

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Diseño y Programación de un Sistema de Navegación y
Medición para un Robot de Operaciones de Búsqueda y
Rescate**

Trabajo de graduación presentado por Carlo David Sanchinelli Lenis
para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2019

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



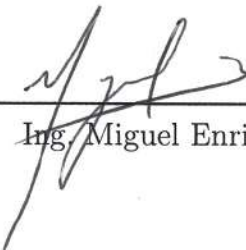
**Diseño y Programación de un Sistema de Navegación y
Medición para un Robot de Operaciones de Búsqueda y
Rescate**

Trabajo de graduación presentado por Carlo David Sanchinelli Lenis
para optar al grado académico de Licenciado en Ingeniería Mecatrónica


Guatemala,


2019

Vo.Bo.:

(f) 
Ing. Miguel Enrique Zea Arenales

Tribunal Examinador:

(f) 
Ing. Miguel Enrique Zea Arenales

(f) 
Ing. Carlos Alberto Esquit Hernández

(f) 
Ing. Luis Pedro Montenegro Mejicanos

Fecha de aprobación: Guatemala, 4 de Diciembre de 2019.

Este trabajo se realizó gracias al apoyo y motivación proporcionada por mi familia y amigos cercanos de la facultad de ingeniería. También agradezco al ingeniero Miguel Zea, quien brindó su asesoría y experiencia durante todo el proyecto para permitirme presentar un trabajo profesional y completo. Finalmente, quiero agradecer a la Universidad del Valle de Guatemala por proporcionar el apoyo financiero de diversos componentes electrónicos necesarios para la implementación de los diseños realizados.

Prefacio	v
Lista de figuras	xii
Lista de cuadros	xiii
Resumen	xv
Abstract	xvii
1. Introducción	1
2. Antecedentes	3
2.1. Robot RHex	3
2.2. Robot Puffer	4
2.3. Robot Quince	5
3. Justificación	7
4. Objetivos	11
4.1. Objetivo general	11
4.2. Objetivos específicos	11
5. Alcance	13
6. Marco teórico	15
6.1. Comunicación	15
6.1.1. Comunicación serial y estándares	15
6.1.2. Hardware y software	18
6.2. Control y comando	20
6.2.1. Arduino	20
6.2.2. Raspberry Pi	21
6.3. Medición y sensores	21
6.3.1. Detección de CO ₂ y sus efectos	21

6.3.2. Cámaras térmicas e infrarrojas	22
7. Diseño general del módulo electrónico	25
7.1. Necesidades del sistema de comunicación	26
7.1.1. Control remoto y cableado	26
7.1.2. Potencia del sistema	27
7.1.3. Componentes necesarios	28
7.2. Necesidades del sistema de comandos	28
7.2.1. Movilización en el entorno	29
7.2.2. Control de los actuadores	30
7.2.3. Componentes necesarios	30
7.3. Necesidades del sistema de medición	31
7.3.1. Recolección de información del entorno	31
7.3.2. Empaquetamiento de información y conexiones	32
7.3.3. Componentes necesarios	33
7.4. Componentes finales a utilizar	33
8. Desarrollo del sistema de comunicación	37
8.1. Diseño del sistema	37
8.1.1. Comunicación externa	37
8.1.2. Comunicación interna	39
8.2. Implementación del sistema	41
8.2.1. Implementación física	41
8.2.2. Implementación de software	42
8.2.3. Programación de interacciones operario - Raspberry Pi	42
8.2.4. Programación de interacciones Raspberry Pi - Arduino	43
8.3. Sistema de potencia	43
9. Desarrollo del sistema de comandos	47
9.1. Diseño del sistema	47
9.1.1. Interfaz gráfica de datos de salida	47
9.1.2. Control de los actuadores	48
9.2. Implementación del sistema	49
9.2.1. Interfaz gráfica de datos de salida	49
9.2.2. Control de actuadores	50
10. Desarrollo del sistema de medición	53
10.1. Diseño del sistema	53
10.1.1. Interfaz gráfica de datos de entrada	53
10.1.2. Operación de sensores y empaquetamiento de información	54
10.2. Implementación del sistema	56
10.2.1. Interfaz gráfica de datos de entrada	56
10.2.2. Operación de sensores y empaquetamiento de información	56
11. Prueba y validación de sistemas	61
11.1. Prueba del sistema de comunicación	61
11.2. Prueba del sistema de comandos	62
11.3. Prueba del sistema de medición	63

12.Resultados finales	65
12.1. Implementación del módulo electrónico	65
12.2. Prototipo final desarrollado	66
13.Conclusiones	69
14.Recomendaciones	71
15.Bibliografía	73
16.Anexos	75
16.1. Repositorio con la programación del módulo electrónico	75
16.2. Despliegue de comandos en la interfaz gráfica	75
16.3. Avances de la segunda fase	78

Lista de figuras

1.	Robot RHex	4
2.	Robot Puffer	5
3.	Robot Quince	5
4.	Rescatistas en la erupción del volcán de fuego	8
5.	Comunicación serial asíncrona [7]	16
6.	Ejemplo de un bloque de datos [7]	18
7.	Interfaz UART en detalle [7]	19
8.	Ejemplo de una imagen térmica	23
9.	Cámara infrarroja vs. cámara convencional [14]	23
10.	Comunicación entre computadoras	27
11.	Funcionamiento de las ruedas	29
12.	Conexión de sistemas internos del robot	35
13.	Conexión entra la computadora del operario y la Raspberry Pi con RS-232	38
14.	Conexión entra la computadora del operario y la Raspberry Pi sin RS-232	39
15.	Conexiones internas de comunicación del robot	40
16.	Conexión entre el operario y el robot con RS-232	41
17.	Conexión entre el operario y el robot sin RS-232	41
18.	Esquemático del sistema de potencia con los actuadores	45
19.	Esquemático del sistema de potencia con los sensores	45
20.	Diseño preliminar para comandos del robot	48
21.	Interfaz gráfica final para comandos del robot	50
22.	Esquemático de conexiones de actuadores	52
23.	Diseño preliminar para información de sensores	54
24.	GUI final con datos de entrada y salida	56
25.	Esquemático de conexiones de sensores	58
26.	Transferencia de datos Arduino<->RPI<->PC	62
27.	Recibo de datos en la interfaz gráfica	62

28.	Recibo de comandos por el Arduino	63
29.	Prueba de medición de niveles de CO2 (ppm)	63
30.	Prueba de medición de temperatura	64
31.	Módulo electrónico del robot	65
32.	Placa de pruebas externa para RS-232	66
33.	Placa de pruebas interna	67
34.	Prototipo final ensamblado	67
35.	Interfaz gráfica con indicador de arriba	75
36.	Interfaz gráfica con indicador de abajo	76
37.	Interfaz gráfica con indicador de izquierda	76
38.	Interfaz gráfica con indicador de derecha	76
39.	Interfaz gráfica con indicador de arriba-derecha	77
40.	Interfaz gráfica con indicador de arriba-izquierda	77
41.	Interfaz gráfica - segunda fase	78
42.	Reestructuración de conexiones internas - segunda fase	78
43.	Diseño de placa para conexiones internas - segunda fase	79
44.	Diseño de placa para conexiones externas - segunda fase	79
45.	Placa de conexiones internas - segunda fase	80
46.	Placa de conexiones externas - segunda fase	80
47.	Conexiones del prototipo final - segunda fase	81
48.	Prototipo final ensamblado- segunda fase	81

Lista de cuadros

1. Comparación de los estándares RS-232 y RS-485 18

Este trabajo hace referencia al sistema electrónico para un robot de operaciones de búsqueda y rescate. El robot, que se compone de un cuerpo o chasis desarrollado por el módulo mecánico, alberga en su interior el sistema electrónico desarrollado en este trabajo. Este cumple con el objetivo general de diseñar e implementar dicho sistema para que maneje adecuadamente las tareas de comando, medición y comunicación de un robot de operaciones que permita la búsqueda y rescate de personas en zonas de accidentes. El sistema electrónico se divide en subsistemas que se encargan del control de los motores, manejo de los sensores, distribución de la energía y comunicación entre controladores/computadoras; los cuales se han desarrollado e implementado con la intención de ofrecer una solución atractiva a los rescatistas. El sistema de comunicación es encargado del recibo y envío de datos y señales; además de otorgar la potencia necesaria para operar los otros sistemas. Por otra parte, el sistema de comandos es el encargado del control de las ruedas y de la sincronización del movimiento. Finalmente, el sistema de medición es el encargado del uso de sensores y cámaras para obtener información del entorno del robot. Para realizar estas tareas, se utiliza una computadora Raspberry Pi y un controlador Arduino. Estos permiten al robot ser controlado por el operario para la navegación del mismo y realizar la medición de factores del entorno para poder realizar las operaciones de búsqueda y rescate de la manera más eficiente y costo-efectiva posible. El robot opera correctamente, obteniéndose una transferencia del 100 % de la información a una velocidad de comunicación de 78.86 Kbps entre el robot y el operario.

The following work concerns the electronics systems for a search-and-rescue operations robot. The robot, which is composed of a body developed by the mechanical module, holds in its interior the electronics systems developed in this work. This is done with the main objective of designing and implementing a system that manages successfully the command, measurement, and communication tasks of a search-and-rescue operations robot that allows the location of people in disaster zones. The electronic system is divided in subsystems that deals with the control of the motors, sensor management, energy distribution, and communications between controllers/computers; which have been designed and implemented with the intention of offering an attractive solution to the rescue crew. The communications system is in charge of receiving and sending data and signals; besides also providing the necessary power to operate the other systems. On the other hand, the command system is in charge of the control of the wheels and the movement synchronization. Finally, the measurement system is in charge of the use of sensors and cameras to obtain information of the robot's surroundings. To be able to perform these tasks, a Raspbery Pi computer and an Arduino controller are being used. These allow the robot to be controlled by the operator to navigate it and perform the measurements of the different factors in the surroundings to be able to perform search-and-rescue operations in the most efficient and cost-effective way possible.

CAPÍTULO 1

Introducción

Este trabajo trata sobre el sistema electrónico de un robot de de operaciones de búsqueda y rescate, bajo el desarrollo de estos robots como robots de navegación todoterreno. Este robot se encuentra inspirado en los robots comerciales todoterreno, tales como el Puffer de la NASA y el robot RHex de Boston Dynamics. Esto debido a que presentan una versatilidad en su funcionamiento y los permite adentrarse en terrenos irregulares y peligrosos. El objetivo de este módulo electrónico es el de diseñar e implementar un sistema que maneje adecuadamente las tareas de comando, medición y comunicación de un robot de operaciones que permita la búsqueda y rescate de personas en zonas de accidentes. Para realizar esta tarea, se ha dividido el desarrollo del módulo en tres subsistemas: sistema de comunicación, sistema de comandos y sistema de medición. Cada sistema se ha desarrollado de manera independiente, sin embargo el correcto funcionamiento de uno conlleva a que el otro se pueda implementar en el prototipo final. Es por esto que se ha establecido una fase de diseño general previo al desarrollo e implementación de cada subsistema, la cual se encuentra en el capítulo 7 de este trabajo. En ella se asegura con una planificación preliminar que los sistemas se van a poder interconectar y se verán cumplidos los objetivos del proyecto. Una vez se ha determinado el desarrollo a seguir para cada sistema, se procede a detallar su planificación y ejecución en los subsecuentes capítulos. El capítulo 8 hace mención sobre el desarrollo del sistema de comunicación, el cual garantiza el funcionamiento correcto de los otros sistemas por lo que se le da especial atención y cuidado. El capítulo 9 y 10 mencionan el desarrollo de los subsistemas de comandos y medición, donde se deja claro el funcionamiento y utilización de los mismos para poder operar al robot correctamente y obtener información de su entorno. Finalmente, una vez se ha elaborado y explicado cada subsistema, se procede a evaluar y mostrar resultados. En esta parte se observan las figuras y pruebas realizadas con el prototipo final. Todo esto para, según los criterios definidos en el capítulo de *Alcance*, cumplir con los tres objetivos específicos que corresponden a cada submódulo. Se espera que el siguiente trabajo transmita un interés a seguir desarrollando el campo de la robótica en el área de rescate y de robots todoterreno, así como de continuar con su desarrollo en fases de implementación futuras bajo este mismo concepto.

La idea de desarrollar robots con la habilidad de movilizarse en diferentes terrenos fácilmente, conocidos como robots todoterreno, ha sido explorada por diferentes instituciones y universidades recientemente. Ejemplo de esto es el desarrollo de robots tales como los RHex o el Puffer. Estos se han desarrollado con diferentes aplicaciones en mente, cubriendo desde estudios meteorológicos hasta la exploración de terrenos hostiles interplanetarios. Sin embargo, aun cuando hay una increíble diversidad de aplicaciones específicas para estos, su desarrollo en el área de operaciones de búsqueda y rescate para utilizarse como robots rescatistas ha sido pobre. Es importante mencionar que sí se encuentran ejemplos de robots rescatistas, tales como el robot Quince, el cual atiende rescates de gran magnitud en Japón (nucleares, explosivos, químicos, etc.). Sin embargo, estos no se encuentran desarrollados al nivel de otros robots todoterreno. Buscando ofrecer entonces una mayor comprensión en el área de desarrollo y de tareas de los robots en desarrollo actualmente, así como un entendimiento de sus características y habilidades, se procede a explicar y detallar el proceso y funcionamiento de algunos de los más populares en esta rama de investigación:

2.1. Robot RHex

Los robots RHex, tal como el de la Figura 1, son robots hexápodos con patas de diseño inteligente de uso individual, los cuales son capaces de avanzar sobre una gran variedad de terrenos. Gracias a su diseño bioinspirado en la locomoción de los artrópodos, estos robots se mueven eficientemente en estos terrenos y a una mayor velocidad en comparación con otros robots todoterreno. Existen diferentes tipos de robots RHex, donde las habilidades y funciones dependen de los objetivos de cada institución que lo desarrolla. El RHex más novedoso es el X-RHex, desarrollado por la Universidad de Pensilvania. Este es un robot que mejora los diseños anteriores de otros RHex ya que cuenta con un sistema modular integrado en la parte superior, el cual permite instalar/remover una variedad de sensores y dispositivos según las necesidades del proyecto. El X-RHex fue desarrollado con la modalidad COTS (commerical-off-the-shelf), por lo que sus componentes son fáciles de conseguir y no

involucran un avanzado y complicado desarrollo de componentes mecánicos y electrónicos (como es el caso de la mayoría de robots de esta categoría). Además, cuenta con un sistema simplificado de fabricación y mantenimiento, lo cual lo hace muy atractivo para tomarlo como base en el desarrollo de otros robots RHex de universidades. Entre los módulos que se pueden instalar están webcams, GPS, IMU, LIDAR, sensores varios, comunicación wireless, USB hubs, baterías auxiliares e incluso un brazo robótico. Como se puede observar, las aplicaciones de un robot clásico RHex se ven muy limitadas en comparación a las de este nuevo diseño modular. Entre las aplicaciones más importantes del robot está la recolección de datos, clasificador de terrenos y el mapeo 3D de diferentes entornos usualmente inaccesibles. La universidad ofrece en su reporte [1] información muy completa de componentes y el desarrollo de toda la electrónica, la cual es de gran interés si uno piensa realizar un proyecto similar (en especial por la modalidad COTS que implementan en su diseño). El robot utiliza una computadora central que se encarga de las comunicaciones de alto nivel. El control se realiza mediante controladores de motores (una por pata) y un board de interface de control, encargado de la distribución de potencia y de la comunicación con los controladores). Se utilizan baterías LiPo para este robot, las cuales son flexibles y livianas, pero altamente volátiles (por lo que cuentan con un sistema avanzado de control).



Figura 1: Robot RHex

2.2. Robot Puffer

El robot Puffer, desarrollado por la NASA, presenta un diseño bastante novedoso y poco común en el área de robots todoterreno. Este cuenta tan solo con dos ruedas, sin embargo es capaz de ajustarlas para avanzar en terrenos irregulares e inclinados. El prototipo del robot se observa en la Figura 2. La intención de este robot es utilizarlo en misiones a otros planetas, siendo un tipo de scout para obtener información del entorno. El robot se ha puesto a prueba en numerosos entornos, desde desiertos hasta la Antártica, y ha presentado buenos resultados. El robot es capaz de avanzar en pendientes de hasta 45 grados, caer de grandes alturas, y atravesar dunas de arena. El robot se utilizaría para acompañar a otros robots de funciones más importantes, siendo un robot de apoyo. Sin embargo, se le quiere dar utilidad aquí en la tierra también. Entre las tareas propuestas se encuentra el posible estudio de volcanes y geología. El robot funciona de manera inalámbrica, controlado mediante Bluetooth por el operario, pero los planes a futuro es que el robot pueda funcionar con autonomía. Además, se le puede agregar una variedad de instrumentos para que cumpla aún más funciones, tales como espectrómetro para estudiar el ambiente o contenedores de muestras para estudiar diversos materiales. La construcción es bastante simple, donde dos ruedas capaces de retraerse impulsan al robot a su destino [2].

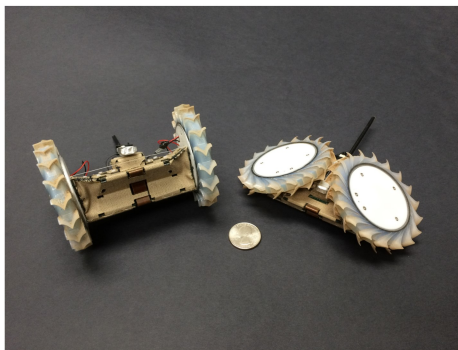


Figura 2: Robot Puffer

2.3. Robot Quince

El robot Quince es un robot diseñado para atender desastres químicos, biológicos, radiológicos, nucleares o explosivos. Muchos de estos desastres no pueden ser atendidos por humanos directamente, por lo que desea utilizar la ayuda de robots para realizar tareas que los humanos tendrían a cargo en una situación similar. Este robot se utilizó para explorar el interior de la planta nuclear de Fukushima en el 2011 cuando ocurrió el desastre del terremoto, donde tomó fotografías de áreas importantes de la planta y realizó un mapa de zonas radioactivas al tomar muestras del aire en determinados sectores. Al utilizar a un robot para estas tareas se está protegiendo a la gente que ayudará a limpiar el desastre, así como a los trabajadores de la planta al reducir el trabajo que se necesitaba hacer para estudiar los daños. Este robot avanza en terrenos irregulares de manera distinta a los otros robots estudiados, donde utiliza patas mezcladas con ruedas para poder avanzar. Se puede observar en acción en la Figura 3. El robot cuenta con instrumentos avanzados como cámaras de alta resolución, dosímetros, sistema de iluminación, sensores de calidad del aire, termómetros, medidor de niveles de agua y un brazo robótico. El equipo de comunicación del robot está protegido contra radiación, así como sus componentes más sensibles[3].



Figura 3: Robot Quince

Guatemala es un país con una geografía bastante interesante y variada. Evidencia de esto es la diferencia marcada entre sus tres regiones principales: las tierras altas de clima agradable, la región tropical de la costa tanto del Pacífico como del Caribe y la región de jungla tropical en el norte del país. Cada región trae consigo una variedad de ventajas y desventajas, donde la variedad de climas y condiciones ha obligado a la población Guatemalteca a entender los cuidados que se deben tener según la región en donde están. La región tropical costera consta de muchos ríos cortos, lagunas y llanuras. Estas características la ponen en riesgo de inundaciones de ríos, inundaciones costeras y tsunamis. Por otro lado, la región de jungla tropical, que consta de tierras bajas y una jungla densa, puede sufrir de incendios forestales, escasez de agua y calor extremo. Finalmente, la región central de tierras altas es la más urbanizada y la que contiene los asentamientos más grandes del país. En esta región pueden ocurrir derrumbes o deslizamientos de tierra, pero los volcanes activos en la región son los que presentan los mayores problemas. Además de todos estos problemas, el país se encuentra propenso a terremotos y actividad sísmica de diferentes grados. Según la GFDRR, una asociación que promueve la ayuda a países en vías de desarrollo al reducir su vulnerabilidad ante desastres naturales, los principales peligros en Guatemala son: actividad sísmica, erupciones volcánicas, inundaciones, deslizamientos de tierra y sequías [4]. Entre 1975 y 2015, las pérdidas y daños por estos accidentes se estiman en 9.1 mil millones de dólares. Entre los accidentes más fuertes en la historia del país están: el terremoto de 1976, la tormenta tropical Agatha y la erupción tanto del volcán Pacaya como el de fuego [5]. Los daños causados han afectado a más de 470,000 personas y destruyeron más de 35,000 hogares. En los estudios pos-desastre realizados, se ha establecido que las prioridades del país para evitar encontrarse vulnerables ante estos accidentes son:

- Mejorar la respuesta ante los desastres naturales, así como en la recuperación y mitigación de las consecuencias que estos conllevan.
- Fomentar a nivel nacional el conocimiento de los riesgos en el país.
- Mejorar la planificación en el uso de tierras en áreas de alto riesgo.



(a) Equipo de rescate en acción



(b) Condiciones en el área del desastre

Figura 4: Rescatistas en la erupción del volcán de fuego

Como se puede observar, los desastres naturales son un punto crítico en el país a un nivel económico, político, social y cultural. Es por esto que se ha decidido desarrollar un proyecto que pueda facilitar en la realización de los puntos establecidos como prioridades del país, sobre todo en lo que es el primer punto. Buscando mejorar la respuesta de las autoridades del país ante los desastres naturales, se ha propuesto desarrollar un robot de operaciones de búsqueda y rescate, el cual tiene como objetivo brindar apoyo a los cuerpos de socorro y rescatistas en una diversidad de accidentes y desastres naturales. Un robot de este tipo controlado de manera remota ofrece los siguientes beneficios a los rescatistas:

- Atender emergencias rápidamente, aun cuando las condiciones climáticas no son las apropiadas para que los rescatistas puedan entrar de inmediato. El robot funciona básicamente como una sonda de reconocimiento, permitiendo que los rescatistas no pierdan tiempo adicional al entrar al accidente y se puedan enfocar en las áreas de interés del accidente.
- Ubicar personas enterradas o en áreas de difícil acceso, lo cual permite salvar una mayor cantidad de vidas en estos accidentes al evitar gastar recursos y tiempo en el búsqueda general de las personas.
- Explorar terrenos donde el ingreso de personas no sería posible bajo ninguna condición, tales como pasajes angostos, ambientes tóxicos o a temperaturas sumamente altas. Esto es especialmente importante para situaciones en montañas o volcanes, ya que Guatemala no cuenta con un servicio de rescate profesional para montañas como en otros países.

Es por estas razones que se considera que el proyecto del robot de operaciones de búsqueda y rescate es un punto crítico en el país.

qu coasta y rescate presenta increíbles ventajas al país, presentando una respuesta para poder cumplir con uno de los objetivos de alta prioridad del país ante los desastres naturales. Además de ayudar al país, este proyecto permite que el estudiante utilice los conceptos y habilidades que ha ido desarrollando durante la carrera en el área electrónica y mecánica. Se está trabajando en un proyecto interesante que promueve el aprendizaje de la ingeniería mecatrónica mientras que se está colaborando para el desarrollo del país en el área de la vulnerabilidad ante estos desastres.

4.1. Objetivo general

Diseñar e implementar un sistema electrónico que maneje adecuadamente las tareas de comando, medición y comunicación de un robot de operaciones que permita la búsqueda y rescate de personas en zonas de accidentes.

4.2. Objetivos específicos

- Realizar la implementación de un sistema de comunicación de sistemas internos y externos del robot que permita su interacción con el operario a distancias mayores a 100m.
- Realizar la implementación de un sistema de comandos que actúe las ruedas del robot para su navegación según las instrucciones del operario.
- Realizar la implementación de un sistema de medición que utilice una cámara y un conjunto de sensores para obtener información de la ubicación y posición del robot de manera constante.

Este proyecto representa la primera fase de desarrollo de un robot de operaciones de búsqueda y rescate en la Universidad del Valle de Guatemala. Este se ha dividido en los siguientes tres módulos:

- Módulo mecánico
- Módulo electrónico
- Módulo del desarrollo de ruedas inteligentes

El objetivo general al unir los tres módulos de investigación es el de diseñar un robot que sea capaz de moverse y recolectar datos según el control de un operario para realizar dichas tareas de búsqueda y rescate. El módulo electrónico, por otra parte, busca manejar adecuadamente las tareas de comando, medición y comunicación del robot; esto con el objetivo de que pueda efectuar todas sus tareas necesarias, con lo cual permitirá cumplir finalmente con el objetivo general del proyecto.

Para cumplir con el objetivo del módulo, se han establecido los siguientes objetivos específicos, los cuales al cumplirse permiten que se cumpla el objetivo general:

- Objetivo específico 1 - Diseñar e implementar el sistema de comunicación: Este objetivo se considera realizado cuando se tenga diseñado un sistema que permita al operario recibir y enviar señales del módulo electrónico del robot a largas distancias. Este sistema se debe probar físicamente o se debe evidenciar digitalmente para garantizar que efectivamente se encuentra funcional.
- Objetivo específico 2 - Diseñar e implementar el sistema de comandos: Este objetivo se considera realizado cuando el robot reaccione en su movimiento según las instrucciones del usuario, evidenciando que la locomoción del robot se encuentra funcionando correctamente. Este sistema se debe probar físicamente o se debe evidenciar digitalmente para garantizar que efectivamente se encuentra funcional.

- Objetivo específico 3 - Diseñar e implementar el sistema de medición: Este objetivo se considera realizado cuando se pueda obtener información de la ubicación del robot en forma de imágenes y datos generales (por ejemplo como temperatura, posición o niveles de CO2) en algún determinado momento. Este sistema se debe probar físicamente o se debe evidenciar digitalmente para garantizar que efectivamente se encuentra funcional.

Existen limitaciones y restricciones en el proyecto, las cuales afectan el alcance final de cada objetivo específico. Es por esto que se han planteado los criterios anteriores, los cuales establecen el momento en que se ha cumplido efectivamente el objetivo. Esto se realiza con la intención de afirmar que se presentarán resultados útiles que cumplan con las metas establecidas, mientras que al mismo tiempo se asegura el investigador que las diferentes limitaciones no presentaran problemas en la entrega de resultados. Las limitaciones y restricciones de cada submódulo son las siguientes:

- Sistema de comunicación: La prueba física de este sistema implica la obtención de un cable serial de una longitud mayor a 100 metros. En caso dicho cable no sea posible de obtener, se garantizan pruebas (ya sea físicas o digitales) con longitudes menores, estableciendo evidencia que demuestre que el sistema funciona efectivamente con cables de longitudes mayores a 100 metros.
- Sistema de comandos: La prueba física de este sistema implica que el responsable del módulo mecánico cuenta ya con un prototipo funcional que permita la locomoción del robot. Si este prototipo no se encuentra disponible, se deja clara evidencia con pruebas o simulaciones digitales que el módulo electrónico puede enviar señales controladas donde en caso se colocaran actuadores estos responderían según las instrucciones del operario.
- Sistema de medición: La prueba física de este sistema implica que se tiene cámaras y sensores disponibles para el proyecto. Si estos no se encontraran disponibles, se asegura que el robot realizará las mediciones necesarias que garantizaría la obtención de los diferentes datos. Esto se puede demostrar con pruebas o simulaciones digitales.

Tomando en cuenta el alcance de los objetivos y sus restricciones, se establece el alcance de este proyecto en general: El módulo electrónico del proyecto del robot de operaciones de búsqueda y rescate tiene como alcance presentar el diseño de un sistema capaz de realizar las tareas de comando, medición y comunicación que permitan, una vez integradas con los otros módulos, realizar un prototipo funcional cuya finalidad es su aplicación en misiones de rescate, permitiendo obtener información de su entorno en estas situaciones para brindar apoyo a los rescatistas. Se espera que este prototipo presente una base sólida a otras fases posteriores del proyecto, las cuales buscarán optimizar y expandir las habilidades del robot.

6.1. Comunicación

6.1.1. Comunicación serial y estándares

La tecnología de comunicación ha permitido que enlacemos dispositivos, herramientas y computadoras para poder desarrollar y mejorar nuevas tecnologías que permiten conectar cada vez mas al mundo. Gracias a esta contamos con teléfonos, celulares, internet, servidores, radios, plantas automatizadas, televisores inteligentes, satélites, etc. Son tantas las tecnologías, sin embargo estas se pueden dividir en dos grandes y distintas categorías:

- Comunicación alámbrica
- Comunicación inalámbrica

La comunicación alámbrica, que es la más antigua y desarrollada, es aquella que utiliza conexiones físicas mediante el cableado. Ejemplos de estas son el Ethernet, fibra óptica, cable coaxial, etc. Esta comunicación se puede dividir aún más en:

- Comunicación alámbrica electrónica
- Comunicación alámbrica de fibra

La comunicación alámbrica electrónica por lo general utiliza comunicaciones digitales cuando se trata de sistemas modernos. Estas tienen la habilidad de transferir datos digitalmente, donde se dividen aún más según la configuración utilizada en computadoras y dispositivos:

- Comunicación paralela

- Comunicación serial

Es así como nos enfrentamos con la comunicación serial. La comunicación serial funciona enviando datos en secuencia mediante pulsos binarios de cero (cero volteos) o uno (voltaje positivo). Esto quiere decir que se envía bit por bit mediante tres diferentes modos de transmisión:

- Simplex: Es una comunicación de una vía, donde solo hay un transmisor y un receptor. Estos nunca realizan la tarea opuesta.
- Half-Duplex: Es una comunicación de dos vías, sin embargo no pueden ocurrir al mismo tiempo. Esto quiere decir que no puede ocurrir un envío y un recibo por parte de un mismo cliente en determinado momento.
- Full-Duplex: Es una comunicación de dos vías, pero a diferencia del Half-Duplex en esta si puede ocurrir un envío y un recibo por parte de un mismo cliente en determinado momento. Esta requiere una conexión adicional.

Si bien la comunicación serial es más lenta, esta ocupa menos espacio que la paralela, es más económica y permite el envío de señales a largas distancias. Esto la hace una de las comunicaciones más simples ya que no requiere de conexiones avanzadas. Además de los aspectos ya mencionados, una comunicación serial se controla por dos métodos:

- Comunicación serial síncrona: La transferencia de datos es controlada por una señal de un reloj sincronizador que dicta el recibo/envío de los datos.
- Comunicación serial asíncrona: La transferencia de datos es controlada por una serie de pulsos establecidos por diferentes protocolos asíncronos de comunicación. Este tipo de comunicación es bastante útil para aplicaciones de largas distancias.

Dos de los protocolos asíncronos más importantes que se utilizan en la industria son el RS-232 y el RS-485 [6].

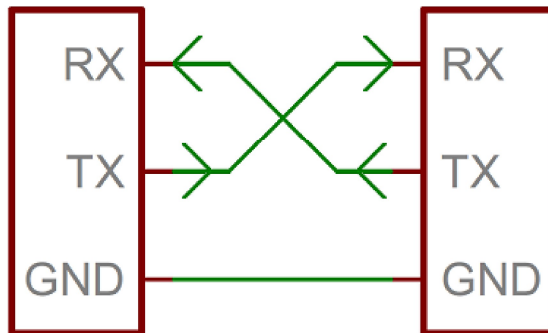


Figura 5: Comunicación serial asíncrona [7]

RS-232

En 1960 el Electronic Industries Association (EIA) desarrolló un protocolo para equipo de comunicación que establecía una comunicación confiable entre estos. Además, la idea detrás de este estándar era que equipo de diferentes compañías funcionara correctamente. El estándar se denominó RS232C. Este estándar establece niveles de voltaje, tipo de señales, tiempos de señales y conectores a utilizar. La última modificación al RS-232 se realizó en 1991, la cual se denomina EIA232F.

El estándar RS-232 es uno de los más simples y populares, por lo que ha perdurado por bastantes años. Este es un protocolo full-duplex, por lo que puede haber tanto un envío y recibo de información por ambos dispositivos simultáneamente. Para poder utilizar el protocolo se necesitan tres cables. Un cable se encarga del envío de datos, otro para recibir datos y el último para referenciar ambos sistemas al conectarlos a la misma tierra. RS-232 intercambia datos con bits binarios, donde un estado es 12V (HIGH) y -12V (LOW). El estándar funciona de modo asíncrono, donde cada dispositivo maneja la misma frecuencia por separado y cada uno cuenta con su propio reloj. La velocidad de transmisión se establece con el baudaje, el cual debe ser el mismo para ambos dispositivos. Algunos baudajes estándar son 110, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200, 128000 y 256000. La distancia máxima entre dispositivos con RS-232 es de 50 pies. Este estándar era bastante común en equipos de computación, los cuales contaban con puertos seriales para diferentes periféricos, sin embargo hoy en día esto ha cambiado ya que se emplean los puertos USB. Estas conexiones aún se encuentran en numerosos dispositivos industriales [8].

RS-485

RS-485 es un protocolo/estándar de transmisión que utiliza una diferenciación de señales. Esto quiere decir que un bit binario tiene un voltaje relativo fluyendo de la terminal positiva a la terminal negativa. Este se diseñó en 1983 con el objetivo de conectar varios emisores a la red, permitiendo realizar conexiones mucho más complejas al interconectar aún más dispositivos. Para poder utilizar este protocolo se requiere de dos cables por cada línea de transmisión, donde usualmente estos van trenzados para evitar disturbios o ruido externo.

Al utilizar una diferenciación de señales se pueden alcanzar distancias aún mayores, de hasta 4000 pies. Este protocolo se estableció para acompañar al RS-232, donde se convierte de RS-232 a RS-485 y luego de regreso a RS-232. Esto permite obtener una señal RS-232 en distancias que no serían posibles por si solo. Otra ventaja que presenta este protocolo es que permite realizar aplicaciones sistemas de comunicación multipunto. Esto quiere decir que múltiples terminales se pueden conectar a un mismo bus, donde cada terminal es capaz de enviar o recibir señales de este bus. RS-485 permite hasta 32 transmisores y 32 receptores en comunicación half-duplex [8].

Características del RS-232 y RS-485		
Parámetros	RS-232	RS-485
Tipo de cableado	Simple	Diferencial
Cantidad de dispositivos	1 transmisor 1 receptor	32 transmisores 32 receptores
Longitud máxima de cableado	50 pies	4000 pies
Tasa máxima de transmisión	20 kbits/s	10 Mbits/s
Tipo de señal	No-balanceada	Balanceada
Voltajes de operación	$\pm 5 \sim \pm 15V$	$\pm 1.5 \sim \pm 6V$
Impedancia mínima de entrada	$3 \sim 7k\Omega$	$12k\Omega$
Sensibilidad del receptor	$\pm 3V$	$\pm 200mV$

Cuadro 1: Comparación de los estándares RS-232 y RS-485

6.1.2. Hardware y software

UART

El UART no hace referencia a un estándar, sino a la interfaz instalada en los dispositivos con comunicación serial asíncrona. La función de este es transmitir y recibir datos seriales de un dispositivo a otro. Esta interfaz es muy común en los microcontroladores, mientras que las computadoras los han abandonado por el uso de los puertos USB. La interfaz UART ("Universal Asynchronous Receiver-Transmitter") establece dos pines, uno de recibo y otro de transmisión. Esta no depende de ningún protocolo, por lo que el formato de los datos y el baudaje son determinados por los usuarios. Como se trata de una comunicación serial asíncrona, la comunicación entre UARTs establece bits de inicio, bits de final y bits de paridad para determinar cuál es el paquete de datos.

- Bits de inicio: La línea de transmisión esta en un nivel alto de voltaje, donde el bit inicial lo pasa de alto a bajo por un ciclo. Cuando un UART detecta este cambio de estado empieza a leer los datos.
- Bits de paridad: Es un bit que permite detectar errores básicos en el recibo de datos. Este indica si la cantidad de bits en estado alto de los datos recibidos es un número par o impar. Si la cantidad no coincide con el bit de paridad es porque hubo un error en el recibo de los datos.
- Bits de final: La línea de transmisión pasa de nivel bajo a alto por cierta duración para indicar que se ha terminado el envío.



Figura 6: Ejemplo de un bloque de datos [7]

Como se puede ver, al no utilizar un reloj es necesario tomar precauciones para garantizar un envío exitoso de la información. Es por esto que se utilizan los bits mencionados

anteriormente, los cuales ayudan a evitar errores de envío/recibo. Las ventajas del UART es que es bastante utilizado (por lo que cuenta con amplia documentación), solo requiere de dos cables y tiene una detección de errores básica. Sin embargo, el baudaje de ambos dispositivos no puede variar en más de 10 por ciento, de lo contrario habrán muchos errores y la comunicación no será confiable [9].

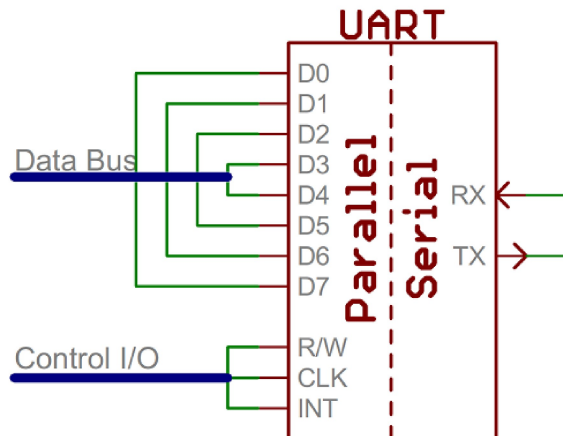


Figura 7: Interfaz UART en detalle [7]

Bit-Banging

Existen numerosos protocolos y estándares de comunicación serial, así como su respectivo hardware que permite que interactuemos con estos. Los protocolos y estándares más comunes son: RS-232, RS-422, RS-485, I2C, SPI, Microwire y 1-Wire.

Para poder utilizar estos protocolos se necesita muchas veces de hardware específico que nos permita obtener una comunicación efectiva. En el caso del hardware UART, este se presenta tanto en los microcontroladores Arduino como en las computadoras Raspberry Pi. El UART nos permite utilizar RS-232, RS-422 y RS-285. Sin embargo, no siempre se tiene disponible el hardware para poder utilizar los protocolos. Cuando no se cuenta con el hardware necesario, se puede utilizar comunicación serial asíncrona simulada con software. A este proceso se le conoce como Bit-Banging.

El Bit-Banging es la utilización de pines I/O de uso general (denominados GPIO) junto a un control mediante software para establecer una comunicación serial. Los pines pasan a ser puertos seriales, donde el software debe ser competente respecto a los tiempos y baudajes para realizar el envío de información exitosamente. El software por parte del emisor tiene entonces la tarea de enviar un bit, esperar determinado tiempo, enviar otro y seguir repitiendo el proceso, mientras que por parte del receptor se debe estar activando la línea de recibo de datos con el mismo tiempo para obtener la información correcta. Existen muchos algoritmos de Bit-Banging, los cuales permiten simular diversos protocolos [8].

Detección de errores

La comunicación serial, especialmente la asíncrona, está propensa a sufrir errores si existen problemas en el baudaje, reloj de cada dispositivo, ruido en algún componente o simplemente alguna mala conexión. Es por esto que se han desarrollado métodos para detectar y señalar errores en la información recibida. Los métodos más comunes de detección de errores son:

- **CRC:** El método CRC protege la información del paquete enviado mediante la inclusión de bits de redundancia de detección al final del paquete, los cuales son generados mediante una división de una clave con los bits del paquete de información. El mensaje recibido toma estos bits y los compara con los bits de los paquetes con la misma clave. Si la comparación no coincide entonces se ha producido un error en el envío de la información. Si se elige una clave adecuada, la fiabilidad de este método es del 99 por ciento.
- **Frame Check:** Este método toma información del paquete de datos, obteniendo el tipo de formato y su tamaño. En base a estas características, se agrega una serie de bits adicionales al final del paquete. El receptor corrobora estas características con el mensaje recibido, pero si la información no coincide entonces se ha producido un error.
- **Errores ACK:** Este método consta de una señal adicional que se envía y recibe entre el receptor y emisor. Una vez llega un paquete de información al receptor, este envía una señal de recibido al emisor. Si el emisor no recibe esta señal entonces vuelve a enviar el paquete hasta recibir una confirmación del receptor. Por lo general este método se acompaña de otro tipo de detección de errores para obtener una comunicación más confiable.

El método de detección de errores CRC es considerado el más seguro de estos, donde existe una diversidad de algoritmos para implementarlos en diferentes lenguajes de programación. La selección de la llave (la cual se representa por polinomios) se hace mediante tablas experimentales que establecen un polinomio que garantiza la mejor exactitud según la cantidad de bits que se desean enviar [6].

6.2. Control y comando

6.2.1. Arduino

Arduino es una compañía que trabaja con open-source hardware y software, la cual utilizan para desarrollar los microcontroladores Arduino (llamados también módulos Arduino). Estos microcontroladores, al ser open-source, pueden ser fabricados y programados por cualquier persona, por lo que encuentran numerosas marcas en el mercado. El énfasis de estos microcontroladores es en la fácil programación y uso, por lo que resultan ser bastante populares para proyectos, investigaciones e instrumentos [10]. Sus ventajas se establecen a continuación:

- Económico: Los módulos resultan ser bastante económicos cuando se comparan con otros microcontroladores.
- Compatibilidad entre diferentes plataformas: Estos se pueden programar ya sea en Windows, Mac o Linux.
- Software de programación simple: El software de programación es bastante claro y simple en el lenguaje C, lo cual permite aprender a utilizarlo rápidamente.
- Open-Source: Esto permite que se implementen numerosas librerías por la comunidad que permiten extender aún más tanto el software como el hardware de Arduino.

Existen módulos más avanzados que otros, donde según el uso que se le dará es como por lo general se escoge cuál utilizar. Se tienen módulos básicos, avanzados, IoT, educativos y usables en la ropa [10].

6.2.2. Raspberry Pi

La Raspberry Pi es una computadora de bajo costo y de pequeñas dimensiones que permite realizar muchas de las funciones de una computadora normal tales como programación, ingresar al internet, reproducir videos, hacer documentos, jugar videojuegos, etc. Debido a su portabilidad y bajo costo, esta computadora es muy popular en proyectos e investigaciones en todo el mundo. Entre sus funciones principales se encuentran:

- Computación general: La Raspberry Pi es una computadora, por lo que puede realizar tareas de computo tradicionales como las mencionadas anteriormente.
- Programación: Las Raspberry Pi incluye una amplia gama de aplicaciones de programación (ya que uno de sus fines principales es ser una herramienta educativa) tales como Java, Python, Ruby, C y Perl.
- Plataforma para proyectos: La Raspberry Pi es muy popular en proyectos ya que es muy fácil integrarla a estos. Tiene una gran variedad de pines GPIO, además de puertos para otros accesorios. Esto, junto a su precio bajo y pequeñas dimensiones, lo hacen muy atractivo para proyectos de este tipo.

Si bien por lo general se utilizan microcontroladores para realizar la mayoría de proyectos, una computadora como la Raspberry Pi ofrece muchas más posibilidades al ser más poderosa y programable [11].

6.3. Medición y sensores

6.3.1. Detección de CO2 y sus efectos

El CO2 es un producto que resulta de procesos metabólicos en los seres vivos. Debido a que los humanos lo producimos constantemente, resulta indispensable entender que es una

prioridad tener una buena ventilación cuando se tienen grandes grupos de personas en un mismo lugar. Cuando una persona se encuentra atrapada, los niveles de CO₂ en el espacio aumentan constantemente debido a la falta de ventilación. Niveles altos de este gas pueden provocar dolor de cabeza, fatiga, náusea, vómito e incluso la pérdida de la consciencia. El CO₂ no tiene olor o sabor, por lo que la mejor manera de controlar su nivel es con un detector de dióxido de carbono. Se han establecido niveles de peligro para diferentes mediciones de CO₂ en partes por millón (ppm), las cuales se detallan a continuación:

- 250-350 ppm: Nivel estándar en un espacio abierto.
- 350-1000 ppm: Nivel aceptable para un espacio cerrado que tiene buena ventilación.
- 1000-2000 ppm: Este nivel representa una calidad pobre del aire. Las personas pueden empezar a sentirse mareados.
- 2000-5000 ppm: Nivel donde se sufre de dolores de cabeza y fatiga. La dificultad en poner atención y el aumento del ritmo cardiaco también pueden ocurrir en este nivel.
- 5000-4000 ppm: Este es un nivel peligroso, donde se da una falta de oxígeno y el ambiente se vuelve tóxico.
- >4000 ppm: Este nivel es dañino a la salud si se está expuesto por un tiempo prolongado. La falta de oxígeno es significativa y puede llevar a perder la conciencia rápidamente y luego a la muerte.

Como se observa, mantener un nivel adecuado es un problema para gente que se encuentra en espacios totalmente cerrados, por lo que su bienestar se convierte en un problema de rapidez en la respuesta de un rescatista [12].

6.3.2. Cámaras térmicas e infrarrojas

Los cámaras son indispensables para poder ubicar a personas o sitios de interés en operaciones de búsqueda y rescate, sobre todo lo que son cámaras térmicas e infrarrojas. Todos los vehículos aéreos no tripulados (UAVs en inglés) así como vehículos de rastreo tienen una variedad de estas cámaras, las cuales utilizan constantemente en operaciones de vigilancia para el gobierno. Sin embargo, estos dos tipos de cámaras se suelen confundir entre sí o se considera que cumplen la misma funcionalidad. La realidad es que ambas funcionan y cumplen conceptos distintos, donde es importante entenderlos para poder saber cuál seleccionar en la elaboración de sistemas de operaciones de búsqueda y rescate.

- Cámaras térmicas: Estas cámaras, que funcionan en realidad como sensores térmicos, obtiene imágenes mediante el calor y no requiere ni utiliza la luz visible para poder operar. Esto quiere decir que una cámara de estas puede tomar fotos tanto de día como de noche, donde se proyecta una escala de calor de los objetos en la imagen. Estas cámaras pueden detectar incluso pequeñas diferencias de calor en las centésimas de grados Celsius, las cuales se observan ya sea en una escala de grises o en una de colores. Debido a que solo se recibe lecturas de la energía térmica de las cosas, no es

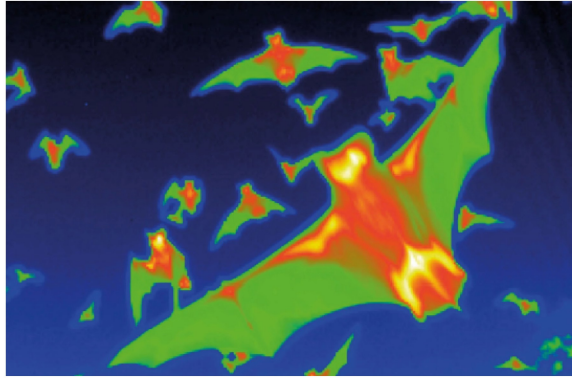


Figura 8: Ejemplo de una imagen térmica

posible detectar o percibir la luz con estas cámaras, donde una misma imagen tomada en el día y en la noche se observarían similares [13].

- Cámaras infrarrojas: Estas cámaras no obtienen imágenes mediante el calor, sino mediante la luz tanto visible como infrarroja. Estas cámaras son bastante similares a las cámaras de fotografía, sin embargo no cuentan con un filtro infrarrojo. Esto quiere decir que no solo capturan el espectro de luz visible por los humanos, sino también parte del invisible al capturar la luz infrarroja. Esto resulta en imágenes con tonalidades distintas en comparación a aquellas de una cámara convencional, como se observa en la siguiente Figura 9. El uso más interesante de estas cámaras es en la noche o en situaciones con poca visibilidad. Ya que mientras haya luz infrarroja presente se puede obtener una imagen bastante clara. Por lo general, se genera esta luz de manera artificial con LEDs cuando se encuentra en espacios sin luz natural.



Figura 9: Cámara infrarroja vs. cámara convencional [14]

Diseño general del módulo electrónico

El robot de operaciones de búsqueda y rescate requiere del diseño, implementación e interconexión de un sistema de comunicación, de comandos y de medición. El correcto funcionamiento de estos tres garantiza que el robot puede cumplir con el objetivo de brindar asistencia a rescatistas ante desastres naturales o accidentes de alto riesgo. Es por esto que es necesario entender las necesidades y restricciones de cada submódulo para poder asegurarse que se están desarrollando de la mejor manera posible y garantizar la interconexión exitosa entre los mismos. Los tres submódulos se detallan a continuación:

- Sistema de comunicación: Encargado del recibo y envío de datos y señales; además de otorgar la potencia necesaria para operar los otros sistemas.
- Sistema de comandos: Encargado del control de las ruedas y de la sincronización del movimiento.
- Sistema de medición: Encargado del uso de sensores y cámaras para obtener información del entorno del robot.

Cada submódulo debe cumplir con ciertos criterios para asegurar que los objetivos del proyecto se estarán cumpliendo de igual manera. Los criterios que los submódulos deben tomar en cuenta son los siguientes:

- El operario debe poder operar el robot aún cuando este se encuentre a distancias largas de más de 100 metros.
- El robot debe controlarse por un operario de forma remota y cableada.
- El operario debe ser capaz de observar y moverse en el entorno del robot con alta destreza en terrenos irregulares y variados.

- El operario debe poder obtener información del entorno del robot que le permita llevar a cabo operaciones de búsqueda y rescate.

Con todos estos puntos en mente, se procede a establecer el diseño preliminar de cada submódulo, el cual permite analizar las necesidades y exigencias de cada tarea. Seguido de esto se hace la selección de componentes para cada caso, los cuales permiten realizar de la mejor manera cada tarea estipulada.

7.1. Necesidades del sistema de comunicación

El sistema de comunicación es el encargado del intercambio de información entre los sistemas internos del robot y de este con el operario. Debido a que los demás submódulos requieren de una correcta comunicación externa e interna para funcionar correctamente, es importante asegurarse que se encuentre muy bien diseñado para poder evitar problemas en el traslado de la información. Las tareas que se deben realizar son las siguientes:

7.1.1. Control remoto y cableado

El robot debe operarse de manera remota, sin embargo la comunicación debe ser alámbrica ya que el fin de este es poder utilizarse en misiones de rescate y adentrarse en terrenos irregulares y peligrosos. Además, en muchas de estas situaciones las condiciones climáticas pueden ser bastante extremas. Es por esto que una comunicación inalámbrica puede traer consigo problemas en la comunicación o incluso el robot puede llegar a perderse en el campo. Por otra parte, un robot inalámbrico utilizaría baterías recargables. Esto presenta un nuevo problema, donde el robot se encuentra condicionado por el tiempo de carga restante para poder realizar las misiones. Evitando entonces perder al robot por una mala comunicación y también evitando limitar su funcionalidad al tiempo de carga, se ha decidido que este debe operar con una comunicación cableada, la cual asegura que no estarán ocurriendo problemas en la comunicación y se encontrará operando el tiempo necesario para realizar las misiones de rescate.

El problema que surge al querer utilizar una conexión cableada es la distancia, debido a que la mayoría de estándares de comunicación serial y paralela dejan de funcionar correctamente después de varios metros. Son pocos los estándares que permiten comunicaciones a distancia larga a un precio bajo, sin embargo la comunicación serial ha permitido en algunos casos trabajar a larga distancia bajo ciertas restricciones. Según las investigaciones realizadas y los protocolos/estándares para comunicación serial explorados, se ha seleccionado el estándar RS-485 como el protocolo a seguir en la comunicación entre el robot y el operario. Este estándar, RS-485, es un estándar de comunicación serial asíncrona. Se ha diseñado con la intención que pueda funcionar correctamente a distancias de hasta incluso 1200m utilizando una diferenciación de señales, por lo que si bien necesita del doble de cables para funcionar, se está alcanzando distancias bastante favorables para la operación del robot. Este estándar se presenta entonces como la solución adecuada para cumplir con el primer y segundo criterio de diseño del robot.

Para utilizar el RS-485, es necesario solamente tener dos circuitos integrados (un receptor y un transmisor), donde cada línea requiere de dos conexiones para realizar la diferenciación de señales. El hardware necesario para realizar la comunicación se puede implementar mediante un microcontrolador o computadora y la computadora del operario, ya que la mayoría de computadoras cuentan con puertos de comunicación. Debido a su fácil implementación y dimensiones pequeñas, tanto un Arduino como una Raspberry Pi pueden encargarse de la comunicación RS-485 con el operario (utilizando el hardware de la UART que ambas tienen). Debido a que se cuenta con un gran número de sensores y periféricos, los cuales podrían saturar al controlador con la cantidad de conexiones y actividad, se ha decidido utilizar tanto un microcontrolador Arduino como una computadora Raspberry Pi interconectados para poder realizar todas las actividades y comunicaciones del robot.

Es importante mencionar que las computadoras con puerto serial funcionan con un estándar RS-232, por lo que es necesario convertir esta señal a niveles de voltaje TTL que puedan aceptar los integrados de RS-485 antes de realizar la comunicación entre dispositivos en caso se desee utilizar un puerto *D-sub*. Si se utiliza un adaptador de USB a TTL se evitan estos problemas y se puede utilizar directamente el estándar RS-485. La interconexión se aprecia en la Figura 10. Es importante mencionar que existe una variedad de circuitos integrados que permiten utilizar estos estándares, donde la mayor diferencia radica en la velocidad máxima de transferencia de información y la habilidad half o full duplex de los mismos. Para la interconexión mostrada, se observan los integrados más populares para RS-485 (MAX485) y RS-232 (MAX232).

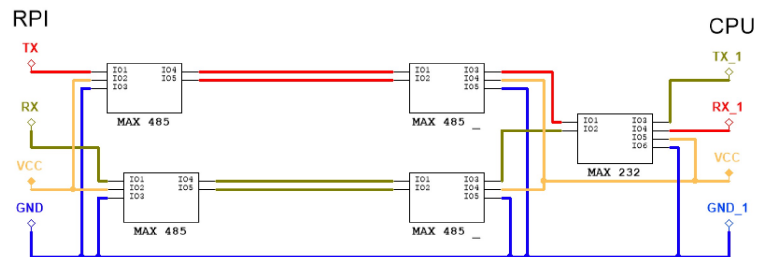


Figura 10: Comunicación entre computadoras

7.1.2. Potencia del sistema

El sistema de potencia debe ser capaz de entregar diferentes valores de voltaje y corriente. Esto se debe a que el robot todoterreno cuenta con una variedad de controladores, sensores y periféricos que tienen diferentes exigencias de potencia. Debido a que se tiene una alimentación cableada, la opción más segura y sencilla para cumplir con esta tarea es incluir una fuente de voltaje con los reguladores de voltaje DC-DC necesarios para suministrar la potencia adecuada a cada componente. Es importante mencionar que solo los actuadores requieren de una alta demanda de corriente, por lo que la demanda de potencia del sistema en general es baja. Además la fuente de voltaje no debe ser pesada ni grande para poder cumplir con la tarea de ubicarla en un robot.

Todos estos puntos permiten que se tenga entonces un diseño preliminar del sistema de

comunicación, donde lo más importante es utilizar el estándar RS-485, RS-232 (en caso se utilice un puerto serial de una computadora), una fuente de voltaje fija y reguladores de voltaje para poder cumplir con todas las tareas necesarias. Los componentes que componen al diseño preliminar se describen a continuación.

7.1.3. Componentes necesarios

En base a las tareas mencionadas anteriormente, se ha decidió que los siguientes componentes se estarán utilizando para el sistema de comunicación:

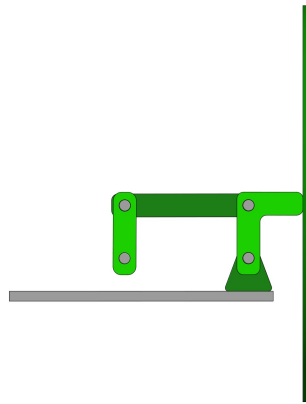
- 1 Computadora Raspberry Pi: Esta se encarga de realizar las instrucciones pesadas que requieren de una computación avanzada y de enviar/recibir la información del operario.
- 1 Microcontrolador Arduino: Este se encarga de realizar las instrucciones más livianas para poder apoyar a la Raspberry Pi.
- 4 Módulos RS-485: Estos se encargan de codificar y decodificar la información con el estándar RS-485 para poder intercambiar información entre el operario y el robot a largas distancias. Se utilizan 4 ya que por lo general cada módulo es half-duplex.
- 1 Convertidor de RS-232 a TTL: Este se encarga de poder transmitir y recibir la información de la computadora del operario cuando se utiliza un puerto serial.
- 1 Convertidor de USB a TTL: Este se encarga de poder transmitir y recibir la información de la computadora del operario cuando se utiliza un puerto USB.
- 1 Fuente de voltaje fija AC-DC: Esta se encarga de otorgar la potencia necesaria a todos los sistemas internos del robot.
- Reguladores de voltaje DC-DC: Estos se encarga de proporcionar los niveles adecuados de voltaje para cada sistema del robot.

7.2. Necesidades del sistema de comandos

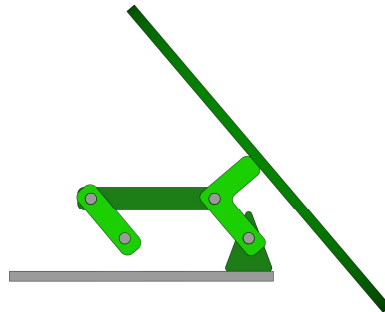
El sistema de control es el encargado del correcto funcionamiento de las ruedas, así como de convertir las señales del operario al movimiento deseado por el robot. Este último aspecto depende en que la comunicación entre el robot y el operario se desarrolle correctamente. Debido a que se requiere de un entorno en donde el operario pueda observar y comprender lo que está ocurriendo en todo momento, parte de las necesidades de este sistema es poder tener una interfaz que ofrezca la información relevante para poder realizar las operaciones de búsqueda y rescate. Las tareas que se deben realizar para satisfacer los criterios de diseño son las siguientes:

7.2.1. Movilización en el entorno

Para establecer la forma en que ocurre la movilización del robot se ha estudiado el movimiento de otro tipo de robots, en especial los robots todoterreno. El que ha presentado resultados más prometedores mientras que al mismo tiempo mantiene un diseño sencillo es el robot Puffer de la NASA, el cual se puede observar en la Figura 2. Este utiliza solamente dos ruedas a las cuales se les puede ajustar el ángulo de inclinación, además de controlar su dirección de movimiento. Las pruebas del Puffer han reflejado que cuando este necesita generar mayor fricción con la superficie que quiere escalar, basta con cambiar el grado de inclinación para que atraviese su obstáculo. Según el movimiento de este robot se ha decidió utilizar dos pares de servomotores. Cada par es responsable de movilizar una rueda y de cambiar su grado de inclinación. El poder tener esta destreza permite al robot funcionar como un robot todoterreno, lo que nos garantiza que se satisface el tercer criterio de que el operario sea capaz de observar y moverse en el entorno del robot con alta destreza en terrenos irregulares y variados. A continuación en la Figura 11 se presenta el mecanismo del funcionamiento de las ruedas desarrollado por el encargado del módulo mecánico. Para obtener información detallada del funcionamiento y diseño, se puede consultar el trabajo del módulo mecánico de este proyecto, *Diseño del sistema mecánico y chasis de un robot todoterreno para operaciones de búsqueda y rescate*.



(a) Ruedas del robot en posición original



(b) Ruedas del robot en posición inclinada

Figura 11: Funcionamiento de las ruedas

7.2.2. Control de los actuadores

El control de los actuadores se llevará a cabo por el microcontrolador Arduino. Los actuadores disponibles para poder realizar pruebas son los servos Dynamixel AX-12A y los HiTec D950TW (debido a que estos se encuentran disponibles para su uso en el departamento de Ingeniería Electrónica y Mecatrónica en la Universidad del Valle de Guatemala), por lo que el diseño y pruebas se realizará teniendo estos modelos específicos en mente. Estos servomotores tienen librerías bastante desarrolladas y completas en la plataforma de Arduino, por lo que su implementación resulta considerablemente más sencilla que si se realizara con la Raspberry Pi. Además, el bit-banging con puertos seriales en Arduino es sencillo de implementar de igual manera. Esto es importante ya que cada actuador requiere de un puerto UART para funcionar y tanto la Raspberry Pi como el Arduino solo cuentan con un puerto UART. Debido a esto, se requiere implementar puertos seriales con software para poder realizar comunicación tanto entre Raspberry Pi - Arduino como con los actuadores. Finalmente, es importante considerar que la Raspberry Pi estará realizando varias instrucciones y comandos pesados del operario y fotografía. Es por esto que se considera importante tratar de aliviar la carga en la Raspberry Pi, permitiendo que el envío de información constante ocurra sin problemas. Para esto se utiliza el controlador Arduino, el cual también realizará todas las operaciones del sistema de comandos. El control del robot llega del operario a la Raspberry Pi, donde es trasladado al Arduino. Este entonces se encarga de controlar los actuadores mediante las librerías implementadas de servomotores. Los servomotores encargados de realizar el desplazamiento del robot deben poder operar en rotación continua (AX-12A), mientras que los encargados de generar la inclinación de las ruedas deben proveer un torque alto (HiTec D950TW).

Ambos puntos permiten que se cumpla el objetivo más importante, el cuál es tener la habilidad de poder controlar y darle instrucciones al robot. Esto se debe a que el robot no presenta ningún tipo de control automático debido a que el campo en donde se desea utilizar requiere de un alto nivel de interacción con humanos.

7.2.3. Componentes necesarios

En base a las tareas mencionadas anteriormente, se ha decidido que los siguientes componentes se estarán utilizando para el sistema de control:

- 2 Servomotores de rotación continua
- 2 Servomotores de ultra torque

Como se puede observar, son pocos los componentes necesarios para realizar este sistema. Sin embargo, se requiere de varias acciones por parte de software y programación para ofrecer el funcionamiento adecuado: la programación de los actuadores en Arduino para controlar los movimientos, la implementación de una interfaz gráfica para observar los comandos y la codificación/decodificación de las diferentes instrucciones para los actuadores.

7.3. Necesidades del sistema de medición

El sistema de medición es el encargado de utilizar los sensores e imágenes del robot para obtener retroalimentación del entorno y ofrecer la información de valor a los rescatistas. Es por esto que en este submódulo la selección de los componentes correctos es de lo más vital para la correcta implementación del cuarto criterio. Las tareas que se deben realizar son las siguientes:

7.3.1. Recolección de información del entorno

La información necesaria para un rescatista y para el operario debe ser la suficiente para que se pueda llegar al sitio de interés con una clara idea de lo que está ocurriendo, ahorrando así tiempo y recursos de los involucrados. Como punto principal es importante conocer la temperatura del lugar, ya que esto le indica rápidamente al rescatista si puede ingresar o no a la zona. Este es un valor bastante común en los sistemas de medición y es un buen punto de partida en la selección de componentes necesarios para operaciones de búsqueda y rescate. Para cumplir con esto se desea utilizar un sensor de temperatura sencillo con un rango amplio de operación. Como referencia para la selección del mismo, se espera que el robot solo se exponga a temperaturas encontradas en la mayoría de zonas de la región de Guatemala.

Otro aspecto importante es la ubicación del robot, lo cual permite llegar al área de interés rápidamente. Este aspecto resulta también importante si se llega a perder comunicación con el robot, por lo que se puede recuperar si se cuenta con su ubicación final. Para cumplir con esto se desea utilizar un módulo GPS que ofrezca una ubicación aproximada del robot mediante la transmisión de sus coordenadas de latitud y longitud al operario.

Además de estos datos, el operario necesita visualizar al robot para poder controlarlo y direccionarlo. Es por esto que se necesitan cámaras en el robot. Se considera que una cámara infrarroja es ideal, ya que en lugares oscuros está resulta tan útil como en lugares iluminados. Como se espera que el robot frecuente sitios de accidentes, estos pueden tener problemas de iluminación. Para cumplir con esto se desea utilizar una cámara infrarroja iluminada artificialmente con LEDs infrarrojos, ya que es probable que el robot se encuentre en situaciones donde no hay luz natural presente.

Los rescatistas necesitan obtener información sobre la situación de las personas en el área del accidente, por lo que se considera útil tener una cámara térmica. Esta permite observar si hay señales térmicas que revelen el paradero de una persona, además que pueden brindarle información útil al rescatista sobre la condición de la persona y el nivel de urgencia que es demandado sobre ellos. Acompañando a esta cámara se desea utilizar un sensor de CO₂, el cual resulta útil para evaluar tanto la toxicidad del ambiente como el estado de una persona que se pueda encontrar atrapada en un área cerrada. Se pueden utilizar sensores térmicos y de CO₂ convencionales que complementen la otra información presente para evaluar el daño en las personas y el ambiente al que se encuentran expuestos.

Es deseable también el poder saber sobre la situación del terreno en el que se encuentra navegando el robot, ya que se busca evitar que este se exponga a inclinaciones muy pronun-

ciadas y que se encuentre atrapado o sufra daños mayores. Es por esto que se desea utilizar una unidad de medición inercial (IMU), la cual combina por lo general un acelerómetro de tres ejes y un giroscopio de tres ejes. Estos valores permiten, mediante la introducción de cálculos y filtros, obtener los ángulos de rotación "xz z" del sistema. Estos ángulos ayudan entonces al operario a determinar la inclinación del terreno sobre el cual el robot se encuentra.

Tomando en cuenta entonces los criterios y requerimientos por el operario, el rescatista y las víctimas, se ha establecido los componentes necesarios para poder tener un robot rescatista que sea útil y funcional al cumplir con el cuarto criterio, el cual indica que el operario debe poder obtener información del entorno del robot que le permita llevar a cabo operaciones de búsqueda y rescate. Sin embargo, no ha sido posible obtener todos los componentes deseados ya que no se encuentran disponibles y se han tenido restricciones de tiempo en consecuencia de esto. Es por esto que, aunque son altamente recomendados, no se utilizan tanto la cámara térmica como la infrarroja en el desarrollo del proyecto. En lugar de estos se utiliza una cámara convencional.

7.3.2. Empaquetamiento de información y conexiones

La información que proviene de tantos componentes diferentes debe agruparse y enviarse para que el operario pueda recibirla y entenderla fácilmente. Es por esto que se debe juntar la información de utilidad, empaquetarla de manera que eficiente y comunicarla al operario. Esto es trabajo del sistema de comunicación, por lo que no se requiere de componentes adicionales. Un esquemático de las conexiones en este submódulo se muestra en la Figura 12 en la siguiente subsección. Aunque originalmente este tomaba en cuenta la selección original de componentes, los obtenidos difieren a los deseados o recomendados para el proyecto. No ha sido posible obtener todos los componentes deseados ya que no se encuentran actualmente disponibles y se han tenido restricciones de tiempo en consecuencia de esto. Es por esto que, aunque los mencionados en el inciso 7.3.1. son altamente recomendados, no se están utilizando tanto la cámara térmica como la infrarroja en el desarrollo del proyecto. En lugar de estos se utiliza una cámara convencional. Se recomienda que en una futura segunda fase se utilice dicha figura como un punto de partida para implementar las mejoras necesarias con los sensores recomendados anteriormente. El esquemático completo presenta numerosas partes y conexiones que pueden resultar un poco confusas, por lo que seguido del esquemático se explican brevemente las mismas. Los sensores específicos seleccionados y presentados se han decidió finalmente según el presupuesto y disponibilidad para el proyecto.

Además, estas conexiones se basan en el funcionamiento estándar de cada sensor y cámara en específico seleccionado, por lo que se presenta un diseño de cómo debe estructurarse los componentes en la Raspberry Pi y el Arduino para poder llevar a cabo de la manera más ordenada y eficiente la obtención de información, su empaquetamiento y envío al operario. Todo esto con la intención de que se interprete y utilice la información en operaciones de búsqueda y rescate.

7.3.3. Componentes necesarios

En base a todos los criterios que se mencionaron anteriormente, se ha decidió que los siguientes componentes se estarán utilizando para el sistema de medición:

- 1 Sensor de temperatura
- 1 Sensor de CO₂
- 1 Módulo de GPS
- 1 Cámara
- 1 IMU

7.4. Componentes finales a utilizar

A continuación se presentan los componentes finales que se utilizan para implementar los sistemas del módulo electrónico del robot de operaciones de búsqueda y rescate:

- 1 Computadora *Raspberry Pi 3 Modelo B+*
- 1 Microcontrolador *Arduino Leonardo*
- 4 Circuitos integrados *MAX485*
- 1 Circuito integrado *MAX3225*
- 1 Adaptador USB a TTL *FTDI232*
- 1 Fuente de voltaje fija AC-DC de 12V 5A
- 1 Módulo regulador de voltaje ajustable DC-DC *LM2596S*
- 2 Reguladores de voltaje ajustables *LM317T*
- 1 Sensor de temperatura contra agua *DS18B20*
- 1 Sensor de gases peligrosos *MQ-135*
- 1 Módulo GPS *GY-GPS6MV2*
- 1 Cámara *Raspberry Pi Camera V2 8MP*
- 1 IMU *MPU6050*
- 2 Servomotores *Dynamixel AX-12A*
- 2 Servomotores *HiTec D950TW*

Con los componentes finales desglosados, se procede a presentar en la Figura 12 un diagrama de las conexiones con los componentes finales. Esta configuración es la que se presenta en el prototipo, donde se detalla en los siguientes capítulos el proceso de diseño y la implementación de cada módulo. Además, para poder entender mejor el esquemático, se procede a detallar las diferentes conexiones del mismo a continuación:

- Conexiones rojas: Conexiones a las diferentes fuentes de voltaje.
- Conexiones negras: Conexiones a tierra.
- Conexiones celestes: Conexiones seriales con circuitos integrados y sensores.
- Conexiones rosadas: Conexiones seriales entre Raspberry Pi y Arduino.
- Conexiones moradas: Conexiones a pines analógicos y digitales.
- Conexiones verdes: Conexiones I2C.
- Conexiones cafés: Conexiones internas de circuitos integrados.

Con este diseño preliminar desarrollado, se procede a trabajar en cada sistema para poder diseñar e implementar sus distintas funcionalidades según las necesidades y componentes estudiados. Cada sistema funciona de manera independiente, sin embargo es la interconexión de estos la que permite obtener resultados finales y un prototipo funcional. El desarrollo de cada sistema se detalla en las siguientes secciones, donde se menciona los pasos y retos necesarios así como los resultados obtenidos que demuestran el cumplimiento con los criterios y objetivos específicos.

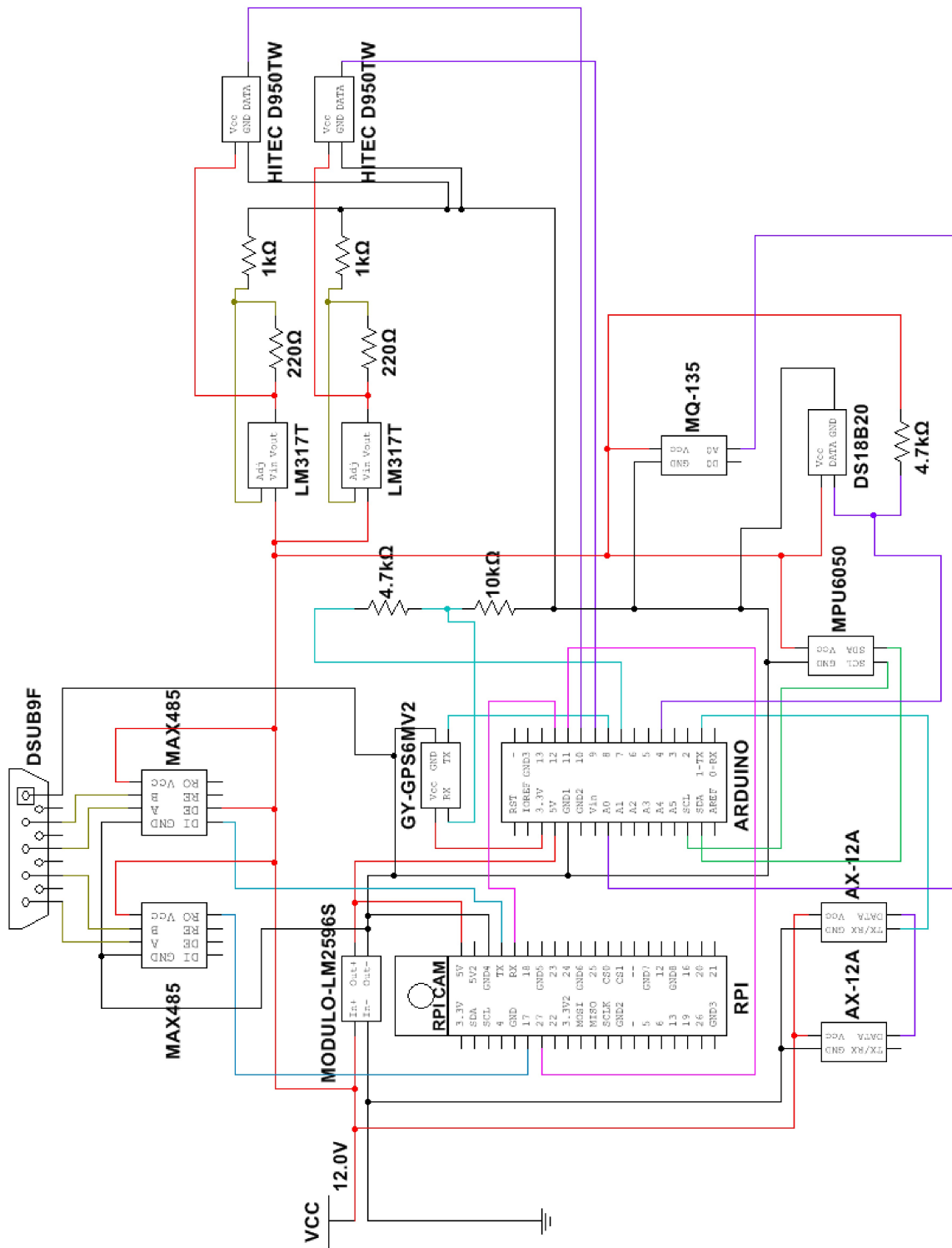


Figura 12: Conexión de sistemas internos del robot

Desarrollo del sistema de comunicación

Para poder desarrollar los sistemas de comandos y de medición de manera exitosa, el sistema de comunicación debe desarrollarse con especial atención debido a que comprende todo lo que son interacciones entre tanto la computadora del operario y la Raspberry Pi como la Raspberry Pi y el Arduino. Esto quiere decir que un sistema de comunicación exitoso evita problemas en los otros sistemas del módulo electrónico. Además de la comunicación, es importante realizar el desarrollo del sistema de potencia que asegure la alimentación adecuada para los otros sistemas del módulo electrónico. Esta sección se agrega al final para evitar confundirla con la documentación propia del sistema de comunicación.

8.1. Diseño del sistema

El diseño de este sistema involucra dos partes, la comunicación externa con el operario (Computadora - Raspberry Pi) y la comunicación interna del robot (Raspberry Pi - Arduino). Debido a que existe diferentes tipos de datos que se están transmitiendo entre sistemas (comandos, fotos e información de sensores), es importante que el diseño se realice con la intención de otorgar la importancia adecuada a cada dirección de traslado de información. Es por esto que no es arbitrario el uso de algún tipo de comunicación sobre otro para cierto traslado de información. Con esto en mente, a continuación se procede a desarrollar dicho diseño para la parte interna y externa:

8.1.1. Comunicación externa

La comunicación externa comprende las interacciones de la computadora del operario y el robot. Debido a que se desea utilizar el estándar RS-485, esta comunicación debe realizarse

mediante comunicación serial asíncrona. Esta comunicación se da por lo general en niveles de voltaje TTL (0V y 5V), por lo que el primer desafío es obtener dichos niveles por parte de la computadora del operario. Por lo general existen dos soluciones bastante comunes para poder utilizar comunicación serial con una computadora, las cuales son la utilización de un puerto serial *D-sub* o la utilización de un puerto USB convencional. La primera solución implica el uso del estándar RS-232, el cual utiliza por lo general niveles de $\pm 12V$. Esto implica que se necesita un convertidor de niveles de RS-232 a niveles aceptables para el RS-485 (TTL). El otro aspecto a considerar es que no todas las computadoras tienen este puerto, y su uso es cada vez más raro e incluso no existente en las computadoras y laptops modernas. La otra solución, mediante el uso de un puerto USB, implica la obtención de un adaptador USB a TTL. Por lo general, existe mayor libertad en el uso de RS-232 ya que es un estándar viejo de comunicación que tiene amplio uso y ofrece un baudaje bastante bueno. Sin embargo, los adaptadores USB a TTL modernos ya presentan baudajes igual de buenos. Para poder ofrecer una amplia variedad en el uso de computadoras por parte del operario, se ha realizado el diseño con la posibilidad de utilizar cualquier de las dos opciones. Para utilizar un puerto serial con el estándar RS-232, se debe utilizar un convertidor de RS-232 a TTL. Un circuito integrado que realiza esto de una manera adecuada es el MAX3225, ya que ofrece la conversión a niveles TTL mientras garantiza un velocidad de transmisión de datos de 1Mbps [15]. Una vez se cuenta con los niveles de voltaje apropiados, se utilizan los integrados MAX485. Estos permiten transferir datos a la misma velocidad que los MAX3225, por lo que no hay mayores sacrificios en este aspecto. La conexión de dicho sistema externo, que es la que acompaña a la computadora del operario, se presenta en la Figura 13. Como se puede observar, el responsable de interpretar la información del operario es la computadora Raspberry Pi dentro del robot. Esto se debe a que esta tiene mayor poder computacional que el Arduino, lo cual le permite codificar/decodificar y procesar información de una mejor manera, como se explica más adelante en la subsección de comunicación interna.

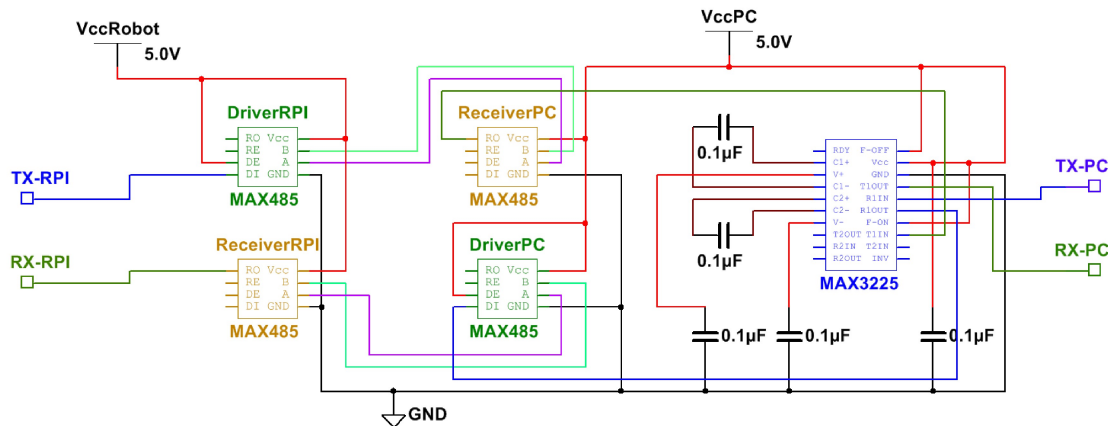


Figura 13: Conexión entre la computadora del operario y la Raspberry Pi con RS-232

Cuando no se desea utilizar el estándar RS-232, es decir no se utiliza un puerto serial, las conexiones resultan más sencillas. Este caso se observa en la Figura 14 a continuación:

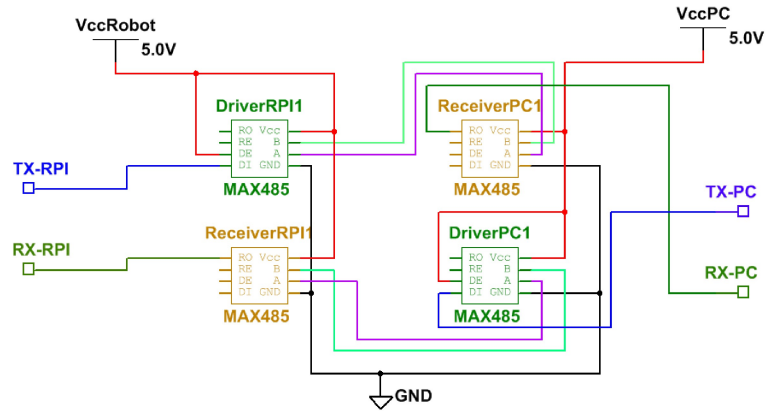


Figura 14: Conexión entre la computadora del operario y la Raspberry Pi sin RS-232

La razón por la que se utilizan dos integrados MAX485 de cada lado es porque estos funcionan bajo una modalidad half-duplex, por lo que para poder garantizar una comunicación sin interrupción en ambas vías se utiliza un integrado específicamente como driver y otro como receptor en cada lado de la comunicación, obteniéndose así un comportamiento full-duplex.

8.1.2. Comunicación interna

La comunicación interna comprende las interacciones de la Raspberry Pi y el Arduino. Como se explicó anteriormente, la Raspberry Pi se encarga de recibir y enviar información con el operario, por lo que se debe garantizar una comunicación efectiva de este con el Arduino para poder realizar todas las operaciones que le corresponden tanto al sistema de comandos como al de medición. Debido a que esta comunicación se realiza entre dos dispositivos que ya manejan los mismos niveles de voltaje TTL y que presentan puertos UART para comunicación serial asíncrona, se puede fácilmente usar para realizar las interacciones internas del robot. Es importante mencionar que, como se detalla más adelante en el capítulo del desarrollo del sistema de medición, algunos sensores utilizan también puertos UART. Tanto el Arduino como la Raspberry Pi solo presentan un par de puertos UART (TX y RX), por lo que surge la necesidad de implementar puertos UART adicionales para poder realizar la comunicación en ambas vías entre los tres dispositivos: computadora del operario, Raspberry Pi y Arduino. Esta implementación de puertos adicionales se realiza mediante el uso de *Bit-Banging* (también conocido como comunicación serial por software), la cual permite utilizar puertos digitales convencionales como puertos de comunicación serial mediante el uso de interruptos. Tanto la Raspberry Pi como el Arduino tienen librerías desarrolladas para realizar esta comunicación serial por software, por lo que su uso no resulta complicado de implementar. El esquemático de las conexiones para las comunicaciones internas se presenta en la Figura 15 a continuación:

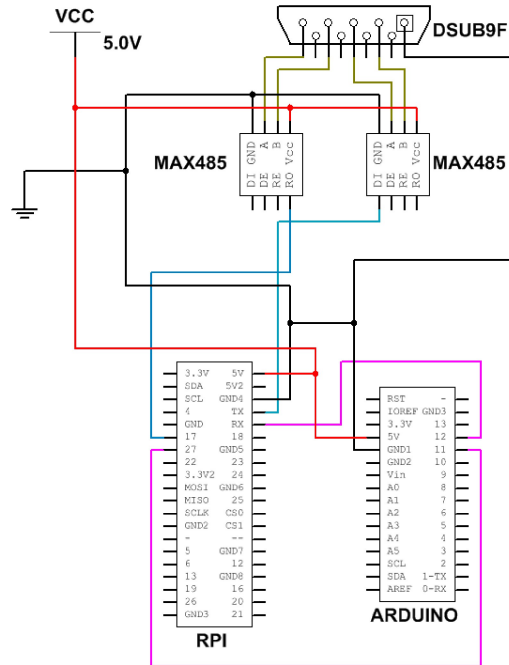


Figura 15: Conexiones internas de comunicación del robot

Como se observa, se utiliza el pin 17 y 27 de la Raspberry Pi para realizar comunicación serial por software. Estos pines se han seleccionado arbitrariamente y no presentan ningún beneficio particular. La librería utilizada para realizar esto en Python es *PIGPIO*, la cual permite utilizar los pines 2-27. Para el caso del Arduino, los pines con comunicación serial con software son los 11 y 12. Estos fueron escogidos ya que la librería utilizada, *SoftwareSerial*, permite utilizar en el Arduino Leonardo solo los pines 8, 9, 10, 11, 14, 15 y 16. Se han apartado los pines del UART por hardware en cada uno para recibir y enviar información de carácter importante de la siguiente manera:

- TX de Arduino: Envío de información a los servomotores Dynamixel AX-12A.
- RX de Arduino: Puerto dedicado a los servomotores Dynamixel AX-12A.
- TX de Raspberry Pi: Envío de información de sensores y cámara a la computadora del operario.
- RX de Raspberry Pi: Envío de información de comandos al Arduino.

Esta decisión se debe a que los pines seriales por software soportan un menor baudaje que los de hardware (hasta 115200 para el Arduino y hasta 250000 para la Raspberry Pi), por lo que las acciones que requieren de una alta velocidad de transferencia de datos deben manejarse con el UART por hardware.

8.2. Implementación del sistema

8.2.1. Implementación física

El sistema físico de comunicación se desarrolla siguiendo los esquemáticos y configuraciones explicadas en la sección anterior. Al utilizar los integrados MAX485 estamos garantizando que se puede realizar una conexión cableada entre los sistemas internos y externos de hasta 1200m (que es el alcance máximo con el estándar RS-485). Para poder conectar los sistemas son necesarios 5 cables largos: 2 para el primer par de integrados MAX485 half-duplex, otros 2 para el segundo par y finalmente uno para conectar la tierra del robot con la del operario. Este último es importante ya que al tener la misma referencia se evita recibir información falsa o lecturas erróneas en los integrados MAX485. La forma más sencilla de hacer estas conexiones es con un cable DB9, el cual ya trae 4 pares de cables trenzados y 1 de tierra. Si bien no se utilizan todos, estos cables se pueden encontrar fácilmente en el mercado incluso a largas distancias y manejan un calibre adecuado para envío de datos (24 AWG). Esta es una opción más favorable y económica que buscar un cable hecho a la medida específicamente para los requisitos de este proyecto. Además, el departamento de Ingeniería Electrónica y Mecatrónica en la Universidad del Valle de Guatemala ofrece cable DB9 para los proyectos de los estudiantes, por lo que facilita su adquisición. Un esquema de la conexión realizada entre los sistemas internos y externos con dicho cable se presenta en la Figura 16 para la implementación con el uso de un puerto serial por parte del operario, mientras que en la Figura 17 se aprecia la misma conexión sin el integrado necesario para la conversión de RS-232 a TTL:

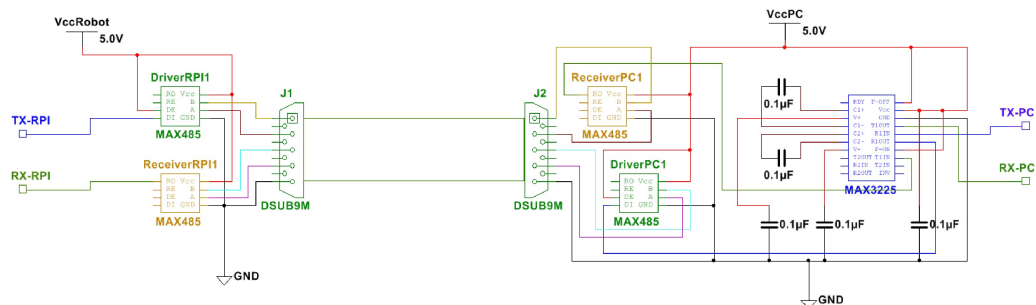


Figura 16: Conexión entre el operario y el robot con RS-232

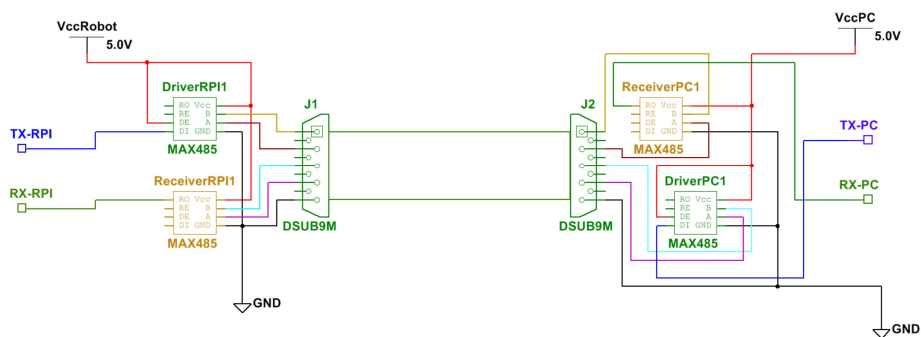


Figura 17: Conexión entre el operario y el robot sin RS-232

Estos esquemáticos y la Figura 15 representan en su totalidad las conexiones físicas necesarias para el módulo de comunicación. Para poder compararlas con los resultados en físico, en el capítulo *Resultados Finales* se pueden observar las conexiones en las placas de pruebas internas y externas.

8.2.2. Implementación de software

El software utilizado para poder desarrollar las funciones de comunicación es tanto el lenguaje Python para las computadoras y como el lenguaje Arduino (C/C++) para el microcontrolador Arduino. Estas permiten, mediante el uso de librerías *open-source*, el control y fácil programación de muchas de las operaciones necesarias para poder efectuar la comunicación serial asíncrona.

8.2.3. Programación de interacciones operario - Raspberry Pi

Toda la programación necesaria para realizar estas operaciones se desarrolla en Python, ya que ambas computadoras pueden utilizar dicho lenguaje sin ningún problema. Para poder establecer una comunicación serial con los integrados MAX485 se está utilizando la librería *pySerial* de Chris Liechti. Esta permite establecer un acceso a los puertos seriales de la computadora, siendo estos los pines del UART para la Raspberry Pi y los puertos seriales de la computadora disponibles mediante los métodos indicados en la subsección *Control Remoto y Cableado* de la sección *Necesidades del Sistema de Comunicación* en el capítulo 7.

Tanto la computadora del operario como la Raspberry Pi utilizan una inicialización de comunicación serial estándar para comunicación serial de bytes de 8 bits sin bits de paridad y con un bit de final. Sin embargo, un detalle de esta librería es que no se incluye una detección de errores más allá de la establecida para este método de comunicación. Debido a que la información proveniente de los sensores y cámara es de un gran tamaño, se ha decidido implementar una detección de errores CRC para la transferencia de los datos de la Raspberry Pi a la computadora del operario. Para realizar esto, es necesario realizar tanto un codificador (en la Raspberry Pi) como decodificador (en la computadora del operario). El codificador funciona mediante la operación de división módulo2 entre el paquete de datos que se desean enviar y una clave de seguridad, de la cual se obtiene un nuevo paquete de datos con codificación CRC. Para realizar esto se ha implementado la división módulo2 mediante la operación XOR, que permite realizar operaciones entre bits. El decodificador funciona de manera muy similar mediante la operación del módulo2 con las operaciones XOR entre los datos recibidos y la llave, sin embargo el resultado obtenido para este caso debe ser un arreglo de ceros de la longitud de la llave. Si el resultado no es igual a un arreglo de ceros, ha ocurrido un problema en la comunicación. Como se explicó anteriormente, esto solo es necesario en una vía de comunicación (Raspberry Pi -> Computadora) ya que el envío de comandos de la Computadora a la Raspberry Pi es de tan solo un bit, como se explica en el capítulo 9, por lo que es excesivo aplicar esta detección de errores.

Según lo mencionado en la subsección *Comunicación Interna* de la sección *Diseño del Sistema* de este capítulo, la lectura de datos por parte de la Raspberry Pi provenientes de la computadora del operario se realiza con un puerto serial por software, el cual se observa

en la Figura 15 con el pin 17. Para poder implementar dicho comportamiento en el pin se utiliza la librería *PIGPIO*, la cual permite utilizar con libertad los pines 2-27 para realizar una diversidad de actividades de comunicación. Para poder activar el recibo de datos se utiliza la serie de comandos *bb-serial-read* de esta librería.

El código tanto de la Raspberry Pi como de la computadora del operario se encuentra disponible en el repositorio detallado en la sección de *Anexos*.

8.2.4. Programación de interacciones Raspberry Pi - Arduino

La programación necesaria para las interacciones internas del robot resulta más sencilla debido a que el Arduino presenta un comportamiento más sencillo en lo que es la comunicación serial tanto por hardware como software (ya a que las librerías de este son bastante amigables al usuario). Según lo observado en la Figura 15, se puede apreciar que no se utiliza ninguno de los pines de comunicación serial por hardware del Arduino (debido a la necesidad de estos para los servomotores Dynamixel). Esto implica que se debe utilizar la librería *SoftwareSerial*, la cual permite utilizar los pines digitales 8, 9, 10, 11, 14, 15 y 16 en el Arduino Leonardo para RX y cualquier pin digital para TX. En este caso se están utilizando los pines 11 para RX y 12 para TX. La aplicación de esta librería es bastante sencilla, donde solo se necesita realizar la inicialización apropiada y se pueden utilizar los pines con las mismas funciones que las de las de la comunicación serial por hardware.

La programación de la Raspberry Pi es similar a la mencionada en la subsección anterior, donde se utiliza la librería *PIGPIO* para realizar la comunicación serial por software. En este caso, se utiliza el pin RX de la UART para leer información del Arduino (donde basta utilizar la librería *pySerial* de la misma manera que en la comunicación de la subsección anterior) y el pin 27 con comunicación serial por software. Para utilizar el pin 27 en modo escritura, se debe utilizar la serie de comandos *wave* de esta librería. Esta envía una serie de pulsos mediante interruptos para simular la comunicación serial por hardware cuando se reciben comandos de la computadora del operario con la intención de transferirlos al Arduino.

De igual manera, el código tanto de la Raspberry Pi como del Arduino se encuentra disponible en el repositorio detallado en la sección de *Anexos*.

8.3. Sistema de potencia

El sistema de potencia tiene la intención de poder alimentar todos los sistemas internos de la manera más práctica y sencilla. Si bien se ha contemplado el utilizar baterías, esta opción no se ha tomado debido a que no se desea que el funcionamiento del robot se encuentre limitado por el tiempo de alimentación de estas. Es así como se ha optado por el uso de una fuente de voltaje, como se describe en el capítulo anterior. Esta fuente ha de ser alimentada por la red eléctrica (120V AC) para poder proveer la potencia adecuada a cada sistema. La demanda del módulo electrónico se desglosa a continuación:

- Raspberry Pi 3 B+: Utiliza entre 200 y 500 mA para operar. Requiere de 5V.
- Arduino Leonardo: Utiliza alrededor de 50 mA para operar. Requiere de 5V.
- Servomotores *AX-12A*: Utiliza alrededor de 900mA cada uno para operar. Requiere de 9-12V.
- Servomotores *D950TW*: Utiliza alrededor de 500mA cada uno para operar. Requiere de 7V.
- Módulo *GY-GPS6MV2*: Utiliza alrededor de 50mA. Requiere de de 2.7-3.6V.
- Sensor *MQ-135*: Utiliza alrededor de 150mA. Requiere de 5V.
- Sensor *DS18B20*: Utiliza alrededor de 1.5mA. Requiere de 5V.
- Sensor *MPU6050*: Utiliza alrededor de 2mA. Requiere de 5V.
- Cámara *Raspberry Pi Camera V2 8MP*: Este se alimenta de la Raspberry Pi.

Tomando estas consideraciones en cuenta, se está trabajando con una demanda de corriente de 3.3-3.5A. Para poder cubrir con la demanda, y tener un margen de seguridad, se ha utilizado la fuente de voltaje fija de 12V 5A. Los únicos sistemas que se pueden alimentar directamente con esta fuente son los servomotores *AX-12A*. Los otros sistemas requieren de reguladores de voltaje para reducirlo. Una solución sencilla es la utilización de reguladores ajustables *LM317T*, sin embargo estos proveen solamente hasta 1A de corriente. Otra solución es adquirir módulos reguladores ajustables de mayor corriente, como el *LM2596S* que provee 3A, para alimentar los sistemas más importantes. Es así como se ha decidido utilizar ambas soluciones, donde el módulo se encarga de alimentar todos los sistemas de 5V (que consumen juntos menos de 3A) y los reguladores de 1A alimentan a los servomotores *D950TW* a 7V al ajustarlos con las resistencias adecuadas. Esto se debe a que es mucho más barato los reguladores que los módulos, por lo que se quiere evitar un mayor gasto en componentes.

A continuación se muestran entonces los esquemáticos para el sistema de potencia que ilustran las conexiones de estos para los actuadores en la Figura 18 y para los sensores en la Figura 19.

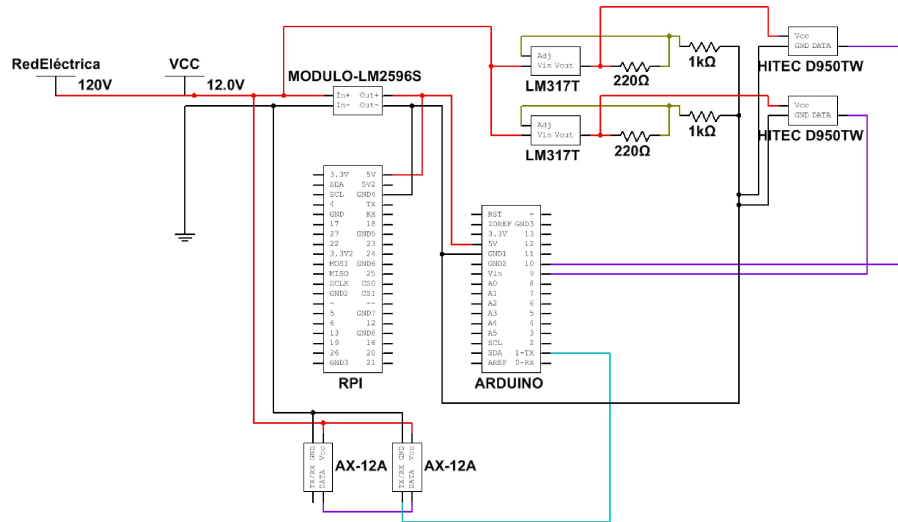


Figura 18: Esquemático del sistema de potencia con los actuadores

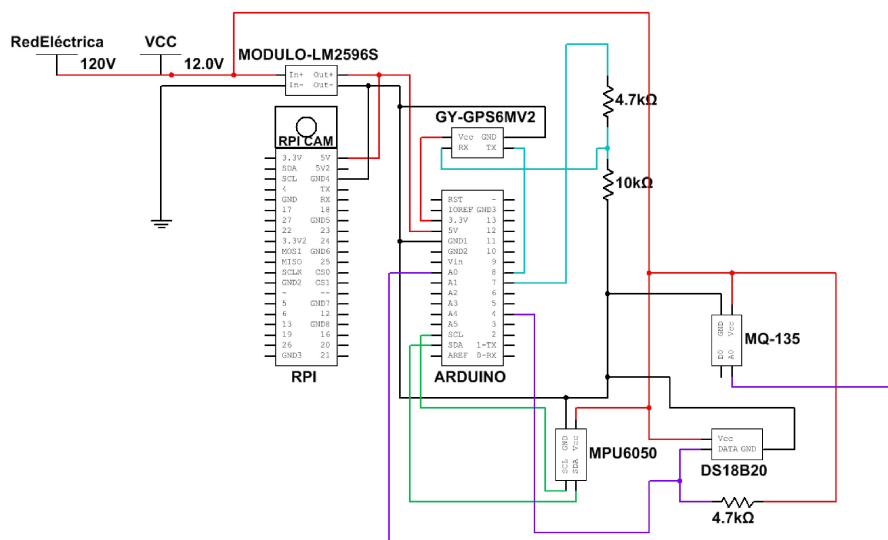


Figura 19: Esquemático del sistema de potencia con los sensores

Es así como entonces se logra alimentar todos los sistemas internos del robot con el uso de una fuente de voltaje y reguladores según las conexiones mostradas. La fuente se conecta a la red eléctrica mediante un cable calibre 14 AWG.

Desarrollo del sistema de comandos

El sistema de comandos, tal como se menciona en el capítulo 7, debe cumplir con la función de operar las ruedas correctamente, así como de convertir las señales del operario al movimiento deseado por el robot. Es así como se se ha establecido un diseño del sistema que garantiza el cumplimiento de dichas funciones según los objetivos específicos establecidos.

9.1. Diseño del sistema

El diseño de este sistema involucra dos partes igual de importantes: la interfaz gráfica operario-robot y el control de los actuadores. La interfaz se asegura el operario pueda observar y cerciorarse de las instrucciones que se están transmitiendo al robot, mientras que el control establece los protocolos y conexiones necesarias para que el robot pueda ejecutar dichas instrucciones. A continuación se detalla el diseño y proceso para cada parte:

9.1.1. Interfaz gráfica de datos de salida

La interfaz gráfica debe desplegarse en la computadora del operario, la cual, además de realizar la comunicación, debe desplegar retroalimentación de las instrucciones y comandos que el robot está realizando. Para poder realizar esto se ha decidido utilizar el mismo lenguaje de programación donde se realizó la configuración de la comunicación en la computadora del operario, Python. Se toma esta decisión ya que se han desarrollado numerosas librerías que permiten el desarrollo de interfaces gráficas robustas, así como la fácil integración de las mismas con la programación desarrollada para el sistema de comunicación.

Se ha propuesto que el robot opere con seis diferentes modos de desplazamiento: avanzar, retroceder, girar a la izquierda, girar a la derecha, avanzar a la izquierda y avanzar a la

derecha. Además de esto, el robot funciona en el modo de operación normal o inclinado, según la demanda de la situación en la que se encuentre. La interfaz debe ser capaz de mostrar el modo de desplazamiento y de operación actual de una manera fácil y entendible, ya que no se desea que el operario requiera de un avanzado nivel de entrenamiento para poder hacer uso del robot. Se ha estudiado las diferencias entre el uso de líneas de comando e interfaces gráficas, tales como su facilidad y control, y se ha podido observar que el uso de una interfaz gráfica permite a un usuario entender el uso de un programa de manera rápida ya que es visualmente intuitiva y ofrece un mayor grado de control sobre la funcionalidad del mismo [16]. Es por esto que se utiliza una interfaz de despliegues para operar y observar el funcionamiento del robot. A continuación se presenta un diseño preliminar de una GUI que ofrece la funcionalidad necesaria para llevar a cabo el trabajo del robot:

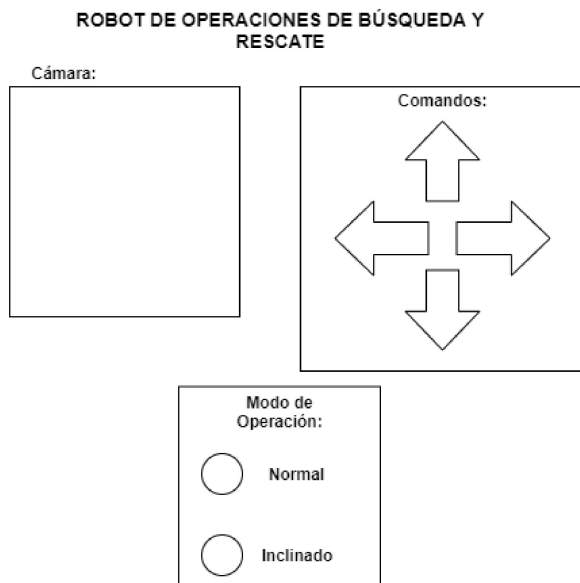


Figura 20: Diseño preliminar para comandos del robot

El objetivo de esta GUI es que los comandos que envíe el operario se puedan observar gráficamente con las flechas, las cuales se iluminan según la dirección a la que se está moviendo el robot. El mismo caso ocurre con los indicadores del modo de operación, donde se indica claramente el modo actual del robot. La dirección de movimiento y el cambio de modo de operación se realizan con diferentes teclas del teclado, las cuales despliegan la información en la interfaz, seguido de su envío mediante el sistema de comunicación a través de los procesos detallados en el capítulo anterior.

9.1.2. Control de los actuadores

El control de los actuadores se realiza en Arduino, ya que este controlador permite utilizar librerías existentes que facilitan el control y programación de los servomotores Dynamixel. Estos servomotores utilizan protocolos avanzados para su configuración, los cuales se reducen a simples comandos con estas librerías. Debido a que la Raspberry Pi es la encargada de recibir toda la comunicación, los comandos enviados a esta por el usuario son recolectados y enviados al Arduino mediante un segundo envío de datos, tal como se indica en el

capítulo anterior. Es así como estas instrucciones hacen su recorrido de la computadora del operario hasta el Arduino. Los servomotores AX-12A son los encargados del desplazamiento del robot, y solamente realizan dos acciones: rotar continuamente a determinada velocidad y detenerse. Por otro lado, los servomotores HiTec son los encargados de ajustar el ángulo de inclinación del robot, donde se encargan de recibir cierto ángulo de posición y fijar las ruedas según el mecanismo propuesto en la Figura 11 para una continua operación inclinada. El funcionamiento que ofrecen entonces los actuadores es establecer los diferentes modos de operación a los que se desea someter al robot en determinado momento. El control del desplazamiento depende de la velocidad a la que se somete cada rueda. Esto entonces permite ya sea avanzar, retroceder o girar en alguna dirección al robot. Para hacer girar el robot basta con detener una rueda mientras la otra avanza, permitiéndolo moverse en su eje. El grado de efectividad de este comportamiento depende de las decisiones y funcionamiento del módulo mecánico del proyecto. Por otro lado, para hacer avanzar el robot hacia la izquierda o derecha es necesario que una rueda gire más despacio que la otra.

Con el sistema de comunicación en operación, basta con codificar y decodificar los comandos en la computadora del operario, Raspberry Pi y Arduino para llevar a cabo el control de los actuadores. Dichas codificaciones se explican más adelante en la siguiente sección de implementación.

9.2. Implementación del sistema

9.2.1. Interfaz gráfica de datos de salida

La interfaz gráfica que se desarrolla para este proyecto no debe influir drásticamente en el funcionamiento de los otros módulos. Es decir, se busca que la interfaz del robot no introduzca retrasos en la comunicación de la información de la computadora del operario a los sistemas internos del robot, ya que esto afecta el funcionamiento de todos los módulos. Con esta idea en mente, se ha decidido desarrollar la interfaz gráfica con la librería *PyQt* de *Riverbank Computing*. Esta librería es bastante robusta, se introduce fácilmente a programas ya existentes de Python y no introduce mayores retrasos a dichos programas. Un aspecto adicional que resulta bastante útil es que esta librería permite establecer el diseño estético de la interfaz en un diseñador que incluye llamado *QtDesigner*. Esto facilita la creación, posicionamiento y configuración de los diferentes widgets necesarios para desplegar toda la información deseada, donde posteriormente se genera el código de dicha interfaz para Python y se puede agregar la funcionalidad con programación adicional. Basándose en el diseño preliminar observado en la Figura 20, se ha desarrollado la siguiente interfaz que ofrece todo el funcionamiento descrito en la subsección *Interfaz Gráfica* de la sección anterior:

Agregando el funcionamiento a este diseño, se obtiene el siguiente comportamiento en la interfaz según las instrucciones que se ingresan en el teclado para su envío al robot:

- Avanzar: Iluminación de la flecha superior.
- Retroceder: Iluminación de la flecha inferior.
- Giro izquierda: Iluminación de la flecha izquierda.

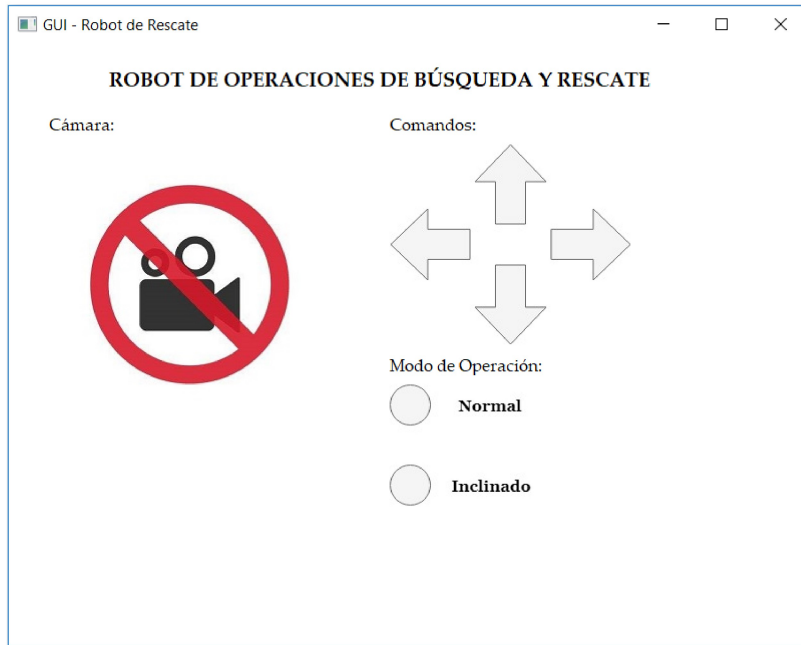


Figura 21: Interfaz gráfica final para comandos del robot

- Giro derecha: Iluminación de la flecha derecha.
- Avanzar izquierda: Iluminación de la flecha superior e izquierda.
- Avanzar derecha: Iluminación de la flecha superior y derecha.
- Modo de operación normal: Iluminación del indicador "normal".
- Modo de operación inclinado: Iluminación del indicador "inclinado".

Con estos indicadores, el operario puede obtener un buen entendimiento de las acciones que se le están solicitando al robot. Sin embargo, estos comandos no se seleccionan directamente en la interfaz; como ya se mencionó, esta solo despliega información. Los comandos se establecen con las teclas de las flechas y con la barra espaciadora. Las flechas se encargan de realizar los seis movimientos disponibles, mientras que la barra espaciadora cambia el modo de operación. Es importante mencionar que con las funciones de movimiento, el robot solo va a mover mientras la o las teclas de movimiento se mantengan presionadas. Una vez estas teclas se suelten, el robot se detiene automáticamente (enviando el comando de *Detenerse*). Además de enviar las respectivas señales a la interfaz para iluminar los indicadores, el seleccionar los comandos con el teclado también envía las señales al robot mediante la codificación establecida para el control de actuadores.

9.2.2. Control de actuadores

El control de actuadores inicia desde la computadora del operario, donde cada comando que se desea enviar al robot es representado por un carácter diferente que ocupa un byte bajo la codificación *UTF-8*. Dicha codificación de los caracteres se detalla a continuación:

- Detenerse: b'0'
- Avanzar: b'1'
- Retroceder: b'2'
- Giro izquierda: b'3'
- Giro derecha: b'4'
- Avanzar izquierda: b'7'
- Avanzar derecha: b'6'
- Cambio de modo de operación: b'5'

Cada vez que se detecta una nueva instrucción por parte del operario, la computadora envía este byte de instrucciones a la Raspberry Pi, la cual lo envía al Arduino bajo la misma codificación. El Arduino es el responsable de realizar el control de los actuadores. Para esto se utiliza la librería *AX-12A Servo Library*, la cual permite cambiar la velocidad y modo de operación de estos servomotores. Los servomotores se colocan en modo de rotación continua para poder girar indefinidamente, y las velocidades que se le colocan a cada uno depende del comando que ha establecido el operario. Estos motores se conectan en *daisy-chain* al puerto TX del Arduino a un baudaje de 9600, por lo que solamente se utiliza un puerto para operar los dos. Además del puerto TX, es necesario establecer un puerto que funcione como memoria de la dirección de la rotación, donde se puede utilizar cualquier pin digital. En el caso de este proyecto se ha decidido utilizar el pin 6. Seguido de la configuración de los servomotores AX-12A, se tiene la conexión y configuración de los de ultra torque. Debido a que el cambio de estados solo opera con dos ángulos, los normales y los inclinados, solo es necesario contar con dos tipos de funcionamiento. Si bien estos por lo general se pueden conectar a cualquier puerto digital común del Arduino, el uso de la librería estándar de servos de Arduino, *Servo*, genera conflictos con la librería de comunicación serial por software *SoftwareSerial* ya que esta deshabilita el uso de interruptos temporalmente, algo necesario de manera constante en la librería de servos, en lo que se realiza el envío de datos. Debido a esto, se está utilizando la antigua librería de servos para Arduino, llamada ahora *Servo2*, la cual no genera problemas al no requerir del uso de interruptos. Sin embargo, el sacrificio que conlleva utilizar dicha librería es que los servos de ultra torque deben conectarse obligatoriamente a los pines digitales 9 y 10, por lo que así se ha realizado en la implementación de los mismos. Las conexiones finales a los actuadores se muestran en el siguiente esquemático:

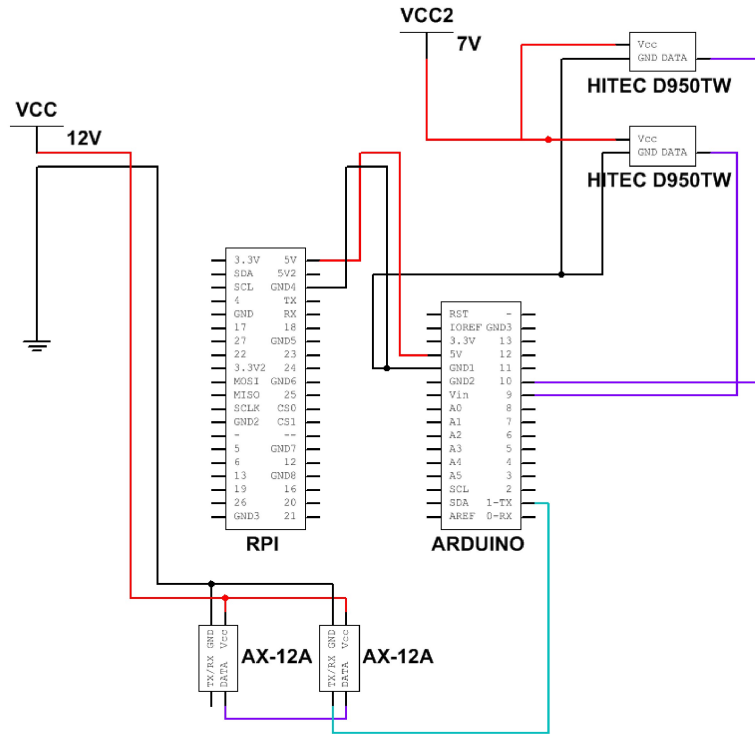


Figura 22: Esquemático de conexiones de actuadores

Es así como obtenemos un funcionamiento adecuado del sistema de control, donde los comandos del operario llegan a establecer claramente las acciones a tomar por el Arduino respecto a las características de operación de los servomotores. Debido a que las velocidades de transferencia de datos son rápidas, la comunicación entre dispositivos no se percibe tan fácilmente.

Desarrollo del sistema de medición

El sistema de medición debe ofrecer al operario información relevante del entorno del robot, ya que como robot de operaciones de búsqueda y rescate se desea que este demuestre tener ventajas que lo presente como una solución atractiva para su uso. Es por esto que el desarrollo debe tomar en cuenta que todos los sensores deben poder alimentarse y utilizarse con facilidad, además de que la información que estos retornen sea de fácil acceso para el operario.

10.1. Diseño del sistema

El diseño del sistema de medición se divide en dos partes fundamentales: la interfaz gráfica operario-robot y la operación y empaquetamiento de datos de los sensores. La interfaz permite al operario observar la información captada por los sensores de una manera atractiva y fácil de entender. Esta se anexa a la interfaz de comandos para poder ofrecer una interfaz completa que presente tanto los datos de salida (comandos) como de entrada (sensores) por el operario. Por otra parte, la operación y empaquetamiento de datos asegura en su diseño que cada sensor se encuentra conectado y operando correctamente para luego agrupar dicha información y enviarla al usuario. Con estos criterios en mente, se presenta el diseño de ambas partes a continuación:

10.1.1. Interfaz gráfica de datos de entrada

La interfaz gráfica debe desplegar en la computadora del operario la información obtenida por los sensores, por lo que es necesario desarrollar etiquetas y apropiadas para cada sensor. En total se está trabajando con cinco sensores: temperatura, CO₂, GPS, cámara e IMU.

La sección de cámara ya ha sido desarrollada en la interfaz gráfica pasada ya que es la que más espacio ocupa. Los otros cuatro sensores solamente otorgan datos numéricos, por lo que su implementación con etiquetas simples de texto es suficiente para desarrollar una interfaz simple y funcional. Tanto la información de temperatura y como el nivel de CO2 son solamente un dato, mientras que para el GPS y la IMU se esperan obtener dos datos. Para el caso del GPS estos son la latitud y longitud, mientras que con la IMU se desea obtener el ángulo roll (Θ) y pitch (Φ) del sistema para conocer su inclinación. Teniendo estos criterios en mente, se procede a realizar un diseño preliminar de la información que se desea desplegar al operario en la interfaz gráfica:

**ROBOT DE OPERACIONES DE BÚSQUEDA Y
RESCATE**

TEMPERATURA:

NIVEL DE CO2:

INCLINACIÓN:

ÁNGULO ϕ :

ÁNGULO θ :

LATITUD: LONGITUD:

Figura 23: Diseño preliminar para información de sensores

Como se mencionó, solo se está desplegando información con etiquetas, donde es trabajo del operario interpretar dicha información para evaluar la situación en una operación de búsqueda y rescate sobre el paradero de las personas. Esta información debe llegar a la computadora del operario a través de la Raspberry Pi, donde es trabajo del empaquetamiento de datos enviar la información de manera correcta. Este punto se discute en la siguiente subsección de diseño.

10.1.2. Operación de sensores y empaquetamiento de información

Los sensores utilizados funcionan con una diversidad de protocolos de comunicación. Es por esto que existe una cantidad considerable de conexiones que garantizan que estas funcionen correctamente. Para poder diseñar correctamente el espacio y puertos a utilizar para cada sensor, se procede a explicar el funcionamiento de cada sensor:

- Sensor de temperatura *DS18B20*: Este sensor funciona mediante el protocolo de comunicación 1-Wire, donde solo requiere de una conexión a un pin digital cualquiera para transmitir la información de la temperatura registrada, además de una conexión a voltaje (5V) y a tierra. El rango de temperatura que permite este sensor es de -55

a 125 °C, con una precisión de ± 0.5 °C [17]. Se considera que el robot no se va a encontrar u operar en temperaturas fuera del rango.

- Sensor de gases peligrosos *MQ-135*: Este sensor tiene la habilidad de detectar una gran cantidad de gases en el aire tales como NH₃, NO_x, alcohol, benceno, humo y CO₂. Debido a que cada gas presenta un curva característica de sensibilidad para el sensor, basta con obtener el comportamiento del CO₂ para establecer su lectura. El problema es que su uso no se puede realizarse exclusivamente para CO₂, por lo que si se encuentra una diversidad de gases juntos, la lectura tiende a verse ligeramente afectada. Su alimentación es de 5V y el pin A0 se conecta a un pin analógico para poder obtener la lectura de resistividad del sensor [18].
- Módulo GPS *GY-GPS6MV2*: Este módulo permite obtener una variedad de información, donde lo más útil su ubicación en latitud y longitud. Para obtener esta información es necesario que este se conecte a un satélite, donde una vez se tiene una comunicación efectiva ya se puede solicitar los datos del mismo. La conexión al satélite puede ser un poco lenta si se encuentra en una ubicación con mala recepción. Este módulo opera bajo voltajes de hasta 3.6V y utiliza comunicación con una UART para transmitir información [19].
- IMU *MPU6050*: Este sensor consta de un acelerómetro y giroscopio, los cuales permiten obtener dos grados de inclinación mediante diferentes relaciones matemáticas y filtros. La comunicación la realiza mediante I2C y se alimenta de 5V [20].
- Cámara *Raspberry Pi Camera V2 8MP*: Esta cámara se conecta al puerto especial CSI que incluye la Raspberry Pi para utilizar cámaras mediante un cable cinta. Debido a esto, su utilización con la Raspberry Pi es bastante sencillo y ya se incluye una diversidad de librerías en la misma para utilizarlo en Python [21].

Teniendo el criterio de cada componente en mente, se puede ajustar fácilmente los sensores de temperatura, CO₂, IMU y el módulo GPS en el Arduino, donde el único sensor que requiere obligatoriamente el uso de la Raspberry Pi es la cámara. El Arduino dispone de una variedad de pines digitales y analógicos, un puerto I2C y la disponibilidad de implementar puertos UART adicionales. Es por esto que se utiliza el Arduino para la obtención de todos los datos, mientras que la obtención de fotografías queda en manos de la Raspberry Pi. Si bien la cámara puede tomar video, el envío de este mediante comunicación serial no es una opción realista. Es así como, para permitir que el operario pueda observar el entorno del robot, se puede enviar fotografías de manera consecutiva cada determinado tiempo, lo cual si es posible con comunicación serial a un baudaje alto.

El empaquetamiento de información se puede realizar de manera sencilla ya que en el Arduino todos los datos obtenidos equivalen a números y símbolos. Se puede proceder a guardar todos los datos en diferentes variables numéricas, donde una vez se han obtenido todos se puede introducir cada uno a un arreglo de caracteres que los contenga todos. Este arreglo de datos se puede enviar a la Raspberry Pi mediante comunicación serial, donde se anexa al arreglo de datos que la Raspberry Pi ya ha creado con la información de la fotografía que se desea enviar. Una vez se tiene el arreglo con todos los datos de sensores y cámaras, este se envía al operario, el cual lo descompone en sus diferentes sensores para desplegarlo en la interfaz gráfica.

Es así como, entendiendo los criterios de cada sensor y el empaquetamiento, se procede a implementar dichos sistemas en la siguiente sección.

10.2. Implementación del sistema

10.2.1. Interfaz gráfica de datos de entrada

La interfaz se ha desarrollado sobre la interfaz de comandos, donde se ha agregado las etiquetas correspondientes según el diseño preliminar ilustrado en la Figura 23. Además de las etiquetas ilustradas, se ha incluido una imagen que permite entender la inclinación del robot bajo los ángulos de roll (Θ) y pitch (Φ), ya que estos no son intuitivos para todos. Con la idea de facilitar su interpretación, se ha agregado esta a la interfaz del operario. Para realizar la interfaz, se ha utilizado nuevamente la herramienta *QtDesigner* de la librería *PyQt* para obtener el diseño deseado e importarlo a código de Python. Seguido de esto, se ha implementado la programación que desempaca la información obtenida del robot para desplegarla en cada etiqueta. El proceso en detalle se menciona en la siguiente subsección. La interfaz final desarrollada se puede apreciar en la siguiente figura:

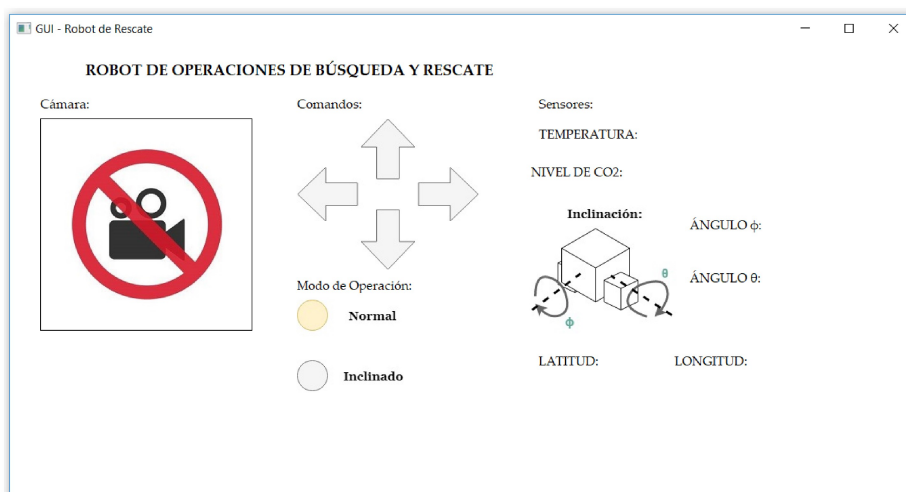


Figura 24: GUI final con datos de entrada y salida

Es así como entonces se tiene una interfaz completa que permite observar e interpretar tanto los datos de entrada como los de salida, permitiendo al operario comprender todo intercambio de información con el robot. Su funcionamiento e imágenes con los diferentes modos de desplazamiento y operación se pueden encontrar en el capítulo de *Anexos*.

10.2.2. Operación de sensores y empaquetamiento de información

Como se explica en la sección anterior, todos los sensores menos la cámara pueden ser operados por el Arduino. A continuación se explica entonces el funcionamiento y operación de cada uno en el Arduino Leonardo utilizado para este proyecto:

- Sensor de temperatura *DS18B20*: La implementación de este sensor es muy sencilla, debido a que puede utilizar cualquier pin digital para operar. En este caso, se ha decidido utilizar el pin 4 para la configuración 1-Wire. Es necesario utilizar la librería *OneWire* de Paul Stoffregen, la cual permite utilizar un pin digital bajo este protocolo. Seguido de esto, se ha decidido utilizar la librería *DallasTemperature* de Miles Burton. Esta librería realiza los cálculos necesarios para convertir el número obtenido del sensor a un valor de temperatura en Celsius, evitando que el usuario tenga que realizar dichos cálculos. Además de las configuraciones de software mencionadas, es necesario introducir una resistencia de 4.7kΩ en pull-up que exige el protocolo 1-Wire para proteger el sensor. El voltaje y tierra se conectan a los 5V proporcionados por el módulo *LM2596S*.
- Sensor de gases peligrosos *MQ-135*: La implementación física de este sensor es extremadamente sencilla, donde solo se requiere conectar el pin A0 a un pin analógico del Arduino. En este caso, se ha decidido utilizar el pin A0. El cálculo de la curva característica de resistividad para CO2 es un poco compleja de interpretar, pero afortunadamente se ha implementado la librería *MQ135* de George Krockner, la cual realiza la mayoría de cálculos necesarios para la obtención de medidas de CO2 al evaluar la curva con los coeficientes y parámetros necesarios. Es así como se obtiene un valor en partes por millón de la concentración de CO2 en el ambiente. Es importante mencionar que este sensor toma veinte segundos en encender debido a que se debe calentar. Este se alimenta de 5V mediante el módulo *LM2596S*.
- Módulo GPS *GY-GPS6MV2*: La implementación en software es bastante sencilla, ya que la librería *TinyGPS* de Mikal Hart reduce los complejos comandos a instrucciones sencillas para obtener la latitud y longitud. Debido a que no se tiene más de un puerto UART en el Arduino, el control de este sensor se realiza con un puerto serial por software en los pines 8 (RX) y 7 (TX). Como se menciona antes, puede tomar incluso un par de minutos que el módulo se conecte a un satélite, por lo que al principio no se obtienen datos en la lectura de la latitud y longitud. Otro aspecto importante en la implementación física es que el módulo soporta hasta 3.6V. Es por esto que se está alimentando con los 3.3V que otorga el Arduino. Además, el puerto 7 (TX) del Arduino pasa por un seguidor de voltaje (para bajar la señal de 5V a 3.3V), con lo que evita quemar el puerto RX del módulo. El puerto TX del módulo se puede conectar directamente al pin 7 (RX) del Arduino ya que este si detecta un voltaje 3.6V como una señal alta, por lo que no es necesario escalar la señal a 5V.
- IMU *MPU6050*: La IMU resulta ser de los sensores más complicados de utilizar, ya que requieren de los cálculos matemáticos más demandantes para el Arduino y de una calibración previa. Sin embargo, las conexiones físicas son bastante sencillas, donde se comunica con el Arduino mediante I2C (con las librerías *I2CDev* y *Wire* que ya trae Arduino) y se alimenta con 5V del módulo *LM2596S*. Ahora, la programación necesaria es un poco robusta, la cual se apoya de la librería *MPU6050* de Jeff Rowberg. Para realizar la calibración, se coloca el sensor en su posición estándar en la que se utilizará y se modifica los offsets de manera constante para eliminar el error que se tiene tanto en el acelerómetro como en el giroscopio. Es recomendable utilizar algún filtro para estabilizar las lecturas de los mismos, donde al realizar la calibración la posición actual será la nueva posición horizontal. Una vez se ha realizado esto, se puede calcular los ángulos de inclinación con las lecturas del giroscopio y acelerómetro. En el programa

se ha además introducido un filtro para tratar de eliminar el error del Drift. La página *Naylamp Mechatronics* hace un buen trabajo en explicar este procedimiento [22]. Este se alimenta de 5V mediante el módulo *LM2596S*.

- Cámara *Raspberry Pi Camera V2 8MP*: La cámara se conecta al puerto CSI de la Raspberry Pi, por lo que desde el inicio se facilita la conexión y alimentación de la misma. Además, Python en la Raspberry Pi incluye la librería *picamera*, la cual permite tomar fotografías a la resolución deseada y en modo a color o en blanco y negro. Para poder obtener un arreglo de bytes de estas fotografías se utiliza la librería *base64*. Con esta librería, se puede convertir una imagen a información en bytes base16. El estándar base16 toma un byte de datos y lo convierte a dos caracteres en hexadecimal. Estos caracteres se pueden pasar a caracteres de un byte bajo la codificación *UTF-8* al agregarlos a un arreglo de caracteres, por lo que finalmente se obtiene un arreglo de bytes con toda la información de una fotografía que se puede enviar al operario.

Entendiéndose las conexiones y configuraciones necesarias para que cada sensor funcione correctamente, se presenta a continuación un esquemático que ilustra lo mencionado en el listado anterior. Esta es entonces la conexión final de los sensores del módulo electrónico.

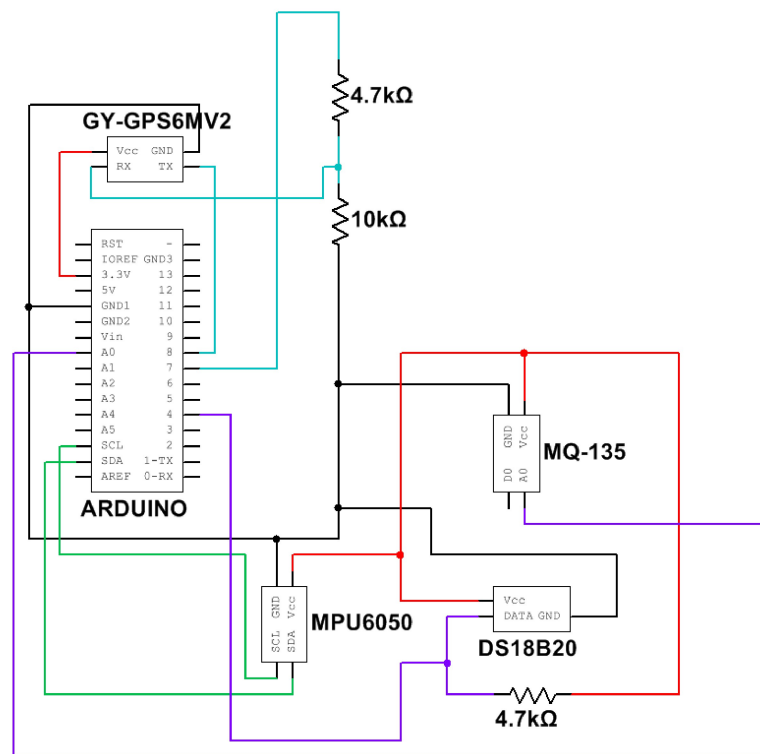


Figura 25: Esquemático de conexiones de sensores

El código tanto de la Raspberry Pi como de la computadora del operario se encuentra disponible en el repositorio detallado en la sección de *Anexos*.

El empaquetamiento de la información se realiza en el Arduino mediante la creación de un arreglo de 32 caracteres que se va llenando de la siguiente manera:

- Caracteres 1-5: Estos corresponden a la información del sensor de CO₂. Se han seleccionado 5 caracteres ya que los valores estándar de partes por millón (ppm) de CO₂ se encuentran en los cientos. Por ejemplo, los ppm actuales atmosféricos son 411. Esta selección permite en las centenas 1 valor decimal. (Ej. 130.5 = 5 caracteres)
- Caracteres 6-13: Estos corresponden a la información de la IMU. Se han seleccionado 8 caracteres para tener 4 en el ángulo roll y otros 4 en el ángulo pitch. Debido a que estos ángulos nunca pasan de 90 grados, esta selección permite en las decenas 1 valor decimal cuando es un valor positivo y 0 valores decimales cuando es un valor negativo. (Ej. 14.5 y 12.7 = 8 caracteres)
- Caracteres 14-17: Estos corresponden a la información del sensor de temperatura. Se han seleccionado 4 caracteres ya que la temperatura se tiende a mantener en las decenas. Esta selección permite 1 valor decimal cuando está en las decenas y 0 en las centenas. (Ej. 35.3 = 4 caracteres)
- Caracteres 18-32: Estos corresponden a la información del GPS, donde se han seleccionado 7 para la latitud y 8 para la longitud. La longitud tiene un carácter más de espacio ya que en Guatemala esta es negativa. Esta selección permite entonces 4 valores decimales para cada dato. (Ej. 14.5160 y -90.3256 = 15 caracteres)

Este arreglo se envía a la Raspberry Pi, donde se anexa al arreglo de datos de la cámara. Ya con el arreglo final, este se envía a la computadora del operario para que lo separe y se despliegue en cada una de las etiquetas de la interfaz. En caso algún sensor no logre realizar una lectura, el arreglo se llena con un indicador para interpretar que no se ha obtenido la lectura de dicho sensor. Estos indicadores son: *G* para datos del sensor de CO₂, *X* y *Y* para datos de la IMU, *T* para datos del sensor de temperatura y *L* para datos del GPS.

Prueba y validación de sistemas

Las siguientes pruebas se han realizado con la intención de demostrar el funcionamiento correcto de cada módulo, donde se busca cumplir de los criterios mencionados en la sección de *Alcance* del proyecto. Estos a su vez aseguran el cumplimiento del objetivo específico de cada módulo. El funcionamiento general sigue las conexiones y configuraciones detalladas en los capítulos anteriores de desarrollo de los distintos sistemas sin embargo la programación ha sufrido de cambios ligeros en varias de las pruebas. Esto con la intención de que se pueda observar fácilmente el funcionamiento, debido a que la mayoría de resultados se ejemplifican fácilmente mediante el uso de videos y no tanto por figuras y explicaciones narrativas. Además, es importante mencionar que todas las pruebas se realizaron con una laptop con sistema operativo *Windows 10*.

11.1. Prueba del sistema de comunicación

Estas pruebas demuestran que se ha realizado la implementación de un sistema que permite al operario recibir y enviar señales del robot a largas distancias, donde las pruebas se han realizado con un cable DB-9 de 12 metros. Esto debido a la poca disponibilidad, y por ende el alto costo de un cable de 100m. Sin embargo, el funcionamiento de hasta 1200m se encuentra presente gracias al estándar RS-485 discutido en los capítulos de comunicación. La velocidad máxima que permiten los integrados es de 1 Mbps.

Las pruebas digitales, en donde se cambio la programación para imprimir un mensaje de envío y recibo de datos, demuestran que los sistemas se encuentran efectivamente interconectados. Se realizaron 10 pruebas de comunicación durante 100 segundos cada una, donde se manejó el envío de información entre sistemas a una velocidad de 78.86 Kbps debido a que es la máxima permitida por los puertos seriales por software. En cada caso ocurrió una transferencia del 100 % de los datos exitosamente. En la Figura 26 se aprecia la transferencia

de información en la computadora del operario, la Raspberry Pi y el Arduino. Para demostrar este funcionamiento se agregaron textos de recibo y envío de datos. La Raspberry Pi es la pantalla en rosado mientras que el Arduino y la computadora se aprecian en el lado izquierdo.

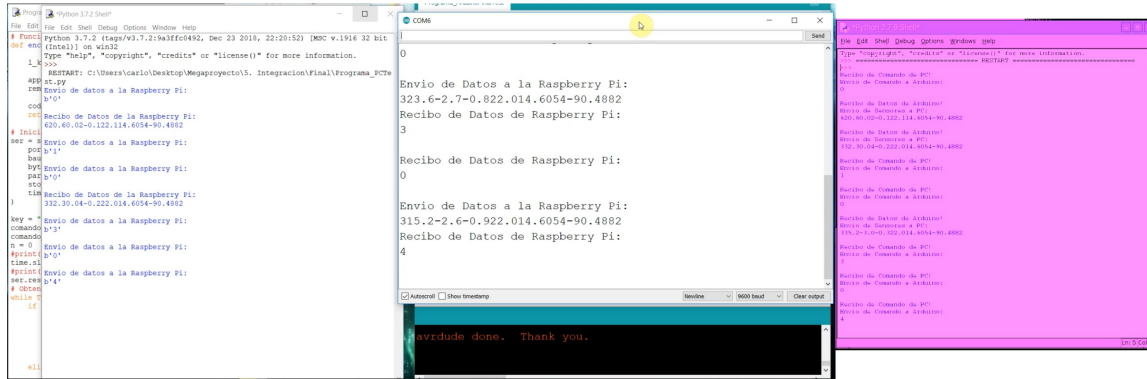


Figura 26: Transferencia de datos Arduino<->RPI<->PC

Además, se puede apreciar que la computadora del operario se encuentra recibiendo información de los sensores en la Figura 27, lo cual ayuda a demostrar que se ha realizado una comunicación exitosa con los sensores de la Raspberry Pi y Arduino.

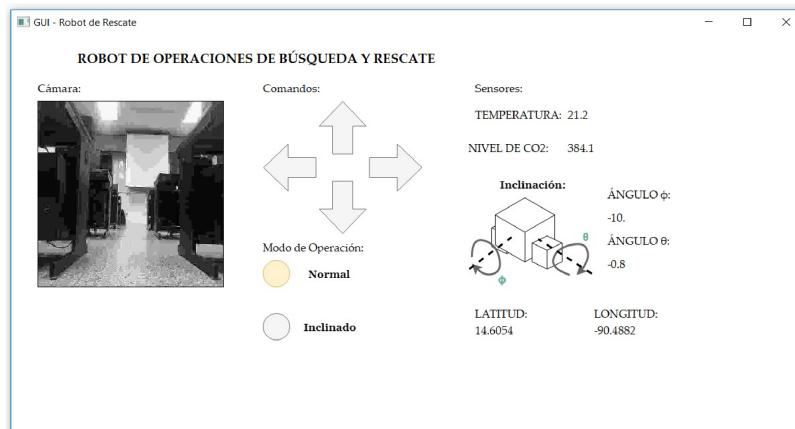


Figura 27: Recibo de datos en la interfaz gráfica

Es así como se demuestra que se encuentra funcional el sistema de comunicación, donde se realiza exitosamente el envío de comandos de la computadora del operario hasta el Arduino y el recibo de datos de los sensores por el operario.

11.2. Prueba del sistema de comandos

Con el sistema de comunicación funcionando correctamente, las pruebas del sistema de comandos resultan sencillas de observar. Se ha agregado el video de funcionalidad en la referencia [23]. En esta se puede observar como responden los servomotores *AX-12A* y *D950TW* a las instrucciones del operario con el teclado. Además, en la Figura 28 se

observa que la Raspberry Pi está recibiendo correctamente los comandos según la codificación mencionada en el capítulo 9.

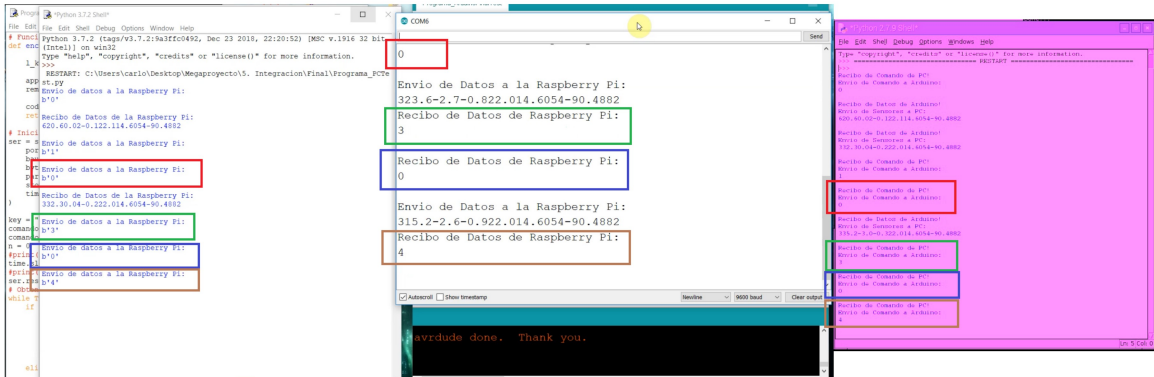


Figura 28: Recibo de comandos por el Arduino

11.3. Prueba del sistema de medición

Como se observa en la Figura 27, la interfaz gráfica se encuentra recibiendo información de los sensores. Adicional a su operación, se realiza dos pruebas para garantizar que estos ofrezcan un funcionamiento adecuado para el rescate de personas. Se ha escogido que la lectura de los sensores sea cada 5 segundos para darle prioridad al envío de comandos.

En la Figura 29 se muestra el comportamiento del sensor de CO₂ al someterse a una fuente constante de CO₂ por 100 segundos. Esta fuente fue la respiración de una persona cercana al sensor en un cuarto ventilado. Como referencia, el nivel atmosférico de CO₂ es de 411 partes por millón (ppm).

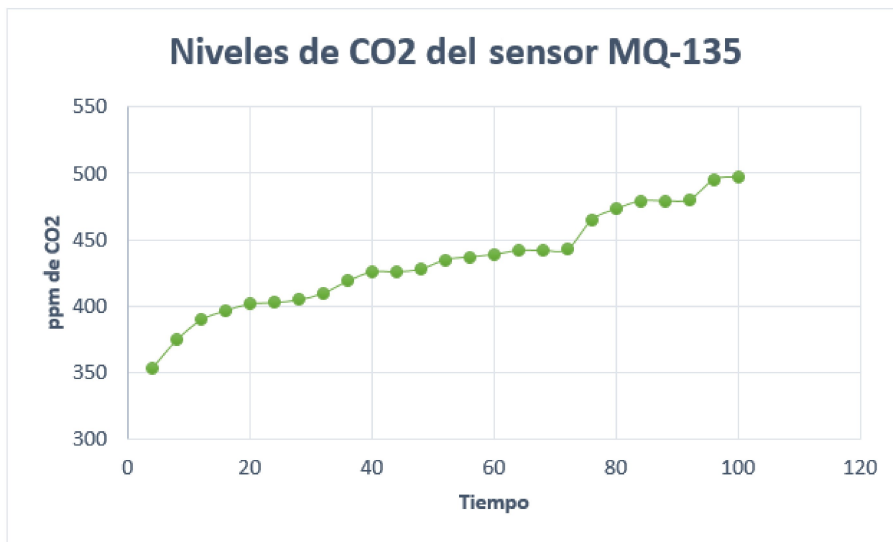


Figura 29: Prueba de medición de niveles de CO₂ (ppm)

Se puede apreciar entonces que el sensor si detecta la presencia del gas y puede ayudar a medir ambientes con niveles altos. Esto es importante ya que puede permitir entonces la detección de posibles personas en socorro. Seguido a esto se realiza una prueba con el sensor de temperatura, donde se agarra de la varilla de metal con la mano para observar si existe un incremento de temperatura.

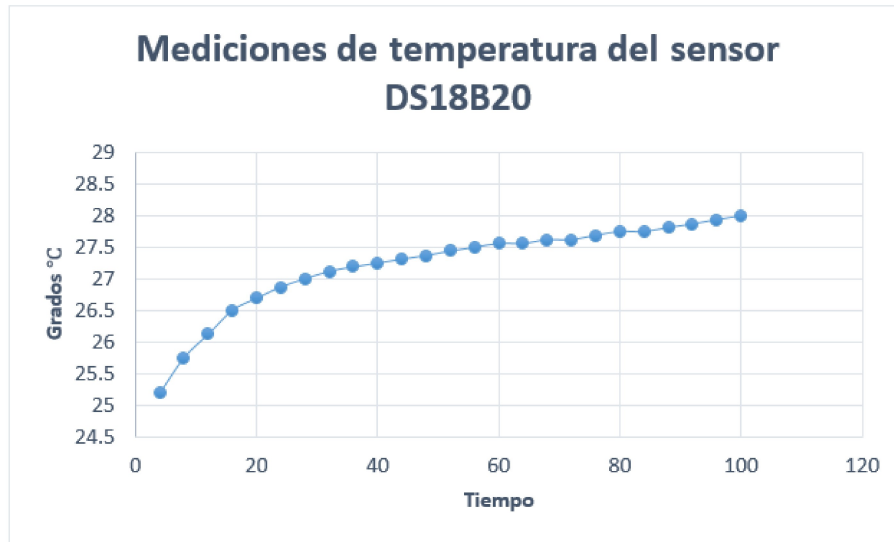


Figura 30: Prueba de medición de temperatura

Como se puede observar, la temperatura empieza a subir rápidamente a niveles corporales (30 °C). Es así como se comprueba entonces que el sensor de encuentra operacional.

Por parte de la IMU y el módulo GPS, el funcionamiento es estándar y no se han presentado datos más relevantes. Para observar diferentes medidas y el envío de sus datos se puede consultar la Figura 26 donde se tiene la información empaquetada en un arreglo de caracteres. Con las pruebas realizadas en la universidad, la latitud y longitud se mantienen alrededor de 14.6054 y -90.4882.

12.1. Implementación del módulo electrónico

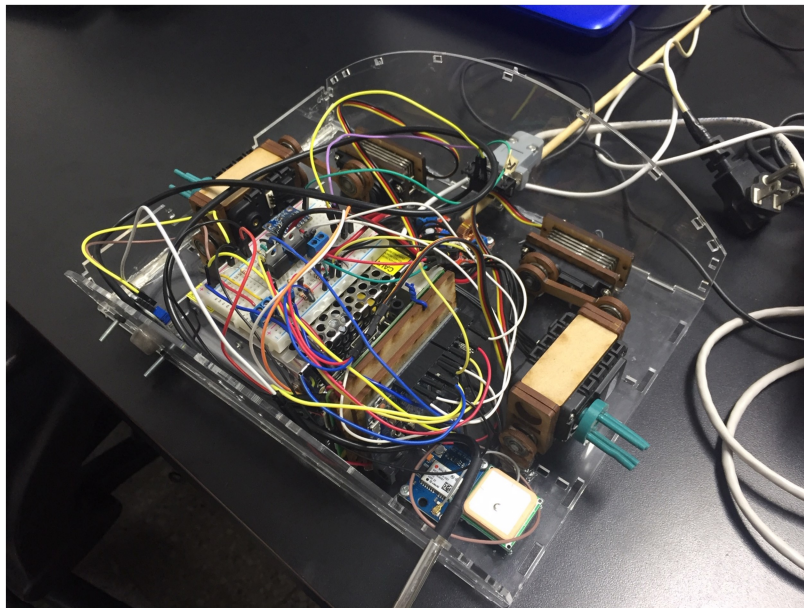


Figura 31: Módulo electrónico del robot

El módulo electrónico permite la comunicación del operario con el robot, donde realiza el envío de comandos que el operario desea que el robot ejecute y el recibo de información de los sensores del robot para que sean interpretados con la intención de dar soporte en las operaciones de búsqueda y rescate. Para poder utilizar el sistema se debe realizar lo siguiente:

- Conexión de la computadora con el robot: Para conectar la computadora del operario y el robot, se debe conectar el sistema externo a la computadora con un puerto serial o un puerto USB. Este sistema se conecta al cable DB-9, el cual se conecta en el otro extremo al sistema interno del robot. Este puerto se observa en la parte posterior del prototipo de la Figura 31.
- Configuración de la comunicación serial: Para conectarse correctamente con el robot, en las configuraciones de la comunicación serial de la computadora se debe seleccionar el puerto correcto que se está utilizando. (Ej. port='COM16') Si se desconoce el puerto, en la ventana de *Device Manager* se pueden observar los nombres de los puertos en uso.

12.2. Prototipo final desarrollado

El desarrollo del sistema interno y externo se ha desarrollado en placas de prueba, las cuales permiten realizar pruebas y modificar conexiones sin mayores problemas. A continuación se presenta la placa interna y la externa:

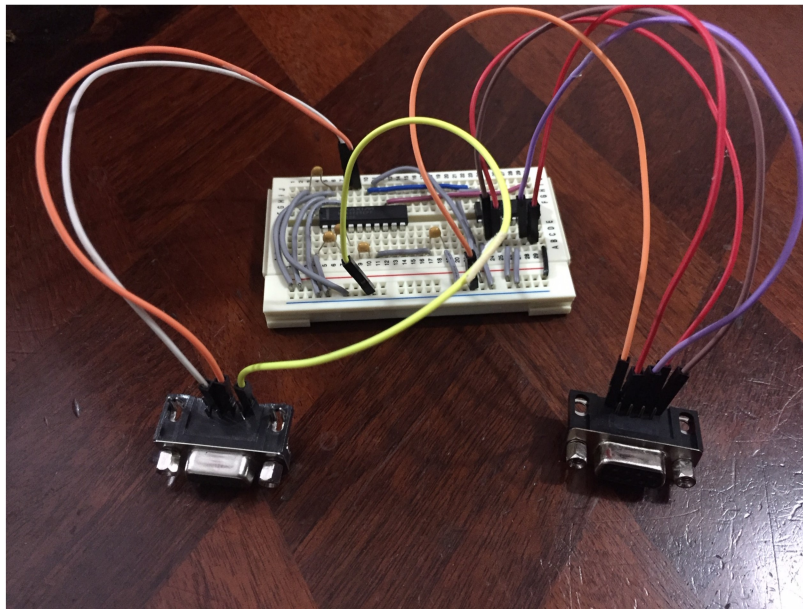


Figura 32: Placa de pruebas externa para RS-232

La placa externa se conecta a el puerto serial de la computadora (o a un puerto USB a TTL), mientras que el otro puerto DB-9 es para conectar este sistema con el cable DB-9 largo y realizar la comunicación con el sistema interno del robot.

Si bien las conexiones del sistema interno parecen bastante complejas, estas son las mismas que las ilustradas en la Figura 12.

A continuación se muestra el prototipo final desarrollado en conjunto al módulo mecánico, encargado de los mecanismos y chasis mostrados, con todo el funcionamiento establecido para cumplir con el objetivo principal del módulo electrónico:

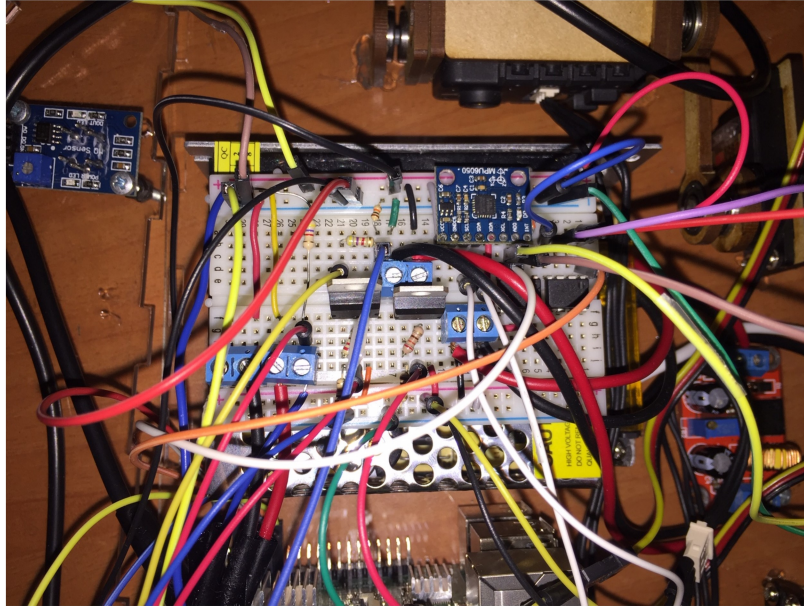


Figura 33: Placa de pruebas interna

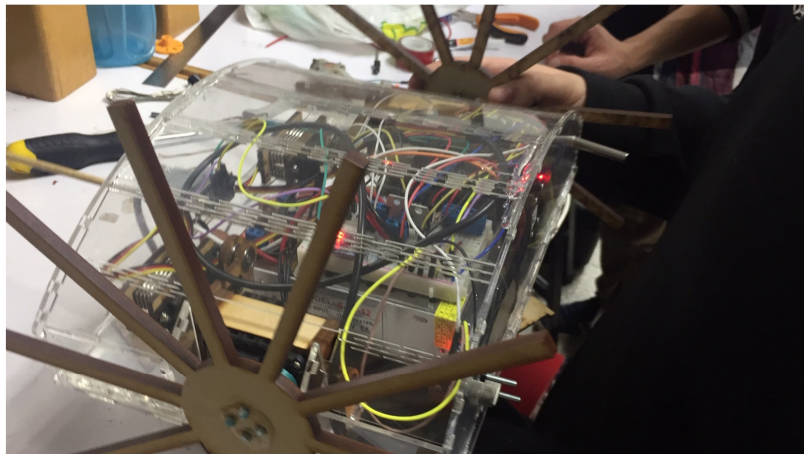


Figura 34: Prototipo final ensamblado

- Se realizó la implementación de un sistema de comunicación de sistemas internos y externos del robot, el cual transmite información el 100 % del tiempo a una velocidad máxima de 78.86 Kbps a una distancia máxima de 12 metros para el envío de comandos y recibo de información de sensores.
- El diseño del sistema de comunicación permite la interacción del robot con el operario a una distancia máxima de 1200 metros según los criterios del estándar RS-485 con el integrado MAX485 a una velocidad máxima de 78.86 Kbps con el uso de *Bit-Banging* o de 92.16 Kbps con el uso de puertos UART.
- Se ha realizado la implementación de un sistema de comandos que envía señales (bytes) de la computadora del operario a los servomotores en el Arduino para su movilización bajo seis modos de desplazamiento y dos modos de operación a una velocidad de 78.86 Kbps.
- Se ha realizado la implementación de un sistema de medición que utiliza una cámara y cuatro tipos de sensores para obtener las cifras de la latitud, longitud, ángulo roll, ángulo pitch, temperatura y niveles de CO2 en el entorno del robot cada aproximadamente 5 segundos de operación.

- Se recomienda realizar pruebas con cables de mayor longitud a 12m para poder garantizar que el estándar RS-485 se encuentra operando tanto en la teoría como en la práctica.
- Se recomienda utilizar un sensor de CO2 más robusto ya que el utilizado mide la concentración de más de un tipo de gas, por lo que puede haber problemas en su lectura si se encuentran muchos de estos combinados en el entorno del robot.
- Se recomienda contar con un cable de alimentación de 14 AWG más largo de 10m para poder realizar las pruebas con cables más largos DB-9, y a su vez se recomienda sostener este junto al DB-9 para evitar que se enrolle en el recorrido del robot.
- Se recomienda cambiar la cámara utilizada por una infrarroja y acompañarla de un sensor térmico, tal como se detalla en el capítulo 1 en la selección de sensores.
- Se recomienda integrar los módulos de manera que se puedan realizar pruebas a la intemperie para poder observar el desempeño del robot en terrenos irregulares.

-
-
- [1] K. C. Galloway, G. C. Haynes, B. D. Ilhan, A. M. Johnson, R. Knopf, G. A. Lynch, B. N. Plotnick, M. White y D. E. Koditschek, “X-RHex: A highly mobile hexapedal robot for sensorimotor tasks”, 2010.
 - [2] M. Perez, *Origami-inspired Robot Can Hitch a Ride with a Rover*, mar. de 2017. dirección: <https://www.nasa.gov/feature/jpl/origami-inspired-robot-can-hitch-a-ride-with-a-rover>.
 - [3] *Disaster Response Robot "Quince"*. dirección: https://www.rm.is.tohoku.ac.jp/quince_eng/.
 - [4] *Guatemala*. dirección: <https://www.gfdrr.org/en/guatemala>.
 - [5] A. Chamy, *Galería: Así trabajan los equipos de rescate tras la erupción del volcán de Fuego en Guatemala*, jun. de 2018. dirección: <https://www.emol.com/noticias/Internacional/2018/06/05/908693/Retoman-busqueda-victimas-erupcion-del-volcan-de-Fuego-en-Guatemala.html>.
 - [6] Y. Bai, *The Windows Serial Port Programming Handbook*. CRC Press, 2005.
 - [7] J. Lindblom, *Serial Communication*. dirección: <https://learn.sparkfun.com/tutorials/serial-communication>.
 - [8] J. Patrick, *Serial Protocols Compared*, mayo de 2002. dirección: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.477.821&rep=rep1&type=pdf>.
 - [9] Y.-y. Fang y X.-j. Chen, “Design and simulation of UART serial communication module based on VHDL”, en *2011 3rd International Workshop on Intelligent Systems and Applications*, IEEE, 2011, págs. 1-4.
 - [10] S. A. Arduino, “Arduino”, *Arduino LLC*, 2015.
 - [11] M. Richardson y S. Wallace, *Getting started with raspberry PI*. O'Reilly Media, Inc.", 2012.
 - [12] S. Bonino, *Carbon Dioxide Detection and Indoor Air Quality Control*, abr. de 2016. dirección: <https://ohsonline.com/articles/2016/04/01/carbon-dioxide-detection-and-indoor-air-quality-control.aspx>.

- [13] *What's The Difference between Thermal Imaging and Night Vision?* Dirección: <https://www.flir.com/discover/ots/thermal-vs-night-vision/>.
- [14] *PiNoIR – what's it for? Comparison of RasPiCam and Pi NoIR output in daylight*, oct. de 2013. dirección: <https://raspi.tv/2013/pinoir-whats-it-for-comparison-of-raspicam-and-pi-noir-output-in-daylight>.
- [15] *RS-232 Transceivers*, feb. de 2011. dirección: <https://datasheets.maximintegrated.com/en/ds/MAX3224-MAX3245.pdf>.
- [16] *Command line vs. GUI*, nov. de 2018. dirección: <https://www.computerhope.com/issues/ch000619.htm>.
- [17] *Programmable Resolution 1-Wire Digital Thermometer*, jul. de 2019. dirección: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>.
- [18] *MQ-135 Gas Sensor*, ene. de 2014. dirección: <https://www.olimex.com/Products/Components/Sensors/Gas/SNS-MQ135/resources/SNS-MQ135.pdf>.
- [19] *NEO-6 GPS Modules*, dic. de 2011. dirección: <https://www.openimpulse.com/blog/wp-content/uploads/wpsc/downloadables/GY-NEO6MV2-GPS-Module-Datasheet.pdf>.
- [20] *MPU-6050 Product Specification*, ago. de 2013. dirección: https://components101.com/sites/default/files/component_datasheet/MPU6050-DataSheet.pdf.
- [21] *Camera Module V2*, abr. de 2016. dirección: <https://www.raspberrypi.org/products/camera-module-v2/>.
- [22] *Tutorial MPU6050 - Acelerómetro y Giroscopio*, jul. de 2016. dirección: https://naylampmechatronics.com/blog/45_Tutorial-MPU6050-Aceler%C3%B3metro-y-Giroscopio.html.
- [23] *Prueba de Funcionamiento de Actuadores*, sep. de 2019. dirección: https://drive.google.com/file/d/1-pm4_-Zwu9JzzBN81PkBY0sdrQ2vaLmA/view?usp=sharing.
- [24] *Repositorio con Programación del Módulo Electrónico*, sep. de 2019. dirección: <https://github.com/san15549/SearchProbe>.

16.1. Repositorio con la programación del módulo electrónico

El repositorio en Github, el cual se encuentra en la referencia [24], contiene la programación de todos los sistemas (PC, Raspberry Pi y Arduino) para que se pueda consultar y utilizar libremente.

16.2. Despliegue de comandos en la interfaz gráfica

A continuación se presentan los diferentes despliegues de información de la interfaz gráfica:

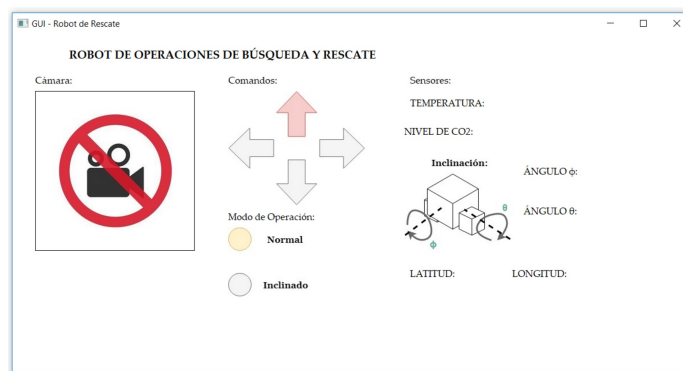


Figura 35: Interfaz gráfica con indicador de arriba

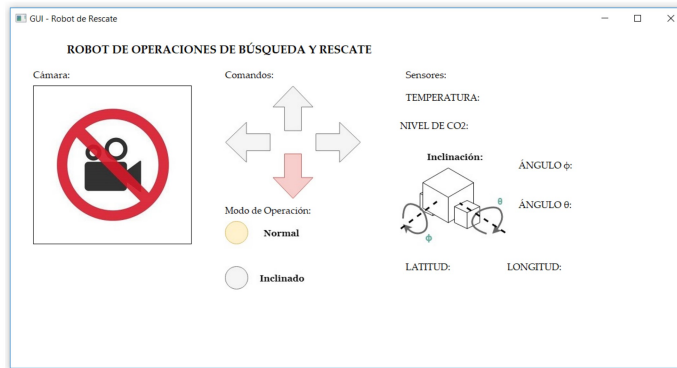


Figura 36: Interfaz gráfica con indicador de abajo

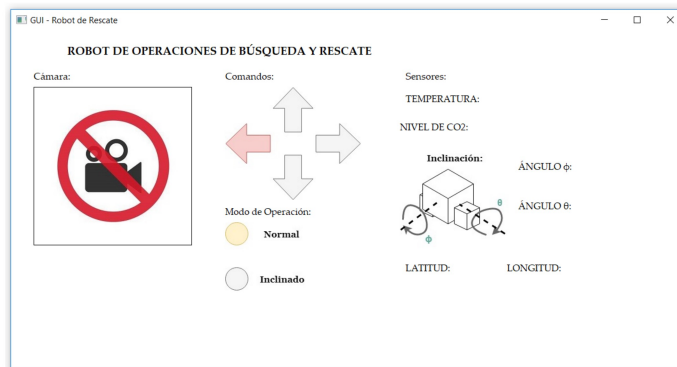


Figura 37: Interfaz gráfica con indicador de izquierda

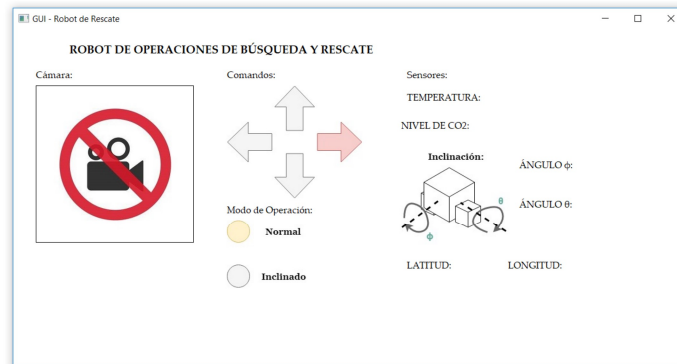


Figura 38: Interfaz gráfica con indicador de derecha

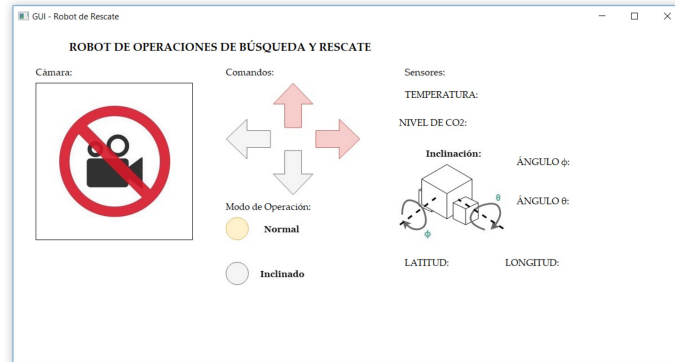


Figura 39: Interfaz gráfica con indicador de arriba-derecha

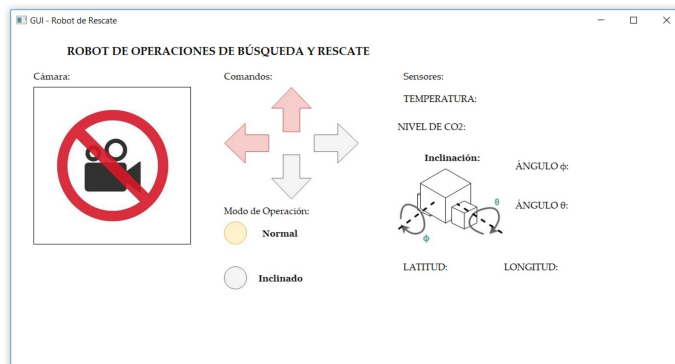


Figura 40: Interfaz gráfica con indicador de arriba-izquierda

16.3. Avances de la segunda fase

La segunda fase del proyecto introdujo ligeros cambios a los sensores utilizados y al ensamble del prototipo. Se ha reemplazado el sensor de CO₂ por el sensor *Grove CO₂ Sensor*, el cuál es más robusto y otorga resultados más exactos. De la misma manera, se ha introducido la cámara infrarroja *PiNoIR* acompañada del sensor térmico *GRID-EYE* para reemplazar la cámara convencional de la primera fase. Estas mejoras introdujeron como consecuencia cambios a la interfaz gráfica y a la ubicación de componentes dentro del prototipo. Finalmente, se han introducido dos servomotores *Dynamixel AX-12A* adicionales para mejorar la tracción y estabilidad. Todos estos cambios se pueden apreciar en las siguientes imágenes:

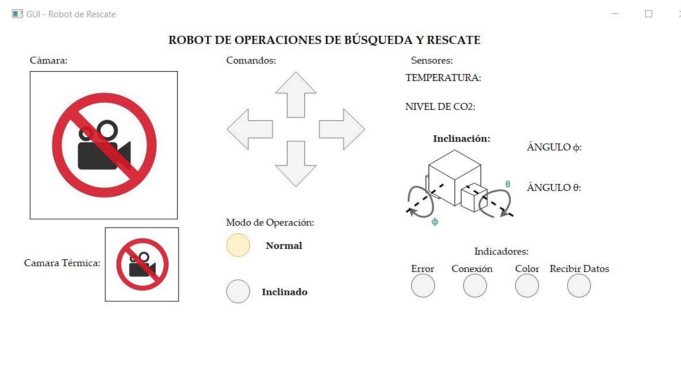


Figura 41: Interfaz gráfica - segunda fase

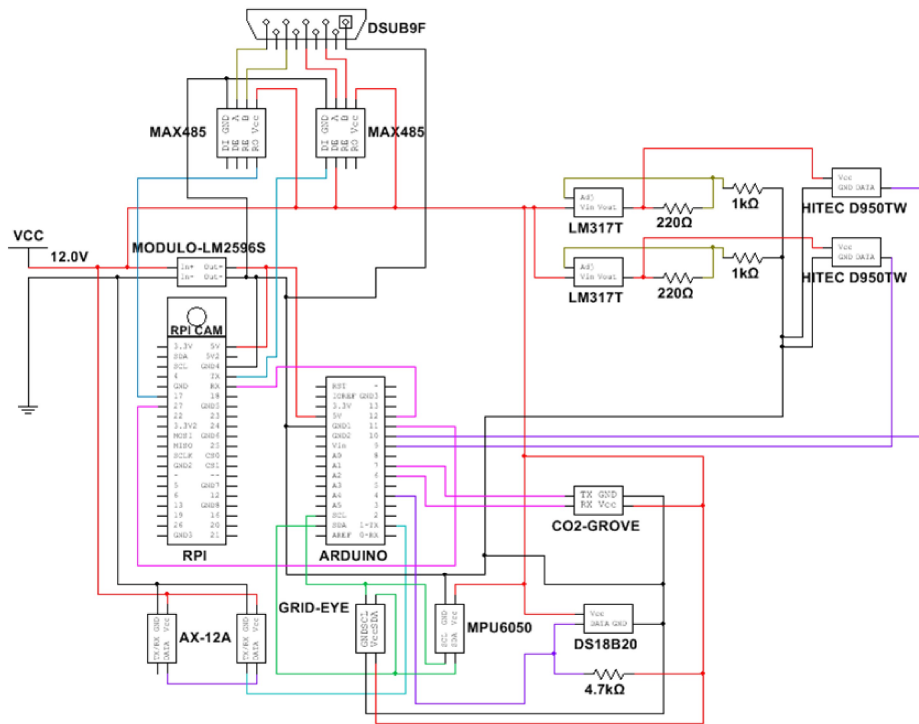


Figura 42: Reestructuración de conexiones internas - segunda fase

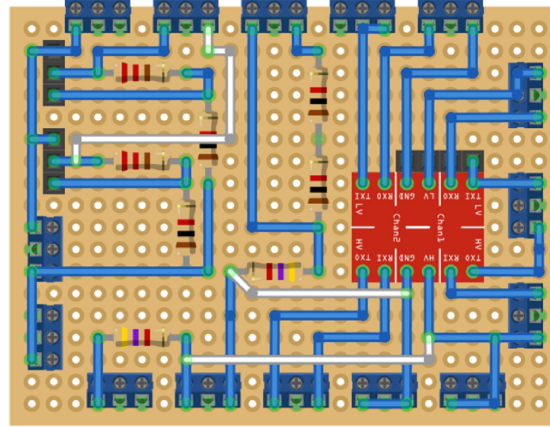


Figura 43: Diseño de placa para conexiones internas - segunda fase

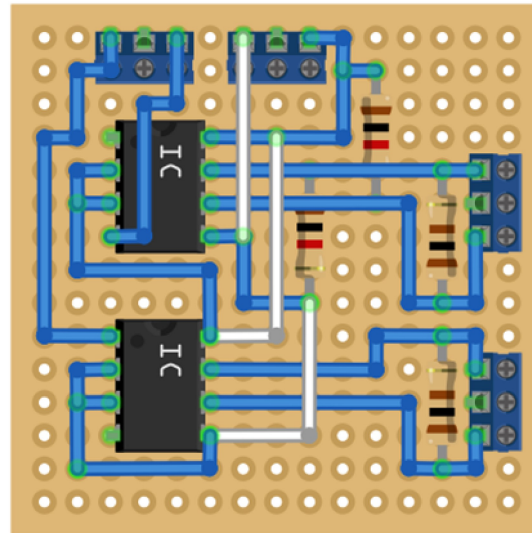


Figura 44: Diseño de placa para conexiones externas - segunda fase

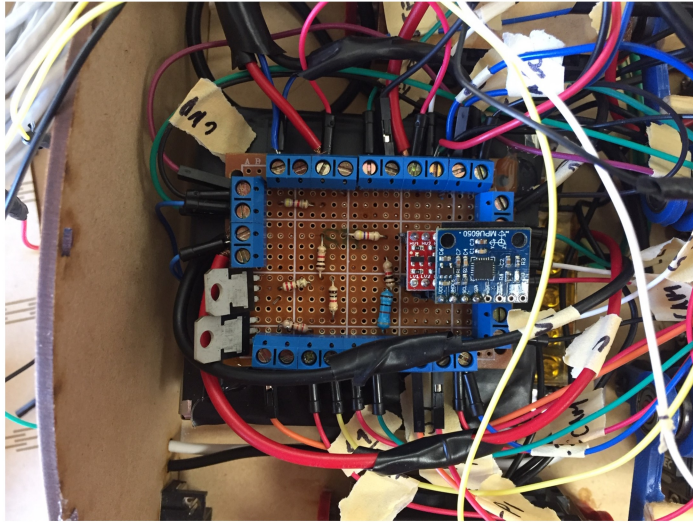


Figura 45: Placa de conexiones internas - segunda fase

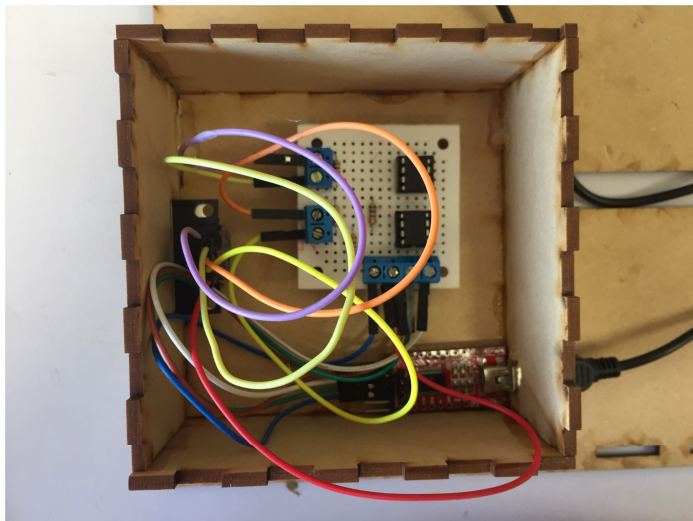


Figura 46: Placa de conexiones externas - segunda fase

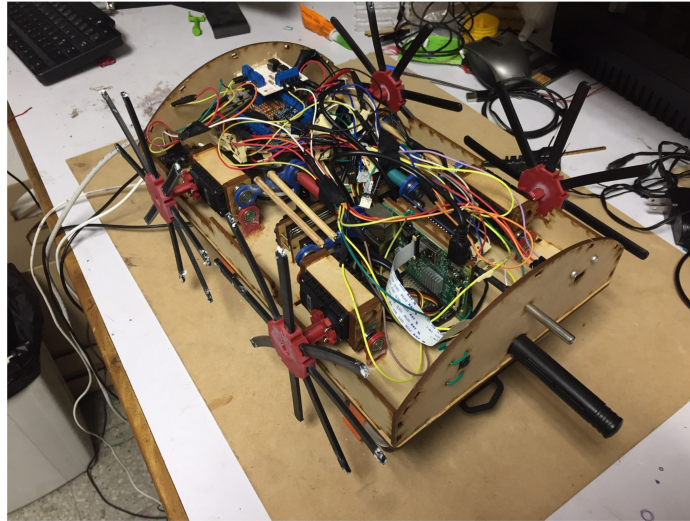


Figura 47: Conexiones del prototipo final - segunda fase

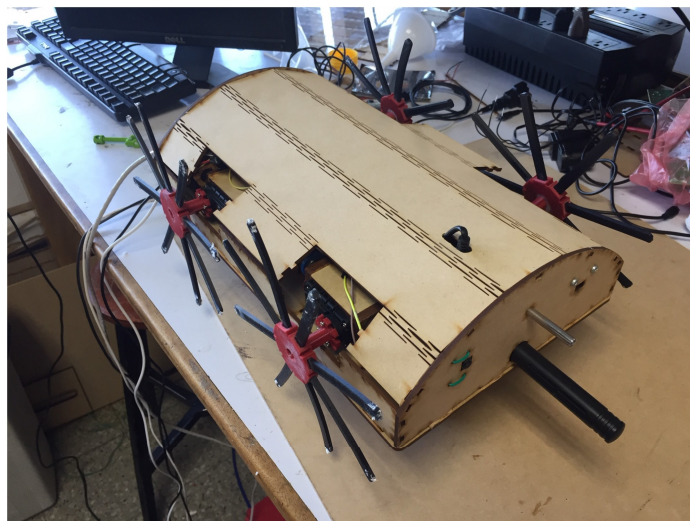


Figura 48: Prototipo final ensamblado- segunda fase