



UNIVERSIDAD ANDRÉS BELLO
FACULTAD DE INGENIERÍA

INGENIERÍA EN AUTOMATIZACIÓN Y ROBOTICA

“Rover en Búsqueda y Rescate”

Alumnos:

Valdivia Jofré, Bastián A.
Venegas Fuenzalida, Matías I.
Vargas Espinoza, Pablo L.
Beltrán Vargas, Cristóbal.

Profesor:

Palominos Néstor.

SANTIAGO – CHILE
2020

Contenido

<i>Indice de figuras.....</i>	<i>5</i>
Glosario	6
Abstract.....	7
Introducción.....	8
Capítulo I El Problema	9
Objetivos.....	9
<i>Generales.....</i>	<i>9</i>
<i>Específicos:.....</i>	<i>9</i>
Alcances y limitaciones	10
Capitulo II Antecedentes	10
<i>Marco Teórico:.....</i>	<i>10</i>
Tipos de Rover:	10
Características:.....	11
Uso de tecnología:	11
¿En qué se diferencia nuestro producto al resto?.....	11
Características del prototipo	12
<i>Diseño:</i>	<i>12</i>
<i>Explicación Proyecto:</i>	<i>12</i>
<i>Propósito:.....</i>	<i>12</i>
Capitulo III Metodología	13
Presupuesto	13
<i>Costeo por unidad de producción</i>	<i>13</i>
Desarrollo	14
Instrumentos	15
Servomotor Arduino:	15
<i>Componentes y Funcionalidad de un Servo:</i>	<i>15</i>
TIPOS DE SERVOMOTOR:.....	16
<i>PWM (MODULACION POR ANCHO DE PULSOS):</i>	<i>17</i>
Ventajas:	17
Desventajas:.....	17
<i>Funcionamiento:</i>	<i>18</i>
Raspberry PI	19
¿Qué es?.....	19
Ventajas:	19

Desventajas:.....	19
<i>¿de dónde nace la Raspberry pi?</i>	19
Características:.....	20
Puente H (L298d)	22
<i>¿para qué sirve?</i>	22
Funcionamiento:.....	22
Controlador de motores con Arduino	24
<i>Shield L293D</i>	24
<i>Características</i>	25
Aplicaciones	25
MEDIDOR DE DISTANCIA LASER (VL53L0X).....	26
<i>¿Qué es?</i>	26
<i>¿Cómo funciona?</i>	26
Posee 4 funciones diferentes:	26
Funcionamiento Default y largo alcance:	27
<i>Especificaciones y características:</i>	27
Pruebas.....	29
Diagrama de bloque.....	35
Diagrama de flujo	36
Bibliografía.....	37
Anexos	38
<i>Código Sensor_MQ9.ino</i>	38
<i>Código Control_Ruedas_Rover.ino</i>	39
<i>Código LM393.ino</i>	40
<i>Código Camera_pi.py</i>	41
<i>código appCam.py</i>	42
<i>código Control_direccion_Camara.ino</i>	43
<i>Código vl53l0x.ino</i>	43

índice de figuras

figura 1. rover.....	10
figura 2 " Costo producción Rover”	13
figura 3. diagrama de conexion servo motor	15
figura 4. estructura interna servo motor.....	15
figura 5. servo motor analógico.....	16
figura 6. micro servo motor digital.....	17
figura 7. referencia PWM.....	18
figura 8. ciclo de trabajo PWM	18
figura 9. modelo A y B.....	20
figura 10. Raspbberry pi 3 modelo B.....	20
figura 11. Raspbberry pi 3 modelo B pinout	21
figura 12. L298D	22
figura 13. pin connection.....	23
figura 14 pin connection.....	24
figura 15 pin connection.....	25
figura 16. VL53L0X.....	26
figura 17. medición de distancia.....	28
figura 18. Total de piezas del proyecto	29
figura 19. ensamblaje motores DC	29
figura 20. ensamblaje Arduino	30
figura 21”ensamblaje cámara web y Raspbery”	30
figura 22 conexion MQ9 y servo motor	31
figura 23 instalacion sensor de vuelo	32
figura 24 conexion pines a raspberry sensor de vuelo.....	33
figura 25 vista lateral rover	33
figura 26 vista rober frontal.....	34
figura 27 vista rover parte trasera.....	34

Glosario

Brigadistas: Miembros de una unidad militar compuesta por varios regimientos o batallones de un arma determinada y mandada normalmente por un general de brigada

INSARAG: Siglas de: Grupo Asesor Internacional de Operaciones de Búsqueda y Rescate

USAR: Siglas en Ingles para: Urban Search and Rescue

Chasis: Armazón que sostiene el motor y la carrocería de un vehículo

Bipolar: Que tiene dos polos

Unipolar: Qué tiene un solo polo

Abstract

This project consists of the creation of an unmanned vehicle that is capable of transmitting a live image of the place where you are.

It will consist of a device that will allow the detection of the most common gases in the home (CO₂, Natural Gas, among others).

The management of this would allow a remote connection through the WIFI signal which will allow us to control the vehicle with the help of a computer keyboard we will be able to guide the direction in which this position and even the total control of the two axes of the camera web that will be transmitted through a customized website that will allow us to know the functions of the "Rover," and as a little tutorial for connecting this.

This device will be created in order to be able to improve the service of the Brigadistas in Chile and obtain a better result in their rescue operations.

The Vehicle is 4X4 wheels, has a camera connected by WIFI, a thermal sensor and a laser distance sensor, to detect dangerous areas, solar panels to power the robot, which will explore the less accessible or more dangerous areas for people of the Brigadistas.

Resumen

Este proyecto consiste en la creación de un vehículo no tripulado que es capaz de transmitir una imagen en vivo del lugar donde se encuentra.

Consistirá en un dispositivo que permitirá la detección de los gases más habituales en el hogar (CO₂, Gas Natural, entre otros).

La gestión de este permitirá una conexión remota a través de la señal WIFI lo que nos permitirá controlar el vehículo con la ayuda de un teclado del ordenador podremos orientar la dirección en la que se encuentra esta posición e incluso el control total de los dos ejes de la cámara web que se transmitirá a través de un sitio web personalizado que nos permitirá conocer las funciones del "Rover" y como un pequeño tutorial para conectar esto.

Este dispositivo se creará con el fin de poder mejorar el servicio de los brigadistas en Chile y obtener un mejor resultado en sus operaciones de rescate.

El Vehículo es de ruedas 4X4, tiene una cámara conectada por WIFI, un sensor térmico y un sensor de distancia láser, para detectar áreas peligrosas, paneles solares para alimentar el robot, que explorará las zonas menos accesibles o más peligrosas para la gente de los Brigadistas.

Introducción

En este informe se explicará de manera exhaustiva la creación de un **“Rover”**, el cual será diseñado para explorar espacios terrestres pocos accesibles para el ser humano.

Si bien hoy en día existen una cantidad de “Rover” en el área de la astronomía y exploración de planetas no habitables, con este prototipo buscaremos implementar un vehículo no tripulado en el aérea de Búsquedas y Rescates en accidentes de difícil acceso, en el cual se pueda observar a través de una imagen las dimensiones de la catástrofe y señales altas temperaturas.

Con este dispositivo se buscará un trabajo más eficiente en el área de rescates de difícil acceso, en el cual el usuario podrá implementar un plan de trabajo más preciso disminuyendo la posibilidad de algún accidente laboral.

Capítulo I El Problema

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

La Nasa a implementado distintos modelos para la investigación del espacio, logrando avances de grandes magnitudes en el área Aeroespacial, pero ¿Qué tiene que ver esto en el área de Búsqueda y Rescate?

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 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 segura o alguna emanación de calor.

Esto permitirá a que los brigadistas puedan ejecutar de una manera efectiva el plan anteriormente conversado disminuyendo los riesgos de lesiones u desapariciones repentinas.

Objetivos

Generales

En el presente informe se propone la realización de un” **Rover**”. Con el objetivo de ayudar a los Brigadistas a desarrollar de forma más efectiva 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.

Específicos:

- Lograr diseñar un prototipo moldeable y resistente
- Poder incluir un modelo agradable a la vista y simple de controlar mediante WIFI
- Cumplir con las expectativas del área seleccionada
- Una duración de largas horas frente a condiciones extremas.

Alcances y limitaciones

Este dispositivo será capaz de moverse y adecuarse a los distintos problemas que se presente en terreno, por lo cual será controlado por personal a cargo de las labores de Búsqueda y Rescate que se encuentre en el lugar de los incidentes.

Una de sus limitaciones es que será diseñado exclusivamente para el equipo de brigadistas de Búsqueda y Rescate puesto que se enfocará en brindarles un mejor plan de trabajo.

Si bien este prototipo será exclusivo de estos brigadistas, este podrá ser modificado de acuerdo de a las exigencias del operador.

Capitulo II Antecedentes

Marco Teórico:

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

Tipos de Rover:

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

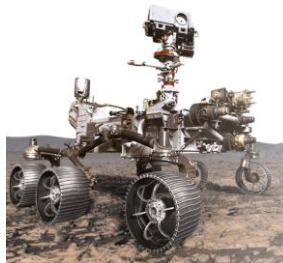


figura 1. rover

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 en 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 contruidos 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 (investigar IP65)

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 puedan ir equipándose con distintos y mejorados implementos como son las cámaras, las ruedas, los brazos robóticos, entre otros.

¿En qué se diferencia nuestro producto al resto?

Este tipo de vehículo nos ha entusiasmado para poder crear un prototipo e implementarlo en el área de “Búsqueda y Rescate” por lo que este contara con:

- sensor térmico de temperatura LM93
- motores DC
- Raspbery 3
- cámara RPI
- panel solar
- servomotores
- chasis doble automóvil pequeño
- sensor de distancia laser (VL53L0XV2)
- Modulo 134N3P Cargador y Regulador Boost Bateria Litio

Características del prototipo

Diseño:

Este vehículo consta de un chasis doble (de dos pisos) el cual nos permite tener mayor control y visualización de los componentes presentes, y mayor accesibilidad a la hora de visualizar algún componente en desperfecto o que se encuentre desconectado

Explicación Proyecto:

El Rover contemplara: chasis de auto con sus respectivos ejes, Raspberry 3 B, cámara rpi, panel solar, 4 motores DC, sensor de distancia laser, sensor de gas de la familia MQ, sensor de temperatura LM393

Esto nos permitirá visualizar en tiempo real las presencias de altas temperaturas, gas inflamable y una imagen en vivo del lugar donde se encuentre el Rover

Propósito:

El fin de este proyecto es poder mejorar las labores y seguridad del equipo de brigadistas que trabajan en circunstancias

Capítulo III Metodología

La metodología fue basada en las habilidades de cada compañero procurando que cada uno pueda ir puliendo sus talentos. Además de incorporar los resultados obtenidos de las pruebas realizadas a través del aprendizaje “ensayo y error”.

Para la realización de este prototipo captamos como referencia los Rover utilizados por la NASA de los cuales se moldearán para que pueda ser utilizados en el área de Búsqueda y Rescate.

Por ello realizaremos un prototipo el cual se capaz de cargarse con panel solar, transmitir imagen a través de una cámara integrada, que sea controlado por computadora y con un sensor de calor.

Presupuesto

Costeo por unidad de producción

cantidad	producto	proveedor	precio
1	Cargador Y Regulador Boost Bateria Litio Arduino Raspberry	Jorge Olea	\$ 3.500
1	Sensor De Distancia Láser 50 A 1200mm Modelo VL5310x	Ingeniería José Luis Muñoz Parra E.i.r.l	\$ 4.999
1	Kit 9 Sensores De Gas Mq - Arduino - Raspberry	Afel Ingeniería Spa	\$ 23.218
1	Chasis Auto Smart Car 4wd 4 Ruedas Arduino	Electroardu Limitada	\$ 17.990
1	Raspberry Pi 3 Modelo B	ingenieria Mci Limitada	\$ 48.390
1	Cámara Raspberry Pi 5mp Rev 1.3 Con Conector Plano	Ingeniería José Luis Muñoz Parra E.i.r.l	\$ 9.995
1	kit Solar	ingenieria Mci Limitada	\$ 22.990
1	l293d	Mercado Libre	\$ 4.790
1	micro servo motor	Afel Ingeniería Spa	\$ 1.990
1	Sensor Temperatura lm393	Mercado Libre	\$ 6.600
Total	7	\$	144.462

figura 2 " Costo producción Rover"

Desarrollo

Para los Motores de continua soldaremos dos cables a cada uno para suministrarle energía, los cuales irán montados en un chasis doble de carro posteriormente conectados a los pines de motor 1;2;3 y 4 de la Shield I293 que nos permitirá tener un control exclusivo de la dirección del vehículo. Esta Shield ira montada en una placa Arduino uno la cual mediante el código Control_Ruedas_Rover.ino, el cual podremos tener control exclusivo de este.

Para el uso del módulo de la Cámara Raspberry, entraremos al a terminal de la Raspbberry con “`sudo raspi-config`” y activaremos en “Interfacing Options” y activaremos “Camera” y el módulo la conectaremos en el puerto “camera” que trae por defecto la Raspbberry, Para generar altura a la Cámara se creará un soporte montado en un servo motor, el cual estará conectado al pin numero 13 para que este permita el giro proporcionado por el código Control_direccion_Camara.ino. Estas instalaciones irán montadas en el segundo nivel del chasis del vehículo

El uso del sensor de gas MQ9 y el sensor de temperatura lm393 serán conectadas a una protobaord conectada a placa Arduino el cual estarán ubicadas en el segundo nivel junto a la Raspbberry y el soporte de Cámara estos dispositivos mostrarán su lectura en el código Sensor_MQ9.ino y LM393.ino

Usaremos el sensor vl53L0X capaz de detectar movimiento con precisión laser montado en conjunto con la Cámara Raspberry controlado mediante el código V153L0X.ino y sus conexiones en los pines de Raspbberry.

Instrumentos

Servomotor Arduino:

Componentes y Funcionalidad de un Servo:

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.

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
		

- 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 0 voltios.

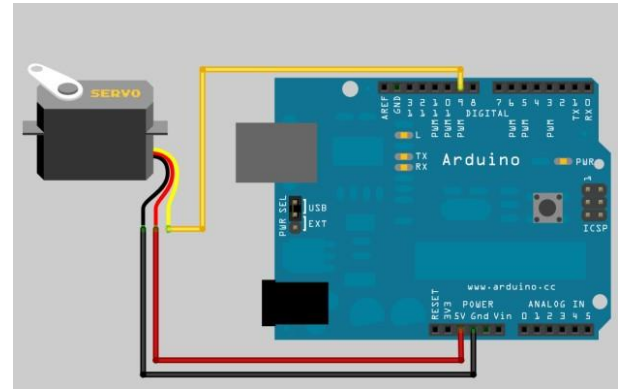


figura 3. diagrama de conexion servo motor

Un servo este 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.

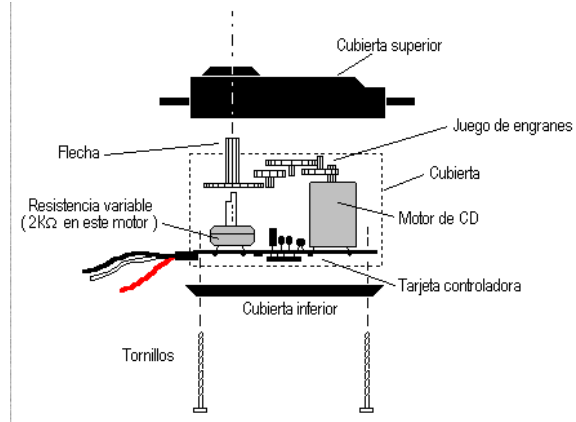


figura 4. estructura interna servo motor

- Un motor de corriente continua o más conocido como DC es el componente motriz del servo el cual le brinda movimiento.
- El conjunto de engranajes que lo conforman tiene la funcionalidad de disminuir la velocidad elevada del giro del DC.
- Es un potenciómetro el cual va en el eje de salida del servo y su funcionalidad es
- Es una pequeña placa electrónica que compara la señal de referencia de entrada con la posición medida por el potenciómetro y así lograr amplificar y mover el servo en la dirección requerida y disminuir el error.
- determinar la posición

TIPOS DE SERVOMOTOR:

Existen diferentes formatos de un servomotor, los cuales todos cumplen la misma función que es girar en un ángulo determinado.

Existen dos tipos de estos que son los **digitales y analógicos**, ambos presentan características similares, debido a que poseen una organización similar entre ellos. La diferencia principal entre ambos tipos se encuentra en la manera en la que se procesa la señal recibida desde el receptor, y en como controla el envío de potencia al servo.



figura 5. servo motor analógico



figura 6. micro servo motor digital

PWM (MODULACION POR ANCHO DE PULSOS):

Es un método para generar una señal analógica utilizando una fuente digital, a su vez consta de dos componentes principales que definen su comportamiento: un ciclo de trabajo y una frecuencia.

Ventajas:

- Funcionamiento de bajo ruido
- En el controlador los niveles de brillo más bajos siempre mantienen a las pérdidas de conducción más bajas

Desventajas:

- El parpadeo se percibe en la visión periférica si el controlador se ejecuta a 100 Hz de frecuencia, por lo que se le hace más costo.
- Tiene un efecto estroboscópico evidente en entornos de movimiento rápido cuando la frecuencia es baja en el controlador.
- Si el controlador se monta de forma remota desde la fuente de luz causa un problema de rendimiento.

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.

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 si se reduce el tiempo (6V) se le entregara menos potencia.

El Ciclo de trabajo o más conocido como Duty Cycle, es la relación que existe entre el

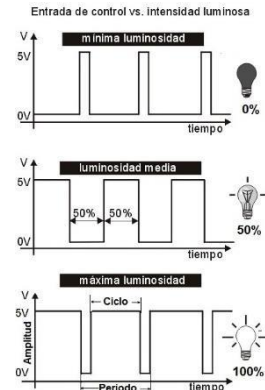


figura 7. referencia PWM

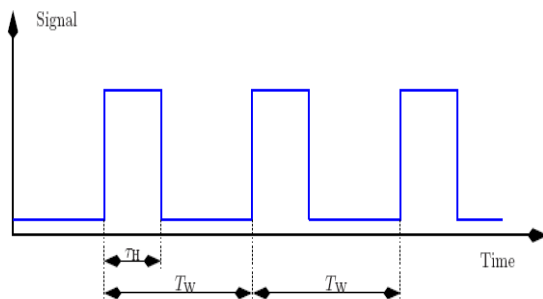


figura 8. ciclo de trabajo PWM

tiempo en que una señal se encuentra en estado activo con la frecuencia de la misma señal que se está emitiendo. Para calcular el ciclo del trabajo se debe ocupar la siguiente formula:

Donde

- T_H = tiempo en estado activo
- T_W = 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

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.

Raspberry PI

¿Qué es?

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 conectada a distintos dispositivos, tales como: un monitor de computador o a un 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 presentara ventajas y desventajas.

Ventajas:

- Se puede tener un programa personalizado ya que la 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 originalmente se lanzaron dos modelos A Y B.



figura 9. modelo A y B

MODELO PARA UTILIZAR EN EL PROYECTO:

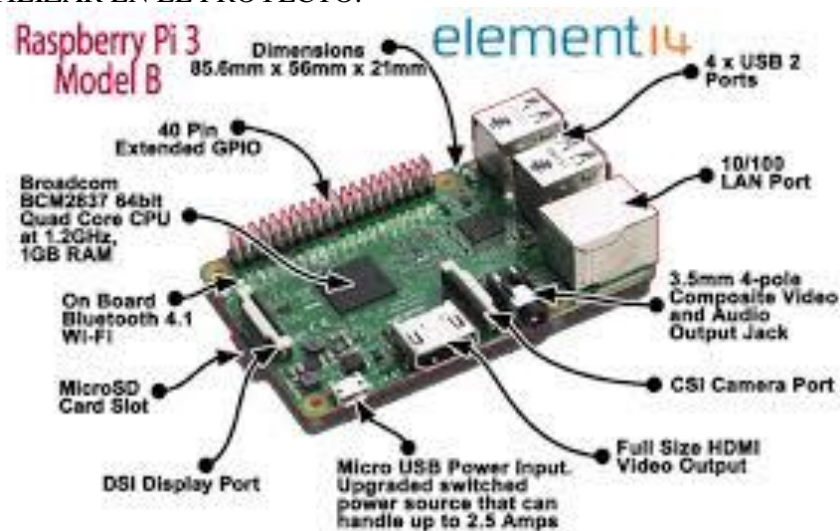


figura 10. Raspbberry pi 3 modelo B

Características:

- Chipset Broadcom más a 1,2 GHz
- ARM Cortex-A53 de 64 bits y cuatro núcleos
- LAN inalámbrica 802.11 b/g/n
- Bluetooth 4.1 (posee modo bajo consumo)
- Coprocesador multimedia de doble núcleo Video Core IV®
- Memoria LPDDR2 de 1 GB
- Compatible con todas las últimas distribuciones de ARM GNU/Linux y Windows 10 IOT
- 1x Puerto Ethernet 10/100
- 1x Conector de vídeo/audio HDMI
- 1x Conector 3.5mm audio/video compuesto.
- 4x Puertos USB 2.0

- 40x Pines GPIO
- Antena de chip (Wifi y Bluetooth integrados)
- Conector de pantalla DSI
- Ranura de tarjeta microSD
- Dimensiones: 85 x 56 x 17 mm
- Amperaje: 2,5A

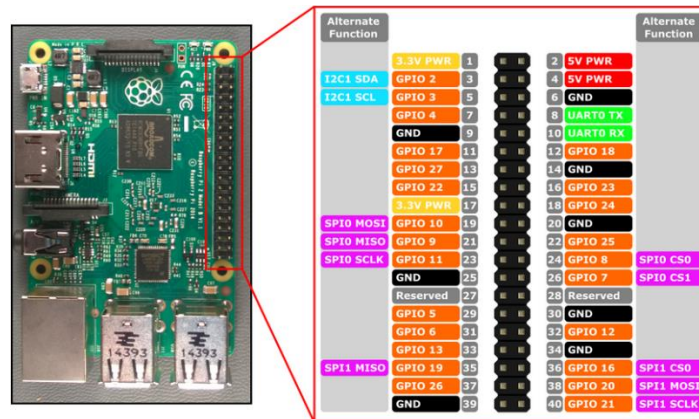


figura 11. Raspberry pi 3 modelo B pinout

Puente H (L298d)

¿para qué sirve?

Este dispositivo fue diseñado para facilitar el control de motores con Arduino, la función de estos es tomar la señal de control de baja corriente y luego convertirla en una señal de corriente más alta que pueda conducir un motor.

El puente H, es un modulo que permite el control de la velocidad y el sentido de giro en motores DC o motores de paso a paso bipolar/unipolares, por medio de una señal TTL (lógico transistor a transistor). Puede estar conectado a un Arduino o una Raspberry, entre otros microcontroladores.

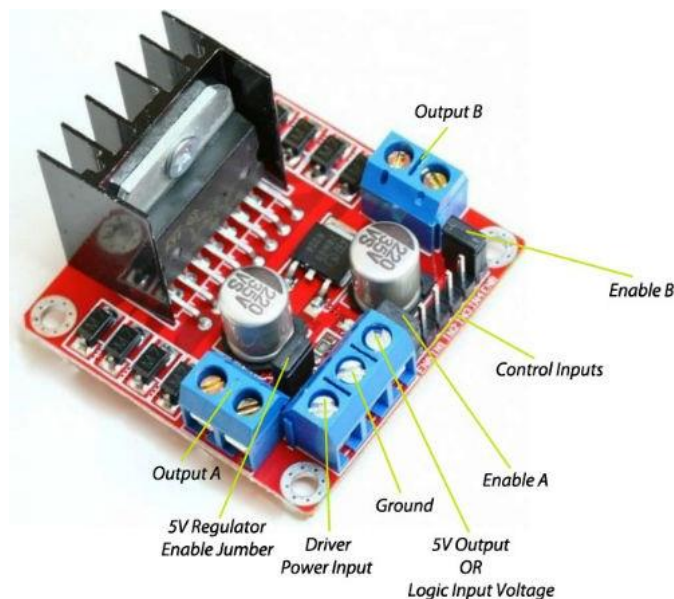


figura 12. L298D

Funcionamiento:

El modulo posee un regulador integrado LM7805, que le da la propiedad de alimentarse de dos maneras distintas;

- si el jumper de selección esta activo, el módulo podrá usar alimentación de 6V hasta 12V DC, como tendrá el regulador activado el pin +5V tendrá un voltaje de 5V DC
- si el jumper esta inactivo se podrá usar una alimentación de 12V hasta 35V DC y como el regulador no esta funcionando, se tendrá que conectar el pin de +5V a la tención de 5v para la alimentación lógica del modulo

Especificaciones y características:

- Tensión de alimentación mínima: 5 Volts
- Amperaje Máximo: 4A
- Voltaje lógico: 5 Volts
- Voltaje de Operación: 5 Volts – 35 Volts
- Posee dos entradas: una de 5 volts para controlar la parte lógica
- Alimentación para la salida a los motores: 5 volts o más
- Posee un regulador de voltaje de 5V 78M05, para alimentar la etapa lógica del L298N, sin embargo, cuando la alimentación supera los 12 V, se recomienda, utilizar una fuente de 5 V externa como fuente de alimentación
- Consumo de corriente digital: 0 a 36 mA
- Capacidad de corriente: 2A (picos de hasta 3A)
- Potencia máxima: 25 Watts
- Admite entradas de señal PWM para el control de velocidad
- Posee 8 diodos de protección contra corriente inversas
- Chip: L298N
- Canales: 2 (soporta 2 motores DC o 1 motor paso a paso)
- Peso: 30 gramos
- Dimensión: 43 x 43 mm
- Altura: 27 mm

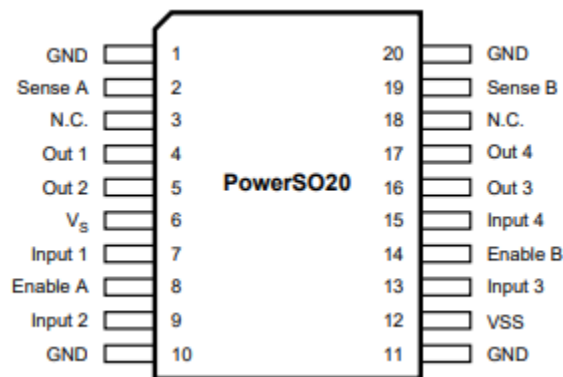


figura 13. pin connection

Controlador de motores con Arduino

Este dispositivo fue diseñado para facilitar el control de motores con Arduino, la función de estos es tomar la señal de control de baja corriente y luego convertirla en una señal de corriente mas alta que pueda conducir un motor.

Los con controladores (en ingles Driver) de motores son un sistema. La carga de un sistema de diferentes dispositivos, ya sea de una herramienta de corte de una máquina, el compresor de un acondicionador de aire, el brazo de un robot y muchos más, pero nosotros lo usaremos (especificar si se usara para las ruedas del Rover únicamente o en un caso un brazo robótico).

El boque convertidor electrónico de potencia puede usar diodos, MOSFETS, GTO, IGBT o tristores.

El que nosotros usaremos es el modelo

Shield L293D

Hay diferentes modelos de los controladores de motor, pero nosotros usaremos el modelo Shield L293D, este modelo está diseñado para proveer corriente de control bidireccional para motores de paso y de DC, Puede controlar motores de 4.5V hasta los 36V con una corriente pico máxima de 1.2A, el L293D usa el sistema de puente en H. Es un sistema para controlar el sentid de giro de un motor DC usando cuatro transistores y también la velocidad del motor.

En los modelos L293 es muy importante utilizar diodos para evitar dañar el integrado con corriente parasitas generadas con los propios solenoides de las cargas, en cambio el modelo L293D ya posee protección de estas dando libertad de otras opciones ya sea para un uso más simple y/o económico.

Para controlar la velocidad del motor se usa la técnica de PWM, se conoce que hay que utilizar los pines 2 y 7 del L293D desde dos salidas del Arduino. En estas dos salidas habrán PWM a cada una, pero tenemos que invertir un PWM, esto quiere decir que cuando un PWM tenga un pulso de alto valor el otro debe tener la misma cantidad, pero en negativo.

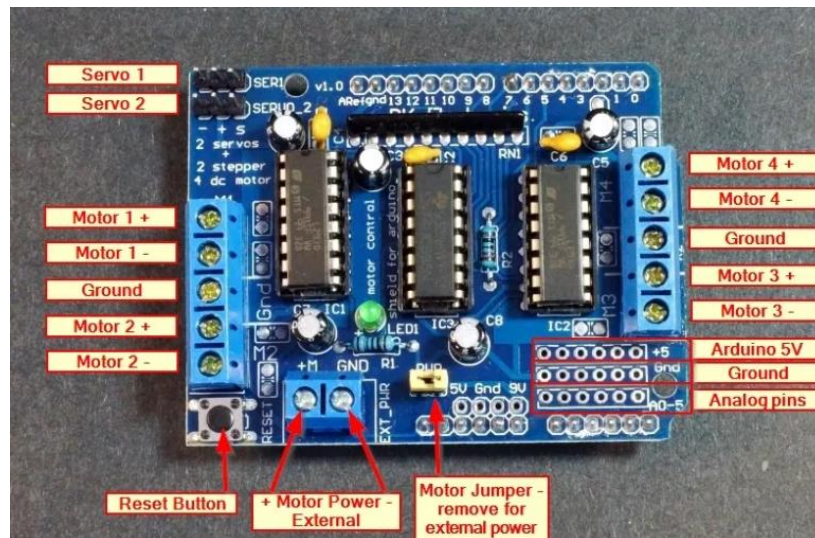


figura 14 pin connection

Características

- Alimentación: 45 a 36 VDC
- Corriente de salida: 600 mA.
- Corriente pico de salida: 1 A por canal (no repetitiva).
- Encapsulado: DIP de 16 pines.
- Alta inmunidad al ruido eléctrico.
- Protección contra exceso de temperatura.
- Diodos de protección (flyback) incorporados

Aplicaciones

- Control de cargas inductivas: Relevos, Solenoides, Motores DC, Motores PAP.
- Puentes H.

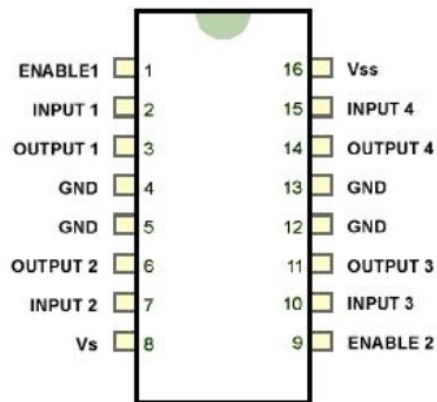


figura 15 pin connection

MEDIDOR DE DISTANCIA LASER (VL53L0X)

¿Qué es?

El VL53L0X es un sensor que mide distancia por tiempo de vuelo (ToF; time of flight), tiene la capacidad de medir distancia sin importar la irregularidad del terreno, con un rango de 5cm hasta 2m.

Este sensor laser se diferencia de los sensores de ultrasonido en que posee una mayor precisión además de que la luz ambiental no le supone un problema muy grande.

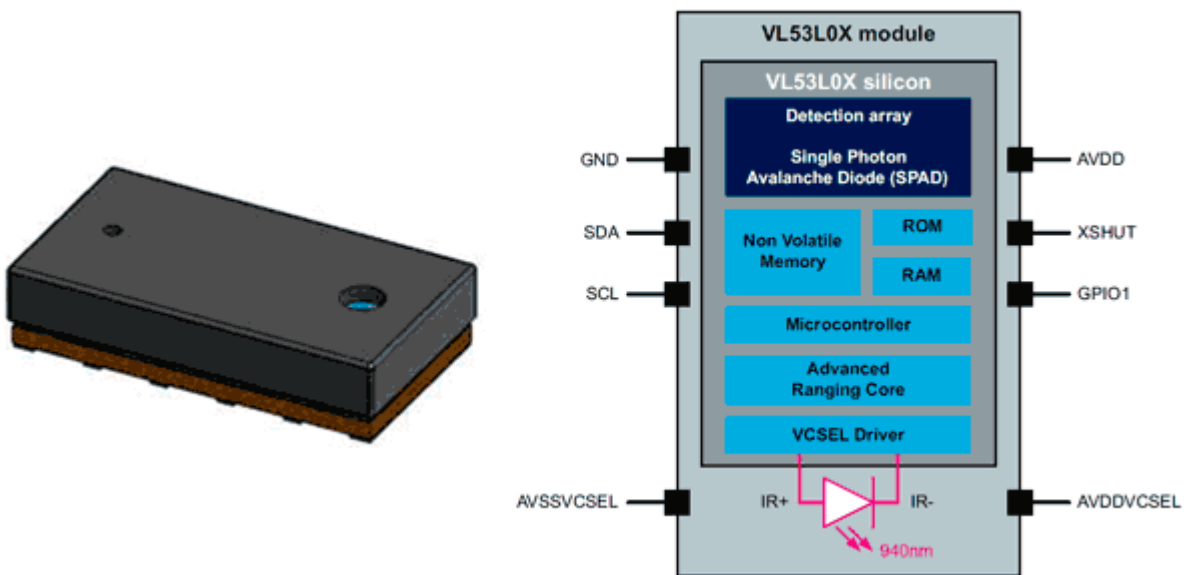


figura 16. VL53L0X

¿Cómo funciona?

En el VL53L0X posee un emisor laser de 940nm de longitud de onda y de tipo VCSEL (Vertical Cavity Surface-Emitting Laser) y el receptor es un SPAD (Single Photon Avalanche Diodes). Midiendo la distancia por medio de una electrónica interna (FlightSense™),

Posee 4 funciones diferentes:

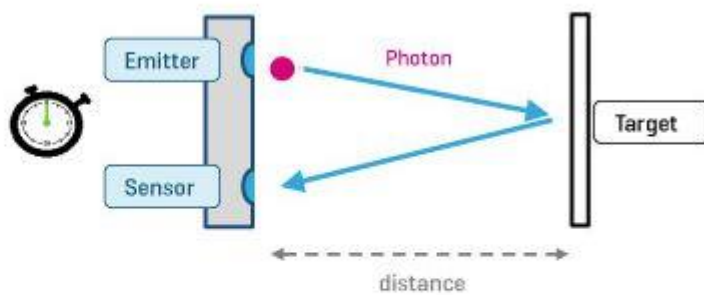
modo	timing	alcance	precision
defalut	30ms	1.2m	Siguiente tabla
Precisión alta	200ms	1.2m	+/-3%
Largo alcance	33ms	2m	Siguiente tabla
Alta velocidad	20ms	1.2m	+/-5%

Funcionamiento Default y largo alcance:

	Indoor			Outdoor		
Objeto	distancia	33ms	66ms	distancia	33ms	66ms
Blanco	120cm	4%	3%	60cm	7%	6%
gris	70cm	7%	6%	40cm	12%	9%

Especificaciones y características:

- Tensión de alimentación mínima: 5 Volts
- Voltaje lógico: 5 Volts
- Voltaje de Operación: 5 Volts – 35 Volts
- Posee dos entradas: una de 5 volts para controlar la parte lógica
- Alimentación para la salida a los motores: 5 volts o más
- Posee un regulador de voltaje de 5V 78M05, para alimentar la etapa lógica del L298N, sin embargo, cuando la alimentación supera los 12 V, se recomienda, utilizar una fuente de 5 V externa como fuente de alimentación
- Consumo de corriente digital: 0 a 36 mA
- Capacidad de corriente: 2A (picos de hasta 3A)
- Potencia máxima: 25 Watts
- Admite entradas de señal PWM para el control de velocidad
- Posee 8 diodos de protección contra corriente inversas
- Chip: L298N
- Canales: 2 (soporta 2 motores DC o 1 motor paso a paso)
- Peso: 30 gramos
- Dimensión: 43 x 43 mm
- Altura: 27 mm



$$\text{Measured distance} = \text{Photon travel time / 2} \times \text{Speed of light}$$

1cm round-trip at 67ps

figura 17. medición de distancia

Pruebas

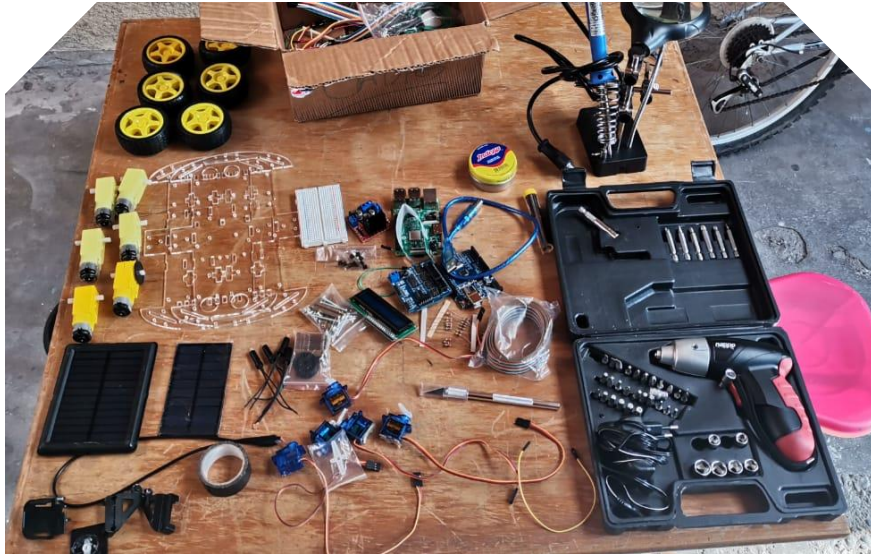


figura 18. Total de piezas del proyecto

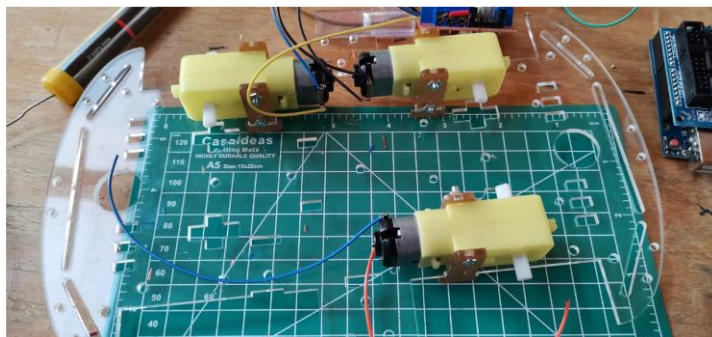


figura 19. ensamblaje motores DC

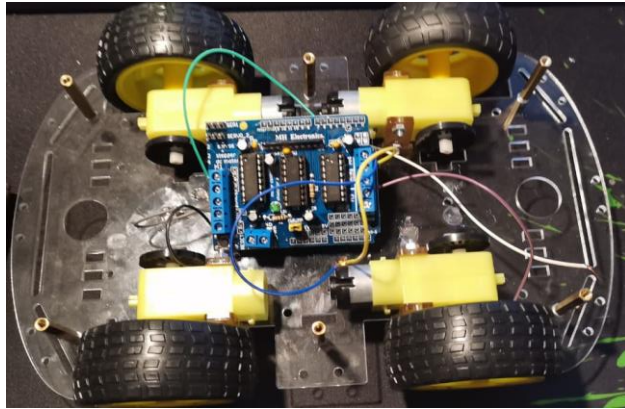


figura 20. ensamblaje Arduino

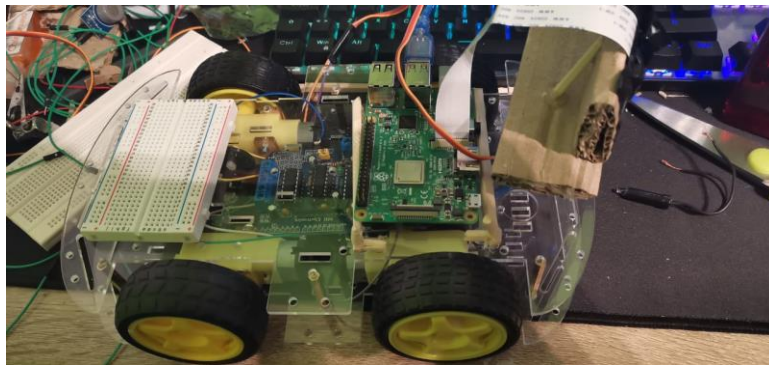


figura 21"ensamblaje cámara web y Raspbery"

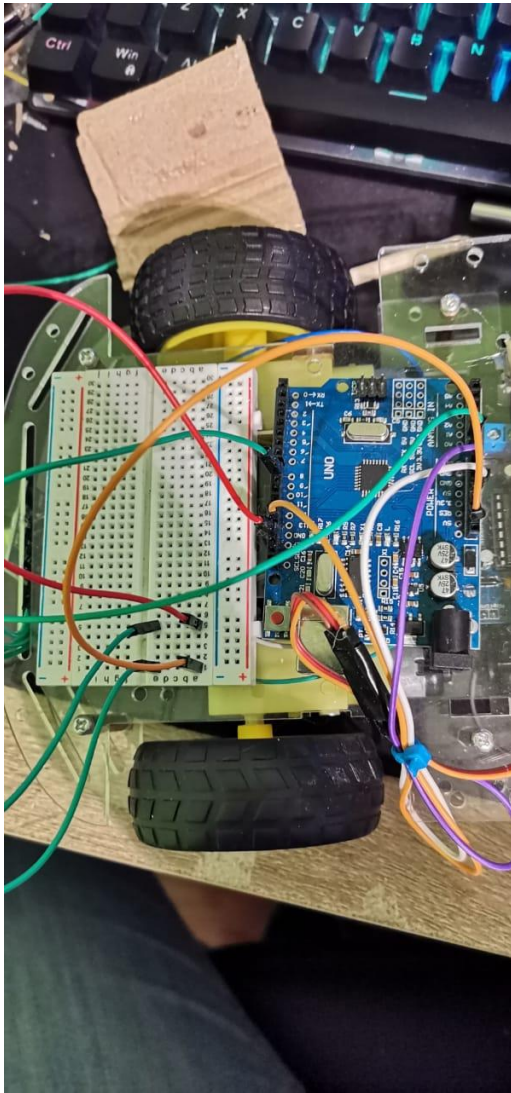


figura 22 conexion MQ9 y servo motor

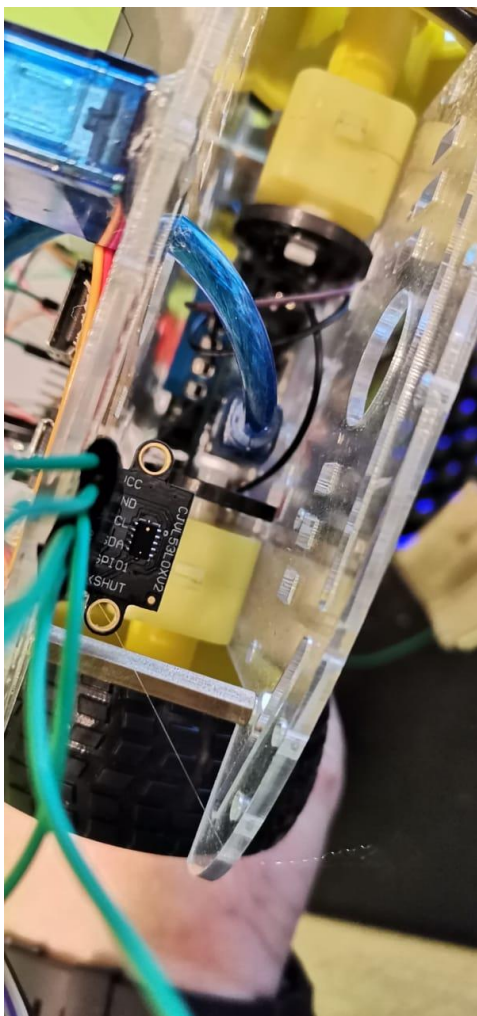


figura 23 instalacion sensor de vuelo

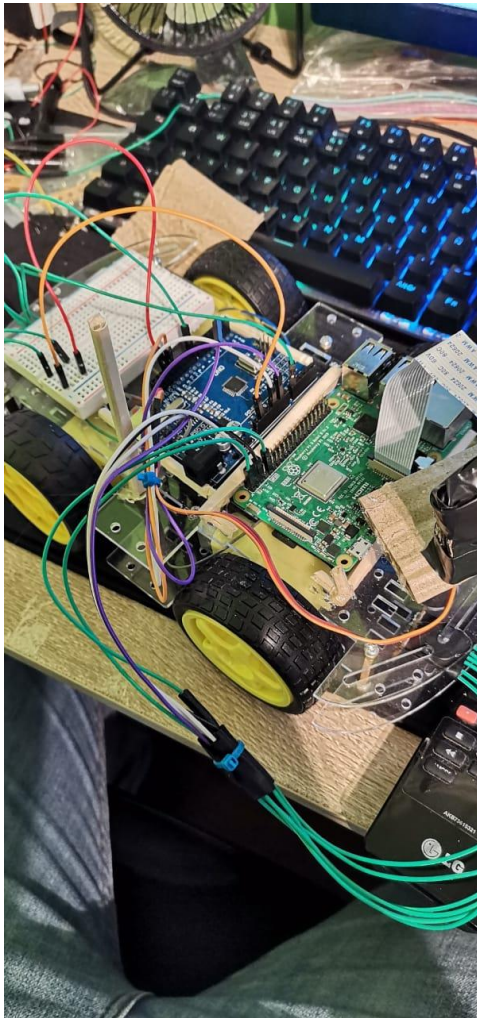


figura 24 conexion pines a raspberry sensor de vuelo



figura 25 vista lateral rover

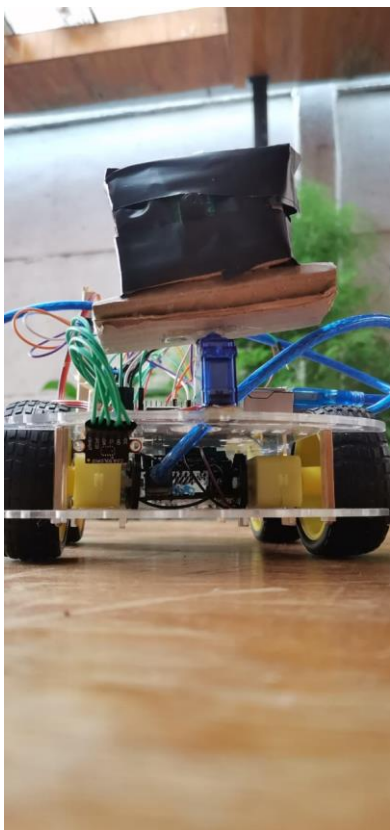


figura 26 vista rober frontal



figura 27 vista rover parte trasera

Diagrama de bloque

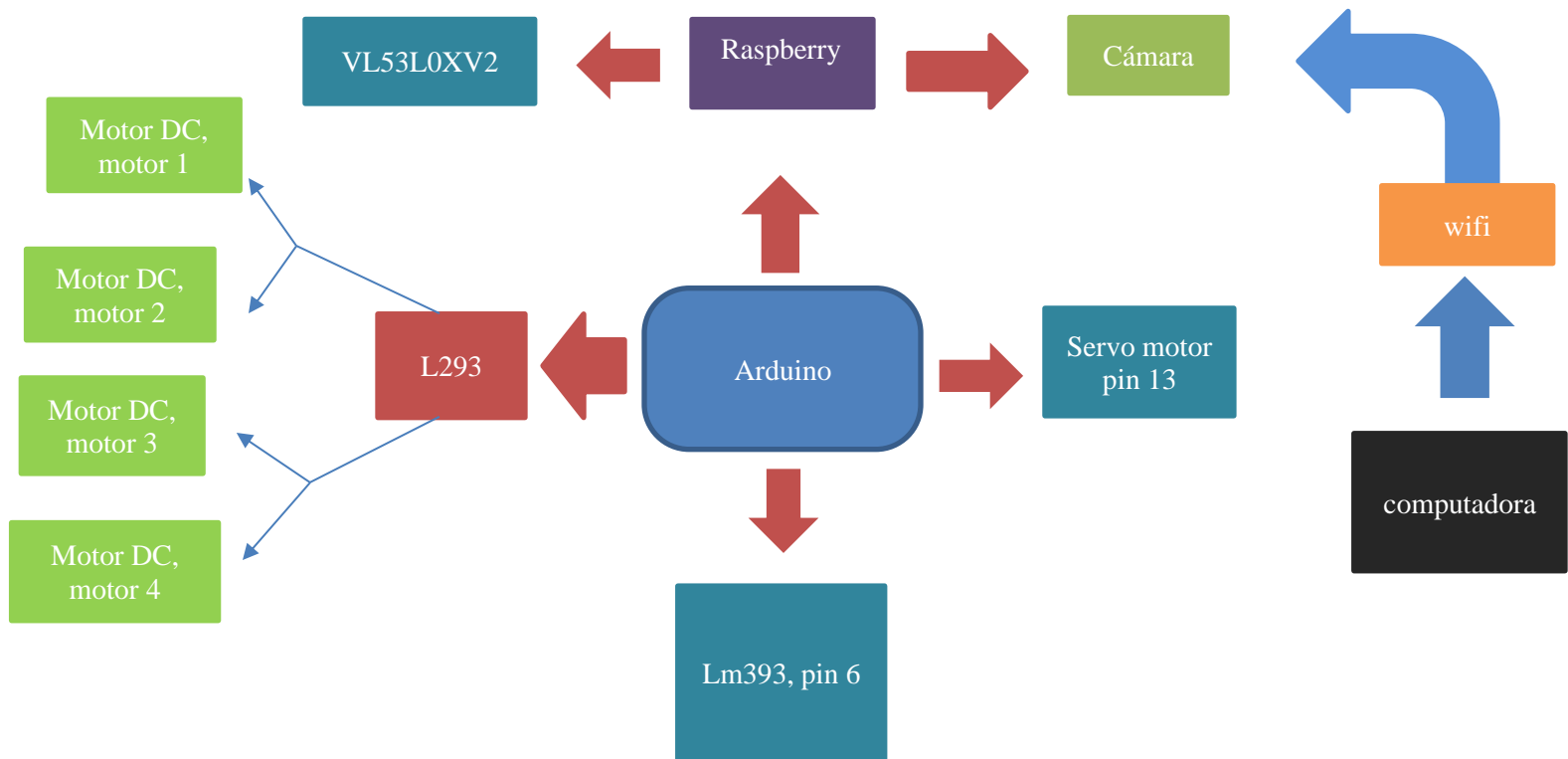


Diagrama de flujo

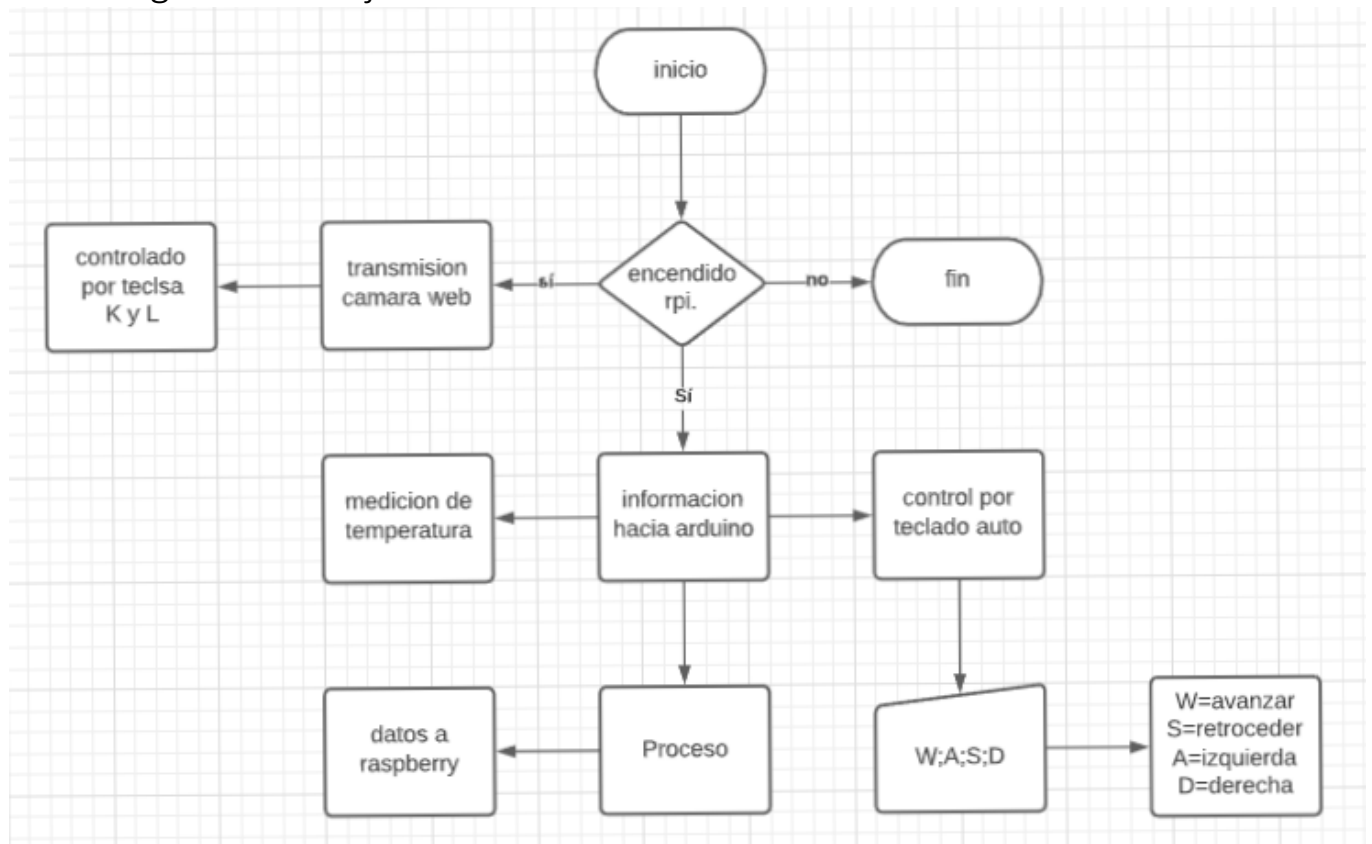


Figura 19 Diagrama de Flujo

Bibliografía

- [1] Correal, G. Pajares y J.J. Ruz. Febrero 2016 “Autonomy for groundlevel robotic space exploration: framework, simulation, architecture, algorithms and experiments”, Robotica, 34.
- [2] Ramirez, R. y Reyes, R. Abril 2015 “Diseño e implementación de un robot autónomo móvil usando tecnología FPGA” Universidad Politecnica Salesiana.
- [3] Tordesillas, J. 2015/2016 “Búsqueda y simulación de Locomocionde un robot Hexapodo para tareas de Búsqueda y rescate”. Universidad Politecnica Madrid
- [4] Bastian, A. 2017 “Sistema de asistencia en rescate mediante robot móvil” Universidad Andres Bello.
- [5] Rivas, J. 2014/2015 “Definición y análisis de los modos de marcha de un robot hexápodo para tareas de búsqueda y resctae” Universidad Politecnica de Madrid.
- [6] Hernandez, D. Londoño, D. y Ardilla, J. 2005 “Diseño y construcción de un vehículo autónomo tipo Rover” Universidad de San Buenaventura.
- [7] Perenne, S. 2019 “Como Hacer un Robot Coche | Robot todo terreno”

Anexos

Código Sensor_MQ9.ino

```
const int LED = 2;
const int DO = 8;

void setup() {
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
  pinMode(DO, INPUT);
}

void loop() {

  int alarm = 0;
  float sensor_volt;
  float RS_gas;
  float ratio;

  float R0 = 0.91;

  int sensorValue = analogRead(A0);
  sensor_volt = ((float)sensorValue / 1024) * 5.0;
  RS_gas = (5.0 - sensor_volt) / sensor_volt;

  ratio = RS_gas / R0; // ratio = RS/R0

  Serial.print("sensor_volt = ");
  Serial.println(sensor_volt);
  Serial.print("RS_ratio = ");
  Serial.println(RS_gas);
  Serial.print("Rs/R0 = ");
  Serial.println(ratio);

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

  alarm = digitalRead(DO);
  if (alarm == 1) digitalWrite(LED, HIGH);
  else if (alarm == 0) digitalWrite(LED, LOW);

  delay(1000);
}
```

Código Control_Ruedas_Rover.ino

```
AF_DCMotor motorC(3);
AF_DCMotor motorD(4);

void setup()
{
  motorA.setSpeed(250);
  motorB.setSpeed(250);
  motorC.setSpeed(250);
  motorD.setSpeed(250);
  Serial.begin(9600);
}

void loop()
{
  while(Serial.available()){
    char dato_in=(char)Serial.read();

    if(dato_in=='w' or 'W'){
      motorA.run(FORWARD);
      motorB.run(FORWARD);
      motorC.run(FORWARD);
      motorD.run(FORWARD);
      delay(5000);
    }

    if(dato_in=='s' or 'S'){
      motorA.run(BACKWARD);
      motorB.run(BACKWARD);
      motorC.run(BACKWARD);
      motorD.run(BACKWARD);
      delay(5000);
    }

    if (dato_in=='A' or 'a'){
      motorA.run(RELEASE);
      motorB.run(FORWARD);
      motorC.run(FORWARD);
      motorD.run(RELEASE);
      delay(5000);
    }

    if(dato_in=='d" or 'D'){
      motorA.run(FORWARD);
      motorB.run(BACKWARD);
      motorC.run(BACKWARD);
      motorD.run(FORWARD);
      delay(5000);
    }
  }
}
```

```
tempC = (5.0 * tempC *
100.0)/1024.0;
Serial.println((byte)tempC);
delay(1000);
}
```

```
    Código LM393.ino
float tempC;
int tempPin = 6;

void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
}

void loop() {
    // put your main code here, to run repeatedly:
    tempC = analogRead(tempPin);
}
```

Codigo Camera_pi.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# camera_pi.py
import time
import io
import threading
import picamera

class Camera(object):
    thread = None # background thread that reads frames from camera
    frame = None # current frame is stored here by background thread
    last_access = 0 # time of last client access to the camera

    def initialize(self):
        if Camera.thread is None:
            # start background frame thread
            Camera.thread = threading.Thread(target=self._thread)
            Camera.thread.start()

            # wait until frames start to be available
            while self.frame is None:
                time.sleep(0)

    def get_frame(self):
        Camera.last_access = time.time()
        self.initialize()
        return self.frame

    @classmethod
    def _thread(cls):
        with picamera.PiCamera() as camera:
            # camera setup
            camera.resolution = (320, 240)
            camera.hflip = True
            camera.vflip = True

            # let camera warm up
            camera.start_preview()
            time.sleep(2)

            stream = io.BytesIO()
            for foo in camera.capture_continuous(stream, 'jpeg',
                                                use_video_port=True):
                # store frame
                stream.seek(0)
                cls.frame = stream.read()

                # reset stream for next frame
                stream.seek(0)
                stream.truncate()

                # if there hasn't been any
                # clients asking for frames in
                # the last 10 seconds
                # stop the thread
                if time.time() -
                cls.last_access > 10:
                    break
                cls.thread = None
```



```

        código appCam.py
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#
#     appCam.py
#     based on tutorial ==> https://blog.miguelgrinberg.com/post/video-streaming-with-flask
#     PiCam Local Web Server with Flask

from flask import Flask, render_template, Response

# Raspberry Pi camera module (requires picamera package)
from camera_pi import Camera

app = Flask(__name__)

@app.route('/')
def index():
    """Video streaming home page."""
    return render_template('rover.html')

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

@app.route('/video_feed')
def video_feed():
    """Video streaming route. Put this in the src attribute of an img tag."""
    return Response(gen(Camera()),
                    mimetype='multipart/x-mixed-replace; boundary=frame')

if __name__ == '__main__':
    app.run(host='192.168.0.18', port=80, debug=True, threaded=True)
# if there hasn't been any clients asking for frames in
# the last 10 seconds stop the thread
if time.time() - cls.last_access > 10:
    break
cls.thread = None

```

código Control_direccion_Camara.ino
Código vl53l0x.ino

```
#include "Adafruit_VL53L0X.h"

Adafruit_VL53L0X lox = Adafruit_VL53L0X();

void setup() {
  Serial.begin(9600);
  while (!Serial) {
    delay(1);
  }

  Serial.println("Adafruit VL53L0X test");
  if (!lox.begin()) {
    Serial.println(F("Failed to boot VL53L0X"));
    while(1);
  }
  Serial.println(F("VL53L0X API Simple Ranging example\n\n"));
}

void loop() {
  VL53L0X_RangingMeasurementData_t measure;

  Serial.print("Reading a measurement... ");
  lox.rangingTest(&measure, false);

  if (measure.RangeStatus != 4) {
    Serial.print("Distance (mm): "); Serial.println(measure.RangeMilliMeter);
  } else {
    Serial.println(" out of range ");
  }

  delay(100);
}
```