

Robot de seguridad controlado por WIFI “Cerberus 1.0”

Angie Tatiana Rengifo Oviedo

**Proyecto de grado para optar al título de
Tecnóloga en Electricidad**

Universidad Tecnológica de Pereira
Facultad de Tecnologías
Tecnología Eléctrica
Pereira, Colombia
Noviembre 2016

Robot de seguridad controlado por WIFI “Cerberus 1.0”

**Por:
Angie Tatiana Rengifo Oviedo
Código 1.088.316.183**

**Proyecto de grado para optar al título de
Tecnóloga en Electricidad**

**Director:
M.Sc. Hugo Baldomiro Cano Garzón
Profesor del Programa Tecnología Eléctrica
Ingeniero Electricista**

**Universidad Tecnológica de Pereira
Facultad de Tecnologías
Tecnología Eléctrica
Pereira, Colombia
Noviembre 2016**

Nota de aceptación:

Director:

Jurado:

Noviembre 2016

Dedicatoria

Que la dedicatoria de este trabajo de grado sea un pequeño reconocimiento y agradecimiento a las dos personas que siempre han dado lo mejor de si, con su apoyo incondicional y acompañamiento para alcanzar mis metas. Mis padres: Carlos Mario Rengifo y Marielli Oviedo.

Angie Tatiana Rengifo Oviedo.

Agradecimientos

“Nuestra recompensa se encuentra en el esfuerzo y no en el resultado. Un esfuerzo total es una victoria completa” Mahatma Gandhi

Son muchas las personas a las que debo agradecer por su ayuda y colaboración durante la ejecución de este proyecto y de toda mi carrera.

Para empezar un especial agradecimiento al M.Sc. Hugo Baldomiro Cano Garzón quien ha sido el director de éste proyecto porque a pesar de sus múltiples ocupaciones siempre ha tenido tiempo y disposición para guiar el desarrollo y evolución del mismo; al M.Sc. José Andrés Chavés Osorio por su respaldo, sus aportes y sus comentarios; al Ing. Edward Andrés González Ríos por el apoyo en el desarrollo del diseño mecánico, revisión del diseño electrónico y aportes en el sistema de comunicación; al Ing. Santiago Gómez Estrada director del programa de Tecnología Eléctrica por su acompañamiento durante toda la carrera.

A mi familia que me ha respaldado en todo momento durante el desarrollo de éste pregrado, mis abuelos, tías y especialmente a mis padres quienes siempre han estado presentes sin importar la dificultad o complejidad de la situación en la que me encuentre; a los miembros del grupo de investigación robótica aplicada que con sus aportes grandes o pequeños han contribuido en el mejoramiento del presente proyecto y a todas las personas que han aportado para alcanzar los objetivos de éste pregrado.

Angie Tatiana Rengifo Oviedo

Índice general

1. DESCRIPCIÓN DEL PROYECTO	9
1.1. INTRODUCCIÓN	9
1.2. DEFINICIÓN DEL PROBLEMA	10
1.3. JUSTIFICACIÓN	10
1.4. OBJETIVOS	10
1.4.1. OBJETIVO GENERAL	10
1.4.2. OBJETIVOS ESPECÍFICOS	10
2. MARCO REFERENCIAL	11
2.1. MARCO CONCEPTUAL	11
2.2. MARCO TEÓRICO	12
2.2.1. Sistema mecánico	12
2.2.1.1. Rueda diferencial.	12
2.2.1.2. Motores	12
2.2.2. Sistema sensorico	13
2.2.2.1. Sensor de visión	13
2.2.3. Sistema de inteligencia o control	15
2.2.3.1. Raspberry pi 2 modelo B+	15
2.2.4. Sistema de comunicación	15
2.2.4.1. Modelo TCP/IP	16
2.2.5. Sistema de alimentación	17
2.2.5.1. Baterías de polímero de Litio	17
2.2.5.2. Carga de baterías de polímero de litio	17
2.3. MARCO HISTÓRICO	18
3. DESARROLLO DEL PROYECTO	21
3.1. Sistema mecánico	22
3.2. Sistema electrónico	23
3.2.1. Subsistema de sensores	23
3.2.2. Subsistema de conexión de dispositivos	27
3.3. Sistema de inteligencia o control	28
3.3.1. Subsistema de control de motores	28
3.3.2. Subsistema de control y proceso de la información	29
3.4. Sistema de comunicación	33
3.5. Sistema de alimentación	35
4. PRUEBAS Y RESULTADOS	37
4.1. Sistema mecánico	37
4.2. Sistema electrónico	38
4.2.1. Subsistema de sensores	38
4.2.2. Subsistema de conexión de dispositivos	38
4.3. Sistema de inteligencia o control	39

4.3.1. Subsistema de control de motores	39
4.3.2. Subsistema de control y proceso de información	39
4.4. Sistema de comunicación	40
4.5. Sistema de alimentación	42
4.6. Pruebas generales de todo el sistema	42
5. CONCLUSIONES Y FUTUROS TRABAJOS	44
5.1. CONCLUSIONES	44
5.2. FUTUROS TRABAJOS	45
6. Anexos	46
6.1. Anexo Datos técnicos de Motores	46
6.2. Anexo Datos técnicos de la Cámara IP	48
6.3. Anexo datos técnicos Regulador de Tensión LM323k	49
6.4. Anexo Programa página de interfaz de usuario	52
Bibliografía	56

Índice de figuras

2.1. Tracción diferencial [Autor]	12
2.2. Sensor CCD [8]	14
2.3. Sensor CMOS [8]	14
2.4. Comparación entre el modelo OSI y el modelo TCP/IP [Autor]	16
2.5. Primeros robots móviles. Izquierda Robot de 1949, centro Robot de 1963, derecha Robot de los 60 [13]	18
2.6. Robots utilizados para la exploración de Marte. SOJOURNER y SPIRIT [13]	19
2.7. T-34 Robot de seguridad manejado por teléfono móvil [14]	19
2.8. Robot de Seguridad Aquiles [15]	19
2.9. Robot de seguridad MIR [16]	20
2.10. Prototipo robótico detector de minas antipersonas [17]	20
3.1. Partes de Cerberus 1.0 [Autor]	21
3.2. Diseño en SolidWorks de Cerberus 1.0 [Autor]	22
3.3. Motores de Cerberus 1.0 [Autor]	22
3.4. Cámara ip utilizada en Cerberus 1.0 [Autor]	23
3.5. Identificación de cámara ip [Autor]	24
3.6. Opciones del ipcamera Tool [Autor]	24
3.7. Página Web de la cámara [Autor]	25
3.8. Opciones de la cámara en la página web [Autor]	25
3.9. Administración de la cámara [Autor]	26
3.10. Configuración de red inalámbrica de la cámara [Autor]	26
3.11. Tarjeta General de Conexión [Autor]	27
3.12. Tarjeta conexión de carga [Autor]	28
3.13. Puente H utilizado en Cerberus 1.0[Autor]	29
3.14. Instalación del sistema operativo en la raspberry pi [Autor]	30
3.15. Configuraciones iniciales de la raspberry Pi [Autor]	30
3.16. Programa de Python para controlar el movimiento del robot [Autor]	32
3.17. Interfaz de usuario [Autor]	33
3.18. Configuraciones del router [Autor]	34
3.19. Funcionamiento de la red [Autor]	34
3.20. Cargador de baterías Turnigy 4x1000W [Autor]	35
3.21. Reconocimiento de baterías en el cargador [Autor]	36
4.1. Chasis Cerberus 1.0 modular [Autor]	37
4.2. Visión nocturna de la cámara [Autor]	38
4.3. Conexión tarjeta general y prueba de reconocimiento de carga [Autor]	39
4.4. Identificación de GPIO [Autor]	40
4.5. Escritorio de monitoreo [Autor]	41
4.6. Prueba en área local [Autor]	41
4.7. Verificación de ip publica [Autor]	42
4.8. Carga de Baterías dentro del prototipo [Autor]	43

Capítulo 1

DESCRIPCIÓN DEL PROYECTO

1.1 INTRODUCCIÓN

Las empresas enfrentan día a día las problemáticas que se dan en los sistemas de seguridad, como el robo de equipos, pérdida de mercancía, entre otros. En países subdesarrollados se utilizan equipos de seguridad fijos como cámaras, alarmas y personal de vigilancia; sin embargo, los equipos no se verifican constantemente con el fin de observar que las imágenes que envían las cámaras por ejemplo, estén actualizadas constantemente y con el ángulo que menos genere puntos ciegos (espacio o lugar que el equipo no puede observar lo que sucede); así mismo, la rotación que en ocasiones se genera en el personal de vigilancia hace que las personas nuevas no tengan suficiente conocimiento sobre las salidas de emergencia de la empresa, como reaccionar en caso de estar expuesto a un incendio o inundación, lo que hace vulnerable la integridad física de estas personas, especialmente en zonas industriales donde pueden estar expuestos a altas temperaturas o fugas de gases.

Mientras que a nivel internacional en países desarrollados, se podrá observar por medio de este documento algunos robots que han sido implementados con el fin de salvaguardar la vida de estas personas pero estos equipos son demasiados costosos. Por consiguiente, con Cerberus 1.0 se pretende iniciar el desarrollo de la primera etapa de un robot de vigilancia, el cual sea menos costoso y que en un futuro cercano pueda ser utilizado por las empresas nacionales.

Así mismo, se planteó el desarrollo de éste proyecto con el fin de utilizar los conocimientos aprendidos durante el programa de Tecnología Eléctrica de la Universidad Tecnológica de Pereira, debido a que contiene asignaturas relacionadas con los sistemas electrónicos y de comunicación que permiten la ejecución de éste trabajo de grado. Del mismo modo, al estar vinculada al Grupo de Investigación de Robótica Aplicada involucra el desarrollo práctico y cognitivo que a permitido el gusto y el conocimiento por el área de la robótica.

El siguiente documento, contiene una parte inicial del desarrollo tecnológico que se ha observado en la robótica y los diferentes sistemas que componen a un robot. También se podrá encontrar en el capítulo 3 el desarrollo del proyecto y en el capítulo 4 las pruebas realizadas y los resultados obtenidos.

1.2 DEFINICIÓN DEL PROBLEMA

Los sistemas de vigilancia implementados actualmente en Colombia, presentan en ocasiones problemas como: puntos ciegos, debido a la ubicación de los equipos u ángulo de visión que estos poseen; fallas no detectadas en el funcionamiento de las cámaras de vigilancia, ya que se tiene un estándar de revisión de cada 24 horas [1] el cuál es determinado por la Superintendencia de Vigilancia y Seguridad Privada en el protocolo de operación de servicios de vigilancia electrónica; y el riesgo que sufre el personal de vigilancia al estar expuesto a posibles eventos catastróficos que puedan afectar su integridad física.

Éstas dificultades pueden ser solucionadas por medio de sistemas como robots móviles que realizan “rondas” de vigilancia teleoperados, lo cual disminuye los puntos ciegos de los equipos que se encuentran fijos debido que al estar el robot en movimiento permite observar diferentes ángulos y zonas, del mismo modo se puede verificar el estado de los equipos de seguridad puesto que al pasar el robot por donde están los otros sistemas, éstos deberían actualizar la imagen donde se encuentra el robot y los vigilantes tendrían un riesgo menor, ya que no se verían expuestos en las rondas a eventos que puedan ocasionarles lesiones.

1.3 JUSTIFICACIÓN

CERBERUS 1.0 es una herramienta de vigilancia la cual reduce los riesgos a los que puede estar expuestos los vigilantes fuera del centro de control como incendios, inundaciones, fuga de gases, derrame de químicos, explosiones, personas peligrosas, entre otros; ya que al ser teleoperado desde la central el personal de vigilancia puede verificar el estado de las zonas sin necesidad de ir hasta éstas. Del mismo modo, permite apoyar los equipos de seguridad estáticos con el fin de disminuir los puntos ciegos y verificar el estado de éstos.

A nivel internacional, existen diferentes robots de vigilancia con características técnicas como visión artificial, monitoreo remoto, navegación autónoma, entre otras; sin embargo, estos equipos son demasiado costosos lo cual hace que en Colombia sean poco utilizados por las empresas de seguridad. Con Cerberus 1.0 se podría implementar en las empresas con un costo menor comparado con el de importación siendo el control de forma remota una primera etapa.

1.4 OBJETIVOS

1.4.1. OBJETIVO GENERAL

Construir un prototipo robótico controlado por WiFi para implementación en seguridad, con movilidad en superficies planas y sólidas, siendo supervisado de forma remota.

1.4.2. OBJETIVOS ESPECÍFICOS

- Recopilar información sobre Robótica móvil y sobre la construcción de prototipos robóticos.
- Implementar de forma modular la estructura electrónica y mecánica del prototipo robótico.
- Desarrollar un sistema de comunicación inalámbrica entre el robot y un computador remoto, por medio de WiFi.
- Construir de forma simplificada la plataforma de alimentación del robot.
- Realizar pruebas de operación y funcionamiento del prototipo robótico.

Capítulo 2

MARCO REFERENCIAL

2.1 MARCO CONCEPTUAL

- Robot: Es una máquina programable capaz de realizar una acción o trabajo de forma automática con la capacidad de adaptar su actividad en función a la información que percibe del entorno [2].
- Red de área local (LAN): Es aquella que conecta a computadores y servidores, se extiende pocos metros y generalmente se utiliza para compartir recursos comunes como periféricos, datos o aplicaciones.
- Red de área extendida (WAN): Está compuesta por varias redes LAN y es capaz de transmitir información a miles de kilómetros, donde la más conocida es Internet.
- Servidor dedicado: Es aquel que no realiza aplicaciones como cliente, sino que se optimiza para responder a peticiones [3].
- WiFi: Es un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica, también es llamada WILAN o estándar IEEE802.11 [4].
- Lenguaje de programación: Es un sistema de comunicación estructurado que establece una serie de reglas sintácticas y semánticas para definir adecuadamente una secuencia de instrucciones que puedan ser interpretadas y ejecutadas en una computadora o máquina [5].
- Computador de placa reducida: Es un dispositivo el cual tiene las características de un computador, es decir, tiene entradas periféricas, sistema operativo, procesador, etc. Sin embargo, las dimensiones de éste son muy inferiores a las de un computador.
- Tracción diferencial: Es la fuerza de movimiento que se realiza en las ruedas de un vehículo, donde se regula esta fuerza motriz para corregir o controlar la diferencia de tracción entre ruedas.
- Puente H: Es un circuito electrónico que contiene un arreglo de transistores el cual permite controlar el giro y la velocidad de un motor de corriente directa.

2.2 MARCO TEÓRICO

La estructura general de un robot se compone de tres sistemas: mecánico, sensorio y de control [6]. Del mismo modo, se tiene en cuenta otros sistemas que son el electrónico, la comunicación y la alimentación, los cuales en la unión de los diferentes sistemas principales se hacen fundamentales para su buen funcionamiento.

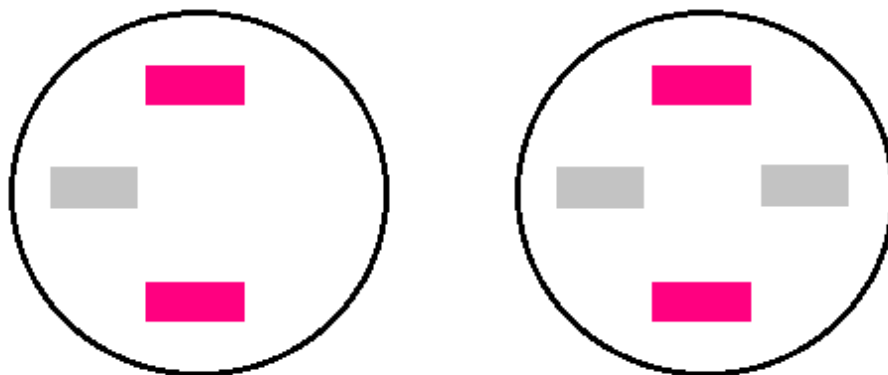
2.2.1. Sistema mecánico

En el sistema mecánico se tiene en cuenta la estructura o chasis que debe llevar el robot según el ambiente al que va a estar expuesto, su estructura interna, los motores que se vayan a utilizar y su peso. Dependiendo de la distribución del peso con respecto al centro de gravedad del robot se determina que tan estable es éste.

De igual forma, el movimiento, locomoción o desplazamiento del robot se puede realizar a través de ruedas, orugas o patas. En el movimiento por medio de ruedas dependiendo de la maniobrabilidad, el control, la tracción, la capacidad de subir pendientes, la estabilidad, la eficiencia, entre otros que se necesite, se elige el tipo de configuración o arreglo como: rueda diferencial, sincrona, tipo triciclo, tipo carro, omnidireccional, etc [7]. Se explica a continuación la rueda de tracción diferencial la cual se utiliza en Cerberus 1.0.

2.2.1.1. Rueda diferencial.

Es el mecanismo más sencillo que consta de dos ruedas sobre un mismo eje pero controladas de forma independiente. Puede moverse en línea recta, en arco y girar sobre su propio eje y generalmente lleva una rueda de apoyo como se muestra en la figura 2.1a o dos ruedas de apoyo como se muestra en la figura 2.1b, donde en la segunda posee mayor estabilidad.



(a) Tracción diferencial con una rueda de apoyo (b) Tracción diferencial con dos ruedas de apoyo

Figura 2.1: Tracción diferencial [Autor]

2.2.1.2. Motores

Los motores son actuadores que se han utilizado desde años atrás, por medio del movimiento cinemático del agua, del vapor, de la combustión interna y a partir de Werner von Siemens quien inventó la dinamo en 1866 se comenzó a usar motores por medio de energía eléctrica. El motor eléctrico es una máquina destinada a convertir energía eléctrica en mecánica a través del efecto electromagnético que se produce entre el rotor y el estátor del motor. Existen motores eléctricos tipo asíncrono, síncrono, universal, paso a paso, servomotor, corriente continua, entre otros.

El motor de corriente continua es muy utilizado en la industria debido a su fácil control de posición, par y velocidad para aplicaciones de potencia y de precisión. Dependiendo de la conexión de las bobinas de excitación interna del motor donde la más utilizada es la excitación independiente que tiene velocidad proporcional al valor de la tensión media de corriente continua con condiciones constantes de excitación y par mecánico es muy utilizado en la robótica móvil a pesar de que el sistema de alimentación utilizado es por medio de baterías y estas se ven sujetas al cambio de tensión en el momento de descarga. Otro tipo de motores utilizados en la robótica son los motores paso a paso, servomotores y micromotores.

2.2.2. Sistema sensoríoico

El sistema Sensoríoico brinda la información del entorno para cumplir con las funciones definidas por el equipo; utiliza el principio de los sentidos humanos para captar los valores de las variables físicas y los lleva al sistema de control en forma de energía eléctrica. El sistema sensoríoico o de detección se debe diseñar de tal forma que siempre la comparación que éste realice sea confiable.

Existen sensores analógicos y digitales, los sensores de tipo analógico, son sensores que convierten una magnitud física en una señal analógica, principalmente en una señal de tensión o corriente eléctrica. Los de tipo digital son aquellos que convierten una magnitud física en una señal binaria, principalmente en una señal eléctrica con dos estados (on/off). Algunos de ellos son: sensor mecánico, sensor inductivo, sensor capacitivo, sensor óptico, sensor ultrasónico, sensor de visión, etc. Para Cerberus 1.0 se utiliza el sensor de visión ya que permite reconocer la localización y forma de los objetos existentes cerca del mismo.

2.2.2.1. Sensor de visión

En este tipo de sensor el componente principal es la cámara incluyendo la electrónica asociada, el formato de señal de salida y un lente. La mayoría de las cámaras tienen el software, de manera tal que se puedan realizar grabaciones y la imagen pueda ser tratada desde el computador. Para elegir este sensor los parámetros más importantes son el campo de vista, la resolución, la profundidad de campo y el índice de adquisición de datos.

De igual manera, están conformadas por un compartimiento oscuro y cerrado con un orificio en el extremo para que la luz ingrese por medio de un juego de lentes que permiten enfocar el entorno que se desea capturar y a partir de un pequeño chip compuesto por millones de componentes sensibles a la luz, llamados píxeles; una vez que la luz entra es capturada y se obtiene la imagen. Los sensores utilizados por las cámaras se pueden clasificar según su tecnología y el tipo de cámara que se tenga. Los más usuales en las cámaras son el sensor CCD y el CMOS[8].

El sensor de carga acoplada CCD es un circuito integrado que convierte las cargas de cada celda de la matriz de píxeles en niveles de tensión y entrega una señal analógica a la salida donde es digitalizada por medio de otros componentes de la cámara. En los sensores CCD se realiza la lectura de cada uno de los valores correspondiente a cada celda y luego un convertidor análogo-digital los traduce en forma de datos tal como se muestra en la Figura 2.2[9].

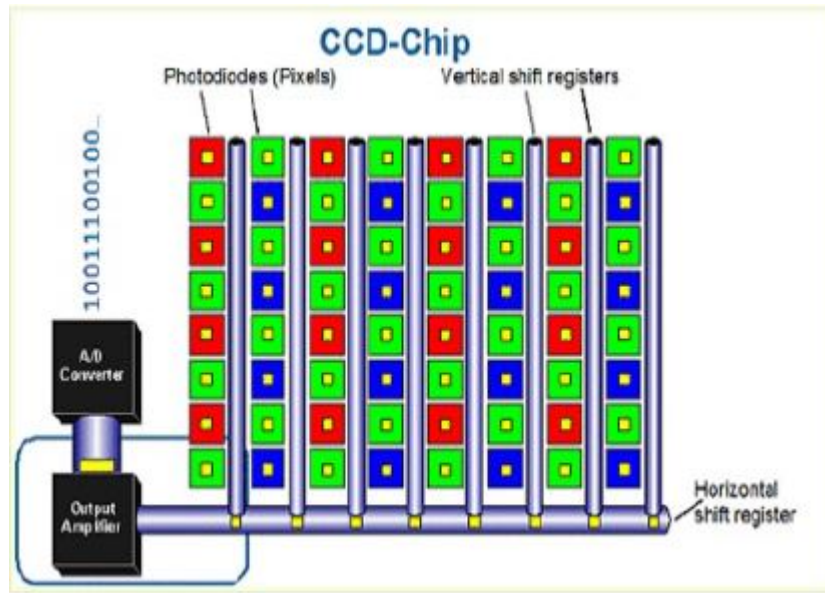


Figura 2.2: Sensor CCD [8]

El sensor semiconductor complementario del óxido de metal CMOS tiene cada celda independiente y la principal diferencia con el sensor anterior es que la digitalización de los píxeles se realiza internamente por medio de transistores que tiene cada celda; por lo tanto, no necesita un chip externo que realice el trabajo, tal como se muestra en la Figura 2.3 . Estos sensores son más sensibles a la luz y al estar expuestos a entornos con poca iluminación, se comportan mejor, así como, realizan todo el procesamiento internamente aumentando la velocidad.

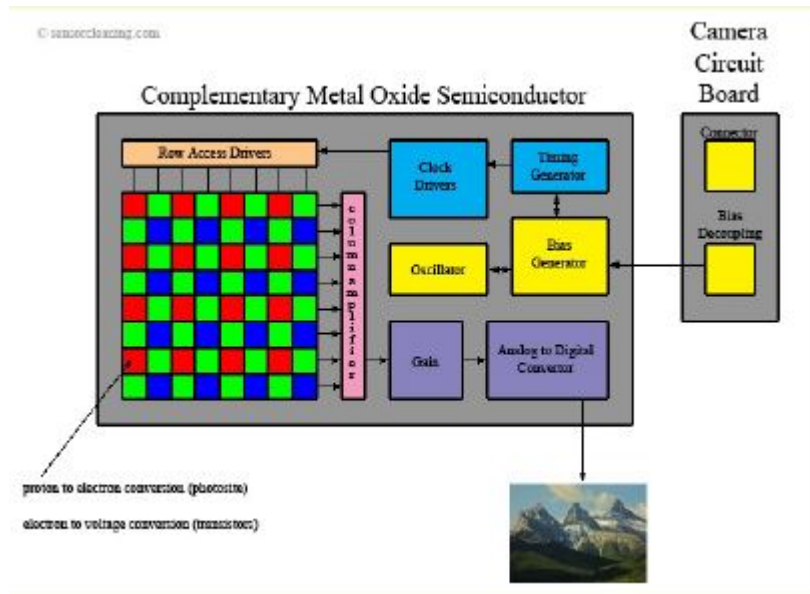


Figura 2.3: Sensor CMOS [8]

2.2.3. Sistema de inteligencia o control

Es el sistema principal del robot ya que procesa toda la información y envía las órdenes para que éste realice la operación correspondiente. Los sistemas de control por lo general, manejan uno o varios lenguajes de programación para realizar el control y la comunicación entre los demás sistemas.

Se puede decir que es el cerebro del robot, por consiguiente dependiendo de la simplicidad del robot se puede utilizar diferentes controladores como lo son las FPGA, los microprocesadores, los computadores de placa reducida, las tarjetas de desarrollo, los PLC, entre otros.

Para Cerberus 1.0 se usó un computador de placa reducida Raspberry pi, el cual ha tenido gran auge en la academia y existen diferentes modelos donde en este trabajo se habla de la raspberry pi 2 modelo B+.

2.2.3.1. Raspberry pi 2 modelo B+

La raspberry pi es un computador de placa reducida el cual ha tenido varios modelos, entre ellos el que se utiliza para Cerberus 1.0 es la Raspberry pi 2 B+ que cuenta con las siguientes características:

- 1 CPU de cuatro núcleos ARM Cortex-A7 de 900MHz
- 1 GB de RAM
- 4 puertos USB
- 40 pines GPIO
- Puerto HDMI
- Puerto Ethernet
- Conector combinado de audio de 3,5 mm y de vídeo compuesto
- Interfaz de la cámara (CSI)
- Interfaz de pantalla (DSI)
- Ranura para tarjetas micro SD VideoCore IV 3D núcleo de gráficos

Debido a que tiene un procesador ARMv7 , se puede ejecutar toda la gama de distribuciones ARM GNU / Linux y Microsoft Windows 10. También, se le pueden instalar diferentes lenguajes de programación que facilitan la interpretación entre el hardware y el usuario. Por lo tanto se tienen en cuenta los siguientes lenguajes de programación utilizados:

- Python: Es un lenguaje de programación interpretado, lo que significa que no necesita compilar el código fuente para poder ejecutarlo. Esta enfocado al desarrollo tanto de aplicaciones como de páginas web, es popular ya que es gratuito y tiene varias librerías y datos que permiten hacer desarrollos sin tener que comenzar desde cero.
- HTML: Es un lenguaje con el que se definen las páginas web, en consecuencia se pueden combinar e incorporar otros lenguajes para definir el formato más apropiado dependiendo de las características que se quieran tener. Este lenguaje se escribe en un editor de texto sencillo con extensión .html y al abrir este archivo permite observar los resultados en cualquier navegador que se tenga instalado.
- PHP: Es un lenguaje de programación del lado del servidor, es decir, se ejecuta en el servidor web justo antes de que se envíe la página al cliente a través de Internet. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red y otras tareas para crear la página final que verá el cliente. Éste solamente recibe una página con el código HTML resultante de la ejecución de PHP.

2.2.4. Sistema de comunicación

El sistema de comunicación permite que los dispositivos del robot puedan operar e intercambiar información entre ellos de una manera secuencial. Por otra parte, cuando se quiere tener una comunicación entre el robot y otro dispositivo, también se debe tener unos equipos que permitan la transmisión y recepción de datos, donde estos pueden ser alámbricos o inalámbricos. Los alámbricos se pueden comunicar por cables de

cobre o por fibra óptica y los inalámbricos a través de ondas de radio. En cada uno se pueden tener diferentes protocolos (reglas para el intercambio de mensajes) de acuerdo a lo que se requiera.

Existen diferentes protocolos según la aplicación, donde se tiene de referencia el modelo OSI el cual especifica qué tipo de protocolo debe ser usado en cada capa de la comunicación entre dispositivos. Este modelo está dividido en 7 capas: Física, enlace de datos, red, transporte, sesión, presentación y aplicación [10]. Para Cerberus 1.0 utilizamos el modelo TCP/IP el cual es un conjunto de protocolos que se basa en el modelo OSI.

2.2.4.1. Modelo TCP/IP

El modelo TCP/IP fue creado inicialmente para fines militares con el fin de cumplir criterios como la división de mensajes en paquetes, uso de un sistema de direcciones, enrutamiento de datos por la red y la detección de errores en las transmisión de datos.

Éste modelo une dos protocolos, el protocolo de control de transmisión y el protocolo de Internet; el cual al estar basado en el modelo OSI es un modelo que se maneja por capas y por esta razón se divide en cuatro (4) capas como se muestra en la figura 2.4 donde se procesa sucesivamente los datos que circulan por la red y se le agrega información al paquete con el fin de definir cuidadosamente cómo se mueve la información desde el remitente hasta el destinatario y se garantiza la transmisión. Las capas del modelo TCP/IP son: Acceso a la red, Internet, transporte, aplicación.

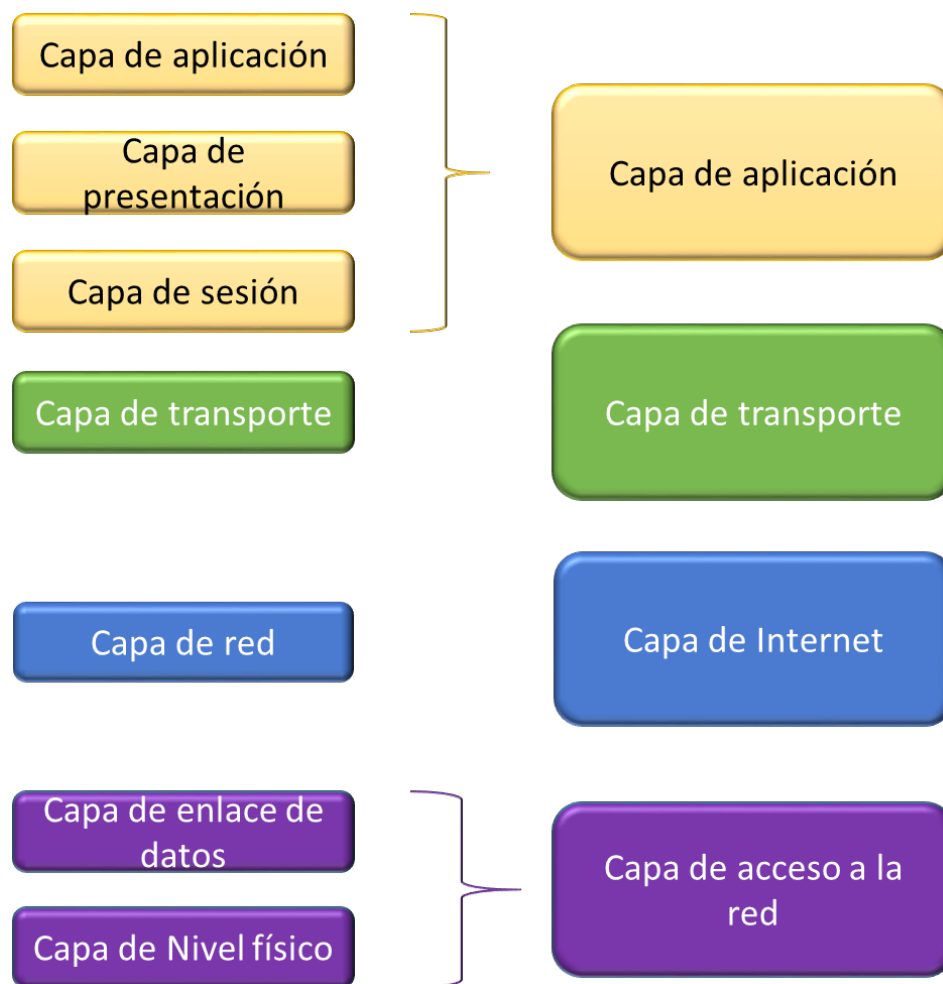


Figura 2.4: Comparación entre el modelo OSI y el modelo TCP/IP [Autor]

La capa de acceso a la red específica la forma en que los datos se deben codificar dependiendo del tipo de hardware que se vaya a utilizar si es por ethernet (IEEE802.3) o WiFi (IEEE802.11); así mismo, le asigna la dirección MAC tanto de origen como de destino y se realiza la traducción físico-eléctrico a lógica computacional.

La capa de Internet es responsable de asignar la dirección ip de origen y de destino; la capa de transporte brinda los datos de enrutamiento junto con los mecanismos que permiten conocer el estado de la transmisión, asigna si se envía por TCP (Transmission Control Protocol) o UDP (User Datagram Protocol) donde por lo general el vídeo se envía por medio de UDP. También, asigna los puertos de operación y en caso de que no sean puertos conocidos para poder ingresar se utilizan sockets que están formados por la combinación **ip:puerto** utilizados.

La capa de aplicación incorpora aplicaciones de red estándar, es decir, es la capa donde se procesa toda la interfaz con el usuario, donde se generan y llegan los datos.[11]

2.2.5. Sistema de alimentación

El sistema de alimentación consta de baterías que poseen una potencia eléctrica con el cual energizan los demás sistemas y le permiten al robot funcionar. Éstos dispositivos almacenan energía utilizando procesos electroquímicos como la óxido-reducción debido a que poseen un material metálico (celdas) que generalmente va al ánodo y materiales que se reducen al ganar electrones que van al cátodo. Dichos materiales posteriormente devuelven la energía casi en su totalidad y pueden repetir este procedimiento un determinado número de veces dependiendo del tipo de batería. Algunas baterías sufren de efecto memoria que es un fenómeno que produce cristalización de los componentes internos de las baterías y hace que se reduzca la capacidad de energía. Las baterías de polímero de litio son muy utilizadas en la robótica por su versatilidad debido a que poseen una buena capacidad de potencia en relación con su peso aunque son demasiado delicadas y peligrosas.

2.2.5.1. Baterías de polímero de Litio

Las baterías de Polímero de iones de Litio (Li-Po) son una de las principales fuentes de energía portátil actualmente usadas, debido a que no sufren de efecto memoria, tienen alta densidad de carga y son libres de mantenimiento. Sin embargo, su sensibilidad a la descarga profunda, sobrecarga y fácil explosión hacen que estas sean tratadas con bastante cuidado y siempre se respete la tensión (3V por celda de descarga profunda y 4,2V por celda máximo de carga), los límites de corriente y la temperatura para no ocasionar accidentes. También, hay que tener en cuenta que la resistencia interna de una celda de Lipo típicamente es de 2 a 6 miliohms (0,002 a 0,006ohms) donde ésta resistencia aumenta con la edad de la batería disminuyendo la tasa de descarga de la misma.

Estas baterías pueden estar conectadas en paquetes de serie o paralelo para alcanzar una tensión o corriente de funcionamiento determinada; algunos paquetes pueden consistir en una combinación de conexiones en serie y en paralelo por ejemplo, las baterías de los portátiles suelen tener cuatro celdas 3,6V en serie para alcanzar 14,4V y dos en paralelo para aumentar la capacidad de 2,400mAh a 4,800mAh. Tal configuración, se denomina 4S2P, es decir, cuatro celdas en serie y dos en paralelo. Los aislantes de aluminio entre las celdas evita que la parte metálica conductora de las baterías puedan causar un cortocircuito eléctrico; para esto, es importante utilizar el mismo tipo de celda con la misma tensión y la misma capacidad (Ah), nunca mezclar diferentes marcas y tamaños. Una celda más débil podría causar un desequilibrio, esto es especialmente crítico en una configuración en serie debido a que ésta celda se agotará más rápido que las otras que son más fuertes y en el momento de carga se llenará antes que las otras y permanecerá en exceso de carga mientras las otras se cargan.

2.2.5.2. Carga de baterías de polímero de litio

La carga y descarga de una batería se rige por la tasa de descarga “C” que es la rapidez con que una batería puede ser descargada de forma segura, por ejemplo, la capacidad de una batería es comúnmente de una potencia de 1C, lo que significa que una batería completamente cargada nominal de 1Ah debe suministrar

1A durante una hora, así mismo, 1C también se conoce como una descarga de una hora, 0,5C o $C / 2$ es una descarga de dos horas y 0,2C o $C / 5$ es una descarga de 5 horas.

Así mismo, en el proceso de carga de las baterías siempre se debe realizar vigilancia para poder reaccionar ante cualquier problema que se pueda presentar como por ejemplo, la sobrecarga que puede provocar un incendio si se deja mucho tiempo; se debe tener en cuenta al seleccionar el cargador el número de elementos en serie que tiene la batería y el tipo de carga que se desea aplicar. Existen varios tipos de carga entre ellos carga lenta, carga rápida, carga por pulsos (goteo) y carga por flotación sin embargo, el más utilizado es la carga de tres etapas, donde se combinan la carga rápida, la lenta y la de flotación. En la primera etapa, que es la carga rápida, el cargador intentará alcanzar la tensión requerida para una carga lenta o etapa de absorción, pero al estar la batería descargada la corriente intentará aumentar rápidamente debiendo ser limitada por el cargador. En esta primera etapa la corriente de carga se mantiene constante y la tensión de la batería (o de carga) aumenta lentamente. En la segunda etapa que es de absorción o carga lenta la batería está aproximadamente un 80% recargada. El cargador en esta etapa va disminuyendo la corriente y la tensión en la batería se mantiene constante. La transición entre la etapa de absorción y la siguiente etapa se determina por tiempo o midiendo la corriente de carga y cuando ésta disminuya de cierto valor, se cambia a la tercera etapa que es la de flotación donde una forma de realizarla es mantener el cargador en un modo de tensión constante. El valor de esta tensión estará algunas décimas por encima de la tensión (en vacío) de una batería cargada. Otra forma es apagar el cargador y medir la tensión de la batería, cuando la tensión disminuye de algún valor prefijado, el cargador se enciende nuevamente con un valor de tensión constante [12].

2.3 MARCO HISTÓRICO

El avance de la robótica ha demostrado grandes pasos desde el siglo XX, sin embargo, los primeros robots móviles aparecieron en 1949; estos tenían una forma mecánica experimental como se aprecia en la figura 2.5



Figura 2.5: Primeros robots móviles. Izquierda Robot de 1949, centro Robot de 1963, derecha Robot de los 60 [13]

Desde de 1963, las aplicaciones prácticas que se le han dado a los robots móviles han sido muy grandes y satisfactorias; una de estas es la exploración de otros planetas como Marte. En 1997, llegó a Marte el primer robot móvil llamado SOJOURNER, el cual era controlado desde la tierra e incluía navegación autónoma con reconocimiento de obstáculos por un sensor láser; el cual se puede apreciar en el lado izquierdo de la figura 2.6 y en el 2004, llegó a Marte un robot más sofisticado llamado SPIRIT, mostrado a la derecha en la figura 2.6; el cual cuenta con control remoto desde la tierra pero también con un sistema autónomo el cual tiene una mejor navegación ya que posee una visión estereoscópica para el reconocimiento de obstáculos [13].



Figura 2.6: Robots utilizados para la exploración de Marte. SOJOURNER y SPIRIT [13]

En el 2009 Tmsuk Co Ltd. y Alacom Co Ltd. ubicadas en Japón, desarrollaron un robot de seguridad controlado por el teléfono móvil T-34 mostrado en la figura 2.7. Es utilizado para la prevención de delincuencia, funciona por sí mismo y no hay necesidad de hacer un sistema de control especial [14].



Figura 2.7: T-34 Robot de seguridad manejado por teléfono móvil [14]

En Barcelona se encuentra la Empresa AQUILES ROBOTIC SYSTEMS que cuenta con un robot para la seguridad humana llamado Aquiles protector que tiene un sistema electromecánico capaz de operar en entornos contaminados y hostiles. Se le pueden agregar accesorios como lo son brazos robóticos, telémetros y focos de Leds como se aprecia en la figura 2.8; además porta información relativa de la ubicación del robot por medio de GPRS [15].



Figura 2.8: Robot de Seguridad Aquiles [15]

En Corea se tiene el robot llamado MIR mostrado en la figura 2.9; fue desarrollado en el 2006 y hasta el momento se tiene en el mercado. Posee un sistema operativo desarrollado en Linux y un sistema mecánico de rueda diferencial, sensores de ultrasonido y sensores táctiles; realiza almacenamiento de imágenes por medio de una videocámara y utiliza baterías de litio-ion [16].



Figura 2.9: Robot de seguridad MIR [16]

Por último, a nivel institucional, tenemos diferentes robots desarrollados de forma inalámbrica por comunicación RF, utilizados para el beneficio de la vida humana. Uno de ellos, es el robot desarrollado en el Grupo de Investigación Robótica Aplicada de la Universidad Tecnológica de Pereira, llamado Angus mostrado en la figura 2.10, el cual es un robot detector de minas anti-personas cuyo contenido en su mayoría es metálico y además muestra una potencial ruta segura para una desactivación posterior y/o trabajo en la zona de exploración [17].



Figura 2.10: Prototipo robótico detector de minas antipersonas [17]

Capítulo 3

DESARROLLO DEL PROYECTO

Como se mencionó en el marco teórico, la estructura general de un robot se compone de varios sistemas como se muestra en la figura 3.1, los cuales se tuvieron en cuenta en el desarrollo del proyecto y para tener coherencia con la estructura del documento se describe a continuación en el orden de la sección 2.2.

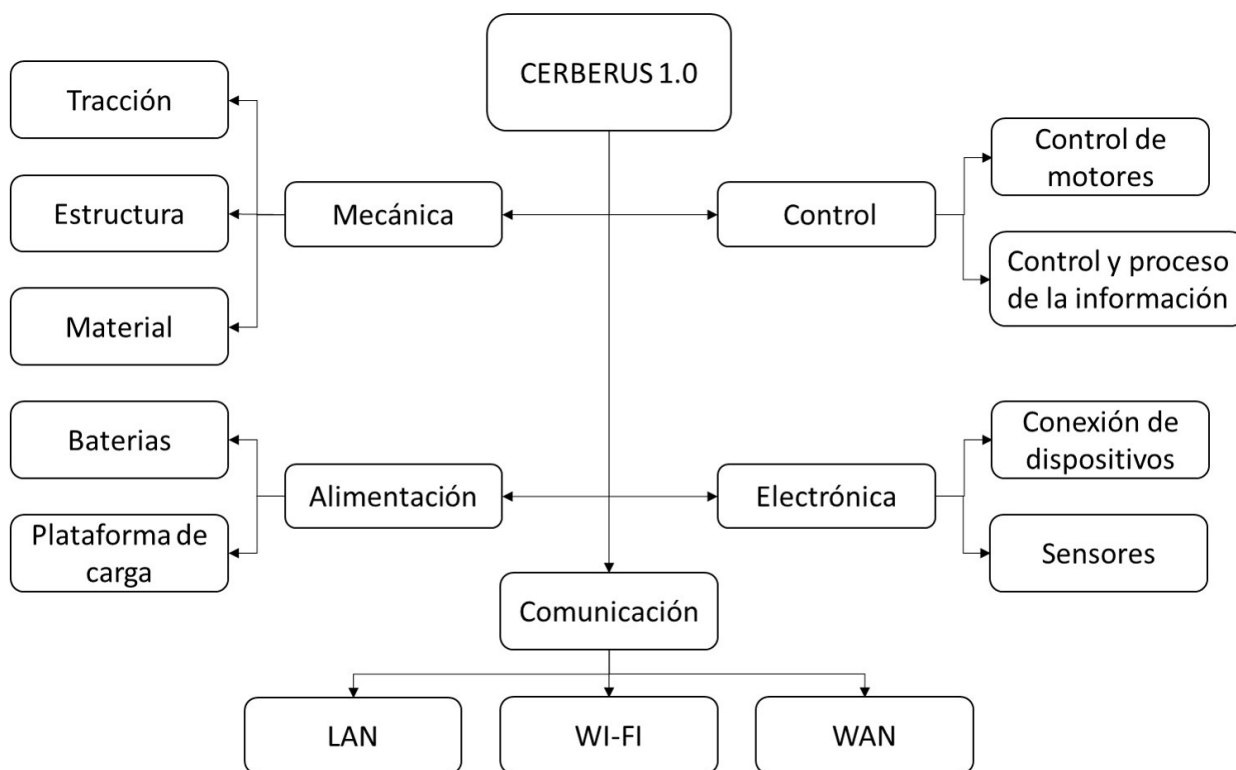


Figura 3.1: Partes de Cerberus 1.0 [Autor]

3.1 Sistema mecánico

Se realizó el diseño en SolidWorks de cada componente para tener en cuenta las dimensiones de cada uno y así poder realizar todos los ajustes necesarios en la carcasa o chasis. Así mismo al ser un prototipo se eligió como material para la carcasa el acrílico ya que es un polímero fácil de trabajar, no conduce electricidad, soporta aproximadamente 60°C antes de deformarse y es capaz de soportar el peso de los elementos que lleva internamente el robot.

Teniendo en cuenta la distribución que se realiza internamente en el robot, para que sus dimensiones externas sean pequeñas, se le incorpora un sistema de tracción de rueda diferencial el cual permite que sea lo más estable posible.

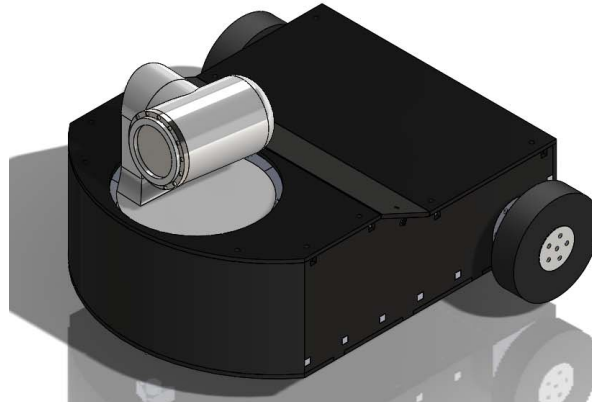


Figura 3.2: Diseño en SolidWorks de Cerberus 1.0 [Autor]

Debido al peso aproximado de las piezas del robot, se eligió los motores de corriente continua con caja reductora marca pololu con relación 50:1 de 200rpm y torque de 12kg-cm, los cuales tienen una tensión nominal 12V y corriente pico de 2,5A. En el anexo 6.1 se puede encontrar las especificaciones técnicas y medidas del diseño.

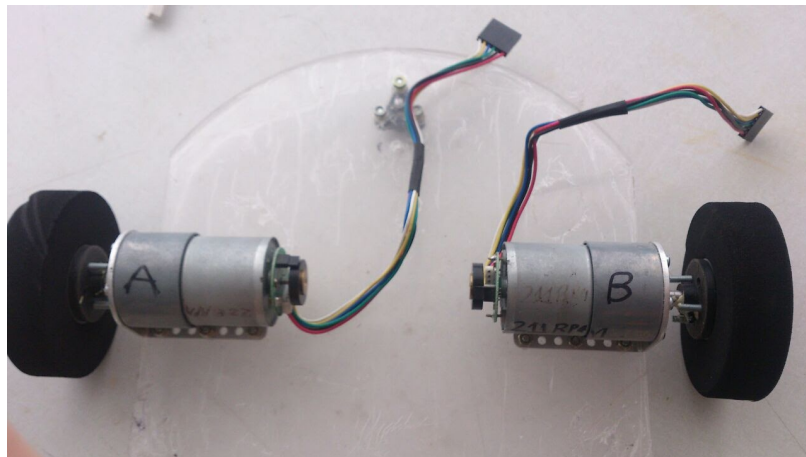


Figura 3.3: Motores de Cerberus 1.0 [Autor]

3.2 Sistema electrónico

3.2.1. Subsistema de sensores

En el sistema sensorial se utilizó una cámara inalámbrica para interiores como sensor de visión la cual tiene zoom óptico de 3X, posee 12 leds infrarrojos que le permite una visión nocturna de 15 metros nominal, permite tomar vídeos y fotos instantáneas de forma manual o configurar que al detectar un movimiento se tome automáticamente. Esta ultima opción no se utilizó puesto que el robot se encuentra continuamente en movimiento y en consecuencia enviá fotos y alarmas cada vez que el robot se desplaza. También, posee audio de dos vías a través de un parlante y un micrófono que tiene incorporado. Las especificaciones técnicas se pueden observar en el anexo 6.2.



Figura 3.4: Cámara ip utilizada en Cerberus 1.0 [Autor]

Para poder hacer uso de la cámara, se conecta directamente al computador o al router por cable de red para así, garantizar que quede conectada en la misma red del equipo con el que se va a configurar la cámara. Se instala en el computador el programa ipcamera, con el que se puede identificar la ip de la cámara con la que esta conectada en la red como se muestra en la figura 3.5.

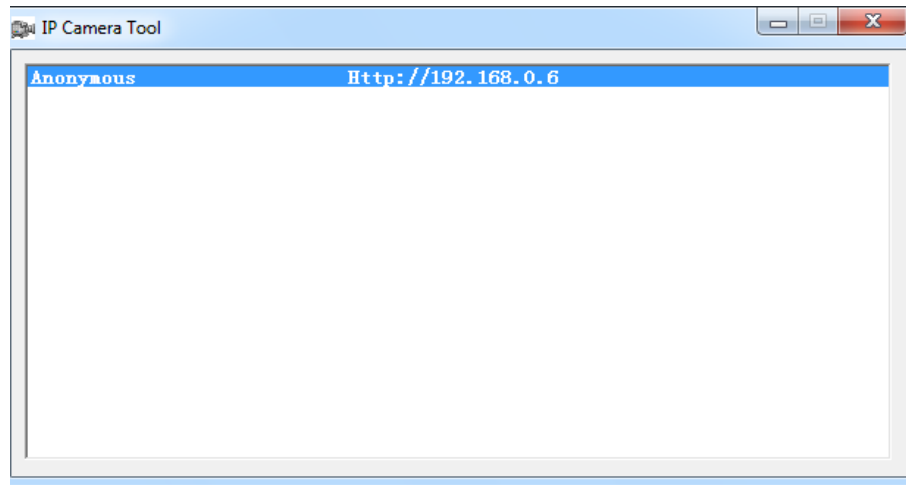


Figura 3.5: Identificación de cámara ip [Autor]

Al dar clic derecho encima de la ip que aparece identificada como se muestra en la figura 3.6 se presentan varias opciones como por ejemplo, las propiedades básicas que al darle clic muestra la versión del sistema, el ID y la versión web en la que fue diseñado el software de la cámara.

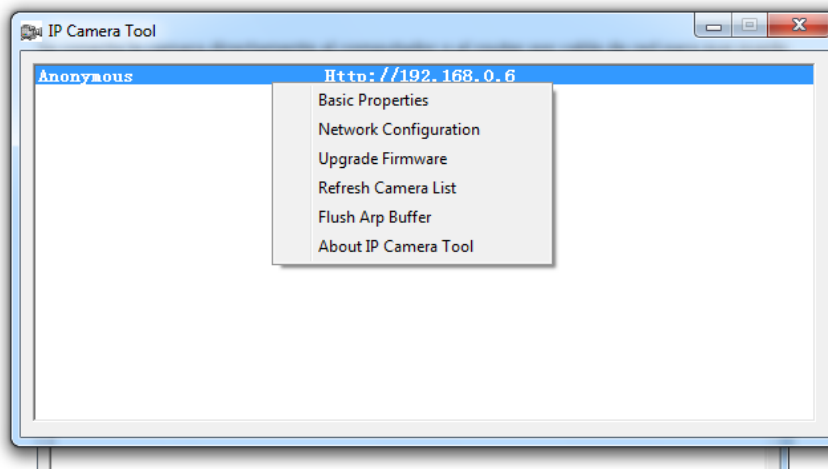


Figura 3.6: Opciones del ipcamera Tool [Autor]

En las configuraciones de la red presenta la dirección ip, la máscara de red, el puerto de entrada, el puerto HTTP usado, el usuario y la contraseña de la cámara; en la actualización del sistema, muestra la opción de actualizar la contraseña del usuario por defecto así como la versión del sistema y la versión web del software que viene por defecto.

Este software solo se usa inicialmente para saber la dirección ip con la cual esta conectada la cámara a la red para así ingresar al navegador web e ingresar en la página que trae diseñada internamente por defecto, como se muestra en la figura 3.7.

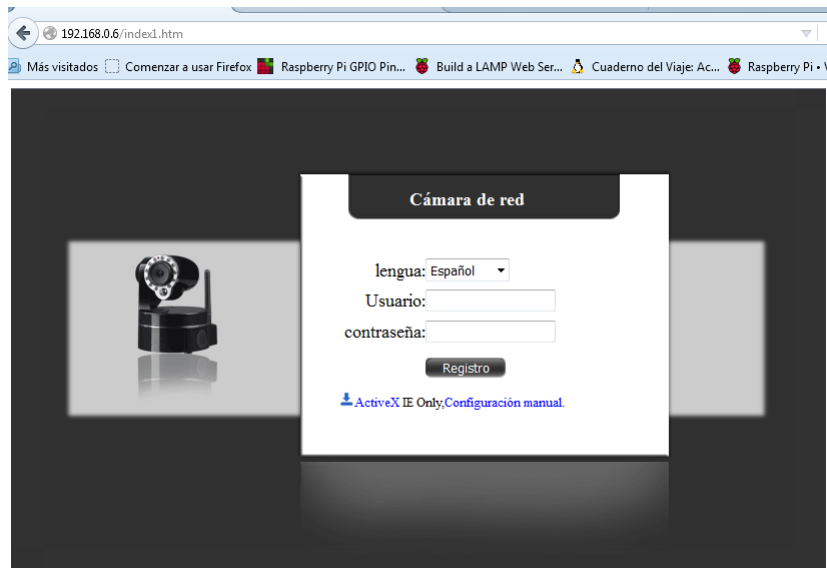


Figura 3.7: Página Web de la cámara [Autor]

Al ingresar en el navegador web la dirección ip de la cámara y conectarse a la página de ésta, se ingresa el usuario y contraseña que trae preconfigurado (usuario: admin y el espacio de la contraseña vacía). De allí, esta muestra una primera pantalla donde se encuentran los datos generales de la cámara a la cual se está conectando; al lado izquierdo se encuentra las diferentes opciones para su configuración como se muestra en la figura 3.8. En live Video se observa la imagen que detecta la cámara y las diferentes opciones que tiene para su uso. Es de recordar que esta cámara tiene opción de zoom óptico y por lo tanto se muestra en las opciones el zoom, el desplazamiento vertical, horizontal y diagonal. También esta la opción de pausar la imagen, de realizar y guardar una secuencia para que ésta la realice de manera autónoma, configurar la resolución a la que se quiere ver y configurar el modo al que va a estar expuesta la cámara, es decir a la frecuencia de la luz que va a estar expuesta donde en este caso sería 60 Hz en un ambiente interno que este alumbrado por luz artificial. Así mismo da la opción de tomar fotos instantáneas, de configurarle el brillo y el contraste, entre otras.

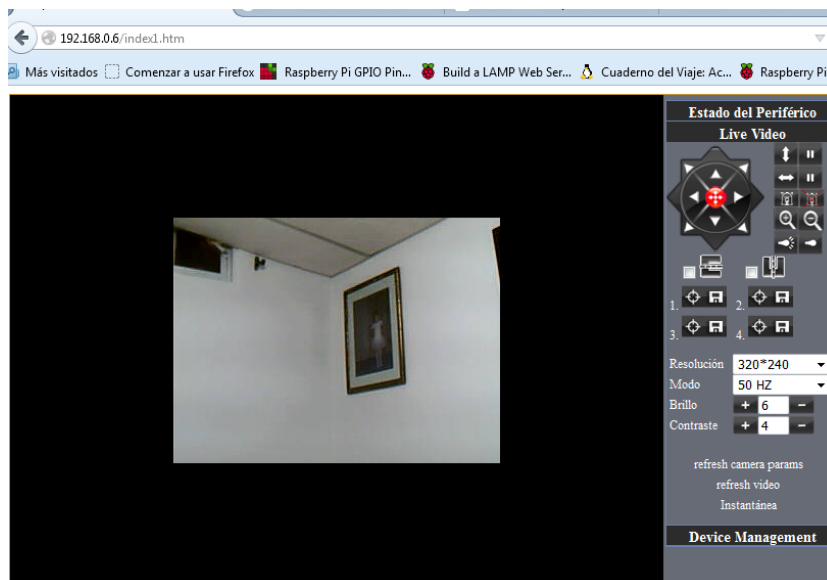


Figura 3.8: Opciones de la cámara en la página web [Autor]

En la parte de administración de la cámara (Device management) como se muestra en la figura 3.9, primero está la configuración del alias donde se le asigna un nombre para que aparezca en la red interna, para este caso se le puso Cerberus 1.0.; en fecha y hora, se le configuró la zona horaria; en la parte de configuración de usuarios se le puede configurar diferentes usuarios, con diferentes permisos como: Administrador, Operador o Visitante. Para mayor seguridad se le cambió el usuario y contraseña. De ahí para poder ingresar a la cámara desde otra red, se le cambio la configuración del puerto http y se le configuró una ip estática en el router donde va a estar conectada la cámara, con socket 192.168.0.13:3200.

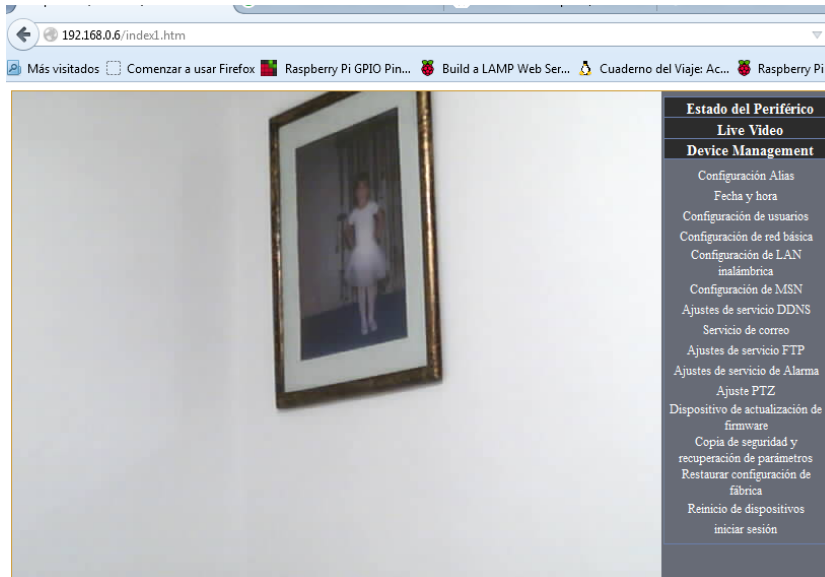


Figura 3.9: Administración de la cámara [Autor]

Ya que la cámara permite la conexión inalámbrica, se le configuró la red para así no tener que usar un cable de red para la misma y poder ubicarla en cualquier parte como se muestra en la figura 3.10.

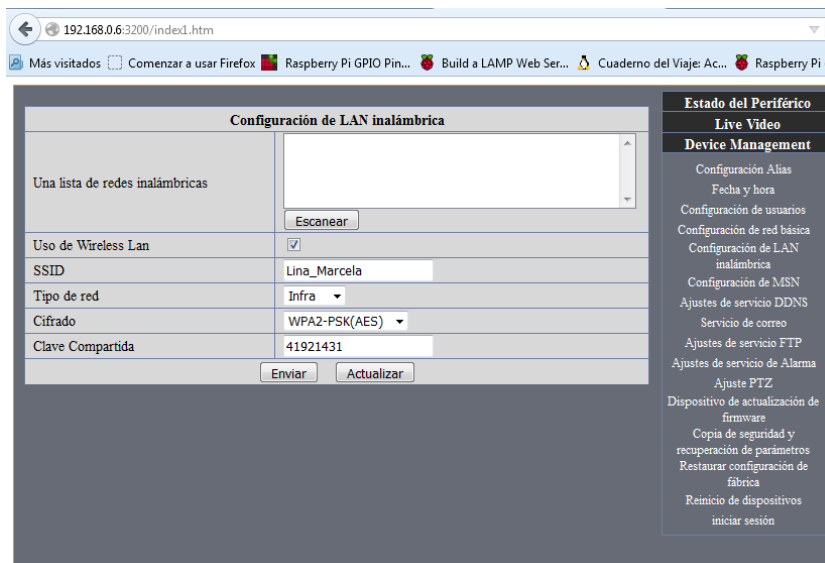


Figura 3.10: Configuración de red inalámbrica de la cámara [Autor]

3.2.2. Subsistema de conexión de dispositivos

En la conexión o unión de todos los componentes del prototipo se realizó una tarjeta general, donde se involucra también la regulación de tensión ya que las baterías son de 7,4V y la tensión a la que se alimenta la raspberry pi y la cámara es de 5V. Así mismo, se realizó la conexión de la carga de las baterías y se montó encima de esta tarjeta el puente H para que todas las salidas y entradas de alimentación y conexión estuvieran unidas por medio de la tarjeta general. En la figura 3.11a se puede observar el diseño del esquemático donde se usaron los reguladores lm323k debido a sus características técnicas que se pueden verificar en el anexo 6.3; en la figura 3.11b y 3.11c se observa el diseño impreso y el montaje de la tarjeta.

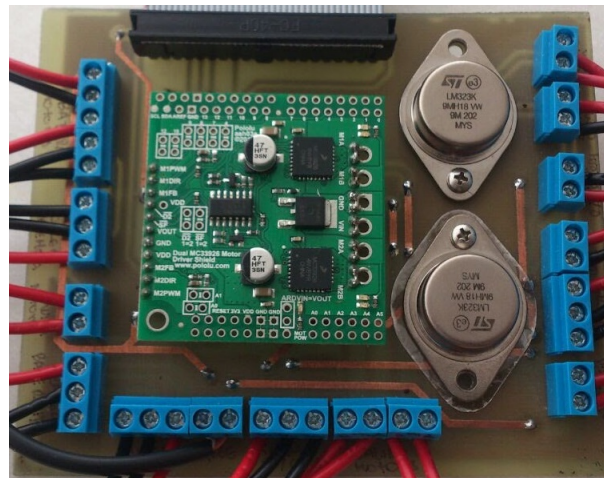
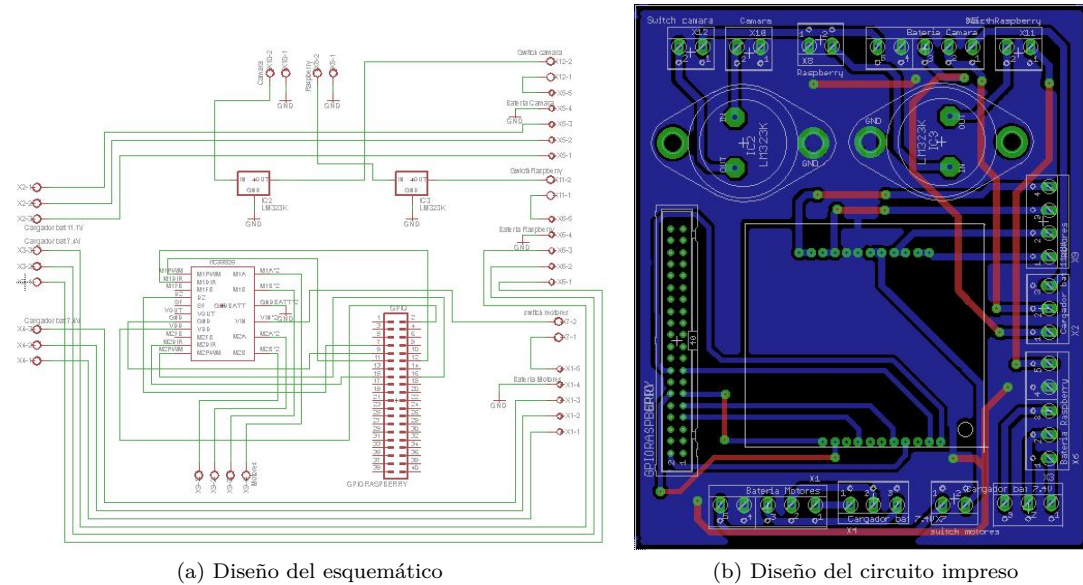


Figura 3.11: Tarjeta General de Conexión [Autor]

Para la conexión del robot al cargador, se decidió que se conectara por medio de patch cord por su facilidad de manejo y de conexión rápida entre dispositivos. Dado esto, se realizó también las tarjetas donde se realiza la conexión de los cables que van a las baterías y cargador respectivamente hacia el toma jack, en el cual se conecta el patch cord como se muestra en la figura 3.12 que muestra el diseño del esquemático, el circuito impreso y el montaje de la tarjeta.

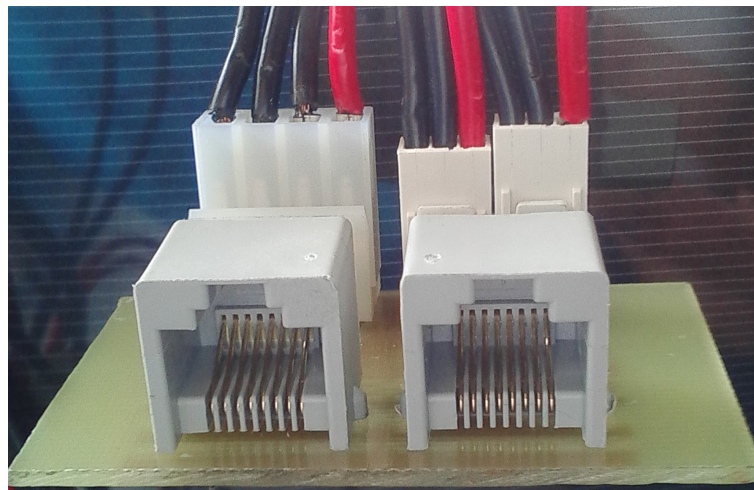
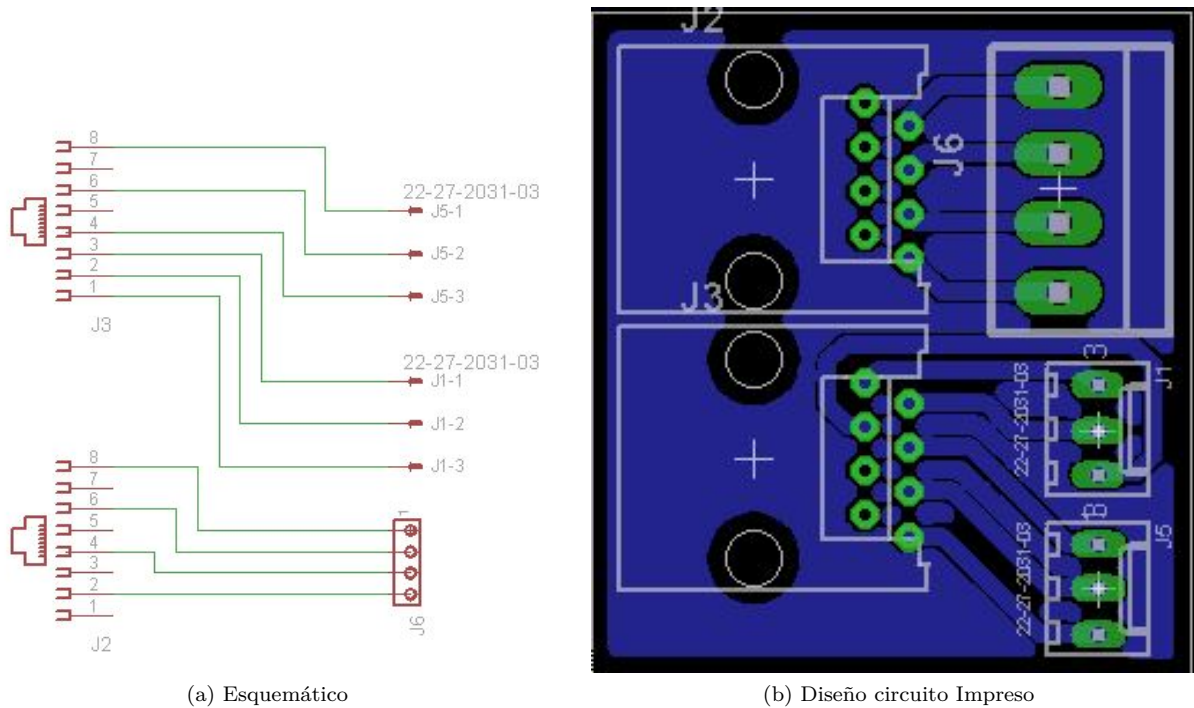


Figura 3.12: Tarjeta conexión de carga [Autor]

3.3 Sistema de inteligencia o control

En este sistema se tiene en cuenta, la parte de control de motores y de control de todo el prototipo, donde en este último se habla del control y proceso de la información, ya que se realiza toda la parte de interfaz de usuario para el movimiento del robot.

3.3.1. Subsistema de control de motores

Para realizar el control de los motores se decidió utilizar el puente H referencia MC33926 marca pololu, el cual soporta por cada canal 3A continuos y 5A pico debido a que los motores tienen una corriente pico de 2,5A cada uno, donde este puente H, cumple lo exigido en protección de equipos para la corriente solicitada

por cada motor. Para el control de los motores por medio del puente H mencionado, se utilizan los pines de entrada los cuales son dirección, PWM, Habilitación (D2), alimentación lógica (VCC y GND). El pin de dirección al poner un 1 lógico hace que el motor gire en un sentido y al poner un 0 lógico gira en sentido contrario al anterior, con el pin de PWM podemos controlar la velocidad con la cual va a operar el motor y el pin de habilitación a pesar de que aparece como D2 negado, se habilita al poner un 1 lógico en este pin. La alimentación lógica esta separada de la de potencia por lo tanto, la alimentación lógica debe ser la misma con la cual se ponen los 1 o 0 lógicos en los otros pines para que las tierras lógicas estén unificadas. En la salida tiene los pines que van hacia los motores, donde se tiene la alimentación y conexión de estos.

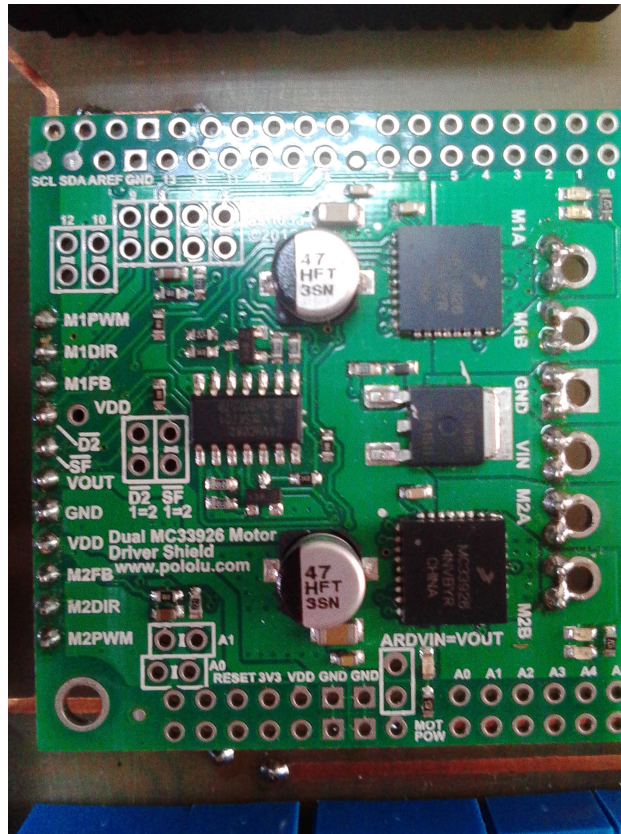


Figura 3.13: Puente H utilizado en Cerberus 1.0[Autor]

3.3.2. Subsistema de control y proceso de la información

En este subsistema se utilizó la raspberry pi 2 modelo B+ que se describe en la subsección 2.2.3 a la cual se le instaló el sistema operativo raspbian wheezy que es una distribución de linux donde a diferencia de las otras distribuciones es más fácil de manipular los permisos de acceso y permite usarla por medio de escritorio o simplemente por la interfaz de línea de comandos . Para poder hacer uso del computador de placa reducida, se debe realizar algunas configuraciones junto con la instalación de raspbian como se describe a continuación.

Para realizar la instalación de raspbian en la raspberry pi se debe tener en cuenta que la tarjeta de memoria SD debe tener una capacidad mínima de 4GB para que pueda soportar el sistema operativo a instalar. Teniendo en cuenta esto, se descarga la imagen del sistema operativo y se formatea completamente la tarjeta de memoria (Se puede formatear con el software SDFormatter). También se descarga el software Win32DiskImager el cual permite realizar la instalación del sistema operativo en la SD. Al ejecutar éste programa, se muestra un icono de carpeta donde se selecciona la imagen de raspbian y en device la ubicación de la tarjeta SD como se muestra en la figura 3.14.

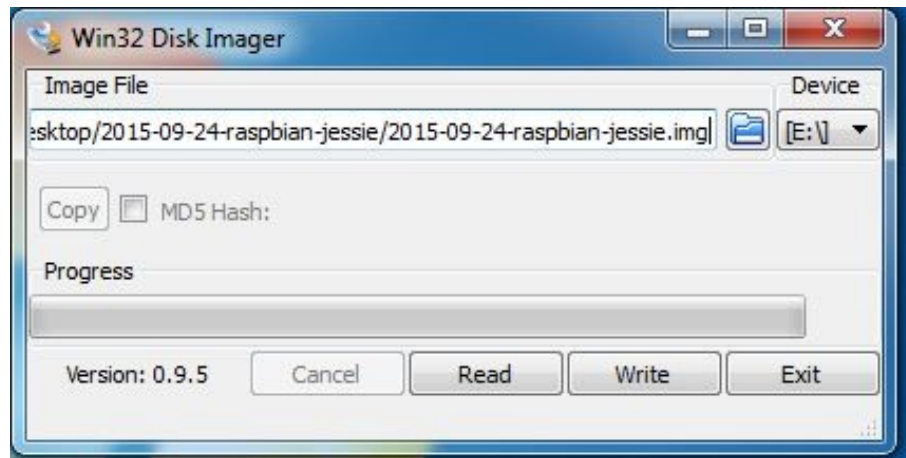


Figura 3.14: Instalación del sistema operativo en la raspberry pi [Autor]

De allí se procede a darle clic en write y esperar a que termine el proceso. Al terminar se da clic en exit, se saca la tarjeta sd del equipo y se inserta en la raspberry pi. Se realiza la debida conexión del computador de placa reducida (adaptador de energía, teclado y cable de red) y se comienza a desarrollar las configuraciones iniciales del mismo. En ocasiones, no aparece directamente la pantalla de configuraciones, entonces por medio de la pantalla de línea de comandos se utiliza la instrucción **sudo raspi-config** para que salga como se muestra en la figura 3.15.

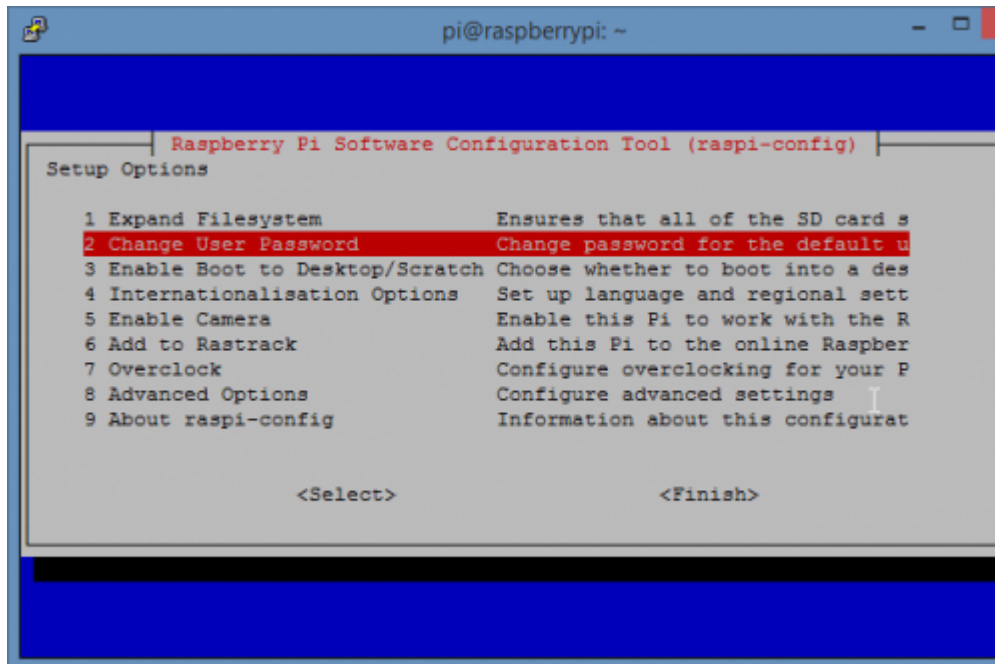


Figura 3.15: Configuraciones iniciales de la raspberry Pi [Autor]

La primera opción, es para que la raspberry pi pueda usar toda la capacidad de la SD puesto que la imagen del sistema operativo es menor a la capacidad de la SD y la raspberry pi ve el tamaño de la capacidad de la sd como si fuera solo el tamaño de la imagen del sistema operativo que trae por defecto, por ejemplo, si la sd es de 8GB y la imagen es de 32MB, el computador de placa reducida leerá la SD como si tuviera una capacidad de 32MB, entonces al seleccionar ésta opción, ya leerá la SD con su capacidad completa de 8GB.

La segunda opción, es para cambiar la clave que trae inicialmente la raspberry pi. Para este caso se dejó igual, donde el usuario es pi y la contraseña es raspberry pi. La tercera opción, es para elegir como desea que arranque la raspberry pi en el momento de ser encendida, si por consola de comandos o por escritorio con o sin logearse. La cuarta opción, es para configurar como desea que se comporte la raspberry pi al iniciar con red, para este caso se eligió que sea lenta para que se pueda tener la conexión de la red antes de arrancar completamente. La quinta opción, es para configurar la localidad, la hora y el teclado. En la configuración de la localidad, se elige para Colombia: es_CO.UTF-8 UTF-8 y cuando da la opción por defecto se da enter en en_GB.UTF-8; en la configuración de la hora, se elige como área geográfica América, zona horaria Bogotá y para la configuración del teclado se elige según el modelo de teclado, para este caso se utilizó el Generic 105-key (Intl) Pc, de allí se elige la opción diseño del teclado otro, spanish (latín American), de nuevo spanish (latín American) – Spanish (latín American, include dead tilde, de ahí se elige la opción de la función altGR: Right Alt (AltGR), la composición de la clave: Right Alt (AltGr) y con uso de la opción control+alt+backspace para terminar la ventana de comandos.

La sexta opción permite habilitar el puerto para insertar la cámara compatible con la raspberry pi, la séptima es para añadir la raspberry pi a la página donde se registra las diferentes personas en el mundo que usan este computador, que en este caso se utilizó con el usuario: Cerberus y correo: atrengifo@utp.edu.co. La octava es para configurar a otra frecuencia el reloj interno de la raspberry pi, por defecto está configurado a 700MHz. La novena opción, son opciones avanzadas donde está la configuración de la imagen de pantalla, el nombre del host, espacio para la gpu, ssh, device tree, puerto SPI, puerto I2C, puerto serial, puerto de audio y por último actualizaciones.

Para poder tener conexión remota habilitamos SSH donde éste queda habilitado por defecto por el puerto 22. Para Cerberus 1.0 después de realizar las modificaciones previas de configuración inicial se modificó el puerto de ssh por el puerto 4000 a través del siguiente comando “Sudo cd /etc/ssh Sudo nano ssh_config y donde dice el puerto se le cambia por 4000”, luego, se actualiza los archivos básicos de la raspberry pi con el comando “sudo apt-get update && sudo apt-get upgrade”.

Como se hace uso de los puertos GPIO se configura instalando la librería git-core y la Python-rpi.gpio, de allí para validar que hallan instalado correctamente se da la instrucción gpio readall y este muestra en la consola de comandos como están distribuidos los pines de manera lógica y física. Así mismo, se instala los programas apache2, php5, la librería libapache2-mod-php5 los cuales permiten que la raspberry pi funcione como servidor. Al terminar de instalar, se reinicia la raspberry pi.

De tal manera se verifica si quedó bien instalado ingresando a la ip de la raspberry pi desde otro equipo o desde la misma en el navegador con http://localhost. Para poder acceder a la carpeta del servidor, se modifican los permisos a través de comandos y por último se configura la raspberry pi para que quede con una ip fija por medio del comando “sudo cp /etc/network/interfaces interfaces.old sudo nano -w /etc/network/interfaces” y en la línea iface eth0 inet manual se cambia en address la dirección que se quiere. quedando de la siguiente forma:

```
iface eth0 inet static
address 192.168.0.11
netmask 255.255.255.0
gateway 192.168.0.1
```

Tan pronto como se realizan las configuraciones iniciales de la raspberry pi se comienza a programar los diferentes secuencias en python para controlar el movimiento del robot. Para ello se pensó que pudiera trasladarse hacia adelante, atrás, derecha e izquierda con una velocidad normal y una rápida, donde el usuario pudiera elegir cual de las dos utilizar. Así mismo, en la interfaz del usuario por medio de un botón pudiera apagar el robot para así no estar apagandolo de forma brusca. Teniendo en cuenta esto, se realizó un programa para cada opción de movimiento y para el botón de apagado. Todos los programas de movimiento, sigue la secuencia que se muestra en la figura 3.16, la diferencia esta en el cambio de estado bajo o alto de los pines de control de dirección.

```

pi@raspberrypi: ~
GNU nano 2.2.6      File: adelante.py

# Librerias utilizadas
import time
import RPi.GPIO as GPIO
import parar

# Configuracion de modo de puertos GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

# Configuracion inicial de los puertos GPIO

GPIO.setup(18, GPIO.OUT) # Motor 1A
GPIO.setup(17, GPIO.OUT) # Motor 1B
GPIO.setup(23, GPIO.OUT) # Motor 2A
GPIO.setup(22, GPIO.OUT) # Motor 2B
GPIO.setup(4,GPIO.OUT)  # Habilitador D2 del PH

#SELECCION DE FRECUENCIA PARA LOS PUERTOS

motor1dir= GPIO.output(22,GPIO.HIGH) # direccion
motor1pwm= GPIO.PWM(23,18) # channel=23 frequency=100Hz

motor2dir= GPIO.output(18,GPIO.HIGH) # direccion
motor2pwm= GPIO.PWM(17,20) # channel=17 frequency=100Hz

motor1pwm.start(25) #inicializacion del ciclo de los motores
motor2pwm.start(25) #inicializacion del ciclo de los motores

D2=GPIO.output(4,True)

#CICLO CONTINUO PARA QUE SE MANTENGA EL MOVIMIENTO
if (D2 == True):
    while (D2):
        motor1pwm.ChangeDutyCycle(5) #inicializacion del ciclo de los motores
        motor2pwm.ChangeDutyCycle(5) #inicializacion del ciclo de los motores
    else:
        motor1pwm.stop()
        motor2pwm.stop()

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page  ^U UnCut Text ^T To Spell

```

Figura 3.16: Programa de Python para controlar el movimiento del robot [Autor]

Teniendo esto, se procedió a realizar el diseño de la página web la cual es la interfaz de usuario, donde se integró el código de python y php para facilitar la programación. Este se realizó a partir de la opción de formularios, con el fin de que al momento de oprimir un botón este funcionara como un submit y con la etiqueta post se enviará el nombre de la opción y con el código en php se ejecutara el archivo de python. El código de la página se puede observar en el anexo 6.4 y al ingresar a la página de interfaz se ve como en la figura 3.17.

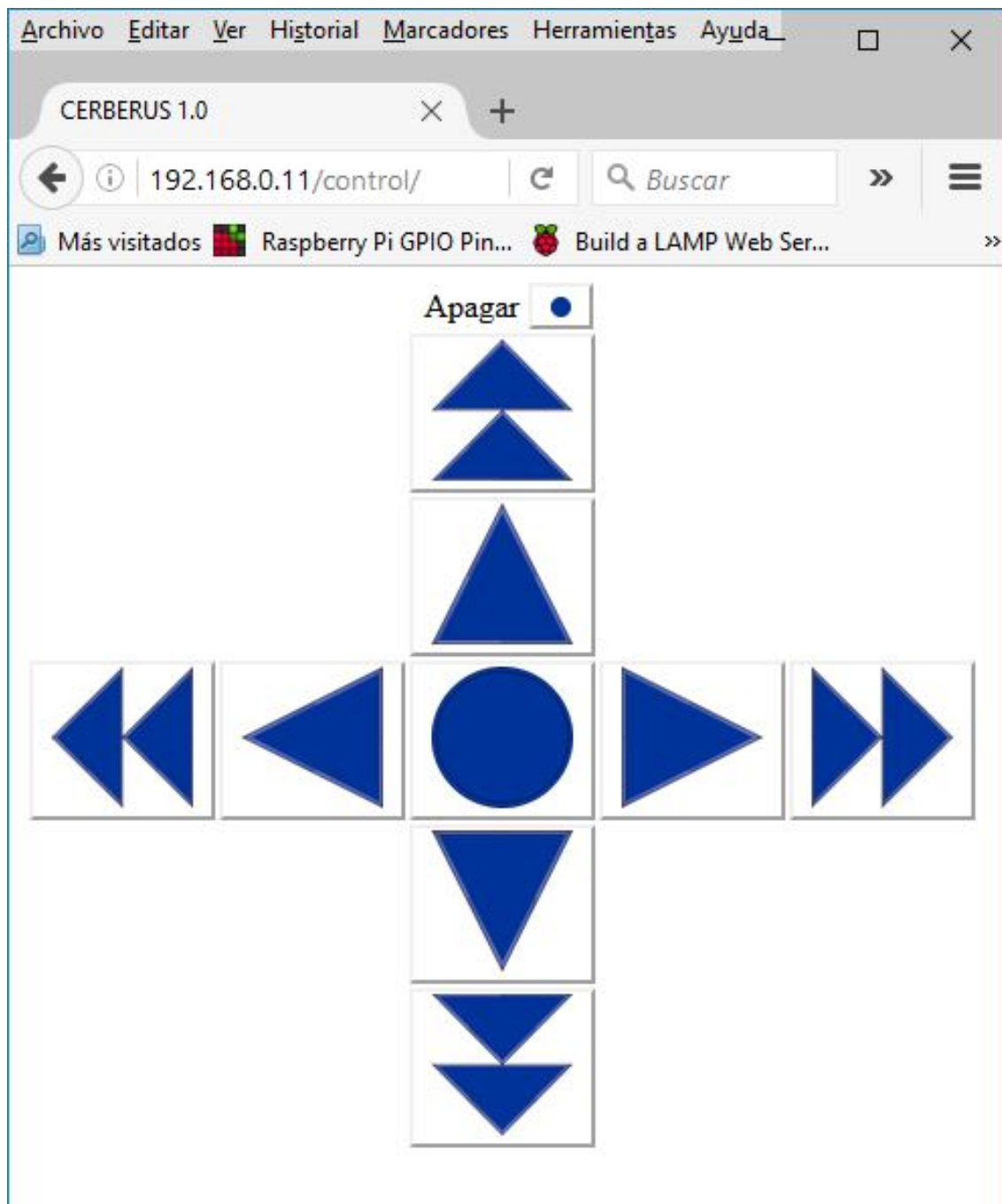


Figura 3.17: Interfaz de usuario [Autor]

3.4 Sistema de comunicación

El sistema de comunicación se desarrolló por medio del modelo TCP/ip donde se hizo uso de sockets, con el fin de usar el robot como servidor y se pudiera acceder a él por medio de reenvío de puertos. Para la configuración del reenvío de puertos, se realizó la configuración del port forwarding del router, dándole la dirección ip 192.168.0.11 estática a la raspberry pi y la dirección ip estática 192.168.0.13 a la cámara ip; donde a cada una se le asignó el puerto 4000 y el puerto 3200 respectivamente como se muestra en la figura 3.10. Para los permisos de ingreso, se configuró la ip 0.0.0.0 que indica que se permitan todas las direcciones

ip de ingreso por el puerto 80 debido a que el usuario puede estar en cualquier parte del mundo y puede tener cualquier ip pública dentro de Internet. Cabe anotar, que las direcciones ip estáticas de la raspberry pi y de la cámara son direcciones ip privadas, es decir, están en la red local y tienen la misma máscara de red y la misma puerta de enlace.

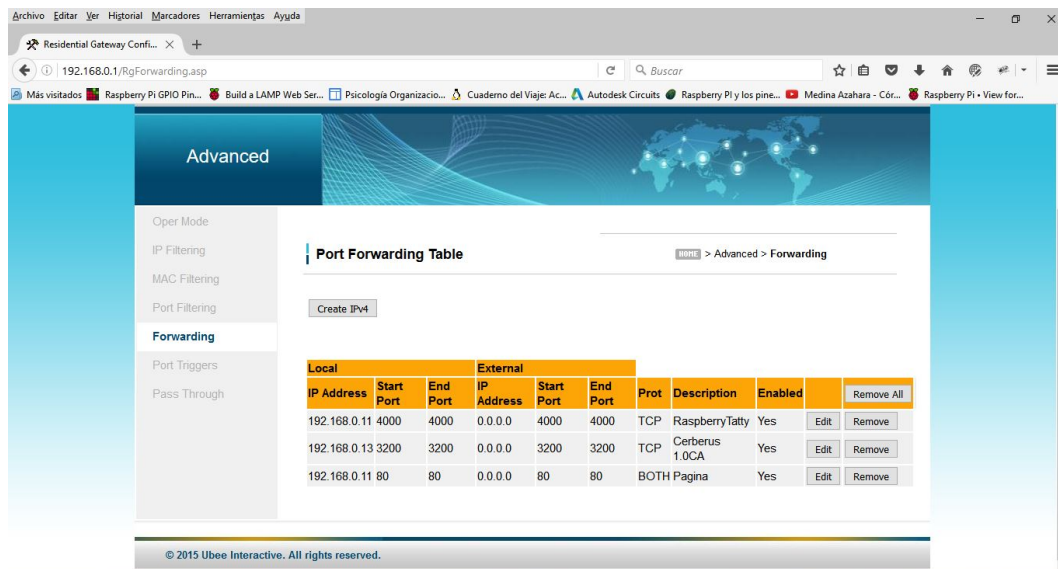


Figura 3.18: Configuraciones del router [Autor]

En la figura 3.19 se puede apreciar al lado izquierdo la red privada en la cual se encuentra Cerberus 1.0 como servidor donde al ser una red doméstica el router DOCSIS 2.0 marca Ubee funciona tanto como router y como switch, une la red local con la red extendida llamada Internet donde éste tiene en la red local la dirección ip de la puerta de enlace y en la red extendida tiene la dirección ip pública que es dinámica y única. Para la conexión externa del usuario hacia el robot la dirección ip pública se debe estar verificando debido a que ésta es cambiada por el operador periódicamente y se puede llegar a errores de conexión por no realizar la comunicación con la ip correcta. Esta verificación se puede realizar a través de un buscador por medio de consultas a páginas como www.cual-es-mi-ip.net.

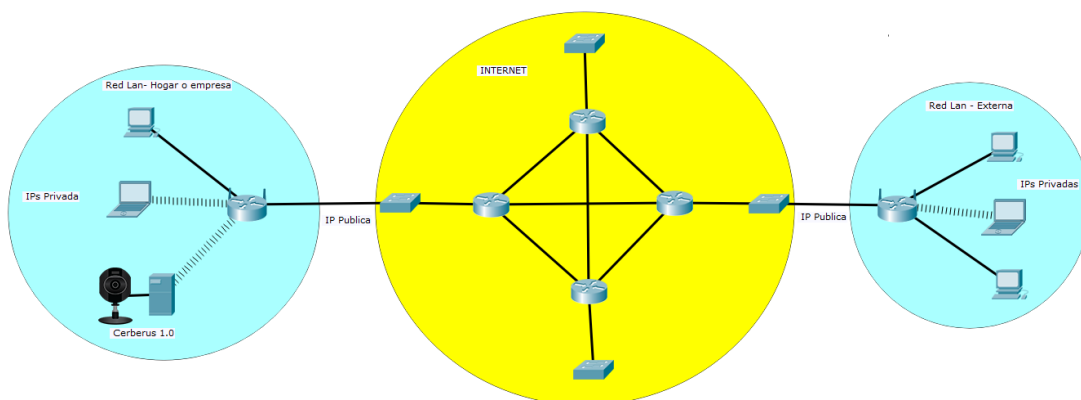


Figura 3.19: Funcionamiento de la red [Autor]

Teniendo la ip pública a la que esta conectado el servidor o robot, se ingresa el socket y el router a partir del reenvío de puertos hace el direccionamiento interno para conectarse al servidor, logrando así la comunicación entre el usuario y el robot por medio de Internet.

3.5 Sistema de alimentación

Para la alimentación se utilizaron tres baterías de litio, dos de 7,4V 2,2Ah 2S para la cámara y para los motores y otra de 11,1V 1,5Ah 3S para la raspberry pi; con el fin de que Cerberus 1.0 tenga una autonomía de aproximadamente dos horas en general según la capacidad de carga de las baterías, donde este tiempo se puede estimar de la siguiente forma:

$$\text{Tiempo de autonomia} = \frac{\text{Capacidad de corriente de la batería}}{\text{Corriente nominal solicitada por la carga}}$$

Dando así los siguientes tiempos para cada una de las baterías:

$$\text{Tiempo de autonomia batería de la cámara} = \frac{2,2Ah}{1A} = 2,2h$$

$$\text{Tiempo de autonomia batería de la raspberry pi} = \frac{2,2Ah}{1A} = 2,2h$$

$$\text{Tiempo de autonomia batería de los motores} = \frac{1500mAh}{300mA} = 5h$$

Se debe tener en cuenta que para el cálculo de la autonomía de los motores se destinó con la carga de free run, es decir, cuando los motores no están con peso por lo tanto, en el momento de realizar las pruebas éste debe descender.

Al ser baterías de litio se debe tener un cuidado especial como se especificó en la subsección 2.2.5.2 por lo tanto se utiliza el cargador Turnigy 4x1000W que se muestra en la figura 3.20, el cual posee un ecualizador que permite cargar las baterías de forma balanceada y celda por celda utilizando la carga de tres etapas. Con este cargador se pueden cargar cuatro baterías al mismo tiempo conectando cada batería a una tarjeta que se tiene por canal y se configura el tipo de carga que se quiere realizar a cada batería.



Figura 3.20: Cargador de baterías Turnigy 4x1000W [Autor]

Antes de realizar la carga de las baterías se debe configurar el tipo de batería que para este caso sería de lipo y dos de 7,4V 2S a 2200mAh y una a 11,1V (3S) a 1500mAh. También se le configura el tipo de carga que se va a realizar, entre estos es recomendable el balanceado y el de almacenamiento para baterías de polímero de litio ya que al ser tan delicadas tiene que mantener una tensión entre el rango establecido de mínimo 3,2V y máximo 4,2V por celda. Después de configurarlo, se conectan las baterías y se le da clic en start, donde el cargador reconoce las baterías y la tensión a la que se encuentra cada celda y el total de las mismas como se muestra en la figura 3.21, de allí inicia el proceso de carga. Al terminar emite un sonido y para el proceso de carga.



Figura 3.21: Reconocimiento de baterías en el cargador [Autor]

Capítulo 4

PRUEBAS Y RESULTADOS

En este capítulo se presentan los resultados de las pruebas realizadas sobre cada uno de los subsistemas de Cerberus 1.0 los cuales se han explicado en las secciones anteriores y el funcionamiento en conjunto de todos los sistemas del prototipo.

4.1 Sistema mecánico

Se construyó la plataforma de manera modular con cortes de acrílicos negro unidos por medio de toques con tornillos M4x12 y M4x8 de tal forma que éste se pueda desarmar de forma fácil como se muestra en la figura 4.1, de la misma forma se puede apreciar que la conexión entre las bases de los motores y las ruedas son fijadas por medio de bujes y para fijar los elementos internamente se utilizó velcro con el fin de poderlos fijar y sacar sin necesidad de desarmar piezas.



Figura 4.1: Chasis Cerberus 1.0 modular [Autor]

Al realizar pruebas se identificó que un motor se desplaza a mayor velocidad que el otro, por lo tanto se etiquetó cada uno con una letra como se observa en la figura 3.3, donde el motor “B” tiene una velocidad de 211 rpm y el motor “A” tiene una velocidad de 228 rpm.

La rueda libre que se encuentra en la parte frontal de Cerberus 1.0 se ubicó de tal forma que quedara balanceado el peso y su centro de gravedad quedara aproximadamente en el centro geométrico del robot.

4.2 Sistema electrónico

4.2.1. Subsistema de sensores

Al realizar las pruebas con la cámara se muestra unos retrasos en la comunicación donde en ocasiones es necesario actualizar la imagen con el fin de reanudar nuevamente la comunicación y envío de datos de vídeo. En la noche los leds infrarrojos de la cámara permiten tener una visión amplia del lugar que se esta vigilando como se observa en la figura 4.2. La grabación en vídeo se realiza en el equipo que se este monitoreando el robot, por lo tanto, la capacidad máxima de grabación depende de la memoria del equipo y de la comunicación. La toma de imágenes así como la grabación también queda en el equipo, de ser necesario que quede montado en el robot se le debe instalar una memoria SD con adaptador en la ranura que posee la cámara.

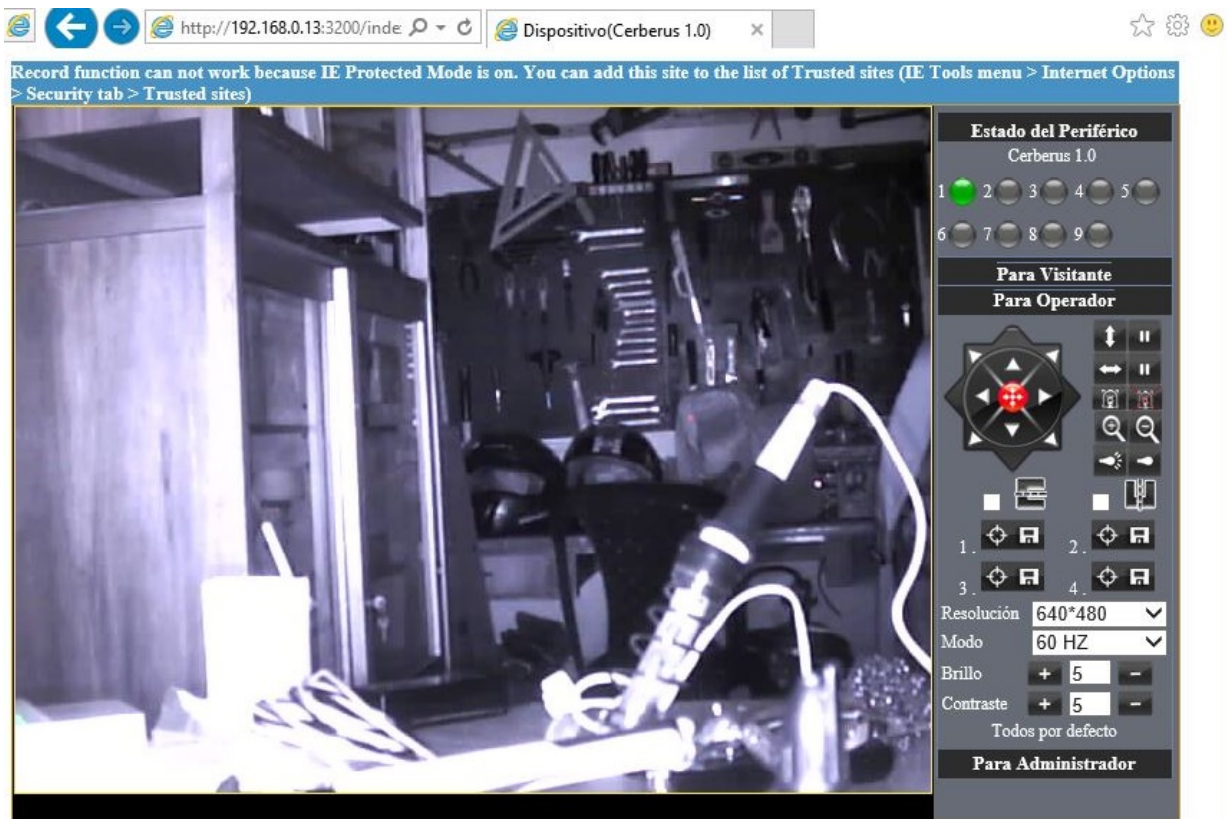


Figura 4.2: Visión nocturna de la cámara [Autor]

4.2.2. Subsistema de conexión de dispositivos

En la tarjeta general se realizó la conexión de las baterías además de la unión entre la tarjeta general y la tarjeta para carga de baterías como se muestra en la figura 4.3, de allí, se realizó la conexión de alimentación de motores, raspberry pi y cámara además de las pruebas de encendido de todos los sistemas. Al dejarse los

cables largos de tal forma que fuera más fácil manipularlos, se amarraron con amarras velcro donde éstas permiten que se desamarran de forma fácil.

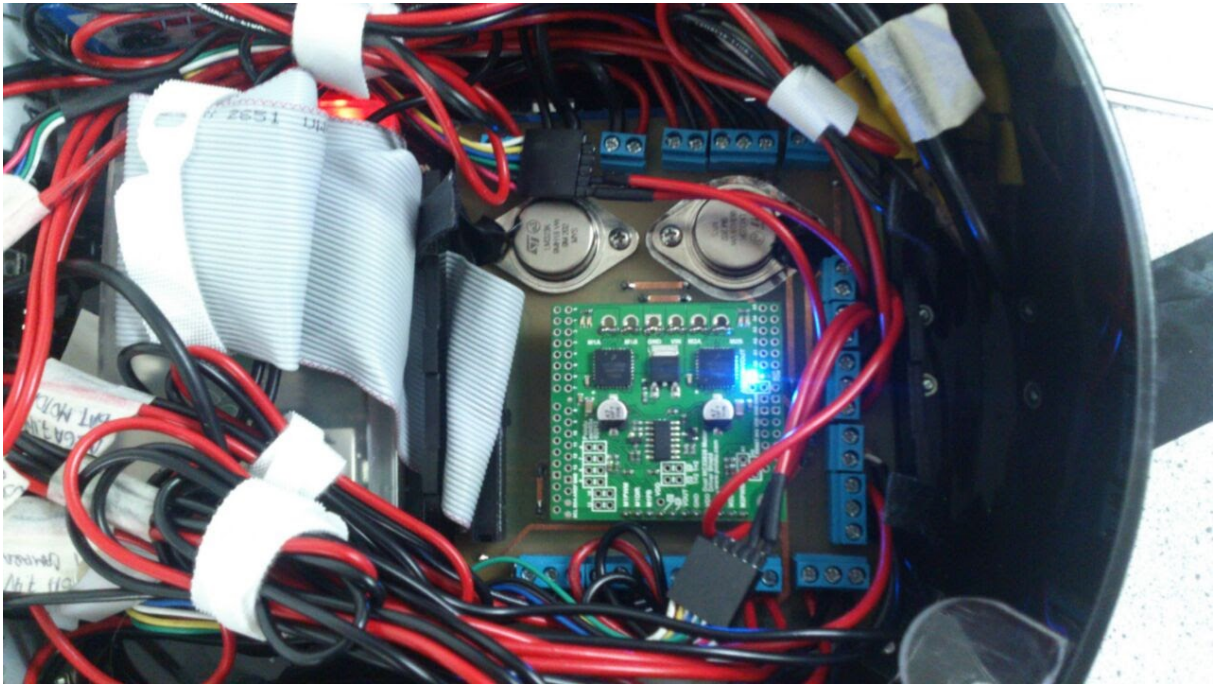


Figura 4.3: Conexión tarjeta general y prueba de reconocimiento de carga [Autor]

La tarjeta general se desarrolló con el fin de unificar de tener una tierra común entre todos los sistemas y no crear diferencia de potencial entre un equipo u otro.

4.3 Sistema de inteligencia o control

4.3.1. Subsistema de control de motores

Con el control de motores se realizó pruebas de control de giro de motores, primero por medio de pulsos lógicos y después con la raspberry pi. El pin D2 del puente H a pesar de estar negado, siempre debe estar en alto para que éste envíe la señal a los motores. En el momento en que se alimenta el puente H se enciende un indicador azul y así mismo cuando se envía un uno lógico para cada sentido de giro se enciende un indicador verde o rojo respectivamente.

4.3.2. Subsistema de control y proceso de información

Con la raspberry pi, se realizó primero una prueba de identificación de pines como se muestra en la figura 4.4, con el fin de determinar los pines que se iban a utilizar para el control de motores.


```

pi@raspberrypi: ~
pi@raspberrypi ~ $ sudo gpio readall
sudo: /etc/sudoers.d is owned by uid 1000, should be 0
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi |   Name | Mode | V | Physical | V | Mode | Name   | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|     |     | 3.3v   |      |   | 1  | 2  |     | 5v     |     |     | |
|  2  |  8  | SDA.1  | IN   | 1 | 3  | 4  |     | 5V     |     |     |
|  3  |  9  | SCL.1  | IN   | 1 | 5  | 6  |     | 0v     |     |     |
|  4  |  7  | GPIO. 7 | IN   | 1 | 7  | 8  |  1  | ALTO   | TxD  | 15  | 14  |
|     |     | 0v     |      |   | 9  | 10 |  1  | ALTO   | RxD  | 16  | 15  |
| 17  |  0  | GPIO. 0 | IN   | 0 | 11 | 12 |  0  | IN     | GPIO. 1 | 1  | 18  |
| 27  |  2  | GPIO. 2 | OUT  | 1 | 13 | 14 |     | 0v     |     |     |
| 22  |  3  | GPIO. 3 | IN   | 0 | 15 | 16 |  0  | IN     | GPIO. 4 | 4  | 23  |
|     |     | 3.3v   |      |   | 17 | 18 |  0  | IN     | GPIO. 5 | 5  | 24  |
| 10  | 12  | MOSI   | IN   | 0 | 19 | 20 |     | 0v     |     |     |
|  9  | 13  | MISO   | IN   | 0 | 21 | 22 |  0  | IN     | GPIO. 6 | 6  | 25  |
| 11  | 14  | SCLK   | IN   | 0 | 23 | 24 |  1  | IN     | CE0   | 10  |  8  |
|     |     | 0v     |      |   | 25 | 26 |  1  | IN     | CE1   | 11  |  7  |
|  0  | 30  | SDA.0  | IN   | 1 | 27 | 28 |  1  | IN     | SCL.0 | 31  |  1  |
|  5  | 21  | GPIO.21 | IN   | 0 | 29 | 30 |     | 0v     |     |     |
|  6  | 22  | GPIO.22 | OUT  | 1 | 31 | 32 |  0  | IN     | GPIO.26 | 26  | 12  |
| 13  | 23  | GPIO.23 | OUT  | 1 | 33 | 34 |     | 0v     |     |     |
| 19  | 24  | GPIO.24 | IN   | 0 | 35 | 36 |  0  | IN     | GPIO.27 | 27  | 16  |
| 26  | 25  | GPIO.25 | IN   | 0 | 37 | 38 |  0  | IN     | GPIO.28 | 28  | 20  |
|     |     | 0v     |      |   | 39 | 40 |  0  | IN     | GPIO.29 | 29  | 21  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi |   Name | Mode | V | Physical | V | Mode | Name   | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figura 4.4: Identificación de GPIO [Autor]

De allí, se procedió a realizar el programa mostrado en la figura 3.16 para cada giro y el de la página mostrado en la figura 3.17, después de esto, se realizaron pruebas de ejecución por consola y desde la página realizando la conexión por red LAN. Como un motor tenía una mayor velocidad que el otro como se especificó en la sección 4.1, se cambiaron los ciclos a 26 % para velocidad normal y 76 % para velocidad rápida para el motor “B” y 25 % y 75 % para el motor “A” respectivamente.

En el momento de apagar la raspberry pi a través de la página, se pierde conexión por lo tanto hasta que se vuelva a encender de forma física a través del suiche de llave no se podrá realizar la conexión.

4.4 Sistema de comunicación

Para realizar la comunicación entre el computador de monitoreo y el robot se abren dos páginas del navegador, donde en una se ingresa a la cámara y en la otra se ingresa a la página de movimiento del robot como se muestra en la figura 4.5.

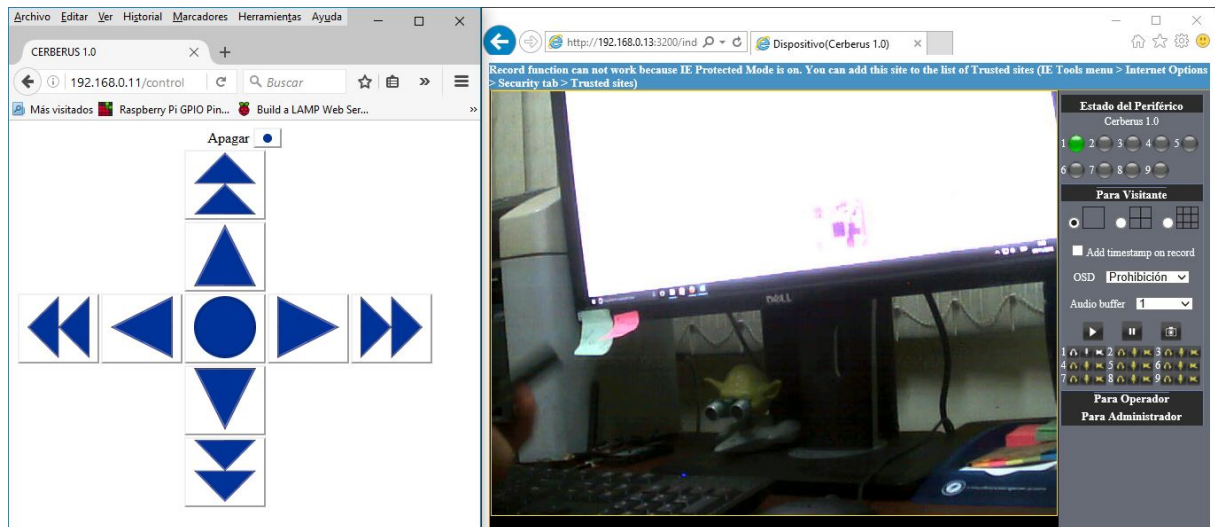


Figura 4.5: Escritorio de monitoreo [Autor]

Primero se realiza pruebas en red local ingresando por medio del navegador con la dirección ip privada de cada equipo, es decir, 192.168.0.13:3200 para la cámara y 192.168.0.11 para la raspberry pi, en la figura 4.6 se puede observar que el robot se encuentra en la misma área donde se encuentra el equipo de monitoreo.

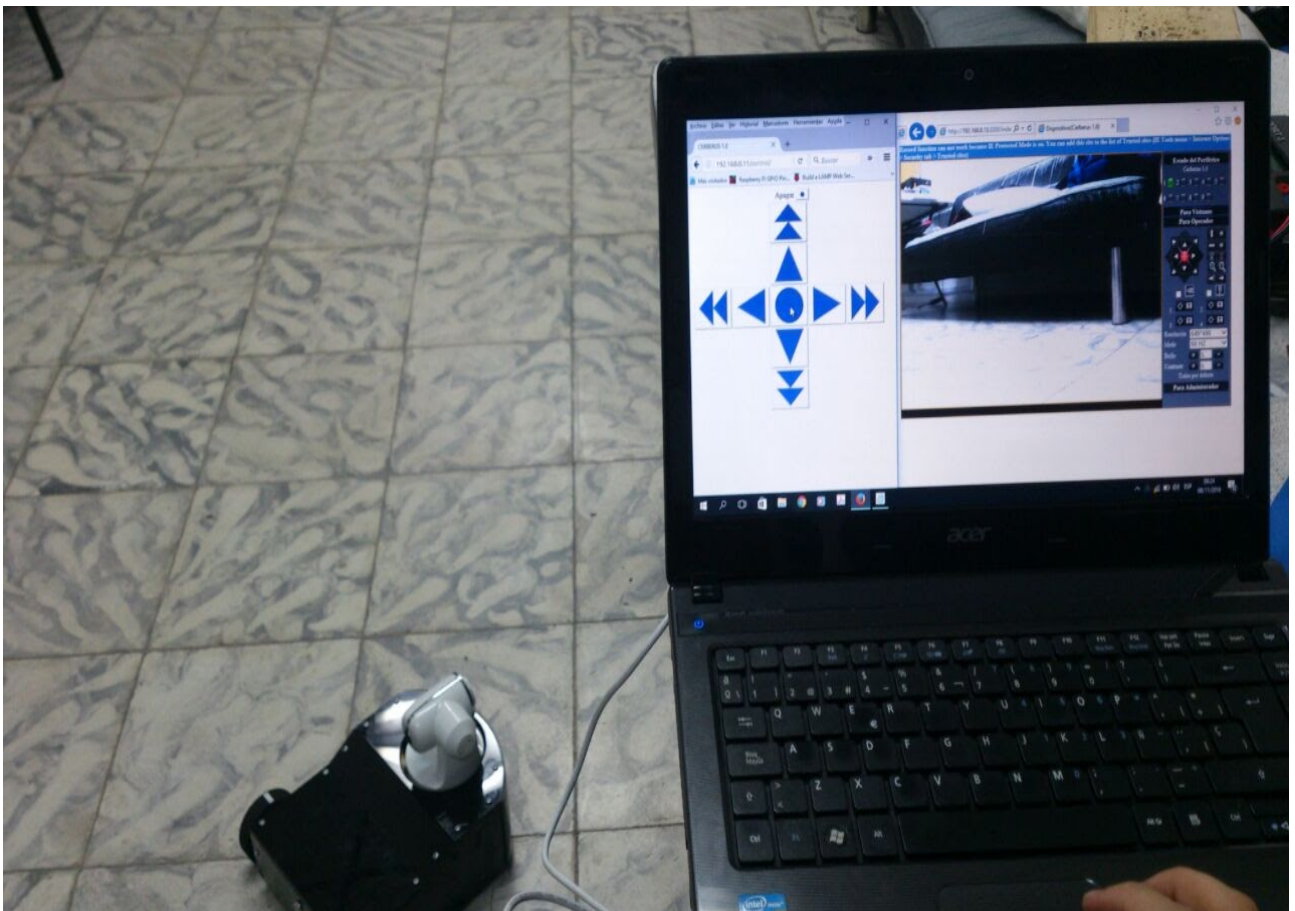


Figura 4.6: Prueba en área local [Autor]

De allí se realizaron pruebas de red de área extendida a partir de un equipo con datos y el ingreso por la dirección pública de la red donde estaba ubicado el robot. Para ello primero se verificó la dirección ip como se muestra en la figura 4.7 y se procedió a ingresar desde el equipo con ip pública 201.236.221.23.

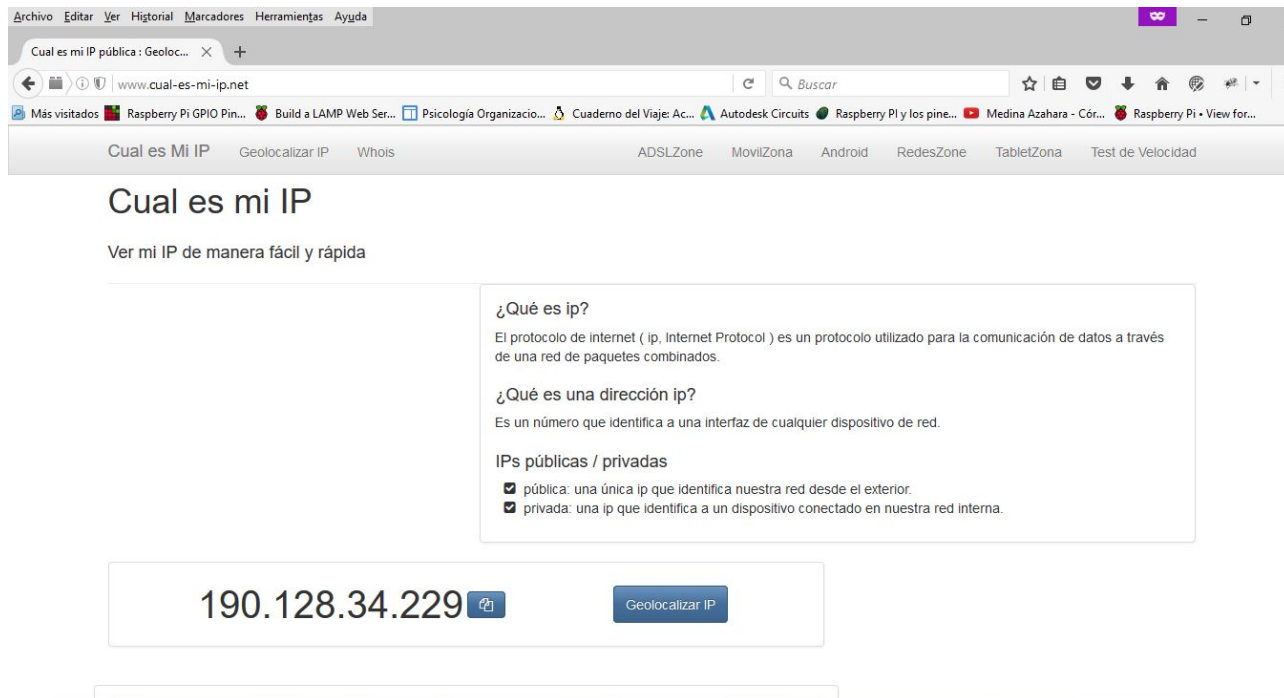


Figura 4.7: Verificación de ip publica [Autor]

Luego se realizaron pruebas ingresando desde la red de la Universidad Tecnológica de Pereira hacia servidor del robot y se observó que se tiene un retraso de tiempo debido al tráfico de la red.

4.5 Sistema de alimentación

Se cargaron las baterías a partir del cargador turnigy como se describe en la sección anterior, se pudo observar que a medida que la batería se va cargando, el cargador sigue censando por cada celda la carga y si hay una de las celdas de la batería dañada, el cargador no la reconoce y muestra un error en la pantalla.

La carga de las baterías es de aproximadamente una (1) hora, sin embargo al estar algunas celdas más descargadas que otras el cargador se demorará más al tratar de balancearlas y de igual forma se debe estar verificando la carga con la finalidad de desconectarlas en el momento en que se carguen totalmente sin pasar el nivel máximo de tensión.

Cuando se deje mucho tiempo el robot sin utilizar o cargar (más de un día), se debe destapar y desconectar las baterías, ya que al ser tan delicadas, pueden sufrir una descarga profunda y dañarse las celdas de esta.

4.6 Pruebas generales de todo el sistema

Ya realizadas todas las pruebas a cada sistema por aparte, se comenzó a unir y a realizar pruebas en conjunto; se inicio con conectar toda la alimentación y energizar la cámara, la raspberry y el puente H. De allí se procedió a realizar pruebas de movimiento solo con el sistema mecánico y el sistema de control, teniendo en cuenta el peso total del robot y observando su velocidad, sin embargo, al no ser tan alta la variación, se dejó el mismo ciclo útil configurado en la sección 4.3.2.

Luego se inicio con pruebas de monitoreo por medio de la cámara y el movimiento del robot, verificando así la conexión y el tiempo de retardo donde se observó que al no ser una red dedicada se tiene retraso tanto en el movimiento como en el vídeo de la cámara. Por último, se realizaron pruebas de autonomía del robot, la cual en trabajo continuo es de aproximadamente dos horas y media (2.30h) y se realizó pruebas de carga sin desmontar las baterías del prototipo como se muestra en la figura 4.8 donde al estar conectado al robot, tiene una descarga continua de 1mAh, por lo tanto, la carga de las baterías se realiza en aproximadamente dos horas (2h).

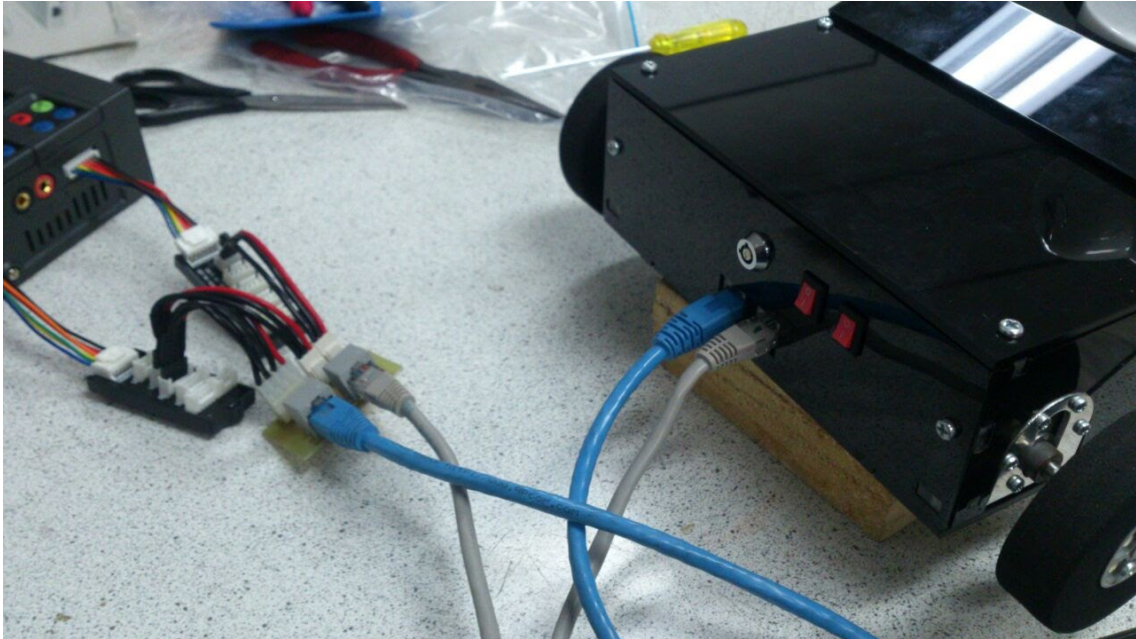


Figura 4.8: Carga de Baterías dentro del prototipo [Autor]

Capítulo 5

CONCLUSIONES Y FUTUROS TRABAJOS

5.1 CONCLUSIONES

La raspberry pi es un computador de placa reducida que permite realizar diferentes aplicaciones como la conducción, operación y monitoreo de un robot móvil remotamente por medio de Internet en cualquier parte del mundo, sin embargo, al tener un sistema operativo hace que no se pueda dedicar solo a desarrollar una tarea sino que ejecute varias al tiempo, donde esto afecta en gran medida una aplicación que requiera precisión debido a los retardos de tiempo.

Con la raspberry pi se debe tener cuidado cuando se realizan las conexiones dado que no posee protección alguna y fácilmente puede ser dañada. Así mismo, al procesador se le debe poner un disipador térmico debido a que al ejecutar varios procesos su temperatura se eleva considerablemente y esto hace que se queme.

Cerberus 1.0 es un prototipo robótico que se puede usar para la inspección de seguridad en áreas que poseen red inalámbrica dando la posibilidad de monitorearlo desde diferentes partes del mundo. De igual forma se observa que en una segunda etapa el sistema mecánico se debe cambiar para tener una cobertura en entornos más amplios con superficies no uniforme.

La robótica móvil día a día esta avanzando y se relaciona más con la inteligencia y visión artificial, donde una de las aplicaciones que se lleva a cabo son automóviles capaces de tomar decisiones para desplazarse sin tener supervisión humana.

Todos los sistemas modulares hacen que el mantenimiento y mejora sea más práctico y menos costoso, por ello la estructura modular de Cerberus 1.0 hace que se pueda mejorar de manera continua sin elevar los costos y poder utilizar los diferentes elementos que tiene en las mejoras realizadas.

Se pudo observar en la comunicación que para garantizar los tiempos de ejecución en tiempo real se tendría que tener un dispositivo de control dedicado a los PWM enviados a los motores para el movimiento del robot y un sistema de sensores que apoyen a la cámara cuando se tienen retrasos en el envío de imagen para evitar que Cerberus 1.0 pueda colisionar.

En la alimentación del robot se pudo observar que las baterías de polímero de litio son demasiado delicadas, por lo tanto es mejor utilizar otro tipo de baterías como las celdas de litio-ion o níquel hidruro metálico debido a que las baterías de lipo al tener una de las celdas desbalanceada, se debe realizar una descarga y carga con un cargador especial que tenga un balanceador con el fin de que no se vaya a dañar las otras celdas mientras la que esta más descargada se cargue. Así mismo, estas baterías siempre que se carguen deben ser supervisadas debido a que al presentar un error pueden comenzar a inflarse hasta el punto de explotar y

ocasionar daños.

5.2 FUTUROS TRABAJOS

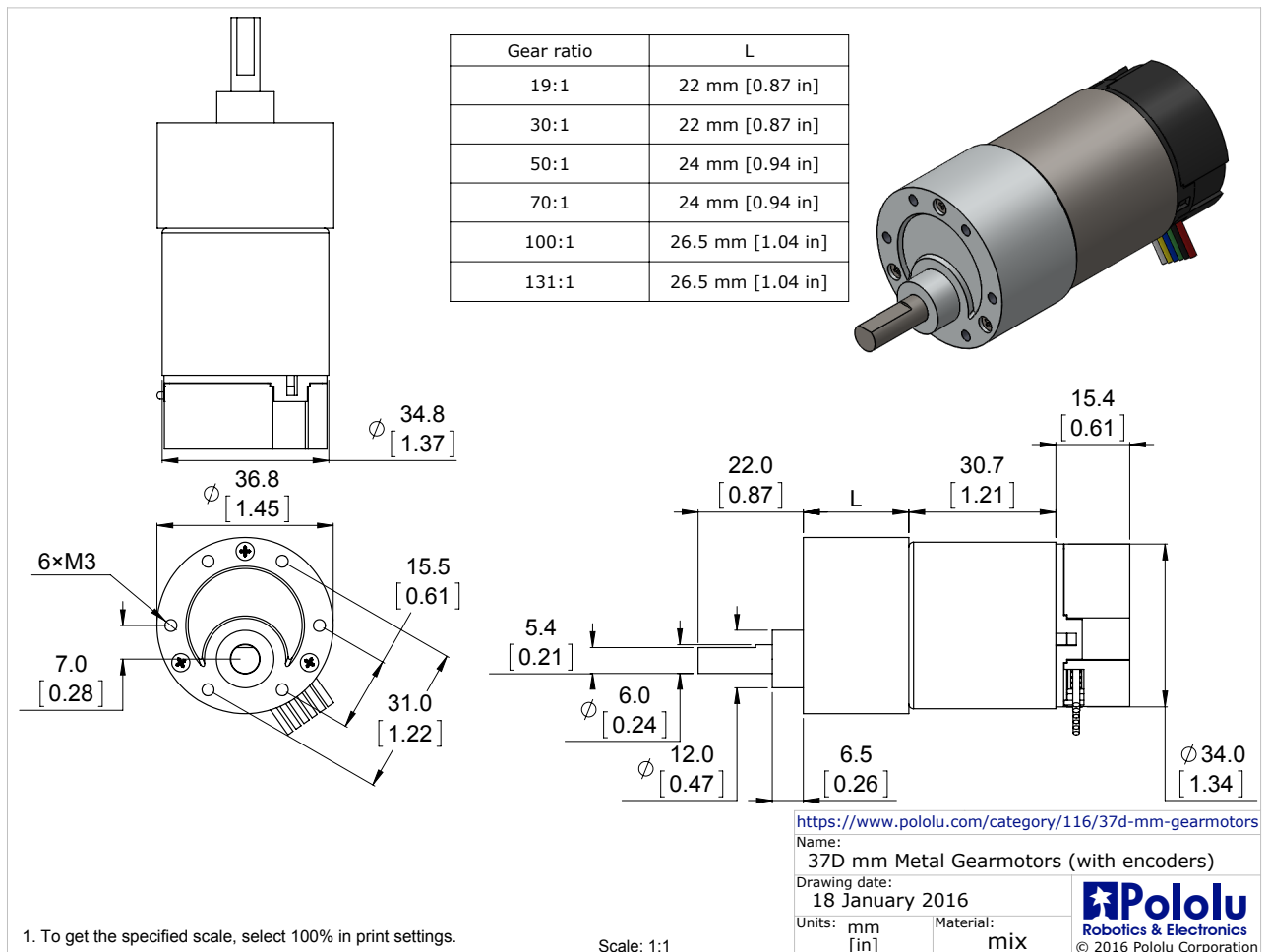
Para futuros trabajos que se pueden realizar con base en el desarrollo proyecto se encuentra:

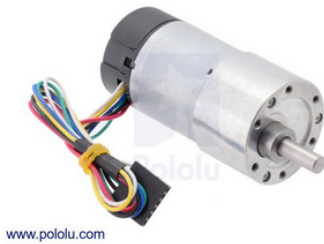
1. Mejorar la plataforma de usuario donde se pueda observar la cámara y el movimiento en una sola pagina
2. Mejorar el sistema mecánico de tal forma que pueda moverse por estructuras no uniformes
3. Incluir inteligencia artificial donde se le pueda dar una ruta de vigilancia al robot, de tal forma que realice la ronda de manera autónoma
4. Implementar un sistema de visión artificial con reconocimiento de rostros y bases de datos de personal autorizado
5. Implementación de optimizan de consumo de energético

Capítulo 6

Anexos

6.1 Anexo Datos técnicos de Motores



50:1 Metal Gearmotor 37Dx70L mm with 64 CPR Encoder

www.pololu.com

Details for item #2824

$$\text{Exact gear ratio: } \frac{25 \times 30 \times 25 \times 40}{10 \times 10 \times 15 \times 10} = 50:1$$

Size:	37D x 70L mm ¹
Weight:	225 g
Shaft diameter:	6 mm

General specifications

Gear ratio:	50:1
Free-run speed @ 6V:	100 rpm ²
Free-run current @ 6V:	250 mA ²
Stall current @ 6V:	2500 mA ²
Stall torque @ 6V:	85 oz-in ²
Free-run speed @ 12V:	200 rpm
Free-run current @ 12V:	300 mA
Stall current @ 12V:	5000 mA
Stall torque @ 12V:	170 oz-in
Lead length:	11 in

This gearmotor is a powerful 12V brushed DC motor with a **50:1** metal gearbox and an integrated quadrature encoder that provides a resolution of 64 counts per revolution of the motor shaft, which corresponds to **3200 counts per revolution** of the gearbox's output shaft. These units have a 0.61"-long, 6 mm-diameter D-shaped output shaft. This gearmotor is also available [without an encoder](#).

Key specs at **12 V**: 200 RPM and 300 mA free-run, 170 oz-in (12 kg-cm) and 5 A stall.

These motors are intended for use at 12 V, though in general, these kinds of motors can run at voltages above and below the nominal voltage (they can begin rotating at voltages as low as 1 V). Lower voltages might not be practical, and higher voltages could start negatively affecting the life of the motor.



These gearmotors are functionally identical to the previous versions we carried [without end caps](#) (they use the same motor, encoder, and gearboxes). The black plastic end cap is easily removable if you need to access the encoder or want to slightly reduce the overall gearmotor size, but there is a little bit of base plastic that will remain, as shown in the pictures below:



Pololu

6.2 Anexo Datos técnicos de la Cámara IP

ESPECIFICACIONES TÉCNICAS

ITEM		IP CAM
Image Sensor	Image sensor	1/5" Color CMOS Sensor
	Display	640 x 480 (300K Pixels)
	Lens	f:3.6mm, F:2.4 (IR Lens)
	Mini illumination	0.5 Lux
Lens	Lens Type	4-9MM ZOOM Lens
	Viewing Angle	60 Degree
Audio	Microphone	Audio of two way
Domain name	Server	MSN server/ DDNS server
Video	Image Compression	M-JPEG
	Image Frame Rate	15 FPS(VGA), 30FPS(QVGA)
	Resolution	640 x 480(VGA), 320 x 240(QVGA)
	Flip Mirror Images mode	rRical / Horizontal
	Video Parameters	Brightness, Contrast
	Communication	Ethernet
Supported Protocol		HTTP/DHCP/IP/TCP/UDP/FTP/SMTP/DDNS/PPPoE/UPnP
Mobilephone monitor		Support iPhone/iPad/3G phone/Android smartphone
Wireless Standard		IEEE 802.11b/g
Wireless Standard		802.11b:11Mbps(max.) 802.11g:54Mbps(max.)
Wireless Security		64/128-bitWEP Encryption
Physical	Pan/Tilt Angle	Horizontal: 330° & Vertical: 90°
	Infrared Light	12 IR LEDs, Night visibility up to 15 Mete
	Product size	100(L) x 100(W) x 125mm(H)
	Gross Weight	590g/pcs (packing size: 196x165x123mm)
	Net Weight	348g/pcs (only product)
Power	Specification	DC 5V/2.5A 1.5 meter
	Power Consumption	5 Watts

Environment	Operate Temper	0°~ 55°C (14°F~122°F)
	Operating Humidity	20% ~ 85% non-condensing
	Storage Temper	-10°C ~ 60° (14°F ~ 140°F)
	Storage Humidity	0% ~ 90% non-condensing
PC Requirements	CPU	2.0GHZ or above (suggested 3.0GHz)
	Memory Size	512MB or above (suggested 1.0GHz)
	Display card	64M or above
	Supported OS	Microsoft Windows XP/Vista/Windows7
Certification	Browser	IE 6.0, IE7.0 IE8.0 firefox2.0 firefox 3.0 .Google CE,FCC,RoHS

6.3 Anexo datos técnicos Regulador de Tensión LM323k



LM123/LM223
LM323

THREE-TERMINAL 3A-5V POSITIVE VOLTAGE REGULATORS

- OUTPUT CURRENT : 3A
- INTERNAL CURRENT AND THERMAL LIMITING
- TYPICAL OUTPUT IMPEDANCE : 0.01Ω
- MINIMUM INPUT VOLTAGE : 7.5V
- POWER DISSIPATION : 30W

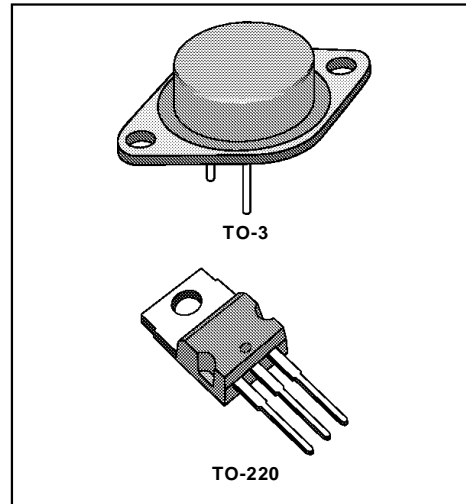
DESCRIPTION

The LM123, LM223, LM323 are three-terminal positive voltage regulators with a preset 5V output and a load driving capability of 3A. New circuit design and processing techniques are used to provide the high output current without sacrificing the regulation characteristics of lower current devices.

The 3A regulator is virtually blowout proof.

Current limiting, power limiting and thermal shutdown provide the same high level of reliability obtained with these techniques in the LM209, 1A regulator.

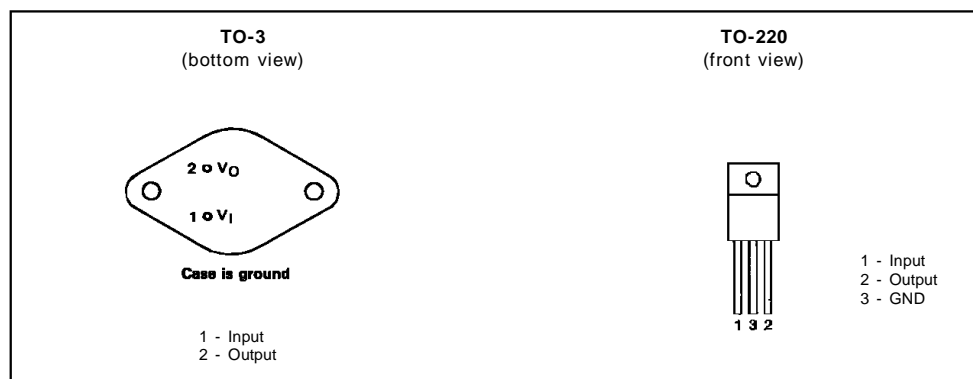
An overall worst case specification for the combined effects of input voltage, load current, ambient temperature, and power dissipation ensure that the LM123, LM223, LM323 will perform satisfactorily as a system element.



ORDER CODES

Part Number	Temperature Range	Package	
		K	T
LM123	-55 °C to 150 °C	•	
LM223	-25 °C to 150 °C	•	
LM323	0 °C to 125 °C	•	•

PIN CONNECTION



LM123-LM223-LM323

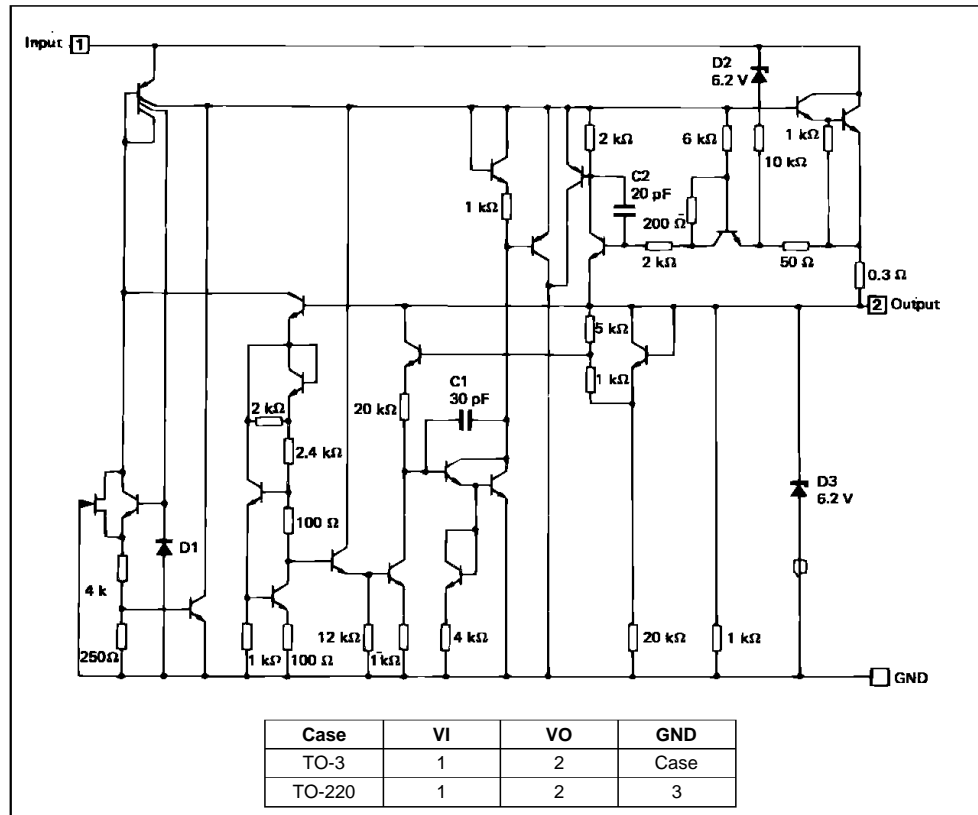
ABSOLUTE MAXIMUM RATING

Symbol	Parameter	Value	Unit
V_I	Input Voltage	20	V
I_O	Output Current	Internally Limited	
P_{tot}	Power Dissipation	Internally Limited	
T_{oper}	Operating Junction Temperature Range	LM123	-55 to 150
		LM223	-25 to 150
		LM323	0 to 125
T_{stg}	Storage Temperature Range	-65 to 150	°C

THERMAL CHARACTERISTICS

Symbol	Parameter	Typ.	Max.	Unit
$R_{thj-case}$	Junction-case Thermal Resistance	TO-3	2	°C/W
		TO-220	3	
$R_{thj-amb}$	Junction-ambient Thermal Resistance	TO-3	35	°C/W
		TO-220	50	

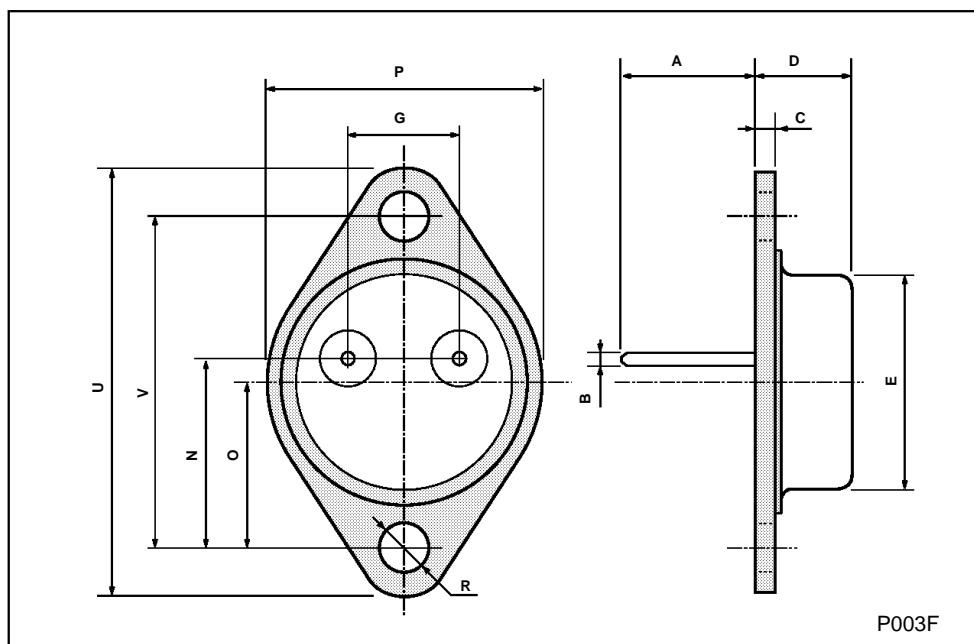
SCHEMATIC DIAGRAM



LM123-LM223-LM323

TO-3 MECHANICAL DATA

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A	11.00		13.10	0.433		0.516
B	0.97		1.15	0.038		0.045
C	1.50		1.65	0.059		0.065
D	8.32		8.92	0.327		0.351
E	19.00		20.00	0.748		0.787
G	10.70		11.10	0.421		0.437
N	16.50		17.20	0.649		0.677
P	25.00		26.00	0.984		1.023
R	4.00		4.09	0.157		0.161
U	38.50		39.30	1.515		1.547
V	30.00		30.30	1.187		1.193



6.4 Anexo Programa página de interfaz de usuario

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="UTF-8" />
5 <title>CERBERUS 1.0</title>
6 </head>
7 <body>
8 <form action="" method="post" id="formulario">
9 <div align=center>Apagar <button type="submit" name="apagar" value="apagar" id
    ="apagar" style="background-color:transparent" onclick="apagar">
10  </button></div/>
11 <section>
12 <div align="left">
13 <table border="0" align="left" cellpadding="0">
14 <tr>
15 <td width="70"><div align="left"></div></td>
16 <td width="70"><div align="left"></div></td>
17 <td width="70"><div align="left">
18 <button type="submit" name="adelanterapido" value="adelanterapido" id="
    adelanterapido" style="background-color:transparent">
19  </button></div
    ></td>
20 </tr>
21 <tr>
22 <td width="70"><div align=left></div></td>
23 <td width="70"><div align=left></div></td>
24 <td width="70"><div align="left">
25 <button type="submit" name="adelante" value="adelante" id="adelante" style="
    background-color:transparent">
26  </button></div></td>
27 </tr>
28 <tr>
29 <td width="70"><div align="left">
30 <button type="submit" name="izquierdarapido" id="izquierdarapido" value
    ="izquierdarapido" style="background-color:transparent">
31  </button
    > </div> </td>
32 <td width="70"><div align="left">
33 <button type="submit" name="izquierda" id="izquierda" value="izquierda"
    style="background-color:transparent">
34  </button> </
    div> </td>
35 <td width="70"><div align="left">
36 <button type="submit" name="parar" value="parar" id="parar" style="
    background-color:transparent">
37  </button> </div>
    </td>
38 <button type="submit" name="derecha" value="derecha" id="derecha" style
    ="background-color:transparent">
39  </button> </div>
    </td>

```



```

40 <td width="70"><div align="left">
41   <button type="submit" name="derecharapido" value="derecharapido" id="
      derecharapido" style="background-color:transparent">
42    </button>
      </div> </td>
43 </tr>
44 <tr>
45 <td width="70"><div align="left"></div></td>
46 <td width="70"><div align="left"></div></td>
47 <td width="70"><div align="left">
48   <button type="submit" name="atras" value="atras" id="atras" style="
      background-color:transparent">
49    </button> </div>
      </td>
50 </tr>
51 <tr>
52 <td width="70"><div align="left"></div></td>
53 <td width="70"><div align="left"></div></td>
54 <td width="70"><div align="left">
55   <button type="submit" name="atrasrapido" value="atrasrapido" id="
      atrasrapido" style="background-color:transparent">
56    </button> </
      div> </td>
57 </tr>
58 </table>
59 </div>
60 </section>
61 </form>
62 </body>
63 </html>
64
65 <?php
66
67   if ($_POST[adelanterapido]) {
68     $a- exec("sudo python /var/www/control/py/adelanterapido.py");
69     echo $a;
70   }
71   if ($_POST[adelante]) {
72     $a- exec("sudo python /var/www/control/py/adelante.py");
73     echo $a;
74   }
75   if ($_POST[izquierdarapido]) {
76     $a- exec("sudo python /var/www/control/py/izquierdarapido.py");
77     echo $a;
78   }
79   if ($_POST[izquierda]) {
80     $a- exec("sudo python /var/www/control/py/izquierda.py");
81     echo $a;
82   }
83   if ($_POST[derecharapido]) {
84     $a- exec("sudo python /var/www/control/py/derecharapido.py");
85     echo $a;
86   }
87   if ($_POST[derecha]) {

```

```
88     $a- exec("sudo_python_/var/www/control/py/derecha.py");
89     echo $a;
90 }
91 if ($_POST[atrasrapido]) {
92     $a- exec("sudo_python_/var/www/control/py/atrasrapido.py");
93     echo $a;
94 }
95 if ($_POST[atras]) {
96     $a- exec("sudo_python_/var/www/control/py/atras.py");
97     echo $a;
98 }
99 if ($_POST[parar]) {
100     $a- exec("sudo_python_/var/www/control/py/parar.py");
101     echo $a;
102 }
103 if ($_POST[apagar]) {
104     $a- exec("sudo_python_/var/www/control/py/apagar.py");
105     echo $a;
106     $a- exec("limpiar_cadena");
107     self.close();
108 }
109 ?>
```


Bibliografía

- [1] S. de Vigilancia y Seguridad Privada, “Protocolo de operación servicios de vigilancia electrónica.” <http://www.supervigilancia.gov.co/index.php?idcategoria=57430>, 2013. visto:17-08-2016. 1.2
- [2] A. Barrientos and A. Barrientos, “Fundamentos de robótica,” tech. rep., e-libro, Corp., 2007. 2.1
- [3] P. Atelin and J. Dordogne, *Redes informáticas: conceptos fundamentales: normas, arquitectura, modelo OSI, TCP/IP, Ethernet, Wi-Fi...* Ediciones ENI, 2006. 2.1
- [4] E. García Mancebo *et al.*, “Análisis de prestaciones de sistemas de comunicaciones inalámbricas en aplicaciones de robótica: sistema software,” 2009. 2.1
- [5] RENA, Red Escolar Nacional, Ministerio del Poder Popular para la educación universitaria ciencia y tecnología, Gobierno Bolivariano de Venezuela, “Lenguajes de programación.” <http://www.rena.edu.ve/cuartaEtapa/Informatica/Tema13.html>, 2008. Visto: 08-07-2016. 2.1
- [6] S. K. Saha, *Introducción a la Robótica*. McGraw-Hill Interamericana, 2000. 2.2
- [7] G. Bermudez, “Robots móviles. teoría, aplicaciones y experiencias,” *Revista Tecnura*, vol. 5, no. 10, pp. 6–17, 2002. 2.2.1
- [8] D. Peláez Panesso *et al.*, “Diseño de un prototipo que permita realizar la inspección del interior de tuberías de acueducto por medio de un sistema teleoperado,” 2015. 2.2.2.1
- [9] J. C. López Rodríguez, “Estructura, funcionamiento y aplicación de las cámaras ip,” 2007. 2.2.2.1
- [10] E. ALCALDE LANCHORRO, *Introducción a la Teleinformática*. 1993. 2.2.4
- [11] C. B. Group, “Tcp/ip.” <http://es.ccm.net/contents/282-tcp-ip>, Julio 2016. Visto: 23-09-2016. 2.2.4.1
- [12] Pedro Lagos, INGENIEROS SIDEI LTDA, “Conceptos básicos de carga de baterías y algoritmos de carga.” www.sidei.cl. Visto 05-10-2016. 2.2.5.2
- [13] M. Á. Fernández Lancha, D. Fernández Sanz, and C. Valmaseda Plasencia, “Planificación de trayectorias para un robot móvil,” 2010. 2.3
- [14] P. R. S. TMSUK, “Robots.” <http://www.tmsuk.co.jp/english/robots.html>. Visto: 28 de marzo de 2013. 2.3
- [15] A. Robotic, “Aquiles protector.” <http://www.aquilesrobotic.com/robotica/aquiles-protector/>. Visto: Marzo 28 de 2013. 2.3
- [16] Mostitech, “Mostitech.” <http://www.mostitech.com/home/product/product.jsp>. Visto: Marzo 28 de 2013. 2.3
- [17] Á. Bedoya, J. Chávez, and G. Guzmán, “Propuestas de desarrollo robótico para el desminado humanitario,” *Scientia et Technica*, pp. 239–244, 2011. 2.3