

# Esperimento di Robotica:

## Pick & Place con manipolatore Dobot

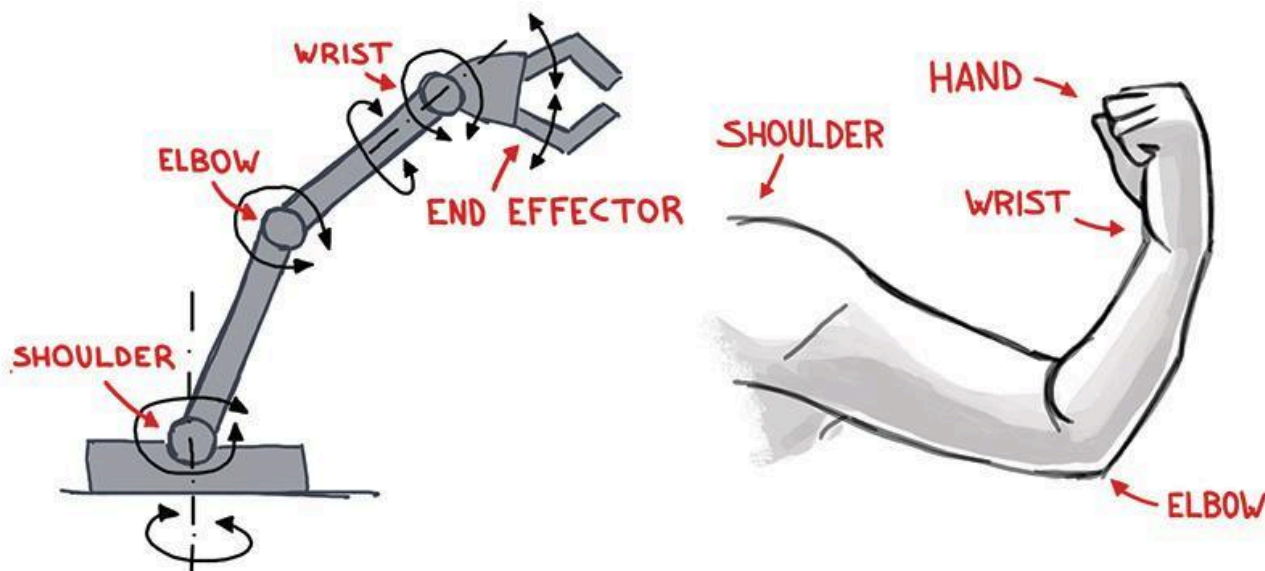
### Introduzione

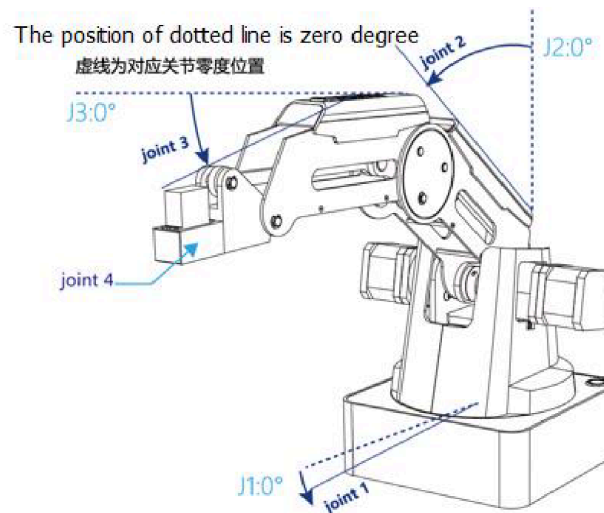
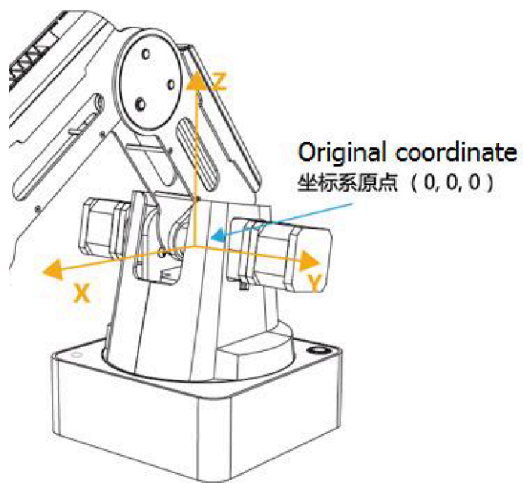
Il Pick and Place rappresenta una delle applicazioni più diffuse e fondamentali nella robotica industriale moderna. Questa operazione, apparentemente semplice, racchiude in sé numerosi concetti chiave della robotica e dell'automazione industriale, rappresentando spesso il primo passo verso la comprensione dei sistemi robotici più complessi.

### Anatomia di un Robot Manipolatore

Un robot manipolatore, come il kit didattico a 4 assi Dobot Magician (<https://www.dobot-robots.com/products/education/magician.html>), è progettato per replicare i movimenti di un braccio umano nello spazio tridimensionale. La sua struttura è composta da:

- Base: il punto di ancoraggio del robot
- Links: i segmenti rigidi che formano il "braccio"
- Joints (giunti): le articolazioni che permettono il movimento relativo tra i links
- End-effector: l'elemento terminale che interagisce con gli oggetti
- Sistema di controllo: il "cervello" che coordina i movimenti





## Applicazioni Industriali del Pick and Place

Nel mondo industriale, le operazioni di Pick and Place trovano numerosissime applicazioni, ognuna con le proprie specificità:

### 1. Industria Elettronica

- Posizionamento di componenti su PCB
- Assemblaggio di dispositivi elettronici
- Controllo qualità e sorting di componenti
- Packaging di prodotti finiti

### 2. Industria Alimentare

- Confezionamento di alimenti
- Selezione e sorting di prodotti
- Pallettizzazione di confezioni
- Movimentazione in ambienti sterili



### 3. Industria Automobilistica

- Assemblaggio di componenti
- Movimentazione di parti pesanti
- Applicazione di adesivi e sigillanti

- Posizionamento preciso di elementi



#### 4. Industria Farmaceutica

- Manipolazione di provette e campioni
- Dosaggio di precisione
- Confezionamento di medicinali
- Gestione di materiali in ambiente sterile

#### 5. Logistica e Magazzino

- Picking di ordini
- Sorting di pacchi
- Carico/scarico merci
- Organizzazione del magazzino

## **Vantaggi dell'Automazione Pick and Place**

L'implementazione di sistemi robotizzati per operazioni di Pick and Place offre numerosi benefici:

#### 1. Efficienza Operativa

- Aumento della velocità di esecuzione
- Riduzione dei tempi ciclo
- Operatività 24/7
- Ottimizzazione dei processi

#### 2. Qualità e Precisione

- Elevata ripetibilità dei movimenti
- Minimizzazione degli errori
- Controllo qualità costante
- Tracciabilità delle operazioni

#### 3. Sicurezza e Ergonomia

- Riduzione dei rischi per gli operatori
- Eliminazione di movimenti ripetitivi
- Gestione di carichi pesanti
- Manipolazione di materiali pericolosi

#### 4. Aspetti Economici

- Riduzione dei costi operativi
- Aumento della produttività
- Minimizzazione degli scarti
- Rapido ritorno dell'investimento

### Tecnologie e Controllo

Le moderne operazioni di Pick and Place possono essere arricchite da tecnologie avanzate:

- Sistemi di visione artificiale per il riconoscimento degli oggetti
- Sensori di forza per il controllo della presa
- Algoritmi di path planning per l'ottimizzazione delle traiettorie
- Interfacce uomo-macchina intuitive per la programmazione

In questo esperimento, utilizzeremo un approccio semplificato attraverso la programmazione a blocchi (sfruttando Dobot Studio), che permette di concentrarsi sulla logica del processo senza dover affrontare la complessità della programmazione tradizionale. Questo approccio didattico permetterà di comprendere i principi fondamentali del Pick and Place, creando una solida base per future applicazioni più complesse.

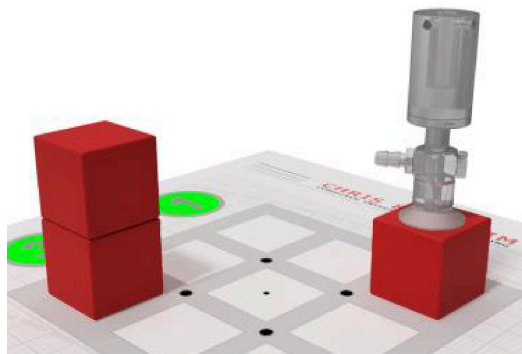
La comprensione di queste operazioni base è fondamentale per chiunque voglia intraprendere un percorso nel campo della robotica industriale, rappresentando un punto di partenza per lo sviluppo di applicazioni più sofisticate e per l'integrazione di sistemi robotici in contesti produttivi sempre più complessi.

### Scopo dell'esperimento

L'obiettivo di questo esperimento è familiarizzare con la programmazione e il controllo di un robot manipolatore didattico (Dobot Magician) per eseguire un'operazione di Pick and Place. Grazie a questa attività imparerete:

- I principi fondamentali del controllo robotico
- La programmazione dei movimenti punto-punto attraverso interfaccia a blocchi
- L'utilizzo dell'end-effector per la manipolazione degli oggetti
- L'ottimizzazione delle traiettorie di movimento
- Il lavoro di gruppo e la collaborazione nella risoluzione di problemi robotici

In particolare, dovrete provare a posizionare due cubetti uno sopra l'altro, utilizzando come *end-effector* l'elemento *suction pump*.

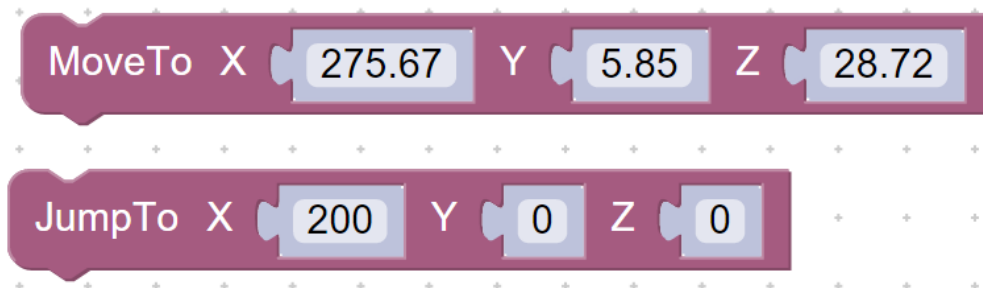


Il processo di pick & place dovrà essere progettato secondo le seguenti specifiche:

- Recuperare il primo cubetto dalla posizione A
- Posizionare il primo cubetto nella posizione B

- Recuperare il secondo cubetto dalla posizione A
- Posizionare il secondo cubetto nella posizione B (sopra il primo cubetto)

L'obiettivo per la fase di programmazione sarà utilizzare il minor numero possibile di blocchi Movimento:



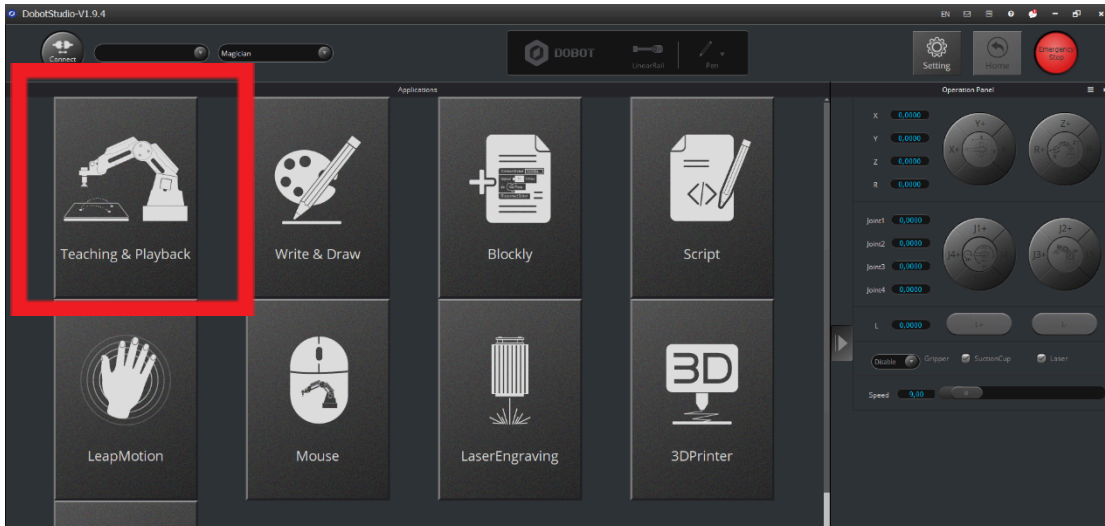
## Materiali e Strumenti

- Robot Dobot Magician
- Software DobotStudio con interfaccia di programmazione a blocchi
- Computer con sistema operativo compatibile
- Oggetti per la manipolazione (cubetto)
- Area di lavoro delimitata
- End-effector (suction pump)
- Supporto per gli oggetti da manipolare
- Scheda di documentazione per il gruppo

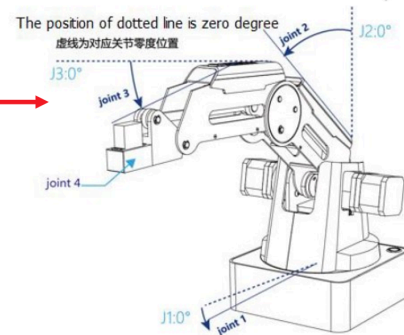
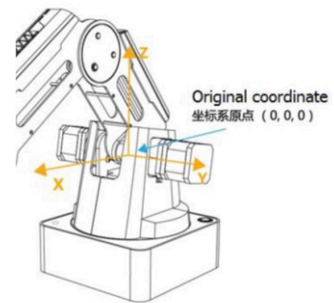
## Procedura

Durante la fase sperimentale avrete sempre il supporto di Lorenzo così da utilizzare il robot e il software in sicurezza.

1. Fase di Setup:
  - o Accendere il robot e verificare la connessione con il software
  - o Calibrare il robot attraverso la procedura di homing
  - o Preparare l'area di lavoro posizionando gli oggetti da manipolare
2. Fase di Pianificazione:
  - o Identificare i punti di prelievo e deposito
  - o Disegnare un diagramma del percorso del robot
  - o Definire la sequenza di operazioni necessarie
  - o Fase di test del percorso con la funzione Teaching and Playback di Dobot Studio

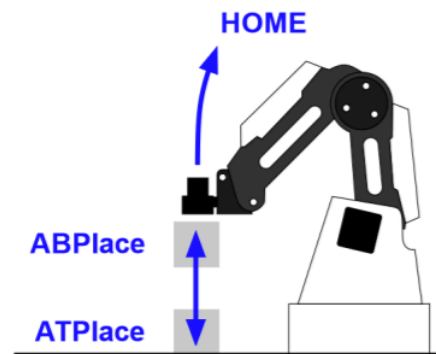
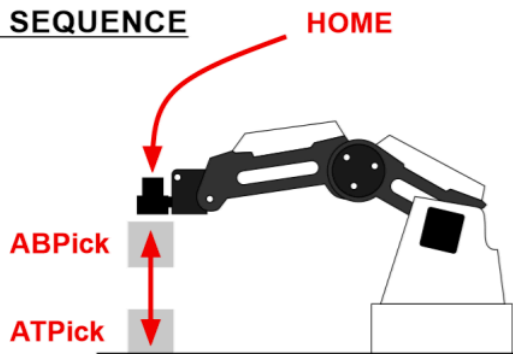


## First Overview: Operation Panel



### PICK & PLACE SEQUENCE

1. HOME
2. ABPick
3. ATPick
4. Vacuum On
5. ABPick
6. ABPlace
7. ATPlace
8. Vacuum Off
9. ABPlace
10. HOME



3. Fase di Programmazione:

- o Familiarizzare con l'interfaccia di programmazione a blocchi
- o Creare il programma base per il movimento punto-punto
- o Implementare il controllo dell'end-effector
- o Testare il programma a velocità ridotte

**CONFIGURAZIONE INIZIALE**

---

**ESECUZIONE DELLA TRAIETTORIA E ATTIVAZIONE DELLA VENTOSA**

Fase di Test e Ottimizzazione:

- Eseguire il programma a velocità intermedie
- Verificare la precisione dei movimenti
- Ottimizzare le traiettorie e la velocità (obiettivo: ridurre il più possibile il numero di blocchi movimento, individuando quali pattern ricorrono nel task)

Un esempio di ottimizzazione (per diminuire il numero di blocchi Movimento):

In order to add 35mm to the current PickX value, and assign that value to the old PickX value, replace the first value with PickX and the second value with 35.

Current X value equals itself plus a constant of 35

**Mathematically:  $PickX = PickX - 35$**

---

## Approfondimenti Pratici: Ottimizzare il Movimento del Robot

La programmazione di un robot non si limita a dirgli *dove* andare, ma anche *come* arrivarci. La scelta dei comandi di movimento è fondamentale per creare un programma efficiente, veloce e, soprattutto, sicuro.

### La Differenza tra **MoveTo** e **JumpTo**: Precisione vs. Velocità

Nel Dobot Studio, i blocchi di movimento principali che useremo sono **MoveTo** e **JumpTo**. Sebbene entrambi spostino il robot, lo fanno in modi diversi, ciascuno ottimizzato per uno scopo specifico.

- **Il Comando **JumpTo**: La Via della Velocità e della Sicurezza** Il comando **JumpTo** esegue un movimento "a salto" per spostarsi rapidamente tra due punti. Come suggerisce la documentazione, la sua logica è quella di creare un arco di sicurezza:
  1. Il robot **solleva l'end-effector** lungo l'asse Z di un'altezza predefinita (il valore di default è 20 mm).
  2. Si sposta orizzontalmente fino a trovarsi sopra le coordinate X e Y di destinazione.
  3. Si abbassa alla coordinata Z finale.
  - **Quando usarlo:** È la scelta ideale per tutti i movimenti "di transito" in uno spazio di lavoro aperto, dove non è richiesto un percorso rettilineo.
    - **Spostamenti rapidi:** Per muoversi dall'area di prelievo (**ATPick**) a quella di deposito (**ATPlace**).
    - **Ritorno alla posizione HOME:** Per tornare rapidamente a una posizione di riposo.
  - **Vantaggio:** Questo movimento è veloce e incorpora un meccanismo di sicurezza intrinseco, "saltando" sopra potenziali ostacoli bassi presenti sul piano di lavoro.
- **I Comandi **MoveTo** e **MoveDistance**: La Via della Precisione** Quando è richiesto un controllo assoluto del percorso, si usano i comandi di movimento lineare.
  - **MoveTo (Movimento Assoluto):** Esegue un movimento **lineare interpolato**, disegnando una linea retta perfetta nello spazio dal punto attuale a delle **coordinate assolute** che tu specifichi (es. vai al punto X=250, Y=50, Z=10).
    - **Quando usarlo:** È indispensabile per tutti i movimenti finali che richiedono un percorso prevedibile.
      - **Avvicinamento e prelievo:** Per scendere verticalmente sul cubetto dal punto **ATPick**.
      - **Posizionamento e rilascio:** Per scendere verticalmente e posizionare il cubetto dal punto **ATPlace**.
  - **MoveDistance (Movimento Relativo):** Esegue un movimento **lineare** rispetto alla **posizione attuale** del robot. Non gli dici "vai a", ma "spostati di".
    - **Esempio:** **MoveDistance X(10) Y(0) Z(-5)** significa: "Da dove ti trovi adesso, spostati in avanti di 10mm sull'asse X e scendi di 5mm sull'asse Z".
    - **Quando usarlo:** È utilissimo per fare piccoli aggiustamenti o per creare movimenti ripetitivi (es. in un ciclo **for**, "a ogni passaggio, spostati di 20mm a destra").

**In sintesi:** Usa **JumpTo** per "viaggiare" velocemente tra punti sicuri. Usa **MoveTo** per raggiungere con precisione le posizioni chiave del tuo programma. Usa **MoveDistance** quando hai bisogno di fare un movimento preciso relativo alla posizione corrente del robot.

---

## L'Importanza dei Punti Intermedi ("Via Points")

Osservando la sequenza di Pick & Place, noterai la presenza dei punti **ATPick** (Above Pick) e **ATPlace** (Above Place). Perché non muovere il robot direttamente sul cubetto?

Questi punti intermedi, noti come "**Via Points**" o punti di sicurezza, sono cruciali per rendere il programma **robusto e affidabile**. La loro funzione è creare una sequenza logica che sfrutta i diversi tipi di movimento:

1. **Avvicinamento Rapido e Sicuro:** Il robot usa un **JumpTo** per muoversi velocemente fino a un punto sicuro sopra il target (**ATPick**).
2. **Prelievo di Precisione:** Da **ATPick**, usa un **MoveTo** per scendere in linea retta e afferrare l'oggetto. Questo approccio verticale previene urti con oggetti circostanti.
3. **Ritirata di Sicurezza:** Dopo la presa, il robot si solleva di nuovo in verticale (**MoveTo**) fino al punto **ATPick**. Questo evita di "trascinare" il pezzo e garantisce che sia a un'altezza sicura prima del prossimo spostamento rapido.

Senza questi punti, un comando **JumpTo** diretto sul pezzo potrebbe causare una collisione, mentre usare solo comandi **MoveTo** per l'intero percorso renderebbe il programma inutilmente lento.

---

## Troubleshooting: Guida alla Risoluzione dei Problemi Comuni

Durante la programmazione e i test, qualcosa potrebbe non andare come previsto. Ecco una guida per risolvere i problemi più comuni.

- **Problema: Il robot si ferma a metà o non riesce a raggiungere il punto desiderato.**
  - **Causa Probabile:** Le coordinate (X, Y, Z) che hai inserito si trovano al di fuori dello **spazio di lavoro (workspace)** del robot. Ogni manipolatore ha un raggio d'azione limitato.
  - **Soluzione:** Verifica i valori delle coordinate. Usa la funzione **Teaching** del software per muovere manualmente il braccio fino al punto desiderato e leggi le sue coordinate reali. Assicurati che i tuoi punti siano all'interno di questi limiti.
- **Problema: Il robot esegue il movimento, ma la ventosa non afferra il cubetto.**
  - **Cause Probabili e Soluzioni:**
    1. **Comando Mancante:** Hai dimenticato di inserire il blocco **SuctionCup ON** prima del punto di prelievo?
    2. **Distanza Eccessiva:** La coordinata Z del punto di prelievo è troppo alta e la ventosa non tocca fisicamente il cubetto. Prova a diminuire leggermente il valore di Z.
    3. **Verifica Hardware:** Controlla che il tubicino dell'aria sia correttamente collegato alla ventosa e alla pompa nel corpo del Dobot.
- **Problema: Il robot urta un oggetto durante lo spostamento.**
  - **Cause Probabili e Soluzioni:**
    1. **Uso Improprio di JumpTo:** Molto probabilmente hai usato un comando **JumpTo** per un movimento che richiedeva una traiettoria controllata. Il percorso ad arco del robot ha causato la collisione. Sostituiscilo con un **MoveTo**.

2. **Punti Intermedi Inadeguati:** I tuoi punti di sicurezza (ATPick o ATPlace) potrebbero essere troppo bassi. Aumenta il loro valore Z per assicurarti che il robot sollevi l'oggetto abbastanza in alto da superare tutti gli ostacoli prima di muoversi orizzontalmente.

## Descrizione dei comportamenti da implementare

Portate a termine queste attività nel minor tempo possibile!

Tempo massimo: 45 minuti.

ID	Descrizione Task
4	Il robot riesce a posizionare correttamente 2 blocchi uno sopra l'altro (esecuzione automatica tramite ambiente Blockly)
3	Il robot riesce a posizionare correttamente 1 blocco (esecuzione automatica tramite ambiente Blockly)
2	Il robot riesce a posizionare correttamente 2 blocchi (esecuzione automatica tramite funzione Teaching & Playback)
1	Il robot riesce a posizionare correttamente 1 blocco (esecuzione automatica tramite funzione Teaching & Playback)