

RELAZIONE SULLA REGOLAZIONE VELOCITA' DI UN MOTORE CC 775

OBIETTIVO

Regolare la velocità di rotazione di un motore CC di tipo 775 tramite Arduino.

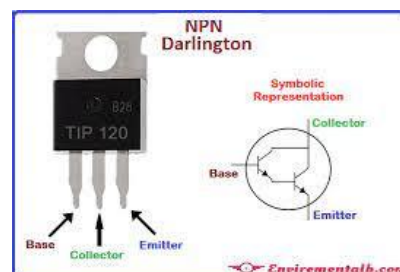
DESCRIZIONE SCELTE PROGETTUALI

Il motore 775 è un motore elettrico da 12V-24V con doppio cuscinetto a sfere con un numero di giri variabile fra 2000-7000Rpm. E' caratterizzato da elevata coppia, basso rumore ed elevate affidabilità. Ideale per l'impiego in applicazioni di automazione industrial in bassa tensione.

La corrente assorbita senza carico è di circa 0.27A mentre sotto massimo sforzo si ha assorbimento di 3.25A.



+



Per il controllo della velocità del motore è necessario l'impiego di un transistor di potenza come ad esempio il classico transistor di potenza TIP120 Darlington NPN.

Può pilotare carichi fino a 60 V con una corrente di picco di 8 A e una corrente continua di 5 A.

Questo lo rende adatto al controllo del motore CC 775 che assorbe al massimo 3.25A.

Electrical Characteristics* $T_a=25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
$V_{CE(sus)}$	Collector-Emitter Sustaining Voltage : TIP120 : TIP121 : TIP122	$I_C = 100\text{mA}, I_B = 0$	60 80 100			V V V
I_{CEO}	Collector Cut-off Current : TIP120 : TIP121 : TIP122	$V_{CE} = 30\text{V}, I_B = 0$ $V_{CE} = 40\text{V}, I_B = 0$ $V_{CE} = 50\text{V}, I_B = 0$			0.5 0.5 0.5	mA mA mA
I_{CBO}	Collector Cut-off Current : TIP120 : TIP121 : TIP122	$V_{CB} = 60\text{V}, I_E = 0$ $V_{CB} = 80\text{V}, I_E = 0$ $V_{CB} = 100\text{V}, I_E = 0$			0.2 0.2 0.2	mA mA mA
I_{EBO}	Emitter Cut-off Current	$V_{BE} = 5\text{V}, I_C = 0$			2	mA
h_{FE}	* DC Current Gain	$V_{CE} = 3\text{V}, I_C = 0.5\text{A}$ $V_{CE} = 3\text{V}, I_C = 3\text{A}$	1000 1000			
$V_{CE(sat)}$	* Collector-Emitter Saturation Voltage	$I_C = 3\text{A}, I_B = 12\text{mA}$ $I_C = 5\text{A}, I_B = 20\text{mA}$			2.0 4.0	V V
$V_{BE(on)}$	* Base-Emitter On Voltage	$V_{CE} = 3\text{V}, I_C = 3\text{A}$			2.5	V
C_{ob}	Output Capacitance	$V_{CB} = 10\text{V}, I_E = 0, f = 0.1\text{MHz}$			200	pF

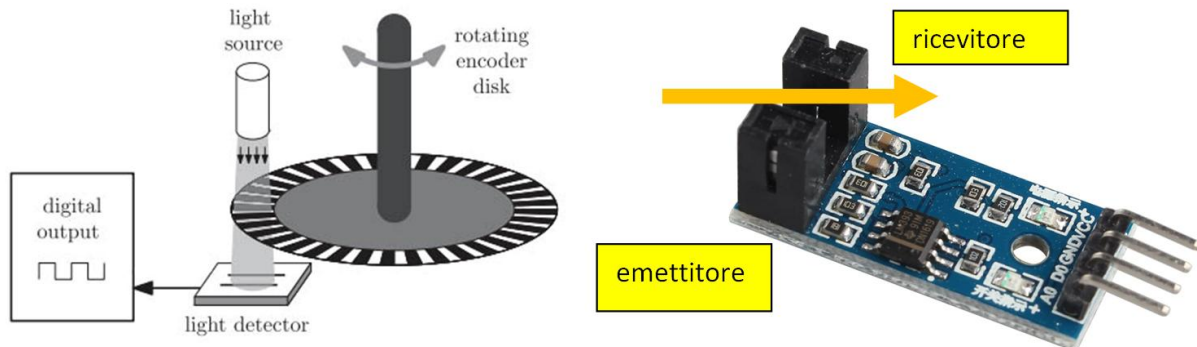
* Pulse Test: Pulse Width \leq 300 μ s, Duty Cycle \leq 2%

Per poter variare il numero di giri del motore verrà utilizzata la tecnica PWM che ha il vantaggio di mantenere elevata la coppia motrice anche ad un basso regime di rotazione del motore. Se variassimo solo la tensione di alimentazione anche la coppia motrice calerebbe di conseguenza.

Per rilevare il numero di giri del motore verrà utilizzato lo shield IR LM393 per Arduino accoppiato ad un encoder incrementale.

L'encoder è costituito da un semplice disco forato poiché ci volgiamo rilevare la velocità e non la posizione angolare.

Lo shield LM393 comprende un sensore ad infrarossi che fornisce un impulso ogni qualvolta viene interrotto il raggio luminoso fra emettitore e ricevitore.



NOTA:

Visto l'elevato numero di giri del motore (oltre 6000 rpm) è stato necessario ridurre il numero di fori presenti sul disco a 3. Con troppi fori la rilevazione della velocità diventa imprecisa.

SIMULAZIONE DEL CIRCUITO DI REGOLAZIONE

La simulazione del circuito di controllo del motore è stata fatta su Thinkercad utilizzando un generatore di impulsi per simulare lo shield LM393. Il generatore va impostato in modalità onda quadra (impulsi).

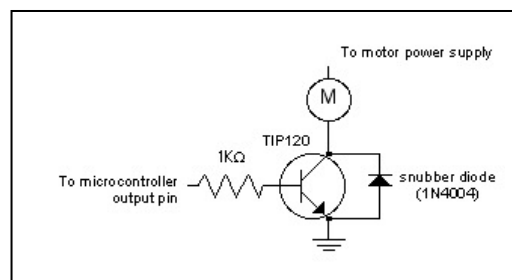
La lettura degli impulsi fornisce l'ipotetica velocità di rotazione del motore che in realtà mostrerà una velocità del tutto differente essendo il motore totalmente sconnesso dal sensore simulato.

Per contare il numero di impulsi che arrivano dal generatore è possibile adottare varie tecniche. La più semplice è quella che fa uso degli INTERRUPT (funzioni interne che non bloccano il LOOP). Per poter utilizzare gli interrupt sulla scheda Arduino UNO è necessario utilizzare il PIN digitale 2.

Il transistor TIP120 è stato collegato nella classica configurazione ad emettitore comune.

La resistenza di base si calcola noto l'assorbimento del motore e le caratteristiche del TIP120 (vedere tabella allegata):

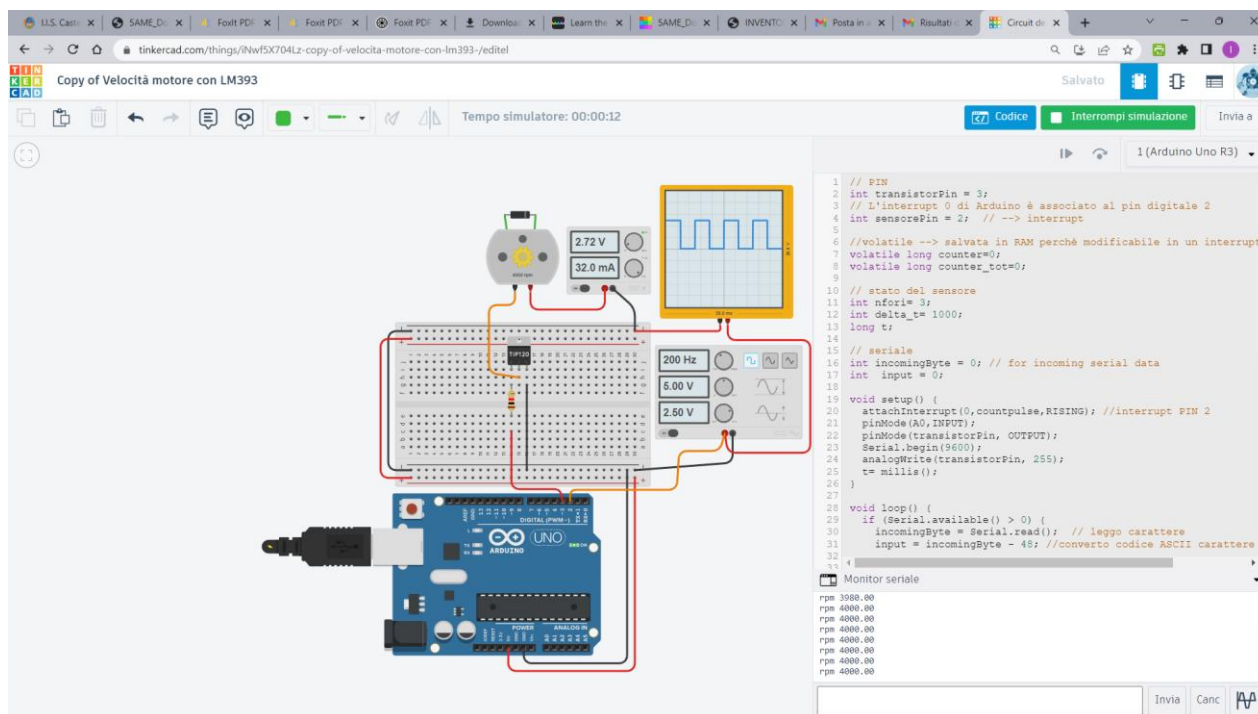
Ic max	3,5 A
hFE	1000
Ib	0,0035 A
VBE max	2,5 V
Vcc	5 V
Rb	714 ohm



Scegliamo come resistenza di base 1K poiché, come regola generale, non conviene superare il 70-80% della capacità di carico del motore che quindi avrà un assorbimento massimo nettamente inferiore ai 3.5°.

La figura sottostante mostra lo schema elettrico utilizzato per controllare il motore e rilevare il numero di giri.

In parallelo al motore va sempre collegato un diodo di protezione per il transistor poiché, in presenza di bobine, ogni volta che si spegne il motore si genera un picco di corrente elettrica che potrebbe danneggiare il transistor.



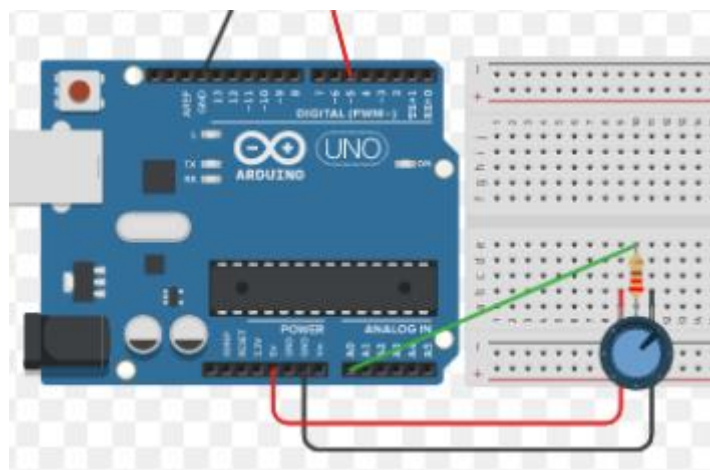
La figura mostra la velocità rilevata contando gli impulsi con una frequenza di 200Hz, cioè 200 impulsi al secondo.

Si hanno quindi $200 \times 60 = 12000$ impulsi al minuto.

Poiché il disco ha 3 fori la velocità rilevata varrà quindi: $Rpm = \text{numero impulsi} / 3 \text{ fori} = 4000 \text{ giri/minuto}$

Nel codice Arduino la variazione della velocità del motore viene effettuata tramite comandi sulla porta seriale con la tecnica PWM (analogWrite).

Per ottenere una variazione continua della velocità si può collegare ad un pin Arduino un potenziometro che fornisce un valore digitale variabile da 0 a 1023 (0V \rightarrow 5V) poiché il convertitore da analogico a digitale di Arduino ha una risoluzione di 10 bit ($2^{10} \rightarrow 1024$).



STUDENTE: _____

CLASSE _____

DATA _____

CODICE ARDUINO

```
// PIN
int transistorPin = 3;
// L'interrupt 0 di Arduino è associato al pin digitale 2
int sensorePin = 2; // --> interrupt

//volatile --> salvata in RAM perchè modificabile in un interrupt
volatile long counter=0;
volatile long counter_tot=0;

// stato del sensore
int nfori= 3;
int delta_t= 1000;
long t;

// seriale
int incomingByte = 0; // for incoming serial data
int input = 0;

void setup() {
  attachInterrupt(0,countpulse,RISING); //interrupt PIN 2
  pinMode(A0,INPUT);
  pinMode(transistorPin, OUTPUT);
  Serial.begin(9600);
  analogWrite(transistorPin, 255);
  t= millis();
}

void loop() {
  if (Serial.available() > 0) {
    incomingByte = Serial.read(); // leggo carattere
    input = incomingByte - 48; //converto codice ASCII carattere in numero 1,2,3

    switch (input) {
      case 0: analogWrite(transistorPin, 0);      break;
      case 1: analogWrite(transistorPin, 255/3); break;
      case 2: analogWrite(transistorPin, 255/2); break;
      case 3: analogWrite(transistorPin, 255);   break;
    }
    input=0;
  }

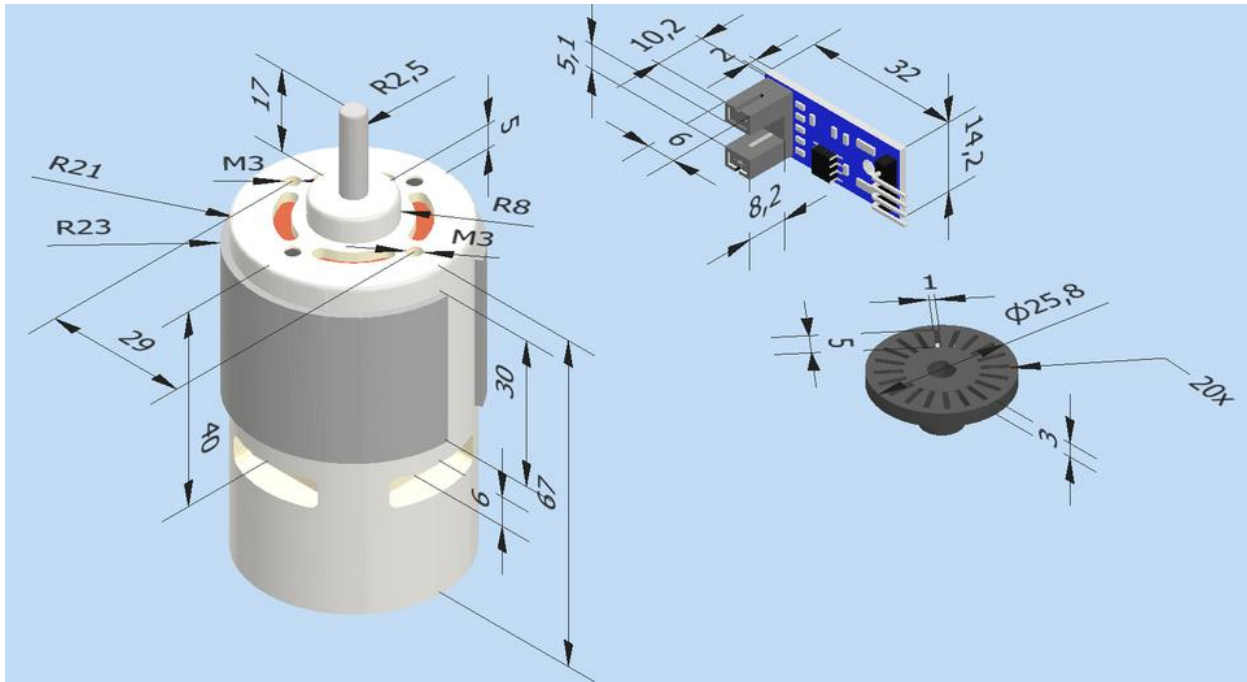
  // durante la stampa non leggo impulsi per non sfalsare calcolo)
  if ((millis() - t)>=delta_t) {
    // rpm con delta_t= 1 sec
    float rpm= counter *20.0; //counter * 60/nfori
    Serial.print("rpm "); Serial.println(rpm);
    counter= 0;
    t= millis();
  }
  //delayMicroseconds(10);
}

void countpulse(){
  counter++;
}
```

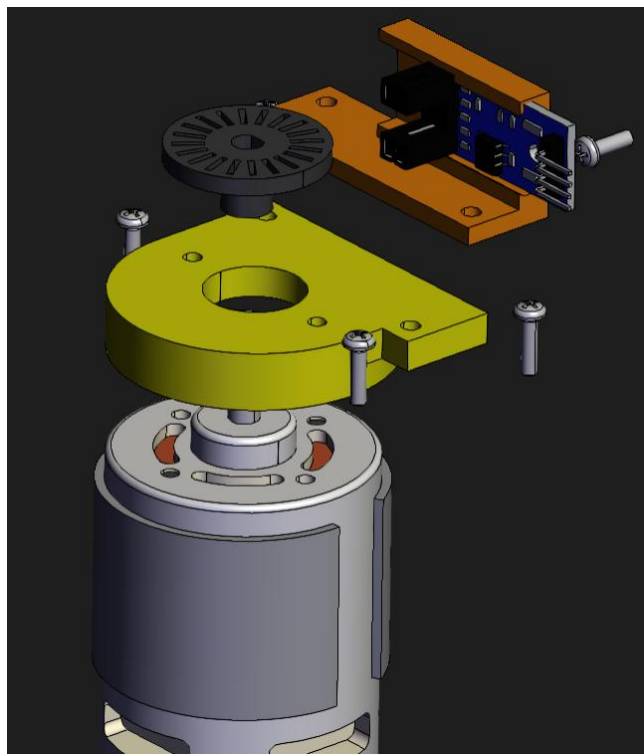
PROGETTO SUPPORTO SENSORE LM393

Per rilevare la velocità del motore è necessario collegare il sensore al motore in modo da rilevare gli impulsi tramite il disco forato che funziona da encoder incrementale.

Poiché il sistema di collegamento verrà stampato in 3D è necessario prestare attenzione alla progettazione affinché si riducano al minimo eventuali supporti necessari alla stampa 3D.



La soluzione proposta suddivide il sistema di blocco del sensore in due parti distinte in modo da ridurre a zero la necessità di supporti in fase di stampa.



Il modello 3D in Inventor è allegato alle relazione.