

# Fase 3-Robot móvil para desinfectar ambientes interiores con luces UV

Duque R. Santiago, López F. Juan José, y Leiton A. Jorge Luis, *Facultad de Ingeniería, EIEE, Universidad del Valle.*

**Abstract**—En este documento se presentan los sistemas de movimiento, percepción, limpieza y comunicación, para el robot móvil, implementados y probados utilizando el microcontrolador. También se presenta el diseño y funcionamiento de la plataforma del robot mecánico. Todo esto, utilizando la metodología de desarrollo de software RUP.

## I. INTRODUCCIÓN

Han pasado 9 meses de pandemia, y al lado de tantas preocupaciones referente a evitar el contagio del Covid-19, también hay una creciente pregunta de cómo mantenerse seguro, limpio y aseado. Debido a esta situación actual, se hace importante desarrollar plataformas móviles que ayuden a limpiar los espacios donde se frecuenta la mayor parte del tiempo, como por ejemplo los hogares. Por esa razón, con el desarrollo de este proyecto se busca encontrar una estrategia de limpieza a espacios interiores, utilizando luces ultravioleta.

## II. PLANTEAMIENTO Y DESCRIPCIÓN DEL PROBLEMA.

Se desea desarrollar un robot móvil para desinfectar un ambiente interior utilizando luces Led UV.

El robot móvil debe ser capaz de: navegar de forma autónoma en el entorno, seguir cerca las paredes del ambiente, evitar los obstáculos y mientras tanto desinfectar el interior utilizando luces ultravioleta. Además, el funcionamiento del robot móvil se configurará mediante un aplicación móvil.

## III. SISTEMA DE MOVIMIENTO

Para poder implementar el sistema de movimiento se tienen que configurar los registros del microcontrolador ATMEGA328p (ver Fig. 1), y también se deben tener en cuenta las características de los 2 motores (Motorreductor 3v-6v) como se observa en la Fig. 2, y el puente H TB6612FNG, ver Fig. 3.

Para mover el robot, se debe variar la velocidad de los dos motores del robot, es decir se debe avanzar o retroceder cada motor, por medio de los PWM, de esta manera se realizan los movimientos hacia adelante, atrás, derecha e izquierda.

Santiago Duque Ramos es estudiante del programa académico de Ingeniería Electrónica en la Universidad del Valle, Valle del Cauca, Cali, Colombia, Código: 1733527, e-mail: duque.santiago@correounivalle.edu.co

Juan José López Flórez es estudiante del programa académico de Ingeniería Electrónica en la Universidad del Valle, Valle del Cauca, Cali, Colombia, Código: 1730514, e-mail: juan.jose.lopez@correounivalle.edu.co

Jorge Luis Leiton Arias es estudiante del programa académico de Ingeniería Electrónica en la Universidad del Valle, Valle del Cauca, Cali, Colombia, Código: 1730004, e-mail: jorge.leiton@correounivalle.edu.co

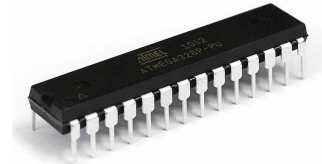


Fig. 1. Microcontrolador ATMEGA328p.



Fig. 2. Motorreductor 3v-6v.

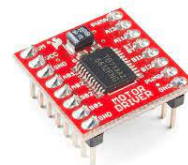


Fig. 3. Puente H TB6612FNG.

Para realizar en el ATMEGA328p lo dicho anteriormente, se tiene en cuenta que, se debe usar el control de velocidad por medio del PWM del microcontrolador; además, es importante saber que la frecuencia máxima que soporta el puente H es de 100 kHz, por lo tanto, la salida PWM no puede sobrepasar ese límite.

Para realizar una correcta configuración, se revisó el datasheet del microcontrolador ATMEGA328p [1].

Para ser más precisos, para configurar el microcontrolador, se debe usar el Timer 0, en modo Fast PWM no inversor, esto se configuró con el registro TCCR0A (Ver Fig. 4), utilizando los bits que aparecen la Fig. 6.

Con el registro TCCR0B (Ver Fig. 7), se configuró el prescaler de 8, utilizando los bits que aparecen la Fig. 8.

Bit	7	6	5	4	3	2	1	0
0x24 (0x44)	COM0A1	COM0A0	COM0B1	COM0B0	—	—	WGM01	WGM00
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Fig. 4. Registro TCCR0A.

Se puede observar, con la ecuación de la Fig. 5 que la salida para el PWM es de 7.8kHz por lo que está muy por de bajo del limite del puente H.

The PWM frequency for the output can be calculated by the following equation:

$$f_{\text{OchxPWM}} = \frac{f_{\text{clk I/O}}}{N \times 256}$$

Fig. 5. Formula para la frecuencia de salida del PWM.

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on <sup>(1)(2)</sup>
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, phase correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	—	—	—
5	1	0	1	PWM, phase correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	—	—	—
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Fig. 6. Modo de Operación.

Bit	7	6	5	4	3	2	1	0
0x25 (0x45)	FOC0A	FOC0B	—	—	WGM02	CS02	CS01	CS00
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Fig. 7. Registro TCCR0B.

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	clk <sub>IO</sub> /(no prescaling)
0	1	0	clk <sub>IO</sub> /8 (from prescaler)
0	1	1	clk <sub>IO</sub> /64 (from prescaler)
1	0	0	clk <sub>IO</sub> /256 (from prescaler)
1	0	1	clk <sub>IO</sub> /1024 (from prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

Fig. 8. Selección del Prescaler.

Para simular el comportamiento del módulo PWM, se realizó una implementación del circuito requerido en el software de simulación Proteus. Con el esquema implementado se realizó una pequeña prueba que consiste en aumentar progresivamente el ciclo de trabajo del PWM. Este cambio se observaba en el osciloscopio, mientras el módulo puente H convierte la señal PWM en un mayor voltaje para los motores, aumentando así la velocidad de estos.

A continuación, en las figuras: Fig. 9 y Fig. 10, se presenta una evidencia del funcionamiento del código para el sistema de movimiento.

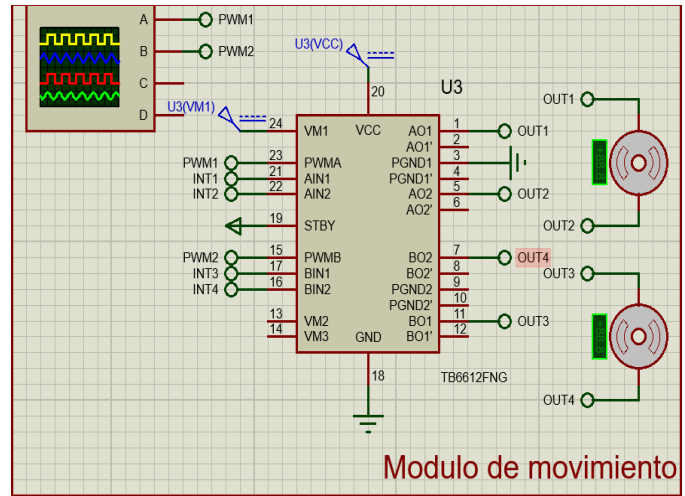


Fig. 9. Esquema del sistema de movimiento implementado en Proteus.

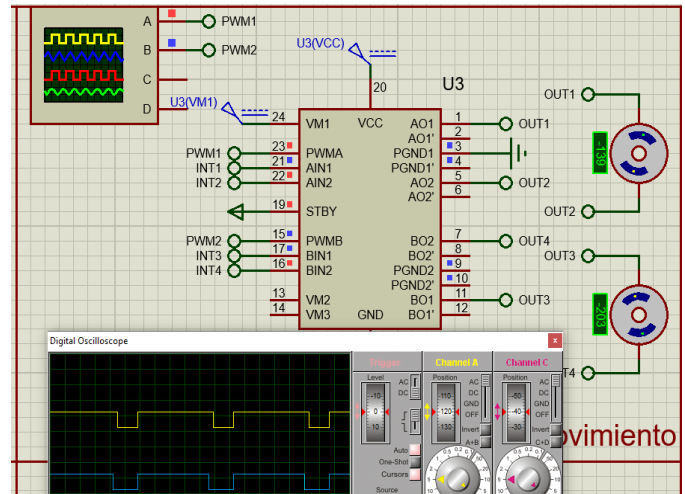


Fig. 10. Simulación del funcionamiento del sistema de movimiento implementado en Proteus.

#### IV. SISTEMA DE PERCEPCIÓN

Para implementar el sistema de percepción se deben configurar los registros que controlan el ADC del microcontrolador del ATMEGA328p, para de esta manera, realizar la lectura de los 3 sensores Sharp GP2Y0A21YK0F. En la Fig. 11. se puede observar uno de los sensores de distancia Sharp GP2Y0A21YK0F.

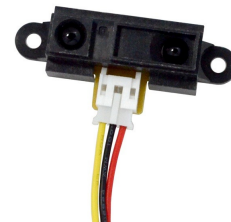


Fig. 11. Sensor de distancia GP2Y0A21YK0F.

Para el ATMEGA328P, con la ayuda del registro ADMUX (Ver Fig. 12), se configuraron los canales a leer (Ver Fig. 13)

y la referencia de voltaje externo (Ver Fig. 14).

Con el registro ADCSRA (Ver Fig. 15), se configuró el prescaler en 128 (Ver Fig. 16).

Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	—	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Fig. 12. Registro ADMUX.

MUX3..0	Single Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5
0110	ADC6
0111	ADC7
1000	ADC8 <sup>(1)</sup>
1001	(reserved)
1010	(reserved)
1011	(reserved)
1100	(reserved)
1101	(reserved)
1110	1.1V (V <sub>DD</sub> )
1111	0V (GND)

Fig. 13. Selección del Canal de ADC.

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, internal V <sub>REF</sub> turned off
0	1	AV <sub>CC</sub> with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 1.1V voltage reference with external capacitor at AREF pin

Fig. 14. Modo de conexión a un voltaje de referencia.

Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Fig. 15. Registro ADCSRA.

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Fig. 16. Prescaler para ADC.

Para simular el comportamiento del sistema de percepción, se realizó una implementación del circuito requerido en el software de simulación Proteus. El diseño consiste en un programa que toma la lectura de un canal de ADC, y después la envía a la terminal serial del Proteus; posteriormente, lee el siguiente canal de ADC, repitiéndose el proceso.

A continuación, en las figuras: Fig. 17 y Fig. 18, se presenta una evidencia del funcionamiento del código para el sistema de percepción. Para ello se muestran los resultados del terminal serial que posee Proteus.

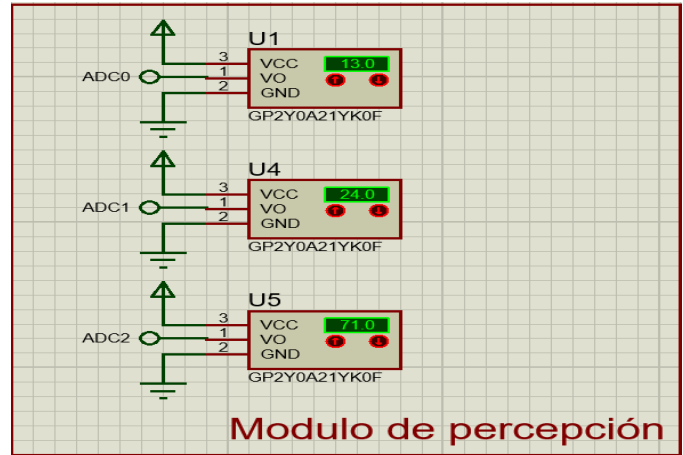


Fig. 17. Esquema del sistema de percepción implementado en Proteus.

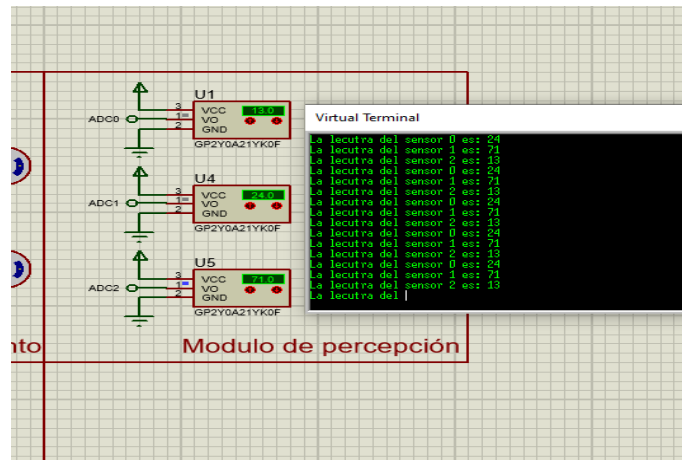


Fig. 18. Simulación del funcionamiento del sistema de percepción implementado en Proteus.

## V. SISTEMA DE COMUNICACIÓN

Para realizar la comunicación, se usa el módulo de comunicación Bluetooth HC-05.

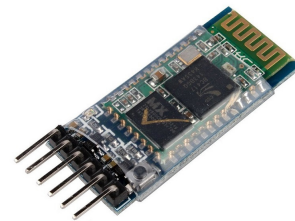


Fig. 19. Módulo HC-05.

Para realizar la comunicación vía Bluetooth, se deben configurar los registros del ATMEGA328P para tal fin. Se procedió a realizar la configuración del USART como UART, teniendo en cuenta, los registros UCSRnA (Fig. 20), UCSRnB (Fig. 21), UCSRnC (Fig. 22), y la información de la Fig. 23.

**UCSRnA – USART Control and Status Register n A**

Bit	7	6	5	4	3	2	1	0	
	RXCn	TXCn	UDREN	FEEn	DORn	UPEn	U2Xn	MPCMn	UCSRnA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

Fig. 20. Registro UCSRnA - USART Control and Status Register n A.

Bit	7	6	5	4	3	2	1	0	
	RXCIEEn	TXCIEEn	UDRIEn	RXENn	TXENn	UCSZn2	RXB8n	TXB8n	UCSRnB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Fig. 21. Registro UCSRnB – USART Control and Status Register n B.

Bit	7	6	5	4	3	2	1	0	
	UMSELn1	UMSELn0	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	UCSRnC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

Fig. 22. Registro UCSRnC – USART Control and Status Register n C.

**Table 19-4. UMSELn Bits Settings**

UMSELn1	UMSELn0	Mode
0	0	Asynchronous USART
0	1	Synchronous USART
1	0	(Reserved)
1	1	Master SPI (MSPIM) <sup>(1)</sup>

Note: 1. See Section 20, "USART in SPI Mode" on page 166 for full description of the master SPI mode (MSPIM) operation

Fig. 23. Configuración UMSELn Bits.

Configurado el UART, se procedió a realizar la configuración para la lectura y escritura en los registros destinados para tal fin, para eso se tuvo en cuenta, el registro UDRn (Fig. 24.)

**Register Description****UDRn – USART I/O Data Register n**

Bit	7	6	5	4	3	2	1	0	
	RXB[7:0]								UDRn (Read)
	TXB[7:0]								UDRn (Write)
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Fig. 24. UDRn – USART I/O Data Register n.

A continuación, en las figuras: Fig. 25 y Fig. 26, se presenta una evidencia del funcionamiento del código para el sistema de comunicación.

Para ello se realizó la implementación del circuito requerido en el Software de simulación Proteus. En la simulación se muestra un mensaje enviado por el microcontrolador hacia la terminal serial, y la terminal serial enviando un mensaje para encender el módulo de limpieza.

Todo este sistema de comunicación, se implantará con la ayuda de una App móvil. A continuación, en la Fig. 27 se presenta la interfaz gráfica beta de la App Móvil, diseñada para la interacción con el Operario. La App móvil se diseñó con ayuda del programa **MIT App Inventor 2** [2].

En la Fig. 28 y 29 se presenta el diagrama de bloques empleado para el diseño beta de la App móvil.

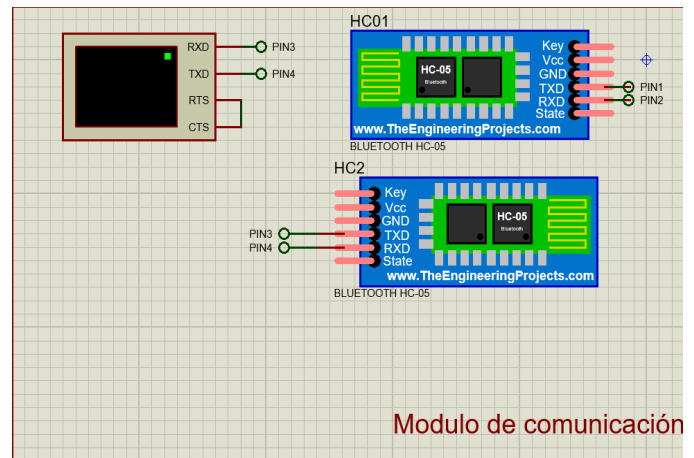


Fig. 25. Esquema del sistema de comunicación implementado en Proteus.

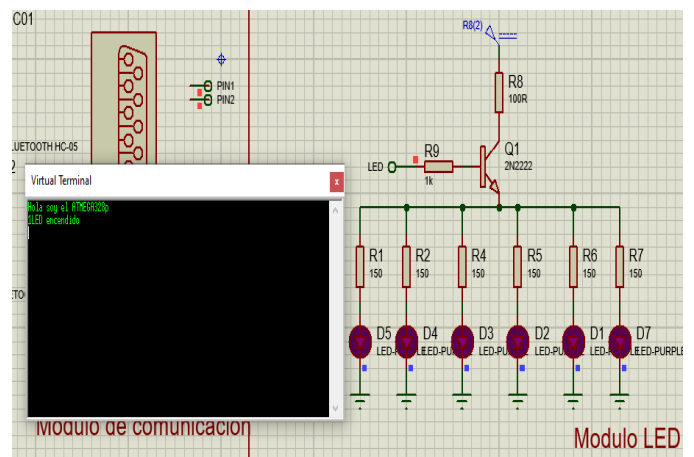


Fig. 26. Simulación del funcionamiento del sistema de comunicación implementado en Proteus.



Fig. 27. Interfaz Gráfica de la App móvil Diseñada.

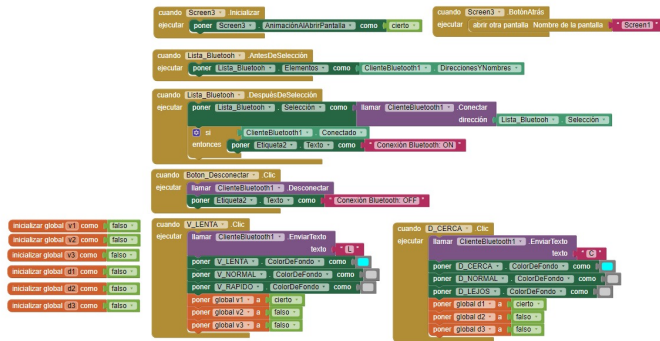


Fig. 28. Diagrama de Bloques del diseño de la App-Móvil.



Fig. 29. Diagrama de Bloques del diseño de la App-Móvil.

## VI. SISTEMA DE LIMPIEZA (LUCES UV)

Para el sistema o módulo de limpieza se utilizó el pin PB4 del ATMEGA328p como salida para controlar el encendido o apagado de los LEDs UV. Debido al alto consumo que puede provocar un arreglo de 6 LEDs, la corriente de estos es suministrada directamente por la fuente de alimentación del robot, y el microcontrolador satura el transistor NPN cuando se desea activa el arreglo de LEDS. Las pruebas del módulo se realizaron junto a las del módulo de comunicación.

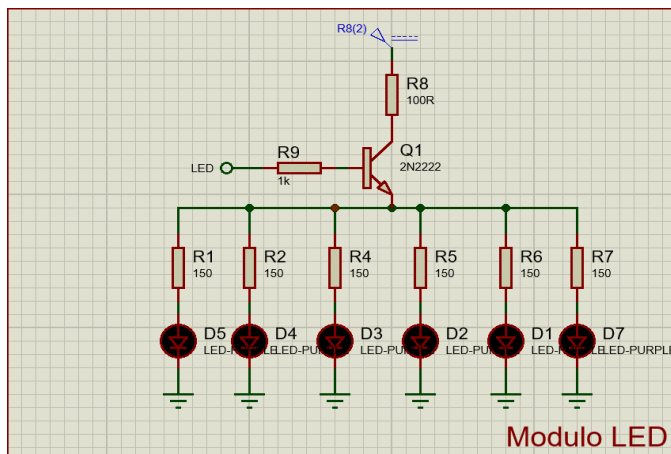


Fig. 30. Esquema del sistema de limpieza en Proteus.

## VII. DISEÑO PLATAFORMA MÓVIL

Para el diseño de la plataforma móvil, se desarrolló una PCB la cual permite de una manera práctica la implementación de los sensores, etapa de potencia, activación de la luz UV y la etapa de control. Por medio de pines del microcontrolador (Arduino uno), se toman decisiones de funcionamiento, dependiendo de las señales que reciba. La diseño de la PCB se desarrolló en el programa **EAGLE (versión 9.0)** [3].

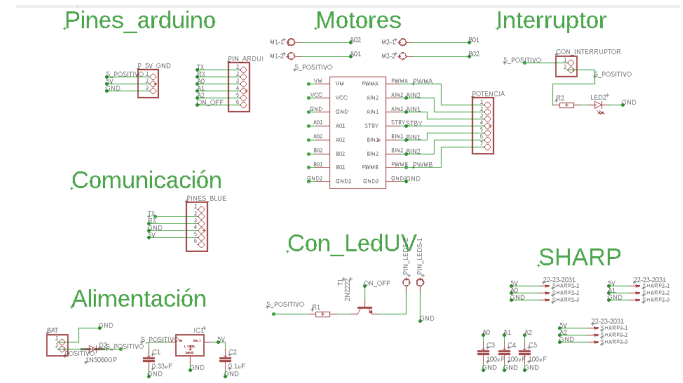


Fig. 31. Esquemático de la PCB.

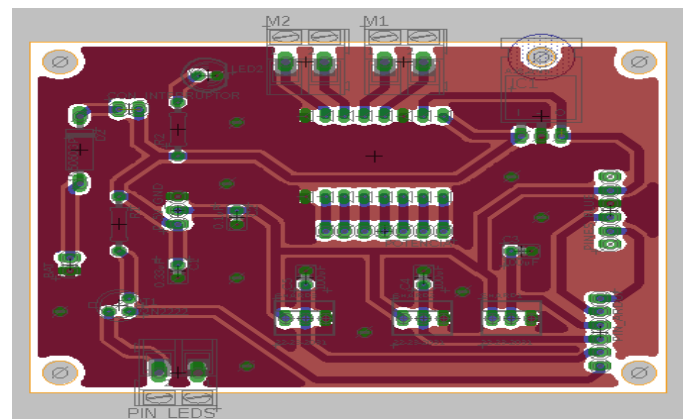


Fig. 32. Board de la PCB.

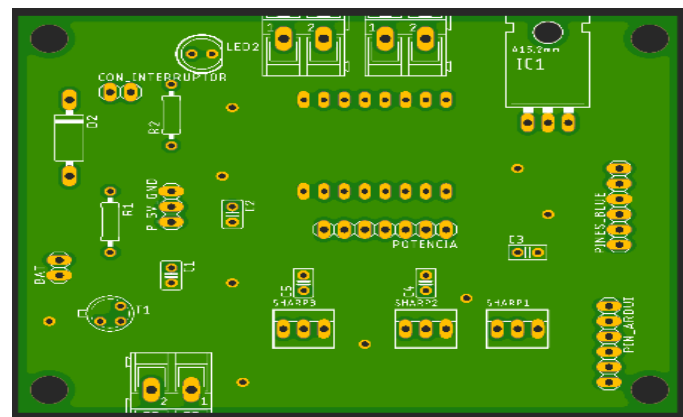


Fig. 33. Diseño para impresión de la PCB.



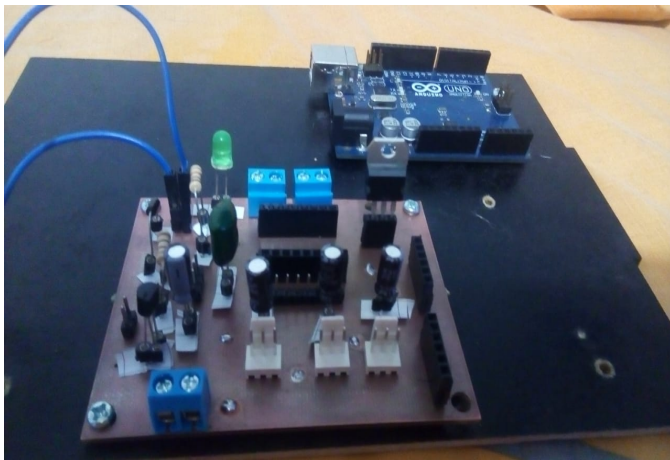


Fig. 34. PCB en físico con los componentes.

Además se desarrolló el diseño del robot, en cual se pensó que fuera de fácil fabricación (MDF 3 mm) y que se pudiera modificar en caso de mejoras. El presente diseño se desarrolló en el programa **SolidWorks (versión 2019)** [4].

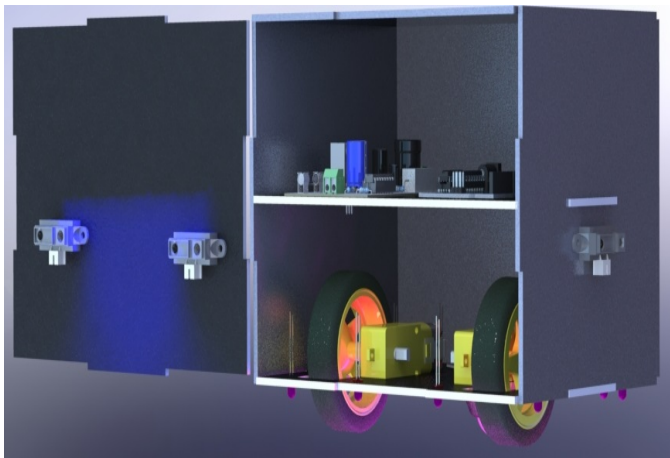


Fig. 35. Diseño del robot para desinfección - vista lateral.

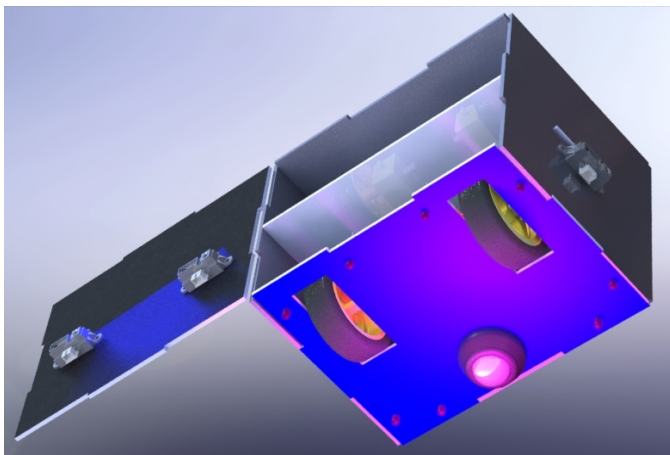


Fig. 36. Diseño del robot para desinfección - vista inferior.



Fig. 37. Diseño del robot para desinfección - vista interior.

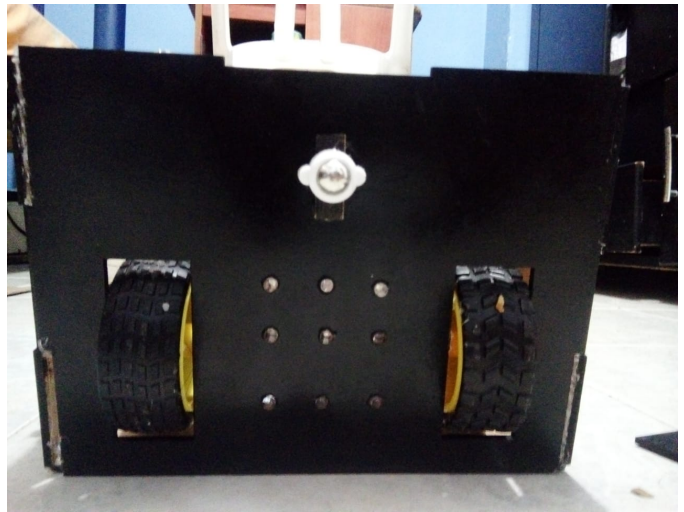


Fig. 38. Implementación del robot para desinfección - vista inferior.



Fig. 39. Implementación del robot para desinfección - vista lateral.

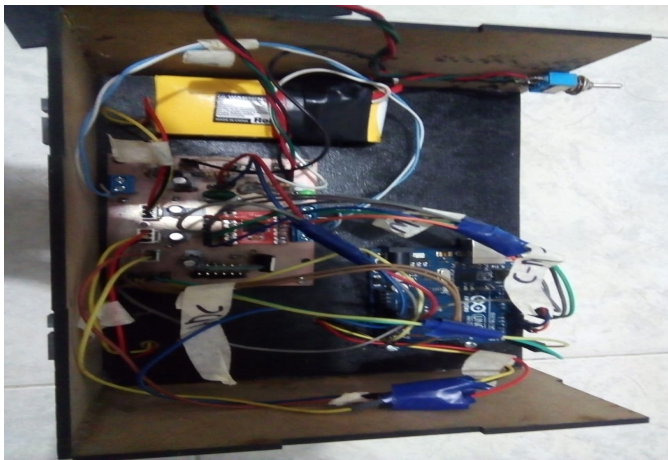


Fig. 40. Implementación del robot para desinfección - vista interior.

### VIII. R.U.P

Para la presente fase, se vuelve a presentar el RUP actualizado, con las nuevas características definidas y implementadas de funcionamiento.

#### A. Requerimientos funcionales

- El robot móvil debe ser capaz de desinfectar zonas interiores moviéndose de una manera autónoma.
- El robot móvil debe ser capaz de detectar y seguir las paredes, esquivar obstáculos del entorno, utilizando 3 sensores de proximidad de referencia **Sharp GP2Y0A21YK0F**.
- El robot móvil debe poder recibir instrucciones del usuario desde una app como: prenderse/apagarse. Esta comunicación se llevará a cabo por medio del módulo bluetooth **HC05**.
- El robot móvil debe poder recibir instrucciones del usuario desde una app como: variar la distancia entre la pared y el robot; y de variar la velocidad del recorrido del robot. Esta comunicación se llevará a cabo por medio del módulo bluetooth **HC05**.

#### B. Requerimientos NO funcionales

- Para la realización de este proyecto se usará el **Visual Studio Code** [5].
- Se usará un microcontrolador **ATMEGA328P**.
- Se usará el software de diseño **Proteus 8 Professional (versión 8.8)**.
- Para la realización de la app se usará el software **Appinventor**.
- Para que el robot móvil se pueda mover se usarán los motores **Motorreductor 3v-6v + Llantá 65mm Motor Arduino Robot Carro..**
- Se usará el puente H **TB6612FNG** que servirá como driver para los motores.

#### C. Casos de uso real

En la Fig. 42, se presenta el caso de uso real para el primer requerimiento.

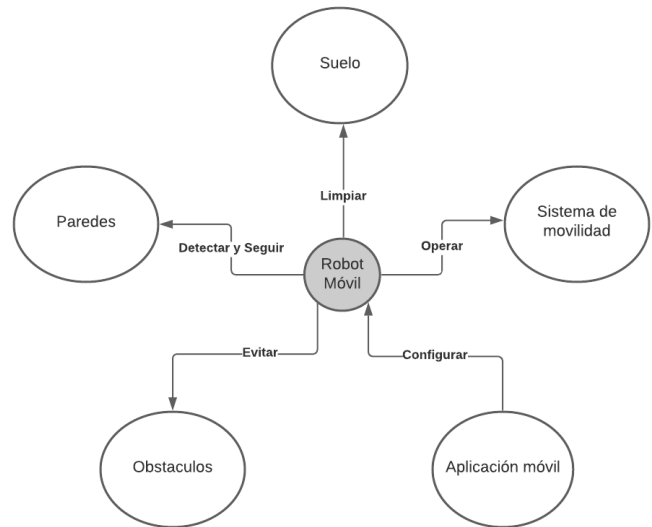


Fig. 41. Diagrama conceptual del sistema.

Curso Normal de los Eventos	
Acción de los Actores -Robot	Respuesta del Sistema - Firmware
	1. El microcontrolador envía la señal de encender las luces Led UV.
2. Las luces Led UV del robot se encienden.	
	3. Se procede a pasar a la tarea encargada del control automático del robot

Fig. 42. Curso normal de eventos para el CUR 1.

En la Fig. 43 se presenta el caso de uso real para el segundo requerimiento.

Curso Normal de los Eventos	
Acción de los Actores - Robot	Respuesta del Sistema - Firmware
	1. Configura el ADC para la lectura de los canales analógicos.
2. Las señales analógicas son entregadas a los puertos GPIO del microcontrolador.	
	3. Las señales analógicas son leídas en los puertos del microcontrolador.
	4. Se calcula la distancia entre el robot y el obstáculo.
	5. Se asigna la prioridad más alta a la tarea de evasión y se pasa a realizar el control.
6. Los motores del robot reciben la señal para el giro del motor, alejando el robot del obstáculo	

Fig. 43. Curso normal de eventos para el CUR 2.

En la Fig. 44 se presenta el caso de uso real para el tercer requerimiento.

Curso Normal de los Eventos		
Acción de los Actores-Usuario	Acción de los Actores-App móvil	Acción de los Actores-robot
1. El usuario mediante la app selecciona con quien establecer la comunicación.		
	2. La aplicación accede al módulo de comunicación del móvil y establece la comunicación.	
3. El usuario selecciona el botón de inicio o de parar desde la app.		
	4. La app procesa la orden y la prepara para enviar por el módulo bluetooth del móvil.	
	5. La app envía la información al robot.	
		6. El módulo bluetooth del robot recibe la información y la envía al microcontrolador mediante el puerto TX.

Fig. 44. Curso normal de eventos para el CUR 3.

En la Fig. 45 se presenta el caso de uso real para el cuarto requerimiento.

Curso Normal de los Eventos		
Acción de los Actores-Usuario	Acción de los Actores-App móvil	Acción de los Actores-robot
1. El usuario mediante la app selecciona con quien establecer la comunicación.		
	2. La aplicación accede al módulo de comunicación del móvil y establece la comunicación.	
3. El usuario digita en la casilla de distancia, la distancia que desea entre el robot y la pared. También la casilla de velocidad, la velocidad a la que el robot limpiará el piso.		
	4. La app procesa la orden y la prepara para enviar por el módulo bluetooth del móvil.	
	5. La app envía la información al robot.	
		6. El módulo bluetooth del robot recibe la información y la envía al microcontrolador mediante el puerto TX.

Fig. 45. Curso normal de eventos para el CUR 4.

#### D. Diagramas de secuencia

A continuación en las Figuras 46, 47, 48 y 49 se presenta los diagramas de secuencia para los diferentes uso de caso real del primer, segundo, tercer y cuarto requerimiento respectivamente.

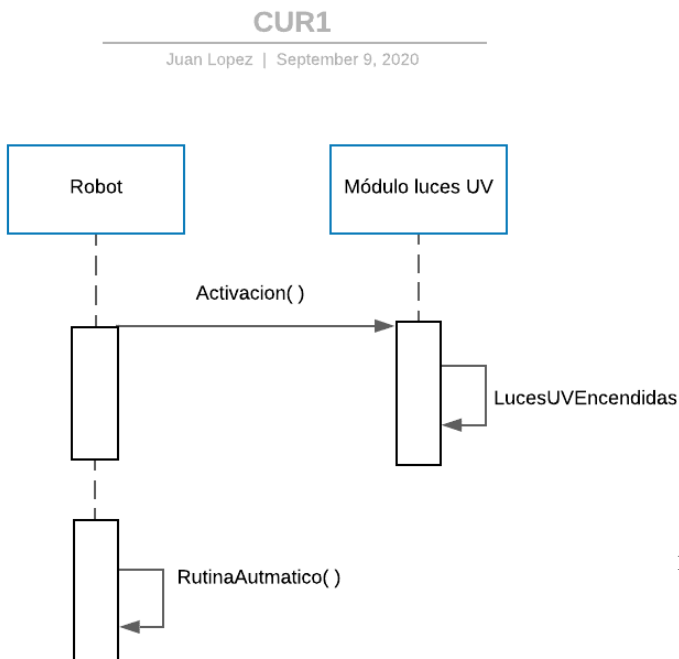


Fig. 46. Diagrama de secuencia CUR1.

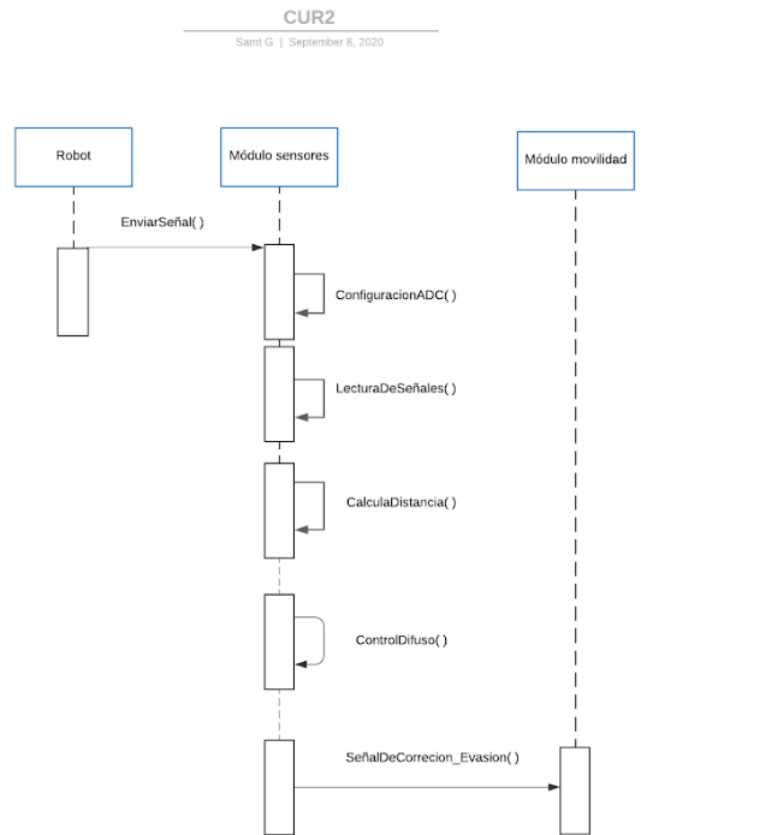


Fig. 47. Diagrama de secuencia CUR2.

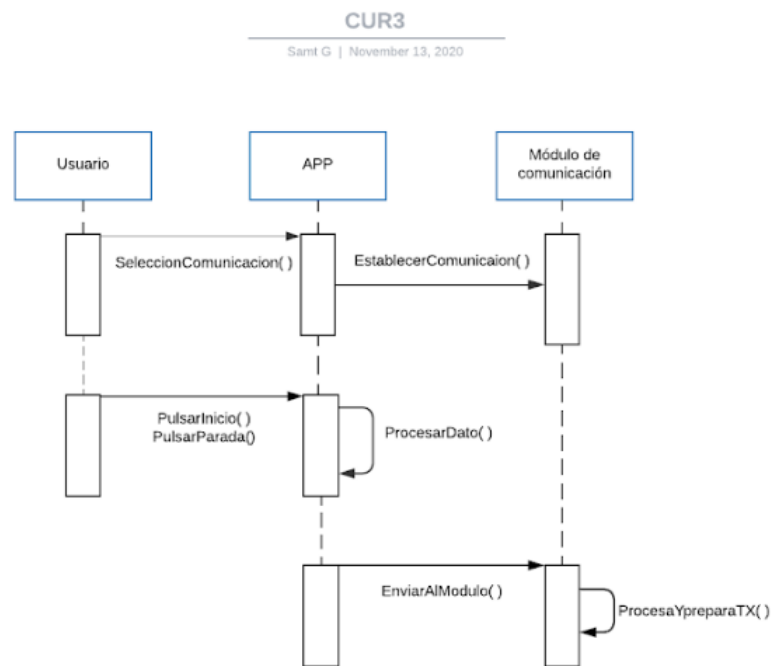


Fig. 48. Diagrama de secuencia CUR3.



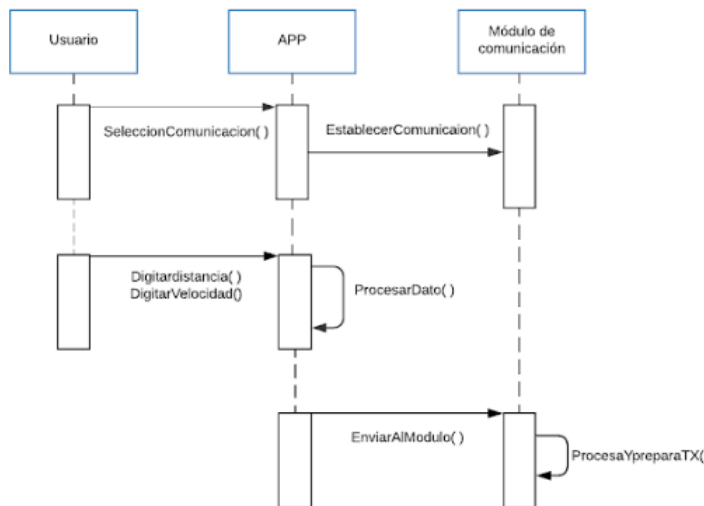


Fig. 49. Diagrama de secuencia CUR4.

## IX. DIAGRAMA DE FLUJO

En la Fig. 50 se presenta el diagrama de flujo del funcionamiento del software.

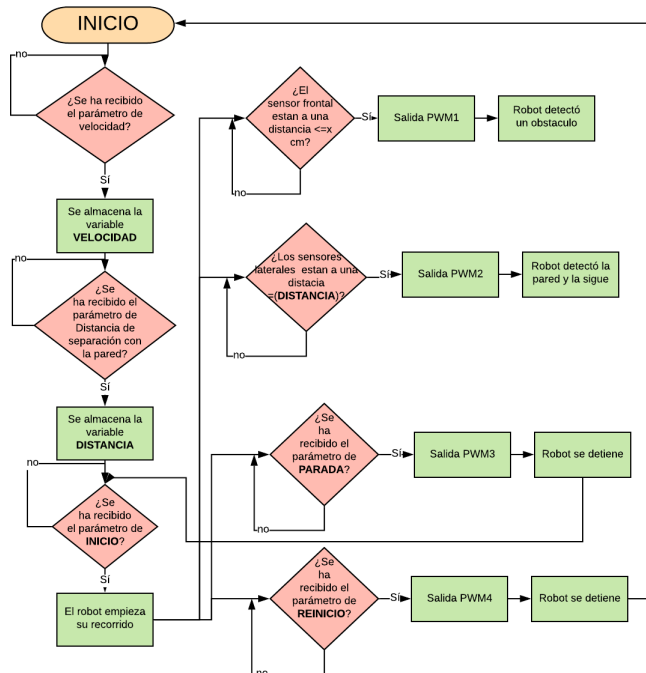


Fig. 50. Diagrama de flujo del funcionamiento del software.

En la Fig. 51 se presenta el diagrama de flujo del funcionamiento de la App.

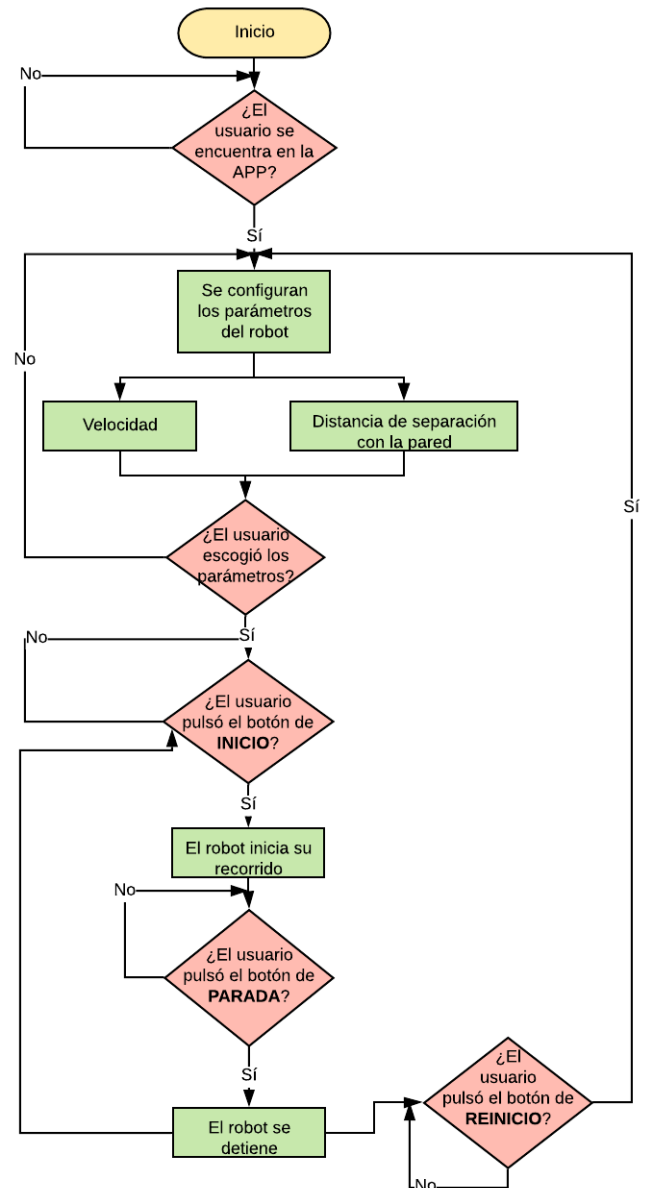


Fig. 51. Diagrama de flujo del funcionamiento de la app.

## X. CONCLUSIONES

- El desarrollo de un proyecto por medio de partes, módulos o etapas, es una buena estrategia para solucionar un problema planteado, debido a que permite tratar de forma individual cada requerimiento solicitado, problema encontrando y solución propuesta.
- Al realizar el desarrollo de la PCB y el diseño del robot, permite dar una visión más amplia de como podría ser la distribución de los componentes y poder así analizar dimensiones según las necesidades y desarrollos planteados.

- Al realizar el proyecto se pudo observar que la ubicación de los sensores no fue la correcta debido a que existen puntos ciegos donde los sensores no puede tomar las medidas correctas o debido al ángulo del robot respecto a la pared se toma una media de distancia que no es útil para el seguimiento.
- A través de la implementación se pudo apreciar la necesidad de agregar filtros por software y por hardware para los sensores, debido a que las variables del ambiente y el ruido generado por los motores introducen medidas erróneas en los sensores.
- El diseño de la estructura fue pensado para ser fácil de ensamblar y producir pero se puede observar que el diseño de "caja" tiene algunas complicaciones como la posibilidad que en un punto ciego la estructura quedé enganchada con algún borde.
- Es importante recalcar que la realización de este tipo de proyecto permite el desarrollo de otras habilidades como el diseño, ya sea de la PCB, App o de estructura, y el desarrollo de arquitecturas de software.
- Para finalizar, a través de las pruebas realizadas es posible afirmar que el robot cumple con todas las especificaciones para la fase siendo capaz de seguir paredes, esquivar obstáculos pequeños y ser operado mediante una App.

#### REFERENCES

- [1] <https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P-Datasheet.pdf>
- [2] <https://appinventor.mit.edu/>
- [3] <https://www.autodesk.com/products/eagle/overview>
- [4] <https://www.solidworks.com/es>
- [5] <https://code.visualstudio.com/>
- [6] <https://www.labcenter.com/>