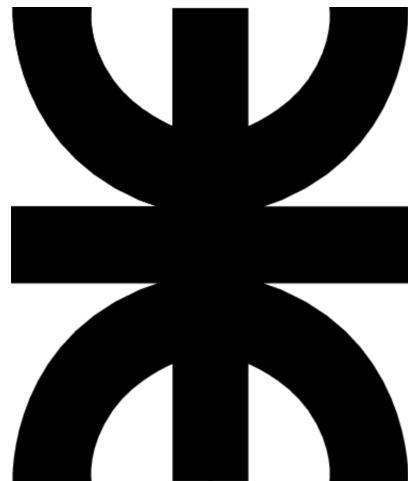


Universidad Tecnológica Nacional

Facultad Regional Resistencia



Tecnicatura Universitaria en Mecatrónica

Asignatura: Pasantías en Entes Oficiales o Empresas

Responsable de cátedra: Ing. Matías Ibarra Aranda

Proyecto: Reconocimiento automático de patentes (LPR)

Integrantes:

- Nardoni, Enzo Nahuel
- Sabo, Carlos Leandro Gabriel

2022

Índice

Índice	2
1. Introducción	3
2. Metodología	5
3. Ensayos	14
4. Resultados	17
5. Conclusiones	23
6. Anexos	26
1) Raspberry Pi	26
2) Puente H	30
3) Sensor ultrasónico	36
4) Cámara Digital	41
7. Referencias	48

1. Introducción

Se pretende desarrollar un sistema que permita o rechace el pase de diferentes vehículos en una entrada a un establecimiento sin intervención humana. El desarrollo de esta práctica tiene como finalidad diseñar un prototipo para dicho sistema, a fin de poder efectuar variadas pruebas para determinar el ajuste y las prestaciones del mismo, evaluando su desempeño y tiempo de respuesta. También esto permitirá una mejor estimación de los costos de producción y el tiempo de fabricación. Asimismo, con las pruebas establecidas se podría lograr un mayor convencimiento para su posterior implementación del sistema real (demostrando los resultados obtenidos).

Este control de acceso en principio estaría dotado de un mecanismo (como una barrera) que impida el pase de un vehículo al establecimiento, y si el vehículo posee autorización de ingreso, se accionará el mecanismo para permitir el pase del mismo.

Para determinar si un móvil puede ingresar, se prevé la lectura de la patente del mismo a través de una cámara y se realizaría un reconocimiento a través de inteligencia artificial. Para esto, se hará uso de machine-learning, donde se creará una red neuronal que se pondrá a entrenar para la tarea de reconocimiento. Luego,

el número de la patente sería contrastado contra una lista blanca a través de un software para saber si posee autorización para acceder a la playa de estacionamiento. En una mejora posterior del programa, se implementarían las funcionalidades de turnos de ingreso por personal y por áreas (restringiendo a franjas horarias específicas).

2. Metodología

Durante el desarrollo del proyecto, se han presentado diferentes problemáticas, las cuales se irán presentando a continuación.

Una de las primeras cuestiones a resolver era el lenguaje de programación a elegir para la generación del algoritmo que reconozca las patentes. Luego de un análisis sobre varios lenguajes, se tomó la decisión de elegir a Python, ya que el mismo cuenta con muchas librerías que resultarían sumamente útiles para el tratamiento de imágenes.

Luego de ello, hubo que buscar las librerías acordes para el caso. Las dos más destacables son OPENCV (que es una librería liberada por Intel, sumamente potente para aplicar transformaciones a las imágenes) y TESSERACT (liberada por Google, que permite realizar reconocimiento óptico de caracteres). También se han utilizado otras librerías como: SCIKIT-IMAGE, RPI.GPIO y NUMPY.

Una vez que ya se tenía elegido el lenguaje de programación más las librerías, llegó la hora de generar el algoritmo. Para ello, se hizo uso de una computadora portátil más una cámara web para ir realizando diferentes pruebas (comenzando desde tomar una fotografía y guardarla, hasta poder reconocer caracteres en esa imagen).

Ya cuando se contaba con el código de programación, se presentó el problema de qué soporte físico utilizar para ejecutar el código. Resulta que una PC no se podía ocupar (aún pensando en las mini PC's, que son compactas y pequeñas), ya que se dificultaría establecer una comunicación con entradas y salidas. Tampoco se podría utilizar una placa como Arduino o la ESP, ya que si bien tienen la facilidad para conectar entradas y salidas (estableciendo comunicación con sensores y actuadores respectivamente), no cuentan con el poder de procesamiento suficiente para correr las librerías usadas, que son un tanto pesadas. Así que la solución debía ser la combinación de ambas situaciones: una plataforma que facilite la conexión con sensores y actuadores, pero que también cuente con un buen poder de procesamiento. Aquí es cuando surge Raspberry Pi, que reunía esos requisitos. Se pudo conseguir prestada una de esas placas en la versión Raspberry Pi 3 modelo B+, contando con el puerto ethernet quemado luego de una tormenta eléctrica.

La placa que se ha conseguido prestada fue recibida sin la fuente de poder. Se sabía que se la podía alimentar con 5 voltios, así que se procedió utilizar un cargador de teléfono que se tenía a mano para probar que funcionara la placa. Para ello, se recurrió a instalar una imagen en la placa: la clásica de 32 bits para

ARM, que es Rasbian, un sistema operativo derivado de Debian (de ahí su nombre derivado de la conjunción de los vocablos «Raspberry» y «Debian»). Ahora bien, luego de probar varias tareas, se notó que el uso del procesador no llegaba siquiera al 50%, incluso con tareas pesadas como la reproducción de videos en alta definición (HD). Viendo las especificaciones, se determinó que la alimentación debía ser de 5 V con una corriente de entre 2,2 A a 3 A. Entonces se decidió ir a una tienda de venta de accesorios para teléfonos celulares para conseguir un transformador de esas características (pensando en alguno de carga rápida). Las opciones eran uno de 3,2A y otro de 2,4A. Para evitar que la placa se vea forzada, se optó por el de 2,4 amperios. Una vez que se pudo iniciar el sistema operativo en la placa, se volvió a probarla con varias tareas y nuevamente no se llegaba a ocupar un gran porcentaje el procesador (como mucho llegaba alrededor del 30%). Luego de pensar mucho en la problemática y revisando detalles del sistema operativo, se encontró una advertencia con un símbolo de un rayo, lo cual hizo pensar que se trataba de un problema de potencia. Ahora bien, ya se contaba con una fuente de poder que cumpliera con las especificaciones de la placa. Así que se probó con otro cable, uno proveniente de un teléfono celular BlackBerry, que aparentaba que podría estar en muy buenas condiciones. Sin embargo, el problema persistía. Ya en

esa instancia, se probó un cable muy corto (de tan solo 30 cm), total no había nada que perder. Para la sorpresa de todos, ese cable descartado de una casa de computación (que era para la recarga de algún dispositivo, ya que tiene la inscripción de «Genius») fue el que dio buenos resultados. Ya no aparecía más el warning de potencia y el procesador llegaba al tope en tareas exhaustivas, no solo superaba el 50% sino que fácilmente llegaba al 100% de a momentos (dependiendo de las tareas).

Ya una vez solucionado el inconveniente de la alimentación, se procedió a probar el algoritmo diseñado ya en el soporte físico elegido para el proyecto (la Raspberry Pi). Sin embargo, las pruebas de reconocimiento no se podían ejecutar en la placa. A la hora de ejecutar el algoritmo, aparecían diversos errores de las librerías. Luego de pensar en diferentes cuestiones, se llegó a sospechar que las librerías utilizadas no tenían un buen funcionamiento en arquitecturas de 32 bits. El algoritmo había sido diseñado y probado en una PC (con la arquitectura conocida como X86_64, también llamada AMD64). En la placa Raspberry Pi se estaba utilizando la clásica imagen de Raspbian de 32 bits (que era lo más confiable para asegurarse de que la placa funcionara). Se decidió entonces probar instalar una imagen que sea de 64 bits para la arquitectura ARMv8 (ya que desde esa versión,

los procesadores ARM soportan un conjunto de instrucciones de 64 bits), que es la que posee la placa. Fue entonces cuando se probó instalando la imagen oficial de Raspbian de 64 bits. Cuando se intentó cargar esa imagen, se lo hizo con la misma aplicación que ofrece la página oficial de Raspberry Pi (llamada Raspberry Pi Imager), para evitar mayores inconvenientes y acudir a lo seguro. No obstante, al intentar cargar la imagen, llegado al 99% fallaba y aparecía un mensaje de error de escritura en la tarjeta de memoria. Se siguió intentando otras veces más, llegando a pensar que la tarjeta SD podría tener fallas en ciertos sectores. También se probó cargando la imagen sin usar la aplicación específica para ese fin; haciéndolo de forma manual, a través de comandos desde una terminal. Al hacerlo de este modo, forzosamente quedaba escrita toda la imagen del sistema operativo. Pero cuando se intentó encender la placa, el sistema operativo no iniciaba. Entonces se pensó en otras alternativas de imágenes ofrecidas por la aplicación de Raspberry (que sean de 64 bits). Las imágenes de Raspbian (con sus variaciones) nos seguían generando el mismo error al llegar al 99%. Se probó con otra imagen que no sea Raspbian, en este caso con Kali Linux, cargándola con la aplicación Raspberry Pi Imager. Cuando se conectó a la alimentación la placa, el sistema operativo parecía iniciarse, pero no brindaba señal de video. Se procedió a cargar una imagen de Kali

Linux descargándola desde la página oficial de Kali, pero usando Raspberry Pi Imager. Con ello, al fin se obtuvo éxito en poder iniciar un sistema operativo de 64 bits. Habiendo llegado a esta instancia, se probó el algoritmo y ahora sí se ejecutaba sin problemas, lo cual evidenció que la sospecha que se tenía era correcta.

Con respecto a la falta de entrega de la señal de video que se tuvo antes, luego de varios apagados y nuevos arranques de la placa, fue percatado que para que la placa entregue señal de video previamente debe estar encendida la pantalla a la que se conecte, ya que si se la conecta después, directamente no proyecta imagen alguna. Sabiendo eso, ya no se tenía la incertidumbre de por qué a veces sí enviaba señal de video y otras veces no.

El algoritmo que se tenía hasta entonces permitía interpretar los caracteres alfanuméricos de una serie de imágenes que se tenían ya guardadas, e incluso se lograba hacer una captura con una cámara web con el posterior almacenamiento de la fotografía. Luego de esto, se escribió un nuevo algoritmo para probar un actuador y así simular el movimiento de una barrera. Para ello, se conectó un motor de 5 V con un puente H entre este y la placa. Se probó con líneas sencillas de programación sin llegar a accionar el motor. Se ensayaron distintas formas de

códigos sin obtener éxito. Se sospechaba que quizás la alimentación del puente H no daba abasto, ya que este necesita un voltaje de entre 3,5 y 5 voltios y se lo había conectado a un pin de 3,5 V. Entonces se intentó con un pin de 5 V, siguiendo con el mismo problema. También se conectó a una alimentación externa (que no provenga de la placa). Después de mucho tiempo, se llegó a la conclusión de que el puente H se encontraba quemado, luego de probar con uno distinto.

Ya habiendo solucionado el problema anterior, ahora el puente H indicaba su buen funcionamiento destellando una luz roja desde un led. Pero aún así, no se podía hacer girar el motor. Después de revisar varias veces el código y de probar con otros motores también, pasado un buen tiempo, se pudo hallar el problema. Había unos cables que no tenían continuidad. Así que se procedió en la adquisición de nuevos cables. Para colmo, queriendo probar con cables de red y soldando tramos con estaño, la pistola de soldar llegó al final de su vida útil.

Cuando todo parecía acorde para que funcionara el actuador, resulta que al querer establecer una llamada al código, no se podía ejecutar por una falta de permisos, que impedía utilizar la interfaz GPIO. Esto fue solucionado sencillamente ejecutando el comando como usuario root, que es el que posee los máximos privilegios en un sistema operativo GNU/Linux.

Llegando a esta etapa, se logró hacer funcionar varias cosas que fueron ensayadas por separado: la captura de una fotografía con su posterior guardado, el reconocimiento óptico de caracteres alfanuméricos, la lectura de distancia a través de un sensor de ultrasonido y el giro de un motor. A posteriori, se buscó la idea de apuntalar todo a un diseño modularizado, a fin de tener un buen orden y un mejor mantenimiento del software. En este punto ya se tenían diseñados cada uno de los módulos de software, pero había problemas de diversas índoles para ser llamados desde un módulo principal. Los inconvenientes iban desde las llamadas a procedimientos, paso de parámetros hasta la falta de un procedimiento principal en cada módulo.

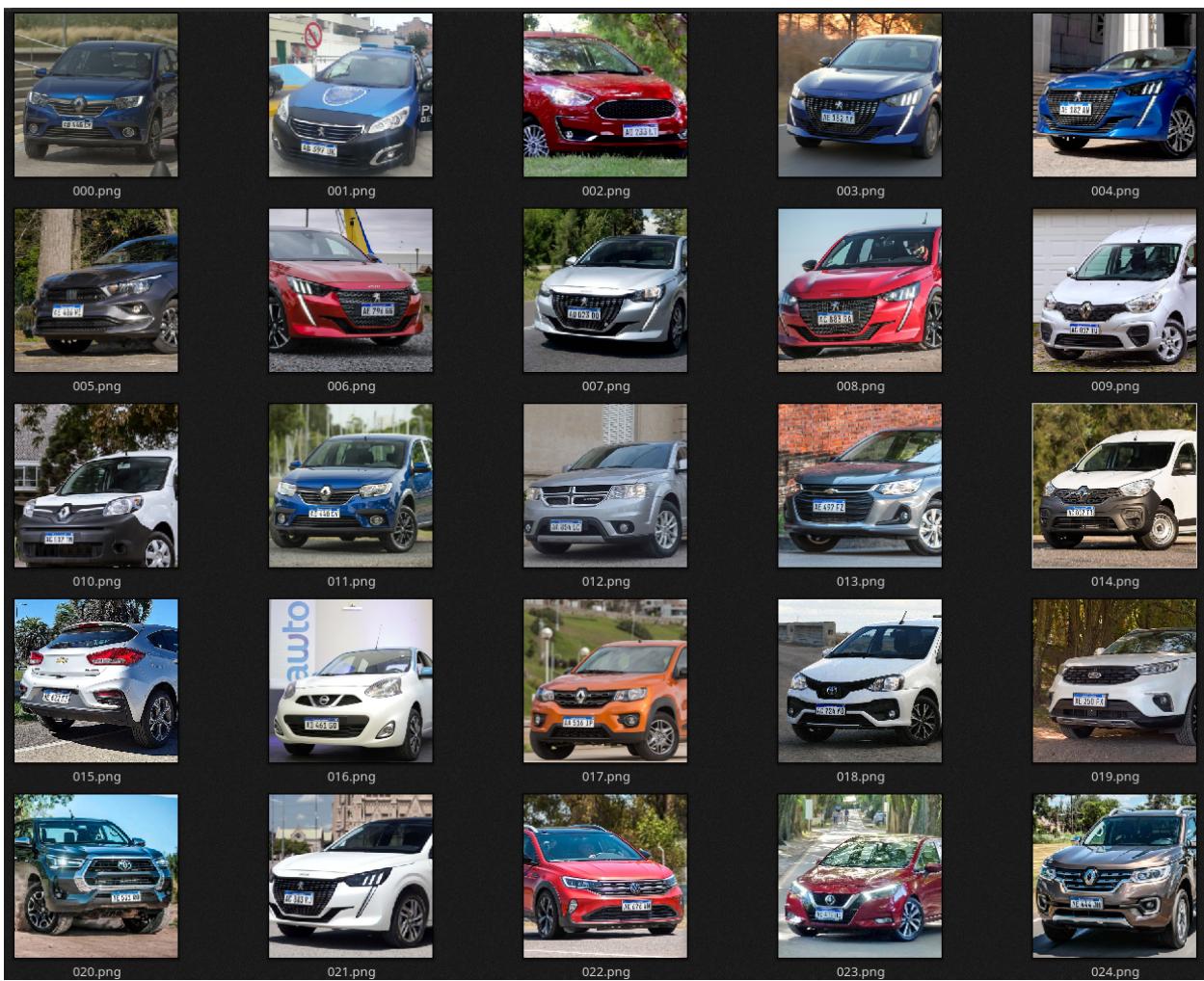
Hasta aquí se pudieron establecer todas las conexiones con la placa: el sensor de proximidad ultrasónico, el actuador de 5V y la cámara para las capturas, todo montado en una protoboard. Las pruebas resultaron exitosas y todo funcionaba como lo deseado. Más tarde, cuando se quiso trasladar ese conexionado a una placa de circuito impreso, comenzaron nuevamente los problemas. Ya el algoritmo no se ejecutaba por completo, debido a que en el punto inicial no se podía sensar la distancia con el sensor. Luego de mucho tiempo revisando, se detectó que al trasladar el diseño de la protoboard a la PCB, se olvidó de

implementar una resistencia de $2\text{ K}\Omega$ como se pretendía; en vez de eso, se colocó solo una resistencia de $1\text{ K}\Omega$. Esa confusión surgió debido a que para las pruebas con la protoboard, se habían instalado dos resistencias en serie de $1\text{ K}\Omega$, que eran con las que se contaba al alcance de la mano. Se insertó una resistencia más en la PCB y aún así el sistema seguía sin funcionar. También fue evidenciado que el puente H había dejado de responder. Se intentó desoldar el sensor de ultrasonido para reemplazarlo por el que se había usado en la protoboard (la idea era mantener el conexionado de la protoboard intacto, que era una configuración correcta y estable). En ese momento se quemó el soldador tipo lápiz, el cual era uno nuevo, ya que el anterior también se había quemado (pero aquel llevaba unos quince años de uso). Después algunas indagaciones, se arribó a que se deterioraron dos soldadores debido al mal estado de un toma de corriente.

Con un soldador nuevo en mano, se colocó la resistencia correcta (de $2\text{ K}\Omega$) reemplazando por el conexionado en serie. Además de ello, se volvió a revisar íntegramente el conexionado contrastando con los pines declarados en los distintos módulos de software escritos. Encontrándose diferencias, se volvieron a conectar como correspondía y nuevamente volvió a funcionar todo el sistema como se había previsto.

3. Ensayos

Los ensayos que se realizaron para probar el sistema diseñado fueron llevados a cabo sobre un total de veinticuatro diferentes imágenes de distintos automóviles que poseen la patente nueva, las cuales fueron recopiladas buscando en internet y pueden apreciarse a continuación.



Para conocer la fiabilidad de la detección, se desarrolló un testeo automático. A partir de esto, se pudo comprobar que al algoritmo se le dificulta la detección en algunos casos, reconociendo una matrícula incorrecta.

```
● (envs) lean@Multics:~/Proyectos/Mecatrónica/LPR/Endro$ python test.py
000 OK
001 OK
002 OK
003 OK
004 ERROR | Original: AE182AW Reconocido: AE182AH
005 ERROR | Original: AE486WE Reconocido: AE486HE
006 OK
007 OK
008 OK
009 OK
010 OK
011 OK
012 ERROR | Original: AA854LC Reconocido: AA854L0
013 OK
014 OK
015 OK
016 ERROR | Original: AD461GQ Reconocido: AD461GG
017 OK
018 OK
019 OK
020 ERROR | Original: AE521RQ Reconocido: AE521R0
021 ERROR | Original: AC883RJ Reconocido: AC883RI
022 ERROR | Original: AE676WN Reconocido: AE676HN
023 ERROR | Original: AE410HE Reconocido: AE420HE
024 OK
```

Se evidencia que se presta a confusión entre algunos caracteres: la «H» con la

«W», la «C» con la «O», la «Q» con la «G» y la «O», la «J» con la «I» y en un caso también confundió el dígito «1» con el «2».

En algunos de estos casos de prueba, los errores se debieron a la resolución de la fotografía, que no era de la mejor, sumado a la similitud de esos caracteres dada la tipografía que se usan en las patentes argentinas en la actualidad.

4. Resultados

Se logra obtener un sistema de reconocimiento óptico de matrículas de patentes, específicamente de las patentes nuevas (Patente Única del Mercosur) para vehículos argentinos (no para motocicletas).



En cuanto al diseño de esta patente, se puede describir lo siguiente:

Las características del diseño de la patente del Mercosur es igual para vehículos y motovehículos, salvo que para las motos es más pequeña.

Caracteres: está compuesta por 7 caracteres entre números y letras combinados. La secuencia y el espaciado entre caracteres lo decidirá cada estado.

Tipografía: caja de caracteres de 65 mm de alto.

Color de fondo: blanco.

Color de caracteres: para vehículos particulares será NEGRO, para vehículos comerciales será de color ROJO, para vehículos oficiales el color será AZUL, y para los vehículos diplomáticos el color será DORADO. También habrá otro tipo de patentes para el Mercosur como para los vehículos de uso especial que tendrán el color VERDE y para los vehículos antiguos o clásicos el color será GRIS PLATA.

Franja Azul: la patente del Mercosur tiene una franja de color azul de un grosor de 30mm. de alto y 396mm. de ancho. La misma está ubicada en la parte superior de la patente.

Identificación del país: el nombre del país se ubica sobre la franja azul.

Tamaño de la patente: el tamaño de la patente del Mercosur será 400mm. de ancho y 130mm. de alto.

Emblema del Mercosur: con un tamaño de 25mm. de ancho y 20mm. de alto ubicado en la parte superior izquierda de la patente.

Bandera de cada país: ubicada en la parte superior derecha de la patente.

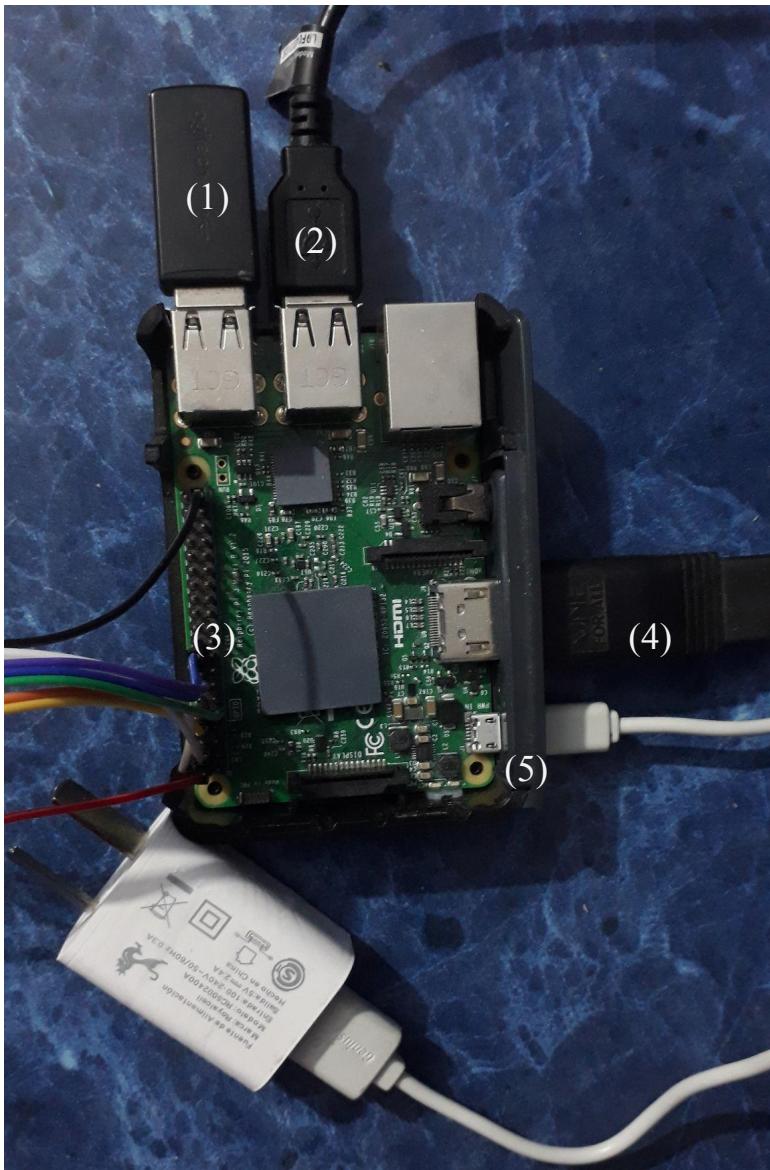
Material de la patente del Mercosur: el material es de aluminio con un grosor de 1mm.

Gracias al uso de la placa Raspberry Pi, se cuenta con un diseño compacto, de bajo consumo y fácilmente reemplazable por otras placas de similares características. Su interfaz GPIO permite una interconexión sencilla y un buen manejo de interrupciones para que el sistema operativo pueda administrarlas.

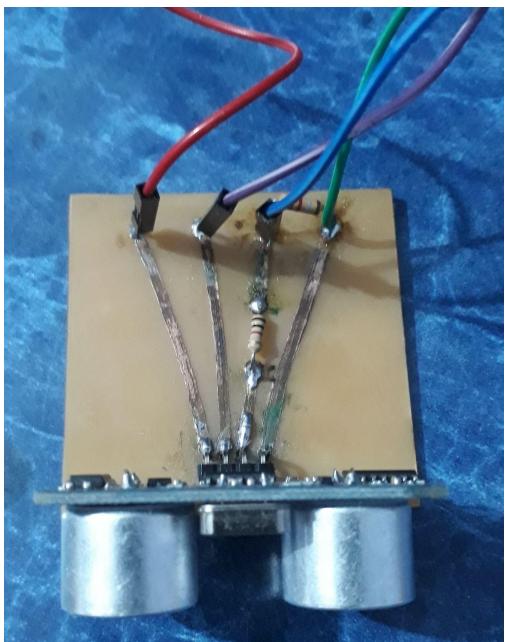


		Pin no.		
DC Power	3.3V	1	2	5V
SDA1, I ² C	GPIO 2	3	4	5V
SCL1, I ² C	GPIO 3	5	6	GND
GPIO_GCLK	GPIO 4	7	8	GPIO 14
	GND	9	10	GPIO 15
GPIO_GEN0	GPIO 17	11	12	GPIO 18
GPIO_GEN2	GPIO 27	13	14	GND
GPIO_GEN3	GPIO 22	15	16	GPIO 23
DC Power	3.3V	17	18	GPIO 24
SPI_MOSI	GPIO 10	19	20	GND
SPI_MISO	GPIO 9	21	22	GPIO 25
SPI_CLK	GPIO 11	23	24	GPIO 8
I ² C ID EEPROM	GND	25	26	GPIO 7
	DNC	27	28	DNC
	GPIO 5	29	30	GND
	GPIO 6	31	32	GPIO 12
	GPIO 13	33	34	GND
	GPIO 19	35	36	GPIO 16
	GPIO 26	37	38	GPIO 20
	GND	39	40	GPIO 21

En la siguiente imagen se puede apreciar la Raspberry Pi utilizada con las diferentes conexiones.

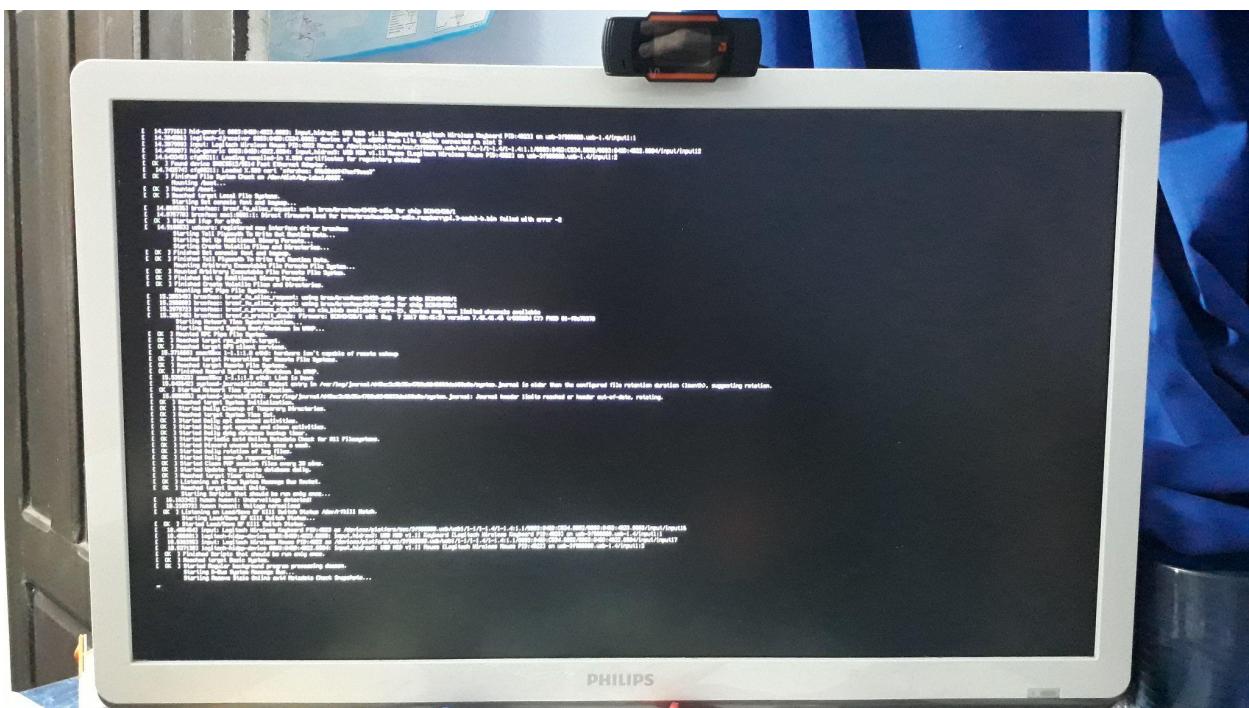


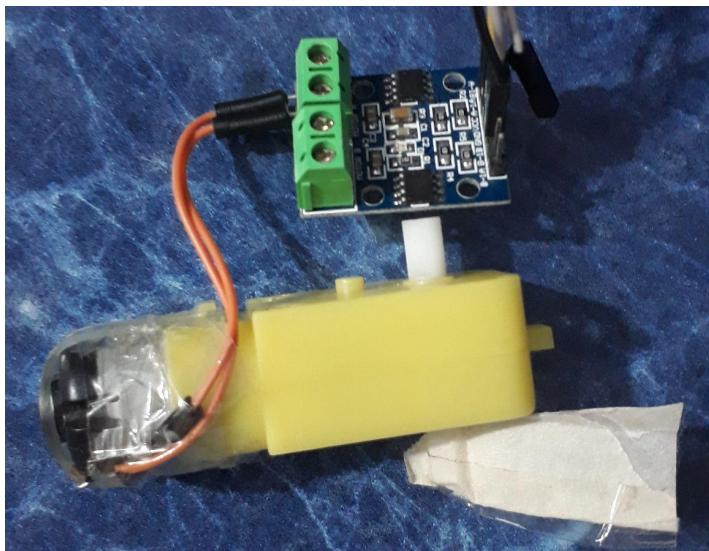
- 1) Conector USB para teclado y mouse inalámbricos.
- 2) Conector USB para la cámara fotográfica.
- 3) Pines de conexión GPIO.
- 4) Interfaz HDMI.
- 5) Entrada de alimentación micro USB.



La imagen de la izquierda se corresponde con la placa de circuito impreso realizada a fin de reemplazar la protoboard usada para las pruebas de las conexiones. Posee cuatro simples pistas que unen el sensor a los cables GPIO, con una resistencia de $1\text{ K}\Omega$ y otra de $2\text{ K}\Omega$.

La siguiente captura ilustra el arranque del sistema operativo y la cámara usada.





Y por último, se puede divisar el puente H usado para alimentar a el actuador de 5 V, que simula el accionar de una barrera.

5. Conclusiones

El equipo diseñado consta físicamente de cuatro partes. Primeramente, se cuenta con un ordenador monoplaca: «Raspberry Pi Modelo 3 B +», que posee una arquitectura ARM64 . El mismo tiene conectado como entradas, una cámara para realizar la captura de fotografías y un sensor de ultrasonido para sensar la distancia; en tanto que como salida tiene conectado un puente H, del que se desprende un motor de 5V que simula el accionar de una barrera.

En cuanto al software, el sistema consta de un módulo central, que tiene la lógica central del problema, el cual va produciendo llamadas a módulos secundarios, los cuales hacen tareas específicas (uno se encarga de capturar imágenes, otro de sensar distancia, otro de girar el actuador y el otro realizar el reconocimiento de los caracteres alfanuméricos). Todo esto ejecutándose en el sistema operativo Kali Linux, sin interfaz gráfica, para que el rendimiento sea mayor, ya que se está utilizando un ordenador de muy bajos recursos.

En cuanto a las limitaciones del proyecto, se puede mencionar que solo está enfocado en reconocer las patentes nuevas de Argentina de automóviles, camionetas o camiones (no para motocicletas).

Como sugerencias para futuras modificaciones, se pueden mencionar el mejoramiento del algoritmo de detección de patentes (para que tenga una tasa de éxitos mayor), volcar el diseño del algoritmo con una orientación hacia la inteligencia artificial (para que aprenda de sus propios errores y sea cada vez más eficiente, dependiendo de diferentes escenarios) y por último, modificar el algoritmo para que incluya los tipos de patentes más viejas, también de motos e inclusive las patentes de otras nacionalidades.



También se podría contemplar un esquema físico de protección contra altas temperaturas y ruidos, sumado a salvaguardar la alimentación con una autonomía de una buena cantidad de horas. El uso de otros sensores más robustos sería una mejora sustancial en la confiabilidad y vida útil de todo el sistema.

Otra opción sumamente destacable para incorporar es la detección de patentes adulteradas o falsificadas, buscando reconocer ciertos patrones de seguridad implementados en la chapa patente, como los que se ilustran a continuación.



6. Anexos

Componentes utilizados

A continuación, se detallan técnicamente los componentes utilizados en el proyecto, se describen con detalles relevantes y con sus funciones específicas.

1) Raspberry Pi



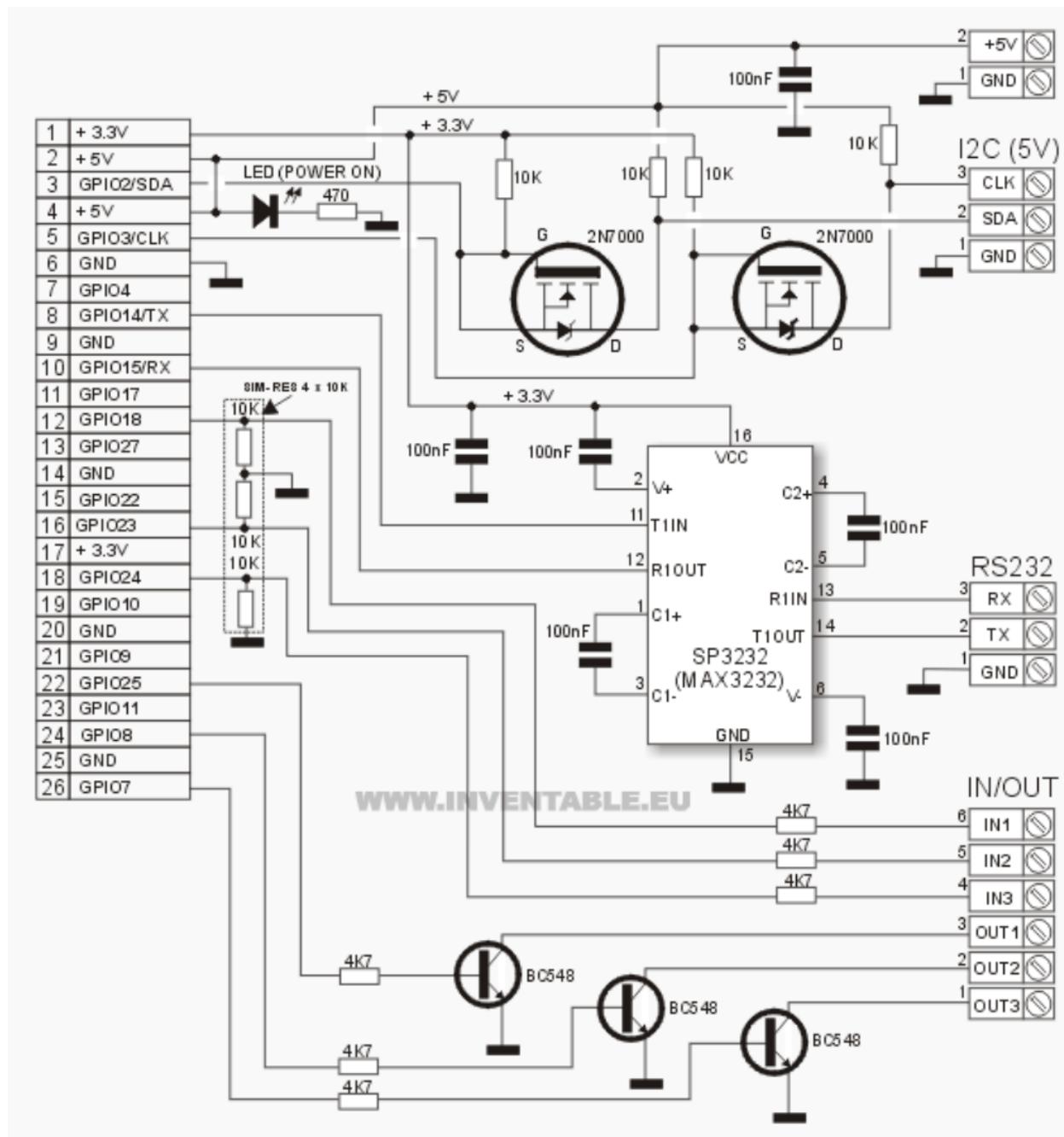
La Raspberry Pi es una serie de ordenadores monoplaca u ordenadores de placa simple (SBC por las siglas del anglicismo Single Board Computer) de bajo costo

desarrollado en el Reino Unido por la Raspberry Pi Fundación, con el objetivo de crear un mini ordenador accesible.

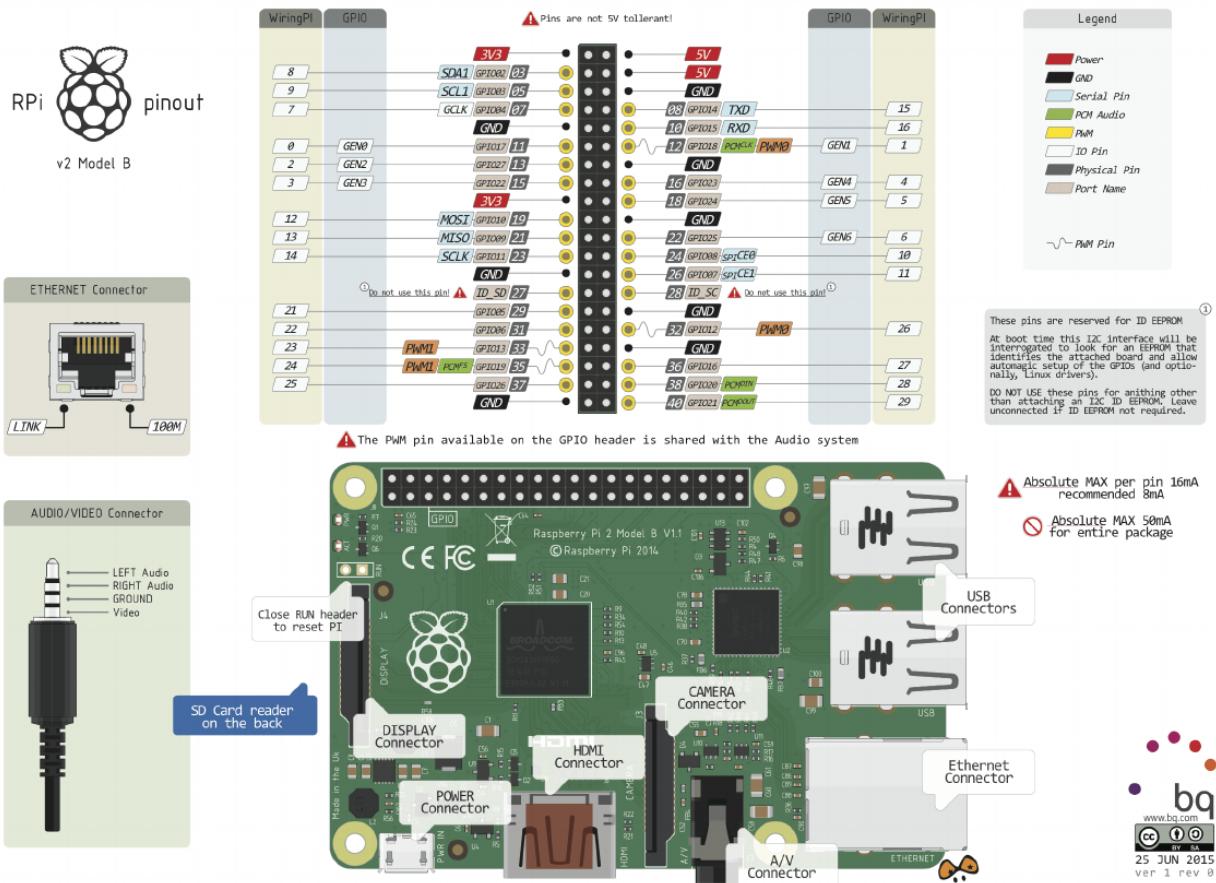
Esta “mini computadora” tiene muchas funciones y utilidades, en este proyecto en particular, nos brinda la potencia de procesamiento necesaria para correr los programas pertinentes a la inteligencia artificial y librerías, nos proporciona salidas y entradas (tanto analógicas como digitales) y permite alivianar la carga de trabajo y programación para casi cualquier uso.

En este proyecto en particular se encarga de gestionar eficientemente todas las salidas y entradas, ejecutar los códigos de programación y almacenar los datos como las imágenes de la patente entre otras cosas.

Diagrama eléctrico de la Raspberry Pi:

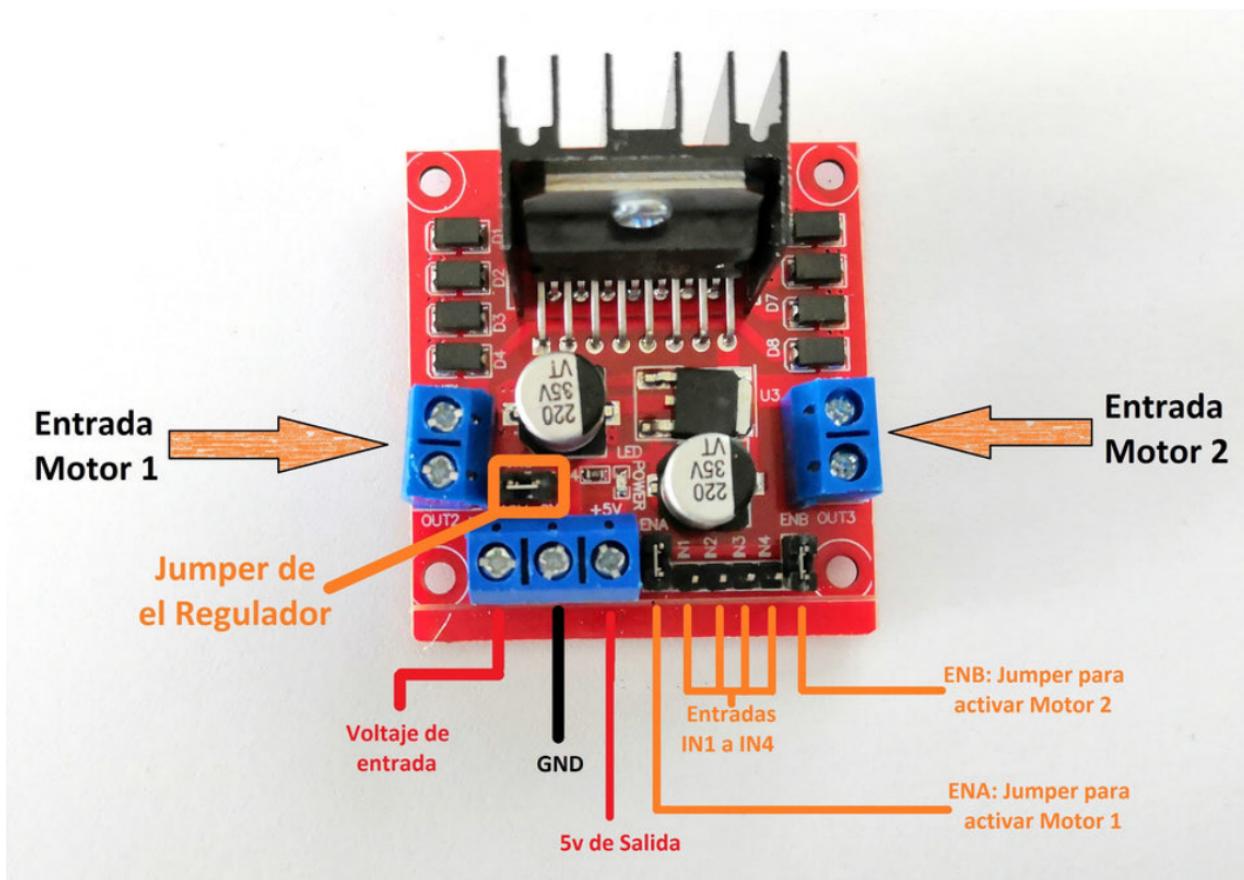


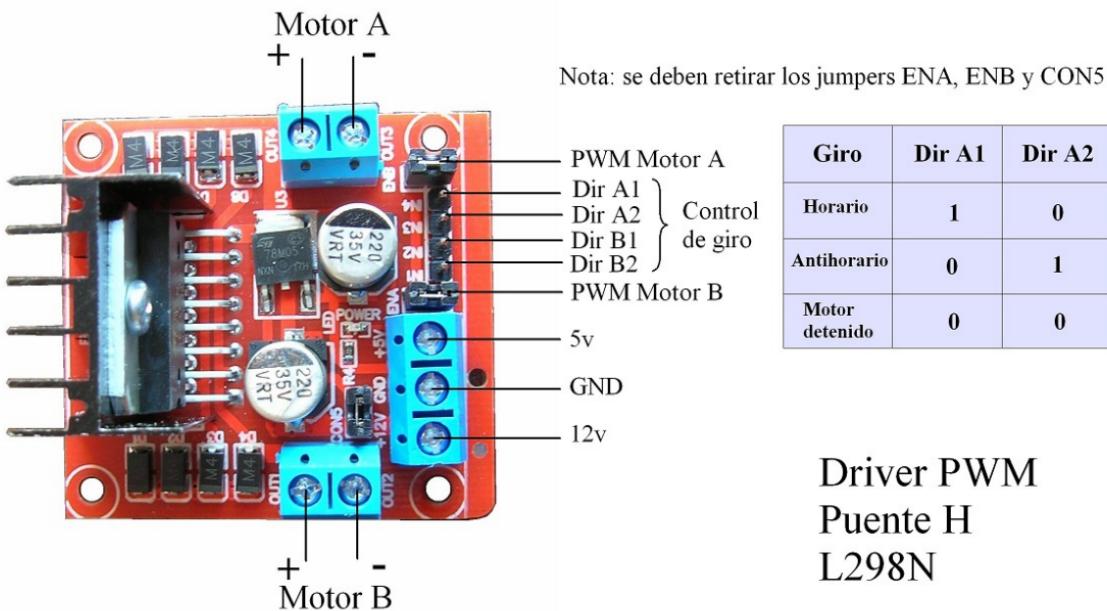
A continuación se adjunta un esquema básico de funcionamiento de salidas y entradas del prototipo:



2) Puente H

Este elemento es necesario para controlar de forma eficiente los actuadores, en este caso un motor de corriente continua que se utiliza para mover la barrera de acceso en caso que sea afirmativa la respuesta a la patente detectada.





Descripción del componente:

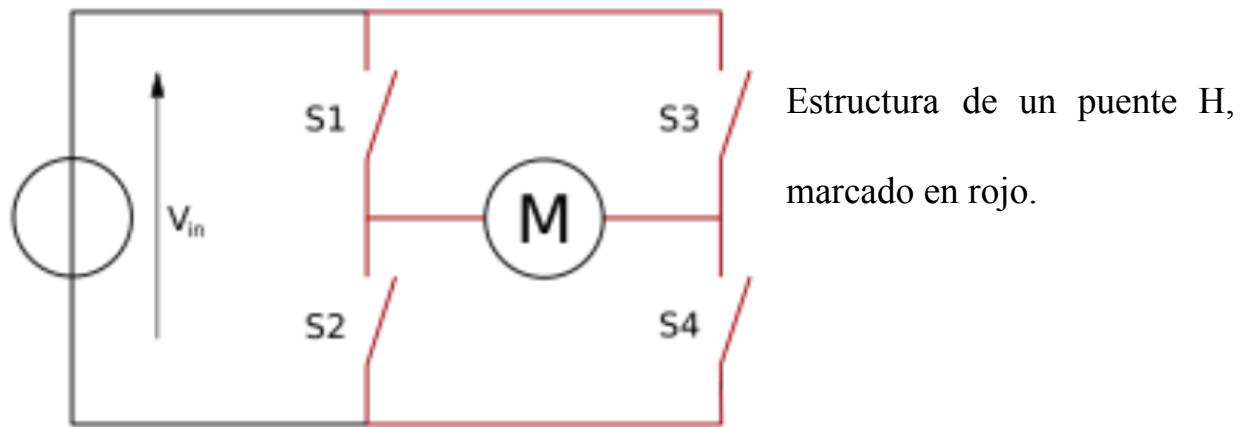
Un Puente en H es un circuito electrónico que generalmente se usa para permitir a un motor eléctrico DC girar en ambos sentidos, avance y retroceso. Son ampliamente usados en robótica y como convertidores de potencia. Los puentes H están disponibles como circuitos integrados, pero también pueden construirse a partir de componentes discretos.

Aplicaciones:

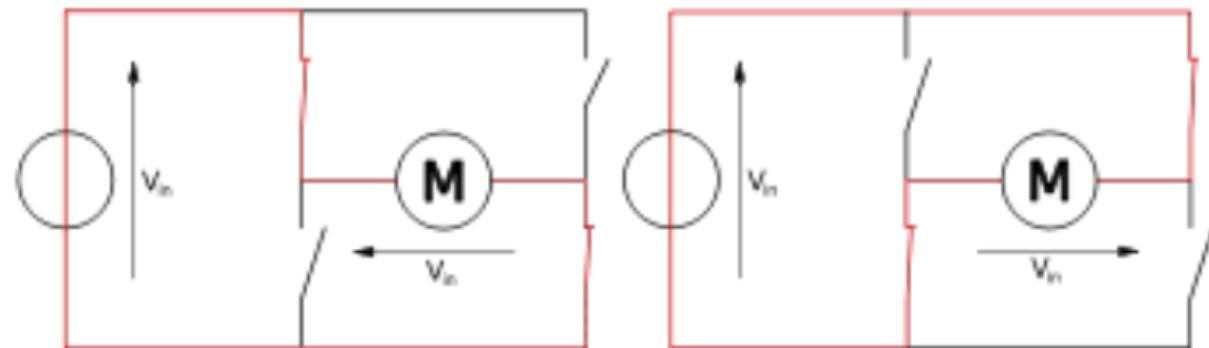
Como hemos dicho el puente H se usa para invertir el giro de un motor, pero también puede usarse para frenarlo, al hacer un corto entre las bornas del motor, o incluso puede usarse para permitir que el motor frene bajo su propia inercia,

cuando desconectamos el motor de la fuente que lo alimenta. En el siguiente cuadro luego de los diagramas se resumen las diferentes acciones.

A continuación se adjuntan los diagramas de un puente H convencional, para comprender su estado y funcionamiento básico para luego ir a sus acciones.



Estructura de un puente H,
marcado en rojo.

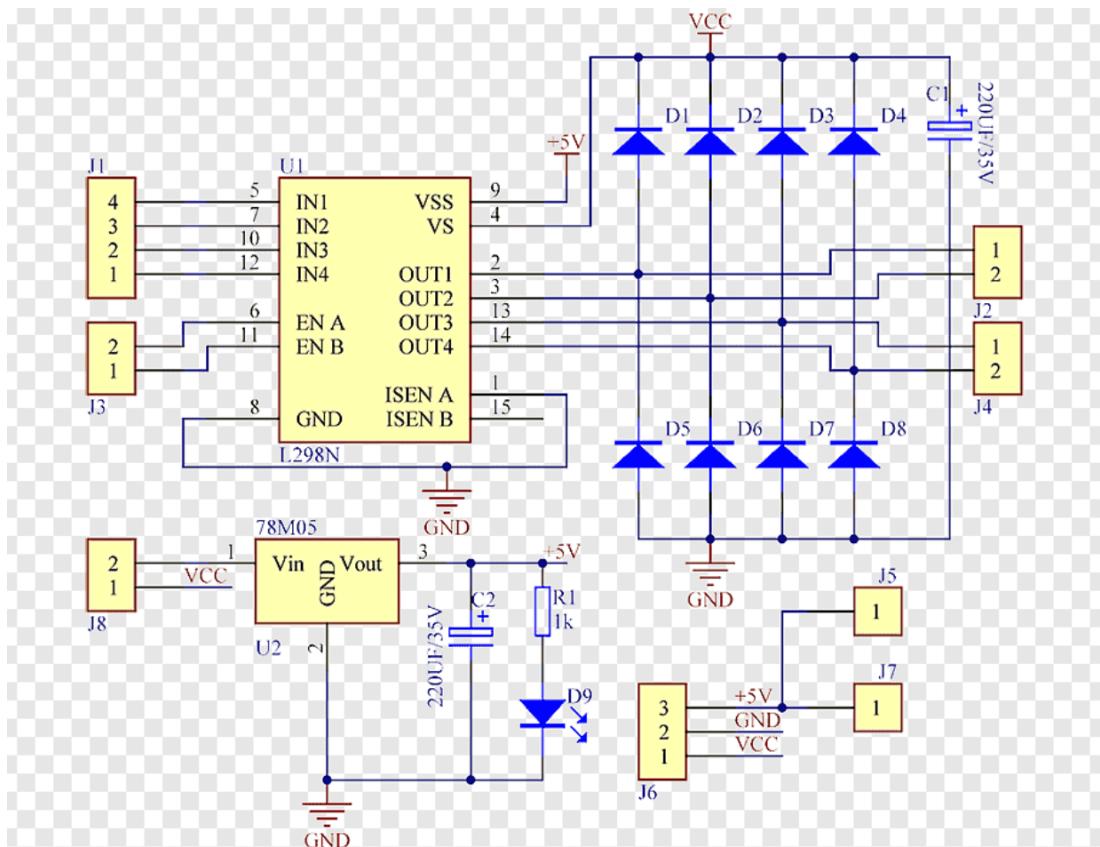


Los dos estados básicos del circuito.

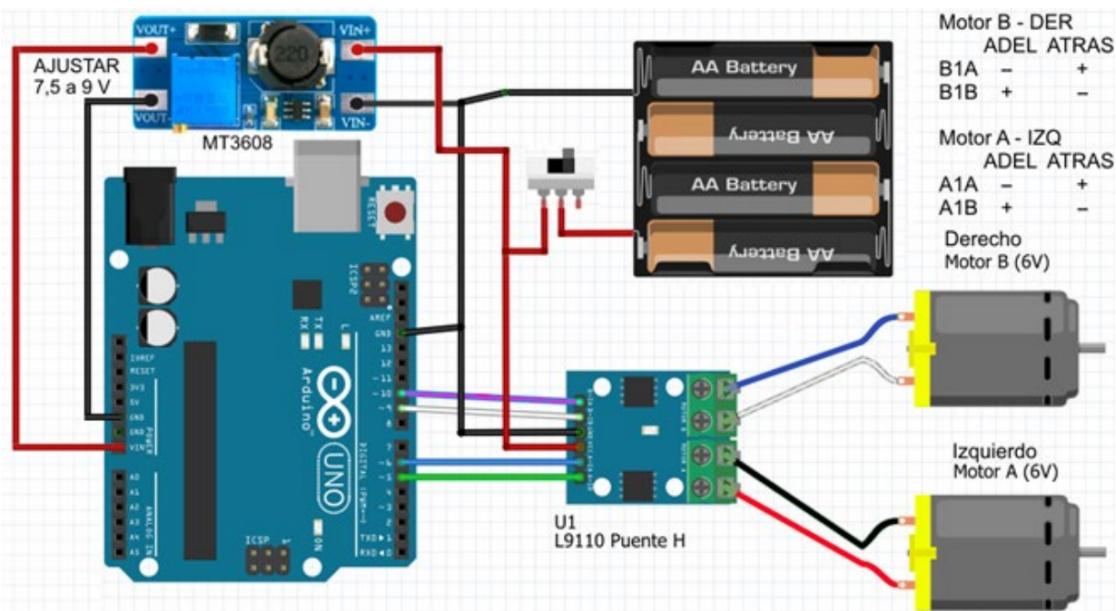
S1	S2	S3	S4	Resultado
1	0	0	1	El motor gira en <i>avance</i>
0	1	1	0	El motor gira en <i>retroceso</i>
0	0	0	0	El motor se detiene bajo su inercia
1	0	1	0	El motor se detiene
0	1	0	1	El motor se detiene
1	1	0	0	Cortocircuito
0	0	1	1	Cortocircuito
1	1	1	1	Cortocircuito

Tabla de estados con sus acciones asociadas del puente H.

Diagrama eléctrico del puente H:



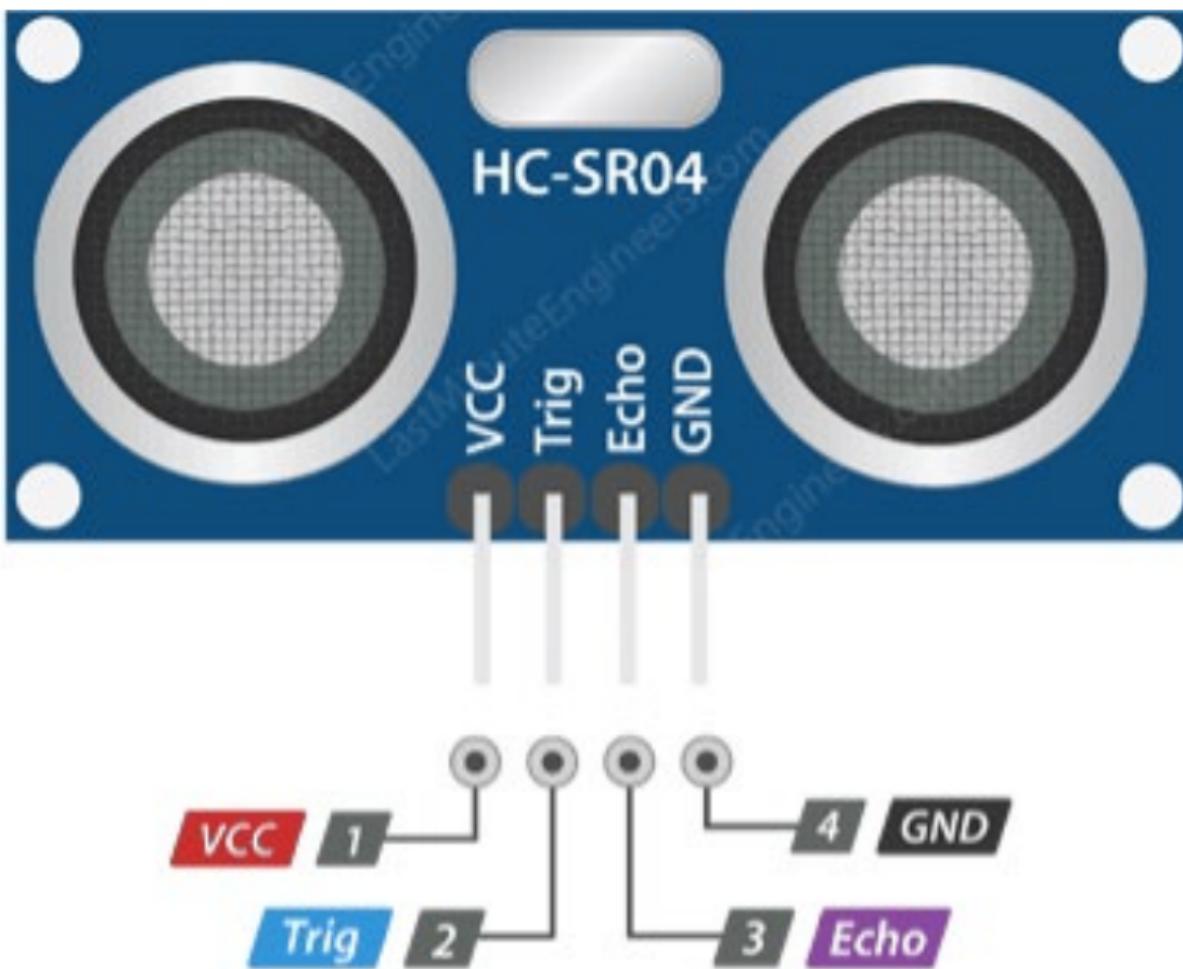
Se adjunta un diagrama básico de un puente H genérico utilizado en el proyecto para controlar los motores de corriente continua.



Se adjunta también esquema de funcionamiento básico de un puente H a modo de ejemplo, esta vez, con un Arduino y dos motores de salida con su respectiva fuente y potenciómetro.

El único cambio respecto al prototipo es la utilización de la raspberry por motivos de procesamiento requerido para el uso en particular del sistema implementado.

3) Sensor ultrasónico



Los sensores de ultrasonidos o sensores ultrasónicos son detectores de proximidad que trabajan libres de roces mecánicos y que detectan objetos a distancias que van desde pocos centímetros hasta varios metros. El sensor emite un sonido y mide el tiempo que la señal tarda en regresar.

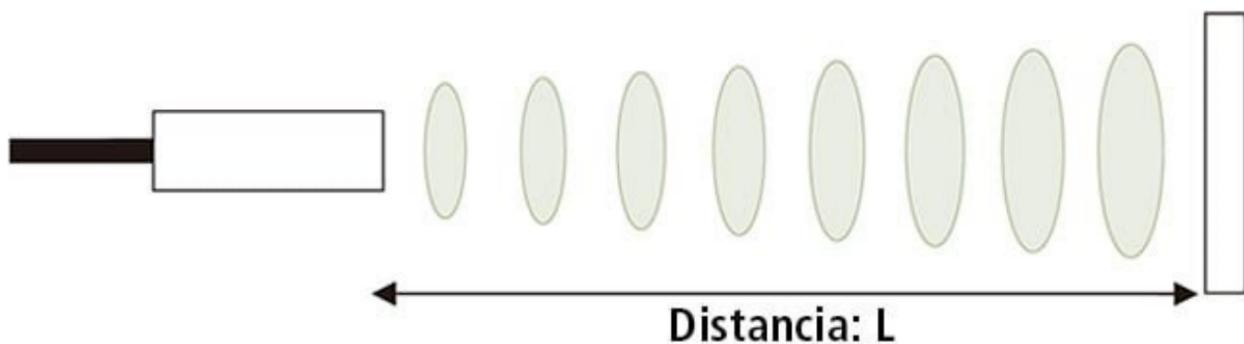
Estos reflejan en un objeto, el sensor recibe el eco producido y lo convierte en señales eléctricas, las cuales son elaboradas en el aparato de valoración. Estos

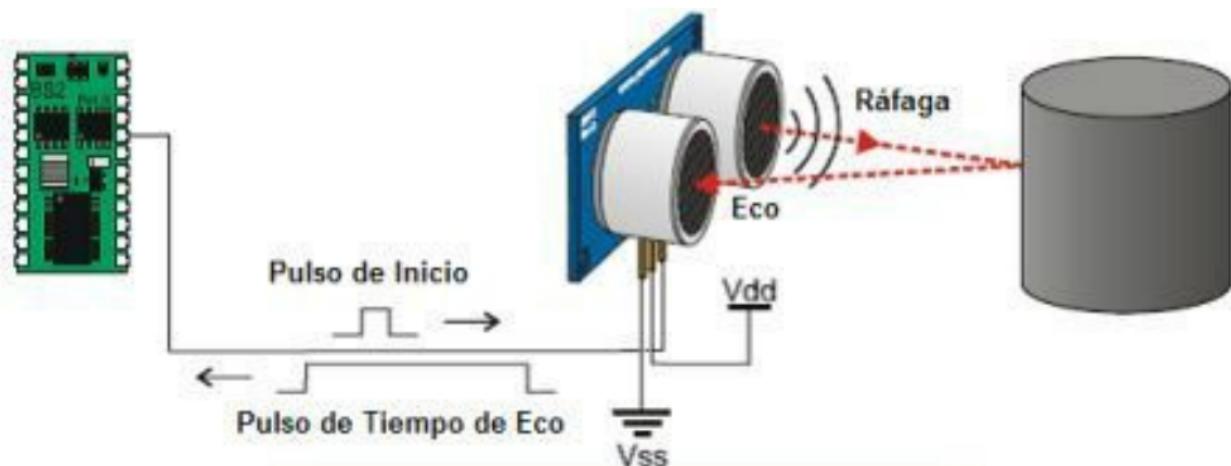
sensores trabajan solamente donde tenemos presencia de aire (no pueden trabajar en el vacío, necesitan medio de propagación) y pueden detectar objetos con diferentes formas, colores y superficies. Así como materiales, que pueden ser sólidos, líquidos o polvorrientos. Sin embargo, han de ser deflectores de sonido.

Los sensores trabajan según el tiempo de transcurso del eco, es decir, se valora la distancia temporal entre el impulso de emisión y el impulso del eco.

Reseña y principio de detección y/o funcionamiento:

Como su nombre lo indica, los sensores ultrasónicos miden la distancia mediante el uso de ondas ultrasónicas. El cabezal emite una onda ultrasónica y recibe la onda reflejada que retorna desde el objeto. Los sensores ultrasónicos miden la distancia al objeto contando el tiempo entre la emisión y la recepción.

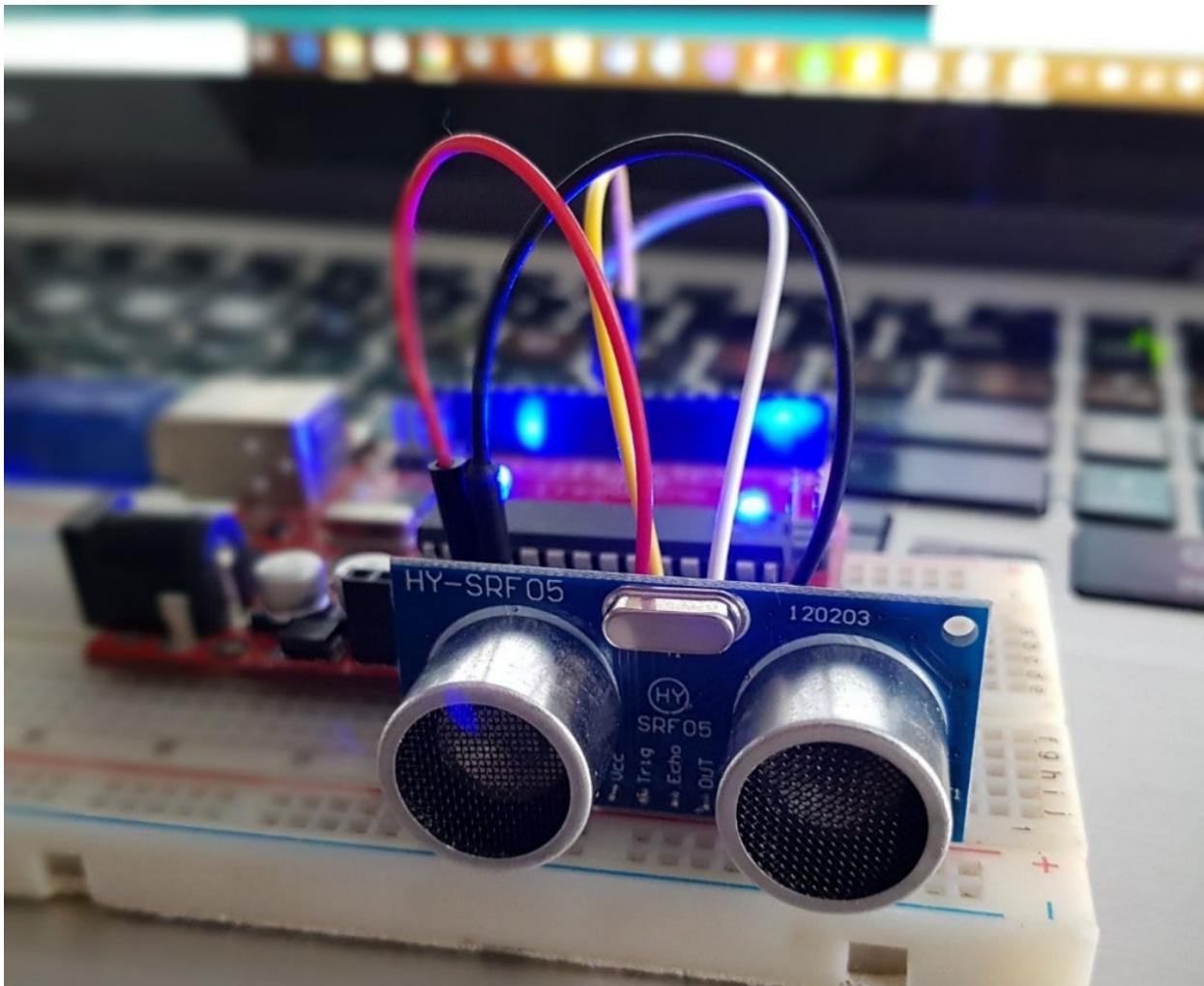




Ventajas e inconvenientes del Hc – Sr04:

Este sensor, al no necesitar el contacto físico con el objeto, ofrece la posibilidad de detectar objetos cercanos o lejanos, además detecta cualquier material, independientemente del color o superficie, al mismo alcance, sin ajuste ni factor de corrección. Los sensores ultrasónicos tienen una función de aprendizaje para definir el campo de detección, con un alcance mínimo de 6 mm y máximo de precisión de hasta 2 metros en caso de este modelo.

El problema que presentan estos dispositivos son las zonas ciegas y el problema de las falsas alarmas. La zona ciega es la zona comprendida entre el lado sensible del detector y el alcance mínimo en el que ningún objeto puede detectarse de forma fiable y las falsas alarmas dependen de objetos extraños o por ejemplo un animal que se cruce por enfrente interrumpiendo el flujo de sonidos puede generar una falsa lectura de distancia.



Sensor montado en una protoboard listo para usarse.

Esquema eléctrico de un sensor ultrasónico:

Se adjunta en este caso con un Arduino que cumple la misma función que nuestra raspberry, que es la de activar y procesar los datos provenientes del sensor. Se adjunta el esquema básico del sensor HC-SR04 utilizado en el proyecto.

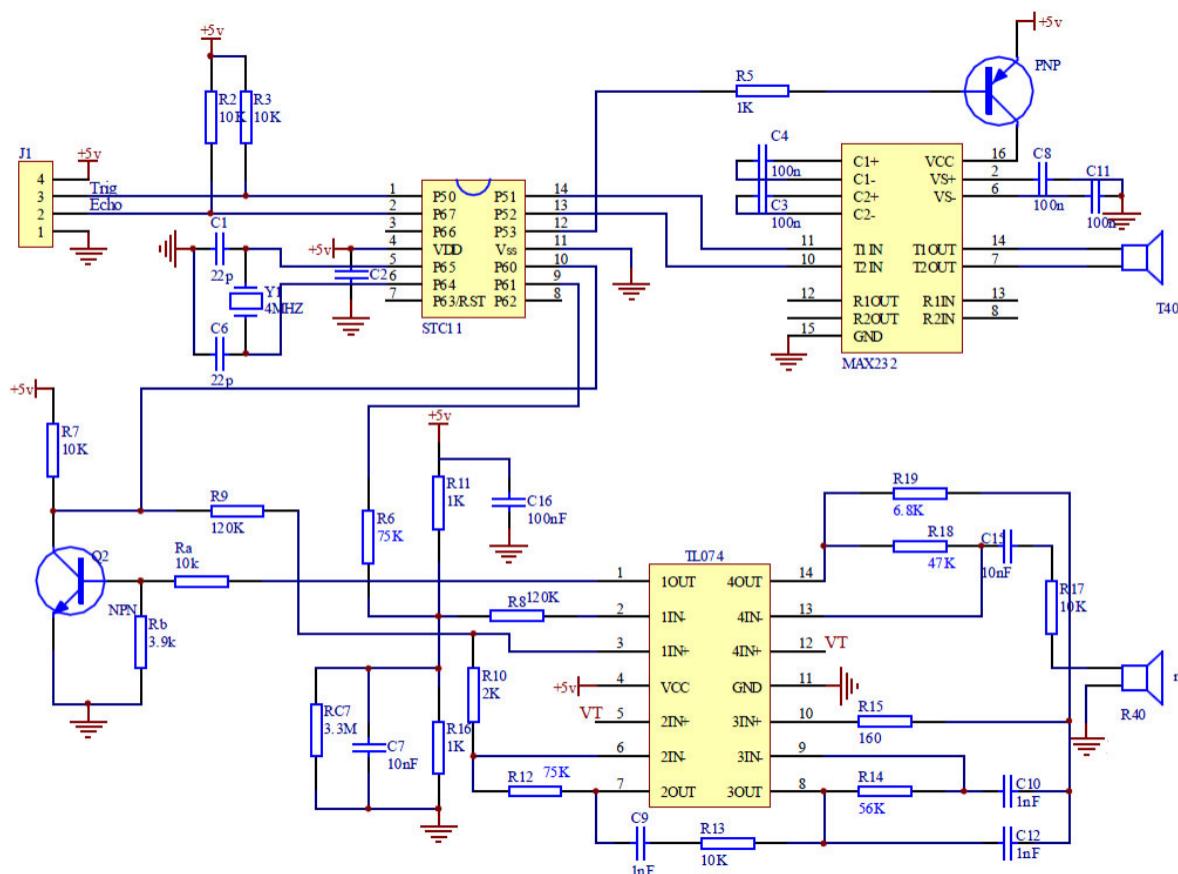
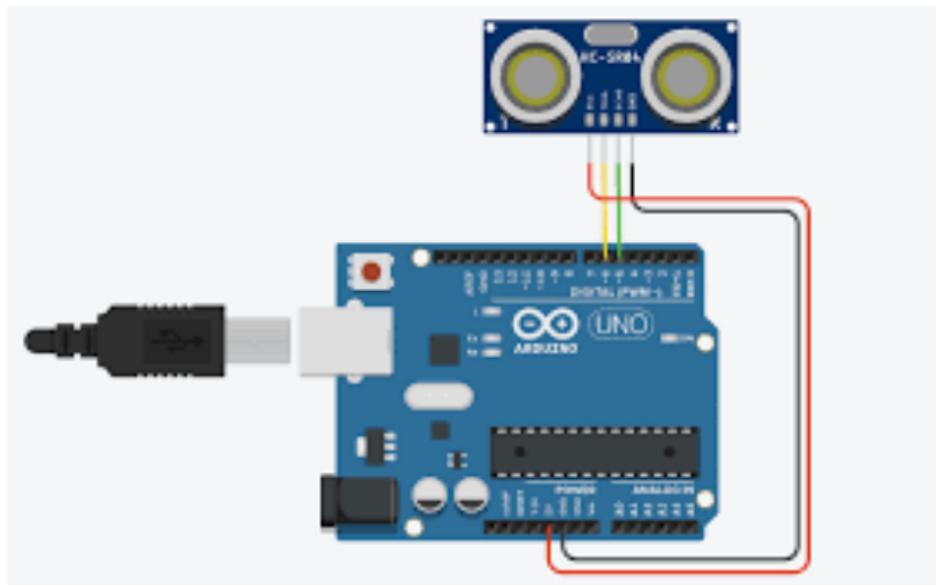


Diagrama eléctrico básico correspondiente al modelo utilizado

4) Cámara Digital



Cámara digital utilizada para la captura de las imágenes de los vehículos con sus respectivas patentes.

Se adjunta un diagrama de una cámara de fotos digital clásica de formato estándar.

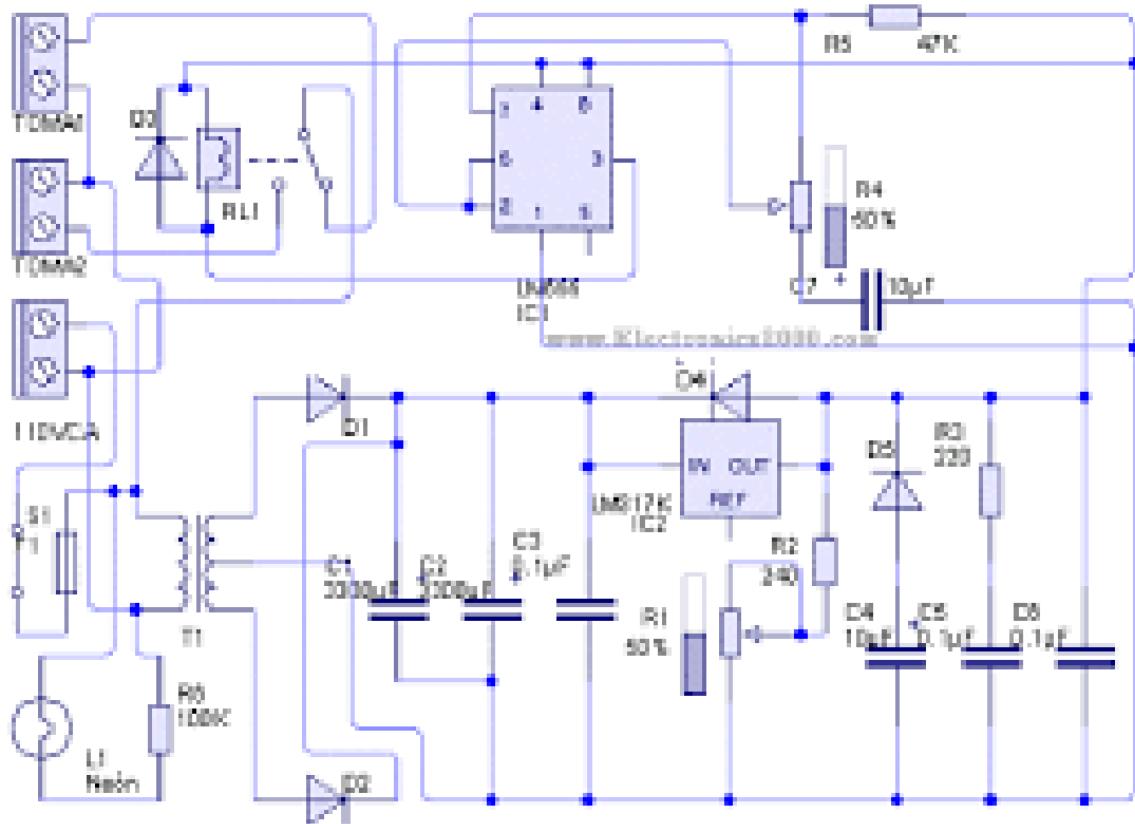


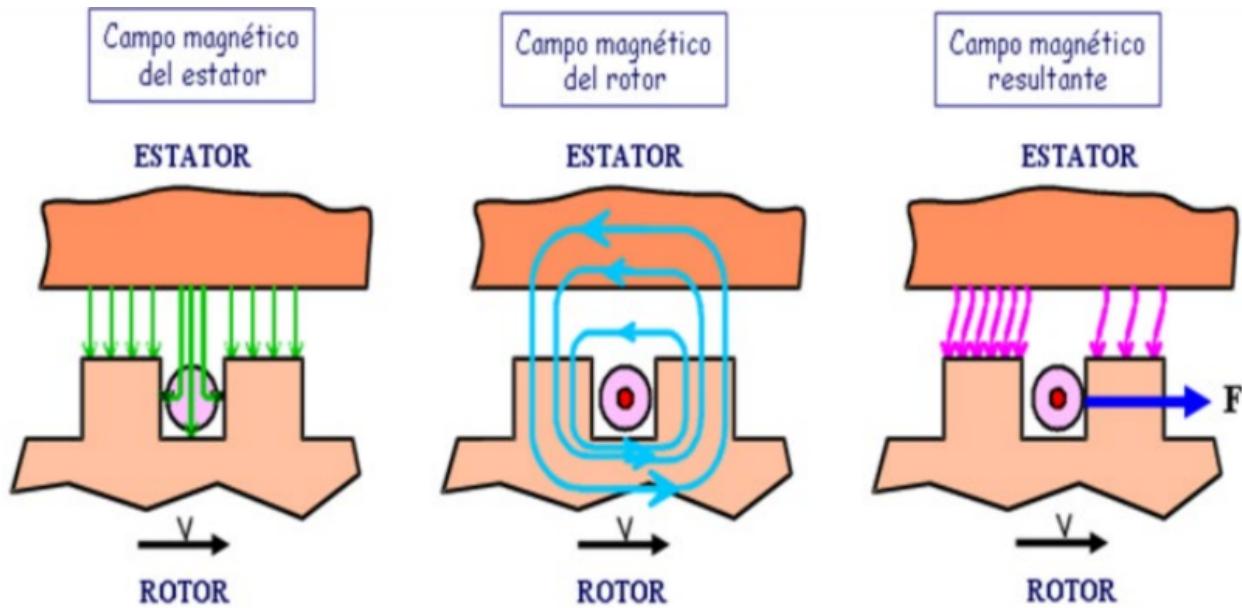
Diagrama eléctrico de una cámara digital fotográfica como la utilizada.

5) Motor de corriente continua

Un motor de corriente continua (CC) es una máquina giratoria que convierte energía eléctrica en energía mecánica. Esta funcionalidad está basada en el principio de inducción, bajo el cual se crea una fuerza electromagnética a partir de una corriente de entrada que, a la vez, crea un movimiento giratorio. En los motores de corriente directa el par motor depende del campo giratorio.

Funcionamiento:

El funcionamiento está basado en el principio de inducción, bajo el cual se crea una fuerza electromagnética a partir de una corriente de entrada que, a la vez, crea un movimiento giratorio.



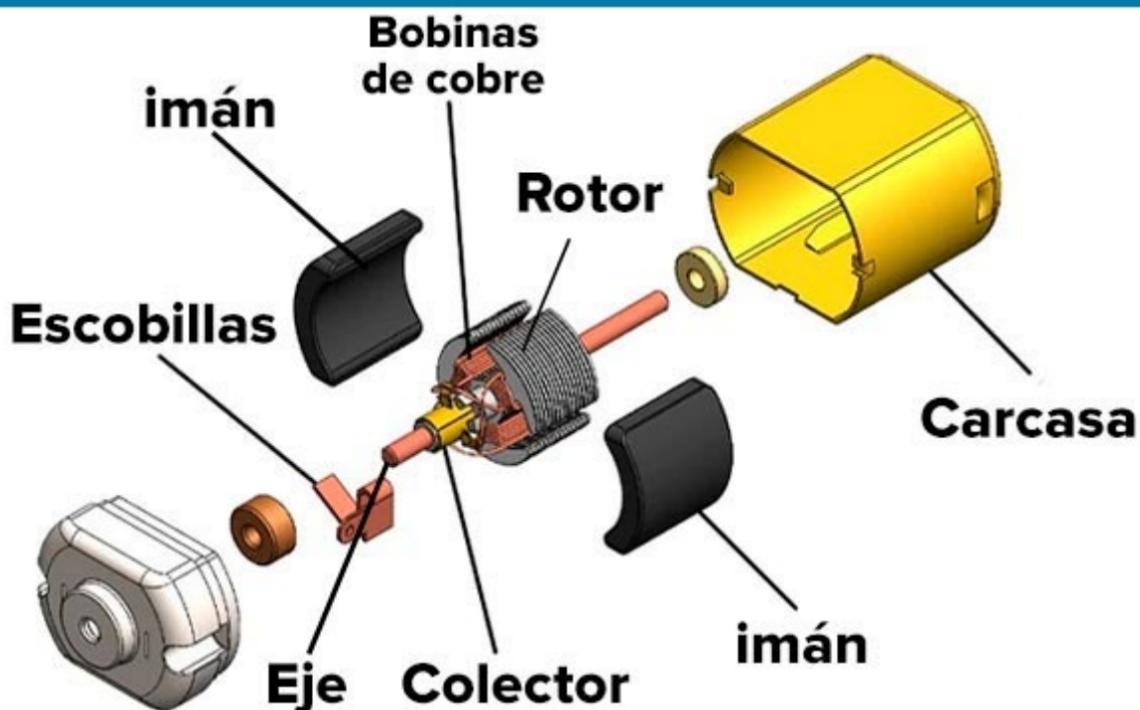
La eficiencia de un motor de CC es la relación entre la potencia de entrada y la potencia de salida medida en vatios. Muchas de las piezas y partes el motor influyen (reduciendo el impacto) en la eficiencia del motor, teniendo como resultado el calor generado.



Vista en corte de un motor de corriente directa donde se puede apreciar sus distintas partes.

Componentes básicos de un motor cc:

Partes de un Motor CC (corriente continua)



Despiece y vistas de las partes básicas de un motor de corriente continua

Los componentes de un motor de corriente continua se dividen en

las siguientes partes:

- **Estator:** parte que da soporte mecánico al aparato y contiene los polos de la máquina, que pueden ser devanados de hilo de cobre sobre un núcleo de hierro o imanes permanentes.

- **Rotor:** es un componente generalmente de forma cilíndrica, también devanado y con núcleo, alimentado con corriente continua a través del colector formado por delgas. Las delgas se fabrican generalmente de cobre y están en contacto alternante con las escobillas fijas.

- **Imán:** se encarga de generar la atracción mediante la corriente para mover el núcleo

- **Carcaza:** cubierta exterior encargada de proteger el interior del motor

- **Eje:** encargado de transmitir el movimiento a lo largo del motor y darle una salida útil a la potencia generada

Diagrama eléctrico de un motor de corriente continua:

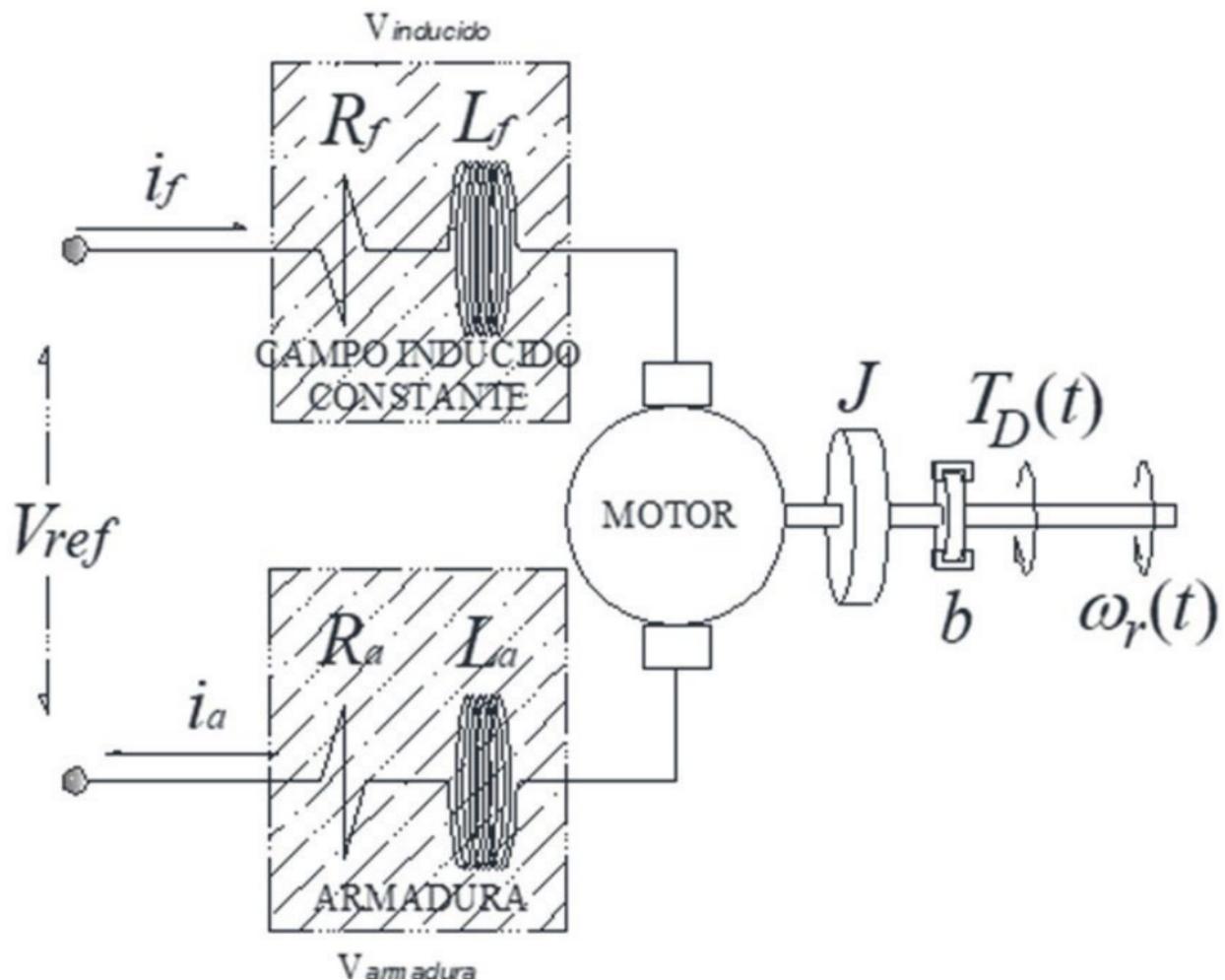
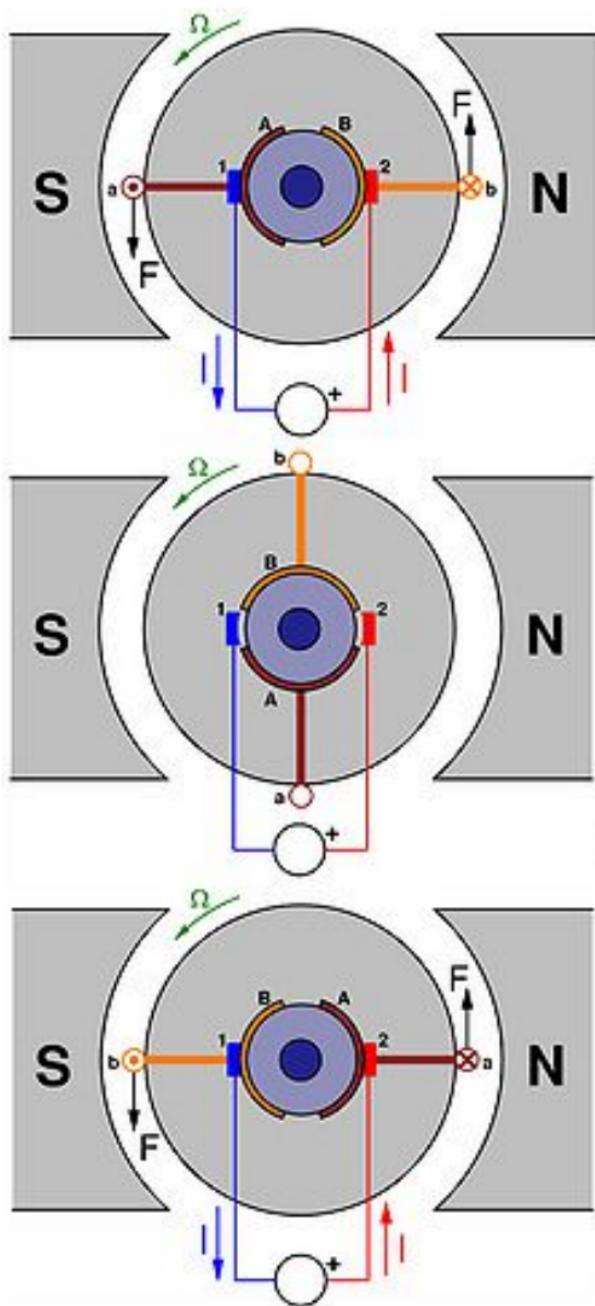


Diagrama básico de un motor de cc.



Esquema de las fuerzas actuantes en un motor cc que provocan el giro del mismo.

7. Referencias

MECATRÓNICA - Sistemas de control electrónico en la ingeniería mecánica y eléctrica 6^a

Edición Autor: BOLTON, William. Páginas: 664

ELECTROTECNIA I- Fundamentos de la Electrotecnia. Autor: Vv.aa.

Editorial: MARCOMBO, S.A. ISBN: 9788426722232 Páginas: 250

<https://pypi.org/>

<https://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/os/index.html>

<https://robotica.sandria.org/2017/12/controlar-motor-cd-con-python-en-raspberry-pi-y-modulo-puente-h/>

<https://www.infobae.com/2016/03/16/1797317-como-sera-la-nueva-patente-del-mercosur-que-en-trara-vigencia-abril/>