

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Laboratorio de Fundamentos Embebidos

Práctica 5:

Uso de sensores y motores en Raspberry pi

Integrantes:

Cárdenas Cárdenas Jorge

Garrido Sánchez Samuel Arturo

Murrieta Villegas Alfonso

Reza Chavarría Sergio Gabriel

Valdespino Mendieta Joaquín

## Práctica 5: Uso de sensores y motores en Raspberry pi

### Objetivo:

Implementar el uso de algunos sensores y el control de motores para el proyecto de casa inteligente.

### Introducción:

Una de las mayores ventajas que tienen los sistemas embebidos es la versatilidad al adaptarse a una gran variedad de sensores y actuadores.

Anteriormente, vimos que un sensor ultrasónico es un dispositivo electrónico que mide la distancia de un objeto objetivo mediante la emisión de ondas sonoras ultrasónicas y convierte el sonido reflejado en una señal eléctrica.

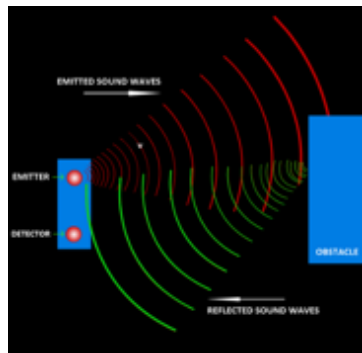


Imagen 1: Funcionamiento del sensor ultrasónico

Las ondas ultrasónicas viajan más rápido que la velocidad del sonido audible (es decir, el sonido que los humanos pueden escuchar). Los sensores ultrasónicos tienen dos componentes principales: el transmisor (que emite el sonido mediante cristales piezoeléctricos) y el receptor (que encuentra el sonido después de que ha viajado hacia y desde el objetivo).

Además y como parte del hardware que usaremos en esta práctica, un motor de CD es una máquina eléctrica que convierte la energía eléctrica en energía mecánica, de manera general la forma en la que trabaja es a través de un campo magnético que surge en el entre hierro cuando se energiza la bobina de campo del motor. El campo magnético creado está en la dirección de los radios de la armadura, es decir, entra inducido desde el lado del polo norte de la bobina de campo y "sale" por el lado del polo sur de la bobina de campo.

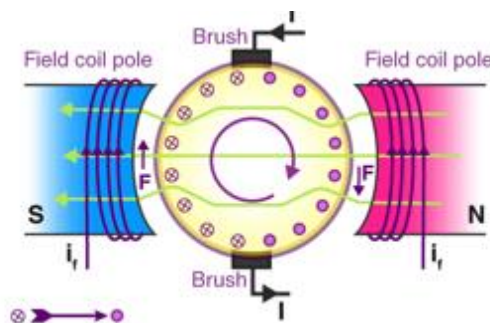


Imagen 2: Funcionamiento de un motor de DC

## Desarrollo y resultados:

### 1. Uso de sensor ultrasónico

A continuación, se muestra el diagrama de conexión con nuestra Raspberry Pi, además del código encargado de llevar a cabo la obtención de la distancia mediante el trigger y el echo de nuestro sensor:

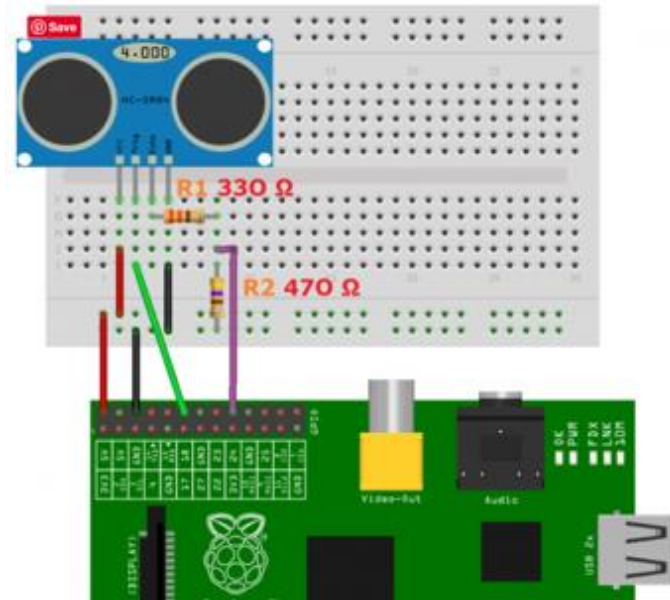


Imagen 3: Diagrama de conexiones para la implementación de nuestro sensor ultrasónico

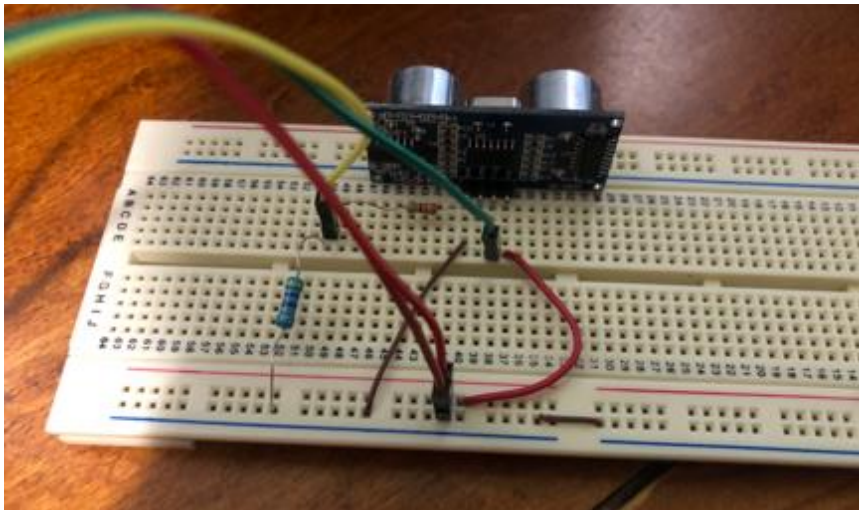


Imagen 4: Conexiones para la implementación de nuestro sensor ultrasónico

Como vimos en la práctica anterior, para obtener la distancia entre el echo y trigger de nuestro sensor, simplemente debemos realizar ya sea manualmente una diferencia entre dos señales, o directamente emplear una función como la que se incluye en gpizero.

Para el caso de nuestro ejercicio, empleamos una la función que hicimos en la práctica previa, a continuación el código fuente:

```

1 import RPi.GPIO as GPIO
2 import time
3
4 #GPIO Mode (BOARD / BCM)
5 GPIO.setmode(GPIO.BCM)
6 GPIO_TRIGGER = 17
7 GPIO_ECHO = 27
8
9 #set GPIO direction (IN / OUT)
10 GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
11 GPIO.setup(GPIO_ECHO, GPIO.IN)
12
13 def distance():
14     # set Trigger to HIGH
15     GPIO.output(GPIO_TRIGGER, True)
16
17     # set Trigger after 0.5ms to LOW
18     time.sleep(0.00001)
19     GPIO.output(GPIO_TRIGGER, False)
20     StartTime = time.time()
21     StopTime = time.time()
22
23     # save StartTime
24     while GPIO.input(GPIO_ECHO) == 0:
25         StartTime = time.time()
26
27     # save time of arrival
28     while GPIO.input(GPIO_ECHO) == 1:
29         StopTime = time.time()
30
31     # time difference between start and arrival
32     TimeElapsed = StopTime - StartTime
33     distance = (TimeElapsed * 34300) / 2
34
35     return distance
36
37 if __name__ == '__main__':
38     try:
39         while True:
40             dist = distance()
41             print ("Measured Distance = %.1f cm" % dist)
42             time.sleep(1)
43     except KeyboardInterrupt:
44         print("Measurement stopped by User")
45         GPIO.cleanup()

```

Imagen 5: Función de obtención de distancia mediante un sensor ultrasónico

### NOTA 1:

Para el caso de este ejercicio empleamos la biblioteca de Rpi para la manipulación de los puertos de nuestro Raspberry.

A continuación, los resultados obtenidos tras ejecutar nuestro código:

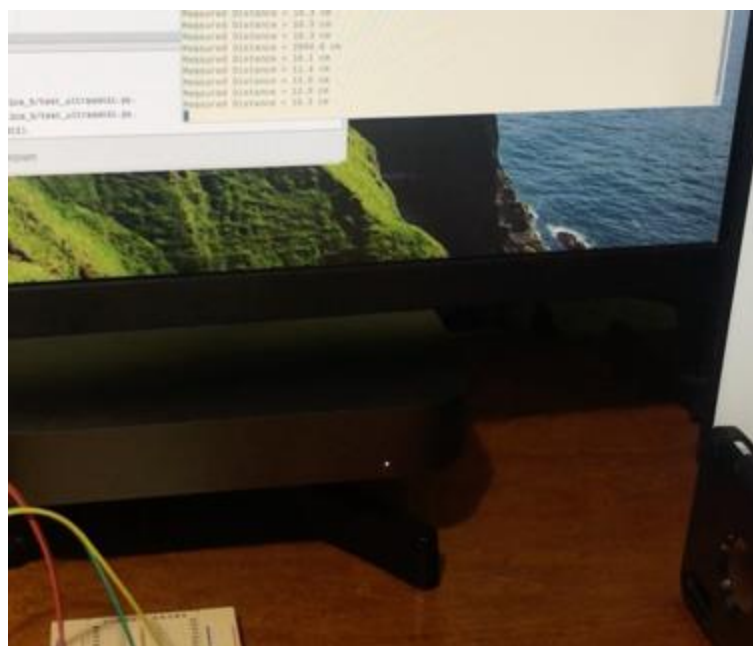
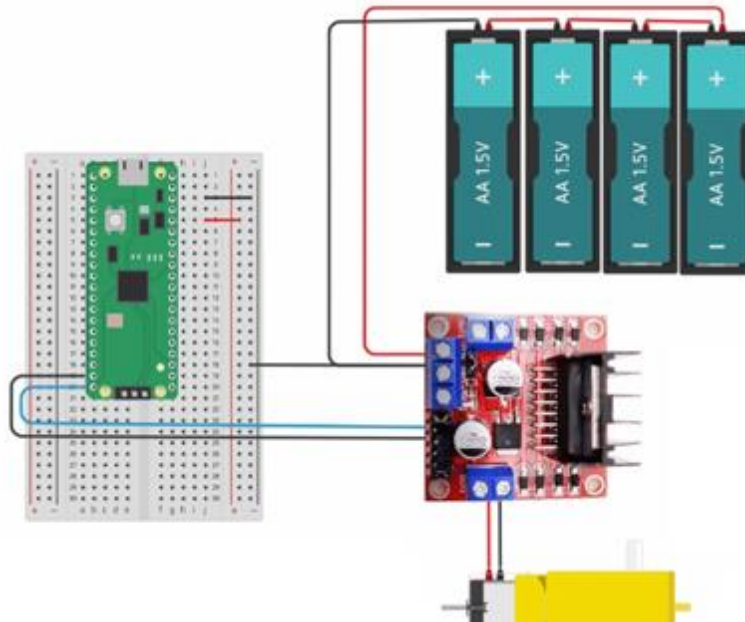


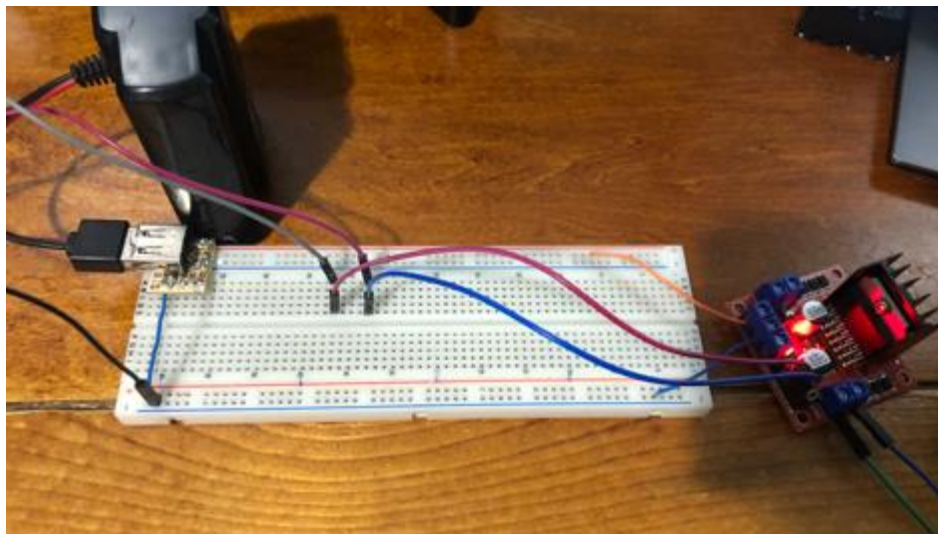
Imagen 6: Funcionamiento del sensor ultrasónico, se observa en la parte superior la obtención de la distancia, y en la parte inferior el sensor en la protoboard

## 2. Uso de motor mediante Raspberry Pi

A continuación, se muestra el diagrama de conexión con nuestra Raspberry Pi, además del código encargado de llevar a cabo el movimiento de nuestros motores



*Imagen 7: Diagrama de conexiones para la implementación de nuestro motor y puente H*



*Imagen 8: Conexiones para la implementación de nuestro motor de CD además de las conexiones realizadas en nuestro puente H*

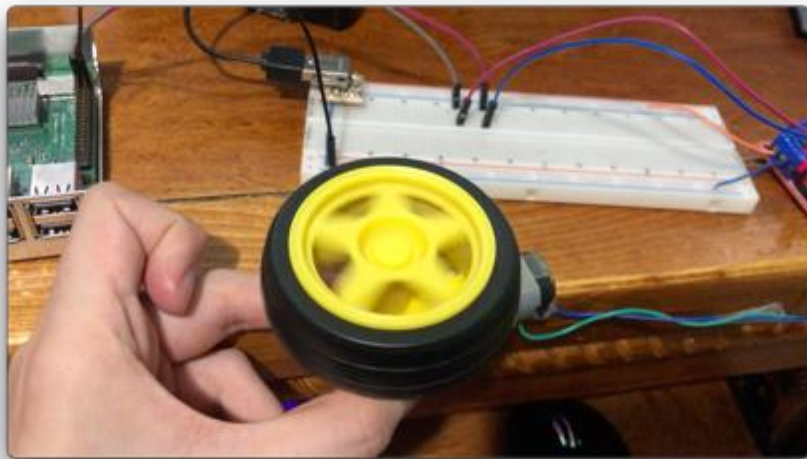
Para poder emplear nuestro motor CD es necesario considerar un puente H por dos principales razones, la primera es debido a la potencia que se requiere para poder mover a nuestro motor (Por ello la alimentación de nuestro motor a través del puente H es mediante una batería de 9v) y la segunda es por la forma en que debemos hacer funcionar a nuestro motor analógicamente.

Para el caso de nuestro ejercicio, empleamos el objeto “motor” de la biblioteca gpiozero, donde solamente usamos los métodos forward y backward para mover hacia adelante o hacia atrás el motor, a continuación, el código de este ejercicio:

```
1 from gpiozero import Motor
2 from time import sleep
3
4 motor = Motor(forward=4, backward=14)
5
6 while True:
7     motor.forward()
8     sleep(5)
9     motor.backward()
10    sleep(5)
```

*Imagen 9: Código para el movimiento del motor*

A continuación, los resultados obtenidos tras ejecutar nuestro código, se recomienda ampliamente ver el vídeo para observar el movimiento de nuestro motor:



*Imagen 10: Funcionamiento del motor*

## Conclusiones:

### Cárdenas Cárdenas Jorge

Para un sistema embebido el uso de sensores para recabar información del medio es primordial; incluso se podría que el sistema estaría incompleto si no tuviera sensores y actuadores conectados a él. La cantidad de sistemas que se pueden diseñar donde intervengan sensores y/o motores es enorme, los encontramos en una gran cantidad de aplicaciones, por ejemplo, puede crearse puertas automáticas que se abran o cierren en función de si hay o no alguien, o en cortineros que se controlen de forma remota, entre muchas otras aplicaciones más. En este sentido, los módulos de Raspberry Pi en Python proveen un conjunto de funciones ya integradas que permiten manipular los sensores y motores de forma muy práctica y sencilla, haciendo que como desarrolladores solo nos preocupamos por la lógica del programa y no por ver cómo hacer girar un motor.

### Garrido Sánchez Samuel Arturo

La práctica nos permite amplificar nuestros conocimientos con el uso de otras herramientas como son las entradas o lecturas de sensores de la Raspberry Pi. GPIOZero nos permite manipular las entradas de 3.3V para saber qué lectura hay sobre el sensor y así saber qué nos quiere decir. Con el uso de motores es necesario utilizar puentes H, por



ejemplo, el L298 que se utilizará para las bombas de agua, es necesaria una fuente externa o la de 5V de la Raspberry si el motor lo permite en las entradas del puente H. Por el momento no se manipula velocidades con el PWM de la Raspberry conectada al puente H, pero puede ser una posibilidad.

### **Murrieta Villegas Alfonso**

En la presente práctica aprendimos a emplear diferentes sensores con nuestra Raspberry Pi y Python como lenguaje de programación, principalmente observamos que además de emplear bibliotecas hechas como Gpiozero o RPi, también podemos desarrollar funciones destinadas a entender las salidas y entradas de nuestros sensores.

Por otro lado, mediante el uso de diferentes sensores sabemos que poco a poco nos acercamos a completar todos los requisitos indispensables de nuestro futuro proyecto.

### **Reza Chavarria Sergio Gabriel**

Mediante la implementación de motores se puede dar la interacción con elementos más complejos, como lo son el movimiento del sistema por medio de ruedas o manejo de elementos externos como dispensadores de agua.

Además, el uso de diferentes tipos de sensores implica la obtención de información del ambiente para poder hacer uso de estas, como la obtención de datos para su estudio, realizar acciones específicas por los sensores, interrumpir un proceso o realizar notificaciones o alertas a los usuarios.

### **Valdespino Mendieta Joaquín**

En la presente practica se pudo aplicar de forma acumulativa lo visto en anteriores prácticas, con el fin de mantener un flujo de datos, con los diversos dispositivos que interactúan con el ambiente, como es el caso de los sensores y aquellos que ejercen una tarea específica como es el caso de los motores, con ello podemos observar la utilidad y escalabilidad que pueden tener los proyectos, ya que podemos utilizar diversos dispositivos con un propósito y tarea específico, todo ello gracias a la implementación de las bibliotecas como Gpiozero o Rpi, que nos permiten dicha comunicación, empezando con las definición de entradas y salidas, pasando por el procesamiento y terminando con una respuesta.

### **Referencias:**

1. GPIO ZERO Docs. *gpiozero*. Recuperado el 3 de Octubre de 2021, de <https://gpiozero.readthedocs.io/en/stable/>

2. Raspberry Pi tutorials. *Using a RaspberryPi Distance sensor*. Recuperado el 3 de octubre de 2021, <https://tutorials-raspberrypi.com/raspberry-pi-ultrasonic-sensor-hc-sr04/>