

Fase 2. Robot móvil para recorrer un camino con obstáculos

Carlos Fernando Quintero Olaya (2010006),
carlos.fernando.quintero@correounivalle.edu.co,
Joan Esteban Velasco Larrea (1827539)
joan.velasco@correounivalle.edu.co
Jhonier Andrés Vargas (1745647),
jhonier.vargas@correounivalle.edu.co

Resumen. En el siguiente informe se explica cómo fue diseñado la plataforma mecánica, especificando sus dimensiones, su material y como fue la distribución de cada componente, dentro de la plataforma, de tal manera que se explica el funcionamiento de cada sensor para el diseño de un Robot seguidor de línea que evade obstáculos. Por otra parte, también se explica la configuración y el desarrollo de software por módulos, para el sistema de movimiento, detección de obstáculo y detección de la línea, de tal manera que se explica cómo se programa un AVR, a través del lenguaje C. Cabe resaltar que para el diseño de este robot, se sigue con la misma metodología RUP planteada en la Fase 1, en la cual se hace un análisis de requerimientos funcionales y no funcionales, y el diseño de diagramas secuenciales, conceptuales y diagramas de flujo.

Palabras clave. Robot Seguidor de Línea, Programación AVR, Puente H, TCRT5000, FC-51

I. INTRODUCCIÓN

Al momento de llevar a cabo el desarrollo de un proyecto que esté relacionado con producción de software para un fin o acción, en este caso un robot móvil capaz de recorrer un camino y evadir obstáculos que estén presentes en medio de este, es importante tener claro un norte u objetivos, así como también el cómo llevar a cabo la programación del microcontrolador que será usado para este proyecto como medio para el funcionamiento del robot móvil. Es por eso por lo que siguiendo la metodología RUP, se llevaron a cabo los desarrollos de los módulos correspondientes al sistema de movimiento, sistema de detección de obstáculo y sistema de detección de línea, como también el diseño de la plataforma mecánica.

II. DISEÑO DE LA PLATAFORMA MECÁNICA

El robot móvil es un sistema sencillo, en donde su plataforma fue diseñada con acrílico. Este sistema cuenta 2 motores y 2 ruedas omnidireccionales, las cuales se controlan con el puente H - L298N, dependiendo de las detecciones que arrojan los sensores infrarrojos FC-51 para la detección de obstáculos y el sensor TCRT5000 para el seguimiento de la línea, siendo estos conectados directamente a la plataforma de Arduino (Hardware), la cual tiene el microcontrolador ATMEGA328P. Cabe resaltar, que el sensor HC- SR04 que fue mencionado en la fase 1, fue reemplazado por el sensor FC-51, ya que este es mucho más cómodo de manejar y configurar ya que no hay necesidad de configurar una señal de reloj para la medición de distancia, y a su vez no hace uso de muchos pines.

Por otro lado, otro detalle importante a mencionar es que el robot tiene una altura de 7.5 cm, compuesto por una plataforma circular de 2 pisos, la cual tiene una dimensión de 14 cm de diámetro, tal como se observa en la Figura 1.

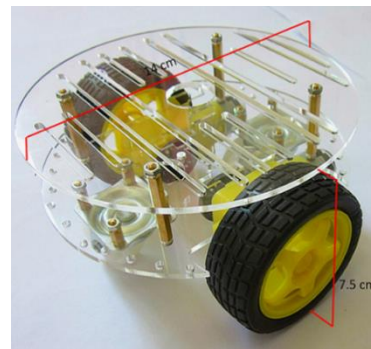


Fig 1. Dimensiones del Robot móvil.

Ahora bien, la distribución de los componentes dentro de esta plataforma quedó de la siguiente manera:

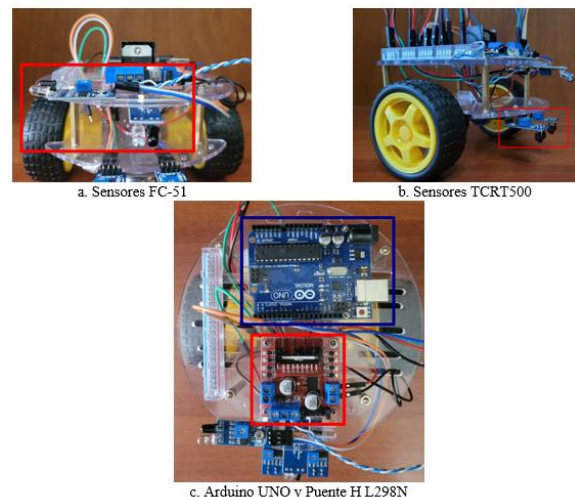


Fig 2. Configuración y Distribución de los sensores

Como se observa en la Figura 2, se observan los siguientes componentes:

- Para la Figura 2.a, se observa los dos sensores FC-51 en el recuadro rojo, los cuales están puestos en el segundo piso del robot. De este modo, uno monitorea

si en la parte frontal hay un obstáculo y el otro sirve para monitorear si en la parte derecha del robot sigue el obstáculo.

- Para la Figura 2.b , se observa que los dos sensores TCRT5000 en el recuadro rojo, los cuales están puestos en la primera planta del robot. Estos sensores, se encargan de verificar por donde va el robot, es decir, está constantemente evaluando si esta sobre la línea negra o sobre la superficie blanca.
- Para la Figura 2.c, se observa la plataforma Arduino con el microchip ATMEGA328P, señalada por el recuadro azul, la cual sirve para controlar y recibir las señales digitales que le envían cada sensor, de tal manera que se encarga de procesar y ejecutar diferentes acciones para cada evento. Por otra parte, en la misma figura 2.C, se observa el puente H, señalado por el recuadro rojo, el cual se encarga de ejecutar como deberán girar los motores, dependiendo de los eventos que procese el microchip ATMEGA328.

Cabe resaltar, que a pesar del cambio del sensor de ultrasonido (HC-SR04) por el sensor infrarrojo (FC-51), el sistema mantiene la misma lógica conceptual, por lo tanto, el diagrama conceptual es el mismo al de la fase 1.

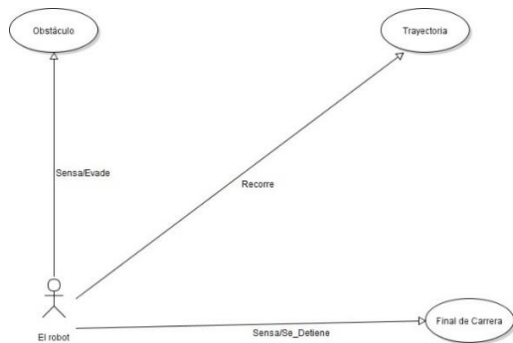


Figura 3. Diagrama Conceptual del Robot Móvil para Recorrer un Camino con Obstáculos.

Como se observa en la Figura 3, el robot móvil se encarga de evadir los Obstáculos, recorrer la Trayectoria, y detenerse en el Final de Carrera.

III. DESARROLLO MODULAR

A. SISTEMA DE MOVIMIENTO

En este sistema de movimiento se hace uso, además del microcontrolador ATmega328p, el módulo L298N. Este módulo es un puente H que permite regular la velocidad y el sentido de giro de los motores que se le conecten. Éste módulo cuenta con una serie de jumpers, que según como estén configurados estos jumpers el módulo puede manejar voltajes de alimentación mayores a 12 volts, agregando una alimentación para la parte lógica del módulo, y también puede manejar señales PWM (a través del pin ENA y ENB). La

siguiente figura muestra el funcionamiento del módulo para uno de los motores, el motor contrario funciona de igual forma pero con las entradas correspondientes ENB, IN3 e IN4.

ENA	IN1	IN2	Descripción Del MotorA
0	x	x	Apagado
1	0	0	Apagado
1	0	1	Girar hacia adelante
1	1	0	Girar hacia atrás
1	1	1	Apagado

Tabla 1. Tabla de Verdad del módulo L298N.

Los pines de ENA y ENB son útiles para las señales del tipo PWM, sin embargo en este proyecto se dejó conectado el jumper en estos pines. También se dejó conectado el jumper del regulador de voltaje, pues se utiliza una fuente menor a 12 volts (6 volts).

Para el control en el sentido de giro en los motores se creó un archivo “.h” en el que se definieron una serie de constantes: Una constante I que representa el puerto de salida D para la comunicación unidireccional con el módulo L298N, cuatro constantes I1, I2, I3 e I4 que almacenan la posición del bit del puerto que se encuentra directamente conectado con el módulo. Además, en este “.h” se definió cinco funciones del tipo void: “TForwards” (Avanzar), “TBackwards” (Retroceder), “TRight” (Girar Derecha), “TLeft” (Girar Izquierda), “Stop” (Detenerse). Tal como se evidencia a continuación:

```
#include "Motores.h"
void TRight(){
    I &= ~(1<<I1) & ~(1<<I2) & ~(1<<I3);
    I|=(1<<I4);
}
//////////////////////////////////////////////////
void TLeft(){
    I &= ~(1<<I1) & ~(1<<I3) & ~(1<<I4);
    I|=(1<<I2);
}
//////////////////////////////////////////////////
void TForwards(){
    I &= ~(1<<I1) & ~(1<<I3);
    I|=(1<<I2)|(1<<I4);
}
//////////////////////////////////////////////////
void TBackwards(){
    I &= ~(1<<I2) & ~(1<<I4);
    I|=(1<<I1)|(1<<I3);
}
//////////////////////////////////////////////////
void stop(){
    I &= 0xf0;
}
```

Fig 4. Funciones para la activación de los Motores

De acuerdo con lo dicho anteriormente y a lo que se evidencia en la figura 4, se plantearon las 5 funciones, las cuales se controlan por el puerto D del ATMEGA, que en el código está definido por la I. Ahora bien, teniendo en cuenta lo dicho, estas funciones hacen lo siguiente (Ver la sección de Pruebas dentro de los Anexos):

- Para la función TRighth() hacemos que I1 = 0 e I2 = 0, y que I3 = 0 e I4 = 1. De esta manera, usando la tabla

hace que la rueda izquierda que está en el MOTORA no gire, y que solo la rueda derecha del MOTORB gire hacia adelante, creando así una trayectoria de giro, moviéndose hacia la derecha.

- Para la función TLeft() hacemos que I3 = 0 e I4 = 0, y que I1 = 0 e I2 = 1. De esta manera, usando la tabla hace que la rueda izquierda que está en el MOTORA no gire, y que solo la rueda derecha del MOTORB gire hacia adelante, creando así una trayectoria de giro, moviéndose hacia la derecha.
- Para la función TForwards() hacemos que I1 = 0 e I2 = 1, y que I3 = 0 e I4 = 1. De esta manera, usando la tabla hace que la rueda izquierda que está en el MOTORA gire, y que también la rueda derecha del MOTORB gire hacia adelante, realizando así una trayectoria recta.
- Para la función TBackwards() hacemos que I1 = 1 e I2 = 0, y que I3 = 1 e I4 = 0. De esta manera, usando la tabla hace que la rueda izquierda que está en el MOTORA gire hacia atrás, y que también la rueda derecha del MOTORB gire hacia atrás, permitiendo así que el robot pueda retroceder.
- Para la función stop() hacemos que I1 = 0 e I2 = 0, y que I3 = 0 e I4 = 0. De esta manera, hace que el MOTORA y el MOTORB se detengan.

B. SISTEMA DE DETECCIÓN DE OBSTACULO

El módulo encargado de la detección de los obstáculos basa su funcionamiento en base de lo que los sensores infrarrojos detecten. Para esta detección se disponen de 2 sensores infrarrojos, un sensor se encarga de la detección de los objetos que están al frente y el otro de los objetos que estarán en la parte lateral del robot y en función a esta información detectada se comunica en qué caso se está:

- Si el objeto es detectado al frente y no al lado.
- Si el objeto es detectado al lado y no al frente.
- Si el objeto es detectado tanto al frente como al lado.

```
int main(void)
{
    unsigned char Ofrente = 0x01; //mascara para la detección de un objeto al frente del robot.
    unsigned char Olateral = 0x02; //mascara para la detección de un objeto al lado del robot.

    DDRC &= ~(1<<0) & ~(1<<1); // Configura el puerto C0 y C1 como entradas
    DDRC |= (1<<0) | (1<<1); // Configura el puerto B0 y B1 como salidas

    while (1)
    {
        if( ObjetoAlFrente(Ofrente) && ObjetoAlLado(Olateral) ){
            PORTB = (1<<0) | (1<<1);
        }
        else if ( ObjetoAlFrente(Ofrente) && !ObjetoAlLado(Olateral) ){
            PORTB |= (1<<0);
            PORTB &= ~(1<<1);
        }
        else if ( !ObjetoAlFrente(Ofrente) && ObjetoAlLado(Olateral) ){
            PORTB &= ~(1<<0);
            PORTB |= (1<<1);
        }
        else{
            PORTB &= ~(1<<0) & ~(1<<1);
        }
    }
    return 1;
}
```

Fig 5. Código main usando el módulo de detección obstáculos.

Lo que se ha programado son diferentes filtros para la información que brinden los dos sensores, de esta manera se obtiene la información de qué tipo de detección se tiene al momento y se comunica por medio de los pines del puerto B qué tipo de detección según la información que entregan los sensores a los pines del puerto C que fueron configurados para recibir la información de la detección.

La implementación de las funciones usadas en el main son las siguientes.

```
int ObjetoAlFrente(unsigned char ofrente)
{
    if (PINC & ofrente)
        return 0;
    else
        return 1;
}

////////////////////////////////////
int ObjetoAlLado(unsigned char olado)
{
    if (PINC & olado)
        return 0;
    else
        return 1;
}
```

Fig 6. Implementación de las funciones usadas en el main.

La comprobación de la información que envían los sensores se hace por medio de los pines dispuestos para esto, en este caso pines usados del puerto C, junto a la máscara que se le entrega a la función en el main, entonces si hay detección retorna un 1, de lo contrario retorna un 0 que significa que no hay detección, esto funciona de esta manera pues los sensores infrarrojos envían 1 lógico al microcontrolador cuando no hay detección entonces lo que se hace es retornar un cero cuando se tiene la comprobación de un 1 en la entrada del pin correspondiente de recibir esta información junto con la máscara. (Ver la sección de Pruebas dentro de los Anexos).

C. SISTEMA DE DETECCIÓN DE LINEA

Para creación del modulo de detección de línea, es necesario saber cuál es la salida del pin digital D0 del TCRT5000 frente a una superficie blanca o negra. De esta manera, al hacer las pruebas se obtiene que:

	Salida D0
Superficie blanca	0
Superficie negra	1

Tabla 2.Comportamiento del TCRT5000.

Como se observa en la Tabla 2, el sensor TCRT5000, para una superficie blanca en el pin D0 se encuentra un 0 lógico. En cambio, para una superficie negra se encuentra que en la salida es un 1 lógico. De esta manera, teniendo en cuenta esto se programaron los siguientes módulos:

```
#include <avr/io.h>
#include "Modulo_DeteccionLinea.h"

int DeteccionLineaDER(unsigned char LineaDER)
{
    if (PINC & LineaDER) return 1;
    else return 0;
}

///////////////////////////////////////////////////

int DeteccionLineaIZQ(unsigned char LineaIZQ)
{
    if (PINC & LineaIZQ) return 1;
    else return 0;
}
```

Fig 7. Módulos para la detección de Línea

Como se observa en la Figura 7, la estrategia planteada es crear dos funciones que sirven para detectar la línea negra, a través del sensor derecho e izquierdo, creando así dos funciones para su detección. Por otra parte, dentro de estas funciones se evidencia que cuando detecta la línea negra, el sistema retorna un 1. En cambio, para cuando detecta la superficie blanca, el programa retorna un 0.

Ahora bien, para la comprobación de que las funciones cumplen con su labor, se diseñó el siguiente código main.c.

```
#define F_CPU 8000000U
#include <avr/io.h>
#include <util/delay.h>
#include "Modulo_DeteccionLinea.h"

int main(void)
{
    unsigned char LineaDerecha = 0x04; //mascara para la detección de la línea por el sensor Derecho del robot.
    unsigned char LineaIzquierda = 0x08; //mascara para la detección de la línea por el sensor Izquierdo del robot.

    DDRC &= ~(1<<2) & ~(1<<3); // Configura el puerto C2 y C3 como entradas
    DDHB |= (1<<2) | (1<<3); // Configura el puerto B2 y B3 como salidas

    while (1)
    {
        if( DeteccionLineaDER(LineaDerecha) && DeteccionLineaIZQ(LineaIzquierda) ){
            PORTB = (1<<2) | (1<<3);
        }
        else if ( DeteccionLineaDER(LineaDerecha) && !DeteccionLineaIZQ(LineaIzquierda) ){
            PORTB |= (1<<2);
            PORTB &= ~(1<<3);
        }
        else if ( !DeteccionLineaDER(LineaDerecha) && DeteccionLineaIZQ(LineaIzquierda) ){
            PORTB &= ~(1<<2);
            PORTB |= (1<<3);
        }
        else{
            PORTB &= ~(1<<2) & ~(1<<3);
        }
    }
    return 1;
}
```

Fig 8. Código main usando el módulo de detección de línea

Como se observaron en las Figuras 7 y 8, los sensores están conectados en el puerto C, específicamente el sensor derecho al pin C2 y el sensor izquierdo al pin C3, configurándose como entradas dentro del ATMEGA328P. Por otro parte, se destaca que se usan 2 LEDs que se usan en forma de prueba, los cuales son conectados al puerto B, en donde el LED Derecho esta en el pin B2 y el LED Izquierdo en el pin B3, configurándose así estos pines como salida.

Ahora bien, en la Figura 8 se plantean 4 situaciones (Ver la sección de Pruebas dentro de los Anexos):

- Cuando el sensor derecho e izquierdo detectan la línea negra, se encienden ambos LEDs.
- Cuando el sensor derecho detecta la línea negra y el sensor izquierdo no, se enciende solo el LED Derecho.
- Cuando el sensor derecho no detecta la línea negra y el sensor izquierdo si la detecta, se enciende solo el LED Izquierdo.
- Cuando el sensor esta sobre una superficie blanca, no se enciende ningún LED.

IV. CONCLUSIÓN

Durante el desarrollo de la fase 2 se aprendió y se destaca que:

- Es importante hacer de una división correcta de todos los módulos que compondrán la solución o funcionamiento del proyecto pues de lo contrario al momento de trabajar en cada módulo se pueden presentar confusiones a la hora de desarrollar el código en específico del módulo como también para que sirva al momento de formar parte del todo o código completo. Además, se debe también comprender muy bien el funcionamiento de los dispositivos con los que el módulo está involucrado, porque de lo contrario no se tendrá la operación correcta tanto a nivel modular como global.

- El sistema de movimiento funciona mediante la activación de pines, de tal manera que activa y desactiva los pines indicados para la configuración del movimiento de los motores. De tal manera que:

- Para crear una trayectoria recta los dos motores deben estar activas.
- Para crear una trayectoria de retroceso los dos motores deben estar girando en sentido contrario.
- Para crear una trayectoria de giro hacia la derecha, el motor izquierdo debe estar desactivado y el Motor de la derecha debe estar activo.
- Para crear una trayectoria de giro hacia la izquierda, el motor derecho se desactiva y el motor izquierdo es el que debe estar activo.

- El sensor FC-51 sirve para verificar si hay obstáculos en el camino, de tal manera que cuando no hay un obstáculo, en la salida del pin D0 es un 1 lógico. En cambio, cuando hay un obstáculo en la salida es un 0 lógico. De esta manera, para hacer una función de detección de obstáculos hay que programar un Negador, es decir, para cuando no haya obstáculos retorne un 0, y cuando hay un obstáculo retorne un 1 lógico.

- El sensor TCRT5000 sirve para detectar si esta sobre una superficie blanca o negra, de tal manera que cuando el sensor esta sobre una superficie blanca en el pin D0 se encuentra un 0 lógico. En cambio, para una superficie negra se encuentra que en la salida es un 1 lógico. Siendo así, una situación favorable ya que si se desea hacer una función detección de superficies de color negro, basta tener la lógica de un Buffer, de tal manera cuando esta sobre una superficie negra retorne un 1, y para una superficie blanca retorne un 0 lógico.

ANEXOS

 Universidad del Valle	Universidad del Valle Robot móvil para recorrer un camino con obstáculos	Rev.: 000
Título: ESPECIFICACIÓN DE REQUERIMIENTOS FUNCIONALES		Documento : ERF-001
		Página : 1 de 1

REVISIÓN HISTÓRICA			
Rev.	Descripción del Cambio	Autor	Fecha
001	Construcción del documento	Carlos Fernando Quintero, Jhonier Andrés Vargas, Joan Velasco	2021-11-21
002	Actualización	Carlos Fernando Quintero, Joan Velasco, Jhonier Andrés Vargas.	2022-02-04

Ref #	Funciones	Categoría
1.0	El robot móvil debe ser capaz de detectar los diferentes obstáculos en el camino, usando dos sensores infrarrojos con referencia FC-51	E
2.0	El robot móvil debe ser capaz de detectar la línea negra, usando 2 sensores ópticos infrarrojos TCRT5000	E
3.0	El robot móvil debe ser capaz de controlar el sentido de giro de cada uno de los motores por medio del módulo L298N y el microcontrolador ATmega328p.	E
4.0	El robot debe ser capaz de mantenerse dentro de la trayectoria y detenerse una vez llega a un final de carrera señalado en la pista.	E
5.0	El robot debe ser capaz de evadir los obstáculos rodeándolos.	E

Anexo 1. Requerimientos Funcionales del Robot móvil para recorrer un camino con obstáculos

 Universidad del Valle	<div>Universidad del Valle</div> <div>Robot móvil para recorrer un camino con obstáculos</div>	Rev.: 000
Título: ESPECIFICACIÓN DE REQUERIMIENTOS NO FUNCIONALES		Documento : ERF-001
		Página : 1 de 1

REVISIÓN HISTÓRICA			
Rev.	Descripción del Cambio	Autor	Fecha
001	Construcción del documento	Carlos Fernando Quintero, Jhonier Andrés Vargas, Joan Velasco	2021-11-21
002	Actualización	Carlos Fernando Quintero, Joan Velasco, Jhonier Andrés Vargas.	2022-02-04

Ref #	Descripción	Categoría
1.0	Para la realización de este proyecto se usará el Atmel Studio 7.0	E
2.0	Se usará un microcontrolador Atmega328 a través del Hardware Arduino	E
3.0	Para la realización del esquema físico y conexiones se usará Proteus 8.8	O

Anexo 2. Requerimientos No Funcionales del Robot móvil para recorrer un camino con obstáculos

 Universidad del Valle	Universidad del Valle Robot móvil para recorrer un camino con obstáculos		Rev.: 002
Título: CASO DE USO Requerimiento funcional 1. El robot móvil debe ser capaz de detectar los diferentes obstáculos en el camino, usando dos sensores infrarrojos (uno enfrente y otro a un lado) con referencia FC-51		Documento : CUR-001	Página : 1 de 1

REVISIÓN HISTÓRICA			
Rev.	Descripción del Cambio	Autor	Fecha
001	Construcción de caso uso	Carlos Fernando Quintero, Jhonier Andrés Vargas, Joan Velasco	2021-11-21
002	Actualización	Carlos Fernando Quintero, Joan Velasco, Jhonier Andrés Vargas.	2022-02-04

INFORMACIÓN GENERAL	
Actores:	Robot (HW) y el firmware (SW)
Propósito:	El robot debe ser capaz de detectar los obstáculos para evitar choques, a través de dos sensores infrarrojos (uno en frente y otro a un lado) con referencia FC-51
Resumen:	Cuando el robot móvil detecte un obstáculo, por medio de los sensores de infrarrojos, se enviará las señales al microcontrolador para que la procese según sea el caso de detección, es decir, si se detectó objeto al frente, al lado o en ambos lados.
Tipo:	Real

Curso Normal de los Eventos	
Acción del Robot	Respuesta del Firmware
	1. Configurar los puertos de entrada (DDRC) y salida (DDRB) para configurar los sensores infrarrojos, tanto para la detección del obstáculo, como de la comunicación del tipo de detección (objeto al frente, al lado o ambos lados).
2. Los dos sensores infrarrojos envían señales eléctricas al puerto DDRC de entrada según la detección hecha.	
	3. Se verifica por medio de 3 filtros qué tipo de detección es por medio de la lectura del puerto C a través de PINC (objeto al frente, al lado o ambos lados).
	4. Dependiendo del tipo de detección (objeto al frente, al lado o ambos lados) se ponen en 1 o 0 los pines del puerto DDRB encargado para la comunicación del tipo de detección (objeto al frente, al lado o ambos lados).ssS

Anexo 3.1. Caso de uso para el Requerimiento funcional 1

 Universidad del Valle	Universidad del Valle Robot móvil para recorrer un camino con obstáculos		Rev.: 000
Título: CASO DE USO Requerimiento funcional 2. El robot móvil debe ser capaz de detectar la línea negra, usando 2 sensores ópticos infrarrojos TCRT5000		Documento : CUR-001	Página : 1 de 1

REVISIÓN HISTÓRICA			
Rev.	Descripción del Cambio	Autor	Fecha
001	Construcción de caso uso	Carlos Fernando Quintero, Jhonier Andrés Vargas, Joan Velasco	2021-11-21
002	Actualización	Carlos Fernando Quintero, Joan Velasco, Jhonier Andrés Vargas.	2022-02-04

INFORMACIÓN GENERAL	
Actores:	Robot, firmware.
Propósito:	El robot móvil debe ser capaz de detectar la línea negra, usando 2 sensores ópticos infrarrojos TCRT5000
Resumen:	Lo que se busca es que el robot móvil sea capaz de reconocer la trayectoria a seguir, la cual es una línea negra por medio de 2 sensores ópticos infrarrojos.
Tipo:	Real

Curso Normal de los Eventos	
Acción del robot	Respuesta del firmware
	1. Configurar los puertos de entrada y salida para configurar los dos sensores ópticos infrarrojos, con el comando DDRC.
2. Los dos receptores infrarrojos envían una señal eléctrica al puerto de entrada y salida.	
	3. Se realiza una lectura del puerto C, a través del comando PINC.
	4. Se compara si la mascara correspondiente a cada sensor si es igual a la lectura del PINC
	5. Retorna un 1, si la mascara es igual al PINC, significando que el sensor paso sobre la línea negra. En caso contrario, si la mascara es diferente a la lectura del PINC retorna un 0.

Anexo 3.2. Caso de uso para el Requerimiento funcional 2

 Universidad del Valle	Universidad del Valle Robot móvil para recorrer un camino con obstáculos		Rev.: 000
Título: CASO DE USO Requerimiento funcional 3 El robot móvil debe ser capaz de controlar el sentido de giro de cada uno de los motores por medio del módulo L298N y el microcontrolador ATmega328p.		Documento : CUR-001	Página : 1 de 1

REVISIÓN HISTÓRICA			
Rev.	Descripción del Cambio	Autor	Fecha
001	Construcción de caso uso	Carlos Fernando Quintero, Jhonier Andrés Vargas, Joan Velasco	2021-11-21
002	Actualización	Carlos Fernando Quintero, Joan Velasco, Jhonier Andrés Vargas.	2022-02-04

INFORMACIÓN GENERAL	
Actores:	Robot, firmware.
Propósito:	El robot debe ser capaz de controlar el sentido de giro de los motores para moverse.
Resumen:	Se espera que se pueda cambiar el sentido de giro de cada una de las ruedas para poder seguir la trayectoria marcada o esquivar el obstáculo, según el robot lo requiera.
Tipo:	Real

Curso Normal de los Eventos
Respuesta del firmware
1. Se configuran los puertos de salida (Puerto D) para la comunicación unidireccional entre el microcontrolador ATmega328p y el puente H L298N.
2. Se hace el llamado de la dirección a la que se requiere avanzar (Parar, Avanzar, Derecha, Izquierda y Retroceder).

Si el valor de la variable dirección es Parar, entonces:

Curso Normal de los Eventos
Respuesta del firmware
3. Se escribe en el registro del puerto de salida (PORTD) y en los bits menos significativos el valor de cero, para desactivar todos los pines I1, I2, I3, e I4 del L298N.

Si el valor de la variable dirección es Avanzar, entonces:

Curso Normal de los Eventos
Respuesta del firmware
4. Se escribe en el registro del puerto de salida (PORTD) y en los bits menos significativos el valor de 0xA, para activar los pines I2 e I4 del L298N

Si el valor de la variable dirección es Derecha, entonces:

Curso Normal de los Eventos
Respuesta del firmware
5. Se escribe en el registro del puerto de salida (PORTD) y en los bits menos significativos el valor de 0x8, para activar el pin I4 del L298N.

Si el valor de la variable dirección es izquierda, entonces:

Curso Normal de los Eventos
Respuesta del firmware
6. Se escribe en el registro del puerto de salida (PORTD) y en los bits menos significativos el valor de 0x2, para activar el pin I2 del L298N

Si el valor de la variable dirección es Retroceder, entonces:

Curso Normal de los Eventos
Respuesta del firmware
7. Se escribe en el registro del puerto de salida (PORTD) y en los bits menos significativos el alor de 0x5, para activar los pines I1 e I3 del L298N

Anexo 3.3. Caso de uso para el Requerimiento funcional 3

 Universidad del Valle	<div> <div>Universidad del Valle</div> <div>Robot móvil para recorrer un camino con obstáculos</div> </div>	Rev.: 000
Título: CASO DE USO Requerimiento funcional 4. El robot debe ser capaz de mantenerse dentro de la trayectoria y detenerse una vez llega a un final de carrera señalado en la pista.		Documento : CUR-001 Página : 1 de 1

REVISIÓN HISTÓRICA			
Rev.	Descripción del Cambio	Autor	Fecha
001	Construcción de caso uso	Carlos Fernando Quintero, Jhonier Andrés Vargas, Joan Velasco	2021-11-21

INFORMACIÓN GENERAL	
Actores:	firmware.
Propósito:	El robot debe ser capaz de seguir la línea negra detectada y avanzar
Resumen:	Lo que se busca es que el robot móvil sea capaz de reconocer la trayectoria a seguir, la cual es una línea negra por medio de 2 sensores ópticos infrarrojos.
Tipo:	Real

Curso Normal de los Eventos
Respuesta del firmware
1. Se lee el valor de las variables de software que almacenan el estado de los sensores izquierdo y derecho: 00, 01, 10, 11.

Si el valor de la variable es 00, entonces:

Curso Normal de los Eventos
Respuesta del firmware
1. Se cambia el valor de la variable dirección a "Parar"

Si el valor de la variable es 01, entonces:

Curso Normal de los Eventos
Respuesta del firmware
1. Se cambia el valor de la variable dirección a "Izquierda"

Si el valor de la variable es 10, entonces:

Curso Normal de los Eventos
Respuesta del firmware
1. Se cambia el valor de la variable dirección a "Derecha"

Si el valor de la variable es 11, entonces:

Curso Normal de los Eventos
Respuesta del firmware
1. Se cambia el valor de la variable dirección a "Avanzar"

 Universidad del Valle	Universidad del Valle Robot móvil para recorrer un camino con obstáculos	Rev.: 000
Título: CASO DE USO Requerimiento funcional 5 El robot móvil debe ser capaz de evadir los obstáculos rodeandolos		Documento : CUR-001 Página : 1 de 1

REVISIÓN HISTÓRICA			
Rev.	Descripción del Cambio	Autor	Fecha
001	Construcción de caso uso	Carlos Fernando Quintero, Jhonier Andrés Vargas, Joan Velasco	2021-11-21

INFORMACIÓN GENERAL	
Actores:	firmware.
Propósito:	El robot debe girar y avanzar para rodear un obstáculo que detecte
Resumen:	Se espera que a partir de los cambios que puedan existir en la trayectoria marcada y los obstáculos el robot cambie la velocidad en cada rueda para girar.
Tipo:	Real

Curso Normal de los Eventos
Respuesta del firmware
1. Se lee el valor de las variables de software que almacén la presencia del obstáculo (Enfrente, A_un_lado, Enfrente_y_A_un_Lado, Ninguno).
2. Se define e inicializa una variable Estado_robot en "0" (dentro de la línea).

Nota: Los conceptos "Enfrente", "A un lado", "Enfrente y a un lado" y "Ninguno" se refieren:

- Enfrente: Significa que hay presencia de un objeto detectado por el sensor frontal del robot.
- A un lado: Significa que hay presencia de un objeto detectado por solo el sensor lateral del robot.
- Enfrente y a un lado: Significa que hay presencia de un objeto detectado por el sensor frontal y lateral del robot.
- Ninguno: Significa que el sensor frontal y lateral no detectan ningún objeto.

Para cuando Estado_robot= 0

Si la variable de presencia de obstáculo es "Enfrente", entonces:

Curso Normal de los Eventos
Respuesta del firmware
1. Se cambia el valor de la variable dirección a "Izquierda"

Si la variable de presencia de obstáculo es "A_un_lado", entonces:

Curso Normal de los Eventos
Respuesta del firmware
1. Se cambia el valor de la variable Estado_robot a "1"

Para cuando Estado_robot=1

Si la variable de presencia de obstáculo es "Enfrente", entonces:

Curso Normal de los Eventos
Respuesta del firmware
1. Se cambia el valor de la variable dirección a "derecha"

Si la variable de presencia de obstáculo es "A_un_lado", entonces:

Curso Normal de los Eventos
Respuesta del firmware
1. Se cambia el valor de la variable dirección a "Avanzar"

Si la variable de presencia de obstáculo es "Enfrente_y_A_un_lado", entonces:

Curso Normal de los Eventos
Respuesta del firmware
1. Se cambia el valor de la variable dirección a "Izquierda"

la variable de presencia de obstáculo es "Ninguno" y la variable estado_robot es 1, entonces:

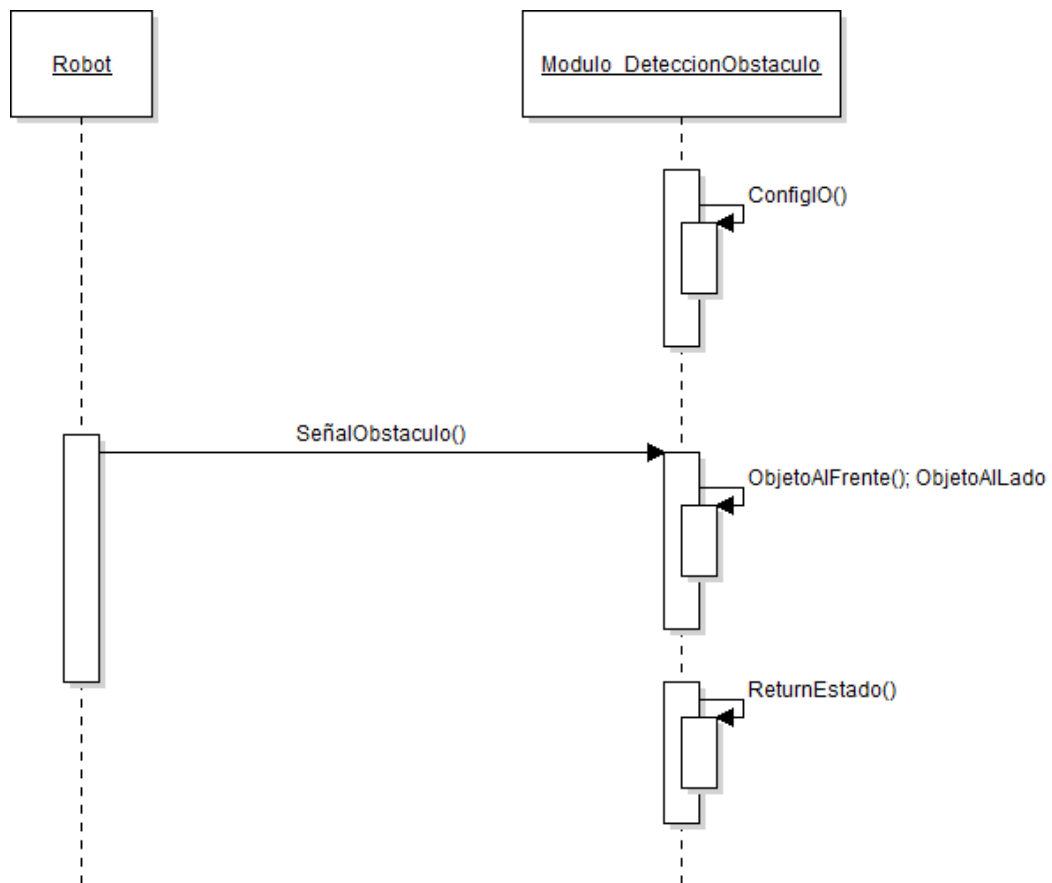
Curso Normal de los Eventos
Respuesta del firmware
1. Se cambia el valor de la variable dirección a "derecha"

Curso Normal de los Eventos
Respuesta del firmware
1. Se lee el valor de la variable de software que almacena el estado de los sensores infrarrojos.

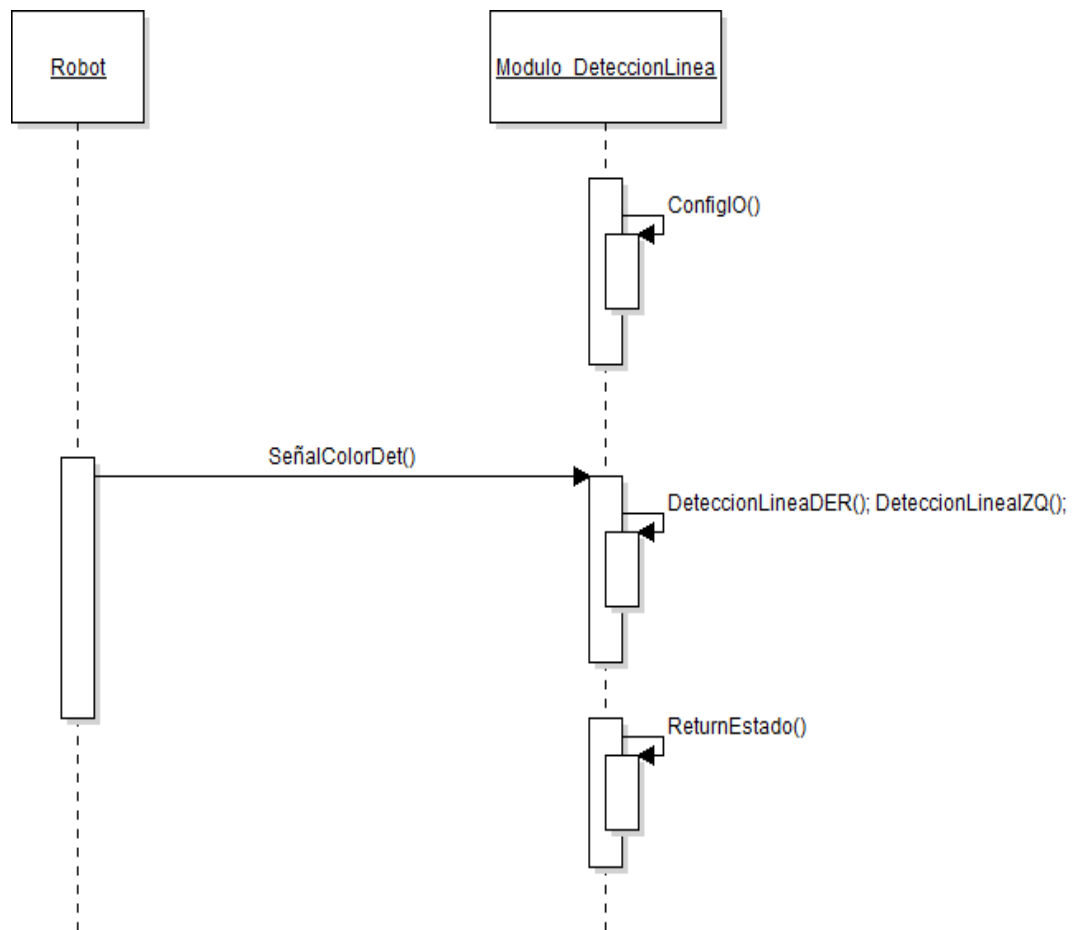
Si la variable de estado de los sensores es "00", "01" o "10", entonces:

Curso Normal de los Eventos
Respuesta del firmware
1. Se cambia el valor de la variable dirección a "Izquierda"
2. Se cambia el valor de la variable estado_robot a "0"

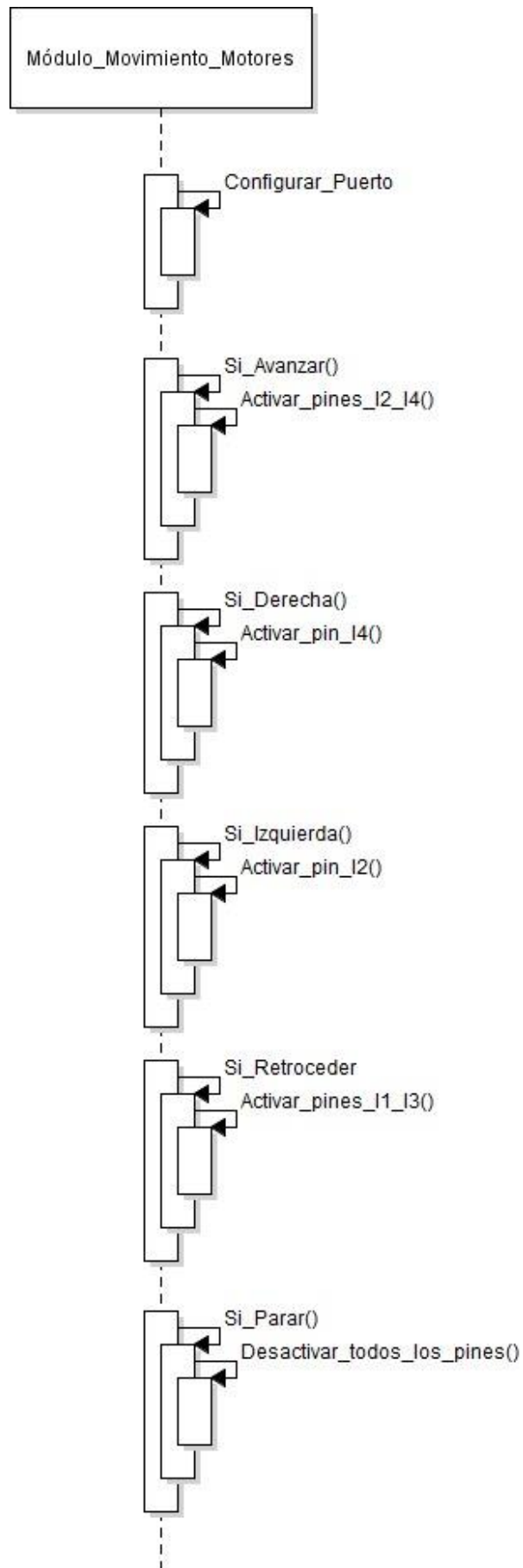
Anexo 3.5. Caso de uso para el Requerimiento funcional 5



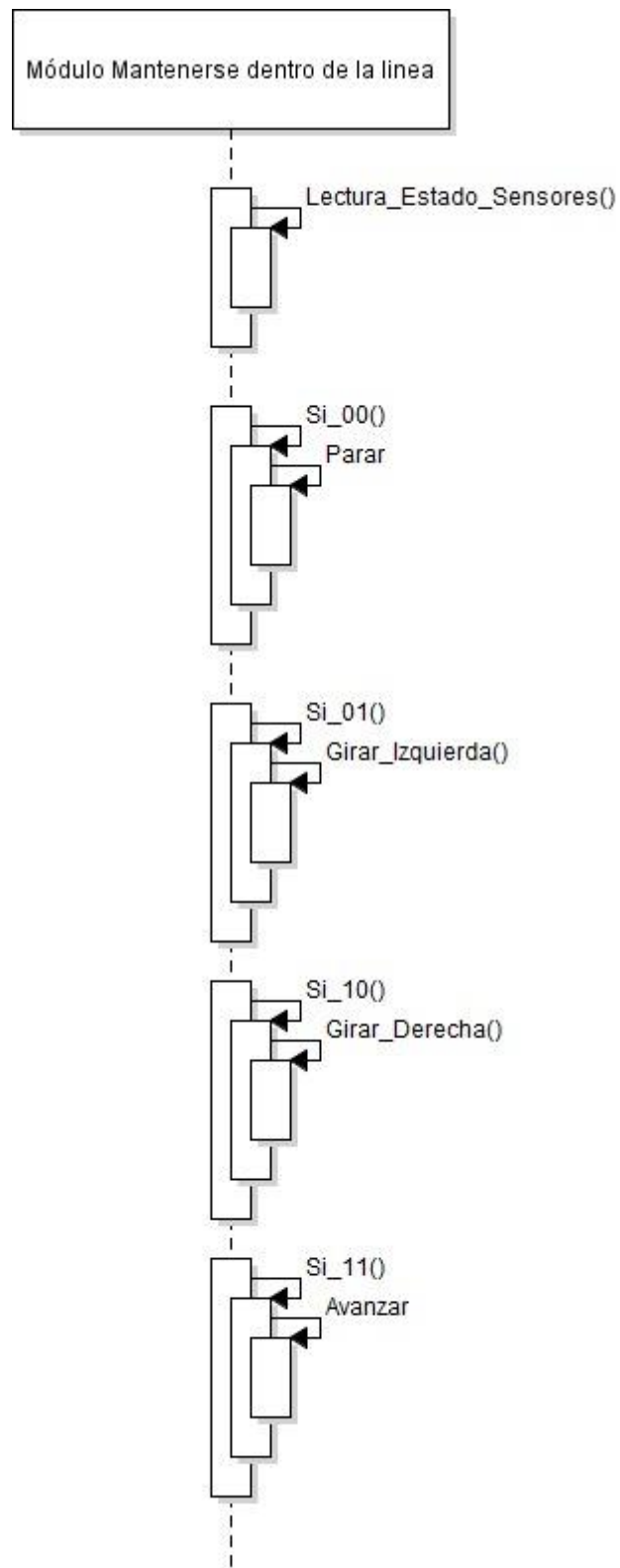
Anexo 4.1. Diagrama secuencial para el Requerimiento funcional 1



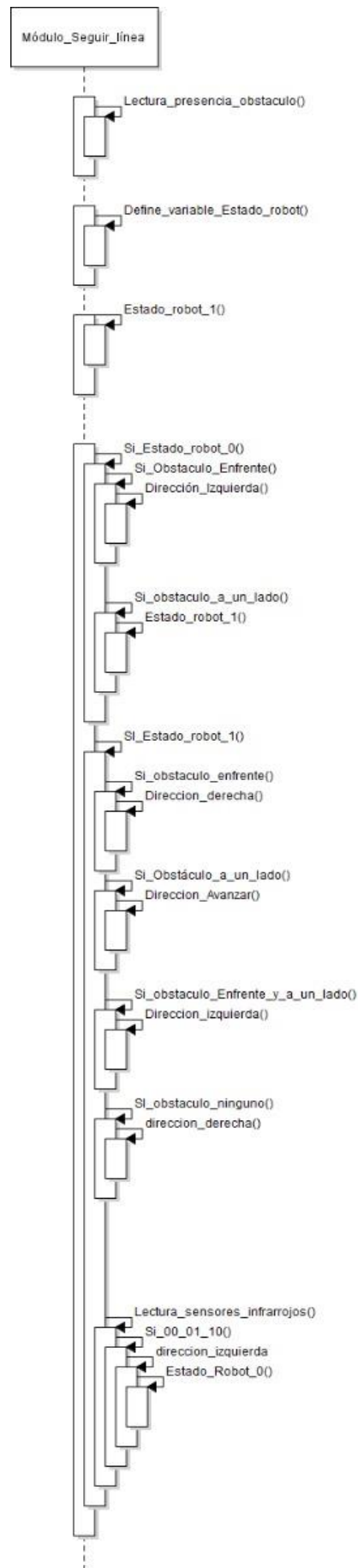
Anexo 4.2. Diagrama secuencial para el Requerimiento funcional 2



Anexo 4.3. Diagrama secuencial para el Requerimiento funcional 3



Anexo 4.4. Diagrama secuencial para el Requerimiento funcional 4



Anexo 4.5. Diagrama secuencial para el Requerimiento funcional 5

ENLACE DE VIDEOS Y PRUEBAS DE LOS MODULOS

- Sistema de movimiento (Trayectoria recta):

https://drive.google.com/file/d/1Y9M8XXO_DqB0jsYb5wePPXT5H_-FIviV/view?usp=sharing

- Sistema de detección de obstáculo:

<https://drive.google.com/file/d/1Pw8Y3IypGXolfZwoPmqVFo4OdSZIKlqX/view?usp=sharing>

- Sistema de detección de Línea:

https://drive.google.com/file/d/1AI1ujGjSOLtysyRygbGTpCd6PaWBYc_6/view?usp=sharing