

TECH ARM

TESINA DI
LUCA SEVÀ

INDIRIZZO ELETTRONICA

ARTICOLAZIONE AUTOMAZIONE

ANNO SCOLASTICO
2017/2018

Sommario

| | |
|---|----|
| PREMESSA..... | 2 |
| Aspetti teorici | 3 |
| Tecnologia e cinematica dei robot..... | 3 |
| Tipologie di robot..... | 3 |
| Cinematica | 6 |
| Servomotori..... | 7 |
| Giroscopi basati su mems | 9 |
| Protocollo di comunicazione Bluetooth..... | 12 |
| Tech arm | 14 |
| Braccio robotico | 14 |
| Sviluppo hardware: meccanica braccio robot | 14 |
| Sviluppo hardware: PCB di controllo | 16 |
| Sviluppo software: ricezione ed esecuzione di movimenti e comandi | 18 |
| Guanto di controllo | 21 |
| Sviluppo hardware: guanto | 21 |
| Sviluppo hardware: circuito di trasmissione | 22 |
| Sviluppo software: acquisizione ed invio di movimenti e comandi | 23 |
| Sviluppi futuri..... | 24 |
| Bibliografia e sitografia | 25 |

PREMESSA

Nell'ambito del percorso scolastico intrapreso e della continua evoluzione tecnologica del settore dell'automazione, ho deciso di sviluppare un braccio robotico controllato in remoto attraverso il movimento della mano umana.

Le funzionalità principali del progetto sono le seguenti:

- esecuzione in real time dei movimenti della mano
- apprendimento dei movimenti con la possibilità di eseguirli su richiesta
- possibilità di afferrare piccoli oggetti mediante una morsa
- pannello di controllo con informazioni rilevanti sul funzionamento
- funzioni relative alla sicurezza.

Sono state implementate alcune funzioni attivabili mediante una pulsantiera posta sul palmo del guanto di controllo come il pulsante di emergenza, l'apertura e chiusura della morsa e l'avvio delle funzioni di apprendimento. La comunicazione tra le schede è realizzata mediante il protocollo di comunicazione Bluetooth così da eliminare impedimenti legati a cavi di comunicazione mantenendo allo stesso tempo una velocità di trasmissione dati accettabile.

Al fine di creare un progetto il più possibile verosimile ad un prodotto industriale, sono stati inclusi strumenti per la diagnostica e la risoluzione di problemi.

Per facilitare l'installazione e la messa in opera del sistema è stato creato un manuale utente contenente istruzioni ed informazioni utili per identificare e risolvere problemi comuni che potrebbero presentarsi durante l'uso.

Il progetto può essere ulteriormente sviluppato, aggiungendo funzioni e dispositivi di sicurezza e aumentandone l'efficienza energetica.

Nell'ottica della produzione in serie potrebbe essere implementato in applicazioni altamente specializzate come robot per il disarmo di ordigni esplosivi, ambienti sterilizzati (cappe di sicurezza biohazard) per radiologia o medicina nucleare o dispositivi di manipolazione fisica in ambienti pericolosi. Inoltre, si renderà necessario implementare dispositivi aggiuntivi per poter adempiere ai requisiti minimi per ottenere certificazioni di settore come "UL Listed", "FCC Compliant", "RoHS", "ENERGY STAR®" e "CE".

Aspetti teorici

Al fine di avere una comprensione completa del sistema trattato in questa tesina si procederà all'esposizione dei seguenti argomenti in forma teorica:

- tecnologia e cinematica dei robot
- servomotori
- giroscopi basati su mems
- protocollo di comunicazione Bluetooth.

Tecnologia e cinematica dei robot

In generale si definiscono robot tutti quei manipolatori automatici progettati per eseguire cicli di operazioni ripetitive, lungo diversi assi di rotazione.

Un robot è quindi in grado di eseguire operazioni come:

- movimentazione di materiali, pezzi e utensili
- montaggio di componenti
- lavorazione meccaniche mediante strumenti appositi.

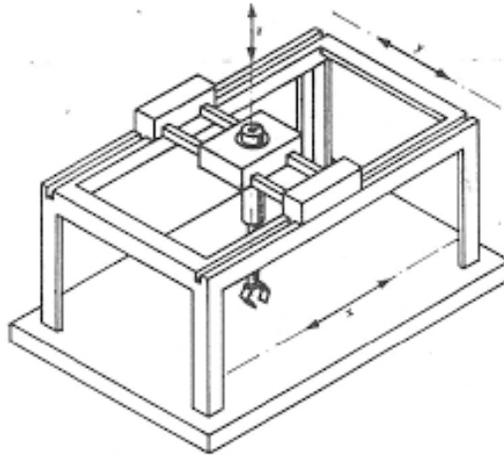
Tipologie di robot

Esistono varie tipologie di robot, classificati in base alla loro geometria cinematica:

- robot cartesiano
- robot cilindrico
- robot sferico o polare
- robot SCARA
- robot antropomorfo.

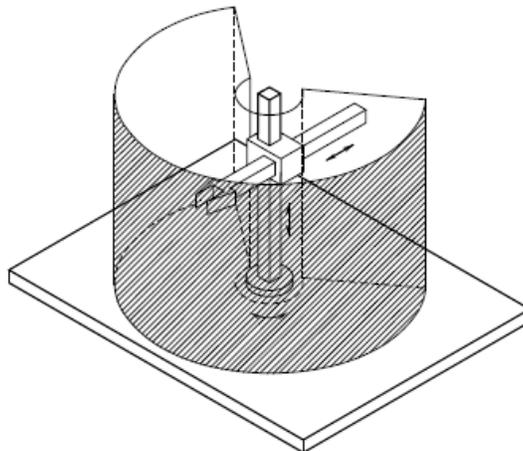
1. Robot cartesiano

Utilizza tre articolazioni prismatiche che realizzano movimenti lineari lungo gli assi X, Y e Z. Lo spazio di movimento è un parallelepipedo. Questi robot sono utilizzati nei processi dove il piano cartesiano viene utilizzato come riferimento per le istruzioni di movimentazione. Un'esempio applicativo sono le stampanti 3D o le macchine per il taglio laser.



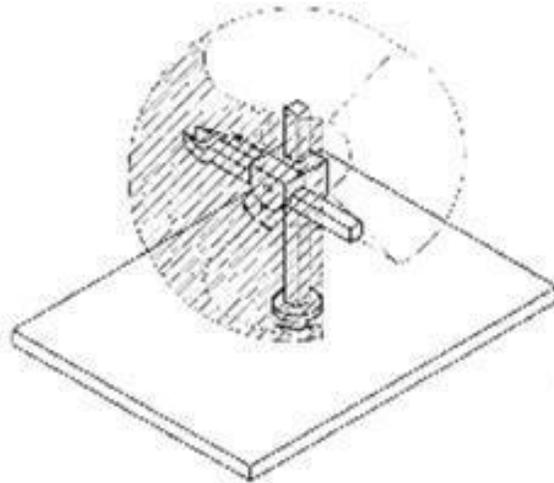
2. Robot cilindrico

Utilizza tre articolazioni rotoidali per realizzare movimenti in uno spazio di lavoro cilindrico cavo. Vengono impiegati in ambienti di lavoro dove i materiali da movimentare sono particolarmente pesanti e quindi è giustificato l'uso di motori pneumatici piuttosto che elettrici.



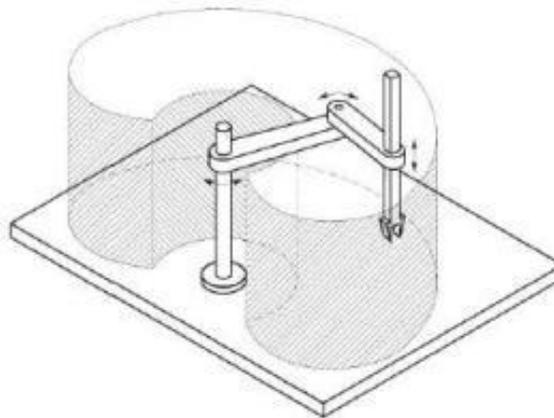
3. Robot sferico o polare

Utilizza due articolazioni di rivoluzione e una prismatica. Lo spazio di lavoro è una semi sfera. Permettono di eseguire lavorazioni precise su materiali posti alla loro base. Esempi applicativi sono perforatrici e levigatrici di materiali metallici.



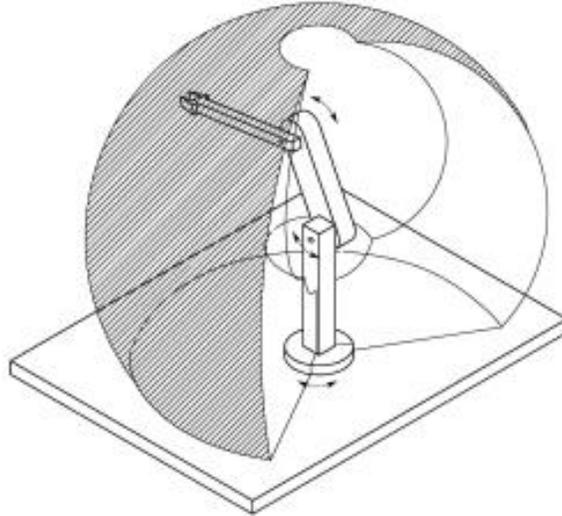
4. Robot SCARA

Questo tipo di robot SCARA (Selective Compliant Assembly Robot Arm) è costituito da due bracci articolati collegati ad una base centrale fissa. I movimenti sono di tipo orizzontale tranne per l'asse Z che viene gestivo dall'end effector. Sono utilizzati molto in ambito medico ed elettronico per la loro velocità di ciclo completo. Infatti, i robot SCARA sono costruiti per eseguire un ciclo completo di lavorazione in meno di un secondo.



5. Robot antropomorfo

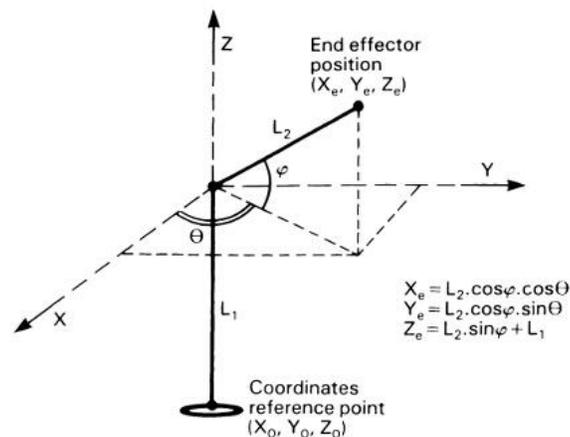
Il nome di questa tipologia chiarisce la struttura del robot: simula la meccanica e la cinematica del braccio umano. Il numero di assi di rotazione dipende dal numero di motori montati sulla struttura. Lo spazio di lavoro è rappresentato, approssimativamente, da una sfera. Permette di avere enormi gradi di libertà a scapito della semplicità dei modelli di movimento, per questo non si fa più riferimento ad una singola matrice di coordinate cartesiane. Le coordinate di movimento sono composte da un'attenta interpolazione di valori degli assi X, Y e Z rispetto alla base e il valore angolare del movimento eseguito.



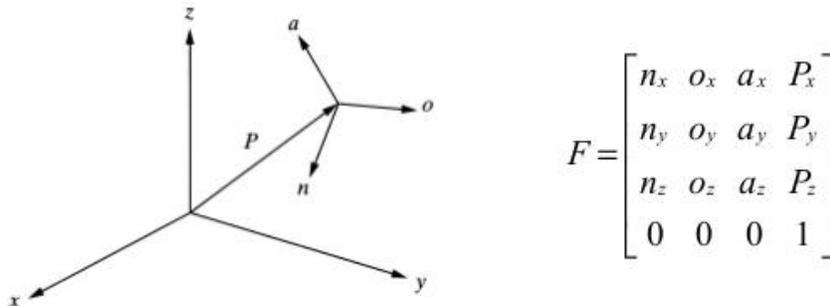
Ognuna delle tipologie di robot sopra indicati possono montare dispositivi per la manipolazione dei materiali così da rendere operativo in ambito produttivo il robot.

Cinematica

Nella cinematica di un robot si deve considerare che ogni movimento deve essere compiuto mediante precise regole di traslazione. In base alla tipologia di robot e all'ambiente dove viene impiegato non è possibile creare un modello universalmente applicabile per la gestione del movimento. Lo spazio di lavoro del robot viene mappato con un sistema di spazio cartesiano a tre variabili (X, Y e Z) mentre i movimenti degli assi e degli end effector sono descritti mediante un sistema di vettori ed angoli (L_2 , Θ e φ) comunemente chiamato sistema polare. Per descrivere posizione, direzione e traiettoria degli assi viene quindi impiegato un sistema a matrici. La figura seguente rappresenta gli assi, i vettori e angoli che compongono il sistema di rappresentazione descritto.



La figura seguente mostra la rappresentazione spaziale della matrice usata per definire i vettori di movimento che il robot può eseguire lungo un suo giunto.

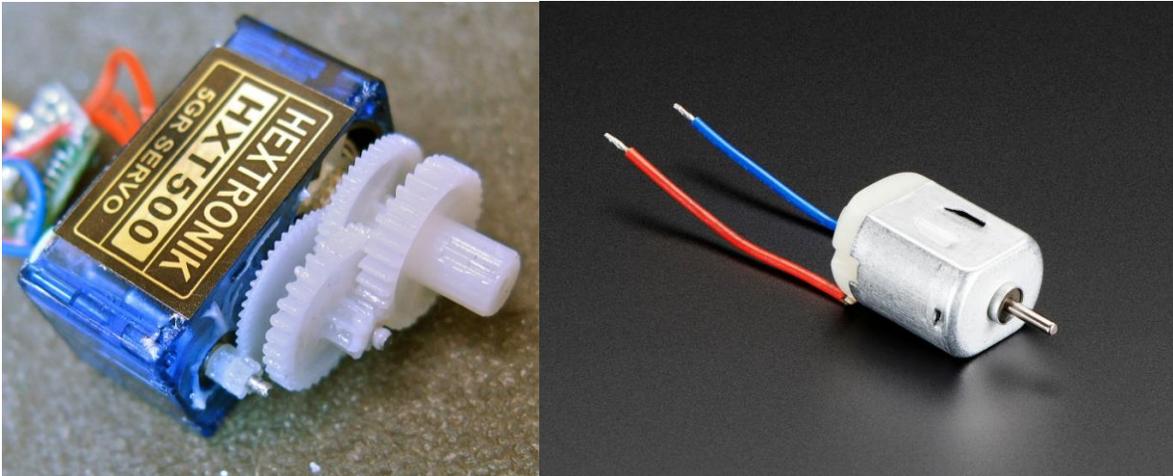


Servomotori

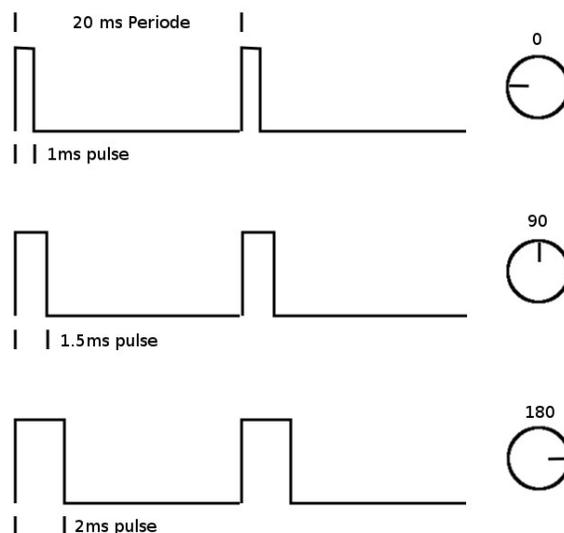
Il servomotore è una particolare tipologia di motore impiegato in vari ambiti specializzati, come automazione, modellismo ed elettronica grazie alla sua funzionalità più importante: posizionarsi precisamente in corrispondenza dell'angolo desiderato e di mantenere la posizione nonostante grandi coppie di carico applicate. Sono dispositivi molto versatili grazie al loro ingombro volumetrico limitato e peso ridotto e alla semplicità di utilizzo. Esistono vari modelli di servomotori che si differenziano tra loro per dimensioni, peso e capacità meccaniche.



I servomotori nonostante siano pilotati da segnali PWM (Pulse Width Modulation) consistono, meccanicamente, di un motore DC con una serie di riduttori. Quest'ultimi permettono di ridurre la rotazione finale del motore così da ottenere una coppia maggiore.



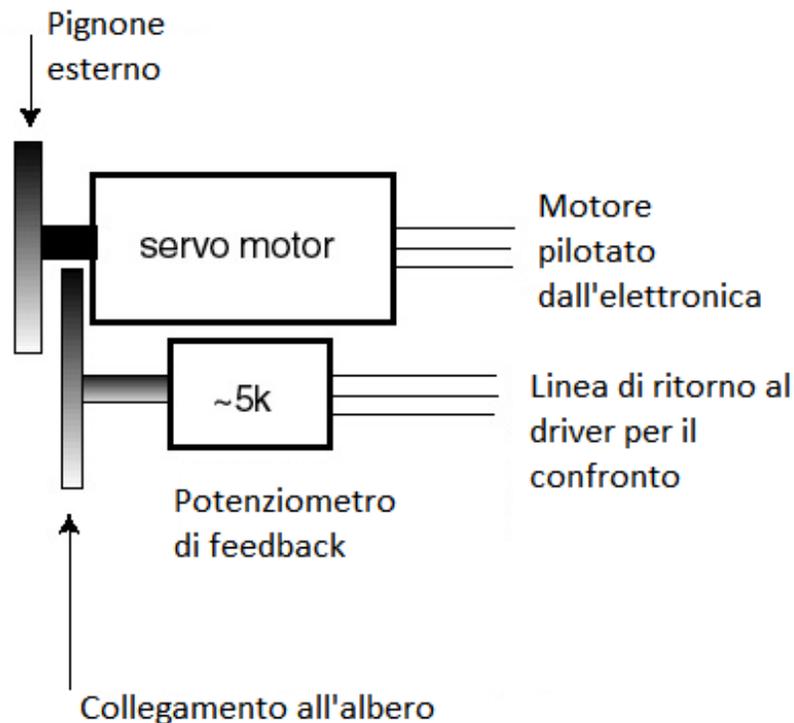
Generalmente il segnale PWM viene fornito da un driver per motori apposito o da un microcontrollore con funzionalità legate al controllo dei timer interni per temporizzare il segnale. Il segnale presenta caratteristiche ben definite: periodo dell'impulso (20ms) e duty cycle (5%÷10%). La variazione del duty cycle viene interpretato dalla logica interna del motore per poter posizionarsi al valore angolare desiderato.



I servomotori più comuni hanno un grado di rotazione di $0^\circ \div 180^\circ$ ma esistono versioni che sono in grado di compiere rotazioni complete.

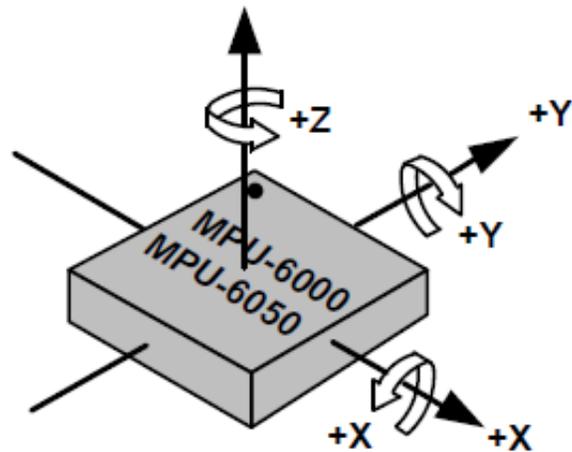
La funzionalità legata all'autoregolazione è gestita da un potenziometro controllato da logica interna. Internamente al motore, collegato all'albero, è posto un potenziometro che viene monitorato continuamente [1.1]. Quest'ultimo rileva il valore resistivo e lo usa per determinare se il motore è nella posizione indicata e se vi sono delle coppie applicate

al motore che tendono a modificare l'angolo. Se viene rilevata una variazioni non voluta il chip comanda il motore nella direzione inversa rispetto alla direzione ortogonale della coppia. Durante questa operazione il motore tende a consumare una quantità di energia maggiore rispetto al movimento normale, dato che deve muovere il motore molto più spesso per cercare di bilanciare l'effetto della coppia di carico.

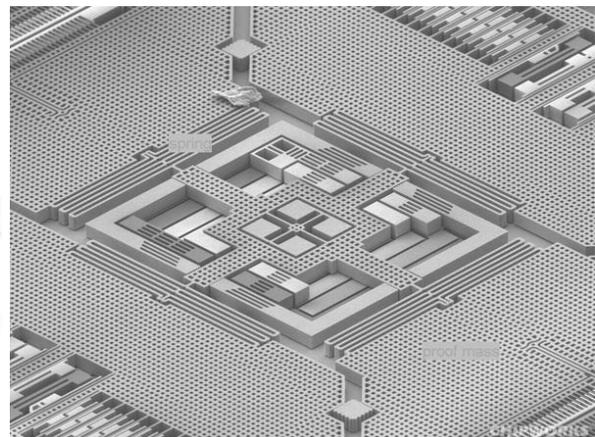
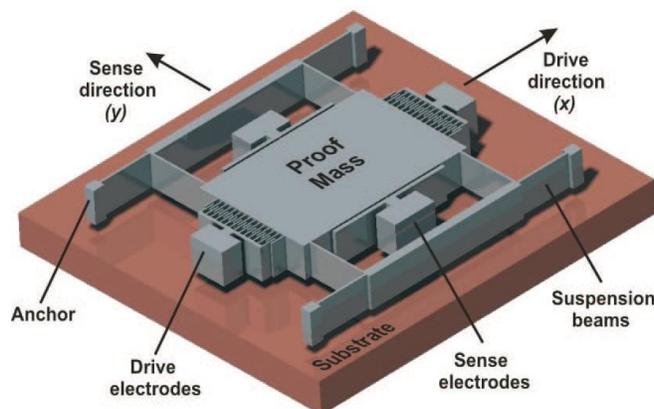


Giroscopi basati su mems

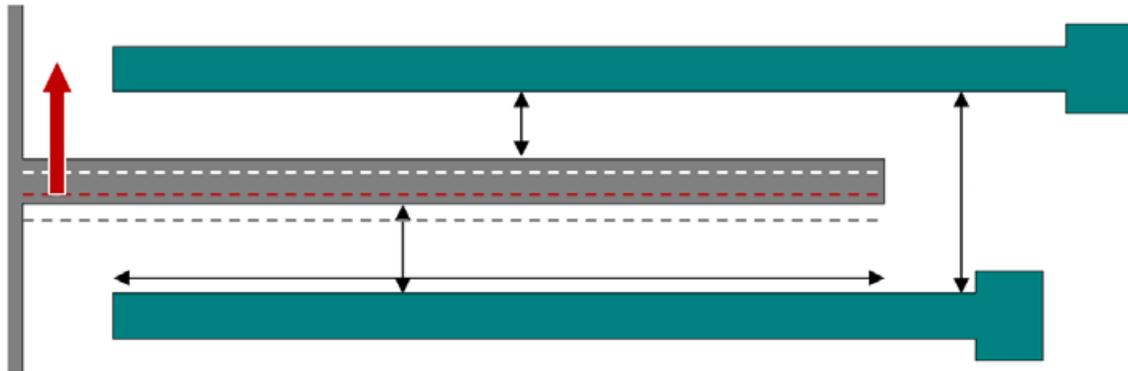
Grazie all'utilizzo dei MEMS (Micro Electro-Mechanical Systems) è possibile realizzare una moltitudine di dispositivi che sfruttano proprietà e fenomeni meccanici in uno spazio molto ridotto. I giroscopi integrati nei dispositivi elettronici odierni sono di tipo inerziale. Il principio di funzionamento, sebbene di semplice comprensione, richiede approfondimenti a livello matematico molto complessi che non verranno trattati in questa tesina. Un corpo, in questo caso il package contenente il giroscopio, subisce un'accelerazione impressa dal movimento nello spazio di riferimento. Questa accelerazione verrà poi usata per determinare su quale asse il dispositivo sta ruotando e permetterà di determinarne l'angolo di rotazione. La figura seguente mostra l'orientamento degli assi di sensibilità di un giroscopio MEMS:



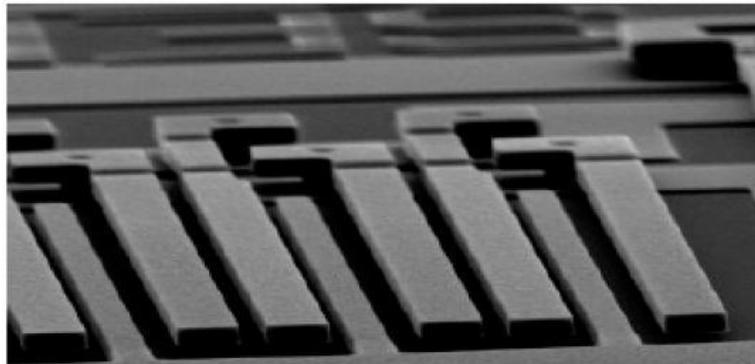
Dal punto di vista funzionale il giroscopio permette di rilevare le velocità angolari che sono applicate al dispositivo così da tramutarle in valori angolari. In generale un giroscopio basato su MEMS è composto da una massa sospesa su di un substrato di silicio definito “strato di drive”. La massa viene ancorata a delle strutture di supporto mediante delle molle di sospensione: queste permetteranno il movimento della massa quando una accelerazione esterna viene applicata. Il movimento della massa può essere acquisito mediante varie implementazioni di strutture in silicio come elettrodi e capacità sensibili alla posizione della massa. Questo è reso possibile grazie all’accelerazione di Coriolis. Questa accelerazione, applicata alla massa sospesa del giroscopio, si tramuta in una forza che agisce sulla massa (traslazione di massa) e sui giunti di supporto (torsione di supporto). Nelle figure seguenti sono descritti la struttura concettuale e a vista microscopica di un giroscopio MEMS.



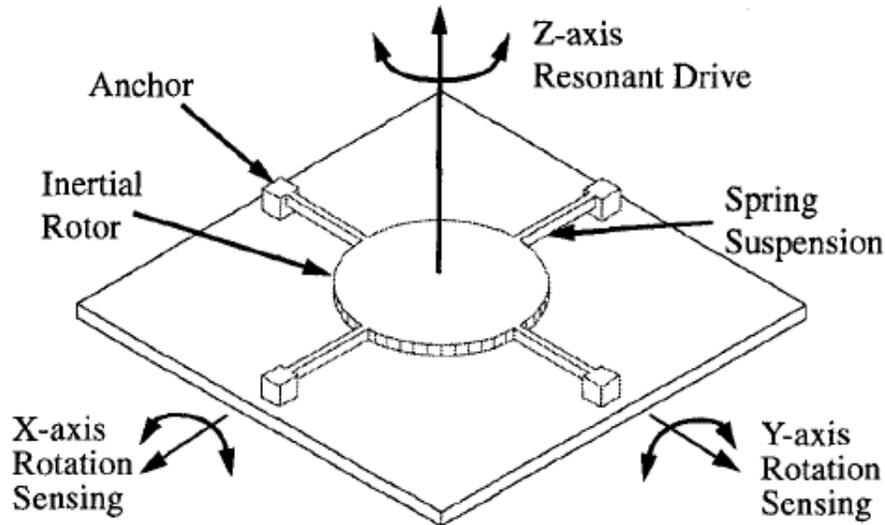
Nel caso di un giroscopio a tre assi (giroscopio rotazionale, [1.2]) la struttura interna rimane pressoché uguale eccetto per il sistema con cui vengono ancorate le masse interne allo strato di drive. I sistemi di ancoraggio e di sospensione della massa hanno integrati i dispositivi di trasduzione, definite strutture capacitive. Nella figura di seguito la struttura grigia è l'estensione della massa sospesa mentre quelle in turchese sono strutture poste sui sistemi di ancoraggio.



Particolare delle strutture capacitive viste al microscopio elettronico.



Lo spostamento della struttura della massa all'interno dello spazio tra le lamelle capacitive provoca una variazione di capacità. Quest'ultima verrà usata per determinare la quantità di accelerazione di Coriolis e, di conseguenza, la forza applicata sugli ancoraggi. In base alla tipologia e alla sensibilità che si vuole ottenere con il giroscopio MEMS, la struttura in silicio può variare per ottenere proprietà particolari in assi singoli (asse Z più sensibile o maggiore rigidità dei supporti). In generale la struttura di un giroscopio rotazionale è la seguente:



Protocollo di comunicazione Bluetooth

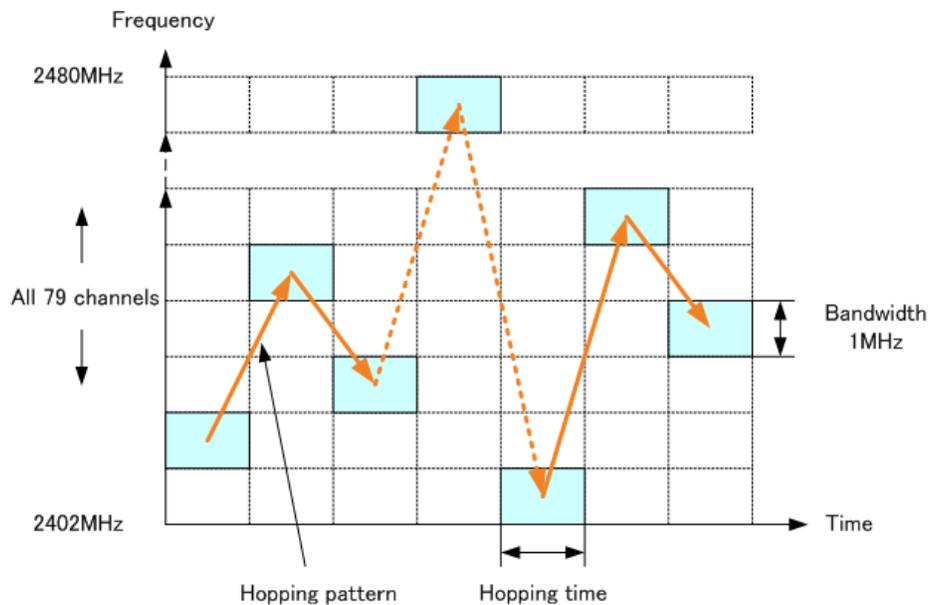
Tra le varie tipologie di protocolli di comunicazione il Bluetooth è ormai un pilastro sia nell'industria che nei prodotti consumer. Questo è reso possibile dal continuo supporto e sviluppo di nuove funzionalità da parte di aziende ed enti internazionali di standard. I vantaggi principali sono i seguenti:

- bassissimo consumo energetico
- copertura di medie distanze (~ 10m-15m)
- banda dati elevata
- velocità di trasmissione.

Per quanto riguarda il principio di funzionamento del protocollo Bluetooth l'analisi può essere svolta su più piani di astrazione, in base al livello hardware e software dell'applicazione. In questa tesina si svilupperà il layer di protocollo e non si approfondirà la sezione riguardo il flusso di dati e di controllo dell'integrità degli stessi.

La comunicazione Bluetooth si basa su due tipi di identità: il master (o host) e lo slave. A seconda di chi instaura la comunicazione viene stabilito il livello gerarchico della comunicazione. Alla prima connessione (pairing) i due dispositivi condividono delle informazioni che permetteranno la riconnessione nel caso uno dei due venga spento o disconnesso. Questi dati sono in genere una chiave di autenticazione per il master e una per lo slave, gli indirizzi fisici dei dispositivi e il canale di comunicazione che usano. I dispositivi Bluetooth sfruttano un metodo di trasmissione denominato "Frequency hopping" che permette di associare più dispositivi ad un singolo master. La banda di

frequenza del protocollo Bluetooth ha uno spettro dai 2.402GHz ai 2.480GHz. ogni canale, chiamato "hops", occupa 1MHz della banda sopra citata (vi sono 79 hops in totale). Oltre a permettere il cosiddetto multi-pairing, la tecnica degli hops permette di aumentare la banda totale di trasmissione dislocando i dati da trasmettere in pacchetti di dimensioni minori che verranno inviati su hops separati [1.3].



Tech arm

Il progetto sviluppato per questa tesina nasce da un'esigenza personale che potrebbe essere estesa a molte persone nel mondo: avere un modo veloce e soprattutto semplice di controllare e programmare un piccolo braccio robotico. Oggigiorno questi dispositivi sono utilizzati soprattutto in ambienti industriali ma l'andamento del settore creativo ed hobbistico (makers) sta creando un interesse sempre maggiore da parte delle aziende di settore come KUKA e Mitsubishi che vogliono proporre soluzioni a prezzi ridotti. Purtroppo, avendo maturato negli anni un interesse particolare per sole applicazioni industriali, i metodi di programmazione e controllo dei robot risultano complessi sia dal punto di vista hardware che software per coloro che non hanno ricevuto un training specializzato. Ho deciso quindi di abbattere interamente il layer concettuale della programmazione e controllo che viene usato oggi e di sostituirlo con un sistema più semplice ed intuitivo: il movimento della mano umana.

Braccio robotico

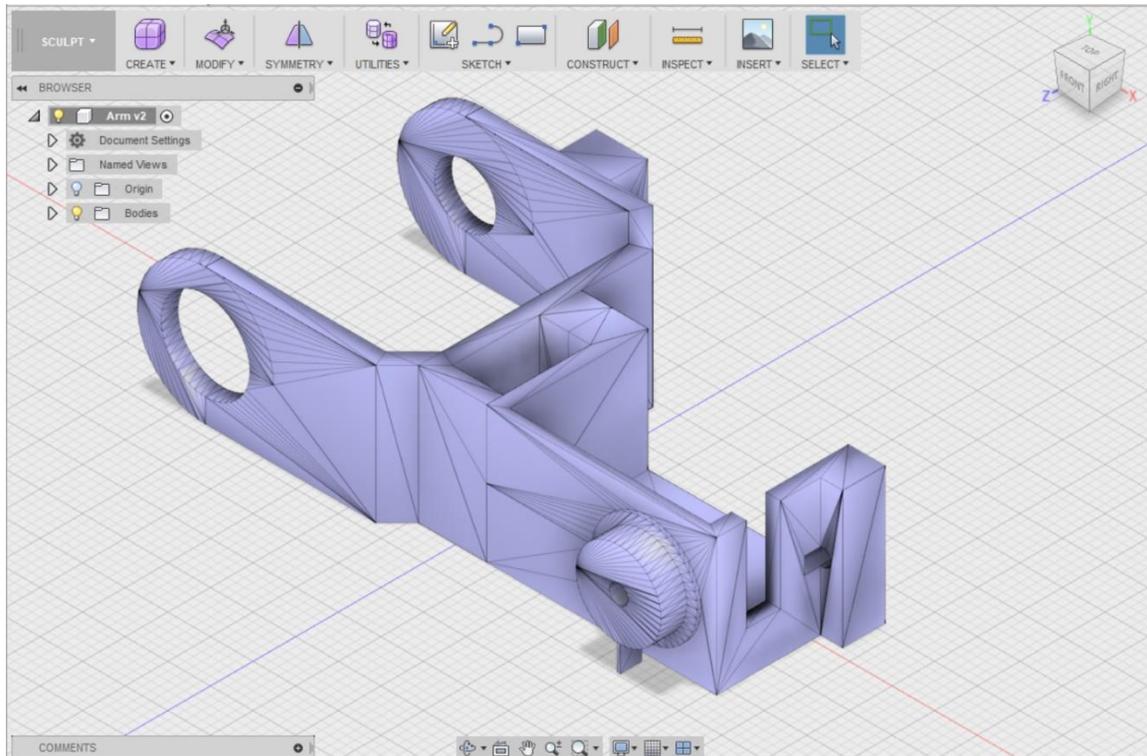
Il braccio robotico è controllato da remoto, via Bluetooth, dal guanto. I dati ricevuti servono a determinare il movimento del braccio nello spazio, così come l'apertura e chiusura della morsa ed il blocco di emergenza. È possibile inoltre acquisire pattern di movimenti che possono essere eseguiti a richiesta. Lo sviluppo hardware del braccio robotico ha coinvolto la meccanica della struttura del braccio stesso e la circuiteria di controllo. Lo sviluppo software è stato incentrato sulla gestione dei dati e l'autoapprendimento.

Sviluppo hardware: meccanica braccio robot

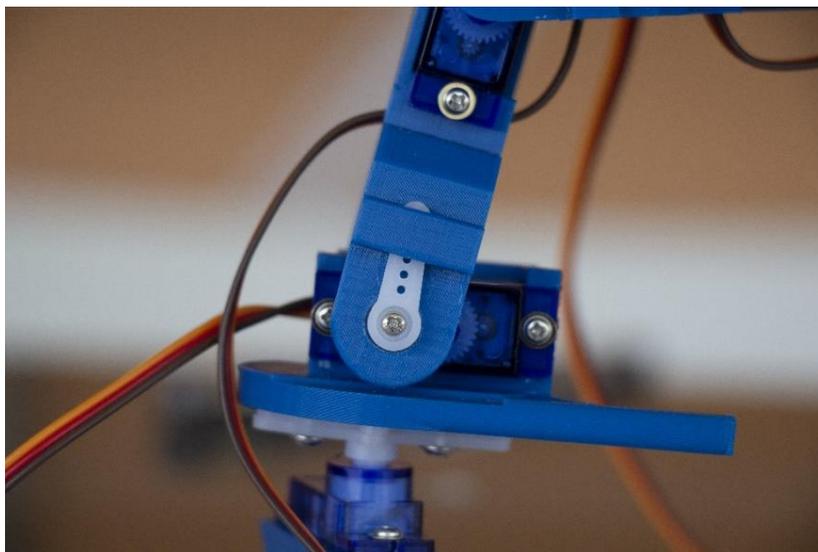
Nell'ambito dell'utilizzo di tecnologie avanzate ho deciso di creare la struttura del braccio robotico mediante l'uso della stampa 3D. Questo ha reso possibile l'implementazione di strutture complesse per la movimentazione e l'alloggiamento dei motori servo. La struttura risulta così solida ma allo stesso tempo leggera e modulare. Le parti che compongono la struttura sono:

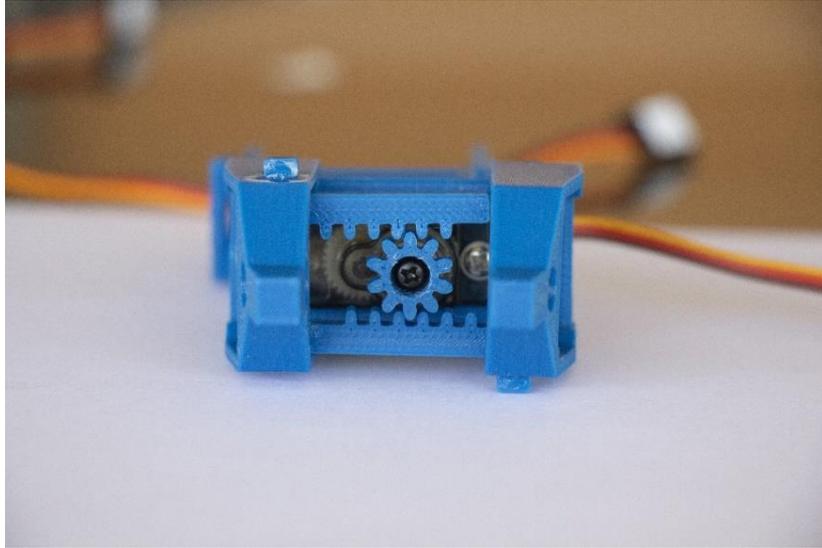
- base di montaggio
- base rotante
- moduli centrali
- morsa meccanica.

Il materiale utilizzato per la stampa è chiamato “Ultrat”, un filamento in abs rinforzato, che conferisce stabilità anche alle strutture più sottili e fragili. La progettazione dei modelli è stata realizzata mediante il software Fusion360.



Dopo la stampa dei modelli ho assemblato il braccio con i motori servo nei rispettivi alloggiamenti come mostrato nelle seguenti foto.





Sviluppo hardware: PCB di controllo

Per il circuito di controllo del braccio robotico ho definito le seguenti specifiche:

- integrazione di un pannello operativo
- integrazione di un modulo Bluetooth per la comunicazione
- alimentazioni per logica programmabile e motori separate
- dimensioni ridotte.

Il circuito è stato sviluppato, in forma schematica e layout PCB, mediante il software Altium Designer. Il circuito è suddiviso nei seguenti blocchi di funzione:

- microcontrollore ATMEGA2560
- display LCD da 2.8" con lettore di microSD
- modulo Bluetooth con relativo circuito di level shift
- regolatore per tensione di alimentazione
- connettori motori.

Il microcontrollore è integrato in una scheda di sviluppo Arduino (Mega), questo rende più semplice il collegamento tra i vari dispositivi utilizzati. L'intero circuito è stato realizzato con lo scopo di rendere semplice e immediato l'identificazione di problemi e danni hardware così da facilitare la manutenzione. Di seguito è proposto il circuito completo del PCB di controllo del braccio robotico.

Il modulo Bluetooth [P3; HC-05], il cui livello logico è a 3V3, crea problemi di coerenza di logica: il microcontrollore ha livello logico alto a 5V. Durante la fase di scrittura del modulo Bluetooth, benché il livello logico non corrisponda, non vi sono problemi in quanto la soglia di riconoscimento di livello alto da parte dell'ATMEGA2560 ha un range di 3V÷5V. Il problema sorge quando avviene la lettura da parte del modulo: la tolleranza è di 3V3 massimo, qualsiasi valore al di sopra di tale soglia potrebbe causare comportamenti imprevedibili. Per questo viene usato un level-shifter, costituito da un partitore di tensione [R1;R2] per portare il segnale ad una soglia accettabile. Sebbene non sia la soluzione più corretta dal punto di vista elettrico (le resistenze assorbono corrente modificando il valore di tensione) essa rappresenta un'alternativa economica ad integrati dedicati per questa funzione.

La regolazione della tensione di alimentazione in ingresso avviene per mezzo di un regolatore di tensione L7805CV [U2; L7805CV] e coppia di condensatori di filtraggio [C3;C4], con valore da datasheet del componente. Questo regolatore, ponendo in ingresso una tensione da 7V fino ad un massimo di 25V, fornisce in uscita 5V costanti (con assorbimento di corrente fino a 1.5A). Questa tensione viene usata esclusivamente per alimentare i motori servo mediante una linea di alimentazione separata [+E5].

Il microcontrollore [U1; Arduino_Mega2560R3] viene alimentato direttamente dalla tensione non regolata di ingresso [VIN]. Il circuito complementare al ATMEGA2560 si occupa della generazione di una tensione di 5V [+5] che viene usata per alimentare il modulo Bluetooth. Questa precauzione è stata presa per garantire che i disturbi indotti nell'alimentazione dai motori in movimento non influisca sulla comunicazione e di conseguenza sull'integrità della comunicazione e il livello di sicurezza del dispositivo.

Sviluppo software: ricezione ed esecuzione di movimenti e comandi

Lo sviluppo hardware è stato affiancato da quello software. La programmazione del microcontrollore è stata compiuta con l'uso del linguaggio di programmazione ad alto livello derivato dal C/C++. Il codice sorgente è stato suddiviso in blocchi funzionali separati, e richiamati durante l'esecuzione mediante le "call di funzione". I blocchi principali sono:

- gestione Bluetooth ed acquisizione dati
- gestione del display LCD con microSD
- gestione dei movimenti
- gestione delle funzioni aggiuntive.

Di seguito verranno proposti stralci del codice con commenti.

```

if (Serial1.available()) {
  //read and substring division process
  //receive = Serial1.read();
  receiveA = Serial1.readStringUntil('A');
  receiveX = Serial1.readStringUntil('X');
  receiveY = Serial1.readStringUntil('Y');
  receiveZ = Serial1.readStringUntil('Z');
  moveX = receiveX.toInt();
  moveY = receiveY.toInt();
  moveZ = receiveZ.toInt();
}

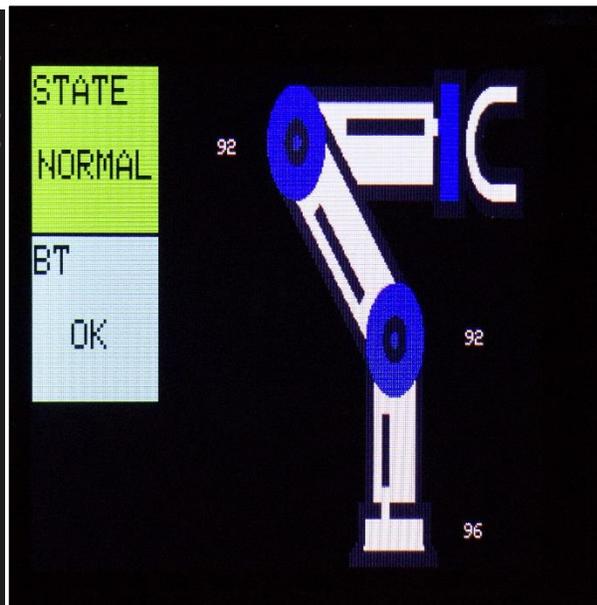
```

I dati inerenti al valore angolare del giroscopio e ai codici operativi per le funzionalità sono organizzati in un buffer inizializzato String. Questo permette di utilizzare le funzioni integrate nelle librerie che permettono di suddividere il buffer, a mano a mano che viene ricevuto, in sezioni più piccole mediante dei separatori [A, X, Y, Z]. I dati vengono poi convertiti in variabili Int per poi essere utilizzati per movimentare i motori. Un esempio di buffer completo potrebbe essere: [25A100X87Y174Z].

```

int TFTNORMAL() {
  tft.fillRect(SB_X, SB_Y, SB_W, SB_H, GREEN);
  tft.fillRect(CN_X, CN_Y, CN_W, CN_H, CYAN);
  tft.drawRect(SB_X, SB_Y, SB_W, SB_H, BLACK);
  tft.drawRect(CN_X, CN_Y, CN_W, CN_H, BLACK);
  //tft.drawRect(0, 160, 80, 26, WHITE);
  //tft.drawRect(0, 186, 80, 26, WHITE);
  //tft.drawRect(0, 212, 80, 26, WHITE);
  //Text for layout
  tft.setCursor(SB_X + 2, SB_Y + 5);
  tft.setTextColor(BLACK);
  tft.setTextSize(SB_TSIZE);
  tft.print("STATE");
  tft.setCursor(CN_X + 2, CN_Y + 5);
  tft.setTextColor(BLACK);
  tft.setTextSize(CN_TSIZE);
  tft.print("BT");
  tft.setCursor(SB_X + 5, 40);
  tft.setTextColor(BLACK);
  tft.setTextSize(CN_TSIZE);
  tft.print("NORMAL");
}

```



Il display LCD viene gestito mediante una libreria specifica, rendendo più semplice il processo di sviluppo di interfacce grafiche. Questo procedimento utilizza la sovrapposizione di elementi grafici in un unico layer. Questo aumenta la velocità ma, se si vuole modificare un elemento che è sovrapposto ad un altro, bisogna ricreare l'intera sezione, elemento per elemento. Il codice proposto è responsabile della visualizzazione del pannello operativo durante le normali operazioni di funzionamento. La prima parte

[tft.fillRect; tft.drawRect] crea e riempie di colore due rettangoli che verranno usati come indicatori per lo stato generale del dispositivo e della connessione Bluetooth. La seconda parte [tft.set] permette di impostare la posizione ed il testo che verrà visualizzato sul display. È possibile modificare la posizione, la dimensione del carattere ed il colore del carattere stesso.

```
servo1.write(moveX); //motor_1
servo2.write(moveY); //base
if (moveX < 100) {
    servo3.write(moveX); //motor_2
    if (moveX > 70)
    {
        servo4.write(moveX); //motor_3
    }
}
```

Il movimento dei motori avviene mediante una libreria in grado di gestire la temporizzazione dei segnali PWM dei motori. La funzione “servo*.write(move*)” converte il valore dell’angolo desiderato in un segnale che posizionerà il motore. Questo valore è in continuo cambiamento essendo generato dal giroscopio posto sul guanto di controllo.

```
//Datalog routine
if (learn == true) {
    learndata.print(moveX);
    learndata.print("A");
    learndata.print(moveX);
    learndata.print("X");
    learndata.print(moveY);
    learndata.print("Y");
    learndata.print(moveZ);
    learndata.println("Z");
    lines++; //increment the number of lines
    learndata = SD.open("data.txt", FILE_READ);
    for (int j = 0; j < lines; j++) { //write moves
        receiveA = learndata.readStringUntil('A');
        receiveX = learndata.readStringUntil('X');
        receiveY = learndata.readStringUntil('Y');
        receiveZ = learndata.readStringUntil('Z');
        moveX = receiveX.toInt();
        moveY = receiveY.toInt();
        moveZ = receiveZ.toInt();
    }
}
```

La funzione di apprendimento e ripetizione dei movimenti avviene mediante l’uso di una memoria microSD. Quando il braccio è in apprendimento i movimenti di ogni asse vengono memorizzati mediante la funzione “learndata.print(*)” (immagine a sinistra), nella memoria. Questa operazione permetterà alla funzione “readStringUntil(*)”

(immagine a destra) di gestire la suddivisione e la conversione dei valori angolari per poter inviare i comandi memorizzati ai motori.

Guanto di controllo

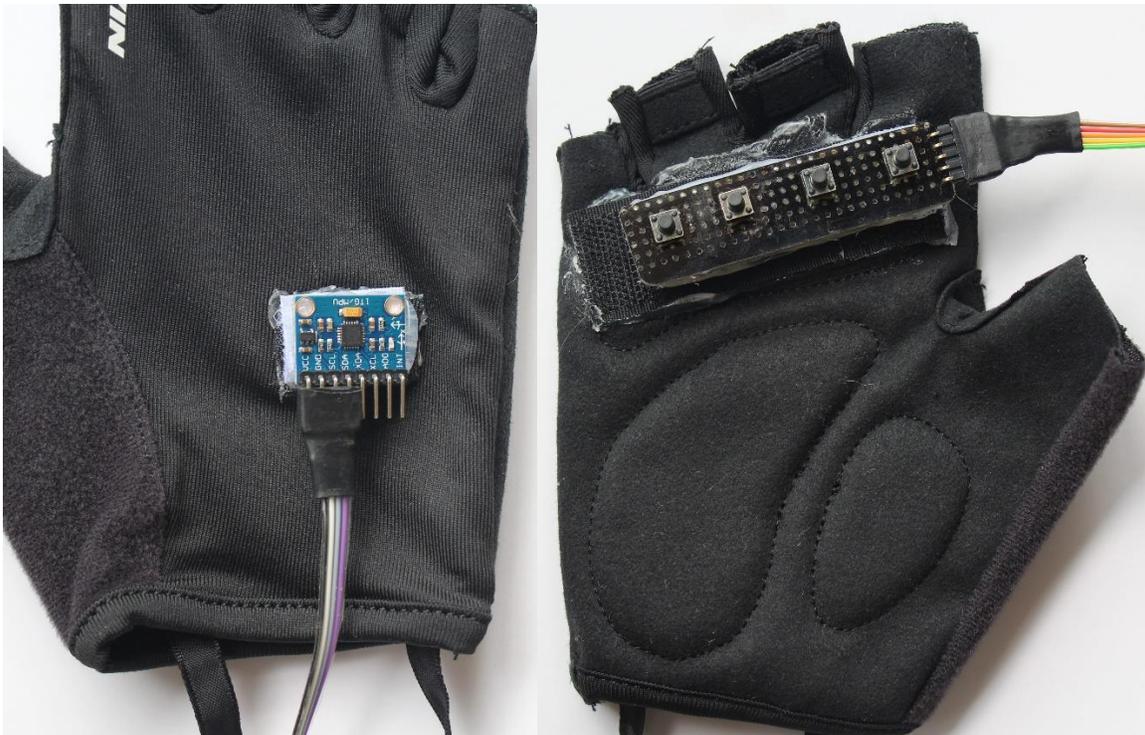
Il guanto è il dispositivo con cui l'utente interagisce per controllare il braccio robotico. Alimentato a batteria e con Bluetooth integrato offre una soluzione portatile con funzionalità avanzate:

- giroscopio integrato
- pulsantiera per funzioni
- modulo di ricarica per batteria Li-Ion

Lo sviluppo hardware ha compreso la progettazione del guanto e del circuito di trasmissione mentre il software è stato incentrato sull'acquisizione e trasmissione dei dati.

Sviluppo hardware: guanto

Il guanto utilizzato ha dei supporti in velcro per montare il giroscopio e la pulsantiera.



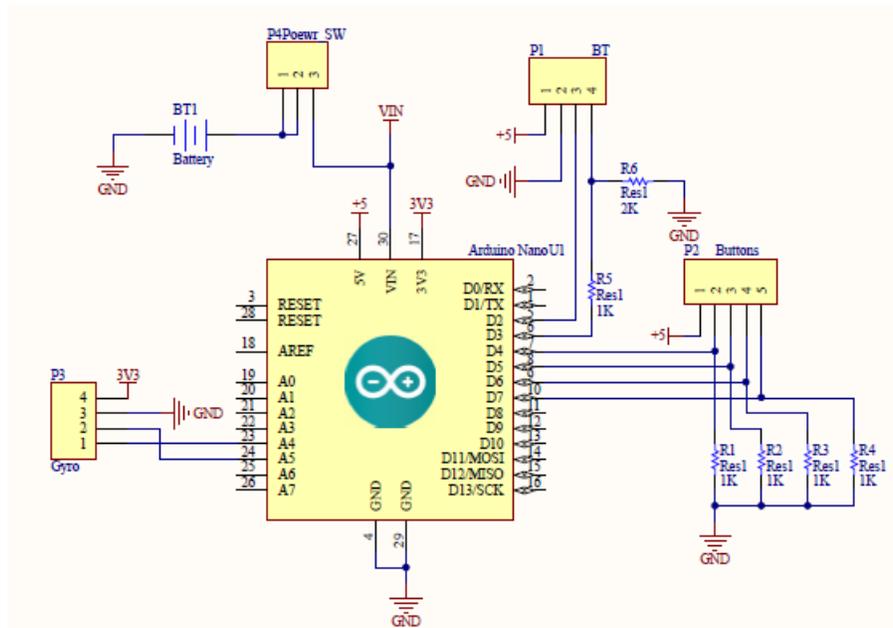
In questo modo il giroscopio non ha elementi che potrebbero influenzarne il funzionamento e la pulsantiera è facilmente accessibile con le dita della mano. Per il collegamento al circuito di trasmissione è stato utilizzato un multi-cavo piatto con due connettori.

Sviluppo hardware: circuito di trasmissione

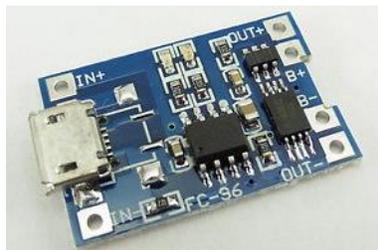
Il circuito è stato sviluppato in modo tale da occupare il minor spazio possibile sul braccio dell'utente. Viene alloggiato in una fascia dalla quale fuoriesce il multi-cavo usato per connettere il guanto. Gli elementi principali del circuito di trasmissione sono i seguenti:

- microcontrollore ATMEGA328P
- modulo Bluetooth con relativo circuito di level shift
- connettori per giroscopio e pulsantiera

Di seguito è proposto il circuito completo.



Come per il circuito di controllo del braccio robotico, il microcontrollore è integrato in una scheda Arduino (Nano) così da poter usufruire del medesimo linguaggio di programmazione e librerie. Il modulo Bluetooth e la rete di level shift sono identici a quelli precedentemente descritti. In aggiunta sono stati usati due connettori per i cavi che andranno collegati al giroscopio e pulsantiera [P3, Gyro; P2, Buttons], un connettore per la batteria Li-Ion e l'interruttore di accensione [BT1; P4, Power_SW]. Il circuito di protezione e ricarica della batteria non è integrato nel circuito di figura, viene invece utilizzato un modulo esterno.



Sviluppo software: acquisizione ed invio di movimenti e comandi

Lo sviluppo software per il guanto di controllo è stato incentrato sui seguenti blocchi funzionali:

- acquisizione della posizione del giroscopio nello spazio
- invio dei dati alla scheda di controllo del braccio robotico

Di seguito vengono proposti stralci del codice sviluppato.

```
sensor.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);  
//junk data to prevent bugs in the BT/Serial com. in th ARM code  
junk = map(ay, -17000, 17000, 0, 180);  
//raw data mapped to servo's min/max degrees[90 default position]  
moveX = map(ax, -17000, 17000, 0, 180);  
moveY = map(ay, -17000, 17000, 180, 0);  
moveZ = map(az, -17000, 17000, 0, 180);
```

La gestione della comunicazione e acquisizione dati è affidata ad una libreria appositamente sviluppata per il giroscopio utilizzato: il MPU6050. Questo permette di avere un set di funzioni pronte all'uso, come "sensor.getMotion6(****)", per manipolare i dati generati dal giroscopio e poterli usare come valori numerici per il pilotaggio dei motori servo. Alla funzione sopracitata vengono fornite delle variabili in cui verranno memorizzati i valori angolari del giroscopio. Quest'ultimi vengono poi mappati (proporzionati, "map(**)") rispetto al valore minimo e massimo di angolo motore.

```
BT.print(junk);  
BT.println("A");  
BT.print(moveX);  
BT.println("X");  
BT.print(moveY);  
BT.println("Y");  
BT.print(moveZ);  
BT.println("Z");
```

I valori appena proporzionati vengono usati per generare il buffer di dati menzionato nel software di gestione del braccio robotico. I dati vengono intervallati da dei caratteri ("A", "B", etc.) che identificano a quale asse corrisponde il movimento. Una volta che l'intero buffer viene creato si procede all'invio mediante il Bluetooth.

Sviluppi futuri

Il progetto esposto è un prototipo per una tecnologia ancora in fase di sviluppo. L'obiettivo primario del mio progetto, oltre allo sviluppo delle mie capacità e conoscenze nell'ambito delle applicazioni hardware e software, è quello di fornire un metodo alternativo per il controllo e la programmazione di bracci robotici. Questo prevede che vi siano ambiti di applicazione dove l'introduzione di nuove metodologie e tecnologie possano costituire dei fattori fondamentali per lo sviluppo di procedure operative più semplici ed efficaci.

Esempi di applicazioni possono essere:

- robot altamente specializzati (es. disinnescano ordigni esplosivi)
- dispositivi di manipolazione fisica in ambienti pericolosi
- medicina nucleare e radiologica (radiofarmaci)
- settore creativo ed hobbistico (makers).

Riconducendomi in modo particolare al mondo dei makers, riscontro che i costi di sistema, programmazione e gestione sono fattori scoraggianti per coloro che dispongono di risorse e conoscenze specifiche limitate. Il mio progetto può essere facilmente integrato con dispositivi già utilizzati dagli utenti, dal punto di vista meccanico e software. Così come già successo per la stampa 3D ed il taglio laser, il mio progetto si prefigge di favorire una diffusione più ampia e semplice di una tecnologia ad oggi ancora poco accessibile.

Bibliografia e sitografia

- [1.1] <https://www.electronicshub.org/servo-motors/>
- [1.2] A. Castro, “Sviluppo di una metodologia per l’ottimizzazione di prove sperimentali per la caratterizzazione di giroscopi MEMS monoassiali e biassiali”, tesi di laurea, Politecnico di Milano, 2009
- [1.3] Bluetooth SIG, “Specifications of the Bluetooth system”, “Master Table of Contents & Compliance Requirements”, 2004