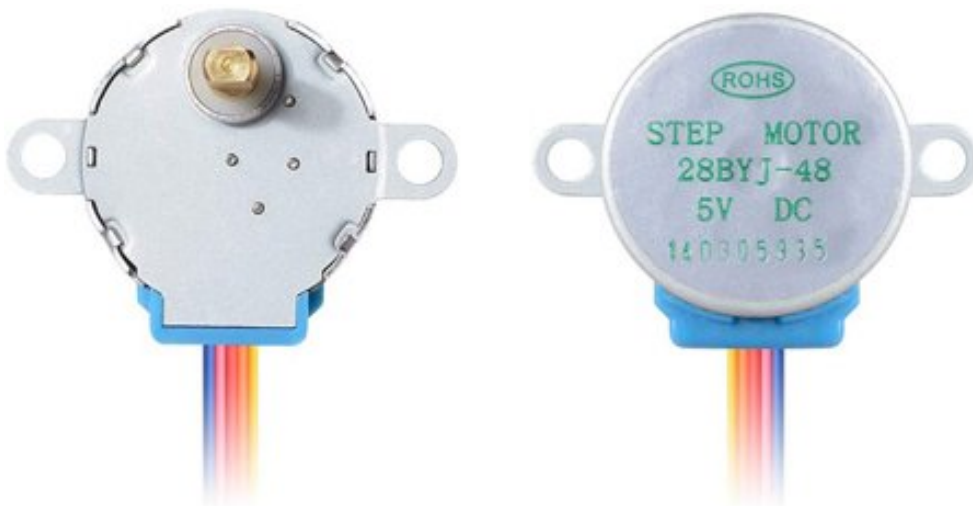


ARDUINO MOTOR STEPPER 28BYJ-48

I motori passo-passo si collocano tra un motore CC convenzionale e un servomotore. Possono ruotare continuamente come i motori DC ed essere posizionati con precisione (a passi discreti) come i servomotori. I motori passo-passo utilizzano una ruota dentata ed elettromagneti per spingere la ruota di un "passo" alla volta. Ogni impulso alto inviato eccita la bobina, attirando i denti più vicini alla ruota dentata e facendo ruotare il motore in incrementi di angolo precisi e fissi noti come passi. Il numero di passi che il motore passo-passo ha in una rotazione di 360 gradi è in realtà il numero di denti sull'ingranaggio. Il modo in cui si pulsano queste bobine determina il funzionamento del motore.

- La sequenza degli impulsi determina la direzione di rotazione del motore.
- La frequenza degli impulsi determina la velocità del motore.
- Il numero di impulsi determina quanto girerà il motore.
- Eccitando le bobine nella sequenza corretta, il motore viene messo in rotazione.

Il 28BYJ-48 è un motore passo-passo unipolare a 5 fili che funziona a 5V. È perfetto per progetti che richiedono un posizionamento preciso



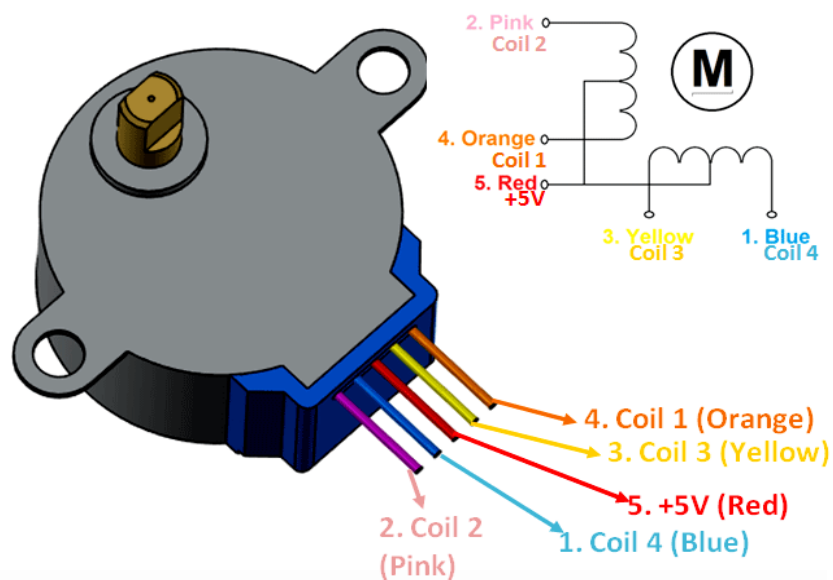
Poiché il motore non utilizza spazzole di contatto, ha un movimento relativamente preciso ed è abbastanza affidabile. Nonostante le sue dimensioni ridotte, il motore eroga una discreta coppia di 34,3 mN, ma ad una velocità di circa 15 giri/min. Fornisce una buona coppia anche da fermo e la mantiene finché il motore riceve potenza. L'unico inconveniente è che assorbe una discreta quantità di corrente e consuma energia anche quando è fermo.

Consumo di energia

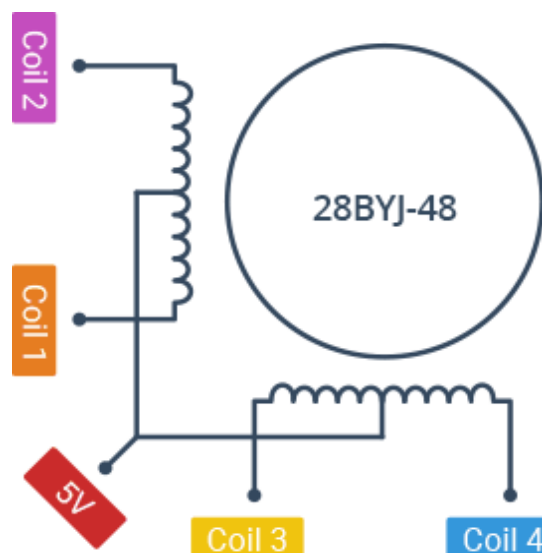
Il 28BYJ-48 assorbe tipicamente circa 240 mA. Poiché il motore consuma una notevole quantità di energia, è preferibile alimentarlo direttamente da un alimentatore esterno a 5 V piuttosto che da Arduino. Vale la pena notare che il motore consuma energia anche quando è fermo per mantenere la sua posizione.

Piedinatura

Il motore passo-passo 28BYJ-48 ha cinque fili. La piedinatura è la seguente:



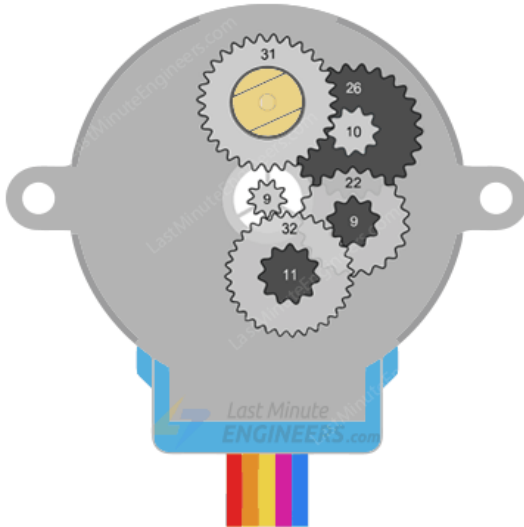
Il 28BYJ-48 ha due bobine, ognuna delle quali ha un collegamento centrale. Queste due prese centrali sono collegate internamente e portate fuori come il 5° filo (filo rosso). Insieme, un'estremità della bobina e la presa centrale formano una fase. Pertanto, 28BYJ-48 ha un totale di quattro fasi.



Il filo rosso è sempre alimentato a livello logico ALTO (+5V), quindi quando uno degli altri fili è portato a livello logico BASSO, la fase è eccitata. Il motore passo-passo ruota solo quando le fasi sono eccitate in una sequenza logica nota come sequenza a gradini.

Rapporto di riduzione dell'ingranaggio

Secondo la scheda tecnica, quando il motore 28BYJ-48 viene azionato in modalità a passo intero, ogni passo corrisponde a una rotazione di 11,25°. Ciò significa che ci sono 32 passi per giro ($360^\circ/11,25^\circ = 32$).



Gear Ratios:

- 32 / 9
- 22 / 11
- 26 / 9
- 31 / 10

Multiplying the gear ratios:

$$\frac{32}{9} \times \frac{22}{11} \times \frac{26}{9} \times \frac{31}{10} = 63.68395$$

Round 63.68395 up: 64

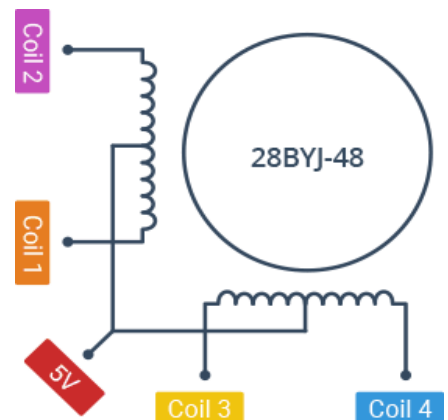
This gives us a 64:1 gear ratio over all

Inoltre, il motore è dotato di un set di riduttori 1/64 (in realtà è 1/63,68395, ma 1/64 è un'approssimazione abbastanza buona per la maggior parte degli scopi).

Ciò significa che ci sono infatti 2038 passi (32×63.68395 passi per giro = 2037.8864 ovvero circa 2038 passi).

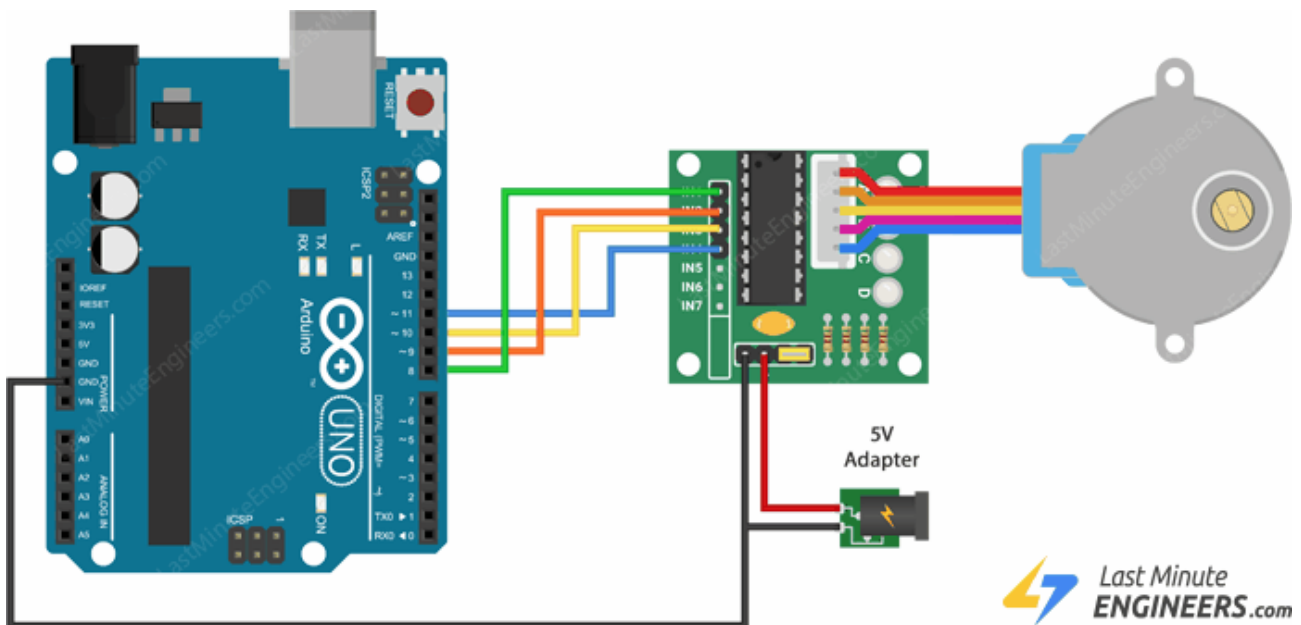
Il filo rosso è sempre tirato ALTO, quindi quando l'altro filo è tirato BASSO, la fase è eccitata.

Il motore passo-passo ruota solo quando le fasi sono eccitate in una sequenza logica nota come sequenza a gradini .



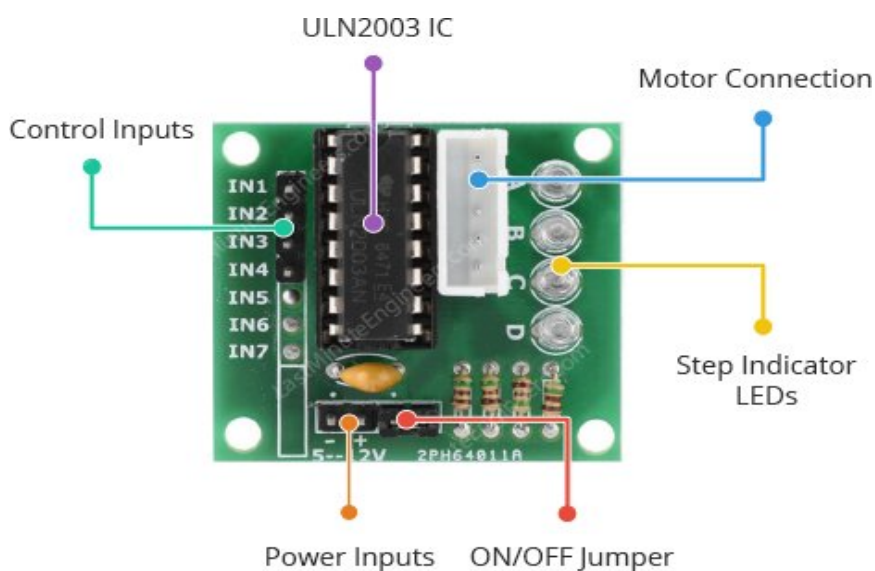
Per la nostra prova utilizzeremo due circuiti: il primo farà uso di un solo motore mentre il secondo ne utilizza due. Scriveremo un codice (sketch) che permetta di far ruotare il primo stepper motor in un senso e il secondo ruoterà nel senso opposto al primo.

Lo schema tipico di connessione per il primo circuito è il seguente:



La scheda driver ULN2003

Poiché il motore passo-passo 28BYJ-48 consuma una notevole quantità di energia, non può essere controllato direttamente Arduino. Per controllare il motore, è richiesto un driver pertanto, questo motore viene generalmente fornito con una scheda driver basata su IC ULN2003. Il circuito integrato ULN2003, noto per la sua capacità di alta corrente e alta tensione, fornisce un guadagno di corrente più elevato rispetto a un singolo transistor e consente all'uscita a bassa tensione e bassa corrente di un microcontrollore di pilotare un motore passo-passo ad alta corrente. L'ULN2003 è costituito da un array di sette coppie di transistor Darlington, ciascuna delle quali può pilotare un carico fino a 500 mA e 50 V.



La scheda ha quattro ingressi di controllo, una uscita motore e una connessione di alimentazione.

IN1-IN4 sono i pin di ingresso del controllo del motore. Collegali ai pin di uscita digitale di Arduino.

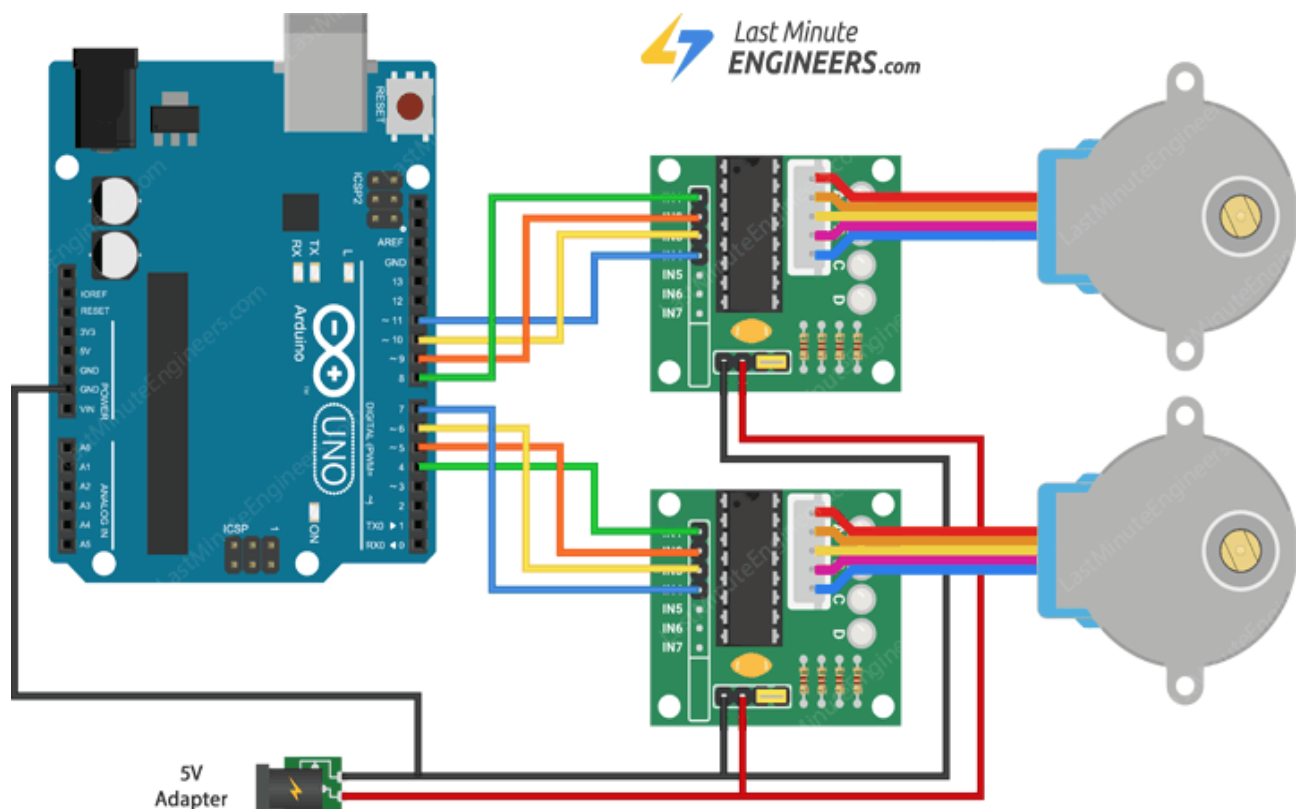
GND è il pin di terra.

VCC è il pin che alimenta il motore. Poiché il motore consuma una notevole quantità di energia, è preferibile utilizzare un alimentatore esterno da 5 V piuttosto che da Arduino.

Connettore motore Qui è dove si collega il motore. Il connettore è codificato, quindi potrà essere inserito solo in un modo.

Per il nostro primo esperimento, potremo utilizzare uno degli esempi forniti con l'IDE di Arduino. La libreria Arduino stepper, gestisce la sequenza di passi e consente di controllare un'ampia gamma di motori passo-passo unipolari e bipolari.

Di seguito lo schema per il nostro secondo esperimento che farà ruotare i motori il primo in senso orario e il secondo in senso antiorario:



Il nostro codice sarà il seguente:

```
/*Controllo di due motori con Arduino
 * 8.5.2023
 * Prof. Francesco Benignetti
 */

//Includiamo la libreria Stepper di Arduino
#include <Stepper.h>

// definizione del numero di passi per rotazione
const int stepsPerRevolution = 2038;

// creazione di due istanze della classe stepper con la definizione dei pin utilizzati
Stepper myStepper_1 (stepsPerRevolution, 4, 6, 5, 7);
Stepper myStepper_2 (stepsPerRevolution, 8, 10, 9, 11);

/* PIN_Motor_1
 * IN1 -> 4
 * IN2 -> 5
 * IN3 -> 6
 * IN4 -> 7
 * PIN_Motor_2
 * IN1 -> 8
 * IN2 -> 9
 * IN3 -> 10
 * IN4 -> 11
 */
void setup() {

  /*
   * i pin di Arduino
   * a cui collegare la scheda dello stepper
   * vengono gestiti dalla libreria
   */

  // imposta la velocità di entrambi i motori a 15 rpm:
  myStepper_1.setSpeed(15);
  myStepper_2.setSpeed(15);

  // inizializzazione della porta seriale
  Serial.begin(9600);
}

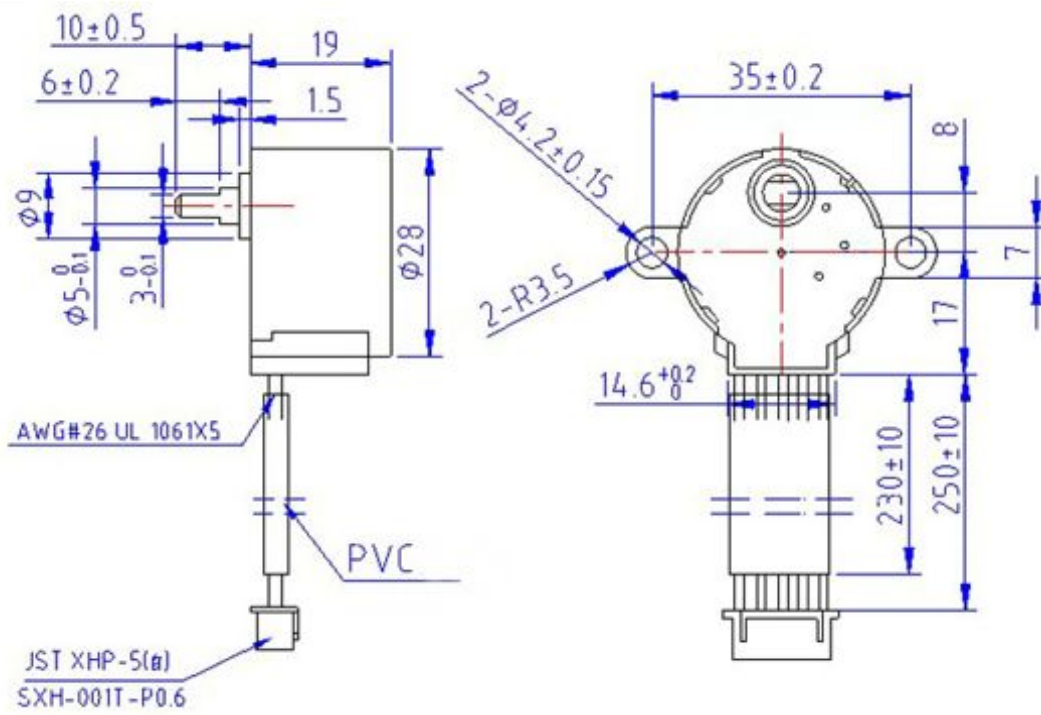
void loop() {

  // imposta una rotazione in senso orario
  Serial.println("orario");
  myStepper_1.step(stepsPerRevolution);
  delay(500);

  // imposta una rotazione in senso antiorario

  Serial.println("antiorario");
  myStepper_2.step(-stepsPerRevolution);
  delay(500);
}
```

Mechanical Data Sheet



Note:

