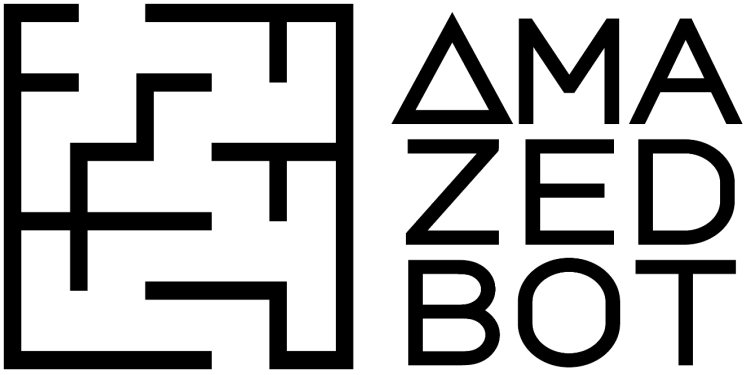
Documentación Técnica



Integrantes: Sofía da Silva
Gian Luca Porto
Mateo Moreira
Ignacio Carvallo

1. Introducción

En la actualidad, el enfoque STEAM+ (Ciencia, Tecnología, Ingeniería, Arte y Matemáticas, con la integración de otras áreas del conocimiento) es fundamental para el desarrollo de competencias clave que preparan a los estudiantes para resolver problemas del mundo real mediante la aplicación de habilidades interdisciplinares. En este proyecto, hemos diseñado e implementado una solución tecnológica que aprovecha este enfoque para enfrentar el desafío de construir un robot autónomo capaz de navegar por un laberinto. La estrategia de navegación se basa en la detección de una línea negra, que actúa como obstáculo. Utilizando tres sensores TCRT5000, el robot puede identificar la presencia o ausencia de esta línea y, en función de ello, decide la dirección a seguir, permitiéndole así evitar la línea y avanzar de manera autónoma.

Objetivo del proyecto:

Desarrollar un robot autónomo que, utilizando micro:bit y sensores, sea capaz de navegar y encontrar la salida de un laberinto. Este proyecto tiene como objetivo combinar habilidades de programación, electrónica y robótica para construir un robot que pueda enfrentar el desafío de los laberintos de manera efectiva.

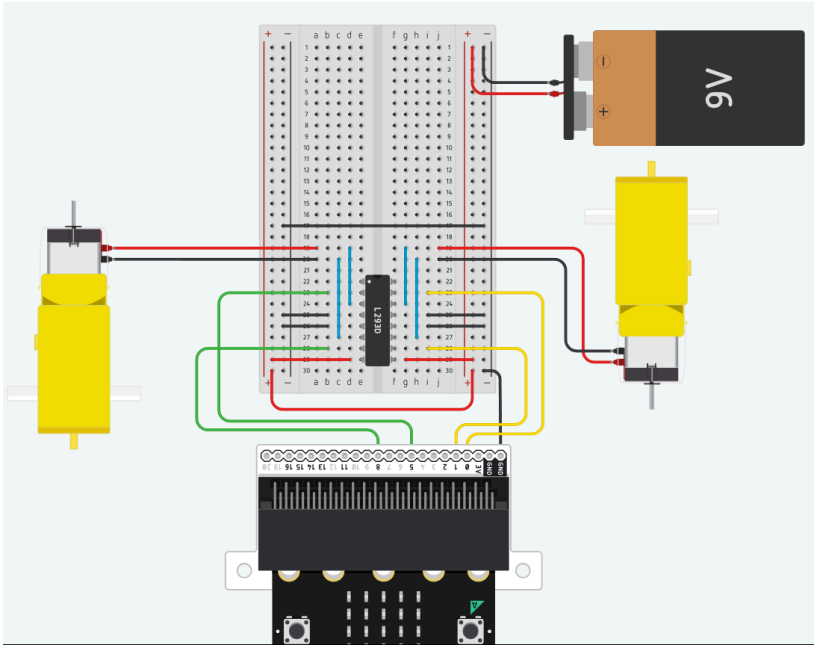
Resumen del sistema:

El sistema consiste en un robot autónomo diseñado para navegar a través de laberintos utilizando micro:bit y sensores. Equipado con sensores de seguidor de línea y motores, el robot detecta obstáculos y toma decisiones en tiempo real para encontrar la salida. La programación implementada en la micro:bit permite al robot procesar la información de los sensores, ejecutar algoritmos de navegación y adaptarse a las condiciones cambiantes del laberinto. Este enfoque integrador combina conceptos de programación, electrónica y robótica, fomentando el desarrollo de habilidades técnicas y de resolución de problemas.

2. Diseño del sistema

Esquema de conexiones:

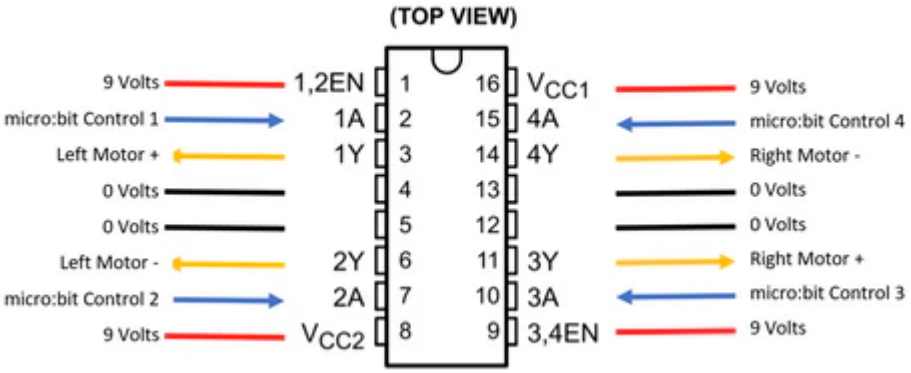
A continuación mostraremos el circuito utilizado para usar los motores con el microchip:



Explicación del circuito

1. **Batería de 9V**: La batería de 9V está conectada a la protoboard y se usa como fuente de alimentación para los motores. El polo positivo (+) se conecta a la línea de alimentación positiva de la protoboard, mientras que el negativo (-) va a la línea de tierra.

2. **Microchip**:
    - El microchip tiene la función de recibir y distribuir la energía de la batería a los motores DC, dependiendo de las señales enviadas por la micro:bit. El mismo cuenta con 16 pines, siendo 8 por cada lado.
    - **Pines**
        - Como se mencionó anteriormente, el microchip cuenta con ocho pines en cada lado. Los ocho pines del lado izquierdo desempeñan funciones idénticas, pero en configuración espejada, a las de los ocho pines del lado derecho.
        - A continuación, utilizando la imagen inferior como referencia, detallaremos las características y funciones del lado izquierdo del microchip.



- Guiándonos por los colores, podemos categorizar de una manera más simple el funcionamiento de cada uno de los pines.
- **Cables rojos**
    1. Los cables rojos representan la fuente de alimentación del chip. Es necesario suministrar 9 V a cada uno de los cuatro pines correspondientes, los cuales se encuentran en las extremidades del microchip, específicamente en los pines 1, 8, 9 y 16.
- **Cables negros**
    1. Los cables negros corresponden a la tierra, los mismos van en la parte central del microchip, en los pines 4, 5, 12 y 13.
- **Cables azules**
    1. Los cables azules se conectan a los pines de la micro y son responsables de transmitir la señal al microchip, permitiendo que la corriente fluya hacia los motores de corriente continua en una dirección específica. Para el movimiento de la rueda izquierda, se envía una señal positiva a través del pin 2 y una señal negativa a través del pin 7. En el caso de la rueda derecha, se transmite una señal negativa por medio del pin 10 y una señal positiva a través del pin 15.
- **Cables amarillos**
    1. Estos cables son los encargados de pasar la corriente que envía el microchip a los motores DC. Para la rueda izquierda se usan los pines 3 y 6, mientras que para la rueda derecha, los pines 11 y 14.

3. **Motores de corriente directa**:
    - Hay dos motores DC en este circuito: uno a la izquierda y otro a la derecha.

- Cada motor está conectado a dos salidas del microchip, lo cual permite invertir la polaridad para controlar la dirección del giro de los motores.
- La potencia para los motores es provista por la batería de 9V, lo que permite que cada motor funcione de manera independiente.

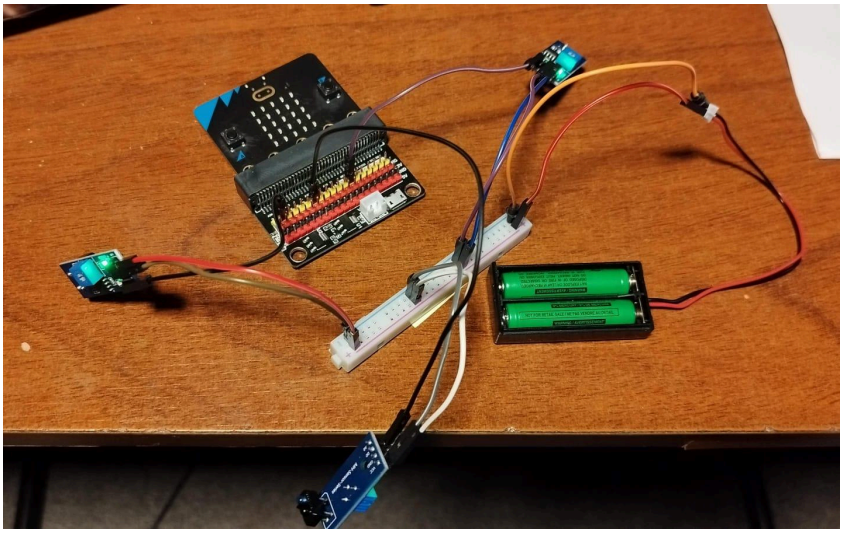
4. **Microcontrolador (micro:bit)**:
   - La microbit actúa como la unidad de control para manejar las señales de entrada del microchip.
   - Los pines de salida de la microbit están conectados a los pines de entrada del microchip para controlar la dirección y la activación de los motores.
   - Este dispositivo envía señales de 0V o 5V al controlador de motores para hacer que giren en una dirección u otra o se detengan.

Funcionamiento

- Al enviar una señal de voltaje desde la microbit a los pines de entrada del microchip, el controlador dirige la corriente de la batería de 9V hacia el motor correspondiente, haciendo que gire en una dirección específica.
- La dirección del giro de cada motor depende de la combinación de las señales enviadas a los pines de entrada del microchip.

Este circuito permite controlar la dirección de los motores de manera independiente, por lo que podría usarse para mover un robot hacia adelante, atrás o girar hacia ambos lados.

Ahora mostramos el circuito para usar los sensores de seguimiento de línea:



Explicación del circuito:

1. **Pilas AA:** Las dos pilas AA se conectan a la protoboard para servir de fuente de alimentación para los 3 sensores. El polo positivo (+) se conecta a la línea de alimentación positiva de la protoboard, mientras que el negativo (-) va a la línea de tierra. Cabe destacar que un cable de la tierra debe ir conectado a la micro:bit, para así cerrar el circuito.

2. **Sensores de seguidor de línea:**
    - Los sensores constan de 4 pines diferentes. 2 de ellos se encargan de la alimentación del sensor, el VCC (corriente positiva) y el GND (tierra). Los otros dos son el pin análogo y el pin digital.
    - **Pin análogo**
        - El pin análogo envía una señal entre 0 y 1023, dependiendo de qué tan clara o que tan oscura sea la señal recibida por el sensor, cuanto más oscura la señal, más alto el valor.
    - **Pin digital**
        - El pin digital envía una señal entre 0 o 1, dependiendo de si la señal recibida es más blanca o más negra. Si es oscura devolverá 1, caso contrario 0.

Descripción de componentes

1. Micro:bit: Especificaciones y funciones.
2. Sensores: Modelo, características y cómo se utilizan en el proyecto.
3. Motores y ruedas: Especificaciones y cómo se integran en el robot.
4. Batería: Tipo, capacidad.

1. Micro:bit

La microbit es un microcontrolador compacto y versátil, diseñado por la BBC para facilitar el aprendizaje de la programación y la creación de proyectos interactivos. A continuación se presentan sus principales especificaciones y funciones:

- **Procesador:** Cuenta con un microprocesador ARM Cortex-M0 de 32 bits, que permite un procesamiento eficiente de datos.
- **Memoria:** Dispone de 16 KB de RAM y 256 KB de almacenamiento flash, lo que proporciona capacidad suficiente para ejecutar programas y almacenar datos.
- **Conectividad:** Incorpora conectividad Bluetooth para la comunicación inalámbrica, así como un puerto USB para la programación y la carga de energía.

- **Entradas y salidas:** Posee 1 conector USB, lego touch, 25 luces led, 2 botones, indicador de micrófono, 5 pines, antena de radio, micrófono, botón de reinicio, socket de batería, procesador, parlante, conector edge para accesorios, brújula y acelerómetro.

Uso en el proyecto:
En el contexto del proyecto, la microbit se utiliza para procesar la información de los sensores, controlar los motores y ejecutar el algoritmo de navegación del robot, facilitando así su capacidad para encontrar la salida del laberinto.

2. Sensores

Modelo: TCRT5000
Características:

- **Tipo:** Sensor de seguimiento de línea.
- **Rango de medición:** 1mm a 8mm.
- **Precisión**: Hasta 2.5mm.
- **Tensión de operación:** 3.3V - 5V.
- **Salida:** Digital o analógica.

Uso en el proyecto:
Los sensores TCRT5000 se utilizan para encontrar paredes en el laberinto impreso. Al ser las paredes de color negro, el sensor capta el cambio de color y envía una señal al micro:bit, el cual se encarga de redireccionar el robot en función de las opciones disponibles para continuar, teniendo en cuenta la información brindada por el sensor.

3. Motores y ruedas

Modelo: Motor DC 3-6V con rueda
Características:

- **Tensión de operación**: 3-6 V.
- **Tipo de motor**: Motor de corriente directa (DC).
- **Dimensiones**: Compacto, lo que facilita su integración en robots.
- **Ruedas**: Incluye ruedas que permiten un movimiento efectivo y preciso.
- **Eje de salida:** Proporciona una forma de conexión para las ruedas.
- **Uso:** Ideal para proyectos de robótica y automatización.

Uso en el proyecto:
En el proyecto, este motor se utiliza para propulsar el robot y permitirle navegar por el laberinto. Su capacidad para funcionar a diferentes voltajes nos permite no tener que preocuparnos tanto por el voltaje de las baterías, para poder ahorrar costos.

Cómo se integran:
Se utiliza un microchip para controlar cada rueda del robot. La función del microchip es regular la corriente que se asigna al motor, gestionada desde la micro:bit.

4. Baterías

Baterías AAA (6 para las dos ruedas)

- **Tipo**: Batería alcalina.
- **Voltaje**: 1.5V por batería (9V en total para 6 baterías).
- **Uso:** Proporcionan energía a los motores de las ruedas, permitiendo el movimiento del robot.
- **Características:** Compactas y ligeras, ideales para aplicaciones portátiles.

Baterías AAA (2 para los 3 sensores)

- **Tipo:** Batería alcalina
- **Voltaje:** 1.5V por batería (3V en total para 2).
- **Uso:** Alimentan los sensores utilizados en el robot, que sería el caso de los de seguimiento de línea (TCRT5000).
- **Características:** Proporcionan un suministro constante de energía necesario para el funcionamiento de los sensores.

Baterías AAA (2 para la micro:bit)

- **Tipo:** Batería alcalina
- **Voltaje:** 1.5V por batería (3V en total para 2 baterías).
- **Uso:** Suministran energía a la micro:bit, permitiendo su funcionamiento y procesamiento de datos.
- **Características:** Compactas y ligeras, por lo que no suponen gran peso a la hora de usar el robot.

3. Programación

Código fuente:

- **Prueba con los sensores de seguimiento de línea**

```
def on_forever():
    serial.write_line("" +
    str(pins.analog_read_pin(AnalogReadWritePin.P0)) +
```

```
        "," +
        str(pins.analog_read_pin(AnalogReadWritePin.P1)) +
        "," +
        str(pins.analog_read_pin(AnalogReadWritePin.P2)))
        if pins.analog_read_pin(AnalogReadWritePin.P0) <= 511:
            led.plot(0, 0)
            led.unplot(0, 4)
        else:
            led.plot(0, 4)
            led.unplot(0, 0)
        if pins.analog_read_pin(AnalogReadWritePin.P1) <= 511:
            led.plot(2, 0)
            led.unplot(2, 4)
        else:
            led.plot(2, 4)
            led.unplot(2, 0)
        if pins.analog_read_pin(AnalogReadWritePin.P2) <= 511:
            led.plot(4, 0)
            led.unplot(4, 4)
        else:
            led.plot(4, 4)
            led.unplot(4, 0)
    basic.forever(on_forever)
```

- **Prueba con los motores DC**

```
    led.enable(False)

    def on_button_pressed_a():
        pins.digital_write_pin(DigitalPin.P3, 1)
        pins.digital_write_pin(DigitalPin.P6, 1)
        basic.pause(1000)
        pins.digital_write_pin(DigitalPin.P3, 0)
        pins.digital_write_pin(DigitalPin.P6, 0)
    input.on_button_pressed(Button.A, on_button_pressed_a)

    def on_button_pressed_b():
        pins.digital_write_pin(DigitalPin.P4, 1)
        pins.digital_write_pin(DigitalPin.P7, 1)
        basic.pause(1000)
```

```
        pins.digital_write_pin(DigitalPin.P4, 0)
        pins.digital_write_pin(DigitalPin.P7, 0)
    input.on_button_pressed(Button.B, on_button_pressed_b)
```

- **Prototipo código final**

```
# Ruedas
RUEDA_DER_F = DigitalPin.P3
RUEDA_DER_B = DigitalPin.P4
RUEDA_IZQ_F = DigitalPin.P6
RUEDA_IZQ_B = DigitalPin.P7

# Sensores
SENSOR_IZQ = DigitalPin.P0
SENSOR_CEN = DigitalPin.P1
SENSOR_DER = DigitalPin.P2
SENSOR_ANALOGO_VAL = 511 # Se leen de forma análoga, este
número define la diferencia entre 0 y 1

# Otras constantes
GIRO_MCR = 0 # Por cuantos microsegundos debe girar el robot
para darse vuelta 90°

# Globales
x = 0
y = 0
d = 0
nodos = []
infoSensores = []
enejec = False

# Pantalla de inicio
input.on_button_pressed(Button.A, iniciar)
led.enable(False)

### FUNCIONES DE LÓGICA PRINCIPALES ###

# Funcion llamada al apretar "A"
def iniciar():
    global enejec
```

```
    if enejec:
        return # Solo ejecuta esta función si el programa no
esta en ejecución
    enejec = True
    music.set_built_in_speaker_enabled(True)
    basic.pause(1000)
    calibrar()
    pass

# Calibración
def calibrar():
    global infoSensores
    global GIRO_MCR

    # Calibrar giro
    dir_i = input.compass_heading()
    girar("d",1000000)
    dir_f = input.compass_heading()
    if dir_f < dir_i:
        dir_f += 360
    dir_total = dir_f - dir_i
    serial.write_line(str(dir_i) + str(",") + str(dir_f) +
str(",") + str(dir_total))
    if dir_total == 0:
        music.play(music.tone_playable(Note.A,
music.beat(BeatFraction.WHOLE)),
music.PlaybackMode.UNTIL_DONE)
        control.reset()
    GIRO_MCR = int((90/dir_total)*200000)
    girar("i",200000)
    basic.pause(100)

    # Asegurarse que esté en el laberinto
    if verCaminos()[1] == 1:
        music.play(music.tone_playable(Note.B,
music.beat(BeatFraction.WHOLE)),
music.PlaybackMode.UNTIL_DONE)
        control.reset()

    bucle()
```

```
# Bucle Principal
def bucle():
    global x
    global y
    global d
    global nodos
    global infoSensores
    global GIRO_MCR
    escapado = False

    while(not escapado):
        pass

    return


### FUNCIONES DE SENSORES ###
def verCaminos():
    global SENSOR_ANALOGO_VAL
    global SENSOR_IZQ
    global SENSOR_CEN
    global SENSOR_DER

    v1 = 0
    v2 = 0
    v3 = 0
    if pins.analog_read_pin(SENSOR_IZQ) > SENSOR_ANALOGO_VAL:
        v1=1
    if pins.analog_read_pin(SENSOR_CEN) > SENSOR_ANALOGO_VAL:
        v2=1
    if pins.analog_read_pin(SENSOR_DER) > SENSOR_ANALOGO_VAL:
        v3=1

    return [
        v1,
        v2,
        v3
    ]
```

```
### FUNCIONES DE MOTORES ###
def girar(direccion,tiempo):
    global RUEDA_IZQ_F
    global RUEDA_IZQ_B
    global RUEDA_DER_F
    global RUEDA_DER_B

    if direccion == "d":
        serial.write_line("girando?")
        pins.digital_write_pin(RUEDA_IZQ_F, 1)
        pins.digital_write_pin(RUEDA_DER_B, 1)
        control.wait_micros(tiempo)
        pins.digital_write_pin(RUEDA_IZQ_F, 0)
        pins.digital_write_pin(RUEDA_DER_B, 0)
    elif direccion == "i":
        pins.digital_write_pin(RUEDA_IZQ_B, 1)
        pins.digital_write_pin(RUEDA_DER_F, 1)
        control.wait_micros(tiempo)
        pins.digital_write_pin(RUEDA_IZQ_B, 0)
        pins.digital_write_pin(RUEDA_DER_F, 0)

    basic.pause(20)
    return

def mover(direccion = True):
    global RUEDA_IZQ_F
    global RUEDA_IZQ_B
    global RUEDA_DER_F
    global RUEDA_DER_B

    if direccion:
        pins.digital_write_pin(RUEDA_IZQ_F, 1)
        pins.digital_write_pin(RUEDA_DER_F, 1)
    else:
        pins.digital_write_pin(RUEDA_IZQ_B, 1)
        pins.digital_write_pin(RUEDA_DER_B, 1)

    return

def parar():
    global RUEDA_IZQ_F
```

```
        global RUEDA_IZQ_B
        global RUEDA_DER_F
        global RUEDA_DER_B

        pins.digital_write_pin(RUEDA_IZQ_F, 0)
        pins.digital_write_pin(RUEDA_IZQ_B, 0)
        pins.digital_write_pin(RUEDA_DER_F, 0)
        pins.digital_write_pin(RUEDA_DER_B, 0)

        return
```

Algoritmo de navegación: En desarrollo

Aquí irá la descripción del algoritmo implementado para resolver el laberinto

3. Proceso de montaje

Instrucciones de montaje:

Para el proceso de montaje, no disponemos de un video o explicación propia debido a limitaciones de tiempo. Sin embargo, hemos adjuntado el video que utilizamos como referencia para construir la estructura, específicamente el ensamblaje del chasis y la instalación de los motores junto con las ruedas:

- https://youtu.be/WSMFLkL-niY?t=14

Materiales

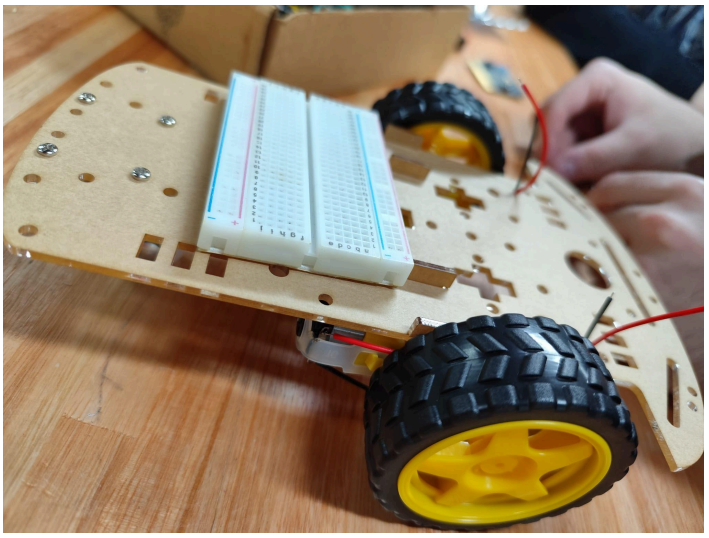
Para el chasis del robot utilizamos este kit

INCLUYE:

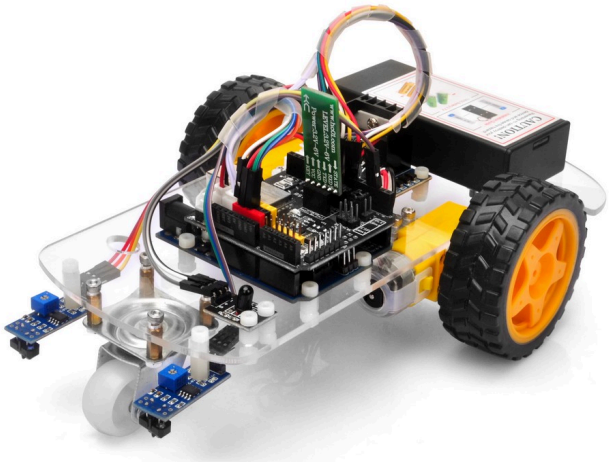
- Chasis
- 2 motores
- 2 ruedas
- 2 codificadores de velocidad
- 2 sujetadores
- 1 rueda universal
- 1 destornillador
- 1 caja para pilas: 4 AA
- Todos los tornillos y tuercas necesarias

Comenzamos montando las ruedas, los atornillamos a la placa de acrílico

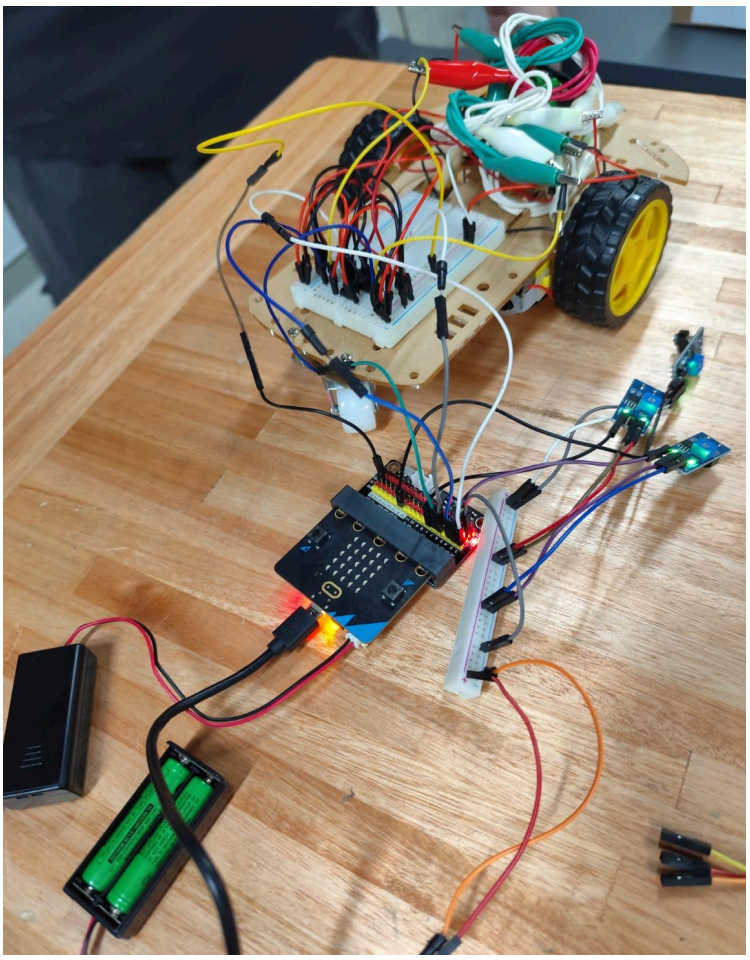
Soldamos los cables para el funcionamiento de los motores y ruedas



Aún no hemos completado el ensamblaje total del robot, ya que continuamos realizando pruebas para optimizar su estructura y funcionamiento. No obstante, así es como se vería una vez terminado.



Este es nuestro modelo preliminar



Pruebas y validación

Plan de pruebas: En desarrollo

Aquí irán los procedimientos utilizados para probar el robot, incluyendo escenarios de prueba y criterios de éxito.

Referencias

Recursos utilizados:

Enlaces a tutoriales, artículos, y manuales relevantes que ayudaron en el desarrollo del proyecto.

- Plataforma MakeCode para programar en la micro:bit: https://makecode.microbit.org/

- Plataforma TinkerCAD para realizar los circuitos simulados: https://tinkercad.com/

- Página con información de como usar el microchip para controlar los motores DC con micro:bit: https://learningdevelopments.co.nz/blogs/troystutorials/micro-bit-dc-motor-control