

Mobile Robot

Francesco Argentieri ^{*}, Alessandro Luchetti [†], Luca Nicolodi [‡]

Università Degli Studi di Trento, Dipartimento Ingegneria Industriale
C.D.L.M Ingegneria Meccatronica
Corso di Informatica e Programmazione

Abstract

Nel report si espongono i risultati raggiunti nello sviluppo del progetto *Mobile Robot*, un robot mobile con azionamento differenziale a due ruote che presenta queste caratteristiche di base: LineFollower; Riconoscimento incroci nel tracciato; Comandi da remoto. Si è scelto di implementarne il funzionamento mediante *macchina a stati finiti*.

1 Introduzione

Per il progetto del corso di Informatica e Programmazione si è scelto di realizzare un robot mobile con azionamento differenziale a due ruote con componenti *Makeblock*.

1.1 Hardware

Non avendo vincoli progettuali si è scelto di utilizzare una struttura semplice come riportata in figura ???. La parte elettronica è basata sulla scheda MeOrion baseboard, un modulo a ultra suono per il rilevamento degli ostacoli, due moduli per i sensori tipo MeLineFollower, un modulo bluetooth, in figura 1, il dettaglio dei collegamenti. La lista completa dei componenti in tabella 1. La scelta di adottare una coppia di sensori di linea è stata vagliata per una migliore lettura

^{*}ID: 183892 mail: francesco.argentieri@studenti.unitn.it

[†]ID: 111111 mail: @studenti.unitn.it

[‡]ID: 111111 mail: @studenti.unitn.it

del percorso, identificazione di casi particolari come ad esempio curve a 90°, incroci e per un miglior controllo dei motori.

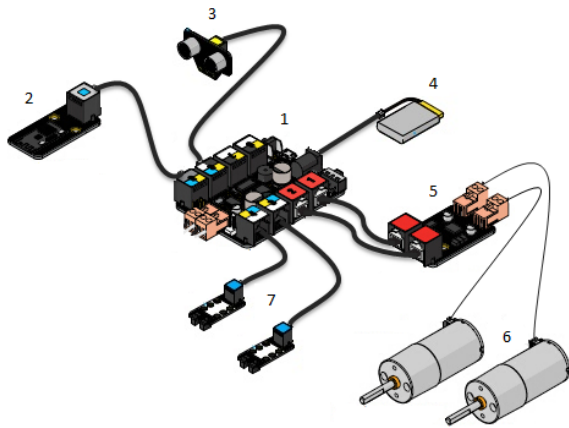


Figura 1: Schema di collegamento

Num.	Q.ta	Modulo
1	1	MeOrion
2	1	MeBluetooth
3	1	MeUltra
4	1	Batteria Li-Po
5	1	Me
6	2	MeDCMotor
7	2	MeLineFollower

Tabella 1: Lista componenti

1.2 Software

Il software che gestisce la logica del è stato pensato modulare per aumentare la facilità di realizzazione e manutenzione. Si articola su tre *layer* le librerie proprietarie Makeblock utilizzate per gestire i moduli collegati alle porte della scheda, una classe che implementa la macchina a stati finiti che gestisce gli stati dell'automa infine una classe che connette i due livelli precedenti permettendo la lettura degli input dai sensori e il verificarsi delle condizioni per passare da uno stato all'altro.

2 Soluzione del problema

Nel dettaglio si descrive il software del robot partendo dallo strato che interfaccia le periferiche con l'unità di elaborazione.

2.1 Libreria Makeblock

L'utilizzo di questa libreria è stato fondamentale perché rappresenta la base dei tre livelli e permette allo sviluppatore di istanziare i moduli fisici come oggetti e di renderli operativi fin da subito dato che viene richiesta solo la porta a cui sono collegati. Utilizzando i metodi disponibili

dalla classe è possibile ricavare gli input provenienti dai sensori, permettere la comunicazione con altri dispositivi mediante bluetooth, pilotare i motori.

2.2 Libreria Macchina a stati finiti

Questo secondo strato software è stato sviluppato durante le esercitazioni di laboratorio.

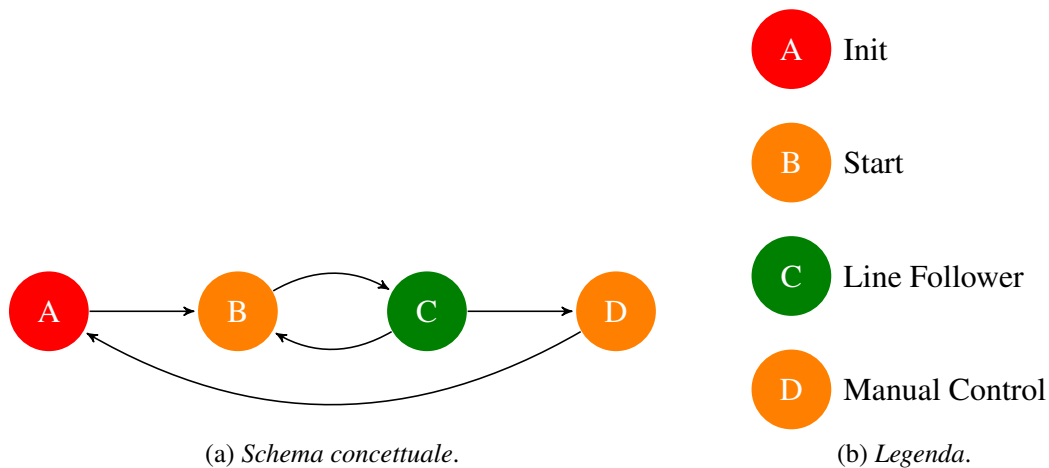


Figura 2: Macchina a stati finiti

2.3 Libreria

Infine questa libreria gestisce la risposta del robot al variare degli input durante il suo funzionamento, rappresenta lo strato di interconnessione tra i due prima analizzati. Questa classe inizializza un oggetto che ammette in input i dati provenienti dal sensore a ultrasuoni e dai sensori di linea: con il primo dato verifica la presenza di ostacoli nella parte frontale se assenti o distanti almeno 15 cm da lui, via libera a proseguire lungo la direzione altrimenti impartisce al robot il comando di ruotare sul proprio asse e proseguire nella direzione opposta. Con il secondo input proveniente dai sensori di linea riconosce il percorso affrontato secondo una serie di casi codificati secondo la tabella 2 attribuendo un errore. Tale valore viene utilizzato per correggere la velocità dei motori tramite un controller PD, si descriverà in dettaglio nella sezione 3. Vengono inoltre codificati dei casi eccezionali consultabili nella tabella 3 a cui è stato assegnato un errore 0 perché ...

		DX SX	SX DX	Errore
GO FORWARD	==	1, 2	O X X O	0
TURN LEFT VERY SOFT	==	3, 2	O X O O	1
TURN LEFT SOFT	==	1, 0	X X X O	2
TURN LEFT HARD	==	3, 0	X X O O	3
TURN LEFT VERYHARD	==	3, 1	X O O O	4
TURN RIGHT VERY SOFT	==	1, 3	O O X O	-1
TURN RIGHT SOFT	==	0, 2	O X X X	-2
TURN RIGHT HARD	==	0, 3	O O X X	-3
TURN RIGHT VERYHARD	==	2, 3	O O O X	-4
NO LINE	==	3, 3	O O O O	5
CROSS	==	0, 0	X X X X	6

Tabella 2: Codifica input dei sensori di linea (X = tracciato nero; O = sfondo bianco)

		DX SX	SX DX	Errore
GO FORWARD bis	==	2, 1	X O O X	0
EXCEPTION1	==	2, 0	X X O X	0
EXCEPTION2	==	0, 1	X O X X	0
EXCEPTION3	==	2, 2	O X O X	0

Tabella 3: Codifica CASI ECCEZIONALI (X = tracciato nero; O = sfondo bianco)

Quando il robot raggiunge un incrocio si arresta uscendo dallo stato di line follower e passa nello stato idle attendendo istruzioni dall'utente che può scegliere dalla maschera disponibile sul terminale, quindi inviare il comando tramite connessione bluetooth. Il robot torna nello stato di linefollower.

3 Controller PID

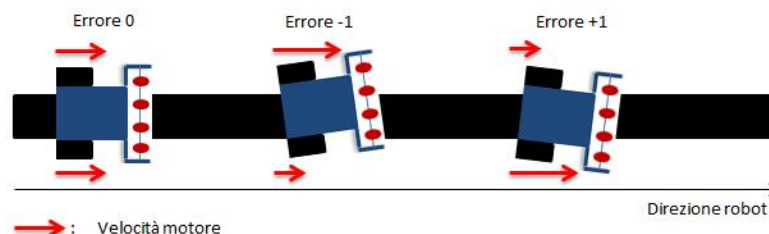


Figura 3: prova

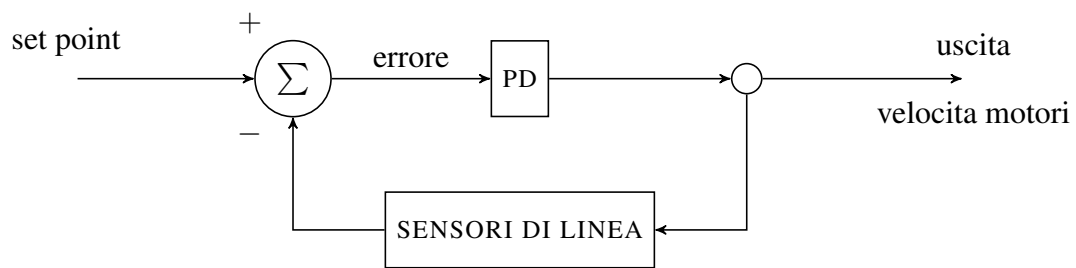


Figura 4: prova

4 Conclusione

Riferimenti bibliografici

1. <http://makeblock.com/en>.
2. <http://en.cppreference.com/w/>.
3. M. Schmidt, *Il manuale di Arduino*, Guida completa (Apogeo, 2011).