



Universidad de Guadalajara
Centro Universitario de Ciencias Exactas e Ingenierías
División de Tecnologías para la Integración Ciber – Humana
Seminario de problemas de programación de sistemas embebidos
Sección -D01



Proyecto Final

Carrito Seguidor de Luz con ESP32

Profesor:

Chávez Martínez, Ehecatl Joel

Alumnos:

Cabrera Gómez, Sergio Nicolas Código: 222328115

Davalos Magaña, Jonathan Israel Código: 219339068

Martínez Flores, Cassandra Frine Código: 218456125

20 – 04 - 2025

Introducción

Este proyecto tiene como objetivo desarrollar un sistema robótico básico que se desplace en dirección a una fuente de luz. Utilizando un microcontrolador ESP32, sensores foto-resistivos (LDR), motores y un puente H L298N, se construyó un carrito capaz de identificar diferencias de iluminación entre dos puntos y ajustar su movimiento en consecuencia. Esta actividad permite explorar conceptos clave de control embebido, actuadores, sensores y programación.

Además, se incorporó un componente adicional de interacción física: un globo ubicado bajo la fuente de luz debe ser explotado por el carrito al alcanzarla. Para ello, se adaptó un objeto punzante al frente del vehículo que permite dicha acción al contacto directo.

Objetivo general

Diseñar e implementar un carrito seguidor de luz con ESP32 que sea capaz de desplazarse hacia una fuente luminosa fija y explotar un globo colocado debajo mediante un objeto punzante adaptado al chasis.

Objetivos específicos

- Leer datos de sensores LDR y procesar la información analógica.
- Implementar control PWM en motores para direccionamiento.
- Dirigir el movimiento del carrito hacia una fuente de luz fija.
- Lograr interacción física con el entorno (explotar un globo al llegar al destino).

Marco Teórico

Los sensores foto-resistivos cambian su resistencia según la intensidad de luz. Con un divisor de voltaje se puede leer esta variación como señal analógica. El ESP32 permite el uso de PWM mediante su módulo LEDC para controlar la velocidad de los motores a través de un puente H L298N, el cual permite invertir polaridad y controlar el giro de motores DC.

El principio de funcionamiento es detectar qué lado recibe más luz y mover el carrito hacia esa dirección reduciendo la velocidad del lado opuesto, de manera proporcional. Al integrar una interacción física (como la explosión de un globo), se agrega una acción final basada en la llegada precisa del vehículo a la fuente de luz.

Materiales

Componente	Cantidad
Motor TT	4
Módulo puente H L398N	1
Microcontrolador ESP32	1
Módulo de expansión para ESP32	1
Sensor de luz (foto-resistivo/LDR)	2
Base inferior de acrílico	1
Base superior de acrílico	1
Ruedas	4
Jumpers hembra-hembra	-
Caja de baterías (para 18650)	1
Pilas/baterías 18650	2
Tuercas	-
Tornillos (cabeza redonda o avellanada)	-
Objeto punzante (adaptado al chasis)	1
Globo	1
Foco LED (fuente de luz fija)	1

Desarrollo del Proyecto

Se ensambló el chasis utilizando placas de acrílico y se montaron los motores TT. El montaje de los sensores foto-resistivos se realizó siguiendo el diseño presentado en la guía "KUONGSHUN AD178 ESP32 WiFi Multifunctional Robot Car", específicamente en la sección correspondiente al ensamblado de los sensores de luz. Estos se ubicaron en la parte frontal del chasis, uno a la izquierda y otro a la derecha, orientados hacia adelante. Se conectaron a los pines analógicos 34 y 35 del ESP32.

El módulo L298N se conectó a los motores, permitiendo el control de dirección mediante los pines IN1 a IN4 y velocidad por PWM (ENA, ENB). El código fue implementado en Arduino IDE y se estructuró con funciones para cada acción (avanzar, detener, girar, etc.).

Posteriormente, se adaptó un objeto punzante al frente del carrito mediante sujeción física segura, con el objetivo de interactuar con un globo colocado bajo una fuente de luz LED. Al llegar a dicha fuente, el carrito impacta el globo, logrando su explosión.

Implementación Física

La implementación física consistió en montar todos los componentes sobre un chasis de doble nivel de acrílico. En la parte inferior se colocaron los cuatro motores TT con sus respectivas ruedas, firmemente sujetos mediante tornillos y tuercas. En la parte superior se instaló el ESP32 sobre su módulo de expansión, lo que facilitó el cableado hacia el puente H L298N y los sensores de luz. La batería doble 18650 fue colocada centrada para balancear el peso. Los sensores foto-resistivos fueron orientados hacia el frente, permitiendo captar la luz de manera directa. Todos los cables se organizaron para evitar interferencias mecánicas durante el desplazamiento del carrito.

El objeto punzante fue montado al frente del carrito en una posición estable, con la altura suficiente para impactar el globo colocado en el suelo bajo una fuente de luz LED fija. Esto permitió comprobar la capacidad del sistema para ejecutar una acción física al final de su recorrido.

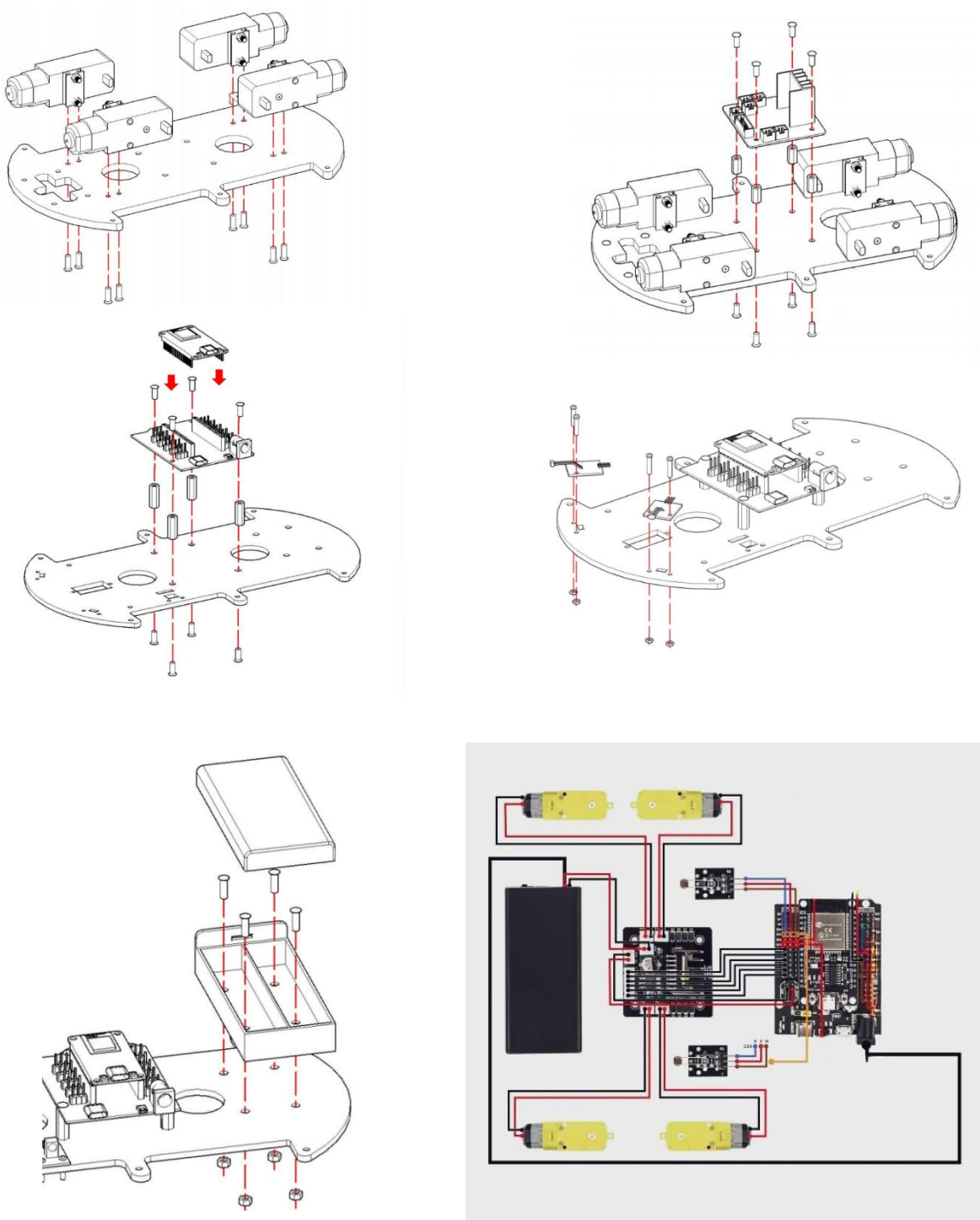
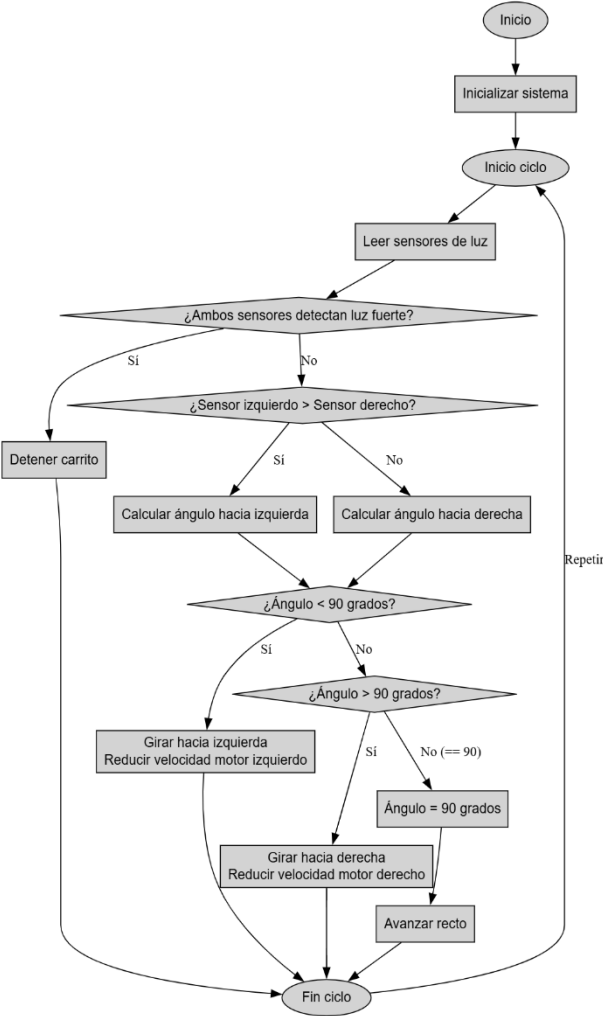
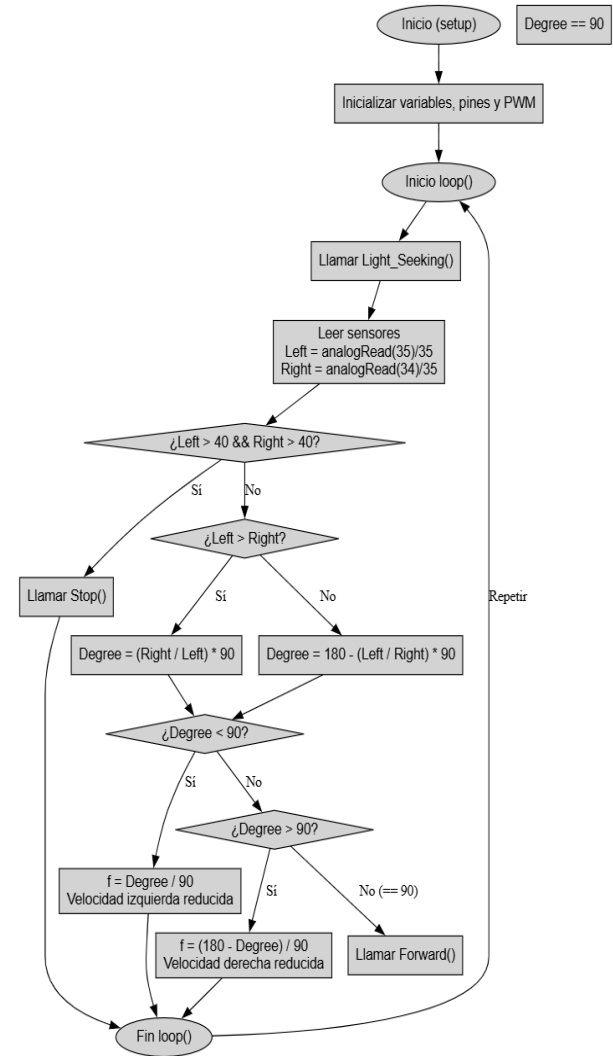


Diagrama de flujo

Se hizo un diagrama inicial para poder visualizar la idea para poder crear el código y así poder implementarlo en la ESP32.



Se hizo un segundo diagrama después de la creación del código para mejorar su funcionamiento.



Se llego al resultado planeado, con varias dificultades, pero la lógica es la misma por lo tanto su funcionamiento es el mismo.

Procedimiento

1. Ensamblar el chasis y colocar motores y ruedas.
2. Montar sensores LDR con sus resistencias según la guía de ensamblado del modelo KUONGSHUN AD178.
3. Conectar el ESP32 al módulo L298N y a los sensores.
4. Instalar físicamente todos los componentes sobre el chasis asegurando estabilidad.
5. Colocar el objeto punzante al frente del carrito.
6. Programar el ESP32 usando Arduino IDE (Versión de la ESP32 → 2.0.14, para partes del código).
7. Colocar un foco LED con un globo justo debajo, como objetivo.
8. Realizar pruebas de seguimiento hacia la luz y evaluar si el globo es impactado y explotado correctamente.
9. Ajustar umbrales y velocidad si es necesario.

Resultados

Durante las pruebas del prototipo se evaluó el comportamiento del carrito en diferentes condiciones de iluminación y se comprobó su capacidad para cumplir el objetivo de desplazarse hacia una fuente de luz y realizar una acción física (explotar un globo). A continuación, se presenta una tabla resumen de los escenarios observados:

Condición de la luz	Comportamiento del carrito
Luz centrada (simétrica en ambos sensores)	El carrito avanza en línea recta hacia la luz (cuando ambos sensores detectan luz).
Luz predominante en el sensor izquierdo	El carrito gira hacia la izquierda para alinearse con la fuente. (activa los motores derechos hasta alinearse)
Luz predominante en el sensor derecho	El carrito gira hacia la derecha para alinearse con la fuente. (activa los motores izquierdos hasta alinearse)
Luz intensa en ambos sensores (> umbral)	El sistema lo interpreta como saturación y se detiene. (en ocasiones continua hasta evitar la luz)
Ausencia total de luz	El carrito permanece detenido.
Fuente de luz con globo	El carrito avanza, impacta el globo con el objeto punzante y lo revienta.

Estas pruebas demostraron que el sistema reacciona de manera proporcional a las diferencias de luz detectadas por los sensores, lo que permite una navegación continua y suave hacia el objetivo. El control de velocidad mediante PWM fue fundamental para mejorar la precisión del giro y reducir movimientos bruscos.

La explosión del globo se logró de manera confiable en repetidas ocasiones, confirmando la correcta alineación del objeto punzante con el punto de impacto, así como la efectividad del algoritmo de navegación.

Conclusión

El proyecto permitió implementar con éxito un sistema robótico básico que responde a estímulos luminosos y realiza una acción final física al alcanzar su destino. Se cumplieron todos los requerimientos, y las pruebas demostraron una respuesta eficaz. El uso del ESP32 y el puente H facilitó un control versátil de los motores, y la lectura de sensores permitió una navegación reactiva funcional.

Además, el objetivo adicional de explotar un globo mediante un objeto punzante fue logrado exitosamente, lo cual demuestra la posibilidad de interacción con el entorno físico en sistemas de automatización sencilla.

Anexos

➔ Código

```
#include <Arduino.h>
#include <esp32-hal-ledc.h>

volatile float Left_photosensitive;
volatile float Right_photosensitive;
volatile int Lightseeking_Degree;
volatile float f;
volatile int speed_value;
#define L_light_PIN 35
#define R_light_PIN 34
#define ENA 3
#define ENA_PIN 25
#define IN1 27
#define IN2 26
#define IN3 12
#define IN4 14
#define ENB 4
#define ENB_PIN 13
#define carSpeed 250

void Forward(){
    ledcWrite(ENA, carSpeed); //enable L298n A channel
    ledcWrite(ENB, carSpeed); //enable L298n B channel
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    //Serial.println("Forward");
}

void Back() {
    ledcWrite(ENA, carSpeed); //enable L298n A channel
    ledcWrite(ENB, carSpeed); //enable L298n B channel
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    //Serial.println("Back");
}

void Left() {
    ledcWrite(ENA, carSpeed); //enable L298n A channel
    ledcWrite(ENB, carSpeed); //enable L298n B channel
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    //Serial.println("Left");
}

void Right() {
    ledcWrite(ENA, carSpeed); //enable L298n A channel
    ledcWrite(ENB, carSpeed); //enable L298n B channel
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
```

```

digitalWrite(IN4, HIGH);
//Serial.println("Right");
}

void Stop() {
  ledcWrite(ENA, 0); //enable L298n A channel
  ledcWrite(ENB, 0); //enable L298n B channel
  //Serial.println("Stop!");
}

void Light_Seeking() {
  Left_photosensitive = analogRead(L_light_PIN) / 35;
  Right_photosensitive = analogRead(R_light_PIN) / 35;
  Serial.print("Left_photosensitive:");
  Serial.println(Left_photosensitive);
  Serial.print("Right_photosensitive:");
  Serial.println(Right_photosensitive);
  Serial.println("");
  if (Left_photosensitive > 40 && Right_photosensitive > 40) {
    Stop();
  } else {
    if (Left_photosensitive > Right_photosensitive) {
      Lightseeking_Degree = ((float)(Right_photosensitive /
Left_photosensitive)) * 90;
    } else if (Left_photosensitive <= Right_photosensitive) {
      Lightseeking_Degree = 180 - ((float)(Left_photosensitive /
Right_photosensitive)) * 90;
    }
    if (Lightseeking_Degree < 90) {
      f = ((float)(Lightseeking_Degree)) / 90;
      ledcWrite(ENA, carSpeed* 0.5*f); //enable L298n A channel
      ledcWrite(ENB, carSpeed); //enable L298n B channel
      digitalWrite(IN1, LOW);
      digitalWrite(IN2, HIGH);
      digitalWrite(IN3, LOW);
      digitalWrite(IN4, HIGH);
    }
    if (Lightseeking_Degree > 90) {
      f = ((float)(180 - Lightseeking_Degree)) / 90;
      ledcWrite(ENA, carSpeed); //enable L298n A channel
      ledcWrite(ENB, carSpeed* 0.5*f); //enable L298n B channel
      digitalWrite(IN1, LOW);
      digitalWrite(IN2, HIGH);
      digitalWrite(IN3, LOW);
      digitalWrite(IN4, HIGH);
    }
    if (Lightseeking_Degree == 90) {
      f = ((float)(Lightseeking_Degree)) / 90;
      Forward();
    }
  }
}

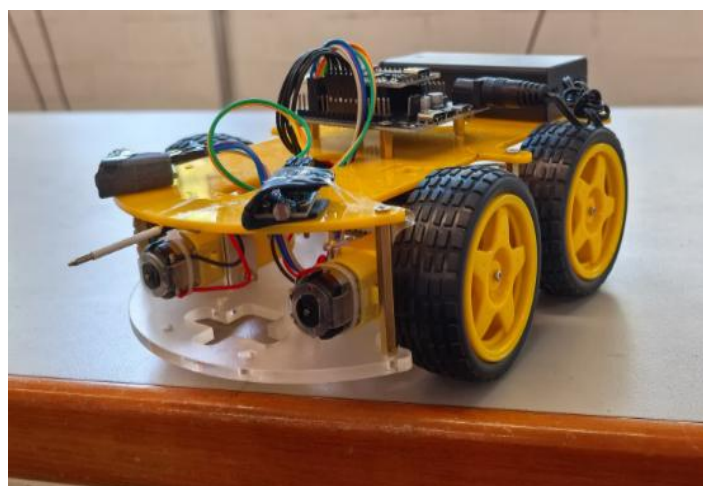
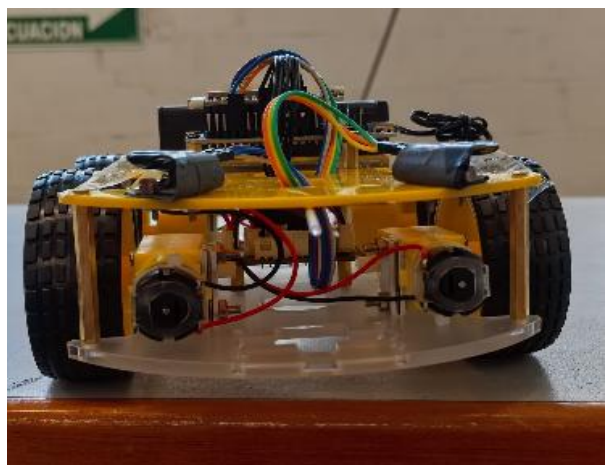
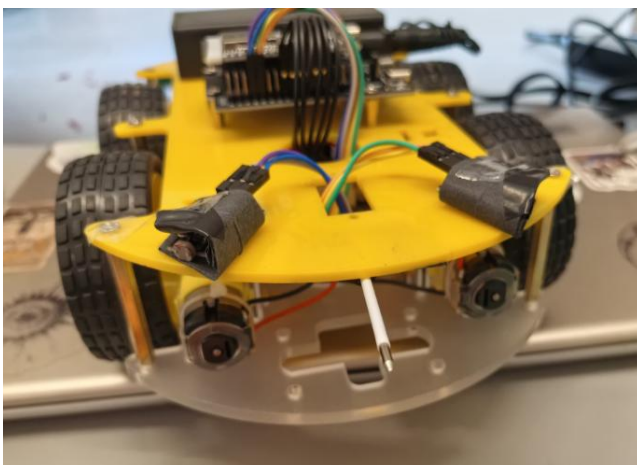
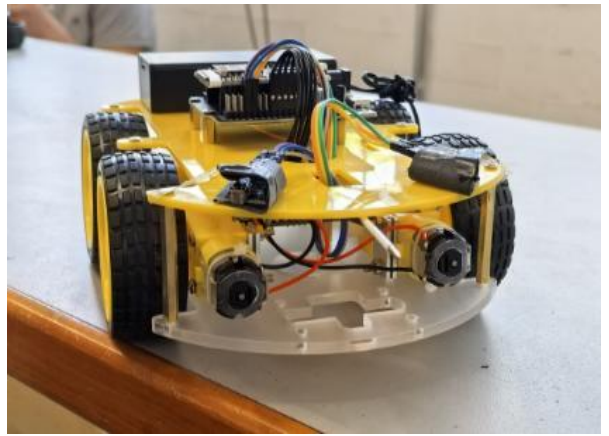
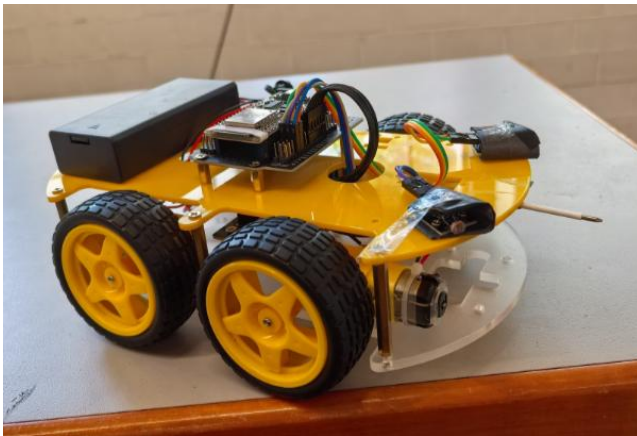
void setup(){
  Left_photosensitive = 0;
  Right_photosensitive = 0;
  Lightseeking_Degree = 0;
  f = 0;
  Serial.begin(9600);
  pinMode(L_light_PIN, INPUT);
}

```



```
pinMode(R_light_PIN, INPUT);  
pinMode(IN1, OUTPUT);  
pinMode(IN2, OUTPUT);  
pinMode(IN3, OUTPUT);  
pinMode(IN4, OUTPUT);  
pinMode(ENA_PIN, OUTPUT);  
pinMode(ENB_PIN, OUTPUT);  
ledcSetup(ENA,5000,8);  
ledcSetup(ENB,5000,8);  
ledcAttachPin(ENA_PIN,ENA);  
ledcAttachPin(ENB_PIN,ENB);  
}  
  
void loop(){  
  Light_Seeking();  
}
```

➔ Fotografías



Referencias

- Arduino - Home. Accedido el 21 de mayo de 2025. [En línea]. Disponible: <https://www.arduino.cc/reference/en/>
- “LED Control (LEDC) - ESP32 - — ESP-IDF Programming Guide latest documentation”. Technical Documents | Espressif Systems. Accedido el 21 de mayo de 2025. [En línea]. Disponible: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/ledc.html>
- “L298N Datasheet, PDF”. ALLDATASHEET.COM - Electronic Parts Datasheet Search. Accedido el 21 de mayo de 2025. [En línea]. Disponible: <https://www.alldatasheet.com/view.jsp?Searchword=datasheet%20l298n&msclkid=9f2688b735d81e5d2693e09ab09634d1>
- “Dropbox”. Dropbox.com. Accedido el 21 de mayo de 2025. [En línea]. Disponible: https://www.dropbox.com/scl/fi/9wj58r9lyj1bbaiicl2ys/KUONGSHUN-AD178-ESP32-WiFi-Multifunctional-Robot-Car.zip?dl=0&e=3&file_subpath=/KUONGSHUN+AD178+ESP32+WiFi+Multifunctional+Robot+Car/1_Assembly_Guide/Assembly+Guide.pdf&rlkey=i4z9lg4k1de383db868t7wq68&st=8ulb6g3b
- “KY-018 Datasheet(PDF)”. ALLDATASHEET.COM - Electronic Parts Datasheet Search. Accedido el 21 de mayo de 2025. [En línea]. Disponible: <https://www.alldatasheet.com/datasheet-pdf/pdf/2045019/AGELECTRONICA/KY-018.html>