## ¿Qué es un Robot Mini Sumo RC?

Un robot mini sumo RC es un pequeño vehículo controlado por radiofrecuencia diseñado para participar en competencias de sumo robótico. El objetivo de este tipo de robot es detectar, empujar y sacar a su oponente fuera de un área circular sin salirse él mismo. Los mini sumo suelen tener dimensiones reducidas (hasta 10x10 cm) y un peso máximo de 500 gramos, lo que exige un diseño compacto, eficiente y con estrategias inteligentes de detección y movimiento. En su modalidad RC (radio controlado), el operador puede dirigir directamente sus acciones, combinando la agilidad del robot con la estrategia del piloto.

## Materiales

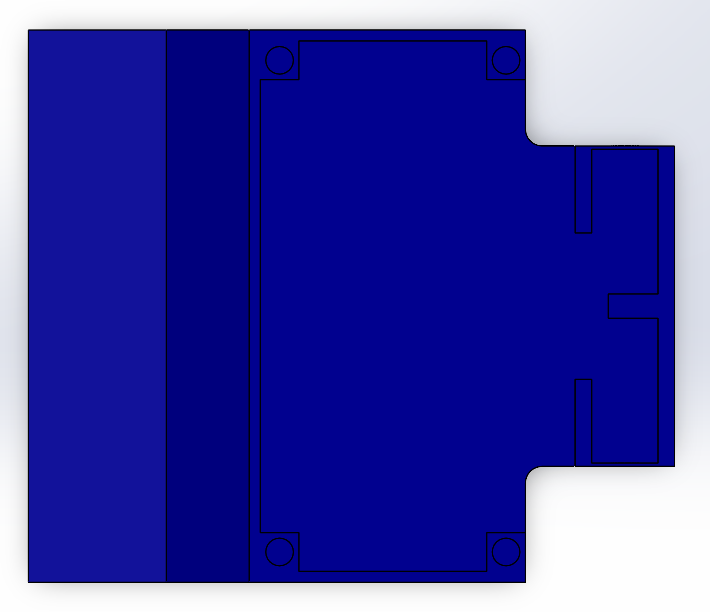
Los materiales, instrumentos, software y/hardware requeridos para la construcción del robot mini sumo RC son los siguientes:

|  |  |
| --- | --- |
| Elemento | Cantidad |
| MIT App Inventor | 1 |
| Celular con Bluetooth | 1 |
| Computador con SolidWorks | 1 |
| Arduino Nano | 1 |
| Puente H | 1 |
| Cables de conexión | NA |
| Ruedas para mini sumo | 2 |
| Motor N20 - 150 RPM | 2 |
| Batería 12 V | 1 |
| Impresión 3D de la estructura | 1 |

## Planos y Estructura en STL

Logotipo

El contenido generado por IA puede ser incorrecto.



Forma, Rectángulo

El contenido generado por IA puede ser incorrecto.

Antes de utilizar el diseño del chasis, es importante rectificar las medidas reales de los motores y ruedas adquiridos, comparándolas con el modelo CAD proporcionado. En caso de encontrar diferencias, se recomienda realizar los ajustes necesarios para asegurar un montaje correcto. Además, el diseño está abierto a modificaciones, por lo que cada estudiante o equipo es libre de realizar los cambios que considere pertinentes para potenciar el rendimiento de su mini sumo. Tanto el archivo original en formato CAD como el modelo exportado en formato STL se encuentran disponibles en el repositorio de GitHub dispuesto para el grupo.

En caso de realizar modificaciones al diseño original, es fundamental que el nuevo archivo sea exportado en formato STL, ya que este es el requerido para los procesos de impresión 3D.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

## Programación

La programación del robot se divide en dos. Para la conexión y control de movimiento por bluetooth se realiza la aplicación desarrollada en MIT App Inventor (implementando el celular como control de radiofrecuencia). Y para los motores y funcionamiento del robot, se realiza el código en el IDE de Arduino para un microcontrolador Arduino Nano, el cual funciona con interrupciones y funciones según cada una de las acciones.

Para el desarrollo de la app se sugiere tener en cuenta los siguientes tutoriales de creación de una app básica en MIT App Inventor y cómo agregar una conexión bluetooth a esta:

<https://youtu.be/YoYyMRggpro?si=q9OfGDkHdYM4N5B5>

<https://youtu.be/4iePO8k0SwU?si=JmSpojfSXJHG0p8s>

Así mismo, el código base para Arduino Nano en el IDE de Arduino, es el siguiente:

// Código base Robot mini sumo – Club de Robótica

#define IN1 3 // Motor Left

#define IN2 4 // Motor Left

#define IN3 5 // Motor Right

#define IN4 6 // Motor Right

volatile bool Stop = false; // Banderas interrupciones

volatile bool Adelante = false;

volatile bool Backward = false;

volatile bool Left = false;

volatile bool Right = false;

char command = 0; // Comando Bluetooth

void setup() {

Serial.begin(9600); // Serial y HC-05

pinMode(IN1, OUTPUT); // Definición pines

pinMode(IN2, OUTPUT);

pinMode(IN3, OUTPUT);

pinMode(IN4, OUTPUT);

attachInterrupt(digitalPinToInterrupt(2), StopI, FALLING); // Pin 2 para detener motores

attachInterrupt(digitalPinToInterrupt(3), AdelanteI, FALLING); // Pin 3 para mover hacia adelante

attachInterrupt(digitalPinToInterrupt(4), BackwardI, FALLING); // Pin 4 para mover hacia atrás

attachInterrupt(digitalPinToInterrupt(5), LeftI, FALLING); // Pin 5 para girar a la izquierda

attachInterrupt(digitalPinToInterrupt(6), RightI, FALLING); // Pin 6 para girar a la derecha

}

void loop() {

if (Serial.available()) {

command = Serial.read();

if (command == 'A') {

Adelante = true;

}

else if (command == 'B') {

Backward = true;

}

else if (command == 'L') {

Left = true;

}

else if (command == 'R') {

Right = true;

}

else if (command == 'S') {

Stop = true;

}

}

if (Adelante) {

moveAdelante();

Adelante = false;

}

if (Backward) {

moveBackward();

Backward = false;

}

if (Left) {

turnLeft();

Left = false;

}

if (Right) {

turnRight();

Right = false;

}

if (Stop) {

stopMotors();

Stop = false;

}

}

// Interrupciones

void StopI() {

stopMotors();

}

void AdelanteI() {

moveAdelante();

}

void BackwardI() {

moveBackward();

}

void LeftI() {

turnLeft();

}

void RightI() {

turnRight();

}

void stopMotors() {

digitalWrite(IN1, LOW);

digitalWrite(IN2, LOW);

digitalWrite(IN3, LOW);

digitalWrite(IN4, LOW);

}

void moveAdelante() {

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

}

void moveBackward() {

digitalWrite(IN1, LOW);

digitalWrite(IN2, HIGH);

digitalWrite(IN3, LOW);

digitalWrite(IN4, HIGH);

}

void turnLeft() {

digitalWrite(IN1, LOW);

digitalWrite(IN2, HIGH);

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

delay(280); // Revisar en prueba según ángulo de giro

stopMotors();

}

void turnRight() {

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

digitalWrite(IN3, LOW);

digitalWrite(IN4, HIGH);

delay(280);

stopMotors();

}

## Cronograma

El desarrollo del robot mini sumo RC se plantea en 9 semanas o sesiones, en las cuales se distribuye el trabajo de la siguiente manera:

|  |  |
| --- | --- |
| Semana / Sesión | Actividad |
| 1 | Introducción al Minisumo |
| 2 | Programación de la aplicación (teniendo en cuenta el tutorial base) e introducción a los componentes |
| 3 | Diseño de la estructura del robot mini sumo RC (mejoras o adaptaciones al CAD proporcionado) |
| 4 | Impresión y ensamblaje de la estructura y las partes del robot |
| 5 | Programación del microcontrolador y los motores (teniendo en cuenta el código base) |
| 6 | Conexiones del sistema eléctrico |
| 7 | Pruebas de conexión bluetooth entre el serial y el microcontrolador |
| 8 | Control del robot desde la aplicación diseñada |
| 9 | Pruebas finales del robot |

## Retos Semanales

Semana 2: Conexión exitosa del puente H con un motor y código básico para mover el motor en ambas direcciones.

Semana 3: Adaptar el diseño CAD al gusto y necesidad del equipo.

Semana 4: Imprimir y ensamblar correctamente la estructura del robot con los motores, microcontrolador y ruedas bien colocados. Evaluar que todo encaje adecuadamente y que los componentes estén fijos y centrados.

Semana 5: Utilizando el código base proporcionado, lograr que el robot realice movimientos básicos con comandos escritos directamente desde el monitor serial.

Semana 6: Realizar todas las conexiones eléctricas del sistema: microcontrolador, puente H, alimentación y motores. Verificar que no haya cortocircuitos y que los pines usados correspondan al código

Semana 7: Enviar los comandos programados desde una aplicación de comunicación serial instalada en el dispositivo que se vaya a utilizar para controlar el robot.

Semana 8: Utilizar la aplicación diseñada por el equipo para controlar completamente al robot mini sumo. El sistema debe permitir ejecutar comandos para mover o detener el robot, todo en tiempo real y sin errores de conexión.