INSTITUTO POLITECNICO NACIONAL

Escuela Superior de Ingeniería Mecánica y Eléctrica – Unidad Zacatenco

Carrera:

Ingeniera en Comunicaciones y Electrónica (ICE)

Materia:

Microprocesadores

Proyecto final

Carrito Rueda Loca con control remoto por WiFi

Profesor:

Galicia Galicia Roberto

Grupo: 6CM1 **Periodo:** 2023/2

Alumna:

Cruz Munguia Danna Jael

Fecha:

Julio 13, 2023

Objetivos.

Durante el desarrollo de este proyecto el alumno mostrara las habilidades desarrolladas a lo largo del curso implementándolas en algún proyecto en este caso un carrito de rueda loca, complementando códigos y añadiendo características que crean sean necesarias.

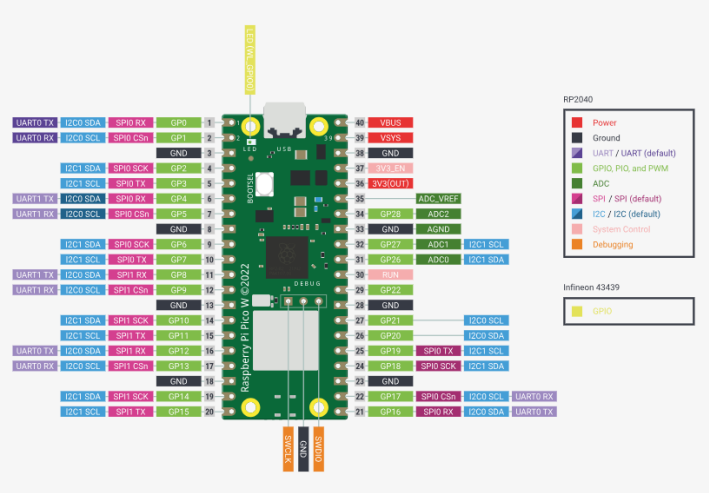
Introducción.

Raspberry pi pico w

Raspberry Pi Pico W agrega interfaces inalámbricas integradas de banda única de 2,4 GHz (802.11n) utilizando el Infineon CYW43439 mientras conserva el factor de forma Pico. La interfaz inalámbrica integrada de 2,4 GHz tiene las siguientes características:

- Inalámbrico (802.11n), banda única (2,4 GHz)
- WPA3
- Punto de acceso suave que admite hasta cuatro clientes
- Bluetooth 5.2
    - Compatibilidad con funciones de Bluetooth LE Central y Periférico
    - Compatibilidad con Bluetooth clásico

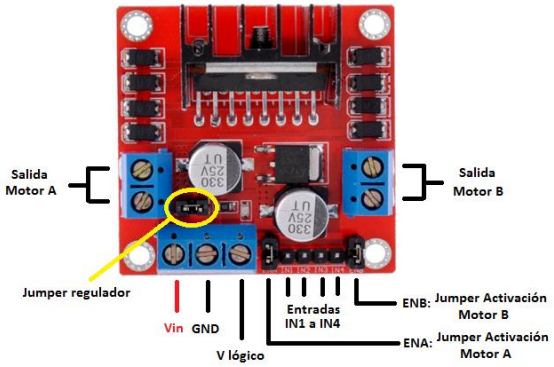
La antena es una antena integrada con licencia de ABRACON (anteriormente ProAnt). La interfaz inalámbrica está conectada a través de SPI al microcontrolador RP2040.



Debido a las limitaciones de los pines, algunos de los pines de la interfaz inalámbrica se comparten. El CLK se comparte con el monitor VSYS, por lo que solo cuando no hay una transacción SPI en curso se puede leer VSYS a través del ADC. El Infineon CYW43439 DIN/DOUT e IRQ comparten un pin en el RP2040. Solo cuando una transacción SPI no está en curso, es adecuado verificar las IRQ. La interfaz normalmente funciona a 33 MHz.

Modulo L298N

Este módulo posee dos puentes H que permiten controlar 2 motores DC o un motor paso a paso bipolar/unipolar. El módulo permite controlar el sentido de giro y velocidad mediante señales TTL que se pueden obtener de microcontroladores y tarjetas de desarrollo como Arduino, Raspberry Pi o Launchpads de Texas Instruments. Tiene integrado un regulador de voltaje de 5V encargado de alimentar la parte lógica del L298N, el uso de este regulador se hace a través de un Jumper y se puede usar para alimentar la etapa de control.



Características técnicas

- Chip: L298N
- Canales: 2 (soporta 2 motores DC 1 motor PAP)
- Voltaje lógico: 5V
- Voltaje de Operación: 5V-35V
- Consumo de corriente (Digital): 0 a 36mA
- Capacidad de corriente: 2A (picos de hasta 3A)
- Potencia máxima: 25W
- Voltaje de alimentación, mínimo de 5 V. Posee dos entradas, una de 5V para controlar la parte lógica y otra para alimentar las salidas al motor, que pueden ser de 5V o más.

Servidor web.

Un servidor web (server) es un ordenador de gran potencia que se encarga de "prestar el servicio" de transmitir la información pedida por sus clientes (otros ordenadores, dispositivos móviles, impresoras, personas, etc.)

Los servidores web (web server) son un componente de los servidores que tienen como principal función almacenar, en web hosting todos los archivos propios de una página web (imágenes, textos, videos, etc.) y transmitirlos a los usuarios a través de los navegadores mediante el protocolo HTTP (Hipertext Transfer Protocol).

Microphyton

Es un pequeño pero eficiente interprete del Lenguaje de Programación Python, optimizado para funcionar en microcontroladores y ambientes restringidos.

Un intérprete se define como la capa lógica de software entre el código y el hardware, dicho de otro modo, es el encargado de procesar el código de programación y hacer posible que el hardware (ordenador, microcontrolador…) ejecute las acciones en él descritas

Debido a las limitaciones de recursos de los microcontroladores, en su mayoría los módulos de la biblioteca estándar se han simplificado, pero proporcionando sus funcionalidades principales –un subconjunto de las funcionalidades totales-.

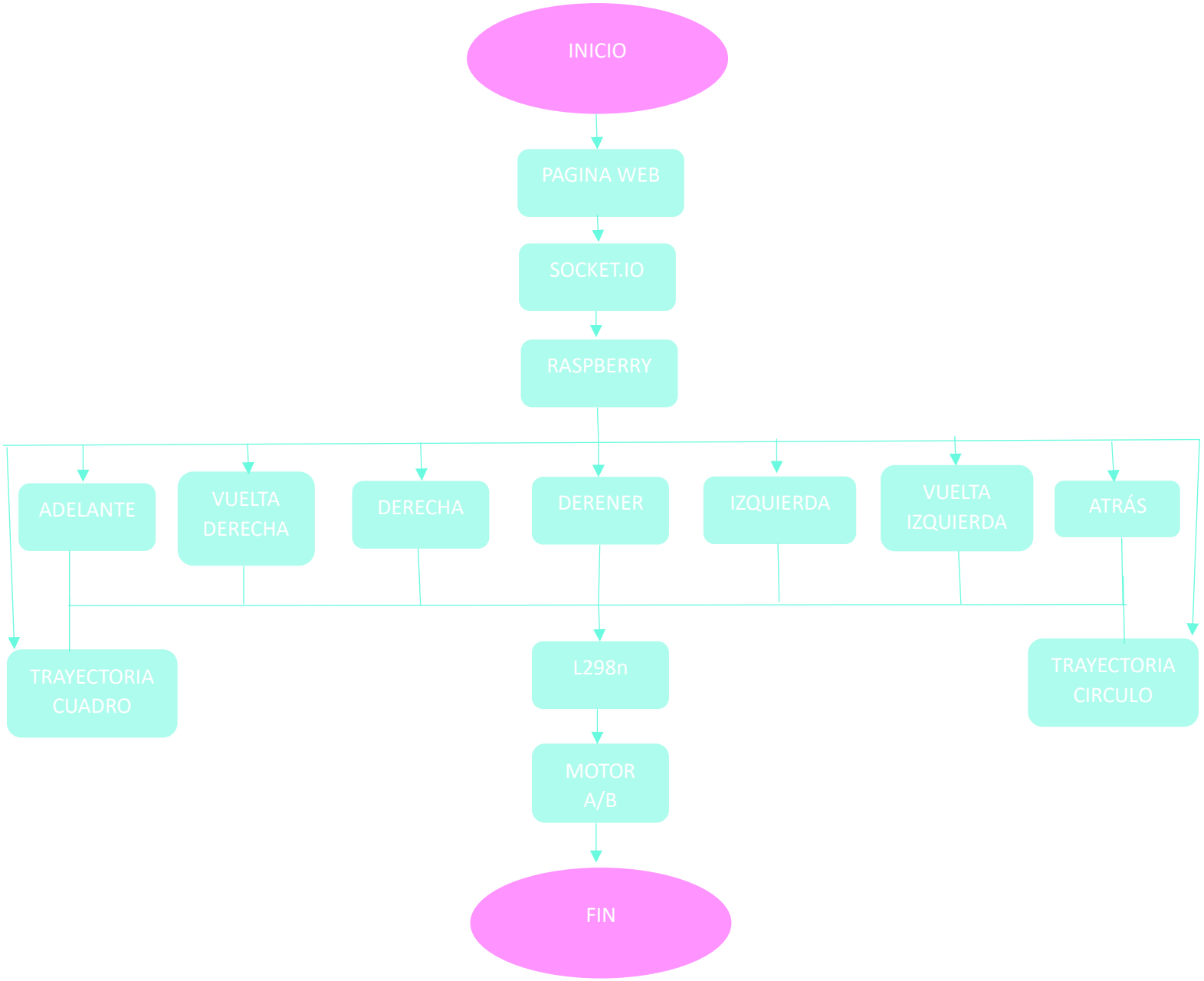
MicroPython también dispone de módulos específicos dentro de la biblioteca estándar que permiten al programador el acceso al hardware del microcontrolador.

Hay algunas características que MicroPython tiene y es lo que lo hace único y diferente de otros sistemas embebidos:

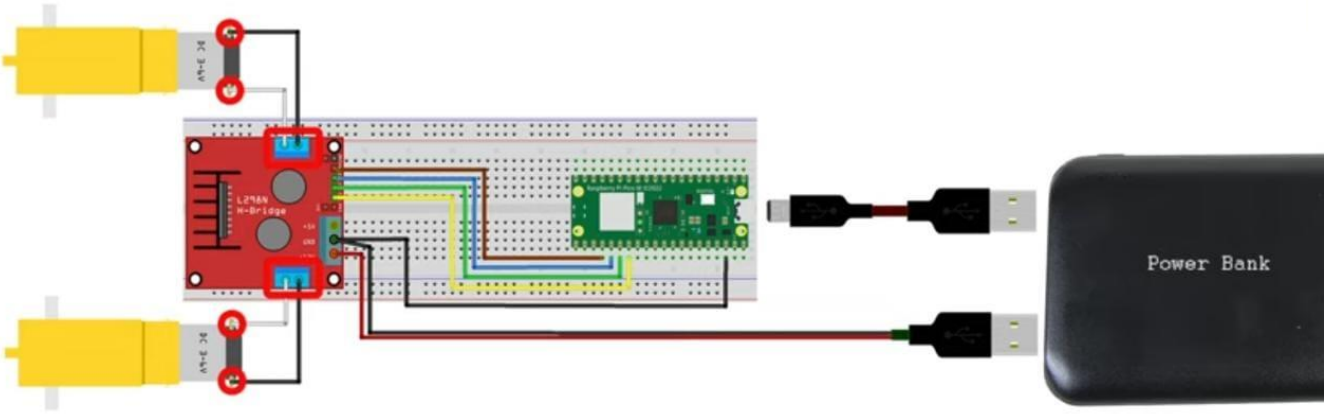
- Dispone, así como el Lenguaje de Programación Python, de multitud de librerías para la ejecución de tareas.
- Tiene un editor de código propio muy sencillo.
- Es extensible. Para los usuarios avanzados de MicroPython, pueden extender de Python a funciones de más bajo nivel como C o C++, pudiendo mezclar códigos que requieran de ejecución más rápida a bajo nivel con MicroPython.
- Con MicroPython, se pueden hacer muchas cosas, como controlar las entradas/salidas de un microcontroladores haciendo parpadear un LED, obteniendo lecturas de señales analógicas y digitales, generando señales PWM, controlando servomotores, pantallas OLED, pantallas NeoPixel, realizando comunicación I2C, SPI, etc. En algunos microcontroladores como el ESP32 también permite conexiones de Red y WiFi.

Desarrollo.

- **Diagrama de flujo**



- **Diagrama del circuito**



- **Codigo utilizado**

#Tiene correcciones en dereha e izquierda.

#librerias para menejo de redes

import time #controla el tiempo

import network #configura las redes

import socket #comunicacion de la red

from time import sleep

import machine #con lo que trabaja la raspberry pi pico w

from machine import Pin #controla los pines GPIO

RED WiFi

ssid = ‘nombre de la red’

password = 'contraseña de la red '

ssid = 'INFINITUMusrb'

password = 'ca34300df8'

```
# Asignación de Pins!
#objetos pin de la biblioteca machine
MotorA_Adelante = Pin(5, Pin.OUT) #asignan los dines GPIO a variables
MotorA_Atras = Pin(4, Pin.OUT)    #para controlar los motores y la entrada
MotorB_Adelante = Pin(3, Pin.OUT) #y saida de energiaa de los motores
MotorB_Atras = Pin(2, Pin.OUT)

enA = Pin(6, Pin.OUT)
enB = Pin(7, Pin.OUT)
```

#definicion de las funciones implementar acciones para controlar el carrito, como avanzar, retroceder, detenerse, girar a la izquierda, girar a la derecha y realizar trayectorias específicas.

```
def adelante(duration):
    print('adelante')
    start_time = time.time()
    while time.time() - start_time < duration:
        enA.value(1)
        enB.value(1)
        MotorA_Adelante.value(0)
        MotorB_Adelante.value(0)
        MotorA_Atras.value(1)
        MotorB_Atras.value(1)

def reversa():
    enA(1)
    enB(1)
```

```
    MotorA_Adelante.value(1)
    MotorB_Adelante.value(1)
    MotorA_Atras.value(0)
    MotorB_Atras.value(0)
    print('reversa')

def stop():
    enA(1)
    enB(1)

    MotorA_Adelante.value(0)
    MotorB_Adelante.value(0)
    MotorA_Atras.value(0)
    MotorB_Atras.value(0)
    print('parar')

def izquierda():  # rueda izquierda detenida
    enA(0)
    enB(1)

    MotorA_Adelante.value(0)
    MotorB_Adelante.value(1)
    MotorA_Atras.value(0)
    MotorB_Atras.value(0)
    print('izquierda')
```

```
def derecha(duration):  # rueda derecha detenida
    start_time = time.time()
    while time.time() - start_time < duration:
        enA(1)
        enB(0)
        MotorA_Adelante.value(1)
        MotorB_Adelante.value(0)
        MotorA_Atras.value(0)
        MotorB_Atras.value(0)
    print('derecha')

def giroderecha():
    print('giraderecha')
    # una rueda adelante, la otra atras
    enA(1)
    enB(1)
    MotorA_Adelante.value(1)
    MotorB_Adelante.value(0)
    MotorA_Atras.value(0)
    MotorB_Atras.value(1)

def giroizquierda():  # una rueda adelante, la otra atras
    enA(1)
    enB(1)
    MotorA_Adelante.value(0)
    MotorB_Adelante.value(1)
    MotorA_Atras.value(1)
```

```
    MotorB_Atras.value(0)
    print('gira izquierda')

# Detener el carrito
stop()

def trayectoriaCuadro():
    print('vector')
    adelante(0.2)
    stop()
    derecha(0.001)
    stop()
    adelante(0.2)
    stop()
    derecha(0.001)
    stop()
    adelante(0.2)
    stop()
    derecha(0.001)
    stop()
    adelante(0.2)
    stop()
    derecha(0.001)
    stop()
    stop()
```

```
def trayectoriaCircular():
    print('En circulos')
    for _ in range(6):
        adelante(0.001)
        stop()
        derecha(0.00001)
        stop()
        adelante(0.001)
        stop()
        derecha(0.00001)
        stop()

def connect():
    # Connect to WLAN
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    wlan.connect(ssid, password)
    while wlan.isconnected() == False:
        print('Esperando Conexión...')
        sleep(1)
    ip = wlan.ifconfig()[0]
    print(f'Conectado a {ip}')
    return ip

def open_socket(ip):
    # Open socket
    address = (ip, 80)
```

```
    connection = socket.socket()
    connection.bind(address)
    connection.listen(1)
    return connection

# genera una página HTML que se enviará como respuesta cuando se realicen solicitudes a través del socket
def webpage():
    html = f"""
    <!DOCTYPE html>
<html lang="es">
<head>

    <title>CARRITO RUEDA LOCA</title>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="AUTOR:" content="DANNA CRUZ">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>
<header id="headr" style="width: 100%; height: 40px; position: static; background: pink; font-family:'century gothic'; padding: 10px 10px 50px 0; ">
        <section id="TOP">
            <div id="titulo">
                <ul>
                    <li>
```

```
                <h1 style="text-align: center; color: #FF00DD; font-size:1.5em">CONTROL RUEDA LOCA</h1>
                </li>
            </ul>
        </div>
    </section>
</header>
<body style="margin: 0; padding: 0; background-color: pink;">
    <script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/4.5.4/socket.io.js"></script> <!-- include socket.io client side script -->
    <script> var socket = io(); //load socket.io-client and connect to the host that serves the page </script>
    <!--Inicio Controles-->
    <section id="controles" style="width: 100%; height: 400px;
        background: url(https://w0.peakpx.com/wallpaper/30/833/HD-wallpaper-artistic-mountain-pink-minimalist.jpg) 50% 0 ; background-size: cover; background-position:center;">
    <div id="control" style="position: relative; top: 25px; width: 60%; height: 90%; margin: 0 auto; background-color: rgba( 245, 176, 208, 0.5); border-radius: 50px;">
        </td>
        <center>
            <form action="./adelante">
                <input type="submit" value="FORWARD "
                style="background-color: #FF0077;
                border-radius: 05px; width:80px; height:45px;"/>
            </form>
            <table >
                <tr>
```

```
<td>
   <form action="./giroizquierda">
   <input type="submit" value="TURN LEFT"
   style="background-color: #FF0077;
   border-radius: 05px; width:95px; height:45px;"/>
   </form>
</td>
<td>
   <form action="./izquierda">
   <input type="submit" value="LEFT"
   style="background-color: #FF0077;
   border-radius: 05px; width:65px; height:45px;"/>
   </form>

</td>
<td>
   <form action="./stop">
   <input type="submit" value="STOP"
   style="background-color: #FF0077;
   border-radius: 05px; width:65px; height:45px;"/>
   </form>

</td>
<td>
   <form action="./derecha">
   <input type="submit" value="RIGTH"
   style="background-color: #FF0077;
```

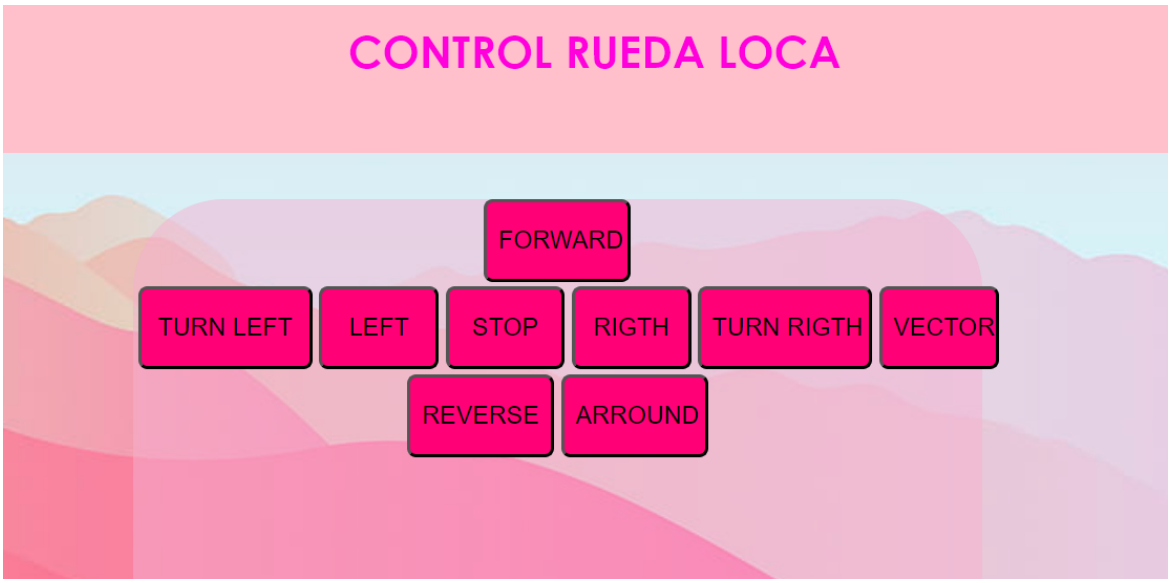
```
            border-radius: 05px; width:65px; height:45px;"/>
            </form>
        </td>
        <td>
            <form action="./giroderecha">
            <input type="submit" value="TURN RIGTH"
            style="background-color: #FF0077;
            border-radius: 05px; width:95px; height:45px;"/>
            </form>
        </td>
        <td>
            <form action="./trayectoriaCuadro">
            <input type="submit" value="VECTOR"
            style="background-color: #FF0077;
            border-radius: 05px; width:65px; height:45px;"/>
            </form>
        </td>
    </tr>
</table>
<form action="./reversa">
            <input type="submit" value="REVERSE"
            style="background-color: #FF0077;
            border-radius: 05px; width:80px; height:45px;"/>
<form action="./trayectoriaCircular">
            <input type="submit" value="ARROUND"
            style="background-color: #FF0077;
            border-radius: 05px; width:80px; height:45px;"/>
```

```
            </table>


        </center>
    </div>
    </section><!--controles-->
</body>
</html>
            """
    return str(html)


def serve(connection):
    # Inicio del servidor
    #Cuando se recibe una solicitud, se procesa y se realizan acciones específicas según la solicitud. Además, se envía la página HTML generada por la función webpage como respuesta.
    while True:
        client = connection.accept()[0]
        request = client.recv(1024)
        request = str(request)
        try:
            request = request.split()[1]
        except IndexError:
            pass
        if request == '/adelante?':
            adelante(2)
        elif request == '/izquierda?':
            izquierda()
        elif request == '/stop?':
```

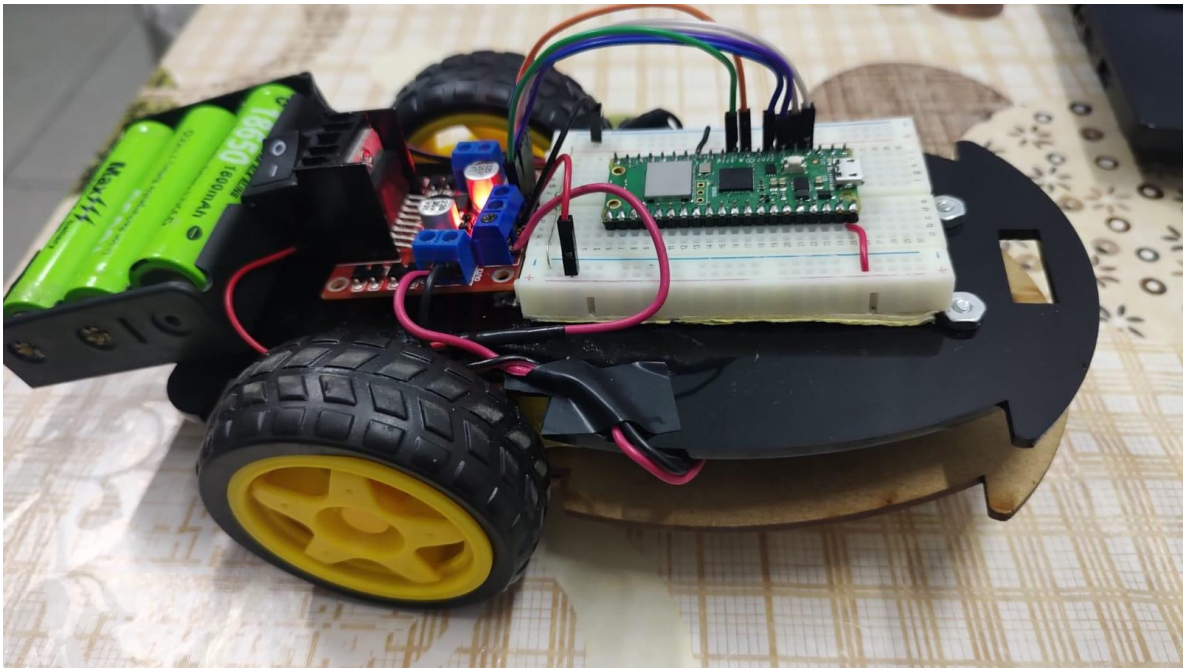
```
            stop()
        elif request == '/derecha?':
            derecha()
        elif request == '/reversa?':
            reversa()
        elif request == '/giroderecha?':
            giroderecha(2)
        elif request == '/giroizquierda?':
            giroizquierda()
        elif request == '/trayectoriaCuadro?':
            trayectoriaCuadro()
        elif request == '/trayectoriaCircular?':
            trayectoriaCircular()

        html = webpage()
        client.send(html)
        client.close()

try:
    ip = connect()    #establece una conexión a la red WiFi utilizando el SSID y la contraseña proporcionados.
    connection = open_socket(ip) #socket en una dirección IP y un puerto específicos para recibir y enviar datos a través de una conexión de red.
    serve(connection) #inicia el servidor web y maneja las solicitudes entrantes.

except KeyboardInterrupt:
    machine.reset()
```

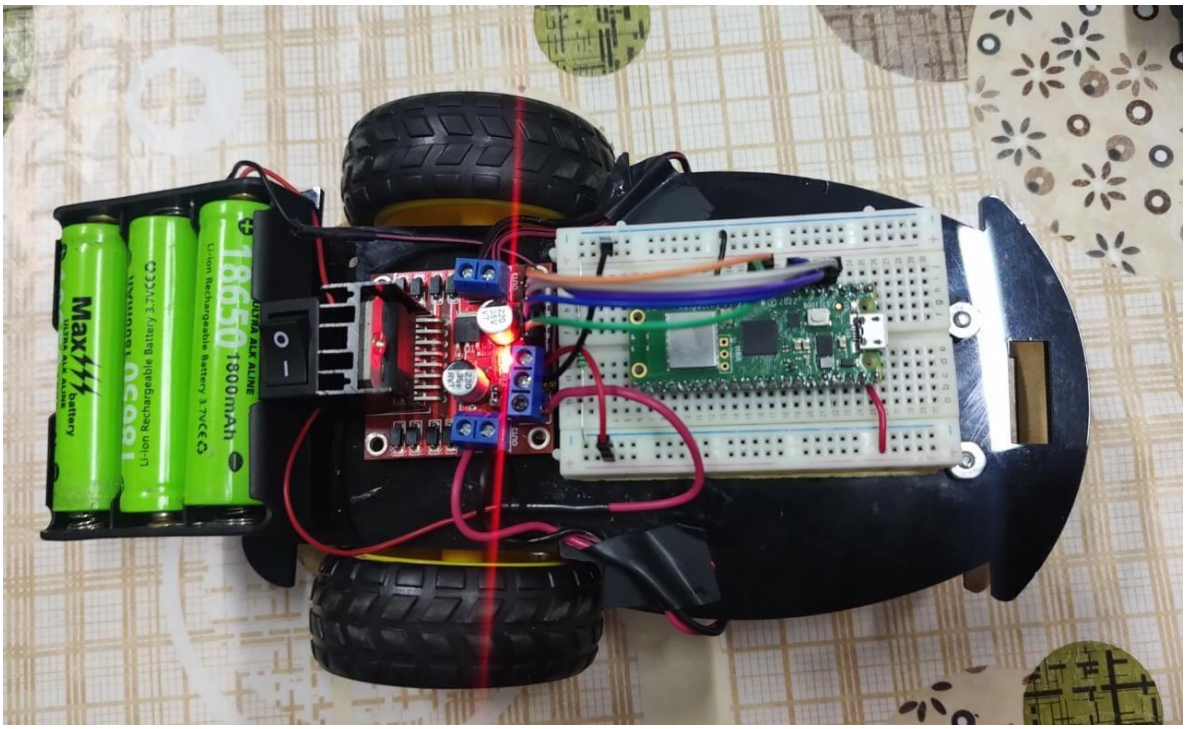
Conclusiones.

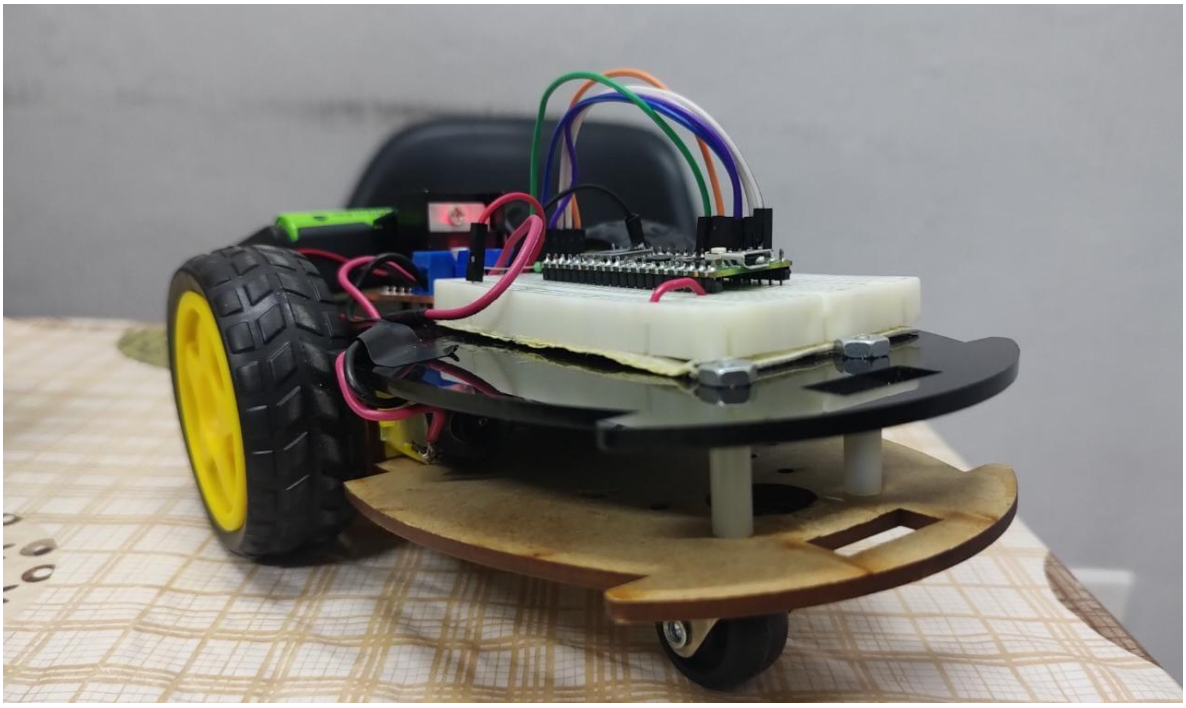
- **Vista servidor**

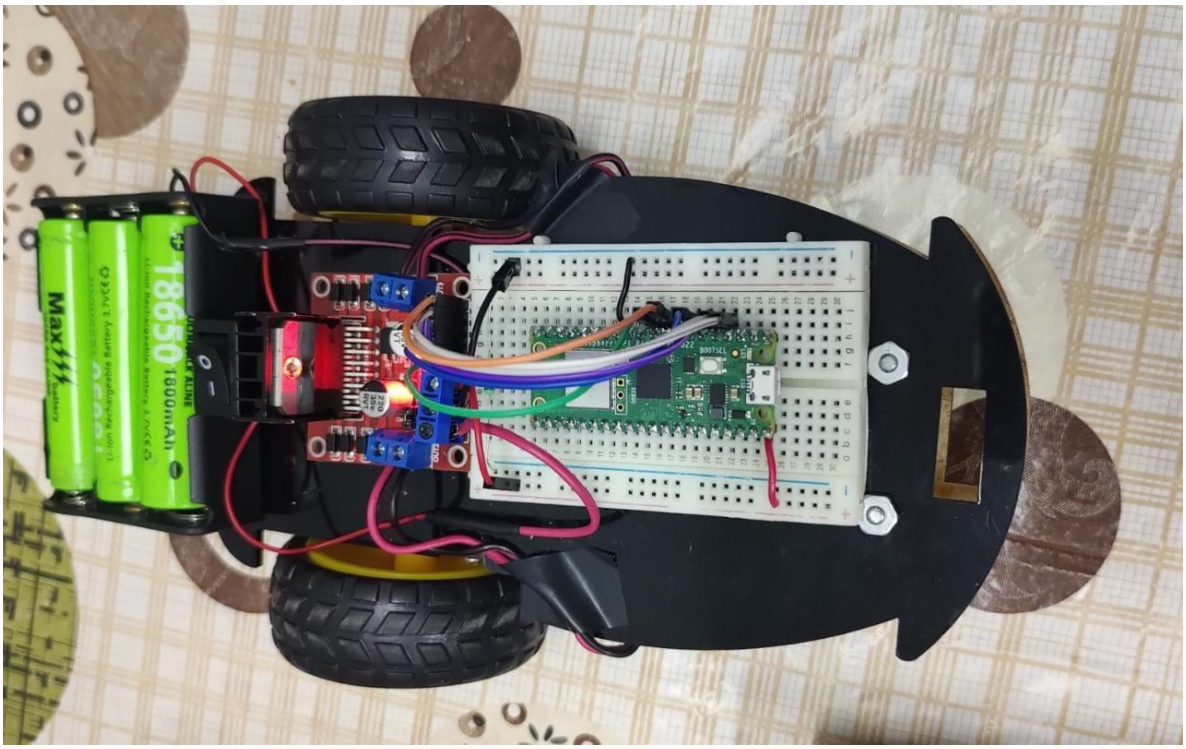


- **Carrito armado**




Max
battery
ULTRA ALK ALINE
18650 1800mAh
Li-ion Rechargeable Battery 3.7V





Referencias

- https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html
- https://www.xatakahome.com/domotica/raspberry-pi-pico-w-caracteristicas-precio-ficha-tecnica
- https://www.esploradores.com/python_y_micropython_que_son/
- https://github.com/ComputadorasySensores/Capitulo75/blob/main/picow-robot.py
- https://github.com/rpi-jefer/rpi-ultrasonido
- https://github.com/DMelesio/Carrito_Control_Web.git