

# Memoria Práctica 2

Robótica

UCM - FDI

Luis Javier Cabrera Sagbay; Santiago Gómez Muñoz; Ángel Luis Ortiz Folgado; Víctor Vicente Sánchez

## Desarrollo de la práctica:

### Familiarización con la Raspberry Pi. Compilación y ejemplo sencillo

Empezamos la práctica conectándonos a la Raspberry, mediante SSH. Después de realizar el montaje, del circuito con el LED, y la resistencia, conectamos la Raspberry, compilamos el programa `blink.c` y ejecutamos.

En este punto, descubrimos que la Raspberry estaba mal montada y fallaba, y por tanto, no funcionaba. Entregaba corriente, pero no funcionaban las salidas.

Cambiamos la Raspberry, volvimos a realizar el montaje, compilamos y ejecutamos el programa y observamos que funcionaba, ya que el LED, parpadeaba, cambiamos la frecuencia de parpadeo y observamos como el LED parpadeaba, más deprisa o más despacio.

### Generación de PWM y uso de un servomotor.

Para este apartado, primero realizamos la conexión del motor y ejecutamos el programa `blink.c`, de esa manera, el motor, debería girar sólo cuando parpadeara, el led, pero eso no sucedía. Descubrimos que había que poner una tierra común para el motor y para la Raspberry, ya que si no se producían pequeñas diferencias de tensión, entre ambas tierras y no se obtenía el comportamiento deseado.

A la hora de hacer el programa, hicimos la llamada a la función `softPwmCreate (int PIN, int iniVal, int RANGO)` con parámetros: `pin = 0`, `iniVal = 0` y `RANGO = 100`.

Para la función `softPwmWrite (int PIN, int VALOR)` para encontrar que rangos hacían girar el servo en cada sentido, decidimos realizar un bucle que fuera probando valores de 0 al valor de Rango y los fuera mostrando, en la pantalla. Compilamos el programa mediante la orden: `gcc -o ejemplo -l wiringPi ejemplo.c` y lo ejecutamos.

Realizamos varias pruebas con distintos rangos, los resultados fueron:

Rango	Giro sentido horario	Parada	Giro sentido antihorario
100	[0,17]	[18]	[19,100]
50	[0,17]	[18]	[19,100]
10	[0,6]	[7]	[8,10]

### Código del programa realizado:

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

int main(void){
    wiringPiSetup();
    int i;
    char a;
    for(i=0; i<100; i++){
        softPwmCreate(0, 0, 50);
        printf("%d",i);
        softPwmWrite(0, i);
        getchar();
    }
    return 0;
}
```

Al llegar al valor que para el servo, se podía apreciar que aunque el servo no giraba, se seguía escuchando el ruido que hacía el motor. Esto se debe a que el voltaje no es estable del todo; lo que provoca que el servo vibre y emita un leve sonido.

Para parar el motor usaremos los valores indicados anteriormente que paran el motor que dependerán del rango establecido. A pesar del problema descrito anteriormente.