



UNIVERSIDAD DE SALAMANCA
ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA INDUSTRIAL

Avda. Fernando Ballesteros, 2
37700 BÉJAR
Fax 923 40 81 27
Telf. 923 40 80 80

ÁREA DE INGENIERÍA ELÉCTRICA

CONTROL DE MOTOR PASO A PASO

PRÁCTICAS DE MÁQUINAS ELÉCTRICAS

Norberto Redondo Melchor
Profesor Asociado
Ingeniero Industrial
Doctor por la Universidad de Salamanca

Béjar (Salamanca - España)

Copyright

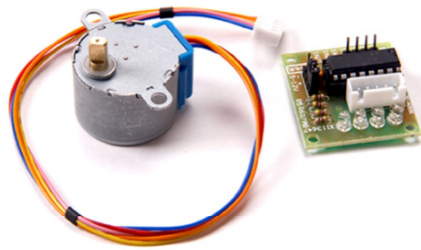
© Norberto Redondo Melchor
(NRM) 2016

Todos los derechos reservados

Reproducción autorizada siempre
que se haga de todo el documento
íntegro y sin alterarlo lo más
mínimo, incluyendo esta nota.

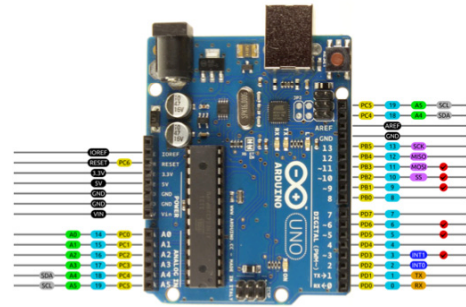
Esta práctica pretende mostrar una manera de controlar un motor paso a paso mediante un teclado de cuatro pulsadores.

Material necesario:



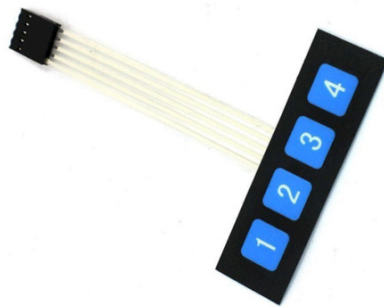
1 Motor 28BJY-48

1 Controlador basado en el ULN2003A

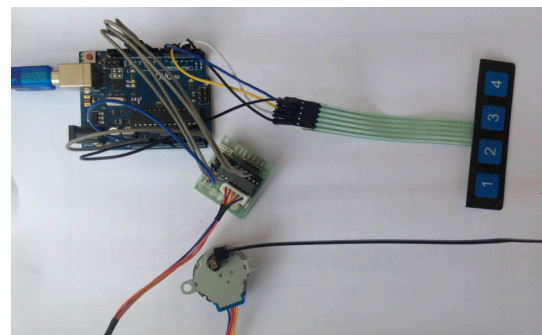


1 Microcontrolador Arduino Uno o equivalente

1 Cable USB

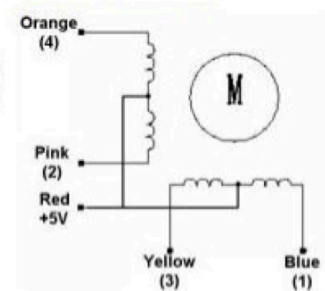


1 Teclado de membrana

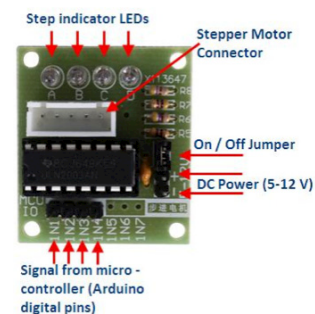


Montaje

El motor 28BJY-48 viene cableado como muestra la figura de la derecha. El movimiento se consigue excitando ordenadamente cada una de las 4 bobinas de sus devanados internos.



El microcontrolador servirá para proporcionar los pulsos requeridos y de la duración necesaria (ms). Pero no puede entregar la potencia que precisan los devanados del motor, por lo que es necesario intercalar una serie de transistores que, excitados en su base por el micro, permitan pasar la intensidad necesaria sin daño para el Arduino.



El ULN2003A es una matriz de 7 transistores NPN de los que se usan 4. Al poner a +5V respecto de GND cualquiera de los pines IN1 a IN4 del circuito donde va instalada se habilita el paso de potencia hacia el motor por la salida correspondiente. Además el led asociado también se encenderá.

Cualquiera de las dos secuencias de las tablas de la derecha moverán el motor en sentido horario, forzando $\frac{1}{4}$ de vuelta del motor (un 'paso') cada fila. Si se siguen las secuencias al revés el sentido de giro será el antihorario.

El eje de 5 mm Ø de color latón no es el eje del motor eléctrico, sino el eje del último engranaje de una cascada de ellos, que proporcionan una desmultiplicación de

283712 / 4455
(64 : 1 aprox)

Si ninguna de las bobinas está excitada el motor se parará, pero su eje no ejercerá mucha oposición a ser arrastrado por la carga. Si se mantienen activas las bobinas del último paso, en cambio, se podrá notar una mayor resistencia del motor que quedará suficientemente bloqueado.

El programa de la página siguiente permite dar los pulsos correspondientes a las posiciones 1 a 4 de la tabla de arriba (*Full Step Sequence*): al soltar la tecla 1 se encenderán durante 200 ms las salidas IN1 & IN4 del controlador, excitando las bobinas 1a & 2b del motor respectivamente; con la tecla 2 se excitarán 200 ms la 1a & 1b; con la tecla 3 serán la 1b & 2a, y con la tecla 4 la 2a & 2b. El resultado habrán sido 4 pasos sucesivos del motor para mover el eje exterior en sentido antihorario.

La secuencia de teclas 4-3-2-1 moverá el motor de manera que el eje exterior lo haga en el sentido de las agujas del reloj.

El otro programa hace girar continuamente el motor primero en un sentido y luego en el otro.

Nota: la holgura de la cascada de engranajes induce cierta 'pereza' hasta que se nota el movimiento del eje externo. Esto es más patente cuando se invierte la secuencia de teclas para girar en sentido contrario. El motor eléctrico da el paso requerido siempre, pero hasta que la holgura de los engranajes no se ha compensado ese movimiento no se nota.

Agradecimientos: a Vaquero y a Fortu por el interés mostrado en este proyecto.

Index	1a	1b	2a	2b
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0
4	0	0	1	1
5	1	0	0	1
6	1	1	0	0
7	0	1	1	0
8	0	0	1	1

Alternate Full Step Sequence
(Provides more torque)

Index	1a	1b	2a	2b
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1
9	1	0	0	0
10	1	1	0	0
11	0	1	0	0
12	0	1	1	0
13	0	0	1	0
14	0	0	1	1
15	0	0	0	1
16	1	0	0	1

Half Step Sequence



```
/* Programa demostración del funcionamiento del motor paso a paso
28BYJ-48, controlado mediante la pulsación ordenada de 4 teclas
(P1,P2,P3,P4) en Full Step mode

Para Arduino Uno, Arduino Nano, Arduino Mini, etc.
By Norber ©2016

Freely distributable if you cite the source
*/
#define IN1 9 // Pines de salidas al motor
#define IN2 10
#define IN3 11
#define IN4 12
#define P1 2 // Pines de entradas de teclado
#define P2 3
#define P3 4
#define P4 5
#define LED_ON digitalWrite(13, HIGH);
#define LED_OFF digitalWrite(13, LOW);
#define DUR_PULSO 200 // En ms
#define NO_HAY_PULSACION 20

struct Botonera { // Definición del teclado
  byte numPuls;
  byte pinPuls;
  int rebotePuls;
};
Botonera Pulsador[] = { // numPuls, pinPuls, rebotePuls
  {0, P1, 0}, {1, P2, 0}, {2, P3, 0}, {3, P4, 0}
};

void setup() { // Configuración de entradas y salidas
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(P1, INPUT_PULLUP);
  pinMode(P2, INPUT_PULLUP);
  pinMode(P3, INPUT_PULLUP);
  pinMode(P4, INPUT_PULLUP);
  pinMode(13, OUTPUT);
}

byte leerPulsadores() { // Devuelve el pin pulsado, con antirrebote
  for (int i = 0; i < 4; i++) {
    Pulsador[i].rebotePuls = (Pulsador[i].rebotePuls <<
1) | !digitalRead(Pulsador[i].pinPuls) | 0xFE00;
    if (Pulsador[i].rebotePuls == 0xFF00) return Pulsador[i].pinPuls;
  }
  return NO_HAY_PULSACION;
}

void pulsoEn(byte pin) { // Ejecuta los pulsos en 'pin' y en 'sigPin'
  long ahora = millis();
  LED_ON;
  byte sigPin = (pin + 1);
  if (sigPin > IN4) sigPin = IN1;
  digitalWrite(pin, HIGH);
  digitalWrite(sigPin, HIGH);
  while ((millis()-ahora) < DUR_PULSO) {}
  digitalWrite(pin, LOW);
  digitalWrite(sigPin, LOW);
  LED_OFF;
}

void loop() { // Rutina principal que se ejecuta continuamente
  byte pulsado = leerPulsadores();
  if (pulsado != NO_HAY_PULSACION) {
    if (pulsado == P1) pulsoEn(IN1);
    if (pulsado == P2) pulsoEn(IN2);
    if (pulsado == P3) pulsoEn(IN3);
    if (pulsado == P4) pulsoEn(IN4);
  }
}
```

```

/* Programa demostración del funcionamiento del motor paso a paso
   28BYJ-48 que gira continuamente consumiendo 200 mA en Half Step mode

   Obtenido de http://elcajondeardu.blogspot.com.es
*/

//Declaración de las variables para los pines del motor
int motorPin1 = 9;      // Azul      - 28BYJ48 pin 1 motor
int motorPin2 = 10;     // Rosa     - 28BYJ48 pin 2 motor
int motorPin3 = 11;     // Amarillo - 28BYJ48 pin 3 motor
int motorPin4 = 12;     // Naranja  - 28BYJ48 pin 4 motor
                        // Rojo     - 28BYJ48 pin 5 (VCC) motor

int motorSpeed = 1200; // Fija la velocidad del motor (el retraso entre cada
secuencia)
int count = 0;         // Cuenta de los pasos dados
int countsperrev = 512; // Número de pasos por vuelta completa
int lookup[8] = {B01000, B01100, B00100, B00110, B00010, B00011, B00001, B01001};

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

void loop(){
  if(count < countsperrev) clockwise(); //sentido agujas del reloj
  else if (count == countsperrev * 2) count = 0;
  else anticlockwise(); //sentido antihorario
  count++;
}

void anticlockwise() {
  for(int i = 0; i < 8; i++) {
    setOutput(i);
    delayMicroseconds(motorSpeed);
  }
}

void clockwise() {
  for(int i = 7; i >= 0; i--) {
    setOutput(i);
    delayMicroseconds(motorSpeed);
  }
}

void setOutput(int out) { //función secuencia giro
  digitalWrite(motorPin1, bitRead(lookup[out], 0));
  digitalWrite(motorPin2, bitRead(lookup[out], 1));
  digitalWrite(motorPin3, bitRead(lookup[out], 2));
  digitalWrite(motorPin4, bitRead(lookup[out], 3));
}

```