



UNIVERSIDAD
NACIONAL
DE TUCUMÁN



Ministerio de Educación
Universidad Tecnológica Nacional
Facultad Regional Tucumán

Universidad Nacional de
Tucumán

Facultad de Ciencias
Exactas
y Tecnología

Laboratorio de Robótica e
Inteligencia Artificial

Universidad
Tecnológica Nacional

Facultad Regional
Tucumán

Grupo de Investigación
en Tecnologías
Informáticas Avanzadas
- G.I.T.I.A.

Reporte Técnico

Nro 1 / 2015

"Auto Robot de carreras seguidor de líneas"



Líder de proyecto:

Ing. Jorge Bustos^{1,2}

Autores:

Rubén Rosino³, Federico Luna⁴, Ángel Urrere⁵

Colaboradores:

Lic. Cristina Werenitzky⁶, Lic. Jorge Namour⁶, Ing. Jorge Ávila⁷, Ing. Julio Garay⁴, Ing. Nicolás Majorel Padilla^{2,8}

Responsables:

Dr. Adrián Will^{2,8}, Dr. Sebastián Rodríguez²

1 Ingeniería Mecánica, FACET, U.N.T.

2 Grupo de Investigación en Tecnologías Informáticas Avanzadas, UTN-FRT.

3 Ingeniería en Computación, FACET, U.N.T.

4 Ingeniería Electrónica, UTN-FRT.

5 Ingeniería en Sistemas de Información, UTN-FRT.

6 Licenciatura en Informática, FACET, U.N.T.

7 Ingeniería Mecánica, UTN-FRT.

8 Laboratorio de Robótica e Inteligencia Artificial, FACET, U.N.T.

Resumen:

Se presentan los detalles del diseño y construcción de un robot autónomo tipo automóvil seguidor de líneas. El mismo, de aproximadamente 20 cm de longitud, posee cuatro ruedas, dirección delantera, tracción trasera, sensores infrarrojos, una cámara de video, está controlado por una placa Raspberry Pi 2 y puede recorrer un circuito con curvas continuas con cambio de sentido a una velocidad aproximada de 0.4 m/s sin perder la línea. El robot fue diseñado para participar en la XIII Competencia Nacional de Robótica, realizada en la ciudad de Bahía Blanca en el mes de Noviembre de 2015.

Palabras clave: robot, seguidor de línea, Raspberry Pi.

Objetivos

1. Diseñar y construir un auto robot seguidor de líneas, utilizando dos sistemas ópticos combinados: uno del tipo infrarrojo y una cámara de video de 5 MP.
2. Coordinar un trabajo interdisciplinario entre las áreas de mecánica, electrónica y computación que permita la construcción del auto robot seguidor de líneas de acuerdo a los requerimientos exigidos por la competencia nacional de robótica.
3. Participar en competencias nacionales de robótica.
4. Desarrollar know-how en el área de robótica y disponer de una plataforma de desarrollo propio que permita implementar y probar otros algoritmos inteligentes

Descripción técnica

Diseño mecánico

Para el prototipo se optó por un diseño de cuatro ruedas con motor accionando una de las ruedas traseras, mientras la otra gira libre, para evitar el deslizamiento por diferencia de radios de giro, lo que implicaría una pérdida de potencia. La dirección del vehículo se desarrolló según un esquema de polígono deformable accionado por servo motor. El diseño de la geometría de dicho polígono deformable tiene en cuenta los diferentes radios de giro de las ruedas, verificando la confluencia de los nodos evitando de esa manera el deslizamiento de las ruedas sobre la superficie, para evitar pérdidas de rendimiento y disminución de la velocidad, según se puede observar en la Figura 1 a continuación:

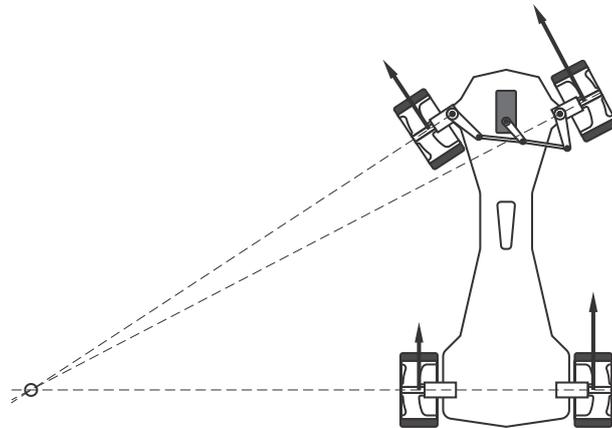


Figura 1. Esquema de confluencia de radios de curvatura.

La estructura se diseñó íntegramente en programas CAD, y fue impreso mediante una impresora 3D de depósito en capas de material plástico fundido.

Hardware

Sistema de control - Generalidades

El sistema de control está basado en una computadora de placa única Raspberry Pi modelo 2B, cuyas características más notables son:

- Microprocesador quad-core ARM Cortex-A7 de 900 MHz.
- 1 GB de memoria RAM.
- Puerto ethernet.



- 4 puertos USB.
- 40 GPIO (pines de propósito general de E/S).
- Interfaces para cámara de video y display.
- Soporta sistema operativo Linux. Particularmente se instaló la distribución Raspbian, que está basada en Debian.

El auto robot, al cual denominaremos *GreenBot*, recibe por un lado, información de 4 sensores infrarrojos colocados en el paragolpe delantero del chasis, los cuales permiten discriminar si el auto se encuentra sobre una superficie blanca o negra. Esta información ingresa a la Raspberry Pi (RPi) a través de los puertos GPIO. Por otro lado el auto también cuenta con una cámara de video conectada a la interfaz correspondiente.

Para controlar el motor se utiliza un puerto GPIO y se activa el módulo de PWM (Modulación por Ancho de Pulsos, en sus siglas en inglés), mediante el cual se puede variar la tensión de alimentación que recibe el motor, y por lo tanto su velocidad, de una manera muy práctica.

Para controlar el servomotor que comanda la dirección se utiliza otro puerto GPIO y otro módulo PWM.

De esta forma se puede controlar tanto velocidad como dirección de manera simultánea.

El auto robot posee adicionalmente un switch de encendido/apagado, y un pulsador que al ser presionado genera un retardo de 5 segundos y recién comienza a moverse el vehículo.

Teniendo la idea planteada de lo que se quería hacer se realizó una búsqueda de materiales y componentes necesarios. Posteriormente se hizo la compra de los mismos en el mercado local como a través de internet.

A modo de organizar se diferencian en el proyecto cuatro etapas donde se necesitó aplicar electrónica, las cuales son: **dirección**, **tracción**, **sensores** y **fuentes de alimentación**.

Dirección

Ante la necesidad de darle movimiento controlado a las ruedas delanteras del vehículo para que tuviera dirección al desplazarse, se eligió trabajar con un sistema de servomotor.

Un servomotor es un pequeño motor DC adosado a un sistema de engranajes, que aumentan su torque, junto con un circuito de control electrónico que posibilita controlar el ángulo de posicionamiento del eje del servomotor que a través de un brazo o palanca se logra tener un movimiento angular. Esto sirve así para mover las ruedas delanteras del vehículo.

Se eligió un modelo de servomotor SG90 debido a su bajo peso (11 gramos) y reducidas dimensiones. El mismo tiene 1,5 Kg de torque y se alimenta eléctricamente con tensión continua de 5 V, y puede observarse en la Figura 2.

Para controlar el ángulo de giro y posicionamiento del servomotor se debe aplicar una serie de pulsos eléctricos como se explica en la Figura 3.



Figura 2. Servomotor SG90.

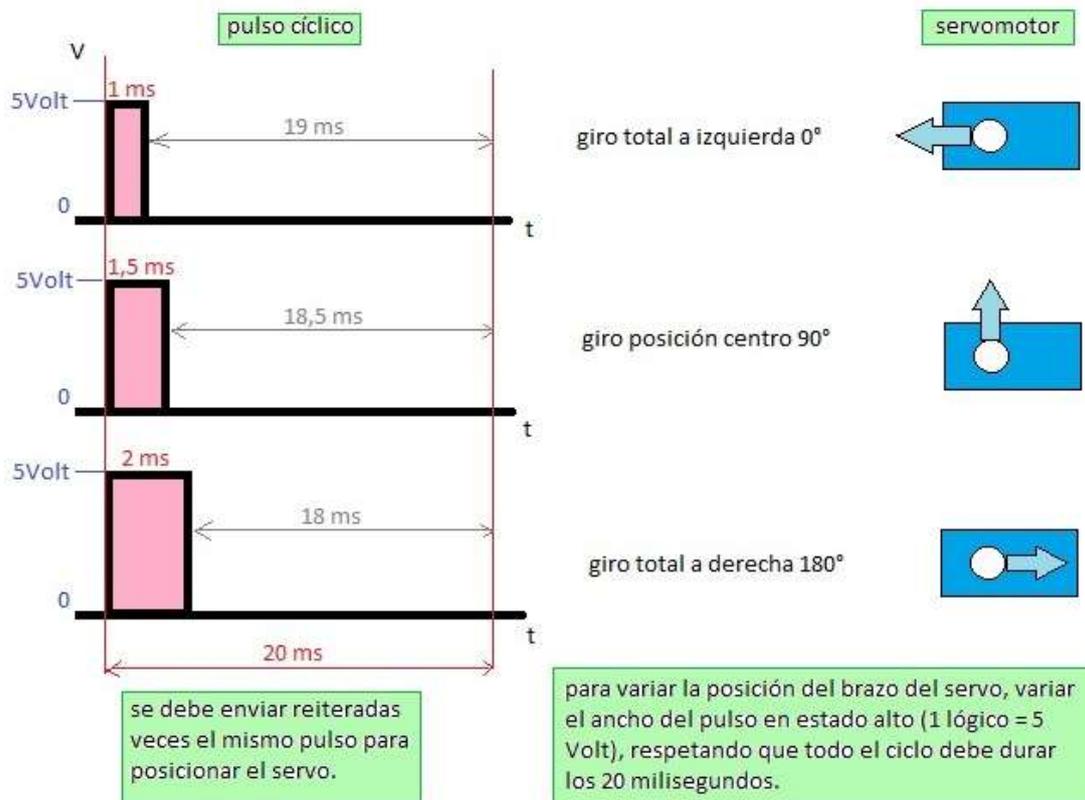


Figura 3. Tiempos para control del servomotor.

Tracción

Permitir que el vehículo avance con velocidad y agilidad fue el objetivo de esta etapa. Para ello se eligió darle movimiento a las ruedas traseras con un motor eléctrico DC (de corriente continua) acoplado por un sistema de engranajes al eje de ruedas traseras.

El motor DC se alimenta con tensión continua de 12 V y tiene un consumo aproximado en vacío de 70 mA.

Para poder controlar la velocidad de giro y también así el sentido del mismo a través de la programación de la placa Raspberry Pi se necesita en el medio una interfaz que permita adaptarlos. Para esto se requiere de un circuito denominado puente H, con el cual a través de una señal PWM se puede controlar la velocidad y también el sentido de giro en un sentido u otro.

Se decidió utilizar el modulo comercial de puente H L298 (que se muestra en la Figura 4) porque tiene reducidas dimensiones y cuenta con suficiente capacidad de entregar corriente a través de su salida disponible para el motor.



Figura 4. Puente H L298.

El esquema de conexión del puente H se muestra en la Figura 5. Desde el pin E1 se habilita el puente H poniendo a estado lógico alto o “1” desde el puerto I/O de la placa Raspberry Pi. Y configurando los pines de entrada 1,2, en estado lógico alto (“1”) uno y bajo (“0”) el otro se logra que a la salida Out1 del puente H donde se conecta el motor tenga giro en sentido horario o anti-horario dependiendo la configuración.

Por las entradas Vs y Gnd se alimenta todo el módulo, de aquí se tiene tanto la alimentación eléctrica que va al motor como la necesaria para el funcionamiento de la lógica del puente H.

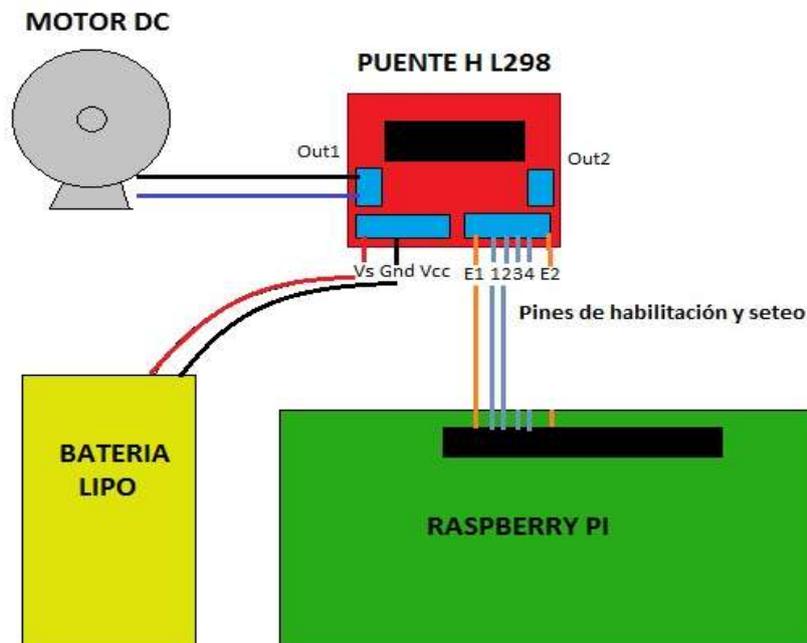


Figura 5. Diagrama de conexión de puente H.

Sensores

Como el vehículo debe poder reconocer una línea blanca sobre un fondo negro que le indica el camino a seguir, se decidió utilizar sensores infrarrojos enfocados al suelo para distinguir el color blanco de la línea del fondo negro de la pista.

Para alimentar y utilizar la información de los sensores en la placa Raspberry Pi se necesitó en el medio una interfaz. La cual se diseñó pensando que el puerto de entrada de la placa de desarrollo Raspberry admite como máximo una tensión de 3,3 volts equivalente a un estado lógico alto ("1").

El funcionamiento del sensor infrarrojo (IR) básicamente es por reflejo sobre la superficie, ya que el mismo va ubicado al ras (aprox. 6 mm de altura) del piso (pista) y cuenta con un diodo led emisor de luz infrarroja y un fototransistor receptor de luz infrarroja. El módulo de sensor comercial elegido es el CNY70 que ya cuenta con un emisor y receptor infrarrojos en su interior. Se muestra un esquema del mismo en la figura siguiente:

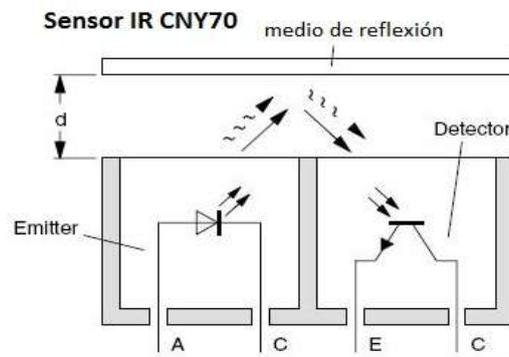


Figura 6. Esquema de funcionamiento del sensor infrarrojo CNY70.

El diodo led emite luz IR constantemente pero cuando ésta se refleja en una superficie blanca puede ser captada por el fototransistor receptor que conduce y se obtiene un nivel de tensión alto ("1" lógico) a su salida, caso contrario si está ubicado sobre superficie negra no refleja, porque dicho color absorbe la luz IR, y no conduce el fototransistor receptor generando un nivel de tensión bajo ("0" lógico).

También se eligió trabajar con una compuerta lógica Schmit trigger (disparador) 74HC14 para forzar las salidas de los 4 sensores en; ya sea estado bajo o alto dependiendo de q el sensor esté ubicado sobre superficie color blanca o negra. Esto para que no tener problemas con valores analógicos intermedios por variabilidad de tonos, luz, suciedad de la pista, etc. Esta compuerta es también inversora a su salida, o sea que al ingresar un estado lógico alto a la salida de ese canal tengo un estado lógico bajo y viceversa. La información obtenida de la salida de la compuerta se envía al puerto I/O de la placa Raspberry Pi para ser procesada. La conexión realizada se muestra en la Figura 7.

Como se desea contar con cuatro sensores para cubrir la pista a lo ancho y así detectar dónde está la línea blanca de 2 cm de ancho, q marca el camino a seguir, se dispusieron los sensores de forma transversal en frente del vehículo enfocando hacia abajo. Se fabricó una placa con toda la interfaz para enviar los datos a la placa Raspberry Pi. Su diagrama esquemático se muestra en la Figura 8.

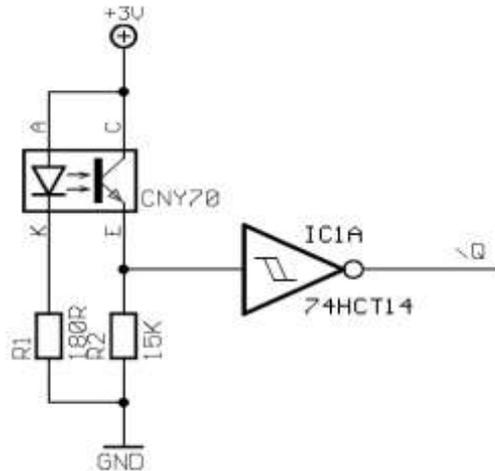


Figura 7. Esquemático de la conexión de un sensor infrarrojo.

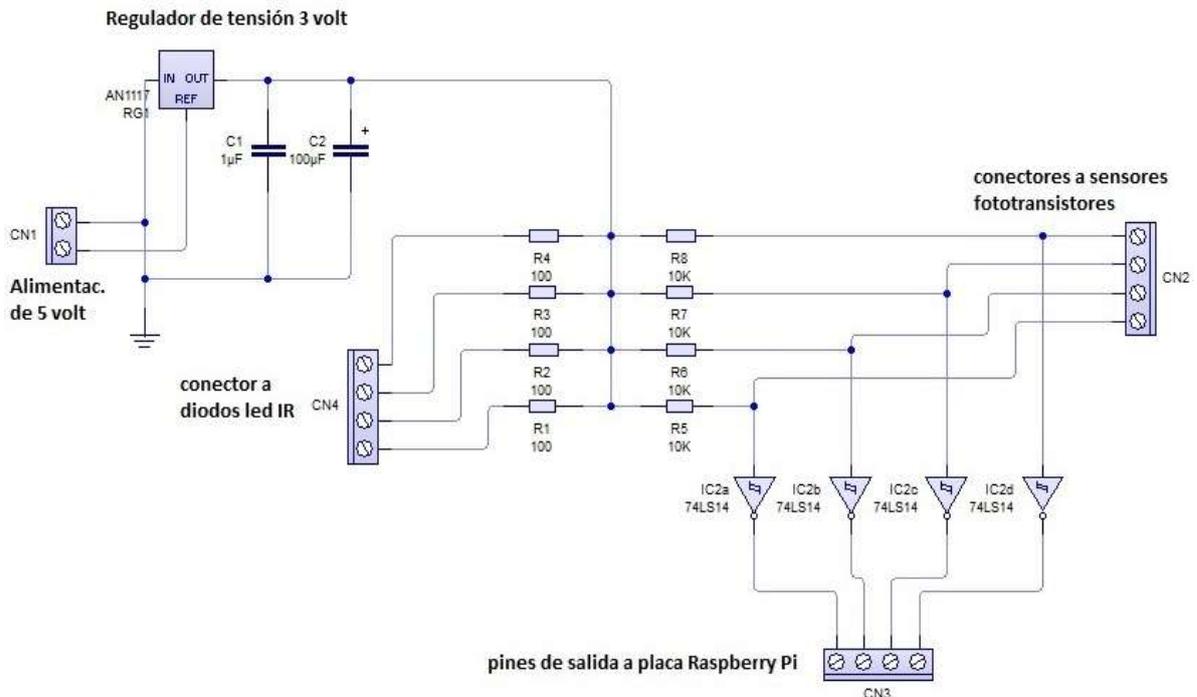


Figura 8. Esquemático completo de la interfaz de sensores.

Fuente de alimentación

Dado que el vehículo necesita alimentar eléctricamente y tener a disposición distintos niveles de tensión de acuerdo a los diseños de cada etapa y sus funciones, por lo que se tuvieron en cuenta los siguientes requerimientos:

- 5 V de tensión continua para la placa de control Raspberry Pi.
- 12 V de tensión continua para el motor de la tracción del vehículo.
- 3,3 V de tensión continua para alimentar sensores infrarrojos y su interface.
- 5 V de tensión continua para alimentar el servomotor de la dirección del

vehículo.

Se eligió para suministro de energía una batería tipo *Lipo* de 3 celdas, capaz de entregar 12 V, con una capacidad de corriente de 1500 mAh.

Se diseñó un circuito con un regulador de tensión LM7805, que justamente mantiene la tensión estable a su salida a un nivel de 5 volts de continua, y un Transistor de potencia 2SC5200 por el cual se provee la corriente necesaria de hasta 2 A como máximo disponible para el alto consumo de la placa de desarrollo Raspberry Pi. Esto último fue un gran inconveniente a resolver porque diseños anteriores fueron fallidos al verse superados en capacidad de corriente. A continuación se muestra el circuito esquemático:

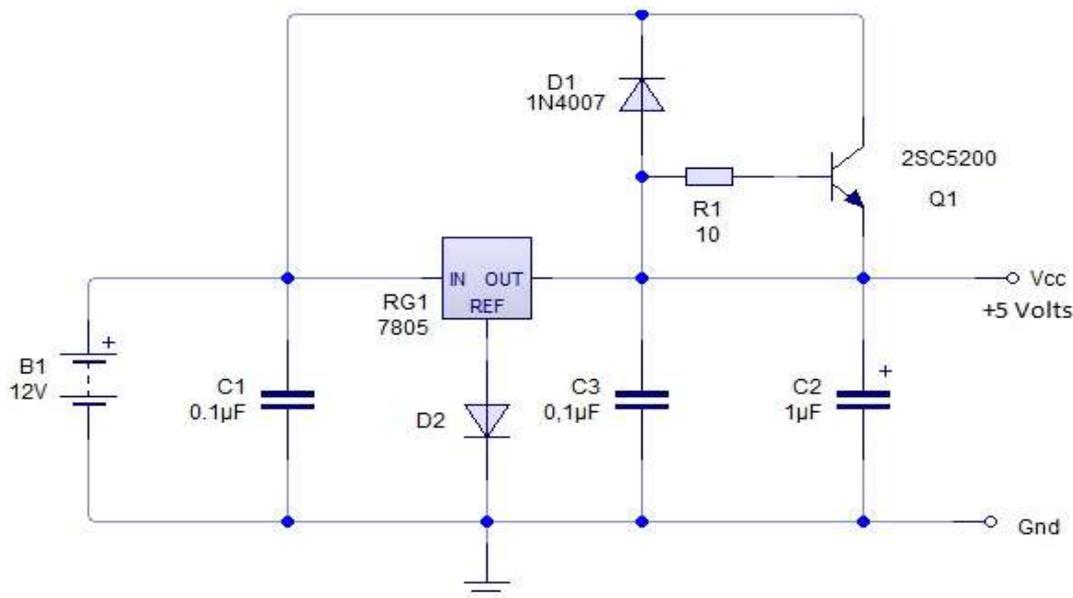


Figura 9. Circuito de fuente de alimentación regulada

Cabe destacar que del mismo circuito anterior se cuenta con la tensión continua de 5 volts necesaria para alimentar el servomotor de la dirección.

La tensión continua de 12 V de la batería *Lipo* son los adecuados para suministrar al motor DC de la tracción del vehículo, por lo que va directamente al módulo del puente H.

Se dispuso de otro regulador de tensión AN1117 que se alimenta con los 5 volts y a su salida entrega una tensión regulada de 3,3 volts continua para el funcionamiento del circuito de los sensores. Dicho circuito eléctrico puede observarse en la Figura 8 mostrada con anterioridad.

Todos estos circuitos fueron realizados y verificado su correcto funcionamiento de manera satisfactoria.

Software

Todo el código fue desarrollado en lenguaje Python, dado que la Raspberry Pi pone a disposición una muy completa librería en este lenguaje para el manejo de las entradas y salidas tanto de los puertos de propósito general como de la interfaz para la cámara. El código realizado está organizado en dos librerías, la primera de ellas destinada a la interfaz con el hardware, brindando tanto funciones para la inicialización y



configuración de los puertos como para la interacción con lo periféricos, es decir para leer datos desde los sensores, y enviar señales de control de velocidad y dirección. La segunda librería contiene todas las funciones necesarias para el control y toma de decisiones del vehículo. Finalmente, desde el programa principal del *GreenBot* se hace uso de las funciones de las librerías para poner en funcionamiento todo el sistema.

Descripción del software

El cuerpo principal del programa es un lazo que está permanentemente encuestando los sensores infrarrojos y llamando a la función de control para determinar el movimiento del vehículo.

Las funciones más importantes de la librería de interfaz con el hardware son las siguientes:

- **setDir(direccion):** esta función se encarga de enviar las señales correspondiente para el control del servomotor. Recibe un valor real que representa el ángulo que debe adoptar la dirección del vehículo.
- **setMot(velocidad):** controla la velocidad del motor de tracción.
- **start():** configuración, inicialización de los puertos GPIO de la Raspberry Pi.
- **marcha(sentido):** permite configurar el sentido de marcha del vehículo (marcha atrás y hacia adelante).
- **set(direccion,velocidad,sentido):** esta función permite controlar todos los parámetros anteriores.
- **getButton():** lee el estado de un pulsador, el cual funciona como botón de arranque.
- **getIR1(), getIR2(), getIR3() y getIR4():** estas funciones toman las lecturas de los sensores infrarrojos.
- **close():** detiene el vehículo y cierra los puertos del hardware.

La función más importantes de la librería de control es la siguiente:

- **direccion (IR1,IR2,IR3,IR4,oldDirec):** recibe como parámetro los valores leídos de los 4 sensores infrarrojos y la dirección anterior, y a partir de ellos se determina y devuelve el nuevo ángulo de giro que debe adoptar la dirección del vehículo para seguir la línea blanca.

Otras funciones implementadas son:

- **captura():** guarda en memoria una captura instantánea tomada por la cámara.
- **record():** guarda en memoria una serie de imágenes tomadas por la cámara con un determinado intervalo con el fin de usarlas para entrenar un red neuronal.
- **save_rec():** guarda en la tarjeta micro SD las imágenes capturadas por la función anterior.

Pruebas realizadas

Dado que uno de los objetivos propuestos fue participar en la Competencia Nacional de Robótica, las pruebas se realizaron sobre un circuito construido con características similares a las de la competencia, de acuerdo a la reglamentación de la misma.

Entre las características más relevantes del circuito se encuentran las siguientes:

- Ancho total de la pista de 40 cm (dos carriles de 20 cm cada uno).
- Línea blanca de seguimiento de 2 cm de ancho.

- Curvatura mínima de 20 cm de radio, y máxima de 35 cm de radio.
- Curvas consecutivas con cambio de sentido sin tramos rectos intermedios.
- Pendientes máximas de 20°.

En la Figura 10 puede observarse una sección del circuito de pruebas construido, en la que se distinguen las dimensiones mencionadas y las curvas consecutivas con cambio de sentido.



Figura 10. Sector del circuito de prueba

Las pruebas incluyeron hacer que el robot recorra el circuito con diferentes velocidades, en ambos sentidos, verificando su estabilidad y su capacidad de mantenerse sobre la línea sin muchas oscilaciones, tanto en recta como en curva. Por otra parte durante el recorrido se fueron tomando imágenes del circuito al tiempo que se guardaron los datos de las respuestas de los sistemas de control asociados a cada situación, que se pretenden utilizar como datos de entrada y salida para un entrenamiento eficaz de la red neuronal.

Resultados y discusión

Las pruebas realizadas permitieron mejorar características de diseño y de funcionamiento del vehículo, tanto de las partes mecánicas, como de las electrónicas y de software. Entre los resultados más destacados se pueden mencionar los siguientes:

- En las primeras versiones la distancia entre ejes no permitía un radio de giro apropiado. Esta distancia fue reducida en versiones posteriores lo cual otorgó un mayor rango de radios de curvatura, permitiendo un control más versátil del vehículo.
- Durante las pruebas se hizo mover el vehículo en forma continua y sin inconvenientes por el circuito con velocidades del orden de los 0.4 m/s. A mayores velocidades en rectas no presentó ningún inconveniente, pero se notaron tendencias a la pérdida del seguimiento en curvas, con oscilaciones de corrección pronunciadas, y hasta la pérdida total del seguimiento. Se buscará optimizar el código de control de los sensores con la finalidad de mejorar la respuesta en curvas y así poder aumentar la velocidad.
- Se pudo notar que en las rectas se podría aumentar aún más la velocidad sin que ello implique pérdidas de control, por lo que se pretende incorporar un sistema que permita la detección a distancia de inicio y final de curvas para así poder variar la velocidad del vehículo en los diferentes tramos del circuito.
- Con esta finalidad, la incorporación de datos de imágenes mediante la cámara

en conjunto con la red neuronal permitirá la detección a distancia de inicios y finales de curva.

En la Figura 11 se puede observar una imagen completa del último prototipo realizado.

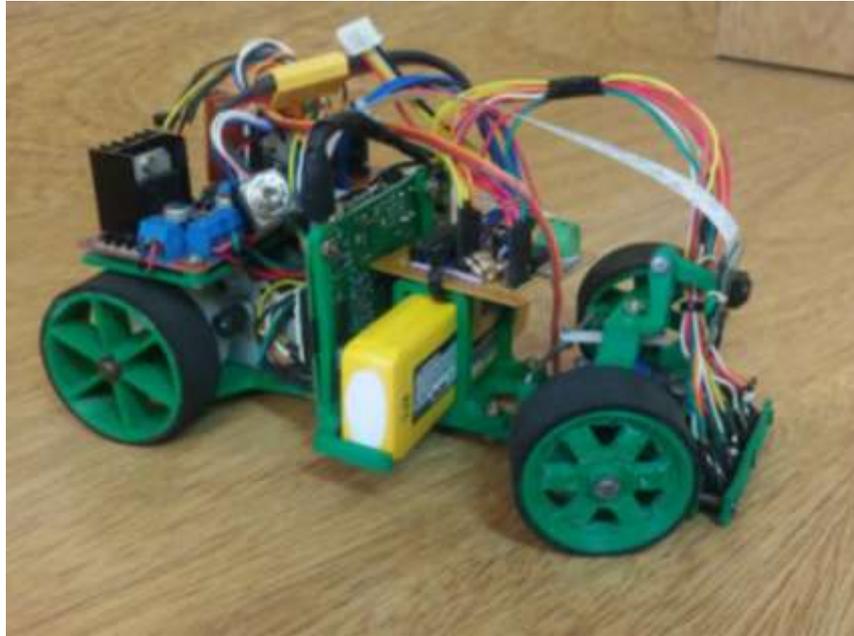


Figura 11. Imagen del último prototipo realizado.

Conclusiones

Las conclusiones de los trabajos realizados son las siguientes:

1. Se diseñaron y construyeron diversos prototipos de vehículos robóticos que cumplen con los requisitos propuestos.
2. Las pruebas realizadas sobre el último prototipo mostraron que el diseño propuesto recorre eficientemente el circuito sin perder el seguimiento de la línea y con oscilaciones mínimas, a velocidades no superiores a 0.4 m/s en curvas.
3. La reducción de la distancia entre ejes mejoró la respuesta del seguimiento en curvas, especialmente en las de menor radio de curvatura.
4. El equipo se inscribió y asistió a la XIII Competencia Nacional de Robótica, realizada en la ciudad de Bahía Blanca en el mes de Noviembre de 2015, aunque finalmente no se pudo competir por inconvenientes con el hardware que no pudieron solucionarse por falta de disponibilidad de repuestos adecuados.
5. Se cuenta con un conjunto de librerías de software que permiten el control del vehículo de forma sencilla y la implementación de algoritmos inteligentes.

Como posible mejora futura, se hace necesaria la incorporación de un sistema de detección de entradas y salidas de curvas para un mejor aprovechamiento de la velocidad final del auto en rectas. Dado que las velocidades teóricas que se pueden alcanzar son elevadas, el sistema deberá detectar el inicio de curvas con antelación, otorgando el espacio necesario para la desaceleración del vehículo.

Por otra parte, se debe incorporar nuevas funciones de control del vehículo que utilicen diferentes algoritmos de inteligencia artificial.