Universidad Mariano Gálvez – Boca del Monte

Ingeniería en sistemas

Inteligencia Artificial

Inge. Henrry Sontay

Proyecto Carro autónomo con Raspeberry Pi y OpenCV

Gari Lester Mendoza Bedoya 7690-17-14228

Selvin Omar Castellanos 7690-17-14269

José Pablo Araiz 7690-17-9477

Víctor Guillermo Flores Dubón 7690-17-13943

Rony Alberto Cux Vásquez 7690-17-889

**Librerías utilizadas**

**Librerías de Python**

**-threading**

Esta librería permite que se puedan ejecutar varios procesos al mismo tiempo, a cada flujo de ejecución que se origina durante el procesamiento se le denomina hilo o subproceso.

Para cada hilo de un proceso existe un puntero que realizar el seguimiento de instrucciones que se ejecutan en cada momento.

**-socketserver**

Esta librería es la encardada de crear los servidores de red. Define clases para manejar solicitudes de red síncronas. La responsabilidad de procesar una solicitud se divide entre una clase servidor y una clase manejador de solicitudes.

**- socket**

Esta librería permite crear un canal de comunicación bidireccional. Los sockets se pueden comunicar dentro de un proceso, entre procesos dentro de la misma maquina o entre procesos de máquinas diferentes. Provee una interfaz para la API de los sockets Berkeley.

**- cv2**

Esta librería es la encargada de la visión de la computadora para la detección de objetos, una imagen no es mas que una matriz estándar de Numpy que contiene pixeles de puntos de datos. Cuanto mayor será el numero de pixeles en una imagen, mejor será su resolución.

Para ser procesado por una computadora, una imagen debe de convertirse en una forma binaria.

**- numpy**

Esta librería es un paquete de Python que significa “Numerical Python” es la librería principal para la informática científica, proporciona potentes estructuras de datos, implementación de matrices y matrices multidimensionales. Estas estructuras de datos garantizan cálculos eficientes con matrices.

**- math**

Esta librería proporciona acceso a las funciones matemáticas definidas en el estándar de C. Estas funciones no pueden ser usadas con numero s complejos y usa las funciones con el mismo nombre del módulo cmath.

**- time**

Esta librería proporciona varias funciones relacionadas con el tiempo. Aunque este módulo siempre está disponible, no totas las funciones están disponibles en todas las plataformas, en la mayoría de las funciones definidas en este modulo llaman a la plataforma de biblioteca C funciones con el mismo nombre.

**Librerías usadas en Raspberry**

**- RPi.GPIO**

Esta librería puede usar la numeración fisca del pin (Board) o la numeración Broadcom GPIO (BCM). Ambas bibliotecas están preinstaladas en los sistemas operativos Raspbian. nomenclatura de los pines GPIO, hay varias maneras de identificarlos; el método más simple y obvio es definitivamente el que se refiere a su ubicación física.

**-time**

Esta librería proporciona varias funciones relacionadas con el tiempo. Aunque este módulo siempre está disponible, no totas las funciones están disponibles en todas las plataformas, en la mayoría de las funciones definidas en este módulo llaman a la plataforma de biblioteca C funciones con el mismo nombre.

**-io**

Esta librería implementa las clases detrás de la función Open() incorporada del intérprete para operaciones de entrada y salida basada en archivos. Las clases se descomponen de tal manera que pueden ser recombinadas para fines alternativos.

**-socket**

Esta librería permite crear un canal de comunicación bidireccional. Los sockets se pueden comunicar dentro de un proceso, entre procesos dentro de la misma maquina o entre procesos de máquinas diferentes. Provee una interfaz para la API de los sockets Berkeley.

**- struct**

Esta librería lo que realiza es crear una estructura. El identificador ejemplo es el nombre de la estructura. Las variables declaradas dentro de las llaves de la definición de estructura son los miembros de la estructura. Los miembros de la misma estructura deben tener nombres únicos mientras que dos estructuras diferentes pueden tener miembros con el mismo nombre

**- picamera**

Esta librería funciona para el uso de la cámara de Raspberry Pi, el objetivo de esta librería es utilizar código sencillo y legible, sin necesidad de hacer llamadas a líneas de comando.

**- threading**

Esta librería hace posible la programación con hilos, los hilos son en ejecución de un proceso que comparten el mismo espacio de datos que el hilo principal y pueden, por tanto, tener acceso a la misma información o comunicarse entre sí más fácilmente que si estuvieran separados.

**Descripción de instalación de librerías y configuraciones.**

* Se instalo python 3 tanto en la Raspberry como en la computadora.
* Se instalo y configuro OpenCV tanto en la Raspberry (cliente) y computadora (servidor)
* En la Raspberry se habilitaron los puertos GPIO y el puerto de la cámara.
* En la computadora fue necesario configurar el firewall para permitir la comunicación en los puertos donde se comunicaban entres si el cliente y el servidor.

**Datos de entramiento y datos pruebas utilizados para la red neuronal**

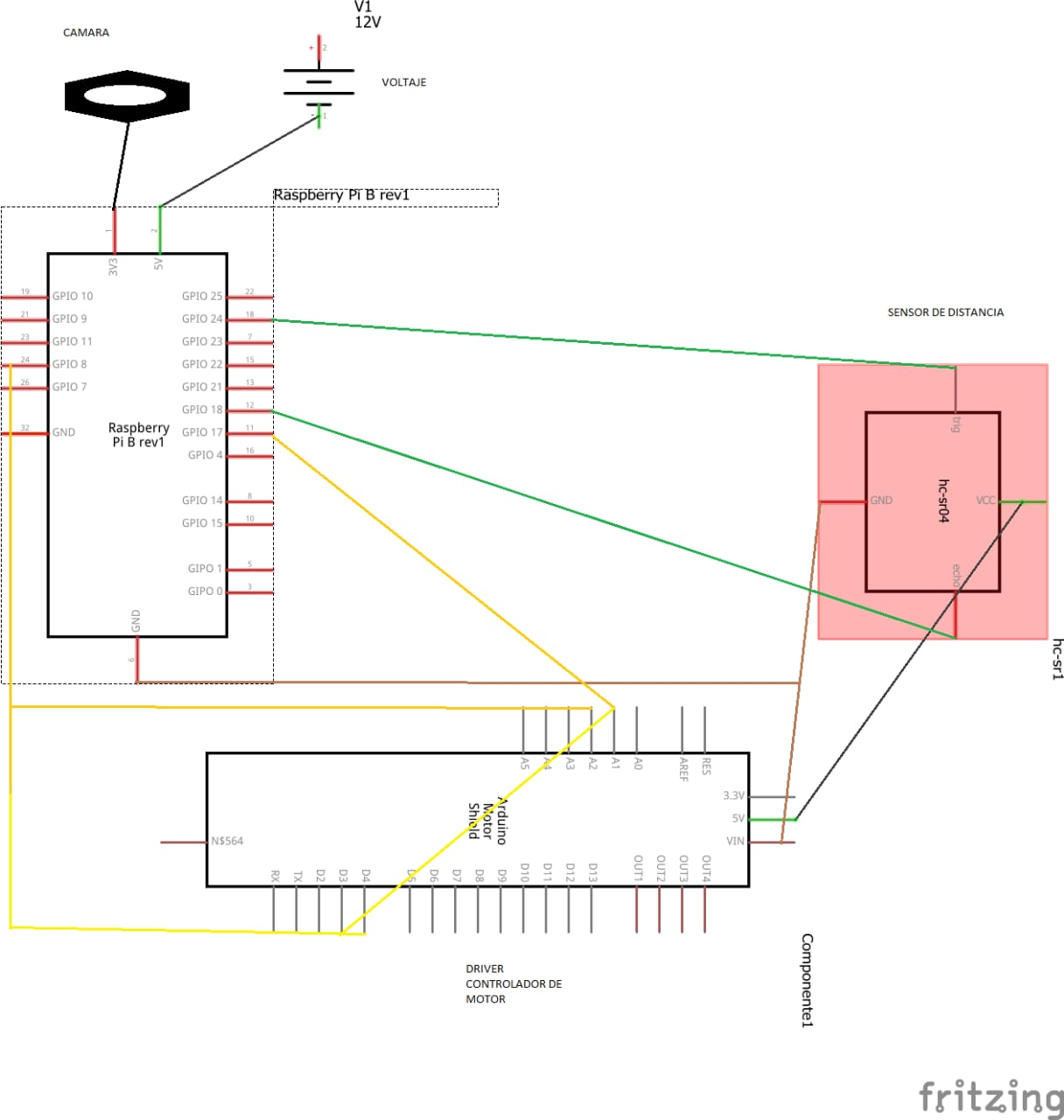
<https://github.com/victorFloresDubon/CarroAutonomo/tree/master/scripts/training_data>

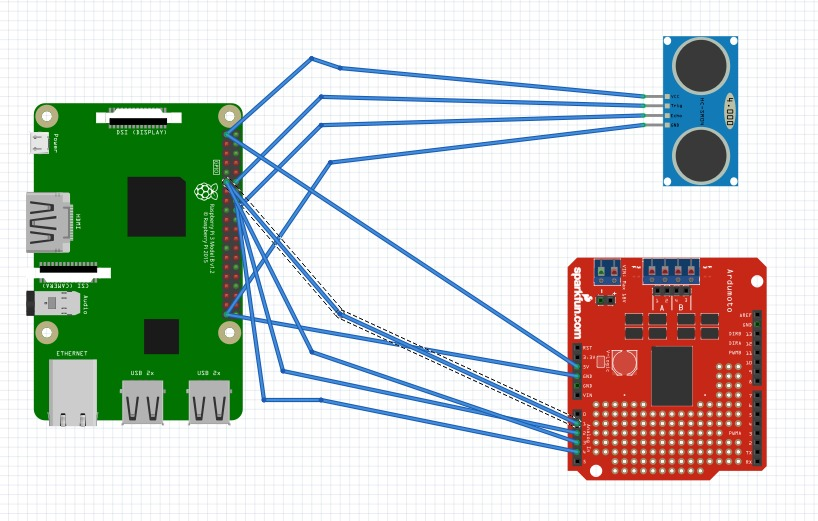
<https://github.com/victorFloresDubon/CarroAutonomo/tree/master/scripts/training_images>

**Descripción de Materiales utilizados**

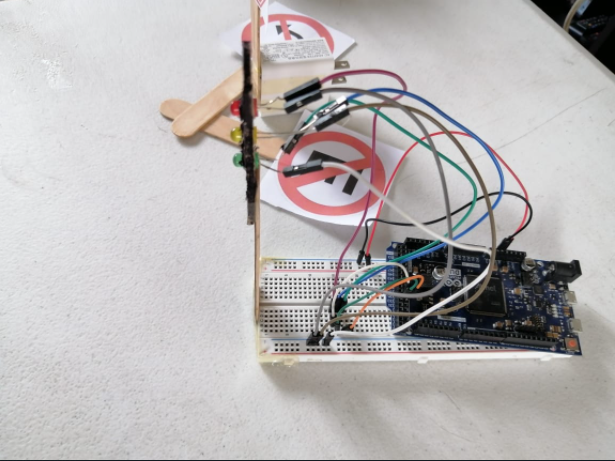
* Ultrasónico (HC-SR04) - para medir la distancia y evitar colisiones con objetos delante del carrito.
* Controlador Motores DC (L298N) – para manejar el movimiento de los motores.
* Motores DC – Se utilizaron 2 para mover el carrito.
* Raspberry Pi Camera Rev 1.3 – Cámara para obtener el video y realizar el reconocimiento de las imágenes.
* Raspberry Pi 3B+
* Arduino Due – Se utilizo para hacer el semáforo y pudiera tener el cambio en las luces.
* Chasis de Carro – Se utilizo un chasis para carro con 2 motores y una llantita.

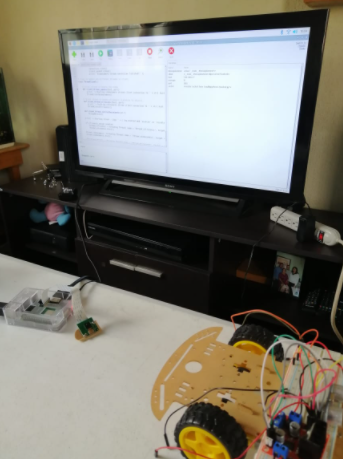
**Diagrama del circuito utilizado**

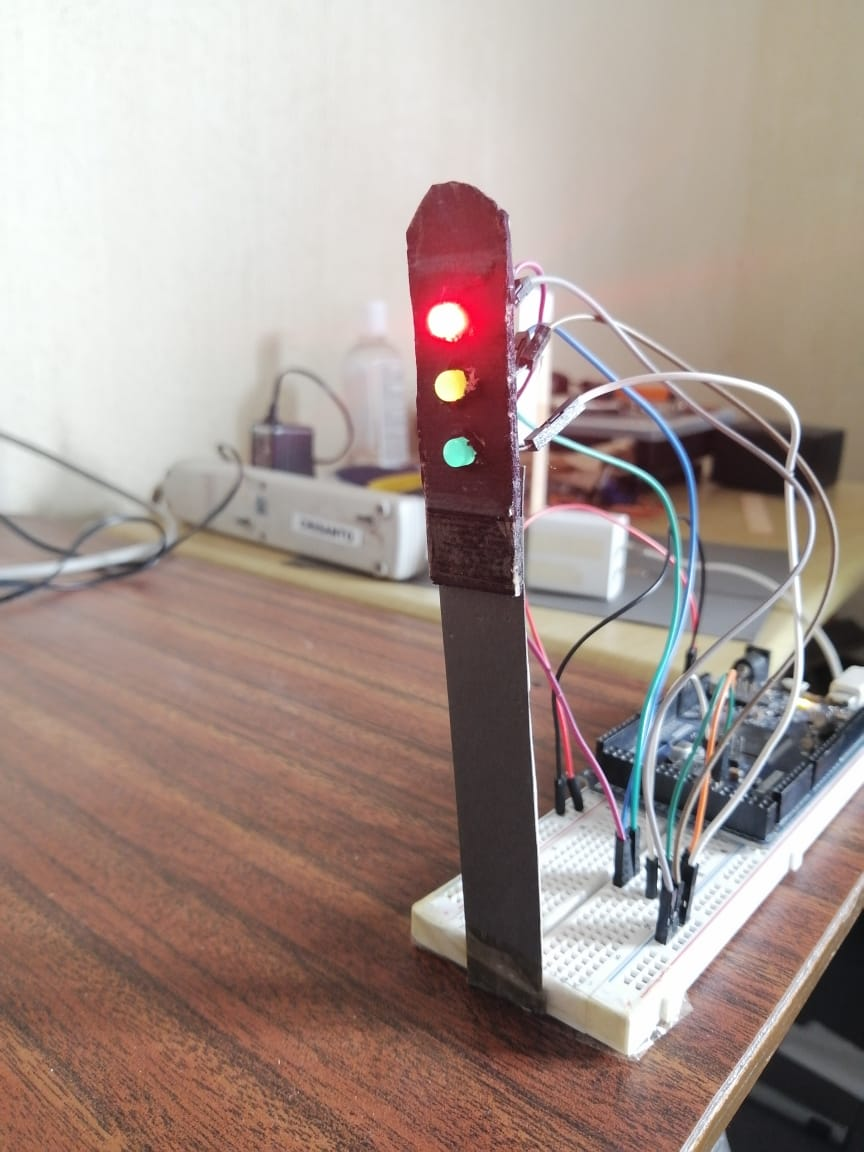
****



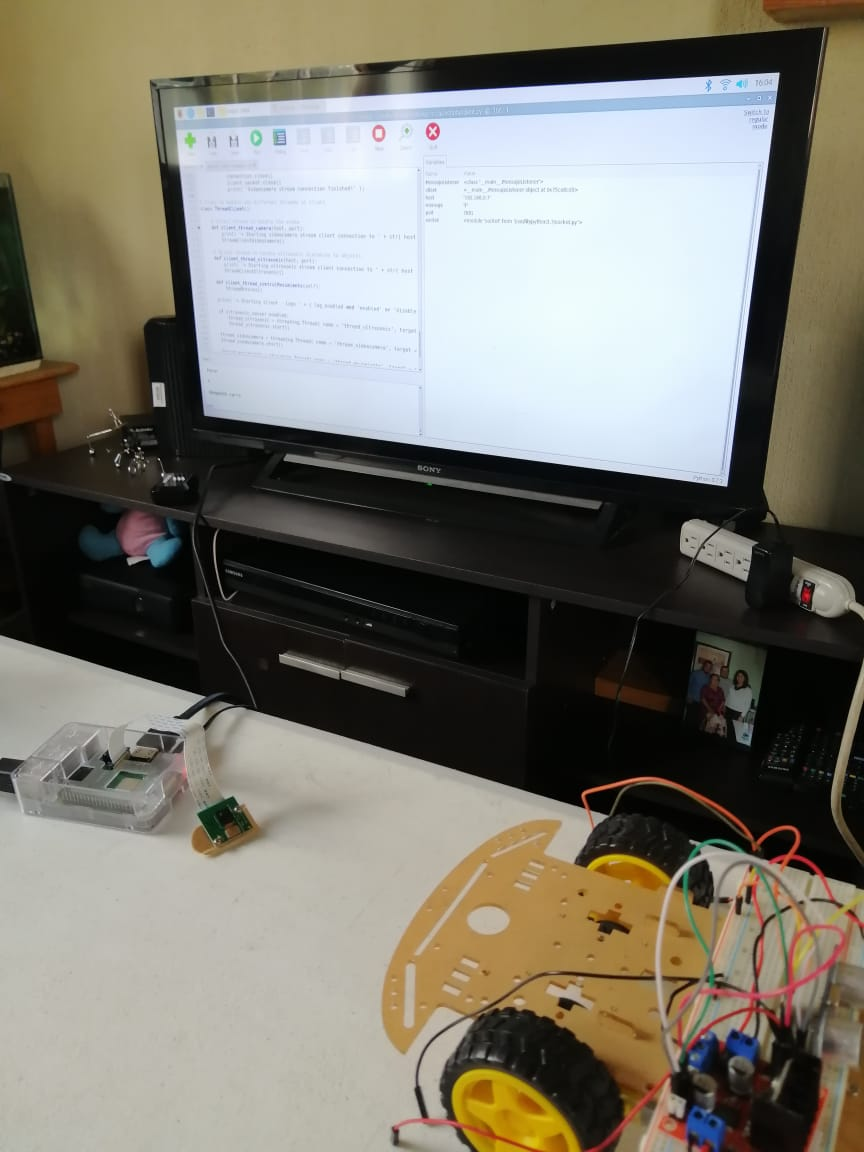
**Fotografías del proyecto paso a paso**

****

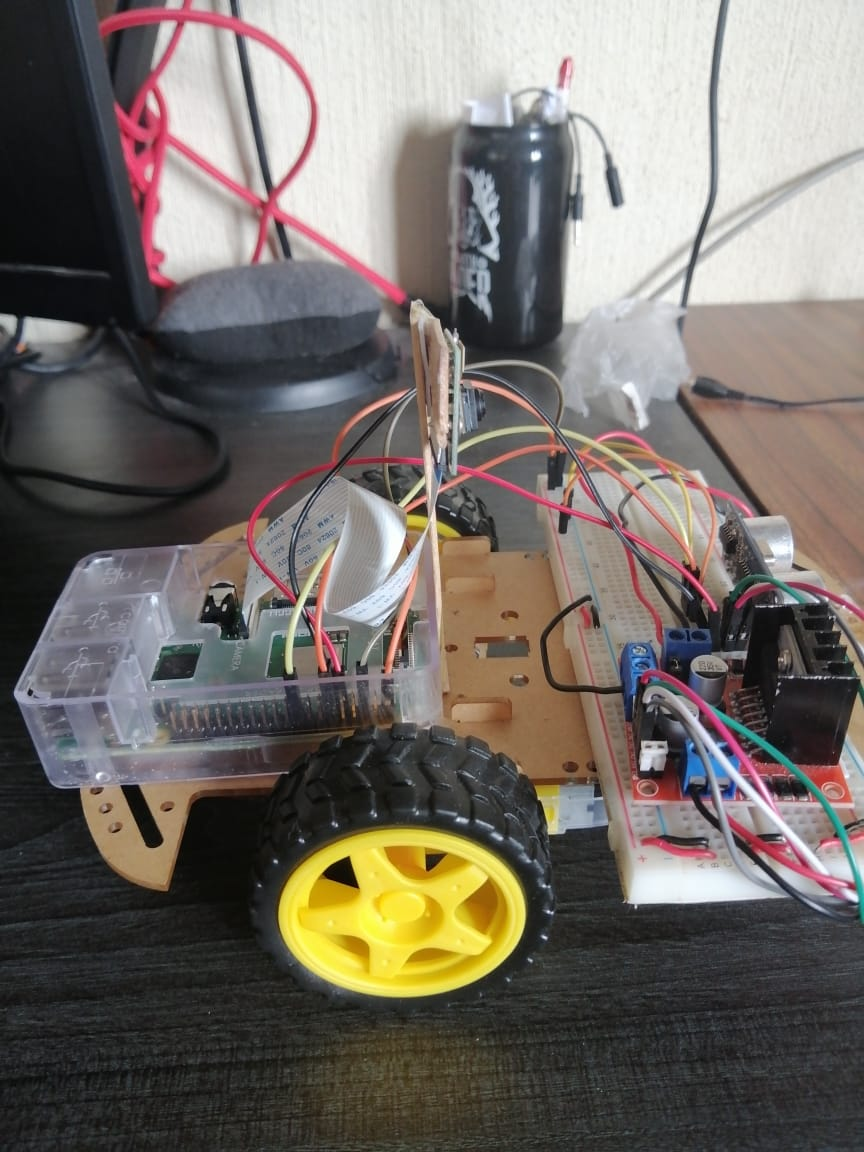


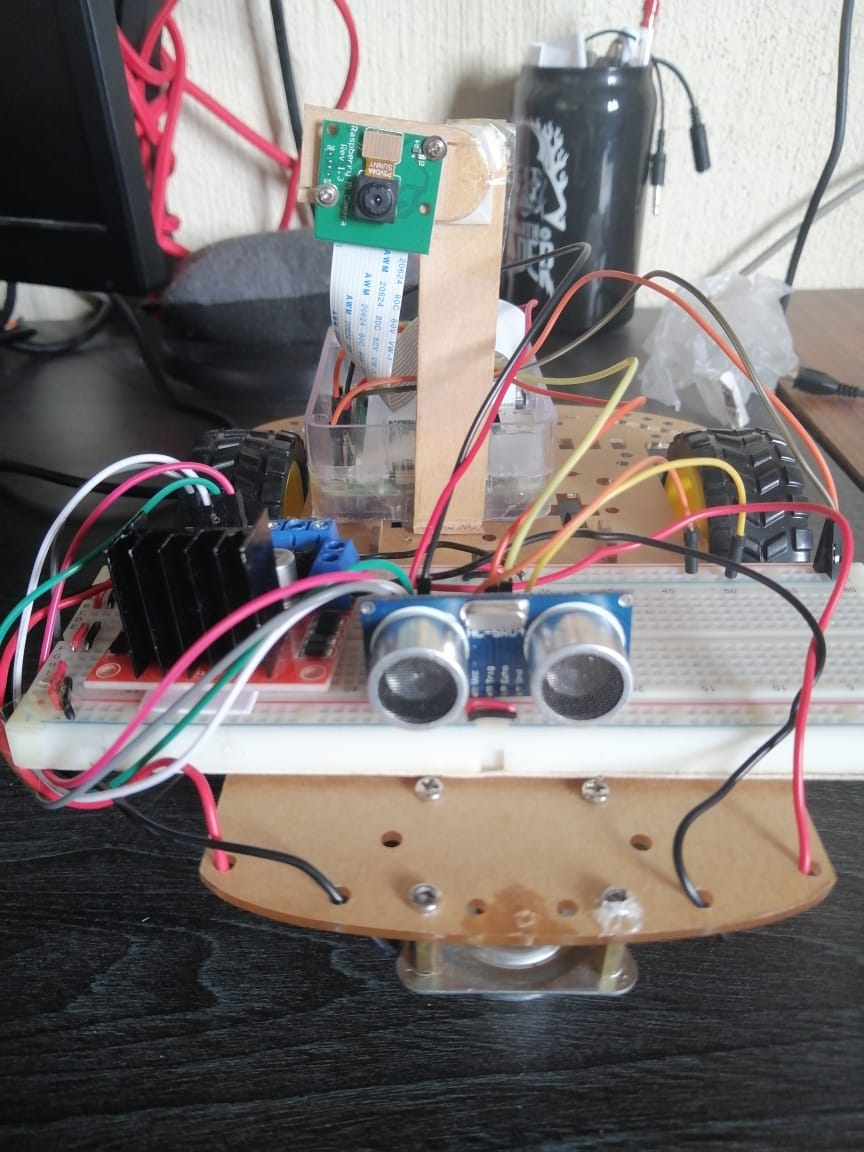












**Código publicado en GitHub**

<https://github.com/victorFloresDubon/CarroAutonomo>

**Video en youtube**

<https://www.youtube.com/watch?v=J9SpnhGPc40>