

UNIVERSITÀ degli STUDI di CATANIA

Dipartimento di Matematica e Informatica

Corso di Laurea in Informatica

PROGETTO PROGRAMMAZIONE DI SISTEMI ROBOTICI AUTONOMI E LABORATORIO

Davide Scalisi

Luigi Seminara

**Phisical Cart 2D**

PROGETTAZIONE DI UN SISTEMA ROBOTICO HARDWARE E SOFTWARE PER IL CONTROLLO DI UN CART 2D A DUE RUOTE ED A DUE ENCODER ROTATIVI.

Documentazione e progetto sono stati realizzati da Davide Scalisi (1000038316) e Luigi Seminara (1000037583) per poter conseguire l’esame di Programmazione di Sistemi Robotici autonomi e laboratorio del corso di laurea magistrale in informatica dell’anno accademico 2021/2022 con il docente Corrado Santoro.

ANNO ACCADEMICO 2021/2022

**Sommario**

Nessuna voce di sommario trovata.

# Introduzione

Lo scopo di questo progetto è quello di realizzare un sistema robotico hardware e software in grado di interfacciarsi e controllare un carrellino a due ruote (cart 2D) al fine di eseguire un algoritmo di controllo e di collision avoidance che sia in grado di fare muovere tale carrellino da un punto di partenza ad un punto di arrivo evitando al più un ostacolo che si frappone sul suo percorso.

Più nello specifico, …

## SERVE GIGI PER PARTE DI MACHINE LEARNING

# [Repository GitHub del progetto](https://github.com/LoZioo/RoboticSystems-Project)

## Tecnologie utilizzate

Il progetto si sviluppa in due principali ambiti e per ognuno sono state utilizzate le seguenti tecnologie ed i seguenti software di sviluppo:

* **Hardware**
  + Simulatore di circuiti [Falstad](https://www.falstad.com/circuit/circuitjs.html).
  + Schede di sviluppo [Arduino Nano](https://www.amazon.it/Elegoo-Scheda-Microcontrollore-ATmega328P-Arduino/dp/B0716S43Q2/ref=sr_1_1_sspa?__mk_it_IT=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crid=1Z0EFB0MPF833&keywords=elegoo+nano&qid=1661682817&sprefix=elegoo+nano%2Caps%2C137&sr=8-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUExOTFTM0I1RkFCVDVaJmVuY3J5cHRlZElkPUEwNzU5MTUwM1UxMDVGRlZDQTZJTiZlbmNyeXB0ZWRBZElkPUEwODMwNDA4MTNPWEY0MzVTU005RyZ3aWRnZXROYW1lPXNwX2F0ZiZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU=) ed [ESP32-Cam](https://www.amazon.it/Sviluppo-Fotocamera-USB-TTL-Conversione-Seriale/dp/B094FRMXSZ/ref=sr_1_6?keywords=esp32+cam&qid=1661682759&sprefix=esp32%2Caps%2C104&sr=8-6).
  + Modulo Bluetooth [HC-05](https://www.amazon.it/AZDelivery-ricetrasmettitore-wireless-Bluetooth-Comunicazione/dp/B0722MD4FY/ref=sr_1_1_sspa?__mk_it_IT=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crid=VCMQS6S8EDFM&keywords=HC-05&qid=1661683082&sprefix=hc-05%2Caps%2C93&sr=8-1-spons&smid=A1X7QLRQH87QA3&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUFUWllOVEZOVUpSWTQmZW5jcnlwdGVkSWQ9QTAyODI1NjMzUkhTQ1RLMVAzMTlYJmVuY3J5cHRlZEFkSWQ9QTA0NTE3NzFVSFo4Q1ZQUDFVM1Umd2lkZ2V0TmFtZT1zcF9hdGYmYWN0aW9uPWNsaWNrUmVkaXJlY3QmZG9Ob3RMb2dDbGljaz10cnVl&th=1) (serial over Bluetooth)
  + 2x ponte ad H singolo [LMD18200](https://www.ti.com/lit/ds/symlink/lmd18200.pdf?ts=1661683009317&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FLMD18200%253Futm_source%253Dgoogle%2526utm_medium%253Dcpc%2526utm_campaign%253Dasc-null-null-GPN_EN-cpc-pf-google-eu%2526utm_content%253DLMD18200%2526ds_k%253DLMD18200%2526DCM%253Dyes%2526gclid%253DCjwKCAjwpKyYBhB7EiwAU2Hn2ZTW0jX3bP8a9QMBnMB4dV7YMbWM8N3cSoBCGZ0LFfB1N3rDOxUEcxoCvgUQAvD_BwE%2526gclsrc%253Daw.ds).
  + 2x encoder rotativi 1000 ticks [RI-32](https://www.hengstler.de/gfx/file/shop/encoder/RI32/Datasheet_RI32_en.pdf).
  + Batteria ai polimeri di litio 11.1V (alimentazione logica).
  + Batteria al piombo 12V (alimentazione motori).
* **Firmware**
  + Linguaggio di programmazione C++.
  + IDE [VS Code](https://code.visualstudio.com/) + [PlatformIO](https://platformio.org/).
  + Framework [Arduino](https://www.arduino.cc/).
  + Sistema operativo real-time [FreeRTOS](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/freertos.html).

# Hardware

Il primo passo per lo sviluppo di questo progetto è stato sicuramente quello di progettare l’hardware necessario al pilotaggio dei due motori ed alla lettura dei due segnali in quadratura (sfasati di 90° tra di loro) in uscita dai due encoder.

## Schema funzionale dell’hardware

Più nello specifico, si può riassumere il funzionamento logico della parte hardware del sistema robotico con la seguente figura:

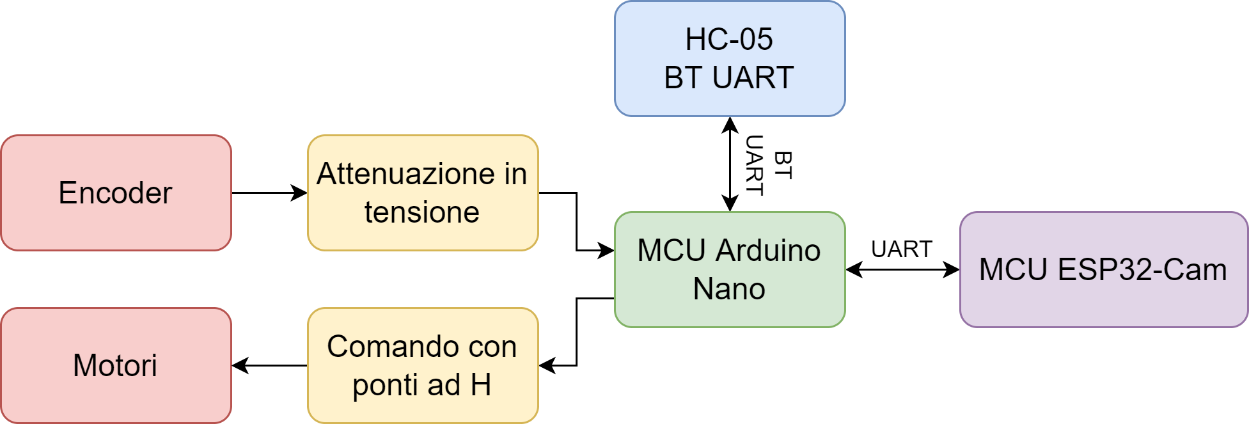


Figura : schema funzionale dell'hardware dell'intero progetto.

## Funzionamento dei due encoder

I due encoder rotativi RI-32 montati alla destra ed alla sinistra del carrellino del robot costituiscono la parte hardware fondamentale per la realizzazione di un qualsiasi sistema robotico autonomo retroazionato; il compito di questi due componenti è quello di leggere la distanza percorsa nel corso del movimento del robot.

Per ognuno di questi due componenti, vengono dati in uscita due segnali ad onda quadra in quadratura tra di loro, i quali possono facilmente essere acquisiti dalla maggioranza dei microcontrollori in circolazione.

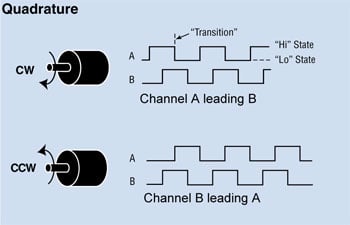


Figura : esempio di due segnali in quadratura in uscita da un encoder rotativo.

L’acquisizione e l’elaborazione di questa tipologia di segnali di natura digitale è molto semplice, in quanto basta fare uso della classica tecnica di interrupt hardware su GPIO su uno solo dei due segnali (ad esempio, solamente sul segnale “A” in figura 2) per poi andare a leggere all’interno della ISR stessa lo stato dell’altro canale (in figura 2, del canale “B”).

Tramite questo meccanismo è quindi possibile relazionare i due segnali appena acquisiti al verso del moto rotazionale che la ruota di misura in questione sta subendo (orario o antiorario); per quanto riguarda la misura dello spazio percorso, successivamente si vedrà come tradurre la serie di impulsi del segnale stesso in una variazione di posizione.

## Attenuazione in tensione dei due segnali A e B

Come suggerito dal datasheet degli encoder RI-32 utilizzati in questo progetto, è fortemente consigliata un alimentazione intorno ai 12V; questo aspetto necessita quindi di particolare attenzione in quanto i due segnali A e B, se collegati direttamente all’MCU, andrebbero a danneggiare i GPIO dell’MCU stesso, il quale è un ATmega328 operante ad una tensione nominale di 5V.

Per permettere quindi un acquisizione digitale sicura, occorre eseguire uno shift dei due segnali dal range di partenza a quello di arrivo:

Essendo i segnali di tipo unidirezionale e push-pull, il circuito di adattamento si riduce a due semplici partitori resistivi:

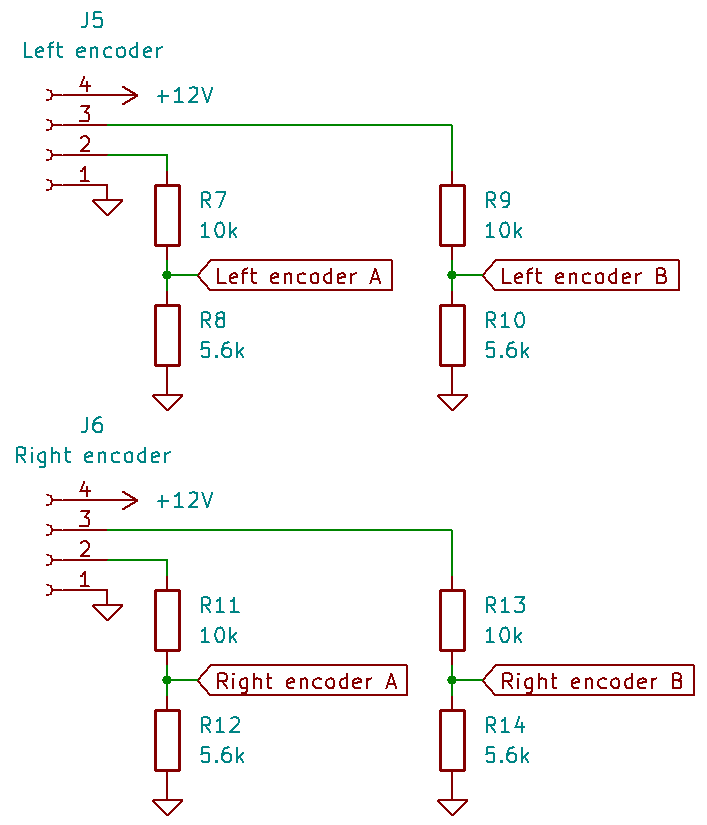


Figura : circuito di adattamento in tensione per due coppie di segnali in quadratura (in totale, quattro segnali da leggere).

Successivamente si osserverà come i segnali “Left encoder A” e “Right encoder A” sono connessi agli unici due GPIO dell’ATmega328 che supportano la funzionalità di interrupt hardware.

## Funzionamento dei ponti ad H

Per quanto riguarda la parte di controllo motori, l’adattamento in tensione ed in potenza dei segnali PWM e di direzione generati dall’MCU è effettuato tramite due integrati LMD18200; ognuno di questi integrati contiene al suo interno un ponte ad H, cioè un circuito elettrico atto a convertire un segnale PWM ed uno di direzione provenienti da un MCU in un apposito segnale PWM, adattato in tensione e potenza, pronto ad essere immesso direttamente nel motore.

L’LMD18200, in particolare, ha un utilizzo molto semplice ed ha un’implementazione interna sia dei dead-time tra i segnali di controllo degli interruttori di potenza, sia del bootstrap degli interruttori alti; questo integrato richiede quindi al suo esterno due condensatori di bootstrap per le pompe di carica e delle resistenze di pull-down sugli ingressi PWM e direzione per poter funzionare correttamente in maniera minimale.

Ognuno dei due integrati ha il compito di manovrare il suo rispettivo motore, in modo tale da permettere al carrellino un movimento controllato sia in velocità che in direzione; l’informazione viene trasmessa ai due ponti ad H tramite due segnali PWM e due segnali direzione, per un totale di quattro segnali di controllo:

* **Direzione**:
  + **LOW**: avanti.
  + **HIGH**: indietro.
* **PWM**: in base al duty cicle del segnale stesso, viene determinata la velocità di rotazione del motore, in percentuale dallo 0% al 100%.

Le resistenze di pull-down discusse prima sono necessarie in quanto al riavvio dell’MCU (conseguenza diretta di una singola riprogrammazione), tutti i GPIO si trovano in alta impedenza, cosa che in presenza di interferenze elettriche, può portare i due motori, a muoversi senza che alcun comando di movimento sia stato effettivamente dato.

## Schema elettrico della scheda Arduino Nano

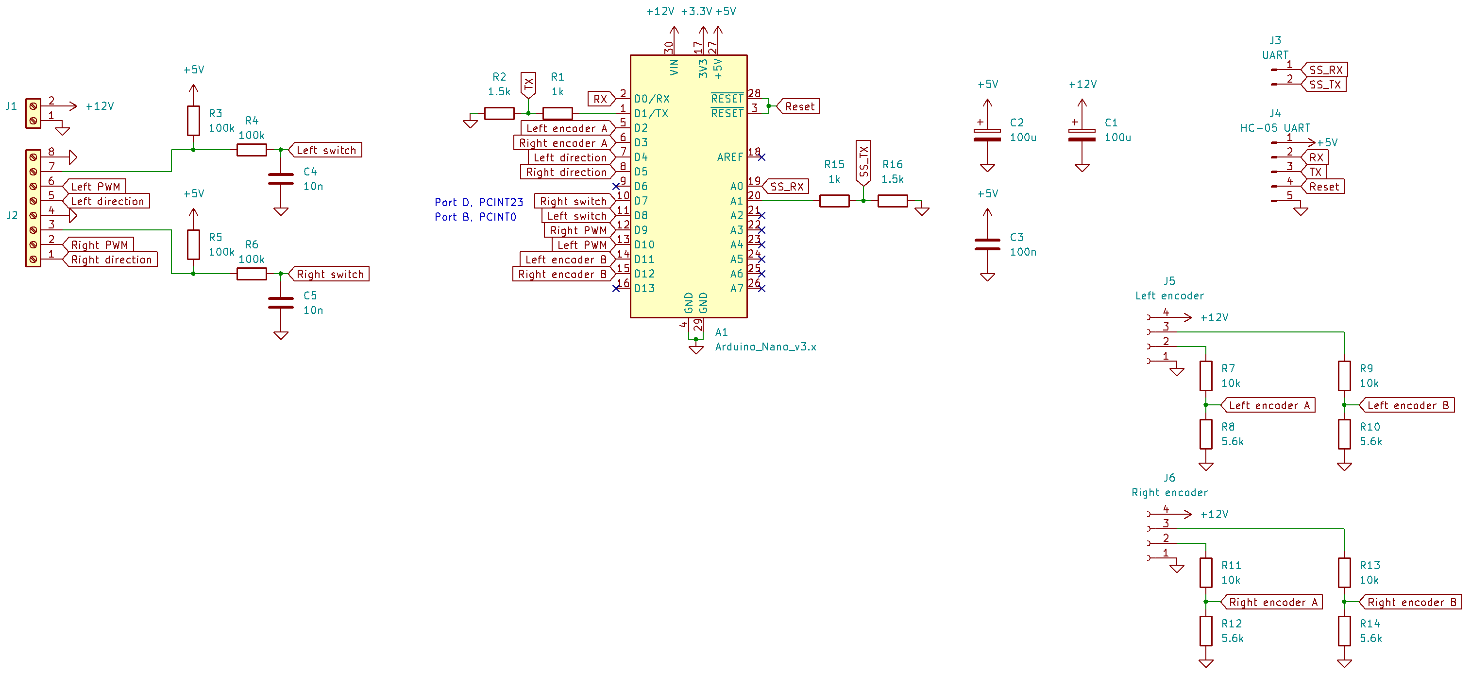


Figura : schema elettrico della scheda Arduino Nano.

## Schema elettrico della scheda dei ponti ad H

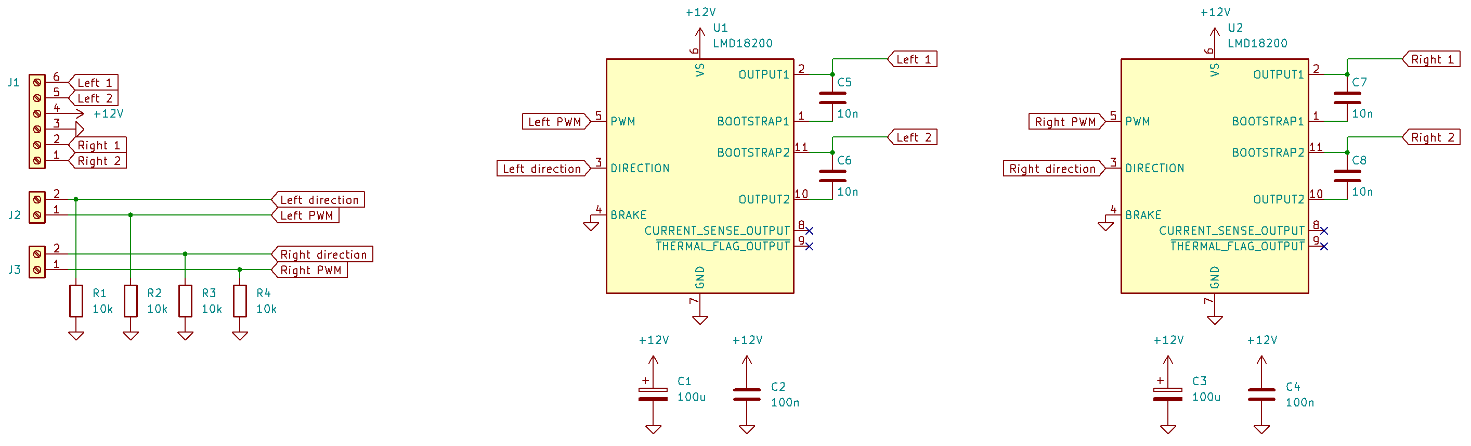


Figura : schema elettrico della scheda dei ponti ad H.