

Report di Laboratorio di Making

Daniele Romanella - daniele.romanella@studio.unibo.it - 0001170158

22 luglio 2025

1 Introduzione

Il progetto prevede la realizzazione di un veicolo telecomandato tramite smartphone, utilizzando una connessione Bluetooth e un'applicazione dedicata. A differenza delle tradizionali auto giocattolo, il sistema proposto è concepito come una piattaforma espandibile in pochi passi, consentendo l'integrazione di funzionalità avanzate e personalizzabili (i.e. Computer Vision, guida autonoma, etc.), ampliando così le potenzialità d'uso.

1.1 Risultati

Nelle **figure 1, 2, 3 e 4** sono rappresentati rispettivamente i risultati del progetto: la macchina con la carrozzeria montata (vista frontale), la macchina con la carrozzeria montata (vista posteriore), la struttura senza carrozzeria e l'interfaccia dell'applicazione per il controllo tramite smartphone.



Figura 1: La macchina con la carrozzeria montata. (vista frontale)

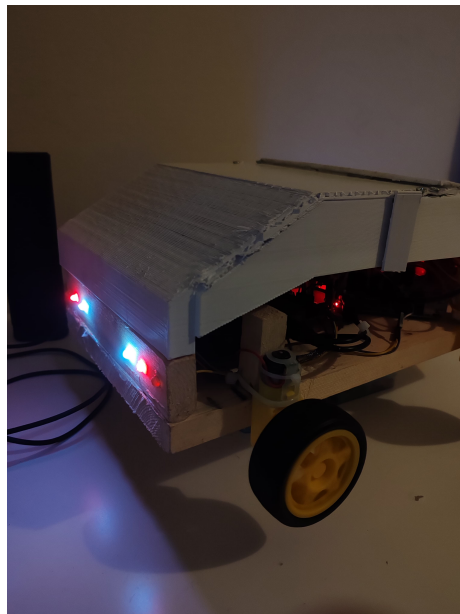


Figura 2: La macchina con la carrozzeria montata. (vista posteriore)

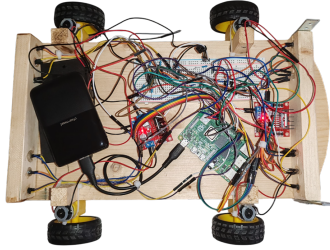


Figura 3: La macchina senza carrozzeria.

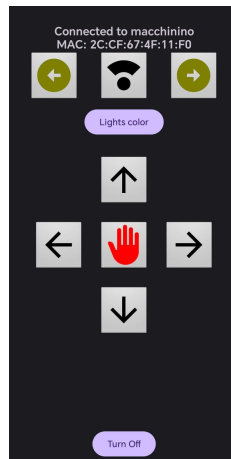


Figura 4: Applicazione android per controllare il progetto

2 Costi ed Analisi di mercato

Il costo complessivo sostenuto per la realizzazione del progetto ammonta a 62,62€. Tuttavia, sarebbe stato possibile contenere ulteriormente la spesa ricorrendo a fornitori con tempi di consegna più lunghi: in tal caso, il costo minimo teoricamente raggiungibile, considerando esclusivamente prezzi al dettaglio e l'acquisto di tutti i componenti ex novo, sarebbe stato pari a 55,41€.

Per quanto riguarda i prodotti concorrenti, i prototipi in grado di offrire funzionalità analoghe — ovvero veicoli controllabili via Bluetooth — presentano generalmente un costo elevato e una reperibilità limitata sul mercato. Il presente progetto si distingue non solo per il costo contenuto, ma anche per la possibilità di effettuare il deployment di applicazioni aggiuntive, come algoritmi di visione artificiale, grazie all'integrazione di una Pi Camera.

Nel confronto con le comuni auto giocattolo telecomandate, il prototipo

sviluppato risulta più costoso; tuttavia, offre funzionalità avanzate generalmente assenti nei prodotti commerciali, come il controllo individuale delle luci.

Il target di riferimento è duplice: da un lato, si rivolge a utenti esperti e appassionati di tecnologie embedded, che ricercano una piattaforma flessibile per sperimentazioni avanzate (es. computer vision, guida autonoma); dall'altro, il dispositivo risulta facilmente utilizzabile anche da un pubblico non esperto, in quanto il sistema è perfettamente operativo dopo il semplice abbinamento Bluetooth, senza necessità di ulteriori configurazioni.

2.1 Materiale usato

La **Tabella 1** riporta i materiali impiegati nel progetto. Quando compatibile, alcuni componenti sono stati riutilizzati da precedenti realizzazioni, con l'obiettivo di ottimizzare l'utilizzo delle risorse. Nel presente report non sono stati considerati i costi di ammortamento degli strumenti impiegati, assumendo per semplicità una disponibilità illimitata degli stessi.

Componente	Prezzo	Provenienza
Raspberry Pi 4 (4 GB)	Riutilizzato	-
Legno per pianale (20 mm)	5€	Negozi di fai da te
Led, breadboard e cavi	19,99€	Amazon.it
Motori (6V) e ruote	12,99€	Amazon.it
L298N	3,49€ (x 2)	Amazon.it
Portabatterie	1,33€ (x 2)	Amazon.it
Powerbank	Riutilizzato	-
Stampa carrozzeria superiore	15€	-
Scheda SD	Riutilizzato	-

Tabella 1: Tabella dei componenti con prezzo e provenienza.

2.2 Possibili ottimizzazioni

I materiali elencati nella **Tabella 1** sono stati selezionati in base al buon rapporto qualità-prezzo e alla loro pronta disponibilità. La **Tabella 2**, invece, riporta alternative a costo inferiore, ma caratterizzate da tempi di approvvigionamento più lunghi, oppure da materiale che offre meno performance ma che sarebbe più indicato per questo scopo.

¹Richiede saldatura degli headers

²È stato cercato un kit che comprendesse due LED RGB, esattamente come nel progetto. Una variazione sarebbe quella di usare dei led non RGB per risparmiare ulteriormente

³Rendendo alcune parti meno spesse.

⁴Al momento della redazione di questo documento i tagli da 8GB, 16GB, 32GB e 64GB hanno lo stesso importo.

Componente	Prezzo	Risparmio	Provenienza
Raspberry Pi Zero 2W ¹	21,59€	-	AliExpress
Headers da saldare	0,87€	-	AliExpress
Legno per pianale (20 mm)	1,50€	3,50€	Negozi di fai da te
Led, breadboard e cavi	10,88€	9,11€	TEMU ²
Motori (6V) e ruote	0,87€ (x 2)	11,25€	AliExpress
L298N	0,87€ (x 2)	5,24€	AliExpress
Portabatterie	1,98€	0,68€	Amazon.it
Powerbank	9,24€	-	Temu
Stampa carrozzeria superiore ³	5€	10€	Prodotta in autonomia
Scheda SD (64 GB) ⁴	0,87€	-	AliExpress

Tabella 2: Tabella dei componenti con prezzo e provenienza.

3 Implementazione hardware

In questa sezione viene descritta l'implementazione hardware del progetto. Le sottosezioni successive approfondiscono la realizzazione del pianale, della carrozzeria e l'integrazione dei componenti elettronici.

3.1 Pianale

Il pianale inferiore presenta una lunghezza di 37cm, una larghezza di 19cm e uno spessore di 20mm. La sagoma è stata realizzata mediante taglio manuale con seghetto per legno. La curvatura frontale è stata ottenuta utilizzando un utensile flessibile dotato di disco abrasivo, lo stesso impiegato anche per la levigatura della superficie. Tale operazione ha permesso di ottenere una finitura più liscia, riducendo il rischio di schegge e migliorando la maneggevolezza del componente.

3.1.1 Alloggio luci

Gli alloggiamenti per i LED sono stati ricavati da una porzione residua della tavola di legno utilizzata per il pianale. Ogni alloggio misura 15mm di lunghezza, 19cm di larghezza e 35mm di altezza. Sono presenti fori per l'inserimento dei LED, realizzati con trapano e punta da 5mm di diametro. I LED sono stati posizionati all'interno dei fori e fissati mediante colla a caldo; successivamente sono stati collegati i cavi necessari per il cablaggio e l'integrazione con il circuito di controllo.

3.1.2 Componenti di supporto per i motori

I supporti destinati al fissaggio dei motori presentano una lunghezza e una larghezza di 20mm, con un'altezza pari a 5cm. Essi sono stati montati sulla base del veicolo mediante una vite. Il foro necessario per l'inserimento della vite è stato realizzato con l'ausilio di un trapano, utilizzando una punta da 7mm di diametro.

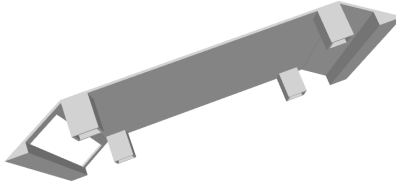


Figura 5: Carrozzeria finale, vista laterale. (Progetto)

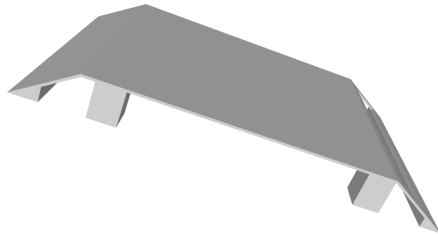


Figura 6: Carrozzeria finale, vista superiore. (Progetto)

I motori sono stati ancorati ai supporti tramite l'utilizzo della colla a caldo e una fascetta in plastica (fascetta da elettricista), garantendo un montaggio stabile.

3.2 Carrozzeria

La carrozzeria del veicolo è stata realizzata tramite stampa 3D. Sono stati sviluppati due progetti distinti:

- **Il primo progetto**, effettivamente stampato e implementato, presenta diverse criticità legate a una progettazione non ottimale. È visibile nella **figura 1** e nella **figura 2**.
- **Il secondo progetto**, considerato definitivo, non è stato stampato a causa dell'eccessivo peso che avrebbe comportato e degli elevati costi di stampa. Una trattazione più approfondita di tali problematiche è riportata nella sezione *Problematiche*. Il modello relativo è illustrato nella **Figura 5** e nella **Figura 6**.

3.3 Circuito

L'implementazione del circuito deve rispettare i seguenti diagrammi Fritzing: Uno per i motori ed uno per le luci.

Nei circuiti si fa notare che i motori hanno un voltaggio di 6V. Per quanto

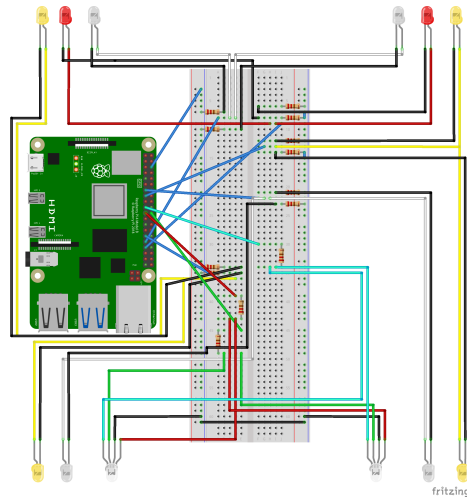


Figura 7: Circuito relativo alle luci.

riguarda le luci RGB del tipo *Common anode*. Le immagini dei circuiti sono in **Figura 7** e in **Figura 8**.

4 Implementazione software - Raspberry

Il server RFCOMM e il processo responsabile della gestione dei PIN GPIO sono entrambi implementati in Python. L'avvio dei due servizi è gestito tramite **systemd**, al fine di garantirne l'esecuzione automatica all'accensione del dispositivo. Il file di configurazione utilizzato per **systemd** è riportato nella sezione *Artefatti e materiale utile*.

4.1 Server RFCOMM

Il server RFCOMM implementato espone una serie di comandi che possono essere inviati tramite connessione Bluetooth per controllare il veicolo. Le chiamate disponibili sono le seguenti:

- **MOVE_STOP**: Arresta il movimento del veicolo e attiva la luce di STOP.
- **MOVE_LEFT**: Effettua una rotazione verso sinistra (le ruote sinistre si muovono in retromarcia, quelle destre in avanti).
- **MOVE_RIGHT**: Comando duale a **MOVE_LEFT**, ruota il veicolo verso destra.
- **MOVE_BACK**: Attiva la retromarcia, accendendo contemporaneamente le luci di STOP e di retromarcia.
- **MOVE_AHEAD**: Avvia il movimento in avanti.

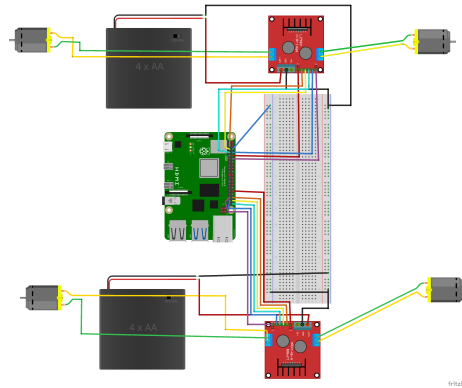


Figura 8: Circuito relativo ai motori.

- `TOGGLE_TR_LF`: Attiva o disattiva l'indicatore di direzione sinistro.
- `TOGGLE_TR_RG`: Attiva o disattiva l'indicatore di direzione destro.
- `TOGGLE_HIGH_LIGHTS`: Attiva o disattiva le luci alte.
- `LIGHTS_R_G_B`: Imposta l'intensità dei LED anteriori nei tre canali RGB. I parametri R, G e B devono essere interi compresi tra 0 e 255.
- `SYSTEM_SHUTDOWN`: Spegne il sistema (Raspberry Pi).
- `SYSTEM_REBOOT`: Riavvia il sistema (Raspberry Pi).

4.2 Comunicazione tra i processi

La comunicazione tra il server RFCOMM e il processo di gestione dei GPIO avviene mediante una struttura dati condivisa, implementata come un dizionario. Il processo incaricato dell'esecuzione dei comandi effettua un controllo ciclico ogni 100ms per verificare la presenza di aggiornamenti; in caso positivo, applica le modifiche rilevate.

5 Implementazione software – Applicazione

L'applicazione mobile è stata sviluppata in linguaggio nativo (Kotlin) utilizzando l'ambiente di sviluppo Android Studio.

All'avvio, l'interfaccia mostra l'elenco dei dispositivi Bluetooth precedentemente associati. Una volta selezionato il Raspberry Pi, l'applicazione stabilisce una connessione con il server RFCOMM, consentendo l'interazione tramite la UI per l'invio dei comandi.

Tra le funzionalità implementate, è prevista la possibilità di impostare il colore delle luci anteriori tramite un selettore cromatico (color picker).

6 Replicazione

Per quanto riguarda la replicazione lato software, sono stati previsti due approcci distinti, in funzione del livello di competenza dell'utente: uno dedicato a utenti esperti e uno pensato per utenti meno esperti.

È opportuno sottolineare che, prima di procedere con l'implementazione software, è necessario completare la ricostruzione dell'hardware. A tal fine, si rimanda alla **Sezione 3**, che fornisce i dettagli relativi alla realizzazione fisica del dispositivo.

6.1 Utenti esperti

Gli utenti con maggiore esperienza possono replicare il progetto clonando il repository indicato nella **Sezione 7** e adattando gli script alle specifiche della propria installazione.

È necessario installare il servizio `macchinino.service`, disponibile nella directory `./rpi-code/systemd/`. Il file va copiato o spostato all'interno di `/etc/systemd/system/`:

```
sudo cp ./rpi-code/systemd/macchinino.service /etc/systemd/system/
```

Prima dell'attivazione, è necessario modificare i riferimenti al server all'interno del file `macchinino.service`, affinché corrispondano alla configurazione della propria macchina.

6.2 Utenti meno esperti

Gli artefatti necessari alla replicazione del progetto sono riportati nella **Sezione 7**. Per gli utenti che non hanno familiarità con l'utilizzo di `git`, è disponibile un'immagine preconfigurata del sistema, che consente di avviare il progetto senza necessità di configurazioni manuali.

Dopo aver scaricato l'immagine del sistema, denominata `rpi-image.img.gz`, inserire una scheda SD nel dispositivo in uso. Utilizzare quindi il comando `lsblk` per individuare il nome del dispositivo corrispondente alla scheda SD (ad esempio `/dev/sdX`).

Una volta identificato correttamente il disco, eseguire il seguente comando per decomprimere e scrivere l'immagine:

```
gzip -dc rpi-image.img.gz | sudo dd of=/dev/sdX bs=4M status=progress conv=fsync
```

Attenzione: sostituire `/dev/sdX` con il percorso corretto della propria scheda SD. L'uso errato di `dd` può comportare la perdita di dati su altri dispositivi di archiviazione.

Per facilitare l'avvio del sistema, è sufficiente creare una rete Wi-Fi utilizzando anche semplicemente uno smartphone, assegnandole il nome `macchinino` e impostando la password `macchinino1234`.

Il dispositivo si collegherà automaticamente a tale rete, dopodiché sarà possibile accedere al Raspberry Pi tramite un'applicazione come **JuiceSSH** (o un qualsiasi client SSH).

Per autenticarsi usare l'utente **macchinino** e password **macchinino1234**⁵.

Una volta stabilita la connessione, eseguire il seguente comando per avviare la modalità di accoppiamento Bluetooth:

```
sudo bash pair.sh
```

Completata la fase di pairing, sarà sufficiente installare l'applicazione mobile realizzata per iniziare a controllare il veicolo.

7 Artefatti e materiale utile

Il seguente materiale potrebbe essere utili:

- Repository Github ([link](#))
- Immagine del progetto ([link](#))
- Applicazione per il progetto ([link](#))

8 Problematiche del progetto

Le criticità riscontrate durante lo sviluppo del progetto sono principalmente riconducibili a una limitata esperienza pregressa nell'ambito della progettazione meccanica.

Una delle problematiche principali riguarda il peso complessivo del veicolo, che risulta eccessivo soprattutto quando viene montata la carrozzeria superiore. In tale configurazione, i motori installati non sono in grado di generare una coppia sufficiente per consentire le manovre di sterzata.

In assenza della carrozzeria superiore, invece, il veicolo è in grado di curvare, anche se con alcune manovre aggiuntive, risultando nel complesso funzionale.

Per ovviare a tale limitazione, si suggerisce di alleggerire ulteriormente la struttura, ad esempio utilizzando un pianale più leggero o realizzandolo interamente tramite stampa 3D, oppure adottando motori più potenti in grado di sopportare il carico attuale.

⁵È consigliato usare **passwd** per cambiare la password al primo accesso