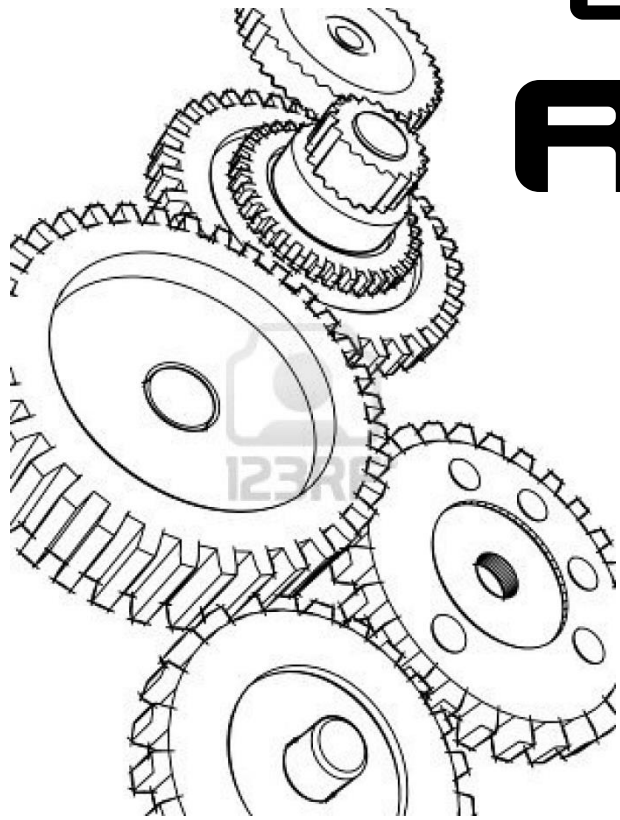
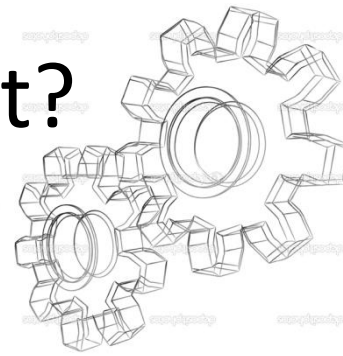


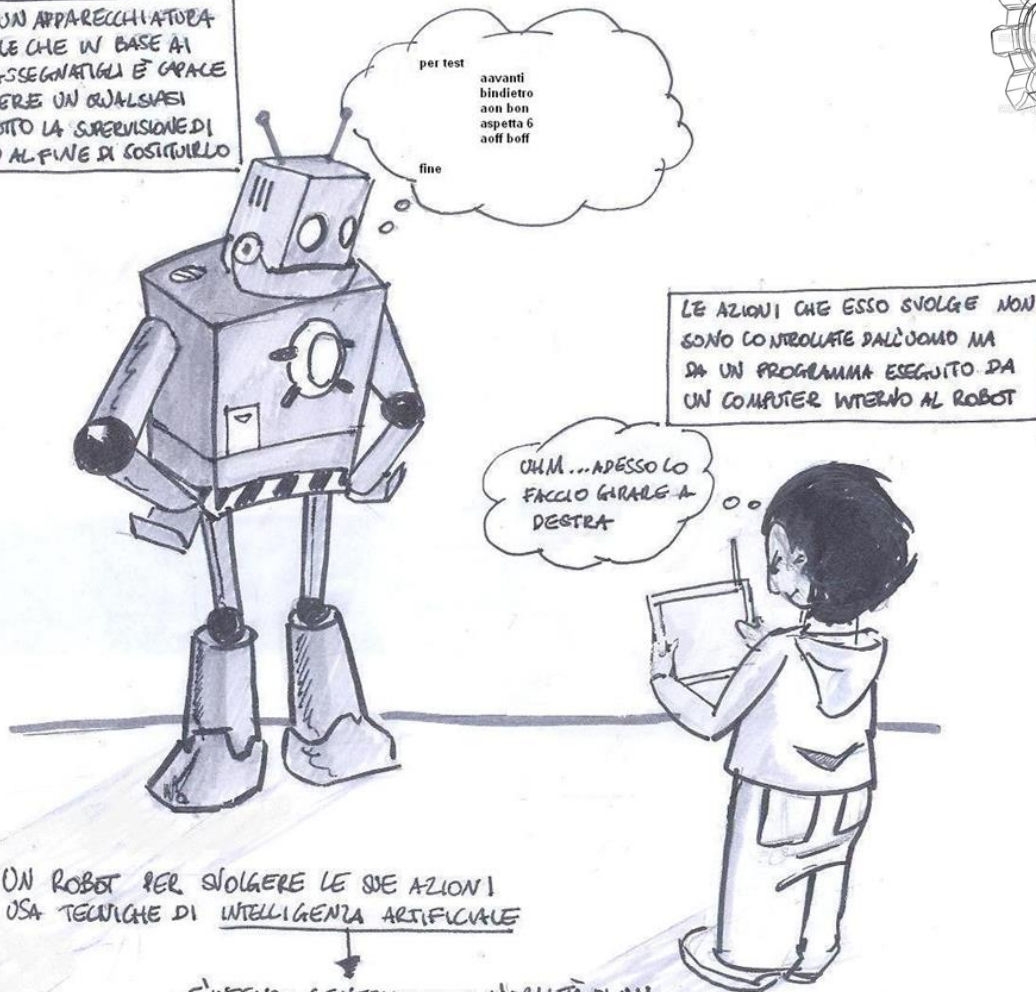
Laboratorio di Robotica



Cosa sono i robot?



IL ROBOT È UN APPARECCHIATURA ARTIFICIALE CHE IN BASE AI COMANDI ASSEGNATIGLI È CAPACE DI SVOLGERE UN QUALSIASI LAVORO SOTTO LA SUPERVISIONE DI UN UOMO AL FINE DI SOSTITUIRLO



per test
aavanti
bindietro
aon bon
aspetta 6
aoff boff
fine

LE AZIONI CHE ESSO SVOLGE NON SONO CONTROLLATE DALL'UOMO MA DA UN PROGRAMMA ESEGUITO DA UN COMPUTER INTERNO AL ROBOT

UHM...ADesso LO FACCO GIRARE A DESTRA

UN ROBOT PER SVOLGERE LE SUE AZIONI USA TECNICHE DI INTELLIGENZA ARTIFICIALE

S'INTENDE GENERALMENTE L'ABILITÀ DI UN COMPUTER DI SVOLGERE FUNZIONI E RAGIONAMENTI TIPICI DELLA MENTE UMANA. NEL SUO ASPETTO INFORMATICO ESSA COMPRENDE UN METODO PER LA RISOLUZIONE DI PROBLEMI, USANDO UN NUMERO FINITO DI PASSI

ALGORITMO



Riassumendo, possiamo considerare il Robot come un apparato meccanico in grado di compiere operazioni complesse, tipicamente pesanti, ripetitive o pericolose per gli uomini.

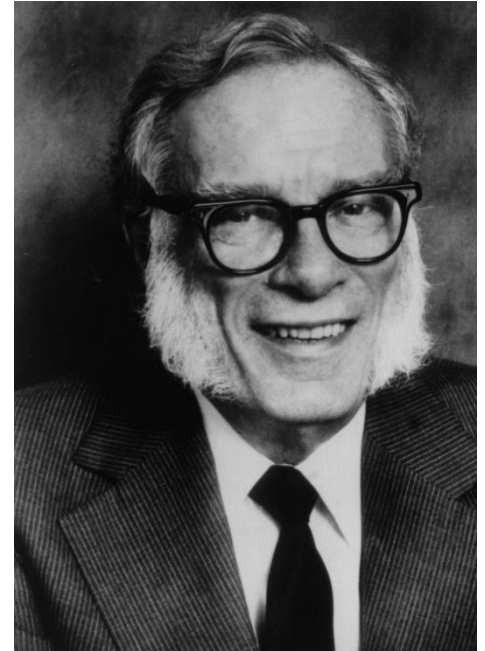
Quanti tipi di robot esistono?



Leggi della robotica

Ecco il codice d'onore dei robot:

- **Legge Zero** - **Un robot non può danneggiare l'Umanità**, né può permettere che, a causa del suo mancato intervento, l'Umanità riceva danno.
- **I** - **Un robot non può recar danno** a un essere umano ne può permettere che, a causa del proprio mancato intervento, un essere umano riceva danno.
- **II** - **Un robot deve obbedire agli ordini** impartiti dagli esseri umani, purché tali ordini non contravvengano alla Legge Zero.
- **III** - **Un robot deve proteggere la propria esistenza**, purché questa autodifesa non contrasti con la Legge Zero.



Queste leggi sono state scritte da **Isaac Asimov**, il padre della robotica.

La parola robotica proviene dal ceco *robota*, dove ha il significato di **lavoro pesante o lavoro forzato**.

... il nostro Robot



Il nostro robot è costituito principalmente da:

✘ assi 

✘ ingranaggi 

✘ mattoncini, travi 

✘ piastrine, 

✘ boccole, perni.. 



- Le travi e i mattoncini (forati nel Lego Technic) sono alla base di tutte le costruzioni Lego.
- Le travi hanno i bordi arrotondati e vengono indicate dal numero di fori, i mattoncini forati vengono indicati dal numero di perni sulla superficie.
- Perna e boccole permettono di collegare travi tra loro per creare una struttura.
- I perni neri sono leggermente più larghi mentre quelli grigi sono un poco più piccoli e possono ruotare liberamente all'interno delle travi.
- Le boccole servono per bloccare gli assi.

✘ Le travi e i mattoncini (forati nel Lego Technic) sono alla base di tutte le costruzioni Lego.

✘ Le travi hanno i bordi arrotondati e vengono indicate dal numero di fori, i mattoncini forati vengono indicati dal numero di perni sulla superficie.

✘ Perni e boccole permettono di collegare travi tra loro per creare una struttura.

- I perni neri sono leggermente più larghi mentre quelli grigi sono un poco più piccoli e possono ruotare liberamente all'interno delle travi.
- Le boccole servono per bloccare gli assi.

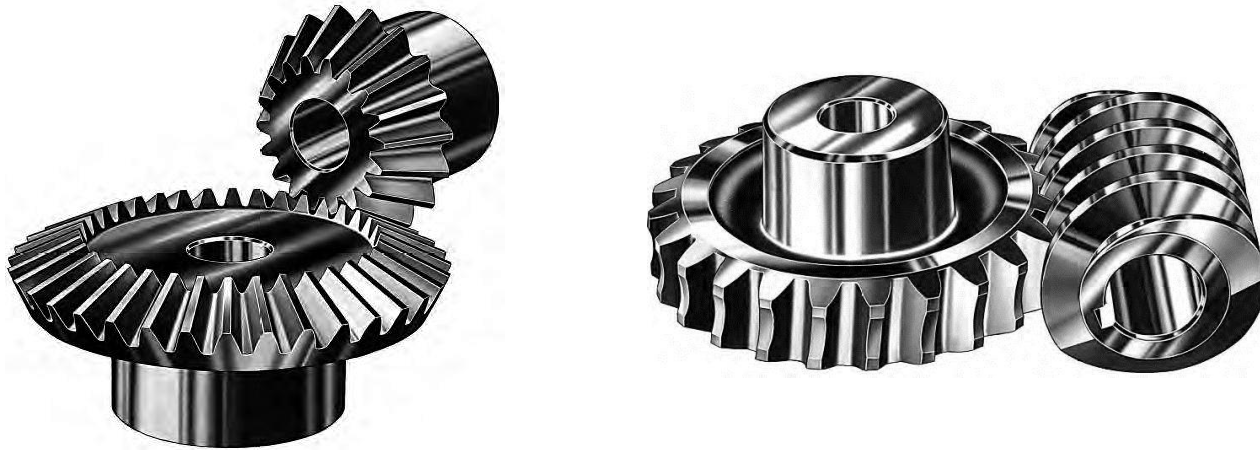
Il computer NXT con:

- 3 porte di ingresso per i motori.
- Porta USB
- 4 porte di uscita per i sensori
- Il Display e i tasti
- Arancione (invio)
- Grigio (Sposta su)
- Tasti freccia di spostamento



... ingranaggio?!

Un ingranaggio è un insieme di ruote dentate che, ingranando tra di loro, mettono rotazione due alberi ai quali possono essere collegate, ad esempio, due ruote.



Quindi un ingranaggio serve a trasmettere il moto da un albero all'altro.

Ora, come è facile intuire dall'esperienza



Potete vedere che la ruota più piccola gira più velocemente della ruota grande, questo perché le velocità di rotazione delle due ruote che chiameremo n_1 e n_2 (rispettivamente della conduttrice e della condotta) sono legate tra di loro dal cosiddetto **rapporto di trasmissione** che indicheremo con una lettera greca τ (si pronuncia «tau»):

$$n_2 = \tau n_1$$

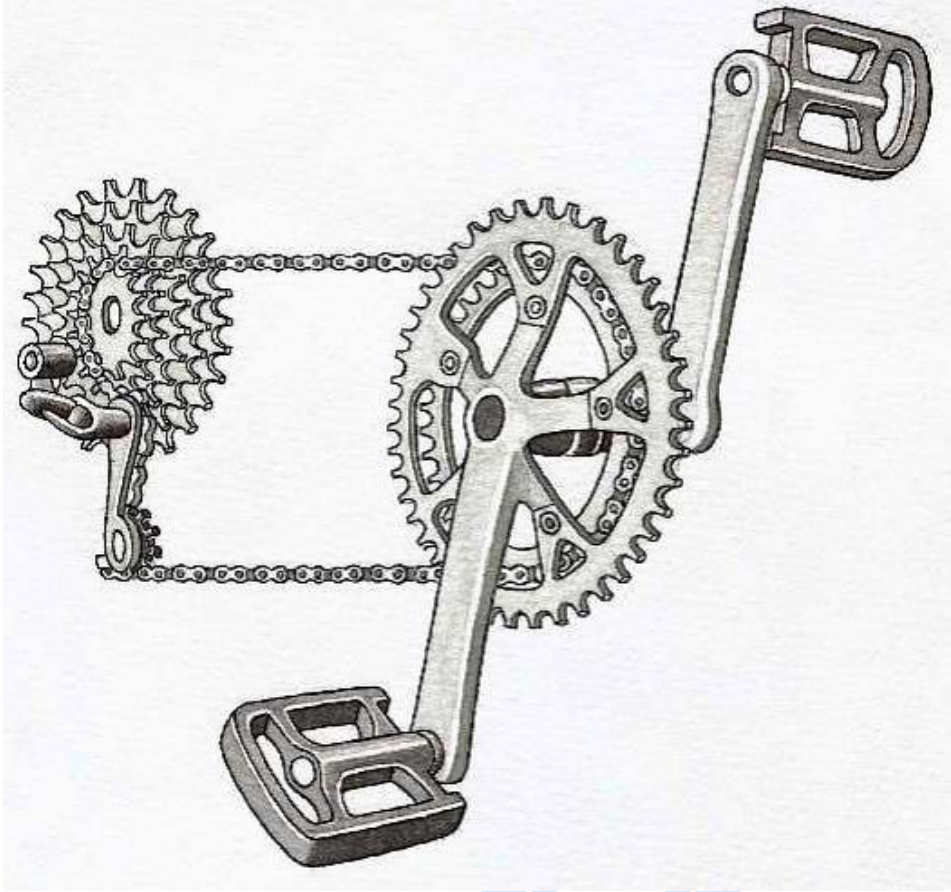
Questa formula può anche essere scritta in questo modo:

$$z_1 = \tau z_2$$

Dove z_1 è il numero di denti della conduttrice e z_2 il numero di della condotta.

Quindi il rapporto di trasmissione non è altro che il rapporto tra il numero di denti della condotta o della conduttrice o tra la velocità di rotazione della condotta e la conduttrice.

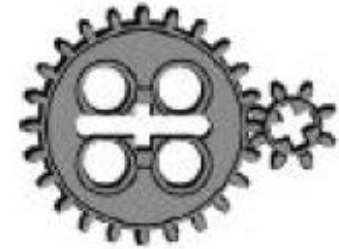
$$\frac{n_2}{n_1} = \frac{z_1}{z_2} = \tau$$



Qualche **esempio**:



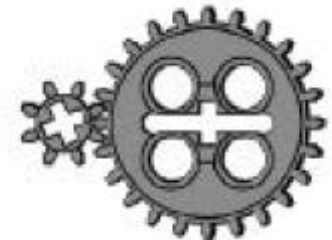
Supponiamo che la ruota condotta abbia 8 denti e quella conduttrice 24 denti
Qual'è il rapporto di trasmissione?



La ruota condotta gira 3 volte più velocemente della ruota conduttrice!!!
Un ingranaggio di questo tipo viene detto moltiplicatore.



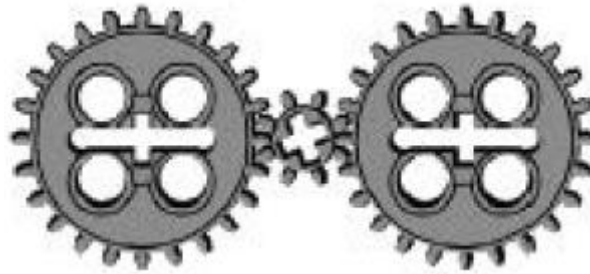
Supponiamo invece che la ruota motrice (conduttrice) abbia 8 denti mentre quella
Condotta 24. Quale sarà il rapporto di trasmissione?



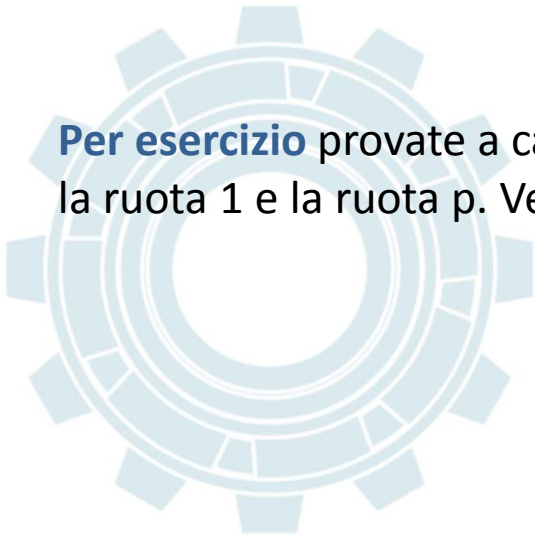
In questo caso la ruota condotta gira 3 volte più lentamente della ruota conduttrice.
Un ingranaggio di questo tipo è detto demoltiplicatore.

Notate che le due ruote dentate hanno versi di rotazione opposti, quindi una ruota gira in senso antiorario e l'altra in senso orario.

Come faccio a farle girare entrambe nella stessa direzione? Basta inserire tra la ruota conduttrice e la condotta un'altra ruota dentata



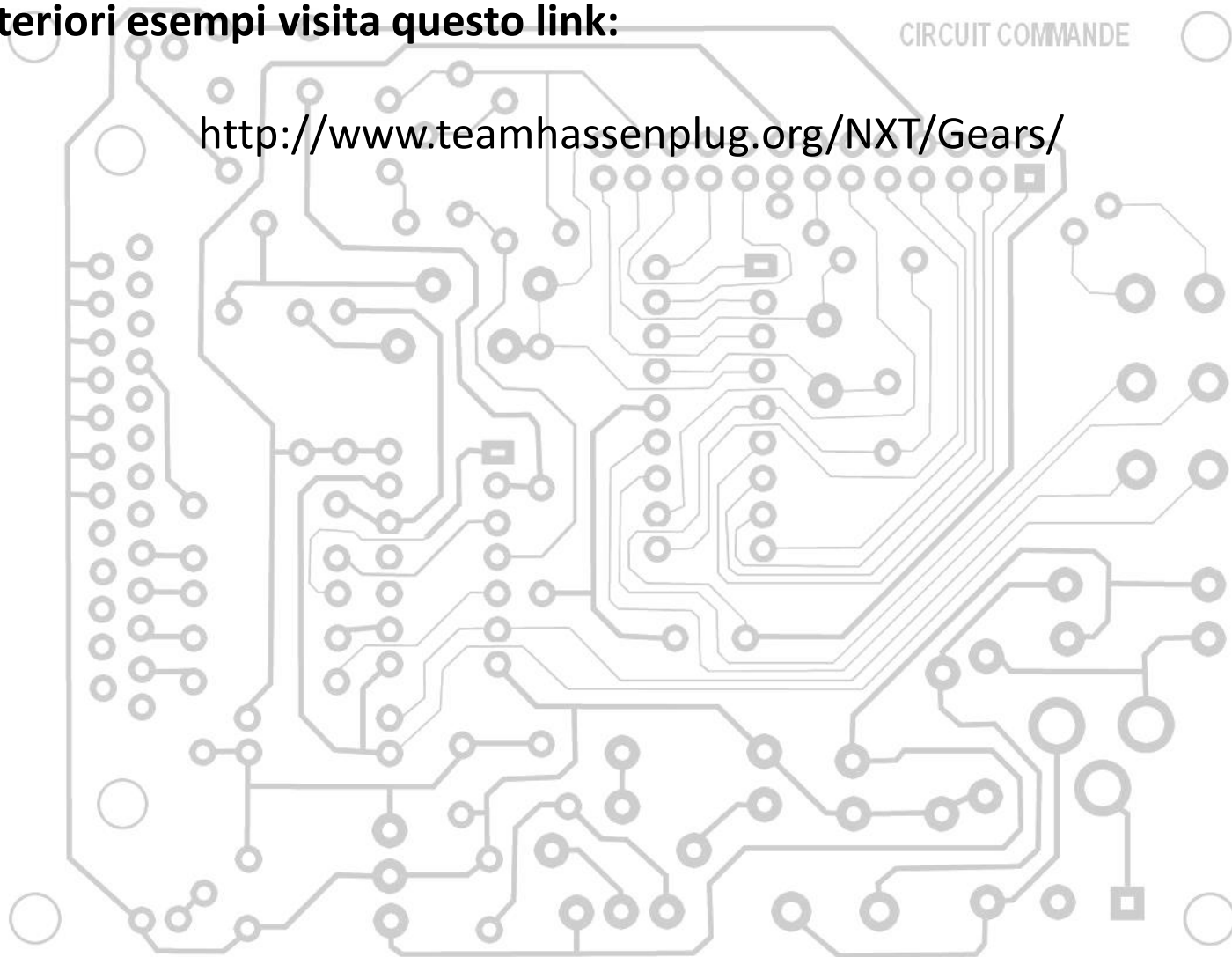
Per esercizio provate a calcolare il rapporto di trasmissione tra la ruota 1 e 2 e poi tra la ruota 1 e la ruota p. Verifica se vi è un effetto di moltiplicazione o demoltiplicazione.



Per ulteriori esempi visita questo link:

CIRCUIT COMMANDE

<http://www.teamhassenplug.org/NXT/Gears/>



... passiamo alla **PROGRAMMAZIONE**

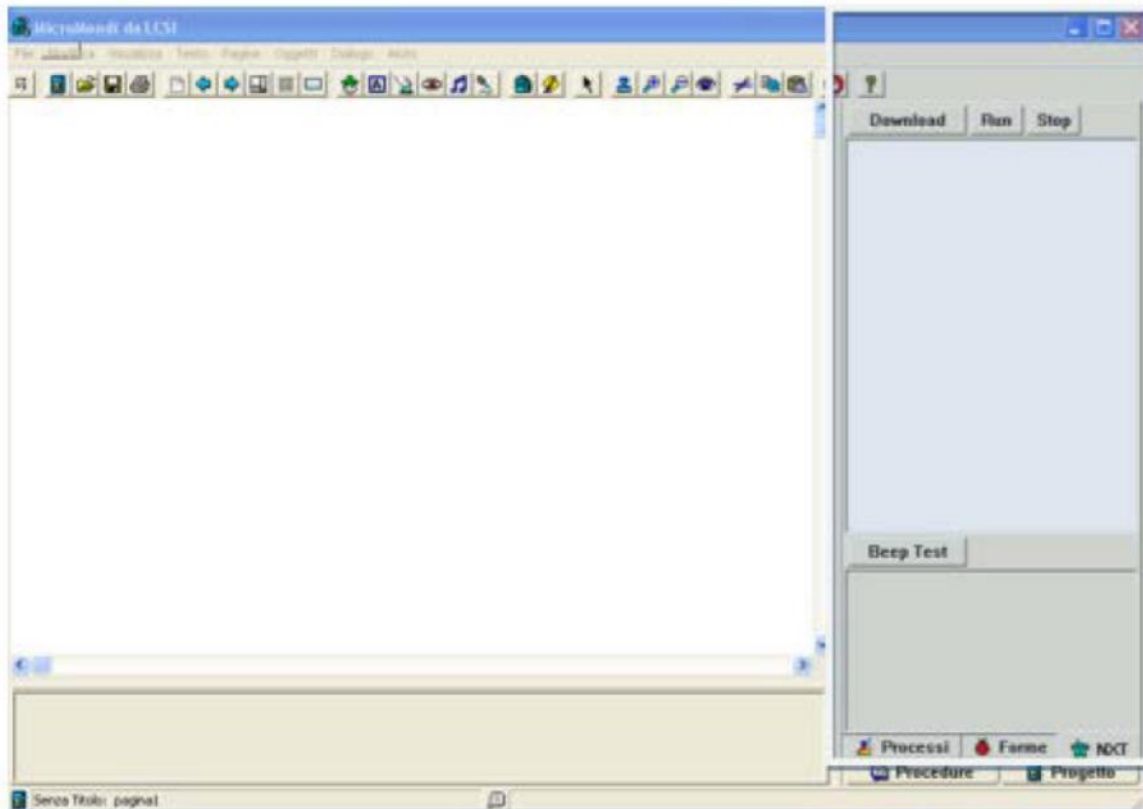
Il software MicroMondi EX Robotica

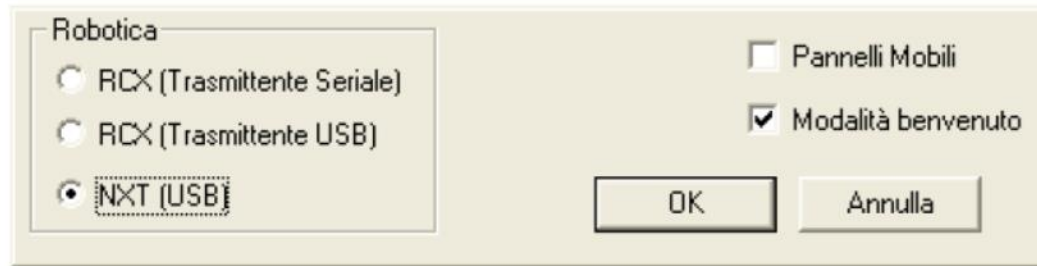
Il software MicroMondi EX Robotica è un vero e proprio linguaggio di programmazione, composto da comandi tradizionali e oggetti multimediali programmabili, inoltre ha un apposito pannello per programmare il robot NXT.

Puoi utilizzare il software anche conoscendo solo i comandi robotici e alcune **regole sintattiche** per la costruzione delle procedure.

In questa schermata scegli **Crea nuovo progetto**, facendo clic sul primo pulsante in alto a sinistra.

Quello che vedi nell'immagine successiva è l'ambiente di lavoro di MicroMondi EX Robotica ma, come ti accennavo, utilizzerai solo il pannello a destra quello indicato con NXT:





Se non ti appare il pannello NXT, fai ripartire MicroMondi EX Robotica, scegli **Preferenze** nel menu **File** e seleziona NXT (USB): ricorda, quando fai un cambiamento nel pannello Preferenze, devi riavviare MicroMondi EX Robotica.

Se l'opzione è disabilitata, devi installare i driver USB della LEGO per l'NXT, che troverai nella cartella **Mindstorms NXT driver o sul sito Lego.**

Se invece tutto è andato a buon fine sei pronto per lavorare con MicroMondi EX Robotica e con il kit Lego Mindstorm NXT.

Pannello NXT in Micromondi EX

Il pannello NXT contiene due aree o zone: **Modalità Diretta** e **Procedure**.



→ **Comandi**

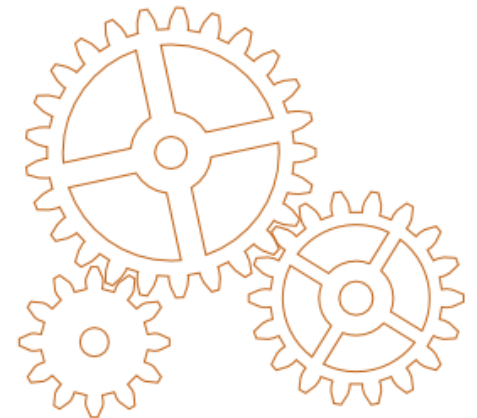
→ **Zona Procedure**
(Qui puoi scrivere le procedure da scaricare sull'NXT.
Le procedure iniziano con PER e terminano con FINE)

→ **Beep test** (verifica se è collegato l'NXT)

→ **Zona Modalità diretta**
(Per provare i comandi)

Zona modalità diretta

La zona Modalità Diretta (è una zona di prova) è quella in cui scrivi i comandi che devono essere inviati ed eseguiti immediatamente dall' NXT. Puoi scrivere anche più istruzioni su una singola linea, ma appena premi il tasto **Invio**, l'intera linea viene inviata ed eseguita dall'NXT.



Usa la zona **Modalità Diretta** per verificare se le tue istruzioni nell'NXT funzionano in accordo con le aspettative.

Se il dispositivo che hai costruito si muove (ad esempio è un veicolo con ruote), ti consiglio di metterlo su " blocchi " così che le ruote non tocchino terra quando provi le istruzioni.

Ricorda comunque che:

- alcune istruzioni (si chiamano anche comandi o primitive) non possono essere usate nel pannello Modalità Diretta;
- devi mantenere l'NXT connesso al PC (il pulsante **Beep Test** ti aiuta a controllare);
- ogni volta che premi **Invio** l'istruzione è inviata ed eseguita dall'NXT.

Prove in modalità diretta

Per effettuare le prime prove prepara un semplice robot (vedi scheda) e collega l'NXT al computer:

1. inserisci il cavo USB nel computer e nell'NXT;
2. accendi l'NXT con il pulsante arancione;
3. effettua il Beep Test (il pulsante **Beep Test** viene utilizzato per verificare la comunicazione tra il computer e l'NXT. Quando fai clic sul pulsante **Beep Test** l'NXT effettua un beep. Se non senti il beep, controlla che l'NXT sia connesso con il computer).

Scrivi nella Zona modalità diretta:

aon (premi **Invio**) - *la ruota collegata al motore A si muove*
aoff (premi **Invio**) - *la ruota collegata al motore A si ferma*

Se invece scrivi:

aon aoff (premi **Invio**) - *la ruota non si muove*

Perché?

L'esecuzione consecutiva delle istruzioni impedisce al motore di avviarsi.

Zona procedure

La zona **Procedure** è, invece, quella in cui scrivi le procedure che devono essere scaricate nell'NXT. Le procedure sono un insieme di comandi.

Regole di base per scrivere **Procedure**:

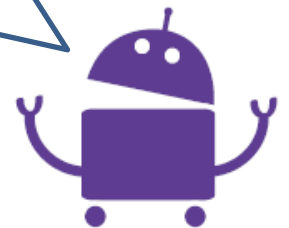
- Ogni procedura deve iniziare con **Per** seguita da un nome (a tua scelta), e terminare con l'ultima linea che deve contenere solo la parola **Fine**.

Il **nome della prima procedura**, diventa il **nome del file (*)** che troverai quando scorri i files nell'area "**Software Files**" dell'NXT. Il nome non può avere più di 15 caratteri e se già esiste un file con lo stesso nome in NXT, viene semplicemente sovrascritto. Il pulsante **Download** viene utilizzato per scaricare le procedure nell'NXT.

Il pulsante **Run** esegue l'ultimo programma che è stato scaricato nell' NXT (la prima procedura nel file).

Il pulsante **Stop** ferma qualsiasi programma che è stato lanciato con il pulsante **Run**.

() Il file è per definizione un insieme di informazioni omogenee al computer. Sono ad esempio file, parti di programmi, documenti di testo, immagini, e appunto una procedura creata con MicroMondi EX e inviata all'NXT.*





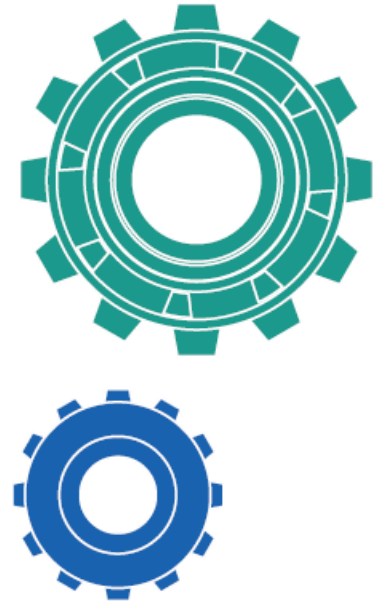
In quest'esempio:

Fai clic su **Download** per scaricare le procedure (**test** e **corsa**) nell'NXT. Il file sarà chiamato "**test**" perchè questo è il nome della **prima** procedura.

Fai clic su **Run** per eseguire la procedura "**test**".

Fai clic su **Stop** se vuoi fermare la procedura prima che finisca.

Dopo che hai scaricato il programma, puoi utilizzare altre volte i comandi (**Run, Stop**) direttamente sull'NXT premendo rispettivamente il tasto arancione e il tasto grigio.



Dopo il **download**, puoi staccare l'NXT dal computer e farlo lavorare in modo "**autonomo**" senza il cavo USB.

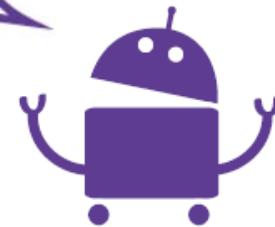
I pulsanti sull'NXT ti permetteranno di selezionare i files e di eseguire le procedure. I Files rimangono nell'NXT fino a quando non li cancelli.

I comandi di Micromondi EX Robotica

Ricorda che per dare un comando ai motori devi farlo precedere dalla lettera a cui è collegato (a es. **Aon** attiva il motore "collegato alla porta **A**"; **Bon** attiva il motore "collegato alla porta **B**"; **Con** attiva il motore "collegato alla porta **C**").

ATTENZIONE

Nel seguito farò riferimento sempre alla porta **A**, se il motore o i motori sono collegati ad altre porte il comando va modificato sostituendo la lettera corretta.



Aon - fa partire il motore collegato alla porta A

Aoff - ferma il motore collegato alla porta A

Aonper numero - attiva il motore per i decimi di secondo indicati dal numero

Aonpergradi numero - attiva il motore per il numero di gradi indicati

Acambiaverso - inverte la direzione del motore

Adaipotenza numero (da 0 a 7) - aumenta o diminuisce la potenza

Aavanti - fa muovere il motore avanti

Aindietro - fa muovere il motore nella posizione opposta

Comandi generali

Aspetta numero - fa una pausa in decimi di secondi indicati dal numero

Esempio:

Aspetta 10 - fa una pausa di 1 sec a 10/10 sec = 1 sec

Ripeti numero - esegue un numero di volte le istruzioni in parentesi

Ripetisempre - esegue sempre le istruzioni in parentesi

Ricorda: le istruzioni dopo "**Ripetisempre**" vanno in parentesi quadra.

Procedure di movimento

Connetti l'NXT al computer, verifica se MicroMondi EX è correttamente configurato (vedi il pannello NXT sul lato destro?). Attiva l'NXT e fai un **Beep Test**.

Ci sono tre comandi che fanno ruotare i motori. Uno che li fa ruotare "**sempre**", uno che li fa ruotare per un "**certo tempo**", e uno che li fa ruotare per un "**certo numero di gradi**".

Ripetisempre

Comincia con il comando più semplice: "**ripetisempre**", Scrivi nel **Pannello Modalità Diretta** i seguenti comandi . Premi **invio** alla fine di ogni linea. Guarda con attenzione ogni istruzione, puoi capire facilmente cosa sta accadendo. Se i motori non sono connessi alle Porte **A** e **B**, cambia le lettere con quelle che utilizzi:

`aon` (fa muovere il motore A per sempre... finché non scrivi l'istruzione successiva)

`aoff` (fa fermare il motore A)

`aon aoff` (il motore A parte e si ferma così velocemente che non ti accorgi del movimento!)

`aon bon aspetta 20 aoff boff`

`ripeti 5 [aon aspetta 10 aoff]`

Per un certo tempo

aonper : attiva il motore connesso alla porta A per il tempo che desideri.

Importante: aonper non funziona nel pannello **Modalità Diretta**. Pertanto gli esempi si basano sulla **procedura** test.

In questo esempio, aonper 20 attiva il motore A per 2 secondi (il numero rappresenta decimi di secondi). Prova valori differenti:

per test

aonper 20

bonper 10

fine

Per un certo numero di gradi

aonpergradi : attiva il motore connesso alla porta A e lo fa ruotare del numero di gradi indicati.

Importante: aonpergradi non funziona nel pannello **Modalità Diretta**. Pertanto, crea, scarica ed esegui questa procedura. Nell'esempio, aonpergradi 360 fa ruotare il motore esattamente di **una rotazione completa** (360 gradi), e il motore B di **mezza rotazione**.

per test

aonpergradi 360

bonpergradi 180

fine

Quale devo utilizzare?

- aonper è utile se vuoi far ruotare un motore per un tempo specificato. Va bene se stai facendo fare al robot una corsa che deve durare esattamente 30 secondi. Ma se stai costruendo un veicolo, non sai esattamente di quanto si muoverà in 30 secondi. Dipende dalla superficie, dalla frizione, dall'angolo, dal vento, ecc..
- aonpergradi è la scelta corretta se vuoi far ruotare il motore per un numero di gradi specificato. Non dipende dal materiale, frizione o angolo, aonpergradi 180, ad esempio coprirà sempre la stessa distanza

Inverti la direzione

Ecco un nuovo comando: acambiaverso (o bcambiaverso, o cambiaverso).
Acambiaverso sta per Inverti la Direzione al motore connesso alla Porta A.

Scrivi le seguenti linee nel Pannello Modalità Diretta e premi Invio alla fine di ogni linea:

```
aon  
acambiaverso  
acambiaverso  
acambiaverso  
Aoff
```

Vedrai che acambiaverso inverte **sempre** la direzione. Se il robot si muove in avanti acambiaverso lo fa muovere indietro, e acambiaverso ancora in avanti, e così via...

In questo esempio, il robot si muove e poi torna indietro esattamente alla posizione di partenza. Scrivi queste istruzioni in una procedura perché `aonpergradi` non funziona nel **Pannello Modalità Diretta**:

per test

`aonpergradi 360`

`acambiaverso`

`aonpergradi 360`

fine

Vai in questa o in quella direzione

Ci sono altri due comandi che possono stabilire la direzione dei motori, ma sono un po' diversi: aavanti e aindietro. Diversamente da acambiaverso, bcambiaverso e ccambiaverso, che invertono sempre la direzione del motore, aavanti, bavanti, cavanti **impostano** la direzione in un **determinato modo**, non necessariamente l'opposto della direzione corrente. In altre parole, se esegui aavanti molte volte, solo la prima *avrà* effetto. la seconda no, perché il motore è già in direzione "avanti". La stessa logica si applica a aindietro, bindietro e cindietro.

Scrivi i seguenti comandi nel **Pannello Modalità Diretta** e premi **Invio** alla fine di ogni linea.

Scrivi i seguenti comandi nel **Pannello Modalità Diretta** e premi **Invio** alla fine di ogni linea.

aon

aavanti (questo **può** avere un effetto, o no... se il motore è già in direzione "avanti")

aindietro (questo avrà certamente effetto)

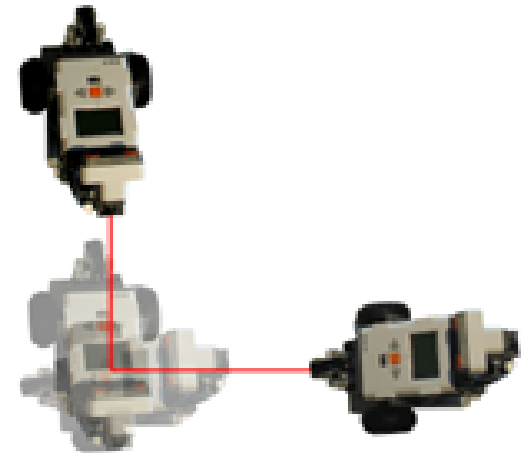
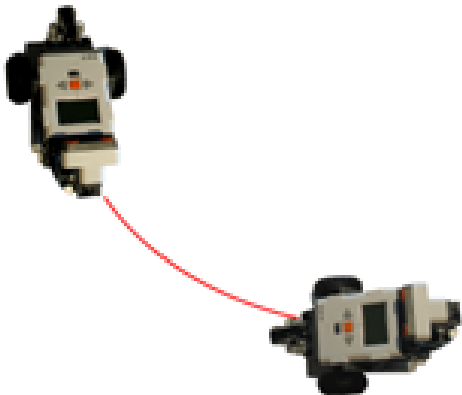
aindietro (questo certamente **non** avrà effetto, il motore è già in direzione "indietro")

Quando è meglio usare aavanti e aindietro (invece di acambiaverso)? Dipende da come è costruito il robot, potrebbe essere necessario impostare la direzione dei motori in un determinato modo per muoverli nella direzione desiderata. In questo caso, acambiaverso non è l'opzione migliore. Ogni volta che esegue acambiaverso il motore si muove nella direzione opposta. Può essere ciò che vuoi oppure no! Nel passo successivo di questo tutorial vedrai un buon esempio.

Curve strette, curve larghe...

Cerca di avere innanzitutto le idee chiare sulle curve strette e larghe. Guarda queste immagini:

Il robot fa una curva stretta nell'angolo, poi continua a muoversi.



Curve larghe: il robot percorre un grande arco mentre si muove.

Come si fa a fare una o l'altra rotazione. Pensa alle due ruote e al fatto che puoi controllarne ognuna individualmente. Fai questi due movimenti sul tavolo, trascinando manualmente il robot - guarda le ruote - cosa osservi?.

Pensa ai comandi usati nel *Passo 2*: *a*cambiaverso, *a*avanti, *a*indietro. Gioca con questi comandi. Crea, scarica ed esegui questa procedura. Usa i nomi corretti delle porte per i motori del robot (**A** e **B** nell'esempio):

per test

*a*avanti

*b*indietro

*a*on bon

*a*spetta 6

*a*off *b*off

fine

- Cosa osservi?
- Come te lo spieghi?
- Puoi far ruotare il robot attorno in un altro modo ?
- Cosa accade se cambi il valore (10) dopo "aspetta"?
- Capisci quale è l'esatto valore che ti necessita per fare un "angolo" perfetto?

Adesso dai un'occhiata alla "curva larga", quella che crea la forma di un arco. Cosa ti sembra che succeda? Chiaramente, una ruota gira più velocemente dell'altra, non va indietro. Esplora questa situazione.

Metti il robot su un blocco e prova questi comandi dal **Pannello Modalità**

Diretta:

```
aon
```

```
adaipotenza 7
```

```
adaipotenza 2
```

```
adaipotenza 4
```


`adaipotenza` cambia il livello di potenza del motore A, e "4" è il livello standard quando fai partire l'NXT. Al di sopra di 4 è più veloce (maggiore potenza) e al di sotto di 4 è meno veloce (minore potenza). Prova ancora, usando le istruzioni in una procedura. Nel caso in cui le ruote sono nella stessa direzione, e nel caso in cui non lo sono (`aavanti bavanti`) (Nota: puoi scrivere anche più di un comando per linea):

```
per test
aavanti bavanti
adaipotenza 4
bdaipotenza 7
aon bon
aspetta 10
bdaipotenza 4
aoff boff
fine
```

In questo esempio, il motore **B** riceve più potenza per un poco quando viene reimpostato a potenza 4. Prova, e cambia il valori per vedere l'effetto sulla curva.

Comandi di logica

Dal momento che il nostro simpatico NXT è dotato di sensori, questo può interagire con ciò che lo circonda.

Ad es. SE il robot mentre avanza, con il suo radar rileva un ostacolo, deve tornare indietro; oppure deve emettere un suono SE il sensore di tocco rileva un urto.

Tieni presente che il «SE» è proprio un comando di logica infatti la presenza dei sensori permette al robot di scegliere tra due opzioni o di attendere che accada qualcosa.

Sensori

Il sensore è un dispositivo che fornisce “i sensi” al robot simulando ad esempio la vista, il tatto, l’udito.

Ricorda che per dare un comando ai sensori deve farlo seguire dal numero della porta a cui il sensore è collegato.

Esempio:

Se il sensore di “**contatto**” è collegato alla porta 1 scriverai **contatto1**, altrimenti **contatto2**, **contatto3** o **contatto4**.

ATTENZIONE

Nel seguito farò riferimento sempre alla porta **1**, se il sensore è collegato ad un’altra porta dovrai sostituire il numero.



Contatto1 - Riporta **1(vero)** se è stato premuto o **0 (falso)** in posizione di riposo non premuto.

Distanza1 - Riporta il **valore in cm della distanza da un ostacolo** o dalla parete (misura la distanza da 15 cm a 2 metri)

Suono1 - Riporta un **valore da 0 a 100** che rappresenta in percentuale la quantità di suono percepito nell'ambiente vicino al sensore

Luce1 - Riporta un **valore da 0 a 100** che rappresenta in percentuale la quantità di luce riflessa dal pavimento o dall'ambiente

Temperatura1 - Riporta la temperatura in gradi

- **“Se”** - Se la condizione è **vera** (1) attiva i comandi in parentesi

Esempio:

Collega un motore alla porta **A** e un sensore di contatto alla porta **1**.

Per fermare

Aon

Ripetisempre [verifica]

Fine

Per verifica

Se contatto1 [Acambiaverso]

Aon per 20

Fine

Spiegazione:

Sono state utilizzate **due procedure**, la prima (**fermare**) fa partire il motore A e ripete sempre la seconda procedura di nome **“verifica”**.

Nella seconda procedura, ogni qualvolta viene premuto il sensore di contatto il motore A cambia il verso di rotazione per 2 sec.

- **“Se altrimenti”** - Se la condizione è **vera** (1) esegue i comandi che si trovano nella prima parentesi, altrimenti esegue quelli nella seconda.

Esempio:

Collega un motore alla porta A e un sensore di contatto alla porta 1

Per test

Aon

Ripetisempre [verifica]

Fine

Per verifica

Sealtrimenti contatto1 [Aavanti][Aindietro]

Fine

Spiegazione:

Se viene premuto il sensore di contatto il motore va avanti **altrimenti**, se non viene premuto il sensore di contatto, il motore va indietro.

- **“Aspettache”** (*riporta il valore 0 o 1*) - Segue ripetutamente le istruzioni in parentesi quadre, finché non riporta 1 (**vero**). Quando accade, **aspettache** ferma l'esecuzione delle istruzioni e la procedura passa alle istruzioni successive. Nell'esempio seguente la procedura **ferma** fa partire il motore A che si ferma non appena premi il sensore di contatto collegato alla porta 1, in tal caso viene eseguita l'istruzione successiva.

Esempio:

Per ferma

Aon

Aspettache

[contatto1]

Aoff

Fine