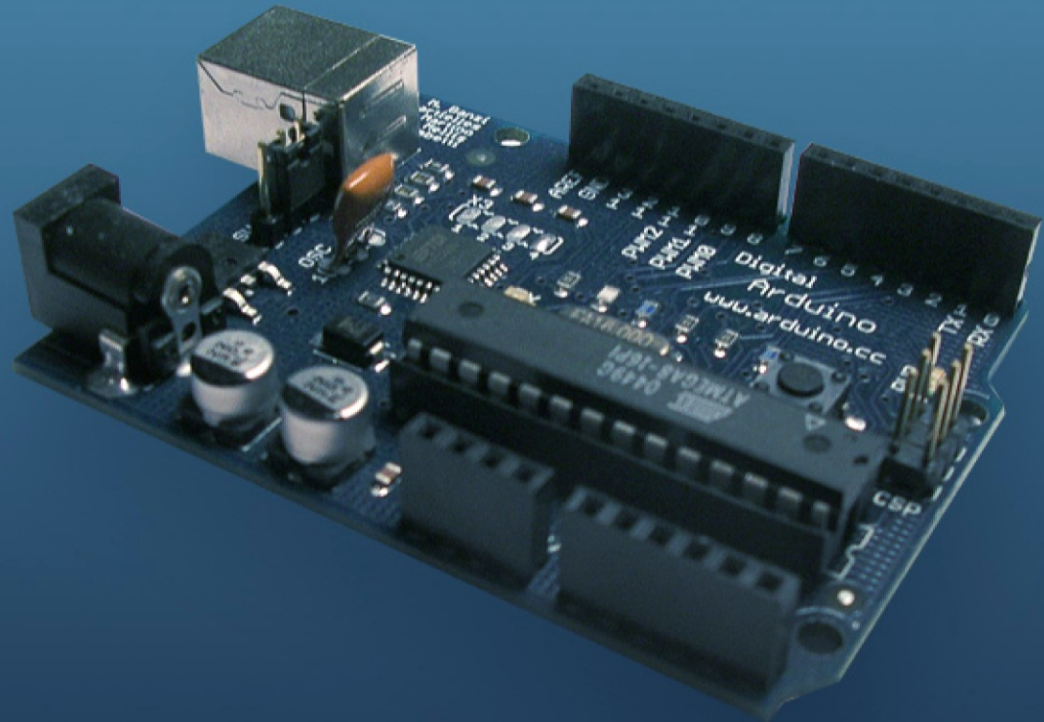


Arduino
Physical Computing I/O board



8[^] parte: Pilotare Motori passo-passo bipolari usando l'integrato L293D



Author: Ing. Sebastiano Giannitto (ITIS "M.BARTOLO" –PACHINO)

In questo tutorial imparerai come controllare un motore passo-passo usando il chip di controllo del motore L293D

Il L293D é un integrato che ha 2 Ponti H.

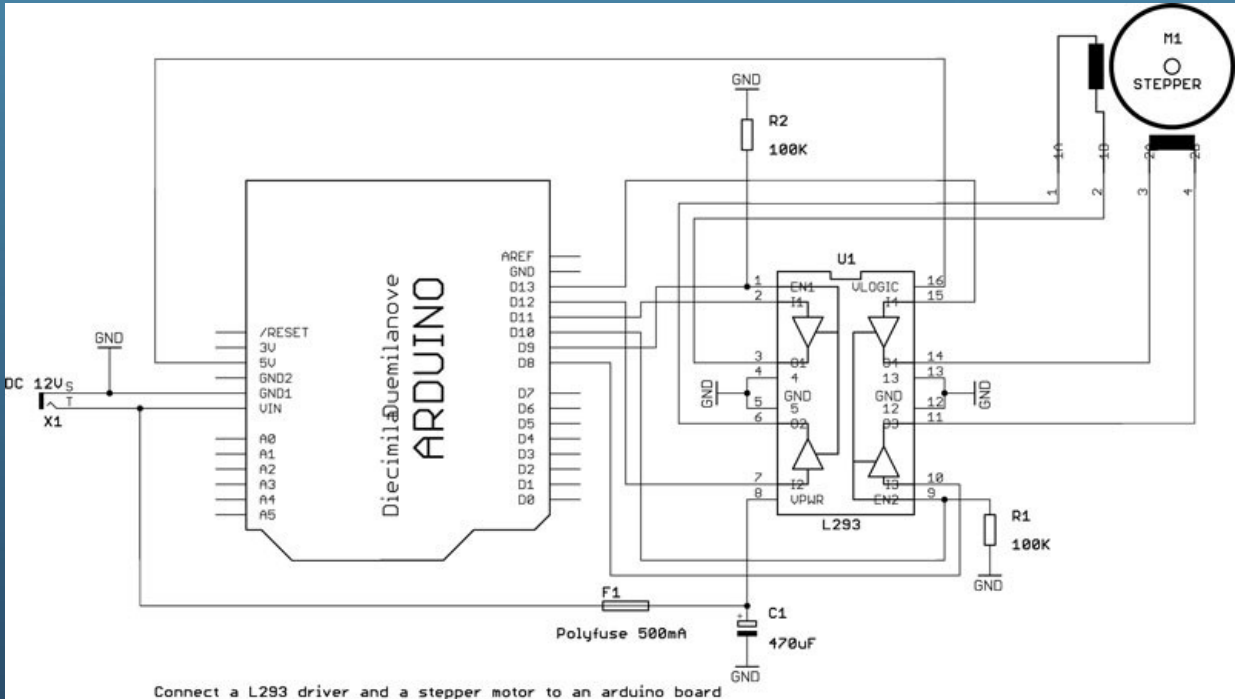
Con quello possiamo pilotare 2 motori in corrente continua in entrambe le direzioni oppure 1 motore passo passo bipolare o 1 motore passo passo unipolare (non collegando le conessioni comuni).

C'è un ingresso PWM per driver in modo da poter controllare la velocità del motore. Funziona a logica 5V. Buono per motori a 3-4,5 V fino a 36 V (chiaramente ad alimentazione separata)

Ricordiamo invece che:

- Il ULN2003 sono 7 transistori darlington con collettore aperto e diodo di protezione verso +alimentazione.
- **ULN2004A è COMPATIBILE con ULN2003**
- Il ULN2803 é la stessa cosa ma ha 8 transistori. Entrambi portano l' uscita a massa e possiamo pilotare dei carichi che sono collegati fissi a +alimentazione.
- Il A3981 o UDN2981 é una cosa simile ma porta l' uscita a +alimentazione.
- Possiamo pilotare dei carichi che sono collegati fissi a massa.
- Con un ULN2003 / 2803 possiamo pilotare dei motori corrente continua in una direzione e dei motori passo passo unipolari.
- Tutti gli integrati non hanno tantissima potenza, al massimo sopportano 500mA (600mA il L293D).
- I motori passo-passo hanno il vantaggio di poter essere posizionati con precisione, spostati in avanti o indietro di un "passo" alla volta, ma possono anche ruotare continuamente.

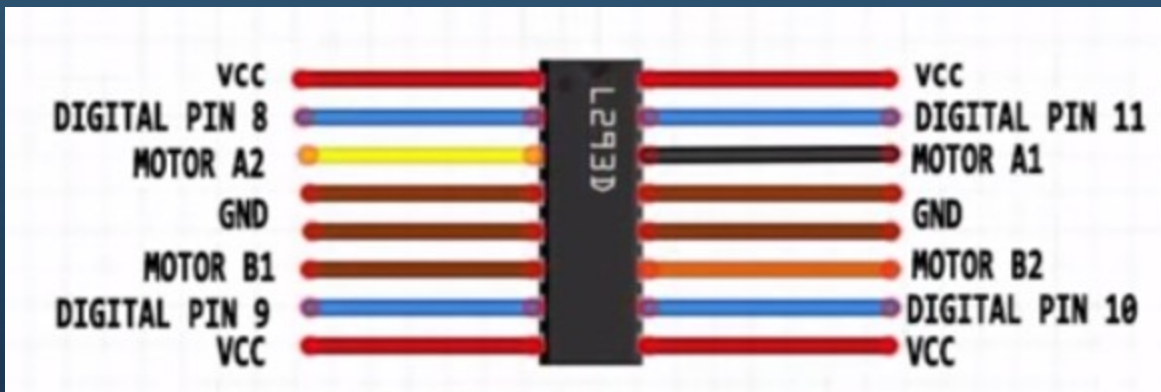
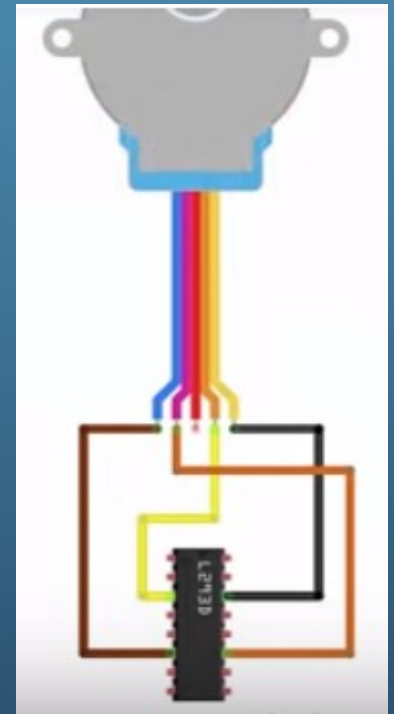
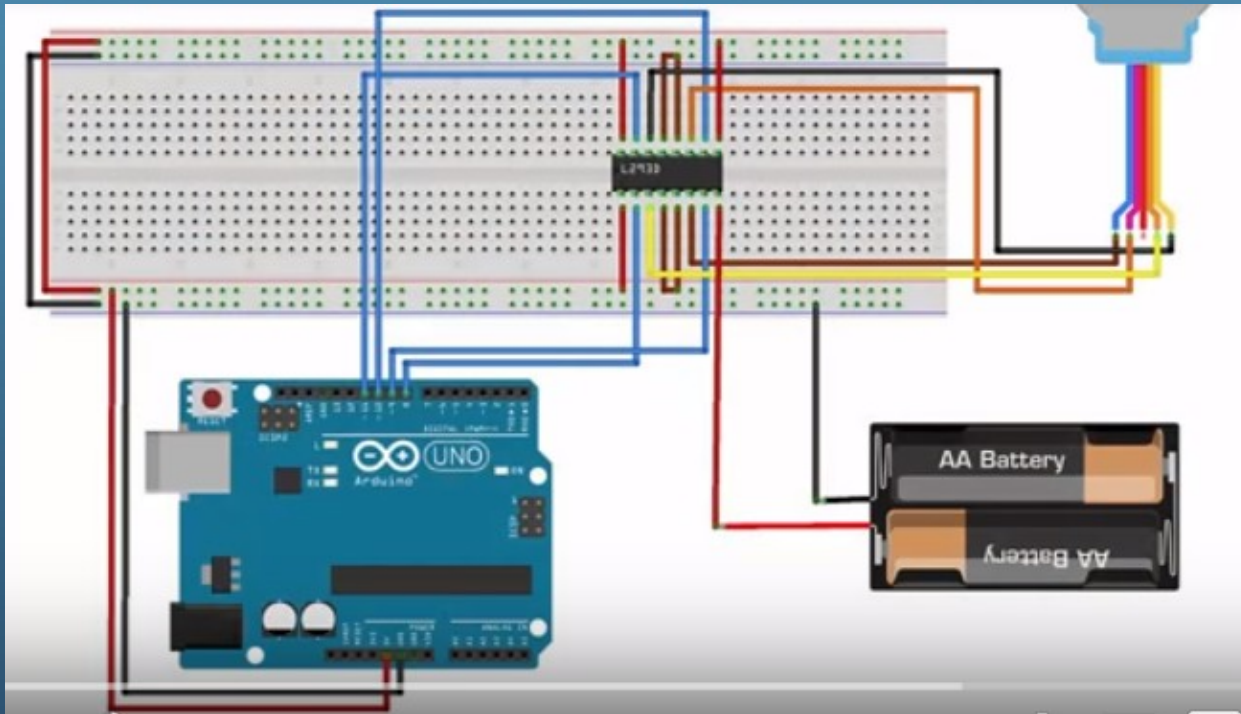
Consultando il sito ufficiale Arduino possiamo trovare uno schema di collegamento.



La guida di un motore stepper bipolare con L293D è molto simile alla guida di un motore stepper unipolare. La sequenza di impulsi è la stessa. L'unica differenza tra guidare un motore stepper unipolare e guidare un motore stepper bipolare è che c'è un filo in più in un motore stepper unipolare che devi collegare.

Per cui proponiamo questo'altro circuito sfruttando lo stepper unipolare a 5 fili a corredo nei kit più comuni

(*) L293N / L293NE / L293DNE sono i integrati a 16 pin.
L293DWP è ha 28 pin. L293N / L293NE non hanno diodi di protezione.
L293DNE / L293DWP invece li ha . La E indica senza piombo.



Notare che il cavo rosso del motore passo-passo non è collegato a nulla.

Il codice

```
int motorPin1 = 8; // impostano la corrispondenza tra la fase del motore e il pin a cui è
int motorPin2 = 9; // collegata su Arduino;
int motorPin3 = 10;
int motorPin4 = 11;
int delayTime = 500; // imposta il ritardo di rotazione, in millisecondi, tra un passo e quello successivo;

void setup()
{
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
  pinMode(motorPin3, OUTPUT);
  pinMode(motorPin4, OUTPUT);
}
```

```
void loop() {
```

```
  digitalWrite(motorPin1, HIGH);  
  digitalWrite(motorPin2, LOW);  
  digitalWrite(motorPin3, LOW);  
  digitalWrite(motorPin4, LOW);  
  delay(delayTime);  
  digitalWrite(motorPin1, LOW);  
  digitalWrite(motorPin2, HIGH);  
  digitalWrite(motorPin3, LOW);  
  digitalWrite(motorPin4, LOW);  
  delay(delayTime);  
  digitalWrite(motorPin1, LOW);  
  digitalWrite(motorPin2, LOW);  
  digitalWrite(motorPin3, HIGH);  
  digitalWrite(motorPin4, LOW);  
  delay(delayTime);  
  digitalWrite(motorPin1, LOW);  
  digitalWrite(motorPin2, LOW);  
  digitalWrite(motorPin3, LOW);  
  digitalWrite(motorPin4, HIGH);  
  delay(delayTime);
```

```
}
```

Le prime 4 linee indicano ad arduino di inviare un segnale alto (HIGH) sul pin 8, quest'ultimo collegato all'ULN2003A, collegherà la fase corrispondente facendo avanzare l'asse del motore, le stesse impostano come segnale basso (LOW) i pin dal 9 al 11, mentre la linea successiva indica di attendere il tempo di ritardo impostato (500 millisecondi) prima di procedere ad eseguire le istruzioni successive.

Le linee che seguono impostano a livello alto in successione i pin 9,10 e 11 creando l'effetto rotazione ed attendendo per ogni passo del motore un tempo di 500 millisecondi prima di procedere oltre.

Essendo le linee inserite in una speciale funzione detta "loop" di arduino il risultato è un ciclo continuo dell'alberino del motore.

Sperimentare cosa accade se modifichi il "delayTime" da 500 ad un valore inferiore o superiore

Guarda i video: https://www.youtube.com/watch?v=z7_orCAeGJQ
https://www.youtube.com/watch?v=3lVefz-kG_M