

# RoboGino

## Unità Didattiche



© Copyright By Nabla tecnologie S.R.L.

© Copyright By Nabla tecnologie S.R.L.

**RoboGino - Unità didattiche**

# **IMPORTANTE**

*Questo è un'anteprima del libro “RoboGino – Unità Didattiche”.*

*Nella versione definitiva del libro saranno presenti altri esempi e unità pratiche, corredate anche da video tutorial.*

*Inoltre sul sito [www.vexrobot.it](http://www.vexrobot.it) potrete trovare altri unità didattiche realizzate da Nabla Tecnologie e da altri docenti, e sarà possibile anche condividere i vostri lavori.*

Tutti gli esempi e le unità didattiche presenti in questo libro possono essere eseguiti con l'utilizzo del *Competition Kit VEX IQ* di VEX Robotics, acquistabile su MEPA.

€ 470,00 (iva esclusa)



**acquistinretepa.it**  
Il Portale degli acquisti della Pubblica Amministrazione

Codice MEPA  
228-3670-ITA-PLUS

### Scheda tecnica

Il kit è composto da più di 850 pezzi con cui costruire qualsiasi tipo di robot autonomo o controllato.

Il kit comprende:

- n° 1 unità programmabile dotata di 12 porte a cui poter connettere indifferentemente motori o sensori, schermo LCD utilizzabile attraverso quattro pulsanti;
- n° 4 motori con processore integrato, encoder di quadratura e sistema di monitoraggio di corrente che lavorano sinergicamente per permettere all'utente di monitorare tramite l'unità centrale programmabile gli stati delle variabili velocità, direzione, tempo, angolo di rotazione);
- n° 7 Sensori (due di pressione, due per il riconoscimento tocco con illuminazione led, distanza, riconoscimento colore, riconoscimento posizione angolare);
- n° 1 Joystick wireless;
- n° 2 adattatori wireless a 2,4Ghz;
- n° 1 batteria Lithium Ion 3,7V 800 mAh per il Joystick;
- n° 1 batteria Nickel metal Hyride technology 2000 mAh 7,2v per alimentare l'unità programmabile;
- n° 1 Docking station di ricarica per la batteria;
- n° 1 Box contenitore;



Software di programmazione on cloud in cui gli alunni potranno condividere i programmi realizzati;

Completo di tutti i cavi patch necessari.

Sul sito del produttore sono disponibili i disegni tecnici CAD di ogni elemento che compone il kit utilizzabili come base di partenza per

creare modifiche su pezzi originali. Il file CAD può essere inviato alla stampante 3D per la replica di un pezzo originale o per crearne uno modificato.

Il sistema è comprensivo di software proprietario per la modellazione 3D con cui poter progettare virtualmente i modelli di robot e in un secondo momento costruirli o condividere le proprie creazioni sotto forma di istruzioni passo-passo per la costruzione.

Possibilità di integrare accessori da competizione proprietari in grado di aumentare le prestazioni del sistema.

Comprensivo di manuale in italiano, unità didattiche, tappeto con percorso graduato indispensabile per lo svolgimento delle unità didattiche.

Ulteriori informazioni possono essere trovate su

[www.vexrobot.it](http://www.vexrobot.it)



© Copyright By Nabla tecnologie S.R.L.

**RoboGino - Unità didattiche**

# 1. Sommario

Introduzione.....	5
2. Robot, Pensiero computazionale e coding .....	6
2.1. I Robot.....	6
2.2. Il pensiero computazionale.....	6
2.3. Diagramma di flusso .....	7
3. Guida all'assemblaggio del RoboGino.....	10
3.1. Il Robot Brain .....	12
3.2. Il motore di destra .....	13
3.3. Il motore di sinistra .....	13
3.4. Base con i due motori e le ruote.....	15
3.5. Il RoboGino.....	16
3.6. Pronti per partire .....	17
4. La Programmazione .....	17
4.1 Coding.....	17
4.2 VexCode IQBlock.....	17
4.3 Avvio di VexCode IQBlock .....	18
4.4 Apertura nuovo progetto .....	18
4.5 Sezione Devices.....	19
4.7 Salvataggio Configurazione RoboGino con Controller .....	23
4.8 RoboGino Autonomo.....	23
4.9 Primo esempio pratico- "VAIAVANTI" .....	24
4.10 Conclusione.....	0
5. Unità didattiche .....	1
5.1. Imparare a misurare lunghezze ed angoli.....	1
5.2. Il Quadrato .....	4
5.3. Il Cerchio .....	6
5.4. Relazione Spazio Tempo e Velocità .....	9

# Introduzione

Questo manuale vuole essere una guida all'utilizzo didattico della piattaforma VEX IQ ponendo l'attenzione ad un semplice robot, "RoboGino" che potete costruire facilmente seguendo il manuale di costruzione nel Capitolo 3.

Prima di Presentare le attività farò una brevissima introduzione al pensiero computazione e al mondo dei Robot.

Le attività presentate presuppongono che il lettore abbia a portata di mano un Kit VEX IQ. Esistono vari kit VEX IQ, quelli più economici contengono solo parti strutturali e motori mentre Kit più avanzati contengono anche sensori e il Remote Controller che serve per comandare a distanza i vostri robot.

Per costruire "RoboGino" ci basterà un semplice Kit contenente alcune parti strutturali e due Motori.

Sebbene la prima attività che presenterò presupporrà l'utilizzo del Remote Controller, in tutte le altre mi concentrerò su aspetti di programmazione che non presuppongono l'utilizzo del Remote Controller.

Ogni attività può essere considerata un punto di partenza per costruirne altre più complesse o più semplici adatte a studenti compresi in una fascia di età [8-19].

## 2. Robot, Pensiero computazionale e coding

### 2.1. I Robot

La parola robot deriva dal ceco *robota* che significa lavoro pesante e sta ad indicare una qualsiasi macchina in grado di svolgere in modo più o meno indipendente un lavoro al posto dell'uomo.

Pensiamo ad esempio alle gru usate nei cantieri per muovere oggetti pesanti che come avrete più volte notato sono controllate a distanza da un operatore tramite un telecomando, oppure a tutte quelle macchine per movimento terra come ruspe, trattori ecc. Si tratta di macchine che sono mosse da una sequenza di azioni o input che partono dal cervello dell'operatore umano e sono trasmesse alla macchina tramite leve pedali o controlli a distanza. In questo caso parliamo di *Robot Controllati*. A volte però abbiamo l'esigenza di programmare un cervello artificiale che si sostituisca a quello umano e sia in grado di far seguire alla macchina una serie di azioni in sequenza come ad esempio accade nelle catene di montaggio. Programmiamo quindi il cervello elettronico che controlla la macchina inserendo al suo interno un programma ovvero una serie di input che il cervello elettronico tradurrà in movimenti della macchina. Si parla questa volta di *Robot Autonomi* come ad esempio le lavatrici che eseguono cicli in base al programma di lavaggio da noi impostato oppure ai robot da cucina che sono in grado di variare velocità di rotazione delle lame e temperatura del cestello in modo autonomo seguendo svariati programmi di cottura preimpostati. Un passo in più che ci avvicina a quella che comunemente chiamiamo *Intelligenza artificiale* consiste nel inserire all'interno del cervello artificiale programmi più complessi che prevedono anche la gestione di stimoli esterni. Ovvero si possono prevedere dei cicli decisionali basati su input che il cervello elettronico può ricevere dall'esterno tramite sensori che possono rilevare informazioni di vario tipo come ad esempio, temperatura, pressione non solo atmosferica ma anche pressione intesa come pressione di un tasto, distanza, luce, colori etc.

Nelle nostre attività ci concentreremo inizialmente a programmare RoboGino semplice macchina in grado di muoversi e priva di sensori in un secondo momento proveremo a fare degli upgrade di RoboGino applicandogli dei sensori sperimentando in questo modo la gestione di stimoli dall'esterno.

### 2.2. Il pensiero computazionale

Si definisce pensiero computazionale, la modalità di pensare alla risoluzione di un problema scomponendolo in piccoli parti più semplici. Procedimento tipico delle menti scientifiche che pensano in modo algoritmico e a livelli multipli di astrazione. L'algoritmo o ciò che più semplicemente chiamiamo programma non è altro che un procedimento che risolve un determinato problema attraverso un numero finito di passi elementari. La sequenza di passi può essere rappresentata graficamente attraverso un diagramma di flusso.

© Copyright By Nabla tecnologie S.R.L.

**RoboGino - Unità didattiche**

### 2.3. Diagramma di flusso

L'uso di un opportuno diagramma di flusso permette di rappresentare graficamente un qualsiasi algoritmo tramite una sequenza di blocchi. Ciascun blocco ha un significato diverso come si può vedere di seguito.

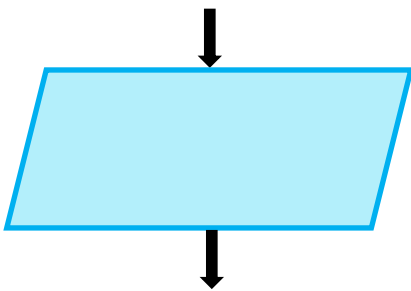
#### Ovale

Indica l'inizio o la fine del diagramma. Da ogni blocco iniziale deve uscire una sola freccia mentre in ogni blocco finale deve entrare una sola freccia.



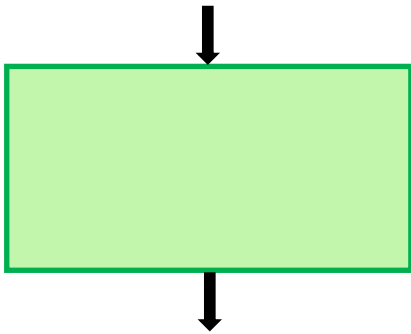
#### Parallelogramma

Indica l'introduzione dei dati. Presenta una freccia in entrata ed una in uscita.



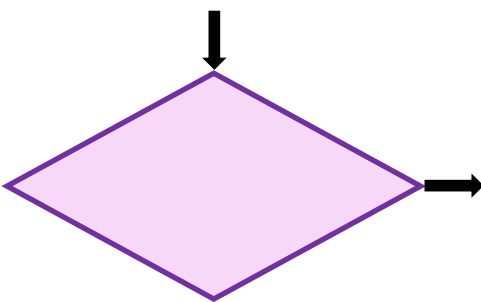
#### Rettangolo

Indica le operazioni da seguire. Presenta una freccia in entrata ed una in uscita. due



#### Rombo

Indica l'analisi di una situazione e la scelta di un procedimento. Nel rombo entra una freccia e ne escono.



#### Esempio Applicativo

Si scelgano due alunni della classe, il primo lo si nomini “Programmatore”, il secondo “Robot”. Il programmatore dovrà scrivere alla lavagna l’opportuno diagramma di flusso che permetta al Robot di eseguire i seguenti programmi:

### **Programma 1 - “Accendi la luce”**

Immaginiamo che il Robot sia seduto su uno sgabello, e che sia in grado di recepire le seguenti azioni elementari:

- Alzati o sediti
- Fai un passo avanti o indietro
- Voltati a destra o sinistra di X gradi
- Premi l’interruttore

Scrivere il diagramma di flusso che consenta al robot di alzarsi, raggiungere l’interruttore davanti a lui sulla destra, accendere la luce e tornare a sedersi al suo posto.

### **Programma 2 - “Accendi la luce se è Buio”**

Immaginando che il Robot del problema precedente abbia anche un sensore in grado di rilevare la luce e che questo ritorni il valore “sì” in caso rilevi luce o “no” in caso rilevi buio, scrivere il diagramma di flusso affinché il Robot accenda la luce solo se buio.

Diagramma di flusso dell'Esempio 1

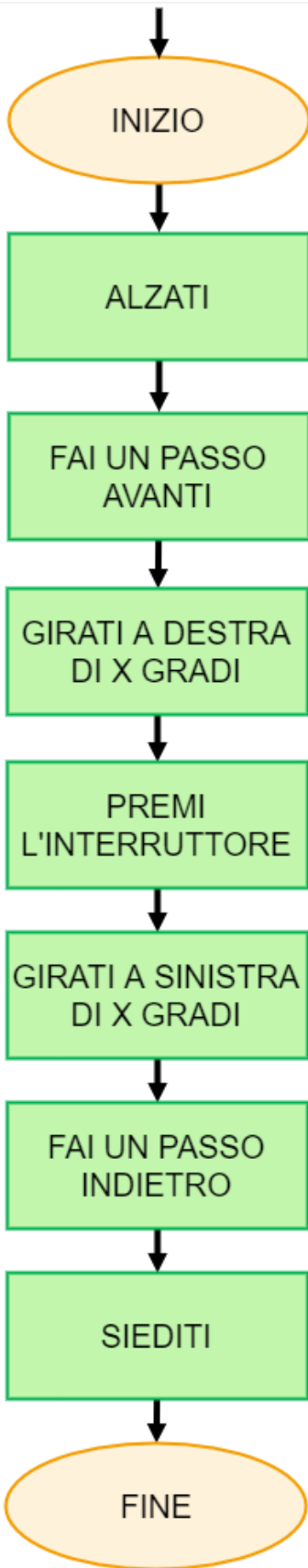


Diagramma di flusso dell'Esempio 2



### 3. Guida all'assemblaggio del RoboGino

Il RoboGino è stato pensato per essere assemblato facilmente con il minor numero possibile di parti strutturali, contenute nei Kit VEQ IQ più economici, e al tempo stesso per essere ampliato con i vari tipi di sensori.

Le parti necessarie per la sua costruzione sono le seguenti:

- 1 Robot Brain
- 1 Radio
- 1 batteria
- 16 pin 1x1
- 2 connettori a base larga 2 2x1,5
- 3 travi 1x8
- 2 travi 2x8
- 2 assi da 2.5M con fermata a 0,5M
- 6 distanziatori per assi da 0,25M
- 4 connettori ad angolo 2x2x1
- 2 ruote da 200m
- 2 boccole per assi
- 1 connettori a base larga 2 1x1,5
- 1 assi da 2M con fermata alla fine
- 1 trave 1x3
- 3 rondelle
- 1 connettore a base larga 1 1x1,5
- 1 asse da 4M con fermata alla fine
- 2 ruote da 100m
- 2 smart cable

Utilizzando il codice QR qui a fianco potete visualizzare il video tutorial della costruzione del RoboGino.



© Copyright By Nabla tecnologie S.R.L.

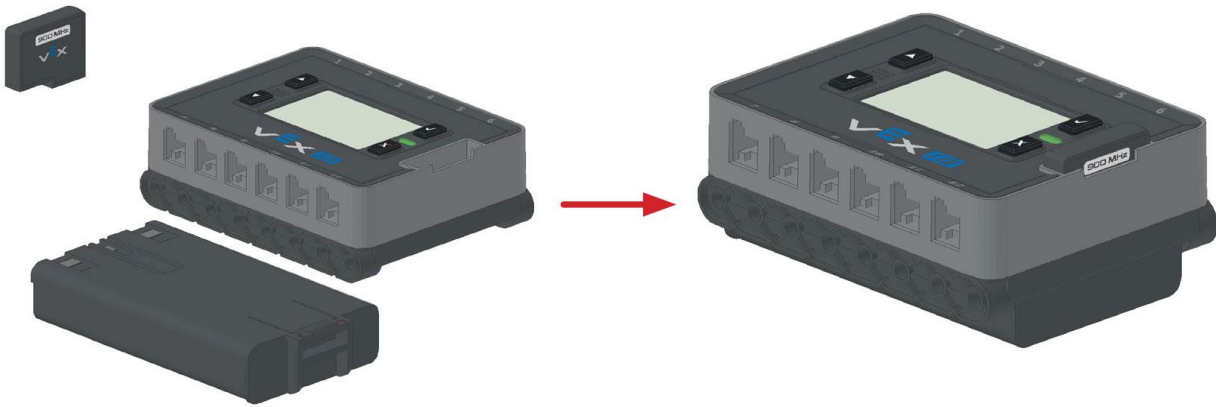
**RoboGino - Unità didattiche**

In seguito sono riportati tutti i passaggi necessari all'assemblaggio del RoboGino. Per le istruzioni necessarie all'installazione e la rimozione della Batteria e della Radio fare riferimento al Manuale d'uso presente in ogni kit VEX IQ.

© Copyright By Nabla tecnologie S.R.L.

**RoboGino - Unità didattiche**

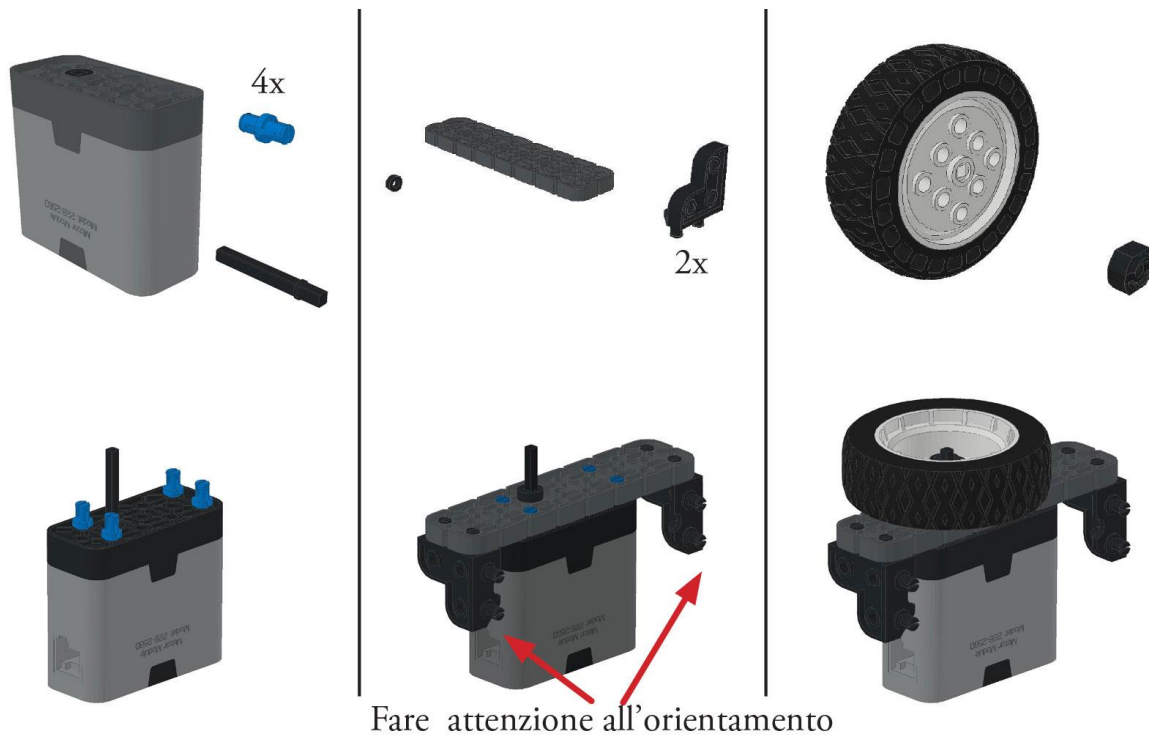
### 3.1. Il Robot Brain



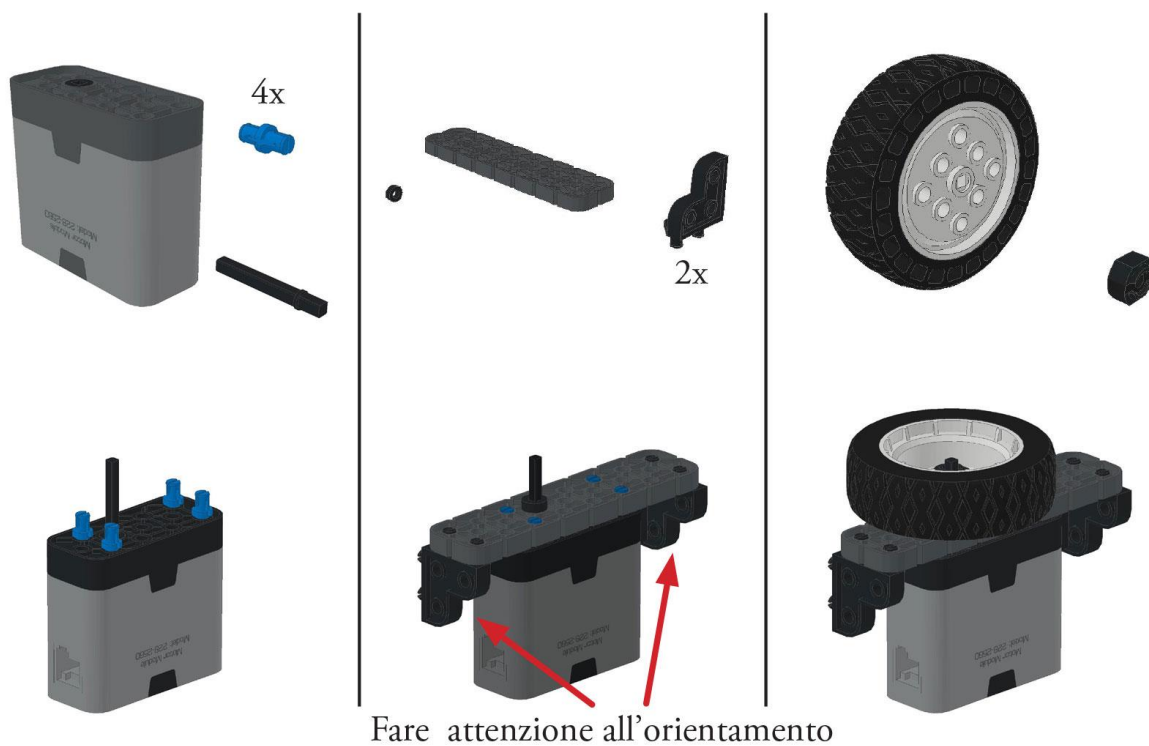
© Copyright By Nabla tecnologie S.R.L.

RoboGino - Unità didattiche

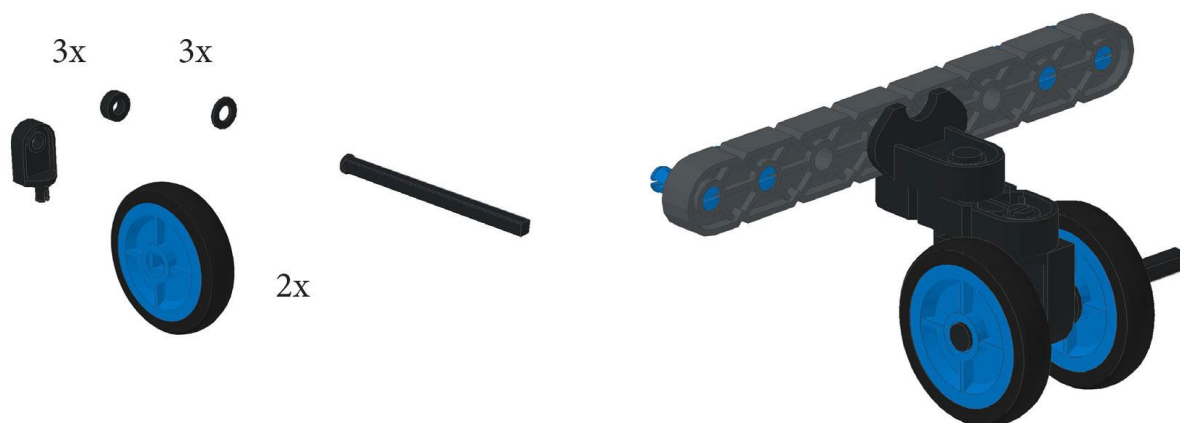
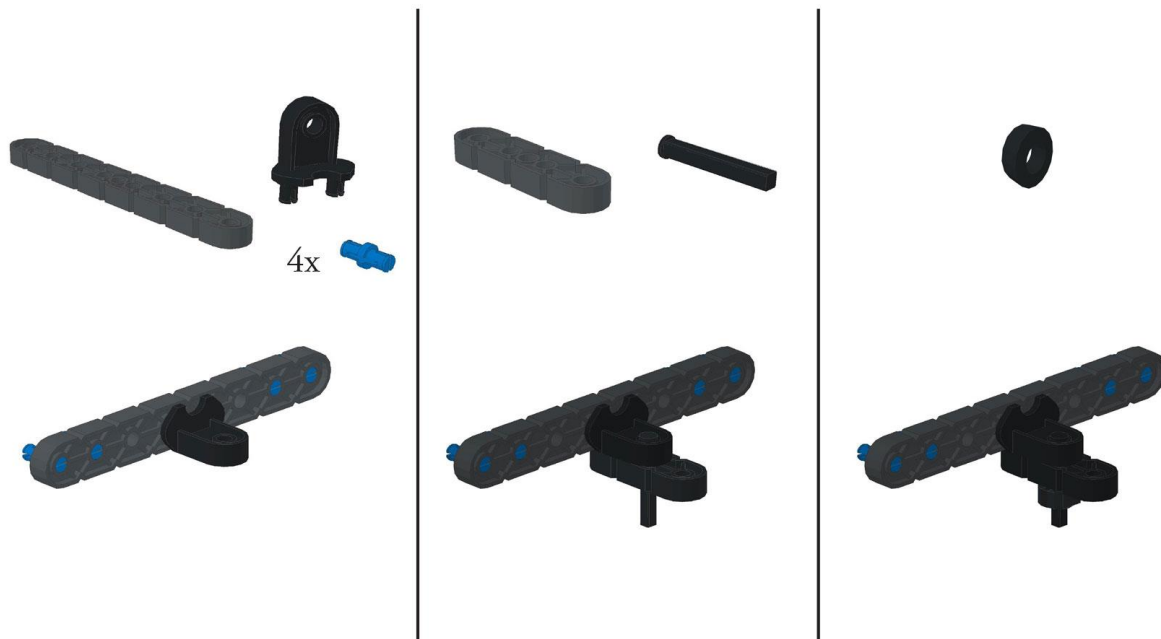
### 3.2. Il motore di destra



### 3.3. Il motore di sinistra



## Rotella posteriore

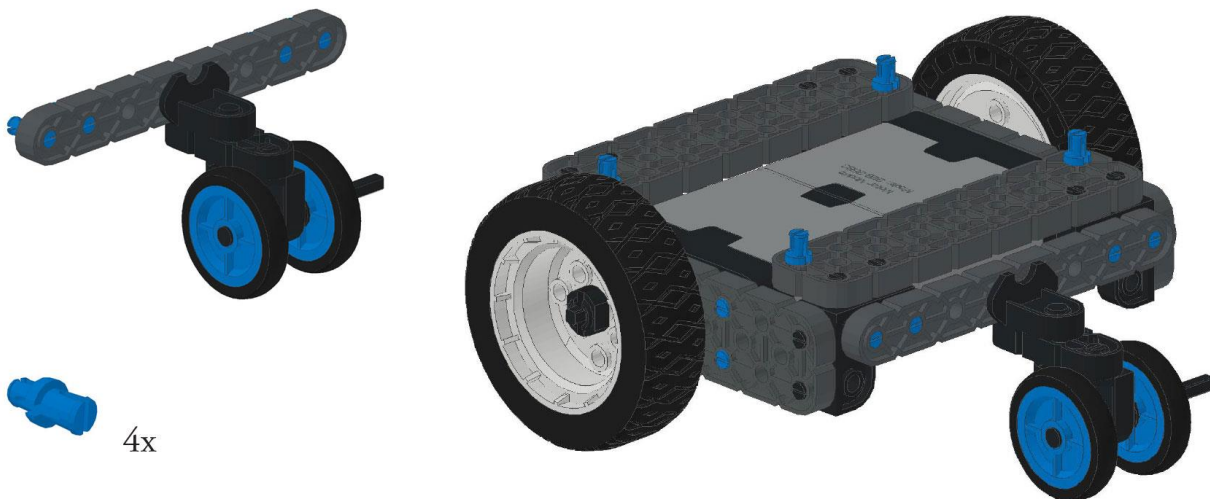
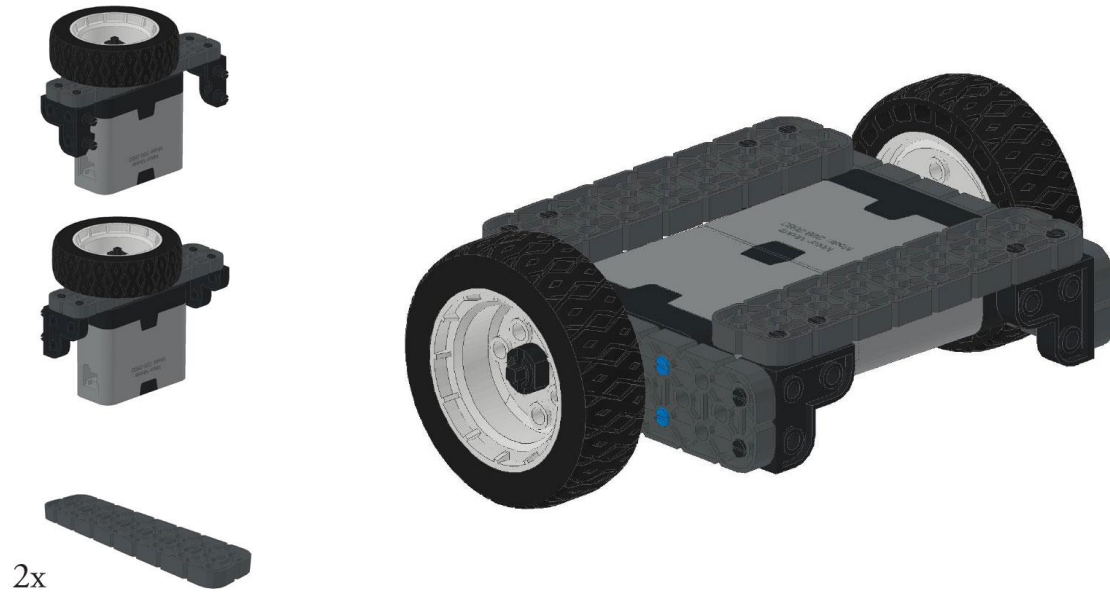


N.B.: utilizzare i cuscinetti in modo da garantire una rotazione senza attrito alle ruote.

© Copyright By Nabla tecnologie S.R.L.

RoboGino - Unità didattiche

### 3.4. Base con i due motori e le ruote

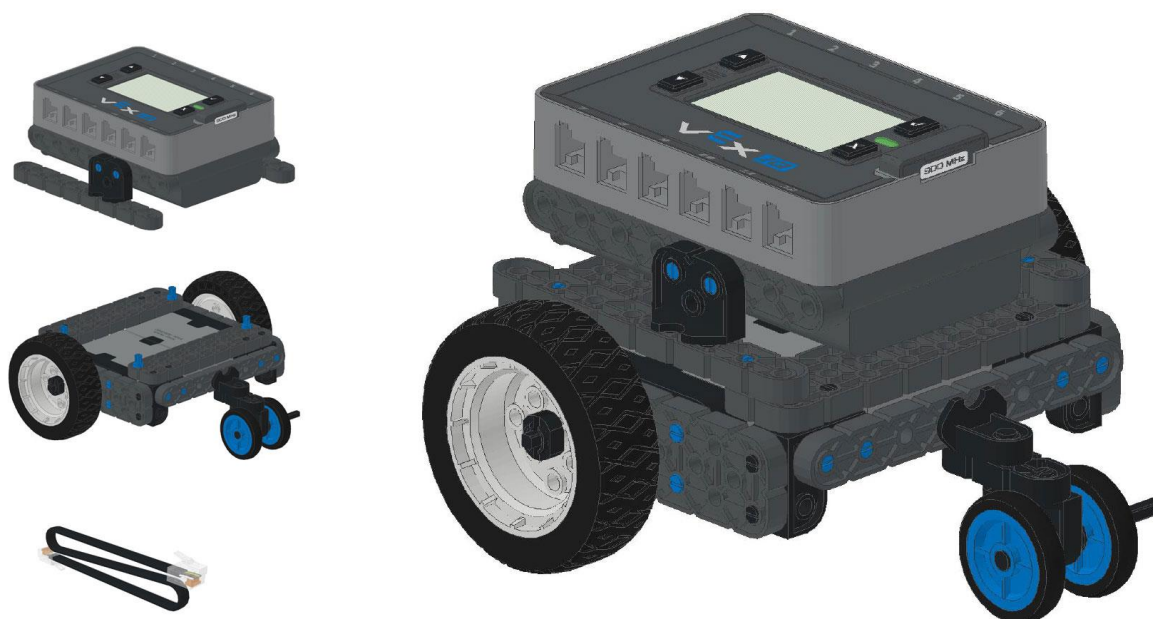


© Copyright By Nabla tecnologie S.R.L.

RoboGino - Unità didattiche

### 3.5. Il RoboGino

Uniamo il Robot Brain e la base, collegando i due smart cable ai due motori e alle porte 1 e 7.



Ed ecco il risultato finale.



© Copyright By Nabla tecnologie S.R.L.

RoboGino - Unità didattiche

## 3.6. Pronti per partire

A questo punto i nostri obiettivi sono:

- Caricare la batteria del Robot Brain, par. 1 del manuale.
- Caricare la batteria del radiocomando, par. 1 del manuale.
- Installare le radio sia sul Robot Brain che sul Controller, par. 2.1 del manuale.
- Sincronizzazione del Robot Brain e del Controller, par. 2.2 del manuale
- Aggiornare il firmware utilizzando VEXos Utility par. 7.1 del manuale.



## 4. La Programmazione

### 4.1 Coding

Da questo punto in poi della lettura, si presume che abbiate il vostro RoboGino montato nella sua versione più semplice cioè senza sensori. Per avviare l'attività di coding bisogna trasformare il diagramma di flusso in linguaggio comprensibile al nostro cervello elettronico o Robot Brain, ovvero in "Linguaggio macchina" cioè una sequenza di bit infinita di zeri e uni incomprensibile a noi ma lingua madre delle macchine. Esistono dei linguaggi di programmazione ovvero dei codici che si collocano a livelli di astrazione superiori rispetto alle pure sequenze di bit e sono supportati da compilatori che riconvertono poi il codice in bit. Questi linguaggi di astrazione rendono più semplice la programmazione delle macchine.

### 4.2 VexCode IQ Block

© Copyright By Nabla tecnologie S.R.L.

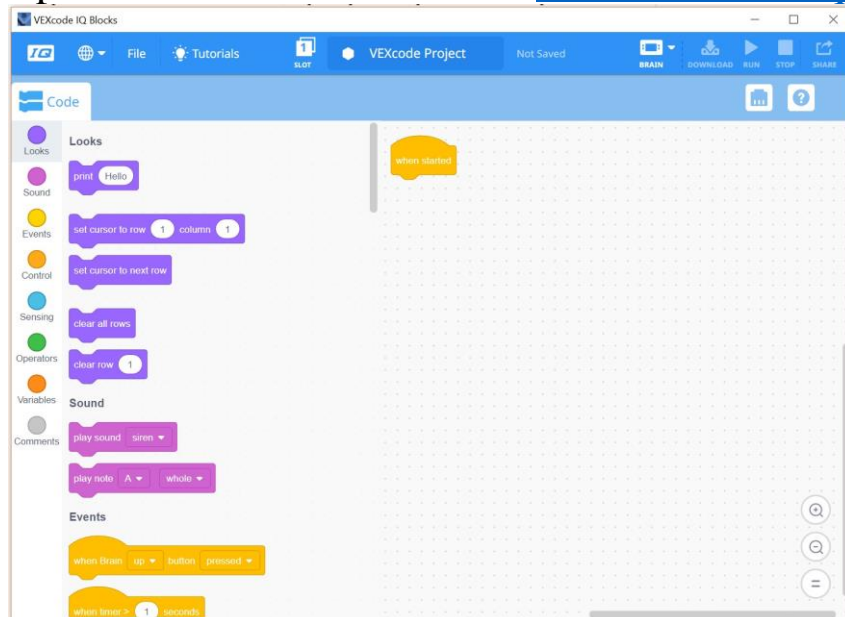
**RoboGino - Unità didattiche**

Noi useremo VexCode IQ Block, un linguaggio di programmazione a blocchi, su misura per chi è alle prime armi, che si colloca quindi ad un livello di astrazione molto alto rispetto alle sequenze di bit.

### 4.3 Avvio di VexCode IQBlock

Per prima cosa quindi, scarichiamo VexCode IQ Block Link e seguiamo le istruzioni a schermo per portare a termine l'installazione e di conseguenza avviamo il software.

#### 4.3.1 Link per il download del software - [vexrobot.it/vex-iq/download/](https://vexrobot.it/vex-iq/download/)

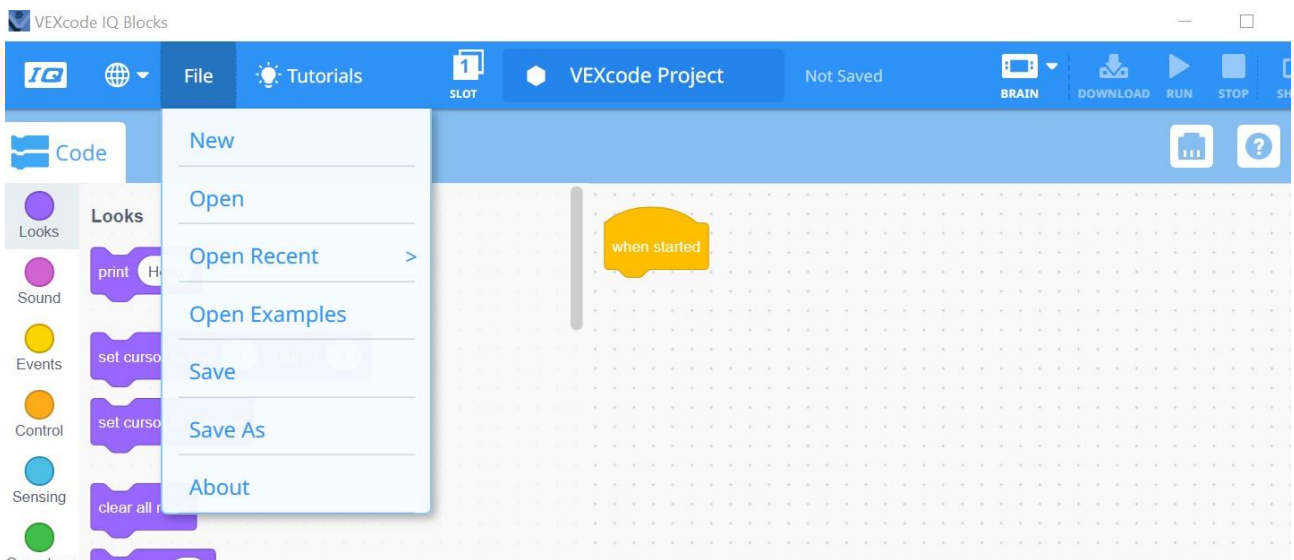


### 4.4 Apertura nuovo progetto

All'apertura del software ritroviamo l'interfaccia mostrata nella figura al paragrafo precedente. C'è già proposto un nome per il nostro primo progetto "VexCode Project" facendo clic sopra con tasto destro si apre la finestra di Windows che ci permette la scelta di directory, cartella e nome per il salvataggio. La stessa operazione la possiamo fare anche attraverso il percorso File>>Save As. Utilizziamo invece Save per salvare le modifiche di volta in volta effettuate su un progetto esistente.

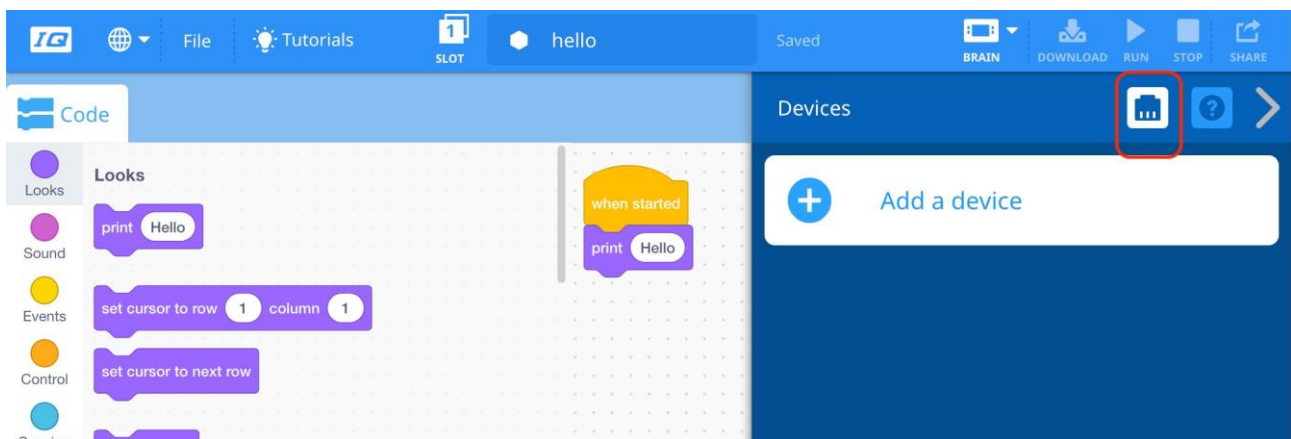
© Copyright By Nabla tecnologie S.R.L.

**RoboGino - Unità didattiche**



## 4.5 Sezione Devices

Per entrare nella sezione Devices bisogna cliccare sull'icona evidenziata in rosso.



In questa sezione andiamo ad indicare al software la nostra architettura di sistema, come è fatto RoboGino?

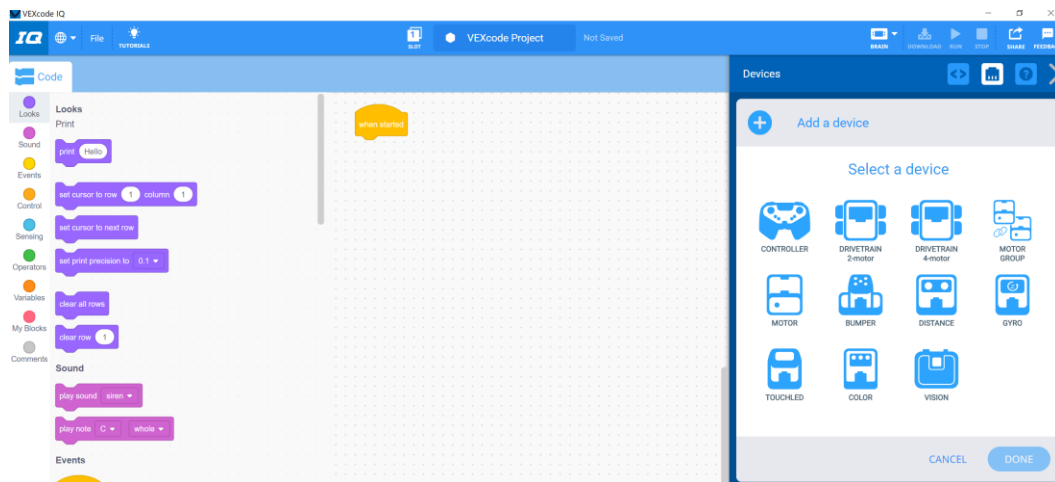
a parte il Brain, aggiungiamo tutti sensori e motori che compongono il nostro robot e indichiamo per ciascuno la porta di connessione corrispondente sul Brain.

Vi consiglio di definire il design del robot e salvare il programma (esempio: RoboGino). Potrete poi utilizzare questo programma come

base per crearne altri che presuppongono lo stesso design, salvandoli di volta in volta con un nome differente.

#### 4.6 Configurazione di RoboGino su Vex Code Iq Blocks

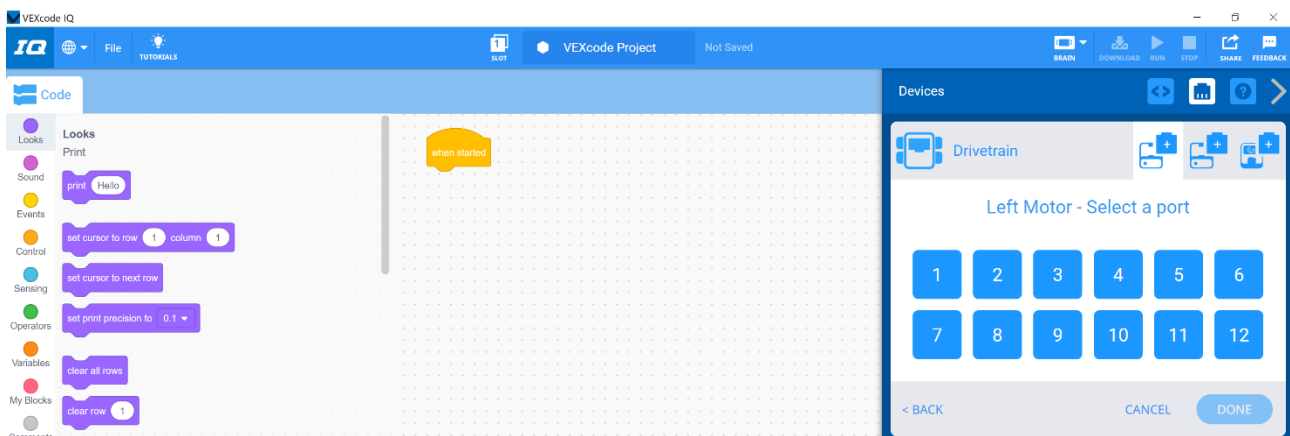
Cliccando sul tasto + compaiono tutti i Devices disponibili (drivetrain, motori, sensori e radiocomando).



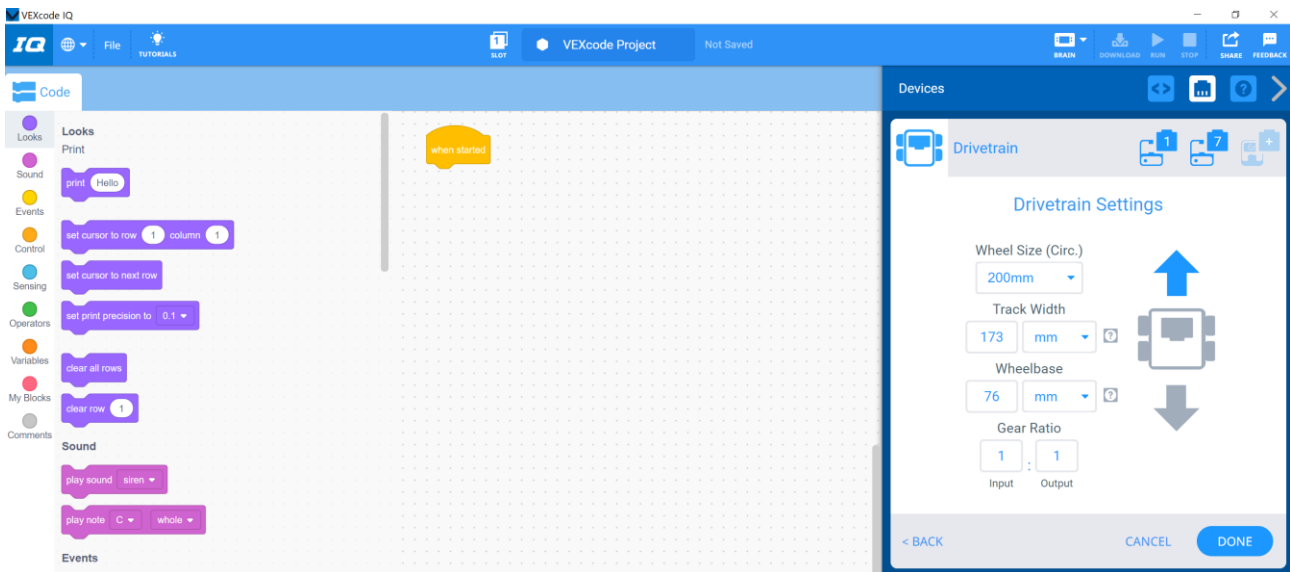
IL drivetrain è primo componente che consiglio di aggiungere. Esso è lo chassis o telaio meccanico, la struttura portante di un qualsiasi mezzo di trasporto. Le informazioni che il drivetrain contiene all'interno di Vex Code Iq Blocks sono le seguenti:

- Motore n°1 e relativa porta di connessione
- Motore n°2 e relativa porta di connessione
- Diametro delle ruote e loro distanza
- Presenza giroscopio e relativa porta di connessione

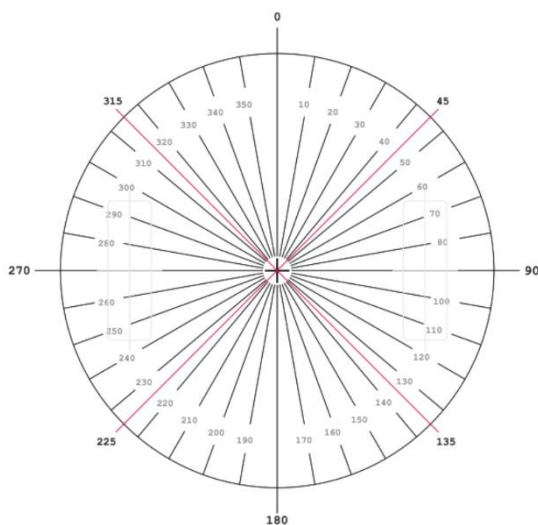
Cliccando su drivetrain 2 motors compare la seguente schermata. Inseriamo le porte di connessione del motore destro e sinistro nel nostro caso 1 e 7.



Togliamo la spunta dal Gyro.



Qui di fianco vedete il drivetrain di RoboGino. Il giroscopio non è stato installato nel nostro esempio. Inserite Wheel Size 200 mm. Gear Ratio 1:1. Per ciò che riguarda i valori Track Width e Wheelbase , questi

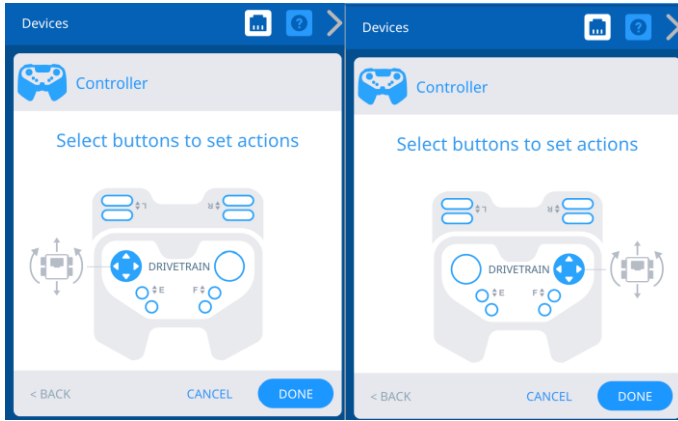


vanno ricavati empiricamente. Come avrete notato per RoboGino abbiamo utilizzato un chassis con due sole ruote, l'intenzione era quella di costruire un robot molto semplice in pochi minuti. Questi due valori da trovare sperimentalmente, influiscono sul grado di rotazione che ha il robot. Quindi utilizzando il tappetino di calibrazione andiamo a effettuare diverse prove facendo ruotare

RoboGino di volta in volta di angoli diversi. Per ora tralasciamo questa parte perché non abbiamo ancora visto come realizzare ed eseguire con RoboGino un semplice programma.

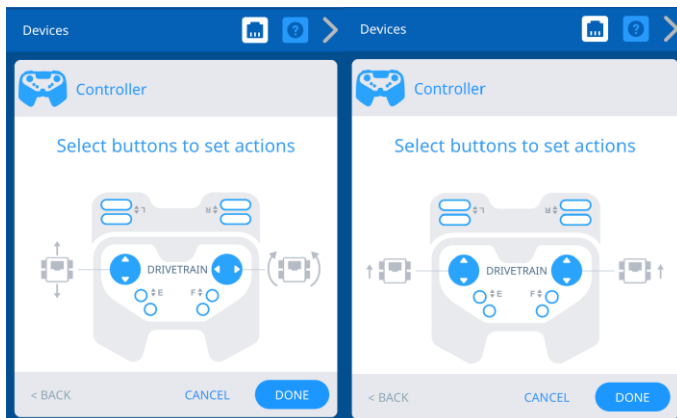
© Copyright By Nabla tecnologie S.R.L.

RoboGino - Unità didattiche



Ora è la volta del Controller ovvero il Radiocomando. Potete decidere se utilizzarlo oppure no. Se lo utilizzate dovete inserire anche questo come device. Per farlo cliccate sull' icona raffigurante il controller situata in basso a destra sul Drivetrain come mostrato di seguito in figura.

Nel caso di Robogino il controller serve per poter muovere il nostro robot nelle quattro direzioni: avanti, indietro, destra, sinistra, Lo possiamo fare in quattro modi diversi. Cliccando sui joystick di sinistra avremo come prima possibilità quella di controllare i quattro movimenti con il joystick di sinistra. Cliccando sul joystick di destra attiviamo la seconda modalità che consente di governare con esso i quattro movimenti. Cliccando ancora sul joystick di sinistra abbiamo la terza possibilità ossia avere avanti e indietro sul joystick di sinistra e destra e sinistra sul joystick di destra. Quarta ed ultima possibilità è quella di



controllare rispettivamente con il joystick di sinistra i I movimento avanti indietro del motore di sinistra e con il joystick di destra il movimento avanti, indietro del motore di destra. La modalità che consiglio è la quarta che ci permette di gestire RoboGino come se fosse un carro armato, spingendo il joystick di sinistra

verso l'alto otterremo un a rotazione verso destra otterremo invece l'effetto contrario spingendo il joystick di destra verso l'alto. Per mandare RoboGino avanti o indietro bisognerà spingere rispettivamente avanti o indietro contemporaneamente i due joystick.

© Copyright By Nabla tecnologie S.R.L.

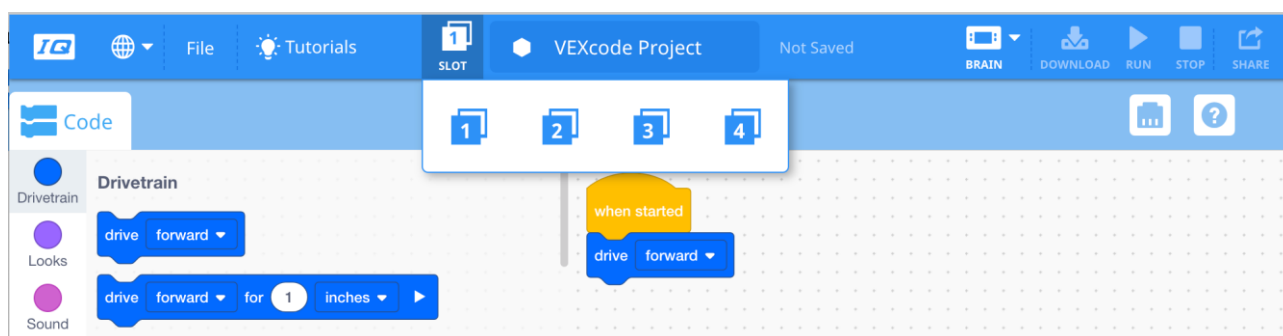
**RoboGino - Unità didattiche**

## 4.7 Salvataggio Configurazione RoboGino con Controller

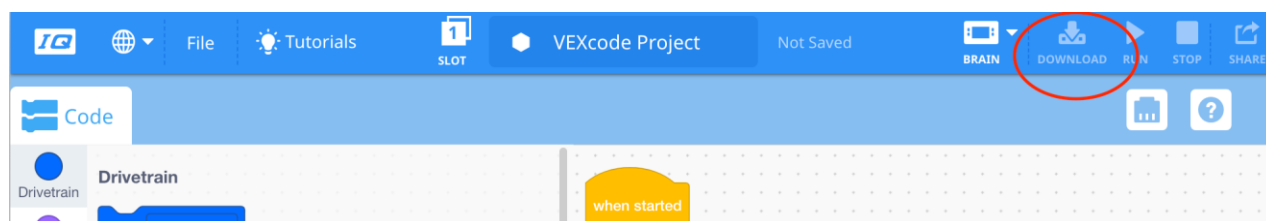
Assicuratevi di essere nella seguente situazione:

- Avere RoboGino Costruito perfettamente
- Avere avviato Vex IQ Code Blocks sul computer
- Avere RoboGino Connesso via USB al Computer
- Avere effettuato i passaggi precedenti per la configurazione del design RoboGino.

Rinominiamo il nostro programma RoboGino oppure lasciamo il nome di default VEXcode Project. (c.f.r. par.4.4). Connettiamo con cavo USB il Robot Brain al Computer. Accendiamo il Robot Brain premendo il bottone “Conferma” (✓) situato di fianco lo schermo LCD. Ci posizioniamo con il cursore sul menu a tendina SLOT e scegliamo lo slot 1.



Premiamo poi il tasto con il simbolo Download (cerchiato in rosso).



Ci sono ora due modi per far partire il programma, il primo tenendo il Robot connesso al computer con cavo USB e premendo Run (▶) sulla consolle di Vexcode. L'altro modo sicuramente più pratico, consiste nel disconnettere il cavo USB ed andare a richiamare il programma dal display LCD situato sul Robot Brain (cfr. manuale utente a pag. 12).

## 4.8 RoboGino Autonomo

Per le nostre attività di coding ci focalizzeremo sulla programmazione di RoboGino come robot controllato, ovvero la sua architettura di sistema sarà costituita solo dal drivetrain, motori e sensori ma non

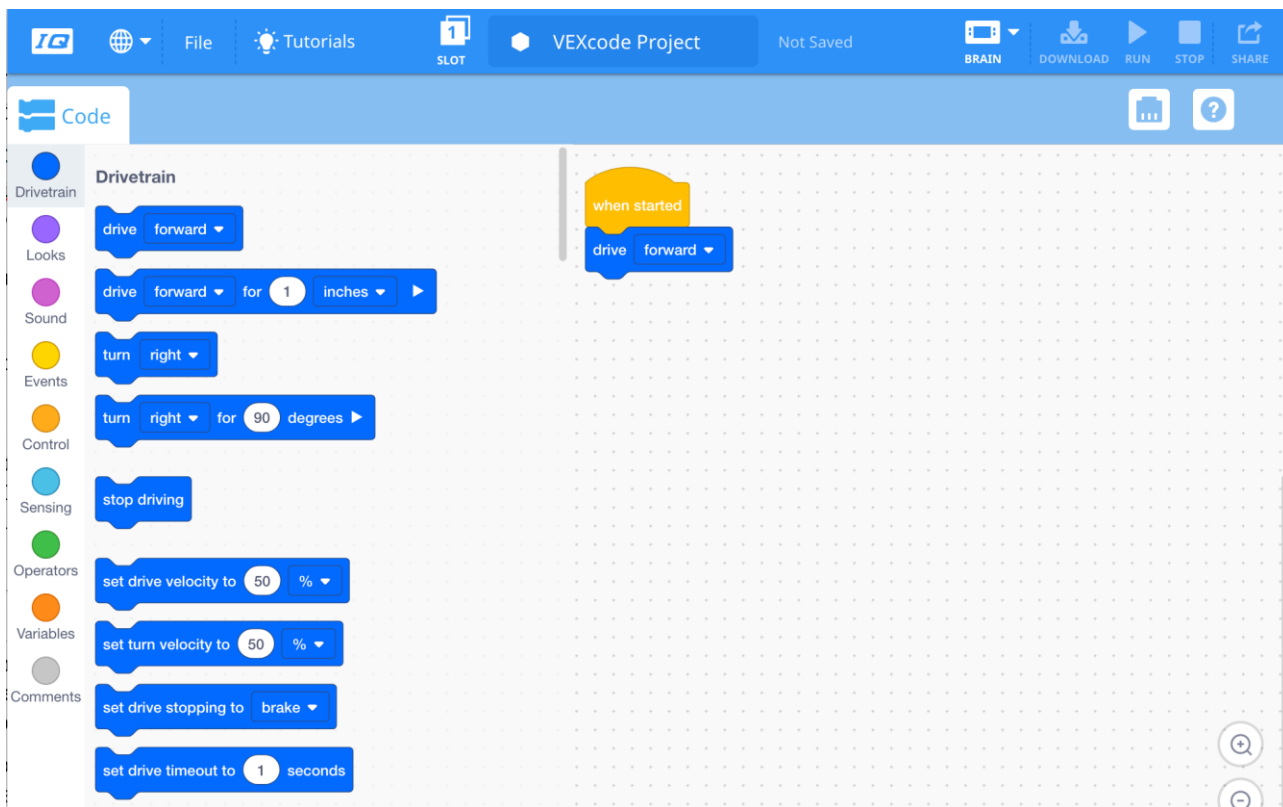
comparirà il controller. Il robot sarà autonomo e si muoverà eseguendo il flusso del codice a blocchi da noi scritto.

#### 4.9 Primo esempio pratico- “VAIAVANTI”

Programmiamo il drivetrain per per compiere uno spostamento in avanti. Apriamo un nuovo progetto (c.f.r. par.4.4). Configuriamo il nostro RoboGino composto questa volta dal solo Drivetrain ma senza controller. Ripercorriamo quindi il par.4.6 senza inserire il controller.

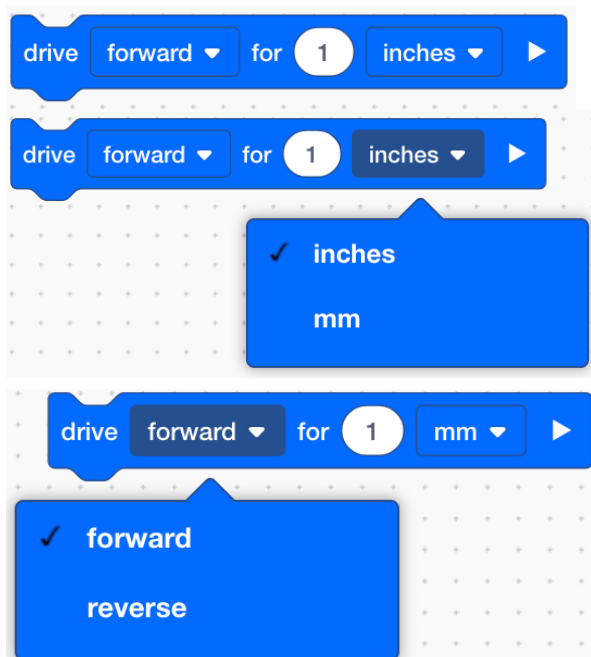
Trasciniamo ora nell’area di programmazione i blocchi indicati nella figura sottostante.

Dopo come già mostrato nel par.4.7, connettiamo con cavo USB il Robot Brain al Computer. Accendiamo il Robot Brain premendo il bottone “Conferma” (✓) situato di fianco lo schermo LCD ci posizioniamo sul menu a tendina SLOT e scegliamo questa volta di salvare il nostro programma nello slot 2 visto che lo slot 1 è stato occupato in precedenza. Come già visto nel par.4.7 Ci sono ora due modi per far partire il programma, il primo tenendo il Robot connesso al computer con cavo USB e premendo Run (▶) sulla consolle di Vexcode. L’altro modo sicuramente più pratico, consiste nel disconnettere il cavo USB ed



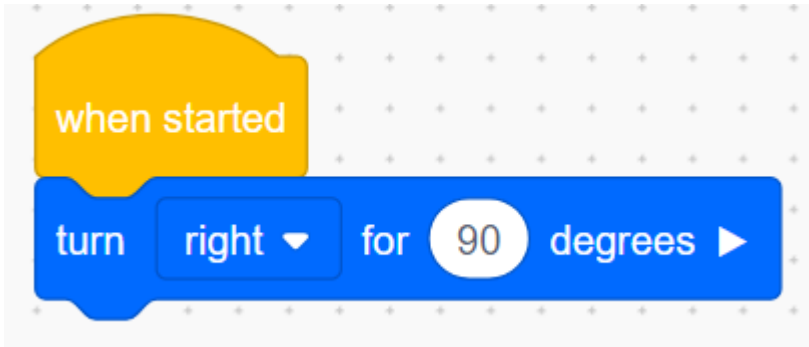
andare a richiamare il programma dal display LCD situato sul Robot Brain (cfr. manuale utente a pag. 12).

Una volta fatto partire il programma vedremo il nostro Robot muoversi di 25mm.



Potete ora provare a fargli compiere distanze diverse utilizzando il blocco drive forward for. Inserendo un numero all'interno del riquadro che segue il for otterremo come risultato un determinato spostamento in pollici. Esempio se inseriamo <<drive forward for 3 inches>> otteniamo uno spostamento in avanti di 3 pollici. Possiamo anche cambiare l'unità di misura aprendo il menu a tendina e sostituendo inches con mm. Altra cosa possiamo cambiare forward con reverse e di modo che Robo Gino si sposterà indietro

anziché avanti. Ritorniamo ora a ciò che avevamo lasciato al termine del paragrafo 4.6., mi riferisco ai valori Track Width e Wheelbase. Per determinare questi valori dobbiamo lanciare il seguente programma.



Andiamo a variare di volta in volta i valori Track Width e Wheelbase finché il nostro RoboGino non compirà una rotazione perfetta di 90°. Possiamo anche provare diversi valori di angoli.

#### 4.10 Conclusione

A questo punto è terminata l'infarinatura che avevo intenzione di dare sia sulla programmazione che sull'uso della piattaforma IQ abbinata al linguaggio di programmazione a blocchi VEXcode IQ Blocks. Dai prossimi capitoli presenterò solo unità pratiche e vi introdurrò mano all'interno di esse concetti nuovi come digrammi di flusso iterativi e utilizzo dei sensori.

## 5. Unità didattiche

Di seguito propongo delle unità didattiche da svolgere in classe, ciascuna è descritta secondo uno schema ben preciso che prevede la compilazione di una scheda e la successiva descrizione dello svolgimento. Nel sito [www.vexrobot.it](http://www.vexrobot.it) potete trovare alcune di queste unità didattiche inserite seguendo lo stesso schema. Anche voi registrandovi al sito potete contribuire con l'inserimento di vostre unità.

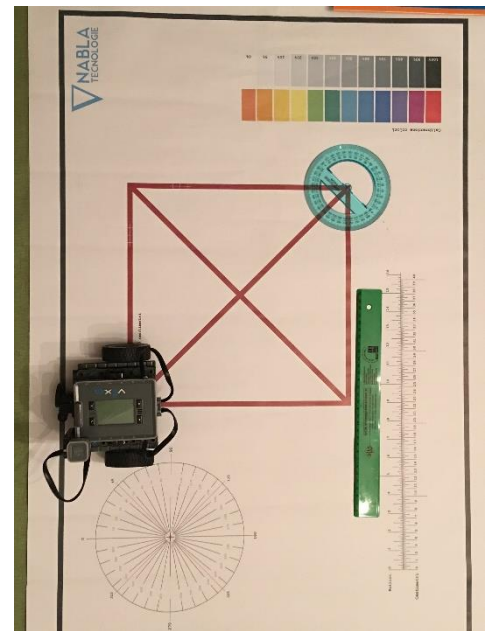
### 5.1. Imparare a misurare lunghezze ed angoli

#### Scheda

<i>Argomenti</i>	Imparare a fare misure di lunghezze e angoli utilizzando il righello, il goniometro o altri strumenti es. il Robot
<i>Coding</i>	Programmare un robot per percorrere una certa distanza in linea retta o per farlo ruotare di un certo angolo
<i>Robotica educativa</i>	Robot dato.
<i>Fascia d'età</i>	[8-12]
<i>Design del robot</i>	Il RoboGino
<i>Materiale</i>	Tappetino per unità didattiche incluso nel Competition kit IQ, righello, goniometro
<i>Descrizione</i>	L'insegnante propone di misurare con il righello il lato del quadrato rosso disegnato sul tappetino. La misura osservata empiricamente con l'utilizzo del righello viene poi verificata utilizzando il robot come strumento di misura. Scriviamo il diagramma di flusso che rappresenta la sequenza di azioni che il robot deve compiere per percorrere un lato del quadrato. Traduciamo il diagramma di flusso in sequenza di blocchi.

## Svolgimento

Scelto uno studente lo si chiama gli si chiede di avvicinarsi al tappetino di calibrazione e di misurare il lato del quadrato sul tappetino di calibrazione. Rilevata la misura la si vuole ora confrontare con quella rilevata dal Robot. Per farlo posizioniamo il robot con l'asse delle ruote sul punto di partenza della linea che rappresenta il lato del quadrato. Per un posizionamento corretto sono evidenziate sul tappetino di calibrazione delle linee guida. Ora bisogna programmare RoboGino affinché percorra la distanza rilevata dall'alunno. Una volta programmato, RoboGino percorrerà esattamente la distanza che noi gli chiederemo di percorrere. Verificheremo in questo modo se l'alunno ha effettuato la misura corretta utilizzando il righello.

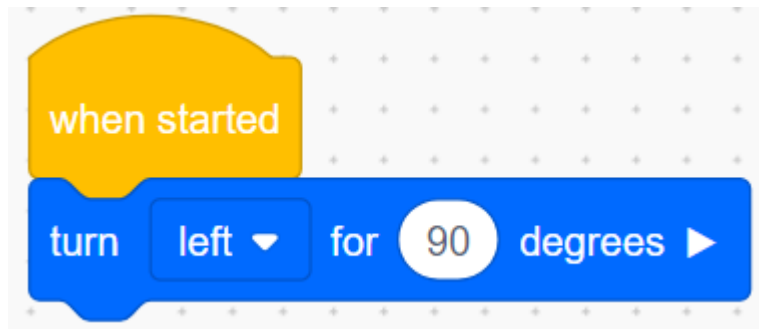
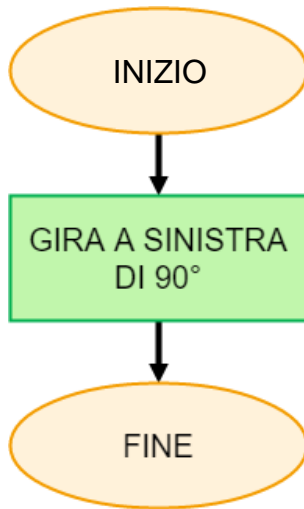


Si chiede all'alunno di misurare gli angoli del quadrato con il goniometro, la stessa misura sarà riportata con un segno a matita nell'area goniometro sul tappetino, si posiziona poi il robot nella stessa area utilizzando le giustapposte guide e si effettua la programmazione del robot per far sì che effettui una rotazione dello stesso angolo misurato in partenza.

## Programmazione del Drivetrain

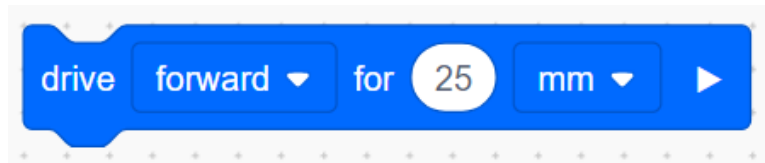
Diagramma di flusso	Codice corrispondente
Prima parte, programmo il robot per andare avanti	

Seconda parte, programmo il robot per girare di un certo angolo



### Varianti e spunti di riflessione

Come avrete notato in VexCode le lunghezze sono espresse in mm, questa è un'occasione per i ragazzi di esercitarsi anche con le equivalenze per cui fanno la misura con il righello in cm e poi la trasformano in mm per programmare il robot.



Possiamo chiedere ai ragazzi di misurare la diagonale del quadrato e verificare poi la misura con il robot. Da qui in base all'età degli alunni proseguire con teorema di Pitagora, proprietà dei triangoli etc.

## 5.2. Il Quadrato

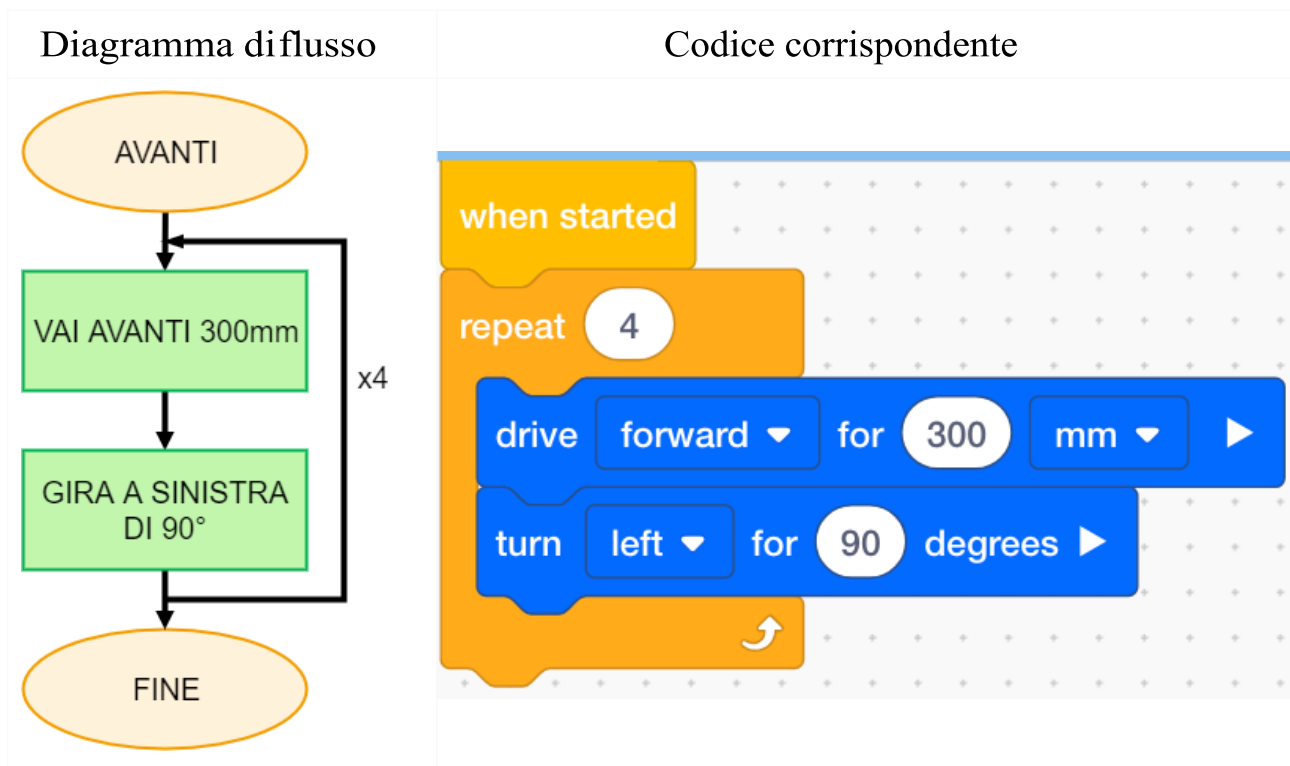
### Scheda

<i>Argomenti</i>	Imparare a fare misure di lunghezze e angoli utilizzando il righello, il goniometro o altri strumenti es. il Robot
<i>Coding</i>	Programmare il robot per percorrere il perimetro del quadrato, studio ciclo repeat
<i>Robotica educativa</i>	Robot dato.
<i>Fascia d'età</i>	[8-12]
<i>Design del robot</i>	Il RoboGino
<i>Materiale</i>	Tappetino per unità didattiche incluso nel Competition kit IQ, righello, goniometro
<i>Descrizione</i>	L'insegnante propone di misurare con il righello il lato del quadrato rosso disegnato sul tappetino di calibrazione. La misura osservata empiricamente con l'utilizzo del righello viene poi verificata utilizzando il robot come strumento di misura. Si programma il robot per percorrere il perimetro del quadrato mettendo in evidenza prima il diagramma di flusso poi la corrispondente sequenza di blocchi.

## Svolgimento

Utilizzando quanto già visto nell'Unità didattica 1, possiamo quindi invitare l'alunno a misurare sul tappetino di calibrazione con l'utilizzo di righello e goniometro le misure del quadrato. Rilevate queste misure possiamo scrivere correttamente il diagramma di flusso ed il programma che permetteranno al robot di seguire il corretto perimetro.

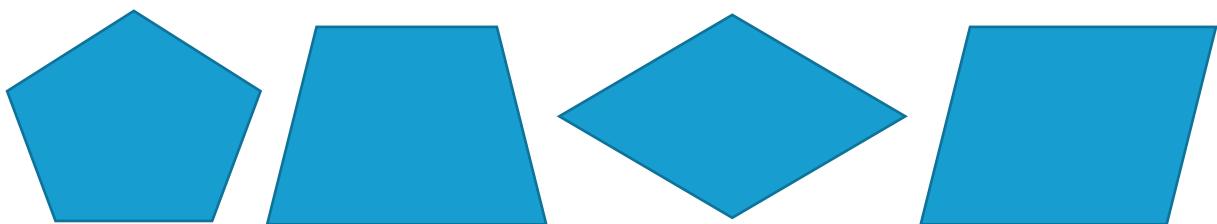
## Programmazione del Drivetrain



Come si vede con questo esempio mettiamo in evidenza un diagramma di flusso con ciclo repeat in modo da rendere più compatta la descrizione del flusso del programma e diminuire di conseguenza i blocchi di codice.

## Varianti e spunti di riflessione

Dato il design del robot che esegue perfettamente tutti i nostri comandi, vai avanti, gira a destra, sinistra etc. possiamo con dello scotch colorato disegnare delle forme sul pavimento e chiedere agli alunni di scrivere i corretti diagrammi di flusso e codice che permettano al robot di percorrere le varie figure.



Come faccio a programmare RoboGino per percorrere un cerchio di dato raggio?

Per questa attività bisognerà fare delle considerazioni sul design del robot adatte a studenti con una fascia di età [11-14], come nella prossima unità didattica.

### 5.3. Il Cerchio

A differenza delle unità precedenti dove si considerava RoboGino come un robot dato pronto ad essere programmato con dei blocchi che rappresentano azioni semplici come vai avanti, gira destra, etc., in questa possiamo renderci conto dei limiti del design dato in precedenza, proprio perché con gli elementi che abbiamo acquisito in queste due unità precedenti potremmo non essere in grado di programmare RoboGino per percorrere la circonferenza di un cerchio.

#### Scheda

<i>Argomenti</i>	Imparare a fare misure di lunghezze e angoli utilizzando il righello, il goniometro o altri strumenti es. il Robot
<i>Coding</i>	Programmare il robot per percorrere il perimetro del cerchio, livelli di astrazione del linguaggio di programmazione
<i>Robotica educativa</i>	Robot dato.
<i>Fascia d'età</i>	[11-19]
<i>Design del robot</i>	Il RoboGino
<i>Materiale</i>	Tappetino per unità didattiche incluso nel Competition kit IQ, righello, goniometro
<i>Descrizione</i>	L'insegnante propone di utilizzare il design dato di RoboGino descritto al paragrafo 3.10 del manuale (senza controller) e scrivere il diagramma di flusso e la conseguente successione di blocchi che permetta al robot di percorrere il perimetro del cerchio di diametro 1m.

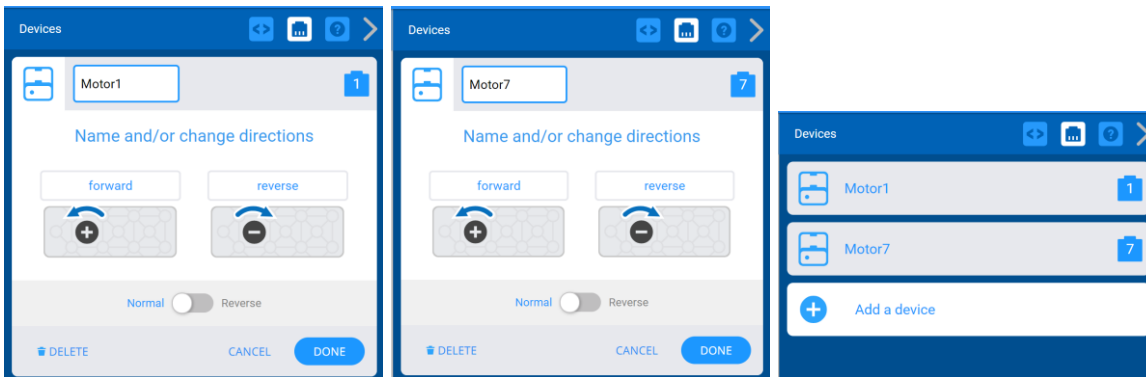
#### Svolgimento

Dopo aver svolto l'unità didattica 2 e avendo acquisito pratica sulla programmazione del RoboGino dato (come descritto a partire dal par. 4.6) si vuole ora programmare il robot per percorrere un cerchio. Siccome tra i blocchi di programmazione del Drivetrain abbiamo solo vai avanti e gira, ci si rende subito conto che dobbiamo scendere di un livello di astrazione per considerare la struttura di



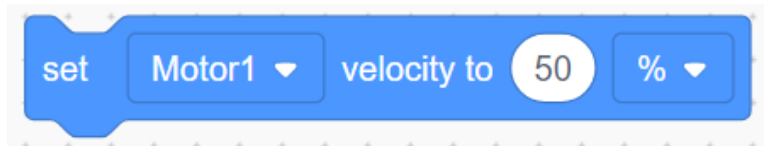
RoboGino non come un unico blocco che comprende Drivetrain e Brain come descritto nel paragrafo 4.6 ma ridefinirla in modo da poter programmare separatamente i due motori perché vadano a due diverse velocità nello stesso istante.

In sostanza va ridefinito il design del robot, non più un drivetrain predefinito ma semplicemente unità centrale e due motori.



Come vanno settate le due diverse velocità?

Notiamo che VexCode accetta in ingresso una certa percentuale della velocità massima quindi i motori dovranno avere nello stesso istante un diverso rapporto tra le velocità.



Dato un cerchio di raggio 1m, che rapporto deve esserci tra le velocità?

Considerato che la ruota esterna percorrerà un cerchio di raggio  $R = 1\text{m}$  e quella interna un raggio  $r$  nello stesso tempo  $t$ .

Spazio = Velocità \* Tempo

Spazio1 =  $2 \pi r$  con  $r$  il raggio della circonferenza più piccola

Spazio2 =  $2\pi R$  con  $R$  il raggio della circonferenza di 1m

Ponendo l'eguaglianza tra i tempi si ricava che  $\frac{V}{v} = \frac{R}{r}$  dove  $V$  è la velocità del motore connesso alla ruota più esterna e  $v$  è la velocità del motore connesso alla ruota più interna.

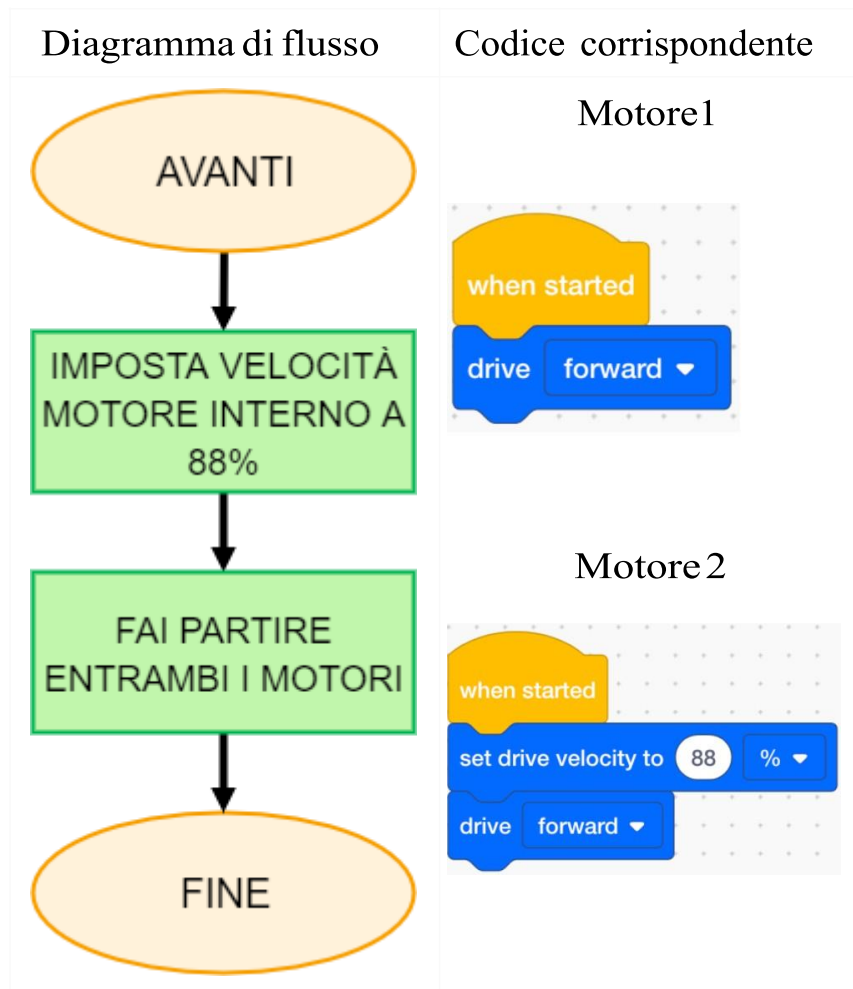
Sappiamo che  $r = R - d$ , dove  $d$  è la distanza tra le ruote.

Riusciamo quindi a trovare il rapporto tra le due velocità che ci permette di far percorrere un cerchio al nostro RoboGino.

$$v = \frac{R - d}{R} V$$

Nel nostro caso  $R = 1\text{m}$ ,  $d = 12\text{cm}$ , quindi otteniamo che  $v$  debba essere all'88% di  $V$ .

### Programmazione del Drivetrain



### Varianti e spunti di riflessione

Possiamo riprogrammare il RoboGino in modo che si fermi nel punto di partenza dopo aver percorso uno o più giri.

Perché non continuare con la fisica e calcolare la velocità di RoboGino in relazione alla velocità dei motori?

Se avessi a disposizione un dinamometro potrei anche misurare la forza di trazione del robot al variare della velocità dei motori.

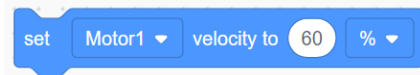
## 5.4. Relazione Spazio Tempo e Velocità

### Scheda

<i>Argomenti</i>	Relazione tra spazio tempo e velocità, moto rettilineo uniforme, velocità istantanea.
<i>Coding</i>	Programmare il Robot per avanzare di moto rettilineo uniforme per diversi istanti di tempo, imparare ad utilizzare ciclo while, blocco condizionale minore di, timer del robot brain,
<i>Robotica educativa</i>	Robot dato.
<i>Fascia d'età</i>	[11-19]
<i>Design del robot</i>	Il RoboGino
<i>Materiale</i>	Tabella spazio/tempo, cronometro
<i>Descrizione</i>	Fissato il design del robot che prevede in questo caso ruote con circonferenza di 200mm

### Svolgimento

Si programma il Drivetrain di RoboGino affinché percorra per tempi diversi un percorso rettilineo alla velocità fissata del 60%.



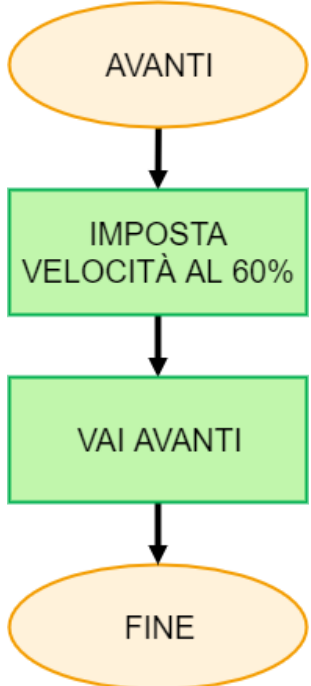
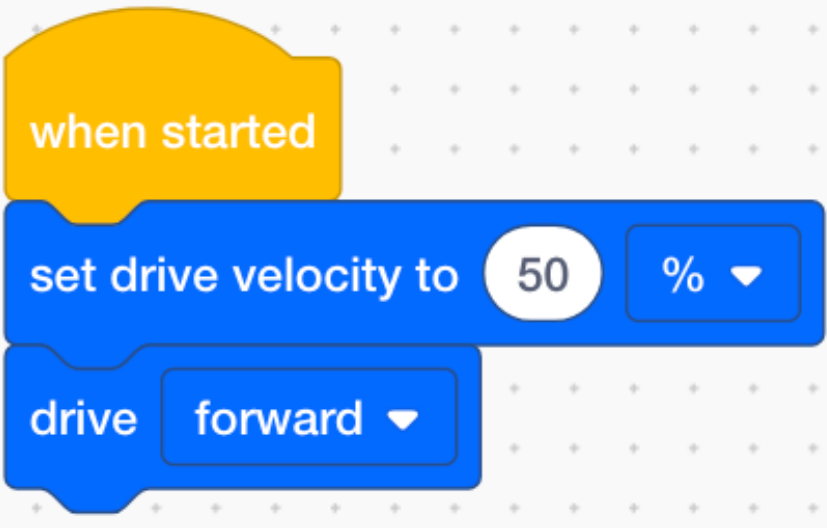
Gli alunni misureranno le distanze percorse nelle unità di tempo e le inseriranno nella seguente tabella.

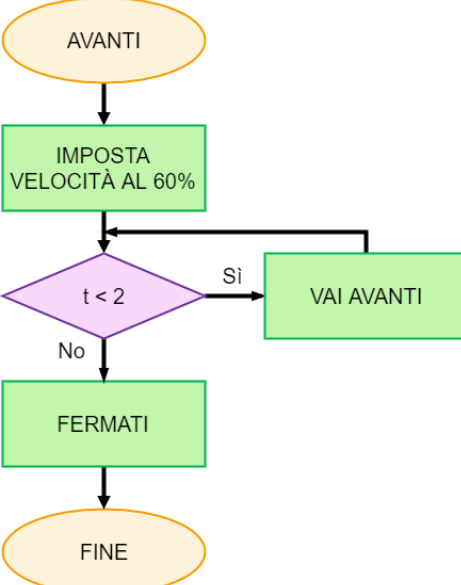
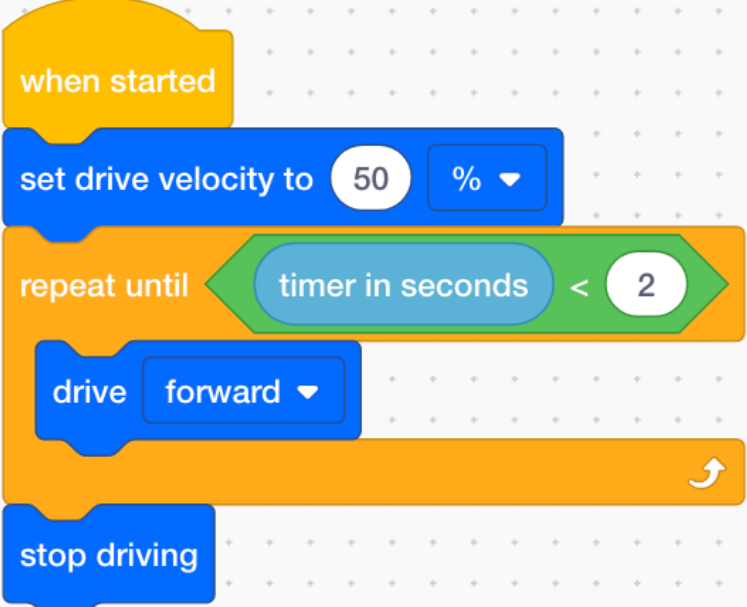
Spazio percorso in cm									
Tempo in secondi	2	4	5	7	8	10	12	15	16

Compilata la tabella, possiamo fare un grafico dove sull'asse delle ascisse mettiamo il tempo e sulle ordinate lo spazio percorso.

Aumentando il numero di istanti in cui rileviamo la posizione del robot, il grafico spazio/tempo diventa una retta. Per calcolare la velocità istantanea basta scegliere un istante  $t$  sull'asse delle  $x$  e determinare il corrispondente valore dello spazio percorso

sull'asse delle y a questo punto facendo il rapporto spazio/tempo otteniamo la velocità istantanea.

Diagramma di flusso	Codice corrispondente
<b>Utilizzo un cronometro per prendere i tempi manualmente</b>	
 <pre>graph TD; A([AVANTI]) --&gt; B[IMPOSTA VELOCITÀ AL 60%]; B --&gt; C[VAI AVANTI]; C --&gt; D([FINE]);</pre>	 <pre>when started   set drive velocity to 50 %   drive forward</pre>

<b>Utilizzo il timer del Robot Brain</b>	
 <pre>graph TD; A([AVANTI]) --&gt; B[IMPOSTA VELOCITÀ AL 60%]; B --&gt; C{t &lt; 2}; C -- Si --&gt; D[VAI AVANTI]; C -- No --&gt; E[FERMATI]; E --&gt; F([FINE]);</pre>	 <pre>when started   set drive velocity to 50 %   repeat until timer in seconds &lt; 2     drive forward   stop driving</pre>

## **Varianti e spunti di riflessione**

L'esperimento è stato fatto facendo muovere il robot su una superficie liscia, per questo motivo una volta impostata la velocità del Drivetrain, questa resta costante per la quantità di tempo di volta in volta considerato.

Come cambierebbe il grafico se il robot percorresse una superficie con diversi gradi di ruvidezza?

**Altre unità didattiche basate sull'utilizzo dei sensori saranno presto presenti su**

**[www.vexrobot.it](http://www.vexrobot.it)**

**Continua.....**