

# **INTERFAZ GRÁFICA PARA LA CONDUCCIÓN REMOTA DE UN PROTOTIPO CARROBOT MEDIANTE EL PROTOCOLO DE COMUNICACIÓN SSH**

*GRAPHICAL INTERFACE FOR REMOTE DRIVING  
OF A CARROBOT PROTOTYPE THROUGH THE SSH  
COMMUNICATION PROTOCOL*

***Katia Martínez Coca***

Universidad Autónoma del Carmen, México  
*katiamtzcoca@gmail.com*

***José Luis Vázquez Ávila***

Universidad Autónoma del Carmen, México  
*javazquez@pampano.unacar.mx*

***Jorge Gabriel Pacheco Richard***

Universidad Autónoma del Carmen, México  
*jpacheco@pampano.unacar.mx*

***Rafael Sánchez Lara***

Universidad Autónoma del Carmen, México  
*rsanchez@pampano.unacar.mx*

***Marco Antonio Rodríguez Blanco***

Universidad Autónoma del Carmen, México  
*mrodriguezblanco@gmail.com*

***Manuel May Alarcón***

Universidad Autónoma del Carmen, México  
*mmay@pampano.unacar.mx*

**Recepción:** 29/noviembre/2023

**Aceptación:** 13/marzo/2023

## **Resumen**

El presente trabajo desarrolla una interfaz gráfica aplicada a un sistema móvil, carrobot, que se conduce de manera remota desde una computadora mediante el protocolo SSH, el carrobot puede avanzar, retroceder, girar a la izquierda, a la derecha y activar una cámara para detectar si hay alguna señal de tránsito del tipo "ALTO". El objetivo principal consiste en desarrollar una interfaz gráfica de usuario (GUI) usando QT4 con botones de control para la guía o direccionamiento del

prototipo. Como resultado principal, se tiene una interfaz gráfica de usuario amigable aplicada a una cascada de clasificadores basada en Haar/Adaboost capaz de mostrar en tiempo real la detección de señales de tránsito usadas en la vía pública. Una de las principales conclusiones de este trabajo, es que a la GUI se le podrían agregar más señales de la vía pública para hacer cada vez más compleja la toma de decisiones del carrobot.

**Palabras Clave:** Comunicación SSH, interfaz gráfica QT4, robótica móvil, visión artificial.

## **Abstract**

*The present work develops a graphical interface applied to a mobile system, carrobot, which is driven remotely from a computer through the SSH protocol, the carrobot can move forward, backward, turn left, turn right and activate a camera to detect if there is any traffic signal of the type "STOP". The main objective is to develop a graphical user interface (GUI) using QT4 with control buttons for guiding or addressing the prototype. As a main result, there is a friendly graphical user interface applied to a cascade of classifiers based on Haar/Adaboost capable of showing the detection of traffic signal used on public roads in real time. One of the main conclusions of this work is that more road signals could be added to the GUI to make the decision-making of the carrobot more complex.*

**Keywords:** Artificial vision, mobile robotics, QT4, SSH communication.

## **1. Introducción**

El uso de GUIs para conducción remota es muy variado debido a que se puede aplicar para diferentes características a ser evaluadas como pueden ser, características personales, características del móvil o características de la vía o camino por donde se desea circular. Hacer la conducción remota más fácil y productiva ha sido una de las tareas de muchos investigadores [Fong, 2002], Fong desarrolló tres interfaces, GestureDriver, HapticDriver y PdaDriver. GestureDriver utiliza el reconocimiento de gestos visuales para un control del vehículo fácil y adaptable al operador, HapticDriver facilita las tareas de conducción, como pueden

ser estacionarse o maniobrar en espacios abarrotados y PdaDriver permite la conducción remota en cualquier lugar y en cualquier momento utilizando una computadora tipo Palm. En este trabajo se presenta una interfaz para detectar señales de tránsito que hay en la vía pública, para este caso especial es la señal de "ALTO". Para la conducción remota [Nádvornik, 2014] lo hace por medio de un dispositivo móvil con sistema operativo Android mientras que para el trabajo aquí presentado se hace por medio una computadora con sistema operativo Windows, la comunicación se hace entre una Raspberry localizada en el carrobot y una computadora por medio de SSH.

El framework QT se dio a conocer por primera vez en el año 1995, fue desarrollado por dos ingenieros noruegos, Haavard Nord y Eirik Chanble-Eng, como respuesta a la necesidad de desarrollar principalmente interfaces gráficas de usuario (GUI; por sus siglas en inglés Graphical-User Interface). QT Designer es un programa que permite desarrollar GUIs, el cual permite personalizar ventanas o cuadros de diálogo para su empleo de los sistemas de información geográfica (GIS; por sus siglas en inglés Geographic Information System) por ejemplo u otros casos. QT es un lenguaje de programación desarrollado en C++ pero se encuentra disponible para sistemas tipo Unix, Apple Mac OS X y Windows. El API multiplataforma de la biblioteca proporciona distintos modos para acceder a bases de datos utilizando SQL (Structured Query Language; en español Lenguaje de consulta estructurada) [Viera, 2017]. El paquete QT integra herramientas de desarrollo y soporte tales como QTDesigner y QTAssistant, así como librerías de clases auxiliares a QT. También incluye el compilador Gcc MinGW. QTDesigner, figura 1, es una herramienta de desarrollo que permite crear interfaces gráficas de usuario. Proporciona un conjunto de componentes estándar y un mecanismo de conexión llamado signal-slot, con el que se conectarán eventos de la interfaz con la lógica de programa que han de soportar [González, 2009].

Ahora bien, SSH (Secure Shell), es un protocolo de comunicaciones de órdenes seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente; SSH encripta la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas no

encriptadas [Quintero, 2009]. El servicio se creó como un reemplazo seguro para el Telnet sin cifrar y utiliza técnicas criptográficas para garantizar que todas las comunicaciones hacia y desde el servidor remoto sucedan de manera encriptada [Deyimar, 2022]. Los shell son necesarios para invocar o ejecutar los distintos programas disponibles en la computadora. En informática, el shell o intérprete de órdenes o intérprete de comandos es el programa informático que provee una interfaz de usuario para acceder a los servicios del sistema operativo.

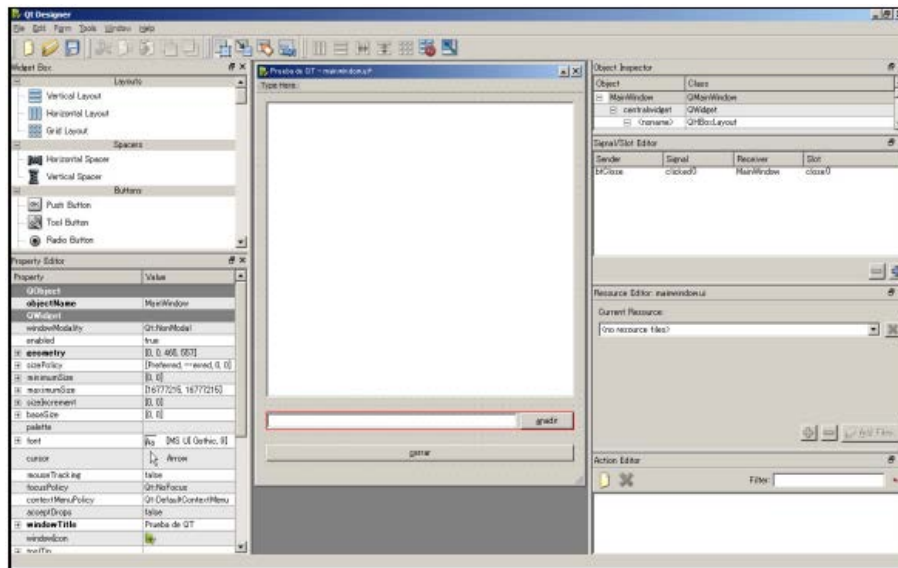


Figura 1 Pantalla principal de QTDesigner para el diseño de GUI.

La figura 2 muestra el funcionamiento de este:

- Se inicia una conexión TCP sobre el puerto 22 del servicio; puerto utilizado por defecto mediante el protocolo.

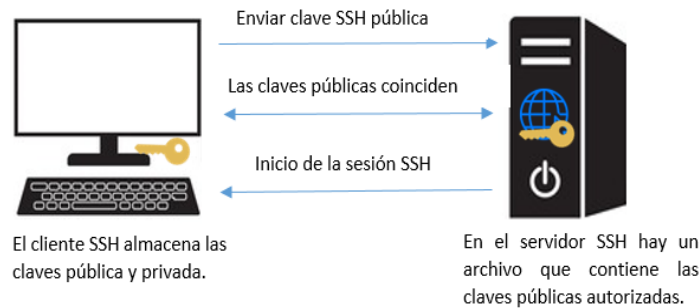


Figura 2 Funcionamiento del protocolo SSH.

- El cliente y el servidor acuerdan que versión del protocolo utilizar, así como también el algoritmo de cifrado utilizado para intercambiar la información.
- El servidor tiene en su poder dos claves, una privada y una pública; envía su clave pública al cliente.
- Cuando el cliente recibe la clave por parte del servidor, la compara con la que posee almacenada para verificar su autenticidad. El protocolo SSH exige que el cliente confirme la clave la primera vez.
- Con la clave pública del servidor en su dominio, el cliente genera una clave de sesión aleatoria, creando un mensaje que contiene dicha clave y el algoritmo seleccionado para la encriptación de la información. Toda la información es enviada al servidor haciendo uso de la clave pública que envió en un paso anterior de forma cifrada [Gardoki, 2006].
- Si todo es correcto, el cliente queda autenticado iniciando la sesión para comunicarse con el servidor.

## 2. Métodos

Como primer punto, el diseño del prototipo parte del diagrama que se observa en la figura 3, en donde para la recopilación de datos de entrada se utilizará una placa Raspberry Pi 3, conectada a una cámara, de acuerdo a las pruebas que se requieran realizar, esta información será procesada y posteriormente se enviarán órdenes a la placa Arduino mediante la comunicación serial.

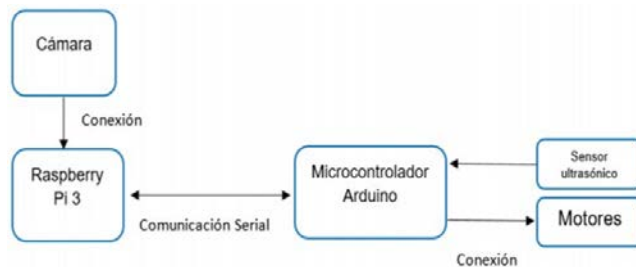


Figura 3 Diagrama de bloques del sistema del prototipo.

La unidad de procesamiento Raspberry Pi 3 puede manejar múltiples tareas como la ejecución de técnicas de visión artificial y envío de instrucciones a la placa Arduino

a través de una conexión serial, estas instrucciones serán dadas a partir de la conducción remota mediante la interfaz de operación del prototipo.

La figura 4 muestra el diagrama eléctrico del prototipo, el cual sirvió de base para armar el carrobot en donde se podrán realizar las pruebas y prácticas que sean requeridas.

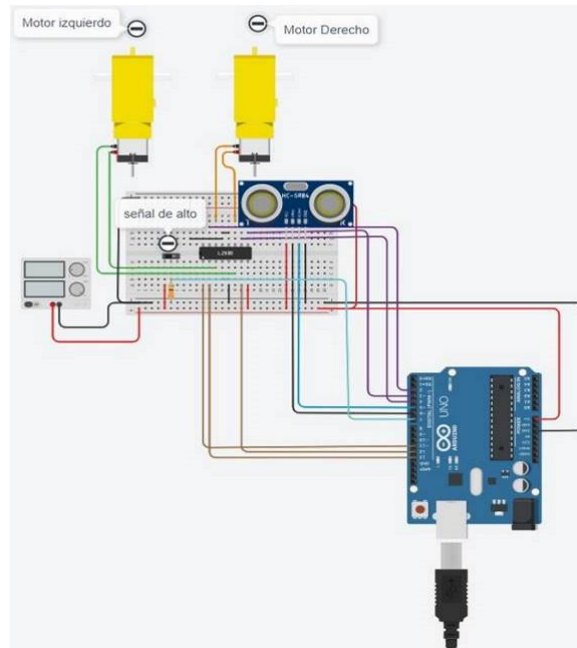


Figura 4 Diagrama eléctrico del carrobot.

Los componentes que se utilizaron para el prototipo mostrado en la figura 5, son:

- Arduino NANO: Es una placa microcontroladora, la cual es una versión reducida del Arduino UNO, eso minimiza la demanda de energía que consume y también hace que no se necesite tanto espacio para alojar la placa.



Figura 5 Prototipo carrobot.

- **Motorreductores:** Son mecanismos que regulan la velocidad de motores eléctricos, haciéndoles funcionar a un determinado ritmo. Están formados por una serie de engranajes que conforman la cadena cinemática, que son los que funcionan sobre las piezas giratorias.
- **Puente H:** Es un circuito electrónico que se usa para permitir a un motor eléctrico dc girar en ambos sentidos, avance y retroceso.
- **Sensor ultrasónico:** este sensor mide la distancia al objeto, contando el tiempo de emisión y la recepción, utiliza un elemento ultrasónico único tanto para la emisión como para la recepción.
- **Raspberry Pi 3:** es un ordenador de placa reducida de la fundación Raspberry, con sistema integrado, unidad central de procesamiento, memoria RAM, almacenamiento integrado, puertos de propósito general, salidas multimedia, periféricos de bajo nivel y conectividad de red ethernet. [Salcedo, 2015]
- **Cámara web:** Con una resolución de video 640x480 píxeles.

Para realizar pruebas físicas de la conducción remota del prototipo, se creó la interfaz gráfica con el framework QT4, y mediante la herramienta QTDesigner se hizo el diseño, el cual se muestra en la figura 6. El diseño de esta GUI constituye una de las principales aportaciones de este trabajo, en la figura 6 se muestra como verá el usuario la pantalla de computadora con sus respectivos botones.



Figura 6 Interfaz gráfica para la conducción remota del prototipo carrobot.

La interfaz establece la comunicación serial entre el Arduino y la Raspberry Pi 3, además, sirve para poder obtener la imagen de la cámara y ver en tiempo real lo que hay delante del carrobot. Es muy importante resaltar que en el presente trabajo se cargó una cascada de clasificadores basada en Adaboost, como método de clasificación, y Haar para la extracción de características para la detección de señales de alto, este método fue creado y utilizado en [Viola, 2004] para la detección de rostros. Por medio de la interfaz, se podrá ver cuando la señal de “ALTO” esté siendo detectada, por otra parte, se incluyeron los botones de adelante para avanzar, atrás para retroceder, izquierda, derecha y detener para conducir el carrobot de manera remota.

La comunicación remota se realiza mediante SSH, ya que el carrobot se maneja remotamente desde la PC en donde se ve la imagen de la interfaz, así como todo lo que esté en la Raspberry Pi 3 y para ello se hace la conexión a la Raspberry Pi 3 mediante SSH (Secure Shell) el cual permite la comunicación con otro equipo por medio de una interfaz de comandos, pero usando un canal seguro con encriptación y el puerto TCP 22 [Martínez, 2009]. Para instalar en Debían/Ubuntu usar: `#sudo apt-get install openssh-server`. Por defecto, el servidor SSH queda habilitado después de instalarlo y se puede usar SSH directamente desde la ventana de la terminal como se muestra en la figura 7, donde `pi@10.42.0.76` es la dirección IP de la Raspberry Pi 3 o el nombre de dominio al que se está tratando de conectar y el password es la contraseña que actualmente tiene la Raspberry Pi.

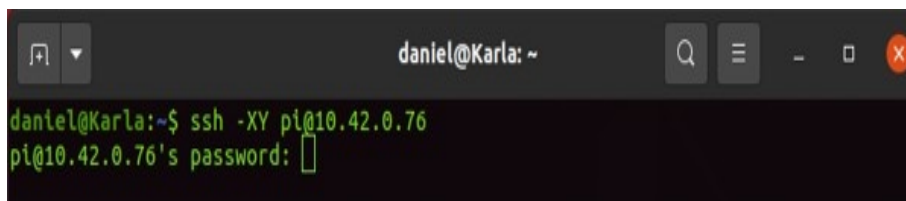
A screenshot of a terminal window on a PC. The window title is "daniel@Karla: ~". The terminal shows a green prompt "daniel@Karla:~\$" followed by the command "ssh -XY pi@10.42.0.76". Below the command, the prompt "pi@10.42.0.76's password:" is visible with a cursor. The terminal window has standard Linux window controls (minimize, maximize, close) in the top right corner.

Figura 7 Conexión de PC a Raspberry Pi 3 mediante SSH.

Al tener una conexión exitosa se puede navegar dentro de la Raspberry Pi a través del Shell, por consiguiente, se abre la interfaz gráfica mediante la línea de comando como se muestra en la figura 8.



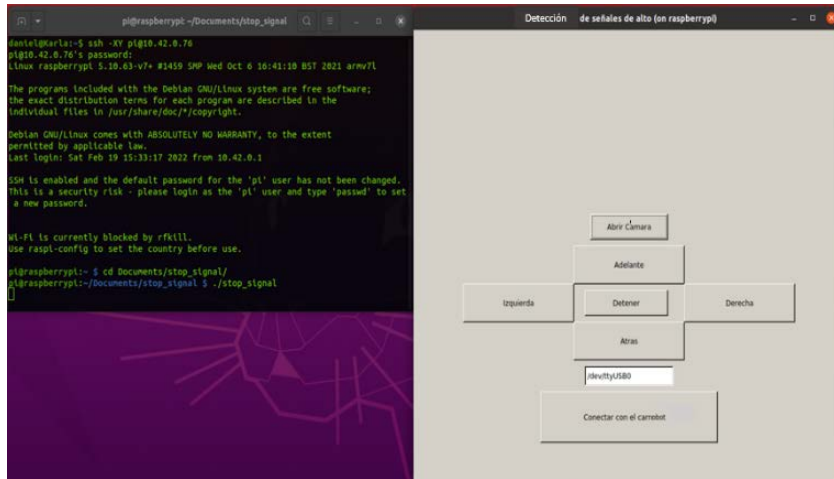


Figura 8 Conexión con la interfaz gráfica.

### 3. Resultados

En esta sección se presentan los resultados por medio de imágenes de la pantalla de comandos junto con los botones de la interfaz gráfica. Se muestran las imágenes del funcionamiento real de la interfaz gráfica en conjunto con el carrobot el cual se conduce remotamente mediante el protocolo SSH y un clasificador de objetos enfocado en la detección de señales que hay en la vía pública, como son las señales de “ALTO”. En la presentación de los resultados también se describe que tipo de botones se le han agregado a la interfaz.

Como primer punto una vez que se haya establecido la conexión SSH, se busca el directorio de la interfaz para ejecutarla y se carga el clasificador en cascada para la detección de señales de “ALTO” que se ha elegido para esta propuesta. Con la tecla c, al cargar correctamente el clasificador, los botones de “Adelante”, “Atrás”, “Derecha” e “Izquierda” se colorean en rojo como se observa en la figura 9. En este punto se puede abrir la cámara con el botón de “Abrir cámara”, como se observa en la figura 10 al presionar el botón “Abrir cámara” este se pone de color verde. A continuación, se establece la comunicación entre la placa Arduino y la Raspberry Pi, las cuales están conectadas mediante el puerto serial, si la conexión es exitosa manda un mensaje a la terminal “Conexión con el Arduino con éxito” y el botón de “conectar con el carrobot” de la interfaz se colorea de verde como se observa en la figura 11. Si la conexión no es exitosa, este botón se pinta de rojo.

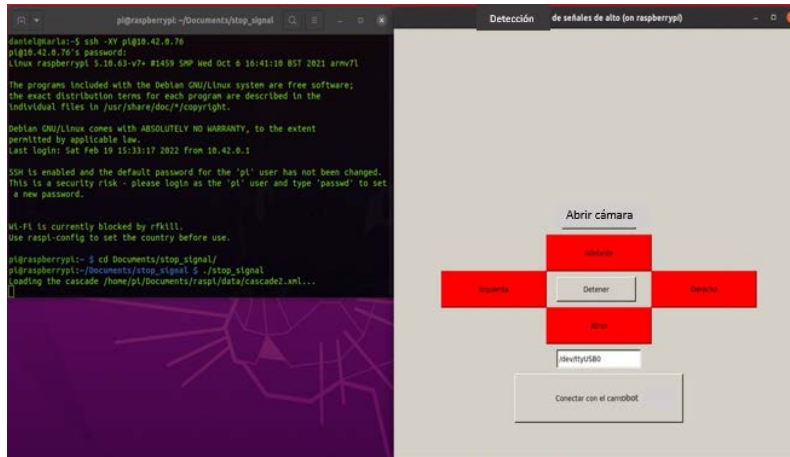


Figura 9 Ejecución de la interfaz y carga del clasificador.

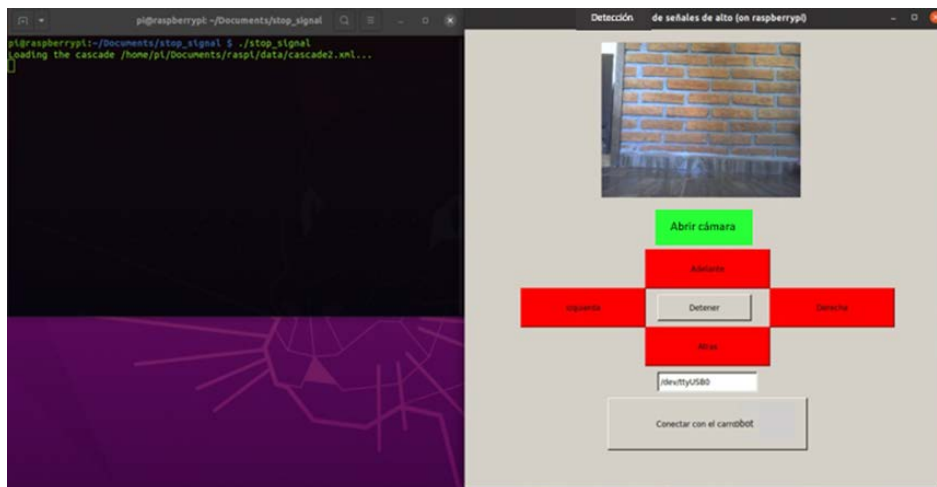


Figura 10 Pruebas de funcionamiento del botón “Abrir cámara”.

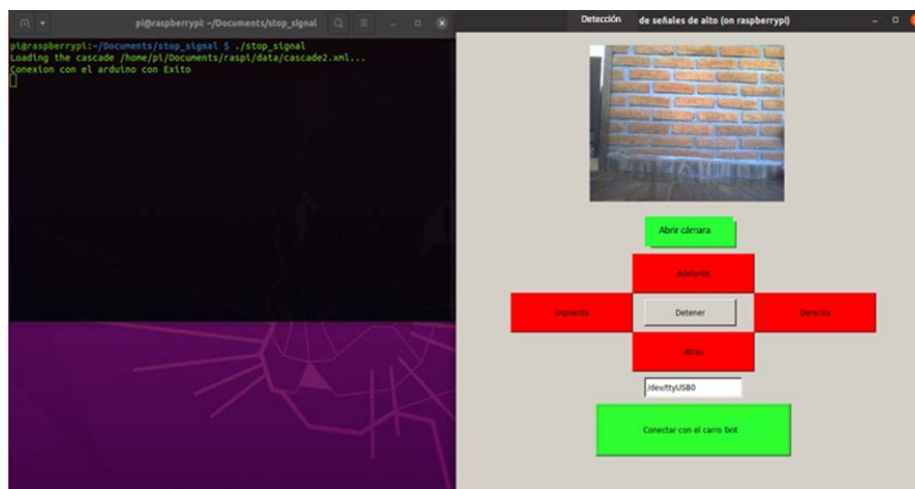


Figura 11 Pruebas de funcionamiento del botón “Conectar con el carrobot”.

Una vez que ya se estableció una conexión exitosa con el Arduino, se puede controlar el carrobot con las teclas w, a, d, x, s, a través del teclado del Raspberry. Con la tecla “w” se activa el botón adelante de la interfaz, como se muestra en la figura 12. De la misma manera, con la tecla “a” se activa el botón izquierdo, con la tecla “d” se activa el botón derecho, con la tecla “x” se activa el botón atrás y con la tecla “s” se activa el botón detener. Con estas teclas (w, a, d, x, s) se envía un paquete de datos de tres dígitos por el puerto serie, el primer dígito indica la acción del carrobot (avanzar, retroceder, izquierda, derecha y detener) tomando valores en el intervalo del 1-5, el segundo dígito marca un espacio para diferenciar el primer dígito del tercero y el tercer dígito indica si hay o no una señal de “ALTO”.

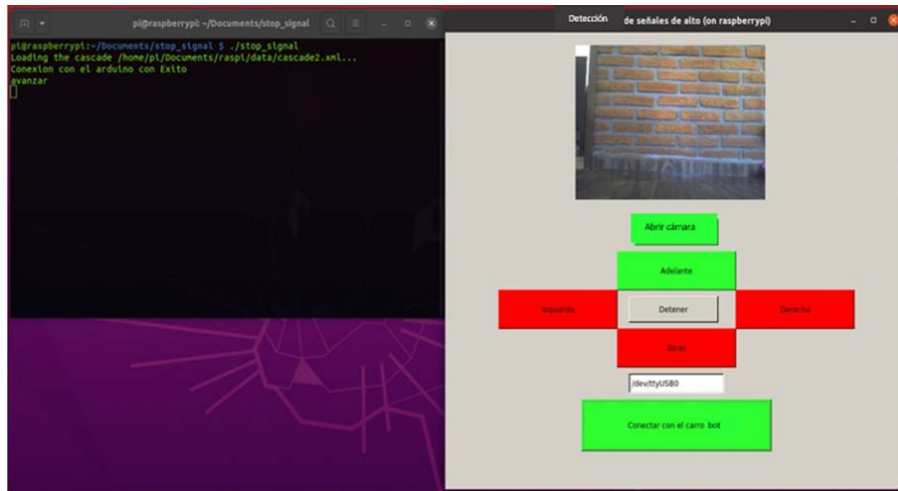


Figura12 Pruebas de funcionamiento del botón “Adelante”.

Cuando el carrobot no ve ninguna señal de alto, entonces este dígito se encuentra en 0 y puede avanzar libremente, cuando ve una señal de alto este dígito pasa a 1 indicando que el carrobot no puede realizar ningún movimiento.

La figura 13 muestra las pruebas realizadas al detectar una señal de alto, el usuario manda una indicación a la Raspberry Pi 3 mediante la interfaz que está visualizada en la PC, al pulsar los botones para avanzar, retroceder o girar el carro robot a la izquierda o derecha, pero al momento de detectar un alto se notifica su presencia y se manda la indicación al Arduino mediante la comunicación serial de detenerse debido a la presencia de la señal de alto.

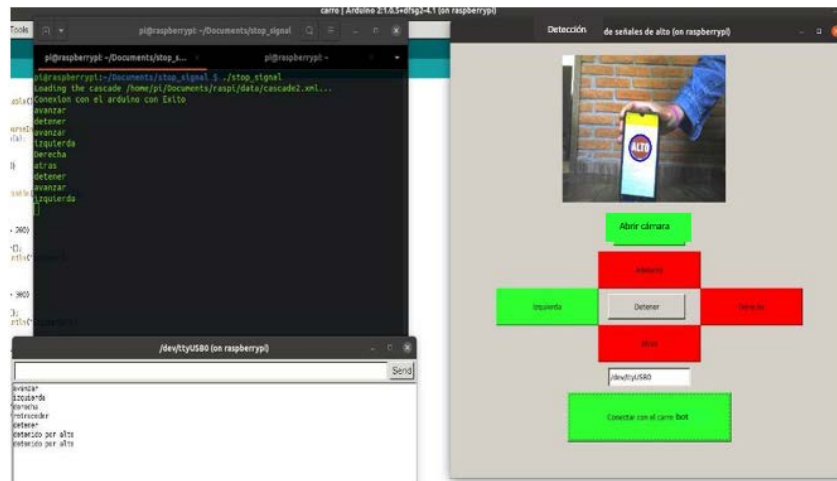


Figura 13 Pruebas de funcionamiento al detectar señal de “ALTO”.

#### 4. Discusión

En el sistema planteado se obtuvieron como resultados una GUI que es capaz de guiar de manera remota un carrobot, desde la cual se puede controlar su guiado por medio de unos botones que están en la pantalla de la computadora, el sistema puede avanzar o detenerse ya sea por instrucciones del usuario o por haber encontrado en su camino una señal de tránsito de “ALTO”. Dentro de las limitaciones que presenta el sistema en general es que solamente se ha puesto una señal de tránsito para ser interpretada como una señal de acción. Una de las posibles mejoras, es la interpretación de límites de velocidad, interpretación de señales de semáforo. Para una propuesta más avanzada, sería la conducción del carrobot considerando todas esas señales que puedan ocurrir de manera aleatoria, como sucede cuando se conduce en la vía pública. Por otro lado, para establecer el protocolo de comunicación SSH, se podría realizar a través de wifi en caso de que la Raspberry Pi tenga módulo wifi o en su caso adaptar un módulo wifi a la Raspberry pi o a través de un cable de red.

#### 5. Conclusiones

Como conclusión, se ha implementado una interfaz para la realización de pruebas con un robot móvil, el cual es conducido de manera remota por medio del protocolo SSH, con lo que se destaca que se ha cumplido con el objetivo principal de trabajo.

Otra conclusión es que a la GUI se le podrían agregar más señales de tránsito para que el carrobot ejecute tareas más complejas, como por ejemplo avanzar o detenerse ante señales de un semáforo, o realizar otras acciones cuando detecte otras señales. Gracias al protocolo SSH implementado, se puede hacer uso del sistema de forma segura, eficiente y el rango de la comunicación remota sólo va a depender del servicio de internet que se tenga contratado. La interfaz gráfica y la comunicación SSH que se ha implementado en este trabajo pueden ser aplicadas a cualquier otro robot móvil que se asemeje a las especificaciones del diagrama de bloques del sistema presentado. Para finalizar la interfaz realizada en QT ofrece un entorno colorido y amigable con el usuario, por lo que la conducción remota será fácil y cómoda, gracias a la imagen que da la interfaz y a la visualización en tiempo real del escenario del carrobot.

## **6. Bibliografía y Referencias**

- [1] Deyimar C., (2022). ¿Cómo funciona el SSH? «hostinger,» [En línea]. Available: <https://www.hostinger.es/tutoriales/que-es-ssh>. [Último acceso: 10 06 2022].
- [2] Fong, T. W., Conti, F., Grange, S., & Baur, C., (2001). Novel interfaces for remote driving: gesture, haptic, and PDA. In *Mobile Robots XV and Telem manipulator and Telepresence Technologies VII* (Vol. 4195, pp. 300-311). SPIE.
- [3] Gardoki. C., (2006). Qué es el protocolo SSH y cómo configurarlo para mejorar la seguridad de acceso a los servidores Linux. Hostalia whitepapers, pp. 1–8.
- [4] González G. D., (2009). Tutorial de QT4 Designer (Proyecto/Treball Final de Carrera). UPC, Facultat d'Informàtica de Barcelona, Departament de Llenguatges i Sistemes Informàtics.
- [5] Viera, G. (2017). Procesamiento de imágenes usando OpenCV aplicado en Raspberry Pi para la clasificación del cacao. Universidad de Piura, Facultad de Ingeniería. Programa Académico de Ingeniería Mecánico-Eléctrica. Piura, Perú.

- [6] Martínez, J. P., (2009). IPv6 para Todos: Guía de uso y aplicación para diversos entornos. Jordi Palet Martínez.
- [7] Nádorník, J., & Smutný, P., (2014). Remote control robot using Android mobile device. In *Proceedings of the 2014 15th International Carpathian Control Conference (ICCC)*. pp. 373-378.
- [8] Quintero Posso D. A., (2009). Diseño y programación de herramienta informática para la asignación del ancho de banda en tiempo real basado en políticas qos. pp. 1–34.
- [9] Salcedo Tovar, M. L., (2015). Minicomputador educacional de bajo costo raspberry pi: primera parte. *Revista Ethos Venezolana*, 7(1): 28-45.
- [10] Viola, P. and Jones, M. J., (2004). Robust real-time face detection. *International journal of computer vision*, 57(2):137–154.