

PRÁCTICA DE ROBÓTICA 1 : Motores

Parte I : Introducción y motores paso a paso

Carlos Tardón Rubio
Marcos Herrero Agustín
Martín Fernández de Diego

21 de octubre, 2021

Resumen

Esto es una introducción al uso de la Raspberry Pi. Permite al usuario familiarizarse con el establecimiento de conexión vía Ethernet y la generación y ejecución de código sobre esta plataforma. Además realizamos experimentos sobre un motor paso a paso. Estos motores son muy utilizados si se desea mucha precisión en el movimiento o se quiere posicionar en un ángulo determinado. En este documento recogemos los pasos que hemos seguido y los problemas que se nos han presentado a lo largo de la práctica.

1. Objetivos

- Aprender a utilizar el entorno de programación de la Raspberry Pi.
- Uso de motores paso a paso y modificación de su velocidad.

2. Desarrollo

2.1. Entorno de programación y control del robot

El primer objetivo: conseguir hacer parpadear una LED ejecutando un programa *blink* de ejemplo del directorio *wiringPi/examples*. Para ello, alimentamos la raspberry, conectamos el portátil mediante ethernet, e hicimos ssh con el comando `ssh pi@192.168.10.10`. Después, montamos el sencillo circuito que se nos indica en el pdf, es decir, un cable negro conectado a tierra de la raspberry(GND) y a la resistencia de 220 ohmios. El led se conecta a la resistencia y a un cable rojo que va al pin 17 de la raspberry(wiring pin 0).

Problemas presentados

- Al principio, el portátil y la raspberry no se conectaban. Esto se solucionó estableciendo una ip fija en la configuración de red del portátil, y desactivando dhcp.
- La bombilla LED proporcionada estaba defectuosa y parpadeaba tenuemente. Al reemplazarla pudimos ver que el ejemplo funcionaba correctamente.

2.2. Control de un motor paso a paso

Nuestro objetivo ahora: con un programa muy similar al anterior, hacer girar un motor PaP mediante el controlador NJM2671D2.

Ejercicios

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

Para generar pulsos a 50Hz, debemos mantener los flancos HIGH y LOW durante 10ms cada uno. Hemos estimado que la velocidad del motor es de 1 vuelta por segundo. Concuerda con lo esperado. Si la frecuencia es muy alta estos motores comienzan a presentar diversos problemas, en nuestro caso no realizaba ningún movimiento.

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

Hemos anotado 477 pulsos en 10 vueltas. Por tanto, aproximadamente 47 pulsos son necesarios para dar una vuelta completa.

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.

Efectivamente, una vez conectado el interruptor del entrenador al PIN 6 del NJM2671D2, el sentido cambiaba al pulsar el botón. Conectamos el PIN 23 GPIO (4 en la numeración de wiringPi) y cambiamos el programa, `digitalWrite(ORIENT, HIGH)` y `digitalWrite(ORIENT, LOW)` para cambiar el sentido. Como sabemos que una vuelta son 47 pulsos, hay que cambiar de LOW a HIGH o de HIGH a LOW cada 94 vueltas del bucle.

A continuación, el esquema de conexiones.

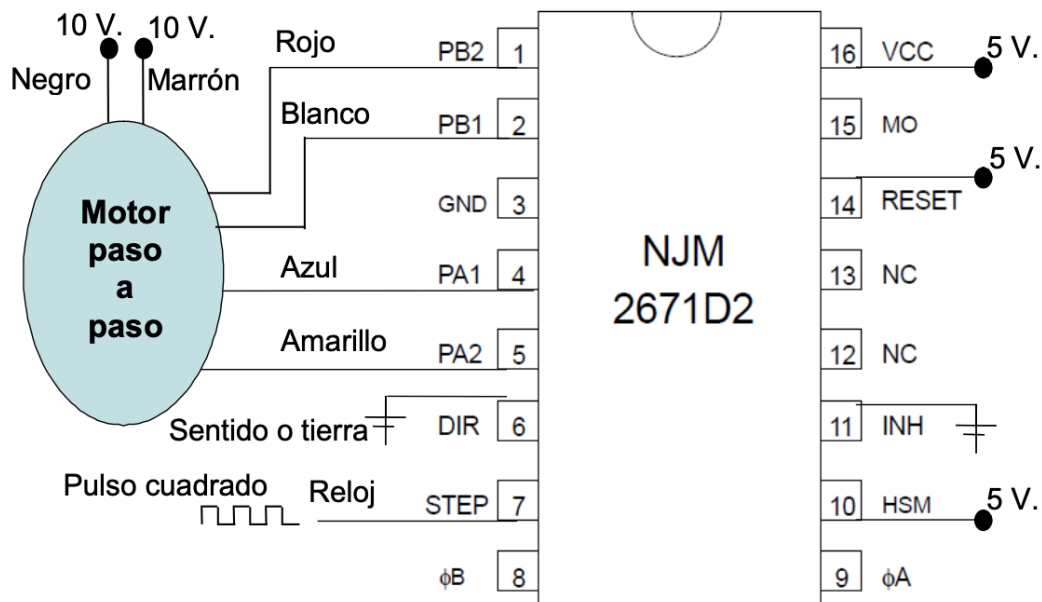


Figura 1: Esquema de conexiones

El **controlador** se alimenta con 5V del entrenador por los pines 16, 14 y 10 y conecta con tierra¹ por el PIN 3 y el 11. Para realizar estas conexiones, conectamos un cable negro a la línea negativa (-) de la placa de conexiones (que venía en el maletín junto a la raspberry), que va al GND del entrenador. De esta línea se sacan cables más pequeños para conectarlos al NJM2671D2. Además, esta línea también hay que conectarla a la masa de la raspberry. Para las conexiones +5V hicimos lo mismo, conectamos un cable largo rojo al +5V del entrenador y a la línea positiva (+) de la placa de conexiones.

La **Raspberry Pi** indica al PIN 7 la frecuencia de paso que deseamos en el motor a través del PIN de salida 17 de GPIO (0 en la numeración de wiringPi). Para ello tuvimos que conectar un cable macho-hembra del pin 17 GPIO, al pin 7 del NJM

El **motor PaP** también se alimenta con 10V del entrenador, para lo cual fue necesario medir con el voltímetro y girar la rueda del entrenador hasta obtener el voltaje requerido.

Problemas presentados

- Inicialmente, montamos las conexiones con el chip equivocado. Nos dimos cuenta cuándo solicitamos el motor y vimos que venía con su propio chip.
- El PIN 23 del GPIO no emitía señal de output. Lo solucionamos inicializando correctamente el PIN con la instrucción `pinMode(ORIENT, OUTPUT);`.
- El puerto GND no lo habíamos conectado a tierra. Esto produjo que el chip se sobrecalentara.

A continuación el programa que usamos para esta práctica, configurado para invertir la señal de giro cada dos vueltas:

```
#include <stdio.h>
#include <wiringPi.h>

// LED Pin - wiringPi pin 0 is BCM_GPIO 17.

#define LED      0
#define ORIENT   4 // GPIO 23

int main (void)
{
    printf ("Raspberry_Pi_blink\n") ;

    wiringPiSetup () ;
    pinMode (LED, OUTPUT) ;
    pinMode (ORIENT, OUTPUT);

    int time =20; // para 50hz poner time=10
    int i = 1;
    int ori = 0;
    digitalWrite(ORIENT, HIGH);
    for (;;)
    {
        digitalWrite (LED, HIGH) ; // On
```

¹Muy importante conectar todas las líneas de tierra entre sí para tener toma de masa compartida en todos los componentes.

```

delay (time) ;           // mS
digitalWrite (LED, LOW) ; // Off
if(i % 94 == 0){
    ori = 1 - ori;
    if(ori == 0)
        digitalWrite(ORIENT, HIGH);
    else    digitalWrite(ORIENT, LOW);
    printf("Cambiamos la orientacion a %i\n", ori);
}
delay (time) ;
printf("%d\n", i);
i++;
}
return 0 ;
}

```
