

# Muoversi con i sensori di Thymio

Robotica Educativa

Camponogara - 19 aprile 2017

Valentino Pietrobon



# Sensori di prossimità

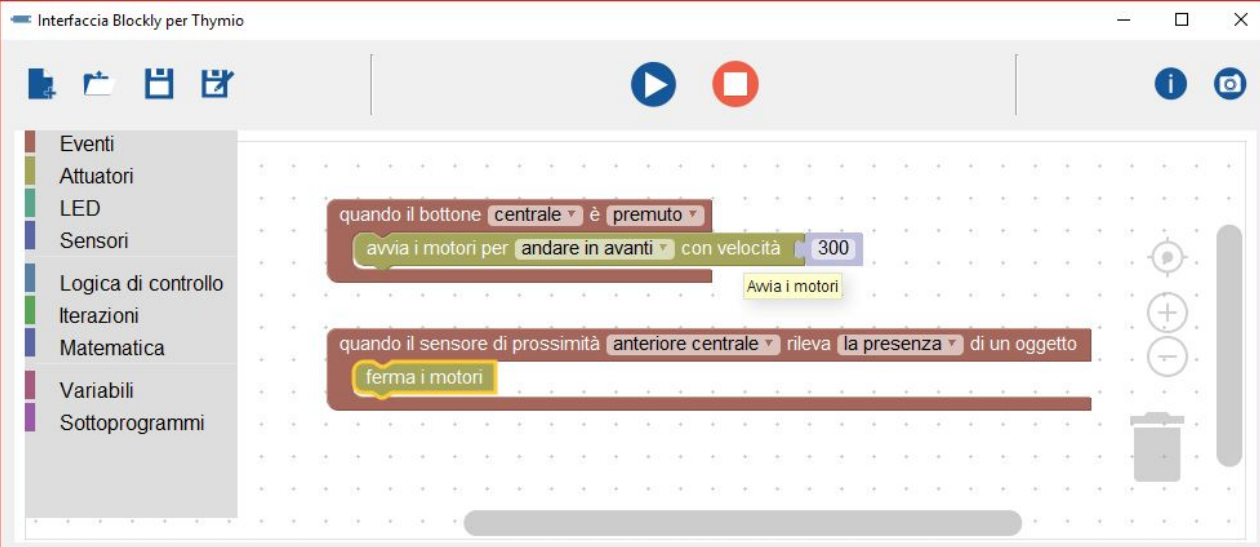
Thymio possiede 5 sensori di prossimità anteriori e 2 sensori di prossimità posteriori che rilevano la presenza di un ostacolo quando esso si trova a una decina di centimetri da Thymio.



# Come evitare gli ostacoli

La presenza di ben 5 sensori anteriori permette a Thymio non solo di evitare ostacoli ma anche di girare in modo opportuno per poterli evitare.

Iniziamo con un semplice esempio che ferma Thymio davanti ad un ostacolo.



The screenshot shows the 'Interfaccia Blockly per Thymio' window. The interface includes a top toolbar with icons for file operations, a play button, a stop button, and help/info icons. On the left, a category palette lists various blocks: Eventi, Attuatori, LED, Sensori, Logica di controllo, Iterazioni, Matematica, Variabili, and Sottoprogrammi. The main workspace contains two event-driven code blocks:

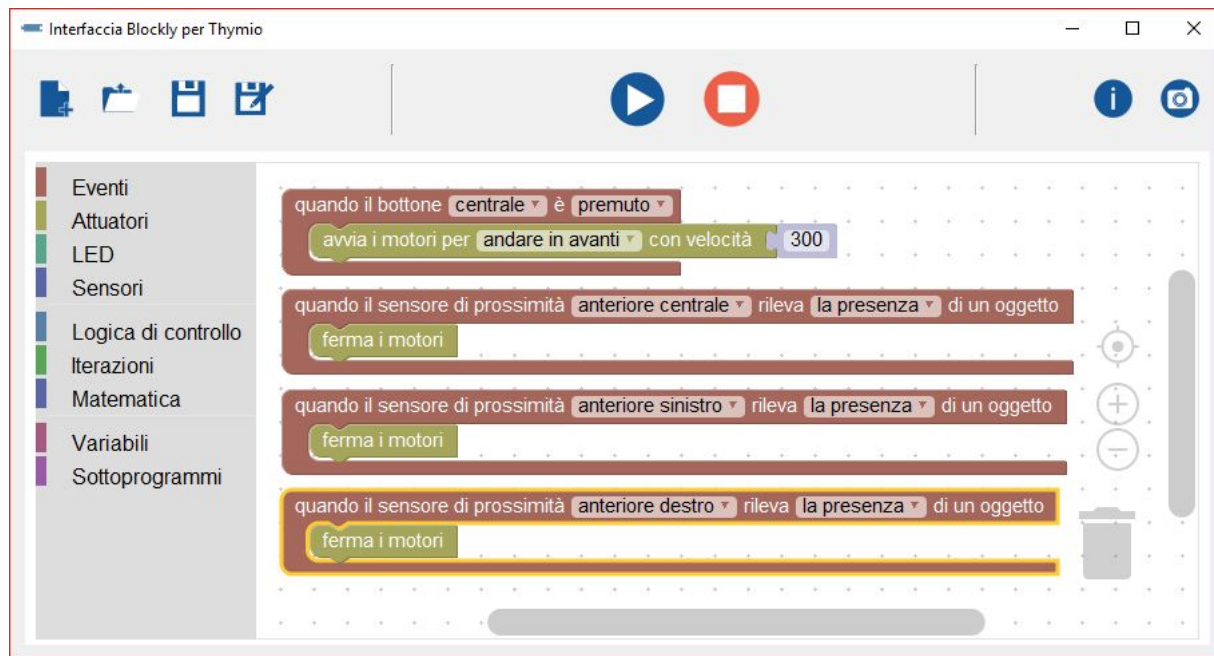
- The first block is an event trigger: "quando il bottone centrale è premuto". It is connected to an action block: "avvia i motori per andare in avanti con velocità 300".
- The second block is another event trigger: "quando il sensore di prossimità anteriore centrale rileva la presenza di un oggetto". It is connected to an action block: "ferma i motori".

On the right side of the workspace, there are several utility icons: a target icon, a plus sign, a minus sign, and a trash can icon.

# Come evitare gli ostacoli

Il programma proposto ferma Thymio a 10 centimetri dall'ostacolo.

Se però l'ostacolo non è perfettamente davanti al sensore centrale, il robot non si ferma. Ecco che diventa importante utilizzare anche i sensori laterali.



# Come evitare ostacoli laterali

Con questo programma si evitano gli ostacoli laterali e si prosegue fino ad un ostacolo anteriore.

Il passo sarà breve per avere un programma che fa tornare indietro Thymio quando trova un ostacolo anteriore e ferma Thymio con un pulsante.

The screenshot shows the 'Interfaccia Blockly per Thymio' window. The interface includes a toolbar with icons for file operations and execution (play and stop buttons). A sidebar on the left lists categories: Eventi, Attuatori, LED, Sensori, Logica di controllo, Iterazioni, Matematica, Variabili, and Sottoprogrammi. The main workspace contains five logic blocks:

- quando il bottone **avanti** è premuto → avvia i motori per **andare in avanti** con velocità **100**
- quando il sensore di prossimità **anteriore sinistro** rileva **la presenza** di un oggetto → avvia i motori per **ruotare in senso orario** con velocità **100** (Reagire a un evento di prossimità)
- quando il sensore di prossimità **anteriore destro** rileva **la presenza** di un oggetto → avvia i motori per **ruotare in senso antiorario** con velocità **100**
- quando il sensore di prossimità **anteriore sinistro** rileva **l'assenza** di un oggetto → avvia i motori per **andare in avanti** con velocità **100**
- quando il sensore di prossimità **anteriore destro** rileva **l'assenza** di un oggetto → avvia i motori per **andare in avanti** con velocità **100**

# Come evitare che parta da solo

Interfaccia Blockly per Thymio

The image shows the Blockly interface for Thymio, a visual programming environment. The interface includes a toolbar with icons for file operations, a play button, a stop button, and help/info icons. A left sidebar lists categories: Eventi, Attuatori, LED, Sensori, Logica di controllo, Iterazioni, Matematica, Variabili, and Sottoprogrammi. The main workspace contains several logic blocks:

- quando il bottone avanti è premuto**:
  - imposta la variabile stato a 1
  - avvia i motori per andare in avanti con velocità 100
- quando il sensore di prossimità anteriore destro rileva l'assenza di un oggetto**:
  - se stato = 1
    - avvia i motori per andare in avanti con velocità 100
- quando il sensore di prossimità anteriore sinistro rileva la presenza di un oggetto**:
  - se stato = 1
    - avvia i motori per ruotare in senso orario con velocità 100
- quando il sensore di prossimità anteriore destro rileva la presenza di un oggetto**:
  - se stato = 1
    - avvia i motori per ruotare in senso antiorario con velocità 100
- quando il sensore di prossimità anteriore sinistro rileva l'assenza di un oggetto**:
  - se stato = 1
    - avvia i motori per andare in avanti con velocità 100
- quando il bottone centrale è premuto**:
  - imposta la variabile stato a 0
  - ferma i motori

Additional elements include a 'Reagire a un evento di prossimità' block and a trash icon on the right side of the workspace.

# Per andare avanti dritto evitando gli ostacoli

Un'idea potrebbe essere quella di utilizzare una variabile per indicare se si sta ruotando a sinistra o a destra da utilizzare quando ruotare, e mentre si ruota attivare il contatore per 100 millisecondi. Ogni volta che il contatore si azzerà, si andrà a incrementare o a decrementare una variabile che tiene conto di quanto tempo il robotino sta ruotando da una parte e si fa ripartire il contatore.

Appena il robotino può andare dritto si vedrà se bisogna farlo ruotare un po' a seconda di quanto è ruotato in precedenza e poi si farà andare dritto.

Una ottimizzazione per farlo ruotare quando davanti c'è un ostacolo è guardare i sensori di prossimità anteriori per decidere se farlo ruotare a destra o a sinistra.