

FACULTAD DE INGENIERÍA - UNIVERSIDAD DE LA REPÚBLICA.

INFORME DE PROYECTO DE GRADO

Robótica educativa controlada por Entorno

Guillermo Amorín Casella

Tutor
Jorge Visca

17 de abril de 2020

Resumen

El presente trabajo tiene como objetivo la creación de un prototipo de robot para ser utilizado en el proyecto "Programando robots, jugando con el entorno". Este proyecto busca evaluar la efectividad del uso de robots como herramienta educativa en niños de 4 a 7 años aproximadamente.

En ese contexto, se busca que el prototipo permita evaluar diferentes tecnologías y que los integrantes del equipo de "Programando robots, jugando con el entorno" puedan utilizarlo para diseñar las distintas actividades que realizarán con los niños durante la etapa de trabajo de campo.

Al trabajar con niños de edades tan pequeñas, se busca que puedan programar el robot mediante la configuración de los elementos de su entorno. Por este motivo, el prototipo debe reaccionar a los diferentes elementos del entorno como forma de determinar su comportamiento.

A partir de lo obtenido en el estudio del estado del arte, la base del esquema de sensado del prototipo se realiza con sensores de distancia y se presenta un estudio comparativo de los distintos sensores disponibles para la construcción.

Una vez construido el prototipo, se realizaron pruebas de comportamiento frente a distintos escenarios y en base a 2 instancias de presentación a expertos, se determinaron mejoras a realizar junto con líneas de trabajo a seguir para continuar con la construcción del robot.

Índice general

Índice general	2
Lista de figuras	5
Índice de figuras	5
Lista de tablas	7
Índice de tablas	7
1 Introducción	9
1.1. Contexto	9
1.2. Robótica Educativa	10
1.3. Programando robots jugando con el entorno	10
1.4. Objetivos del proyecto.	11
2 Estado del arte y trabajos Relacionados	13
2.1. Pautas para la revisión	13
2.2. Conclusiones de la revisión	14
3 Diseño de Hardware	17
3.1. Esquema de actuación	18
3.2. Esquema de sensado	21
3.3. Plataforma de cómputo	28
4 Diseño de Software	33
4.1. Firmware Particle	33
4.2. Firmware del prototipo	37
5 Pruebas de Sensores	47
5.1. Descripción	47
5.2. Análisis de resultados	50
6 Pruebas de Prototipo	61
6.1. Descripción	61
6.2. Análisis de resultados	64
7 Presentación a expertos	65
7.1. Descripción	65
7.2. Análisis de resultados	67
8 Conclusiones y trabajo a futuro	69

ÍNDICE GENERAL

3

Bibliografía

71

Índice de figuras

3.1. Prototipo desarrollado.	17
3.2. Power banks utilizados para alimentar los motores y le microcontrolador.	18
3.3. Trayectoria de desplazamiento de un robot del punto A hacia un punto B. Izquierda: Robot omnidireccional. Derecha: Robot diferencial	18
3.4. a) Rueda omnidireccional compatibles con el kit NXT Lego Mindstorm. b) Vista de perfil. c) Vista superior. Imágenes tomadas de [41].	19
3.5. a) Plataforma. b) Agarre de los motores a la plataforma. c) Piezas ensambladas al robot. Las imágenes a y b fueron tomadas de 3.5 y la imagen c es de elaboración propia.	20
3.6. Módulo MAX7219 con una matrix de 8x8 leds monocromáticos. [47]	20
3.7. a) Esquema de formación del hexágono en la matriz led. Imagen de elaboración propia. b) Imagen del robot, con el comportamiento de encendido y apagado de los lados del hexágono implementado.	21
3.8. En amarillo se muestra una vista de arriba del contorno de un robot, y en rojo la ubicación de los sensores.	22
3.9. Principio de funcionamiento de los sensores ultrasónicos. [15]	24
3.10. Sensor ultrasónico HC-SR04. [15]	24
3.11. Sensor ultrasónico LV-MaxSonar-Ez MB1000. [16]	25
3.12. a) Sensor Sharp-gp2D120X. b) Sensor Sharp-2Y0A02. c) Sensor Sharp-2Y0A21.[43, 44, 45]	26
3.13. Principio de funcionamiento del los sensores láser. Mide con precisión el tiempo que tardan los pulsos emitidos de luz láser en alcanzar un objeto y ser detectados nuevamente. [21]	27
3.14. a) Sensor VL53L0X, con sus pines de entrada y salida. b) Medidas del VL53L0X en pulgadas y comparadas con una moneda de 1 dolar americano. Imágenes tomadas de [46]	27
4.1. Máquina de estados que interpreta mensajes UDP, dirigidos a MCC.	35
4.2. Diagrama de la arquitectura del software construido.	37
4.3. a) Campo de potencial generado por el objeto Goal. b) Campo de potencial generado por el objeto Obstacle. c) Combinación de los campos de potencial generados por ambos objetos y actuando sobre todo el entorno. d) Respuesta del robot frente a la fuerza que ejercen los campos en ti, hasta alcanzar el objeto Goal.[6]	39
4.4. a) Esquema del robot construido. Derecha: Diagrama de cuerpo libre mostrando como actua la fuerza F_i ejercida por el objeto O_i	40
4.5. Construcción de las constantes del módulo action en función de los leds que deseamos prender de la matrix.	41
4.6. Construcción de la matrix de configuración para indicarle al microchip MAX7219 los leds que debe encender. Matrix lado 1 y Matrix lado 2, son las mostradas en la figura 4.5	42

4.7. Diagrama de cinemática de un robot omnidireccional de 3 ruedas. Los vectores unitarios D_i determinan la dirección de avance de la <i>rueda_i</i> [8]	43
5.1. a) Entorno construido para la realización de experimentos. b) Marcas colocadas en el entorno. c) Esquema del entorno construido, donde se marcan las medidas alineadas y las marcas de corrimiento.	47
5.3. Alineación del robot con la marca 0.	48
5.2. Objetos utilizados para las pruebas de sensores. a) Prisma - 5cm x 20cm x 20 cm. b) Cilindro - 15 cm de alto y 2.5 cm de radio	48
5.4. Colocación de objetos en la marcas desalineadas.	49
5.5. Momento en el que se tomaron medidas con el prisma alineado con el sensor. En la figura a, se observa la medición a 1 m, y en la b a 15 cm.	50
5.6. Datos en crudo del sensor HC-SR04 utilizando el prisma.	51
5.7. Datos en crudo del sensor HC-SR04 utilizando el cilindro.	52
5.8. Datos tomados con el sensor HC-SR04 utilizando un filtro pasabajo y el objeto prisma.	53
5.9. Datos tomados con el sensor HC-SR04 utilizando un filtro pasabajo y el objeto cilindro.	53
5.10. Datos tomados con el sensor HC-SR04 utilizando un filtro de ventana con tamaño 3 y el objeto prisma.	54
5.11. Datos tomados con el sensor HC-SR04 utilizando un filtro de ventana con tamaño 3 y el objeto cilindro.	55
5.12. Datos tomados con el sensor HC-SR04 utilizando un filtro de ventana con tamaño 10 y el objeto prisma.	55
5.13. Datos tomados con el sensor HC-SR04 utilizando un filtro de ventana con tamaño 10 y el objeto cilindro.	56
5.14. Datos tomados con el sensor Sharp-2Y0A21 utilizando un filtro de ventana con tamaño 10 y el objeto prisma.	56
5.15. Datos tomados con el sensor Sharp-2Y0A02 utilizando un filtro de ventana con tamaño 10 y el objeto prisma.	57
5.16. Datos tomados con el sensor Sharp-2Y0A02 utilizando un filtro de ventana con tamaño 10 y el objeto cilindro.	58
5.17. Datos tomados con el sensor VL53L0X utilizando un filtro de ventana con tamaño 10 y el objeto prisma.	59
5.18. Datos tomados con el sensor VL53L0X utilizando un filtro de ventana con tamaño 10 y el objeto cilindro.	60
6.1. Escenario de prueba con un objeto. a) Estado inicial, donde el robot se coloca a 23 cm de el prisma. b) Estado final del escenario ejecutado.	61
6.2. Escenario de prueba con 2 objetos. a) Estado inicial, donde el robot se coloca a 23 cm de el prisma. b) Estado final del escenario ejecutado.	62
6.3. Escenario de prueba con 4 objetos.	62
6.4. Escenario de prueba con 3 objetos distintos.	63
6.5. Escenario de prueba con 3 objetos distintos.	63
7.1. Prototipo 0, con solo 3 sensores y la protoboard.	66

Índice de tablas

2.1. Resumen de las características de los robots investigados. *- El tamaño se interpreta como Alto x Largo x Ancho.	16
3.1. Especificaciones de las ruedas omnidireccionales compatibles con el kit NXT Lego Mindstorm.[10]	20
3.2. Tabla 2 - Comparación de los motores SpringRC SM-S4303R y Dynamixel AX-12.[11] [12]	20
3.3. Interfaces utilizadas para la transmisión de información por parte de algunos sensores. Tabla tomada de [25]	22
3.4. Tabla tomada de [14]	25
3.5. Especificaciones del LV-MaxSonar-Ez MB1000. Tabla tomada de [16]	25
3.6. Tabla construida a partir de los datos de [17][18][19]	26
3.7. Características del sensor VL53L0X	27
3.8. Tabla comparativa de las principales características para los microcontroladores y SBC considerados en el proyecto.	30
4.1. Identificación de los distintos módulos.	34

Capítulo 1

Introducción

El presente capítulo brinda una introducción al proyecto “Robótica educativa controlada por Entorno”, donde se expone el contexto en que se originó el proyecto y se presenta el proyecto dentro del cual se enmarca el trabajo realizado en el presente informe.

1.1. Contexto

Hace ya algunas décadas, el mundo que nos rodea se ha venido transformando paulatinamente volviéndose cada vez más computarizado, digitalizado e informatizado. Al día de hoy, los avances tecnológicos en el área de la computación, la electrónica y las tecnologías de la comunicación han tenido un enorme impacto en la vida de las personas, ya sea mejorando su calidad de vida, o evidenciando un cambio donde su uso es ética y moralmente cuestionable, por ejemplo en el área armamentística. Fácilmente comprobable son estas afirmaciones al ver día a día el uso de los teléfonos celulares, las computadoras personales, las tarjetas STM y la facturación electrónica, entre otros.

Como parte de esas transformaciones, innumerables iniciativas han surgido para llevar adelante la inclusión de las nuevas tecnologías en los curriculums educativos de los distintos países del mundo. Varios son los organismos internacionales como la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO) y la Comisión Económica para América Latina y el Caribe en las Naciones Unidas (Cepal), que están llevando adelante políticas para promover estos cambios [1].

En nuestro país esto se refleja en la puesta en marcha del Plan Ceibal desde el 2007 con el objetivo de apoyar a través de la tecnología las políticas educativas uruguayas. Desde su implementación, cada niño que ingresa al sistema educativo público en todo el país accede a una computadora o tablet para su uso personal con conexión a Internet gratuita desde el centro educativo. Además, provee un conjunto de programas, recursos educativos y capacitación docente que pretende transformar la manera de enseñar y aprender [9]. Otra institución que aporta en ese sentido es el INET, que aborda los aspectos pedagógicos y didácticos así como el involucramiento y protagonismo de los docentes.

La importancia de este tipo de políticas debido a los profundos cambios culturales que puede introducir en la sociedad, las obligan a estar en permanente discusión y análisis para poder implementarlas de la mejor manera, en base a información cuantitativa que respalde sus resultados y avances.

Como parte de la motivación y en un contexto donde es necesario avanzar en una discusión lo más abierta, clara, y objetiva posible. Para ello, entiende necesario desarrollar herramientas

que faciliten la obtención de datos así como la evaluación de las políticas llevadas adelante y amplíen el espectro de usuarios que acceden a las mismas, ya sea para profundizarlas o evaluando la pertinencia de las mismas.

1.2. Robótica Educativa

Esta coyuntura hace que la Robótica educativa juegue un rol fundamental, pues ha sido presentada como una herramienta revolucionaria capaz de promover el desarrollo de una forma del pensamiento denominado computacional. Esta forma al día de hoy no tiene una definición única pues es un concepto que aún se encuentra en construcción, pero los autores que lo mencionan lo relacionan al pensamiento lógico matemático, la abstracción, la secuenciación y el separar problemas grandes en otros más pequeños para facilitar su resolución.

Producto de esta imprecisión de su definición tampoco son claras las consecuencias que tiene su introducción al sistema educativo. Por este motivo se han iniciado innumerables líneas de investigación tanto en el campo de la psicología cognitiva, como de la pedagogía y la educación en general. Dichas investigaciones buscan obtener evidencia empírica significativa que realmente respalde que la inclusión de herramientas y técnicas que desarrollen este tipo de pensamiento sean un avance y no un retroceso para el sistema educativo. Si bien existen trabajos que sugieren la veracidad de estos potenciales beneficios, así como la utilidad de la robótica educativa en otras áreas más allá de las relacionadas con las ciencias y la tecnología [2], de no tener evidencia cuantitativa proveniente de distintas investigaciones para apoyar su impacto en el formación los estudiantes, las actividades de robótica pueden ser solo una "moda" [3].

Hace más de 40 años que se utiliza la robótica en ámbitos educativos [6] como universidades y centros de investigación. Esto se hace poniendo a los robots, su relacionamiento con el entorno y su funcionamiento como objetos de estudio. Sin embargo con la robótica educativa se pretende que el educando adquiera conocimiento de otras áreas que no tienen que ser estrictamente la robótica.

Según [5], la robótica educativa es una disciplina que busca involucrar al estudiante para generar procesos de aprendizaje donde se tenga como objetivo profundizar conocimientos de ciertas áreas, con la motivación de no necesariamente enseñar a los estudiantes a convertirse en expertos en robótica, sino con el objetivo de desarrollar capacidades que se consideran esenciales en la actualidad como lo son la autonomía, la iniciativa, la responsabilidad, la creatividad, el trabajo en equipo, el autoestima y el interés por la investigación. Un aspecto importante de esta definición es que no hace referencia explícita a el diseño y construcción de robots, y por lo tanto involucra actividades donde se utiliza un robot previamente construido.

1.3. Programando robots jugando con el entorno

En el contexto previamente descrito, el Centro Interdisciplinario en Cognición para la Enseñanza y el Aprendizaje (CICEA), con el objetivo de comprender los procesos implicados en la adquisición de información en la edad temprana, a mediados del 2017 comienza desarrollar un proyecto enmarcado una línea de investigación en Desarrollo Cognitivo, el cual incluye y fomenta el trabajo presentado dentro del informe.

Dicho proyecto se denomina Programando robots jugando con el entorno (PRJE de aquí en más) y busca desarrollar una plataforma robótica que le permita a niños preescolares y escolares en edades tempranas comenzar a desarrollar conceptos relacionados al pensamiento computacional como la descomposición de un problema complejo en un conjunto de problemas con

menor complejidad, reconocer problemas similares a los cuales se les puede aplicar una solución ya utilizada (reconocimiento de patrones), eliminación de información irrelevante para resolver un problema (abstracción) y algoritmia (desarrollar una solución a un problema paso a paso o elaborar reglas para resolver un problema dado) [4].

La principal característica que posee esta plataforma robótica que se pretende desarrollar es que defina su comportamiento según la disposición física de los elementos que se encuentran a su alrededor, presentando comportamientos tales como: seguir o esquivar obstáculos, realizar determinadas trayectorias, o buscar elementos evadiendo obstáculos. Es decir, que los niños no programan directamente la plataforma como se hace con otros kits, sino que para programarla deberán modificar su entorno y explorar su comportamiento.

Se pretende que la plataforma a desarrollar sea sencilla y posea las siguientes características: bajo costo de construcción, robusta, segura, de un tamaño acorde para el trabajo con niños y capacidades sensoriales suficientes para generar actividades educativas relevantes y motivadoras. Para esto se pretenden estudiar y evaluar distintas tecnologías, poniendo énfasis en el visión artificial, emisión infrarroja y de radiofrecuencia. Dicha evaluación estará fuertemente basada en la robustez, y sencillez de la posible solución.

En este proyecto para evaluar el posible impacto de la plataforma desarrollada se trabajará con un diseño cuasiexperimental de Discontinuidad en la Regresión (RD) [7]. El RD es un diseño Pre-Test/Post-Test de dos grupos que supone que la misma evaluación se aplica antes y después de la intervención, se pretende medir aspectos relacionados con la motivación, el trabajo colaborativo, funciones ejecutivas, y precursores del pensamiento computacional. Antes del comienzo del tratamiento se toma una medida del nivel de inteligencia para la constitución de los grupos experimental y control a niños de mismo nivel educativo, dicha medida permite el emparejamiento de los grupos. El proyecto se llevará a cabo durante la totalidad del año educativo, se estima una intervención de una vez por semana durante dos horas. Las actividades se realizarán en un lugar amplio y despejado, con un área disponible de aproximadamente 9 metros cuadrados, superficie de piso lisa y buena iluminación.

1.4. Objetivos del proyecto.

El presente proyecto tiene 2 objetivos principales:

1. La creación de un prototipo que permita evaluar tecnologías y brindar una base a partir de la cual se comience a construir el robot que se utilizará en PRJE.
2. El prototipo creado debe ser lo suficientemente robusto para que los integrantes del equipo de PRJE puedan utilizarlo para diseñar y probar las distintas actividades de trabajo y evaluación que realizarán con los niños durante la etapa de trabajo de campo del proyecto.

Como objetivos específicos se planteó:

- Evaluar y comparar distintas tecnologías de sensores de distancia en función del costo, la precisión y el tamaño de los mismos.
- Evaluar tecnologías que permitan obtener feedback de la actuación del robot a los usuarios.
- Construir un diseño arquitectónico que permita agregar fácilmente sensores y actuadores.
- El prototipo debe ser lo más económico y estándar posible.

Organización del documento

A continuación se listan y describen los capítulos contenidos en el presente informe:

- Capítulo 2 - Estado del arte y trabajos Relacionados: Se presenta un análisis de los datos obtenidos en la investigación de trabajos relacionados.
- Capítulo 3 - Aspectos de diseño: Se detallan las decisiones de diseño tomadas a lo largo del proyecto. Primeramente se presenta una descripción general del prototipo construido, y luego el diseño de hardware y software.
- Capítulo 4 - Pruebas de sensores: Se describen las pruebas de sensores y se analizan sus resultados.
- Capítulo 5 - Pruebas de sensores: Se describen las pruebas de prototipo y se analizan sus resultados.
- Capítulo 5 - Presentación a expertos: Se describen las instancias de presentación a expertos, y se presentan los resultados obtenidos de estas instancias.
- Capítulo 6 - Conclusiones y trabajo a futuro: Se presentan las conclusiones obtenidas y se sugieren líneas de trabajo para continuar el desarrollo.
- Capítulo 7 - Referencias.

Capítulo 2

Estado del arte y trabajos Relacionados

El presente capítulo presenta los aspectos que orientaron la revisión de trabajos relacionados, así como las características que se buscó evaluar de cada robot. A partir de la investigación realizada, se elaboran conclusiones que orientaron el trabajo a lo largo del proyecto. En el documento *Estado del Arte*, es posible encontrar los robots investigados, así como un análisis de la información obtenida para cada uno.

2.1. Pautas para la revisión

Para fijar las pautas se tuvo en cuenta principalmente la edad de los niños con los que se pretende trabajar en el proyecto “Programando Robots Jugando con el Entorno”. Dicha edad se encuentran en un rango de 5 a 7 años aproximadamente lo que implica que la forma de programar el robot debe ser sencilla y mediante alguna de las siguientes maneras:

1. La indicación explícita de instrucciones de alto nivel, por ejemplo avanzar, detenerse, doblar a la izquierda o doblar a la derecha.
2. Mediante algún lenguaje basado en bloques, o algún lenguaje con una interfaz gráfica “drag and drop” que permita el encadenamiento de instrucciones.
3. Mediante algún tipo de interacción con el robot, donde este reaccione a estímulos provenientes del entorno, ya sea mediante la disposición de elementos en el mismo, o debido a la interacción con algún tipo de accesorio por parte del usuario. Por ejemplo, los accesorios del kit Dino que simulan ser piedras, y permiten que el robot realice determinado comportamientos.

Otra cuestión importante para definir pautas, fue el tipo de interacción que desean tener los maestros y psicólogos con los niños, ya que se desea desarrollar actividades utilizando al robot como una herramienta, sin la necesidad de pasar por etapas de ensamblaje del mismo. Esto sumado a la peligrosidad que implica tener piezas pequeñas, o con bordes que puedan hacer que los niños se lastimen hacen que la búsqueda se halla centrado principalmente en robots ya ensamblados.

Para finalizar, con el objetivo de identificar aspectos de los distintos kits robóticos que se han utilizado en investigaciones científicas, se revisaron 2 revisiones sistemáticas que tenían como objetivo identificar los posibles beneficios de la utilización de la robótica educativa en diferentes áreas del conocimiento.

En [27] los autores destacan que el 80% de las temáticas abordadas como objeto de estudio están relacionadas directamente con matemática y física, concluyendo que existe una falta de investigación con evaluación cuantitativa del aprendizaje ampliando el uso de la robótica a otras áreas del conocimiento (es decir, áreas no relacionadas con las ciencias exactas). El 90% de los trabajos se realizaron utilizando robots lego y por último la edad de los estudiantes con los que se trabajó se dividía en 2 franjas, de 6 a 10 años y de 13 a 16, pero las muestras eran en su gran mayoría muestras chicas (de menos de 100 estudiantes). Un dato no menor a destacar es que solo 2 estudios presentaban muestras elegidas aleatoriamente y había un grupo de control con el cual validar los resultados.

La otra revisión sistemática realizada por Lai Poh Emily Toh, et al. en el 2015 [28] concluye nuevamente que existe una falta de información cuantitativa como lo señalaban en [27]. En esta revisión se seleccionaron 27 trabajos de las bases de datos IEEE Explore, Academic Search Premiere, ERIC, Science Direct y Springer Link. Y si bien los aspectos analizados en esta instancia eran diferentes a los de Barreto y Benitti (tipo de estudio realizado, influencia del robot en el desarrollo y comportamiento del niño, percepción del robot desde el punto de vista de los padres, educadores y niños y reacción de los niños a la apariencia y diseño del robot), el hecho de que la gran mayoría de los trabajos analizados fueron no experimentales lleva a los autores a reafirmar la conclusión.

A su vez en base los trabajos de Woods [26] y de Sullivan and Bers [29] los autores concluyen que el diseño de un robot puede hacer la diferencia entre como un niño percibe y se relaciona con un robot.

De estas 2 revisiones se desprende no descartar el 100% de los kits que utilizan piezas encastrables ya que en la gran mayoría de las investigaciones realizadas hasta el momento los utilizan.

Una vez determinadas las características que deben tener los robots que serán incluidos en la revisión, se determinaron algunos aspectos importantes a tener en cuenta al momento de realizar el estudio de cada robot. Dentro de ellos se encuentran:

- Modo de programación.
- Tamaño.
- Forma de alimentación.
- Edades objetivos.
- Costo en el mercado (se tomó como referencia la página de amazon y en caso de no encontrarse disponible en dicha página se tomó como referencia la página del fabricante).
- Esquema de sensado.
- Esquema de actuadores.

Para finalizar, es importante tener en cuenta que no fue posible acceder a datos técnicos de todos los kits, ya que por lo general, algunos fabricantes mantienen oculta esta información.

2.2. Conclusiones de la revisión

Establecidas las pautas para realizar la revisión, se analizaron 12 kits robóticos de diferentes características como muestra la tabla 2.1. De ella podemos desprender las siguientes observaciones:

- Salvo el robot Chip, ningún otro robot presenta un esquema motor omnidireccional.
- 7 de los 12 robots utilizan algún tipo de tecnología (IR o ultrasonido) para medir distancias y detectar objetos a su alrededor.
- 9 de los 12 robots, utilizan leds como indicadores hacia el usuario.
- Salvo el robot Pleo, el tamaño de los robots no supera los 20 cm en cualquiera de las 3 dimensiones.

Teniendo en cuenta que a pesar de introducir ciertas dificultades mecánicas, un robot omnidireccional, posee la capacidad de desplazarse en cualquier dirección sin importar la posición actual del robot y que esto provoca que los mecanismos de control de movimiento sean más sencillos en comparación con un robot diferencial. El hecho de encontrar un solo robot omnidireccional motivo a que el prototipo a construir tuviera esta característica.

Se decidió también, mantener la característica que poseen la mayoría de los robots de comunicarse con el usuario utilizando los leds como interfaz de comunicación. Además, se pretende construir un robot que no supere un tamaño de 20 cm x 20 cm x 20 cm, ya que salvo el robot Pleo el resto no lo superan.

Por último, en relación a la forma del robot, se pretende que el robot se aleje de las formas convencionales a la que los niños por lo general acostumbran a observar en televisión (humanoides, o similares a un auto o dron) ya que según [28] podría llegar a aumentar la curiosidad de quienes lo utilicen y deseen explorar sus comportamientos.

Robot	Modo de Programación	Tamaño* (cm)	Alimentación	Costo (US\$)	Edad (años)	Esquema Motor	Sensores	Actuadores
Chip	Aplicación compatible con android e IOs, y/o modificando su entorno	30x30x30	4 pilas AAA	165	+ de 8	Omnidireccional de 4 ruedas	Sensores de contacto, botón, micrófono, receptor IR, giroscopio y acelerómetro	Emisor IR, parlante, leds, bluetooth BLE, motores
Chippies	Control remoto	15x15x15	4 pilas AAA	40	5	Diferencial de 4 ruedas	Sensores de contacto, botón, receptor IR, giroscopio y acelerómetro	Emisor IR , parlante, leds y motores
MIP	Aplicación compatible con Android e IOs, y/o programando en un framework basado en bloques	12x20x8	4 pilas AAA	60	+ de 8	Diferencial de 2 ruedas	Sensores de contacto, botón, micrófono, receptor IR, giroscopio y acelerómetro	Emisor IR, parlante, leds, bluetooth BLE y motores
Dash and Dot	Mediante una aplicación compatible con Android e IOs	17x18x16	Batería de litio.	150	+ de 6	Diferencial de 2 ruedas con una rueda loca de apoyo	Sensores de distancia, encoders, receptores IR, micrófono y potenciómetro	Motores, parlante, emisor IR y bluetooth BLE
Pet	No es programable	Depende del diseño ensamblado.	2 pilas AA	50	+ de 6	Depende del diseño ensamblado	Botón	Motores
Ollobot	R+Scrach (basada en Scrach) y R+ Task 2.0 (Basada en C++)	Depende del diseño ensamblado.	3 pilas AA	100	+ de 8	Depende del diseño ensamblado	Sensor de grises micrófono	Motores y bluetooth BLE
Pleo	Mediante una SDK (software development kit)	17,8x50,8x12,7	Batería de litio	469	+ de 12	Cuadripedo	Camara, receptor IR, sensor de temperatura, lector RFID, 2 micrófons y 7 sensores de contacto	14 motores, 2 parlantes y emisor IR
Ozobot Bit	Mediante un código de colores y es posible utilizar un framework compatible con Android donde se programa en base a bloques	3x3x3	Batería de lipo	60	+ de 6	Diferencial de 2 ruedas	Sensores de gris y sensor de color.	Motores y leds
Ozobot Evo	Mediante un código de colores y es posible utilizar un framework compatible con Android donde se programa en base a bloques	4x4x4	Batería de lipo	90	+ de 6	Diferencial de 2 ruedas	Sensores de gris, sensor de color y 2 sensores de distancia	Motores, leds y parlantes
Ciber Robot	Mediante los botones de su PCB o mediante una aplicación compatible con Android e iOS que permite usar el dispositivo como control remoto.	10x20x20	4 pilas AA	80	+ de 8	Diferencial de 2	7 botones	2 motores, un parlante y un led
Cody & Rocky	Mediante una interfaz gráfica basada en scrach que utiliza bloques.	10x10x10	Batería de litio	100	+ de 6	Diferencial con 2 orugas	Receptor IR, potenciómetro, giroscopio, led RGB, sensor de voz, sensor de luz, botones, sensor de color	Receptor IR, motores, parlante
Doc Robot	Mediante sus botones, indicándole direcciones.	10x20x20	3 pilas AA	35	4 a 7	Diferencial de 2 ruedas	Botones	Motores
Bee Bot	Mediante sus botones, indicándole direcciones.	12x10x7,5	Batería de litio	80	+ de 1	Diferencial de 2 ruedas	Botones	Motores

Tabla 2.1: Resumen de las características de los robots investigados. *- El tamaño se interpreta como Alto x Largo x Ancho.

Capítulo 3

Diseño de Hardware

La presente sección detalla los aspectos relacionados al diseño del prototipo. Ver Figura 3.1. Para eso se comienza con una descripción general del producto obtenido. Luego se detallan las decisiones relacionadas al hardware, y finalmente se realiza una descripción del software desarrollado.

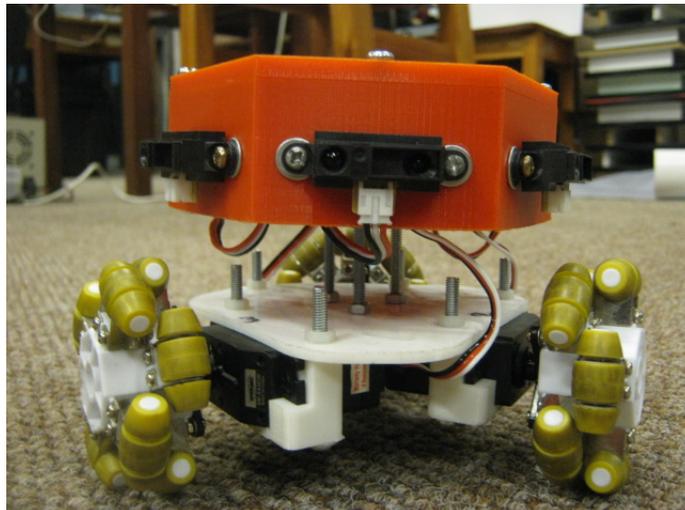


Figura 3.1: Prototipo desarrollado.

Resumidamente el producto obtenido puede describirse como un robot omnidireccional de 3 ruedas con un anillo de sensores de distancia, y una matriz led de 8x8 leds en la parte superior controlados por un microcontrolador.

El movimiento del robot se realiza gracias a la utilización de motores servo que se encuentran acoplados al robot gracias a una plataforma plástica impresa en la impresora 3d del laboratorio de robótica y tomada de [42].

Para la implementación del anillo de sensores se utilizaron 6 sensores sharp infrarrojos dispuestos de forma centrada en cada lado de una estructura hexagonal plástica y según las lecturas de los sensores que componen el anillo, se despliegan distintas configuraciones de una matriz led ubicada en la parte superior del robot.

La alimentación del robot se realiza a través de power banks, como los que se muestran en la figura 3.2.



Figura 3.2: Power banks utilizados para alimentar los motores y le microcontrolador.

3.1. Esquema de actuación

Entendiendo un actuador como un elemento que le permite al robot transformar un valor almacenado en un elemento de computo a una magnitud física/química. Construimos el esquema de actuación con la composición de un subesquema compuesto por una matriz de 8x8 leds, y un subesquema compuesto por los motores. Este ultimo subesquema se integra por la plataforma inferior, los acoples para los motores, los motores y las ruedas; permitiendole al robot desplazarse y girar en el espacio. El subesquema del módulo led, tiene como objetivo comunicar a los usuarios información relativa al sensado del robot. En este caso, la matriz de leds indicará que sensores se encuentran activados en el robot.

Subesquema motor

Tanto su construcción y la elección de sus partes se realizaron en base a la decisión de que el prototipo a realizar debía ser omnidireccional. Particularmente se decidió que el prototipo sea omnidireccional de 3 ruedas.

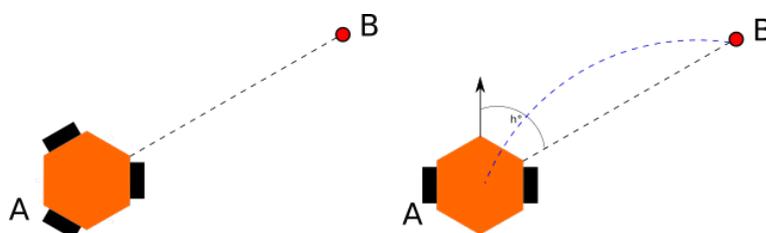


Figura 3.3: Trayectoria de desplazamiento de un robot del punto A hacia un punto B. Izquierda: Robot omnidireccional. Derecha: Robot diferencial

Para ilustrar las facilidades que introduce la omnidireccionalidad al momento de realizar el control de movimiento, analicemos la siguiente situación. Supongamos que tenemos 2 robots, uno diferencial y otro omnidireccional (Ver figura 3.3) y ambos se desean desplazar desde su posición actual al punto B. Al robot omnidireccional bastará con indicarle la dirección hacia donde deseamos que se desplace y la velocidad con la que queremos que lo haga. Luego el robot

omnidireccional saldrá dirigido al punto B, en la trayectoria indicada por la línea punteada que se muestra en la figura 3.3 y para ello, deberá controlar que la velocidad de sus ruedas se mantenga en todo el trayecto. o, El robot diferencial en este escenario no podrá salir dirigido en la dirección hacia donde se encuentra el punto B, por lo que deberá planificar una trayectoria similar a la línea azul mostrada en la figura 3.3 o girar una cantidad de grados suficientes que le permitan alinearse con el punto B, para luego comenzar a avanzar hasta llegar a él. En el primer caso, el robot deberá permanentemente controlar que la velocidad de las ruedas mantengan la diferencia necesaria para que el robot pueda realizar una trayectoria similar a la que muestra la línea azul. Sin embargo el cálculo de estas velocidades no es tan sencillo como el del robot omnidireccional. En el segundo caso, primero se deberá controlar que el giro se realiza correctamente, y en segundo lugar realizar un control para que los motores mantengan la misma velocidad y poder avanzar en línea recta. Es decir se realiza un número mayor de controles.

Las ruedas utilizadas, son ruedas compatibles con el kit NXT Lego Mindstorm compuestas principalmente por plástico. Dichas ruedas se seleccionaron porque son ruedas que poseen la particularidad de tener rodillos de goma ubicados de manera perpendicular tanto al radio de la rueda como al eje de la misma. Esta particularidad le brinda un grado más de libertad, y ubicándolas en la disposición mostrada en la Figura 3.4, transforma al prototipo en un prototipo holonómico.

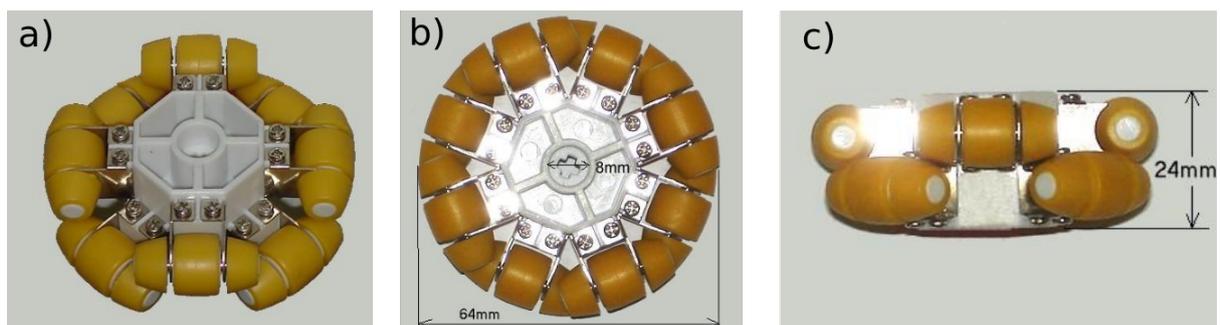


Figura 3.4: a) Rueda omnidireccional compatibles con el kit NXT Lego Mindstorm. b) Vista de perfil. c) Vista superior. Imágenes tomadas de [41].

Si bien existen una infinidad de ruedas con esas características, al igual que los motores, el modelo utilizado es el que se encontraba disponible en el laboratorio y que permitió pasar rápidamente a la implementación de comportamientos relacionados al movimiento del robot. Este argumento se debe a la necesidad de tener rápidamente un prototipo funcional con el cual los investigadores del proyecto PRJE puedan experimentar, probar actividades y validar ideas. En la Tabla 3.1 se muestran las especificaciones de las ruedas utilizadas.

Los motores utilizados para la construcción del prototipo, son motores servo de rotación continua contruidos por SpringRC, modelo SM-S4303R. Estos motores al ser servo, no requieren de componentes adicionales para realizar el control, como lo requeriría un motor de corriente continua. A su vez, la comunicación con dicho motor es una comunicación directa y sencilla. En comparación con otros motores disponibles en el laboratorio como los dynamixel AX-12, para los cuales es necesario implementar un protocolo para tener una comunicación exitosa. Además, los motores elegidos presentan un menor consumo y un mayor tamaño. Ver 3.2.

Para montar los motores a la plataforma, se utilizó como base un modelo de acoples obtenidos de [42], la cual se encuentra diseñada para montar el modelo de motores elegidos con el objetivo de crear una plataforma omnidireccional.

Características	Detalles
Dimensiones	64 mm de diámetro, 12 mm de ancho.
Peso	52 gr.
Capacidad de carga	25 Kg por rueda
Material	Goma de alta fricción, marco de plástico, ejes y marcos de los rodillos en acero inoxidable.
Compatibilidad	Directa con el kit Lego MinsStorm adaptable a otros kits

Tabla 3.1: Especificaciones de las ruedas omnidireccionales compatibles con el kit NXT Lego Mindstorm.[10]

	Velocidad Máxima (RPM)	Torque de detención (N*m)	Peso (g)	Tamaño (mm)	Precio (US\$)
SpringRC SM-S4303R	54 (6V)	0.54 (6V)	41	41.3 x 20.7 x 40.2	12.95
Dynamixel AX-12	59 (12V)	1.5 (12V)	54.6	32mm x 50mm x 40mm	44.90

Tabla 3.2: Tabla 2 - Comparación de los motores SpringRC SM-S4303R y Dynamixel AX-12.[11] [12]

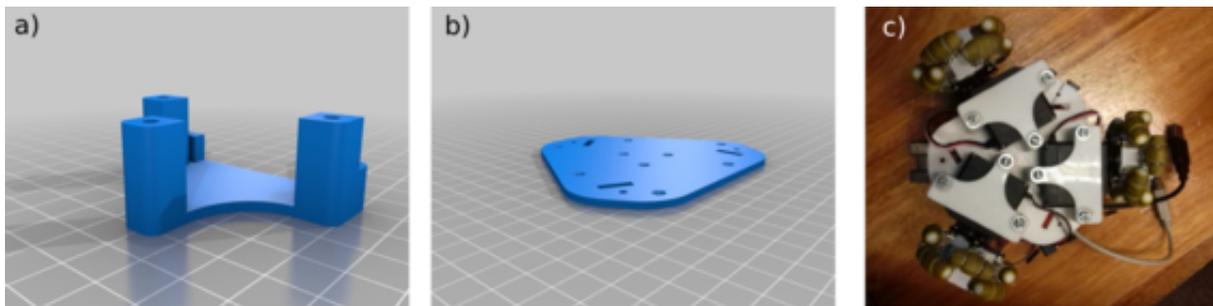


Figura 3.5: a) Plataforma. b) Agarre de los motores a la plataforma. c) Piezas ensambladas al robot. Las imágenes a y b fueron tomadas de 3.5 y la imagen c es de elaboración propia.

Matriz Led

Como ya se mencionó al inicio de esta sección, el otro esquema que compone el esquema de actuación aparte del esquema motor, es el esquema compuesto por la Matriz led.



Figura 3.6: Módulo MAX7219 con una matrix de 8x8 leds monocromáticos. [47]

El prototipo cuenta con una matriz de 8x8 leds monocromáticos, figura 3.7. La integración

de dicha matriz al prototipo se justifica en la necesidad de que el robot indique que sensores se encuentran activos en todo momento. Para ello, se representan los 6 lados de un hexágono con un subconjunto de los 64 leds de la matriz, y a cada representación se buscó alinearla a un sensor. De esta manera, cada vez que un sensor se encuentra activo, es decir está midiendo una distancia donde se supone que detectó un objeto, el lado correspondiente del hexágono se encenderá. Ver figura 3.7.

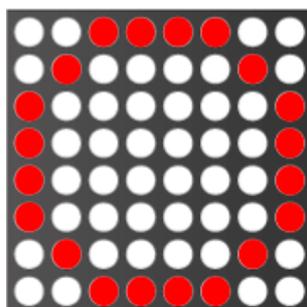


Figura 3.7: a) Esquema de formación del hexágono en la matriz led. Imagen de elaboración propia. b) Imagen del robot, con el comportamiento de encendido y apagado de los lados del hexágono implementado.

La matriz led se encuentra controlada por un circuito integrado llamado MAX7219 [23]. Este circuito también es utilizado para controlar displays 7 segmentos y posee la capacidad de extender la matriz led, encadenando módulos de 8x8 leds.

3.2. Esquema de sensado

En la construcción del estado del arte, fue posible observar que la mayoría de los robots utilizan sensores de distancia para conocer los objetos que se encuentran a su alrededor, principalmente para detectar aquellos que se encuentran en la trayectoria en la cual desea moverse. Por ejemplo, el Ozobot (robot investigado en el documento de Estado del Arte) posee sensores hacia adelante ya que es un robot diferencial, y por lo tanto desea poder detectar obstáculos que se encuentren frente al mismo al momento de avanzar.

Al pretender que el robot utilizado en el proyecto PRJE defina su comportamiento según la disposición física de los elementos que se encuentran a su alrededor, se definió que el robot presentara un anillo de sensores que le permita detectar objetos en la mayor cantidad de direcciones posibles. Esta técnica es una técnica conocida y utilizada en varios robots como por ejemplo el Kephra [13]. Para implementar dicha técnica se decidió utilizar 6 sensores, ubicados uno en cada vértice de un hexágono imaginario con centro en el eje vertical del robot como se muestra en la Figura 3.8.

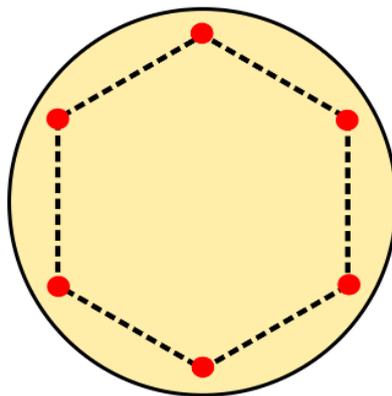


Figura 3.8: En amarillo se muestra una vista de arriba del contorno de un robot, y en rojo la ubicación de los sensores.

Aspectos importantes para la elección de un sensor

Podría definirse un sensor como un transductor que transforma una magnitud física en otra procesable. De la mano de esta definición, es importante identificar algunas características que nos permiten clasificar los sensores en distintas subcategorías. Por lo tanto, y previo a presentar los sensores de distancia elegidos, se presentarán algunos de los aspectos que nos permitieron identificar algunas de las características deseables de los sensores electos.[25]

Desde el punto de vista de la transferencia de la información, esta se puede realizar de 2 maneras:

1. Vía interrupciones, donde el sensor genera una interrupción para indicar un nuevo valor sentido.
2. Mediante la técnica de polling, donde la CPU rutinariamente está chequeando si existe una nueva medida o no.

A estas 2 formas de transmitir la información se las denomina iniciada por el sensor e iniciada por la CPU. Esta última es la que consume una mayor cantidad de tiempo de CPU, ya que debe estar de forma periódica consultando si existe una nueva medida. En cambio con la transferencia por interrupciones, por lo general la CPU dedica tiempo a obtener los datos una vez que el sensor genera una interrupción indicando que tiene una nueva medida [25]. La interfaz utilizada para la transmisión de información puede verse resumida en la tabla 3.2

Interfaz	Ejemplo
Señal binaria (0 o 1)	Sensor táctil
Señal analógica (ej. 0 a 5V)	Inclinómetro, sensor de distancia
Señal de tiempo (ej. PWM)	Giroscopio
Link serial (RS232, USB)	Módulo GPS
Link paralelo	Cámara digital

Tabla 3.3: Interfaces utilizadas para la transmisión de información por parte de algunos sensores. Tabla tomada de [25]

Si bien desde el punto de vista de un ingeniero es necesario clasificar los sensores según lo mostrado por su interfaz, también puede ser necesario clasificarlo desde otros punto de vista como:

- Punto de vista del Robot
 - Local: sensores montados sobre el robot.
 - Globales: sensores que están montados en el entorno y transmiten
- Sistema robótico móvil
 - Miden características internas del robot
 - Miden características del entorno externas al robot.
- En relación a la estimulación del ambiente
 - Pasivos: sensores que monitorean el ambiente sin distorsionarlo, por ejemplo una cámara, una brújula o un termómetro.
 - Activos: estimulan el entorno para tomar una medida, por ejemplo un sonar o un sensor láser.

Los sensores que presentan una interfaz binaria son los más simples de todos, ya que transmiten un bit de información. Esto facilita mucho la construcción de dicha interfaz para comunicarse con el microcontrolador, ya que basta con utilizar una entrada digital.

Un gran número de sensores producen una salida analógica en lugar de digital. Esto implica la necesidad de convertir esa señal analógica a digital mediante un conversor Analógico Digital (A/D), el cual podemos definir como un dispositivo electrónico capaz de convertir una señal analógica, ya sea de tensión o corriente, en una señal digital. Estos conversores por lo general presentan características como: 1) Accuracy: que se expresa en la cantidad de dígitos que produce a partir de un valor, 2) Velocidad: expresada en la máxima cantidad de conversiones por segundo y 3) Rango de medición: expresado en voltios.

Los sensores digitales son por lo general más complejos que los sensores analógicos, pero a la vez más precisos. El formato de salida de los sensores digitales puede ser implementado de diversas maneras mediante una interfaz paralela (por ejemplo un bus de 8 o 16 bits), una interfaz serial (USB o el estándar RS232) o una interfaz “sincrónica serial”.

Sensores de distancia

Los sensores de distancia se encuentran dentro de los sensores más comúnmente usados para realizar diverso tipo de tareas como: detectar el objeto más cercano al robot, construcción de mapas, ubicarse, entre otras. Dentro de las tecnologías utilizadas para implementar dichos sensores se encuentran el ultrasonido, el láser y el infrarrojo. A continuación se describe cada una de ellas junto con ejemplos de sensores comúnmente utilizados.

Los sensores que utilizan la tecnología ultrasónica, se basan en el siguiente principio: se emite una señal acústica de aproximadamente 1 ms de duración a una frecuencia ultrasónica, por lo general de 50 a 250 kHz, y se mide el tiempo en que se demora en recibir el retorno de la señal. El tiempo medido es proporcional a 2 veces la distancia al objeto más cercano. En caso de haber esperado un tiempo suficiente y no haber recibido retorno se considera que no hay un objeto cerca. Típicamente se repite este proceso 20 veces por segundo. En la Figura 3.9 se puede observar un esquema que lo representa. Las principales desventajas de este tipo de sensores son la interferencia y las reflexiones. Cuando la onda enviada es reflejada, un objeto parece estar más lejos de lo que verdaderamente está. La interferencia se expresa cuando se trabaja con varios sensores, o existe otra señal ultrasónica de frecuencias similares provocando que las señales interfieran entre sí. Al trabajar con varios sensores, puede suceder que un sensor capture la onda enviada por otro e incorrectamente asuma que el objeto se encuentre más cerca. Sin

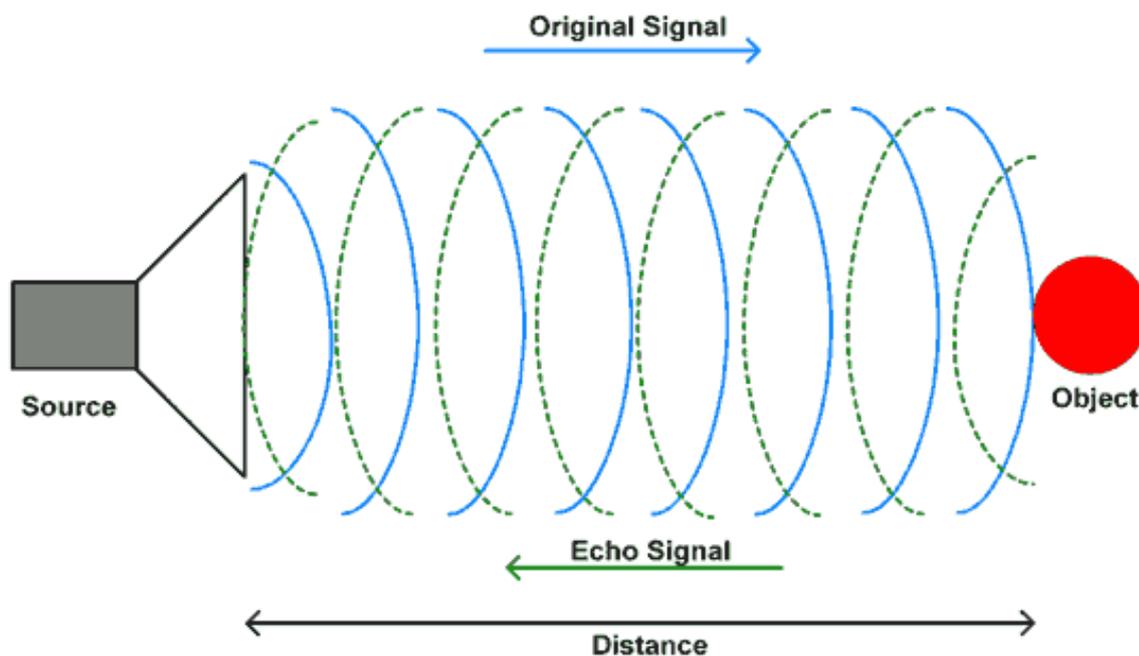


Figura 3.9: Principio de funcionamiento de los sensores ultrasónicos. [15]

embargo existen algunas soluciones para mitigar el problema de interferencia, como codificar la señal enviada.

El HC-SR04, figura 3.10, es un sensor ultrasónico que permite medir distancias de 2 cm a 400 cm. El módulo del sensor se compone por un emisor ultrasónico, un receptor y un circuito de control, y tiene un tamaño aproximado de 15mm x 20mm x 45mm. [14]

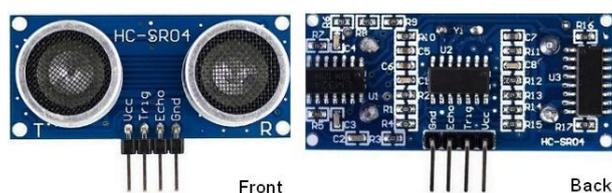


Figura 3.10: Sensor ultrasónico HC-SR04. [15]

El funcionamiento de este sensor podría resumirse en el envío de un pulso ultrasónico corto en tiempo 0. En caso de ser reflejado en un objeto, la señal recibida se transforma en una señal eléctrica para ser enviada como la salida del sensor. Para enviar el siguiente pulso, es necesario esperar un tiempo suficiente para que el eco de la señal enviada anteriormente se haya desvanecido. En la tabla 3.2 se muestran algunas características eléctricas del sensor.[15]

El sensor LV-MaxSonar-Ez MB1000, figura 3.11, presenta un rango de medición que va de 15,5 cm a 6 m, con una resolución de 2.5 cm. También es capaz de detectar objetos a distancia 0, sin embargo todo aquello que detecte por debajo de 15,5 cm lo mostrará como si estuviera a 15,5 cm. El principio de medición que utiliza es el mismo que el HC-SR04, donde permite tomar mediciones cada 50 μ s, es decir con una frecuencia de 20 Hz. A su vez presenta una salida analógica que está continuamente enviando las medidas tomadas, así como un modo donde las medidas son iniciadas a demanda. Estas interfaces están activas simultáneamente.[16]

Voltaje de trabajo	5 V
Corriente de trabajo	15 mA
Frecuencia de trabajo	40 Hz
Rango Máximo	400 cm
Rango Mínimo	2 cm
Ángulo de medida	15
Trigger input signal	10 uS TTL Pulse
Echo output signal	Input TTL lever signal
Dimensiones	15mm x 20mm x 45mm.

Tabla 3.4: Tabla tomada de [14]

Voltaje de trabajo	2,5 a 5,5 V
Corriente de trabajo	2 mA
Frecuencia de trabajo	20 Hz
Rango Máximo	6 m
Rango Mínimo	0 cm
Dimensiones	20mm x 22 mm x 15 mm.

Tabla 3.5: Especificaciones del LV-MaxSonar-Ez MB1000. Tabla tomada de [16]

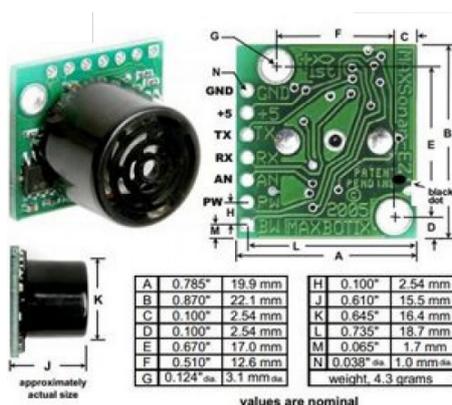


Figura 3.11: Sensor ultrasónico LV-MaxSonar-Ez MB1000. [16]

Una particularidad de este sensor es que presenta una interfaz que permite conectar varios sensores en serie, siendo esto una ventaja desde el punto de vista de la cantidad de puertos necesarios para su utilización. En la tabla 3.2 se muestran algunas características eléctricas del sensor.[16]

Otra tecnología para desarrollar sensores de distancias es la utilización de haces de luz infraroja. Este tipo de sensores se basa en un principio distinto a la medición del tiempo de vuelo de una señal, ya que el tiempo de vuelo de un fotón sería muy corto para medirlo con un sensor relativamente barato. Por este motivo se utiliza un pulso led de aproximadamente 40 kHz junto con un detector que nos permite observar cómo cambia el ángulo con el cual refleja el haz emitido en función de la distancia a los distintos objetos. La longitud de onda utilizada es por lo general 800nm. [25]

Los sensores Sharp, figura 3.12 son sensores que se componen por una combinación de un PSD (position sensitive detector), IRED (infrared emitting diode) y un circuito procesador de

señales. Utilizan el ángulo de incidencia de los haces emitidos para calcular la distancia a un objeto. Gracias a un sistema de triangulación interno que utilizan, estos sensores tienen gran tolerancia a los cambios en la reflectividad de los objetos y la temperatura del ambiente.



Figura 3.12: a) Sensor Sharp-gp2D120X. b) Sensor Sharp-2Y0A02. c) Sensor Sharp-2Y0A21.[43, 44, 45]

En la tabla 3.2 se muestran las características de 3 modelos de sensores sharp: Sharp-gp2D120X, Sharp-2Y0A02 y Sharp-2Y0A21, mostrados en la figura 3.12. [17][18][19]

Sensor	Sharp-gp2D120X	Sharp-2Y0A02	Sharp-2Y0A21
Rango de medida (cm)	4 to 30	20 to 150	10 to 80
Tiempo de respuesta (ms)	39	38 ± 10	38 ± 10
Corriente de trabajo promedio (mA)	33 mA	33	33
Voltaje de trabajo(V)	-0.3 to +7	-0.3 to +7	-0.3 to +7
Voltaje de salida en la terminal	-0.3 to ($V_{cc} + 0.3$)	-0.3 to ($V_{cc} + 0.3$)	-0.3 to ($V_{cc} + 0.3$)
Tamaño (mm)	29.5x13.0x13.5	44.5 x 18.9 x 21.6	44.5
Peso (g)	3.5	5	3.5

Tabla 3.6: Tabla construida a partir de los datos de [17][18][19]

Actualmente muchos de los sensores que utilizaban la tecnología de ultrasonido se han ido reemplazando por sensores que utilizan tecnología láser. El láser es una tecnología muy utilizada en la construcción de mapas, pudiendo construir mapas casi perfectos desde el punto de vista del robot, incluso algunos sensores permiten una construcción 3d del entorno. Hasta hace unos años se consideraba que los sensores que utilizaban tecnología láser eran muy grandes y costosos para ser utilizados en sistemas móviles y pequeños, sin embargo con el avance de la tecnología esto se ha podido superar, llegando a existir sensores del tamaño de una moneda, e incluso más chicos.

El VL53L0X de ST Microelectronics [20] es un sistema utilizado para medir distancia en función del tiempo de vuelo integrado en un módulo compacto (figura 3.14) que mide con precisión el tiempo que tardan los pulsos emitidos de la luz láser en llegar al objeto más cercano y reflejarse de nuevo en un detector (Ver figura 3.13). Este sensor puede informar distancias de hasta 2 m (6,6 pies) con una resolución de 1 mm, pero su alcance efectivo y precisión dependen en gran medida de las condiciones ambientales y las características del objetivo, así como la configuración del sensor. (La precisión del sensor se especifica para que oscile entre $\pm 3\%$ como máximo y más de $\pm 10\%$ en condiciones menos óptimas). Las mediciones de rango están disponibles a través de la interfaz I²C del sensor, que también se usa para configurar los ajustes

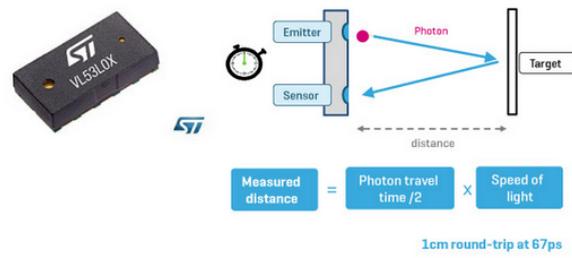


Figura 3.13: Principio de funcionamiento de los sensores láser. Mide con precisión el tiempo que tardan los pulsos emitidos de luz láser en alcanzar un objeto y ser detectados nuevamente. [21]

Dimensiones (mm)	13 x 18 x 2
Peso (g)	0.5
Voltaje de trabajo (V)	2.6 to 5.5
Corriente de trabajo* (mA)	10
Formato de salida (I ² C)	6-bit distance reading (in millimeters)
Rango de medida**	hasta 2 m (6.6 ft)

Tabla 3.7: Características del sensor VL53L0X. * - Varía con la configuración, objetivo, y condiciones de ambiente. El pico de corriente puede alcanzar los 40 mA. ** - El rango efectivo depende de la configuración, ambiente y el objetivo. Si bien la hoja de datos no especifica una distancia mínima, gracias a los experimentos realizados, puede suponerse que ronda los 4 cm. [22]

del sensor, y el sensor proporciona dos pines adicionales: una entrada de apagado y una salida de interrupción.

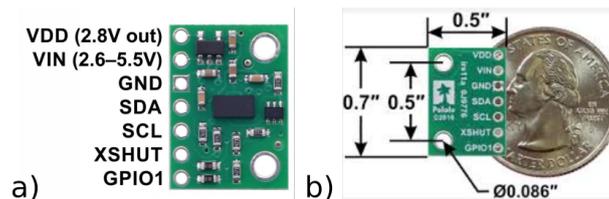


Figura 3.14: a) Sensor VL53L0X, con sus pines de entrada y salida. b) Medidas del VL53L0X en pulgadas y comparadas con una moneda de 1 dólar americano. Imágenes tomadas de [46]

El VL53L0X funciona con una tensión recomendada de 2,8 V, lo que puede dificultar la interconexión para microcontroladores que funcionan a 3,3 V o 5 V. Sin embargo la placa de soporte incluye un regulador de voltaje lineal de baja caída que proporciona los 2.8 V requeridos por el VL53L0X, que permite alimentar el sensor desde una fuente de 2.6 V a 5.5 V. En la tabla 3.2 se muestran las características de este sensor.

Elección de un sensor

La elección de los sensores a utilizar estuvo basada en los experimentos realizados que se presentan en el siguiente capítulo, y en los criterios que se detallan a continuación:

- Tamaño reducido: Surge de la necesidad de aprovechar al máximo el espacio disponible

para trabajar con los niños en PRJE. Se pretende trabajar con grupos de niños, donde cada uno trabajará con un robot buscando resolver ciertos desafíos, como modificar el entorno para que el robot pueda desplazarse de un punto a otro. Esto se pretende realizar en espacios menores a 2 m² por grupo, por lo que el tamaño del robot puede llegar a ser un factor determinante para alcanzar este objetivo. En ese sentido, se pretende seleccionar un sensor de distancia que limite lo menos posible el tamaño del robot a medida que aumenta la cantidad de sensores.

- Transmisión de la información: Como ya se mencionó esta transmisión puede realizarse de dos maneras, vía interrupciones o mediante la técnica de polling. La principal diferencia entre ambas es el tiempo de CPU que consumen. En la técnica de polling, cada vez que se requiere una medida es necesario consultarla, lo cual requiere por lo general de un tiempo de espera inactivo por parte de la CPU esperando que el sensor tome una medida. La técnica de interrupciones implica que cada vez que el sensor tenga una nueva medida intente interrumpir la CPU para que almacene o procese la medida. En caso de almacenarla, cuando el programador necesite de una medida, deberá consultar en el lugar de almacenamiento de dicha medida, y no deberá esperar a que el sensor realice todo el proceso de medición. Por estos motivos, y sumado a que la técnica de interrupciones permite paralelizar tareas mientras se toman medidas por parte del sensor, se buscará que el sensor a elegir transmita la información vía interrupciones.
- Consumo eléctrico: De la mano del criterio sobre el tamaño reducido, también es necesario considerar el consumo eléctrico de los sensores. Al sumar todas las componentes eléctricas, necesitamos una cantidad de energía determinada para lograr que nuestro robot funcione de manera correcta y autónoma durante determinada cantidad de tiempo. Esto genera que la batería a elegir para alimentar el robot deba cumplir con determinadas características, lo cual por lo general implica también que a medida que aumenta la necesidad de energía también aumenta el tamaño de la batería. Se pretende entonces que el sensor a elegir consuma la menor cantidad de corriente posible.
- Rango de medida: Asociado al tamaño del robot, es necesario que el robot tenga un rango de sensado adecuado ya que necesitará tomar acciones en función de las cosas que se encuentran a su alrededor. Por este motivo, se buscará que el rango de sensado sea el mayor posible de forma que permita experimentar con objetos a corta y larga distancia. Y en caso de necesidad, siempre es posible establecer un límite y tomar como desconocido todas aquellas medidas que excedan dicho límite.
- Precisión: La precisión juega un papel importante, debido a que se programarán distintos comportamientos en el robot, y alguno de ellos puede requerir que el robot actúe de determinada manera según la cercanía a un objeto. De no ser preciso, podría generar que el robot se comporte de forma diferente a la deseada en determinada situación y provocar una dificultad en el entendimiento de su comportamiento por parte de los niños.

El prototipo desarrollado usa sensores Sharp-2Y0A02 por las razones que se detallan en el Capítulo 5, donde se analizan los resultados de los experimentos realizados con los distintos sensores. Además, en el Capítulo de conclusiones, se exponen de los motivos de los cuales derivó la sugerencia de utilizar sensores VL53L0X para el desarrollo de futuros prototipos.

3.3. Plataforma de cómputo

Con el objetivo de determinar una plataforma de cómputo adecuada para construir el prototipo, se buscaron aquellas que presentaran las siguientes características:

- Tamaño reducido: Como ya se mencionó al elegir un sensor, en PRJE se pretende realizar en espacios menores a $2m^2$ por grupo de niños. Por lo tanto del resto de sus componentes es una cuestión fundamental y se buscará que sus componentes sean del menor tamaño posible.
- Bajo costo: Se pretende que la plataforma a construir sea económica en comparación con el resto de los kits robóticos que se utilizan para este tipo de actividades.
- Comunicación inalámbrica: Se pretende que el dispositivo utilice alguna tecnología que le permita comunicarse de manera inalámbrica, ya que se desea poder monitorear su estado interno a medida que se ejecutan los distintos comportamientos, así como poder configurar un modo para su controlar su movimiento mediante un joystick. Por ejemplo: Bluetooth o Wifi.
- Cantidad de entradas analógicas: Con el objetivo de poder utilizar sensores analógicos, se valorará que la placa a utilizar posea la mayor cantidad de entradas analógicas posibles.
- Buses i2c: Se pretende poder utilizar sensores que se comuniquen vía el protocolo i2c, por lo que la placa a utilizar debe poder soportarlo.
- Consumo: Considerando que deberá compartir la fuente de alimentación con los sensores y actuadores que se encuentren en el robot. Se considera como una ventaja que el dispositivo a elegir consuma la menor cantidad de energía posible.

La primer decisión de diseño que se tomó para elegir la plataforma de cómputo fue elegir entre utilizar un microcontrolador u otro sistema computador. Un sistema computador esta compuesto por un computador más periféricos. Con un computador, nos referimos a una o varias CPUs, más memoria y puertos de entrada y salida.

La particularidad de un microcontrolador es ser un sistema computador contenido un solo circuito integrado orientado a aplicaciones de control. Es decir, un sistema donde la CPU, memoria y entrada/salida se presentan dentro de un mismo circuito integrado. [24] Ejemplo de este tipo de placas son: Arduino Uno, Arduino Mega, Arduino Nano, Esp 32 thing, Photon particle, etc.

Como alternativa se manejó el uso de una Single Boards Computer (SBC). Las SBCs cuentan con uno o varios procesadores, cuya comunicación con dispositivos externos se logra a través de un bus de datos, por lo tanto, el chip que contiene a los microprocesadores presenta principalmente pines de datos y direcciones, así como algunos de pines de control. Todos los dispositivos periféricos (memoria, controlador de disquete, controlador USB, temporizador, etc.) están conectados a uno o más buses[30]. Para el proyecto se consideraron como posibles opciones algunos modelos de SBCs como por ejemplo ODROID-C0, Orange Pi Cero H2 , Odroid XU4, entre otras.

La principal diferencia entre las placas microcontroladas y las microprocesadas radica en que las segundas poseen una Unidad de manejo de memoria, lo que les permite tener capacidad del orden de los MB o GB, e implementar técnicas de swapping, paginado y memoria virtual. Mientras que los microcontroladores poseen memorias pequeñas, en el orden de kB a MB.

Otra diferencia es que los microcontroladores poseen una gran cantidad de puertos de entrada y salida, para comunicarse con los periféricos, en comparación con algunos pocos que poseen los microprocesadores. Incluso, algunos llegan a tener soporte en hardware para implementar protocolos de comunicación comúnmente utilizados por algunos periféricos.

Es común que los microcontroladores se utilicen en aplicaciones de control con restricciones

Plataforma	Micro ó SBC	Peso (g)	Tamaño (mm)	GPIOs *	Precio (US\$) **
Orange Pi Cero H2	SBC	70	85 x 55	26	16,8
Odroid C0	SBC	50	58 x 56 x 11	40	28,95
Odroid XU4	SBC	50	83 x 58 x 20	40	80
Photon particle	Micro	5	36.58 x 20.32 x 6.86	18	26
Arduino Nano	Micro	7	18 x 45	22	5
Esp 32 thing	Micro	15	55x22	32	20

Tabla 3.8: Tabla comparativa de las principales características para los microcontroladores y SBC considerados en el proyecto. * - General Purpose Input and Output. ** - Se tuvo como referencia los sitios web de amazon [33] y ebay [32]

de tiempo real fuertes, pues es posible mantener un control preciso del flujo del programa y del reloj. Algunos tienen la capacidad de responder a interrupciones generadas por eventos del hardware. Por otra parte, los microprocesadores por lo general utilizan sistemas operativos de propósito general con varios modelos de concurrencia, como la memoria compartida, pipes, etc. Son utilizados para correr múltiples aplicaciones en paralelo y satisfacer demandas en tiempo real blando. [24]

La primer selección de plataforma de cómputo se realizó según el tamaño de la plataforma ya que desde el proyecto PRJE se pretende obtener una plataforma robótica de tamaño pequeño. Luego de realizar varias búsquedas en distintos sitios de venta a través en la web, se consideraron las siguientes opciones (se incluyen placas de entrada y salida y algunas Single Board Computer) que se muestran en la tabla 3.3.

Luego de realizada esta búsqueda, y teniendo en cuenta los criterios antes mencionados, se tomaron las siguientes decisiones:

1. Proponer la placa Esp 32 thing para desarrollo de futuros prototipos, por ser una placa que presenta una gran cantidad de pines analógicos a disposición del desarrollador y dimensiones pequeñas en comparación con el resto de las SBCs. Es una placa barata en comparación con ambas Odroids, con el precio es similar al de la Photon particle.
2. Por disponibilidad en el laboratorio, se decidió implementar un primer prototipo para realizar pruebas de concepto con una Photon particle [34]. Esta placa microcontrolada posee un chip Cypress WI-FI [35] que le permite utilizar dicha tecnología, y por lo general es usado para implementar aplicaciones relacionadas al internet de las cosas. El microcontrolador que posee es un STM32 ARM Cortex M3 [36].

A continuación se listan algunas características de la placa Photon particle obtenidas de [34]:

- Particle PO Wi-Fi module
 - Broadcom BCM43362 Wi-Fi chip
 - 802.11b/g/n Wi-Fi
 - STM32F205RGY6 120Mhz ARM Cortex M3
 - 1MB flash, 128KB RAM
- On-board RGB status LED (ext. drive provided)
- 18 Mixed-signal GPIO and advanced peripherals
- Open source design

- Real-time operating system (FreeRTOS)
- Soft AP setup
- FCC, CE and IC certified

Capítulo 4

Diseño de Software

En esta sección presentaremos en forma detallada el diseño y desarrollo de software para ser utilizado en el prototipo a construir. A diferencia de la sección Diseño de Hardware, en la presente, se detallan decisiones relacionadas al prototipo construido para la validación de ideas en relación a las actividades a realizar con el proyecto PRJE.

Comenzaremos este capítulo con una descripción del firmware utilizado, al cual se le integrarán los módulos desarrollados como forma de lograr implementar los comportamientos deseados en el robot. A continuación, se describirán los módulos desarrollados con el objetivo de controlar el hardware del robot (led, sensores de distancia y motores). Se presentará un módulo que utiliza a los anteriormente descritos y oficia de vínculo de estos con el resto del firmware. Finalmente en dicha sección como parte de la descripción del módulo de control, se presentará el modelo utilizado para interpretar la información proveniente de los sensores, así como para generar la información que se le envía a los motores.

4.1. Firmware Particle

Como se indicó en el capítulo 3, la plataforma de computo a utilizar fue placa Photon Particle. Esta placa, al igual que arduino, posee un firmware de bajo nivel que implementa las funciones básicas de cualquiera de los dispositivos Particle. Se podría ver este firmware como una especie de sistema operativo para los dispositivos Particle. [37]

Así como un sistema operativo a una computadora, dicho firmware proporciona una base para que se ejecuten otras aplicaciones. Más específicamente, permite que el código de la aplicación (la parte del firmware que escribe el usuario programador) se ejecute correctamente al exponer los comportamientos subyacentes del dispositivo.

El firmware particle abstrae gran parte de la complejidad de la experiencia de desarrollo de firmware tradicional. Algunas de sus responsabilidades específicas incluyen:

1. Comunicación segura: En los casos de que existe una conexión con la nube particle, garantizar que todas las comunicaciones entre el dispositivo y la nube particle estén autorizadas y encriptadas.
2. Abstracción de hardware: Proporcionar una interfaz única y unificada para el dispositivo, independientemente de la arquitectura de hardware subyacente.
3. Habilitación de la aplicación: Exponer una API de variadas características que los desarrolladores utilizan para escribir aplicaciones para el dispositivo.

Módulo	Identificador
Admin	0
Action	1
Distancia	2
Control	3

Tabla 4.1: Identificación de los distintos módulos.

4. Actualizaciones por aire: Al estar conectado a la nube Particle, permitir cambios rápidos y remotos en el código que se ejecuta en el dispositivo, a la vez que brinda flexibilidad en entornos de conectividad deficiente.

A diferencia del firmware realizado por el programador de la aplicación, el firmware particle del dispositivo es escrito y mantenido principalmente por el equipo de desarrollo de Particle. Esta es una decisión de la empresa Particle con el objetivo de mantener al desarrollador enfocado en su caso de uso particular sin necesidad de incluir los matices de los comportamientos de dispositivos de bajo nivel para otros casos de uso.[38]

Las bibliotecas son una parte central del desarrollo de proyectos en las plataformas Particle. La idea detrás de una biblioteca Particle es poner a disposición del programador una colección de código de firmware reutilizable que se puede agregar fácilmente a uno o varios proyectos de partículas.

Aprovechar las bibliotecas permite mantener y reutilizar fácilmente bloques de código comunes, así como incorporar y aprovechar otras bibliotecas de terceros existentes del ecosistema de dispositivos Particle.[38]

Para el desarrollo del dispositivo, se utilizó el firmware Particle, con el fin de tener a disposición las bibliotecas desarrolladas por el equipo Particle. Se utilizaron bibliotecas para: el control de los servo, la conexión a una red WIFI y la utilización del protocolo UDP y la inclusión de bibliotecas arduino compatibles.

MCC

MCC es una aplicación obtenida de [39] que se incorporó al firmware Particle, y que permite mediante una comunicación por sockets UDP, intercambiar mensajes con otro host.

Esta aplicación es una implementación de una máquina de estados que se ejecuta en el microcontrolador, y se utiliza para atender solicitudes que recibe el robot mediante mensajes UDP. Esta nueva funcionalidad, permitió debuggear el código implementado, así como obtener un estado interno del robot mientras se encontraba en funcionamiento.

La máquina de estados utilizada es la que se encuentra en la figura 4.1. Al recibir un mensaje, la máquina debe procesarlo para poder extraer la información necesaria para ejecutar alguna función implementada en los distintos módulos que contiene el firmware desarrollado y luego deberá elaborar un mensaje que será enviado como respuesta al mensaje que desencadenó la ejecución. Para procesar un mensaje, es necesario poder identificar los distintos módulos que se han desarrollado. Por tanto, a medida que se desarrollan módulos se debe respetar una serie de identificadores para que cada módulo posea un identificador único. En la tabla 4.1 se indican los identificadores de los distintos módulos desarrollados.

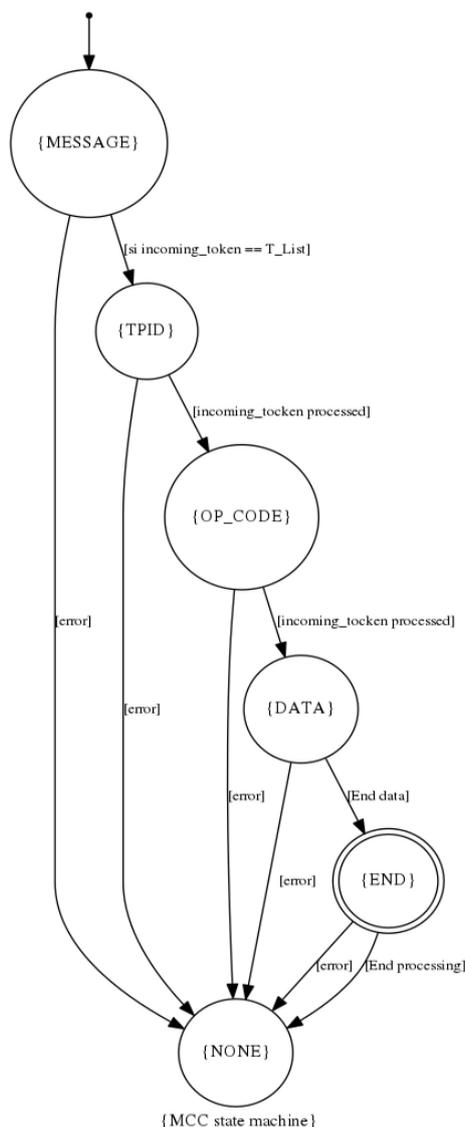


Figura 4.1: Máquina de estados que interpreta mensajes UDP, dirigidos a MCC.

En la figura 4.1 se encuentra la máquina de estados que implementa la aplicación MCC. La misma tiene 6 estados que se describen a continuación:

- MESSAGE: Procesa el inicio de un mensaje.
- T_PID: Indica el módulo al cual se dirige el mensaje.
- OP_CODE: Indica la función del módulo se desea ejecutar.
- DATA: Si es necesario, se procesan los datos que se pasan como parámetros para la función correspondiente del módulo identificado con el parámetro t_pid y que se identifica con op_code.
- END: Envía una respuesta que la elabora a partir de ejecutar la función indicada en el mensaje entrante, con los parámetros obtenidos en el parseo.
- NONE: Este implica que ya se terminó de procesar el mensaje.

Los mensajes que el microcontrolador puede procesar deben respetar el formato bencode [31]. Este formato utiliza caracteres ASCII como delimitadores y dígitos. A continuación se listan las reglas de codificación que se utilizaron en el intercambio de mensajes con el robot:

1. Un entero se codifica como $i <entero\ codificado\ en\ base\ 10\ ASCII> e$. Por ejemplo, el número 42 se representa como $i42e$, el 0 como $i0e$, y el -42 como $i-42e$. El -0 no está permitido.
2. Un byte string o secuencia de bytes se especifica como $<largo> :<contenido>$. El largo debe estar codificado en base 10 como los enteros y debe ser no negativo. El 0 está permitido. El contenido es la secuencia de bytes que constituyen el string. Así el string "ping", se codifica como $4:ping$.
3. Una lista se representa como $l<contenido>e$. El contenido de la lista, consiste en la codificación en bencode de los elementos que contiene, en orden y concatenados. Así la codificación de la lista que contiene el número 2 y el string "ping", se codifica como $li2e4:pinge$.

Un mensaje bien formado que será enviado al microcontrolador, debe estar construido como una lista de elementos donde el primer elemento es el identificador del módulo que contiene la función que se desea ejecutar. El segundo es el identificador de la función que se desea ejecutar (los identificadores de las funciones son únicos para cada módulo). Y por último, una lista con los parámetros que debe recibir la función. Es decir:

$$\text{Mensaje} = l<id_módulo><id_función>l<parámetros>ee$$

Donde $<id_módulo>$, $<id_función>$ y $<parámetros>$ se encuentran codificados en bencode.

Con MCC integrada al firmware particle, podemos representar el funcionamiento del firmware que contiene el microcontrolador del robot con el siguiente pseudocódigo:

```
while (true):

    /* ejecutar todas las funciones para
    el correcto funcionamiento del firmware
    particle */

    if ((existe un mensaje) and (el mensaje es correcto)) then
        ejecutar la función correspondiente
    end /* if */

    for función in poll_callbacks
        ejecutar función
    end /* for */
end /* while */
```

En este pseudocódigo, podemos observar una lista llamada `poll_callbacks`. Esta es una lista donde al momento de inicializar el firmware, es posible agregar algunas funciones que deben ser ejecutadas en todos los ciclos de ejecución del `while`. Además, permite también, agregar y quitar funciones dinámicamente en tiempo de ejecución. Por tanto, sería posible modificar el comportamiento del robot en dicho tiempo.

4.2. Firmware del prototipo

Luego de presentado el funcionamiento general del Firmware y de la aplicación MCC que nos permite intercambiar mensajes para monitorear el funcionamiento del robot de manera remota. En esta sección presentaremos cómo se desarrolló e introdujo el firmware para controlar el prototipo.

Basado en lo expuesto en el diseño de hardware es posible identificar 3 categorías de componentes. Por un lado las componentes de sensado, donde se obtiene la información del medio, por otro las componentes de actuación como son los motores y la matriz de leds, y finalmente el componente de control. Cada módulo controla un módulo de hardware y a su vez se compone por otros submódulos que se especializan en determinadas tareas.

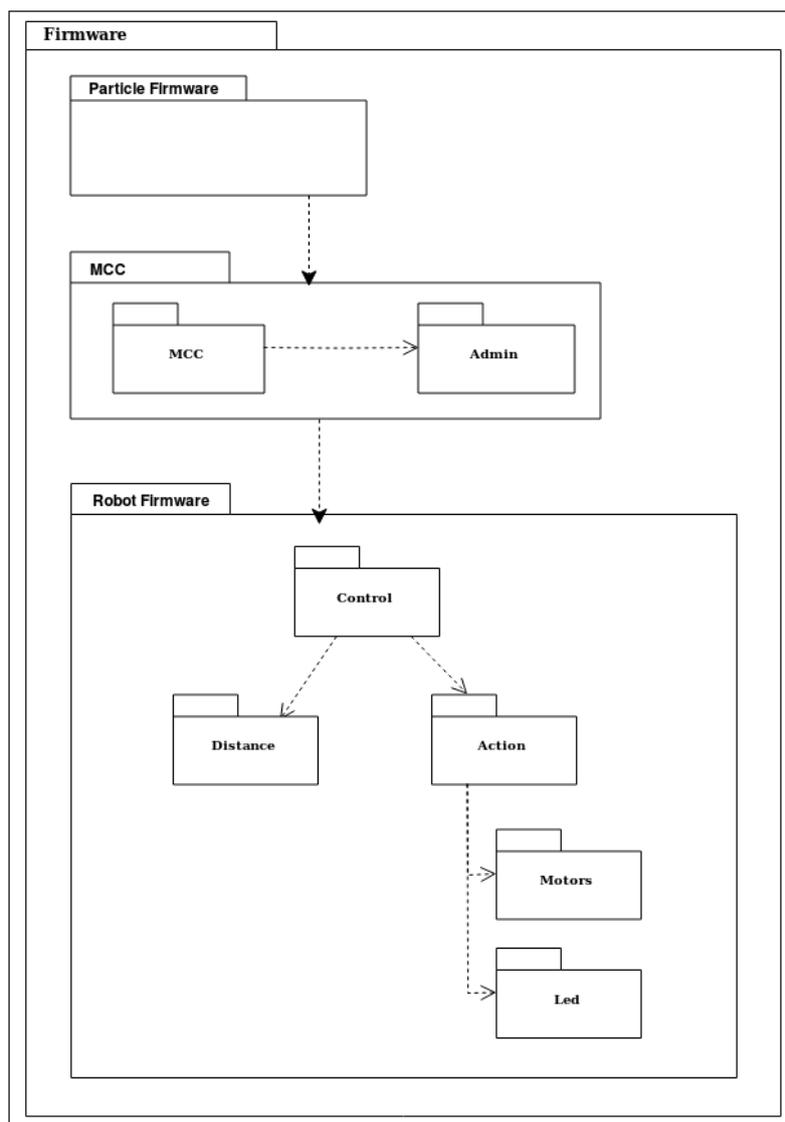


Figura 4.2: Diagrama de la arquitectura del software construido.

De esta manera, la arquitectura quedó planteada como se muestra en la figura 4.2. En dicha figura es posible observar 3 grandes módulos dentro del firmware que contiene el microcontrolador. El módulo correspondiente a el Firmware particle y el módulo MCC, el cual tiene 2 submódulos. Uno de ellos presenta el mismo nombre, y es el que implementa la máquina de

estados para procesar los mensajes y ejecuta todas las funciones en cada loop de ejecución. El otro submódulo es Admin, el cual contiene una serie de funciones orientadas principalmente a debugging y se describen en la siguiente sección.

Por último tenemos el módulo del firmware del robot, implementado como parte del trabajo realizado durante el presente proyecto. Este módulo se compone por 3 submódulos principales, que son:

1. Módulo Control: fue implementado en base a un paradigma reactivo y es donde se implementan los distintos comportamientos.
2. Módulo Distancia: tiene como objetivo la comunicación con el hardware de los sensores de distancia y realizar un pequeño preprocesamiento de la información obtenida.
3. Módulo Action: tiene como objetivo la comunicación con los elementos de hardware correspondientes a los actuadores, donde para cada actuador presenta un submódulo diferente (Motores y Led).

Esta arquitectura construida a partir del esquema determinado en el diseño y construcción del hardware presenta algunas ventajas que pueden considerarse importantes desde el punto de vista del desarrollo de software y construcción del robot para utilizar en el proyecto PRJE.

Presenta un bajo acoplamiento ya que los módulos son independientes entre sí, y es posible crear nuevos módulos, modificar los existentes o eliminarlos, sin necesidad de modificar de manera sustancial el resto de los módulos siempre que se respeten las interfaces.

Los módulos tienen una alta cohesión debido a que todo el código que contienen está orientado a trabajar con el mismo elemento de hardware. Por lo tanto en caso de querer modificar la forma de comunicarse con algún elemento de hardware es fácilmente identificable a que módulo hay que dirigirse.

Es posible cambiar la tecnología a utilizar para prototipar sin tener que modificar el firmware salvo el módulo orientado a comunicarse con el elemento a cambiar. Donde nos podemos enfrentar a 2 situaciones, se modifica dicho módulo o se cambia por uno que respete las interfaces e implemente la comunicación con otra tecnología.

A continuación se detallarán los distintos módulos antes mencionados. Se describirá la lógica de su funcionamiento, las funciones que contienen y qué representan.

Módulo Control

Como ya se mencionó, al programar este módulo se utilizó un paradigma reactivo. Un sistema con estas características vincula estrechamente la percepción con la acción sin la necesidad de hacer intervenir a alguna representación abstracta del entorno o historial.[25] Es decir que la información sensada, es minimamente procesada y se utiliza para desencadenar comportamientos o reacciones, los cuales determinan el comportamiento general del robot.

Esto aplicado a la construcción del módulo control, se refleja en que dicho módulo una vez inicializado, toma la información proveniente del módulo distancia, determina a dónde debe dirigirse el robot y desencadena 2 comportamientos diferentes: le indica al módulo motores la dirección hacia donde debe dirigirse el robot y le indica al módulo led los sensores que se encuentran frente a un objeto.

Al momento de su creación, este módulo registra una función para realizar un reporte de su estado en tiempo de ejecución al recibir un mensaje. Y en la lista `poll_callbacks` registra una función que a partir de las distancias sensadas por los sensores de distancia, calcula y almacena una dirección en la cual el robot debe moverse. De esta manera le indica a el módulo motores en qué dirección debe moverse el robot.

Para realizar el cálculo de la dirección en la que el robot debe moverse, se utilizaron los principios de una arquitectura de campo potencial. Donde la acción motora de un comportamiento debe representarse como un campo potencial. Es decir, como una matriz, o campo de vectores. Un vector es una construcción matemática que consta de una magnitud y una dirección. Los vectores se usan a menudo para representar una fuerza de algún tipo.

Considerando que nuestro robot se moverá en un plano xy ubicado de forma paralela al suelo, y que durante la resolución de problemas el robot se mantendrá a la misma altura siempre. Suponiendo que los objetos perceptibles en el mundo ejercen un campo de fuerza en el espacio circundante, análogo a un campo magnético o de gravitación. Por tanto el robot puede ser considerado como una partícula que se mueve en un plano y se encuentra bajo los efectos del campo exudado por uno o varios objetos del entorno. Además, consideramos que los campos potenciales son continuos porque no importa cuán pequeño sea el elemento; en cada punto del espacio, presenta un vector asociado.

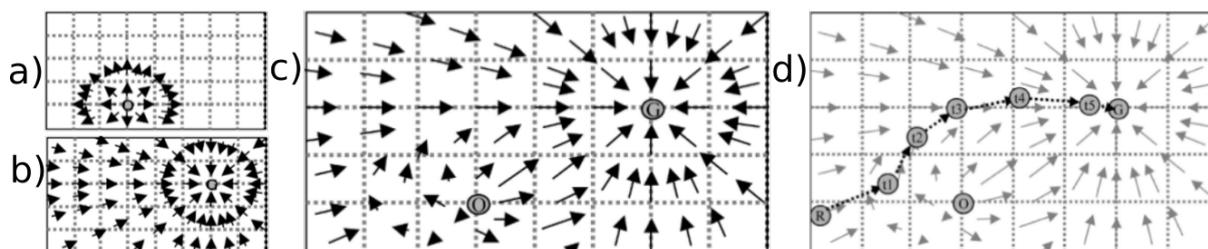


Figura 4.3: a) Campo de potencial generado por el objeto Goal. b) Campo de potencial generado por el objeto Obstacle. c) Combinación de los campos de potencial generados por ambos objetos y actuando sobre todo el entorno. d) Respuesta del robot frente a la fuerza que ejercen los campos en ti, hasta alcanzar el objeto Goal.[6]

Con este modelo, en cada punto en que se encuentre el robot, estará siendo sometido a una fuerza que es producto de la suma vectorial de todas las fuerzas que ejercen los distintos objetos que se encuentren a su alrededor. Y será dicha fuerza la que determine en qué sentido y con qué velocidad se moverá el robot. Ver figura 4.3.

Para determinar qué campos son aquellos que están realizando una fuerza sobre el robot, se considera la distancia del robot al objeto que emite el campo. Por tanto, el robot sólo considerará, para realizar el cálculo de la dirección y velocidad con que moverse, aquellas fuerzas que superen determinada magnitud. Para determinar la magnitud de las fuerzas que se ejercen sobre él, el robot realizará la diferencia entre la máxima distancia a la que se detectan objetos menos la distancia a la que actualmente encuentra el objeto que esta realizando la fuerza.

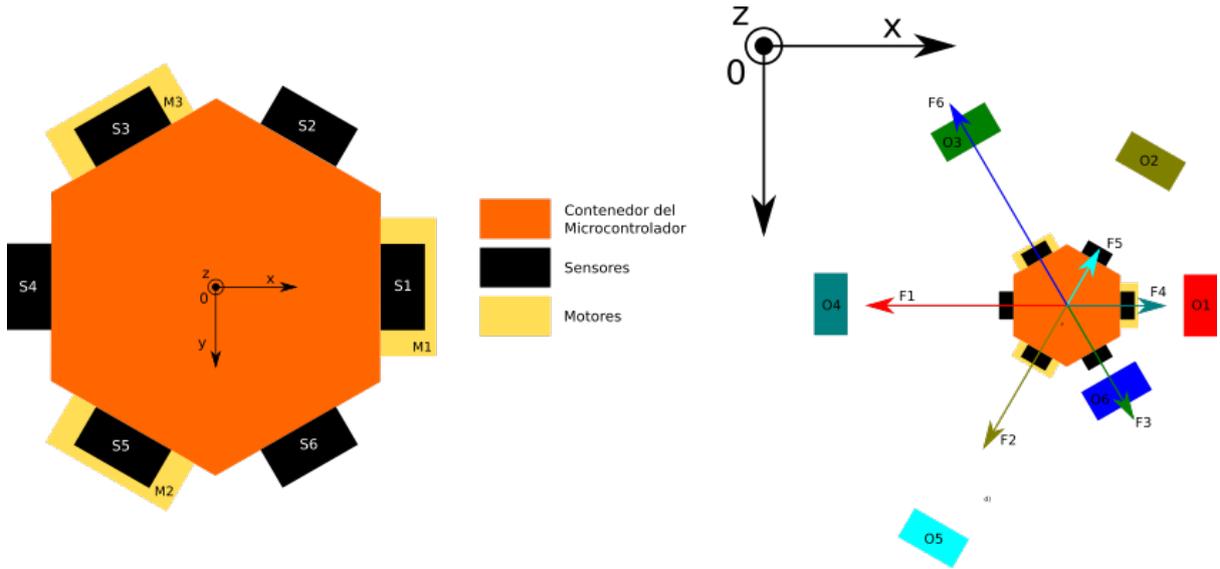


Figura 4.4: a) Esquema del robot construido. Derecha: Diagrama de cuerpo libre mostrando como actúa la fuerza F_i ejercida por el objeto O_i .

En la figura 4.4a se muestra al robot con un conjunto de coordenadas que se consideraron fijas en el robot y se utilizan para calcular la Fuerza neta que ejercen los objetos sobre el mismo una vez que se encuentra inmerso en el entorno de funcionamiento. Con eso se logra obtener, mediante un diagrama de cuerpo libre, planteando la suma de fuerzas y despejando las componentes en los ejes x e y, las expresiones generales para las componentes x e y de la fuerza neta que son las que se utilizan en el robot para calcular la dirección y velocidad de movimiento.

En la figura 4.4b se muestra el diagrama de cuerpo libre realizado. Tanto para la construcción del diagrama de cuerpo libre como para la construcción de las ecuaciones, se consideró que los objetos realizaban sobre el robot un campo repulsor y por lo tanto la fuerza que ejercen sobre él presenta una magnitud según la distancia a la que se encuentra el objeto y en sentido contrario a donde se ubica el objeto respecto al robot. De dicho diagrama, se desprenden las siguientes ecuaciones que dan origen a el modelo en base al cual se realizó la programación del software:

$$F_{neta}^{\vec{}} = \vec{F}_1 + \vec{F}_2 + \vec{F}_3 + \vec{F}_4 + \vec{F}_5 + \vec{F}_6 \quad (4.1)$$

$$\vec{F}_x = \sum_{i=1}^6 F_{ix}^{\vec{}} \text{ y } \vec{F}_y = \sum_{i=1}^6 F_{iy}^{\vec{}} \quad (4.2)$$

De estas ecuaciones podemos deducir entonces que:

$$\vec{F}_x = (\vec{F}_3 - \vec{F}_2 + \vec{F}_5 - \vec{F}_6) \cos(60) + \vec{F}_4 - \vec{F}_1 \quad (4.3)$$

$$\vec{F}_y = (\vec{F}_3 - \vec{F}_5 + \vec{F}_2 - \vec{F}_6) \sin(60) \quad (4.4)$$

Una vez obtenida la información de sensado, el módulo calcula la dirección y velocidad en la que se debe mover el robot en función de las componentes en x e y de la fuerza neta (Ecuaciones 4.3 y 4.4) ejercida sobre el robot que es determinada según los sensores que están detectando un objeto. Además, en función de que objetos detecta, el módulo arma un vector, donde la componente i del vector indica si el sensor i se encuentra detectando un objeto. Este vector se lo envía al módulo action para que éste encienda los leds correspondientes de la matriz.

A modo de resumen, es posible representar el funcionamiento del módulo control con el siguiente pseudocódigo:

```

s[6] /* vector de 6 lugares para          */
      /* guardar la inforamción de sensado */
v[6] /* vector que se construye para      */
      /* enviarle a el módulo action     */

while (true):

    for i = 1 to 6
        s[i] = obtener_info_sensor(i)
    end /* for */

    Fx = calcular_componente_en_x(s)
    Fy = calcular_componente_en_y(s)

    for i = 1 to 6
        v[i] = observo_objeto(s[i])
    end /* for */

    moduloAction.set_components(Fx, Fy)
    moduloAction.set_leds(v)
end /* while */

```

El prototipo construido utiliza sensores Sharp, los cuales tienen la particularidad de que el valor de retorno es mayor cuando la distancia del sensor al objeto es menor. Utilizando esta particularidad, se decidió interpretar la magnitud medida por el sensor como la magnitud de la fuerza que realiza el objeto detectado por el sensor. De esta manera, las ecuaciones 5.3 y 5.4 se transforman en las ecuaciones 5.5 y 5.6 respectivamente, donde S_i representa la magnitud obtenida por el sensor i .

$$\vec{F}_x = (S_3 - S_2 + S_5 - S_6) \cos(60) + S_4 - S_1 \quad (4.5)$$

$$\vec{F}_y = (S_3 - S_5 + S_2 - S_6) \sin(60) \quad (4.6)$$

Módulo action

Este módulo tiene 2 tareas importantes:

- Indicarle al módulo led la información que debe comunicar a la matriz para encender los leds correspondientes al lado del hexágono que representa a cada sensor.
- A partir de las componentes en x e y provenientes del módulo de control, generar las velocidades correspondientes a cada motor para lograr la velocidad y dirección deseados.

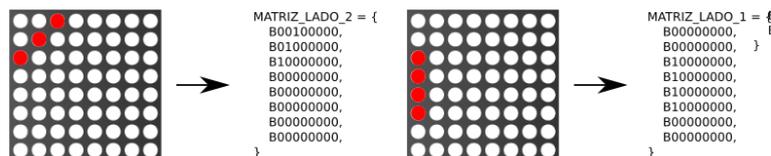


Figura 4.5: Construcción de las constantes del módulo action en función de los leds que deseamos prender de la matriz.

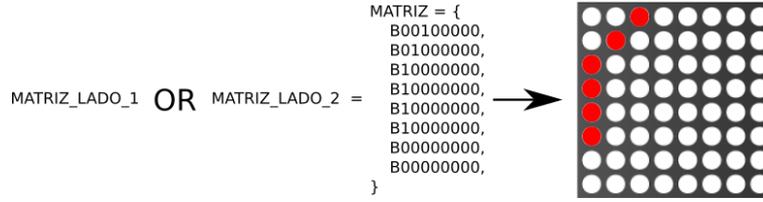


Figura 4.6: Construcción de la matriz de configuración para indicarle al microchip MAX7219 los leds que debe encender. Matriz lado 1 y Matriz lado 2, son las mostradas en la figura 4.5

Para indicarle al módulo led cuales son los leds que debe encender, el módulo action representa la configuración de estado de la matriz led como una matriz de 8 x 8 celdas, donde cada celda toma como valor 0 o 1 y representan el estado del led que se encuentra en esa posición de la matriz (1 prendido y 0 apagado). Además, posee 6 constantes correspondientes a cada lado del hexágono, donde cada constante es una matriz de 8 x 8 y las celdas que representan leds prendidos para dicho lado contienen el valor 1 y el resto 0 (Ver figura 4.5). Luego, para indicar que es necesario encender más de un lado a la vez, el módulo construye la matriz que se le debe enviar al microchip como el OR celda a celda de las constantes correspondientes a los lados encendidos (Ver figura 4.6). Una vez construida la configuración que debe tener la matriz led, esta es enviada mediante una interfaz serial al microchip MAX7219. Cuando el microchip recibe la configuración de bytes de la matriz desde el módulo action, se encarga de que la matriz de puntos muestre la configuración deseada prendiendo y apagando los leds correspondientes. Al implementar esta comunicación, se hace uso de la compatibilidad del firmware photon con el desarrollo de librerías para arduino y se utiliza una librería de arduino para controlar el MAX7219.

Para determinar las velocidades correspondientes a cada motor y así lograr el movimiento deseado, en primer lugar se decidió calcular velocidad del robot en x e y se de manera proporcional a la fuerza que ejercen los objetos sobre el robot. Esto nos lleva a construir las siguientes ecuaciones :

$$\vec{V}_x = K_x * \vec{F}_x \quad (4.7)$$

$$\vec{V}_y = K_y * \vec{F}_y \quad (4.8)$$

donde K_x y K_y son constantes que se determinan empíricamente, en nuestro caso, el valor definido para estas constantes fue 1. Entonces componiendo las ecuaciones 5.7 y 5.8 con las ecuaciones 5.5 y 5.6 respectivamente, tenemos que:

$$\vec{V}_x = K_x * ((S_3 - S_2 + S_5 - S_6) \cos(60) + S_4 - S_1) \quad (4.9)$$

$$\vec{V}_y = K_y * ((S_3 - S_5 + S_2 - S_6) \sin(60)) \quad (4.10)$$

Sabiendo que el rango válido devuelto por un sensor se encuentra en el intervalo $[S_{MIN}, S_{MAX}]$, podemos deducir que:

$$VM\vec{a}_{x_x} = K_x * ((S_{MAX} - S_{MIN} + S_{MAX} - S_{MIN}) \cos(60) + S_{MAX} - S_{MIN})$$

$$\Rightarrow VM\vec{a}_{x_x} = (S_{MAX} - S_{MIN}) * (2K_x * \cos(60) + 1)$$

$$\Rightarrow VM\vec{a}_{x_x} = (S_{MAX} - S_{MIN}) * K_x + 1$$

$$\Rightarrow V\vec{M}ax_x = (S_{MAX} - S_{MIN}) * C_x \text{ con } C_x = K_x + 1 \quad (4.11)$$

Análogamente, podemos ver que:

$$V\vec{M}in_x = (-1) * (S_{MAX} - S_{MIN}) * C_x \quad (4.12)$$

Por lo que podemos concluir que la magnitud de V_x pertenece al intervalo $[0, VMax_x]$ y que V_x pertenece al intervalo $[-VMax_x, VMax_x]$. Análogamente podemos obtener que $VMax_y = \sqrt{3}(S_{MAX} - S_{MIN})K_y$ y V_y pertenece al intervalo $[-VMax_y, VMax_y]$.

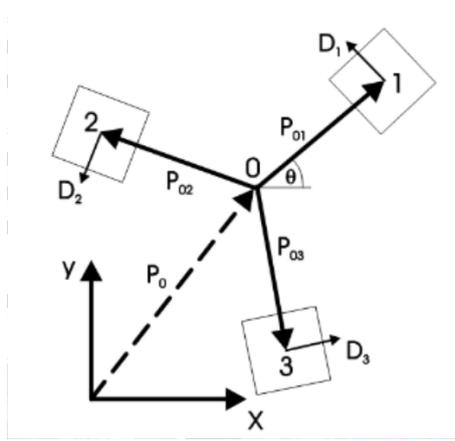


Figura 4.7: Diagrama de cinemática de un robot omnidireccional de 3 ruedas. Los vectores unitarios D_i determinan la dirección de avance de la *rueda* _{i} [8]

En la figura 4.7, podemos ver un diagrama de la cinemática de un robot omnidireccional de 3 ruedas, con las características del prototipo construido. De la cual podemos derivar las siguientes ecuaciones, como lo indica [8]:

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} = \frac{1}{r} * \begin{bmatrix} -\sin(\theta) & \cos(\theta) & L \\ -\sin(\frac{\pi}{3} - \theta) & -\cos(\frac{\pi}{3} - \theta) & L \\ \sin(\frac{\pi}{3} + \theta) & -\cos(\frac{\pi}{3} + \theta) & L \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (4.13)$$

Donde r es el radio de las ruedas, L es la distancia desde el centro del robot a las ruedas, y θ es el ángulo entre el eje X y la primera rueda.

En el caso del prototipo construido, al no tener posibilidad de llevar registro de odometría que nos permita obtener θ en función del tiempo, se determinó que el movimiento del robot se realizará siempre en función de un eje de coordenadas fijo en el robot como el que se muestra en la figura 4.4a. Por lo tanto, tenemos que en todo momento, $\theta = 0$. Lo que nos convierte la ecuación 5.13 en lo siguiente:

$$\begin{aligned} \Rightarrow \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} &= \frac{1}{r} * \begin{bmatrix} -\sin(0) & \cos(0) & L \\ -\sin(\frac{\pi}{3}) & -\cos(\frac{\pi}{3}) & L \\ \sin(\frac{\pi}{3}) & -\cos(\frac{\pi}{3}) & L \end{bmatrix} \begin{bmatrix} \dot{x} \\ -\dot{y} \\ \dot{\theta} \end{bmatrix} \\ &\Rightarrow \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} = \frac{1}{r} * \begin{bmatrix} 0 & -1 & L \\ \frac{-\sqrt{3}}{2} & \frac{1}{2} & L \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & L \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (4.14) \end{aligned}$$

De estas ecuaciones, es posible deducir una ecuación para cada ϕ_i en función de \dot{x} , \dot{y} , $\dot{\theta}$:

$$\phi_1 = -\dot{y} + L\dot{\theta} \quad (4.15)$$

$$\phi_2 = \frac{-\sqrt{3}}{2}\dot{x} + \frac{\dot{y}}{2} + L\dot{\theta} \quad (4.16)$$

$$\phi_3 = \frac{\sqrt{3}}{2}\dot{x} + \frac{\dot{y}}{2} + L\dot{\theta} \quad (4.17)$$

Una decisión importante que se tomó para desarrollar el prototipo y realizar los experimentos, fue acotarse a los casos donde el prototipo se desplaza sin girar sobre si mismo. Es decir, en los momentos que el prototipo esta detectando un objeto, el módulo action, recibirá las velocidades \dot{x} e \dot{y} correspondientes y $\dot{\theta}$ será 0. En el caso en que no esté detectando ningún objeto, el prototipo buscará a su alrededor par ver si encuentra algún objeto. En ese caso, las velocidades que recibirá en 0 serán \dot{x} e \dot{y} y $\dot{\theta}$ será un valor constante que se determinará empíricamente. Esta decisión nos permite trabajar con las ecuaciones 5.15, 5.16 y 5.17 de diferente forma según el caso en el que nos encontremos.

Para el caso en que no se detecta ningún objeto, las ecuaciones se transforman en:

$$\phi_i = L\dot{\theta} \quad (4.18)$$

Lo que nos indica que para una determinada velocidad $\dot{\theta}$, los motores deben girar todos con la misma velocidad, independientemente del sentido.

Luego, en el caso donde se detecta un objeto, las ecuaciones 5.15, 5.16 y 5.17 se transforman en:

$$\phi_1 = -\dot{y} \quad (4.19)$$

$$\phi_2 = \frac{-\sqrt{3}}{2}\dot{x} + \frac{\dot{y}}{2} \quad (4.20)$$

$$\phi_3 = \frac{\sqrt{3}}{2}\dot{x} + \frac{\dot{y}}{2} \quad (4.21)$$

Combinando estas ecuaciones con los valores de $VMax_y$ y $VMax_x$ tenemos que:

- ϕ_1 es máxima cuando $\dot{y} = -VMax_y$ y mínima cuando $\dot{y} = VMax_y$.
 $\phi_{1MAX} = VMax_y$ y $\phi_{1MIN} = -VMax_y$
- ϕ_2 es máxima cuando $\dot{y} = VMax_y$ y $\dot{x} = -VMax_x$. Y es mínima cuando $\dot{y} = -VMax_y$ y $\dot{x} = VMax_x$.
 $\phi_{2MAX} = (\frac{\sqrt{3}}{2}VMax_x + \frac{VMax_y}{2})$ y $\phi_{2MIN} = (-1)(\frac{\sqrt{3}}{2}VMax_x + \frac{VMax_y}{2})$
- ϕ_3 es máxima cuando $\dot{y} = VMax_y$ y $\dot{x} = VMax_x$. Y es mínima cuando $\dot{y} = -VMax_y$ y $\dot{x} = -VMax_x$.
 $\phi_{3MAX} = (\frac{\sqrt{3}}{2}VMax_x + \frac{VMax_y}{2})$ y $\phi_{3MIN} = (-1)(\frac{\sqrt{3}}{2}VMax_x + \frac{VMax_y}{2})$

Sustituyendo por las definiciones de $VMax_x$ y $VMax_y$, teniendo en cuneta que $K_y = 1$ y $K_x = 1$ y definiendo $\Delta S = (S_{MAX} - S_{MIN})$ tenemos:

$$\phi_1 \in [-\sqrt{3}\Delta S, \sqrt{3}\Delta S] \quad (4.22)$$

$$\phi_2 \in \left[-3\frac{\sqrt{3}}{2}\Delta S, 3\frac{\sqrt{3}}{2}\Delta S\right] \quad (4.23)$$

$$\phi_3 \in \left[-3\frac{\sqrt{3}}{2}\Delta S, 3\frac{\sqrt{3}}{2}\Delta S\right] \quad (4.24)$$

Como $\sqrt{3}\Delta S < 3\frac{\sqrt{3}}{2}\Delta S$, vemos:

$$\forall i = 1, 2, 3 \phi_i \in \left[-\frac{3\sqrt{3}}{2}\Delta S, \frac{3\sqrt{3}}{2}\Delta S\right], \text{ con } \Delta S = S_{MAX} - S_{MIN} \quad (4.25)$$

Como acabamos de demostrar, los motores recibirán velocidades que se encuentran en el intervalo mostrado en la ecuación 5.25. Sin embargo, la biblioteca con la cual se manejan los servos, tal y como lo indica su API en la documentación del firmware Photon [38] reciben un valor correspondiente a la potencia con la que se moverán los motores. Este valor, $Pot \in [-Pot_{Max}, Pot_{Max}]$.

Por este motivo, es necesario, una vez que el módulo de control calcula la velocidad con la que se tiene que mover el robot en x y en y se las envíe al módulo action y este módulo al recibir las velocidades en x e y, calcula la velocidad correspondiente a cada motor. Sin embargo, previo a indicarle a cada motor la potencia para que este se mueva, es necesario relacionar los valores de ϕ_i con la potencia que recibe cada motor. Esta relación se implemento normalizando el módulo de cada ϕ_i y multiplicandolo por Pot_{Max} . De esta manera, cuando el valor normalizado $\phi_{i_{Norm}}$ es 1 (correspondiente a $\phi_i = \frac{3\sqrt{3}}{2}\Delta S$) los motores reciben la potencia máxima, en caso de ser 0 recibe 0 y si la normalización se encuentra entre 0 y 1, se le envía la fracción correspondiente. Luego el sentido de giro se le indica con el signo de ϕ_i .

Capítulo 5

Pruebas de Sensores

En esta sección, se presentarán las pruebas realizadas para la selección de los sensores a utilizar en la construcción de prototipo. Se presentará una descripción detallada de las pruebas que se realizaron para obtener datos experimentales sobre los distintos sensores y luego se analizarán los resultados. Para este análisis se incluyeron aquellos que mostraban detalles interesantes y que aportaron a las decisiones tomadas. En el documento "Resultados experimentales", es posible encontrar los resultados de todos los experimentos realizados.

5.1. Descripción

Las pruebas de sensores realizadas, tuvieron los siguientes objetivos:

- Comprobar la reacción de los sensores frente a objetos de distintas características.
- Comprobar la reacción de los sensores frente a objetos colocados a distintas distancias y posiciones respecto al robot, en comparación con lo determinado en las distintas hojas de datos.
- Realizar un análisis de cuán sensibles son los sensores al ruido, y como mejora esta sensibilidad con la aplicación de filtros.
- Finalizar el análisis de resultados de estas pruebas con una proposición fundada empíricamente sobre qué sensor utilizar en futuros prototipos.

Las pruebas se ejecutaron dentro del laboratorio de robótica de Facultad de Ingeniería, en una mesa de trabajo donde se realizaron una serie de marcas, utilizando las herramientas de medición del laboratorio.

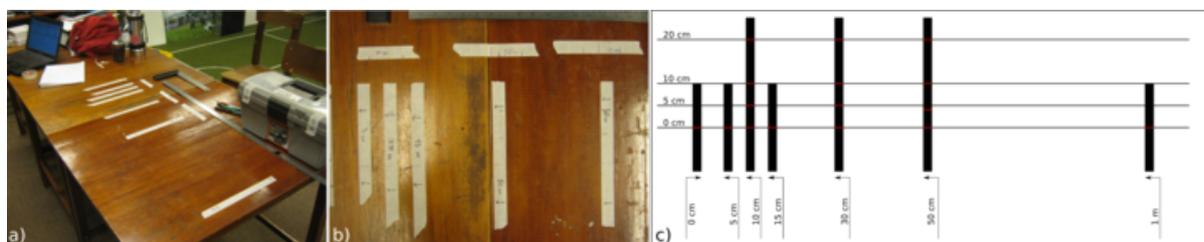


Figura 5.1: a) Entorno construido para la realización de experimentos. b) Marcas colocadas en el entorno. c) Esquema del entorno construido, donde se marcan las medidas alineadas y las marcas de corrimiento.

En la figura 5.1a y 5.1b se muestra el entorno construido. En ella se pueden observar las siguientes marcas: 0cm, 5 cm, 10 cm, 15 cm, 30 cm, 50 cm y 1 m. Para colocar estas marcas, en

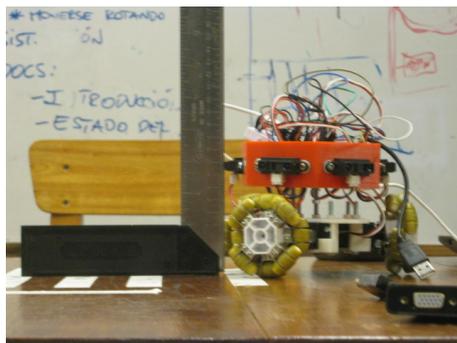


Figura 5.3: Alineación del robot con la marca 0.

primer lugar se determinó la posición de la marca 0cm, para luego medir las sucesivas distancias. El objetivo de colocar estas marcas es tener la certeza de que la medida que estamos queriendo medir desde el sensor al objeto colocado es la correcta. En la figura 5.1a, además, se pueden apreciar otras marcas ubicadas de manera que no se encuentran alineadas a la línea imaginaria que une las marcas antes mencionadas con la marca de origen donde se encuentra el sensor y que representa el punto 0 cm. Estas marcas no alineadas se encuentran a los 10 cm, 30 cm y 50 cm. Y se desvían de la línea imaginaria 5 cm, 10 cm y 20 cm. Un esquema del entorno construido se puede observar en la figura 5.1c.

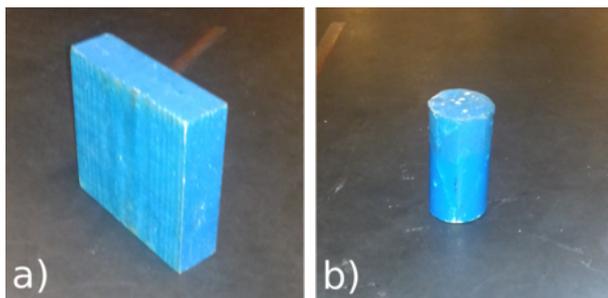


Figura 5.2: Objetos utilizados para las pruebas de sensores. a) Prisma - 5cm x 20cm x 20 cm. b) Cilindro - 15 cm de alto y 2.5 cm de radio

En ese escenario marcado, se trabajó con 2 objetos del mismo color, pero distinta forma. Uno es un cilindro de 15 cm de alto y 2.5 cm de radio. El otro es un prisma de 5cm x 20cm x 20 cm. Figuras 5.2a y 5.2b respectivamente.

Estos objetos se colocaron en todas las marcas y se tomaron 1000 muestras en cada marca. Para tomar las muestras, se colocó el robot en la marca 0, como se muestra en la figura 5.3. Se utilizó una escuadra para determinar que el sensor quedara efectivamente en el punto que representa los 0 cm. Luego se procedió a colocar el objeto centrado en la marca correspondiente a cada medida alineada, y para cada marca se tomaron las muestras.

La colocación de los objetos en las marcas desalineadas se realizó colocando el vértice más cercano al robot en la marca realizada. Para el caso del cilindro se colocó de manera tangencial la marca que representa la medida de alejamiento de la línea imaginaria y la que corresponde a la distancia de la marca. Ver figura 5.4.

Una vez tomadas todas las muestras para cada marca, se realizó una toma de muestras sin ningún objeto delante.

Las muestras tomadas, se procesaron en crudo, y aplicando filtros: un filtro pasabajo y un filtro de ventana con tamaño de ventana correspondiente a 3 y 10 medidas. A continuación se muestran las ecuaciones correspondientes a cada filtro.

$$\text{Pasabajo} : Out_{t_i} = Out_{t_{i-1}} + \alpha * Sensor_{t_i} - Out_{t_{i-1}} \text{ con } \alpha \in [0, 1] \quad (5.1)$$

$$\text{WindowN} : \frac{\sum_{i=1}^N Sensor_{t_i}}{N} \text{ con } N = \text{muestras consideradas} \quad (5.2)$$

Como es posible observar, el filtro pasabajo depende de un parámetro que al variar prioriza en mayor o menor medida la nueva muestra tomada. En nuestro caso la elección de 0.75 se debe a que es un valor comúnmente usado.

Una vez finalizada la toma de muestras y realizados los filtros, se graficaron y analizaron los resultados.

Es importante destacar que los sensores utilizados para estas pruebas fueron los siguientes: HC-SR04, Sharp-2Y0A21, Sharp-2Y0A02 y el VL53L0X.

La elección del sensor Sharp-2Y0A21 se debe a que presenta menor tamaño y mayor mayor rango de medida (en relación a los demás sharp). Posee un rango de medida que se ajusta mejor al requerimiento de trabajar en un espacio reducido y no es tan grande como para aumentar la cantidad de ruido que generan los objetos que se encuentran más allá de los límites del espacio de trabajo. Por último luego de realizadas las pruebas y analizados los resultados, las observaciones obtenidas para el sensor Sharp-2Y0A21 no permitieron obtener demasiadas conclusiones, por lo que se decidió repetir las pruebas para el sensor Sharp-2Y0A02.

La elección del sensor de tiempo de vuelo se debe a que es el sensor de menor tamaño que se encuentra en el laboratorio, y según la hoja de datos es el más preciso de todos los presentados hasta el momento. Además es un sensor que presenta un buen límite mínimo para detectar un objeto (3 cm) y en caso de necesitar que los sensores sean retirados hacia el interior del robot para lograr que la primer medida que el sensor detecte sea el límite exterior del robot, este sería el sensor que menos espacio necesitaría.



Figura 5.4: Colocación de objetos en la marcas desalineadas.

Se realizaron pruebas con el sensor HC-SR04 ya que es el sensor de ultrasonido que presenta un menor límite mínimo de medida ya que ambos sensores presentan un rango máximo excesivo para las distancias que se pretenden medir, lo cual puede introducir una mayor cantidad de ruido en las mediciones del sensor. Ver tabla 3.2

5.2. Análisis de resultados

En esta sección se presenta un análisis de los resultados obtenidos de las pruebas de sensores y se realiza una comparación para los distintos sensores en relación a las diferentes medidas que se tomaron.

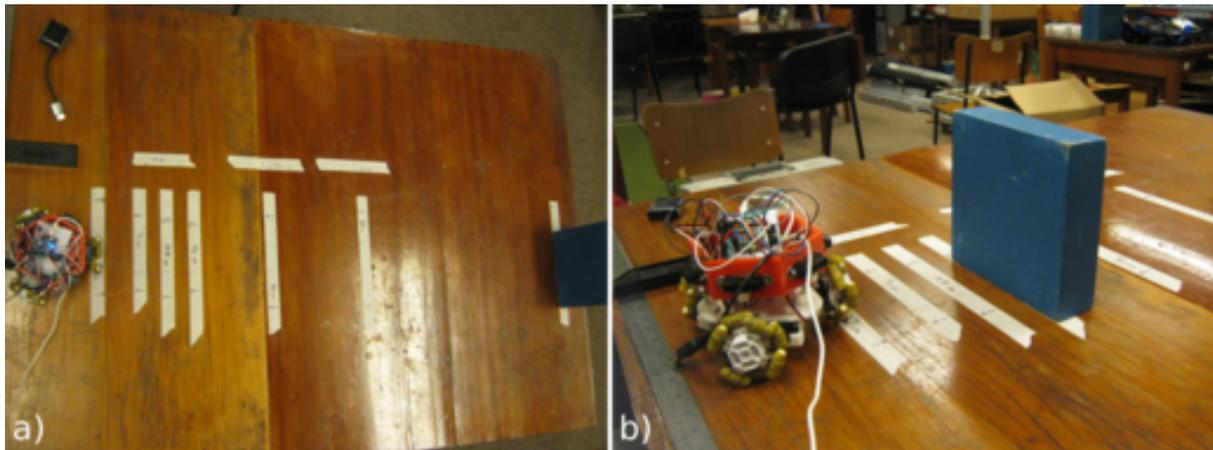


Figura 5.5: Momento en el que se tomaron medidas con el prisma alineado con el sensor. En la figura a, se observa la medición a 1 m, y en la b a 15 cm.

El análisis de las pruebas de sensores, se realizó de la misma manera tanto para el cilindro, como para el prisma. Para cada set de datos (en crudo y los distintos conjuntos con los filtros aplicados), se siguieron los siguientes pasos:

1. A cada conjunto de muestras se calculó la media, el máximo, el mínimo, y la diferencia entre el máximo valor de la muestra y la media, y la media y el mínimo valor de la muestra. Esto se realizó para cada marca desde donde se tomaron muestras.
2. Luego se construyó una tabla, donde se exponen los valores calculados y en base a ello, se construyeron 2 gráficas.
3. En ambas gráficas se colocó una función $x = 60$. Esta función es una referencia que representa el valor límite para el cual se espera poder obtener buenas medidas. Además se colocaron las rectas $y = media_{vacío}$, $y = maximo_{vacío}$ e $y = minimo_{vacío}$, que representan el valor de la media con el cual se identifica el vacío para el sensor, y el intervalo máximo y mínimo de las muestras tomadas para el vacío. Dentro de este intervalo, podría considerarse cualquier medida poco confiable, debido al ruido y a la confusión con considerar que no hay ningún objeto delante.
4. En una de las gráficas se colocaron los valores para las marcas que se encuentran alineadas frente al sensor (ver figura 5.5), donde además de la media, se indica la diferencia de la media con el máximo y con el mínimo. Esto se utiliza para observar que el intervalo donde se identifica una medida no comparta valores con el de otra medida y pueda llevar a confusiones.

5. En la segunda gráfica, se muestran las medidas de corrimiento, es decir, las medidas correspondientes a las marcas colocadas a X cm del sensor, pero corridas hacia un lado una cantidad Y de centímetros. Por lo tanto, se muestran las medidas para 10, 30 y 50 cm corridos 5, 10 y 20 cm. Con esto buscamos observar cuanto ruido generan los objetos que se colocan desalineados con el sensor. Particularmente, aquellos que no se encuentran frente al robot. Por ejemplo, los elementos que se corrieron unos 20 cm de la posición alineada.

Una vez que culminamos estos pasos, obtendremos para cada filtro, una imagen como la que se muestra en la figura 5.6

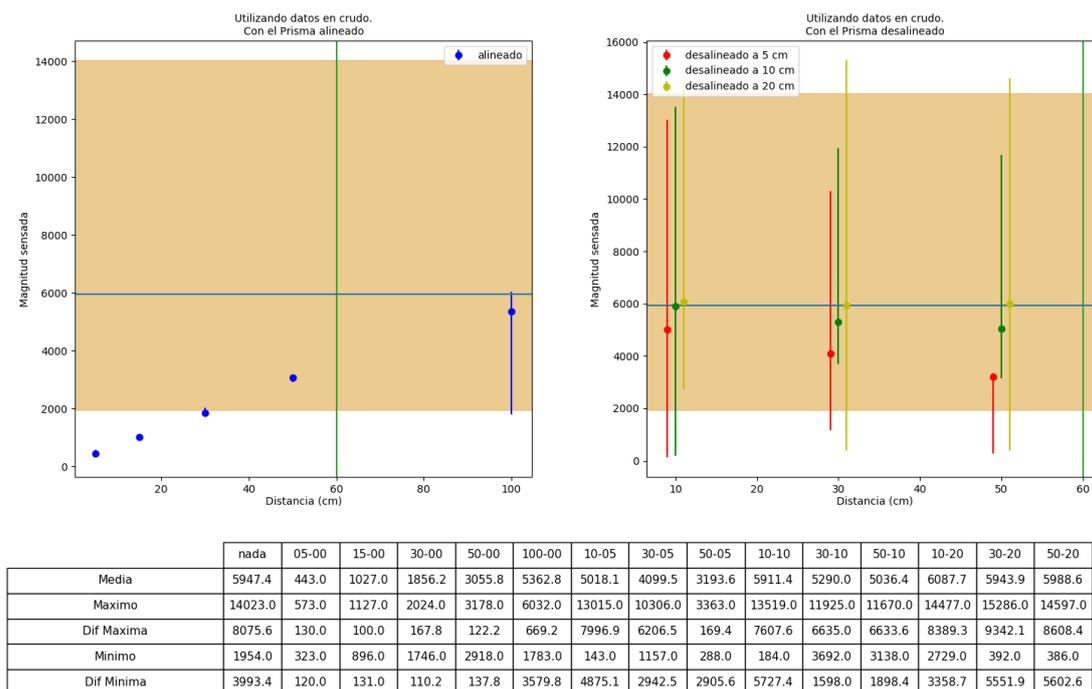


Figura 5.6: Datos en crudo del sensor HC-SR04 utilizando el prisma.

HC-SR04

La figura 5.6 muestra los resultados obtenidos luego de analizar los datos en crudo del sensor HC-SR04 utilizando el prisma. En base a ella es posible realizar las siguientes afirmaciones:

- En primer lugar, respecto a las medidas tomadas en las marcas alineadas, es posible observar que la intersección de los intervalos [Mínimo, Máximo] para cada marca es vacía. Esto nos permite diferenciar entre sí las medidas tomadas.
- A pesar de que las medidas tomadas es posible diferenciarlas, el intervalo [Mínimo, Máximo] obtenido de sensar sin ningún objeto delante, posee valores en común con los intervalos de las medidas mayores a 30cm.
- Otra observación es que en la gráfica b de la figura 5.6, es evidente que para las medidas donde se corrió el objeto hacia un lado, también se confunden con las medidas tomadas sin ningún objeto. Sin embargo, existen mediciones fuera del intervalo [Máximo, Mínimo] para el vacío, lo cual podría interpretarse como que los elementos no alineados con el sensor generan ruido en la mediciones y dificultan la identificación de objetos.

Estas 3 observaciones pueden realizarse de observar también la figura 5.7, donde se muestran los datos en crudo para el objeto cilindro. Sin embargo, es posible agregar que para el cilindro el intervalo [Mínimo, Máximo] para 30 cm se encuentra totalmente incluido en el intervalo [Mínimo, Máximo] de las medidas tomadas sin ningún objeto. Esto puede deberse a como interactúan físicamente las ondas que emite el sensor de ultrasonido, en una superficie convexa, lo cual genera mayor probabilidad de introducir ruido a la medición.

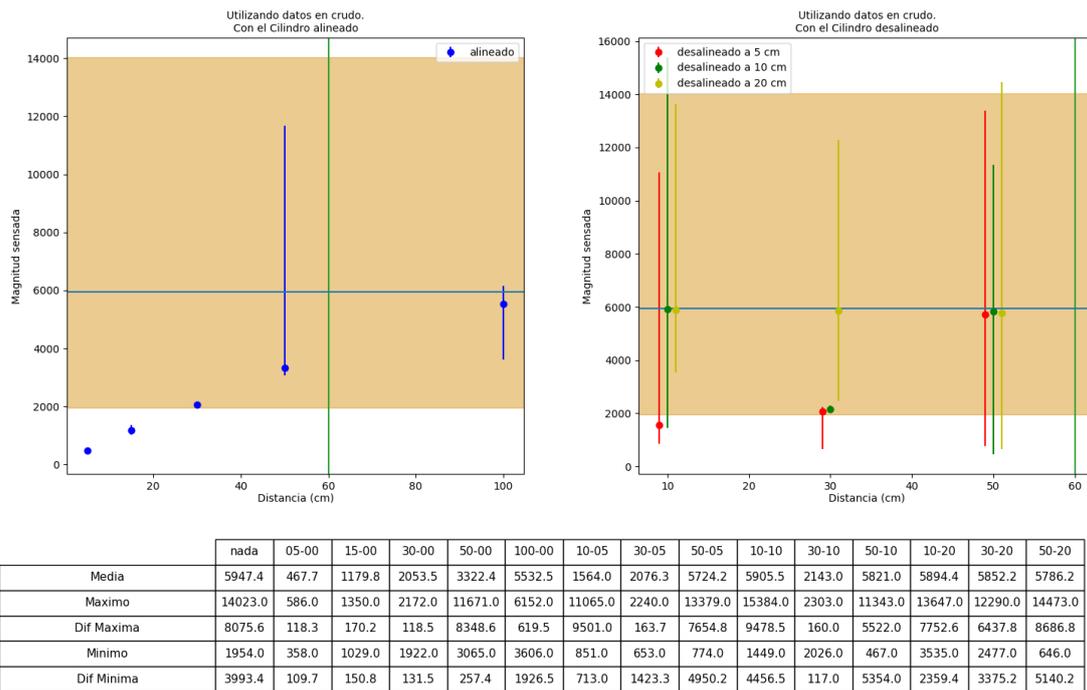


Figura 5.7: Datos en crudo del sensor HC-SR04 utilizando el cilindro.

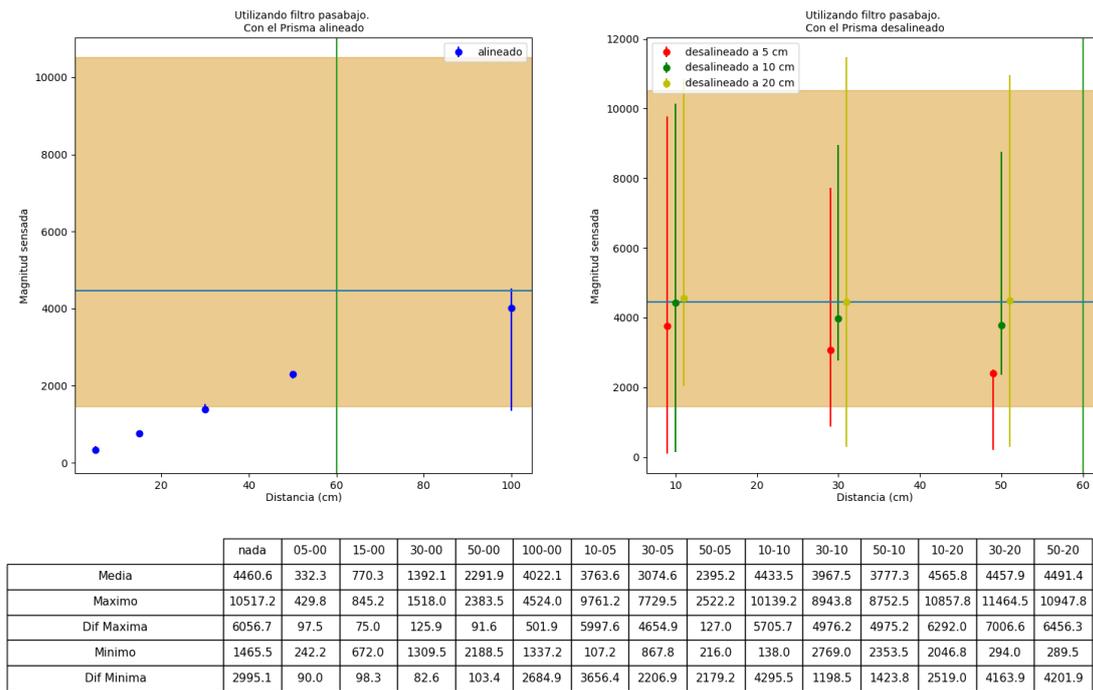


Figura 5.8: Datos tomados con el sensor HC-SR04 utilizando un filtro pasabajo y el objeto prisma.

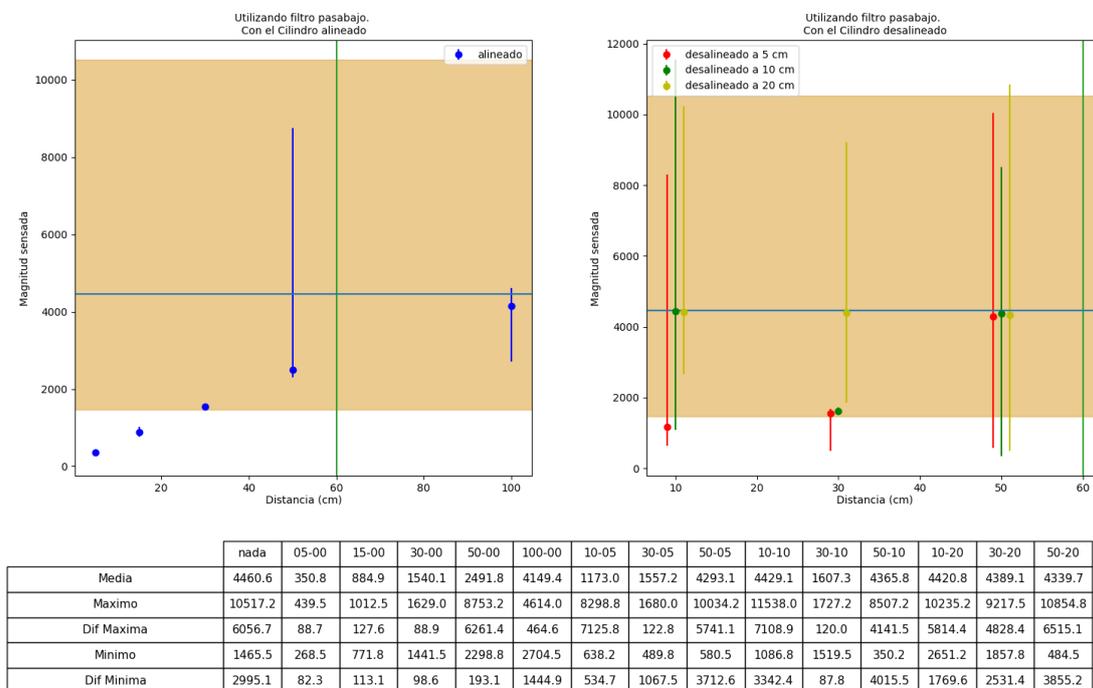
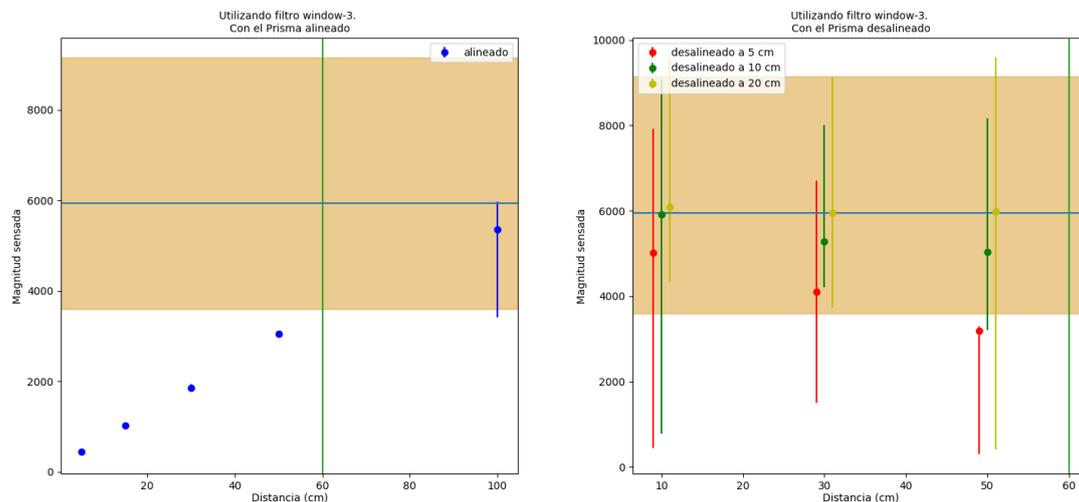


Figura 5.9: Datos tomados con el sensor HC-SR04 utilizando un filtro pasabajo y el objeto cilindro.

Las figura 5.8 y 5.9 corresponden a la aplicación de un filtro pasabajo a las medidas en crudo,

de donde no es posible obtener mejor información que con los datos en crudo. Por otro lado, al aplicar una ventana de tamaño 3, ver figuras 5.10 y 5.11, es posible observar que se mejora en la identificación de las distintas medidas. Por un lado se logra diferenciar las medidas tomadas para el prisma colocado a 50 cm y por otro, la mayoría de las veces, se logra lo mismo con un cilindro. Además, es posible observar que se comienzan a identificar objetos en las mediciones corridas, es decir que se comienza a lograr observar el cono de medición que presentan los sensores.

Al aplicar una ventana de mayor tamaño, las identificaciones mejoran aún más y las medidas menores o iguales a 50 cm se logran identificar completamente para ambos objetos. Ver figuras 5.12 y 5.13. Aumenta la cantidad de marcas corridas que logran identificarse. Y si bien las marcas corridas es posible diferenciarlas de la medida del vacío, no es posible saber si el objeto que se detectó se encuentra alineado al sensor o en un lugar del cono de medición ajeno a su bisectriz.



	nada	05-00	15-00	30-00	50-00	100-00	10-05	30-05	50-05	10-10	30-10	50-10	10-20	30-20	50-20
Media	5946.9	442.8	1026.7	1855.9	3055.5	5362.2	5017.3	4098.3	3193.3	5910.4	5289.3	5036.7	6086.8	5943.4	5988.1
Maximo	9156.0	496.0	1073.0	1931.0	3107.0	5973.0	7926.0	6704.0	3290.0	9072.0	8007.0	8169.0	9575.0	9106.0	9581.0
Dif Maxima	3209.1	53.2	46.3	75.1	51.5	610.8	2908.7	2605.7	96.7	3161.6	2717.7	3132.3	3488.2	3162.6	3592.9
Minimo	3591.0	397.0	976.0	1815.0	3002.0	3422.0	449.0	1499.0	312.0	784.0	4211.0	3211.0	4324.0	3734.0	413.0
Dif Minima	2355.9	45.8	50.7	40.9	53.5	1940.2	4568.3	2599.3	2881.3	5126.4	1078.3	1825.7	1762.8	2209.4	5575.1

Figura 5.10: Datos tomados con el sensor HC-SR04 utilizando un filtro de ventana con tamaño 3 y el objeto prisma.

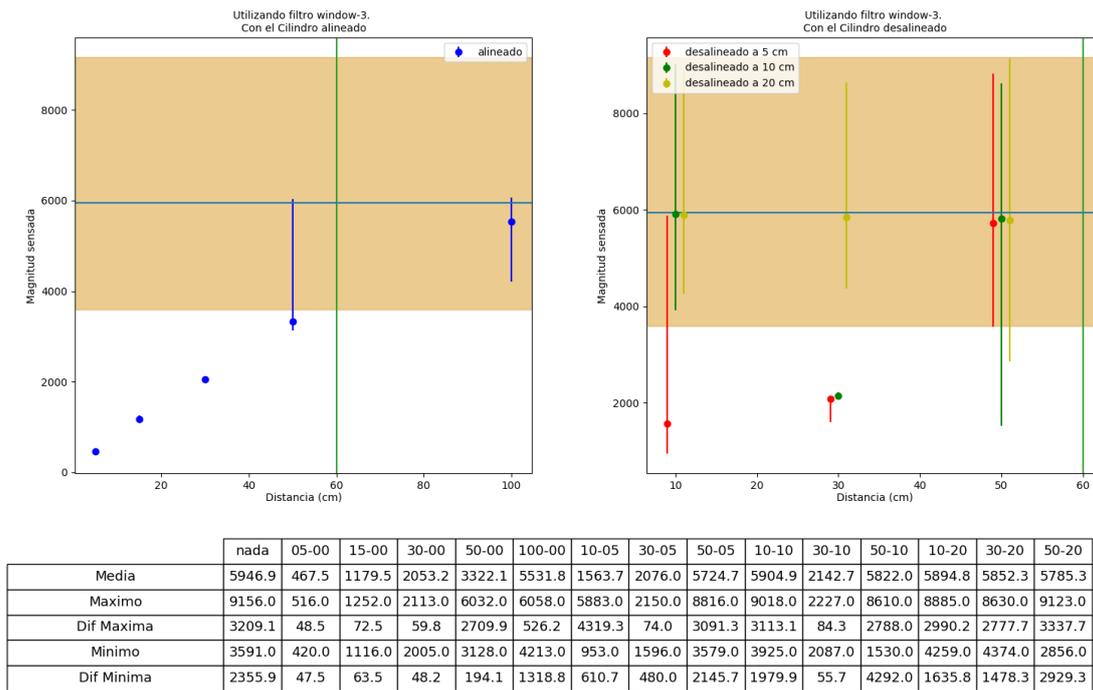


Figura 5.11: Datos tomados con el sensor HC-SR04 utilizando un filtro de ventana con tamaño 3 y el objeto cilindro.

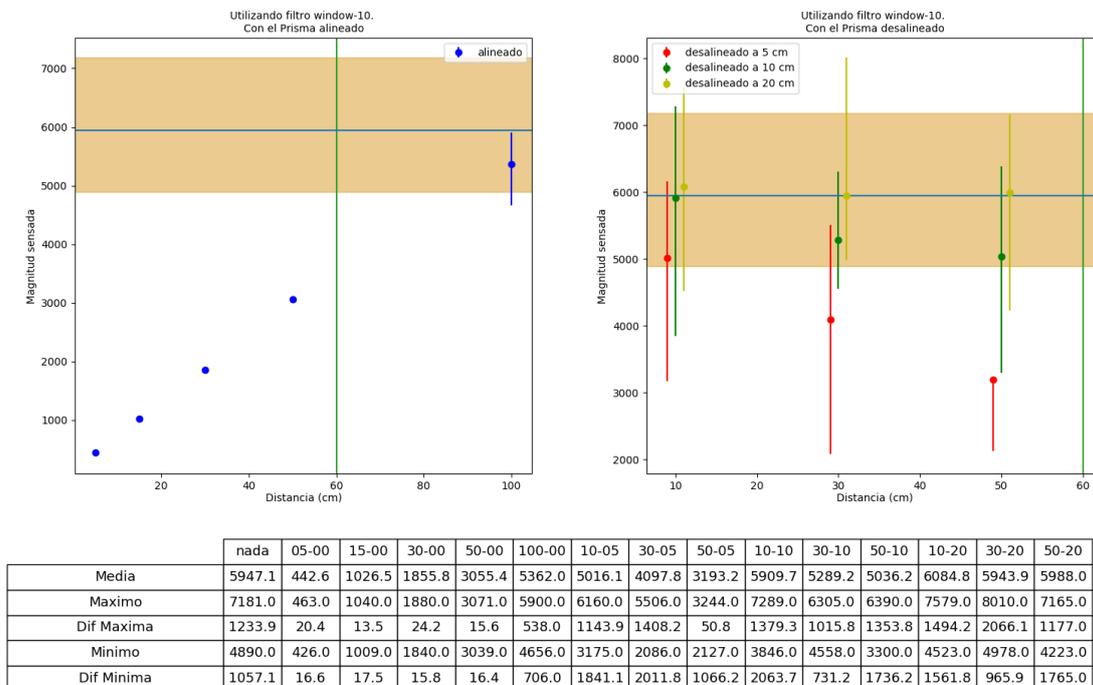


Figura 5.12: Datos tomados con el sensor HC-SR04 utilizando un filtro de ventana con tamaño 10 y el objeto prisma.

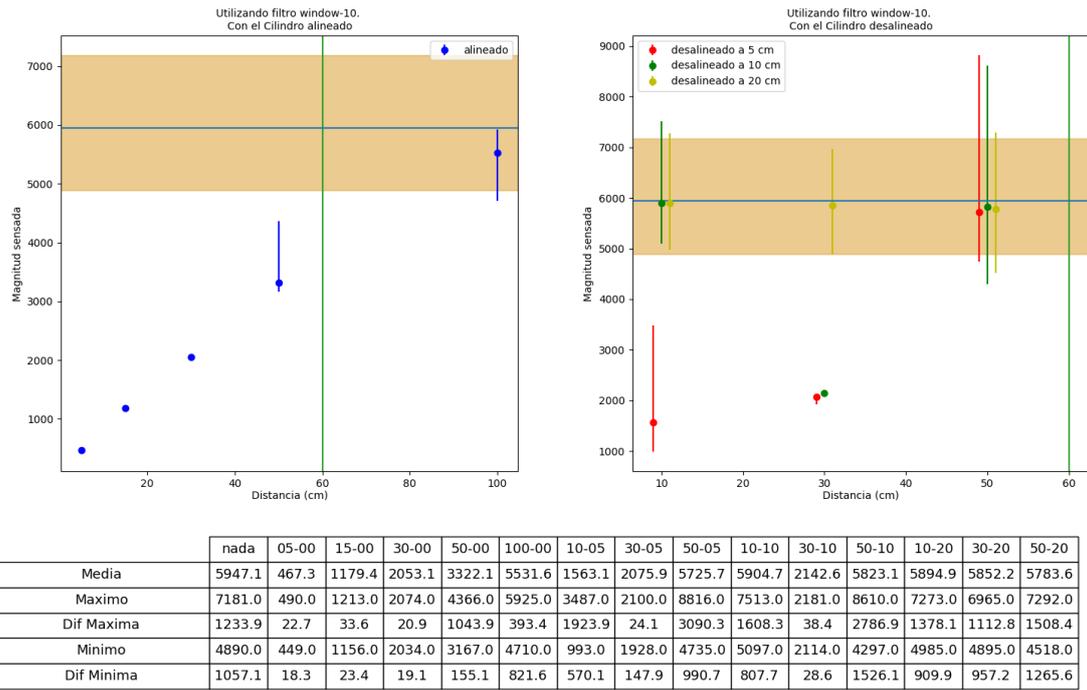


Figura 5.13: Datos tomados con el sensor HC-SR04 utilizando un filtro de ventana con tamaño 10 y el objeto cilindro.

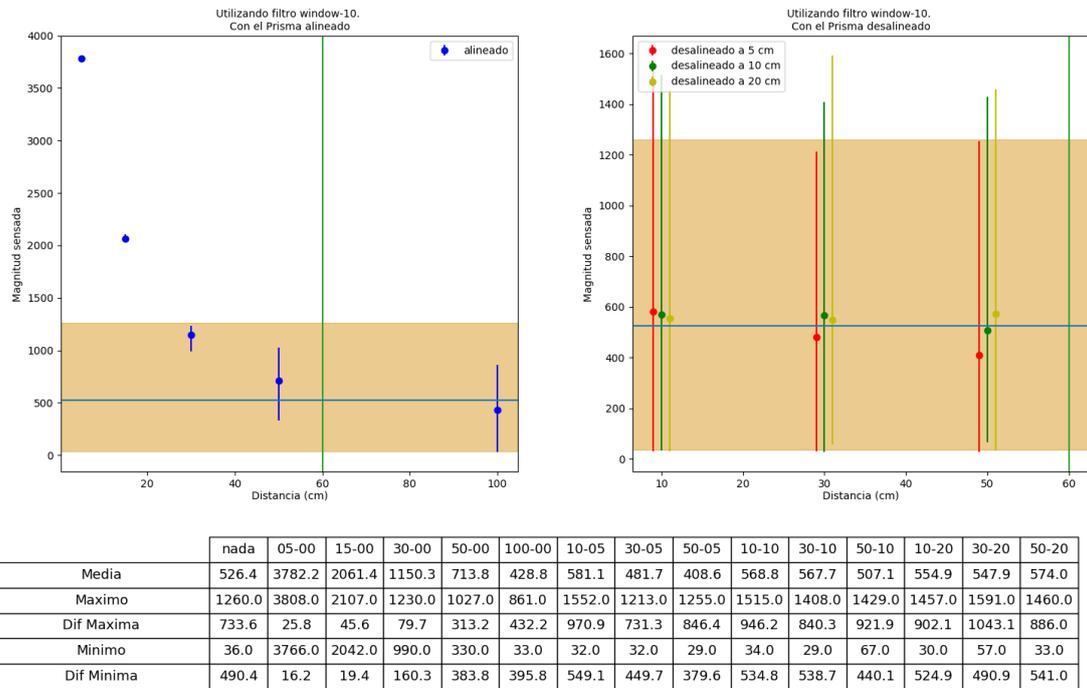


Figura 5.14: Datos tomados con el sensor Sharp-2Y0A21 utilizando un filtro de ventana con tamaño 10 y el objeto prisma.

Sharp-2Y0A21

De manera similar al análisis realizado para el HC-SR04, se analizaron los datos obtenidos para el sensor Sharp-2Y0A21. Sin embargo, para este sensor, del análisis de los resultados obtenidos, no fue posible llegar a conclusiones tan concretas. Esto se debe a que salvo para las mediciones de 5 cm y 15 cm sin ningún corrimiento, el resto de las medidas no fue posible diferenciarlas de las mediciones sin ningún objeto delante. Ver la figura 5.14. El resto de las gráficas analizadas para este sensor se encuentran en el documento "Resultados experimentales".

Sharp-2Y0A02

Para este sensor hay que tener presente que el rango de medición indicado en la tabla 3.2 era de 20 cm a 150 cm. Por lo tanto, además de realizar un análisis similar al de los otros sensores, se prestó especial atención al comportamiento de que presentó este sensor en valores menores a los 20 cm.

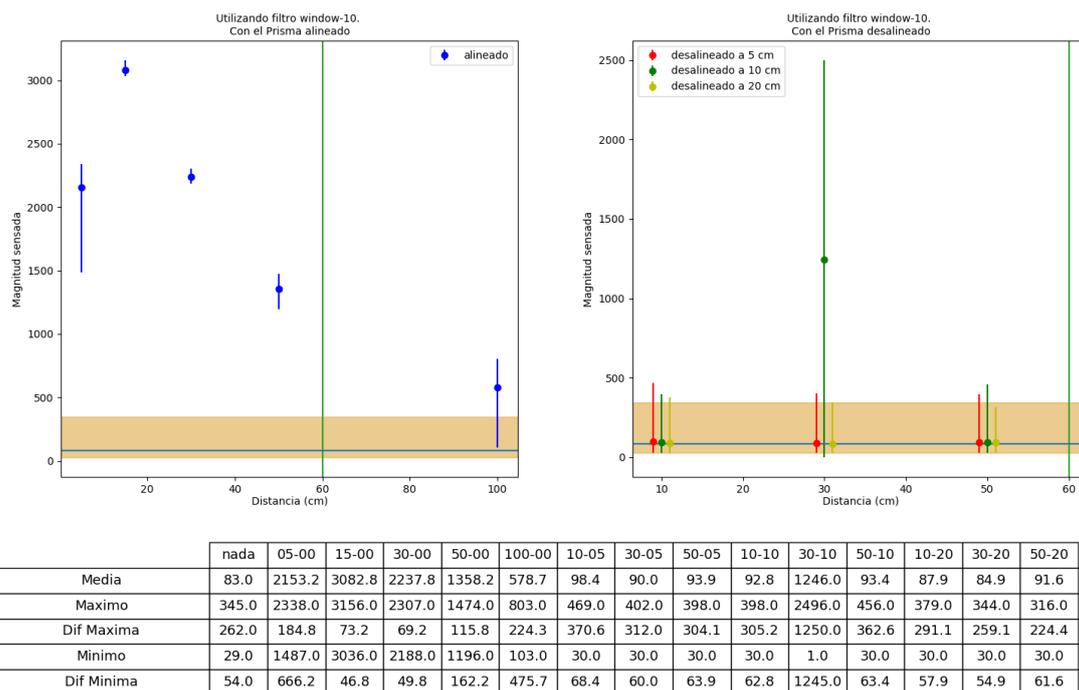


Figura 5.15: Datos tomados con el sensor Sharp-2Y0A02 utilizando un filtro de ventana con tamaño 10 y el objeto prisma.

Nuevamente las gráficas tanto para los datos en crudo, como para los datos con los filtros pasabajo y ventana de tamaño 3, se encuentra en el documento "Resultados experimentales". Aquí nos concentraremos en analizar las gráficas para el filtro de ventana con tamaño 10. Ya que en el análisis del sensor HC-SR04 fue el filtro que permitió diferenciar una mayor cantidad de medidas y en los datos analizados para el sensor Sharp-2Y0A02 es posible decir lo mismo.

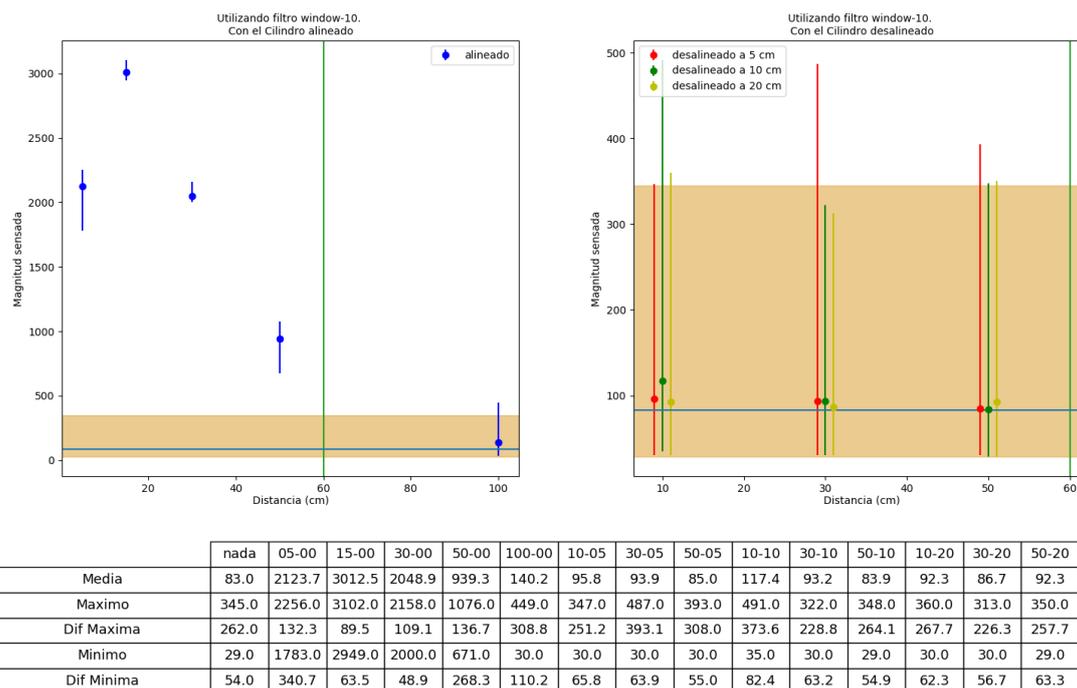


Figura 5.16: Datos tomados con el sensor Sharp-2Y0A02 utilizando un filtro de ventana con tamaño 10 y el objeto cilindro.

Como es posible observar en la figura 5.15, si no tomáramos en cuenta las medidas menores a 20 cm, es decir, si nos aseguráramos que los objetos se encuentran siempre a una distancia mayor o iguala 20 cm. Es posible identificar todas las medidas a objetos que se encuentran alineados al sensor, hasta alcanzar una distancia de 1m. Haciendo la misma suposición para el cilindro, es posible identificar todas las marcas alineadas, excepto la de 1m. Por lo tanto, para distancias mayores a 20 cm, y menores a 60 cm, podemos decir que el sensor se comporta igual tanto para el prisma como para el cilindro.

Considerando los objetos que se encuentran a una distancia menor a 20 cm, es posible observar como las medidas obtenidas por el sensor, presentan un intervalo que comparte valores con mediciones tomadas en marcas superiores a los 20 cm. Esto hace que sea posible confundir las medidas entre si. Sin embargo, al controlar el ambiente de pruebas, este sensor se transforma en una buena opción para la construcción del prototipo y la realización de las pruebas de los escenarios anteriormente descriptos.

Otro factor a tener en cuenta para utilizar este sensor en la construcción del prototipo es que las medidas con corrimiento afectan menos al sensor si lo comparamos con el HC-SR04.

VL53L0X

Analizando las muestras tomadas para este sensor, podemos observar como a pesar de los filtros aplicados, la variación para las medidas lineales menores o iguales a 50 cm es muy poca. Esto se debe a que el sensor es bastante robusto frente al ruido en las condiciones en las que se dio el experimento. Ver figuras 5.17 y 5.18.

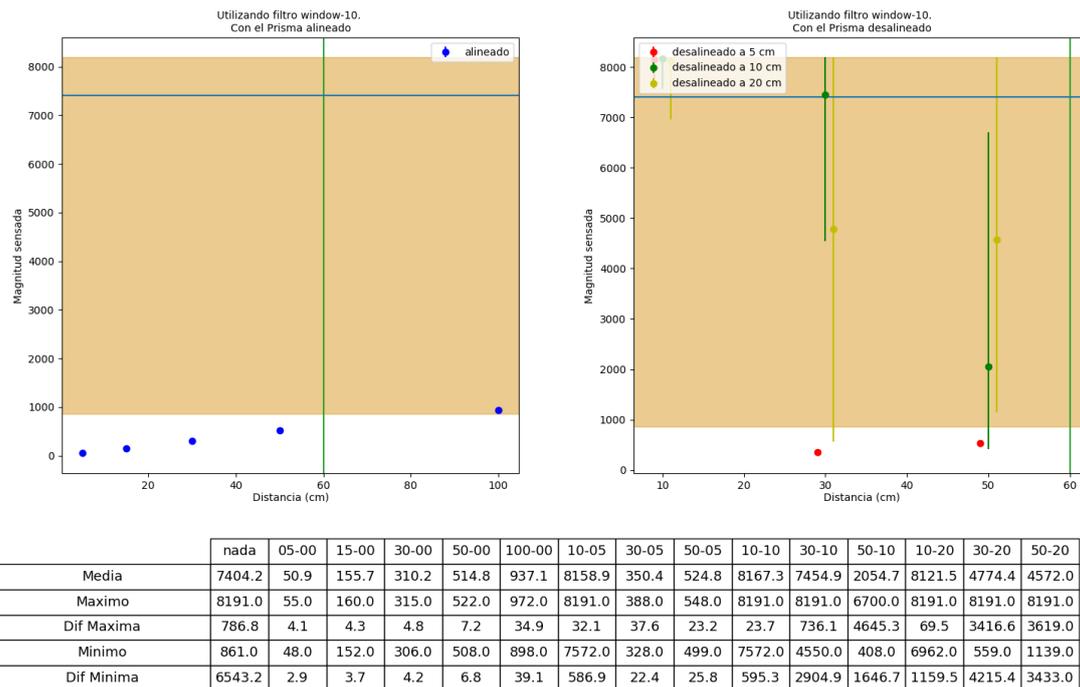


Figura 5.17: Datos tomados con el sensor VL53L0X utilizando un filtro de ventana con tamaño 10 y el objeto prisma.

Al igual que en los otros experimentos, en este es posible observar que el sensor logra identificar las medidas tomadas en las marcas con un corrimiento de 5 cm. Una particularidad de los datos obtenidos al realizar este análisis es que la diferencia entre los valores obtenidos para 30 cm sin corrimiento y con un corrimiento de 5 cm son muy similares en comparación con el espectro de medidas que devuelve el sensor. Mientras que el rango de valores que se obtienen para una medida alineada de 30cm es de [306 , 315] y el que toma con 5 cm de corrimiento es [328, 388], los valores que devuelve el sensor varían en [0, 8190]. Lo mismo sucede con los valores para 50 cm. A pesar de esto, es posible confundir algunas medidas, ya que la intersección de los intervalos para 50 cm sin corrimiento y 30 cm con 5 cm de corrimiento es no vacía.

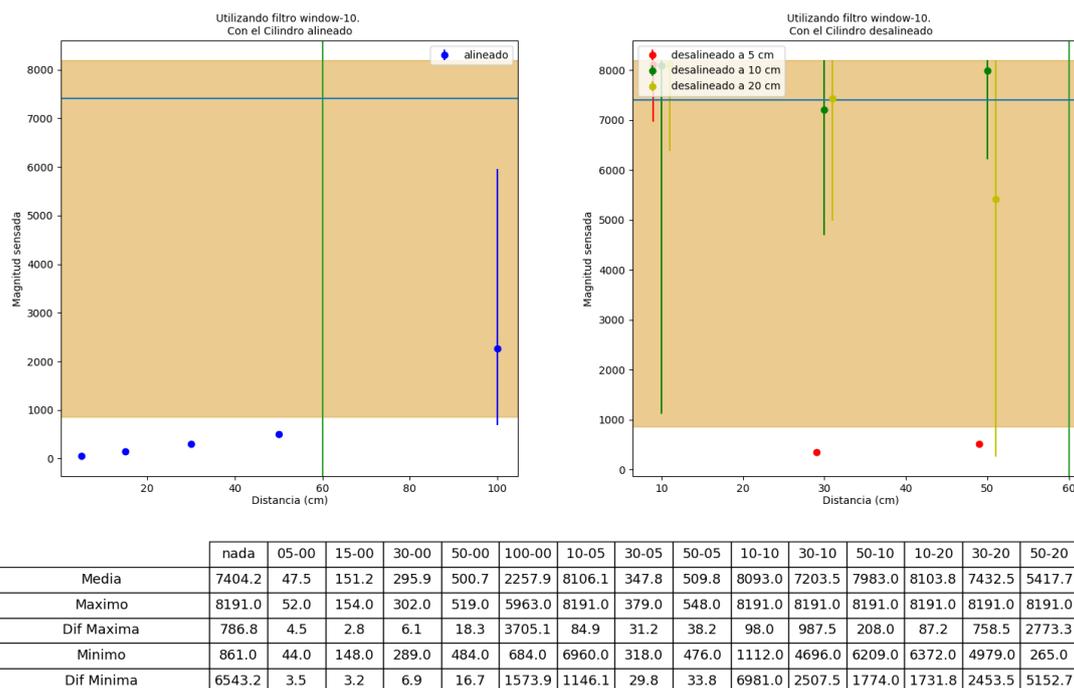


Figura 5.18: Datos tomados con el sensor VL53L0X utilizando un filtro de ventana con tamaño 10 y el objeto cilindro.

En comparación con el sensor sharp Shap-2Y0A02, este sensor presenta una ventaja y es que reconoce medidas menores a 20 cm.

Además, es necesario destacar que la implementación que se realizó para tomar las muestras fue con los valores por defecto que tiene el sensor en sus registros internos. Lo cual permite, según la hoja de datos, interpretar las mediciones del sensor como una medida en milímetros. Esto es una ventaja frente a los demás sensores, donde la medida en metros o centímetros se calcula haciendo una estimación en base a la forma en que el sonido se propaga por el ambiente, o lo mismo con el haz infrarrojo.

Teniendo esto en cuenta, es posible observar de la tabla de valores de la figura 5.17, como el error de las mediciones crece a medida que la distancia crece. Sin embargo hasta los 50 cm, el error varía de 2 a 6 mm, mostrando la gran precisión y exactitud que presenta el sensor, en el espectro de medidas con las que se pretende trabajar.

Capítulo 6

Pruebas de Prototipo

En este capítulo se detallarán los experimentos que se realizaron con el prototipo construido, principalmente relacionados al movimiento y la relación con los objetos de su entorno.

6.1. Descripción

Una vez realizadas las pruebas de sensores, y teniendo el prototipo construido, se realizaron las pruebas de prototipo. Estas tenían el objetivo de chequear la precisión y la correctitud con la cual se movía el robot, así como su grado de determinismo.

Para esto, se diseñaron algunos escenarios donde en función del comportamiento esperado del robot, se colocaron objetos para que al detectarlos, el robot cumpliera con determinados objetivos.

A continuación se describen los escenarios construidos y los objetivos que se pretendían alcanzar con cada escenario.

Escenario con objetos iguales

Este escenario tiene como objetivo probar los aspectos básicos del comportamiento evasivo del robot. Es decir, sabiendo que el comportamiento esperado del robot es que huya de los distintos objetos que sensa, se pretende colocar distintas configuraciones de objetos para que el robot reaccione frente a estos de la manera esperada. La cantidad de objetos con la que se configuró el entorno fue en aumento de uno a tres objetos.

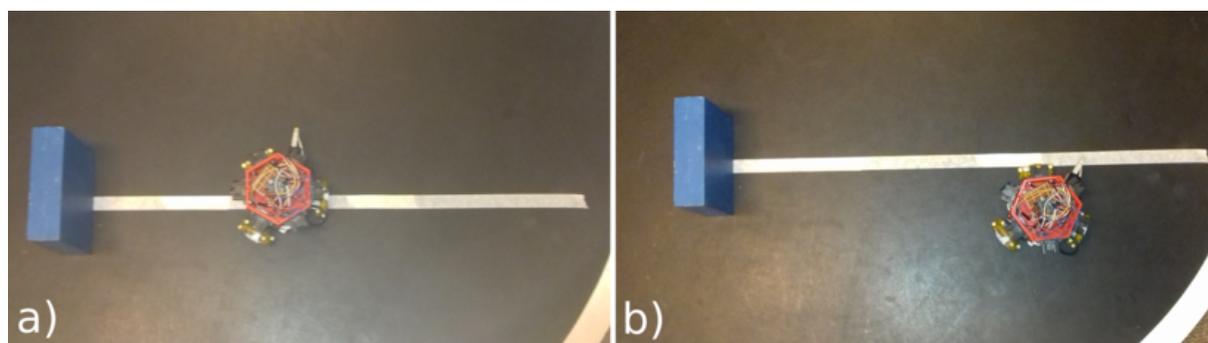


Figura 6.1: Escenario de prueba con un objeto. a) Estado inicial, donde el robot se coloca a 23 cm de el prisma. b) Estado final del escenario ejecutado.

Con un objeto, el entorno fue planteado como se muestra en la figura 6.1. Se colocó el robot sobre una línea de referencia que pasa por el centro del objeto y se esperaba como resultado que el robot se alejara del objeto permaneciendo sobre la línea de referencia. Sin embargo, el resultado obtenido fue que el robot si bien se alejó del objeto como se esperaba, también se desvió de la línea de referencia. Ver figura 6.1b

Con 2 objetos, se planteó como objetivo poder encadenar el comportamiento esperado para un objeto, y lograr que el robot reaccione de manera consecutiva a 2 objetos distintos. Para ello se planteó el escenario mostrado en la figura 6.2. El resultado esperado para este escenario es que el robot comience a moverse alejándose del primer obstáculo que detecta, y a medida que comienza a detectar el segundo modifique su recorrido, para escapar de ese objeto también. Por tanto, mientras ve el primer objeto, se aleja de este. Cuando comienza a detectar el segundo y no ha dejado de detectar el primero, el robot suma los vectores de fuerza que genera cada objeto y se mueve en función de ese resultado. Cuando dejó de detectar el primer objeto y solo detecta el segundo se moverá en consecuencia. Cuando deje de detectar objetos quedará girando en el lugar.

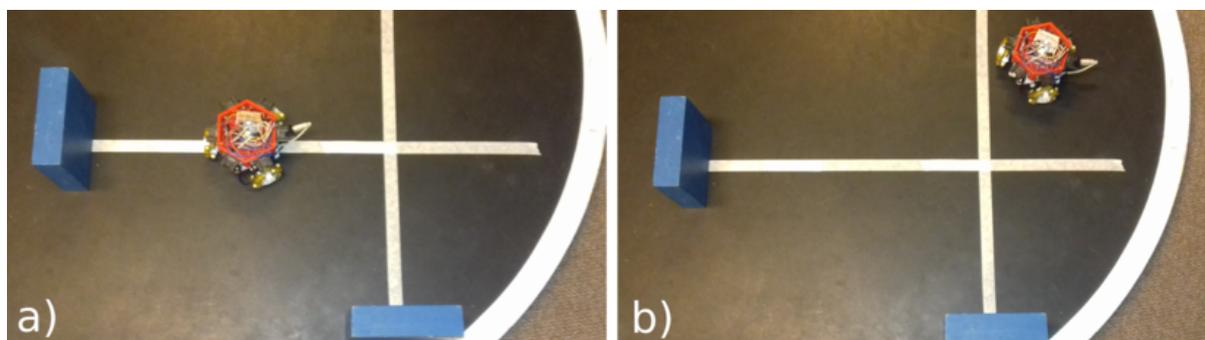


Figura 6.2: Escenario de prueba con 2 objetos. a) Estado inicial, donde el robot se coloca a 23 cm de el prisma. b) Estado final del escenario ejecutado.

El resultado obtenido luego de ejecutar este escenario de prueba fue que el robot si bien se alejaba de los obstáculos, siempre reaccionaba de una manera distinta. Pues por lo general, se desviaba del recorrido esperado para el primer objeto, provocando que en cada ejecución detectara de una forma diferente al segundo objeto.



Figura 6.3: Escenario de prueba con 4 objetos.

El escenario anterior se repitió con 4 objetos, y se pretendía que el robot alcanzara determinada zona, lo cual significa que cruzó la marca que se había realizado en el escenario. Ver figura 6.3. El robot alcanzó la zona de llegada, 6 de 10 veces que se ejecutaron los casos de prueba, pero no siempre lo hacía con el mismo recorrido.

Escenarios con objetos distintos



Figura 6.4: Escenario de prueba con 3 objetos distintos.

En este escenario, se pretendía evaluar la robustez del robot frente a las distintas formas que pueden presentar los objetos. Para ello, se configuró el entorno con 3 objetos de distinto tipo como se muestra en la figura 6.4. Es posible apreciar que si bien los objetos son distintos, la configuración es la misma que en el escenario con 3 objetos iguales. Esto se debe a que el resultado esperado es el mismo que para ese escenario. Se pretende que al colocar objetos distintos el robot presente un comportamiento similar al presentado con objetos iguales.

Si bien en este escenario se pretendía lograr obtener un resultado similar al escenario con 3 objetos iguales, al reaccionar siempre de manera distinta, se decidió separar este escenario en 3 escenarios uno con cada uno de los objetos distintos y compararlos con el escenario de 1 objeto solo.

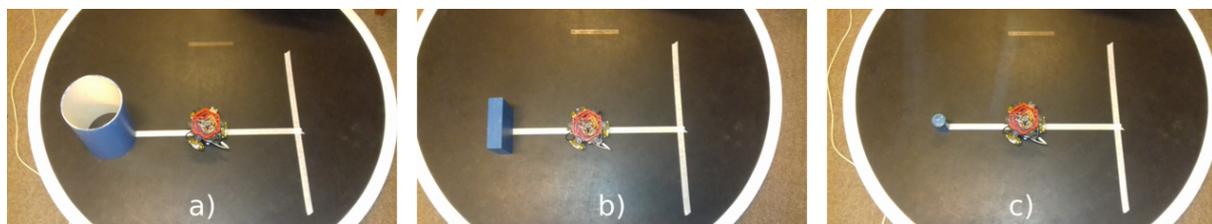


Figura 6.5: Escenario de prueba con 3 objetos distintos.

Los escenarios construidos se muestran en la figura 6.5. Luego de ejecutados los casos de prueba en reiteradas ocasiones, se comprobó que el comportamiento que presentaba el robot, era similar en todos los escenarios.

Escenarios donde se colocan y quitan objetos iguales

Este escenario se pretende evaluar como reacciona el robot a entornos con cierto grado de dinamismo.

Para implementar este escenario, se colocó al robot en una posición inicial, y se marcó una zona que se esperaba alcanzar, a un metro y medio de distancia. Luego se marcó un camino que se pretendía seguir con el robot (ver figura ??).

Para comenzar, luego de haber posicionado al robot, se colocó un objeto de manera que fuera detectado por el robot y este empezara a moverse. Una vez que el robot dejaba de detectar el

objeto o alcanzaba una posición deseada, se movía el objeto a un segundo lugar. Este proceso se repetía no más de 10 veces, hasta que el robot alcanzara la zona de llegada.

6.2. Análisis de resultados

En esta sección se analizarán los resultados obtenidos al poner en funcionamiento el robot en los escenarios descritos en la sección Pruebas de prototipo del capítulo anterior.

Escenario con objetos iguales

El resultado obtenido para este escenario no fue el esperado, ya que si bien el robot comenzaba a moverse en la dirección correcta, también se desvió de la línea de referencia. Esto podría deberse a la falta de una retroalimentación por parte de los motores, que le permita al software de control, realizar un seguimiento más estricto del movimiento que se encuentra realizando el robot. Así se podría lograr que el robot realizara movimientos en determinadas direcciones. Esto al día de hoy, con el prototipo construido es muy difícil de asegurar, ya que no es posible tener retroalimentación de los motores. Además, si bien los motores que se utilizan son servos, el control de estos se realiza mediante un parámetro que indica una potencia, y no es posible asegurar que mantengan una determinada velocidad de giro constante. Esto hace más difícil poder seguir una trayectoria determinada, como por ejemplo, en el caso de nuestro escenario, una línea recta.

Con 2 objetos, el resultado obtenido luego de la ejecución del escenario de prueba fue que el robot si bien se alejaba de los obstáculos, siempre reaccionaba de una manera distinta. Por lo general, se desviaba de una forma distinta del recorrido esperado para el primer objeto, provocando que en cada ejecución detectara de una forma diferente al segundo objeto.

Al repetir el escenario con 6 objetos, el robot alcanzó la zona de llegada, 6 de 10 veces que se ejecutaron los casos de prueba, sin embargo no siempre lo hacía con el mismo recorrido. Una observación que pudo realizarse durante el recorrido, fue que por momentos una de las ruedas del robot patinaba sobre la alfombra.

Escenarios con objetos distintos

En este escenario, no fue posible afirmar que se obtuvo el mismo resultado que en el escenario con objetos iguales debido a que el robot si bien era repelido por los objetos, siempre lo hacía de manera distinta.

Al separar el escenario en 3 escenarios distintos se comprobó que el comportamiento que presentaba el robot, era similar en todos los escenarios. Es decir, el robot siempre terminaba en una zona aproximada.

Por lo que podría decirse que el robot construido presenta cierto grado de robustez frente al cambio de forma en los elementos que lo rodean.

Escenarios donde se colocan y quitan objetos iguales

El resultado obtenido para esta actividad fue satisfactorio ya que en 9 de 10 oportunidades se logró que el robot alcanzara la zona marcada para la finalización. Si bien nuevamente el robot presentaba un desvío al ser repelido por los objetos, una vez que superaba un objeto, no volvía a detectarlo, ya que era quitado del escenario. El total de veces que se alcanzó el objetivo fue 9 oportunidades de 10.

Capítulo 7

Presentación a expertos

La presentación a expertos consistió en presentarle los avances del trabajo que se realizó a diferentes actores involucrados con el trabajo con niños. Este tipo de instancias se realizaron en 2 oportunidades y tenían como objetivo la validación de ideas para continuar con el desarrollo, así como obtener información acerca de las actividades que se podrían realizar con el robot y los niños. Esta información se utilizó para diseñar los escenarios donde se probó el robot, se concluyeron aspectos a mejorar y se obtuvieron lineamientos para un futuro desarrollo.

7.1. Descripción

A continuación se describen las 2 instancias donde se le presentó el prototipo a los expertos y se presentan los resultados obtenidos.

Primera instancia

La primera instancia de la presentación a expertos se realizó en la sala de reuniones 004 del Instituto de Computación de la Facultad de Ingeniería. Dicha sala cuenta con mesas de melamina, y buena iluminación. En dicha oportunidad, el objetivo principal era presentarle a una maestra de primer año de escuela, proveniente de la Escuela Simón Bolívar, el proyecto “Programando robots, jugando con el entorno” y validar ideas para continuar con el desarrollo del prototipo. Se realizó una demostración con el prototipo desarrollado hasta el momento, figura 7.1, y se le presentó el robot a un niño de 6 años, al cual se le prestó el robot para que lo controlara. En la reunión participaron 3 docentes de facultad de ingeniería, pertenecientes al grupo MINA, y una docente de facultad de psicología y su hijo de 6 años.

Como se puede observar en la figura 7.1 el prototipo que se tenía desarrollado hasta el momento poseía una menor cantidad de sensores que el prototipo final descrito en secciones anteriores. Además el software que tenía también era muy simple, reflejando una etapa temprana del desarrollo, donde el principal objetivo era validar la idea para definir si continuar trabajando en la misma línea.

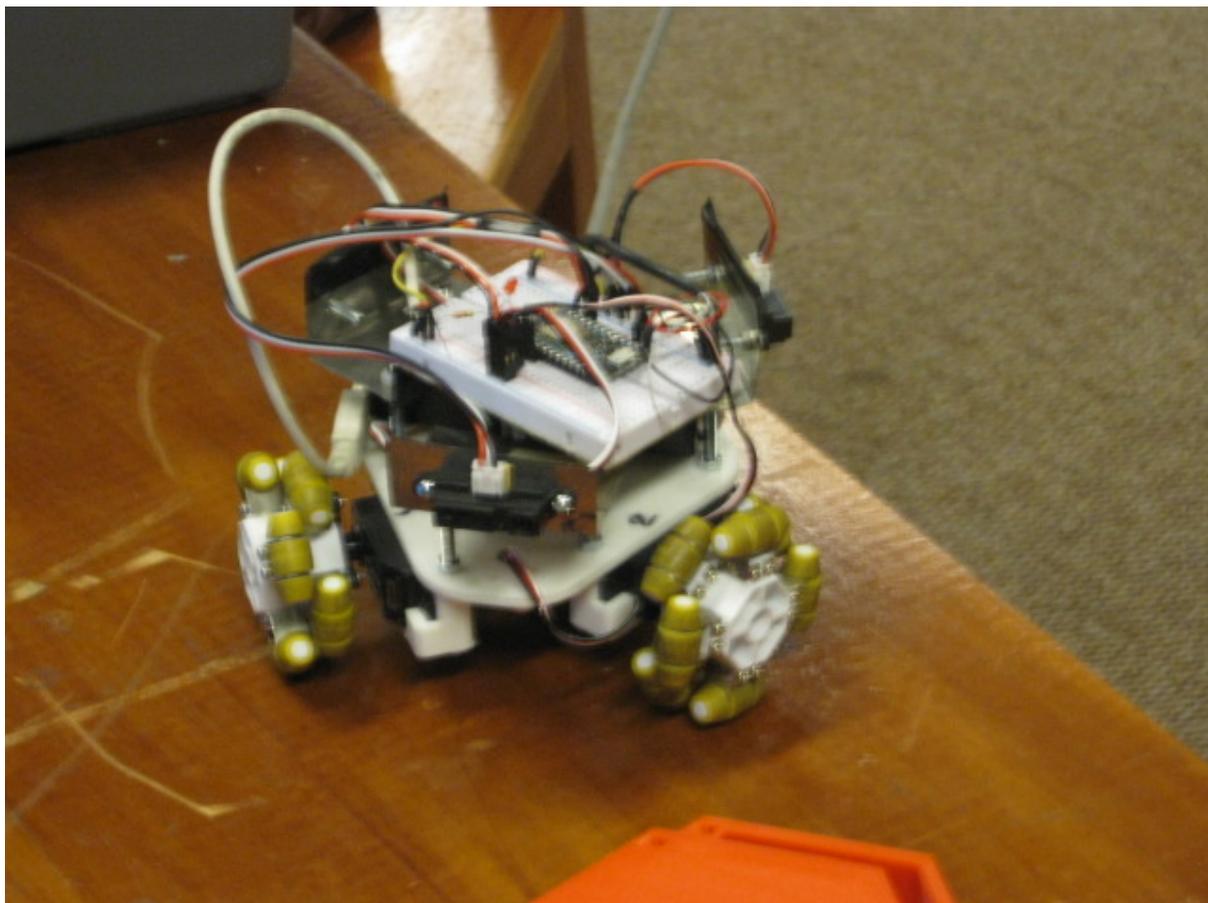


Figura 7.1: Prototipo 0, con solo 3 sensores y la protoboard.

En la figura 7.1 se observa la distribución de los sensores del prototipo presentado. Gracias a esta distribución, el software tenía la función de transformar la distancia medida por alguno de los sensores a una potencia que se le asociaba a los motores que se encuentran en los lados adyacentes. Luego, una vez que todos los sensores habían transformado la distancia en potencia, para cada motor, se sumaba todas la potencias asociadas y se le indicaba dicha potencia. En caso que alguna de ellas excediera la potencia máxima, se le indica al motor la potencia máxima. Es importante destacar que en esta oportunidad, el robot no tenía la totalidad de sus sensores provocando un alejamiento con los objetos que sensaba, sino que uno de ellos provocaba una atracción.

La demostración realizada consistió en colocar y quitar un objeto sobre la mesa para que el robot huyera de él y controlar el robot con las manos, a modo de lograr que el robot llegue de un extremo de la mesa al otro sin caerse.

Luego de realizada la demostración, la devolución de la muestra consistió en que con esa herramienta ella podría llegar a trabajar cualquier punto de su programa escolar.

Por ultimo, se le presentó el robot al niño que se encontraba en la sala, y este luego de explorarlo unos minutos, dedicó aproximadamente 15 minutos a jugar con el robot, mostrando un gran entusiasmo y logrando comprender en gran medida cómo se comportaba el robot.

Segunda instancia

La segunda instancia consistió en una reunión realizada en el Laboratorio de robótica de la Facultad de Ingeniería, donde se le presentó el prototipo final al conjunto de investigadores del proyecto PRJE. En esta oportunidad el objetivo de la actividad era evaluar si era posible realizar las actividades que hasta el momento se venían diseñando para realizar con los niños, teniendo en cuenta que el robot presentaba un comportamiento donde los objetos realizaban fuerzas repulsoras sobre el robot.

Una vez iniciada la actividad se presentó el prototipo, describiendo su comportamiento y se pasó a trabajar con las actividades diseñadas. La primera consistía en colocar un objeto y el robot apagado frente al mismo e intentar predecir hacia dónde se movería el robot. La segunda consistió en colocar el robot entre 2 objetos con el objetivo de que oscilara entre ellos, hasta converger a un punto. Por último, se seleccionó un punto de partida y una zona de llegada, y se intentó mediante una configuración previa por parte del equipo, que el robot se trasladará desde el punto inicial a la zona de llegada.

Los resultados obtenidos de esta instancia, fueron los siguientes:

- La primer actividad, era muy similar a la planteada en el escenario donde se colocaba un objeto. Además de presentar un comportamiento consistente con el obtenido al realizar el escenario, en la actividad planteada todos los presentes lograron deducir bien la dirección con que saldría moviéndose el robot. Pero ninguno predijo que el robot podría desviar su movimiento. La actividad se realizó 2 veces, presentando en la segunda una mayor desviación que en la primera.
- El resultado de la segunda actividad fue el esperado en la mayoría de las veces que se realizó la actividad, sin embargo en otras, cuando el robot se desviaba de la recta que conectaba a los 2 objetos, este se alejaba hasta ya no detectar ningún objeto y en ese instante comenzaba a girar en el lugar sin desplazarse.
- En la tercer actividad, si bien el robot llegó del punto de partida a la zona de llegada, este no presentó el comportamiento esperado. Al repetir el experimento, el robot nuevamente alcanzó el objetivo, pero no con el comportamiento esperado, y además el comportamiento que presentó fue distinto a la vez anterior. Esto se repitió 10 veces, donde en 7 alcanzó el objetivo, y en 3 no. No siempre que se alcanzaba el objetivo el comportamiento expuesto por el robot era el mismo.

Para finalizar la actividad, se configuró el software del robot para que los sensores sean atractores y se realizó un actividad, donde se colocaba un objeto y el robot debía verse atraído por el y dirigirse a su ubicación. Si bien el robot lograba aproximarse al objeto, la cantidad de veces que lo perdía en el camino eran considerables para la corta distancia a la que se encontraba el objeto. En aproximadamente 50 cm el robot se perdió de 3 a 6 veces en las distintas oportunidades.

7.2. Análisis de resultados

La presentación a expertos fueron las instancias donde se validó lo construido hasta el momento y se definieron líneas para la continuación del trabajo. En la primer instancia, el resultado se evaluó como positivo, ya que la maestra describió que podría trabajar cualquier tema del programa con el robot. Esta afirmación es de carácter muy general, y si bien no aporta a los objetivos del proyecto PRJE, para el presente trabajo significó una afirmación de que como herramienta el robot podría llegar a ser usado para trabajar con niños pequeños. Otra interpretación que se le dió a la afirmación realizada por la maestra es que tal generalidad podría

deberse a la necesidad de conocer aún mejor la herramienta que se le estaba presentado. Esto ultimo fue el motivo de que las actividades de la segunda instancia fueran pensadas de manera gradual a modo de ir incrementando la complejidad de las actividades. Y pudiendo avanzar en la comprensión del funcionamiento del robot, a medida que se iban completando las actividades.

Como parte de la primer instancia, observamos durante 15 minutos, como un niño de 6 años jugaba con el robot. Esto nuevamente generó la idea de que el robot podría llegar a ser usado para trabajar con niños pequeño. Sin embargo fue posible observar que el niño no comprendía cabalmente el funcionamiento del robot, ya que si bien utilizaba el sensor que oficiaba de atractor, para mover el robot de un lado a otro, al momento de evitar que el robot cayera al piso, el niño tendía a poner las manos frente a el. Es decir, ignoraba el conocimiento de que si ponía la mano frente al sensor el robot la iba a seguir. Si el niño efectivamente hubiera entendido el comportamiento del robot, debería haber quitado la mano frente al sensor que atraía al robot, y se debería haber acercado por un punto ciego.

En la segunda instancia de presentación, el resultado de las actividades se considera positivo, pues se logro confirmar algunas conclusiones que se habían obtenido de ejecutar los escenarios de las pruebas del prototipo. Y también se plantearon mejoras por parte de los integrantes del equipo. Se reafirmó la necesidad de poder controlar mejor el movimiento para lograr que el robot sea lo más determinista posible, es decir que reaccione frente a un mismo escenario de la misma manera. Incluso respetando la simetría del robot.

Otra observación que se realizó en esta instancia, fue que el espacio necesario para trabajar con el prototipo construido era aproximadamente de 9 metros cuadrados, espacio demasiado grande si se tiene en cuenta el tamaño de los salones de clase donde conviven aproximadamente 35 niños, con bancos y mesas entre otras cosas.

Previo a la finalización, se configuraron los sensores a modo de prueba de concepto para que los objetos en lugar de ejercer fuerzas repulsoras, ejercieran fueras atractoras. Esto se probó con un solo objeto, y si bien presento el comportamiento esperado de que el robot tendiera a acercarse al objeto que detectaba, lo hacía con mucha dificultad debido a que en un tramo menor a 50 cm, el robot llegaba a perder el objeto hasta 5 veces.

Para finalizar la reunión se recabaron algunas características que los presentes entendían importantes a tener en cuenta durante el desarrollo de próximos prototipos. Ellas fueron:

- Contar con otros comportamientos como atractores, y una combinación de atractores y repulsores.
- Tener la capacidad de expresar otras situaciones además de lo que detectan los sensores de distancia, por ejemplo colocando un sensor de color y mostrando luces acorde al color que se encuentra viendo.
- Se planteo la necesidad de que el robot pudiera controlarse a control remoto.

Capítulo 8

Conclusiones y trabajo a futuro

En este capítulo presentaremos las conclusiones obtenidas a partir del trabajo realizado, y finalizaremos el presente informe, con una sección donde detallamos las propuestas de trabajo a futuro que se tratarán en las reuniones de trabajo del proyecto PRJE.

En relación a los objetivos principales, planteados en el capítulo de introducción, es posible afirmar que estos fueron cumplidos al finalizar el trabajo. Se evaluaron distintas tecnologías permitiendo determinar que sensores se usarían para la implementación del prototipo, y cuales serían considerados como parte del trabajo a futuro, al momento de comenzar la construcción de un próximo prototipo. También, a partir del trabajo con los expertos fue posible validar la idea de que es posible realizar actividades que aporten a los objetivos del proyecto PRJE con un robot con las características del prototipo construido. En ese sentido, es necesario continuar trabajando en futuros prototipos, para realizar mejoras en relación al hardware, y además, continuar con el desarrollo del software para mejorar su mantenibilidad y que permita fácilmente agregar nuevos comportamientos.

Particularmente se pretende mejorar la eficiencia del software al incorporar sensores VL53L0X. Estos sensores pueden ser configurados para que al momento de tener una nueva medida, generen una interrupción, lo que permite que el procesador del microcontrolador evite estar continuamente consultando a los sensores por una nueva medida.

De la investigación previamente realizada a la construcción del prototipo, se desprende la necesidad de realizar pruebas con la placa Esp 32 thing. Pues, como ya se expresó es la placa que presenta una mayor cantidad de pines analógicos a disposición del desarrollador. Esta necesidad se ve reflejada en el desarrollo actual del prototipo, ya que la Photon particle no posee la cantidad suficiente para agregarle por ejemplo el sensor de color como pidieron el resto de los integrantes del equipo. Otras ventajas que presenta esta placa son dimensiones pequeñas en comparación con el resto las placas estudiadas y también en comparación con el resto es una placa de bajo costo.

Con las pruebas de los sensores de distancia, podemos concluir que solo 2 sensores presentaron un comportamiento que se puede considerar adecuado para el trabajo que pretendemos realizar. Estos fueron el Sharp-2Y0A21 y el VL53L0X53. El primero, por ser el sensor que se encontraba disponible en el laboratorio se utilizó para construir el prototipo y realizar las pruebas de comportamiento en los distintos escenarios. Por la forma en que el sensor devuelve la información sensada al programador, la elección de este sensor fue buena, ya que facilitó la forma de relacionar la fuerza ejercida por los objetos al robot con la velocidad con que debía moverse el robot. Sin embargo, el sensor presentaba algunas desventajas frente al VL53L0X. La primera, era el tamaño, ya que el sensor Sharp-2Y0A21 duplicaba el del VL53L0X. La se-

gunda, es la precisión, el sensor VL53L0X como se vio en los experimentos realizados, es un sensor sumamente preciso con los parámetros por defecto, y esa medida se puede ajustar aún más modificando dichos parámetros. Además, como su hoja de datos lo indica, también se le puede ajustar la periodicidad con que toma las medidas, factor que en el Sharp-2Y0A21 es fijo. Por último, el sensor VL53L0X, presenta un mejor comportamiento para medidas menores a 20 cm, donde además, no es posible confundirlas con valores mayores a 20 cm como sucede con el Sharp-2Y0A21.

Luego de poner el prototipo en funcionamiento en los distintos escenarios y de realizar las instancias de presentación a expertos, fue posible constatar la necesidad de tener un mejor control de movimiento, para asegurarse que el robot pueda seguir determinadas trayectorias. Frente a esta necesidad, se plantea investigar las diferentes opciones de motores con capacidad de retroalimentación y ruedas, buscando que ambos sean del menor tamaño posible. Mejorar el control de movimiento no evita que las ruedas patinen y por lo tanto desvíen al robot de la trayectoria deseada, por eso se pretende estudiar otro tipo de ruedas omnidireccionales que se integren por un material más adherente que el de las ruedas actuales.

Por último, se plantea continuar desarrollando el software construido e implementar los distintos comportamientos que se solicitaron por parte del equipo. En particular, explorar la lógica de atractores, que requiere modificar la reacción que tiene el robot frente a los objetos, para que en lugar de recibir una fuerza repulsiva reciba una fuerza atractora.

Bibliografía

- [1] *Policy Papers UNESCO, TIC, educación y desarrollo social en América Latina y el Caribe, 2017.* https://crds.cepal.org/2/sites/crds2/files/tic_educacion_y_desarrollo_social_en_alc.pdf, Visitada Octubre 2018.
- [2] BENAVIDES F., OTEGUI X., AGUIRRE A., ANDRADE F.. *Robótica educativa de la mano del robot Butiá, Congreso Internacional de Informática en la Educación, Inforedu2013.*
- [3] JOHNSON, JEFFREY *Children, robotics, and education. Artificial Life and Robotics, 7(1-2) pp. 16–21.*
- [4] WING, J. M. *Computational Thinking (2006), 49(3), 33–35.*
- [5] RUIZ-VELAZCO, E. *Robótica Pedagógica. UNAM. 2007* http://cecte.ilce.edu.mx/seiem/file.php/12/sesion12/lec_rec/robotica_pedagogica.doc, Visitada Noviembre 2018.
- [6] MURPHY, ROBIN R. *Introduction to AI Robotics, MIT Press, 2000*
- [7] CAPPELLERI, J. C. Y TROCHIM, W. M. *Regression Discontinuity Design. In International Encyclopedia of the Social and Behavioral Sciences 2015 (Vol. 19, pp. 152–159). Elsevier Ltd.*
- [8] SISTO R. Y MATÍNEZ S. *Control y Comp ortamiento de Robots Omnidireccionales.* <https://www.fing.edu.uy/inco/grupos/mina/pGrado/easyrobots/doc/SOA.pdf>, Visitada Octubre 2018.
- [9] *Plan Ceibal - Uruguay.* <https://www.ceibal.edu.uy/es/institucional>, Visitada Octubre 2018.
- [10] *School of Robotics.* <http://www.schoolofrobotics.com/>, Visitada Octubre 2018.
- [11] *SM-S4303R.* <https://www.pololu.com/product/1248>, Visitada Octubre 2018.
- [12] *Dinamixel.* <http://www.robotis.us/ax-12a/>, Visitada Octubre 2018.
- [13] *Robot Khepera IV.* <https://www.k-team.com/khepera-iv>, Visitada Octubre 2018.
- [14] *HC-SR04 Datasheet.* <https://www.mouser.com/ds/2/813/HCSR04-1022824.pdf>, Visitada Octubre 2018.
- [15] *Ultrasonic Module HC-SR04.* <http://www.electronicwings.com/sensors-modules/ultrasonic-module-hc-sr04>, Visitada Octubre 2018.
- [16] *LV-MaxSonar-EZ-MB1000 Data sheet.* https://www.maxbotix.com/documents/LV-MaxSonar-EZ_Datasheet.pdf, Visitada Octubre 2018.
- [17] *Sharp-gp2D120X Data sheet.* <https://www.pololu.com/file/0J157/GP2D120-DATA-SHEET.pdf>, Visitada Octubre 2018.

- [18] *Sharp-2Y0A02 Data sheet.* https://www.sparkfun.com/datasheets/Sensors/Infrared/gp2y0a02yk_e.pdf, Visitada Octubre 2018.
- [19] *Sharp-2Y0A21 Data sheet.* http://www.sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf, Visitada Octubre 2018.
- [20] *VL53L0X - STMicroelectronics.* <https://www.st.com/en/imaging-and-photonics-solutions/vl53l0x.html>, Visitada Octubre 2018.
- [21] *Sensor Láser VL53L0X.* <http://www.playbyte.es/electronica/arduino/sensor-laser-vl53l0x/>, Visitada Octubre 2018.
- [22] *Sensor Láser VL53L0X - Data sheet.* <https://www.st.com/resource/en/datasheet/vl53l0x.pdf>, Visitada Octubre 2018.
- [23] *MAX7219 - Data sheet.* <http://www.alldatasheet.com/datasheet-pdf/pdf/73745/MAXIM/MAX7219.html>, Visitada Octubre 2018.
- [24] MYKE PREDKO. *Handbook of Microcontrollers*, McGraw-Hill, 1998, ISBN: 0-07-913717-2.
- [25] ARKIN, RONALD C. *Behavior-Based Robotics*, 1998, ISBN 9780262011655.
- [26] WOODS, S. *Exploring the design space of robots: Children's perspectives. Interacting with Computers (2006)*, 18(6), 1390-1418.
- [27] BARRETO F. AND BENITTI V. *Exploring the educational potential of robotics in schools: A systematic review*, *Computers & Education* 58 (2012) 978-988.
- [28] TOH, L. P. E., CAUSO, A., TZUO, P. W., CHEN, I. M., & YEO, S. H. *A Review on the Use of Robots in Education and Young Children. Educational Technology & Society (2016)*, 19 (2), 148-163.
- [29] SULLIVAN, A., AND BERS, M. U. *Gender differences in kindergarteners' robotics and programming achievement.* *International Journal of Technology and Design Education*, 23(3), 691-702.
- [30] MICHAEL BAR and ANTHONY MASSA. *Programming Embedded Systems*, Second Edition with C and GNU Development Tools.
- [31] *Bendocode.* http://www.bittorrent.org/beps/bep_0003.html, Visitada Octubre 2018.
- [32] *Sitio de eBay.* www.ebay.com, Visitada Octubre 2018.
- [33] *Sitio de Amazon.* www.amazon.com, Visitada Octubre 2018.
- [34] *Particle store.* <https://store.particle.io/products/photon>, Visitada Octubre 2018.
- [35] *Cypress.* <http://www.cypress.com/documentation/datasheets/cyw43362-single-chip-ieee-80211tm-bgn-macbasebandradio>, Visitada Octubre 2018
- [36] *ST Microcontrollers.* <https://www.st.com/en/microcontrollers/stm32f2-series.html>, Visitada Octubre 2018
- [37] *Particle presentation.* <https://www.particle.io/what-is-particle>, Visitada Octubre 2018
- [38] *Particle guide.* <https://docs.particle.io/guide/tools-and-features>, Visitada Octubre 2018

- [39] *MCC repository*. <https://github.com/xopxe/firmware>, Visitada Octubre 2018.
- [40] *Imagen de plataforma omnidireccional de 4 ruedas*. <https://www.electan.com/chasis-robot-omnidireccional-mecanum-4wd-p-6278.html>, Visitada Octubre 2018.
- [41] *Rueda Omnidireccional*. <http://www.holonomicwheel.com/>, Visitada Octubre 2018.
- [42] *Acoples*. <https://www.thingiverse.com/thing:923605>, Visitada Octubre 2018.
- [43] *Sharp gp2D120X*. [https://www.dfrobot.com/wiki/index.php/SHARP_GP2D120X_IR_ranger_sensor_\(4-30cm\)_\(SKU:SEN0143\)](https://www.dfrobot.com/wiki/index.php/SHARP_GP2D120X_IR_ranger_sensor_(4-30cm)_(SKU:SEN0143)), Visitada Octubre 2018.
- [44] *Sharp-2Y0A21*. <http://us.100y.com.tw/viewproduct.asp?MNo=94354>, Visitada Octubre 2018.
- [45] *Sharp-2Y0A02*. <http://us.100y.com.tw/viewproduct.asp?MNo=94354>, Visitada Octubre 2018.
- [46] *VL53L0X*. <https://cdn.littlebird.com.au/images/files/000/080/793/large/PL-2489.jpg?1535764308>, Visitada Octubre 2018.
- [47] *MAX7219*. <https://www.instructables.com/id/16x8-LED-dot-matrix-with-MAX7219-module/>, Visitada Octubre 2018.