Práctica - Motores II

Daniela Alejandra Córdova Porta

Richard Junior Mercado Correa

UNIVERSIDAD COMPLUTENSE DE MADRID

Madrid, diciembre de 2020

Índice

Resumen

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

1. **Objetivos**

- Conocer funcionamiento de los servomotores y PWM.
- Aprender el montaje, control y uso de los servomotores.
- Diseño y construcción inicial del robot (plataforma robótica).

2. **Procedimiento**

   a. **Experimento I. Uso de servomotor (Actuador del Robot)**

Conectamos la RaspBerry Pi a través del cable Ethernet y usamos el VNC viewer para poder realizar la práctica. Nos ubicamos en la carpeta *wiringPi* y *examples*. Usando *softPWM.c*, observamos lo que necesitamos para que el servomotor funcione.

Usando la librería *#include <wiringPi.h>* e *#include <softPwm.h>*, obtenemos las funciones necesarias para crear el PWM que queremos. Estas funciones son *softPwmCreate (int PIN, int iniVal,int RANGO)* que nos permite inicializar la rutina para generar el PWM en el pin que indicamos con un rango establecido. Decidimos hacerlo con un rango de 0 a 100 y cómo en este primer procedimiento sólo usamos un servomotor, usamos el PIN 0 que sería el GPIO 17.

La otra función que usamos es *softPwmWrite (int PIN, int VALOR)* que actualiza la salida del PWM en el PIN que indicamos, en este caso el PIN es el 0.

Código:

```
1.  #include <stdio.h>
2.  #include <errno.h>
3.  #include <string.h>
4.  #include <wiringPi.h>
5.  #include <softPwm.h>
6.  #define RANGE 100
7.  #define PIN0 0
8.  #define PIN1 1
9.
10. int main()
11. {
12.     wiringPiSetup();
13.
```

```
    softPwmCreate(PIN0, 0, RANGE);

    for (;;)
    {
        softPwmWrite(PIN0, 20);
    }
}
```

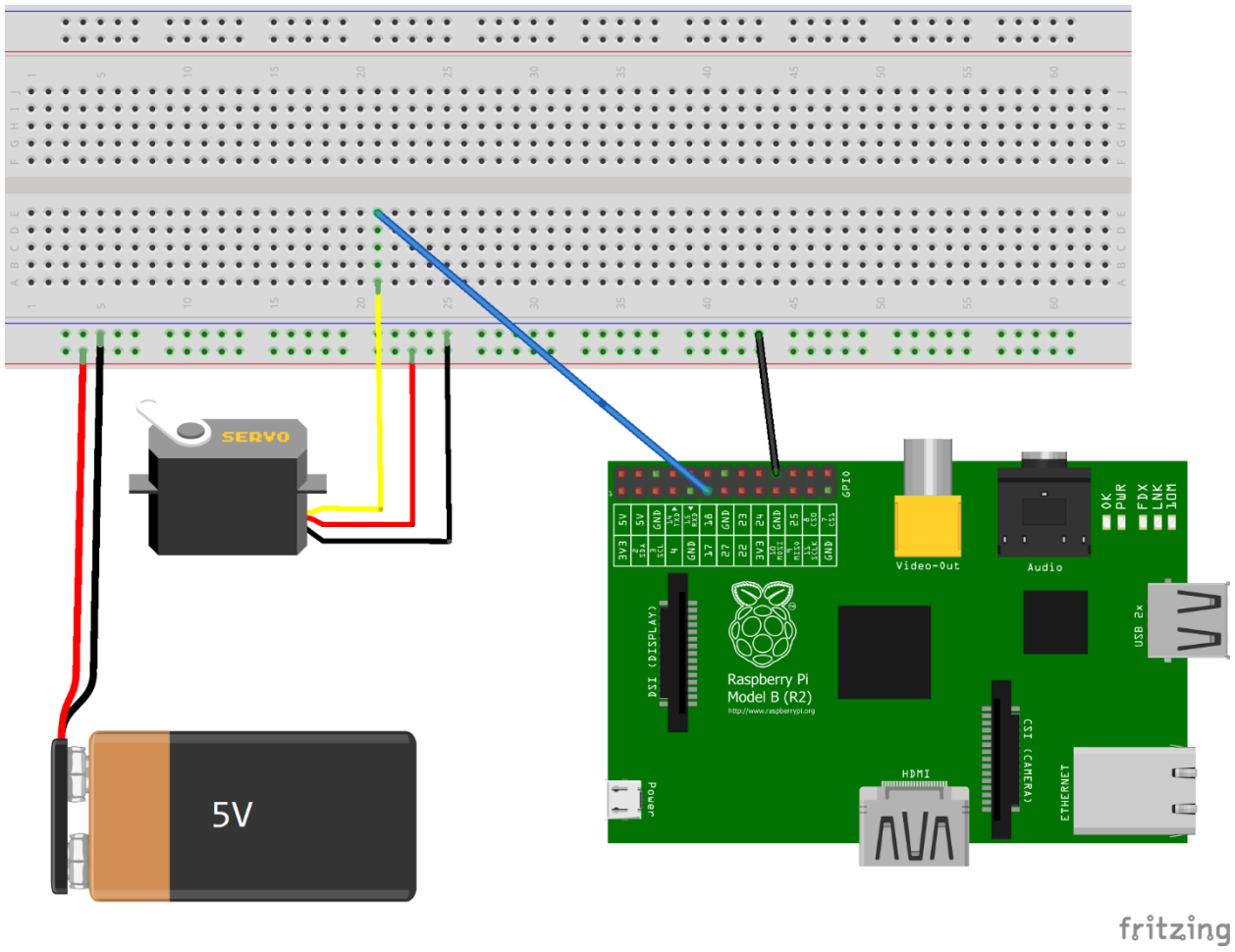

Fig1. Montaje 1 para el Experimento 1

b. Experimento II. Montaje de los motores en el robot

Código:

En el código siguiente se muestra que programa se usó para que el robot recorra los 2 metros en línea recta. Se usaron los valores 10 y 35 para cada uno de los servos ya que estos tienen igual velocidad pero sentido contrario, que es lo que se buscaba.

```c
#include <stdio.h>
#include <errno.h>
#include <string.h>
#include <wiringPi.h>
#include <softPwm.h>
#define RANGE 100
#define PIN0 0
#define PIN1 1

int main()
{
    wiringPiSetup();

    softPwmCreate(PIN0, 0, RANGE);

    softPwmCreate(PIN1, 0, RANGE);

    for (;;)
    {
        softPwmWrite(PIN0, 10);
        softPwmWrite(PIN1, 35);
    }
}
```

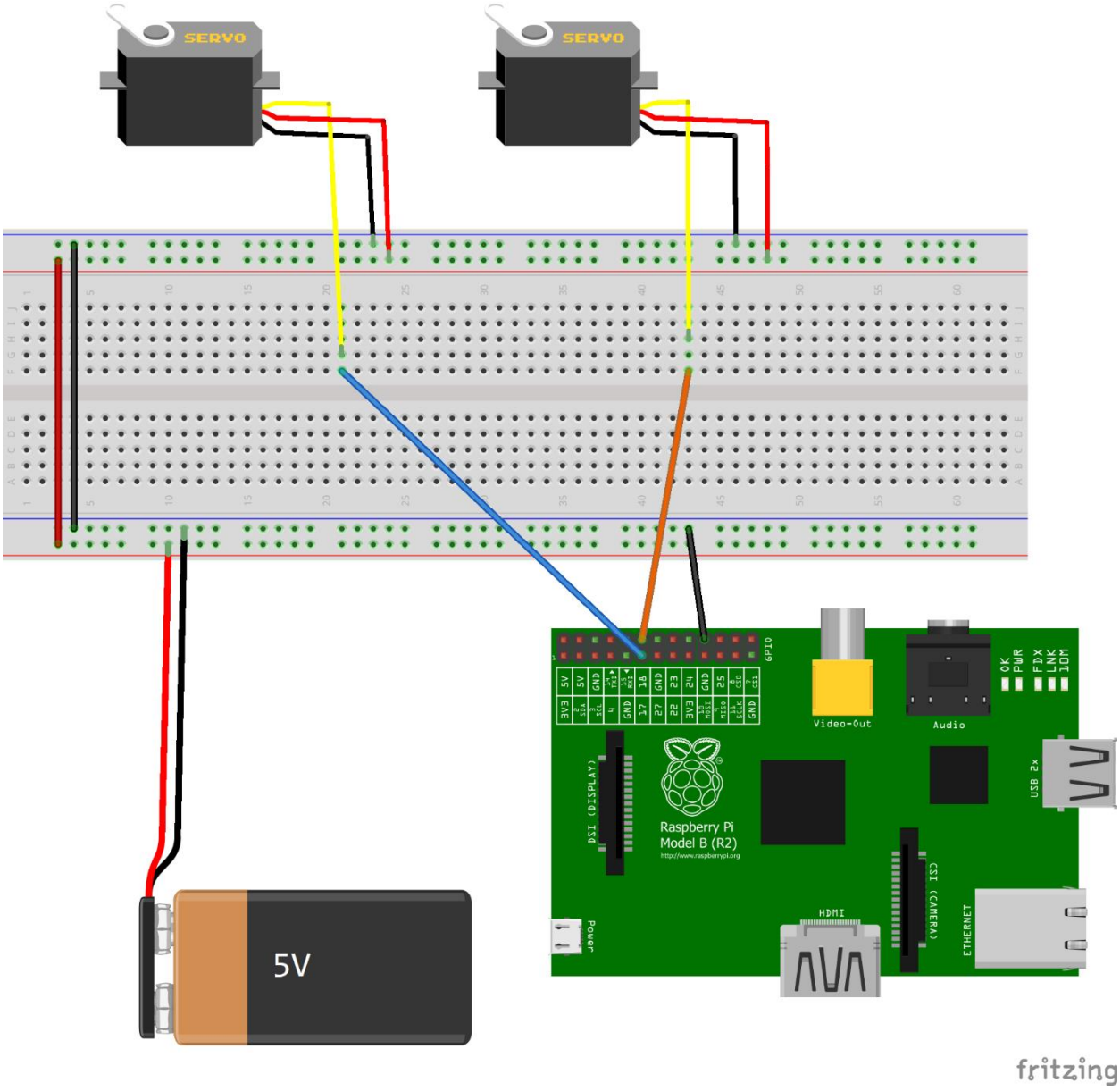

Fig 2. Montaje 2 para el Experimento 2

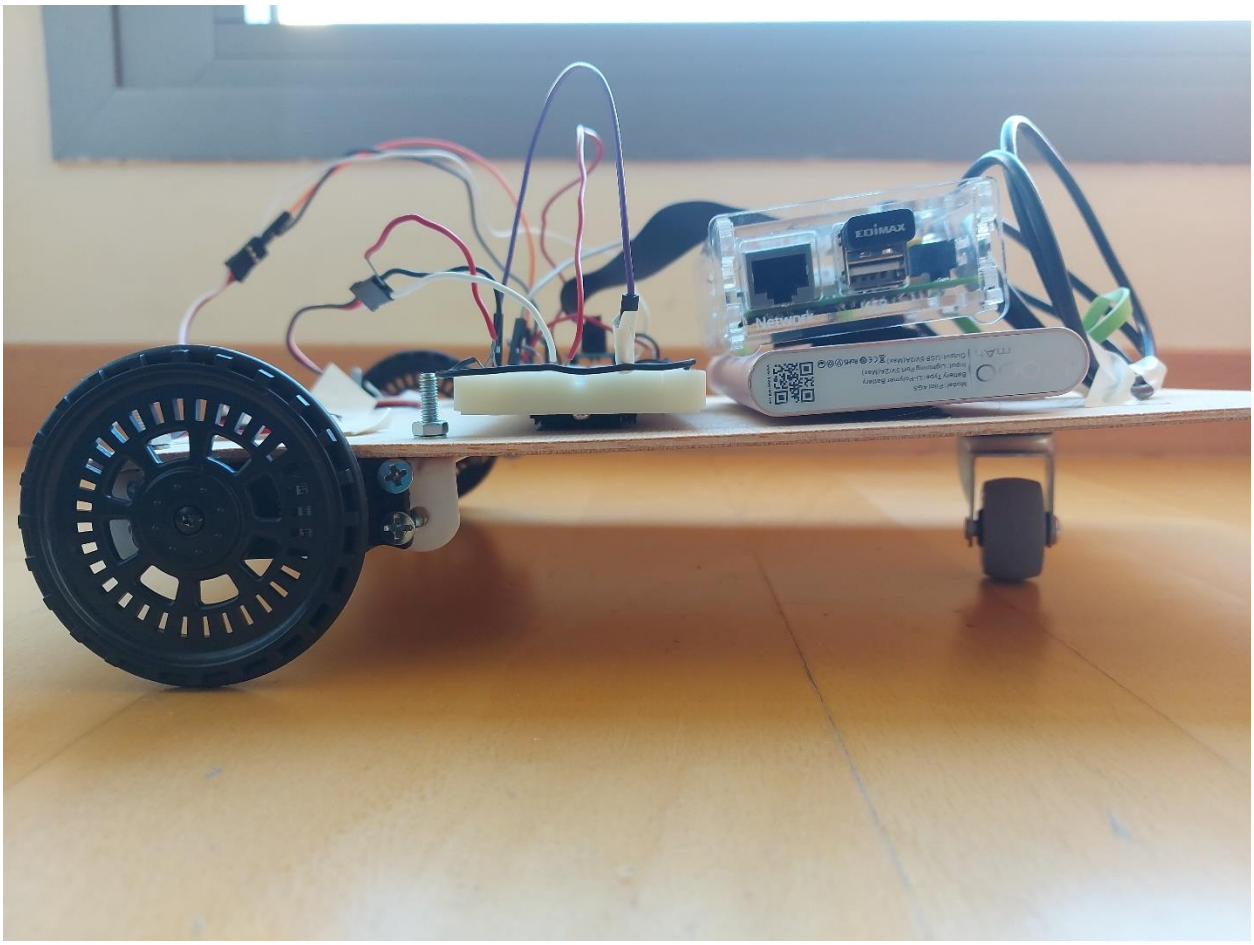

Fig 3. Primer montaje del Robot.

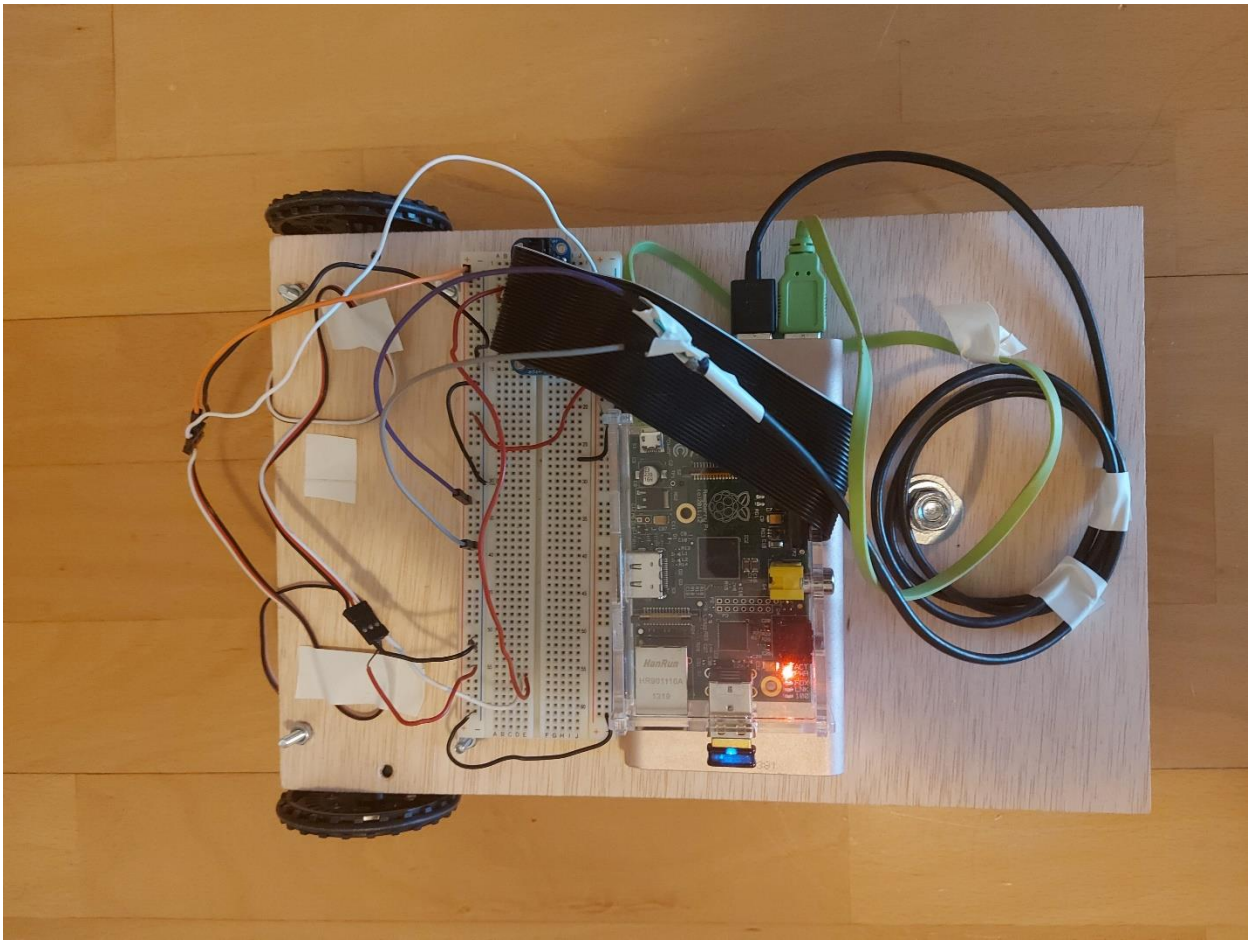

Fig 4. Primer montaje del Robot.

Para el montaje se tuvo en cuenta las cuestiones siguientes:

- Coloque sobre la plataforma el circuito anterior y la Rasp Pi para mover los motores a la vez. Si el diseño y sujeción de los motores es el adecuado debe ser capaz de moverse en línea recta sin desviarse.
- Como mínimo debería recorrer un tramo de unos 2 m con una desviación máxima de un 10% hacia cualquiera de los dos lados, como se indica en la siguiente figura. Procure mover el robot una distancia determinada a partir del tiempo que está en funcionamiento los motores (control en lazo abierto).
- Pruébelo con la distancia anterior y ver si es capaz de parar a los 2 m.

Se decidió por una tabla de manualidades dónde cupiese la powerbank, protoboard, raspberry pi y cables necesarios. En la misma se abrieron 4 huecos en su parte delantera, 2 en cada lado y se atornillaron las ruedas de la siguiente manera:

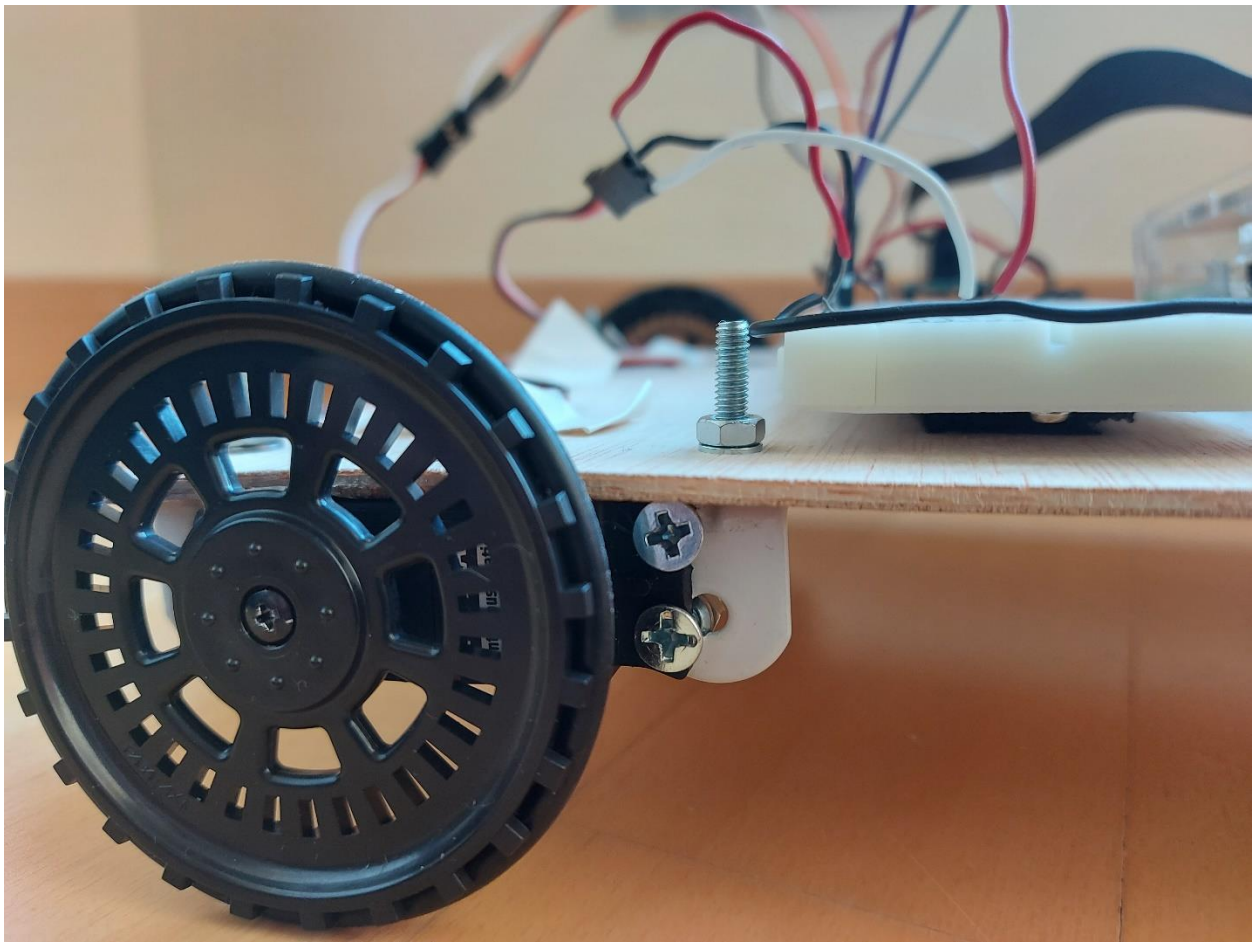

Fig 5. Montaje de las ruedas laterales.

Para tener un buen apoyo en la parte posterior, se usa la siguiente rueda (“rueda loca”) para que esta solo ayude en el movimiento:

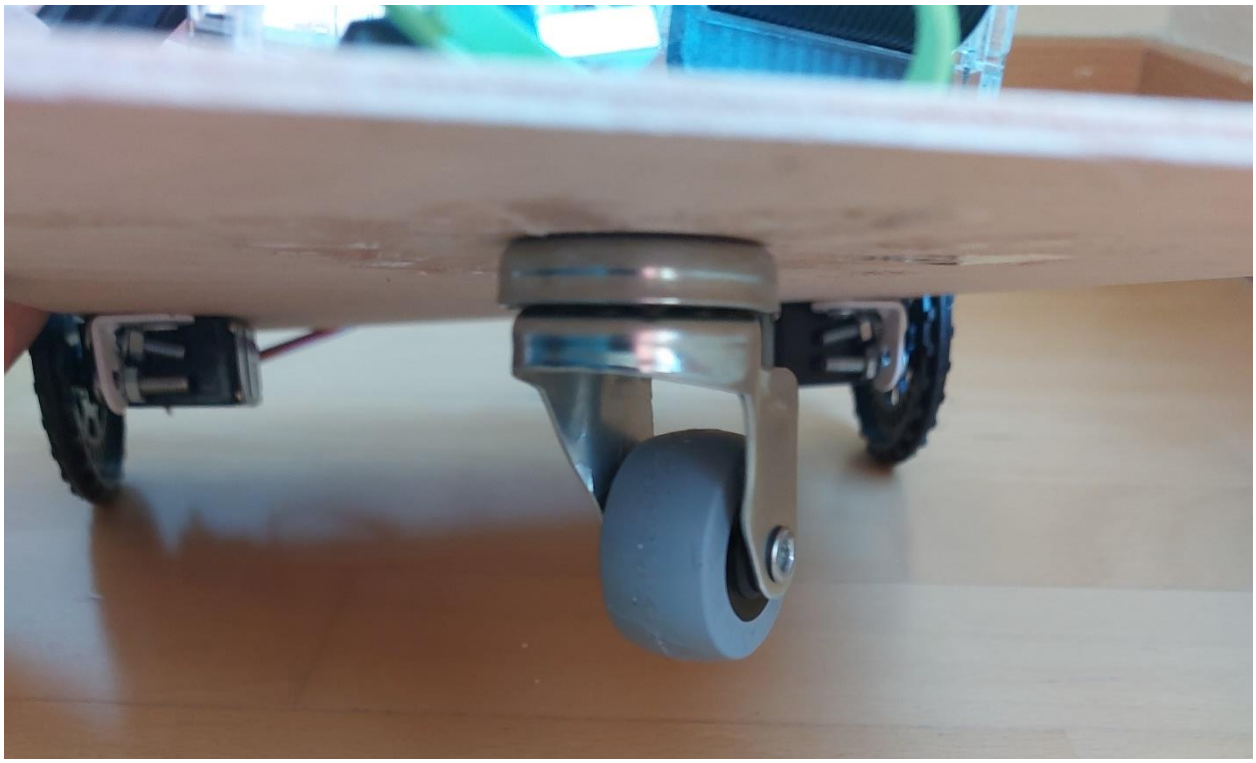

Fig 6. Montaje de la rueda loca.

Los demás equipos fueron pegados con cierre mágico y los cables con cinta adhesiva.

Luego de tener el robot montando, se configuró la red Wifi en la casa usando la conexión compartida del móvil, estableciendo su nombre como *labrob* por facilidad. Finalmente usando el comando *ifconfig* se obtuvo la nueva IP para realizar el experimento.

A continuación, se encuentra la grabación del robot recorriendo 2m en línea recta usando un metro para visualizar la desviación posible y lo recorrido. Se colocó el inicio del robot en los 120 cm indicados del metro:

Links:

- **Video 1:**

  https://drive.google.com/file/d/1G9zbtj0TEuWompIAKQBrdoUAU7JptKjP/view?usp=sharing

- **Video 2:**

https://drive.google.com/file/d/1H5nXpLoA2UNQDYHl-AOWigoAwU7IRBdz/view?usp=sharing

Se tarda aproximadamente 9 segundos en recorrer este tramo, por lo cual para ver si se para tuvimos que agregar un delay de 9 segundos y luego realizar un *softPwmWrite (0,0)* y *softPwmWrite (1,0).*

Código

```
#include <stdio.h>
#include <errno.h>
#include <string.h>
#include <wiringPi.h>
#include <softPwm.h>
#define RANGE 100
#define PIN0 0
#define PIN1 1

int main()
{
    wiringPiSetup();

    softPwmCreate(PIN0, 0, RANGE);

    softPwmCreate(PIN1, 0, RANGE);

    for (;;)
    {
            softPwmWrite (0, 10) ;
            softPwmWrite (1, 35) ;
            delay(9000);
            softPwmWrite (0, 0) ;
            softPwmWrite (1, 0) ;
            delay(10000);
    }
}
```

Links video : https://drive.google.com/file/d/1H5xExudNcL5NfhzR7RT9vjX-z2oMacNz/view?usp=sharing

3. Resultados y Análisis de Resultados

a. Experimento I. Uso de servomotor (Actuador del Robot)

a) Conectar el motor anterior a la RaspBerry Pi y moverlo mediante el programa creado. Verifique el valor necesario que debe utilizar en el programa para mover las ruedas (máximo dos velocidades) en ambos sentidos.

Usando el código descrito anteriormente y probando con distintos valores en la función *softPwmWrite* pudimos observar qué pasa cuando le administras ciertos valores:

- **0:** el servo se mantiene parado.

- **20 hasta el 90:** el servo gira en un mismo sentido, pero a mayor valor dado, más lento gira.

- **10:** gira en sentido contario con la misma velocidad que el valor 35

El PWM permite variar la energía que recibe el servomotor variando la energía que recibe. Una variación en el PWM produce un cambio en el Ciclo de trabajo $D = \frac{\tau}{\mathrm{T}}$

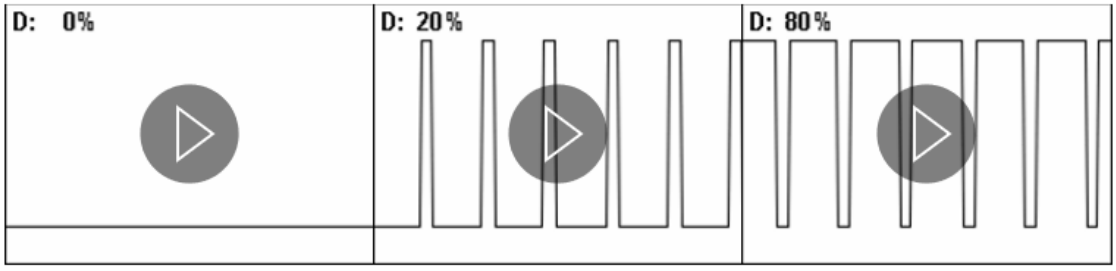

Fig 7. Dibujo PWM.

τ duración donde la función está en nivel alto (normalmente cuando la función es mayor que cero).

T = Período de la función.

El valor en *softPwmWrite* explica cuánto dura cada ciclo en unidades de pulso básicas. Como declaramos el rango de 0 a 100 (softPwmCreate (0,0,100)), esto quiere decir que crea un ciclo de 10 ms de duración compuesto por 100 pasos. Luego si coloco un valor con *softPwmWrite* (1,20), dice que mantenemos el pulso alto durante 2 ms en cada ciclo de 100 ms.

Se decidió *softPwmWrite* (0,10) y *softPwmWrite* (1,35) para los servos al tener estos dos la misma velocidad, pero en sentidos distintos.

b) Comprobar que puede para el motor desde el programa. ¿Se para totalmente? Justifique los resultados obtenidos.

Sabemos que en estos servos el punto de parada es cuando los dos anchos del PWM (alto y bajo) son iguales. Entonces tendríamos que darle un valor de “0” o “100” para que pare en *softPwmWrite* al pin deseado.

b. Experimento II. Montaje de los motores en el robot

a) ¿Puede girar también el ángulo que se desee? Indique cómo lo haría.

Para girar podemos parar las ruedas y dependiendo del giro que queremos hacer, una de las ruedas se queda parada y a la otra se le da un valor dependiendo del giro:

Si queremos girar:

- 90° a la derecha: se le da al servo izquierdo un valor de 35 durante un tiempo determinado. En este caso son 2 segundos
- 90° a la izquierda: se le da al servo derecho un valor de 10 durante un tiempo determinando. En este caso son 2 segundos

Y para otros ángulos se pueden sacar cuentas a partir de los resultados anteriores, siempre calculando en base al tiempo que necesita (“delay” en el código).

Código para girar 90° a la derecha

```
#include <stdio.h>
#include <errno.h>
#include <string.h>
#include <wiringPi.h>
#include <softPwm.h>
#define RANGE 100
#define PIN0 0
#define PIN1 1

int main()
{
    wiringPiSetup();

    softPwmCreate(PIN0, 0, RANGE);

```

```
    softPwmCreate(PIN1, 0, RANGE);

    for (;;)
    {
            softPwmWrite (0, 0) ;
            softPwmWrite (1, 35) ;
            delay(2000);
            softPwmWrite (0, 0) ;
            softPwmWrite (1, 0) ;
            delay(10000);
    }
}
```

Link para el video:

https://drive.google.com/file/d/1H7Of0lPWlGEAE3RJDtdsQuCl-s_qhqFd/view?usp=sharing

4. Discusión y conclusiones

a. Conclusiones:

- A mayor valor dado para el pulso PWM en *softPwmWrite*, menos velocidad tiene.
- Para detener el servo se le tiene que dar un valor donde el alto y bajo de la señal PWM es igual, en este caso sería los valores 0 y 100
- Para realizar rotaciones se debe parar uno de los servos y el otro debe avanzar en un tiempo determinando, dependiendo del ángulo.
- Los motores deben estar al frente de la plataforma para que ellos decidan la dirección.

b. Errores:

- Se deben ajustar correctamente los motores a la plataforma, sino estos no mantienen una dirección adecuada y el robot se inclinaría constantemente.
- Al ser los servos iguales pero colocados de manera contraria en la plataforma, deben tener sentidos distintos en el código. Por ello, a uno se le da el valor de 35 y al otro de 0. Esto trae otros errores como no saber si el valor asignado al servo de 35 sea igual al de 0. Para ello se tuvieron que realizar varias pruebas viendo las diferencias de velocidad con

diferentes valores hasta que se observaba que estaban en la misma velocidad. Si no se tomaba en cuenta esto, el robot se inclinaba a la izquierda o derecha dependiendo de cuál servo va a mayor velocidad.

- La pila al ser pesada se tiene que colocar en una posición específica, si no el robot se inclina por alguno de los dos lados, dependiendo de su ubicación.