Práctica - Motores I

Daniela Alejandra Cordova Porta

Richard Junior Mercado Correa

UNIVERSIDAD COMPLUTENSE DE MADRID

Madrid, noviembre de 2020

Índice

Resumen

Este documento expone el procedimiento y análisis realizado en la práctica 1 - Motores I de la asignatura Robótica en la Universidad Complutense de Madrid. En primer lugar, se establecen los objetivos de la práctica y su fundamento teórico. Luego, se establece el procedimiento para realizarla; explicando las conexiones y código necesarios. Finalmente se discuten los resultados obtenidos y se exponen los errores cometidos.

1. Objetivos

- Aprender a utilizar el entorno de programación de la Raspberry Pi.
- Uso de motores paso a paso y modificación de su velocidad.
- Realizar un programa para controlar la posición y velocidad del motor PaP.

2. Fundamento Teórico

Los robots se mueven usando motores eléctricos de los cuales existen 2 grandes familias que son: de alterna y de continua. Los motores de alterna pueden ser asíncronos, síncronos y especiales. En cuanto a los de continua, pueden ser de excitación independiente, autoexcitados, paso a paso o especiales. En esta práctica estudiaremos los motores paso a paso.

Los motores paso a paso pueden ser bipolar (formado por dos bobinas) o unipolares (formado por cuatro bobinas). Se utilizará para el experimento un motor unipolar. Para que este funcione, se le debe dar corriente durante un intervalo de tiempo a cada una de las bobinas, una detrás de la otra; creando así un campo magnético giratorio y así el motor gira paso a paso.

Si se desea que el motor vaya más rápido, se disminuye el intervalo T o se aumenta la frecuencia implementada. De manera contraria, si se desea que vaya más lento, se aumenta el intervalo T o se reduce la frecuencia.

Para generar la secuencia de pulsos de las bobinas se utilizará el controlador del motor NJM2671D2. Este genera la secuencia de pulsos para ir dando corriente a las bobinas. La patilla STEP permite darle la señal de reloj para generar los pulsos en el motor y la patilla DIR cambia el sentido de giro del motor.

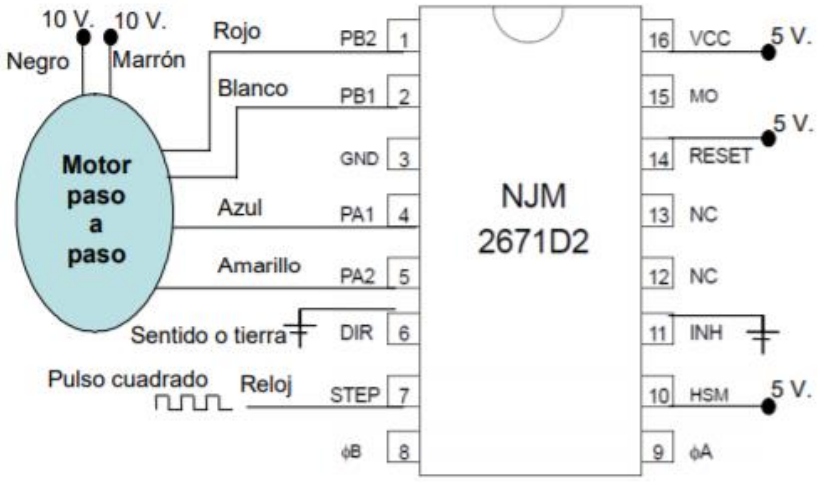


Fig. 1. Circuito

3. Procedimiento

a. Entorno de programación y control del robot

Conectamos la RaspBerry Pi a través del cable Ethernet y usamos el VNC viewer para poder realizar la práctica. Se lanza un terminal y nos ubicamos en la carpeta dónde está el archivo a ejecutar. Compilamos el programa “blink.c” con make blink. Luego realizamos el montaje mostrado en la figura siguiente, conectando el led al Pin GPIO 17 (el Wiring Pin 0) y usando la resistencia de 220 Ω. Esta también se conecta a GND de la RaspBerry Pi. Finalmente ejecutamos el programa con “sudo ./blink” y parpadea el led. Si queremos que el led parpadee más rápido debemos aumentar la frecuencia, es decir, reducir el delay entre cada subida y bajada del pulso en el código.

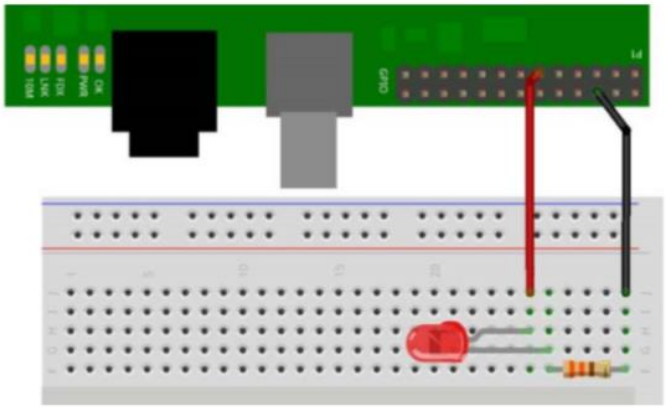


Fig. 2. Circuito para el led

b. Control de un motor paso a paso

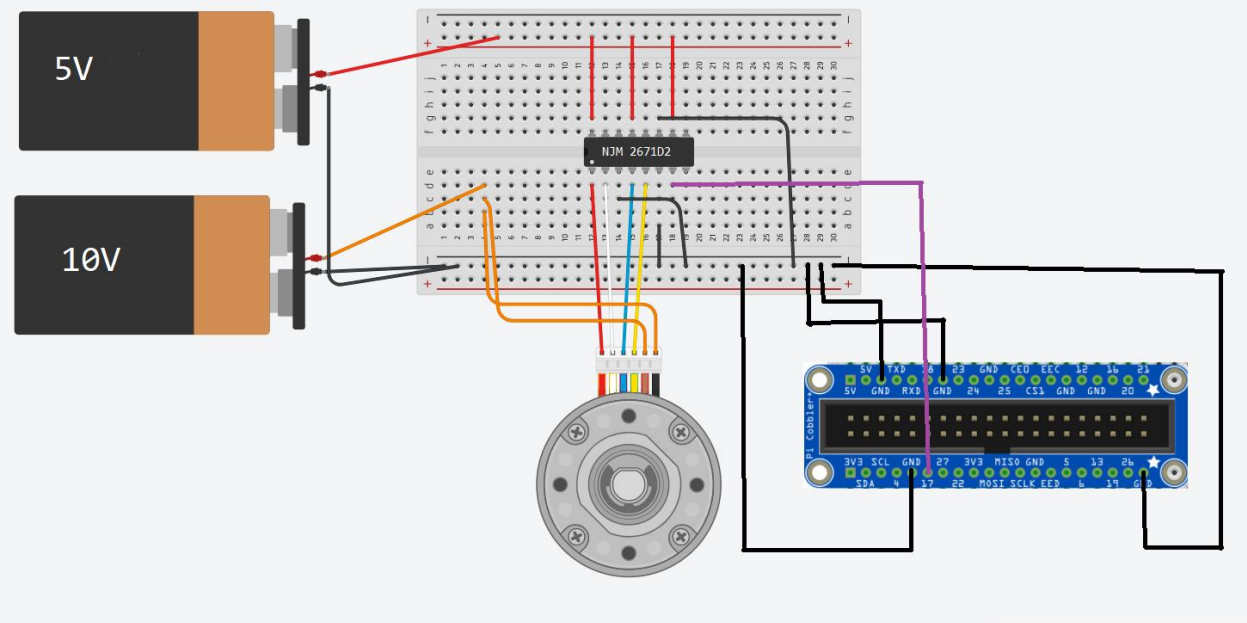


Fig. 3. Circuito Básico con conexiones con el motor

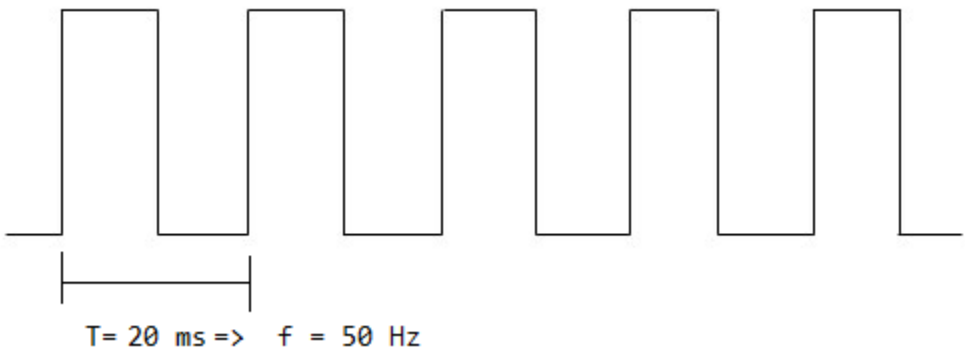
Se armó el circuito mostrado en la figura, se conectaron el pin 7 del controlador con el pin 17 de la raspberry pi 2 (GPIO 17 - Wiring Pin 0). Con el código siguiente, se generaron los pulsos para el motor:

Código:

```
#include <wiringPi.h>

int main (void)

{

  wiringPiSetup () ;

  pinMode (0, OUTPUT) ;

  for (;;)

  {

    digitalWrite (0, HIGH) ; delay (10) ;  /* delay: se le da números en
milisegundos */

    digitalWrite (0,  LOW) ; delay (10) ;

  }

  return 0 ;

}
```

- digitalWrite (0, HIGH): crea la subida del pulso
- digitalWrite (0, LOW): crea la bajada del pulso
- delay (10): espera 10 ms antes de realizar la siguiente subida o bajada de la onda cuadrada

De esta forma se crea un pulso de frecuencia 50 Hz (T = 20 ms), es decir, crearía pulsos cada 20 ms.



4. Resultados y Análisis de Resultados

a) Genere una frecuencia de pulsos de unos 50 Hz. Estime la velocidad del motor (en vueltas/s), ¿concuerda con lo esperado?. ¿Qué pasa si la frecuencia es muy alta?.

$$\omega = \frac{2\pi}{T} = 2\pi f$$

Se espera que la velocidad sea:

w=2π(50 Hz) =100π = 15,915494 vueltas por segundo

Se observa que sí concuerda con lo esperado, daba vueltas tan rápido que no se podía apreciar la velocidad a la que iba. Si la frecuencia es más alta, más rápido iría; es decir, aumenta la velocidad.

b) Estime los pulsos necesarios para que el motor de una vuelta completa.

45 pulsos necesarios para la vuelta la completa

c) Coloque un interruptor (del entrenador) en la señal de sentido (DIR) y compruebe que al pulsarse se invierte la señal de giro. Después, conecte otro pin de salida de la RaspBerry y haga que el programa cambie periódicamente el sentido del motor (cada dos vueltas completas cambie de sentido). Indique el esquema de las conexiones realizadas.

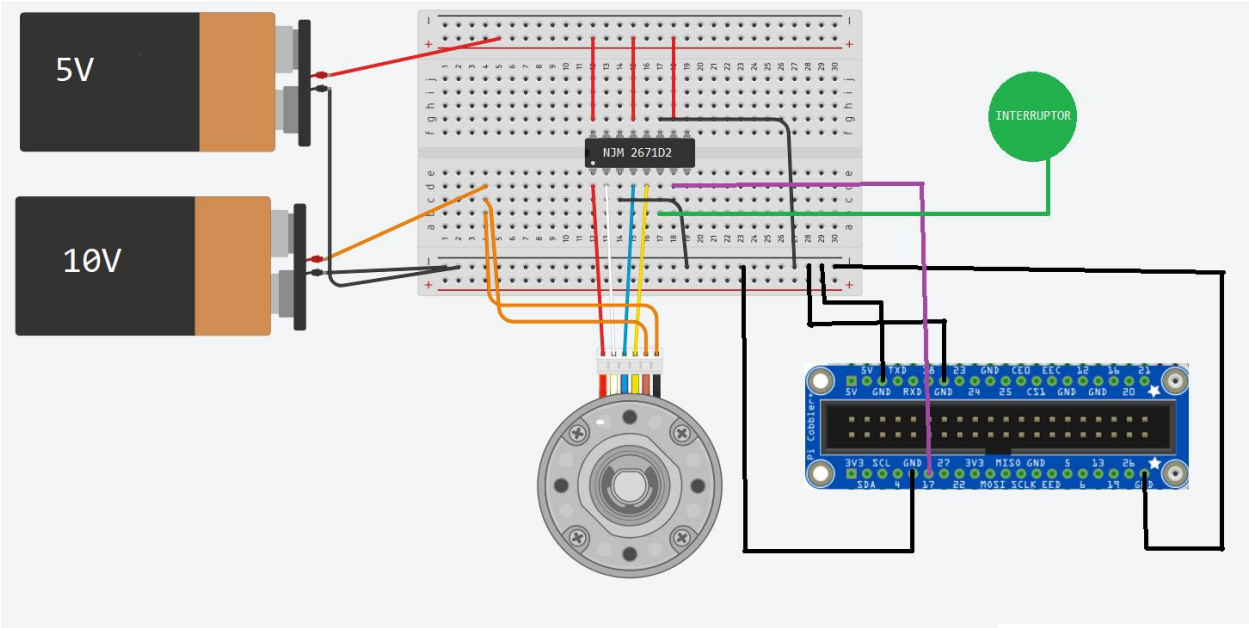


Fig. 4. Circuito Básico con conexiones con el motor y el interruptor

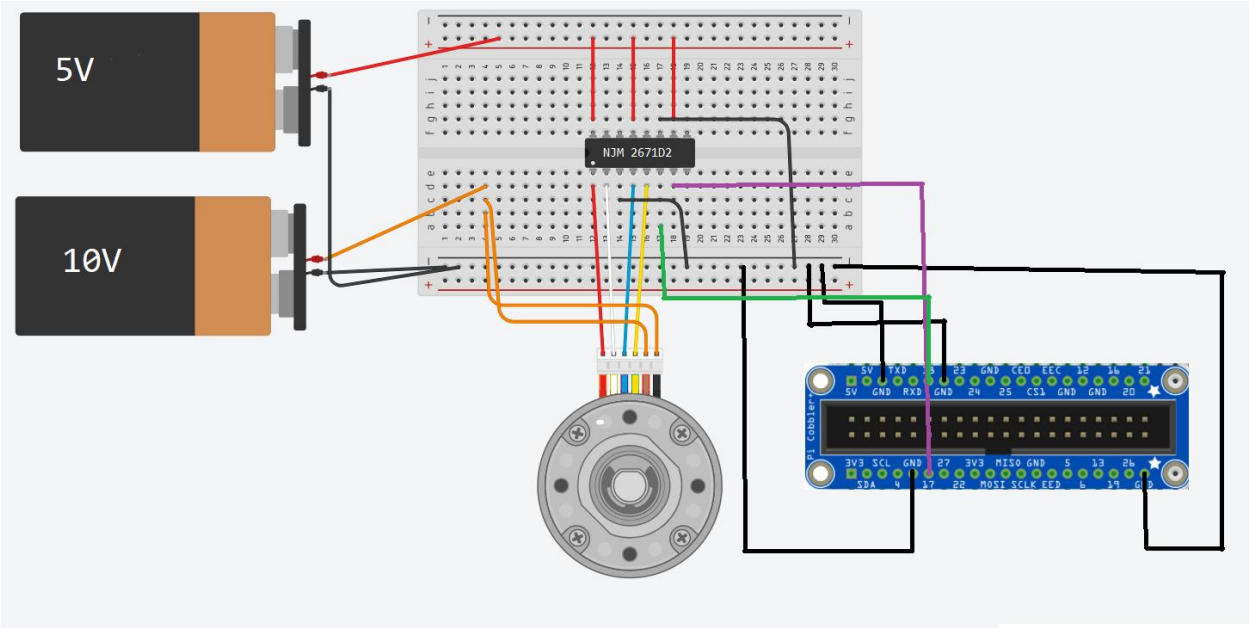


Fig. 5. Circuito con las conexiones de la RaspBerry pi para los cambios de dirección

Con el siguiente código se muestra cómo se hacían los pulsos y los cambios de dirección que se hacen con el pin 18 que es el GPIO 18 (Wiring Pin 1) de la RaspBerry pi. Este código cambia la dirección cada 90 pulsos, ya que cada 45 es una vuelta completa.

```
#include <wiringPi.h>
int main (void)
{
     int cont =0;
    wiringPiSetup () ;
    pinMode (0, OUTPUT) ;    //Pulsos
    pinMode (1, OUTPUT) ; //Dirección
    digitalWrite (1, LOW);
    int valor = LOW;
    for (;;){
       if(cont==90)
        {   cont=0;
            if(valor==LOW){
                digitalWrite (1, HIGH);
                valor = HIGH;}
            else{
                digitalWrite (1, LOW);
                VALOR = LOW;}
           }

      digitalWrite (0, HIGH) ; delay (20) ;
      digitalWrite (0,  LOW) ; delay (20) ;
      cont = cont +1;
  return 0 ;
}
```

5. Discusión y conclusiones

a. Conclusiones:

- A mayor frecuencia, mayor velocidad del motor.
- A mayor período (o delay en el código), menor velocidad en el motor.
- El motor necesita 45 pulsos para poder dar una vuelta completa
- Para cambiar de dirección, el motor debe recibir una señal. En este caso, la señal debe ser recibida en el pin 6 del controlador

b. Errores:

En la primera sesión del laboratorio se tuvo problemas con las conexiones ya que no se habían conectado todas las tierras (GND) y se había colocado como salida de 5V la dada por la RaspBerry pi en vez la del entrenador.

Luego, al tratar de cambiar la dirección con el código explicado anteriormente; retrocedía antes de lo que debía ya que se habían colocado llamadas al `digitalWrite` erróneamente por lo cual volvía a cambiar de sentido.