

**DISEÑO Y CONSTRUCCIÓN DE UNA PLATAFORMA ROBÓTICA
COOPERATIVA BASADA EN EL SISTEMA EMBEBIDO PRIMER_2 PARA LA
GESTIÓN Y MANEJO DE INVENTARIOS**

RAUL HERNANDO CADENA CONTRERAS
JHONATTAN FREDY MORENO BERNAL

UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍAS ELECTRICA, ELECTRONICA Y DE
TELECOMUNICACIONES
INGENIERIA ELECTRONICA
BUCARAMANGA
2011

**DISEÑO Y CONSTRUCCIÓN DE UNA PLATAFORMA ROBÓTICA
COOPERATIVA BASADA EN EL SISTEMA EMBEBIDO PRIMER_2 PARA LA
GESTIÓN Y MANEJO DE INVENTARIOS**

RAUL HERNANDO CADENA CONTRERAS
JHONATTAN FREDY MORENO BERNAL



Trabajo de Grado para optar al título de
Ingeniero Electrónico

Director
MSc. JORGE HERNANDO RAMÓN SUÁREZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍAS ELECTRICA, ELECTRONICA Y DE
TELECOMUNICACIONES
INGENIERIA ELECTRONICA
BUCARAMANGA
2011

A mis padres quienes siempre me han apoyado en este largo camino de experiencias y a mi Abuelo Alfonso por compartir sus experiencias.

“las cosas más sencillas siempre funcionan”

Raul Hernando

*A toda mi familia por su apoyo incondicional,
a la familia Carrillo Arenas por su colaboración, a
todos los profesores y compañeros de Universidad
que de una u otra forma me han acompañado a lo
largo de este proceso de formación, y en especial
a todo el Grupo ERA, pues con ellos aprendí
lo que se requiere para ser ingeniero.*

Jhonattan Fredy

Agradecimientos

Al profesor Jorge Hernando Ramón, por su apoyo incondicional en este proyecto, a nuestros compañeros: Carlos Andrés, Christian, Javier, Robin, Juan, Yesid y demás personas que conforman el Grupo ERA, al Profesor Jorge Meneses y a Alexander Velilla por su desinteresada colaboración en el laboratorio FMS. A Roxana por las correcciones de este libro, a mi “Cucha-Rita” por la comida, la paciencia, y los consejos que nos dio durante todo este tiempo en la Universidad.

Jhonattan Fredy Moreno Bernal
Raul Hernando Cadena Contreras

TABLA DE CONTENIDO

Capitulo I.....	20
1.PRESENTACION.....	20
1.1.Introducción	20
1.2.Estado del Arte	21
Capitulo II.....	27
2.MARCO TEORICO	27
2.1.Sistema Embebido PRIMER – 2.....	27
2.2.Sistema Automatizado de Gestión de Inventario	29
2.3.Redes de Comunicación en Malla (MESH).....	30
2.3.1.Definición.	30
2.3.2.Ventajas.	30
2.3.3.Aplicaciones.....	31
2.3.4.Operación de la Red en Malla.....	32
2.4.Sistemas Robóticos Cooperativos	33
2.4.1.Características de los Sistemas Robóticos Cooperativos.....	34
2.4.2.Arquitectura de Agrupación.....	34
2.4.3.Diferenciación.	35
2.4.4.Comunicación	35
2.4.5.Arquitecturas Representativas	36
2.4.6.Solución de Conflictos de Recursos.	37
Capítulo III	39
3.DESCRIPCIÓN DEL SISTEMA	39
3.1.Agentes.....	41
3.1.1.Componente electrónico	41
3.1.1.1.Sistema Embebido.....	41

3.1.1.2 Sistema de Control.	41
3.1.1.3 Cámara de video on board.....	42
3.1.1.4 Red Inalámbrica xbee.	43
3.1.1.5 Lectores RFID.....	44
3.1.1.6 Motores.	45
3.1.2.Componente Mecánico	46
3.1.2.1 Chasis.	46
3.1.2.2 Ruedas.....	46
Capitulo IV	49
4.SELECCIÓN Y CARACTERIZACION DE COMPONENTES.....	49
4.1.Selección del Sistema de Control	50
4.2.Selección de Motores	54
4.3.Selección de Puente H.	55
Capitulo V	64
5.DISEÑO DEL HARDWARE	64
5.1.Diseño de la etapa de potencia.....	64
5.2.Tarjeta principal (Main Board).....	65
5.3.Sensores.....	66
5.3.1 Cámara TL1402	66
5.3.2 Sensores reflectivos IS471	67
5.3.3 Sensores RFID	68
5.3.4 Sensores Ultrasónicos	68
5.3.5 Sensores Infrarrojos.....	69
Capitulo VI	70
6.RESULTADOS EXPERIMENTALES	70
6.1.Seguimiento de trayectoria basdo en la cámara de video	70
6.2.Carga útil.....	71
6.3.Alcance maximo XBEE	71
6.4Configuración de la red MESH.....	72

Capitulo VII	73
7.CONCLUSIONES Y RECOMENDACIONES.....	73
BIBLIOGRAFIA.....	76
ANEXO	85

LISTA DE FIGURAS

Figura 1. Robot Móvil Individual.....	21
Figura 2. Centro de Control de Robots	22
Figura 3. Elementos de Control para Robots Cooperativos	24
Figura 4. KIVA System.....	25
Figura 5: Vistas isométricas del sistema embebido PRIMER II	28
Figura 6. Robots del sistema <i>KIVA</i> y disposición de manejo de diferentes objetos dentro de una bodega.....	30
Figura 7. Conexión entre diferentes dispositivos en una red inalámbrica, conformando lo que se denomina una nube de malla	31
Figura 8. Uso y distribución de una red <i>mesh</i> una red de emergencias	32
Figura 9. Esquema del comportamiento cooperativo.....	33
Figura 10. Detalle de un conjunto de robots en una arquitectura CEBOT	36
Figura 11. Ejemplo de robots cooperativos homogéneos (SWARM)	37
Figura 12. Diagrama de componentes del sistema de gestión de inventario propuesto.....	39
Figura 13. Bloques Funcionales.....	41
Figura 14. Cámara de Video on Board	42
Figura 15. Bandas de Frecuencias ISM.....	43
Figura 16. Transponder RFID ID-20	44
Figura 17. Motores usados para el sistema de tracción de los agentes y para el sistema de carga.....	45
Figura 18. Modelado del agente por medio del software Sketchup	46
Figura 19. Rueda de goma con el soporte en acrílico fundido	47
Figura 20. Modelos de ruedas mecanum usados en sistemas de carga	48
Figura 21. Arquitectura PIC 32.....	51
Figura 22. Tarjeta Arduino Mega	52

Figura 23. Flexis QE128	52
Figura 24. Tarjeta de depuración OSBDM diseñada en colaboración en el foro de freescale	54
Figura 25. Esquemático de un Puente H tradicional	55
Figura 26. Arquitectura del Puente H L298	59
Figura 27. Arquitectura del Puente H L6205	60
Figura 28. Arquitectura del puente H DRV8402.....	61
Figura 29. Vista isométrica de un DRV843262	62
Figura 30. Diagrama de conexión de la fuente auxiliar	65
Figura 31. Imagen de la tarjeta principal.....	66
Figura 32. Vistas de modelo de ruedas Mecanum.....	85
Figura 33. Vistas de los moldes obtenidos luego del proceso de mecanizado	86
Figura 34. Piezas en aluminio antes y despues del refrentado.....	86
Figura 35. Diseño final en Solid Works, rodillos definitivos incluidos	87
Figura 36. Bloque de memorias y RTC.....	88
Figura 37. Etapa de Alimentación del sistema	89
Figura 38. Circuito de control	90
Figura 39. Celda puente H (4 celdas por tarjeta)	91
Figura 40. Layout tarjeta de potencia (top-rojo/bottom-azul)	92
Figura 41. Layout tarjeta principal, bloque de polarización y banco de memorias (top-rojo/bottom-azul).....	93
Figura 42. Layout de las tarjetas de expansión PRIMER 2 (Agentes)	94
Figura 43. Layout de las tarjeta de expansión PRIMER 2 (Xbee, mando del modulo de control)	94
Figura 44. Modulo de control compuesto por un PRIMER 2	113
Figura 45. Periféricos empleados.	113
Figura 46. PRIMER 2 conectado adecuadamente al módulo de comunicación ..	114
Figura 47. Pantalla de inicio del PRIMER 2	115
Figura 48. Menú principal del PRIMER 2.....	115

Figura 49. GUI para control remoto de moviles	116
Figura 50. Aplicación en espera de comandos	116
Figura 51. Aplicación enviando comandos a los móviles	117
Figura 52. GUI para pedidos de inventario	118
Figura 53. Visualización de items en el pedido	119
Figura 54. Aplicación enviando la lista del pedido al sistema	119

TABLAS

Tabla 1. Características de microcontroladores Freescale	53
Tabla 2. Caracterización de Dispositivos de control de motores.....	58

LISTA DE ANEXOS

ANEXO A: Construcción y fabricación ruedas Mecanum	85
ANEXO B: Esquemáticos y negativos de las tarjetas	88
ANEXO C: Código fuente PRIMER 2.....	95
ANEXO D: Guía para cargar aplicaciones en el PRIMER 2	111
ANEXO E: Manual de usuario del módulo de control	113

RESUMEN

TITULO: DISEÑO Y CONSTRUCCION DE UNA PLATAFORMA ROBOTICA COOPERATIVA BASADA EN EL SISTEMA EMBEBIDO PRIMER 2 PARA LA GESTION Y MANEJO DE INVENTARIOS¹

AUTORES: RAUL HERNANDO CADENA CONTRERAS²
JHONATTAN FREDY MORENO BERNAL²

PALABRAS CLAVES: Robots, Agentes, Cooperativo, PRIMER 2, Sistema Operativo Embebido, ZigBee, RFID, MECANUM.

DESCRIPCION:

La robótica cooperativa nace como posible solución para problemas complejos mediante el uso de diferentes agentes cooperativos que mediante la interacción entre estos y entre los agentes con el medio, logran cumplir con la ejecución de tareas complejas.

Este trabajo presenta el proceso de diseño y construcción de tres robots móviles que interactúan de forma cooperativa para la gestión y manejo de inventarios, en donde cada uno cuenta con un PRIMER 2 como sistema de control central, el cual a su vez ejecuta las aplicaciones de control y gestión de inventarios sobre un sistema operativo embebido.

En cuanto al diseño electrónico, cada agente o robot, cuenta con un coprocesador encargado del intercambio de información entre el PRIMER 2 y los sensores; el sistema hace uso de tecnología RFID para ubicación dentro de la bodega, un arreglo de sensores en disposición de "scanner" para la navegación y un modulo de comunicación inalámbrica ZigBEE para la transferencia de información entre los agentes y el usuario mediante una red MESH. En cuanto al diseño mecánico, lo más relevante, recae en el diseño y construcción de ruedas MECANUM para brindarle al robot la habilidad de movimiento holonómico.

¹ Proyecto de Grado

² Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.
Director: Msc. Jorge Hernando Ramón Suárez.

ABSTRACT

TITLE: DESING AND CONSTRUCTION OF A COOPERATIVE ROBOTIC PLATAFORM BASED ON THE EMBEDDED SYSTEM PRIMER 2 FOR INVENTORY MANAGEMENT³

AUTHORS: RAUL HERNANDO CADENA CONTRERAS⁴
JHONATTAN FREDY MORENO BERNAL⁴

KEY WORDS: Robots, Agents, Cooperative, PRIMER 2, Embedded Operative System, ZigBee, RFID, MECANUM.

CONTENT:

The cooperative robotic systems born as a possible solution for complex problems using different cooperative agents through the interaction between them and the medium for achieve complex tasks.

This work presents the process of design and construction of three mobile robots that interact cooperatively for inventory management, where each one of them has a PRIMER 2 for central control system. This PRIMER 2 features an embedded operating system, which runs the inventory management applications.

About the electronic desing, each agent or robot have a coprocessor for for the exchange of information between the PRIMER 2 and the robot sensors; the system uses the RFID technology for location in the warehouse, an scanner type array of sensors for line following and one ZigBEE communication module for data transfer between robots over a mesh grid tipe network. And about the mechanic desing, the more relevant of this project is the use of MECANUM wheels for holonómico movements of the robot.

³ Final Graduate Project.

⁴ Physical- Mechanical Engineering Faculty. Electrical, Electronic and Telecommunications Engineering School.
Director: Msc. Jorge Hernando Ramón Suárez

Capítulo I

1. PRESENTACIÓN

1.1 Introducción

La idea de trabajar con un sistema de robots de trabajo en equipo nace en el grupo de Investigación de Electrónica y Robótica Aplicada (ERA) como respuesta a la iniciativa de explorar en el uso de sistemas cooperativos automatizados, ya que en su misión contempla incentivar y promover proyectos académicos y de investigación relacionados con la Ingeniería Electrónica.

El manejo adecuado de un inventario dentro de una empresa o dentro de un sistema de gestión de mercancías como lo puede ser las centrales de mensajería, bibliotecas, hospitales, cadenas de restaurantes, entre otros, es de vital importancia para prevenir el colapso del servicio; de manera tal que un buen manejo y control de inventarios permite optimizar los tiempos de entrega cuando la bodega es de enormes proporciones. Esta investigación está proyectada hacia aquellas empresas en donde la manipulación y gestión de inventarios se convierte en algo tedioso y complejo, ya que gracias a su implementación se consigue una solución óptima, económica y eficaz a este tipo de problemas.

La propuesta a desarrollar con este trabajo pretende hacer una contribución en el desarrollo de una plataforma robusta, que permita llevar a cabo distintas aplicaciones del tipo cooperativo bajo un entorno semicontrolado.

1.2 Estado del Arte

Durante los últimos años, los mayores esfuerzos de la investigación estaban enfocados a mejorar la actuación de un robot móvil individual utilizando sensores avanzados, actuadores y algoritmos de control inteligente. Esto estuvo motivado principalmente por la necesidad de realizar tareas complejas requeridas por aplicaciones del mundo real. Como resultado, los robots móviles individuales se han vuelto muy sofisticados. (Ver figura 1)



Figura 1. Robot Móvil Individual

Una alternativa de investigación es utilizar varios robots más simples para lograr tareas complejas por medio del funcionamiento cooperativo [\[1\]](#). Esta alternativa ofrece ventajas potenciales en términos de flexibilidad de funcionamiento y tolerancia a fallos debido a la redundancia en el número de robots móviles disponibles [\[2\]](#).

A medida que el número de robots móviles aumenta, controlar dicho sistema se vuelve cada vez más complejo. Los métodos para manejar tal complejidad incluyen dos clases de control: centralizado y el no centralizado. Más específicamente, en un método de control centralizado todas las funciones de

CAPITULO 1. PRESENTACIÓN

planificación y decisión se realizan en un solo centro de control. Cada robot móvil contiene pocos sensores simples para el control y guiado, el actuador para el funcionamiento, y los medios de comunicación para intercambiar los datos con el centro de control. Todos los movimientos y conflictos entre los múltiples robots móviles se resuelven fácilmente en el centro de control. La desventaja del sistema es que el sistema se detiene si hay alguna falla en el centro de control. En contraste, un método del control no centralizado consiste en equipar cada robot móvil con múltiples sensores e incluir el controlador para que reconozca su entorno y planifique las acciones de control. [3]

En cualquier situación imprevista, el robot está capacitado para planificar un nuevo camino o encontrar una solución sin esperar las ordenes de un centro de control. La función del centro de control solo se limita a la transmisión del flujo de información de los robots y la asignación de tareas del sistema.

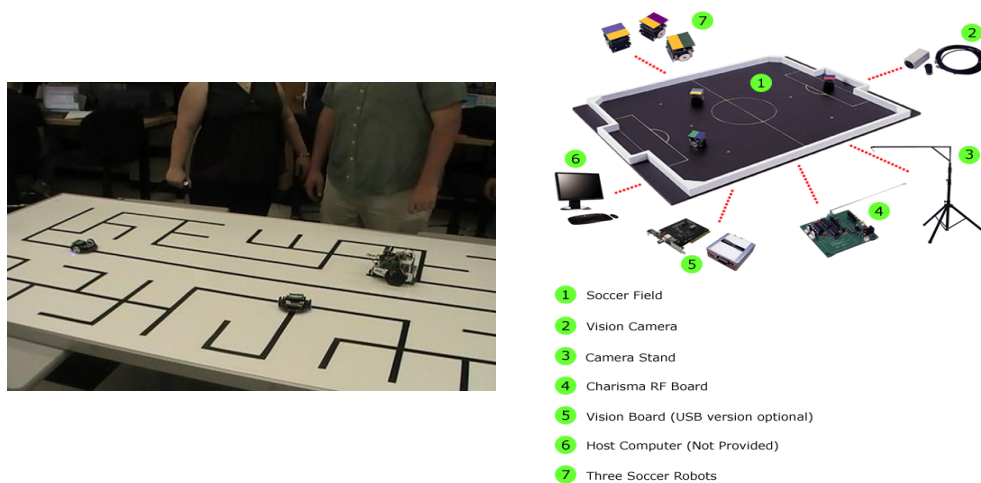


Figura 2. Centro de Control de Robots

Un objetivo primario en el desarrollo de equipos de robots móviles es sintetizar el comportamiento cooperativo de los múltiples robots móviles. Los comportamientos cooperativos habilitan a un equipo de robots móviles a lograr misiones con las que

no se puede lograr fácilmente con un solo robot. El guiado de un robot móvil involucra su localización en el entorno. [4] La localización precisa se vuelve sobre todo pertinente en el caso de compartir múltiples robots móviles un entorno común.

Los primeros intentos para establecer una relación entre el control automático y los robots que realizan una tarea compartida, están referidos a la formulación del espacio operacional de Khatib (1987) [5]. Durante los años 80 del siglo XX, los resultados más importantes consideraban la evolución del contacto durante la manipulación y transporte, lo que requiere una combinación perfecta del control de posición y fuerza. Algunos de los primeros enfoques que persiguen este objetivo son presentados en Li et al. (1989) [6] y Cole (1990) [7], donde se considera explícitamente la dinámica del objeto.

En Parra y Arimoto (1996) [8], Liu et al. (1997) [9] y Parra-Vega et al. (2001) [10], se diseñaron esquemas de control que no necesitan del conocimiento de la dinámica del objeto, sino de las restricciones de movimiento. Los enfoques de control presentados en estos trabajos tienen la ventaja de que no requieren del conocimiento exacto de los parámetros del sistema, dado que se introduce un enfoque adaptable. Más recientemente, en Schlegl et al. (2001) [11] se muestran algunos avances en enfoques de control híbridos.

El principal tópico que tratan los estudios sobre robots cooperativos, son el control de su manejo, actitud y trayectoria. Investigaciones como el control de robots cooperativos por medio de visión artificial de la Universidad Nacional de San Juan Argentina, los mini robots cooperativos de la Universidad Pontificia Bolivariana o el Control de Robots cooperativos con diseño de un observador de la Universidad Autónoma de México; son unos de los tantos trabajos investigativos que se han

desarrollado en el campo de la Robótica en Latinoamérica compartiendo el concepto que cada robot estará en capacidad de desplazarse en forma autónoma en un ambiente controlado, detectar obstáculos, medir distancias, intercambiar información con los demás robots de la colonia vía radiofrecuencia para cumplir una tarea conjunta. (Ver figura 3)

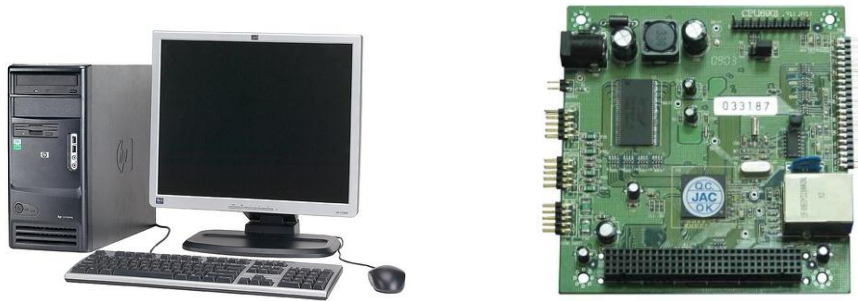


Figura 3. Elementos de Control para Robots Cooperativos

Con el desarrollo de estas plataformas se da un paso hacia la implementación de algoritmos de sistemas cooperativos como respuesta a las necesidades de un sistema de gestión eficiente de recursos, además, se obtendrá un medio para la implementación de sistemas robóticos robustos que permitirán continuar con los estudios de las diferentes topologías en robótica cooperativa.

Si bien la idea de robots que trabajen de manera conjunta en la resolución de problemas como la de organizar una bodega, no es nueva, en los últimos años se ha desarrollado fuertemente dando como resultado alternativas de control, gestión y despacho de mercancías de forma automatizada muy innovadoras y eficaces, tanto así que ha pasado de ser un negocio de solo implementación y adecuación del entorno, a un diseño estructurado en donde la relación hombre maquina se combinan de forma casi perfecta, un ejemplo de ello es el novedoso sistema de gestión de inventario KIVA, de la empresa KYVA Systems [12].

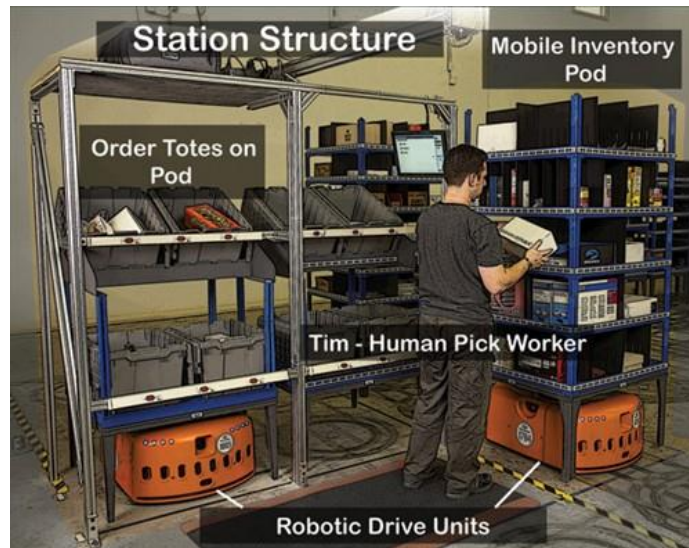


Figura 4. KIVA Systems

Este sistema en particular hace uso de una serie de servidores o estaciones para el despacho y carga de las diferentes mercancías las cuales se encuentran organizadas y etiquetadas adecuadamente en estantes plásticos.

Es de vital importancia sostener un medio de comunicación robusto que permita transferir información ente las centrales y cada uno de los robots o agentes del sistema, para ello se usan topologías que van desde enlaces punto a punto en diferentes tipos de modulación hasta complejos sistemas con pre-procesador para gestión de redes como las de tipo mesh inalámbricas o de tipo estrella en donde se usan modulaciones como saltos en frecuencia en espectro expandido o FHSS Por sus siglas en ingles lo cual brinda mayor robustez al sistema.

El uso de un nuevo sistema embebido, abre las puertas a nuevos desarrollos basados en este tipo de herramientas, ya que una adecuada caracterización de este permitirá que los tiempos requeridos para desarrollos futuros se reduzcan significativamente, así como su uso será indispensable en el diseño de sistemas

CAPITULO 1. PRESENTACIÓN

de bajo costo y tamaño para tareas complejas como se puede ver en los actuales smart phones y otros dispositivos de uso cotidiano, que integran funciones como manejo de agenda, listado telefónico, tareas, calendarios, entre muchas otras a gusto del Diseñador y el enfoque dado al dispositivo.

Capitulo II

2. MARCO TEORICO

Una vez descrito el problema y sus alcances se procede a realizar un estudio sobre los métodos y sistemas que se usan actualmente para una óptima solución según los requerimientos planteados y sus posibles avances referenciados al estado del arte. A continuación se hace una descripción de la documentación recopilada para este efecto:

2.1 Sistema Embebido PRIMER – 2

El sistema embebido *PRIMER-2* [13] proporciona la versatilidad y herramientas necesarias para cualquier tipo de aplicación en donde se necesita tener un sistema integrado de sensores, como acelerómetros, micrófonos, entre otros, sistemas de señalización visual y auditivas como pantallas, leds, y bocinas, y procesamiento de datos de alta velocidad en un módulo de pequeñas dimensiones y permite la expansión de sus capacidades hacia otros dispositivos no incluidos como lo pueden ser, módulos GPS⁵, comunicación bluetooth, zigbee, wifi⁶, entre otros.

⁵Sistema de Posicionamiento Global

⁶es una marca de la *Wi-Fi Alliance* (anteriormente [WECA](#): *Wireless Ethernet Compatibility Alliance*)



Figura 5: Vistas isométricas del sistema embebido PRIMER 2

El sistema embebido PRIMER-2 cuenta con un microcontrolador STM32F103E en cuyo interior se alberga un núcleo ARM Cortex-3M [14], con una velocidad máxima de procesamiento de 1.25 DMIPS/MHz [15] [16], batería de Li-Ion con una autonomía de 6 horas continuas en modo normal, pantalla TFT a color con tecnología *touchscreen*, módulo de grabación/reproducción de audio en formato MP3, slot para memoria micro SD, *transceiver* infrarrojo para transferencia de datos, puerto USB para aplicaciones con PC, acelerómetro de 2 ejes (MEMS⁷) [17] para detección de movimientos laterales, y conector de expansión de 20 pines en donde se puede acceder a los módulos I2C, SPI, USART, CAN, entradas y salidas de propósito general así como entradas analógicas.

No sólo en su *Hardware* son relucientes sus bondades, en lo que a *Software* se refiere, puede ser programado por medio del software Ride7 [18] que permite ejecutar programas en tiempo real (Debugging) con una limitación de 32 Kb de código, o por medio de una licencia GNU ilimitada para compilación y ejecución de programas con todas las librerías requeridas, a gusto del desarrollador basada en

⁷Sistemas Microelectromecánicos

el entorno de desarrollo Eclipse [19]. Posee soporte en línea en la página del fabricante y un foro de discusión en donde hasta el momento se encuentran registrados más de 6000 desarrolladores independientes aportando información sobre cada uno de sus desarrollos.

2.2 Sistema Automatizado de Gestión de Inventario

Los sistemas de inventario tradicionales requieren de cierta experiencia a la hora de gestionar diferentes clases de objetos dentro de una gran bodega [20], además de un número considerable de trabajadores que puedan acatar la demanda requerida de productos en un determinado momento; en sistemas un poco más automatizados se hace el uso de bases de datos y sistemas de información en línea a fin de controlar la cantidad y localización de cada uno de los productos, un ejemplo de ello es el manejo de productos por medio de códigos de barras o más recientemente los llamados sistemas RFID que permiten un grado de robustez mayor que el código de barras tradicional, y un sistema un poco más elaborado que los anteriores consiste en la implantación de maquinaria automatizada a fin de mejorar el rendimiento en la gestión de objetos y su transporte dentro de la bodega, sin embargo este tipo de sistemas automatizados requieren un gran espacio para ser viables debido a su costo y mantenimiento, además de requerir un sistema de control más complejo a fin de sincronizar toda la cadena de transporte dentro de la bodega, y algo que indudablemente es su punto débil, es la construcción y adecuación de la bodega para su correcto funcionamiento.

El sistema *KIVA* presenta una serie de ventajas tanto de maquinaria, costos y energía que hace posible la gestión de productos de una manera ordenada, cómoda, eficiente, y su flexibilidad lo hace aplicable a cualquier tipo de bodega, sin necesidad de hacer mayores cambios estructurales, redes u otras adecuaciones requeridas [21].

KIVA hace uso del sistema de seguimiento de guías demarcadas en la bodega, y códigos de barras en dos dimensiones para permitir su correcta ubicación dentro de la bodega [22], dichas guías y códigos son fácilmente removibles, de modo que se pueda transportar todo el sistema de un lugar a otro y reacondicionar en función de la forma y el tamaño de la bodega, adicionalmente el sistema cuenta con un servidor centralizado que comanda cada una de las actividades de los robots encargados de transportar, reubicar y ordenar los objetos dentro de la bodega [23].



Figura 6: Robots del sistema *KIVA* y disposición de manejo de diferentes objetos dentro de una bodega

2.3 Redes de Comunicación en Malla (MESH)

2.3.1 Definición. La red de comunicación de tipo *mesh* [24] se define como aquella topología que permite la comunicación inalámbrica entre diferentes dispositivos por medio de una red de tipo malla, garantizando de esta manera una cobertura permanente entre todos los nodos de comunicación, compartiendo datos entre estos y realizando a su vez la conexión dinámica entre ellos. El área de cobertura del conjunto de nodos que trabajan como una sola red se denomina en algunos casos como nube de malla.

2.3.2 Ventajas. El acceso a la nube de malla depende de la distribución y la cantidad de nodos que trabajan en armonía unos con otros para crear una red de

radio [25]. Una malla de la red es fiable y ofrece redundancia. Cuando un nodo ya no puede funcionar, bien sea por falla en el hardware o debido a que pertenece a un sistema portable y queda fuera del alcance de los demás, el resto de los nodos pueden todavía comunicarse unos con otros, directamente o por medio de uno o más nodos intermedios.

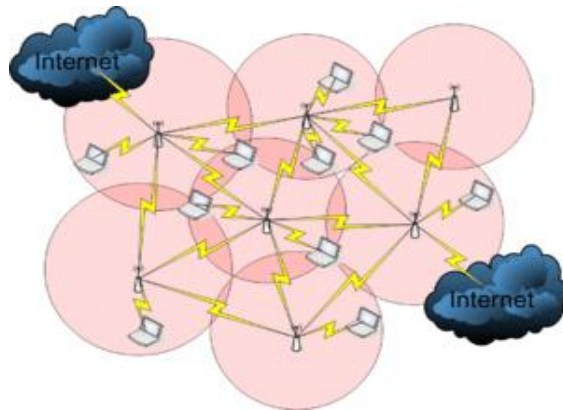


Figura 7. Conexión entre diferentes dispositivos en una red inalámbrica, conformando lo que se denomina una nube de malla.

Este tipo de infraestructura puede ser descentralizado (sin servidor central) o de gestión centralizada (con un servidor central), ambos son relativamente baratos, y muy fiables, en un momento determinado un nodo puede actuar como *Router* para transmitir datos desde los nodos cercanos a los compañeros que están demasiado lejos de los demás, lo que resulta en una red que pueden extenderse a grandes distancias.

2.3.3 Aplicaciones. La nube de malla puede implicar cualquier tipo de dispositivos fijos o móviles [26]. Las soluciones son tan variadas como las necesidades de comunicación, por ejemplo, en entornos difíciles, como las situaciones de emergencia, túneles y plataformas petroleras, aplicaciones de

vídeo móvil a bordo de transporte público o de tiempo real en las carreras de automóviles, para monitoreo de variables remotas o telemetría.

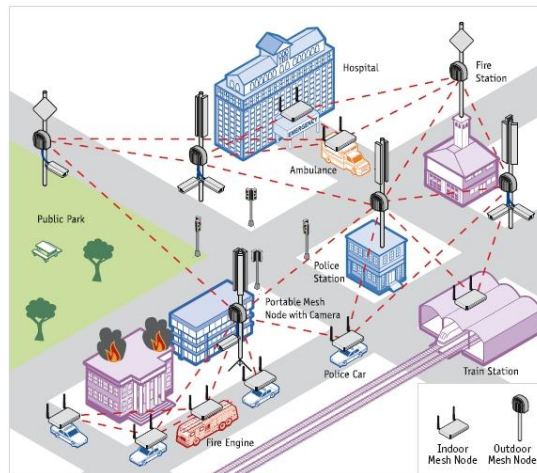


Figura 8. Uso y distribución de una red *mesh* una red de emergencias.

2.3.4 Operación de la Red en Malla. El principio es similar a la forma en la forma en que viajan los paquetes de datos en Internet con conexión de cable [27], los paquetes de datos “saltarán” de un dispositivo a otro hasta que llega a su destino [28]. En cada nodo existe un algoritmo de encaminamiento dinámico permitiendo que esto suceda. Para llevar a cabo tales protocolos de enrutamiento dinámico, cada dispositivo (nodo) debe comunicar la información de enrutamiento a otros dispositivos en la red. Cada dispositivo determina entonces qué hacer con los datos que recibe, ya sea que los envíe al siguiente dispositivo o los conserve, esto varía según el protocolo utilizado. Adicionalmente el algoritmo de enrutamiento utilizado debe garantizar siempre que los paquetes de datos recorran el camino mas corto hacia su ruta de destino, a fin de optimizar la comunicación entre nodos.

2.4 Sistemas Robóticos Cooperativos

Los sistemas robóticos cooperativos o sistemas robóticos multi-agente (*MultiagentRoboticSystem* – MARS) [29] nacen como respuesta al estudio del comportamiento grupal, solución de conflictos, manejo de recursos, movilización y ubicación espacial en los sistemas inteligentes. Estos sistemas resultan muy interesantes ya que las tareas que son complejas para un solo robot (o Agente), resultan no serlo cuando se trata de un grupo de robots que trabajan juntos para darle solución al mismo [30]. La implementación de estos sistemas tienen muchas ventajas, entre las cuales se tienen: mejor rendimiento, optimización de uso de recursos, reducción de costos y construcción más simple, ya que muchos robots pequeños, económicos y flexibles son mejor opción que un solo robot grande, costoso y susceptible a fallos. De los desafíos comunes que enfrentan estos sistemas están: control de tráfico, manipulación cooperativa y recolección de objetos; a todos estos problemas se les da solución mediante la implementación de reglas de comportamiento, herramientas de planificación de acciones y mecanismos para compartir información. Ahora bien, ¿en qué consiste la cooperación? La cooperación comprende las actividades de Colaboración, Distribución, Respaldo, Organización (para poder optimizar el tiempo de ejecución de las tareas) y Planificación. Solamente cuando todas estas acciones ocurren, se puede hablar de un comportamiento cooperativo.



Figura 9: Esquema del comportamiento cooperativo.

El origen de la cooperación en los sistemas robóticos multi-agentes puede ser originado por cada agente mismo como una iniciativa o convicción propia [31], también puede ser motivado por un agente externo o intervención humana. Existen dos modelos generales para el comportamiento cooperativo, estos son:

- *Modelo Eusocial:* este modelo de comportamiento esta inspirado en el comportamiento social de las colonias de insectos, en donde se requiere de la colaboración por convicción de todos para poder sobrevivir, este comportamiento cooperativo es estimulado o motivado por las amenazas hacia el agente. Por ejemplo, ante la amenaza de bajo nivel poder en las baterías, surge la necesidad de trabajar colectivamente en la misma tarea para garantizar que se cumpla, ya sea empleando los agentes robóticos necesarios para completar la tarea en el menor tiempo posible o llamando a otros agentes en mejor disposición para la ejecución de la misma.
- *Modelo Cooperativo:* Esta basado en el comportamiento social de los animales vertebrados, en donde, la cooperación surge para cumplir con las metas individuales o grupales, es decir, esta motivado por los deseos del agente o de un grupo de agentes por cumplir con la tarea.

2.4.1 Características de los Sistemas Robóticos Cooperativos. Los sistemas cooperativos se pueden clasificar de diversas maneras dando flexibilidad a las acciones a tomar o tareas a realizar, según esto, se puede realizar la siguiente clasificación de los sistemas cooperativos.

2.4.2 Arquitectura de Agrupación.

- Centralizado: Esta configuración centra el control en un solo agente [32], que es quien se encarga de reunir la información, procesarla y a partir de esto,

generar la planificación y órdenes para el resto de los agentes. La desventaja de este modelo, radica en el hecho de dejar el procesamiento y toma de decisiones en un solo agente, que causa la propensión a fallos del sistema.

- Descentralizado: Este modelo distribuye el procesamiento, planificación y ejecución de ordenes repartido entre todos los agentes [33], permitiendo, mediante la división de trabajo que el sistema sea mas eficiente, además permite paralelismo de información, escalabilidad, confiabilidad y tolerancia a fallos; por esto, es que este esquema el mas empleado en los sistemas robóticos multi-agente.

A su vez, la arquitectura descentralizada se divide en dos categorías: Jerárquico (centralización local) y Distribuido (todos los agentes aportan el mismo grado de control).

2.4.3 Diferenciación.

- Homogéneo: Los sistemas homogéneos son aquellos en donde todos los agentes son iguales en todos los aspectos, es decir, no hay diferenciación entre ellos; esto da versatilidad al sistema.
- Heterogéneo: En este caso, existen diferencias entre los agentes, como ventaja, se tiene la especialización en tareas.

2.4.4 Comunicación. Estos sistemas tienen a grandes rasgos tres modalidades de comunicación, estas son:

- Vía Ambiente: cuando el ambiente en si es el medio de comunicación.
- Vía Sensado: esta comunicación se transmite físicamente, a través del contacto o interacción de los sensores de los agentes.

- Vía Comunicación: cuando se emplean herramientas de comunicación para compartir la información, por ejemplo: radio comunicaciones.

2.4.5 Arquitecturas Representativas. Las investigaciones y desarrollos hechos durante el estudio de los Sistemas Robóticos Multi-agente, se han basado en el estudio de los ejemplos de comportamiento cooperativo en la naturaleza. Entre los modelos mas usados para la implementación de dichos sistemas, se tienen:

CEBOT (CellularRoboticSystem): Este sistema se basa en el comportamiento cooperativo de las células [34]. Permite la reconfiguración de sistema, de tal forma que dos o más “células” (nombre que reciben los agentes robóticos en este esquema) se pueden unir en una nueva “célula”.

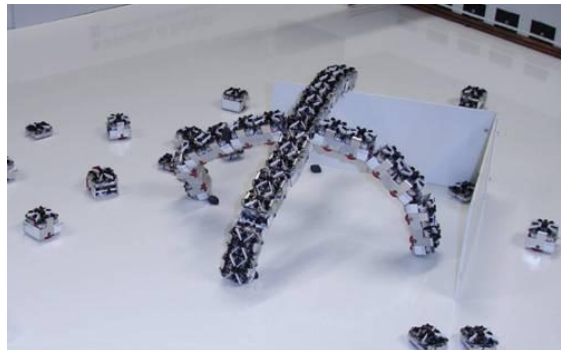


Figura 10. Detalle de un conjunto de robots en una arquitectura CEBOT

Entre las características más importantes se tienen: Descentralización, Organización jerárquica (Existen “Células Maestras”), dinámico, reconfigurable y fluido uso de las comunicaciones [35].

SWARM (Enjambre): Modelo basado en el comportamiento grupal de los enjambres de insectos [36] y nació a partir del estudio del modelo CEBOT y se

caracteriza también por su gran número de individuos (o agentes robóticos), característica que permite que este sistema demuestre poca inteligencia individual, pero gran inteligencia colectiva [37]. Se centra en la división de trabajo (sistema descentralizado y distribuido) y dado que los individuos son todos iguales (homogeneidad), permite versatilidad y respaldo entre agentes.

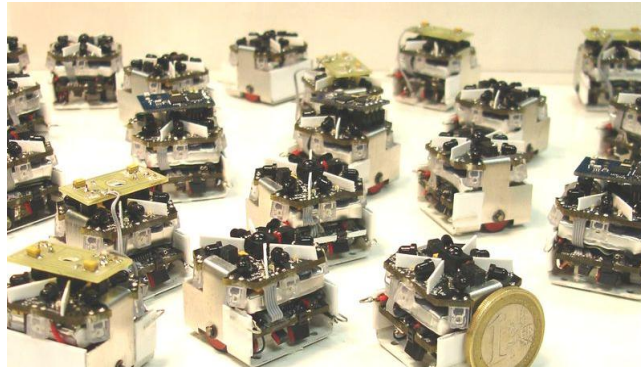


Figura 11. Ejemplo de robots cooperativos homogéneos (SWARM)

GOFER: Este es un sistema centralizado [38] en donde existe un agente encargado de toda la tarea de procesamiento de datos, planeación y asignación de tareas. Además, este agente central es consciente de todos los demás agentes y esta coordinándolos constantemente. Por otra parte, los demás agentes solo se encargan de seguir las órdenes del agente central, así como de reunir-transmitir información y ejecutar las tareas asignadas.

2.4.6 Solución de Conflictos de Recursos. Para un sistema robótico, los recursos son: el espacio, las comunicaciones, las tareas entre otros. Al tratarse de un sistema en donde interactúan muchos agentes [39], puede darse el caso en el que uno o mas de dichos agentes requieran hacer uso de un recurso, y es allí donde surgen los conflictos por la asignación y uso de los recursos en cuestión. Para evitar estos inconvenientes, se requiere que el sistema administre, planifique y asigne adecuadamente los recursos. Esto puede hacerse, mediante la implementación de planes y comportamientos pre-establecidos que se ejecuten de forma rigurosa, es decir, definiendo comportamientos y planes que se deben seguir según sea el caso. Existen tres diferentes formas de evitar y sobrellevar estos inconvenientes, estas son:

- Control sobre el uso estricto y uso restringido de los recursos.
- Definición de múltiples opciones para el uso de los recursos (según sean las condiciones del sistema) y los agentes seleccionando “inteligentemente” la mejor de ellas.
- Definición de múltiples opciones para el uso de los recursos y un agente central encargado de la administración de los recursos hacia los demás agentes.

En los desarrollos e investigaciones actuales en este campo, lo que se esta buscando, es un sistema multi-agente inteligente que seleccione la mejor forma de administrar los recursos, en donde las decisiones recaen en todos los agentes.

Capítulo III

3. DESCRIPCIÓN DEL SISTEMA

Descripción global y sus partes fundamentales.

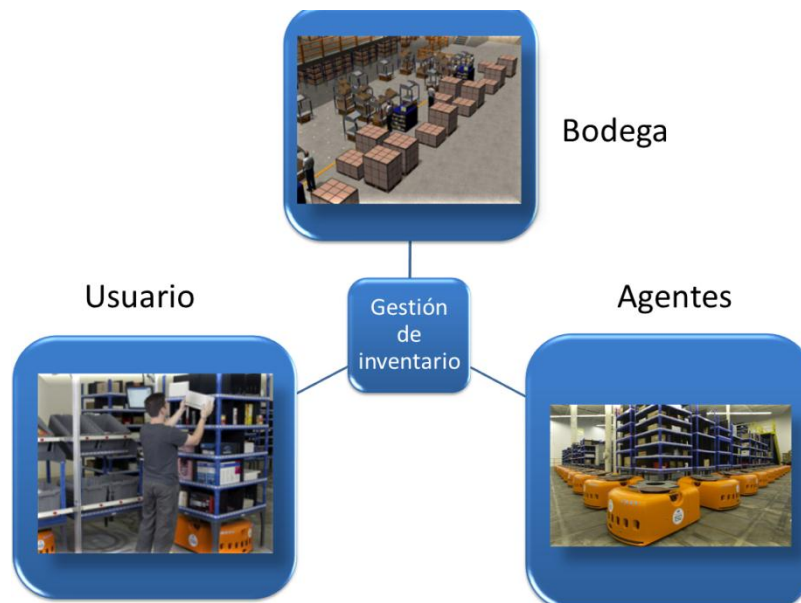


Figura 12. Diagrama de componentes del sistema de gestión de inventario propuesto.

Es sistema para gestión de inventario está conformado por tres partes fundamentales: el área de trabajo o bodega, los agentes y los módulos de comandos manejados por el usuario. El primero por ser variante según su área y disposición no se hará mucho énfasis, puesto que es más una restricción que un diseño como tal. El segundo de ellos, integra un *sistema de procesamiento embebido con su sistema de sensores básicos* para seguimiento de trayectorias y con los aditamentos mecánicos para el transporte de la mercancía, así mismo

CAPITULO 3. DESCRIPCIÓN DEL SISTEMA

cada agente comparte la información del inventario y la ubicación de los demás agentes respecto a un punto de origen determinado el cual llamaremos inicio o cero espacial. El número de agentes puede variar dependiendo de múltiples factores como el espacio requerido, y los requerimientos de optimización en la gestión de inventario, así mismo se hace evidente el uso de al menos un agente para el funcionamiento, El número máximo de agentes del sistema estará determinado por la capacidad de memoria de cada agente ya que comparten información de posición de cada uno de ellos.

El segundo componente es el *módulo de comando*, el cual está soportado en el sistema embebido PRIMER-2, este sistema posee un diseño ergonómico con una pantalla a color e interfaz *touchscreen*, además de contener todos los dispositivos electrónicos necesarios para crear una interfaz de comunicación directa entre el sistema de gestión de inventario y el usuario. Cabe resaltar que el sistema de gestión está compartido entre los agentes, por lo que se puede asociar a un sistema distribuido; este módulo solo se requiere para generar ordenes al sistema, así mismo permite cargar tareas repetitivas en los agentes, de forma que una vez se ha cargado la aplicación en los agentes, ellos lo harán automáticamente sin supervisión del módulo de comando. Dentro de las muchas posibilidades que el sistema presenta en cuanto a flexibilidad, se pueden usar múltiples módulos de comando, ya que el sistema está capacitado para esto a nivel de hardware, pero no ha sido implementado en software en cada uno de los agentes por lo cual se deja para trabajos posteriores.

3.1 Agentes

3.1.1 Componente electrónico. Cada agente se compone de bloques funcionales agrupados por funciones específicas:

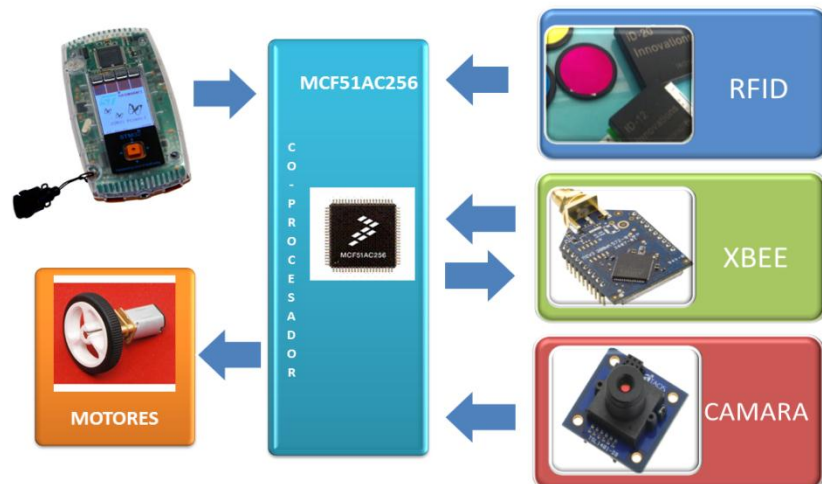


Figura 13. Bloques Funcionales

3.1.1.1 Sistema Embebido: Está soportado en un procesador ARM Cortex M3 el cual hace parte del PRIMER-2, allí se encuentra albergada una tarjeta SD que contiene el inventario de la bodega a intervenir, un Buzzer para generación de alarmas y un sistema operativo de tiempo real CIRCLE-OS proporcionado por st microelectronics [40] en alianza con raisonance [41] el cual es de distribución libre y permite editar el código fuente según las características requeridas por el usuario, así mismo sobre este RTOS se corre una aplicación dedicada a la supervisión del sistema de manera global, esto permite estratificar las tareas según prioridades.

3.1.1.2 Sistema de Control (pre-procesador): Este es el sistema que da robustez al agente en cuanto a la parte electrónica se refiere, pues consta de un microcontrolador MCF51AC256 el cual se encarga de monitorear los sensores

para evasión de obstáculos, manipular la red inalámbrica, controlar la trayectoria del agente y tener una copia de seguridad de la ubicación de los demás agentes, al ser este dispositivo un sistema vital para el funcionamiento se requirió de una selección cuidadosa en donde se tuvieron en cuenta parámetros como consumo, tamaño, capacidad de procesamiento, memoria de sistema y protecciones industriales (EMI, ECM), en este microcontrolador se programaron las rutinas de control de trayectoria, el control de velocidad de los motores.

3.1.1.3 Cámara de video on board: el procesamiento de imagen proveniente de un sensor óptico que consiste en un array de 128x1 pixeles, acondicionado a un sistema de lente óptico con ajuste de enfoque y filtro óptico infrarrojo, el chip que contiene el array de sensores es de referencia TLS1402 y es ensamblado junto con el sistema de enfoque por la empresa parallax[42], dicho sensor entrega una señal analógica en escala de grises y el barrido de la imagen se controla por medio de un reloj de sincronismo. Su simplicidad en el manejo de la señal pre-procesada, el bajo costo, y el tamaño fueron factores primordiales para su selección.

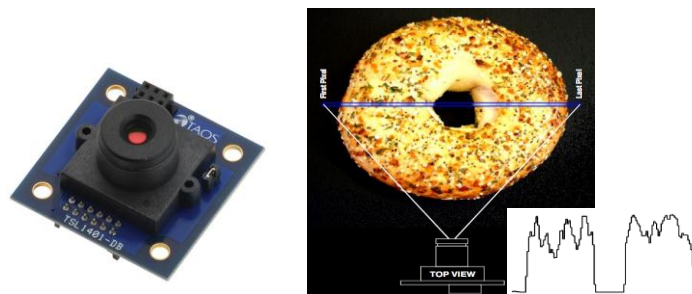


Figura 14. Cámara de Video on Board

El uso de una cámara de video como alternativa al seguimiento de trayectorias, si bien no es nuevo, proporciona robustez al sistema, para dar soporte a esta implementación, se basó en el sistema de seguimiento con cámara desarrollado

CAPITULO 3. DESCRIPCIÓN DEL SISTEMA

en el Grupo ERA para la asignatura de Control Digital, el cual funcionó satisfactoriamente bajo un entorno de iluminación *In-Door*.

3.1.1.4 Red Inalámbrica xbee: Para la implementación de la red inalámbrica se hizo uso de los módulos Xbee de la empresa Digi internacional [42] el cual permite de manera sencilla implementar redes mesh, para esto se requiere que uno de los módulos trabaje en modo coordinador ya que es el nodo principal o pivote de la red, la gestión de esta red la hace el primer por medio del co-procesador.

Estos módulos trabajan en la frecuencia de 920 Mhz, lo que permite reducir significativamente las interferencias producto de dispositivos que trabajan en la banda libre de 2.4 Ghz como routers WiFi, bluetooth, electrodomésticos entre otros, en el que se usan normalmente estas redes inalámbricas, cabe resaltar que las frecuencias de 2.4Ghz y 900 Mhz hacen parte de las bandas denominadas ISM (industrial, Scientific and Medical) por sus siglas en inglés, las cuales están reguladas internacionalmente.

REPÚBLICA DE COLOMBIA
 MINISTERIO DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES
 ATRIBUCIÓN DE BANDAS DE FRECUENCIAS DEL ESPECTRO RADIOELÉCTRICO

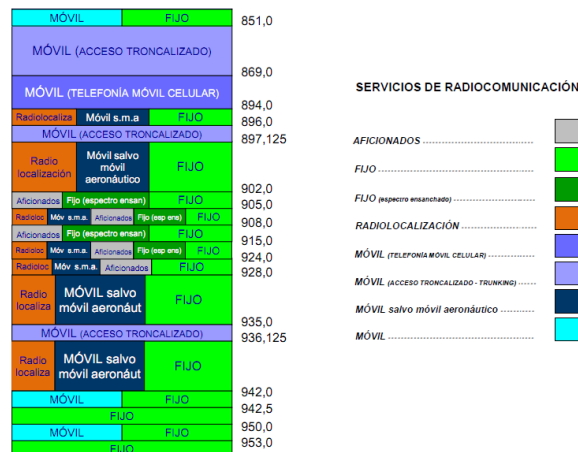


Figura 15. Bandas de Frecuencia ISM

3.1.1.5 Lectores RFID: Son dispositivos que permiten identificar distintos objetos por medio del sistema de Identificación por radiofrecuencia (RF-ID) [43]; generalmente los identificadores o tags son adheridos al producto o mercancía, que en la mayoría de los casos se despacha sin retirarlos, lo que aumenta de cierta manera el costo del manejo dentro de la bodega, para dar solución a este inconveniente, los tags rfid son colocados en sitios estratégicos de la bodega a fin de dar una ubicación espacial de la situación de la misma, de esta manera solo se requiere de unas decenas de ellos para el correcto funcionamiento del sistema de ubicación de ítems en bodega, adicionalmente se pueden encontrar de diferentes clases, frecuencias y nivel de seguridad; así como tags de lectura y escritura. Para esta aplicación específica no se requiere de seguridad ya que los tags se emplean solo como puntos de referencia, la frecuencia de trabajo está en la banda de 125 khz, la cual es una de las más difundidas en el mundo [44], por motivos de tamaño (la antena para esta frecuencia no es tan voluminosa) y consumo de energía del transponder, como por regulaciones internacionales para sistemas de radio frecuencia.



Figura 16. Transponder RFID ID-20

Cada agente cuenta con un transponder RFID para la detección e identificación de los puntos de referencia dentro de la bodega, este transponder es un ID-20 de la empresa ID-INNOVATIONS [45], el cual opera bajo protocolos de comunicación serial ASCII o WIEGAND26 [46] configurable por el diseñador mediante hardware. Para este caso, dichos transponders están configurados para que la comunicación con el pre-procesador sea mediante el protocolo WIEGAND26,

pues, este protocolo es un estándar industrial en aplicaciones con sistemas de identificación por radio frecuencia (RFID) además, este protocolo posee mayor inmunidad al ruido que el estándar RS232 usado para la transmisión de los caracteres ASCII, lo que le hace aceptable para este caso.

3.1.1.6 Motores: Para la movilización de los agentes se hizo uso de cuatro moto reductores de 120 rpm @ 24 voltios [47], los cuales permiten movilizar una carga efectiva de hasta 80 kg incluido el estante de servicio, adicionalmente permiten el intercambio entre ruedas tradicionales de caucho o silicona y ruedas tipo mecanum [48] [49], las cuales permiten bajo la misma plataforma un sistema holonómico [50], aumentando las posibilidades de control a este tipo de sistemas.

Para el sistema de carga se usó un motor de tsukasa de 180 rpm @ 24 voltios el cual por un sistema de barras en cruz permite levantar el estante de carga/ despacho 5 centímetros sobre la superficie a fin de permitir la movilidad del estante dentro de la bodega.



Figura 17. Motores usados para el sistema de tracción de los agentes y para el sistema de carga

3.1.2 Componente Mecánico

3.1.2.1 Chasis: El chasis fue elaborado en perfil cuadrado de aluminio de 10cm de lado, con el que se consigue poco peso y buena resistencia de soporte para los dispositivos mecánicos y electrónicos que componen el agente. Los soportes de los motores al chasis fueron elaborados en lamina de Acero cold rolled y maquinados en el laboratorio de FMS de la escuela de Ingeniería mecánica UIS, para el diseño del chasis se hizo un bosquejo previo con la herramienta gratuita Google Sketcup [\[51\]](#) en la cual se determinaron las medias máximas de cada agente y la disposición de los motores y demás componentes que ocupan un espacio significativo.

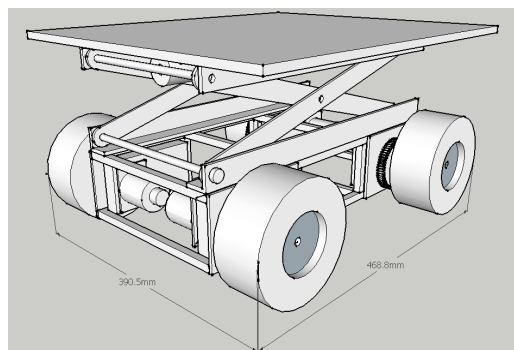


Figura 18. Modelado del agente por medio del software Sketchup

Los parámetros mínimos para el cálculo de las partes que conforman el chasis se basaron en una carga máxima de 50 Kg con área de carga efectiva de 45 cm², y con una restricción de altura máxima del agente de 22 centímetros (con su plataforma contraída).

3.1.2.2 Ruedas: Las ruedas usadas se pueden clasificar de dos clases, las solidas y omnidireccionales (mecanum)

- Solidas: conformadas por un núcleo de polipropileno con recubrimiento en poliuretano blando, pueden soportar una carga de hasta 200 Kg, dentro de la familia de ruedas IMSA-PB [52] se eligieron unas ruedas de 5" de diámetro y se realizó un acople en metacrilato fundido en frio entre la rueda y el eje del motor.



Figura 19. Rueda de goma con el soporte en acrílico fundido

- Omnidireccionales (mecanum): este tipo de ruedas que pertenecen a la familia de ruedas omnidireccionales, se caracterizan por tener una disposición de rodamientos con recubrimiento de caucho a 45° respecto al eje de rotación {ver imagen}, este tipo de llantas son las que permiten que la plataforma se convierta en holonómica, si bien las ruedas sólidas tienen el inconveniente de no permitir movimientos laterales, son las más eficientes respecto a lo que consumo energético se refiere, sin embargo el control de las trayectorias requiere de mayor tiempo pues no posee los grados de libertad necesarios para un desplazamiento más uniforme de un punto a otro, dando una ventaja de las ruedas mecanum sobre las sólidas.

Dentro del mercado de robótica el uso de estas llantas está limitado a pequeños móviles en donde los diámetros de rodadura no superan las 3" de diámetro, existe una gran barrera en la consecución de este tipo de llantas siendo el más notorio las medias de diámetro requerida para el proyecto en cuestión; para algunas

CAPITULO 3. DESCRIPCIÓN DEL SISTEMA

empresas especializadas en este tipo de locomoción como andymark [\[49\]](#) y omnixthecnology [\[53\]](#) solo proporcionan los sistemas completos u ofrecen diámetros diferentes a los requeridos, además su alto costo.



Figura 20. Modelos de ruedas mecanum usadas en sistemas de carga

Debido a esto, se revisaron diferentes modelos existentes y se determinó realizar un juego de llantas con el modelo de soporte lateral sesgado, ya que en la Universidad se disponen de las herramientas y los recursos para fabricación de piezas como el laboratorio FMS de la Escuela de Ing. Mecánica UIS (ver anexo: proceso de fabricación ruedas mecanum).

Capitulo IV

4. SELECCIÓN Y CARACTERIZACION DE COMPONENTES

Para hacer la correcta selección de los componentes se hace necesario fijar unos parámetros mínimos o requerimientos para el correcto funcionamiento del sistema:

- **Lenguaje de programación:** Este ítem es de vital importancia debido a que es la base fundamental para la correcta programación del co-procesador, es requisito indispensable que el lenguaje sea ampliamente extendido y que cumpla con estándares de programación como C, C++, Java entre otros, así mismo se buscará que el uso de dichos compiladores sea de uso libre bajo licencias GNU o en su defecto de tipo estudiantil.
- **Herramientas de depuración:** El sistema debe permitir no solo la programación si no la verificación del código en tiempo real a fin de corregir errores durante la ejecución de las tareas del co-procesador, y facilitando de esta manera la labor del programador reduciendo errores derivados del código (común mente llamados: "buggs").
- **Periféricos:** Dentro de los módulos que debe poseer el sistema, debe contar con bloques de comunicación SPI, SCI e I2C, sistema de interrupciones IRQ y dispositivo de conversión de señales Analógicas a digitales ADC, para que el agente pueda tomar medias del entorno y comunicar estas variaciones a los otros agentes y con el sistema de control y memorias del sistema.
- **Memoria:** El dispositivo co-procesador deberá tener la suficiente capacidad de memoria de programa para soportar extensos códigos y suficiente memoria

volátil para albergar los cálculos derivados del procesamiento de imagen y de la trayectoria a seguir. Es indispensable que el sistema posea una memoria interna o externa lo suficiente mente grane a fin de tener copia de los datos del inventario (Backup) en cada uno de los agentes.

4.1. Selección del Sistema de Control

Dentro de las posibilidades que existen hoy en día para el control de procesos se pueden encontrar desde las FPGAs, CPLDs y demás hasta los microcontroladores con núcleos de 8, 16, o 32 bits usados extendidamente en muchos dispositivos portables. Después de analizar estas posibilidades se decidió usar un microcontrolador de ser posible entre 16 y 32 bits, debido a su auge en la implementación de Sistema Operativos en Tiempo Real RTOS y su mejor relación costo/beneficio, además cumplen con la mayoría de los requerimientos mínimos del sistema.

Una vez fijado el tipo de dispositivo controlador se procedió a seleccionar las casas fabricantes más populares, las cuales gozan de este beneficio por la extensa literatura que se encuentra en la Web y por la cantidad de proyectos desarrollador por diferentes ingenieros y aficionados de la electrónica alrededor del mundo. Dentro de este ramillete de fabricantes se seleccionaron las empresas Microchip con sus micros PIC32, la empresa Atmel [\[54\]](#) con su popular sistema abierto Arduino [\[55\]](#) y Freescale [\[56\]](#) con sus núcleos coldfire de extenso uso en automatización y control industrial.

- Microchip: Su línea de 32 bits es muy robusta e ideal para procesamiento de datos a altas velocidades ya que han optimizado los periféricos. Sin embargo la disponibilidad de un compilador para su programación oscurece sus ventajas pues

CAPITULO 4. SELECCIÓN Y CARACTERIZACIÓN DE COMPONENTES

la disponibilidad es de evaluación por 60 días y el sistema de depuración es costoso.

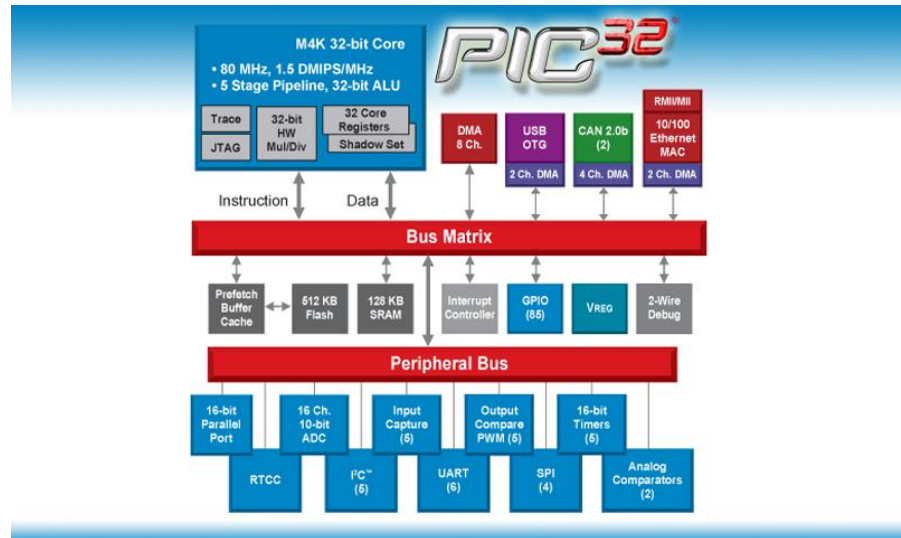


Figura 21: Arquitectura PIC32

- Atmel: Líder en sistemas basados en la plataforma Arduino y una de las más populares para pequeñas aplicaciones, dando facilidad de programación al utilizar el entorno de programación processing también difundido ampliamente, dentro de las desventajas del sistema arduino están la limitación de aplicaciones industriales y por no disponer de núcleos de más de 8 bits para esas plataformas, otra desventaja predominante es la ausencia de depurador paso a paso, pues hace uso del puerto serie como terminal de control lo que genera retardos no deseados en el funcionamiento del programa.

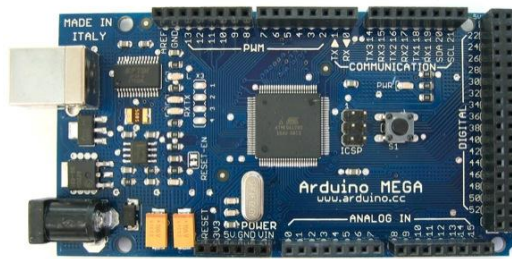


Figura 22. Tarjeta Arduino Mega

- Freescale: Cuenta con una amplia gama de microcontroladores y con herramientas de programación de uso extendido como codewarrior IDE, el cual en su versión estudiantil está limitado a 64 KB de memoria de programa y posee un sistema integrado de depuración conocido como Hiwave, dentro de las fortalezas está el soporte de hardware necesario para la depuración OSBDM el cual es de distribución libre, las debilidades de esta empresa radica en la baja velocidad de los procesadores al usar arquitectura Von Newman como pilar fundamental de la CPU.

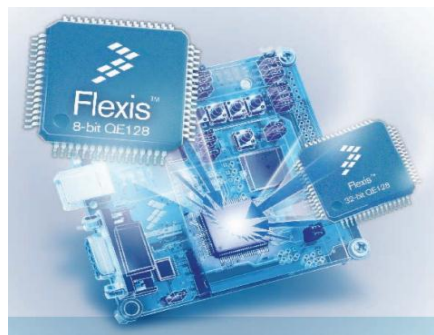


Figura 23. Flexis QE128

De esta pre-selección se decidió por usar un microcontrolador de la familia Flexis de Freescale ya que con las características antes mencionadas se agrega la

CAPITULO 4. SELECCIÓN Y CARACTERIZACIÓN DE COMPONENTES

posibilidad de disponer fácilmente de las herramientas de desarrollo y a un bajo costo.

Seleccionadas las herramientas con las que se va a programar y a depurar, se hizo una comparación entre las diferentes características que poseen los microcontroladores de Freescale entre distintas familias a fin de seleccionar la mejor opción.

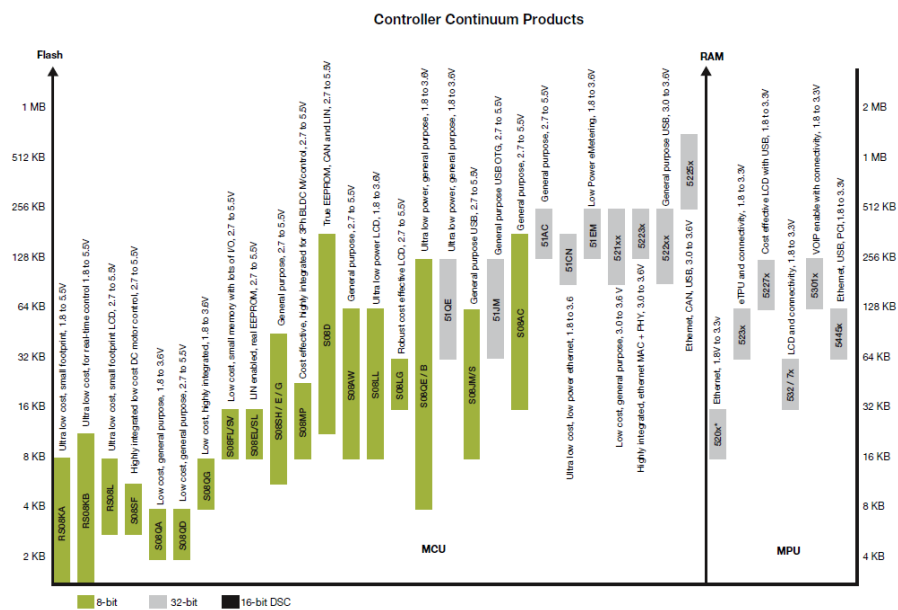


Tabla 1: Características de Microcontroladores Freescale

Teniendo en cuenta la tabla de características proporcionada por Freescale, decidió usar el microcontrolador MCF51AC256 con núcleo Coldfire V1 el cual permite correr el programa tanto en micros de 8 bits como en 32 bits, Además está diseñado para aplicaciones industriales [57], por lo que le confiere mayor inmunidad a EMC/EMI dando un punto a favor importante por el entorno donde se va a desenvolver, así mismo la capacidad de memoria de 256 KB en flash y de 32 KB de RAM lo hacen suficientemente robusto para otro tipo de aplicaciones de control como sistemas Fuzzy entre otros.

También dispone de comunicación SPI, SCI e I2C para dispositivos adicionales como memorias, sensores de distancia, sensores inerciales MEMS, entre muchos otros dispositivos [58].

Otra de las ventajas de la familia de microcontroladores MCF51 de Freescale es la facilidad para hacer depuración en tiempo real gracias a las tarjetas de programación de distribución libre OSBDM y USBDM desarrolladas por PGO en el foro comunitario de Freescale, las cuales fueron adaptadas y optimizadas por el grupo ERA.

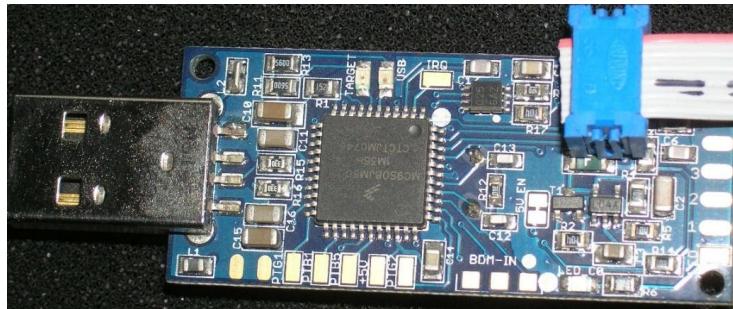


Figura 24. Tarjeta de depuración OSBDM diseñada en colaboración en el foro de Freescale

4.2. Selección de Motores

La difícil consecución de motores en el mercado local obligó de cierta manera a buscar un proveedor internacional en los mercados de china y Estados unidos, siendo estos los mayores fabricante de moto-reductores del mundo; haciendo una comparativa entre capacidades de distintos modelos comerciales disponibles se preseleccionaron los motores de DS-119SS555 de la casa fabricante Dong Shun Motor Company de Hong Kong[47], por su desempeño y capacidad de carga requerida para el sistema, si bien se hizo un contacto directo con dicha empresa, la compra de solo 12 motores para el prototipo a construir impidió la compra de

estos, sin embargo se obtuvo información suficiente para buscar un proveedor local con un lote de motores usados de características similares, con el cual se concretó finalmente la compra.

Para asegurar las características de los motores adquiridos se procedieron a realizar las pruebas de velocidad, consumo en bloqueo y sin carga, siendo estos muy similares a los ofrecidos por Dong Shun Motor Company.

4.3 Selección del puente H

El control de velocidad y sentido de giro de los motores se realiza mediante el uso de un *puente h* (ver figura 18) el cual se encarga de la administración del sentido e intensidad del flujo de corriente para así poder controlar tanto la velocidad como el torque generado por el motor.

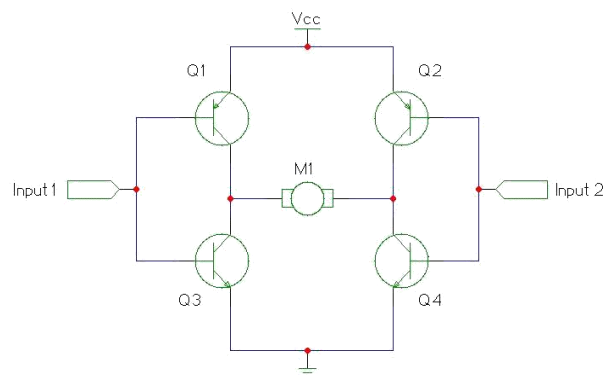


Figura 25. Esquemático de un Puente H tradicional

Según las características eléctricas de los motores que empleará el *agente* y pruebas sobre la plataforma con carga, para las cuales, la plataforma se sometió a el peso de una persona promedio (aproximadamente 75 kg) mientras se encontraba en movimiento, el consumo de corriente por motor puede llegar a

presentar una corriente pico de 3.2 [A], por lo tanto, uno de los requisitos del circuito encargado del control es la corriente que este es capaz de manejar sin sufrir daños.

La selección de dicho es un proceso que debe ser realizado con sumo cuidado, pues, este se encarga de la regulación de potencia eléctrica que es entregada al motor, para ello este circuito electrónico emplea transistores en un arreglo llamado: “puente H”, el cual posee la facultad de administrar el flujo de corriente en magnitud y sentido a través del motor. Los criterios a tener en cuenta para la selección del puente H son:

- Corriente: La corriente eléctrica es el parámetro más crítico a la hora de seleccionar el puente h que se ajuste a los requerimientos del proyecto, en este caso, se requiere de un dispositivo capaz de soportar una corriente pico de 3.2 [A] como mínimo y una corriente de operación nominal de aproximadamente 2.5 [A].
- Tensiones de polarización: Dado que el sistema va a operar con baterías LiPO, las cuales aportan una tensión mínima de 19.2 [V], nominal de 22.5 [V] y una tensión máxima de 25.2 [V]; el puente H debe ser capaz de operar dentro de estos márgenes de operación.
- Dimensiones: El espacio que ocupa cada puente h es crítico para el uso adecuado del espacio, pues todo el sistema de control debe estar ‘on board’ sobre el vehículo. Dentro de las consideraciones de espacio requerido, se debe tener en cuenta el espacio que ocupará el disipador asociado a cada puente h.
- Costos: Por requerir un número considerable de dispositivos, pues cada motor requiere un puente h, el costo por unidad de cada uno es importante para poder estimar y administrar adecuadamente los recursos del proyecto.

- Potencia disipada: todo circuito electrónico tiene el inconveniente de requerir cierta cantidad de energía para poder funcionar adecuadamente, la selección de un dispositivo, en este caso el puente H, que tenga un consumo mínimo para su funcionamiento y que durante su funcionamiento normal la relación de transferencia de potencia sea la más cercana a la unidad es un parámetro vital que determinará la duración de las baterías.

Una vez establecidos los requisitos que debe cumplir el dispositivo seleccionado, se deben encontrar unos dispositivos que cumplan con los requisitos mínimos y para este caso se empleó como un requisito adicional, la facilidad de adquisición, ya sea por medio del mercado local o por intermedio de las muestras que las empresas fabricantes de dispositivos semiconductores tienen disponibles, ya que es necesario poder realizar pruebas con dichos dispositivos para poder calificar el desempeño de cada uno y no es presupuestalmente viable comprarlos para dichas pruebas. A continuación en la Tabla 1 se presentan los dispositivos que fueron pre-seleccionados.

Tabla 2: Caracterización de dispositivos de control de motores

Referencia	Fabricante	V.pol.min. [V]	V..pol.max. [V]	I. max [A]	Arquitectura	Pre-driver	Disipador	Costo	Observaciones.
L298	ST Microelectronics	5	46	2.5 (soporta conexión en paralelo, puede soportar hasta 5 [A])	BJT	N.a.	Montaje externo	4.67USD (8632 COP aprox.)	Perdida de potencia en el puente H, por lo tanto, no se emplea toda la potencia mecánica nominal del motor.
L6205	ST Microelectronics	8	52	2.8 (5.6 [A] Pico y soporta conexión en paralelo)	MOSFET	Bomba de carga	Sobre PCB	12.2 USD (22550 COP Aprox.)	No hay soporte por parte del fabricante para envío de muestras de esta referencia.
drv8402	Texas Instruments	0	52.5	5	MOSFET	Bootstrap	Sobre PCB	17.58USD (32495 COP aprox.)	Según el fabricante: No recomendado para futuros proyectos, discontinuación próxima.
drv8432	Texas Instruments	0	52.5	7	MOSFET	Bootstrap	Sobre PCB	14.52USD (26839COP Aprox) - Stock no disponible.	Presenta problemas de control de polarización al suprimir el PWM (PWM>90%)
drv8828	Texas Instruments	0	47	3	MOSFET	Bomba de carga	Sobre PCB	5.46USD (10092COP Aprox.)	La corriente que soportan no es suficiente para la aplicación, ya que el margen de corriente es muy estrecho

El L298 es un dispositivo de la empresa ST Microelectronics, implementa el puente h empleando transistores BJT, cuenta con dos puentes h, cada uno con capacidad de controlar hasta una corriente máxima de 2.5 [A], por lo que no se ajusta a los requerimientos; pero, es posible conectar 2 puentes h en paralelo brindándole la posibilidad de controlar hasta 5 [A].

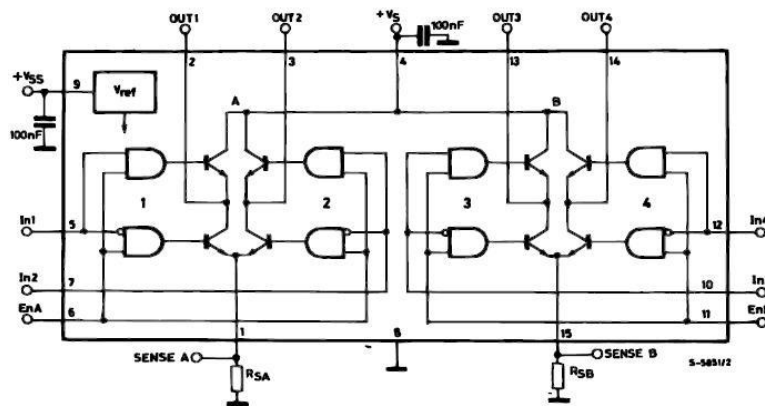


Figura 26. Arquitectura del Puente H L298

El inconveniente con este controlador recae en el hecho de ser un puente H implementado con transistores BJT, y es que se pierde una parte de la potencia de salida en los transistores del puente h, debido a la potencia que se pierde en los transistores del puente h; por lo tanto, debido a esta pérdida de potencia eléctrica en el puente h (se manifiesta a través de energía térmica disipada en calor), la potencia mecánica del motor se reduce.

El L6205 es un puente h de la empresa ST Microelectronics el cual cuenta con transistores mosfet canal N en la arquitectura de su circuito. Este dispositivo, hace uso de un circuito “boost”, para generar la tensión de polarización de los transistores de la parte superior del puente h para que los transistores conmuten adecuadamente.

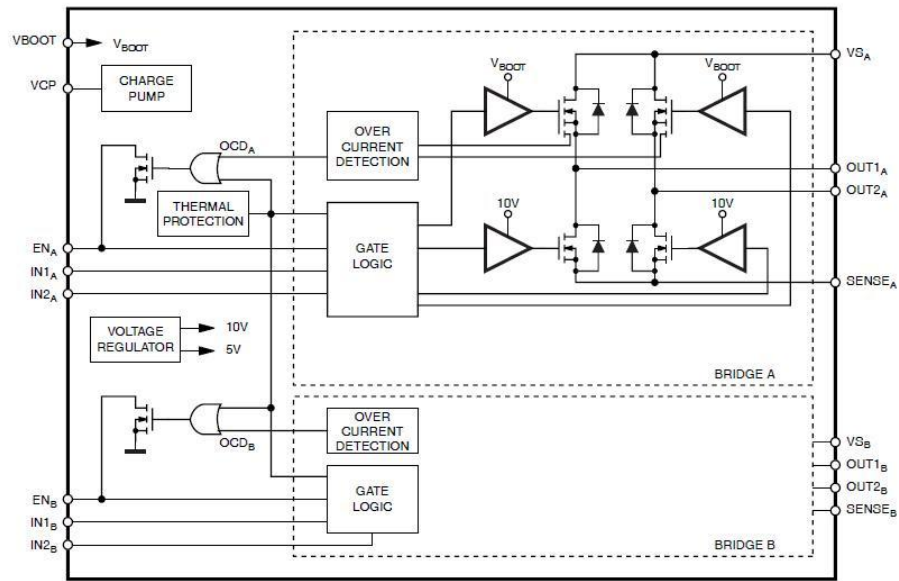


Figura 27. Arquitectura del Puento H L6205

Entre las bondades de este puente h, se tiene la corriente que es capaz de soportar, ya que es de 2.8 [A] continua y una corriente pico máxima de 5.6 [A], si bien, el margen de corriente de operación continua del puente h es bastante estrecho, respecto al consumo de corriente de los motores; el L6205 cuenta con dos puentes h que pueden ser conectados en paralelo, brindando la posibilidad de soportar el doble de corriente, soportando sin problema alguno así la carga eléctrica a la cual estaría sometido.

El único inconveniente es la factibilidad de adquisición del mismo, pues, al requerirse un puente h de este tipo por cada motor y dado su valor comercial, lo hace inviable para ser empleado en este proyecto, además, de no conseguirse con facilidad en el mercado local o nacional, lo que implica adquisición mediante importación, con lo cual los costos aumentan considerablemente. Por otra parte, el fabricante no cuenta con soporte mediante el servicio de muestras gratis para

CAPITULO 4. SELECCIÓN Y CARACTERIZACIÓN DE COMPONENTES

actividades académicas, por lo tanto, la adquisición de un puente h de estos no es fácil.

El DRV8402 es otro puente H fabricado por Texas Instruments que cuenta con las siguientes particularidades: corriente continua máxima de 5[A], disponibilidad en programas de muestras gratis y es un puente h diseñado empleando transistores mosfet, los cuales son polarizados mediante el uso de un circuito bootstrap integrado.

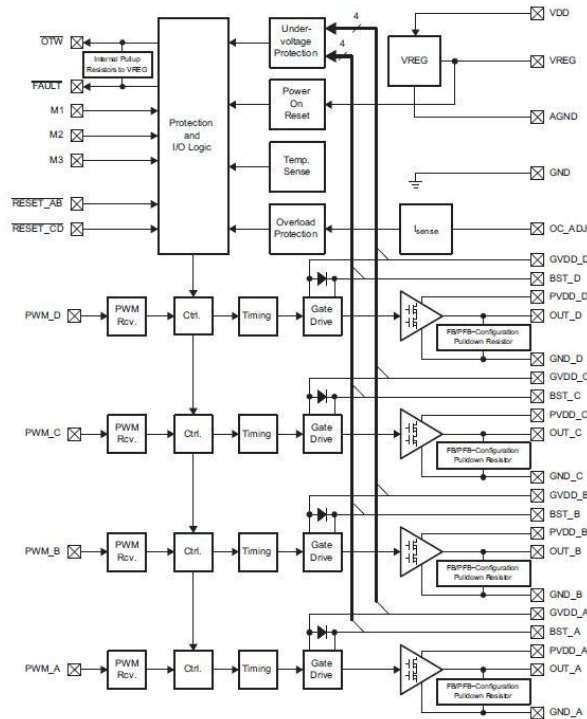


Figura 28. Arquitectura del puente H DRV8402

Entre los inconvenientes que presenta este dispositivo, se tienen: la presentación o empaquetado en la que viene este elemento, se trata de un circuito integrado con un área de disipación de temperatura muy pequeño, por lo cual, puede llegar a fundirse por sobre calentamiento, por ello requiere el uso de una buena porción

del PCB para efecto de disipación de temperatura, por otra parte, el fabricante ya no ofrece soporte sobre este dispositivo, pues, ya se encuentra discontinuado y el stock disponible en las tiendas de venta de elementos y circuitos integrados es limitado además de costoso, pese a disponer de estos puentes h mediante muestras. Luego de realizar montajes de prueba para el circuito, se descarta este dispositivo debido a la poca confiabilidad que demostró al no prestar buen desempeño al controlar los motores que se usan en los agentes.

El DRV 8432 es un puente H fabricado por Texas Instruments de características similares al DRV 8402 pues, la arquitectura de este es similar, pero este dispositivo tiene mejores características en cuanto a disponibilidad en el mercado, así como, mejor desempeño en la disipación de temperatura.



Figura 29. Vista isométrica de un DRV8432

Igualmente, este puente h se puede adquirir fácilmente gracias a los programas de muestras con los que cuenta Texas Instruments, con lo cual la adquisición de estos para propósitos de evaluación y pruebas es una ventaja. Pero, luego de realizar algunas pruebas con él, se encontró un problema que puede ser crítico: la señal de PWM que controla el puente h debe estar siempre presente, pues, si en algún instante esta señal se control se suspende o permanece en un nivel de tensión constante por mucho tiempo mientras este esté energizado y con carga, el puente h entra en un estado en el cual una o más de sus ramas superiores que

son comandadas por transistores MOS pasan del estado de trío a saturación, disipando excesivo calor en pocos segundos y por consecuencia la posterior destrucción del dispositivo. Dentro del análisis realizado se encontró a que al ser un circuito del tipo bootstrap, los transistores superiores de cada rama, pierden su polarización de compuerta, la cual es suministrada por un condensador externo que es cargado en cada ciclo de PWM, pero si se requiere una modulación superior al 90% en PWM (máxima velocidad) el dispositivo falla inminentemente.

Hecha la revisión de los circuitos disponibles en el mercado y sus limitaciones se decidió realizar un circuito de tipo discreto a la medida del sistema, el diseño correspondiente esta explicado ampliamente en el capítulo 6: Diseño del Hardware.

Capítulo V

5. DISEÑO DEL HARDWARE

5.1 Diseño de la etapa de Potencia. Como primera medida se tomó como referencia el tradicional L298 en su esquema lógico de manera que no exista cortocircuito en cada rama por mal manejo en el control de los transistores, luego se procedió a hacer el montaje basado en transistores BJT como el TIP122 o el TIP41, dando como resultado una moderada disipación de calor en cada transistor bajo condición de carga nominal (3 Amperios), adicionalmente cada transistor muestra una pérdida de 4 voltios por transistor, por lo cual se descarta su uso, así mismo se realizaron pruebas basadas en transistores MOS de canal N y P, pero el alto costo de los transistores P y sus rangos limitados de funcionamiento imposibilitaron su implementación, finalmente se decide hacer un montaje basado en transistores MOS de canal N.

Las pruebas realizadas demostraron que se requiere de una polarización adicional en los transistores superiores por al menos 5 voltios arriba del umbral de conducción (V_t), así se garantiza que se encuentren en la región de triodo, en la primera prueba se usaron transistores de potencia de consecución local por lo que sus altos valores de voltaje de umbral (V_t) eran superiores a los 4 voltios, de manera que para su correcta polarización se requería de al menos 9 voltios por encima de la alimentación de los motores, es decir 34 voltios en el peor de los casos.

Con esta limitante se hizo una búsqueda de fabricantes de transistores de potencia que dentro de sus rangos de voltaje de umbral fuera menor al que se tenía el momento de las pruebas, finalmente se seleccionó la familia de

transistores HexFet CSD1750XX de Texas Instruments, del cual por medio de muestras se adquirieron 60 transistores, todos ellos con voltaje de umbral inferior a los 2 voltios.

El último inconveniente presentado recae sobre la necesidad de una fuente independiente adicional de polarización para los transistores superiores en cada una de las ramas del circuito, la solución a esto se resolvió usando una fuente DC-DC aislada la cual provee 5 voltios a la salida la cual se coloca en serie con la batería LiPo.

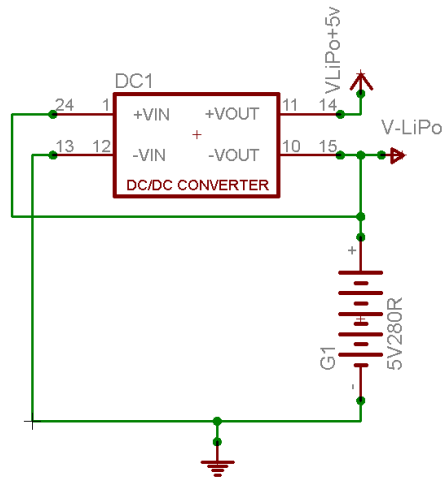


Figura 30. Diagrama de conexión de la fuente auxiliar

El diagrama del puente H completo se encuentra en el Anexo correspondiente a esquemáticos

5.2 Tarjeta principal (Main Board). En ella se encuentra el co-procesador y las etapas de regulación de voltaje requeridas para el funcionamiento de los sensores, el modulo PRIMER de cada agente y demás dispositivos asociados al sistema, adicionalmente posee los conectores que permiten la comunicación con los

periféricos del sistema así como puertos adicionales de expansión para sensores adicionales u otros dispositivos.

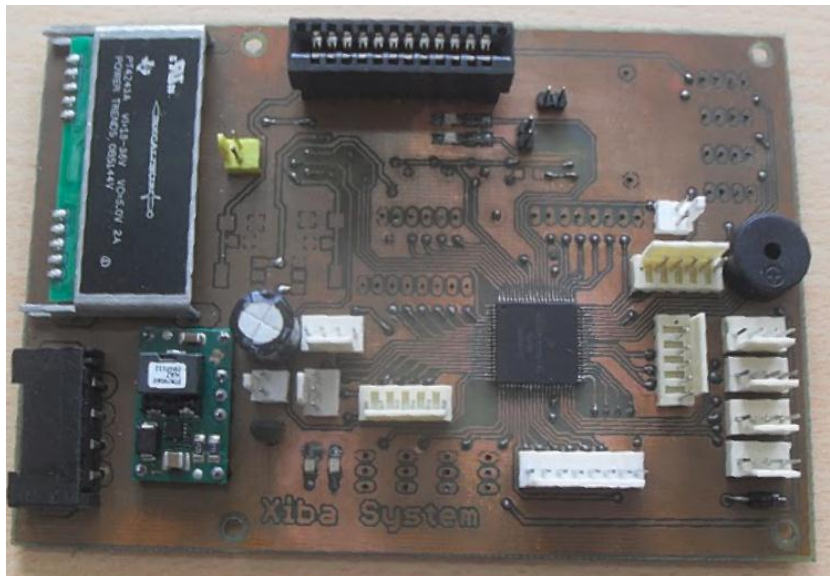


Figura 31 Imagen de la tarjeta principal

5.3 Sensores. El sistema cuenta con capacidad de albergar múltiples sensores de diversa clase y diferentes tipos de comunicación como SPI, I2C, analógicos y de PWM como los ultrasónicos, dentro de los sensores usados encontramos:

5.3.1 Cámara TL1402: Es ensamblada por la empresa Parallax y contiene en su interior el chip TL1402 de TAOS Optoelectronics su señal de control se basa en un tren de pulsos el cual envía por cada pulso un respectivo píxel de nivel de iluminación en escala de grises sobre un rango de 0 a 3 voltios, permitiendo de esta manera el seguimiento de trayectorias con niveles de contraste notorios no críticos.

El control de contraste está determinado por la frecuencia de muestreo de la señal digital (pulsos entre pixeles) y por la tasa de muestreo entre cuadro y cuadro, esto es debido a el circuito de integración de señal que posee el sensor internamente, a medida que se disminuye la frecuencia tanto en la de pixel-pixel como en la de cuadro a cuadro se aumenta el contraste haciéndola más sensible y requiriendo menos iluminación posterior (Back Light).

Uno de los problemas encontrados durante la implementación es la llamada aberración esférica debido al corte esférico del lente usado, cuando se hace una lectura uniforme de una superficie blanca homogéneamente iluminada, la señal analógica entregada por la cámara muestra una no linealidad que se acentúa en el centro de la imagen dando como consecuencia una distorsión de la imagen recibida, uno de los métodos sugeridos por el fabricante consiste en aplicar una regresión lineal basada en componentes cosenoidales de manera que se pueda aminorar el efecto producido por la lente, este método si bien es efectivo trae el inconveniente de realizar cálculos complejos dentro del co-procesador, lo que genera largos periodos de cálculo imposibilitando el correcto ajuste de la tasa de muestreo entre cuadros y limitándola a valores muy estrechos.

El segundo método consiste en generar unas zonas de iluminación no homogénea sobre el área de sensado de manera que se ilumine en la periferia de la cámara y se reduzca considerablemente la iluminación en el centro de esta, de esta forma se disminuye el efecto esférico; cabe resaltar que este método es más impreciso por ser un ajuste meramente experimental, pero permite un resultado que se ajusta a la aplicación.

5.3.2 Sensores reflectivos IS 471: son sensores infrarrojos que se basan en el efecto reflectivo de las ondas infrarrojas para la detección de contraste y/o

objetos a corta distancia, para la detección de objetos se puede obtener un rango de hasta 10 centímetros y para detección de contraste es efectivo a distancias por debajo de los 4 centímetros, dentro de las ventajas que este sensor brinda, se encuentran la modulación de la señal emitida, evitando que otro tipo de fuentes de iluminación constante como el sol u otras que se encuentran moduladas a otras frecuencias como luces incandescentes y fluorescentes que pueden afectar el desempeño del sensor.

La desventaja que posee es el efecto de histéresis por la cual bajo ciertas condiciones de reflexión no es posible detectar el cambio de contraste, adicionalmente la detección no se realiza puntualmente como si se hace con la cámara de video, si no que detecta la emisión infrarroja recibida en un área, esto es debido a que el sistema no hace uso de lentes que concentren los rayos en un solo punto.

5.3.3 Sensores RFID: Estos sensores son usados para detectar los tags RFID localizados a lo largo de la bodega, de esta manera permite referenciar su posición dentro de la bodega y respecto a los demás agentes que intervienen en el sistema, para esto se usó un lector ID-20 de la empresa ID-INNOVATIONS que permite un rango de detección de hasta 2 pulgadas entre el tag y el lector RFID, así mismo el lector cuenta con sistema anticrash, que le permite ignorar la señal de 2 tags que se encuentren en las proximidades del área de detección, evitando lecturas erróneas.

5.3.4 Sensores ultrasónicos: permiten la detección de obstáculos en un rango de hasta 10 metros dependiendo de la referencia de sensor usada, en la mayoría de casos los sistemas de ultrasonido emiten una ráfaga de pulsos a una frecuencia ultrasónica, y cuando la señal retorna al sensor después de una reflexión con un objeto emite un pulso indicando que la señal ha

retornado, teniendo en cuenta el tiempo que transcurre desde el inicio de la ráfaga hasta la llegada del eco, se determina la distancia a la cual se encuentra el objeto. En otras referencias de sensores más modernos este sistema se ha cambiado por protocolos digitales como I2C o SPI permitiendo colocar múltiples sensores sobre una misma línea de datos ahorrando pines y conexiones utilizadas para su lectura; la tarjeta de control está habilitada para ambos tipos de sensores.

El mayor inconveniente de estos sensores radica en el área de incertidumbre generada por la forma de la onda emitida, la cual corresponde a una onda semi-esférica, por lo que es posible determinar su distancia (radio desde el sensor) pero no es posible detectar el ángulo vertical ni horizontal respecto al centro del sensor, además algunos objetos tienen la propiedad de dispersar y/o absorber las ondas emitidas por el sensor, por lo que la medida será errónea o simple mente el sensor indica la no presencia de objetos.

5.3.5 Sensores infrarrojos de medición de distancia. Otro tipo de sensor compatible con el sistema es la familia de sensores ópticos de Sharp Optoelectronics, de la línea GP2DXX, estos sensores permiten medir distancias de hasta 2 metros por medio de señales infrarrojas emitidas por el dispositivo y por medio de lentes y triangulación es posible determinar de manera muy precisa la distancia entre el sensor y un objeto.

Capítulo VI

6. RESULTADOS EXPERIMENTALES

6.1 Seguimiento de trayectoria basado en la cámara de video: las pruebas realizadas a la cámara demostraron que el nivel de contraste entregado por esta depende directamente de la frecuencia utilizada para su muestreo tanto pixel a pixel como cuadro a cuadro, esto de cierta manera limita las tasas de muestreo disponibles al momento de implementar un sistema de control clásico como lo son los derivados del PID digital, dentro de las pruebas realizadas a la etapa de video se realizó un control de la luz de backlight encendiéndola solo durante los periodos de adquisición de señal mostrando que la señal de salida aumenta el contraste significativamente, pudiendo de esta manera aumentar la tasa de muestreo para adquirir la señal y dando mayor rango de procesamiento de otro tipo de datos en el código implementado en el co-procesador.

La segunda prueba realizada a la cámara consistió en aplicar la luz de backlighth de forma constante, de esta manera se aumentó el contraste al punto de generar saturación en la imagen obtenida, y por consecuencia aumentando la frecuencia de muestreo se logra obtener una imagen aceptable.

Una vez caracterizada la cámara se procedió a realizar la corrección de la imagen ya que el lente inserta lo que en óptica se denomina aberración esférica, el primer método probado en el sistema fue el de regresión lineal con aproximación cosenoidal, pero el tiempo requerido para hacer el ajuste requerido superaba los 200 ms, por lo que el control del sistema estaba limitado a un muestreo superior a este.

El segundo método consistió en usar una iluminación no homogénea con dos Leds de color blanco con una intensidad lumínica de 3.2 lúmenes cada uno, de esta manera se logró compensar la no linealidad de la imagen, sin embargo la compensación no es 100% confiable debido a que iluminación externa puede afectarla, sin embargo es lo suficiente para hacer un procesamiento aceptable.

6.2 Carga útil: En este aspecto la plataforma mecánica fue diseñada para una carga útil de 50 Kgr, al realizar las pruebas se tuvo en cuenta el consumo de corriente en los 4 motores, los cuales mostraron un consumo sin carga útil de 1700 mA, luego con una carga de 50 Kgr mostró un consumo de 2100 mA en los cuatro motores, por último se incrementó la carga or encima del diseñado a fin de comprobar el margen de falla del sistema (Prueba destructiva), para dicho experimento se usó una cargad de 85 Kgr, mostrando un consumo total de 2500 mA, adicionalmente los motores no mostraron un esfuerzo excesivo puesto que la corriente nominal de cada motor es 1 A, por lo que en la medida final debería mostrar un consumo cercano a los 4 amperios, demostrando que está muy por encima del diseño predefinido.

6.3 Alcance Módulos Xbee: Se realizaron tres pruebas a saber:

- Sin antena (solo conector RPSMA): el alcance no fue superior a las 30 centímetros.
- Uno solo con antena: el alcance se extendió a 23 metros en línea de vista, con obstáculos (paredes) no superó los 10 metros.
- Con antena: el alcance es mayor a los 100 metros, no se realizó pruebas para determinar el alcance máximo puesto que la bodega de pruebas está restringida a 25 metros cuadrados por lo que una distancia máxima se fijó en 50 metros.

Las pruebas de alcance se realizaron en configuración punto a punto puesto que es el peor de los casos.

6.4 configuración de la red MESH: Se realizó con los parámetros de fábrica, recibiendo el 100% de los paquetes por debajo de una distancia de 50 metros en línea de vista, tomando un xbee como emisor y los otros como nodos End Device o nodos terminales. Se repitió la prueba anterior direccionando datos a nodos específicos bajo las mismas condiciones de distancia y configuración de paquete de datos y codificación recibiendo nuevamente el 100% de los datos.

Capitulo VII

7. CONCLUSIONES Y RECOMENDACIONES

- El uso de un puente H discreto aumenta las probabilidades de fallos debido a la suciedad presente en el entorno donde se desenvuelve el robot, por lo tanto se recomienda usar un driver que sea integrado y que tenga la capacidad de corriente suficiente para la aplicación puesto que esta fue una de las mayores limitantes.
- La implementación de la red mesh es sobredimensionada puesto que para la aplicación realizada y las restricciones de área dadas, no se pueden notar las ventajas de una red mesh respecto a topologías como la tipo estrella, árbol o punto a punto. Sin embargo la implementación es completamente válida para áreas más grandes en donde las ventajas de una red mesh son notorias.
- El seguimiento de una trayectoria por medio de sensores ópticos no es una alternativa confiable, puesto que las variaciones de luz que uedan existir en el entorno pueden afectar la respuesta del control del seguimiento de trayectoria, por eso se recomienda explorar nuevas alternativas como seguimientos de líneas de transmisión de datos dentro de la estructura de la bodega o línea de alimentación del sitio donde se ensamble el sistema, asi mismo métodos como balizamiento y localización global son algunas de las alternativas por explorar.
- Debido al software implementado en el sistema PRIMER es necesario reprogramar la aplicación para ingresar ítems nuevos a la bodega e incrementar su capacidad, se sugiere el uso de una base de datos local o una externa que permita almacenar nuevos ítems en cualquier momento y sin ningún tipo de restricción.

- El sistema PRIMER es un sistema en constante evolución por lo que a la fecha ya se encuentra disponible la nueva versión del PRIMER por lo que se recomienda usar no solo las últimas versiones si no otros sistemas embebidos que tienen niveles de flexibilidad y aplicabilidad similares.
- Como se dijo anteriormente el sistema primer es un sistema que evoluciona día a día y como es de esperar presenta algunos errores, entre ellos se pueden resaltar la falta de protecciones ESD en los reguladores de voltaje del sistema por lo que se queman con facilidad, así mismo el circuito impreso presenta errores de layout por lo que la etapa del oscilador local se ve afectada por capacitancias parásitas haciendo que los tiempos de transmisión asíncrona varíen erráticamente.
- El sistema de tracción diferencial presenta problemas de giro puesto que depende de la ubicación del centro de masa, haciendo que no gire sobre el centro geométrico de la plataforma si no sobre uno diferente dependiendo de la concentración de masa de la carga sobre el plano XY.
- Si bien las ruedas tipo mecanum funcionan de la manera esperada, el sistema no presenta movimiento lateral uniforme debido a la no uniformidad de los rodillos, puesto que su fabricación fue completamente manual y no son idénticos, se recomienda usar una matriz de fundición o de inyección de caucho u otro material sintético con coeficiente de fricción alto de manera que se mejore la uniformidad de la rodadura sobre la marcha.
- El sistema de elevación es muy complejo y requiere de muchos mecanismos para su funcionamiento, así mismo su eficiencia mecánica no es la deseada, por lo que se recomienda usar otras alternativas que reemplacen este dispositivo.
- El diseño de las ruedas mecanum en su bloque de soporte resultó muy robusto y sobredimensionado, por lo que el peso adicional es considerable, es necesario probar otro tipo de estructura de soporte como lamina de

CAPITULO 7. RESULTADOS Y CONCLUSIONES

acero cold rolled u otra estructura similar a fin de reducir el costo de fabricación y el peso de cada rueda.

- Al ser una plataforma muy completa, permite el uso de múltiples aplicaciones cooperativas así como individuales, por lo que se sugiere integrar este sistema como una herramienta de apoyo pedagógico en el área de robótica y automatización.

BIBLIOGRAFIA

- [1] ALAMI R (1998). Multirobot cooperation in the Martha Project, IEEE Robotics and Automation Magazine, vol 5, No. 1, p 36 -47.
- [2] ARKIN R.C (1990), integrating behavioural, perceptual and world knowledge in reactive navigation, Journal of Robotics and Autonomous Systems. Vol 6.
- [3] ASAMA H: (1994). Operation of cooperative multiple robots using communication in a decentralised robotic system, Proc. Conf. on From Perception to Action.
- [4] BOREINSTEIN J. (1996). Navigating Mobile Robots Systems and Techniques, A. K Peters, Wellesley, MA, USA, 225p.
- [5] KHATIB, O. (1987). A unified approach for motion and force control af robot manipulators: The operational space formulation. IEEE Journal of Robotics and Automation 3, 43-53.
- [6] LI, Z; P. Hsu and Sastry (1989). Grasping and coordinated manipulation by a multifingered robot hand. International Journal of Robotics Research 8 (4), 33 – 50.
- [7] COLE, A. (1990). Constrained motion of grasped objects by hybrid control. In: Proceedings of the 1990 IEEE International Conference on Robotics and Automation. Pp. 1954 – 1960.
- [8] PARRA, V. y S Arimoto (1996). Passivity – based adaptive sliding mode position forcé control for robot manipulators. International Journal of Adaptive Control and Signal Processing 10, 365 – 377.
- [9] LIU, Y, H; S Arimoto, V. Parra-Vega y K. Kitagaki (1997). Decentralized adaptive control of multiple manipulators in cooperations. International Journal of Control. P 649 – 673.

BIBLIOGRAFÍA

- [10] PARRA VEGA, V; A. Rodriguez Angeles, S. Arimoto y G. Hirzinger (2001). High precision constrained grasping with cooperative adaptive control. Journal of Intelligent and Robotic Systems 32, 235 – 254.
- [11] SCHLEGL, T; M. Buss, T. Omata y G. Schmidt (2001). Fast dextrous grasping with optimal contact forces and contact sensor based impedance control. In: Proc. IEEE International Conference on Robotics and Automation. Pp. 103 – 109.
- [12] MASON, M. T. y J. K. Salisbury (1985). Robot Hands and the Mechanics of Manipulation. The MIT Press. London, Great Britain.
- [13] PRIMER family, Embedded Evaluation and Development Tool Package. Disponible a Feb18-2010 en:
http://www.st.com/mcu/contentid-105-110-STM3210B_PRIMER.html
- [14] ARM Cortex-M3 Core Family. Disponible a Feb18-2010 en:
<http://www.arm.com/products/processors/cortex-m/cortex-m3.php>
- [15] WEICKER, R.P. Dhrystone: A Synthetic Systems Programming Benchmark, Siemens Corporate Research and Technology, Inc., Princeton, NJ; communications of the ACM Volume 27 , Issue 10 (Octubre 1984) pp. 1013 - 1030 año de publicacion: 1984
- [16] WEISS A.R. Dhrystone Benchmark: History, Analysis, Scores and Recommendations. Disponible a Feb18-2010 en:
<http://www.eembc.org/techlit/Datasheets/dhrystonewp.pdf>.
- [17] IEEE Standard for Inertial Sensor Terminology; IEEE Aerospace and Electronic Systems Society; Publicado por: El Instituto de Ingenieros Eléctricos y Electrónicos IEEE, Nueva York, Noviembre 29, 2001
- [18] RIDE 7, Raisonance Integrated Development Environment For Seamless Control of Application Development From Compilation to Debugging. Disponible a Feb18-2010 en: http://www.mcu_raisonance.com

BIBLIOGRAFÍA

- [19] ECLIPSE, open source community. Disponible a Feb18-2010 en: <http://www.eclipse.org/org/>
- [20] GUIZZO E. Three Engineers, Hundreds of Robots, One Warehouse, July 2008. Disponible a Feb18-2010 en: <http://spectrum.ieee.org/robotics/robotics-software/three-engineers-hundreds-of-robots-one-warehouse>.
- [21] DEFYING the laws of fulfillment: the kiva mobile fulfillment system, Kiva Systems. Disponible a Feb18-2010 en: <http://www.kivasystems.com/brochure/KivaLaws-Webprint.pdf>
- [22] FG Scalability, Flexibility, Portability: Zappos Re-Writes the Rules for Warehouse Design; Aberdeen Group, may 2008. Disponible a Feb18-2010 en: <http://www.kivasystems.com/briefings/index.php>
- [23] WAREHOUSE Automation: How to Implement Tomorrow's Order Fulfillment System Today, B. Wyland, October 2008. Disponible a Feb18-2010 en: <http://www.kivasystems.com/briefings2/index.php>.
- [24] AKYILDIZ I.F, X. Wang and W. Wang. Wireless mesh networks: a survey. In Computer Networks Volume 47, Issue 4, 15 Marzo 2005, pp. 445-487.
- [25] AKYILDIZ I.F, J. McNair, J.S.M. Ho, H. Uzunalioglu, W. Wang, Mobility management in next generation wireless systems, IEEE Proceedings 87 (1999) (8), pp. 1347–1385.
- [26] AKYILDIZ I.F, J. Xie, S. Mohanty, A survey of mobility management in next-generation all-IP-based wireless systems, IEEE Wireless Communications 11 (2004) (4), pp. 16–28.
- [27] LEE M.J., J. Zheng, Y.B. Ko, D.M. Shrestha, Ajou University, Korea. Emerging Standards For Wireless Mesh Technology. In IEEE Wireless Communications, April 2006.

BIBLIOGRAFÍA

- [28] ACHARYA A, A. Misra, S. Bansal, High-performance architectures for IP-based multihop 802.11 networks, *IEEE Wireless Communications* 10 (2003) (5), pp. 22–28.
- [29] CAO Y.U, A.S. Fukunaga, A.B. Kahng. Cooperative Mobile Robotics: Antecedents and Directions, in *Autonomous Robots*, 4, pp. 1–23 (1997).
- [30] PREMVUTI S, S. Yuta. Consideration on the cooperation of multiple autonomous mobile robots. In *IEEE/RSJ IROS*, pp. 59–63, 1990.
- [31] MCFARLAND D. Towards robot cooperation. In *Proc. Simulation of Adaptive Behavior*, 1994.
- [32] LEPAPE C. A combination of centralized and distributed methods for multi-agent planning and scheduling. In *IEEE ICRA*, pp. 488–493, 1990.
- [33] JIN K, P. Liang, and G. Beni. Stability of synchronized distributed control of discrete worm structures. In *IEEE ICRA*, pp. 1033–1038, 1994.
- [34] HACKWOOD S. and J. Wang. The engineering of cellular robotic systems. In *IEEE International Symposium on Intelligent Control*, pp. 70–75, 1988.
- [35] BENI G. The concept of cellular robotic system. In *IEEE International Symposium on Intelligent Control*, pp. 57–62, 1988.
- [36] GENOVESE V, P. Dario, R. Magni, L. Odetti. Self organizing behavior and swarm intelligence in a pack of mobile miniature robots in search of pollutants. In *IEEE/RSJ IROS*, pp 1575–1582, 1992.
- [37] DUDEK G, M. Jenkin, E. Miliotis, and D. Wilkes. Taxonomy for swarm robots. In *IEEE/RSJ IROS*, pp. 441–447, 1993.

BIBLIOGRAFÍA

[38] CALOUD P., W. Choi, J.-C.Latombe, C. Le Pape, M. Yin. Indoor automation with many mobile robots. In IEEE/RSJ IROS, pp. 67–72, 1990.

[39] MITSUMOTO N, T. Fukuda, K. Shimojina, and A. Ogawa. Micro autonomous robotic system and biologically inspired immune swarm strategy as a multi agent robotic system. In IEEE ICRA, pp. 2187–2192, 1995.

[40] ST Electronics, A company of ST Engineering. Disponible a Abril 6 de 2011. Disponible en: <http://www.st.com/>

[41] RAISONANCE. Corporate Home Page. Disponible a Abril 6 de 2011. Disponible en: <http://www.raisonance.com/>

[42] DIGI International. Making Wireless M2M Easy. Disponible a Abril 6 de 2011. Disponible en: <http://www.digi.com/>

[43] AIM, Association for Automatic Identification and Mobility. RFID. Disponible a Abril 6 de 2011. Disponible en: http://www.aimglobal.org/technologies/rfid/what_is_rfid.asp

[44] FINKENZELLER. K. RFID Handbook: Fundamentals and aplicaciones in contactless smart cards and identification. pp. 167 - 181. 2003

[45] ID Innovations. Better by Design. Disponible a Abril 6 de 2011. Disponible en: [http://www.id-innovations.com/httpdocs/Modules\(non%20write\).htm](http://www.id-innovations.com/httpdocs/Modules(non%20write).htm)

[46] HID Technology Basics White Paper. Disponible a Abril 6 de 2011. Disponible en: http://www.hidglobal.com/documents/understandCardDataFormats_wp_en.pdf

[47] DONGSHUN, DcGear Motor Series. Disponible a Abril 6 de 2011. Disponible en: <http://www.dongshunmotor.com/zl/clxjs/DS-119SS775.pdf>

[48] DIEGUEL Olaf, Improved mecanum Wheel design for omni-directional robots. Disponible a Abril 6 de 2011. Disponible en: <http://ftp.mi.fu-berlin.de/Rojas/omniwheel/Diegel-Badve-Bright-Potgieter-Tlale.pdf>

BIBLIOGRAFÍA

- [49] ANDY Mark. Disponible a Abril 6 de 2011.
Disponible en: <http://d1pytrrjwm20z9.cloudfront.net/MecanumWheelSpecSheet.pdf>
- [50] KRUHAK Igor. Holonomic Robot. Disponible a Abril 6 de 2011.
Disponible en: cs.nmu.edu/~SeniorProjects/ikruhak/Holonomic%20Robot.ppt
- [51] SKETCHUP, modelado en 3D. Disponible a Abril 7 de 2011. Disponible en:
<http://sketchup.google.com/intl/es/>
- [52] IMSA, Tecnología en Movimiento. Disponible a Abril 7 de 2011. Disponible en:
<http://www.imsa-colson.com/imsa/index.php/imsa/content/view/full/525>
- [53] OMNIX, Revolution in motion. Disponible a Abril 7 de 2011. Disponible en:
http://omnixtechnology.com/direct_components.html
- [54] ATMEL, Corporation, Disponible a Abril 7 de 2011. Disponible en:
<http://www2.atmel.com/>
- [55] ARDUINO. Disponible a Abril 7 de 2011. Disponible en: <http://arduino.cc>
- [56] FREESCALE, Semiconductors. Disponible a Abril 7 de 2011. Disponible en:
<http://www.freescale.com/>
- [57] FLEXIS Microcontrollers Series. MCF51AC256. Disponible a Abril 6 de 2011.
Disponible en:
http://cache.freescale.com/files/microcontrollers/doc/fact_sheet/MCF51AC256128FS.pdf?fsp=1
- [58] FREESCALE, Microcontrollers. Disponible a Abril 7 de 2011. Disponible en:
<http://www.freescale.com/files/microcontrollers/doc/brochure/BRCIPRODUCTS.pdf?fsrch=1>

REFERENCIA GRÁFICA

Figura 1 Fuente: Adaptada por autores. Extraída de: <http://todoelectronica.com/>

Figura 2 Fuente: Tomado de: Robot Store. Disponible en: <http://www.robotstorehk.com/soccer/RS-SOCCER-SYSTEM.html>

Figura 3 Fuente: Adaptada por los autores.

Figura 4 Fuente. Adaptado por autores. Tomado de: Defying the laws of fulfillment: the kiva mobile fulfillment system, Kiva Systems. /Feb18-2010: <http://www.kivasystems.com/brochure/KivaLaws-Webprint.pdf>

Figura 5 Fuente. Editado por los autores, tomado de: /Feb18-2010: http://www.st.com/mcu/contentid-105-110-STM3210B_PRIMER.html

Figura 6 Fuente. Editado por los autores, tomado de: Defying the laws of fulfillment: the kiva mobile fulfillment system, Kiva Systems. /Feb18-2010: <http://www.kivasystems.com/brochure/KivaLaws-Webprint.pdf>

Figura 7 Fuente. Tomado de: Blog de informático. / Feb18-2010 <http://blog-del-linformatico.blogspot.com>

Figura 8 Fuente. Tomado de: Zion Comunicaciones y redes srl. /Feb18-2010: <http://www.zion-srl.com.ar/productos.php>

Figura 9 Fuente. Autores.

Figura 10 Fuente. Tomado de: Stuttgart University ipvs. / Feb18-2010: http://www.ipvs.uni-stuttgart.de/abteilungen/bv/lehre/studentische_arbeiten/studienarbeiten/MasterTh_Collective_Robotics/de

Figura 11 Fuente. Tomado de: Open-source micro-robotic project. /Feb18-2010: <http://www.swarmrobot.org/tiki-index.php>

Figura 12 Fuente. Adaptada por autores. tomado de: Defying the laws of fulfillment: the kiva mobile fulfillment system, Kiva Systems. /Feb18-2010: <http://www.kivasystems.com/brochure/KivaLaws-Webprint.pdf>

Figura 13 Fuente. Adaptada por los autores.

Figura 14 Fuente. Manual de operación. Disponible en: http://www.parallax.com/Portals/0/Downloads/docs/prod/acc/TSL1401-DB_manual.pdf

Figura 15 Fuente. Adaptada por autores, extraído del documento: “Cuadro Nacional de Atribución del Espectro Radioeléctrico”. Disponible en: <http://archivo.mintic.gov.co/mincom/faces/index.jsp?id=1054>

Figura 16 Fuente. Catalogo del fabricante. Disponible en: [http://www.id-innovations.com/httpdocs/Modules\(non%20write\).htm](http://www.id-innovations.com/httpdocs/Modules(non%20write).htm)

BIBLIOGRAFÍA

Figura 17 Fuente. Adaptada por los autores.

Figura 18 Fuente. Autores.

Figura 19 Fuente. Autores.

Figura 20 Fuente. Adaptada por autores, extraído del catálogo de fabricantes.

Disponible en: www.omnixtechnology.com/direct_components.html

Figura 21 Fuente. Adaptada por autores. Disponible en sitio web del fabricante:

<http://ww1.microchip.com/downloads/en/DeviceDoc/DS-39904j.pdf>

Figura 22 Fuente. Catalogo del fabricante. Disponible en:

<http://arduino.cc/en/Main/Hardware>

Figura 23 Fuente. Adaptada por autores. Extraída de la guía de principiante para la tarjeta de desarrollo DEMOQE 128, disponible en:

http://www.pemicro.com/downloads/main_downloads_temp/201104110239402668108/DEMOQE128QSG.PDF

Figura 24 Fuente. Autores.

Figura 25 Fuente. Disponible en:

www.monigot.com/tutoriales/robotica/motores.php

Figura 26 Fuente. Hoja de datos del dispositivo.

Figura 27 Fuente. Hoja de datos del dispositivo.

Figura 28 Fuente. Hoja de datos del dispositivo.

Figura 29 Fuente. Adaptada por los autores. Hoja de datos del dispositivo.

Figura 30 Fuente. Hoja de datos del dispositivo.

Figura 31 Fuente. Autores.

Figura 32 Fuente. Autores.

Figura 33 Fuente. Autores.

Figura 34 Fuente. Autores.

Figura 35 Fuente. Autores.

Figura 36 Fuente. Autores.

Figura 37 Fuente. Autores.

Figura 38 Fuente. Autores.

Figura 39 Fuente. Autores.

Figura 40 Fuente. Autores.

Figura 41 Fuente. Autores.

Figura 42 Fuente. Autores.

Figura 43 Fuente. Autores.

Figura 44 Fuente. Autores.

Figura 45 Fuente. Autores.

BIBLIOGRAFÍA

Figura 46 Fuente. Autores.
Figura 47 Fuente. Autores.
Figura 48 Fuente. Autores.
Figura 49 Fuente. Autores.
Figura 50 Fuente. Autores.
Figura 51 Fuente. Autores.
Figura 52 Fuente. Autores.
Figura 53 Fuente. Autores.
Figura 54 Fuente. Autores.

ANEXO A

CONSTRUCCION Y FABRICACIÓN RUEDAS MECANUM

Proceso de fabricación: se inició con un modelo preliminar en Solid Works en el cual se fijaron como parámetros básicos el diámetro externo, el ancho de la rueda y los ángulos de los rodillos; con este modelo preliminar se procedió a realizar ajustes como distancias mínimas entre rodillos, número mínimo y máximo de rodillos por rueda, diámetro de los rodillos y perfil de ataque de las tapas laterales, de esta manera se obtuvo un primer acercamiento al modelo real.

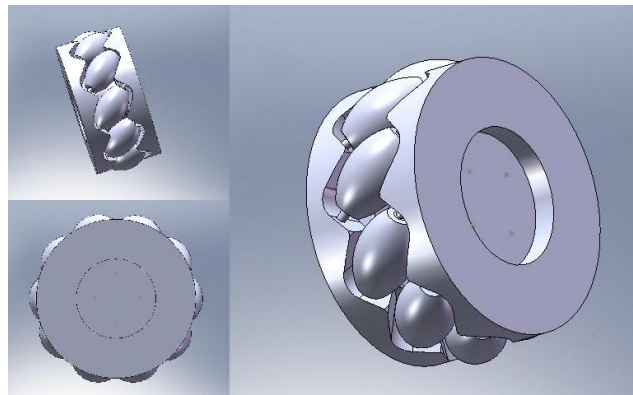


Figura 32. Vistas del Modelo de Rueda MECANUM

Una vez definidos los parámetros preliminares se procedió a realizar un modelo en madera en el sistema de mecanizado automático CNC del laboratorio FMS obteniendo como resultado un molde (positivo) para realizar la fundición en aluminio que dará soporte final a la rueda, este método permite ahorrar material que se pierde en el momento de hacer el maquinado a partir de un bloque sólido de material.



Figura 33. Vistas de los moldes obtenidos luego del proceso de mecanizado.

Dentro de los parámetros se vio la necesidad de realizar dos moldes con ángulos de ataque diferentes es decir uno con ángulo positivo y otro con ángulo negativo, los ángulos positivos corresponden a las ruedas delantera derecha y trasera izquierda, las otras dos ruedas deben tener perfil negativo.

Realizada la fundición del aluminio se obtienen las piezas de aluminio sin rectificar, dicha rectificación es obligatoria debido a las imperfecciones de los gránulos de arena usados durante la fabricación del molde (negativo) para su posterior fundición, así como las imperfecciones producto de contracciones y dilataciones del material durante el proceso de enfriamiento.

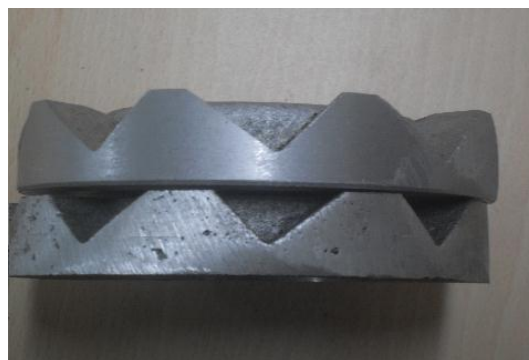


Figura 34. Piezas en aluminio antes y después del enfrentado.

ANEXO A: CONSTRUCCIÓN Y FABRICACIÓN RUEDAS MECANUM

Por medio del torno automatizado se hace un refrentado de las tapas de las ruedas, para poderlas fijar adecuadamente en el centro de mecanizado (CNC).

Antes de rectificar las caras de los conos de ataque se procede a hacer los ajustes de los rodillos de manera que la rodadura sea uniforme y sin sobresaltos que puedan producir vibraciones, este ajuste se realiza proyectando un radio de rodadura imaginario sobre los cilindros de caucho a fin de determinar el perfil de cada uno de ellos, finalizando de esta manera el diseño de las llantas propuestas.

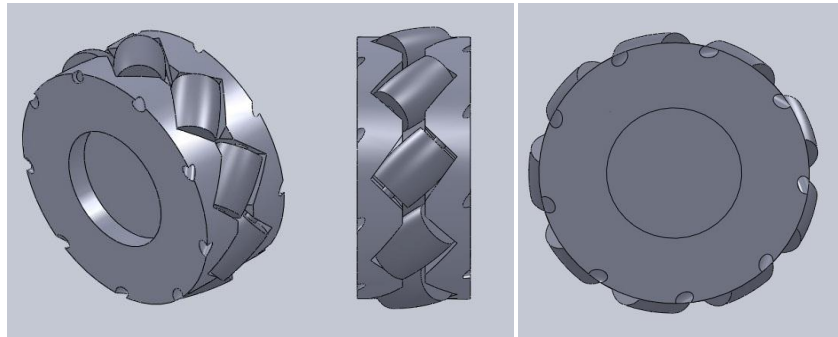


Figura 35. Diseño final en Solid Works, rodillos definitivos incluidos.

ANEXO B:

ESQUEMATICOS Y NEGATIVOS DE LAS TARJETAS

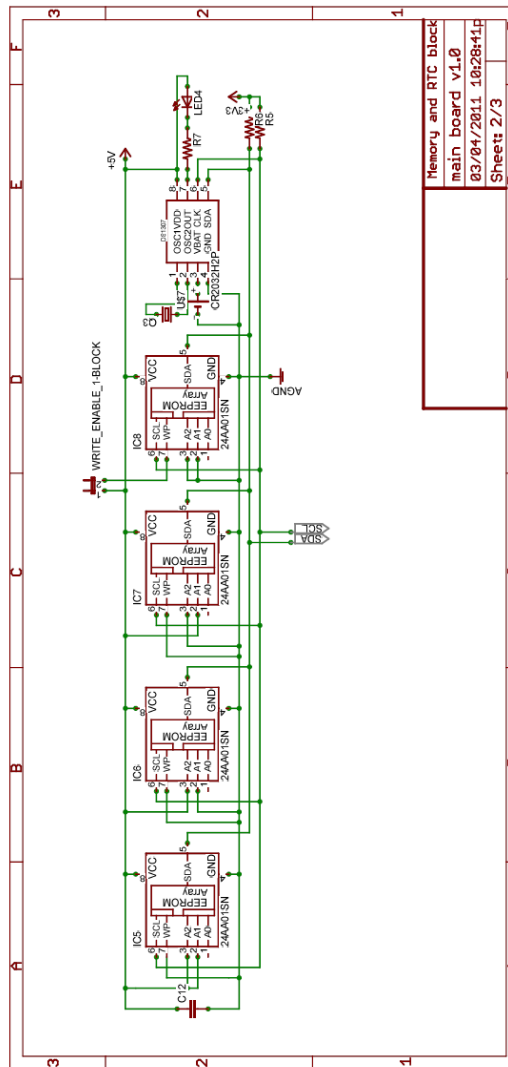
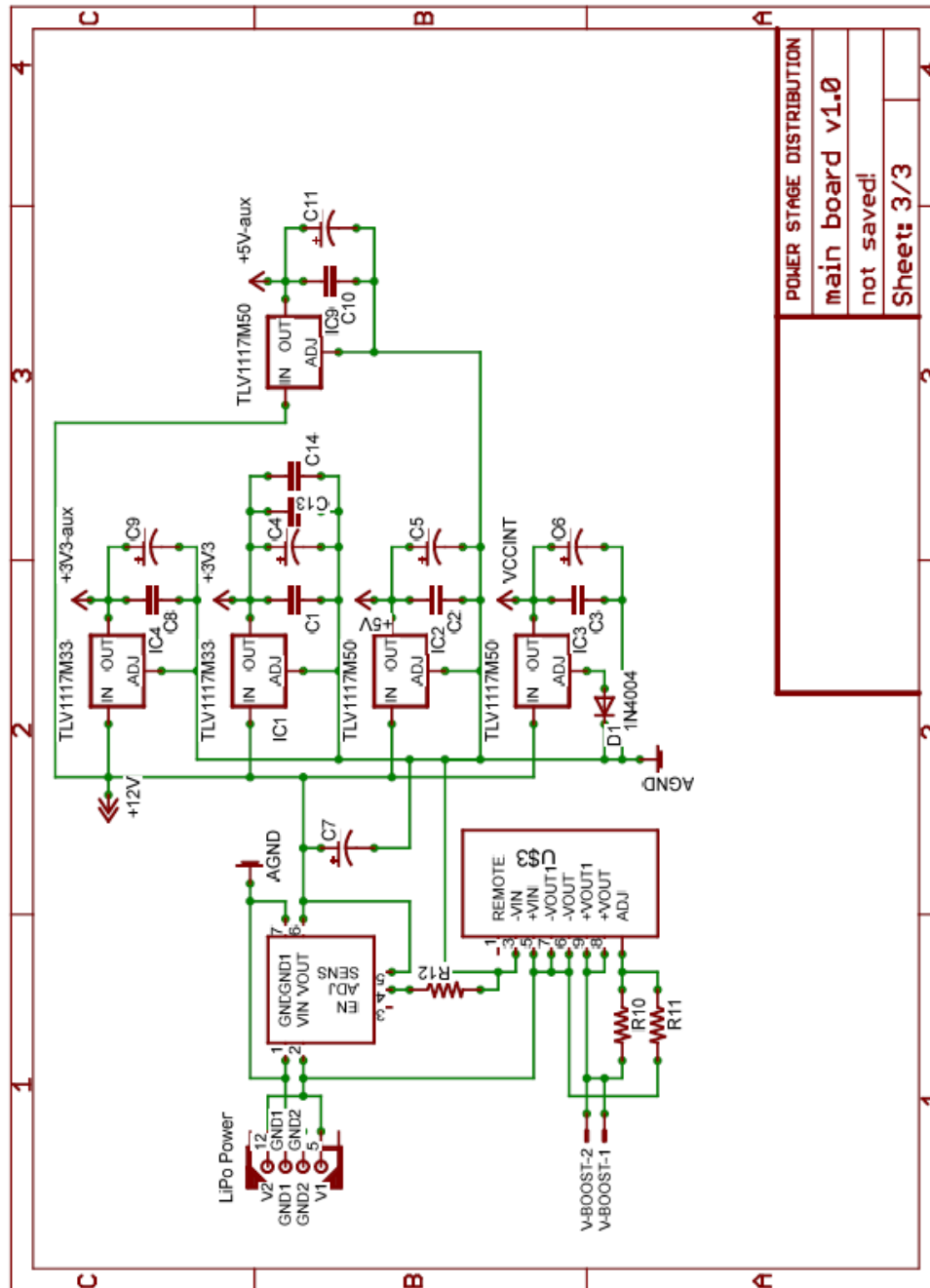


Figura 36: Bloque de memorias y RTC



POWER STAGE DISTRIBUTION	
main board v1.0	
not saved!	
Sheet: 3/3	

Figura 37: Etapa de Alimentación del sistema:

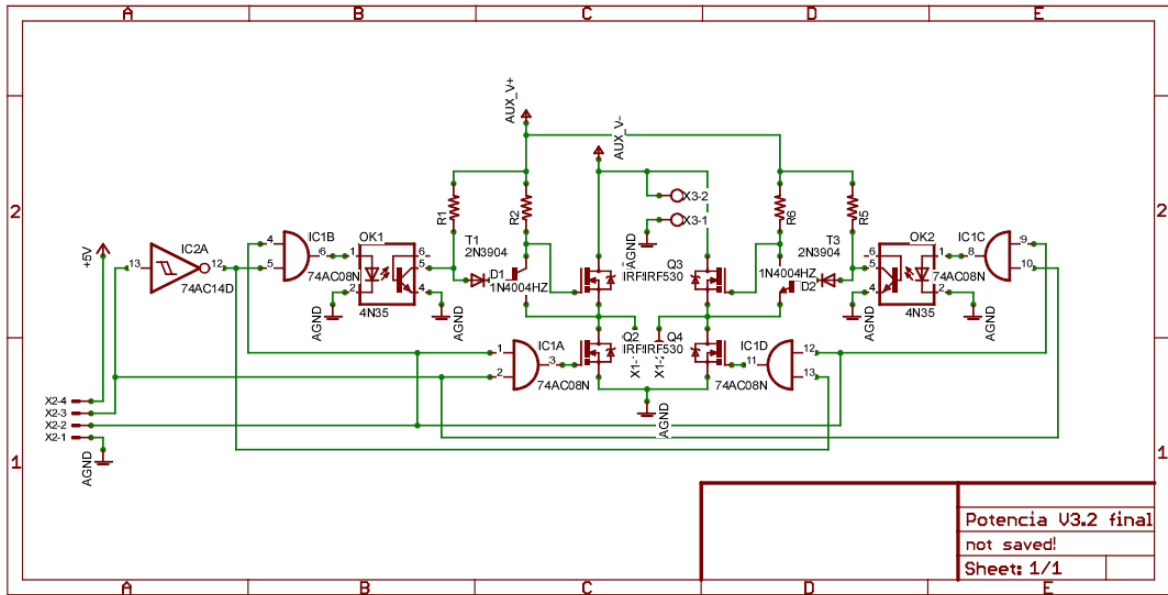


Figura 39: Celda puente H (4 celdas por tarjeta)

Nota: todas las imágenes se encuentran escaladas

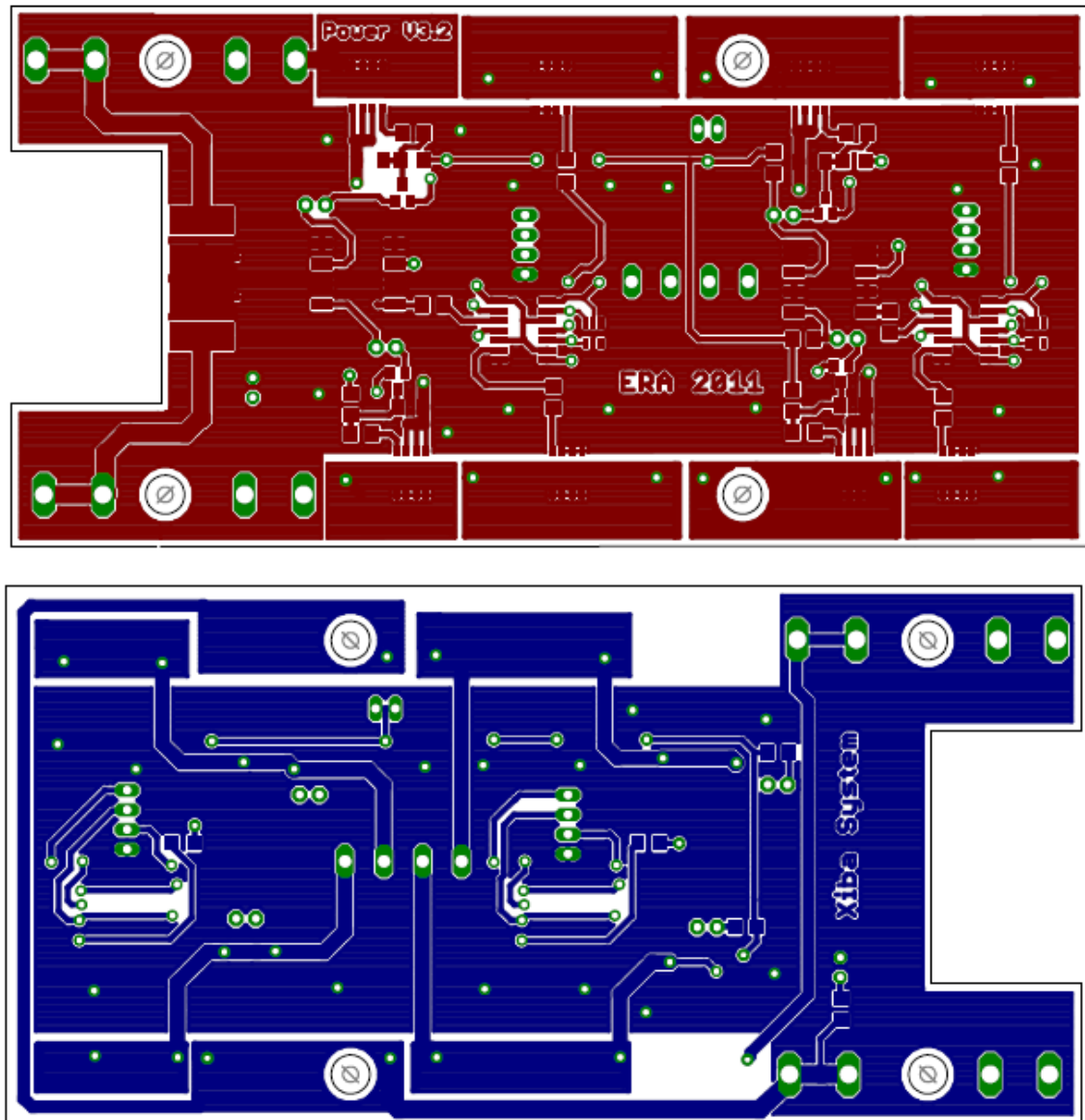


Figura 40: Layout tarjeta de potencia (top-rojo/bottom-azul)

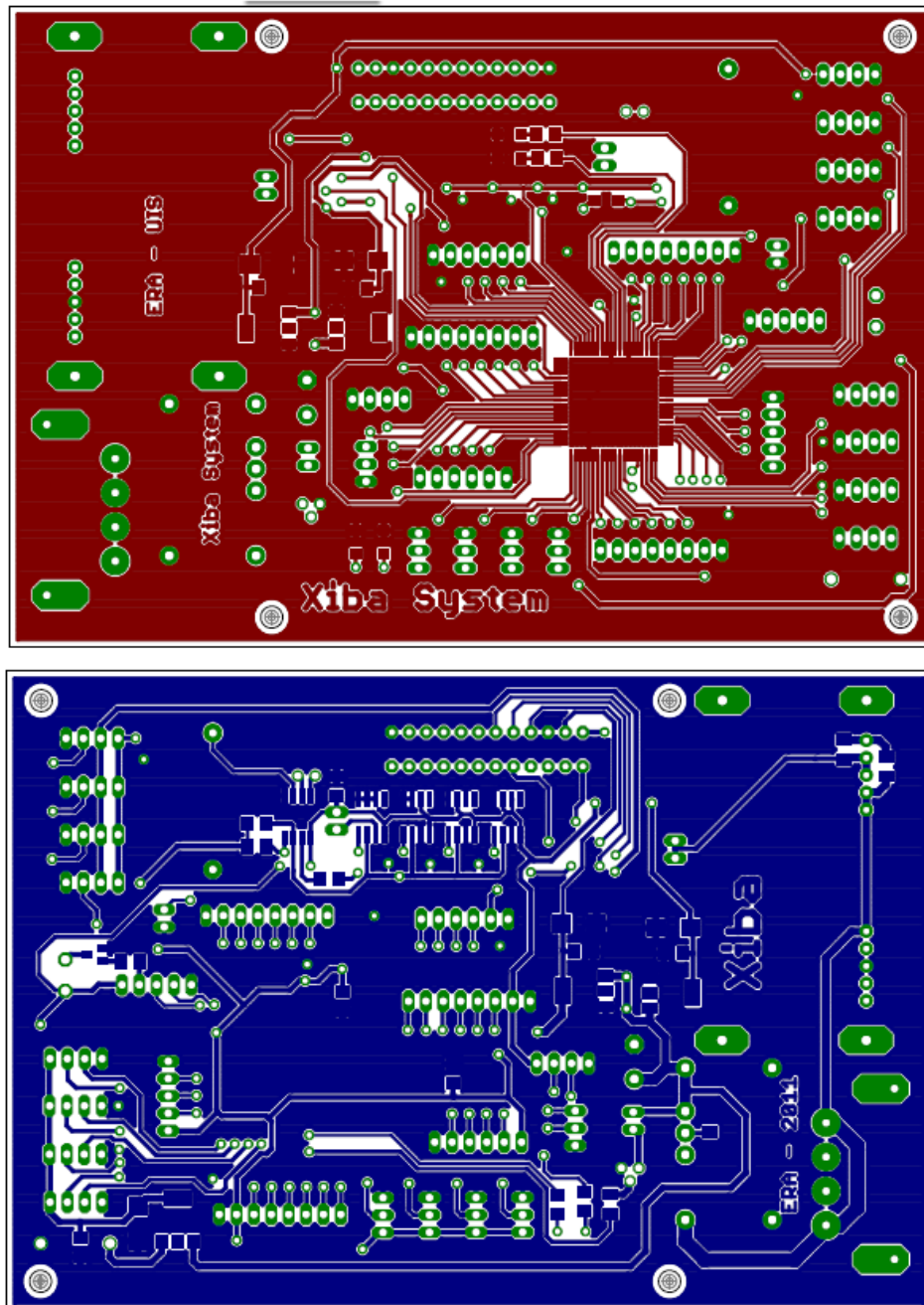


Figura 41: Layout tarjeta principal, bloque de polarización y banco de memorias (top-rojo/botom-azul)

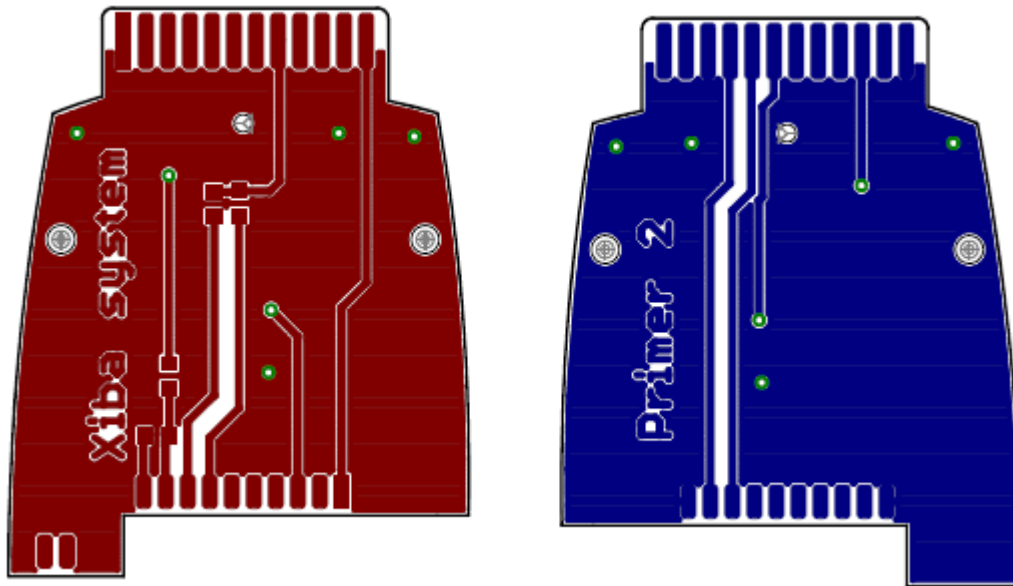


Figura 42: Layout tarjetas de expansión PRIMER 2 (Agentes)

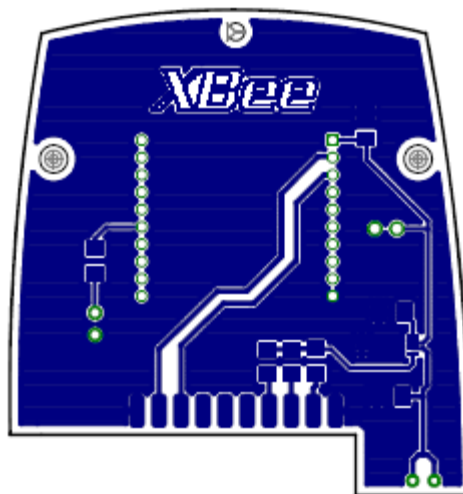


Figura 43: Layout tarjeta de expansión PRIMER 2 (Xbee, mando de control)

ANEXO C

CODIGO FUENTE PRIMER_2

ALGORITMOS DEL PRIMER 2

Aplicación para el pedido de itens de inventario:

/****** (C) COPYRIGHT 2007-2009
RAISONANCE *****

* File Name : Application.c
* Author :
* Date First Issued :
* Description : Circle_App CircleOS
application template.
* Revision :

*****/

/* Includes -----
-----*/

#include "circle_api.h"
#include "stm32f10x_usart.h"
#include "stm32f10x_rcc.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_nvic.h"

/* Private defines -----
-----*/

// The following should be the minimal CircleOS
version needed by your application

#define NEEDEDVERSION "V 3.7"

/* Private functions -----
-----*/

static enum MENU_code MsgVersion(void);
void inicio(void);
void teclado(void);

```
void ClearScreen(void);

void move_joystick(void);

void imprime(void);

void enviar();

/* Public variables -----
-----*/

const char Application_Name[8+1] =
{"XIVA_com"}; // Max 8 characters

u8 RxData,TxData,i;

int currentMsgIdx,j;

u8 BallPointerBmp[POINTER_WIDTH] =
{0x38,0x7C,0xFF,0xFF,0xFF,0x7C,0x38};

u8 s,t,k,h,m,l,n;

u8 bandera;

u16 let;

u16 Text[100];

USART_InitTypeDef USART_InitStructure;

/******
*****

* Function Name : Application_Ini

* Description : Initialization function of
Circle_App. This function will

* be called only once by CircleOS.

* Input : None

* Return : MENU_CONTINUE_COMMAND

*****
*****/

enum MENU_code Application_Ini(void)

{
```

ANEXO C: CODIGO FUENTE

```
// Ensure that the current OS version is recent
enough

if(strcmp(UTIL_GetVersion(),
NEEDEDVERSION) < 0)

{

return MsgVersion();

}

// TODO: Write your application initialization
function here.

//inicializacion de la interfaz grafica

LCD_FillRect ( 0,0,127,130, RGB_WHITE
);//clr scr

inicio();

DRAW_Line(0,114,120,114,RGB_RED);//BARRA
S DE TITULO

DRAW_Line(0,111,120,111,RGB_RED);

DRAW_DisplayString( 3, 115, "COMANDOS -
XIVA", 17);//17

DRAW_Line(0,76,120,76,RGB_RED);//BARRAS
DE CUADRO DE SELECCION

DRAW_Line(0,60,120,60,RGB_RED);

DRAW_Line(0,59,120,59,RGB_RED);

DRAW_DisplayString(5,95,"ESTADO:",17);

DRAW_DisplayString(5,80,"EN
ESPERA",17);

DRAW_Line(0,37,120,37,RGB_RED);

DRAW_Line(0,16,120,16,RGB_RED);

LED_Set (LED_RED,LED_ON);

s = 0;

i = 48;

k = 0;

h = 0;

l = 0;

n = 0;

bandera = 0;

// Init App

return MENU_CONTINUE_COMMAND;

}

/*****
*****

* Function Name : Application_Handler

* Description : Management of the Circle_App.

*

* Input : None

* Return : MENU_CONTINUE

*****/

enum MENU_code Application_Handler(void)

{

// TODO: Write your application handling here.

POINTER_SetColor(RGB_WHITE);

POINTER_Draw(t,8,5,5,0);

POINTER_SetColor(RGB_BLUE);

POINTER_Draw(s,8,5,5,0);

//mueve el joystick

move_joystick();

//selecciona la letra que quiere escribir

teclado();

// escribe la letra

imprime();

//enviar

enviar();

// This routine will get called repeatedly by
CircleOS, until we
```

ANEXO C: CODIGO FUENTE

```

// return MENU_LEAVE
#endif

// If the button is pressed, the application is
// exited
if(BUTTON_GetState() == BUTTON_PUSHED)
{
    BUTTON_WaitForRelease();
    /*USART_Cmd(USART2, DISABLE);

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_
    GPIOA, DISABLE);

    /* Enable USART2 clocks */

    /*RCC_APB1PeriphClockCmd(RCC_APB1Periph_
    USART2, DISABLE);

    NVIC_InitTypeDef NVIC_InitStructure;

    NVIC_InitStructure.NVIC_IRQChannelCmd =
    DISABLE;*/

    return MENU_Quit();
}

return MENU_CONTINUE; // Returning
MENU_LEAVE will quit to CircleOS
}

/*****
*****

* Function Name : MsgVersion

* Description : Display the current CircleOS
version and the version needed

* exit to main menu after 4 secondes

*

* Input : None

* Return : MENU_REFRESH

*****
*****/

static enum MENU_code MsgVersion(void)
{
    int hh,mm,ss,ss2;

    DRAW_DisplayString(5,60,"CircleOS",17);

    DRAW_DisplayString(80,60,UTIL_GetVersion(),6);

    DRAW_DisplayString(5,34,NEEDEDVERSION,6);

    DRAW_DisplayString(50,34," required",12);

    RTC_GetTime( &hh, &mm, &ss);

    ss = ss + 4; // 4 secondes

    ss = ss%60;

    do
    {
        RTC_GetTime( &hh, &mm, &ss2 );
    }

    while ( ss2 != ss ); // do while < 4
    secondes

    DRAW_Clear();

    return MENU_REFRESH;
}

/*****
*****

* Function Name : RCC_Configuration

* Description : Configures the different system
clocks.

* Input : None

* Output : None

* Return : None

*****
*****/

void RCC_Configuration(void)

```

ANEXO C: CODIGO FUENTE

```

{
    /* RCC system reset(for debug purpose) */
    RCC_DeInit();

    /* Enable GPIOA clocks */

RCC_APB2PeriphClockCmd(RCC_APB2Periph_
GPIOA, ENABLE);

    /* Enable USART2 clocks */

RCC_APB1PeriphClockCmd(RCC_APB1Periph_U
SART2, ENABLE);
}

/*****
*****

* Function Name : GPIO_Configuration

* Description   : Configures the different GPIO
ports.

* Input        : None

* Output       : None

* Return       : None

*****/

void GPIO_Configuration(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;

    //podria ser GPIO_Mode_IPD

    GPIO_InitStructure.GPIO_Mode      =
GPIO_Mode_IN_FLOATING;

    /* Configure USART2 Rx as input floating */
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_3;
    GPIO_Init(GPIOA, &GPIO_InitStructure);

    /* Configure USART2 Tx as alternate function
push-pull */

    GPIO_InitStructure.GPIO_Speed     =
GPIO_Speed_50MHz;
}

    GPIO_InitStructure.GPIO_Mode      =
GPIO_Mode_AF_PP;

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2;

    GPIO_Init(GPIOA, &GPIO_InitStructure);
}

/*****
*****

* Function Name : NVIC_Configuration

* Description   : Configures the nested vectored
interrupt controller.

* Input        : None

* Output       : None

* Return       : None

*****/

void NVIC_Configuration(void)
{
    NVIC_InitTypeDef NVIC_InitStructure;

    /* Configure the NVIC Preemption Priority Bits */

    NVIC_PriorityGroupConfig(NVIC_PriorityGroup_0)
;

    /* Enable the USART2 Interrupt */

    NVIC_InitStructure.NVIC_IRQChannel      =
USART2_IRQChannel;

    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;

    NVIC_InitStructure.NVIC_IRQChannelCmd   =
ENABLE;

    NVIC_Init(&NVIC_InitStructure);
}

/*****
*****

* Function Name : InitUSART

```

ANEXO C: CODIGO FUENTE

```

* Description   : Configures the nested vectored
interrupt controller.

* Input        : None

* Output       : None

* Return       : None

*****
*****/

void InitUSART(void)
{
/* USART1 and USART2 configuration -----
-----*/

/* USART and USART2 configured as follow:

- BaudRate = 9600 baud
- Word Length = 8 Bits
- One Stop Bit
- No parity
- Hardware flow control disabled (RTS and
CTS signals)
- Receive and transmit enabled*/

USART_InitStructure.USART_BaudRate = 9600;

USART_InitStructure.USART_WordLength =
USART_WordLength_8b;

USART_InitStructure.USART_StopBits =
USART_StopBits_1;

USART_InitStructure.USART_Parity =
USART_Parity_No;

USART_InitStructure.USART_HardwareFlowControl =
USART_HardwareFlowControl_None;

USART_InitStructure.USART_Mode =
USART_Mode_Rx | USART_Mode_Tx;

// Configure USART2

USART_Init(USART2, &USART_InitStructure);

USART_ClearITPendingBit(USART2,
USART_IT_RXNE);

USART_ClearITPendingBit(USART2,
USART_IT_TC);

USART_ClearITPendingBit(USART2,
USART_IT_TXE);

/* Enable USART2 Receive and Transmit
interrupts */

USART_ITConfig(USART2, USART_IT_RXNE,
ENABLE);

USART_ITConfig(USART2, USART_IT_TXE,
ENABLE);

USART_ITConfig(USART2, USART_IT_TC,
ENABLE);

USART_SetPrescaler(USART2, 0x56);

/* Enable the USART2 */

USART_Cmd(USART2, ENABLE);
}

/*****
*****/

* Function Name : inicio

* Description   : Configures the nested vectored
interrupt controller.

* Input        : None

* Output       : None

* Return       : None

*****
*****/

void inicio(void)
{
/* System Clocks Configuration */

RCC_Configuration();

/* NVIC configuration */

NVIC_Configuration();

/* Configure the GPIO ports */

GPIO_Configuration();

/* USART configuration */

```

ANEXO C: CODIGO FUENTE

```
InitUSART();
}
void ClearScreen()
{
    LCD_FillRect ( 0,0,127,130, RGB_WHITE );
}
//funciones de la usart
//Envio del vector de comandos a traves de
comunicacion serial
void enviar()
{
    if(JOYSTICK_GetState()==JOYSTICK_UP)
    {
        JOYSTICK_WaitForRelease();
        h = 0;
        t = 48;
        l = 0;
        if(bandera == 1)
        {
            n = 1;
            i = 48;
            while(Text[h]!=0x00){
                LED_Set (LED_RED,LED_OFF);
                LED_Set (LED_GREEN,LED_ON);
            }
            DRAW_DisplayString(5,78,"ENVIANDO... ",17);
            TxData = Text[h];
            USART_SendData(USART2,TxData);
        }
        while(USART_GetFlagStatus(USART2,
USART_FLAG_TXE)==RESET) {};
        //
        USART_ClearITPendingBit(USART2,
USART_IT_TC);
        DRAW_DisplayString(5,20,"Envio:",6);
        if(t>108)
        {
            t = 48;
        }
        DRAW_SetCursorPos(t, 20);
        if(l<k)
        {
            DRAW_Putc(Text[l]);
        }
        t = t + 6;
        Text[h] = 0x00; //limpiado del vector
        luego del envio
        bandera = 2;
        l++;
        h++;
        for(j=0;j<800;j++){ //retardo
            for(m=0;m<100;m++){}}
        DRAW_DisplayString(5,78,"ENVIANDO...ok",17);
    }
    k = 0;
    DRAW_DisplayString(0,40,"
",17);
}
//FUNCIONES DEL TECLADO
void teclado(){
    //listado del inventario
    if(s==0){
        DRAW_DisplayString(5,61,"1:TORNILLOS",17);//1
```

ANEXO C: CODIGO FUENTE

```
        let = '1';
    }
} else if(s==4){
}
/*-----*/
DRAW_DisplayString(5,61,"2:TUERCAS",17);//2
let = '2';
// Seleccion de items
} else if(s==8){
void move_joystick(){
    if(JOYSTICK_GetState()==JOYSTICK_RIGHT)
    {
        JOYSTICK_WaitForRelease();
        LED_Set (LED_RED,LED_ON);
        LED_Set (LED_GREEN,LED_OFF);
        bandera = 0;
    }
    DRAW_DisplayString(5,78,"Seleccionando",17);
    DRAW_DisplayString(0,20,"",17);
    //limpia linea de envio en pantalla
    s = s + 4;
    t = s - 4;
}
    if(JOYSTICK_GetState()==JOYSTICK_LEFT)
    {
        JOYSTICK_WaitForRelease();
        LED_Set (LED_RED,LED_ON);
        LED_Set (LED_GREEN,LED_OFF);
        bandera = 0;
    }
    DRAW_DisplayString(5,78,"Seleccionando",17);
    DRAW_DisplayString(0,20,"",17);
    //limpia linea de envio en pantalla
    s = s - 4;
    t = s + 4;
}
    if(s<0)
DRAW_DisplayString(5,61,"3:ARANDELAS",17);//3
let = '3';
} else if(s==12){
DRAW_DisplayString(5,61,"4:PUNTILLAS",17);//4
let = '4';
} else if(s==16){
DRAW_DisplayString(5,61,"5:TACHUELAS",17);//5
let = '5';
} else if(s==20){
DRAW_DisplayString(5,61,"6:DESTORNILLADORES",17);//6
let = '6';
} else if(s==24){
DRAW_DisplayString(5,61,"7:MARTILLOS",17);//7
let = '7';
} else if(s==28){
DRAW_DisplayString(5,61,"8:BROCAS",17);//8
let = '8';
} else {
    DRAW_DisplayString(5,61,"9:taladros",17);//9 -
    APARECE CUANDO HAY ERROR
    let = '9';
}
```

ANEXO C: CODIGO FUENTE

```

        {
            s = 0;
        }
        if(s>28)
        {
            s = 0;
        }
    }

//Almacena item en vector de pedido e imprime la
orden

void imprime(){
    if(JOYSTICK_GetState() == JOYSTICK_DOWN)
    {
        JOYSTICK_WaitForRelease();
        LED_Set (LED_RED,LED_ON);
        LED_Set (LED_GREEN,LED_OFF);
        bandera = 1;
        DRAW_DisplayString(5,78,"Guardando
",17);
        DRAW_DisplayString(0,20,"
",17); //limpia linea de envio en pantalla
        DRAW_DisplayString(5,40,"Lista:",6);
        if(i>108)
        {
            i = 48;
        }
        DRAW_SetCursorPos(i,40);
        Text[k]=let;
        n = k+1;
        Text[n]=0x2D;//identificador de final del
vector de comando ("-")
        DRAW_Putc(Text[k]);
    }
    i = i + 6;
    DRAW_Putc(',');
    i = i + 6;
    // k = k+2;
    k++;
}
}

Aplicación para el control remoto de los moviles:

/***** (C) COPYRIGHT 2007-2009
RAISONANCE *****/

* File Name      : Application.c

* Author        :

* Date First Issued :

* Description    : Circle_App CircleOS
application template.

* Revision      :

*****/

/* Includes -----
-----*/

#include "circle_api.h"
#include "stm32f10x_usart.h"
#include "stm32f10x_rcc.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_nvic.h"

/* Private defines -----
-----*/

// The following should be the minimal CircleOS
version needed by your application

#define NEEDEDVERSION "V 3.7"

/* Private functions -----
-----*/

static enum MENU_code MsgVersion(void);

void inicio(void);

```

ANEXO C: CODIGO FUENTE

```

void lista(void);
void ClearScreen(void);
void busqueda(void);
void seleccion(void);
void control(void);

/* Public variables -----
-----*/

const char Application_Name[8+1] = {"Remoto"};
// Max 8 characters

u8 RxData,TxData,i;

int currentMsgIdx,j;

u8 BallPointerBmp[POINTER_WIDTH] =
{0x38,0x7C,0xFF,0xFF,0xFF,0x7C,0x38};

u8 s,t,k,h,m,l,n;

u8 bandera;

u16 let;

u16 Text[100];

USART_InitTypeDef USART_InitStructure;

/*****
*****

* Function Name : Application_Ini

* Description : Initialization function of
Circle_App. This function will

* be called only once by CircleOS.

* Input : None

* Return : MENU_CONTINUE_COMMAND

*****/

enum MENU_code Application_Ini(void)
{
// Ensure that the current OS version is recent
enough

if(strcmp(UTIL_GetVersion(),
NEEDEDVERSION) < 0)

```

```

{
return MsgVersion();
}

// TODO: Write your application initialization
function here.

//inicializacion de la interfaz grafica

LCD_FillRect ( 0,0,127,130, RGB_WHITE
);//clear screen

inicio();

DRAW_Line(0,114,120,114,RGB_RED);//BARRA
S DE TITULO

DRAW_Line(0,111,120,111,RGB_RED);

DRAW_DisplayString( 3, 115, "Control
Remoto", 17);//17

DRAW_Line(0,76,120,76,RGB_RED);//BARRAS
DE CUADRO DE SELECCION

DRAW_Line(0,60,120,60,RGB_RED);

DRAW_Line(0,59,120,59,RGB_RED);

DRAW_DisplayString(5,95,"ESTADO:",17);

DRAW_DisplayString(5,80,"EN
ESPERA",17);

s = 0;

i = 48;

k = 0;

h = 0;

l = 0;

n = 0;

bandera = 0;

// Init App

return MENU_CONTINUE_COMMAND;

}

```


ANEXO C: CODIGO FUENTE

```

DRAW_DisplayString(80,60,UTIL_GetVersion(),6);
DRAW_DisplayString(5,34,NEEDEDVERSION,6);
DRAW_DisplayString(50,34," required",12);
RTC_GetTime( &hh, &mm, &ss);
ss = ss + 4;          // 4 secondes
ss = ss%60;
do
{
    RTC_GetTime( &hh, &mm, &ss2 );
}
while ( ss2 != ss );    // do while < 4
secondes
DRAW_Clear();
return MENU_REFRESH;
}

/*****
*****

* Function Name : RCC_Configuration
* Description   : Configures the different system
clocks.
* Input        : None
* Output       : None
* Return       : None

*****/
void RCC_Configuration(void)
{
    /* RCC system reset(for debug purpose) */
    RCC_DeInit();

    /* Enable GPIOA clocks */
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_
GPIOA, ENABLE);

    /* Enable USART2 clocks */

    RCC_APB1PeriphClockCmd(RCC_APB1Periph_U
SART2, ENABLE);
}

/*****
*****

* Function Name : GPIO_Configuration
* Description    : Configures the different GPIO
ports.
* Input         : None
* Output        : None
* Return        : None

*****/
void GPIO_Configuration(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;

    //podria ser GPIO_Mode_IPD

    GPIO_InitStructure.GPIO_Mode      =
GPIO_Mode_IN_FLOATING;

    /* Configure USART2 Rx as input floating */

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_3;

    GPIO_Init(GPIOA, &GPIO_InitStructure);

    /* Configure USART2 Tx as alternate function
push-pull */

    GPIO_InitStructure.GPIO_Speed      =
GPIO_Speed_50MHz;

    GPIO_InitStructure.GPIO_Mode      =
GPIO_Mode_AF_PP;

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2;

    GPIO_Init(GPIOA, &GPIO_InitStructure);
}

```

ANEXO C: CODIGO FUENTE

```

*****
*****/

/*****
*****

* Function Name : NVIC_Configuration

* Description : Configures the nested vectored
interrupt controller.

* Input : None

* Output : None

* Return : None

*****
*****/

void NVIC_Configuration(void)
{
    NVIC_InitTypeDef NVIC_InitStructure;

    /* Configure the NVIC Preemption Priority Bits */

    NVIC_PriorityGroupConfig(NVIC_PriorityGroup_0)
    ;

    /* Enable the USART2 Interrupt */

    NVIC_InitStructure.NVIC_IRQChannel =
    USART2_IRQChannel;

    NVIC_InitStructure.NVIC_IRQChannelSubPriority
    = 0;

    NVIC_InitStructure.NVIC_IRQChannelCmd =
    ENABLE;

    NVIC_Init(&NVIC_InitStructure);
}

/*****
*****

* Function Name : InitUSART

* Description : Configures the nested vectored
interrupt controller.

* Input : None

* Output : None

* Return : None

*****
*****/

void InitUSART(void)
{
    /* USART1 and USART2 configuration -----
    -----*/

    /* USART and USART2 configured as follow:

        - BaudRate = 9600 baud

        - Word Length = 8 Bits

        - One Stop Bit

        - No parity

        - Hardware flow control disabled (RTS and
        CTS signals)

        - Receive and transmit enabled*/

    USART_InitStructure.USART_BaudRate = 9600;

    USART_InitStructure.USART_WordLength =
    USART_WordLength_8b;

    USART_InitStructure.USART_StopBits =
    USART_StopBits_1;

    USART_InitStructure.USART_Parity =
    USART_Parity_No;

    USART_InitStructure.USART_HardwareFlowControl
    = USART_HardwareFlowControl_None;

    USART_InitStructure.USART_Mode =
    USART_Mode_Rx | USART_Mode_Tx;

    // Configure USART2

    USART_Init(USART2, &USART_InitStructure);

    USART_ClearITPendingBit(USART2,
    USART_IT_RXNE);

    USART_ClearITPendingBit(USART2,
    USART_IT_TC);

    USART_ClearITPendingBit(USART2,
    USART_IT_TXE);

    /* Enable USART2 Receive and Transmit
    interrupts */
}

```


ANEXO C: CODIGO FUENTE

```
        while(USART_GetFlagStatus(USART2,
USART_FLAG_TXE)==RESET) {};
```

```
    }
```

```
    for(j=0;j<800;j++){ //retardo
```

```
    for(m=0;m<100;m++){}}
```

```
    }
```

```
if(JOYSTICK_GetState()==JOYSTICK_RIGHT)
```

```
{
```

```
    // JOYSTICK_WaitForRelease();
```

```
    DRAW_DisplayString(5,20,"Derecha...
```

```
    ",17);
```

```
    LED_Set (LED_RED,LED_OFF);
```

```
    LED_Set (LED_GREEN,LED_ON);
```

```
    if(s == 0) {
```

```
        USART_SendData(USART2,0x64);//d
```

```
        while(USART_GetFlagStatus(USART2,
USART_FLAG_TXE)==RESET) {};
```

```
    } else if(s == 1) {
```

```
        USART_SendData(USART2,0x65);//e
```

```
        while(USART_GetFlagStatus(USART2,
USART_FLAG_TXE)==RESET) {};
```

```
    } else {
```

```
        USART_SendData(USART2,0x66);//f
```

```
        while(USART_GetFlagStatus(USART2,
USART_FLAG_TXE)==RESET) {};
```

```
    }
```

```
    for(j=0;j<800;j++){ //retardo
```

```
    for(m=0;m<100;m++){}}
```

```
    }
```

```
    if(JOYSTICK_GetState() ==
```

```
    JOYSTICK_DOWN)
```

```
    {
```

```
        // JOYSTICK_WaitForRelease();
```

```
        DRAW_DisplayString(5,20,"Atras...
```

```
        ",17);
```

```
        LED_Set (LED_RED,LED_OFF);
```

```
        LED_Set (LED_GREEN,LED_ON);
```

```
        if(s == 0) {
```

```
            USART_SendData(USART2,0x67);//g
```

```
            while(USART_GetFlagStatus(USART2,
USART_FLAG_TXE)==RESET) {};
```

```
        } else if(s == 1) {
```

```
            USART_SendData(USART2,0x68);//h
```

```
            while(USART_GetFlagStatus(USART2,
USART_FLAG_TXE)==RESET) {};
```

```
        } else {
```

```
            USART_SendData(USART2,0x69);//i
```

```
            while(USART_GetFlagStatus(USART2,
USART_FLAG_TXE)==RESET) {};
```

```
        }
```

```
        for(j=0;j<800;j++){ //retardo
```

```
        for(m=0;m<100;m++){}}
```

```
        }
```

```
        if(JOYSTICK_GetState()==JOYSTICK_LEFT)
```

```
        {
```

```
            // JOYSTICK_WaitForRelease();
```

```
            DRAW_DisplayString(5,20,"Izquierda...
```

```
            ",17);
```

```
            LED_Set (LED_RED,LED_OFF);
```

```
            LED_Set (LED_GREEN,LED_ON);
```

```
            if(s == 0) {
```

```
                USART_SendData(USART2,0x6A);//j
```

```
                while(USART_GetFlagStatus(USART2,
USART_FLAG_TXE)==RESET) {};
```

```
            } else if(s == 1) {
```

```
                USART_SendData(USART2,0x6B);//k
```

ANEXO C: CODIGO FUENTE

```
        while(USART_GetFlagStatus(USART2,
USART_FLAG_TXE)==RESET) {};
```

```
        } else {
```

```
            USART_SendData(USART2,0x6C);//l
```

```
            while(USART_GetFlagStatus(USART2,
USART_FLAG_TXE)==RESET) {};
```

```
        }
```

```
        for(j=0;j<800;j++){ //retardo
```

```
            for(m=0;m<100;m++){}}
```

```
        }
```

```
    }
```

```
    }
```

```
//FUNCIONES DEL TECLADO
```

```
void lista(){
```

```
    //listado del inventario
```

```
    if(s==0){
```

```
        DRAW_DisplayString(5,61,"Movil 1",17);//1
```

```
        let = '1';
```

```
    }else if(s==1){
```

```
        DRAW_DisplayString(5,61,"Movil 2",17);//2
```

```
        let = '2';
```

```
    }else if(s==2){
```

```
        DRAW_DisplayString(5,61,"Movil 3",17);//3
```

```
        let = '3';
```

```
    }
```

```
    }
```

```
    /*-----*/
```

```
    /*-----*/
```

```
    // Seleccion de items
```

```
void busqueda(){
```

```
    if(bandera == 0)
```

```
        {
```

```
            while(USART_GetFlagStatus(USART2,
USART_FLAG_TXE)==RESET) {};
```

```
            USART_SendData(USART2,0x6C);//l
```

```
            while(USART_GetFlagStatus(USART2,
USART_FLAG_TXE)==RESET) {};
```

```
        }
```

```
        for(j=0;j<800;j++){ //retardo
```

```
            for(m=0;m<100;m++){}}
```

```
        }
```

```
    }
```

```
    }
```

```
    if(JOYSTICK_GetState()==JOYSTICK_RIGHT)
```

```
        {
```

```
            JOYSTICK_WaitForRelease();
```

```
            bandera = 0;
```

```
            DRAW_DisplayString(5,78,"Seleccionando",17);
```

```
            s++;
```

```
        }
```

```
        if(JOYSTICK_GetState()==JOYSTICK_LEFT)
```

```
            {
```

```
                JOYSTICK_WaitForRelease();
```

```
                bandera = 0;
```

```
                DRAW_DisplayString(5,78,"Seleccionando",17);
```

```
                if(s == 0)
```

```
                    {
```

```
                        s = 2;
```

```
                    }
```

```
                else
```

```
                    {
```

```
                        s--;
```

```
                    }
```

```
                }
```

```
                if(s<0)
```

```
                    {
```

```
                        s = 0;
```

```
                    }
```

```
                }
```

```
                if(s>2)
```

```
                    {
```

```
                        s = 0;
```

```
                    }
```

```
                }
```

```
            }
```

ANEXO C: CODIGO FUENTE

```
    }  
}  
  
//Almacena item en vector de pedido e imprime la  
orden  
  
void seleccion(){  
    if(JOYSTICK_GetState()          ==  
    JOYSTICK_DOWN)  
    {  
        JOYSTICK_WaitForRelease();  
        LED_Set (LED_RED,LED_ON);  
        LED_Set (LED_GREEN,LED_OFF);  
        bandera = 1;  
        DRAW_DisplayString(5,78,"Seleccionado  
",17);  
        if(s == 0)  
        {  
            DRAW_DisplayString(5,40,"Sel.   Movil  
1",17);  
        }  
        else if(s == 1)  
        {  
            DRAW_DisplayString(5,40,"Sel.   Movil  
2",17);  
        }  
        else  
        {  
            DRAW_DisplayString(5,40,"Sel.   Movil  
3",17);  
        }  
    }  
}
```

ANEXO D

GUIA PARA CARGAR APLICACIONES EN EL PRIMER 2

Para poder cargar aplicaciones en el PRIMER 2, se recomienda borrar todas las aplicaciones demostrativas que posee el PRIMER 2 por defecto, para poder tener suficiente memoria flash libre para cargar las aplicaciones que se requieren. Para ello, se deben seguir los siguientes pasos:

1. Instalar el set de software de herramientas para el desarrollo de las aplicaciones para el PRIMER 2, dicho software se encuentra en el CD que viene con el PRIMER 2 o en el sitio web: <http://www.stm32circle.com>.

Nota: solo disponible para plataformas con sistema operativo Windows.

2. Conectar el PRIMER 2 mediante un cable USB entre el PC y el puerto "STM32" del PRIMER 2; y esperar a que el sistema operativo del PC reconozca el dispositivo.
3. Encender el PRIMER 2.
4. Acceder a la raíz en donde fue instalado el software. Por lo general dicha raíz es: Program Files/Raisonance/Ride/lib/ARM/CircleOS; en esta raíz se encuentra una aplicación llamada "Update_Primer2_Circle_OS.bat", se trata de una aplicación que se encarga de cargar en el PRIMER 2 solo el sistema operativo.
5. Ejecutar el archivo batch: "Update_Primer2_Circle_OS.bat".
6. Esperar a que el proceso de carga del OS termine.
7. Apagar el PRIMER 2 manteniendo pulsado el joystick del mismo.

Por otra parte, las aplicaciones empleadas para el control de los móviles, así como de la gestión de inventarios, pueden ser cargadas al PRIMER 2 de forma similar. Para ello se debe buscar dentro de los archivos adjuntos de este documento de grado por los archivos: "XIVA_COM" y "XIVA_REMOTO" y luego seguir los siguientes pasos para cualquiera de los dos.

1. Acceder a uno de estos archivos y abrir la carpeta con nombre: "objdebug".
2. Conectar el PRIMER 2 al PC mediante un cable USB que interconecte uno de los puertos USB del PC con el puerto "STM32" del PRIMER 2.
3. Encender el PRIMER 2 pulsando el joystick del mismo.

ANEXO C: CODIGO FUENTE

4. Buscar dentro de esta carpeta, un archivo batch llamado: "add_to_Circle.bat" y ejecutarla.
5. Esperar a que el proceso termine y apagar el PRIMER 2.

Nota: Estos pasos son validos para cargar cualquiera de las dos aplicaciones en el PRIMER 2.

Una vez ejecutados estos simples pasos, el PRIMER 2 quedaría programado con las dos aplicaciones empleadas para el sistema de gestión de inventarios.

Nota: los archivos con los códigos de dichas aplicaciones se pueden encontrar [aquí](#)

ANEXO E

MANUAL DE USUARIO DEL MODULO DE CONTROL



Figura 44 Módulo de control compuesto por un PRIMER 2 (vista frontal y posterior) y el módulo de comunicación inalámbrica.

El módulo de control consta de un PRIMER 2 conectado a la tarjeta de expansión para comunicación inalámbrica (ver figura.44), Con el cual el usuario puede ingresar la lista de ítems requeridos al sistema de gestión de inventario, haciendo uso de la interfaz gráfica de usuario (Graphic User Interface, GUI).

Descripción de los periféricos



Figura 45 Periféricos empleados. Pantalla LCD como periférico de salida (1) y Joystick como periférico de entrada (2).

ANEXO C: CODIGO FUENTE

Las aplicaciones para el ingreso de órdenes al sistema de gestión de inventarios a través del PRIMER 2 hacen uso principalmente del joystick (ver figura 45) para la navegación entre las aplicaciones, el ingreso de comandos y la orden de ejecución de los mismos; así mismo, las aplicaciones ejecutadas por el PRIMER 2 son visualizadas en la pantalla LCD de este.

Descripción de las aplicaciones

El modulo cuenta con dos aplicaciones: la primera, permite al usuario escoger uno o más agentes del sistema y luego permite el control de movimiento de este móvil o móviles seleccionados mediante la interacción con el joystick del PRIMER 2; la segunda aplicación es empleada para el ingreso de la lista de ítems requeridos por el usuario al sistema de inventarios, para que este le lleve los ítems requeridos al punto de entrega.

Uso de las aplicaciones

Control Remoto

1. Asegurarse que la tarjeta de expansión para comunicación inalámbrica se encuentre conectada adecuadamente al puerto de expansión del PRIMER 2 y que el LED de la tarjeta de expansión encienda de forma intermitente (ver figura 46).



Figura 46: Primer 2 conectado adecuadamente al módulo de comunicaciones.

2. Encender el PRIMER 2 pulsado el joystick.
3. Luego de encender el PRIMER 2, se debe seguir con la calibración de pantalla (si el sistema pide tal calibración) y esperar a llegar a la pantalla de inicio del PRIMER 2 (ver figura 47).



Figura 47: Pantalla de inicio del PRIMER 2.

4. Pulsar el joystick para acceder al menú del PRIMER 2, seleccionar “applic.” y presionar el joystick para acceder al menú de selección de aplicaciones, luego, proceder a seleccionar la aplicación llamada “Remoto” y pulsar el joystick para dar inicio a la aplicación.

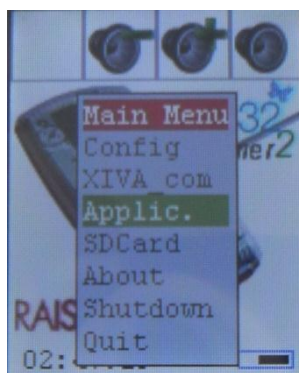


Figura 48: Menú principal del PRIMER 2.

5. Una vez iniciada la aplicación, se tendrá en pantalla la siguiente GUI:

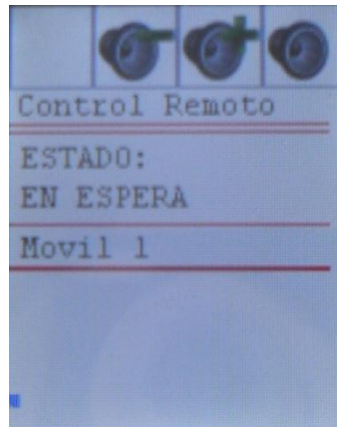


Figura 49: GUI para control remoto de los móviles.

Debe mover el joystick hacia la derecha o izquierda para seleccionar el móvil o móviles a controlar moviendo hacia abajo el joystick.

6. Una vez seleccionado el móvil o móviles, se visualiza en pantalla la confirmación de la selección, así como el cambio de estado de la aplicación de “seleccionando” a “seleccionado” con lo que se verifica que la selección es la correcta. Además, en pantalla se visualiza también el estado de dicho móvil seleccionado (ver figura 50).



Figura 50: Aplicación en espera de comandos.

Nota: los leds en la parte superior del PRIMER 2 indican el estado del móvil o móviles seleccionados. El led rojo indica que dicho móvil se encuentra en reposo o “parada”, por otra parte, el led verde indica que el móvil se encuentra realizando algún movimiento

7. Ahora estando en el modo de control (led rojo encendido), es posible emplear las direcciones del joystick para controlar los movimientos del móvil o móviles seleccionados.
(ver figura 51).



Figura 51: Aplicación enviando comandos a los móviles.

Nota: los leds del PRIMER 2 notifican el estado de movimiento (led verde) o de reposo (led rojo).

8. Para finalizar la aplicación debe mantener pulsado el joystick hasta que la aplicación se cierre y vuelva a la pantalla de inicio del PRIMER 2.
9. Para apagar el PRIMER 2 debe pulsar el joystick para acceder al menú principal, luego moviendo hacia arriba o hacia abajo el joystick seleccionar “shutdown” y luego pulsar el joystick. Al aparecer un menú de confirmación se selecciona “yes” para apagar el PRIMER 2.

Nota: Si no va a emplear el PRIMER 2 en los próximos minutos, se recomienda extraer la tarjeta del puerto de expansión.

Pedidos de Inventario

1. Seguir los pasos 1, 2 y 3 del procedimiento anterior que corresponde a la ejecución de la aplicación de control remoto de los móviles.
2. Pulsar el joystick para acceder al menú del PRIMER 2, seleccionar “applic.” y presionar el joystick para acceder al menú de selección de aplicaciones, luego, proceder a seleccionar la aplicación llamada “XIVA_com” y pulsar el joystick para dar inicio a la aplicación.
3. Luego de iniciar la aplicación, se tendrá la siguiente GUI:



Figura 52: GUI Para pedidos de inventario.

4. Búsqueda del ítem que va a la lista de petición, para ello, basta con mover el joystick a derecha o izquierda para tal fin.
5. Selección del ítem para almacenar en la lista del pedido, esto se logra moviendo el joystick hacia abajo.



Figura 53: Visualización de ítems en el pedido.

6. Repetir los pasos 4 y 5 hasta tener en la lista del pedido los ítems requeridos.
7. Para enviar la orden de pedido al sistema de gestión de inventario se mueve el joystick hacia arriba. Para confirmar que el pedido este siendo enviado, se disponen de leds indicadores del estado de transmisión, en donde, el led rojo indica que no hay transmisión de comandos y así mismo, el led verde indica la transmisión de los comandos.



Figura 54: Aplicación enviando la lista del pedido al sistema.

8. Verificar que en pantalla aparezcan los datos de pedido enviado, así como la notificación de envío exitoso.

Nota: luego de enviar un pedido, el sistema queda a la espera de más pedidos, sin la necesidad de tener que reiniciar la aplicación.

9. Para salir de la aplicación, se debe mantener pulsado el joystick hasta que la aplicación se cierre y aparezca de nuevo la imagen de inicio en el PRIMER 2.

Nota: Una vez fuera de la aplicación, se puede iniciar cualquiera de las dos aplicaciones siguiendo los pasos anteriormente mencionados.

10. Para apagar el PRIMER 2 debe pulsar el joystick para acceder al menú principal, luego moviendo hacia arriba o hacia abajo el joystick seleccionar "shutdown" y luego pulsar el joystick. Al aparecer un menú de confirmación se selecciona "yes" para apagar el PRIMER 2.

Nota: Si el PRIMER 2 no va a ser empleada en los próximos minutos, se recomienda extraer la tarjeta del puerto de expansión.