

Collegare il motore passo passo 28BYJ-48 ad Arduino UNO R3



aspettandoilbus.blogspot.it/2014/09/collegare-il-motore-passo-passo-28byj.html

Ciao e ben ritrovati sul Blog *Aspettando il Bus*. Negli ultimi giorni sono stato abbastanza impegnato e non ho avuto il tempo di mostrarvi il motorino 28BYJ-48 e come controllarlo con Arduino UNO R3, ma rimediamo subito...

(nota: La versione per **Olimexino 85 è qui**)

Il progetto che vi propongo oggi è abbastanza economico, il modulo con l' ULN2003A ed il motore passo passo costano circa **8 Euro** ma si possono trovare anche a meno. Vediamo la foto del progetto di oggi:

Iniziamo col motorino 28BYJ-48 e dal suo **datasheet**. Il motorino è già provvisto di connettore per il collegamento. Si tratta di un motore passo passo unipolare con 4096 passi per una rotazione di 360 gradi. Il suo perno non si ruota facilmente, anche senza alimentazione elettrica resta bloccato. Quindi si può risparmiare energia non alimentando gli avvolgimenti quando deve restare fermo e così si riscalderà molto meno. I colori dei fili del motore sono Blu, Rosa, Giallo, Arancio e Rosso.

Vediamo le istruzioni del modulo di controllo che utilizza l'integrato ULN2003A con il suo **datasheet**.

Sulle mie perplessità su questi moduli e di come controllare gli step motor ne ho già parlato **qui**. Tornando a noi, guardando nel PDF troviamo alcune utili informazioni:

1. Quali sono le fasi del motore ed il colore dei fili delle fasi;

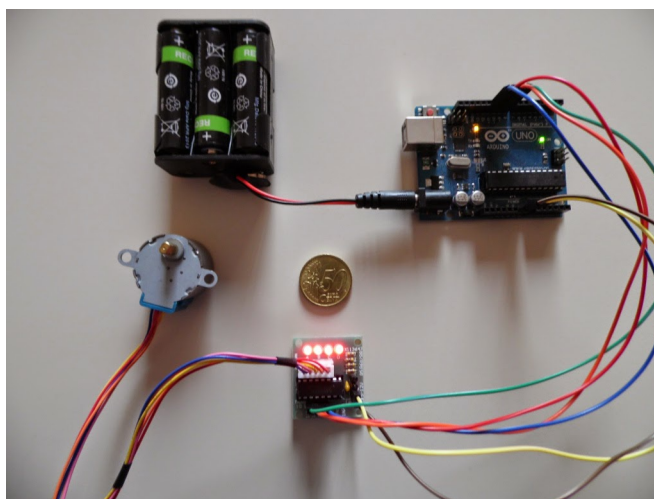


Fig. 1 - Arduino UNO R3 e motore passo passo 28BYJ-48 - Foto di Paolo Luongo



Fig. 2 - Step Motor 28BYJ-48 - Foto di Paolo Luongo

2. In che modo si devono attivare le varie fasi per farlo girare correttamente.

Per questo modulo ho collegato i fili, da sinistra a destra, di colore Verde, Arancio, Rosso, Blu. L'alimentazione +12 Volt o +5 Volt dipende dal motore passo passo utilizzato, (nella foto il filo Giallo) e GND (Marrone). Il Giallo si

può collegare al piedino Vin di Arduino UNO R3 per motori a 12 volt e ai 5 Volt se utilizzate il 28BYJ-48. Il marrone va collegato ad un piedino GND sulla board Arduino UNO R3.

Su Arduino UNO R3 il filo Blu va collegato al Pin11, il Rosso al Pin10, l' Arancio al Pin 9 ed il Verde al Pin8. Ovviamente i piedini si possono cambiare ma si deve modificare lo sketch.

Dopo aver scaricato lo **sketch** si carica nell' IDE di Arduino e si programma Arduino UNO R3. Vista la finalità puramente didattica vediamo alcune parti dello sketch. Nella foto precedente ci sono le indicazioni di quali Pin di uscita si voglio utilizzare, questo è il posto dove cambiarli.

La funzione *Ritardo*

Accensione permette di

avere un ritardo iniziale di 5 secondi prima di far iniziare la rotazione del motore. E' buona norma, prima di azionare dei meccanismi, prevedere una breve pausa per permettere all'operatore di allontanarsi dal macchinario prima che inizi la lavorazione.

Se date uno sguardo al PDF del modulo scoprirete che per far ruotare il motore passo passo 28BYJ-48 bisogna alimentare le fasi seguendo un ordine preciso. Bene questa

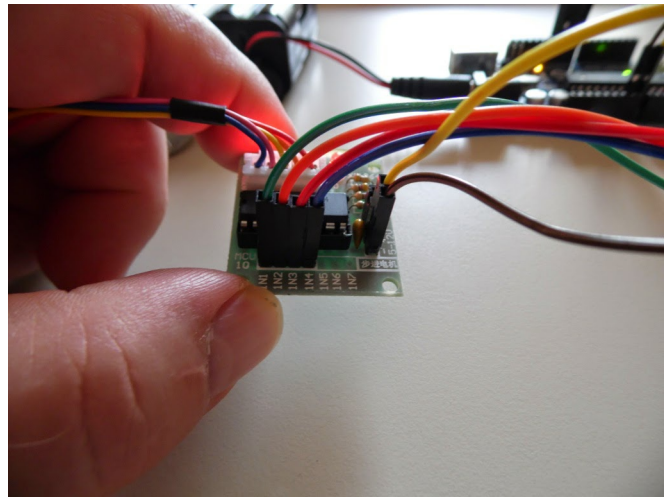


Fig. 3 - Modulo di controllo motore 28BYJ-48 con UNL2003A - Foto di Paolo Luongo

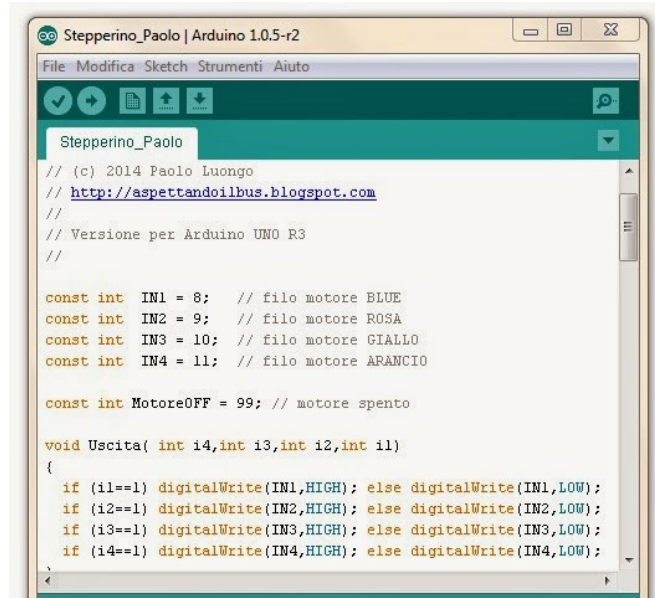


Fig. 4 - Sketch per il comando del motore con Arduino UNO R3 di Paolo Luongo

sequenzialità è fatta da questa parte di codice. Ho scelto di far muovere il motore con otto sequenze. Se preferite farlo con quattro sequenze ecco la parte che dovete modificare.

Questo è il solito blocco di inizializzazione: notate la funzione di ritardo all'accensione.

Questa è solo la prima parte del programma di controllo. E' la parte che si occupa della rotazione oraria. Il motore ruota di 90 gradi, fa una pausa di mezzo secondo poi ruota, pausa...fino a fare una rotazione di 360 gradi. Notate che per far ruotare il motore di 90 Gradi servono 1024 impulsi.

Dopo la rotazione oraria c'è una pausa di 1 secondo per poi fare la rotazione antioraria.

Bene, per oggi è tutto. Devo scappare: FINALMENTE è arrivato il BUS !

Se hai dei dubbi, commenta l'articolo e [leggi le FAQ](#).

Ciao
Paolo :-)

Leggi altri [articoli del Blog](#).

```
void RitardoAccensione()
{ //attesa prima di attivare il motore
  EseguiPasso(MotoreOFF);
  for(int i=0; i<20; i++)
  {
    digitalWrite(13,HIGH);
    delay(250);
    digitalWrite(13,LOW);
    delay(250);
  }
}
```

Fig. 5 - Sketch:Ritardo iniziale - di Paolo Luongo

```
Stepperino_Paolo
void EseguiPasso(int stato)
{
  int i1,i2,i3,i4;

  switch ( stato )
  { // vedi tabella nel pdf del motore passo passo
    case 0: Uscita(0,0,0,1); break;
    case 1: Uscita(0,0,1,1); break;
    case 2: Uscita(0,0,1,0); break;
    case 3: Uscita(0,1,1,0); break;
    case 4: Uscita(0,1,0,0); break;
    case 5: Uscita(1,1,0,0); break;
    case 6: Uscita(1,0,0,0); break;
    case 7: Uscita(1,0,0,1); break;
    case MotoreOFF: //OFF
      Uscita(0,0,0,0); break;
  }
  delay(1); //ritardo almeno 1 mS
}
```

Fig. 6 - Sketch: Controllo motore - di Paolo Luongo

```
void setup()
{
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(13,OUTPUT);
  RitardoAccensione(); //5 secondi di pausa prima di iniziare
}
```

Fig. 7 - Scketch: Inizializzazione della Board Arduino Uno R3 - di Paolo Luongo

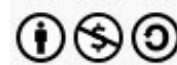
```

void loop()
{
  int stato;

  stato=0; //inizio da uno stato arbitrario
  digitalWrite(13,HIGH); //acceso LED13
  for(int k=0; k<4; k++) //Rotazione Oraria
  {
    for (int i=0; i<1024; i++) //90 gradi a ciclo
    {
      EseguiPasso(stato);
      stato+=1; // avanza nella tabella
      if ((stato)>7) stato=0;
    }
    EseguiPasso(MotoreOFF);
    delay(500); // pausa di mezzo secondo
  }
  delay(1000); // pausa di 1 secondo
  digitalWrite(13,LOW); //spento LED13
  //Rotazione Antioraria
}

```

Fig. 8 - Sketch: Parte del programma principale di controllo - di Paolo Luongo



CC BY-NC-SA