

	Laboratorio di Robotica	Classe 5RA
Data 30/5/24	Alunno: <i>Lupi Matteo</i>	A.S.: 2023/24

OGGETTO:	Coil Winder
----------	--------------------

Coil Winder
1.0

Indice

Introduzione.....	2
Obiettivi del progetto:.....	2
Introduzione all'Arduino:.....	3
Descrizione coil winder:.....	3
Fasi del progetto:.....	4
Schema a blocchi del sistema:.....	5
Componenti e materiali utilizzati:.....	5
-Arduino Uno R3:.....	5
-Hall sensor:.....	5
-Servo motore con rotazione 180°:.....	5
-LCD I2C Display (Liquid Crystal Display):.....	6
-Cavi maschio femmina e femmina femmina:.....	6
-Potenziometro 10kΩ:.....	6
-Pulsante:.....	6
-Dual shaft motor:.....	6
-Buzzer:.....	6
-Copper coil:.....	6
-Li-Po battery ricaricabili da 9V 800mAh:.....	7
-Modulo L298N:.....	7
Realizzazione.....	8
TEST E DEBUG:.....	8
Conclusione.....	8

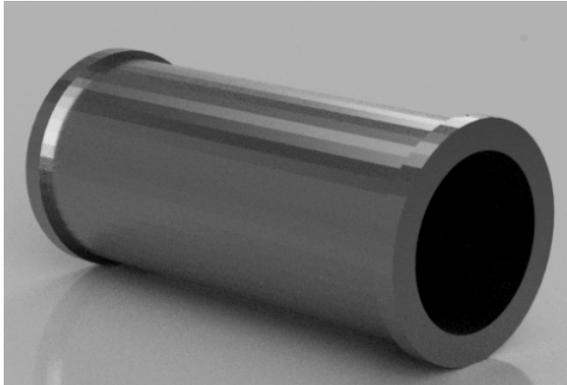
Introduzione

Questo progetto è stato realizzato in seguito all'esperienza svolta durante il PCTO nell'estate 23/24. Esso si tratta di una coil winder, ossia una ribobinatrice, è una macchina utilizzata per avvolgere fili conduttori intorno a un nucleo per creare bobine e avvolgimenti. La sua funzione principale è quella di avvolgere il filo in modo uniforme e controllato intorno al nucleo, seguendo un modello specifico, come determinato dalle specifiche del progetto. Questo processo è essenziale per la produzione di componenti elettronici e sistemi elettromeccanici che richiedono bobine con caratteristiche precise, come induttori, trasformatori, solenoidi e motori elettrici. Durante l'alternanza scuola-lavoro i macchinari che ho potuto osservare erano tutti programmati tramite PLC, nel mio progetto ho pensato di replicare tali macchinari realizzando una struttura in miniatura che possa svolgere una funzione simile ai macchinari reali utilizzando però la scheda Arduino.

Obiettivi del progetto:

L'obiettivo del progetto è quello di realizzare una coil winder in miniatura utilizzando Arduino e le funzioni che esso offre, in quanto mi è impossibile realizzare una ribobinatrice delle

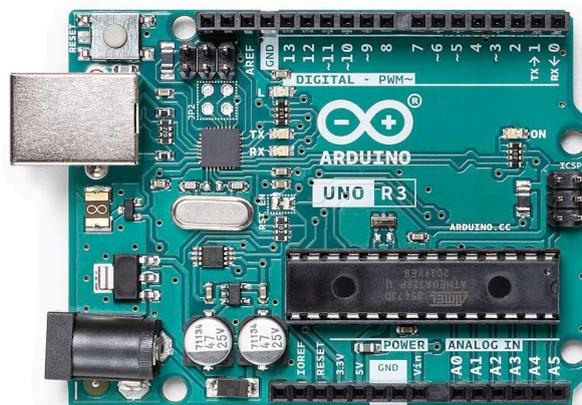
dimensioni naturali. Il mio progetto consiste nel realizzare delle bobine più piccole partendo da una bobina più grande costituita da una bobina di rame. Le bobine più piccole che si andranno a creare saranno a completa discrezione dell'utente, il quale potrà scegliere quanti avvolgimenti fare e il peso finale che dovrà avere la bobina che gli serve.



Questo è un esempio di bobina che ho realizzato. Essa è stata prodotta in PLA e contiene un piccolo foro nel quale va inserito il filo da avvolgere.

Introduzione all'Arduino:

Per realizzare questo progetto ho utilizzato la scheda Arduino UNO R3, esso rappresenta la parte intelligente che gestisce tutti i comandi da inviare ai vari componenti del circuito: Arduino è una piattaforma open-source utilizzata per la creazione di prototipi di dispositivi interattivi e progetti di automazione. È composta da hardware e software progettati per essere accessibili anche ai principianti, consentendo loro di creare rapidamente e facilmente progetti elettronici e di programmare senza necessariamente avere una profonda conoscenza di elettronica o programmazione. La piattaforma si basa su una scheda microcontroller che può essere programmata per eseguire una vasta gamma di compiti. Questa scheda è dotata di ingressi e uscite digitali e analogici che consentono agli utenti di interagire con sensori, attuatori e altri componenti elettronici. Il cuore della scheda Arduino è un microcontroller AVR o ARM, che esegue il codice scritto dagli utenti per controllare il funzionamento del dispositivo. Esso è accompagnato da un ambiente di sviluppo integrato (IDE) che fornisce un'interfaccia user-friendly per scrivere, caricare ed eseguire il debug del codice sorgente. Il linguaggio di programmazione utilizzato è basato su C/C++.



Descrizione coil winder:

Una coil winder o ribobinatrice è una macchina utilizzata per avvolgere fili conduttori intorno a un nucleo per creare bobine e avvolgimenti. Ecco come funziona e come viene utilizzato nel campo della produzione di bobine e avvolgimenti:

-Preparazione del nucleo: Prima di avviare il processo di avvolgimento, il nucleo della bobina viene preparato. Questo può essere un componente fisso, come un cilindro di plastica o metallo, o un nucleo mobile che viene fatto ruotare durante il processo di avvolgimento. Nel mio caso è un cilindro fisso realizzato tramite stampa in 3D.

-Montaggio del filo: Il filo conduttore viene montato sulla macchina e posizionato in modo che possa essere avvolto intorno al nucleo.

-Avvolgimento del filo: Una volta che tutto è pronto, il processo di avvolgimento inizia. Il motore o il meccanismo di alimentazione sposta il filo intorno al nucleo secondo un modello specifico, che può variare in base alle esigenze della bobina da creare. Il numero di avvolgimenti, il passo e la tensione del filo sono controllati per garantire che la bobina finale abbia le specifiche desiderate.

Fasi del progetto:

-Analisi dei requisiti: In questa fase saranno identificati e analizzati i requisiti dell'avvolgitore di bobine, tenendo conto delle specifiche dell'applicazione desiderata. Ciò include il tipo di bobina da avvolgere, il diametro del filo, la velocità di avvolgimento, la precisione richiesta e altre caratteristiche importanti. Saranno anche considerati i vincoli di costo e di materiali.

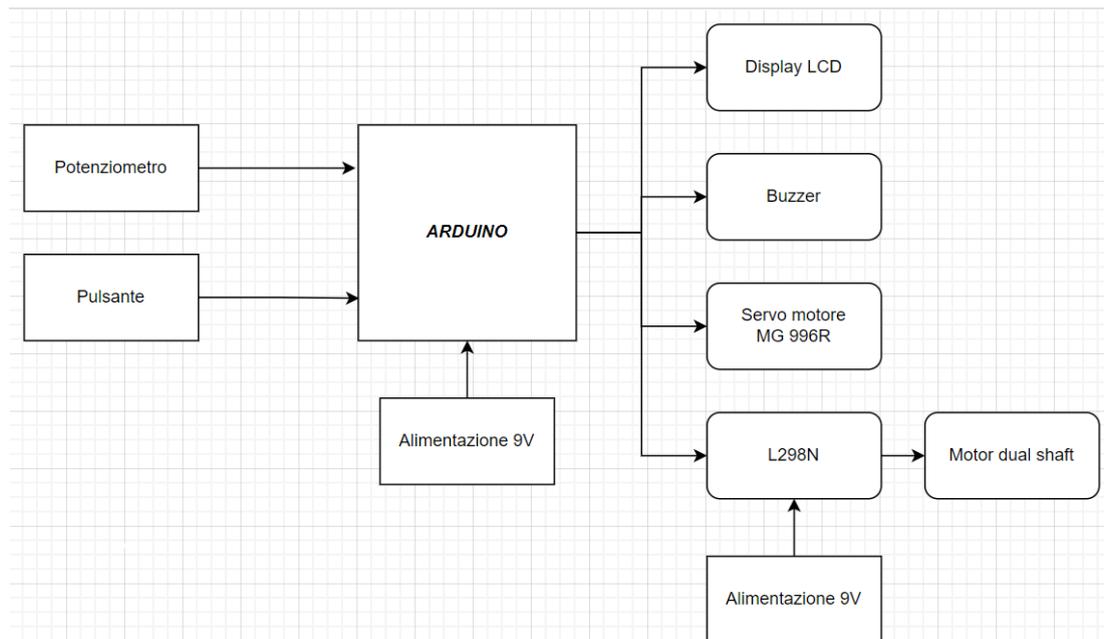
-Progettazione meccanica: Sarà progettato il sistema meccanico dell'avvolgitore di bobine. Questo comprende la struttura, i supporti per bobine, i meccanismi di trascinamento del filo, i motori e altri componenti necessari. La progettazione dovrà garantire la stabilità, la precisione e la sicurezza del dispositivo.

-Progettazione elettronica: Sarà progettato il sistema elettronico di controllo dell'avvolgitore di bobine che l'utente potrà vedere attraverso un display. Ciò include la selezione e l'integrazione di un sensore di hall, che è un sensore magnetico, in modo da rilevare il numero effettivo di giri che l'avvolgitore sta eseguendo per il monitoraggio della velocità, della tensione e di altre variabili, nonché la progettazione di un'interfaccia utente intuitiva per il controllo del dispositivo. Verranno utilizzati un arduino UNO R3 per controllare e coordinare il funzionamento dell'avvolgitore.

-Assemblaggio e collaudo: Saranno acquisiti i componenti necessari e l'avvolgitore di bobine verrà assemblato seguendo i progetti elettronici e meccanici sviluppati. Successivamente, il dispositivo sarà sottoposto a test e collaudi per verificarne il corretto funzionamento e la precisione dell'avvolgimento.

-Ottimizzazione e miglioramento: Il vantaggio di questo progetto è appunto quello di poterlo modificare in base alle richieste del cliente per soddisfare requisiti aggiuntivi. Ciò potrebbe coinvolgere l'aggiornamento dei componenti, l'ottimizzazione dei parametri di controllo o l'implementazione di nuove funzionalità.

Schema a blocchi del sistema:

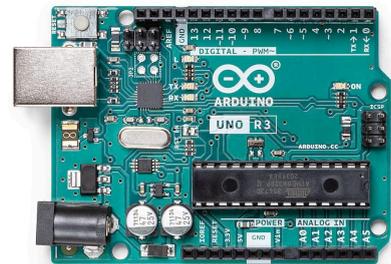


Componenti e materiali utilizzati:

Per realizzare tale esperienza ho utilizzato i seguenti componenti:

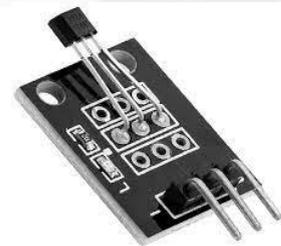
-Arduino Uno R3:

Esso rappresenta la parte intelligente del mio progetto che servirà per coordinare gli altri componenti per lavorare insieme al fine di realizzare gli obiettivi del progetto.



-Hall sensor:

É un dispositivo che rileva la presenza e la misura di un campo magnetico. Funziona sfruttando l'effetto Hall, che produce una tensione quando un materiale conduttore attraversato da corrente è esposto a un campo magnetico. Il sensore di hall è stato utilizzato per contare il numero dei giri della bobina. Esso rileva il numero di giri della bobina grazie a un magnete che è attaccato al motorino che fa girare la bobina.



-Servo motore con rotazione 180°:

Esso è utilizzato per spostare il filo da avvolgere da destra a sinistra della bobina e viceversa. Svolge una funzione simile a quella dei tergicristalli della macchina.



-LCD I2C Display (Liquid Crystal Display):

É uno schermo utilizzato per la visualizzazione grafica. Funziona utilizzando cristalli liquidi che, quando sollecitati da un campo elettrico, modificano la loro opacità per bloccare o permettere il passaggio della luce, creando così le immagini visibili. Esso è utilizzato per mostrare all'utente i dati che sta selezionando. Mostrerà il tempo necessario per l'avvolgimento e tramite un potenziometro sarà possibile all'utente osservare quanti avvolgimenti dovrà eseguire.



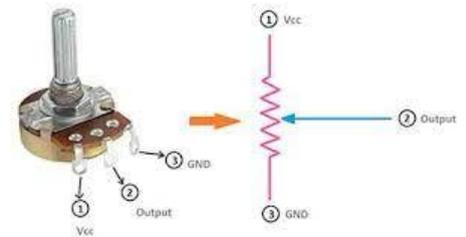
-Cavi maschio femmina e femmina femmina:

Quest'ultimi sono stati utilizzati per collegare tutti gli apparati ad Arduino.



-Potenziometro 10kΩ:

Esso è stato utilizzato per poter far regolare all'utente la velocità di esecuzione dell'operazione e il numero di avvolgimenti da eseguire.



-Pulsante:

Il pulsante viene utilizzato per confermare l'input dato tramite il potenziometro. Esso è collegato ad Arduino tramite una resistenza di pull-up. Invece



-Dual shaft motor:

Esso è un tipo di motore che presenta due alberi di uscita anziché uno solo ed è stato utilizzato per far ruotare la bobina stampata in 3D in modo che il filo possa avvolgersi su di essa correttamente, inoltre fa ruotare anche il magnete per permettere al sensore di hall di contare quanti giri ha eseguito dall'inizio del programma.



-Buzzer:

É un dispositivo che serve per emanare un suono acustico. Esso è stato utilizzato per avvisare quando il processo di ribobinatura inizia e quando si ferma, emanando un breve suono.



-Copper coil:

Rappresenta la bobina iniziale, la fonte dalla quale saranno realizzate delle bobine più piccole.



-Li-Po battery ricaricabili da 9V 800mAh:

Esse sono state utilizzate per alimentare la scheda di arduino e il motore dual shaft in modo da avere una alimentazione costante che permetta di realizzare le operazioni senza troppi problemi.



-Modulo L298N:

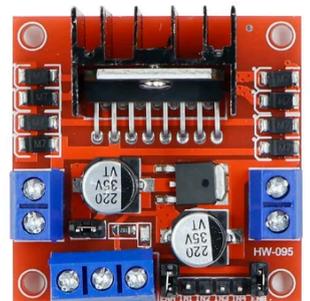
È un modulo driver per motori basato sul chip L298, utilizzato per controllare il movimento di motori a corrente continua (DC) e motori passo-passo. Il modulo è capace di gestire tensioni di alimentazione fino a 46V e correnti fino a 2A per canale. Ecco una sintesi delle sue caratteristiche e funzioni principali:

-Controllo bidirezionale: Può controllare la direzione di rotazione del motore, permettendo sia il movimento avanti che indietro.

-Controllo della velocità: Utilizza il controllo PWM (Pulse Width Modulation) per regolare la velocità del motore.

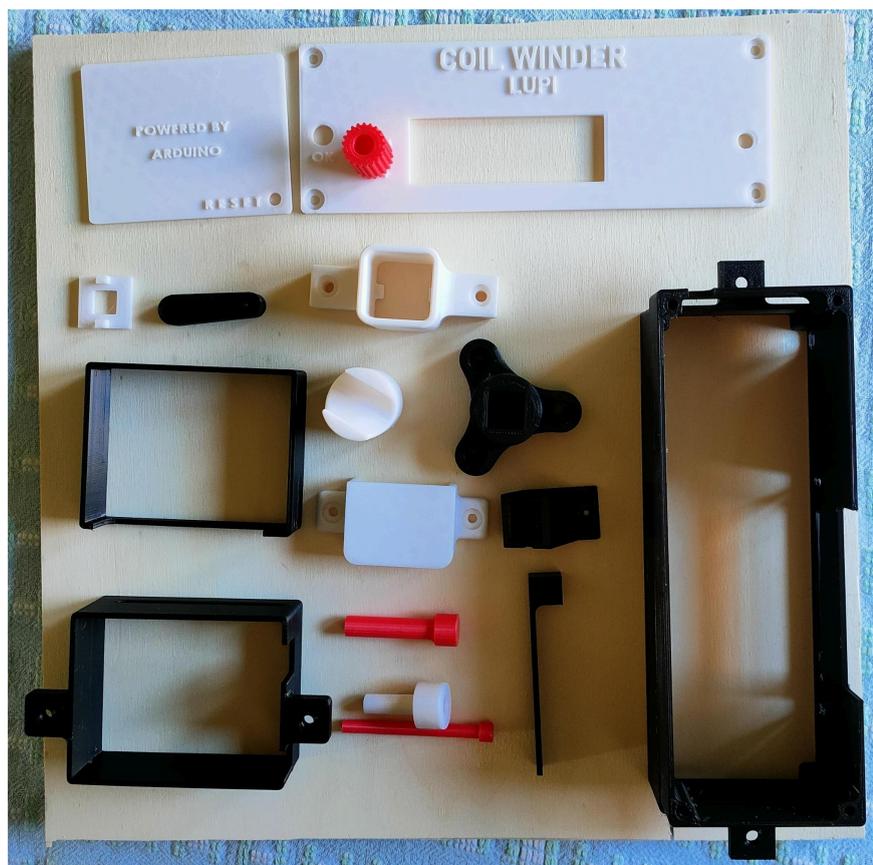
-Supporto per due motori: Può controllare due motori DC indipendentemente, oppure un motore passo-passo bipolare.

-Protezione termica e contro sovraccarichi: Include protezioni per evitare danni dovuti a surriscaldamento o sovraccarico di corrente.



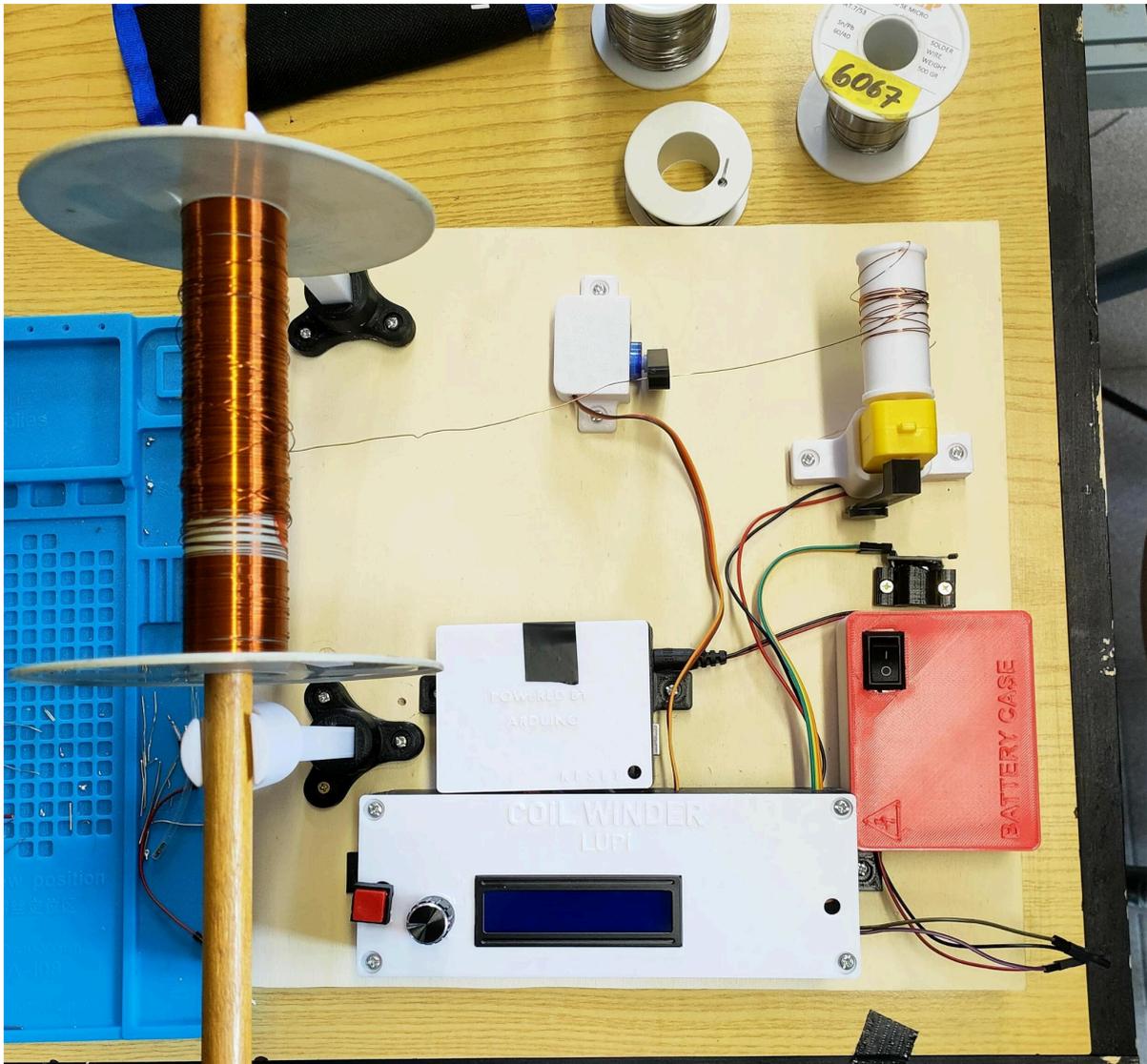
Inoltre per la realizzazione del progetto ho intagliato una base di legno da 30cmx30cm e ho dovuto realizzare delle stampe 3D in PLA per realizzare la struttura complessiva. Ecco qui un'immagine dei seguenti materiali:

Qui si può osservare la base di legno intagliata 30cmx30cm e i materiali realizzati dalla stampa 3D che serviranno per montare i vari componenti necessari per la realizzazione del progetto.



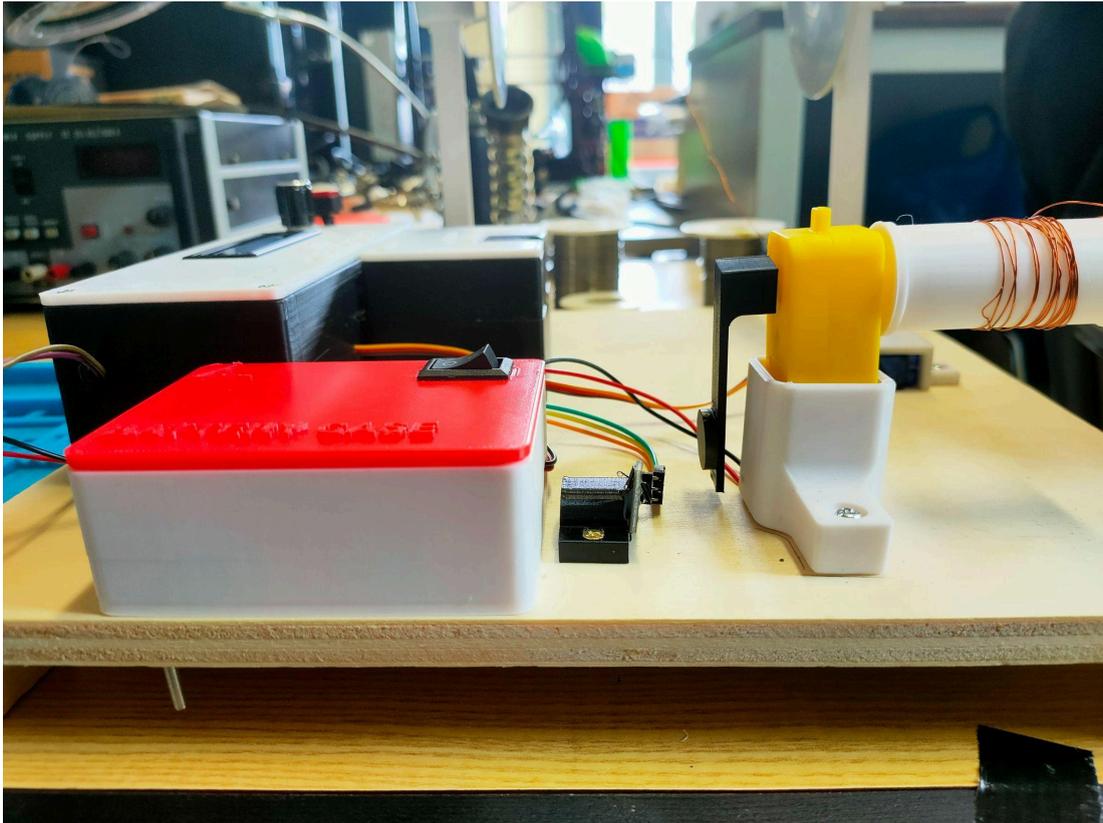
Realizzazione

Durante la realizzazione del progetto ho dovuto utilizzare delle viti m3 con i relativi bulloni per consolidare la struttura e delle viti autofilettanti per saldarla alla base di legno. Inoltre ho utilizzato la colla a caldo per incollare i componenti ai relativi supporti realizzati in stampa 3D e ho realizzato dei supporti in legno per mantenere la struttura stabile.



Da quest' immagine possiamo notare tutta la struttura realizzata: All'interno della scocca sono presenti tutti i collegamenti dei componenti ad arduino e il relativo modulo L298N per regolare la velocità del motore. All'interno del battery case sono presenti due batterie ricaricabili da 9V, una delle quali è stata saldata all'interruttore per alimentare il progetto e quindi arduino, mentre l'altra batteria alimenta il modulo L298N per far muovere il motore.

Facciamo ora uno zoom sulla parte fondamentale del progetto:



In questa immagine possiamo notare il sensore di hall e il magnete che rivestono un ruolo fondamentale per il corretto funzionamento di questo progetto. Il sensore di hall è stato incollato con la colla a caldo su un supporto realizzato in PLA mentre il magnete è stato incollato con dell'attak all'asta che gira nel momento in cui si aziona il motore:

Come funziona? Praticamente una volta che l'utente decide il peso della bobina il programma, dopo un countdown che l'utente può interrompere, si avvia facendo ruotare il motore, quindi il magnete ruota di conseguenza e il sensore di hall, contando i giri compiuti dal magnete, permette al programma di capire quando la bobina ha raggiunto il peso selezionato dall'utente. Una volta che il sensore di Hall dice al programma che è stato raggiunto il numero di avvolgimenti esso si interrompe e la nostra ribobinatrice sarà pronta ad accogliere una nuova istruzione.

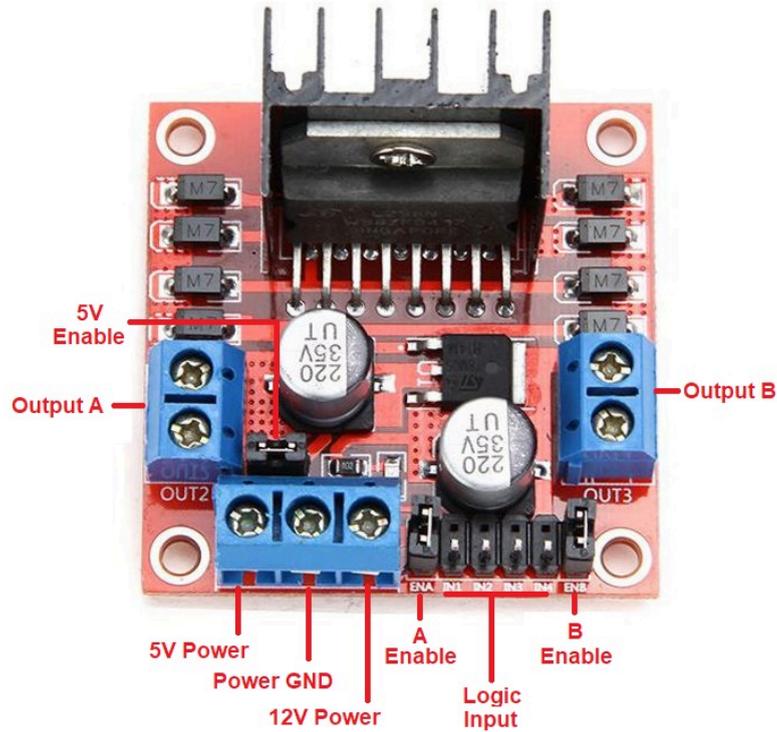
Schema di Collegamento

Arduino Uno:

- Collegare il driver L298N ai pin digitali di Arduino per controllare il motore (IN1, IN2, ENA).
- Collegare il sensore di Hall ai pin digitali di Arduino per rilevare i giri del motore.
- Collegare il display LCD ai pin analogici di Arduino per visualizzare informazioni.
- Collegare i pulsanti ai pin digitali per avviare, fermare e resettare il conteggio.
- Collegare il potenziometro ai pin analogici per regolare la velocità.

Motore DC:

- Connettere il motore al driver L298N.



-
- Assicurarsi che l'albero sia collegato al meccanismo di avvolgimento del filo.

Sensore di Hall:

- Posizionare il sensore in modo che possa rilevare i magneti sull'albero del motore per contare i giri.

Display LCD:

- Collegare il display LCD all'Arduino utilizzando i pin di comunicazione (GND, Vcc, SCL, SDA).

TEST E DEBUG:

Successivamente per la realizzazione di tale progetto è stato anche necessario eseguire delle prove per accertarsi che i componenti acquistati funzionassero completamente.

Ora siamo giunti alla parte più importante del progetto, al cuore del nostro lavoro, ovvero la programmazione del software. Esso è stato realizzato utilizzando arduino ed il codice è il seguente:

```
// Matteo Lupi   Coil Winder   SAROB
// Commento: Questo programma controlla un avvolgitore di bobine che utilizza un display LCD,
//un servo motore, un sensore di hall e un potenziometro.
// Commento: Include le librerie necessarie per l'utilizzo del display LCD e dei servomotori.
#include <LiquidCrystal_I2C.h>// Libreria per utilizzare il display LCD
#include <Servo.h>// Libreria per utilizzare i servomotori

//buzzer pin
int buzzerPin = 7;

//servo
Servo myServo;
int servoPin = 9;
int servoPos = 90;
int servoAdd = 1;

//hall pin
volatile int hallPin = 2;
volatile int hallState;

// "ok" push button
int okButton = 10;
int okButtonState;

//potentiometer pin
int potPin = A0;
int potState;

// Pin per il motore dual shaft + modulo L298N
int motor1pin1 = 4;
int motor1pin2 = 3;

int EN_A = 6; // Pin per controllo PWM sul MOTORE

int totalWinding;// Numero totale di avvolgimenti
volatile int windCount = 0;// Contatore degli avvolgimenti
int mappedPotState; // Valore mappato dello stato del potenziometro

LiquidCrystal_I2C lcd(0x3F, 16, 2);// Inizializzazione del display LCD

void setup() {
  Serial.begin(9600);// Inizializza la comunicazione seriale
  //the usual stuff
  pinMode(okButton, INPUT_PULLUP);// Imposta il pulsante "OK" come input
  pinMode(hallPin, INPUT_PULLUP);// Imposta il pin del sensore di Hall come input
  pinMode(buzzerPin, OUTPUT);// Imposta il pin del buzzer come output
```

```

pinMode(buzzerPin, OUTPUT);// Imposta il pin del buzzer come output
pinMode(motor1pin1, OUTPUT); // Imposta il pin del motore come output
pinMode(motor1pin2, OUTPUT); // Imposta il pin del motore come output
myServo.attach(servoPin); // Collega il servo al pin specificato

okButtonState = digitalRead(okButton);// Legge lo stato del pulsante "OK"

//lcd related stuff
lcd.init();
lcd.backlight();

introFunction();// Esegue la funzione di introduzione
promptFunction(); // Esegue la funzione di prompt
promptFunction();// Esegue la funzione di prompt

lcd.clear();// Pulisce il display
windCalc();// Calcola il numero totale di avvolgimenti
lcd.clear();// Pulisce il display
countDown();// Esegue la funzione di conto alla rovescia
lcd.clear();// Pulisce il display
attachInterrupt(digitalPinToInterrupt(hallPin), windCounter, FALLING); // Attacca l'interruzione al sensore di Hall

tone(buzzerPin, 1000, 500);

//motorSpinFunction();
}

void loop() {
  hallState = digitalRead(hallPin);// Legge lo stato del sensore di Hall
  Serial.println(hallState); // Stampa lo stato del sensore di Hall sulla porta seriale
  //lcd.clear();
  lcd.setCursor(0, 0);// Imposta il cursore LCD sulla riga 0
  lcd.print("Windings: ");// Stampa il numero di avvolgimenti sul display LCD
  lcd.print(windCount);// Stampa il numero di avvolgimenti sul display LCD
  // Controllo del movimento del servo
  if (servoPos == 135 || servoPos == 40) {
    servoAdd = -servoAdd;// Inverte la direzione del movimento
  }
  servoPos = servoPos + servoAdd;// Aggiorna la posizione del servo
  myServo.write(servoPos);// Comanda il servo a muoversi nella posizione desiderata

  motorSpinFunction(EN_A, 100); // fa rotoreare il motore in senso orario alla velocità di 255 (val pwm da 0-255)

  // Controlla se il numero totale di avvolgimenti è stato raggiunto
  while (windCount >= totalWinding) {
    //digitalWrite (motorPin,0);

```

```

while (windCount >= totalWinding) {
  //digitalWrite (motorPin,0);
  motorSpinOff(EN_A, 0); // spegne motore dual shaft con velocità 0
  tone(buzzerPin, 1000, 500);// Attiva il buzzer

  switchOnFunction();// Esegue la funzione di accensione
  lcd.clear();// Pulisce il display
  lcd.setCursor(0, 0);// Imposta il cursore LCD sulla riga 0
  lcd.print("Winding Complete");// Stampa "Winding Complete" sul display LCD
  lcd.setCursor(0, 1);// Imposta il cursore LCD sulla riga 1
  lcd.print("Reset --> restart");// Stampa "Reset --> restart" sul display LCD
  delay(1000);// Attendi
  //myServo.write (0);
}
}

void motorSpinFunction(int pin, int speed){
  analogWrite(pin, speed);

  digitalWrite(motor1pin1, HIGH);
  digitalWrite(motor1pin2, LOW);

  digitalWrite(motor1pin1, LOW);
  digitalWrite(motor1pin2, HIGH);
}

void motorSpinOff(int pin, int speed){
  analogWrite(pin, speed);

  digitalWrite(motor1pin1, LOW);
  digitalWrite(motor1pin2, LOW);
}

// Funzione per mostrare il prompt sul display LCD
void promptFunction() {
  lcd.setCursor(0, 0);// Imposta il cursore LCD sulla riga 0
  lcd.print("Enter Weight: "); // Stampa "Enter Weight: " sul display LCD
  lcd.setCursor(0, 1);// Imposta il cursore LCD sulla riga 1
  lcd.print("Ok to confirm");// Stampa "Ok to confirm" sul display LCD
  delay(1000);// Attendi
}

// Funzione di introduzione
void introFunction() {
  lcd.setCursor(0, 0);// Imposta il cursore LCD sulla riga 0
  lcd.print("Coil Winder");// Stampa "Coil Winder" sul display LCD
  lcd.setCursor(0, 1);// Imposta il cursore LCD sulla riga 1
}

```

```

// Funzione per il conto alla rovescia
void countdown() {

  lcd.print("Starting in "); // Stampa "Starting in " sul display LCD
  lcd.setCursor(0, 0); // Imposta il cursore LCD sulla riga 0
  lcd.setCursor(0, 1); // Imposta il cursore LCD sulla riga 1
  lcd.print("Reset -> Cancel"); // Stampa "Reset -> Cancel" sul display LCD
  for (int i = 5; i >= 0; i--) { // Conto alla rovescia da 5 a 0
    lcd.setCursor(13, 0); // Imposta il cursore LCD sulla colonna 13 e sulla riga 0
    lcd.print(i); // Stampa il contatore sul display LCD
    delay(1000); // Attendi un secondo
  }
}

// Interruzione per contare gli avvolgimenti
volatile int windCounter() {
  hallState = digitalRead(hallPin); // Legge lo stato del sensore di Hall
  if (hallState == LOW) { // Se il sensore di Hall rileva un avvolgimento
    windCount = windCount + 1; // Incrementa il contatore degli avvolgimenti
    detachInterrupt(hallPin); // Disattiva l'interruzione
  }
}

// Funzione per accendere il motore
void switchOnFunction() {
  myServo.write(150); // Ruota il servo a 150 gradi
  delay(1000); // Attendi un secondo
  myServo.write(90); // Riporta il servo alla posizione iniziale (90 gradi)
}

```

Dopo aver realizzato il codice ho seguito questi punti:

- Verificare i collegamenti:** Assicurarsi che tutti i collegamenti elettrici siano corretti.
- Caricare il codice:** Caricare il codice sull'Arduino e monitorare il funzionamento del sistema.
- Regolazioni:** Effettuare le regolazioni necessarie per la velocità del motore e la sensibilità del sensore di Hall.
- Validazione:** Testare la ribobinatrice con diversi tipi di fili e nuclei per garantire che funzioni correttamente.

Conclusione

Posso concludere che l'esperienza si è svolta al termine correttamente. Ho riscontrato una serie di intoppi durante la realizzazione di tale progetto, però sono stato in grado di risolverli. Questa è solamente la prima versione del progetto e naturalmente può essere sviluppato ulteriormente. Nella prossima versione prevedo di realizzare un'app dalla quale si potrà selezionare il numero di avvolgimenti e il materiale da utilizzare.