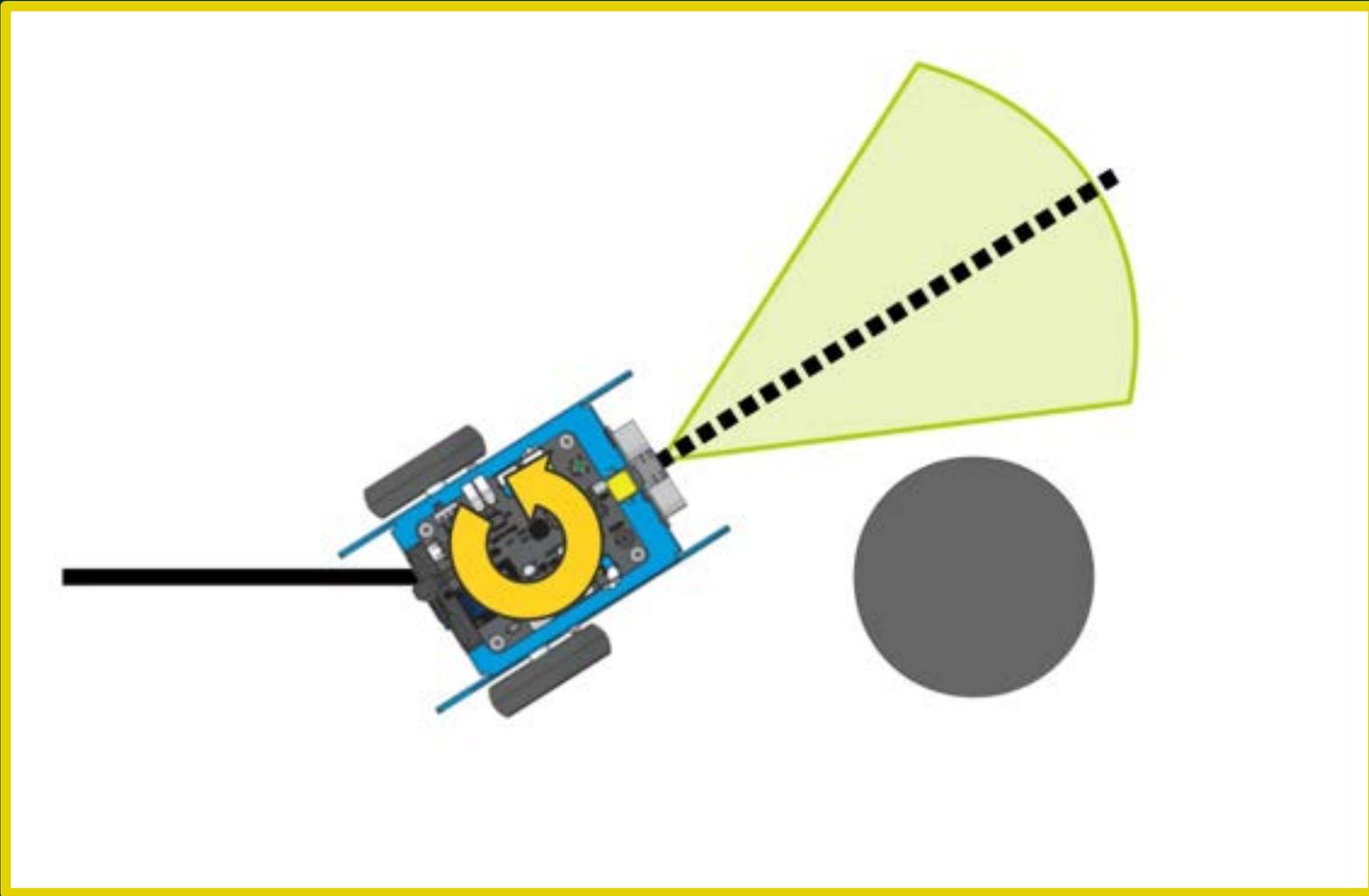
# Esemplificazione



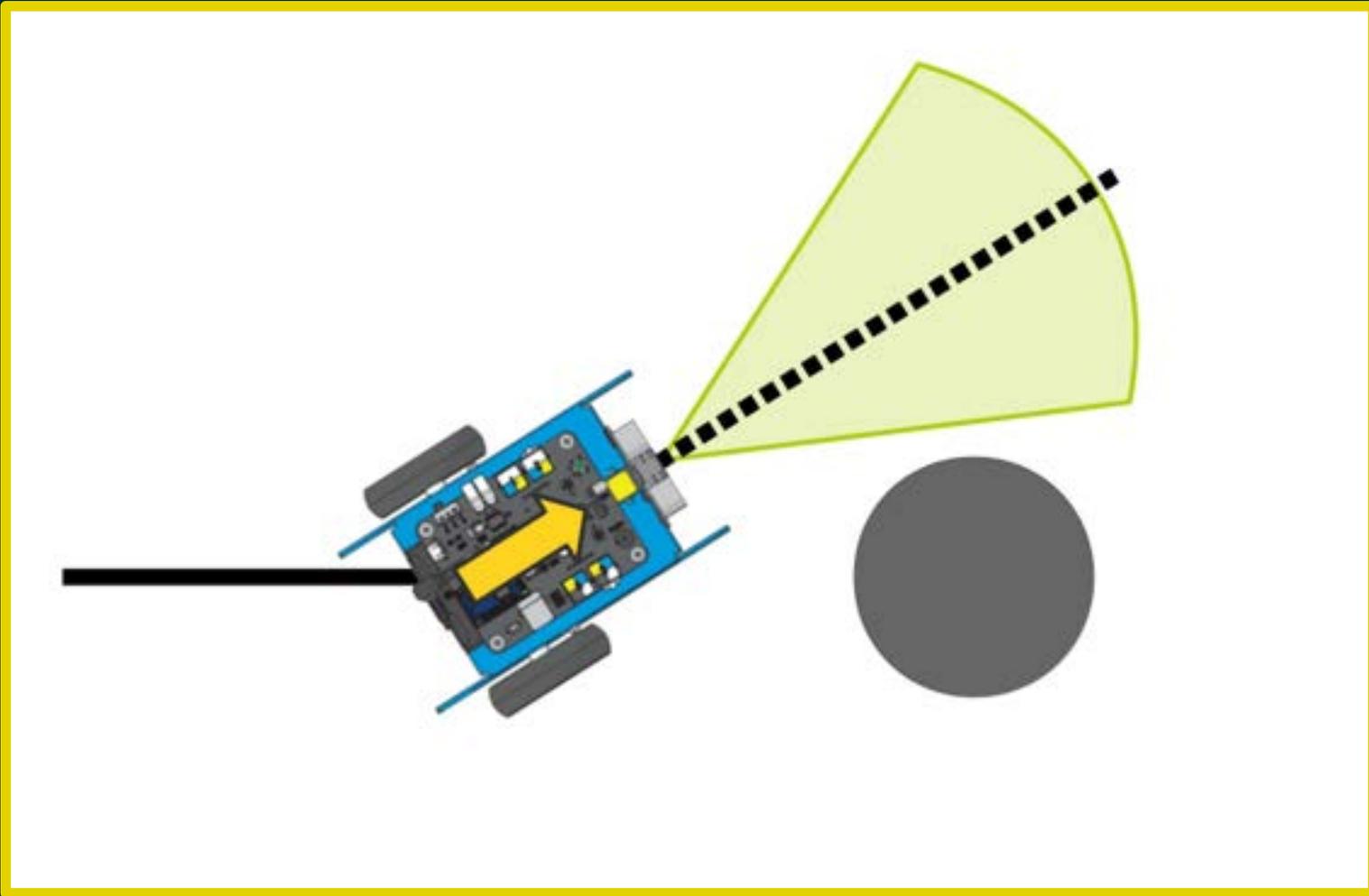
# Esemplificazione



# Esemplificazione



# Esemplificazione



# Diagramma di flusso



# Sfida n.2

Poniamo mBot al centro di un'area delimitata da barriere e cosparsa di ostacoli. Vince il robot che resiste più tempo muovendosi all'interno dell'arena senza urtare gli ostacoli e la recinzione perimetrica.

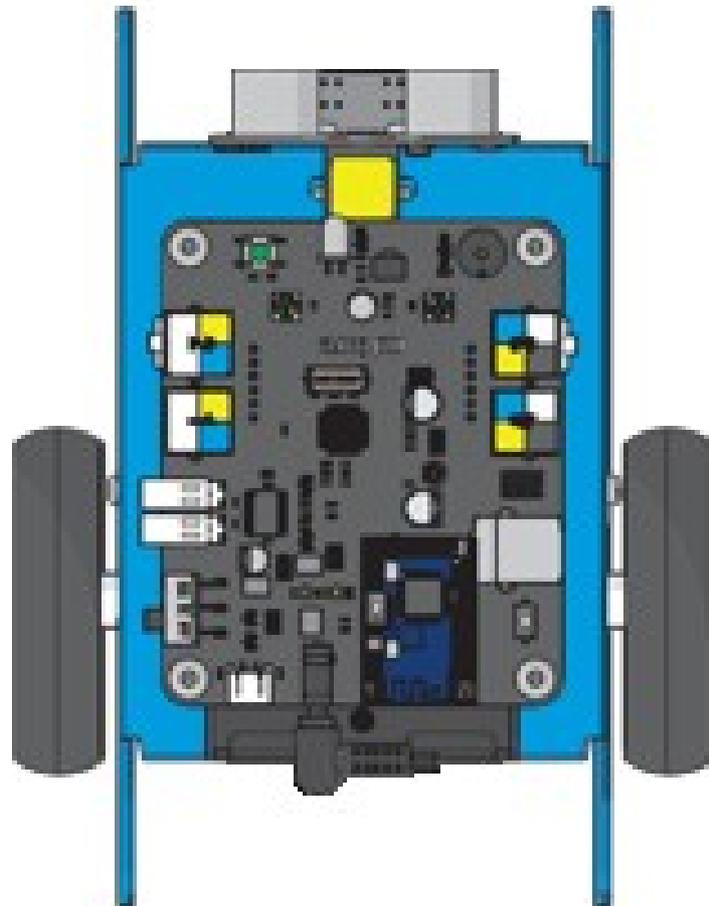
# Controllo dei motori

# Controllo dei motori

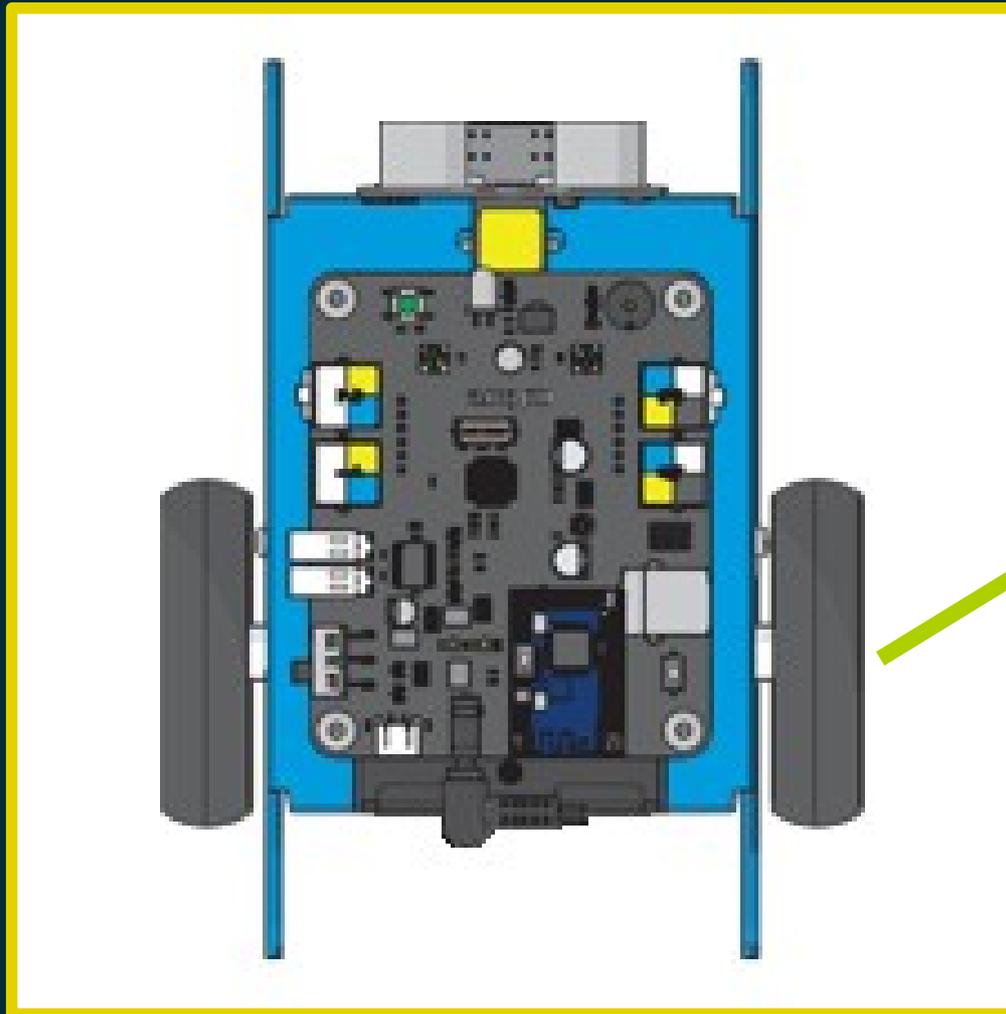
I due motori di mBot sono indipendenti.

# Controllo dei motori

MI

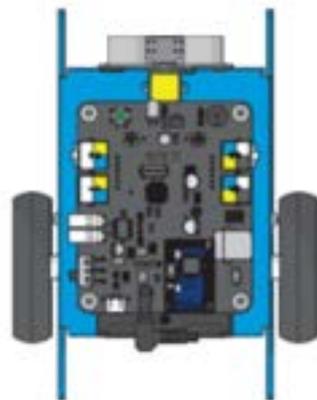


# Controllo dei motori

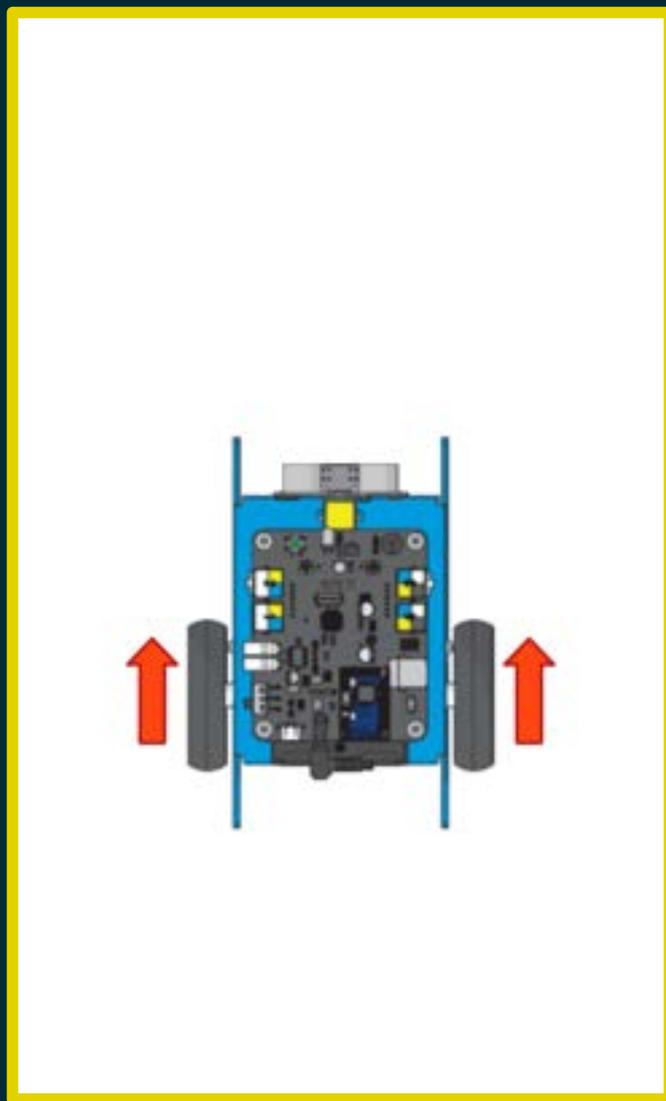


M2

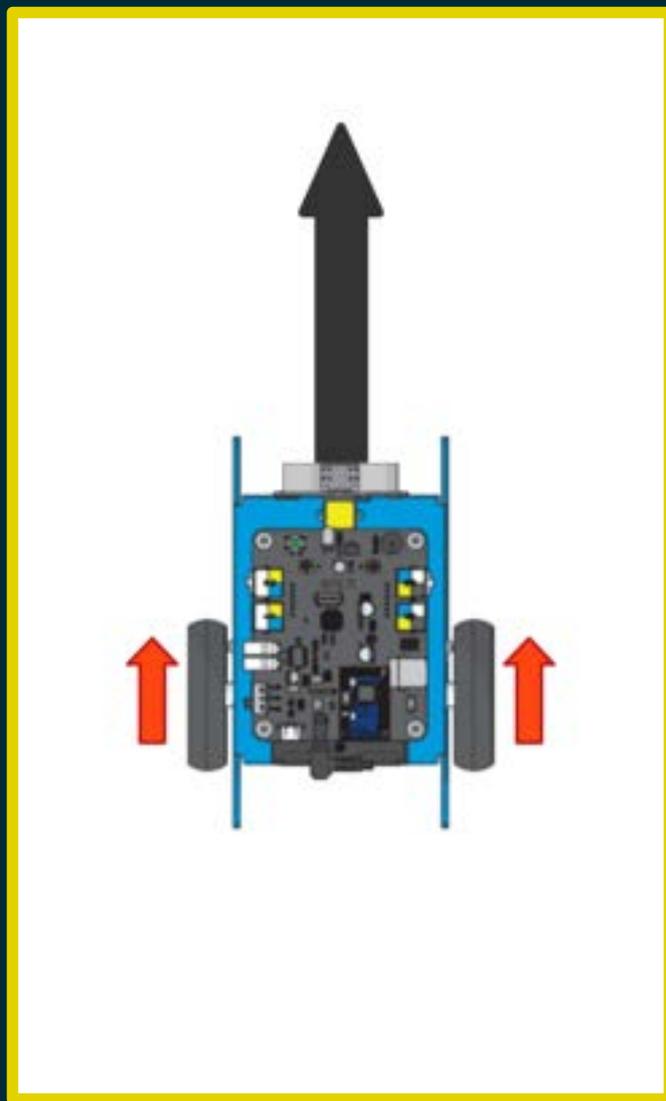
# Traiettorie



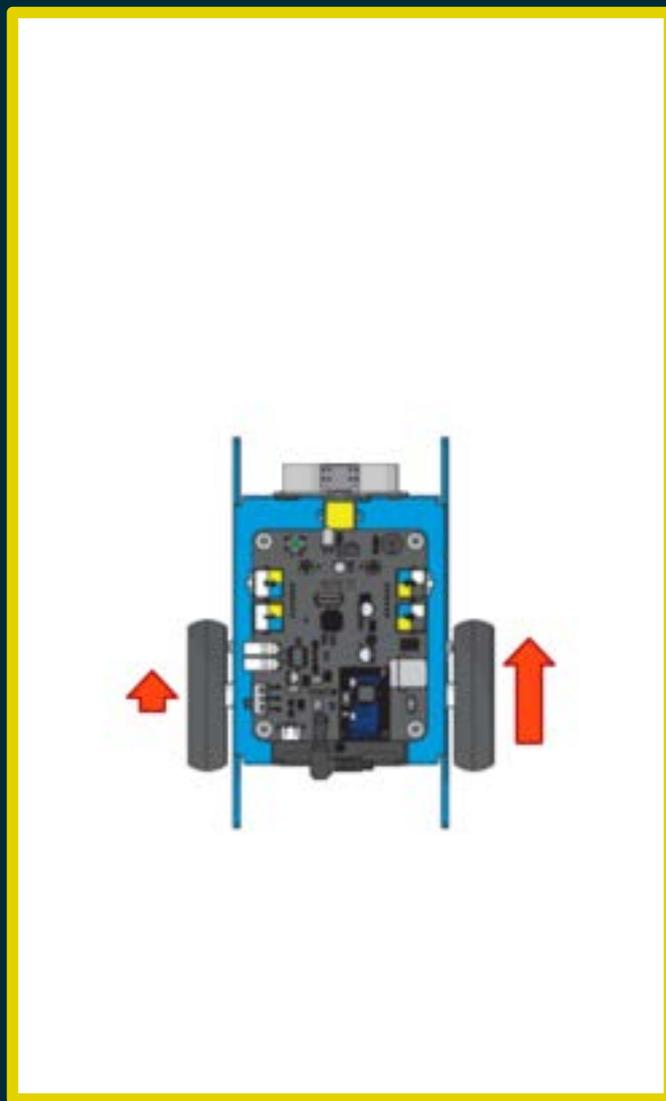
# Traiettorie



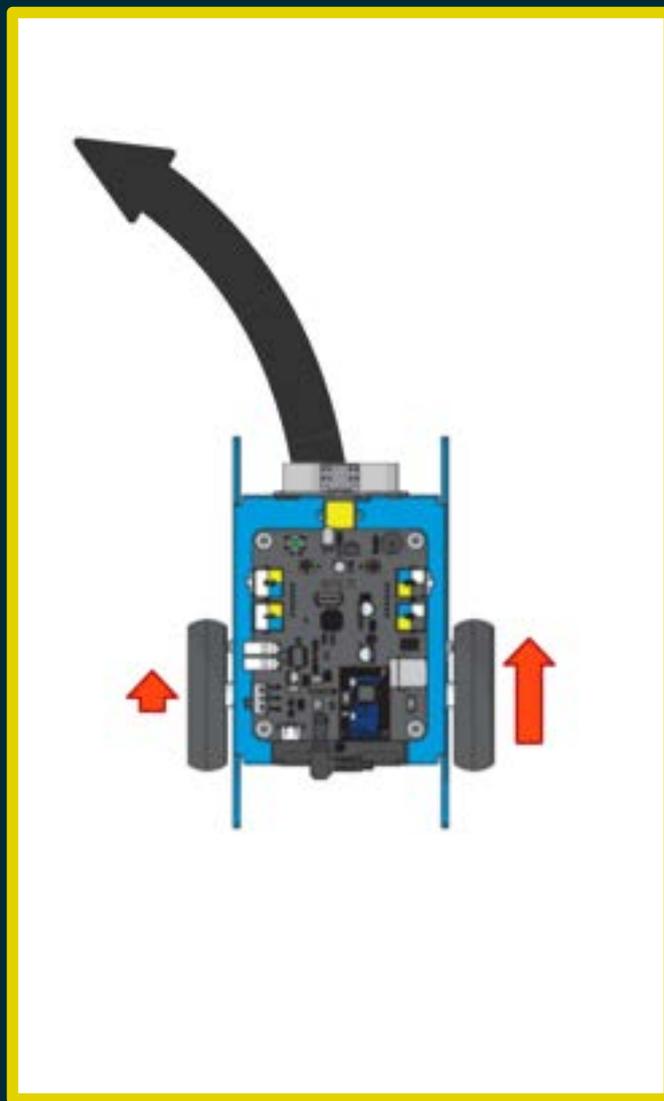
# Traiettorie



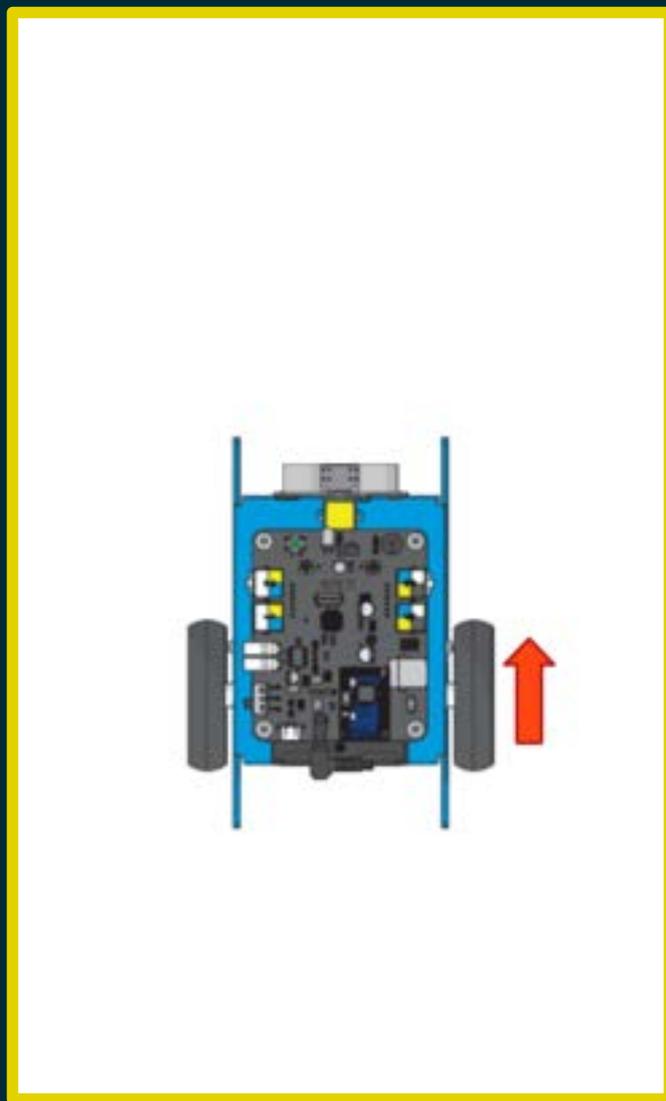
# Traiettorie



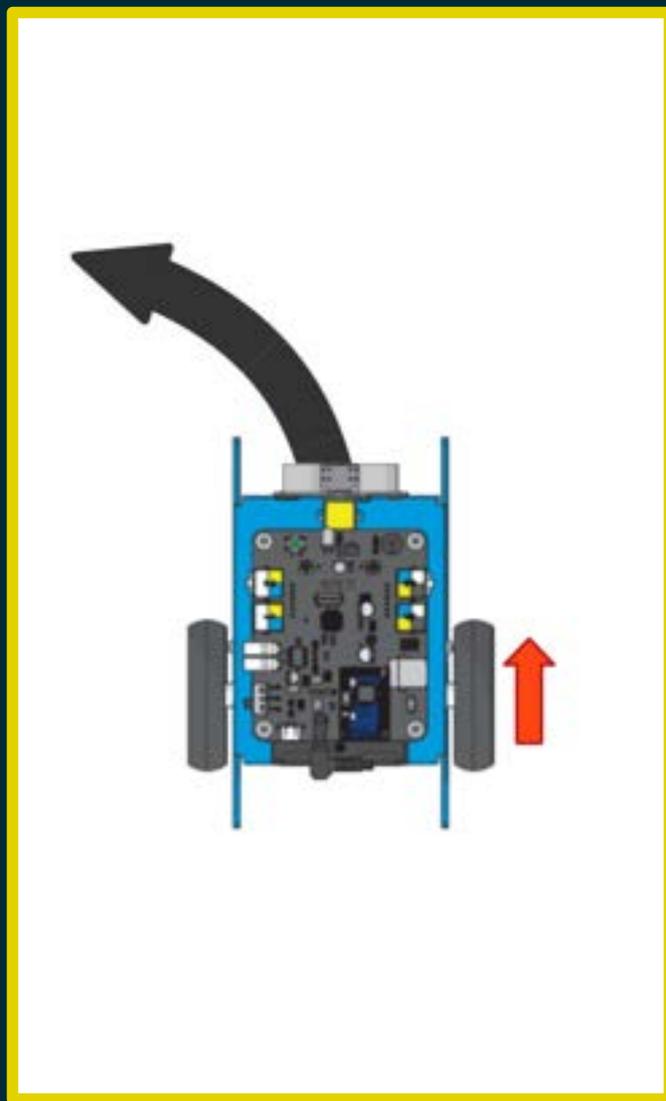
# Traiettorie



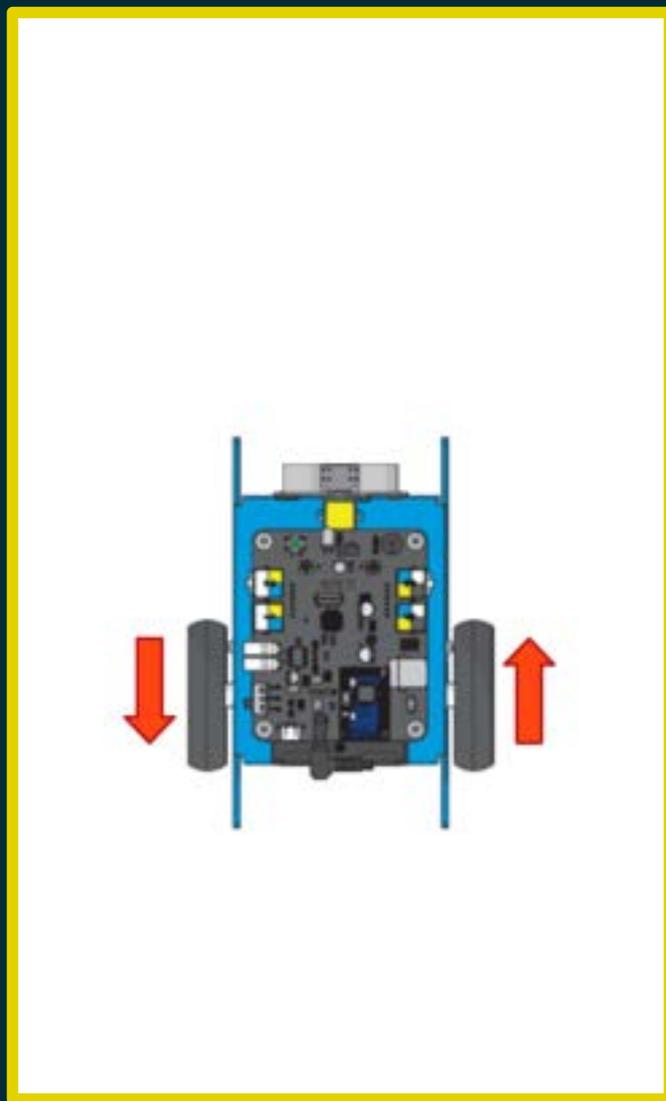
# Traiettorie



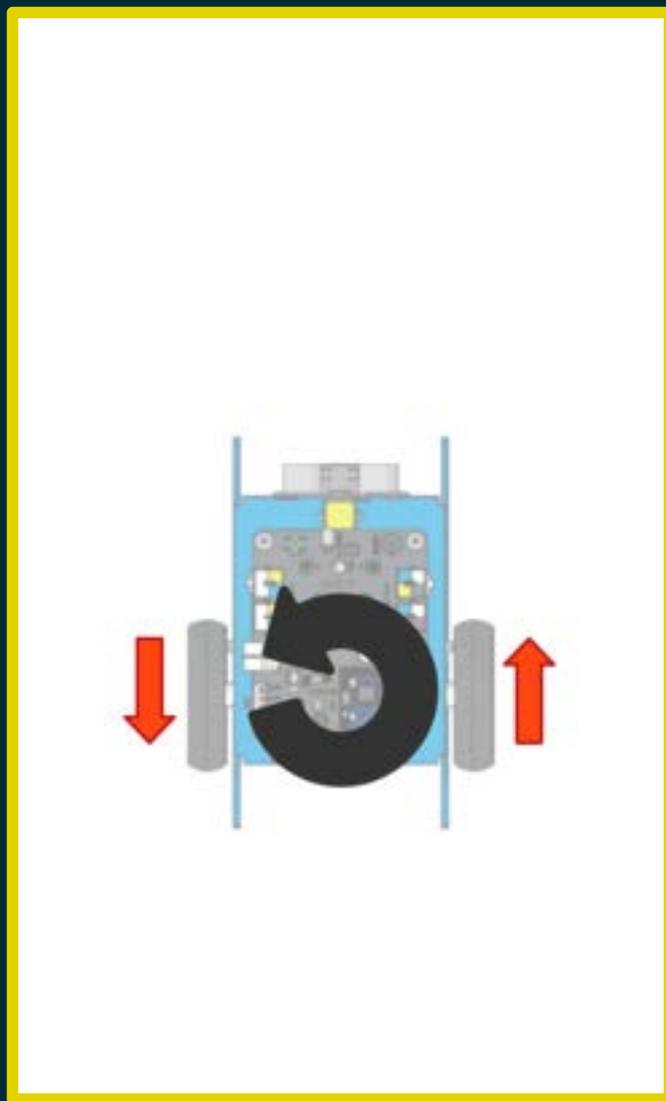
# Traiettorie



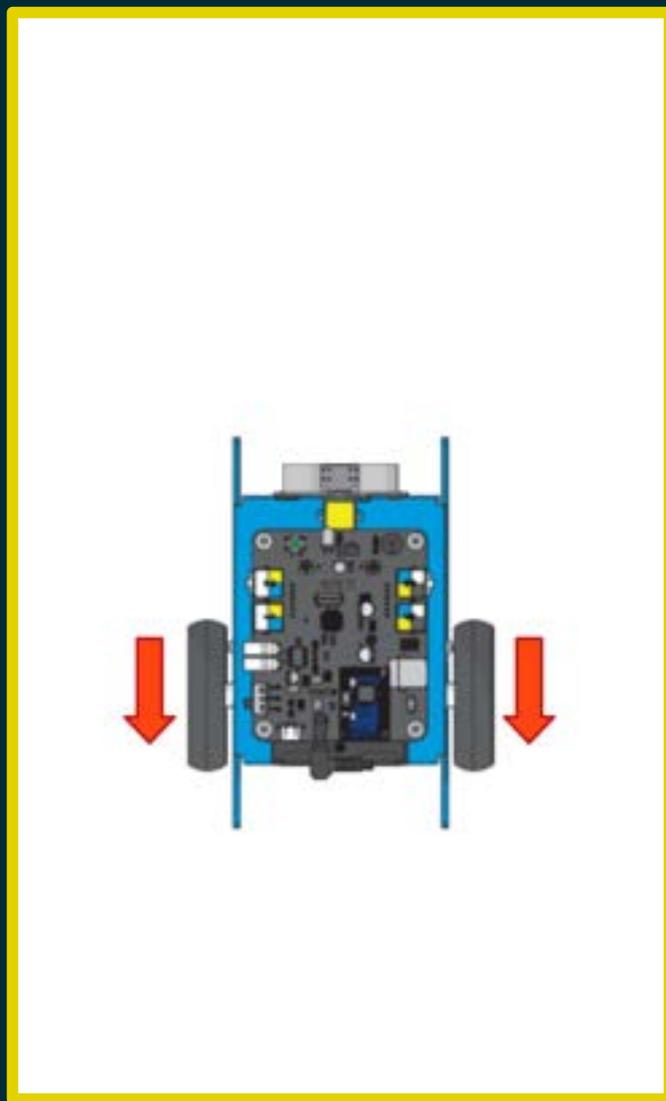
# Traiettorie



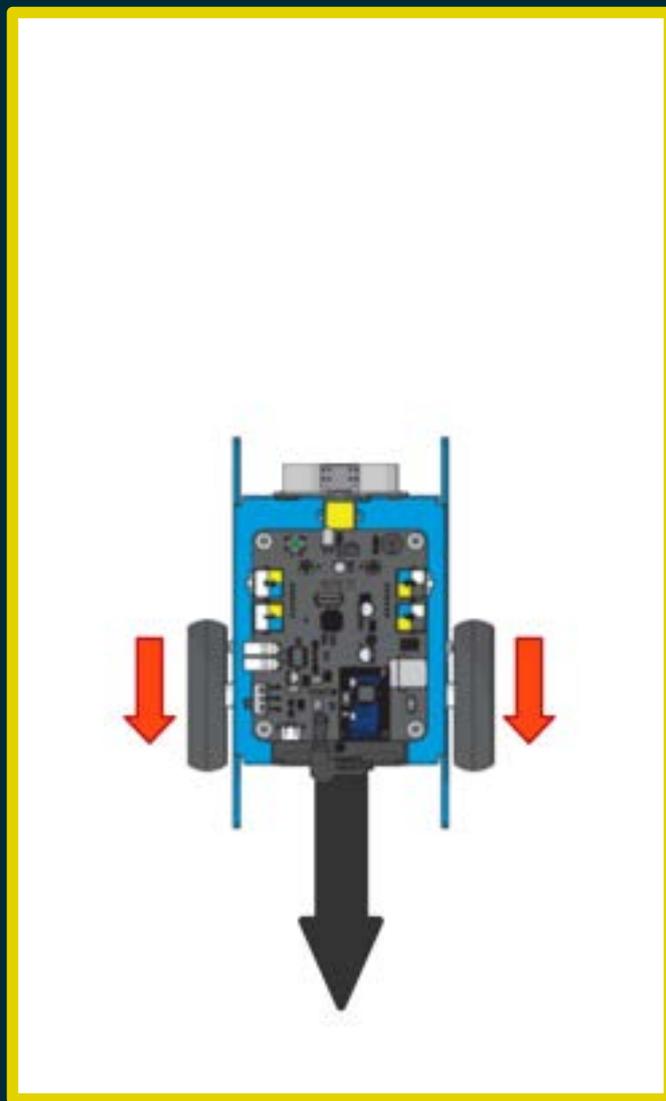
# Traiettorie



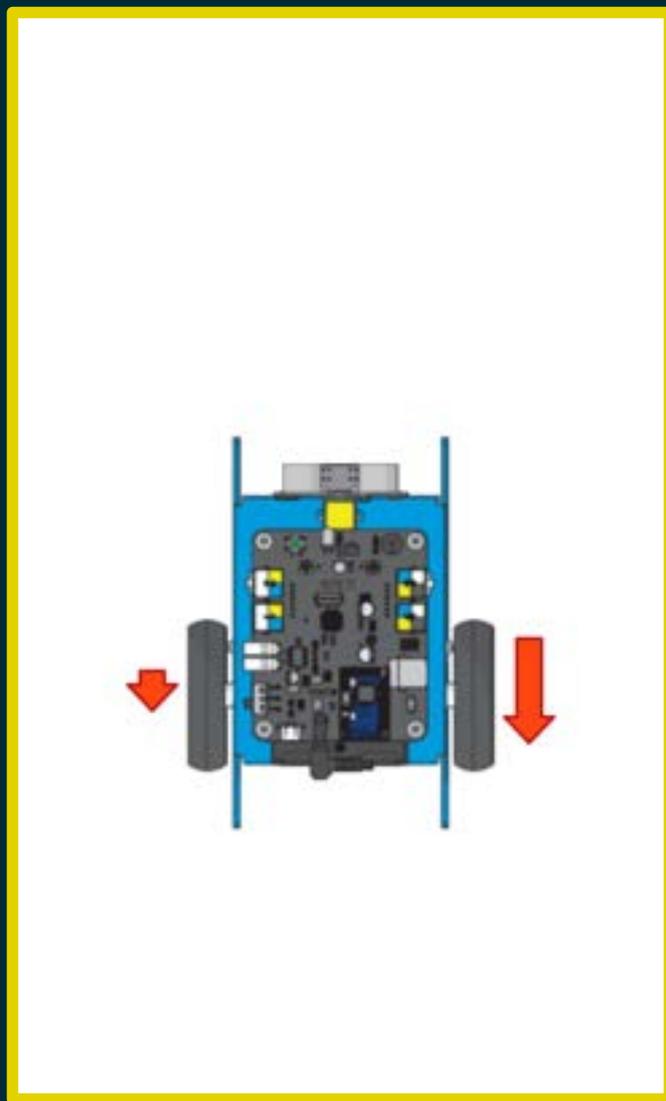
# Traiettorie



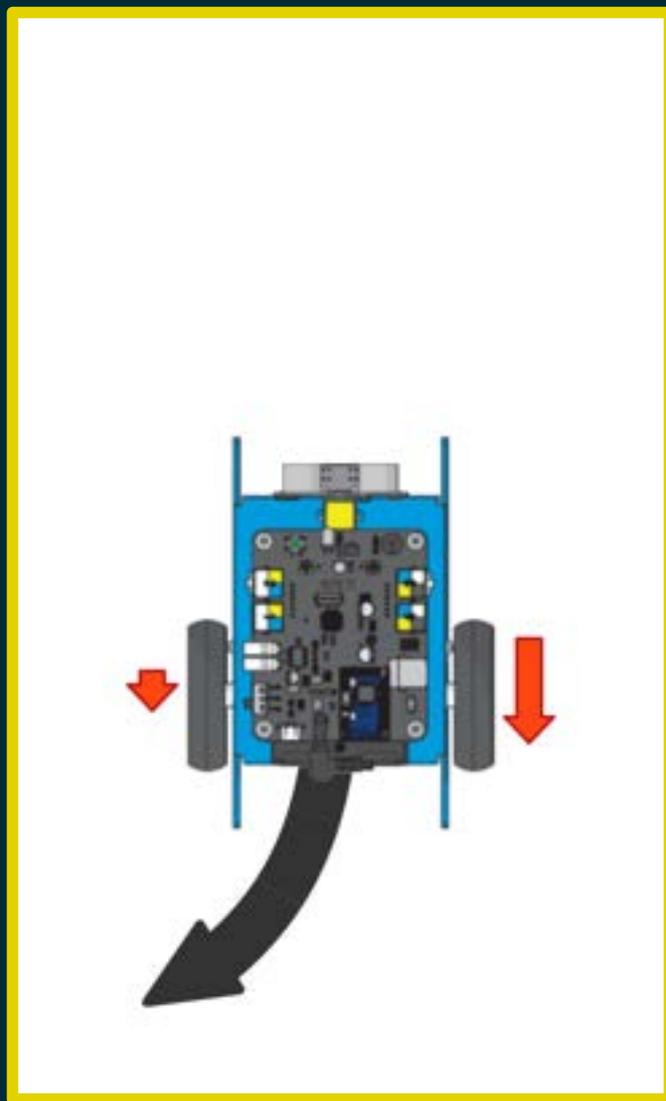
# Traiettorie



# Traiettorie



# Traiettorie



# Quiz!

when mBot(mcore) starts up

wait until  when on-board button pressed ▼ ?

 left wheel turns at power 50 %, right wheel at power 75 %

wait 8 seconds

 stop moving

# Esercitazione

Percorrere una traiettoria a S.

Percorrere una traiettoria a U.

Percorrere una traiettoria a M.

# Scansa ostacoli /4

Modificare il programma scansa ostacoli sostituendo le rotazioni con le sterzate.

# Scansa ostacoli /4

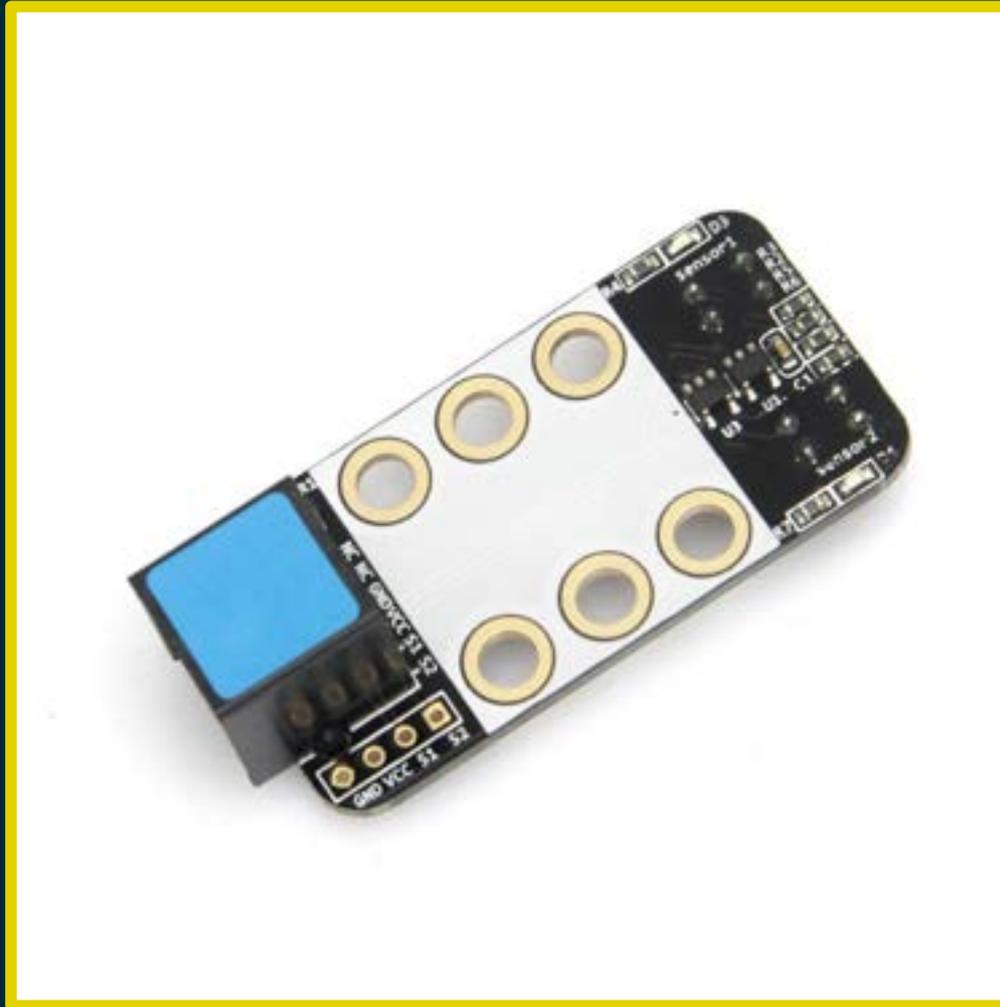
Modificare il programma scansa ostacoli sostituendo le rotazioni con le sterzate.

La retromarcia in linea retta può essere sostituita con una sterzata all'indietro.

# Programma n.6

Inseguire una linea

# Sensore di linea



# Applicazione



# Sensore di linea



# Sensore di linea



# Sensore riflettivo IR



# Sensore riflettivo

Trasmittitore



# Sensore riflettivo

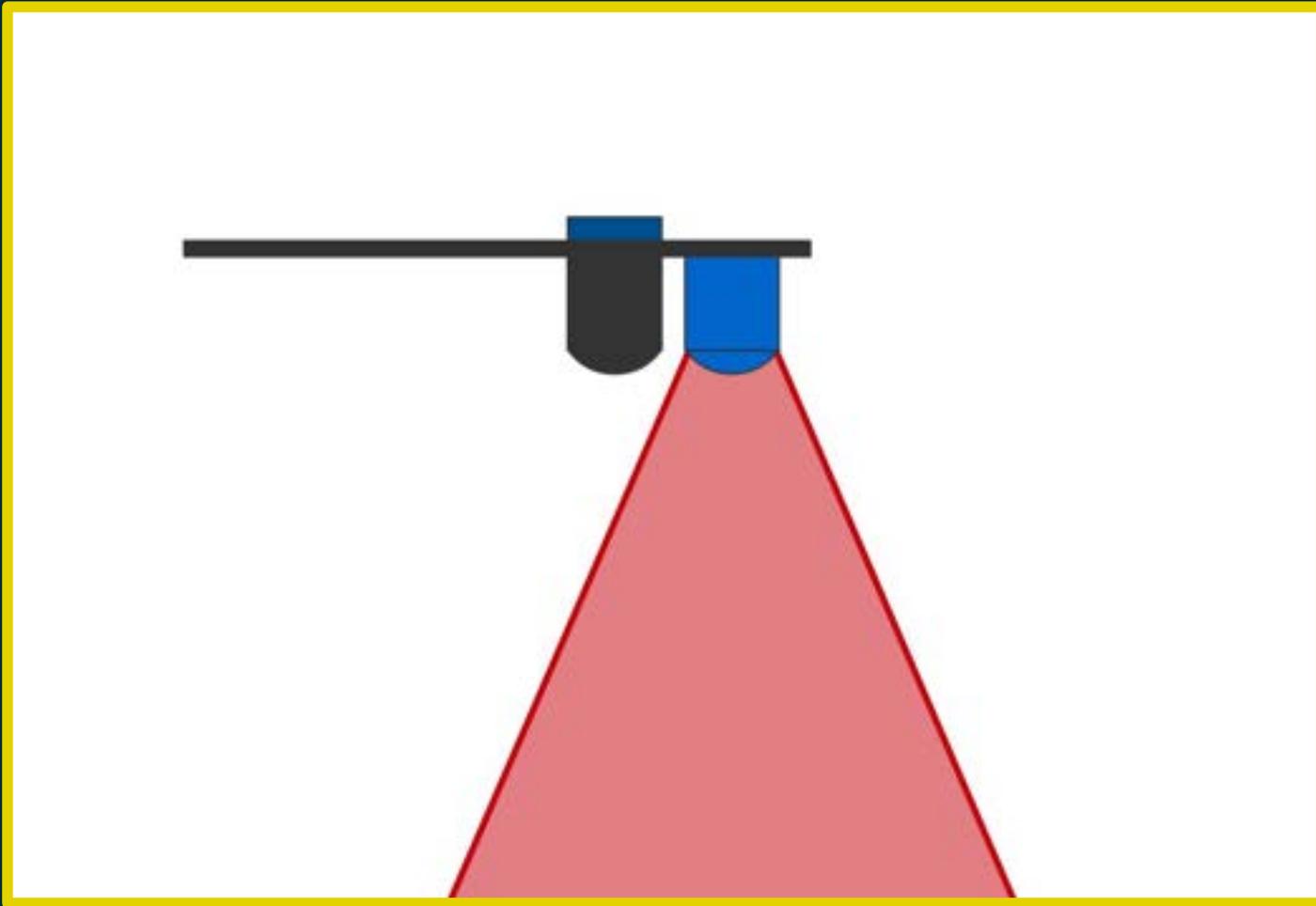


Ricevitore

# Principio di funzionamento



# Principio di funzionamento



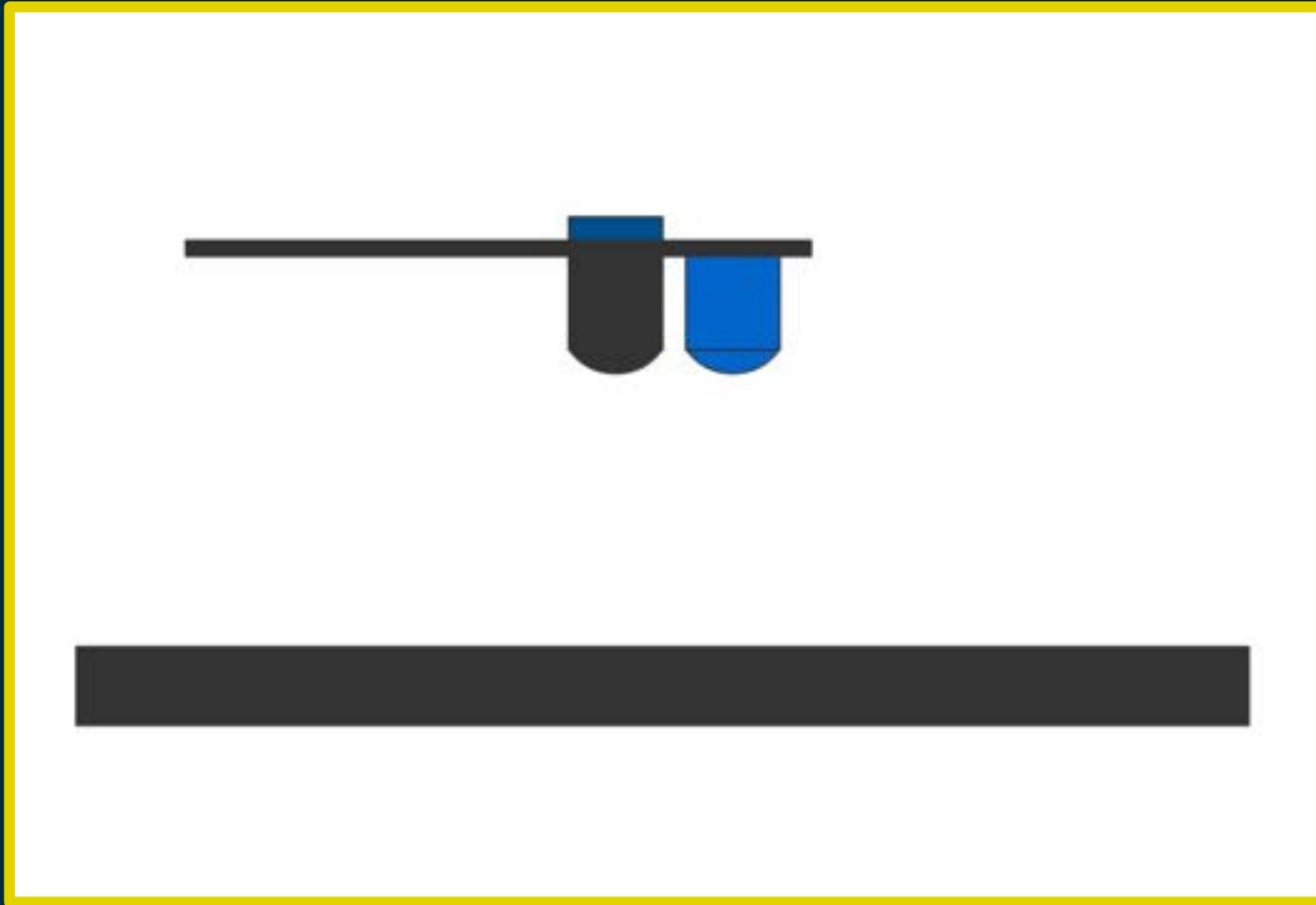
# Principio di funzionamento



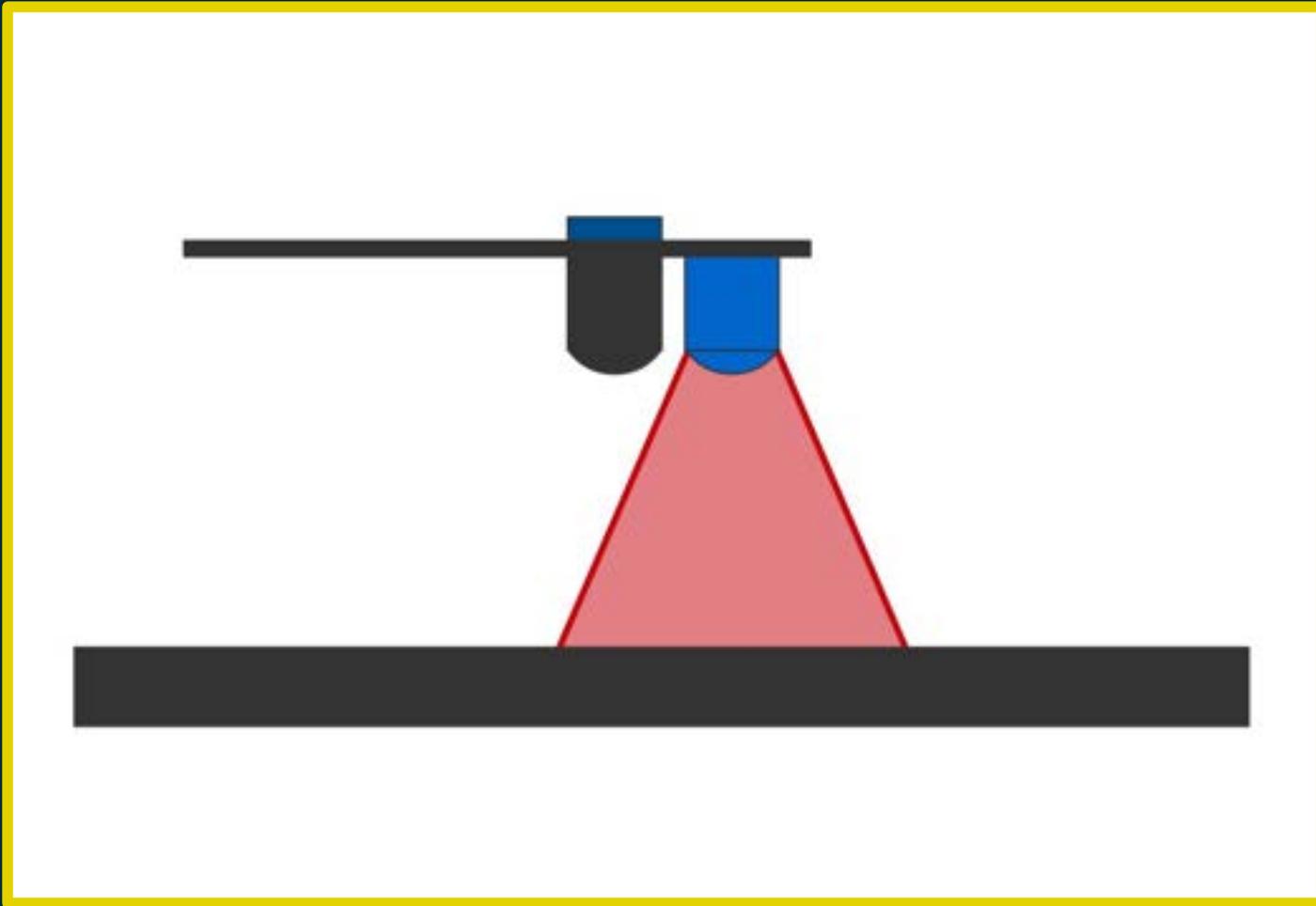
Risposta del sensore:  
nessuna riflessione

The diagram illustrates a sensor's operating principle. It features a horizontal black line representing a light beam. Below this line, a blue rectangular shape represents a lens. A dark purple triangular area, representing the sensor's field of view, originates from the lens and extends downwards. The text 'Risposta del sensore: nessuna riflessione' is written in yellow, indicating that the sensor receives a signal without any reflection.

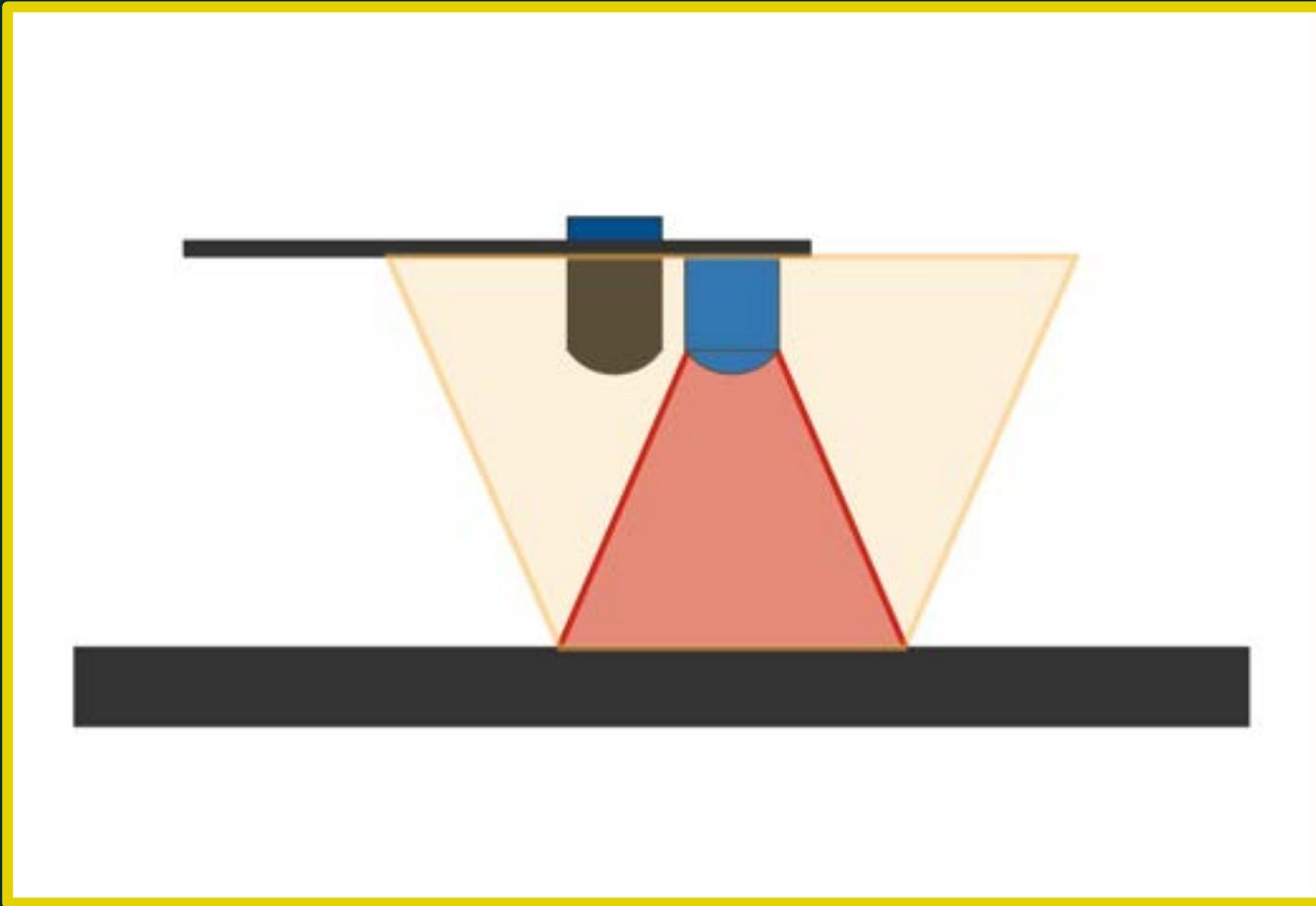
# Principio di funzionamento



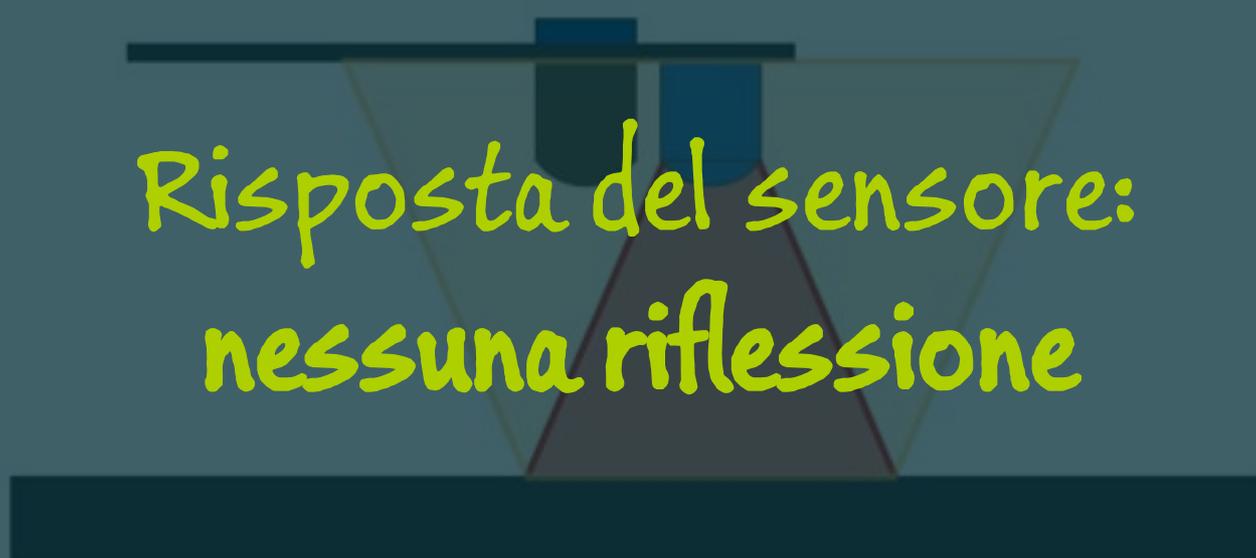
# Principio di funzionamento



# Principio di funzionamento

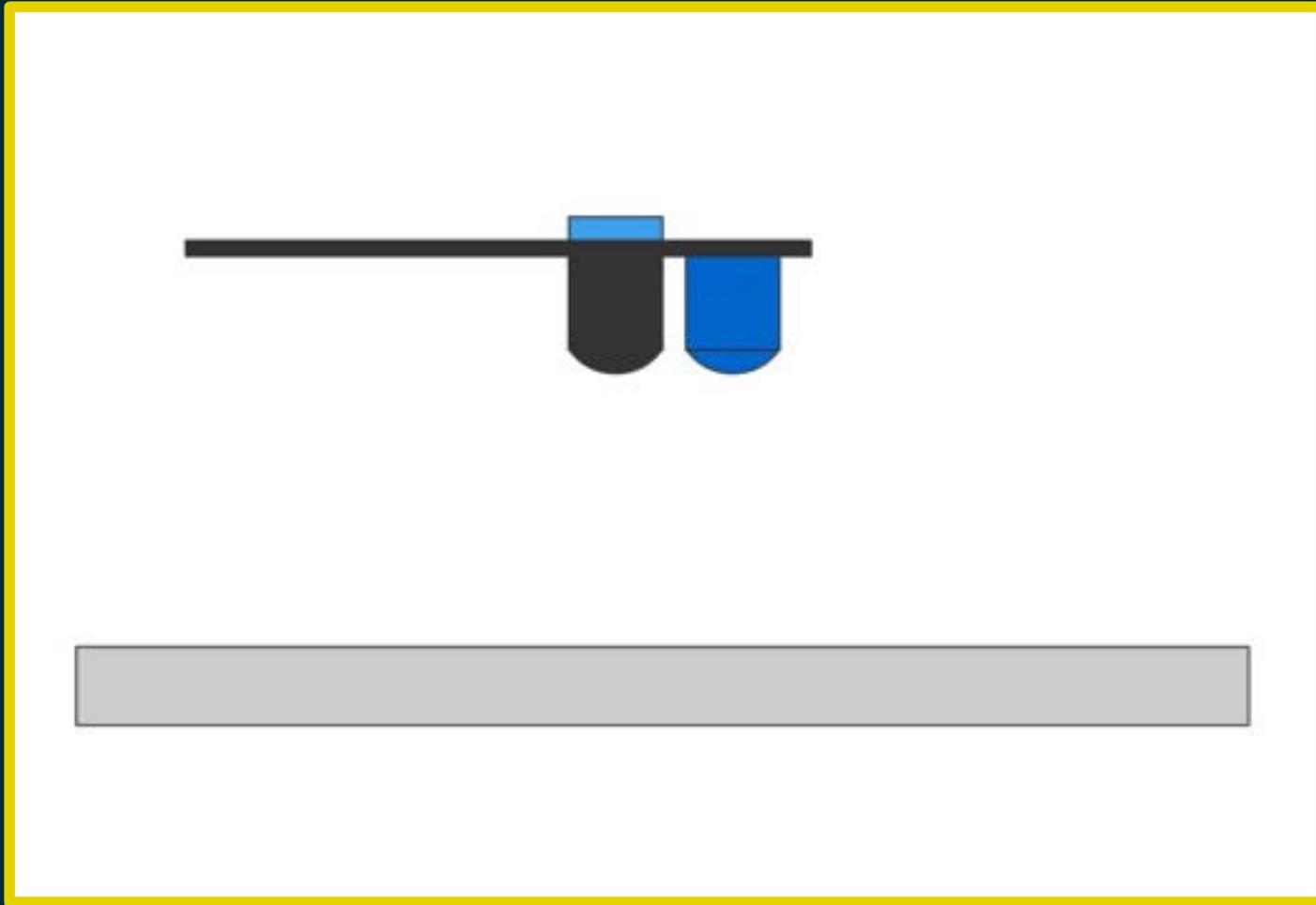


# Principio di funzionamento

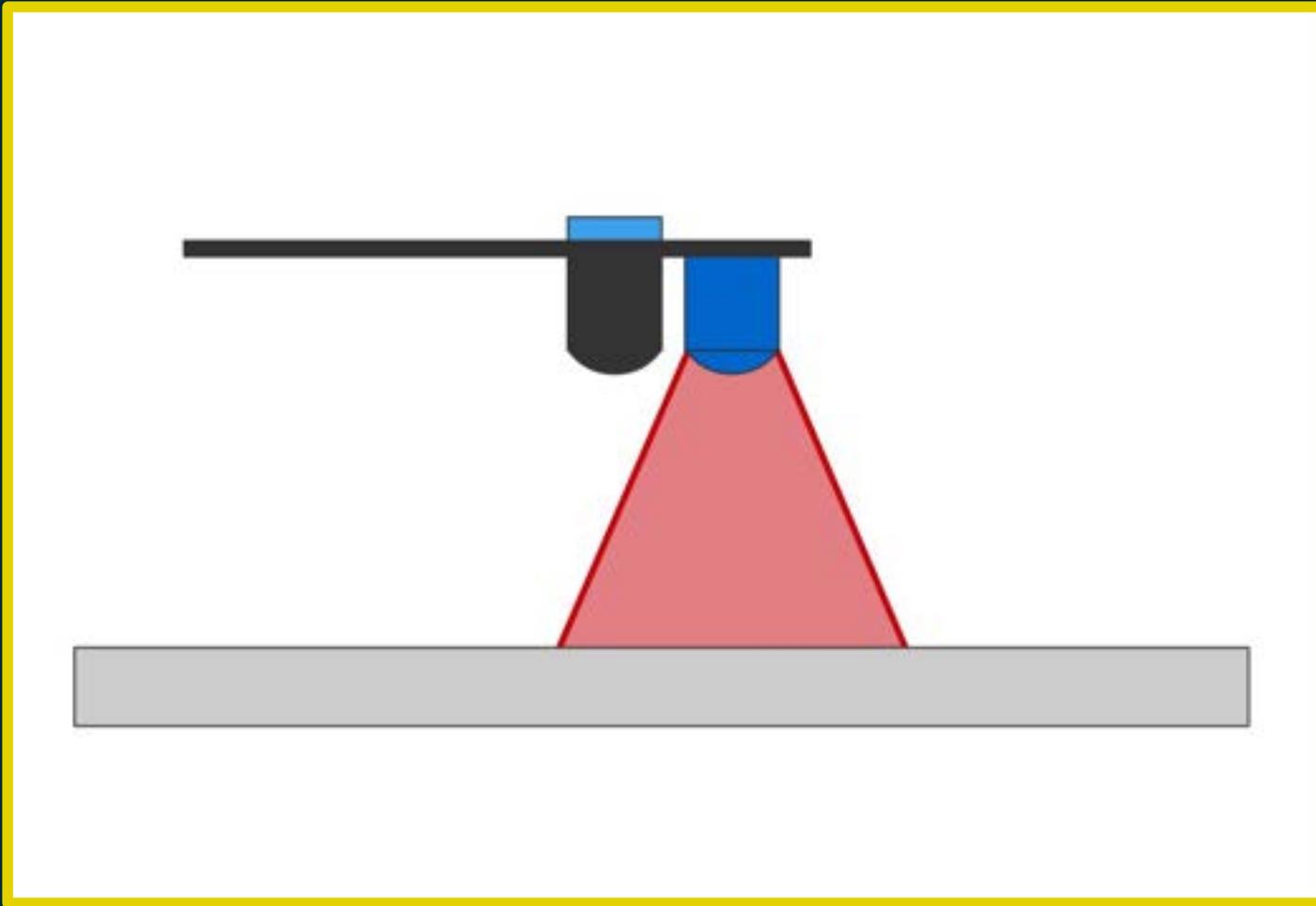


Risposta del sensore:  
nessuna riflessione

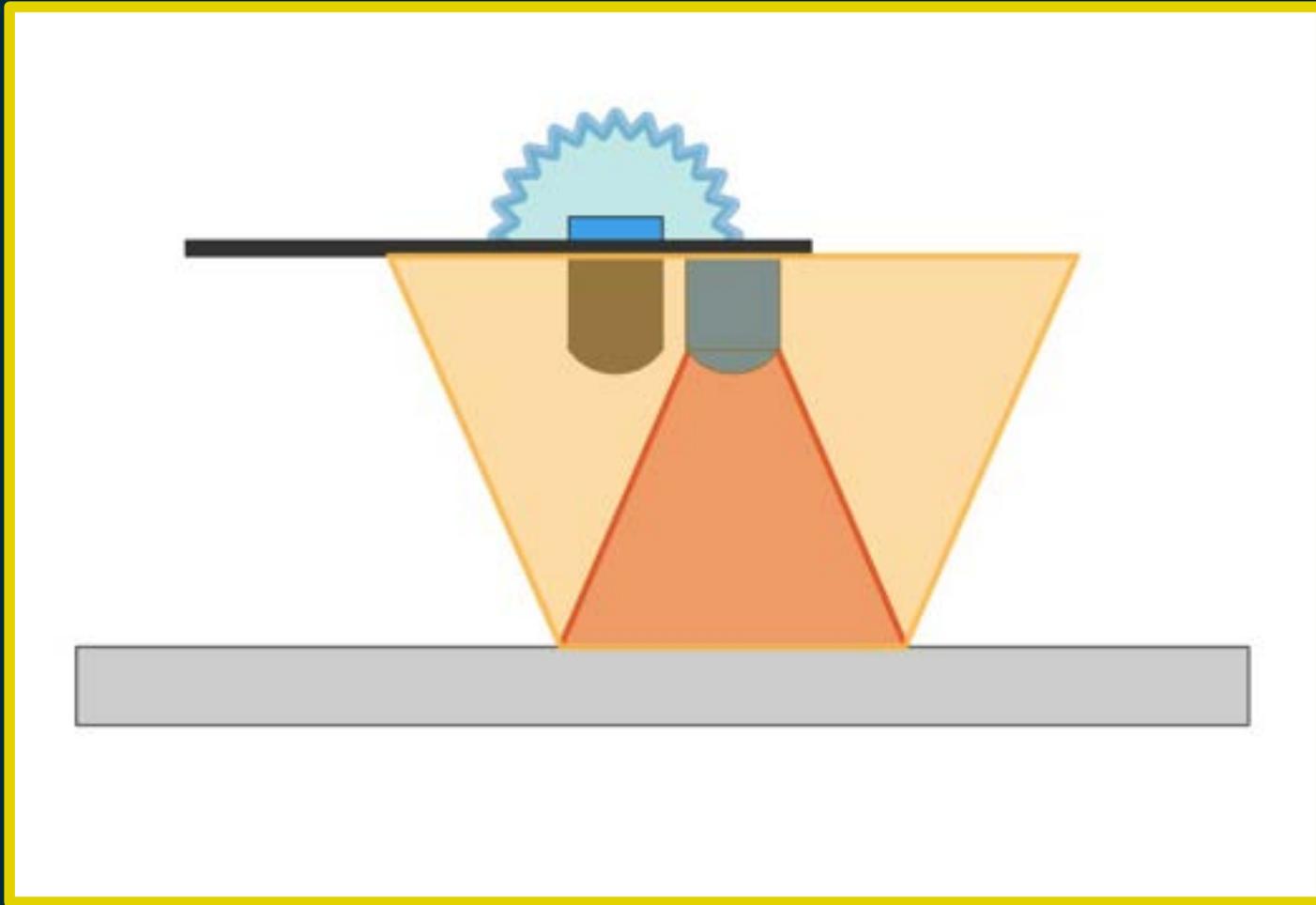
# Principio di funzionamento



# Principio di funzionamento



# Principio di funzionamento



# Principio di funzionamento



Risposta del sensore:  
c'è riflessione!

The diagram illustrates a sensor principle. A blue wavy line representing a signal or wave is shown above a horizontal black line. Below this line, a blue rectangular block is positioned. A red line indicates a reflection path from the block to a horizontal grey bar at the bottom. A green trapezoidal shape is overlaid on the diagram, and the text 'Risposta del sensore: c'è riflessione!' is written in yellow.

# Sensore riflettivo

Mettiamo alla prova il sensore...

# Sensore riflettivo

Mettiamo alla prova il sensore...

- sul bianco/nero

# Sensore riflettivo

Mettiamo alla prova il sensore...

- sul bianco/nero
- con varie tinte

# Sensore riflettivo

Mettiamo alla prova il sensore...

- sul bianco/nero
- con varie tinte
- con differenti materiali

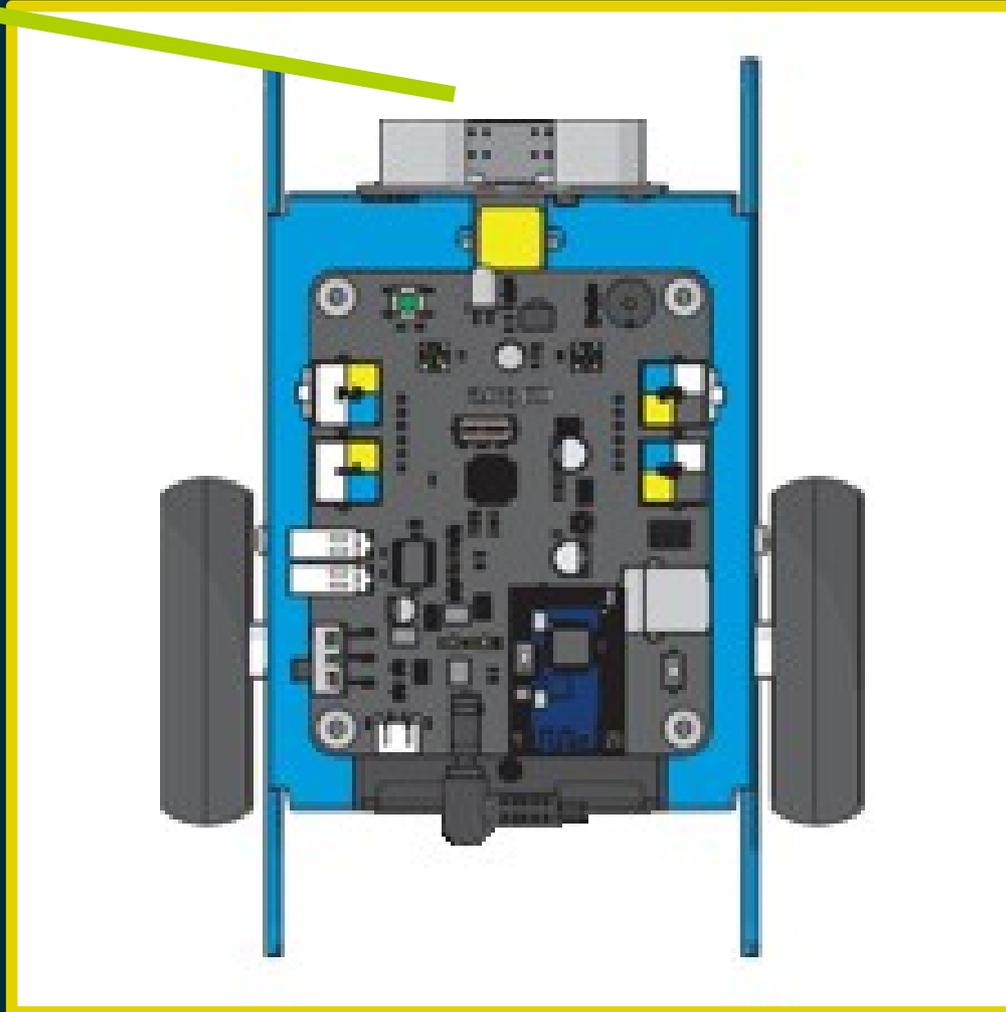
# Sensore riflettivo

Mettiamo alla prova il sensore...

- sul bianco/nero
- con varie tinte
- con differenti materiali
- su diverse superfici

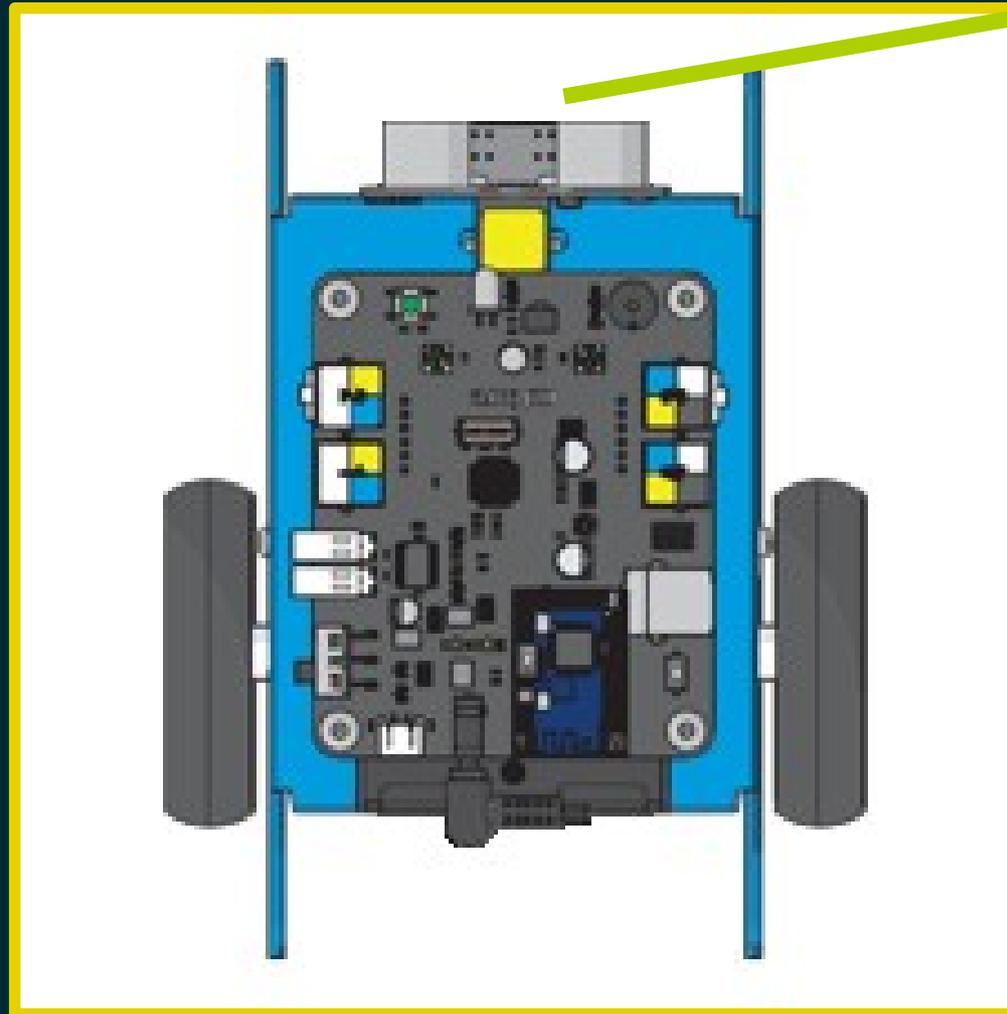
# Sensore di linea

SI

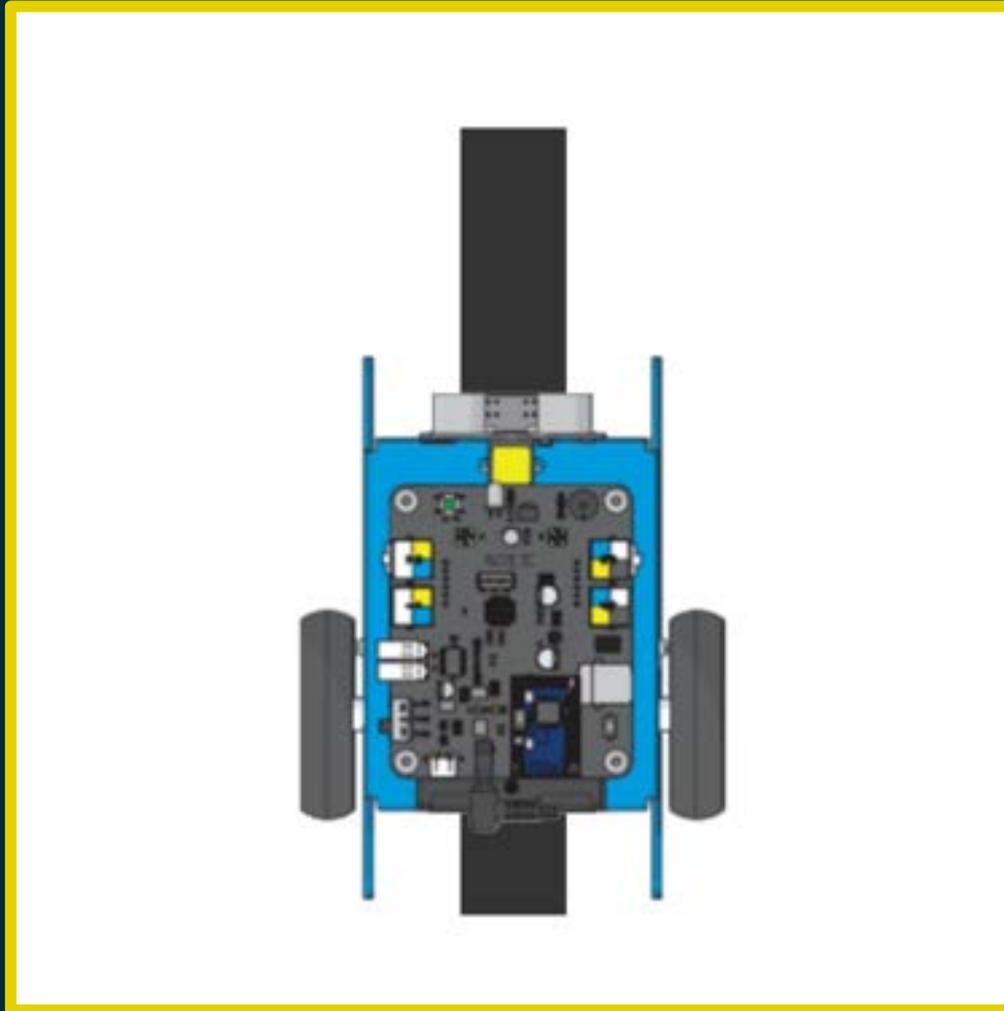


# Sensore di linea

S2



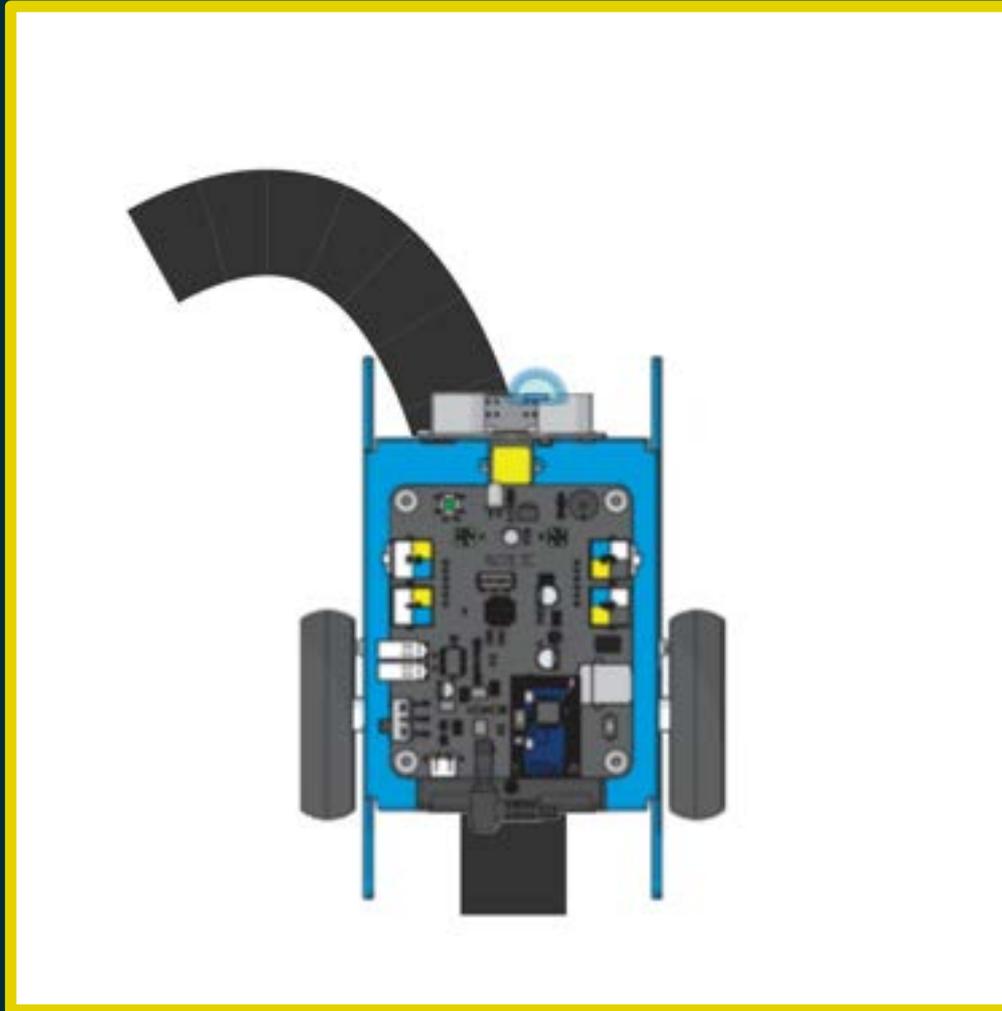
# Valore del sensore di linea



# Valore del sensore di linea



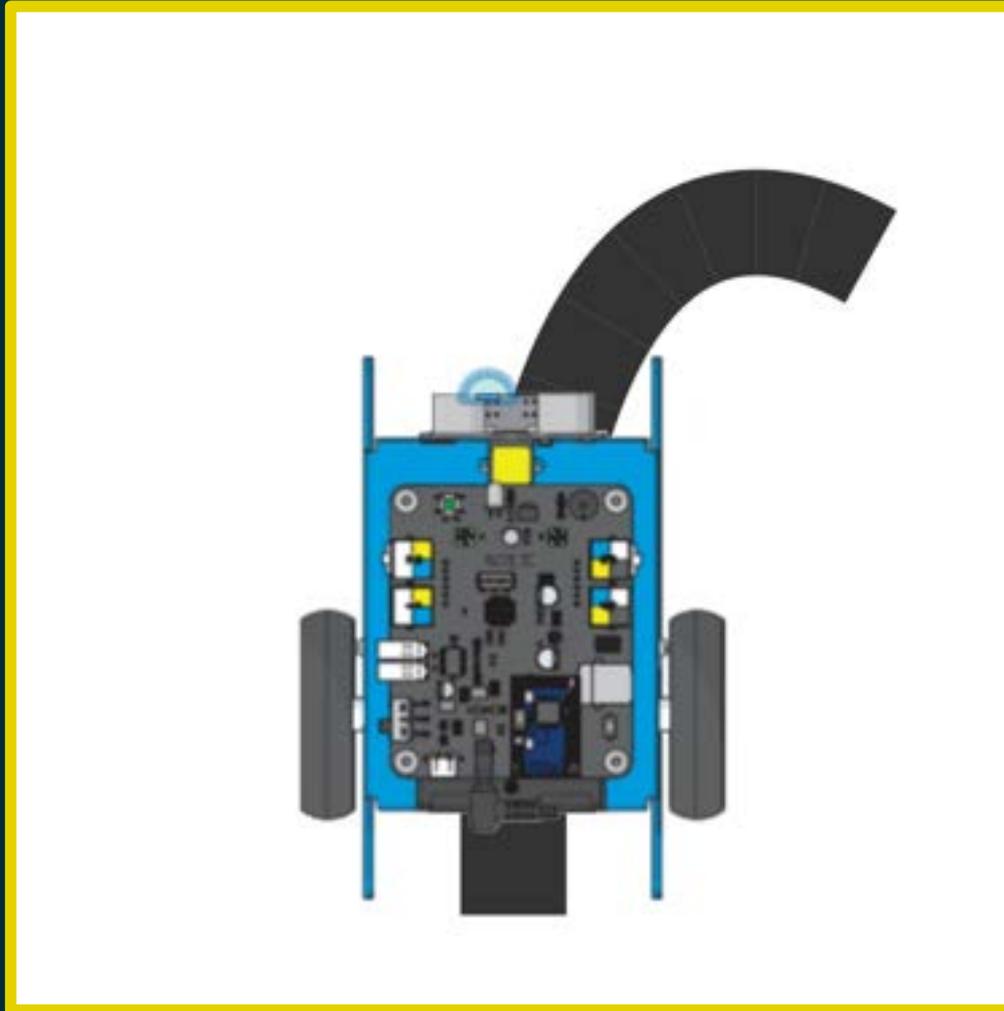
# Valore del sensore di linea



# Valore del sensore di linea



# Valore del sensore di linea



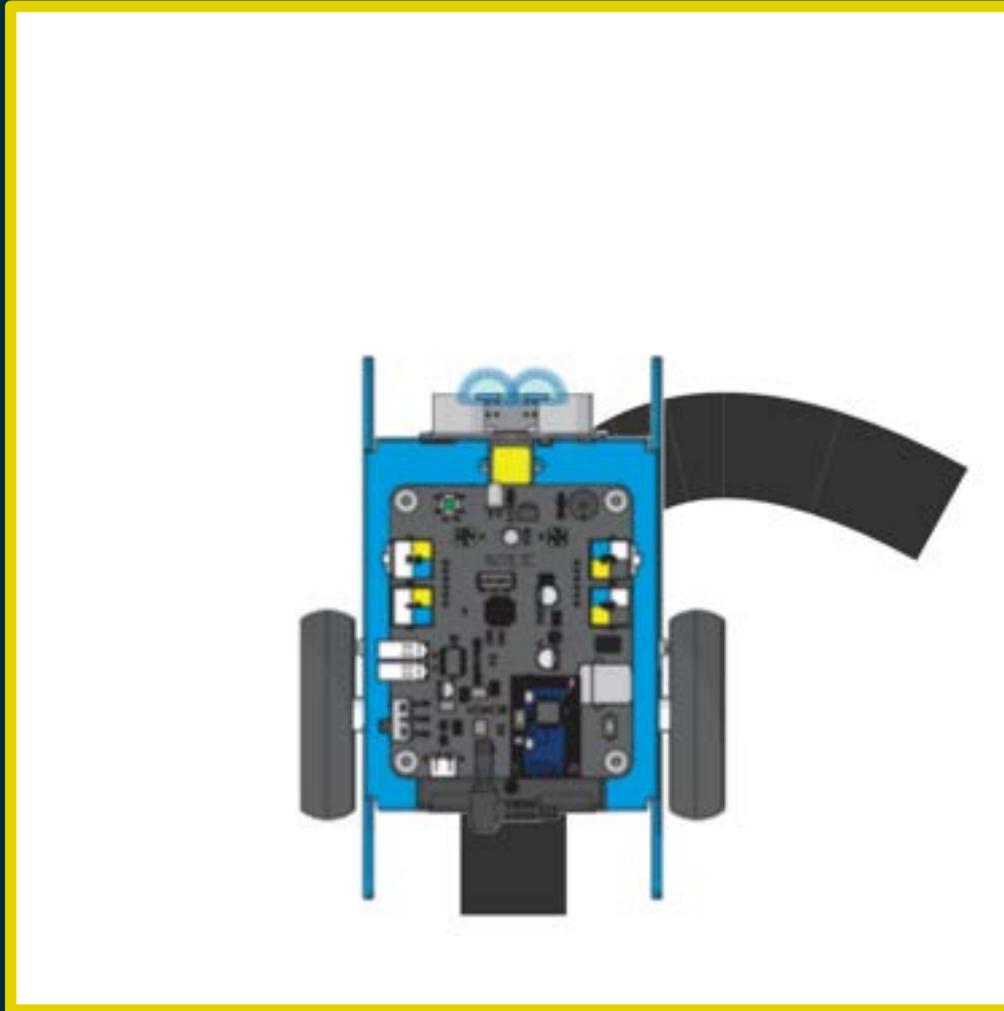
# Valore del sensore di linea

S1 fuori dalla linea  
S2 sulla linea

valore del sensore: 2



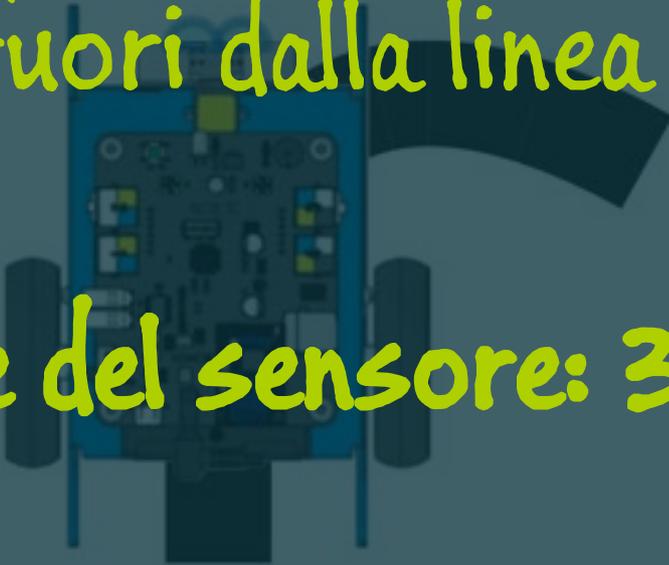
# Valore del sensore di linea



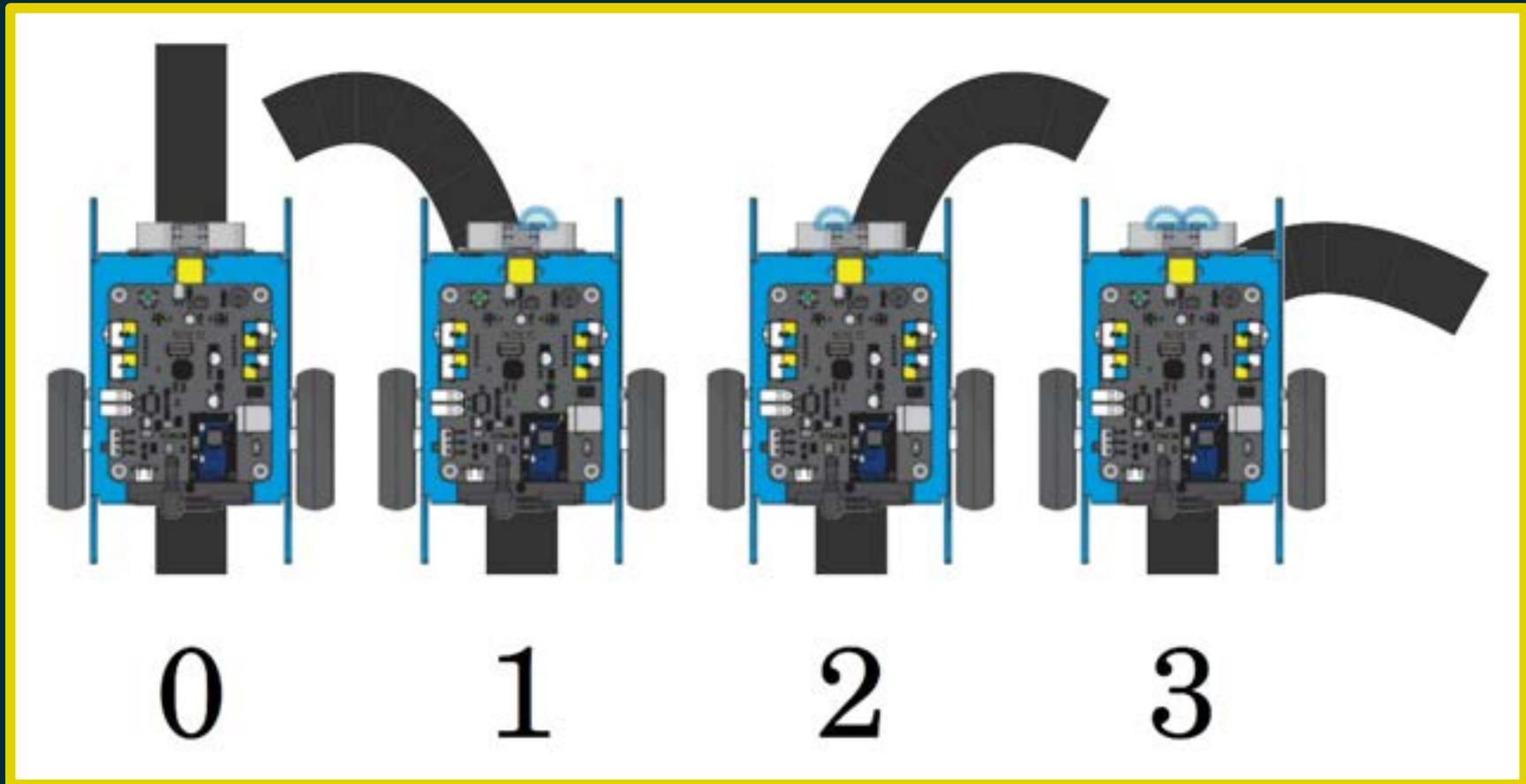
# Valore del sensore di linea

S1 fuori dalla linea  
S2 fuori dalla linea

valore del sensore: 3



# Valore del sensore di linea



Inseguitore di linea

# Inseguitore di linea

Leggiamo il valore registrato dal sensore.

# Inseguitore di linea

Se il valore è zero...

# Inseguitore di linea

Se il valore è zero

allora proseguiamo dritti;

# Inseguitore di linea

Se il valore è zero

allora proseguiamo dritti;

altrimenti, se il valore è uno...

# Inseguitore di linea

Se il valore è zero

allora proseguiamo dritti;

altrimenti, se il valore è uno

allora sterziamo verso sinistra;

# Inseguitore di linea

Se il valore è zero

allora proseguiamo dritti;

altrimenti, se il valore è uno

allora sterziamo verso sinistra;

altrimenti, se il valore è due...

# Inseguitore di linea

Se il valore è zero

allora proseguiamo dritti;

altrimenti, se il valore è uno

allora sterziamo verso sinistra;

altrimenti, se il valore è due

allora sterziamo verso destra;

# Inseguitore di linea

Se il valore è zero

allora proseguiamo dritti;

altrimenti, se il valore è uno

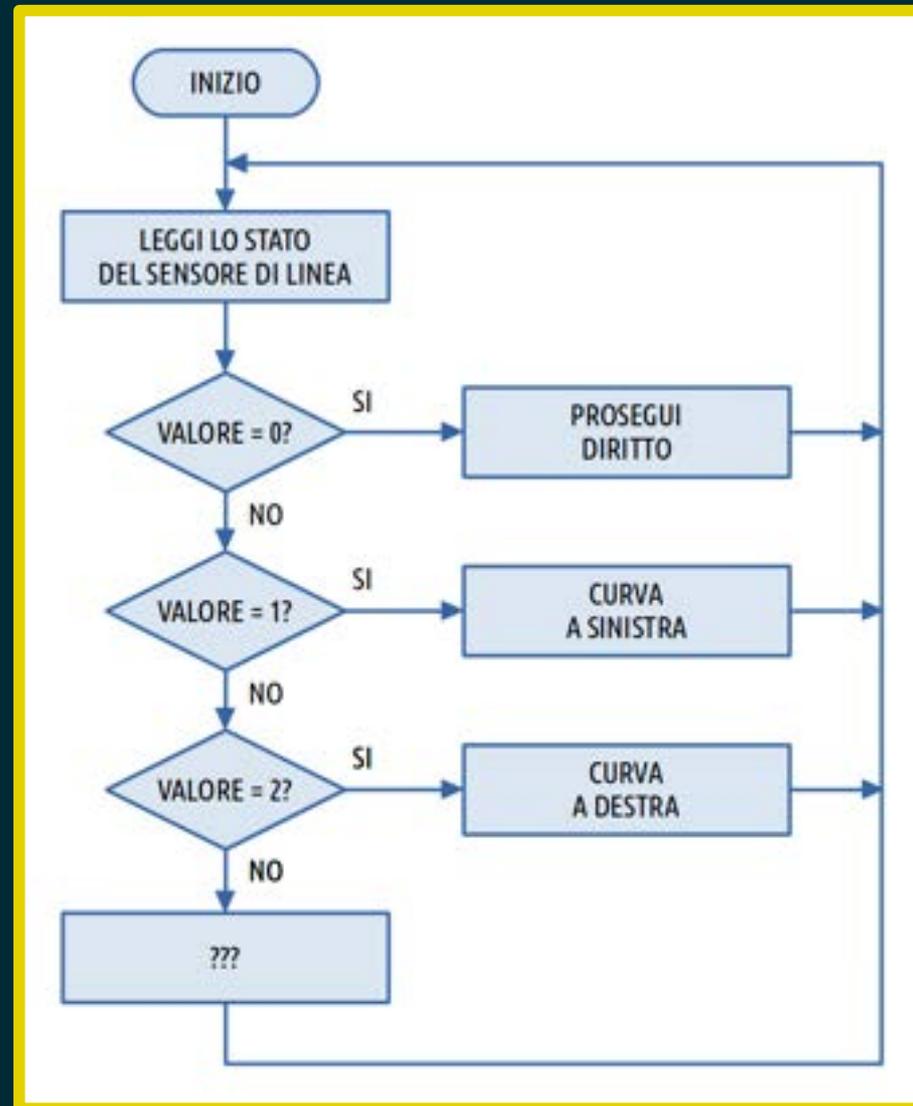
allora sterziamo verso sinistra;

altrimenti, se il valore è due

allora sterziamo verso destra;

altrimenti, se il valore è tre...

# Diagramma di flusso



Inseguitore di linea /2

# Inseguitore di linea /2

(quello pre-programmato in mBot)

# Inseguitore di linea /2

Se almeno un sensore è sulla linea...

# Inseguitore di linea /2

Se almeno un sensore è sulla linea  
si prosegue dritti;

# Inseguitore di linea /2

Se almeno un sensore è sulla linea  
si prosegue dritti;  
altrimenti...

# Inseguitore di linea /2

Se almeno un sensore è sulla linea

si prosegue dritti;

altrimenti

si sterza dalla parte del sensore che è rimasto fuori dalla linea per più tempo.

# Inseguitore di linea /2

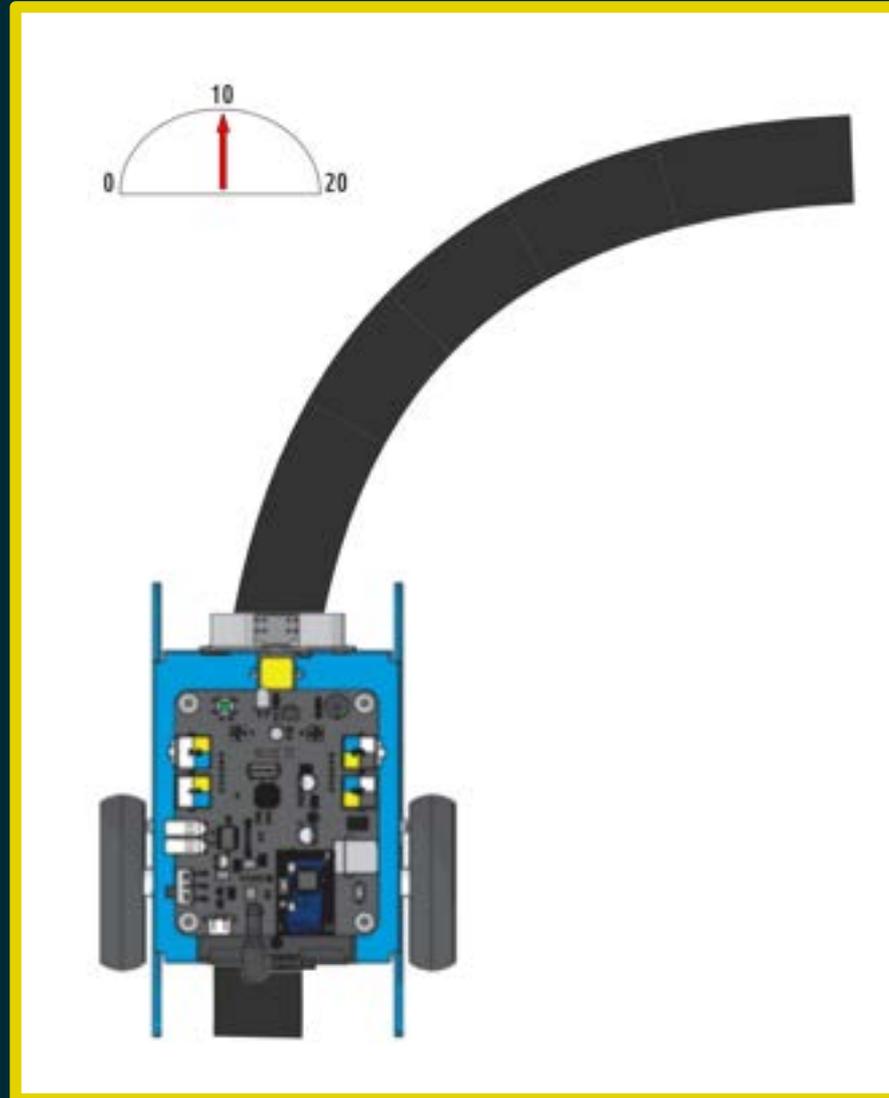
Serve una forma di memoria per tener traccia dello stato passato dei sensori...

# Inseguitore di linea /2

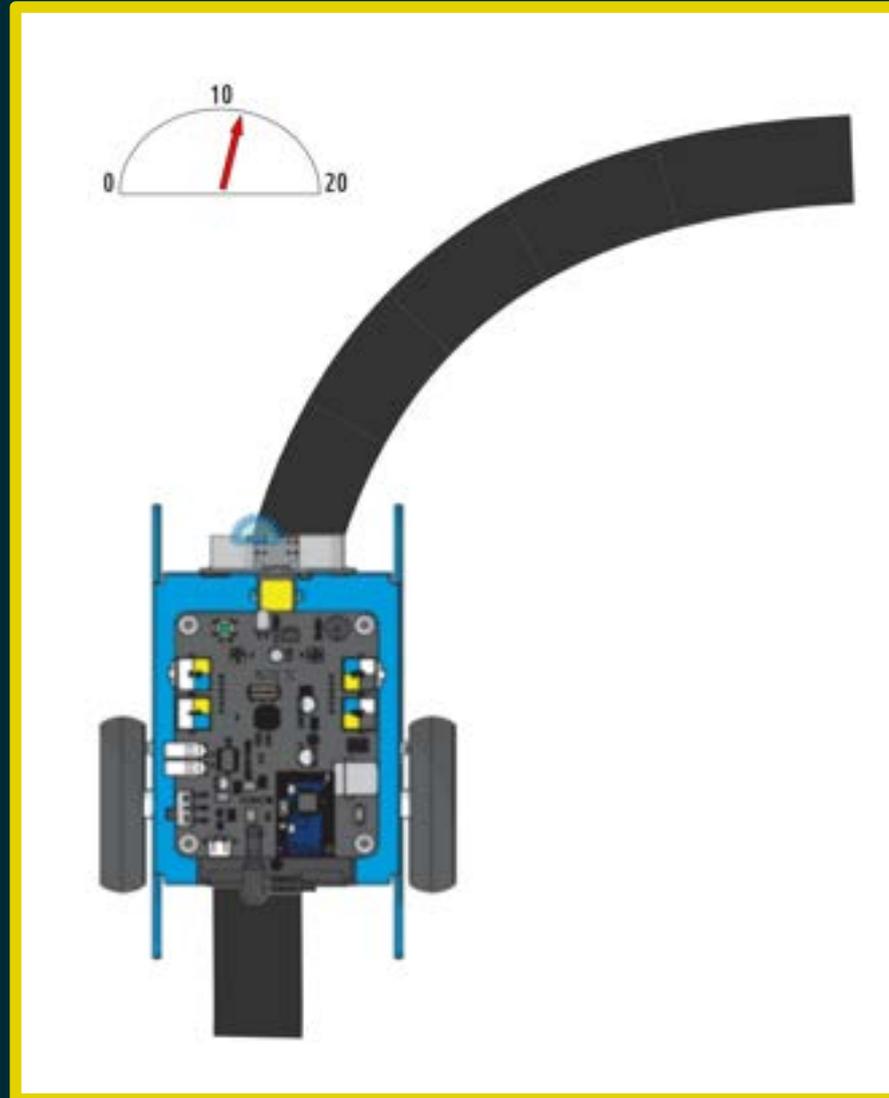
Serve una forma di memoria per tener traccia dello stato passato dei sensori.

Suggerimento: usare una variabile per "contare" le volte che un sensore si è trovato al di fuori della linea..

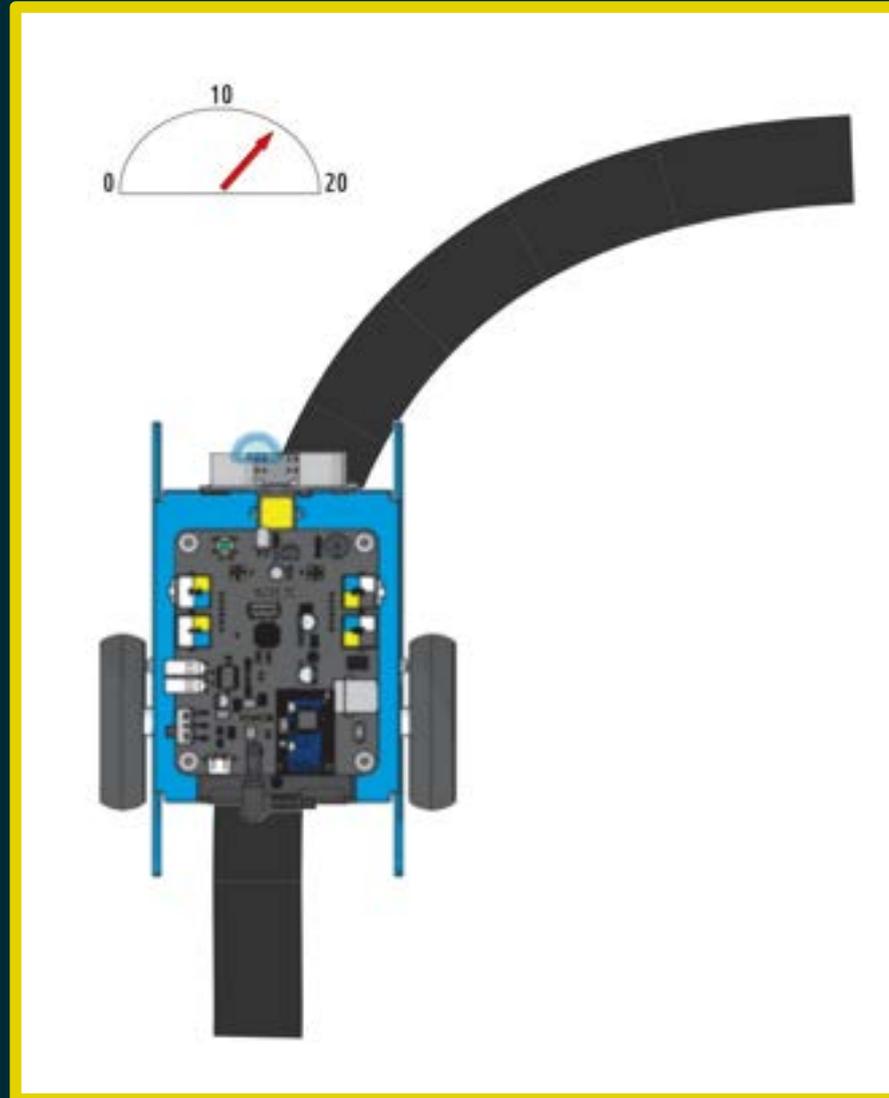
# Inseguitore di linea /2



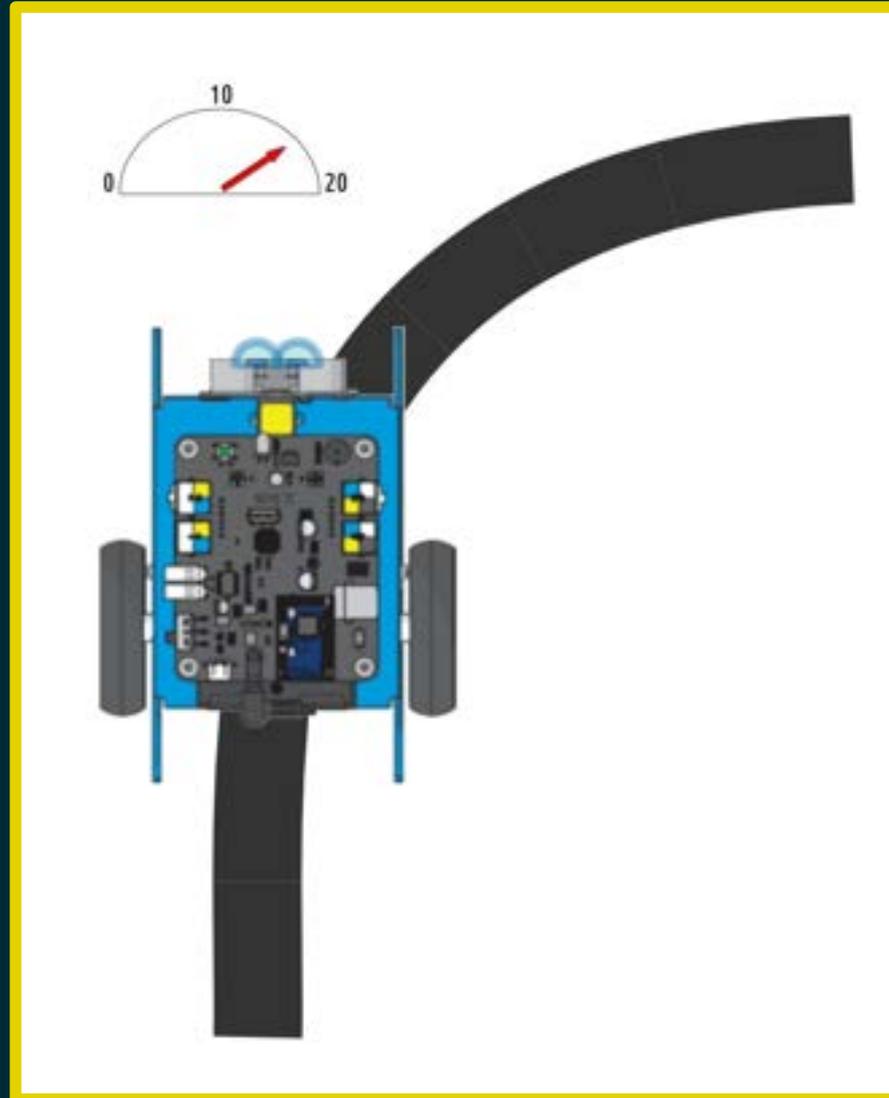
# Inseguitore di linea /2



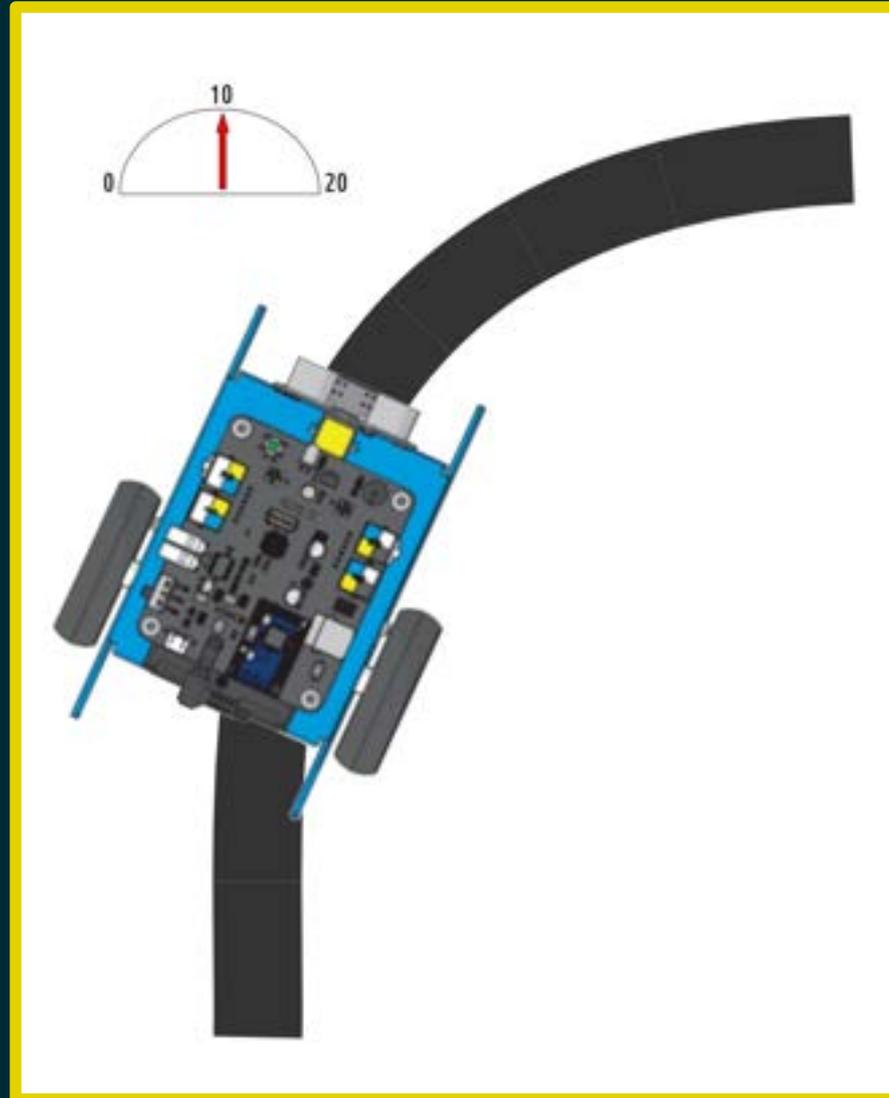
# Inseguitore di linea /2



# Inseguitore di linea /2



# Inseguitore di linea /2



# Sfida

Completare il circuito proposto nel minor tempo possibile. Determinare le velocità e i raggi di curvatura ottimali per tentativi; cambiano a seconda del tracciato!

# Esercitazione

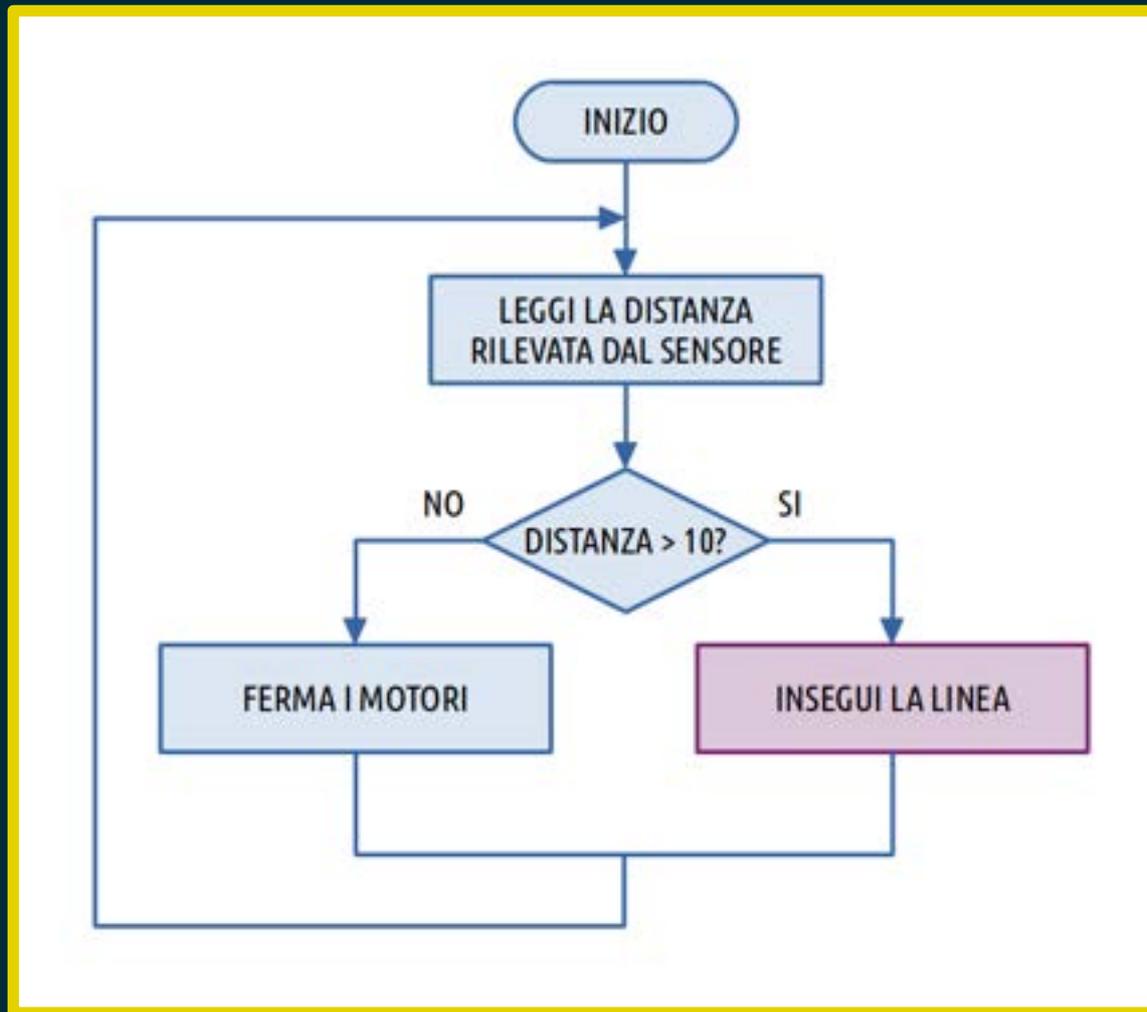
Realizzare un inseguitore di linea che effettua una frenata di sicurezza se il robot che lo precede è troppo vicino.

# Esercitazione

Realizzare un inseguitore di linea che effettua una frenata di sicurezza se il robot che lo precede è troppo vicino.

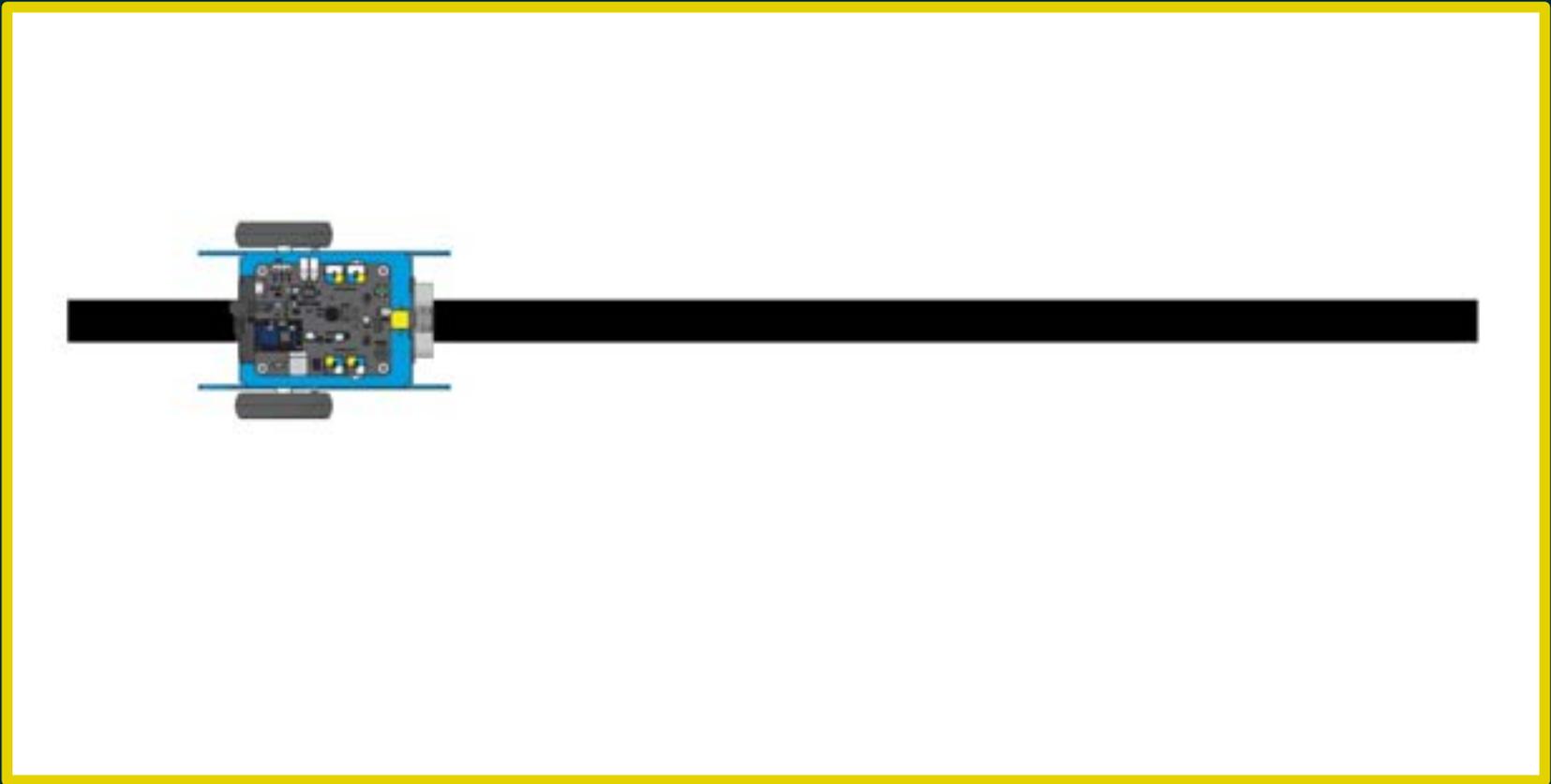
Se la distanza è maggiore di 10cm si segue la linea, in caso contrario ci si ferma.

# Diagramma di flusso

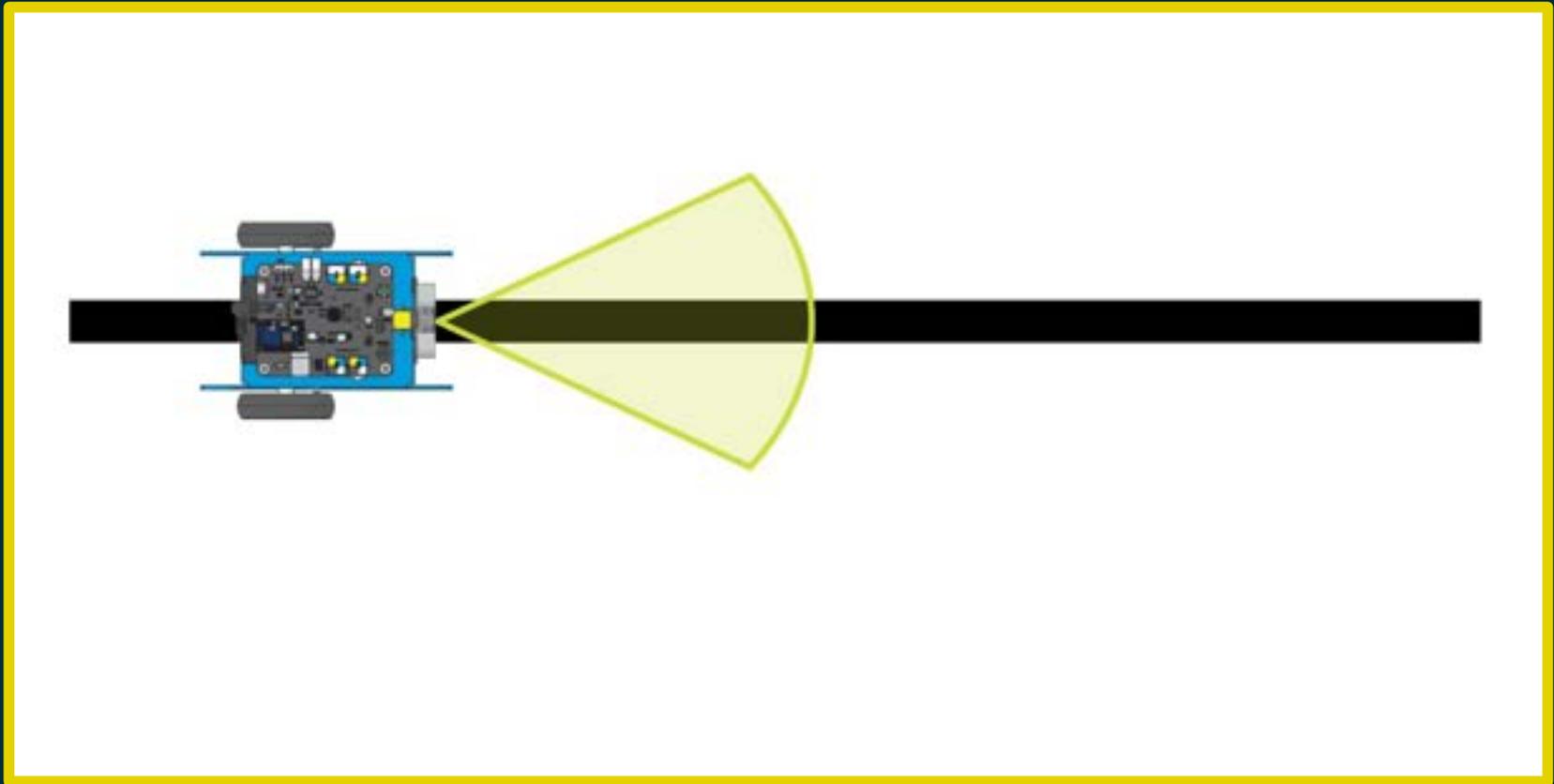


Possibili sviluppi

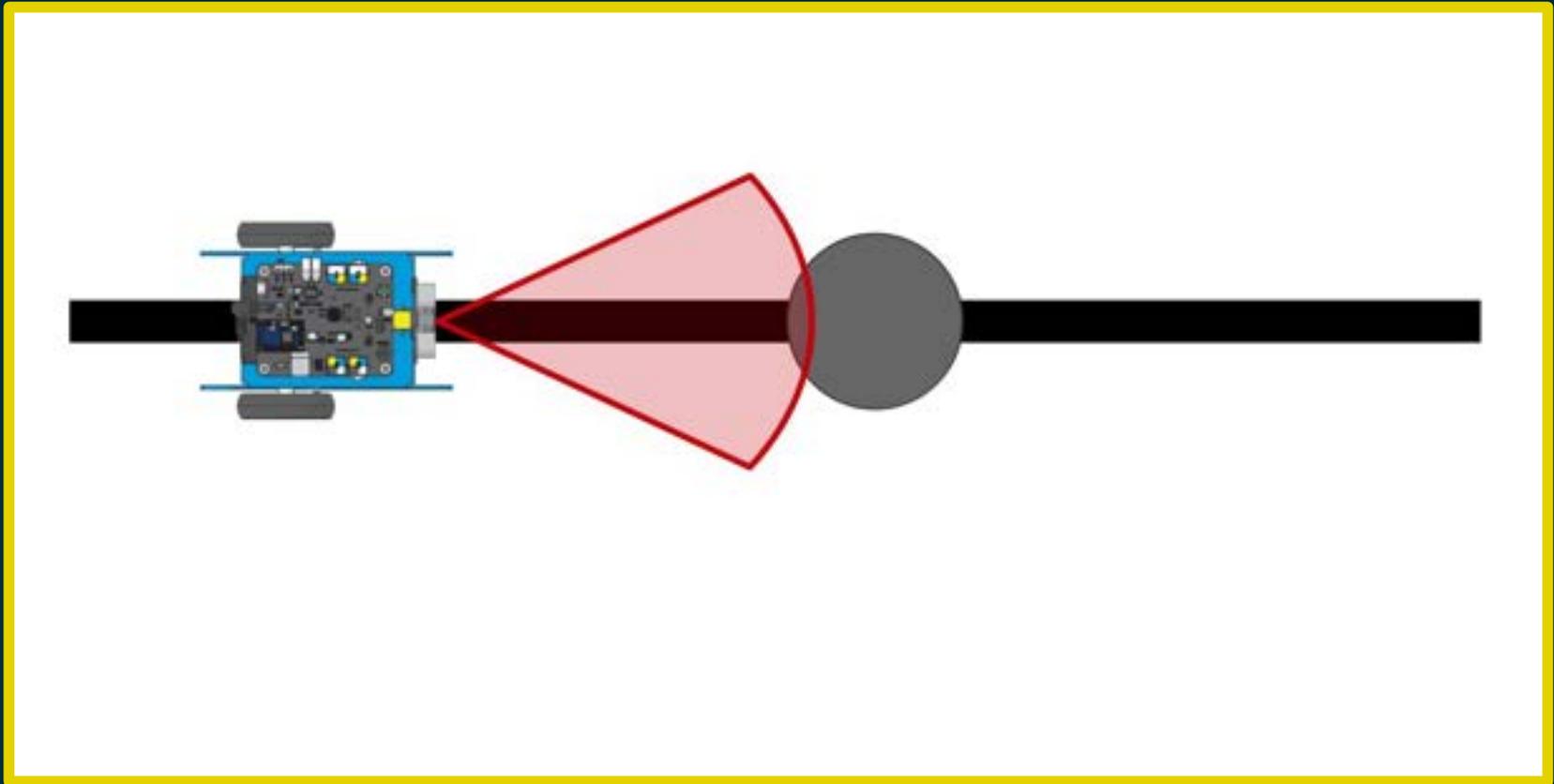
# Inseguitore scansa-ostacoli



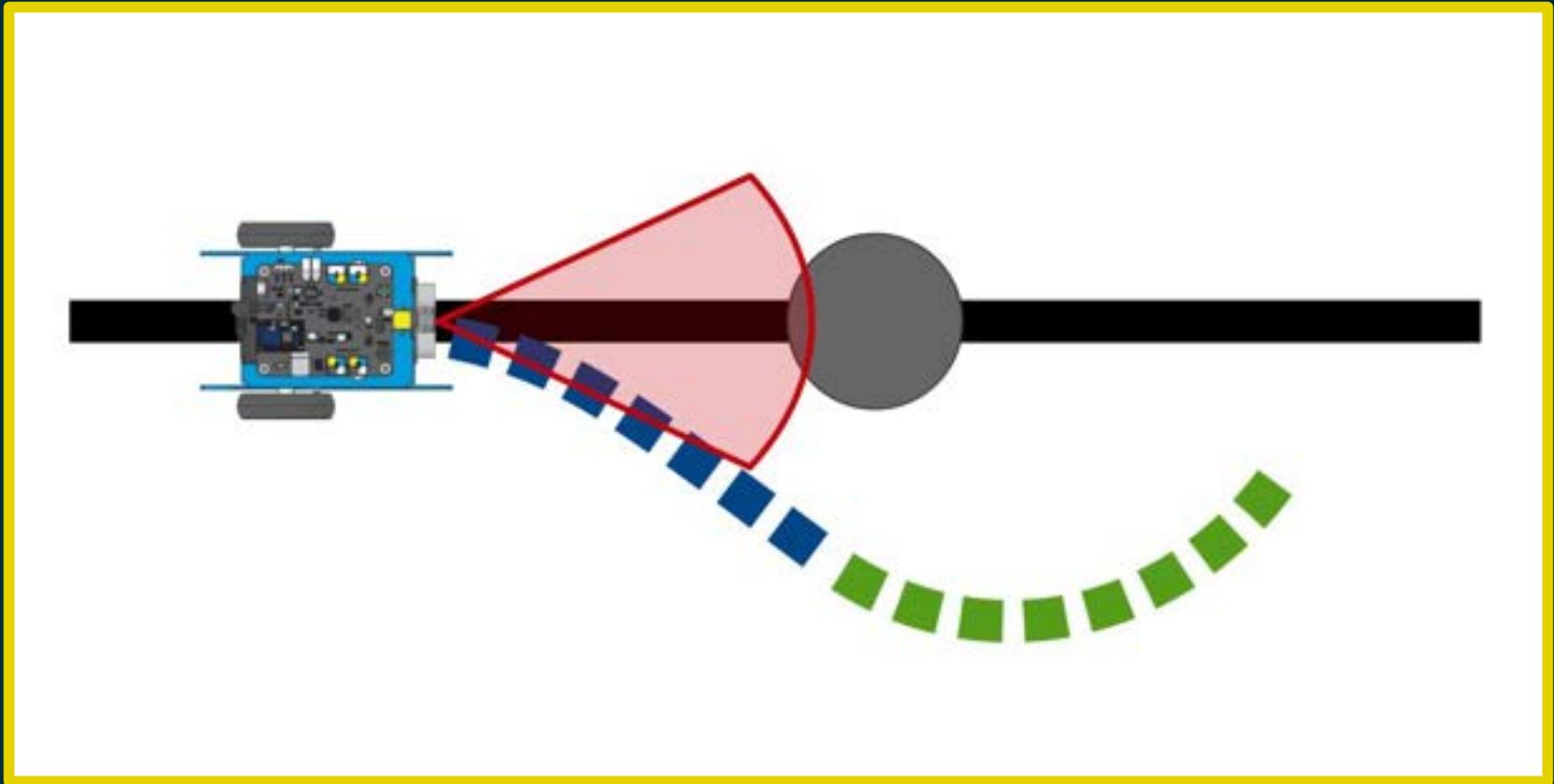
# Inseguitore scansa-ostacoli



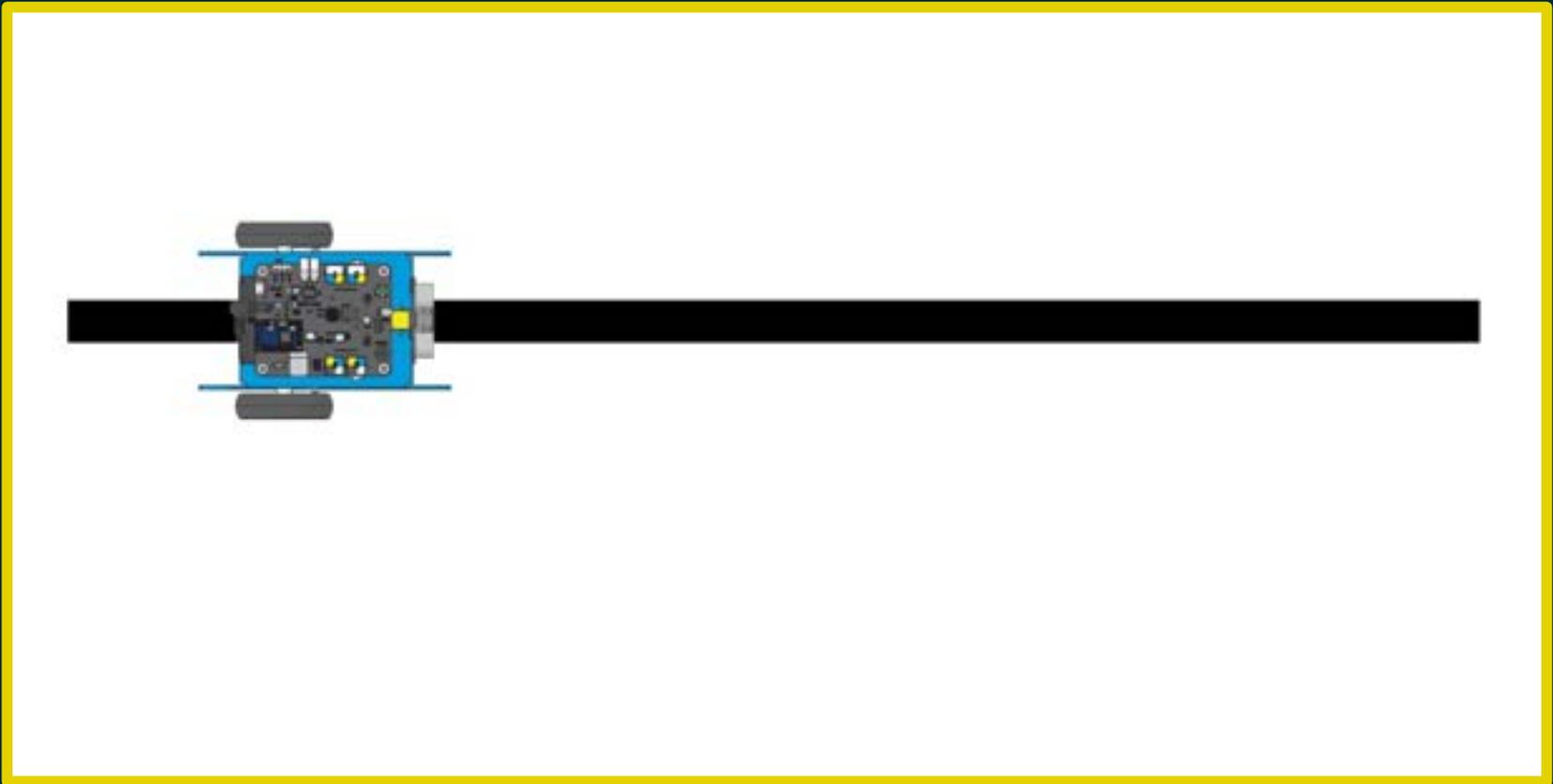
# Inseguitore scansa-ostacoli



# Inseguitore scansa-ostacoli



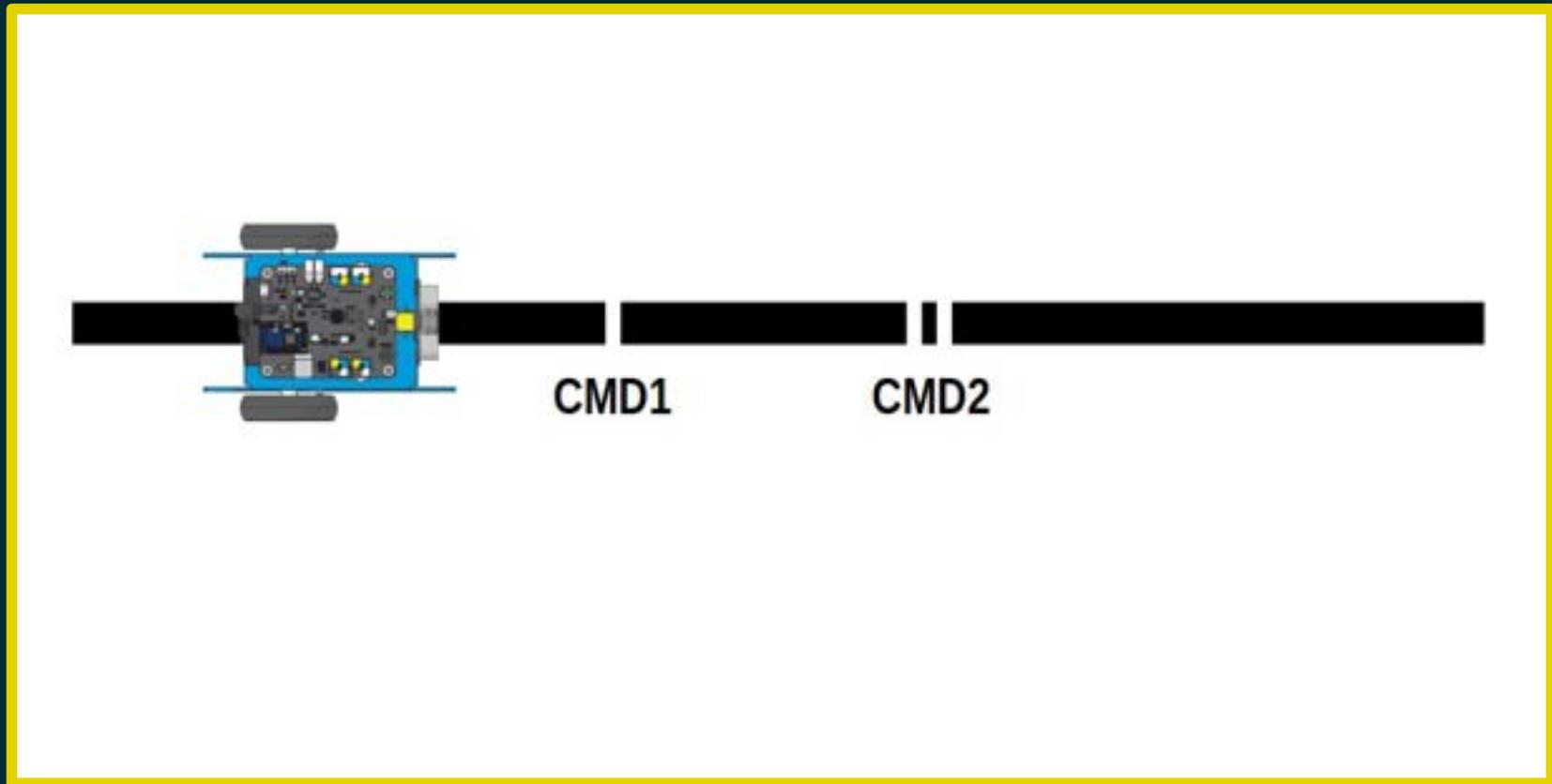
# Comandi codificati su pista



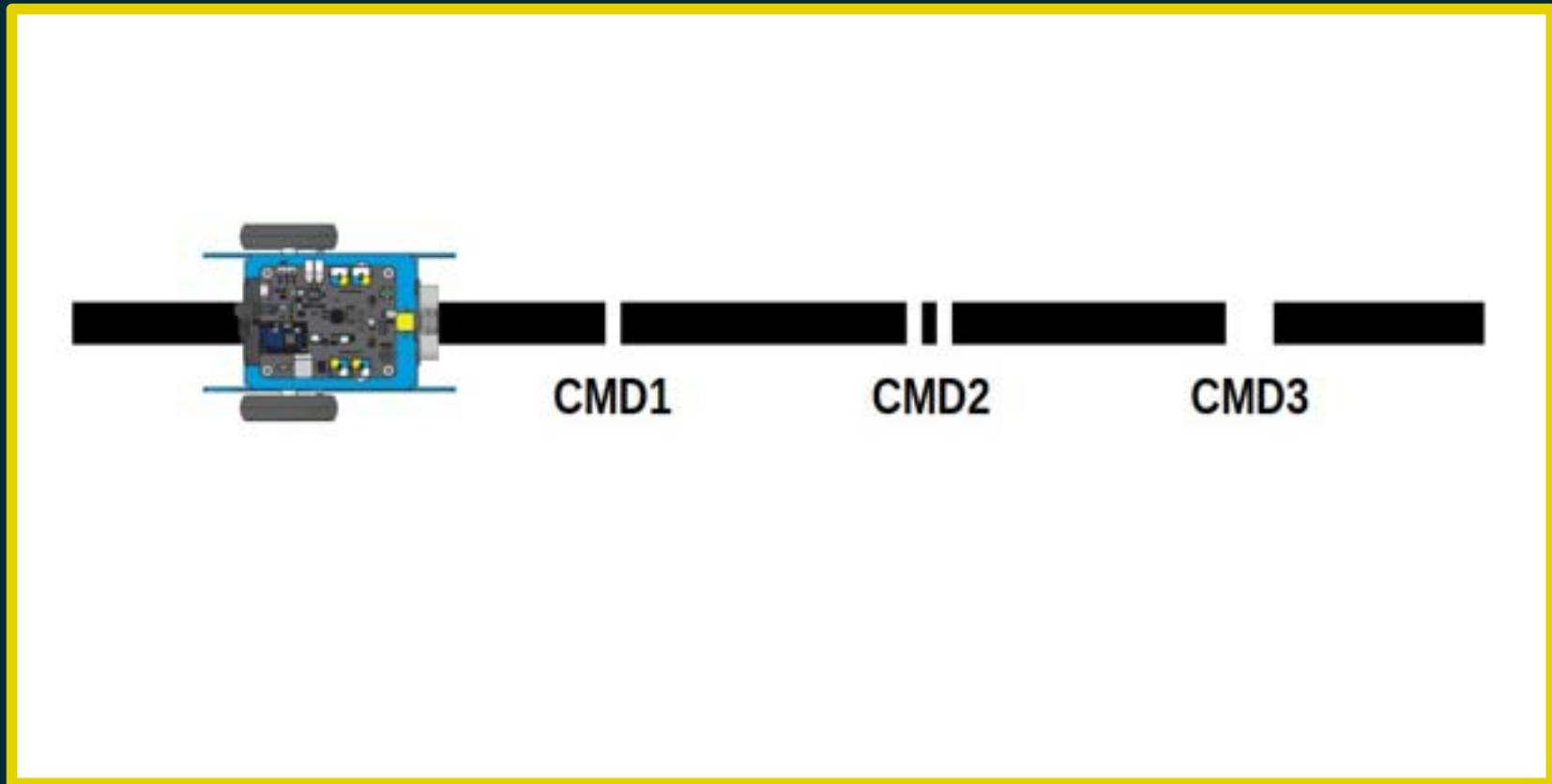
# Comandi codificati su pista



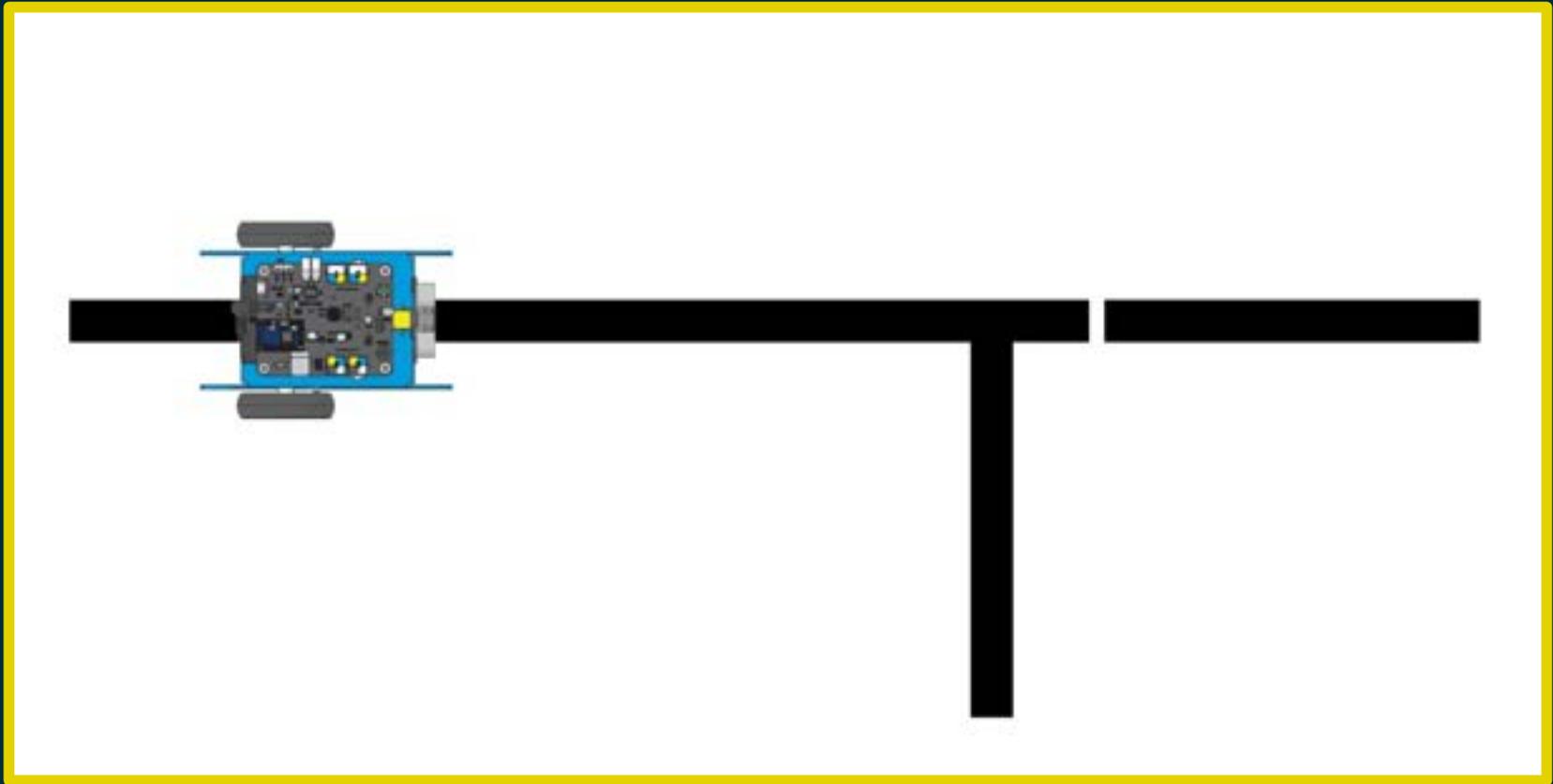
# Comandi codificati su pista



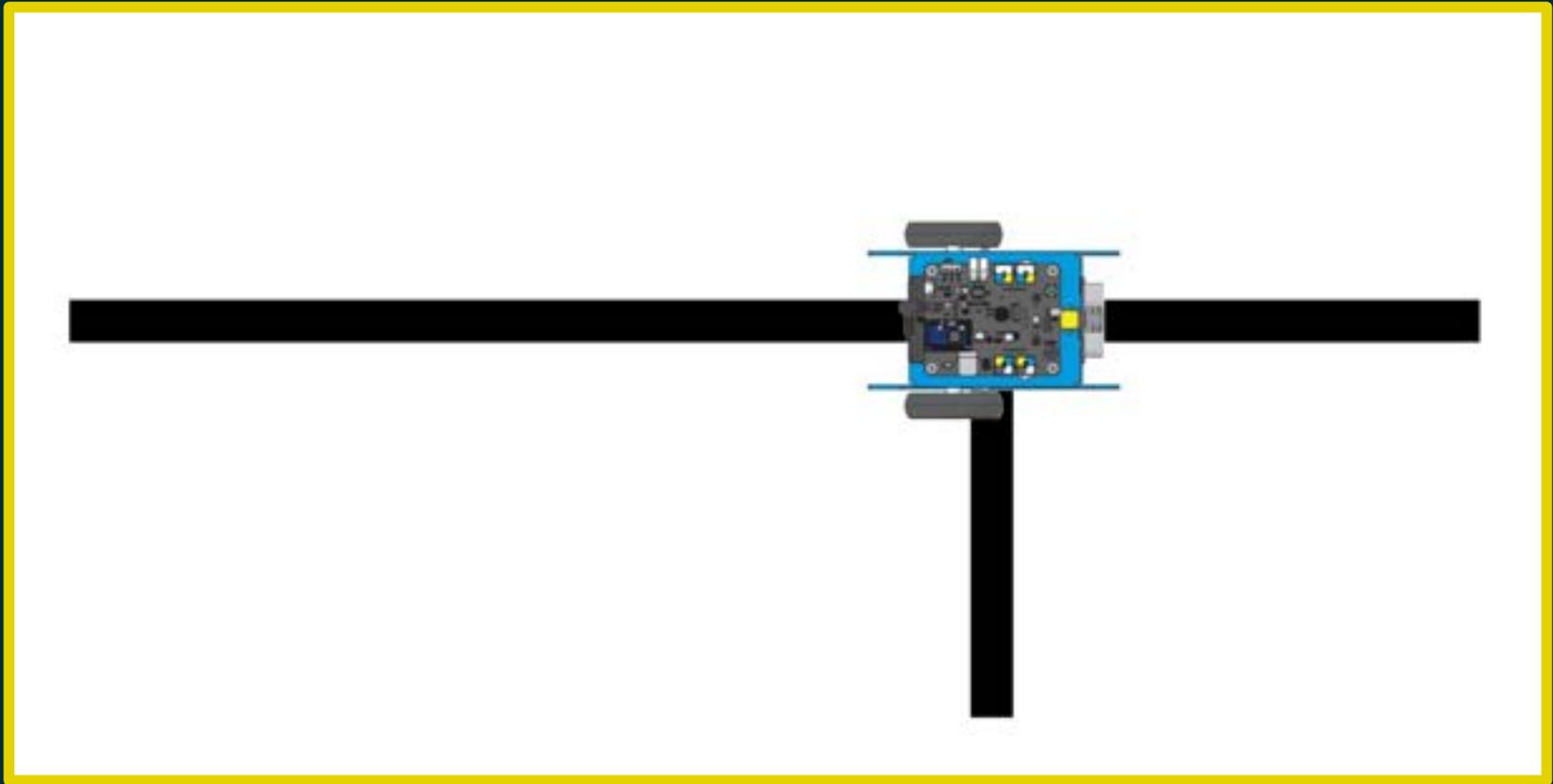
# Comandi codificati su pista



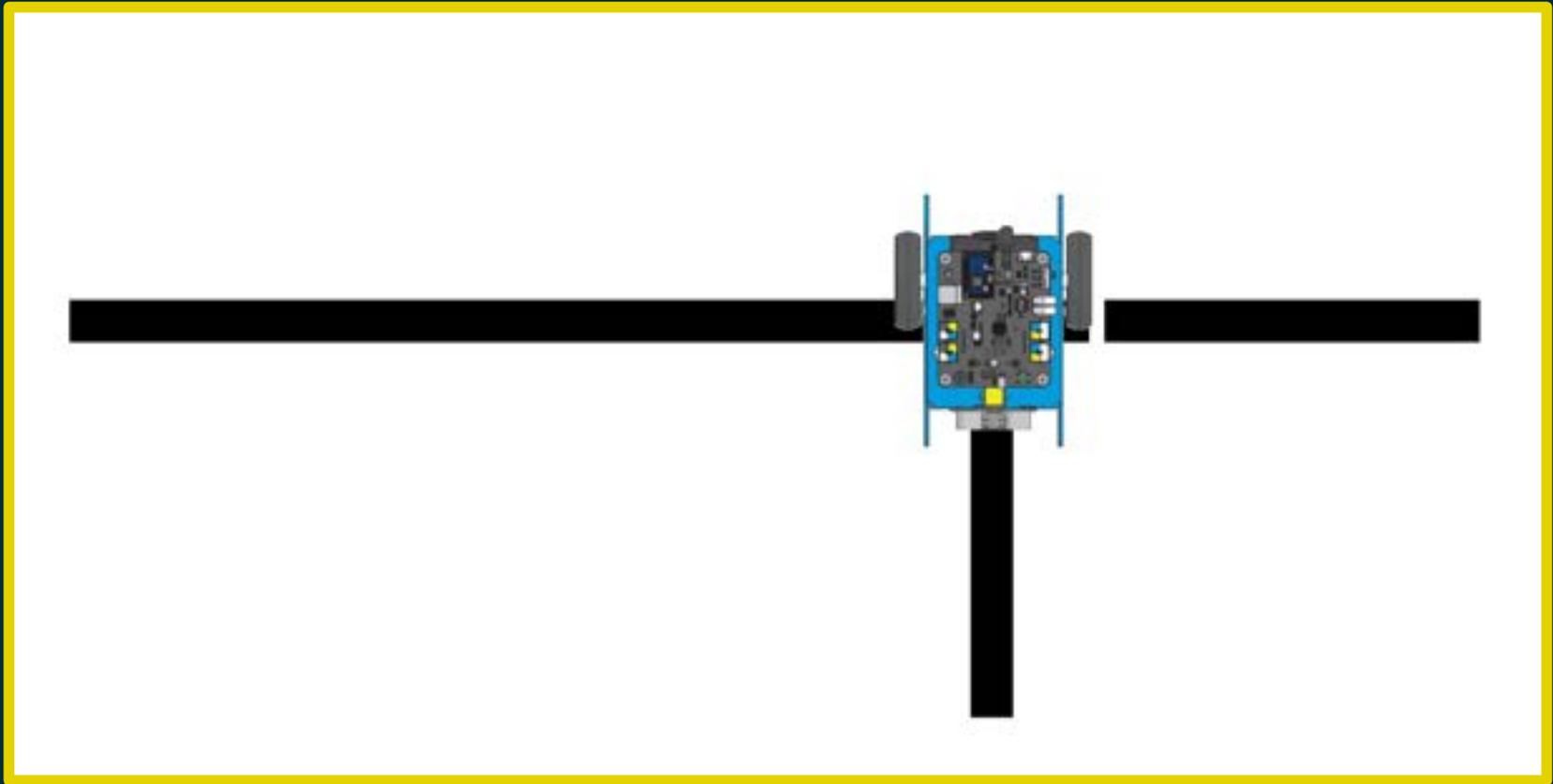
# Comandi codificati su pista



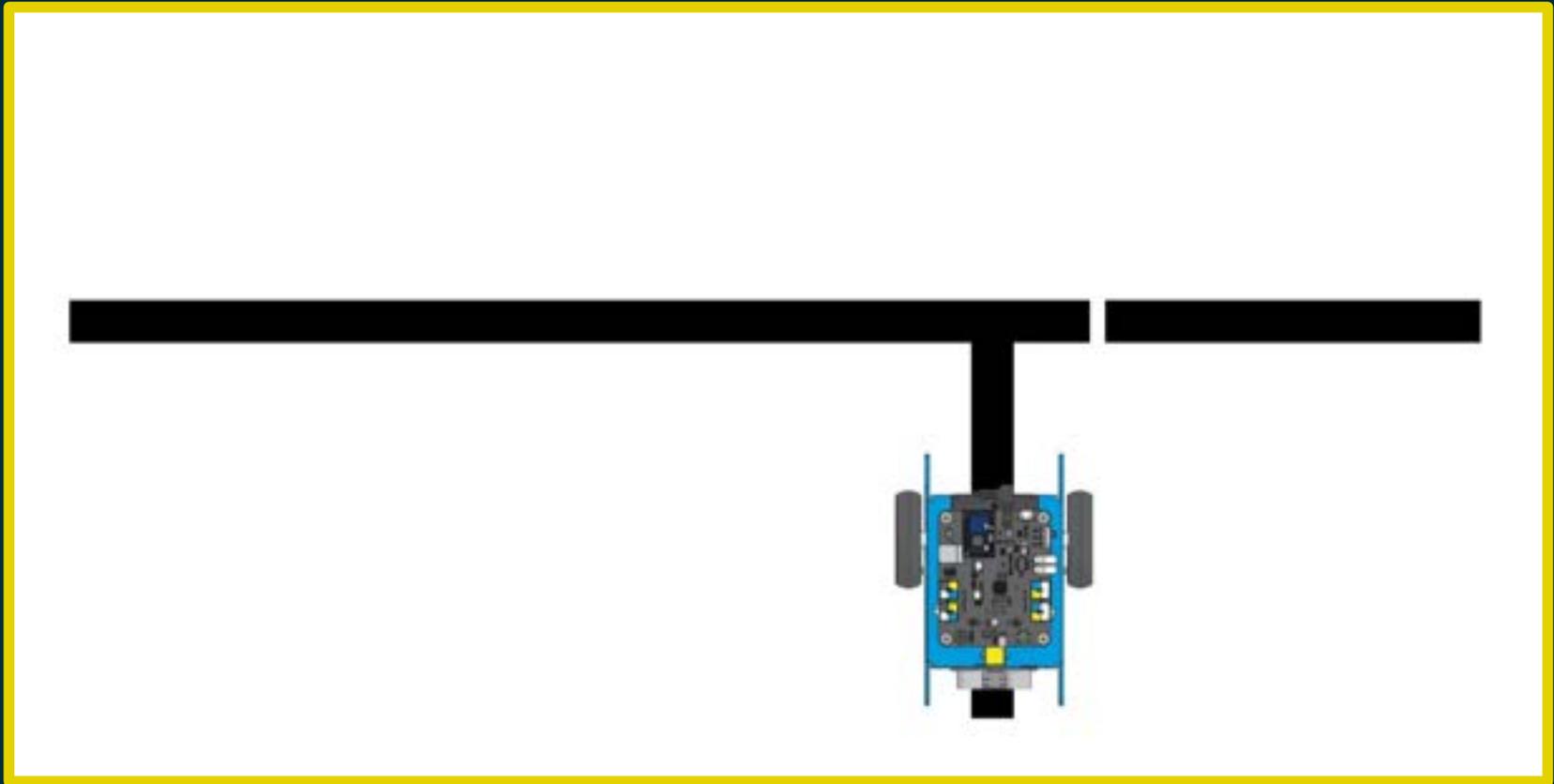
# Comandi codificati su pista



# Comandi codificati su pista

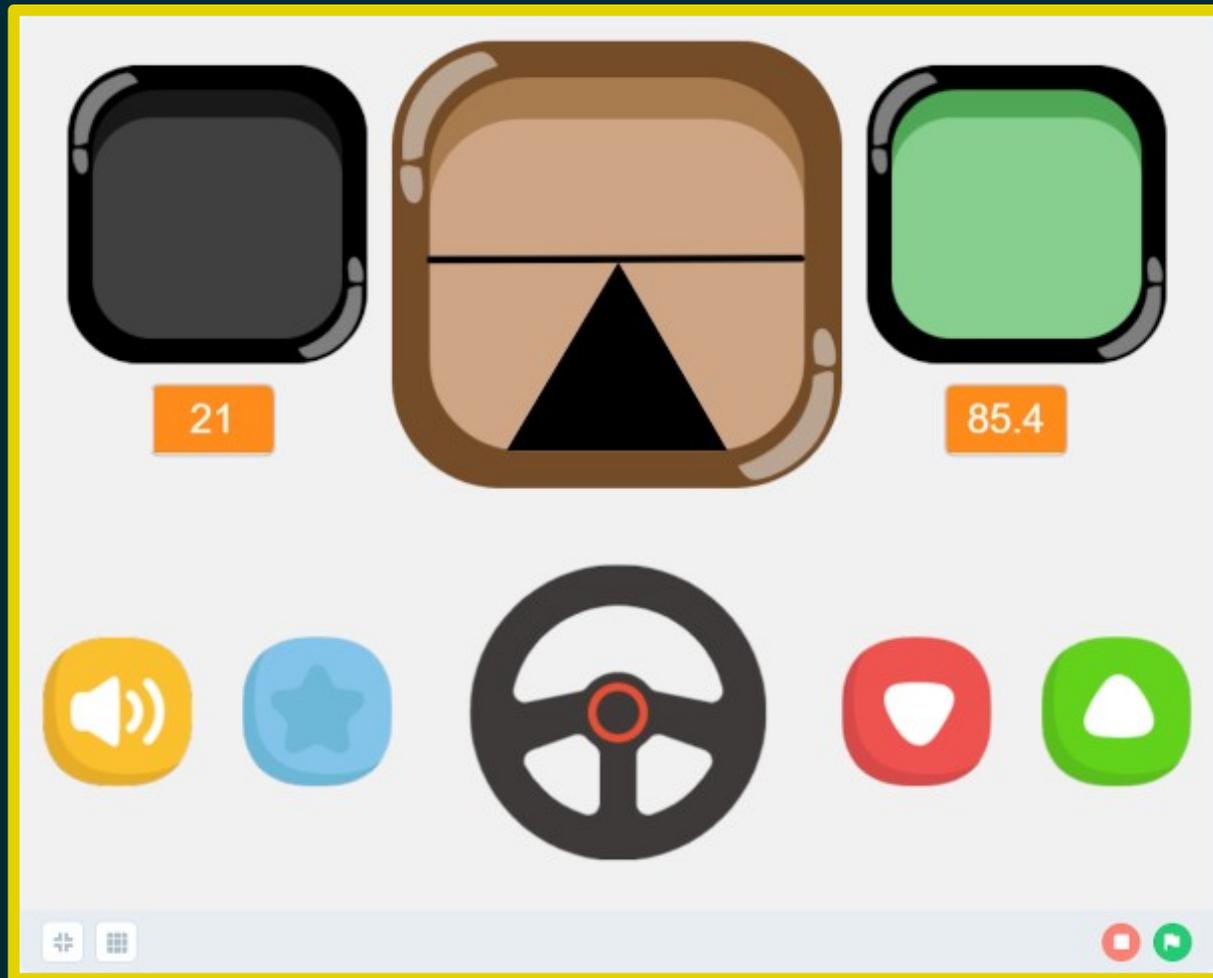


# Comandi codificati su pista

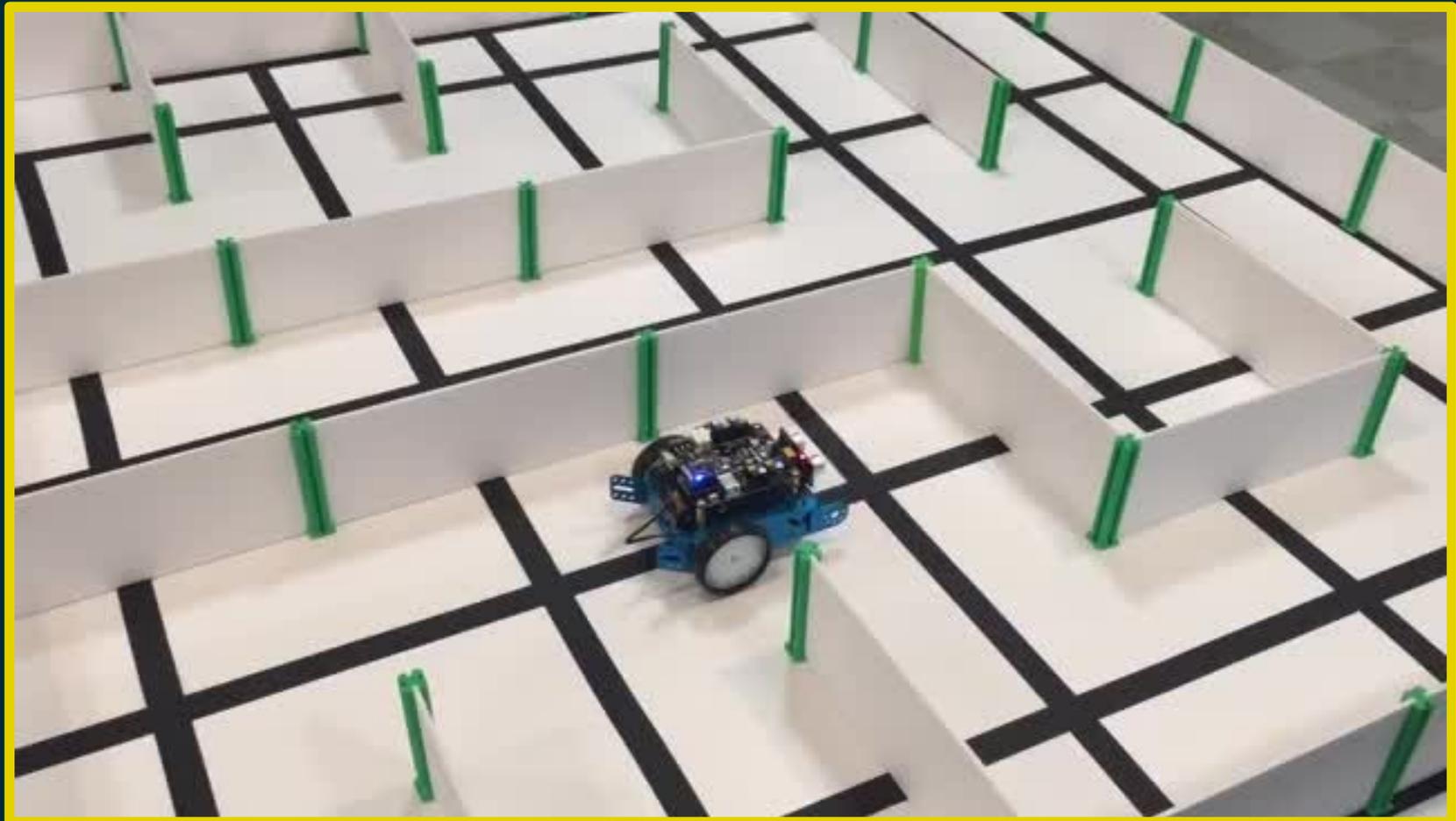


Altri spunti

# Rover



# Uscita dal labirinto



Aspetti non considerati

# Aspetti non considerati

Fotoresistore

# Aspetti non considerati

Fotoresistore

Uso del telecomando

# Aspetti non considerati

Fotoresistore

Uso del telecomando

Modulo Bluetooth/WiFi

# Aspetti non considerati

Fotoresistore

Uso del telecomando

Modulo Bluetooth/WiFi

Modalità "rover" da mBlock

# Aspetti non considerati

Fotoresistore

Uso del telecomando

Modulo Bluetooth/WiFi

Modalità "rover" da mBlock

Intercomunicazione mBot - mBot

# Aspetti non considerati

Fotoresistore

Uso del telecomando

Modulo Bluetooth/WiFi

Modalità "rover" da mBlock

Intercomunicazione mBot - mBot

Pacchetti di estensione

# Conclusioni

# Le tre leggi della robotica

1. Un robot non può recar danno a un essere umano, né permettere che, a causa della propria negligenza, un essere umano patisca danno.
2. Un robot deve sempre obbedire agli ordini degli esseri umani, a meno che contrastino con la Prima Legge.
3. Un robot deve proteggere la propria esistenza, purché questo non contrasti con la Prima o la Seconda Legge.

Grazie!