



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Diseño de un robot móvil interactivo y modular

TESIS

Que para obtener el título de
Ingeniero Mecatrónico

P R E S E N T A N

Javier Navarro García

Axel Freddy Zurita Lozano

DIRECTOR DE TESIS

M.I. Ulises Martín Peñuelas Rivas



Ciudad Universitaria, Cd. Mx., 2016

Índice

	Página
Objetivo	ix
Introducción	x
1 Estado del arte	1
1.1 Antecedentes de la Robótica	1
1.2 Molecubes	2
1.3 Linkbot	4
1.4 Topobo	5
1.4.1 Partes pasivas	5
1.4.2 Partes activas	5
1.4.3 <i>Reina</i>	6
1.4.4 Cuestiones técnicas	6
1.5 Cubelets	8
1.6 MUNAMBOT	9
2 Diseño conceptual	11
2.1 Identificación de las necesidades	11
2.2 Establecimiento de las especificaciones	13
2.3 Generación del concepto	17
2.3.1 Propuestas para la mecánica	17
A) Geometría	17
B) Locomoción	17
C) Soportes para la electrónica	18
D) Propuestas para cerrar el robot	19
E) Integración con otras tecnologías - pines de <i>LEGO</i>	20
2.3.2 Propuestas para la electrónica	21
A) Medición de distancia	21
B) Medición de velocidad de los motores	21
C) Medición de la orientación	21
D) Motores disponibles	22
E) LEDs para las caras	22
F) Fuente de energía	22
G) Comunicación inalámbrica	23
H) Procesador central	23
2.4 Selección del concepto	24
2.4.1 Selección de la mecánica	24

A)	Locomoción	24
B)	Soportes para la electrónica y cerrar el robot	24
C)	Pines de <i>LEGO</i>	25
2.4.2	Selección del procesador	25
2.4.3	Selección de la electrónica	27
A)	Medición de distancia	27
B)	Medición de velocidad de los motores	27
C)	Medición de la orientación del robot	28
D)	Selección de LEDs	29
E)	Fuente de energía	29
F)	Comunicación inalámbrica	29
2.4.4	Resumen final de la selección	31
3	Diseño a detalle	33
3.1	Mecánica	34
3.1.1	Cuerpo central	34
A)	Soporte para las PCBs	37
B)	Soportes para sensor y botones	39
C)	Soporte para las baterías	42
D)	Soporte para los LEDs	43
E)	Soporte para las tapas	44
3.1.2	Tapas	45
A)	Soportes para los tornillos	46
B)	Soporte para el motor	47
3.1.3	<i>Brackets</i>	49
3.1.4	Micro motores y llantas	52
3.2	Electrónica	54
3.2.1	Sensores	54
A)	Encoders	54
	Descripción general	54
	Adecuación de la señal	55
	Lectura de los encoders	58
	Esquema electrónico	58
B)	IMU	59
	Descripción general	59
	Giroscopios	60
	Acelerómetros	62
	Termómetro	64
	Preparación para su lectura	64
	Selección del filtro paso bajas	64
	Configuración de la interrupción externa	64
	Esquema electrónico de la IMU	65
C)	Nivel de carga de la batería	65
	Descripción general	65
	Comparadores	66
	Habilitación y deshabilitación de los comparadores	69
D)	Ultrasónico	72

	Descripción general	72
	Operación	72
	Primeras pruebas	73
	Esquema electrónico	73
3.2.2	Actuadores	74
	A) LEDs RGB	74
	Descripción general	74
	Inconvenientes de usar el microcontrolador como generador de las señales PWM	75
	<i>PCA9956A</i>	75
	Montaje	78
	B) Motores	79
3.2.3	Energía	83
	A) Batería	83
	B) Regulación de la alimentación	84
	Encendido y apagado del robot	85
	MOSFET de alimentación	85
3.2.4	Comunicación inalámbrica	87
	A) RN171	87
	Circuito	88
	Configuración	89
3.2.5	PCBs	90
	A) Placa de <i>Energía</i>	91
	B) Placa del <i>Maestro</i>	94
	C) Placa del <i>Esclavo</i>	97
3.3	Microcontroladores	100
	3.3.1 <i>dsPIC30F4011</i>	101
	3.3.2 dsPIC Maestro	102
	A) Encendido del Maestro	103
	B) <i>Firmware</i> del Maestro	106
	Interactividad mínima con el usuario usuario	106
	Funcionamiento interno del Maestro	106
	3.3.3 dsPIC Esclavo	107
	A) Encendido del Esclavo	108
	B) Configuración y uso de los encoders	110
	C) Configuración y uso de la IMU	113
	D) Configuración y uso del Ultrasónico	119
	E) Configuración y uso del LED Driver	123
	F) Configuración y uso de los motores	125
	G) Programa general del Esclavo	129
4	Pruebas y análisis	135
	4.1 Pruebas a los sensores	135
	4.1.1 Pruebas a las señales de salida de los Encoders	135
	4.1.2 Pruebas a los ejes de la IMU	138
	4.1.3 Pruebas con los comparadores	141
	4.1.4 Lectura del sensor ultrasónico	142

4.2	Pruebas a los actuadores	147
4.2.1	Encendiendo los LEDs RGB	147
4.2.2	Pruebas de velocidad de los motores	148
4.3	Calibración y prueba de los componentes finales	150
4.3.1	Calibración de colores de los LEDs	150
4.3.2	Modelo físico del robot	153
4.3.3	Ensamble de la electrónica dentro del cuerpo central	155
4.3.4	Pruebas al módulo de procesamiento y comunicación	157
4.3.5	Pruebas a los módulos de sensado y actuación	160
	A) Encoders y motores	160
	B) IMU	163
	C) Sensor ultrasónico	166
	D) LEDs RGB	168
	E) Pruebas a otros componentes	169
	Reguladores de voltaje y MOSFET's	169
	Comparadores	170
	Comunicación entre microcontroladores	170
5	Conclusiones	171
6	Trabajos a futuro	175
	Bibliografía	177
A	Planos mecánicos	A1
A.1	Planos del cuerpo central	A2
A.2	Planos de la tapa	A2
A.3	Planos del <i>bracket</i>	A2
B	Esquemas electrónicos	B1
B.1	Esquema de la placa del Maestro	B2
B.2	Esquema de la placa del Esclavo	B2
B.3	Esquema de la placa de Energía	B2
C	Periféricos del dsPIC	C1
C.1	I/O	C1
C.2	I ² C	C2
C.3	PWM	C6
C.4	QEI	C8
C.5	Timers	C10
C.6	UART	C12

Agradecimientos

Javier Navarro

Dedico esta tesis a la Dra. Concepción Abellán Giran quien siempre creyó en mí y me brindó la motivación necesaria para llegar hasta este momento.

Sin su apoyo jamás lo habría logrado.

Gracias.

Axel Zurita

El trabajo que a continuación se lee es el producto del esfuerzo de varios años, durante el cual tuve el apoyo de muchas personas.

Primero que nada agradezco el enorme apoyo que me ha brindado mi familia durante toda la carrera. Todo el apoyo emocional alentando que siguiera adelante, asesoría con algunos temas escolares, facilidad de transporte y la capacidad de brindarme todo material que necesité durante la carrera ha sido uno de los pilares fundamentales para que yo pudiera avanzar centrándome en mis estudios y poder así triunfar durante mi vida universitaria. Así mismo agradezco el enorme apoyo que me han brindado los últimos años y me seguirán brindando ahora que continuaré mis estudios de Maestría en los EEUU.

Agradezco a la UNAM y a la Facultad de Ingeniería por haberme brindado la oportunidad de estudiar mi carrera en este grandioso espacio y desarrollar mis habilidades y adquirir los conocimientos necesarios para poder desenvolverme en un futuro, primeramente para continuar mis estudios y, posteriormente, en la vida profesional.

Un abrazo para Antonio Fonseca Alvarado, un hermano para mí, quien me ha acompañado en las buenas y en las malas, en los retos y en el éxito, en clase y en proyectos, a quien he conocido por más de una década y tengo el gusto de llamar *Manix*. Con él he colaborado y competido, siempre con el enfoque de impulsarnos el uno al otro.

Nunca uno avanza solo durante la carrera, siempre hay personas que lo impulsan a uno para alcanzar el máximo potencial. Aprecio mucho el gran impulso que me han dado el M.I. Ulises Martín Peñuelas Rivas, el Dr. Jesús Manuel Dorador González y el M.I. Billy Arturo Flores Medero Navarro, quienes gracias a su apoyo logré forjarme como el Ingeniero que soy actualmente y me dieron el impulso para ser aceptado a la Maestría en los EEUU.

Un beso para Cinthya Bárbara Carrillo Sánchez, quien estuvo a mi lado desde principios de la carrera y con quien aprendí lo que es la vida en pareja. Los bellos momentos, las peleas amargas, el descubrir los secretos de cada uno, me llevo todos estos recuerdos ahora que inicio una nueva etapa lejos de aquí pero que nunca olvidaré.

Finalmente, como mención especial, un saludo a todos aquellos amigos de la generación con quien conviví durante clase y trabajé en proyectos; Andrés, Martín, Pepe, Ale, fue un gusto cursar la carrera con todos ustedes. Un saludo para todo aquel que olvidé mencionar pero que fue parte de este camino.

Un mensaje para todos los futuros ingenieros

La intención de esta tesis es demostrar que las cosas se pueden hacer diferente, con una gran calidad y buscando la innovación a pesar de lo difícil que pueda resultar el aprender cosas nuevas.

El trabajo realizado es muy extenso, hay mucho que profundizar y mucho por aprender. El desarrollo de *CU-10-80-T* tomó poco más de un año, desde ser una idea en papel hasta un prototipo.

Con este proyecto buscamos integrar nuevas cosas, emplear componentes distintos, aprovechar tecnologías nuevas para obtener un prototipo capaz de convertirse en un producto final. Durante el desarrollo del robot aprendimos a manejar con maestría varios programas para las diferentes etapas del diseño.

A través de la experiencia, determinamos que los microcontroladores con los que aprendemos en el aula eran insuficientes y nos dimos a la tarea de aprender a usar uno más poderoso. Pudimos aprender a usarlo gracias a los fundamentos, las bases y el apoyo que nos brindaron nuestros profesores.

Nuestras necesidades para probar los circuitos nos llevaron a diseñar inclusive 2 tarjetas de desarrollo para experimentar todo sensor y actuador del robot. El diseño de estas tarjetas no fue una tarea que se pudo hacer de la noche a la mañana, requirió indagar y conocer a detalle el nuevo microcontrolador.

La creación de las PCBs y el soldado de los componentes de montaje superficial fue un proceso que tuvimos que desarrollar desde cero, por falta de recursos económicos, y nos tomó 3 meses lograr el objetivo.

Así como estos retos, hay muchos más que se encuentran a lo largo de toda la tesis y queremos recalcar que cada uno de ellos nos hizo crecer como futuros profesionistas.

Con todo lo anterior, vemos con mucha tristeza cómo las nuevas generaciones se están acostumbrando a tomar siempre el camino fácil, el camino de la flojera, el camino del mínimo esfuerzo. Sabemos que existen herramientas de desarrollo que hacen muy sencillo el obtener modelos funcionales en poco tiempo. No nos oponemos al uso de tales herramientas, como el *Arduino* por ejemplo; a lo que nos oponemos es que uses estas herramientas sin cuestionar su funcionamiento, a que no intentes invertir tiempo en conocer y dominar tecnologías más avanzadas, a que intentes desarrollar todos tus proyectos con las mismas soluciones.

Nos tocó ver compañeros de carrera batallar con una solución por una librería que no funcio-

naba. Su única solución era buscar una nueva librería que realizara el trabajo. En la Facultad se te enseñan las herramientas existentes, corre por tu cuenta que aprendas cómo funciona cada una y domines las que consideras útiles para ti.

Conoce a fondo tus herramientas, no te pierdas en el camino fácil de usar interfaces bonitas y desentenderte del funcionamiento de las cosas con las que vas a diseñar. Emplear una tarjeta de desarrollo en un producto que vas a vender no es digno de alguien que estudia en esta Facultad, no demuestra la capacidad que tienes de diseñar los elementos personalizados para tu producto. Intenta innovar, busca nuevos componentes y herramientas e intenta superarte día a día en vez de estancarte.

No tomes esto como regaño sino una oportunidad para seguir el camino correcto, aún estás a tiempo, apenas vas a empezar tu tesis o la vida laboral, da lo mejor de ti.

Sabemos que no es fácil, pero todo lo que vale la pena lleva un costo y conlleva un esfuerzo.

Ayudémonos a que se diga con orgullo que eres y somos hijos de la FACULTAD DE INGENIERÍA DE LA UNAM.

Objetivo

Desarrollar un robot móvil reprogramable, interactivo y modular con la finalidad de mostrar la integración de los subsistemas que conforman un sistema mecatrónico. Asentando el uso de tecnologías actuales para motivar su difusión en proyectos futuros.

Introducción

CU-10-80-T es un sistema mecatrónico enfocado a desarrollar tareas variadas mediante la conjunción de diferentes módulos de procesamiento, comunicación, sensado y actuación. Un sistema mecatrónico es aquel que combina de manera sinérgica elementos tanto mecánicos como electrónicos como de programación, es decir, un sistema en donde las partes que lo conforman están fuertemente ligadas en diseño y funcionamiento, no son entes separados [20]. Puede desarrollar dichas tareas, basándose en un firmware base programado en su módulo de procesamiento, a través de un set de comandos reducidos. Esto permite que el usuario no necesite conocimientos en la programación de microcontroladores para crear una gran variedad de acciones.

Cuenta con la capacidad de reaccionar a diferentes tipos de entradas como: distancia a objetos, botones, su orientación e instrucciones transmitidas inalámbricamente. Puede actuar mediante luces multicolores, movimiento y transmitir información inalámbricamente.

El robot se compone de un módulo de procesamiento, uno de sensado, uno de actuación y uno de comunicación, todo alimentado mediante baterías. El módulo de procesamiento es el encargado de administrar el funcionamiento de los demás módulos presentes en el robot; permite, a través del módulo de comunicación, compartir la información de todos sus módulos con otros robots.

Hace uso de una comunicación inalámbrica tipo wifi, emplea microcontroladores de 16 bits que combinan una máquina DSP y sus circuitos están desarrollados con componentes de montaje superficial.

Durante el desarrollo de esta tesis se abarca el proceso de diseño del robot, desde su diseño conceptual hasta su diseño a detalle, y las pruebas y análisis ejecutadas a cada uno de los componentes del robot para comprobar el diseño realizado. Al final se presentan las conclusiones y los trabajos a futuro que se plantean para el robot.

Capítulo 1

Estado del arte

1.1 Antecedentes de la Robótica

A lo largo de la historia, el hombre se ha sentido fascinado por las máquinas y dispositivos capaces de imitar las funciones y los movimientos de los seres vivos. Los griegos tenían una palabra específica para denominar a estas máquinas: *αυτόματος*. De esta palabra se deriva la actual autómata: máquina que imita la figura y movimientos de un ser animado [16].

Durante los siglos XVII y XVIII se crearon ingenios mecánicos que tenían alguna de las características de los robots actuales. Estos dispositivos fueron creados en su mayoría por artesanos del gremio de la relojería. Su misión principal era la de entretener a los cortesanos y servir de atracción en las ferias. Estos autómatas representaban figuras humanas, animales o pueblos enteros.

En octubre de 1945 se publicó en la revista *Galaxy Science Fiction* una historia en la que por primera vez enunciaron las tres leyes de la robótica:

- 1.- Un robot no puede perjudicar a un ser humano ni con su inacción permitir que un ser humano sufra daño.
- 2.- Un robot ha de obedecer las órdenes recibidas de un ser humano, excepto si tales órdenes entran en conflicto con la primera ley.
- 3.- Un robot debe proteger su propia existencia mientras tal protección no entre en conflicto con la primera o la segunda ley.

Así como existen distintos tipos y clasificaciones para máquinas y distintos entes, la robótica y los robots pueden clasificarse de distintas maneras. Algunas de las clasificaciones pueden ser como a continuación se menciona:

- Robótica industrial, didáctica y de exploración.
- Robots inteligentes y autónomos.
- Robots manipuladores, móviles e híbridos.

1.2 Molecubes

“El aprendizaje mediante la experimentación y la programación”.

Uno de los dispositivos que está causando una gran revolución en el campo de la robótica modular en los últimos años son los llamados “Molecubes” [7]. Una de sus características más importantes es la variedad de módulos que maneja, la cual permite que rápidamente se construyan diferentes configuraciones de robots con el simple hecho de ensamblar diferentes módulos.

El diseño del “Molecube” fue realizado por la empresa *Creative Machines Lab*. Su primer diseño fue realizado en el 2004. A través de los años se han realizado varias iteraciones de su diseño, con lo cual han logrado miniaturizarlo cada vez más, simplificarlo y se conseguido un diseño más robusto. La geometría que lo caracteriza es la forma de un cubo (Figura 1.1-a) con esquinas redondeadas.

Para formar la estructura cúbica es necesario ensamblar dos partes con forma de pirámide triangular, de tal manera que se conectan los ejes de ambos a las bases. Cada una de las seis caras del módulo está equipada con un conector electromecánico para poder acoplar otros módulos (Figura 1.1-b). Cada módulo cuenta con 6 piezas fundamentales (Figura 1.2) fabricadas en polímero ABS.

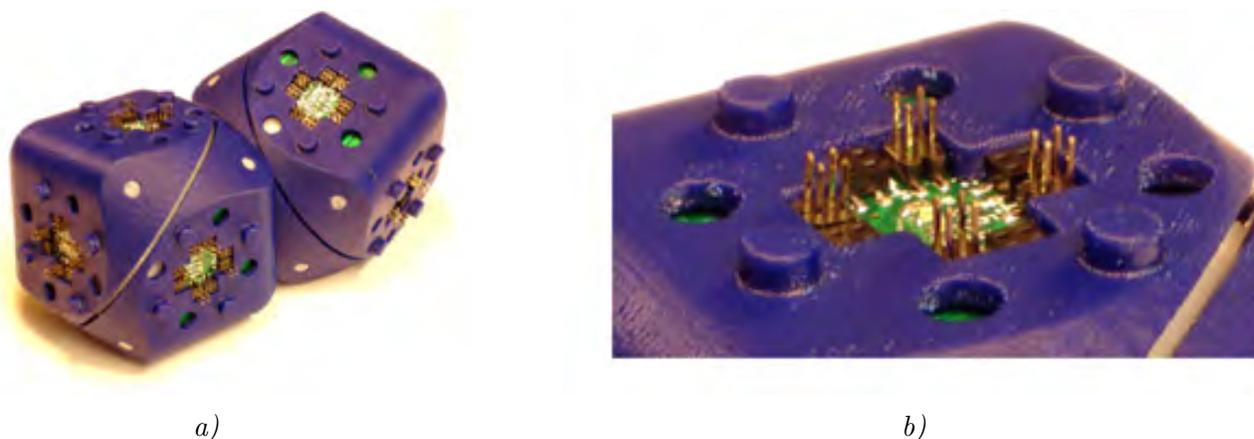


Figura 1.1: a) “Molecubes”; b) soporte electromecánico de los “Molecubes”.

Cada uno de los “Molecubes” está equipado con un juego de circuitos electrónicos que le permiten realizar cada una de sus funciones ya previamente programadas, tales como girar un actuador a un ángulo deseado, ajuste de velocidad y par de su movimiento; además de ello, tiene la capacidad de tener el control del módulo, tanto de sus salidas como de la lectura de las entradas digitales, con el fin de leer o escribir valores del registro interno del motor, entre otras tareas.

En su diseño se toman en cuenta dos aspectos base: el software de bajo nivel se programó de la manera más eficiente y se simplificó el diseño de todo el sistema; para lograrlo se diseñó un conjunto acoplable de placas que se utilizan en los diferentes tipos de módulos. Las diferentes combinaciones de estas placas tienen la capacidad de alterar las funciones dependiendo del módulo que se haya conectado.

De igual forma, se optimizó el volumen ocupado por las tarjetas dentro de los módulos. Esto se logra a partir de la combinación de la interfaz de la comunicación/potencia y funciones de control, de tal forma que, cada módulo esté equipado con la interfaz en las placas y que pueda ser reprogramable.

Su unidad de procesamiento principal es un microcontrolador ATMEGA324P, el cual trabaja

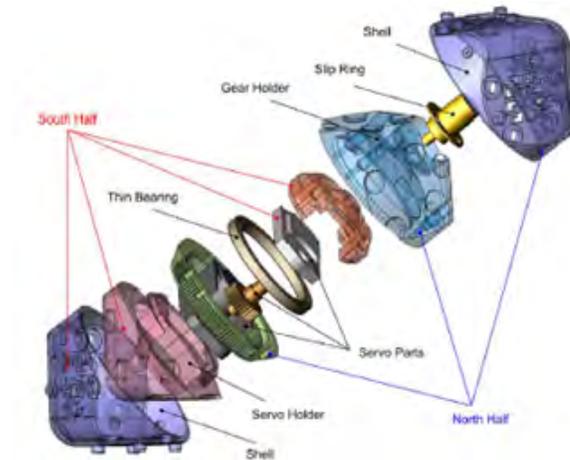


Figura 1.2: *Piezas que conforman al “Molecube”.*

a una frecuencia de 16 [MHz]. Este microcontrolador proporciona el control y tiene la capacidad de establecer la comunicación. Para la transmisión de comandos de un módulo a otro diseñaron un protocolo de comunicación especializado en el trabajo de sus módulos, el cual se trata de un protocolo en serie semidúplex con señales de recepción/transmisión, el cual es modulado en un solo bus de comunicación. Este protocolo de comunicación está diseñado para ser compatible con el protocolo de comunicación empleado por los Robotics AX-12, de tal manera que el controlador de los módulos pueda comunicarse directamente y controlar los servomotores.

El software del “Molecube” está diseñado para probar y evaluar las diferentes morfologías y comportamientos en los módulos, la interfaz permite que el usuario pueda añadir cualquier módulo en la configuración de la interfaz virtual y así poder emular su comportamiento con el modelo físico. De esta manera el software permite al usuario pueda simular y evaluar lo que se puede ocurrir con las diferentes configuraciones con los módulos (Figura 1.3).

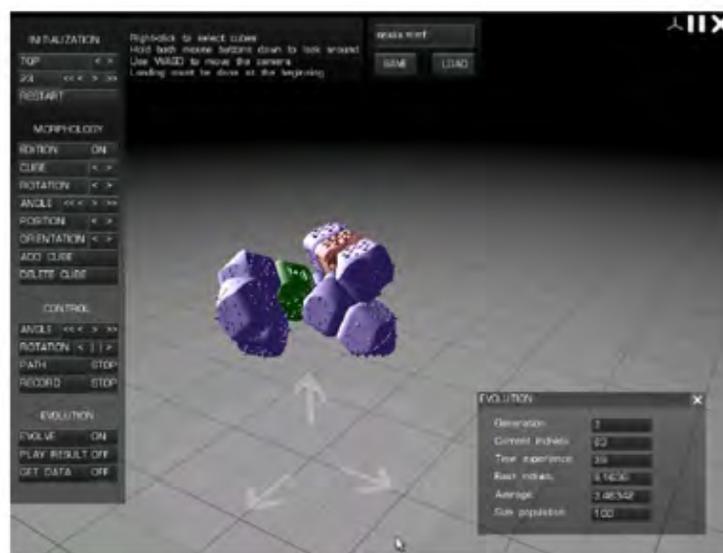


Figura 1.3: *Interfaz propia de los “Molecubes”.*

1.3 Linkbot

“I think the thing that sets Linkbot apart from most robots is our absolute encoding and torque to weight ratio. We want you to do more than just roll, we want you to walk!” Graham Ryland.

Linkbot [69] es un robot modular que cuenta con un LED RGB, un altavoz, una batería de ion litio de 3 horas de duración alimentada mediante un conector micro USB, motores de 7.2 [kg/cm] con 360° de rotación y un encóder absoluto, un acelerómetro de 3 ejes, botones para su programación directa y un conector RJ11 para ampliar la gama de sensores y actuadores que puedes utilizar (Figura 1.4).

Es capaz de utilizar Zig Bee para su comunicación. Cuenta con tornillos estándar para ampliar su compatibilidad y poder unir todo lo que se desee en la creación de nuevos modelos.

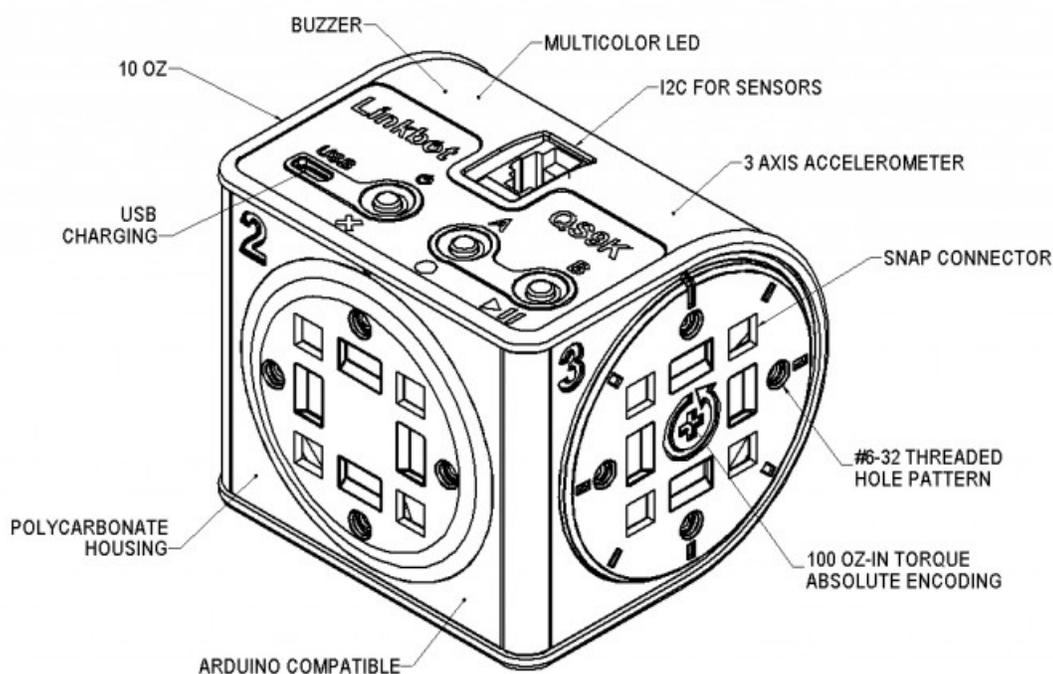


Figura 1.4: Componentes básicos del “Linkbot”.

Utiliza un microcontrolador Atmel AVR compatible con Arduino, por lo que unido a sus capacidades de conexión, puede ser modificado como se desee. A la fecha soporta como lenguajes de programación a C y Python.

Tanto el software como el hardware son “open source” y las piezas están a disposición de la comunidad para poder fabricarlas y modificarlas cuando se requiera.

Ofrece una gama de funciones disponibles directamente salido de su caja:

- BumpConnect, para conectar inalámbricamente los módulos entre sí con solo tocarlos una vez.
- TiltDrive hace uso del acelerómetro de uno de los módulos para controlar a los demás.
- Copycat, que permite a las articulaciones de uno controlar las articulaciones de otros e incluso existe una pequeña retroalimentación entre los movimientos.

- Finalmente PoseTeach es una especie de animación cuadro por cuadro que permite grabar movimientos con los Linkbots y reproducirlos sin la necesidad de una computadora o bien conectarlo a ella para poder modificar directamente el código y crear movimientos complejos.

1.4 Topobo

Topobo [68] es un robot modular con memoria cinemática desarrollado principalmente por Hayes Solos Raffle y Amanda Parkes en el *Massachusetts Institute of Technology* (MIT, por sus siglas en inglés).

Topobo es un robot didáctico pensado para el desarrollo de las habilidades motrices y creatividad de niños y adolescentes. Por ello se diseñó como un robot con una interfaz muy intuitiva, con la menor cantidad de elementos posibles, simétrico y de pocas acciones.

Topobo se compone de dos grandes elementos: las partes activas y las partes pasivas.

1.4.1 Partes pasivas

Los elementos pasivos son partes plásticas rígidas con formas simétricas que permiten crear diferentes figuras (generalmente simétricas) dependiendo del arreglo que le haya dado el niño. Topobo está pensado para modelar formas naturales como esqueletos o mallas que se entrelacen. Su geometría está basada en cristales cúbicos y tetraédricos.

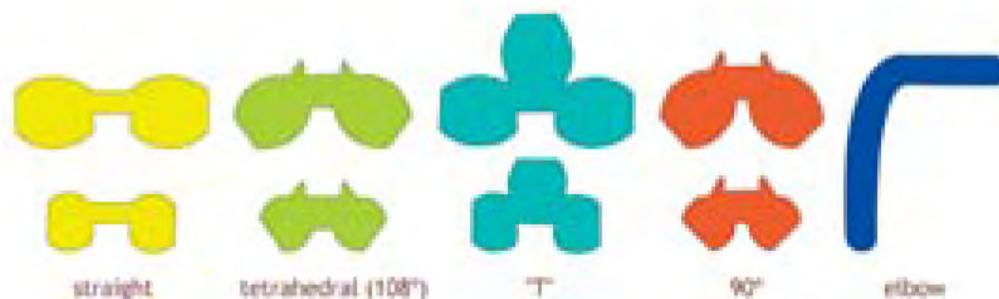


Figura 1.5: *Diferentes piezas pasivas de "Topobo".*

Los elementos pasivos presentan escalas 2:3 basadas en el radio de Fibonacci que describe las escalas de sistemas en crecimiento como los esqueletos de mamíferos. A su vez, los colores están codificados para distinguir las diferentes formas y unificar la paleta de colores dependiendo la forma, de tal manera que las estructuras tendrían patrones definidos y no una mezcla aleatoria de colores.

1.4.2 Partes activas

Las partes activas son los objetos con forma de huevo que desarrollan todo el movimiento. Se componen de un servomotor, una conexión para una red de activos y un botón retroiluminado por un LED que indica si el sistema está en modo de grabado (rojo) o de ejecución (verde).

Cuando el usuario quiere grabar movimientos, debe presionar el botón para pasar al modo de grabado, realizar los movimientos y volver a presionar el botón. La parte activa pasará automáticamente a modo de ejecución, en la que realizará los diferentes movimientos grabados. Si el usuario quiere detener su ejecución, basta con presionar una tercera vez el botón. Presionarlo dos veces de

manera rápida, similar a hacer un doble click con el mouse de una computadora, hará que la parte activa ejecute la última acción grabada únicamente.

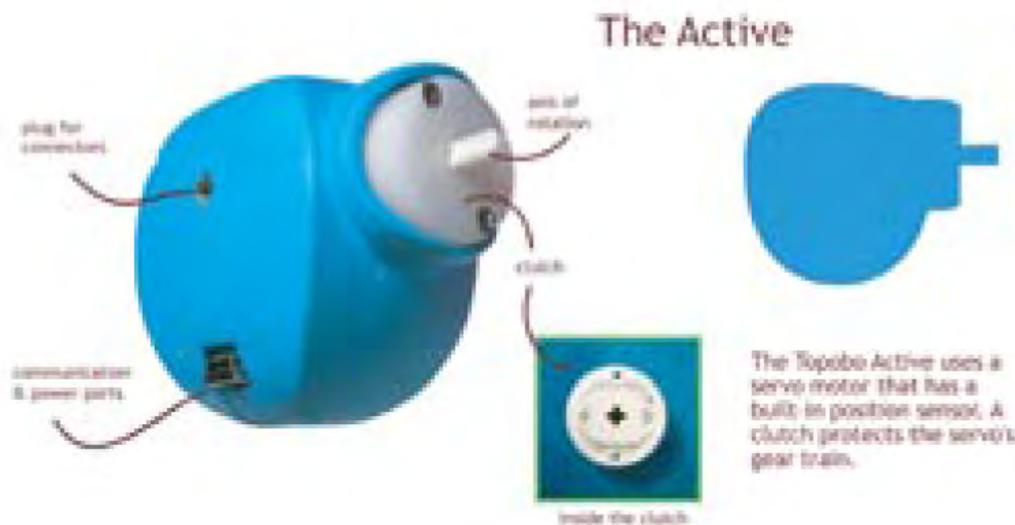


Figura 1.6: Parte activa de "Topobo".

Cuando existe una red de Activos, todos grabarían o ejecutarían al mismo tiempo. La secuencia es la misma, pero aunque el niño mueva la estructura de manera global, cada Activo grabará su movimiento de manera local, de tal forma que cuando alguno de ellos pase a modo ejecución, todos lo harán al mismo tiempo y desarrollarán el movimiento de manera local. Este movimiento, visto de manera global, sería el que el niño grabó.

Los Activos están compuestos por un servomotor y componentes electrónicos en un empaque plástico. El empaque tiene 6 puntos de conexión mecánica, 3 puertos para conexiones de energía/comunicación y un botón con retroiluminación por un LED. Una de las conexiones mecánicas está conectada al eje del servomotor, que tiene una capacidad de rotación de 170° . Los circuitos electrónicos manejan la distribución de energía, la memoria y procesamiento, y la comunicación punto a punto para las comunicaciones seriales. Todo Activo es autónomo e idéntico, por lo que sólo se requiere que tenga una fuente de alimentación para poder entrar en funcionamiento.

1.4.3 *Reina*

La Reina es un elemento que tiene la capacidad de control y sincronización sobre todos los Activos que se encuentren en la red. De igual forma que con una red convencional con Activos, si la Reina entra en modo de grabación, todos los activos entrarán en modo de grabación. La diferencia de la Reina con cualquier Activo radica en que, en una red, el movimiento que se graba sobre la Reina quedará grabado de manera idéntica sobre todos los Activos en la red.

Al tener control sobre la grabación de todos los Activos, la Reina puede funcionar como control remoto y no necesariamente estar conectada a la estructura final con los activos, lo cual permite versatilidad si se desea que el movimiento sea el mismo para varios Activos a la vez.

1.4.4 Cuestiones técnicas

Las partes pasivas están compuestas en su totalidad de un polímero ABS, fabricados mediante inyección en dos partes que se sueldan posteriormente, dando como resultado una pieza hueca de



Figura 1.7: *Reina controlando una red de Activos.*

alta resistencia. El encapsulado de las partes activas se compone de resina de uretano.

Los servomotores son de la marca HiTec, con reducción 60:1. Su caja de reducción está compuesta por engranes metálicos. Aunado al gran torque y velocidad angular que puede desarrollar el motor mismo, este servomotor puede resistir movimientos bruscos sin que se rompan los dientes de los engranes, que por lo general son frágiles. Adicionalmente, los servomotores están protegidos por un *clutch* que impide que el usuario dé movimientos sumamente bruscos o rápidos que dañen los dientes de los engranes.

Las líneas de alimentación de energía manejan 18 [V], que son posteriormente escalados a 6 [V] y finalmente regulados a 5 [V] para alimentar los circuitos electrónicos.

El control de los componentes activos lo realiza un PIC cuyo oscilador principal corre a 40 [MHz]. Éste se encarga de medir el potenciómetro del servomotor a una frecuencia de 36 [Hz] mediante su convertidor ADC de 10 bits y guardarlo en memoria, a su vez, en modo ejecución, controla la señal que se manda al servomotor para alterar la posición de la flecha. De igual manera se encarga de la comunicación punto a punto a una velocidad de 57000 baudios y el modo en el que se encuentra el activo al momento.

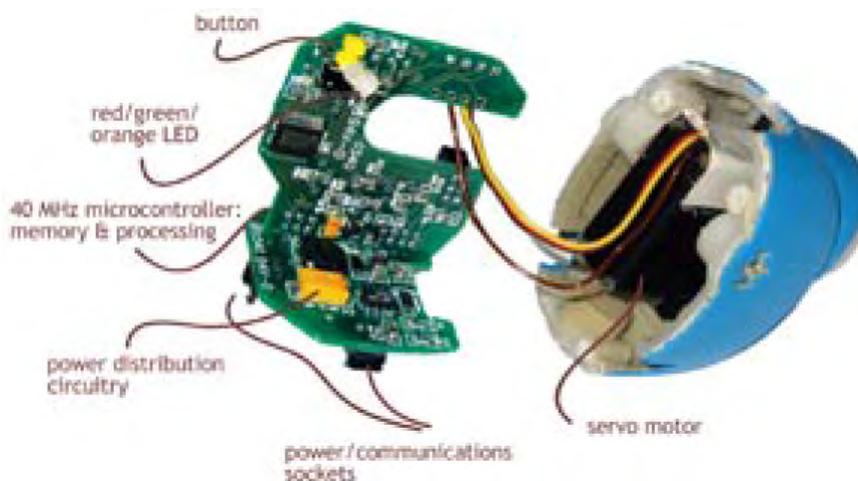


Figura 1.8: *Esquema de los circuitos y el servomotor de un Activo.*

1.5 Cubelets

Los *Cubelets* [40] son robots magnéticos que se acoplan mutuamente para conseguir una diversa variedad de juguetes robóticos. No necesitan alambirse ni programarse, y están listos para usarse por niños y sus padres para construir una amplia gama de robots diminutos. Tienen forma cúbica, lo cual les permite ensamblarse de diferentes maneras.

Los *Cubelets* son desarrollados por la compañía Modular Robotics Incorporated.



Figura 1.9: “Cubelets”.

Los robots se dividen en 3 categorías principales: actuador, sensor y batería. Actualmente existen 16 tipos de robots. Los actuadores pueden desplazar al robot en movimientos lineales o rotacionales, emitir sonido o encender luces (Figura 1.10-a). Los sensores pueden detectar distancia y obstáculos, temperatura o comunicación Bluetooth (Figura 1.10-b). El robot batería actual tiene la capacidad de recargarse mediante un puerto USB y dura un 30% más que la versión previa (Figura 1.10-c).



a)



b)



c)

Figura 1.10: a) Ejemplo de actuador de los “Cubelets”; b) sensor de temperatura; c) batería de los “Cubelets”.

1.6 MUNAMBOT

MUNAMBOT es el robot precursor de CU-10-80-T, desarrollado durante la materia “Diseño Mecatrónico” en el semestre 2014-1.

MUNAMBOT es un robot con forma cúbica orientado a ser un juguete que pudiera trabajar de manera individual o de manera colectiva junto a otros robots de su mismo tipo. Se puede desplazar como un robot diferencial mediante dos ruedas desmontables.

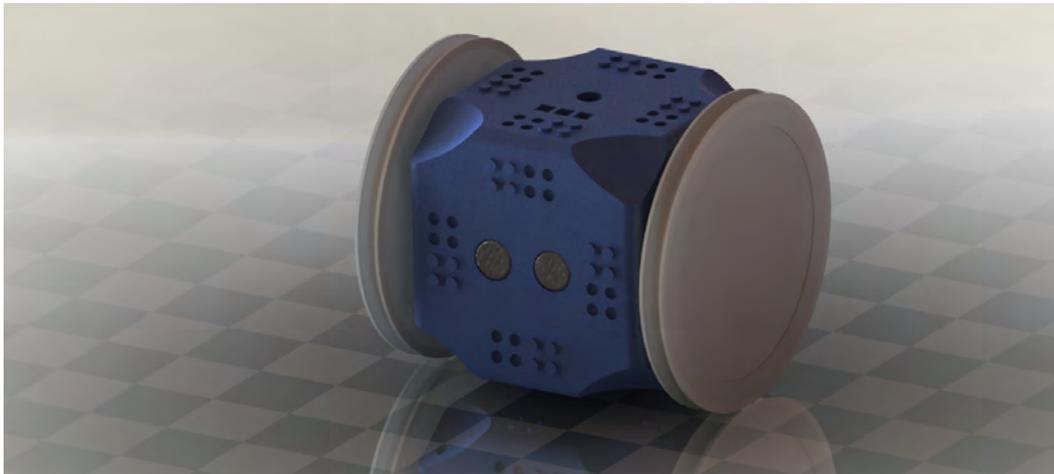


Figura 1.11: CAD final de “MUNAMBOT”.

MUNAMBOT está programado con 4 funciones:

- Modo “mascota”: la interacción con él haría que mostrara diferentes estados de ánimo, reflejados en los LEDs RGB que llevaría en las caras (rojo: enojado, azul: triste, verde: emocionado).
- Modo “lámpara de noche”: apoyar al robot en una de sus llantas y el robot encendería todos los LEDs de diferentes colores como lámpara.
- Modo “bop-it”: similar al juguete, una cara se encendería y el jugador tendría que poner esa cara hacia arriba, errar la cara o tardar demasiado lo haría perder; el juego se iría acelerando.
- Modo “beat box”: al reproducir una canción, seguiría con las caras los tonos bajos, medios y altos.

El robot nunca fue funcional y sólo se alcanzó un modelo físico y se fabricaron algunas placas electrónicas, de las cuales algunas no funcionaron por cortos o errores de diseño. Se hicieron algunos intentos de usar los dos microcontroladores *PIC16F887* para todas las tareas planeadas pero por falta de poder de procesamiento se lograban unas u otras.

Capítulo 2

Diseño conceptual

El capítulo, siguiendo como base la metodología de diseño propuesta por Karl T. Ulrich y Steven D. Eppinger [82], abarca la identificación de las necesidades del proyecto, la definición de las especificaciones, la generación y selección del concepto.

2.1 Identificación de las necesidades

Las necesidades son todos los atributos que sustenten el objetivo de la tesis; son independientes de cualquier proyecto que se pudiera desarrollar. La identificación de dichas necesidades tiene la finalidad de hallar los requisitos que satisfagan este objetivo. Nótese que se utilizan las palabras necesidades, atributos y requisitos como sinónimos dentro de este trabajo.

Algunas de las necesidades surgen de los términos literales del objetivo y otras necesidades surgen de las consideraciones hechas por nosotros dada su importancia en la complementación del objetivo.

Las necesidades identificadas son jerarquizadas en dos niveles. El nivel primario engloba las necesidades más generales y un nivel secundario brinda un mayor detalle a las necesidades primarias.

La lista es la siguiente:

1.- Reprogramable: el usuario es capaz de cambiar las tareas del robot.

- Programación a través de su propio entorno
- Facilidad en la reprogramación

2.- Interactivo: el robot es capaz de ofrecer diversas respuestas ante un estímulo¹.

- Detectar sonido
- Generar sonido
- Luces
- Movimiento
- Comunicación con otros módulos y con usuarios

¹Entiéndase estímulo como el poner en funcionamiento una tarea del robot

- Detección de obstáculos
 - Detección de orientación
- 3.- **Modular: conformado por diversos sistemas que interactúan entre sí.**
- Separación de las tareas a realizar en subsistemas
 - Intercambio de información entre los diferentes módulos
- 4.- **Móvil: capaz de desplazarse por el piso.**
- 5.- **Inalámbrico: Sin uso de alambres o cables que conecten al robot con dispositivos externos.**
- Uso de comunicación inalámbrica
 - Uso con baterías
 - Baterías recargables
- 6.- **Pequeño: manipulación sencilla con ambas manos (transporte, colocación, almacenamiento).**
- 7.- **Resistente: que dure bajo el uso normal del usuario.**
- Geometrías adecuadas
 - Materiales adecuados
- 8.- **Flexible: que se adapte fácilmente a los cambios o las circunstancias.**

Para complementar la identificación de las necesidades, la Tabla 2.1 muestra la justificación de las necesidades primarias.

Tabla 2.1: Justificación de las necesidades primarias.

Núm. de necesidad	Necesidad	Justificación
1	Reprogramable	Dada por el objetivo
2	Interactivo	Dada por el objetivo
3	Modular	Dada por el objetivo
4	Móvil	Dada por el objetivo
5	Inalámbrico	Dada por el objetivo
6	Pequeño	Facilitar pruebas
7	Resistente	Soportar pruebas
8	Flexible	Probar la capacidad de reacción

2.2 Establecimiento de las especificaciones

Las especificaciones representan una base sobre lo que se deberá hacer para satisfacer las necesidades del robot. Explican con detalle preciso, mediante palabras técnicas y medibles lo que el robot tiene que lograr.

Cada una de las necesidades identificadas en la sección anterior se traducen a una o más especificaciones objetivo que, consecuentemente, tendrán una métrica y un valor.

Las especificaciones objetivo son las metas que se piensa satisfarán con total éxito el objetivo del robot. Posteriormente, son refinadas con base en las limitaciones presentes; tales como presupuesto, herramientas y tiempo.

La lista de especificaciones objetivo se organiza en la Tabla 2.2 que permite mostrar sus métricas, sus valores, su relación con las necesidades primarias (Tabla 2.1) y su importancia relativa (siendo 1 mayor importancia y 5 menor).

Tabla 2.2: Especificaciones objetivo.

Núm. métrica	Núm. necesidad	Métrica	Importancia	Unidades	Valor ideal
1	1, 2, 8	Cantidad de veces que se puede sustituir el programa	1	Cant. veces	1 millón
2	1, 2, 3, 8	Cantidad de instrucciones máximas	5	Cant. instrucciones	100
3	1, 2, 8	Comandos mínimos disponibles para la programación de tareas	4	Cant. instrucciones	10
4	1, 2, 8	Set de comandos ASCII simple	5	Cant. instrucciones	20
5	1, 2, 8	Tipos de estímulos	1	Tipos	Botones Sonido Comandos Obstáculos Orientación
6	1, 2, 8	Tipos de respuestas	1	Tipos	Luces Sonido Movimiento Comunicación
7	2, 3	Rango de frecuencias sonoras detectables	5	Hz	20 a 20k
8	2, 3	Potencia de la señal sonora recibida	5	dB	0
9	2, 3	Rango de frecuencias sonoras reproducibles	5	Hz	20 a 20k

2.2. Establecimiento de las especificaciones

Núm. métrica	Núm. necesidad	Métrica	Impor- tancia	Unidades	Valor ideal
10	2, 3	Potencia de la señal sonora emiti- da	5	dB	0
11	1, 2, 3, 4	Velocidad de desplazamiento	2	cm/s	10
12	1, 2, 3, 4	Orientación de desplazamiento	2	rad	0 a 2π
13	2, 3, 5	Velocidad de comunicación	2	kbaud	115.2 a 1850
14	2, 3, 5	Tipos de comunicación	3	Tipos	Serie Paralelo Secuencial
15	1, 2, 3, 5, 8	Rango de dispositivos comu- nicándose	2	Cantidad de dispositivos	2 a 8
16	1, 2, 3, 4, 8	Rango de detección de obstáculos	2	cm	2 a 50
17	1, 2, 8	Número de botones en el robot	1	Número de botones	3
18	1, 2, 8	Funciones de respuesta por botón	1	Número de funciones	1 a 5
19	1, 2, 4, 8	Grados de libertad medibles por los sensores	2	Número de GDL	3 a 9
20	1, 2, 3, 4, 5, 8	Rango de módulos interactuando	1	Número de módulos	4 a 32
21	2, 3, 5	Rango de tareas asignada por módulo	1	Número de tareas	1 a 10
22	2, 3, 5	Velocidad de comunicación entre módulos	2	kbaud	115.2 a 1850
23	1, 3, 5	Tipo de información comunicada	2	Tipo de variable	Entera Flotante Texto Comando
24	4	Rugosidad de la superficie	2	μm	1.6 a 12.5
25	4	Inclinación de la superficie	2	$^\circ$	10
26	4	Número de configuraciones para locomoción de robot móvil	4	Configuración	Diferencial Omnidirec- cional Síncrona Triciclo Ackerman

Núm. métrica	Núm. necesidad	Métrica	Importancia	Unidades	Valor ideal
27	4, 6, 8	Rango de dimensiones para el dispositivo	1	cm	4 a 10
28	4, 6, 8	Prueba de esfuerzo	2	Pa	500
29	3, 4, 6, 7	Tipos de geometrías disponibles para el robot	2	Tipos	Esférica Cilíndrica Piramidal Cúbica Poliédrica
30	4, 6, 7	Materiales aptos para el robot	2	Material	ABS Aluminio
31	2, 3, 5	Velocidad de respuesta a los estímulos	3	ms	250
32	2, 3, 5	Frecuencia máxima de cambio del estímulo	3	Hz	20
33	5, 8	Alcance máximo de la comunicación	2	m	50
34	5	Tiempo mínimo de autonomía del robot	2	hr	2
35	5	Voltaje de recarga de las baterías	3	V	5
36	5	Tiempo mínimo de recarga de las baterías	2	hr	2
37	5	Capacidad de carga	3	mAh	2000

En la matriz de necesidades – métricas, Tabla 2.3, se observa de manera gráfica la relación entre las métricas y las necesidades que atiende cada una de ellas.

Tabla 2.3: Matriz de necesidades – métricas.

Métricas	Necesidades	1	2	3	4	5	6	7	8
		Reprogramable	Interactivo	Modular	Móvil	Pequeño	Resistente	Flexible	Inalámbrico
1	Cantidad de veces que se puede sustituir el programa	●	●					●	
2	Cantidad de instrucciones máximas	●	●	●				●	
3	Comandos mínimos disponibles para la programación de tareas	●	●					●	
4	Set de comandos ASCII simple	●	●					●	
5	Cantidad y tipo de estímulos	●	●					●	
6	Cantidad y tipo de respuestas	●	●					●	
7	Rango de frecuencias sonoras detectables		●	●					
8	Potencia de la señal sonora recibida		●	●					
9	Rango de frecuencias sonoras reproducibles		●	●					
10	Potencia de la señal sonora emitida		●	●					
11	Velocidad de desplazamiento	●	●	●	●				
12	Orientación de desplazamiento	●	●	●	●				
13	Velocidad de comunicación		●	●					●
14	Tipos de comunicación		●	●					●
15	Rango de dispositivos comunicándose	●	●	●				●	●
16	Rango de detección de obstáculos	●	●	●	●			●	
17	Número de botones en el robot	●	●					●	
18	Funciones de respuesta por botón	●	●					●	
19	Grados de libertad medibles por los sensores	●	●		●			●	
20	Rango de módulos interactuando	●	●	●	●			●	●
21	Rango de tareas asignada por módulo		●	●				●	
22	Velocidad de comunicación entre módulos		●	●				●	
23	Tipo de información comunicada	●		●					●
24	Rugosidad de la superficie				●				
25	Inclinación de la superficie				●				
26	Número de configuraciones para locomoción de robot móvil				●				
27	Rango de dimensiones para el dispositivo				●	●	●		
28	Prueba de esfuerzo				●	●	●		
29	Tipos de geometrías disponibles para el robot			●	●	●	●		
30	Materiales aptos para el robot				●	●	●		
31	Velocidad de respuesta a los estímulos		●	●				●	
32	Frecuencia máxima de cambio del estímulo		●	●				●	
33	Alcance máximo de la comunicación							●	●
34	Tiempo mínimo de autonomía del robot								●
35	Voltaje de recarga de las baterías								●
36	Tiempo mínimo de recarga de las baterías								●
37	Capacidad de carga								●

2.3 Generación del concepto

Los conceptos son descripciones concisas de las formas en que se satisfarán las necesidades del robot. Son, por lo general, bosquejos o modelos aproximados acompañados por una breve descripción.

La generación de conceptos para este trabajo se vio enriquecida de la experiencia en el diseño de *MUNAMBOT*, gracias a la cual se desecharon algunos conceptos que no solucionarían las necesidades de este robot.

MUNAMBOT fue la primer aproximación a un robot modular por parte del equipo que marcó una pauta en el diseño conceptual del presente desarrollo. Se retomaron varios de los conceptos generados tanto para la mecánica, electrónica y programación; si bien los conceptos generados durante ese proyecto sirvieron como base, las necesidades del robot forzaron a generar algunos nuevos.

Las propuestas son organizadas de tal manera que se satisfaga la definición de un sistema mecatrónico y muestre sus subsistemas.

2.3.1 Propuestas para la mecánica

A) Geometría

Las propuestas de concepto para la geometría del robot derivaron en cuatro ideas principales:

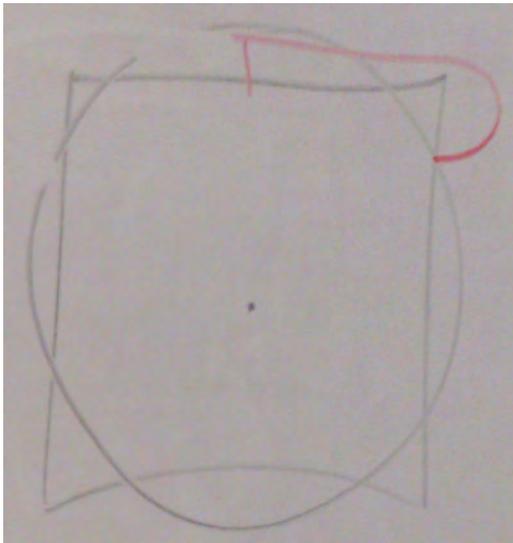
- Forma cúbica.
- Forma prismática rectangular.
- Forma piramidal.
- Forma esférica.

Sin embargo, por facilidad de diseño y manufactura, todas las ideas, salvo la forma cúbica, fueron descartadas. El resto de los conceptos tanto para la mecánica como la electrónica se basaron en la idea de que el robot tendría forma cúbica.

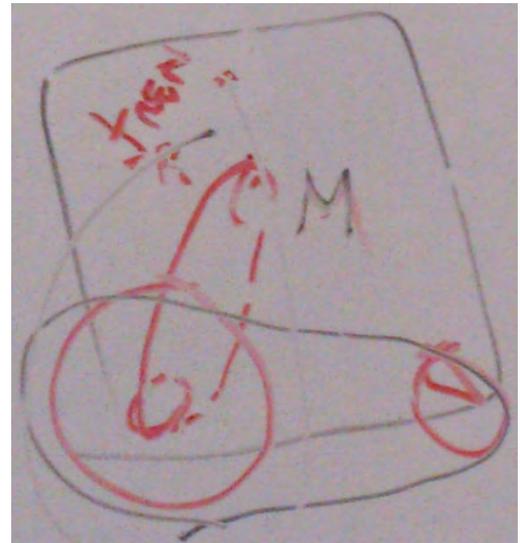
B) Locomoción

Los conceptos para la locomoción del robot se retomaron de las ideas generadas para *MUNAMBOT* [57], de entre los más representativos destacan:

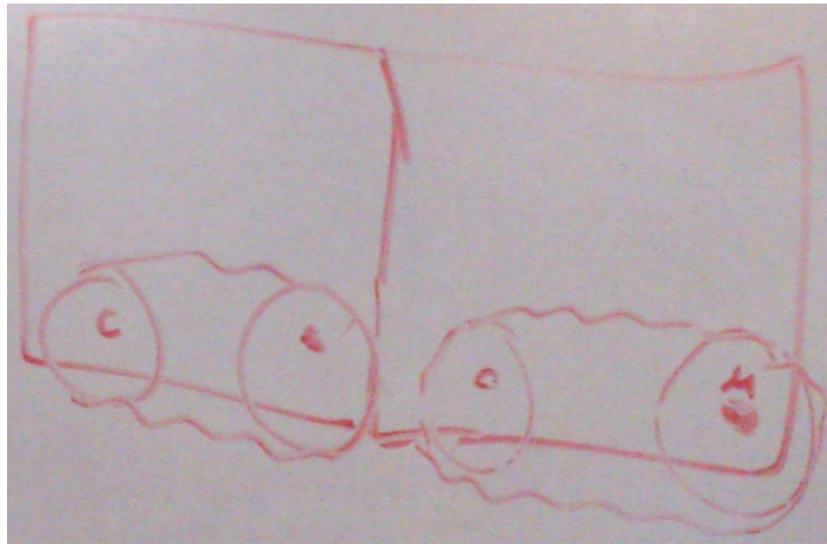
- El robot contaría con dos ruedas de diámetro superior al cuerpo del robot y un solo motor por rueda (Figura 2.1-a).
- Una oruga en la parte inferior cuya tracción estaría dada por un solo motor (Figura 2.1-b).
- Una oruga en la parte inferior con tracción dada por dos motores (Figura 2.1-c).



a)



b)



c)

Figura 2.1: Principales propuestas de locomoción realizadas para *MUNAMBOT* que se retomaron para el robot.

C) Soportes para la electrónica

En *MUNAMBOT* las placas electrónicas iban ensambladas en las diferentes contracaras del cubo, la idea era bastante deficiente y desperdiciaba todo el espacio interno, aparte de generar placas con formas que tenían que adaptarse a espacios reducidos.

Las propuestas van encaminadas a aprovechar lo más posible de los espacios internos:

- El cubo tendría una sola tapa y las placas se montarían empleando 4 tornillos y separadores de plástico (Figura 2.2-a).
- Las placas se pegarían a una tapa en forma perpendicular y el ensamble sería insertando toda la electrónica junto con la tapa (Figura 2.2-b).

- El cubo tendría 2 tapas y las placas se podrían insertar de cualquier lado a la pieza central, similar a una repisa (Figura 2.2-c).

El principal reto de todas las propuestas era determinar dónde se colocarían las baterías, que ocuparían un espacio considerable dentro del robot.

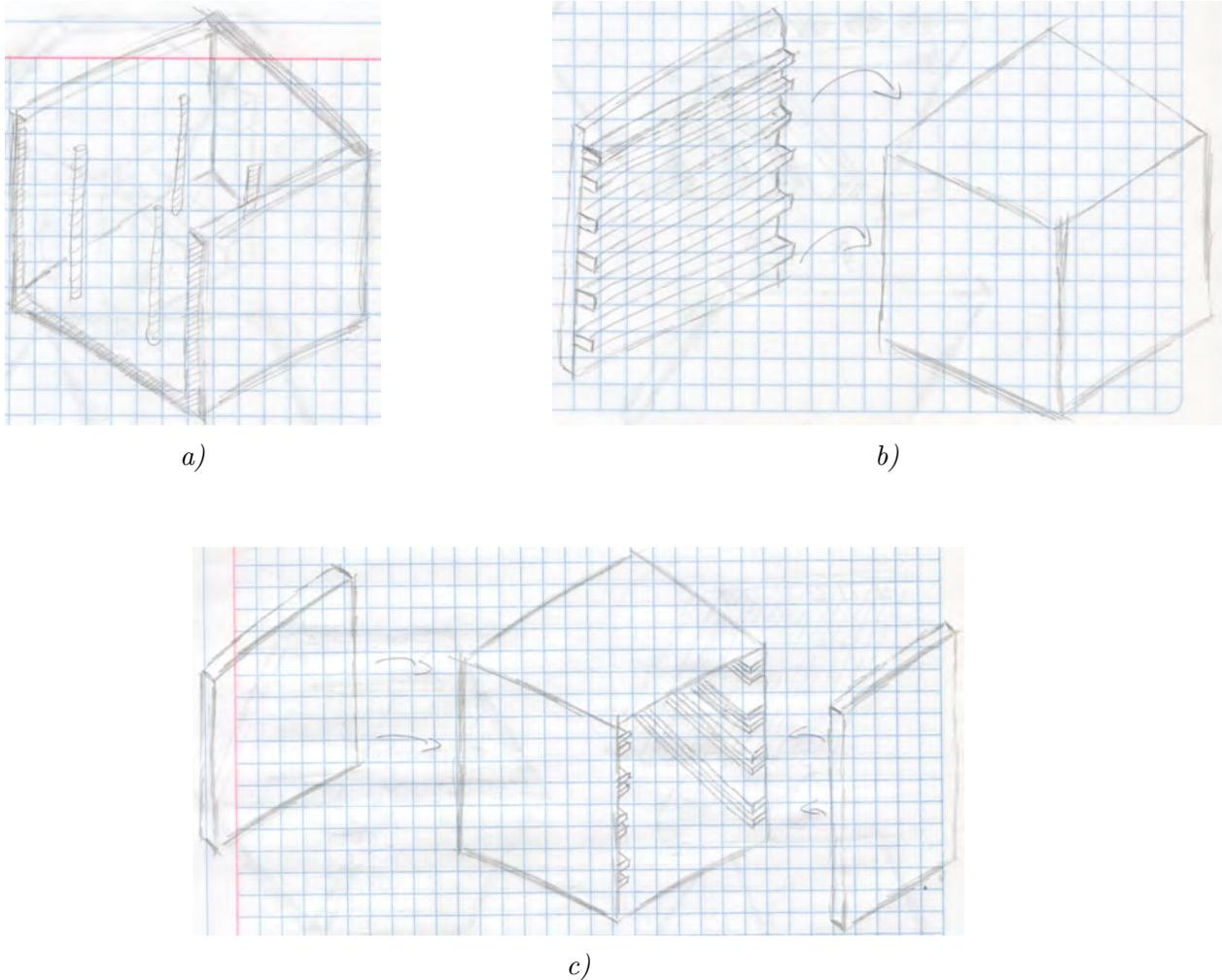


Figura 2.2: *Propuestas para los soportes de las placas electrónicas; todas las propuestas están encaminadas en aprovechar el espacio interno del robot lo mejor posible.*

D) Propuestas para cerrar el robot

El robot tiene que ser flexible en su cerrado para poder realizar cambios y ajustes en las pruebas a realizar. Las principales propuestas son las siguientes:

- Cerrar la(s) tapa(s) empleando 4 pijas o 4 tornillos y tuercas (Figura 2.3-a).
- Cerrar la(s) tapas(s) mediante una especie de clip plástico similar al que se usa en controles remotos, etc., (Figura 2.3-b).

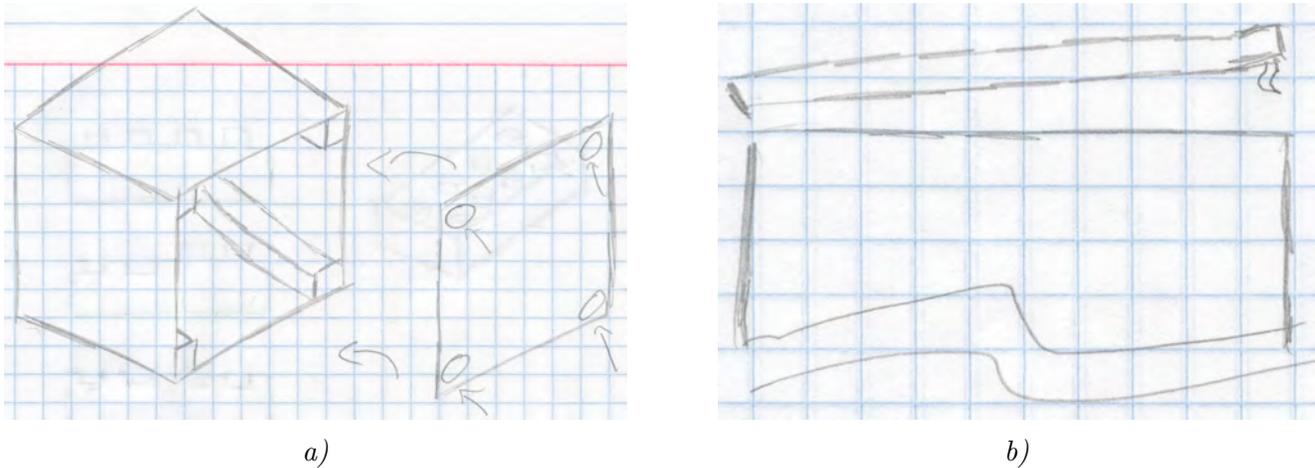


Figura 2.3: *Propuestas para cerrar el robot. La propuesta a) determina usar 4 pijas o tornillos y tuercas y la propuesta b) determina emplear un clip plástico como los que se emplean en controles remotos.*

E) Integración con otras tecnologías - pines de **LEGO**

Durante el proyecto de *MUNAMBOT* se seleccionó usar una configuración de tener 4 bloques de 8 pines por cara, donde 4 de los pines serían machos y 4 de los pines serían hembras (Figura 2.4-a).

La idea de la posición de los bloques se mantiene y se vuelven a proponer 3 posibles configuraciones para la forma que tendrán los pines tipo *LEGO*, mismos que se aprecian en la Figura 2.4-b:

- Todos los pines son machos.
- La mitad de los pines son machos y el resto hembras.
- Todos los pines son hembras.

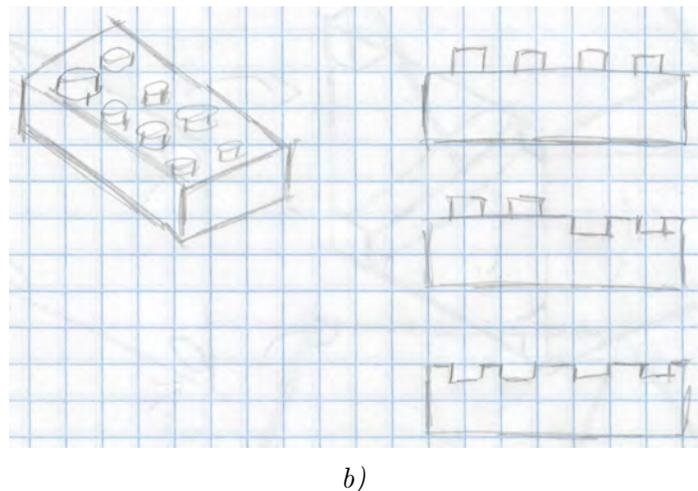
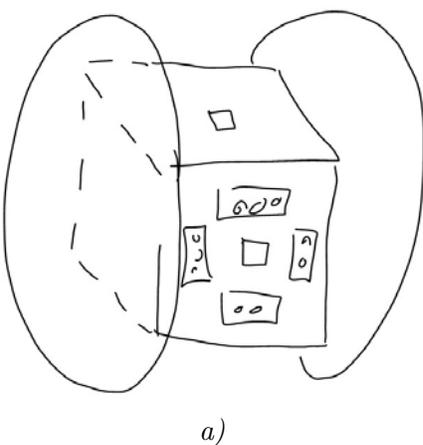


Figura 2.4: *a) Concepto seleccionado para los pines tipo **LEGO** en *MUNAMBOT*; b) conceptos con diferentes combinaciones para los pines de **LEGO**: machos, combinados o hembras.*

2.3.2 Propuestas para la electrónica

Se buscan diferentes tipos de sensores y actuadores que puedan cumplir con las necesidades; por conveniencia se engloban en la electrónica el “cerebro” del robot y la comunicación inalámbrica que tiene, así como el suministro de energía al robot.

A) Medición de distancia

Hay varios métodos para medir la distancia a un objeto; se seleccionaron los sensores comerciales más comunes y con mayor disponibilidad en tiendas de electrónica al menudeo para tener la capacidad de realizar pruebas cuando fuera necesario. Los sensores comerciales disponibles son:

- Sensor ultrasónico HC-SR04: sensor que emplea eco mediante un emisor y un receptor ultrasónicos para detectar la distancia a un objeto desde 2 [cm] hasta 4 [m] en un ángulo de $\pm 15^\circ$ que funciona a través de un protocolo paralelo de tipo I/O y su salida es lineal.
- Sensor infrarrojo *Sharp*: existen distintos modelos que manejan distintos rangos, sin embargo el típico tiene un rango de detección de 10 a 80 [cm]. La salida del sensor no es lineal y es analógica.

B) Medición de velocidad de los motores

Para medir la velocidad de los motores se pueden recurrir a sensores internos montados en la flecha de un motor o bien sensores que midan la velocidad directamente sobre las llantas. Para cualquiera de los casos, se obtuvieron las siguientes propuestas para medir la velocidad de los motores:

- Soldando un imán a la flecha del motor y montar un sensor magnético rotatorio. Existen varios modelos comerciales que realizan esa tarea y sólo estaría limitado a conseguir motores con flecha extendida para poder pegar el imán de alguna manera, perpendicular a la flecha. Este tipo de sensores tienen salida mediante protocolos seriales o analógicos.
- Encoders diseñados para ser montados en motores únicos. Esta solución involucra encoders de encuadratura que se sueldan directamente en algunos modelos de motores diseñados para trabajar con estos encoders. La salida es de tipo I/O de encuadratura, por lo que la señal debe ser decodificada.
- Sensor infrarrojo montado sobre las caras del robot; de esta manera leería cada que la estructura de la llanta pasa frente al sensor. El sensor sería visible y para medir la velocidad sería necesario contar con las llantas colocadas en todo momento.

C) Medición de la orientación

La idea es que un solo robot puede desempeñar las tareas que se le programen directamente sin el apoyo de ningún otro robot o computadora, por lo que un sistema de visión externo contradice la idea original del robot. La medición de la orientación se debe realizar entonces mediante sensores internos en el robot. Después de analizar varias soluciones posibles se llegó a la conclusión de que las opciones más viables son:

- Emplear una IMU, de 6 ó 9 ejes, para medir la orientación y posición del robot.

- Emplear encoders de encuadratura que midan aparte de la velocidad de los motores, el ángulo al que se desplaza el robot mediante relaciones matemáticas.

La segunda solución forzaría a que la medición de velocidad se realice mediante el uso de encoders internos.

D) Motores disponibles

La búsqueda de motores estuvo enfocada a dos rubros principales: con flecha extendida para poder emplear encoders o sin flecha extendida para emplear las demás soluciones. Todos los motores deberían contar con caja de reducción para tener el torque suficiente para mover al robot, aunque se moviera lento. Las opciones encontradas fueron:

- Micro motor *Pololu* metálico de flecha extendida.
- Micro motor *Pololu* plástico.
- Motores con reductor *Gorila*.

Las llantas se buscan de modelos comerciales existentes; dada la experiencia de diseño de *MUNAMBOT*, diseñar e imprimir en 3D las llantas no es una solución óptima. Para todas las opciones encontradas existen llantas que hacen juego y se pueden embonar de manera directa o usando un cople. Las llantas de *Pololu* existen hasta en diámetro de 9 [cm] y las llantas que hacen juego con los motores *Gorila* existen en una amplia gama de llantas de marcas genéricas de diversos diámetros y grosores.

E) LEDs para las caras

Los LEDs RGB existen en diversos empaquetados tanto de montaje superficial como de montaje por agujeros, cada uno con sus características físicas propias. Existen en versiones tanto de ánodo común como de cátodo común, o bien, con unidades de control embebidas que se comunican al procesador del robot. Los principales LEDs tomados en cuenta para el concepto son:

- LEDs redondos de 5 [mm], en cualquiera de las presentaciones de ánodo o cátodo común. Perforarían las caras del cubo para ser montados. Esta propuesta involucra tener que implementar un controlador para los colores e intensidad lumínica de los LEDs.
- LEDs de montaje superficial en cualquiera de las presentaciones de ánodo o cátodo común. Estarían montados justo en la superficie de las caras del robot. Esta propuesta involucra tener que implementar un controlador para los colores e intensidad lumínica de los LEDs.
- LEDs de montaje superficial con unidad de control. Estarían montados justo en la superficie de las caras del robot.

F) Fuente de energía

La autonomía planteada para el robot se tiene que dar mediante el uso de baterías. En el mercado existe una diversidad de baterías de distintos tipos gigantesca; van desde las pilas alcalinas que no son recargables (de tamaños AA y rectangulares de 9 [V]) hasta las recargables de NiMH, ión-litio y Li-Po (en una gama de tamaños, formas y capacidades muy diversas). Por lo mismo, los conceptos se vieron reducidos a los candidatos que mejor se ajustarían a las necesidades, siendo:

- Batería recargable de 9 [V] de NiMH.
- Batería recargable de ión-litio propia de algún modelo de teléfono celular.
- Batería recargable de Li-Po.

Aunque se descartó el circuito recargador de batería interno en el robot, cabe mencionar que existen cargadores comerciales para cada una de las opciones mencionadas anteriormente, por lo que cualquier diseño podría ser probado con éxito recargando las baterías de manera externa.

G) Comunicación inalámbrica

La comunicación inalámbrica le permitiría al robot comunicarse con otros semejantes y con el usuario a través de la interfaz, compartiendo la información que así se le programe.

Aunque se tomó en consideración la posibilidad de emplear un receptor y un emisor infrarrojos (similar al concepto de funcionamiento de un control remoto) esa solución imposibilita la comunicación entre más de 2 robots de manera simultánea y no podría haber comunicación con el usuario. Por lo mismo los conceptos que solucionarían la comunicación inalámbrica son:

- Radiofrecuencia: un trasceptor (emisor-receptor) y un controlador a una cierta frecuencia instalado en el robot. El usuario necesitaría un adaptador para la computadora para recibir y transmitir mensajes. Todo mensaje sería recibido por todo receptor, por lo que se tiene que implementar un protocolo que indique si el mensaje es específico a algún dispositivo o es general.
- Bluetooth: a través de protocolos existentes, los robots se comunicarían formando una red junto con el usuario. Es probable que el usuario necesite de un adaptador Bluetooth. Los mensajes deberán ser retransmitidos dado que la conexión Bluetooth se realiza punto a punto.
- Wifi: una solución que involucra que exista un ruteador, todos los robots y el usuario estarían conectados en una misma red por donde se pueden mandar los mensajes a un dispositivo específico o de manera abierta a todos.

H) Procesador central

De la experiencia pasada con *MUNAMBOT* se aprendió que todas las tareas de sentido y actuación no se alcanzan a desempeñar sin retraso empleando microcontroladores *PIC16F887*, de 8 bits y cuya capacidad de procesamiento es de 5 MIPS (millones de instrucciones por segundo).

El cambio de 2 microcontroladores *PIC16F887* a cualquier otra combinación implicaría un aprendizaje del nuevo procesador.

Después de analizar varias soluciones, dispositivos y *chips*, se determinó que las mejores soluciones de mayor potencia para el procesador son:

- Mantener la configuración de 2 microcontroladores, ahora de 16 bits, y mantener separadas las funciones de sentido y actuación con la de planificación; las marcas de microcontroladores a seleccionar serían: *Microchip*, *Texas Instruments*, *Atmel* y *Freescall*, marcas de microcontroladores que en algún momento de la carrera se abarcaron aunque sea de manera breve.

- Emplear una sola microcomputadora en tarjeta como la *Raspberry Pi 2*, *UDOO*, *ODROID XU3* o *BeagleBone*, en cuyo caso la mecánica interna del robot tendría que adaptarse a la forma de la tarjeta.

2.4 Selección del concepto

La metodología de diseño propuesta por los autores sugiere el uso árboles y tablas de selección, donde se ponderan todas las posibles combinaciones entre los conceptos generados, descartando los conceptos cuya solución sea inviable o se aprecie inviable por las razones que sean, acortando así el número de combinaciones posible.

Sin embargo, dado que el robot se compone de distintos subsistemas que no necesariamente van ligados el uno con el otro, la selección cruzada sólo se discutió para los elementos que realmente lo ameritaban. El resto de los conceptos se discutían de manera individual.

2.4.1 Selección de la mecánica

Dado que el robot se había seleccionado ya previamente con forma cúbica, y bajo la cual se generaron los demás conceptos para la mecánica del robot, se abarcan entonces los demás conceptos de la mecánica.

A) Locomoción

Como se mencionó en el concepto de las llantas, se determinó por experiencia no diseñar e imprimir en 3D las llantas para el robot sino comprar algún modelo comercial existente. Bajo esa premisa, se busca la solución más simple para la locomoción del robot, donde la mecánica sea lo más simple posible y evitar arreglos complejos.

Las soluciones de oruga le permitirían al robot desplazarse por varias superficies fácilmente, sin embargo, el arreglo mecánico incluye distribuir la transmisión de la potencia del motor a las orugas mediante poleas o bandas o simplemente poner dos motores en paralelo. La primera solución es bastante más compleja y ocuparía espacio disponible para las placas electrónicas y la segunda forzaría a diseñar un robot muy grande para poder acomodar dos motores en paralelo.

La solución más simple y viable entonces es colocar un motor al centro del costado que sostenga la llanta directamente desde la flecha de la reducción. Gracias a esta solución se evitan arreglos mecánicos complejos, se ocupa un solo motor por llanta y el robot se podría desplazar apoyado sobre dos caras distintas. Las dimensiones máximas del robot estarían limitadas a las dimensiones de las llantas que se puedan encontrar.

B) Soportes para la electrónica y cerrar el robot

Ambos conceptos están íntimamente ligados. Las combinaciones posibles son que el robot tenga dos tapas y soportes como estante de panadería, cerrando el robot en 4 puntos; o una sola tapa con soportes en la tapa o internamente en el cuerpo cerrando con una especie de clip.

Dado que el robot se diseñaría con la menor dimensión posible, lo más conveniente sería acceder a la electrónica a través de dos lados y no sólo de uno. El tener aberturas por ambos costados del robot permitiría insertar las placas y acceder a ellas para realizar cualquier conexión de manera más sencilla a que si se coloca todo con una sola abertura.

Aunque el cerrar el robot de una sola cara ayudaría que fuera más sencillo el ensamble, es preferible tener más acceso a la electrónica en caso de conexión o falla, para así evitar desensamblar todo el robot cuando sea necesario por alguna razón.

Abrir el robot mediante un clip es notablemente más sencillo que emplear 4 tornillos o pijas, pero el emplear esos 4 tornillos o pijas, dejando una costilla en cada esquina con material, le daría rigidez al cuerpo del robot. Al final, el ensamble sería más rígido con los 4 tornillos o pijas que con un clip.

Dado que las pruebas del robot se realizarán en un modelo impreso en 3D, se desconoce el daño a la estructura o el nivel de fatiga que pueda tener estar atornillando y desatornillando una pija cada ocasión que se vaya a abrir o cerrar el robot. Para evitar eso, se opta mejor por la solución de emplear tornillos y tuercas; las tuercas estarían embebidas en el robot y el tornillo se ensamblaría en la tuerca, evitando así que se deba generar la cuerda en el polímero y sólo se tendría un agujero por donde pasaría el tornillo.

C) Pines de *LEGO*

Aunque en *MUNAMBOT* se intentó dar variedad al usuario para colocar piezas de *LEGO* en uno u otro sentido, empleando 4 pines macho y 4 hembras; se decidió recurrir a la costumbre y dejar todos los pines como machos, asemejando la terminación de un bloque de 4×2 .

2.4.2 Selección del procesador

Antes de decidir el resto de la electrónica se tuvo que decidir el procesador que el robot emplearía. Dado que, tanto para los microcontroladores como para las microcomputadoras existen configuraciones de alimentación de 5 [V] y de 3.3 [V], se tenía que decidir primero el procesador para después determinar la configuración de alimentación necesaria para el resto de la electrónica.

La selección se hace bajo la premisa de que cualquiera de las posibles soluciones implicaría un aprendizaje dado que ninguno de los dispositivos disponibles eran familiares ya o se tenían muy pocas horas de trabajo con algunas.

Aunque la idea de usar una microcomputadora le brindaría un poder de procesamiento muy grande al robot, facilidad para conectar periféricos y la suficiente cantidad de pines I/O para controlar el resto de la electrónica interna; las microcomputadoras presentan de arranque la desventaja de que el modelo físico tendría que adaptarse a las dimensiones de la tarjeta, entrando en conflicto con la solución mecánica de basar el tamaño del robot a la diversidad de llantas comerciales; el encendido del robot sería lento porque hay que esperar un tiempo en que el sistema operativo instalado en la tarjeta cargue y posteriormente pueda correr el programa interno que controle al robot y consumen hasta 2.5 [A] de corriente cuando el procesador se encuentra a máxima demanda de procesamiento, lo que forzaría a buscar una batería muy grande para el robot, provocando de

igual forma que se tenga que agrandar el tamaño del robot.

La solución que pudiera adaptarse a las dimensiones del robot sin consumir tanta energía es entonces el emplear 2 microcontroladores y dividir las tareas. De experiencias pasadas se aprendió que microcontroladores usados para la enseñanza, como el *PIC16F887*, no tienen la capacidad suficiente para procesar toda la información que involucra al robot a pesar de que se empleen dos en paralelo para las tareas. Por ende se seleccionan microcontroladores de una gama más alta.

Para seleccionarlos se toman en cuenta los siguientes puntos:

- Los microcontroladores *PIC16F887* pueden ejecutar hasta 5 MIPS (millones de instrucciones por segundo). La capacidad de procesamiento debe ser mayor.
- En caso de seleccionarse como opción, el sensor más especializado es la IMU, cuya información es de 16 bits, por lo que el nuevo microcontrolador debe tener un procesador de 16 bits nativo para manejar datos tipo *word* de manera nativa.
- Todos los microcontroladores a los que estábamos habituados son de 8 bits; el pasar de 8 a 16 bits es una transición razonable en términos de tiempo de aprendizaje y capacidad de procesamiento; una transición a 32 bits da la posibilidad de tener una mayor capacidad de procesamiento aún, pero llevaría más tiempo aprender a usarlo.
- El microcontrolador debe tener un empaquetado tanto en montaje superficial como DIP para poder aprender a usarlo sin la necesidad de comprar una placa de desarrollo.
- El microcontrolador debe tener incluido por hardware elementos que faciliten la lectura de los sensores, como los protocolos UART, I²C, decodificación de encoders de encuadratura y algún procesador de señales.
- Debe de trabajar a 5 [V], por la costumbre de siempre haber empleado ese voltaje.
- Deben ser microcontroladores que resistan el uso rudo mientras se experimenta con ellos.
- Un microcontrolador de *Microchip* tendría una curva de aprendizaje más rápida al ser familiar.

Existen muchos fabricantes de microcontroladores, cada uno tiene una muy amplia gama de modelos. Los fabricantes más comunes que se llegan a emplear en la Facultad son *Microchip*, *Atmel*, *Texas Instruments* y *Freescale*. Sólo *Atmel* no cuenta con una gama de 16 bits, dando el brinco de microcontroladores de 8 bits a microcontroladores de 32 bits.

De las marcas restantes, *Texas Instruments* no es de nuestro uso familiar, por lo que también se descarta. *Microchip* y *Freescale* ambos tienen gamas de *DSCs* (digital signal controllers, por sus siglas en inglés), que combinan una máquina simple DSP (digital signal processor, por sus siglas en inglés) con la facilidad de manejo de un MCU (microcontroller, por sus siglas en inglés), sin embargo, los ICs de *Freescale* son altamente sensibles a la estática y se arruinan con facilidad.

Los DSC elegidos son entonces los de *Microchip*, por familiaridad, resistencia al uso rudo y flexibilidad de modelos.

2.4.3 Selección de la electrónica

Una vez seleccionados los procesadores del robot se continúa con la selección del resto de componentes electrónicos que conformarán al robot.

A) Medición de distancia

La ventaja que muestra el sensor infrarrojo frente al ultrasónico es que la velocidad de lectura se puede realizar a altas frecuencias de muestreo que depende de la frecuencia a la que reacciona el receptor infrarrojo; a diferencia, el sensor ultrasónico requiere un tiempo para emitir las ondas y recibir el eco, para posteriormente obtener una lectura equivalente en tiempo del tiempo en el que tardó en regresar la onda.

Adicionalmente, al no depender de un protocolo de lectura, el sensor infrarrojo tiene la ventaja de poder ser empleado tal cual es y sólo implementar la lectura mediante el convertidor analógico a digital del microcontrolador.

Sin embargo, la señal infrarroja acarrea un nivel de ruido importante que forzaría a implementar un filtro de manera nativa, tiene un ángulo de detección relativamente reducido en comparación al del ultrasónico y tiene un consumo de corriente 4 veces mayor; de igual forma, muchos materiales y colores absorben el haz emitido, dando lecturas erróneas. Aunque el sensor ultrasónico requiere un protocolo que involucra medición de tiempos, la lectura tiene menos ruido y da un valor más fiable de la distancia a la que se encuentra el objeto, a pesar de que algunos materiales (telas principalmente) dan lecturas erróneas, son los mínimos. La selección es entonces el sensor ultrasónico *HC-SR04*.

B) Medición de velocidad de los motores

La toma de decisión del sensor de velocidad para los motores involucra analizarlo en conjunto con los motores.

Para poder conectar un encóder a un motor con reducción, la flecha principal debe tener una manera de acoplarle el encóder de encuadratura, ya sea con una flecha extendida o recortando el extremo del motor para acoplar una estrella o un imán. Mecánicamente hablando, un motor más pequeño permitiría diseñar un modelo más pequeño y un motor que de manera nativa esté preparado para emplear encoders evitaría generar plataformas adicionales para sostener los encoders.

De los modelos comerciales encontrados, sólo los micromotores *Pololu* existen en versión con flecha extendida para colocar encoders, de los cuales, los mismos fabricantes cuentan con dos modelos de encoders de encuadratura: encoders ópticos y encoders magnéticos. Todos estos encoders se sueldan a las terminales del motor y se coloca en la flecha extendida una estrella o un imán.

Ninguno de los demás motores evaluados cuenta con versión de flecha extendida, por lo que para soldar un imán o un encóder sería necesario modificar la cubierta del motor para hacer modificaciones, lo que podría provocar fallos en el motor o errores en la colocación del encóder.

La idea de emplear un encóder externo que sense la estructura de la llanta se descarta por la necesidad de fabricar soportes internos y circuitería adicionales para conectarlos, el encóder podría

estar sujeto a interferencia de luz o de cualquier objeto que se atravesara en su rango de detección e implicaría que el encóder sólo funcionaría siempre y cuando se tengan ensambladas las llantas al robot, lo cual limitaría su aplicación y la capacidad de personalización futura en caso de que se desee acoplar cualquier otro objeto a la reducción del motor.

Finalmente, el motor que se selecciona es el modelo con reducción 297:1 de baja potencia. El que sea de baja potencia contribuye a que la batería que se seleccione no sea de gran tamaño pero provee un par lo suficientemente grande para desplazar al robot aunque sea a una velocidad lenta.

Los encoders seleccionados son los encoders ópticos. Aunque la salida de ambos encoders (ópticos y magnéticos) es la misma de encuadratura, se desconoce si el tener dos motores cerca el uno de otro afectaría la lectura entre ambos encoders y si el campo magnético podría llegar a afectar el resto de la circuitería. Tomando eso en cuenta, se opta por los encoders ópticos, considerando que una barrera mecánica en el soporte de los motores podría llegar a ser necesaria para evitar interferencia entre los encoders dado que ambos encoders se encontrarían de frente una vez montados en el robot.

Por consiguiente, las llantas que se seleccionan que hacen juego con los motores seleccionados son las más grandes con las que cuenta en su catálogo *Pololu*, llantas de 90 [mm].

C) Medición de la orientación del robot

Dado que se seleccionó previamente que los encoders serían internos, ambas opciones planteadas para medir la orientación del robot estaban habilitadas para su análisis.

De experiencias en proyectos pasados se aprendió que la orientación de un robot se puede medir gracias a los encoders de encuadratura gracias a una relación matemática que toma en cuenta el número de pulsos que ha avanzado cada motor. El promedio de pulsos indica la orientación del robot.

Sin embargo, esto sólo funciona en un plano ideal; durante el mismo proyecto se observó que cuando las llantas patinan, la lectura es errónea y el valor medido es incorrecto. Inclusive si alguno de los encoders falla, el robot intentará seguir midiendo el ángulo pero sólo tendrá disponible la información de un encóder, lo cual lleva a fallas y comportamientos erróneos.

Una IMU también tiene sus desventajas. Generalmente la información que entregan tiene un nivel de ruido considerable, son más complejas de usar e involucra generar un circuito adicional al existente de los encoders, circuito con pistas muy delgadas dado que estos dispositivos son de unos pocos milímetros de longitud. A pesar de todo ello se considera como la solución más viable porque una IMU de 6 ejes le brindaría la posibilidad al robot de conocer su orientación no sólo en un plano sino en el espacio y le brindaría información de aceleraciones, útil en caso de golpes, caídas o hasta para detectar movimiento.

Es muy probable que para usar la información que brinda la IMU sea necesario programar filtros y quizá hasta un análisis matemático complejo para aprovechar al máximo la información que se puede extraer de este sensor.

D) Selección de LEDs

Los LEDs que irán en las caras del robot tienen tanto función interactiva como estética. Por lo mismo se busca que sean los más atractivos posibles.

Los LEDs redondos de 5 [mm] son grandes y hasta cierto punto estorbosos, aunque se pueden manejar con facilidad y su uso sería atravesando las paredes del robot con los LEDs, sin necesidad de generar soportes adicionales, sólo sería necesario extender los cables que los conecten al circuito controlador.

Los LEDs de montaje superficial son más discretos, pero su uso requiere forzosamente de placas especiales debido a la forma en la que se sueldan los componentes; adicionalmente, en caso de fallo, sería muy complejo su reemplazo sin dañar al robot. Por su parte, los LEDs que incluyen un controlador tienen un precio individual muy caro y son inviables para su experimentación.

Los LEDs que se seleccionan son los redondos, por su amplia disponibilidad, fácil colocación y conexión y fácil manipulación. El circuito controlador sería el mismo con mínimas adaptaciones para LEDs de ánodo común o de cátodo común, por lo que el tipo de circuito dependerá del tipo de LEDs que se consigan más fácilmente.

E) Fuente de energía

Existe una enorme gama de tipos de batería disponibles en el mercado, por lo que se desechan rápidamente las baterías recargables de 9 [V] al no existir baterías que puedan brindar un nivel de corriente alto (tan sólo los motores exceden la capacidad de carga de esas baterías).

Las baterías de Li-Po son baterías con un excelente nivel de carga y ciclos de recarga, se emplearía una de dos celdas para alimentar el robot. Lamentablemente requieren un circuito de recarga muy especializado del que se desconoce su implementación. Dado que el objetivo del proyecto no es entender cómo funcionan las baterías Li-Po ni cómo recargarlas de la manera más eficiente se descarta la idea.

Las baterías que se seleccionan son las baterías ión-litio propias de algún modelo de teléfono celular. Existe una amplia gama de diversas capacidades de corriente y su selección final se realizaría una vez analizado el consumo total de corriente del robot. De igual manera se ocuparían dos celdas de estas baterías para alimentar al robot. Estas baterías tienen la flexibilidad de que se pueden recargar empleando directamente un eliminador a un cierto voltaje, aunque reduce sus ciclos de recarga, pero son fáciles de manipular. Adicionalmente, existen diseños compactos que serían ideales, dado que están diseñadas para alimentar un dispositivo que actualmente requiere que los diseños sean cada vez más delgados.

F) Comunicación inalámbrica

La radiofrecuencia tiene la versatilidad de que empleando un buen transceptor se puede lograr la comunicación punto a punto o a todos los dispositivos implementando en cada robot un protocolo de direccionamiento previo a los mensajes que se tienen que enviar. Generalmente los circuitos ya vienen preparados con una antena.

Las desventajas de este tipo de comunicación es que la señal puede sufrir mucha interferencia y

es posible que se llegue a necesitar un tipo de encriptación de la señal dado que cualquier antena podría interceptar la señal y hasta enviar información y/o comandos a los robots. Finalmente, cualquier usuario que requiera monitorear lo que ocurre en la red deberá contar con un adaptador especial que permita conectar la computadora a los robots mediante radiofrecuencia.

El Bluetooth tiene la ventaja de que es una comunicación firmemente establecida y tiene sus propios protocolos de seguridad. Los módulos ya vienen preparados con una antena PCB y es el método con el que se ha experimentado más a lo largo de la carrera, dada la versatilidad de proyectos que se han realizado conectando un microcontrolador con un teléfono celular mediante Bluetooth. Actualmente todo teléfono inteligente y una amplia gama de computadoras personales cuentan con Bluetooth integrado, lo cual permitiría al usuario conectarse a la red fácilmente.

Las desventajas que presenta este tipo de comunicación es que al ser una conexión punto a punto, hay que implementar un protocolo de recepción-envío entre los robots para mandar el mensaje hasta el último robot. Para lograrlo se podría implementar un protocolo que conecte y desconecte rápidamente los módulos para polear si hay nuevo mensaje o no y a quién está dirigido o implementar dos transceptores por robot, armando una cadena.

Cada que un nuevo robot se integre a la red, tendría que buscar por un robot que esté libre para conexión. El implementar la cadena corre el riesgo de que si algún robot se desconecta, debe de existir un protocolo de reconexión y recuperación de la información.

Cada robot tendría que manejar un *buffer* de recepción y transmisión que limpiaría hasta que el mensaje sea completamente enviado y el receptor. Este protocolo en cadena resultaría demasiado lento, sobre todo monitoreando la actividad de la red entera, dado que hay que estar retransmitiendo el mensaje hasta el robot que requiera la información. En el peor de los casos, el mayor retraso de la señal iría desde el usuario hasta el robot más lejano en la cadena y de vuelta con la respuesta hasta el usuario; esto corre el riesgo de cuellos de botella a lo largo de toda la red.

Implementar wifi en los robots implica tener un conocimiento asentado ya para su configuración y manejo pero brinda las ventajas de que un *router* es el encargado de administrar los mensajes y la seguridad en la red. Existen circuitos capaces de actuar de *routers* para un cierto número de dispositivos, por lo que la implementación de wifi no está bloqueada a contar con un *router* inalámbrico, aunque la adición de un *router* permitiría conectar inclusive más de 100 robots en la red.

Una vez configurado, la comunicación vía wifi no requiere protocolos adicionales de encriptación del mensaje o retransmisión de la señal punto a punto, sólo se envía el mensaje a la dirección deseada o en general a todo el mundo. El usuario tiene la posibilidad de monitorear la red con un mayor número de dispositivos que con Bluetooth, dado que este tipo de comunicación tiene un nivel de implementación mayor.

La principal desventaja es el nivel de conocimientos fundamentales que se requieren para aplicar este tipo de comunicación, dada que su teoría es muy amplia. Implementar un módulo wifi que soporte generar la red tiene un precio al menudeo grande y generalmente requiere de adecuaciones especiales al circuito, como el tener que diseñar la antena para el módulo.

Una vez que se evaluaron los pros y contras de los 3 tipos de comunicación inalámbrica, se determinó seleccionar wifi dado que se requiere bastante tiempo para conocer su funcionamiento pero no requiere un trabajo extenso para diseñar protocolos de encriptación o de recepción-transmisión dado que el *router* se encarga de todo eso de manera nativa. Por lo que el wifi es la mejor opción y la que brindaría la mayor flexibilidad en términos de comunicación y facilidad de conexión a un

número muy extenso de dispositivos y hasta de usuarios.

2.4.4 Resumen final de la selección

El robot será un cubo cuyas dimensiones no excedan los 90 [mm] de diámetro de las llantas. Se abrirá por ambos costados para poder insertar las placas electrónicas en soportes tipo *estante de panadería*. El robot será alimentado por baterías de ión-litio propias de algún modelo de teléfono celular.

El sensor ultrasónico será el encargado de sensar la distancia a los objetos que se encuentren frente al robot y se montará al frente del robot; los micro motores con flecha extendida irán montados sobre las tapas del robot y su velocidad será medida mediante los encoders que se acoplan a la flecha extendida.

La orientación del robot será medida empleando una IMU de 6 ejes en donde será necesario evaluar el tipo de filtros que se le acomoden mejor al microcontrolador.

El robot contará con LEDs RGB redondos montados sobre todas las caras del robot y se implementará un circuito controlador para encender de diferentes colores los LEDs. Cabe mencionar que toda cara del robot será independiente, por lo que se podrían tener hasta 6 colores distintos encendidos en el robot.

La comunicación será vía wifi donde se aprenderá lo básico para su configuración y diseño de circuito electrónico para ser soldado en las placas electrónicas.

Todo será controlado por dos microcontroladores *dsPIC* que separarán las tareas de sensado, actuación, comunicación y planificación.

Para esta versión del robot no se implementarán el micrófono, ni la bocina ni el circuito recargador de baterías dada la falta de conocimiento o de recursos para implementar tales circuitos.

Capítulo 3

Diseño a detalle

El capítulo comprende el desarrollo de la mecánica del robot (diseño de la carcasa y tapas) y el diseño de la electrónica del robot (diseño de los circuitos, implementación y diseño de las placas electrónicas).

En este capítulo se detallará el desarrollo de la implementación de cada una de las partes que conforman al robot para dar solución al objetivo de la tesis.

Los sistemas que conforman al robot y su interacción se muestran en la Figura 3.1.

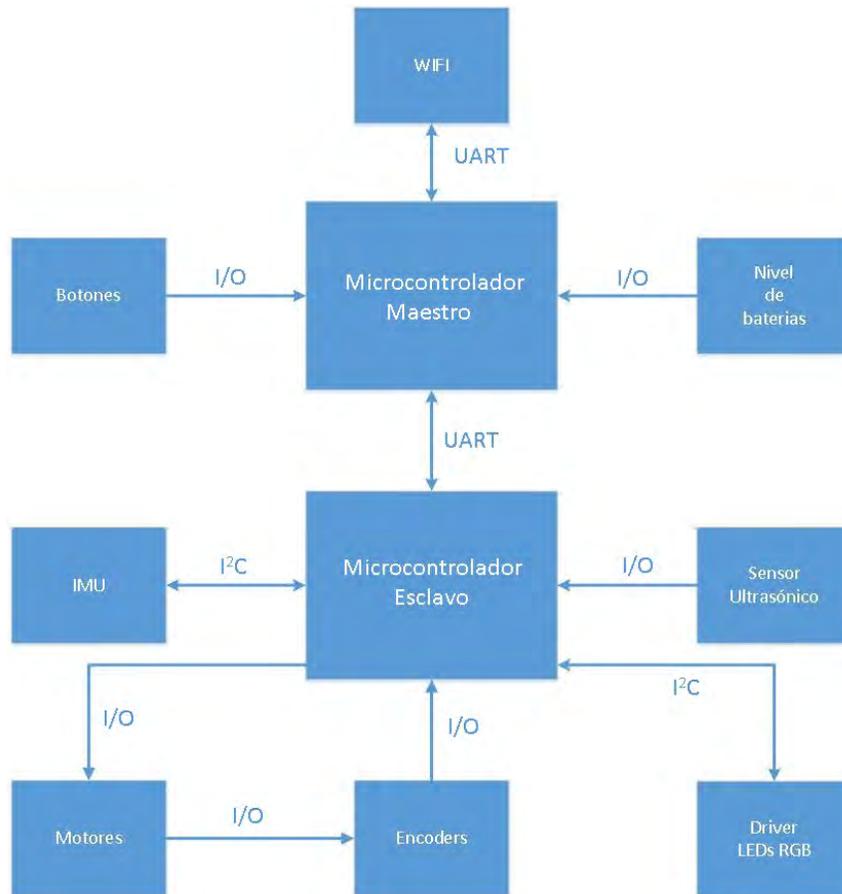


Figura 3.1: Diagrama de interacción entre los componentes del robot.

3.1 Mecánica

En esta sección se detalla el diseño mecánico del robot, que se compone de un cuerpo central, dos tapas y dos *brackets* para los motores; al final se describen brevemente las dimensiones de los micro motores y las llantas. El cuerpo central contiene los soportes para las placas electrónicas, sensores, baterías, LEDs y botones. Las tapas tienen los soportes para los motores, que se sujetan mediante los *brackets*. Las llantas se soportan directamente sobre la flecha de los motores.

El modelo físico se obtuvo mediante impresión 3D en polímero ABS, en colores azul y naranja. Las llantas y los motores son modelos comerciales marca *Pololu*.

En las siguientes secciones se muestran de manera general las dimensiones de las partes que componen al robot y se explican brevemente sus funciones; las dimensiones ya contemplan la tolerancia. Los planos a detalle del robot se encuentran en el apéndice [A Planos mecánicos](#).



Figura 3.2: *Modelo físico de los robots.*

3.1.1 Cuerpo central

El robot tiene una forma cúbica de 80 [mm] de lado y se puede abrir de ambos costados. El cuerpo central (Figuras 3.3) tiene dimensiones de $70 \times 80 \times 80$ [mm] (Figuras 3.4), el restante de la longitud del cuerpo corresponde a las tapas. El grosor de las paredes tanto del cuerpo central como de las tapas es de 2 [mm].

Por facilidad, se le da un indicador a las caras (Figuras 3.5). La cara ‘1’ tiene la ranura para sostener el sensor ultrasónico; en la cara ‘2’ se sostienen los botones de propósito general; en la cara ‘3’ se tiene la ranura para sostener el botón de encendido/apagado. Las tapas se ensamblan sobre las caras ‘5’ y ‘6’ empleando tornillos.

Internamente, se aprecian las ranuras para las PCBs y las baterías de forma paralela a las caras ‘2’ y ‘4’. Aparte, en las caras ‘1’ y ‘3’ se tienen 4 ranuras para insertar una pequeña tuerca para sostener el tornillo al ensamblar las tapas. En todas las caras se colocaron de manera estética pines macho de tipo *LEGO* totalmente funcionales, con la finalidad de que se pueda personalizar al gusto posteriormente con piezas *LEGO*. Las dimensiones de los pines se aprecian en la Figura 3.6.

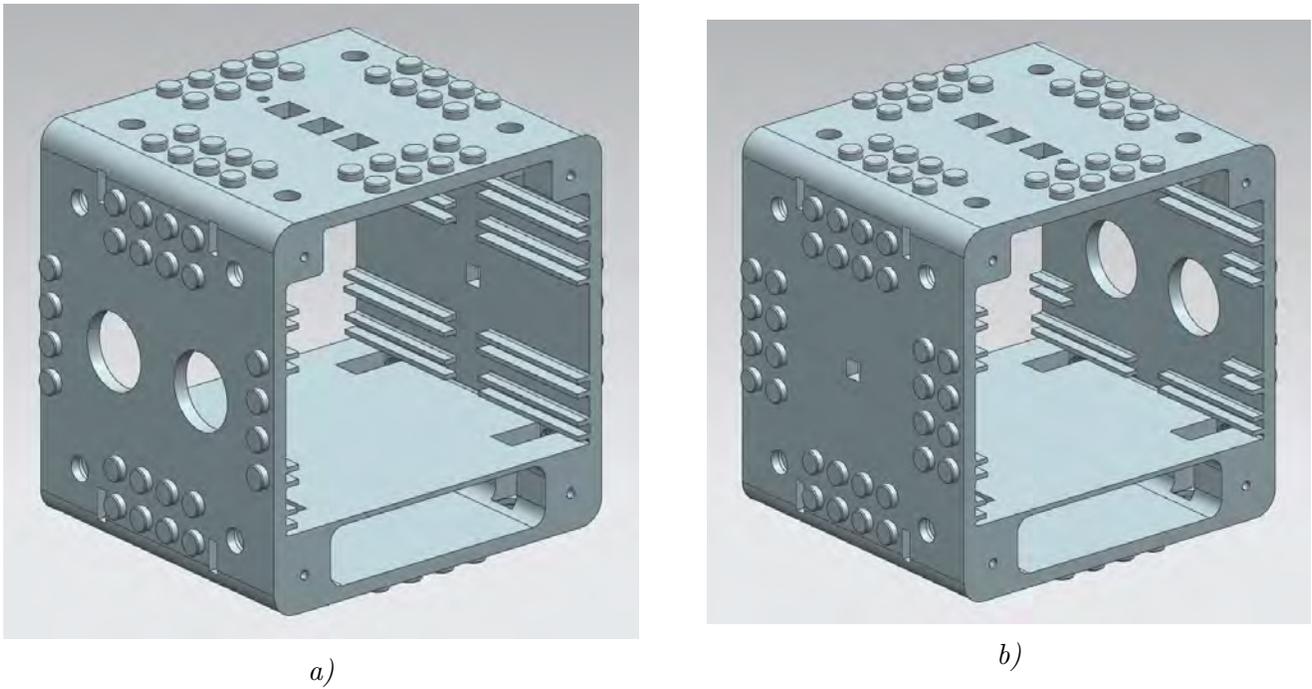


Figura 3.3: *Cuerpo central del robot, que se aprecia en una vista trimétrica, a) de frente, b) por detrás.*

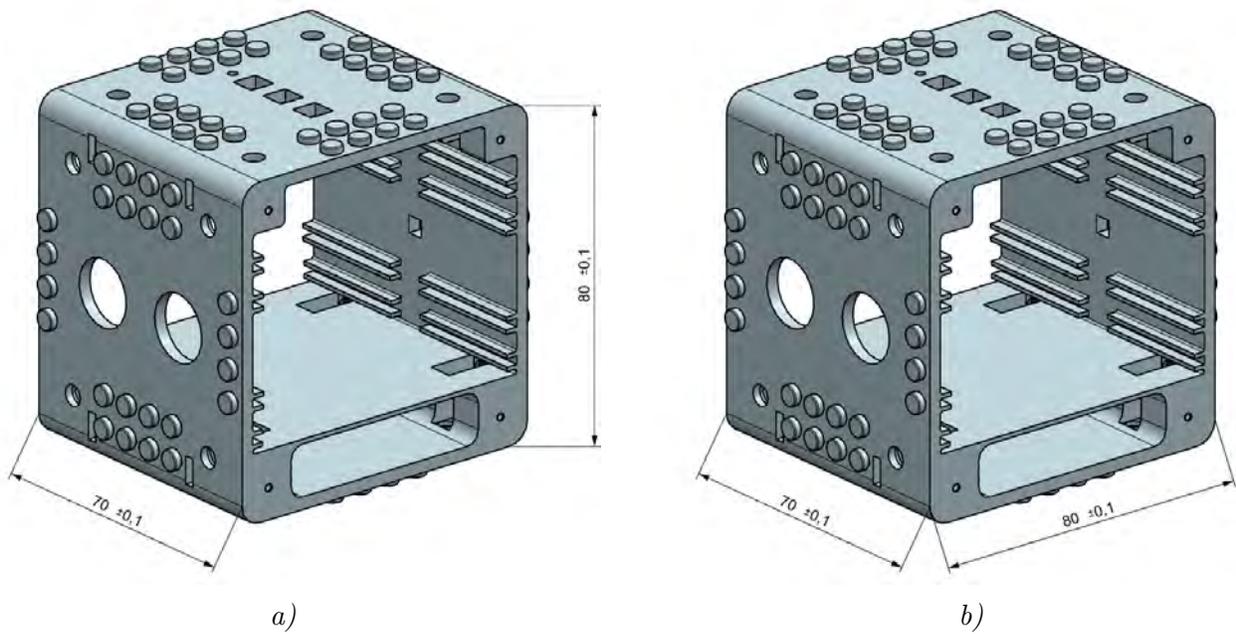


Figura 3.4: *Dimensiones generales del cubo, donde se aprecian: a) largo y alto, b) largo y ancho.*

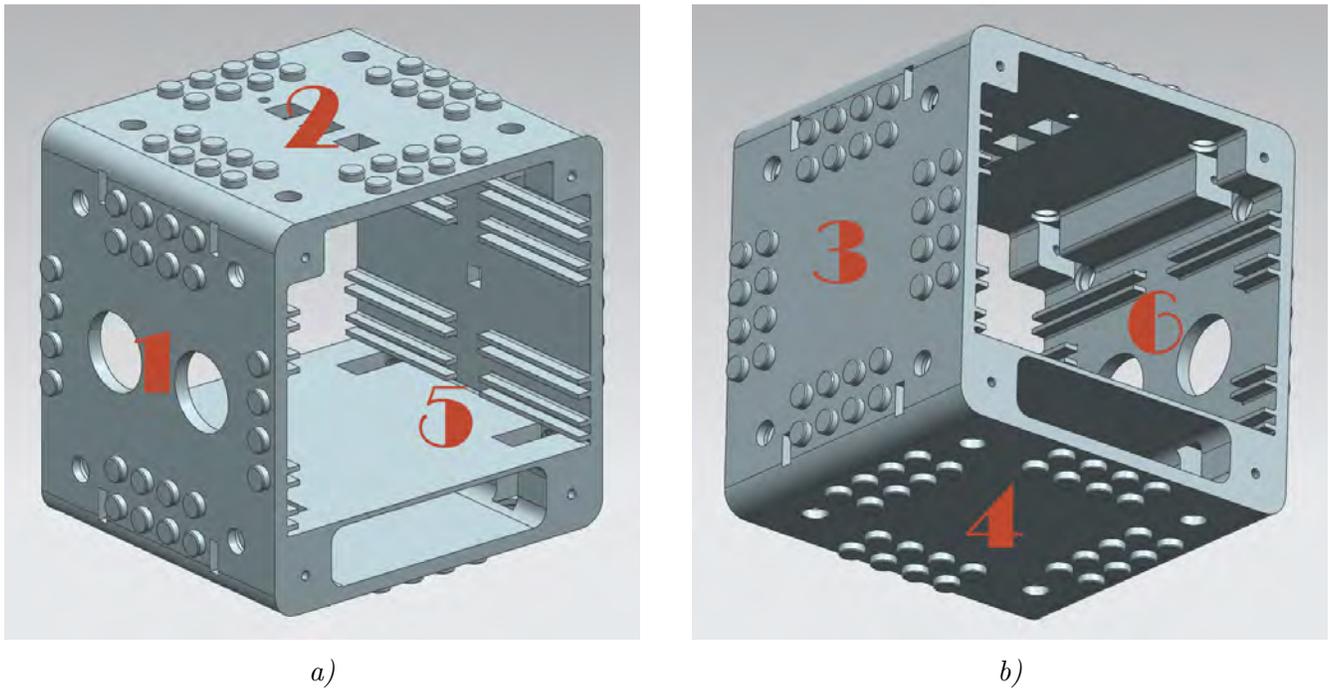


Figura 3.5: Nomenclatura que se le asigna a las caras del cubo por facilidad de identificación, se abarcan 3 caras en a) y las otras 3 en b).

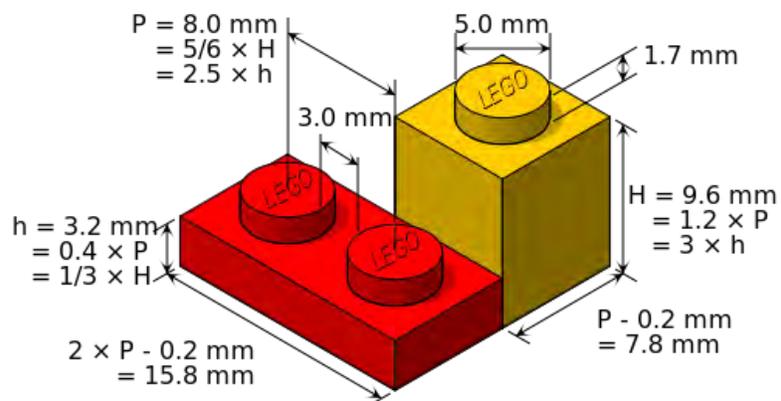


Figura 3.6: Dimensiones de los pines "LEGO" que se ponen de manera estética en las caras del cubo [5].

Los bordes exteriores del cuerpo tienen un redondeado de 5 [mm]. En la zona interior, los bordes del área para los tornillos tienen un redondeado de 1 [mm], mientras que los redondeados de los bordes de los soportes de las placas son de 0.5 [mm] (Figura 3.7).

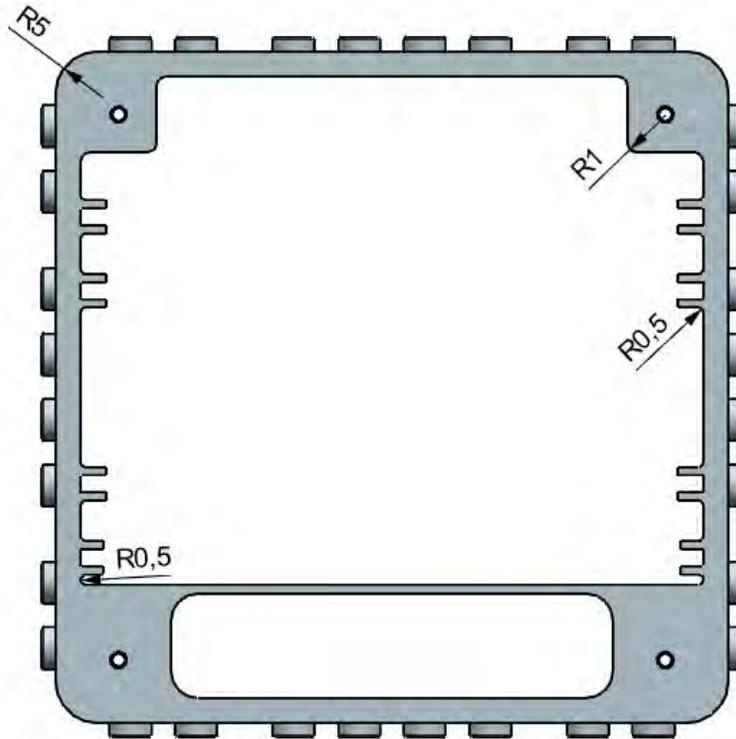


Figura 3.7: Redondeados que se dan tanto de manera interna como externa al robot para eliminar bordes.

A) Soporte para las PCBs

El robot tiene soportes para 4 placas de 70 [mm] \times 74 [mm] (Figura 3.8). Las PCBs FR4 tienen un grosor que varía de los 0.50 a los 2.40 [mm], típicamente. La medida estándar de las placas que se venden al menudeo en tiendas de electrónica tiene un grosor de 1.60 [mm] generalmente. Dado que las placas se fabricaron a mano empleando ese tipo de placas comerciales, considerando una tolerancia para el grosor de la capa de máscara antisoldante, se les da una abertura a los soportes de 2 [mm]. Los soportes para las placas se extienden a lo largo de las caras '1' y '3'. Tienen forma de riel para deslizar las placas y dimensiones de 2 \times 1 [mm] (Figura 3.9-a)).

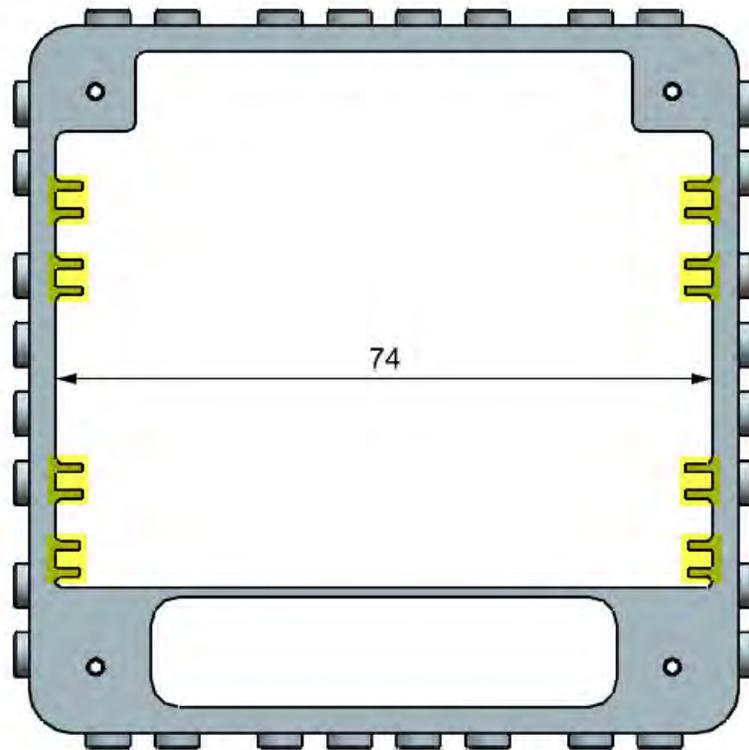


Figura 3.8: Los soportes para las PCBs están resaltados de amarillo, de tal forma que el robot puede cargar hasta 4 placas.

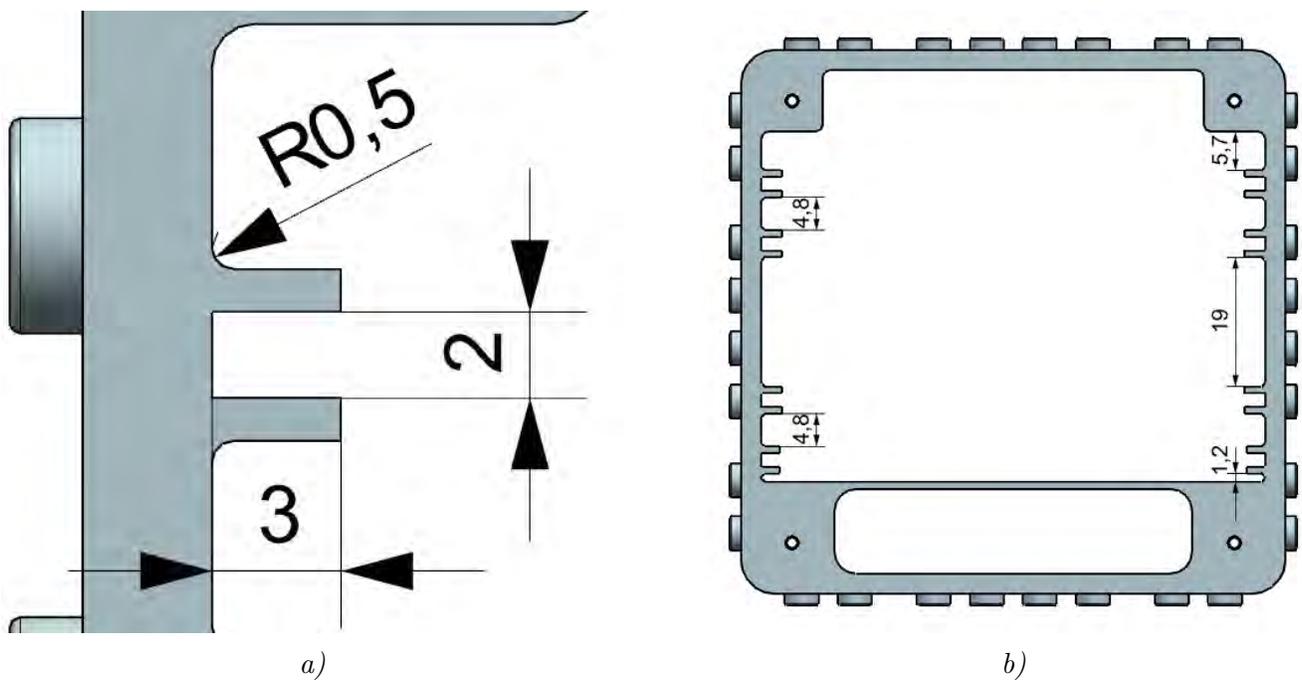


Figura 3.9: a) Se aprecian las dimensiones individuales de cada soporte; b) se aprecia la separación entre todos los soportes dentro del cuerpo.

La separación mínima entre las placas es de 4.80 [mm], espacio suficiente para la mayoría de

los componentes SMT empleados en las PCBs (véase sección 3.2.5 PCBs), sólo algunos capacitores tienen una altura superior, ya sea de 5.00 [mm] o de 6.30 [mm], los cuales se sueldan en dirección a donde existe más espacio. Entre el 2do y el 3er soporte hay una separación de 19 [mm] para dar espacio suficiente a que se ensamblen al centro del robot tanto el sensor ultrasónico como los micro motores (Figura 3.9-b).

Los soportes tienen una interrupción a lo largo de un eje vertical que corre en la parte posterior de las caras ‘1’ y ‘3’, dicha interrupción de 10 [mm] (Figura 3.10-a) sirve para permitir el paso de los cables que conectan diferentes placas y elementos del robot, como los botones o los LEDs. Adicionalmente, los soportes más cercanos a donde se coloca el sensor ultrasónico tienen una interrupción de 50 [mm] para dar espacio suficiente al sensor (Figura 3.10-b).

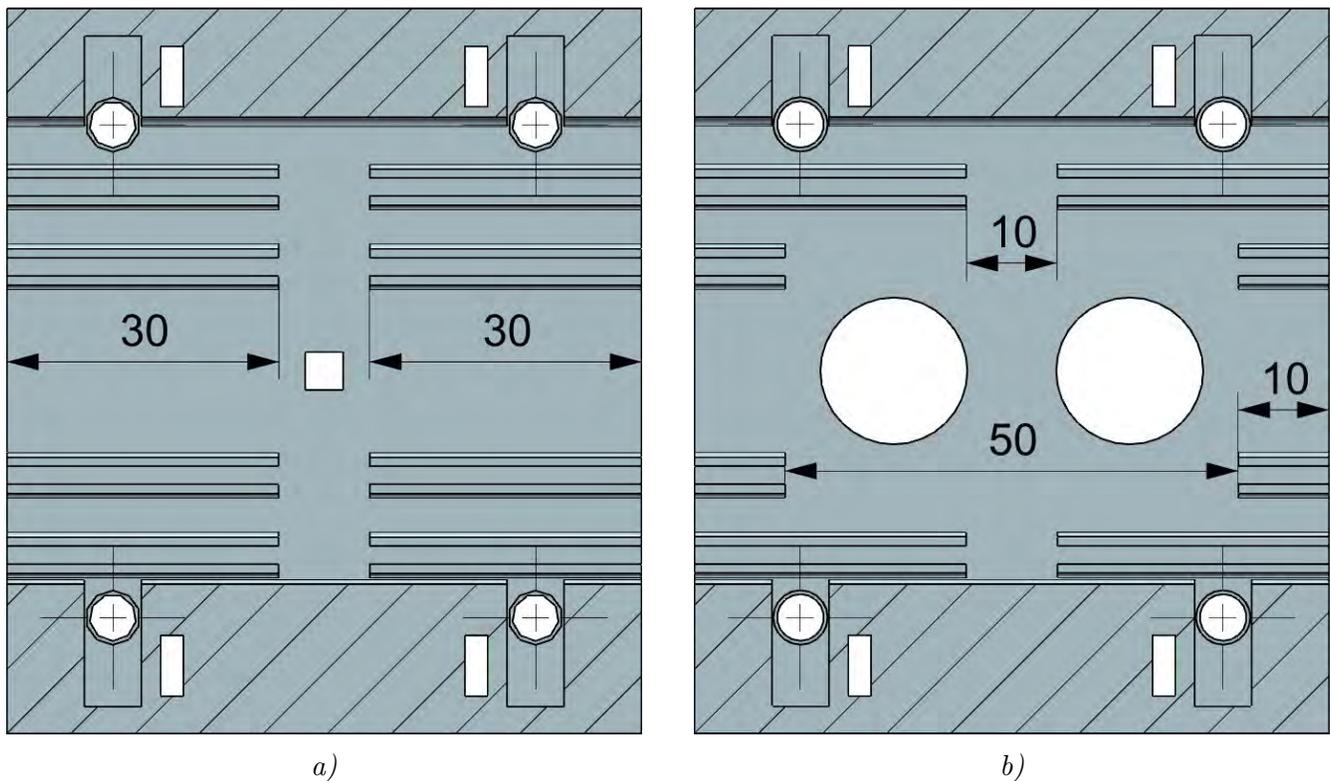


Figura 3.10: a) Se abre un canal para dar espacio para acomodar los cables de la electrónica; b) se abre el espacio suficiente para poder acomodar el sensor ultrasónico.

B) Soportes para sensor y botones

El sensor ultrasónico se ensambla en la cara frontal del cuerpo (‘1’), se sostiene de manera directa en barrenos de diámetro congruente al diámetro del emisor y receptor del sensor (16 [mm]). Las dimensiones nominales del sensor se muestran en la Figura 3.11-a, sin embargo, las dimensiones para montaje en la cara del robot se ajustaron de acuerdo con la medición real de la posición y diámetros del emisor y receptor de varios sensores (Figura 3.11-b).

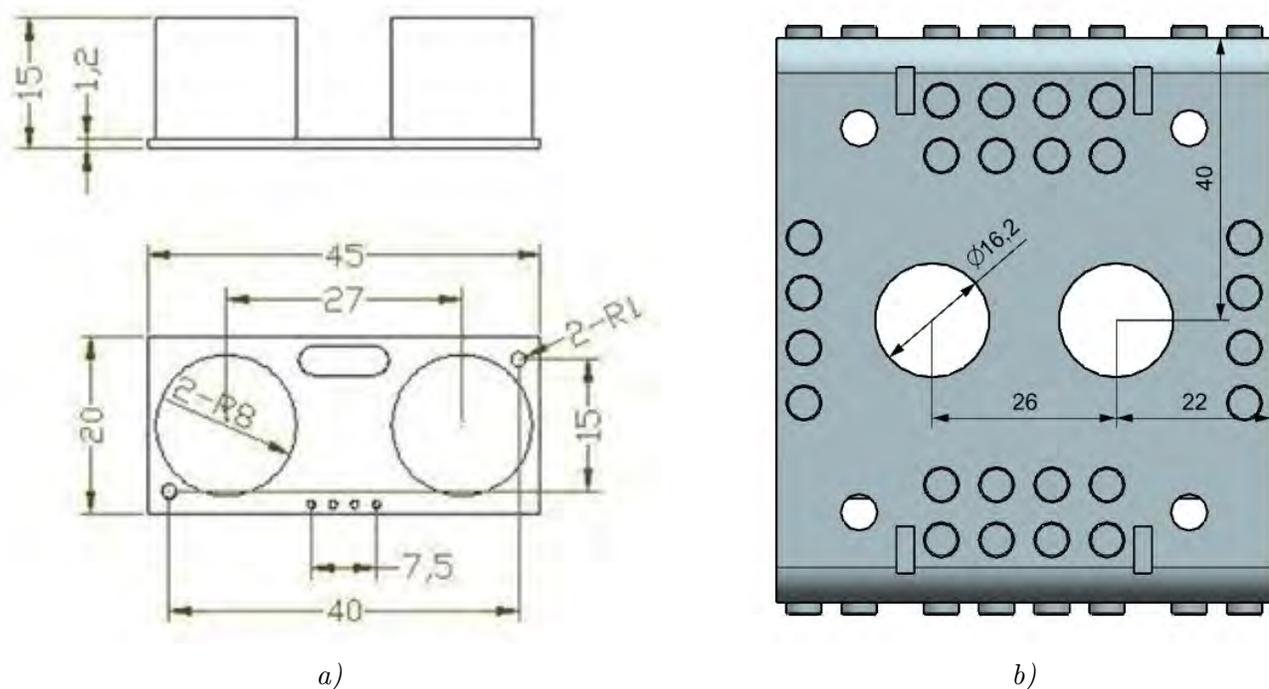


Figura 3.11: a) Esquema de dimensiones que se encuentra en la hoja de especificaciones del sensor ultrasónico [10]; b) dimensiones dadas con base en mediciones para colocar el sensor ultrasónico.

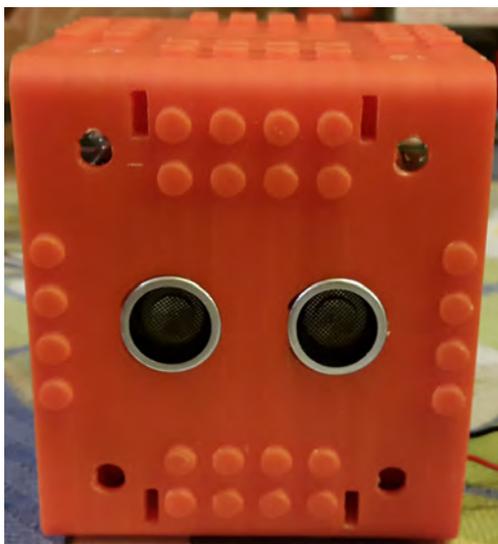


Figura 3.12: Sensor montado en el robot.

Los botones que se ensamblan al robot son el botón de encendido y los botones de propósito general. El botón de encendido (**TL2201EEYA**) es un *switch pushbutton* sub miniatura que se puede acoplar fácilmente al robot sin ocupar mucho espacio dadas sus dimensiones (Figura 3.13-a). Sólo sobresale el vástago del botón, de cabeza de 4 [mm], a través de la cara del robot (se le agregan 0.25 [mm] de tolerancia a la ranura), el resto se adhiere mediante pegamento directamente a la parte anterior de la cara, de esta manera se obtiene un resultado funcional y estético (Figura 3.13-b). El vástago sólo sobresale unos milímetros del robot, al ser presionado sobresale aproximadamente 1 [mm].

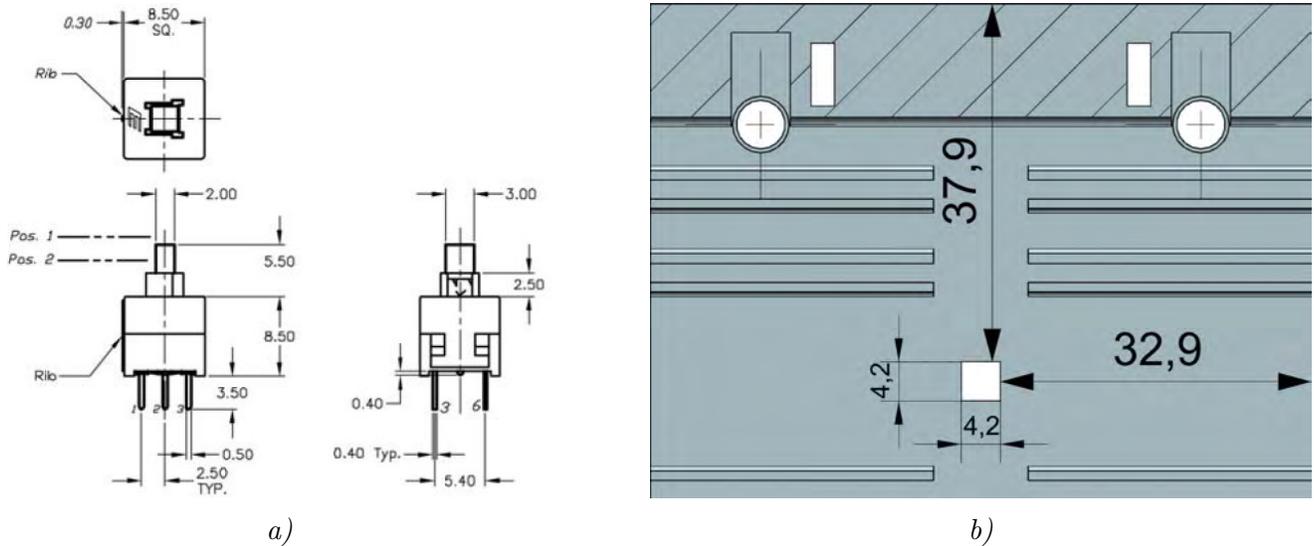


Figura 3.13: a) Esquema de dimensiones del botón que se encuentra en la hoja de especificaciones [18]; b) dimensiones y colocación de la ranura para el botón.

Los botones de propósito general se colocan en la cara superior del robot ('2'), tales botones se pegan de manera directa al reverso de la cara. Es una placa con 4 botones, donde uno sirve como reset general y es distinto a los demás, este botón sólo es accesible introduciendo algún objeto de menor diámetro que el barreno de 2.40 [mm]. Los otros 3 botones son todos del tipo usual de *pushbutton* de 6 [mm] (se agregan 0.25 [mm] de tolerancia a las ranuras) y se montan al ras de la cara del robot. En la Figura 3.14 se aprecian las dimensiones dadas para la botonera.

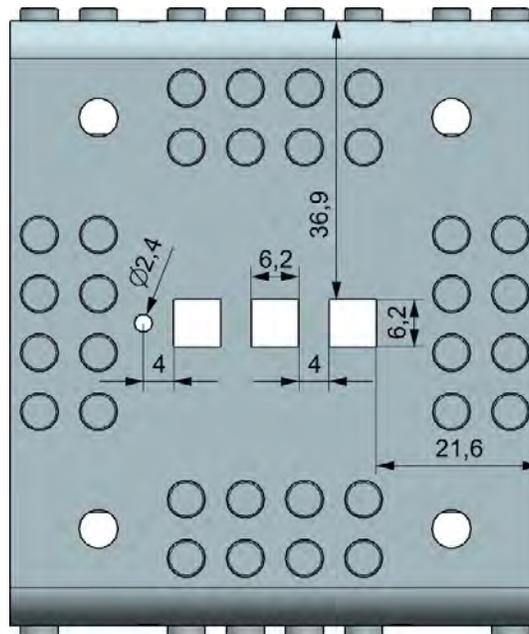


Figura 3.14: Dimensiones para la botonera. La botonera tiene 3 botones *pushbutton* convencionales y uno más pequeño que funciona como reset.

C) Soporte para las baterías

Las baterías se montan en la base del robot (cara '4'). Las baterías tienen dimensiones nominales de $67.50 \text{ [mm]} \times 51.90 \text{ [mm]} \times 6.10 \text{ [mm]}$. Se apilan 2 baterías una encima de la otra y se deslizan dentro de un compartimiento con forma rectangular con esquinas redondeadas (Figura 3.15). El compartimiento es accesible desde las caras laterales ('5' y '6'), cuyas dimensiones son $51.90 + 0.60 \times 12.20 + 0.20 \text{ [mm]}$, con redondeados de 3.10 [mm] .

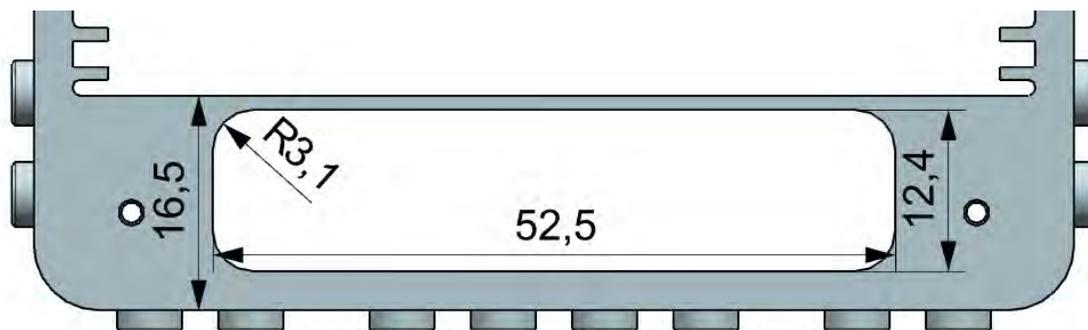


Figura 3.15: Dimensiones del soporte donde se colocan las baterías del robot.

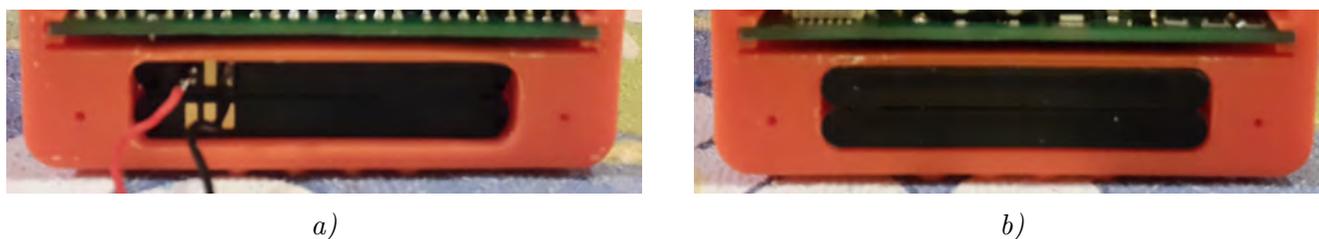


Figura 3.16: En a) se aprecian las baterías montadas desde un costado y en b) se aprecia el otro costado, de tal forma que se puede acceder a las baterías por ambos costados laterales del cubo.

Dado el extra de espacio longitudinal, los cables que se conectan a las terminales se pueden extraer de cualquiera de las dos caras laterales (Figura 3.17). El resto de material alrededor de las baterías y hasta una altura de 16.50 [mm] se rellena de material para soportar las baterías.

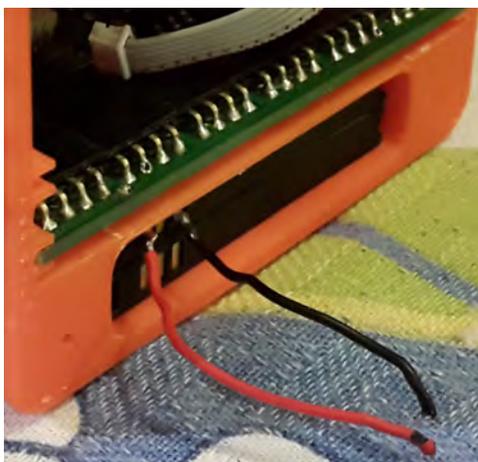


Figura 3.17: Existen unos milímetros adicionales que son suficientes para hacer pasar los cables de las baterías hacia las placas.

D) Soporte para los LEDs

El robot cuenta con 4 LEDs por cara. Cada LED es soldado en placas de 1×0.50 [mm] que se ensamblan en la parte posterior de la cara, de tal forma que sólo sobresale una parte del encapsulado de los LEDs. El encapsulado de los LEDs tiene un diámetro de 5 [mm] y un anillo en su base de 5.80 [mm].

La posición de los LEDs en las caras del cubo es simétrica con respecto a dos ejes, misma que se aprecia en la Figura 3.18. El barreno para los LEDs tiene una tolerancia de 0.15 [mm] para el encapsulado y de 0.20 [mm] para el anillo.

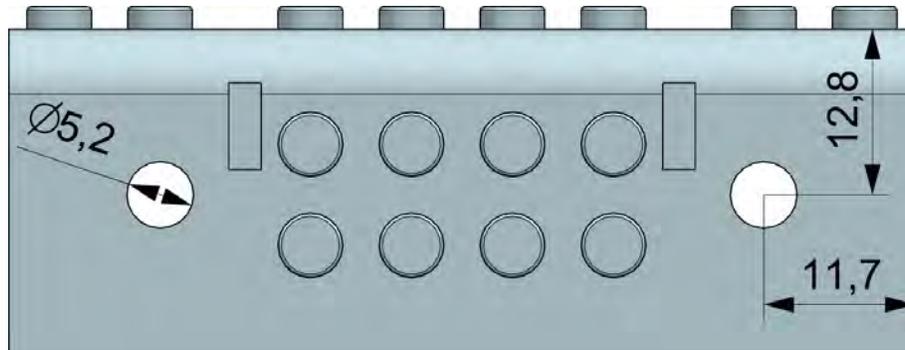


Figura 3.18: Posición en la que se colocan los LEDs RGB, la posición es simétrica con respecto al eje horizontal de la cara que pasa por el centro.

Las dimensiones de la profundidad del soporte para los LEDs, en general, se aprecia en la Figura 3.19-a. Algunos de los LEDs tienen la facilidad de ser colocados directamente dado que no hay obstrucciones, sin embargo, 8 LEDs se colocan en la misma zona donde se encuentra el soporte para las baterías. Debido a ello, se hace un pequeño hueco para que los LEDs puedan ser colocados (Figuras 3.19 y 3.20).

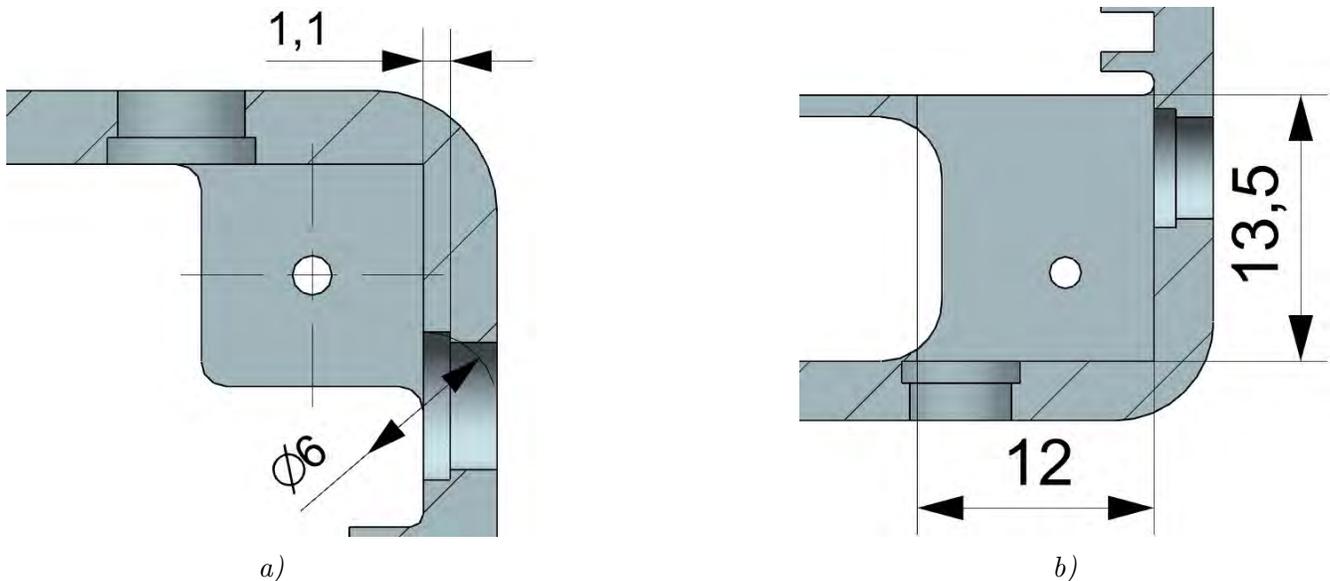


Figura 3.19: a) Dimensiones del barreno donde se coloca el LED; b) dimensiones del hueco que permite colocar los LEDs donde existe el soporte para la batería.

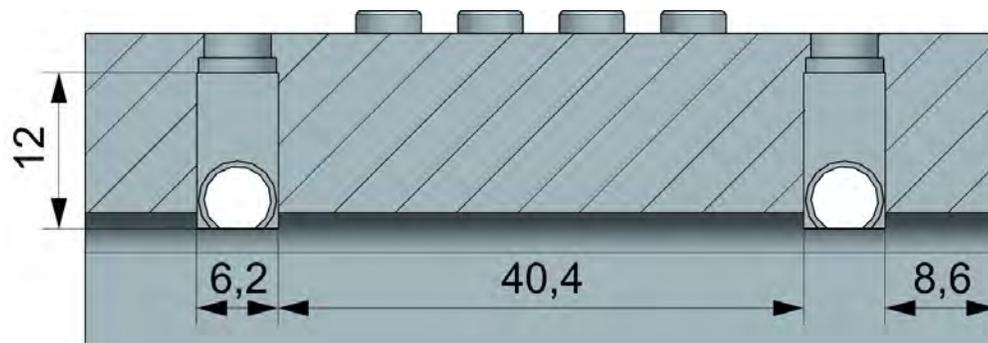


Figura 3.20: Posición del hueco que permite colocar los LEDs donde existe el soporte para la batería.

E) Soporte para las tapas

Las tapas del robot se ensamblan mediante el uso de tornillo y tuerca. Se opta por esta solución para poder abrir y cerrar el robot sin lastimar el polímero, a diferencia de usar pijas.

La tuerca se encuentra embebida muy cerca de los pines tipo *LEGO* en las caras '1' y '3'. Las tuercas son del tipo #4-40, los tornillos #4-40 tienen una longitud de 1". Para sostener la tuerca, las esquinas del robot cuentan con un soporte de material extra en la cara posterior de 9×9 [mm], al centro de este soporte se encuentra el barreno para el tornillo (Figuras 3.21-a y 3.21-b).

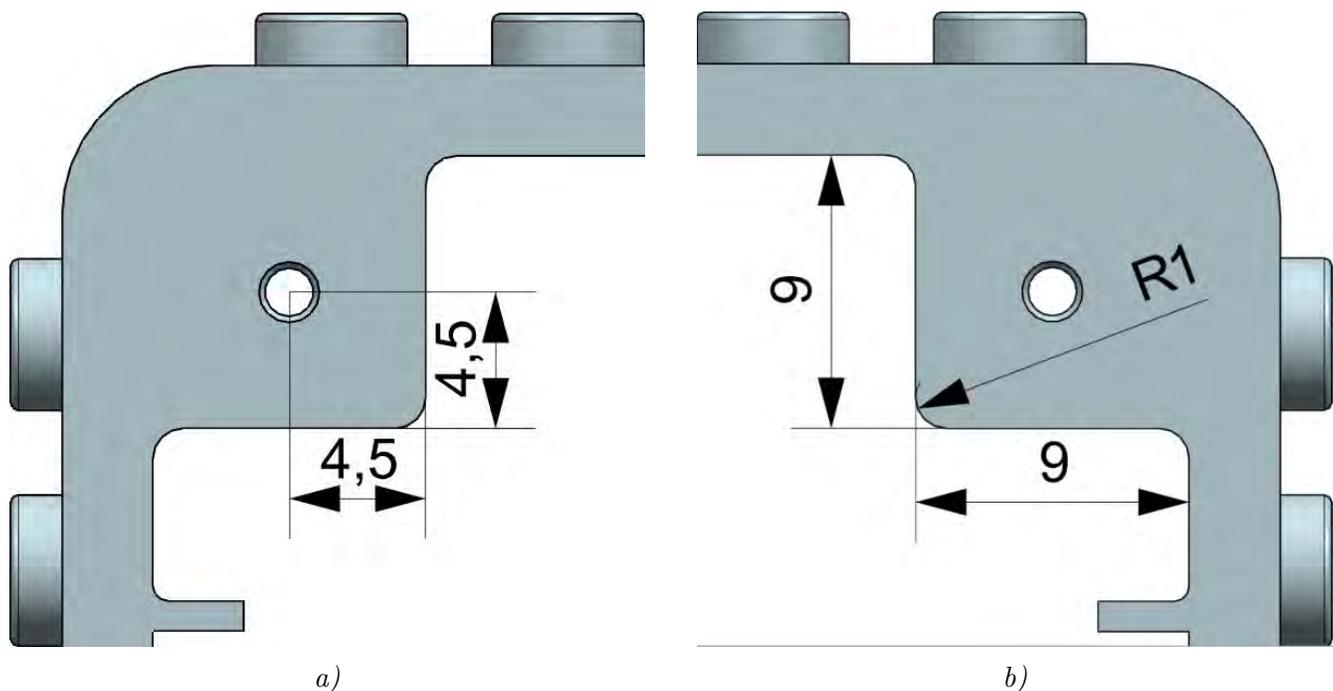


Figura 3.21: a) Se aprecia la posición del barreno para que entre el tornillo que sujeta la tapa; b) se aprecian las dimensiones del material que sostiene la tuerca que hace juego con el tornillo.

Se hace un hueco en la cara exterior de $6.70 \times 2.50 \times 11.20$ [mm] (Figuras 3.22) donde se inserta la tuerca.

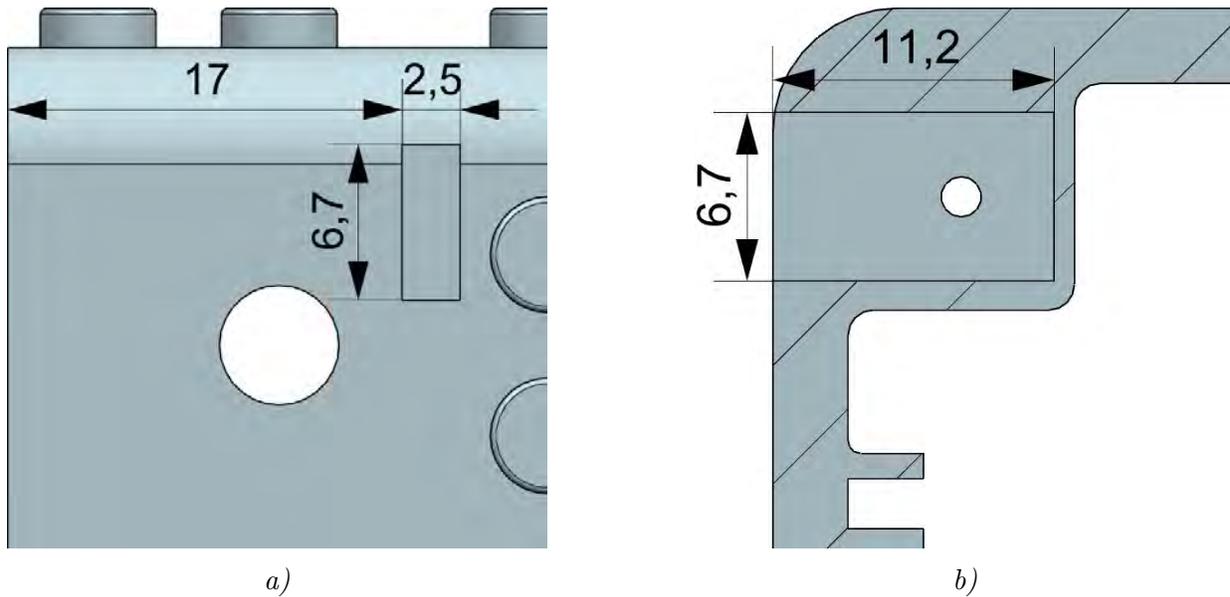


Figura 3.22: a) Posición con respecto al extremo del cuerpo donde se hace el hueco; b) profundidad del hueco donde se inserta la tuerca.

3.1.2 Tapas

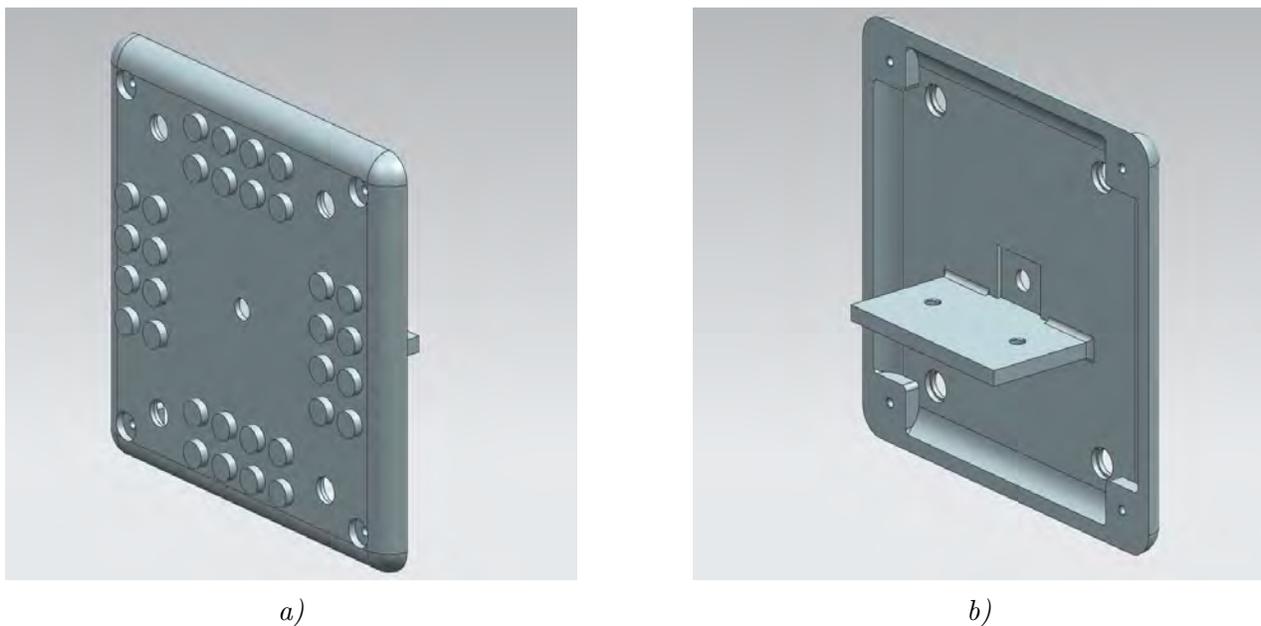


Figura 3.23: Tapas del robot, que se aprecia en una vista trimétrica, a) de frente, b) por detrás.

Las tapas abarcan el tamaño adicional para que el robot, al ser ensamblado, tenga las dimensiones finales de $80 \times 80 \times 80$ [mm], es decir, tienen dimensiones de $80 \times 80 \times 5$ [mm] cada una (Figuras 3.24). Al igual que el cuerpo central, las paredes tienen un grosor de 2 [mm] y se

encuentran los pines tipo *LEGO*. Las tapas están redondeadas en los bordes, el mismo redondeado de radio de 5 [mm] que el resto de las paredes del cuerpo central. La posición para los LEDs es similar que para las caras del cuerpo central y los soportes para su ensamble con el cuerpo central es una continuación del material extra.

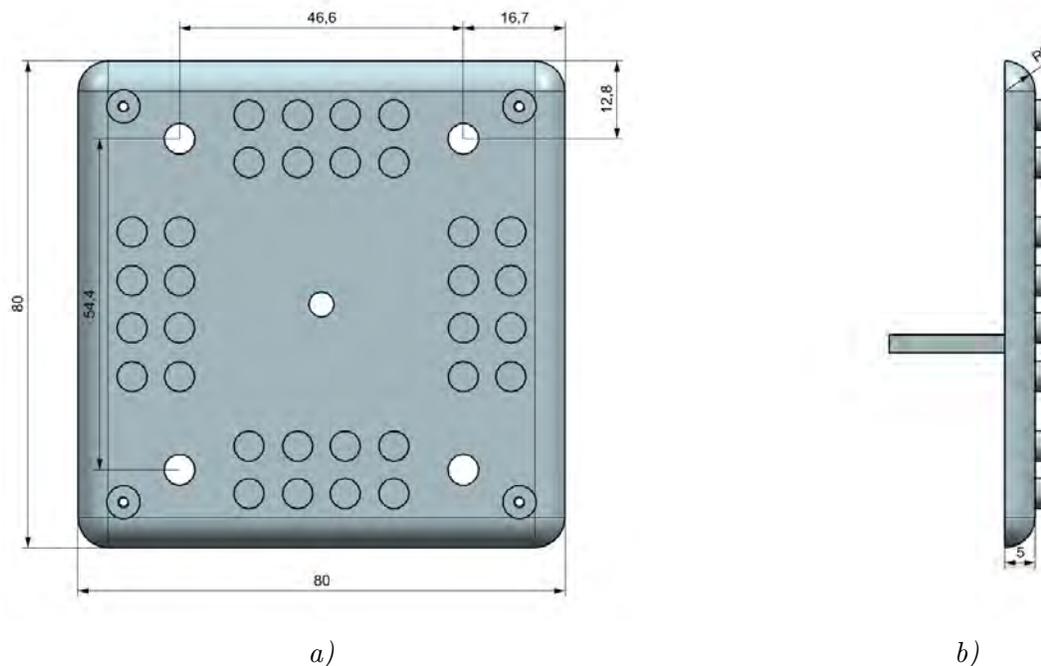


Figura 3.24: a) Se aprecian las dimensiones generales de la tapa, así como la posición de los LEDs; b) se aprecia la tapa de perfil para observar el redondeado.

Al centro del reverso de la tapa se encuentra el soporte para la instalación del motor. El motor se sujeta a la tapa mediante el uso de un *bracket*.

A) Soportes para los tornillos

Al igual que el cuerpo central, las tapas tienen el barreno de 1.60 [mm] de diámetro donde se introduce el tornillo. Dado que la tuerca se encuentra embebida en el cuerpo central, en las tapas se hace un barreno para soportar la cabeza de los tornillos y así ejercer la fuerza para mantener unidas ambas partes.

En las Figuras 3.25 se aprecian las dimensiones para soportar los tornillos. La posición es la misma que para el cuerpo central y se mantiene el material de soporte a los alrededores del barreno. Se da una profundidad de 1.5 [mm] para la cabeza del tornillo, con un diámetro de 5.5 [mm] (Figura 3.26).

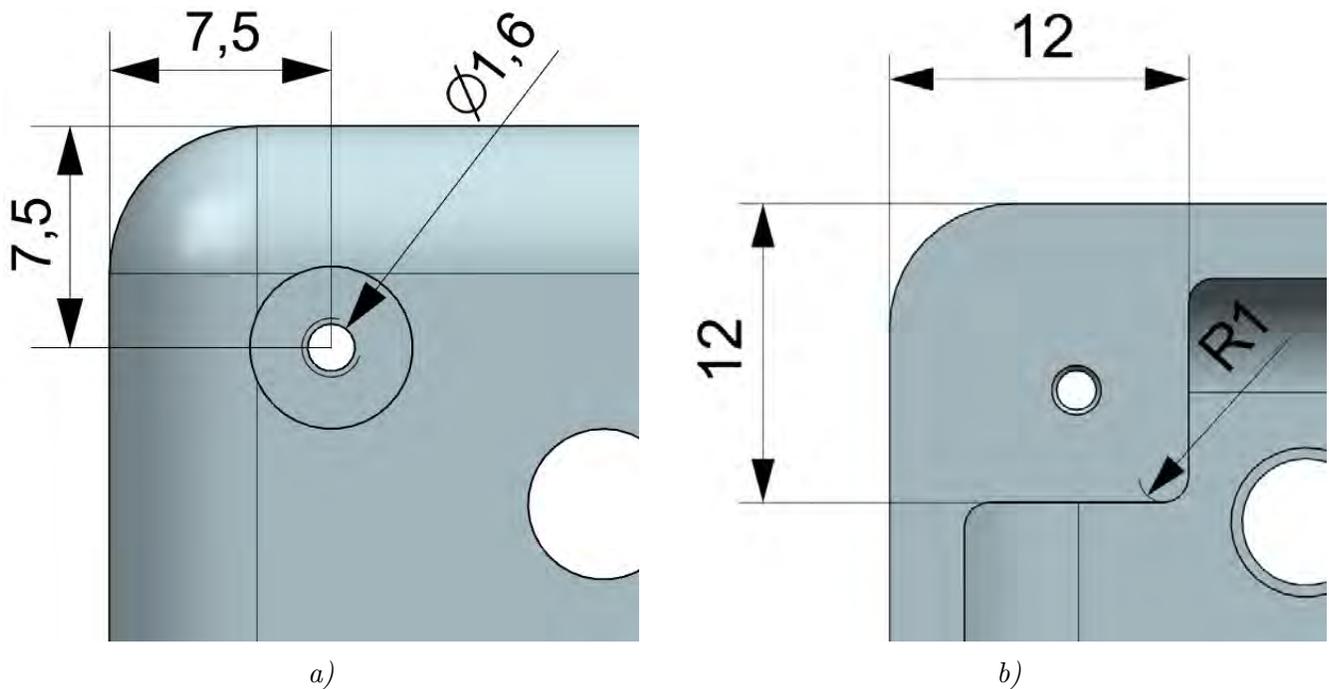


Figura 3.25: a) Posición del barreno para el tornillo; b) dimensiones del material de soporte para el tornillo, coinciden con las dimensiones en el cuerpo central.

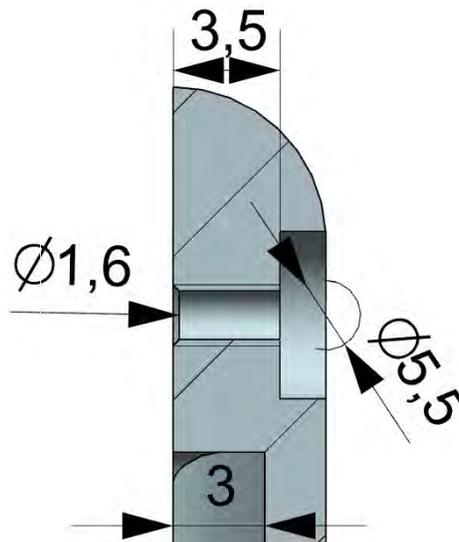


Figura 3.26: Profundidad del barreno para el tornillo; se aprecia la profundidad a la que puede entrar la cabeza del tornillo.

B) Soporte para el motor

El motor se monta de tal forma que la flecha es perpendicular a la cara anterior de la tapa y se encuentra en el centro, donde la flecha sobresale de un barreno de 4.15 [mm]. El motor se coloca en una plataforma y se abraza mediante un *bracket* personalizado. Con esta combinación el motor se sujeta perfectamente a la tapa.

La plataforma (Figuras 3.27 y 3.28) tiene dimensiones de $22 \times 40 \times 3$ [mm], todos los bordes

tienen un redondeado sobre la cara posterior de la tapa de 1 [mm]. La plataforma tiene 2 barrenos para sujetar el *bracket* de 2.50 [mm] de diámetro. El *bracket* se fija a la plataforma mediante 2 tornillos #2-56 \times 1/4".

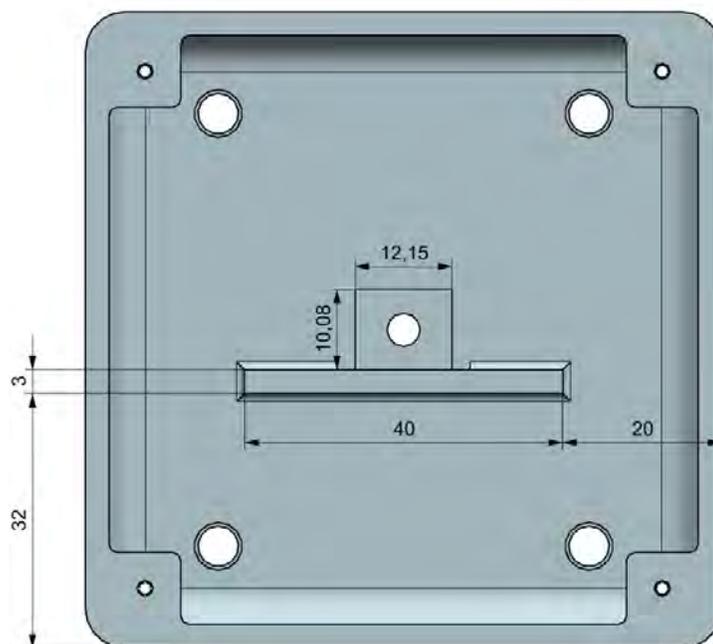


Figura 3.27: Dimensiones y posición para la plataforma que soporta al motor.

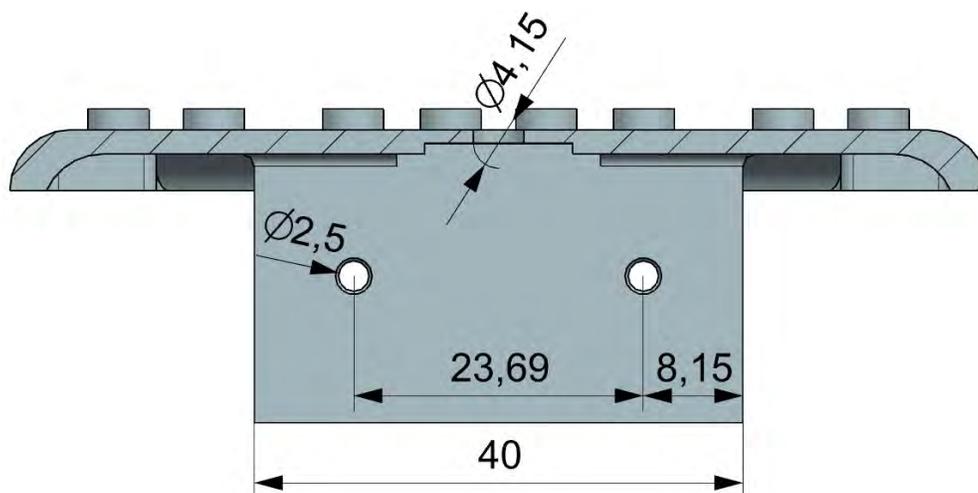


Figura 3.28: Dimensiones y posición de los barrenos donde se sujetan los tornillos y *bracket*, así como la eliminación de material para el ensamble del *bracket* y la última parte de la reducción del motor.

Al centro de la cara posterior, se le subtrae un rectángulo de 12.15 \times 10.075 \times 0.9 [mm], correspondiente al final de la reducción del motor, con la finalidad de que sobresalga la mayor cantidad de la flecha de la reducción. A los alrededores del rectángulo se recortan 2.25 [mm] del redondeado de la plataforma para permitir el ensamble del *bracket*.

3.1.3 Brackets

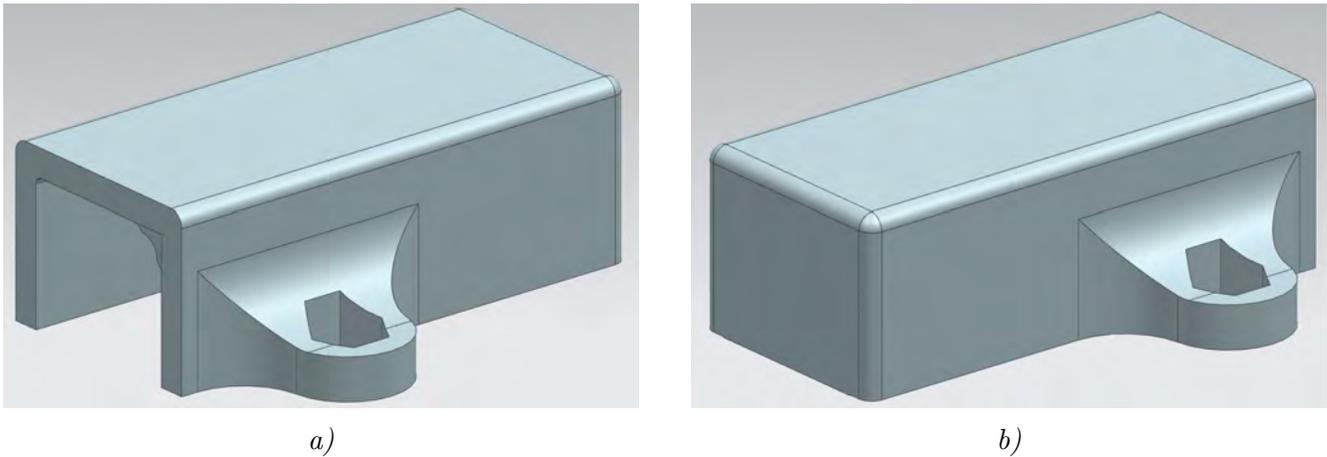


Figura 3.29: Se aprecia la forma general en vista trimétrica de frente, a), y por detrás, b), del bracket.

Los *brackets* sirven para sostener los motores en las tapas del robot. Tienen dimensiones generales de $16.15 \times 35.30 \times 12.15$ [mm] (Figuras 3.30 y 3.31) más 8 [mm] del espacio para las tuercas. Se sostienen a la plataforma de la tapa mediante el uso de tornillo y tuerca de tipo M3. El *bracket* está diseñado de tal forma que la última “hoja” de la caja de reducción se monta dentro de la tapa del robot, por lo que el *bracket* sostiene prácticamente todo el motor, exceptuando esa última “hoja”. Los bordes internos de las paredes tienen un radio de 0.50 [mm].

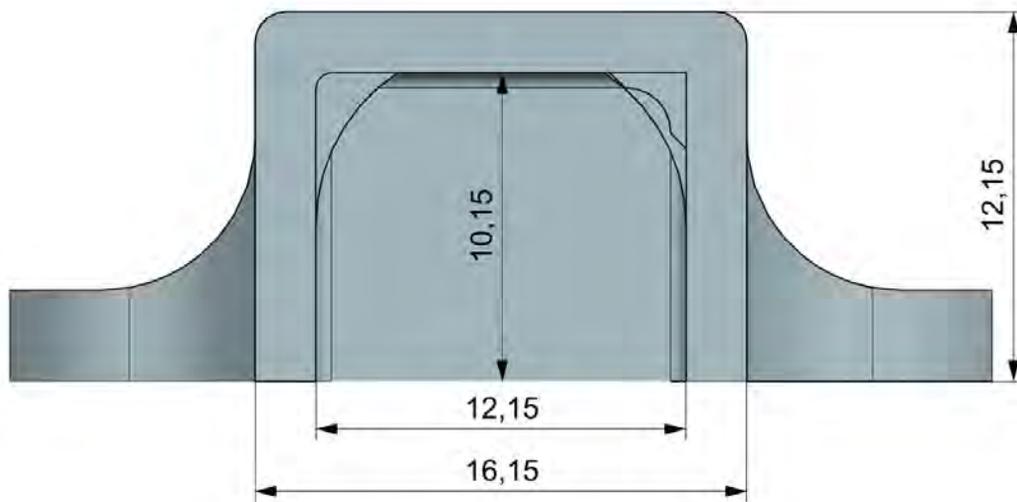


Figura 3.30: Dimensiones de longitud y altura del bracket sin tomar en cuenta los soportes para las tuercas.

Los *brackets* encapsulan al motor con un espacio interno de 12.15×10.15 [mm], con la forma redondeada del motor y la forma de la caja de reducción a lo largo del *bracket* (véase subsección 3.1.4 Micro motores y llantas). Los arcos que sostienen el cuerpo del motor son de 5.00×2.68 [mm] con un radio de 6.00 [mm], propio del motor, con una longitud de 14 [mm] y se encuentran a ambos lados del robot. Los arcos que sostienen la caja de reducción contemplan el espacio de las “hojas” de la caja de reducción y el soporte de los cilindros, dando como resultado una extensión de material añadido de 2.50×2.50 [mm], un radio de 1.45 [mm] y un chanfle de 45°

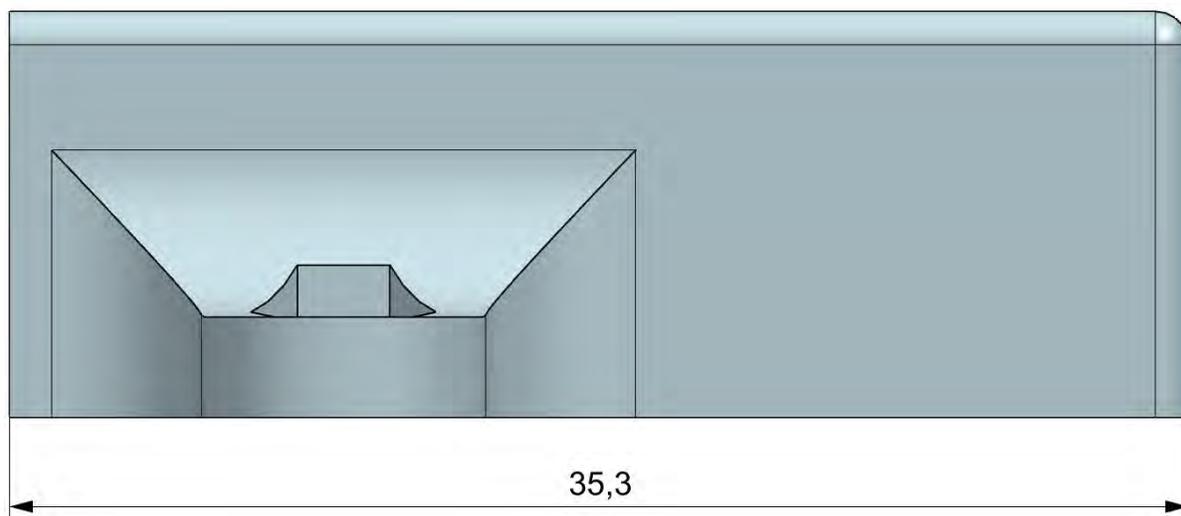


Figura 3.31: *Dimensión de profundidad del bracket.*

para evitar interferencia al momento de montar el *bracket* encima del motor, se encuentran sólo de un lado, que es donde se puede sostener la caja de reducción. Las distancias a las que corren todos los arcos se aprecian en la Figura 3.32, mientras que las dimensiones de los arcos se aprecian en las Figuras 3.33.

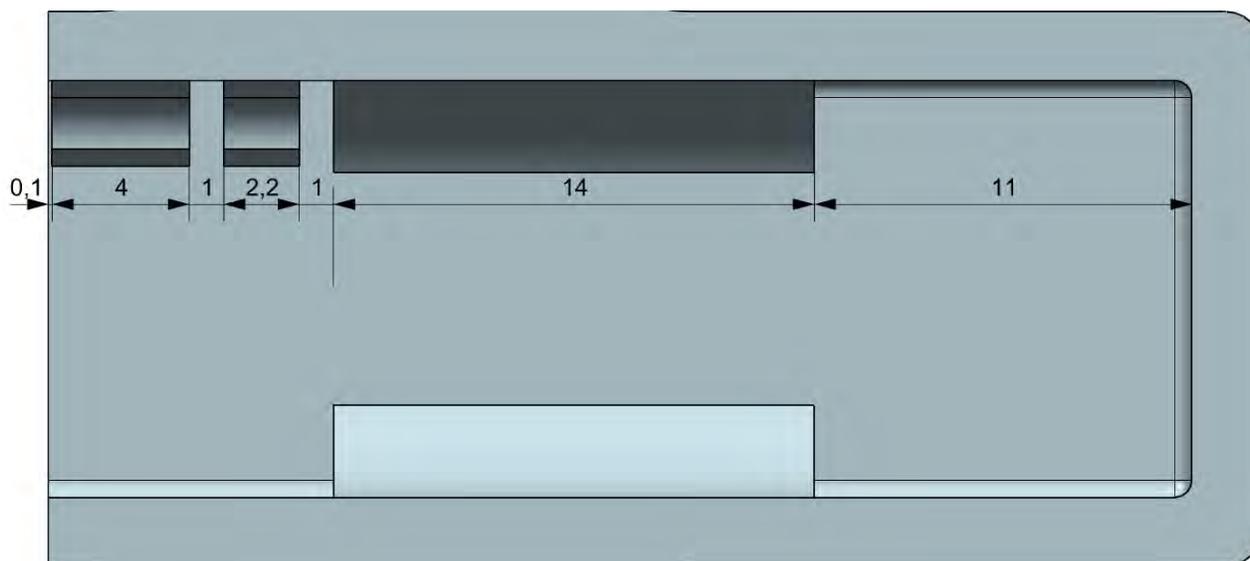


Figura 3.32: *Distancias al extremo del **bracket** donde se encuentran los arcos que sostienen al micro motor.*

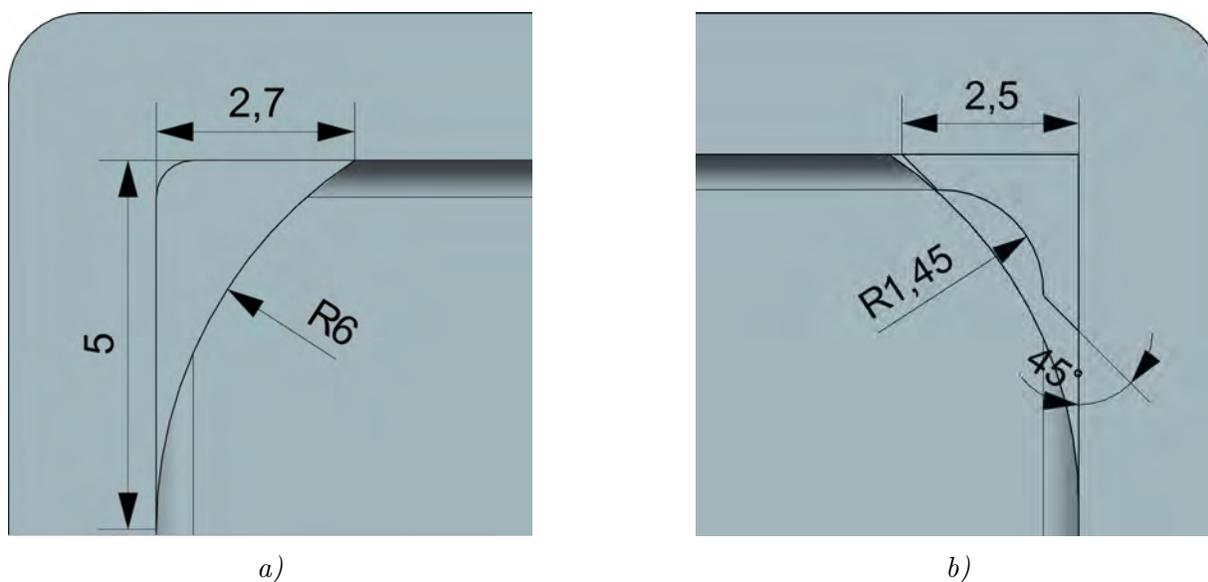


Figura 3.33: a) Dimensiones generales de los arcos que sostienen el cuerpo del motor; b) dimensiones generales de los arcos que sostienen la caja de reducción del motor.

La longitud del *bracket* supera el ensamble del motor con el encóder; termina con la pared definida para prevenir que los encoders se interfiran entre ellos al estar de frente el uno con el otro.

El espacio para las tuercas comprende dos extensiones a los costados con diferentes redondeados de 5.12 [mm] y 4.50 [mm], así como un medio círculo de 4.27 [mm] de diámetro. El receptáculo de la tuerca tiene forma hexagonal regular, cuyo lado es de 2.77 [mm]. La longitud del soporte es de 4.13 [mm]. Las dimensiones generales se aprecian en las Figuras 3.34.

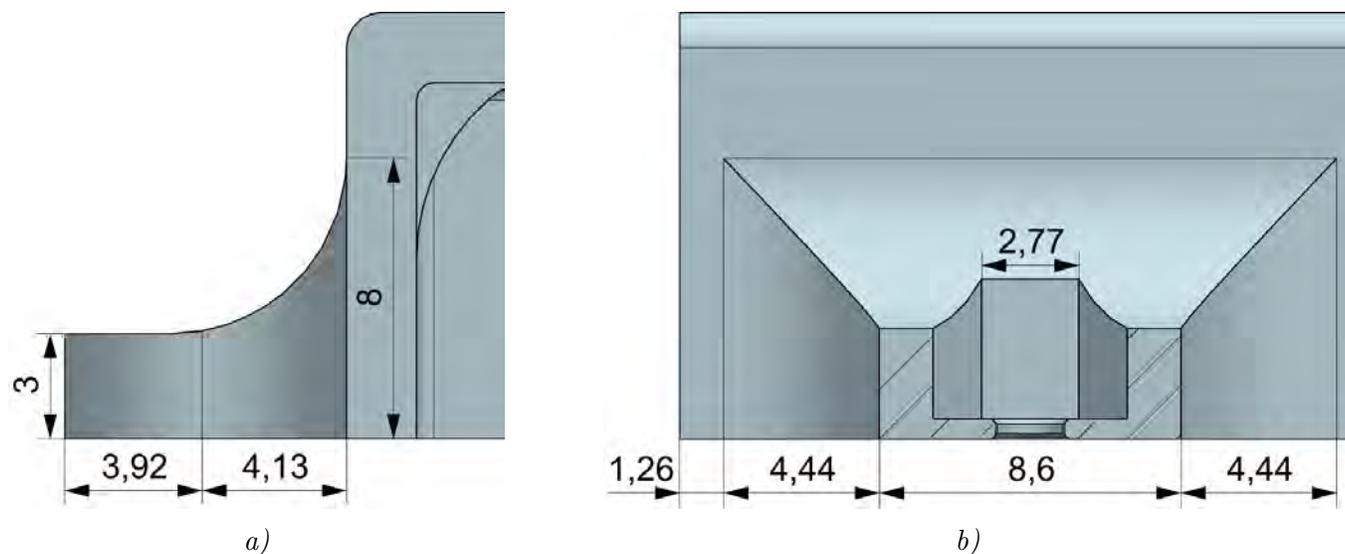


Figura 3.34: a) Dimensiones generales del soporte para las tuercas; b) se genera una sección para apreciar el material sobre el que descansa la tuerca.

3.1.4 Micro motores y llantas

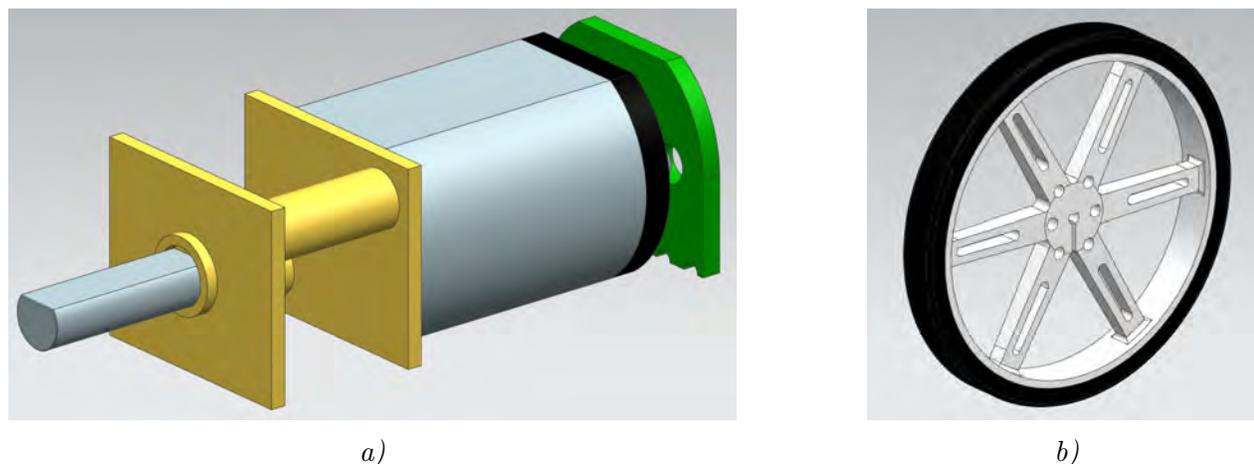


Figura 3.35: a) CAD del micro motor Pololu; b) CAD de la llanta Pololu.

Los micro motores de *Pololu* se componen de dos partes; el motor de corriente directa y la caja de reducción, en este caso, con una relación de 297.92:1. El motor es de escobillas regulares de carbón, para el cual *Pololu* liberó un esquema con las dimensiones, mismas que se aprecian en la Figura 3.36.

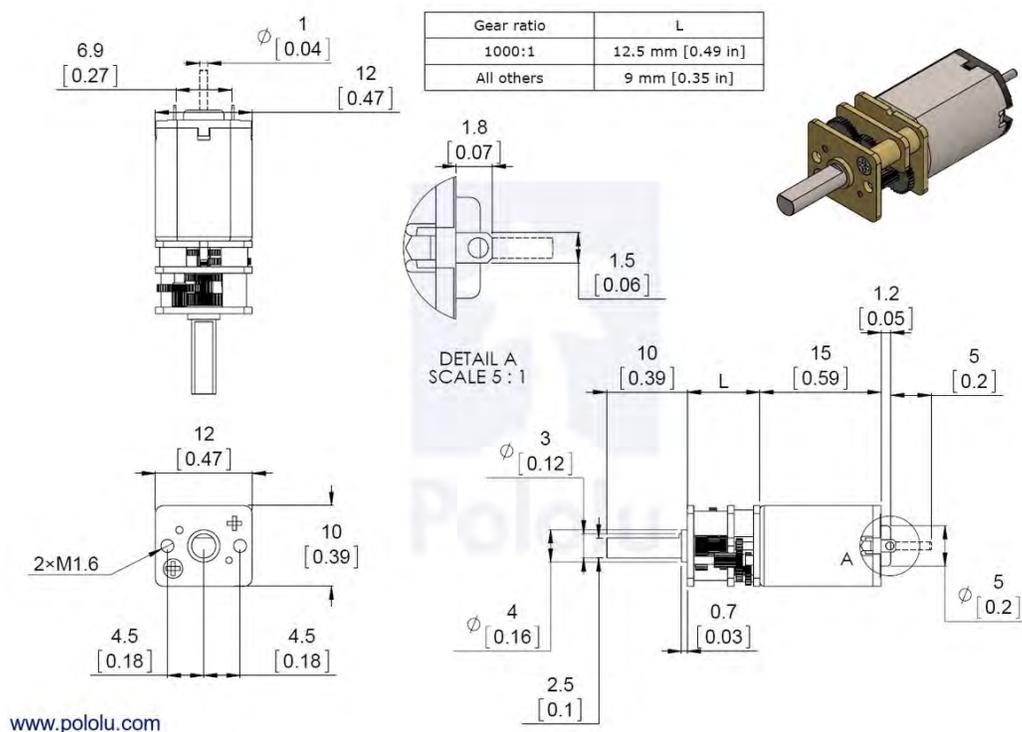


Figura 3.36: Dimensiones del micro motor Pololu [64].

Las llantas están diseñadas para ser ensambladas directamente sobre la flecha de 3 [mm] de salida de la reducción del motor. Las llantas tienen un aro de silicona de 90 [mm] de diámetro. Se le pueden montar bases universales para ser acoplados con otros motores, pero son innecesarias

para el caso de la reducción de estos micro motores. Las dimensiones de las llantas se muestran en un esquema de *Pololu* que se aprecian en la Figura 3.37. El ensamble de un micro motor con la llanta se puede observar en la Figura 3.38.

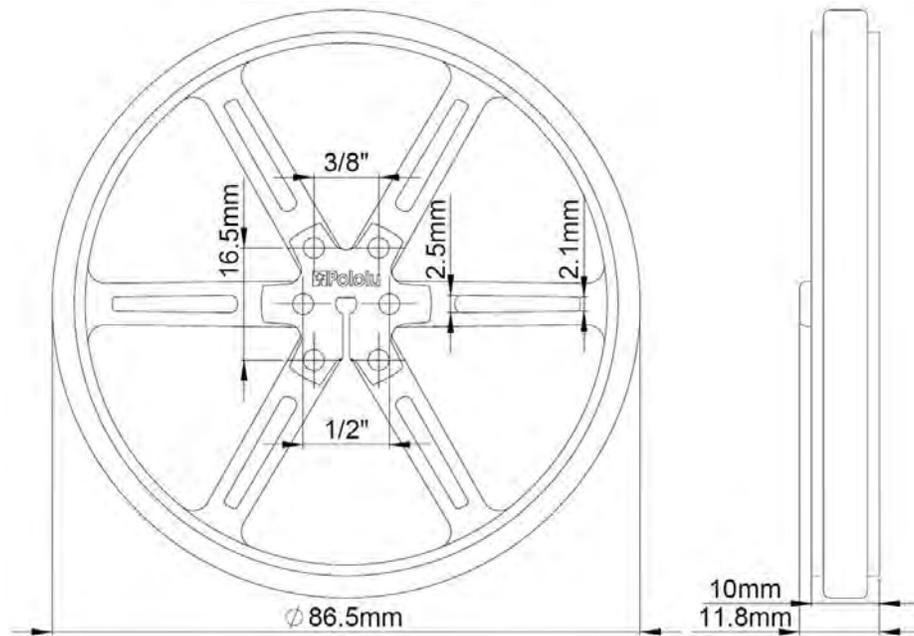


Figura 3.37: Dimensiones de las llantas Pololu [66]. Existen varios diámetros para las llantas, el seleccionado es de 90 [mm].



Figura 3.38: Ejemplo de ensamble entre un micro motor con una llanta de Pololu [63].

3.2 Electrónica

En esta sección se detallan los diferentes elementos que integran al sistema electrónico, pertenecientes a los sensores, los actuadores y la etapa de regulación de energía. También se incluye la descripción y funcionamiento de los microcontroladores y del wifi. Al final se muestra el diseño de las placas electrónicas (PCB, por sus siglas en inglés) del robot en donde se sueldan y conectan sus componentes electrónicos. Todos los circuitos integrados que se emplean son tecnología de montaje superficial (SMT, por sus siglas en inglés); también se integran elementos de tecnología de agujeros pasantes (THT, por sus siglas en inglés) como botones y algunos conectores.

3.2.1 Sensores

El robot cuenta con 4 sensores; 3 sensores propioceptivos que le indican su estado interno y 1 sensor exteroceptivo que le brinda información de su entorno.

Los 3 sensores propioceptivos son:

- Una IMU (inertial measurement unit, por sus siglas en inglés) de 6 ejes que se usa para medir la orientación del robot y la aceleración en los ejes del robot.
- Un par de encoders de cuadratura para calcular la velocidad de la flecha principal de ambos motores.
- Un arreglo de comparadores para indicar el nivel mínimo aceptable de carga de la batería.

El sensor exteroceptivo es un sensor ultrasónico utilizado para medir la distancia a la que se encuentren obstáculos al frente del robot.

A) Encoders

Descripción general

El par de encoders ópticos seleccionados son encoders de cuadratura que se sueldan en las terminales de la parte posterior del motor de flecha extendida. Consisten en 2 placas y dos juegos de estrellas de 3 y 5 dientes. Existen tanto las versiones de 3.3 [V] y de 5 [V], de las que se seleccionó la versión de 5 [V], acorde con el suministro de voltaje de los microcontroladores. Dependiendo el tipo de decodificador, se pueden tener resoluciones de 3 a 12 flancos por revolución para el caso de la estrella de 3 dientes, y de 5 a 20 flancos por revolución con la estrella de 5 dientes. En la Figura 3.39-a se pueden apreciar los encoders y las estrellas.

Los encoders se sueldan directamente en la parte trasera del motor, sobresaliendo así la flecha extendida al centro de las tarjetas, que es donde se monta la estrella. Una vez soldados, se accede a las terminales del motor a través de los pines de los encoders, *M1* y *M2*. En la Figura 3.39-b se aprecian todos los pines de los encoders.

En la Figura 3.40 se aprecia el esquema electrónico de los encoders. Para el caso de los encoders empleados de 5 [V] existe sólo una resistencia que regula la corriente y voltaje de los emisores infrarrojos, mismos que se encuentran en serie. Los encoders de 5 [V] consumen nominalmente 12 [mA] en funcionamiento.

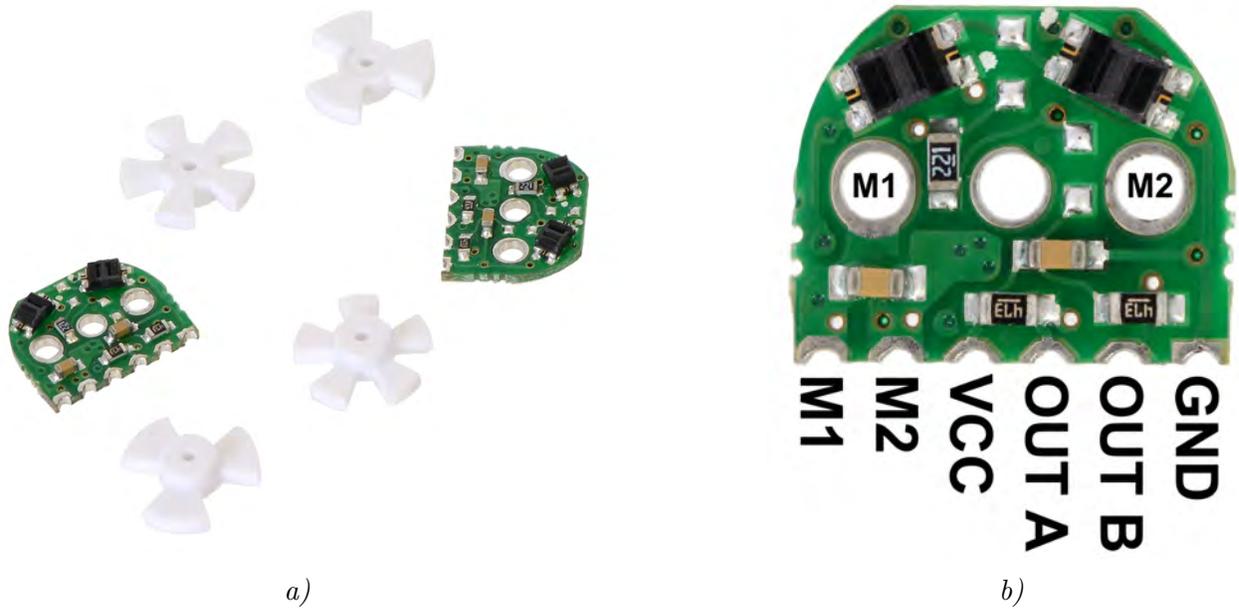
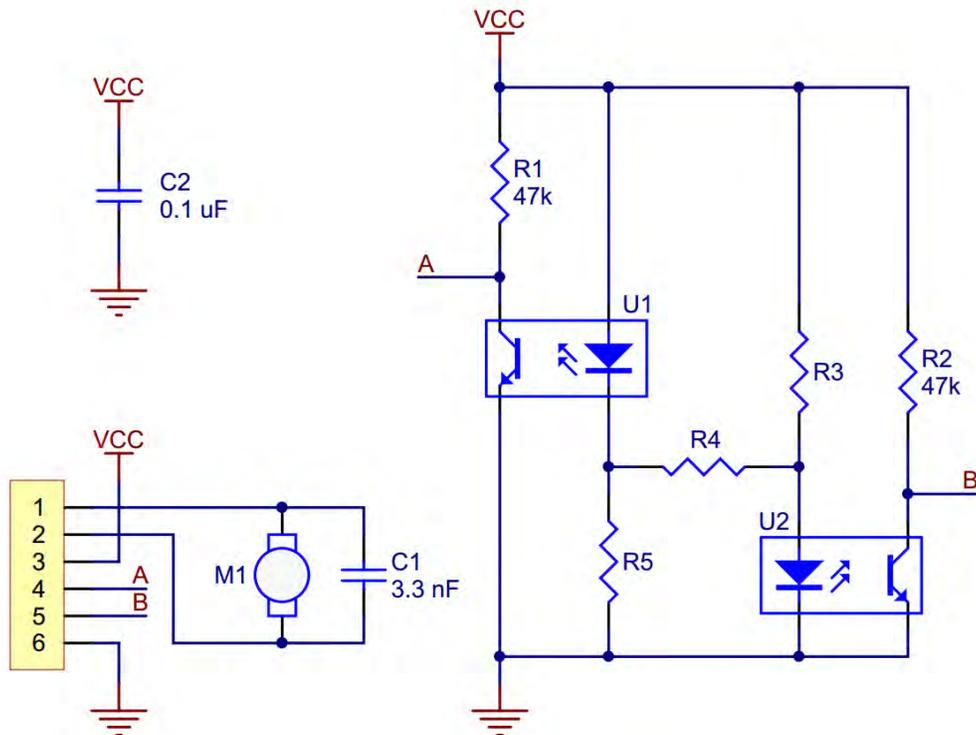


Figura 3.39: a) Encoders ópticos de Pololu y configuraciones de 3 y 5 estrellas que se montan en la flecha extendida del micro motor [?]; b) pines de los encoders [?].



5 V version: R4 = 220 Ω, R3 and R5 not populated (emitters in series)

3.3 V version: R4 not populated, R3 and R5 = 180 Ω (emitters in parallel)

Figura 3.40: Esquema electrónico de los encoders como lo indica Pololu [?].

Adecuación de la señal

Para poder leer los sensores es necesario adecuar la señal antes de decodificarla, después de observar las primeras pruebas a los motores (véase sección 4.1.1 Pruebas a las señales de salida de

los Encoders). Uno de los sensores (motor izquierdo) es adecuado y decodificado directamente por el microcontrolador esclavo (véase sección 3.3.3 dsPIC Esclavo), el otro sensor (motor derecho) es adecuado mediante el uso de una compuerta lógica tipo NOT, integrada con un comparador con histéresis tipo Schmitt trigger, o disparador de Schmitt. Posteriormente es decodificada mediante un Flip-Flop tipo D.

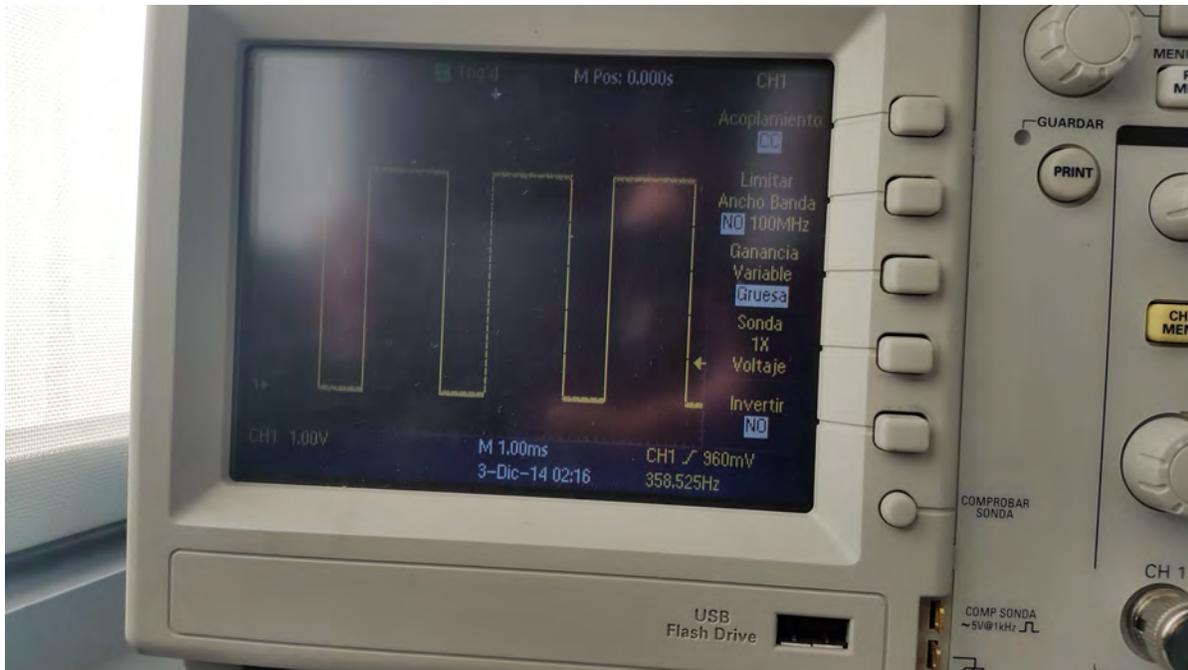


Figura 3.41: Adecuación de la señal mediante la compuerta NOT con ST.

El IC de compuertas NOT con ST (Schmitt trigger) elegido es el *CD40106BM*, de tecnología tipo CMOS (complementary metal-oxide-semiconductor, por sus siglas en inglés), similar en funcionamiento al IC de tecnología TTL *74LS14*. Se elige el *CD40106BM* porque, a diferencia de su similar de carácter TTL, no distorsiona la señal de entrada al no tener la entrada de voltaje y/o corriente suficiente para que deje de encontrarse en el triestado del TTL. El *CD40106BM* tiene 6 compuertas tipo NOT, de las cuales únicamente se usan 2 de ellas, que corresponden a las señales *OUT A* y *OUT B* del sensor. Un ejemplo del resultado de su uso se muestra en la Figura 3.41, donde ahora sí se puede apreciar una señal cuadrada.

Una vez que las señales han sido adecuadas el siguiente paso es decodificarlas. Esto se consigue mediante el uso de un Flip-Flop tipo D (FF-D), el IC *CD4013BC*, donde una de las señales se conecta al CLK (reloj) del FF-D y la otra señal a la entrada D, obteniéndose una salida en la terminal Q de '0' lógico cuando el motor gira en un sentido y '1' lógico cuando gira en sentido contrario. El esquema del circuito, propuesto por el M.I. Ulises Martín Peñuelas Rivas, se muestra en la Figura 3.42.

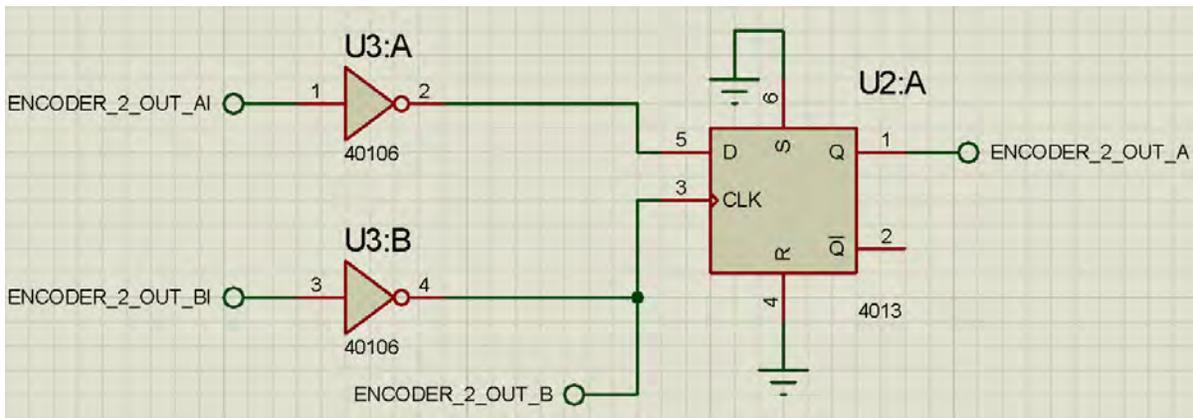


Figura 3.42: Circuito decodificador de la señal del encóder.

En la Tabla 3.1 se muestran los resultados de las pruebas empleando el módulo QEI del microcontrolador y el decodificador con el FF-D de la salida que arrojan ambos sensores una vez adecuada la señal (véase sección 4.1.1 Pruebas a las señales de salida de los Encoders). También se aprecian las configuraciones de conexión para que el sentido de giro sea positivo o negativo; se considera como positivo cuando el valor del registro del QEI del microcontrolador tiene signo positivo y que el LED de la terminal Q del FF-D esté encendido. El sentido de giro se mide con respecto a la reducción vista de frente, como se aprecia en la Figura 3.43.

Tabla 3.1: Conexión de pines y sentidos de giro del motor donde la salida es positiva.

Entradas	Señales	Sentido de giro del motor
QEA ↔ QEB	OUT A ↔ OUT B	Horario
QEA ↔ QEB	OUT B ↔ OUT A	Antihorario
FF-D/D ↔ FF-D/C	OUT A ↔ OUT B	Horario
FF-D/D ↔ FF-D/C	OUT B ↔ OUT A	Antihorario



Figura 3.43: La convención para determinar si el ángulo de giro del motor es horaria o antihoraria se determina viendo el motor desde el frente de la reducción [?].

Lectura de los encoders

La información que proveen los encoders se lee cada 50 [ms]. El valor obtenido de pulsos se debe traducir a revoluciones por minuto o a velocidad angular para su posterior tratamiento. La conversión de pulsos a velocidad angular depende del número de dientes de la estrella (5 para incrementar al máximo la resolución), el periodo de muestreo, la capacidad del sensor para medir los flancos (4 para el caso del QEI del microcontrolador, véase sección 3.3.1 *dsPIC30F4011*, y 1 para el FF-D) y la relación de la reducción acoplada (297.92:1). El factor de conversión se calcula de la siguiente manera:

$$REV_{FP}[=] \frac{\text{flancos}}{\#dientes \cdot \text{resolución}} = \frac{\text{flancos}}{5 \cdot \text{resolución}}$$

$$REV_R : REV_{FP} \rightarrow 1 : 297.92$$

$$RPM[=] \frac{\text{rev}}{\text{min}}$$

$$P_m = 50[\text{ms}]$$

$$P_m = 50[\text{ms}] \left[\frac{1[\cancel{s}]}{1000[\cancel{ms}]} \right] \left[\frac{1[\text{min}]}{60[\cancel{s}]} \right] = \frac{50}{60000}[\text{min}]$$

$$RPM = \frac{REV_R}{\text{min}} = \frac{REV_{FP}}{\frac{50}{60000}} = \frac{\frac{\text{flancos}}{5 \cdot \text{resolución}}}{\frac{50}{60000}} = 0.8056 \frac{\text{flancos}}{\text{resolución}}$$

$$RPM = 0.8056 \frac{\text{flancos}}{\text{resolución}} \left[\frac{\text{rev}}{\text{min}} \right] \quad (3.1)$$

$$S = 0.8056 \frac{\text{flancos}}{\text{resolución}} \left[\frac{\text{rev}}{\text{min}} \right] \left[\frac{2 \cdot \pi[\text{rad}]}{1[\text{rev}]} \right] \left[\frac{1[\text{min}]}{60[\text{s}]} \right]$$

$$S = 0.08436 \frac{\text{flancos}}{\text{resolución}} \left[\frac{\text{rad}}{\text{s}} \right] \quad (3.2)$$

donde:

REV_{FP} : revoluciones en la flecha principal

REV_R : revoluciones en la reducción

flancos : flancos leídos

$\#dientes$: número de dientes de la estrella

resolución : resolución de número de flancos que puede leer el sensor

P_m : periodo de muestreo

RPM : revoluciones por minuto

S : velocidad angular

Esquema electrónico

Cada encóder se conecta a un MOSFET *DMN6040SSD* para poder controlar su encendido y apagado mediante la señal del microcontrolador. La terminal **GND** del encóder se conecta a la terminal drenadora del MOSFET, la terminal sustrato a la tierra general y la compuerta a una terminal del microcontrolador. Los encoders tienen que conectarse mediante un cable a la placa del microcontrolador. Su esquema se aprecia en la Figuras 3.44 y 3.45.

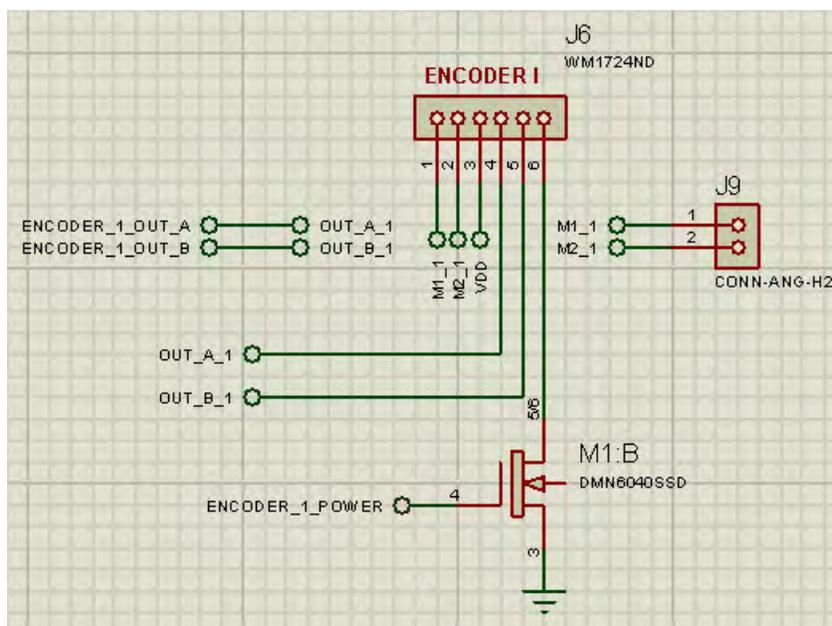


Figura 3.44: Esquema del encóder izquierdo.

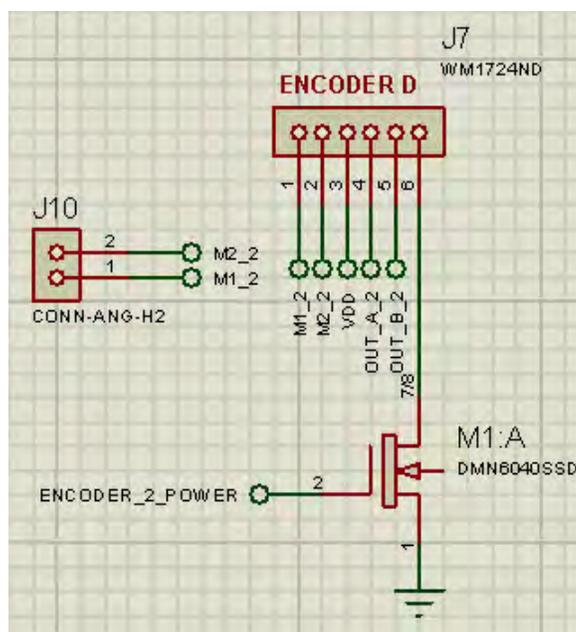


Figura 3.45: Esquema del encóder derecho.

B) IMU

Descripción general

El IC *MPU-6050* es una unidad de medición inercial (IMU, por sus siglas en inglés), de tecnología MEMS (microelectromechanical systems, por sus siglas en inglés), que se compone de 6 ejes; 3 ejes corresponden al acelerómetro y 3 ejes corresponden al giroscopio. El *MPU-6050* tiene 3 ADCs (analog-to-digital, por sus siglas en inglés), o convertidores analógico a digital, de 16 bits para digitalizar la salida de los giroscopios y 3 ADCs para la salida de los acelerómetros.

Los registros de la IMU se acceden a través de un bus I²C (Inter-Integrated Circuit, por sus

siglas en inglés) a una velocidad de 400 [kHz] (fast). El IC también incluye un sensor de temperatura embebido y un oscilador con una variación de $\pm 1\%$ a lo largo del rango de temperatura. Su alimentación es a un voltaje de 3.3 [V].

Las características principales del giroscopio son:

- Salida digital de los ejes X , Y y Z con una resolución programable de ± 250 , ± 500 , ± 1000 y ± 2000 [°/s].
- Los ADCs integrados de 16 bits permiten el muestreo simultáneo de los giroscopios.
- Margen de error mejorado y estabilidad en la sensibilidad térmica, reduciendo la necesidad de calibración.
- Desempeño mejorado de ruido de baja frecuencia.
- Filtro paso bajas programable.

Las características principales del acelerómetro son:

- Salidas digitales de los 3 ejes del acelerómetro con una resolución programable de $\pm 2g$, $\pm 4g$, $\pm 8g$ y $\pm 16g$.
- Los ADCs integrados de 16 bits permiten el muestreo simultáneo de los acelerómetros.
- Detección de orientación.
- Detección de caída libre.

Giroscopios

El IC *MPU-6050* integra tres giroscopios de categoría MEMS, vibratorios e independientes entre sí, que detectan la rotación a lo largo de los ejes X , Y y Z . Cuando los giroscopios son rotados alrededor de cualquiera de los ejes de sensado, el efecto de Coriolis causa una vibración que es detectada por un elemento capacitivo. La señal resultante es amplificada, demodulada y filtrada para producir un voltaje que es proporcional al ángulo de giro. Este voltaje es digitalizado usando ADCs individuales integrados en el chip con una resolución de 16 bits para muestrear cada eje. La resolución de los sensores se puede programar digitalmente a ± 250 , ± 500 , ± 1000 o ± 2000 [°/s] (dps). El muestreo del ADC se puede programar de 8000 muestras por segundo a 3.9 muestras por segundo, el filtro paso bajas que selecciona el usuario permite un amplio rango de frecuencias de corte.

Los registros asociados principales son:

- Frecuencia de muestreo (SMPLRT_DIV)
Especifica la división de la tasa de salida del giroscopio usada para generar la tasa de muestreo. La tasa de muestreo es generada dividiendo la tasa de salida del giroscopio entre el valor del registro:

$$\text{Sample Rate} = \frac{\text{Gyroscope Output Rate}}{1 + \text{SMPLRT_DIV}}$$

donde *Gyroscope Output Rate* = 8[kHz] ó 1[kHz], dependiendo del valor del DLPF.

- Configuración (CONFIG)

Este registro configura el pin de sincronización externa y el filtro digital paso bajas (DLPF). La configuración aplica tanto para los giroscopios como para los acelerómetros.

El filtro digital paso bajas se configura con los bits DLPF_CFG. El giroscopio y el acelerómetro son filtrados de acuerdo al valor de DLPF_CFG como se muestra en la Tabla 3.2:

Tabla 3.2: Frecuencias de corte para el filtro paso bajas.

DLPF_CFG	Acelerómetro		Giroscopio		
	Frec. corte [Hz]	Delay [ms]	Frec. corte [Hz]	Delay [ms]	Fs [kHz]
0	260	0	256	0.98	8
1	184	2.0	188	1.9	1
2	94	3.0	98	2.8	1
3	44	4.9	42	4.8	1
4	21	8.5	20	8.3	1
5	10	13.8	10	13.4	1
6	5	19.0	5	18.6	1
7	Reservado		Reservado		8

- Configuración del giroscopio (GYRO_CONFIG)

Este registro se usa para arrancar la auto-prueba de los giroscopios y para configurar la resolución de los giroscopios. Los bits FS_SEL seleccionan la resolución de salida del giroscopio como se muestra en la Tabla 3.3.

Tabla 3.3: Resolución de los giroscopios.

FS_SEL	Resolución [°/s]
0	±250
1	±500
2	±1000
3	±2000

- Mediciones del giroscopio

6 registros almacenan la última medición del giroscopio. Las mediciones del giroscopio se escriben en estos registros a la frecuencia de muestreo definida previamente.

La información de esos registros se actualiza a la frecuencia de muestreo siempre y cuando el bus serial se encuentre inactivo.

Cada medición de 16 bits del giroscopio tiene definida su resolución por el valor de FS_SEL. Para cada configuración de resolución, la sensibilidad del giroscopio cambia, como se muestra en la Tabla 3.4.

Tabla 3.4: Sensibilidad de los giroscopios.

FS_SEL	Resolución [$^{\circ}/s$]	Sensibilidad [$LSB/^{\circ}/s$]
0	± 250	131
1	± 500	65.5
2	± 1000	32.8
3	± 2000	16.4

El valor de salida de cada eje del giroscopio se traduce como se muestra en las ecuaciones 3.3, 3.4 y 3.5.

$$Giroscopio_X = \frac{GYRO_X_OUT}{Sensibilidad} \quad [^{\circ}/s] \quad (3.3)$$

$$Giroscopio_Y = \frac{GYRO_Y_OUT}{Sensibilidad} \quad [^{\circ}/s] \quad (3.4)$$

$$Giroscopio_Z = \frac{GYRO_Z_OUT}{Sensibilidad} \quad [^{\circ}/s] \quad (3.5)$$

donde:

$Giroscopio_X$: último valor de giro en el eje X del giroscopio

$Giroscopio_Y$: último valor de giro en el eje Y del giroscopio

$Giroscopio_Z$: último valor de giro en el eje Z del giroscopio

$GYRO_X_OUT$: valor signado de 16 bits

$GYRO_Y_OUT$: valor signado de 16 bits

$GYRO_Z_OUT$: valor signado de 16 bits

$Sensibilidad$: valor de sensibilidad marcado por el valor FS_SEL

Acelerómetros

Los acelerómetros del *MPU-6050* emplean masas de prueba independientes para cada eje. La aceleración a lo largo de un eje en particular induce un desplazamiento de la masa de prueba correspondiente, donde sensores capacitivos detectan el diferencial del desplazamiento.

La arquitectura del *MPU-6050* reduce la susceptibilidad del acelerómetro a variaciones de fabricación y al *drift* térmico.

Cuando el dispositivo es colocado en una superficie plana, medirá $0g$ en los ejes X y Y y $+1g$ en el eje Z .

La resolución del acelerómetro está calibrada de fábrica y es nominalmente independiente a su voltaje de alimentación. Cada sensor tiene un ADC $\sigma - \delta$ dedicada para proveer las salidas digitales. La resolución de la salida digital puede ser ajustada a $\pm 2g$, $\pm 4g$, $\pm 8g$ o $\pm 16g$.

Los registros asociados principales son:

- Frecuencia de muestreo (SMPLRT_DIV)
- Configuración (CONFIG)

- Configuración del acelerómetro (ACCEL_CONFIG)

Los 3 bits más significativos corresponden a los mecanismos de auto-prueba, necesarios en caso de necesitar calibración adicional. Los bits AFS_SEL seleccionan la resolución para las salidas de acuerdo con la siguiente tabla:

Tabla 3.5: Resolución de los acelerómetros.

AFS_SEL	Resolución
0	$\pm 2g$
1	$\pm 4g$
2	$\pm 8g$
3	$\pm 16g$

- Mediciones del acelerómetro

6 registros almacenan la última medición del acelerómetro. Las mediciones del acelerómetro, junto con las mediciones del sensor de temperatura y del giroscopio, así como los sensores externos, están compuestos de dos sets de registros: un registro interno y uno visible al usuario.

La información del registro interno de los sensores del acelerómetro siempre es actualizada a la frecuencia de muestreo. Mientras tanto, el valor del set de registros accesibles al usuario se actualiza con el valor del registro interno siempre que el bus de interfaz serial se encuentre “libre”.

Esto garantiza que al hacer una lectura corrida de los sensores, se obtendrán las lecturas de un mismo instante de muestreo.

Cada medición de 16 bits del acelerómetro tiene una resolución definida en los bits AFS_SEL. Para cada configuración de la resolución, la sensibilidad del acelerómetro cambia como se muestra en la siguiente tabla:

Tabla 3.6: Sensibilidad de los acelerómetros.

AFS_SEL	Resolución	Sensibilidad [LSB/g]
0	$\pm 2g$	16384
1	$\pm 4g$	8192
2	$\pm 8g$	4096
3	$\pm 16g$	2048

El valor de salida de cada eje del acelerómetro se traduce como se aprecia en las ecuaciones 3.6, 3.7 y 3.8.

$$Acelerómetro_x = \frac{ACCEL_X_OUT}{Sensibilidad} \quad [g] \quad (3.6)$$

$$Acelerómetro_y = \frac{ACCEL_Y_OUT}{Sensibilidad} \quad [g] \quad (3.7)$$

$$Acelerómetro_z = \frac{ACCEL_Z_OUT}{Sensibilidad} \quad [g] \quad (3.8)$$

donde:

Acelerómetro_X: último valor de aceleración en el eje *X* del acelerómetro

Acelerómetro_Y: último valor de aceleración en el eje *Y* del acelerómetro

Acelerómetro_Z: último valor de aceleración en el eje *Z* del acelerómetro

ACCEL_X_OUT: valor signado de 16 bits

ACCEL_Y_OUT: valor signado de 16 bits

ACCEL_Z_OUT: valor signado de 16 bits

Sensibilidad: valor de sensibilidad marcado por el valor *AFS_SEL*

Termómetro

Adicionalmente, la IMU tiene un termómetro integrado en el empaquetado. Al igual que los ejes del acelerómetro y giroscopio, el dato que se lee es producto de una lectura de un ADC independiente de 16 bits. De la misma manera, los registros asociados a las lecturas del termómetro se actualizan a la frecuencia de muestreo definida por el usuario. El dato de temperatura se mide en grados Celsius y la ecuación con la que se traduce el valor leído a través del bus es la siguiente:

$$Temperatura_{\circ C} = \frac{TEMP_OUT}{340} + 36.53 \quad [^{\circ}C] \quad (3.9)$$

donde:

Temperatura_{°C}: último valor de la temperatura

TEMP_OUT: valor signado de 16 bits

Preparación para su lectura

Antes de empezar a usar la IMU, una vez que ha sido encendida, la primera acción es limpiar el registro Manejo de Energía 1 (PWR_MGMT_1), dado que la IMU después de que se estabiliza tras el encendido ($T_{on} \leq 3[ms]$) inicia en modo “SLEEP”. De igual forma se debe limpiar el registro Manejo de Energía 2 (PWR_MGMT_2) para asegurar el funcionamiento sin intermitencia de bajo consumo de energía que tiene la IMU.

Selección del filtro paso bajas

Se realizan pruebas del filtro integrado de la IMU a 3 frecuencias distintas para determinar el idóneo a emplear (véase sección [4.1.2 Pruebas a los ejes de la IMU](#)). De entre los 3 filtros probados, se toma el filtro a la frecuencia de corte de 21 [Hz] por ser el que se encuentra en el punto medio entre la desviación estándar de los datos y el retardo en la señal.

Configuración de la interrupción externa

Una vez que se han configurado los registros de energía, del acelerómetro y giroscopio, el siguiente paso es configurar los registros de configuración de la interrupción (INT_PIN_CFG e INT_ENABLE), de tal forma que la interrupción externa que brinda la IMU, cuando tiene información disponible, sea del tipo “*Open Drain*” (colector abierto) cuyo valor de ‘1’ lógico sean 0 [V].

Finalmente, la IMU está totalmente configurada para ser leída constantemente. Se dejan pasar los datos de los primeros 5 segundos para permitir a los ejes estabilizarse y se pueden tomar

ahora una muestra de datos que servirán como referencia para eliminar el error permanente del giroscopio.

Esquema electrónico de la IMU

El *datasheet* del *MPU-6050* sugiere un esquema típico para el IC. Se compone de algunos capacitores para estabilización a las diferentes terminales de alimentación del IC. La IMU se alimenta con un voltaje de 3.3 [V], suministrado por un regulador lógico.

Para la comunicación I²C es necesario emplear 2 resistencias tipo “pull-up” acordes a la velocidad que se manejará en el bus, en este caso, 2 resistencias de 2.2 [kΩ], conectadas a las terminales **SCL** y **SDA**.

El pin de interrupción externa, para ser configurado del tipo “*Open Drain*”, requiere de una resistencia tipo “pull-up” de 1 [kΩ]. Tanto las resistencias del bus I²C como la de la interrupción externa se conectan al voltaje lógico de 3.3 [V] y luego a la terminal indicada.

Adicionalmente, para poder reiniciar el dispositivo mediante hardware cuando falle reiteradamente la comunicación en el bus I²C o cuando todos los datos leídos sean siempre *0x00*; o bien, para poder apagar la IMU cuando no se requiera de su uso, la alimentación de la IMU se conecta a un MOSFET *DMN6040SSD*. La terminal drenadora del MOSFET se conecta a la tierra del dispositivo y la terminal sustrato a la tierra general. Por su parte, la terminal compuerta se conecta a una terminal del microcontrolador esclavo, que sirve como activación del tipo encendido/apagado que da la señal de control para encender o apagar la IMU. El esquema de la IMU se aprecia en la Figura 3.46.

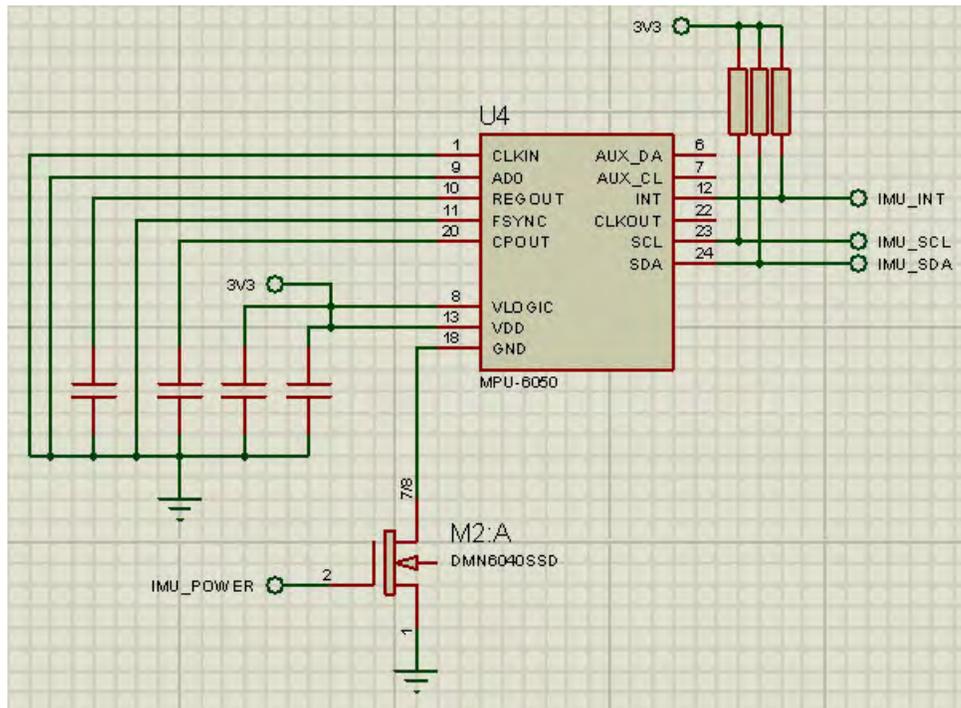


Figura 3.46: Esquema electrónico de la IMU.

C) Nivel de carga de la batería

Descripción general

El nivel de carga de las baterías se sensa de manera independiente la una de la otra. Cualquiera

de las baterías se considera descargada cuando su nivel de carga ha llegado a un voltaje de 3.0 [V]. En el caso ideal de que ambas baterías se descargaran al mismo ritmo, el voltaje mínimo a regular para la etapa de regulación sería de 6.0 [V], voltaje que corresponde con la caída máxima de voltaje de los reguladores a 5 [V]. El esquema de las baterías en serie se aprecia en la Figura 3.47.

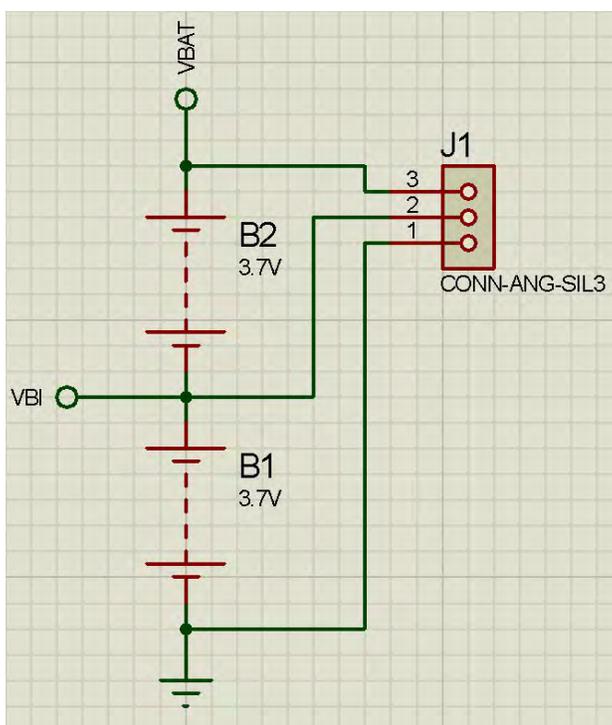


Figura 3.47: Esquema de las baterías en serie.

Existen varios métodos e IC's (integrated circuits, por sus siglas en inglés) para tal fin. El método elegido es a través de un juego de comparadores con amplificadores operacionales para evitar problemas de alimentación para un IC o exceso en las entradas de voltaje a las terminales tanto de los IC's como de los microcontroladores, así como evitar carga al procesador al tener que medir el nivel constantemente a través de un ADC.

Comparadores

Para poder leer de manera independiente cada batería se emplean comparadores *MCP6541*. Estos comparadores dan la flexibilidad de ser alimentados a través de una fuente simple, en este caso, cada batería; sólo necesitan ser conectados a una referencia y al voltaje a medir para dar una salida tipo digital.

Una de las maneras para establecer una referencia de voltaje fija es mediante el uso de diodos Zener. Dado que el voltaje mínimo a considerar como batería descargada es de 3.0 [V], entonces, cuando la batería se haya descargado hasta ese punto, la referencia debe estar calibrada a un voltaje menor a los 3.0 [V], de lo contrario no se podría comparar una referencia a 3.0 [V] con el mismo voltaje a 3.0 [V]; por ello, la referencia debe calibrarse a un valor menor a los 3.0 [V].

Se elige un valor lejano a los 3.0 [V] que exista de manera comercial y se encuentre disponible en un empaquetado de montaje superficial lo suficientemente pequeño pero manejable con las manos. El diodo seleccionado es el diodo Zener *MMSZ4680T1G*, cuyo voltaje Zener nominal es de 2.2 [V]. La corriente mínima para alimentar el diodo es de 50 [μ A]. El *MMSZ4680T1G* no indica una corriente mínima para las terminales de entrada más una corriente máxima de entrada de hasta 2

[mA]. La resistencia mínima de carga que alimente el diodo Zener y el pin de entrada del IC debe ser de al menos 50 $[\mu A]$, el voltaje en ese momento será de 3.0 [V], que es el voltaje mínimo al que se dejará caer la carga de la batería. La resistencia de carga debe de disipar el resto de voltaje en el punto en el que las baterías se encuentren descargadas, obteniéndose la resistencia mínima en ese instante. Por la ley de Ohm, el cálculo de la resistencia mínima es

$$R_{min} = \frac{V}{I} = \frac{3.0[V] - 2.2[V]}{50[\mu A]} = 16[k\Omega].$$

De manera arbitraria, para asegurar el funcionamiento del diodo y del pin de entrada del IC, se considera una corriente 3 veces mayor a la mínima para alimentar tanto el diodo Zener como el pin de entrada. Usando una resistencia de 5.3 $[k\Omega]$, se cumplen los parámetros mínimos de corriente tanto para el diodo como para el pin de entrada del IC, considerando como voltaje máximo el nominal de una batería común de celular tipo ión-litio de 3.7 [V]:

$$I_{min} = \frac{V}{R} = \frac{3.0[V] - 2.2[V]}{5.3[k\Omega]} = 150.9[\mu A].$$

Para el pin de entrada que involucra el voltaje a sensar, se usa un divisor de voltaje que cumpla con las condiciones de que a 3.0 [V] debe de arrojar un voltaje muy cercano a los 2.2 [V] del diodo Zener. Fijando el valor de uno de los resistores, $R_2 = 448[k\Omega]$, se puede determinar el valor del primer resistor. La fórmula del divisor de voltaje es

$$V_o = \frac{V_i \cdot R_2}{R_1 + R_2} \rightarrow R_1 = \frac{V_i \cdot R_2}{V_o} - R_2$$

sustituyendo los valores de voltaje del divisor y R_2 se tiene que

$$R_1 = \frac{3.0[V] \cdot 448[k\Omega]}{2.2[V]} - 330[k\Omega] = 162.9[k\Omega];$$

el valor comercial hallado más cercano es de 162 $[k\Omega]$, por lo que el voltaje del divisor termina siendo

$$V_o = \frac{V_i \cdot R_2}{R_1 + R_2} = \frac{3.0[V] \cdot 448[k\Omega]}{168[k\Omega] + 448[k\Omega]} = 2.18[V].$$

Dado que el *MMSZ4680T1G* es un amplificador operacional configurado ya en modo comparador, se debe colocar la referencia en su terminal no inversora y el voltaje a sensar a la terminal inversora. Al no tener capacitores ni elementos adicionales conectados, salvo los mencionados anteriormente, su salida no tiene histéresis y tendrá un voltaje muy cercano al de alimentación.

De acuerdo con la hoja de especificaciones de los microcontroladores, el voltaje mínimo en un pin de entrada para ser considerado como ‘1’ lógico es de 2.5 [V], por lo que la salida de los comparadores es correcta en términos de proporcionar al menos ese voltaje lógico. Sin embargo, un comparador está alimentado con las terminales de la segunda batería, donde ambas están en serie, por lo que su voltaje para el ‘0’ lógico correspondería, en el caso de que la primer batería esté ya descargada, a 3.0 [V]; de igual manera, en el caso de que la primer batería esté totalmente cargada y la segunda descargada, el voltaje del ‘1’ lógico correspondería a 6.7 [V], el cual dañaría el microcontrolador. Para solucionar este problema, al segundo comparador se le coloca un resistor y un diodo Zener a su salida para regular su voltaje máximo a un voltaje adecuado. El voltaje

máximo de la salida se regula a 4.3 [V] mediante otro diodo Zener, el *MMSZ4687T1G*. Ahora se referenciará ese voltaje a la tierra general del sistema, por lo que el cálculo de la resistencia de ese diodo se calcula considerando el momento en el que ambas baterías se encuentren descargadas y de nuevo a 3 veces la corriente nominal para encender el diodo, que de la misma manera es de 50 [μ A]

$$R = \frac{V}{I} = \frac{6.0[V] - 4.3[V]}{150[\mu A]} = 11.33[k\Omega].$$

Finalmente, el resistor comercial seleccionado tiene una resistencia nominal de 11.3 [k Ω].

La salida de voltaje máxima del segundo comparador ya está regulada, pero el voltaje de la señal **LOWBAT2** sigue oscilando al nivel de carga que tenga la primer batería. Para compensarlo se agrega un nuevo comparador. Este último comparador va a adecuar la salida de la señal comparando entre la señal oscilante y una nueva referencia. Está alimentado por el voltaje lógico regulado de 5.0 [V] que alimenta a todos los circuitos lógicos (véase sección [B](#)) [Regulación de la alimentación](#)).

La nueva referencia, conectada a la terminal no inversora, es un divisor de voltaje calibrado a una salida de 4.0 [V]; esto es porque, cuando la salida del comparador sea nula, el voltaje en el pin oscilará entre 3.7 [V] y 3.0 [V], y cuando se active será el voltaje regulado de 4.3 [V]. Mientras que la referencia sea mayor a la señal, conectada a la terminal inversora, la salida del comparador estará “apagada”, así, cuando la referencia sea menor al voltaje de la señal, la salida se “encenderá”. El primer resistor del divisor se calcula fijando la segunda resistencia a 330 [k Ω], por lo que su cálculo da

$$R_1 = \frac{5.0[V] \cdot 330[k\Omega]}{4.0[V]} - 330[k\Omega] = 82.5[k\Omega],$$

donde el valor comercial más cercano es de 82 [k Ω], por lo que el voltaje del divisor termina siendo

$$V_o = \frac{V_i \cdot R_2}{R_1 + R_2} = \frac{5.0[V] \cdot 330[k\Omega]}{82[k\Omega] + 330[k\Omega]} = 4.005[V].$$

La salida del comparador es prácticamente la misma que la de un IC de tecnología TTL (transistor-transistor logic, por sus siglas en inglés), que es la habitual del microcontrolador. El esquema de los comparadores que se conectan a las baterías se aprecian en la [Figura 3.48](#) y del comparador adicional que adecúa el voltaje para no dañar el microcontrolador se aprecia en la [Figura 3.49](#).

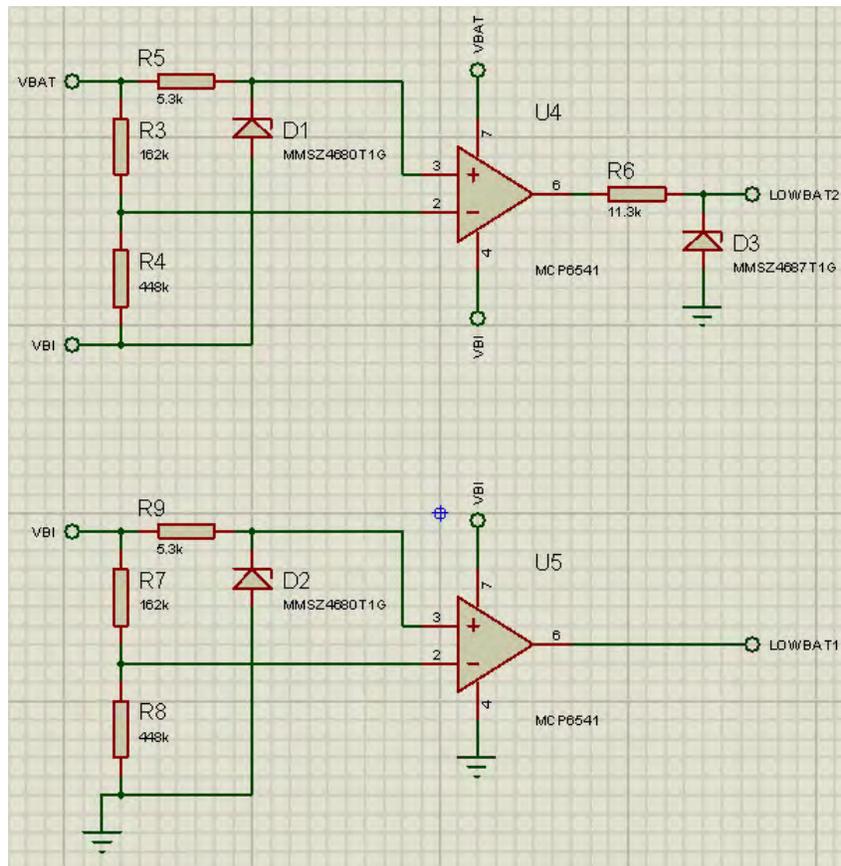


Figura 3.48: Esquema de los comparadores previo a tener una válvula que los habilite o deshabilite.

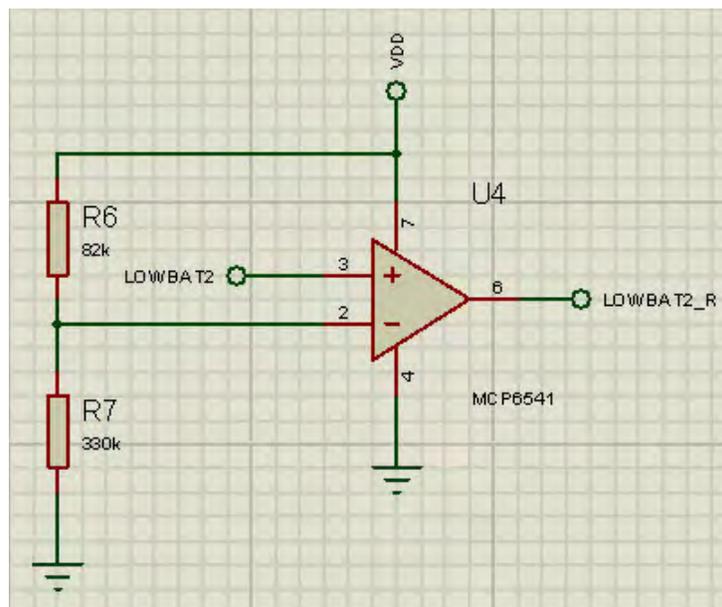


Figura 3.49: Esquema del comparador que recibe la segunda señal y adecúa el voltaje para ser admitido al microcontrolador.

Habilitación y deshabilitación de los comparadores

Hasta este punto los sensores del nivel de carga de la batería funcionan. Pero demandan una

corriente continuamente, por lo que, aunque el robot se apague, los comparadores seguirán drenando corriente de las baterías. Para evitar eso, se emplean un par de MOSFET conectados a cada comparador y un MOSFET general que regula la corriente general de todo el robot, (véase [B\) Regulación de la alimentación](#)).

Los MOSFET empleados se encuentran empaquetados por pares en el IC *DMN6040SSD*. Son de canal “N”, por lo que los comparadores se conectan, en su terminal de alimentación positiva, a la alimentación de cada batería; mientras que sus terminales de alimentación negativa se conectan a la terminal drenadora (D) del MOSFET; a su vez, la terminal sustrato (S) se conecta a la terminal negativa de la batería. La compuerta (G) es la terminal que al alimentar con voltaje permite el paso de la corriente a través del MOSFET, la cual está conectada al botón general de encendido/apagado. De esta manera, cuando se enciende el robot a través del botón de encendido/apagado, se alimentan los comparadores que van conectados a las baterías y el tercero que está alimentado a través del voltaje regulado lógico; en el momento en el que se apague el robot, los comparadores dejarán de recibir corriente y la batería no se drenará. El esquema de conexión de los MOSFET con el botón se aprecia en la Figura 3.50.

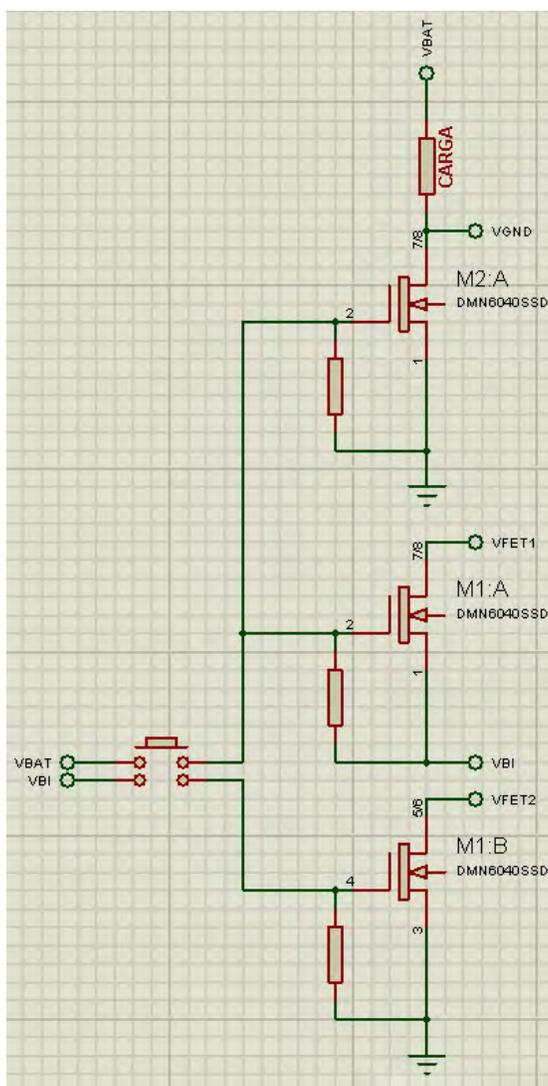


Figura 3.50: Esquema del botón de encendido/apagado y los MOSFET.

Los esquemas finales de los comparadores se aprecian en la Figura 3.51 y la Figura 3.52.

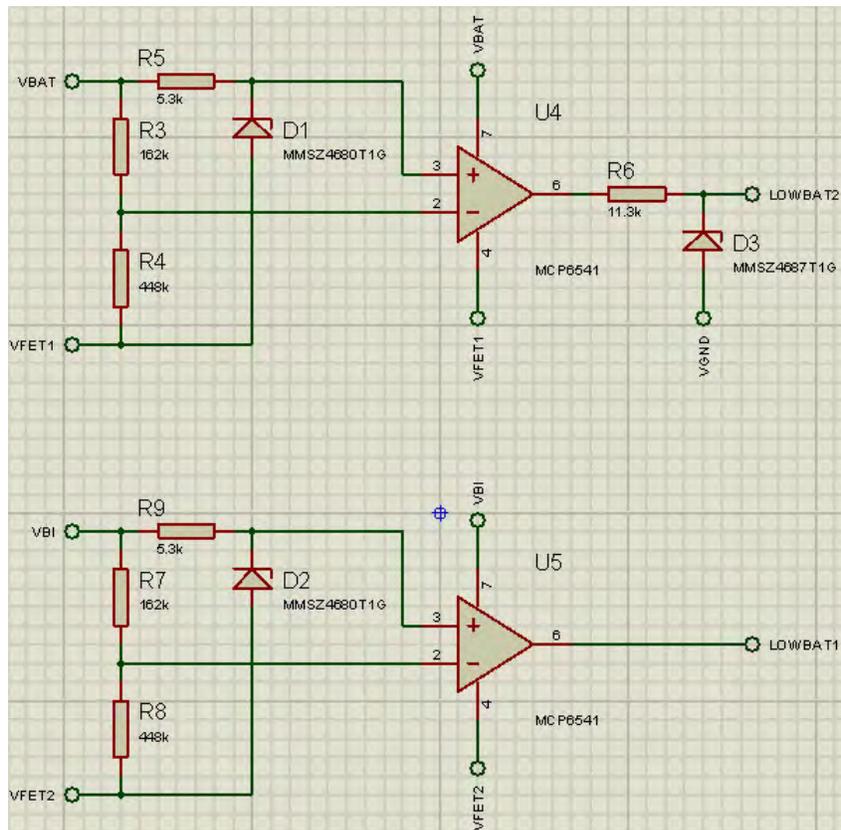


Figura 3.51: Esquema final de los comparadores una vez acoplado ya el MOSFET de encendido/apagado.

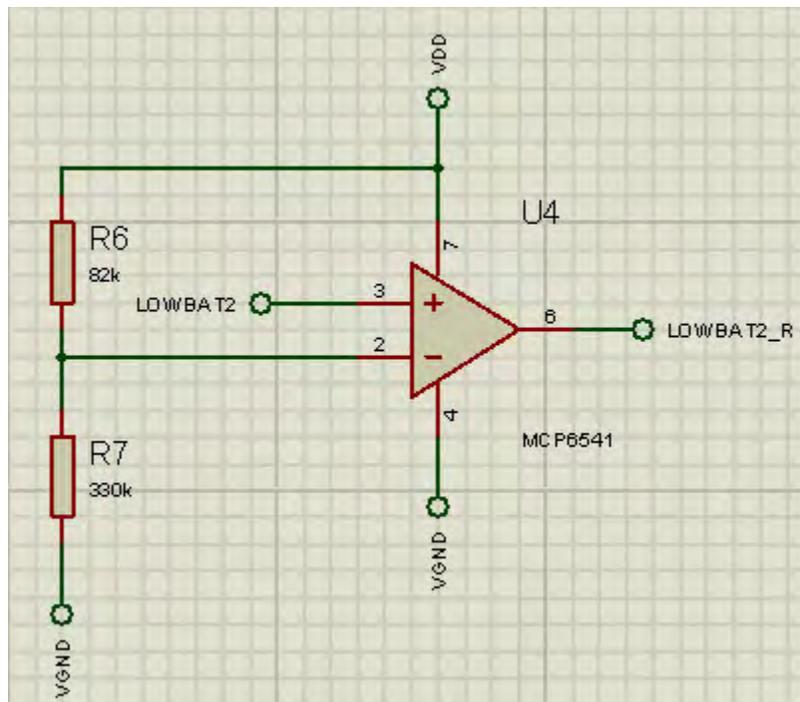


Figura 3.52: Esquema final del comparador que adecua la segunda señal al microcontrolador, ya con el MOSFET de encendido/apagado.

Las pruebas realizadas con los comparadores se encuentran en la sección [4.1.3 Pruebas con los comparadores](#).

D) Ultrasónico

Descripción general

El sensor *HC-SR04* es un sensor ultrasónico que emplea un sonar para determinar la distancia que existe a un objeto frente a él. Tiene un rango de detección nominal de 2 [cm] a 4 [m] en un ángulo de detección de 30°. Su operación no se ve afectada por la intensidad luminosa ni por materiales oscuros; sin embargo, sí se ve afectado por ciertos objetos suaves, como telas, dando mediciones erróneas frente a ese tipo de objetos. El sensor se encuentra empaquetado en una placa con su transmisor y receptor, con 4 pines de funcionamiento.

Las características generales del sensor son las siguientes:

- Voltaje de alimentación: 5 [V]
- Corriente en stand by: ≤ 2 [mA]
- Corriente en funcionamiento: 15 [mA]
- Ángulo de detección: $\pm 15^\circ$
- Rango de detección: 2 [cm] a 4 [m]
- Resolución: 0.3 [cm]
- Placa de dimensiones: $45 \times 20 \times 15$ [cm]

En la Figura [3.53](#) se aprecia el sensor y sus 4 pines de funcionamiento, que de izquierda a derecha son:

- **VCC**: voltaje de alimentación
- **Trig**: disparador de la señal
- **Echo**: señal recibida
- **GND**: tierra del circuito

Operación

El diagrama de tiempos del sensor se muestra en la Figura [3.54](#). Para iniciar una medición, se debe aplicar un pulso en alto de al menos 10 [us] en el pin **Trig**. Este pulso iniciará una transmisión de una ráfaga ultrasónica de 8 ciclos a una frecuencia de 40 [kHz], y luego esperará por la reflexión de la ráfaga. Cuando el sensor detecta la reflexión, activa en alto el pin **Echo** y lo mantiene así en un tiempo proporcional a la distancia del objeto más próximo frente al sensor. Para obtener el cálculo de la distancia, es necesario medir el tiempo en alto en el pin **Echo**. La ecuación con la que se calcula la distancia es la siguiente:

$$distancia \quad [cm] = \frac{tiempo \quad [us]}{58} \quad (3.10)$$



Figura 3.53: Sensor ultrasónico HC-SR04 [8].

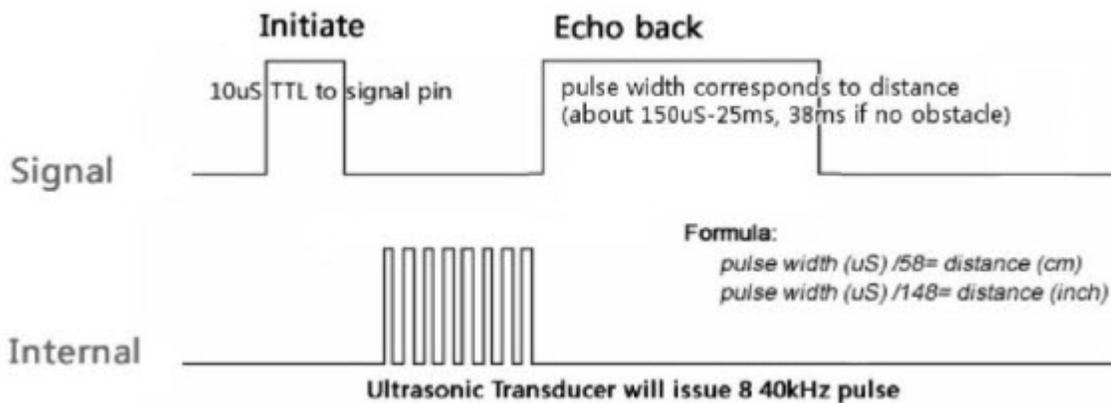


Figura 3.54: Diagrama de tiempos del sensor HC-SR04 [8].

Primeras pruebas

Las primeras pruebas con el sensor arrojaron como resultado lo siguiente:

- El rango real del sensor es mayor a los 4 [m], incrementándose cuando no encuentra obstáculo en campo despejado hasta 5.25 [m].
- El sensor no se puede leer a una frecuencia constante ni limitar el rango sin elementos adicionales, dado que aunque se le vuelva a dar el disparo al pin **Trig**, si el sensor no ha terminado de dar el pulso por el pin **Echo**, no terminará en ese instante el pulso actual ni reaccionará a un nuevo disparo.
- Si se corta la alimentación del sensor, a través de un transistor BJT o un MOSFET, se puede terminar la última lectura y se puede iniciar un nuevo disparo para una nueva lectura.

Esquema electrónico

Mediante el uso de un MOSFET *DMN6040SSD*, conectando el pin **GND** al drenador, tierra general al sustrato y la señal de encendido/apagado al microcontrolador, se consigue la uniformización de la frecuencia de muestreo del sensor y se puede limitar el rango del sensor a una distancia menor a la nominal. El sensor se conecta a la placa del microcontrolador a través de un cable. Su esquema se aprecia en la Figura 3.55.

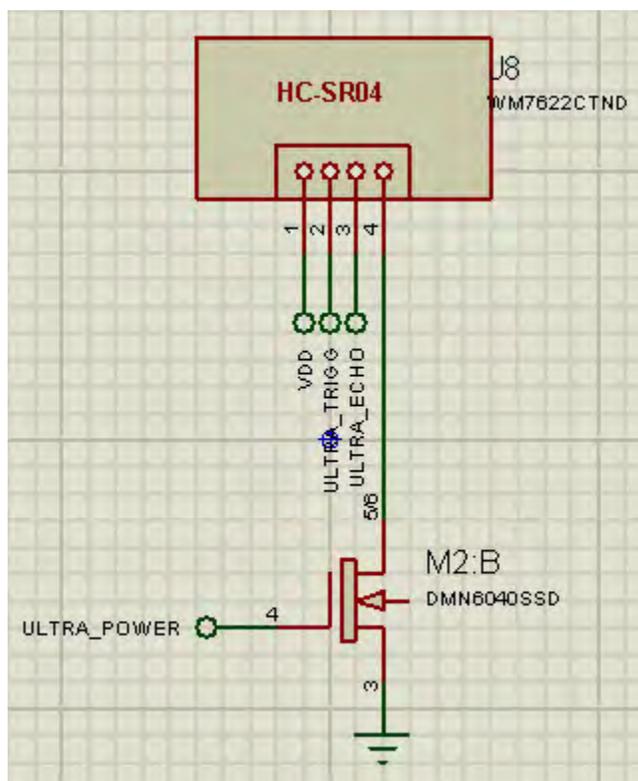


Figura 3.55: Esquema de conexión del sensor HC-SR04.

Las pruebas realizadas para conocer el comportamiento del sensor se encuentran en la sección [4.1.4 Lectura del sensor ultrasónico](#).

3.2.2 Actuadores

El robot cuenta con los siguientes actuadores:

- Dos micro motores de corriente directa.
- Veinticuatro LEDs RGB (cuatro LEDs por cara).

A) LEDs RGB

Descripción general

Los LEDs RGB (red-green-blue, por sus siglas en inglés) elegidos para el robot son del tipo redondo de 5 [mm] ultrabrillantes (Figura 3.56-a). Los LEDs son de ánodo común; lo que significa que los 3 colores integrados (rojo, verde y azul) reciben energía desde el mismo punto, mientras que el cátodo correspondiente a cada uno de ellos es independiente (Figura 3.56-b).

Cada color tiene un voltaje de funcionamiento a una corriente nominal de 20 [mA], como se muestra en la Tabla 3.7.

La alimentación de los LEDs se toma desde el regulador de “potencia” a 5 [V], de tal forma que, para prevenir daño a cada uno de los colores de los LEDs, se tiene que disipar el resto del voltaje a través de un resistor. Dado que los LEDs son de ánodo común, para poder encender los colores es necesario tener una etapa de potencia capaz de invertir la lógica convencional de ‘0’

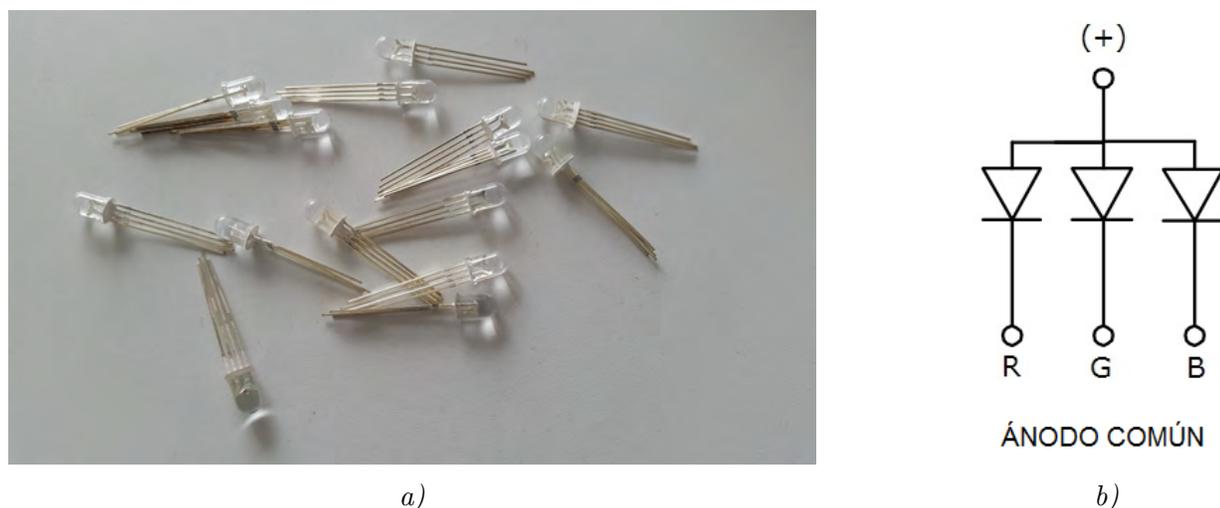


Figura 3.56: a) LEDs RGB empleados en el robot; b) Esquema eléctrico de un LED RGB de ánodo común [11].

Tabla 3.7: Voltajes de alimentación por color del LED RGB.

Color	Condición	V_{\min} [V]	$V_{\text{típ}}$ [V]	V_{\max} [V]	Intensidad luminosa [mcd]
Rojo	$I_F = 20[mA]$	1.8	2.0	2.2	2800
Verde		3.0	3.2	3.4	6500
Azul		3.0	3.2	3.4	1200

lógico \longleftrightarrow 0 [V] y ‘1’ lógico \longleftrightarrow 5 [V].

El robot cuenta con 4 LEDs RGB en paralelo por cara, dando un total de 24 LEDs. Cada cara del robot se puede encender en un color independiente a las demás. Cada LED tiene 3 pines de “control” para poder obtener el color deseado, de tal forma que, se necesitan 18 señales de control para conseguir la independencia de todas las caras del cubo.

Inconvenientes de usar el microcontrolador como generador de las señales PWM

Para conocer el comportamiento de los LEDs RGB se les hicieron unas primeras pruebas empleando un microcontrolador y un *ULN2803A*, de las que se descarta esta solución y se determina aparte pintar los LEDs (véase subsección 4.2.1 Encendiendo los LEDs RGB).

Para poder emplear la solución microcontrolador + *ULN2803A*, se requiere destinar 18 pines de I/O del microcontrolador, 3 IC *ULN2803A* y una carga al procesamiento del microcontrolador considerable al tener que realizar 18 PWM’s virtuales.

Al optar por esta combinación, el microcontrolador seleccionado no cuenta con los pines de propósito general suficientes para poder ejecutar los 18 PWM’s virtuales y realizar todas las demás funciones del robot; adicionalmente incrementa la carga de procesamiento para el microcontrolador. Aunado a lo anterior, el espacio requerido para los 3 IC *ULN2803A* es excesivo para las dimensiones del robot. Esta situación se resuelve al cambiar la unidad de control por un IC *PCA9956A*.

PCA9956A

El IC *PCA9956A* es un driver de LEDs a corriente constante de 24 canales controlado por un

bus I²C, de tecnología SMT, optimizado para oscurecimiento y parpadeo de LEDs RGBA (rojo, verde, azul y ámbar, por sus siglas en inglés) en productos de entretenimiento.

Cada salida cuenta con una resolución de 8 bits (256 niveles) a frecuencia fija de un controlador PWM que opera a 31.25 [kHz] con un ciclo de trabajo ajustable de 0 % a 99.6 % que le permite al LED ser configurado a un nivel específico de brillo. Una resolución adicional de 8 bits (256 niveles) de un controlador PWM grupal a una frecuencia fija de 122 [Hz] y una frecuencia ajustable entre 15 [Hz] a cada 16.8 segundos, con un ciclo de trabajo ajustable de 0 % a 99.6 %, se usa para oscurecer o parpadear todos los LEDs con el mismo valor. El esquema de conexión típico del IC es el que se aprecia en la Figura 3.57.

El IC *PCA9956A* es un excelente reemplazo de la combinación microcontrolador + *ULN2803A* ya que concentra en él la carga de generar los 18 canales de PWM y funciona como unidad de potencia para los LEDs. Aparte proporciona niveles configurables de PWM y de corriente máxima individual de cada canal, a través de una comunicación I²C.

Para determinar la dirección de esclavo en el bus I²C del *PCA9956A* se tienen 3 pines de direccionamiento, los cuales pueden ser alterados tanto por hardware como por software. El IC tiene 125 posibles combinaciones de direccionamiento que se logran mediante las siguientes conexiones:

- Conectar a tierra
- Resistor en pull-down a tierra
- Circuito abierto
- Resistor en pull-up a V_{DD}
- Conectar a V_{DD}

La conexión con resistores en pull-down a tierra seleccionada para los 3 pines, da como resultado la dirección del esclavo $0x20$.

El *PCA9956A* tiene un pin al que se le debe conectar una resistencia externa que dictamina la corriente máxima por canal. Dado que el regulador de “potencia” puede brindar hasta 1 [A] de corriente máximo (véase sección B) [Regulación de la alimentación](#)) y del cual, 600 [mA] se destinan al caso en que ambos motores se encuentren a rotor bloqueado (véase sección B) [Motores](#)); los 400 [mA] restantes se tienen que distribuir entre las corrientes de todos los LEDs (24) que conforman las caras del robot. Por ende, la corriente máxima por color se calcula de la siguiente manera:

$$I_{color} = \frac{I_{m\acute{a}x}}{\#colores \cdot \#caras} = \frac{400 \text{ [mA]}}{3 \cdot 6} = 22.22 \text{ [mA]}$$

La resistencia de corriente externa debe limitar la corriente a un valor similar a 22.22 [mA]. Calculando la corriente tomando en cuenta el número total de LEDs, a una corriente cada uno de 2.15 [mA] por color (medición experimental para una intensidad lumínica que consideramos adecuada) el cálculo arroja:

$$I_{color} = (2.15 \text{ [mA]}) \times (\#LEDs_{\times cara}) \times (\#colores) = (2.15 \text{ [mA]}) \times (4) \times (3) = 26 \text{ [mA]}$$

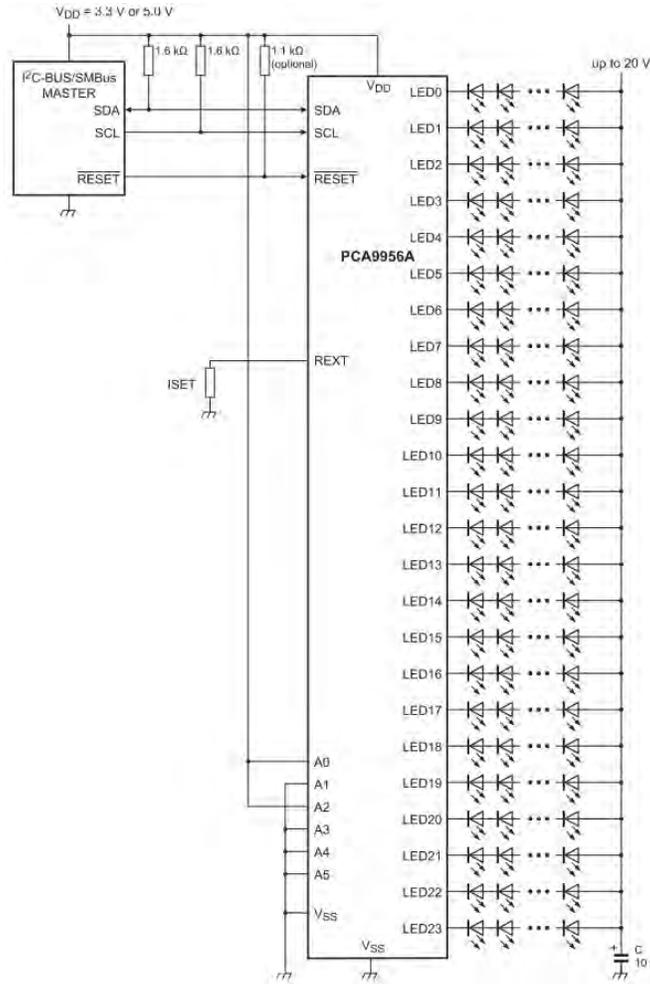


Figura 3.57: Esquema típico del **PCA9956A** sugerido en el datasheet [56].

El valor de 26 [mA] es aproximado y, aunque es un poco mayor y excedería la corriente máxima restante, la medición de 2.15 [mA] se realizó con 1 solo LED, por lo que se requiere una nueva calibración de corriente una vez que estén montados los 4 LEDs de la misma cara.

La fórmula con la que se calcula el valor de la resistencia externa es la siguiente:

$$R_{ext} = \frac{255}{4} \times \frac{900 \text{ [mV]}}{I_{color}} \quad (3.11)$$

$$R_{ext} = \frac{255}{4} \times \frac{900 \text{ [mV]}}{26 \text{ [mA]}} = 2.207 \text{ [k}\Omega\text{]}$$

El valor de la resistencia comercial más cercano es de 2.2 [kΩ], que se ajusta perfectamente al cálculo. De cualquier forma, el *PCA9956A* tiene un valor individual de corriente por canal regulable. En futuras pruebas se determina el valor concreto de cada una de las corrientes para intensidades lumínicas tolerables para el ojo humano a consideración nuestra.

Para disipar el voltaje restante al voltaje de trabajo de los LEDs en su máximo, se requiere de una resistencia por color que lo haga. Dado que el *PCA9956A* tiene la flexibilidad de ajustar la corriente mediante software, por facilidad se elige una resistencia única para todos los colores.

Como voltaje nominal de los LEDs se toma el del color verde o azul, colores cuya intensidad lumínica es mayor. El cálculo es el siguiente:

$$R_{color} = \frac{V_{disipar}}{I_{color}} = \frac{V_{alimentación} - V_{nominal}}{I_{color}} = \frac{5 [V] - 3.0 [V]}{26 [mA]} = 76.92 [\Omega]$$

Un valor de resistencia un poco mayor a los 76.92 $[\Omega]$ ayuda a limitar la intensidad de los colores verde y azul y promueve la intensidad del color rojo, que es el que se encuentra en mayor desventaja frente a los otros dos. El valor comercial cercano disponible es de 80.6 $[\Omega]$. A una corriente de 26 $[mA]$, el voltaje por color sería de:

$$V_{disipado} = R_{color} \times I_{color} = (80.6 [\Omega]) \times (26 [mA]) = 2.096 [V]$$

siendo casi el nominal de los colores verde y azul y siendo casi 1 $[V]$ menor al que requiere el color rojo. El esquema general del driver LED se aprecia en la Figura 3.58.

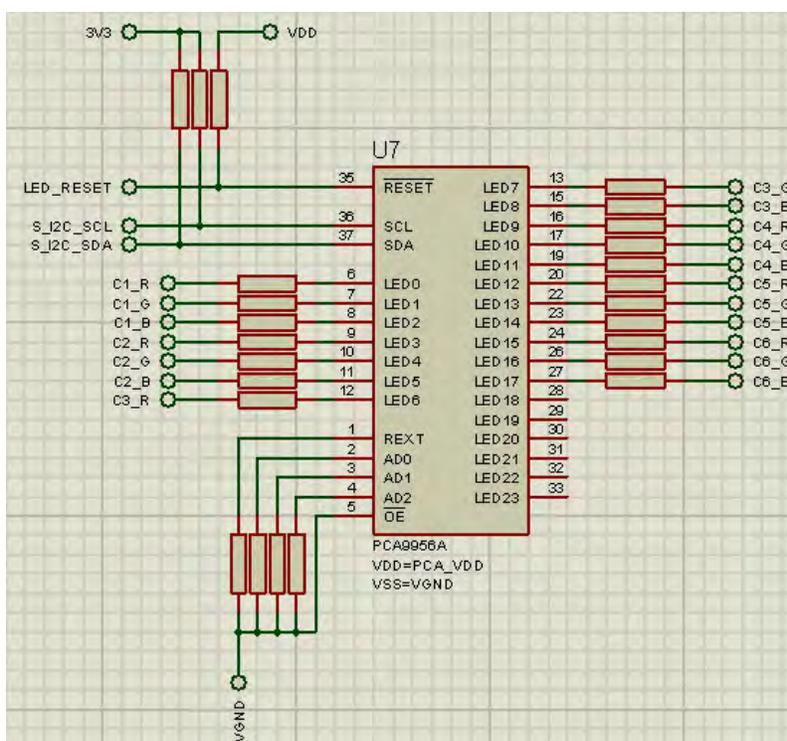


Figura 3.58: Esquema del **PCA9956A** empleado en el robot.

El resultado de la calibración de colores empleando el *PCA9956A* se encuentra en el Capítulo 4 subsección 4.3.1 Calibración de colores de los LEDs.

Montaje

Los LEDs RGB se sueldan a unas PCBs que permiten montarlos al tope con la placa y tener un área suficiente para soldar los cables que los conectan a los otros LEDs de su respectiva cara. También facilitan el montaje en las caras del cubo. En la Figura 3.59 se aprecian las placas a las que se sueldan los LEDs.

A los LEDs se les redujo la longitud del capuchón para evitar el contacto de los LEDs con las llantas del robot. El siguiente paso es soldar los LEDs a las placas.



Figura 3.59: PCBs para los LEDs RGB.

Finalmente, el último paso es montar los LEDs ya soldados al cubo, tanto a las tapas como al cuerpo. Se emplea cable plano para soldar los LEDs entre sí y obtener la terminal que se conecta a la placa de Energía para su energización.

Los LEDs se pegan en los agujeros destinados para su montaje empleando pegamento, también se emplea para fijar los cables y evitar su movimiento una vez todo montado. En la Figura 3.60 se aprecian los LEDs montados en la tapa.



Figura 3.60: LEDs montados en una tapa.

B) Motores

Los micro motores con flecha extendida de *Pololu* (Figura 3.61-a) se clasifican en las categorías: alta potencia, media potencia y baja potencia. La flecha extendida sirve para poder soldar al motor la placa de encoders para la lectura de la velocidad angular del motor (Figura 3.61-b) (véase sección

A) Encoders).

Todos los motores pueden ser adaptados con diferentes reducciones, que van de relaciones 5:1 a 1000:1. Los motores no deben de consumir más de 1 [A] porque el regulador no soporta un consumo mayor a esa corriente (véase sección B) [Regulación de la alimentación](#)). El caso de mayor consumo de corriente se presenta cuando los motores se encuentran a rotor bloqueado; las corrientes que consumen, nominalmente, son de 360 [mA] para los motores de baja potencia, 700 [mA] para los de media potencia y 1600 [mA] para los de alta potencia, todos a 6 [V].

Ese parámetro se toma como referencia para seleccionar los motores para el robot: los motores de baja potencia. Pensando en el máximo consumo de corriente, ambos motores a rotor bloqueado, consumen alrededor de 300 [mA] cada uno a 5 [V].



Figura 3.61: a) Micromotor de Pololu con reducción y flecha extendida [62]; b) se aprecia el motor con el encóder montado sobre la flecha extendida [?].

Los motores seleccionados de flecha extendida son de baja potencia y reducción 298:1, cuyos valores nominales, todos a 6 [V] son:

- Corriente a rotor bloqueado: 360 [mA]
- Velocidad sin carga: 45 [RPM]
- Torque a rotor bloqueado: 2.88 [kg·cm]

Pololu indica que bloquear los rotores puede decrementar rápidamente la vida de los motores, ocasionalmente resultando en un daño directo a las reducciones de los motores. Esto pasa especialmente para las reducciones de mayor relación, como las reducciones 298:1 y 1000:1, que pueden generar suficiente torque para dañarse a sí mismas. Bloqueos prolongados (en el orden de segundos) puede provocar un daño interno a los embobinados y las escobillas del motor, especialmente para las versiones de alta potencia.

A pesar de ello, tras algunas pruebas con los motores, bloqueando el rotor a 5 [V] e inclusive a 6 [V] por algunos segundos, no se ha generado daño aparente ni a la reducción ni al motor en sí y el desempeño se ha mantenido constante.

En el Capítulo 4 subsección [4.2.2 Pruebas de velocidad de los motores](#) se encuentran las primeras pruebas de velocidad realizadas a los motores para conocer sus características. De las gráficas obtenidas (véase la subsección) se intenta obtener una función de transferencia cuya aproximación

sea lo más apegada a las gráficas obtenidas experimentalmente. La función de transferencia es de primer orden y se aproxima mediante simulaciones en *Matlab* con el fin de conseguir una función que relacione el voltaje y la velocidad angular lo más rápido y mejor aproximado posible sin la necesidad de elaborar un banco de pruebas para modelarlos; aunque la segunda opción es la más correcta y la que arrojaría la función de transferencia mejor aproximada, la aproximación actual es buena y permite tener una función tolerablemente aproximada en poco tiempo y con pocas iteraciones.

La simulación se puede hacer mediante el entorno *Simulink* de *Matlab* o mediante la ejecución de un script con la simulación numérica. La función de transferencia de primer orden del motor tiene la estructura

$$T(s) = H(s) = \frac{Y(s)}{U(s)} = \frac{c}{s + p}$$

cuya representación en el espacio de estado es

$$\begin{aligned} \frac{Y(s)}{U(s)} &= \frac{c}{s + p} \\ Y(s)(s + p) &= c \cdot U(s) \\ s \cdot Y(s) + p \cdot Y(s) &= c \cdot U(s) \\ \dot{y}(t) + p \cdot y(t) &= c \cdot u(t) \end{aligned}$$

despejando $\dot{y}(t)$

$$\dot{y}(t) = -p \cdot y(t) + c \cdot u(t)$$

haciendo un cambio de variable

$$\begin{aligned} u(t) &= v(t) \\ y(t) &= \omega(t) \end{aligned}$$

finalmente la ecuación dinámica que relaciona el voltaje de entrada al motor $v(t)$ y la velocidad angular del motor $\omega(t)$ es

$$\dot{\omega}(t) = -p \cdot \omega(t) + c \cdot v(t). \quad (3.12)$$

Con la ecuación 3.12 se puede aproximar ahora el valor de c y p . El valor de p se determina a través de la constante de tiempo. De las pruebas y las gráficas a periodos de muestreo de 2.5 [ms] y 5 [ms] se determina que el motor alcanza su valor final en un periodo de 50 [ms] a 70 [ms]. El valor exacto no se puede determinar con este sensor dada la lectura ruidosa que se observa en todas las muestras. El ruido de la señal disminuye a mayores periodos de muestreo, pero se pierde información. Finalmente se toma como periodo de asentamiento del motor el valor promedio de las lecturas, que es de 60 [ms]. El valor del polo p se determina comúnmente como

$$p = \frac{4}{T} \text{ o bien } p = \frac{5}{T}$$

pero el valor de la simulación empleando esos parámetros da como resultado una gráfica que dista de la original, como se muestra en la Figura 3.62.

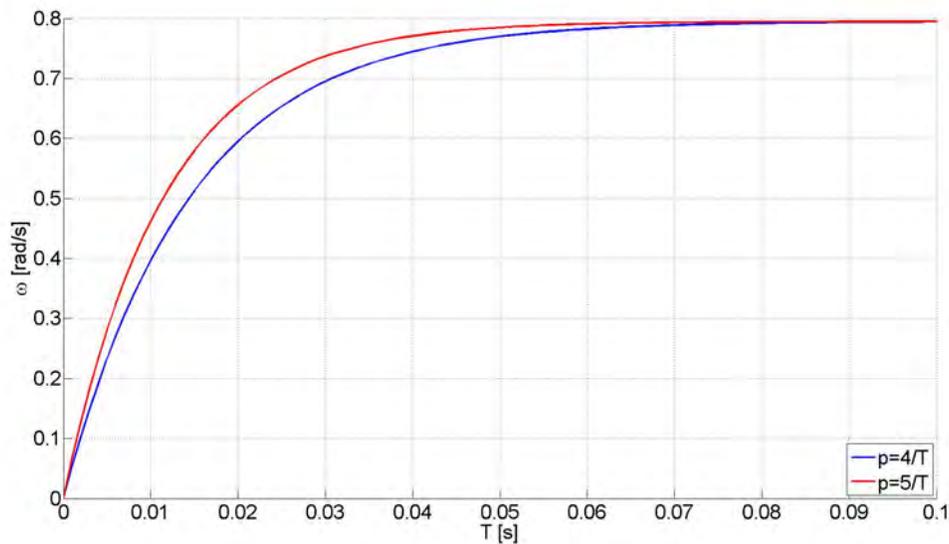


Figura 3.62: Simulación de velocidad del motor con constantes de tiempo $4/T$ y $5/T$.

Tras algunas simulaciones, y mediante el método de prueba y error, se encontró un valor para el polo bastante aproximado de

$$p = 8/0.060 = 133.33$$

dando como resultado la gráfica de la Figura 3.63.

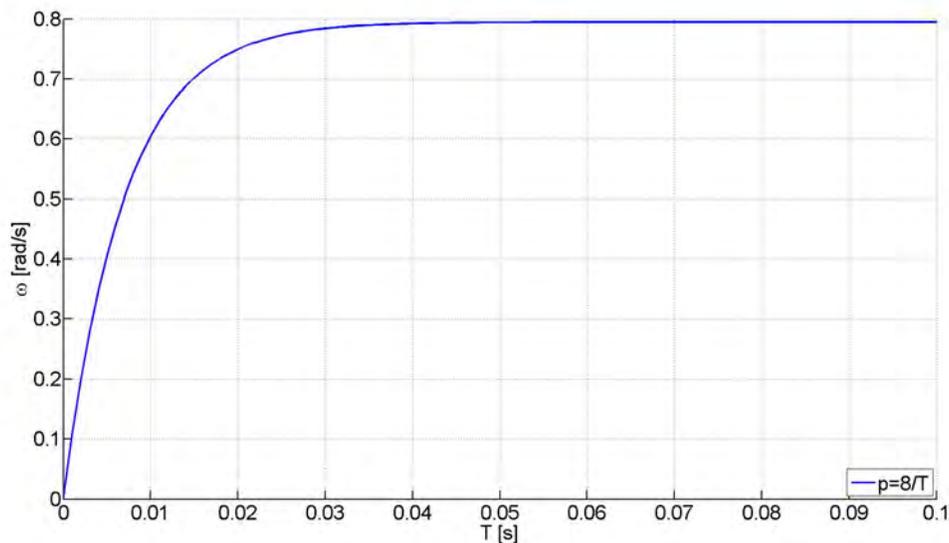


Figura 3.63: Simulación de velocidad del motor con constante de tiempo $8/T$.

El valor de la constante c depende del valor en estado estable del motor ante un escalón de 1 [V], pero se ajusta para corresponder al valor del escalón de 5 [V], el cual es de aproximadamente 0.795 [rad/s]. Finalmente, c equivale a

$$c = 0.795 \cdot p = 0.795 \cdot 133.33 = 106.0$$

Finalmente, la ecuación dinámica que relaciona el voltaje aplicado y la velocidad angular queda de la siguiente manera:

$$\dot{\omega}(t) = -133.33 \cdot \omega(t) + 106.0 \cdot v(t) \quad (3.13)$$

3.2.3 Energía

El robot se alimenta a través de 2 baterías recargables tipo ión-litio. Las baterías se encuentran en serie y tienen una capacidad de 2500 [mAh], propias de una *tablet Samsung Galaxy Note*.

Para regular el voltaje de las baterías se emplean 3 reguladores, 2 de 5 [V] y uno de 3.3 [V]. Uno de los reguladores de 5 [V], identificado como el regulador de “potencia”, es el encargado de energizar los circuitos de potencia (motores y LEDs); los otros 2 reguladores son destinados para los circuitos lógicos y se encargan de alimentar sensores, microcontroladores y wifi.

A) Batería

Las baterías empeladas son de ión-litio (Figura 3.64-a), de una sola celda, con un voltaje nominal máximo de 3.7 [V] y corriente de 2500 [mAh]. Sus dimensiones físicas son: 67.50 [mm] × 51.90 [mm] × 6.10 [mm].

Para poder determinar la carga de las baterías se calculan los valores de consumo máximos de todos los elementos del robot. Para esto, la corriente de los LEDs y los motores se mide de manera experimental y el resto se obtienen de las hojas de datos de los elementos restantes.

En la siguiente tabla se muestran los valores máximos de corriente:

Tabla 3.8: Consumo de corriente máximo del robot.

Elemento	Cantidad	Corriente individual [mA]	Corriente total [mA]
Encóder	2	12	24
IMU	1	3.8	3.8
LEDs RGB	24	12	288
Microcontoladores	2	250	500
Motores	2	300	600
Ultrasónico	1	20	20
Wifi	1	160	160
Extras no contabilizados			200
Total			1795.8

La batería se elige a partir de:

- Un modelo comercial existente en el mercado que se adapte a las dimensiones del robot (véase sección C) [Soporte para las baterías](#)).
- Una carga eléctrica que sea mayor al consumo máximo del robot.

La batería elegida tiene una capacidad de suministro mayor a la requerida de:

$$c_s = \frac{I_{fuente}}{I_{m\acute{a}x\ consumo}} = \frac{2500 \text{ [mAh]}}{1795.8 \text{ [mAh]}} = 1.3921$$

$$c_s = 1.3921$$

Las baterías se conectan en serie para poder alcanzar el voltaje necesario para ser regulado a los voltajes de alimentación de 5 [V] y 3.3 [V]. En la Figura 3.64-b se aprecia el esquema de conexión en serie de las baterías.

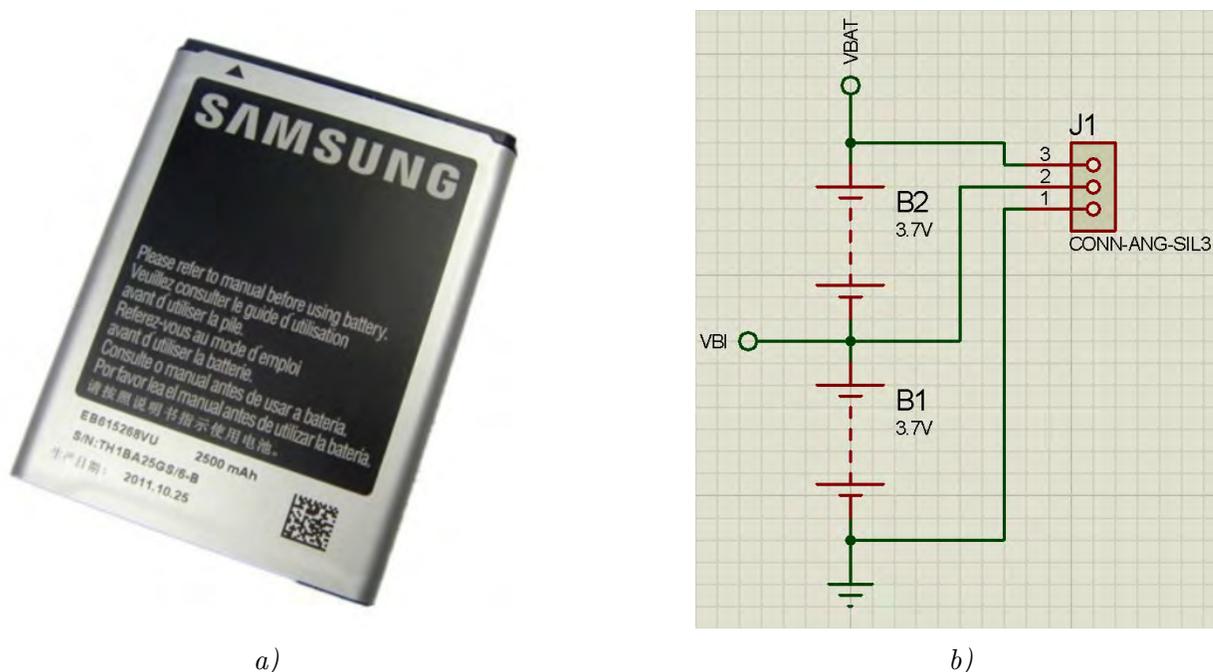


Figura 3.64: a) Batería empleada en el robot [9]; b) esquema de las baterías en serie del robot.

Típicamente, estas baterías no se deben dejar descargar a menos de 2.5 [V], por lo que el nivel de carga de las baterías se sensa a través de un circuito de comparadores (véase apartado C) Comparadores), para protegerlas de una descarga excesiva. No se implementó un circuito que recargue las baterías por lo que, una vez descargadas, las baterías se recargan de manera externa al robot.

B) Regulación de la alimentación

Los circuitos se alimentan tanto a 5 [V] como a 3.3 [V]. Para lograrlo, se regula el voltaje de las baterías que oscila en un rango de 6.0 [V] (carga mínima) a 7.4 [V] (carga completa).

Los modelos de los reguladores seleccionados son el IC *LM2940* para la salida de 5.0 [V] y el IC *LM3940* para la salida de 3.3 [V]. Todos los reguladores pueden soportar una corriente de hasta 1.0 [A].

El IC *LM2940* es un regulador positivo con una caída típica de voltaje de 0.5 [V] a corriente máxima, funcionando en un rango de 6.0 [V] a 26.0 [V], donde su alimentación se toma directamente desde la segunda batería en serie; su esquema típico se muestra en la Figura 3.65-a. El IC *LM3940* es un regulador diseñado para hacer la conversión de 5.0 [V] a 3.3 [V], donde su alimentación se toma desde el regulador lógico a 5.0 [V]; su esquema típico se muestra en la Figura 3.65-b.

Los capacitores que se muestran en los esquemas típicos son sugeridos y pueden sustituirse por los que requiera la aplicación. El capacitor que se conecta a la alimentación es un capacitor de filtro, necesario para el caso en que la alimentación sea a partir de una batería. Ese capacitor se

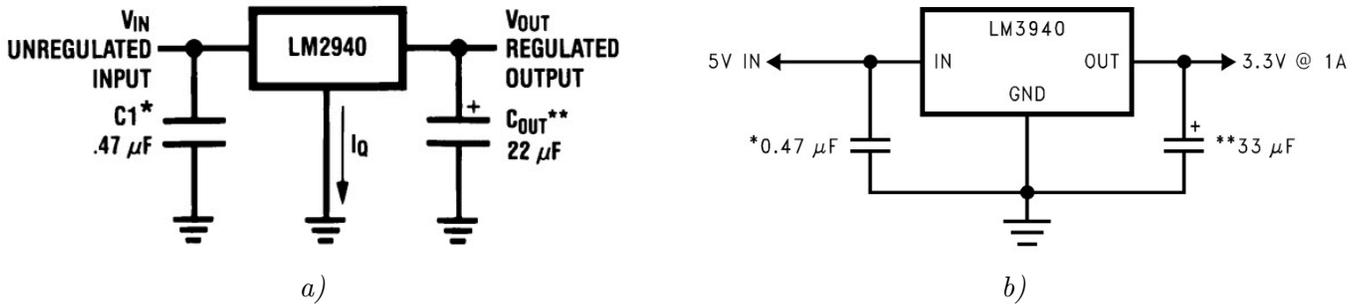


Figura 3.65: Esquemas típicos de los reguladores de voltaje a) a 5 [V] [77] y b) a 3.3 [V] [79].

mantiene como en el esquema típico, es decir, un capacitor cerámico de 470 [nF].

El capacitor de la salida regulada se aumenta en todos los reguladores para estabilizar la salida. Para el regulador de “potencia” se aumenta a 200 [μF] con 2 capacitores en paralelo; para el regulador “lógico” a 5.0 [V] se emplea uno de 100 [μF] y para el regulador “lógico” a 3.3 [V] se emplea uno de 47 [μF]. El esquema de los 3 reguladores se aprecia en la Figura 3.66.

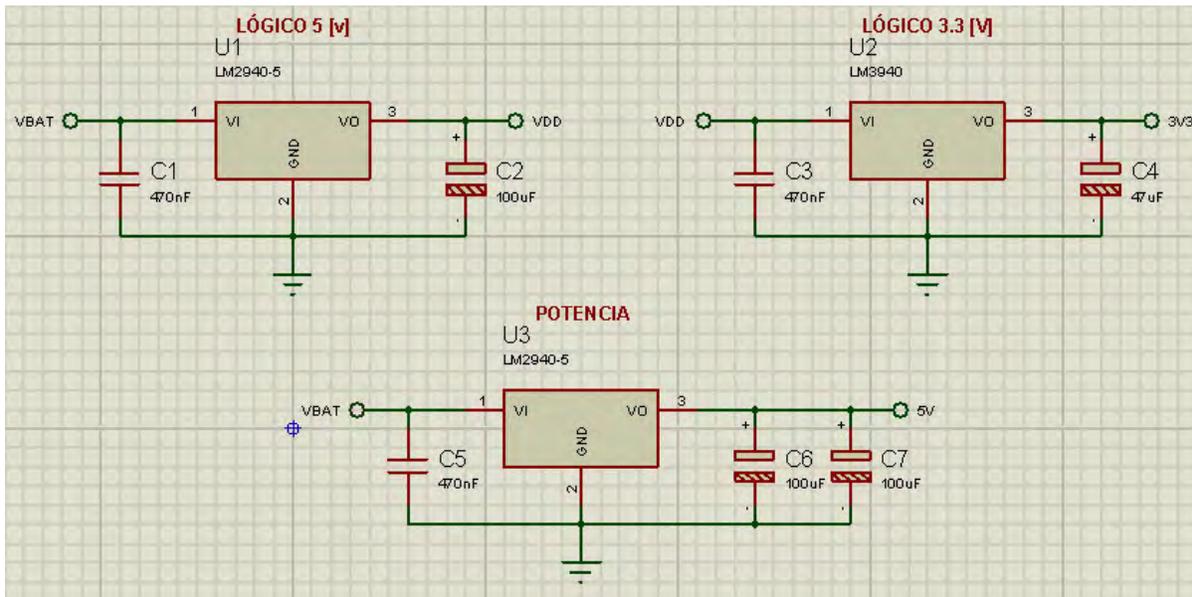


Figura 3.66: Esquema de los reguladores empleado para alimentar los circuitos del robot.

Encendido y apagado del robot

El robot cuenta con un botón de encendido/apagado. Es el botón *TL2201EEYA* (Figura 3.67-a), de dos polos. Es un *pushbutton* miniatura enclavado de dos posiciones (encendido/apagado). Tiene 6 pines y es capaz de manejar hasta 100 [mA] a 30 [V]. En la Figura 3.67-b se aprecia el esquema de posiciones del switch.

El botón *TL2201EEYA* fue seleccionado por ser pequeño, aunque no es capaz de ser empleado directamente para alimentar los circuitos del robot. Debido a ello, se combina con electrónica de potencia para alimentar al robot.

MOSFET de alimentación

Para alimentar al robot, el botón de encendido/apagado se combina con un MOSFET general

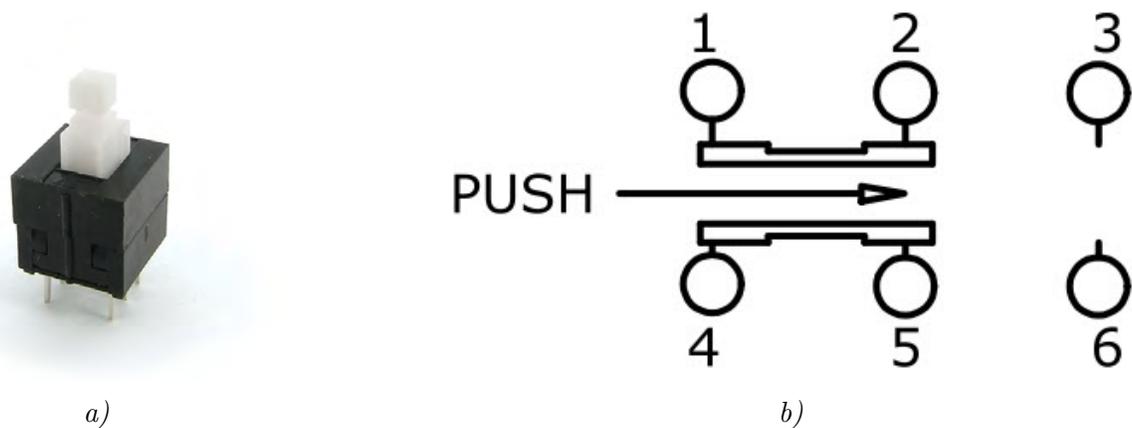


Figura 3.67: a) Botón **TL2201EEYA** de encendido/apagado del robot [19]; b) esquema de posición del switch del botón, se aprovechan únicamente 2 conexiones para el encendido/apagado [17].

de alimentación, cuya activación se da al presionar el botón a la posición de encendido. El MOSFET es un *DMN6040SSD*, capaz de tolerar corrientes de hasta 5.0 [A] a 60 [V].

El MOSFET *DMN6040SSD* es de tipo canal “N”, lo que significa que toda la carga se alimenta directamente con sus voltajes de alimentación propios, pero la tierra de todos los circuitos se conecta a la terminal drenadora (D) del MOSFET, dando como resultado una “tierra virtual”, que en lo sucesivo se le llamará tierra general o simplemente tierra; con un voltaje que varía de entre 0.0 [V] a 0.125 [V] (V_{DS}) dependiendo de la corriente que pasa por el MOSFET.

La terminal sustrato (S) se conecta a la tierra general del sistema, es decir, la terminal negativa de la primera batería. La compuerta (G) se conecta a una fuente de voltaje que active el MOSFET, se aterriza mediante un resistor de 470 [k Ω] a la tierra general del sistema, para “apagar” el MOSFET cuando el botón se encuentre en su posición de apagado. De acuerdo a las gráficas del datasheet del MOSFET, entre mayor sea el voltaje aplicado a la compuerta, hasta 20 [V], el voltaje V_{GS} disminuirá. El mayor voltaje que se le puede aplicar es el de las baterías en serie (6.0 [V] a 7.4 [V]).

Las terminales **2** y **5** se conectan a las terminales positivas de las baterías, que corresponden a los pines “comunes” del botón. Las terminales **3** y **6** se conectan al MOSFET de alimentación y a los MOSFET de sensado del nivel de carga de la batería (véase sección **C**) [Nivel de carga de la batería](#)). El esquema de la alimentación general con el botón se aprecia en la Figura 3.68. En el esquema, la **CARGA** representa los sensores (menos los sensores del nivel de carga de la batería), actuadores, microcontroladores y wifi, sin contar los reguladores.

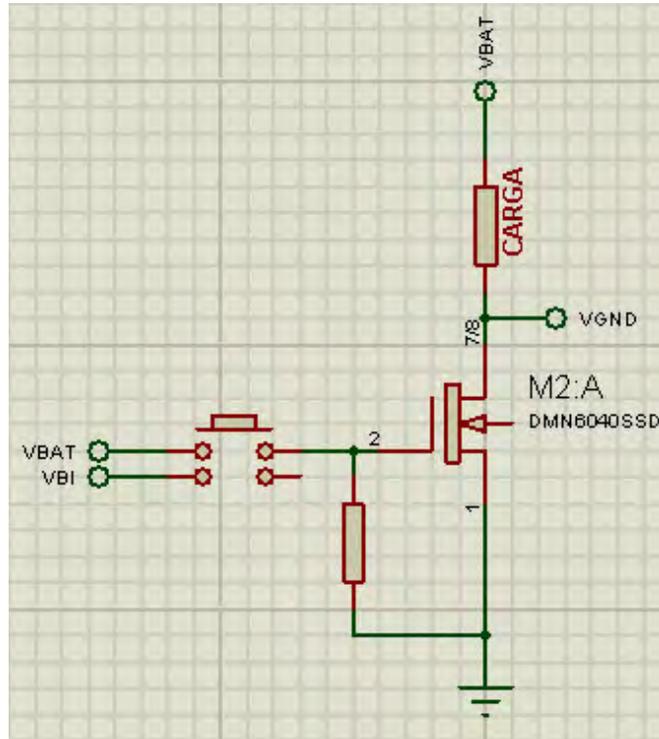


Figura 3.68: Esquema del botón y alimentación general.

3.2.4 Comunicación inalámbrica

La comunicación inalámbrica del robot está basada en el estándar IEEE 802.11, comúnmente conocido como **wifi**. Esto permite que pueda comunicarse con cualquier dispositivo que se encuentre conectado a una red basada en el protocolo TCP/IP.

El dispositivo elegido que realiza dicha tarea es el *RN171* de *Microchip*.

A) RN171

El módulo *RN171* (Figura 3.69) es un dispositivo de acceso inalámbrico LAN embebido. El dispositivo tiene integradas una pila TCP/IP y aplicaciones, y en su configuración más simple (a nivel de *hardware*) requiere únicamente de 4 pines (alimentación, transmisión, recepción y tierra). Una vez hecha la configuración inicial, el dispositivo accede automáticamente a una red wifi y envía y recibe información serial.

El *RN171*, al igual que modelos similares o variaciones de éste, se configura y controla mediante comandos ASCII.

Algunas de sus características principales son:

- Módulo LAN inalámbrico de 2.4 [GHz] IEEE 802.11 b/g.
- Empaquetado de montaje superficial.
- Configurable mediante un set de comandos ASCII
- La antena es externa, soporta antenas tipo PCB, chip y cable.



Figura 3.69: *Dispositivo embebido RN171, dispositivo wifi del robot que permite la comunicación inalámbrica [38].*

- Soporta protocolos de seguridad WEP, WPA-PSK y WPA-PSK2.
- Opera a un voltaje de 3.3 [V].
- Puede ser empleado como punto de acceso para una red de hasta 7 dispositivos.

Circuito

El *RN171* se alimenta directamente de la línea de energía de 3.3 [V]. Dado que el dispositivo opera a este voltaje, pero los microcontroladores operan a 5 [V], la línea de recepción del dispositivo se debe proteger mediante un arreglo de diodo Schottky y resistor para adecuar la señal de 5 [V] a 3.3 [V]. El microcontrolador se conecta al dispositivo únicamente mediante los pines TX y RX, es decir, los pines de comunicación serial RS-232, que es la configuración más simple disponible.

La hoja de especificaciones indica que la placa donde se suelde el dispositivo debe tener a contracara un plano de tierra excepto en un área rectangular cercana a la antena, de igual forma menciona que se deben de tener dos planos de tierra interconectados entre ambas caras de la placa.

La antena que se debe de conectar al dispositivo debe tener una impedancia de 50 [Ω]. Dada la complejidad que presenta el diseño de una antena tipo PCB se optó por emplear una antena de chip de 50 [Ω]. El circuito en cuestión se aprecia en la Figura 3.70.

9.- **save**: almacenar las configuraciones

10.- **reboot**: reiniciar el dispositivo

Tras reiniciar el dispositivo, se encontrará en modo AP (punto de acceso) con las configuraciones que se le hayan dado.

Para conectar el dispositivo a una red existente sólo basta con la configuración:

1.- **set wlan pass** *¡texto!*: dar la clave de acceso

2.- **join** *¡texto!*: dar el nombre de la SSID

En caso de que se hayan escaneado las redes disponibles mediante el comando *scan*, bastará con cambiar el comando *join* por el número de red que haya sido encontrado, por ejemplo: *join # 1* hará que el dispositivo se conecte a la red listada número 1; si la red tiene una clave de seguridad, se tiene que configurar antes de intentar conectar a la red.

Para poder realizar pruebas, uno de los robots se configura con una red abierta con SSID de *CU-10-80-T*.

3.2.5 PCBs

Las placas de circuito impreso (PCB, por sus siglas en inglés) diseñadas para el robot tienen soldados todos los componentes y conectores electrónicos mencionados anteriormente. Se diseñaron 3 placas: *Energía*, *Maestro* y *Esclavo*.

Todos los componentes empleados son de montaje superficial, para reducir al máximo posible el tamaño de los componentes y poder aprovechar todo el espacio en las PCBs. Existe una gran cantidad de tamaños, formas y empaquetados. Los empaquetados son lo suficientemente grandes para ser manipulados y soldados a mano.

La lista de empaquetados de los componentes es la siguiente:

- TQFP para los microcontroladores
- QFN para la IMU
- SOIC (8 y 14 pines) para la memoria, comparadores, MOSFET's, Schmitt triggers, Flip-Flop D
- PowerSO20 para el puente H
- SOT-223 para los reguladores de voltaje
- HTSSOP38 para el *LED driver*
- 1206 para resistores, jumpers y capacitores cerámicos y de tantalio
- SMA para los diodos de libre rodada
- SOD-123 para los diodos Schottky y Zener

- SMD (can) para los capacitores de aluminio
- XTAL-SMT personalizado para el cristal oscilador
- RN171 personalizado para el wifi
- Los conectores tienen un empaquetado personalizado presente en sus hojas de especificaciones

Las placas tienen dimensiones de 74×70 [mm], con huecos en los costados de 18×10 [mm] para permitir el paso de los cables en el caso de las placas de *Energía* y *Maestro* y un hueco de 50×16 [mm] para dar espacio al sensor ultrasónico en el caso de la placa del *Esclavo*. Todas las placas tienen un claro de 3 [mm] para poder ser montadas en los soportes del robot, donde no se colocan componentes aunque sí pistas.

En todas las placas se evitan a toda costa pistas en ángulo recto para evitar interferencia eléctrica o hasta funcionamientos similares a una antena [76]; las pistas se conectan en ángulos de 45° o directamente a planos de tierra.

A) Placa de *Energía*

Esta placa alberga los componentes que brindan la energía regulada o lógica y la de potencia. De esta placa se desprende el conector que distribuye la energía lógica a 5 [V] y a 3.3 [V] a las otras dos placas. También contiene el puente H y el LED driver, para no tener que distribuir el voltaje de potencia de 5 [V] a las demás placas.

En esta placa también se incluyen los comparadores del nivel de carga de las baterías y el MOSFET de alimentación, que se activa al encender el botón de alimentación al costado del robot.

Dado que esta placa se coloca en la repisa más cercana a las baterías, todo componente está soldado de un mismo lado de la placa, el anverso, dejando así el reverso de la placa libre para permitir el paso de pistas conectadas a vías. Sobre el anverso se sueldan todos los componentes.

En la Figura 3.71 se aprecia el anverso de la placa, en la Figura 3.72 se aprecia el reverso de la placa. Viendo a detalle el anverso se aprecia:

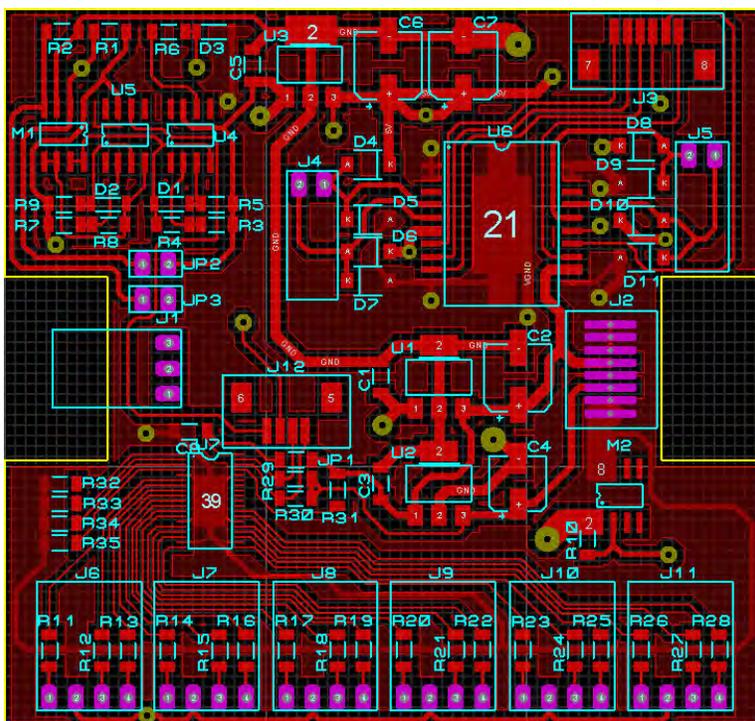


Figura 3.71: Anverso de la placa de energía. En esta cara se sueldan todos los componentes, que van desde la regulación de la energía hasta brindar la energía al puente H y al LED driver.

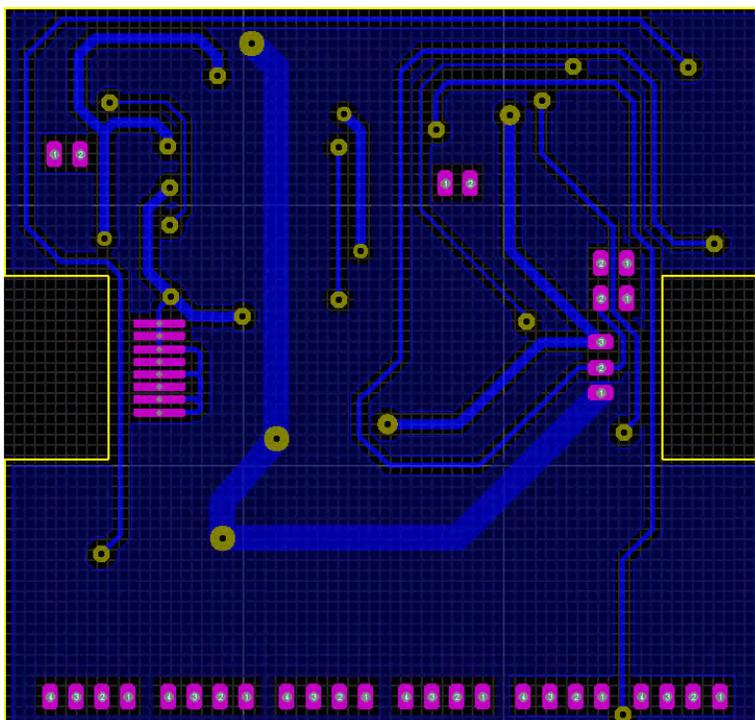


Figura 3.72: Reverso de la placa de energía, de este lado sólo se encuentran pistas empleadas para las vías, se aprecian diversos grosores de pistas que llegan hasta los 100 [th].

La distribución de componentes, misma que está indicada en la Figura 3.73 se describe a continuación:

- En la esquina superior izquierda se encuentran tanto los comparadores como el MOSFET de activación de los comparadores. Los comparadores están acompañados de los divisores de voltaje y los diodos Zener correspondiente a cada arreglo.
- Centro superior, se encuentra el regulador de potencia y dos capacitores, así como el puente H, un par de sus diodos de libre rodada y el conector para el motor izquierdo.
- En la esquina superior derecha se encuentran los otros diodos de libre rodada y un conector cuyo par se encuentra en la placa del *Esclavo* que brinda las señales de control para los motores. El segundo conector también para el motor derecho se encuentra pegado al costado.
- En la zona central se encuentran varios conectores, botón de encendido, baterías, alimentación a las otras placas y señales a la placa del *Esclavo*, también se aprecian los otros dos reguladores, los reguladores lógicos con sus respectivos capacitores.
- En la esquina inferior izquierda y centro inferior se encuentra el LED driver, sus resistencias de control del bus I²C, resistencia de corriente máxima y algunos de los conectores de LEDs con las resistencias limitantes por color.
- En la esquina inferior derecha se encuentran los últimos conectores y sus resistencias, así como el MOSFET de alimentación del robot.

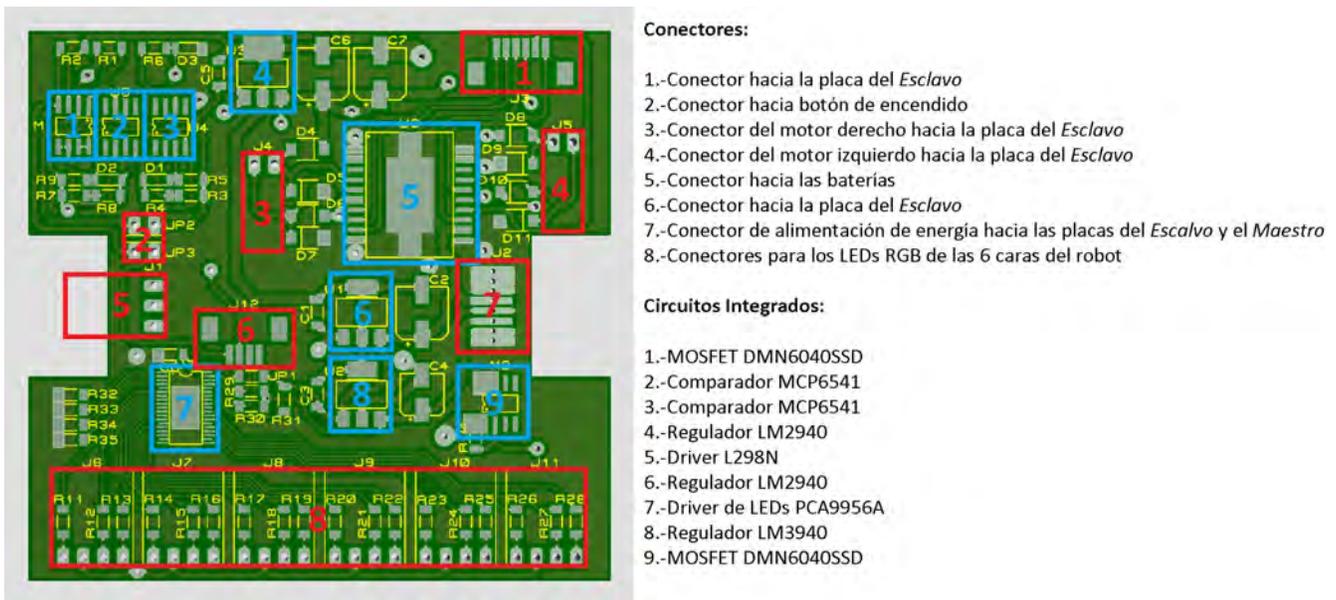


Figura 3.73: Distribución de los diferentes componentes en la placa de Energía.

Los conectores que no son de montaje superficial (de color blanco en la Figura 3.74-b) son conectores convencionales hembra de 100 [th] de separación en ángulo.

El grosor de las pistas varía; el LED driver tiene pistas de 10 [th], las pistas generales de señales 15 [th], las pistas de motores y cercanas a los bordes 20 [th], las pistas de los reguladores 40 [th] y, finalmente, la pista de tierra virtual y que hace conexión con la batería es de 100 [th].

El resultado de la fabricación de las placas se aprecia en las Figuras 3.74.

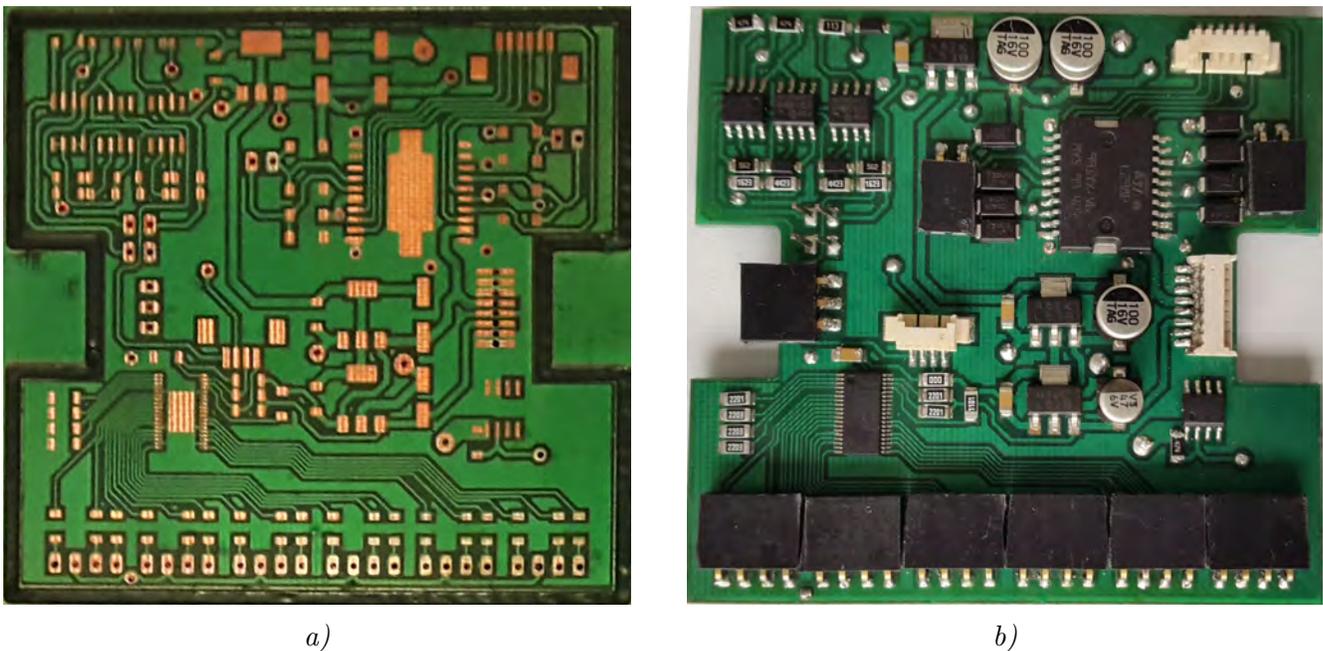


Figura 3.74: Resultado de la fabricación, a), y soldado, b), de la placa de Energía.

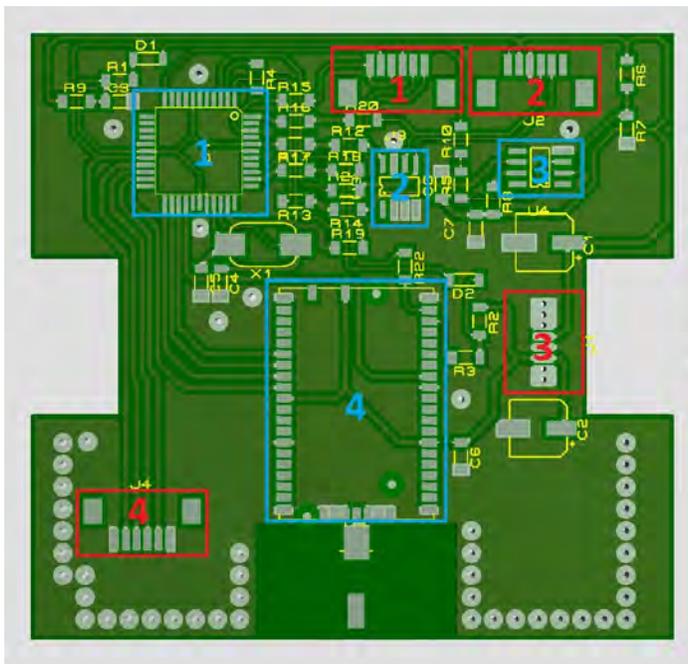
B) Placa del *Maestro*

Esta placa alberga el microcontrolador **Maestro**, el módulo wifi y la memoria. Se interconecta con la placa del *Esclavo* y la de botones; adicionalmente, tiene un conector extra para poder reprogramar el microcontrolador.

Al igual que la placa de *Energía*, la placa del *Maestro* se suelda en su totalidad de un mismo lado, sin embargo, esto es un requerimiento de la antena wifi, que requiere de un plano de tierra en el reverso de la placa; por lo mismo, hay una gran cantidad de vías de tierra que interconectan ambos planos de tierra tanto del anverso como del reverso.

En la Figura 3.75 se aprecia el anverso de la placa, mientras que en la Figura 3.76 se aprecia el reverso. Al estar todos los componentes soldados en el anverso de la placa, se aprecia que:

- En la zona superior de la placa se encuentra el **Maestro** con su respectivo oscilador, los componentes para programar el microcontrolador, la memoria y el comparador final que adecua el voltaje de la segunda señal; de igual forma se encuentran los conectores para la programación del microcontrolador y el conector que enlaza con la placa del *Esclavo*.
- En la zona centro e inferior se aprecia el módulo wifi, que ocupa un porcentaje considerable de la placa, y el conector que enlaza con la placa de botones.
- En el costado derecho se encuentra el conector de energía y capacitores de almacenamiento de energía.

**Conectores:**

- 1.-Conector hacia la placa del *Esclavo*
- 2.-Conector para el programador del microcontrolador *Maestro*
- 3.-Conector de alimentación de energía desde la placa de *Energía*
- 4.-Conector hacia los botones del robot

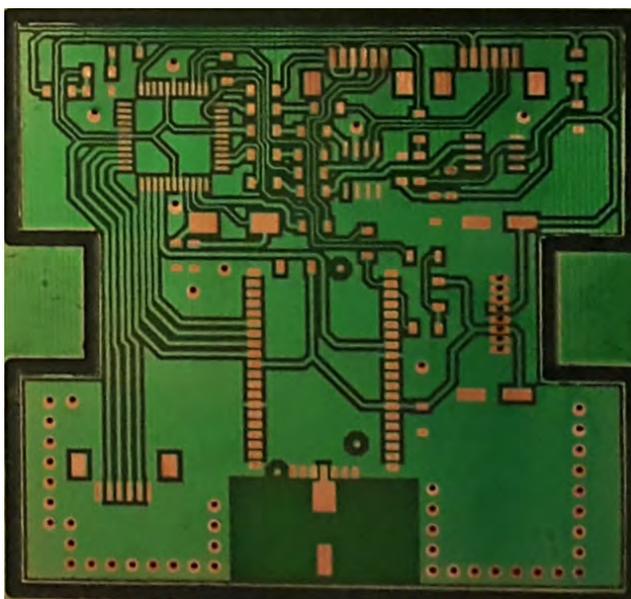
Circuitos Integrados:

- 1.-Microcontrolador dsPIC30F4011
- 2.-Memoria 24FC1026
- 3.-Comparador MCP6541
- 4.-Módulo WiFi RN171

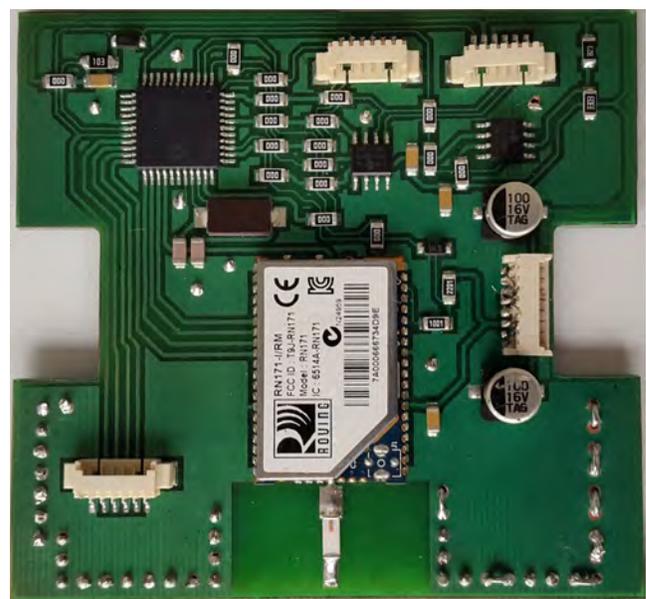
Figura 3.77: Distribución de los diferentes componentes en la placa del *Maestro*.

Hay varias pistas que pasan por debajo de *jumpers* (tienen forma de resistor de montaje superficial) cuyo grosor es de 10 [th]; el resto de las pistas, ya sea para señal o comunicación es de 20 [th]. Sin embargo, la pista que conecta la antena con el módulo wifi tiene un grosor de casi 3 [mm].

El resultado de la fabricación de las placas se aprecia en las Figuras 3.78.



a)



b)

Figura 3.78: Resultado de la fabricación, a), y soldado, b), de la placa del *Maestro*.

C) Placa del *Esclavo*

Esta placa alberga el microcontrolador **Esclavo**, el circuito QEI para el segundo motor, MOSFET's para controlar los sensores y varios conectores; en el reverso de la placa se encuentra la IMU y su circuito mínimo y capacitores de almacenamiento de energía. Se interconecta con la placa del *Maestro* y la de *Energía*; también tiene los conectores para el sensor ultrasónico y los motores.

En la Figura 3.75 se aprecia el anverso de la placa, mientras que en la Figura 3.76 se aprecia el reverso.

En el anverso de la placa se encuentran:

- En la zona superior se encuentran los conectores para reprogramar el microcontrolador, interconexión con la placa del *Maestro*, y dos conectores a la placa de *Energía*, uno que carga señales del bus I²C y otro que lleva las señales de control al puente H.
- Al centro de la placa se encuentra el microcontrolador **Esclavo** con su respectivo cristal oscilador, conector de energía y conector del sensor ultrasónico; MOSFET de control de sensores y varios *jumpers*.
- En la parte inferior se encuentran los conectores directos al encóder de los motores y el circuito QEI del segundo motor.

En el reverso de la placa se encuentran la IMU con su respectivo circuito mínimo (véase sección B) IMU) y los capacitores de almacenamiento de energía. La IMU se encuentra centrada en la placa para evitar desarrollar un algoritmo que corrija el valor obtenido al estar fuera del centro. De este lado también se encuentran los conectores que reciben el voltaje desde el puente H, cabe mencionar que la salida del puente H para los motores se conecta en este lado de la placa para no separar en dos conectores distintos la salida del encóder y la unión se conecta mediante pista de cobre.

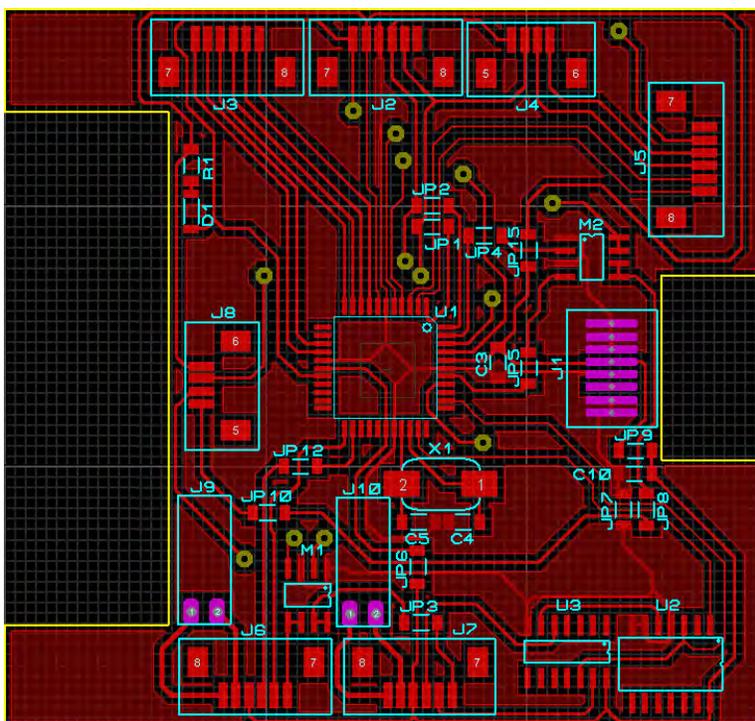


Figura 3.79: Anverso de la placa del Esclavo. En esta cara se sueldan el microcontrolador Esclavo, el flip flop D, el inversor ST y los conectores.

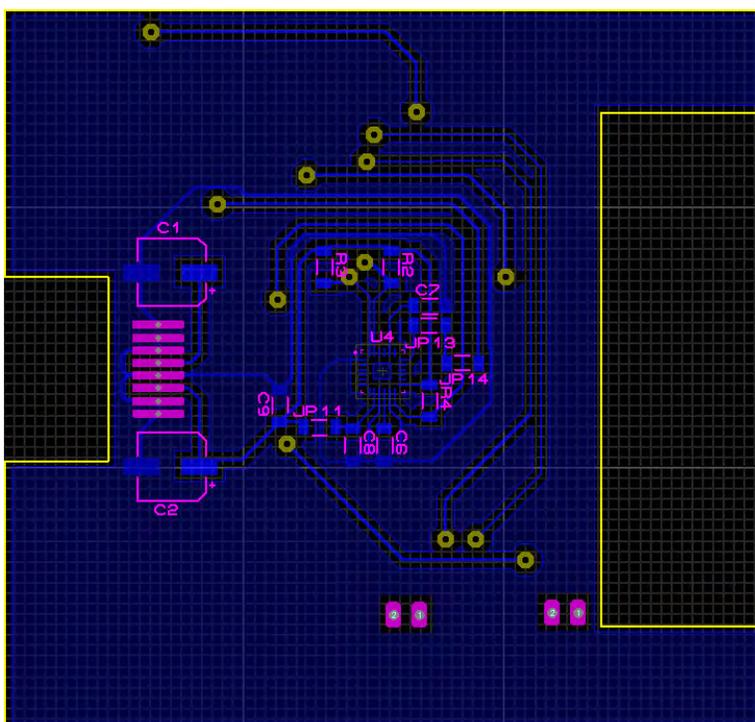
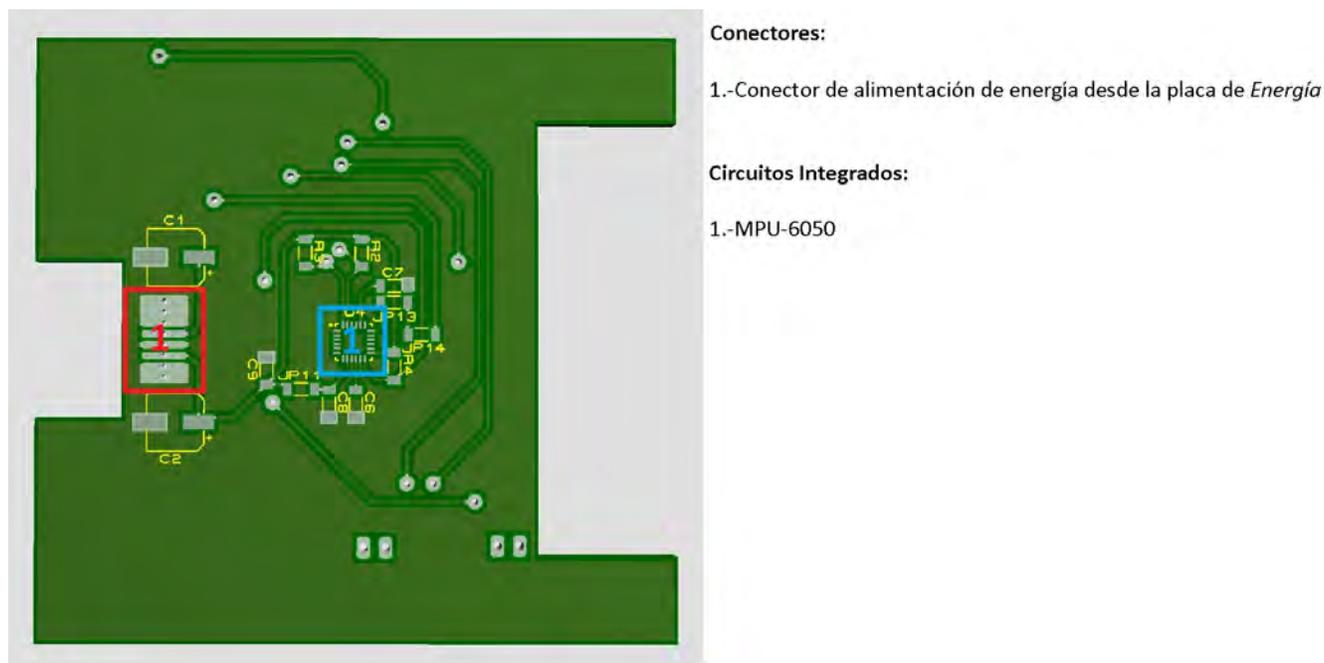
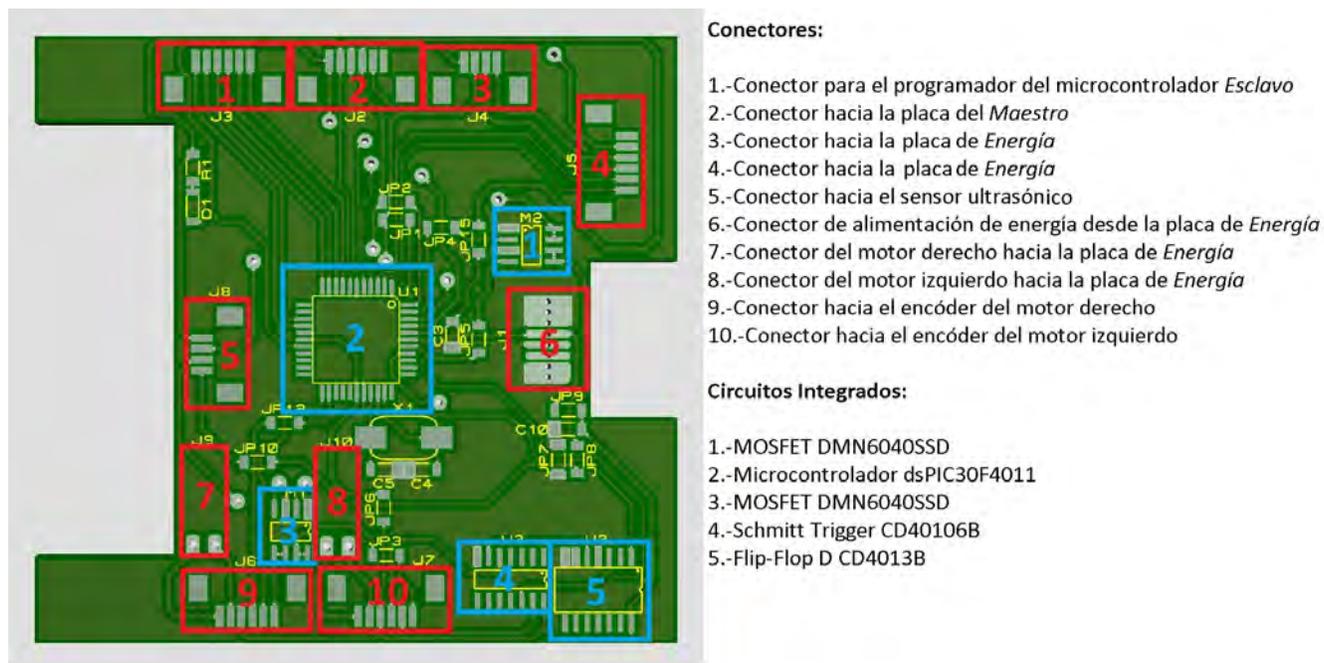


Figura 3.80: Reverso de la placa del Esclavo. En este lado se encuentran la IMU y su circuito mínimo.

Las pistas de menor grosor son las que conectan la IMU con el resto de circuitos, siendo de 10 [th], el resto de pistas de señales tienen un grosor de 15 [th].

La distribución de los diferentes componentes en la placa del **Maestro** se aprecian en las Figuras 3.81.



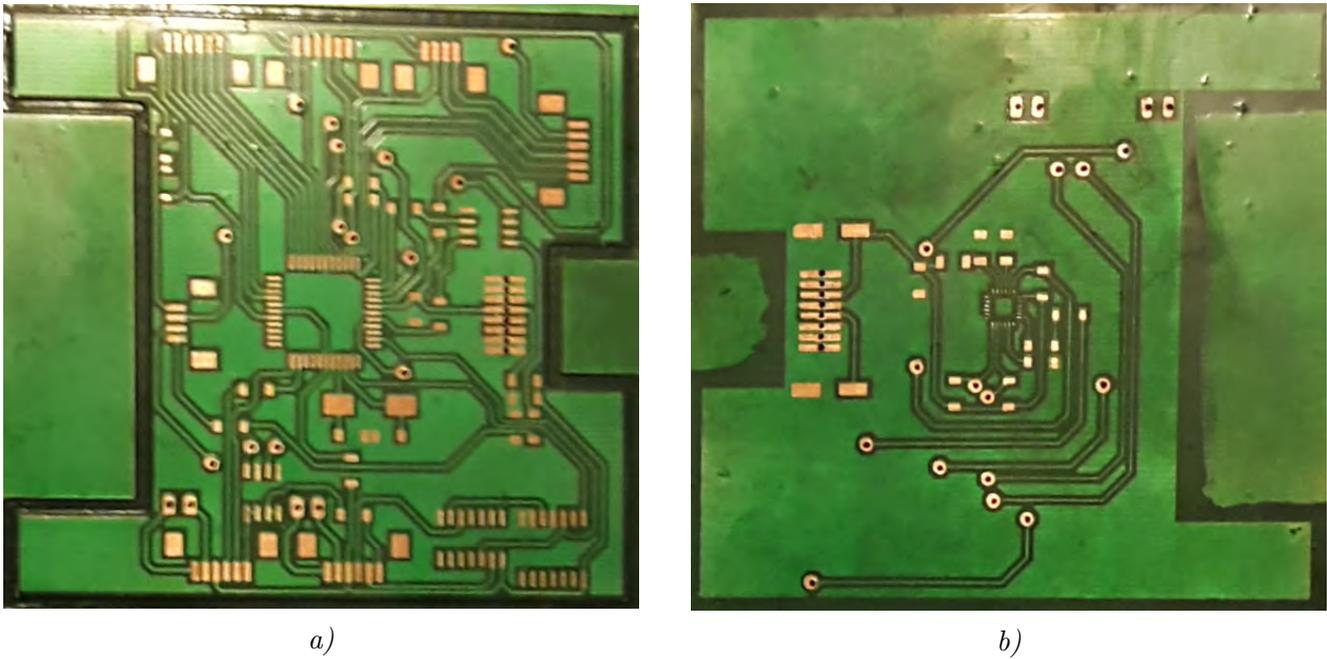


Figura 3.82: Resultado de la fabricación de la placa del Esclavo, a) anverso, b) reverso.

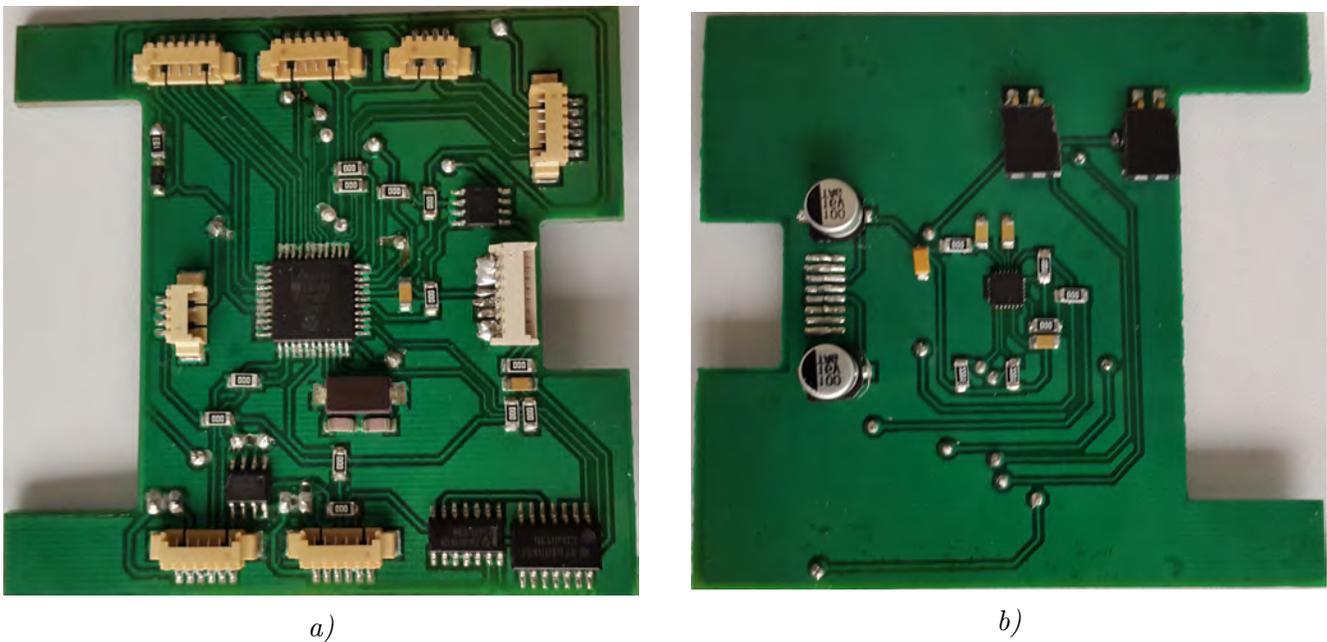


Figura 3.83: Resultado del soldado de la placa del Esclavo, a) anverso, b) reverso.

3.3 Microcontroladores

Los microcontroladores son el “cerebro” del robot. Tienen que encargarse de procesar, sensar, comunicar y controlar todos los eventos que suceden en el robot. El robot tiene que poder interactuar con sus elementos internos, con otros robots y con el usuario; debido a ello, las tareas de procesamiento del robot se separan en 2 microcontroladores: un microcontrolador *Maestro* y un microcontrolador *Esclavo*.

Las funciones del **Maestro** son las de almacenar en memoria e interpretar la ejecución de los programas guardados por el usuario con las acciones deseadas, comunicación con otros robots y el usuario en tiempo real y enviar y recibir información al Esclavo de lo que el robot hace y/o sensa.

El **Esclavo** tiene como funciones sensar y controlar todos los elementos físicos en el robot; toda la información de los sensores, el control de los motores y los LEDs lo ejecuta el Esclavo.

De cierta manera, se puede asemejar de una manera primitiva al **Maestro** como su sistema nervioso central y al **Esclavo** como su sistema nervioso periférico.

3.3.1 dsPIC30F4011

El DSC *dsPIC30F4011*, en lo futuro referido como dsPIC, es un microcontrolador de 16 bits que combina una máquina DSP simple. Capaz de manejar hasta 30 MIPS, cuenta con 48 [kB] de memoria flash para los programas, 2 [kB] de memoria RAM y 30 fuentes de interrupción. Cuenta con periféricos como múltiples pines tipo I/O, 5 timers, 2 módulos UART y uno I²C, 6 salidas de PWM, QEI, entre otros. Se programa a través de una terminal ICSP.

Existe en versiones de empaquetados tanto DIP de 40 pines como TQFP de 44 pines, por lo que es idóneo tanto para hacer pruebas en protoboard como para usarlo en las placas del robot. Es el DSC con más memoria y terminales PWM disponible en ambas versiones de empaquetado, trabajando a 5 [V]; en la Figura 3.84-a se aprecia el microcontrolador en empaque THT y en la Figura 3.84-b en su empaque SMT.

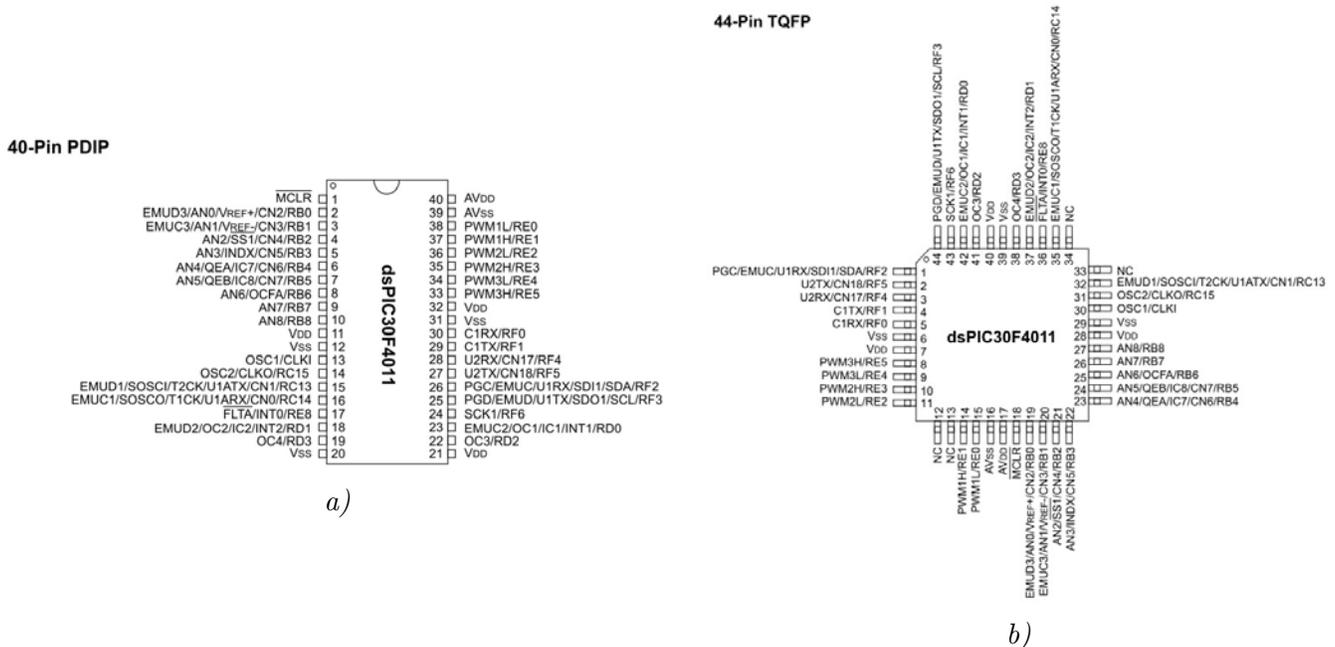


Figura 3.84: Esquema del dsPIC [34] en empaquetado a) PDIP, b) TQFP.

El dsPIC cuenta con un PLL (phase-locked loop, por sus siglas en inglés) que al combinarse con un oscilador o con una señal de reloj, alimenta el procesador del microcontrolador. La combinación máxima se alcanza con un oscilador de 7.5 [MHz] y el PLL configurado a $\times 16$, alcanzando una frecuencia interna de 120 [MHz]. A partir de eso, se contabiliza un ciclo máquina (CM) como 4 pulsos del oscilador principal, es decir, la frecuencia de trabajo del microcontrolador es

$$F_{CY} = \frac{OSC}{4} = \frac{120 \text{ [MHz]}}{4} = 30 \text{ [MHz]}$$

de ahí que su capacidad sea de hasta 30 MIPS.

Los periféricos que emplea el robot se enlistan a continuación, un resumen de la descripción de cada periférico se encuentra en el Anexo [C Periféricos del dsPIC](#) [31]:

- I/O: pines de propósito general
- I²C: comunicación serial I²C
- PWM: señal modulada de ancho de pulso
- QEI: decodificador de señales de cuadratura
- Timers: temporizadores con señales internas y externas
- UART: comunicación serial RS-232

3.3.2 dsPIC Maestro

El **dsPIC Maestro** se encarga de interactuar con el **Esclavo**, monitorear el estado de las baterías, enviar la información por el wifi y la lectura y procesamiento de comandos almacenados en memoria.

Su función principal es la de interpretar los comandos almacenados en la memoria *24FC1026* y transmitir las órdenes al **dsPIC Esclavo** tanto para acción como para transmisión de datos.

Se comunica con los diferentes componentes de empleando lo siguientes pines o protocolos:

- **Esclavo**: RS-232 (UART 2)
- Wifi: RS-232 (UART 1)
- Memoria *24FC1026*: I²C
- Nivel de baterías: I/O
- Botones de propósito general: I/O

El **Maestro** se encuentra en la misma placa que la memoria y el wifi, se interconecta con el **Esclavo**, botonera y sensores de batería a través de diferentes cables. Emplea un oscilador de 7.5 [MHz] conectado a las terminales **OSC1** y **OSC2** y capacitores de 22 [pF] en conjunto con un PLL interno que multiplica la señal $\times 16$ que le permite alcanzar los 30 MIPS de procesamiento. Su esquema electrónico se aprecia en la Figura 3.85.

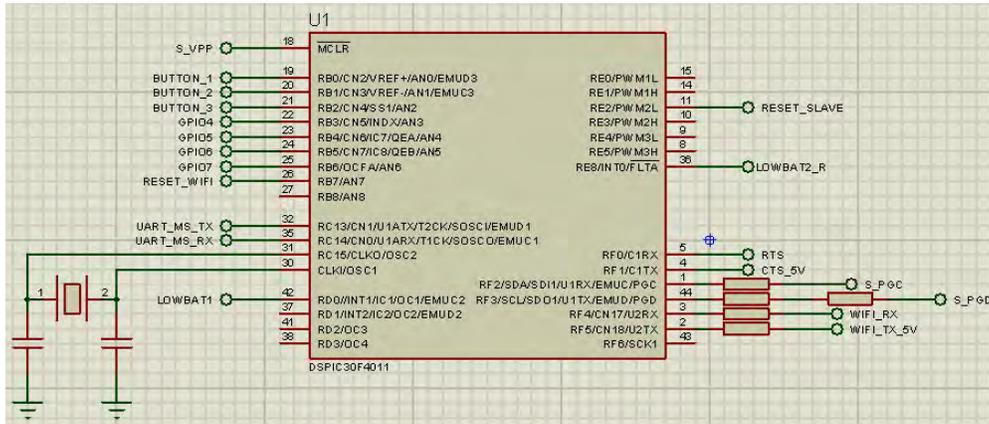


Figura 3.85: Esquema electrónico del dsPIC Maestro.

El **Maestro** tiene un set de comandos muy reducido dado que el set de comandos final se debe desarrollar en conjunto con la interfaz de usuario que permita ampliar la capacidad de procesamiento de tareas del maestro, facilitar la reprogramación de la memoria y el wifi y ampliar el número de indicadores que brinda el robot.

A) Encendido del Maestro

El **Maestro** al encenderse debe revisar en principio que exista batería disponible; en caso de que alguna de las terminales de batería (**LOWBAT1** o **LOWBAT2**) se encuentre en '1' lógico procederá a despertar al **Esclavo** para mostrar un par de parpadeos de los LEDs del robot en color rojo, ese será el indicativo de que el robot se ha quedado sin batería.

En caso contrario, el **Maestro** habilitará el wifi y despertará al **Esclavo**, posteriormente leerá las instrucciones que debe realizar el **Esclavo** desde la memoria y las transmitirá al **Esclavo** a través del puerto serial. Los pines de entrada **LOWBAT** se configuran como interrupciones externas para evitar estar poleando nuevamente el estado de las baterías; en caso de que alguna batería se descargue, mandará la señal lógica y el **Maestro** procederá a apagar el wifi y mandar los parpadeos en rojo al **Esclavo**.

El **Maestro** se comunica con el **Esclavo** al baudaje máximo posible (1875 [kbps]) por el puerto serial; en contraste, dado que no se tienen disponibles las líneas **RTS** y **CTS** conectadas al wifi, sólo se puede comunicar con él a 115200 [baud].

En la Figura 3.86 se aprecia el diagrama de flujo de encendido del **Maestro**.

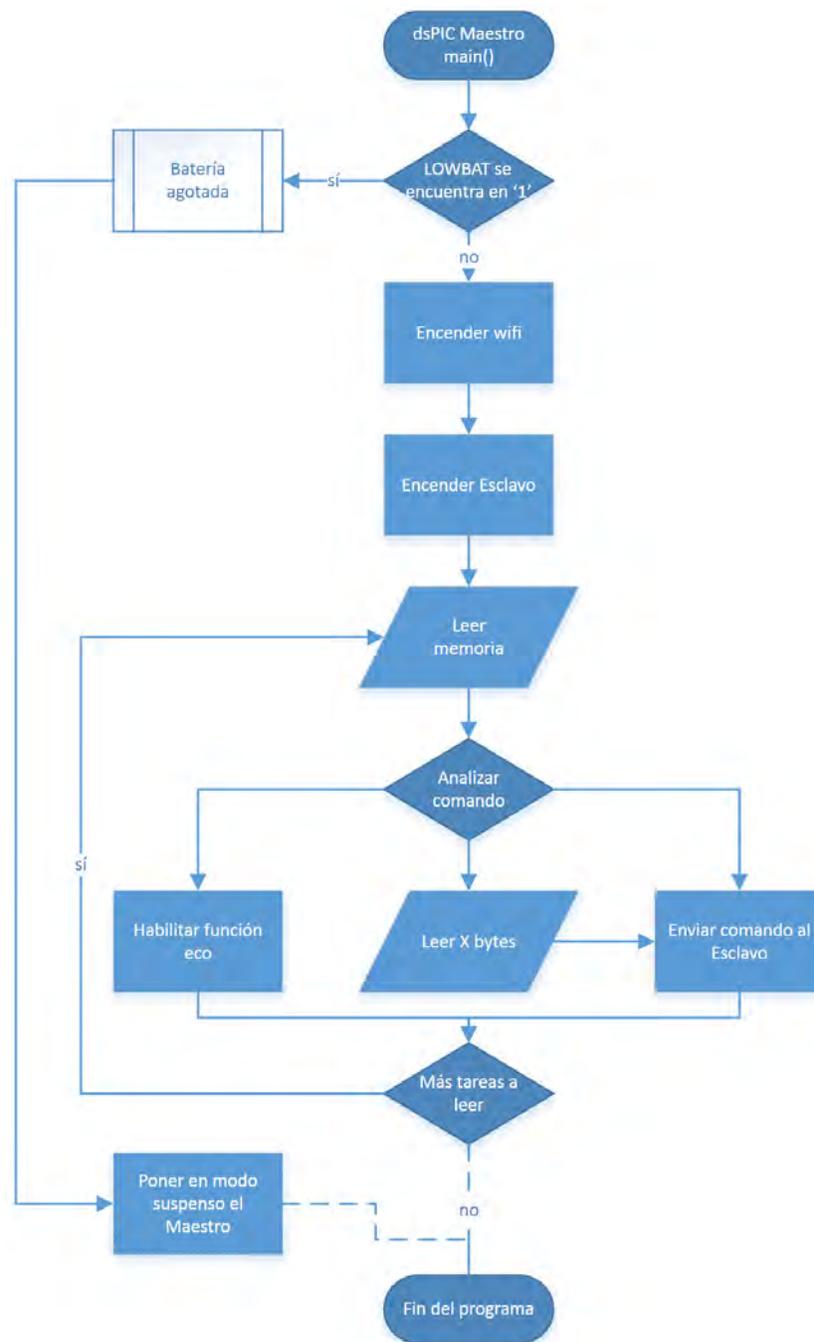


Figura 3.86: *Diagrama de flujo de encendido del Maestro; en caso de que haya batería suficiente procederá a encender todos los dispositivos.*



Figura 3.87: Diagrama de flujo de la función de parpadeo para indicar que la batería se ha agotado.

B) *Firmware* del Maestro

Interactividad mínima con el usuario usuario

El robot cuenta con unas cuantas señales que indican los estados mínimos del robot, tales como la disminución de la batería, el encendido correcto y conexiones con el wifi.

Todas se basan en el empleo de los LEDs del robot. Con cada evento el **Maestro** envía los comandos al **Esclavo** de un color a todas las caras del robot.

La falta de batería se refleja con una serie de 5 parpadeos en color rojo, posterior a eso el **Maestro** procederá a apagar el robot para evitar consumir más energía.

Cuando se enciende el robot, en caso de que no se haya agotado la batería y el robot haya realizado las configuraciones iniciales, dará un parpadeo largo de 1 [s] en color verde. Posteriormente procederá a leer las instrucciones almacenadas en memoria.

Cada conexión realizada vía wifi, ya sea porque algún robot se encuentra configurado como punto de acceso o simplemente se conectó a la red a la que está configurado hacerlo, lo reflejará dando un parpadeo largo de 1 [s] en color azul. Por el contrario, la desconexión se verá reflejada en color amarillo.

La escritura a la memoria se realiza manteniendo presionados los 3 botones en la parte superior durante 5 [s]. Al hacerlo el robot dará el parpadeo largo en color azul y el **Maestro** configurará el wifi como AP con una configuración abierta con un SSID “*Memory_write*”. Al realizar la conexión al punto de acceso, toda información enviada será escrita a la memoria. El proceso de escritura termina al enviar un caracter de salto de página (`\n`). Durante este periodo se puede reconfigurar el wifi empleando los comandos propios del dispositivo.

Funcionamiento interno del Maestro

Para leer la memoria, el **Maestro** se comunica mediante el protocolo I²C, cuyos diagramas de flujo generales se aprecian en el las Figuras 3.106, 3.92 y 3.94. En caso de que algún caracter común con el **Esclavo** sea leído, se mandará ese caracter de instrucción al **Esclavo** por su puerto serial.

Los caracteres disponibles para interactuar con el **Esclavo** son:

- ‘A’: Encender o apagar el acelerómetro.
- ‘G’: Encender o apagar el giroscopio.
- ‘U’: Encender el sensor ultrasónico.
- ‘T’: Apagar el sensor ultrasónico.
- ‘E’: Encender los encoders.
- ‘W’: Apagar los encoders.
- ‘B’: Alternar el sentido de giro de motores.
- ‘M’: Encender los motores.

- ‘N’: Apagar los motores.
- ‘L’: Mandar velocidad de motor izquierdo.
- ‘R’: Mandar velocidad de motor derecho.
- ‘1’ a ‘6’: Encender de algún color alguna cara del robot.

El detalle de las funciones específicas de cada acción del **Esclavo** se encuentran descritas en la sección [3.3.3 dsPIC Esclavo](#).

En casos especiales donde el **Maestro** deba leer más de un caracter a enviar al **Esclavo**, se antepone el número de bytes a leer de forma hexadecimal y no en formato ASCII. De esta manera, se limita que el rango de caracteres ASCII disponibles para interactuar con el **Esclavo** consista en los números ‘0’ al ‘9’ y letras (mayúsculas y minúsculas).

En caso de que al leer la memoria el **Maestro** se tope con el caracter ‘C’, activará una función de eco que mandará toda cadena que mande el esclavo por el wifi. Si se encuentra con el caracter ‘S’ terminará de leer el programa de la memoria.

3.3.3 dsPIC Esclavo

El **dsPIC Esclavo**, llamado **Esclavo** únicamente para referencias futuras, es el microcontrolador que se encarga de obtener la información de todos los sensores mediante sus diferentes protocolos, controlar a los actuadores y comunicarse con el **Maestro** para recibir órdenes de sentido, actuación y compartir información del estado actual con el **Maestro**.

El **Esclavo** se vale de diferentes protocolos para comunicarse con los elementos con los que interactúa, que son:

- **Maestro**: RS-232 (UART 2)
- Encoders: QEI para el encóder izquierdo, Timer 1 para el encóder derecho; Timer 3 para la frecuencia de muestreo
- IMU: I²C
- Ultrasónico: I/O
- LED driver: I²C
- Motores: PWM para la velocidad, I/O para la dirección
- MOSFETs: I/O; cuando el pin tiene valor de ‘1’ enciende el MOSFET, cuando es ‘0’ lo apaga.



Figura 3.89: Diagrama de flujo de la configuración inicial del **Esclavo**.

El módulo UART se configura al baudaje máximo posible de 1875 [kbps], dado que ambos microcontroladores están configurados con el mismo oscilador y PLL. El módulo I²C se configura a la velocidad “fast” de 400 [kHz], que es la velocidad máxima que soporta la IMU, que al ser el dispositivo de menor baudaje, limita la velocidad del bus I²C.

B) Configuración y uso de los encoders

La lectura de los encoders se divide en dos diferentes protocolos dado que el dsPIC sólo contiene un módulo QEI, sin embargo, ambos encoders se leen a una frecuencia de 20 [Hz]. El QEI tiene la capacidad de sensar los encoders a una resolución de hasta $\times 4$, el circuito decodificador empleando el FF-D sólo alcanza una resolución $\times 1$.

El módulo QEI tiene la capacidad de sensar los pulsos y el sentido de rotación del motor, al que se conectó el motor izquierdo. La configuración del módulo se hace para tener la resolución máxima de $\times 4$, con reinicio a través de alcanzar el valor de **MAXCNT**, que se configura con el valor máximo de `0xFFFF` para tener todo el contador disponible. Al módulo se conectan únicamente las salidas de los encoders del motor, al no ser necesario ni existir una señal de referencia. Las fases se conectan a los pines **QEA** y **QEB**.

Para el motor derecho se emplea el circuito decodificador con el FF-D. Del circuito se desprende una señal de reloj, la cual se conecta al pin **TICK**, que corresponde al Timer 1 (tipo A) capaz de recibir señales externas. La segunda señal es una señal lógica de un bit, conectada a una entrada del pin **RC13**, que indica el sentido de giro. Los pulsos se acumulan en el registro del timer, que funciona como contador, y el sentido de giro se ajusta cada lectura de los encoders.

La lectura de los encoders se realiza a una frecuencia de 20 [Hz], es decir, cada 50 [ms]. Para activarla, el **Maestro** debe enviar el caracter ‘E’. El **Esclavo** entonces enciende los MOSFETs que alimentan los encoders, donde las compuertas están conectadas a los pines de salida **RF0** y **RF1**. Posteriormente, configura el módulo QEI, configura el Timer 1 para recibir pulsos externos y lo enciende y, finalmente, configura el Timer 3 para lanzar la interrupción cada 50 [ms]. En la Figura 3.90 se aprecia el diagrama de flujo de la configuración.

Una vez encendido, el Timer 3 lanza una interrupción cada 50 [ms]; cuando se da la interrupción, se hace la lectura de los encoders. El valor se obtiene de los registros correspondientes: **POSCNT** para el encóder izquierdo (valor ya signado) y **TMR1** para el encóder derecho; el signo se obtiene de polear el pin de entrada de la otra fase (‘0’ signo negativo, ‘1’ signo positivo).

Ahora que se tienen los valores de pulsos, el valor se convierte a su equivalente en velocidad angular, mediante las fórmulas de la Ecuación 3.2.

En caso de que esté habilitado el control de velocidad, se procede en este momento a realizarlo, de lo contrario, el proceso se detiene en este momento y se da por terminado el proceso de lectura de los encoders. En la Figura 3.91 se aprecia el diagrama de flujo de la lectura.



Figura 3.90: *Diagrama de flujo de la configuración de los encoders.*

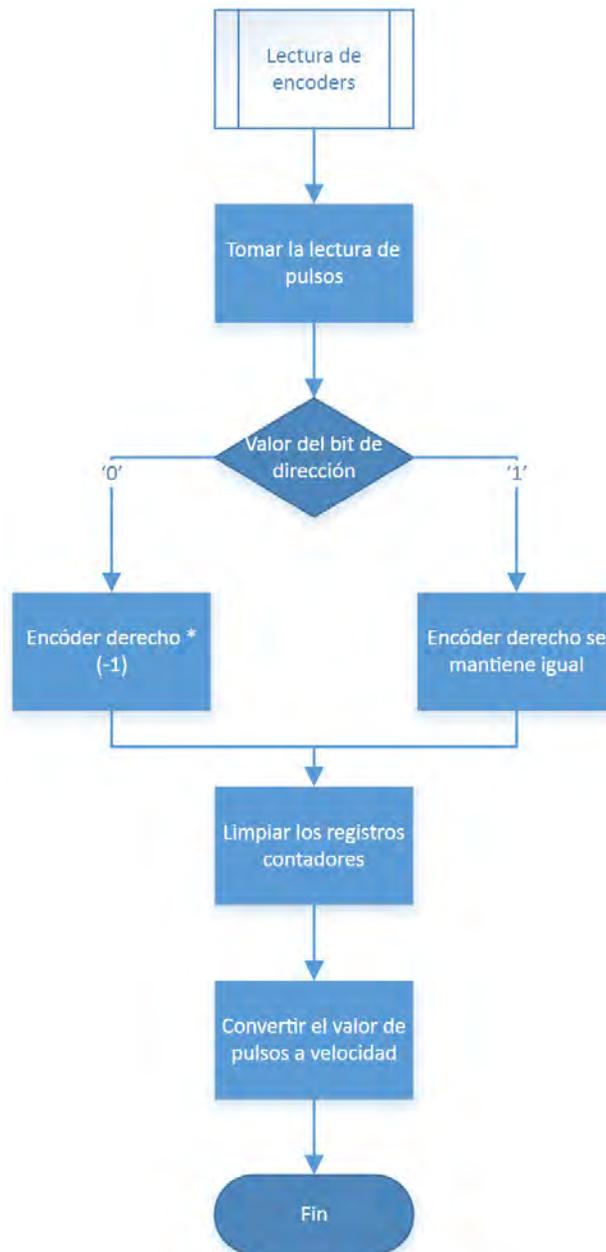


Figura 3.91: Diagrama de flujo de la lectura de los encoders.

C) Configuración y uso de la IMU

La IMU tiene un termómetro embebido dentro del empaquetado que sirve para monitorear la temperatura interna del sistema, por lo que la IMU se enciende desde un principio con tal fin. La comunicación con la IMU se realiza totalmente a través del bus I²C, a través de protocolos definidos de escritura y lectura. En la Figura 3.92 se muestra el diagrama de flujo de escritura de un registro por el bus I²C.

Primero se enciende la IMU a través del MOSFET, una vez encendida se le da un tiempo de asentamiento de 550 [μs] para permitir el encendido del módulo I²C de la IMU. Después del tiempo de asentamiento se configuran de manera independiente los registros mencionados en la descripción de la IMU (véase subsección B) IMU). En la Figura 3.93 se aprecia el diagrama de flujo de encendido de la IMU.

La IMU se lee a una frecuencia de 20 [Hz]. Cada que la IMU tiene información nueva lanza una interrupción a través del pin INT1. Cuando eso ocurre, el **Esclavo** revisa si el bus I²C no se encuentra en uso, en caso contrario retornará a la acción de lectura cuando el bus se libere. Cuando el bus está libre, el **Esclavo** inicia la comunicación a través del bus I²C, direccionando a la IMU dentro del bus, en modo de lectura. En la Figura 3.94 se aprecia el diagrama de flujo de la lectura en serie de registros.

El número de registros a leer en serie depende de la configuración que el **Maestro** le haya mandado al **Esclavo**, de tal forma que el **Esclavo** puede o no leer los 6 registros del acelerómetro, leerá los 2 registros del termómetro y puede o no leer los 6 registros del giroscopio; por ende, el **Esclavo** leerá desde 2 hasta 14 registros en serie.

El registro inicial y el número de registros a leer se configura cuando el **Maestro** manda al **Esclavo** los caracteres ‘A’ y ‘G’ para el acelerómetro y giroscopio, respectivamente; el enviar cualquier caracter desencadena una interrupción por recepción en el UART. Dependiendo del estado actual en que se encuentre el modo de lectura del acelerómetro y giroscopio, lectura o no lectura, el mandar el caracter alternará el estado del sensor de la IMU correspondiente.

Cuando se envía el caracter ‘A’ para habilitar la lectura se escribe en un registro de tipo *byte* el primer registro que se debe leer, que corresponde al registro ACCEL_XOUT (0x3B); de igual forma, se actualiza el número de registros a leer, sumando 6 registros al valor total de registros a leer. En el caso contrario, se regresa como registro índice el TEMP_OUT (0x41), y se reduce en 6 el número de registros a leer. Finalmente se actualiza el valor de una bandera indicativa del estado de lectura del acelerómetro.

Para el caso del caracter ‘G’, sólo se actualiza el número de registros a leer, dado que esos registros van a continuación de los registros de salida del termómetro, por lo que al encenderlo sólo se suma 6 al valor de registros a leer y cuando se apaga se restan 6 al valor de registros a leer. Finalmente se actualiza el valor de una bandera indicativa del estado de lectura del giroscopio. En la Figura 3.95 se aprecia el diagrama de flujo de configuración de ejes a leer.

Cuando el **Esclavo** ha terminado de leer los registros de la IMU, mismos que almacena en un buffer de RAM, tiene que reconstruir los datos leídos de la IMU. Para reconstruirlos toma los datos leídos por pares, montándolos en una palabra signada para después aplicar las operaciones matemáticas pertinentes, mismas que se aprecian en las fórmulas 3.6, 3.7 y 3.8 para el acelerómetro; 3.9 para el termómetro y 3.3, 3.4 y 3.5 para el giroscopio (véase subsección B) IMU). En la Figura 3.96 se aprecia el diagrama de lectura de la IMU.

Los datos reconstruidos son almacenados en variables *RAW*, es decir, el dato como fue leído, sin filtrar. Actualmente sólo el giroscopio tiene un protocolo simple de filtrado, que consiste en

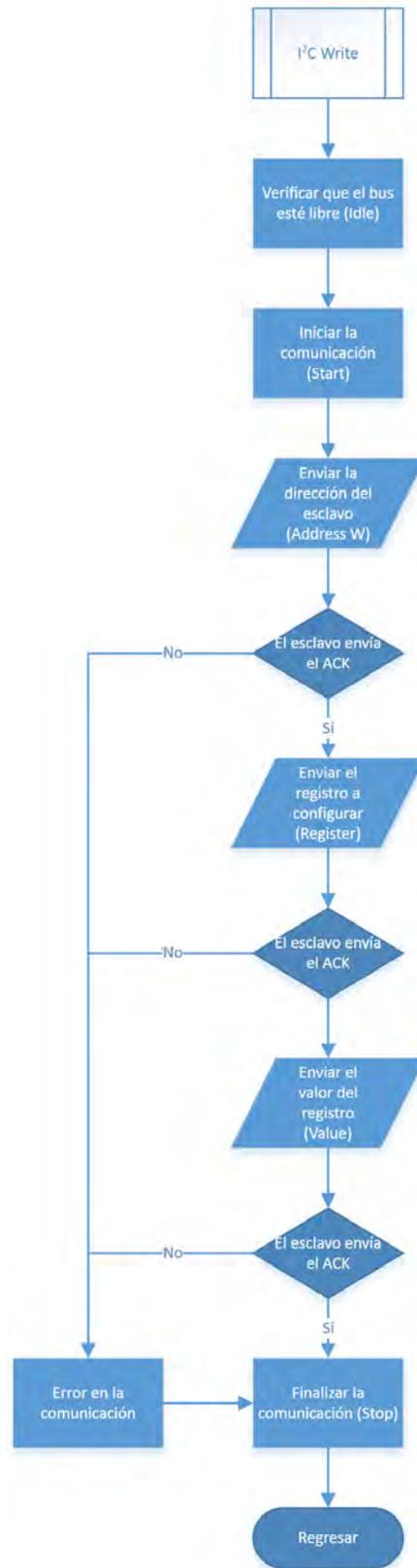


Figura 3.92: Diagrama de flujo de escritura en el bus I²C.

eliminar el error permanente de los giroscopios, para el cual cuando se encienden, se debe dejar el robot sin mover durante 5 segundos, tiempo en el que almacena 100 lecturas para generar ese



Figura 3.93: Diagrama de flujo de configuración de la IMU.

valor de error permanente por eje.

El error permanente es un promedio de las 100 lecturas obtenidas, el cual se resta a los valores obtenidos de la lectura de los giroscopios de la IMU. Estos valores se almacenan en variables *FILTERED* (filtradas).

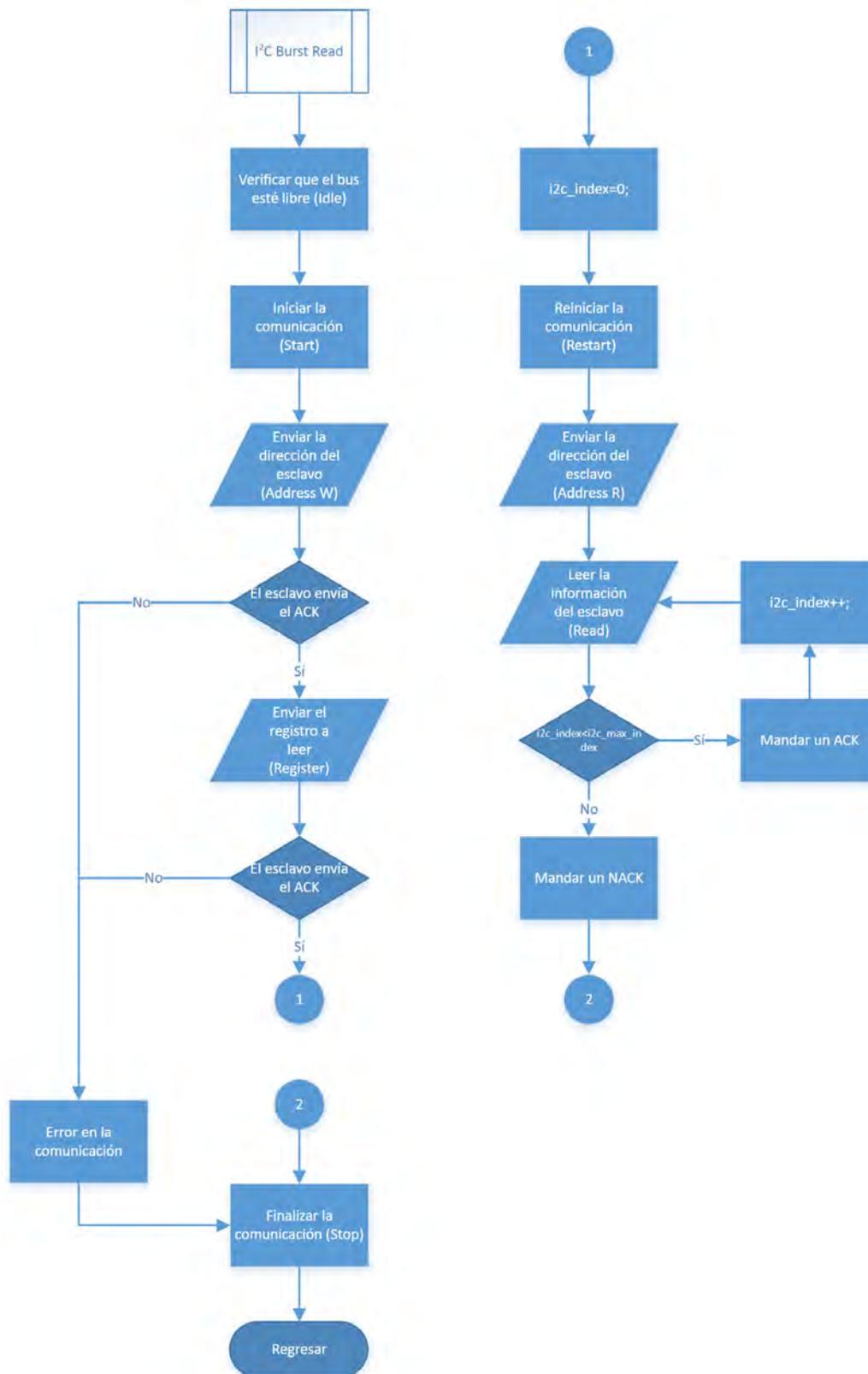


Figura 3.94: Diagrama de flujo de lectura en serie de registros.

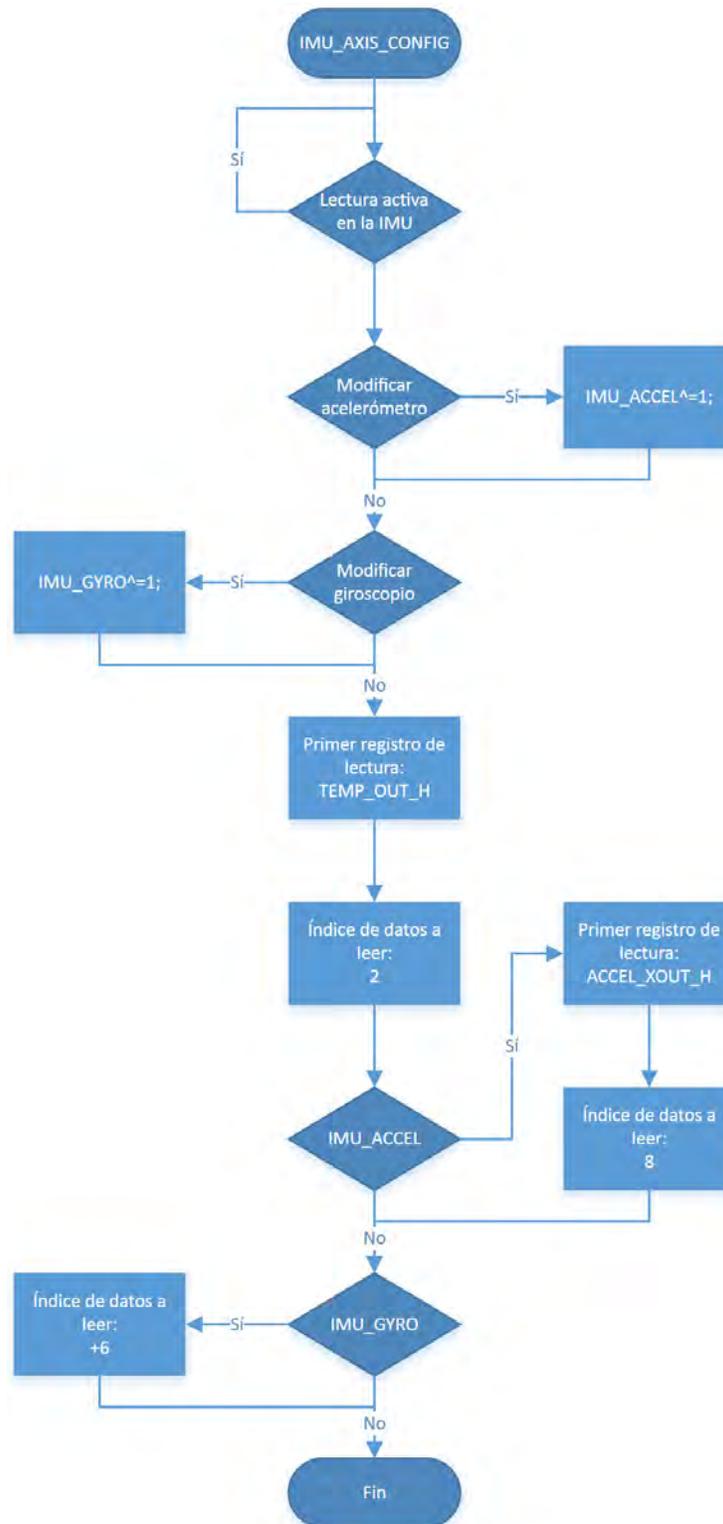


Figura 3.95: Diagrama de flujo de configuración de ejes a leer.

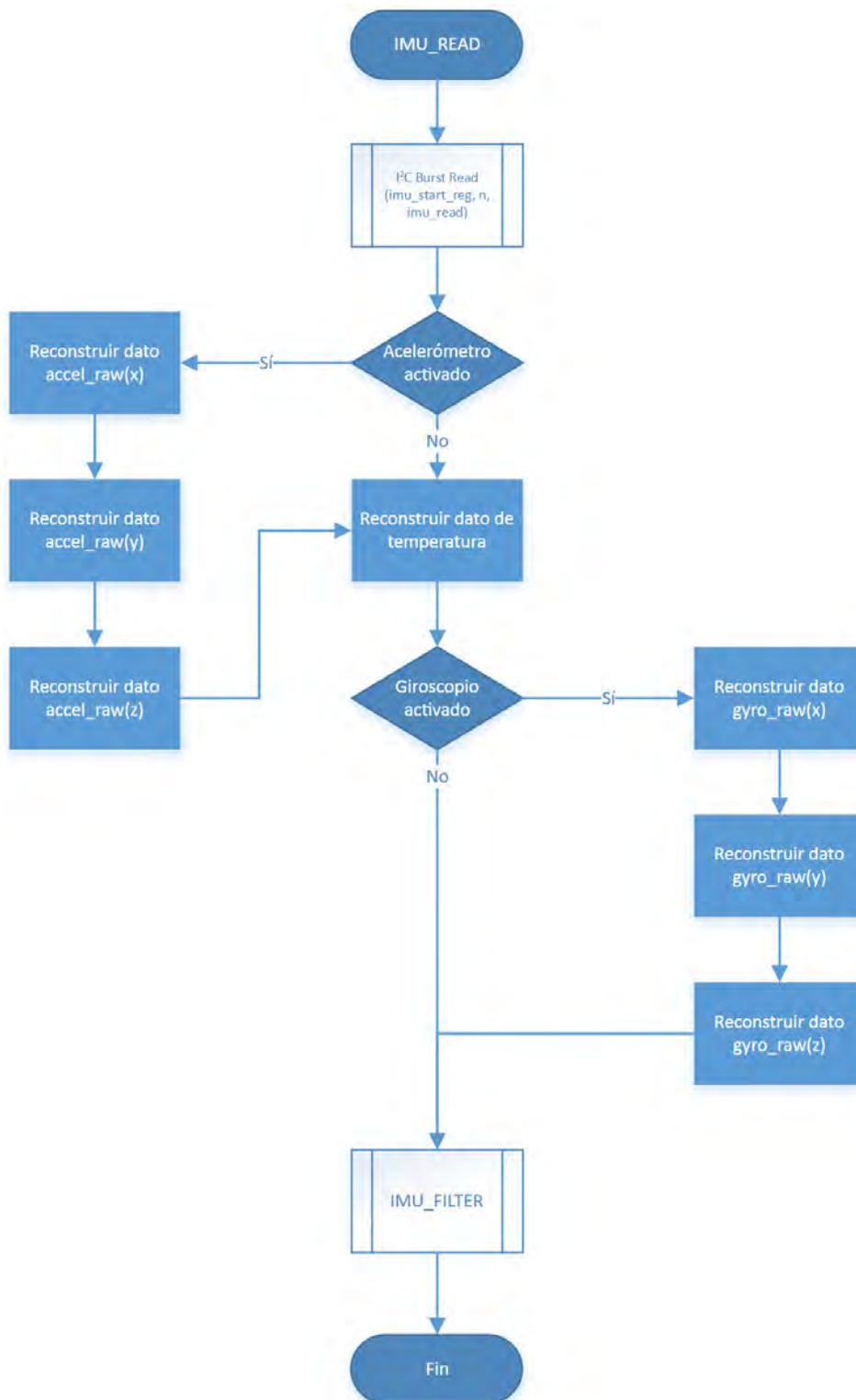


Figura 3.96: Diagrama de flujo de lectura de la IMU.

D) Configuración y uso del Ultrasónico

La lectura del sensor ultrasónico se realiza a una frecuencia de 10 [Hz], siguiendo el protocolo descrito en la sección del sensor (véase subsubsección D) Ultrasónico).

El uso del sensor se activa cuando el **Maestro** envía el caracter ‘U’, Figura 3.97. Al ejecutar la interrupción, el **Esclavo** configura el Timer 2, que es el timer que emplea durante todas las fases de la lectura del sensor; también enciende el sensor a través del MOSFET conectado e inicializa el proceso después de esperar que el sensor se estabilice durante 20 [ms].

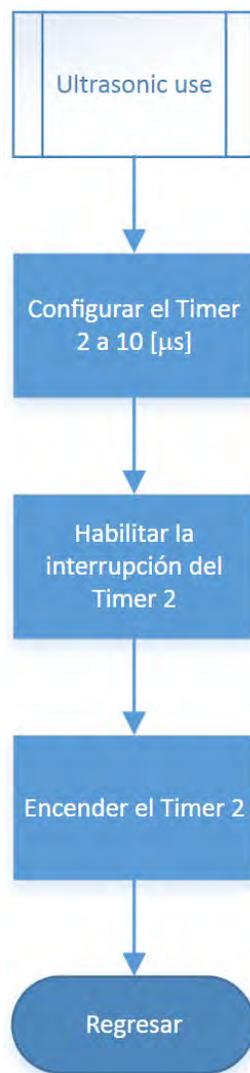


Figura 3.97: Diagrama de flujo de configuración de uso del sensor ultrasónico.

Después de que transcurre el tiempo de asentamiento de encendido, el **Esclavo** manda un pulso de 10 [μs] a través del pin **TRIG**.

Después de mandar el pulso, el **Esclavo** esperará a que el sensor responda y envíe la señal a través del pin **ECHO**. La señal es captada a través de una interrupción externa, lo cual permite realizar otros procesos mientras que el sensor manda la señal correspondiente a la distancia al objeto más cercano.

Posteriormente, el **Esclavo** toma la lectura del tiempo de la señal recolectando el dato en el timer,

en donde puede ocurrir uno de los siguientes eventos: que el pulso termine antes de desbordar el timer, mismo que se refleja a través de la interrupción externa; o bien, se desborda el timer, cuando el tiempo máximo ha sido alcanzado.

El pulso se debe limitar para poder conseguir una lectura uniforme del sensor, que en este caso está ajustada a una distancia máxima de 50 [cm]. Una vez terminada esta fase, se traduce el valor del timer a distancia mediante la fórmula de la Ecuación 3.10. Posteriormente, se apaga el MOSFET que alimenta al sensor y se mantiene en ese estado durante el resto de tiempo para acompletar 80 [ms], finalizando así el proceso de lectura. En la Figura 3.98 se aprecia el diagrama de flujo del timer y en la Figura 3.99 se aprecia el diagrama de flujo de la interrupción externa.

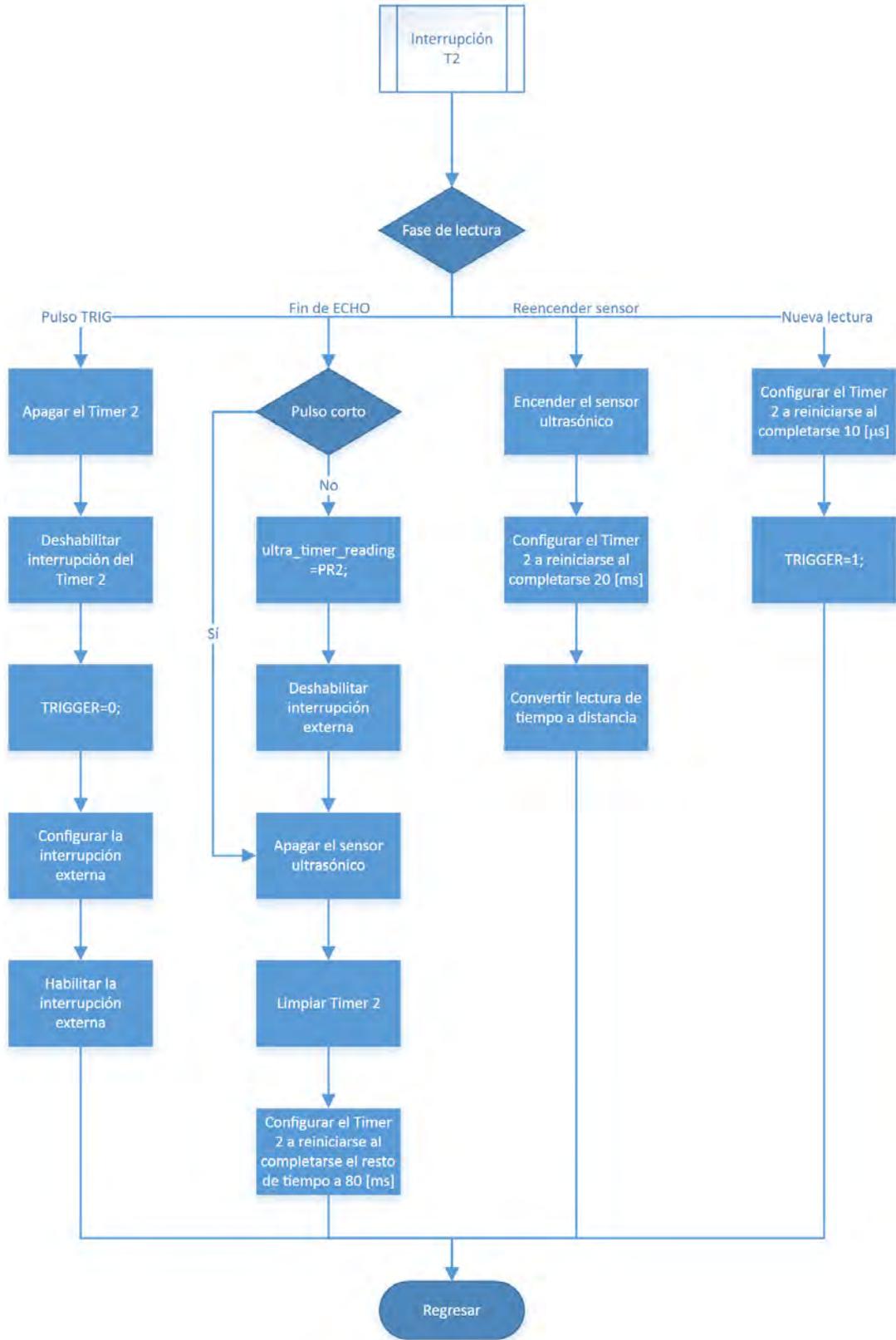


Figura 3.98: Diagrama de flujo del Timer 2.

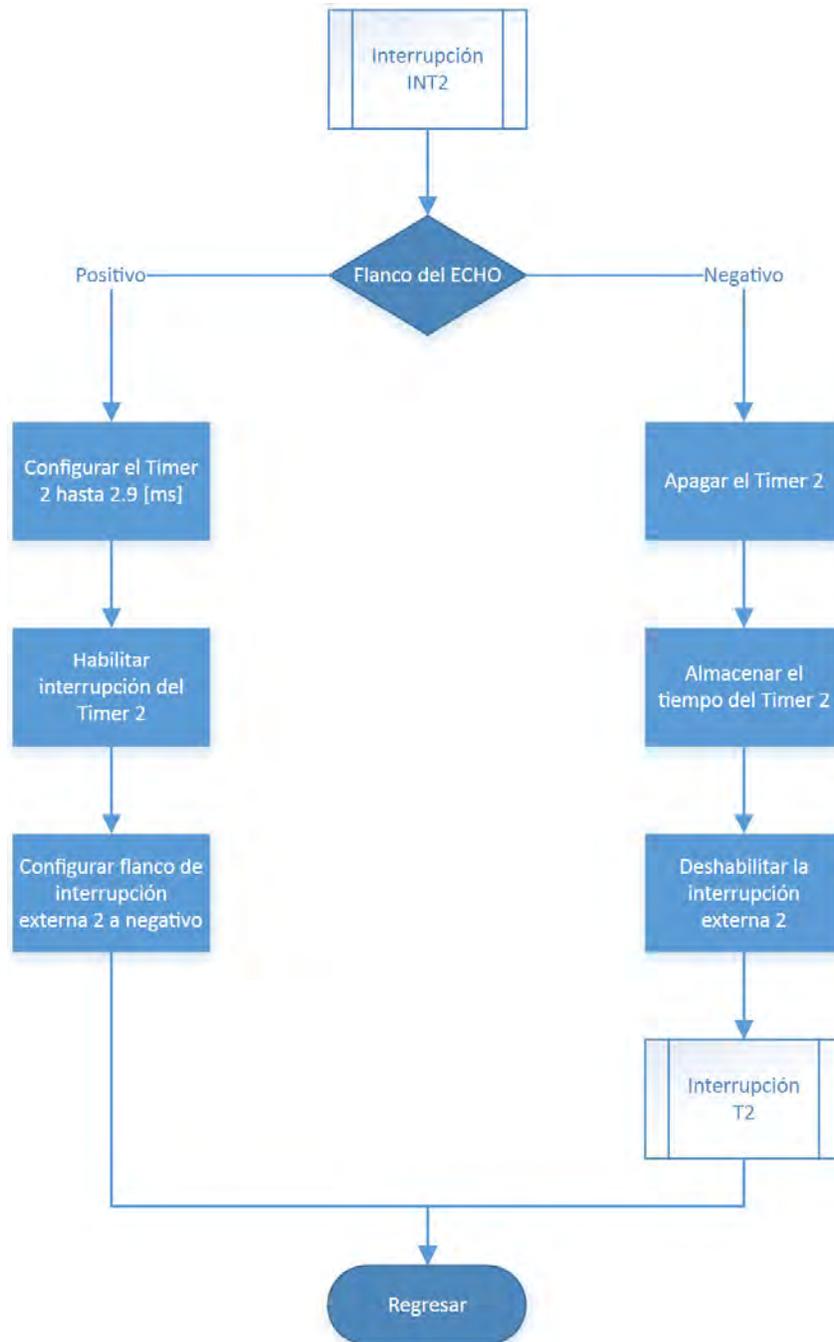


Figura 3.99: Diagrama de flujo del pin ECHO.

E) Configuración y uso del LED Driver

Tras el encendido y el caracter de arranque del **Maestro**, el **Esclavo** realiza las configuraciones iniciales de la IMU y del LED Driver, en lo consiguiente PCA. La configuración de arranque incluye escribir los registros **MODE1** (registro de control de desplazamiento interno), **LEDOUT0...5** (salidas a usar), **IREF0...17** (corriente inicial), Figura 3.100.

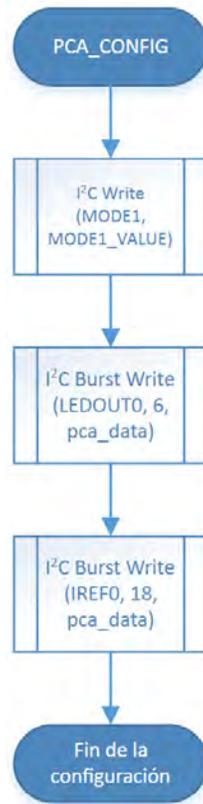


Figura 3.100: Diagrama de flujo de configuración inicial del PCA.

Cada cara del robot tiene un comando de configuración distinto. Para cambiar el color de encendido de los LEDs de cada cara se deben mandar 2 caracteres: el primer caracter es un número en texto ASCII que va del ‘1’ al ‘6’; el segundo caracter es un valor que va del 0x00 al 0x0B, este valor no tiene codificación alguna. El segundo caracter es el que indica el color al que deberá cambiar la cara del robot. En la Tabla ?? se aprecian las diferentes posibilidades de colores disponibles y el valor que se escribe en los registros de PWM del PCA.

Para optimizar el tiempo de configuración que requiere el **Esclavo** para enviar toda la configuración de cada cara al PCA, se reorganizó la tabla de colores dependiendo si el color combina los colores rojo y verde o no, dando como resultado la numeración de la Tabla 3.9.

Cuando el **Maestro** envía un caracter acorde a cualquiera de las caras, una interrupción se lanza. Se espera la recepción del segundo caracter y elige la configuración que posteriormente se envía a través del bus I²C al PCA.

Al configurar el PCA primero se configura la corriente de cada terminal de acuerdo al color que se seleccionó. Para ello se direcciona cualquiera de los registros **IREF0**, **IREF3**, **IREF6**, **IREF9**, **IREF12** o **IREF15** y se escribe en los 3 registros contiguos el valor de corriente deseado, que varía en una escala de 0 a 255.

Tabla 3.9: Organización de colores disponibles de los LEDs.

Valor (hex)	Color	RGB		
		Rojo	Verde	Azul
0x00	Blanco	255	255	255
0x01	Morado	56	12	75
0x02	Naranja	255	55	0
0x03	Amarillo	255	160	0
0x04	Rosa	255	90	120
0x05	Aquamarine	50	255	212
0x06	Rojo	255	0	0
0x07	Verde	0	255	0
0x08	Azul	0	0	255
0x09	Magenta	255	0	255
0x0A	Cyan	0	255	255

Posteriormente se vuelve a iniciar la comunicación con el PCA para configurar el valor del PWM para conseguir el color. Se direccionan los registros **PWM0**, **PWM3**, **PWM6**, **PWM9**, **PWM12** o **PWM15** y se escribe en los 3 registros contiguos el valor de PWM deseado, que varía en una escala de 0 a 255. En la Figura 3.101 se aprecia el diagrama de flujo de la configuración de colores de los LEDs.

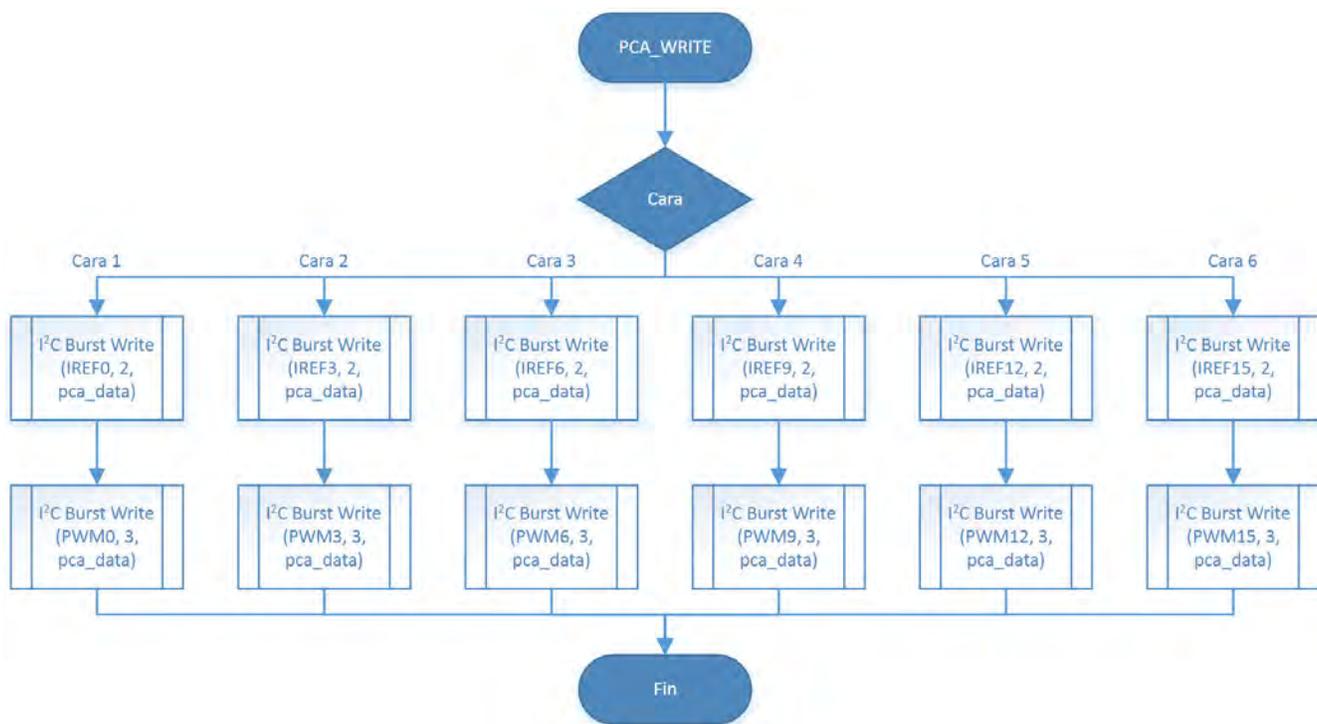


Figura 3.101: Diagrama de flujo de configuración de color por cara.

F) Configuración y uso de los motores

Los motores no están conectados de manera directa al **Esclavo** sino a un puente H que brinda la potencia a los motores. El **Esclavo** funciona como la unidad de control, donde para cada motor se emplean 3 pines: dos de propósito general I/O, que se conectan a los pines de dirección del puente H, y uno de PWM, quien se conecta al enable del puente H.

Para activar el uso de los motores, el **Maestro** debe enviar un caracter ‘M’ al **Esclavo** para habilitar el uso de los motores. Al activar la interrupción por el UART 2, el **Esclavo** configura el módulo PWM y configura los pines para su funcionamiento. El PWM se configura a 20 [kHz] pero no se asigna un valor inicial de velocidad a los motores. La dirección por defecto es nula, lo que para los motores representa libre rodada. Una bandera le indicará al **Esclavo** que los motores ya están habilitados. En la Figura 3.102 se aprecia el diagrama de flujo de configuración de los motores.



Figura 3.102: Diagrama de flujo de configuración del PWM.

Una vez habilitado el uso de los motores, el **Maestro** debe enviar la velocidad que se asignará a cada motor. Para ello, el **Maestro** debe enviar primero un caracter ‘L’ (motor izquierdo) o ‘R’ (motor derecho) para indicarle al **Esclavo** a qué motor debe configurar la velocidad. Durante la interrupción, el **Esclavo** configurará la interrupción UART para activarse en cuanto haya recibido 4 caracteres y habilita una bandera del motor a configurar la velocidad.

El formato de velocidad de los motores corresponde a un número tipo Float32, es decir, un número punto flotante de precisión simple, especificado en la norma IEEE 754. El dato comprende 32 bits, 1 bit para el signo, 8 bits para el exponente y 23 bits para la fracción.

Para evitar la pérdida de tiempo del dato haciendo una conversión a texto por parte del **Maestro**, enviando el dato con la mantisa particular de cada número y finalmente haciendo la reconversión de vuelta a punto flotante, el dato se envía de manera directa a través del UART. El **Esclavo** únicamente hace la reconstrucción del número mediante el uso de un puntero de 8 bits que, a medida que extrae byte por byte el número de la pila del UART, lo coloca en el registro correspondiente, logrando así la reconstrucción del número punto flotante. En la Figura 3.103 se aprecia el diagrama de flujo de la reconstrucción del número.

Una vez que se reconstruyó el dato se reconfigura la interrupción a efectuarse al recibirse un solo carácter y se prosigue con la determinación del sentido de giro del motor. Para determinar el sentido de giro se verifica el valor del bit de signo del dato float, que es el último bit del dato, o bien, el bit 31. Para verificar el estado de ese bit se asigna a un puntero el valor del último byte del dato y se hace una comparación tipo AND con el valor $0x80$. Cuando el bit vale '0', significa que el dato es positivo y el motor debe girar en un sentido; si el bit vale '1', el valor es negativo y el motor girará en sentido contrario; la convención se aprecia en la Tabla 3.10. Esta comparación se hace para ambos motores en la misma función y se actualiza el valor de los pines de dirección de cada motor. En la Figura 3.104 se aprecia el diagrama de flujo de la comparación.

Tabla 3.10: Valor de los pines y sentidos de giro de los motores.

Pin	Valor	Sentido de giro del motor
DIR_1_A	←→ '1'	Horario
DIR_1_B	←→ '0'	
DIR_1_A	←→ '0'	Antihorario
DIR_1_B	←→ '1'	
DIR_2_A	←→ '0'	Horario
DIR_2_B	←→ '1'	
DIR_2_A	←→ '1'	Antihorario
DIR_2_B	←→ '0'	



Figura 3.103: Diagrama de flujo de reconstrucción del dato float.

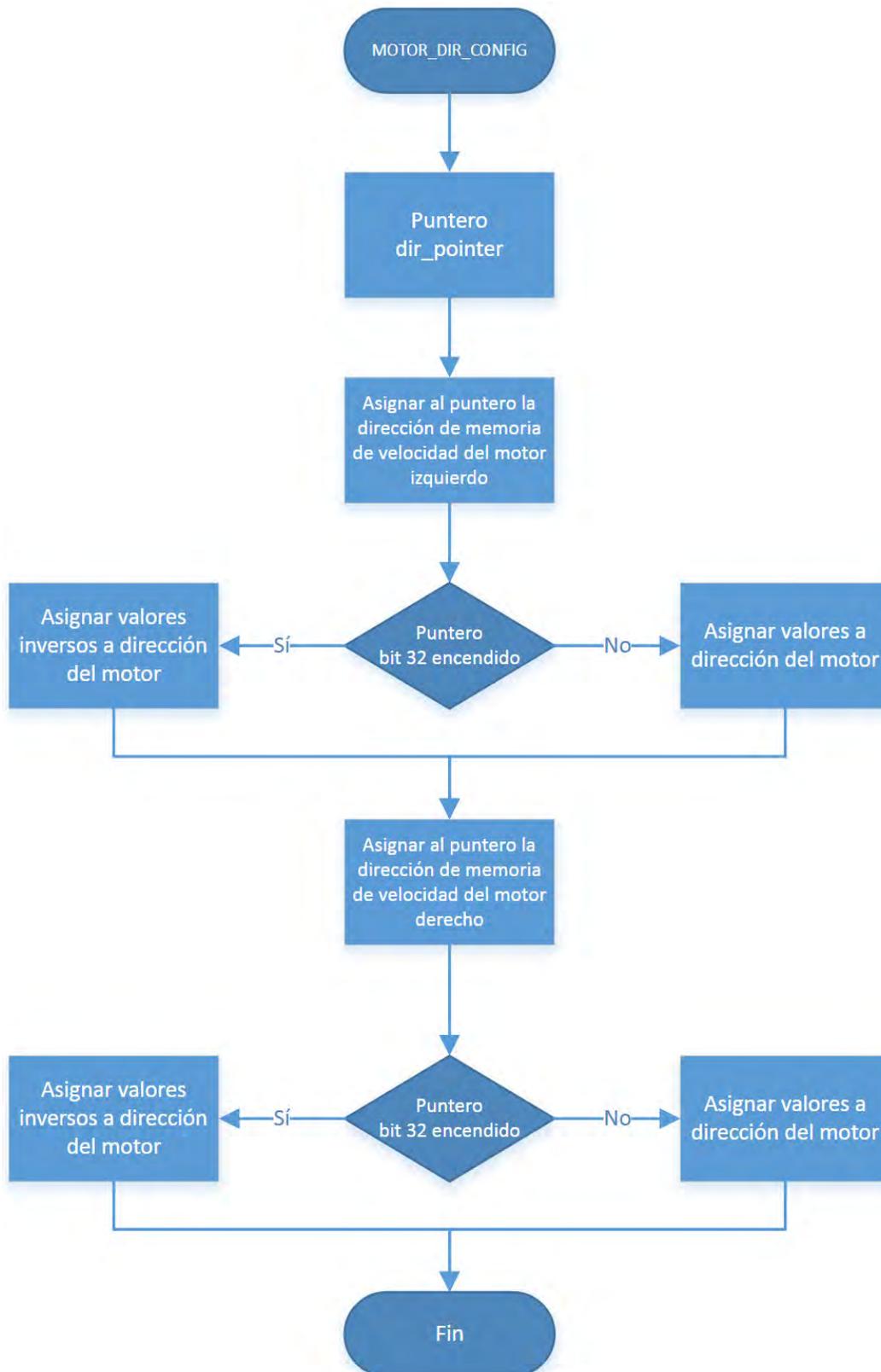


Figura 3.104: Diagrama de flujo de configuración de sentido de giro de los motores.

G) Programa general del Esclavo

El dsPIC no tiene un procesador multinúcleo que pueda manejar más de un hilo a la vez, como el caso de diferentes procesadores x86/x64 o ARM instalados en computadoras y diversos dispositivos móviles. Por lo que se evita en lo mayor posible los tiempos muertos de esperas (delays).

Diferentes procesos de lectura y ejecución pueden abarcar grandes tiempos, como la lectura de múltiples registros en la IMU que puede alcanzar hasta 14 registros. Para ese caso, por ejemplo, el leer los 14 registros, dejando de lado el direccionar el primer registro, requiere un tiempo de

$$14 \text{ registros} + \text{START} + \text{STOP}$$

$$1 \text{ registro} = 8 \text{ bits} + 1 \text{ ACK} = 9 \text{ ciclos}$$

$$14 \text{ registros} + \text{START} + \text{STOP} = 128 \text{ ciclos}$$

$$\text{a la frecuencia de } 400 \text{ [kHz]}$$

$$t_{lectura} = \frac{128}{400E03} = 320 \text{ [us]}$$

tomando en cuenta que un ciclo máquina del dsPIC equivale a

$$t_{CM} = \frac{4}{120 \text{ [MHz]}} = 33.\bar{3} \text{ [ns]}$$

significa entonces que el tiempo de leer los 14 registros equivale a

$$t_{equiv} = \frac{320 \text{ [us]}}{33.\bar{3} \text{ [ns]}} = 9600 \text{ CM}$$

y, aunque cada instrucción consume un cierto número de ciclos máquina y el alcance de esta tesis no es determinar el equivalente de instrucciones para cada caso particular de procesamiento de toda función que realizará el dsPIC, sí se puede deducir que ese tiempo “muerto” se puede aprovechar en el procesamiento de otras funciones.

Todo tiempo “muerto” que tenga el **Esclavo** se puede reaprovechar para calcular el resto de las funciones que requieran atención en su momento; esos tiempos muertos están generalmente asociados al envío y recepción de información, también existen en la lectura del sensor ultrasónico.

Con la finalidad de aprovechar todo tiempo “muerto” disponible, se aprovechan todas las interrupciones disponibles en los protocolos de comunicación, timers y cambios de estado, mismas que se mencionan en el apartado A). Un ejemplo de ello ya ha sido descrito en la subsubsección D) **Ultrasónico**, donde en vez de esperar a que se cumplan los tiempos del pulso de arranque o que se dé el evento de cambio de estado, se recurren a dos interrupciones distintas para conocer el momento en el que se dieron los diferentes eventos.

Las interrupciones se desencadenan tras los siguientes eventos:

- UART: Dependiendo dos bits de configuración, cuando el buffer se ha llenado con 1 caracter o está lleno con 4 caracteres. Diagrama: Figura 3.105.

- I²C: Al finalizar un evento START, STOP, RESTART, ACK o recepción de un byte. Diagrama: Figura 3.106.
- Interrupción externa: Dependiendo del bit de configuración de flanco, la interrupción ocurre cuando hay un flanco ascendente o descendente en el pin de entrada.
- Timer: Cuando se desborda la cuenta del timer.

Las interrupciones se combinan con banderas. Cuando algún proceso requiere un tiempo de espera, el **Esclavo** se saldrá del proceso y continuará con el proceso siguiente que requiera atención. De esta manera, cuando llega a algún proceso cuya bandera indica que requiere atención, entra al proceso y realiza las instrucciones pertinentes, que pueden desembocar en otro tiempo de espera, activar la bandera de otro proceso o finalizar el proceso.

Por consiguiente, la mayoría de las interrupciones se enfocan en la activación o desactivación de banderas que, al salir de la interrupción, durante la ejecución del *main* el **Esclavo** accederá al proceso y ejecutará las instrucciones pertinentes.

Dentro del *main* se encuentran entonces todos los procesos anidados bajo diferentes banderas. Una bandera particular controla el acceso al cada uno de ellos.

En conjunto, más de 57 banderas controlan el acceso a cada una de las tareas y sus respectivos subprocesos dentro del *main*. Al no existir la capacidad multi hilo dentro del dsPIC, los procesos se anidan en secuencia; cuando el puntero de dirección llega al siguiente, polea la bandera para determinar si debe de entrar; en caso de que la bandera esté apagada, el programa omitirá entrar a la función. Cuando llega al final del programa, regresa a la dirección del primer proceso y continuará poleando las banderas de cada una de las tareas en su momento respectivo, dando atención así únicamente a las funciones que requieran atención, aprovechando el tiempo que, de otra manera, habría perdido esperando a que se cumpliera alguna transmisión o esperar un evento como cambio de estado.

En la Figura 3.107 se aprecia el diagrama de flujo de la función principal del **Esclavo**. En ella se ejemplifican todos los procesos que desarrolla el microcontrolador cuando requieren atención, mismos que fueron descritos en las secciones anteriores y cuyos diagramas de flujo muestran gráficamente de manera particular para cada proceso y subproceso su funcionamiento, con la modificación de que los tiempos de espera se sustituyen por poleos de banderas únicamente.

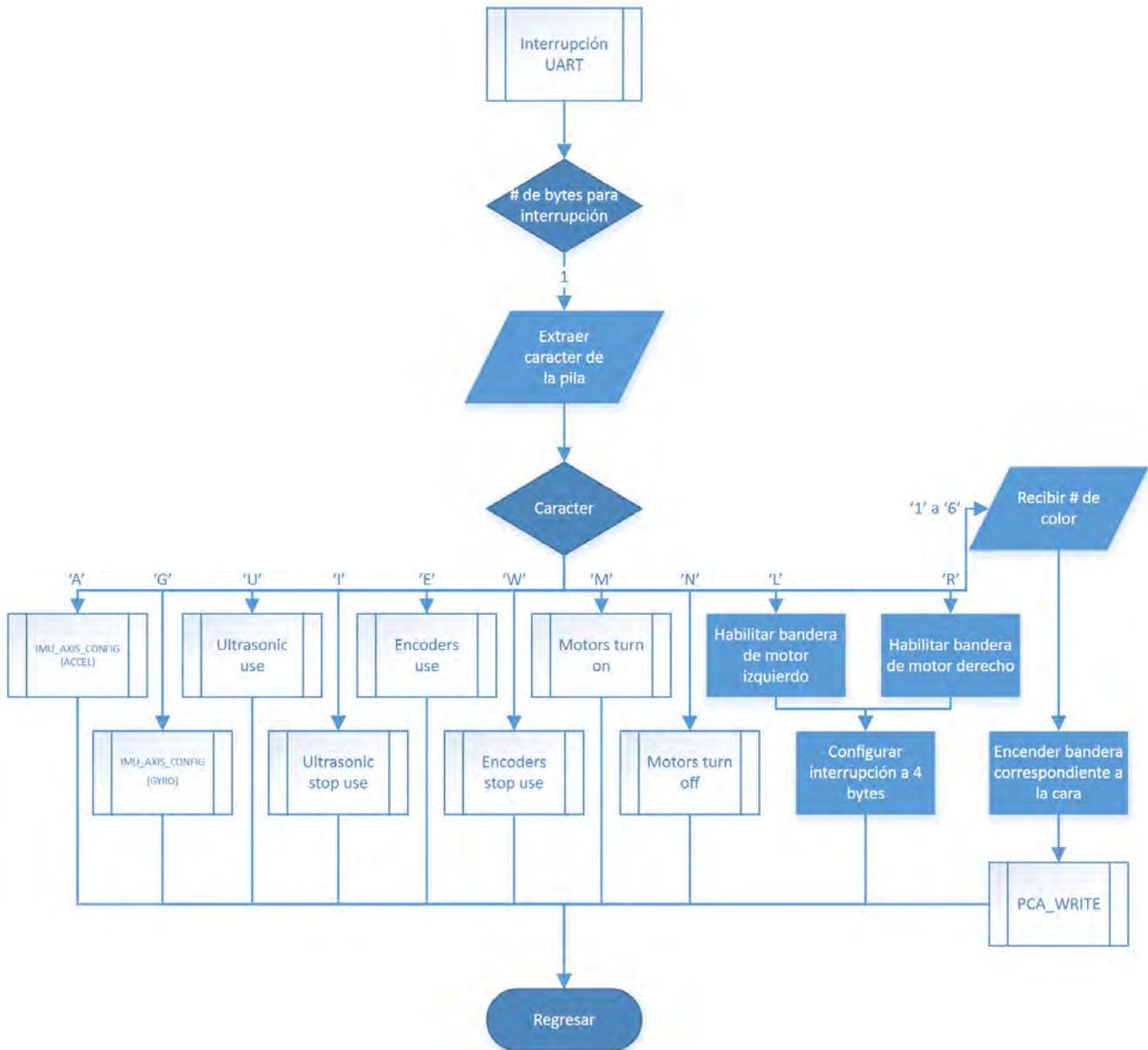


Figura 3.105: Diagrama de flujo de la interrupción UART2.

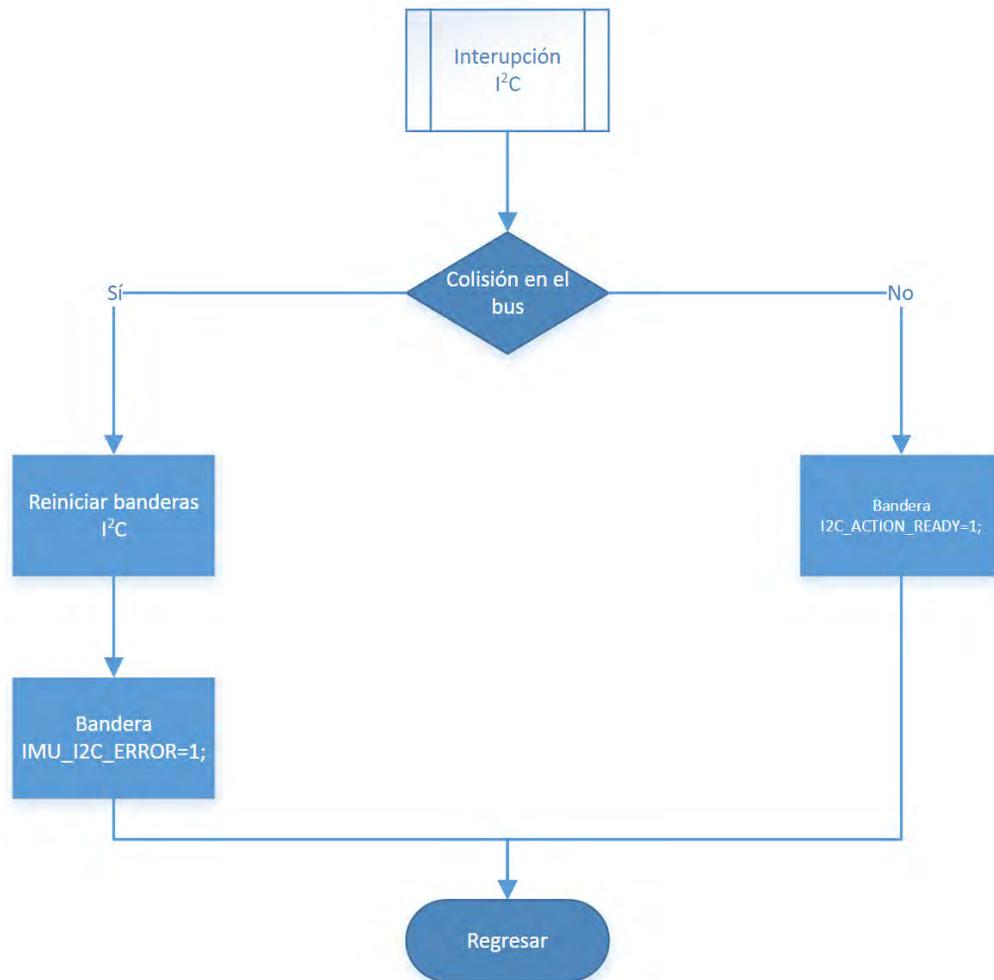


Figura 3.106: Diagrama de flujo de la interrupción I²C.

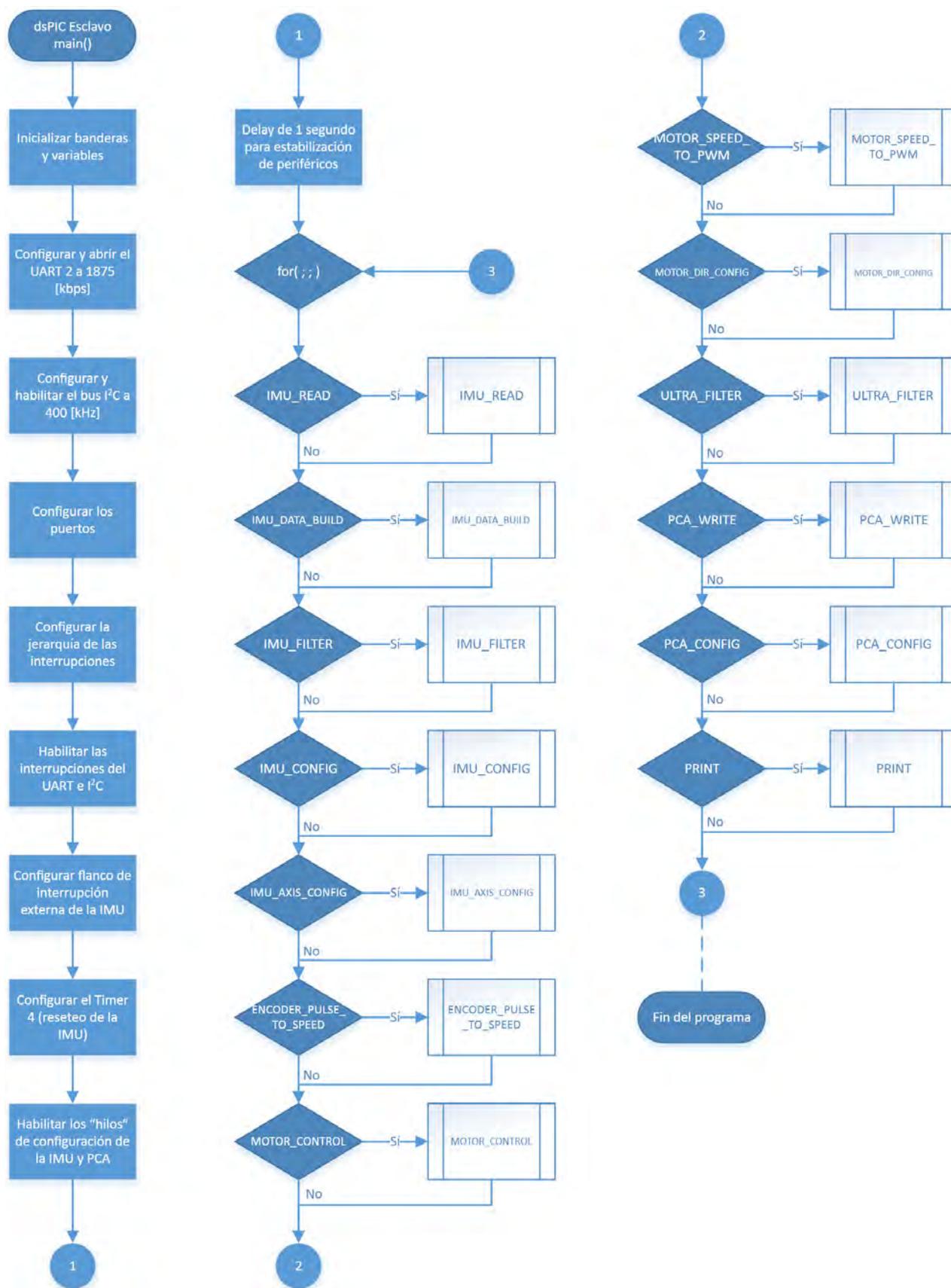


Figura 3.107: Diagrama de flujo del main del **Esclavo**.

Capítulo 4

Pruebas y análisis

En este capítulo se abarcan las pruebas preliminares realizadas a los componentes para conocer su comportamiento y tomarlo en consideración para el diseño; posteriormente se abarcan las pruebas a los diferentes módulos del robot, donde se analiza el funcionamiento de cada uno de los módulos y/o componentes. En esta última sección se detallan los errores de diseño y dificultades en general encontradas a lo largo de las pruebas y se dan algunas posibles soluciones a esos problemas.

4.1 Pruebas a los sensores

Se realizaron pruebas a todos los sensores para conocer su comportamiento, de los cuales se obtuvieron diversas gráficas que sirvieron para comprender el comportamiento de los encoders, IMU y sensor ultrasónico y se observó el comportamiento ante diferentes voltajes en el caso de los comparadores.

Las pruebas realizadas, en algunos casos con circuitos similares a los diseños finales presentados en el Capítulo 3, sirvieron para pulir el diseño de la electrónica y los protocolos de sensado para los diferentes sensores, así como las consideraciones finales de tiempos de procesamiento requeridos para cada sensor.

4.1.1 Pruebas a las señales de salida de los Encoders

Para poder realizar cualquier prueba con los encoders, primero se tuvieron que soldar a las terminales del motor y se les tuvieron que soldar cables. Una vez realizado, se encontró que la señal de los encoders se puede interferir fácilmente con la luz ambiental, por lo que las estrellas tienen que estar lo mejor ocultas posible, aunque es posible obtener una salida capaz de ser distinguida por los microcontroladores con tan sólo bloquear la luz ambiental con la mano, como se aprecia en la Figura 4.1. La señal, al ser afectada, da como resultado que la salida pico se reduzca en algunos casos a menos de 1 [V]. Por el contrario, cuando los encoders son puestos a cubierto, llegan a dar pulsos de entre 3 [V] y 4 [V] pico. También se puede apreciar que la señal no es cuadrada; tiene un periodo de subida y un periodo de bajada, dependiendo también de la cantidad de luz ambiental. El comportamiento de los encoders se muestra en las Figuras 4.1, 4.2 y 4.3.

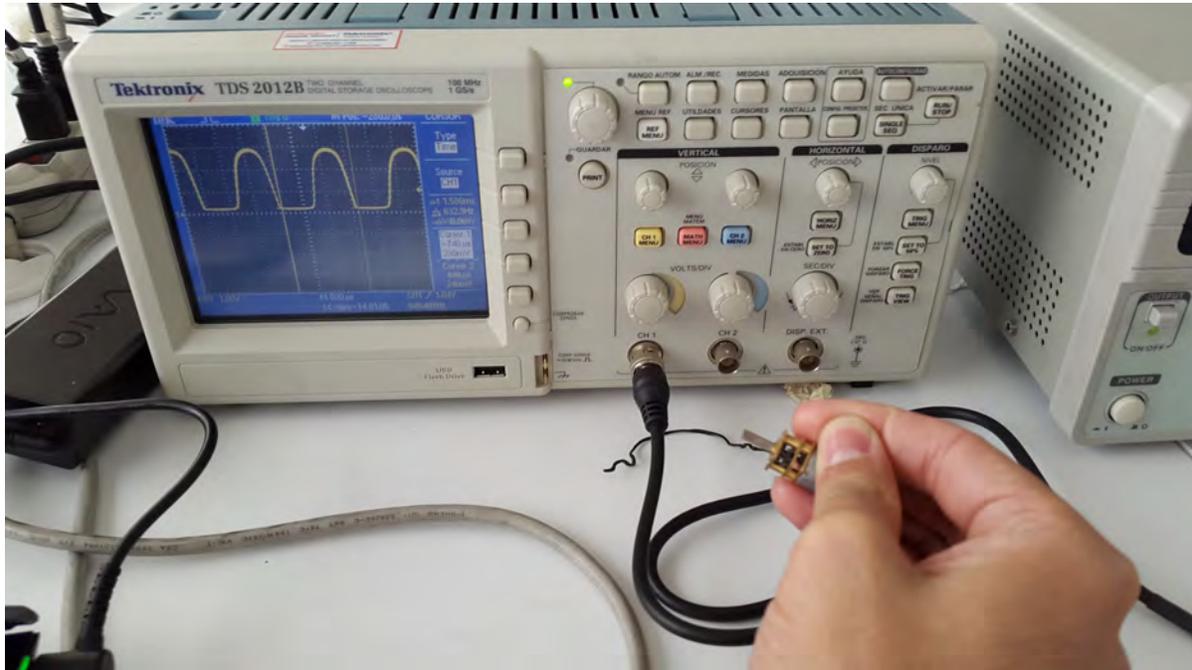


Figura 4.1: Se encontró que al cubrir el encóder es que la señal se puede leer de manera efectiva, de lo contrario, la señal es muy tenue o simplemente se aprecia como nula.

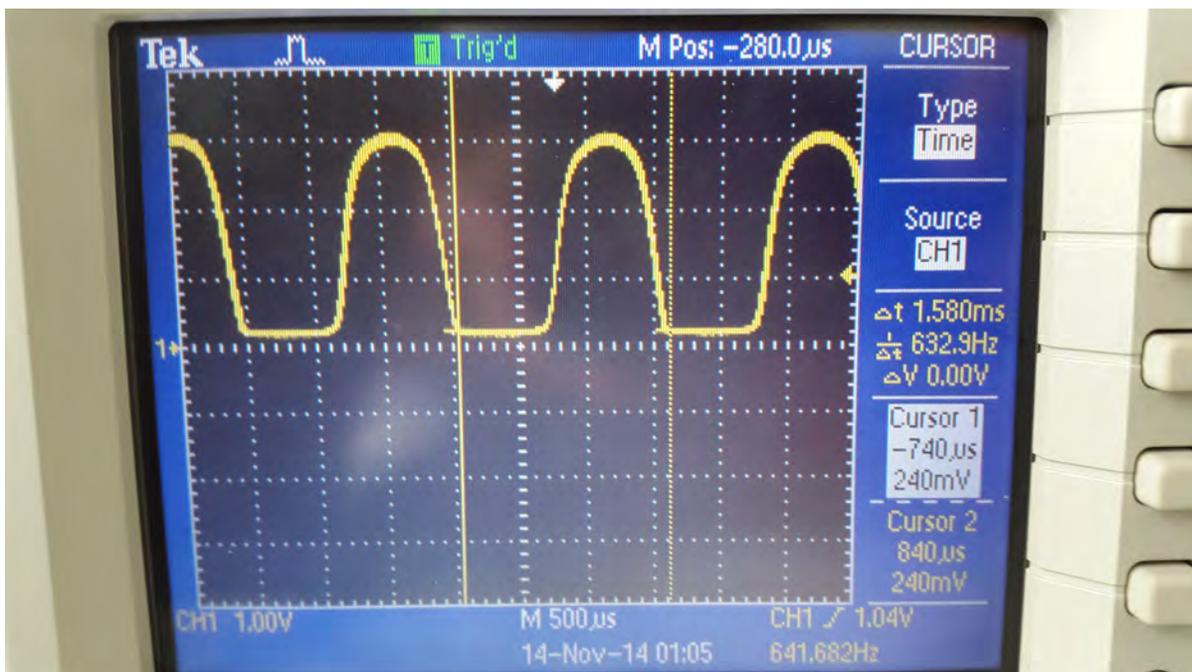


Figura 4.2: Se aprecia una señal del motor alimentado a 6 [V]. La señal no es cuadrada pero se puede diferenciar bien el estado alto y el bajo.

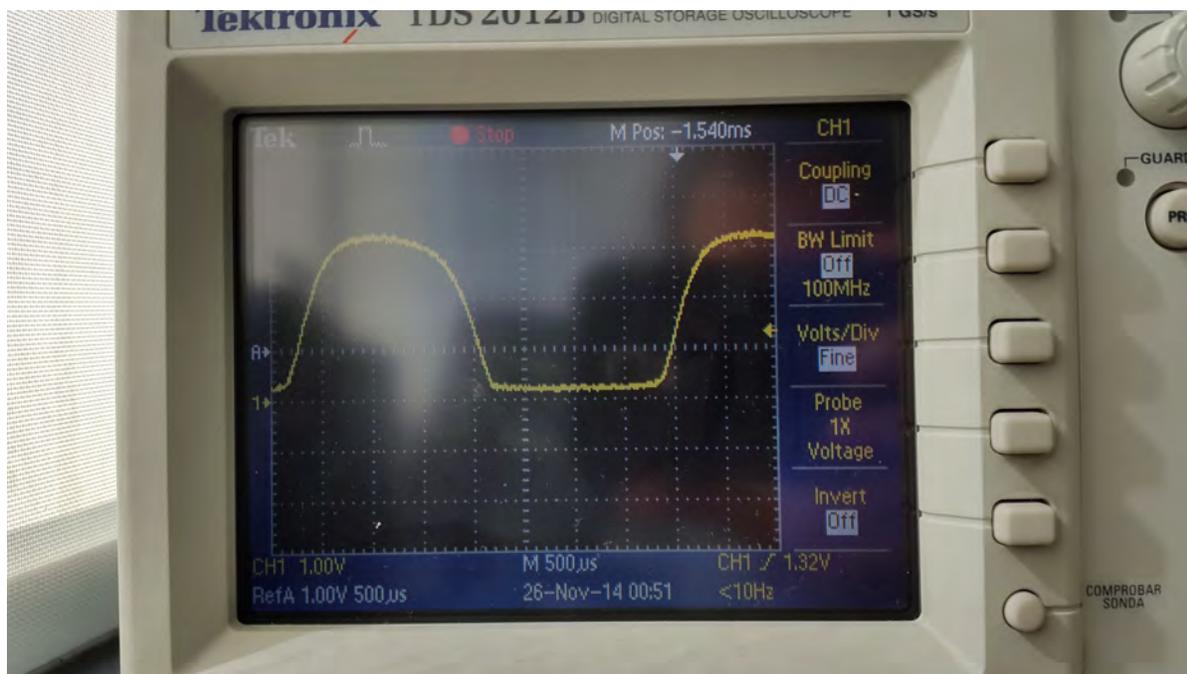


Figura 4.3: Zoom a la señal del encóder, se puede apreciar de mejor manera un comportamiento tipo “meseta” de la señal proveniente del encóder.

De estas pruebas se determinó que en el diseño de los *brackets* se debía agregar una tapa que impida la interferencia entre ambos encoders una vez montados en el robot (véase sección 3.1.3 *Brackets*). Las gráficas también indican que los encoders de encuadratura ópticos, al ser tan fácilmente alterables, pueden no ser la mejor solución para muchos casos, pudiendo ser reemplazados por encoders de encuadratura magnéticos o de láser en un futuro.

El circuito decodificador empleando el FF-D se probó después de tener el diseño preliminar, empleando circuitos equivalentes en empaquetados que se pudieran montar en una *proto-board*, en las Figuras 4.4-a y 4.4-b se muestra la salida que arroja el decodificador de tipo FF-D cuando el motor gira en uno u otro sentido. Para tal prueba se conectó un LED ultrabrillante azul al CLK del FF-D para poder observar los pulsos y un LED ultrabrillante verde para observar el sentido, conectado a la terminal Q del FF-D. En la Figura 4.4-a se aprecia que el LED se enciende ('1' lógico). En la Figura 4.4-b el LED se encuentra apagado ('0' lógico). El resultado de la señal adecuada para ser decodificada se aprecia en la Figura 4.5.

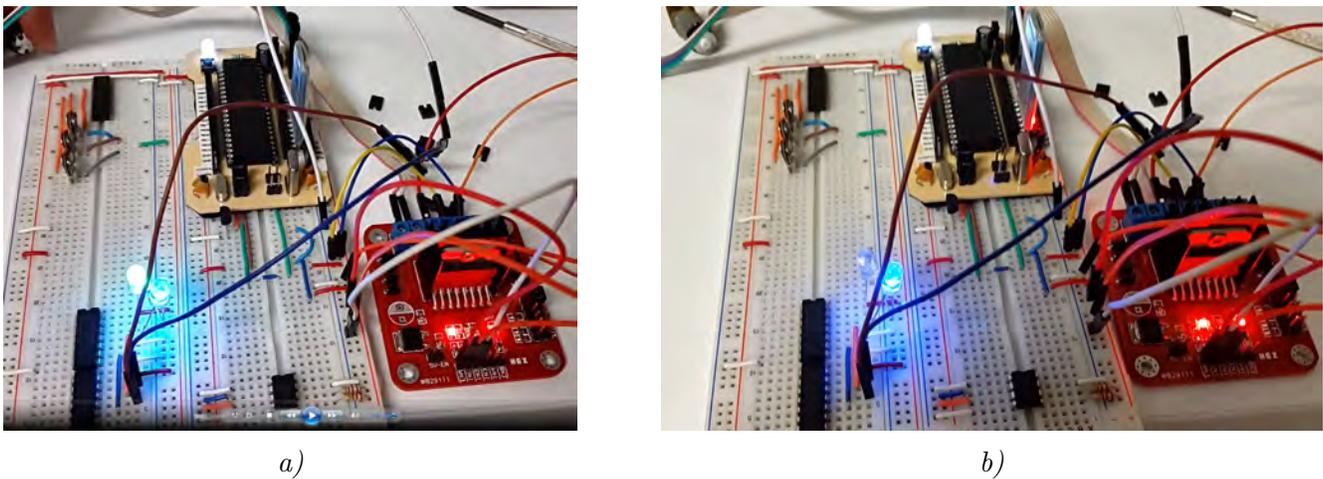


Figura 4.4: Prueba del encóder empleando el circuito con FF-D; en a) se aprecian encendidos tanto la señal de la terminal *Q* y en b) se aprecia encendido únicamente la señal *CLK*.

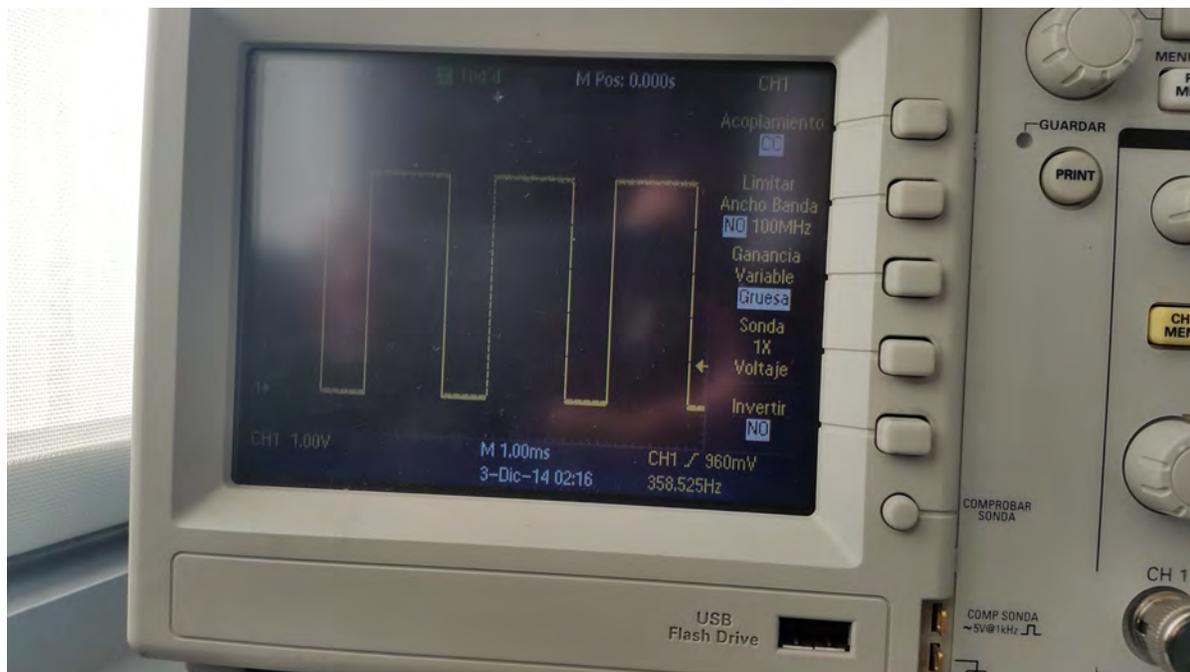


Figura 4.5: Señal adecuada para ser decodificada por el FF-D.

4.1.2 Pruebas a los ejes de la IMU

Para determinar el mejor filtro para las lecturas tanto del acelerómetro como del giroscopio es necesario probar una muestra de los filtros; uno cuya frecuencia de corte sea cercana a 0 (conocer los datos “crudos”), uno con frecuencia de corte intermedia (filtro intermedio con retardo intermedio de la señal) y uno cercano a la frecuencia de corte más grande posible (el mejor filtro pero con el mayor retardo de la señal). Las frecuencias de corte seleccionadas para probar los filtros son: 5 [Hz], 21 [Hz] y 184 [Hz], para evitar el valor reservado y la frecuencia de corte que incrementa la frecuencia de muestreo del giroscopio (Tabla 3.2).

Las pruebas se realizan encendiendo la IMU, configurando la resolución a la más baja con

una frecuencia de muestreo de 20 [Hz], misma que usa el robot para leer la IMU. Se leen todos los registros asociados a las mediciones del acelerómetro y el giroscopio manteniendo la IMU de manera estática durante toda la prueba. Esos datos se van enviando a través de un puerto serial a la computadora, que los recolecta y almacena en un archivo de texto para su posterior análisis. Los datos recolectados son procesados y graficados en Matlab, de donde se puede obtener la desviación estándar de los mismos, para observar el desempeño de los filtros paso bajas. La desviación estándar de los datos obtenidos se calcula a partir de que los ejes de la IMU ya se estabilizaron, tiempo que le lleva de entre 3 a 5 segundos una vez configurados.

En la Tabla 4.1 se muestran los resultados de las pruebas obtenidas para todos los ejes de la IMU a las frecuencias de corte de 5 [Hz], 21 [Hz] y 184 [Hz].

Tabla 4.1: Desviación estándar para los ejes de la IMU.

Eje	Frec. corte		
	5 [Hz]	21 [Hz]	184 [Hz]
Acelerómetro <i>X</i>	0.000540	0.001053	0.002637
Acelerómetro <i>Y</i>	0.000529	0.001067	0.002715
Acelerómetro <i>Z</i>	0.000891	0.001707	0.004177
Giroscopio <i>X</i>	0.015299	0.022843	0.050091
Giroscopio <i>Y</i>	0.018667	0.030662	0.061373
Giroscopio <i>Z</i>	0.013061	0.021103	0.044419

En todas las pruebas, el eje con más “ruido” siempre es el eje *Y* del giroscopio, al tener en todas las pruebas la mayor desviación estándar. Con respecto a las diferentes frecuencias de corte; la frecuencia de 21 [Hz] arroja datos de aproximadamente el doble de ruido que la frecuencia de 5 [Hz]; a su vez, la frecuencia de 184 [Hz] arroja datos de más del doble de ruido que el filtro de 21 [Hz]. Sin embargo, el retardo de la señal incrementa en una proporción casi 4 veces mayor entre las frecuencias de corte, siendo de 2 [ms] para la de 184 [Hz], 8.5 [ms] para la de 21 [Hz] y 19 [ms] para la de 5 [Hz]. Las gráficas obtenidas del eje *Y* del giroscopio se aprecian en la Figura 4.6 a 5 [Hz], Figura 4.7 a 21 [Hz] y Figura 4.8 a 184 [Hz].

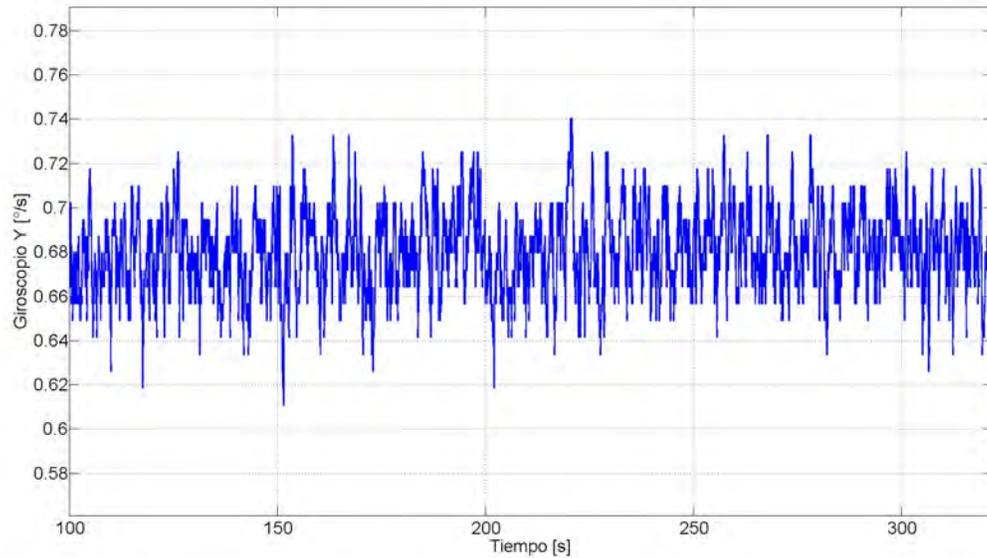


Figura 4.6: Eje Y del giroscopio con filtro paso bajas a 5 [Hz]. Es la menor cantidad de ruido proveniente del eje más ruidoso de la IMU pero retrasa la señal la mayor cantidad de tiempo.

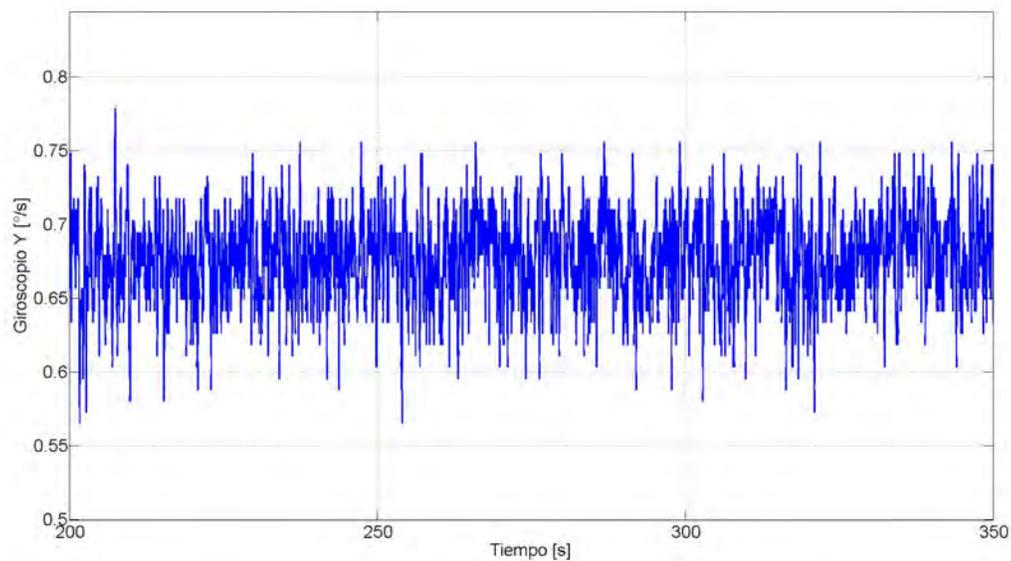


Figura 4.7: Eje Y del giroscopio con filtro paso bajas a 21 [Hz]. Es la cantidad de ruido intermedia con respecto al atraso de la señal.

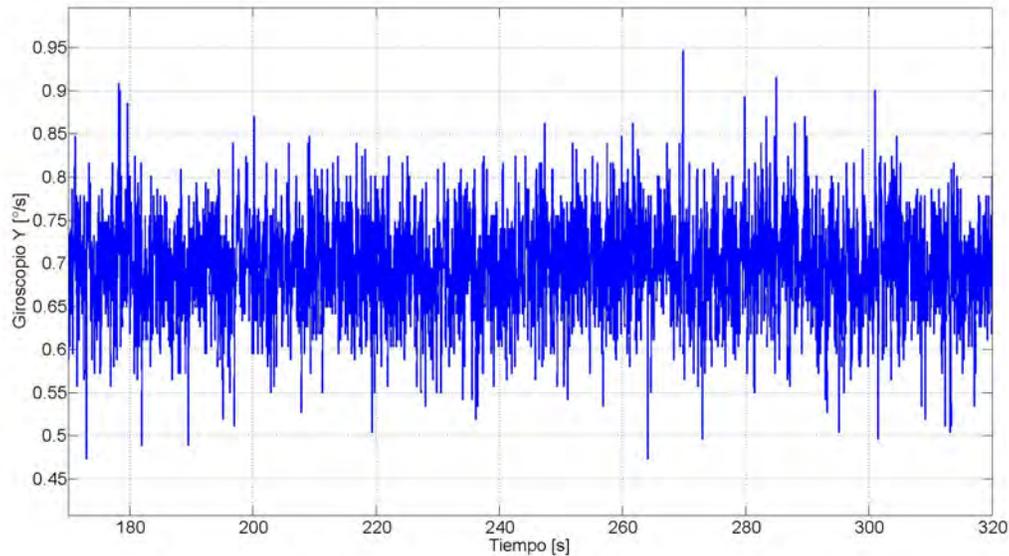


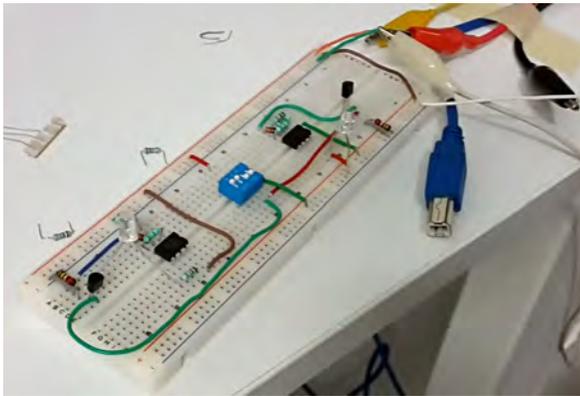
Figura 4.8: *Eje Y del giroscopio con filtro paso bajas a 184 [Hz]. El ruido es notablemente superior en comparación a los demás filtros pero es el menor atraso de la señal.*

4.1.3 Pruebas con los comparadores

Los comparadores *MCP6541* se encuentran tanto en empaquetados DIP (*Dual in-line*, por sus siglas en inglés) para hacer pruebas preliminares, como en empaquetado SOIC (*Small Outline Integrated Circuit*, por sus siglas en inglés) de tecnología SMT para el diseño final.

Dado que existe la dualidad de empaquetados para los comparadores, las pruebas se realizaron con componentes equivalentes que se pueden montar en una protoboard. Los diodos Zener empleados tienen un voltaje Zener de 3.3 [V], por lo que tanto la alimentación como el divisor son ajustados sólo para esta prueba. El circuito de prueba se aprecia en la Figura 4.9-a, donde las baterías se simulan con una fuente de voltaje, Figura 4.9-b.

Al disminuir el voltaje de alimentación la salida del comparador se simula con el encendido de un LED. En la Figura 4.9-a el circuito se encuentra alimentado, siendo la salida un '0' lógico dado que no se ha llegado al límite inferior. Caso contrario cuando el voltaje de alimentación cae por debajo del límite, primero el de la primera batería (Figura 4.10-a), representado mediante el LED verde, y finalmente el de la segunda batería (Figura 4.10b), representado por el LED azul.

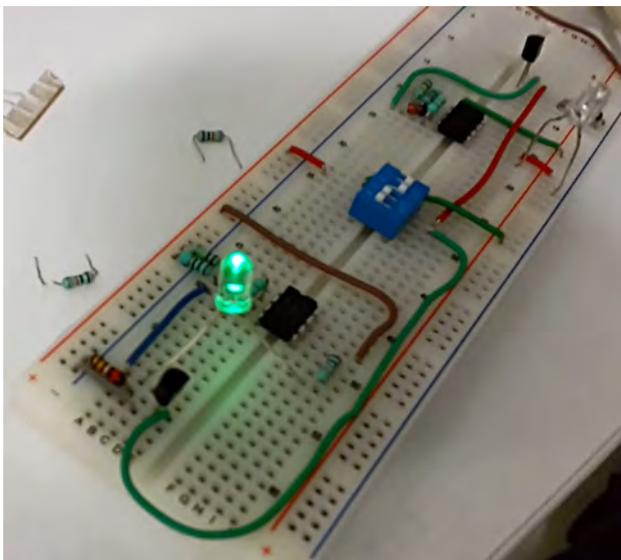


a)

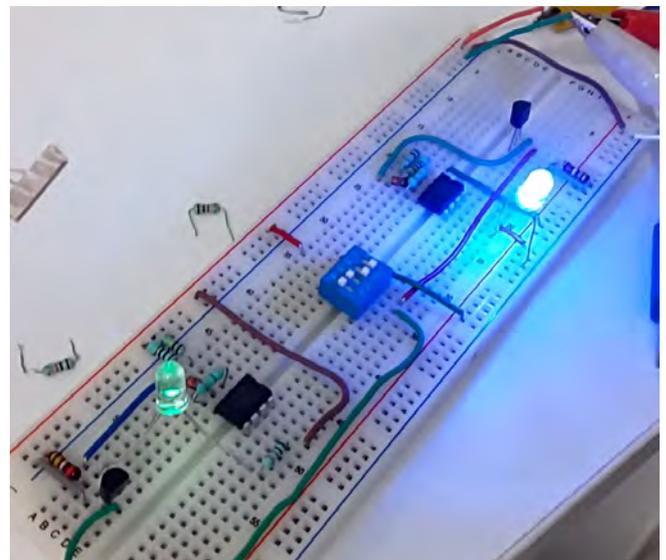


b)

Figura 4.9: a) Circuito de prueba equivalente de los comparadores; b) simulación de las “baterías”.



a)



b)

Figura 4.10: a) Simulación de caída del voltaje de la primera batería; b) simulación de caída del voltaje de la segunda batería.

4.1.4 Lectura del sensor ultrasónico

Con el fin de probar diferentes frecuencias de muestreo para determinar la más conveniente y el nivel de ruido que la señal pueda tener, se limita el rango del sensor a 100 [cm] con un circuito que emplea un MOSFET, similar al del diseño final (véase sección [D\) Ultrasónico](#)) y se ejecutan pruebas con diferentes objetos a diferentes frecuencias de muestreo.

Como regla general, una vez que se ha cumplido el tiempo equivalente a los 100 [cm], 5800 [us], se corta la alimentación del sensor hasta 20 [ms] antes de cumplir el periodo de muestreo, tiempo que se le otorga para el encendido y estabilización.

Las pruebas se realizan a 3 frecuencias de muestreo: 4 [Hz], 5 [Hz] y 10 [Hz]; empleando 2 objetos para las pruebas: un separador de libros de vidrio (Figura [4.11-a](#)) y un cartucho de videojuego portátil de unos 4 [cm²] (Figura [4.11-b](#)). Se pusieron 3 marcas de control a distancias de 2, 10 y 100 [cm] para las pruebas.



Figura 4.11: *Elementos de prueba para conocer el comportamiento del sensor. Un separador de libros, a), que tiene un área bastante grande para ser detectado y un cartucho de videojuegos, b), muy pequeño en comparación.*

En las Figuras 4.12 y 4.13 se muestran las pruebas a 4 [Hz], en las Figuras 4.14 y 4.15 se muestran las pruebas a 5 [Hz] y en las Figuras 4.16 y 4.17 se muestran las pruebas a 10 [Hz]. Las pruebas realizadas con el separador de libros (Figuras 4.12, 4.14 y 4.16) son las más estables, donde existen lapsos en los que el separador se encuentra estático, por lo que las mediciones son muy consistentes.

Las pruebas realizadas con el cartucho (Figuras 4.13, 4.15 y 4.17) son bastante inestables, debido a que hay instantes en los que el cartucho sale del ángulo de medición del sensor, lo que provoca que la medición se dispare hasta la distancia máxima limitada, los 100 [cm].

En las gráficas se puede apreciar que a medida que aumenta la frecuencia de muestreo, la respuesta del sensor es más rápida y acertada frente a cambios, pero aumenta el ruido de la señal y es más susceptible a tener brinco, como los que se aprecian en las mediciones del cartucho, que se asemeja más a la detección de objetos en movimiento.

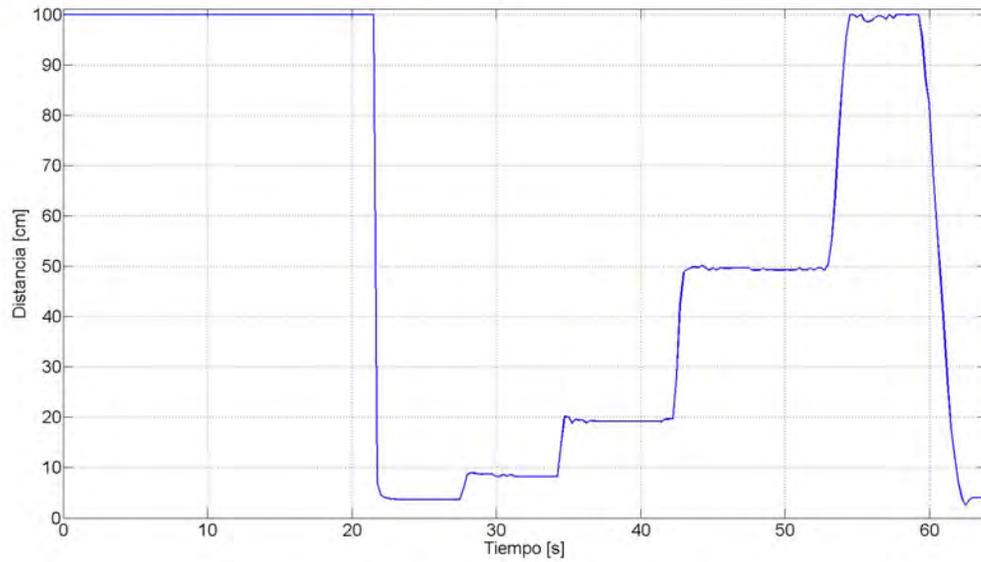


Figura 4.12: *Prueba con el separador de libros a 4 [Hz]. Al ser un objeto grande, el nivel ruido es muy bajo.*

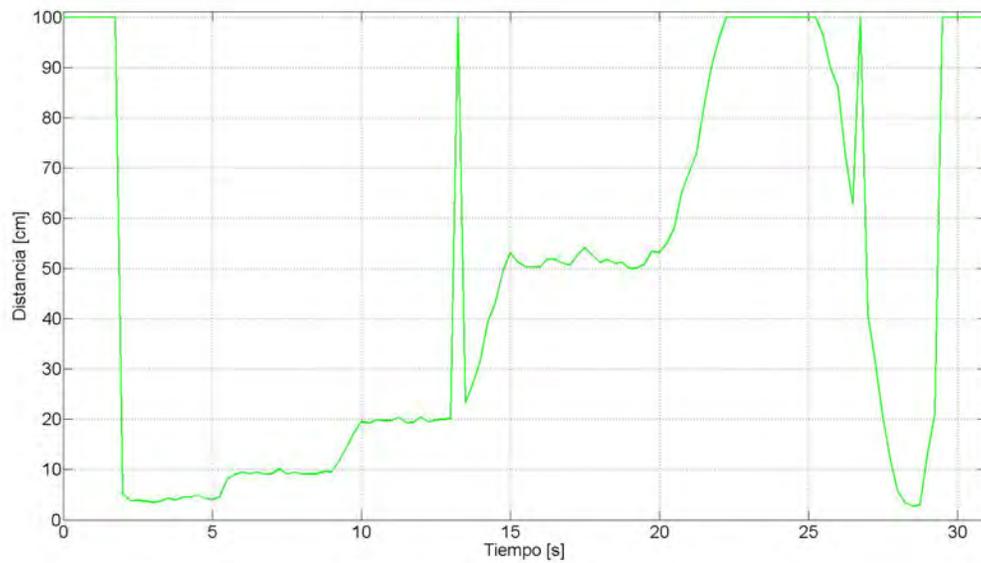


Figura 4.13: *Prueba con el cartucho a 4 [Hz]. El nivel de ruido se incrementa en comparación con el separador y se aprecian un par de picos donde el sensor dejó de detectar el cartucho.*

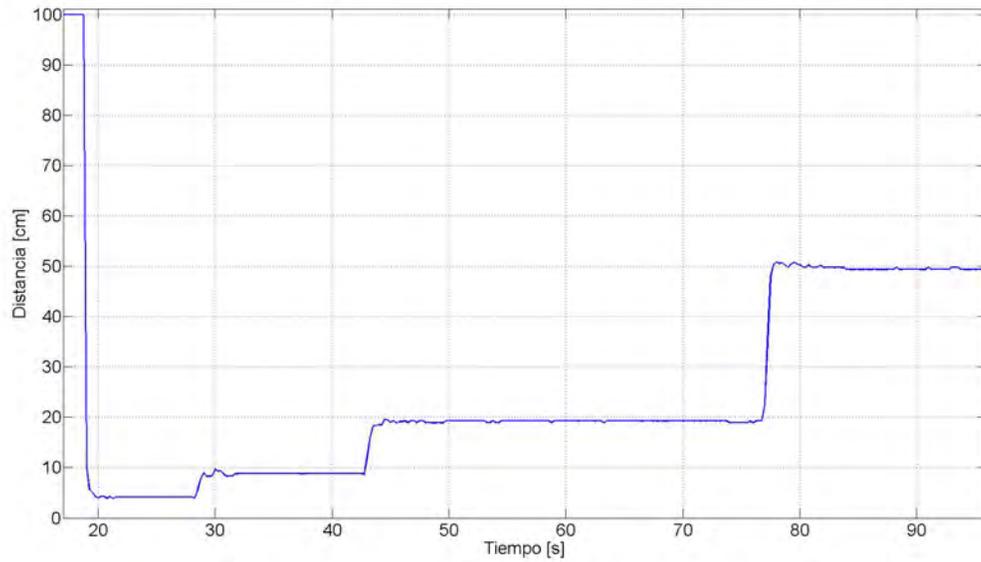


Figura 4.14: Prueba con el separador de libros a 5 [Hz]. El nivel de ruido incrementa muy ligeramente, la señal se mantiene prácticamente constante a las diferentes distancias de sensado.

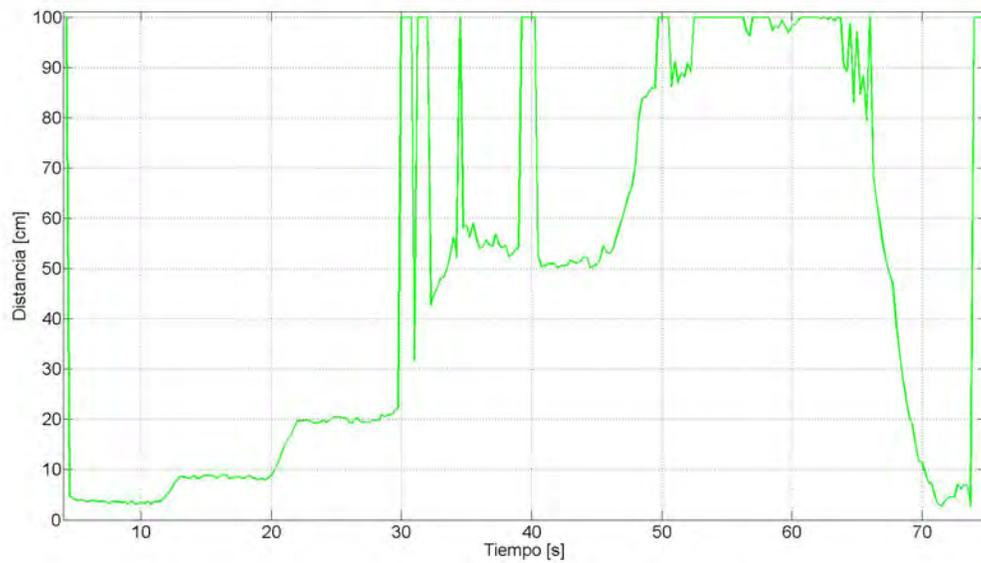


Figura 4.15: Prueba con el cartucho a 5 [Hz]. El nivel de ruido aumentó notablemente a distancias superiores a los 50 [cm] y se pierde la lectura en varias ocasiones.

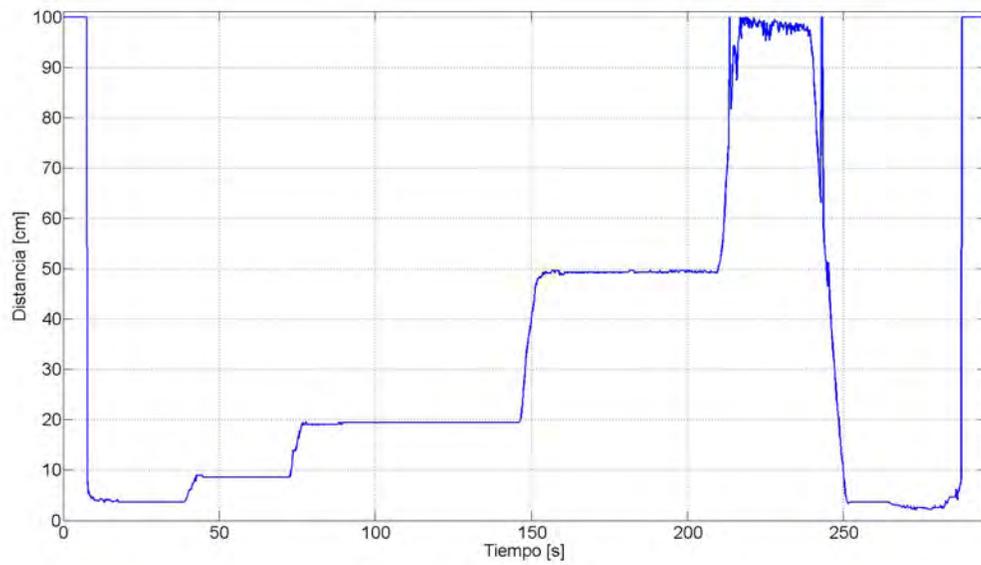


Figura 4.16: *Prueba con el separador de libros a 10 [Hz]. Aunque el nivel de ruido es prácticamente similar a distancias cortas, a distancias lejanas se pierde la lectura del sensor, donde es notorio que al incrementar la frecuencia de muestreo se incrementa el nivel de ruido.*

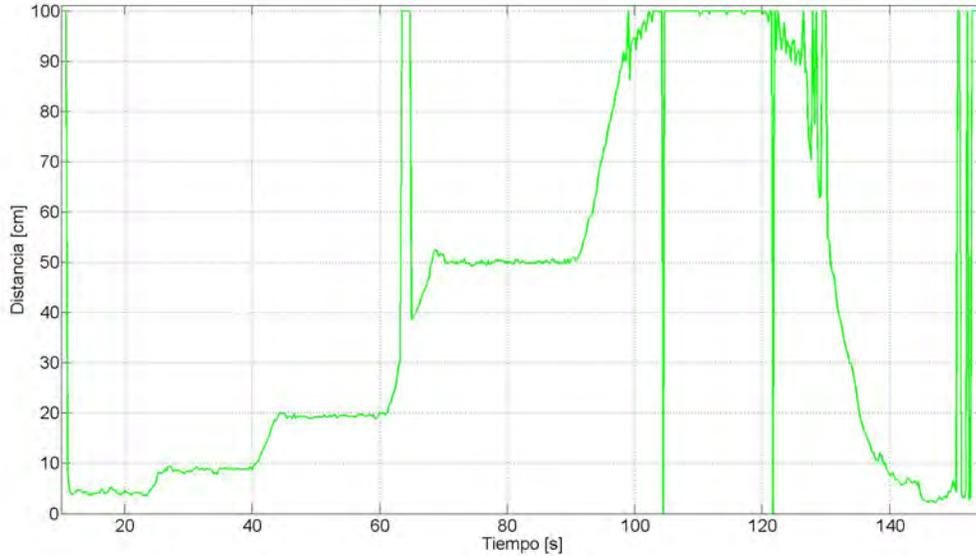


Figura 4.17: Prueba con el cartucho a 10 [Hz]. El nivel de ruido se incrementa notablemente a diferentes distancias y la lectura se pierde en constantes ocasiones, forzando a que a esta frecuencia de muestreo sea necesario implementar un filtro más fuerte.

4.2 Pruebas a los actuadores

Se realizaron pruebas a los actuadores, donde se encuentran las particularidades de los LEDs RGB y se hacen pruebas de velocidad a los micro motores, que servirán de apoyo para caracterizar la velocidad del motor.

4.2.1 Encendiendo los LEDs RGB

Con el fin de caracterizar los colores de los LEDs y la intensidad lumínica, se hace una primera prueba empleando como unidad de potencia un IC *ULN2803A*, un circuito integrado que se compone de 8 transistores tipo darlington con emisores comunes y diodos de libre rodada para cargas inductivas.

Se aplica como señal de control a cada color una señal de PWM virtual a 60 [Hz], generada desde un microcontrolador. El resultado inicial es un brillo muy intenso a corriente y voltajes nominales y una mala “mezcla” de los colores dentro del encapsulado. Dado que la cubierta epóxica de los LEDs no es opaca, los colores se ven separados por secciones dentro del encapsulado y no se aprecia una mezcla uniforme en la combinación de colores. En la Figura 4.18 se aprecian un par de imágenes de la primera prueba.

Con el fin de mejorar la miscibilidad de colores dentro del encapsulado del LED se optó por utilizar laca para vitral, Figura 4.18, después de realizar pruebas infructíferas con pintura vinílica.

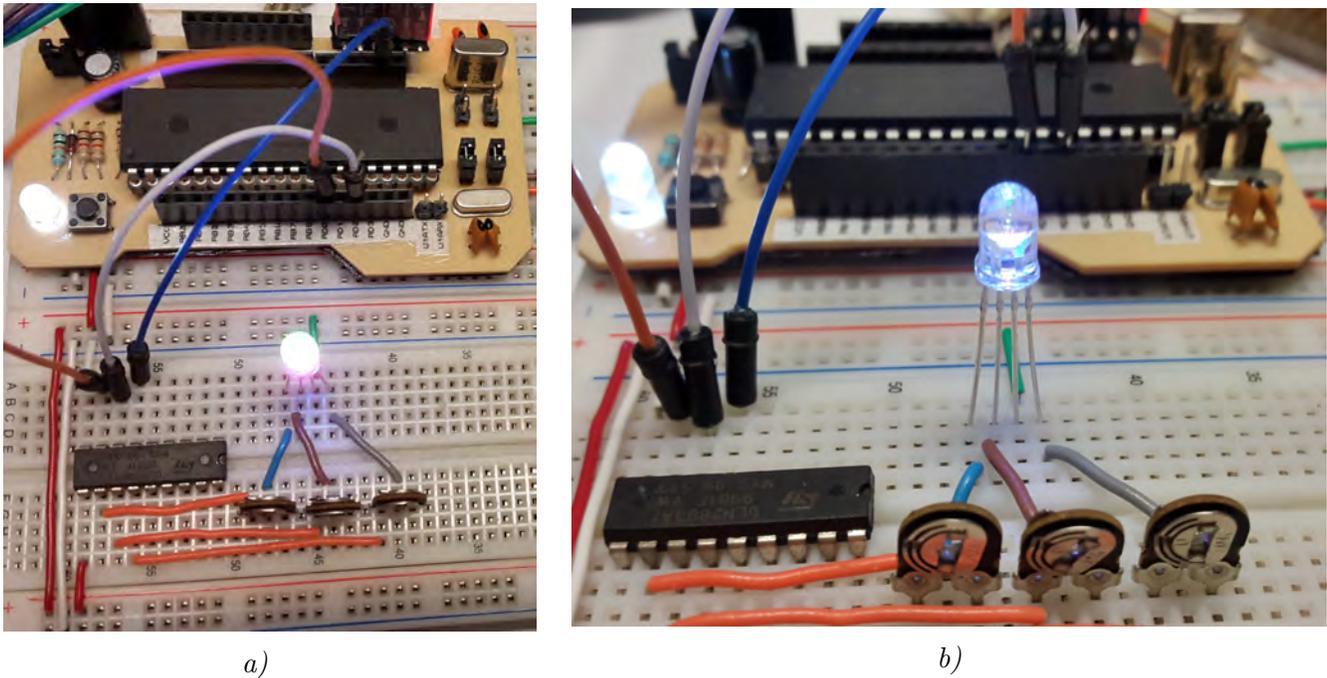


Figura 4.18: *Imágenes de prueba del LED RGB, se intentan encender en color morado, a), y azul, b). No tienen tratamiento alguno y no hay una correcta miscibilidad del color.*

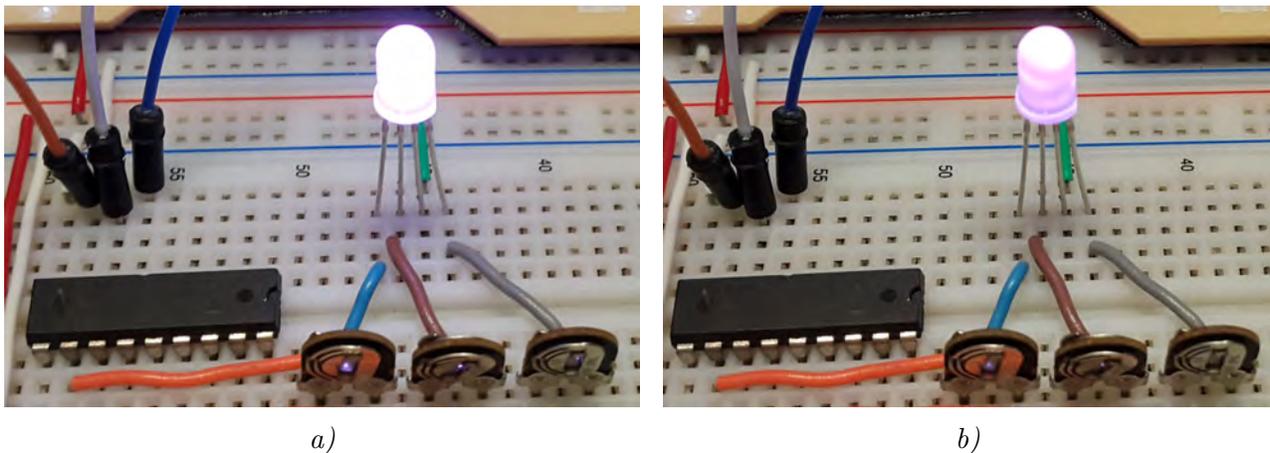


Figura 4.19: *Segunda prueba del LED RGB, se intentan encender en color blanco, a), y morado, b). Los LEDs recubiertos de laca para vitral para mejorar la miscibilidad.*

Una vez verificado que la laca para vitral sirve para mezclar dentro del encapsulado el color, la calibración (véase sección [4.3.1 Calibración de colores de los LEDs](#)) se realiza con LEDs ya pintados así. El inconveniente de esta pintura es que es muy delicada y no se fusiona en el encapsulado epóxico, por lo que cualquier rasguño puede tirar la laca de los LEDs.

4.2.2 Pruebas de velocidad de los motores

Se aplican escalones de voltaje al motor para conocer su comportamiento, a 1 [V] y a 5 [V]. El encóder se muestrea cada 2.5 [ms], 5 [ms] y 50 [ms]. Los valores de velocidad obtenidos se transmiten a través de un puerto serial, mismos que son guardados en un archivo para su posterior

análisis. Los datos se grafican en *Matlab*, donde los datos se grafican y presentan en las Figuras 4.20, 4.21, 4.22 y 4.23.

De las gráficas se puede concluir que los motores tienen un tiempo de asentamiento menor a los 100 [ms], dato que se emplea para la caracterización rudimentaria de los motores (véase sección B) Motores).

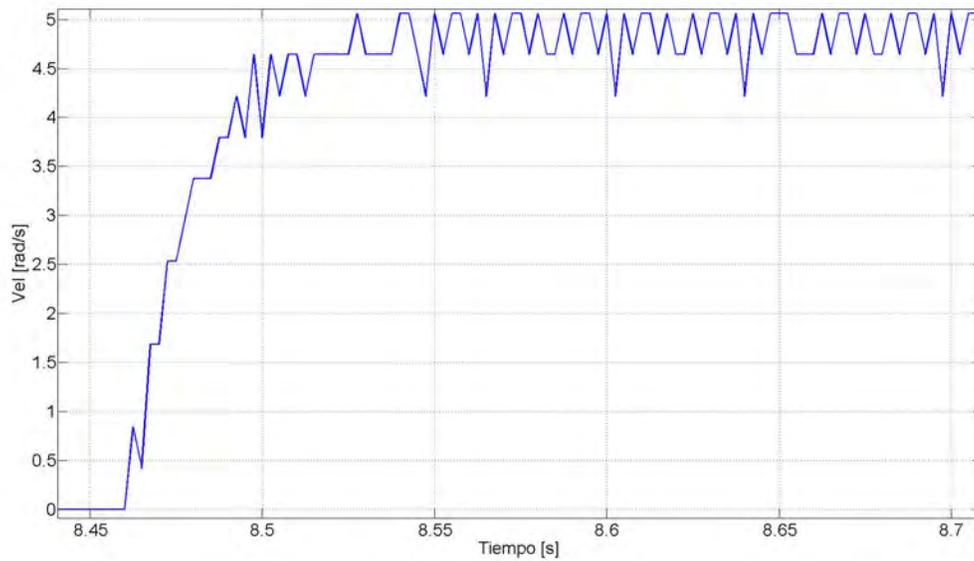


Figura 4.20: *Periodo de muestreo: 2.5 [ms]. Escalón: 5 [V].*

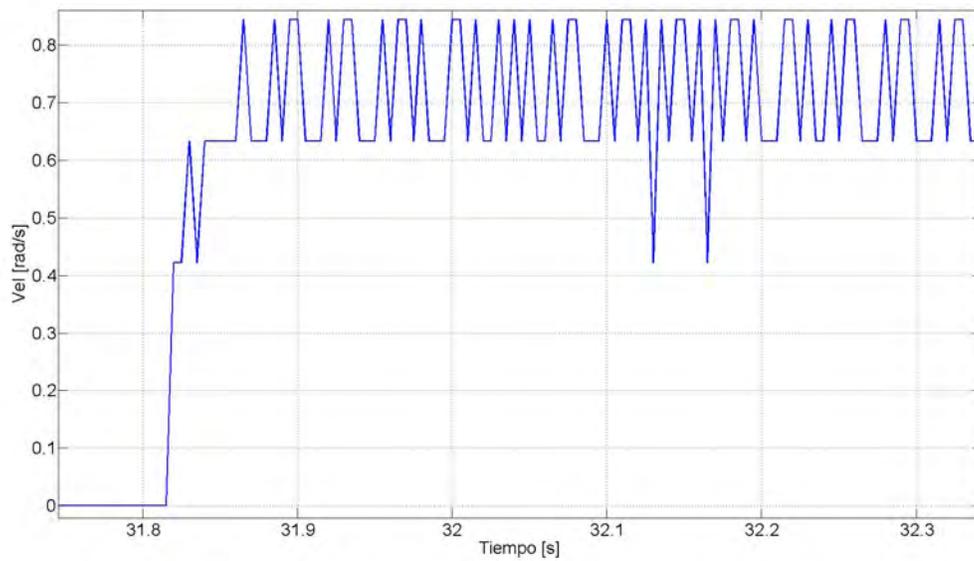


Figura 4.21: *Periodo de muestreo: 5 [ms]. Escalón: 1 [V].*

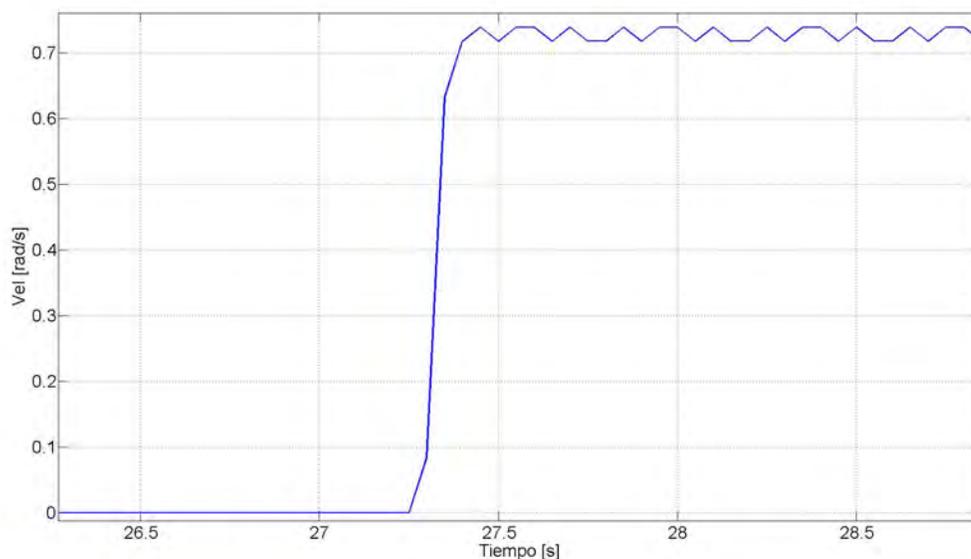


Figura 4.22: *Periodo de muestreo: 50 [ms]. Escalón: 1 [V].*

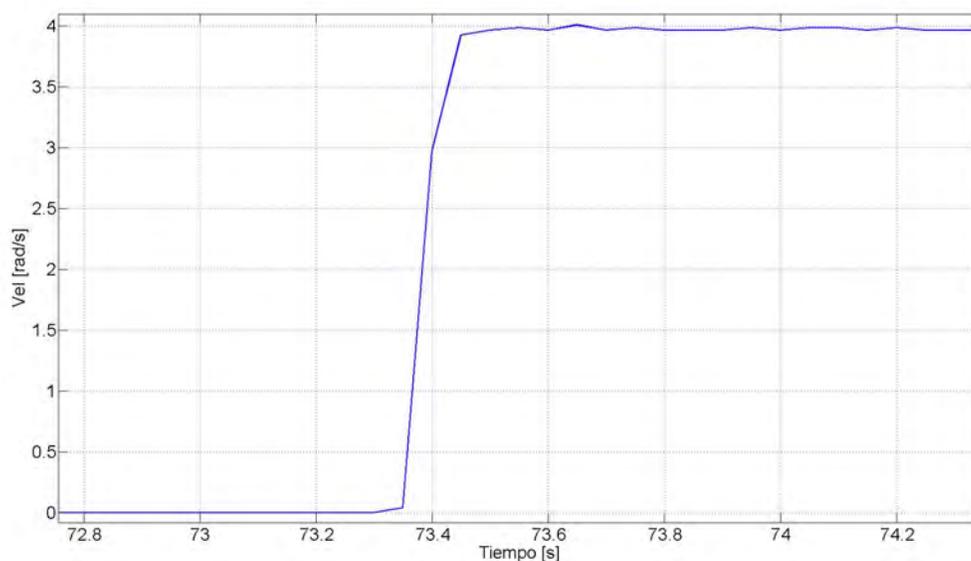


Figura 4.23: *Periodo de muestreo: 50 [ms]. Escalón: 5 [V].*

4.3 Calibración y prueba de los componentes finales

En esta sección se abarca la calibración de colores de los LEDs RGB y el resultado de la manufactura y pruebas de los componentes del robot.

4.3.1 Calibración de colores de los LEDs

Dado que el *PCA9956A* es de tecnología SMT y no existen módulos de desarrollo disponibles, la calibración se realiza una vez que se tienen fabricadas las placas del robot. Como primera prueba

se emplea un solo LED conectado a la placa, con el fin de observar el resultado de encender todos sus colores y verificar el funcionamiento de la placa. En la Figura 4.24 se aprecia el arreglo. El microcontrolador esclavo manda la señal de control a través del bus I²C para encender el LED.

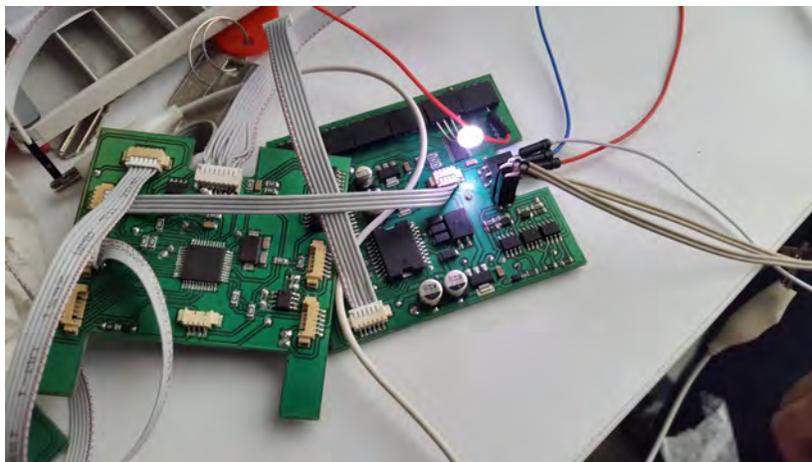


Figura 4.24: Prueba del **PCA9956A** con un LED. Se hace la prueba directamente empleando las placas del robot y se enciende en color blanco.

Después de una serie de pruebas, cuando se tienen conectados los 4 LEDs de una cara, una corriente total $I_{color} = 16$ [mA] limita la intensidad lumínica de los 4 LEDs lo suficiente para que sea tolerable pero que se vea llamativo a nuestra consideración.

De las pruebas se desprende que, al encender simultáneamente los colores rojo y verde, la corriente en ambas terminales disminuye; el color azul, no es afectado ni afecta el consumo de los otros dos colores. Los valores de calibración de corriente cuando el LED se encuentra totalmente encendido (PWM del 100 %) se muestran en la Tabla 4.2.

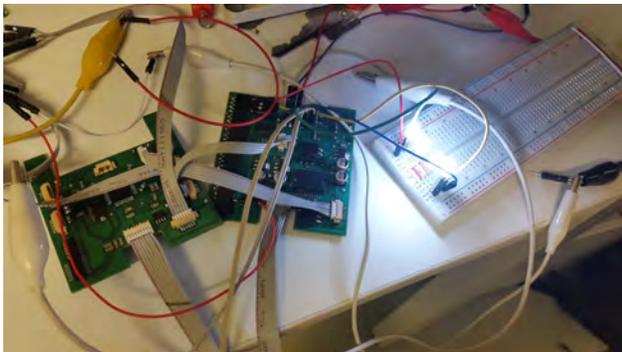
Tabla 4.2: Valores de configuración de corriente.

Color	Valor del registro	Corriente [mA]	
Rojo	V:✘	165	16.08
	V:✔	167	16.05
Verde	R:✘	166	16.03
	R:✔	169	16.06
Azul		162	15.98

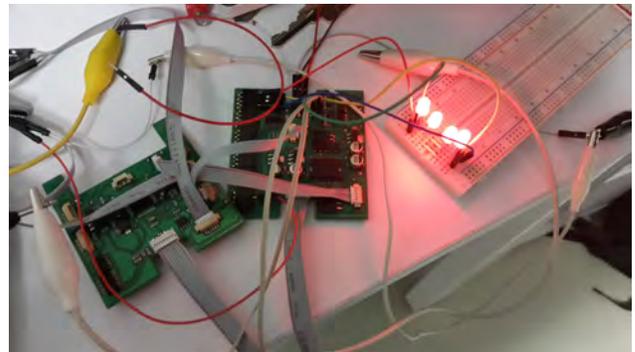
A partir de ello, se hace la calibración para 11 colores, mediante el ajuste del PWM, tomando en cuenta que las corrientes ya están configuradas. Los valores para obtener los colores se encuentran en la Tabla 4.3. Algunos de los colores se pueden apreciar en las Figuras 4.25 y 4.26.

Tabla 4.3: Colores calibrados.

Color \ RGB	Rojo	Verde	Azul
Blanco	255	255	255
Rojo	 255	0	0
Verde	 0	255	0
Azul	 0	0	255
Morado	 56	12	75
Naranja	 255	55	0
Amarillo	 255	160	0
Rosa	 255	90	120
Cyan	 0	255	255
Magenta	 255	0	255
Aquamarine	 50	255	212



a)

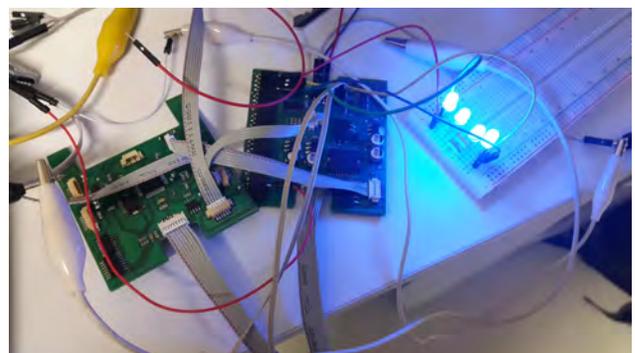


b)

Figura 4.25: Pruebas de calibración de colores de los LEDs RGB: a) blanco; b) rojo.



a)



b)

Figura 4.26: Pruebas de calibración de colores de los LEDs RGB: a) verde; b) azul.

4.3.2 Modelo físico del robot

El modelo físico se mandó imprimir mediante impresión 3D en colores azul y naranja para comprobar el diseño mecánico y corroborar que los diferentes componentes se pudieran ensamblar de manera correcta.

Las piezas diseñadas e impresas son el cuerpo central, las tapas y los *brackets*. Para comprobar el ensamble entre las piezas primero se comprueba que las tapas ajustan a las dimensiones del cuerpo central. Posteriormente, una vez insertadas las tuercas en sus respectivos compartimientos, los cubos se ensamblan con los tornillos. En la Figura 4.27 se aprecia que las tapas corresponden a las dimensiones del cuerpo central y ajustan sin problemas mediante la combinación de tornillo-tuerca.

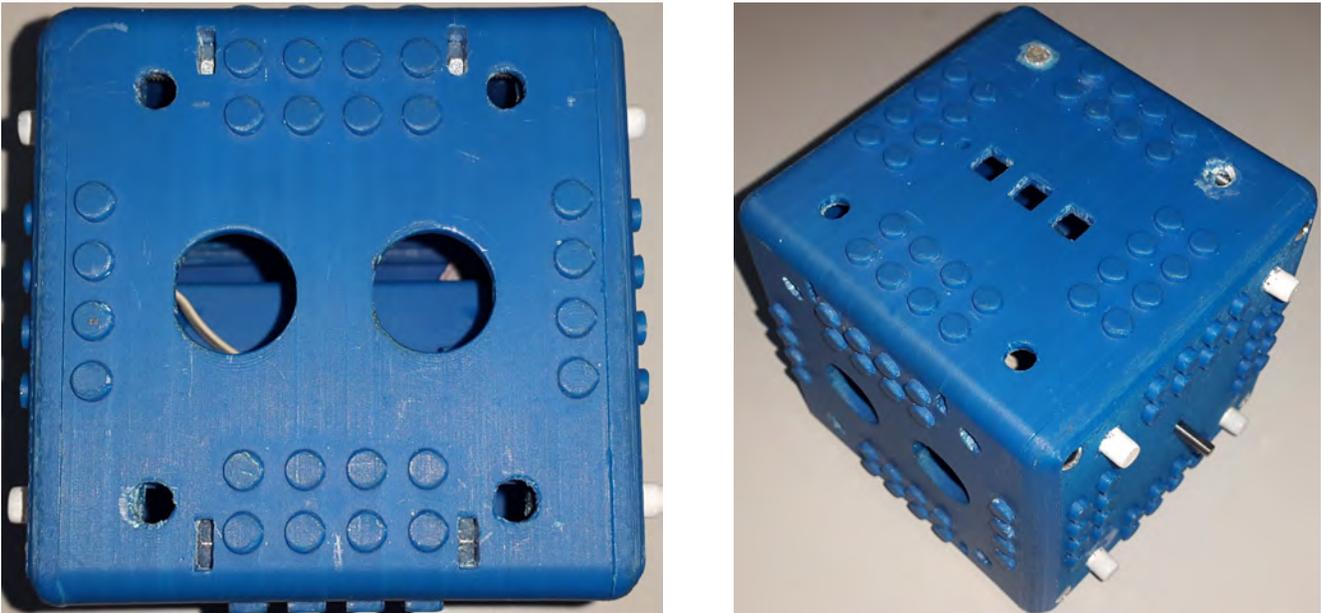


Figura 4.27: *En las fotos se aprecia que las tapas ajustan correctamente sobre el cuerpo central y se ensamblan de manera correcta empleando los tornillos.*

Se verifica que los motores y *brackets* se ensamblen de manera correcta sobre las tapas. En la Figura 4.28 se muestra el ensamble de un *bracket* en una tapa de manera correcta. De igual forma, en la Figura 4.29 se aprecia el arreglo de motor-*bracket*-tapa.

4.3. Calibración y prueba de los componentes finales

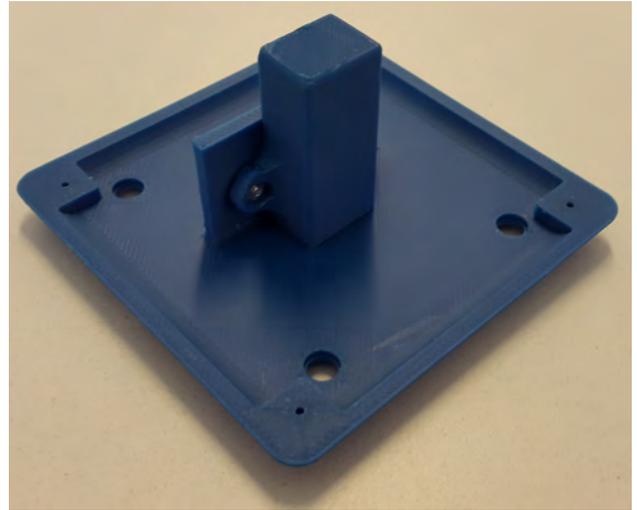
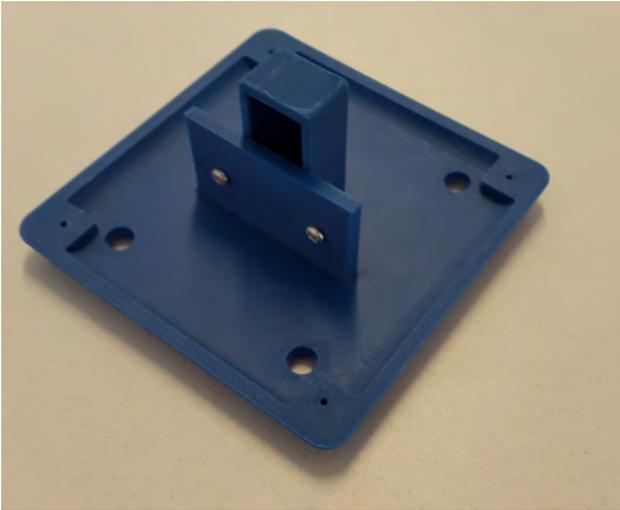


Figura 4.28: *En las fotos se aprecia que los brackets se ensamblan de manera correcta en el reverso de las tapas mediante su ajuste de tornillo-tuerca.*

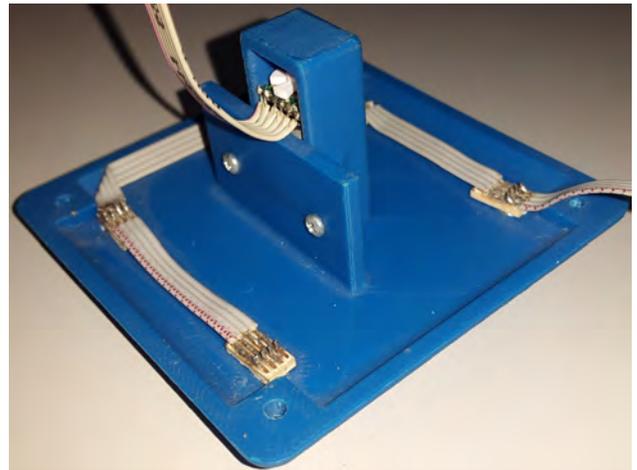
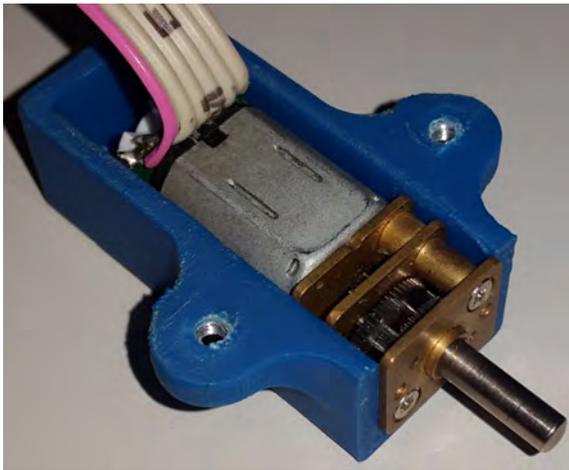


Figura 4.29: *En las fotos se aprecia que el motor se ensambla correctamente tanto al bracket como a la tapa.*

Finalmente se montan las llantas a las flechas de los motores. Las llantas no interfieren con los pines tipo *LEGO* y se aprecia el modelo físico del robot completo (Figura 4.30).



Figura 4.30: *Los LEDs fueron recortados lo suficiente para no interferir con la estructura de la llanta.*

4.3.3 Ensamble de la electrónica dentro del cuerpo central

El sensor más voluminoso es el sensor ultrasónico, que se determinó que se montaría en el robot directamente en los barrenos en la cara frontal del robot. A pesar de que los sensores conseguidos no tienen una alta calidad de manufactura y las dimensiones no concuerdan con los esquemas, los sensores se sostienen firmemente en los barrenos de la cara, por lo que se cumple el cometido de eliminar la necesidad de adicionar al robot soportes para los tornillos que sostengan la placa del sensor a la cara del robot. En la Figura 4.32 se aprecia el sensor montado dentro del robot.

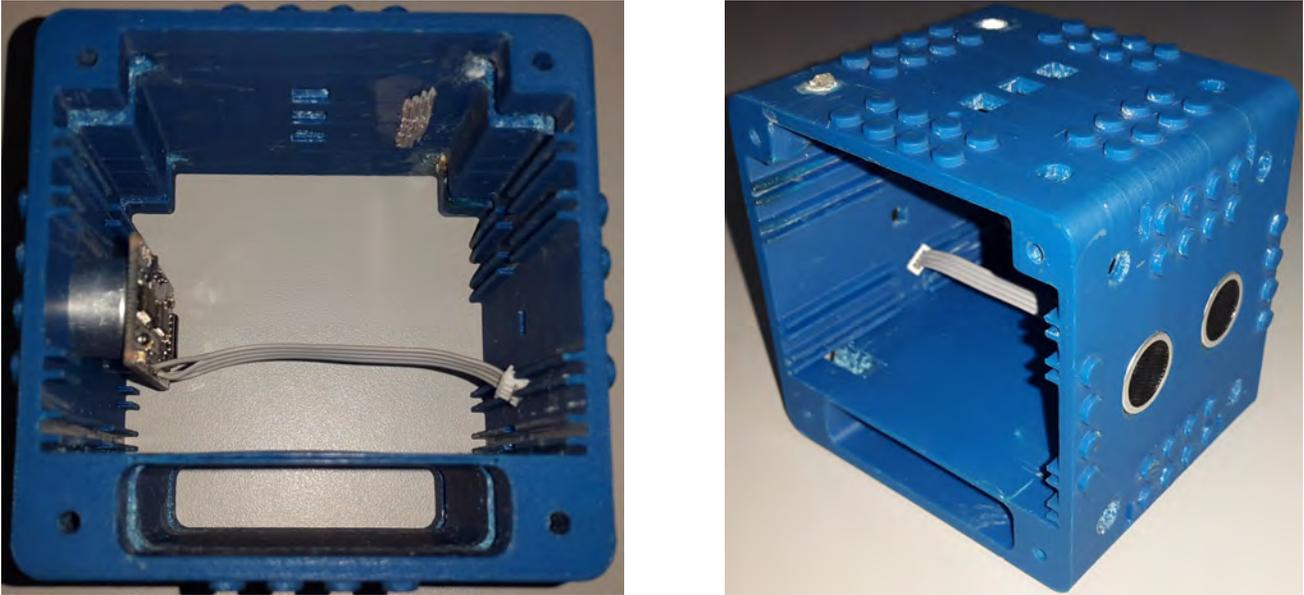


Figura 4.31: *En las fotos se aprecia el sensor ultrasónico montado dentro el robot, en el reverso de la cara, sostenido directamente dentro de los barrenos dedicados al emisor y receptor del sensor.*

Una vez montado el sensor, se insertan las placas electrónicas, corroborando que todas pueden ser insertadas a pesar de que el sensor ultrasónico se encuentra montado. Las placas se deslizan correctamente en los rieles destinados a cada una de ellas. A continuación se insertan las baterías en el compartimiento destinado. Las baterías se deslizan correctamente sin forzar las paredes que las confinan. En la Figura 4.32 se aprecian las placas y las baterías montadas en el cuerpo del robot junto con el sensor ultrasónico.

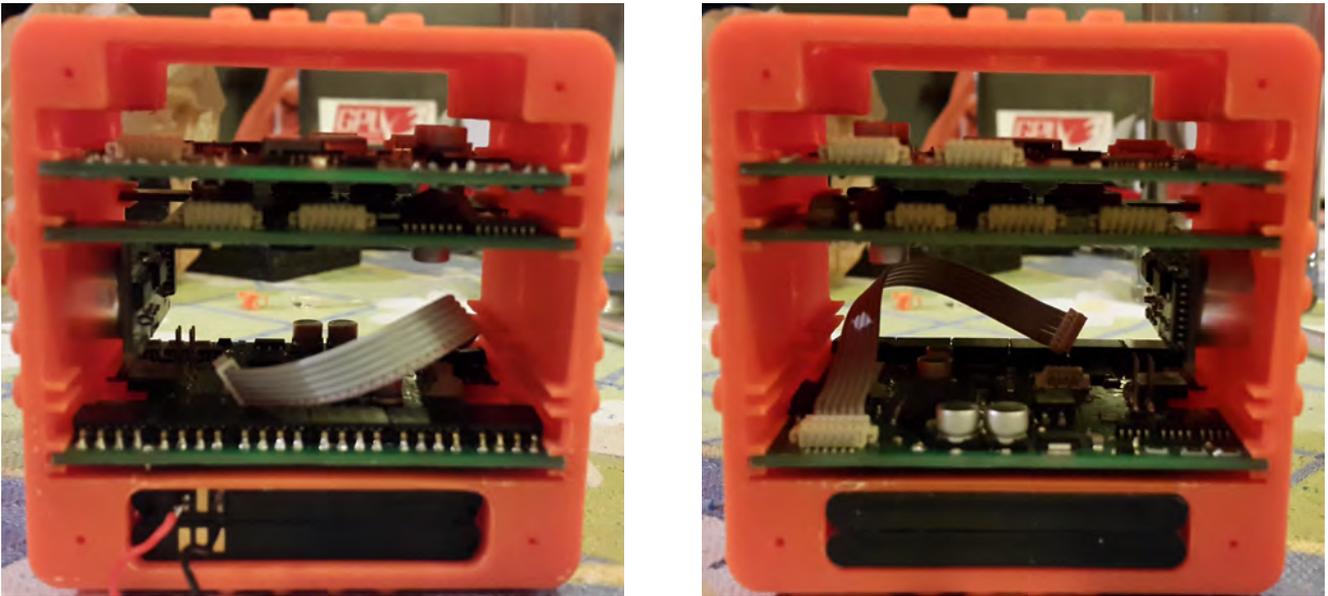


Figura 4.32: Las placas electrónicas y las baterías se insertan sin problemas a pesar de lo voluminoso del sensor ultrasónico.

Durante el posicionamiento de los LEDs se detectó que hay un error en el diseño del espacio destinado alrededor a las baterías. Como se puede apreciar en la Figura 4.33, existe un error en el espacio para las placas al faltar un par de milímetros menos en la zona del encapsulado y anillo. Se corrige ampliando un par más de milímetros el rectángulo, sin embargo, el modelo físico ya había sido impreso, por lo que se optó por lijar la zona ligeramente para conseguir el espacio suficiente para colocar las placas con LEDs. El resto de los LEDs se ensamblan sin problema.

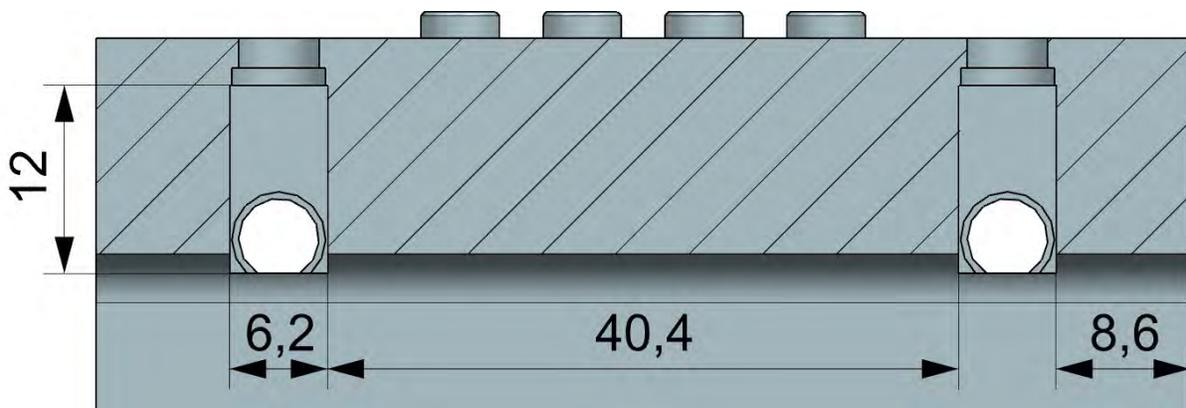


Figura 4.33: Un error en el diseño mecánico evita que los LEDs cercanos a las baterías se puedan ensamblar directamente, se corrige ampliando el rectángulo de recorte.

El principal problema a la hora de ensamblar todas las placas, sensores, baterías y cables es que el proceso es largo y muy complejo; la organización actual de los conectores no permite que el proceso sea fácil y muchos conectores se podrían dañar si se ensamblan y desmantelan los robots seguidos.

4.3.4 Pruebas al módulo de procesamiento y comunicación

El módulo de procesamiento se compone del conjunto **Maestro** y **Esclavo**. Se realizaron varias pruebas para probar el funcionamiento del **Maestro** en conjunto con el **Esclavo** para poder analizar si ambos microcontroladores podían comunicarse de manera efectiva entre ambos y si la red wifi que puede crear el *RN171* funciona como es deseado.

Primero se revisa que el módulo de comunicación se encuentre configurado como se describe en la sección [A\) RN171](#). El punto de acceso debe tener de nombre *CU-10-80-T* y ser abierta para que cualquier dispositivo se pueda conectar. Esto se revisa empleando la herramienta integrada la computadora para buscar redes inalámbricas, cuyo resultado se aprecia en la [Figura 4.34](#).

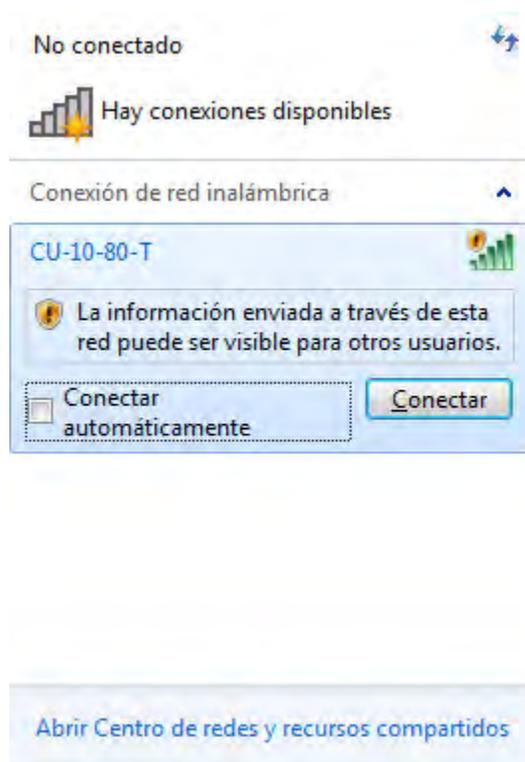


Figura 4.34: *El punto de acceso se configuró de manera correcta, mostrando el SSID configurado y es una red abierta.*

El programa almacenado en memoria para el **Maestro** es simple y tiene la finalidad de ir probando uno a uno los módulos de sensado y actuación y, finalmente, una prueba en donde estén encendidos todos los sensores del módulo de sensado para comprobar que, aunque la comunicación con el *RN171* es mucho más lenta que con la comunicación entre ambos microcontroladores, el enviar un número limitado de caracteres por la red inalámbrica no genera un cuello de botella.

Las pruebas exhaustivas a los módulos de sensado y actuación se hicieron inicialmente de manera independiente (sección [4.3.5 Pruebas a los módulos de sensado y actuación](#)); con el conocimiento de que los módulos de sensado y actuación funcionan de manera correcta se procedió a realizar pruebas con el enlace entre ambos microcontroladores.

Las pruebas fueron grabadas desde una computadora conectada a la red generada por el *RN171*

a través de una terminal *Telnet*.

El primer paso del **Maestro** es configurarse en modo “Eco” para transmitir por el wifi toda la información que el **Esclavo** le arroje por el puerto serial. En la Figura 4.35-a se aprecia una primera prueba donde el programa consiste en activar únicamente el acelerómetro y giroscopio de la IMU; se muestran en la Figura el instante desde que el **Esclavo** responde a la activación del **Maestro**. Posteriormente de mandar los comandos para encender ambos sensores se aprecian las lecturas completas que realiza la IMU.

En la Figura 4.35-b se aprecia la información que el **Esclavo** transmite al **Maestro** de todo el módulo de sensado. En la Figura se puede apreciar la diferencia entre las frecuencias de muestreo de la IMU y encoders con respecto al sensor ultrasónico, con una relación de 4:1. A pesar de que el número de caracteres que recibe el **Maestro** es amplio, no hay un cuello de botella entre el **Maestro** y el *RN171*.

```

dsPIC
T:38
T:38
T:38
T:38
T:38
T:38
A
G
A:-0.05
A:-0.00
A:-1.01
T:39
G:0
G:0
G:0
A:-0.05
A:-0.00
A:-1.02
T:39
G:0
G:0
G:0
A:-0.05
A:-0.00
A:-1.01
T:39
G:0
G:0
G:0
A:-0.05
A:-0.00
A:-1.02
T:39
G:0
G:0
G:0
A:-0.05
A:-0.00
A:-1.01
T:39
G:0
G:0
G:0
A:-0.05
A:-0.00
A:-1.02
T:39
G:0
G:0
G:0
A:-0.05
A:-0.00
A:-1.01
T:39
G:0
G:0
G:0

```

a)

```

U:32
A:-0.04
A:-0.00
A:-1.01
T:45
G:-3
G:0
G:-26
L:2.80
R:0
A:-0.04
A:-0.00
A:-1.01
T:45
G:-3
G:0
G:-26
L:2.78
R:0
A:-0.05
A:-0.00
A:-1.01
T:45
G:-3
G:0
G:-26
A:-0.04
A:-0.00
A:-1.02
T:45
G:-3
G:0
G:-26
L:2.80
R:0
U:37
A:-0.05
A:-0.00
A:-1.02
T:45
G:-4
G:0
G:-26
L:2.78
R:0
A:-0.05
A:-0.00
A:-1.02
T:45
G:-4
G:0
G:-27
A:-0.04
A:-0.00
A:-1.02
T:45
G:-4
G:0
G:-27

```

b)

Figura 4.35: La comunicación entre todos los módulos del robot funciona a pesar de los múltiples protocolos y velocidades de transmisión. El **Maestro** tiene la capacidad de leer la memoria y mandar las instrucciones al **Esclavo**.

4.3.5 Pruebas a los módulos de sensado y actuación

El microcontrolador que se encarga de ambos módulos es el **Esclavo** que funge como intermediario leyendo la información de los módulos de sensado y actuación y transmitiéndola al **Maestro** cuando es requerido.

Las pruebas a ambos módulos inicialmente se realizan conectando las terminales del UART del **Esclavo** que lo conectan con el **Maestro** a un convertidor serial UART-USB a una velocidad de 921600 baudios. El **Esclavo** arroja toda la información que recolecta al instante con el fin de analizar los datos de los diferentes sensores. Con fines de prueba, el **Esclavo** responde a un cualquier para confirmar que encendió correctamente con el texto *dsPIC*. Cada comando correcto que recibe dará como resultado que mande el eco del caracter recibido.

En las pruebas finales se realiza la conexión entre ambos microcontroladores para verificar el correcto funcionamiento de ambos en conjunto, descrito en el apartado anterior [4.3.4 Pruebas al módulo de procesamiento y comunicación](#).

A) Encoders y motores

La prueba de los encoders se realizó conectando un motor a la terminal del motor izquierdo, montado sobre el *bracket* que lo sostendrá dentro del robot, como el que se aprecia en la Figura 4.29; el motor derecho no se conecta para mostrar como referencia la velocidad nula del motor. El motor se enciende dando la combinación de caracteres ‘*M*’, ‘*L*’, un valor numérico en punto flotante (4 números en hexadecimal de la forma *0xNN*) y ‘*E*’. En la Figura 4.36 se aprecia que al encender el motor, el encóder izquierdo muestra un cambio en la velocidad del motor, a diferencia del motor derecho que no presenta cambios. Tal como se aprecia en las gráficas de caracterización (Figuras 4.20, 4.21, 4.22 y 4.23), las primeras lecturas son altamente ruidosas, aunado a que se está frenando la flecha de salida; después de liberar la flecha y dejar estabilizar la velocidad del motor, la velocidad se aprecia hasta cierto punto constante (Figura 4.37). Al medir el voltaje que el puente H suministra a los motores se encontró que no llega a suministrar los 5 [V] que mantiene el regulador de potencia, lo cual provoca que los motores no alcancen la velocidad de 3.9 [rad/s] que llegan a alcanzar cuando se les suministran los 5 [V]. Típicamente, los motores bajo la configuración diseñada alcanzan velocidades de hasta 2.9 [rad/s].

Se necesitarían más pruebas para determinar el error para poder ser corregido en futuras iteraciones; en dado caso de que el error sea provocado porque tanto la terminal lógica como de potencia en el puente H están al mismo voltaje, la terminal de potencia del puente H tendría que ser conectada a la terminal de la segunda batería. Sin embargo, eso provocaría que el control de velocidad que se llegue a programar para los motores deba contemplar que la alimentación no es constante y varía con el tiempo de manera irregular.

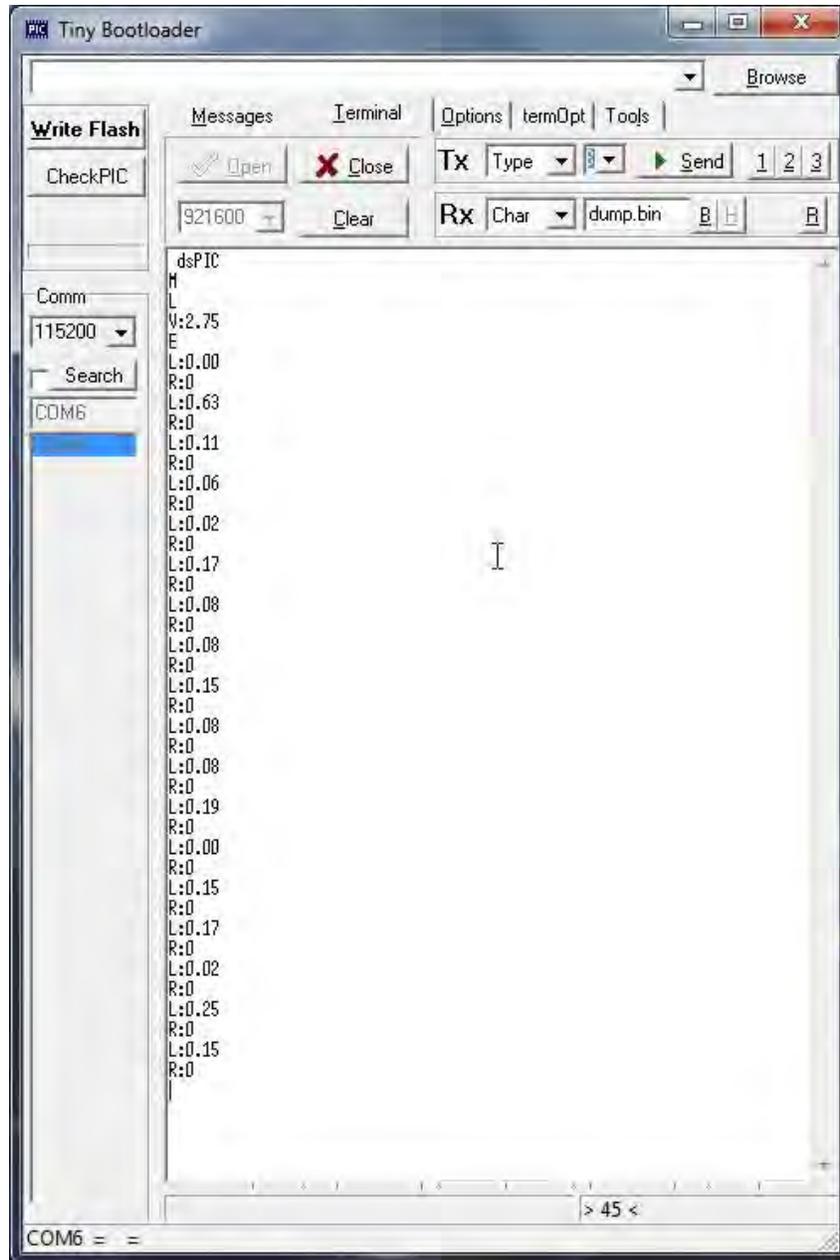


Figura 4.36: Encendido del motor y prueba frenando la flecha del motor; las primeras lecturas son muy ruidosas.

4.3. Calibración y prueba de los componentes finales

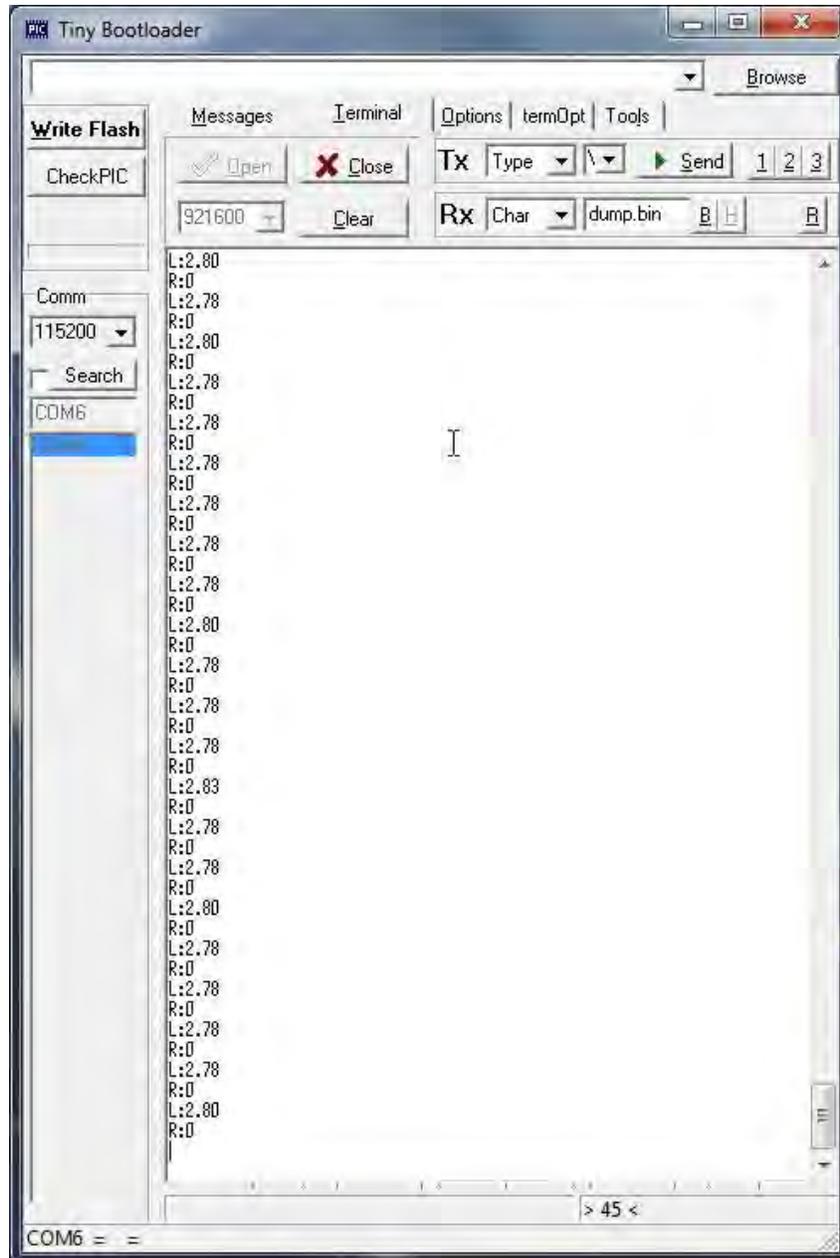


Figura 4.37: Una vez que se libera la flecha del motor, éste se estabiliza y alcanza una velocidad donde la lectura es cuasi estable, aunque se aprecia un poco de ruido en la lectura.

B) IMU

La IMU tiene la versatilidad de contener un termómetro embebido en el mismo empaquetado que permite supervisar la temperatura interna del cubo. Durante las pruebas se detectó que los microcontroladores llegan a alcanzar temperaturas alrededor de los 50 [°C]. Este nivel de calentamiento está presente en todos los modelos de la gama *dsPIC30F* que alcanzan los 30 MIPS en empaquetados de montaje superficial; las pruebas realizadas con empaquetados *through-hole* no alcanzan tal nivel de temperatura aunque sean empleados a su máxima capacidad de procesamiento.

La IMU se encuentra a contracara del **Esclavo**, por lo que permite monitorear una temperatura similar a la que se encuentra el microcontrolador; no se puede saber con exactitud la temperatura del microcontrolador de manera directa porque la PCB alcanza a disipar parte de la energía.

Durante el funcionamiento, la temperatura es siempre monitoreada. En la Figura 4.38 se muestra una de las pantallas de muestreo, donde la temperatura en este caso alcanzó los 44 [°C]. Todos los circuitos integrados tienen un límite de temperatura en funcionamiento de 80 [°C], y aunque la IMU no puede medir de manera directa la temperatura de todos los circuitos, sí puede medir una temperatura general de todo el cubo, que permitiría implementar un código de emergencia para prevenir sobrecalentamiento interno.

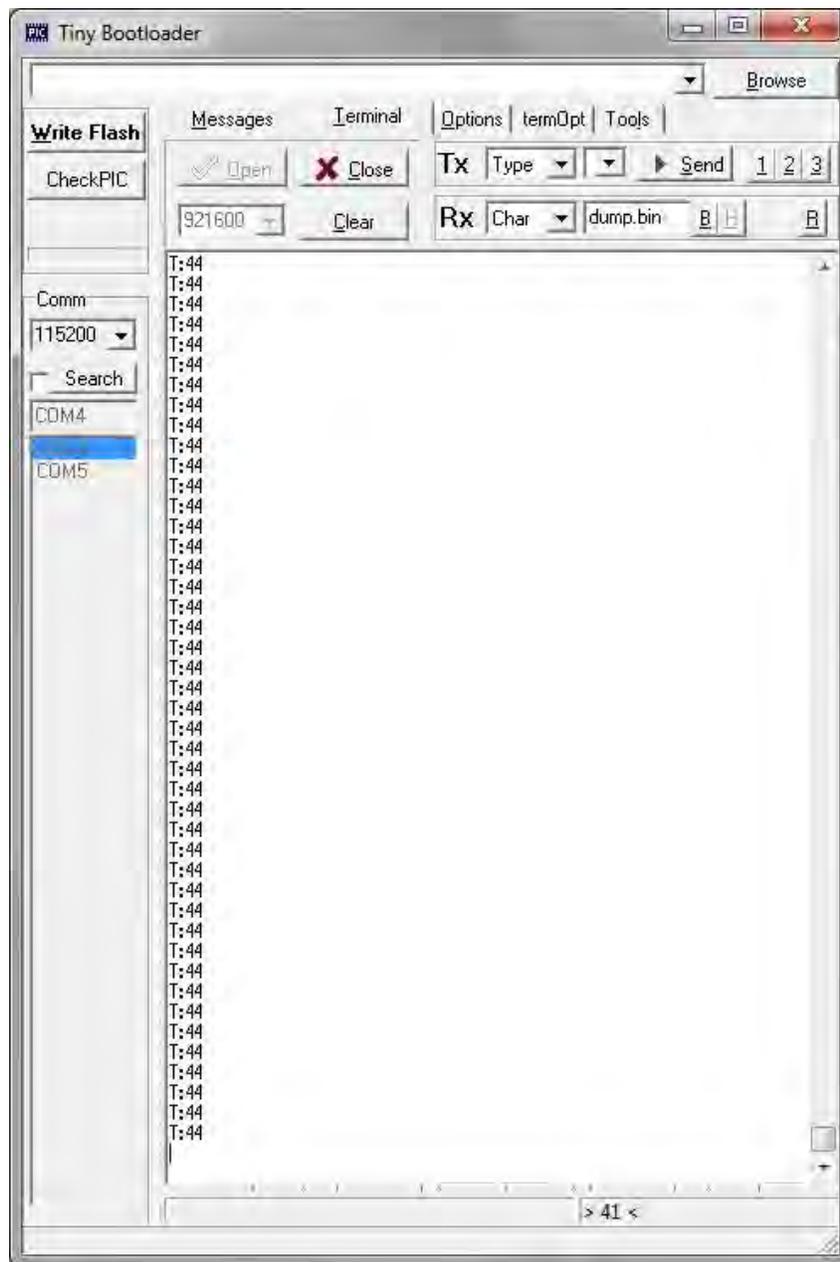


Figura 4.38: La IMU contiene un termómetro que permite monitorear la temperatura general interna del robot, necesario dado que los microcontroladores se calientan a temperaturas muy altas por el simple hecho de estar funcionando.

El acelerómetro no tiene programado ningún filtro ni protocolo de reacción ante ciertos valores o umbrales de movimiento, por lo que únicamente se corrobora que funcione y que el valor de la vertical del eje Z sea cercano en magnitud a 1. En la Figura 4.39 se aprecia una muestra de datos recolectados; van entrelazados con la información del termómetro, porque aunque es optativo recolectar la información del acelerómetro y giroscopio, la temperatura siempre se recolecta. El valor que arroja del eje 'Z' es correcto y tiene signo invertido dada la posición en la que se encuentra montada la IMU con respecto al robot, es decir, de cabeza, dado que se encuentra en el reverso de la placa del **Esclavo**.

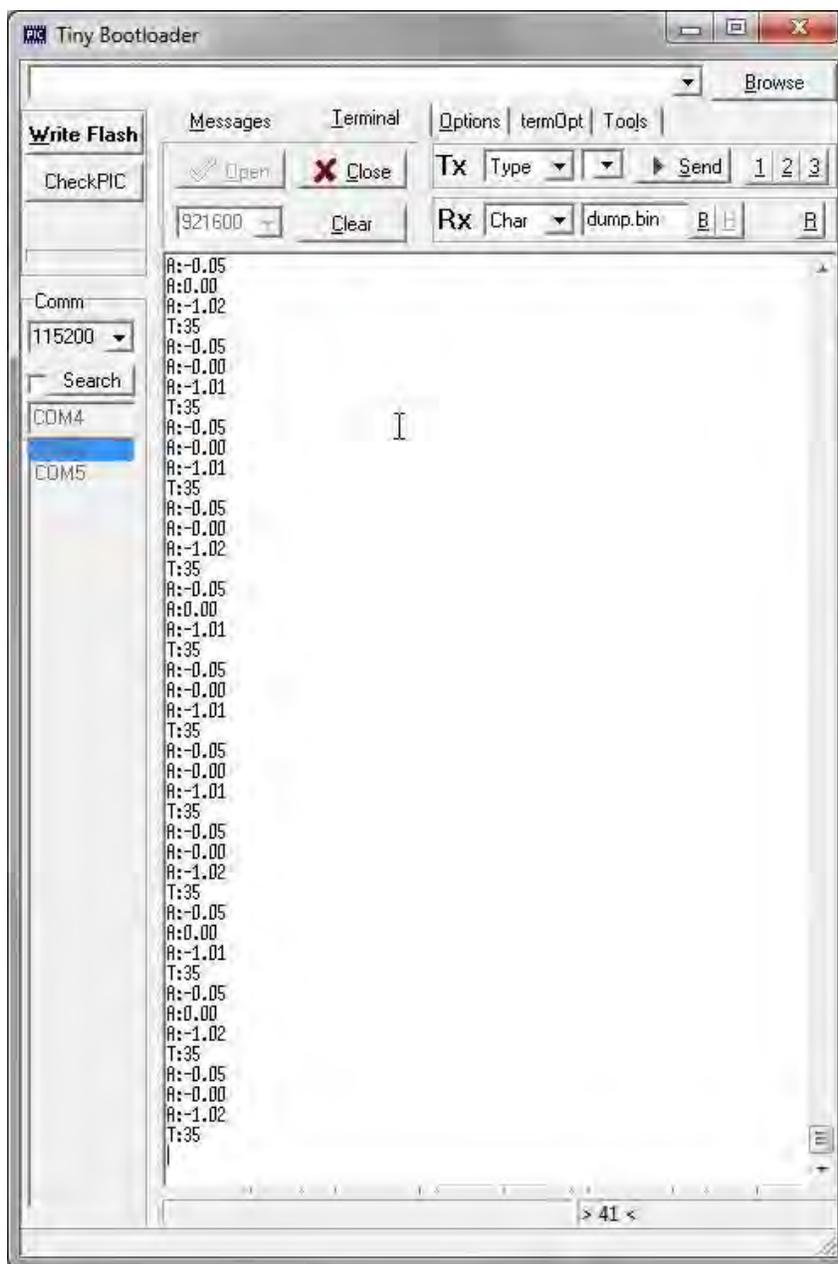


Figura 4.39: Actualmente el acelerómetro sólo se encuentra instalado en el robot, no hay ningún filtro ni protocolo programados que interprete la información que brinda.

Para el giroscopio se programó un filtro muy sencillo que promedia 50 lecturas para eliminar el error permanente en los ejes y que la lectura se desvirtúe rápidamente de tan sólo estar encendido. El filtro logra que el robot pueda ser rotado con una velocidad relativamente lenta sin que se pierda la referencia de los ejes y da una lectura bastante acertada; sin embargo, luego de movimientos bruscos o después de mucho tiempo, la lectura se vicia y la referencia original se pierde. Para corregir eso, cada que el giroscopio es encendido, se reinicia la posición actual. En la Figura 4.40 se aprecia una pequeña muestra de lo que arroja el giroscopio, donde se rotó al robot alrededor de 2 ejes; si se evitan los movimientos bruscos, la lectura es consistente.



Figura 4.40: El giroscopio tiene un filtro muy sencillo que elimina el error permanente. Si se evitan movimientos bruscos las lecturas del giroscopio son bastante consistentes y acertadas.

C) Sensor ultrasónico

El sensor ultrasónico está calibrado para limitar la lectura si no detecta obstáculo alguno hasta los 50 [cm], con la finalidad de evitar tener brincos muy marcados en la lectura. A pesar de ello, inclusive al probarlo acercando y alejando la mano se puede notar que la lectura se pierde en repetidas ocasiones (Figura 4.41) y cada ocasión que la lectura se pierde el sensor se dispara a los 50 [cm], de ahí que se aprecien los picos en las gráficas de caracterización cuando se emplea el cartucho como objeto a detectar (Figuras 4.13, 4.15 y 4.17). Sólo objetos de gran tamaño generan una lectura poco ruidosa (Figura 4.42).

El sensor está calibrado a una frecuencia de muestreo de 5 [Hz], punto intermedio entre las frecuencias probadas, sin embargo, el nivel de ruido es considerable y se necesita implementar un

filtro para adecuar la señal resultante, ya sea un paso bajas o similares.

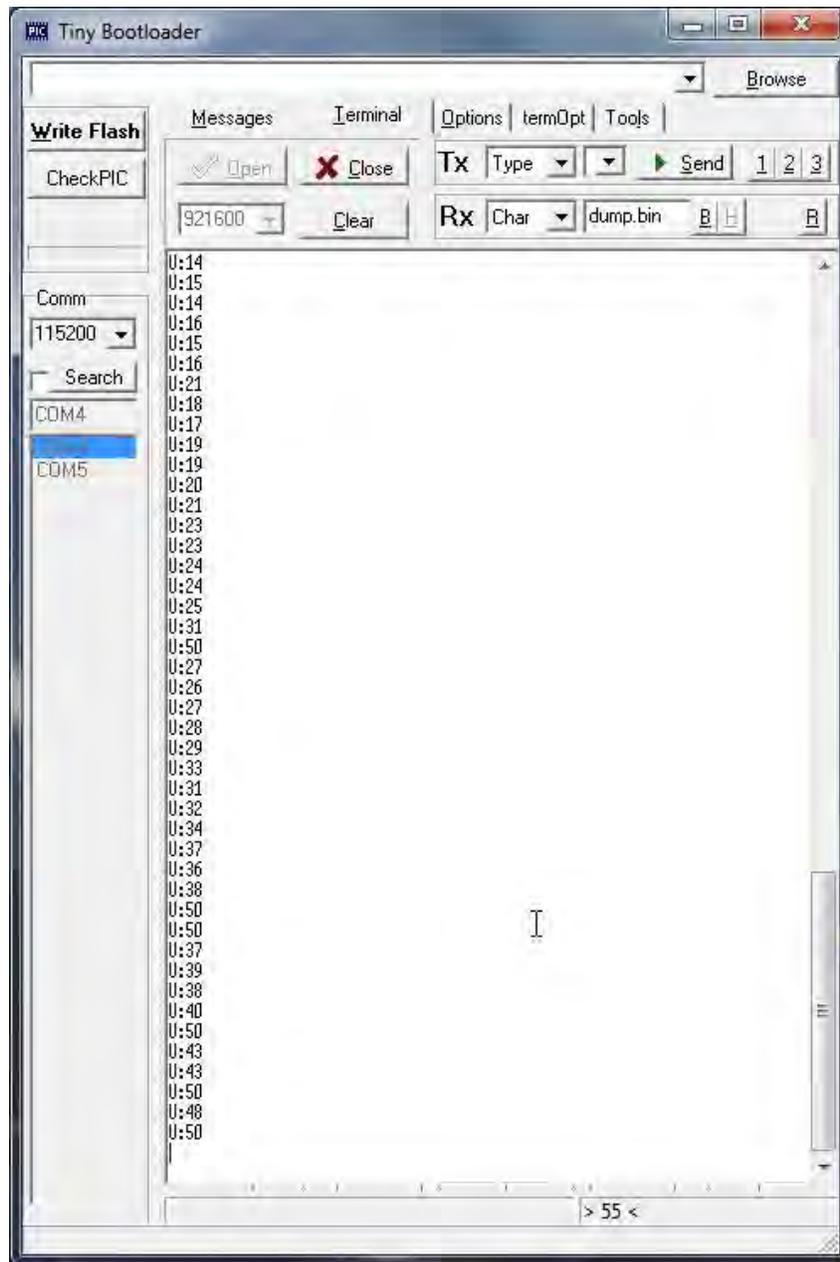


Figura 4.41: A pesar de que la frecuencia de muestreo es notablemente menor a comparación de los otros sensores del robot, el nivel de ruido en la señal puede llegar a ser muy alto cuando sólo tiene enfrente objetos pequeños.

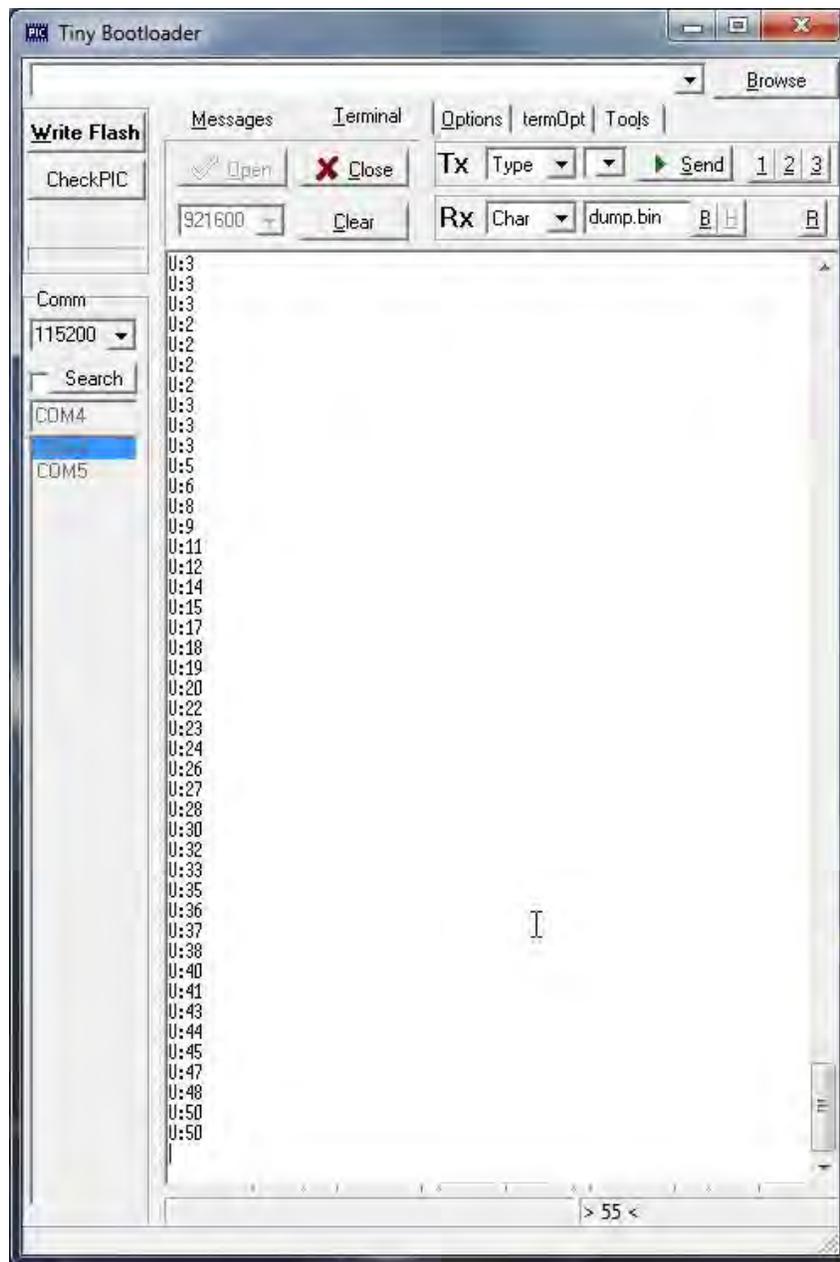


Figura 4.42: Al contrario, objetos grandes darán como resultado una señal suave.

D) LEDs RGB

Los LEDs RGB son controlados por el PCA, al que se le dan los comandos vía I²C desde el **Esclavo**. Todas las caras del robot son independientes para encender el color que sea dentro de los 11 calibrados (véase Tabla 4.3). Después de mandar dos caracteres, número de cara y color deseado, el **Esclavo** hace los ajustes de corriente mandando la información vía I²C al PCA y, finalmente, el PCA enciende los LEDs de las caras del robot. Los valores que puede tomar se encuentran en la Tabla 3.9

En las Figuras 4.43, 4.44, 4.45 y 4.46 se aprecian los colores en las tapas del robot, un color distinto por cara para mostrar la independencia, pero sólo en las tapas para evitar iluminación

excesiva en las fotos.



Figura 4.43: *Primeros 3 colores calibrados disponibles en el robot.*

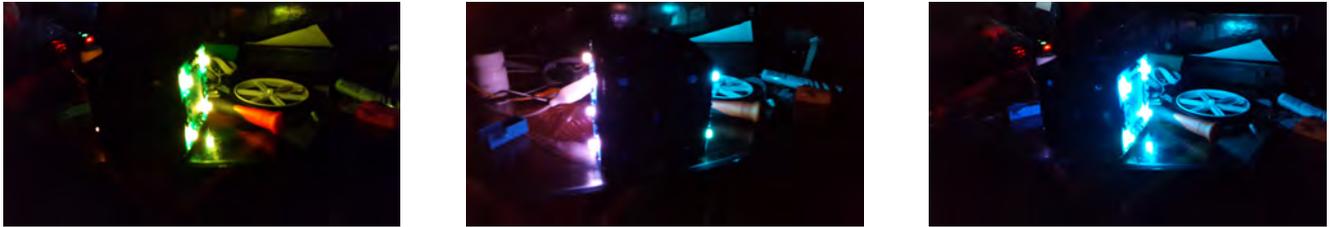


Figura 4.44: *Siguientes 3 colores calibrados disponibles en el robot.*



Figura 4.45: *Últimos colores calibrados disponibles en el robot.*

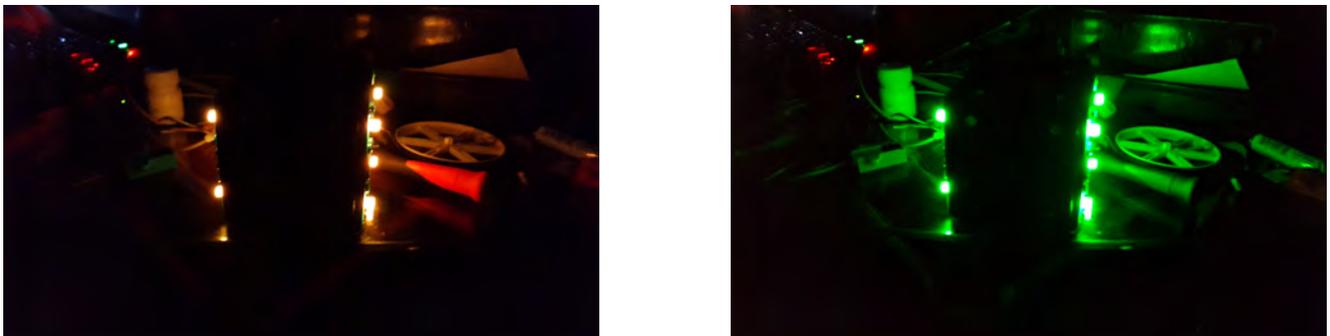


Figura 4.46: *Colores rojo y verde en ambas tapas.*

E) Pruebas a otros componentes

Reguladores de voltaje y MOSFET's

La intención de usar MOSFETs para la alimentación como para los sensores es que exista una carga mínima, menor a los 100 [μ A], a la batería cuando el botón se encuentre en su posición de apagado.

Al no tomarse en cuenta que los reguladores se conectaron directamente en su terminal **GND** a la tierra general del sistema, existe una corriente parásita de 30 [mA] inherente a los reguladores mismos. En el *datasheet* se especifica que esa corriente interna de los reguladores puede llegar hasta los 30 [mA] a máxima carga, pero al medir la corriente de consumo al apagar el robot, se observa nada más el consumo de 30 [mA] totales. Es un fallo en el diseño que se corregiría en trabajos a futuro cambiando el tipo de MOSFET a uno canal “P” y su posición en el circuito general de alimentación.

Por lo mismo, es necesario desconectar las baterías cuando el robot no está en uso; el dejarlas conectadas drenaría las baterías a un punto en el que se dañarían, inflarían y podrían inclusive afectar la placa de Energía.

Comparadores

El diseño de los comparadores permite al robot conocer cuando las baterías se drenan hasta cierto nivel, el diseño seleccionado (véase sección C) [Nivel de carga de la batería](#)) permite determinar el momento en el que el voltaje de las baterías cae de manera individual a 3.0 [V]. De mediciones hechas se verificó que el voltaje bajo de las baterías es detectado entre los 2.9 y 3.1 [V].

Comunicación entre microcontroladores

Aunque las pruebas mostradas en las secciones anteriores muestran una gran cantidad de información en la terminal, el *firmware* contempla que el **Esclavo** enviará información al **Maestro** sólo cuando éste lo solicite y sólo en casos en donde la atención sea necesaria, el **Esclavo** alertaría al **Maestro**, tales como sobrecalentamiento.

Por ende, el **Maestro** no arroja a una computadora que recolecte la información toda la información que sense el **Esclavo** a menos de que se le programe a futuro.

Capítulo 5

Conclusiones

En este capítulo se brindan las impresiones finales del robot, donde se analiza si y de qué manera cumple con los objetivos planteados al principio de la tesis.

El trabajo consistió en el diseño de un robot móvil, interactivo y modular para mostrar la integración de los subsistemas que conforman un sistema mecatrónico. Como se puede constatar, a lo largo de la tesis son abordados todos los subsistemas que mínimamente pueden conformar a un sistema mecatrónico. Tales subsistemas son el mecánico, el electrónico y la programación. Cada subsistema engloba muchos aspectos distintos del robot pero, a su vez, existen casos donde ciertos elementos comparten características que constituyen parte de un subsistema u otro.

La integración total de los subsistemas da como resultado un robot que cumple con los objetivos deseados, unos en mayor grado que otros.

- El robot cumple con el objetivo de ser móvil. Cuenta con un sistema completo para desarrollar la movilidad en superficies planas y con poca inclinación. El robot puede desplazarse cuando se le programa como tal, donde se puede controlar la velocidad a la que tienen que rotar los motores y se pueden implementar algoritmos y filtros para tener la orientación del robot en todo momento. El tema central de esta tesis no es el de implementar algoritmos de navegación en el robot, sino de diseñar el robot. Como tal, al robot se le brindaron durante el diseño todas las herramientas necesarias para poder ser móvil.
- El robot cumple con el objetivo de ser interactivo. La interactividad se da de diversas maneras: tiene la capacidad de interactuar con movimiento, tiene la capacidad de interactuar con diversos patrones de luz, tiene la capacidad de reaccionar ante obstáculos, tiene la capacidad de modificar sus patrones al cambiar modos predeterminados almacenados en funciones de sus botones, el robot puede interactuar con otros robots y con el usuario brindando información y cualquier cosa que se le programe a través de la red wifi; el robot puede interactuar con el usuario en términos de personalización al contar con bloques tipo *LEGO*, donde el usuario puede montar la personalización externa a su gusto e imaginación. Aunque no se pudo brindar al robot la capacidad de interactuar con sonido, no puede recibir ni emitir sonido, el robot cuenta con muchas herramientas y funciones que le brindan una gama muy grande de posibilidades de interactividad, sujeta a cómo el usuario lo programe.
- El robot cumple con el objetivo de ser modular. Internamente el robot se compone de módulos que cumplen en conjunto las tareas que se programan en el robot, estos módulos son los

subsistemas que conforman el sistema mecatrónico en conjunto como unidad individual. De manera externa, los robots son modulares y todo robot conectado en la red puede compartir información y tareas entre diferentes módulos similares en la red. Dependiendo de su programación, mientras que un robot desempeña una tarea, otro robot puede desempeñar una totalmente distinta, donde su coordinación esté dictaminada a un proceso de sincronización entre diferentes módulos de ambos robots.

Gracias a ello el robot puede ser visto como ente individual o como su conjunto de módulos, para programar distintas tareas y donde la información es distribuida en la red mediante el módulo de comunicación.

- El robot cumple con el objetivo de ser modular. Internamente el robot se compone de módulos que cumplen en conjunto las tareas que se programan en el robot, estos módulos son los subsistemas que conforman el sistema mecatrónico en conjunto como unidad individual. De manera externa, los robots son modulares y todo robot conectado en la red puede compartir información y tareas entre diferentes módulos similares en la red. Dependiendo de su programación, mientras que un robot desempeña una tarea, otro robot puede desempeñar una totalmente distinta, donde su coordinación esté dictaminada a un proceso de sincronización entre diferentes módulos de ambos robots.
Gracias a ello el robot puede ser visto como ente individual o como su conjunto de módulos, para programar distintas tareas y donde la información es distribuida en la red mediante el módulo de comunicación.
- El robot es reprogramable. Actualmente es necesario ampliar el *firmware* del robot para poder ampliar la gama de acciones y funciones predeterminadas; sin embargo, es necesario coordinar la ampliación del *firmware* con el desarrollo de la interfaz de usuario que servirá de enlace entre los robots y los usuarios.
A pesar de ello, el robot cuenta con un *firmware* base mínimo que puede desarrollar tareas de calibración y muestra de funciones de todos los módulos con los que cuenta el robot, y cuenta con la memoria suficiente para almacenar un programa.
- El robot es pequeño. El concepto de pequeño es subjetivo y cambia de persona a persona. El robot tiene dimensiones menores a las de *MUNAMBOT* y similares a las de *Linkbot*; en general, la referencia se toma con la capacidad de ser manipulado con una sola mano.
- No se han realizado pruebas para determinar la resistencia a impacto o fatiga en los componentes y partes del robot, sin embargo, durante su diseño mecánico se redujeron los concentradores de esfuerzo y se hicieron adecuaciones estructurales que ayudarían a satisfacer la necesidad de ser resistente. Por ende, no se puede determinar en este momento si el diseño cumple cabalmente con la necesidad o no.
- El robot es flexible porque sus funciones no están bloqueadas por un programa en específico. Dependiendo de la visión del usuario, el robot puede ser programado con diferentes patrones y comportamientos ante diferentes situaciones. No hay una sola manera en que el usuario pueda reprogramar al robot y los módulos que lo integran dan esa flexibilidad.
- El robot es inalámbrico. Cuenta con dos baterías que lo alimentan durante un tiempo no medido hasta el momento pero que es variable dependiendo qué módulos se utilizan. El robot cuenta con comunicación wifi, lo que lo libera de tener que emplear algún cable para comunicarse con el usuario.

Finalmente, la realización de este proyecto, cumple con el objetivo de demostrar que se tiene la capacidad técnica y los conocimientos fundamentales para realizar este tipo de proyectos, donde se realizó diseño para todas las áreas aprendidas durante la carrera.

El diseño del robot no es perfecto. A lo largo del capítulo 4 se fueron detallando los errores encontrados a lo largo de las pruebas. Aunque existen los errores detectados y no se pudieron implementar todos los circuitos que se deseaban implementar originalmente, el robot cumple con los objetivos trazados desde un inicio. En futuras iteraciones, gracias a la experiencia de diseño adquirida se pueden implementar aún más funciones y circuitos, dando así la posibilidad de diferentes modelos de robots similares que puedan interactuar en la misma red.

Capítulo 6

Trabajos a futuro

En futuras iteraciones se tienen que corregir los errores de diseño encontrados a lo largo de las pruebas con los modelos físicos. Tales errores son:

- 1.- Corregir el MOSFET general de alimentación por uno canal-P, de esta manera se evitaría el consumo parasitario de los reguladores.
- 2.- Corregir los espacios para los LEDs cercanos a las baterías.
- 3.- Mejorar la posición de los conectores, así como idear una solución que permita el fácil ensamble y desensamble del robot.
- 4.- Detectar el error y corregir el diseño del puente H para aprovechar todo el voltaje de alimentación. En caso de corregirlo conectando la alimentación al voltaje de las baterías, sería necesario detectar en todo momento qué voltaje máximo se tiene con las baterías en caso de querer hacer un control dado que la alimentación no sería constante.
- 5.- Evitar el calentamiento excesivo de los microcontroladores, sustituyéndolos por modelos de 3.3 [V] (familia dsPIC33F) o quizá implementar ya microcontroladores de 32 bits.
- 6.- Desarrollo de la interfaz de usuario para programación y monitoreo en conjunto con un firmware más actualizado y poderoso para el **dsPIC Maestro** para poder interpretar muchas más acciones y tareas a realizar para todo el módulo de procesamiento.
- 7.- Implementar en el **Maestro** los protocolos de escritura a la memoria.
- 8.- Desarrollo de un intérprete de lenguaje para robustecer la configuración al vuelo del *RN171*, de tal forma que sea configurable al vuelo y el microcontrolador tenga la capacidad de interpretar los mensajes de configuración y error que tenga el dispositivo, así como capacidad de búsqueda automática de redes, etc.

Aparte de la corrección de los errores, se pueden hacer varias adiciones al diseño:

- Adición del circuito recargador de baterías.
- Adición del micrófono y bocina para incorporar la interactividad por sonido.
- Optimizar aún más las placas para poder meter más componentes o hacerlas más compactas.

- Un circuito de “debuggeo” para realizar pruebas a los microcontroladores sin la necesidad de abrir y cerrar el robot para acceder a las terminales de reprogramación.

Finalmente, como propuesta general, se tienen los conceptos y visiones de futuros modelos del robot:

- Un modelo que en vez de sensor ultrasónico cuente con una cámara de video.
- Baterías de mayor capacidad y de similar tamaño; cada año salen al mercado nuevas baterías con mayor capacidad, un nuevo conjunto de baterías aportaría mayor tiempo de autonomía al robot.
- Con las nuevas baterías, probar motores de mayor potencia.
- Experimentar con encoders magnéticos para analizar si no alteran el funcionamiento de otros circuitos, que en dado caso evitarían los problemas de interferencia lumínica detectados con los ópticos.
- Desarrollar algoritmos de filtro para navegación con el robot. En caso de ser necesario agregar más herramientas al diseño del robot para que se pueda navegar con él.
- Investigación con filtros aplicados a la IMU (por ejemplo, un filtro Kalman), con la finalidad de aprovechar al máximo ese sensor.
- Posibilidad de adición de una memoria flash como una micro SD para almacenar ahí distintos programas.

Bibliografía

- [1] *IEEE Standard for Floating-Point Arithmetic*. IEEE Std 754-2008, páginas 1–70, Aug 2008.
- [2] Analog Devices Inc.: *AN-1057 Application Note*, 2010. http://www.analog.com/static/imported-files/application_notes/AN-1057.pdf, consultado el 29 de octubre de 2014.
- [3] China Young Sun LED Technology Co.: *RED/GREEN/BLUE Triple Color LED white diffused lens*. <http://cdn.sparkfun.com/datasheets/Components/LED/YSL-R596AR3G4B5C-C10.pdf>, consultado el 01 de marzo de 2015.
- [4] Chip Quik Inc.: *SMD291SNL10 Solder Paste No-Clean SAC305 in 10cc Syringe 35g T3 Mesh*, 2015. <http://www.chipquik.com/datasheets/SMD291SNL10.pdf>, consultado el 19 de abril de 2015.
- [5] Cmglee: *Nominal dimensions of standard Lego bricks*. https://upload.wikimedia.org/wikipedia/commons/thumb/1/1a/Lego_dimensions.svg/1280px-Lego_dimensions.svg.png, consultado el 24 de abril de 2014.
- [6] Colton, Shane: *The Balance Filter. A Simple Solution for Integrating Accelerometer and Gyroscope Measurements for a Balancing Platform*, June 25th 2007. <http://web.mit.edu/~jinstone/Public/filter.pdf>, consultado el 31 de octubre de 2014.
- [7] Creative Machines Lab: *Molecubes*, 2014. <http://www.creativemachineslab.com/>, consultado el 15 de octubre de 2014.
- [8] Cytron Technologies: *Product User's Manual – HC-SR04 Ultrasonic Sensor*, 2013. https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/edit, consultado el 31 de octubre de 2014.
- [9] Desconocido: *Batería EB615268VU*, 2014. http://brain.pan.e-merchant.com/7/2/12234927/1_12234927.jpg, consultado el 01 de junio de 2014.
- [10] Desconocido: *HC-SR04 Dimension*, 2015. http://img.alibaba.com/img/pb/289/262/368/368262289_217.jpg, consultado el 01 de junio de 2015.
- [11] Desconocido: *Ánodo y cátodo*, 2015. <http://i0.wp.com/67.222.1.89/~etolocka/wp-content/uploads/2015/01/Anodo-y-catodo1.png>, consultado el 01 de junio de 2015.
- [12] Diodes Incorporated: *1N5711W Surface Mount Schottky Barrier Diode*, 2008. <http://diodes.com/datasheets/ds11015.pdf>, consultado el 12 de marzo de 2015.
- [13] Diodes Incorporated: *1N4148W Surface Mount Fast Switching Diode*, 2013. <http://www.diodes.com/datasheets/ds30086.pdf>, consultado el 12 de marzo de 2015.

- [14] Diodes Incorporated: *DMN6040SSD 60V Dual N-Channel Enhancement Mode MOSFET*, 2014. <http://www.diodes.com/datasheets/DMN6040SSD.pdf>, consultado el 03 de agosto de 2015.
- [15] Diodes Incorporated: *S1A/B - S1M/B 1.0A Surface Mount Glass Passivated Rectifier*, 2014. <http://diodes.com/datasheets/ds16003.pdf>, consultado el 07 de enero de 2015.
- [16] Dudek, Gregory y Michael Jenkin: *Computational Principles of Mobile Robotics*. Cambridge University Press, 2da edición, 2000.
- [17] E-Switch Inc.: *TL2201EEYA*, 2008. http://spec_sheets.e-switch.com/specs/P010354.pdf, consultado el 01 de marzo de 2015.
- [18] E-Switch Inc.: *Series TL 2201 Switches Pushbutton Switches - Sub Miniature*, 2015. https://www.e-switch.com/system/asset/product_line/data_sheet/61/TL2201.pdf, consultado el 01 de marzo de 2015.
- [19] E-Switch Inc.: *TL2201-series*, 2015. https://www.e-switch.com/system/asset/product_line/image/61/TL2201-series.jpg, consultado el 01 de marzo de 2015.
- [20] Espinosa Maya, Luis Eduardo: *Fundamentos de mecatrónica*, 2007. ftp://ece.buap.mx/pub/profesor/espinosa/Mecatronica/Fundamentos_Mecatronica/PRIMAVERA%2008/MATERIAL/FUNDAMENTOS%20DE%20MECATRONICA_U2.pdf, consultado el 15 de octubre de 2014.
- [21] Fairchild Semiconductor Corporation: *MM3Z2V4B-MM3Z75VB Zener Diodes*, 2010. <https://www.fairchildsemi.com/datasheets/MM/MM3Z10VB.pdf>, consultado el 01 de junio de 2014.
- [22] Fastenal Company: *Hex Small Pattern Machine Screw Nuts, Stainless Steel*, 2014. https://www.fastenal.com/content/product_specifications/HMSN.SMP.SS.pdf, consultado el 19 de mayo de 2014.
- [23] Glynn, Earl F.: *Color Chart*, 2005. <http://research.stowers-institute.org/efg/R/Color/Chart/ColorChart.pdf>, consultado el 16 de octubre de 2014.
- [24] Harris, Tom: *How Robots Work*, 16 April 2002. <http://science.howstuffworks.com/robot.htm>, consultado el 17 de abril de 2014.
- [25] Hokuriku Electric Industry Co. Ltd: *Tactile Switches 6mm x 6mm Top Push*, 2007. http://site.gravitech.us/Components/BUTT-4/BUTT-4_datasheet.pdf, consultado el 12 de marzo de 2015.
- [26] InvenSense Inc.: *MPU-6000 and MPU-6050 Product Specification*, 2013. <http://www.invensense.com/mems/gyro/documents/PS-MPU-6000A-00v3.4.pdf>, consultado el 31 de octubre de 2014.
- [27] InvenSense Inc.: *MPU-6000 and MPU-6050 Register Map and Descriptions*, 2013. <http://www.invensense.com/mems/gyro/documents/RM-MPU-6000A-00v4.2.pdf>, consultado el 31 de octubre de 2014.

- [28] KEMET Electronics Corporation: *Tantalum Surface Mount Capacitors T491 Series Industrial Grade MnO₂*, 2015. http://www.kemet.com/Lists/ProductCatalog/Attachments/254/KEM_T2005_T491.pdf, consultado el 11 de marzo de 2015.
- [29] Linden, David y Thomas B. Reddy: *Handbook of Batteries*. McGraw-Hill Handbooks. McGraw-Hill, 3ra edición, 2002.
- [30] Microchip Technology Inc.: *dsPIC30F Programmer's Reference Manual*, 2005. <http://ww1.microchip.com/downloads/en/DeviceDoc/70030f.pdf>, consultado el 31 de octubre de 2014.
- [31] Microchip Technology Inc.: *dsPIC30F Family Reference Manual*, 2006. <http://ww1.microchip.com/downloads/en/DeviceDoc/70046E.pdf>, consultado el 31 de octubre de 2014.
- [32] Microchip Technology Inc.: *MPLAB® C30 C Compiler User's Guide*, 2007. http://ww1.microchip.com/downloads/en/devicedoc/c30_users_guide_51284f.pdf, consultado el 31 de octubre de 2014.
- [33] Microchip Technology Inc.: *dsPIC30F3010/3011 Data Sheet*, 2010. <http://ww1.microchip.com/downloads/en/DeviceDoc/70141F.pdf>, consultado el 31 de octubre de 2014.
- [34] Microchip Technology Inc.: *dsPIC30F4011/4012 Data Sheet*, 2010. <http://ww1.microchip.com/downloads/en/DeviceDoc/70135G.pdf>, consultado el 31 de octubre de 2014.
- [35] Microchip Technology Inc.: *MCP6541/1R/1U/2/3/4 Push-Pull Output Sub-Microamp Comparators*, 2011. <http://ww1.microchip.com/downloads/en/DeviceDoc/21696H.pdf>, consultado el 01 de marzo de 2015.
- [36] Microchip Technology Inc.: *24FC1026 1024K I²CTM Serial EEPROM*, 2013. <http://ww1.microchip.com/downloads/en/DeviceDoc/20002270D.pdf>, consultado el 12 de marzo de 2015.
- [37] Microchip Technology Inc.: *Packaging Specification*, 2014. <http://ww1.microchip.com/downloads/en/PackagingSpec/00000049CE.pdf>, consultado el 01 de junio de 2014.
- [38] Microchip Technology Inc.: *RN 171. 2.4 GHz IEEE Std. 802.11 b/g Wireless LAN Module*, 2014. <http://ww1.microchip.com/downloads/en/DeviceDoc/70005171A.pdf>, consultado el 31 de octubre de 2014.
- [39] Microchip Technology Inc.: *WiFly Command Reference Manual*, 2014. <http://ww1.microchip.com/downloads/en/DeviceDoc/50002230B.pdf>, consultado el 31 de octubre de 2014.
- [40] Modular Robotics Incorporated: *Cubelets Robot Construction for Kids*, 2014. <http://www.modrobotics.com/cubelets/>, consultado el 15 de enero de 2015.
- [41] Molex: *1.25 Wire to Board Conn. Crimp Rec. Terminal - Lead Free-*, 1998. http://www.molex.com/pdm_docs/sd/500798000_sd.pdf, consultado el 12 de marzo de 2015.
- [42] Molex: *1.25 Wire to Board Wafer Ass for SMT*, 2004. http://www.molex.com/pdm_docs/sd/532610671_sd.pdf, consultado el 12 de marzo de 2015.

- [43] Molex: *1.25 W-to-B Conn. 51021,53261 Mating Cross Section*, 2006. http://www.molex.com/pdm_docs/sd/510210600_sd.pdf, consultado el 13 de marzo de 2015.
- [44] Molex: *1.25 W/B Conn Wafer Assy R/A*, 2014. http://www.molex.com/pdm_docs/sd/530480810_sd.pdf, consultado el 12 de marzo de 2015.
- [45] Molex: *1.25mm Pitch PicoBlade Crimp Terminal, 26-28 AWG, Reel*, 2015. http://www.molex.com/webdocs/datasheets/pdf/en-us/0500798000_CRIMP_TERMINALS.pdf, consultado el 12 de marzo de 2015.
- [46] Molex: *1.25mm Pitch PicoBlade Header, Right Angle, 8 Circuits*, 2015. http://www.molex.com/webdocs/datasheets/pdf/en-us/0530480810_PCB_HEADERS.pdf, consultado el 12 de marzo de 2015.
- [47] Molex: *1.25mm Pitch PicoBlade Header, Surface Mount, Right Angle, 4 Circuits*, 2015. http://www.molex.com/webdocs/datasheets/pdf/en-us/0532610471_PCB_HEADERS.pdf, consultado el 12 de marzo de 2015.
- [48] Molex: *1.25mm Pitch PicoBlade Header, Surface Mount, Right Angle, 6 Circuits*, 2015. http://www.molex.com/webdocs/datasheets/pdf/en-us/0532610671_PCB_HEADERS.pdf, consultado el 12 de marzo de 2015.
- [49] Molex: *1.25mm Pitch PicoBlade Wire-to-Wire and Wire-to-Board Housing, Female, 4 Circuits*, 2015. http://www.molex.com/webdocs/datasheets/pdf/en-us/0510210400_CRIMP_HOUSINGS.pdf, consultado el 12 de marzo de 2015.
- [50] Molex: *1.25mm Pitch PicoBlade Wire-to-Wire and Wire-to-Board Housing, Female, 6 Circuits*, 2015. http://www.molex.com/webdocs/datasheets/pdf/en-us/0510210600_CRIMP_HOUSINGS.pdf, consultado el 13 de marzo de 2015.
- [51] Molex: *1.25mm Pitch PicoBlade Wire-to-Wire and Wire-to-Board Housing, Female, 8 Circuits*, 2015. http://www.molex.com/webdocs/datasheets/pdf/en-us/0510210800_CRIMP_HOUSINGS.pdf, consultado el 13 de marzo de 2015.
- [52] Nichicon Corporation: *WJ Aluminum Electrolytic Capacitors*, 2014. <http://www.nichicon.co.jp/english/products/pdfs/e-wj.pdf>, consultado el 11 de marzo de 2015.
- [53] Nichicon Corporation: *WX Aluminum Electrolytic Capacitors*, 2014. <http://www.nichicon.co.jp/english/products/pdfs/e-wx.pdf>, consultado el 11 de marzo de 2015.
- [54] Nihon Dempa Kogyo Co. Ltd.: *NX8045GB (for OA / AV)*, 2015. http://www.ndk.com/images/products/catalog/c_NX8045GB-STD-CSF-4_e.pdf, consultado el 02 de enero de 2015.
- [55] NXP Semiconductors: *HEF4013B*, 2011. https://www.nxp.com/documents/data_sheet/HEF4013B.pdf, consultado el 01 de marzo de 2015.
- [56] NXP Semiconductors: *PCA9956A*, 2014. http://cache.nxp.com/documents/data_sheet/PCA9956A.pdf?fsrch=1&sr=1&pageNum=1, consultado el 01 de marzo de 2015.
- [57] Ollero Baturone, Aníbal: *Robótica Manipuladores y robots móviles*. Marcombo, 1ra edición, 2001, ISBN 9788426713131. <http://books.google.com.mx/books?id=TtMfuy6FNCcC>.

- [58] Panasonic Industrial Devices: *Aluminum Electrolytic Capacitors/ S*, 2014. <https://industrial.panasonic.com/cdbs/www-data/pdf/RDE0000/ABA0000C8.pdf>, consultado el 02 de enero de 2015.
- [59] Panasonic Industrial Devices: *Precision Thick Film Chip Resistors*, 2014. <http://industrial.panasonic.com/lecs/www-data/pdf/A0A0000/A0A0000CE2.pdf>, consultado el 11 de marzo de 2015.
- [60] Panasonic Industrial Devices: *Thick Film Chip Resistors*, 2014. <http://industrial.panasonic.com/lecs/www-data/pdf/A0A0000/A0A0000CE1.pdf>, consultado el 11 de marzo de 2015.
- [61] Pololu Corporation: *Diagram with dimensions of the Pololu micro metal gearmotor bracket extended.*, 2014. <http://www.pololu.com/picture/view/0J798>, consultado el 24 de abril de 2014.
- [62] Pololu Corporation: *298:1 Micro Metal Gearmotor with Extended Motor Shaft*, 2015. <https://www.pololu.com/product/2208/specs>, consultado el 12 de marzo de 2015.
- [63] Pololu Corporation: *Black Pololu 90x10mm wheel on a Pololu micro metal gearmotor.*, 2015. <https://www.pololu.com/picture/view/0J2599>, consultado el 01 de junio de 2015.
- [64] Pololu Corporation: *Dimensions of the Pololu micro metal gearmotors with precious metal brushes: low-power, medium-power (MP), and high-power (HP). Units are mm over [inches].*, 2015. <https://www.pololu.com/picture/view/0J6671>, consultado el 01 de junio de 2015.
- [65] Pololu Corporation: <https://www.pololu.com/category/60/micro-metal-gearmotors>, 2015. <https://www.pololu.com/category/60/micro-metal-gearmotors>, consultado el 01 de junio de 2015.
- [66] Pololu Corporation: *Mechanical drawing of Pololu wheel 90x10mm without tire.*, 2015. <https://a.pololu-files.com/picture/0J2573.1200.jpg?2e9678021222e4d12b3eb6d9bc67ad2a>, consultado el 01 de junio de 2015.
- [67] Pololu Corporation: *Micro metal gearmotor with extended motor shaft.*, 2015. <https://www.pololu.com/picture/view/0J2814>, consultado el 01 de junio de 2015.
- [68] Raffle, Hayes y Amanda Parkes: *Topobo construction kit with kinetic memory*, 2014. <http://www.topobo.com/>, consultado el 15 de octubre de 2014.
- [69] Ridden, Paul: *Linkbot modular robot platform grows with your knowledge*, Mayo 27, 2013. <http://www.gizmag.com/linkbot-modular-robots/27677/>, consultado el 15 de octubre de 2014.
- [70] Samsung Electro-Mechanics: *Multilayer Ceramic Capacitors*, 2014. http://www.samsungsem.com/global/front/downloadcms.do?path=%2F%2Fglobal%2F%2Fsupport%2F%2Flibrary%2F%2Fproduct-catalog%2F%2F__icsFiles%2F%2Fafieldfile%2F%2F2014%2F%2F11%2F%2F13&fileName=MLCC.pdf, consultado el 11 de marzo de 2015.
- [71] Samtec: *Through-Hole .025" Sq Post Header*, 2014. http://www.samtec.com/documents/webfiles/pdf/TSW_TH.PDF, consultado el 12 de marzo de 2015.

- [72] Semiconductor Components Industries LLC: *MMSZ4xxxT1G Series, SZMMSZ4xxxT1G Series*, 2013. http://www.onsemi.com/pub_link/Collateral/MMSZ4678T1-D.PDF, consultado el 01 de marzo de 2015.
- [73] STMicroelectronics: *L298*, 2010. <http://www.st.com/content/ccc/resource/technical/document/datasheet/82/cc/3f/39/0a/29/4d/f0/CD00000240.pdf/files/CD00000240.pdf/jcr:content/translations/en.CD00000240.pdf>, consultado el 01 de junio de 2014.
- [74] Texas Instruments Incorporated: *CD40106B Types CMOS Hex Schmitt Triggers*, 2003. <http://www.ti.com/lit/ds/symlink/cd40106b.pdf>, consultado el 02 de abril de 2015.
- [75] Texas Instruments Incorporated: *CD4013B Types CMOS Dual 'D'-Type Flip-Flop*, 2005. <http://www.ti.com/lit/ds/symlink/cd4013b.pdf>, consultado el 12 de abril de 2015.
- [76] Texas Instruments Incorporated: *Circuit Board Layout Techniques*, 2008. <http://www.ti.com/lit/ml/sloa089/sloa089.pdf?keyMatch=chip+antenna&tisearch=Search-EN-TechDocs>, consultado el 17 de febrero de 2015.
- [77] Texas Instruments Incorporated: *LM2940-N/LM2940C 1A Low Dropout Regulator*, 2013. <http://www.ti.com/lit/ds/symlink/lm2940-n.pdf>, consultado el 01 de marzo de 2014.
- [78] Texas Instruments Incorporated: *Analog Engineer's Pocket Reference*, 2014. <https://www.ti.com/seclit/sl/slyw038b/slyw038b.pdf>, consultado el 01 de junio de 2014.
- [79] Texas Instruments Incorporated: *LM3940 1-A Low-Dropout Regulator for 5-V to 3.3-V Conversion*, 2014. <http://www.ti.com/lit/ds/symlink/lm3940.pdf>, consultado el 01 de febrero de 2015.
- [80] Texas Instruments Incorporated: *ULN2803A Darlington Transistor Arrays*, 2014. <http://www.ti.com/lit/ds/symlink/uln2803a.pdf>, consultado el 01 de marzo de 2014.
- [81] Time Inc.: *The Smartest Toy Blocks Ever Made: Cubelets*, 2012. <http://techland.time.com/2012/01/11/the-smartest-toy-blocks-ever-made-cubelets/>, consultado el 15 de enero de 2015.
- [82] Ulrich, Karl T. y Steven D. Eppinger: *Diseño y desarrollo de productos*. McGraw-Hill, 4ta edición, 2009.
- [83] Vishay Intertechnology Inc.: *SMD Radial Axial Aluminum Capacitors*, 2008. http://images.vishay.com/books/VSE-DB0111-0810_SMD.Radial.Axial%20Aluminium%20Capacitors_INTERACTIVE.pdf, consultado el 02 de enero de 2015.
- [84] Zortrax S.A.: *Material Data Sheet: Z-ABS*, 2014. https://zortrax.com/wp-content/uploads/2015/05/Z-ABS_Material_Data_Sheet_eng1.pdf, consultado el 01 de marzo de 2015.

Apéndice A

Planos mecánicos

En este apéndice se muestran los planos mecánicos del robot.

Se componen de:

- [A.1 Planos del cuerpo central](#) (13 páginas).
- [A.2 Planos de la tapa](#) (6 páginas).
- [A.3 Planos del *bracket*](#) (5 páginas).

A.1 Planos del cuerpo central

En esta sección se encuentran los planos del cuerpo central.

A.2 Planos de la tapa

En esta sección se encuentran los planos de la tapa.

A.3 Planos del *bracket*

En esta sección se encuentran los planos del *bracket*.

Apéndice B

Esquemas electrónicos

En este apéndice se muestran los esquemas electrónicos del robot.

Se componen de:

- [B.1 Esquema de la placa del Maestro](#) (3 páginas).
- [B.2 Esquema de la placa del Esclavo](#) (4 páginas).
- [B.3 Esquema de la placa de Energía](#) (4 páginas).

B.1 Esquema de la placa del Maestro

En esta sección se encuentran los esquemas de la placa del **Maestro**.

B.2 Esquema de la placa del Esclavo

En esta sección se encuentran los esquemas de la placa del **Esclavo**.

B.3 Esquema de la placa de Energía

En esta sección se encuentran los esquemas de la placa de **Energía**.

Anexo C

Periféricos del dsPIC

C.1 I/O

Son pines de propósito general de tipo entrada/salida, que le permiten al dsPIC monitorear y controlar otros dispositivos. Casi todos los pines I/O están multiplexados con funciones alternativas o periféricos, donde, dependiendo la configuración particular de ese periférico, desactivará el uso general del pin I/O. Los pines I/O se agrupan en bloques, llamados puertos. En la Figura C.1 se aprecia el diagrama de bloques general de los puertos I/O. En la Figura C.2 se aprecia el diagrama general compartido con otros periféricos.

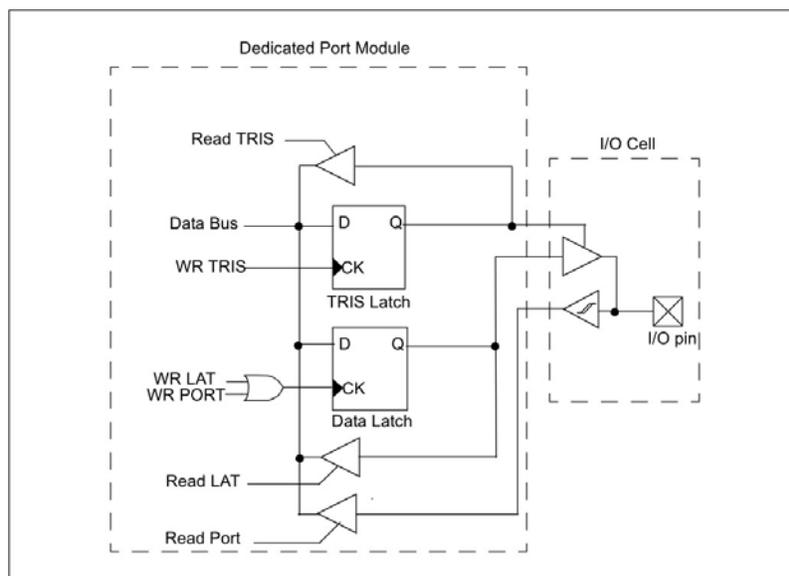


Figura C.1: Diagrama de bloques de los puertos I/O.

Los registros asociados son:

- TRISx (B, C, D, E y F): registros de configuración para entrada o salida de cada uno de los pines del puerto.
- PORTx (B, C, D, E y F): registros para la lectura del valor lógico actual en el pin, sirve tanto para entradas como salidas.

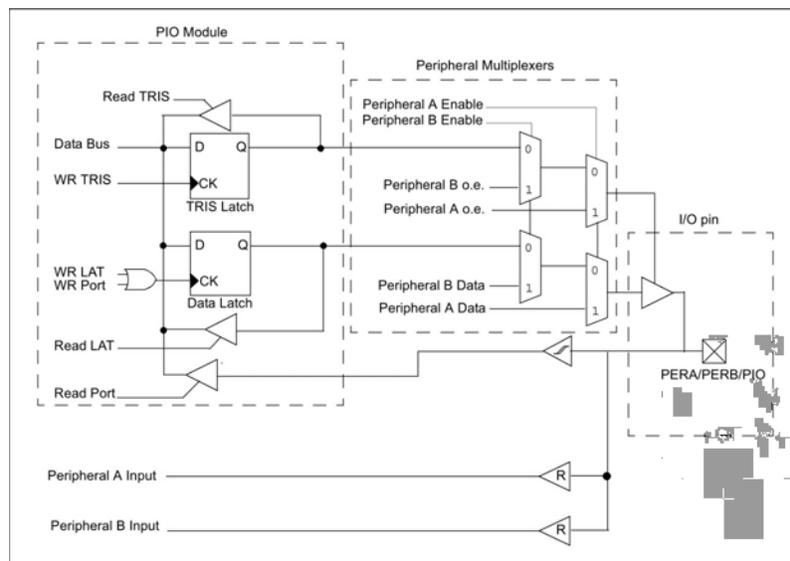


Figura C.2: Diagrama de bloques de los puertos I/O considerando el multiplexeo con los periféricos.

- LATx (B, C, D, E y F): registros para la escritura del valor lógico deseado para el caso de salidas.

C.2 I²C

El módulo I²C es una interfaz serial usada para comunicación con otros periféricos u otros microcontroladores. Estos periféricos pueden ser memorias EEPROM, pantallas, conversores ADC, sensores, etc.

El módulo en particular opera bajo el esquema “Dispositivo Maestro” en un entorno de “Maestro Único”. Se puede operar en otros tipos de entornos, como el “Multi Maestros”, pero no es necesario para este caso en específico. A los elementos que constituyen el entorno se les conoce como “maestros” o “esclavos”.

El módulo I²C contiene una lógica independiente para el maestro y para el esclavo, cada uno generando interrupciones basadas en sus eventos propios. Aunque se configure el módulo para funcionar como maestro, la lógica del esclavo sigue activa, detectando el estado del bus y potencialmente recibiendo mensajes de él mismo, lo cual sirve para depuración.

El módulo I²C contiene su propio generador de baudaje, el cual no consume los recursos de ningún timer en el dsPIC. El módulo soporta las especificaciones de las velocidades estándar (100 [kHz]), rápida (400 [kHz]) y rápida plus (1 [MHz]). El diagrama de bloques del módulo se aprecia en la Figura C.3.

El I²C es una interfaz serial de 2 hilos que emplea un protocolo comprensivo para asegurar una transmisión y recepción fiable. Cuando existe comunicación, un dispositivo “maestro” inicia la transferencia en el bus y genera las señales de reloj para permitir la transferencia, mientras que los demás dispositivos que actúan como “esclavos” responden a esa transferencia. La línea del reloj, **SCL**, es la salida del maestro y entrada de los dispositivos esclavos. La línea de datos, **SDA**, puede ser tanto entrada como salida para el maestro y los esclavos.

Debido a que las líneas **SDA** y **SCL** son bidireccionales, las líneas de comunicación son de tipo open drain (colector abierto), valiéndose de resistores tipo pull-up para asegurar un nivel alto en

- Start Data Transfer (S): es la acción del maestro que determina que el bus se va a ocupar.
- Stop Data Transfer (P): es la acción del maestro que determina que la comunicación se ha terminado.
- Repeated Start (R): similar a un Start, le permite al maestro hacer un cambio en la dirección del bus sin soltar el mando del bus.
- Data Valid (D): posterior a un Start, la línea **SDA** debe permanecer estable durante el periodo alto de la señal de reloj.
- Acknowledge (A) o Not-Acknowledge (N): después de la transferencia de cada byte, el dispositivo receptor debe reconocer (ACK) o no (NACK) la información.
- WAIT/Data Invalid (Q): durante una transmisión, el receptor puede evitar la generación de pulsos manteniendo la línea del reloj en bajo, provocando un periodo de espera en la comunicación; durante este periodo, es válido cambiar la información en la línea **SDA**.
- Bus Idle (I): ambas líneas, **SDA** y **SCL** deben permanecer en alto tras una condición de Stop.

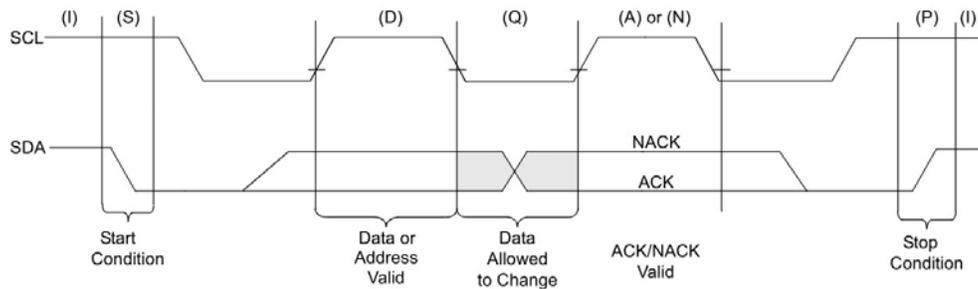


Figura C.4: Estados del protocolo I²C, desde el START hasta el STOP.

Como ejemplo, un mensaje típico entre un maestro y un esclavo, abarcando escritura a un registro del esclavo, involucra los siguientes pasos:

- 1.- Inicio del mensaje; el maestro lanza el Start.
- 2.- Dirección del esclavo; el maestro escribe la dirección del dispositivo a comunicar, en modo escritura.
- 3.- Reconocimiento del esclavo; el esclavo direccionado reconoce la dirección.
- 4.- Registro a escribir; el maestro escribe el registro al que va a escribir información.
- 5.- Reconocimiento del esclavo; el esclavo direccionado reconoce el registro.
- 6.- Dato a escribir; el maestro escribe el dato.
- 7.- Reconocimiento del esclavo; el esclavo direccionado reconoce el dato.
- 8.- Fin del mensaje; el maestro libera el bus después de mandar un Stop.

Un mensaje de lectura de un registro en particular involucra los siguientes pasos:

- 1.- Inicio del mensaje; el maestro lanza el Start.
- 2.- Dirección del esclavo; el maestro escribe la dirección del dispositivo a comunicar, en modo escritura.
- 3.- Reconocimiento del esclavo; el esclavo direccionado reconoce la dirección.
- 4.- Registro a escribir; el maestro escribe el registro al que va a escribir información.
- 5.- Reconocimiento del esclavo; el esclavo direccionado reconoce el registro.
- 6.- Reinicio de la línea; el maestro lanza un Restart.
- 7.- Dirección del esclavo; el maestro escribe la dirección del dispositivo a comunicar, en modo lectura.
- 8.- Reconocimiento del esclavo; el esclavo direccionado reconoce la dirección.
- 9.- Lectura del dato; el maestro recibe el dato desde el esclavo.
- 10.- Reconocimiento del maestro; el maestro lanza un no reconocimiento para indicar que no se leerán más registros.
- 11.- Fin del mensaje; el maestro libera el bus después de mandar un Stop.

Los registros asociados al módulo I²C son:

- I2CRCV: almacena el valor recibido desde otro dispositivo.
- I2CTRN: el valor escrito en este registro es el que se envía por el bus.
- I2CBRG: registro para el divisor del baudaje.
- I2CCON: registro de configuración general del módulo I²C.
- I2CSTAT: registro de banderas indicativas del estado del bus.
- I2CADD: dirección a la que contesta el dsPIC en modo esclavo.

El valor del registro de baudaje se calcula a partir de la siguiente fórmula:

$$I2CBRG = \left(\frac{F_{CY}}{F_{SCL}} - \frac{F_{CY}}{1,111,111} \right) - 1 \quad (C.1)$$

donde:

F_{CY} : frecuencia de ciclo máquina del dsPIC

F_{SCL} : frecuencia del bus

C.3 PWM

El PWM (pulse-width modulation, por sus siglas en inglés), o modulación de ancho de pulso, es una técnica que se puede usar para el control de alimentación de dispositivos eléctricos. El módulo instalado en el dsPIC está diseñado para facilitar el diseño de control de motores.

El módulo PWM tiene las siguientes características:

- Una base de tiempo dedicada ($TCY/2$) para la resolución del PWM.
- La polaridad de salida se configura a través de los bits de configuración del dsPIC.
- Múltiples tipos de salida:
 - modo alineado al borde,
 - modo alineado al centro,
 - modo alineado al centro con doble actualización y,
 - modo de evento simple.
- La actualización de los ciclos puede ser instantánea o sincronizada al periodo del PWM.
- Todo pin asociado al PWM se puede encender de manera individual.

El dsPIC cuenta con un módulo PWM de 6 salidas - 3 generadores. Las salidas van por pares a los generadores, por lo que su funcionamiento se puede programar de manera individual, complementaria o apagado. La base de tiempo para todos los generadores de PWM es compartida, lo que significa que, al seleccionar un periodo para el PWM, será el mismo para los 3 generadores. Cada generador se puede configurar de manera independiente para su modo de salida y para el periodo de trabajo de la señal, donde la salida de su par se configura en los registros de configuración asociados al PWM. En la Figura C.5 se aprecia el diagrama de bloques del módulo PWM.

Los registros principales asociados son:

- PTCON: registro de control de la base de tiempo del PWM.
- PTMR: registro de base de tiempo del PWM.
- PTPER: registro del periodo de la base de tiempo del PWM.
- PWMCON1: registro de configuración del PWM.
- PWMCON2: registro de configuración del PWM.
- PDC1: ciclo de trabajo del generador 1.
- PDC2: ciclo de trabajo del generador 2.
- PDC3: ciclo de trabajo del generador 3.

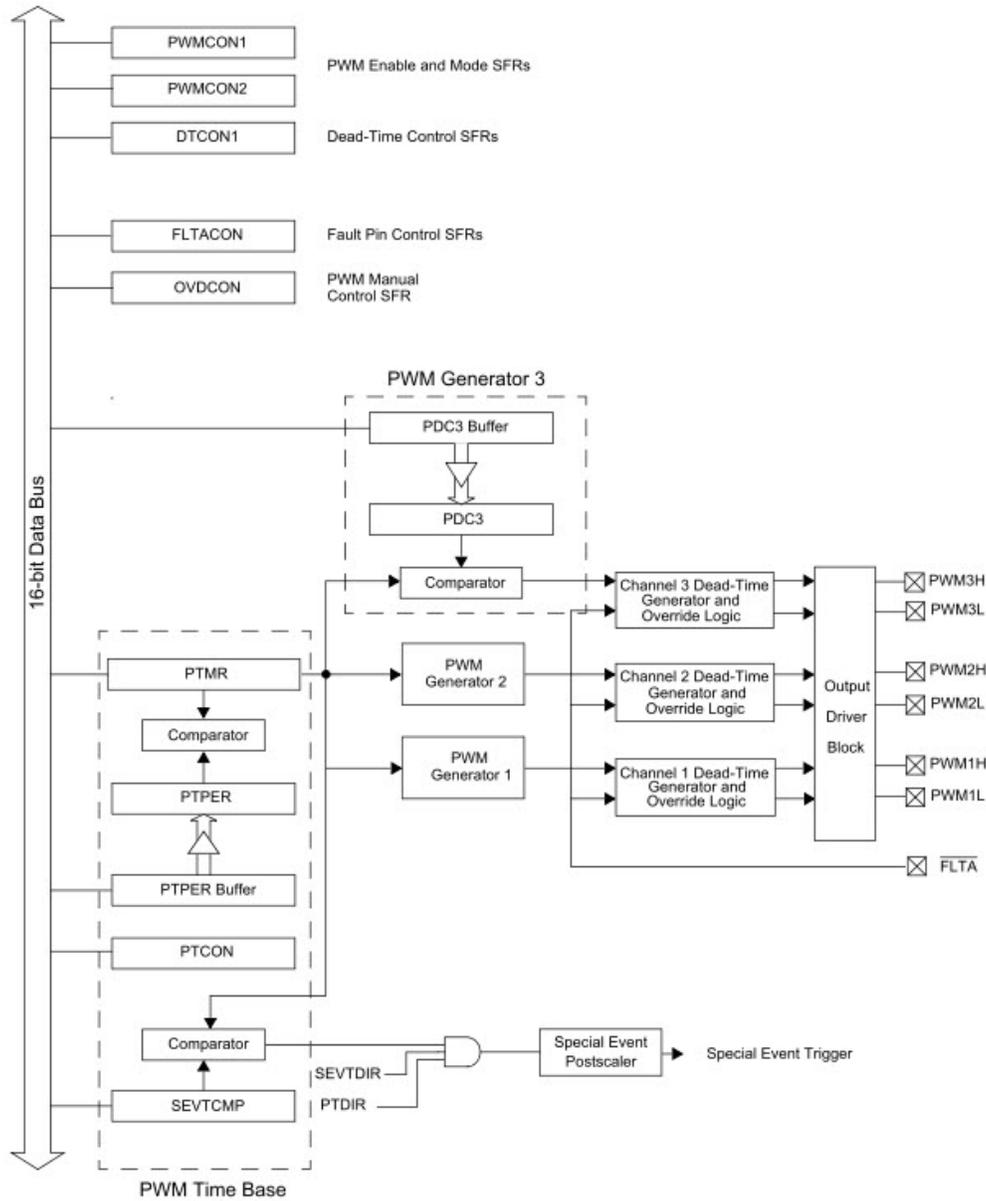


Figura C.5: Diagrama de bloques del PWM

La base de tiempo del PWM es un temporizador independiente de todo Timer, de 15 bits de resolución, cuya fórmula se calcula de la siguiente manera para el modo libre (el registro se reinicializa al llegar al valor delimitado y reinicia el conteo):

$$PTPER = \frac{F_{CY}}{F_{PWM} \cdot (\text{Preescalador } PTMR)} - 1 \quad (C.2)$$

donde:

F_{CY} : frecuencia de ciclo máquina del dsPIC

F_{PWM} : frecuencia del PWM

Preescalador PTMR: preescalador del temporizador (1:1, 1:4, 1:16 y 1:64)

Cabe mencionar que, entre mayor sea la frecuencia del PWM, menor será la resolución disponible de discretización de la señal; de igual forma, incrementar el valor del preescalador reducirá

la resolución. Si el valor de los registros **PDCx** es mayor al de **PTPER**, la señal permanecerá totalmente encendida.

C.4 QEI

Los encoders de cuadratura, también conocidos como encoders incrementales o encoders ópticos, se usan para detección de posición y velocidad en sistemas rotatorios. Un encoder incremental típico incluye una rueda perforada fija a la flecha del motor y un módulo emisor/detector sensa los agujeros en la rueda. Típicamente se tienen de dos a tres salidas; **Fase A**, **Fase B** y **Referencia**. Al ser decodificadas, proporcionan información del movimiento de la flecha, incluyendo distancia y dirección.

Las fases A (**QEA**) y B (**QEB**) tienen una relación única. Siempre una fase se adelantará a otra, mediante esta relación se puede determinar si la flecha gira en un sentido o en otro. La señal referencia (**INDX**) permite establecer una relación absoluta de la posición de la flecha, al marcar que el rotor ha pasado por un punto en específico. En la Figura C.6 se aprecia el diagrama de las señales.

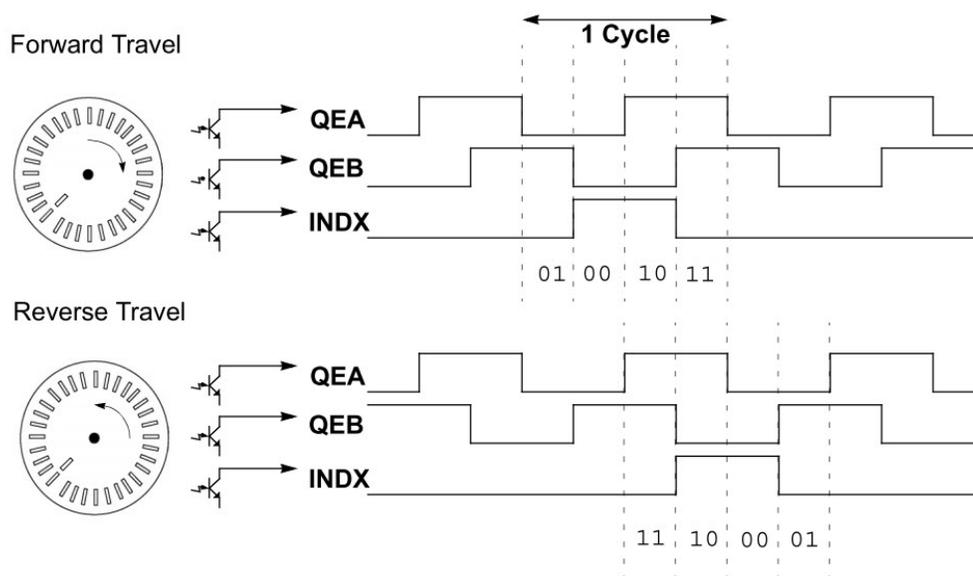


Figura C.6: Diagrama de fases de las señales de cuadratura.

Un decodificador de cuadratura captura los pulsos de las fases de las señales, y de la referencia en caso de que exista, y convierte esta información en una cuenta numérica de los pulsos de la posición. Generalmente, la cuenta se incrementa cuando la flecha rota en un sentido y decrementa cuando rota en el sentido inverso.

El módulo QEI (Quadrature Encoder Interface, por sus siglas en inglés), o interfaz de encóder de cuadratura, provee una interfaz para encoders incrementales. El QEI consiste en un decodificador lógico de cuadratura para interpretar las fases **QEA** y **QEB** y un contador ascendente/descendente para acumular la cuenta. Su diagrama de bloques se aprecia en la Figura C.7.

Las características principales del QEI son:

- Tres pines de entrada para las dos fases y la referencia.

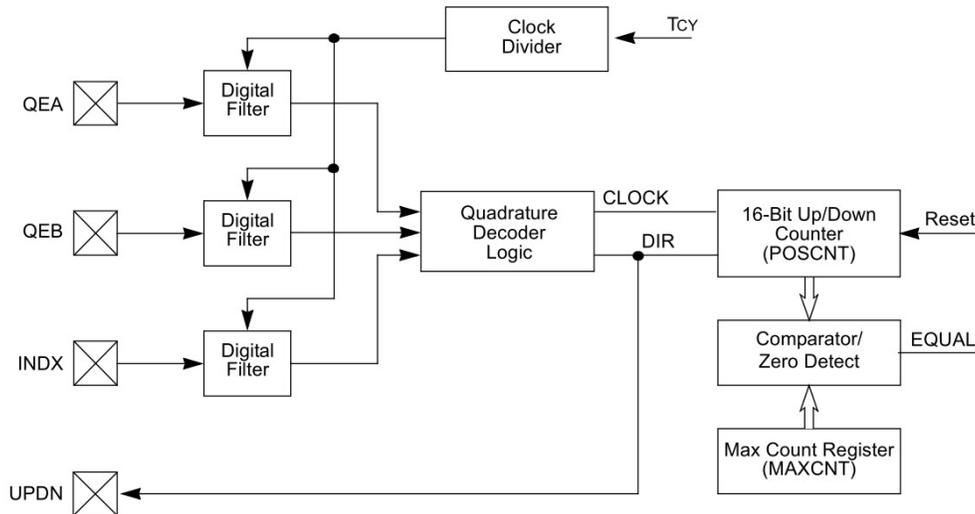


Figura C.7: Diagrama de bloques del **QEI**.

- Filtros digitales programables contra ruido a las entradas.
- El decodificador de cuadratura provee la cuenta de pulsos y la dirección de la cuenta.
- Contador ascendente/descendente de 16 bits.
- Resolución $\times 2$ y $\times 4$.
- Interrupciones generadas por eventos del QEI o el contador.

Los registros asociados son:

- QEICON: registro de configuración e indicador de banderas de la operación del QEI.
- DLFTCON: registro de control para los filtros digitales.
- POSCNT: registro signado de almacenamiento de los pulsos leídos.
- MAXCNT: registro de cuenta máxima para reiniciar la cuenta del módulo y desencadenar una interrupción.

El módulo basa su sensado a partir de una base de tiempo independiente de $T_{CY}/3$, por lo que la frecuencia de las señales no debe exceder ese valor. Por ejemplo, si la frecuencia de trabajo del dsPIC es de 30 [MHz], la señal no debe exceder una frecuencia de 10 [MHz].

Para su uso, se debe configurar para sensar las fases como contador, reiniciado por evento de cuenta máxima o por referencia. La resolución $\times 2$ leerá los flancos de subida y bajada de la fase **QEA**, a su vez, la resolución $\times 4$ leerá los flancos de subida y bajada de ambas fases, para una resolución mayor. El módulo al hacer las lecturas almacena el valor en un registro signado, es decir, si el rotor gira en un sentido positivo, el contador incrementará su valor y viceversa, esto permite que, junto con el bit de dirección, se pueda determinar en un lapso de tiempo el sentido de giro del rotor.

C.5 Timers

Un timer, o temporizador, es un periférico que permite llevar una cuenta de tiempo basada en la frecuencia de oscilación del microcontrolador o un reloj u oscilador externo. La señal de entrada se puede preescalar y, en algunos casos, posescalarla. Se emplean principalmente para medir tiempos o controlar procesos en un tiempo determinado.

El dsPIC cuenta con 5 timers de 16 bits, donde 4 de ellos se pueden combinar para obtener 2 timers de 32 bits. Se designan **Timer 1**, **Timer 2**, ..., **Timer 5**.

Cada timer tiene asociados sus registros:

- TMRx: registro contador de 16 bits.
- PRx: registro asociado al periodo del timer.
- TxCON: registro de configuración del timer.

Cuando se combinan 2 timers para formar 1 de 32 bits, la cuenta se almacena en los registros combinados **TMR3:TMR2** y **TMR5:TMR4**.

Los timers son prácticamente iguales, con las excepciones de que el **Timer 1** puede leer pulsos desde una fuente externa y el resto de los timers se puede combinar para conseguir timers de 32 bits. En la Figura C.8 se aprecia el diagrama de bloques del **Timer 1**, en la Figura C.9 se aprecia el diagrama de bloques de los timers **Timer 2** y **Timer 4** y en la Figura C.10 se aprecia el diagrama de bloques de los timers **Timer 3** y **Timer 5**.

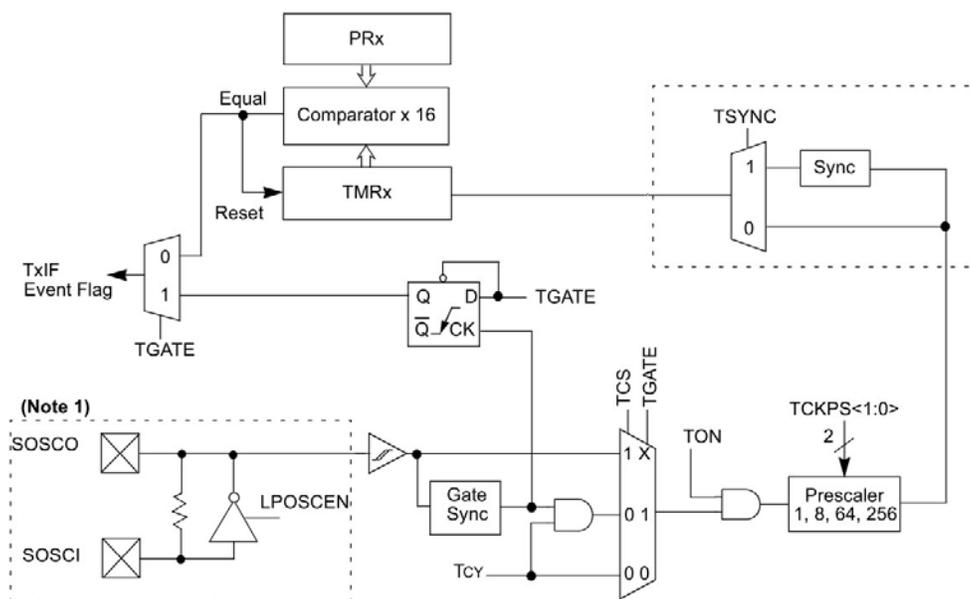


Figura C.8: Diagrama de bloques de timer tipo A.

Los registros **PRx** se emplean para determinar el momento en el que se reinicia el conteo del timer y lanzan una interrupción. Para determinar ese valor se emplea la siguiente fórmula:

$$PRx = \frac{t_d \cdot F_{CY}}{\text{Preescala}} \quad (\text{C.3})$$

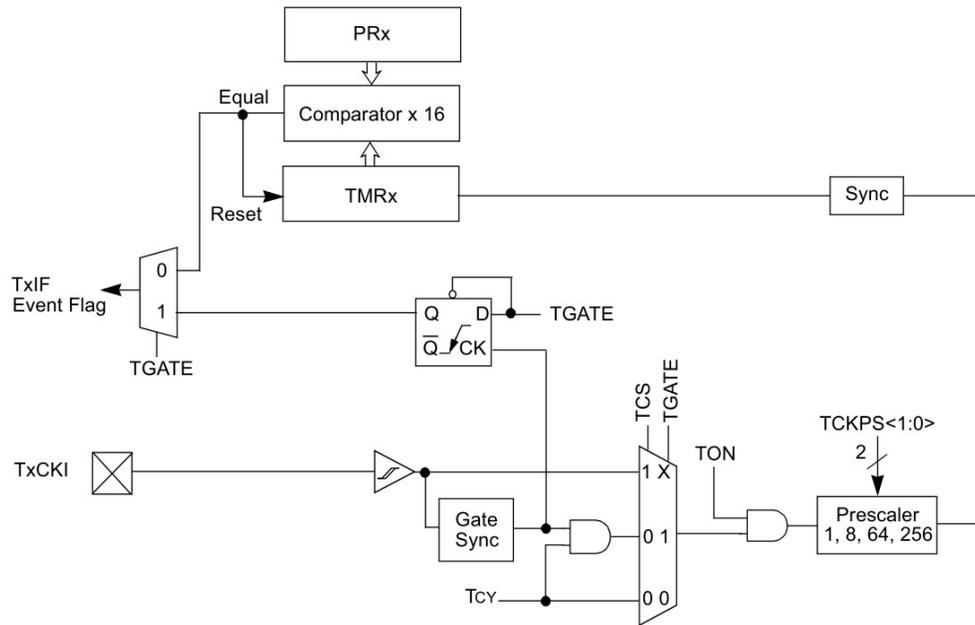


Figura C.9: Diagrama de bloques de timer tipo B.

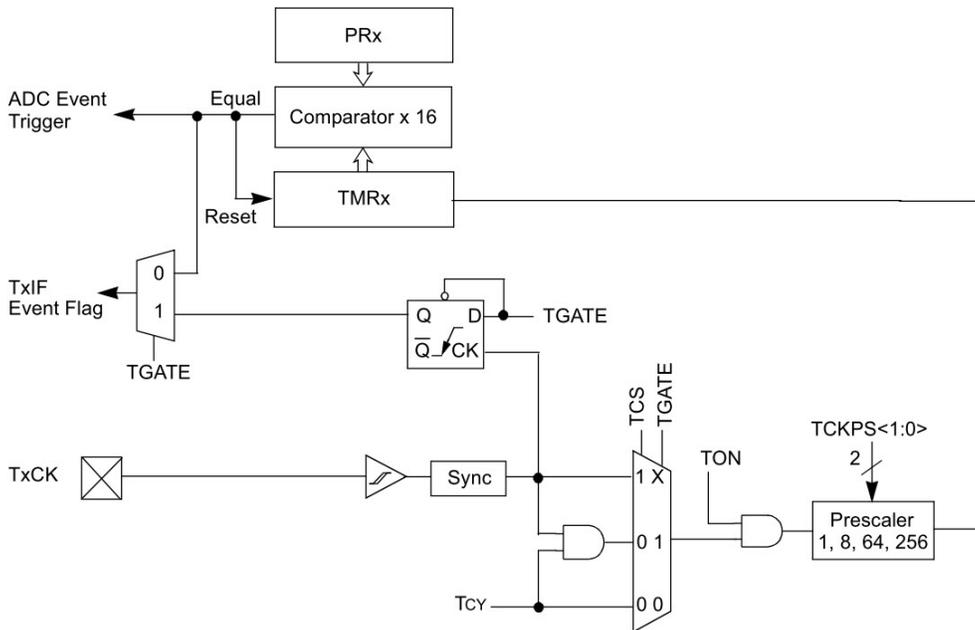


Figura C.10: Diagrama de bloques de timer tipo C.

donde:

PRx : registro asociado al periodo de tiempo del timer

t_d : tiempo deseado

F_{CY} : frecuencia de ciclo máquina del dsPIC

$Preescala$: preescalador del temporizador (1:1, 1:4, 1:16 y 1:64)

de tal forma que los tiempos máximos para los timers, tanto para las versiones de 16 y de 32 bits, se aprecian en la Tabla C.1, calibrados todos a $F_{CY} = 30$ [MHz].

Tabla C.1: *Tiempos máximos de los timers.*

Preescala	Pulso	Desbordamiento [ms]	Desbordamiento [horas]
1	33.3 [ns]	2.18	0.04
8	267 [ns]	17.47	0.32
64	2.13 [μ s]	139.8	2.55
256	8.53 [μ s]	559	10.18
		Timer de 16 bits	Timer de 32 bits

C.6 UART

El módulo UART (Universal Asynchronous Receiver Transmitter, por sus siglas en inglés) es un sistema asíncrono full-duplex que puede comunicarse con dispositivos periféricos, mediante las interfaces RS-232 y RS-485. El dsPIC contiene 2 de estos módulos, independientes entre sí.

Las características principales del módulo UART son:

- Transmisiones full-duplex de 8 ó 9 bits a través de los pines **UxTX** y **UxRX**.
- Para información de 8 bits existen las opciones de paridad par, non o no paridad.
- Uno o dos bits de alto.
- Generador independiente de baudaje integrado con un preescalador de 16 bits.
- El baudaje se puede configurar desde los 29 bps (bits por segundo) a los 1.875 Mbps.
- Buffer de 4 bytes tipo cola tanto para la recepción como la transmisión.
- Interrupciones de transmisión y recepción.

A grosso modo, el módulo se compone de 3 componentes principales: el generador de baudaje, el receptor y el transmisor, representados en la Figura C.11.

Los registros asociados son:

- UxMODE: registro de configuración del UART.
- UxSTA: registro de indicadores del UART.
- UxTXREG: registro de transmisión del UART (se le puede escribir un valor hasta 4 veces).
- UxRXREG: registro de recepción del UART (puede recibir hasta 4 valores).
- UxBRG: registro del valor del preescalador para el baudaje.

El valor del registro **UxBRG** se calcula a partir de la siguiente fórmula:

$$UxBRG = \frac{F_{CY}}{16 \cdot Baud\ Rate} - 1 \quad (C.4)$$

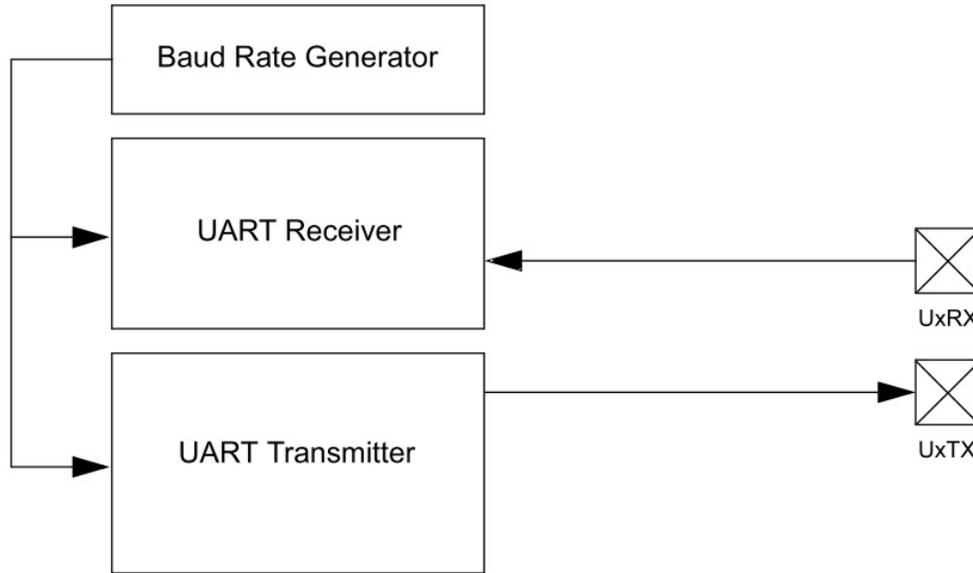


Figura C.11: Diagrama de bloques general del UART.

Tabla C.2: Baudajes típicos.

Baudaje [kbps]	UxBRG (decimal)	% Error
9.6	194	+0.2
115.2	15	+1.7
921.6	1	+1.7
1875	0	0

donde típicamente, el error en el baudaje no debe exceder más allá del $\pm 5\%$. En la Tabla C.2 se muestran diferentes valores para el registro **UxBRG** para baudajes usuales y el máximo baudaje, calibrados a $F_{CY} = 30$ [MHz].

La configuración típica más simple para el módulo es emplear la comunicación sin paridad, de 8 bits y con sólo 1 bit de parada. El módulo tiene la flexibilidad de poder encolar hasta 4 bytes tanto para la recepción como para la transmisión, con varias combinaciones para las interrupciones. El módulo 1 tiene la capacidad de emplear pines alternos, de tal forma que los pines regulares sirvan para otro módulo. El diagrama completo del transmisor se aprecia en la Figura C.12, mientras que el del receptor se aprecia en la Figura C.13.

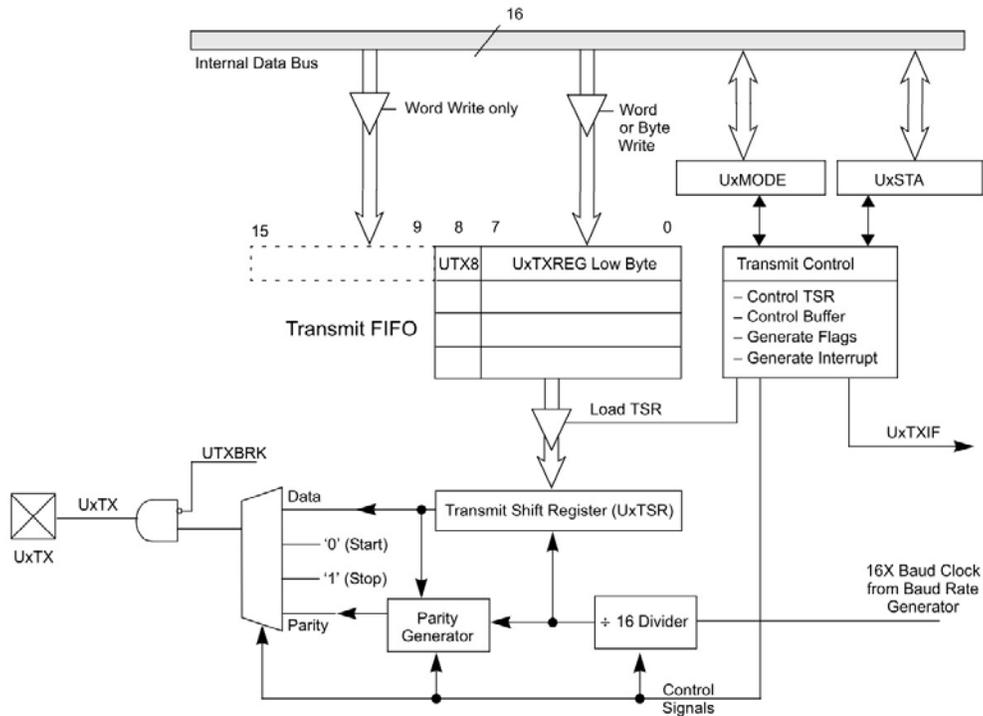


Figura C.12: Diagrama de bloques del transmisor UART.

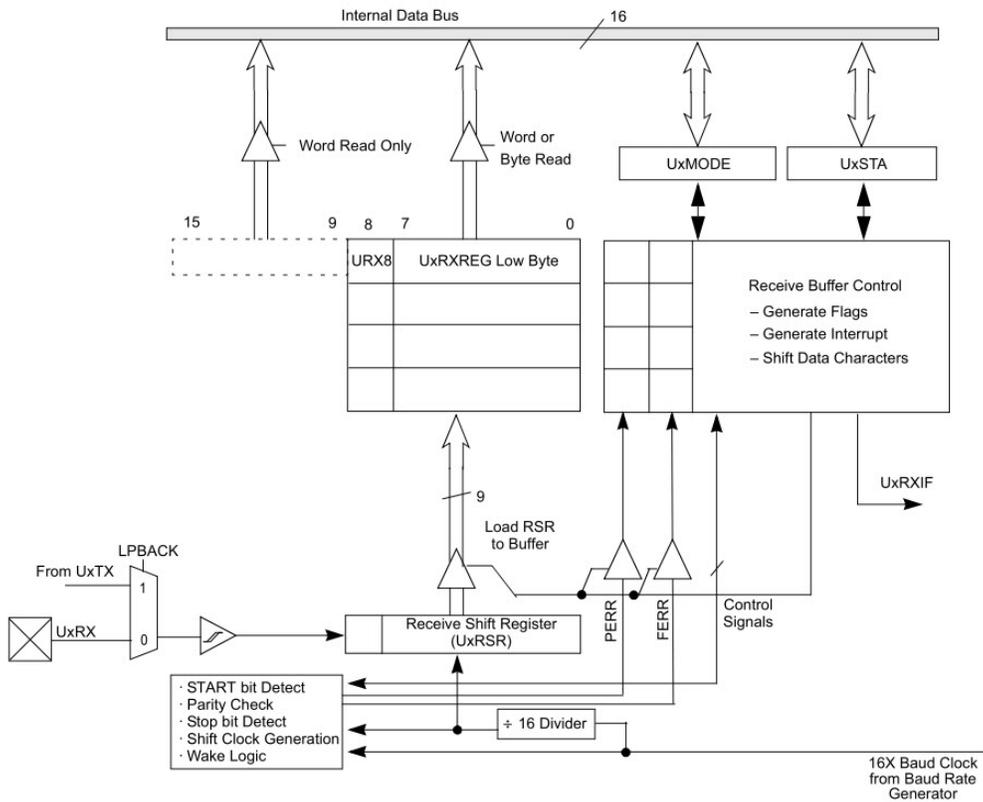


Figura C.13: Diagrama de bloques del receptor UART.