

PROYECTO INTEGRADOR

Dilan Santiago Valencia Ramos. (126857)¹ and Maykol Estiben Valero Hernández. (127014)¹

¹ dilans.valenciar@ecci.edu.co

¹ Mayole.valerohe@ecci.edu.co

Resumen— Para este proyecto, desarrollaremos un carro seguidor de línea, cuyas dimensiones aproximadas son de 15 cm de alto, 15 cm de ancho y 28 cm de largo. Este carro tiene la capacidad de seguir una línea blanca y, al detectar una línea negra ubicada a los costados de la pista, debe retomar su trayectoria en la línea blanca. El objetivo es que, al detectar una línea negra con ambos sensores simultáneamente, el carro frene. Adicionalmente, el carro debe completar el recorrido en menos de 15 segundos. Durante su trayecto, debe ser capaz de identificar una serie de colores situados en su camino y, al detectar dichos colores, debe mostrar por cuál color está pasando. Todo este sistema se desarrollará en Quartus, ya que es el software adecuado para enviar las señales a través de la FPGA (Field-Programmable Gate Array). El diseño y construcción del carro implican una serie de pasos específicos: 1) Diseño y Ensamblaje del Carro: Chasis y Estructura: Selección de materiales livianos pero resistentes para construir el chasis. 2) Sensores: Instalación de sensores infrarrojos o de color que permitan detectar las líneas y colores en la pista. Motores y Ruedas: Integración de motores de corriente continua (DC) con ruedas adecuadas para proporcionar tracción y control de velocidad. Programación y Control: 3) Algoritmo de Seguimiento: Desarrollo de un algoritmo que permita al carro seguir la línea blanca y corregir su dirección al detectar la línea negra. Control de Motores: Implementación de un control preciso de los motores para asegurar que el carro siga la trayectoria deseada y frene adecuadamente cuando sea necesario. Detección de Colores: Programación de los sensores para identificar los colores en el camino y mostrar esta información en tiempo real. Implementación en FPGA: 4) Diseño de Hardware en Quartus: Creación del diseño lógico en Quartus para controlar el carro mediante la FPGA. 5) Pruebas y Validación: Realización de pruebas exhaustivas para asegurar que el sistema funcione correctamente bajo diferentes condiciones y ajustes. Este proyecto no solo implica la construcción física del carro, sino también el diseño y programación de un sistema complejo de control y detección. La utilización de la FPGA y el software Quartus garantiza que el carro tendrá la capacidad de procesar las señales y ejecutar las acciones necesarias en tiempo real, asegurando un rendimiento óptimo y cumpliendo con los requisitos establecidos.

Palabras clave— Sensores, Control de motores, FPGA

Abstract— For this project, we will develop a line-following car with approximate dimensions of 15 cm in height, 15 cm in width, and 28 cm in length. This car has the capability to follow a white line and, upon detecting a black line located on the sides of the track, it should return to its path on the white line. The objective is for the car to stop when it detects a black line with both sensors simultaneously. Additionally, the car must complete the course in less than 15 seconds. During its journey, it should be able to identify a series of colors located along its path and, upon detecting these colors, it should display which color it is passing over. This entire system will be developed in Quartus, as it is the appropriate software for sending signals via the FPGA (Field-Programmable Gate Array). The design and construction of the car involve several specific steps: Selection of lightweight yet sturdy materials to construct the chassis. Installation of infrared or color sensors to detect the lines and colors on the track. Integration of direct current (DC) motors with appropriate wheels to provide traction and speed control. Development of an algorithm that allows the car to follow the white line and correct its direction upon detecting the black line. Implementation of precise motor control to ensure the car follows the desired path and stops adequately when necessary. Programming the sensors to identify the colors on the path and display this information in real-time. Creation of the logical design in Quartus to control the car via the FPGA. Conducting exhaustive tests to ensure the system functions correctly under different conditions and adjustments. This project not only involves the physical construction of the car but also the design and programming of a complex control and detection system. Utilizing the FPGA and Quartus software ensures that the car will have the capability to process signals and execute the necessary actions in real-time, guaranteeing optimal performance and meeting the established requirements. In addition to its technical capabilities, this project aims to provide a comprehensive learning experience in robotics, automation, and embedded systems. By integrating various technologies and methodologies, it will enhance our understanding of real-time systems, sensor integration, and control algorithms. The process of troubleshooting and refining the car's performance will also develop critical problem-solving skills and encourage innovation. As we iterate on the design, we will document our findings and improvements, contributing valuable insights to the field of autonomous vehicles. Ultimately, this project not only seeks to build a functional line-following car but also to advance our knowledge and skills in engineering and technology, preparing us for future challenges in the rapidly evolving landscape of robotics and automation.

Keywords— Motor control, Sensors, FPGA .

1. INTRODUCCIÓN

El desarrollo de un carro seguidor de línea constituye un proyecto integral que abarca desde el diseño mecánico hasta la programación avanzada y la implementación de sistemas electrónicos. Este carro, con dimensiones aproximadas de 15 cm de alto, 15 cm de ancho y 28 cm de largo, estará diseñado para seguir una línea blanca en una pista y corregir su trayectoria al detectar una línea negra en los costados, frenando automáticamente cuando ambos sensores detecten la línea negra simultáneamente. Además, el carro debe completar el recorrido en menos de 15 segundos, y durante su trayecto, debe ser capaz de identificar y mostrar una serie de colores ubicados en su camino. Para llevar a cabo este proyecto, utilizaremos el software Quartus, que es esencial para programar y enviar señales a través de la FPGA (Field-Programmable Gate Array). La elección de Quartus y la FPGA responde a la necesidad de un control preciso y una rápida respuesta a las señales de los sensores, garantizando así que el carro pueda realizar las acciones necesarias en tiempo real. El proceso de diseño y construcción del carro implicará varios pasos clave, desde la selección de materiales livianos pero resistentes para el chasis, hasta la instalación de sensores infrarrojos o de color, y la integración de motores de corriente continua (DC) con ruedas adecuadas para proporcionar la tracción y el control de velocidad necesarios. Además, se desarrollará un algoritmo de seguimiento que permitirá al carro seguir la línea blanca y corregir su dirección al detectar la línea negra. También se implementará un control preciso de los motores para asegurar que el carro siga la trayectoria deseada y frene adecuadamente cuando sea necesario. La programación de los sensores permitirá identificar los colores en el camino y mostrar esta información en tiempo real. Finalmente, se crearán el diseño lógico en Quartus y se realizarán pruebas exhaustivas para asegurar que el sistema funcione correctamente bajo diferentes condiciones y ajustes. Este proyecto no solo implica la construcción física del carro, sino también el diseño y programación de un sistema complejo de control y detección. La utilización de la FPGA y el software Quartus garantizará que el carro tenga la capacidad de procesar señales y ejecutar las acciones necesarias en tiempo real, asegurando un rendimiento óptimo y cumpliendo con los requisitos establecidos.

2. METODOLOGIA

Para la elaboración de este carro, en primer lugar, procederemos a adquirir un chasis de calidad que posea la resistencia adecuada para soportar el peso de los componentes que serán instalados sobre él. Además, es fundamental que este chasis cuente con una movilidad óptima para readaptar su posición ante los eventuales contratiempos que puedan surgir durante el recorrido en la pista. El chasis seleccionado también estará equipado con dos motores, los cuales serán responsables de proporcionar la fuerza necesaria para el desplazamiento del vehículo. Para garantizar que estos motores reciban las señales de control adecuadas, será necesario instalar un puente H. Este componente, ampliamente utilizado en diversos campos como la robótica, el control de motores en vehículos

eléctricos y los sistemas de posicionamiento, permite controlar con precisión y eficiencia la velocidad y la dirección de un motor eléctrico. La implementación de estos motores con puente H asegurará que el vehículo pueda ajustar su rotación cuando se encuentre con una línea negra, así como proporcionar la fuerza motriz necesaria para su desplazamiento. Los detalles específicos sobre cómo se llevó a cabo esta implementación se detallarán conforme se muestra en la Figura 1.

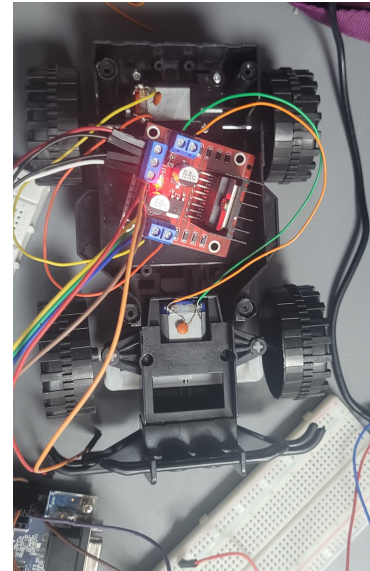


Fig. 1: Montaje Motores y PH

el código Debounce implementa un mecanismo de debounce para el botón de un sistema digital, crucial para evitar múltiples pulsaciones o fluctuaciones de señal al presionar un botón físico. Con entradas como el reloj del sistema para sincronización (clk), una señal de reinicio (Reset) y el estado del botón de entrada (Buttonin), junto con un parámetro que define el tiempo de debounce (DEBOUNCETIME), el módulo busca garantizar que solo se registre una pulsación limpia y confiable del botón. Al operar en flancos de subida del reloj o la señal de reinicio, este mecanismo ajusta su estado actual y siguiente según el estado del botón y el tiempo transcurrido, lo que permite determinar cuándo la señal del botón es válida tras un período de estabilización definido. Mediante un proceso bien definido, el código gestiona el estado del debounce, asegurando que el sistema responda de manera confiable a las pulsaciones del botón. Este proceso implica la transición entre estados, la evaluación de la señal del botón y el conteo del tiempo transcurrido, todo ello diseñado para garantizar que la señal del botón esté estable durante un tiempo definido antes de ser considerada válida. De esta manera, el módulo se encarga de mitigar problemas asociados con el ruido eléctrico y las transiciones no deseadas de la señal, asegurando un comportamiento coherente y confiable del botón este se programa como se ve en la Figura 2.

```

1 module Debounce(
2     input clk,
3     input Reset,
4     input Button_in,
5     output reg Button_out
6 );
7
8     parameter DEBOUNCE_TIME = 100000;
9
10    typedef enum {IDLE, COUNTING, VALID} states_t;
11    states_t current_state, next_state;
12    reg [31:0] counter = 0;
13
14    always @(posedge clk or posedge Reset) begin
15        if (Reset) begin
16            current_state <= IDLE;
17            counter <= 0;
18            Button_out <= 0;
19        end else begin
20            current_state <= next_state;
21            case (current_state)
22                IDLE: begin
23                    if (Button_in) begin
24                        counter <= counter + 1;
25                    end else begin
26                        counter <= 0;
27                    end
28                end
29                COUNTING: begin
30                    if (Button_in) begin
31                        counter <= counter + 1;
32                        if (counter == DEBOUNCE_TIME) begin
33                            Button_out <= 1;
34                        end
35                    end else begin
36                        counter <= 0;
37                    end
38                end
39                VALID: begin
40                    if (!Button_in) begin
41                        counter <= counter + 1;
42                        if (counter == DEBOUNCE_TIME) begin
43                            Button_out <= 0;
44                        end
45                    end else begin
46                        counter <= 0;
47                    end
48                end
49            endcase
50        end
51    end
52
53    always @(current_state or Button_in or counter) begin
54        case (current_state)
55            IDLE: begin
56                if (Button_in) next_state = COUNTING;
57            end
58            COUNTING: begin
59                if (!Button_in) next_state = IDLE;
60                else if (counter == DEBOUNCE_TIME) next_state = VALID;
61                else next_state = COUNTING;
62            end
63            VALID: begin
64                if (!Button_in) next_state = COUNTING;
65                else next_state = VALID;
66            end
67        endcase
68    end
69 end
70 endmodule
71
72

```

Fig. 2: Debounce

Este código implementa una lógica de frenado para el carro, crucial para su seguridad y control durante su funcionamiento. Utilizando señales de entrada como el estado del carro y el reloj del sistema, junto con un módulo de debounce para garantizar la fiabilidad de las señales de entrada, el sistema gestiona eficazmente el proceso de frenado en respuesta al estado actual del vehículo. El módulo de debounce asegura que la señal de entrada que indica el estado del carro se establezca antes de ser procesada, evitando así fluctuaciones no deseadas que podrían causar acciones de frenado incorrectas. Esta estabilidad se logra mediante un proceso de filtrado que elimina el ruido eléctrico y las transiciones no deseadas, garantizando que el sistema responda de manera precisa y confiable a los cambios de estado del carro. El proceso de control de frenado se basa en la enumeración de estados del carro, que incluyen "Stop"(detenido), "Goto"(en movimiento) y "Still"(quieto). Dependiendo del estado actual del vehículo, el sistema activa o desactiva la acción de frenado, asegurando que el carro se detenga adecuadamente cuando sea necesario y se mueva de manera segura cuando corresponda. Esta lógica de frenado contribuye significativamente a la seguridad y eficiencia operativa del carro, brindando un control preciso y fiable en diversas condiciones de funcionamiento.

```

1 module Freno(
2     input state,
3     input clk,
4     output reg Brake
5 );
6
7     wire Brake1;
8
9     reg [31:0] count = 0;
10
11    Debounce Debounce_inst1(
12        .clk(clk),
13        .Reset(!state),
14        .Button_in(state),
15        .Button_out(Brake1)
16    );
17
18    typedef enum reg [2:0] {Stop, Goto, Still} state_type;
19    state_type first_state, next_state;
20
21    always @(posedge Brake1) begin
22        first_state <= next_state;
23    end
24
25    always @(first_state) begin
26        case(first_state)
27            Stop: next_state = Still;
28            Still: next_state = Goto;
29            Goto: next_state = Stop;
30        endcase
31    end
32
33    always @(*) begin
34        Brake = ~(first_state == Stop | first_state == Still);
35    end
36 endmodule
37

```

Fig. 3: Freno

Este módulo implementa un generador de señales PWM para controlar la velocidad de un motor u otra carga en el carro. Utilizando una señal de reloj para sincronizar las operaciones, así como las entradas de periodo y ciclo de trabajo, el módulo produce una señal de salida PWM que varía la velocidad de la carga según el ciclo de trabajo especificado. Esto permite una gestión precisa y eficiente de la velocidad, lo que es fundamental para el control y la funcionalidad óptima del vehículo en diversas condiciones de operación.

El proceso de generación de la señal PWM se basa en un contador que lleva el seguimiento del tiempo y se reinicia al final de cada período. Durante cada ciclo de reloj, la señal de salida PWM se ajusta según el ciclo de trabajo especificado, determinando la proporción de tiempo que la señal está en estado alto en relación con el período completo. Esta capacidad de control fino proporciona al carro una respuesta precisa y adaptable, lo que mejora su rendimiento y su capacidad de adaptación a distintas demandas de operación..

```

1 module Generador_PWM (
2     input clk,
3     input [15:0] period,
4     input [15:0] duty_cycle,
5     output reg pwm_out
6 );
7
8     reg [15:0] counter = 16'b0;
9
10    always @(posedge clk) begin
11        if (counter < duty_cycle) begin
12            pwm_out <= 1'b1;
13        end else if (counter < period) begin
14            pwm_out <= 1'b0;
15        end
16        if (counter == period - 1) begin
17            counter <= 16'b0;
18        end else begin
19            counter <= counter + 1'b1;
20        end
21    end
22 endmodule
23

```

Fig. 4: Generador

El módulo Control Motor T.^{es} un componente fundamental en el sistema de control del carro, amalgamando una variedad de funcionalidades para asegurar su operación segura y eficiente. Con entradas para los motores derecho e izquierdo, un sensor de luz LDR, una señal de reloj y una de freno, este módulo orquesta el control de los motores y la administración del freno en respuesta a las condiciones del entorno y las acciones del conductor. Además, emplea parámetros y registros para definir los

períodos y ciclos de trabajo de las señales PWM, junto con un mecanismo de debounce para estabilizar las señales de entrada y prevenir fluctuaciones no deseadas. El proceso de control de los motores se sustenta en condiciones lógicas que determinan la dirección y la velocidad del carro en función de las entradas de los botones derecho e izquierdo. Simultáneamente, se generan señales PWM mediante dos módulos "Generador PWM", lo que permite regular con precisión la velocidad de cada motor. Además, se implementa un módulo de freno que ajusta la acción de frenado del carro en respuesta a la señal correspondiente, asegurando una detención segura cuando sea necesaria. A través de la integración de estas funcionalidades, el módulo Control Motor T ϕ contribuye a la seguridad y la funcionalidad óptima del carro, proporcionando un control preciso de los motores y una respuesta efectiva ante diversas condiciones de conducción. Al coordinar el control de los motores, la generación de PWM y la gestión del freno, este módulo garantiza un rendimiento consistente y confiable del vehículo en distintos escenarios de operación. Su diseño modular y su enfoque holístico en el control y la seguridad hacen de este componente una pieza clave en el funcionamiento integral del carro.

```

1 module Control_Motor_T (
2     input wire Right,
3     input wire Left,
4     input wire LDR,
5     input clk,
6     input Brake,
7     output wire pwm_out,
8     output wire pwm_out2,
9     output reg Input1,
10    output reg Input2,
11    output reg Input3,
12    output reg Input4
13);
14
15    parameter [15:0] period = 2500;
16    parameter [15:0] duty_cycle = 1000;
17    parameter [15:0] duty_cycle_two = 1825;
18
19    reg [0:1] start;
20
21    wire Button_out;
22    wire Button_out_right;
23    wire Button_out_left;
24    wire Button_out_Brake;
25
26
27    initial begin
28        Input1 <= 0;
29    end
30
31    Debounce Debounce_inst(
32        .clk(clk),
33        .Reset(!LDR),
34        .Button_in(LDR),
35        .Button_out(Button_out)
36    );
37
38    always @(posedge Button_out) begin
39        if (Button_out) begin
40            start = start + 1;
41            if (start == 3) begin
42                Input1 <= 1;
43            end
44        end
45    end
46
47    always @(posedge clk) begin
48        if (Left == 1) begin
49            Input3 = 1;
50            Input4 = 0;
51        end
52        if (Right == 1) begin
53            Input3 = 0;
54            Input4 = 1;
55        end
56        if ((Right == 0 && Left == 0) || (Right == 1 && Left == 1)) begin
57            Input3 <= 0;
58            Input4 <= 0;
59        end
60    end
61

```

Fig. 5: Motores Final

3. RESULTADOS Y DISCUSIÓN

El proyecto de desarrollo del carro seguidor de línea se complementó con una variedad de materiales y componentes adicionales para garantizar su funcionalidad y rendimiento óptimo. Entre estos materiales se incluyeron comparadores de voltaje, como el LM393, LM358 o LM324, que se utilizaron para procesar las señales de los sensores y

convertirlas en señales digitales claras para la placa base del vehículo. Los sensores, incluyendo un módulo fotoresistivo, se utilizaron para detectar las franjas de color en la pista y ajustar la trayectoria del carro en consecuencia. Los jumpers se emplearon para interconectar los diferentes componentes del sistema, mientras que la FPGA actuó como el cerebro del carro, procesando las señales de los sensores y controlando los actuadores según sea necesario. Además, se utilizaron actuadores específicos para el funcionamiento del carro, como motores DC para la propulsión del vehículo y drivers de motor como el L293D o L298N para controlar los motores DC. Para indicar la detección de las franjas de colores en la pista, se emplearon LEDs de diferentes colores, como amarillo, verde, rojo y azul, proporcionando una retroalimentación visual clara al usuario. El puente H fue utilizado para controlar la dirección y la velocidad de los motores, permitiendo al carro seguir la línea de manera precisa y eficiente. En cuanto a la alimentación del carro, se utilizó una batería recargable, como una batería LiPo o NiMH, junto con un conector tipo XT60 hembra para la conexión de la batería. Además, se incorporaron reguladores de voltaje, convertidores buck o fuentes step down como el LM2596 para ajustar el voltaje a las necesidades específicas del circuito del carro. Las borneras se utilizaron para conectar los diferentes componentes del sistema, proporcionando una interfaz segura y confiable. Otros componentes esenciales para la construcción del vehículo incluyeron ruedas y chasis adecuados, una placa de prototipos o PCB para montar y conectar los componentes electrónicos, un interruptor de encendido/apagado y un LED indicador de encendido para indicar el estado del sistema. Se requirieron también resistencias, transistores, diodos y otros componentes electrónicos básicos según sea necesario, así como cables y conectores para la interconexión de los diferentes elementos del carro. Los sensores antirrebote se utilizaron para estabilizar las señales de entrada, evitando fluctuaciones no deseadas y garantizando un funcionamiento fiable del sistema...

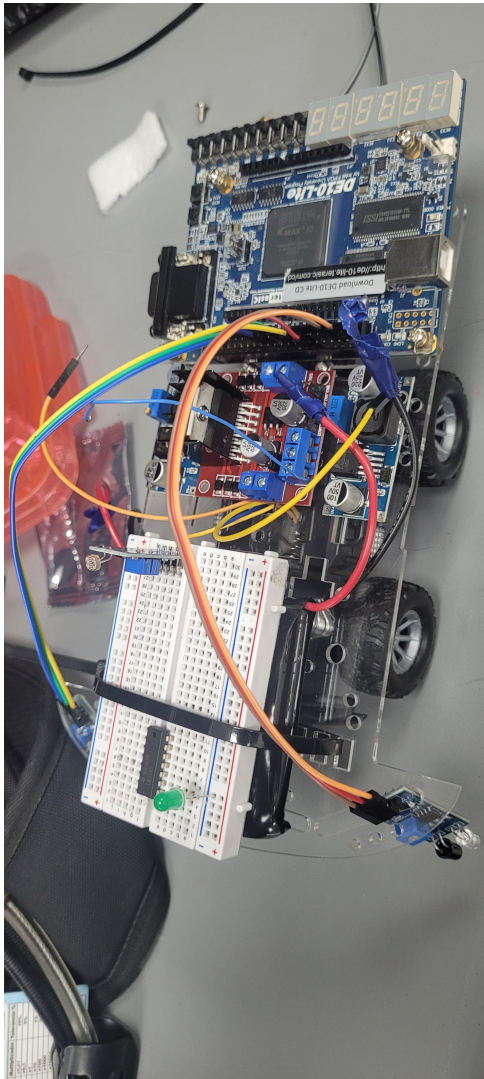


Fig. 6: CARRITO FINAL

4. CONCLUSIONES

En este proyecto de desarrollo de un carro seguidor de línea, se realizó una meticulosa selección de materiales y componentes para garantizar el óptimo rendimiento del vehículo. Comenzando con la adquisición de un chasis robusto y con movilidad adecuada, se aseguró la base sólida sobre la cual construir el sistema. La elección de motores DC y la implementación de un puente H permitieron un control preciso de la dirección y velocidad del carro, esenciales para seguir la línea y ajustar su trayectoria según las condiciones del entorno. El uso del código de debounce desempeñó un papel crítico en la estabilización de las señales de entrada, evitando pulsaciones múltiples o fluctuaciones no deseadas al interactuar con los botones físicos. Esta función fue esencial para garantizar una respuesta confiable del sistema, especialmente en situaciones críticas como el frenado del vehículo. La combinación del módulo de frenado y el control preciso de los motores mediante señales PWM contribuyó significativamente a la seguridad y eficiencia operativa del carro, asegurando su capacidad para detenerse adecuadamente y responder de manera precisa a los cambios en el entorno. La integración de una variedad de componentes, desde comparadores de

voltaje hasta actuadores y sistemas de alimentación, permitió la construcción de un carro completo y funcional. La combinación de estos elementos, junto con una cuidadosa programación y diseño electrónico, resultó en un vehículo capaz de seguir una línea blanca en una pista, corregir su trayectoria al detectar una línea negra, y completar su recorrido en un tiempo específico. El enfoque modular y holístico en el diseño y la implementación de este proyecto garantizó un rendimiento consistente y confiable del carro seguidor de línea en diversas condiciones de operación, cumpliendo así con los objetivos establecidos para el proyecto.

4. CONCLUSIONES

En este proyecto, se desarrolló un carro seguidor de línea con dimensiones específicas que le permiten seguir una línea blanca en una pista, corregir su trayectoria al detectar una línea negra y frenar cuando ambos sensores detectan simultáneamente una línea negra. Adicionalmente, el carro debía completar el recorrido en menos de 15 segundos e identificar una serie de colores en su camino, mostrando esta información en tiempo real. La elección de materiales ligeros pero resistentes para el chasis y la instalación de sensores infrarrojos y de color fueron fundamentales para la detección precisa de las líneas y los colores en la pista.

La programación y el control del carro se lograron mediante el desarrollo de un algoritmo para seguir la línea blanca y corregir la dirección al detectar la línea negra. También se implementó un control preciso de los motores utilizando señales PWM para asegurar que el carro siguiera la trayectoria deseada y frenara adecuadamente cuando fuera necesario. La programación de los sensores para identificar los colores en el camino y mostrar esta información en tiempo real añadió una capa adicional de funcionalidad y complejidad al sistema.

La implementación en FPGA a través del software Quartus fue crucial para el control del carro. El uso de la FPGA permitió un procesamiento rápido y eficiente de las señales de los sensores y la ejecución de las acciones necesarias en tiempo real. La creación del diseño lógico en Quartus y la realización de pruebas exhaustivas aseguraron que el sistema funcionara correctamente bajo diversas condiciones, garantizando un rendimiento óptimo y fiable del carro seguidor de línea.

En conclusión, este proyecto no solo logró construir un carro seguidor de línea funcional, sino que también proporcionó una valiosa experiencia en el diseño y programación de sistemas de control y detección complejos. La combinación de un chasis bien diseñado, motores y sensores precisos, y el uso de FPGA y Quartus permitió alcanzar los objetivos del proyecto. La integración de estas tecnologías demostró la importancia de un enfoque holístico y modular para el desarrollo de sistemas autónomos eficientes y fiables en el campo de la robótica y la automatización.

5. AGRADECIMIENTOS

Los autores desean expresar su sincero agradecimiento a Ronald Rodriguez Rodriguez y a pirlo por ayudarme con

Belico para el progreso y la velocidad de la ejecución de este y por su invaluable apoyo y asistencia a lo largo de la duración de este proyecto de investigación. Su experiencia y orientación fueron fundamentales para la finalización exitosa de este trabajo. Además, los autores expresan su aprecio lo que contribuyó significativamente a la calidad de la investigación. Este trabajo no habría sido posible sin su constante estímulo y cooperación.