Avance proyecto 2 Seguidor de línea con arduino

Nathaly Sanchez Hincapie, Beymar Ruidiaz Martinez.
Universidad del Quindío, laboratorio general de electrónica, Microprocesadores

Armenia, Quindío, Colombia

ns7hincapie@gmail.com
beymar.ruidiaz.m@gmail.com

***Resumen*— Este es un avance del proyecto dos, el cual es la implementación un seguidor de línea con la tarjeta programable arduino. Este está constituido por una parte en el software Arduino, que es la parte programable o (nivel de control) y la parte de implementación y mecánica o (nivel físico), con sus respectivos componentes como : ruedas, motores, batería, puente H (L293D),etc.**

Palabras clave — . puente H, ruedas,batería.

I. INTRODUCCIÓN

Un seguidor de línea es un robot móvil que se clasifica en el grupo de la rama de la robótica móvil. Su tarea fundamental es el desplazamiento en un entorno ya sea conocido o desconocido, por ende es necesario que posea tres funciones fundamentales, la locomoción (nivel físico),la percepción (nivel sensorial) y la decisión (nivel de control).En este caso la pista va a presentar una trayectoria marcada la cual se le configura de manera que debe diferenciar el entorno en este caso, negra con un fondo blanco. Para poder lograr este objetivo, se implementó un sensor de infrarrojos de corto alcance (CNY70), por su propiedad de emisor de luz y un receptor siempre apuntando en la misma dirección.

II. OBJETIVOS

- Objetivo General:
  Diseñar un Robot seguidor de línea (en este caso negra) en el suelo, con código de C++ en el programa arduino.
- Objetivo especifico:
  1. Estudiar el funcionamiento de un Robot seguidor de línea y sus características.
  2. Despertar la creatividad de los alumnos para tener un buen desempeño en un desarrollo de análisis y tecnológico.
  3. Implementar un algoritmo, para realizar la toma de decisiones del Robot seguidor de línea.
  4. Construir la estructura adecuada para que se adapte y facilite el desplazamiento del robot en la pista.
  5. Aprender a interpretar lo que se tiene a nivel físico y transportarlo al software de simulación Proteus.

III. MARCO TEORICO

- Nivel Físico:

1. Estructura :
   Se implementó una estructura para el seguidor de línea de una lámina de acrílico cuyas dimensiones son (15x15). Medidas que se tomaron después de hacer ciertos montajes. Esta estructura tiene como fin ser el apoyo para los motores, el circuito, la tarjeta programable Arduino, y otros componentes. Véase en la figura 1.

Figura 1. Base o estructura en Acrilico.

2. Motores y llantas:
   La mayor parte de los motores que se utilizan en un robot presentan un torque bajo por lo tanto gira demasiado rápido, también que los motores serán controlados de corriente (700mAh) y de voltaje (6v).
   Al tener en cuenta esto se escogió un motor y llanta que presentara una buena piñonería y las llantas deben ser de caucho o de un plástico blando para que estas no patinen. Véase en la figura 2.

Figura 2. Motor y llantas escogidas (traseras).

Figura 3. Rueda Loca (delantera).

Se escogieron tres llantas, para que la parte delantera de esta estructura quedará un poco más baja debido a que allí irán los sensores detectando lo que encuentra este en el suelo. En este caso (línea negra) en un entorno blanco.
Al implementar ya las tres ruedas, se obtiene una estructura así. Véase en la figura 4 (capa superior) figura 5 (capa inferior).

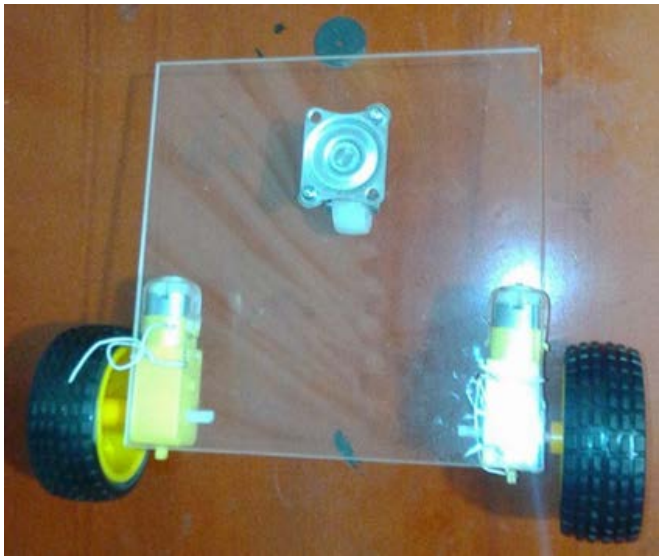

Figura 4. (Capa superior) Montaje de las tres llantas.

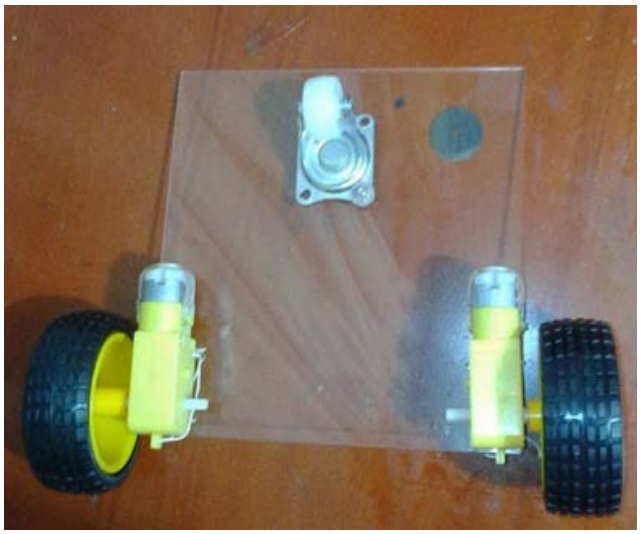

Figura 5. (Capa inferior) Montaje de las tres llantas.

3. Nivel sensorial:
   Debido a que este robot para poder avanzar se guía en la trayectoria negra dibujada en un entorno blanco, su percepción es tipo visual, esta captación consiste en poder diferenciar entre dos colores (blanco y negro), para este caso trayectoria negra en una superficie blanca.
   Se usó el sensor CNY70, ya que presenta una propiedad física llamada reflexión, este diodo emite una luz infrarroja dirigida hacia el suelo y un fototransistor recibe los fotones que se generan gracias al proceso de reflexión que se produce en el suelo.
   Al implementarse en el seguidor de línea, si este sensor se encuentra con la línea negra el voltaje sube, pero si este lee la superficie blanca el voltaje baja.

Figura 6. Físico del Diodo CNY70.

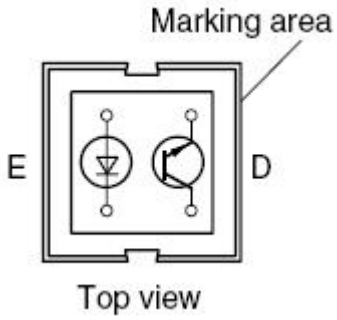

Figura 7. Interno del Diodo CNY70.

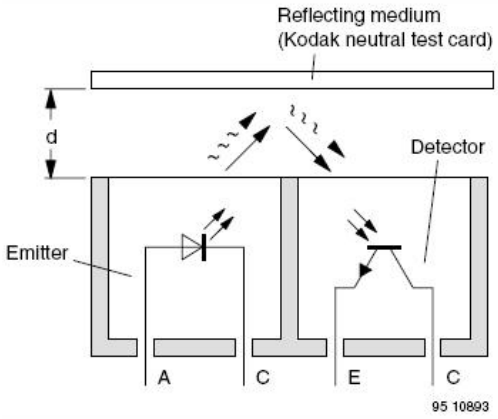

Figura 8. Gráfico de la detección de la luz.

4. Nivel de control:
   El circuito de control es el que proporciona las señales hacia los actuadores dependiendo de las señales

que se obtengan de los sensores.

4.1 Circuito comparador y etapa de potencia:

Para permitir el cambio de dirección de un motor de corriente continua (motor CC) o motor de mediana potencia, se usa el puente H, el sentido de giro del motor, depende de los niveles de voltaje que existan en los puntos del circuito llamados (avance y retroceso), en el cual solo uno de estos dos puntos puede estar a nivel alto para activar los transistores correspondientes.

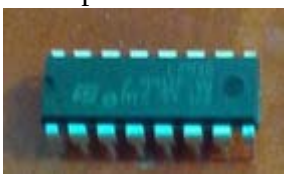

Figura 9. Puente H.

Los drivers para motores (L293 o L298) en este caso se hace uso del (L293) debido a que este se encuentra en el mercado fácilmente, permiten el cambio de dirección, frenado y manejo de mayores corrientes.

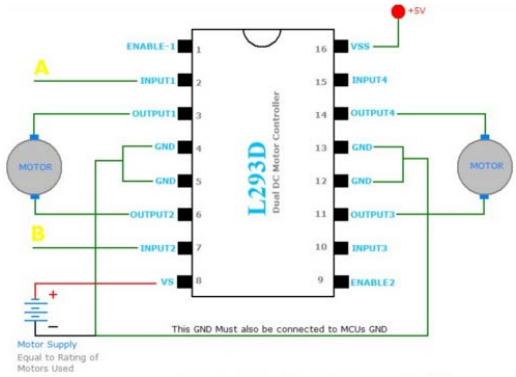

Figura 10. Esquemático del Puente H .

4.2 Tabla de valores de la relación de las señales de los sensores con las señales de control sobre los motores.

Estado	S_i	S_d
A	0	0
B	0	1
C	1	0
D	1	1

Tabla 1 . Valores de relación de señales.

Donde Sx=o, este indica que el sensor no esta sobre la trayectoria negra, y Sx=1, indica que el sensor se encuentra sobre la línea.

- Estado A: En este estado ambos sensores se encuentran fuera de la trayectoria, por lo tanto ambos motores se detienen y el seguidor de línea debe se colocado encima de esta de manera manual. Pero al implementar el algoritmo de corrección de trayectoria mejora.
- Estado B: En este estado el móvil se desvió hacia el lado izquierdo, dejando así el sensor derecho sobre la trayectoria. La solución es suspender la circulación de la energía de el motor derecho para que el motor izquierdo activo corrija la trayectoria.
- Estado C: El seguidor de línea se ha desviado hacia el lado derecho respecto a la trayectoria, solo el sensor izquierdo se encuentra sobre la trayectoria negra. La solución es suspender la energía de el motor izquierdo para que el motor derecho activo corrija la trayectoria.
- Estado D: Ambos sensores se encuentran sobre la trayectoria negra, no necesita ningún tipo de suspensión, ya que ambos se encuentran activos de manera correcta y en la trayectoria recta, manteniendo su velocidad hasta que encuentre una curva nuevamente.

Una vez se tengan estos componentes, se realizo el montaje físico. Véase en la Figura 11.

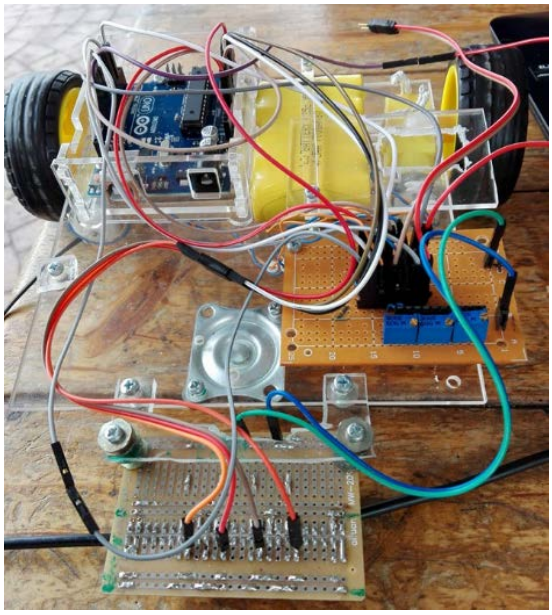

Figura11. Montaje final del Robot seguidor de línea.

IV. DESARROLLO

Una vez se establece la parte del montaje físico, se realiza el nivel de control mediante un código en la plataforma arduino. Como se mostrará a continuación.

```
int izqA = 5;
int izqB = 6;
int derA = 9;
int derB = 10;
int vel = 255; // Velocidad de los motores (0-255)

void setup()  {
  pinMode(derA, OUTPUT);
  pinMode(derB, OUTPUT);
  pinMode(izqA, OUTPUT);
  pinMode(izqB, OUTPUT);
 }

void loop()  {
    analogWrite(derB, 0);  // Detiene los Motores
    analogWrite(izqB, 0);
    delay (500);
    analogWrite(derA, vel);  // Frente 2 segundos
    analogWrite(izqA, vel);
    delay (2000);

    analogWrite(derA, vel);  // Derecha 0,5 segundos
    analogWrite(izqA, 0);
    delay (500);

    analogWrite(derA, 0);    // Izquierda 0,5 segundos
```

Figura 12. Primera parte del código del seguidor de línea.

```
void loop()  {
    analogWrite(derB, 0);  // Detiene los Motores
    analogWrite(izqB, 0);
    delay (500);
    analogWrite(derA, vel);  // Frente 2 segundos
    analogWrite(izqA, vel);
    delay (2000);

    analogWrite(derA, vel);  // Derecha 0,5 segundos
    analogWrite(izqA, 0);
    delay (500);

    analogWrite(derA, 0);    // Izquierda 0,5 segundos
    analogWrite(izqA, vel);
    delay (500);

    analogWrite(derA, 0);    // Detiene los Motores
    analogWrite(izqA, 0);
    delay (500);
    analogWrite(derB, vel);  // Reversa 2 segundos
    analogWrite(izqB, vel);
    delay (2000);
}
```

Figura 13. Segunda parte del código del seguidor de línea.

Este código se realiza, para poder verificar que el integrado (L293D) realizara correctamente su funcionamiento, dándole un buen control manejo a los motores respectivos con sus llantas.

Una vez comprobado esto, se implementó el código para el control del robot seguidor de línea.

```
// Definimos los pines digitales que usaremos
int boton = 8;
int motor_derechaadelante = 11;   // Con este pin, dirigimos el motor derecho hacia adelante
int motor_izquierdaadelante = 12; // Con este pin, dirigimos el motor izquierdo hacia adelante
int motor_derechaatras = 4;    // Con este pin, dirigimos el motor derecho hacia atras
int motor_izquierdaatras = 2;  // Con este pin, dirigimos el motor izquierdo hacia atras
int pwm_izquierdo = 10;            // pin de manejo de velocidad del motor izquierdo
int pwm_derecho = 9;           // Pin de manejo de la velocidad del motor derecho
int ledOn = 6;
int ledPlay = 7;

// Pines analogos
int sensor_1 = 0;
int sensor_2 = 1;
int sensor_3 = 2;
int sensor_4 = 3;


// Variables iniciadas en cero
float proporcional = 0;
float proporcional_anterior= 0;
int posicion = 0;
int posicion_ultima = 0;
float integral = 0;
float derivativo = 0;
float pwm_valor = 0;
```

Figura 14. Primera parte del programa para el nivel de control del robot seguidor de línea.

En esta parte se inicia con la declaración de las variables que se van a utilizar, definiendo el tipo de variable, es decir, en este caso int (entero), que quiere decir que todas las variables serán implementadas tipo entero, cada una con su respectivo pin, adicional a esto se inicializan

algunas variables, ya que al hacer esto, facilita más la implementación del PID.

Se definen cuatro sensores, (sensor_1, sensor_2,sensor_3,sensor_4) cada uno de estos van a los pines del arduino analógico que se representan con A0, A1,A2,A3, respectivamente.

Adicionalmente se conectan unos pines (pwm) que representados en los puertos del arduino 9, 10, que van conectados a los pines del puente H, controlando características de los motores como la velocidad.

Se implementa un PID que está conformado por un mecanismo de control por realimentación ampliamente usado en sistemas de control industrial. Este calcula el error entre un valor medido y un valor deseado. Este control se basa en tres pautas, **proporcional** depende del error actual. El **Integral** depende de los errores pasados y el **Derivativo** es una predicción de los errores futuros, adicionalmente el derivativo corrige el sobre impulso del sistema permitiendo de esta forma que el sistema llegue a su referencia. La suma de estas tres acciones es usada para ajustar al proceso por medio de un elemento de control como la posición de una válvula de control o la potencia suministrada a un calentador, o en este caso el robot seguidor de línea.

```
// Constantes del PID
float kp=0.20; //2,
float kd=12;
float ki=0.001;


void setup() {

  delay(2000);   // Hacemos una espera de 2 segundos

  Serial.begin(9600);  // iniciamos comunicacion serial con 9600 bouds


  // Configuramos los pines como entrada o salida
  pinMode(boton,INPUT_PULLUP);
  pinMode(motor_derechaadelante,OUTPUT);
  pinMode(motor_izquierdaadelante,OUTPUT);
  pinMode(motor_derechaatras,OUTPUT);
  pinMode(motor_izquierdaatras,OUTPUT);
  pinMode(ledOn,OUTPUT);
  pinMode(ledPlay,OUTPUT);
  pinMode(pwm_izquierdo,OUTPUT);
  pinMode(pwm_derecho,OUTPUT);

  int b = digitalRead(boton); // Leemos el valor en el pin boton (8)

// Ciclo de espera del pulso del boton de inicio
```

Figura 15. Segunda parte del código establecido para el robot seguidor de línea.

Se definen algunos pines con configuración de entrada(INPUT) o de salida(OUTPUT), dependiendo de la variable que se vaya a medir para hacer uso de este valor y de esta forma implementarlo en el resto de líneas de código. Como se observa en la Figura 5.

.

```
digitalWrite(ledPlay,HIGH); // Led que indica inicio de lectura de sensores

 // Lectura de los sensores en las entradas analogas
int s_1 = analogRead(sensor_1);
int s_2 = analogRead(sensor_2);
int s_3 = analogRead(sensor_3);
int s_4 = analogRead(sensor_4);



int s_11 = 0; // Variables usadas para la conversion
int s_12 = 0;
int s_13 = 0;
int s_14 = 0;
//Serial.println(s_2);

// Calibracion de sensores para convertirlos a binario
if(s_1>150){
  s_11=1;
}
if(s_2>250){
  s_12=1;
}
if(s_3>150){
  s_13=1;
}
if(s_4>300){
  s_14=1;
```

Figura 16. Cuarta parte del código seguidor de línea.

Una vez estén definidas las entradas analógicas, se hace uso de unas variables que permiten la conversión, realizando una calibración para convertirlos en binario, haciendo uso de la función de decisión IF, las variables relacionadas con los números de calibración de los sensores. Ver Figura16.

```
    posicion = (1*s_11)+(2*s_12)+(4*s_13)+(8*s_14);  // Convertimos los numeros binarios obtenidos de los sensores en decimales

if(posicion <= 14){
  posicion_ultima = posicion;   // Almacenamos las posiciones leidas
}

Serial.println(posicion_ultima);  // Mostramos en pantalla el valor de la posicion leida por los sensores

  // Calculamos los valores proporcional, integral y derivativo
  proporcional = (posicion - 9);
  integral = (integral + proporcional_anterior);
  derivativo = (proporcional - proporcional_anterior);

  // Valor obtenidos del PID
  pwm_valor = (kp*proporcional)+(ki*integral)+(kd*derivativo);

  //Serial.println(valor_pwm);  // Mostramos el valor del PID obtenido

  // Limitamos el valor de la integral
  if(integral>200){
   integral=200;
  }
  if(integral<-200){
```

Figura 17. Quinta parte del código seguidor de línea.

Gracias a los números convertidos a binario, se puede saber la posición en la que se encuentra el

seguidor, una vez se tiene esta, se va almacenando sucesivamente las posiciones que se van leyendo,
Serial.println(posicion_ultima) esta línea muestra el valor en pantalla de la posición leída por los sensores.
Una vez calculados los valores proporcional, integral y derivativo, por medio de fórmulas establecidas. Los valores obtenidos del PID harán parte de la modulación por ancho de pulso.

Figura 18. Sexta parte del código del seguidor de línea.
Se establecen condiciones para la lectura de los sensores, ya que lo que se lee de estos les dará la dirección a motores. Las condiciones que se usaron fueron IF, y cada una de las salidas fue puesto en pines ya sea analogWrite, todo esto basándose en si los sensores se encuentran en las posiciones 9 y 8.

Figura 19. Séptima parte del código del seguidor de línea.
Basándose en la variable pwm_valor, se usa la estructura de decisión IF, para establecer el direccionamiento del motor si este gira hacia la derecha o hacia la izquierda.
Debido a que los sensores que se utilizaron no tienen el voltaje convencional es decir (5 en negro, 0 en blanco), si todos los sensores se encuentran sobre la línea negra el carro frena, es decir, para.

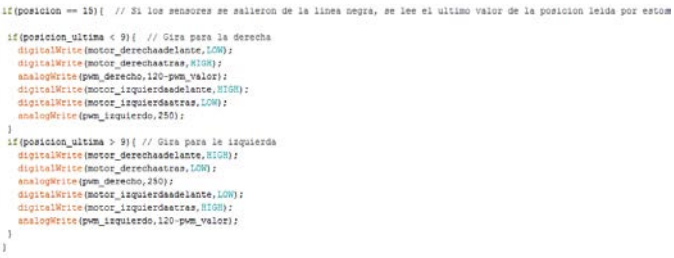

Figura 20. Octava parte del código del seguidor de línea.
En esta parte se configura la opción en la que dado el caso los sensores se salieran de la línea negra, este leerá el ultimo valor de la posición leída por estos y por último se configuro cuando este girara hacia la izquierda.
Una vez se tiene el montaje en nivel físico, y el nivel de control, se realizan pruebas, para corroborar el buen funcionamiento del robot seguidor de línea, una vez hecho esto, se comprueba que el estado de este es correcto y funciona a la perfección.

V. CONCLUSIONES

- El diseño implementado para el montaje del seguidor de línea fue el correcto, porque permitió la ubicación de los sensores, las llantas fueron aptas y tuvieron buen desempeño, manteniendo en equilibrio su eje y los componentes que estas sostenían.
- La herramienta de programación Arduino permite simplificar la funcionalidad del robot seguidor de línea.
- Estimula la creatividad y desarrollo de los conocimientos adquiridos hasta el momento, tanto en la parte programable como en la parte mecánica y técnica de este.
- Los emisores infrarrojos se deben calibrar a una distancia prudente del suelo ya que deben tener el espacio necesario para que cuando el emita un rayo de luz tenga la oportunidad de reflejarse y regresar para ser captada por el sensor.
- Los sensores CNY70 depende de los valores correctos de las resistencias y de una buena ubicación de los sensores es fundamental ya que estos son los encargados de regular que el carro siga la trayectoria de la línea negra.

VI. REFERENCIAS BBLIOGRAFICAS

[1] http://es.slideshare.net/Michael_CrL/seguidor-de-linea-13770660

[2] https://prezi.com/zs1t6ijtmr3g/robot-seguidor-de-linea/

[3] http://aprender.tdrobotica.co/seguidor-de-lineaprofesional/

[4] http://www.jmnlab.com/cny70/cny_70.html.

[5] http://www.x-robotics.com/robots_simples.html.

[6] http://expoelectronica.upbbga.edu.co/pdf_2010_XVI/jhair_rodriguez_c.pdf.

[7] http://elprofegarcia.com/?p=111.

Nathaly Sanchez Hincapie. nació en Toro, Valle Colombia, en enero 6 de 1998.Se graduó de bachiller en el Colegio Corazonista. En la actualidad se encuentra estudiando ingeniería electrónica en la Universidad Del Quindío. Ponente en la VIII Jornada Técnico-científica de la Facultad de Ingeniería.

Beymar Ruidiaz Martinez. Nació en Puente nacional Santander, Colombia, el 22 de diciembre de 1986. Actualmente es estudiante de decimo semestre de Ingeniería Electrónica de la Universidad del Quindío, área de interés automatización y control, perteneció al semillero de investigación de automatización y control.