



Reporte. Seguidor de Línea

Unidad Académica Multidisciplinaria Mante (UAMM)

Arquitectura de Computadoras

Montoya Garza Luis Ángel

Hernández Ruiz Haydee Michelle

Rueda Martínez Alison Michelle

Silva Sánchez Yamilka Arely

7° "E"

M.C. López Piña Daniel

Ciudad Mante, Tamaulipas. Octubre del 2025

Introducción

El robot Seguidor de Línea es un proyecto fundamental en robótica, diseñado para moverse de forma autónoma a lo largo de una trayectoria marcada, generalmente una línea negra sobre un fondo claro. Este sistema es un excelente ejemplo de un control de retroalimentación básico: el robot utiliza sensores para "ver" la línea, y luego el cerebro (Arduino) decide cómo activar los motores para corregir su rumbo y mantenerse siempre en el camino. Esencialmente, es un juego constante de detección de errores y corrección de la dirección.

Desarrollo

Un seguidor de línea es un robot móvil diseñado para desplazarse a lo largo de una trayectoria marcada en el suelo, generalmente una línea negra sobre fondo blanco. Los componentes que se usaron fueron: el Arduino uno, el cual Es el cerebro del robot. Recibe la información de los sensores y, basándose en la programación, envía las órdenes al puente H para controlar los motores. Un puente H que Actúa como la interfaz de potencia. El Arduino solo puede emitir señales de bajo voltaje y corriente, insuficientes para los motores. El puente H recibe las señales lógicas del Arduino y utiliza una fuente de energía externa (batería) para suministrar la potencia necesaria, permitiendo además controlar la dirección de giro. Los 2 sensores infrarrojos son básicamente los ojos del robot. Estos sensores típicamente emiten luz infrarroja y miden la cantidad que rebota. Por último, un chasis que proporciona la movilidad al robot.

El núcleo del programa es una serie de decisiones (basadas en estructuras if/else if) que evalúan las cuatro posibles combinaciones de lecturas de los sensores para determinar el movimiento del robot. Cuando ambos sensores ven blanco (lectura 0 y 0), la condición es que el robot está centrado perfectamente sobre el fondo, lo que significa que aún no ha encontrado la línea o que esta se encuentra justo entre ellos. La acción es avanzar recto. La estrategia es simple: los dos motores giran hacia adelante a la misma velocidad para mantener el curso. Si el sensor izquierdo ve negro y el sensor derecho ve blanco (lectura 1 y 0), la condición es que el robot se ha desviado demasiado hacia la izquierda y su sensor izquierdo

ha pisado la línea negra. La acción inmediata es girar a la derecha para volver a centrarse. La estrategia para este giro es un giro de pivote: el motor izquierdo avanza y el motor derecho retrocede. Esta acción provoca que el robot rote rápidamente sobre su eje, empujándolo hacia la derecha hasta que el sensor izquierdo vuelva a ver el fondo blanco. A la inversa, si el sensor derecho ve negro y el sensor izquierdo ve blanco (lectura 0 y 1), la condición es que el robot se ha desviado hacia la derecha. La acción necesaria es girar a la izquierda para corregir la trayectoria. La estrategia es el mismo giro de pivote, pero invertido: el motor derecho avanza y el motor izquierdo retrocede, rotando el robot hacia la izquierda. Finalmente, si ambos sensores ven negro (lectura 1 y 1), la condición es que el robot está completamente sobre la línea negra. Esto generalmente indica que ha llegado a un cruce, al final del recorrido o ha ocurrido un error significativo en la trayectoria. La acción predeterminada es detenerse como medida de seguridad. La estrategia consiste en mandar la señal para que ambos motores se detengan (velocidad cero y pines de dirección en LOW).

Código:

```
// Pines del motor A (izquierdo)
int IN1 = 8;
int IN2 = 9;

// Pines del motor B (derecho)
int IN3 = 10;
int IN4 = 11;

// Pines de velocidad (PWM)
int ENA = 5;
int ENB = 3;
```

```

// Sensores infrarrojos
int sensorIzquierdo = 12;
int sensorDerecho = 13;

void setup() {
  // Configurar pines de motor
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENA, OUTPUT);
  pinMode(ENB, OUTPUT);

  // Sensores
  pinMode(sensorIzquierdo, INPUT);
  pinMode(sensorDerecho, INPUT);

  // (opcional) Inicia el monitor serial para depuración
  Serial.begin(9600);
}

void loop() {
  // Leer sensores
  int lecturalzq = digitalRead(sensorIzquierdo);
  int lecturaDer = digitalRead(sensorDerecho);

  // Mostrar lecturas por serial
  Serial.print("Izq: ");
  Serial.print(lecturalzq);
  Serial.print(" | Der: ");
  Serial.println(lecturaDer);

  // Decisiones según sensores
  if (lecturalzq == 0 && lecturaDer == 0) {
    // Ambos sensores ven blanco → avanzar
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);

    analogWrite(ENA, 90);
    analogWrite(ENB, 90);
  }
}

```

```

else if (lecturalzq == 1 && lecturaDer == 0) {
    // Sensor izquierdo ve negro → girar a la derecha
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);

    analogWrite(ENA, 90);
    analogWrite(ENB, 90);
}
else if (lecturalzq == 0 && lecturaDer == 1) {
    // Sensor derecho ve negro → girar a la izquierda
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);

    analogWrite(ENA, 90);
    analogWrite(ENB, 90);
}
else {
    // Ambos sensores ven negro → detener (cruce, final, error)
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);

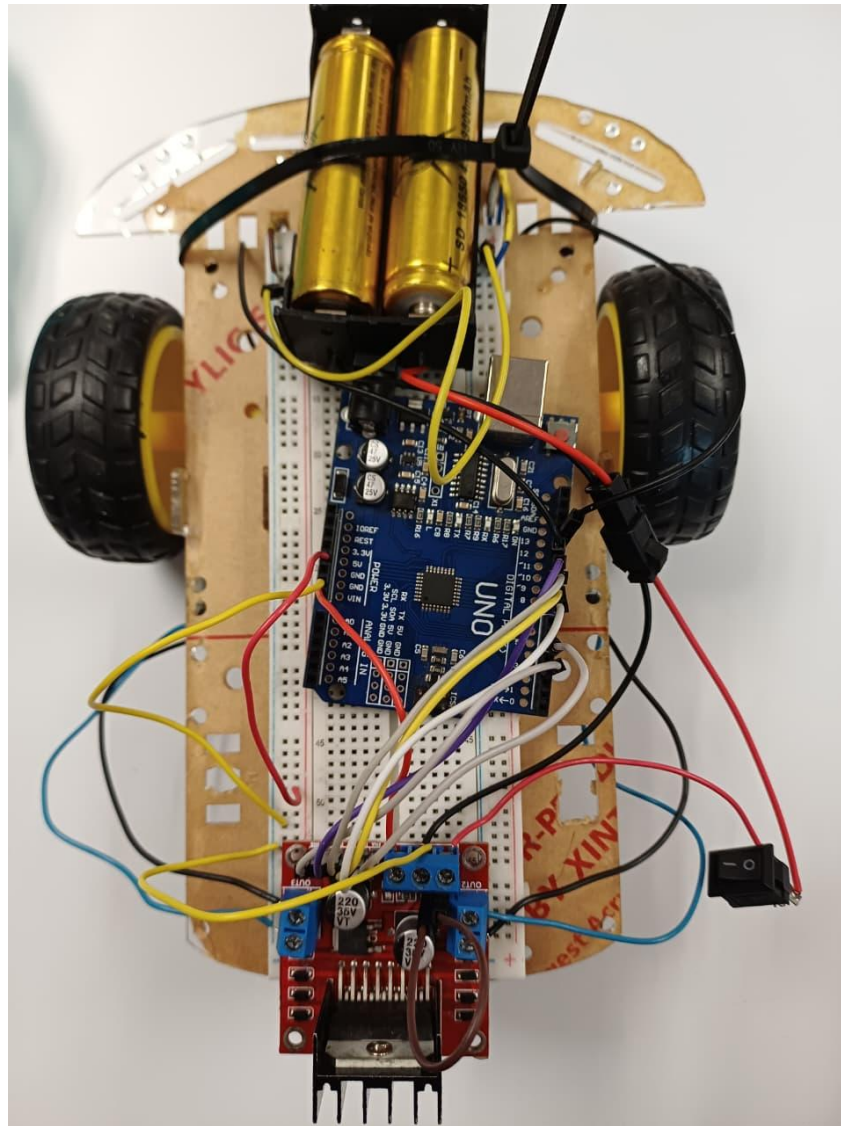
    analogWrite(ENA, 0);
    analogWrite(ENB, 0);
}

delay(10); // Pequeña pausa para estabilidad
}

```

Evidencia

<https://github.com/luisangelmg11/Arquitectura-de-Computadoras/blob/main/1.8%20Practica%20Evidencia%20Video.mp4>



Conclusión

El desarrollo de este robot Seguidor de Línea con Arduino Uno representa un logro fundamental en la comprensión de los sistemas mecatrónicos. El proyecto demuestra con éxito cómo la interacción entre los tres componentes principales (sensores, controlador y actuadores) permite la autonomía de una máquina.

La implementación de la lógica de control de dos puntos (usando dos sensores para detectar desviaciones) permite que el robot realice correcciones binarias (girar izquierda o girar derecha) de manera eficiente para mantenerse sobre el trayecto. A través del control de potencia proporcionado por el Puente H y la toma de decisiones del Arduino, el robot ejecuta el ciclo esencial de la robótica: Sentir, Pensar y Actuar.

Perfiles GitHub

<https://github.com/luisangelmg11>
<https://github.com/HaydeesitaSJJH>
<https://github.com/alisonrueda>
<https://github.com/Yamixjk>