

TRABAJO FINAL CLOUD COMPUTING Y CLOUD ROBOTICS. 2024

Integrantes: Julian Order y Gastón Ginestet.

Introducción

Este informe detalla el desarrollo del proyecto "**Integración de Sensores y Actuadores con RaspiRobot y Node-RED**", cuyo objetivo fue controlar un robot utilizando sensores, actuadores y flujos diseñados en Node-RED. Se describen los componentes utilizados, los flujos creados y las soluciones implementadas ante los desafíos encontrados, proporcionando una visión general del proceso y los resultados obtenidos.

Desarrollo

Descripción de Sensores y Actuadores:

- Sensores

Sensor Ultrasónico HC-SR04

- **Propósito:**
El sensor ultrasónico HC-SR04 se utiliza para medir distancias mediante el uso de ondas de sonido.
- **Funcionamiento básico:**
El sensor funciona emitiendo pulsos ultrasónicos a 40 kHz desde el pin *Trigger*. Cuando el pulso rebota en un objeto y regresa al pin *Echo*, el tiempo transcurrido entre la emisión y la recepción del pulso se usa para calcular la distancia al objeto. La idea es que en node-red, cada 0.5 seg mande la info de si detectó o no movimiento/algo por delante. Osea, si detectamos que en el msg.payload < 60 le seteamos una variable de estado "freno" en 1, indicando que debe frenar.
- **Conexión:**
El HC-SR04 está conectado al frente del robot, utilizando los siguientes pines GPIO de la Raspirobot:
 - **Pin Trigger:** GPIO 40.
 - **Pin Echo:** GPIO 38.
 - **Alimentación (Vcc):** Pin de 5V.
 - **GND:** Pin de tierra (GND).

Utilizamos a su vez, la librería de `node-red-node-pisrf` para poder realizar la conexión de manera eficiente ya que de manera manual detectamos que nos faltaba muchas configuraciones, se nos presentó por ej el problema de ir tomando cada cierto tiempo la distancia, concurrencia, ya que la idea es que el robot frene, etc.

- **Esquema o Foto:**



- Actuadores

Servo Motor

- **Propósito:**
Controlar movimientos angulares específicos para permitir que el robot avance, retroceda o ajuste su dirección según las instrucciones del sistema de control.
- **Funcionamiento básico:**
El servo motor funciona como un actuador que interpreta señales PWM (Modulación por Ancho de Pulso) para mover su eje a una posición específica o controlar su velocidad. En este caso, se utiliza en combinación con otros pines de configuración para gestionar el movimiento hacia adelante, hacia atrás o detener el robot.
 - Los pines **RIGHT_1_PIN** y **RIGHT_2_PIN** determinan la dirección del movimiento:
 - 1 y 0 para avanzar.
 - 0 y 1 para retroceder.
 - El pin **RIGHT_PWM_PIN** controla la intensidad de movimiento basada en el ángulo y la distancia configurados en el flujo de Node-RED.

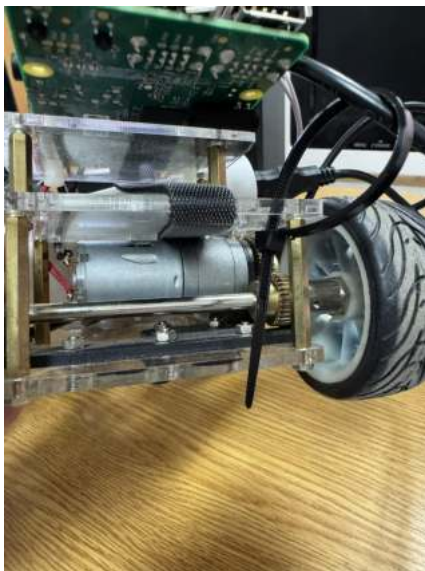
La lógica implementada incluye una verificación adicional que evalúa el estado de la variable global *“freno”*. Si *“freno”* está activado y la dirección solicitada es *“up”* (hacia adelante), el sistema establece el movimiento en cero para evitar el avance. Si no hay restricciones, el valor de movimiento se configura en función de la distancia proporcionada por el sensor.

- **Conexión:**

El servo motor está conectado a la Raspirobot utilizando los siguientes pines GPIO:

- **RIGHT_PWM_PIN:** GPIO 14, encargado de recibir señales PWM para regular el movimiento.
- **RIGHT_1_PIN:** GPIO 10, define una de las configuraciones de dirección.
- **RIGHT_2_PIN:** GPIO 25, define la segunda configuración de dirección.
- **Vcc:** Conectado al pin de 5V para la alimentación del motor.
- **GND:** Conectado al pin GND para cerrar el circuito.

- **Esquema o Foto:**



Servo para Controlar la Dirección

- **Propósito:**

Modificar la dirección del robot ajustando el ángulo de las ruedas delanteras o de un eje de control. Esto permite que el dispositivo se oriente hacia la izquierda, derecha o mantenga una posición neutra.

- **Funcionamiento básico:**

Este servo utiliza señales PWM (Modulación por Ancho de Pulso) para controlar su posición angular. Dependiendo de la duración del pulso enviado desde la Raspirobot, el eje del servo se ajusta a un ángulo específico:

- **Ángulo 5:** Dirección hacia la izquierda (*left*).
- **Ángulo 9:** Dirección hacia la derecha (*right*).
- **Ángulo 7:** Posición central (neutral).

- Este control asegura que el servo ajuste su orientación de manera precisa según la entrada recibida en el flujo de Node-RED, permitiendo una dirección adecuada del movimiento del robot.

- **Conexión:**

El servo para la dirección está conectado a la Raspirobot de la siguiente manera:

- **Cable de señal (GPIO):** Conectado al pin GPIO 27, encargado de recibir señales PWM para regular la posición angular.
- **Cable de alimentación (Vcc):** Conectado al pin de 5V para la alimentación del servo.
- **Cable de tierra (GND):** Conectado al pin GND para cerrar el circuito eléctrico.

- **Esquema o Foto:**



- Flujos de Node-RED

Flujo 1: Control de Dirección del Servo

Descripción:

Este flujo controla la dirección de las ruedas delanteras del robot utilizando un joystick. Según el movimiento (izquierda, centro, derecha), el flujo ajusta la posición del servo.

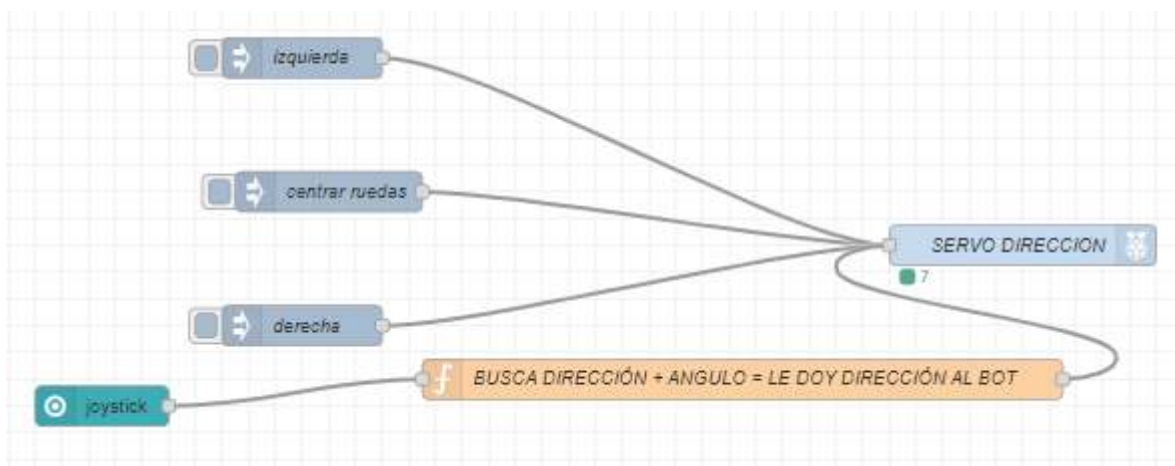
Inicio:

El flujo se activa mediante la posición del joystick conectado al nodo joystick.

Resultado final:

El servo se posiciona en una dirección específica (izquierda, centro, derecha) para controlar las ruedas del robot.

Captura del Flujo:



Descripción de Nodos:

- **Nodo 1: joystick**
 - **Función:** Recoge la dirección del joystick.
- **Nodo 2: izquierda**
 - **Función:** Detecta si el joystick se mueve hacia la izquierda.
- **Nodo 3: centrar ruedas**
 - **Función:** Detecta si el joystick se encuentra en la posición central.
- **Nodo 4: derecha**
 - **Función:** Detecta si el joystick se mueve hacia la derecha.
- **Nodo 5: SERVO DIRECCION**
 - **Función:** Controla el movimiento del servo para direccionar las ruedas.

Flujo 2: Control de Velocidad y Frenado con Sensor Ultrasonido

Descripción:

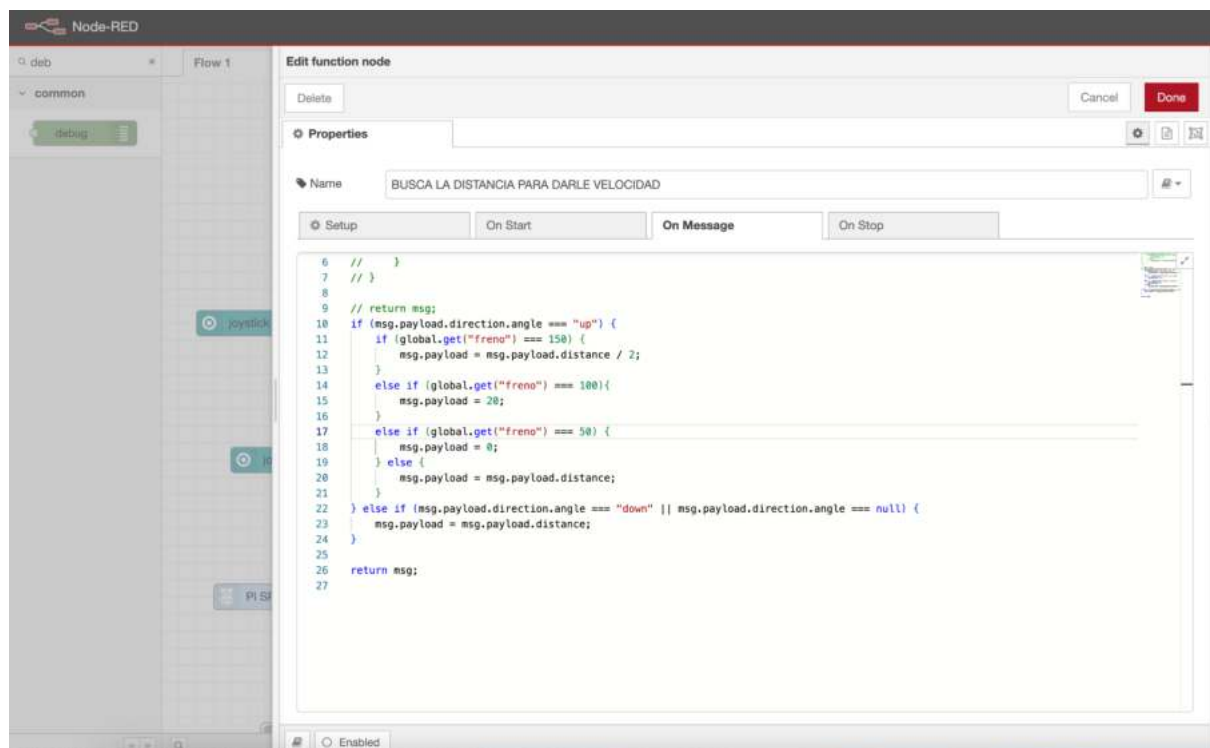
Este flujo utiliza un joystick para controlar la velocidad del robot (adelante o atrás). También incluye un sensor ultrasónico que detiene el robot si detecta un obstáculo.

Inicio:

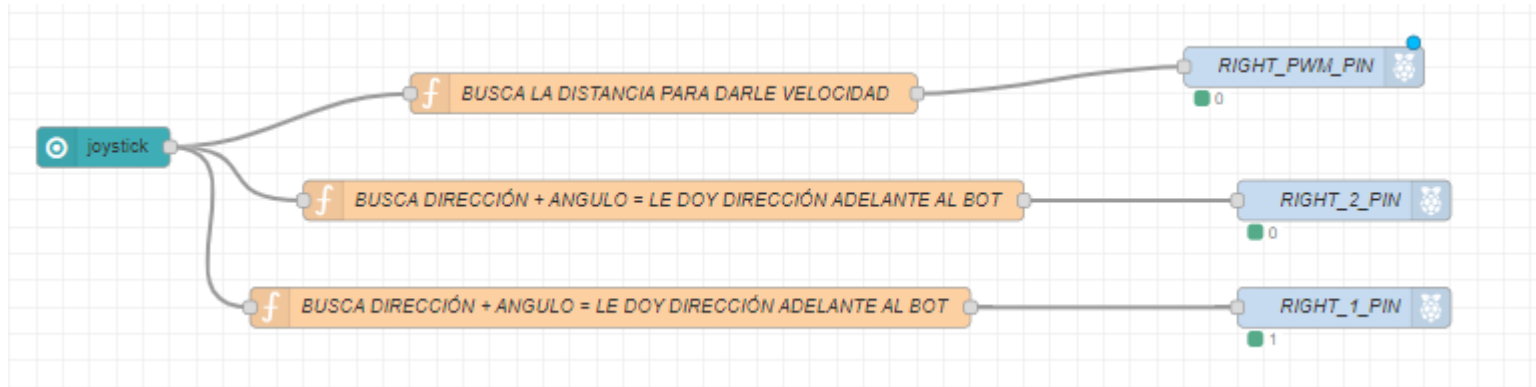
El flujo se activa mediante el movimiento del joystick y las lecturas del sensor ultrasónico.

Resultado final:

El robot avanza o retrocede según el joystick, o se detiene si el sensor ultrasónico detecta un obstáculo.



Captura del Flujo:



Descripción de Nodos:

- **Nodo 1: joystick**
 - **Función:** Recoge la dirección del joystick (arriba o abajo) para mover el robot.
- **Nodo 2: Pi SRF**
 - **Función:** Mide la distancia mediante un sensor ultrasónico para detectar obstáculos.
- **Nodo 3: BUSCA DIRECCIÓN + ÁNGULO (Función personalizada)**
 - **Función:** Determina si el robot avanza o retrocede basándose en la dirección del joystick y las lecturas del sensor ultrasónico.
- **Nodo 4: RIGHT_PWM_PIN**
 - **Función:** Controla la velocidad del motor mediante PWM.
- **Nodo 5: RIGHT_1_PIN y RIGHT_2_PIN**
 - **Función:** Controlan la dirección del motor para avanzar o retroceder.

Flujo 3: Gestión Global del Frenado

Descripción:

Este flujo utiliza el sensor ultrasónico para crear una variable global de frenado que se utiliza en otros flujos para detener el robot.

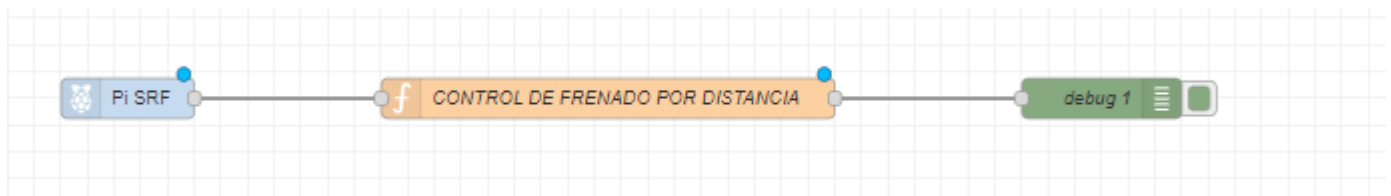
Inicio:

El flujo se activa por las lecturas del sensor ultrasónico.

Resultado final:

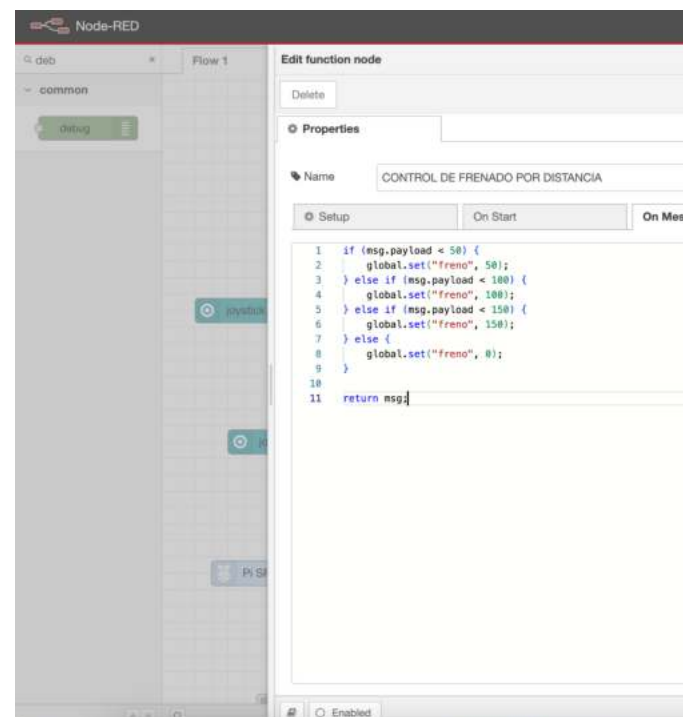
Establece una variable global con valor 1 si se detecta un obstáculo.

Captura del Flujo:



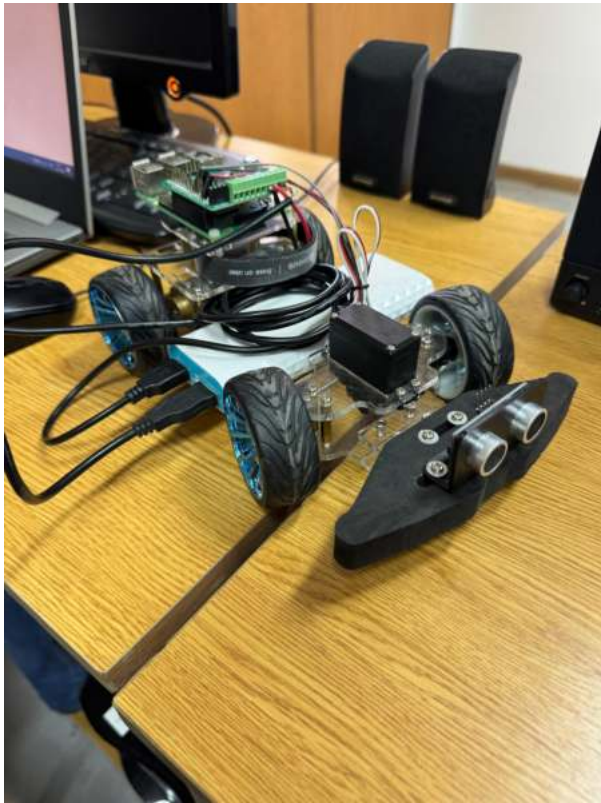
Descripción de Nodos:

- **Nodo 1: Pi SRF**
 - **Función:** Lee las mediciones del sensor ultrasónico.
- **Nodo 2: Control de Frenado por Distancia**
 - **Función:** Asigna la variable global freno para detener el robot cuando se detecta un obstáculo.

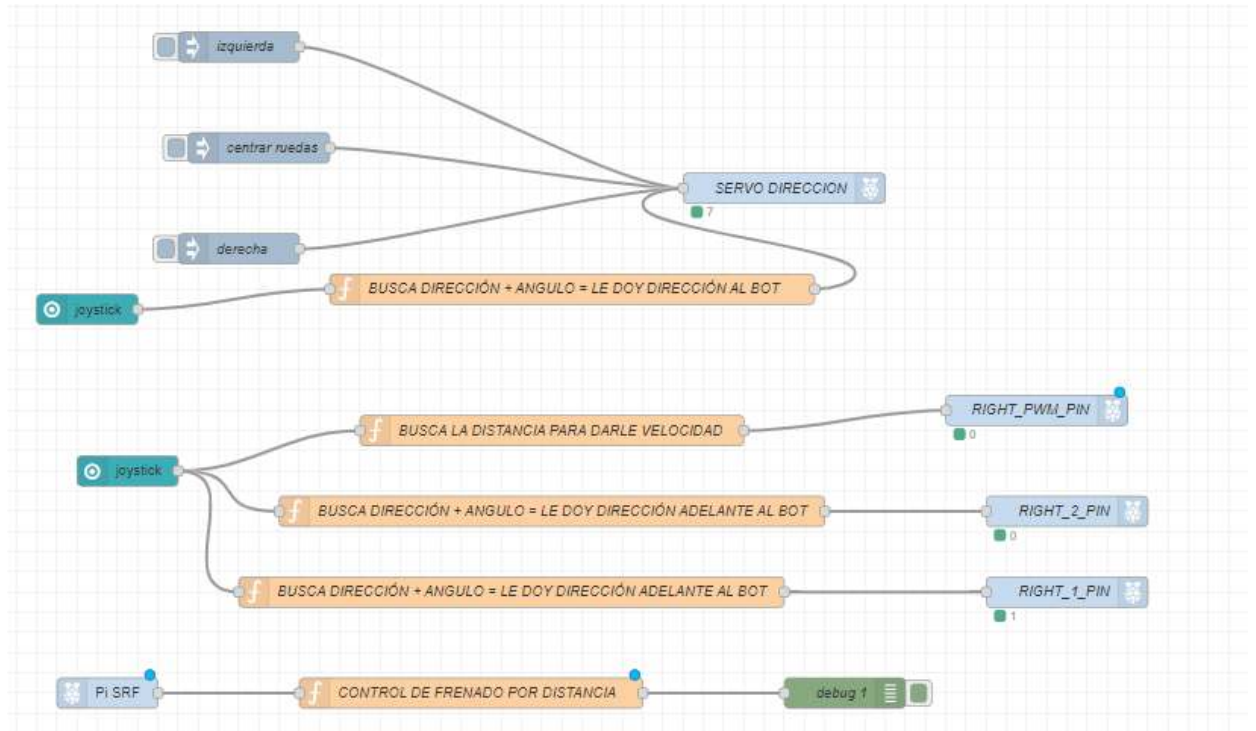


Documentación Visual

Fotos del Montaje



Capturas de Node-RED



Videos

Primera demostración sin frenado automático:

Demostración sin frenado automático: <https://youtube.com/shorts/tWk7w58SPcg>

Demostración con frenado automatico: <https://youtu.be/4cyRWU6X7ql>

Conclusión y Análisis

- **Conclusión**

Este proyecto permitió integrar sensores y actuadores utilizando RaspiRobot y Node-RED, logrando implementar funciones como el frenado automático y el control de dirección. Si bien surgieron algunos problemas, como la configuración de pines y la calibración del sensor, estos se resolvieron investigando y trabajando en equipo. En general, los resultados fueron positivos y dejaron lecciones importantes para proyectos futuros.

- **Dificultades encontradas:**

Configuración de hardware:

- Identificar correctamente los pines GPIO para cada componente, como teníamos raspirobot no supimos en un principio que pines debíamos usar ni como utilizar los motores.

Compatibilidad de software:

- Intentamos usar "rpi-gpio in" y "rpi-gpio out" para el sensor ultrasónico, pero resultó complicado. Optamos por "node-red-node-pisrf", que simplificó la configuración.

Gestión de variables globales:

- Confusiones al utilizar variables globales para compartir estados entre diferentes flujos, lo que generó comportamientos inesperados en el control del freno.

Problemas con la calibración:

- Dificultad para ajustar la sensibilidad del sensor ultrasónico y el umbral de distancia para activar el freno del vehículo, generando que el mismo choque.
- Configurar el joystick para que los ángulos y direcciones sean interpretados correctamente en Node-RED.

Depuración de errores:

- Identificar la causa de mensajes de error como "Invalid input: NaN" o "TypeError: Cannot read properties of null", lo que demandó tiempo para inspeccionar los datos de entrada y salida de los nodos.

Integración de múltiples dispositivos:

- Coordinar la interacción entre los datos del joystick y los del sensor ultrasónico dentro de un único flujo, asegurando que no interfieran entre sí.

- **Soluciones:**

La mayoría de los problemas los resolvimos usando ChatGPT y buscando en internet. También fue útil colaborar con compañeros que estaban atascados en cosas similares, ya que compartimos soluciones e ideas.

6. Referencias

- [Practical Node-RED Programming](#)
- [Adafruit](#)