

RAE

1. **TIPO DE DOCUMENTO:** Trabajo de grado para optar por el título de INGENIERO ELECTRÓNICO
2. **TÍTULO:** DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA MULTI-AGENTE ROBÓTICO MÓVIL PARA TAREAS COOPERATIVAS.
3. **AUTOR:** Germán Andrés Sanmiguel Mateus
4. **LUGAR:** Bogotá, D.C.
5. **FECHA:** Junio de 2014
6. **PALABRAS CLAVE:** Actuadores, Agente, Entorno, Inteligencia Artificial, Máquina de estados, Modelo de comunicaciones, Micro-controlador, Modelo de coordinación y organización, Modelo de tareas y experiencia, Sensores, Sistemas embebidos.
7. **DESCRIPCIÓN DEL TRABAJO:** El objetivo principal de este proyecto es el diseño de todo un sistema donde dos entes robóticos móviles pueden realizar tareas específicas gracias a su desplazamiento y a la coordinación de sus acciones; esto con el fin de incursionar en el campo de la inteligencia artificial, la cual se encarga de estudiar y construir agentes robóticos o de software con el fin de dar sus acciones como solución a un problema en un entorno específico. Se presentan los aspectos importantes para diseñar tanto el agente de manera individual como de manera colectiva, pues se organizan los instantes de acción para cada agente.
8. **LÍNEAS DE INVESTIGACIÓN:** Línea de investigación: Análisis y procesamiento de señales. Núcleo problemático: Robótica y control.
9. **METODOLOGÍA:** El enfoque de la investigación es empírico-analítico, ya que propone una manera de crear sistemas automáticos de robots móviles a partir de nuevas tecnologías en programación y procesos de control utilizado las máquinas de estado.
10. **CONCLUSIONES:** Los agentes cuentan con la capacidad de moverse al guiar su desplazamiento por medio de las referencias. Cuentan con la habilidad suficiente para ejecutar una tarea gracias al desplazamiento y al conocimiento interno que se les otorga en la descripción de cada estado en la máquina de estados. Se encontró que no fue necesario realizar un modelamiento cinemático para el desplazamiento del robot, pues para crear un control automático funcional basta con un programa de comportamiento de control reactivo basado en la información que va obteniendo al transcurrir el tiempo por medio de los sensores, las características del medio y los lazos de control. Los robots al comunicarse organizan un grupo de trabajo de líder y esclavo en el cual realizan las mismas tareas pero sólo uno comanda el momento de realizar las acciones al dar las órdenes para cumplir un objetivo más ambicioso.

**DISEÑO E IMPLEMENTACION DE UN SISTEMA MULTI-AGENTE ROBÓTICO
MÓVIL PARA REALIZAR TAREAS COOPERATIVAS**

GERMÁN ANDRÉS SANMIGUEL MATEUS

UNIVERSIDAD DE SAN BUENAVETURA

FACULTAD DE INGENIERÍA

INGENIERÍA ELECTRÓNICA

BOGOTÁ, D.C. - 2014

**DISEÑO E IMPLEMENTACION DE UN SISTEMA MULTI-AGENTE ROBÓTICO
MÓVIL PARA REALIZAR TAREAS COOPERATIVAS**

GERMÁN ANDRÉS SANMIGUEL MATEUS

Trabajo presentado como requisito parcial para optar por el título de profesional en Ingeniería

Electrónica

Asesor: Ingeniero Javier Alejandro Sáenz Leguizamón

UNIVERSIDAD DE SAN BUENAVETURA

FACULTAD DE INGENIERÍA

INGENIERÍA ELECTRÓNICA

BOGOTÁ, D.C. - 2014

**“DISEÑO E IMPLEMENTACION DE UN SISTEMA MULTI-AGENTE ROBÓTICO
MÓVIL PARA REALIZAR TAREAS COOPERATIVAS”**

GERMÁN ANDRÉS SANMIGUEL MATEUS

UNIVERSIDAD DE SAN BUENAVETURA

FACULTAD DE INGENIERÍA

INGENIERÍA ELECTRÓNICA

BOGOTÁ

2014

INTRODUCCIÓN

Gracias a los inmensos avances en el análisis de señales, en la transmisión de datos y en la creación de entes artificiales mecánicos, electrónicos y de software, un futuro tangible es aquel donde máquinas y seres humanos interactúan constantemente para vivir.

Nuevas formas de percepción y de acción se abren en el mundo como se conoce, ya que es muy probable que máquinas automáticas e inteligentes tomen decisiones propias ante los problemas cotidianos sin la necesidad de un operador manual. Será común ver agentes robóticos simulando ser humanos, capaces de interactuar con los objetos e individuos del mundo real gracias a las capacidades mecánicas, lógicas y perceptuales con las que fueron diseñados; la inteligencia artificial permite el estudio de agentes de software y hardware que combinados generan un agente de control. Hoy en día es común ver innumerables aplicaciones de software que permiten un nivel de interacción agradable con mundos virtuales creados por medio del celular, las consolas de video juegos, internet, etc. Es casi seguro que otro nivel de interacción existirá al poder generar aplicaciones para ser implementadas en dispositivos robóticos, donde software y hardware se complementan para permitir el desarrollo de nuevas formas de acción mediante entes artificiales. Las nuevas tecnologías han causado tal influencia en nuestras vidas que sin electricidad, sin computadoras, sin celulares, sin automóviles, y sin muchos otros dispositivos, no sólo eléctricos sino artesanales y mecánicos, sería muy tortuosa la existencia, ya que la calidad de vida general del ser humano estaría disminuida.

Por todo lo anterior se ve la necesidad de que los esfuerzos de este proyecto vayan enfocados al estudio y desarrollo de agentes robóticos móviles cooperativos programables que ayuden a la humanidad de alguna manera siendo herramientas intelectuales y locomotoras de acción e interacción con los objetos del mundo real.

PLANTEAMIENTO DEL PROBLEMA

ANTECEDENTES

LOCAL

Título del proyecto: Robotsumo, “Pepe”.

Desarrollado por: Grupo de investigación de la Universidad de San Buenaventura (semillero SIRA).

Resumen del proyecto: Uno de los proyectos adelantados en robótica móvil es un robotsumo, presentado al concurso organizado por la Universidad Nacional en el año 2011, en el cual ocuparon el primer puesto. Como su definición lo dice, el comportamiento del robot viene ligado al deporte sumo, donde los competidores se enfrentan empujándose el uno a otro hasta sacarse de un área delimitada en el suelo. Su diseño e implementación se realizó con circuitos integrados de tecnología DIP para implementar todo el sistema de control general para el comportamiento de Pepe. Contaba con sistema de regulación de velocidad y dirección para los motores, sistema de sensores para percibir al oponente y al mismo tiempo otro sistema encargado de examinar el área de operación implementando un detector de línea. Además incluía un sistema de empuje para desplazar más fácilmente al contrincante en su interacción.

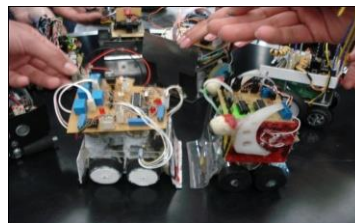


Figura 1. Diseño robotsumo

NACIONALES

Título del proyecto: Diseño e implementación de un módulo didáctico para aprendizaje en la construcción, implementación y manipulación de robots. (Lancheros, 2010)

Desarrollado por: Grupo de automatización industrial, Universidad Cooperativa de Bogotá-Colombia.

Resumen del proyecto: Es un proyecto realizado como herramienta pedagógica que permite afianzar el aprendizaje en la construcción, implementación y manipulación de robots utilizando piezas Lego, enseñando a los estudiantes a diseñar proyectos con robots móviles con inteligencia artificial. Seleccionaron los módulos de comunicación entre los robots, lenguajes de programación, UML (Modelamiento Unificado de Lenguaje), y desarrollaron un software para adquisición de datos de los robots móviles. Usaron netbeans 6.0, compilador grafico de Java. El software muestra al estudiante los pasos para construir, implementar y programar robots móviles básicos.

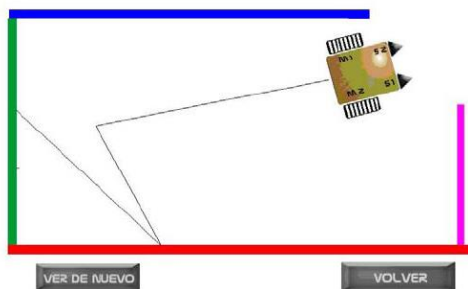


Figura 2. Simulación de un robot tipo topo, netbeans 6.0

Cuenta con una tarjeta de adquisición de datos para la comunicación con el software y algunos telerobots. Usaron microcontrolador de Motorola y los módulos de comunicación son XBEETM/802.15.4

fabricados por MaxStream. Proponen tutoriales para construir robots tipo topo, ratón, polilla, gato o perro, con posibilidades diversas de desplazamiento y se programan inalámbricamente a través del módulo.

Título del proyecto: SMART: Sistema multi-agente robótico. (www.unalmed.edu.co, 2007)

Desarrollado por: Jovani Alberto Jiménez Builes; Demetrio Arturo; Ovalle Carranza; John Fredy Ochoa Gómez. Universidad Nacional de Colombia.

Resumen del proyecto: El siguiente proyecto es un Sistema Multi-Agentes Robóticos (MARS) que mediante una explicación de las áreas relacionadas con la inteligencia artificial presentan el Sistema Multi-Agente Robótico (SMART). SMART es un enjambre inteligente conformado por un robot nodriza y tres robot tipo baliza (guías) que navegan de manera colaborativa en un escenario estructurado.

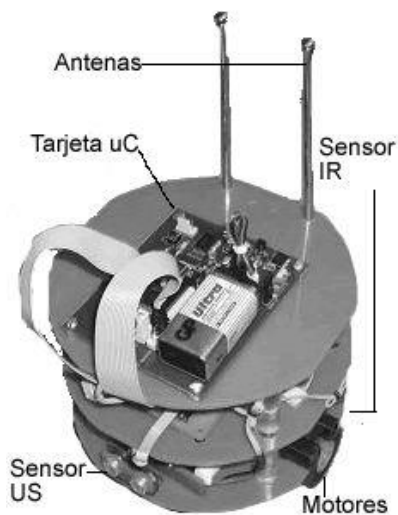


Figura 3. Agente Robótico del proyecto SMART.

INTERNACIONAL

Título del proyecto: Sistema SMART para crear robots organizados socialmente.

Desarrollado por: Investigadores de la Universidad Politécnica de Madrid (UPM).

Resumen del proyecto: Las máquinas se ayudan mutuamente para cumplir un objetivo, y para ello se valen de las mejores aptitudes del robot-individuo. (Cena C. G., Saltaren R., Blazquez J. L., Aracil R., 2010)

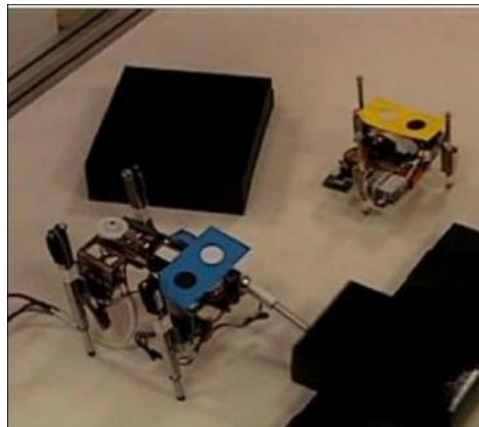


Figura 4. Grupo de Robots y Máquinas Inteligentes. UPM.

El grupo de Robots y Máquinas Inteligentes de la UPM ha desarrollado el sistema SMART1, un grupo de agentes (robots) cuya finalidad principal es lograr la mejor configuración para cumplir con un objetivo. Si algún agente no puede completar su misión, otro(s) agente(s) acuden en su ayuda, a tal extremo que pueden sustituirlo en la tarea original o bien ayudarlo para que la termine. Para ello, se ha desarrollado una sofisticada arquitectura software y hardware donde la cinemática, la dinámica y la planificación de trayectorias, entre otras, se combinan a través de un sistema de control inteligente para que los agentes robóticos desarrollen sus cometidos. El grupo de SMART, cuenta con robots de tres y cuatro patas (ocho y

trece grados de libertad, respectivamente). Además, cada robot dispone de un sistema de comunicación *bluetooth* para contactar con un puesto central de control desde el cual se establecen las tareas a realizar.

Cuando un robot no puede cumplir su misión envía un mensaje con una codificación específica al ordenador central, que se encarga de distribuirlo a aquellos otros robots que se encuentren en posibilidad de acudir a su llamada. Dependiendo de la prioridad del mensaje enviado, es posible que los miembros del grupo dejen de realizar sus tareas y acuda en “auxilio” de su compañero.

Las capacidades de los agentes de 3 y 4 patas son diferentes. Técnicamente, los de 4 patas tienen mayor capacidad de locomoción y son más estables, pero son grandes, pesados, y más lentos que los agentes de tres patas que, por el contrario, no tienen más opciones que reptar para desplazarse.

1.2 DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA

Gracias a la electricidad y a la electrónica han cambiado muchos aspectos de la cotidianidad y se han facilitado las acciones diarias de los seres humanos. Para los ingenieros, el desarrollo software y hardware ha sido una herramienta primordial para encontrar soluciones a muchos problemas cotidianos.

Hoy en día los computadores modernos, gracias a su acelerado desarrollo, permiten acortar cada vez más la barrera de imposibilidad que existía hace algún tiempo al crear programas para solucionar y automatizar una situación dada. Existían algunas limitaciones a la hora de proponer algoritmos de solución para un problema debido a la cantidad de líneas necesarias frente a la cantidad de líneas posibles en su implementación. Además la capacidad de procesamiento y manejo de la información ha ido aumentando exponencialmente con el tiempo y ha facilitado la instrumentación de los procesos, antes era insuficiente. Hace poco tiempo era imposible hablar de máquinas que lucieran como humanos, animales o

vehículos, y que actuaran según las leyes de conocimiento básicas, pero es muy fácil encontrar en la red múltiples desarrollos de robots humanoides (como ASIMO), robots animales y robots móviles y transportadores.

En la industria existen montacargas de cuatro ruedas encargados de transportar cosas pesadas. “Los controles básicos para levantar incluyen "levantar" hacia arriba y hacia abajo las horquillas, inclinación de las horquillas (para una mejor estabilidad). Ser capaz de manipular estos controles, sin mirar, es esencial para una operación apropiada del montacargas” (www.ehowenespanol.com). “En las construcciones existen cargadoras; es un tipo de tractor que consiste en un gran cubo montado en la parte delantera que se utiliza para recoger materiales. Una cargadora de ruedas puede verse a menudo fuera de las zonas de construcción, ya que con frecuencia se utilizan para limpiar escombros o nivelar el terreno de un área de construcción. Estas cargadoras eliminan la suciedad, los materiales de construcción y ladrillos. Sin embargo cuando se les da otros accesorios pueden utilizarse para todo tipo de aplicaciones, como el acaparamiento de las tuberías con un tenedor tubo, para el almacenamiento de materiales con un cubo de alta descarga o manipulación de basura con cubos de basura. La versatilidad de las cargadoras sobre ruedas las convierte en una herramienta valiosa en la industria de la construcción” (www.ehowenespanol.com). Sería de gran ayuda si en este tipo de máquinas móviles no fuera necesaria y obligatoria la operación de un ser humano, sino que bastara con un sistema de sensores, actuadores y código para controlar y ejecutar las tareas de recolección en una fábrica o empresa.

¿Qué elementos serían necesarios para que el ente artificial pudiera desenvolverse en un entorno de manera ágil y coordinada?

1.3 JUSTIFICACIÓN

No se consideraba el alcance en las acciones que podría llegar a lograr un robot, pero hoy en día es un hecho que se pueden crear robots para propósitos específicos. El aprovechamiento de las herramientas de programación, control y robótica actuales, brindan amplias posibilidades de éxito a la hora de proponer alternativas para resolver problemas mediante el control automático de agentes robóticos móviles y comunicativos. La idea mundial es poder llegar a tal nivel de interacción entre humanos y máquinas automáticas en el que estos últimos lleguen a parecer y actuar tanto como un humano común y corriente, el cual convive y ayuda a verdaderos seres humanos en distintas labores cotidianas. Por ello es sensato creer que en un futuro cercano entes robóticos realizarán tareas cotidianas de manera autónoma como individuo y a la vez colectiva como equipo, mejorando y facilitando un poco más la calidad de vida. Si se analiza y se establece de alguna forma un modelo que explique el proceso que realiza el cuerpo humano para pensar y actuar, entonces se puede llegar a automatizar varias situaciones en las que las acciones robóticas son la solución a un problema dado. Todo esto para crear entes artificiales especialmente diseñados para funciones específicas.

1.4 OBJETIVOS

OBJETIVOS GENERAL

Diseñar e implementar un sistema de dispositivos robóticos móviles cooperativos capaces de desplazar un objeto en un ciclo.

1.4.2 OBJETIVO ESPECIFICOS

1. Definir el problema a resolver en el entorno.
2. Seleccionar los elementos mecánicos y electrónicos a implementar.
3. Ensamblar las partes electrónicas en la arquitectura robótica.
4. Diseñar los algoritmos para el comportamiento, la comunicación y la coordinación de los robots.
5. Implementar el programa y realizar pruebas de comportamiento.

ALCANCES Y DELIMITACIONES DEL PROYECTO

Los robots móviles realizarán las siguientes tareas: trabajar de manera cooperativa para desplazar objetos, movilizarse de manera autónoma por el entorno, capacidad para comunicar la información, recolectar o empujar los objetos de interés.

Las figuras geométricas deben ser de material de balsa no más grandes que los robots cooperativos, deben tener: de altura entre 4-10 cm, de ancho entre 5-20 cm y de profundidad entre 5-30 cm.

Número de robots

Se implementaran 2 robots para demostrar tareas cooperativas mediante la interacción relevada de un objeto.

Geometría física

Los robots móviles tendrán de: Altura entre 10-15 cm, ancho y profundidad entre 15-25 cm.

Comunicación

Transmisión serial de la información. Sólo se transmite información en una sola dirección; es decir que hay un agente Líder R1 y un Esclavo R2, pero los agentes no se conocen ni se identifican.

Espacios de movilidad

Los robots móviles serán de movimiento diferencial. No podrán subir ni bajar escaleras por lo que el piso deberá estar a un mismo nivel. Las pruebas de comportamiento de los agentes deben realizarse siempre en el mismo terreno. Características diferentes de suelo cambian por completo la operatividad del sistema en cuanto a giros y velocidades de desplazamientos.

Las paredes de referencia son la frontera que limita los espacios del mundo del agente, por lo que en el entorno no puede existir otra cosa más que las referencias y los objetos del mundo.

La distancia entre los puntos A-B debe ser igual que entre los puntos B-C de la figura 15 para evitar choques. Ocurre que R2 entregaría un objeto mientras R1 dos o tres dependiendo de la diferencia de distancias.

La posición inicial de los agentes debe ser lo más alineada posible con respecto a los objetos para mejorar la continuidad del ciclo.

2. METODOLOGÍA

2.1 ENFOQUE DE LA INVESTIGACIÓN

El enfoque de la investigación es empírico-analítico, ya que propone una manera de crear sistemas automáticos de robots móviles a partir de nuevas tecnologías en programación y procesos de control utilizando las máquinas de estado.

Para generar robótica se incorporan la mecánica, la electrónica y la informática en una sola idea. La estrategia de trabajo comienza ideando un dispositivo móvil capaz de desplazarse en un ambiente controlado especificando los alcances de sus capacidades mecánicas (carcasa, motores o patas, articulaciones o pinzas). Al mismo tiempo se especifican los dispositivos electrónicos necesarios con los que contará la estructura mecánica para desempeñar las tareas (sensores y actuadores). Posteriormente se escogen los circuitos integrados que comandaran el funcionamiento lógico de la carcasa y el comportamiento de sus elementos (comunicación, electrónica de potencia y de control).

Contando con la arquitectura robótica lista para ser implementada, se procede a realizar el análisis del problema estudiando los datos de entrada y salida involucrados en el proceso para generar acciones de comportamiento e interacción controladas entre los robots.

Posteriormente se define muy específicamente: ¿Qué debe hacer el sistema multi-agente? ¿Qué acciones deben ejecutar los robots? ¿Qué datos se necesitan para obtener los resultados deseados? Aquí se propone el algoritmo solución que es codificado en un lenguaje de programación para realizar pruebas de comportamiento en la arquitectura robótica.

Finalmente se comprueba y se corrige el programa de acuerdo a los resultados obtenidos frente a los resultados deseados en el comportamiento de los robots.

2.2 HIPÓTESIS

Se desea implementar un sistema robótico automático para generar interacciones con los objetos del mundo real, coordinando las acciones de dos robots ante un problema que busca ser solucionado colectivamente y a manera de ciclo, donde la única interacción humana deseada es la indicación para indicar su inicio o final de funcionamiento.

3. LÍNEA DE INVESTIGACIÓN DE LA UNIVERSIDAD/SUB-LÍNEA DE LA FACULTAD/CAMPO TEMÁTICO DEL PROGRAMA

Línea de investigación: Análisis y procesamiento de señales.

Núcleo problemático: Robótica y control.

4. MARCO TEÓRICO Y CONCEPTUAL

4.1. ROBOT

Cada vez que se escucha el término robot se asocia a un tipo de máquina androide que luce y actúa como un ser humano. Esa idea es debida al filósofo Karel Capek quien utilizó por primera vez el término ruso “robot”, que significa servidumbre o esclavitud, para referirse a máquinas humanoides que aniquilan la raza humana en su obra “R.U.R., Robots Universales Rossum”.

Según Mikell Groover en su libro *Automation, Production Systems and Computer Integrated Manufacturing*¹, define al robot industrial como una máquina programable, de propósito general, que posee ciertas características antropomórficas, es decir, con características basadas en la figura humana. No todos los robots parecen humanos, aunque si todos son dispositivos mecánico-electrónicos que interactúan con las cosas del mundo, al convertir señales de un programa de cómputo en acciones mecánicas ejecutadas por una plataforma móvil por medio de movimientos y desplazamientos en un entorno.

Los robots que podemos encontrar están divididos en tres categorías principales que son:

“Manipuladores o poliarticulados: cuentan con extremidades artificiales que les permiten interactuar con los objetos del mundo de manera precisa, ya que al estar fijos a su estructura proveen alta fidelidad en sus movimientos, generalmente de agarre; **Móviles:** cuentan con patas, piernas, ruedas u otros mecanismos similares que les permite movilizarse por un medio para cumplir un objetivo específico derivado de su desplazamiento y de su percepción; **Híbridos:**

¹ En español Automatización, sistemas de producción y manufacturación de computadores integrados.

son robots móviles con manipuladores incorporados. Es por ello que entre estos híbridos podemos encontrar diversidad de especímenes tales como androides (humanos), zoomórficos (animales) o transportadores (vehiculares)” (Russell, S. y Norvig, P., 2004)

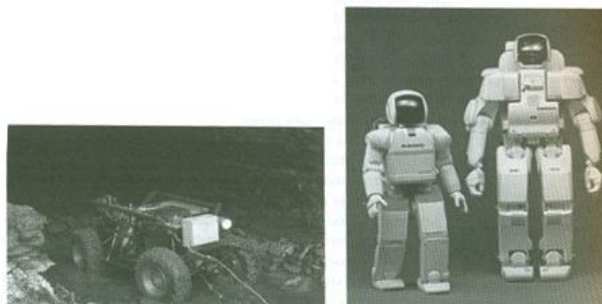


Figura 5. Robot Sojourner de la NASA. Híbridos P3 y Asimo de Honda.²

Los robots manipuladores son multifuncionales, por lo que su uso se ha extendido para diversas aplicaciones que implican movimientos e interacciones con los objetos del mundo. Se defienden tres modos de control de robots manipuladores: “**Manual** (controlado directamente por un operador), **De secuencia fija** (proceso controlado es previamente estudiado y se repite de forma invariante), y **De secuencia variable** (controlado de acuerdo a alteraciones en las características de los ciclos de trabajo)” (<http://www.proton.ucting.udg.mx/materias/robotica/r166/r64/r64.htm>).

Existe otra forma de clasificarlos de acuerdo al tipo de control implementado de la siguiente manera. “**Tipo-A** Manipulador con control manual o telemando. **Tipo-B** Manipulador automático con ciclos pre ajustados, regulación mediante fines de carrera o topes, control PLC (Controladores Lógicos Programables) o accionamiento neumático, eléctrico o hidráulico. **Tipo-C** Robot programable con trayectoria continua o punto a punto. Carece de conocimiento sobre su entorno. **Tipo-D** Robot capaz de

² Russell y Norvig, Inteligencia Artificial un enfoque moderno, 2ª edición, 2004, p. 1024.

adquirir datos de su entorno, readaptando su tarea en función de estos.”

(http://www.profesormolina.com.ar/tecnologia/robotica/res_rob.htm).

4.2. ROBOTS COMO INSTRUMENTOS INDUSTRIALES

Un robot puede ser visto como un proceso industrial donde cada acción, operación o movimiento es destinada a su automatización que está regida por una técnica de control adecuada. Un robot humanoide bien estructurado sería de gran ayuda en cuanto a labores domésticas se refiere, aunque sería una muy ardua y costosa tarea de mecánica, control y electrónica a llevar a cabo.

Las técnicas de control determinan el comportamiento general de una tarea automatizada. Los elementos de un lazo de control se muestran en la figura 6.

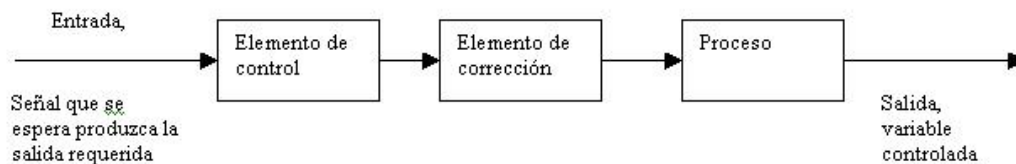


Figura 6. Diagrama de flujo de un lazo abierto de control abierto.³

Los sistemas a controlar deben tener cuatro elementos principales para llevar a cabo un objetivo de control: una **planta** o estructura denominada proceso (arquitectura móvil), el **transductor** o elemento de medida de la variable controlada (sensores), el **elemento de control** (micro-controlador) y el elemento final de corrección o **actuador** (motores y servomotres).

³ [en línea], disponible en:
<http://3.bp.blogspot.com/C6yL6DJSv10/URKEw1ARbSI/AAAAAAAAAAg/pXKCNPFdo/s1600/control+1.png>,
 recuperado: 19 de marzo de 2013.

4.2.1 Lazo de control cerrado

El lazo de control es el método para automatizar los procesos. Se ve la necesidad de implementar métodos que realizaran las tareas de un sistema de control de manera automática al manipular las variables involucradas para “mantenerlas en un valor fijo deseado, en un valor variable con el tiempo, para guardar alguna relación determinada con otra variable (...) o para generar una acción específica donde se vean reflejadas condiciones estables de calidad y características de un producto” (Creus, 1997). El procesamiento de señales se realiza usualmente por un micro-controlador, asistido por un bloque de memoria de acuerdo a las instrucciones y condiciones de un programa para su comportamiento. Los programas consisten en un número considerable de operaciones lógicas y aritméticas, realizadas a gran velocidad donde se tiene lazos de control específicos codificados en un circuito integrado. En la figura 7 muestra un lazo cerrado de control.

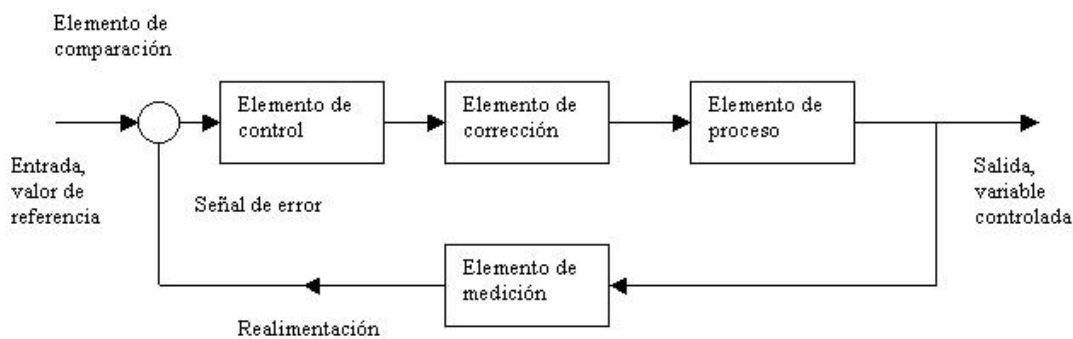


Figura 7. Diagrama de flujo de un lazo cerrado de control. ⁴

La señal de referencia es el comportamiento deseado codificado en reglas de control implícitas en el programa implementado en un micro-controlador. La salida del producto es cada ejecución de

⁴ [en línea], disponible en: <http://3.bp.blogspot.com/C6yL6DJSv10/URKEw1ARbSI/AAAAAAAAAAg/pXKCNPFdo/s1600/control+1.png>, recuperado: 19 de marzo de 2013.

movimientos o acciones por parte de la arquitectura robótica al comparar la señal de salida con la señal de referencia para tomar acciones de control.

4.2.2 SENSORES O TRANSDUCTORES

Para conocer las condiciones en las que se encuentra una variable determinada se usa un transductor o sensor. Un transductor es un elemento que puede convertir alguna información física en información eléctrica para ser procesada y controlada con circuitos electrónicos. Una variable natural puede ser una altura, un color, un sonido, etc. la cual puede ser representada de manera fiel con un valor de corriente o voltaje para su manipulación. Un sensor es una clase de transductor que mide o indica el estado de dicha magnitud.

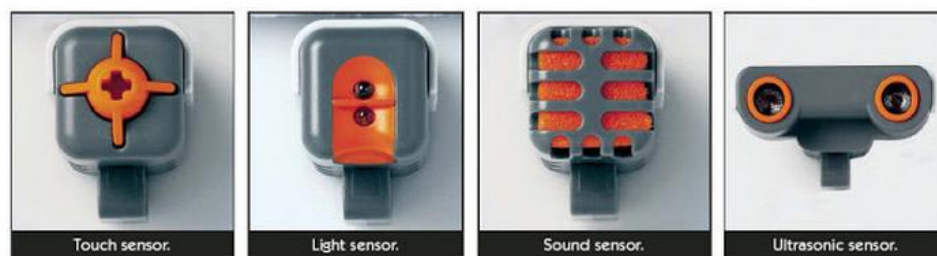


Figura 8. Clases de sensores robóticos.⁵

En un sistema de control es indispensable que un elemento actúe como observador e indicador del estado de la variable en proceso todo el tiempo para mejores resultados. Quiere decir esto que los sensores son esenciales para que un sistema de control retroalimentado reconozca, identifique y corrija el

⁵ [en línea], disponible en: <http://tecno4merced1213.blogspot.com/2013/03/robot-mindstorms-nxt-20-alvaro-y-victor.html>, recuperado: 29 de abril de 2013.

comportamiento de las magnitudes físicas mediante una acción correcta de regulación. “Los sensores son la interfaz perceptual entre el robot y su entorno” (Russell, S. y Norvig, P., 2004).

4.2.3 ACTUADORES O EFECTORES

Un actuador es un dispositivo que convierte una magnitud eléctrica en una salida generalmente mecánica al interactuar o manipular los elementos del mundo físico; de allí que se le llamen en robótica más comúnmente efectores, pues esa es una definición más acertada de acuerdo a las acciones que lleva a cabo un robot.

Las características del efector dependen de las capacidades con las que debe contar el agente robótico según la complejidad de las tareas que debe llevar a cabo y de la inteligencia para su manejo. “Los efectores son el medio por el cual los robots se mueven y cambian la forma de sus cuerpos” (Russell, S. y Norvig, P., 2004).



Figura 9. Switchblade, robot multimodal diseñado por estudiantes de la UCSD para control de desplazamientos.⁶

⁶ [en línea], disponible en: <http://www.ni.com/white-paper/11565/es/>, recuperado: 2 de mayo del 2013.

Algunos ejemplos más comunes de actuadores son motores, válvulas de presión de agua o gas o articulaciones robóticas, cuyo único objetivo es interpretar una señal de control mediante acciones de corrección frente a una referencia o plan. Un actuador puede ser un simple dispositivo eléctrico como un transistor, un potenciómetro, los SCR, los TRIAC, los Amplificadores Operacionales o los micro-controladores.

4.3 ROBOTS COMO INSTRUMENTOS EDUCACIONALES

La inclusión de la robótica en los procesos de control y aprendizaje es cada vez más común, ya que permite un desarrollo más completo de las teorías de programación y electrónica actuales. La combinación de programación, mecánica y electrónica abren diferentes campos de aplicación donde lo que se busca es proponer otras alternativas en la solución de problemas para ser resueltos más eficientemente por la acción de robots.

Hoy en día gracias al amplio desarrollo en circuitos integrados, tanto hardware como software abren la posibilidad de generar nuevas alternativas de desarrollo para crear sistemas robóticos móviles autónomos de tareas específicas en una plataforma de calidad que evita sobrecostos, daños o esfuerzos innecesarios a la hora de diseñar. Gracias a la inclusión de arquitecturas robóticas comerciales y de sistemas embebidos, las pruebas y simulaciones pueden ser codificadas e implementadas más rápidamente en la planta de proceso, en este caso, en el robot móvil, lo permite observar el comportamiento del robot al programa propuesto como solución de una manera más rápida y veraz.

4.3.1. Agente

La robótica es una ciencia o rama de la tecnología que estudia el diseño y construcción de máquinas capaces de desempeñar tareas realizadas por el ser humano o que requieren el uso de la inteligencia. Isaac Asimov en su libro “Yo robot” publicado en 1942 define a la robótica como la combinación de las ciencias de la inteligencia artificial y de la mecánica (incluida la electrónica) para crear nuevas formas de vida. Un agente puede ser visto como un programa de cómputo que determina las salidas establecidas a cierto conjunto de entradas conocidas. “Véase una calculadora como un agente para el cual $\ll 4 \gg$ es la salida a la entrada $\ll 2+2 = \gg$ ” (Russell, S. y Norvig, P., 2004).

“La primera máquina calculadora se creó en 1623 por el científico alemán Wilhelm Shickard, aunque la Pascalina en 1642 por Blaise Pascal sea más famosa. Rudolf Carnap (1891-1970) desarrolló la doctrina del positivismo donde todo el conocimiento se puede caracterizar mediante teorías lógicas relacionadas con sentencias de observación que corresponden a estímulos sensoriales. La primera teoría en mostrar la mente como un proceso computacional fue gracias al libro *The Logical Structure of the World* en 1928 por Carnap y Hempel, donde explican como el conocimiento se obtiene de la experiencia. El concepto de lógica formal fue gracias a George Boole (1815-1864) quien definió la lógica proposicional o Booleana. En 1870 Gottlob Frege (1848-1925) extendió la lógica Booleana para incluir objetos y relaciones, y creó la lógica de primer orden. Los primeros autores de un trabajo de IA fueron Warren McCulloch y Walter Pitts en 1943, donde se incluía estudios del comportamiento de las neuronas del cerebro, además del análisis formal de la lógica Booleana y la teoría computacional de Turing, y “propusieron el modelo de neuronas artificiales donde cada neurona

equivale a una proposición con unos estímulos adecuados. Mostraron que cualquier función de cómputo podría calcularse mediante alguna red de neuronas interconectadas, donde los conectores lógicos (and, or, not,...) se pueden implementar con estructuras de redes sencillas” (Russell, S. y Norvig, P., 2004).

Existen varias definiciones para explicar lo que es un agente, pero para entender en pocas palabras su definición es más fácil usar sinónimos que encajan perfectamente con la explicación:

“knowbots/Robots Basados en Conocimiento (knowledge based robots), softbots/Robots de Software (software robots), taskbots/Robots (Basados en Tareas), userbots/Robots de Usuario, Agentes personales, Agentes autónomos y asistentes personales” (Hycinth, 1996), donde todas ellos abarcan lo que un agente artificial puede ser mediante sus acciones, naturaleza e inteligencia.

En la figura 10 se observa la representación del agente más simple y sus relaciones. Si el ser humano fuera un agente, el mundo en la figura sería el medio ambiente; las percepciones serían toda la información que nos rodea (visual, auditiva, etc); los sensores serían los elementos que nos dejan percibir dicha información, estos sería los ojos, la nariz, la lengua, la piel, etc. El cuadro con el signo de interrogación sería el cerebro controlador que nos permite entender de alguna manera los sonidos, las imágenes, los olores, el contacto, la temperatura, etc. y es quien toma las decisiones de la acción a realizar; los actuadores serían las piernas, los brazos, los dientes, los poros, etc. Por último las acciones devueltas al ambiente serían los movimientos tales como masticar, sudar, hablar, correr, etc. dadas como salidas.

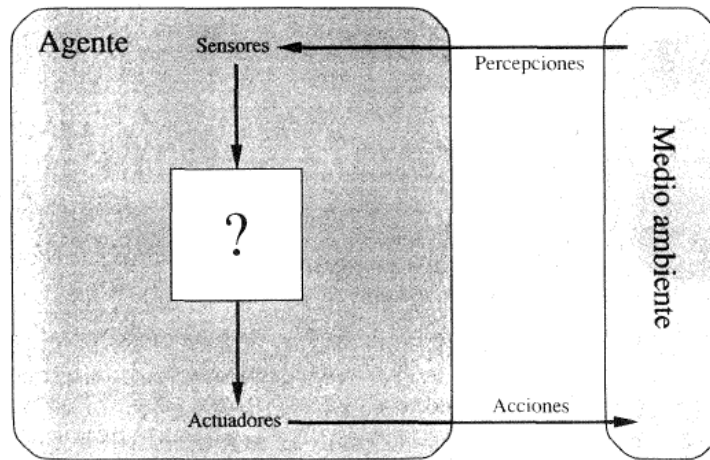


Figura 10. Esquema representativo de un agente.⁷

4.3.2 Estructura de un agente

Para entender lo que puede llegar a ser un agente robótico es fácil asociarlo al cuerpo humano, ya que ambos necesitan como elementos básicos de funcionamiento un cerebro que maneje la información recibida y una estructura física que le permita percibir y actuar con su medio ambiente.

Agente= arquitectura + programa

“El trabajo de la IA es diseñar un programa del agente que implemente un programa de cómputo que proyecte las percepciones del agente en acciones” (Russell, S. y Norvig, P., 2004) por medio de su arquitectura compuesta por sensores, actuadores, y circuitos de comparación y de control de movimiento.

⁷ Russell y Norvig, Inteligencia Artificial un enfoque moderno, 2ª edición, 2004, p. 38.

4.3.3 Agente Reactivo Simple

“Se trata de los agentes más simples cuyo comportamiento aleatorio de un tipo adecuado puede resultar en acciones racionales en algunos entornos multi-agente. Poseen una inteligencia limitada. El agente funcionará solo si puede tomar la decisión correcta sobre la base de percepción actual. El agente genera las conexiones desde las percepciones a las acciones gracias al programa general que contiene las reglas de condición acción” (Russell, S. y Norvig, P., 2004). Un agente reactivo es aquel que por medio de un grupo de sentencias, sensores y actuadores es capaz de transmitir algún tipo de energía a los objetos de un medio ambiente. Un agente robot recibe pulsaciones de un teclado, de archivos de información y de paquetes de comunicaciones a modo de entradas sensoriales (para entender o interpretar cómo es el mundo ahora) y actúa (de acuerdo las reglas de condición-acción establecidas) con movimientos, mensajes de monitor, escribiendo ficheros o enviando información.

4.3.4. Agentes Reactivos simples Basados en modelos

“Generalmente almacenan información de las partes del mundo que no pueden ver, o lo que es lo mismo, el agente debe mantener algún tipo de estado interno que dependa de la historia percibida. Se necesita alguna información acerca de cómo evoluciona el mundo tanto si está implementado con un circuito Booleano simple o teóricas científicas completas” (Russell, S. y Norvig, P., 2004).

4.4 ENTORNOS DE TRABAJO

Para facilitar el diseño del sistema de agentes se clasifica al medio de trabajo de acuerdo a las características del entorno, para así determinar un método de solución adecuado y afrontar de la mejor manera el diseño del comportamiento del sistema.

A continuación se explican las propiedades de un entorno de manera precisa. (Russell y Norvig, 2004, p. 47-49).

4.4.1 Totalmente observable Vs Parcialmente Observable

Si los sensores detectan los aspectos relevantes para la toma de decisiones se dice que el entorno es totalmente observable, que es lo más conveniente. Un entorno puede ser parcialmente observable debido a ruidos en el entorno o a sensores inexactos o porque no se recibe información por parte del sistema.

4.4.2 Determinista Vs Estocástico

Si el siguiente estado del medio depende totalmente del estado actual y de la acción ejecutada por el agente, el entorno es determinista. De lo contrario es estocástico.

4.4.3 Episódico Vs Secuencial

Consiste en episodios atómicos, donde cada episodio es la percepción del agente y la realización de una única acción depende sólo del episodio en sí mismo. El siguiente episodio no depende de las acciones

que se realizaron en episodios previos. Muchas de las acciones de clasificación son episódicas. En entornos secuenciales las decisiones presentes pueden afectar a decisiones futuras.

4.4.4 Estático Vs Dinámico

En un entorno estático el agente no necesita estar pendiente del ambiente mientras está tomando una decisión sobre una acción, ni preocuparse sobre el paso del tiempo. En el dinámico el entorno puede cambiar mientras el agente delibera.

Si el entorno no cambia, pero el rendimiento del agente cambia, entonces el entorno es semi-dinámico.

4.4.5 Discreto Vs Continuo

La distinción se hace en cuanto al estado del medio. A la forma en cómo se maneja el tiempo, las percepciones y las acciones del agente. El ajedrez tiene un conjunto discreto de percepciones y acciones. Un taxista conduciendo define un estado continuo y un problema de tiempo continuo, como la velocidad y la ubicación de los agentes.

4.4.6 Agente Individual Vs Agente Multi-agente

Se entiende como entorno multi-agente aquel en el que más de dos agentes son puestos en él. Pueden o no estar relacionados para solucionar un problema de manera colectiva o competitiva. Ejemplos

comunes de tareas donde varios individuos o agentes trabajan son los deportes en equipo o individuales; en ingeniería encontramos la administración de procesos de control o el tráfico en las redes computacionales.

“En términos de comunicación es importante aclarar que influye en cada agente si su modo de operación es cooperativo o competitivo ya que, o usa la cooperación y la comunicación como regla primordial o la descarta por completo al no comunicarse, y opta por la alternativa de observar el comportamiento de otros agentes para operar, lo que genera competición. Los entornos competitivos son aquellos en los cuales las tareas de los agentes son las mismas en cuanto a objetivos, lo cual quiere decir que los recursos del entorno están en conflicto y cada agente se encuentra en un juego entre adversarios” (Russell, S. y Norvig, P., 2004).

Lo importante entonces es analizar de qué manera puede afectar la operación de un agente a otro para facilitar el diseño de un sistema multi-agente. El pensamiento matemático comienza con frecuencia con el proceso de abstracción; esto es observar una similitud entre dos o más acontecimientos u objetos [9]. Aquella percepción y abstracción de la información se logra con el diseño de un lazo de control y de razonamiento, comandado por una máquina de estados donde los agentes saben que hacer gracias a la definición de sus estados y sus relaciones.

La IA no sólo intenta comprender el proceso complejo del pensamiento y la acción humana, sino que también trata de construir máquinas inteligentes que emulan las capacidades para solucionar un problema dado en un entorno determinado, al tomar decisiones ante situaciones de la vida.

Las capacidades que pueda poseer cualquier ente robótico se pueden expresar en términos del enfoque deseado para el elemento creado; pueden ser softbots o robots, por lo que se convierten en

agentes de tareas específicas más que robots industriales. Por eso, al desarrollar un sistema con agentes es práctico relacionarlo con las características del entorno de trabajo.

4.5 METODOLOGÍAS DE DESARROLLO MULTI-AGENTE

El primer desafío para un entorno multi-agente es el desarrollo del programa de movimiento para cada agente como individuo, es decir, la forma en la cual se desplazan de manera automática por el entorno.

El siguiente paso es el desarrollo del programa de acciones individuales, es decir, establecer las tareas planeadas que debe llevar a cabo según lo demande el estado en el que se encuentra.

Por último, se integran todos los demás agentes que están involucrados en el proceso (los cuales en la mayoría de los casos tienen capacidades similares), es decir, un programa general de organización. Para este paso es necesario el desarrollo de un programa de coordinación de los movimientos, las decisiones, las acciones y las interacciones de los agentes mediante un protocolo que controle su continuidad y funcionabilidad (máquina de estados).

Un sistema multi-agente está compuesto por varios módulos principales que expresan las relaciones existentes entre los sub-sistemas del sistema. La funcionalidad y el orden del comportamiento de cada agente en el sistema viene ligado a la definición de los siguientes modelos que juntos generan un protocolo para el ciclo de operación al resolver una tarea en un entorno multi-agente.

“Modelo de agente

Características del agente como capacidad de razonamiento, destrezas (sensores, actuadores), servicios, objetivos, etc.

Modelo de tareas

Describe los objetivos o tareas que llevan a cabo los agentes y la descomposición de dichas tareas.

Modelo de experiencia

Conocimiento necesario o interior necesaria para llevar a cabo las tareas.

Modelo de coordinación

La información ocurrente o pasada de la comunicación permanente entre agentes, sus interacciones y protocolos.

Modelo de organización

Organización de la sociedad del sistema multi-agente en el entorno por medio de jerarquías y las relaciones de los agentes con el entorno.

Modelo de comunicación

Desarrollo de interfaces de usuario. Interacción con humanos u otros agentes.” (Fernández, 1998)

4.6 INTELIGENCIA ARTIFICIAL Y COOPERACIÓN

En robótica cada agente cumple una función determinada dentro de un sistema completo, por lo que hay que tener muy en cuenta la forma en que se produce la cooperación dentro del sistema; es centralizada cuando las acciones están coordinadas por un módulo mando, y es descentralizada cuando la cooperación se realiza directamente entre los agentes.

La Inteligencia Artificial (IA) es la ciencia encargada de diseñar y construir agentes (robóticos o software). Es una de las ciencias más recientes en el mundo. Como herramienta para resolver problemas de optimización, se proponen ambientes cooperativos para disminuir el tiempo de búsqueda de una solución ante un problema, y así nace la IA paralela y distribuida. Cooperando se pretende fundamentalmente aumentar la eficiencia o la eficacia de un algoritmo de búsqueda.

4.6.1 IA PARALELA

El paralelismo surge como una necesidad de lograr un mayor poder computacional en una sociedad tecnológica cada vez más abrumada por la ingente cantidad de operaciones necesarias para el análisis de la realidad que la envuelve, donde la cantidad de información y de acciones involucradas en el problema provocan tiempos de cálculos computacionales considerables.

Paralelismo del Tipo 1 o de Bajo Nivel: La fuente de paralelismo se encuentra normalmente dentro de una iteración del método de búsqueda. El paralelismo tipo 1 se emplea para reducir el tiempo de cómputo.

Paralelismo del Tipo 2 o Descomposición del dominio: Se obtiene al descomponerla región factible en subregiones disjuntas. Esta estrategia se implementa generalmente en un modelo maestro-esclavo, en el que el proceso maestro divide las variables de decisión.

La paralelización de un único camino se emplea, principalmente, para aumentar la velocidad del recorrido secuencial al explorar eficientemente el espacio de búsqueda. La paralelización de múltiples caminos se caracteriza por la investigación en paralelo de múltiples trayectorias, cada una en un procesador diferente. (www.redalyc.org)

4.6.2 IA DISTRIBUIDA

Se centra en los comportamientos inteligentes colectivos que son producto de la cooperación de diversas entidades denominadas agentes.

“Resolución Cooperativa de Problemas Distribuidos (DPS) donde se estudia la manera de cómo un conjunto de módulos cooperan para compartir y dividir el conocimiento de un problema y desarrollo de la solución. Los Sistemas Multi-Agente (MAS) estudian la coordinación de la conducta inteligente entre un grupo de agentes autónomos. La principal diferencia entre ambas áreas estriba en la flexibilidad de la coordinación entre los agentes. En la DPS, las interacciones y tareas que cada agente realiza están prefijadas de antemano: hay un plan centralizado de resolución del problema. En contraposición en los MAS, los agentes tienen un grado de autonomía mayor y pueden decidir dinámicamente qué interacciones son adecuadas, qué tareas deben realizar, quién realiza cada tarea y, además, incluso los agentes pueden mantener objetivos globales diferentes” (Fernández, 1998).

4.7 MÁQUINA DE ESTADOS

“A partir de circuitos combinacionales se pueden generar circuitos de lógica secuencial como una alternativa para resolver problemas implementando electrónica digital, donde las salidas dependen no sólo del estado presente del circuito sino también de las entradas que hubo en el pasado” (Bishop, 2011)

Una máquina de estados es una estructura que permite organizar el comportamiento y/o acciones de un sistema de resolución de problemas al organizar su funcionamiento, continuidad y correspondencia mediante una la secuencia lógica de operaciones. Comúnmente usados para implementar circuitos secuenciales.

"Los circuitos secuenciales pueden clasificarse en dos tipos: (1) aquellos en los que la salida dependen únicamente del estado interno actual (denominados circuitos de Moore) y (2) aquellos en los que la salida dependen tanto del estado actual como de la entrada (denominados de Mealy)" (Floyd, 2006).

El almacenamiento de datos es importante en los sistemas que llevan a cabo procesos basados en estados ya que deben almacenar información para encontrar soluciones ante los problemas. En una máquina de estados se define cada uno de los estados por los que atraviesa el robot para establecer las acciones que se deben ejecutar dependiendo del estado en el que se encuentra. Posteriormente se genera el código que propone una solución al problema.

Dentro de un esquema de máquina de estados encontramos cuatro elementos importantes: el inicio y el final de la operación, los estados y las transiciones entre estados. Las transiciones entre estados vienen dadas según sea circuito de Moore o Mealy, o combinado.

Estado: Representado comúnmente con un círculo en el que se especifica la acción a realizar.

Transición: Representa el flujo de la información entre estados por medio de una flecha.

Inicio: Estado inicial del comportamiento del sistema.

Final: Estado final del sistema.

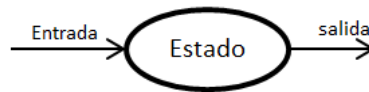


Figura 11. Características de un diagrama de estado.

4.8 TRANSMISIÓN DE DATOS EN SERIE Y EN PARALELO

Los datos binarios, que se representan mediante señales digitales, deben transferirse de un circuito a otro dentro de un sistema digital o desde un sistema a otro, para poder servir a un propósito determinado.

Las máquinas tienen todo tipo de sistemas para procesar señales, pero no todos los sistemas dentro de la máquina son capaces de darle un manejo adecuado a los datos para su debida interpretación, por lo que es necesario moverlos constantemente a un sistema de procesamiento adecuado. Ese movimiento de datos entre sistemas puede ser de dos maneras:

1. Se envían de manera instantánea todos los datos.
2. Se envía de a un dato por cada instante de tiempo.

Dicho tratamiento de las señales es gracias a un micro-controlador encargado de comandar el comportamiento del proceso en general al decidir el tipo de salidas que debe entregar un circuito frente a las entradas.

4.9 RADIO FRECUENCIA RF

Una onda de radio es una señal electromagnética que se propaga por el medio ambiente por medio de antenas de emisión y recepción. Una antena es un dispositivo capaz de convertir señales eléctricas en señales electromagnéticas, y viceversa, por medio de cables conductores o guías de onda para poder ser manipuladas en circuitos de control. “Las señales transmitidas están conformadas por una onda portadora modulada por pulsos. Los pulsos son todos iguales, pero los pulsos de amplitud y los espacios entre ellos pueden variar en longitud. Los pulsos son un código que representa la información que se está realizando” (Bishop, 2011).

Existe un amplio rango de espectro electromagnético. En dicho espectro existe un pequeño intervalo en el que se encuentra la luz visible. En el resto del espectro existen múltiples aplicaciones que se pueden apreciar en la figura 12.

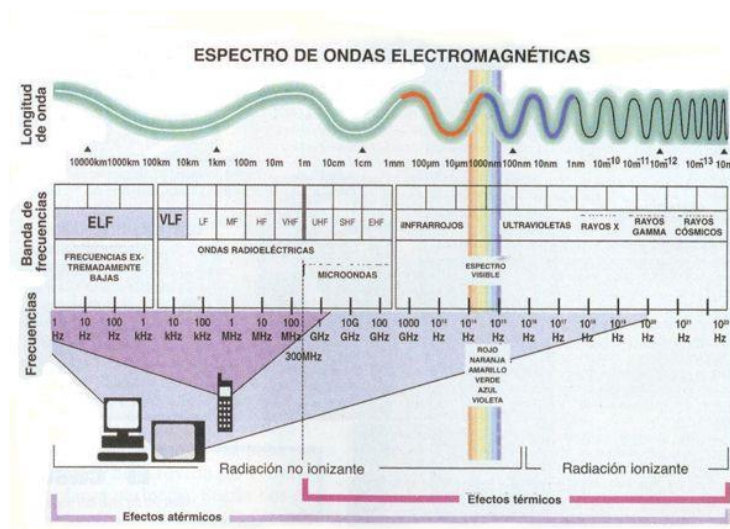


Figura 12. Espectro de ondas electromagnéticas.⁸

⁸ [en línea], disponible en: <http://www.proteccioncivil.org/catalogo/carpeta02/carpeta24/vademecum12/vdm020.htm>, recuperado: 3 de febrero del 2014.

La mayoría de las bandas de frecuencia existentes implican una licencia para poder emitir información en forma de ondas de radio, aunque existen unos intervalos de frecuencia que permiten su uso libre. "El Espectro Electromagnético es un bien público que forma parte del espacio Colombiano, es inajenable e imprescriptible, y está sujeto a la gestión y control del Estado, quien debe garantizar el acceso a su uso en igualdad de oportunidades y en los términos que fije la ley. El Estado puede intervenir por mandato de la ley para garantizar el pluralismo informativo y la competencia y evitar las prácticas monopolísticas en el uso del citado bien. La radio, la televisión, la telefonía, la difusión por cable, el telégrafo, el télex, etc., son algunos de los medios que utilizan el espectro electromagnético para enviar y recibir mensajes, y en general toda clase de datos o información. Por tanto, la libertad de fundar medios masivos de comunicación, haciendo uso de este bien público, también se ve limitada, pues al hacer uso del espectro electromagnético, se tiene que subordinar necesariamente a las normas que lo reglamentan." (Sentencia C-189 de 1994).

La resolución 689 de 21 de abril de 2004 es por la cual el ministerio de comunicaciones de la república de Colombia atribuye unas bandas de frecuencia para su libre utilización dentro del territorio nacional, mediante el acceso inalámbrico y redes inalámbricas de área local, que utilicen tecnologías de espectro ensanchado y modulación digital, de banda ancha y baja potencia, y se dictan otras disposiciones. Se tiene en cuenta que el numeral 32.6 del artículo 32 del Decreto 1972 de 2003 establece que: "El uso del espectro radioeléctrico para aplicaciones industriales, científicas y médicas (ICM), así como para aparatos, equipos o sistemas cuya instalación y operación sean autorizadas de manera general y expresa por el Ministerio de Comunicaciones en las bandas y frecuencias atribuidas nacionalmente para el efecto, es libre".

4.9.1 Aplicaciones ICM [32]

Los equipos para aplicaciones Industriales, Científicas y Médicas ICM (ó ISM: Industrial, Scientific and Medical applications, por sus sigla en inglés), son dispositivos que producen energía de radiofrecuencia (RF), sin embargo, conviene aclarar, que dichos equipos utilizan internamente dicha energía para generar efectos de tipo físico, mecánico, biológico y/o químico. Entre las aplicaciones típicas ICM se encuentran: la calefacción industrial en procesos de manufactura, la diatermia médica, la aceleración de partículas cargadas, los transductores electromecánicos para producir energía mecánica, los limpiadores ultrasónicos de joyería, los humidificadores ultrasónicos, los hornos de microondas, etc. Por lo tanto, los equipos ICM no son equipos de radiocomunicación debido a que no transmiten información de telecomunicación.

5.138	Las bandas:	
	6765-6795 kHz	(frecuencia central 6 780 kHz),
	433,05-434,79 MHz	(frecuencia central 433,92 MHz) en la Región 1, excepto en los países mencionados en el número 5.280 ,
	61-61,5 GHz	(frecuencia central 61,25 GHz),
	122-123 GHz	(frecuencia central 122,5 GHz), y
	244-246 GHz	(frecuencia central 245 GHz)

están designadas para aplicaciones industriales, científicas y médicas (ICM). La utilización de estas bandas para las aplicaciones ICM está sujeta a una autorización especial concedida por la administración interesada de acuerdo con las otras administraciones cuyos servicios de radiocomunicación puedan resultar afectados. Al aplicar esta disposición, las administraciones tendrán debidamente en cuenta las últimas Recomendaciones UIT-R pertinentes.

Figura 13. Atribución de bandas de frecuencias libres.⁹

⁹ [en línea], disponible en: http://www.itu.int/en/ITU-R/space/AmateurDoc/ARS-ART5_S.pdf, recuperado: 13 de marzo de 2014.

Toda vez que las ICM no son aplicaciones de radiocomunicación o telecomunicación, las diferentes Administraciones (países) permiten el uso libre del espectro radioeléctrico para dichos usos. Por esto, el uso de Hornos microondas no requiere de ningún tipo de licencia o genera pago alguno por el uso del espectro radioeléctrico. Las aplicaciones ICM conviven con otros servicios primarios de radiocomunicación, como: los radares del servicio aeronáutico, los sistemas de radiolocalización y el servicio de radioaficionados. Todas estas formas de utilización del espectro radioeléctrico son protegidas y reglamentadas por las administraciones de los diferentes Estados miembros de la UIT. Las frecuencias o bandas de frecuencias para las aplicaciones Industriales, Científicas y Médicas ICM, atribuidas para la Región Americana, según la Nota S5.150 del Reglamento de Radiocomunicaciones de la Unión Internacional de Telecomunicaciones UIT, fueron incorporadas al Cuadro Nacional de Atribución de bandas de Frecuencias de Colombia y adoptadas por el Decreto 555 de 1998.

4.10 LÓGICA PROGRAMABLE

Todos los elementos y funciones lógicas están disponibles como circuitos integrados (CI) y se han ido posicionando de tal manera en el mercado, que hoy en día son la mejor alternativa a la hora de implementar todo tipo de sistemas electrónicos.

La utilización de máquinas automáticas ha facilitado considerablemente la vida cotidiana al ahorrar tiempo en la ejecución de tareas o acciones. Casi siempre dichas máquinas electromecánicas son muy grandes, ya que entre más acciones sean necesarias para solucionar un problema, mayor será el tamaño de una posible solución; por lo que los CI han significado la solución a este problema en implementación de dispositivos electrónicos cada vez más pequeños y potentes.

Los agentes robóticos de este proyecto serán implementados a partir de sentencias lógicas, reglas de control y de elementos electrónicos digitales y analógicos, por lo que hay un indicio de que el tamaño del robot será bastante significativo. Gracias a las diferentes tecnologías de encapsulado y de tarjetas impresas, ha disminuido considerablemente el tamaño de los circuitos impresos que se implementan hoy en día para dar solución a una problemática específica mediante la instrumentación mecánica y electrónica, y muy seguramente el tamaño de los circuitos impresos disminuirá aún más a medida que pasan los años.

De acuerdo al tamaño y características de conexión los CI se clasifican en dos tipos. Tipo paquete en línea dual DIP (Dual Inline Package) es el encapsulado de inserción más común; Tecnología de montaje superficial STM (Surface-Mount Technology) que permite el ahorro de espacio en el circuito impreso de la aplicación, ya que es mucho más pequeño que uno DIP.

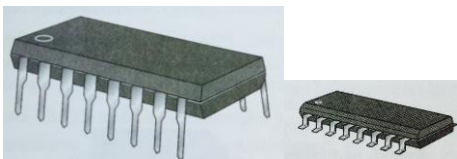


Figura 14. Encapsulados de circuitos integrados DIP y SMT respectivamente. (Floyd, 2006)

4.10.1 Dispositivos lógicos programables PLD

Un dispositivo PLD es la integración de miles de CI en una sola tarjeta que permite el manejo de señales por medio de hardware y software para su debido control. Estos dispositivos pueden programarse a partir de sentencias lógicas para ejecutar tareas específicas.

“Puede pensarse en un PLD como un circuito en blanco en el que se va a implementar un circuito o sistema específico utilizando un determinado proceso. Con la lógica programable los diseños se pueden modificar fácilmente sin tener que re-cablear o reemplazar componentes y se pueden implementar más rápidamente y con menos coste” (Floyd, 2006).

4.10.2 Programación dentro del sistema (ISP)

“Los PLD pueden programarse después de haberse incluido en la placa de circuito impreso de un sistema. También, si es necesario incorporar un cambio en el diseño, el dispositivo ya montado en la placa puede reconfigurarse para incluir las modificaciones al diseño. Procesador integrado es otra técnica de programación dentro del sistema con una memoria y un microprocesador integrado” (Floyd, 2006).

5. DESARROLLO INGENIERIL

5.1 OPERACIÓN DEL SISTEMA MULTI-AGENTE ROBÓTICO MÓVIL

La fase inicial en el desarrollo de un sistema con agentes para resolver un problema dado es el análisis a los requerimientos del cliente o del usuario, ya que a partir de dichos requerimientos es posible plantear un sistema de robots móviles que se organizan para solucionarlo.

Un cliente podría plantear la siguiente situación: Se requiere automatizar la distribución, tratamiento y atención de los pedidos dentro una compañía en la que los sectores de producción y ventas están distanciados geográficamente, y debido al incremento en la demanda de sus productos ha aumentado considerablemente el flujo de mercancía entre estas dos áreas. En cualquier momento puede llegar un nuevo pedido para transportar la mercancía del sector de producción a la sala de ventas pero ocurre en diferentes lapsos de tiempo. Es indispensable que el cliente reciba su pedido en el menor tiempo posible. Debido al tamaño de los productos, la mercancía debe ser transportada por un vehículo capacitado para su transporte y entrega de manera ágil y eficaz.

Un sistema de agentes robóticos para transportar los suministros de manera automática de un área a otra a partir de una indicación humana (vendedor) sería una solución de automatización. Se propone un sistema de 2 robots móviles dentro de la empresa, denominados R1 y R2, capaces de comunicarse y desplazarse de manera autónoma por una ruta de referencia, mostrada en la figura 15, que une las dos áreas de interés.

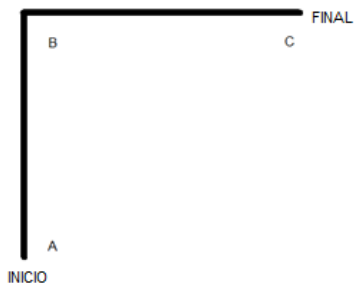


Figura 15. Ruta de referencia.

El inicio es la zona de producción; el final es la zona de ventas. R1 inicia el ciclo en el punto B esperando indicaciones por parte del vendedor para recoger un objeto en el punto A; R2 está inicialmente en el punto C esperando indicaciones de R1 para recoger el objeto y llevarlo al final del recorrido para ser recibido por el vendedor. La idea es acarrear uno o más objetos desde el punto A al punto C mediante la participación e interacción tanto de R1 como R2 para automatizar el transporte de los pedidos. El recorrido del objeto desde el punto A hasta el C se le denomina un ciclo de operación.

El entorno de trabajo de los agentes debe tener ciertas características propias para que éstos puedan desenvolverse de la mejor manera. En realidad, las acciones que llevan a cabo los agentes son la solución al problema, por lo que tareas simples, secuenciales, de selección e iterativas colocan al entorno en un estado final deseado gracias a las acciones de los robots.

5.2. ENTORNO

“Hay que centrarse primero en las características del entorno de trabajo a la hora de diseñar un sistema de agentes robóticos antes de especificar los aspectos propios en sí del agente, ya que los agentes

son esencialmente la solución al problema que se presenta al entorno determinado” (Russell, S. y Norvig, P., 2004).

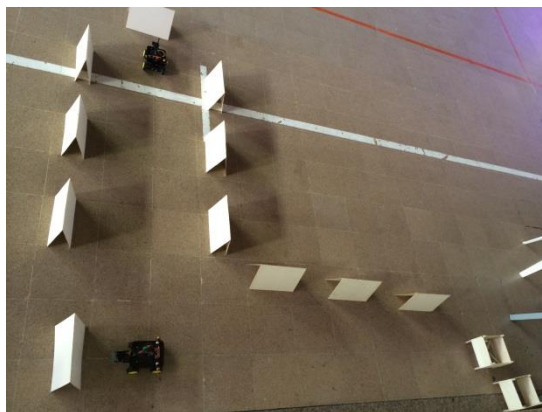


Figura 16. Entorno de trabajo de los robots.

El entorno tendrá fronteras que delimitan el mundo de los agentes para evitar confusiones en sus inferencias de movimiento. Son necesarios unos sensores encargados enteramente del direccionamiento del agente por la ruta de referencia (control lateral). Son necesarios unos sensores encargados en la identificación del objeto de interés (control frontal).

De acuerdo a las características del entorno de trabajo se puede decir que el entorno de los agentes es parcialmente observable, ya que a los agentes les basta con conocer su estado interno para determinar la acción que deben ejecutar; los sensores no reciben mayor información por parte del sistema en cuanto al estado del entorno. El entorno es determinista y secuencial, pues el siguiente estado de los robots depende totalmente del estado actual, ya que o carga o descarga el objeto de acuerdo al tiempo transcurrido y a la transición entre estados. El entorno es estático, ya que los robots no deben estar pendientes a cambios en el entorno mientras actúan pues los objetos aparecen en el inicio y desaparecen en el final al ser; a los agentes no les interesa cómo fue que llegaron allí los objetos, o qué pasa con los objetos al ser entregados. El entorno es continuo, ya que el ciclo de operación se maneja a medida que transcurre el tiempo y las

transiciones en las lecturas de los sensores son consecuentes a medida que los robots desarrollan sus acciones. Por último, el entorno es multi-agente pues hay dos robots puestos en él.

5.3 ELEMENTOS Y CARACTERÍSTICAS DEL AGENTE ROBÓTICO

Ya que se busca que el agente interactúe con objetos del mundo y los manipule, el robot será un híbrido-transportador que contará con algún tipo de manipulador; es decir, un efector que actúe a partir de la información que recibe de los sensores para recoger un objeto. Será un robot multifuncional de secuencia fija, ya que se estudian y definen los estados por los que debe atravesar. Será un robot tipo B, ya que su desplazamiento y acciones son automáticas reguladas por una referencia de ruta, y el control se efectuará por medio de Controladores Lógicos Programables.

El diseño del sistema multi-agente empieza especificando las características del problema general para cada agente como individuo, como se muestra en la tabla 1. Es aquí donde se enmarcan las capacidades de cada agente frente al entorno de trabajo y a las tareas que debe ejecutar en un ciclo de operación.

Tipo de agente	Conductor, comunicador y transportador.
Entorno	Terreno llano, paredes y objetos objetivo.
Actuadores	Motores, ruedas, servomotores y pinzas u otros.
Sensores	Infrarrojos y ultrasónicos.

Tabla 1. Características del sistema artificial.

A partir de la tabla 1 se enumeran las cualidades de cada robot para desenvolverse de la mejor manera en un ciclo cooperativo:

- Navegación por el entorno con controles autónomos.
- Regulación de velocidad para su desplazamiento.
- Sistema de comunicaciones para los acontecimientos importantes del proceso.
- Mecanismo de interacción para manipular los objetos del entorno.
- Cierta nivel de abstracción de la información recibida relevante para el proceso.

5.3.1 Modelo del agente

El éxito en la operación de un agente robótico viene ligado al programa que ejecuta; cada estado en el que se encuentra debe estar bien definido en el programa de control para poder encontrar una acción pertinente a llevar a cabo.

La figura 10, que representa la composición de un agente, puede ampliarse como se muestra en la figura 17, donde el cuadro en interrogación se convierte en reglas de control. Los sensores y actuadores son parte de la arquitectura, y los cuadros son las sentencias lógicas del programa de control.

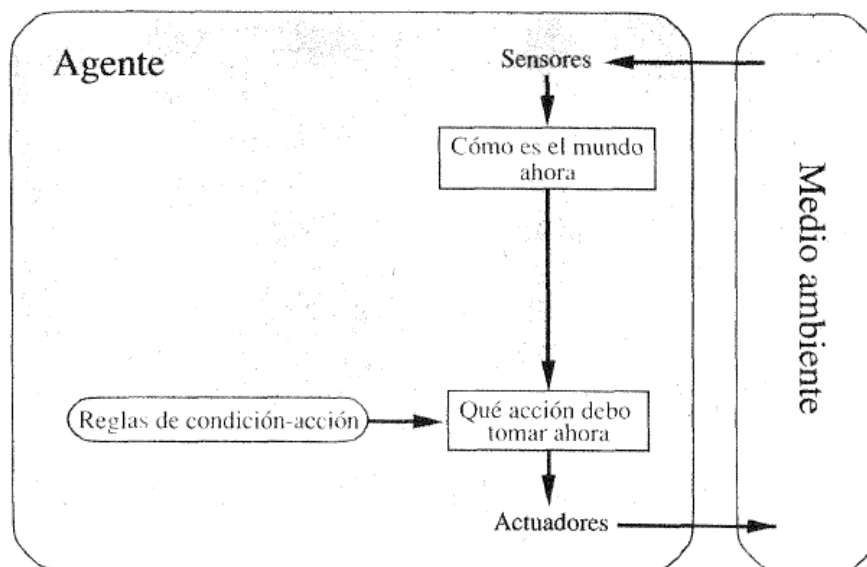


Figura 17. Diagrama de flujo de un agente reactivo simple.¹⁰

A partir de las características de los sensores y los actuadores, y gracias a las reglas de control, los robots logran los procesos de movimiento, dirección e interacción con los objetos y con otros agentes, mediante sentencias lógicas que luego son codificadas en algún lenguaje de programación.

“Al conjunto de dichas reglas de movimiento se le denomina programa agente, el cual proyecta las percepciones en acciones” (Russell, S. y Norvig, P., 2004). Las reglas básicas de control para un agente robótico dependen de la procedencia de las entradas, pues así se determina la acción a llevar a cabo.

Cuatro motores **M1, M2, M3 y M4**.

Dos sensores ultrasónicos **U1 (delantero) y U2 (trasero)**.

Dos sensores infrarrojos **IR1 (derecho) y IR2 (izquierdo)**.

Manipulador **MA**.

¹⁰ Russell y Norvig, Inteligencia Artificial un enfoque moderno, 2ª edición, 2004, p. 54.

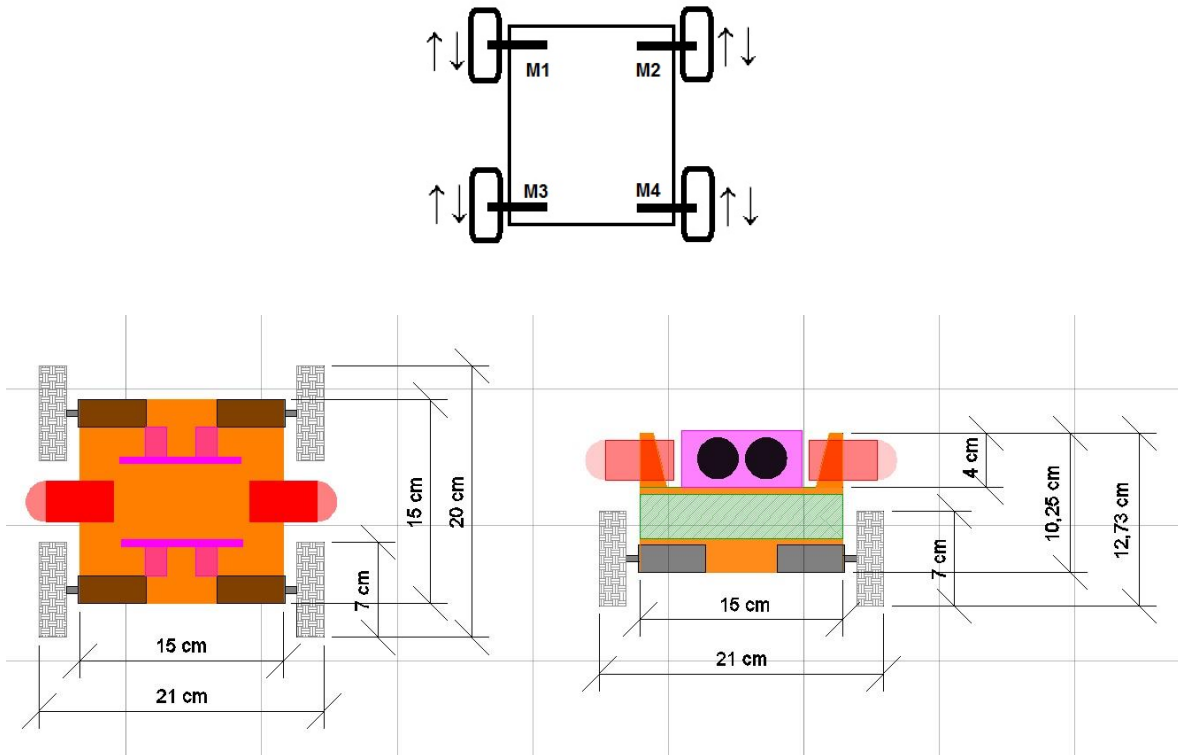


Figura 18. Primer diseño de la arquitectura.

5.3.2. Modelo de tareas y experiencia

“Un agente puede utilizar sus percepciones (...) en un enfoque donde detectan un pequeño número de características en su entrada sensorial y las pasan directamente a su programa agente que puede responder de forma reactiva a ellas” (Russell, S. y Norvig, P., 2004), dando así el primer paso a un tipo de control reactivo; pero lo que se quiere lograr con el comportamiento de los agentes va un poco más allá.

Los agentes deben contar con la capacidad de moverse para evitar choques. Además con la habilidad suficiente para poder solucionar un problema o realizar una tarea gracias a dicho desplazamiento, por lo que el sólo hecho de moverse no los convierte agentes resolventes de problemas. Se trata de definir dichos

movimientos y acciones en un estado que contiene las tareas a realizar, para así resolver el problema proponiendo un método organizado para su funcionamiento.

Para su comportamiento los robots utilizan los sensores IR y U como se muestra en la figura 19. Cuando se desplaza hacia adelante, U1 le dirá la distancia del objeto a recoger, se establece una referencia adelante US para recoger el objeto, U2 está inactivo; hacia atrás, U2 le dirá la distancia al punto de entrega del objeto, se establece una referencia atrás US para entregar el objeto. En el movimiento hacia adelante, si la referencia al objeto se cumple, recoge el objeto. Hacia atrás, si la referencia al punto de encuentro se cumple, entrega objeto.

Para mantenerse en ruta hacia el objeto cada IR tiene una referencia de movimiento (IReferencia Izq., IReferencia Der.), si se sobrepasa dicha referencia se harán las correcciones de movimiento respectivas. Si se activa el sensor izquierdo, hacer corrección a la derecha; si se activa el sensor derecho, hacer la corrección a la izquierda.

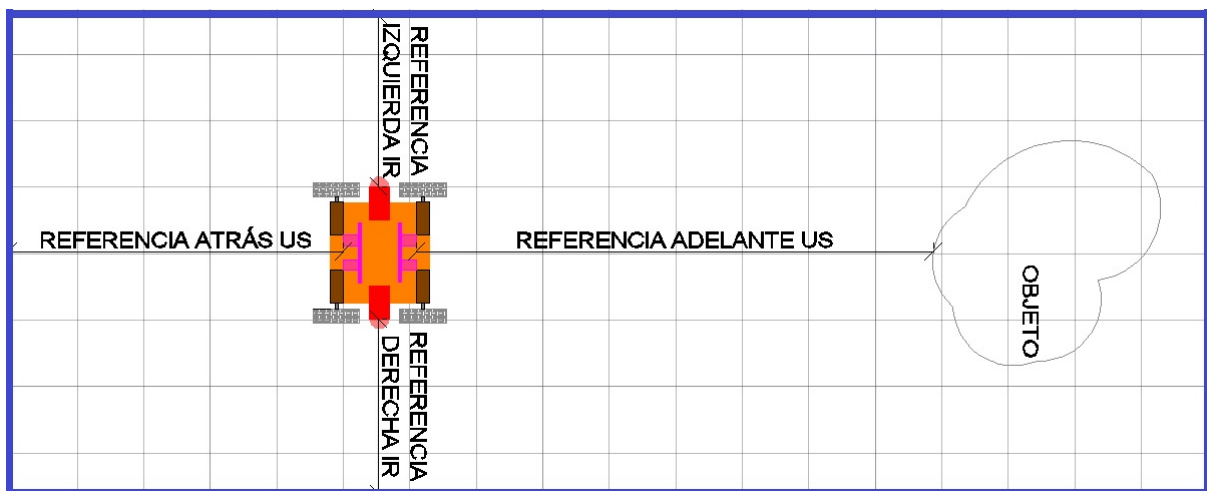


Figura 19. Referencias de movimiento y acción.

Para la interacción con los objetos se piensa en construir las figuras hechas por materiales livianos pero a la vez resistentes para evitar deformaciones a la hora de ser manipuladas. Sea cual sea el

mecanismo para la recolección, los robots simplemente cargan o descargan según lo demande la situación de manera coordinada.

Tareas por estado: Una solución a un problema de agentes es un camino de estados desde el inicial a uno objetivo [3], para así poder completar todos los estados. Se realiza una descripción de cada estado de tal manera que exprese e informe sólo los aspectos relevantes para realizar su acción.

Estado 1: Inicio (R1: Orden humana. R2: Orden de R1)

Descripción del estado 1: El robot se encuentra inmóvil, libre de todo obstáculo o referencia.

Espera la orden para empezar a funcionar.

Estado 2: Enganchar

Descripción del estado 2: El robot se desplaza buscando el objeto de interés. Al momento de

detectarlo se detiene.

Estado 3: Levantar objeto

Descripción del estado 3: El robot recoge el objeto.

Estado 4: Entregar

Descripción del estado 4: El robot se desplaza buscando el punto de entrega, descarga el objeto.

R1 al descargar el objeto envía a R2 la orden de arranque.

Estado 5: Enganchar

Descripción del estado 5: El robot entra nuevamente al estado 2. Se repite el ciclo hasta cuando una indicación humana lo ordene (R1 recibe la orden de pare y la comunica a R2).

Estado 6: Fin. En la figura 24 se muestra la secuencia de acciones en una máquina de estados.

5.3.3 Lazos de control por tarea (Sub-sistemas)

Las tareas que realiza el robot son acciones que se llevan a cabo por medio de un sistema completo de entradas, operaciones y salidas. Cada agente tiene 4 sub-sistemas de control que formaran posteriormente el sistema multi-agente completo. En la figura 20 se muestra el lazo de control para el sub-sistema de locomoción.

5.3.3.1 Sub-sistema de locomoción

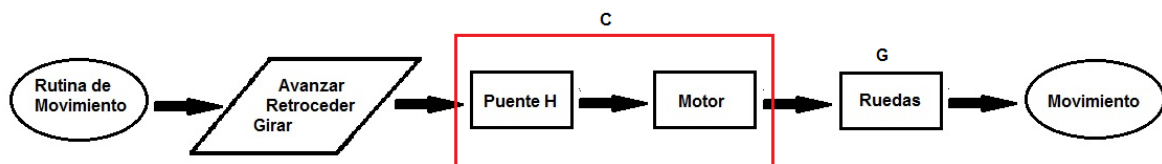


Figura 20. Lazo de control del sistema de locomoción.

El sistema de locomoción es el sub-sistema que otorga las funciones básicas de desplazamiento del robot. El circuito puente H y los motores pueden integrarse como uno solo, denominado C. Al entrar en la rutina de movimiento el agente tiene las siguientes opciones a realizar: movimientos hacia adelante, hacia atrás o giros.

MOVIMIENTOS	M1	M2	M3	M4
DERECHA	↑	↓	↑	↓
IZQUIERDA	↓	↑	↓	↑
ADELANTE	↑	↑	↑	↑
ATRÁS	↓	↓	↓	↓

5.3.3.2 Sub-sistema de sensores

En la figura 21 se observa el sub-sistema de sensores: C (motor), G (ruedas) y S (sensor). Este lazo es útil para realizar tanto el control con sensores infrarrojos como el control con sensores ultrasónicos.

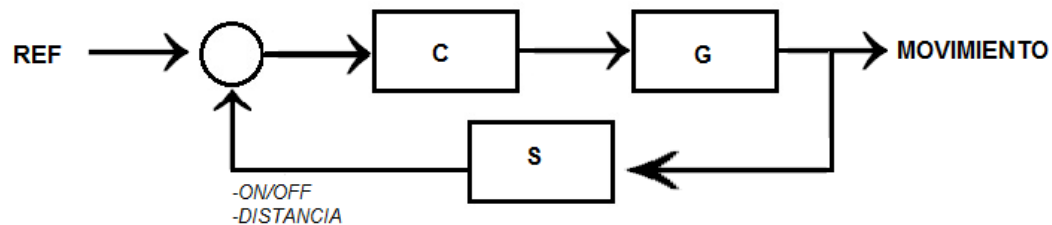


Figura 21. Lazo de control del sistema de sensores.

Los sensores infrarrojos serán los encargados de adquirir la información para el movimiento del robot por una ruta de referencia. El sensor infrarrojo detecta los obstáculos activándose y desactivándose. Una entrada ON indica que hay un obstáculo, por lo que debe tomar medidas; una entrada OFF indica que el camino está libre y puede desplazarse sin chocar.

Los sensores ultrasónicos serán los encargados de adquirir la información para el reconocimiento del objeto de interés. El sensor detecta los objetos y entrega una medición de DISTANCIA. De acuerdo a la distancia a la que se encuentre el objeto, se regula el movimiento para las acciones de carga y descarga.

ENTRADA INFRARROJOS	SALIDA
Si IRizq=ON y IRder=OFF	MOVERSE DERECHA
Si IRder=ON y IRizq=OFF	MOVERSE IZQUIERDA
Si IRder=OFF y IRizq=OFF	MOVERSE ADELANTE

ENTRADA ULTRASÓNICO	SALIDA
Si US1 está operando y Distancia \geq REF	MOVERSE ADELANTE PARA BUSCAR OBJETO
Si US2 está operando y Distancia \geq REF	MOVERSE ATRÁS PARA BUSCAR PUNTO DE ENTREGA

5.3.3.3 Sub-sistema de recolección

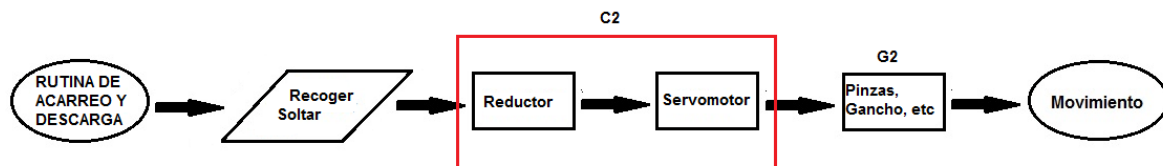


Figura 22. Lazo de control del sistema de recolección.

El sub-sistema de interacción es aquel que permite la manipulación de los objetos de interés. Cuando se entra en la rutina de carga y/o descarga, se elige entre las acciones de recoger o de soltar el objeto, que se lleva a cabo por medio de acciones de pinzas, adhesivos o ganchos.

ENTRADA ULTRASÓNICO	SALIDA
Si US1 está operando y Distancia=REF	MOVER GANCHO PARA RECOGER OBJETO
Si US2 está operando y Distancia=REF	MOVER GANCHO PARA DESCARGAR OBJETO

5.3.3.4 Sub-sistema de comunicaciones

El sub-sistema de comunicaciones empieza cuando una indicación humana da la orden a R1, y para R2 empieza con la orden de R1 al momento de entregar el objeto. Este subsistema establece un pequeño protocolo que comanda el envío y recepción de mensajes.

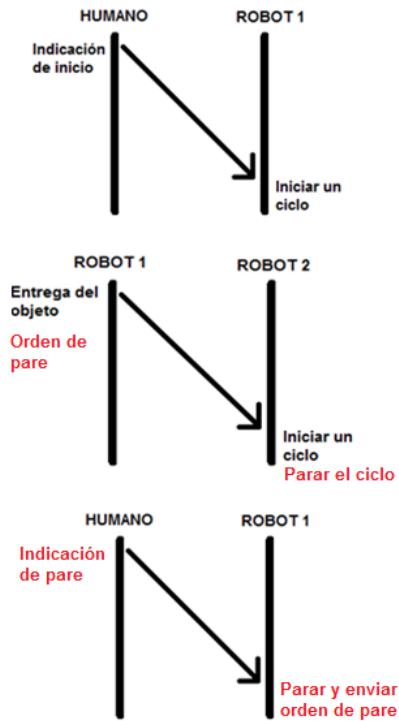


Figura 23. Protocolo de transmisión de datos. (a) Estado: inicio. (b) Estado: Entregar/Estado fin de R1 a R2. (c) Estado fin de H a R1.

5.4 MODELO DE COORDINACIÓN AGENTE-AGENTE

Es aquí donde se enmarca la relación existente entre los agentes de acuerdo al estado. Los agentes deben comunicarse mediante mensajes organizados. Dichos mensajes pueden ser vistos como actos de habla en los que a partir de las señales recibidas del entorno se coordinan los movimientos del ciclo. En el Anexo A se muestran los diagramas de flujo propuestos para la ejecución de las tareas de cada agente; el Anexo A.1. es del robot número 1, el Anexo A.2. es del robot 2. Tan pronto como R1 reciba la indicación mediante control remoto empieza a atravesar todos los estados de la máquina de estados. Las transiciones entre estados definen el protocolo para el inicio, el transcurrir y el final del ciclo. Así un agente conoce el

estado en el que se encuentra para poder ejecutar una acción en el momento propicio. En la figura 24 se muestra la primera máquina de estados que se propone a partir de los diagramas de flujo.

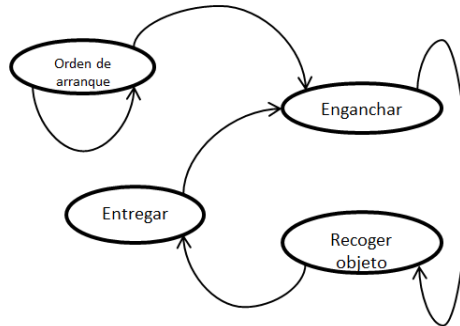


Figura 24. Primera máquina de estados.

La máquina de estados es la misma tanto para R1 como para R2, la única diferencia radica en el estado Orden de arranque. Para R1, la orden de arranque viene dada por una indicación humana; para R2 la orden de arranque proviene de R1 al entregar el objeto. En el estado Enganchar los robots se desplazan hacia el objeto. En el estado Recoger objeto, se posicionan para moverlo. En el estado Entregar, se desplazan al punto de donde partieron para descargarlo y volver nuevamente al estado Enganchar.

De acuerdo a la máquina de estados, se organiza el protocolo que comanda la secuencia de las acciones en el ciclo de operación para el sistema multi-agente. En la figura 25 se muestra el diagrama de flujo para controlar los robots.

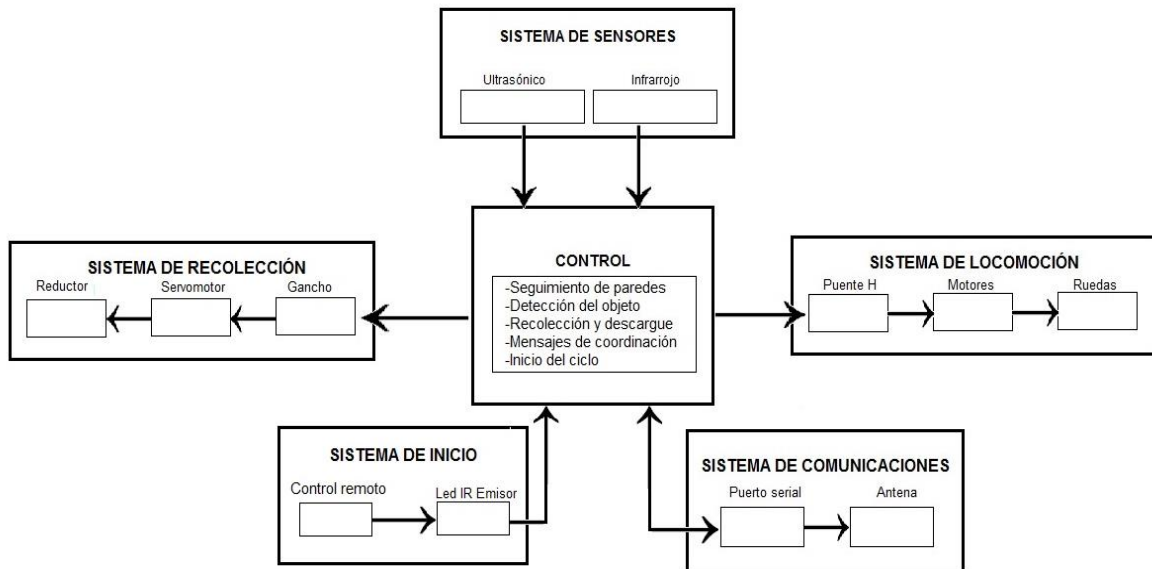


Figura 25. Organización de los sub-sistemas para formar el sistema multi-agente.

5.5 ELEMENTOS ELECTRÓNICOS

Ahora se especifican las características de los elementos elegidos para la implementación de los sensores, los actuadores y el micro-controlador del agente.

5.5.1 Motor de corriente directa

“Dentro de la gran variedad de tipos existentes en el mercado, los más económicos son los que se utilizan en algunos juguetes. Su número de revoluciones por minuto (RPM) es muy elevado, lo que los hace muy apropiados para la construcción de un microbot que por ejemplo, siga una línea, si no se utilizan reductores adicionales o un sistema de regulación electrónico” (www.serverpruebas.com.ar).

“Los aspectos que podemos controlar en un motor son: velocidad y dirección. La velocidad la podemos manipular por medio de PWM. Escogidos por su revolución continua. Aunque necesitan un circuito de realimentación para aplicaciones de control, son muy fáciles de controlar, ya que tienen sólo dos pines de control. Normalmente funcionan con voltajes bajos, tales como 6 V hasta incluso tan sólo con 1,5 V. Por lo general requieren unos cientos de miliamperios para funcionar. Los motores se deben conectar, ya sea con un transistor de potencia o un relé para ser controlado. Esto permite que el flujo de corriente a través del motor se invierta” (www.serverpruebas.com.ar) al manipular el sentido en la polarización aplicada con un circuito de conmutación denominado puente H, que permite activar la dirección del flujo de corriente para generar giro dual del motor.

Por lo que se elige un motor de corriente continua con caja de giro. Estos motores son baratos, pequeños, fáciles de instalar y controlar, ideales para el uso de un robot móvil. (Ver anexo B, figura B.1.)

5.5.2 Servomotor

Un servomotor es un dispositivo que consta de un sistema de piñones conectado a un potenciómetro que permite controlar la posición en rotación de un motor mediante un ángulo medido en grados que va desde 0° a 180° . Utilizados en robots articulados debido a su ángulo de giro. El control de posición es interno y fácilmente controlable por medio de micro-controladores. Reducido coste. En los juguetes del tipo Mecano o Lego, podemos encontrar motores con reductores o sistemas reductores para acoplar a los motores.

“Se trata de unos motores con un circuito electrónico. Estos servomotores cumplen una características que los hacen idóneos para la construcción de un microbot, tales como un buen par de salida, potencia suficiente para trasladar objetos o una batería, baja inercia, son capaces de mover 3,5 Kg

x cm, incluyen multitud de accesorios para poder fijar las ruedas del microbot, son fáciles de fijar a una estructura” (www.serverpruebas.com.ar). La posición deseada se especifica mediante las asignaciones que se hacen a las duraciones de un pulso en un tren de pulsos; entre menor sea la duración del pulso, menor será el ángulo de posición del servomotor.

Sus tres pines son: línea de alimentación, línea de control y la línea de tierra. Por la línea de control recibe la información que permite el posicionamiento a gusto del servo, el cual depende del ancho del tren de pulsos que reciben. (Ver anexo B, figura B.2.)

5.5.3 Sensor ultrasónico

“Sensor que permite calcular la distancia a un obstáculo midiendo el tiempo de retorno de un pulso auditivo emitido. El ultrasonido es de sonido de alta frecuencia, demasiado alto para ser audible para el oído humano. Por lo general, los transmisores y receptores ultrasónicos operan a 40 kHz” (Bishop, 2011). Mediante un cálculo simple que involucra la velocidad del sonido y el tiempo transcurrido desde la emisión del pulso hasta la recepción del eco, se puede conocer la distancia al objeto.

El sensor SRF235 es un medidor de distancias por ultrasonidos de alta frecuencia y haz estrecho que posee un procesador de comunicaciones I2C. Su principal utilidad es en robótica. Este sensor de ultrasonidos se destaca por funcionar a una frecuencia de 235Khz y generar un haz de tan solo 15 grados, lo que proporciona una medida de elevada precisión y gran direccionalidad. Las ondas ultrasonoras de 235KHz no se transmiten tan fácilmente a través del aire como las de 40KHz. Esto hace que el alcance del sensor SRF235 esté limitado aproximadamente a 1 m para objetos normales y un máximo de 1,2 m para superficies extensas” (www.marcombo.com).

El SRF04 es un sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 3 a 300 cm. El sensor funciona por ultrasonidos y contiene toda la electrónica encargada de hacer la medición. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno. De muy pequeño tamaño, SRF04 destaca por su bajo consumo, gran precisión y bajo precio” (www.superrobotica.com).

“El sensor ultrasónico URM37 V3.2 utiliza un procesador AVR nivel industrial como la unidad de procesamiento principal. Viene con un sensor de temperatura que es único en su clase. El sensor tiene tres modos de salida (PWM, RS232 o TTL) lo cual le permite adaptarse más fácilmente a la mayoría de los dispositivos y aplicaciones. El sensor proporciona un pin para controlar la salida de un servo que se puede utilizar para funciones de barrido” (www.daedalus.ei.tum.de), ideal para el posicionamiento a gusto del sensor para detectar los objetos.

El sensor ultrasónico consta de 9 pines: pin 1 es la alimentación, pin 2 es tierra, pin 3 es reset, pin 4 es entrada al PWM, pin 5 es la señal de salida para un servomotor, pin 6 es el comparador de referencia, pin 7 es la línea de control para la activación del sensor, pin 8 es receptor y pin 9 es transmisor. Ver anexo B, figura B.3.

5.5.4 Sensor de luz e infrarrojo

Los dos sensores de luz más comúnmente utilizados son la resistencia dependiente de la luz (LDR) y el fotodiodo. Un resistor dependiente de la luz consiste en un disco de material semiconductor de dos electrodos en su superficie. Esto quiere decir que es un mal conductor de la corriente eléctrica. Su resistencia es alta. A la luz, más electrones se escapan de los átomos del semiconductor. Hay más electrones para llevar carga eléctrica. Tienen 2 pines: alimentación y tierra.

“Un sensor infrarrojo (o reflectivo) funciona a partir de la emisión y la recepción de luz. La luz emitida por un led debe reflejarse en el momento en que exista un objeto enfrente, y un sensor de luz o fotodiodo que se activa cuando hay reflexión. El sensor de efecto reflectivo H21A1 consta de un led infra-rojo y de un fototransistor, es usado como un sistema de detección de objetos, lectura de encoders y en aplicaciones de robótica, entre otros. Para esta figura lo que se desea obtener es el rebote que se produce cuando el led emite el haz infra-rojo y lo detecte el fototransistor, el cuál operará en las regiones de corte y saturación; esta señal se adecuará para obtener un nivel lógico” (www.profesores.fi-b.unam.mx).

El sensor SHARP IS471F es inmune a interferencias de luz normal. Este sensor incorpora un modulador/demodulador integrado en su carcasa y a través de su pin 4 controla un diodo LED de infrarrojos externo, modulando la señal que este emitirá, para ser captada por el IS471F que contiene el receptor. Cuando un objeto se sitúa enfrente del conjunto emisor/receptor parte de la luz emitida es reflejada y demodulada para activar la salida en el pin 2 que pasará a nivel bajo si la señal captada es suficientemente fuerte” (www.x-robotics.com).

El sensor E18-D80NK es un interruptor infrarrojo de distancia de rango ajustable entre 3 a 80 cm. Es pequeño, fácil de usar y de bajo costo. Útil para robots, medios interactivos, líneas industriales de ensamblaje, etc. Se eligieron debido a su rango ajustable. (Ver anexo B, figura B.4.)

5.5.5 Control remoto

En un control remoto infrarrojo los botones tienen en su parte posterior un material que conduce la electricidad. Cuando se presiona el botón, este material hace contacto con una plaqueta y cierra el circuito que corresponde al botón. Un pequeño circuito integrado reconoce la señal y determina qué botón fue presionado; con base a esa información envía una señal al resonador de cuarzo (cristal); éste la devuelve con una frecuencia determinada. Ese impulso es transmitido a un LED que lo envía convertido en radiación infrarroja. El receptor (por ejemplo, un televisor) puede reconocer el botón pulsado midiendo la frecuencia de la radiación” (www.taringa.net/posts/ciencia-educacion/). Ver anexo B, figura B.5.)

5.5.6 Módulo de comunicaciones

Para la comunicación en serie de robots móviles se requieren módulos de comunicaciones inalámbricos. Los módulos ya vienen listos con pines de tipo macho para insertarse en un circuito integrado. El corazón de estos módulos son los circuitos integrados, ya que estos sólo requieren recibir del micro-controlador los datos de control por comunicación serial para emitir una salida. “La mayoría de los micro-controladores de gama media contienen miles de compuertas. Los compiladores de lenguajes de alto nivel como son 'C' o PASCAL cuentan con librerías para el manejo del puerto inalámbrico y hacen muy sencillo el envío y recepción de datos” (www.aulati.net).

Para compartir información entre los agentes es conveniente la transmisión y recepción de datos inalámbrica, debido a que son robots móviles que están en constante movimiento. Las comunicaciones inalámbricas se realizan por el aire por medio de radiofrecuencia y por medio de transmisión en puertos seriales. Hay dos posibilidades de módulos de implementación para el intercambio de mensajes wireless: Bluetooth y APC220.



Figura 26. Modulos de radio frecuencia y bluetooth.¹¹

APC220:

- Transmit distance up to 1000m (line of sight) @9600 bps
- 256 bytes data buffer
- High sensitivity (-112dbm @9600 bps)
- GFSK modulation
- UART/TTL interface
- Embedded watch dog
- Size: 37x17x6.5 mm
- RF Frequency: 433 MHz
- Output Power: 20mw
- EN pin: Enabled (if not connected it will be enabled)

BLUETOOTH:

- The Bluetooth chip: CSR BC417143
- Bluetooth protocol: Bluetooth Specification v2.0 + EDR

¹¹ [en línea], disponible en: http://www.dfrobot.com/index.php?route=product/product&filter_name=bluetooth&product_id=360#.UxP72fl5NrU, recuperado: 27 de febrero del 2014.

- USB Protocol: USB v1.1/2.0
- Operating frequency: 2.4 ~ 2.48GHz unlicensed ISM band
- Modulation: GFSK (Gaussian Frequency Shift Keying)
- Transmit Power: $\leq 4\text{dBm}$, Class 2
- Transmission distance: 20 ~ 30m in free space
- Sensitivity: $\leq -84\text{dBm}$ at 0.1% BER
- Transfer rate: Asynchronous: 2.1Mbps (Max) / 160 kbps; Synchronous: 1Mbps/1Mbps
- Support profiles: Bluetooth serial port
- Serial port baud rate: 4800 ~ 1382400 / N / 8 / 1 default: 9600
- Input Voltage: +3.5 V ~ +8 V DC and 3.3V DC/50mA
- Working temperature: $-20\text{ }^{\circ}\text{C}$ ~ $+55\text{ }^{\circ}\text{C}$
- Module Size: $40 \times 20 \times 13\text{mm}$

Se realizan pruebas para la debida implementación de cada módulo al enviar datos de un robot a otro para determinar el tiempo de respuesta y el rango de cubrimiento de la comunicación. Para el módulo APC220 es necesario cambiar la posición del pin ENABLE, ya que tiene que estar desconectado para su debido funcionamiento; para el Bluetooth no hay nada más que hacer que conectarlo para configurarlo y programarlo. Se implementa el APC220 principalmente porque tiene una distancia de transmisión mayor que el Bluetooth. (Ver anexo B, figura B.6.)

5.5.7 Micro-controlador

ATxmega: procesadores muy potentes con 16 a 384 kB de memoria flash programable, encapsulados de 44, 64 y 100 pines (A4, A3, A1), capacidad de DMA, eventos, criptografía y amplio conjunto de periféricos con DACs.

ATmega: microcontroladores AVR grandes con 4 a 256 kB de memoria flash programable, encapsulados de 28 a 100 pines, conjunto de instrucciones extendido (multiplicación y direccionamiento de programas mayores) y amplio conjunto de periféricos.

ATtiny: pequeños microcontroladores AVR con 0,5 a 8 kB de memoria flash programable, encapsulados de 6 a 20 pines y un limitado set de periféricos.

AT90USB: ATmega integrado con controlador USB.

ATmega 328:

- Arquitectura RISC avanzada.
- Ciclos de lectura/escritura: 10,000 Flash/100,000 EEPROM.
- Retención de la información: 20 años a 85°C/100 años a 25°C.
- In-System Programming (ISP)
- Dos contadores de 8-bit con modo de comparación.
- Un contador de 16-bit con modo comparador y de captura.
- 6 canales PWM.
- 8-channel 10-bit ADC.

- 6-channel 10-bit ADC en empaquetado DIP.
- Puerto Serial Programable USART.
- Interruptores internos y externos.
- 23 líneas de E/S Programables.
- Active Mode: 0.2 mA

Gracias a la búsqueda de los dispositivos fue posible encontrar el sistema embebido DFRduino Romeo, el cual es una tarjeta micro-controladora todo en uno, especialmente diseñada para aplicaciones robóticas. El beneficio principal es la integración para el control de 4 motores de corriente directa con el circuito integrado L298SO20 y socket para comunicaciones inalámbricas con los pines necesarios para la utilización de módulos de comunicación inalámbrica: Bluetooth y APC220, lo que facilita su implementación en gran medida. Además, la tarjeta puede ser programada al conectarse directamente con el ordenador cuantas veces sea necesario, disminuyendo el tiempo y las dificultades a la hora de programar el robot, pues tiene el controlador ATmega 328 que es compatible con la plataforma de código abierto Arduino.

5.6 ARQUITECTURA DE HARDWARE

El diseño básico estructural de la figura 18 le otorga la capacidad de desplazarse por un entorno para actuar en él de manera autónoma. Todo compuesto en su mayoría por elementos semiconductores e instrumentos de acción y medición.

Ventajas:

- Una de las ventajas de una arquitectura robótica comercial a un desarrollo desde cero es el precio del proyecto, pues la diferencia de costos y tamaño es importante.
- Para el desarrollo de proyectos de agentes robóticos con más de dos integrantes, es común el desarrollo de una arquitectura robótica propia e
- Los esfuerzos en la creación del sistemas multi-agente deben ir enfocados al desarrollo comportamental del robot como individuo grupal y a la manera como se realizan las interacciones entre robots con el entorno.
- Las arquitecturas comerciales ofrecen un nivel de calidad establecido por el fabricante debido a procesos estructurados en la elaboración de sus partes de ensamblaje.

5.6.1 Ensamble

Las siguientes imágenes de la arquitectura son tomadas de la página web www.dfrobot.com, donde se adquieren las partes de ensamblaje y algunos de los elementos electrónicos.



Figura 27. Partes de la carcasa del robot móvil.

En el anexo C se ilustran los planos del robot y el ensamble de las partes.

5.7 ARQUITECTURA DE SOFTWARE

Comparando las implicaciones entre el desarrollo de una arquitectura robótica propia y una plataforma comercial mostrados en el anexo D, se evidencian ciertos aspectos importantes.

El desarrollo de este proyecto se basa principalmente en el programa que interpreta las condiciones del entorno para luego ser representadas en acciones reales mediante la arquitectura hardware. Los lenguajes de programación, junto con las plataformas de código abierto y las nuevas herramientas computacionales proporcionan nuevas alternativas para afrontar un problema y dar soluciones integrales a tareas específicas mediante sistemas embebidos.

5.7.1 Tarjeta controlador Romeo V1.1

La DFRduino v1.1 Romeo tiene conexiones exclusivas para controlar 4 motores DC, junto con un conjunto de pines digitales y analógicos de E / S. El código de lenguaje C que se utiliza con la tarjeta Romeo tiene extensiones, librerías y otras ayudas para controlar de manera compacta (en cuanto a líneas de código) muchos otros dispositivos electrónicos con el ATmega. Romeo tiene dos fuentes de energía: 7,5 V CC para los motores y 9 VCC para el controlador, y aunque sólo requiere 5 VCC para operar, es insuficiente para trabajar con los motores. Con cable USB se otorgan 5 VDC para permitir la carga del código al micro-controlador.

La tarjeta Romeo cuenta con:

- 14 entradas/salidas digitales.
- 6 canales predeterminados para PWM.
- 8 entradas/salidas análogas de 10 bits.
- Puerto USB.
- ICSP In-Circuit Serial Programming.
- Interface Serial TTL.
- Socket para módulos RF APC220 o DF bluetooth
- 6 pines interface para protocolo I2C.
- Control de dos motores CC en doble sentido, 2A corriente máxima.
- DC alimentación: 7-12 V DC, alimentación USB o externo.
- Salida DC: 5/3.3 DC.

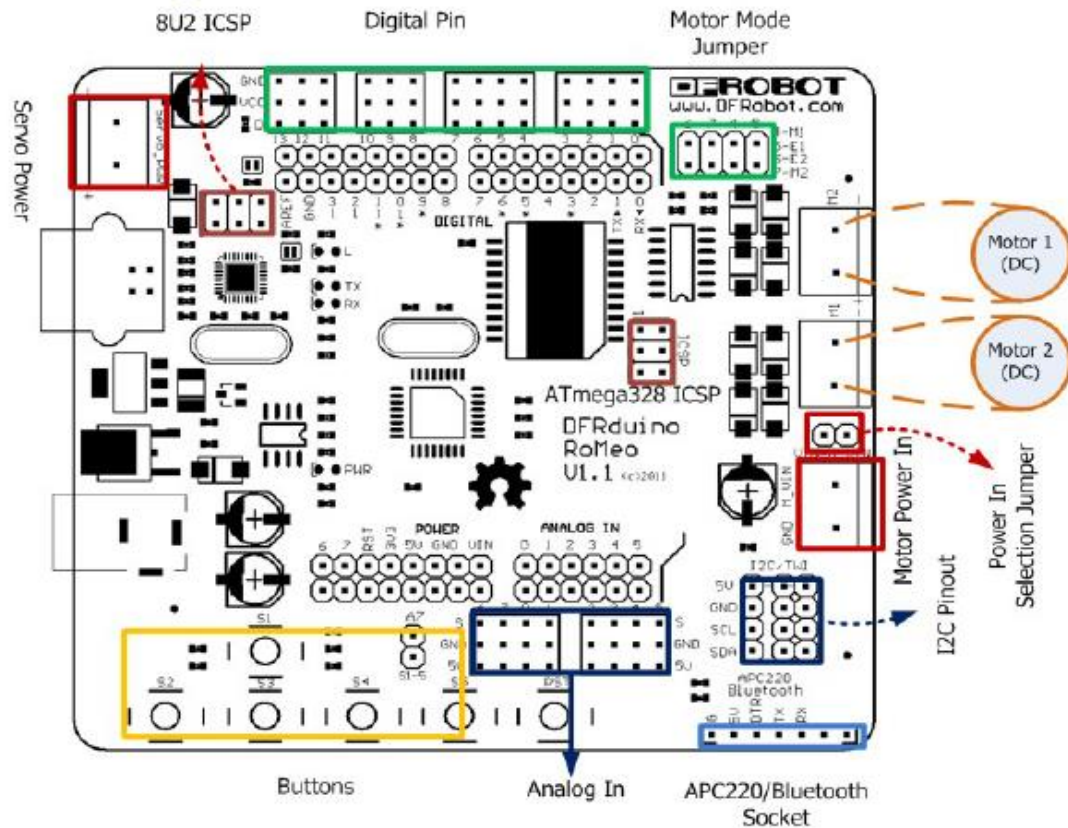


Figura 28. Tarjeta controlador ROMEO.¹²

Ventajas:

- La ventaja más importante radica que en la tarjeta ROMEO se otorgan 4 pines exclusivos para el control de giro y dirección de motores (ver figura 29).
- El circuito puente H para el control de giro de los motores esta embebido en la tarjeta ROMEO en el circuito integrado L298SO20.

¹² [en línea], disponible en:

http://www.dfrobot.com/wiki/index.php/DFRduino_RomeoAll_in_one_Controller_V1.1, recuperado: 5 de agosto del 2013.

- La inexistencia de circuitos de implementación adicionales para programar el micro con el pickit 3 es importante, pues gracias a circuitos integrados embebidos en Romeo, como el manejo del micro-controlador atmega328 para propósito general y el atmega8u2 para capacidad de programación interna (ISP), sólo basta la conexión USB al ordenador.
- La disminución considerable en el tamaño de la tarjeta debido a encapsulados DIP y STM en el manejo de los circuitos para alimentación y regulación, existe el máximo aprovechamiento del espacio y permite un fácil manejo de otros dispositivos electro-mecánicos que necesiten ser implementados posteriormente.
- La comunicación es importante en el funcionamiento del sistema y la tarjeta ROMEO cuenta con sockets para el uso de módulo Bluetooth y módulo APC220 por radio frecuencia.

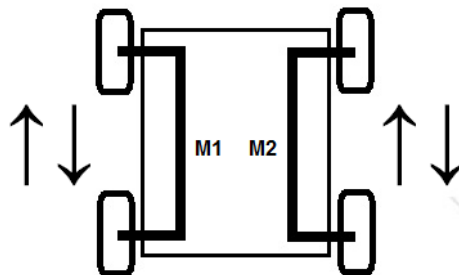


Figura 29. Configuración de ejes para los motores.

Esta plataforma facilita los métodos para diseñar las leyes de comportamiento de algún sistema dado, pues proporciona facilidades al programar los elementos que se desean controlar. La gran ventaja con estas plataformas es que facilitan la implementación de elementos electrónicos que manejan señales analógicas o digitales que implican mayor número de líneas de código para crear procesos aún más estructurados. Sistemas embebidos permiten generar soluciones de forma más económica, ágil y sencilla. Esta ventaja es sólo bien aprovechada por aquellos que entienden y conocen el manejo de las señales y su

debida interpretación y manejo, pues el adquirir una plataforma no garantiza el éxito al querer crear alguna acción robótica pues el diseño y la implementación implican tener conocimientos sólidos sobre la manera cómo funciona y se controla un dispositivo.

5.8 SISTEMA DE INTERACCIÓN CON LOS OBJETOS DEL MUNDO

Como mecanismo de interacción que posee cada agente para con los objetos, se han estudiado varias alternativas que permiten el acarreo, en este caso de figuras creadas especialmente para facilitar su manipulación.

Como primer criterio se piensa que cada figura no puede tener una gran masa, puesto que dificultaría en gran manera su acarreo por el entorno, por lo que su tamaño y peso es importante. Se decide realizar el mecanismo de recolección por medio de un servomotor que cuenta con un gancho de adhesión que le permite encajar con el objeto. Se ve la necesidad de modificar el diseño básico debido a dos consideraciones. Primero, no hay espacio para la colocación del sistema de interacción, ya que de una u otra forma interferirá en el camino de algún sensor y no podrá ejecutar su proceso adecuadamente. Segundo, no es necesario tener 2 sensores ultrasónicos ya que el agente puede girar; es decir que el agente tendrá un frente fijo y girará para acarrear y avanzar. Acarrea con su parte posterior que tiene un manipulador.

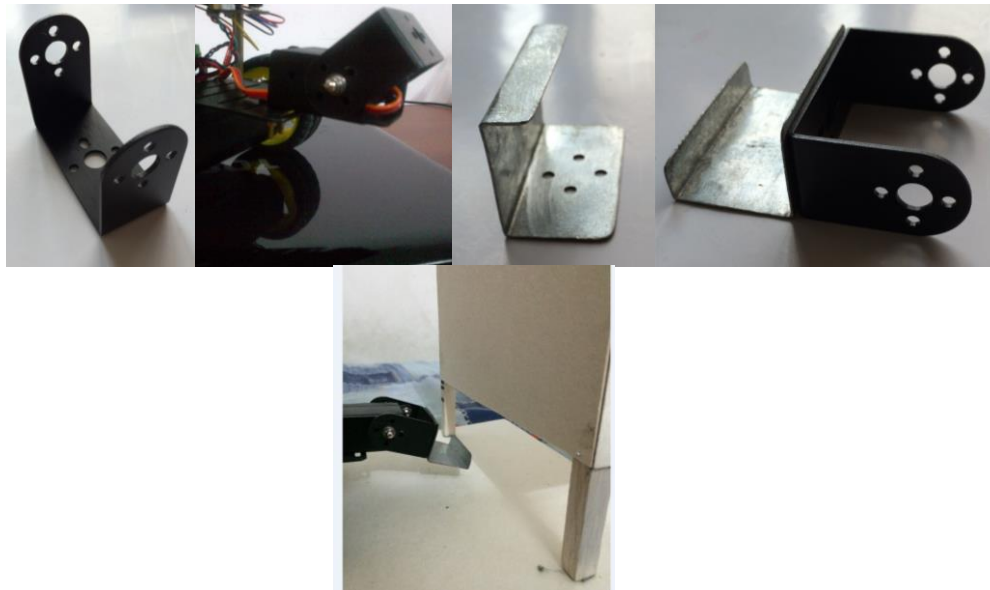


Figura 30. Volcador DF05BB.¹³ Gancho. Ubicación y ensamble. Objeto.

De acuerdo al estado en el que se encuentre el agente entrará a funcionar el mecanismo del servo para cargar o descargar el objeto. El servomotor empieza con el mecanismo libre, es decir que el gancho no tiene ningún objeto enganchado. Se realizan varias pruebas donde se establecen los ángulos exigidos en el servomotor para enganchar y desenganchar los objetos.

5.9 PROGRAMA DE ORGANIZACIÓN

La máquina de estados de la figura 24 se amplía como se muestra en la figura 31 al incluir más estados que describen las tareas a realizar y sus relaciones. Las indicaciones de color púrpura establecen las condiciones para pasar de un estado al otro.

¹³ [en línea], disponible en:
http://www.dfrobot.com/index.php?route=product/product&filter_name=df05bb&product_id=146#.UxP1af15NrU,
recuperado: 27 de febrero del 2014.

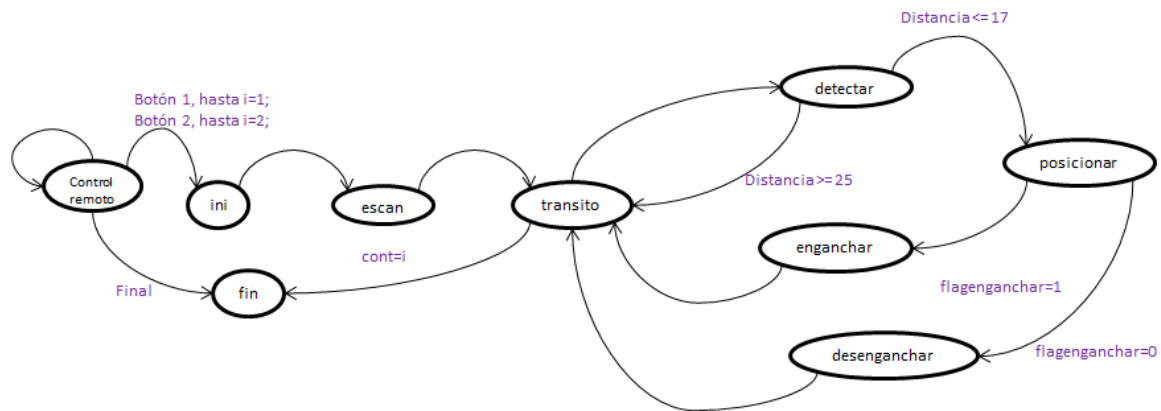


Figura 31. Máquina de estados.

Se incluye un indicador que permite establecer si el agente debe enganchar o desenganchar el objeto luego del tránsito. Un indicador llamado flag inicia en 1. Al pasar del estado enganchar al estado tránsito, flag cambia a 0 indicando que ya enganchó el objeto y que del próximo paso por el estado de posicionamiento es para desenganchar; al hacerlo flag vuelve a 1.

La diferencia entre las máquinas de estados de R1 y de R2 radica en que el estado control remoto sólo aplica para R1, ya que para R2 ese estado podría llamarse Orden de R1; así, cuando R1 atraviesa el estado desenganchar envía a R2 la orden de arranque.

El número de veces que hacen el ciclo viene dado al oprimir el botón del control remoto. Cada vez que R1 pasa del estado desenganchar al estado tránsito, el contador i aumenta en uno. El ciclo continúa hasta que la condición del control remoto lo indique. Cuando R1 llegue a cumplir la condición envía a R2 la orden de que no hay más objetos para recoger por lo que al desenganchar debe parar el ciclo.

Estado 1: ini (R1: Orden humana. R2: Orden de R1)

Descripción del estado: El robot se encuentra inmóvil, libre de todo obstáculo o referencia.

Espera la orden para empezar a funcionar.

Estado 2: escan

Descripción del estado: El robot inicializa la posición tanto del servo de carga como del servo del sensor para detectar el objeto.

Estado 3: tránsito

Descripción del estado: El robot avanza de acuerdo a las lecturas de los sensores infrarrojos derecho e izquierdo.

Estado 4: detectar

Descripción del estado: El robot busca el objeto con el sensor ultrasónico. Si no lo encuentra continua en el estado tránsito.

Estado 5: posicionar

Descripción del estado: El robot ha encontrado el objeto y se posiciona para cargar el objeto.

Estado 6: enganchar

Descripción del estado 2: El robot levanta el objeto.

Estado 7: tránsito

Descripción del estado: El robot entra nuevamente en el estado tránsito y avanza de acuerdo a las lecturas de los sensores infrarrojos derecho e izquierdo.

Estado 8: detectar

Descripción del estado: El robot busca el punto de encuentro con el sensor ultrasónico. Si no lo encuentra continua en el estado tránsito.

Estado 9: posicionar

Descripción del estado: El robot ha encontrado el punto de encuentro y se posiciona para descargar el objeto.

Estado 10: desenganchar

Descripción del estado: El robot libera el objeto. R1 envía a R2 la orden de arranque.

Estado 11: tránsito

Descripción del estado: El robot entra nuevamente al estado 3. Se repite el ciclo hasta cuando una indicación humana lo ordene (R1 recibe la orden de pare y la comunica a R2), o se cumpla un número de ciclos de operación establecido.

La programación para el firmware se muestra en el anexo F, y se organiza de la siguiente manera:

1. Diagramas de flujo del programa.
2. Definición de librerías a usar.
3. Definición de variables globales y locales por rutina y vectores.
4. Definición de protocolos para el movimiento.
5. Inicialización de periféricos (sensores, puertos de comunicación, puente h, motores, servomotores).

6. Desarrollo de rutinas de movimiento del robot.
7. Desarrollo de rutina de comunicación y protocolos.
8. Desarrollo de rutinas para sensores infrarrojos (guía motores dc).
9. Desarrollo de rutinas servomotores.
10. Desarrollo de rutinas sensor ultrasónico (detección de la carga a transportar).
11. Desarrollo de máquina de estados.

Desarrollo:

Definición de librerías:

<Servo.h>

<Metro.h>

<IRremote.h>

Librerías fijas dentro del compilador donde se encuentran funciones orientado a objetos, funciones matemáticas, controladores para el manejo de dispositivos como el control remoto por ir (infrarrojo) entre otras funciones.

La definición de variables se define dependiendo de los tamaños y tipos de datos a manejar, en su mayoría son de 8 bits para las variables globales, también existen vectores y booleanos de 2 estados. Por cada rutina se manejan datos de retorno donde se manejan variables locales dentro de la función.

Para el movimiento se conectaron dos ejes (izquierdo y derecho) con el fin de generar una rotación de 360° sobre su centro. Se define los protocolos para:

- Marcha adelante
- Marcha atrás
- Girar derecha
- Girar izquierda
- Parada

Inicialización de periféricos, se cuenta con funciones dentro de las librerías, se inicializa el puerto de comunicación a 9600 bps, se definen los puertos de entrada para los sensores y salida para motores y servomotores.

Una vez definido los protocolos para el movimiento se realizan las funciones respectivas con un valor de retorno el cual debe indicar la velocidad de tránsito de la determinada rutina; Por ejemplo `advance(200,200)`; indica que se inicia el movimiento adelante y el valor de retorno "200" indica la velocidad del PWM el cual genera la velocidad en cada eje lateral empezando por el izquierdo, luego el derecho. Las velocidades se pueden ajustar en base a los números del 0 al 255.

En el desarrollo de la rutina de comunicación se manejan funciones dependiendo de la tarea a realizar, existe un protocolo para la comunicación por control remoto IR y otro para la comunicación RF. Una vez definida se realiza el tratamiento de los datos, ejemplo para la comunicación por IR se generan códigos en hexadecimal los cuales a través del receptor nos indica el tipo de comando a ejecutar, por otro lado los códigos manejados por la comunicación RF se realizarán en ASCII o caracteres.

Los sensores infrarrojos constan de un transmisor IR y un receptor IR el cual al rebotar sobre el objeto me indica un obstáculo, usando esta información se desarrolla la rutina de corrección de ruta del

robot donde estos sensores tienen directo control sobre la rutina de movimiento del robot tanto en velocidad como en giro.

Los servomotores están dispuestos para dos tareas específicas:

- Recoger y dejar objetos
- Girar el sensor ultrasónico cuando lo requiera el sistema.

Para el control se utiliza directamente la librería de servo.h donde se crean los objetos “servos” y tienen un control de 0° a 180°, el servo dispuesto para recoger y dejar cargas tendrá un control de 0° a 100° y el servo para el sensor tendrá de 0° a 180°.

Para el sensor ultrasónico se desarrolló la rutina según las especificaciones del fabricante del sensor donde se muestrea la frecuencia del PWM generado por el sensor y se guarda el dato de la distancia en la variable.

Se implementa la máquina de estados definitiva (ver figura 31). En el anexo E se muestran los diagramas de flujo de cada estado. Dentro del estado tránsito existen unas subrutinas encargadas de los sensores IR y ultrasónico que brindan la posibilidad a cada agente de interactuar con los objetos del mundo.

Cada tarea simple depende del estado actual de la figura 31 en el cual se encuentra el agente y de las leyes o líneas de código que establecen su comportamiento. “La tarea del agente es encontrar qué secuencia de acciones permite obtener un estado objetivo” (Russell y Norvig, 2004, p. 68), es decir, aquellos estados que se esperan o se desean para establecerlos como éxito. Por lo que la formulación del problema se centra en elegir los estados que permitan al agente acercarse cada vez más, por medio de acciones secuenciales y ordenadas a un estado objetivo. Este proceso implica la búsqueda de un objetivo en el entorno al establecer un problema por medio de las entradas sensoriales y encontrar una solución en

forma de secuencia de acciones, que serán ejecutadas por los efectores, mediante el programa del agente según los parámetros retroalimentados que recibe constantemente.

6. ANÁLISIS DE RESULTADOS

Limitantes de implementación:

Las pruebas de comportamiento de los agentes deben realizarse siempre en el mismo terreno. Características diferentes de suelo cambian por completo la operatividad del sistema en cuanto a giros y velocidades de desplazamientos.

Las paredes de referencia son la frontera que limita los espacios del mundo del agente, por lo que en el entorno no puede existir otra cosa más que las referencias y los objetos del mundo.

Sólo se transmite información en una sola dirección; es decir que hay un agente Líder R1 y un Esclavo R2, pero los agentes no se conocen ni se identifican.

La distancia entre los puntos A-B debe ser igual que entre los puntos B-C de la figura 15 para evitar choques. Ocurre que R2 entregaría un objeto mientras R1 dos o tres dependiendo de la diferencia de distancias.

La posición inicial de los agentes debe ser la más alineada posible con respecto a los objetos para mejorar la continuidad del ciclo.

6.1. MÁQUINA DE ESTADOS

El correcto funcionamiento del sistema cooperativo depende totalmente del programa basado en la máquina de estados. A partir de la máquina de estados se logra escribir el código que comanda el comportamiento de los agentes en el sistema. La máquina de estados de la figura 24 es la primera propuesta, pero la definitiva es la de la figura 31. Se vio la necesidad de postular otros estados y otra dinámica en las transiciones entre estados pues al momento de ejecutar el programa aparecían comportamientos inadecuados que impedían el progreso al siguiente paso del ciclo de operación. Aquí se ve la gran ventaja de la tarjeta ROMEO a la hora de realizar cambios inesperados en el código para ser implementados constantemente en la arquitectura robótica, pues sin necesidad de realizar conexiones complicadas que consumen tiempo y esfuerzo, se logran los cambios necesarios en la codificación y en la conexión de la tarjeta para probar nuevas implementaciones en la arquitectura.

6.2. TAREAS Y ESTADOS

Se hizo la descripción de cada estado y las acciones con la máquina de estados con cierto nivel de abstracción que expresa e informa de manera completa los aspectos importantes para mantener la continuidad del ciclo.

El comportamiento de un agente viene ligado a la tarea que debe realizar en cierto momento del ciclo en un entorno multi-agente, y se identifica que el comportamiento elemental del agente está organizado mediante un protocolo de comunicaciones encargado de formar las relaciones para el comportamiento cooperativo. Cada agente debe interpretar la información que recibe de los sensores

(procesan la información física en notación lógica) para identificar características u objetos en el entorno, y así determinar la acción a realizar.

Se evidencia que a pesar de una implementación correcta de cada sensor y actuador mediante la tarjeta controladora ROMEO, el éxito del sistema cíclico multi-agente viene ligado al programa del agente, el cual está organizado según la máquina de estados hallada para el sistema.

6.3. AGENTE AUTÓMATA

Los agentes cuentan con la capacidad de moverse al guiar su desplazamiento por medio de las referencias. Cuentan con la habilidad suficiente para ejecutar una tarea gracias al desplazamiento y al conocimiento interno que se les otorga en la descripción de cada estado de la máquina; por lo que el sólo hecho de moverse no lo convierte en un agente resolvente de problemas.

Se vio la necesidad de regular la velocidad del agente en su desplazamiento, ya que en el momento en que se topaba con un objeto, no tenía el suficiente espacio para frenar y evitar la colisión. Se cambió de una referencia para avanzar y detenerse, a varias referencias, dos para ser exactos. Ahora el agente inicia su recorrido a toda velocidad con una referencia específica. Cuando supera esa referencia de distancia, disminuye a un 50% su velocidad. Al sobrepasar otra referencia se detiene ya que encontró el punto de carga o descarga.

Se encontró que no fue necesario realizar un modelamiento cinemático para el desplazamiento del robot, pues para crear un control automático funcional basta con un programa de control reactivo basado en la información que va obteniendo por medio de los sensores, las características del medio y los lazos de control.

6.4. ENTORNO

Para que el agente logre el éxito en su tarea se le debe otorgar sólo la información necesaria e indispensable para que actúe de forma adecuada, o su comportamiento será errático y sin sentido debido a la cantidad de información que pueda existir en su entorno. Si el agente no es capaz de elegir entre un estado de su máquina de estados es debido a errores de diseño de la máquina de estados, o que los sensores no pudieron obtener la información necesaria del entorno para que fuera relacionada adecuadamente con su conocimiento interior. Es por esto que se debe otorgar los detalles precisos y necesarios de cada estado del mundo del agente para que pueda cumplir su objetivo llegando a un estado final.

Aquí es importante volver a la delimitación y a las características del entorno pues para el desarrollo del sistema es importante establecer un nivel de abstracción tal que facilite encontrar soluciones a los agentes. Por esto que se vio la necesidad de restringir el entorno al eliminar los obstáculos involucrados en el proceso para establecer fronteras en el mundo de los agentes.

Se establecieron nuevas formas de guiar el desplazamiento del agente de acuerdo a las características del entorno según las referencias de movimiento.

Todo conocimiento se basa en la asociación de términos, de cosas, de características, de lugares, de conjuntos, de ideas, de recuerdos, etc., por lo que para las deducciones o decisiones diarias que tomamos necesitamos de un conocimiento previo o adquirido mediante conceptos abstractos. La abstracción puede ser un término que propone maneras en las que un ente receptor crea filtros que dan pie a una categorización de los elementos del mundo y sus relaciones. Como el mundo real es tan complicado y no es fácil asociar los elementos entre sí, se parte de la idea de que toda categorización implica similitudes o asociaciones o deducciones que implican una sub-categoría para organizar la información acerca del mundo. Estas asociaciones entre elementos y categorías permiten predecir a qué categoría pertenece un

elemento dado para lograr identificarlo mediante su propia percepción y realizar así las inferencias y las acciones correctas. La idea es tener una representación del mundo lo más completa y sencilla posible para evitar confusiones en el momento en que el agente ejecuta sus acciones según la información que lo rodea.

6.5. COMUNICACIÓN Y COORDINACIÓN

Los robots al comunicarse organizan un grupo de trabajo de líder a esclavo en el cual realizan las mismas tareas pero donde sólo uno comanda el momento de realizar las acciones al dar las órdenes para cumplir un objetivo más ambicioso. Aquí es donde se especifica la manera en la que los agentes realizan y coordinan sus movimientos, es decir, la forma en cómo organizan sus acciones para crear el grupo de trabajo necesario, siempre tratando de hacer lo indicado en la máquina de estados del ciclo de operación.

En entornos multi-agentes cooperativos, la comunicación permanente y la coordinación de acciones son primordiales para que en el entorno se lleve a cabo la acción necesaria en el momento adecuado. Una alternativa común es distribuir las tareas a los respectivos agentes, los cuales cuentan de antemano con los elementos necesarios para realizar tareas específicas y sincronizan sus movimientos como miembros de un equipo de trabajo. La coordinación y la comunicación son problemas propios de ejercicios en equipo por lo que la transmisión de eventos relevantes para un agente es importante para la realización del objetivo. Para cualquier objetivo en cualquier entorno, debe existir un plan donde se distribuye la naturaleza del problema y se explica la manera como se realiza la tarea principal por medio de sub-rutinas.

6.6. BATERIAS

La arquitectura cuenta con un panel contenedor para 5 baterías doble AA conectadas en serie para otorgar 7 V a la tarjeta para operar. Inicialmente se usan tipo recargables.

Al momento de compilar la máquina de estados, y a medida que los robots realizaban el proceso, fue evidente que el rendimiento de las baterías disminuía demasiado rápido, y para cada prueba realizada al implementar un código se obtenían cambios en su comportamiento y funcionalidad de acuerdo al estado de la carga de las baterías. Además, era muy engorroso el estar abriendo y cerrando el contenedor de las baterías para poner a cargarlas, pues necesitan de un dispositivo adicional.

Fue necesaria la implementación de baterías tipo industrial que mediante un adaptador de 7 V de alimentación externa, conectado directamente a la tarjeta, permite cargar la batería gracias a los picos de voltaje que entrega, evitando la constante manipulación del contenedor para recargar las baterías.

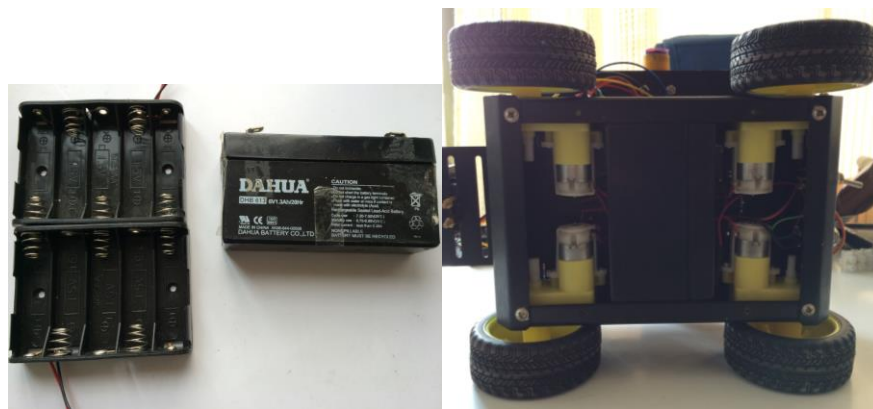


Figura 32. Contenedor de baterías.

6.7 PRIMER PASO PARA GENERAR INTELIGENCIA ARIFICIAL

Lo que se deduce de las proposiciones de Russell y Norving en algunos capítulos de su libro de IA es que: al querer solucionar un problema con agentes en un ambiente controlado lo que en realidad se busca es modificar el entorno mediante uno o más agentes introducidos, donde cada cual cuenta con sensores y con actuadores que informan y provocan un efecto en los objetos del entorno al moverlos o identificarlos, para de alguna manera llevar al entorno a los estados deseados y solucionar el problema propuesto mediante las acciones tomadas y realizadas por cada agente de manera autónoma; cuya única función es actuar al complementar la información obtenida por sus sensores con la información previa dada por el diseñador, y así transmitir fuerzas físicas al entorno por medio de sus efectores.

En el tomar decisiones y en el posterior actuar se trata siempre de realizar los deberes de manera coherente, eficaz y rápida, donde se procura no cometer errores en las acciones que se realizan; algo casi imposible en la mayoría de las situaciones donde hay muchas opciones y caminos para solucionar un problema pues existen muchísimos factores innecesarios que llegan a entorpecer su óptima realización. Lo que se busca siempre en una acción es obtener felicidad, cumplir con un objetivo, suplir una necesidad o generar alguna utilidad, beneficio o entretenimiento, pero es inevitable cometer errores y siempre hay algo que lo dificulta aún más.

La manera en cómo afrontamos los problemas y cómo, a partir de alguna recolección o retención de información generamos acciones para su pronta y efectiva resolución, es lo que de alguna manera genera métodos de solución ante los problemas. El conocer sí las acciones que tomamos son las adecuadas sería información útil para evitar errores y pérdida de tiempo al proponer soluciones; además conseguiríamos la perfección en nuestros actos. Es por ello que se analizaron algunas de las teorías que pueden aportar algún indicio de cómo afrontar de la mejor manera la temática con agentes robóticos móviles automáticos

creados artificialmente con una inteligencia limitada para dar pie a un estudio aún más profundo sobre Inteligencia Artificial.

7. CONCLUSIONES

- Cada agente de este proyecto ofrece sus movimientos, acciones y desplazamientos al sistema multi-agente cooperativo gracias al modelado de la organización y coordinación de sus tareas por medio de una máquina de estados.
- La máquina de estados de este proyecto podría ser implementada en un montacargas industrial, o en un recogedor con ruedas, de acuerdo a los requerimientos industriales necesarios para realizar el transporte de materiales.
- La definición de las tareas en un estado viene conformada por sistemas de control reactivo en el que los lazos pueden estar o no involucrados para desarrollar el programa del agente en dicho instante.
- El comportamiento del agente reactivo puede compararse con el comportamiento de un ser humano en su etapa de infante donde se logran acciones humanas como manejar las manos (*manejo del volcador manipulador*), caminar (*move*), hablar (*comunicar*) y desplazarse por un medio (*posicionarse para llevar a cabo una acción*).
- La utilización de plataformas de código libre permiten al diseñador generar múltiples soluciones con muy poco esfuerzo a la hora de programar un micro-controlador y conectar nuevos dispositivos electrónicos; además evitan la adquisición de software comercial que puede acarrear costos adicionales significativos.

- La adquisición de una plataforma robótica móvil comercial proporciona ventajas al momento de ejecutar un programa agente para observar su desempeño, pues proporciona un más alto nivel de calidad, lo que genera mejores resultados a la hora de su implementación; además disminuye significativamente el tiempo de implementación de una aplicación con robots móviles.

8. REFERENCIAS

[en línea], disponible en: <http://www.buscabiografias.com/bios/biografia/verDetalle/946/Karel%20Capek>,

recuperado: 20 de enero del 2014.

Pérez, V. (2010), Antología y manual de prácticas.

Russell, S. J. y Norvig, P. (2004), Inteligencia Artificial: Un enfoque moderno. Editorial Prentice Hill.

[en línea], disponible en: <http://proton.ucting.udg.mx/materias/robotica/r166/r64/r64.htm>, recuperado: 17

de marzo de 2014.

[en línea], disponible en: http://www.profesormolina.com.ar/tecnologia/robotica/res_rob.htm, recuperado:

17 de marzo de 2014.

Creus Solé, A. (1997), Instrumentación Industrial. Editorial Alfa Omega.

[en línea], disponible en: <http://robotica.wordpress.com/about/>, recuperado 18 de marzo de 2014.

Nwana Hyacinth, S. (1996), “Software agentes: An overview” [en línea], disponible en:

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.50.2589&rep=rep1&type=pdf>, recuperado: 22

de septiembre de 2013.

[en línea], disponible en: <https://www.project2061.org/esp/publications/sfaa/online/chap2.htm>,
recuperado: 17 de marzo de 2014.

Floyd, T. (2006), Fundamentos de sistemas digitales. Editorial Prentice Hall.

Bronson, G. (2000) C++ Para ingeniería y ciencias. Editorial Thomson.

Iglesias Fernández, C. (1998), “Definición de una metodología para el desarrollo de sistemas multiagente”, [en línea], disponible en: <http://www.upv.es/sma/teoria/agentes/tesiscif.pdf>,
recuperado: 2 de septiembre de 2013.

Joshua Noble, (2009), “Programming Interactivity: A Designer’s Guide to Processing, Arduino and OpenFrameworks”, Editorial O’reilly.

Oven Bishop, (2011), “Electronics: A First Course”, Editorial Newnes.

[en línea], disponible en: <http://www.scielo.cl/pdf/formuniv/v3n5/art02.pdf>, recuperado: 3 de junio del 2013.

[en línea], disponible en: <http://dyna.unalmed.edu.co/ediciones/154/articulos/a17v75n154/a17v75n154.pdf>,
recuperado: 3 de junio del 2013.

[en línea], disponible en: <http://www.agenciasinc.es/Noticias/Crean-SMART-un-sistema-de-agentes-roboticos-cooperativos>, recuperado: 25 de febrero del 2013.

Cruz Corona, C. Moreno Vega, J. M., “Estrategias cooperativas paralelas en la solución de problemas de optimización”, [en línea], disponible en: <http://www.redalyc.org/pdf/925/92503408.pdf>, recuperado: 18 de marzo de 2014.

[en línea], disponible en: <http://www.taringa.net/posts/ciencia-educacion/13931420/Control-remoto-o-mando-a-distancia-funcionamiento.html>, recuperado: 10 de marzo de 2014.

[en línea], disponible en: <http://www.aulati.net/comunicacion-inalambrica-con-microcontroladores/>, recuperado: 10 de marzo de 2014.

“ASIMO de honda”, [en línea], disponible en:

<https://www.youtube.com/watch?v=1V9XUMCPGF8>.

“Enfermera robot”, [en línea], disponible en: <https://www.youtube.com/watch?v=rIT0mCeiGG8>.

“Canguro robot de Festo”, [en línea], disponible en:

https://www.youtube.com/watch?v=_4luJ0ZSgy8.

“Ave robot de Festo”, [en línea], disponible en:

<https://www.youtube.com/watch?v=nnR8fDW3Ilo>.

“Pez robot”, [en línea], disponible en: <https://www.youtube.com/watch?v=eO9oseiCTdk>.

“Avión no tripulado”, [en línea], disponible en:

<https://www.youtube.com/watch?v=d84FjQ320UA>.

“Roomba, aspiradora automática”, [en línea], disponible en:

<https://www.youtube.com/watch?v=OB9W3h0guVY>

“Modelo ROVER a control remoto del Sojourner”, [en línea], disponible en:

<https://www.youtube.com/watch?v=p6m4B8IIa7Y>.

[en línea], disponible en:

http://www.ehowenespanol.com/informacion-operacion-montacargas-manera_151088/.

[en línea], disponible en:

http://www.ehowenespanol.com/principales-usos-cargadora-ruedas-lista_485616/.

[en línea], disponible en: <http://serverpruebas.com.ar/news29/nota05.htm>, recuperado: 10 de marzo de 2014.

[en línea], disponible en:

<https://www.gobiernoenlinea.gov.co/web/guest/encyclopedia/wiki/Enciclopedia%20de%20Estado/Preguntas+y+respuestas+frecuentes+de+Espectro+Radioel%C3%A9ctrico+en+Colombia>, recuperado: 11 de marzo de 2014.

[en línea], disponible en: http://www.itu.int/en/ITU-R/space/AmateurDoc/ARS-ART5_S.pdf, recuperado: 13 de marzo de 2014.

[en línea], disponible en:

<http://www.marcombo.com/Descargas/9788426715753/SENSORES/TEMAS/SA%20TEMA%2010-ULTRASONIDOS.pdf>, recuperado: 1 de octubre de 2013.

[en línea], disponible en: <http://www.superrobotica.com/S320110.htm>, recuperado 1 de octubre de 2013.

[en línea], disponible en:

<http://www.daedalus.ei.tum.de/attachments/article/186/URM3.2%20DFROBOT%20Ultraschalltransceiver.pdf>, recuperado: 2 de octubre de 2013.

[en línea], disponible en:

<http://profesores.fi-b.unam.mx/vicflo/Dise%F1o%20de%20Sistemas%20Digitales/Datasheet/Sensores.pdf>, recuperado: 2 de octubre de 2013.

[en línea], disponible en: <http://www.x-robotics.com/sensores.htm>, recuperado: 3 de octubre de 2013.

[en línea], disponible en: http://dl.btc.pl/kamami_wa/e18-d80nk-ds.pdf, recuperado: 4 de octubre de 2013.

[en línea], disponible en: <http://www.elemon.net/elemon/NovedadesDet.aspx?Id=15>, recuperado: 20 de agosto de 2013.

[en línea], disponible en: http://www.datasheetcatalog.net/es/datasheets_pdf/M/C/9/S/MC9S08GT60.shtml, recuperado: 20 de agosto de 2013.

[en línea], disponible en: <http://es.wikipedia.org/wiki/AVR>, recuperado: 21 de agosto de 2013.

[en línea], disponible en: <http://www.atmel.com/Images/doc8161.pdf>, recuperado: 21 de agosto de 2013.

[en línea], disponible en:

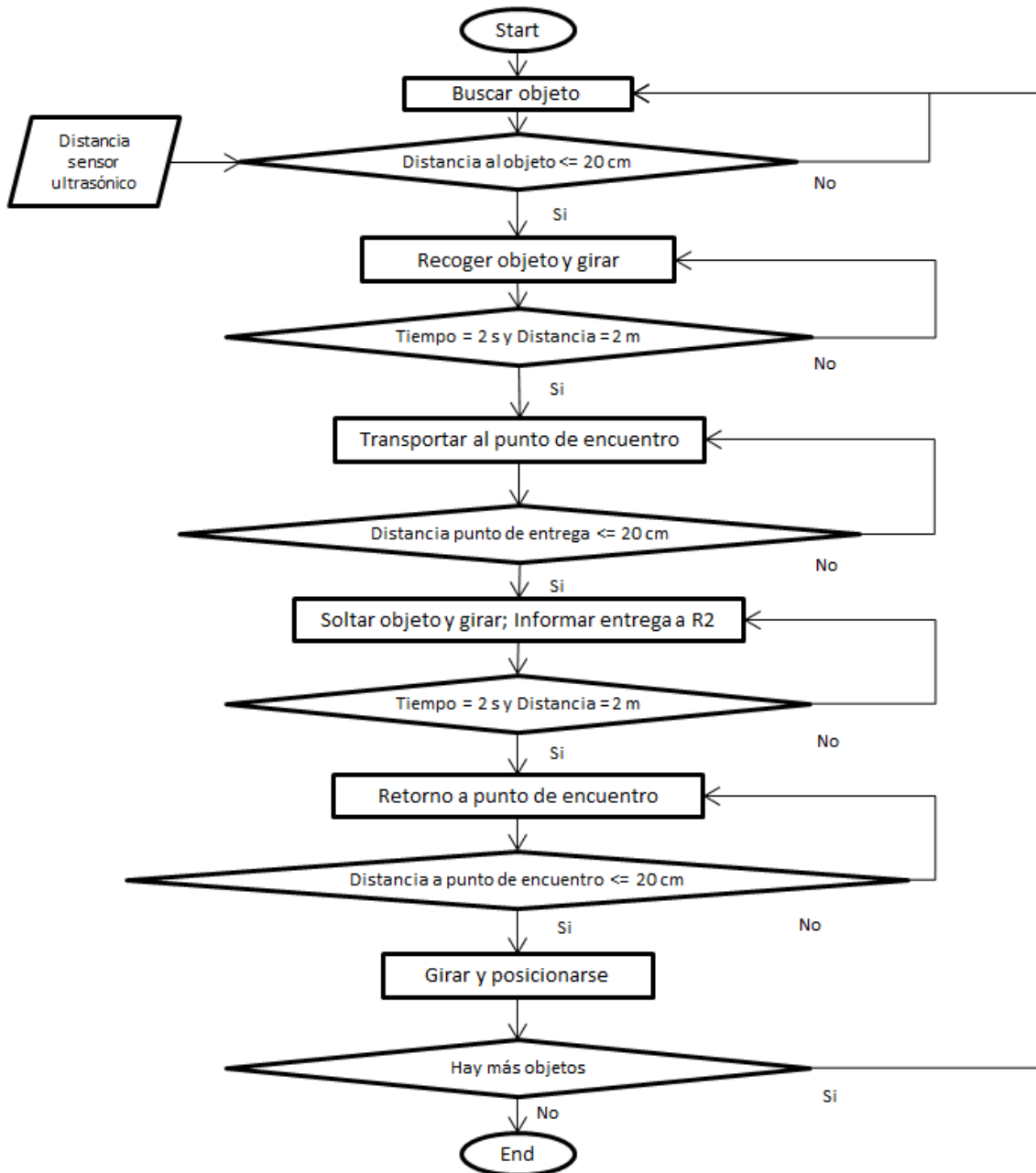
http://www.dfrobot.com/index.php?route=product/product&product_id=57#.U1SXkf15NrU, recuperado: 21 de agosto de 2013.

[en línea], disponible en: <http://www.jameco.com/1/1/47043-tel0026-bluetooth-module-v3-the-bluetooth-chip-csr-bc417143.html>, recuperado 22 de agosto de 2013.

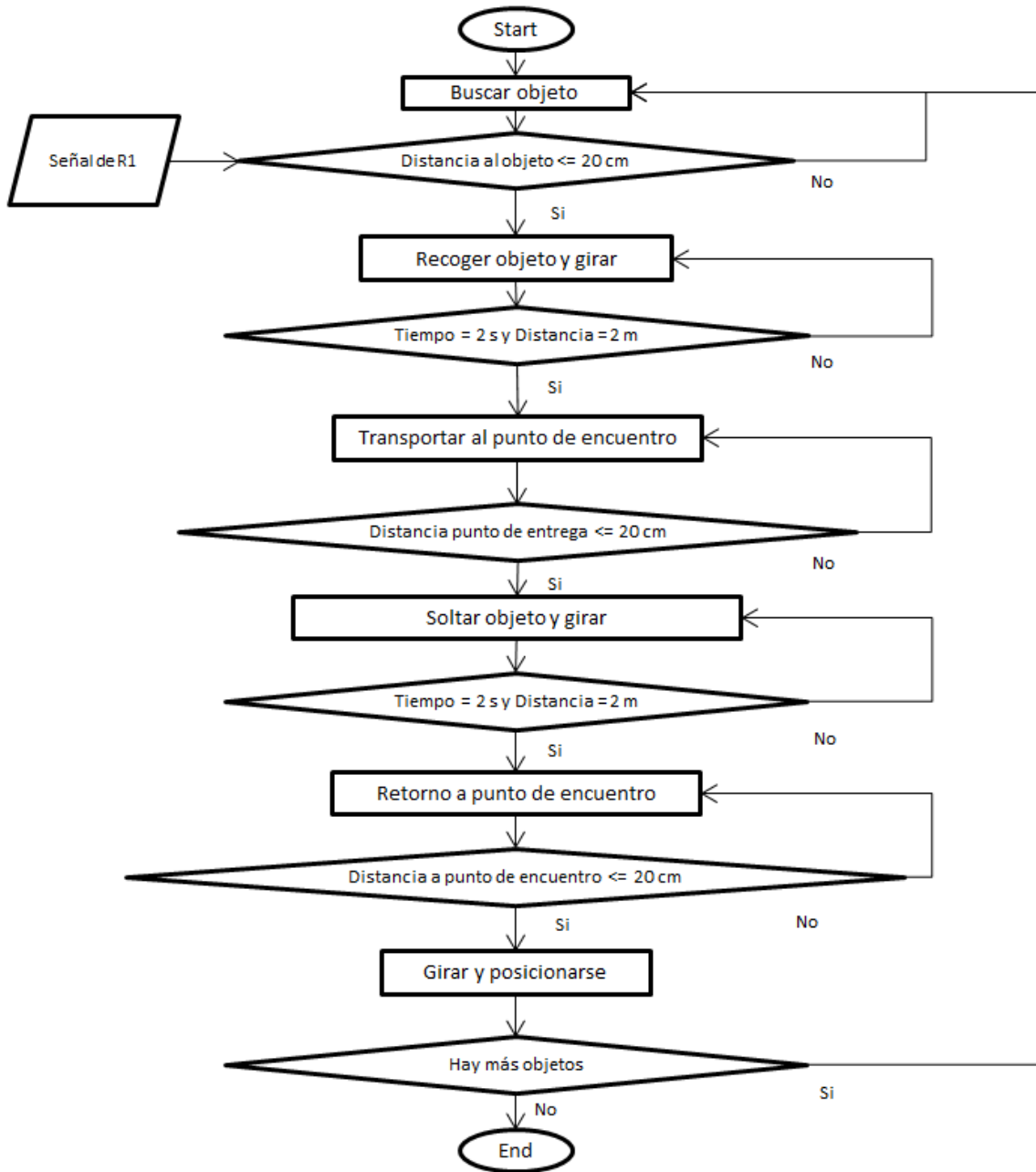
[en línea], disponible en: <http://es.wikipedia.org/wiki/AVR>, recuperado: 21 de agosto de 2013.

ANEXOS

ANEXO A.1.



ANEXO A.2.



ANEXO B

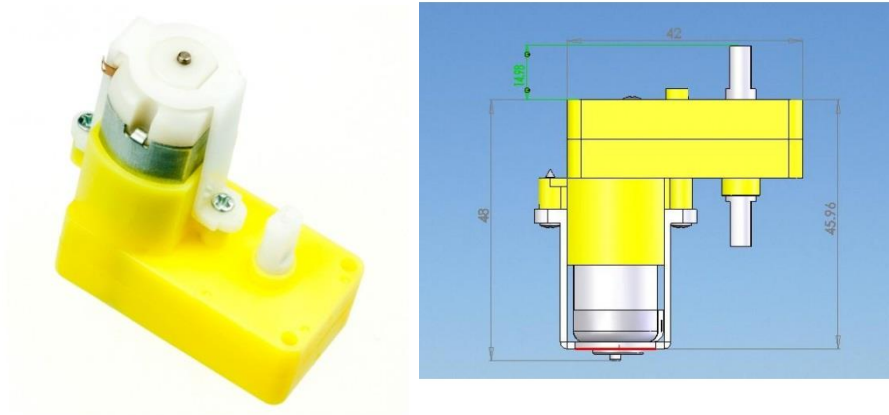


Figura B.1. Motor DC con caja de giro, [en línea], disponible en:

http://www.dfrobot.com/index.php?route=product/product&product_id=100#.U1UO0PI5NrU, recuperado: 3 de junio de 2013.

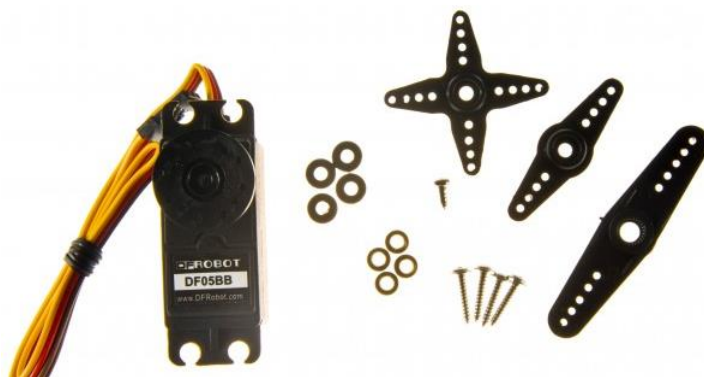


Figura B.2. Servomotor estándar, [en línea], disponible en:

http://www.dfrobot.com/index.php?route=product/product&path=47_108&product_id=236#.U1UT_fl5NrU,

recuperado: 3 de junio de 2013.



Figura B.3. Sensor ultrasónico URM37 V3.2, [en línea], disponible en:

http://www.dfrobot.com/index.php?route=product/product&product_id=53#.U3A4MP15OvM, recuperado: 3 de junio de 2013



Figura B.4. Sensor infrarrojo de switch E18-D80NK, [en línea], disponible en:

http://www.dfrobot.com/index.php?route=product/product&product_id=114#.U1UV5v15NrU, recuperado: 3 de junio de 2013



Figura B.5. Módulo de radio control, [en línea], disponible en: <http://www.ebay.com/itm/IR-Remote-Control-Infrared-DIY-Kit-For-Arduino-Uno-Mega-PIC-Project-/200929109288>



Figura B.6. Módulo de radio comunicación APC220, [en línea], disponible en: http://www.tinyosshop.com/index.php?route=product/product&product_id=194, recuperado: 3 de junio de 2013

ANEXO C

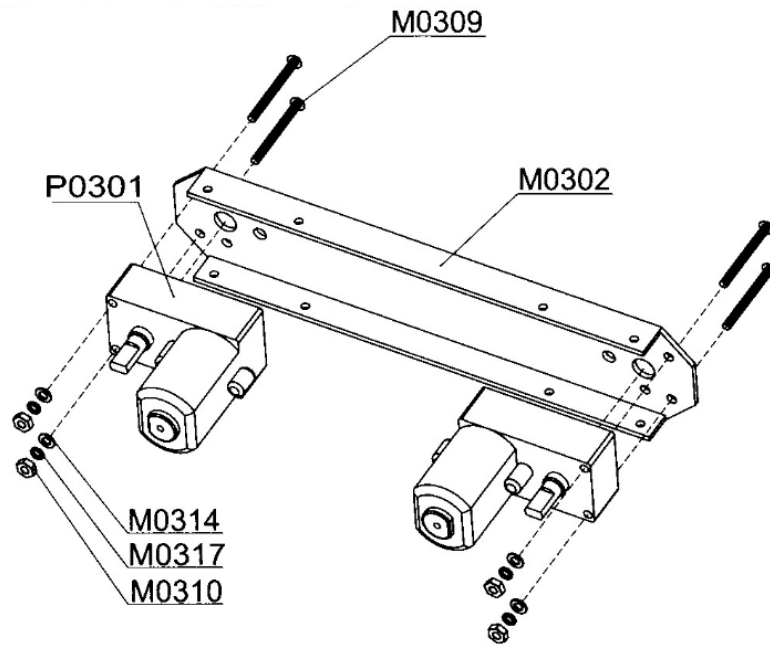


Figura C.1. Motores y cara lateral del agente móvil. Manual de instrucciones 4WD Mobile Platform.

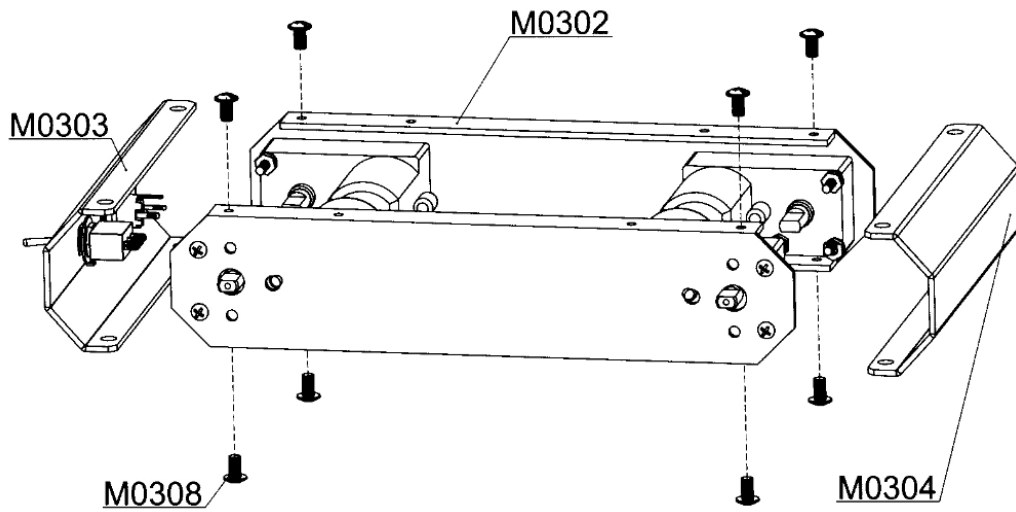


Figura C.2. Cara trasera y delantera del agente móvil. Manual de instrucciones 4WD Mobile Platform.

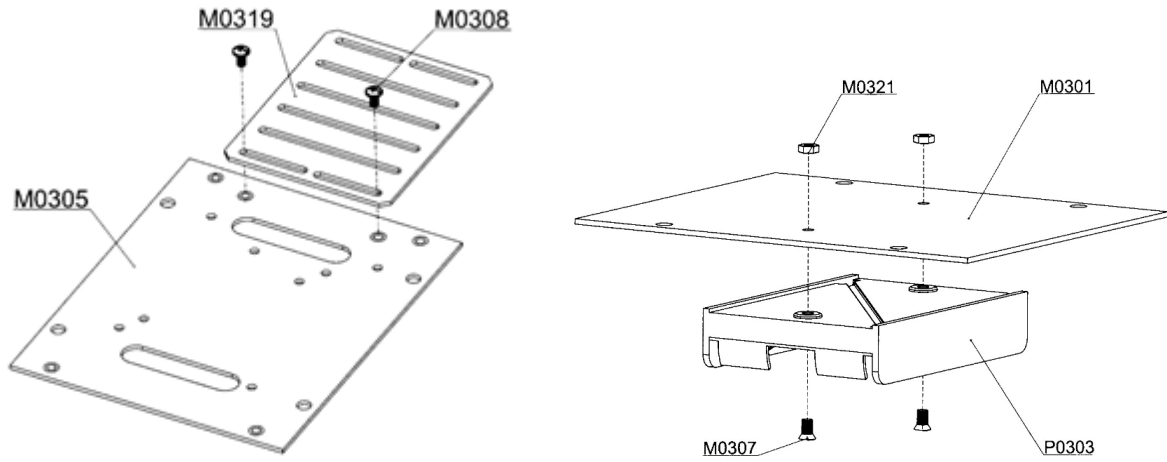


Figura C.3. Tapa y fondo con soporte de baterías de la carcasa del agente móvil respectivamente. Manual de instrucciones 4WD Mobile Platform.

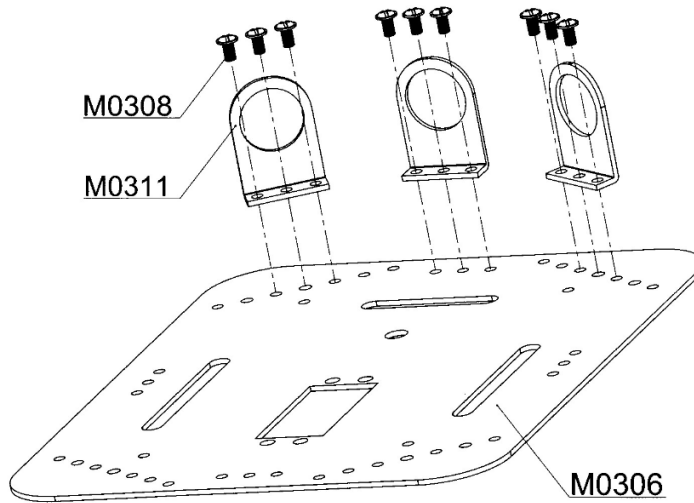


Figura C.4. Soporte superior para sensores infrarrojos del agente móvil. Manual de instrucciones 4WD Mobile Platform.

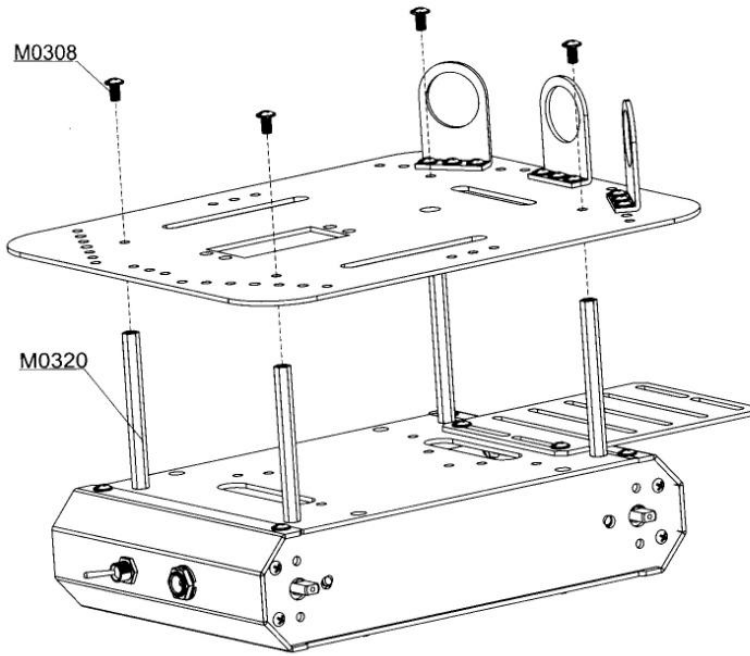


Figura C.5. Unión del soporte con la carcasa del agente móvil. Manual de instrucciones 4WD Mobile Platform.

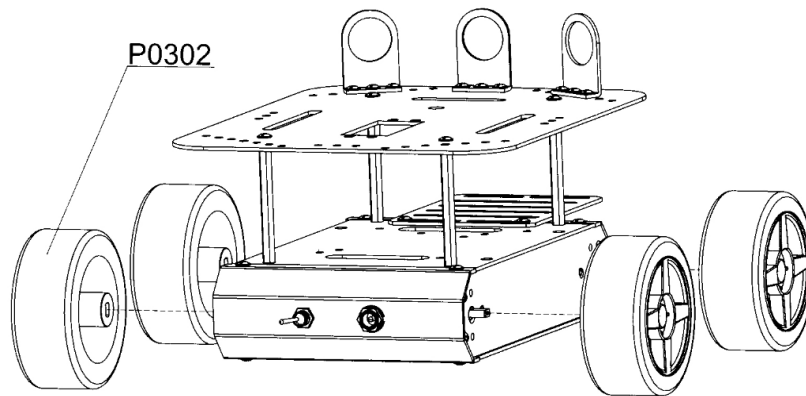


Figura C.6. Llantas en la carcasa del agente móvil. Manual de instrucciones 4WD Mobile Platform.

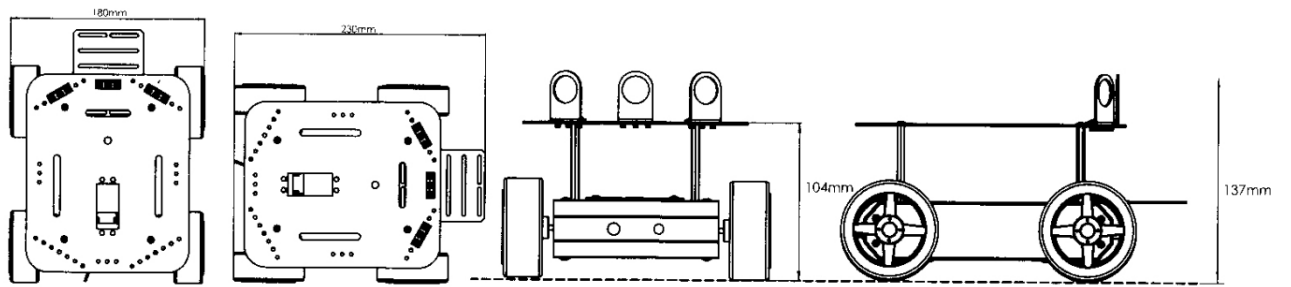


Figura C.7. Dimensiones de la carcasa del agente móvil. Manual de instrucciones 4WD Mobile Platform.

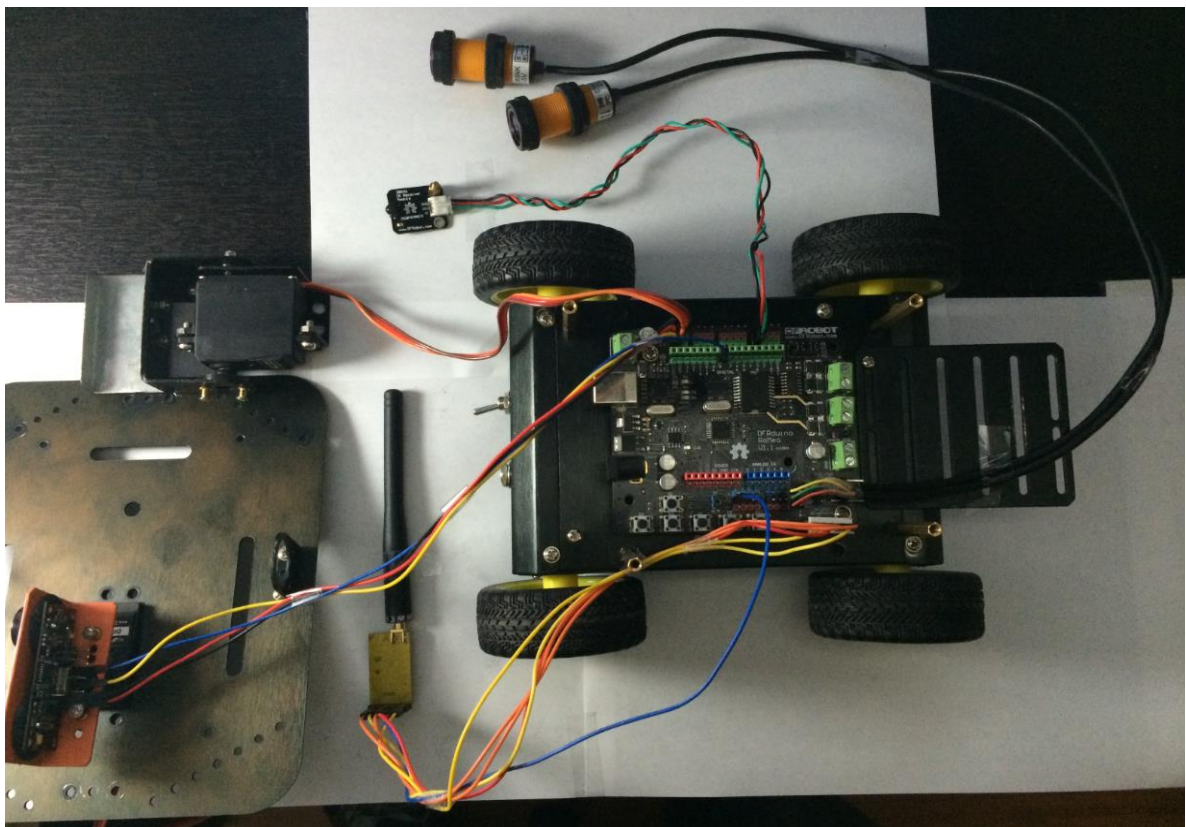


Figura C.8. Conexiones del agente móvil.

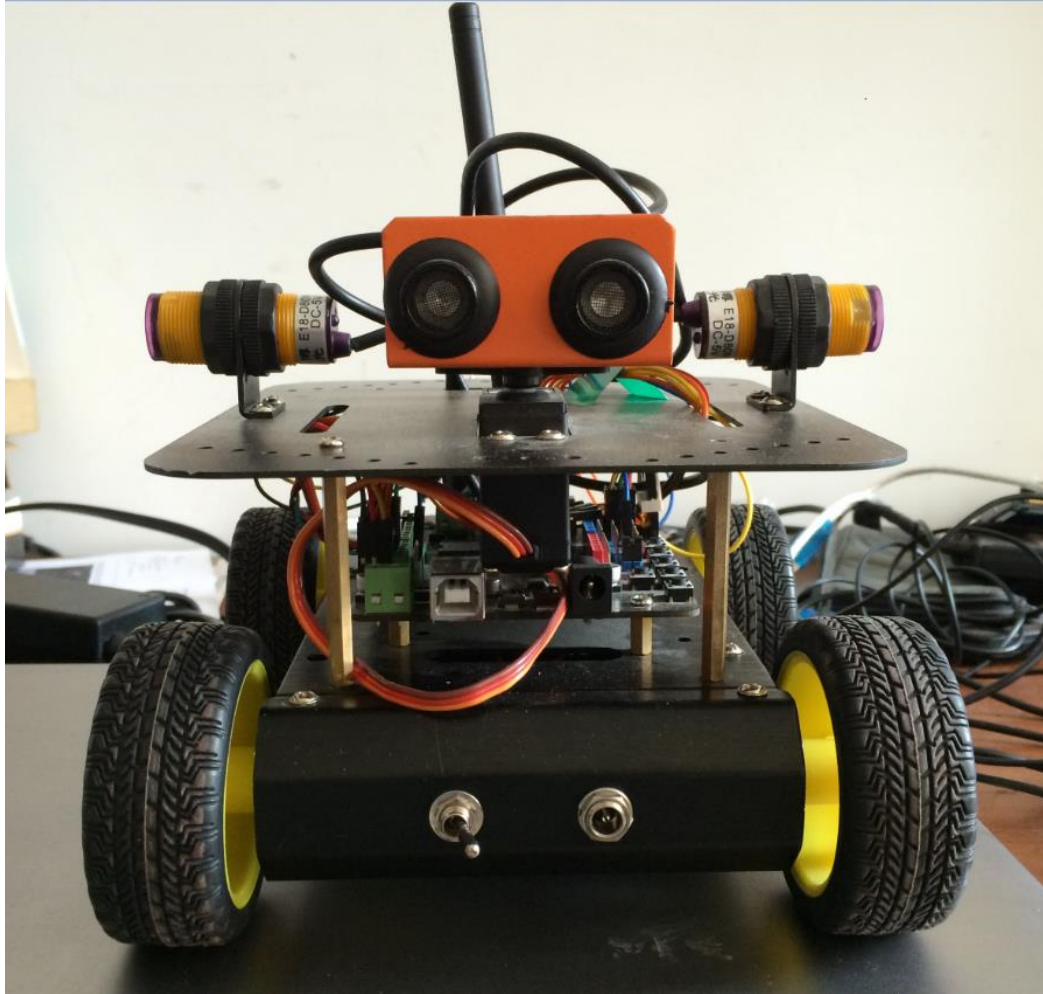


Figura C.9. Agente móvil.

ANEXO D

Presupuestos	Unidades	Valor unitario	Valor total
Microcontrolador PIC 16F877A	2	\$ 18 000	\$ 36 000
Programador PICKIT 3	1	\$ 130 000	\$130 000
Moto reductores	8	\$ 25 000	\$ 200 000
Sensores infrarrojos EE-SB5	6	\$ 20 300	\$ 121 800
Memoria EPROM-OTP 27C2048	2	\$ 23 200	\$ 46 400
Sensores ultrasónicos HC-SR04	2	\$ 11020	\$ 22040
Módulos de comunicación XB24-AWI-001	2	\$ 75 400	\$ 150 800
Material de diseño (Circuitos Integrados, Cables, conectores, impresos, etc.)	-	\$ 300 000	\$ 300 000
Llantas	8	\$ 20 000	\$ 160 000
TOTAL			\$ 1'167 040

Tabla D.1. Cotización FUENTES: SIGMA ELECTRÓNICA LTDA, TNC ELECTRONICS S.A.S.,
MOVILTRONICS LTDA.

Presupuestos	Unidades	Valor Unitario	Valor TOTAL
Plataforma móvil PIRATE-4WD	2	\$ 95 608	\$ 191 216
Sensor ultrasónico URM37 con servo motor	2	\$ 51 732	\$ 103 464
Sensor infrarrojo E18-D80NK	4	\$ 14 370	\$ 57 480
Módulo Bluetooth	2	\$ 41 960	\$ 83 920
Tarjeta controlador Romeo-All in one (Arduino compatible con Atmega 328)	2	\$ 76 448	\$ 152 896
Cables y conectores	-	\$ 150 000	\$ 150 000
TOTAL			\$ 738 976

Tabla D.2. FUENTE: DFROBOT.COM, ROBOTSHOP.COM.

Gastos totales	Valor
Tarjeta controlador AtmegaG 328 Romeo-All in one (Compatible con Arduino))	\$ 152 896
Plataforma móvil PIRATE-4WD	\$ 191 216
Kit Control Remoto Infrarrojo	\$ 12 380
APC220 kit de radio comunicación	\$ 65 980
Sensor ultrasónico URM37 con servo motor	\$ 58 000
Sensor infrarrojo E18-D80NK	\$ 43 110
Volcador con servo motor	\$ 72 000
Soporte para ultrasónico	\$ 22 000
Gancho	\$ 5 000
Baterías 7 V	\$ 26 500
TOTAL	\$ 649 082

Tabla D.3. Gastos totales.

ANEXO E

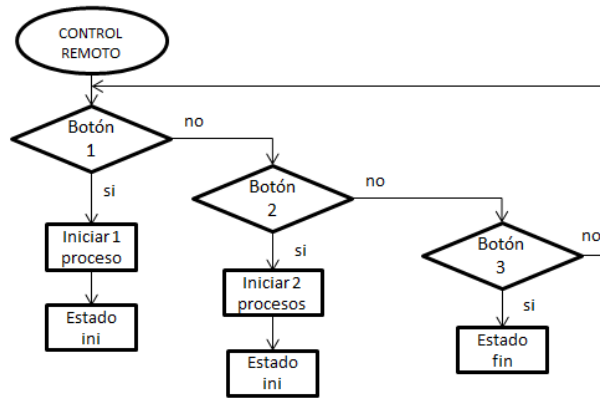


Figura E.1. Diagrama de flujo de control remoto.

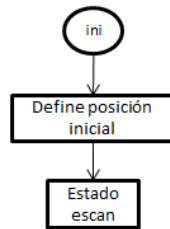


Figura E.2. Diagrama de flujo de estado ini.

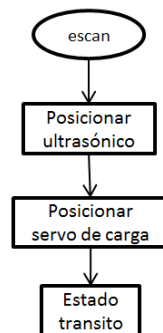


Figura E.3. Diagrama de flujo de estado escan.

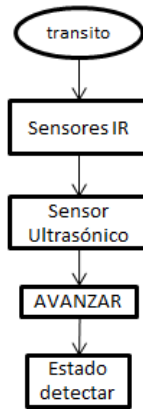


Figura E.4. Diagrama de flujo de estado tránsito.

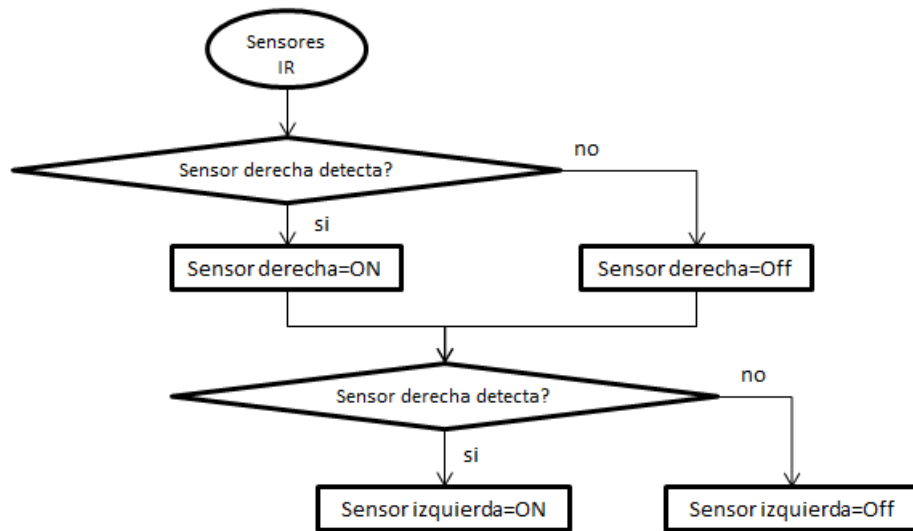


Figura E.5. Diagrama de flujo de control de sensores IR.

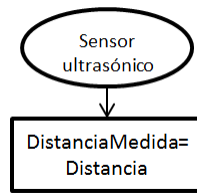


Figura E.6. Diagrama de flujo de control de sensor ultrasónico.

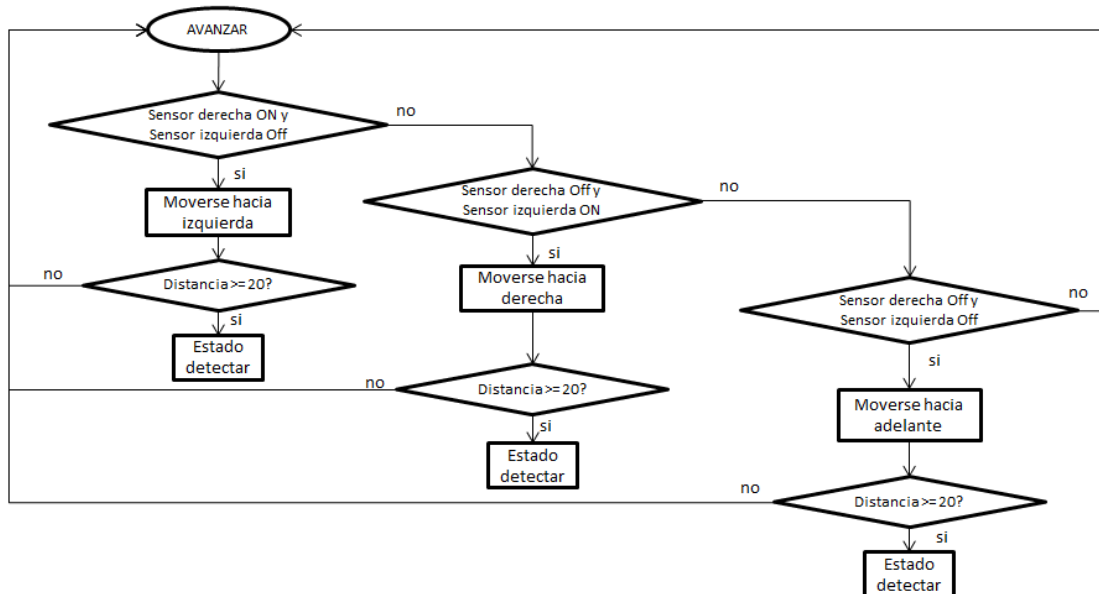


Figura E.7. Diagrama de flujo de control avanzar.

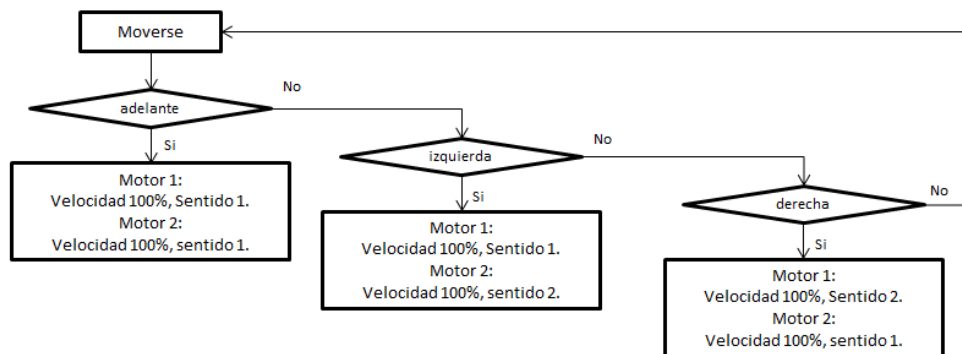


Figura E.8. Diagrama de flujo de control moverse.

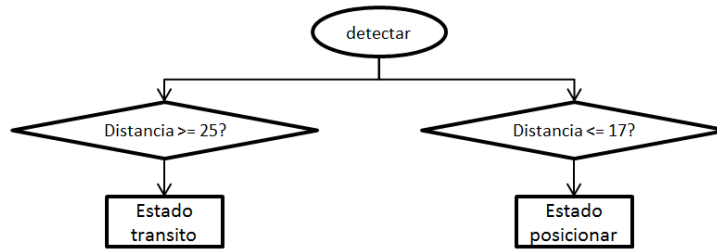


Figura E.9. Diagrama de flujo de estado detectar.

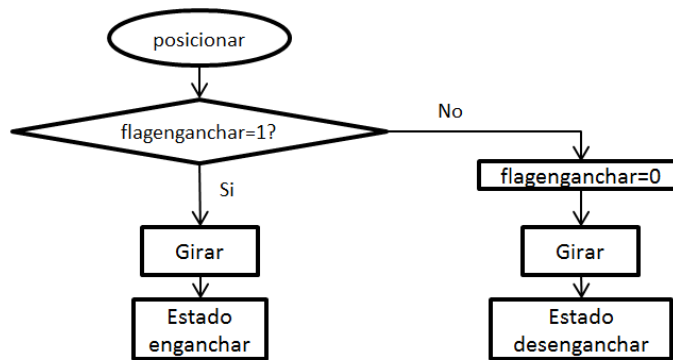


Figura E.10. Diagrama de flujo de estado posicionar.

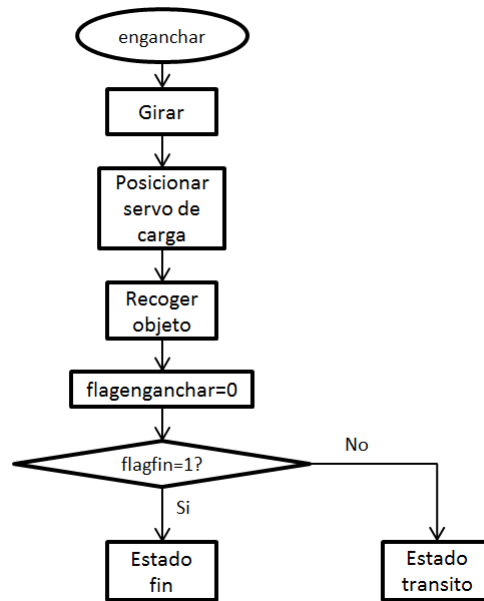


Figura E.11. Diagrama de flujo de estado tránsito.

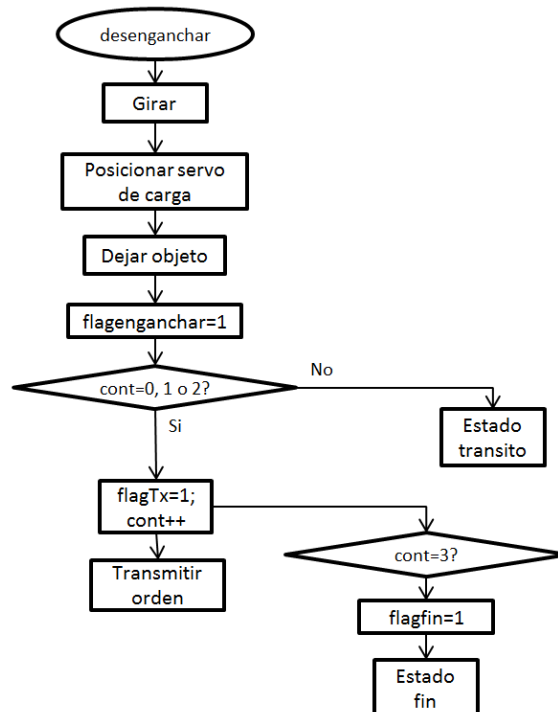


Figura E.12. Diagrama de flujo de estado desenganchar.

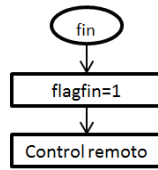


Figura E.13. Diagrama de flujo de estado fin.

ANEXO F

Código fuente

```
#include <Servo.h>                                // librerias usadas

#include <Metro.h>

#include <IRremote.h>

Servo servocarga;

Servo servosensor;                               // objetos de control " control a servomotores"

//////////////////////////////////// variables irremote////////////////////////////////////

int RECV_PIN = 3;

Metro output = Metro(30,true);

IRrecv irrecv(RECV_PIN);

decode_results results;

//// inicializacion del control remoto////

//////////////////////////////////// definicion estados////////////////////////////////////

#define ini 0

#define escan 1

#define detectar 2

#define enganchar 3

#define posicionar 4

#define transito 5

#define desenganchar 6

#define fin 7
```

```
#define TZ 0

#define TC 1

#define TD 2

#define verificar 0

#define validar 1

unsigned char estado,test,comunicaciones; ////definicion de variables dentro del programa

////////// sensor infrarojo//////////7777

int left = 4;    //// sensor izquierda

int right = 5;  ///sensor derecha

int vall=0,detectl=0;

int valr=0,detectr=0;

////////// CONTROL MOTOR //////////

int E1 = 5; //M1 Control de velocidad

int E2 = 6; //M2 Control de velocidad

int M1 = 4; //M1 Direccion Control

int M2 = 7; //M1 Direccion Control

////////// CONTROL SERVOS PWM//////////

int pos=0;                // variable to store the servo
position

int pos1=0;
```

```
int flag1=0; //indica descargar o recoger objeto

int URPWM=13; // PWM Output 0-25000us,cada 50us
representa 1cm

int URTRIG=8; // PWM trigger pin

boolean up=true; // creación de variable boolean

unsigned long time; // creación variable de tiempo

unsigned long urmTimer = 0; // timer para el control de lectura
del sensor

unsigned long timeini=0; // timer inicio estados

unsigned char startproces=0;

unsigned char estadofin=0,flagenganchar=1,flagtest=0,flagtransito=0;//flagini=0;

int m=0,k=0;

//////////////////configuraciones ////////////////////77

int incomingByte = 0; // buffer entrada de datos serial

char buffer[6];

int cmd1=0;

//////////////////FUNCIONES ULTRASONIDO//////////////////

unsigned int Distance=0; //Guarda el dato de la distancia

void setup(){ // INICIO FUNCION DE CONFIGURACIÓN

    delay(1000); // inicialización del puerto Serial
```

```

    Serial.begin(9600);                // configuracion del puerto baud rate
to 9600

    irrecv.enableIRIn();              // Start the receiver

    servocarga.attach(12);            // Pin 12 control servocarga

    delay(10);

    servosensor.attach(11);          // Pin 11 control servosensor

    digitalWrite(URTRIG, LOW);

    digitalWrite(URTRIG, HIGH);

    pinMode(left , INPUT);           //// CONFIGURACION SENSORES INFRAROJO IZQUIERDA

    pinMode(right , INPUT);          //// CONFIGURACION SENSORES INFRAROJO IZQUIERDA

    int i;

    for(i=4;i<=7;i++)                 /////CONFIGURACION DE SALIDAS DE MOTOR DEL PIN 4 AL 7
COMO SALIDAS

    pinMode(i, OUTPUT);

    test = TC;

    comunicaciones=verificar;

    startproces=0;

    PWM_Mode_Setup();                 ///// INICIALIZACION DE LA FUNCION PARA EL SENSOR
ULTRASONICO

}

////////// FUNCIONES MOTOR//////////

void stop(void) //Stop

{

```

```
digitalWrite(E1,LOW);

digitalWrite(E2,LOW);

}

void advance(char a,char b) //Move forward

{

analogWrite (E1,a); //PWM Speed Control

digitalWrite(M1,LOW);

analogWrite (E2,b);

digitalWrite(M2,HIGH);

}

void back_off (char a,char b) //Move backward

{

analogWrite (E1,a);

digitalWrite(M1,HIGH);

analogWrite (E2,b);

digitalWrite(M2,LOW);

}

void turn_L (char a,char b) //Turn Left

{

analogWrite (E1,a);

digitalWrite(M1,HIGH);

analogWrite (E2,b);

digitalWrite(M2,HIGH);
```

```
}

void turn_R (char a,char b) //Turn Right

{

analogWrite (E1,a);

digitalWrite(M1,LOW);

analogWrite (E2,b);

digitalWrite(M2,LOW);

}

void avanzar(){

    Serial.println("transito");

        delay(10);

switch(test){

    case TZ:

        Serial.println("TZ");

        delay(10);

        PWM_Mode();          ///lectura de la distancia

        delay(20);

        Serial.print(Distance);
```

```
    if (detectl==HIGH){  
        advance(0,200);  
        delay(300);  
    }  
    else{  
        advance(200,200);  
    }  
  
    if(detectr==HIGH){  
        advance(200,0);  
        delay(300);  
    }  
    else{  
        advance(200,200);  
    }  
  
    if(detectr ==LOW && detectl==LOW){  
        advance(200,200);  
        delay(300);  
    }  
  
    flagtest = 1;  
  
    test = TC;
```

```
break;

case TC:

    advance(200,200);

    Serial.println("TC");

    delay(20);

    servosensor.write(90);

    delay(300);

    PWM_Mode();          ///lectura de la distancia

    delay(20);

    Serial.print(Distance);

    if(Distance <=40){

        advance(150,150);

        if(Distance <=20){

            advance(0,0);

            estado = detectar;

        }

    }

    delay(200);

    break;
```



```
    }//fin switch

} //fin avanzar

void com(){

    if (Serial.available() > 0) {

        // read the incoming byte:

        incomingByte = Serial.read();

        delay(20);

        if(incomingByte == '*'){

            estado = ini;

            startproces=1;

            delay(100);

        }

        for(int i=0;i<6;i++){

            buffer[i]=incomingByte;

            cmd1=buffer[0];

            k=buffer[4];

            comunicaciones = verificar;

        }

    }

} //fin com
```

```
void sensorir(){

    vall = analogRead(left); // read input value

    valr = analogRead(right);

    if (valr <= 20) { // check if the input is HIGH (button released)

        detectr=HIGH;

    }

    else{

        detectr=LOW;

    }

    if (detectr == HIGH){

        test = avanzar;

    }

}

if (vall <= 20) {

    detectl=HIGH;

}

else{

    detectl=LOW;

}

if (detectl == HIGH){

    test = avanzar;
```

```
    }  
}  
  
void loop(){  
  
    sensorir();  
  
    com();  
  
    maquina_estados();  
}  
  
void PWM_Mode_Setup(){  
  
    pinMode(URTRIG,OUTPUT);           // A low pull on pin COMP/TRIG  
  
    digitalWrite(URTRIG,HIGH);       // Set to HIGH  
  
    pinMode(URPWM, INPUT);           // Sending Enable PWM mode command  
  
}  
  
void PWM_Mode(){                     // a low pull on pin COMP/TRIG  
triggering a sensor reading  
  
    digitalWrite(URTRIG, LOW);  
  
    digitalWrite(URTRIG, HIGH);      // reading Pin PWM will output pulses  
  
    unsigned long DistanceMeasured=pulseIn(URPWM,LOW);
```

```
        if(DistanceMeasured==50000){                                // the reading is invalid.

            Serial.print("Invalid");

        }

        else{

            Distance=DistanceMeasured/50;                          // every 50us low level stands for
1cm

        }

    }

}
```

```
void maquina_estados(){

    switch(estado){

    case ini:

        delay(200);

        if(startproces){

            m=58;

            estado = escan;

            Serial.println("inicio");

        }

    }
```

```
break;

case escan:

    if(millis()-timeini>=1000){                // intervalo de un segundo

        timeini = millis();                    // guarda el el conteo en
milisegundos desde la ultima lectura

        pos1=90;        /// posicion del sensor

        pos=160;        /// posicion del motor para la carga

        servosensor.write(90);

        servocarga.write(160);

        delay(1000);

        //Serial.println("escan");

        estado = transito;

    }

break;

case detectar:

    PWM_Mode();        ///lectura de la distancia

    delay(20);

    if (Distance >= 25){

        advance(200,200);
```

```
    }  
  
    if(Distance <=20){  
  
        advance(0,0);  
  
        delay(1000);  
  
        timeini = millis();  
  
        estado = posicionar;  
  
    }  
  
    timeini = millis();  
  
break;  
  
case desenganchar:  
  
    timeini = millis();  
  
    pos=180;  
  
    servocarga.write(pos);  
  
    flagenganchar = 1;  
  
    m=45;  
  
    PWM_Mode();  
  
    delay(10);  
  
    if(Distance >=58){  
  
        estado = transito;  
  
        delay(20);
```

```
    advance(0,0);  
  
    delay (500);  
  
}
```

```
break;
```

```
case enganchar:
```

```
    timeini = millis();  
  
    pos=70;  
  
    servocarga.write(pos);  
  
    PWM_Mode();  
  
    delay(10);  
  
    if(Distance >=40){  
  
        flagenganchar = 0;  
  
        m=61;  
  
        estado = transito;  
  
        delay(20);  
  
        advance(0,0);  
  
        delay (500);  
  
    }
```

```
    break;

    case posicionar:

turn_L (0,0);

    for (int i=0; i <= m; i++){

        turn_L (228,228);

        }

        if(flagenganchar==0){

            estado = desenganchar;

            delay(1000);

        }

        if(flagenganchar==1){

            estado = enganchar;

            delay(1000);

        }

    break;

    case transito:

        advance(140,190);

        delay(10);

        PWM_Mode();
```



```
        delay(10);

        avanzar();

        break;

    case fin:

        advance(0,0);

        if(estadofin){

            estado =ini;

        }

        Serial.println(estado);

        break;

}

} //fin switch estado

} //fin maquina estado
```