

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Diseño e implementación de una nueva plataforma móvil para
aplicaciones en robótica de enjambre**

Trabajo de graduación presentado por Carlos Javier Lima Cerdón para
optar al grado académico de Licenciado en Ingeniería Electrónica

Guatemala,

2018

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería




**Diseño e implementación de una nueva plataforma móvil para
aplicaciones en robótica de enjambre**

Trabajo de graduación presentado por Carlos Javier Lima Cerdón para
optar al grado académico de Licenciado en Ingeniería Electrónica

Guatemala,


2018

Vo.Bo.:

(f) 

Ing. Pablo Mazariegos


Tribunal Examinador:

(f) 

Ing. Pablo Mazariegos

(f) 

Ing. Luis Pedro Montenegro

(f) 

PhD. Luis Alberto Rivera

Fecha de aprobación: Guatemala, 04 de Diciembre de 2018.

Trabajo de graduación que se llevó a cabo en la modalidad de tesis, aplicando los conocimientos teóricos y prácticos adquiridos a lo largo de la carrera para la implementación del modelo robótico con fines de aplicación en robótica de enjambre.

Agradezco a Dios por ser quien siempre ha guiado mi camino, a mi familia por su apoyo en los buenos y malos momentos, a la fundación Juan Bautista Gutiérrez y su equipo por confiar y beneficiarme con su beca, a mis amigos y compañeros que me apoyaron y aconsejaron en todo este trayecto, a mi asesor de tesis Pablo Mazariegos por su guía y aportes significativos en la realización de este trabajo, al profesor Miguel Zea por su apoyo en este trabajo y por último a la Universidad del Valle de Guatemala por ser mi casa de estudio.

Prefacio	v
Lista de figuras	XIII
Lista de cuadros	XV
Resumen	XVII
Abstract	XIX
1. Introducción	1
2. Antecedentes	3
2.1. AMiR	3
2.1.1. Sensor infra-rojo	4
2.1.2. Unidad de control	4
2.1.3. Control de movimiento	4
2.2. Cellulo	5
2.2.1. Localización	6
2.2.2. Movimiento	7
2.2.3. Diseño de software	7
2.3. Alice	7
2.4. Jasmine	8
2.5. E-Puck	9
2.5.1. Microcontrolador	10
2.5.2. Sensores y actuadores	10
2.5.3. Interfaz con el usuario	11
2.5.4. Estructura mecánica	11
2.6. Kobot	12
2.7. Kilobot	13
2.8. R-One	15
2.9. Colias	16
2.9.1. Controlador	17

2.9.2.	Movimiento	17
2.9.3.	Sistema de sensores	18
2.9.4.	Control de potencia	18
2.10.	Plataforma robótica para aplicaciones en robótica de enjambre fase 1	19
2.10.1.	Partes del robot	20
3.	Justificación	25
4.	Objetivos	27
4.1.	Objetivo general	27
4.2.	Objetivos específicos	27
5.	Alcance	29
6.	Marco teórico	31
6.1.	Comunicación digital inalámbrica	31
6.1.1.	Capas de arquitectura TCP/IP	31
6.1.2.	Wi-Fi	32
6.1.3.	Módulo ESP8266	33
6.2.	Microprocesadores y microcontroladores	33
6.2.1.	Teensy	34
6.2.2.	Microcontrolador de 32 bits	35
6.3.	Servomotor	37
6.3.1.	DFRobot micro servo	38
6.3.2.	Traxxas nano servo digital	38
6.3.3.	BMS-306DMAX digital micro servo	38
6.4.	Sensor de proximidad	39
6.4.1.	Sensor ultrasónico	39
6.4.2.	LiDAR	42
6.5.	Controladores de motores DC	43
6.5.1.	Controlador DRV8833	43
6.5.2.	Controlador TB6612FNG	44
6.6.	Encoders	44
6.6.1.	SEN0116	45
6.7.	Batería Li-Po	46
6.7.1.	PKCELL LP785060	46
6.7.2.	Batería Quanum FPV 7.4V	46
7.	Metodología	47
8.	Diseño electrónico de la nueva plataforma	49
8.1.	Identificación de áreas de mejora de la plataforma de la fase 1	49
8.1.1.	Unidad de control	50
8.1.2.	Sistema de detección de objetos	50
8.2.	Selección de componentes	50
8.2.1.	Sistema de detección de objetos	50
8.2.2.	Unidad de control	53
8.3.	Módulos de la nueva plataforma	55
8.3.1.	Sistema de movimiento	55

8.3.2.	Sistema de medición	56
8.3.3.	Sistema de detección de objetos	57
8.3.4.	Sistema de comunicación	58
8.3.5.	Interfaz usuario-robot	59
8.3.6.	Sistema de potencia	60
8.3.7.	Diagrama electrónico final	61
8.4.	Programación del microcontrolador	63
8.4.1.	Configuración inicial del microcontrolador	63
8.4.2.	Sistema de movimiento	64
8.4.3.	Sistema de medición	66
8.4.4.	Sistema de detección de objetos	67
8.4.5.	Sistema de comunicación	70
9.	Diseño de la estructura de la nueva plataforma	73
9.1.	Dimensiones físicas de la plataforma de la fase 1	73
9.2.	Diseño de base de la estructura	74
9.3.	Diseño PCB	74
9.4.	Estructura final	76
9.5.	Costo de la nueva plataforma	76
9.6.	Comparación de plataformas fase 1 y 2	78
10.	Implementación de librerías en Matlab	81
11.	Pruebas con la nueva plataforma	83
11.1.	Sistema de detección	83
11.2.	Sistemas de movimiento y medición	85
11.3.	Pruebas de temperatura	88
12.	Conclusiones	91
13.	Recomendaciones	93
14.	Bibliografía	95
15.	Anexos	99
15.1.	Archivos generales de diseño de la nueva plataforma	99
15.2.	Código final del microcontrolador PIC32MX250F128B en lenguaje C	99
15.3.	Librería de funciones en Matlab para controlar plataforma	99
15.4.	Archivos gerber de PCB	99
15.5.	Diagrama electrónico final	100
15.6.	Comandos AT ESP-8266	100
15.7.	Diseños de estructura en Inventor	100

Lista de figuras

1.	Prototipo de Autonomous Miniature Robot (AMiR)	3
2.	Placa principal de AMiR, con topología de transeptores de 60°.	4
3.	Tipos de rotaciones de AMiR	5
4.	Dos revisiones del diseño de Cellulo	6
5.	Arquitectura de programación de una actividad de Cellulo	7
6.	El tamaño de Alice es similar al de una hormiga	8
7.	Tercera versión del micro robot Jasmine	9
8.	Robot E-puck [8].	9
9.	Estructura electrónica de E-puck	10
10.	Estructura mecánica de E-puck	12
11.	Simulación, posición básica de sensores y versión básica de Kobot	13
12.	Diagrama de sistema de control de Kobot	14
13.	Vista isométrica e inferior de Kilobot	15
14.	R-One robot, control multi robots y vista simulada de R-One	16
15.	Partes y simulación de Colias	17
16.	Arquitectura básica de Colias	18
17.	Plataforma de fase 1 de UVG	19
18.	Diseño de falda de sensores ultrasónicos junto a PCB de control	21
19.	Ensamblaje de llanta de goma con <i>encoder</i> y motor DC	22
20.	Señales de un <i>encoder</i>	23
21.	Módulo ESP-8266 de Sparkfun	33
22.	Diagrama de bloques general de un microprocesador	34
23.	Diagrama de bloques general de un microcontrolador	34
24.	Teensy 3.2 con ubicación de pines	36
25.	PIC32MX250F128B con empaquetado SPDIP	36
26.	Conexión mínima requerida para operar al PIC32MX250F128B	37
27.	PickIt y sus conexiones a cualquier PIC	37
28.	Micro servomotor DFRobot	38
29.	Nano servomotor digital Traxxas	38
30.	BMS-306DMAX Digital Micro Servo	39
31.	Sensor Ultrasónico HC-SR04	40
32.	Dimensiones físicas del LV-MaxSonar-EZ1	41

33.	Devantech SRF02	41
34.	Dimensiones físicas del ultrasónico Devantech SRF02	41
35.	LIDAR-Lite V3	43
36.	Dimensiones físicas de LeddarOne	43
37.	Pines del DRV8833 Dual Driver Motor	44
38.	Pines del TB6612FNG Dual Driver Motor	44
39.	Construcción básica de un <i>encoder</i> óptico	45
40.	<i>Encoder</i> óptico SEN0116 de DFRobot	45
41.	Batería PKCELL LP785060	46
42.	Batería Quanum FPV de 7.4V y 1500mAh	46
43.	Pines del PIC32 y tipos de empaquetados disponibles	54
44.	Diagrama electrónico del sistema de movimiento.	55
45.	Posición física de los pines de TB6612FNG simulada en Altium.	55
46.	Diagrama electrónico del sistema de medición de movimiento.	56
47.	Diagrama electrónico del sistema de detección de objetos.	57
48.	Diagrama electrónico del sistema de comunicación.	58
49.	Diagrama electrónico de la interfaz Usuario-Robot.	59
50.	Diagrama de conexión de reguladores de la serie L78	61
51.	Diagrama final de la nueva plataforma	62
52.	Diagrama de flujo de la configuración general del microcontrolador.	64
53.	Diagrama de flujo de la configuración del sistema de movimiento.	65
54.	Diagrama de flujo de la configuración del sistema de medición.	67
55.	Diagrama de flujo de la configuración del sistema de detección.	69
56.	Diagrama de flujo de la configuración del sistema de comunicación.	70
57.	Dimensiones y área ocupada por el modelo anterior en el plano <i>XY</i>	73
58.	Vista de planta de la plataforma robótica fase 1.	74
59.	Dimensiones de la base de la nueva plataforma en Inventor.	74
60.	Vista superior e inferior de base con llantas.	75
61.	Vista superior e inferior de placa con componentes simulada	75
62.	Vistas de placa con componentes soldados y con módulos incorporados	76
63.	Conexión de los componentes del robot entre base y placa.	77
64.	Vistas de planta, elevación, perfil e isométrica de la nueva plataforma ensamblada.	77
65.	Señalización de componentes relevantes sobre la placa de cobre.	78
66.	Comparación de plataformas de robótica de enjambre fase 1 y 2	79
67.	Dimensiones del área ocupada por la nueva plataforma en el plano <i>XY</i>	80
68.	Prueba de detección de objetos	84
69.	Prueba de detección de cilindros 1	85
70.	Prueba de detección de cilindros	85
71.	Prueba de detección de cilindros	86
72.	Prueba de detección de cilindros	86
73.	Gráficas de velocidades de llantas contra el duty cycle de las señales PWM	87
74.	Capturas de temperatura en la nueva plataforma cuando no está en funcionamiento	88
75.	Capturas de temperatura luego de 20 minutos de funcionamiento	88

76.	Capturas de temperatura luego de 20 minutos de funcionamiento 2	89
77.	Voltaje de salida del regulador AZ1117-3.3 en función de la temperatura	89

Lista de cuadros

1.	Costo del prototipo robótico Cellulo	6
2.	Comparación de distintas plataformas para robótica de enjambre	16
3.	Partes de la plataforma para robótica de enjambre de UVG Fase 1	20
4.	Modelo de arquitectura TCP/IP	32
5.	Características de distintos sensores de distancia.	51
6.	Estudio de características de sensores de distancia.	52
7.	Características de los servomotores tomados en cuenta.	53
8.	Estudio de características de servomotores.	53
9.	Listado de componentes y pines que se conectarán a la unidad de control.	54
10.	<i>Power Budget</i> de los componentes que utilizará la nueva plataforma.	60
11.	Configuración de sentido de giro de llantas.	65
12.	Comandos para modificar la velocidad y sentido de giro de llantas.	66
13.	Comandos la medición de sentido de giro, <i>ticks</i> y tiempos	68
14.	Comando para realizar un barrido de detecciones de objetos alrededor del robot.	69
15.	Comandos AT para la configuración y comunicación con el módulo ESP-8266	71
16.	Detalle de costos de la nueva plataforma.	78
17.	Funciones de librería de Matlab para controlar a la nueva plataforma.	82

Se diseñó e implementó una nueva plataforma móvil para aplicaciones en robótica de enjambre. La implementación de esta plataforma se realizó como un seguimiento al primer modelo diseñado el año 2017 dentro del megaproyecto “Reingeniería de proyectos”. Se buscó mejorar el rendimiento, el tamaño físico, la plataforma de control y el sistema de detección de objetos alrededor del robot.

La plataforma se controla desde un ordenador con acceso a una red Wi-Fi, utilizando 13 funciones desarrolladas en MatLab. El diseño electrónico y mecánico se realizó, de tal forma que sus dimensiones físicas fueran más compactas, donde se obtuvo una reducción del 36.44 % en las dimensiones físicas del plano XY respecto al modelo anterior. Tiene un nuevo sistema de detección de objetos que posee dos nuevos modelos de ultrasónicos que rotan alrededor del eje Z con un servomotor, que realiza un barrido de detección de objetos sobre todo el plano XY , como mejora en comparación con los 6 puntos inmóviles de detección que realizaba el modelo anterior mencionado.

Se mejoró la plataforma de control sustituyendo el microcontrolador Teensy utilizado en el modelo anterior por una plataforma más robusta utilizando el microcontrolador PIC32MX250F128B, donde se obtuvo un mejor control de todas las capacidades que este ofrece. El nuevo microcontrolador es el centro de comando que recibe desde un ordenador las instrucciones a ejecutar en la plataforma mediante un módulo Wifi, donde así mismo envía de vuelta al ordenador la información requerida.

Se utilizó una batería recargable como fuente principal de alimentación, que provee la potencia necesaria para que la plataforma funcione como mínimo durante 4 horas seguidas en su máximo consumo continuo.

A new mobile platform for applications in swarm robotics was designed and implemented. The implementation of this platform was carried out as a follow-up to the first model designed in 2017 within the mega-project "Reengineering of projects". We sought to improve the performance, the physical size, the control platform and the system for detecting objects around the robot.

The platform is controlled from a computer with access to a Wi-Fi network, using 13 functions developed in MatLab. The electronic and mechanical design was carried out in such a way that its physical dimensions were more compact, where a reduction of 36.44 % in the physical dimensions of the XY plane was obtained with respect to the previous model. It has a new object detection system that has two new ultrasonic models that rotate around the Z axis with a servomotor, which performs an object detection sweep over the entire XY plane, as an improvement compared to the 6 immobile points of detection that the mentioned previous model realized.

The control platform was improved by replacing the Teensy microcontroller used in the previous model with a more robust platform using the PIC32MX250F128B microcontroller, where better control of all the capabilities it offers was obtained. The new microcontroller is the command center that receives from a computer the instructions to execute on the platform through a Wifi module, where it also sends the required information back to the computer.

A rechargeable battery was used as the main source of power, which provides the necessary power for the platform to work for at least 4 hours at its maximum continuous consumption.

La implementación de la nueva plataforma robótica para aplicaciones en robótica de enjambre se desarrolla como un proyecto de seguimiento a la Fase 1 desarrollada en la Universidad del Valle de Guatemala en el año 2017, siendo esta nueva versión la Fase 2.

La plataforma de la Fase 1 consta de las siguientes funcionalidades: Detección de objetos alrededor de la plataforma con una falda fija de 6 sensores ultrasónicos posicionados a 60° entre cada uno; movilización de la plataforma en superficie plana, utilizando un sistema de movimiento del tipo *differential drive* que consta de dos llantas controladas de forma individual y un *ball caster* que sirve como apoyo y equilibrio de la plataforma; medición del movimiento de la plataforma, utilizando un *encoder* en cada llanta para obtener la velocidad, distancia recorrida y sentido de orientación de cada llanta; comunicación Wi-Fi, utilizando un módulo ESP-8266, que permite la comunicación inalámbrica con un ordenador del que recibe instrucciones y al que le envía información solicitada; por último, una librería de instrucciones desarrollada en Matlab para controlar las funcionalidades del robot desde el ordenador. Cabe mencionar que la plataforma móvil no procesa los datos de sus sensores, simplemente realiza las lecturas y las manda al ordenador, donde si son procesadas a tiempo real para determinar las siguientes acciones del robot.

Para la implementación de la nueva plataforma se tomó en cuenta la retroalimentación de parte de los ingenieros del proyecto y las recomendaciones colocadas en el trabajo escrito que detalla el desarrollo de la Fase 1. A partir de esta información se definieron como áreas para optimización al centro de comando y el sistema de detección de objetos de la plataforma. Como centro de comando de la plataforma anterior se utilizó un Teensy 3.2 [1], con programación basada en Arduino [2], que tiene como ventaja que es una herramienta fácil para generar prototipos de forma rápida, utilizando librerías de programación predefinidas, pero que conlleva la desventaja de no tener el control total o exacto de los módulos del microcontrolador, lo que puede incurrir en un ineficiente flujo de control de tiempos del microcontrolador, por lo tanto, se cambia el Teensy 3.2 por el microcontrolador PIC32MX250F128B [3], que no está basado en Arduino, se programa en lenguaje C y permite el acceso a todas sus funcionalidades, teniendo un mejor control de sus módulos, lo que

permite un mejor control del robot y sus sistemas. Por parte del sistema de detección de objetos, el modelo anterior consta de una falda de 6 ultrasónicos inmóviles en la plataforma, con la desventaja de que ocupa bastante espacio por sus dimensiones físicas y la forma en que están puestos, además que solo puede detectar 6 puntos exactos en todo el plano XY alrededor de la plataforma. Por otro lado, en las recomendaciones del trabajo escrito de la Fase 1 se indica que las lecturas de esta falda de sensores fue inconsistente [1], por lo tanto, como nuevo y mejor sistema de detección se utilizarán únicamente dos nuevos sensores de distancia y un servomotor que hace girar estos sensores alrededor del eje Z para realizar un barrido de detección en todo el plano XY , obteniendo como consecuencia mejoras en una menor área física de la plataforma utilizada y una notable mayor cantidad de detecciones alrededor de la plataforma.

2.1. AMiR

Autonomous Miniature Robot (AMiR) es una plataforma de bajo costo para aplicaciones en robótica de enjambre, con dimensiones físicas de 6cm x 7.3cm x 4.7cm. Este robot puede estimar la distancia de obstáculos y reconocer múltiples robots en un entorno, es capaz de realizar comunicaciones de corto alcance utilizando señales infra-rojas (IR) y controla la velocidad de dos ruedas de forma individual para su movilización. Utiliza una batería Li-Po de 3.7v y 400mAh que provee alrededor de 120 minutos de autonomía. El costo de AMiR es de €65. [4]

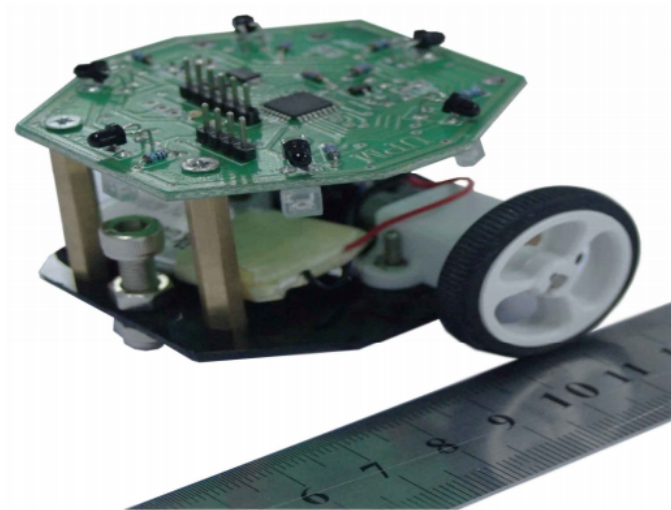


Figura 1: Prototipo de Autonomous Miniature Robot (AMiR) [4].

para sus motores. La velocidad de una llanta se calcula con la ecuación 1. [4]

$$V = \pi d_w N \cong 854 \text{ cm/min} \quad (1)$$

Donde d_w es el diámetro de cada rueda, 32mm para las de AMiR y N es la velocidad de cada motor, en este caso de 85 RPM. [4]

La estimación de la rotación respecto a un punto se realiza mediante dos métodos: i) rotación a largo diámetro, como se muestra en la Figura 3(a) donde el eje de rotación está fuera del área de la plataforma, y ii) girando alrededor de su propio eje central entre las llantas como se muestra en la Figura 3(b). En la rotación a largo diámetro, las velocidades rotacionales de ambos motores son diferentes y se calculan con la ecuación 2. [4]

$$N_L = \frac{2R + W}{2R - W} N_R \quad (2)$$

Donde N_L es la velocidad del motor de la izquierda y N_R es la velocidad del motor de la derecha. W es la distancia entre las dos llantas y R es el radio de rotación. De igual forma, el radio de rotación es calculado por la ecuación 3. [4]

$$R = \frac{W}{2} \left(\frac{N_L + N_R}{N_L - N_R} \right) \quad (3)$$

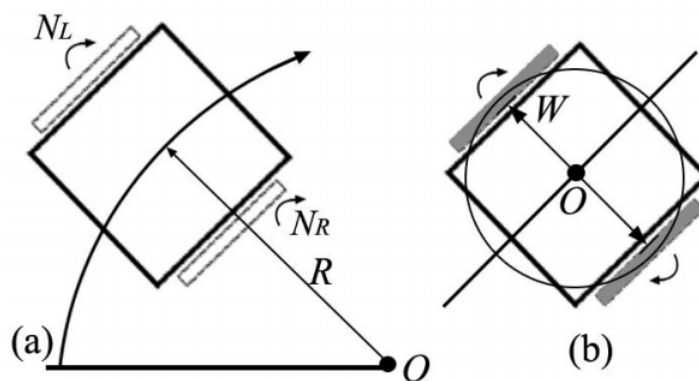


Figura 3: Tipos de rotaciones de AMiR (a) el eje de rotación está fuera del área de AMiR, (b) girando sobre el mismo lugar [4].

2.2. Cellulo

Es una plataforma robótica móvil de bajo costo que opera en hojas de papel impresas, superpuestas con un patrón de micro puntos que permite la auto-localización. Los robots están diseñados para ser de bajo costo y fáciles de operar; todos los robots están conectados de forma inalámbrica a un dispositivo móvil (una tableta o un teléfono inteligente) que ejecuta la actividad y la lógica de orquestación. Cellulo puede auto-localizarse en la hoja de

actividad, puede realizar movimiento en cualquier dirección, posee 5 botones táctiles capacitivos (retro iluminados independientemente en RGB completo) y comunicación inalámbrica Bluetooth. El costo total del prototipo robótico Cellulo es de €125, el costo más detallado se observa en el Cuadro 1. [5]

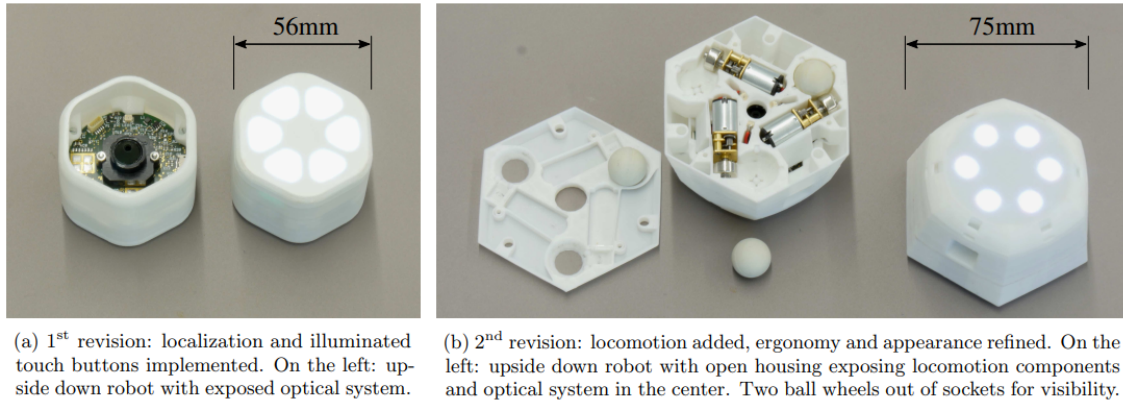


Figura 4: Dos revisiones del diseño de Cellulo. Preserva la escala 1-1 entre las dos imágenes [5].

Componente	Costo (€)
Procesador (<i>PIC32MZ1024ECG064</i>)	8.51
Subsistema de localización	17.24
Subsistema de movimiento	49.77
Comunicación (<i>RN42</i>)	12.12
Subsistema UI (<i>TLC5947, AT42QT1070</i>)	7.36
Batería, cargador y protección	8.86
Otros (Terminales, resistores, capacitores, etc)	8.77
Encapsulado (53.3g PLA) y fijación	1.91
Manufactura de PCB	11.39
Total	125.93

Cuadro 1: Costo del prototipo robótico Cellulo. [5]

2.2.1. Localización

El método de localización de Cellulo utiliza un patrón óptico de micro puntos que se colocan en la superficie donde el robot los detecta e interpreta. Utilizando este patrón óptico de micro puntos, se puede cubrir un área de hasta 170 millones de km^2 con patrones únicos. Los robots están equipados con una cámara orientada hacia abajo, donde el microcontrolador decodifica una región de 1 cm^2 y obtiene su posición en x, y y θ a 93Hz. Este método ofrece localización global y absoluta con precisión de 0.27mm y 1.5° sin la necesidad de ninguna calibración. [5]

2.2.2. Movimiento

Cellulo es capaz de tener movimiento holonómico, es decir, la plataforma puede moverse en cualquier dirección, lo que permite una libertad de movimiento en entornos donde el espacio es estrecho y no importa la orientación que tenga el robot. [5]

2.2.3. Diseño de software

El comportamiento de Cellulo se define a partir de la combinación de los siguientes elementos: papel donde se encuentra el patrón de micro puntos, los robots vecinos y los usuarios que están controlando al robot a través del software específico desarrollado para Cellulo. [5]

El microcontrolador que funciona como unidad de control de la plataforma, está programada en lenguaje C, donde se maneja el control de movimiento, control háptico, procesamiento de imágenes para la localización y el control de los LED's. Cada robot está conectado a través de Bluetooth 2.1 Serial Port Profile (SPP) a la tableta que ejecuta la aplicación *QtQuick* para coordinar la aplicación. La arquitectura de este núcleo de elementos se observa en la Figura 5. [5]

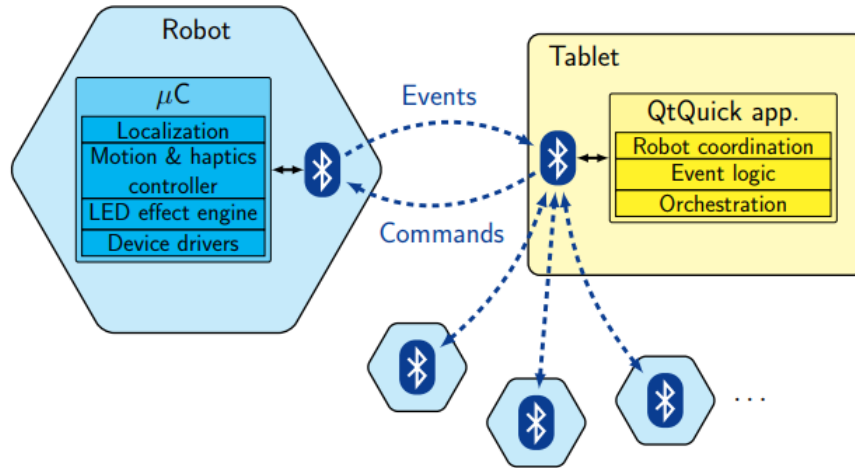


Figura 5: Arquitectura de programación del elemento más básico de una actividad de Cellulo. Múltiples robots están conectados a través de Bluetooth a la tableta móvil [5].

2.3. Alice

Alice tiene una autonomía de 10 horas y es capaz de moverse de acuerdo a simples comportamientos, como por ejemplo moverse evitando obstáculos o seguimiento de paredes. Es uno de los robots inteligentes más pequeños en el mundo [6], teniendo dimensiones de 21 x 21 x 12 mm, peso de 5g, velocidad de hasta 4 cm/s, comunicación remota IR de 6m a 500 bps, comunicación local IR de 4cm a 500 bps y radio comunicación de 10 m a 1000 bps. Alice se

compone de 2 motores de reloj, 2 llantas, batería de 1.5V, 23 mAh con regulador de voltaje, 4 sensores de proximidad IR, PIC16F84 a 4MHz como centro de comando, comunicación local 1 con los mismos 4 sensores de proximidad, comunicación 2 IR en una dirección con 1 diodo y 2 OpAmps, comunicación 3 IR en una dirección con el estándar RC5 y comunicación 4 Radio en ambos sentidos. [6]

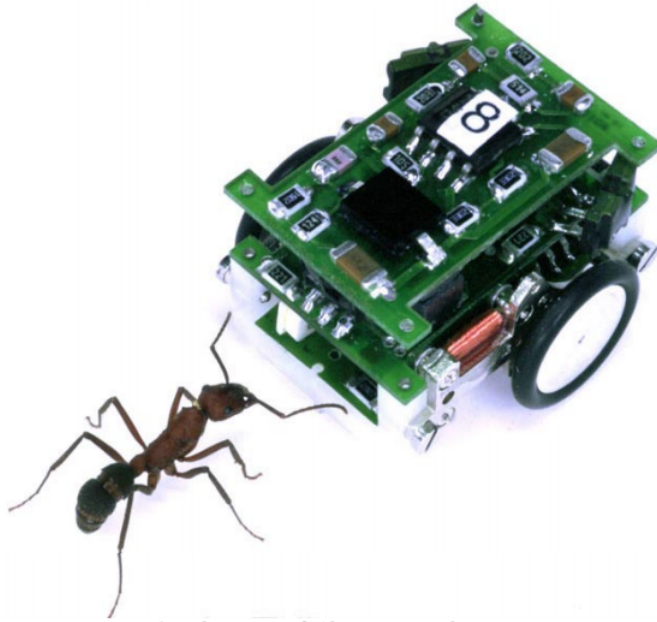


Figura 6: El tamaño de Alice es similar al de una hormiga [6].

2.4. Jasmine

Jasmine se utilizó en [7] para realizar experimentos de enjambre. Es un desarrollo público de hardware abierto con el objetivo de crear una plataforma micro robótica simple y rentable. Este micro robot es de 30 x 30 x 20 mm, utiliza dos Atmel AVR Mega micro controladores: Atmel Mega88 para controlar el motor, la posición del robot, color de los LED's y el monitoreo de los niveles de la batería; Mega168 para controlar la comunicación, el monitoreo con sensores, control remoto, detección y tareas definidas por el usuario. Los dos micro controladores se comunican por I^2C . [7]

Tiene 6 canales de comunicación, separados 60° entre cada uno, también utilizados para detección de proximidad, tiene un canal geométrico de percepción de 15° de ángulo abierto. El área de comunicación cubre los 360° . El robot utiliza dos motores DC con engranajes internos y dos llantas de manejo diferencial con acoplamientos llanta-motor. Sin utilizar un *encoder*, el robot es capaz de estimar la distancia recorrida con una exactitud del 6 % y un ángulo de rotación del 11 %. Jasmine III utiliza una batería Li-Po de 3.7V con un estabilizador de voltaje interno. La corriente que consume durante movimiento constante es de 200 mA, consume 6 mA en modo estacionario y menos de 1 mA en stand-by. Tiene una autonomía entre 1 y 2 horas. Se utilizó lenguaje C para programar los microcontroladores con el compilador *open source gcc*. [7]



Figura 7: Tercera versión del micro robot Jasmine [7].

2.5. E-Puck

E-Puck es un robot móvil de escritorio desarrollado por École Polytechnique Fédérale de Lausanne (EPFL) para actividades docentes. Todo el software, diseño y documentos de producción están disponibles bajo una licencia de hardware y software open source. La producción de una versión básica del E-puck ronda los €250, las compañías que los producen los venden alrededor de los €550. [8]



Figura 8: Robot E-puck [8].

2.5.1. Microcontrolador

La estructura electrónica de E-Puck está construida alrededor de microcontrolador dsPIC30F6014A (Figura 9). Este CPU corre a 64 MHz y provee 16 MIPS pico como fuerza de procesamiento. La versión escogida para E-puck tiene 8 kB de RAM y 144 kB de memoria flash. [8]

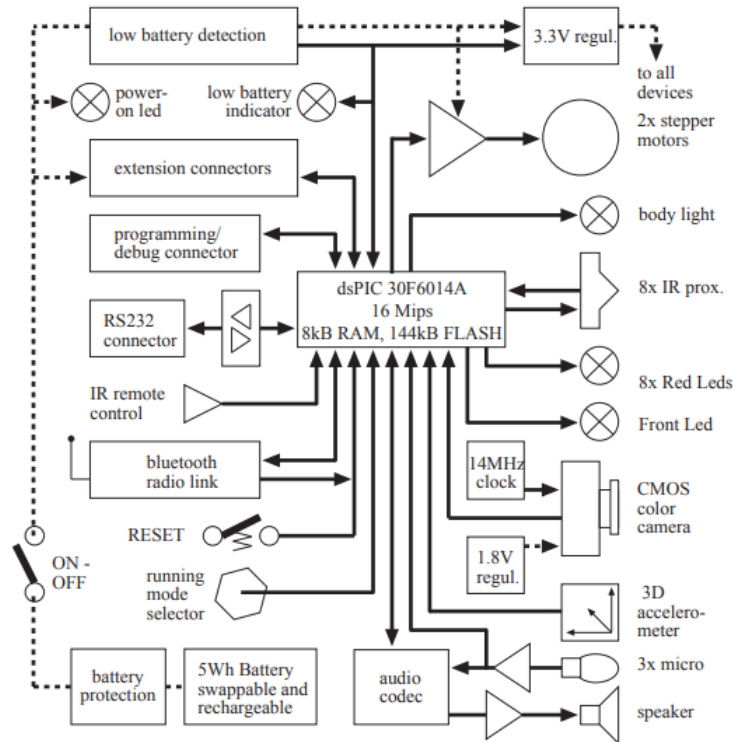


Figura 9: Estructura electrónica de E-puck [8].

2.5.2. Sensores y actuadores

Los sensores que posee el robot son: [8]

- 8 sensores de proximidad IR alrededor de la plataforma para detectar objetos cercanos o la intensidad de luz infrarroja en el ambiente.
- Un acelerómetro 3D se obtiene el vector de aceleración del robot, con este vector se puede medir la inclinación, aceleración, colisiones y si el robot se cae.
- 3 micrófonos para localizar una fuente de sonido por triangulación.
- Una cámara CMOS a color con resolución de 640x480 pixeles al frente del robot.

Los actuadores son: [8]

- 2 motores stepper que controlan el movimiento de las llantas con una resolución de 1000 pasos por revolución de cada llanta.
- Una bocina, que combinada con los micrófonos crean una red de comunicación que sirve para detectar la dirección de robots vecinos o para la interacción con un humano.
- 8 LED's de color rojo posicionados alrededor de la plataforma, con intensidad controlada que pueden ser detectados por la cámara de otro robot.
- Un set de LED's verdes posicionados en el cuerpo transparente para la interacción con usuario.
- Un LED rojo posicionado al lado de la cámara para proyectar un haz rojo sobre objetos al frente del robot, que junto con la cámara, se pueden realizar mediciones de distancia a rangos mayores que con los sensores de proximidad.

2.5.3. Interfaz con el usuario

Posee varios dispositivos para interactuar con un usuario o para comunicarse con otros equipos: [8]

- 2 LED's que muestran el estado de la batería: uno indica si el robot está encendido y el otro indica si la batería tiene poca carga.
- Un conector a la interfaz para programar la memoria flash y depuración de código.
- Un control remoto IR receptor para controlar al E-puck con un control remoto estándar de televisión.
- Un serial RS232 para comunicarse con un ordenador.
- Radio bluetooth para comunicarse con un ordenador o con otros 7 E-pucks.
- Botón de reset.
- Un interruptor giratorio de 16 posiciones para formar un número de 4 bits, utilizado para seleccionar parámetros o comportamiento pre-programados.

2.5.4. Estructura mecánica

El robot tiene un diámetro de 75mm y una altura variable que depende de las extensiones conectadas. Utiliza partes plásticas inyectadas, que minimiza el precio, son cuatro: El cuerpo principal, es el núcleo de la estructura mecánica y encapsula a la batería que puede ser extraída por el usuario desde abajo. Los dos motores son atornillados al cuerpo principal, con las llantas unidas a los ejes de los motores. La placa electrónica principal (PCB) está atornillada sobre el cuerpo plástico principal. [8]

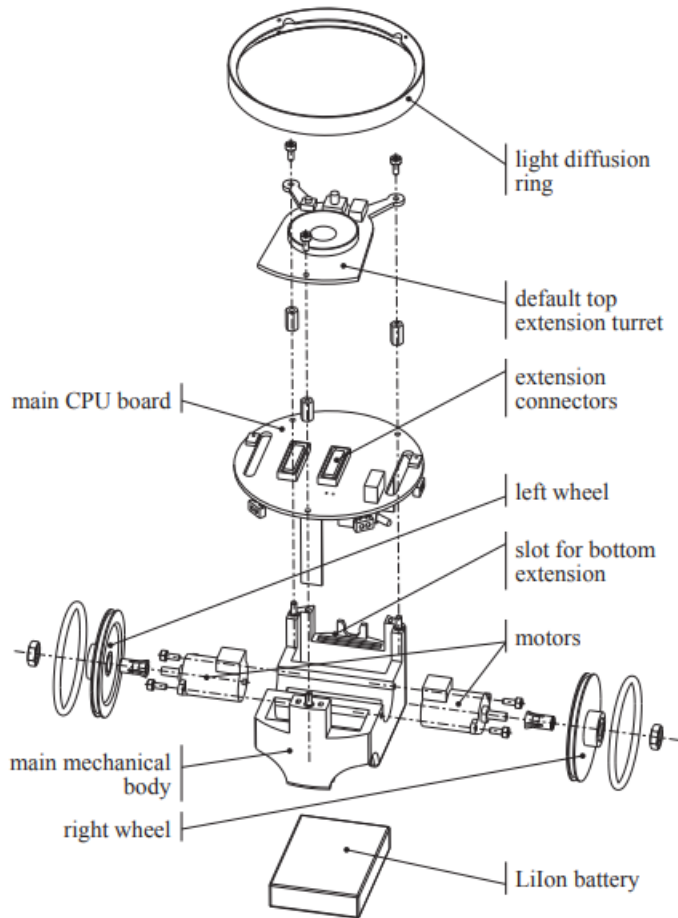


Figura 10: Estructura mecánica de E-puck [8].

2.6. Kobot

Kobot es un robot de manejo-diferencial, con estructura circular de 120 mm de diámetro, pesa 350 gramos con baterías. La estructura se compone de dos partes principales: Una base cilíndrica que aloja los motores, las baterías y la placa de sensores de corto alcance; Una tapa cilíndrica que cubre la estructura. Ambas partes están hechas de fundición de poliuretanos, un material de baja densidad que es ideal para crear estructuras de peso ligero y alta durabilidad. Se utilizaron motores DC de alto torque con cabezal de engranajes que van conectados a las llantas. Se utilizó el controlador de motores Si9988. [9]

El diseño del sistema de Kobot se muestra en la Figura 12. El principal controlador es un PIC16F877A configurado a 20 MHz, que implementa el algoritmo de control que determina el comportamiento del robot. En su básica versión, Kobot viene con un sub sistema a base de 8 sensores IR de corto alcance mostrado en la Figura 11.b y 11.c. Los sensores se encuentran separados a 45°. [9]

Kobot utiliza el módulo XBee Maxstream con el protocolo IEEE802.15.4/ZigBee, que provee una red de baja potencia y soporta comunicación punto a punto, punto a multipunto y

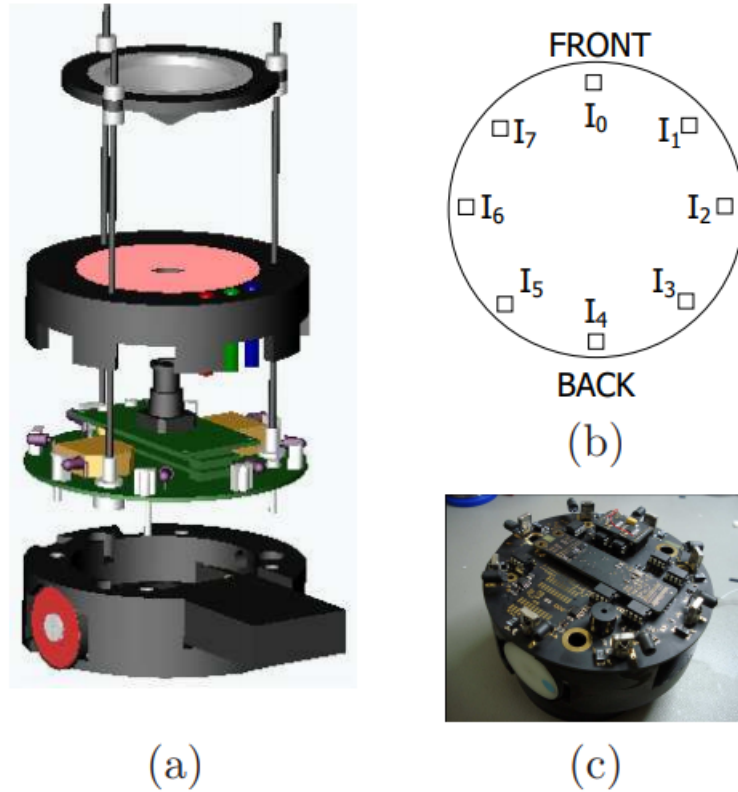


Figura 11: (a) Simulación de Kobot equipado con el sistema opcional de visión omni-direccional. (b) Posición de los sensores IR y su número. (c) Versión básica de Kobot con el controlador principal y la tarjeta de comunicación inalámbrica en lo alto [9].

par a par. Este protocolo soporta 65536 direcciones de módulos, y utilizando el broadcasting se puede implementar la programación de múltiples robots de forma inalámbrica. [9]

Esta plataforma también tiene visión omnidireccional que se observa en la Figura 11.a, compuesto por una cámara a color orientada a un espejo omnidireccional que está puesto en la parte superior del robot. Para controlar esta parte se utiliza un microprocesador PXA255 a 200 MHz, que utiliza el sistema operativo Linux con una memoria EEPROM de 32 MB y una RAM de 32 MB. [9]

Utilizando tres baterías AA NiMH de 2700 mAh cada una Kobot puede operar durante 7.5 horas, utilizando una batería Li-Po de una celda de 2000 mAh opera durante 10 horas. [9]

2.7. Kilobot

Kilobot es un robot de bajo costo diseñado para realizar pruebas de algoritmos colectivos en centenas o miles de robots accesibles para los investigadores. Cada robot cuesta \$ 14 y toma 5 minutos de armar. Esta plataforma es de fácil operación para el usuario, así como también la programación y la carga de batería de todos los robots.

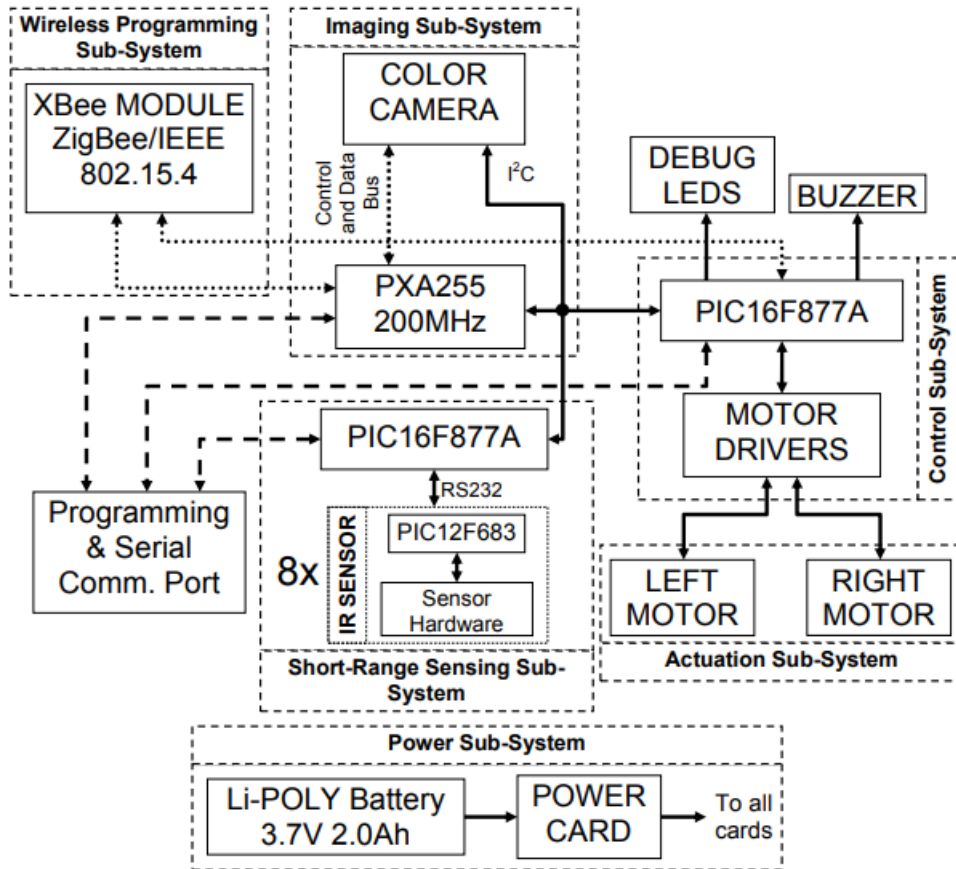


Figura 12: Diagrama de sistema de control de Kobot [9].

SDASH es un algoritmo desarrollado para auto ensamblarse en una sombra colectiva y utilizado para el diseño de hardware de Kilobot. Por lo tanto, esta plataforma puede moverse hacia adelante, rotar, comunicarse con robots vecinos cercanos, medir la distancia de vecinos cercanos y tener suficiente memoria para correr el algoritmo. [10]

Kilobot utiliza dos motores vibratorios para moverse y tres soportes rígidos. Cuando un motor vibra, la fuerza centrípeta generada se convierte en una fuerza que mueve al robot. Este efecto se llama movimiento slip-stick. Cuando solo un motor está vibrando, el robot gira en un sentido, cuando solo el otro motor vibra, el robot gira en el sentido contrario. Al manejar la cantidad de vibración de los motores se puede controlar que el robot avance hacia adelante girando en contra de las manecillas del reloj a la vez. Kilobot puede moverse a 1 cm/seg y rotar a 45°/seg. [10]

Para comunicarse con robots vecinos, Kilobot utiliza un LED IR y un fotodiodo IR, ubicados justo en el centro y en la parte inferior de la PCB, mostrado en la Figura 13.d. [10]

El controlador que utiliza Kilobot es un Atmega328, que se encarga de manejar los robots, la comunicación, circuito de potencia, el LED RGB, y ejecuta el programa del usuario que define el comportamiento del robot. El controlador opera a 8 MHz y tiene 32 KB de memoria, se utilizan los módulos PWM para controlar la vibración de los motores, y ADC's de 10 bits para medir la intensidad de luz infra-roja detectada. Toda la programación se desarrolló en

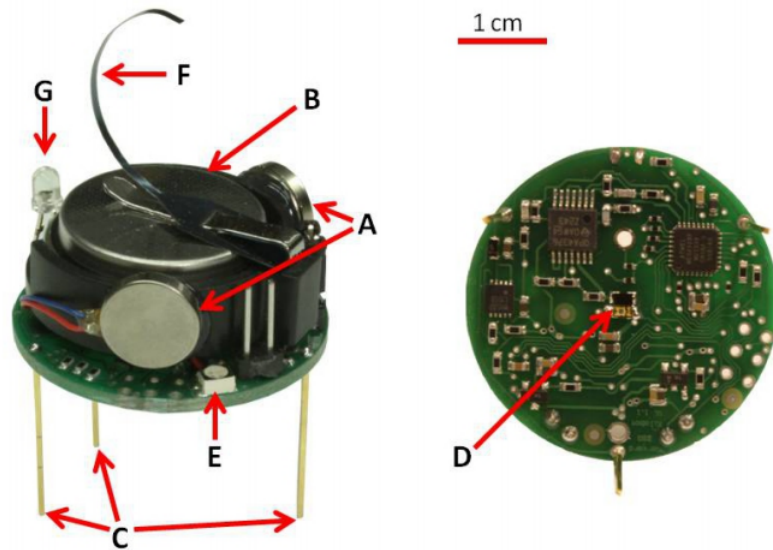


Figura 13: Vista isométrica (izquierda) e inferior (derecha) de Kilobot. (A) Motores de vibración. (B) Batería de Litio-Ion. (C) Patas rígidas de soporte. (D) Trascceptor IR. (E) LED RGB. (F) Terminal de carga. (G) Sensor de luz [10].

lenguaje C. [10]

Cada plataforma Kilobot tiene una batería de litio-ion de 3.4V y 160mAh, que provee una operación de 3-24 horas dependiendo de la actividad del robot. Tiene tres reguladores de voltaje y un cargador conectados a la batería, utiliza dos reguladores para las llantas y la comunicación y el tercero es para el microcontrolador. [10]

2.8. R-One

Los tres principales objetivos del desarrollo de R-one son la docencia, alcance e investigación. De esta forma, R-one, se centró en ser una plataforma de bajo costo, diseño avanzado y fácil uso. Este robot con todas sus partes ensambladas cuesta \$ 247. [11]

R-one posee un chasis de manejo diferencial, con dos encoders, uno por llanta, tres sensores de luz, ocho sensores de proximidad, un giroscopio de tres ejes y un acelerómetro de tres ejes. Tiene un sistema de comunicación IR entre robots, que también mide el rumbo y orientación de los robots vecinos. La interfaz con el usuario consiste en tres botones, un arreglo de LED's, un localizador IR y una radio de 2 Mbps. Tiene una autonomía de cuatro horas, la batería es recargable a través de un puerto USB. Utiliza un microcontrolador ARM de 32 bits, operando a 50 MHz con 256 KB de memoria flash y 64 KB de SRAM. La plataforma mide 10 cm de diámetro y pesa 230 gramos. La velocidad máxima del robot es de 30 cm/seg. [11]

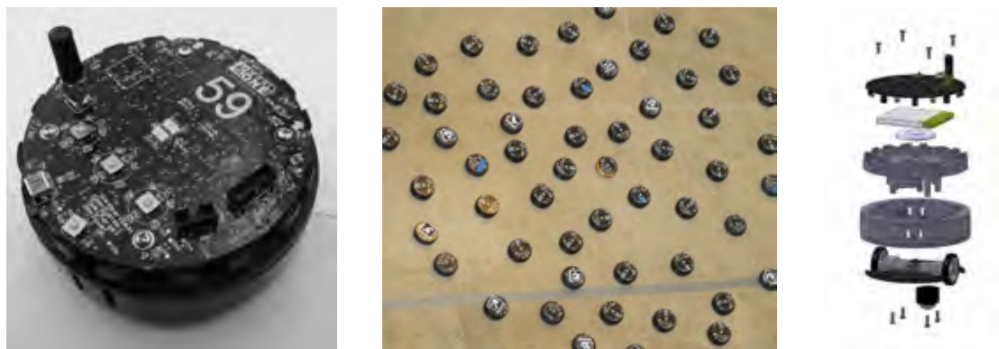


Figura 14: (Izquierda) R-One robot de bajo costo para investigación, docencia y alcance. (Medio) Este tipo de sistema puede conectar la brecha entre la simulación e implementación de control multi robots en un aula. (Derecha) Vista simulada en software de R-one [11].

2.9. Colias

Colias es un micro robot autónomo, de bajo costo y de plataforma abierta para aplicaciones en robótica de enjambre. Este robot utiliza una plataforma circular de 4 cm de diámetro. Tiene una velocidad máxima de 35cm/s. Utiliza módulos infra-rojos de largo alcance con potencia regulable para comunicarse con robots vecinos en un rango de 0.5cm a 2m. Colias se diseñó como una plataforma completa que soporta herramientas de desarrollo de software para robots para investigación y educación. [12]

Robot	Cost	Sensor	Motion/Speed	Size	Autonomy
Colias	€25	distance, light, bump, bearing, range	wheel, 35cm/s	4cm	1-3h
AMiR	€65	distance, light, bearing	wheel, 10cm/s	6.5cm	2h
Alice	N/A	distance, camera	wheel, 4cm/s	2.2	10h
Jasmine	€80	distance, light, bearing	wheel, N/A	3cm	1-2h
E-puck	€580	distance, camera, bearing, accel, mic	wheel, 13cm/s	7.5cm	1-10h
Kobot	€800	distance, bearing, vision, compass	wheel, N/A	12cm	10h
Kilobot	€75	distance, light	vibration, 1cm/s	3.3cm	3-24h
R-one	€220	light, IR, gyro, bump, accelerometer	wheel, 30cm/s	10 cm	6h

Cuadro 2: Comparación de distintas plataformas para robótica de enjambre [12].

Varias plataformas de robots móviles se han desarrollado previamente en el estudio de aplicaciones de enjambres. Estas se muestran en el Cuadro 2. Alice es uno de esos robots de enjambres construido con un tamaño de paquete muy pequeño; se ha empleado en diversas aplicaciones de investigación de enjambres, como la incorporación de la agregación de cucarachas. La última versión de Alice está equipada con sensores de proximidad y receptores remotos de infrarrojos (IR). Se puede comunicar con su vecino directo utilizando los sensores IR a distancias cortas. La Alice comercializada anteriormente costaba alrededor de unos cientos de libras. Otro micro-robot que ha sido ampliamente utilizado en la robótica de enjambres es Jasmine. Este robot utiliza seis sensores IR para detectar obstáculos y robots en las proximidades. Jasmine ha desempeñado el papel de una abeja en varios escenarios de agregación (BEECLUST). E-puck es uno de los robots más exitosos, está diseñado princi-

palmente para la educación en el campo de la ingeniería y está equipado con varios sensores. Sin embargo, la versión comercializada del e-puck básico es de €580 y se necesitan €300 adicionales para obtener un rango adicional y un módulo de soporte. Kilobot es también una plataforma de enjambre de robots con funciones escalables, como un cargador de grupo y un programador. Utiliza un principio de deslizamiento para el movimiento que reduce su costo, ya que el robot no usa motores o ruedas. Sin embargo, el método de movimiento tiene varios inconvenientes, como que la velocidad alcanzada es baja, lo que limita su aplicación en escenarios de enjambre. Su método de movimiento reduce su uso en diversas superficies. Anteriormente desarrollamos AMiR (robot autónomo en miniatura) [4] como una plataforma de hardware abierto de bajo costo para aplicaciones de enjambre. Utiliza sensores IR para estimación de distancia y comunicación de corto alcance. Se han realizado varias aplicaciones de investigación con AMiR, incluida la agregación BEECLUST y la versión extendida. Además, AMiR se simula en Player / Stage. [12]

2.9.1. Controlador

Colias utiliza dos ATMEL AVR micro-controladores en paralelo, μ_1 y μ_2 (Ver Figura 16). Las funciones de control de potencia, control de movimiento y detección de obstáculos son manejados por μ_1 . μ_2 controla la comunicación entre robots y los escenarios programados por el usuario. Las prioridades de los micro controladores son programables, las tareas de prioridad alta son manejadas en μ_1 .

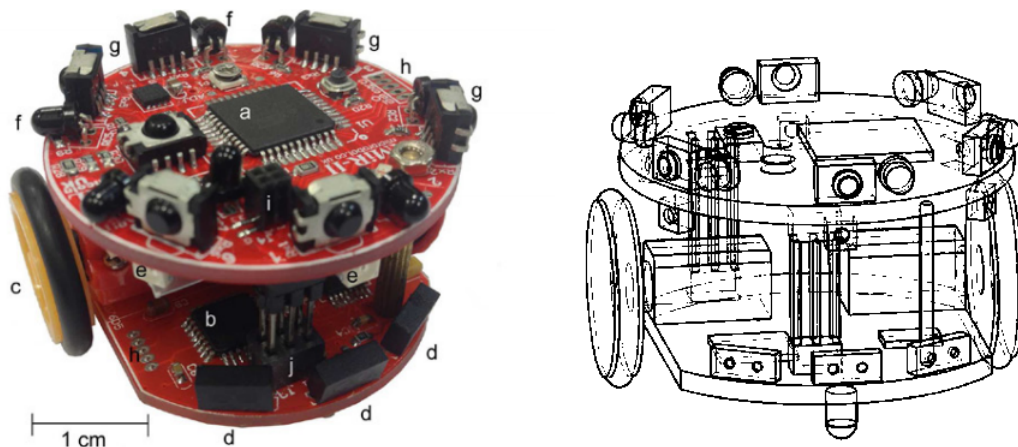


Figura 15: (Izquierda) Colias a) Procesador principal, b) segundo procesador para control de movimiento y potencia, c) Llantas de 2.2cm de diámetro, d) sensores de proximidad, e) Transmisores IR, f) Foto-transistores IR, g) Decodificadores IR, h) Puerto SPI, i) RS-232, j) I^2C y conexiones paralelas. (Derecha) Robot 3D simulado en software [12].

2.9.2. Movimiento

Colias utiliza dos motores DC con engranajes directos anclados a dos ruedas de 2.2cm de diámetro, cada rueda tiene una velocidad máxima de 35cm/s. La velocidad rotacional de cada motor es controlada de forma individual utilizando la técnica de PWM. Cada motor es controlado de forma separada utilizando un controlador de puente-H para motores DC y

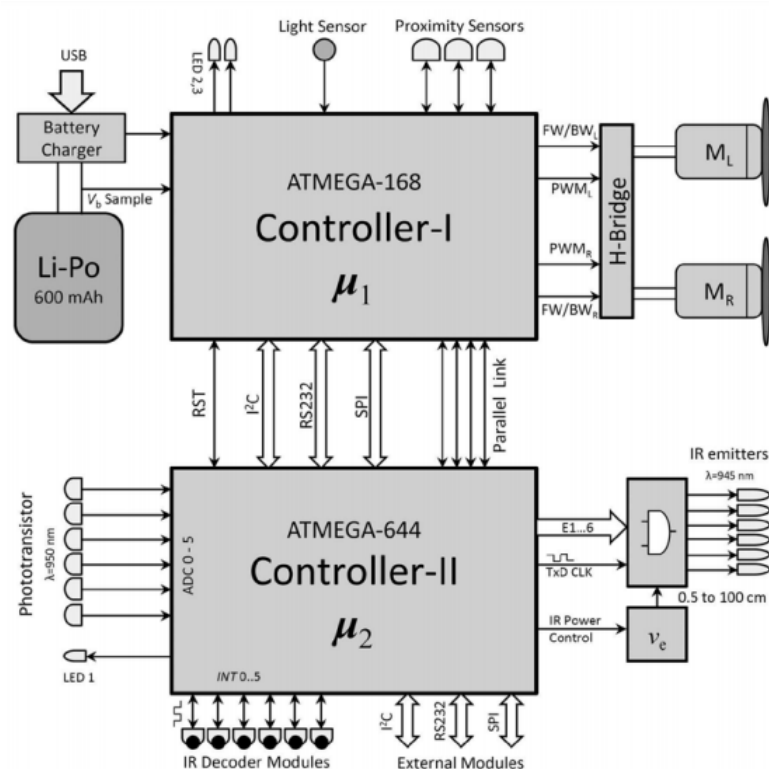


Figura 16: Arquitectura básica de Colias. Utiliza dos microcontroladores en paralelo para monitorear todas sus funciones [12].

consume una corriente promedio de 35 ± 5 mA sin carga alguna, y 150 ± 20 mA con la carga del robot. El robot utiliza la configuración de manejo diferencial, que es un simple método de control utilizando un principio básico de control de movimiento. [12]

2.9.3. Sistema de sensores

Se utilizan sensores de proximidad IR para evitar obstáculos y colisiones con otros robots. Colias también tiene sensor de luz para registrar la iluminación del ambiente. El sistema de sensores IR consisten en dos diferentes tipos de módulos IR, llamados sensores de corto alcance (Figura 15.d) y sensores de largo alcance (Figura 15.e y 15.f). La combinación de tres sensores de corto alcance sirven para detectar objetos que se estén acercando al robot. La combinación de seis sensores de largo alcance (con separación de 60° entre cada uno) se utilizan para la detección de objetos y robots, asimismo, para su diferenciación. [12]

2.9.4. Control de potencia

En Colias, la placa inferior es la encargada del control de potencia y del proceso de carga de batería. El robot utiliza 560mA con un movimiento normal (En una superficie plana) con una comunicación de corto alcance. Se puede reducir a 200mA cuando los emisores se encienden ocasionalmente y el robot se mueve a una rápida velocidad. Se utiliza una batería

de litio-polímero (Li-Po) de 3.7V, 600mAh que provee una autonomía al robot de tres horas. [12]

Para la carga de la batería se utiliza un cargador lineal externo LTC4054-4.2 de corriente y voltaje constante.

2.10. Plataforma robótica para aplicaciones en robótica de enjambre fase 1

Esta plataforma fue desarrollada en la Universidad del Valle de Guatemala en el año 2017 para aplicaciones en robótica de enjambre, como proyecto de graduación en su primera fase. [1]

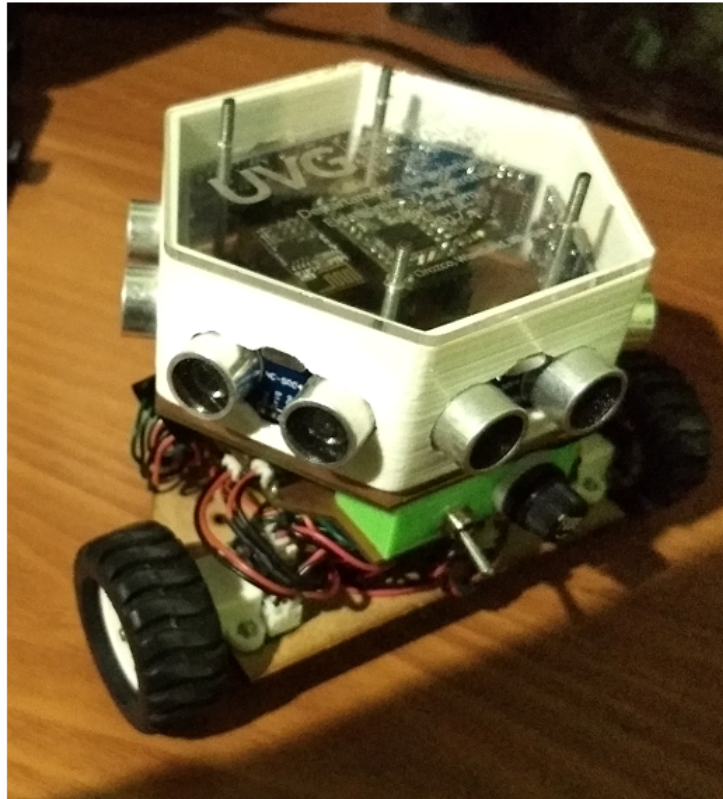


Figura 17: Vista final del robot para aplicaciones en robótica de enjambre de la fase 1 de la Universidad del Valle de Guatemala [1].

Se diseñó con una falda de 6 sensores ultrasónicos HC-SR04 para detectar objetos alrededor de la plataforma posicionados a 60° entre cada uno de ellos, estos 6 sensores se controlan en una placa PCB por separado de la principal, que es un decodificador de 3 a 8 líneas que permite controlar el *Trigger* de todos los sensores, utilizando únicamente 3 pines de salida digitales de un microcontrolador y utiliza lógica combinacional para que el *Echo* de los sensores llegue a un solo pin de entrada digital. [1]

La unidad de control es un microcontrolador Teensy 3.2 basado en ARM Cortex con

256 KB de memoria flash, 256 KB de memoria caché, 64 KB de RAM que se programa en lenguaje de arduino. Tiene una IMU, que se compone de un giroscopio y acelerómetro, ambos de tres ejes. Su movimiento es diferencial, es decir, con dos llantas y un motor anclado en cada una de ellas y controladas de forma independiente, utilizando un ball caster como apoyo. Se utiliza un controlador de motores DC doble H DRV8833 para el manejo de las llantas con señales PWM. Tiene un encoder instalado en cada llanta para obtener datos de la velocidad, distancia recorrida y sentido en el que gira cada una. Utiliza un módulo ESP-8266 para la comunicación Wi-Fi con un ordenador. [1]

En [1] tienen como recomendaciones probar un nuevo diseño de la falda de sensores porque a la hora de realizar las lecturas devolvía varios valores en *timeout*, por otro lado, denotan que se utilizó Teensy por la amplia documentación que existe pero que se recomienda realizar pruebas con otro microcontrolador para obtener mejor desempeño. Los encoders utilizados no dieron problema con las mediciones con la única observación que pueden descalibrarse, para solucionar este problema basta con volverlos a calibrar observando que las señales de salida sean acorde a lo esperado, por lo tanto recomiendan dejar pines o terminales accesibles para medir dichas señales en un futuro modelo de este robot.

2.10.1. Partes del robot

Las partes con las que se integra esta Fase 1 se muestran en el Cuadro 3.

Partes del robot	Función
6 sensores ultrasónicos HC-SR04	Detección de objetos alrededor de la plataforma.
5 IC 74HC138	Demultiplexores para controlar la falda de sensores ultrasónicos solo con 3 pines de salida y 1 de entrada.
Módulo ESP-8266	Comunicación Wi-Fi con un ordenador.
2 llantas de goma y un ball caster	Llantas para movimiento y ball caster para apoyo de la plataforma.
2 Motores DC	Actuadores anclados a cada llanta.
2 Encoders	Sensores para medir velocidad, distancia y sentido de giro de cada llanta.
Controlador DRV8833	Controlador de puente H para dos motores.
Teensy 3.2	Unidad de control de la plataforma.
Batería Li-Po de 3.7V	Fuente de alimentación de la plataforma.
Estructura mecánica	Se utilizó MDF, PLA, y tornillos que alojan las partes y el PCB del robot.

Cuadro 3: Partes de la plataforma para robótica de enjambre de UVG Fase 1. [1]

Falda de sensores ultrasónicos

Se utilizaron seis sensores ultrasónicos HC-SR04 para formar una falda sensorial de objetos alrededor de la plataforma. Los sensores están posicionados a 60° entre cada uno de

ellos como se observa en la Figura 23. Utilizando el IC 74HC138 se desarrolló una placa PCB por separado de la placa principal para controlar los 6 sensores ultrasónicos, esta placa consiste en un decodificador de 3 a 8 líneas que permite controlar el *Trigger* de todos los sensores, utilizando únicamente 3 pines de salida digitales de un microcontrolador y utiliza lógica combinacional para que el *Echo* de los sensores llegue a un solo pin de entrada digital.

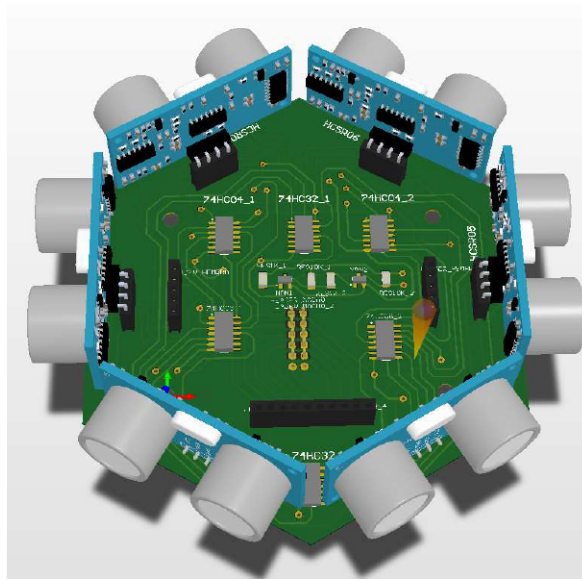


Figura 18: Diseño CAD de falda de 6 sensores ultrasónicos junto con placa PCB de control de estos sensores [1].

Comunicación inalámbrica

Para esta plataforma se consideraron los módulos ESP-8266 para comunicación Wi-Fi y HC-05 para comunicación Bluetooth, escogiendo el módulo ESP-8266, tomando en cuenta que este módulo ofrece un baudrate de hasta 5 Mbps mientras que el módulo HC-05 ofrece solamente 9600 bps. Por otro lado, para controlar su robot es necesario la comunicación de una alta cantidad de comandos a alta velocidad, por lo que la seguridad de comunicación de información debía ser alta, de esta manera, el módulo ESP-8266 es favorito en este aspecto gracias a que la conexión Wi-Fi ofrece sistemas de revisión de errores y protección de datos de 32 bits mientras que los sistemas Bluetooth no ofrecen estos métodos y solo son capaces de manejar 16 bits. Con el módulo ESP-8266 se tiene la flexibilidad de trabajar con los protocolos TCP o UDP. Por otro lado, el precio del módulo HC-05 se encontraba por \$ 5 en la mayoría de tiendas y distribuidoras en línea mientras que el ESP-8266 estaba valorado en tan solo \$ 3. [1]

Con el módulo ESP-8266 se pueden utilizar los protocolos TCP y UDP. Ambos se encargan del envío y recepción de datos a través de aplicaciones, la diferencia radica en que TCP ofrece estabilidad y seguridad de transmisión sobre velocidad, mientras que UDP se enfoca en una transmisión veloz pero sujeta a pérdida de paquetes. Esto se ve notablemente en transmisiones de multimedia donde UDP no retransmite paquetes perdidos y causa una notable pérdida en la calidad de la imagen mientras que TCP maneja una imagen de alta calidad, aunque su tasa de transmisión es bastante reducida. Conociendo la necesidad de

que el sistema sea tanto veloz como confiable se decidió utilizar el protocolo TCP para poder asegurar que el envío de instrucciones fuera recibido sin problemas por el equipo, aún si no se logra alcanzar la velocidad máxima posible con el módulo. Esta decisión se debe a que el equipo será utilizado como una herramienta didáctica, se consideró que era necesario establecer un método de comunicación seguro y confiable que evitara la pérdida de paquetes al máximo para que en un futuro estudiantes que utilicen la herramienta se sientan confiados en que cualquier error que pueda presentarse se puede atribuir a su diseño y no a la pérdida de información por parte del módulo de comunicación. [1]

Movimiento de la plataforma

Esta plataforma utiliza un ball caster como apoyo y para equilibrio del robot y dos motores DC con caja reductora que se alimentan con 3.3V. Cada motor está anclado a una llanta de goma, junto con el *encoder* como se muestra en la Figura 19. La llanta posee dimensiones físicas de 42 mm de diámetro y 19 mm de ancho [13]. Para controlar los motores, tanto el sentido de giro como la velocidad de cada uno, se utilizó el módulo DRV8833 para dos motores (Figura 37).



Figura 19: Ensamblaje de llanta de goma con *encoder* y motor DC [14].

Los motores que se utilizan son de 9.27 mm de largo, con caja reductora de relación 75:1, con eje de metal de forma "D" que es compatible con las llantas de 42 mm de diámetro antes mencionadas. Operan en un rango entre 1V y 6V, gira a 290 RPM a 6V, utiliza 70mA de corriente y ofrece un torque de 0.8 kg/cm a 6V. Estos tienen un costo unitario de \$ 11.50. [15]

Medición de velocidad, distancia recorrida y sentido de giro

Esta plataforma utiliza 2 *encoders* SEN0116 de DFRobot para medir la velocidad, distancia recorrida y sentido de giro de cada llanta. Como se determinó en la sección 4.6 del Marco Teórico, la forma en que funcionan los *encoders* SEN0116 es que cada uno devuelve dos señales cuadradas desfasadas 90°, y dependiendo de qué señal cambie su estado de LOW a HIGH primero, se puede determinar el sentido de giro de la llanta. En la Figura 20 se

observa las señales que devuelven. El periodo de cualquiera de las dos señales cuadradas indica el tiempo que la llanta giró de un pin a otro. La llanta posee 12 pines internos que son los que provocan los pulsos de los *encoders*. Por lo tanto, para calcular la distancia recorrida basta con utilizar la ecuación 4.

$$D = N \frac{d * \pi}{12} \quad (4)$$

Donde D es la distancia recorrida, N es la cantidad de pulsos devueltos por los *encoders*, d es el diámetro de la llanta, que en este caso son 42 mm. Para calcular la velocidad se utiliza la ecuación 5.

$$V = \frac{d\pi}{12 * t} \quad (5)$$

Donde V es la velocidad de la llanta, d es el diámetro de la llanta (42 mm) y t es el periodo de cualquier señal devuelta por el encoder.

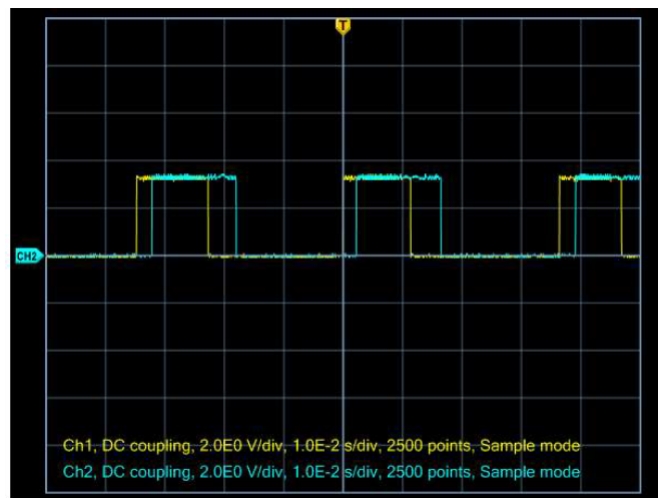


Figura 20: Señales de un *encoder* [1].

Nosotros como sociedad no podríamos estar donde estamos actualmente si nuestras acciones fueran individuales, el desarrollo que se ha logrado hasta ahora es porque se ha realizado en conjunto. En la fauna, una hormiga, una abeja, etc., por sí solas no podrían obtener resultados significativos, sin embargo, en conjunto alcanzan grandes objetivos acorde a sus capacidades. El comportamiento sistémico y en conjunto con el que trabajan las colmenas, enjambres, etc., es un campo todavía explorado vagamente. La robótica, en este caso, ofrece la posibilidad de implementar robots de bajo costo, fácil producción, que ejecuten instrucciones desde un ordenador de forma inalámbrica y que trabajen en conjunto.

Se busca implementar una nueva plataforma que ejecute las instrucciones que le sean enviadas y se comporte de forma individual o en conjunto con otros robots de su misma gama, con librerías de fácil uso desde Matlab para que cualquiera que desee implementar algoritmos de comportamiento de enjambre los simule con esta plataforma y obtenga información relevante para su estudio. Las ventajas del siguiente modelo es que es de fácil implementación, sus componentes son bajo costo y su mayoría de acceso local.

Por otro lado, ya existen este tipo de plataformas en el mundo; sin embargo, no hay ninguna desarrollada en Guatemala ni en Centro América, por lo tanto, se busca que el Departamento de Electrónica de la Universidad del Valle de Guatemala sea la primera en desarrollar este tipo de plataforma en la región.

4.1. Objetivo general

Diseñar e implementar un robot de bajo costo, dimensiones reducidas, con comunicación Wi-Fi para aplicaciones en robótica de enjambre.

4.2. Objetivos específicos

- Mejorar el diseño mecánico de la plataforma robótica ya existente desarrollada por el Departamento de Electrónica, desarrollando un diseño más compacto.
- Mejorar la plataforma de control sustituyendo el microcontrolador Teensy por una plataforma más robusta utilizando el microcontrolador PIC32MX250F128B.
- Mejorar el sistema de detección de objetos sustituyendo la red fija de sensores ultrasónicos por un radar constituido por un servomotor y dos nuevos modelos de sensores ultrasónicos.
- Controlar la plataforma desde Matlab por medio de comunicación Wi-Fi, implementando librerías desarrolladas durante el ciclo de megaproyectos del año 2017 en la Universidad del Valle de Guatemala.
- Realizar un análisis de potencia eléctrica y costo de la plataforma.

Este trabajo es la continuación de la fase 1 de la plataforma desarrollada para aplicaciones en robótica de enjambre el año 2017 en el departamento de electrónica de la Universidad del Valle de Guatemala [1], toda la documentación se encuentra en *Reingeniería de Mega-proyectos Fase 1* [1], donde se tiene como recomendaciones probar un nuevo diseño de la falda de sensores, porque a la hora de realizar las lecturas devolvía varios valores en *timeout*, por otro lado, denotan que se utilizó Teensy por la amplia documentación que existe, pero se recomienda realizar pruebas con otro microcontrolador para obtener mejor desempeño. Los *encoders* utilizados no dieron problema con las mediciones con la única observación que pueden descalibrarse, para solucionar este problema basta con volverlos a calibrar observando que las señales de salida sean acorde a lo esperado, por lo tanto, recomiendan dejar pines o terminales con fácil acceso para medir dichas señales en un futuro modelo de este robot. De esta forma, las dos áreas de mejora a tomar en cuenta para esta segunda fase es la unidad de control y el sistema de detección de objetos alrededor de la plataforma.

El nuevo modelo de robot para aplicaciones en robótica de enjambre se centra en la mejora del diseño mecánico de la plataforma, realizando un nuevo diseño más compacto en las dimensiones físicas respecto al modelo de la Fase 1; en la mejora de la plataforma de control utilizada, sustituyendo el microcontrolador Teensy por una plataforma más robusta utilizando el microcontrolador PIC32MX250F128B; en la mejora del sistema de detección de objetos sustituyendo la red fija de sensores ultrasónicos por un nuevo sistema constituido por un servomotor y dos nuevos modelos de sensores ultrasónicos; y en el uso de la plataforma desde Matlab por medio de comunicación Wi-Fi, implementado las librerías de la fase anterior.

La nueva plataforma tendrá un sistema de movimiento del tipo *differential drive*, un sistema de detección conformado por un servomotor y dos sensores ultrasónicos que pueda realizar un barrido de detección en el plano "XY", un sistema de medición que detecte la velocidad, sentido de giro y distancia recorrida de cada llanta utilizando *encoders*, un sistema de comunicación que transmite y reciba la información de un ordenador vía Wi-Fi, un sistema de potencia que brinde el voltaje y la corriente necesaria para que todos los

sistemas funcionen correctamente. Se desarrollará una librería en Matlab con la que se podrá modificar la velocidad y sentido de giro de las llantas de la nueva plataforma, que active el sistema de detección, que solicite el tiempo que tarda un *tick* de cada llanta, que solicite el sentido de giro actual y cuántos *tick's* ha recorrido el robot por cada llanta, en donde un *tick* es la distancia mínima recorrida y detectada por los *encoders* de las llantas, donde cada *tick* representa $1/12$ de la circunferencia de la llanta.

6.1. Comunicación digital inalámbrica

Estos sistemas operan su transmisión de datos a través del espacio y no con conexiones de cables físicos. Esto posee la ventaja de permitir a los usuarios mandar o recibir información desde prácticamente cualquier lugar y aún en movimiento. Los canales inalámbricos operan con radiación electromagnética desde el transmisor hasta el receptor. [16]

6.1.1. Capas de arquitectura TCP/IP

Este modelo está conformado por cinco capas diferentes cada una dedicada a uno de los conjuntos de tareas necesarias para lograr la interconexión de diversos dispositivos electrónicos. Las cinco capas son: capa física, encargada de definir las características del equipo conectado a la red; capa de enlace de datos, encargada de la transferencia de información a través de los medios en la red; capa de red, encargada del direccionamiento de datos y la entrega de los mismos a través de distintas redes; capa de transporte, la cual maneja la transferencia de información y asegura que lo enviado sea igual a lo recibido en las redes; capa de aplicación, se encarga de la interacción con el usuario y define los servicios estándares de internet que cualquiera puede utilizar. [1]

Un apretón de manos de tres vías es un algoritmo de seguridad aplicado para asegurar una conexión confiable entre dos equipos en una red. Este consiste en el intercambio de tres señales claves entre ambos equipos, *SYN*, *SYN - ACK* y *ACK*. El equipo que desea entablar una conexión envía la señal de *SYN* al servidor, la cual consiste en un número aleatorio que denominaremos *A*. El servidor responde a través de una señal *SYN - ACK* que contiene el número *A* recibido más uno, $A + 1$, además de un segundo número aleatorio que denominaremos *B*. Para finalizar el cliente confirma su conexión con el servidor enviando una señal *ACK*, que contiene nuestro número $A + 1$ y el segundo número $B + 1$ indicando así que la conexión es segura. [1]

OSI Ref. Layer No.	OSI Layer Equivalent	TCP/IP Layer	TCP/IP Protocol Examples
5,6,7	Application, Session, Presentation	Application	NFS, NIS+, telnet, ftp, rlogin, rsh, rcp, RIP, RDISC, SNMP, and others
4	Transport	Transport	TCP, UDP
3	Network	Internet	IP, ARP, ICMP
2	Data Link	Data Link	PPP, IEEE 802.2
1	Physical	Physical Network	Ethernet(IEEE 802.3) Token Ring, RS-232, others

Cuadro 4: Modelo de arquitectura TCP/IP [1].

Además de esto, el protocolo TCP utiliza una secuencia de números para identificar cada byte de información transmitido, asegurando que pueda ser reconstruido en orden por el equipo que lo recibe o que una pérdida de información pueda ser detectada por el mismo. Cualquier información no recibida es retransmitida hasta que el equipo indica que la obtuvo. Asimismo, el protocolo utiliza algoritmos de control de flujo y congestión para enviar los datos a un ritmo moderado previniendo problemas de conexión. Todas estas características hacen de TCP el protocolo más confiable en el envío de datos, pero terminan representando una pérdida en la velocidad de transmisión de la información.[1]

El segundo protocolo que conforma la capa de transporte es el de datagrama de usuario (UDP por sus siglas en inglés). Al igual que el protocolo TCP, UDP se encarga de manejar el envío de datos de una aplicación a otra comunicándose a través de una red IP. A diferencia de TCP, UDP enfatiza la velocidad y la eficiencia sobre la confiabilidad de entregas removiendo muchos de los componentes de seguridad esenciales de TCP. Al remover los servicios de análisis de congestión de red, el apretón de manos de tres vías y los métodos de identificación de paquetes perdidos UDP logra mejorar considerablemente la velocidad de transmisión de un sistema. Esto hace que UDP sea sumamente efectivo en aplicaciones como transmisión de multimedia en tiempo real, donde se necesita una respuesta alta y la pérdida de paquetes no presenta un problema grave. [1]

6.1.2. Wi-Fi

Los protocolos UDP y TCP se encargan de enviar información, sin embargo, es necesario un estándar para definir la manera de entablar una comunicación entre dispositivos. *Wireless Fidelity* (Wi-Fi) es el nombre comercial de la tecnología de red de área local inalámbrica utilizada por múltiples equipos electrónicos que se basa en el estándar IEEE 802.11. Este cuenta con dos modos esenciales de operación: *Ad hoc* y modo de infraestructura. En modo *ad hoc* los equipos son capaces de conectarse directamente entre sí sin necesidad de un equipo intermediario permitiendo así una transmisión de información directa. El modo más común de esta tecnología es el de infraestructura, en el que una estación base funciona como un punto central de conexión con una unión física a la red a través de un cable de fibra óptica o

similar. La estación base se encarga de asignar direcciones IP identificadoras a cada equipo conectado a la red y ayuda a entablar la conexión entre los equipos con la red, así como conexiones entre pares.

6.1.3. Módulo ESP8266

El Módulo WiFi ESP8266 es un SOC autónomo con una pila de protocolos TCP / IP integrados que le permite a cualquier microcontrolador acceder a su red WiFi. El ESP8266 es capaz de alojar una aplicación o descargar todas las funciones de red Wi-Fi de otro procesador de aplicaciones [17]. Este módulo trabaja a base de instrucciones AT transmitidas a través de una conexión serial [1]. ESP8266 tiene integrado el protocolo TCP/IP, conexión Wi-Fi directa punto a punto y punto de acceso suave, 1 MB de memoria Flash, interfaz UART que funciona con un baud rate de hasta 5 MB. Utiliza una corriente nominal de 60 mA. Su costo es de \$ 3. [17]

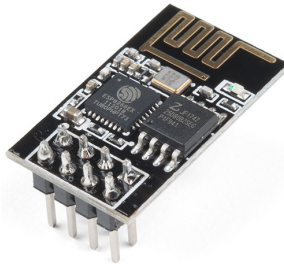


Figura 21: Módulo ESP-8266 de Sparkfun [17].

6.2. Microprocesadores y microcontroladores

Un microprocesador es un circuito integrado que consiste de una unidad lógica y aritmética (ALU), registros de propósito general, stack pointer (SP), program counter (PC), circuito de reloj y circuitos de interrupciones. Su función es ejecutar instrucciones y procesar la información de sus entradas. La ALU, el circuito de reloj y los circuitos de interrupciones son componentes electrónicos interconectados de manera que forman un circuito que realiza una función específica. El *stack pointer*, el *program counter* y los registros de propósito general son espacios de memoria implementados en el silicio del procesador; es memoria rápida, pero de alto costo, y es de corta extensión comparada con memorias externas como la RAM. [1]

Un entorno de microcomputadora no es posible usando solo un microprocesador, se requieren de otros dispositivos periféricos como memoria ROM (*read only memory*), memoria de lectura y escritura RAM (*random access memory*), decodificadores, drivers y dispositivos de entrada y salida externos además del microprocesador. También se puede incluir controladores de interrupciones y temporizadores programables. Una característica importante de una computadora basada en un microprocesador es la flexibilidad que este diseño brinda, ya que se le pueden añadir más módulos periféricos para escalar el sistema. [1]

Un microcontrolador incorpora las características de un microprocesador, así como ROM, RAM, entradas y salidas paralelas, entradas y salidas seriales, contadores y circuito de

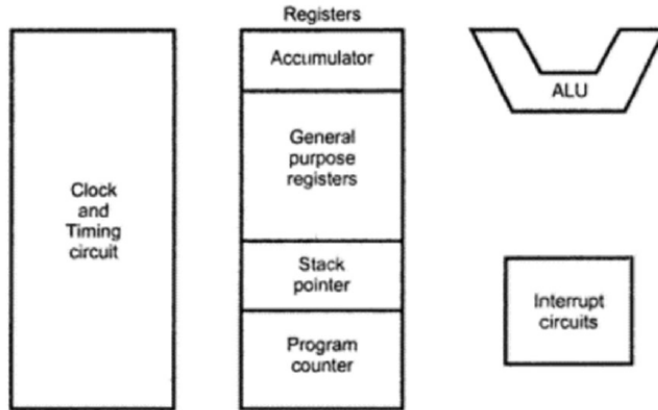


Figura 22: Diagrama de bloques general de un microprocesador [1].

reloj, todo en un mismo circuito integrado. Este dispositivo no ofrece la escalabilidad de un microprocesador, pero ofrece menor tiempo de acceso entre los dispositivos periféricos y el procesador, lo que implica más velocidad. Todo esto lo ofrece un microcontrolador en varios tipos de empaquetados que son bastantes pequeños, lo que reduce el tamaño del PCB. [1]

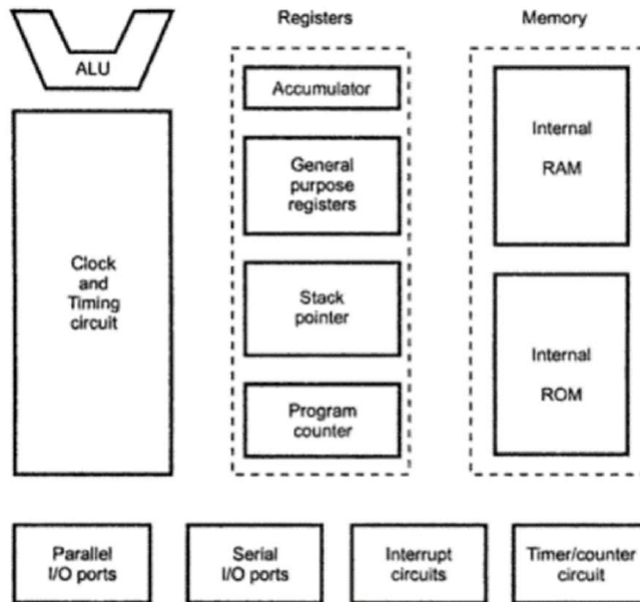


Figura 23: Diagrama de bloques general de un microcontrolador [1].

6.2.1. Teensy

Teensy es un sistema de desarrollo de proyectos con un footprint de dimensiones relativamente pequeñas. Toda la programación se hace a través del puerto USB. Hay Teensy de 8 bits, que contienen microcontroladores AVR (Teensy 2.0, Teensy++ 2.0). También existen Teensy de 32 bits, que contienen microcontroladores ARM Cortex de 32 bits. Estas plataformas son conocidas por ser de tercera generación (Teensy 3.0, Teensy 3.1, Teensy 3.2,

Teensy 3.5, Teensy 3.6, Teensy LC). Cada una de ellas difiere en el microcontrolador que utiliza. El proyecto PJRC, que incluye a Teensy, ha sido desarrollado gracias al trabajo de Paul Stoffregen, ingeniero eléctrico. En el trabajo la acompaña Robin Coon. Existe amplia documentación sobre esta plataforma, existe un foro donde se plantean problemas y Paul les da seguimiento. Teensy es compatible con el IDE de Arduino, la programación se puede hacer en ese entorno. [1]

Teensy LC

El Teensy LC forma parte de la tercera generación de estas plataformas de desarrollo de proyectos. Utiliza el microcontrolador de Freescale MKL26Z64VFT4, basado en ARM Cortex M0+. El procesador es de 32 bits, pero soporta menos instrucciones que los procesadores más potentes de ARM. El compilador traduce las operaciones a instrucciones de un ciclo de reloj, pero puede ser que, por tener un set de instrucciones reducido, las instrucciones necesarias sean más y tome más de un ciclo de reloj ejecutar una operación compleja. [1]

El procesador funciona a 48MHz. Además, el Teensy LC implementa memoria flash de 62 kBytes, cache de 64 Bytes, RAM de 8 kBytes, 24 pines de entrada y salida digitales, opera a 3.3V, 18 interrupciones, 13 entradas analógicas, entradas táctiles, una salida analógica con 12 bits de resolución, 7 timers, 10 salidas PWM, comunicación USB 2.0, 3 puertos seriales, 2 interfaces SPI, 2 interfaces I2C, entre otras características. Estas especificaciones son más que suficiente para controlar el robot que se desea construir. Su precio es de \$ 11.65 por un microcontrolador de 32 bits. [1]

Teensy 3.2

El Teensy 3.2 forma parte de la tercera generación de placas de desarrollo de Teensy. Contiene un microcontrolador MK20DX256VLH7 basado en el procesador ARM Cortex M4. Posee 256 KB de memoria flash e igual cantidad de memoria caché, 64 KB de memoria RAM, 24 pines digitales de entrada y salida, opera a 3.3V, aunque sus entradas también toleran 5V (diferente del Teensy LC). Tiene 21 entradas analógicas, una salida analógica de 12 bits de resolución, 12 timers, 12 salidas PWM, comunicación USB 2.0, 3 puertos seriales, 1 interfaz SPI, 2 interfaces I2C y una interfaz de audio digital. El núcleo funciona a 72 MHz, con capacidad de overclock de 96 MHz. Además, posee una unidad DSP (Digital Signal Processing), que es utilizada en aplicaciones de audio. Es capaz de hacer operaciones de multiplicación y división, que pueden devolver resultados de 64 bits. En otras palabras, el Teensy 3.2 es una placa con un microcontrolador más potente que el que incluye el Teensy LC. Su precio es de \$ 19.80. [1]

6.2.2. Microcontrolador de 32 bits

Un microcontrolador de 32 bits es una pequeña computadora en un circuito integrado por memoria, procesador y periféricos de entrada y salida programables. Hay varios tipos de microcontroladores de 32 bits que se caracterizan por el tamaño de la RAM, tamaño de la

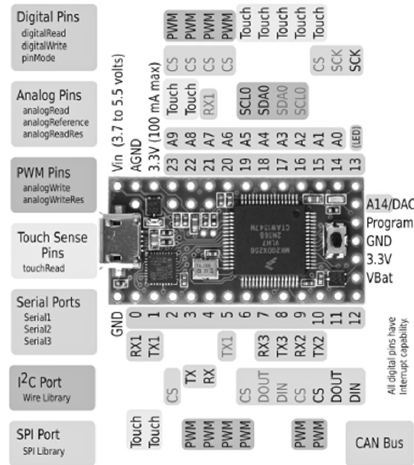


Figura 24: Teensy 3.2 con ubicación de pines [1].

Flash, número de pines de entradas y salidas, velocidad, condiciones de operación, periféricos y tipo de empaquetado. [18]

PIC32MX250F128B

Este microcontrolador con tecnología de 32 bits opera a una frecuencia de 50MHz, con fuente de voltaje entre 2.3V y 3.6V, consume 0.5 mA por MHz. Posee un PLL programable y varias fuentes de osciladores de reloj. Memoria Flash de 256Kb y SRAM de 64Kb. Dos módulos UART con velocidad de 12.5 Mbps, 28 pines, con 19 de entrada y salida remapeables. 5 Timers, 5 módulos Capture and Compare, 5 interrupciones externas, 3 comparadores análogos. Los empaquetados disponibles son SOIC, SSOP, SPDIP y QFN. El costo de este microcontrolador es de \$ 2.76, que comparado con el precio del Teensy 3.2, es mucho más barato [3]. La mínima conexión requerida para operar el PIC32MX250F128B se observa en la Figura 26.



Figura 25: PIC32MX250F128B con empaquetado SPDIP [3].

PickIt 3

Pickit 3 [19] es un programador y debugger de Microchip que utiliza una lógica de depuración integrada a cada chip con memoria Flash para proporcionar hardware programador y debugger de bajo costo. Las ventajas de este programador es que es de bajo costo y utiliza el

de lazo cerrado, utilizando un sensor de posición como retro alimentación. [20]

6.3.1. DFRobot micro servo

Se alimenta con un voltaje entre 4.8V - 6V y utiliza un máximo de 500 mA. Posee un torque en estado estacionario de 1.6 kg/cm. Velocidad de 0.12s/60° sin carga, dimensiones físicas de 22x12.5x29.5 mm. Pesa 9 gramos. La señal de control es de PWM a una frecuencia de 50 Hz y un rango entre 1 - 2 ms posicionan el eje entre 0 y 180°. Este servomotor tiene un costo de \$ 3.5. [21]

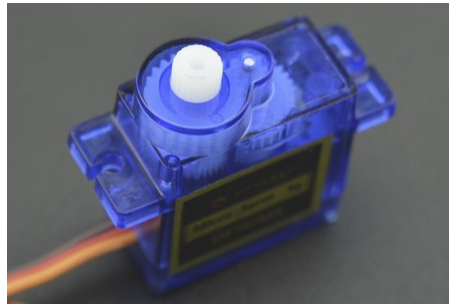


Figura 28: Micro servomotor DFRobot [21].

6.3.2. Traxxas nano servo digital

Se alimenta con un voltaje entre 3V - 6V y consume una corriente nominal de 15mA a 3.3V. Posee un torque de 0.5kg/cm. Velocidad de 0.08s/60°, dimensiones físicas de 16x8.25x20 mm. Este servomotor digital se controla con una señal PWM con frecuencia de 300Hz. Tiene un costo de \$ 4.5 [22]

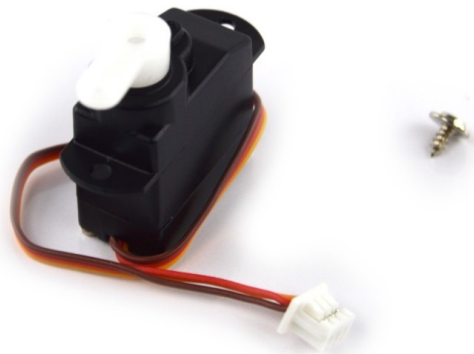


Figura 29: Nano servomotor digital Traxxas [22].

6.3.3. BMS-306DMAX digital micro servo

Se alimenta con un voltaje entre 4.8V - 6V. Posee un torque de 1.6kg/cm. Velocidad de 0.13s/60°, dimensiones físicas de 22x10x23 mm y pesa 7.1 gramos. Este servomotor digital

se controla con una señal PWM con frecuencia de 300Hz. Tiene un costo de \$ 13.63. [23]



Figura 30: BMS-306DMAX Digital Micro Servo [23].

6.4. Sensor de proximidad

Un sensor de proximidad es un sensor electrónico que puede detectar la presencia de objetos en su alrededor sin contacto físico real. Para detectar objetos, el sensor de proximidad irradia o emite un haz de radiación electromagnética, generalmente en forma de luz infrarroja, y detecta el reflejo para determinar la proximidad o distancia del objeto del sensor. [24]

Hay muchos tipos de sensores de proximidad y utilizan diferentes métodos para detectar. Por ejemplo, los sensores capacitivos y fotoeléctricos son más adecuados para objetivos orgánicos y de plástico, mientras que los sensores de proximidad inductivos solo pueden detectar objetivos metálicos. [24]

Los distintos tipos de sensores son: [24]

- Capacitivo.
- Inductivo.
- Efecto Doppler.
- Radar.
- Sonar.
- Efecto Hall.
- Infra rojo.
- Fococelda.

6.4.1. Sensor ultrasónico

Un sensor ultrasónico es un detector de proximidad de objetos, que funciona emitiendo ondas ultrasónicas que chocan contra un objeto (si es que lo hay) dentro del rango de detec-

ción y registrando el tiempo en que estas tardan en regresar al sensor, con esta información se puede obtener la distancia a la que se encuentra el objeto utilizando la ecuación 6.

$$L = (T * C)/2 \tag{6}$$

En dónde L es la distancia entre el sensor y el objeto, T es el tiempo registrado y C es la velocidad del sonido en el aire (343.2m/s).

Sensor ultrasónico HC-SR04

El sensor ultrasónico HC-SR04 es un módulo que mide proximidad que puede leer valores entre 2 cm y 400 cm, con una precisión de hasta 3mm. Posee 4 pines que son voltaje de alimentación, trigger, echo, y tierra, por lo tanto, se utilizan 2 pines para control. [1]

Su voltaje de operación es de 5VDC, consume una corriente de 15mA. Trabaja a una frecuencia máxima de 40 Hz, y sus mediciones son válidas en un ángulo de 15°. Se recomienda usarlo en un área no mayor a 0.5m², con superficies planas, sólidas. El módulo funciona con una señal de *trigger* (en el pin del mismo nombre) en nivel alto de por lo menos 10µs, cuando se envía un pulso al *trigger*, el sensor envía automáticamente 8 pulsos ultrasónicos de 40 kHz, y si detecta una señal de rebote, devuelve un pulso en el pin del *echo* donde la duración del nivel alto es el tiempo que tarda la señal ultrasónica en retornar [1]. El costo de este sensor es de \$ 2.5 [25]. Sus dimensiones físicas son de 40x20x15 mm.

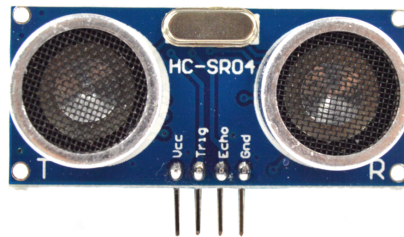


Figura 31: Sensor Ultrasónico HC-SR04 [25].

LV-MaxSonar-EZ1

Se alimenta con un voltaje entre 2.5V - 5.5V y consume una corriente nominal de 2mA. Posee un rango de detección entre 15cm - 6.45m con una resolución de 2.54cm. Puede comunicar la respuesta por PWM, señal analógica o por comunicación Serial UART. Muestra a una frecuencia de 20Hz. Es uno de los ultrasónicos con dimensiones físicas más pequeñas, son de 22x20x15 mm. [26]

Si se desea leer la señal medida en la salida analógica, el factor de escala es Vcc/512 por pulgada (2.54 cm), por ejemplo si se alimenta con 3.3V, el factor es de 6.4mV/plg o de 2.5mV/cm. Si se utiliza la salida PWM, la distancia se calcula con el factor de 147µs por pulgada o de 57.9µs por cm. El costo de este sensor es de \$ 29.95. [26]

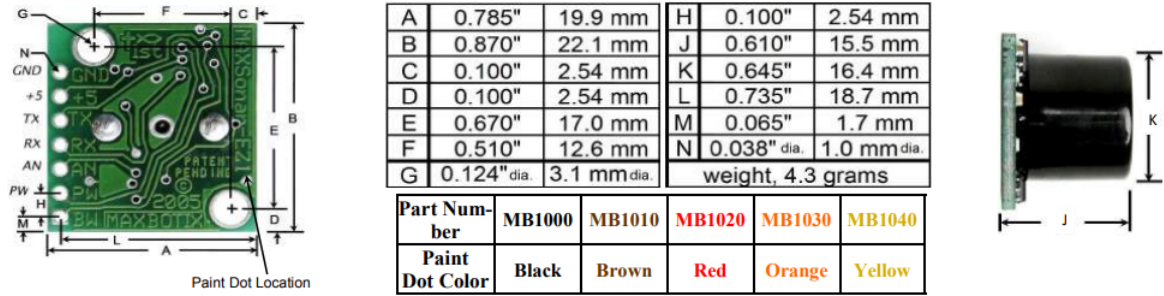


Figura 32: Dimensiones físicas del LV-MaxSonar-EZ1 [26].

Devantech SRF02

Se alimenta con un voltaje entre 5V - 5.5V y consume una corriente nominal de 4mA. Posee un rango de detección entre 15cm - 6m. Puede comunicar la respuesta por interfaz I2C o por comunicación Serial UART con un baudrate de 9600. Sus dimensiones físicas se muestran en la Figura 34. [27]



Figura 33: Devantech SRF02 [27].

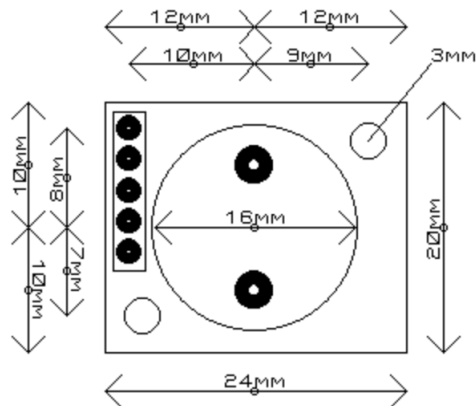


Figura 34: Dimensiones físicas del ultrasónico Devantech SRF02 [27].

6.4.2. LiDAR

Light Detection and Ranging (LiDAR) es un tipo de tecnología que puede detectar, localizar y medir objetos, incluidos líquidos y personas, en un campo de visión determinado. Es esencialmente un conjunto de software, algoritmos y conocimientos que se utilizan para diseñar u optimizar varios tipos de sensores LiDAR de estado sólido. En la taxonomía de tecnologías de teledetección, forma parte de la categoría de tiempo activo de vuelo (Time of Flight), junto con otras tecnologías como radar o ultrasonido. [28]

La función principal de los sensores LiDAR es medir la distancia entre sí mismo y los objetos en su campo de visión. Lo hace calculando el tiempo que toma un pulso de luz viajar a un objeto y volver al sensor, en función de la velocidad de la luz constante. A medida que los pulsos de luz enviados por un LiDAR golpean objetos de formas irregulares con diversas propiedades de reflexión, la señal de luz incidente se dispersa, de modo que solo una pequeña fracción de la luz regresa al receptor LiDAR. [28]

Para conocer la distancia entre el sensor y el objeto detectado se utiliza la ecuación 7. [28]

$$L = (T * C)/2 \tag{7}$$

En dónde L es la distancia entre el sensor y el objeto, T es el tiempo registrado y C es la velocidad de la luz ($2.99 \times 10^8 m/s$). [28]

LIDAR-Lite V3

LIDAR-Lite V3 es un sensor potente, escalable y económico a base de LASER que tiene una gran variedad de aplicaciones (por ejemplo en drones, detección industrial, robótica en general, etc.). Mide distancia, velocidad e intensidad de la señal de los objetivos cooperativos y no cooperativos en un rango de 0 a 40 con una resolución de 2.5cm, se alimenta con un voltaje entre 4.75V y 6V, utiliza 130 mA, muestrea a una frecuencia de hasta 500 Hz, maneja las interfaces I2C y PWM. Utiliza un ancho de onda de 905 nm por 1.3 watts. Sus dimensiones físicas son de 20x48x40 mm y pesa 20.5 gramos. Tiene un costo de \$ 130. [29]

LeddarOne

LeddarOne es un sensor LiDAR de un solo elemento, dedicado a detectar objetos y proporcionar mediciones de distancia precisas de hasta 40 m. Es adecuado para aplicaciones como detección de nivel, medición de proximidad, seguridad y vigilancia, detección de vehículos y advertencia de altura, detección óptica de alcance y altimetría de drones. Es de tamaño compacto (Ver Figura 36), bajo consumo de energía y alta precisión. Utiliza 5V de alimentación, interfaces UART y RS-485 de 3.3V, pesa 14 gramos, detecta objetos entre 0 y 40 m con una resolución de 3 mm y precisión de 5 mm, muestrea a una frecuencia máxima de 140 Hz. Tiene un costo de \$ 115.



Figura 35: LIDAR-Lite V3 [29].

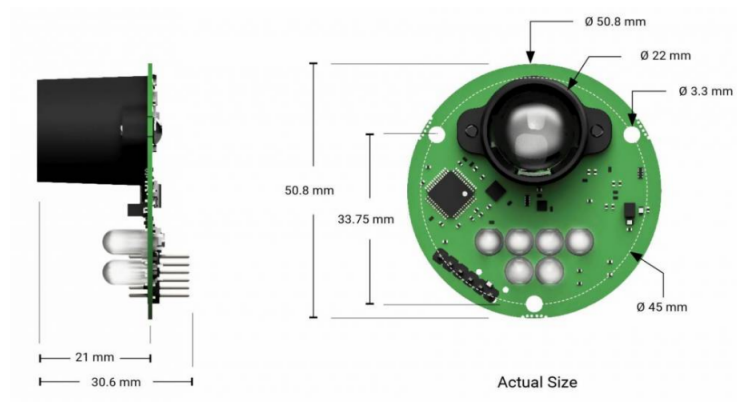


Figura 36: Dimensiones físicas de LeddarOne [30].

6.5. Controladores de motores DC

Un controlador de motores puede manejar el sentido de giro y velocidad un motor DC, además protege al sistema que manda las señales de control de situaciones como un voltaje de reversa, baja de voltaje, sobre corriente y sobre temperatura causada por los motores.

6.5.1. Controlador DRV8833

Este controlador puede manejar dos motores DC, brinda hasta 1.2 A por separado para cada motor con una fuente independiente de voltaje. Opera entre 2.7V y 10.8V. Utiliza 4 señales PWM proveniente de una unidad de control [31]. Este controlador es el que se utilizó para el diseño de la plataforma de robótica de enjambre fase 1 de la Universidad del Valle de Guatemala [1].

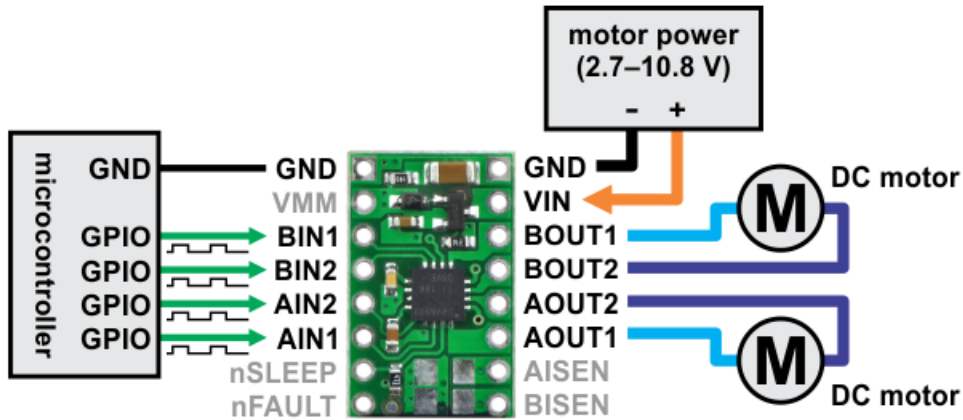


Figura 37: Pines del DRV8833 Dual Driver Motor [31].

6.5.2. Controlador TB6612FNG

Este controlador puede manejar dos motores DC, brinda hasta 1 A por separado para cada motor con una fuente independiente de voltaje. Opera con un voltaje lógico entre 2.7V y 5.5V y entre 4.5V y 13.5V para el motor [32]. La ventaja de este microcontrolador es que utiliza solo dos señales PWM para controlar la velocidad de cada motor, la desventaja es que utiliza por aparte 4 entradas lógicas que son las que definen el sentido de giro de cada motor. El costo de este controlador es de \$3.33. [32]

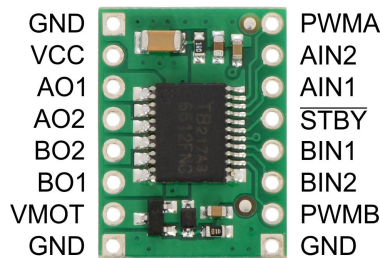


Figura 38: Pines del TB6612FNG Dual Driver Motor [32].

6.6. Encoders

Un *encoder* es un dispositivo sensor que convierte el movimiento radial en una señal eléctrica que puede ser procesada por un dispositivo de control. El *encoder* envía una señal de retroalimentación que se puede utilizar para determinar posición, velocidad, sentido de giro y cantidad de revoluciones. Los *encoders* utilizan distintos tipos de tecnología para crear una señal, pueden ser mecánicos, magnéticos, resistivos y ópticos, siendo estos últimos los más comunes. Un *encoder* óptico provee una retroalimentación basada en intermitencias de luz. En la Figura 39 se observa la construcción básica de un *encoder* óptico, en donde un LED emite un haz de luz que pasa a través de un disco codificador que posee líneas opacas que obstruyen la luz, también hay un fotoresistor que detecta el haz de luz. Cuando el disco

rota, el foto resistor detecta una luz intermitente, que con un sistema de control o se puede procesar dicha señal y obtener los parámetros antes mencionados.[33]

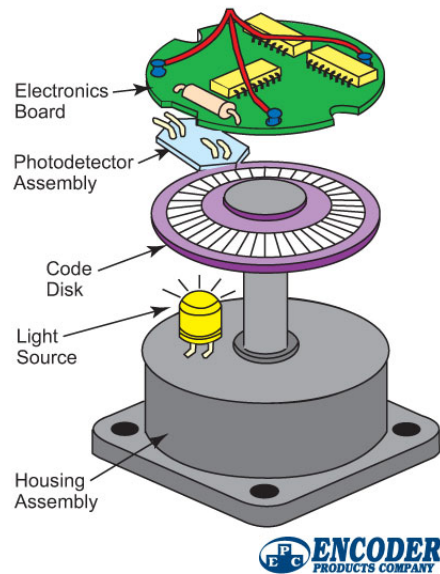


Figura 39: Construcción básica de un *encoder* óptico [33].

6.6.1. SEN0116

El componente SEN0116 es un *encoder* óptico IR fabricado por DFRobot, que provee de dos señales (AB) desfasadas 90° . Trabaja con 3.3V ó 5V y consume una corriente de 14mA a 5V. Es compatible con las llantas de 42 mm de diámetro y 19 mm de ancho que generan 12 pulsos por revolución [13] que se utilizan en la plataforma [1] desarrollada en la Universidad del Valle como se verá más adelante en Antecedentes. En conjunto con la llanta mencionada, provee una señal de 12 pulsos por revolución. El costo de este sensor es de \$ 4.5. [34]



Figura 40: *Encoder* óptico SEN0116 de DFRobot [34].

6.7. Batería Li-Po

Una batería de Polímero de Litio (Lithium Polymer Li-Po) es un tipo de batería recargable compuesta generalmente de varias celdas secundarias idénticas en paralelo para aumentar la capacidad de la corriente de descarga, y están a menudo disponibles en serie de "packs" para aumentar el voltaje total disponible. Se caracterizan por ser ligeras y por poder almacenar una gran cantidad de energía. Cada celda tiene un voltaje máximo de entre 3,7 y 4,2V dependiendo de los materiales con los que se ha hecho la batería. [35]

6.7.1. PKCELL LP785060

Esta batería Li-Po es de 3.7V de 1 celda con una capacidad de 2500mAh que da un total de 10W por hora. Sus dimensiones físicas son de 51x65x8 mm y pesa 52 gramos. [36]

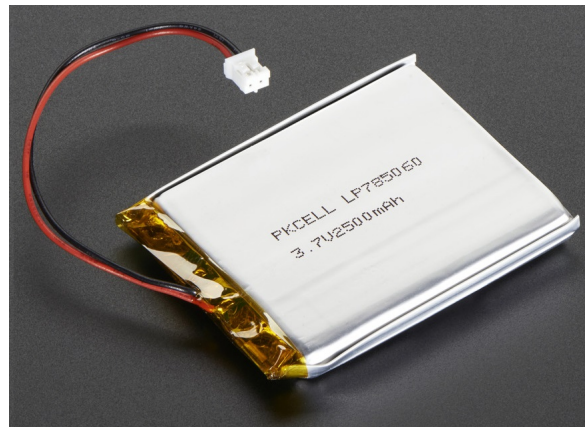


Figura 41: Batería PKCELL LP785060 [36].

6.7.2. Batería Quanam FPV 7.4V

Esta batería Li-Po provee 7.4V con una capacidad de 1500mAh. Sus dimensiones físicas son de 66x32x13 mm y pesa 60 gramos. Esta batería tiene un costo de \$ 9.54. [37]



Figura 42: Batería Quanam FPV de 7.4V y 1500mAh [37].

El desarrollo de la nueva plataforma para aplicaciones en robótica de enjambre se realizó en tres partes, estas son:

- **Diseño electrónico:** A partir de la identificación de las áreas de mejora de la plataforma de la fase 1, se plantearon nuevos sistemas que suplieran las características necesarias para la nueva plataforma. Se eligieron los nuevos componentes electrónicos y se definieron los nuevos módulos y sistemas que llevará. Luego, se definió cómo se controlará cada sistema del robot desde la unidad de control y se diseñó el diagrama electrónico para cada uno de estos. Por último, se desarrolló la programación en C de la unidad de control (PIC32MX250F128B) para el manejo de los sistemas de la nueva plataforma.
- **Diseño de estructura:** Luego de haber definido los sistemas del nuevo robot, sus componentes y su conexión electrónica, se realizó el diseño de la estructura, que se divide en la base y el PCB de la plataforma. En la base irán las llantas, motores, *encoders* y la batería. En el PCB irán todos los componentes electrónicos, la unidad de control, el sistema de detección de objetos, el sistema de control de movimiento para las llantas y motores, y las partes que funcionan como interfaz entre el robot y el usuario, como los LED's, botón de *reset* y de encendido y apagado.
- **Implementación de librerías en Matlab:** En esta parte se implementó una librería con funciones en Matlab para controlar las funciones del robot desde un ordenador. Estas funciones sirven para crear una conexión con el robot, transmitir los distintos comandos que recibe el robot vía Wi-Fi y recibir la información solicitada.

Diseño electrónico de la nueva plataforma

La plataforma desarrollada en este proyecto de graduación es la Fase 2, como seguimiento al robot desarrollado en la Universidad del Valle de Guatemala en el año 2017 como proyecto de graduación, toda la documentación de la Fase 1 de este robot se encuentra en el trabajo de tesis con modalidad de megaproyecto *Reingeniería de Megaproyectos Fase 1* [1] buscando las secciones de *Robótica de Enjambre*, donde encuentran todas las características del robot, sus módulos y funciones (Cuadro 3) que se resumieron anteriormente en la sección de Antecedentes.

8.1. Identificación de áreas de mejora de la plataforma de la fase 1

En *Reingeniería de Megaproyectos Fase 1* [1], la plataforma desarrollada para aplicaciones en robótica de enjambre tienen como recomendaciones probar un nuevo diseño de la falda de sensores, porque a la hora de realizar las lecturas devolvía varios valores en *timeout*, por otro lado, denotan que se utilizó Teensy por la amplia documentación que existe pero que se recomienda realizar pruebas con otro microcontrolador para obtener mejor desempeño. Los *encoders* utilizados no dieron problema con las mediciones con la única observación que pueden descalibrarse, para solucionar este problema basta con volverlos a calibrar observando que las señales de salida sean acorde a lo esperado, por lo tanto recomiendan dejar pines o terminales accesibles para medir dichas señales en un futuro modelo de este robot. Por lo tanto, las dos áreas de mejora a tomar en cuenta para esta segunda fase es la unidad de control y el sistema de detección de objetos alrededor de la plataforma.

8.1.1. Unidad de control

En la fase 1 se utilizó el Teensy 3.2 [1] como unidad de control. Las características de este microcontrolador se detallan en la sección 4.2.1 del Marco Teórico. Para este microcontrolador hay una basta documentación que permite utilizar sus módulos de forma fácil y rápida [1], su programación se basa en Arduino.

Arduino es una plataforma de electrónica de código abierto basada en hardware y software fácil de utilizar. Utiliza el lenguaje de programación Arduino (basado en el cableado) y el software Arduino (IDE), basado en el procesamiento. Arduino es una herramienta fácil para generar prototipos de forma rápida, dirigido a estudiantes sin experiencia en electrónica y programación [2]. Por lo tanto, la ventaja de utilizar Arduino es la fácil implementación de cualquier prototipo, utilizando librerías de programación de Arduino (IDE) que programan el microcontrolador de la plataforma, en este caso Teensy 3.2, pero esto conlleva la desventaja que no se tiene el control total o exacto de los módulos del microcontrolador, lo que puede incurrir en un ineficiente flujo de control de tiempos del microcontrolador.

Un objetivo específico es controlar esta plataforma robótica desde un ordenador utilizando Matlab, donde se envían los comandos de control de los módulos del robot y donde se recibe la información de los sensores a tiempo real, es necesario una unidad de control que maneje de mejor forma los tiempos del microcontrolador, para esto se decidió utilizar el microcontrolador PIC32MX250F128B [3] que posee las características necesarias para controlar todos los módulos de esta plataforma, como se verá más adelante. Por otro lado, la selección de utilizar este microcontrolador en vez del Teensy 3.2 contribuyó a un PCB más pequeño por sus dimensiones físicas reducidas comparadas con el Teensy.

8.1.2. Sistema de detección de objetos

En el prototipo de la fase 1 se utilizó una falda de 6 sensores HC-SR04 [25] ubicados a 60° entre cada uno formando una estructura hexagonal como se observa en la Figura 31, estos sensores miden 4.5 cm de ancho y 2 cm de alto, lo que provoca que las dimensiones de la plataforma tengan dimensiones físicas en el plano XY que no se pueden reducir más. Por otro lado, estos sensores están fijos a 60° , lo que genera lecturas de objetos que estén solo en el campo de visión de estos sensores, dejando una zona muerta entre cada sensor vecino. Como sistema alternativo se propuso realizar un sistema de detección que se compone de dos sensores de distancia y un servomotor que hace girar estos sensores a 180° , ocasionando un barrido de detección de objetos por el plano XY , con esto se espera minimizar el área física que utiliza este sistema, obtener mejores resultados de acuerdo a la recomendación encontrada en [1] y mejorar la detección de objetos realizando un barrido en todo el plano XY comparada los 6 puntos detectables con la plataforma anterior.

8.2. Selección de componentes

8.2.1. Sistema de detección de objetos

Este nuevo sistema se conformará por dos sensores de distancia y un servomotor que al girar 180° en el eje Z ejecutará un barrido de detección en el plano XY .

Sensor de distancia

Para seleccionar el sensor de distancia se investigaron las características de distintos sensores de tipo ultrasónicos y Lidar que se detallan en la sección 4.4 del Marco Teórico, la tabla de características se observa en el Cuadro 5, los primeros dos sensores son tipo Lidar y los siguientes tres sensores son de tipo ultrasónicos.

Sensor	Corriente	Comunicación	Frec. Muestreo	Precio
LIDAR-Lite v3	130mA	I2C o PWM	500	\$ 130
LeddarOne	300mA	UART o RS-485	100	\$ 115
LVMaXSonar-EZ1	2mA	PWM, Analog, UART	20	\$ 29,95
HC-SR04	15mA	Trigger y Echo	40	\$ 3.5
Devantech SRF02	4mA	I2C o UART	20	\$ 15.72

Cuadro 5: Características de distintos sensores de distancia.

Para decidir cuál de estos ultrasónicos elegir, se realizó un estudio de características, que es un método para elegir la mejor opción de entre varias tomando en cuenta las características relevantes para el proyecto que estas poseen, son de preferencia cualitativas. Este método se realiza colocándole un peso entre 1 y 5 (en este caso) a cada característica y la opción que obtiene más puntaje de acuerdo a sus características, se considera que es la mejor opción.

Se tomaron en cuenta las características detalladas en el Cuadro 5 y se les asignó un peso a cada una: el precio de los sensores tiene un peso de 5; el consumo de corriente tiene un peso de 4; los pines utilizados que dependen del tipo de comunicación tiene un peso de 2; las dimensiones físicas tienen un peso de 1; la velocidad de muestreo de los sensores tiene un peso de 3; por último, la disponibilidad local tiene un peso de 2. Luego se asigna un rango de evaluación para cada característica:

- Precio de sensores: De \$ 1.00 a \$ 100.00. \$ 1.00 equivale a obtener 100 % y \$ 100.00 o más equivale a 0 %.
- Consumo de corriente: De 1mA a 300mA. 1mA equivale a obtener 100 % y 300mA o más equivale a obtener 0 %.
- Pines utilizados: De 1 a 2 pines. Utilizar 1 pin equivale a obtener 100 % y utilizar 2 pines equivale a 50 %
- Dimensiones físicas: Un sensor que tiene transmisor y receptor por separado es casi el doble del tamaño de un sensor que tiene un solo transceptor. Por lo tanto, si tienen un transmisor y un receptor por separado, obtienen 50 % y si tiene un solo transceptor tiene un 100 %.

- Velocidad de muestreo: De 10Hz a 150Hz. 10Hz equivale a obtener 0 % y 150Hz equivale a obtener 100 %.
- Disponibilidad: Con disponibilidad local o no. Un sensor con disponibilidad local equivale a 100 % y sin disponibilidad local equivale a 50 %.

El valor que se obtiene bajo este rango de evaluación en cada característica es multiplicado por el peso asignado, luego se suma el punteo obtenido en cada característica para cada sensor. Finalmente, el sensor con mayor punteo acumulado es el que se presenta como mejor opción.

Variables	Peso	Rango		LIDAR Lite	Leddar One	MaxSonar	HCSR04	Devantech
Corriente	4	300	1	2.26	0.00	3.97	3.80	3.95
Precio	5	100	1	0.00	0.00	3.49	4.87	4.21
Pines	2	2	0	2.00	1.00	2.00	1.00	1.00
Dimensiones	1	si	no	0.50	1.00	1.00	0.50	1.00
Muestreo	3	10	150	4.80	2.00	0.40	1.80	0.40
Disponibilidad	2	si	no	1.00	1.00	1.00	2.00	1.00
Puntuación				10.06	5.00	11.86	13.47	11.55

Cuadro 6: Estudio de características de sensores de distancia.

En el Cuadro 6 se muestran los resultados del estudio de las características de los sensores de distancia. La mejor puntuación la obtuvo el sensor HC-SR04 [25] con 13.47 puntos, sin embargo, este es el sensor que se utilizó para la primera fase de este proyecto, y en [1] recomiendan no utilizarlo, por lo tanto el sensor de distancia que se escoge es el LV-MaxSonar-EZ1 [26], mostrado en la Figura 32 junto a sus dimensiones físicas. La ventaja que nos trae este sensor es que es un transceptor, lo que disminuye el espacio físico a utilizar, asimismo, consume tan solo 2mA y se puede utilizar la señal análoga, PWM o UART para obtener las mediciones.

Servomotor

Para seleccionar el servomotor se investigaron las características de tres actuadores de este tipo, que se resumen en el Cuadro 7, aquí hay una columna llamada *Área en PCB*, que representa el área que se utilizaría en el PCB pensando que el servomotor irá incrustado ahí, y mientras menor sea el área utilizada, mejor para el diseño mecánico. Por otro lado, también se incluyó si el servomotor es análogo o digital, ya que un servomotor digital presenta mayor precisión en la posición angular de su eje.

Aquí también se realizó un estudio de las características detalladas en el Cuadro 7. El peso que se le asignó a cada una de estas es: el precio del servomotor tiene un peso de 5; el consumo de corriente tiene un peso de 4; el área en PCB tiene un peso de 3; el tipo de servomotor (análogo o digital) tiene un peso de 2. El rango de evaluación para cada característica es:

Servo Motor	Corriente	Área en PCB	Tipo	Precio
BMS-306DMAx	150mA	220mm ²	Análogo	\$ 13.63
DFRobot Micro Servo	300mA	275mm ²	Análogo	\$ 3.50
Traxxas Nano Servo	15mA	132mm ²	Digital	\$ 4.49

Cuadro 7: Características de los servomotores tomados en cuenta.

- Precio de servomotores: De \$ 1.00 a \$ 30.00. \$ 1.00 equivale a obtener 100 % y \$ 30.00 o más equivale a 0 %.
- Consumo de corriente: De 1mA a 300mA. 1mA equivale a obtener 100 % y 300mA o más equivale a obtener 0 %.
- Área PCB: De 1mm² a 500mm². 1mm² equivale a obtener 100 % y 500mm² equivale a 0 %
- Tipo de servomotor: Digital o análogo. Digital equivale a 100 % y análogo equivale a 50 %.

Variables	Peso	Rango		BMS-306DMAx	DFROBOT Micro Servo	Traxxas Nano Servo
Corriente	4	300	1	2.00	0.00	3.80
Precio	5	30	1	2.73	4.42	4.25
Área PCB	3	500	1	1.68	1.35	2.21
Tipo	2	Dig	Ana	1.00	1.00	2.00
Puntuación				8.41	6.77	12.26

Cuadro 8: Estudio de características de servomotores.

El servomotor elegido como mejor opción es el Traxxas Digital Nano Servo Motor, que trae las ventajas de ocupar un área bastante pequeño, lo que minimiza las dimensiones físicas de la estructura mecánica. Por otro lado, es digital y con mayor precisión que uno análogo y su consumo de corriente es bastante bajo, tan solo de 15mA. [22],

8.2.2. Unidad de control


Para definir la unidad de control a utilizar, es necesario conocer la cantidad de puertos y módulos que necesitaremos, estos están determinados en el Cuadro 9. Cabe mencionar que en la plataforma de la fase 1 se utilizó el controlador DRV8833 [31] para manejar los motores de las llantas, sin embargo esta vez se optó por utilizar el controlador TB6612FNG [32] porque utiliza únicamente 2 señales PWM, como se detalla en la sección 4.5 del Marco Teórico. Por otro lado, para realizar una interfaz plataforma-usuario se incorporan 3 LED's que servirán como interacción entre el robot y el usuario.

Sistema	Componente	Pines	Tipo
Movimiento	TB6612FNG	4	Digitales
		2	PWM
Medición	Encoders	4	Digitales
Detección de Objetos	MaxSonar	2	Digitales
	Trixxas Nano Servo	1	PWM
Comunicación	ESP-8266	2	UART
Interfaz Usuario	LED's	3	Digitales

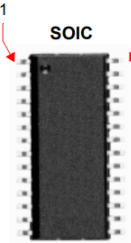
Cuadro 9: Listado de componentes y pines que se conectarán a la unidad de control.

De esta forma, se determinó que el microcontrolador PIC32MX250F128B [3] supe las necesidades de la plataforma para controlar todos sus módulos y sistemas. Las características específicas del microcontrolador se encuentran en [3] y se resumieron anteriormente en la sección 4.2.2 del Marco Teórico. Se alimenta con 3.3V, es de tecnología de 32bits, funciona a una frecuencia máxima de reloj de 50 MHz, tiene disponible un basto vector de interrupciones que ayudarán a tener un mejor control de las señales que se manejarán en la plataforma, tiene la opción de remapear las funciones de sus pines y de los 28 que tiene, 19 pines están disponibles para su uso, de los cuales, 18 se utilizarán.


28-PIN SOIC, SPDIP, SSOP (TOP VIEW)^(1,2,3)



SSOP



SOIC



SPDIP

PIC32MX210F016B
PIC32MX220F032B
PIC32MX230F064B
PIC32MX230F256B
PIC32MX250F128B
PIC32MX270F256B

Pin #	Full Pin Name	Pin #	Full Pin Name
1	MCLR	15	V _{BUS}
2	PGED3/VREF+/CVREF+/AN0/C3INC/RPA0/CTED1/PMD7/RA0	16	TDI/RPB7/CTED3/PMD5/INT0/RB7
3	PGEC3/VREF-/CVREF-/AN1/RPA1/CTED2/PMD6/RA1	17	TCK/RPB8/SCL1/CTED10/PMD4/RB8
4	PGED1/AN2/C1IND/C2INB/C3IND/RPB0/PMD0/RB0	18	TDO/RPB9/SDA1/CTED4/PMD3/RB9
5	PGEC1/AN3/C1INC/C2INA/RPB1/CTED12/PMD1/RB1	19	V _{SS}
6	AN4/C1INB/C2IND/RPB2/SDA2/CTED13/PMD2/RB2	20	V _{CAP}
7	AN5/C1INA/C2INC/RTCC/RPB3/SCL2/PMWR/RB3	21	PGED2/RPB10/D+/CTED11/RB10
8	V _{SS}	22	PGEC2/RPB11/D-/RB11
9	OSC1/CLKI/RPA2/RA2	23	V _{USB3v3}
10	OSC2/CLKO/RPA3/PMA0/RA3	24	AN11/RPB13/CTPLS/PMRD/RB13
11	SOSCI/RPB4/RB4	25	CVREFOUT/AN10/C3INB/RPB14/V _{BUSON} /SCK1/CTED5/RB14
12	SOSCO/RPA4/T1CK/CTED9/PMA1/RA4	26	AN9/C3INA/RPB15/SCK2/CTED6/PMCS1/RB15
13	V _{DD}	27	AV _{SS}
14	TMS/RPB5/USBID/RB5	28	AV _{DD}

Figura 43: Pines del microcontrolador PIC32 y tipos de empaquetados disponibles [3].

8.3. Módulos de la nueva plataforma

8.3.1. Sistema de movimiento

Este sistema se encargará de todo el control de movimiento de la plataforma. Se conformará de un ball caster que servirá como soporte de equilibrio de la plataforma, de dos llantas de goma de 42 mm de diámetro y 19 mm de ancho [13] que son compatibles con los motores utilizados en [1], como actuadores de giro como se muestra en la Figura 19. Estos motores son DC de metal con caja reductora con relación 75:1 [15] se operarán a 5V. Para controlar los motores y proteger a la unidad de control y demás sistemas de la plataforma, se utilizará un controlador TB6612FNG [32] que maneja el sentido de giro y la velocidad de dos motores DC, a los que les puede brindar hasta 1 A de corriente a cada uno en modo continuo. La forma en que se realiza la conexión entre los motores, el controlador TB6612FNG [32] y la unidad de control se muestra en la Figura 44, en donde se puede observar que se utilizarán 3.3V para alimentar el sistema del controlador y 5V para alimentar los motores DC de la plataforma.

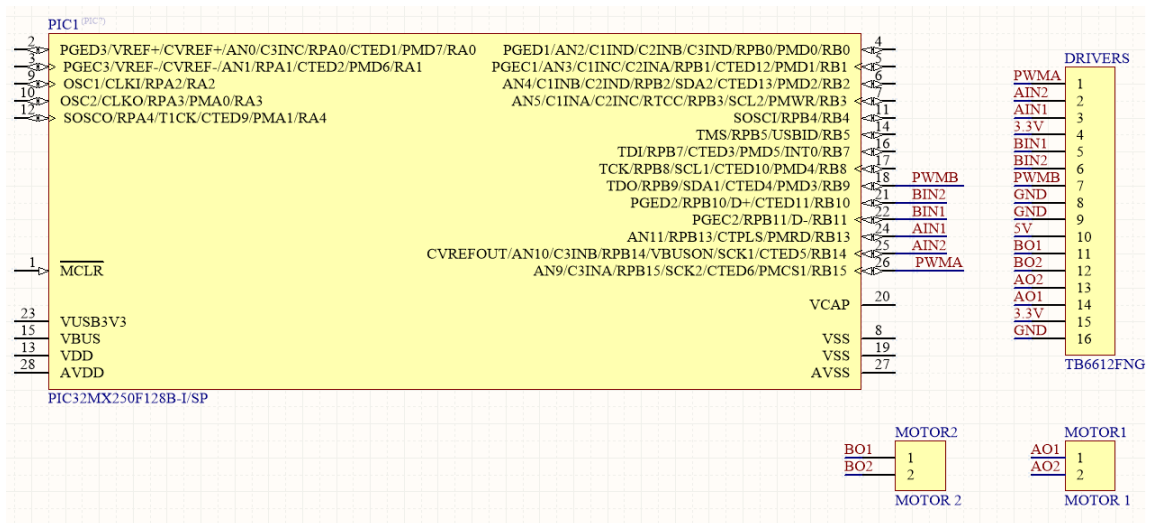


Figura 44: Diagrama electrónico del sistema de movimiento.

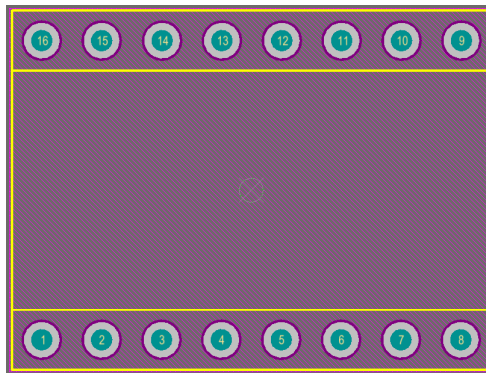


Figura 45: Posición física de los pines de TB6612FNG simulada en Altium.

Los pines que se utilizarán de la unidad de control para este sistema son:

- Pin 18: Señal PWM que controlará la velocidad del motor de la llanta izquierda, *PWMB*.
- Pin 21: *BIN2*.
- Pin 22: *BIN1*, que junto a *BIN2* controlarán el sentido de giro del motor de la llanta izquierda.
- Pin 24: *AIN1*.
- Pin 25: *AIN2*, que junto a *AIN1* controlarán el sentido de giro del motor de la llanta derecha.
- Pin 26: Señal PWM que controlará la velocidad del motor de la llanta derecha, *PWMA*.

8.3.2. Sistema de medición

Este sistema se encargará de medir la velocidad, distancia recorrida y sentido de giro de cada llanta. Se utilizarán 2 *encoders* ópticos SEN0116 [34] para este fin, su funcionamiento se resume en la sección 5.10.1 de Antecedentes. Estos sensores son compatibles con las llantas de goma de 42 mm de diámetro y 19 mm de ancho [13] que se utilizarán en esta plataforma, y la forma en que se acoplan se muestra en la Figura 19. El diagrama electrónico se muestra en la Figura 46, donde *ENCDA* y *ENCDB* son las señales A y B del encoder de la llanta derecha y las otras dos señales, *ENCIA* y *ENCIB*, son del *encoder* de la llanta izquierda, respectivamente.

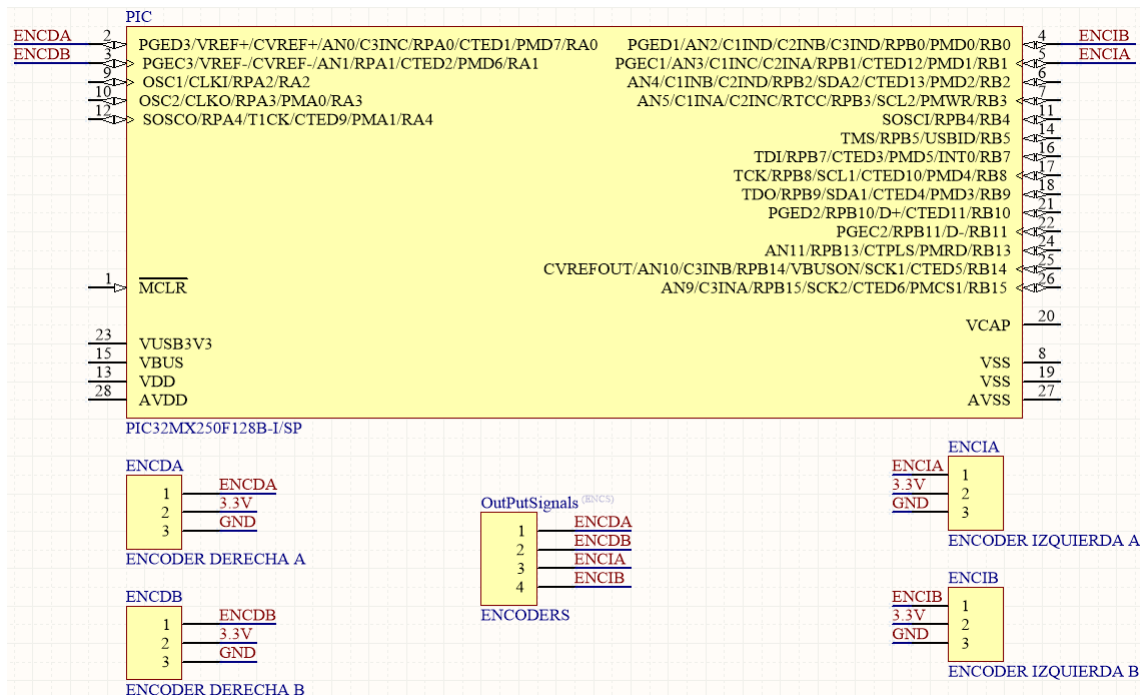


Figura 46: Diagrama electrónico del sistema de medición de movimiento.

Cabe mencionar que estos *encoders* se fabrican para alimentarse con 5V, sin embargo, tienen la opción de acoplarse para que funcionen a 3.3V como se menciona en *Reingeniería de Megaproyectos Fase 1* [1], que solamente se trata de quitar una resistencia y conectar las terminales en donde esta estaba soldada.

Los pines que se utilizarán de la unidad de control para este sistema son:

- Pin 2: Señal A encoder de la llanta derecha, *ENCDA*.
- Pin 3: Señal B encoder de la llanta derecha, *ENCDB*.
- Pin 4: Señal B encoder de la llanta izquierda, *ENCIB*.
- Pin 5: Señal A encoder de la llanta izquierda, *ENCIA*.

8.3.3. Sistema de detección de objetos

En este sistema se utilizarán dos sensores ultrasónicos LV-MaxSonar-EZ1 [26] y el servomotor Traxxas Digital Nano Servo [22] que se colocarán de tal forma que, los sensores queden viendo en direcciones contrarias y el servomotor los haga girar 180° para realizar un barrido de detección de objetos por todo el plano *xy*. La forma en que estos se conectan al microcontrolador se muestra en la Figura 47, en donde *MS1* es la señal proveniente del sensor ultrasónico 1, *MS2* es la señal proveniente del sensor ultrasónico 2 y *PWMS* es la señal PWM que controlará al servomotor.

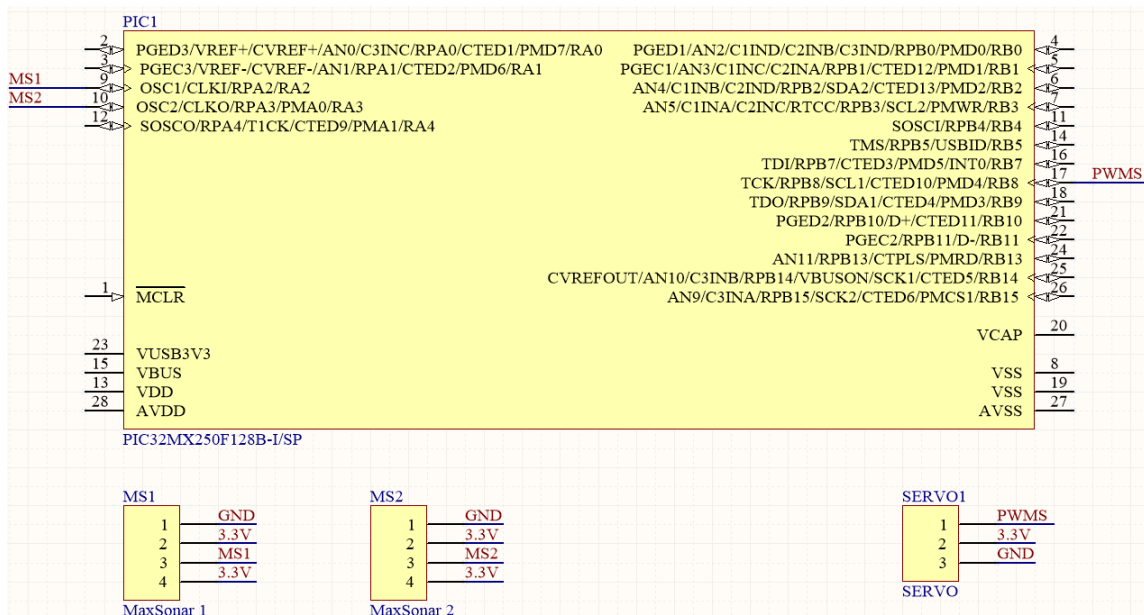


Figura 47: Diagrama electrónico del sistema de detección de objetos.

Las mediciones que realiza el LV-MaxSonar-EZ1 [26] se pueden transmitir a través de una señal analógica, una señal PWM o por interfaz UART. Por lo tanto, para ahorrar pines se descartó la opción UART que utiliza dos pines mientras que con la señal analógica o PWM

se utiliza solo un pin, entre estos dos la señal análoga puede ser fácilmente influenciada por ruido en el sistema o por la descarga temporal de la batería durante el uso la plataforma, por lo que las mediciones pueden ir variando en el tiempo y ocasionar un gran error de precisión en las lecturas, por lo tanto, se escogió utilizar la señal PWM de estos sensores ultrasónicos para leerlas en la unidad de control.

Finalmente, los pines que se utilizarán de la unidad de control para este sistema son:

- Pin 9: Señal PWM del sensor ultrasónico 1, *MS1*.
- Pin 10: Señal PWM del sensor ultrasónico 2, *MS2*.
- Pin 17: Señal PWM que controlará el servomotor, *PWMS*.

8.3.4. Sistema de comunicación

Este sistema se conforma únicamente de la unidad de control y el módulo ESP-8266 [17]. Como se describió anteriormente en el Marco Teórico acerca de este módulo, utiliza 3.3V de alimentación, se configura y utiliza a través de comandos AT, la interfaz para controlarlo es UART, por lo que se utilizará el módulo UART (2 pines configurados como RX y TX) de la unidad de control. La conexión entre el PIC32MX250F128B [3] y el módulo ESP-8266 se muestra en la Figura 48, donde *TX* es el pin TX del módulo ESP-8266 que va conectado con el pin 6 (Configurado como RX) del microcontrolador y *RX* es el pin RX del ESP-8266 que va conectado al pin 7 (Configurado como TX) del microcontrolador.

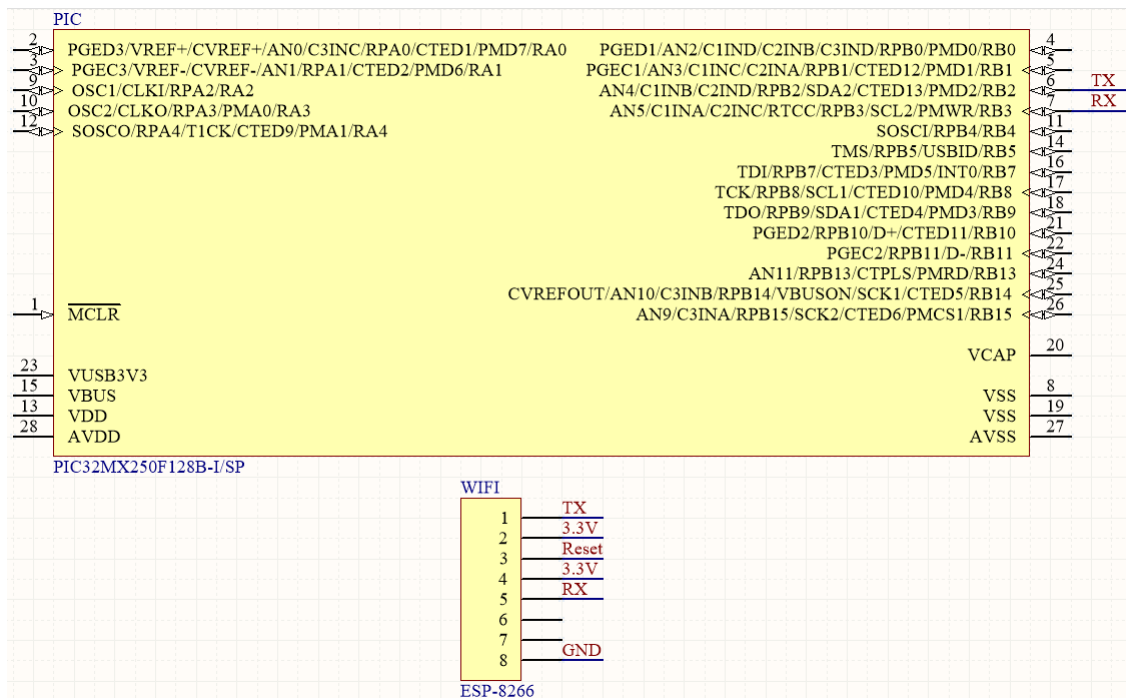


Figura 48: Diagrama electrónico del sistema de comunicación.

Los pines que se utilizarán de la unidad de control para este sistema son:

- Pin 6: *TX* del ESP-8266 conectado al pin 6 del microcontrolador configurado como *RX*.
- Pin 7: *RX* del ESP-8266 conectado al pin 7 del microcontrolador configurado como *TX*.

8.3.5. Interfaz usuario-robot

Como interfaz de usuario-robot nos referimos a las partes del robot que permiten la interacción entre el robot y el usuario que lo controla. Aquí se utilizará un switch de encendido y apagado del robot (Como se verá en la siguiente subsección), 2 LED's blancos y 1 LED rojo para la visualización del usuario, un switch de reset para reiniciar tanto la unidad de control como el módulo ESP-8266, 5 pines hembra que estarán disponibles para la programación del microcontrolador en el robot sin necesidad de tener que extraerlo y 4 pines hembra para medir las señales de los encoders y facilitar la calibración de estos en caso sea necesario, tomando en cuenta la recomendación descrita en Antecedentes acerca del modelo anterior desarrollado [1]. En la Figura 49 se encuentra el diagrama electrónico de estas partes, excepto del switch de encendido y apagado, en donde se utilizan los pines 12, 14 y 16 para los tres LED's, la señal de reset generada por el Push Button ingresa al pin 1 del microcontrolador y al pin 3 del ESP-8266, los pines donde ingresan las señales de los encoders (*ENCDA*, *ENCDB*, *ENCIA*, *ENCIB*) también son los pines que se utilizan para programar al microcontrolador, por eso es que se conectan a un Header de pines hembra para tenerlos disponibles al usuario.

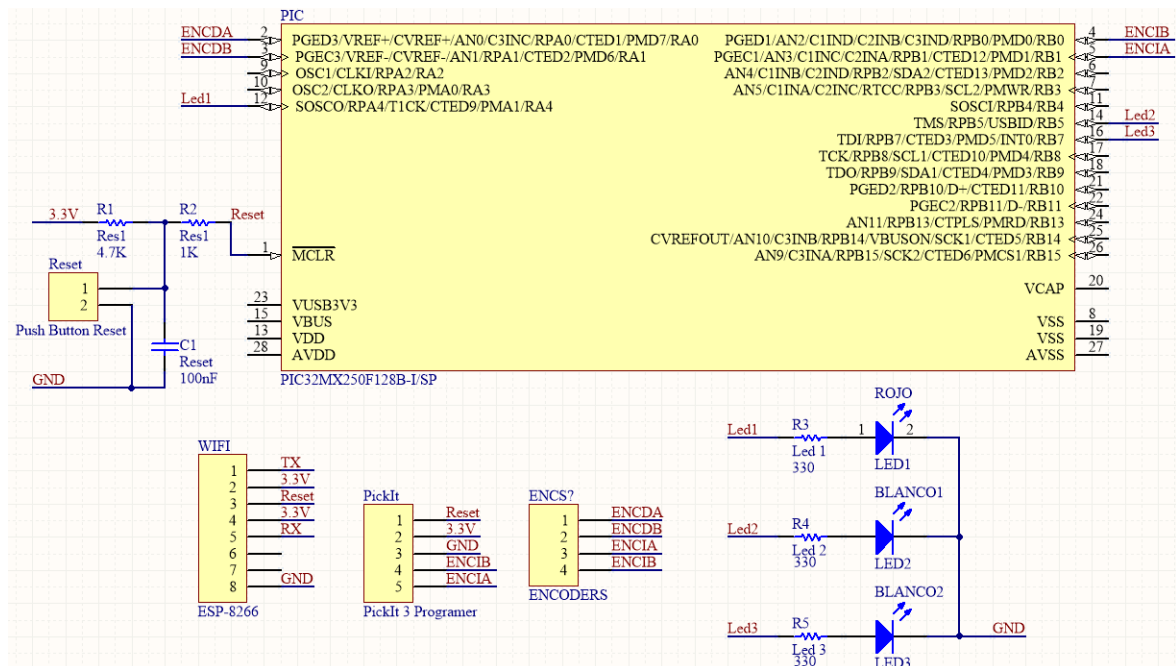


Figura 49: Diagrama electrónico de la interfaz Usuario-Robot.

Los pines que se utilizarán de la unidad de control para este sistema son:

- Pin 1: Donde ingresa la señal del *Reset*.
- Pin 2: Utilizado para programación del microcontrolador, *ENCDA*.
- Pin 3: Utilizado para programación del microcontrolador, *ENCDB*.
- Pin 4: Utilizado para programación del microcontrolador, *ENCIB*.
- Pin 5: Utilizado para programación del microcontrolador, *ENCIA*.
- Pin 12: *Led1*.
- Pin 14: *Led2*.
- Pin 16: *Led3*.

8.3.6. Sistema de potencia

Ahora que ya se han definido todos los sistemas que controlarán las funciones del robot, queda definir cómo se alimentará al circuito en general de forma autónoma, es decir, con fuente portable para la plataforma. Pudo observarse en el diseño de los sistemas que todos sus componentes se alimentarán con 3.3V a excepción de los motores, que se alimentarán con 5V. En el prototipo anterior [1], se utilizó una batería Li-Po PKCELL LP785060 de 3.7V [36], sin embargo, ahora es necesario utilizar una batería con mayor voltaje, para alimentar a los motores DC, lo que descarta la opción de utilizar nuevamente la misma batería.

Cantidad	Componente	Corriente	Voltaje	Potencia
2	Motor [15]	150mA	5V	1500mW
2	<i>Encoder</i> SEN0116 [34]	14mA	3.3V	92.4mW
2	LV-MaxSonar-EZ1 [26]	2mA	3.3v	13.2mW
3	LED	20mA	3.3V	198mW
1	Traxxas Servo [22]	15mA	3.3V	49.5mW
1	ESP-8266 [17]	215mA	3.3V	709.5mW
1	PIC32MX250F128B [3]	20mA	3.3V	66mW
Potencia Total				2628.6mW

Cuadro 10: *Power Budget* de los componentes que utilizará la nueva plataforma.

Para elegir una nueva batería se ha realizado un *Power Budget* que se detalla en el Cuadro 10. Suponiendo que todos los sistemas de la plataforma están siendo utilizados al mismo tiempo y a su máximo rendimiento, utilizaría 2628.6mW. Para calcular la capacidad de carga de la batería que se desea obtener, se utiliza la ecuación 8.

$$C = P * T \tag{8}$$

Donde C es la carga de la batería en [mWh], T es el tiempo de duración de la batería en horas y P es la potencia en [mW] que utilizará la plataforma. Por lo tanto, suponiendo que se desea utilizar una batería que supla por 3 horas como mínimo a la plataforma utilizando

todos los sistemas a la vez, es decir, utilizando 2628.6mW, se necesitaría una batería de 7885.8 mWh.

$$C = P * T = (2628.6mW) * (3h) = 7885.8mWh \quad (9)$$

Después de realizar el cálculo de la carga mínima necesaria de una batería para que la plataforma tenga una autonomía mínima de 3 horas, se propuso utilizar la batería Li-Po Quanum FPV [37] de 7.2V y 1500mAh, con dimensiones de 66x32x13 mm como se detalló anteriormente en el Marco Teórico. Esta batería posee una carga de 10800 mWh. Si se utiliza esta batería, despejando T de la ecuación 8, se obtiene que la nueva plataforma tendrá un tiempo de operación de 4.1 horas.

$$T = \frac{C}{P} = \frac{10800mWh}{2628.6mW} = 4.1horas \quad (10)$$

Por lo tanto, esta batería daría más del mínimo de tiempo esperado y sus dimensiones y pesos son bastante reducidos.

Ahora que ya se definió que se utilizará la batería Quanum FPV [37], hay que determinar el circuito que proveerá los voltajes necesarios (3.3V y 5V) que alimentarán los sistemas de la plataforma, para esto se utilizará el CI L7805CV, que provee 5V y un máximo de 1.5A, tiene protección térmica de sobrecarga y contra corto circuito. El rango de voltaje de entrada es entre 7V y 18V. Tiene un costo de \$ 0.5. [38]

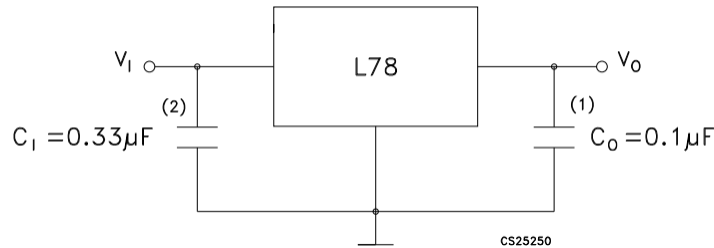


Figura 50: Diagrama de conexión de reguladores de la serie L78 [38].

Por otro lado, se utilizará el CI AZ1117 3.3E1 [39] para los sistemas que utilizan 3.3V. Este IC se conecta de igual forma que el CI L7805CV (Figura 50). Este es un regulador lineal que provee 3.3V y 1A de corriente, tiene un rango entre 4.75V y 10V de voltaje de entrada. Tiene un costo de \$ 0.288.

De esta manera, ambos reguladores de voltaje cumplen con las características necesarias para proveer de voltaje y corriente necesarios a los sistemas del robot.

8.3.7. Diagrama electrónico final

Ahora que ya se definieron los componentes de todos los sistemas y su conexión electrónica, el diagrama final se puede observar en la Figura 51, el documento también se encuentra en Anexos.

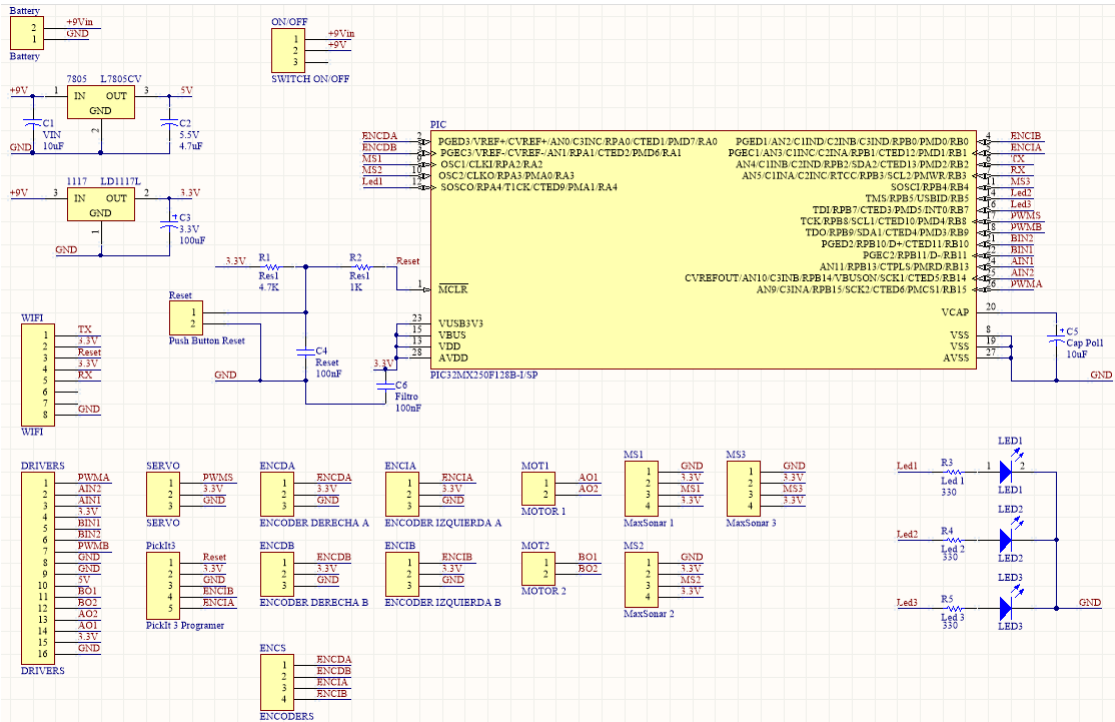


Figura 51: Diagrama final de la nueva plataforma para aplicaciones en robótica de enjambre.

Finalmente, los pines que se utilizarán del microcontrolador son:

- Pin 1: Donde ingresa la señal del *Reset*.
- Pin 2: Utilizado para programación del microcontrolador y donde ingresa la señal A del encoder de la llanta derecha, *ENCDA*.
- Pin 3: Utilizado para programación del microcontrolador y donde ingresa la señal B del encoder de la llanta derecha, *ENADB*.
- Pin 4: Utilizado para programación del microcontrolador y donde ingresa la señal B del encoder de la llanta izquierda, *ENCIB*.
- Pin 5: Utilizado para programación del microcontrolador y donde ingresa la señal A del encoder de la llanta izquierda, *ENCA*.
- Pin 6: *TX* del ESP-8266 conectado al pin 6 del microcontrolador configurado como RX.
- Pin 7: *RX* del ESP-8266 conectado al pin 7 del microcontrolador configurado como TX.
- Pin 9: Ingreso de la señal PWM del sensor ultrasónico 1, *MS1*.
- Pin 10: Ingreso de la señal PWM del sensor ultrasónico 2, *MS2*.
- Pin 11: Ingreso de la señal PWM del sensor ultrasónico 3, *MS3*.

- Pin 12: *Led1*.
- Pin 14: *Led2*.
- Pin 16: *Led3*.
- Pin 17: Salida de la señal PWM que controlará el servomotor, *PWMS*.
- Pin 18: Salida de la señal PWM que controlará la velocidad del motor de la llanta izquierda, *PWMB*.
- Pin 21: *BIN2*.
- Pin 22: *BIN1*, que junto a *BIN2* controlarán el sentido de giro del motor de la llanta izquierda.
- Pin 24: *AIN1*.
- Pin 25: *AIN2*, que junto a *AIN1* controlarán el sentido de giro del motor de la llanta derecha.
- Pin 26: Salida de la señal PWM que controlará la velocidad del motor de la llanta derecha, *PWMA*.

Los pines 8, 13, 15, 19, 20, 23, 27 y 28 son reservados para las conexiones de alimentación y configuración física para el correcto funcionamiento del microcontrolador, las conexiones de estos pines se observan en la Figura 26. Por otro lado, se puede observar que se agregó un ingreso de señal de un tercer ultrasónico LV-MaxSonar-EZ1 [26], esto con la intención de dejarlo disponible posteriormente en el PCB con un pin hembra para que en futuras ocasiones se puede programar y utilizarlo para controlar o recibir señal de un dispositivo que se quiera incorporar a la plataforma.

8.4. Programación del microcontrolador

La programación del microcontrolador PIC32MX250F128B [3] se realizó en lenguaje C utilizando el entorno de desarrollo MPLAB para desarrollar el código y el instrumento PickIt v3 [19] para descargar la programación al microcontrolador. La forma en que el PickIt v3 se conecta al microcontrolador se detalló anteriormente en Marco Teórico.

8.4.1. Configuración inicial del microcontrolador

El microcontrolador PIC32MX250F128B trabaja a un máximo de 50Mhz de reloj interno como se indicó anteriormente en Marco Teórico. Para no forzar el microcontrolador y asegurarse que todas sus funciones y módulos trabajen correctamente, el reloj interno se configuró a 40 Mhz. El diagrama de flujo general de la configuración y programación del microcontrolador se muestra en la Figura 52.

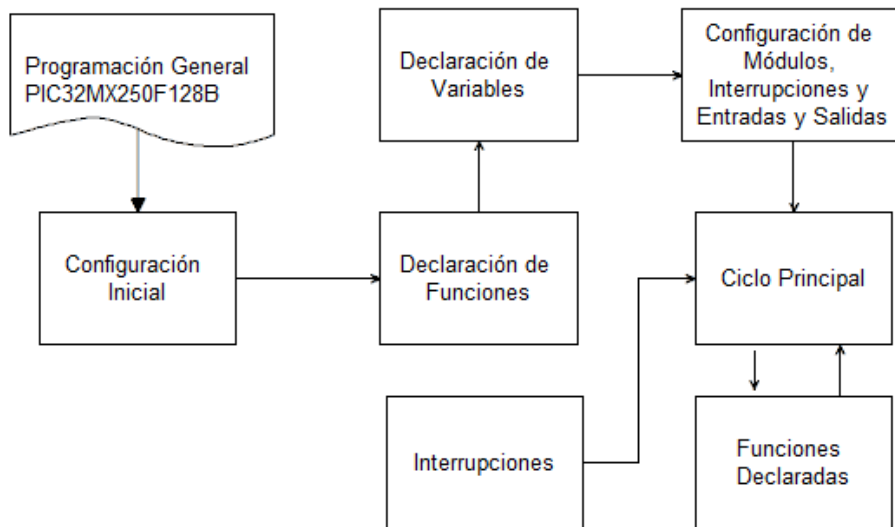


Figura 52: Diagrama de flujo de la configuración general del microcontrolador.

La programación completa del microcontrolador PIC32MX250F128B [3] se encuentra en Anexos, al final de este documento. Siguiendo el diagrama de flujo de la Figura 52, lo primero que se realiza en la programación es la configuración inicial del microcontrolador, es decir, los bits de configuración característicos del microcontrolador PIC32MX250F128B [3], luego se declaran las funciones y las variables a utilizar, se configuran los módulos o periféricos a utilizar (Timers, módulo PWM, comunicación UART, interrupciones, entradas, salidas y mapeo de funciones a determinados pines), por último se programa el ciclo principal del programa, donde solamente se ejecuta la programación para el parpadeo de los LED's y se ejecuta la función acorde al comando recibido desde el ordenador (Por ejemplo, un cambio de velocidad en las llantas o la lectura de los *encoders*). Por otro lado, se definen y desarrollan las funciones que se utilizan en el ciclo principal y también las interrupciones (Timers, Change Notice, UART Receiver). Cuando una interrupción se activa, se atiende y luego vuelve al ciclo principal. Cuando se llama a una función desde el ciclo principal, se ejecuta dicha función y luego vuelve al ciclo principal.

8.4.2. Sistema de movimiento

La programación de este sistema se representa con el diagrama de flujo de la Figura 53. Primero se configuran los pines que utiliza este sistema, como se definió anteriormente en la sección 7.3.1, se utilizan los pines 18, 21, 22, 24, 25 y 26, donde todas son salidas digitales.

Los pines 21 y 22 se utilizan para darle sentido a la llanta izquierda y los pines 24 y 25 para darle sentido a la llanta derecha. El estado de estos pines que definen el sentido de giro de las llantas se muestra en el Cuadro 11.

En el pin 18 se utiliza el módulo Output and Compare 3 [3] donde se genera la señal

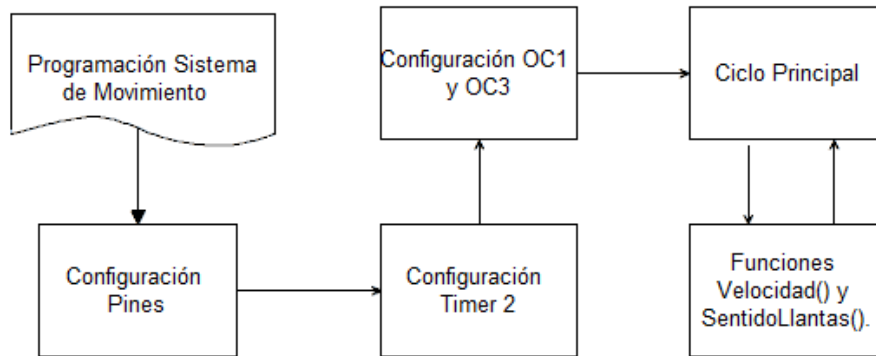


Figura 53: Diagrama de flujo de la configuración del sistema de movimiento.

Pin 21	Pin 22	Pin 24	Pin 25	Llanta Derecha	Llanta Izquierda
L	H	x	x	x	Adelante
H	L	x	x	x	Atrás
x	x	L	H	Adelante	x
x	x	H	L	Atrás	x

Cuadro 11: Configuración de sentido de giro de llantas.

PWM para controlar la velocidad del motor de la llanta izquierda y en el pin 26 se utiliza el módulo Output and Compare 1 [3] para controlar la velocidad del motor de la llanta derecha, por lo tanto, se mapean los módulos OC3 y OC1 a los pines 18 y 26 respectivamente. Para generar las señales PWM en estos módulos se utiliza el Timer 2 como base. Se utilizan señales PWM de 500 Hz para controlar la velocidad de los motores, por lo que el PR2 se calcula utilizando la ecuación 11.

$$PRx = \frac{Freq_{sys}}{TMR_{prescaler} * Freq_{TMR}} - 1 = \frac{40 * 10^6 Hz}{16 * 500 Hz} - 1 = 4999 \quad (11)$$

Donde, PRx es el valor asignado, en este caso, a PR2, $Freq_{sys}$ es la frecuencia del sistema, $TMR_{prescaler}$ es el valor del prescaler, en este caso, del Timer 2 y $Freq_{TMR}$ es la frecuencia en la que se espera que funcione el Timer.

Los módulos Output and Compare 1 y 3 se configuraron para que saquen señales PWM. Para configurar el duty cycle de las señales simplemente se carga un valor entre 0 y 4999 en OC1RS y OC3RS, donde en 0 tendrán un duty cycle de 0% y los motores estarán detenidos, en 4999 tendrán un duty cycle de 100% donde los motores estarán a su máxima velocidad.

Se declararon funciones para cambiar la velocidad de las llantas y el sentido de giro de estas, que son `Velocidad()` y `SentidoLlantas()` respectivamente. Estas funciones no reciben ningún parámetro, y se llaman en el ciclo principal cuando el robot recibe el comando

respectivo desde el ordenador. Estos comandos se detallan en el Cuadro 12 y se envían al robot sin comillas y con los símbolos de retorno de carro y salto de línea (CR-LF) al final. Cabe mencionar que al ejecutar o enviar estos comandos al robot, no dan respuesta alguna de regreso.

Comando	Función
“vmxxxx”	Modifica la velocidad de las llantas. “vixxxx” para la llanta izquierda, “vdxxxx” para la llanta derecha. “xxxx” es el valor que se asigna al módulo OC en un rango de 0 a 4999. Ejemplo: “vd0500” asigna el valor de 500 al módulo OC que controla el motor de la llanta derecha.
“sx”	Modifica el sentido de giro de las llantas. “x” es un número entre 1 y 4. Con 1 van ambas llantas hacia adelante, con 2 la llanta derecha hacia adelante y la llanta izquierda hacia atrás, con 3 la llanta derecha hacia atrás y la llanta izquierda hacia adelante, con 4 ambas llantas hacia atrás.

Cuadro 12: Comandos para modificar la velocidad y sentido de giro de llantas.

8.4.3. Sistema de medición

La programación de este sistema se representa con el diagrama de flujo de la Figura 54. Primero se configuran los pines que utiliza este sistema, como se definió anteriormente en la sección 7.3.2, se utilizan los pines 2, 3, 4 y 5, donde todas son entradas digitales. Los pines 2 y 3 se utilizan para recibir las señales del *encoder* de la llanta derecha y los pines 4 y 5 recibir las señales del *encoder* de la llanta izquierda.

Las señales provenientes de los *encoders* se interpretan así: En el *encoder* de la llanta derecha, cuando el *rising* de la señal B se ejecuta cuando se está recibiendo una señal lógica LOW en la señal A, es porque la llanta derecha tiene un giro hacia adelante, si el *rising* se ejecuta cuando la señal lógica de la señal A es HIGH, es porque la llanta derecha está girando hacia atrás; En el *encoder* de la llanta izquierda, cuando el *rising* de la señal B se ejecuta cuando se está recibiendo una señal lógica LOW en la señal A, es porque la llanta izquierda tiene un giro hacia atrás, si el *rising* se ejecuta cuando la señal lógica de la señal A es HIGH, es porque la llanta izquierda está girando hacia adelante. Por otro lado, para medir la cantidad de pasos o *ticks* dados por cada llanta (Anteriormente se mencionó que las llantas y *encoders* dan 12 pulsos o *ticks* por revolución) se realiza un conteo en una variable de estos pulsos, sumando 1 por cada *rising* de la señal A proveniente de los *encoders*. Para medir el tiempo de cada tick se cuenta (en unidades que equivales a 100us) el tiempo entre cada *rising* de la señal A proveniente de los *encoders*.

Para llevar a cabo el control de las mediciones, se utilizó la interrupción Change Control para detectar un cambio de estado (*rising* y *falling*) en las señales provenientes de los *encoders*. Se utilizó al Timer 1 para realizar la medición del periodo de las señales A, que se reinicia a una frecuencia de 10 KHz, utilizando la misma ecuación 11, se obtuvo que PR1 es igual a 3999 utilizando un prescaler de 1:1 para este Timer.

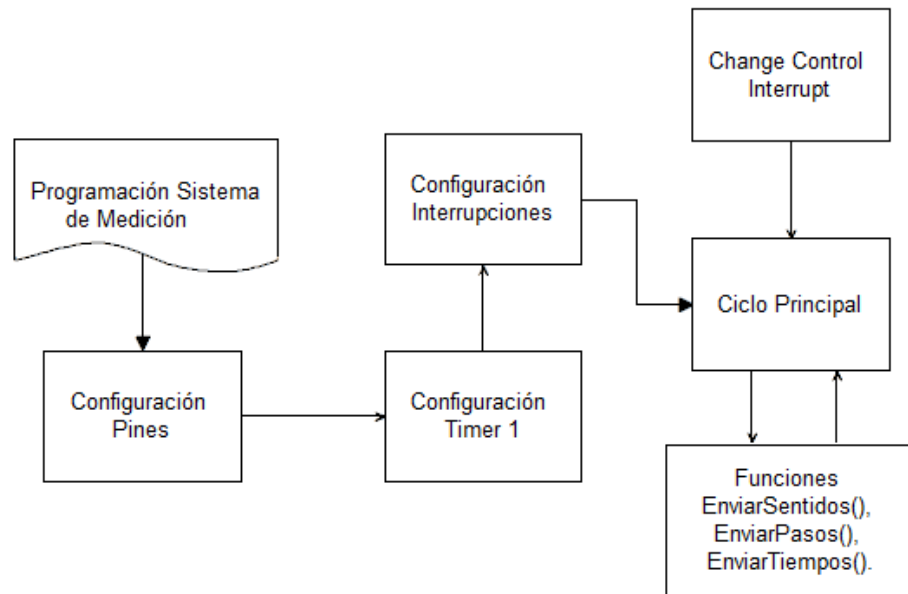


Figura 54: Diagrama de flujo de la configuración del sistema de medición.

Las funciones que se declararon para obtener el sentido de giro de las llantas, el tiempo entre cada *tick* y el conteo de *ticks* por cada llanta son: `EnviarSentido()`, `EnviarTiempo()`, `EnviarPasos()`. Estas funciones no reciben ningún parámetro, y se llaman en el ciclo principal cuando el robot recibe el comando respectivo desde el ordenador. Estos comandos se detallan en el Cuadro 13 y se envían al robot sin comillas y con los símbolos de retorno de carro y salto de línea (CR-LF) al final.

8.4.4. Sistema de detección de objetos

La programación de este sistema se representa con el diagrama de flujo de la Figura 55. Primero se configuran los pines que utiliza este sistema, como se definió anteriormente en la sección 7.3.3, se utilizan los pines 9, 10 y 17, donde las primeras 2 son entradas digitales y el pin 17 es salida digital PWM. Los pines 9 y 10 se utilizan para recibir las señales PWM de los dos ultrasónicos LV-MaxSonar-EZ1 [26], y el pin 17 para controlar la posición del servomotor Traxxas [22].

En el pin 17 se utiliza el módulo Output and Compare 2 [3] donde se genera la señal PWM para controlar la posición del servomotor Traxxas [22], por lo tanto, se mapea el módulo OC2 al pin 17 respectivamente. Para generar la señal PWM en este módulo se utiliza el Timer 3 como base. Se utilizan señales PWM de 300 Hz para controlar la este servomotor como se detalló anteriormente en Marco Teórico, por lo que el PR3 se calcula utilizando la ecuación 12.

Comando	Función	Respuesta
“d”	Solicita el sentido de giro actual de las llantas.	Devuelve 3 caracteres, el primer caracter es el sentido de giro de la llanta derecha, el segundo caracter solo es un espacio separador, el tercer caracter es el sentido de giro de la llanta izquierda. “f” significa hacia adelante y “b” hacia atrás. Ejemplo: “f b” significa que la llanta derecha gira hacia adelante y la izquierda hacia atrás.
“p”	Solicita la cantidad de <i>ticks</i> dados hacia adelante y hacia atrás por cada llanta.	Devuelve 20 caracteres, los primeros 5 caracteres es la cantidad de <i>ticks</i> dados hacia adelante de la llanta derecha, los siguientes 5 caracteres es la cantidad de <i>ticks</i> dados hacia atrás de la llanta derecha y los últimos 5 caracteres es la cantidad de <i>ticks</i> dados hacia atrás de la llanta izquierda. Ejemplo: “01000005001500000050” significa que la llanta derecha recorrió 1000 <i>ticks</i> hacia adelante y 15000 hacia atrás y la llanta izquierda recorrió 500 <i>ticks</i> hacia adelante y 50 hacia atrás.
“t”	Solicita el tiempo en unidades de 100us entre cada <i>tick</i> dado por cada llanta.	Devuelve 10 caracteres, los primeros 5 caracteres es el tiempo actual entre cada <i>tick</i> de la llanta derecha, los últimos 5 caracteres es el tiempo actual entre cada <i>tick</i> de la llanta izquierda. Ejemplo: “0010005000” significa que hay un tiempo de 10ms (100*100us) entre cada <i>tick</i> de la llanta derecha y un tiempo de 500ms (5000*100us) entre cada <i>tick</i> de la llanta izquierda.

Cuadro 13: Comandos para obtener la medición de sentido de giro, conteo de *ticks* y tiempo entre cada *tick* de las llantas.

$$PRx = \frac{Freq_{sys}}{TMR_{Prescaler} * Freq_{TMR}} - 1 = \frac{40 * 10^6 Hz}{8 * 300 Hz} - 1 = 16666 \quad (12)$$

Donde, PRx es el valor asignado, en este caso, a PR3, $Freq_{sys}$ es la frecuencia del sistema, $TMR_{Prescaler}$ es el valor del prescaler, en este caso, del Timer 3 y $Freq_{TMR}$ es la frecuencia en la que se espera que funcione el Timer.

El módulo Output and Compare 2 se configuró para que provea señal PWM. Se realizaron pruebas con el servomotor Traxxas [22] y se determinó que se mueve en un rango de 0 a 146° cuando el duty cycle de la señal PWM de 300 Hz tiene un duty cycle entre el 24% y el 67.8%, por lo tanto, los valores a cargar en OC2RS están en el rango entre 4000 y 11300, donde 4000 lo posiciona en 0° y 11300 en 146°.

Se sabe que el ultrasónico LV-MaxSonar-EZ1 realiza lecturas cada 50ms, por lo tanto, se utiliza la interrupción del Timer 4 para que se active cada 50ms y se registre la lectura de los ultrasónicos. Se utilizó la interrupción del Timer 5, que se activa cada 25us, para medir el tiempo del estado HIGH de la señal PWM proveniente de los ultrasónicos. Utilizando la misma ecuación 12, se determinó que PR4 es 62499 con un prescaler de 1:32 y PR5 es 998

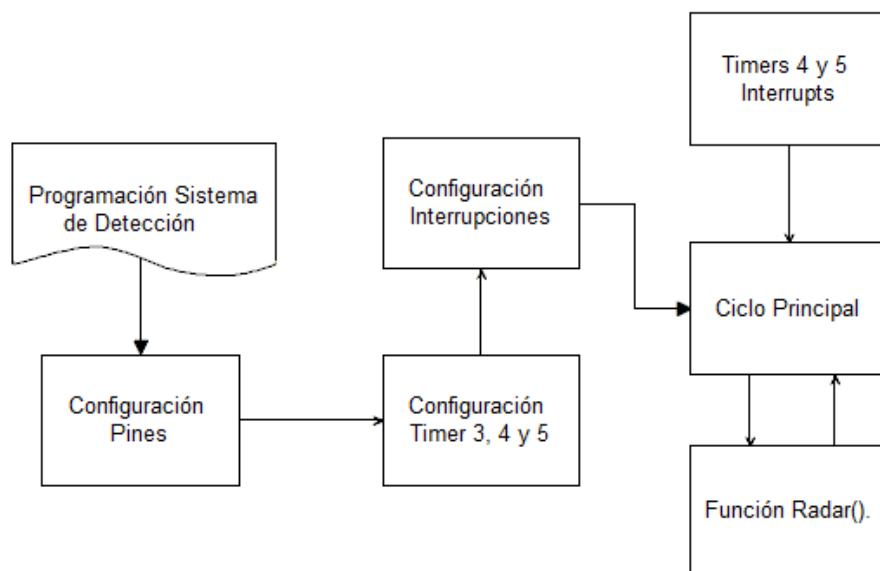


Figura 55: Diagrama de flujo de la configuración del sistema de detección.

con un prescaler de 1:1.

En este sistema se declaró la función Radar() para obtener las lecturas obtenidas de un barrido de 146°, realizando 1 lectura por grado angular. Esta función no reciben ningún parámetro, y se llama en el ciclo principal cuando el robot recibe el comando respectivo desde el ordenador. Este comando se detalla en el Cuadro 14 y se envía al robot sin comillas y con los símbolos de retorno de carro y salto de línea (CR-LF) al final.

Comando	Función	Respuesta
"r"	Solicita las lecturas de los ultrasónicos realizando un barrido de 146° en el plano XY.	Devuelve 1168 caracteres, que son 146 lecturas realizadas por el barrido de detección en el plano XY, que hacen referencia a un barrido entre 0° y 146°, una lectura por grado. Cada lectura es de 8 caracteres, donde los primeros 4 caracteres es el tiempo (en unidades de 25us) actual entre el rising y falling de la señal PWM proveniente del ultrasónico 1, los últimos 4 caracteres es el tiempo actual entre el rising y falling de la señal PWM proveniente del ultrasónico 2. Ejemplo: "05001000" significa que el estado HIGH de la señal PWM del ultrasónico 1 dura 12.5ms (500*25us) y el estado HIGH de la señal PWM del ultrasónico 2 dura 25ms (1000*25us).

Cuadro 14: Comando para realizar un barrido de detecciones de objetos alrededor del robot.

8.4.5. Sistema de comunicación

La programación de este sistema se representa con el diagrama de flujo de la Figura 56. Primero se configuran los pines que utiliza este sistema, como se definió anteriormente en la sección 7.3.4, se utilizan los pines 6 y 7 para la comunicación UART con el módulo ESP-8266 [17], donde el pin 6 es entrada digital por ser el Receptor UART y el pin 7 es salida digital por ser el transmisor UART.

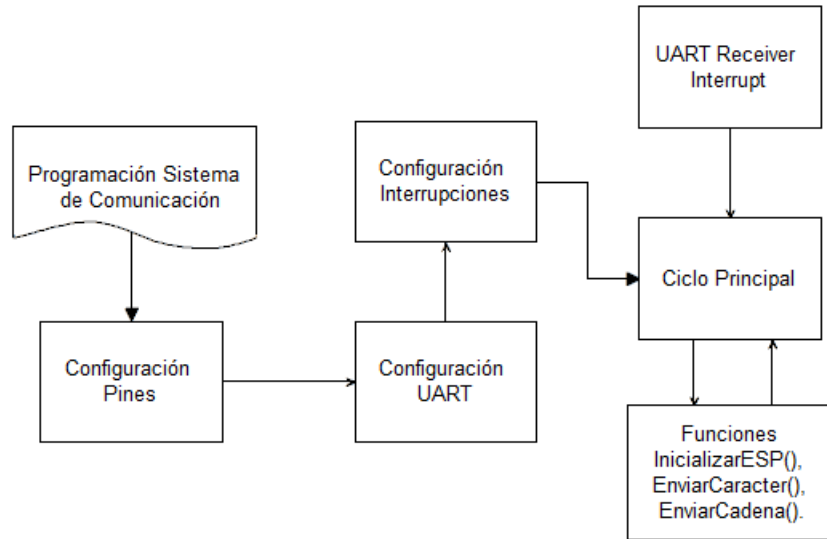


Figura 56: Diagrama de flujo de la configuración del sistema de comunicación.

Para atender de forma inmediata la información o comandos recibidos del módulo ESP-8266 [17], se configuró la interrupción para la recepción del UART. La comunicación entre el módulo ESP-8266 y el microcontrolador se da con un baudrate de 2000000.

En este sistema se declararon las funciones `InicializarESP()`, `EnviarCaracter()` y `EnviarCadena()`. `InicializarESP()` se utiliza para la configuración inicial del módulo ESP-8266 [17], que se encarga de iniciar la comunicación Wi-Fi entre el módulo y el router, y de iniciar al módulo como servidor TCP para la comunicación entre el ordenador y el robot. `EnviarCadena()` sirve para enviar una cadena al módulo ESP-8266, que normalmente son comandos AT [17], y esta función utiliza a `EnviarCaracter()` para enviar un caracter a la vez por comunicación UART. Los comandos AT que se utilizan se muestran en el Cuadro 15.

Comando AT	Función
ATE0	Desactiva el eco serial.
AT+CWMODE_CUR=1	Configura la comunicación Wi-Fi del módulo en modo cliente.
AT+CWJAP_CUR="User";"Pass"	Conecta el módulo a la red Wi-Fi. User es el nombre del router y Pass la contraseña.
AT+CIPMUX=1	Configura al módulo para múltiples conexiones.
AT+CIPSERVER=1,Port	Inicia el servidor TCP. Port es el puerto TCP a utilizar, en este caso es 5400.
AT+CIPSEND=0,X	Prepara al módulo para transmitir datos vía Wi-Fi. X es el número de caracteres a enviar.

Cuadro 15: Comandos AT utilizados para la configuración y comunicación con el módulo ESP-8266 [17].

Diseño de la estructura de la nueva plataforma

En este capítulo se muestra el desarrollo de la nueva plataforma para aplicaciones en robótica de enjambre. Se utilizaron los softwares Inventor para el modelado de la estructura y Altium para el diseño del PCB.

9.1. Dimensiones físicas de la plataforma de la fase 1

La plataforma para robótica de enjambre fase 1 desarrollada el año 2017 en la Universidad del Valle de Guatemala [1], tiene dimensiones físicas de 15.64 cm de ancho, 8.76 cm de alto y 11.49 cm de largo. En la Figura 57 se observa, desde una vista de planta de esta plataforma, el área (incluyendo las llantas) que ocupa en el plano XY , que es de 132.47 cm^2 . También se muestra una figura de este modelo con vista de planta en la Figura 58.

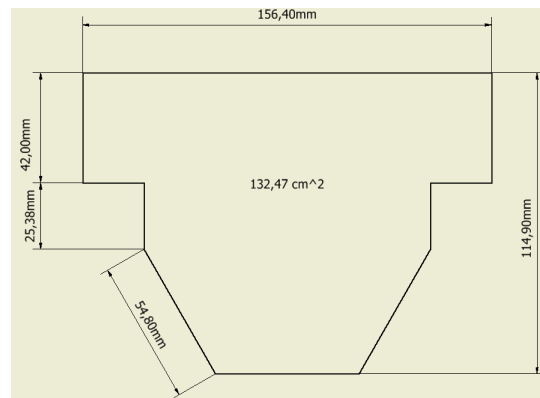


Figura 57: Dimensiones y área ocupada por el modelo anterior en el plano XY desde una vista de planta.

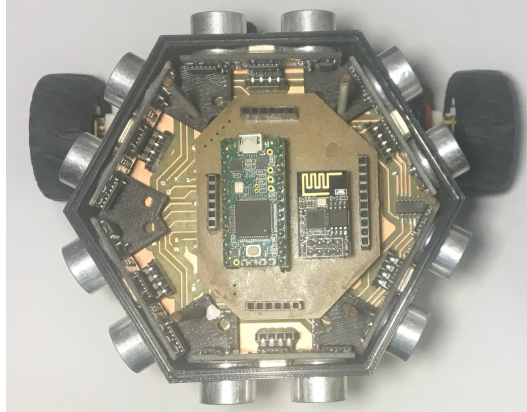


Figura 58: Vista de planta de la plataforma robótica fase 1.

9.2. Diseño de base de la estructura

Lo primero que se diseñó para realizar las pruebas de movimiento fue la base de la nueva plataforma. Las dimensiones de ésta se denota en la Figura 59, y en la Figura 60 se muestra la base acoplada a las llantas con motores y *encoders* desde una vista superior e inferior. El material utilizado para la base es MDF de 3.2mm de grosor. Se puede observar que la base (asumiendo que es un círculo completo) es un círculo de 10cm de diámetro, por lo que ocupa (sin llantas) 78.53 cm^2 . En este nuevo diseño se trató de reducir el espacio físico de la base juntando las llantas y dejando espacio únicamente para la batería Quantum FPV [37] que se utilizará.

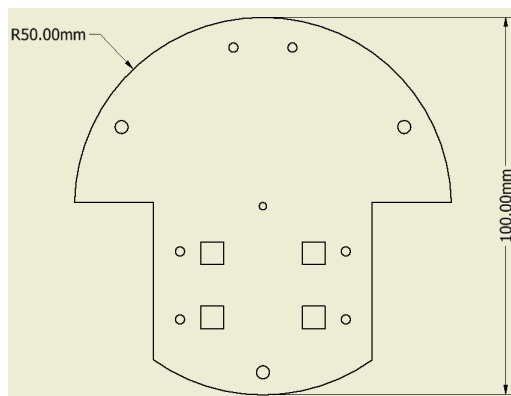


Figura 59: Dimensiones de la base de la nueva plataforma en Inventor.

9.3. Diseño PCB

Para el diseño PCB se utilizó Altium. El PCB se hizo en una placa de cobre de ambos lados, con un grosor de cobre de $1 \text{ oz}/ft^2$. Se utilizó el calculador de grosor de pista que se encuentra en <https://www.desmith.net/NMds/Electronics/TraceWidth.html>, con los parámetros de 1A de corriente máximo, una subida de temperatura de 10°C , una temperatura

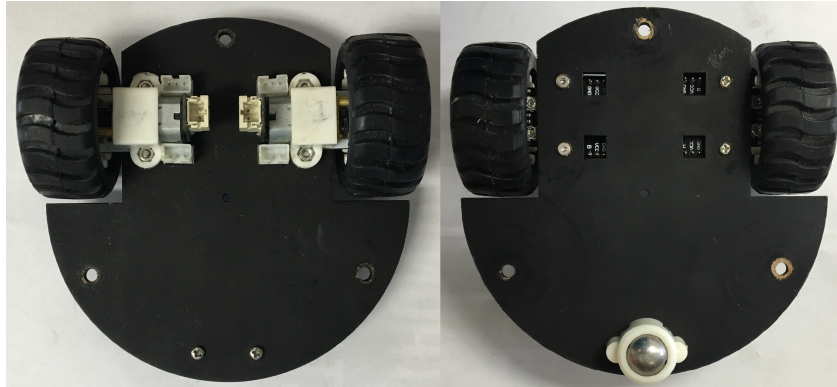


Figura 60: A la izquierda, vista superior de la base con llantas.
A la derecha, vista inferior de la base con llantas.

ambiente de 25°C, un largo máximo de pista de 10cm y un voltaje de 5V, con lo que se obtuvo que había que utilizar un mínimo de 12.13 mils de grosor de pista.

En el trabajo de graduación donde se desarrolló la plataforma robótica de la fase 1 [1], se indicó en recomendaciones que se dejara un ancho de pista mayor al calculado, para garantizar los requerimientos de incremento de temperatura, corriente y caída de voltaje, por lo tanto, se diseñó el PCB con un grosor mínimo de pista de 20 mils. Se utilizó una placa de doble cara de cobre para colocar en la parte inferior los headers donde se conectarán los componentes que irán en la base inferior de robot, estos son la batería, los *encoders*, los motores, el servomotor y los ultrasónicos (estos últimos irán en la parte superior, pero sus cables se lanzarán hacia la parte inferior para que no estorben a la hora de realizar un barrido de detección de objetos). El diseño simulado del PCB con componentes se muestra en la Figura 61, desde una vista isométrica, cabe mencionar que en esta vista no se muestra el orificio donde irá incrustado el servomotor y donde pasarán los cables de los ultrasónicos hacia la parte inferior.

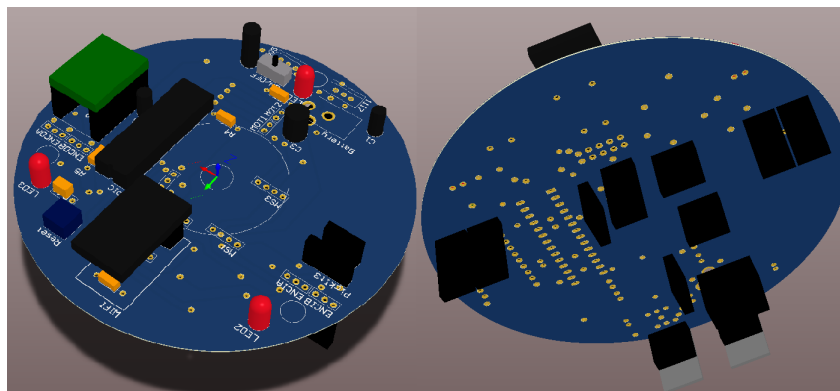


Figura 61: A la izquierda, vista superior simulada de la placa con componentes.
A la derecha, vista inferior. Simulación realizada en Altium.

En la Figura 62 se observa la placa física y soldada con sus componentes, donde en la parte superior se aprecia el orificio donde se incrusta el servomotor y los orificios donde pasan los cables de los ultrasónicos y en la parte inferior se observa la placa con el servomotor, los ultrasónicos, el microcontrolador, el driver de motores y el módulo Wi-Fi puestos.

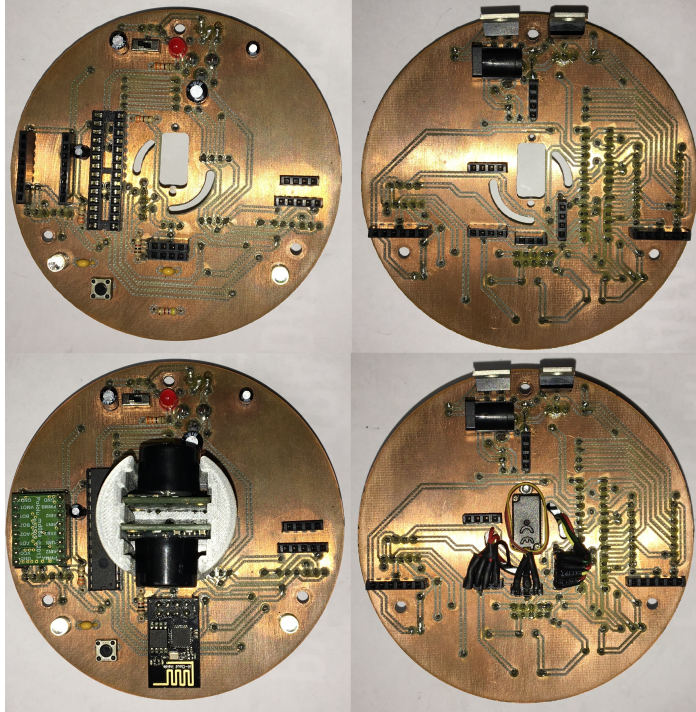


Figura 62: Arriba a la izquierda, vista superior de la placa con componentes soldados. Arriba la derecha, vista inferior. Abajo a la izquierda, vista superior de la placa con módulos, servomotor y ultrasónicos incorporados. Abajo a la derecha, vista inferior con las incorporaciones.

9.4. Estructura final

Teniendo la placa y la base hechas, se realizaron las conexiones de los componentes entre estas, que se pueden observar en la Figura 63. Para sujetar la placa y la base se utilizaron tornillos sin cabeza de 3mm de grosor con tuercas y *washas* de presión. Como separadores se utilizaron tubos de PLA que se fabricaron en impresoras 3D. El ensamblaje final se puede observar en la Figura 64, donde están las vistas de planta, elevación, perfil e isométrica de la nueva plataforma ensamblada. En la Figura 65 se observa la ubicación del switch de encendido y apagado, el push-button de reset, los pines de programación, los pines donde se pueden leer las señales provenientes de los *encoders* de las llantas, la unidad de control PIC32MX250F128B [3], los dos ultrasónicos LV-MaxSonar-EZ1 [26] y el módulo Wi-Fi ESP-8266 [17].

9.5. Costo de la nueva plataforma

En el Cuadro 16 se detalla el costo de todas las partes del prototipo desarrollado, donde se observa que el costo total es de \$ 125.79. Los ultrasónicos LV-MaxSonar-EZ1 son los más caros, costando \$ 59.90 (ambos ultrasónicos) lo que representa el 47.62% del costo total.

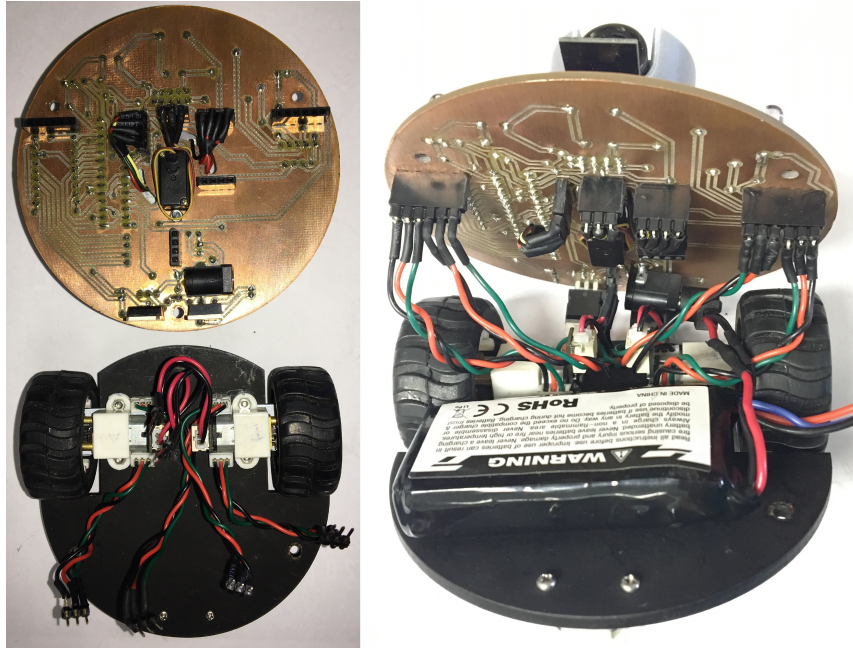


Figura 63: Conexión de los componentes del robot entre base y placa.

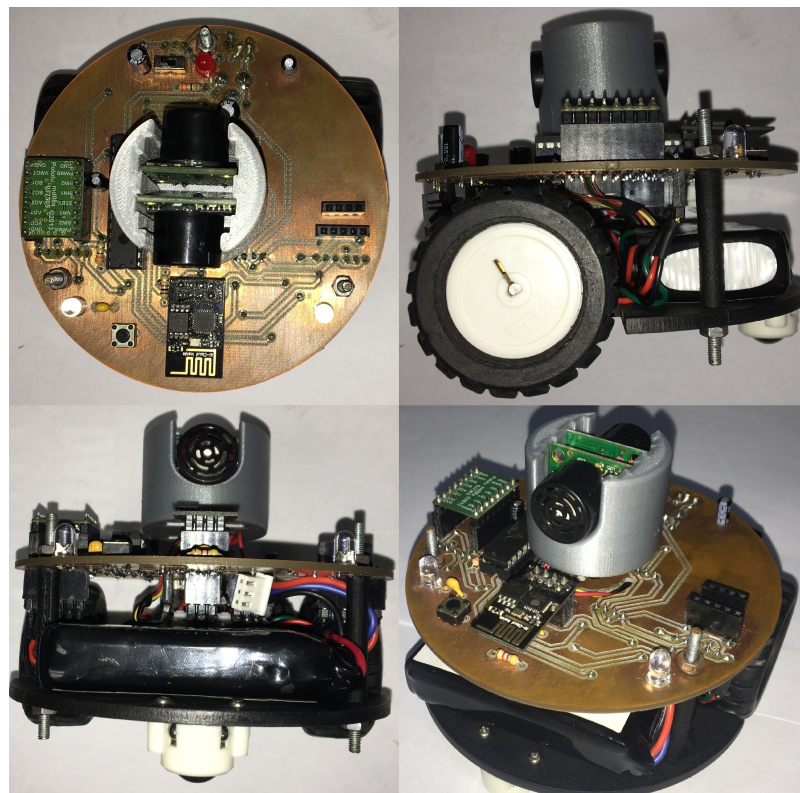


Figura 64: Vistas de planta, elevación, perfil e isométrica de la nueva plataforma ensamblada.

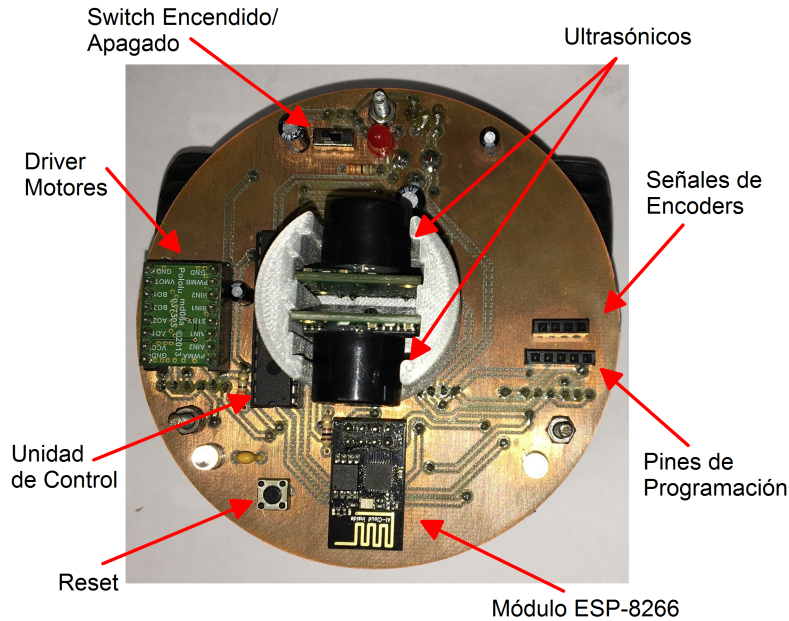


Figura 65: Señalización de componentes relevantes sobre la placa de cobre.

Cantidad	Componente	Costo
1	PIC32MX250F128B [3]	\$ 2.76
2	Micro Metal Gear Motor [15]	\$ 23.00
2	Llantas 42x19mm [13]	\$ 2.90
1	DRV8833 Dual Motor Driver [32]	\$ 3.33
2	Encoder SEN0116 [34]	\$ 9.00
2	LV-MaxSonar-EZ1 [26]	\$ 59.90
1	Traxxas Servo [22]	\$ 4.50
1	ESP-8266 [17]	\$3.00
1	Batería Quanam FPV 7.4V [37]	\$ 9.54
3	LED	\$ 0.58
1	Regulador L7805CV 5V [38]	\$ 0.50
1	Regulador AZ1117 3.3V [39]	\$ 0.29
1	Otros (Resistencias, capacitores, tornillos, etc.)	\$ 6.49
Costo Total		\$ 125.79

Cuadro 16: Detalle de costos de la nueva plataforma.

9.6. Comparación de plataformas fase 1 y 2

Ahora que ya se tienen las dos plataformas (fase 1 y 2) para robótica de enjambre finales al alcance, en la Figura 66 se muestran ambas en vista de planta, elevación y perfil, una al lado de la otra, donde es notoria la reducción de las dimensiones físicas, sobre todo en la vista de planta (la que se encuentra hasta arriba de la figura), el área que ocupa la nueva plataforma en el plano XY es mucho menor al de la plataforma de la fase 1. La nueva

plataforma tiene forma cilíndrica (despreciando las partes que sobresalen de las llantas) de 10cm de diámetro y 8.36 cm de alto. En la Figura 67, se observan las dimensiones exactas de la nueva plataforma en el plano XY incluyendo las llantas, se observa que ocupa un área de 84.2 cm^2 . Como se detalló anteriormente en las dimensiones de la plataforma de la fase 1, esta ocupa un área de 132.47 cm^2 . Por lo tanto para comparar cuantitativamente la reducción de espacio físico utilizado entre las plataformas se utilizó la ecuación 13, donde se comprueba que se obtuvo una reducción del 36.44 % en el espacio ocupado del nuevo diseño mecánico de la plataforma.

$$\%Reducción = \left(\frac{Área_{np}}{Área_{vp}} - 1 \right) * 100\% = \left(\frac{84.2\text{cm}^2}{132.47\text{cm}^2} - 1 \right) * 100\% = -36.44\% \quad (13)$$

Donde $\%Reducción$ es el porcentaje de reducción, $Área_{np}$ es el área que ocupa la nueva plataforma en el plano XY , $Área_{vp}$ es el área que ocupa la vieja plataforma en el plano XY .

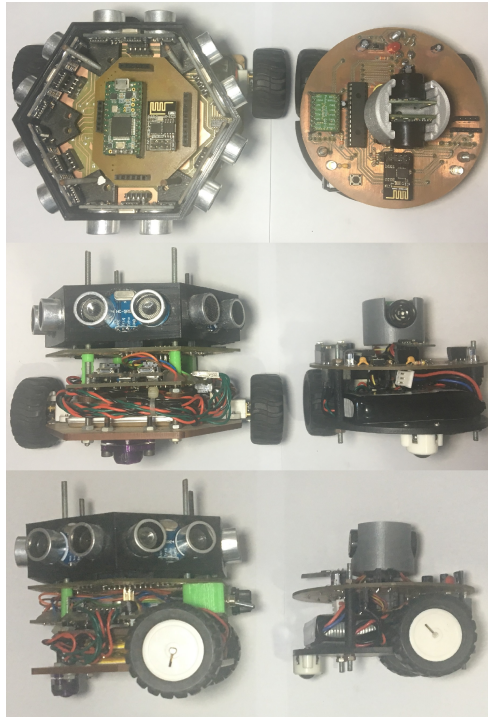


Figura 66: Comparación de plataformas de robótica de enjambre fase 1 y 2 en vistas de planta, elevación y perfil.

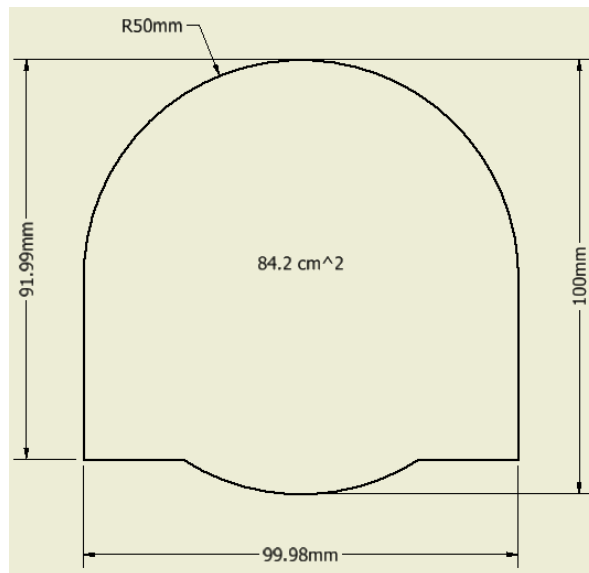


Figura 67: Dimensiones del área ocupada por la nueva plataforma en el plano XY .

Implementación de librerías en Matlab

Para el control de la plataforma robótica de enjambre de la fase 1 [1] desde un ordenador, se utilizó Matlab. La librería que se desarrolló se utilizó como referencia para implementar nuevas funciones que controlen la nueva plataforma. Se implementó un archivo de matlab llamado “SwarmLib.m” que contiene un total de 13 funciones. Este archivo se encuentra en Anexos.

Las características de las funciones implementadas se observan en el Cuadro 17, donde se detallan los parámetros de cada función y la información que devuelven. Cabe mencionar que se utilizaron funciones que solo corren en Matlab 2018. Para utilizar todas estas funciones es necesario importarlas en el script de Matlab, utilizando el comando *import SwarmLib.**.

Para generar una conexión con el robot, se debe instanciar un objeto TCP utilizando la función *ConexinTCP()*, luego de esto ya se pueden utilizar las funciones que controlan al robot. Antes de instanciar el objeto TCP, en dado caso se desconoce la IP del robot, se utiliza la función *GetIP()* que solo necesita el puerto TCP, que se configuró como 5400, y el nombre del robot, que para el prototipo desarrollado es “Robot 1”. Esta función realiza pruebas con todas las direcciones IP’s entre 192.168.1.0 y 192.168.1.255, tomando 1 segundo para probar la conexión con cada una, por lo tanto, si la dirección que se asignó al robot es la 192.168.1.240, se tardará 4 minutos en obtener la IP del robot, por lo que se recomienda utilizar una sola vez esta función y luego guardar en una variable la dirección IP que tiene el robot, para que cada vez que se reinicie un programa desarrollado en Matlab no haya que esperar 4 minutos para encontrar la IP.

Cabe mencionar que una mejora obtenida en esta librería desarrollada para controlar al nuevo modelo de robot respecto a la librería desarrollada en la fase 1, es la función ya mencionada *GetIP()*, que permite encontrar la IP del robot automáticamente en Matlab, ya que para la plataforma anterior era necesario utilizar un programa aparte para encontrar dicha IP del robot.

Función	Descripción
GetIP(Nombre, Puerto)	Obtiene la IP del robot. Pide el nombre del robot del que se desea obtener la IP y el puerto TCP. Devuelve la IP del robot. Si devuelve "192.168.1.255" es porque no pudo encontrar al robot.
ConexionTCP(IPRobot, Puerto)	Instancia la conexión TCP con el robot. Pide la IP del robot y el puerto TCP. Devuelve el objeto TCP instanciado.
EnviarComando(Com, ObjTCP)	Envía un comando o texto directamente al robot. Pide el comando o texto a enviar. No devuelve ningún valor.
GetNombre(ObjTCP)	Obtiene el nombre de un robot. Pide le objeto TCP de conexión con el robot. Devuelve el nombre del robot.
SetVelocidad(Vel, Lla, ObjTCP)	Configura la velocidad de las llantas del robot. Pide la velocidad (valor entre 0 y 100), la llanta (1 para la derecha y 2 para la izquierda) que se configura y el objeto TCP de conexión. No devuelve ningún valor.
SetSentidos(Sentido, ObjTCP)	Configura el sentido de las llantas del robot. Pide el sentido de las llantas (1 adelante ambas, 2 derecha adelante e izquierda atrás, 3 derecha atrás e izquierda adelante, 4 ambas atrás). No devuelve ningún valor.
GetSentidos(ObjTCP)	Obtiene el sentido de giro actual de las llantas. Pide el objeto TCP de conexión. Devuelve dos caracteres, el primer caracter representa el sentido de giro de la llanta derecha y el segundo caracter el sentido de la llanta izquierda. "f" significa hacia adelante y "b" hacia atrás.
GetPasos(ObjTCP)	Obtiene los pasos o "ticks" dados por las llantas. Pide el objeto TCP de conexión. Devuelve un vector con 4 números, el primero y segundo son los pasos de la llanta derecha hacia adelante y hacia atrás, los otros dos son los pasos de la llanta izquierda hacia adelante y hacia atrás.
GetTiempos(ObjTCP)	Obtiene el tiempo entre los últimos dos "ticks" que el encoder detecta. Pide el objeto TCP de conexión. Devuelve un vector con 2 números, el primero es el tiempo actual entre pasos de la llanta derecha y el segundo de la llanta izquierda. Los tiempos están dados en segundos.
Pasos2mt(Pasos)	Convierte los pasos o "ticks" a distancia en metros. Pide la cantidad de pasos a convertir. Devuelve la distancia equivalente en metros.
Pasos2cm(Pasos)	Convierte los pasos o "ticks" a distancia en centímetros. Pide la cantidad de pasos a convertir. Devuelve la distancia equivalente en centímetros.
Tiempos2Vel(Tiempo)	Convierte los tiempos entre "ticks" a velocidad. Pide el tiempo entre "ticks". Devuelve la velocidad en cm/s.
GetDetecciones(ObjTCP)	Función para obtener las lecturas de los ultrasónicos. Pide el objeto TCP de conexión. Devuelve dos vectores, el primero contiene las distancias en metros de 292 lecturas, el segundo vector contiene el ángulo, en radianes, respectivo a cada lectura.

Cuadro 17: Funciones de librería de Matlab para controlar a la nueva plataforma.

Pruebas con la nueva plataforma

Con las funciones implementadas descritas en el capítulo anterior se desarrolló un programa de prueba en Matlab, que permitió realizar las pruebas de los sistemas de detección, movimiento y medición de la plataforma.

11.1. Sistema de detección

Para mostrar el barrido de detección de objetos que realiza la plataforma en el plano XY se colocó el robot al centro de una barrera circular de 93.5cm de diámetro y 15cm de alto, estando a más altura que los ultrasónicos del robot para que pueda ser detectada sin ningún problema. Los sensores LV-MaxSonar-EZ1 [26] emiten las ondas ultrasónicas a 1.75cm del centro del robot, por lo tanto, entre un sensor ultrasónico y la barrera circular hay 45cm de distancia. En la Figura 68, al lado izquierdo se muestra la forma en que se colocaron la barrera y el robot, al lado derecho una gráfica polar muestra las lecturas de los ultrasónicos en metros, donde cada punto es una lectura hecha por cada sensor y la orientación angular es antihoraria, por ser la orientación en que el robot realiza la detección. Como se mencionó anteriormente, el servomotor Traxxas [22] gira 146° , lo que provoca que haya una zona fuera del área de detección de 68° en el plano XY , donde se reparten 34° de zona muerta a cada lado del robot, es decir, los ultrasónicos no detectan objetos entre $343^\circ - 17^\circ$ y entre $163^\circ - 194^\circ$ del robot, tomando 0° a la derecha del robot y yendo en sentido antihorario, por lo tanto, las zonas de detección del robot son entre $17^\circ - 163^\circ$ y entre $197^\circ - 343^\circ$. Observando las lecturas realizadas, nuevamente en la Figura 68, se puede notar que varían entre aproximadamente 42cm y 44cm, tomando como fuente de error que la forma de la barrera no es perfectamente circular y que el sensor LV-MaxSonar-EZ1 tiene una incertidumbre de ± 2.54 cm, a esta incertidumbre se le puede agregar que, como se detalló anteriormente en la sección 7.4.4, el comando utilizado para realizar las lecturas de los ultrasónicos devuelve el tiempo de la señal

PWM proveniente de los sensores en unidades de 25us, lo que da una incertidumbre de $\pm 0.2\text{cm}$, y esto sumado a la incertidumbre del sensor, es un total de $\pm 2.74\text{cm}$ de incertidumbre en las lecturas.

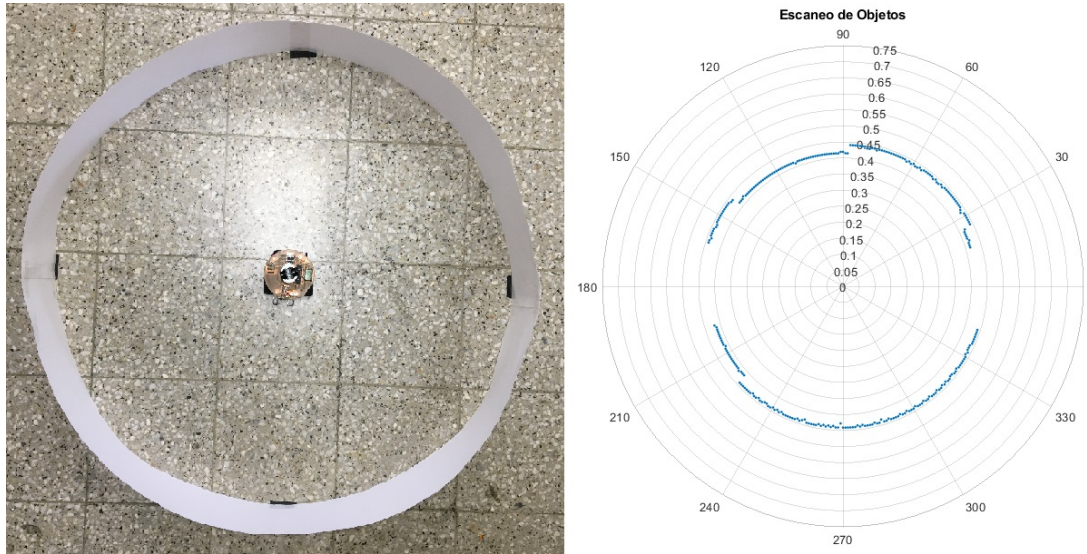


Figura 68: Prueba de detección de objetos, con una barrera circular de 93.5cm de diámetro y 15cm de alto alrededor del robot.

Posteriormente se colocaron cilindros de 1.5cm, 3cm y 5cm de diámetro, todos con 15cm de alto al igual que la barrera, dentro del área de las pruebas de detección. En la Figura 69 se puede observar que se colocaron estos cilindros, de tal forma que cada sensor ultrasónico detectara tres cilindros de distintos tamaños. Todos los cilindros se colocaron a 15cm de la barrera, es decir a 30cm de los sensores. Los cilindros de 1.5cm se colocaron a 90° y 270° , los de 3cm se colocaron a 50° y 310° y los de 5cm a 130° y 230° . En la misma figura, del lado derecho, se observa la gráfica polar con las detecciones del robot, se puede notar que muestra lecturas muy cercanas a 30cm en las posiciones angulares donde se colocaron los cilindros, lo que denota muy buenas lecturas respecto a las distancias de los objetos, sin embargo, en la gráfica se observa como si casi no hubiera espacio entre estos objetos, esto se debe a que las ondas ultrasónicas emitidas no salen en línea recta del sensor, sino que estas se propagan en un determinado ángulo sólido (que no se especifica en la hoja de datos del MaxSonar), lo que ocasiona que mientras más lejos está un objeto, lo detectará con dimensiones cada vez mayores.

Luego se realizaron pruebas colocando solamente los cilindros de 1.5cm y 3cm, los cilindros de 1.5cm y 5cm, por último los cilindros de 3cm y 5cm respectivamente como se muestran en las Figuras 70, 71 y 72. En estas pruebas se quitaron los cilindros de las posiciones de 90° y 270° . Observando las gráficas realizadas con las lecturas en estas tres pruebas, aquí ya se puede apreciar el espacio entre los cilindros. La lecturas indican nuevamente que los objetos están aproximadamente a 30cm de los sensores.

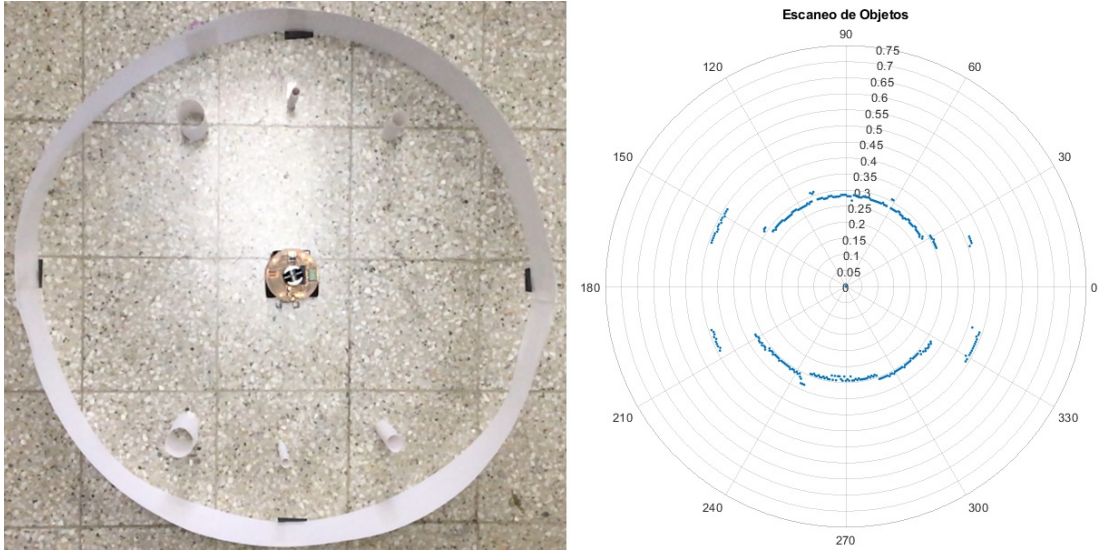


Figura 69: Prueba de detección de cilindros de 1.5cm, 3cm y 5cm dentro de la barrera circular.

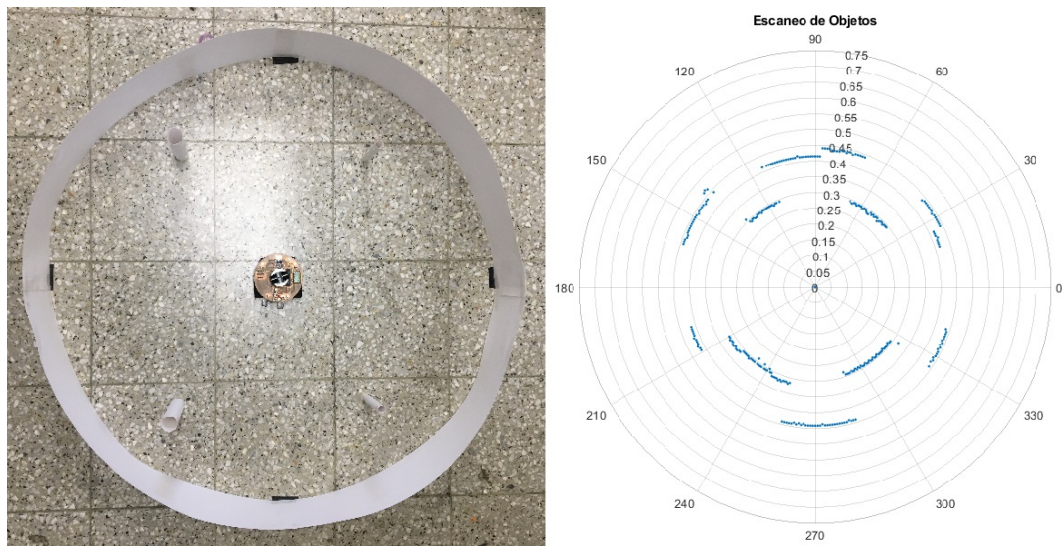


Figura 70: Prueba de detección de cilindros de 1.5cm y 3cm dentro de la barrera circular.

11.2. Sistemas de movimiento y medición

Se realizaron pruebas cambiando las velocidades de las llantas y a la vez se fueron registrando las lecturas de los codificadores para medir la velocidad de las llantas. Para cambiar las velocidades de las llantas se utilizó la instrucción que cambia el periodo de trabajo de las señales PWM que controlan la velocidad de los motores. Se variaron las señales entre 0% y 100% con saltos del 5% del periodo de trabajo de las señales PWM. Para cada cambio de velocidad se registraron las señales de los codificadores para calcular las velocidades. En total fueron 42 cambios de velocidad de las dos llantas, por lo tanto, se registraron 42 lecturas de los codificadores. En la Figura 73 se muestran las velocidades lineales de las llantas respecto al periodo de trabajo asignado. Para ambas gráficas se calcularon las regresiones

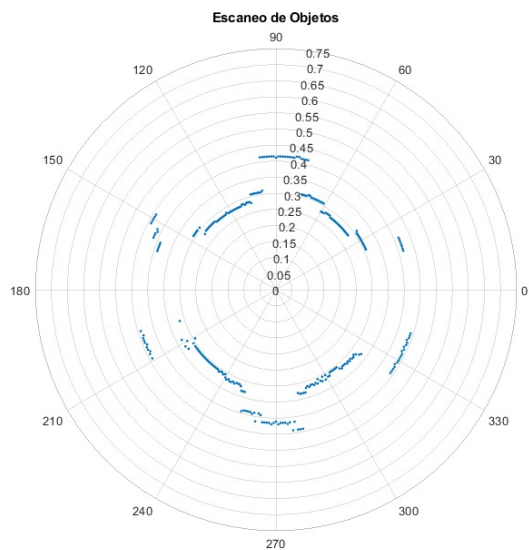


Figura 71: Prueba de detección de cilindros de 1.5cm y 5cm dentro de la barrera circular.

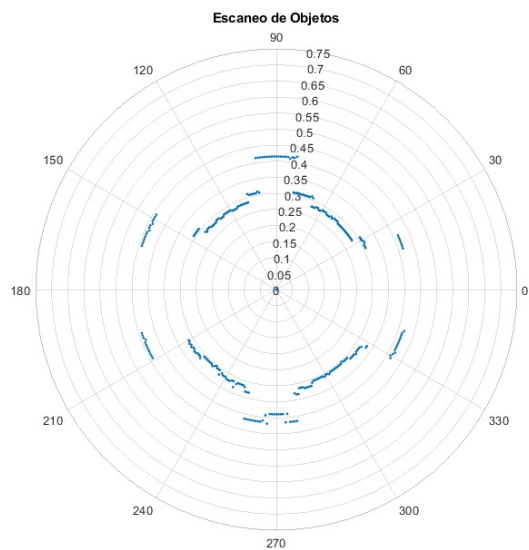


Figura 72: Prueba de detección de cilindros de 3cm y 5cm dentro de la barrera circular.

lineales junto con su ecuación, donde se obtuvo un R^2 de 0.9966 para la llanta derecha y un R^2 de 0.9959 para la llanta izquierda. La llanta derecha varía 0.535cm/s por cada unidad porcentual del periodo de trabajo de la señal PWM y la llanta izquierda varía 0.542cm/s por cada unidad porcentual.

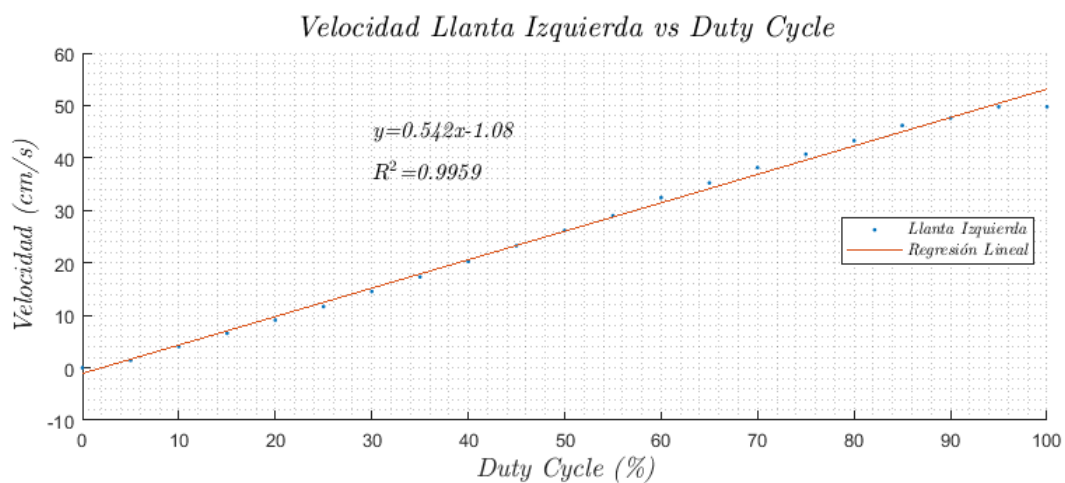
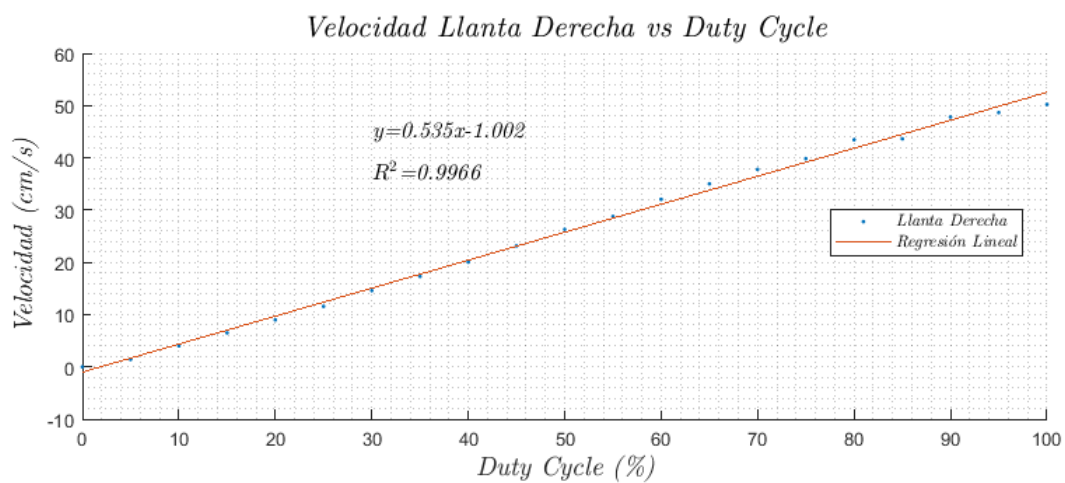


Figura 73: Gráficas de velocidades de llantas contra el duty cycle de las señales PWM que controlan sus velocidades.

11.3. Pruebas de temperatura

Durante las pruebas de funcionamiento en la nueva plataforma, se observó que algunos componentes se estaban calentando a altas temperaturas, por lo tanto, para descartar que esto afectara su funcionamiento se realizaron pruebas con una cámara térmica para observar la temperatura en los componentes, específicamente en el regulador de voltaje de 3.3V y el módulo Wi-Fi. En la Figura 74 se muestran tomas de calor al robot cuando no está en funcionamiento, a la izquierda se observa una toma desde planta y a la derecha una toma trasera del robot. La temperatura máxima que se observa en esta figura es de 29°C.

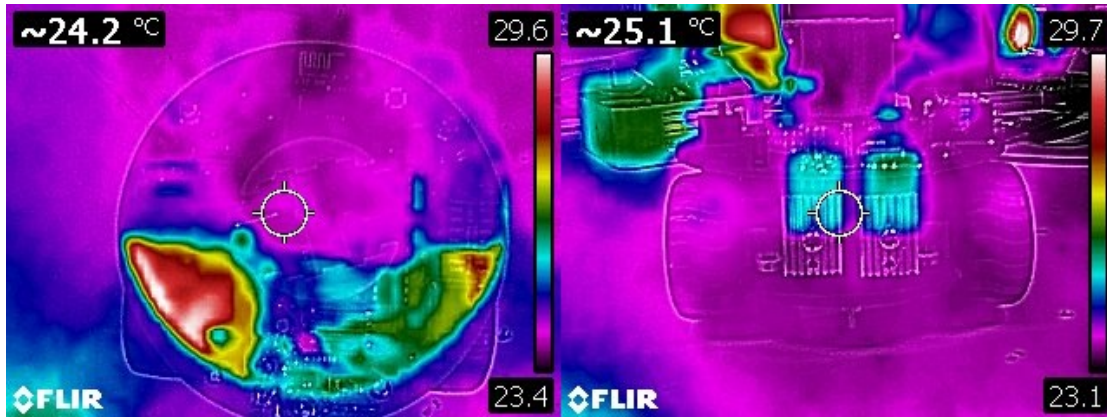


Figura 74: Capturas de temperatura en la nueva plataforma cuando no está en funcionamiento. A la izquierda, vista de planta y a la derecha, vista trasera del robot mostrando los reguladores de voltaje.

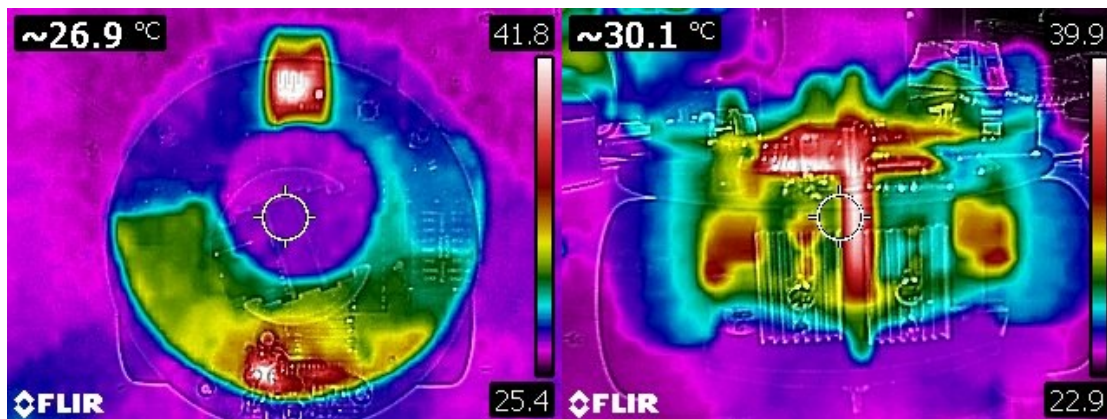


Figura 75: Capturas de temperatura en la nueva plataforma luego de 20 minutos de funcionamiento. A la izquierda, vista de planta y a la derecha, vista trasera del robot mostrando los reguladores de voltaje.

Para observar las temperaturas máximas que pueden alcanzar los componentes anteriormente mencionados, se colocó a funcionar el robot con una rutina implementada en matlab con la librería desarrollada para que los motores de las llantas giren a velocidad máxima, se manden comandos y se accione el radar cada cierto tiempo. Luego de 20 minutos de funcionamiento, se tomaron fotos de calor que se muestran en la Figura 75, donde se observa efectivamente que los componentes que más se calientan son el módulo Wi-Fi y el regulador

de voltaje de 3.3V a una temperatura aproximada de 41.8°C como máximo.

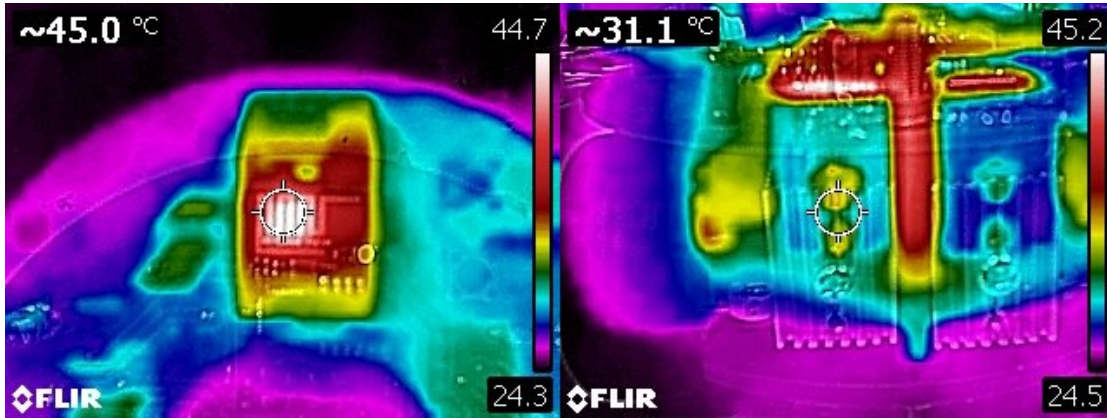


Figura 76: Capturas de temperatura luego de 20 minutos de funcionamiento. A la izquierda, módulo Wi-Fi y la derecha, regulador de 3.3V.

En la Figura 76 se observa de cerca la temperatura del módulo Wi-Fi ESP-8266 y del regulador AZ1117-3.3, donde sus temperaturas son de 45°C aproximadamente. En el datasheet del regulador de voltaje AZ1117-3.3 [39] se encuentra que la temperatura de funcionamiento de este componente se encuentra entre -40°C y 125°C, la gráfica que muestra cómo se comporta el voltaje respecto a la temperatura se encuentra en la Figura 77, que muestra que no hay problema alguno con que el regulador alcance los 45°C de temperatura. Por otro lado, en el datasheet del módulo Wi-Fi ESP-8266 [17] se muestra que la temperatura de funcionamiento correcto del módulo es entre -20°C y 100°C, por lo tanto no hay ningún problema con ninguno de los dos componentes que se más se calientan durante el funcionamiento de la nueva plataforma robótica.

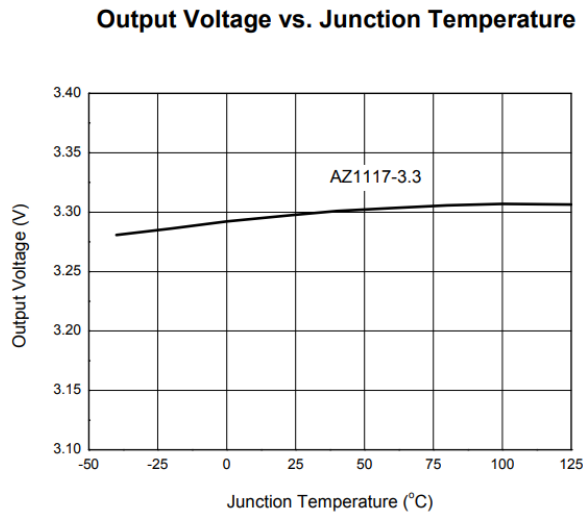


Figura 77: Voltaje de salida del regulador AZ1117-3.3 en función de la temperatura [39].

- Se obtuvo una reducción del 36.44 % en las dimensiones físicas del plano XY entre la plataforma de la fase 1 y la nueva.
- Se logró utilizar al microcontrolador PIC32MX250F128B como unidad de control de la nueva plataforma desarrollada para aplicaciones en robótica de enjambre, sustituyendo al Teensy 3.2 utilizado en la fase 1.
- Con el nuevo sistema de detección de objetos alrededor de la plataforma robótica, constituido por dos ultrasónicos y un servomotor, se logró realizar barridos de detección de objetos en el plano XY , con una zona de detección de 292° .
- Se implementó una librería en Matlab con 13 funciones definidas para controlar la nueva plataforma robótica desde un ordenador con comunicación Wi-Fi.
- La nueva plataforma consume una potencia total de 2.63 Watts.
- El prototipo final de la nueva plataforma tiene un costo total de \$ 125.79.

- Se recomienda utilizar componentes con empaquetado SMD para obtener un diseño de PCB con dimensiones físicas aún más reducidas.
- Se recomienda utilizar sensores ultrasónicos de mayor frecuencia de muestreo para minimizar el tiempo en que se realiza el barrido de detección de objetos alrededor de la plataforma.
- Las llantas y la batería son las que limitan reducir aún más las dimensiones físicas de la estructura de la nueva plataforma, por lo tanto, se recomienda utilizar llantas y baterías con menores dimensiones físicas, lo que permitirá una mayor reducción del tamaño de la plataforma.
- Se recomienda utilizar ultrasónicos con el menor ángulo sólido de detección posible, para obtener mejores resultados en las lecturas.
- Los ultrasónicos representan el 47.62 % del costo total de la nueva plataforma, por lo tanto, se recomienda utilizar otros ultrasónicos que sean más baratos para poder obtener una plataforma para aplicaciones en robótica de enjambre más barata.

-
- [1] E. I. Hernández Woc, J. O. Del Cid Guzmán, J. J. Mérida Rodríguez, O. G. Wantland Conde, V. R. Villegas Zabala, W. A. Orozco Cifuentes y R. A. Ajtún Bulux, “Reingeniería de Megaproyectos Fase 1”, Tesis doct., Universidad del Valle de Guatemala, Ciudad de Guatemala, 27 de nov. de 2017, 237 págs.
 - [2] Arduino. (2018). Arduino - Introduction, dirección: <https://www.arduino.cc/en/Guide/Introduction> (visitado 17-09-2018).
 - [3] Microchip. (2018). PIC32MX250F128B - Microcontrollers and Processors - Microcontrollers and Processors, dirección: <https://www.microchip.com/wwwproducts/en/PIC32MX250F128B> (visitado 15-09-2018).
 - [4] F. Arvin, K. Samsudin y A. R. Ramli, “Development of a miniature robot for swarm robotic application”, *International Journal of Computer and Electrical Engineering*, págs. 436-422, 2009, ISSN: 17938163. DOI: 10.7763/IJCEE.2009.V1.67. dirección: <http://www.ijcee.org/index.php?m=content&c=index&a=show&catid=25&id=63> (visitado 13-09-2018).
 - [5] A. Özgür, S. Lemaignan, W. Johal, M. Beltran, M. Briod, L. Pereyre, F. Mondada y P. Dillenbourg, “Cellulo: Versatile handheld robots for education”, en *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction - HRI '17*, Vienna, Austria: ACM Press, 2017, págs. 119-127, ISBN: 978-1-4503-4336-7. DOI: 10.1145/2909824.3020247. dirección: <http://dl.acm.org/citation.cfm?doid=2909824.3020247> (visitado 13-09-2018).
 - [6] Estier, Caprari y Siegart, “Fascination of down scaling — alice the sugar cube robot”, *Journal of Micromechatronics*, vol. 1, n.º 3, págs. 177-189, 1 de jul. de 2001, ISSN: 1389-2258, 1568-5632. DOI: 10.1163/156856301760132097. dirección: <http://booksandjournals.brillonline.com/content/journals/10.1163/156856301760132097> (visitado 13-09-2018).
 - [7] S. Kernbach, R. Thenius, O. Kernbach y T. Schmickl, “Re-embodiment of honey-bee aggregation behavior in an artificial micro-robotic system”, *Adaptive Behavior*, vol. 17, n.º 3, págs. 237-259, jun. de 2009, ISSN: 1059-7123, 1741-2633. DOI: 10.

- 1177/1059712309104966. dirección: <http://journals.sagepub.com/doi/10.1177/1059712309104966> (visitado 13-09-2018).
- [8] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano y A. Martinoli, “The e-puck, a robot designed for education in engineering”, pág. 7,
- [9] A. Turgut, F. Gökçe, H. Çelikkanat, L. Bayindir y E. Sahin, “Kobot: A mobile robot designed specifically for swarm robotics research”, pág. 13, ene. de 2007.
- [10] M. Rubenstein, C. Ahler y R. Nagpal, “Kilobot: A low cost scalable robot system for collective behaviors”, en *2012 IEEE International Conference on Robotics and Automation*, St Paul, MN, USA: IEEE, mayo de 2012, págs. 3293-3298, ISBN: 978-1-4673-1405-3 978-1-4673-1403-9 978-1-4673-1578-4 978-1-4673-1404-6. DOI: 10.1109/ICRA.2012.6224638. dirección: <http://ieeexplore.ieee.org/document/6224638/> (visitado 14-09-2018).
- [11] J. McLurkin, J. Rykowski, M. John, Q. Kaseman y A. J. Lynch, “Using multi-robot systems for engineering education: Teaching and outreach with large numbers of an advanced, low-cost robot”, *IEEE Transactions on Education*, vol. 56, n.º 1, págs. 24-33, feb. de 2013, ISSN: 0018-9359, 1557-9638. DOI: 10.1109/TE.2012.2222646. dirección: <http://ieeexplore.ieee.org/document/6363493/> (visitado 14-09-2018).
- [12] F. Arvin, J. Murray, C. Zhang y S. Yue, “Colias: An autonomous micro robot for swarm robotic applications”, *International Journal of Advanced Robotic Systems*, vol. 11, n.º 7, pág. 113, 15 de jul. de 2014, ISSN: 1729-8814, 1729-8814. DOI: 10.5772/58730. dirección: <http://journals.sagepub.com/doi/10.5772/58730> (visitado 13-09-2018).
- [13] DFRobot. (2018). Wheel 42x19mm (Pair)-DFRobot, dirección: <https://www.dfrobot.com/product-353.html> (visitado 19-09-2018).
- [14] Core-Electronics. (2018). 250:1 micro metal gearmotor HP, Core Electronics, dirección: <https://core-electronics.com.au/250-1-micro-metal-gearmotor-hp.html> (visitado 17-09-2018).
- [15] DFRobot. (2018). Micro Metal Gear Motor with Connector (75:1)-DFRobot, dirección: <https://www.dfrobot.com/product-1410.html> (visitado 19-09-2018).
- [16] R. G. Gallager, *Principles of Digital Communication*. Cambridge: Cambridge University Press, 2008, ISBN: 978-0-511-81349-8. DOI: 10.1017/CB09780511813498. dirección: <http://ebooks.cambridge.org/ref/id/CB09780511813498> (visitado 16-09-2018).
- [17] SparkFun. (2018). WiFi Module - ESP8266 - WRL-13678 - SparkFun Electronics, dirección: <https://www.sparkfun.com/products/13678> (visitado 17-09-2018).
- [18] F. Electronics. (2018). 32 bit microcontroller, what is a 32 bit microcontroller? Embedded microcontrollers - Future Electronics, dirección: <http://www.futureelectronics.com/en/microcontrollers/32-bit-microcontroller.aspx> (visitado 15-09-2018).
- [19] Microchip. (2018). PICkit™ 3 In-Circuit Debugger, dirección: <https://www.microchip.com/Developmenttools/ProductDetails/PG164130> (visitado 20-09-2018).
- [20] Techopedia. (2018). What is a servo motor? - definition from techopedia, Techopedia.com, dirección: <https://www.techopedia.com/definition/13274/servo-motor> (visitado 15-09-2018).
- [21] DFRobot. (2018). 9g 180° Micro Servo (1.6kg) -DFRobot, dirección: <https://www.dfrobot.com/product-255.html> (visitado 15-09-2018).

- [22] Traxxas. (2018). Traxxas digital nano servomotor, dirección: <https://www.robotshop.com/en/traxxas-digital-nano-servomotor.html> (visitado 15-09-2018).
- [23] H. King. (2018). BMS-306dmax digital micro servo (extra strong) 1.6kg / .13sec / 7.1g, Hobbyking, dirección: https://hobbyking.com/en_us/bms-306dmax-digital-micro-servo-extra-strong-1-6kg-13sec-7-1g.html?__store=en_us (visitado 15-09-2018).
- [24] Techopedia. (2018). What is a proximity sensor? - definition from techopedia, Techopedia.com, dirección: <https://www.techopedia.com/definition/15003/proximity-sensor> (visitado 16-09-2018).
- [25] R. Shop. (2018). HC-SR04 ultrasonic range finder, dirección: <https://www.robotshop.com/en/hc-sr04-ultrasonic-range-finder.html> (visitado 15-09-2018).
- [26] M. Inc., *MB1010 LV-MaxSonar-EZ1 | MaxBotix Inc.* 2018. dirección: https://www.maxbotix.com/Ultrasonic_Sensors/MB1010.htm (visitado 06-06-2018).
- [27] RobotShop. (2018). Telémetro ultrasónico de bajo costo Devantech SRF02, dirección: <https://www.robotshop.com/es/es/telemetro-ultrasonico-bajo-costo-devantech-srf02.html> (visitado 16-09-2018).
- [28] LeddarTech. (2018). Leddar solid-state LiDAR technology fundamentals, LeddarTech, dirección: <https://leddartech.com/technology-fundamentals/> (visitado 16-09-2018).
- [29] RobotShop. (2018). LIDAR-lite 3 laser rangefinder, dirección: <https://www.robotshop.com/en/lidar-lite-3-laser-rangefinder.html> (visitado 16-09-2018).
- [30] LeddarTech. (2018). LeddarOne single element LiDAR sensor module, LeddarTech, dirección: <https://leddartech.com/modules/leddarone/> (visitado 16-09-2018).
- [31] Pololu. (2018). Pololu - DRV8833 dual motor driver carrier, dirección: <https://www.pololu.com/product/2130> (visitado 17-09-2018).
- [32] —, (2018). Pololu - TB6612fng dual motor driver carrier, dirección: <https://www.pololu.com/product/713> (visitado 18-09-2018).
- [33] E. P. Co. (2018). What IS an encoder? > Encoder Products, dirección: <http://encoder.com/blog/company-news/what-is-an-encoder/> (visitado 18-09-2018).
- [34] DFRobot. (2018). Mini Robot chassis Encoder (SKU:SEN0116) - DFRobot Electronic Product Wiki and Tutorial: Arduino and Robot Wiki-DFRobot.com, dirección: [https://www.dfrobot.com/wiki/index.php/Mini_Robot_chassis_Encoder_\(SKU:SEN0116\)](https://www.dfrobot.com/wiki/index.php/Mini_Robot_chassis_Encoder_(SKU:SEN0116)) (visitado 18-09-2018).
- [35] *Batería de polímero de litio*, en *Wikipedia, la enciclopedia libre*, Page Version ID: 108491946, 6 de jun. de 2018. dirección: https://es.wikipedia.org/w/index.php?title=Bater%C3%ADa_de_pol%C3%ADmero_de_litio&oldid=108491946 (visitado 19-09-2018).
- [36] A. Industries. (). Lithium Ion Polymer Battery - 3.7v 2500mAh, dirección: <https://www.adafruit.com/product/328> (visitado 19-09-2018).
- [37] H. King. (2018). Quanam FPV Auricular Batería 7.4V 1500mAh 3C, Hobbyking, dirección: https://hobbyking.com/es_es/quanam-fpv-headset-battery-7-4v-1500mah-3c.html?__store=es_es (visitado 19-09-2018).
- [38] M. Electronics. (2018). L7805CV STMicroelectronics, Mouser Electronics, dirección: <https://www.mouser.com.gt/ProductDetail/511-L7805CV> (visitado 20-09-2018).

- [39] —, (2018). AZ1117T-3.3E1 Diodes Incorporated, Mouser Electronics, dirección: <https://www.mouser.com.gt/ProductDetail/621-AZ1117T-3.3E1> (visitado 20-09-2018).

15.1. Archivos generales de diseño de la nueva plataforma

<https://drive.google.com/drive/u/1/folders/16Ci88dPxx0iQHxv3adG3kEW6cg01VfQq?ogsrc=32>

15.2. Código final del microcontrolador PIC32MX250F128B en lenguaje C

<https://drive.google.com/drive/u/1/folders/1Aid7IPJvp3JehFB0f2u-FS49zYvD3cBq?ogsrc=32>

15.3. Librería de funciones en Matlab para controlar plataforma

https://drive.google.com/drive/u/1/folders/1hr6-_UODZaE13G7NHFgxsHIhVB4Shd7D?ogsrc=32

15.4. Archivos gerber de PCB

<https://drive.google.com/drive/u/1/folders/1jYQcRJbW9zUsaXdyILb0qTLEKOPD0Ap4?ogsrc=32>

15.5. Diagrama electrónico final

https://drive.google.com/drive/u/1/folders/151BmPdRjjZA052CSME7oZm4_ZSbwDryF?ogsrc=32

15.6. Comandos AT ESP-8266

<https://drive.google.com/drive/u/1/folders/1LZmePnJLTQRr5gxANuLsY1bZEO4dRU0?ogsrc=32>

15.7. Diseños de estructura en Inventor

<https://drive.google.com/drive/u/1/folders/1AmifA4N0SowrQq0o0YblEXMZmsrKe96E?ogsrc=32>

