



**Universidad
Andrés Bello®**

UNIVERSIDAD ANDRÉS BELLO

Facultad de Ingeniería

Escuela de Industrias

**“SISTEMA DE ASISTENCIA EN RESCATE MEDIANTE
ROBOT MÓVIL”**

Tesis de pregrado para optar al título de Ingeniero en Automatización y Robótica

AUTOR

ARIEL MAURICIO BASTÍAS MORAGA

PROFESOR GUIA

NÉSTOR PALOMINOS GONZALEZ

SANTIAGO – CHILE

2017



**Universidad
Andrés Bello®**

UNIVERSIDAD ANDRES BELLO
FACULTAD DE INGENIERÍA
ESCUELA DE INDUSTRIAS

INGENIERÍA EN AUTOMATIZACIÓN Y ROBÓTICA

DECLARACIÓN DE ORIGINALIDAD Y PROPIEDAD

Yo, **Ariel Mauricio Bastías Moraga**, declaro que este documento no incorpora material de otros autores sin identificar debidamente la fuente.

Santiago, diciembre 2017

Firma del alumno

Dedicado a todas las personas que han sido víctimas de alguna catástrofe natural, que haya presenciado devastación, causada de forma natural o por otro motivo.

Dedico especialmente también, a las personas que ejercen alguna labor del área del rescate, a quienes ponen en riesgo su vida para socorrer.

A mis compañeros, profesores de la universidad, amigos, hermanos, parientes y evidentemente a mis padres, Paola Moraga y Mauricio Bastias por darme esta oportunidad.

Ariel Mauricio Bastias Moraga.

AGRADECIMIENTOS

Agradezco a mis padres, por la ayuda incondicional, por la confianza que depositaron, para la realización de este proyecto y enseñarme que el conocimiento no tiene fronteras, que el éxito ocurre cuando los sueños son más grandes que las excusas.

También agradezco a quienes han creído en mí y en el desarrollo de este proyecto, a quienes de alguna manera participaron de forma directa o indirecta apoyando en el sustento económico de este, aquellos que también aportaron con críticas positivas y sugerencias, gracias amigos y familiares.

Agradecimiento a mis compañeros de carrera, a mi generación y a los que vi integrarse en transcurso de los años, en especial a mis compañeros que están en el mismo proceso de tesis y aquellos que me corrigieron en el tiempo.

Especialmente Sergio Valenzuela, con quien tuve la oportunidad de compartir el desarrollo de muchos proyectos, como lo fue la participación en la competencia LAUREATE AWARD FOR EXCELLENCE IN ROBOTICS ENGINEERING, MADRID 2014.

Al equipo de robótica *Corazón de Chileno*®, por los años de amistad, trabajo y compañerismo.

Agradezco también a Matías Oyarzun por guiarme en el Campus Creativo UNAB, en el uso de maquinarias y herramientas para el desarrollo del prototipado. Gonzalo Caroca por sus servicios de impresión 3D.

Finalmente, un agradecimiento a quien fue uno de los pilares fundamental en mi crecimiento académico, el profesor Néstor Palominos Gonzalez, profesor guía, por toda la ayuda otorgada y además, por todos estos años de dedicación en guiar e incentivar en el desarrollo de la ciencia aplicada, motivando a realizar distintos proyectos y participación en instancias relacionadas con la Robótica, automatización e innovación.

INDICE

AGRADECIMIENTOS	4
RESUMEN.....	8
ABSTRACT	10
1. ANTECEDENTES GENERALES	12
1.1. INTRODUCCIÓN	12
1.1.1. ESQUEMA DE FUNCIONAMIENTO SIMPLE	15
1.2. OBJETIVOS	16
1.2.1. OBJETIVO GENERAL.....	16
1.2.2. OBJETIVOS ESPECÍFICOS	16
1.3. ORGANIZACIÓN Y PRESENTACIÓN DEL TRABAJO	17
2. ESTADO DEL ARTE Y JUSTIFICACIÓN	18
2.1. HISTORIA.....	18
2.2. CARACTERÍSTICAS DE LOS ROBOTS MÓVILES.....	19
2.3. EL RESCATE.....	20
2.4. MARCO TEORICO	22
2.4.1. ACTUALIDAD	22
2.5. ANÁLISIS ESTRUCTURA, CHASIS Y TRACCIÓN	25
3. DESCRIPCIÓN DE LA PROBLEMÁTICA.....	28
3.1. IDENTIFICACION DEL PROBLEMA	28
3.2. DESARROLLO	30
3.2.1 SOFTWARE UTILIZADO	30
3.3. COMUNICACIONES	33
3.3.1. MODELO OSI Y PROTOCOLOS UTILIZADOS.....	35
3.3.2. PROTOCOLO TCP/IP	36
3.3.3. GPS / NMEA	38
3.4. DESCRIPCION DE HARDWARE.....	41
3.4.1. CONTROLADOR DE PROCESAMIENTO.....	41
3.4.2. CONTROLADOR DE MOTORES.....	43
3.4.3. GEOLOCALIZACION	45
3.4.4. SENSOR DE TEMPETARURA Y HUMEDAD.....	46

3.4.5.	VIDEO CAPTURA.....	48
4.	DESARROLLO DEL PROTOTIPO.....	49
4.1.	CARTA GANTT	49
4.2.	DIAGRAMAS.....	50
4.2.1.	DIAGRAMA DE BLOQUES	50
4.2.2.	DIAGRAMA SERVIDOR	51
4.2.3.	MAPA DE CONEXIONES GENERALES	52
4.2.4.	DIAGRAMA DE ENERGIZACIÓN Y SEÑALES.....	53
4.3.	FUENTE DE ENERGIA.....	55
4.4.	CONTROL DE MOTORES.....	57
5.	PRUEBAS Y FUNCIONAMIENTO DEL PROTOTIPO.....	59
5.1.	PRUEBAS	60
5.1.1.	SOFTWARE	60
5.1.2.	LOCOMOCIÓN.....	66
6.	ANALISIS DE CAMPO Y RESULTADOS	73
6.1.	PRUEBAS	73
6.1.1.	PRUEBAS Y VALIDACIÓN	74
6.1.2.	COMPARACIÓN	75
6.1.3.	CONCLUSIÓN Y DISCUSIÓN.....	76
7.	BIBLIOGRAFÍA.....	77
8.	ANEXOS.....	79
9.	CODIGOS LABVIEW	83
10.	CODIGOS PYTHON.....	85

ÍNDICE DE FIGURAS

Fig. 1: esquema de modo autónomo o tele operado	15
Fig. 2: Robot “CHIMP”, de la compañía DARPA	23
Fig. 3:iRobot 510 PackBot Multi-Mission Robot.	24
Fig. 4:Outdoor Rhino Track	25
Fig. 5: ROBOT modular “quadruped tank”	27
Fig. 6:Ranking de los 10 terremotos más devastadores del mundo.....	29
Fig. 7:CompacRIO National Instrument	32
Fig. 8: Etapas del proceso OSI	35
Fig. 9:Header TCP	36
Fig. 10:Trama TCP	37
Fig. 11: TRAMA GPS captura NMEA	39
Fig. 12:Ejemplo Trama, GPRMC (Recommended minimum specific GPS/Transit data).....	40
Fig. 13: Tarjeta Raspberry PI 3.	41
Fig. 14: Puerto GPIO, Raspberry PI 3.....	42
Fig. 15: Controlador Victor 888, VEX robotics.....	43
Fig. 16: Modulo GPS y antena, (GS-15C).....	45
Fig. 17: Gráfico de rango temperatura y húmeda mediante tren de pulsos	47
Fig. 18: Sensor de temperatura y humedad DTH11.....	47
Fig. 19: Carta Gantt	49
Fig. 20:Diagrama de bloques.....	50
Fig. 21: Diagrama servidor	51
Fig. 22: Mapa de conexiones generales.....	52
Fig. 23: Diagrama de energización y señales	53
Fig. 24: Tabla general del consumo en corriente y voltaje.	54
Fig. 25: Especificaciones técnicas de la Batería	55
Fig. 26: Esquemático del circuito controlador, etapa lógica, del Víctor 888	57
Fig. 27: Esquemático del circuito controlador, etapa control, Víctor 888	58
Fig. 28:Resultante DTH11.....	60
Fig. 29:Captura de código socket LabView	61
Fig. 30:Captura de localización Google-Maps y comandos LabView.....	61
Fig. 31: Simulación de mando control Xbox360. LabView	63
Fig. 32: Interfaz gráfica. LabView	65
Fig. 33: Vista frontal diseño 3D	66
Fig. 34: Vista posterior diseño 3D	67
Fig. 35: Cuadro características PLA	68
Fig. 36: Sistema de rueda Guia.....	69
Fig. 37: Vista general medidas formato DWG.....	70
Fig. 38: Vista general del interior del mecanismo de traslación.....	71
Fig. 39: Sistema engranaje Epicycloidal	72
Fig. 40: Correlación Torque - Corriente	74

RESUMEN

Se sabe que Chile es un país constantemente afectado por algún tipo de catástrofe natural (ONEMI, 2010). En el mercado existen muy pocos desarrollos donde se utiliza la robótica para dar soporte en asistencia a víctimas de estos siniestros. En Chile esta área representa una oportunidad para poder generar nuevos emprendimientos.

El objetivo de esta memoria es desarrollar metodologías y estrategias de rescate mediante un sistema que permita la utilización de un robot móvil, con el fin de dar solución y facilitar la asistencia de víctimas a un equipo de rescate.

Se diseñó e implementó un robot con tracción tipo oruga, con un peso de 25 Kg y un diámetro de 68cm x 75cm x 24cm (profundidad, ancho, altura), donde se utilizaron materiales como, aluminio, Technyl, policarbonato y perfiles de fierro para lo que es el chasis de este.

Este mecanismo es controlado mediante la plataforma Raspberry la cual actúa como servidor, donde por medio de una antena WIFI, un equipo cliente se conecta a ésta, mediante TCP/IP usando una interfaz gráfica en Labview que se utilizará como estación de monitoreo y sistema de adquisición de datos.

La interfaz gráfica que está montada en el computador permite dos funciones esenciales: en primer lugar, monitorear y visualizar el entorno del robot mediante 1 cámara web (streaming), conectada al servidor, se incorpora un sistema de geolocalización con GPS, y también a la vez un sistema capaz de hacer un catastro de temperatura y humedad ambiente por medio de un testeo mediante censado. La segunda función es la de controlar la tracción del robot mediante un operador humano, dos controladores para motores de corriente continua controlan el giro y la velocidad de los motores montados en la tracción.

En el desarrollo, se establecerán pruebas de campo, estableciendo una serie de muestreos en diferentes escenarios, que validarán el proyecto, probando cada uno de los modos de funcionamiento en condiciones reales. Se efectuaron

pruebas en tierra, pasto y concreto, dando resultados aceptables y obteniendo una autonomía de 4 hrs

El sistema no se limita solo a este tipo de robot, sino que da un hincapié a futuras implementaciones conjuntas con tipos de robot aptos para sobrevolar zonas, navegar en agua, deslizar en nieve, o superficies de altas temperaturas, entre otros.

Conceptos claves: Asistencia en desastres, robot móvil, tracción oruga, streaming, Raspberry PI, GPS, PWM

ABSTRACT

It is known that Chile is a country where it is constantly affected by some kind of natural catastrophe (Onemi, 2016), in the market there are few possibilities where robotics is used to support victims of these disasters.

In Chile this area is very underdeveloped and represents an opportunity to generate new ventures. The objective of this report is to develop methodologies and rescue strategies through a system that allows the use of a mobile robot, in order to provide a solution and facilitate the assistance of victims to a rescue team.

We designed and implemented a robot with traction type caterpillar, with a weight of 30.5 Kg and a diameter of 68cm x 75cm x 24cm (depth, width, height), where materials such as aluminum, tecnyl, polycarbonate and iron profiles were used to What is the chassis of this This device is controlled by the Raspberry platform, which can be used as a server, can be connected through TCP / IP using a graphical interface in LabVIEW that can be used as a monitoring station and data acquisition system.

The graphical interface that is mounted on the computer allows the essential functions: first, monitor and visualize the environment of the robot through 1 webcam (streaming), connected to the server, a geolocation system with GPS is incorporated, and also to the In addition, it can also perform a temperature and humidity catastrophe by means of a means test by census.

The second function is to control the robot by a human operator, the controllers for DC motors, the control of the rotation and the speed of the motors mounted in the traction.

In the development, field tests will be established, establishing a succession of data in different scenarios, which will validate the project, testing each of the operating modes in real conditions.

Tests were made on land, grass and concrete, giving acceptable results and obtaining a range of 4 hours for this development, it was necessary to implement

a prototype of land mobile robot, a real scale, which shows how this assistance system can be implemented.

The system is not limited only to the type of terrestrial robot, but it allows to realize future implementations with types of robots apt to overfly zones, to sail in water, to slide in snow, or surfaces of high temperatures, among others.

Concepts: Caterpillar robot, streaming, Raspberry, GPS, PWM

1. ANTECEDENTES GENERALES

1.1. INTRODUCCIÓN

Chile es un país de catástrofes, hace menos de 10 años el país visualizo varios incidentes en donde existe riesgo de pérdida de vida humana como, incendios a gran escala, terremotos, rescate de personas en minera.

El rescate de 33 hombres atrapados a más de 10Km debajo de la tierra. El mayor problema de este fue que no se podía exponer a “hombres” para el rescate de estos, las chimeneas estaban en mal estado y el movimiento de la tierra ante estas situaciones es inestable.

*“En el año 2010, el tsunami dejó 156 chilenos muertos y 25 **desaparecidos**. La madrugada del 27 de febrero de 2010, a las 3.34 horas, cuando la población estaba durmiendo, uno de los mayores terremotos de la historia sacudió el centro-sur de Chile, con una **magnitud** de 8,8 grados en la escala de Richter”. (ONEMI, 2010).*

Es un problema complejo, que los rescatistas pongan su vida en peligro, asumiendo los riesgos que existen en una misión de rescate, ya que cualquier movimiento imprevisto puede significar la muerte. Los estados mentales, situaciones de estrés y la presión en las personas muchas veces hacen cometer errores.

Un robot ligero, con tracción confiable y equipado para facilitar insumos de primeros auxilios facilita el trabajo a la hora de encontrar a personas y dar asistencia en ciertos casos de riesgo.

A continuación, en el presente documento se da a conocer la implementación de una maqueta prototipo funcional, para la experimentación e investigación del proyecto “sistema asistente para rescates”, esta maqueta está pensada y analizada para sobrepasar un tipo de ambiente terrestre orientado en la búsqueda y asistencia de víctimas en situación de catástrofe. Pese a que el proyecto no se limita solo en el escenario terrestre, sino que complementa otros escenarios, esta vez el enfoque es el terrestre enfrentando situaciones como, derrumbes, suelo pedregoso, fango, vegetación e incluso altas temperaturas como casos de incendios.

La robótica utiliza varias herramientas y soluciones que entregan las ingenierías como la mecánica, la electrónica o la ciencia de la programación. Estas son utilizadas para desarrollar distintas clases de robot, que pueden ser catalogados de diversas maneras como las funciones o mecanismos que utiliza.

Una de estas áreas se refiere a los robots móviles, los cuales como su nombre lo indica son capaces de desplazarse en el espacio utilizando varios actuadores y sensores que le permiten obtener datos del escenario, quedando a criterio de cada desarrollador la elección de estos.

Es junto a la idea de los robots móviles que nace el concepto de la autonomía de estos, es decir un robot que puede ser manejado con una programación previa de comportamiento y acción. El robot autónomo puede ser móvil como se señaló, pero también pueden ser robots estacionarios, como los brazos robóticos encargados de diversas tareas como pintura o soldadura, su área de trabajo se define en cuanto a su alcance o radio de acción, no debiendo preocuparse en gran medida por su entorno. Esto cambia radicalmente cuando se intenta trabajar con robots móviles, se debe tener en cuenta los objetos que componen el entorno, objetivos, obstáculos o simplemente componentes del sistema. Estas ideas se pueden entender de menor manera con las siguientes citas:

- *“Funcionar autónomamente implica que un robot puede operar en variadas condiciones, sin la necesidad de supervisión humana. Que un robot sea autónomo significa que puede adaptarse a los cambios en el ambiente o en sí mismo sin dejar de conseguir su objetivo”. (Llofriu, 2000).*
- *“Un sistema es autónomo en la medida en que su comportamiento está determinado por su propia experiencia.” (Noving, 2003).*

De estas citas se puede rescatar la idea de un robot que tiene múltiples capacidades para realizar diversas tareas.

Uno de los problemas que se debe solucionar para poder manejar de manera correcta un robot autónomo es la navegación en espacios desconocidos.

Actualmente una de las herramientas más utilizadas para la localización de un robot es el uso de tecnologías GPS, estas permiten estimar la posición relativa con una exactitud sorprendente variando en medida de la exactitud del sensor.

Con las condiciones adecuadas, el uso de GPS para estimar la posición es la tecnología mejor posicionada, el problema radica cuando se tienen escenarios donde esta tecnología no puede desarrollarse de manera correcta para un fin adecuado. Tal es el caso de instalaciones interiores, esta tecnología no puede posicionar en el piso correspondiente en caso de edificios por nombrar una dificultad. Para tales escenarios es recomendable utilizar otras técnicas como la odometría que nos permite saber la localización por medio de la utilización de sensores tipo encoders, basándose en los datos para estimar el recorrido realizado, a la vez se cuenta con la odometría visual con sensores de visión, que se encarga de extraer componentes del entorno catalogados como landmarks y por medio de algoritmos la posición del sensor respecto a estos.

Estas técnicas nos permiten saber la posición relativa del robot, lo que no nos permite saber es la estructura del mundo en el que estamos presentes. Para esto se pueden utilizar mapas previamente desarrollados que se cargan dentro del sistema del robot, con rutas predeterminadas es posible que la odometría pueda ser descartada usando solo algoritmos de visión para las tareas necesarias, esto puede parecer una buena solución para escenarios estáticos pero sufren una gran inconveniencia cuando ya el escenario pasa a ser dinámico, con cambios imprevistos en el ambiente que pueden entorpecer el movimiento del robot, teniendo que replantear la ruta.

1.1.1. ESQUEMA DE FUNCIONAMIENTO SIMPLE



FIG. 1: ESQUEMA DE MODO AUTÓNOMO O TELE OPERADO

1.2. OBJETIVOS

1.2.1. OBJETIVO GENERAL

Este desarrollo se basa en proporcionar una herramienta, que por medio del diseño, materialización e implementación de un sistema prototipo robotizado, tipo móvil terrestre con tracción oruga, que pueda ser controlado y monitoreado por medio de una interfaz de usuario, con el fin de incorporarse en la ayuda para tácticas de rescate a víctimas humanas ante escenarios de desastres naturales, por lo que no es con fines comerciales. Esta plataforma esta dirigida con un fin investigativo, para operarios que conforman equipos profesionales del rubro del rescate.

1.2.2. OBJETIVOS ESPECÍFICOS

- Establecer una solución de comunicación, por medio de una red telemétrica.
- Desarrollar códigos con Python y Open CV, que implementan a la interfaz de usuario en computador, permitiendo la visualización y control, ya sea manual o autónomo.
- Diseñar interface gráfica utilizando LabView para interacción con usuario y un sistema de adquisición de datos.
- Establecer una base teórica sobre los principios básicos y ergonómicos que subyacen en el diseño en tres dimensiones del modelo mecanizado.
- Implementar un sistema que permita realizar una video conferencia que permita una mejor interacción con la victimas
- Ejecutar pruebas de funcionamiento y operación del prototipo.

1.3. ORGANIZACIÓN Y PRESENTACIÓN DEL TRABAJO

La estructura del desarrollo, de este proyecto se encuentra estructurada de la siguiente forma en una secuencia progresiva de fase:

- En el Capítulo 1 se introduce la idea que se quiere desarrollar, una idea global del proyecto, así como el estado del arte y su justificación. Se presentan los objetivos principales y específicos.
- En el Capítulo 2 se plantea el problema existente con respecto a las soluciones que hay hoy en día.
- En el Capítulo 3 se muestra la solución al problema planteado de forma técnica y detallada, en esta etapa se describe y analiza cada uno de los componentes utilizados. Se investigan los ámbitos científico-técnicos, considerando esto como una base teórica que sustentará las decisiones tomadas al momento del diseño.
- En el Capítulo 4 se desarrolla la proyección del prototipo y se detalla su estructura lógica mediante esquemas que cubren sus módulos estructurales, así como el flujo de información. El planteamiento de la solución y los detalles técnicos se extienden las etapas de mecánica, donde se detallan las características principales de la estructura y control, lo que concluye con todo el detalle del software y hardware.
- En el Capítulo 5 se hará distintas pruebas para ver el funcionamiento del prototipo y conocer los alcances, las fallas y las limitaciones que este tenga, exponiendo las consideraciones experimentales del prototipo.
- En el Capítulo 6 se hará un análisis de resultados, pruebas de campo, para posteriormente realizar conclusiones y evaluaciones con respecto a los distintos pasos ejecutados para lograr el dispositivo.
- Desde el Capítulo 7 en adelante se tratan referencias y anexos.

2. ESTADO DEL ARTE Y JUSTIFICACIÓN

2.1. HISTORIA

La robótica móvil, Uno de los campos de aplicación más significativos de esta, consiste en robots capaces de operar en condiciones exteriores sobre terrenos no preparados. Sin embargo, conseguir que los robots se muevan de forma eficiente y precisa en este tipo de entornos no es una tarea sencilla. Un primer aspecto crítico es el sistema de locomoción. En este caso, las orugas constituyen una alternativa sólida a otro tipo de sistemas y desde principios del siglo XX han demostrado sus bondades en vehículos tripulados.

Fue en los años 80 y principio de los 90 que se publicó el trabajo realizado por Smith junto a P. Chesseman que desarrollo una representación concreta de la incertidumbre para la localización, además de la investigación del profesor Hugh Durrant-White. Ambos estudios son fundamental para que se tomara importancia en el desarrollo de una práctica encargada de la navegación en robots, dejando de ser un aspecto teórico del área.

Proporciona las bases para desarrollar formas de lidiar con el error de los sistemas, implementando representación estocástica (stochastic map), es decir un sistema de comportamiento no determinista, que toma las variables esperadas del sistema en conjunto con probables variables aleatorias y valores aleatorios, comúnmente es una referencia a los sistemas probabilísticos.

Otros trabajos importantes fueron los realizados por Elfes y Moravec, quienes contribuyeron con los algoritmos de occupancy grid. (E. Nelson, 2002).

Posterior a estos años se sigue avanzando con trabajos como los de Crowley y Leonard que proponen métodos de SLAM usando sensores ultrasónicos.

Vadorpe y González realizan lo propio utilizando por su parte sensores laser en su lugar. Al mismo tiempo Ayache y Faungeras desarrollaron el primer trabajo sobre navegación y mapeo visual.

2.2. CARACTERÍSTICAS DE LOS ROBOTS MÓVILES

En la industria podemos encontrar distintos tipos de manipuladores robóticos y entre ellos se encuentran los robots móviles, un robot móvil es una máquina automática, que es capaz de trasladarse dentro de un ambiente dado, es decir, una plataforma mecánica dotada de un sistema de locomoción capaz de navegar a través de un determinado ambiente de trabajo, dotado de cierto nivel de autonomía para su desplazamiento portando cargas. Sus aplicaciones pueden ser muy variadas y siempre están relacionadas con tareas que normalmente son riesgosas o nocivas para la salud humana, en áreas como la agricultura, en el transporte de cargas peligrosas o en tareas de exploración solitarias o cooperativas junto a 5 otros vehículos no tripulados

Dentro de esta descripción podemos identificar robots AGV (Automatic Guided Vehicle), utilizados en la industria generalmente para un tipo de función específica especialmente en tareas repetitivas y con alta cadencia , como por ejemplo el paletizado o traslado de material. La característica principal de un robot AGV es que es un vehículo no tripulado, generalmente comandado en respuesta al tipo de guiado que se asigne este guiado puede ser por medios de; Sistema de guiado láser por reflectores, Sistema de guiado láser por contorno, Sistema de guiado por puntos magnéticos y entre otros.

2.3. EL RESCATE

Para enfrentar la problemática de diseñar y prototipar cualquier tipo de maqueta móvil en el ámbito del diseño industrial, debemos considerar varios aspectos, en lo que respecta la etapa de estudio del arte, recopilación de información y levantamiento del problema, tenemos que tomar en cuenta ciertos conceptos que concluirán en seguir, entender normas y protocolos en lo que respecta a los diseños ergonómicos.

Es clave relacionarse con los conceptos como el Rescate en sí, Según la real academia de la Lengua española, el Rescate se define como la acción de rescatar, Liberar a una persona de cierta situación de peligro o que le oprime, agobia o molesta.

Seguin la ASTM (American Society of Testing Materials):

“La búsqueda y rescate urbano se define como la combinación de estrategias, tácticas y operaciones que permitan localizar, tratar médicamente y sacar a las víctimas atrapadas.” (Messina, 2006).

Las etapas que conforman el rescate, la búsqueda por vía terrestre, en sectores de difícil acceso o espacio urbano, consisten en múltiples etapas y los equipos de rescate deben realizar una variedad de funciones.

Entre los ejemplos de las funciones que puede llevar a cabo un equipo de rescate de la Agencia Federal de Gestión de Emergencias (Federal Emergency Management Agency FEMA, 2017), mencionan que:

- llevar a cabo por sí mismos búsquedas y rescates en edificios derrumbados; facilitar evaluaciones y cuidados médicos a las víctimas atrapadas; evaluar y controlar los riesgos, como tuberías de gas o cables eléctricos dañados; y evaluar y estabilizar estructuras dañadas.

- Los robots podrían potencialmente ayudar al personal de rescate a realizar todas estas funciones

Con el fin de estandarizar y certificar a los robots de rescate, la ASTM en el 2005, se estableció el grupo en misión especial E54.08.01 de ASTM para crear normas de rendimiento para robots con aplicación para búsqueda y rescate urbano.

El Departamento de Seguridad Nacional (Department of Homeland Security, DHS) y otros percibieron la falta de normas exhaustivas que respaldaran el desarrollo, comprobación y certificación de tecnologías robóticas eficaces para el uso en aplicaciones de búsqueda y rescate urbano. Estas normas contemplarán las funciones robóticas de movilidad, detección, navegación, planificación, integración en suministros de reserva operacionales (inventarios de herramientas y equipo en almacén) e interacción entre el sistema y los humanos.

En resumen, esta estandarización, orientada netamente a la acción del rescate mediante la utilización de robots móviles, se resume en los siguientes puntos.

- Desplegarse rápidamente.
- Permitir la operación remota desde una distancia segura.
- Tener la habilidad de moverse con rapidez en entornos complejos.
- Tener suficiente protección como para evitar dañarse mientras cumple su tarea.
- Ser rentable y poder repararse en el campo.
- Estar disponible en diversos tamaños y poder adaptarse a entornos diversos.

2.4. MARCO TEORICO

2.4.1. ACTUALIDAD

“Se define como manipulador a un mecanismo formado generalmente por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Es multifuncional y puede ser gobernado directamente por un operador humano o mediante dispositivo lógico.”

“Un robot es un manipulador reprogramable, multifuncional, controlado automáticamente, que puede estar fijo en un sitio o moverse, y que está diseñado para mover materiales, piezas, herramientas o dispositivos especiales, por medio de movimientos variables programados para la realización de diversas tareas o trabajos.” (ISO 8373, 2012).

Gobiernos, agencias militares, agencias de rescate y agentes privados han puesto sobre la mesa el tema de utilización de tecnologías avanzadas para el rescate de vidas, las mejores universidades del mundo tienen centros de investigación y desarrollo de estos, incluso se han hecho instancias de desafíos competitivos donde la comunidad científica tecnológica busca superar el desarrollo de robots móviles orientados al rescate entre varias casas de estudios.

Un ejemplo es la LAUREATE AWARD FOR EXCELLENCE IN ROBOTICS ENGINEERING en su versión 2014 – 2015, donde se plantea un desafío del diseño y desarrollo de un sistema robótico orientado al rescate de personas.

Los líderes en innovación y desarrollo de robótica móvil orientada al uso militar y rescate son la compañía DARPA (Defense Advanced Research Projects Agency, DARPA, 2015) y ARMY TECHNOLOGY.

DARPA, es una agencia del Departamento de Defensa de Estados Unidos responsable del desarrollo de nuevas tecnologías para uso militar.

Fundada en el año 1958 posterior y como consecuencia de la guerra tecnológica en la Guerra Fría.

Dentro de sus redes DARPA en el año 2013 publicó un desafío para la construcción de un robot para casos de catástrofe. DARPA Robotics Challenge (DRC).

El robot de cuatro extremidades del equipo, CHIMP, es capaz de realizar las ocho tareas del desafío, incluyendo conducir un vehículo, subir escaleras y usar herramientas eléctricas, que encontrará en un curso de desastre simulado que la Agencia de Proyectos de Investigación Avanzada de Defensa está ensamblando en Fairplex en Pomona, California. Pero deberá completar todas las tareas dentro del tiempo de ejecución de una hora.

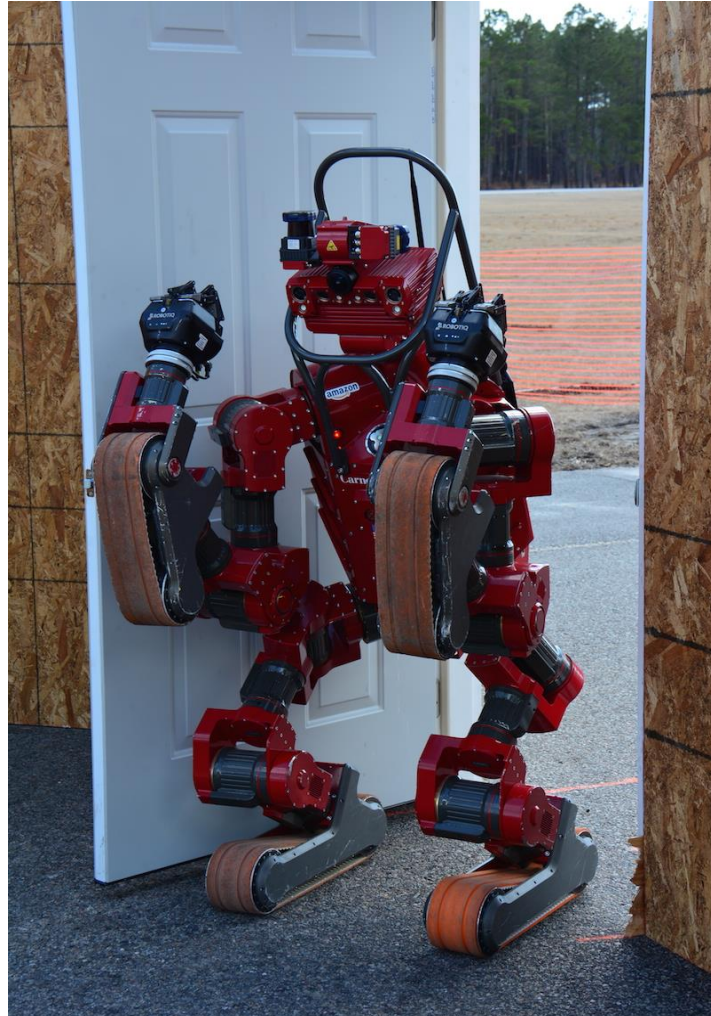


FIG. 2: ROBOT "CHIMP", DE LA COMPAÑÍA DARPA

En octubre de 2010, el Centro de Contrataciones TACOM del Ejército de los EE. UU. En Warren, Michigan, realizó un pedido en millones de dólares para el

software de inteligencia robótica iRobot Aware 2 y repuestos PackBot para actualizar su flota de 510 FasTac al estándar 510 PackBot. iRobot obtuvo \$ 4.4 millones de pedidos internacionales para la entrega de 27 robots en el primer trimestre de 2011.



FIG. 3: iROBOT 510 PACKBOT MULTI-MISSION ROBOT.

“PackBot es una serie de robots militares de iRobot , una compañía internacional de robótica fundada en 1990. Más de 2000 fueron utilizados en Irak y Afganistán .Ellos fueron utilizados de nuevo para ayudar a buscar a través de los escombros del World Trade Center después de 9/11 en 2001. Otro ejemplo de la tecnología PackBot siendo implementado fue a la dañada planta nuclear de Fukushima tras el terremoto de Tohoku de 2011 y Tsunami donde fueron los primeros en evaluar el sitio”. (ARMY Technology, 2011).

2.5. ANÁLISIS ESTRUCTURAL, CHASIS Y TRACCIÓN

Existen muchos tipos de robot terrestres que ocupan la tracción con ruedas tipo orugas en su chasis como método de desplazamiento para terrenos difíciles, tienen distintas formas, distintos materiales de construcción de estas y distintos portes. El criterio que se utiliza para la construcción de estos es en función de la aplicación y tareas asignadas al robot.

Para efectos de esta investigación y basándonos en el objetivo principal, que es plantear una aplicación con el fin de investigar sobre la utilización de estas tecnologías en estrategias de rescate. Se arma un prototipo funcional utilizando materiales sólidos y robustos, pero que también están hechos para hacer estudios y verificar resultados sobre el comportamiento de estos sistemas.

En el diseño, basamos en un modelo de chasis llamado “ Outdoor Rhino Track Drive (am-3446)” de la industria AndyMark (AndyMark, 2015).

Este modelo junto a otros de la misma compañía, fue fabricado con el fin de otorgar esta plataforma a estudiantes e ingenieros que participan en FIRST Robotics Competitions, Torneo mundial de robótica, fundado y administrada por ingenieros líderes a nivel mundial y patrocinada por las compañías más importantes internacionalmente.



FIG. 4:OUTDOOR RHINO TRACK

Es un chasis que cuenta con dos módulos de correderas en paralelo permitiendo el desplazamiento de un vehículo con locomoción de alta tracción.

Estos módulos cuentan con la particularidad de fraccionar la tracción por medio de ruedas tradicionalmente conocidas en el mercado con el nombre de “oruga”.

Características técnicas:

Peso: 32 Kg (peso puede variar en función de personalización). Caída de centro distancia polea: 0,15 pulgadas más baja que las dos poleas adyacentes. La banda de rodamiento Material: Caucho de nitrilo Carbox. Pise Patrón: Roughtop. Correa de distribución Estilo: HV con el Centro de seguimiento de V. Correa de distribución Longitud: 65,5 pulgadas. Longitud total: 30.44 pulgadas con brazo superior en ángulo de 34 grados, (31,3 pulgadas con brazo superior a 5 grados y se trasladó a punto de pivote secundario). Ancho total: 25,875 pulgadas de placa a placa

Lado espesor de la chapa: 3/16 pulgadas. Material de los laterales de la placa: aluminio 6061--T6

En el mercado existen muchas variedades para poder optar en comprar chasis para el modelamiento en el robot móvil, de distinto material y características, otra opción que podría ocuparse para modelamiento es la siguiente:



FIG. 5: ROBOT MODULAR "QUADRUPED TANK"

Este chasis, consta de una estructura de cuerpo sólido, cuatro ruedas con orugas, tren de engranajes y sistema de suspensión independiente. Es un sistema fuerte, estable, movilidad con 4 motores solo para la tracción.

El chasis cuenta con sistema de absorción de choque, de suspensión independiente, al mismo tiempo, para hacer que se ejecute en superficie de la carretera desigual, puede hacer frente fácilmente con el fuselaje intermedio obligan boca del aparato de gobierno de instalar aparato de gobierno y el plano ancho chasis plataforma proporciona suficiente espacio para la instalación.

Características técnicas:

- Tamaño: 393mm * 206mm * 84mm (largo * ancho * alto)
- Voltaje de funcionamiento: 6 a 12 V (12 V velocidad 350 RPM),
- Materia: aluminio 6061-T6,
- Montaje para motores DC: Idealmente motores con engranajes planetarios
- Material Oruga: Plástico.

3. DESCRIPCIÓN DE LA PROBLEMÁTICA

3.1. IDENTIFICACION DEL PROBLEMA

No siempre las situaciones donde la vida humana está en peligro y se requiere de equipos de rescate son por situaciones naturales, sino que el hombre es responsable de estas.

Vivimos en un planeta donde la industrialización se ha apoderado de nuestra geografía, la contaminación y otros factores provocan situaciones donde la vida humana se pone en peligro, desde hace décadas, existe una ocurrencia mayor de desastres provocando cada vez más daños humanos, materiales y económicos en los países de la región latinoamericana. Pero, los datos (muertos, heridos, viviendas dañadas, etc.) no explican la complejidad de estos fenómenos desastrosos. Los estudios científicos no son por ahora suficientes para entender y analizar los desastres. Por lo tanto, las ciencias sociales tienen ahora un papel relevante en este tema debido a su potencial de conocimiento de las complejas relaciones entre sociedad y naturaleza para poder avanzar en la construcción de una propuesta multidisciplinaria.

La clasificación de los desastres naturales, referidos a los "agentes dinámicos", se clasifican en cinco tipos, dos de origen natural (geológicos e hidrometeorológicos) y tres de carácter social (físico- químicos, sanitarios y "socio-organizativos"), que pueden alterar el funcionamiento normal de la sociedad y el entorno natural, produciendo en ellos un estado de desastres.

Ante esto, la problemática que se presenta si aplicamos tecnología de robot móviles, en primer lugar, es la locomoción, los terrenos luego de desastres del tipo terrestre pueden ser de escombros, altas temperaturas, humedad, actividad química, derrames de material, entre otros.

Comunicación y telemetría, ante condiciones de catástrofe, es de conocimiento público, que lo primero que se deja de suministrar es la energía eléctrica y las redes de telefonía.

Por ende, aplicar tecnologías usando robots para rescate, es lo primero que se debe considerar, la comunicación y sustento energético.

Mayor posibilidad de tener pérdidas de ondas de radio frecuencia, es en terrenos montañosos.

Suministros, la gran pregunta es, ¿Qué funciones debe cumplir un robot de asistencia?, que funciones debe ejecutar y cómo? Si pensamos en desastres naturales, rescate andino, es importante saber el factor de los escasos, una persona que necesita ser rescatada, lo más probable es que presente problemas de salud o alimentación.

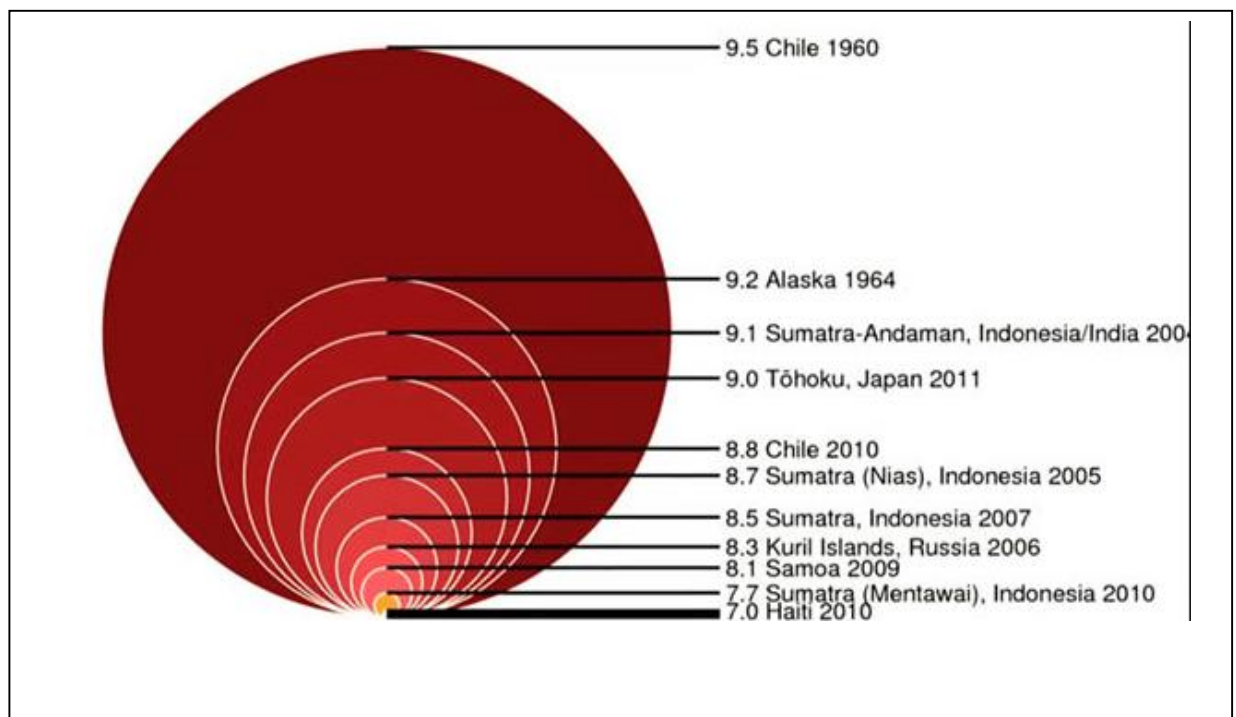


FIG. 6: RANKING DE LOS 10 TERREMOTOS MÁS DEVASTADORES DEL MUNDO.

3.2. DESARROLLO

3.2.1 SOFTWARE UTILIZADO

PYTHON

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Para efectos de este proyecto se utilizará el lenguaje PYTHON para montar un servidor, el servidor en este caso físicamente va a ejecutarse en una tarjeta controladora montada en el robot, la función de este servidor es por medio de socket.

El Socket designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.

El término socket es también usado como el nombre de una interfaz de programación de aplicaciones (API) para la familia de protocolos de Internet TCP/IP, provista usualmente por el sistema operativo.

La función principal, es transmitir datos al cliente, estos datos corresponden a temperatura, humedad, trama GPRS, e imágenes en tiempo real por medio de una cámara web, a su vez, el cliente tiene que validar esta conexión y enviar los comandos

Librerías utilizadas

- Pynmea2: Esta es una librería de Python usada para el protocolo NMEA 0183, se usa para especificar eléctricamente datos del GPS y transformarlos en un formato elegible para Google Maps.
- GPIO (General Purpose Input/Output, Entrada/Salida de Propósito General):
Es la librería para especificar los pines genéricos del microcontrolador de la Raspberry, cuyo comportamiento (incluyendo si es un pin de entrada o salida) se puede controlar (programar) por el usuario en tiempo de ejecución.
- Time: Proporciona un conjunto de funciones para trabajar con fechas, horas y demás de éstas funciones hay otras relacionadas en los módulos como lo que es el **DATE TIME** y **CALENDAR** (Join GitHub today, 2015).

LABVIEW

El entorno de programación LabView, creado y comercializado por National Instruments. En un inicio fue solamente Operar en dispositivos de la marca MAC, Actualmente disponible en muchas plataformas como, Linux, Windows y Vista.

El rubro es generalmente la adquisición de datos, control de instrumentos e instrumentación virtual. En los últimos años también ha extendido su negocio a sistemas de comunicaciones y de CAN BUS, en buena parte apoyándose en las arquitecturas PXI y C-Rio (CompactRIO), C-Rio mencionado mucho en industria utilizando el entorno de programación LabView.



FIG. 7:COMPACRIO NATIONAL INSTRUMENT

3.3. COMUNICACIONES

Se utilizará WIFI como método de comunicación, debido a comparaciones experimentales en pruebas que se realizaron comparando con transmisores de radio frecuencia de, con un equipo de Radio Turnigy 9X 2.4Ghz Tx y Rx, utilizado en aeromodelismo, pese a que tiene un alcance de 2000 metros, este módulo no tenía las prestaciones suficientes para el proyecto.

Las comunicaciones por radio frecuencia son hasta el día de hoy el método más utilizado en los sistemas de telecomunicación. La base subyacente a esta categoría de mecanismos se origina en los fenómenos asociados a la inducción y resonancia entre sistemas de emisión y recepción electromagnética (EM). Esta comunicación es una interacción entre al menos dos sujetos, quienes se pasan información a través de un medio, la interface que soporta este intercambio, en este caso, es proporcionada por ondas electromagnéticas, esencialmente a través del aire.

La antena está integrada dentro de la Raspberry Pi, es una antena genérica con un rango estándar. Si se quiere obtener más rango, lo ideal es adquirir una antena receptora industrial, existen algunas que incluso superan los 5 km de distancia. La industria EMERSON, está invirtiendo en tecnologías WIRILESS

El wifi básicamente es una red wireless (sin cables), utiliza las ondas de radio de la misma forma que lo hacen los teléfonos móviles o celulares, las televisiones y las propias radios. De hecho, la comunicación a través de una red wireless es muy parecida a las dos vías de comunicación de la radio.

Esto es lo que sucede:

El adaptador inalámbrico (wireless) de un ordenador traduce los datos en forma de señal de radio y los transmite utilizando una antena.

Un Router inalámbrico recibe la señal y la decodifica. El router envía la información a Internet utilizando una conexión física, cableada, de Ethernet.

Se utilizo este sistema para transmitir los datos ya que este me permite comunicar tramas bidireccionales como por ejemplo para hacer una video conferencia, prestación que los módulos de radio frecuencia no tienen. Para transmitir los datos bidireccionales se utilizó el protocolo TCP/IP

3.3.1. MODELO OSI Y PROTOCOLOS UTILIZADOS

Modelo de interconexión de sistemas abiertos (ISO/IEC 7498-1) consta de 7 capas de un proceso de transmisión datos entre equipos y dispositivos informáticos, donde cada capa se encarga de ejecutar una determinada parte del proceso global.



FIG. 8: ETAPAS DEL PROCESO OSI

3.3.2. PROTOCOLO TCP/IP

Define cuidadosamente cómo se mueve la información desde el remitente hasta el destinatario. En primer lugar, los programas de aplicación envían mensajes o corrientes de datos a uno de los protocolos de la capa de transporte de Internet, UDP (User Datagram Protocol) o TCP (Transmission Control Protocol). Estos protocolos reciben los datos de la aplicación, los dividen en partes más pequeñas llamadas paquetes, añaden una dirección de destino y, a continuación, pasan los paquetes a la siguiente capa de protocolo, la capa de red de Internet.

La capa de red de Internet pone el paquete en un datagrama de IP (Internet Protocol), pone la cabecera y la cola de datagrama, decide dónde enviar el datagrama (directamente a un destino o a una pasarela) y pasa el datagrama a la capa de interfaz de red.

La capa de interfaz de red acepta los datagramas IP y los transmite como tramas a través de un hardware de red específico, por ejemplo, redes Ethernet o de Red en anillo.

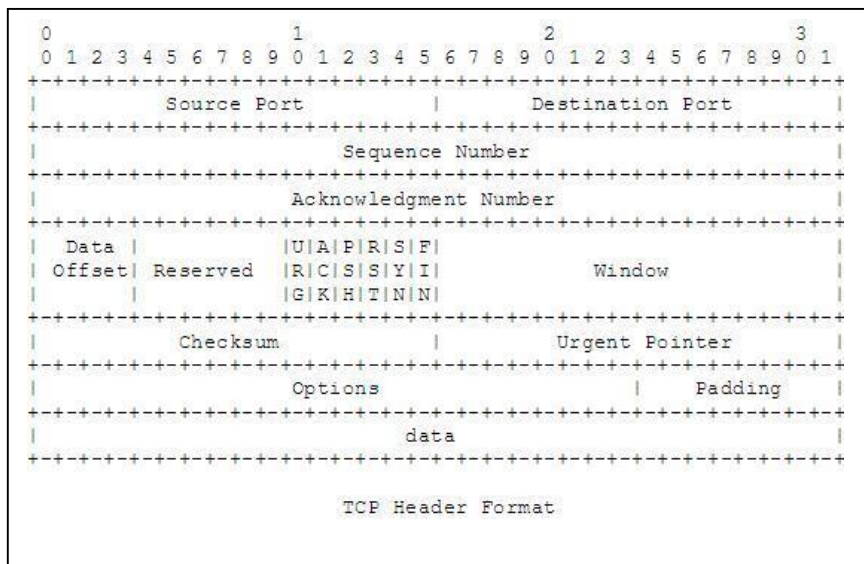


FIG. 9:HEADER TCP

La capa de interfaz de red (en este caso, un adaptador Ethernet) recibe las tramas. La capa de interfaz de red quita la cabecera Ethernet y envía el datagrama hacia arriba hasta la capa de red. En la capa de red, Protocolo Internet quita la cabecera IP y envía el paquete hacia arriba hasta la capa de transporte. En la capa de transporte, TCP (en este caso) quita la cabecera TCP y envía los datos hacia arriba hasta la capa de aplicación.

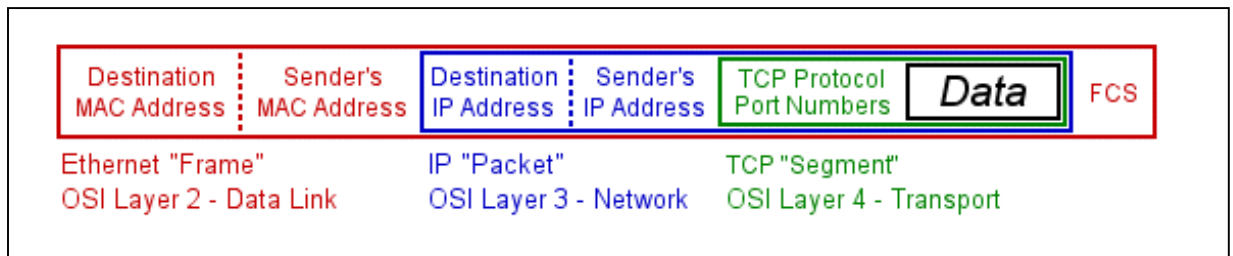


FIG. 10:TRAMA TCP

3.3.3. GPS / NMEA

Por otra parte, la comunicación consta con un módulo GPS para la geolocalización del robot.

El GPS funciona mediante una red de 24 satélites en órbita sobre el planeta Tierra, a 20 200 km de altura, con trayectorias sincronizadas para cubrir toda la superficie de la Tierra.

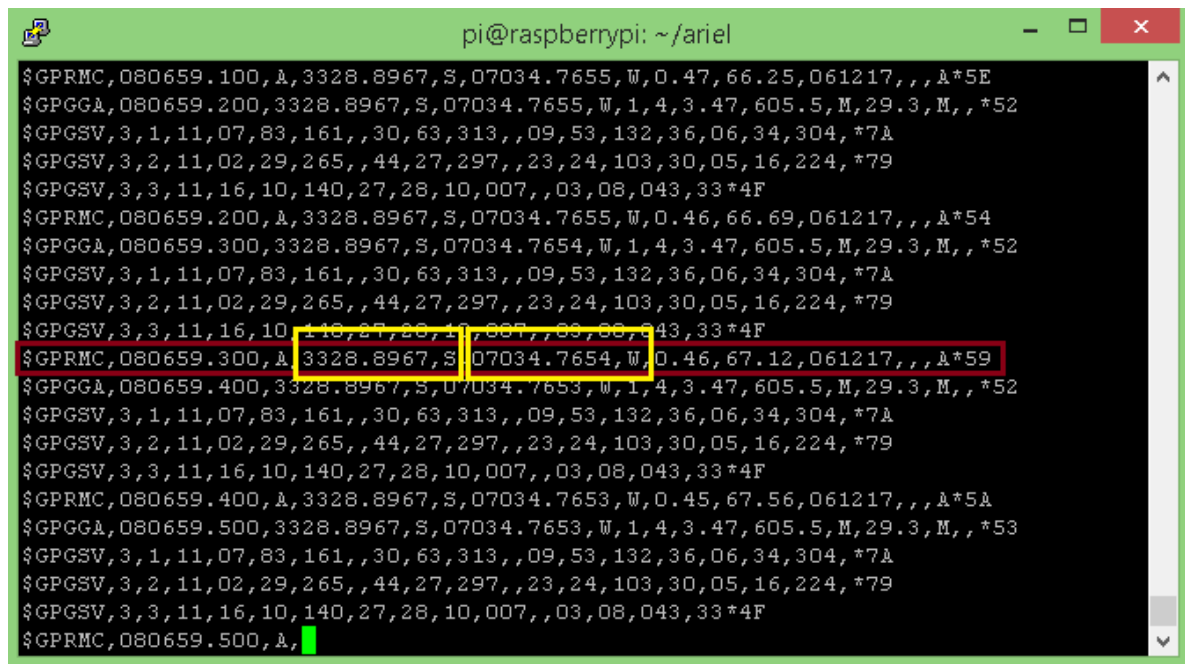
Cuando se desea determinar la posición, el receptor que se utiliza para ello localiza automáticamente como mínimo tres satélites de la red, de los que recibe unas señales indicando la identificación y la hora del reloj de cada uno de ellos.

Con base en estas señales, el aparato sincroniza el reloj del GPS y calcula el tiempo que tardan en llegar las señales al equipo, y de tal modo mide la distancia al satélite mediante el método de trilateración inversa, el cual se basa en determinar la distancia de cada satélite al punto de medición.

Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los satélites. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene la posición absoluta o coordenada del punto de medición.

También se consigue una exactitud extrema en el reloj del GPS, similar a la de los relojes atómicos que lleva a bordo cada uno de los satélites.

Captura de trama GPS



```
pi@raspberrypi: ~/ariel
$GPRMC,080659.100,A,3328.8967,S,07034.7655,W,0.47,66.25,061217,,,A*5E
$GPGGA,080659.200,3328.8967,S,07034.7655,W,1,4,3.47,605.5,M,29.3,M,,*52
$GPGSV,3,1,11,07,83,161,,30,63,313,,09,53,132,36,06,34,304,*7A
$GPGSV,3,2,11,02,29,265,,44,27,297,,23,24,103,30,05,16,224,*79
$GPGSV,3,3,11,16,10,140,27,28,10,007,,03,08,043,33*4F
$GPRMC,080659.200,A,3328.8967,S,07034.7655,W,0.46,66.69,061217,,,A*54
$GPGGA,080659.300,3328.8967,S,07034.7654,W,1,4,3.47,605.5,M,29.3,M,,*52
$GPGSV,3,1,11,07,83,161,,30,63,313,,09,53,132,36,06,34,304,*7A
$GPGSV,3,2,11,02,29,265,,44,27,297,,23,24,103,30,05,16,224,*79
$GPGSV,3,3,11,16,10,140,27,28,10,007,,03,08,043,33*4F
$GPRMC,080659.300,A,3328.8967,S,07034.7654,W,0.46,67.12,061217,,,A*59
$GPGGA,080659.400,3328.8967,S,07034.7653,W,1,4,3.47,605.5,M,29.3,M,,*52
$GPGSV,3,1,11,07,83,161,,30,63,313,,09,53,132,36,06,34,304,*7A
$GPGSV,3,2,11,02,29,265,,44,27,297,,23,24,103,30,05,16,224,*79
$GPGSV,3,3,11,16,10,140,27,28,10,007,,03,08,043,33*4F
$GPRMC,080659.400,A,3328.8967,S,07034.7653,W,0.45,67.56,061217,,,A*5A
$GPGGA,080659.500,3328.8967,S,07034.7653,W,1,4,3.47,605.5,M,29.3,M,,*53
$GPGSV,3,1,11,07,83,161,,30,63,313,,09,53,132,36,06,34,304,*7A
$GPGSV,3,2,11,02,29,265,,44,27,297,,23,24,103,30,05,16,224,*79
$GPGSV,3,3,11,16,10,140,27,28,10,007,,03,08,043,33*4F
$GPRMC,080659.500,A,
```

FIG. 11: TRAMA GPS CAPTURA NMEA

En Python, con el comando SPLIT, podemos capturar la trama que comunica el GPS, específicamente lo que hace el comando, retorna una LISTA a partir de la cadena, la cual se delimita por los índices que indique,

La imagen capturada, tiene la trama que corresponde a las posiciones de latitud longitud (norte, sur, este y oeste) como explica en el siguiente ejemplo:

Estas tramas comienzan con el enunciado \$GPRMC, \$GPGGA, \$GPGSA, \$GPGSV y otros, lo que indican es el tipo de datos contiene esa trama, luego los campos propios de ese tipo de mensaje y finalmente el Checksum.

```
eg1. $GPRMC,081836,A,3751.65,S,14507.36,E,000.0,360.0,130998,011.3,E*62
eg2. $GPRMC,225446,A,4916.45,N,12311.12,W,000.5,054.7,191194,020.3,E*68
```

```
225446      Time of fix 22:54:46 UTC
A           Navigation receiver warning A = OK, V = warning
4916.45,N  Latitude 49 deg. 16.45 min North
12311.12,W Longitude 123 deg. 11.12 min West
000.5      Speed over ground, Knots
054.7      Course Made Good, True
191194     Date of fix 19 November 1994
020.3,E    Magnetic variation 20.3 deg East
*68        mandatory checksum
```

```
eg3. $GPRMC,220516,A,5133.82,N,00042.24,W,173.8,231.8,130694,004.2,W*70
      1      2      3      4      5      6      7      8      9      10     11     12
```

```
1 220516      Time Stamp
2 A           validity - A-ok, V-invalid
3 5133.82    current Latitude
4 N           North/South
5 00042.24   current Longitude
6 W           East/West
7 173.8      Speed in knots
8 231.8      True course
9 130694     Date Stamp
10 004.2     Variation
11 W         East/West
12 *70       checksum
```

```
eg4. $GPRMC,hhmmss.ss,A,llll.ll,a,yyyyy.yy,a,x.x,x.x,ddmmyy,x.x,a*hh
```

```
1 = UTC of position fix
2 = Data status (V=navigation receiver warning)
3 = Latitude of fix
4 = N or S
5 = Longitude of fix
6 = E or W
7 = Speed over ground in knots
8 = Track made good in degrees True
9 = UT date
10 = Magnetic variation degrees (Easterly var. subtracts from true course)
11 = E or W
12 = Checksum
```

FIG. 12: EJEMPLO TRAMA, GPRMC (RECOMMENDED MINIMUM SPECIFIC GPS/TRANSIT DATA)

3.4. DESCRIPCION DE HARDWARE

3.4.1. CONTROLADOR DE PROCESAMIENTO

Se propone un sistema, aplicación de código abierto que se ejecute en un sistema robotizado que puede ser tipo terrestre, marítimo o aéreo. La idea es generar una red, cliente y múltiples servidores, para esto se utilizó como herramienta principal para realizar el procesamiento la tarjeta controladora Raspberry Pi 3, posee un micro procesador ARM con potencia de hasta 1GHz, integrado en un chip Broadcom BCM2835. Además, cuenta con 512 MB de RAM, un GPU Videocore IV, todo lo necesario para poder ejecutar programas básicos, navegar por internet y por supuesto programar.



FIG. 13: TARJETA RASPBERRY PI 3.

- A 1,2 GHz 64-bit de cuatro núcleos ARMv8 CPU //802.11n Wireless LAN
- Bluetooth 4.1 Bluetooth Low Energy (BLE)
- 1 GB de RAM // 4 puertos USB // 40 pines GPIO
- Puerto lleno de HDMI // Puerto Ethernet
- Conector de audio de 3,5 mm combinado y vídeo compuesto
- Interfaz de la cámara (CSI) // Interfaz de pantalla (DSI)
- Ranura para tarjetas micro SD (ahora push-pull en lugar de push-push)
- VideoCore IV 3D núcleo de gráficos

En la tarjeta, los pines GPIO no tienen ningún propósito especial definido, y no se utilizan de forma predeterminada. La idea es que a veces, para el diseño de un sistema completo que utiliza el chip podría ser útil contar con un puñado de líneas digitales de control adicionales, y tenerlas a disposición ahorra el tiempo de tener que organizar circuitos adicionales para proporcionarlos. Por ejemplo, los chips Realtek ALC260 (códec de audio) tienen 8 pines GPIO, que quedan sin utilizar de forma predeterminada. Algunos integradores de sistemas (Acer Inc. laptops) que emplea el ALC260 utilizan la primera GPIO (GPIO0) para encender el amplificador utilizado para los altavoces internos y el conector de auriculares del ordenador portátil.

Para este caso, utilizamos estos terminales para conectar señales de entradas y salidas, conexionado del GPS, sensor de humedad/temperatura, y señales PWM para el control de motores.

Raspberry Pi2 GPIO Header					
Pin#	NAME		NAME	Pin#	
01	3.3v DC Power	⬇	DC Power 5v	02	⬆
03	GPIO02 (SDA1 , PC)	⬇	DC Power 5v	04	⬆
05	GPIO03 (SCL1 , PC)	⬇	Ground	06	⬆
07	GPIO04 (GPIO_GCLK)	⬇	(TXD0) GPIO14	08	⬆
09	Ground	⬇	(RXD0) GPIO15	10	⬆
11	GPIO17 (GPIO_GEN0)	⬇	(GPIO_GEN1) GPIO18	12	⬆
13	GPIO27 (GPIO_GEN2)	⬇	Ground	14	⬆
15	GPIO22 (GPIO_GEN3)	⬇	(GPIO_GEN4) GPIO23	16	⬆
17	3.3v DC Power	⬇	(GPIO_GEN5) GPIO24	18	⬆
19	GPIO10 (SPI_MOSI)	⬇	Ground	20	⬆
21	GPIO09 (SPI_MISO)	⬇	(GPIO_GEN6) GPIO25	22	⬆
23	GPIO11 (SPI_CLK)	⬇	(SPI_CE0_N) GPIO08	24	⬆
25	Ground	⬇	(SPI_CE1_N) GPIO07	26	⬆
<hr/>					
27	ID_SD (PC ID EEPROM)	⬇	(PC ID EEPROM) ID_SC	28	⬆
29	GPIO05	⬇	Ground	30	⬆
31	GPIO06	⬇	GPIO12	32	⬆
33	GPIO13	⬇	Ground	34	⬆
35	GPIO19	⬇	GPIO16	36	⬆
37	GPIO26	⬇	GPIO20	38	⬆
39	Ground	⬇	GPIO21	40	⬆

Rev 1
26/01/2014
http://www.element14.com

FIG. 14: PUERTO GPIO, RASPBERRY PI 3

3.4.2. CONTROLADOR DE MOTORES

Como elemento de control de motores utilizamos los controladores VICTOR 888, que cumple la función de recibir una señal PWM, interpretar esta señal y configurar la velocidad y giro de los motores, el controlador Victor 888 está específicamente diseñado para aplicaciones de robóticas. Lo utilizan mucho en aplicaciones de competencias de robótica donde utilizan carrocería con robot de gran tamaño y peso.

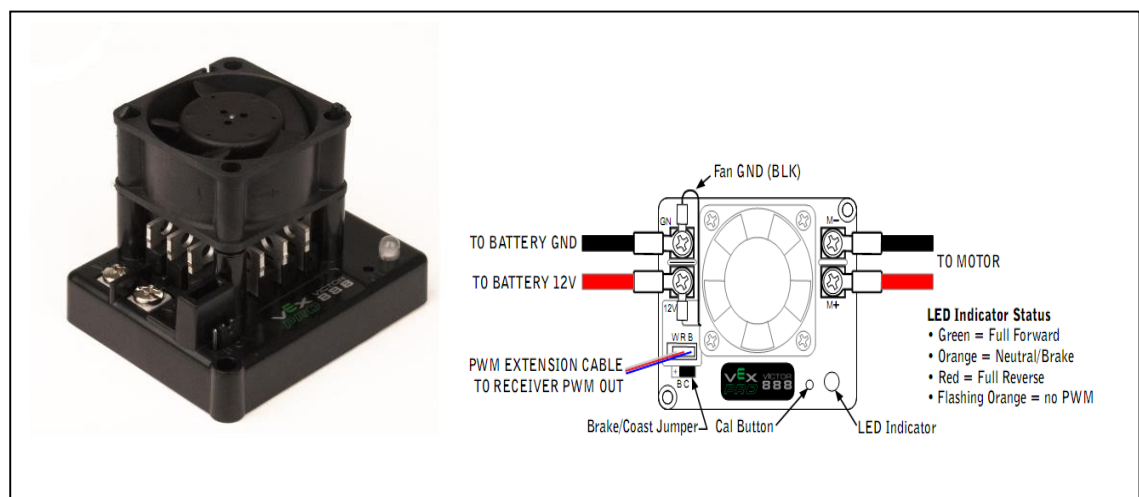


FIG. 15: CONTROLADOR VICTOR 888, VEX ROBOTICS

Su gran capacidad de corriente, baja caída de tensión y máxima capacidad de sobretensión, hacen que el Víctor 888 sea ideal para sistemas de transmisión, mientras que sus opciones control y precisión, satisfacen las exigentes necesidades de la industria, perfectamente podría dar soporte al control de brazos robóticos industriales, sistemas de elevación, sistemas de precisión, entre otros.

En peso y tamaño es muy ligero portátil, además, el Víctor 888 también cuenta con una carcasa para ayudar a proteger los componentes de la placa controladora de los residuos del ambiente.

Características:

- Voltage nominal - 12V
- Min/Max Voltage - 6-15V
- Corriente continua: 60A
- Corriente de sobre tensión (2 sec) - 150A
- PWM pulso de entrada (high time) - 1-2 ms nominal, 0.87-2.14 ms max
- PWM Input Rate (period) - 2.1-500 ms
- PWM tasa de corte de salida - 1 KHz
- Minimum acelerador - 5.4%
- Voltaje ventilador Range - 6V - 16V

Otra característica que me complace mencionar es un puerto I2C de fácil acceso. Este puerto le da al controlador, muchas habilidades potenciales para expandir sus capacidades como, por ejemplo, integrar un sensor de corriente, también se podría utilizar el I2C para conectar en serie varios controladores por medio de un BUS de campo, esto reduciría el costo en cableado y proporcionaría retroalimentación en tiempo real a un microprocesador maestro esclavo.

3.4.3. GEOLOCALIZACION

Se logra gracias al GPS, modulo (GS-15C), ya que es compatible con el protocolo NMEA. El GPS una vez conectado a una interfaz serial mandara cada segundo aproximadamente una serie de comandos siguiendo este protocolo. El usuario debe de diseñar un software capaz de reconocer estos comandos, la Velocidad de transferencia varia en 4800, 9600, 38400, opcional.

GS-15c es un módulo de receptor de satélite inteligente de alto rendimiento y baja potencia, conocido como receptor de satélite, fabricad por MediaTek Inc. Diseñado por una antena receptora de GPS de 32 canales, es un receptor GPS completo con funciones completas.



FIG. 16: MODULO GPS Y ANTENA, (GS-15C)

Características

- Controlador: MTK 3329
- Sensibilidad: -155dB
- Corriente: 30Ma, Después del posicionamiento: 28Ma
- Peso 10.5 gramos
- Voltaje de funcionamiento 5.0 + / - 10% o 3.3 + / - 10% V (VDC)
- Canal de satélite 32 Actualización de información de ubicación 1 segundo
- Tiempo de posicionamiento (promedio) Inicio en caliente: 1 segundo
- Posición de precisión de posicionamiento: 10 metros CEP (CEP)
- Uso de la altitud: menos de 18, 000 metros
- Formato de salida TTL Salida de datos (nmea0183) GPGGA, GPGSA, GPGSV, GPRMC, GPVTG, GPGLL opcional.
- Sistema de coordenadas de la Tierra WGS84

3.4.4. SENSOR DE TEMPETARURA Y HUMEDAD

En caso real, el rastreo y asistencia de víctimas se enfrentan a escenarios con distintas temperaturas y condiciones de humedad, para eso, utilizaremos el módulo sensor DHT11 que nos va a servir para medir tanto la humedad relativa como la temperatura.

Una desventaja, es que el sensor solo nos va a dar medidas enteras, es decir sin decimales, ya que la resolución que presenta es de 1% para la humedad relativa y de 1°C para la temperatura. Pero compensa ya que presenta un valor muy económico para pequeños proyectos y que nos permite obtener medidas tanto para humedad como para temperatura. Para efectos de esta tesis, este módulo sirve, ya que no estamos haciendo un control de precisión de temperatura, solo queremos testear a modo de experimentación, para demostrar si efectivamente funciona e ingresar los valores de la trama por medio de socket.

Este sensor se caracteriza por tener la señal digital, ideal para conectar en los módulos de GPIO de Raspberry y no tendremos problemas con conversión de ADC calibrada por lo que asegura una alta calidad y una fiabilidad a lo largo del tiempo, ya que

Contiene un microcontrolador de 8 bits integrado. Está constituido por dos sensores resistivos (NTC y humedad). Tiene una excelente calidad y una respuesta rápida respecto al testeo. Puede medir la humedad entre el rango 20% – aprox. 95% y la temperatura entre el rango 0°C – 50°C.

Cada sensor DHT11 está estrictamente calibrado en laboratorio, presentando una extrema precisión en la calibración. Los coeficientes de calibración se almacenan como programas en la memoria OTP, que son empleados por el proceso de detección de señal interna del sensor.

El siguiente gráfico, muestra el rango entre voltaje y tiempo en microsegundos que representa el tren de pulsos de la señal digital que llega al controlador para luego ser representado como rango temperatura y humedad.

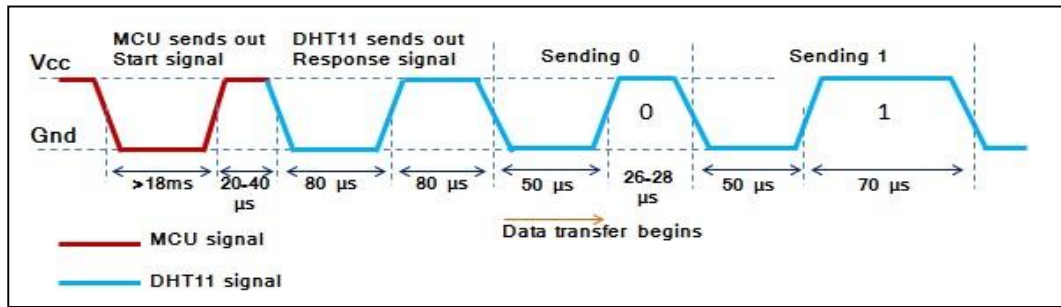


FIG. 17: GRÁFICO DE RANGO TEMPERATURA Y HÚMEDA MEDIANTE TREN DE PULSOS

El protocolo de comunicación es a través de un single-bus (protocolo 1-wire), por lo tanto, hace que la integración de este sensor en nuestros proyectos sea rápida y sencilla. Además, presenta un tamaño reducido, un bajo consumo y la capacidad de transmitir la señal hasta 20 metros de distancia.

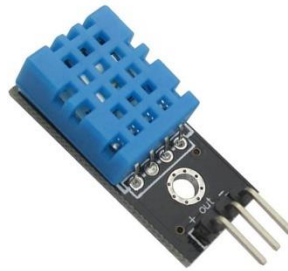


FIG. 18: SENSOR DE TEMPERATURA Y HUMEDAD DTH11

Características:

- Tensión de alimentación de 3 a 5 voltios.
- Corriente máxima de alimentación 2.5 mA.
- Rango de humedad relativa 20% a 80% con 5% de exactitud.
- Rango de temperatura de 0 a 50°C con +-2°C de exactitud.
- Velocidad de 1 medida por segundo.
- Tamaño 15.5mm x 12mm x 5.5mm.
- Señal de salida: digital, vía single-bus

3.4.5. VIDEO CAPTURA

Uno de los objetivos principales planteados al principio de este documento, era poder hacer transmisión de imagen de video en vivo (streaming) para que el usuario pueda identificar víctimas en y reconocer el entorno. Se ocupó una cámara web convencional con las siguientes características:

- Modelo: EC-341
- Cámara: 1.3 Mega pixeles
- sensor CMOS de alta resolución
- micrófono: Integrado
- Formato de color verdadero de 24bits
- Interface: USB de alta velocidad
- Transferencia: 30 cuadros por segundo
- Focos para mejorar iluminación
- Consumo: 5v 20mA

4. DESARROLLO DEL PROTOTIPO

4.1. CARTA GANTT

CARTA GANTT																								
ACIVIDADES	PLAZO																							
	AGOSTO				SEPTIEMBRE				OCTUBRE				NOVIEMBRE				DICIEMBRE							
	s1	s2	s3	s4	s1	s2	s3	s4	s1	s2	s3	s4	s1	s2	s3	s4	s1	s2	s3	s4				
Redaccion resumen documento					X	X																		
Recopilacion informacion, tesis y papers	X	X	X	X																				
Levantamiento informacion terreno			X	X			X	X			X	X					X	X						
Pruebas reconocimiento imagen					X	X	X	X																
Diseño estructura mecanica	X	X	X	X	X	X																		
Diseño y pruebas software, (comunicación)															X	X	X	X						
Diseño y pruebas software, (Interfaz grafica)															X	X	X	X						
Armado estructura prototipo				X	X	X	X	X																
Armado estructura final											X	X	X	X										
Prueba rendimiento motor											X	X	X	X										
Pruebas rendimiento Telemetria									X	X	X	X												
Prueba Rendimiento Bateria				X	X	X	X	X									X	X	X	X				
Prueba rendimiento camara															X	X	X	X						
Prueba Rendimiento Motores													X	X	X									
Pruebas de Rendimiento Material final													X	X	X	X	X	X						

FIG. 19: CARTA GANTT

4.2. DIAGRAMAS

4.2.1. DIAGRAMA DE BLOQUES

El dispositivo de control Raspberry, conjuga todos los elementos conectados a la tarjeta para asignar respectivas tareas por medio de programas y distintos protocolos, Open CV, representa a la biblioteca mediante el cual es posible operar la cámara conectada por USB, para poder visualizar el entorno, GPS se conecta físicamente a un puerto serial habilitado en GPIO de la Raspberry, así también sucede con el sensor DHT y el controlador de motores de corriente continua están conectados a los puertos que ofrece GPIO. Los motores por otra parte, pertenecen a la etapa de potencia del robot, se energizan y operan por medio de los controladores.

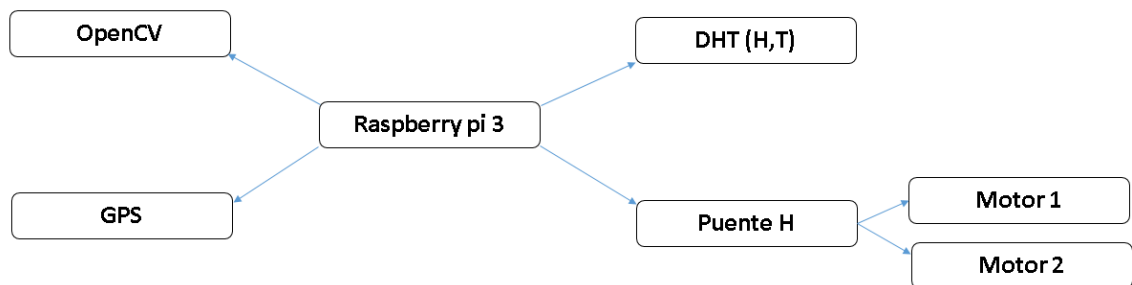


FIG. 20:DIAGRAMA DE BLOQUES

4.2.2. DIAGRAMA SERVIDOR

El servidor cumple con la función de transmitir y recibir información, pero también de controlar todos los enlaces que conectan con los dispositivos del sistema de cámara, lectura de sensores, GPS, y señal de motores. Al iniciar el programa desde el robot, el programa establece el servidor y cumple con la tarea de conectarse por medio de un puerto, esta conexión no es punto a punto con el cliente, sino que, por medio de un Router, al conectarse, comienza a recopilar todos los datos y enviarlos por medio del socket. El modo de tele operación es la interface de usuario que el operador del robot vera desde la computadora.

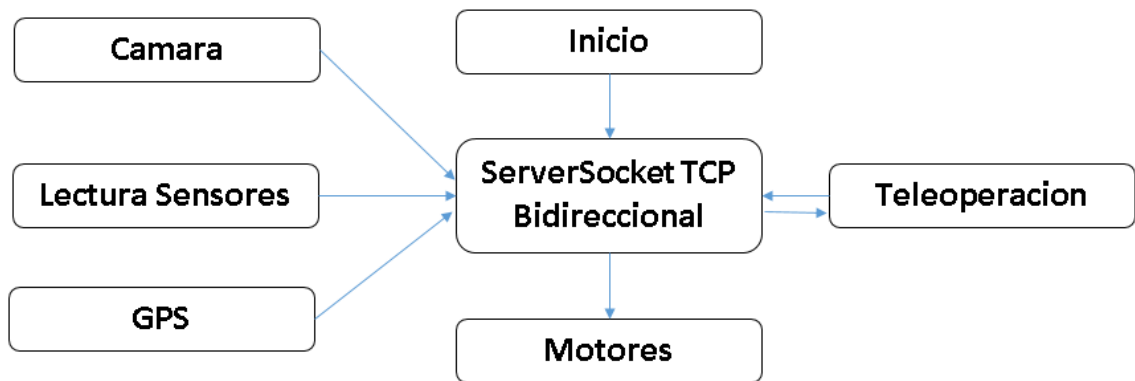


FIG. 21: DIAGRAMA SERVIDOR

4.2.3. MAPA DE CONEXIONES GENERALES

La siguiente imagen es un esquema simple del mapa general de conexión del conjunto del sistema completo, podemos apreciar el conexionado de a red cliente-servidor, por medio de un Router, se aprecia a su vez el resto de dispositivos conectados a la raspberry Pi, elementos tanto de salida como entrada.

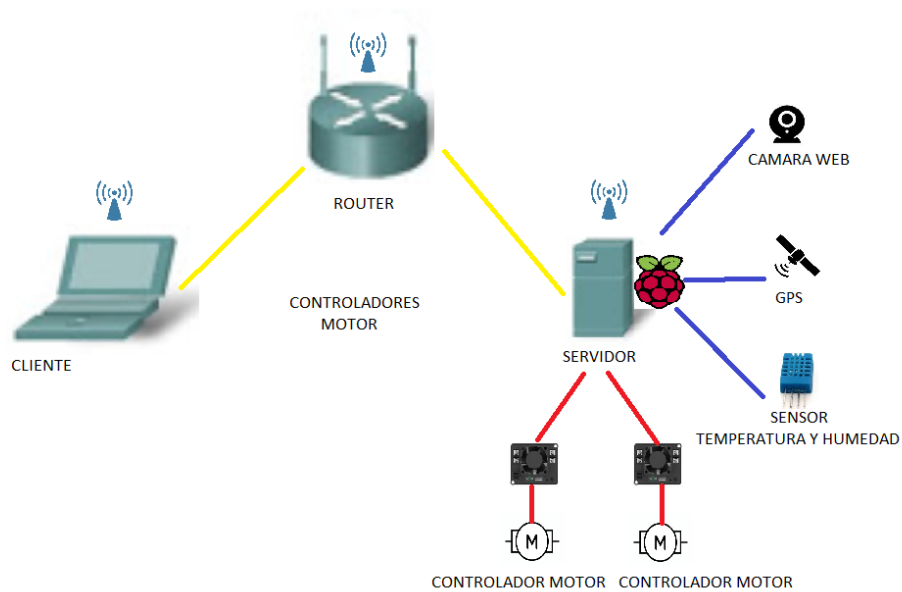


FIG. 22: MAPA DE CONEXIONES GENERALES

4.2.4. DIAGRAMA DE ENERGIZACIÓN Y SEÑALES

La descripción del diagrama comienza por señalar que el circuito se divide en dos etapas, la etapa de potencia, encargada de energizar eléctricamente el sistema y la de control de señales por donde se transmiten datos lógicos.

Se conecta una batería de 12 Vcc 8000 mA que energiza todo el circuito que se vincula entre la Raspberry y el tablero general por medio de un circuito de división de tensión, utilizando el LM 7805, ya que la Raspberry opera con solamente con 5 vcc.

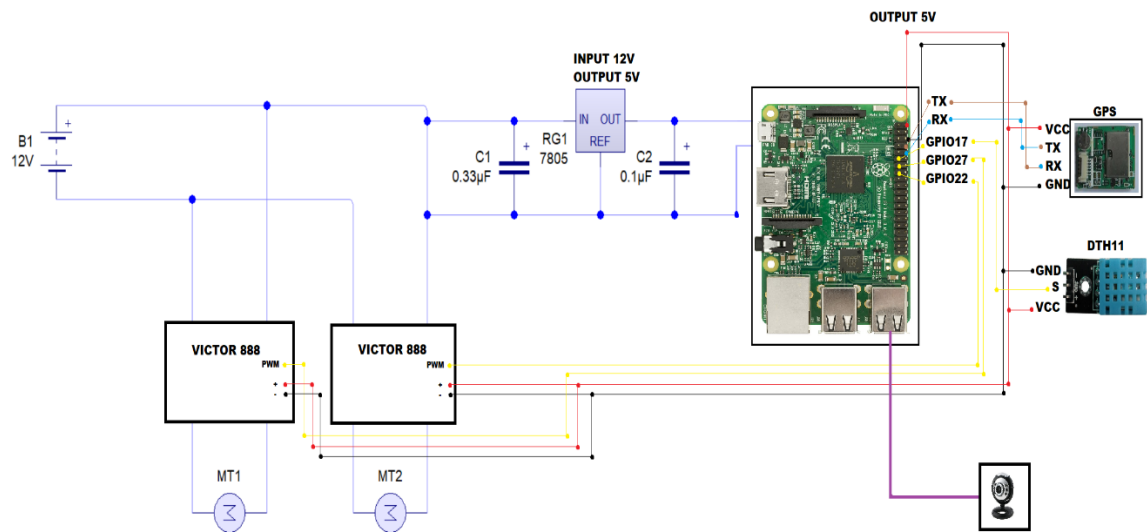


FIG. 23: DIAGRAMA DE ENERGIZACIÓN Y SEÑALES

La etapa de potencia involucra a todos los dispositivos que están conectados directamente a los 12 Vcc, y en parte, la Raspberry, pero considerando el divisor de tensión.

Por otra parte, la otra etapa, es la etapa de señales o de corrientes débiles, está referido a todo el conexionado que está conectado a las entradas y salidas físicas digitales de GPIO de la Raspberry, aquellos componentes son: GPS, sensor DTH11 y señales PWM, también la conexión de la cámara por USB.

En el siguiente cuadro se aprecian los datos aproximados resultantes del muestreo de las mediciones tanto de corrientes como diferencia de potencial.

	Consumo corriente (A)	Voltaje (V)
Victor 888	1,5	12
Motores DC	1,2	12
Raspberry	1	5
Camara	0,02	5
Modulo GPS	0,03	5
Sensor	0,025	5

FIG. 24: TABLA GENERAL DEL CONSUMO EN CORRIENTE Y VOLTAJE.

Calculo de la potencia total en Watt:

$$[\sum (I)] * (Volt) = W$$

$$(3,775) * 12 = 45,3 W$$

4.3. FUENTE DE ENERGIA

Se utilizó como fuente de alimentación, una batería de Polímero de Litio (LiPo), específicamente: “ZIPPY Flightmax 8000mAh 3S1P 30C Lipo Pack”.

La elección de la batería es una de las etapas más importantes cuando se quiere desarrollar un proyecto con una buena proyección y que entregue las mejores prestaciones.

Por esta razón, se usó un criterio de selección basado en los cálculos de la necesidad de autonomía, peso, material, ventajas, desventajas y dimensiones de esta.

Para realizar una correcta elección de la batería, primero debemos considerar La tasa de descarga, más conocida como clasificación C.

Esta clasificación describe la corriente máxima que una batería puede descargar, sin dañarse a sí misma. Aunque estas cifras no son del todo precisas, se consideran buenas para control de calidad. En estricto rigor se habla de la vida útil, entre mayor sea el valor de C, la vida útil de la batería perdurara por más tiempo con respecto a las cargas y descargas de esta.

Las especificaciones AMPER HORA, es el criterio que se utiliza para saber el tiempo durante el cual la batería entrega cierta cantidad de corriente a los dispositivos, para saber esto, se calcula el grado de mAh con la corriente promedio que se gastan los dispositivos.

Capacidad (mAh)	8000
Configuración (s)	3
Descarga (c)	30
Peso (g)	644
Max Índice de carga (C)	5
Longitud - A (mm)	169
Altura – B (mm)	69
Anchura – C (mm)	27

FIG. 25: ESPECIFICACIONES TÉCNICAS DE LA BATERÍA

Para saber el resultado de cuánto tiempo de autonomía tendrá el robot, utilizamos los datos de las tablas de la figura 18 y 19.

De la figura 18, necesitamos el consumo total de corriente, como los dispositivos están conectados en paralelo, las corrientes se suman.

Luego de saber esta suma, utilizamos el dato de “capacidad (mAh)” del recuadro que está en la figura 19. Dividimos ese valor por el valor resultante de la sumatoria anterior.

Conozcamos estos datos por medio de la siguiente ecuación.

$$\text{Capacidad Horas} = \frac{8}{(1,5 + 1,2 + 1 + 0,2 + 0,03 + 0.025)}$$

$$\text{Capacidad Horas} = \frac{8}{3.775}$$

Es decir, según este cálculo la capacidad de la batería es de 2,119 Hrs.

Respecto a la corriente total del circuito. Aproximamos valor decimal a entero.

4.4. CONTROL DE MOTORES.

En el diagrama anterior de energización y señales, se aprecia en la etapa de potencia, que los controladores Víctor 888 están conectados a la fuente de energía directos y también por medio de la señal PWM, esto es porque internamente el circuito integra componentes mosfet, este es un transistor utilizado para amplificar o conmutar señales electrónicas, uno de los más utilizados en la industria.

Estos dispositivos fueron fabricados con el fin de poder utilizarlos con microcontroladores PIC o Arduino, con el fin de dar características industriales pero que se pueda compatibilizar con otros componentes de bajo costo y programar fácilmente para muchas otras aplicaciones.

Los siguientes esquemáticos, muestran las dos etapas del circuito interno del controlador, en la primera figura se aprecia el circuito integrado MEGA 8P, y un MOSFET DRIVER HID4081A que recibe señales desde el integrado MEGA, para enviarlas al motor.

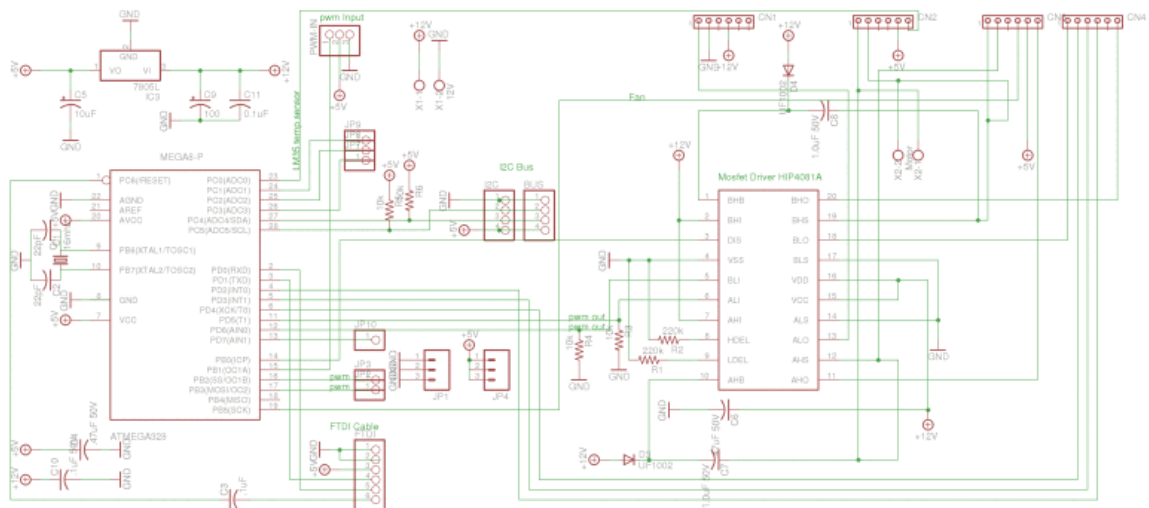


FIG. 26: ESQUEMÁTICO DEL CIRCUITO CONTROLADOR, ETAPA LÓGICA, DEL VÍCTOR 888

5. PRUEBAS Y FUNCIONAMIENTO DEL PROTOTIPO

Para ver el funcionamiento del prototipo en este capítulo se harán distintas pruebas para así conocer los alcances, las fallas y las limitaciones que este tenga.

El desarrollo de las pruebas incluye instrumentos de medición como, por ejemplo, un multi tester, utilizado para medir voltajes y corriente, también se utilizó una pesa para saber el peso en gramos del robot.

Para mantener un mejor orden utilizaremos distintas clasificaciones de los procedimientos realizados.

5.1. PRUEBAS

5.1.1. SOFTWARE

Se ejecutaron distintos programas en la plataforma de Python, para hacer las distintas pruebas para el funcionamiento del robot, de estos se obtuvo resultados y funcionamiento.

La primera Prueba que se realizo fue con el sensor DTH11.

```
pi@raspberrypi ~/$ cd projects $
pi@raspberrypi ~/$ cd projects $ sudo ./dht11
Raspberry Pi wiringPi DHT11 Temperature test program
Humidity = 36.0 % Temperature = 23.0 *C (73.4 *F)
Humidity = 37.0 % Temperature = 23.0 *C (73.4 *F)
Data not good, skip
Humidity = 36.0 % Temperature = 23.0 *C (73.4 *F)
Humidity = 36.0 % Temperature = 23.0 *C (73.4 *F)
Humidity = 37.0 % Temperature = 23.0 *C (73.4 *F)
Humidity = 36.0 % Temperature = 23.0 *C (73.4 *F)
Humidity = 36.0 % Temperature = 23.0 *C (73.4 *F)
Humidity = 36.0 % Temperature = 24.0 *C (75.2 *F)
Humidity = 36.0 % Temperature = 23.0 *C (73.4 *F)
Humidity = 36.0 % Temperature = 23.0 *C (73.4 *F)
Humidity = 37.0 % Temperature = 24.0 *C (75.2 *F)
Data not good, skip
Data not good, skip
Humidity = 37.0 % Temperature = 23.0 *C (73.4 *F)
Humidity = 36.0 % Temperature = 23.0 *C (73.4 *F)
Humidity = 36.0 % Temperature = 24.0 *C (75.2 *F)
```

FIG. 28:RESULTANTE DTH11

Se ejecutó el programa para comandar el robot por medio de la señal PWM,

```
GPIO.setup(27, GPIO.OUT)
```

```
p1=GPIO.PWM(18, 50)
```

```
p1.start(7.5)
```

Con un rango de 0 a 15 grados, frecuencia 50 Hz. Para controlar velocidad y giro de los motores se hizo el siguiente arreglo:

```
try:
```

```
    while True:
```

```
        h=int(input("8:adelante,2:atraz,4:izquierda,6:derecha,5:parar"))
```

```
            if h==8:
```

```
                p1.ChangeDutyCycle(2.5)
```

```
                p2.ChangeDutyCycle(2.5)
```

Se logró establecer comunicación bidireccional por medio de socket, utilizando dos lenguajes diferentes: Python y LabView.

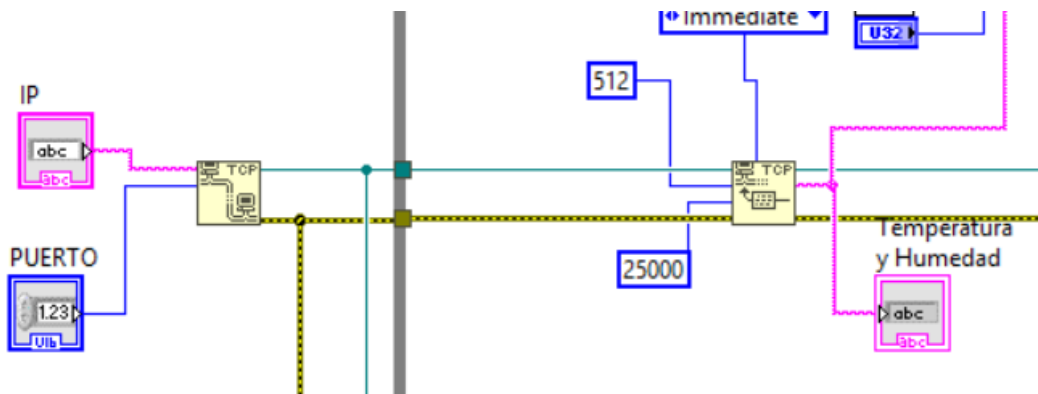


FIG. 29: CAPTURA DE CÓDIGO SOCKET LABVIEW

Luego, se realizó pruebas utilizando el módulo GPS, en primera instancia, se importa la librería pynmea2, que es para obtener la trama de GPS, luego el resultante de la trama permite visualizar la longitud, latitud, y los hemisferios del sistema de coordenadas planetario.

Los datos que se reciben, aun no tiene valor matemático, con el comando Split, se logra separar por (",") la trama del GPS y luego mostramos el resultado en pantalla:

```
print "http://maps.google.com/?q="+str(msg.latitude)+","+str(msg.longitude).
```

Cuando obtenemos el resultado, enviamos ese resultado por medio del socket ya establecido, para que LabView pueda dirigirse al sitio web de Google, y así establecer el punto exacto donde se posiciona el robot.

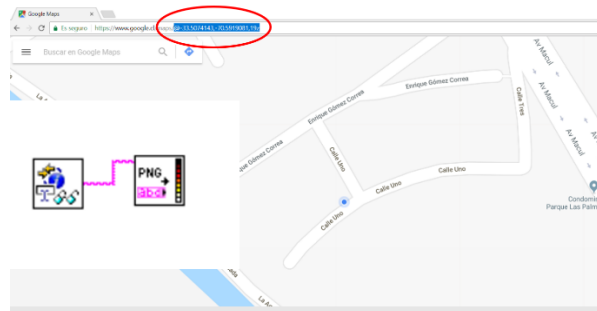


FIG. 30: CAPTURA DE LOCALIZACIÓN GOOGLE-MAPS Y COMANDOS LABVIEW.

Luego compilamos otro programa, esta vez, se realizó la prueba de monitoreo con la WebCam, conectada por puerto usb en la Raspberry, antes de ejecutar cualquier código debemos descargar e instalar OPEN CV, y también instalar MOTION, con el siguiente comando:

```
sudo apt-get install motion
```

Al instalar el programa por medio de la consola, se debe configurar el ancho de las imágenes, la altura, la velocidad de los fotogramas y configuraciones para la detección de movimiento.

Para poder transmitir el video y lograr el streaming, dentro de la red, se necesita montar un HOST, esto se logra a partir del siguiente comando:

```
app.run(host= '#asignar ip, por ejemplo 192.0.0.1', debug=True)
```

Luego el servicio de cliente deberá tomar la trama y abrir el link (ip), y abrir una ventana capturando la imagen para que el usuario reconozca el entorno en que se encuentra el robot.

Se montó el sistema de adquisición de datos recopilando distinta información de los instrumentos montados en el manipulador robótico, se recopiló datos de, temperatura, humedad, geolocalización, captura de imagen y transmisión en tiempo real, la próxima etapa de prueba, fue establecer la “driver-station” para poder enviar las señales para el movimiento de motores por medio de un Joystick, dentro de las libretas de LabView, tenemos varias opciones de configuración de botones y asignación de TAGs, como referencia, se utilizó la simulación de un mando control de la consola para video juegos de Xbox360.



FIG. 31: SIMULACIÓN DE MANDO CONTROL XBOX360. LABVIEW

La última prueba de software es compilar todo en un solo código y ejecutar la interfaz gráfica, se tomó la decisión de importar todos los códigos de las pruebas que se realizaron con el comando `from (nombre programa) import`.

De esta manera se puede establecer un mejor orden, y se puede realizar análisis de forma individual.

En el siguiente párrafo se muestra la manera en que importamos todos los programas `python.py` desde la carpeta `TESIS` para luego ejecutar y comunicar por medio del `socket`.

```
from DTH11_TESIS import TESIS
from DRIVER_TESIS import TESIS
from STREAMING_TESIS import TESIS
from GPS import TESIS
```

Por otra parte, se monta la interfaz gráfica, se analizan los componentes y cumple con lo pre establecido, en una pantalla muestra la localización GPS, y en la otra la captura de la cámara web, se configuraron los botones X,Y,B, A simulando el control de Xbox, la señal de temperatura y humedad se puede apreciar en el costado derecho, el dato de humedad y temperatura cambia cada 1 segundo y lo muestra en pantalla.

El programa tiene 2 botones de parada, uno es para parar el programa y que no ejecute nada, y el otro es para detener los motores en caso de emergencia, pero que no deje de transmitir el resto de los datos.

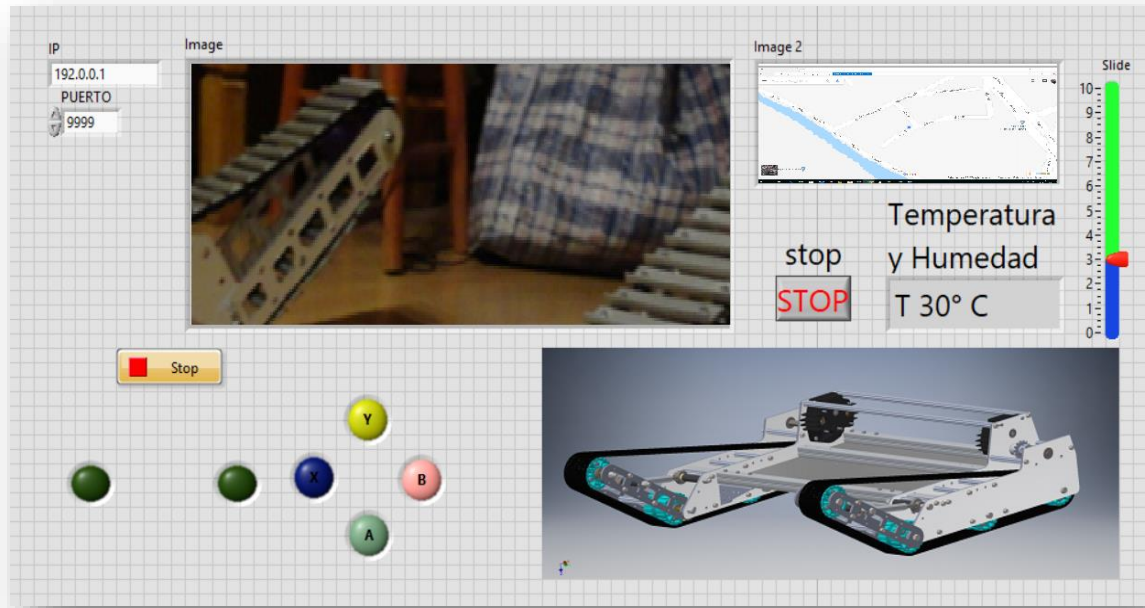


FIG. 32: INTERFAZ GRÁFICA. LABVIEW

5.1.2. LOCOMOCIÓN

El diseño y armado del prototipo comienza al hacer las primeras pruebas de diseño 3D, basado en el diseño mencionado en el capítulo 2 “Outdoor Rhino Track Drive (am-3446)”, se utiliza el programa Inventor, de Autodesk.

Se trata de un paquete de modelado paramétrico de sólidos en 3D producido por la empresa de software Autodesk.

Software que se ocupa mucho en el diseño industrial para fabricación de piezas mecánicas.

A continuación, se ilustran algunas figuras del prototipo realizado, para después hacer el ensamble de piezas y someterlo a distintas pruebas en terreno.

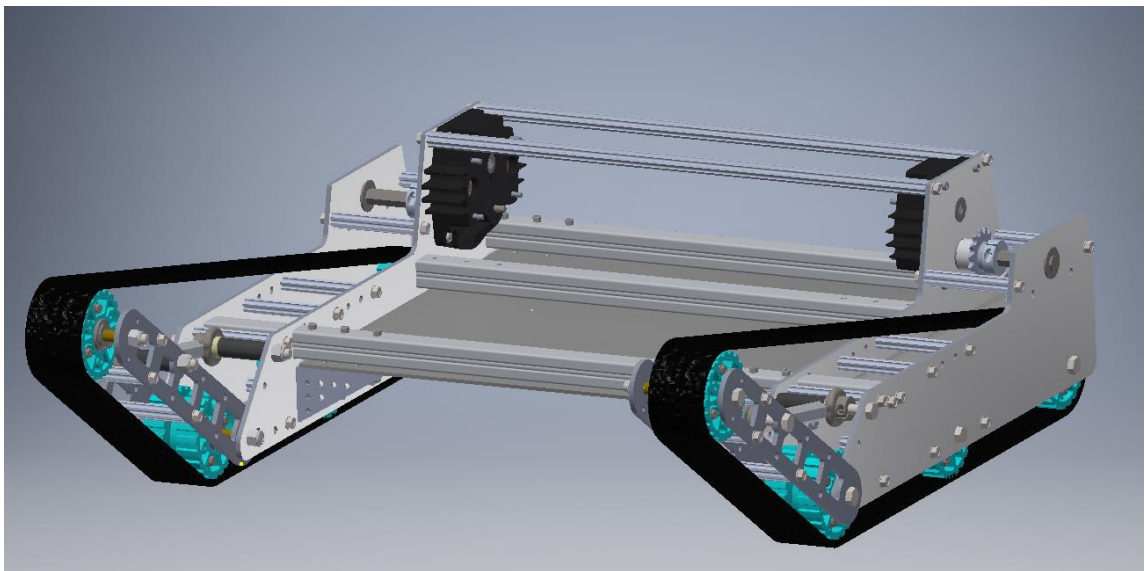


FIG. 33: VISTA FRONTAL DISEÑO 3D

Los materiales utilizados para la estructura mecánica son aluminio compuesto de 4 mm de grosor, las placas de aluminio compuesto se forman a partir de un núcleo de polímero, este composite es rígido, resistente a los golpes, a la rotura, presión, al doblado, y a altas temperaturas. Estas características de alta

resistencia se la confieren las láminas de aluminio en PERALUMAN 100 (A1 Mg1) que lo recubren, cuyas características son:

Resistencia a la tracción: $R_m \geq 130 \text{ N/mm}^2$

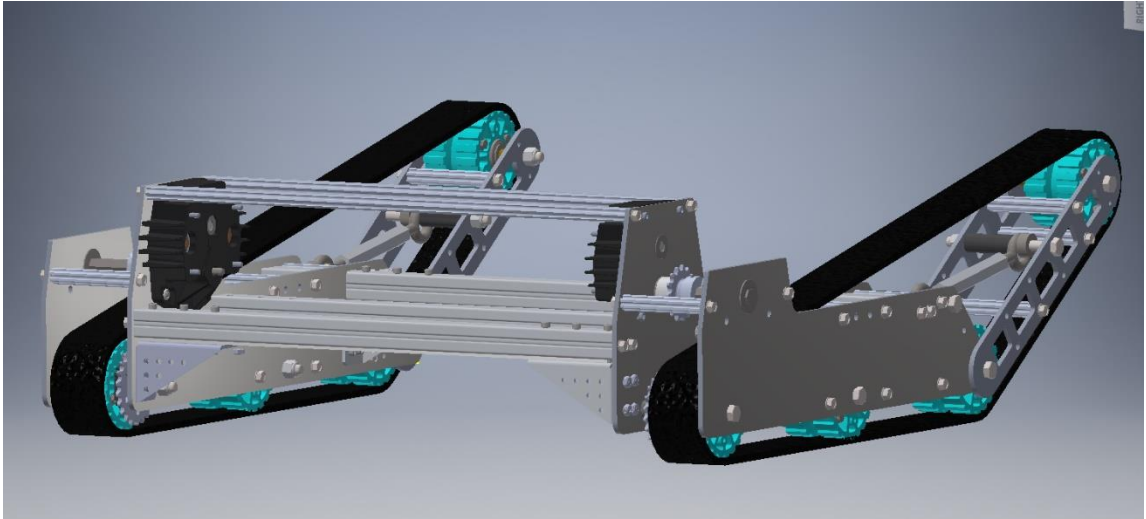


FIG. 34: VISTA POSTERIOR DISEÑO 3D

Otro material utilizado es el Technyl, utilizado en la fabricación de la polea principal que se acopla al motor para ejercer el principal punto de fuerza entre el eje de motor y la copla este material se encuentra con frecuencia en la industria, debido a sus características resistentes, esta poliamida de características termoplásticas se diferencia de los demás polímeros por sus excelentes propiedades mecánicas, dieléctricas, técnicas y químicas. Siendo este plástico de ingeniería un material universal para la fabricación de componentes mecánicos e industriales.

Se sometió este material a realizar las siguientes pruebas:

- Fácil de mecanizar, se hicieron distintas modificaciones entorno al diseño
- La estabilidad térmica de este material permitió soportar temperaturas de 10°C a 100°C en forma continua
- Su superficie puede ser fresada, calada o pulida para conexión de piezas.
- Posee gran resistencia a los golpes

- Excelente resistencia mecánica, rigidez, tenacidad y dureza
- Buena resistencia a la fatiga
- Excelente resistencia al desgaste

Se utilizó el método de impresión plástico PLA (ácido poliláctico o poliácido láctico) para el resto de las poleas, con el fin de reducir peso en la carrocería ya que el relleno del material es mucho más liviano que el Technyl, las poleas que no van conectadas a ningún motor, no cumplen otra función más que guiar la cinta metálica para mantener una posición lineal y en paralelo en ambos rieles.

Se sometió solamente a la prueba que respecta a la función que cumple en el prototipo.

Característica	PLA DEL CDP	PLA de Kuraray	Rayón	Lyocell	Poliéster	
Densidad (g/cm ³)	1.25	1.27	1.51	1.52	1.38	
Punto de fusión (°C)	120-170	170	Ninguno	Ninguno	260	
Cuenta (dtex)	1.7 (e.g.)	1.7	1.7	1.7	1.7	
Tenacidad (cN/tex)		50	25-45	20-25	40-45	35-65
Alargamiento (%)		35	N/A	18-22	14-16	15-40
Recobro de la humedad (%)	0.4 - 0.6	0.48	13	12	0.4	

FIG. 35: CUADRO CARACTERÍSTICAS PLA

Las correas que se usaron para las ruedas fueron fabricadas por un sistema intercalando perfiles de aluminio y cadena para piñón.

En un principio se fabricó una pieza con piñones diseñadas para guiar la cadena, así como el sistema de una bicicleta, pero no fue necesaria la implementación ya que el Technyl cumplía con las prestaciones y gracias a su alto roce y diseño de la pieza tubo la suficiente fuerza para trasladar el robot.

El principio de funcionamiento de este sistema es basado en cintas transportadoras industriales.

La combinación de uso de polímeros y compuestos de aluminio nos dan una garantía confiable y razonable para los fines en que ocupamos el material, la mayoría de estos compuestos usados en la industria aeronáutica.

Se realizaron pruebas de resistencia de material con cada uno de los compuestos anteriormente señalado, se expuso el robot en terreno real, áreas pedregosas, pasto, concreto, gravilla y tierra. Dando resultados positivos.

La siguiente imagen representa el interior de una de las ruedas orugas donde se aprecia la cadena guiada entremedio de una polea impresa con plástico PLA.

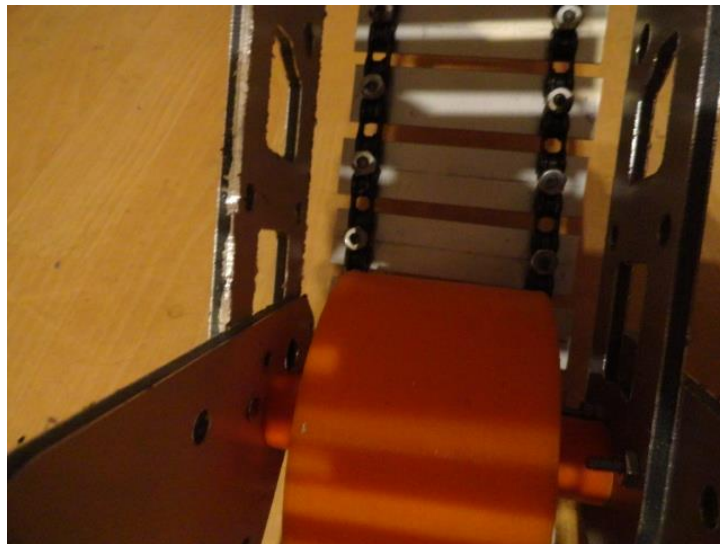


FIG. 36: SISTEMA DE RUEDA GUIA

La siguiente prueba realizada se hizo en relación con el peso total del robot, las dimensiones, el sistema de transmisión de fuerzas y las energías involucradas.

A continuación, algunos datos ilustrados para entender la experimentación.

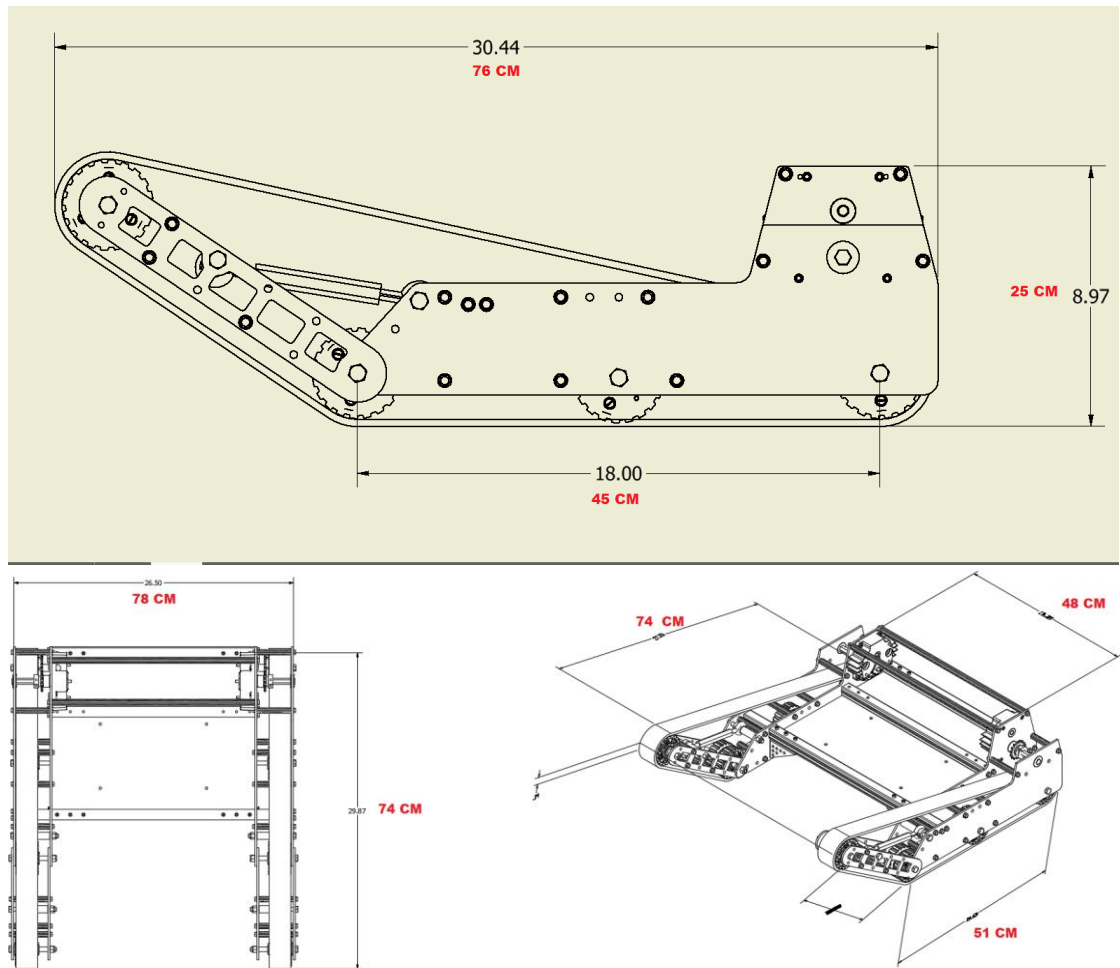


FIG. 37: VISTA GENERAL MEDIDAS FORMATO DWG

Peso Total del sistema 25Kg.

En la figura podemos ver las medidas en cm (las que están en rojo) y las que no, están en pulgadas, pero en general las medidas importantes del sistema son diámetro de 68cm x 75cm x 24cm (profundidad, ancho, altura),

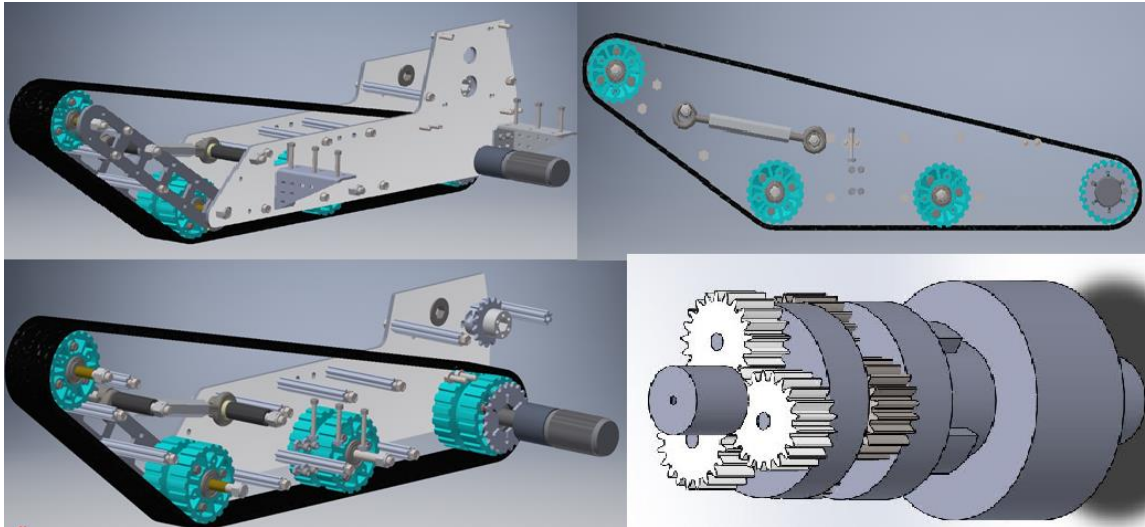


FIG. 38: VISTA GENERAL DEL INTERIOR DEL MECANISMO DE TRASLACIÓN

En esta imagen podemos analizar de mejor manera la mecánica del sistema de traslación. La principal fuerza proviene de la fuente de energía provocada por la batería, la energía eléctrica se transfiere al motor y por medio de la fuerza electromotriz y principios electromagnéticos el eje del motor gira hacia algún sentido, la característica principal del motor seleccionado para el desarrollo es que precisamente implementa un sistema de engranajes planetarios, luego esa fuerza se traslada hacia el sistema de poleas.

La prueba a realizar, consistía en forzar a que el motor nos consuma el máximo amperaje en relación a una fuerza, hasta frenar el motor y realizar las mediciones con un amperímetro, se ingresan los datos a una tabla y se plantea un gráfico relación peso y consumo para saber el máximo peso que puede aguantar el robot.

Un engranaje planetario o engranaje epicicloidial, es un sistema de engranajes externos que giran alrededor del eje principal del motor, simulando el sistema solar y los planetas que giran alrededor del sol.

Los sistemas de engranajes planetarios pueden incorporar también el uso de un engranaje anular externo o corona, que engrana con los demás.

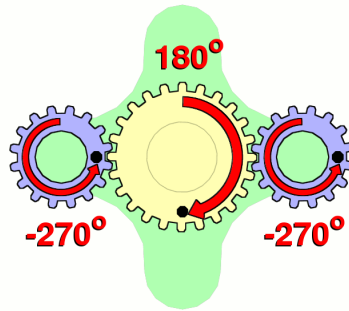


FIG. 39: SISTEMA ENGRANAJE EPICICLOIDAL

El modelo del motor que ocupamos para esta prueba es un motor DC con engranaje 50 kg-cm Modelo: GMP42-TEC4260

Sistema de reducción planetaria,

DC motor con engranaje 50 kg-cm Modelo: GMP42-TEC4260

Descripción

- Torque: 1.1-90kg.cm
- Potencia salida: 6-19W
- Marca: MOTOR TT
- Velocidad: 600rpm
- Voltaje: 6-24V
- Modelo: GMP42-TEC4260
- Consumo de corriente: 450-1200Ma
- Eficiencia: 80%

6. ANALISIS DE CAMPO Y RESULTADOS

6.1. PRUEBAS

En esta sección se muestran los resultados de algunas pruebas hechas en los capítulos anteriores, hacemos un análisis para saber la viabilidad de este.

En este módulo se da a conocer los resultados de las pruebas hechas en los otros capítulos, mediante el cual utilizaremos esos datos para sacar conclusiones respecto a esta tesis.

Uno de los datos más influyente es el torque, ya que es la potencia que puede ejercer el motor.

Es lo que va a definir si los motores van a ser capaces de mover a nuestro robot. Se suele medir en kg/cm; 1kg/cm significa que el motor puede mover hasta 1kg a 1cm del centro de rotación, es decir, que con una rueda de 1cm de radio se podría mover a lo menos de 1kg de peso.

6.1.1. PRUEBAS Y VALIDACIÓN

Los valores adquiridos de las pruebas en campo y mediciones se resumen en:

- Voltaje total = 12 Vcc
- Consumo amperaje = 3.775 A
- Capacidad de la batería = 2 Hrs de autonomía
- Torque motor: Torque: 1.1-90kg.cm
- Peso total del motor: 700g

Se registraron los valores de la prueba de torque que se realizó aplicando fuerza al motor de aproximadamente 100 Kg y estos fueron los resultados de lo que pasaba con la corriente.

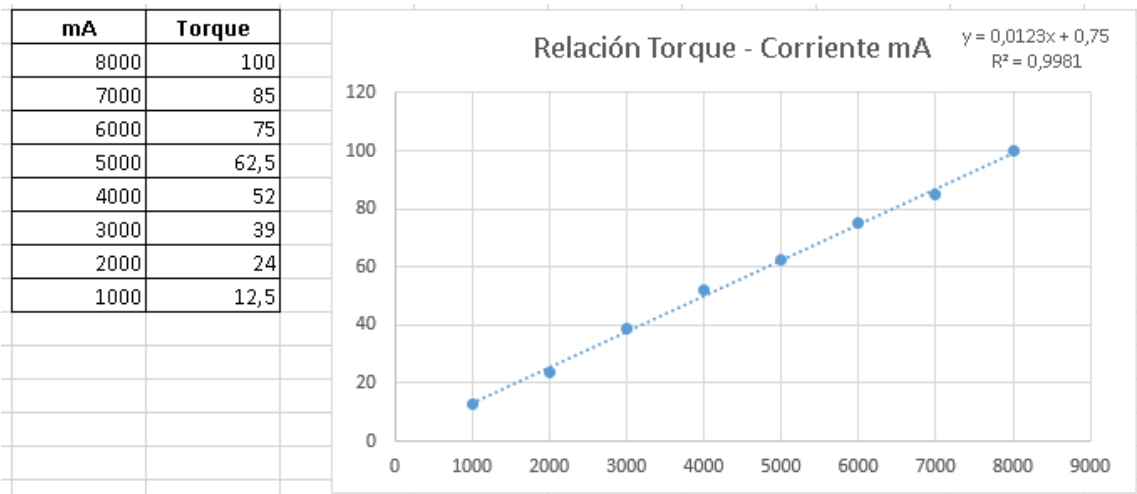


FIG. 40: CORRELACIÓN TORQUE - CORRIENTE

6.1.2. COMPARACIÓN

Comparado con otros sistemas, válidamente se encuentran ventajas y desventajas respecto a factor económico, sustentos de algún sponsor asociado, funcionamiento, estructuras, códigos, entre otros.

Si realizamos un prototipo a modo experimental solo para investigar, no siempre se realizan grandes inversiones, si volvemos al capítulo 2 en el estado del arte y comparamos al Robot CHIMP de Darpa junto con el *iRobot 510* , en esos sistemas si hay una verdadera inversión ya que son de uso militar.

Esta tesis pese a ser un desarrollo prototipo experimental, la estructura otorga un soporte fiable, el costo de reparación o reemplazo de alguna etapa, es sustentable, se puede cubrir los costos de reparación, eso permite asumir más riesgos en la práctica, las compañías no arriesgarían equipos en casos que no sean extremos.

Como es modular además, permite futuras aplicaciones y cambios en la estructura mecánica, y actualización de software, también se puede hacer una red con distintos dispositivos como también agregar otras prestaciones, ya que, al ser, un modelo a escala real, y contempla un buen espacio estructural, podría alojar suministros, también se podría montar un actuador tipo brazo robotizado, extinguidores de incendios, pala mecánica para retiro de escombros, retiro de tierra o nieve.

6.1.3. CONCLUSIÓN Y DISCUSIÓN

En el transcurso de desarrollo del proyecto, gracias al aprendizaje obtenido, pude darme cuenta de que, se originaron soluciones simples ante problemas difíciles.

Este sistema tiene muchas ventajas, pero también desventajas, una de esas es que sus tareas son limitadas, solamente tiene la tracción para movilizarse en el ambiente y nada más, en resumen, en pruebas de campo, solo sirve para rastrear, identificar y suministrar provisiones de emergencia a víctimas.

Otra desventaja se identifica en el código, no se integró sistema de triangulación en caso de pérdida de ambas señales, (GPS y WIFI), al depender de la comunicación, el robot, con algunas modificaciones en el código, podría integrar un sistema de triangulación, en base a las distancias en metros entre el Router y la antena receptora de wifi, haciendo una correlación entre dbm y metros. Pero aun tendríamos problemas en el caso de perder señal, aun que tuviéramos muchas Antenas repetidoras ¿Qué pasaría si todas las redes caen?

El sistema fue desarrollado bajo OPEN SOURCE al ser un código abierto, permite compatibilizar con otras consolas, y sistemas Linux. Es una gran ventaja ya que, como solución ante las desventajas identificadas, se podría implementar un sistema de navegación SLAM.

En la industria existe mucho avance con robots móviles y el desarrollo de estos está en boga. Para distintas aplicaciones y prestaciones, con valores comerciales excedidos y sistemas operativos, con softwares limitados o valores monetarios elevados.

Machining Learning, es lo que se está aplicando en los actuales doctorados, procesamiento y toma de decisiones en base a Inteligencia Artificial.

Este desarrollo permite la comunicación entre dispositivos, está abierta la posibilidad de aplicar inteligencia artificial en este sistema si se requiere.

Es factible aplicar tecnologías robustas para avanzar en el terreno de rescate de personas, los desarrolladores de estas tecnologías generalmente para uso militar o privado. Si bien este sistema tiene algunos déficits, son problemas fáciles y rápidos de resolver.

La mayoría de las tecnologías, protocolos, componentes, y el desarrollo estudiados en esta tesis, son de uso militar y ya han sido probadas, espero que el desarrollo de esta tesis se pueda aplicar en una red de robots de asistencia para salvar vidas y no para atentar contra ella.

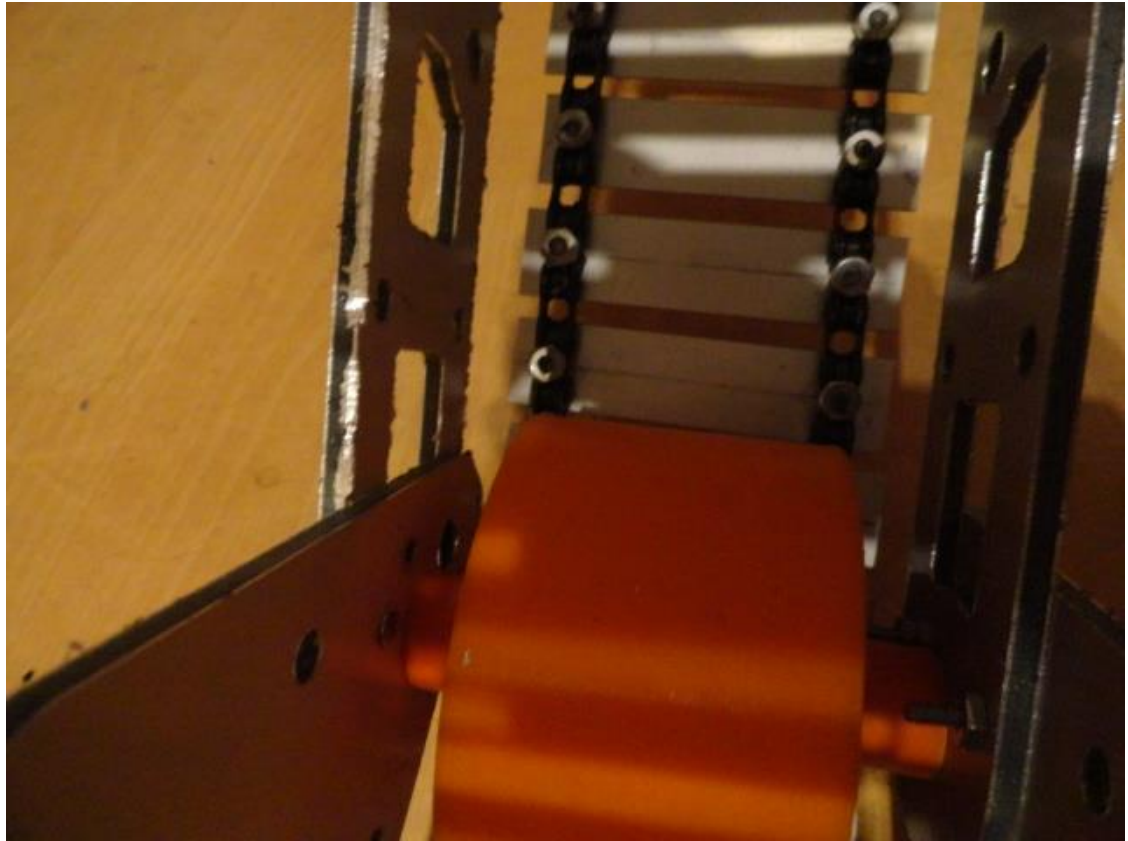
7. BIBLIOGRAFÍA

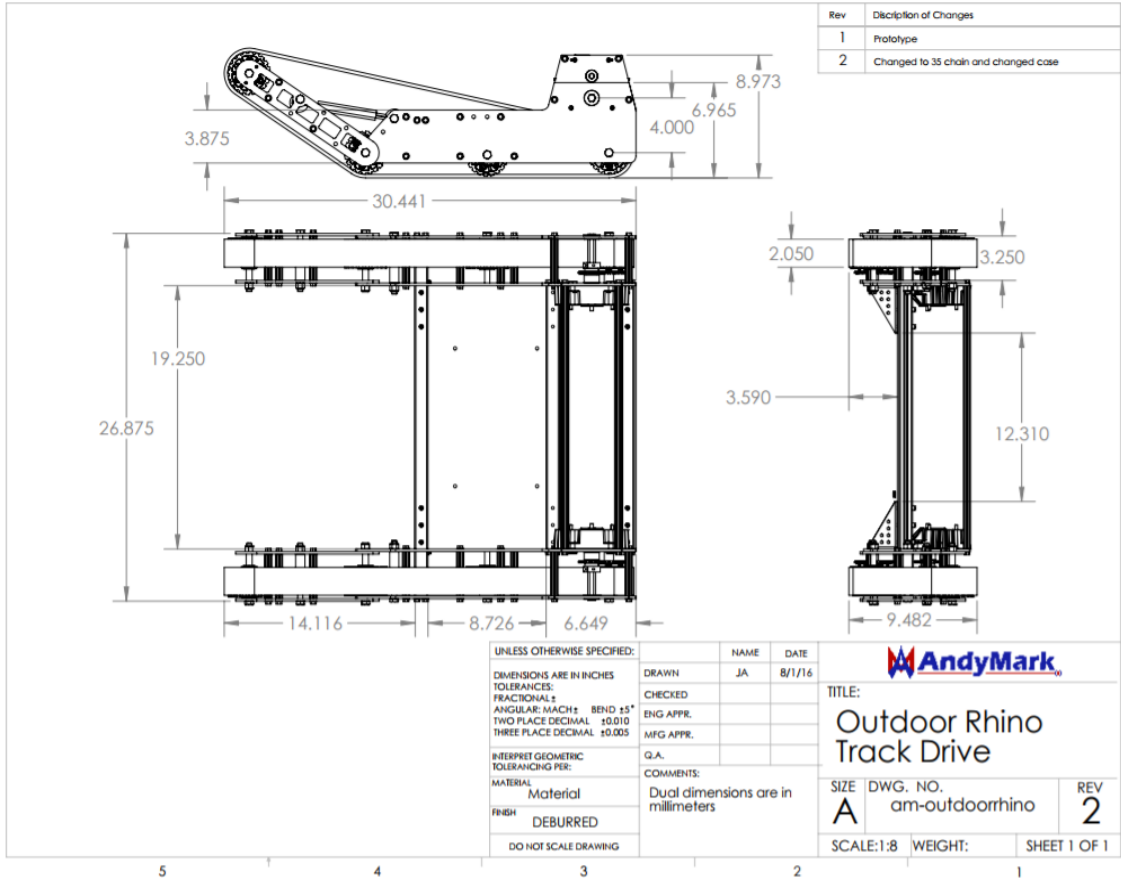
- 8373, I. (2012). *ISO 8373*. International Organization for Standardization.
- AFNOR. (1926). *Association française de Normalisation*.
- André Araújo, D. P. (s.f.). *Desarrollo de un robot móvil compacto integrado en el middleware ROS*. Portugal 2014.
- AndyMark. (enero de 2015). *www.andymark.com*. Obtenido de <https://www.andymark.com/Outdoor-Rhino-p/am-3446.htm>
- ARMY Technology. (2011). *iRobot 510 PackBot Multi-Mission Robot*. Obtenido de <http://www.army-technology.com/projects/irobot-510-packbot-multi-mission-robot/>
- Defense Advanced Research Projects Agency, DARPA. (2015). *Robot Finishes Third In DARPA Robotics Challenge*. Pittsburgh, PA: Carnegie Mellon University.
- Federal Emergency Management Agency FEMA. (2017). *Actualización sobre las operaciones de respuesta y rescate del huracán María*. L.A. California.: www.fema.gov.
- G. Sanahuja a, A. V.-V. (2011). *Control Embebido de Robots Móviles con Recursos Limitados Basado en Flujo Óptico*.
- J.F. Guerrero-Castellanos, , M.-C.-S.-M. (2014). *Seguimiento de trayectorias de un robot m´ovil (3,0) mediante control acotado*. Puebla, Mexico.
- Jesús M. García a, *. I.-C. (2017). *Robot Móvil dotado de Brazo para Contacto con el Suelo*. San Cristobal, Venezuela: Revista Iberoamericana de Automática e Informática industrial.
- Join GitHub today. (13 de OCTUBRE de 2015). *github.com*. Obtenido de <https://github.com/Knio/pynmea2>
- LabVIEW - National Instruments. (2016). <http://www.ni.com>. Obtenido de <http://www.ni.com/es-cl/shop/labview/buy-labview.html>
- Llofriu, M. (2000). *Estudio del estado del arte del SLAM*. Montevideo: Uruguay.
- Messina, E. (2006). Normas de rendimiento para robots para búsqueda y rescate urbano. *American Society of Testing Materials*, www.astm.org.
- Noving, S. R. (2003). Artificial intelligence, 2da edition. En S. R. Noving, *Artificial intelligence, 2da edition* (pág. 23). Englewood Cliffs , NJ: A modern Approach.
- ONEMI, O. N. (2010). CHILE.

- P. A. Niño-Suárez, E. Aranda-Bricaire, M. Velasco-Villa. (2007). *CONTROL MEDIANTE MODOS DESLIZANTES*. MEXICO: CINVESTAV-IPN.
- Ramón González, *. F. (2015). *PAPER Robots Móviles con Orugas*. Almeria, España: Revista Iberoamericana de Automática e Informática industrial.
- Yaveni, B.-C. E.-B. (2007). *Control visual para la formación de robots móviles tipo unicycle bajo*. Instituto Tecnológico de Ensenada, B.C.

8. ANEXOS



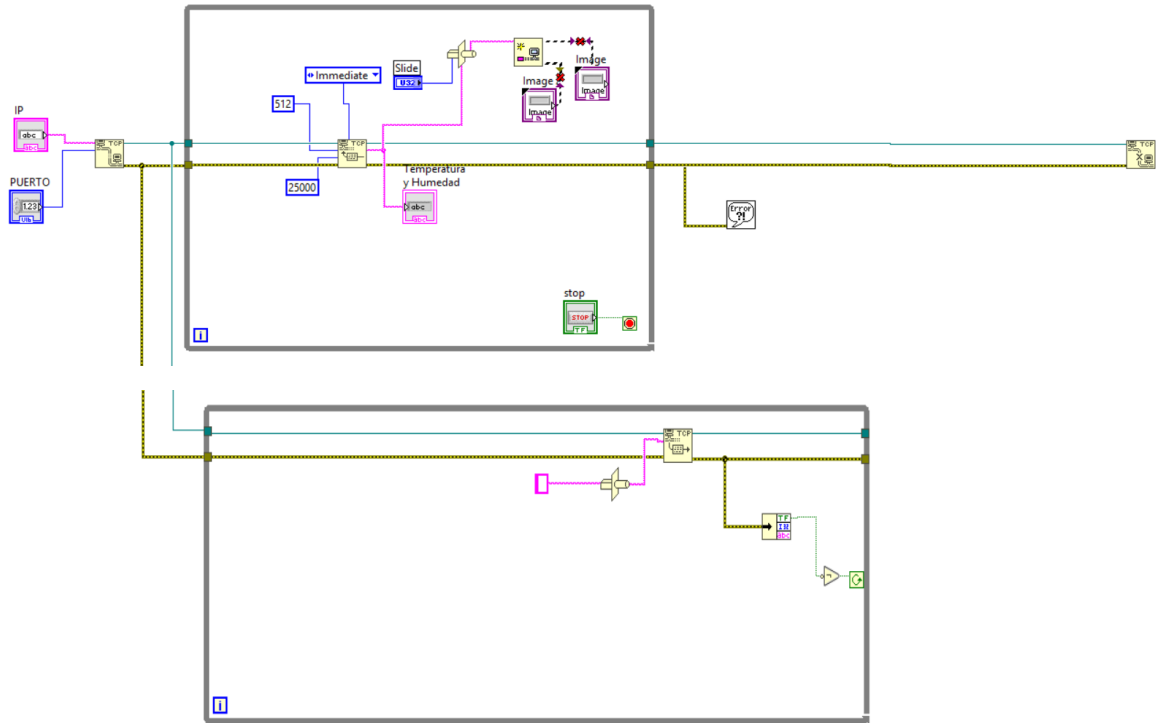




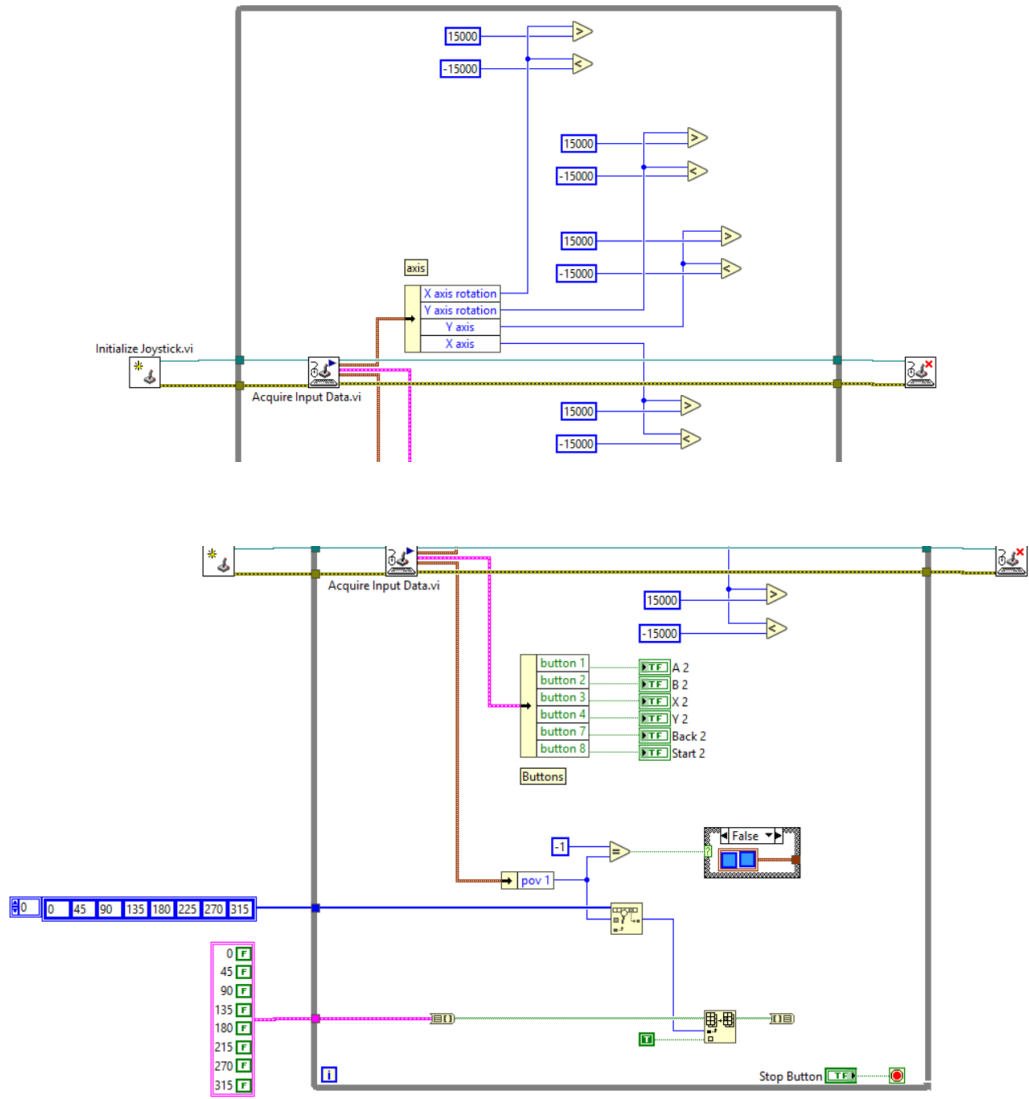


9. CODIGOS LABVIEW

COMUNICACIÓN TCP/IP



COMANDOS DRIVER STATION



(LabVIEW - National Instruments, 2016).

10.CODIGOS PYTHON

Servidor

```
import socket, threading

class ClientThread(threading.Thread):

    def __init__(self,ip,port):
        threading.Thread.__init__(self)
        self.ip = ip
        self.port = port
        print "[+] Nueva conexion aceptada para "+ip+": "+str(port)

    def run(self):
        print "Conexion desde : "+ip+": "+str(port)

        clientsock.send("CONEXION OK\r\n")
        respuesta=bytearray([0x4F, 0x4B, 0x0D, 0x0A]) #OK [enter]
        clientsock.send(respuesta)

        dato_in=" "
        while len(dato_in):
            dato_in = clientsock.recv(2048)
            print ip+" C->S: ASCII:"+dato_in+" " HEX:"+
            ":".join("{:02x}".format(ord(c)) for c in dato_in)

            ##PROCESAMIENTO DE DATOS
            if dato_in[:1]==bytearray([0x40]): #si el primer byte es un "@"
                print "INI"

                print "***"+dato_in[1:3]+"***"
                if dato_in[1:3]==bytearray([0x41,0x42]):
```

```

        print "CMD1"

        if dato_in[1:3]==bytearray([0x43,0x44]):
            print "CMD2"

        dato_out=bytearray([0x41, 0x43, 0x4B, 0x0D, 0x0A])
        clientsock.send(dato_out)

    print "Cliente desconectado..."

host = "0.0.0.0"
port = 9999

tcpsock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
tcpsock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

tcpsock.bind((host,port))
threads = []

while True:
    tcpsock.listen(4)
    print "\nEsperando conexiones..."
    (clientsock, (ip, port)) = tcpsock.accept()
    newthread = ClientThread(ip, port)
    newthread.start()
    threads.append(newthread)

for t in threads:
    t.join()

```

SENSOR

```
import RPi.GPIO as GPIO

import dht11

import time

import datetime

# initialize GPIO

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.cleanup()

# read data using pin 17( 11 en realidad )

instance = dht11.DHT11(pin=17)

while True:

    result = instance.read()

    if result.is_valid():

        print("Last valid input: " + str(datetime.datetime.now()))

        print("Temperature: %d C" % result.temperature)

        print("Temperature: %d F" % (( result.temperature * 9/5)+32 ))

        print("Humidity: %d %" % result.humidity)

    time.sleep(1)

GPS

import serial

import pynmea2

ser=serial.Serial("/dev/ttyS0",115200)
```

```

while (1):

    trama=ser.readline()

    #print(trama)

    dato=trama.split(",")

    #print dato[0]

    if dato[0]=="$GPRMC":

        msg=pynmea2.parse(trama)

        print
        "http://maps.google.com/?q="+str(msg.latitude)+","+str(msg.longitude)

        lat=dato[3]

        NS=dato[4]

        lon=dato[5]

        WE=dato[6]

        #print "LAT: "+lat+" "+NS+" LON:"+lon+" "+WE

CONDUCCION ROBOT (DRIVER)

import RPi.GPIO as GPIO

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

#Motor 1

GPIO.setup(27, GPIO.OUT)

p1=GPIO.PWM(18, 50)

p1.start(7.5)

```



```

#Motor 2
GPIO.setup(22, GPIO.OUT)
p2=GPIO.PWM(17, 50)
p2.start(7.5)

try:
    while True:
        h=int(input("8:adelante,2:atraz,4:izquierda,6:derecha,5:parar"))
        if h==5:
            p1.ChangeDutyCycle(0)
            p2.ChangeDutyCycle(0)
            print ("stop")
        elif h==8:
            p1.ChangeDutyCycle(3)
            p2.ChangeDutyCycle(3)
            print ("adelante")
        elif h==2:
            p1.ChangeDutyCycle(12)
            p2.ChangeDutyCycle(12)
            print ("atraz")
        elif h==4:
            p1.ChangeDutyCycle(6)
            p2.ChangeDutyCycle(9)
            print ("izquierda")
        elif h==6:
            p1.ChangeDutyCycle(9)
            p2.ChangeDutyCycle(6)
            print ("derecha")

except KeyboardInterrupt:
    print ("Exit")
    GPIO.cleanup()

```

OPEN CV

```
import cv

cv.NamedWindow("w1", cv.CV_WINDOW_AUTOSIZE)

capture = cv.CaptureFromCAM(0)

def repeat():

    frame = cv.QueryFrame(capture)

    cv.ShowImage("w1", frame)

while True:

    repeat()

    if cv.WaitKey(33)==27:

        break

cv.DestroyAllWindows()
```

TRANSMITIR IMAGEN

```
import cv2

import urllib

import pdb

import numpy as np

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

class VideoCamera(object):

    def __init__(self):

        self.video = urllib.urlopen('http://192.168.10.12:8080/video')

        #cv2.VideoCapture(0)
```

```

def __del__(self):
    self.video.release()

def get_frame(self):

    bytes=''
    while True:
        # pdb.set_trace()
        bytes+=self.video.read(1024)
        a = bytes.find('\xff\xd8')
        b = bytes.find('\xff\xd9')
        if a!=-1 and b!=-1:
            jpg = bytes[a:b+2]
            bytes= bytes[b+2:]

            img = cv2.imdecode(np.fromstring(jpg,
dtype=np.uint8), cv2.IMREAD_COLOR)
            # pdb.set_trace()
            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            faces = face_cascade.detectMultiScale(gray, 1.3, 5)
            for (x,y,w,h) in faces:
                cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
                roi_gray = gray[y:y+h, x:x+w]
                roi_color = img[y:y+h, x:x+w]

            ret, jpeg = cv2.imencode('.jpg', img)
            return jpeg.tobytes()

```

STREAMING

```
from flask import Flask, render_template, Response
from camera import VideoCamera
import pdb
app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html')

def gen(camera):
    while True:
        frame = camera.get_frame()
        yield (b'--frame\r\n'
              b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

@app.route('/video_feed')
def video_feed():
    return Response(gen(VideoCamera()),
                   mimetype='multipart/x-mixed-replace; boundary=frame')

if __name__ == '__main__':
    app.run(host='127.0.0.1', debug=True)
```