



UNIVERSIDAD ANDRÉS BELLO
FACULTAD DE INGENIERÍA
AUTOMATIZACIÓN Y ROBÓTICA

“Rover en Búsqueda y Rescate”

Alumnos:

Matías Ignacio Venegas Fuenzalida
Alfredo Andrés Gálvez Pradenas
Ignacia Belén Salvatierra Moreno
Bastián Ignacio Figueroa Parada
Roberto Andrés Godoy Villagran

Profesor:

Palominos Néstor
Santiago de Chile, 2022

índice General

| | |
|---------------------------------------|-----------|
| Resumen | 4 |
| Abstract | 6 |
| Introducción | 7 |
| Objetivos | 8 |
| Objetivo General | 8 |
| Objetivos específicos | 8 |
| Marco Teórico | 9 |
| Servomotor | 10 |
| PWM | 12 |
| aplicaciones del PWM: | 14 |
| Instrumentos | 14 |
| Arduino | 14 |
| Raspberry | 15 |
| Cámara | 16 |
| Módulo bluetooth | 16 |
| Sensor de ultrasonido | 17 |
| Sensor fotosensible Lm393 | 18 |
| Sensor de Humedad y Temperatura DHT11 | 19 |
| Estado del arte | 21 |
| Vehículos no tripulados | 21 |
| NASA | 24 |
| Carta Gantt | 26 |
| Desarrollo | 27 |
| Costos y presupuestos | 35 |
| Resultados | 36 |
| Discusiones | 37 |
| Conclusiones | 38 |
| Bibliografía | 39 |
| Anexos | 40 |
| Nombre del Código control rover.ino | 40 |

Indice Ilustraciones

| | |
|---|----|
| ECUACIÓN 1 FÓRMULA PWM | 11 |
| | |
| ILUSTRACIÓN 1 ESKUEMA ANÁLOGO-DIGITAL | 8 |
| ILUSTRACIÓN 2 CONEXIÓN SERVOMOTOR A PLACA ARDUINO | 8 |
| ILUSTRACIÓN 3 SIGNIFICADO DE LAS CONEXIONES | 9 |
| ILUSTRACIÓN 4 ESTRUCTURA INTERNA SERVOMOTOR | 9 |
| ILUSTRACIÓN 5 REFERENCIA TREN DE PULSO | 10 |
| ILUSTRACIÓN 6 ESPECIFICACIÓN DE CADA PARTE | 12 |
| ILUSTRACIÓN 7 CLASIFICACIÓN DE LOS PWM | 13 |
| ILUSTRACIÓN 8 ARDUINO | 13 |
| ILUSTRACIÓN 9 RASPBERRY PI | 14 |
| ILUSTRACIÓN 10 CÁMARA | 14 |
| ILUSTRACIÓN 11 MÓDULO BLUETOOTH | 14 |
| ILUSTRACIÓN 12 SENSOR ULTRASONIDO | 15 |
| ILUSTRACIÓN 13 PULSOS | 16 |
| ILUSTRACIÓN 14 RANGO DE DETECCIÓN | 16 |
| ILUSTRACIÓN 15 SENSOR FOTOSENSIBLE | 16 |
| ILUSTRACIÓN 16 SENSOR DHT11 | 17 |
| ILUSTRACIÓN 17 ESPECIFICACIONES | 18 |
| ILUSTRACIÓN 18 URBOT | 20 |
| ILUSTRACIÓN 19 MATILDA, MMPS MATILDA (II) | 20 |
| ILUSTRACIÓN 20 OTRAS PLATAFORMAS | 21 |
| ILUSTRACIÓN 21 ROVER NASA | 21 |
| ILUSTRACIÓN 22 CARTA GANTT | 23 |
| ILUSTRACIÓN 23 CONEXIONES 1 | 25 |
| ILUSTRACIÓN 24 CONEXIONES 2 | 25 |
| ILUSTRACIÓN 25 DIAGRAMA DE BLOQUES | 26 |
| ILUSTRACIÓN 26 DIAGRAMA DE FLUJOS | 27 |
| ILUSTRACIÓN 27 ROVER PROYECTO V1 | 29 |
| | |
| TABLA 1 COSTOS Y PRESUPUESTOS | 28 |

Resumen

Hoy en día nuestros brigadistas de búsqueda y rescate ponen en peligro sus vidas al rescatar personas de los escombros de los edificios, aunque cuentan con la tecnología y equipamiento para enfrentarse a estas situaciones, siempre existen sucesos inesperados como derrumbes, fuego, entre otros.

Para ello diseñamos un vehículo no tripulado para que pueda atender este tipo de emergencias procurando que nuestros brigadistas puedan observar y tomar las mejores decisiones ante un cambio drástico del escenario de la emergencia.

Este prototipo es controlado a través de conexión remota proporcionada por la web, puede aventurarse en espacios pequeños o de arquitecturas inestables para enseñar los niveles de temperatura, ultrasonido, humedad y una imagen en vivo de lo que ocurre dentro del siniestro. Para su desarrollo contempla la tecnología Raspberry PI 3 B en conjunto con un software libre adicionando con una página web que nos permita conocer más de este prototipo y observar el streaming de este vehículo.

Palabras Claves: Vehículo- RaspberryPI 3B-Streaming- brigadistas-no tripulado

Abstract

Nowadays, our search and rescue brigades endanger their lives by rescuing people from the rubble of buildings, although they have the technology and equipment to deal with these situations, there are always unexpected events like collapses, flames, toxics, among others.

To do this, we designed an unmanned vehicle capable of responding to this emergency and supporting this team so that they can observe the situation by taking a stand.

We designed an unmanned vehicle so that it can attend to this type of emergency, ensuring that our brigade members can observe and make the best decisions in the face of a drastic change in the emergency stage.

This prototype is controlled through a remote connection provided by the web, and can venture into small spaces or unstable architectures to show the temperature levels, ultrasound, humidity and a live image of what happens inside the crash. For its development, Raspberry PI 3 B technology will be considered in conjunction with free software by adding a website that allows us to learn more about this prototype and watch the streaming of this vehicle.

Keywords: Vehicle- RaspberryPI 3B-Streaming- rescue team-unmanned

Introducción

Hoy en día existe una gran cantidad de “Rover” en el área de la astronomía y exploración de planetas no habitables, capaces de recolectar información de forma remota, visualización y envío de señales en vivo, tales como: presión, energía solar, imagen en vivo, etc.

Un Rover es “conocido como vehículo de exploración espacial diseñado para moverse sobre la superficie de planetas u otros objetos astronómicos.”

La Nasa ha implementado distintos modelos para la investigación del espacio, logrando grandes avances en el área Aeroespacial,

Existe un rover para la desactivación de explosivos llamado Aunav.NEO, el cual es capaz de adaptarse en diferentes entornos, es decir, “adaptarse a cualquier escenario operativo imaginable, incluidos pasillos estrechos de aviones, autobuses o subterráneos y túneles subterráneos para vadear escombros o atravesar espacios abiertos. Un robot para todos”.(AUNAV-NEO, 2022)

¿Cómo podemos combinar esto y con qué fin?

El área de Búsqueda y Rescate es mundialmente conocido por trabajar en lugares de difícil acceso y/o en lugares de poca visibilidad, si bien existen distintas especialidades nos centraremos en el “USAR o INSARAG”, ya que este equipo se especializa en Búsqueda y Rescate en escombros, la gran parte de su trabajo es guiarse por sus instintos y elaborar un plan con la ayuda de un topógrafo quien es el que evalúa el terreno.

Generalmente ocurren imprevistos que el equipo no puede detectar por ejemplo el bloqueo de una vía de acceso, emanaciones de calor, persona atrapada, etc. y es justo el momento donde entra este artefacto el cual a través de un manejo y visualización remota podrá guiar a los expertos a una zona, la cual, permita el desarrollo del plan de trabajo y la toma decisiones de acuerdo a los datos recolectados.

El vehículo a realizar mantendrá algunas características de los ROVERS mencionados anteriormente, enfocado en el área de Búsquedas y Rescates, el cual nos permite observar a través de transmisión en vivo las dimensiones de la catástrofe y señales de los diferentes sensores que tendrá incorporado.

Con este dispositivo se buscará la recopilación de datos del entorno, ya sea con sensores, video en tiempo real y <desplazamiento del vehículo, lo cual permitirá la implementación de un plan de trabajo de forma rápida y precisa, anticipando algún escenario riesgoso, disminuyendo la posibilidad de algún accidente laboral y/o dañar a la víctima.

¿Por qué el área de búsqueda y rescate?

Mundialmente conocido por trabajar en lugares de difícil acceso y/o en lugares de poca visibilidad, si bien existen distintas especialidades nos centraremos en el “USAR o INSARAG”, ya que este equipo se especializa en Búsqueda y Rescate en escombros, la gran parte de su trabajo es guiarse por sus instintos y elaborar un plan con la ayuda de un topógrafo quien es el que evalúa el terreno. “Compuesta por un grupo de personas organizadas y capacitadas para emergencias, mismos que serán responsables de combatirlas de manera preventiva o ante eventualidades de un alto riesgo, emergencia, siniestro o desastre, dentro de una empresa, industria o establecimiento y cuya función está orientada a salvaguardar a las personas, sus bienes y el entorno de los mismos” (UNAM)

Objetivos

Objetivo General

Desarrollar e implementar un “Rover” para ayudar a los Brigadistas a desarrollar una forma más efectiva de su plan de trabajo, para así explorar zonas que las personas no pueden atravesar, sacando imágenes con su cámara integrada para detectar personas en peligro.

Objetivos específicos

- Instalar sistema de cámara con Raspberry que sea capaz de realizar streaming en vivo.
- Construir un chasis de auto Smart 4x4 con el material metálico
- Implementar sensores para temperatura, humedad, ultrasonido y de flama
- MQTT

Marco Teórico

Conversor análogo- digital

Una señal eléctrica analógica es una señal cuyo voltaje o valor de voltaje cambia constantemente y puede tomar cualquier valor. En el caso de la corriente alterna, la señal analógica aumenta su valor con un signo positivo (+) en un medio ciclo y luego disminuye con un signo negativo (-) en el siguiente medio ciclo.

Los sistemas de control (como los microcontroladores) no pueden utilizar señales analógicas, por lo que deben convertirse en señales digitales antes de poder utilizarlas.

La señal digital obtenida de la señal analógica tiene dos propiedades básicas: Valores. El valor de voltios define 0 y 1. En nuestro caso, esta es la tecnología TTL (0-5V) Resolución analógica: utilizamos símbolos digitales para indicar el número de bits de la señal analógica.

Para el Arduino Uno, el valor analógico de 0 voltios se representa digitalmente como B0000000000 (0), y el valor analógico de 5V se representa digitalmente como B1111111111 (1023). Por lo tanto, cada valor analógico intermedio está representado por un valor entre 0 y 1023, es decir, sumó 1 al binario por cada 4.883 mV.

La placa Arduino Uno, su resolución es de 10 bits, que es un valor entre 0 y 1023.

Sistema digital

Un sistema digital es cualquier sistema que permite la creación, modificación, transmisión o almacenamiento de información expresada en una cantidad limitada de modo que sus señales de entrada y salida solo permite valores discretos.

Los valores discretos son variables que no aceptan ningún valor, sino que solo aceptan variables que pertenecen a su conjunto, por lo que son limitadas.

En este sentido, un sistema digital es cualquier dispositivo que manipula datos a través de números que casi siempre están representados por códigos binarios. El sistema binario sólo acepta cero (0) y uno (1) como valores, por lo que son valores discretos.

Actualmente, los sistemas digitales están integrados en dispositivos magnéticos, electrónicos y mecánicos.

Sistema analógico

Un sistema analógico es un sistema cuya señal puede permitir cambios continuos en valores infinitos. Los datos naturales provienen de fuentes analógicas: temperatura, distancia,

sonido, voltaje, imagen, etc. Aunque todas estas variables se pueden convertir a datos digitales, originalmente eran analógicas.

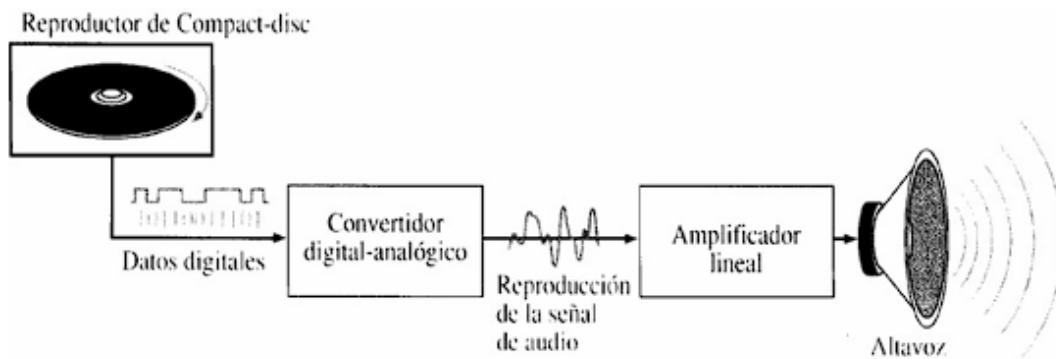


Ilustración 1 Esquema análogo-digital

Servomotor

El servo se controla mediante un código previamente cargado en la placa Arduino, el cual le envía impulsos eléctricos el cual hace que este motor gire.

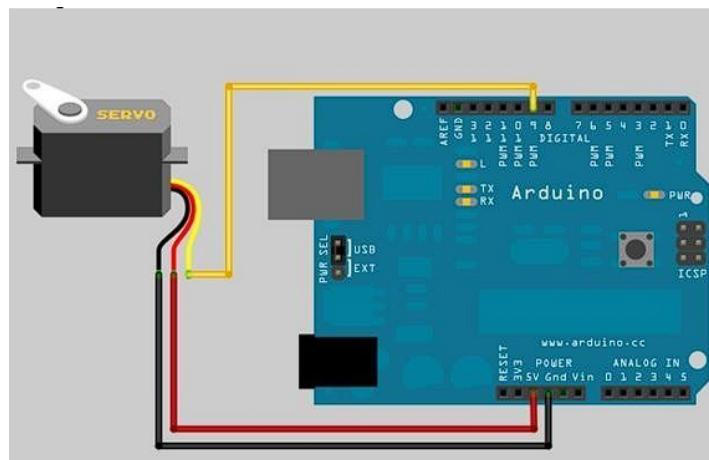


Ilustración 2 Conexión servomotor a placa arduino

Para poder controlar el servo es necesario utilizar la librería “SERVO” que se encuentra por defecto en el IDE de Arduino.

Los cables que le dan vida al servo son 3:

| Voltaje positivo | Tierra (ground) | Señal de control |
|------------------|-----------------|------------------|
| Rojo | Negro | Marrón |

Ilustración 3 Significado de las conexiones

Cable color amarillo: va conectado a uno de los pines de la placa Arduino, generalmente está conectada al pin 9.

Cable color rojo: va conectado a los 5V, que tiene como objetivo darle el voltaje necesario para que este sea utilizado.

El cable negro (imagen) o café como lo encontramos habitualmente en los servos, es conectado al GND que es la referencia de tensiones en un circuito, es decir 0voltios.

Un servo está compuesto por una serie de partes las cuales son esenciales para que al ejecutar el código cargado en la placa del Arduino este logre su funcionalidad y estas son:

- Motor de corriente continua
- Engranajes Reductores
- Sensor de desplazamiento
- Circuito de control

cada uno de ellos forman lo esencial para que un servo motor funcione, es por ello por lo que serán explicadas cada una de ellas.

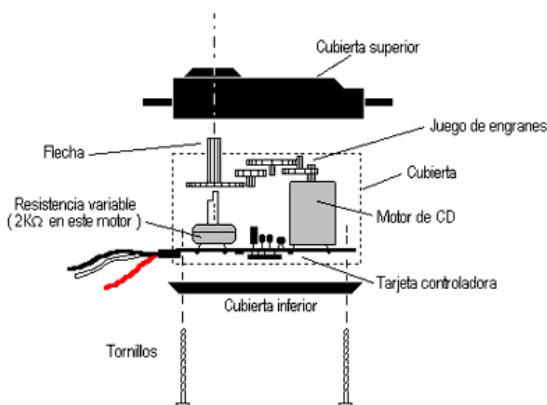


Ilustración 4 Estructura interna servomotor

PWM

Funcionamiento:

El **PWM** funciona como un botón que se activa y desactiva frecuentemente, el cual regula la cantidad de corriente y de potencia que se entrega al dispositivo que se desea controlar, estos pueden ser motores CC o fuentes de luz en CC, etc.

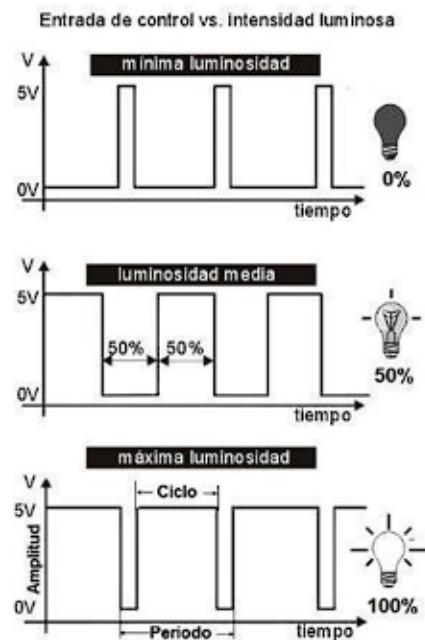


Ilustración 5 Referencia tren de pulso

El sistema del **PWM** el motor recibe corriente por un tiempo y deja de recibirlo en un periodo de tiempo, permitiendo que se repita este proceso continuamente, si se aumenta el tiempo en que la pulsación está en un mayor nivel (12 V), este recibirá más potencia en cambio sí se reduce el tiempo.

(6V) se le entregará menos potencia.

El Ciclo de trabajo o más conocido como Duty Cycle, es la relación que existe entre el tiempo en que una señal se encuentra en estado activo con la frecuencia de la misma señal que se está emitiendo.

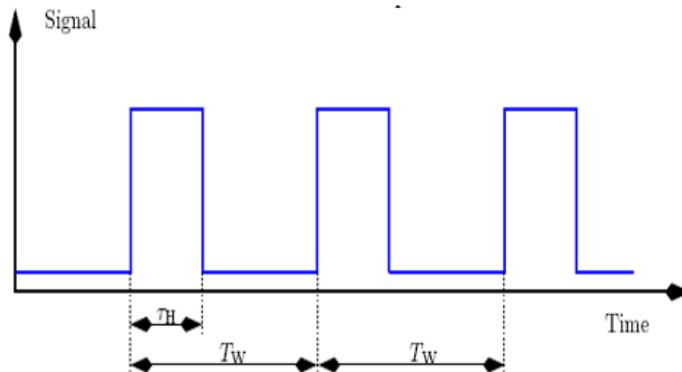
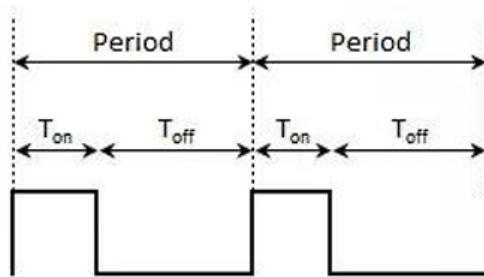


Ilustración 6 Especificación de cada parte

Para calcular el ciclo del trabajo se debe ocupar la siguiente fórmula:



$$\text{Period} = 1 / \text{Frequency}$$

$$\text{Period} = T_{on} + T_{off}$$

$$\text{Duty Cycle} = T_{on} / (T_{on} + T_{off}) * 100$$

(On Percentage)

Ecuación 1 Formula PWM

Donde

T_{on} = tiempo en estado activo

T_{off} = es el periodo o frecuencia de la señal Se multiplica por 100 para expresarlo en porcentajes.

Si como resultado da:

0% = La señal no está entregando potencia

50% =Se está entregando la mitad de la potencia

75% = La señal de salidas está entregando más de la mitad de potencia posible **100% =** La señal está entregando la totalidad de la potencia

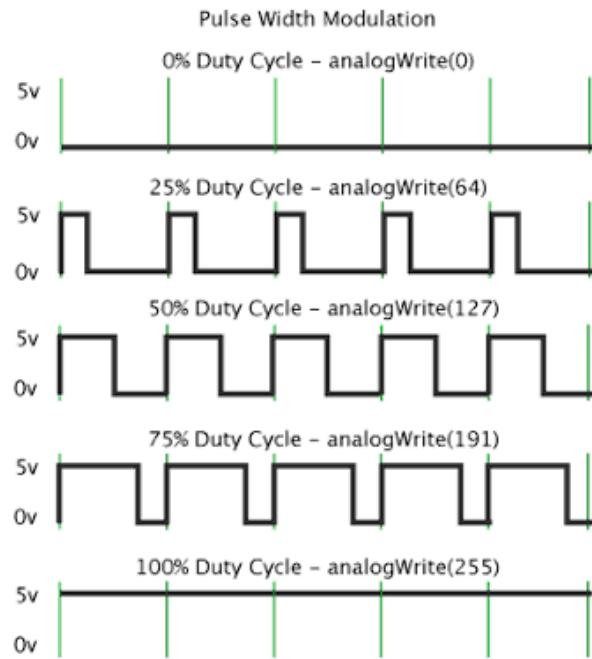


Ilustración 7 Clasificación de los PWM

Aplicaciones del PWM:

Motores: se utiliza para regular la velocidad del motor

Reguladores de luz: Controlar la potencia a la que la lámpara se encuentre utilizando.

Instrumentos

Arduino

Es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso.



Ilustración 8 Arduino

Raspberry

Este dispositivo cumple la función de ser el cerebro de todo el robot, es de bajo costo, tiene el tamaño de una tarjeta de crédito, este puede ser conectado a distintos dispositivos, tales como: un monitor de computador o televisor, además es compatible con mouse y teclados. Este dispositivo fue creado bajo el sistema operativo Linux, el cual permite a las personas explorar la computación y aprender distintos tipos de lenguajes de programación. Es capaz de replicar las acciones típicas de un ordenador de escritorio, desde navegar en internet, reproducir videos en alta resolución, manipular documentos e incluso reproducir videojuegos.

El uso de una Raspberry presenta ventajas y desventajas.

Ventajas:

- Se puede tener un programa personalizado, ya que Raspberry permite la creación de código en C, C++, Python, entre otros.
- Al poder integrar un programa personalizado deja abierto a una mayor personalización del Rover.
- Al ser de un tamaño relativamente pequeño, deja más espacio en otras partes Rover para añadir más dispositivos (ejemplo: un brazo mecánico si fuera posible o cámaras o bien detectores de gases nocivos).

Desventajas:

- Si se desea un programa personalizado se debe conseguir a un individuo con conocimientos en programación,
- Los dispositivos extra que se pueden agregar hay que ver que sean compatibles con la Raspberry o bien que se tenga el conocimiento de cómo implementarlos en el mismo.

¿De dónde nace la Raspberry Pi?

Dicho dispositivo fue creado en febrero del 2012 por la “Raspberry Pi Foundation” que originalmente fue diseñado para promover y enseñar las ciencias básicas de la computación en las escuelas y universidades del reino unido



Ilustración 9 Raspberry Pi

Cámara

Es un sensor de imagen de 8 megapíxeles Sony IMX219 de alta calidad diseñado a medida para Raspberry Pi, con un objetivo de enfoque fijo. Es capaz de imágenes estáticas de 3280 x 2464 píxeles, y también soporta vídeo 1080p30, 720p60 y 640x480p60/90 (Home, n.d.)



Ilustración 10 Cámara

Módulo bluetooth

El módulo bluetooth nos sirve como un método de transmisión de datos inalámbrica de corto alcance entre dispositivos fijos, dispositivos móviles, etc. Hay dos tipos de módulos Bluetooth de uso común en el mercado, los modelos HC-05 y HC-06. La diferencia entre ellos es que el HC-05 transmite y recibe datos y el HC-06 solamente recibe.



Ilustración 11 Módulo Bluetooth

Sensor de ultrasonido

Se utiliza un transmisor y un receptor de ultrasonidos para medir la distancia entre un objeto y el sensor, enviando un pulso de ultrasonidos y midiendo el tiempo transcurrido hasta que retorna dicho pulso.



Ilustración 12 Sensor ultrasonido

Descripción pines:

- Vcc: Pin de alimentación. (5V)
- Trigger: Pin de disparo. Este pin es una entrada, por lo que en el sistema de control, por ejemplo Arduino, se tiene que conectar a una salida.
- Echo: Este pin es una salida del sensor, por lo que ha de ser conectado a una entrada del sistema de control.
- Gnd: Pin negativo de alimentación.

Funcionamiento

Para poder realizar la medición, es necesario aplicar un pulso de 10uS en el pin trigger

A continuación el sensor envía una serie de 8 pulsos de 40 Khz a lo que posteriormente pondrá el pin de Echo a nivel alto, este permanecerá a nivel alto hasta que se reciba la señal de eco y los pulsos enviados.

Para poder saber la distancia a la que se encuentra el objeto , se mide el tiempo al que está el pin echo a nivel alto.

$$\text{Centímetros} = \mu\text{s} * 0,01715$$

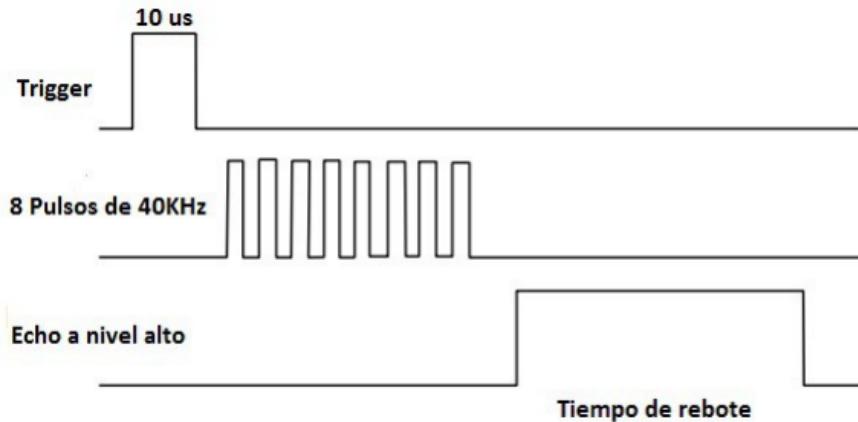


Ilustración 13 Pulos

Sensor fotosensible Lm393

Este sensor responde al cambio en la intensidad de la luz. Para esto “sensores requieren de un componente emisor que genera la luz, y un componente receptor que percibe la luz generada por el emisor” (Arce, n.d.)

Cuando la luz llega al fotorreceptor, este enciende un LED en el sensor, que emite una señal. Cuando el LED no se enciende, el sensor envía una señal HIGH. Mientras que la sensibilidad del fotorreceptor, se ajusta mediante un potenciómetro.

Tiene un rango de detección de 20 a 300mm ajustable



Ilustración 14 Sensor fotosensible

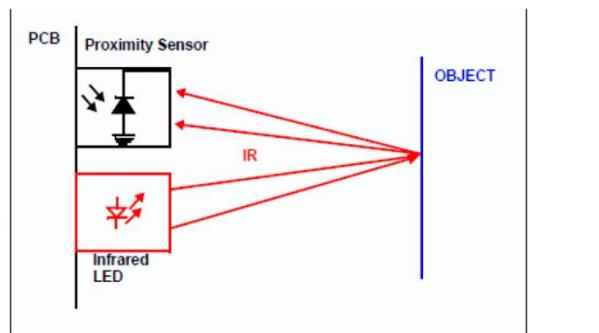


Ilustración 15 Sensor fotosensible

Sensor de Humedad y Temperatura DHT11

Este sensor incluye un componente de medición de humedad de tipo resistivo y un componente de medición de temperatura NTC, y se conecta a un microcontrolador de 8 bits de alto rendimiento, que ofrece excelente calidad, respuesta rápida, capacidad anti interferencias y rentabilidad. (DHT11 Humidity & Temperature Sensor, n.d.)

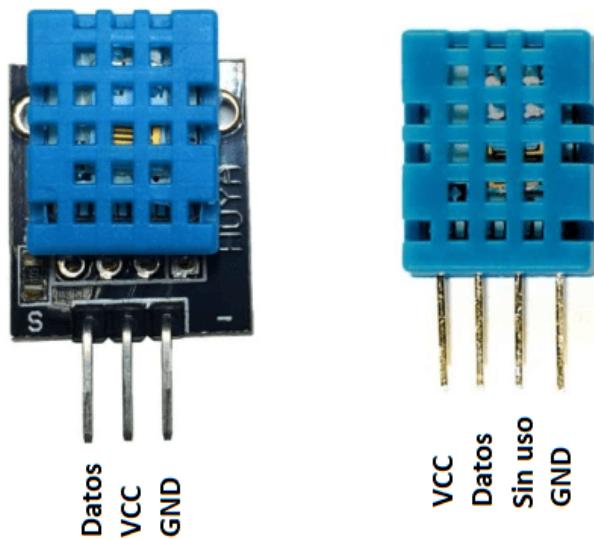


Ilustración 16 Sensor DHT11

Los pines de la versión sin PCB del DHT11 son:

- VCC: alimentación
- I/O: transmisión de datos
- NC: no conecta, pin al aire
- GND: conexión a tierra

Los pines de la versión con PCB del DHT11 son:

- GND: conexión con tierra
- DATA: transmisión de datos
- VCC: alimentación

Especificaciones detalladas:

| Parámetros | Condiciones | Mínimo | Típico | Máximo |
|--------------------------------|---------------------------|--------|-----------|--------------------------|
| Humedad | | | | |
| Resolución | | 1%HR | 1%HR | 1%HR |
| | | | 8 bits | |
| Repetibilidad | | | ±1%HR | |
| Precisión | 25°C | | ±4%HR | |
| | 0-50°C | | | ±5% de humedad relativa |
| Intercambiabilidad | Totalmente intercambiable | | | |
| Medición Rango | 0°C | 30%HR | | 90 % de humedad relativa |
| | 25°C | 20%HR | | 90 % de humedad relativa |
| | 50°C | 20%HR | | 80%HR |
| Tiempo de respuesta (Segundos) | 1/e(63%)25°C, 1m/s Aire | 6 S | 10 S | 15 S |
| Histéresis | | | ±1%HR | |
| A largo plazo Estabilidad | Típico | | ±1%HR/año | |
| La temperatura | | | | |
| Resolución | | 1°C | 1°C | 1°C |
| | | 8 bits | 8 bits | 8 bits |
| Repetibilidad | | | ±1°C | |
| Precisión | | ±1°C | | ±2°C |
| Medición Rango | | 0°C | | 50°C |
| | | | | |
| Tiempo de respuesta (Segundos) | 1/e(63%) | 6 S | | 30 S |

Ilustración 17 Especificaciones

Estado del arte

Vehículos no tripulados

Consisten en “cualquier equipo mecánico móvil que pueda transportar un objeto o sistema con cierto grado de autonomía” (Sanchez)

Los vehículos terrestres no tripulados se consideran máquinas semi autónomas o autónomas que realizan maniobras de transporte complejas y monitorean variables físicas y ambientales. Estos vehículos permiten la personalización, la optimización y la flexibilidad para satisfacer las necesidades y los desafíos de innovación en numerosas áreas de aplicación de la industria, que incluyen cartografía, agricultura, seguridad, minería, telemetría, militar, geociencias, medio ambiente y logística.

La clasificación de estos depende de:

- Propósito.
- Razones específicas, tales como, ambiente peligroso, espacio de trabajo, etc.
- Medio de ejecución, ya sea, interior, exterior, acuáticos, etc.
- Medio de movilidad, estas pueden ser, ruedas, patas, esferas, etc.

Una de las características que predominan en este tipo de vehículo es el “Sistema Robótico Portable”, el cual establece que:

- estos sean determinantes a la hora de ejecución de la tarea en áreas de alto riesgo para el usuario.
- ejecutar misión de reconocimiento y supervivencia.
- Evasión de obstáculos.
- Portable.

Algunos de ejemplos de estos son:

URBOT



Ilustración 18 URBOT

Matilda y MMP5

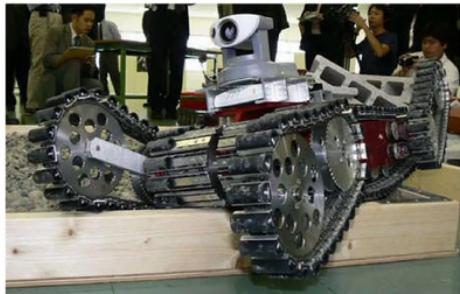


Ilustración 19 Matilda , MMPS, Matilda (II)

Ruedas para hielo

Porta luces

Otras plataformas



iRobot



Dragon Runner, CMU

Ilustración 20 otras plataformas

NASA

Los Rover han complementado las investigaciones de la NASA en relación con la investigación del planeta “Rojo”, es decir “Marte”.



Ilustración 21 Rover NASA

Tipos de Rover:

Hoy en día existen cinco Rover, los cuales son “Sojourner”, “Spirit & Opportunity”, “Curiosity” y por último el “Perseverance”.

Cada uno de ellos está diseñado para cumplir distintas misiones como por ejemplo el Rover “Perseverance”, el cual se caracteriza por buscar signos de vida pasada o presente y se estima que su llegada a Marte sea en el año 2021.

Características:

Estos están diseñados para resistir altas temperaturas, obstáculos e incluso mal tiempo, el estándar para armar estos tipos de robots son las cajas IP65, estas poseen un alto grado de protección, además de que son herméticos al polvo y protegidos contra chorros de agua potentes, otros modelos como los IP68 son herméticos al polvo pero protegido contra la inmersión completa y continua en agua, son construidos con una batería que se recarga con paneles solares en conjunto con una resistencia ante cualquier emergencia, cámaras para documentar en tiempo real, brazos para recoger muestras del suelo del planeta.

Ventajas

- Permiten una capacidad de accesibilidad a lugares remotos.
- Proporcionan información e imagen en tiempo real.

Desventajas

- La energía depende de un panel solar por lo que, si no entra en contacto con la gran estrella, este puede agotar sus reservas y quedar sin carga.

Uso de tecnología:

Los grandes avances tecnológicos han permitido que este tipo de vehículo pueda ir equipándose con distintos y mejorados implementos como son las cámaras, las ruedas, los brazos robóticos, entre otros.

Carta Gantt

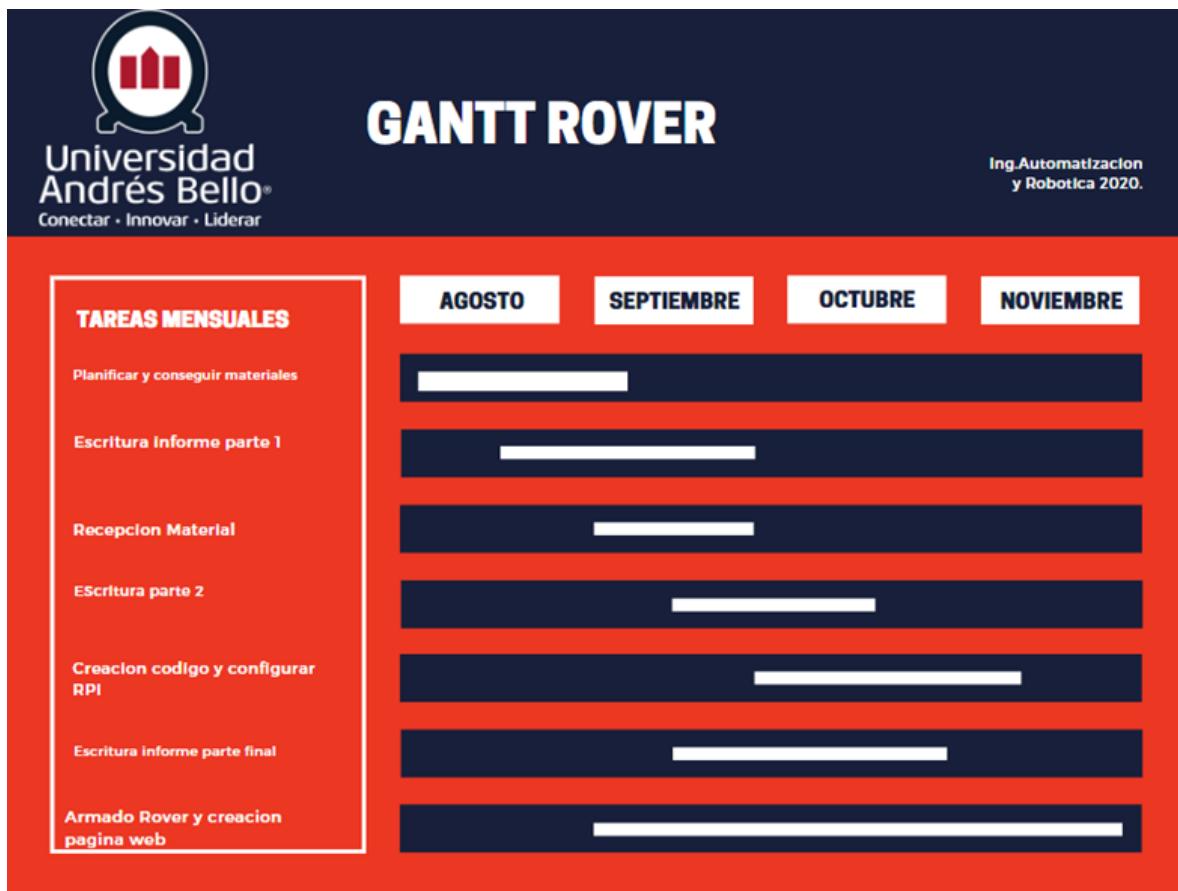


Ilustración 22 Carta Gantt

Desarrollo

Para el control de velocidad y dirección del robot, se ha colocado una placa de expansión que ha integrado el chip de accionamiento del motor. A1 A2 B1 B2 del motor arduino shield es la interfaz del motor, tanto A1 como A2 son iguales, B1 y B2 también son iguales

El motor de la interfaz A está controlado por el PIN D2, y el PIN D5 emite la señal PWG para controlar la velocidad. Para la interfaz B está controlado por el PIN D4, y la velocidad de la señal PWG de salida del PIN D6 controla la velocidad.

El valor PWM está en el rango de 0-255. Cuanto mayor sea el valor, más rápido girarán los motores

Esta shield irá montada en una placa arduino uno la cual mediante el código control “rover.ino” tendremos el control exclusivo de este.

Para el uso del módulo de la Cámara Raspberry, entraremos a la terminal de la Raspberry con “sudo raspi- config” y seleccionaremos “Interfacing Options” y activaremos “Camera”, la cámara se conectará al puerto cámara que trae la raspberry por defecto, Para generar altura a la Cámara se creará un soporte que irá montada en el segundo piso del rover.

Para evitar y detectar obstáculos utilizamos los sensores infrarrojos al observar los resultados devueltos por los sensores para determinar si se detectan obstáculos, será montado junto con la cámara raspberry y sus conexiones en los pines de raspberry

El uso de los sensores de gas, ultrasonido , temperatura y humedad están controlados mediante el código “Sensores.ino” , el sensor de gas y humedad están conectados la nodemcu ESP8266 en conjunto a placa arduino, para así aprovechar las diferentes salidas analogicas y digitales que nos proporciona cada placa de desarrollo, la cual estarán ubicadas en el tercer nivel junto con la raspberry y el soporte para la cámara.

Se ocuparan dos placas arduinos ubicadas una en el primer piso para la configuración exclusiva de los motores y el control de las ruedas del rover, mientras que la otra placa controlara la cámara, la lectura del sensor de gas, temperatura y distancia.

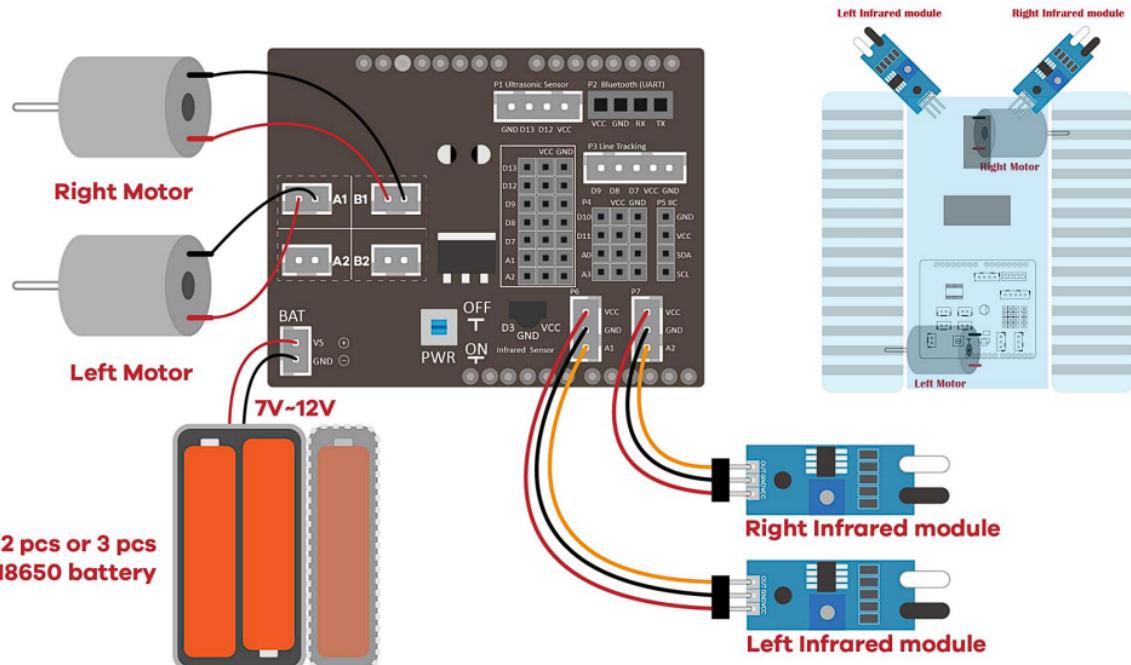


Ilustración 23 Conexiones 1

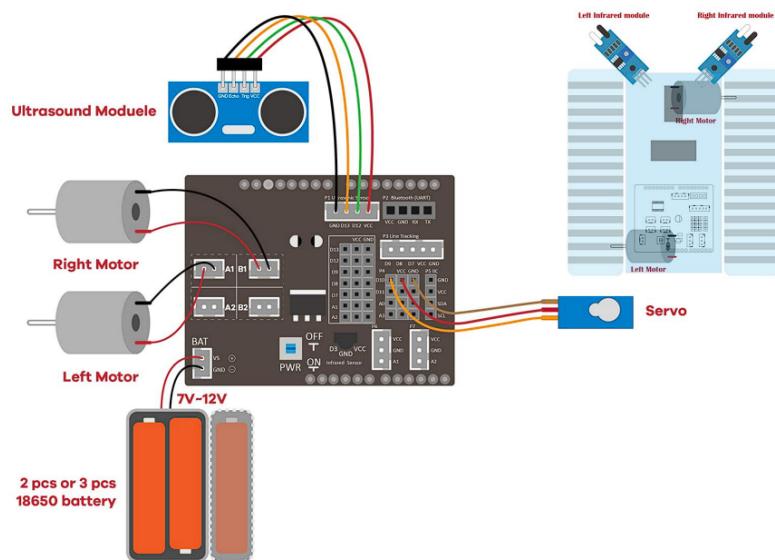


Ilustración 24 Conexiones 2

Para poder controlar al rover, se podrá ocupar un control infrarrojo o mediante la aplicación propia del rover mediante el celular "APP for LAFVIN iBot.apk" este se conectara con el rover vía bluetooth.

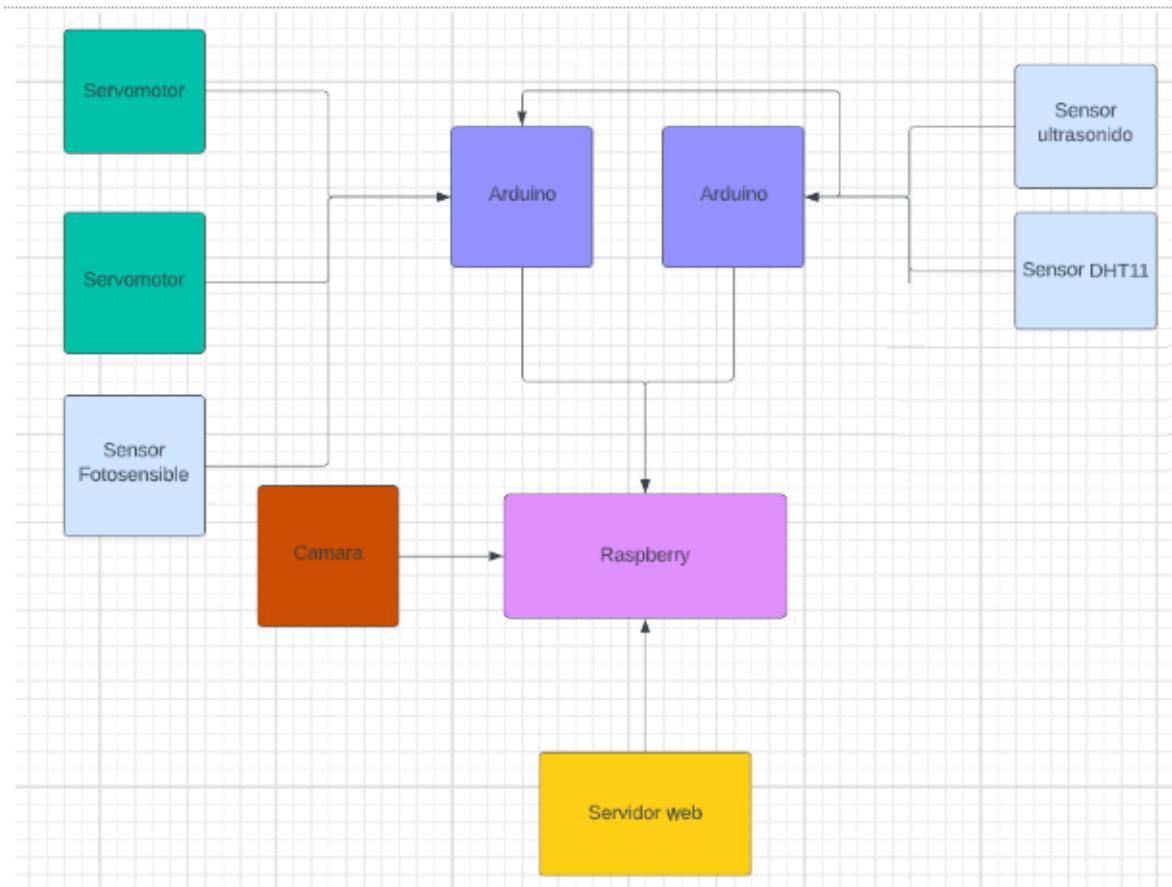


Ilustración 25 Diagrama de bloques

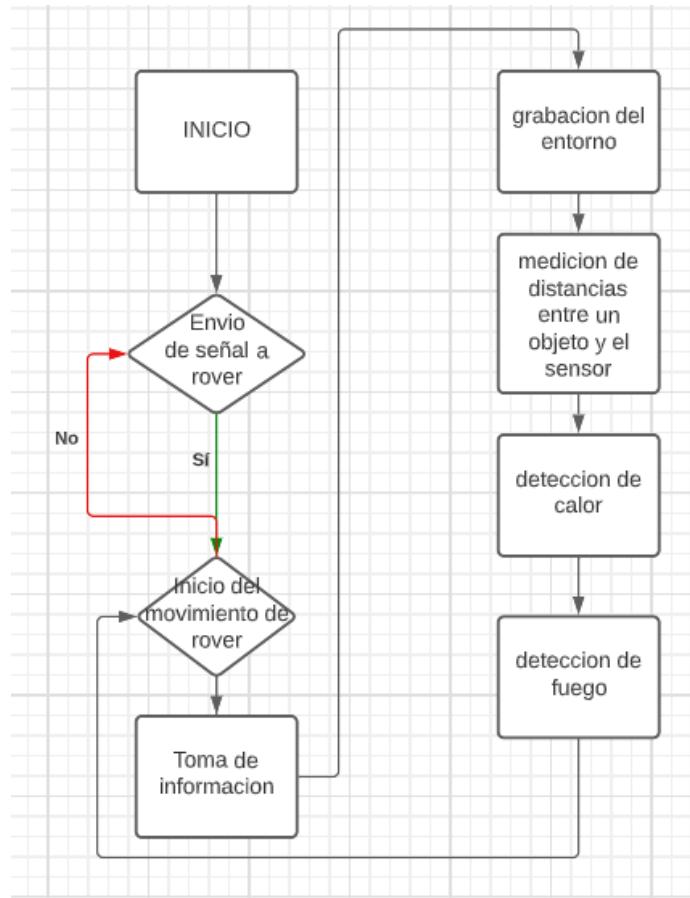


Ilustración 26 Diagrama de flujos

Costos y presupuestos

Resumen del presupuesto construcción rover

| CATEGORÍA | ESTIMADOS | REAL | DIFERENCIA | |
|--------------------------------|----------------------|----------------|-------------------|----------|
| placa arduino | \$ 10.000 CLP | 8.000 | 2.000,00 | |
| HC-sr04 | \$ 3.000 CLP | 2.990 | 10,00 | |
| sevomotor | \$ 1.980 CLP | 2.000 | -20,00 | |
| HC06 | \$ 6.000 CLP | 4.690 | 1.310,00 | |
| 2 motor dc 12V | \$ 15.000 CLP | 32.688 | -17.688,00 | |
| 3 pilas 3,3V | \$ 10.000 CLP | 9.699 | 301,00 | |
| chasis metalico + ruedas oruga | \$ 15.000 CLP | 63.972 | -48.972,00 | |
| battery case | \$ 2.500 CLP | 2.000 | 500,00 | |
| chasis acrilico | \$ - CLP | - | 0,00 | Recilado |
| 2 IR Avoid Module | \$ 5.000 CLP | 2.000 | 3.000,00 | |
| soporte sensor ultrasonico | \$ 1.000 CLP | 890 | 110,00 | |
| Gastos totales | \$ 69.480 CLP | 128.929 | -59.449,00 | |

Tabla SEQ Tabla 1* ARABIC 1 Costos y presupuestos

En la presente tabla se ilustra gastos acordes al prototipo descrito en el documento

Resultados

Como producto final obtuvimos un Rover multifunción capaz de moverse por entornos complicados gracias a su movilidad de oruga, con distintos tipos de sensores que ayudarán al usuario a analizar la situación del lugar del desastre a través de una cámara streaming monitoreada a distancia.

Adicionando un servidor web montado en Raspberry con visualización de imagen en vivo , el cual a través de node-js, html, css y node-red podemos observar

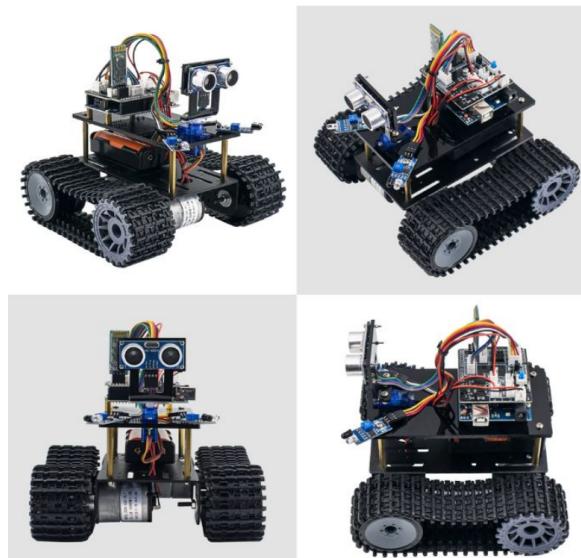


Ilustración 27 Rover proyecto V1

Discusiones

Uno de los problemas que tuvimos al realizar este proyecto fue las diferencias que tiene nuestro rover con el resto ya que mantuvimos ciertos elementos y agregamos otros.

Esto causó que tuviéramos ciertos problemas al realizar la conexión entre los sensores y el node-red.

Conclusiones

Gracias a la elaboración de investigación sobre las labores de los brigadistas se pudo observar la existencia de una gran variedad de piezas electrónicas que, en conjunto a otras, se puede lograr desarrollar un prototipo que permita mejorar la calidad de trabajo y toma de decisiones en el momento menos oportuno en la búsqueda de una persona y/o animal procurando el bienestar tanto de los usuarios como las personas afectadas.

En primer lugar, la construcción del ROVER nos permitió conocer cómo podemos ayudar de manera segura, eficiente, tecnológica y escalable proporcionándoles un vehículo no tripulado que fuese capaz de eliminar gran parte de las dificultades que se les presentan de un imprevisto.

Con la implementación de vehículos rover en situaciones de rescate se podrá ayudar a los equipos de brigada de rescate y bomberos y a futuro evitar posibles incidentes no deseados para así evitar incluso posibles víctimas en situaciones de desastre y poder salvar vidas.

Bibliografía

<https://ecuarobot.com/2022/04/14/como-funciona-el-sensor-de-humo-y-gases-inflamables-mq-2-con-arduino/>

<https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>

http://www.arduinoe.com/index.php?route=product/product&product_id=104

<https://proyectosconarduino.com/sensores/sensor-de-distancia-hc-sr04/>

<https://raspberrypi.cl/que-es-raspberry/>

Anexos

Nombre del Código control rover.ino

```
#include <Servo.h>
//sensor...
volatile int left_IR_Sensor;
volatile int right_IR_Sensor;
//Servo
Servo myservo;
//Ultrasonido-Sensor
float checkdistance() {
    digitalWrite(12, LOW);
    delayMicroseconds(2);
    digitalWrite(12, HIGH);
    delayMicroseconds(10);
    digitalWrite(12, LOW);
    float distance = pulseIn(13, HIGH) / 58.00;
    delay(10);
    return distance;
}
//detección de obstáculo
void Detect_obstacle_distance() {
    myservo.write(160);
    for (int i = 1; i <= 3; i = i + (1)) {
        DL = checkdistance();
        delay(100);
    }
    myservo.write(20);
    for (int i = 1; i <= 3; i = i + (1)) {
        DR = checkdistance();
        delay(100);
    }
}

void setup() {
    pinMode(2, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(6, OUTPUT);
    Serial.begin(9600);
    left_IR_Sensor = 0;
    right_IR_Sensor = 0;
    pinMode(A1, INPUT);
    pinMode(A2, INPUT);
    Serial.begin(9600);
    left_IR_Sensor = 0;
    right_IR_Sensor = 0;
    pinMode(A1, INPUT);
    pinMode(A2, INPUT);
    pinMode(2, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(6, OUTPUT);
```

```
//servomotor
myservo.attach(10);

//sensor-ultrasonido
pinMode(12, OUTPUT);
pinMode(13, INPUT);
Serial.begin(9600);
}

void loop() {
    digitalWrite(2,LOW);
    analogWrite(5,255);
    digitalWrite(4,HIGH);
    analogWrite(6,255);
    delay(1000);
    digitalWrite(2,HIGH);
    analogWrite(5,255);
    digitalWrite(4,LOW);
    analogWrite(6,255);
    delay(1000);
    digitalWrite(2,HIGH);
    analogWrite(5,255);
    digitalWrite(4,HIGH);
    analogWrite(6,255);
    delay(1000);
    digitalWrite(2,LOW);
    analogWrite(5,255);
    digitalWrite(4,LOW);
    analogWrite(6,255);
    delay(1000);
    digitalWrite(2,LOW);
    analogWrite(5,0);
    digitalWrite(4,HIGH);
    analogWrite(6,0);
    delay(1000);
    left_IR_Sensor = digitalRead(A1);
    right_IR_Sensor = digitalRead(A2);
    Serial.print("left_light_value = ");
    Serial.println(left_IR_Sensor);
    Serial.print("right_light_value = ");
    Serial.println(right_IR_Sensor);
    Serial.println("");
    delay(500);
    left_IR_Sensor = digitalRead(A1);
    right_IR_Sensor = digitalRead(A2);
    if (left_IR_Sensor == 0 && right_IR_Sensor == 0) {
        digitalWrite(2,LOW);
        analogWrite(5,0);
        digitalWrite(4,HIGH);
        analogWrite(6,0);
        delay(500);
        digitalWrite(2,HIGH);
        analogWrite(5,220);
        digitalWrite(4,LOW);
        analogWrite(6,220);
        delay(500);
        if (random(1, 10) > 5) {
            digitalWrite(2,HIGH);
```

```

        analogWrite(5,220);
        digitalWrite(4,HIGH);
        analogWrite(6,220);

    } else {
        digitalWrite(2,LOW);
        analogWrite(5,220);
        digitalWrite(4,LOW);
        analogWrite(6,220);

    }
    delay(500);

} else if (left_IR_Sensor == 0 && right_IR_Sensor == 1) {
    digitalWrite(2,HIGH);
    analogWrite(5,220);
    digitalWrite(4,LOW);
    analogWrite(6,220);
    delay(500);
    digitalWrite(2,LOW);
    analogWrite(5,220);
    digitalWrite(4,LOW);
    analogWrite(6,220);
    delay(500);
} else if (left_IR_Sensor == 1 && right_IR_Sensor == 0) {
    digitalWrite(2,HIGH);
    analogWrite(5,220);
    digitalWrite(4,LOW);
    analogWrite(6,220);
    delay(500);
    digitalWrite(2,HIGH);
    analogWrite(5,220);
    digitalWrite(4,HIGH);
    analogWrite(6,220);
    delay(500);
} else {
    digitalWrite(2,LOW);
    analogWrite(5,150);
    digitalWrite(4,HIGH);
    analogWrite(6,150);

}
//servo
for (int POS = 0; POS <= 180; POS = POS + (1)) {
    myservo.write(POS);
    delay(15);
}
for (int POS = 180; POS >= 0; POS = POS + (-1)) {
    myservo.write(POS);
    delay(15);
}
//sensor ultrasonido
Serial.print(checkdistance());
Serial.println("cm");
delay(100);

}

```

flama.ino

```
int buzzer = 7;          // selecciona el pin para el zumbador
int valorSensor = 0;    // variable para almacenar el valor proveniente del sensor
void setup() {
    // Seteo de la velocidad del puerto serial
    Serial.begin(9600);
    // declarar buzzer como una SALIDA:
    pinMode(buzzer, OUTPUT);
}
void loop() {
    LASER();

    // leer el valor del sensor:
    valorSensor = analogRead(A0);
    Serial.println(valorSensor);
    // activa el buzzer
    if (valorSensor < 500){
        digitalWrite(buzzer, HIGH);
        delay(100);
        // Desactiva el buzzer
        digitalWrite(buzzer, LOW);
        delay(50);
    }
}
void LASER(){
    digitalWrite(laserPin, HIGH);    // poner el Pin en HIGH
    delay(5000);                  // esperar 5 segundos
```

```
    digitalWrite(laserPin, LOW);      // poner el Pin en LOW
    delay(20000);                  // esperar 20 segundos
}
```

código esp.ino

```
#include <DHT.h> // Including library for dht
#include <ThingSpeak.h>;
#include <ESP8266WiFi.h>

String apiKey = "N02FGQPWV76QJBR"; // Enter your Write API key from
ThingSpeak

const char *ssid ="VTR-3122195";
const char *pass ="republica2256familiaavenegasfuenzalida1009";
const char* server = "api.thingspeak.com";
int sensor = A0;
unsigned long myChannelNumber = 1947227;
const char * myWriteAPIKey = "XRH7CRTRZU08DJZC";
int laserPin = D4;
#define DHTPIN 0 //pin where the dht11 is connected

DHT dht(DHTPIN, DHT11);

WiFiClient client;

void setup()
{
    Serial.begin(115200);
    delay(10);
    dht.begin();

    Serial.println("Connecting to ");
    Serial.println(ssid);
```

```
WiFi.begin(ssid, pass);

while (WiFi.status() != WL_CONNECTED)

{

    delay(500);

    Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

ThingSpeak.begin(client);

pinMode(sensor, INPUT);

}

void loop() {

    tempHum();

    gas();

}

void tempHum() {

    float h = dht.readHumidity();

    float t = dht.readTemperature();

    if (isnan(h) || isnan(t))

    {

        Serial.println("Failed to read from DHT sensor!");

        return;
}
```

```
}

    if (client.connect(server,80)) // "184.106.153.149"
or api.thingspeak.com

{



    String postStr = apiKey;

    postStr += "&field2=";

    postStr += String(t);

    postStr += "&field3=";

    postStr += String(h);

    postStr += "\r\n\r\n";



    client.print("POST /update HTTP/1.1\n");

    client.print("Host: api.thingspeak.com\n");

    client.print("Connection: close\n");

    client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");

    client.print("Content-Type:

application/x-www-form-urlencoded\n");



    client.print("Content-Length: ");

    client.print(postStr.length());



    client.print("\n\n");



    client.print(postStr);






    Serial.print("Temperature: ");

    Serial.print(t);

    Serial.print(" degrees Celcius, Humidity: ");

    Serial.print(h);

    Serial.println("%. Send to Thingspeak.");





}
```

```
client.stop();

Serial.println("Waiting...");

// thingspeak needs minimum 15 sec delay between updates
delay(1000);
}

void gas(){
    int val = analogRead(sensor); //Read Analog values and Store in val
variable

Serial.println(val); //Print on Serial Monitor

if (val>500){
    Serial.println("Bad air quality.");
}
else {
    Serial.println("Air is safe to breathe");
}

delay(1000);

int x = ThingSpeak.writeField(myChannelNumber, 1,val, myWriteAPIKey);
//Update in ThingSpeak

if(x == 200){
```

```
    Serial.println("Channel update successful.");
}

else

{
    Serial.println("Problem updating channel. HTTP error code " +
String(x));

}

delay(1500);
}
```