

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA  
CAMPUS DI CESENA

---

Dipartimento di Informatica - Scienza e Ingegneria  
Corso di Laurea in Ingegneria e Scienze Informatiche

PROGETTAZIONE E SVILUPPO DI UN MODULO  
SENSORISTICO E DI COMUNICAZIONE PER LA  
PIATTAFORMA ROBOTICA ALPABOT

Elaborato in:

Sistemi Embedded e Internet-of-Things

Relatore:

Prof. Andrea Roli

Presentata da:

Paolo Baldini

Anno Accademico 2018/2019

# Indice

<b>Introduzione</b>	<b>3</b>
<b>1 Tecnologie proposte</b>	<b>7</b>
Comunicazione . . . . .	7
Sensoristica di percezione ambientale . . . . .	10
Sensoristica di posizionamento . . . . .	15
Risultati dello studio . . . . .	20
<b>2 Sviluppo di un modulo di sensoristica</b>	<b>22</b>
Motivi dello sviluppo . . . . .	22
Requisiti di progetto . . . . .	23
La fisica del progetto . . . . .	25
Componenti principali . . . . .	30
Schede e processi produttivi . . . . .	38
Implementazioni proposte . . . . .	42
Dettagli dell'implementazione hardware . . . . .	54
Portabilità verso altre piattaforme . . . . .	60
Risultati e possibili miglioramenti . . . . .	64
<b>3 Sviluppo di un applicativo di controllo</b>	<b>66</b>
Motivi dello sviluppo . . . . .	66
Requisiti di progetto . . . . .	66
Scelta della piattaforma di sviluppo . . . . .	67
Definizione dell'esperienza utente . . . . .	69
Comunicazione col dispositivo . . . . .	72
Risultati e possibili miglioramenti . . . . .	76
<b>4 Implementazione di una libreria software</b>	<b>77</b>
Motivi dello sviluppo . . . . .	77
Requisiti di progetto . . . . .	78
Complicazioni software affrontate . . . . .	81
Risultati e possibili miglioramenti . . . . .	85

**Conclusione**

**87**

**Appendice**

**88**

## Introduzione

La robotica è una disciplina ingegneristica trasversale che combina principalmente le branche di elettronica, fisica, informatica, matematica e meccanica. Questa sovrapposizione rende il suo studio e la sua progettazione materia complessa, dovendo il progettista avere a disposizione informazioni e conoscenze derivanti da tutte queste diverse discipline.

Dal punto di vista di uno studente di informatica ad esempio, la difficoltà si concentrerebbe sulla parte elettronica e meccanica, di cui nei casi più comuni ha carenze conoscitive. Per meccanici ed elettronici invece, la complessità risiederebbe nello sviluppo di applicativi performanti.

Alcuni di questi problemi sono effettivamente già stati affrontati e parzialmente risolti. Per sopperire alle carenze di sviluppo software, sono infatti presenti applicativi, perlopiù utilizzati nell'ambito dell'automazione industriale, che permettono una tipologia di programmazione visuale. Viceversa, per lo sviluppo rapido di progetti mecatronici, uno sviluppatore software può trovare in commercio molti "kit" preassemblati con moduli e componenti "plug and play" che gli permettano di concentrare i propri sforzi sulla parte a lui competente (i.e., software).

È proprio attorno a quest'ultimo concetto che ruota la realizzazione di questo elaborato.

Inizialmente infatti, a causa di un interesse relativo alla *swarm intelligence*, si è valutata la possibilità dello svolgimento di una tesi progettuale in quest'ambito. Si è però scoperto come questo campo di ricerca, all'interno dell'università, fosse affrontato solo tramite l'utilizzo di software di simulazione, e di come mancassero quindi piattaforme hardware atte al test pratico degli applicativi sviluppati.

Avendo già in precedenza riscontrato quanto un'esecuzione su di una piattaforma fisica divergesse nel comportamento rispetto ad un ambiente simulato, questa mancanza ci è sembrata una questione di notevole importanza.

Nella prova pratica infatti, l'essere messi di fronte a problematiche dipendenti anche da elettronica, fisica, meccanica e, non ultima, aleatorietà, permette di

comprendere come il presentarsi di alcune di queste, possa andare a influire anche sul software prodotto.

Come naturale susseguirsi di questo pensiero, abbiamo quindi cominciato a informarci sulle piattaforme per l'apprendimento attualmente in commercio, ed abbiamo notato come queste si trovino a costi spesso non facilmente accessibili per studenti o università. Non sono infatti stati trovati robot, che combinino utilizzi in diverse tipologie di applicazioni robotiche (e.g., swarm intelligence), ad un costo inferiore a qualche centinaio d'euro<sup>1</sup>.

Queste considerazioni han quindi spostato la nostra attenzione, dall'interesse iniziale relativo alla swarm intelligence, alla ricerca di possibilità di creazione di una piattaforma che possa permettere, da un lato di testare in maniera relativamente agile programmi in un ambiente hardware preventivamente sviluppato e collaudato, e dall'altro di avere a che fare con tutte le problematiche che in un test simulato possono essere messe in ombra.

L'obiettivo del progetto è quindi, in sintesi, l'implementazione di un sistema fisico per lo sviluppo e il test di applicazioni robotiche.

Questa piattaforma vuole nascere come **strumento di apprendimento**, e non si pone quindi l'obiettivo di assolvere un ruolo a livello professionale/lavorativo. Questa premessa iniziale è necessaria in quanto pone limiti al progetto, che avrebbe altrimenti potuto essere notevolmente diverso dall'attuale livello raggiunto.

Questa caratteristica, ci ha spinti alla ricerca di una soluzione che potesse combinare un **costo relativamente contenuto**, o quantomeno accessibile, a una quanto più possibile **variegata quantità di operazioni** testabili/effettuabili, sempre relativamente all'ambito robotico.

La volontà di molteplicità di operazioni, ha richiesto poi di valutare la **modularizzazione del sistema** per permettere un adeguamento ed una possibilità di sostituzione dei componenti hardware in base alle esigenze del progetto.

---

<sup>1</sup>È necessario specificare che questa ricerca non ha dato frutti solo in quanto si sono ricercate piattaforme che permettessero lo sviluppo di più tipologie di applicativi. Esistono infatti robot dedicati che è possibile trovare a costi nettamente inferiori (e.g., per swarm intelligence, il Kilobot[1] è costruibile al costo di qualche decina d'euro).

Avendo sempre come focus la didattica, è però necessario specificare come una piattaforma di questo tipo, seppur con l'intento di mettere in luce aspetti spesso trascurati nello sviluppo simulato, debba comunque essere adeguatamente supportata da una collaudata rete di **librerie**. Non è infatti pensabile iniziare un progetto di ricerca, ad esempio riguardante il mapping di un ambiente, dovendo studiare i tempi necessari perché un robot compia una rotazione o come poter effettuare un debug remoto.

A queste caratteristiche principali se ne aggiunge infine un'ultima, che vuole creare la vera differenza (oltre al costo) fra la nostra piattaforma e quelle attualmente disponibili in commercio.

Ci siamo accorti infatti di come uno dei vantaggi principali di una programmazione simulata rispetto ad una in hardware sia la visualizzazione dei così detti "log" in modo agile. Questa caratteristica, che normalmente mancherebbe in una piattaforma robotica, ci è sembrata fin troppo importante, dato l'ambito didattico, per poter essere ignorata. Si è per cui aggiunto come obiettivo la **creazione di un applicativo di controllo/visualizzazione remoto** per permettere un'agile interazione col sistema hardware.

In conclusione quindi, questa tesi tratterà dello studio e sviluppo di alcuni dei moduli facenti parte del progetto relativo alla piattaforma robotica **AlpaBot**. Questi, avranno dunque lo scopo di raggiungere gli obiettivi precedentemente esposti.

Come forse già suggerito da questo preambolo, i componenti che si andranno a sviluppare rappresenteranno solo una parte del sistema. Questo non è infatti frutto di un lavoro singolo, ma è stato ottenuto da un'opera congiunta fra il sottoscritto, Paolo Baldini, e il compagno di corso Alessandro Lombardi. È importante sottolineare comunque come lo sviluppo, nonostante abbia richiesto ovviamente grande collaborazione, soprattutto iniziale e finale, sia stato accuratamente suddiviso in campi di competenza, rendendolo di fatto facente parte di due tesi complementari.

Nello specifico, con finalità il raggiungimento degli obiettivi sopra descritti, gli argomenti che verranno trattati in questo documento saranno:

- sensoristica di posizionamento
- sensoristica per interazione con l'ambiente
- tecnologie di comunicazione
- creazione di un modulo per la piattaforma AlpaBot
- creazione di un applicativo di controllo remoto
- creazione di una libreria software
- portabilità verso altre piattaforme (accenni)

Come facilmente osservabile da una rapida ispezione dell'indice, i temi sopra elencati saranno trattati in diversi capitoli. Per agevolare la navigazione del documento, ci si riserva quindi di dare una breve spiegazione del contenuto di ognuno di questi.

Capitolo 1: In questo capitolo si tratterà delle tecnologie analizzate all'inizio della tesi. Qui se ne discuteranno punti di pregio e difetti e si valuterà quali scegliere.

Capitolo 2: In questo capitolo si tratterà del processo di studio, sviluppo e affinamento di un modulo di sensoristica infrarosso/ultrasonico. Si prenderanno inoltre in considerazione aspetti di elettronica e fisica, grazie ai quali daremo spiegazione delle scelte implementative effettuate.

Capitolo 3: In questo capitolo si tratterà dello sviluppo di un applicativo di controllo. Di questo, si considererà la definizione dell'interfaccia utente e si discuterà la strategia di comunicazione col dispositivo.

Capitolo 4: In quest'ultimo capitolo infine, verrà trattata l'implementazione di una libreria software atta a permettere un uso semplice del sistema AlpaBot. Di questa, si tratteranno le difficoltà affrontate e le soluzioni proposte.

Per ognuna delle implementazioni discusse nei vari capitoli, si discuteranno poi i possibili miglioramenti applicabili.

# 1 Tecnologie proposte

In questo capitolo verranno brevemente trattate alcune tecnologie di comunicazione, alcuni tipi di sensoristica di posizionamento e altri atti a permettere al robot di percepire l'ambiente circostante. Ne verranno valutate e rapportate: funzionalità, costi e, dove possibile, consumi.

## Comunicazione

In scenari normali, i robot necessitano di comunicare e possono farlo con molteplici fonti, ad esempio:

- altri robot: come avviene comunemente nella swarm intelligence;
- una base station: che può raccogliere semplicemente dati (e.g., in data analytics) o può gestire il comportamento di più robot da remoto;
- un controllore: come un joystick che permetta un utilizzo diretto da remoto (e.g., pensiamo al controllo di droni);
- una persona: che comunica col robot tramite luci (e.g., un LED rosso può comunicare un pericolo), pulsanti (e.g., pulsante di inizio di un'operazione) e altri dispositivi spesso montati sul robot stesso.

Varie tecnologie sono state sviluppate negli anni per permettere ciò, e molte di queste sono basate sulle Radio Frequencies (RF). In questo paragrafo andremo ad analizzarne alcune e vedremo come queste siano state utilizzate in scenari di ricerca.

Nel corso della trattazione potremo, anche notare come la maggior parte delle tecnologie non mettano simultaneamente a disposizione buoni range di funzionamento, consumi e velocità di comunicazione, e come sia perciò necessario effettuare la scelta migliore in base alla tipologia di applicazione. Quest'ultima necessità, richiama nuovamente la questione dell'importanza della modularizzazione.



Le tecnologie che prenderemo in analisi sono:

- Bluetooth
- Infrared Radiation (IR)
- NFC
- Radio Frequency (RF)
- WiFi
- Zigbee

**Bluetooth** è uno standard WPAN<sup>2</sup> basato sulle RF. Ha un basso costo, un range da circa un metro fino a un centinaio di metri (a seconda della classe di potenza), una buona velocità di trasmissione e un consumo, proporzionale alla potenza in uscita, che può essere anche particolarmente basso.

La tecnologia Bluetooth si adatta quindi, a causa del range ridotto, ad applicazioni robotiche operanti in contesti indoor.

**Infrared Radiation (IR)** è la radiazione elettromagnetica con lunghezza d'onda compresa fra quella delle microonde e quella della luce visibile. Richiede poca energia per funzionare e può essere usata sia come tecnologia di comunicazione che come sensore di prossimità. I contro di questa tecnologia sono un range di funzionamento ridotto (decine di centimetri) e una sensibilità ai disturbi rilevante. Oltre a questo, la presenza di ostacoli rende del tutto impossibile la comunicazione, rendendo quindi necessaria una line-of-sight libera e diretta.

Nonostante i lati negativi, è una tecnologia molto utilizzata per il robot swarm. Infatti, il consumo ridotto, la compattezza e la multifunzionalità la rendono appetibile per i micro-robot<sup>3</sup>.

---

<sup>2</sup>Wireless Personal Area Network

<sup>3</sup>I robot Alice Microbot [2] e il Jasmine [3] utilizzano questa tecnologia

**Near Field Communication (NFC)** è una tecnologia basata su RF bidirezionale e la succeditrice di RFID. Lavora a corto raggio (fino a un massimo di 10 cm) ed ha una buona velocità.

Non è una tecnologia *mainstream* nella comunicazione fra robot, ma ne è stato valutato il possibile utilizzo durante il TAROS del 2013<sup>4</sup>. Per maggiori approfondimenti si rimanda alla reference relativa [4].

**Radio frequency (RF)** è alla base di gran parte delle tecnologie wireless. Può variare molto in termini di velocità, range e consumi a seconda della frequenza di lavoro e può quindi essere adattabile a varie esigenze.

**WiFi** è una tecnologia sviluppata per lo standard WLAN<sup>5</sup>. Ha consumi rilevanti, un range che varia dalle centinaia di metri a qualche chilometro (in base alla configurazione) e supporta un'elevata velocità di comunicazione.

A causa dei consumi, dell'incapacità di certi microcontrollori di gestirne il data rate e della possibilità crescente di disturbi al crescere dei dispositivi collegati<sup>6</sup>, il suo utilizzo è limitato a gruppi di robot dalle capacità computazionali relativamente elevate e limitati per numero di individui.

**Zigbee** è basato sullo standard WPAN. Ha consumi ridotti, basso data rate e range molto elevato (oltre 1km con Zigbee Pro). Nonostante le specifiche siano molto promettenti, non sono state trovate ricerche correlate nell'utilizzo come tecnologia di comunicazione fra robot.

## **Analisi finale**

Avendo quindi analizzato sommariamente le varie tecnologie di comunicazione,

---

<sup>4</sup>Towards Autonomous Robotic Systems: una conferenza internazionale tenuta ad Oxford, riguardante i sistemi robotici autonomi

<sup>5</sup>Wireless Local Area Network

<sup>6</sup>Un numero superiore alle centinaia (come può avvenire negli swarm) può portare a un tasso d'interferenza molto elevato

si riporta successivamente una tabella riassuntiva delle varie caratteristiche in rapporto alle altre<sup>7</sup>.

	battery consumption	communication range	transfer speed
Bluetooth	Low	Tens of meters	Medium
IR	Very Low	Tens of centimeters	Low
NFC	Low	About ten centimeters	Low
WiFi	Medium	Tens of meters	High
Zigbee	Very low	Tens of meters - A kilometer	Low

## Sensoristica di percezione ambientale

Se la comunicazione rappresenta un aspetto importante per un robot, nella maggior parte dei casi la percezione dell'ambiente risulta anche più importante. Tralasciando infatti i robot teleguidati, tutti i rimanenti necessitano di modalità per capire cosa stia accadendo nello spazio attorno a loro.

Dato lo scopo didattico del progetto, si è ritenuto che una conoscenza dell'ambiente circostante, e quindi una capacità del robot stesso di *autogestirsi* senza dipendere necessariamente da comandi esterni, fosse una caratteristica fondamentale per la piattaforma che si sarebbe andata a progettare. Sono state per questo motivo analizzate alcune possibili tecnologie per permettere al robot di ottenere una conoscenza, seppur limitata, dell'ambiente.

Comunemente utilizzate o proposte in letteratura, le tecnologie trovate per l'acquisizione di informazioni di questo tipo sono state le più disparate, ma essendo molte limitate a scenari specifici o particolari, si è deciso di inserire a seguito una descrizione delle sole tecnologie considerate maggiormente rilevanti o possibilmente utili al progetto. Nello specifico tratteremo di:

- foto riconoscimento

<sup>7</sup>A causa delle sue molteplici varianti, la radio frequency non è stata inserita in tabella

- sensoristica di prossimità:
  - ad ultrasuoni
  - ad effetto ottico

**Il foto riconoscimento** è una tecnica di identificazione ambientale molto potente ma allo stesso tempo molto onerosa a livello di risorse computazionali per eseguire il calcolo. Si basa infatti sull'acquisizione di immagini e sulla loro analisi per estrarne informazioni (Figura 1). Queste possono essere ottenute con vari metodi, quali *pattern matching*, *analisi dei colori*, etc.



Figura 1: In questa figura è possibile vedere l'immagine acquisita dalla fotocamera omnidirezionale di un Foot-Bot, uno degli esemplari di robot appartenenti al progetto Swarmanoid[5]. È interessante notare la strategia di acquisizione ideata dal team. La fotocamera omnidirezionale si basa infatti sull'utilizzo di un cono specchiato orientato verso la lente della stessa, che permette quindi di avere una visione a 360° dell'ambiente circostante.

Dato l'oneroso impiego di memoria e di tempo calcolo però, l'utilizzo di questa tecnica è limitato a dispositivi più potenti di un microcontrollore, il quale non possiede spesso neanche la memoria sufficiente per memorizzare un'immagine su cui lavorare.

Queste caratteristiche stringenti di potenza contrastano, anche se non direttamente, col principio di economicità, avendo un controllore più potente un costo notevolmente maggiore rispetto ad un microcontrollore<sup>8</sup>.

**La sensoristica di prossimità** è una famiglia di trasduttori che sfruttano, per la conoscenza dell'ambiente limitrofo, il calcolo della distanza dagli oggetti vicini. È una tecnica che, a seconda delle tecnologie utilizzate, può essere anche particolarmente economica e/o precisa.

Nel corso dello studio, sono state trovate in questa famiglia due tipologie di sensori che hanno effettivamente la capacità per conquistare un ruolo fondamentale nel progetto:

- sensori di prossimità a ultrasuoni
- sensori di prossimità ottici

Entrambe queste tecnologie utilizzano il fenomeno fisico della riflessione (Figura 2).

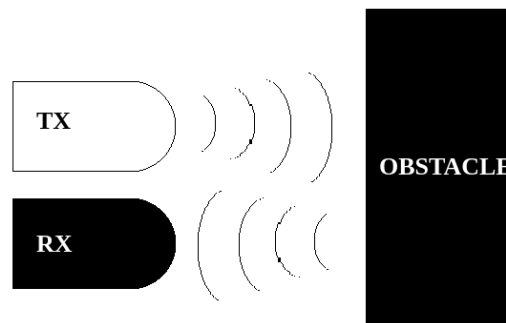


Figura 2: Il principio di funzionamento delle due tecnologie di sensori di prossimità presi in considerazione: emissione del segnale, riflessione e acquisizione/rilevamento del segnale riflesso.

**I sensori ad ultrasuoni** basano il loro funzionamento sul tempo intercorso fra l'emissione di un segnale sonoro e, una volta riflesso, la sua ricezione.

<sup>8</sup>Si parla di decine se non centinaia di euro contro poche decine di centesimi

Questo è possibile grazie alla velocità raggiunta oggi dai microcontrollori, che permette a questi di riuscire a misurare, in maniera abbastanza precisa, il tempo di viaggio di un'onda.

Dall'ottenimento di questo, si può poi ricavare la distanza percorsa dall'onda stessa conoscendone la velocità<sup>9</sup>.

Lo svantaggio principale di questi dispositivi si presenta nel caso di una superficie inclinata o particolarmente piccola. In questa occorrenza l'onda potrebbe non essere riflessa o esserlo ad un angolo non sufficiente alla sua rilevazione.

Considerato il costo relativamente economico di questi dispositivi e la loro affidabilità, il loro utilizzo è stato preso in seria considerazione per il progetto.

**I sensori ottici** hanno principio di funzionamento, come quelli ad ultrasuoni, basato sulla riflessione. La differenza sostanziale è che utilizzando questi un'onda elettromagnetica (la quale ha una velocità di propagazione molto maggiore rispetto ad un'onda sonora), non è possibile calcolare precisamente la distanza percorsa tramite la misura del tempo di viaggio.

I metodi esistenti per sopperire a questo problema sono principalmente tre:

- Il primo basa il suo funzionamento sul decadimento della potenza del fascio di luce nel percorrere una determinata distanza. Percorrendo un tragitto infatti, l'onda si disperde, perdendo man mano "potenza". Rilevando l'intensità del segnale rientrante nel dispositivo, è quindi possibile determinare la distanza percorsa.
- Il secondo basa invece il suo funzionamento sulla misura di una frequenza il cui periodo è determinato, e direttamente proporzionale, al tempo di andata-ritorno della radiazione. Nella pratica, questa tecnica consiste nel continuare ad inviare onde elettromagnetiche in uscita, fin quando non si riceverà in ingresso l'onda riflessa. A quel punto il segnale in uscita verrà spento, per poter poi essere riattivato una volta che il segnale in ingresso

---

<sup>9</sup>Un'onda ha ovviamente velocità differenti a seconda del mezzo trasmissivo attraversato, e.g., acqua, aria, etc.

sarà svanito. Ripetendo questa procedura ciclicamente, si otterrà quindi un'onda la cui frequenza sarà data dalla formula:  $f = \frac{3 * 10^{11}}{2d} + \frac{1}{T_e}$  dove:

- $3 * 10^{11}$  è la velocità della luce in *mm/s*
- $d$  è la distanza in mm
- $T_e$  è il ritardo costante di commutazione introdotto dal circuito

I circuiti utilizzati per questa tecnica sono economici e la distanza è ricavabile con un semplice calcolo e con buona precisione.

La caratteristica limitante di questa è che richiede l'emissione della radiazione con un raggio molto "stretto"<sup>10</sup>, in modo da ridurre al minimo la potenza dispersa. Inoltre, è richiesta una precisa conoscenza del tempo di ritardo del circuito, cosa non facilmente ottenibile senza tecnologie quali oscilloscopi.

- Il terzo si applica invece a una versione di sensoristica di prossimità meno avanzata delle precedenti. Questo viene infatti utilizzato, come nel caso di molti dispositivi commerciali, dove si vuole solo rilevare la presenza o meno di un oggetto in un range. Questa rilevazione è effettuata valutando solo che il segnale riflesso in ingresso abbia una intensità superiore a una determinata soglia.

In questo caso, per cercare di eliminare o ridurre l'interferenza da fonti esterne, alcuni di questi dispositivi adottano la tecnica della modulazione del segnale in uscita (sostanzialmente inviando un onda quadra). Una radiazione in ingresso verrà quindi considerata "presenza" (i.e., riflessa) solo nel caso che questa sia modulata allo stesso modo di quella precedentemente inviata in uscita.

Delle tre possibili soluzioni sopra analizzate, **la seconda**, per quanto migliore in termini di affidabilità e precisione, richiede però componenti hardware scelti accuratamente e dispositivi di analisi del circuito complessi, come ad esempio gli oscilloscopi, che vengono venduti a prezzi non facilmente accessibili.

---

<sup>10</sup>È infatti una tecnica utilizzata principalmente in combinazione con fasci laser, che possono essere focalizzati su aree molto piccole

Considerando quindi il carattere didattico della piattaforma, che vuole comunque mantenere la possibilità di una auto-produzione relativamente facile, questa soluzione risulterebbe in una complicazione importante alla costruzione e messa in funzione del dispositivo.

Si è quindi ritenuto che, nonostante i grandi vantaggi, la seconda proposta illustrata non fosse adatta agli obiettivi del progetto.

Allo stesso modo, la mancanza nella **terza** opzione di una vera e propria misura di distanza la rende non adatta agli obiettivi della piattaforma.

Analizzando infine **la prima** soluzione, questa presenta risvolti interessanti. Il circuito per la ricezione del segnale in ingresso non è necessariamente complesso, necessitando infatti, a seconda del sensore utilizzato, addirittura di una sola resistenza o potenziometro<sup>11</sup>. In aggiunta, il costo risulta addirittura inferiore alla seconda soluzione.

Rispetto a quest'ultima però, un possibile lato negativo potrebbe essere riscontrato nel calo della precisione e in una maggiore sensibilità ai disturbi.

Ad ogni modo, nonostante i possibili svantaggi elencati, questa tecnologia è stata valutata una valida candidata alla realizzazione del progetto.

## **Sensoristica di posizionamento**

Si è fin ora trattato di sensoristica atta alla percezione dell'ambiente, ma anche un'altra tipologia ha catturato la mia attenzione durante lo studio di possibili tecnologie che potessero tornare utili allo sviluppo: quella di posizionamento relativo nello spazio.

Per chiarire meglio cosa si intende approfondire in questa sezione: tratteremo di alcune quelle tecnologie atte a conoscere la posizione di un robot in un ambiente, ovvero rispetto a determinati punti di riferimento.

Una funzionalità di questo tipo, se effettivamente realizzabile, permetterebbe lo sviluppo di applicazioni di robotica avanzate, come ad esempio relative ai

---

<sup>11</sup>In dispositivi quali i fotodiodi ad esempio, una resistenza è sufficiente ad ottenere un valore, più o meno preciso, in tensione



veicoli a guida autonoma<sup>12</sup>. È stato dunque considerato un valido impiego di tempo lo studio di funzionalità che potrebbero permettere testing di livello così avanzato.

Nello specifico, le tecnologie che fin dall'inizio sono risultate più interessanti sono state:

- trasduttori di posizione
- trilaterazione
- tag di posizione
- navigazione satellitare

**I trasduttori di posizione** sono sensori in grado di trasdurre movimento angolare o lineare in valori numerici (angoli e distanze). Esistono di vario tipo, fra cui: capacitivi/induttivi, magnetici, potenziometrici ed ottici. Questi dispositivi non forniscono quindi di per sé una posizione spaziale ma, dato un punto di partenza, permettono di ottenerla tramite "somme di movimenti".

Il difetto principale nell'uso di questi dispositivi per conoscere una posizione, deriva dalla non perfezione nel calcolo delle distanze. Facendo un esempio, un encoder lineare (i.e., un trasduttore di posizione) con errore di 1mm aggiuntivo ogni 10m, una volta percorsi 100m avrà un errore di 1cm, percorso 1km avrà un errore di 10cm, etc. Come possiamo notare, senza mai resettare l'errore, prima o poi questo, continuando a sommarsi, diventerà tanto grande da sfalsare il funzionamento del sistema (una possibile soluzione al reset dell'errore è proposta dalla reference [6]). Ad onor del vero, questa situazione si verificherà verosimilmente solo dopo grandi distanze percorse, rendendo di fatto questo errore probabilmente abbastanza piccolo e trascurabile per la maggior parte delle tipologie di applicazioni per cui è pensata la piattaforma. È infatti importante considerare come molti encoder di costo contenuto raggiungano, senza troppi problemi, precisioni vicine al millimetro<sup>13</sup>.

---

<sup>12</sup>Questi potrebbero ad esempio calcolare un tragitto e tentare di raggiungerlo tramite queste capacità di localizzazione

<sup>13</sup>Encoder più costosi riescono a raggiungere precisioni fino al micrometro

Considerando la possibile utilità, la precisione e un costo che può rivelarsi anche relativamente economico per alcuni modelli, questa tecnologia è quindi sicuramente valida e degna di nota. D'altro canto, la loro caratteristica di avere un processo di ottenimento della posizione macchinoso e sensibile ad errori e problemi esterni al dispositivo (e.g., si può pensare a una ruota che scivola su di un terreno scivoloso, facendo registrare falsi movimenti), ci ha fatto desistere dal renderli parte della piattaforma.

**La trilaterazione** è una tecnica di rilevamento della posizione che sfrutta l'interpolazione delle distanze di un soggetto da (almeno tre) punti di posizione conosciuta, con l'obiettivo di ottenere la posizione a loro relativa. Nello specifico, sfrutta la potenza del segnale proveniente dai tre "fari" (i.e., i punti conosciuti) per calcolare, per ognuno, la distanza che ne intercorre. Una volta ottenute le singole distanze, queste devono venire interpolate per permettere di individuare un singolo punto (Figura 3).

Essendo questa tecnica basata solitamente sull'emissione e l'analisi della potenza delle radiofrequenze, ci si è presto resi conto di come questa non sia in grado di fornire una precisione che vada oltre il metro. Se per dispositivi lavoranti su scale maggiori questo può non rappresentare un problema, per l'ambito didattico (e quindi per distanze di lavoro minori) su cui si basa questo progetto, questa limitazione rende totalmente inutile l'utilizzo di questa tecnologia.

Durante lo studio di questa tecnica, ne è stata proposta **una variante** (che attualmente non ha trovato riscontri in letteratura), che è stata velocemente abbandonata ma che trovo possa risultare interessante. Questa si basa sull'idea di una trilaterazione su scala ridotta, che fonda il proprio funzionamento sulle radiazioni luminose generate dai tre punti noti (per certi versi, dei veri e propri fari). Un ricevitore sul robot, dovrebbe dunque ruotare per trovare, per ogni faro, il punto di luminosità maggiore (i.e., non la corona luminosa di questo, ma proprio il suo centro). Una volta trovato, dovrebbe quindi calcolarne la distanza tramite la potenza della radiazione elettromagnetica, trovando così i valori necessari alla trilaterazione.

Come si può immaginare dalla semplice lettura della proposta, questa soluzione

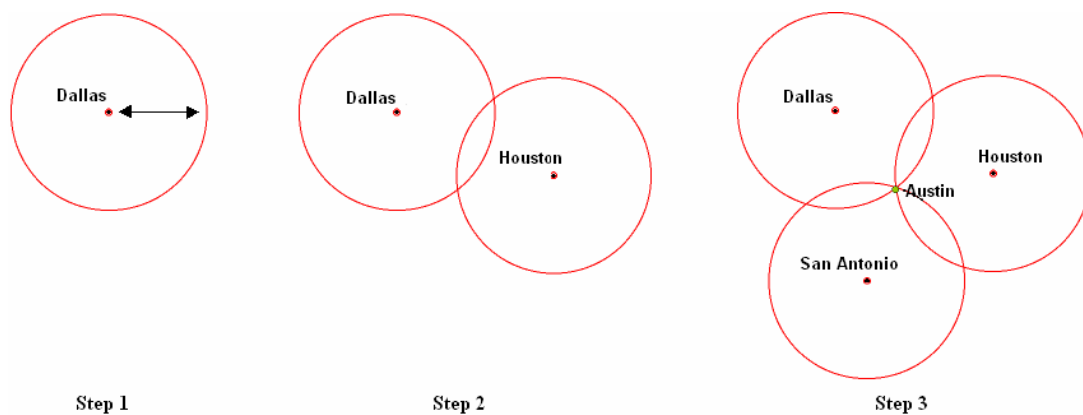


Figura 3: Esempio di trilaterazione. Step 1: conosciamo la distanza (tramite la potenza del segnale) del soggetto triangolato da Dallas, ma non ne conosciamo la "direzione". Step 2: riduciamo i punti candidati a due tramite la combinazione della distanza da Dallas e da Houston. Step 3: otteniamo la posizione del soggetto (in questo caso la città di Austin) dall'interpolazione delle tre distanze.

sarebbe risultata molto, se non troppo complicata per vari motivi, primo fra tutti il dover fornire al sensore della radiazione la possibilità di muoversi di  $360^\circ$  nel piano orizzontale e di  $90^\circ$  nel piano verticale (distinguere il centro del faro dalla corona è un'operazione che dev'essere effettuata su entrambi gli assi). Il sensore di luminosità dovrebbe poi essere particolarmente preciso e con angolo di visione molto stretto per poter trovare il centro del faro. Non sapendo se questa proposta possa effettivamente funzionare, e non conoscendone poi la possibile precisione, si è valutato che lo sforzo nello studio di questa non sarebbe stato probabilmente ripagato dall'eventuale funzionamento della stessa. È stata per cui messa da parte.

**I tag di posizione** <sup>14</sup> (la cui idea è stata tratta da una pubblicazione relativa al reset dell'errore di posizione di uno swarm [6]) consistono in, appunto, tag

<sup>14</sup>È stato recentemente scoperto, durante la ricerca di immagini rappresentative, di come l'idea dei tag da me sviluppata fosse molto simile ad una già esistente pubblicazione. A seguito di ciò, si decide di lasciare la precedente descrizione da me fatta, allegando però una opportuna reference alla pubblicazione preesistente [7], per chi volesse approfondirne la trattazione

(e.g., RFID) appositamente dislocati nell'ambiente. Ognuno di questi tag va supposto contenente l'indicazione della posizione a cui si trova (Figura 4).

Una fitta rete di tag di questo tipo potrebbe andare a formare una maglia di posizioni note sulla quale i robot possano muoversi, mantenendo quindi una conoscenza della loro posizione spaziale particolarmente precisa.

Una maglia con nodi più distanti potrebbe invece, ad esempio, utilizzare l'**RSSI** (i.e., received signal strength indication; la potenza del segnale) per applicare una versione ridotta di trilaterazione che, combinata con la posizione esplicitata dai tag stessi, *potrebbe* creare buoni risultati.

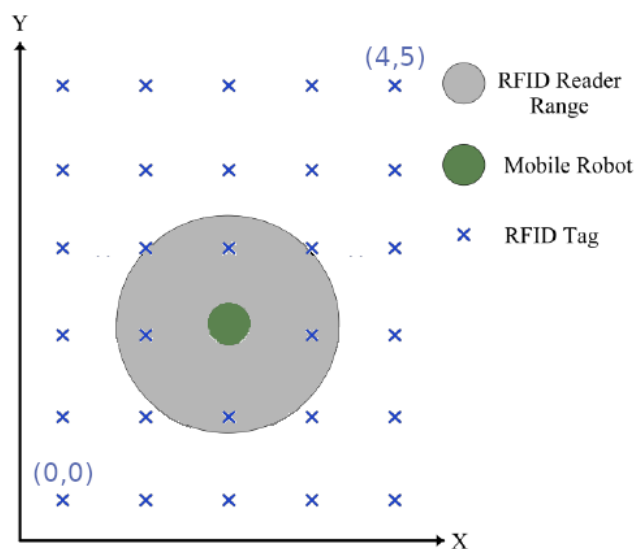


Figura 4: Esempio di maglia di tag RFID.

Nonostante l'idea di una soluzione di questo genere possa essere interessante, la necessità di dislocare un gran numero di tag e quella di fornire il robot di un lettore, ci hanno fatto propendere per abbandonare l'idea.

**La navigazione satellitare**, spesso associata al solo GPS, è in realtà una famiglia di tecnologie atte alla geo-radiolocalizzazione, ovvero alla rilevazione della posizione sulla terra.

Molte di queste sono ancora in via di sviluppo nel mondo e solo alcune sono già disponibili all'utilizzo. Queste sono:

- GPS (Stati Uniti)
- GLONASS (Russia)
- Galileo (Europa)

La differenza fra queste, limitatamente agli scopi del progetto, risiede nella precisione raggiungibile. Il GLONASS, permette di raggiungere precisioni fino a ~3m. Il GPS raggiunge, in campo civile, qualche metro di precisione<sup>15</sup>. Galileo infine, permetterà, una volta totalmente in servizio, di raggiungere precisioni di centimetri[8].

L'accuratezza di questa tecnica potrebbe, nel caso di utilizzo di Galileo<sup>16</sup>, essere effettivamente sufficiente agli scopi del progetto. Il lato negativo dell'utilizzo di questa è quindi solo relativo al costo, non particolarmente economico.

## **Risultati dello studio**

In questo primo capitolo della tesi, sono state prese in esame diverse tecnologie e di queste si è cercato di discernere pregi e difetti. Questi erano in alcuni casi oggettivi, mentre in altri risultavano tali solo prendendo in considerazione gli obiettivi e le caratteristiche richieste alla piattaforma.

Questo sopralluogo iniziale è quindi servito per addentrarsi un poco più a fondo nella comprensione di ciò che sia possibile realizzare e di ciò che possa rappresentare un valido *trade of* fra costi, funzionalità, semplicità di utilizzo e possibilità tecnologiche. Durante questo studio sono inoltre emerse diverse proposte e idee che sarebbe stato interessante affrontare, ma che avrebbero richiesto probabilmente la stesura di una tesi a parte. A causa dei limiti temporali infatti, è stato necessario effettuare una scelta fra solo alcune delle tecnologie che si sarebbe voluto inserire nel progetto. Questa selezione è stata effettuata

---

<sup>15</sup>L'uso militare del servizio permette di usufruire di un secondo canale di comunicazione, permettendo il raggiungimento di precisioni di centimetri

<sup>16</sup>È infatti l'unica tecnologia civile a raggiungere precisioni di centimetri

principalmente in base alle caratteristiche richieste dalla piattaforma, ma si è anche tenuto conto di interessi personali.

Entrando però maggiormente nello specifico, l'ambito delle *tecnologie di posizionamento* non è stato inserito nel progetto. Per quanto riguarda trilaterazione e tag, la causa è attribuibile alle forti limitazioni di precisione e alla necessità di luoghi di testing costruiti *ad hoc*. I trasduttori di posizione, per quanto validi, sono stati messi da parte in quanto si è preferito dedicare il tempo a disposizione nello studio ed utilizzo di altre tecnologie. La navigazione satellitare infine, è stata esclusa a causa del costo, non proibitivo ma sicuramente notevole.

Se per le tecnologie di cui sopra si è preferito non investire tempo, per quanto riguarda invece quelle di comunicazione è stato fatto uno sforzo maggiore. Negli ultimi anni sono sempre più spesso presenti in commercio schede di prototipazione già contenenti moduli di comunicazione integrati<sup>17</sup>. Queste schede hanno spesso approcci alla comunicazione simili, comprendenti processi di pairing e, talvolta, connessione tramite socket. Data questa premessa e l'interessante possibilità di permettere la personalizzazione della piattaforma, è stato ritenuto un valido impiego di tempo lo sviluppo e l'utilizzo di due di queste tecnologie, le più versatili anche nell'interazione con altri dispositivi: **Bluetooth e WiFi**.

Se le due tecnologie scelte possono risultare valide per la maggior parte delle comunicazioni, alcune di queste possono però non essere adeguatamente supportate. Applicazioni di swarm robotic ad esempio, non richiedono potenti tecnologie di comunicazione, ma ne preferiscono spesso altri tipi capaci di una comunicazione soltanto locale. A questo scopo, e in combinazione al fatto che questa tecnologia permette anche l'utilizzo come *sensore di percezione ambientale*, è stato deciso di inserire nella piattaforma un **modulo utilizzante le radiazioni infrarosso**, sia come sensore di prossimità che come mezzo comunicativo.

---

<sup>17</sup>Vedasi ad esempio le schede NodeMCU o Raspberry per l'integrazione con moduli WiFi, e schede come la Bluno Beetle per l'integrazione con moduli Bluetooth

## 2 Sviluppo di un modulo di sensoristica

Questo capitolo della tesi ha l'intenzione di spiegare, più approfonditamente rispetto al precedente, i principi fisici di base, le tecnologie e i componenti utilizzati e i vari passi e motivazioni che hanno portato allo sviluppo della versione finale del modulo della piattaforma atto alla sensoristica e comunicazione infrarossa.

### Motivi dello sviluppo

Come precedentemente accennato, la scelta dello sviluppo di un modulo di questo genere parte dalla sua caratteristica di dualità fra **possibilità comunicative e sensoristiche** e il suo relativamente **economico costo**. La tecnologia infrarosso presenta infatti la capacità di agire come un sensore di prossimità in grado non solo di valutare la presenza o assenza di un oggetto, ma se affiancato da opportuna componentistica, anche di valutarne la distanza. Dualmente, grazie alla natura analogica delle onde, quantizzabile quindi in valori digitali, è anche possibile trarne una componente comunicativa. Sono ad esempio presenti nelle maggior parti delle abitazioni dispositivi che sfruttano quest'ultima proprietà: i televisori.

Ad onor del vero, questa dualità non è spesso mantenuta in dispositivi comuni. È altresì necessario, per mantenerla, dotarsi di un accuratamente progettato insieme di componentistica a supporto del singolo sensore. Non è infatti normalmente possibile, senza un circuito di condizionamento accuratamente progettato, sfruttare sia la componente analogica che quella digitale del segnale infrarosso.

Per questo motivo, necessitando in questo progetto di entrambe le componenti, è stato necessario lo studio e il successivo sviluppo di un modulo che permettesse questo bipolarismo.

## Requisiti di progetto

Fra i requisiti progettuali fondamentali del modulo rientra ovviamente, in primo luogo, la sua capacità di poter **misurare distanze** da oggetti limitrofi al robot. È infatti questo il motivo principale a cui si è dovuta la sua scelta nello sviluppo della piattaforma.

La distanza minima percepibile che ci si è posti come obiettivo è stata di una trentina di centimetri.

Come secondo ma altrettanto fondamentale punto si è presentata ovviamente la **capacità comunicativa**. Questa possibilità ha fatto invero da discriminante nella scelta preferenziale di questa tecnologia rispetto ad altre, ed è quindi uno dei punti necessari per poter considerare il progetto effettivamente valido.

È necessario specificare però che, per gli ambiti applicativi accennati nel capitolo precedente, per questa tecnologia è sufficiente un corto range d'azione. L'obiettivo è stato quindi rendere possibile una comunicazione ad una distanza di almeno 15-20cm dal bordo esterno del robot.

Oltre a queste due caratteristiche fondanti, lo sviluppo di questo modulo ha incontrato fin da subito alcune altre scelte progettuali di non poco conto.

Questo parte infatti con l'intento di essere sì una componente di base della piattaforma, ma di dare anche la possibilità di essere sostituito con moduli di maggior potenza o miglior fattura. A questo scopo, si è fin da subito mostrata la necessità di far in modo che questo fosse *facilmente* sostituibile, aggiungendo quindi paletti alla disposizione, forma, dimensione e peso per renderlo un vero e proprio **modulo intercambiabile** (Figura 5).

Un altro aspetto notevole nella progettazione del modulo è stato l'intento di permettere a questo di utilizzare il **minor numero di pin**, cercando di mantenere al contempo quante più caratteristiche di utilità possibili.

In una scheda di prototipazione (come ad esempio Arduino) infatti, non è spesso presente una particolare abbondanza di connettori. Questa caratteristica, ha reso molto importante il requisito di utilizzarne il minor numero, in modo da



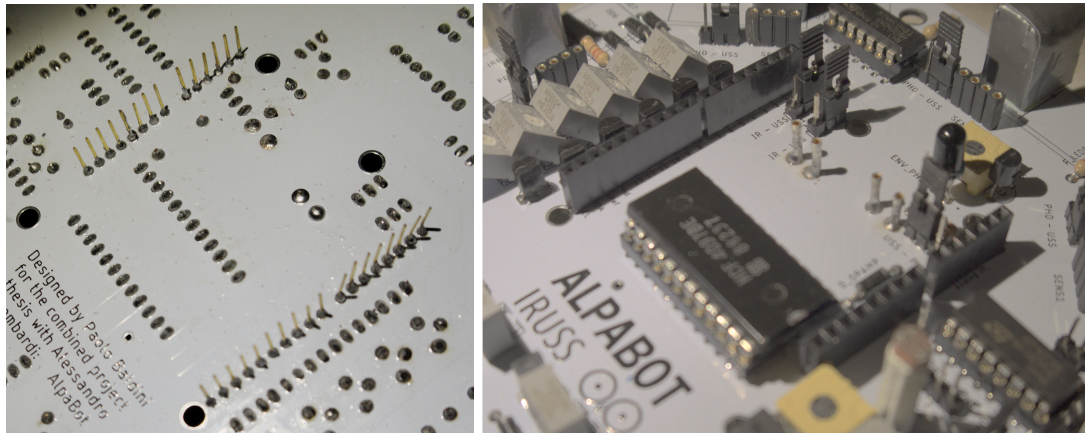


Figura 5: È possibile notare come il modulo disponga di pratici pin che ne permettono la facile sostituibilità.

permettere all'utente della piattaforma di aggiungere personali moduli o componenti alla stessa (Figura 6).

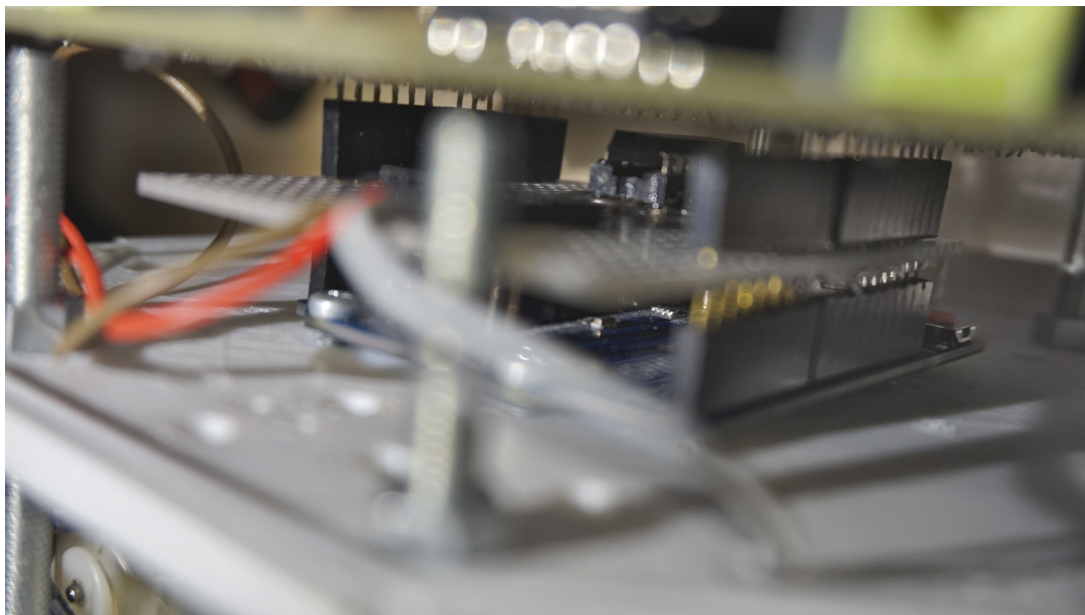


Figura 6: Sono in questa figura visibili due moduli collegati. L'aggiunta di altri non risulterebbe possibile se non si prestasse attenzione all'utilizzo del numero di pin.

Infine, come ultimo punto, se ne presenta uno relativo alla **forma e dimensione**

del modulo e alla **disposizione dei componenti**. Se infatti questo vuol essere utilizzabile anche da schede differenti, ciò non toglie che sia progettato e creato per una configurazione di default.

Nel corso di una discussione riguardo la standardizzazione di forma e dimensione del robot, si è deciso di fare altrettanto col modulo. Quest'ultimo, è stato deciso dover essere un ottagono con, al centro, i pin di connessione per una scheda Arduino Uno.

Il requisito degli otto lati è stato dettato dalla volontà di rendere più efficiente la comunicazione infrarosso. Utilizzando infatti questa una ricezione locale a  $360^\circ$ , era necessario che fosse presente **un ricevitore (e un emettitore) su ogni lato** e che questo avesse un angolo di visione sufficientemente ampio da captare tutti i segnali in ingresso. Considerando che la maggior parte dei ricevitori IR di cui si sono potute trovare le specifiche avevano un angolo di ricezione pari a  $60^\circ$ , sei lati sarebbero in teoria dovuti risultare sufficienti alla copertura. In realtà, questo non risulta vero a causa del fatto che l'area di visione dei ricettori non è effettivamente un cono, ma (circa) un'ellisse<sup>18</sup>. A causa di ciò, sei ricevitori non avrebbero effettivamente coperto i  $360^\circ$  richiesti. Per permettere la copertura è stato quindi necessario aumentare il numero di lati da sei a otto (Figura 7).

Per quanto riguarda invece i pin di connessione, questi è stato deciso debbano essere posizionati al centro del modulo per rendere lo stesso più stabile, se utilizzato senza supporti.

Infine, come ultimo punto relativo alla forma, si è scelta la posizione di due fori atti all'ancoraggio della scheda sul robot.

## La fisica del progetto

Come fin ora discusso, è stata scelta per l'implementazione del modulo la tecnologia basata sulle radiazioni infrarosso. Per capire meglio come e con cosa questa operi, è però necessario fare un piccolo passo all'interno del mondo della fisica.

---

<sup>18</sup>Il campo visivo si riduce infatti all'aumentare della distanza

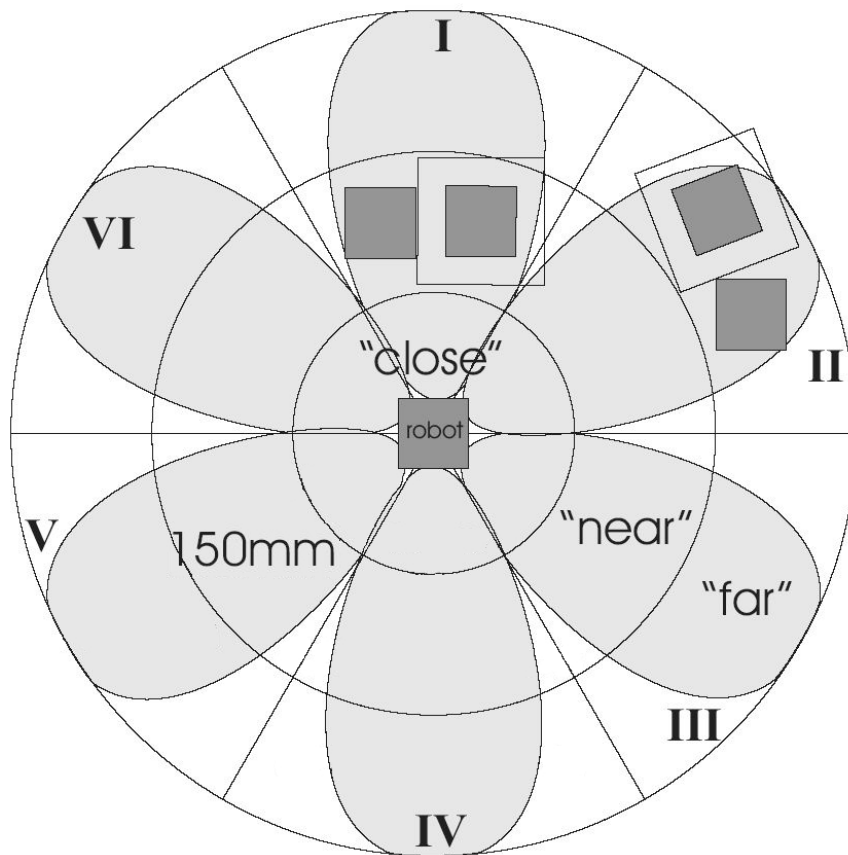


Figura 7: Raggio di "percezione" di un robot con sei lati ed altrettante coppie emettitore/ricevitore IR. Come visibile, i ricevitori in figura, a 15cm di distanza, hanno meno di  $60^\circ$  di visione, non permettendo quindi di coprire adeguatamente i  $360^\circ$  richiesti nella la media distanza. L'aumento ad otto lati risolve o quanto meno mitiga il problema rendendo la comunicazione agile *almeno* fino ai 15cm.

### La radiazione infrarosso

Facenti parte, insieme alla luce visibile, delle radiazioni elettromagnetiche, queste rientrano nello spettro di radiazioni non visibili all'occhio umano. Insieme alle radiazioni UV e alla luce visibile, sono spesso chiamate *radiazioni termiche* per la loro caratteristica di trasmettere calore per irraggiamento. Questo, diventa sempre maggiore all'aumentare dell'intensità della radiazione,

arrivando addirittura a poter causare ustioni<sup>19</sup>.

Questa proprietà di trasmissione dell'energia, è utilizzata in molte tecnologie. È risaputo infatti come gli oggetti generino normalmente radiazioni termiche a causa dell'agitazione molecolare che ne innalza la temperatura. Queste emissioni, possono quindi venire utilizzate, ad esempio, in sensori di temperatura o nelle fotocamere a visione notturna (Figura 8).



Figura 8: Si noti come il corpo di una persona, con temperatura quindi notoriamente maggiore rispetto all'ambiente, emetta un maggior valore di radiazioni.

### **Dispositivi di trasduzione fra correnti e radiazioni**

Abbiamo già parlato brevemente di come le radiazioni infrarosso siano capaci di trasferimento di calore. Essendo quest'ultimo una forma di energia, questa caratteristica di trasmissione viene utilizzata in alcuni dispositivi elettronici per la generazione di correnti. Grazie alla legge della conservazione dell'energia infatti, questi dispositivi trasducono la radiazione entrante in corrente e calore

---

<sup>19</sup>Eventualità decisamente remota, poiché la sensazione di dolore si manifesta già prima che la pelle subisca scottature e permette di allontanarsi in tempo dalla fonte di calore.

proporzionali all'intensità della stessa.

Questa stessa proprietà, viene utilizzata inversamente dai dispositivi in grado di generare radiazioni infrarosse. Qui infatti, sottoponendo il componente a una fonte di energia come quella elettrica, è possibile innalzarne la temperatura, portandolo quindi alla generazione di radiazioni in uscita<sup>20</sup>.

Tornando però a parlare per un momento dei dispositivi di ricezione atti alla generazione di correnti, il principio utilizzato di generare flusso di carica in base all'energia entrante tramite la radiazione, può creare alcuni tipi di problematiche.

Quando il foto-rivelatore (i.e., il sensore) non è orientato verso alcuna fonte luminosa infatti, è possibile comunque leggere un valore in uscita dallo stesso. La causa di questa lettura è dovuta all'effetto termoelettrico che viene generato dal componente stesso. Questo, a temperatura ambiente, possiede infatti sufficiente energia perché avvenga l'emissione di qualche numero di elettroni. Questi, venendo considerati come generati dalla radiazione, andranno poi a formare una corrente uscente, presente anche in assenza di segnale entrante. Questo flusso di carica viene comunemente definito *corrente di buio*.

Per cercare di diminuire il più possibile questo errore, l'unico modo è tenere la temperatura del componente la più bassa possibile. Abbiamo detto infatti come questa generazione di corrente dipenda dalla temperatura del componente, e quindi a sua volta dalla sua energia. Sappiamo anche però che l'energia è proporzionale all'agitazione molecolare, e che questa è minima in presenza dello zero assoluto. Possiamo quindi intuire come abbassando la temperatura del componente, questo, avendo meno energia, possa emettere solo un numero inferiore di elettroni, e quindi generare una minor corrente di buio.

Viceversa, con l'innalzamento della temperatura del componente, otterremo correnti di buio maggiori, e quindi, proporzionalmente, un errore maggiore<sup>21</sup>.

---

<sup>20</sup>Fin quando la temperatura è sufficientemente bassa, si parla di radiazioni infrarosso, quando invece questa supera la soglia di incandescenza, la radiazione emessa diventa quella della luce visibile.

<sup>21</sup>La corrente di buio infatti è nota sommarsi alla corrente di misura

## La riflessione

Come già accennato precedentemente, la misura di distanze o la percezione di prossimità tramite radiazioni elettromagnetiche (e allo stesso modo tramite ultrasuoni) sfrutta il fenomeno della *riflessione*. Questa, è una proprietà per la quale un'onda cambia direzione a seguito dell'impatto con un ostacolo.

A voler approfondire il discorso però, durante un'incidenza non avviene solo una riflessione. L'onda infatti viene parzialmente assorbita dall'oggetto colpito e parzialmente trasmessa attraverso di questo; solo una parte quindi ne risulta effettivamente riflessa.

Questa considerazione ci spinge dunque a prendere in considerazione l'ipotesi che questa perdita di energia possa compromettere il funzionamento di un sensore di rilevazione. In realtà ciò potrebbe essere problematico solo per oggetti relativamente di stanti o per la richiesta di precisioni esageratamente alte.

Per i primi infatti, la perdita a seguito della riflessione verrebbe altresì sommata alla dispersione di energia lungo il tragitto, potendo quindi portare la radiazione ad azzerarsi o risultare impercettibile molto prima rispetto all'incidenza con un materiale con riflessione maggiore. Per quanto riguarda un'alta precisione invece, questa dispersione, dipendente dall'oggetto riflettente, porterebbe alla semplice accettazione di non riuscire a raggiungere precisioni oltre al centimetro per calibrazioni generiche<sup>22</sup>.

Accettando dunque determinate precisioni, decidendo un opportuno range di visualizzazione e calibrando il dispositivo rispetto ad esso, si andrebbe quindi a limitare l'analisi della radiazione ad una distanza entro la quale tutti questi errori risulterebbero sufficientemente piccoli, nella loro somma, da permettere comunque un utilizzo valido della tecnologia.

---

<sup>22</sup>Questa ovviamente può essere aumentata a seguito della calibrazione del dispositivo sulla capacità riflessiva dell'oggetto utilizzato nei test

## Componenti principali

Avendo dato un accenno delle proprietà fisiche entranti in gioco in questo modulo, possiamo ora concentrarci sullo studio della componentistica atta ad utilizzarle. In questo capitolo infatti, si intende entrare maggiormente nel dettaglio riguardo alle caratteristiche di quell'insieme di dispositivi che sono stati utilizzati (o quanto meno presi in considerazione) per lo sviluppo di questo modulo.

Afferenti alla branca dell'optoelettronica<sup>23</sup>; i componenti utilizzati nella sensoristica infrarosso possono variare a seconda dello scopo del sensore ma, in linea di massima, nella maggior parte dei sistemi è possibile trovare:

- Emettitori infrarosso (e.g., LED)
- Fotorivelatori (solitamente un *fotodiodo* o un *fototransistor*)

**L'emettitore infrarosso**, spesso un LED IR, è un componente opzionale ma sfruttato da una grande quantità di sensori basati sull'analisi o utilizzo delle radiazioni (e.g., telecomandi, sensori di prossimità, encoder ottici, etc.).

Nello sviluppo del progetto, è stata utilizzata proprio quest'ultima tipologia di diodo che, come descritto dal nome stesso (i.e., Light Emitting Diode), rappresenta un generatore di radiazioni luminose. Questo è quindi capace di produrre, se sottoposto a una differenza di potenziale e creato per la giusta banda di frequenze, un'emissione di onde del tipo infrarosso.

È stato definito opzionale perché alcuni dispositivi che necessitano solo una lettura della radiazione luminosa ambientale, potrebbero non necessitare di un emettitore della stessa. È anche vero però che molti di questi necessitano invece di interagire con l'ambiente per creare in esso una variazione da misurare.

Un paio di esempi esportano casistiche in cui questo dispositivo sia necessario o meno:

---

<sup>23</sup>Branca che studia i dispositivi elettronici che interagiscono con la luce, fungendo da interfaccia tra il dominio elettrico e quello ottico (e viceversa), e le loro applicazioni

### **Esempio: Fotocamera termica**

Come già detto, le radiazioni infrarosso, anche chiamate radiazioni termiche, sono prodotte dagli oggetti in base alla propria temperatura. Su questo principio di funzionamento lavorano le fotocamere termiche.

Questi dispositivi "scattano" foto della radiazione generata dagli oggetti e possono essere utilizzati per visione notturna o rilevamenti di temperatura.

È ovvio notare come in questi dispositivi sia solo necessario rilevare la radiazione emessa, senza dover in alcun modo modificarne il valore preventivamente.

*In questo caso quindi, un emettitore IR non interverrebbe nella misura.*

Un esempio di rilevazione nella quale un emettitore IR abbia invece una parte attiva:

### **Esempio: Ascensore**

Supponiamo di voler creare un sistema di sicurezza per far in modo che una porta di un ascensore non si chiuda mentre qualcosa la ostruisce.

In questo caso, un fotorivelatore a sé stante non basta. Per funzionare richiede infatti di essere affiancato da un emettitore.

Questi verranno allineati in modo che nel caso standard (i.e., porta libera) il segnale emesso sia rilevato dal sensore. Nel caso un oggetto finisca a ostruire questo fascio, il rivelatore smetterà di ricevere il segnale e la porta dell'ascensore si bloccherà fin quando questo non verrà nuovamente ricevuto.

*In questo caso quindi, la presenza di un emettitore risulta necessaria per creare il fascio da rilevare nella misura.*

**Il fotorivelatore**, anche chiamato fotocellula, è il dispositivo che permette di rilevare la radiazione elettromagnetica fornendo in uscita un segnale



avente un'intensità di corrente o una differenza di potenziale proporzionale all'intensità della radiazione rilevata.

Esistono tipi diversi di fotorivelatore, i quali si distinguono in particolare per la porzione di spettro elettromagnetico che sono in grado di rilevare (Figura 9), e per l'intensità luminosa minima che riescono a misurare.

Il fotorivelatore non è quindi un sensore utilizzato solo in funzione della tecnologia IR, ma a seconda delle caratteristiche può permettere di rilevare anche lunghezze d'onda differenti.

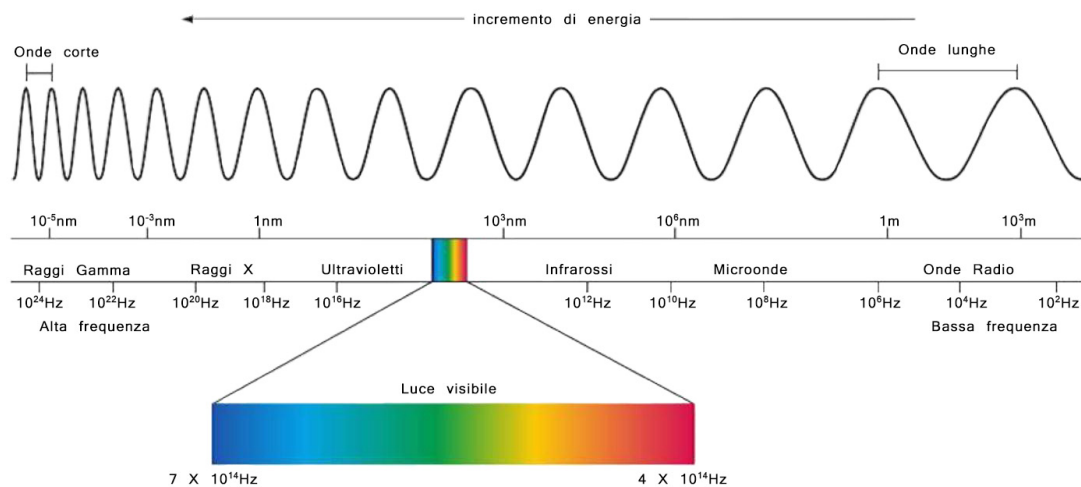
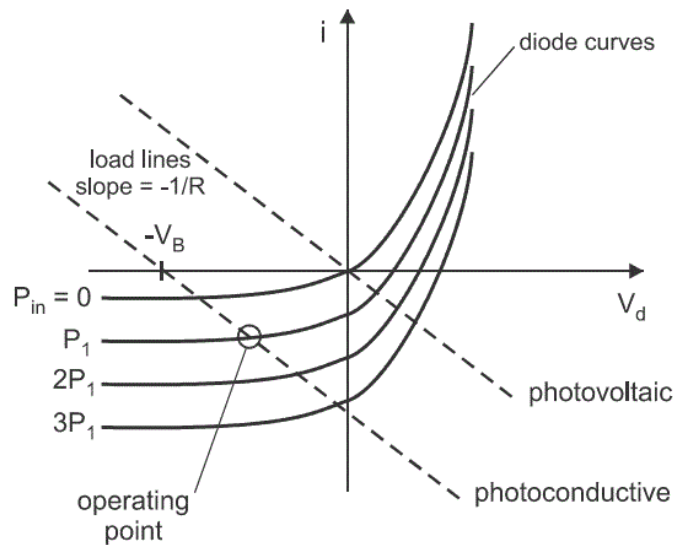


Figura 9: Lo spettro delle radiazioni elettromagnetiche, da quelle di lunghezza d'onda minore a maggiore. Si può notare come la radiazione infrarosso abbia lunghezza d'onda maggiore rispetto a quella della luce visibile.

Utilizzando tecnologia infrarossa però, nel corso dello studio ci si è ovviamente relazionati maggiormente ai dispositivi in grado di rilevare segnali appartenenti a questa banda di radiazioni elettromagnetiche. Nello specifico, i due componenti, appartenenti alla macro famiglia delle fotocellule, su cui si è focalizzata la mia attenzione sono stati:

- fotodiodi
- fototransistor

**Il fotodiode** viene utilizzato come trasduttore da segnale ottico a elettrico, trasformando l'intensità di una determinata lunghezza d'onda in un segnale in corrente (dipendente anche dal potenziale elettrico applicato ai suoi estremi; Figura 10). Questo segnale in corrente in uscita non sarà "binario" (presenza o meno di fonte luminosa) ma "analogico", permettendo quindi di rilevare non solo se sia presente o meno una radiazione infrarosso, ma anche la sua "intensità".



**Figure 14-3** The operating point in a photodiode circuit is determined by the intersection between the load line and the diode  $i$ - $V$  curve. Increasing optical power  $P_{in}$  shifts the  $i$ - $V$  curve downward by an amount  $i_{\lambda} \propto P_{in}$ , moving the operating point down and to the right.

Figura 10: La curva caratteristica di un fotodiode. Nell'immagine si può notare la variazione di questa in base alla polarizzazione.

A livello circuitale, il fotodiode può venir polarizzato in tre modi:

- diretto: si comporta come un comune diode. Non essendo tuttavia progettato per funzionare in questa configurazione, esso non avrà una capacità di corrente tale da suggerirne un simile utilizzo. Il surriscaldamento dovuto a un forte passaggio di corrente potrebbe infatti causare danni agli elementi ottici al suo interno.
- inverso (photoconductive mode): quando la tensione ai capi del fotodi-

do è negativa ( $V_{catodo} > V_{anodo}$ ), questo eroga una corrente (anch'essa negativa; uscente dall'anodo) proporzionale alla potenza ottica P (i.e., proporzionale all'intensità della radiazione luminosa), con sovrapposta la corrente di buio  $i_0$  (i.e., la corrente generata in condizione di buio; corrente minima erogata).

$$I = -i_0 - i_{ph} \quad (1)$$

Un esempio di fotodiodo, in polarizzazione inversa, collegato ad un op-amp è visibile in figura 11

- apolarizzato (photovoltaic mode): quando non sottoposto ad alcuna polarizzazione, la foto-corrente  $i_{ph}$  generata è negativa (i.e., uscente dall'anodo) e di minor intensità. La mancanza di polarizzazione determina inoltre un calo nella temperatura del componente, riducendo anche di molto la corrente di buio<sup>24</sup>.

Se il circuito è aperto o è presente un carico che limiti la foto-corrente, si viene inoltre a creare ai capi del dispositivo una tensione.

Un esempio di fotodiodo, in polarizzazione fotovoltaica, collegato ad un op-amp è visibile in figura 11

Escludendo dunque l'utilizzo nella prima configurazione (i.e., diretta), la scelta fra le due restanti deve considerare punti di vantaggio e svantaggio di entrambe. Nello specifico la configurazione **fotoconduttiva** (inversa) presenta i seguenti vantaggi:

- rapidità di risposta

e svantaggi:

- rumore nel segnale di uscita (alla corrente del fotodiodo è infatti sovrapposta una corrente di buio  $i_0$ ).

La configurazione **fotovoltaica** presenta invece i seguenti vantaggi:

- basso rumore (è infatti ridotta la corrente di buio)

---

<sup>24</sup>Con una grezza approssimazione, nei componenti a base di silicio la  $i_0$  si dimezza ogni 7° centigradi

e svantaggi:

- risposta lenta

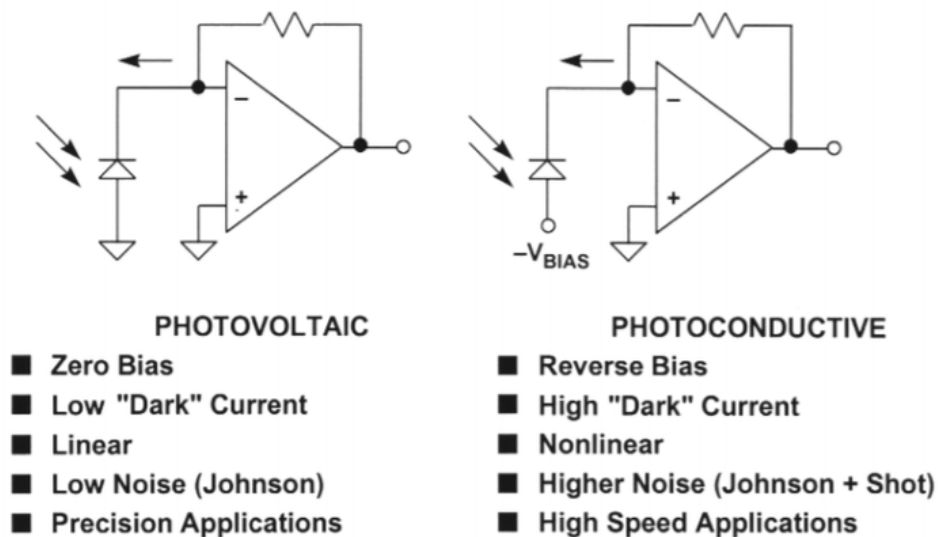


Figura 11: Esempi di configurazione fotovoltaica e fotoconduttiva (inversa).

**Il fototransistor** è, come il fotodiode, un trasduttore di segnale luminoso (Figura 12). Ha due o tre terminali ed è una variante di un normale transistor che presenta, a differenza di questo, una "base" sensibile alla luce. Quando la luce le incide sopra (i.e., sopra la base), il transistor permette il passaggio di una corrente dal collettore proporzionale all'intensità della radiazione luminosa.

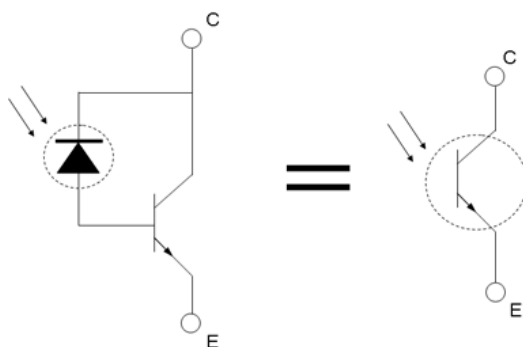


Figura 12: Un'analogia tra fotodiode e fototransistor.

Il fototransistor ha prestazioni simili a quelle del fotodiode, ma con una sensibilità alla luce, in proporzione, molto maggiore. Per contro, è meno sensibile alle radiazioni di bassa intensità luminosa ed ha una risposta nel tempo molto meno rapida<sup>25</sup> (Figura 13).

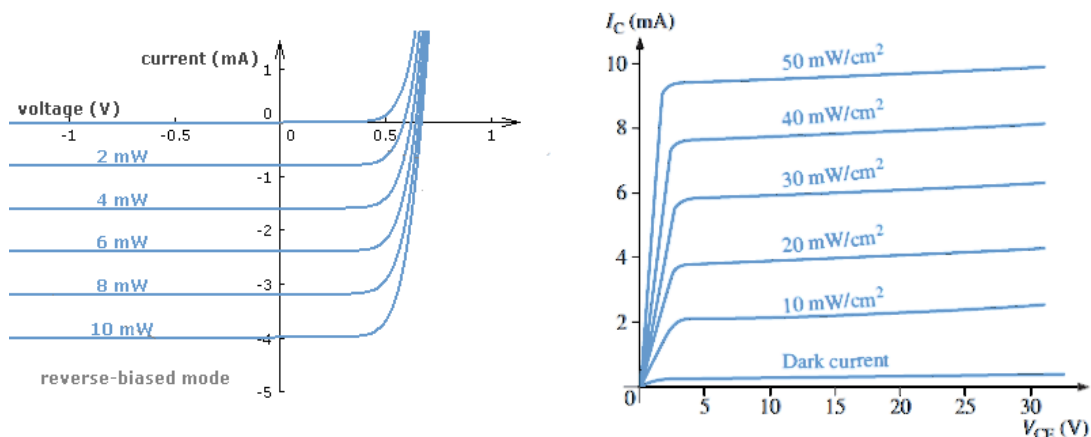


Figura 13: Una comparazione delle curve caratteristiche di un fotodiode e di un fototransistor. È possibile notare come per ottenere 2mA di corrente in uscita il fotodiode necessiti di soli 5mW/cm<sup>2</sup>, mentre il fototransistor richiede 10mW/cm<sup>2</sup>

A livello circuitale il fototransistor viene utilizzato in due configurazioni di base:

- attiva o lineare: fornisce una risposta proporzionale allo stimolo luminoso. La curva caratteristica non è propriamente lineare, ed è meno sensibile alle basse radiazioni di luminosità.
- switch: a causa della ridotta sensibilità alle basse luminosità, è possibile utilizzare il fototransistor come interruttore. In questa configurazione, si distinguono come stato "off" quello relativo alle basse luminosità (i.e., piccole o nulle correnti in uscita) e come stato "on" quello relativo alla regione di saturazione.

<sup>25</sup>Parliamo comunque di velocità minime molte alte, nell'ordine di diversi MHz

A causa della caratteristica non lineare della curva, l'impiego come switch è il più largamente utilizzato. Questo, risulta utile ad esempio in casistiche in cui, per questioni di sicurezza, sia preferibile tenere la parte di controllo di un circuito separata dalla parte di potenza dello stesso<sup>26</sup>, o quando interessi soltanto rilevare un segnale binario, senza curarsi della sua intensità.

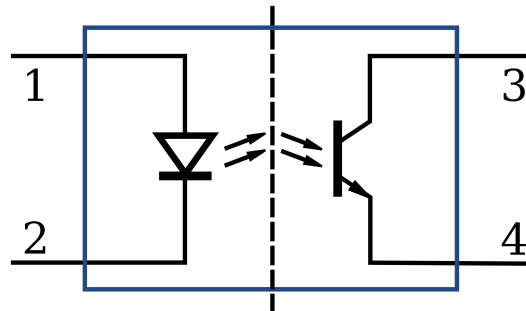


Figura 14: Un optoisolatore. Si può notare come questo contenga al suo interno un fototransistor (l'elemento sulla destra).

### Discussione finale

Ora che sono state discusse alcune tecnologie, possiamo quindi valutare quali di queste siano le migliori per l'implementazione.

Per quanto riguarda l'emettitore, questo è necessario sia per la misurazione delle distanze che per la comunicazione. Per lo scopo, si è scelto di utilizzare un semplice LED capace di generare radiazioni in questa banda.

Per quanto riguarda la scelta della fotocellula per la rilevazione del segnale invece, è stato preferito il fotodiode.

Per questo, è stata scelta la polarizzazione "fotoconduttiva" a causa della sua maggior rapidità di risposta. Il rumore di fondo generato da questa configurazione è infatti normalmente abbastanza piccolo<sup>27</sup> da poter essere trascurato al fine di raggiungere le specifiche richieste dal modulo.

<sup>26</sup>Come viene utilizzato infatti in componenti come l'optoisolatore (Figura 14), che ha lo scopo proprio di effettuare questo isolamento.

<sup>27</sup>Si parla di correnti, a temperatura ambiente, dell'ordine di qualche nA

L'esclusione del fototransistor è infine dovuta al suo meno valido utilizzo in configurazione attiva rispetto a quello come "interruttore", risultando così meno adatto nell'implementazione rispetto al fotodiode.

## **Schede e processi produttivi**

Avendo fin ora trattato della componentistica scelta per l'implementazione del modulo, ci apprestiamo, in questo capitolo, a definire come questa debba venir assemblata durante la produzione finale.

Il modulo che ci si appresta a produrre, a causa della sua probabile complessità di realizzazione, mostra fin da subito la necessità di produrre un circuito con metodi più professionali di una semplice breadboard o millefori<sup>28</sup>.

Questa necessità, ha portato la mia attenzione sui così detti *circuiti stampati*.

### **Il circuito stampato**

Questi supporti permettono di interconnettere tra di loro vari componenti elettronici di un circuito tramite piste conduttive incise su di un materiale non conduttivo.

La produzione di questi circuiti, nonostante gli innegabili vantaggi di eleganza e solidità, si rivela però un problema non semplice da affrontare. A causa della natura DIY (i.e., *Do It Yourself*) ed economica con cui questo progetto nasce, questa procedura produttiva ha la necessità di essere replicabile da privati senza troppa difficoltà e a basso costo.

Nel corso del progetto è stata ovviamente valutata la possibilità di acquistare un circuito prodotto da aziende specializzate, ma a causa del costo non sempre economico per piccole quantità, si è deciso di includere nello studio anche la ricerca di possibilità non troppo complesse di realizzazione autonoma della scheda.

---

<sup>28</sup>Una schedina pre-forata in cui i pin vengono connessi fra loro tramite fili e stagno

## Le tecnologie di produzione

Nel corso del suddetto studio si è venuti a conoscenza di sostanzialmente tre macro-metodi di creazione di PCB (*Printed Circuit Board*):

- chemical etching
- fresatura meccanica
- lamination

Cominciando la discussione dall'ultimo di questi, possiamo immediatamente dire che non rappresenta la soluzione adatta a questo progetto. Questo processo termina infatti con la realizzazione di circuiti stampati multilivello (multi-layer) che, come intuibile dal nome, contengono al loro interno più livelli di piste. Queste, oltre che risultare inutili per la complessità attuale, sarebbero oltretutto particolarmente difficili da raggiungere senza l'utilizzo di fori "plated". Questi, presentano infatti pareti rivestite di materiale conduttore e permettono quindi una conduzione anche senza essere completamente coperti di stagno. In un progetto homemade non è però possibile la creazione di tali tipologie di fori, rendendo il raggiungimento delle piste particolarmente arduo.

Abbiamo assunto temporaneamente che il problema principale di questa tecnica fosse legato al raggiungimento delle piste interne con lo stagno, in realtà la difficoltà è rappresentata proprio dalla creazione della scheda che, per la laminazione di tali "livelli" conduttivi, richiede tecnologie molto costose ed avanzate, totalmente fuori dalla portata di una persona normale.

Continuando in questo ordine inverso, parliamo ora del processo di fresatura meccanica. Questo viene effettuato tramite l'utilizzo di macchine CNC, ovvero a "*controllo numerico*". Questi dispositivi, in questo caso specifico le frese, vengono utilizzati per lavorazioni ad altissima precisione, la quale può addirittura raggiungere il micron.

Il processo produttivo è concettualmente semplice e consiste nel rimuovere la parte eccedente di materiale conduttore tramite la macchina.



Dispositivi come questi sono ovviamente molto costosi, ma non è improbabile riuscire a trovarne in qualche istituto tecnico. Ne è quindi preso in considerazione l'utilizzo.

Terminando infine con la prima tecnologia produttiva, questa utilizza acidi per corrodere la componente rameata della basetta<sup>29</sup>. Questa viene infatti esposta all'azione di composti che ne rimuovono la zona non protetta (tramite film o maschere; Figura 15).

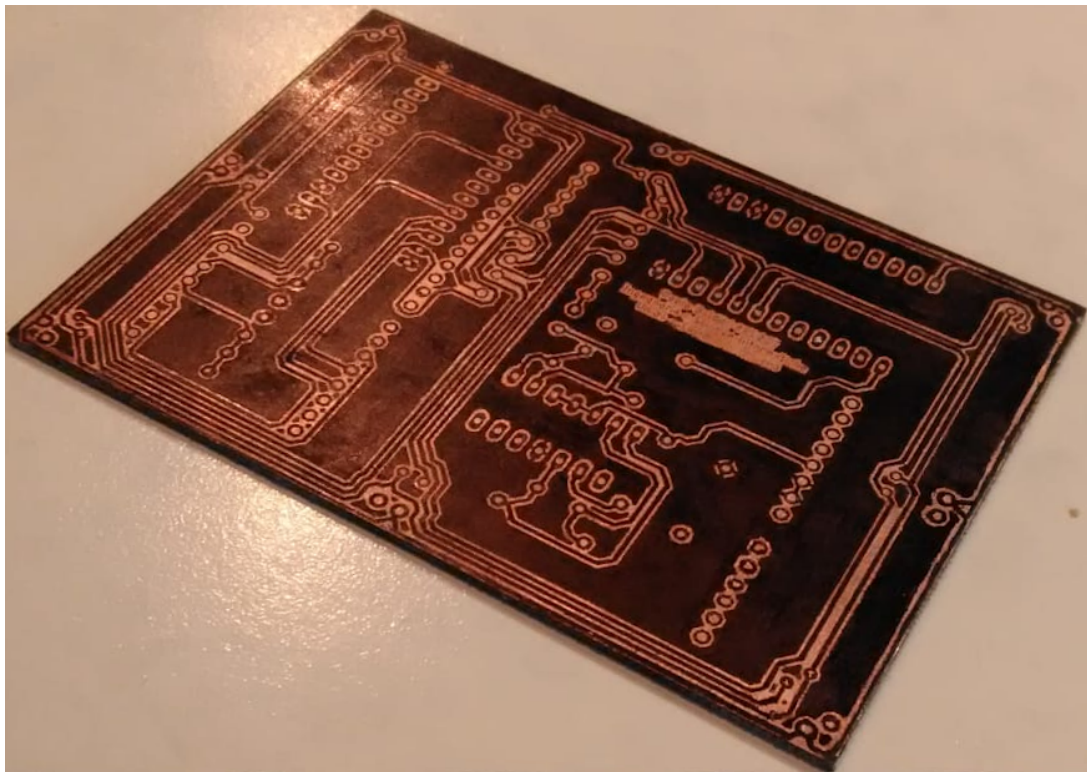


Figura 15: Il primo test di creazione autonoma di un circuito stampato prima di essere immerso nell'acido. Si può notare la maschera nera utilizzata per proteggere la componente rameata.

Se anche quest'ultimo processo è quindi concettualmente semplice, è però necessario mettere in evidenza almeno alcuni metodi di trasferimento della maschera di protezione delle piste:

---

<sup>29</sup>Viene normalmente utilizzato del cloruro ferrico:  $FeCl_3$

- manuale
- toner transfer
- fotoresistiva (bromografo)

Il primo di questi è sicuramente il più intuitivo, e consiste nel deposito, con un pennarello, di uno strato di colore in corrispondenza delle piste, andando a formare la maschera.

Nonostante sia il più intuitivo, per la maggior parte dei progetti, escludendo quelli di dimensioni molto ridotte, risulta estremamente lento e impreciso. È infatti necessario tracciare ogni pista singolarmente.

Questo processo è stato descritto solo per una questione di completezza, ma è veramente il più sconsigliato in questo progetto.

Il toner transfer è una tecnica basata, come dice il nome, sul trasferimento di toner da una stampa su carta alla bassetta. È un procedimento molto più veloce e preciso del tracciare le piste a mano e produce buoni risultati.

Il trasferimento può essere ottenuto con due metodologie, quella a caldo e quella a freddo.

La prima basa il processo sullo "scioglimento" del toner, che fa quindi presa sulla bassetta, creando la maschera.

Nell'atto pratico, questo si può ottenere premendo il foglio di carta sul supporto con un ferro da stiro caldo.

La seconda si ottiene invece tramite un solvente (spesso del semplice acetone) che viene cosparso sulla carta provocando lo scioglimento del toner. Questo viene quindi pressato e trasferito sulla scheda.

Quest'ultimo processo è stato utilizzato per la produzione della bassetta visibile in figura 15.

È importante notare come tipologie differenti di toner e di carta producano effetti anche molto diversi. Per questa tecnica sono infatti reperibili online componenti (carta e toner) appositi per il processo.

La tecnica fotoresistiva, a differenza di quella sfruttante il toner, utilizza strumentazione aggiuntiva e un poco più ricercata, ma nulla che esuli dal campo homemade. Per applicare questa tecnica è innanzitutto necessario l'utilizzo di basette presensibilizzate, ovvero rivestite di una patina di resina fotosensibile. Quando queste vengono esposte a radiazioni ultraviolette (UV), la resina che le ricopre si indurisce, creando la maschera che andrà a proteggere la componente ramata. Per far in modo che questa indurisca nei soli punti corrispondenti alle piste, si utilizzano fogli di acetato con sopra stampato lo schema del circuito da trasferire. Una volta che questo sarà stato trasferito sulla basetta, sarà poi possibile immergere questa nell'acido per sciogliere il rame non protetto. Le schede utilizzate in questo processo possono essere trovate facilmente in commercio, a prezzi maggiori di quelle non patinate, o possono essere sensibilizzate autonomamente tramite l'utilizzo di uno spray ad hoc, spesso non troppo preciso, ma che è reperibile ad un costo notevolmente inferiore rispetto alle basette presensibilizzate.

Il bromografo, ovvero la lampada UV utilizzata nel processo, può essere invece tranquillamente sostituito, per produzioni homemade, da un comune fornello per la ricostruzione delle unghie (anche se questo richiederà qualche decina di minuti in più per l'impressione della maschera).

Quando avremo ottenuto il nostro circuito stampato, l'ultimo passo che sarà necessario all'utilizzo dello stesso sarà la foratura della basetta. Questo procedimento può essere fatto molto semplicemente con un trapano a colonna o, stando più attenti a non rompere le punte, anche con un trapano normale.

## **Implementazioni proposte**

Avendo descritto e definito le tecnologie e le caratteristiche che il modulo dovrà fornire, è quindi ora possibile valutare la macro-struttura che questo dovrà avere. In questo capitolo, saranno quindi esposte le varie implementazioni, di relativamente alto livello, che sono state proposte per il modulo. Per ognuna di

queste, saranno poi valutati i punti di forza e di svantaggio e le caratteristiche salienti.

In tutte queste configurazioni, è possibile notare la presenza di emettitori e ricevitori di radiazioni infrarosso, questi sono infatti considerati elementi essenziali per il funzionamento del modulo, dovendo emettere e ricevere impulsi ogni qual volta si necessiti di comunicare o di misurare distanze.

### Implementazione I - Modulo base

La prima di queste proposte d'implementazioni, il cui schema è visibile in Figura 16, è composta da 2 macro-blocchi circuitali. Questi, ovvero il blocco del circuito di condizionamento e quello di gestione degli interrupt, assolvono rispettivamente alle funzioni di modifica delle caratteristiche del segnale trasdotto dai ricevitori e di generazione di interrupt alla ricezione di messaggi di comunicazione.

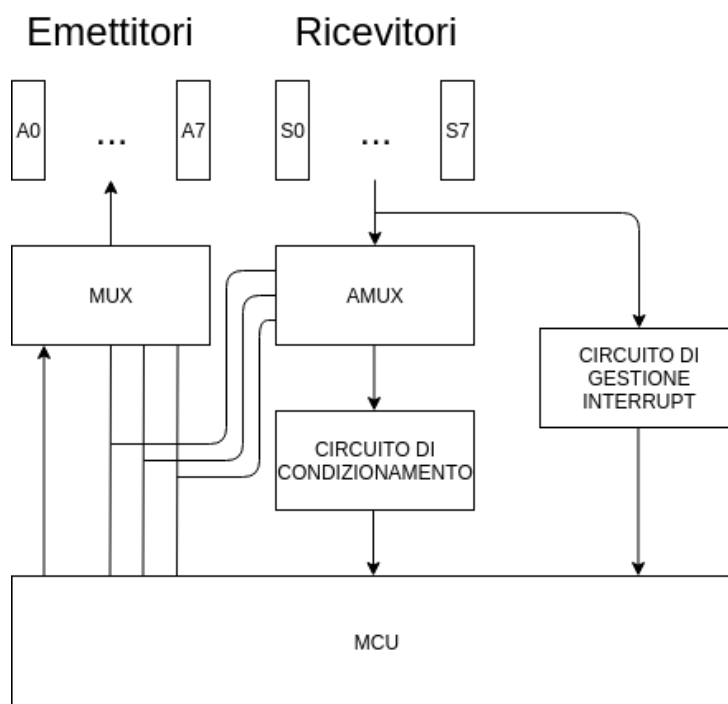


Figura 16: Schema della prima configurazione proposta.

In questa configurazione, si è proposto un approccio a *canali*. Ogni emettitore è infatti accoppiato ad un singolo ricevitore tramite la selezione di una porta su un multiplexer. Questa caratteristica limitante, semplifica il lavoro sia del programmatore sia dello sviluppatore del modulo, senza causare confusione sul possibile utilizzo disaccoppiato dei componenti.

Questa limitazione presenta inoltre un ulteriore vantaggio. È infatti possibile utilizzare una logica di comunicazione direzionale, dove non risulta necessario l'invio di messaggi broadcast, ma questi possono essere inviati anche a un solo dispositivo dialogante su di un singolo canale.

Nonostante i vantaggi, questa proposta presenta però due particolari controindicazioni.

La prima consiste nell'impossibilità di inviare messaggi broadcast che non siano simulati. La necessità di utilizzare un canale di comunicazione richiede infatti, per l'invio di questi, un ciclo in cui il messaggio venga reinviato su ogni uscita. Se questa può non essere una limitazione grave per rapidi messaggi informativi, può diventarlo se si intende inviarne di più lunghi e onerosi di tempo.

A questo proposito, una semplice modifica circuitale che può permettere questo invio broadcast consiste nell'utilizzare un pin di uscita collegato, attraverso un numero adeguato di diodi, ai vari emettitori, permettendo quindi di comandarli contemporaneamente.

La seconda limitazione consiste invece nella ricezione dei suddetti messaggi. Questi infatti richiedono un circuito a parte che binarizzi i valori letti dai ricevitori. Il problema è che questa configurazione, per risparmiare pin e spazio, utilizza un solo ingresso per tutti i canali, impedendo effettivamente una ricezione contemporanea da più fonti.

Quest'ultima limitazione presenta però anche un aspetto interessante. Essendo la ricezione comune su di un solo pin infatti, a seguito di un cambio di sensore ricevente il messaggio (a causa magari dello spostamento dell'emettitore), rimane comunque possibile la ricezione corretta dello stesso.

### **Esempio: Robot in movimento**

Supponiamo ad esempio di avere due robot in movimento. Il robot A comincia a inviare un messaggio al robot B che viene ricevuto sul sensore presente sul lato destro. Il robot A continua però a muoversi e il messaggio finisce di essere inviato quando il segnale viene percepito dal sensore anteriore-destro. In questo caso, se i robot si trovano ad una distanza sufficientemente vicina, il messaggio viene ricevuto correttamente anche a seguito di un cambio di ricevitore.

Concludendo, questa configurazione presenta le seguenti caratteristiche:

- 7 (o 8 nella variante broadcast) pin utilizzati: 5 pin di output (emettitori, multiplexer e uno di controllo) + 2 di input (analogico, interrupt)
- invio dei messaggi canalizzato/direzionale
- una ricezione alla volta
- ricezione dei messaggi senza distinzione del canale

### **Implementazione II - Invio broadcast**

La seconda di queste proposte di implementazioni risulta molto simile alla prima, condividendo buona parte del circuito e della logica. I punti di differenza da questa, visibili in figura 17, sono la sostituzione dell'invio di messaggi canalizzato con l'invio broadcast.

Questa variazione, rispetto alla prima, permette di risparmiare un pin per l'invio di messaggi broadcast, ma perde la capacità di comunicazione direzionale.

La scelta fra le due ricade quindi sul tipo di utilizzo richiesto ma, considerando la capacità della prima di sopperire via software alla mancanza di invio broadcast, questa risulta leggermente meno versatile.

Concludendo, questa configurazione presenta le seguenti caratteristiche:

- 7 pin preventivati: 5 pin di output (emettitori, multiplexer e uno di controllo) + 2 di input (analogico, interrupt)

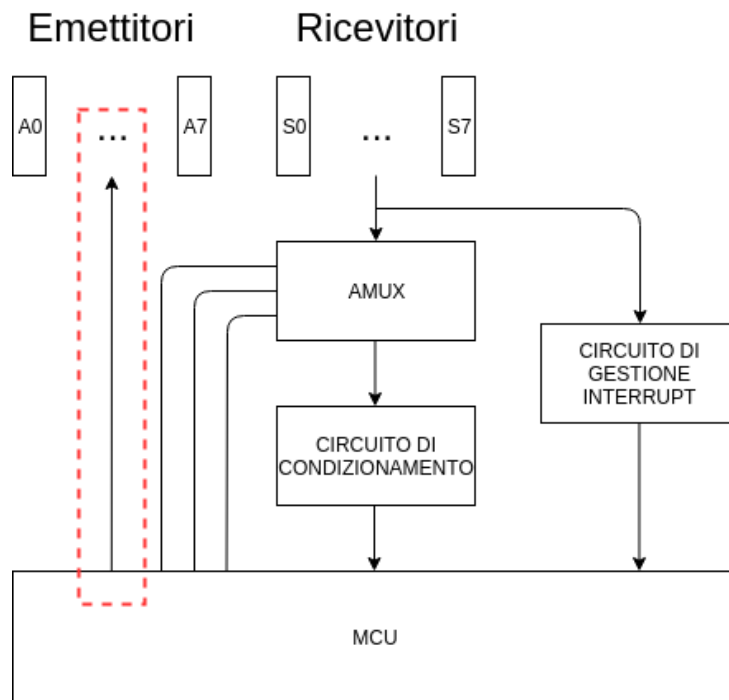


Figura 17: Dettaglio delle differenze della seconda configurazione dalla prima.

- invio dei messaggi solo broadcast
- una ricezione alla volta
- ricezione dei messaggi senza distinzione del canale
- costo minimamente inferiore alla prima soluzione

### **Implementazione III - Separazione dei trasduttori**

La terza di queste implementazioni, sempre rapportata alla prima, non cambia molto sul piano circuitale (vedasi Figura 18), ma differisce notevolmente invece sul piano logico.

Questa nuova proposta infatti, perde il concetto base di accoppiamento fra emettitore e ricevitore, andando a separare completamente i due. Non saremo quindi più in grado, come accadeva per la prima o la seconda, di utilizzare un (almeno) emettitore e un ricevitore contemporaneamente, ma dovremo utilizza-

re i due cambiando canale ed utilizzando il pin dell'amux sia come input che come output a seconda dell'evenienza<sup>30</sup>.

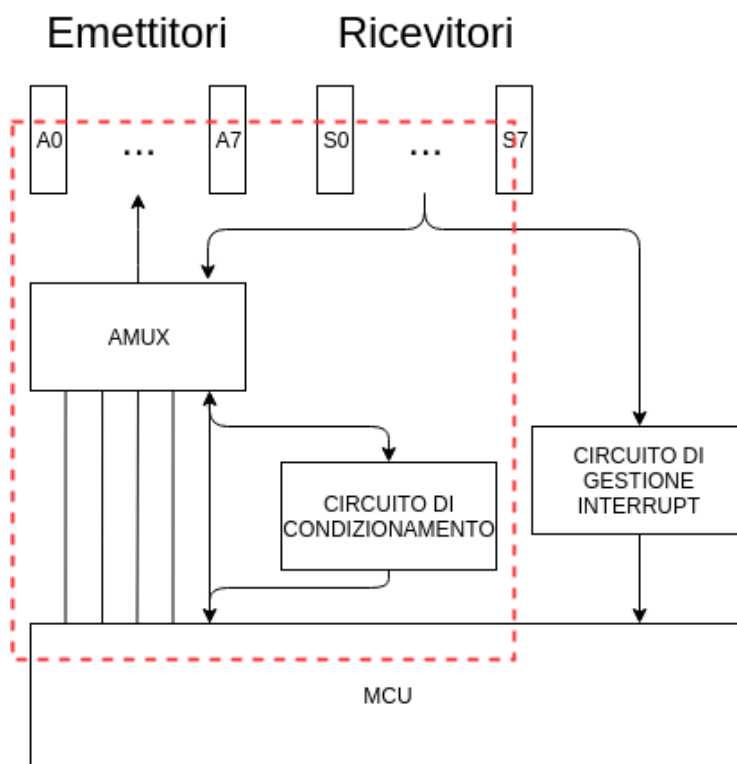


Figura 18: Dettaglio delle differenze della terza configurazione dalla prima e seconda.

Questa soluzione lascia indubbiamente molta più libertà all'utilizzatore del modulo, ma allo stesso tempo risulta incredibilmente macchinosa e soggetta ad errori di utilizzo.

Avendo ora terminato la descrizione della proposta, ritengo necessario inserire una piccola nota per chiarire un possibile dubbio di facile comparsa. Ci si potrebbe infatti domandare se questa richiesta di "switching" di canali possa creare, a livello hardware, latenze troppo elevate. In realtà, i circuiti integrati moderni possiedono capacità di commutazione fra stati incredibilmente veloci, riuscendo quasi sempre a battere in velocità l'esecuzione di istruzioni da parte

<sup>30</sup>Questo ovviamente richiede l'utilizzo di diodi, porte NOT e transistor utilizzati come switch allo scopo di non sovrapporre invio e ricezione



di un microprocessore<sup>31</sup>. Risulta quindi improbabile che l'implementazione hardware riservi problematiche di questo tipo.

Quello che invece potrebbe creare alcuni problemi potrebbe essere il tempo necessario all'operazionale per andare a regime. Questo potrebbe infatti essere maggiore della durata residua dell'impulso generato dall'emettitore. Allo scopo di proporre una soluzione teorica a questo problema, si è pensato di introdurre un condensatore affiancato al LED in uscita. Questo permetterebbe quindi, in teoria, di mantenere l'impulso per un tempo sufficiente al raggiungimento dello stato di regime da parte dell'operazionale. Sarebbero però comunque necessari test, soprattutto relativi alla comunicazione, per valutare la fattibilità della proposta.

Concludendo, questa configurazione presenta le seguenti caratteristiche:

- 7 pin preventivati: 5 pin di output (multiplexer e uno di controllo) + 1 di input (analogico, interrupt) + 1 variabile (input o output)
- invio dei messaggi canalizzato/direzionale
- una ricezione alla volta
- ricezione dei messaggi senza distinzione del canale
- alto rischio d'errore nell'utilizzo
- circuito più complicato
- costo minimamente inferiore alla prima soluzione

#### **Implementazione IV - Controllore integrato**

La quarta implementazione analizzata, a confronto delle precedenti, è basata su una logica del tutto differente (Figura 19).

Quest'implementazione difatti prende in considerazione la creazione di un modulo attivo, ovvero avente la possibilità di essere in funzione anche se non

---

<sup>31</sup>Un' ATmega328 ad esempio, riesce ad eseguire, a 20MHz, 20MIPS (milioni di istruzioni al secondo), che corrispondono a circa 1 istruzione ogni 50ns; un HCF4097 d'altro canto, ha una velocità di commutazione tipica di 30ns.

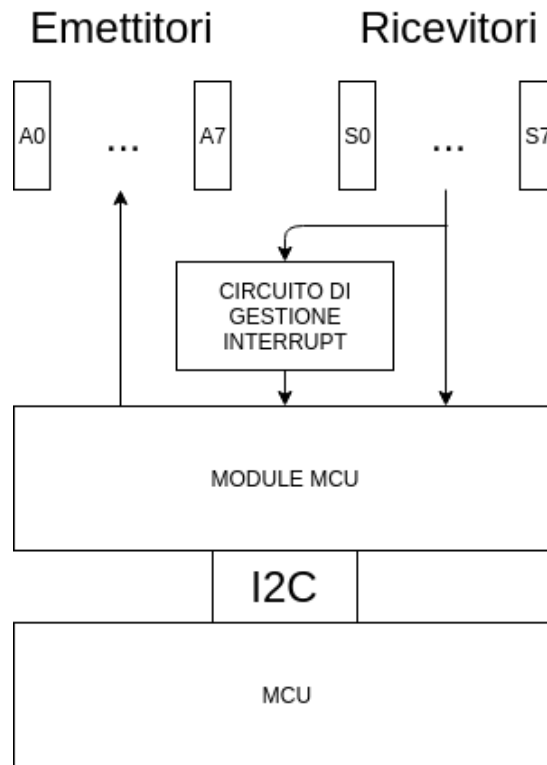


Figura 19: Schema della quarta configurazione proposta.

collegato a schede esterne. Questa situazione, ottenibile fornendo il modulo di un microcontrollore a parte, di potenza ridotta in confronto alla scheda madre, fornirebbe varie possibilità aggiuntive.

In presenza di schede **senza ingressi analogici** ad esempio, questa soluzione garantirebbe comunque il funzionamento senza l'utilizzo di moduli adattatori contenenti ADC<sup>32</sup>.

Per schede meno potenti invece, permetterebbe una sorta di **multitasking**, almeno per quanto attinente al discorso comunicazione/sensoristica ambientale. Se opportunamente configurato inoltre, rappresenta l'unica vera possibilità per la **ricezione contemporanea** di messaggi da più canali (almeno senza l'utilizzo di un numero spropositato di pin).

Per schede con ridotto numero di connettori, si presenta infine la caratteristica più importante. Avendo l'intenzione di sviluppare questa tipologia di modulo

<sup>32</sup>Convertitori analogico-digitale

attorno alla tecnologia  $I^2C$  (o affini) infatti, si permetterebbe un **risparmio di pin** veramente notevole.

Essendo reperibili in commercio schede molto economiche ed adatte a funzionare come unità di calcolo di un modulo separato, questa proposta acquista quindi un senso veramente importante e, a seguito di modifiche successive al progetto, sarebbe stata senza dubbio la scelta migliore. Purtroppo però, non essendo ai tempi ancora stati in grado di definire la possibile autonomia del robot, il consumo incrementato dalla seconda unità di calcolo si sarebbe potuto rivelare fatale allo scopo del progetto. Questa soluzione non ha quindi, purtroppo, mai visto la luce.

Concludendo, questa configurazione presenta le seguenti caratteristiche:

- 2 pin utilizzati: ( $I^2C$ )
- comunicazione canalizzata/direzionale e/o broadcast
- possibilità di ricezione contemporanea da più canali
- individuazione del canale di ricezione del messaggio
- nessuna necessità di ingressi analogici nella scheda madre
- multitasking
- consumo e costi maggiori

### **Implementazione V - Ricezione canalizzata**

Con questa implementazione torniamo sostanzialmente a fornire alternative alla versione iniziale del progetto.

Come distinguibile dallo schema (Figura 20), questa nuova versione separa la ricezione dei messaggi, facendo in modo che ognuno di essi sia ora leggibile solo selezionando il canale relativo.

Questa modifica introduce sia aspetti positivi che negativi. Come evidente, rende logicamente più chiara la ricezione, conoscendo ora il canale da cui arriva il messaggio.

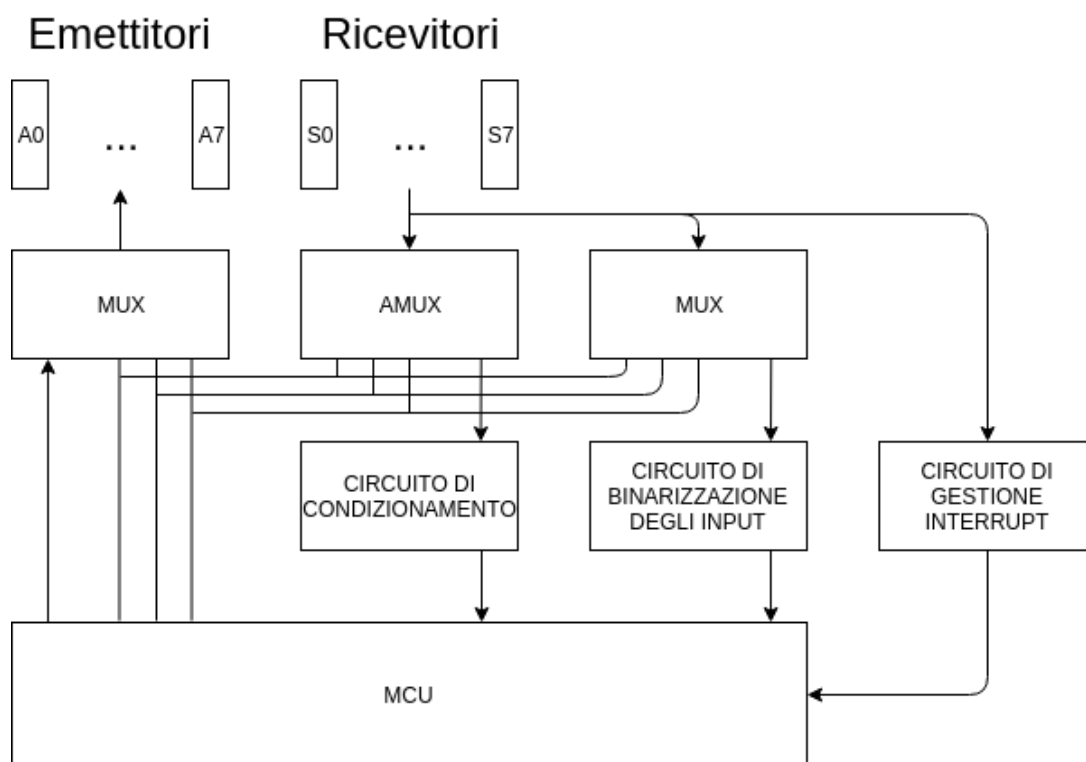


Figura 20: Schema della quinta configurazione proposta.

Questa soluzione risulta anche meno sensibile ai disturbi. Nelle precedenti infatti, messaggi contemporanei da più canali rendevano impossibile la ricezione degli stessi, andando a sovrapporsi. In questa versione invece, una volta sintonizzatisi su di un canale, è possibile ricevere messaggi senza che questi vengano resi illeggibili dall'arrivo di altri.

Come aspetti negativi si presenta la macchinosa operazione di selezione del canale di comunicazione, a seguito di interrupt, richiedente quindi un segnale iniziale per notificare l'intenzione di inviare messaggi e uno secondario per la ricerca del canale corretto<sup>33</sup>.

Infine, come ultimo ma non meno importante aspetto negativo, con questa configurazione il numero di componenti elettroniche richieste crescerebbe notevolmente, rendendo quindi ardua, se non impossibile per un progetto

<sup>33</sup>Questo a meno di utilizzare una codifica dell'identificativo del canale che ha generato l'interrupt, passandola come valore analogico. In questo caso, il segnale iniziale di avvio della comunicazione sarebbe sufficiente

con orientamento homemade, l'operazione di sbroglio o di saldatura della componentistica.

Concludendo, questa configurazione presenta le seguenti caratteristiche:

- 8 pin preventivati: 5 pin di output (emettitori, multiplexer e uno di controllo) + 3 di input (analogico, interrupt, comunicazione)
- comunicazione canalizzata/direzionale
- una ricezione alla volta
- individuazione del canale di ricezione del messaggio
- assenza di errori derivanti da comunicazioni contemporanee
- utilizzo macchinoso e con buona probabilità d'errore

### **Implementazione VI - IRUSS Module**

Questa configurazione nasce da una notevole variazione della prima ed è stata valutata durante lo sviluppo più a basso livello del modulo. Il suo studio è derivato da una situazione imprevista, scaturita quando il mio compagno di progetto, A. Lombardi, si è ritrovato a necessitare di un sensore di prossimità quando l'implementazione del modulo non era ancora terminata. In questo contesto, ha dunque dovuto temporaneamente utilizzare un sensore ad ultrasuoni. Volendo però permettergli il mantenimento del codice che aveva strutturato sopra di esso, e presupponendo al contempo la possibilità di modifica del modulo, si è provveduto alla strutturazione di una nuova versione di questo, diventata poi quella finale, che permettesse *anche* l'utilizzo di quest'ultima tecnologia.

In quest'ultima implementazione, che ha preso infine il nome di IRUSS Module, l'utilizzo dei sensori ad ultrasuoni risulta, relativamente alla misurazione delle distanze, opzionale ed esclusivo, ovvero utilizzabile in modo alternativo alla tecnologia infrarosso.

È importante sottolineare come l'esclusività delle due tecnologie sia infatti *solo e soltanto* relativa alla misurazione delle distanze. La comunicazione infrarosso rimane infatti utilizzabile, opzionalmente, in entrambe le configurazioni.

Oltre a questa dualità tecnologica, tramite questa versione è anche possibile utilizzare la sensoristica di distanza *senza* utilizzare necessariamente anche la tecnologia comunicativa. Questa possibilità permette di lasciare liberi due pin che saranno quindi di possibile utilizzo per altri scopi.

Quest'ultima versione risulta quindi configurabile in quattro tipologie di utilizzo:

- distanza IR - comunicazione IR
- distanza IR - nessuna comunicazione
- distanza USS - comunicazione IR
- distanza USS - nessuna comunicazione

Le seguenti, sono invece le caratteristiche afferenti a quest'implementazione:

- da 5 a 7 pin utilizzati: 4 o 5 pin di output (emettitori, multiplexer ed eventualmente uno di controllo) + 1 o 2 di input (analogico, ed eventualmente interrupt/comunicazione)
- invio dei messaggi canalizzato/direzionale
- una ricezione alla volta
- ricezione dei messaggi senza distinzione del canale
- molteplici configurazioni di utilizzo

### **Discussione finale**

Come già accennato durante l'esposizione delle varie implementazioni proposte, molte di queste sono molto simili fra loro e differiscono perlopiù per qualche dettaglio. Risulta comunque doveroso sottolineare come questi punti di differenza portino cambiamenti anche sostanziali nell'utilizzo del modulo.

Avendo già stilato, per ogni implementazione, una lista delle caratteristiche salienti, non si discorrano nuovamente i vantaggi dell'una o dell'altra, ma si considererà soltanto quale sia la più adatta al progetto.

Come già spiegato durante la trattazione, l'implementazione contenente un controllore integrato, a seguito degli sviluppi relativi all'alimentazione, sarebbe senza dubbio risultata la migliore. Questa infatti avrebbe permesso un notevole aumento di prestazioni e capacità del modulo.

Sfortunatamente, essendosi questi sviluppi mostrati troppo tardi, si è dovuta preferire, a questa, l'ultima implementazione proposta.

## **Dettagli dell'implementazione hardware**

In questo capitolo verranno mostrati e discussi alcuni dettagli dell'implementazione hardware del modulo. È importante notare come questi siano stati frutto di un processo di raffinamento successivo e come, in alcuni casi, potrebbe essere necessario descrivere i passi che hanno portato ad una determinata scelta implementativa.

### **Selezione del canale**

In quasi tutte le implementazioni ad alto livello che abbiamo visto nel capitolo precedente, compresa quella scelta, è presente un dispositivo, chiamato (A)MUX (i.e., multiplexer), che permette la selezione del trasduttore (o della coppia sensore/emettitore).

Nell'implementazione hardware questa struttura non cambia particolarmente. È stato infatti utilizzato un circuito integrato, l'HCF4097, che incorpora al suo interno due multiplexer analogici. In questo dispositivo, i pin di selezione del canale sono comuni, assecondando la scelta dell'utilizzo canalizzato e permettendo quindi di selezionare lo stesso canale contemporaneamente in entrambi i multiplexer.

Nell'implementazione del modulo è stato poi utilizzato un pin messo a disposizione da questo dispositivo (l'inhibit) per controllarne quello che può essere visto come lo stato di accensione/spengimento. Questo pin infatti determina o meno il blocco dell'inoltro del valore dall'ingresso all'uscita del multiplexer.

## Condizionamento del segnale

Abbiamo precedentemente parlato dei sensori utilizzati nel modulo e abbiamo spiegato come questi forniscano una misura in corrente. Questa, non può essere però direttamente letta da una scheda. Dato il normale funzionamento in tensione infatti, questa misura deve preventivamente essere convertita in un valore leggibile.

Quest'operazione è stata effettuata tramite un circuito, detto *convertitore I/V differenziale*, ottenuto tramite un amplificatore operazionale (Figura 21). La scelta di utilizzare questo circuito, invece di una banalissima resistenza sottoposta alla legge di Ohm, deriva dalla volontà di rendere la conversione della corrente indipendente dal carico collegato in uscita.

Questa scelta è stata effettuata non tanto per la scheda su cui si suppone l'utilizzo ma, a causa della volontà di modularizzazione, per un possibile dispositivo collegato all'uscita. Il modulo potrebbe infatti non essere necessariamente collegato direttamente a una scheda ma, ad esempio per necessità di condizionamento del valore di tensione uscente (magari per una scheda con supporto a valori minimi e/o massimi differenti), potrebbe dover venir collegato ad un dispositivo intermedio. In quest'evenienza, non sapendo cosa potrebbe venirvi collegato all'uscita, è stato preferito sostituire alla resistenza utilizzata durante i test questo equivalente circuito che permette però questa separazione. Questa scelta è stata dunque effettuata con l'intento di rendere il più possibile stabile l'utilizzo del modulo in altre realtà.

Ora che abbiamo spiegato il motivo dietro la scelta di quest'implementazione circuitale, può risultare utile fornire qualche formula e dare un'idea generica di come avvenga il dimensionamento dei componenti.

La seguente equazione rappresenta la funzione di condizionamento del circuito sopra descritto:

$$V_o = RI_+ - RI_-$$

Nel nostro caso però, essendo la corrente  $I_-$  entrante nel terminale invertente con segno positivo (è infatti negativa rispetto al solo fotodiode, uscendo dal suo anodo), e volendo ottenere in uscita un'escursione da 0 a 5V, in base alla



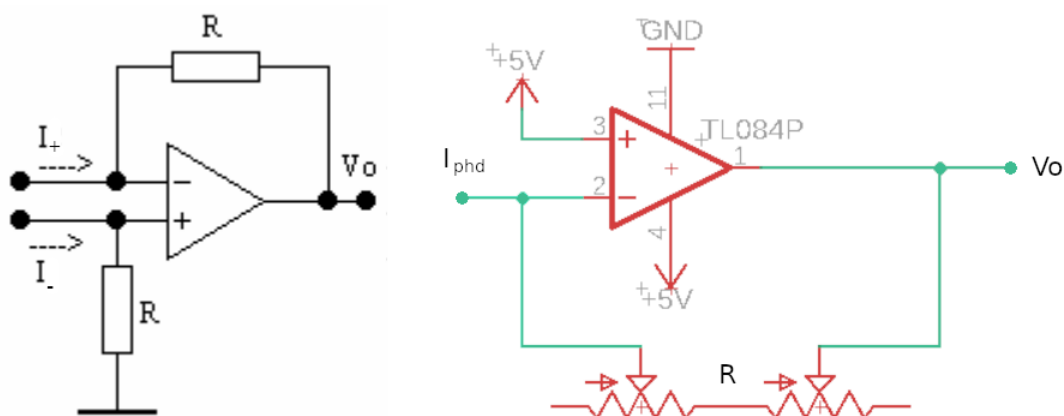


Figura 21: A sinistra, lo schema base di un convertitore I/V differenziale. A destra, l'implementazione dello stesso nel progetto. Si può notare come in quest'ultima il valore in ingresso sul pin + sia fisso.

formula dovremo porre il terminale non invertente dell'operazionale (i.e., il pin +) a 5V.

Ponendo quindi  $RI_+ = 5V$  e  $I_- = I_{phd}$ , otterremo:

$$V_o = 5 - I_{phd}R$$

Essendo quindi, come già detto, la corrente  $I_{phd}$  positiva, l'uscita  $V_o$  dell'operazionale fornirà (una volta scelta accuratamente la resistenza R) un valore nel range 0-5V.

### Guadagno a controllo luminoso

In questo paragrafo andremo a descrivere una parte circuitale sviluppata con l'intento di ampliare il range di funzionamento del dispositivo e dar la possibilità a questo di auto adattarsi a diverse condizioni ambientali.

Abbiamo già parlato della sensibilità dei sensori utilizzati nel modulo e di come un loro valido utilizzo richieda un'accurata calibrazione della scheda. Questo, avviene tramite la regolazione della resistenza dell'operazionale che determina il guadagno dello stesso.

Durante i vari test su breadboard però, ci si è accorti di come una regolazione

statica di questo tipo, seppur ottima in contesti con caratteristiche costanti (come un laboratorio), rischi di non essere sufficientemente valida in ambienti nei quali la radiazione ambientale, seppur molto omogenea, ha la possibilità di variare.

Allo scopo di migliorare il problema, è stato deciso di aggiungere un secondo circuito di condizionamento, relativo al guadagno dell'operazionale, che permetta quantomeno di mitigare la questione. Per fare ciò, questo dovrebbe quindi occuparsi della regolazione automatica della resistenza in base alla luminosità ambientale presente.

Questa proprietà è stata ottenuta inserendo, in parallelo al resistore regolabile dell'operazionale, un circuito funzionante come una *resistenza controllata in corrente* (i.e., dipendente da questa).

Per ottenere una configurazione di questo tipo è stata utilizzata la corrente uscente da un fotodiode. Questa, una volta regolata da un trimmer, finisce sulla base di un transistor (Figura 22), determinando quindi un passaggio di corrente, dal collettore all'emettitore, proporzionale al valore sulla base. Questo flusso determina quindi una variazione della tensione ai capi del transistor, rappresentando di fatto il funzionamento di una resistenza variabile.

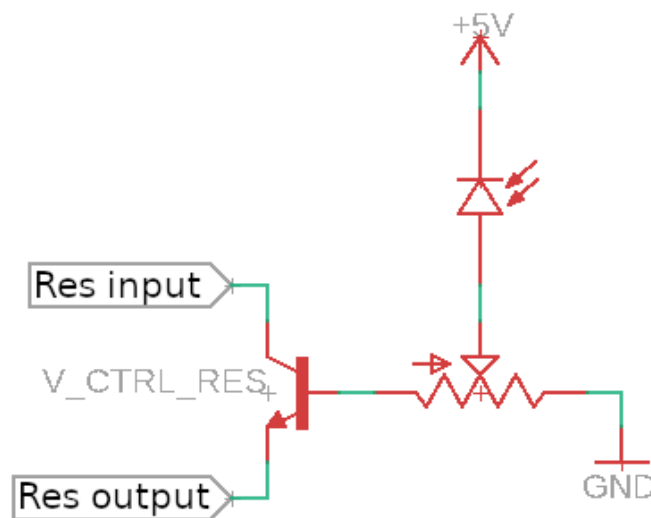


Figura 22: La configurazione della resistenza virtuale.

## **Circuito di binarizzazione dei messaggi**

La configurazione implementativa scelta, come già descritto, ha fin dall'inizio compreso la ricezione dei messaggi tramite ingresso comune. L'implementazione hardware di questa possibilità ha però richiesto più tentativi. Nelle fasi iniziali di progetto era infatti stato commesso un errore derivante dalla non considerazione di un caso limite, il quale ha richiesto, a causa della difficoltà di individuazione, più di un tentativo per essere risolto. Ripercorreremo quindi i passaggi logici che hanno portato alla versione finale di questa parte del modulo, analizzando gli errori e le migliorie che sono comunque state apportate ad ogni versione.

**I -** Fin dalla prima versione del modulo, si era valutata la possibilità di considerare i livelli logici in ingresso da un sensore in maniera binaria, capacità che avrebbe permesso quindi di utilizzare un tipo di comunicazione classica. Nel corso dei test era dunque stato sviluppato un circuito comprendente, in cascata, un convertitore I/V differenziale ed un circuito comparatore (i.e., un circuito che fornisse in uscita  $0V$  se la tensione  $V_-$  fosse risultata maggiore della tensione  $V_+$  e  $5V$  viceversa). Questo avrebbe dovuto inizialmente convertire la corrente in ingresso in tensione e successivamente binarizzarla. A seguito di alcune modifiche si era poi riusciti ad integrare entrambe le funzioni in una sola configurazione.

Durante i test questo circuito si comportava esattamente come previsto.

A seguito di questi risultati positivi si era quindi deciso di aggiungere, a monte di questo, un circuito sommatore per permettere di valutare tutte le correnti contemporaneamente. Questa scelta, in seguito rivelatasi uno sciocco errore, considerava solo il caso di assenza o presenza di radiazione in ingresso (o meglio: di bassa e alta radiazione). In questa eventualità infatti il circuito si comportava perfettamente, riuscendo a distinguere un messaggio in arrivo in qualsiasi porta. Quando questa condizione non era vera però, si presentavano i problemi.

Non avendo individuato subito a livello logico questo filo conduttore e non avendo i mezzi tecnologici per notarlo, le ricerche proseguivano e, senza averlo an-

cora notato sorse il dubbio che il circuito realizzato non fosse sufficientemente veloce per la commutazione del messaggio.

**II -** Volendo testare dunque se il problema fosse effettivamente del circuito con l'operazionale, venne sviluppata la seconda versione di questo, utilizzando un circuito di cui fosse nota l'effettiva velocità. Questo, conteneva un semplice transistor e una resistenza, e non poteva essere dunque più lento della velocità di commutazione del transistor stesso (che era sufficiente a una comunicazione).

Anche in questo circuito, il problema, non essendo legato alle velocità di commutazione, si ripresentò. A questo punto però, grazie al circuito più semplice, era saltata all'occhio la caratteristica che aveva generato il problema. Gli anodi dei fotodiodi erano infatti, a valle di un diodo ciascuno, collegati insieme. Se questa configurazione non creava particolari problemi nell'eventualità precedentemente descritta, ne creava invece quando i fotodiodi erano sottoposti a radiazioni di valore più alto. Questi infatti, anche se singolarmente non generavano un output in corrente sufficiente a superare la soglia logica impostata, una volta sommati a causa della legge di Kirchhoff, rischiavano di farlo.

**III -** Avendo ora riconosciuto il problema si è quindi potuti procedere a risolverlo. Per farlo si è rivelato necessario fornire ogni fotodiodo di un personale circuito di binarizzazione, il quale sarebbe stato poi connesso a tutti gli altri e al pin di input dei messaggi.

Non essendo possibile, a causa delle dimensioni, inserire un operazionale per ogni fotodiodo, è stato necessario utilizzare un circuito semplificato basato sul transistor (Figura 23). Questo non presenta comunque particolari svantaggi rispetto alla versione basata su operazionale. La logica di funzionamento di questo circuito infatti, a causa anche della presenza di un trimmer atto alla regolazione, non richiede una conversione di elevata precisione e necessariamente isolata quanto la parte di condizionamento dei segnali delle distanze.

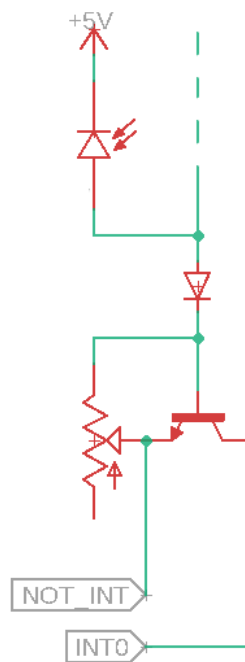


Figura 23: Il circuito di binarizzazione finale. Il trimmer rappresentato permette di variare il valore di tensione, e quindi di corrente, necessario alla messa in conduzione del transistor. Ciò è possibile perché, per entrare in conduzione, il transistor richiede una differenza di potenziale  $V_{BE} > 0.7V$ . *NOT\_INT* permette di attivare/disattivare (rispettivamente 0V e 5V) la ricezione dei messaggi.

## Portabilità verso altre piattaforme

È stato più volte accennato, nel corso di questa tesi, come l'intento del modulo sia di poter essere in qualche modo compatibile ed utilizzabile con diverse schede di prototipazione. Questa capacità non è ovviamente raggiungibile, a causa delle diverse forme, caratteristiche e connessioni, senza la creazione e l'utilizzo di adattatori.

In questo capitolo non ci occuperemo dello sviluppo di questi per una o più schede, impiegheremo invece il nostro tempo nell'affrontare alcuni problemi in cui potrebbe imbattersi un eventuale utilizzatore del modulo che voglia crearne un proprio adattatore.

## Schede con ingressi puramente digitali

Se la maggior parte delle schede disponibili in commercio possiedono già integrati degli ingressi analogici, purtroppo questo non risulta vero per tutte. Prendendo come esempio i single board computer<sup>34</sup>, molti di questi possiedono solo ingressi digitali. Questa caratteristica impedirebbe a questi dispositivi, che potrebbero invece essere particolarmente adatti date le elevate capacità di calcolo, di interagire col modulo. Per permetterne l'utilizzo è quindi necessaria l'implementazione di un modulo adattatore in grado di convertire questi ingressi in entrate digitali.

Per compiti di conversione di questo tipo vengono usualmente utilizzati circuiti integrati denominati ADC (analog-digital converter). Questi hanno la capacità, una volta ricevuto un valore analogico in ingresso, di convertirlo in un valore digitale rappresentato in binario su un certo numero di pin di uscita (Figura 24).

Utilizzando dunque uno di questi dispositivi nel nostro adattatore, non ci troveremo solo un pin occupato, ma bensì tanti quanta sarà la precisione che vorremo ottenere dalla conversione. Facendo un esempio:

### **Esempio: Conversione di un valore analogico**

Supponiamo di aver in ingresso un segnale analogico che possa variare in un range da 0 a 5 volt. Avendo a disposizione un ADC adatto a quel range e 8 pin in output (che possiamo pensare come a bit), otterremo un valore in uscita che potrà assumere 256 valori, ognuno dei quali sarà relativo ad un range di tensioni pari a circa  $20mV$  ( $5V/2^8$ ).

Volendo aumentarne la precisione, l'unico modo che avremo sarà comprare un ADC più preciso o ridurre il range di escursione della tensione. Nel primo caso infatti avremo più bit per quantizzare (i.e., digitalizzare) i valori del range, mentre nel secondo potremo avere lo stesso numero di rappresentazioni ma afferenti a range di minor escursione.

Per completezza, va aggiunto che è in teoria possibile riuscire a rappresentare

---

<sup>34</sup>Come ad esempio la famiglia Raspberry

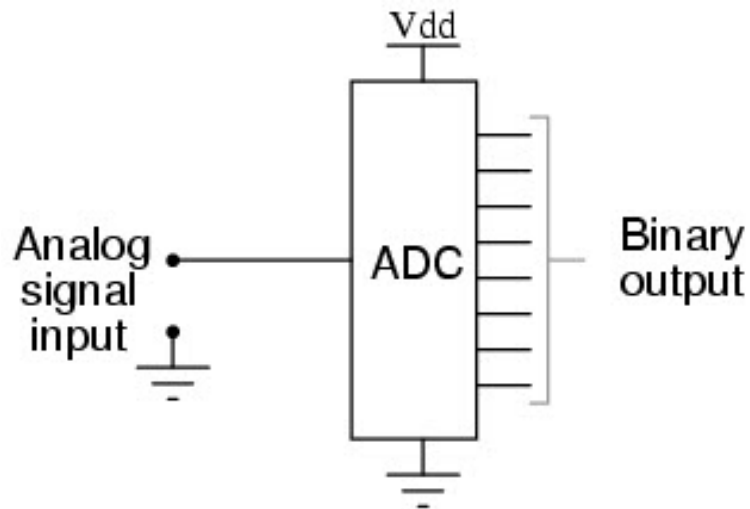


Figura 24: Una rappresentazione di un ADC. Notare come l'output sia formato da più pin, su ognuno dei quali verrà impostato il valore logico della codifica binaria.

su di un solo pin digitale il valore di una grandezza analogica, ma questa tecnica, chiamata in gergo *poor's man ADC*, pecca sia di precisione che di, relativa, lentezza. Questa utilizza infatti il tempo impiegato da un condensatore per caricarsi, permettendo quindi di risalire al valore di tensione analogica a cui era sottoposto.

Data l'evidente imprecisione e il processo leggermente macchinoso e rallentante, si è deciso di fare solo una menzione a questa tecnica, continuando a consigliare i circuiti integrati atti proprio a questo scopo.

### **Schede a differenti tensioni di lavoro**

Nonostante il modulo funzioni a 5V, che è il valore di lavoro probabilmente più diffuso fra microcontroller e single board computer, alcune schede presenti in commercio potrebbero richiedere tensioni di lavoro più basse<sup>35</sup> o, in rari casi, più alte.

<sup>35</sup>Esp e Raspberry ad esempio lavorano a 3.3V

Per le prime, partendo a risolvere il problema dell'alimentazione, è necessario utilizzarne una esterna. È infatti la modalità più semplice per elevare il valore di tensione del **pin di alimentazione** di una scheda.

Un discorso simile è necessario per l'elevazione del livello di tensione di una **uscita (pin) di controllo**. In questo caso è necessario utilizzare un componente chiamato transistor. Questo, nella configurazione che andremo ad utilizzare, funzionerà come un interruttore comandato elettricamente. Quando verrà sottoposto a una differenza di potenziale di almeno  $0.7V$  circa infatti, questo risulterà come un interruttore chiuso, fornendo i  $5V$  dell'alimentazione esterna e risultando come una porta con tale uscita, sufficiente quindi a controllare il modulo.

Per le **uscite PWM** il principio di funzionamento resta il medesimo.

Esistono anche altri dispositivi che permettono di eseguire l'operazione che è stata appena descritta (come ad esempio i fotoaccoppiatori), ma il transistor rappresenta sicuramente, in questo caso, la soluzione più economica, semplice e funzionale.

Avendo descritto come aumentare il livello logico in uscita da una porta, possiamo ora passare a spiegare come regolare correttamente quello in ingresso. In questo caso, per gli **ingressi digitali**, è sufficiente inserire una resistenza di valore tale da creare un calo di tensione pari alla differenza fra il valore fornito e quello richiesto. Se ad esempio la nostra porta richiede un ingresso a  $3.3V$  e quello che abbiamo a disposizione è a  $5V$ , basterà inserire una resistenza che provochi una caduta di un paio di volt per ottenere il segnale adatto.

Se invece il nostro segnale è analogico, anziché inserire una sola resistenza che crei una caduta fissa, dovremo inserire un partitore di tensione. Come il nome lascia intuire, si tratta di un circuito con la capacità di "dividere" una tensione per un coefficiente costante, dato dai valori di due resistenze (che compongono il partitore). Scegliendo opportunamente questi valori quindi, è possibile ottenere un circuito che "mappi" i valori dal range  $0 - 5V$  a quello  $0 - 3.3V$  (o altri, se necessario). In questo modo è possibile convertire un segnale in modo che sia quindi leggibile tramite un **pin analogico** lavorante ad un range inferiore .



Per la seconda tipologia di schede invece, il problema è facilmente risolvibile, sia per i **pin di alimentazione** che per quelli di **output**, inserendo resistenze che abbassino i valori di tensione forniti dalla scheda. Per i pin di **input analogici** il problema non si pone proprio. Potendo infatti questi leggere anche valori minori di tensione, basterà rimappare questi a livello software. In queste tipologie di schede quindi, gli unici pin che necessitano di un circuito relativamente più complesso, sono gli **input digitali**. Infatti, se questi ricevessero dal modulo 5V ed avessero una tensione di soglia logica di, ad esempio, 7V, non rileverebbero il livello logico alto. Per questi pin è quindi necessario un componente che permetta di innalzare il livello logico. Ciò è ottenuto, come abbiamo già potuto vedere, tramite l'utilizzo di transistor.

## **Risultati e possibili miglioramenti**

Lo sviluppo di questo modulo ha riscontrato risultati tutto sommato positivi. Anche se tutti i requisiti sono stati rispettati però, questo non vuol dire che la sua versione finale sia perfetta. Sono infatti sicuramente molti i possibili punti di miglioramento sui quali non è stato possibile lavorare a causa della mancanza di tempo. Non è stato neanche ovviamente possibile applicare alcune idee "dell'ultimo minuto".

In questo paragrafo, si accennerà quindi velocemente a queste proposte di miglioramento, cercando di spiegare perché una determinata modifica possa risultare tale.

**I -** Il primo upgrade hardware proposto è relativo all'introduzione di trimmer di regolazione allo scopo di eliminare, o quantomeno ridurre, le **differenze di uscita in corrente dei vari fotodiodi**. Come verrà infatti descritto anche nel capitolo relativo alla libreria software, questi hanno, nella versione finale del progetto, esposto alcune divergenze nei valori delle correnti in uscita quando sottoposti al medesimo input. Allo scopo di evitare un aggiustamento software quindi, la scelta migliore ricade sull'omogeneizzazione di questi valori tramite componentistica hardware.

**II -** Il secondo miglioramento riguarda invece un sostanziale **cambiamento nell'architettura** del modulo. Come infatti descritto nel capitolo relativo alle implementazioni proposte, l'utilizzo di un controllore separato si sarebbe rivelato, a seguito degli ottimi risultati dell'alimentazione, una scelta incredibilmente valida. Il collegamento del modulo tramite  $I^2C$  rappresenterebbe infatti un valido miglioramento rispetto alla soluzione attuale, non richiedendo neanche più una libreria a parte per l'utilizzo. Questa soluzione si propone quindi come possibile miglioramento, anche se rappresenta effettivamente una semplice variante del progetto sviluppato.

### **3 Sviluppo di un applicativo di controllo**

Come introdotto negli obiettivi di progetto, parte necessaria della tesi è rappresentata dall'implementazione di un applicativo di controllo. In questo capitolo, ci si occuperà proprio di discutere dei requisiti e delle caratteristiche di questo, valutando le scelte implementative più dal punto di vista dell'utilizzo dell'applicativo che dell'implementazione software vera e propria.

#### **Motivi dello sviluppo**

Nell'introduzione alla tesi, abbiamo accennato di come l'interazione con il dispositivo, magari sotto test, sia, dal nostro punto di vista, un elemento di forza nell'ambito dello sviluppo di un progetto.

È infatti vero come la parte spesso maggiore dello sviluppo software non sia relativa alla scrittura, quanto invece al debug dello stesso. Quando si parla di piattaforme che utilizzano componentistica hardware poi, questo impiego di tempo diventa addirittura maggiore, mancando spesso soluzioni che permettano di capire efficacemente i problemi del dispositivo.

È oltretutto vero come in alcuni progetti, questa interazione possa permettere un avanzamento nella complessità e funzionalità del software prodotto. Le possibilità di gestione e controllo remoto infatti, possono permettere uno sviluppo di applicativi con maggior flessibilità nella gestione e risoluzione dei problemi di cui si occupa il sistema.

Con l'obiettivo di fornire queste funzionalità quindi, si delinea l'idea dello sviluppo di un software che possa dialogare col dispositivo.

#### **Requisiti di progetto**

**I -** Come primo punto della trattazione, è necessario specificare quali tecnologie di comunicazione quest'applicativo debba supportare. Come introdotto

nell'analisi iniziale, quelle che sono risultate maggiormente valide agli obiettivi del progetto sono state **Bluetooth** e **WiFi**.

L'applicativo dovrà dunque fornire supporto all'utilizzo di entrambe queste tecnologie.

**II** - Assolvendo agli obiettivi di ideazione, l'applicativo dovrà permettere un'analisi degli eventi che avvengono all'interno del robot. Questo dovrà avvenire tramite una schermata in cui sia possibile **visualizzare i messaggi** (o log) inviati dal dispositivo.

**III** - A seguito della volontà di estendere le capacità della piattaforma, l'applicativo dovrà fornire un metodo di **comunicazione bidirezionale** col robot. Questo dovrà essere sufficientemente generico da permettere utilizzi di diverso tipo ma, allo stesso tempo, abbastanza **semplice** da risultare comodo nell'uso.

**IV** - Infine, l'applicativo dovrà risultare **accessibile** (i.e., utilizzabile) al maggior numero di persone possibili.

## **Scelta della piattaforma di sviluppo**

Una parte importante della trattazione, riguarda i dispositivi per i quali l'applicativo debba essere sviluppato. Non è infatti possibile, nell'ambito di una tesi triennale con un tale monte ore già impiegato, lo sviluppo di più applicativi per sistemi diversi. È quindi necessario operare una scelta riguardante la tipologia di applicazione da realizzare.

Le proposte che sono state prese in considerazione sono state:

- Android
- IOS
- Computer application
- Web service

## **Android**

Partendo dalla prima di queste, uno sviluppo per sistemi Android andrebbe a creare la possibilità di utilizzo dell'applicativo da parte dell'88% dei dispositivi cellulari al mondo<sup>36</sup>.

Oltre a questa caratteristica, la maggior parte degli smartphone moderni supporta sia la tecnologia Bluetooth che quella WiFi, permettendo quindi una vasta possibilità di utilizzo dello stesso.

## **IOS**

Passando invece alla valutazione dei sistemi IOS, seppur anch'essi supportanti entrambe le tecnologie, hanno il loro svantaggio nella complementarità di mercato rispetto ad Android. Questi rappresentano infatti la percentuale di smartphone rimanente, molto minore rispetto al concorrente. A questo svantaggio si aggiunge il costo di questi, non permettente un acquisto atto al solo sviluppo che, ricordiamo essere, di intento didattico.

## **Computer application**

Analizzando un possibile sviluppo dell'applicativo per computer, questo richiederebbe la possibilità di utilizzo su diversi sistemi operativi.

Sarebbe inoltre necessario, a seguito delle tecnologie comunicative scelte, che il computer posseda la capacità di dialogare sia tramite Bluetooth che tramite WiFi.

Se quest'ultimo non è al giorno d'oggi un problema, il primo è invece ancora mancante in un buon numero di PC in commercio. Questa limitazione richiederebbe quindi di fornirsi di un dispositivo Bluetooth esterno che permetta la comunicazione col device.

## **Web service**

Per quanto riguarda infine il servizio Web, questo si rivelerebbe sicuramente

---

<sup>36</sup>Dati relativi al Q2 2018

valido nella casistica in cui il robot agisca come device IoT. Nella sua configurazione Bluetooth invece, si sentirebbe la mancanza di una metodologia di connessione al dispositivo, limitandone le capacità d'uso e non assolvendo alle specifiche tecnologiche precedentemente scelte.

### **Selezione finale**

A seguito delle precedenti analisi, si ritiene che l'applicativo fornisca le migliori possibilità nella sua **versione Android**. Quest'ultima è per cui scelta per l'implementazione.

### **Definizione dell'esperienza utente**

Come risaputo, punto focale del successo di un applicativo mobile è la sua esperienza utente. Concentriamoci dunque sulla sua definizione e sulle scelte che sono state fatte a riguardo.

### **Schermata principale**

L'applicativo si apre mostrando quella che è la sua schermata principale. Potendo questo aver a che fare con più dispositivi di tipi diversi, si è scelto di visualizzare, in questa schermata, quelli associati (i.e., quelli con cui è possibile comunicare), in liste separate (Figura 25).

Se per qualche motivo, le due tecnologie non fossero presenti nel dispositivo (o fossero spente), la lista corrispondente verrebbe nascosta.

Questa schermata presenta poi una possibilità di visualizzazione che migliora notevolmente l'esperienza utente.

Non si è infatti tralasciato il caso in cui all'applicativo possa essere associato un vasto numero di dispositivi. In questo caso, viene fornita la possibilità di creare una lista di visualizzazione ristretta. I dispositivi selezionati vengono aggiunti a una lista di preferiti che è poi successivamente possibile visualizzare

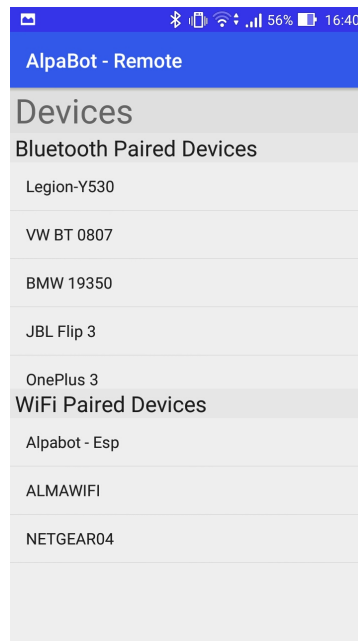


Figura 25: La schermata principale.

(Figura 26). È comunque ovviamente possibile rimuovere questi dispositivi dalla lista.

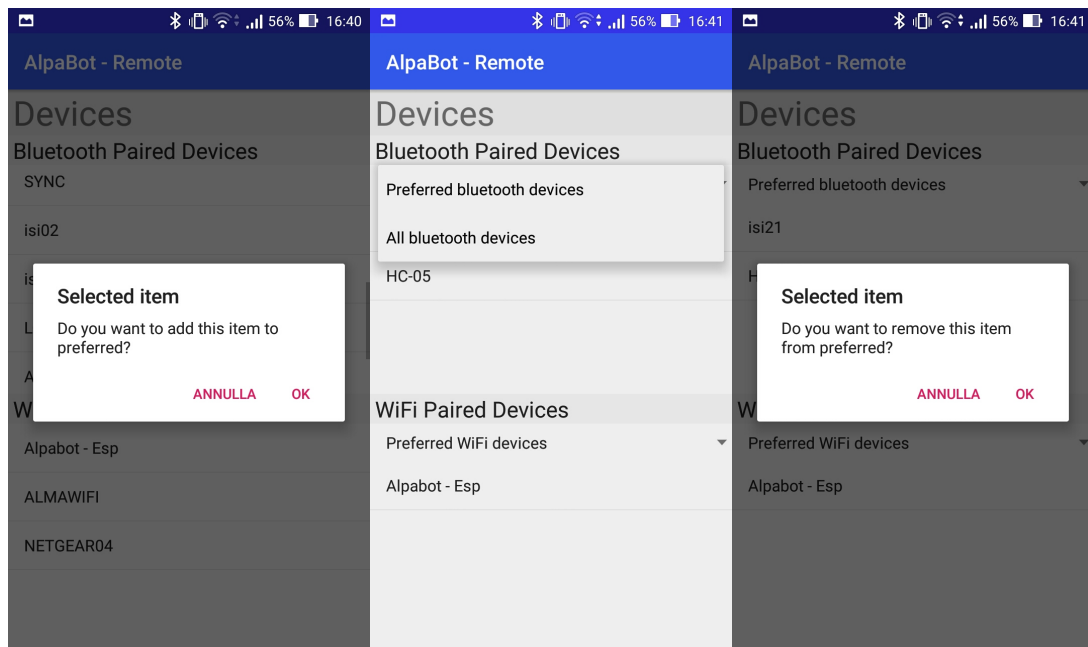


Figura 26: La gestione dei preferiti: aggiunta, visualizzazione e rimozione.

## Schermata dispositivo

Questa schermata, apribile cliccando sul corrispondente dispositivo nella lista, è il fulcro dell'applicazione. È da qui infatti possibile aprire una connessione col robot e, una volta collegati, utilizzare tutte le funzionalità dell'applicazione, disponibili sotto forma di Fragment<sup>37</sup> (Figura 27)

Questi ultimi, escludendo la schermata iniziale che informa della mancanza di connessione, possono essere:

- Terminal
- Chart
- Control

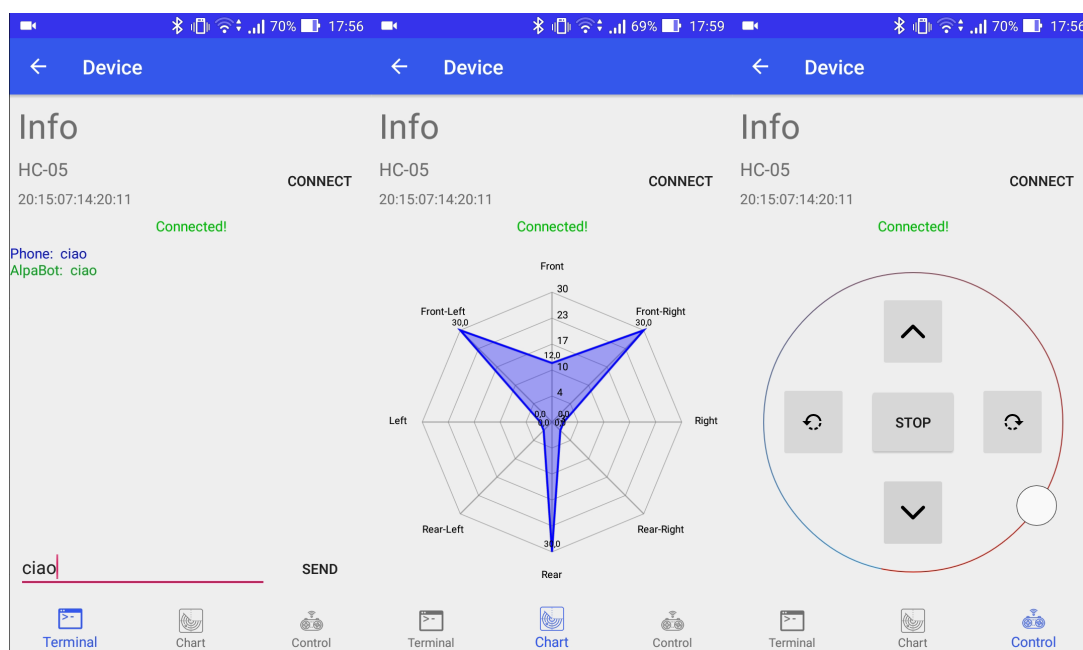


Figura 27: I tre fragment. Da sinistra: Terminal, Chart e Control.

Se si è connessi al dispositivo, lo switch fra le varie funzionalità è possibile tramite una navbar posizionata sul lato inferiore della schermata.

<sup>37</sup>Un componente di Android. È definibile come una "porzione" di Activity. Se quest'ultima può esistere senza un fragment, non è vero il contrario.



**Il terminal** è il fragment che permette la visualizzazione dei log e l'invio dei messaggi. È composto da una "finestra" dove viene visualizzato il testo in arrivo dal dispositivo e quello inviato dallo smartphone. Per evitare stampe errate, il testo da visualizzare deve essere in un apposito formato JSON. I messaggi che non rispettano questa prassi non possono essere visualizzati. Per quanto riguarda l'output invece, viene formulato in una "casella di scrittura" e successivamente inviato tramite un bottone.

**Il chart** è il fragment preposto alla visualizzazione delle distanze lette dal dispositivo. Essendo questa una funzione di default del robot (anche se rimovibile) ed essendo di difficile lettura da un terminale, l'implementazione di questo fragment è stata ritenuta un ottimo soddisfacimento della caratteristica di semplicità d'uso.

La visualizzazione in questo fragment avviene solo a seguito della ricezione di un appositamente formattato JSON.

**Il control** infine è un'utility che si è ritenuto poter essere molto comoda nell'utilizzo dell'applicativo. Si è infatti supposto come possa, in alcune implementazioni, diventare molto comune il comando a distanza del robot.

Con in mente quest'evenienza quindi, si è proceduto all'implementazione di un joystick che permettesse il controllo dei movimenti più comuni. Fra questi compare ovviamente la rotazione, per la quale è possibile impostare l'angolo di cui girare.

I comandi forniti da questa finestra, sempre in formato JSON, hanno una differente formattazione rispetto ai messaggi inviati dal terminale.

## **Comunicazione col dispositivo**

Come già introdotto, le tecnologie di comunicazione col dispositivo che si sono scelte sono state Bluetooth e WiFi. Entrambe utilizzano un approccio orientato alla connessione, comprendente la creazione di socket.

## Apertura della connessione

Ovviamente, il primo passo per l'apertura di un canale di comunicazione consiste nella procedura di pairing col dispositivo. Quest'azione viene effettuata, per entrambe le tipologie di connessione, nei menù dedicati del sistema operativo. Si è infatti ritenuto inutile includere questa funzionalità direttamente nell'applicazione, dato che non avrebbe aggiunto nessun vantaggio al successivo utilizzo del dispositivo.

Una volta che l'associazione sarà andata a buon fine, sarà necessario, ad ogni apertura della connessione, l'istanziamento di una socket. Questo processo, essendo differente per ognuna delle tecnologie, non potrà essere eseguito da una generica classe, ma dovrà essere affidato ad una atta allo scopo (o quanto meno ad un metodo specializzato).

Nella pratica, l'idea iniziale era stata l'utilizzo di un Template Method<sup>38</sup>, ma notando come questo dovesse agire in modo asincrono, si è preferito delegare direttamente il compito ad un AsyncTask definito esternamente alla classe di connessione.

## Scambio di messaggi

Queste socket però, nonostante condividano un buon numero di metodi, non risultano intercambiabili o afferenti a qualche comune interfaccia. Sono invero accomunate dalla sola implementazione di *Closeable*, che non contiene però i metodi utilizzati nei classici canali di comunicazione.

Allo scopo di permettere all'applicativo di sfruttare intercambiabilmente entrambe le tipologie di socket, è stata quindi necessaria la creazione di un'interfaccia comune e di due classi ad essa afferenti, le quali ricoprono dunque lo scopo di wrapper.

Avendo ora a disposizione un metodo per utilizzare indifferentemente entrambe le socket, si è provveduto alla creazione di un'apposita classe che permettesse la comunicazione (i.e., *CommunicationChannelImpl*; Figura 28). Questa, con-

---

<sup>38</sup>Un pattern comportamentale della Gang of Four

tenente un thread per la ricezione dei messaggi in ingresso, si occupa poi della loro distribuzione tramite l'utilizzo di Intent e broadcast.

Anche l'invio dei dati richiede però un approccio asincrono. Il tempo che può venire impiegato in quest'azione può infatti, se eseguito nel main thread, bloccare l'applicazione. A questo scopo, è stato utilizzato un apposito componente di Android chiamato AsyncTask.

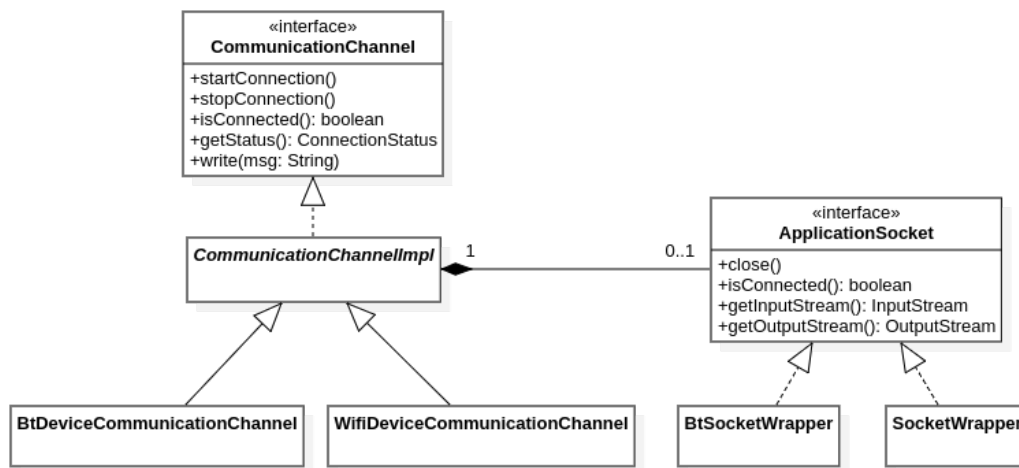


Figura 28: Lo schema UML delle classi e interfacce che concorrono nella comunicazione con AlpaBot.

### Formato dei messaggi

Fin dalla decisione di permettere al dispositivo di poter comunicare e ricevere comandi, si è presentata la necessità di far in modo che questi potessero essere distinguibili. Ogni tipologia di dato potrebbe richiedere infatti un'analisi unica, che non può venir gestita nel modo sbagliato. Con quest'idea in mente si è cercato quindi di assegnare ad ogni classe dell'applicazione la gestione di un particolare tipo di dato, facendo in modo che quelli non supportati dalla stessa venissero ignorati, così da non creare problemi.

Questa suddivisione, oltretutto ben visibile nella gestione dei dati nei tre fragment, ha però bisogno di un valido metodo di identificazione. Questo dev'essere pulito (comprensibile) ma, cosa ancor più importante, utilizzabile facilmente da

altre piattaforme. Pensando infatti all'evenienza dell'ampliamento del progetto, sarebbe stupido creare un sistema di comunicazione ad hoc che ne richieda la re-implementazione in ogni nuova piattaforma. Risulta così molto più valida la definizione di un formato standard di comunicazione tra il dispositivo e l'applicazione.

Questo formato è stato invero strutturato ed adottato all'interno dell'applicativo (Listing 1). Basato su JSON, contiene alcuni elementi statici e altri variabili. La componente fondamentale della struttura è la definizione del **tipo di dato** (messaggio, comando, dati, etc.; all'interno del campo *type*), che permette, come già detto, di gestirlo dall'opportuno operatore.

A seguito del tipo, si trova il **valore/i** (campo *val(s)*), nel quale è presente il contenuto del messaggio. Questo, può assumere il significato di un testo vero e proprio, oppure rappresentare un comando da eseguire o un insieme di valori da visualizzare. Questa possibilità è ovviamente dipendente dal tipo di messaggio inviato.

Ad eventuale seguito del valore si possono quindi trovare dei **parametri opzionali**. Questi possono essere, ad esempio, valori legati al comando o specifiche su come eseguirlo.

```
1   {
2       "type": "cmd",
3       "val": "move",
4       "par": {
5           "dir": "right",
6           "deg": "15"
7       }
8   }
```

Listing 1: Un esempio di comando. Si possono notare i parametri specificanti senso (right, i.e. orario) e angolo della rotazione.

Una struttura di questo tipo, oltre che mantenere una certa flessibilità tramite l'invio, dall'applicazione mobile, di comandi o dati custom all'interno del tipo messaggio, permette una molto più agevole scomposizione dei compiti

dei gestori. Ognuno infatti si occuperà di gestire un determinato tipo di dato, portando in primo luogo alla semplificazione del codice, ed in secondo luogo all'eliminazione di errori derivanti da tipi di messaggio non supportato. In quest'evenienza infatti, il messaggio non verrebbe semplicemente gestito da nessun handler.

Infine, questa caratteristica può tornare molto utile nel caso dell'implementazione di applicativi con scopi diversi.

#### **Esempio: Sistema di acquisizione dati**

Supponiamo infatti di voler collegare il dispositivo ad un sistema di acquisizione dati e alternativamente all'applicativo mobile. In questo caso potremmo voler vedere su uno di essi determinati dati, e sull'altro altri (o magari, più semplicemente, gli stessi dati resi in maniera più leggibile o diversa). In questo caso potremo creare un tipo di dato ad hoc che sarà quindi gestito da uno dei servizi ed ignorato dall'altro.

## **Risultati e possibili miglioramenti**

I risultati ottenuti dallo sviluppo dell'applicativo descritto in questo capitolo, si sono rivelati veramente apprezzabili. Questo infatti non rispetta soltanto tutti i requisiti richiesti, ma fornisce anche un'interfaccia grafica gradevole che riesce a mantenere una certa funzionalità e semplicità.

I possibili miglioramenti qui descritti non saranno quindi relativi all'applicativo in se, che già allo stato attuale genera sufficiente soddisfazione, ma a una più ampia visione di *gestione remota*.

Un miglioramento proposto è infatti concernente l'implementazione di una possibilità di controllo remoto della piattaforma robotica da computer. Durante i test infatti, nonostante l'applicativo si sia rivelato sufficientemente comodo, si è sentita la mancanza di un'interfaccia di questo tipo.

## 4 Implementazione di una libreria software

Abbiamo introdotto fin dall'inizio la volontà di creare delle librerie per l'utilizzo pratico della piattaforma. In questo capitolo discuteremo delle caratteristiche di queste e delle soluzioni implementate per permettere, tramite il loro uso, di migliorare l'efficienza del robot e dei suoi componenti.

### Motivi dello sviluppo

Fin dall'inizio, questo progetto nasce con l'intento di permettere agli studenti di corsi robotici di valutare il funzionamento delle proprie applicazioni su componenti hardware. Come già detto, questa possibilità risulta importante al fine di permettere di ampliare la propria capacità di individuare errori che sarebbero indistinguibili in uno sviluppo simulato.

Questa caratteristica non dev'essere però confusa con l'idea che una persona debba affrontare le stesse problematiche in ogni nuovo progetto. Una volta che di queste si è acquisita coscienza, risulterebbe un grande spreco di tempo il dovervi lavorare sopra ad ogni nuova applicazione sviluppata. Allo scopo di ridurre l'*effort* in relativamente inutili operazioni, nasce quindi l'idea delle librerie.

Oltre a questo obiettivo principale, alcuni componenti della piattaforma potrebbero risultare abbastanza complicati nell'utilizzo senza aver a disposizione una valida interfaccia d'uso.

È questo il caso ad esempio del modulo infrarosso. Col progredire dello sviluppo e con l'aggiunta della possibilità di scelta della configurazione infatti, il suo utilizzo è diventato via via più complesso, risultando ora abbastanza complicato nell'uso senza l'ausilio della libreria.

L'utilizzo di questa fornisce infine la possibilità di collegare logicamente insieme tutti i componenti del robot.

Se nel nostro progetto questa caratteristica potrebbe non risultare in un vero vantaggio, potrebbe però farlo a seguito dell'aggiunta o creazione di altri mo-

duli. Una libreria sfruttante tutte le potenzialità del robot infatti, permetterebbe di svolgere automaticamente operazioni più complicate.

#### **Esempio: Guida autonoma**

Supponiamo, come esempio, di aggiungere al robot un modulo di posizionamento spaziale. Con una libreria capace di sfruttare in combinazione questo modulo con quello motori, si potrebbe semplicemente comandare al robot di raggiungere una determinata posizione, concentrandosi quindi sull'implementazione di altre funzionalità.

## **Requisiti di progetto**

Come per il modulo e l'applicazione mobile, anche la libreria deve rispettare alcuni vincoli. Essendo questa un sistema legato all'ingegneria del software, possiamo distinguerne requisiti funzionali e non.

### **Requisiti non funzionali**

Dall'ingegneria del software, sappiamo come questi descrivano i *vincoli* ai servizi o alle funzioni offerte dal sistema. La loro profilazione, in questo caso, dipende più che altro dalle caratteristiche di alcuni microprocessori su cui la libreria può dover essere eseguita.

Allo scopo iniziale di estendere la versatilità al maggior numero di piattaforme, il primo requisito si rivela l'utilizzo del framework Arduino.

Questo è infatti disponibile per una vasta quantità di schede di prototipazione e risulta utilizzabile anche su alcuni single board computer. Il suo utilizzo fa quindi in modo che la libreria sia importabile in tutti i progetti basati su di esso.

A causa della memoria spesso ridotta di molti microcontrollori, si presenta come secondo pressante requisito la necessità di utilizzarne la minor quantità possibile. Se questa fosse infatti per la maggior parte occupata dalla libreria, ne risulterebbero limitate le potenzialità del progetto, che avrebbe meno spazio

a disposizione per il codice componente l'applicativo. Questa carenza causerebbe quindi un'implementazione obbligatoriamente basilare allo scopo di risparmiare quanta più memoria possibile.

Oltre a questo, è importante che la libreria garantisca una gestione il più possibile semplice ed intuitiva del dispositivo, alla quale si andrà comunque ad affiancare la necessità di mantenere una possibilmente elevata quantità di operazioni eseguibili.

A seguito di quest'ultima considerazione, va sottolineato come queste funzionalità messe a disposizione della libreria, allo scopo di semplificare la modularizzazione, debbano essere suddivise in base alla componente del robot a cui afferiscono. Questa partizione permetterebbe quindi a queste sottoparti di poter essere utilizzate indipendentemente l'una dall'altra.

### **Requisiti funzionali**

In opposizione ai requisiti non funzionali troviamo invece quelli funzionali. Questi, anziché i vincoli, descrivono proprio le funzioni e i servizi che il sistema deve mettere a disposizione.

Nel caso della libreria software in esame, la scelta di questi dovrà tenere in considerazione le possibilità di calcolo dei dispositivi su cui verrà eseguita. Nonostante sia infatti auspicabile la possibilità di scegliere questi requisiti valutando il solo interesse dell'utilizzatore, è d'altro canto vero che le caratteristiche ridotte dei sistemi di calcolo in esame non potrebbero sopportare adeguatamente richieste eccessivamente intense.

A questo scopo si è quindi dovuta effettuare una oculata scelta nella selezione dei requisiti funzionali.

Abbiamo parlato, durante la definizione dei requisiti non funzionali, della necessità di suddividere la libreria in sottoparti. A seguito di questa partizione, si presenta la necessità di esprimere le funzionalità del sistema non più a livello della libreria, quanto piuttosto a livello di questi suoi sottomoduli. Sono quindi a seguito elencate le partizioni che si intende effettuare e, in fine, i requisiti di ognuna di esse.



I sottomoduli della libreria saranno quindi:

- comunicazione
- modulo infrarosso/ultrasonico
- motori

**Il modulo comunicazione** ha il compito di fornire utilità relative allo scambio di messaggi. Queste dovranno almeno essere:

- invio messaggi e distanze
- formattazione automatica del json in uscita
- ricezione json in ingresso
- decodifica del json in arrivo
- individuazione del tipo di messaggio, comando e movimento
- restituzione dell'angolo di rotazione
- disattivazione centralizzata di tutte le stampe di "log" seriali<sup>39</sup>

**Il modulo IRUSS** (i.e., infrarosso e ultrasonico) si occupa di tutte le operazioni riguardanti quest'ultimo. Questa sottoparte della libreria è a sua volta suddivisa per ognuna delle possibili configurazioni.

Le operazioni minime richieste, complessivamente, per questo sottomodulo saranno:

- selezione del software a supporto della configurazione scelta
- ottenimento delle distanze
- comunicazione infrarossa (se supportata dalla configurazione)

---

<sup>39</sup>È stato richiesto in quanto i log sono spesso utili durante il solo debug e non, invece, durante la normale esecuzione di un programma

**Il modulo motori** controlla invece tutto ciò legato a questi ultimi e al movimento in generale.

Le operazioni di cui si richiede la disponibilità sono dunque:

- movimento in avanti o indietro
- rotazione ad un preciso angolo
- selezione della velocità di movimento

## **Complicazioni software affrontate**

Non essendo ovviamente utile o possibile descrivere la libreria nel suo complesso, in questo paragrafo ci limiteremo a discutere delle problematiche di funzionamento che si sono presentate durante lo sviluppo della libreria e di come queste siano state risolte via software. Nello specifico, i problemi riscontrati che hanno necessitato un'implementazione relativamente complessa sono stati:

- variazione della luminosità ambientale
- conversione della luminosità in distanze
- differenza dei valori letti dai diversi fotodiodi

### **Variazione della luminosità ambientale**

All'inizio di questo progetto, non si era colto quanto effettivamente la radiazione ambientale potesse influire sul calcolo delle distanze da parte del dispositivo. Si era ovviamente fin da subito saputo di alcune limitazioni dipendenti dalla tecnologia in sé (e.g., i problemi di utilizzo sotto luce diretta del sole), ma non si era totalmente compreso quanto una variazione, seppur sempre omogenea, nella luminosità dell'ambiente, potesse variare la lettura. Queste discrepanze nelle rilevazioni influivano quindi nel calcolo delle distanze, rendendolo particolarmente impreciso.

Allo scopo di rimuovere o limitare queste variazioni, sono seguiti un considerevole numero di test. Di questi, il più interessante è stato relativo alla lettura delle distanze in diverse condizioni di luminosità.

In questo caso, le semplici letture di valori seguivano, come naturale, un andamento crescente all'aumentare della distanza<sup>40</sup> ma, a seconda della luminosità ambientale, presentavano valori molto distanti le une dalle altre.

Durante il test si è quindi valutata la possibilità, a causa di questa proprietà, di sottrarre al livello della radiazione ambientale la radiazione seguente all'impulso. Sono quindi state effettuate due rilevazioni (una iniziale senza aver inviato nessun segnale e una alla ricezione dello stesso) e sono successivamente state sottratte. Il risultato nell'utilizzo di questa tecnica si è effettivamente rivelato molto migliore rispetto alla rilevazione iniziale. Il grafico risultante dalla sottrazione è, infatti, avente curve molto più vicine e con andamenti molto più simili rispetto a quello iniziale (Figura 29).

### **Conversione della luminosità in distanze**

Come abbiamo potuto discutere nel paragrafo precedente, con la sottrazione delle radiazioni è possibile ottenere curve con andamenti, al variare della luminosità, più simili fra loro. Se questo permette di migliorare il calcolo delle distanze rispetto alla versione iniziale, non basta però per garantire una precisione sufficientemente buona.

La risoluzione di questo problema ha dato non poco da riflettere, è ha richiesto lo studio di non poche possibilità di soluzione. Fra quelle analizzate però, quella che ha presentato i migliori risultati si è rivelata essere l'approssimazione di un andamento con una funzione.

Questa tecnica punta all'ottenimento di una funzione che approssimi al meglio un insieme di valori. Una volta ottenuta questa, è quindi possibile ricavarne l'inversa ed utilizzarla per ottenere, una volta passategli le intensità luminose, le corrispondenti distanze.

---

<sup>40</sup>Questo avviene a causa della configurazione di funzionamento dell'operazionale che, all'aumentare della radiazione riflessa, riduce il valore all'ingresso del pin analogico

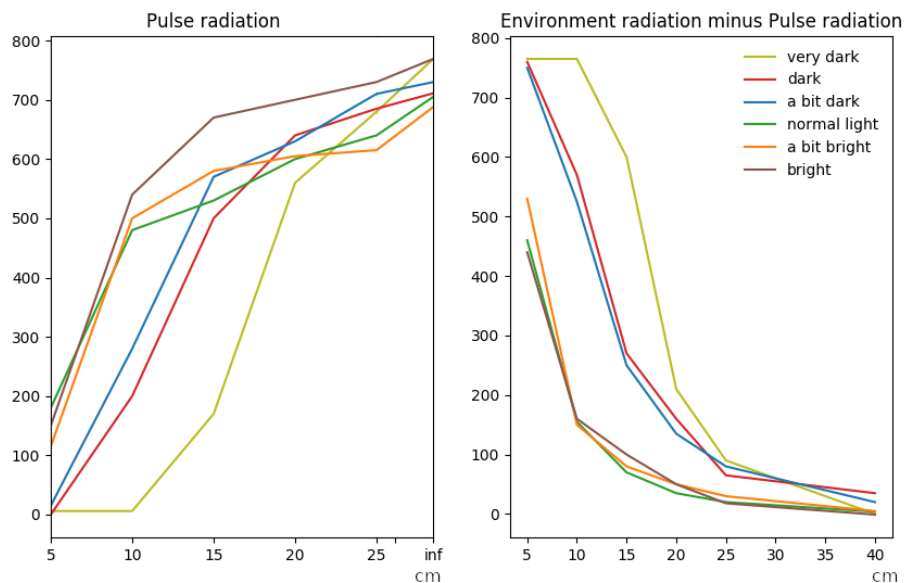


Figura 29: Relativamente alle varie luminosità ambientali, si possono notare:

- a sinistra la radiazione in ingresso a seguito dell'impulso,
- a destra la differenza fra le due radiazioni

L'asse Y rappresenta il valore letto sul pin analogico

Sfortunatamente però, molte di queste funzioni non si adattavano al tipo di progetto. Quelle lineari non presentavano risultati sufficientemente precisi mentre alcune non lineari risultavano troppo onerose per quanto riguarda la complessità di calcolo.

Le uniche tipologie che sono state considerate valide sono state alcune non lineari particolarmente semplici e le *Spline*.

Le prime possono fornire, a seconda della funzione, risultati anche molto buoni, ma hanno lo svantaggio di richiedere, per il loro "fitting", software dedicati (e.g., matlab), rendendo il compito di setup della piattaforma leggermente più macchinoso.

Le *Spline* d'altro canto sono funzioni costituite da un insieme di polinomi raccordati tra loro. Il loro scopo è la connessione di un insieme di nodi in modo tale da avere un andamento continuo in ogni punto dell'intervallo (almeno fino

ad un dato ordine di derivate).

Questa caratteristica si traduce, nel probabile caso di utilizzo di una libreria, in una loro facile definizione<sup>41</sup>. In questo caso infatti, il passaggio dell'insieme dei valori campionati è sufficiente per la creazione della curva. Questa facilità di istanziamento si adatta quindi molto bene agli scopi del progetto.

Queste curve, oltre ai sopraccitati punti a favore, hanno però anche aspetti negativi. Fra questi, il più pressante riguarda il costo computazionale di calcolo che, per un microcontrollore, può risultare troppo oneroso.

A questo proposito si è quindi deciso di utilizzare una particolare tipologia di Spline chiamata *Catmull-Rom*. Questa tipologia di curva infatti, al costo dell'aggiunta di due punti<sup>42</sup>, risulta molto più leggera da calcolare, e quindi valida agli obiettivi del progetto.

### **Differenza dei valori letti dai diversi fotodiodi**

Uno degli ultimi problemi incontrati durante l'utilizzo del modulo è derivato dagli stessi componenti in esso utilizzati. È noto infatti come i dispositivi hardware, a causa dei processi produttivi, presentino sempre leggere differenze gli uni dagli altri.

In questo caso, il problema onestamente inaspettato è stato il notevole divario fra i valori letti nei vari fotodiodi. Ciò è stato rilevato notando come questi fornissero diversi output anche quando sottoposti a medesimi input.

La causa di ciò, che non è stata però verificata, ritengo sia relativa alle piccole correnti in gioco in questi dispositivi. Minime differenze di realizzazione temo abbiano infatti portato al notevole variare di queste.

A prescindere dalla causa però, questa problematica, la cui proposta di soluzione hardware è stata trattata nel capitolo relativo ai possibili miglioramenti del modulo, ha presentato problemi in quanto notificata solo nella versione finale del progetto. Nelle versioni di test precedenti infatti, questi dispositivi avevano, forse per pura fortuna, fornito output molto simili.

---

<sup>41</sup>Nel caso del progetto è stata ad esempio utilizzata la libreria Arduino-Splines[9]

<sup>42</sup>Questi vengono aggiunti all'inizio e alla fine della curva, non essendo però considerati facenti parte della stessa, allo scopo di fornirne l'inclinazione iniziale

Essendo ad ogni modo il progetto hardware ormai completato, non è stato possibile modificarne il circuito. Questa limitazione ha generato quindi la necessità di porvi rimedio attraverso correzioni software.

Considerando quindi come i valori rilevati dai fotodiodi siano diversi ma comunque seguenti lo stesso andamento, la soluzione a questo problema si è trovata, abbastanza semplicemente, nell'assegnare ad ogni fotodiodo un moltiplicatore. Una volta dimensionato correttamente questo, è poi possibile moltiplicarlo per il valore letto in ingresso per ottenere, tramite la Spline, la distanza corretta. Il lato negativo dell'aggiustamento via software, in confronto a quello hardware, consiste nella possibilità che la corrente, se troppo piccola o troppo grande rispetto al range di calibrazione, produca una lettura "tagliata" dall'operazionale.

## **Risultati e possibili miglioramenti**

I risultati relativi all'implementazione di questa libreria sono stati in grado di soddisfare i requisiti richiesti. Alcuni di questi potrebbero essere però notevolmente perfezionati.

Al momento, la libreria per la comunicazione, anche se pienamente in grado di lavorare con i messaggi, deficiata di un mancato **setup automatico** della connessione WiFi. È infatti al momento disponibile la sola istanziazione automatica della connessione Bluetooth. In un possibile futuro miglioramento della libreria, sarà quindi ovviamente necessario fornire questa possibilità.

Altro punto sicuramente migliorabile si trova relativamente all'**utilizzo della memoria**. La libreria infatti, nonostante ne lasci a sufficienza per eseguire applicazioni anche sufficientemente complesse, occupa ancora un po' troppo spazio. I punti su cui si può lavorare per il miglioramento di questa caratteristica sono sicuramente più di uno.

Il primo di questi consiste nell'utilizzo che si fa della libreria ArduinoJson (reference relativa al sito web del progetto [10]). Questa infatti occupa da sola un buon terzo della memoria occupata dall'intera libreria. Anche perdendo qualche funzionalità quindi, una sua sostituzione con una versione più leggera

migliorerebbe di molto lo spazio occupato. Il secondo punto è invece relativo all'utilizzo delle curve Spline nell'applicativo. Al momento infatti, a causa di una questione di tempo, è presente all'interno della libreria una mappa contenente, per ogni range di radiazione ambientale, la curva ad esso più vicina. Questa soluzione è stata necessaria in quanto le curve passano per punti leggermente differenti in base ai vari valori di radiazione ambientale (come visibile in figura 29). Per ridurre quindi il numero di curve "hard coded", si è valutata la possibilità di inserirne solo una ed utilizzarne una seconda per l'ottenimento della deviazione da questa (i.e., da quella standard). Questo scarto sarà infatti dipendente dal valore di radiazione ambientale rilevato. L'utilizzo di questa tecnica, se effettivamente funzionante anche nella pratica, potrebbe ridurre notevolmente la memoria occupata.

Un ultimo punto di miglioramento della libreria è infine relativo al **test su altre schede** della stessa. Questa è infatti stata testata ed utilizzata, attualmente, solo su Arduino. Non essendone quindi stato provato il funzionamento su altre schede, non si è sicuri del fatto che nessuna di esse possa richiedere modifiche specifiche al codice.

## **Conclusione**

Lo sviluppo di questa tesi ha portato alla creazione di un insieme di moduli utilizzabili nell'ambito del progetto della piattaforma robotica AlpaBot. Questi includono: un dispositivo per comunicazione locale e sensoristica di prossimità, un applicativo per l'interazione da remoto con la piattaforma robotica e un insieme di librerie atte a rendere agevole l'utilizzo della stessa.

Il processo di sviluppo delle componenti della tesi ha seguito le fasi di progettazione, implementazione e raffinamento, portando infine all'ottenimento di buoni risultati.

Nonostante il desiderio di migliorarne ulteriormente alcune parti e dettagli, la piattaforma risultante ha generato non poca soddisfazione, sia a livello di funzionamento, sia a livello di difficoltà superate.

Le componenti hardware e software prodotte infatti, si sono rivelate, a seguito di test di funzionamento, sufficientemente funzionali ed adatte allo scopo prefissato relativo alla didattica.

## **Ringraziamenti**

Al termine di questa relazione, ritengo di dover mettere per iscritto dei ringraziamenti ad alcune persone che mi hanno dato una mano durante la realizzazione del progetto e della tesi in sé.

Primo fra tutti devo ringraziare il Professor Roli per aver accettato la proposta di tesi congiunta con A. Lombardi e per aver impiegato tempo nel seguirmi sia durante la fase di progettazione che durante quella di redazione della la tesi.

La seconda persona che merita i miei ringraziamenti è ovviamente il mio compagno di tesi Alessandro Lombardi, grazie al quale è partita e proseguita l'intera idea del progetto.

Come terza persona ritengo giusto ringraziare Andrea Giulianini. È infatti grazie alla sua disponibilità che siamo stati in grado di completare, a livello pratico, le stampe 3D del progetto.



## Appendice

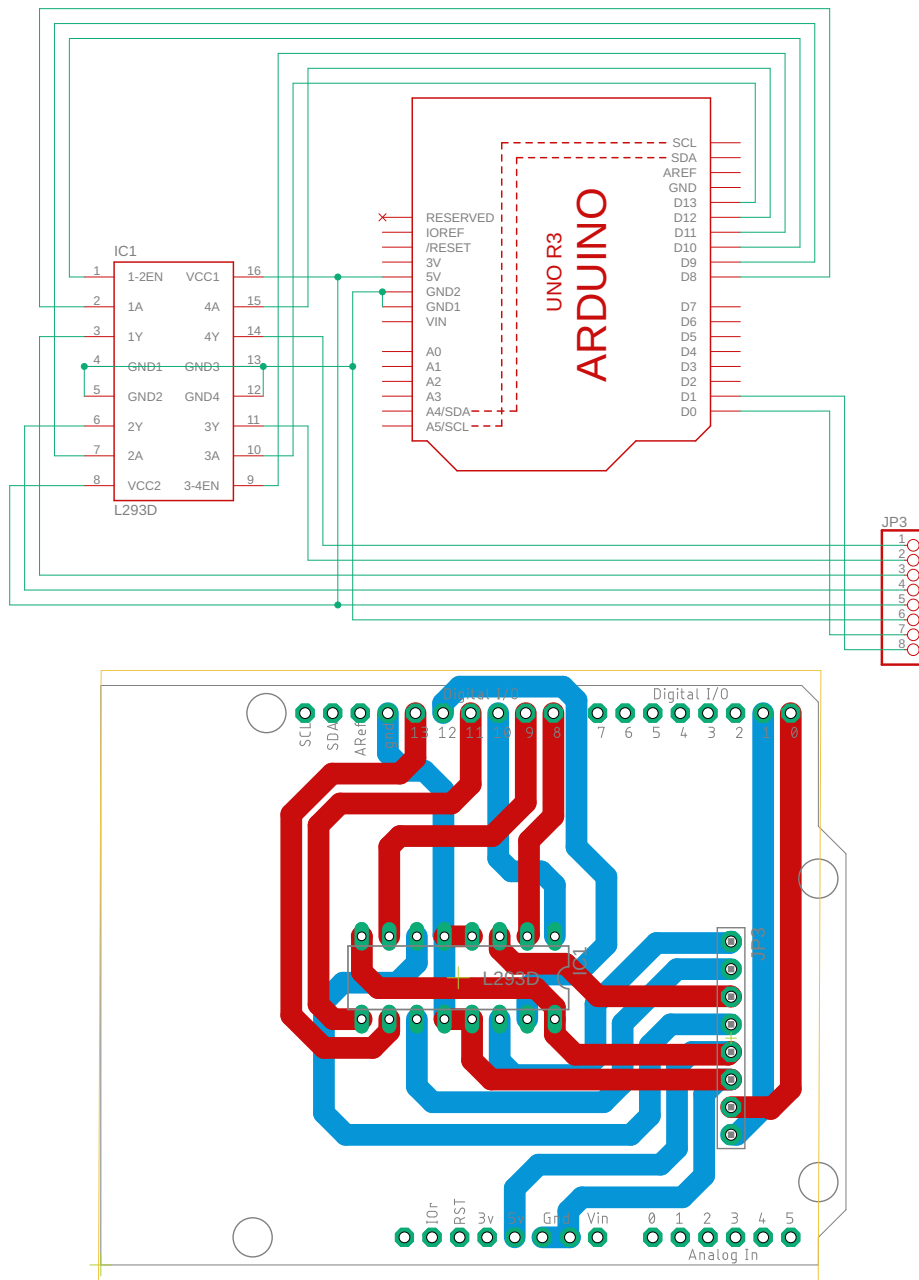


Figura 30: La scheda relativa alla gestione motori. Studiata da Alessandro Lombardi nella sua parte di tesi, viene convertita in modulo vero e proprio dal sottoscritto. Nota: I pin rx e tx della Seriale sono stati collegati al connettore per comodità di utilizzo.

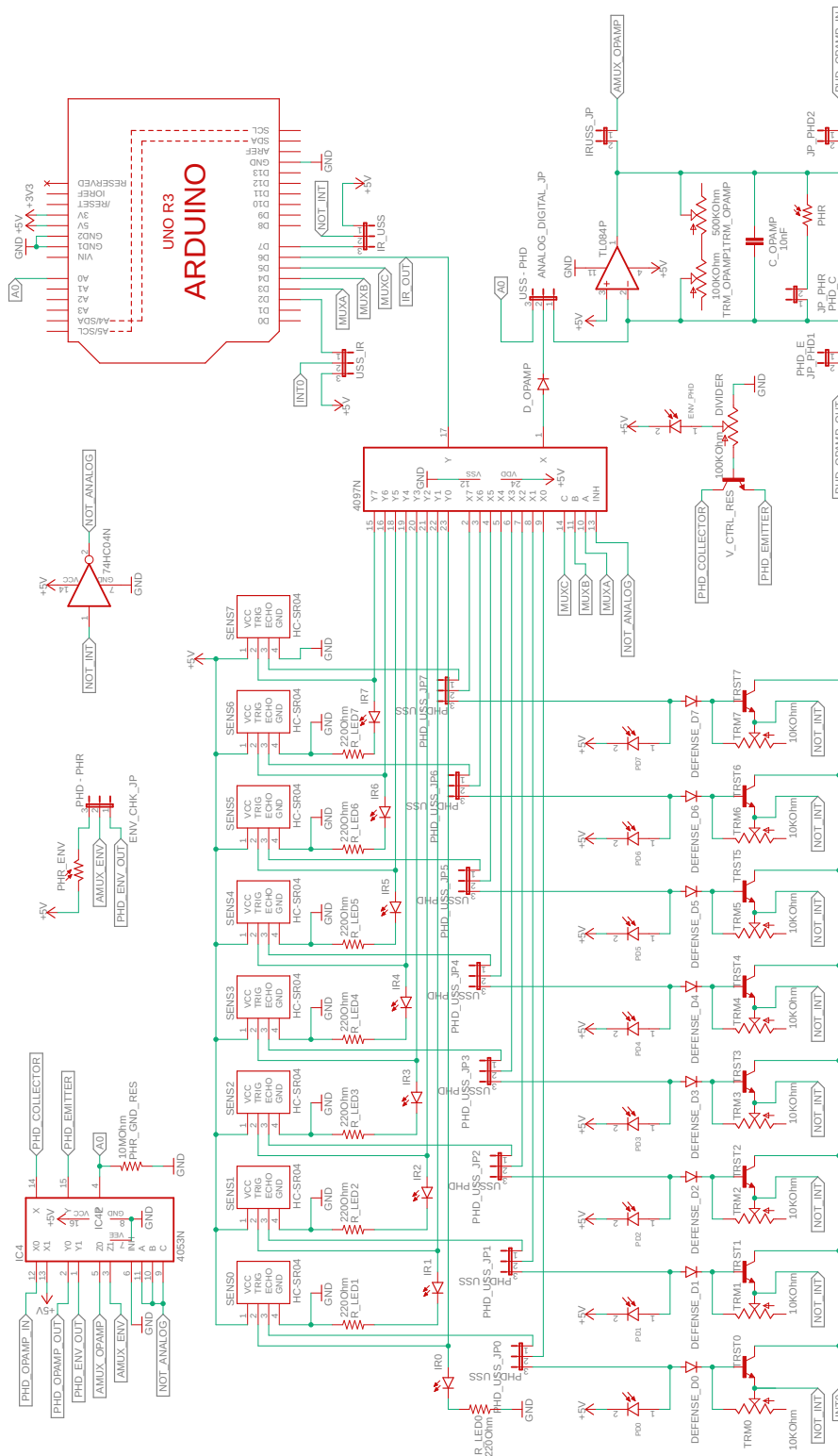


Figura 31: Lo schema elettrico finale del modulo infrarosso/ultrasonico IRUSS.

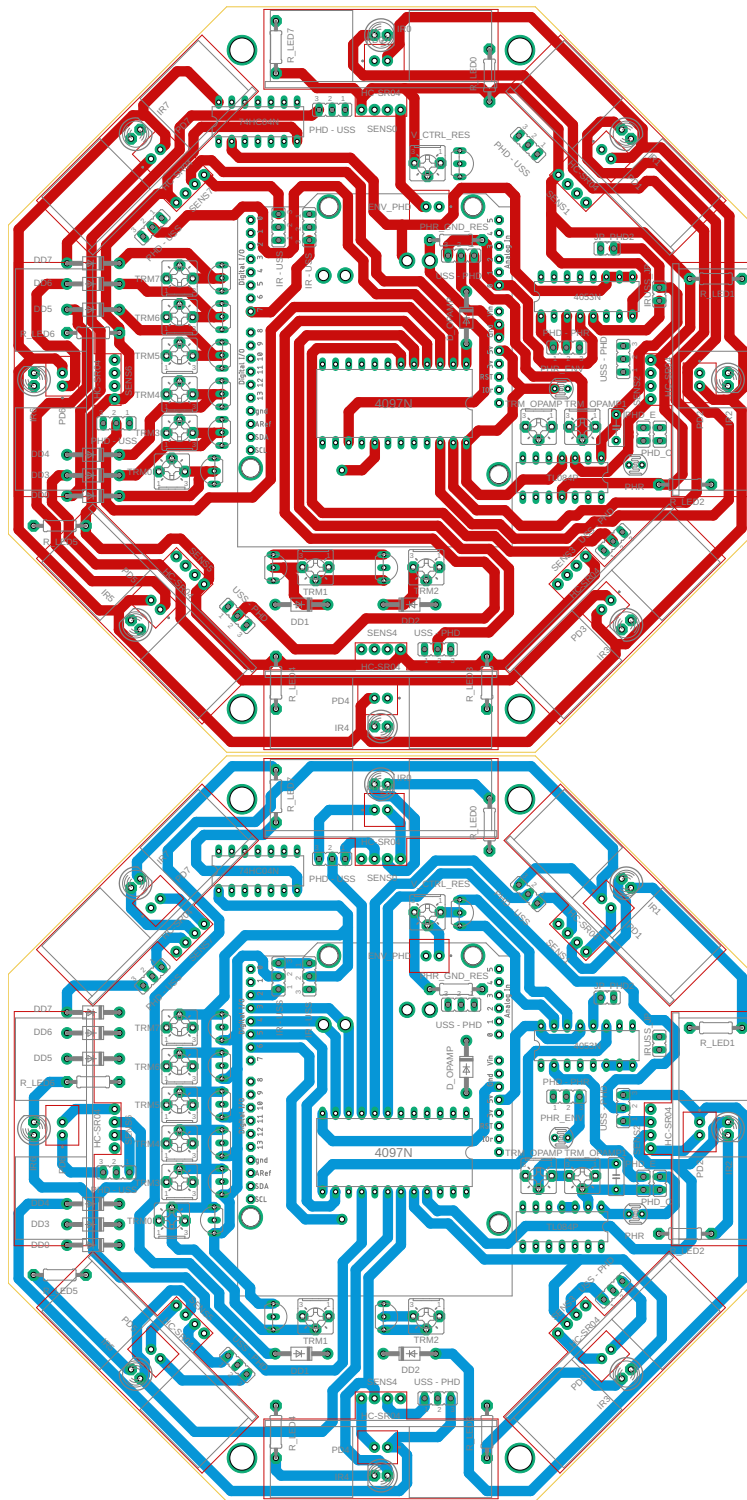


Figura 32: Lo sbroglio delle due facce del PCB del modulo IRUSS. Sopra è visibile la faccia superiore, sotto l'inferiore.

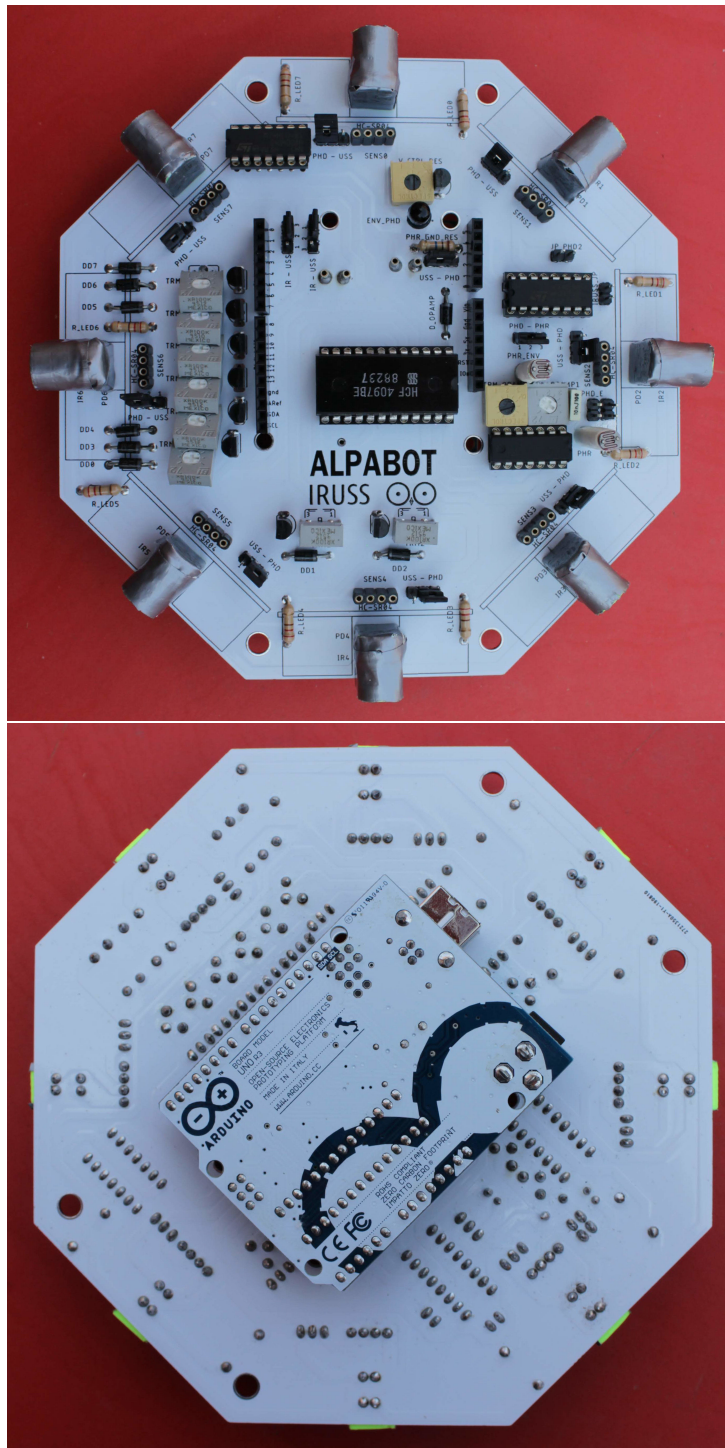


Figura 33: Le due facce del modulo IRUSS. La prima foto rappresenta la configurazione solo infrarosso. La seconda (lato inferiore) mostra un Arduino collegato.

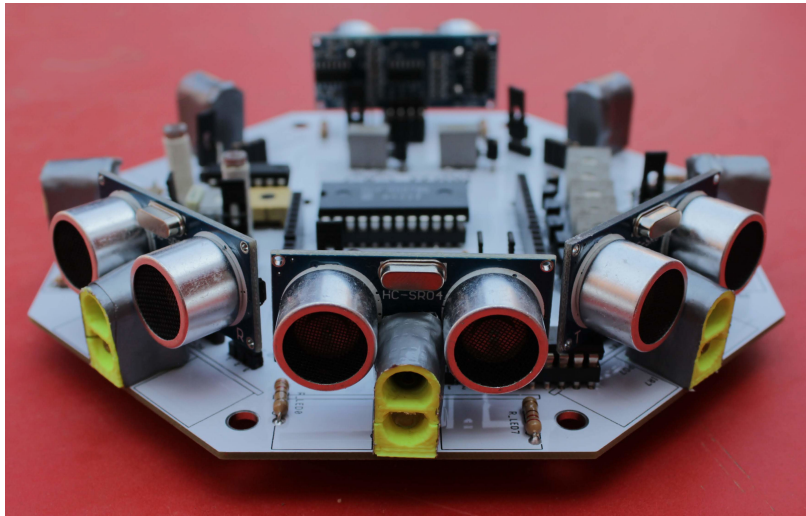


Figura 34: Il modulo IRUSS nella sua configurazione con sensori ad ultrasuoni.

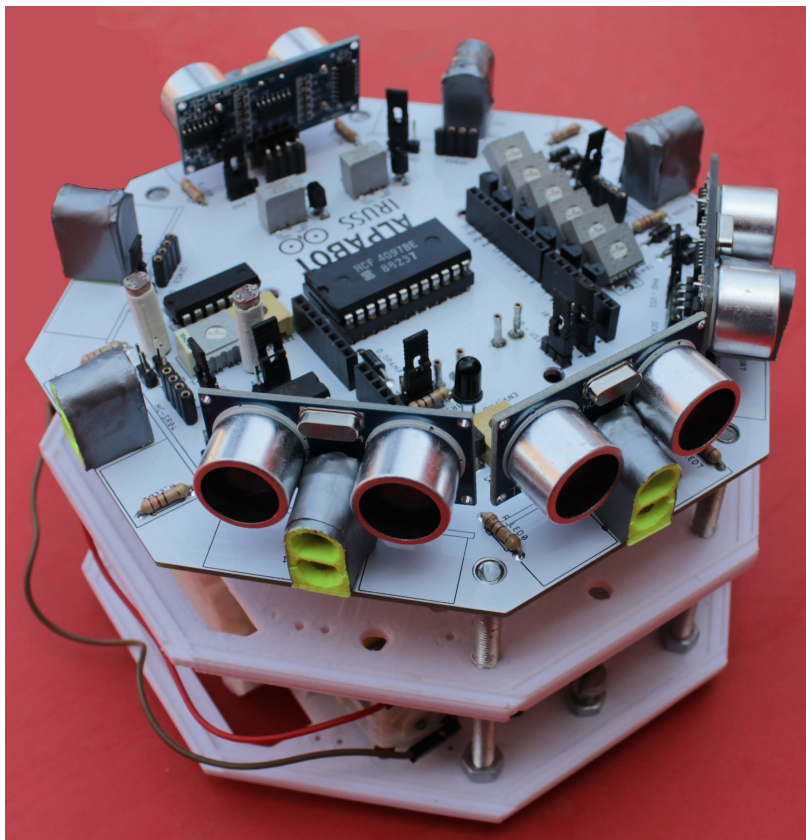


Figura 35: La piattaforma AlpaBot nella sua interezza.

## Riferimenti bibliografici

- [1] Radhika Nagpal and Michael Rubenstein at Harvard University. Kilobot.
- [2] École Polytechnique Fédérale de Lausanne. Alice microbot, 2004.
- [3] University of Stuttgart and Universität Karlsruhe. Jasmine iii.
- [4] Ulf Witkowski and Reza Zandian. *TAROS - Novel Method of Communication in Swarm Robotics Based on the NFC Technology*. Springer, 2013.
- [5] IRIDIA, IDSIA, EPFL-LSRO, EPFL-LIS, and CNR-ISTC. Swarmanoid: Towards humanoid robotic swarms.
- [6] Chunlin Chen, Shizhen Qu, Han Wu, and Dongdong Xu. Precise localization and formation control of swarm robots via wireless sensor networks, November 2014.
- [7] Kumar Yelamarthi. An autonomous passive rfid-assisted mobile robot system for indoor positioning, April 2014.
- [8] ASI. Galileo - global navigation satellite system.
- [9] Ryan Michael. Arduino-splines.
- [10] Benoît Blanchon and Other. Arduinojson v6.