

---

# **ROBOTICS**

**Syllabus Versione 1.0**

## Modulo Robotics

Questo modulo definisce i concetti e le competenze fondamentali relativi ai principi di base di Robotica.

## Module Goals

Chi supera la prova d'esame per questo modulo sarà in grado di:

- Comprendere I principi base della robotica
- Conoscere I campi di applicazione
- Conoscere I componenti e I dispositivi utilizzati nei sistemi embedded
- Essere in grado di programmare una card
- Conoscere e applicare le modalità di test e debug

CATEGORY	SKILL SET	REF.	TASK ITEM
<b>1 Concetti di robotica</b>	<i>1.1 Fondamenti di fisica e meccanica</i>	1.1.1	Comprendere cosa sono le grandezze vettoriali e quali sono le principali operazioni tra vettori (somma, differenza tra due vettori, prodotto scalare e prodotto vettoriale).
		1.1.2	Sapere cosa si intende per Cinematica.
		1.1.3	Definire quali sono le grandezze fisiche fondamentali per la Cinematica, quali: Posizione, velocità, accelerazione.
		1.1.4	Sapere quali sono le tipologie di moto, quali: uniforme, accelerato, lineare, circolare.
		1.1.5	Sapere cosa si intende per Dinamica.
		1.1.6	Definire quali sono le grandezze fisiche fondamentali per la Dinamica, quali: Massa, Forza, Inerzia, Coppia.
		1.1.7	Sapere quali sono i meccanismi fondamentali utilizzati in un robot, quali: leve, pulegge, carrucole, meccanismi di trasformazione del moto.
	<i>1.2 Componenti di controllo</i>	1.2.1	Sapere cosa si intende per controllo in anello aperto e controllo in anello chiuso.
		1.2.2	Sapere quali sono i componenti di un controllo in anello chiuso, quali: controllore, sensore, attuatore, sistema controllato.
		1.2.3	Sapere cosa si intende per PLC ("Programmable Logic Controller").

CATEGORY	SKILL SET	REF.	TASK ITEM
		1.2.4	Sapere che il sistema di governo di un robot risiede nel PLC.
	<i>1.3 Sensori e attuatori</i>	1.3.1	Sapere cosa si intende per sensore e sapere che esistono diversi tipi di sensori, quali sensore di temperatura, sensore di luminosità, sensore di prossimità.
		1.3.2	Saper distinguere tra Sensori analogici e Sensori digitali.
		1.3.3	Sapere cosa si intende per attuatore e sapere che esistono diversi tipi di attuatori, quali pinza, interruttore, motore, solenoide.
<b>2. Campi di utilizzo, domini</b>	<i>2.1 Robotica fissa, robotica mobile</i>	2.1.1	Sapere quali sono le differenze tra robot fissi e robot mobile; sapere che i robot fissi sono utilizzati per la lavorazione e l'assemblaggio di componenti e che i robot mobili sono utilizzati per la movimentazione di componenti e manufatti.
	<i>2.2 Sistemi di monitoraggio e controllo</i>	2.2.1	Sapere cosa si intende per monitoraggio ambientale di grandezze fisiche.
		2.2.2	Sapere cosa si intende per controllo di impianti e processi.
<b>3. Tipologie di schede e dispositivi</b>	<i>3.1 Caratteristiche di un sistema embedded</i>	3.1.1	Sapere cosa s'intende per sistema embedded e quali sono le varie famiglie di sistemi disponibili, quali Arduino, Raspberry PI, CHIP.
		3.1.2	Sapere quali sono i componenti elettronici che costituiscono un sistema embedded: CPU, RAM, ROM, connettori di I/O.
		3.1.3	Conoscere le principali estensioni disponibili per un sistema embedded, quali display, memorie di massa, shield.
		3.1.4	Sapere quali sono le limitazioni dell'utilizzo di un sistema embedded rispetto a un normale PC: quantità di RAM, potenza di calcolo, consumi energetici.
		3.1.5	Sapere quali sono i protocolli di comunicazione supportati dalla scheda embedded, quali RS232 (seriale), I2C (display), UART (Universal Asynchronous Receiver-Transmitter), Bluetooth.

CATEGORY	SKILL SET	REF.	TASK ITEM
	<i>3.2 Prototipazione elettronica</i>	3.2.1	Sapere cosa sono i resistori, la loro funzione e il loro comportamento. Sapere cosa è la Legge di Ohm. Sapere come riconoscere il valore di resistenza di un resistore.
		3.2.2	Sapere cosa sono i diodi, la loro funzione e il loro comportamento.
		3.2.3	Sapere quali sono le configurazioni di base dei resistori (serie e parallelo) e come calcolare le resistenze equivalenti.
		3.2.4	Sapere quali strumenti di misura vengono usati per misurare le grandezze elettriche fondamentali: tester, oscilloscopio.
		3.2.5	Conoscere la differenza tra segnali digitali e segnali analogici. Sapere come si acquisiscono i segnali: ADC (Analog to Digital Converter), DAC (Digital to Analog Converter).
	<i>3.3 Dispositivi di input</i>	3.3.1	Sapere quali sono i principali dispositivi di input disponibili per un sistema embedded, quali tastiera, pulsanti, interruttori, sensori.
		3.3.2	Sapere quali sono le tipologie di pulsante (normalmente aperto, normalmente chiuso, bistabile) e i possibili stati che essi assumono, quali aperto o chiuso.
		3.3.3	Sapere che il segnale generato dai dispositivi di input deve essere compatibile con quanto la scheda si attende; conoscere il concetto di "condizionamento dei segnali".
		3.3.4	Descrivere il comportamento del potenziometro e il suo utilizzo per la generazione di segnali analogici.
	<i>3.4 Dispositivi di output</i>	3.4.1	Conoscere come è possibile generare dei segnali analogici da segnali digitali tramite PWM (Pulse Width Modulation).
		3.4.2	Saper descrivere il funzionamento di un LED.
		3.4.3	Saper dimensionare il circuito di alimentazione di un LED in base alle sue caratteristiche.
		3.4.4	Conoscere la differenza tra i principali motori elettrici (DC, stepper, brushless, servo standard e a rotazione continua).
		3.4.5	Saper controllare un motore elettrico.

CATEGORY	SKILL SET	REF.	TASK ITEM
	<i>3.5 Interfacce</i>	3.5.1	Comprendere la definizione di interfaccia di comunicazione.
		3.5.2	Saper eseguire una comunicazione seriale attraverso una porta USB.
		3.5.3	Saper eseguire una comunicazione seriale attraverso il protocollo Bluetooth.
<b>4. Programmazione di una scheda</b>	<i>4.1 Metodologie di programmazione di una scheda</i>	4.1.1	Sapere quali sono le diverse modalità di programmazione di una scheda embedded: Linguaggi, Ambienti di Sviluppo, modalità di trasferimento del programma.
		4.1.2	Sapere qual è la struttura tipica di un'applicazione software per i sistemi embedded.
		4.1.3	Saper usare un semplice IDE per scrivere ed eseguire un'applicazione su un sistema embedded.
	<i>4.2 Segnali analogici e digitali</i>	4.2.1	Leggere un valore di un segnale analogico o digitale da un PIN di ingresso.
		4.2.2	Scrivere un valore di un segnale analogico o digitale su un PIN di uscita.
	<i>4.3 Principi generali di programmazione</i>	4.3.1	Descrivere le caratteristiche del codice ben strutturato e documentato, quali indentazione, commenti appropriati, nomenclatura descrittiva.
		4.3.2	Usare commenti in un programma.
	<i>4.4 Costanti, variabili e tipi di dati</i>	4.4.1	Definire il concetto di variabile, costante e la differenza tra loro.
		4.4.2	Usare tipi di dati per una variabile o una costante: boolean, char, byte, int, long, float, double, string.
		4.4.3	Usare tipi di dati aggregati come: array, liste, tuple.
		4.4.4	Dichiarare e inizializzare una variabile.
		4.4.5	Assegnare un valore ad una variabile.
		4.4.6	Usare valori speciali delle costanti: High, Low, Input, Output, True, False.
		4.4.7	Dichiarare e inizializzare una costante.
		4.4.8	Conoscere la corrispondenza tra PIN di ingresso e uscita e variabili del software.

CATEGORY	SKILL SET	REF.	TASK ITEM
	4.5 Costrutti, funzioni e parametri	4.5.1	Scrivere e usare istruzioni logiche.
		4.5.2	Scrivere e usare istruzioni condizionali.
		4.5.3	Definire e utilizzare il costrutto "ciclo". Definire lo scopo e i vantaggi dei cicli in un programma.
		4.5.4	Definire il concetto di funzione e il suo scopo in un programma.
		4.5.5	Scrivere una funzione in un programma, assegnarvi un nome appropriato e costruire il circuito equivalente.
		4.5.6	Definire il concetto di parametro e il suo scopo in un programma.
		4.5.7	Sapere cosa s'intende per visibilità di una variabile.
	4.6 Classi, oggetti e metodi	4.6.1	Definire e usare le classi e gli oggetti.
		4.6.2	Definire e usare le proprietà delle classi.
		4.6.3	Definire e usare i metodi.
	4.7 Librerie	4.7.1	Definire il concetto di libreria.
		4.7.2	Utilizzare una generica libreria di sensori e attuatori delle schede embedded: I/O, SoftwareSerial, Servo.
	4.8 Test e debug	4.8.1	Capire i vantaggi del test e del debugging di un programma per la gestione degli errori.
		4.8.2	Comprendere e correggere gli errori di sintassi.
		4.8.3	Comprendere e correggere gli errori nella logica di un programma.