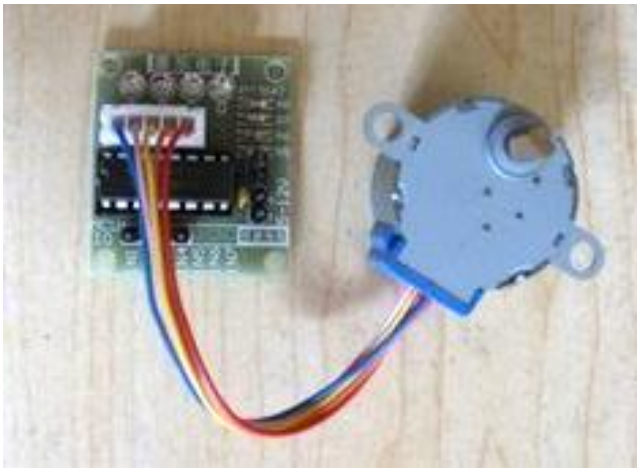
19 - motore passo passo 28YBJ-48 e relativo driver
Stepper motor and related driver (some notes at end of this section)

Un motore passo passo e' un'unita' elettromeccanica composta da un motore elettrico a 4 fasi e da un riduttore. Ogni impulso fa ruotare l'albero motore di un certo numero di gradi, predefiniti e sempre uguali.

Grazie a questa caratteristica e' possibile utilizzare un motore passo passo per far compiere movimenti di grande precisione ad apparecchiature complesse, come ad esempio un braccio meccanico o un selettore. La precisione dell'angolo di rotazione e le caratteristiche del motore sono tali da rendere possibile il controllo dell'ampiezza del movimento e, agendo sulla frequenza degli impulsi, anche della velocita' di rotazione.

Un'altra opportunita' offerta dai motori passo passo e' che non ci si deve preoccupare di installare sistemi di rilevamento e controllo della posizione delle apparecchiature ad esso collegate (la posizione del braccio meccanico, ad esempio) poiche' sara' sufficiente contare il numero degli impulsi inviati per calcolare l'esatta posizione di ogni elemento azionato dal motore.

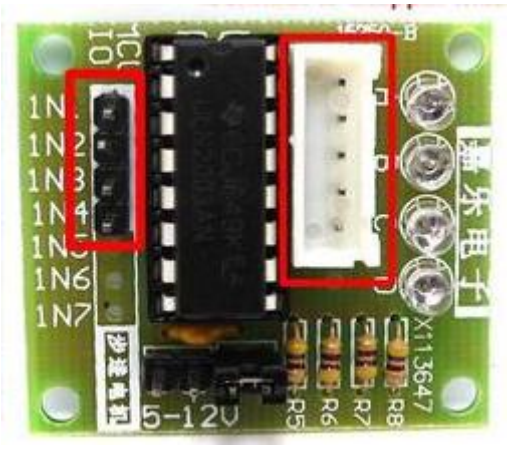


In figura viene proposto il motore passo passo 28YBJ-48, con riduttore incorporato.
Caratteristiche:

- motore a 4 fasi
- alimentazione 5-12 volt, da una fonte che puo' essere esterna ad Arduino
- assorbimento: 320 mA
- rapporto di riduzione 1/64

Questo motore puo' operare in due modalita': a 64 oppure a 32 impulsi per rotazione dell'albero motore. Le librerie di Arduino gestiscono la modalita' a 32 impulsi per cui ad ogni impulso l'albero motore ruota di 11.25 gradi.

Il motore e' pilotato da un driver che, operando sotto il controllo di Arduino, lancia gli impulsi necessari al movimento. Il motore e' dotato di un riduttore con rapporto 1/64, per cui, per una rotazione completa del perno in uscita, sono necessarie 64 rotazioni dell'albero motore e quindi 32*64 = 2048 impulsi . La connessione tra il driver ed il motore e' assicurata da un cavo a 4 fili che termina in uno spinotto da inserire nell'alloggiamento bianco presente sulla scheda driver



La connessione ad Arduino e' realizzata attraverso i 4 pin (IN1 – IN4) di input. L'alimentazione (da 5 a 12 volt, eventualmente esterna ad Arduino), corre invece sui due spinotti posti nella parte bassa della scheda, contrassegnati da un – ed un +

Per ottenere il corretto funzionamento del motore, bisogna attivare e disattivare le porte di ingresso (IN1 – IN4) rispettando la sequenza 1, 3, 2, 4. In realta' tutte le tematiche relative alla produzione ed al rilascio dei singoli impulsi sono gestite dal driver e dalle routine presenti nelle librerie di Arduino per cui la gestione pratica di un motore passo passo e' piuttosto semplice. E' infatti sufficiente fornire alle routine presenti in libreria alcune informazioni quali i numeri delle porte di connessione ad Arduino e la relativa sequenza di attivazione. Per ogni movimento bisogna poi fornire il numero di impulsi desiderati e la frequenza desiderata (in termini

di impulsi al secondo). Il senso di rotazione (orario o antiorario) e' gestito dal segno associato al numero di impulsi. Se il numero di impulsi e' negativo, l'albero motore gira in senso antiorario.

In questo esercizio ci si limitera' ad azionare un motore passo passo, facendo fare al perno in uscita un mezzo giro in senso orario seguito da un mezzo giro in senso antiorario e a velocita' piu' elevata. Prima dei due mezzi giri verra' mostrata, per quattro volte, a velocita' molto bassa e tramite i led presenti sul driver, la sequenza di lancio degli impulsi

Nota: Questo esercizio e questa nota sono parte di una serie che vede protagonisti Arduino ed alcuni dei componenti ad esso collegabili. Per la maggior parte degli esercizi e' anche disponibile un filmato su youtube.

- Esercizi facenti parte della raccolta
- Filmati presenti su youtube
- Informazioni su arduino e sui componenti collegabili (PDF scaricato nell'area di download)
- Breve manuale di programmazione (PDF scaricato nell'area di download)

Per eventuali chiarimenti o suggerimenti sul contenuto di questa scheda scrivere a giocarduino@libero.it

Here some notes about this project, translated by google translator 

A stepper motor is a electromechanical unit composed by a four phase electric motor and by a reducer. Each pulse rotates the crankshaft of a certain number of degrees, predefined and always identical.

Thanks to this feature, a stepper motor can be used to perform highly accurate movements on complex equipment, such as a mechanical arm or a selector. The rotation angle precision and the motor characteristics are such as to make possible monitoring the amplitude of movement and, acting on the frequency of the pulses, also the rotation speed.

Another opportunity offered by stepper motors is that we don't need a position detection systems for connected equipment (the position of the mechanical arm, for example), because will be enough count the number of sent pulses to calculate the exact position of each element driven by the motor.

In figure the stepper motor 28YBJ 48, with reducer. Features:

- 4-phase motor
- 5-12 volt power supply, from a source that can be external to Arduino
- consumption: 320 mA
- reduction ratio 1/64

This engine can operate in two ways: 64 or 32 pulses for a motor shaft rotation. The Arduino libraries manages the 32 pulses mode, so for each pulse the motor shaft rotates 11.25 degrees

The engine is managed by a driver which, operating under Arduino control, launches the needed impulses. The engine is equipped with a 1/64 ratio reducer, for which, a complete rotation of the output pin, needs 64 crankshaft rotations, thats means 32 * 64 = 2048 pulses. The connection between driver and engine is assured by a 4-wire cable that terminates in a plug to be inserted into the driver white slot.

To get the engine correct operation, must be turned on and off the input pins (IN1 - IN4), in sequence 1, 3, 2, 4. All issues relating to pulses are handled by driver and by Arduino libraries routines. The practical management of a stepper motor is quite simple. Is sufficient provide the

Arduino connection pins and their activation sequence. For every movement must then be provided the desired number of pulses and the desired frequency (in terms of pulses per second). The direction of rotation (clockwise or counterclockwise) is managed by the pulses number sign. If the number is negative, the motor shaft rotates in counterclockwise direction.

This project drives a stepper motor by turning shaft half rotation. A slow half turn clockwise followed by a fast half turn counter-clockwise. Before the two half-turns, will be shown some very low movements in order to see, on driver LEDs, the launch sequence of pulses.

Note: This project and this note is part of a series that sees, as main characters, Arduino and some of connectable components. For most projects there is also a video on youtube.

- Projects collection
- Movies on youtube
- About Arduino and components (italian; pdf will be downloaded in your download area
- Quick programming guide (almost english; pdf will be downloaded in your download area)

For any questions or suggestions about this note (and on its english translation), please write to giocarduino@libero.it (simple words and short sentences, please)

Materiali

- un motore passo passo 28YBJ-48 e scheda driver ZC-A0591

Schema

Collegare le porte 8, 9, 10 ed 11 di Arduino rispettivamente agli ingressi In1, In2, In3 ed In4 del driver. Alimentare il driver con una tensione da 5 volt e connettere il motore passo passo al driver mediante lo spinotto di cui e' dotato. Visto il basso assorbimento previsto per questo esercizio, e' possibile utilizzare le fonte di alimentazione da 5 volt fornita da Arduino.

Connect arduino pins 8, 9, 10 and 11, to In1, In2, In3, In4 driver pins. Connect the driver to a 5 volt power supply and to stepper motor. Because of the low uptake expected for this exercise, the driver can be connected to Arduino's 5 volt power supply.

Programma

```
/* Attenzione: facendo il copia/incolla dal PDF all'IDE si perde la formattazione del testo.
 * Per rendere piu' facilmente leggibile il programma e' opportuno formattarlo subito dopo il
 * trasferimento nell'IDE, premendo CTRL+T
 *
 * Esempio di utilizzo di un motore passo passo. Programma derivato da un analogo programma
 * reperito in rete
 *
 *-------------------------------------------------------------------------------------------
 * Warning: cut&paste from PDF to IDE loses formatting. to restore it press CTRL + T.
 * Using a stepping motor. Program derived from a similar program found on web
 *
 */
#include <Stepper.h> // richiama la libreria con le funzionalita' di gestione del motore passo passo
int impulsi; // definzione di una variabile intera chiamata "impulsi"
/* nella prossima istruzione sono forniti il numero di impulsi (32) necessari per far compiere
un giro dell'albero motore e vengono definite le porte da associare al driver con la loro
sequenza di attivazione. Le porte 8, 9, 10 ed 11 di Arduino sono collegate, rispettivamente,
ai pin In1, In2, In3 ed In4 del driver ma la sequenza di attivazione deve essere
In1, In3, In2 ed In4 per cui la dichiarazione delle porte e' 8, 10, 9, e 11

The next instruction provides the number of pulses (32) needed to make an engine shaft rotation,
the arduno's pin to connect to driver pins and their activation sequence. The arduino's pins 8, 9,
```

```
10 and 11 are connected, respectively, to driver pins In1, In2, In3 and In4 but, because the
activation sequence must be In1, In3, In2 and In4, the sequence pins is: 8, 10, 9, and 11.

*/
Stepper small_stepper(32, 8, 10, 9, 11);
//
//
void setup()
{
/* non e' necessaria alcuna dichiarazione iniziale poiche' le porte da utilizzare sono
dichiarate dalle routine presenti in libreria
We don't need any initial declaration because the necessary pins to be used are declared in previous
statement
*/
}
//
//
void loop()
{
//
// parte 1 - visualizzazione della sequenza di lancio degli impulsi
small_stepper.setSpeed(1); /* imposta una bassa velocita' dell'albero motore
(1 impulso al secondo) per mostrare, lentamente, la sequenza di lancio degli impulsi
(osservare la sequenza di illuminazione dei led sulla scheda driver)
sets a low speed crankshaft  (1 pulse per second) , to show, slowly, sequence of pulses (observe
the driver leds sequence illumination) */
impulsi = 4; // imposta 4 nel contatore di impulsi
small_stepper.step(impulsi); // lancia 4 impulsi, facendo muovere l'albero motore di
// 45 gradi (11,25 gradi per 4 volte) ed il perno esterno di 0,7 gradi
delay(2000); // aspetta 2 secondi prima di attivare il successivo movimento
//
// parte 2 - rotazione del perno in uscita di 180 gradi in senso orario e a velocita' moderata
impulsi = 1024; // inserisce nella variabile impulsi il valore 1024,
// e cioe' il numero di impulsi necessari per far compiere mezzo giro al perno di uscita
small_stepper.setSpeed(100); // imposta una bassa velocita' di rotazione dell'albero
// motore (100 impulsi al secondo, pari a 3,125 giri al secondo dell'albero motore e ad
// una rotazione di 17,5 gradi al secondo del perno in uscita.)
small_stepper.step(impulsi); // ripete per 1024 volte la sequenza di lancio degli impulsi
// (la variabile "impulsi" contiene il valore 1024)
delay(1000); // aspetta un secondo prima di attivare il successivo movimento
//
// Parte 3 - rotazione del perno in uscita di 180 gradi, a velocita' elevata ed in senso antiorario
impulsi = -1024; // inserisce in "impulsi" il valore negativo -1024,
// in modo da imprimere un movimento antiorario al perno di uscita
small_stepper.setSpeed(700); // imposta un'alta velocita' di rotazione dell'albero motore
// (700 impulsi al secondo, pari a 21,85 giri al secondo dell'albero motore e 123 gradi al secondo
// del perno in uscita)
small_stepper.step(impulsi); // ripete per 1024 volte la sequenza di lancio degli impulsi che,
// essendo negativa, fa girare l'albero in senso antiorario
delay(2000); // attende due secondi prima di ripetere l'intera sequenza
}
```