Proyecto

“Chola”

****

Aguilera Berrizbeitia, Eduardo Alonso

Tamiche Guzmán, Brayan José

Fabriani Brito, Alessandro Francescho

**INTRODUCCIÓN**

En la era actual, caracterizada por avances tecnológicos exponenciales, la robótica emerge como una disciplina fundamental y transversal, fusionando conocimientos de la mecánica, la electrónica, la informática y la inteligencia artificial. Su impacto se extiende a innumerables sectores, transformando la industria, la medicina, la exploración espacial e incluso nuestra vida cotidiana. En este contexto dinámico, la innovación constante y la colaboración multidisciplinaria se alzan como pilares insustituibles para el desarrollo de soluciones ingeniosas y eficientes que respondan a las crecientes demandas de la sociedad. Este documento se dedica a presentar en detalle el proyecto del robot con nombre “Chola” realizado por el equipo **DB Engineers**, una agrupación de jóvenes conformada por **Brayan Tamiche, Eduardo Aguilera y Alessandro Fabrianni**. Nuestra pasión, dedicación incansable y destacada habilidad técnica se han conjugado para materializar la creación de un sofisticado robot autónomo, meticulosamente construido con el propósito de superar los complejos retos planteados en la prestigiosa competencia internacional World Robot Olympiad (WRO), específicamente dentro de la exigente categoría de Futuros Ingenieros.

La World Robot Olympiad (WRO) no es simplemente una competencia; se consolida año tras año como una plataforma global de invaluable trascendencia, diseñada para encender la curiosidad científica y tecnológica en jovenes de todo el mundo. Su misión principal es fomentar y cultivar habilidades esenciales para el siglo XXI, como la creatividad, el pensamiento crítico, la capacidad analítica y las habilidades para la resolución de problemas complejos. A través de un abanico de diversas categorías, la WRO inspira a sus participantes a sumergirse en el fascinante universo de la ingeniería y la programación, impulsándolos a explorar los límites del conocimiento actual y a desarrollar soluciones innovadoras, preparándolos para afrontar con éxito los retos tecnológicos del futuro. y para convertirse en los futuros líderes de la innovación.

El presente informe tiene como objetivo principal desglosar, con un alto grado de meticulosidad y claridad, cada una de las etapas que comprendieron el proceso de diseño, la construcción y la crucial programación del robot desarrollado por DB Engineers. Se explorarán las estrategias conceptualizadas e implementadas por muestro equipo para abordar los dos exigentes retos que definen su categoría. Estos retos involucran, por un lado, la navegación autónoma precisa y eficiente a través de una pista con una configuración de obstáculos variable y, por otro lado, la capacidad del robot para interactuar de manera inteligente con un sistema de señales de tráfico, lo que implica la toma de decisiones críticas en tiempo real basadas en la percepción sensorial. El recorrido descriptivo abarcará desde la concepción mecánica inicial del robot, detallando la selección de materiales y la cinemática del diseño, pasando por la cuidadosa selección e integración de los diversos componentes electrónicos –sensores, actuadores y unidades de procesamiento–, hasta llegar al desarrollo del software que constituye el "cerebro" del robot, dotándolo de la inteligencia y la autonomía necesarias para operar en el entorno de la competencia. Cada fase del proyecto, incluyendo los obstáculos encontrados y las soluciones implementadas, será expuesta con la mayor transparencia y detalle técnico. La finalidad última es ofrecer una visión integral, exhaustiva del arduo trabajo realizado por el equipo DB Engineers, destacando no solo el resultado técnico, sino también el espíritu colaborativo y nuestro enfoque innovador para "conectar el mundo" a través de las infinitas posibilidades que ofrece la robótica.



Al principio eran únicamente Brayan y Alessandro, luego se unió Eduardo conformando un grupo de tres amigos, desde el principio, DB Engineers se comprometió con un enfoque estructurado y, priorizando la innovación y el trabajo en equipo eficaz. Si bien algunos miembros poseían conocimientos fundamentales en ingeniería y programación, el equipo adoptó una filosofía de aprendizaje y desarrollo continuo. Este compromiso los ha impulsado a diseñar y construir soluciones robóticas sofisticadas.

Formación de equipo

**Eduardo**

**Aguilera**



**Brayan**

**Tamiche**



**Alessandro**

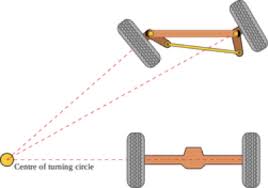
**Fabrianni**

Proceso de Construcción

2.1 Sistema de dirección

Para un sistema de dirección de robot con un servo MG995 y un mecanismo Ackerman ajustable, el servo MG995 estaría conectado al pin PWM 3. Este pin es crucial porque permite controlar la posición del servo al variar el ancho del pulso eléctrico que se le envía. Al ajustar el ancho del pulso desde un microcontrolador, se le indica al servo el ángulo exacto al que debe girar, lo que a su vez permite la dirección precisa del mecanismo Ackerman del robot.





Nuestra inspiración para este vehículo autónomo viene directamente de los avances en vehículos autónomos de China. Ver cómo integran flotas de taxis y autobuses autónomos con tecnología innovadora en ciudades como Beijing y Shanghái nos demostró la viabilidad de un sistema de dirección preciso y adaptable. Esto nos llevó a desarrollar nuestro propio sistema con un servo MG995 y un mecanismo Ackerman ajustable.

k

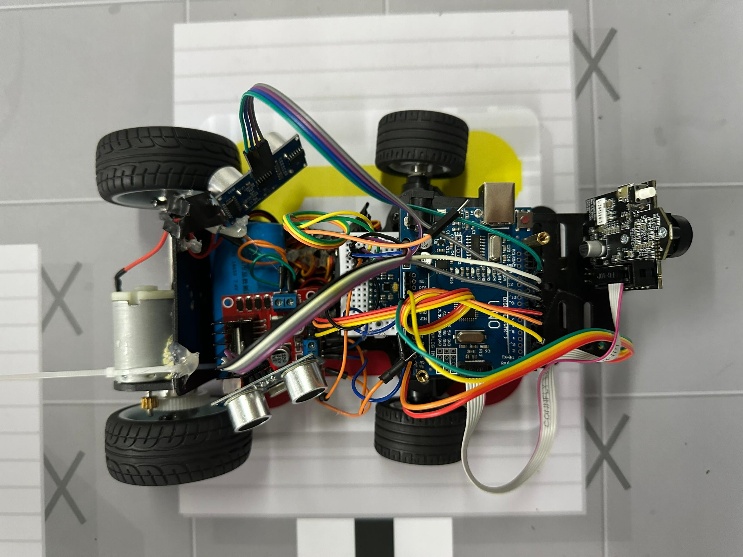
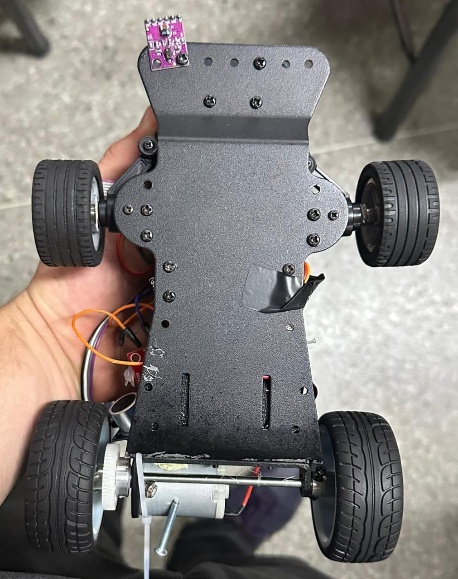
2.2 Inspiracion

A continuación, se describirá el proceso detallado del ensamblado, los componentes (mecánicos y electrónicos) y la programación de nuestro robot ❝Chola❞

2.3 Proceso del diseño del chasis

Al Principio íbamos a usar un sistema de kit de carro robótico proveniente desde china, pero encontramos varios problemas como su tamaño y gracias a esto a nuestro robot no tenía el rango de giro que nosotros deseábamos y se le dificultaba las curvas y esquivar los semáforos, por lo consiguiente tuvimos que buscar una solución distinta

k



Nuestra solución fue agarrar 10 cm de la parte de atrás del chasis y doblarlo 90 grados, esto hizo que se reduzca el tamaño del robot y nos ayudo con el soporte del motor.

Luego de un mar de ideas, llegamos a la conclusión que la mejor manera de que nuestro chasis cumpla nuestras expectativas, es reduciéndolo, esto nos va a beneficiar en cuanto a Angulo de giro(es más ágil y maniobrable en los espacios reducidos de la competencia. Además, es más económico de construir y mantener, usa menos energía, y es más fácil de transportar y manejar).

2.4 Diseño de la electrónica

Análisis del Diagrama del Circuito (Nombres de Pines Corregidos)

Basado en el diagrama de circuito proporcionado, y corrigiendo los nombres de los pines para la comunicación I2C:

1. Suministro de Energía (P1, M1, P1+)

P1 (Símbolo de Batería): Tu fuente de alimentación principal, probablemente una batería, conectada al motor M1.

M1 (Motor): Un motor de corriente continua (DC). Está directamente conectado a la batería P1.

P1+ (Bus de Energía): Parece ser un riel de energía positiva común, que se origina en la batería.

2. Arduino UNO (U1)

U1 (Arduino UNO): El microcontrolador central.

Pin 5V: Conectado a U1\_5V, que es un bus de energía común de 5V.

Pin GND: Conectado a GND, que es un bus de tierra común.

Pines Digitales (D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13): Estos están conectados a varios sensores y al servo.

Pines Analógicos (A0, A1, A2, A3, A4, A5):

A4 (SDA): Generalmente usado para la línea de datos I2C.

A5 (SCL): Generalmente usado para la línea de reloj I2C.

3. Servomotor (SERVO1)

SERVO1: Un servomotor de hobby estándar.

PWR: Conectado a P1+ (positivo de la batería principal). Esto es una observación crítica y una buena práctica. Significa que el servo se alimenta directamente de la batería principal, no del pin de 5V del Arduino. Esto es ideal para servos más grandes que consumen más corriente de la que el regulador de 5V del Arduino puede suministrar de forma segura.

SIG (Señal): Conectado al Pin Digital D9 del Arduino.

GND: Conectado al riel común GND.

4. Controlador de Motor (U2 - L293D o similar Puente H)

U2 (L293D / Controlador Puente H): Este chip se utiliza para controlar el motor de CC M1. El L293D es un controlador de puente H común.

VCC1 (VCC Lógico): Conectado a U1\_5V (suministro de 5V del Arduino). Esto alimenta la parte lógica del controlador.

VCC2 (VCC del Motor): Conectado a P1+ (positivo de la batería principal). Esto alimenta la parte del controlador que acciona el motor.

1,2EN (Habilitar 1,2): Conectado al Pin Digital D2 del Arduino. Esto habilita o deshabilita las salidas del motor para el canal 1 (1A, 2A). Típicamente, este pin se usa para PWM para controlar la velocidad del motor.

1A: Conectado al Pin Digital D3 del Arduino. Esta es una entrada para el motor 1.

2A: Conectado al Pin Digital D4 del Arduino. Esta es una entrada para el motor 1.

3A: Conectado al Pin Digital D5 del Arduino. Esta es una entrada para el motor 2 (aunque solo se muestra un motor M1).

4A: Conectado al Pin Digital D6 del Arduino. Esta es una entrada para el motor 2.

1Y: Conectado a un terminal del motor M1.

2Y: Conectado al otro terminal del motor M1.

GND1, GND2, GND3, GND4: Todos conectados al riel común GND.

Observaciones: El L293D tiene dos canales de motor. Aquí, solo un motor (M1) se muestra conectado a 1Y y 2Y. Las entradas 3A y 4A están conectadas a D5 y D6 respectivamente, pero no se dibuja explícitamente un segundo motor conectado a 3Y y 4Y. Esto sugiere que se implica un segundo motor, o que solo se usa un canal y el diagrama muestra conexiones redundantes para las entradas del segundo canal.

5. Sensores de Ultrasonido y Sensores I2C

Sensores de Ultrasonido (DIST1):

DIST1:

TRIG: Conectado al Pin Digital D7 del Arduino.

ECHO: Conectado al Pin Digital D8 del Arduino.

VCC & GND: Conectados a U1\_5V y GND respectivamente, lo que significa que se alimentan de los 5V del Arduino.

Sensores I2C (DIST2, Pixy Cam, Sensor de Color, Giroscopio):

VCC & GND: Todos conectados a U1\_5V y GND respectivamente, lo que significa que se alimentan de los 5V del Arduino.

DIST2 (Ultrasonido):

TRIG: Conectado al Pin Digital D10 del Arduino.

ECHO: Conectado al Pin Digital D11 del Arduino.

Observación: Este sigue siendo un sensor de ultrasonido.

Pixy Cam (previamente DISTPIXY CAM):

SDA: Conectado al Pin Digital D12 del Arduino.

SCL: Conectado al Pin Digital D13 del Arduino.

Observación: La Pixy Cam usualmente usa SPI (D10, D11, D12, D13, donde D10 es SS), pero también puede comunicarse por I2C. Si usa I2C, lo más común es que se conecte a A4 (SDA) y A5 (SCL) del Arduino, que son los pines I2C hardware dedicados. Conectarlo a D12 y D13 para I2C requeriría I2C por software (SoftwareWire), o podría ser una conexión SPI donde D12/D13 son MISO/SCK/MOSI. Si es I2C, esto es una conexión inusual.

Sensor de Color (previamente DISTSENSOR COLOR, asumiendo TCS3200):

SDA: Conectado al Pin Analógico A0 del Arduino.

SCL: Conectado al Pin Analógico A1 del Arduino.

Observación: El sensor TCS3200 NO usa I2C. Utiliza pines digitales para S0-S3 y un pin digital para la salida de frecuencia (OUT). Si este es un TCS3200, entonces A0 y A1 no son SDA/SCL, sino S2/S3 (o S0/S1/OUT), lo cual es una conexión incorrecta para I2C y no usaría el nombre SDA/SCL. Si es otro tipo de sensor de color que sí usa I2C (como algunos sensores de color con chip I2C), entonces la conexión a A0/A1 como SDA/SCL es igualmente inusual para I2C hardware, pero A0/A1 pueden ser usados como I2C por software.

Giroscopio (previamente DISTGIROSCOPIO, asumiendo MPU-6050/9250):

SDA: Conectado al Pin Analógico A2 del Arduino.

SCL: Conectado al Pin Analógico A3 del Arduino.

Observación: Los MPU-6050/9250 usan I2C. Sin embargo, al igual que con el sensor de color, conectarlos a A2/A3 como SDA/SCL para I2C hardware es inusual. Los pines I2C hardware son A4/A5. Esto implicaría el uso de I2C por software en A2/A3, o una confusión en el cableado o la nomenclatura.

6. Arquitectura General:

El circuito utiliza una fuente de alimentación separada para el motor de CC principal (M1) y el servo (SERVO1), lo que es un excelente diseño para la gestión de la corriente.

El Arduino se alimenta externamente (a través de su propio jack de alimentación o USB, no mostrado, pero implícito ya que U1\_5V se deriva del propio Arduino).

El controlador L293D maneja el motor.

Hay un sensor de ultrasonido claramente identificado (DIST1). El DIST2 también parece ser un ultrasonido.

Las correcciones sugieren que la Pixy Cam, el Sensor de Color y el Giroscopio se comunican a través de I2C. Sin embargo, la asignación de pines (D12/D13 para Pixy Cam, A0/A1 para Sensor de Color, A2/A3 para Giroscopio) para I2C es inusual para la comunicación I2C por hardware del Arduino (que es A4/A5). Esto podría implicar el uso de I2C por software, o una interpretación errónea de los pines del diagrama.

Puntos a Considerar para la Implementación (debido a las correcciones):

Pixy Cam: Si realmente se comunica por I2C, D12/D13 no son los pines I2C estándar. Si usa SPI, entonces D10/D11/D12/D13 son los pines correctos para SPI (con D10 como SS).

Sensor de Color (TCS3200): Este sensor NO usa I2C. Usa 4 pines digitales para control (S0-S3) y 1 pin digital para la salida de frecuencia (OUT). Las conexiones A0/A1 para "SDA/SCL" serían incorrectas para este sensor.

Giroscopio (MPU-6050/9250): Estos SÍ usan I2C. Pero nuevamente, A2/A3 no son los pines I2C estándar (A4/A5). Si se usan A2/A3 como I2C, requeriría la librería SoftwareWire.

Múltiples I2C: Si todos los "sensores I2C" se conectan al mismo bus I2C (SDA y SCL), deberían compartir los mismos pines A4 y A5 en el Arduino, y cada dispositivo debería tener una dirección I2C única. La forma en que están conectados en el diagrama (a diferentes pares de pines) sugiere que no están en el mismo bus I2C estándar, o que la nomenclatura es confusa.

3. Servomotor (SERVO1)

SERVO1: Un servomotor de hobby estándar.

PWR: Conectado a P1+ (positivo de la batería principal). Esto es una observación crítica y una buena práctica. Significa que el servo se alimenta directamente de la batería principal, no del pin de 5V del Arduino. Esto es ideal para servos más grandes que consumen más corriente de la que el regulador de 5V del Arduino puede suministrar de forma segura.

SIG (Señal): Conectado al Pin Digital D9 del Arduino.

GND: Conectado al riel común GND.

4. Controlador de Motor (U2 - L293D o similar Puente H)

U2 (L293D / Controlador Puente H): Este chip se utiliza para controlar el motor de CC M1. El L293D es un controlador de puente H común.

**VCC1 (VCC Lógico)**: Conectado a U1\_5V (suministro de 5V del Arduino). Esto alimenta la parte lógica del controlador.

**VCC2 (VCC del Motor)**: Conectado a P1+ (positivo de la batería principal). Esto alimenta la parte del controlador que acciona el motor.

1,2EN (Habilitar 1,2): Conectado al Pin Digital D2 del Arduino. Esto habilita o deshabilita las salidas del motor para el canal 1 (1A, 2A). Típicamente, este pin se usa para PWM para controlar la velocidad del motor.

**1A**: Conectado al Pin Digital D3 del Arduino. Esta es una entrada para el motor 1.

**2A**: Conectado al Pin Digital D4 del Arduino. Esta es una entrada para el motor 1.

**3A**: Conectado al Pin Digital D5 del Arduino. Esta es una entrada para el motor 2 (aunque solo se muestra un motor M1).

**4A**: Conectado al Pin Digital D6 del Arduino. Esta es una entrada para el motor 2.

1Y: Conectado a un terminal del motor M1.

**2Y**: Conectado al otro terminal del motor M1.

GND1, GND2, GND3, GND4: Todos conectados al riel común GND.

Observaciones: El L293D tiene dos canales de motor. Aquí, solo un motor (M1) se muestra conectado a 1Y y 2Y. Las entradas 3A y 4A están conectadas a D5 y D6 respectivamente, pero no se dibuja explícitamente un segundo motor conectado a 3Y y 4Y. Esto sugiere que se implica un segundo motor, o que solo se usa un canal y el diagrama muestra conexiones redundantes para las entradas del segundo canal.

DIST2 (Ultrasonido):

TRIG: Conectado al Pin Digital D10 del Arduino.

ECHO: Conectado al Pin Digital D11 del Arduino.

Observación: Este sigue siendo un sensor de ultrasonido.

Pixy Cam (previamente DISTPIXY CAM):

SDA: Conectado al Pin Digital D12 del Arduino.

SCL: Conectado al Pin Digital D13 del Arduino.

Observación: La Pixy Cam usualmente usa SPI (D10, D11, D12, D13, donde D10 es SS), pero también puede comunicarse por I2C. Si usa I2C, lo más común es que se conecte a A4 (SDA) y A5 (SCL) del Arduino, que son los pines I2C hardware dedicados. Conectarlo a D12 y D13 para I2C requeriría I2C por software (SoftwareWire), o podría ser una conexión SPI donde D12/D13 son MISO/SCK/MOSI. Si es I2C, esto es una conexión inusual.

Sensor de Color (previamente DISTSENSOR COLOR, asumiendo TCS3200):

SDA: Conectado al Pin Analógico A0 del Arduino.

SCL: Conectado al Pin Analógico A1 del Arduino.

Observación: El sensor TCS3200 NO usa I2C. Utiliza pines digitales para S0-S3 y un pin digital para la salida de frecuencia (OUT). Si este es un TCS3200, entonces A0 y A1 no son SDA/SCL, sino S2/S3 (o S0/S1/OUT), lo cual es una conexión incorrecta para I2C y no usaría el nombre SDA/SCL. Si es otro tipo de sensor de color que sí usa I2C (como algunos sensores de color con chip I2C), entonces la conexión a A0/A1 como SDA/SCL es igualmente inusual para I2C hardware, pero A0/A1 pueden ser usados como I2C por software.

Giroscopio (previamente DISTGIROSCOPIO, asumiendo MPU-6050/9250):

SDA: Conectado al Pin Analógico A2 del Arduino.

SCL: Conectado al Pin Analógico A3 del Arduino.

Observación: Los MPU-6050/9250 usan I2C. Sin embargo, al igual que con el sensor de color, conectarlos a A2/A3 como SDA/SCL para I2C hardware es inusual. Los pines I2C hardware son A4/A5. Esto implicaría el uso de I2C por software en A2/A3, o una confusión en el cableado o la nomenclatura.

6. Arquitectura General:

El circuito utiliza una fuente de alimentación separada para el motor de CC principal (M1) y el servo (SERVO1), lo que es un excelente diseño para la gestión de la corriente.

El Arduino se alimenta externamente (a través de su propio jack de alimentación o USB, no mostrado, pero implícito ya que U1\_5V se deriva del propio Arduino).

El controlador L293D maneja el motor.

Hay un sensor de ultrasonido claramente identificado (DIST1). El DIST2 también parece ser un ultrasonido.

Las correcciones sugieren que la Pixy Cam, el Sensor de Color y el Giroscopio se comunican a través de I2C. Sin embargo, la asignación de pines (D12/D13 para Pixy Cam, A0/A1 para Sensor de Color, A2/A3 para Giroscopio) para I2C es inusual para la comunicación I2C por hardware del Arduino (que es A4/A5). Esto podría implicar el uso de I2C por software, o una interpretación errónea de los pines del diagrama.

Puntos a Considerar para la Implementación (debido a las correcciones):

Pixy Cam: Si realmente se comunica por I2C, D12/D13 no son los pines I2C estándar. Si usa SPI, entonces D10/D11/D12/D13 son los pines correctos para SPI (con D10 como SS).

Sensor de Color (TCS3200): Este sensor NO usa I2C. Usa 4 pines digitales para control (S0-S3) y 1 pin digital para la salida de frecuencia (OUT). Las conexiones A0/A1 para "SDA/SCL" serían incorrectas para este sensor.

Giroscopio (MPU-6050/9250): Estos SÍ usan I2C. Pero nuevamente, A2/A3 no son los pines I2C estándar (A4/A5). Si se usan A2/A3 como I2C, requeriría la librería SoftwareWire.

Múltiples I2C: Si todos los "sensores I2C" se conectan al mismo bus I2C (SDA y SCL), deberían compartir los mismos pines A4 y A5 en el Arduino, y cada dispositivo debería tener una dirección I2C única. La forma en que están conectados en el diagrama (a diferentes pares de pines) sugiere que no están en el mismo bus I2C estándar, o que la nomenclatura es confusa.

Sensores de Ultrasonido (DIST1):

DIST1:

**TRIG**: Conectado al Pin Digital D7 del Arduino.

**ECHO**: Conectado al Pin Digital D8 del Arduino.

**VCC & GND**: Conectados a U1\_5V y GND respectivamente, lo que significa que se alimentan de los 5V del Arduino.

Sensores I2C (DIST2, Pixy Cam, Sensor de Color, Giroscopio):

**VCC & GND**: Todos conectados a U1\_5V y GND respectivamente, lo que significa que se alimentan de los 5V del Arduino.

DIST2 (Ultrasonido):

**TRIG**: Conectado al Pin Digital D10 del Arduino.

**ECHO**: Conectado al Pin Digital D11 del Arduino.

Observación: Este sigue siendo un sensor de ultrasonido.

Pixy Cam (previamente DISTPIXY CAM):

**SDA**: Conectado al Pin Digital D12 del Arduino.

**SCL:** Conectado al Pin Digital D13 del Arduino.

Observación: La Pixy Cam usualmente usa SPI (D10, D11, D12, D13, donde D10 es SS), pero también puede comunicarse por I2C. Si usa I2C, lo más común es que se conecte a A4 (SDA) y A5 (SCL) del Arduino, que son los pines I2C hardware dedicados. Conectarlo a D12 y D13 para I2C requeriría I2C por software (SoftwareWire), o podría ser una conexión SPI donde D12/D13 son MISO/SCK/MOSI. Si es I2C, esto es una conexión inusual.

Sensor de Color (previamente DISTSENSOR COLOR, asumiendo TCS3200):

**SDA**: Conectado al Pin Analógico A0 del Arduino.

**SCL**: Conectado al Pin Analógico A1 del Arduino.

Observación: El sensor TCS3200 NO usa I2C. Utiliza pines digitales para S0-S3 y un pin digital para la salida de frecuencia (OUT). Si este es un TCS3200, entonces A0 y A1 no son SDA/SCL, sino S2/S3 (o S0/S1/OUT), lo cual es una conexión incorrecta para I2C y no usaría el nombre SDA/SCL. Si es otro tipo de sensor de color que sí usa I2C (como algunos sensores de color con chip I2C), entonces la conexión a A0/A1 como SDA/SCL es igualmente inusual para I2C hardware, pero A0/A1 pueden ser usados como I2C por software.

Giroscopio (previamente DISTGI

Giroscopio (previamente DISTGIROSCOPIO, asumiendo MPU-6050/9250):

**SDA:** Conectado al Pin Analógico A2 del Arduino.

**SCL:** Conectado al Pin Analógico A3 del Arduino.

Observación: Los MPU-6050/9250 usan I2C. Sin embargo, al igual que con el sensor de color, conectarlos a A2/A3 como SDA/SCL para I2C hardware es inusual. Los pines I2C hardware son A4/A5. Esto implicaría el uso de I2C por software en A2/A3, o una confusión en el cableado o la nomenclatura.

6. Arquitectura General:

El circuito utiliza una fuente de alimentación separada para el motor de CC principal (M1) y el servo (SERVO1), lo que es un excelente diseño para la gestión de la corriente.

El Arduino se alimenta externamente (a través de su propio jack de alimentación o USB, no mostrado, pero implícito ya que U1\_5V se deriva del propio Arduino).

El controlador L293D maneja el motor.

Hay un sensor de ultrasonido claramente identificado (DIST1). El DIST2 también parece ser un ultrasonido.

Las correcciones sugieren que la Pixy Cam, el Sensor de Color y el Giroscopio se comunican a través de I2C. Sin embargo, la asignación de pines (D12/D13 para Pixy Cam, A0/A1 para Sensor de Color, A2/A3 para Giroscopio) para I2C es inusual para la comunicación I2C por hardware del Arduino (que es A4/A5). Esto podría implicar el uso de I2C por software, o una interpretación errónea de los pines del diagrama.

Puntos a Considerar para la Implementación (debido a las correcciones):

Pixy Cam: Si realmente se comunica por I2C, D12/D13 no son los pines I2C estándar. Si usa SPI, entonces D10/D11/D12/D13 son los pines correctos para SPI (con D10 como SS).

Sensor de Color (TCS3200): Este sensor NO usa I2C. Usa 4 pines digitales para control (S0-S3) y 1 pin digital para la salida de frecuencia (OUT). Las conexiones A0/A1 para "SDA/SCL" serían incorrectas para este sensor.

Giroscopio (MPU-6050/9250): Estos SÍ usan I2C. Pero nuevamente, A2/A3 no son los pines I2C estándar (A4/A5). Si se usan A2/A3 como I2C, requeriría la librería SoftwareWire.

Múltiples I2C: Si todos los "sensores I2C" se conectan al mismo bus I2C (SDA y SCL), deberían compartir los mismos pines A4 y A5 en el Arduino, y cada dispositivo debería tener una dirección I2C única. La forma en que están conectados en el diagrama (a diferentes pares de pines) sugiere que no están en el mismo bus I2C estándar, o que la nomenclatura es confusa.

2.5 Sensores

HC-SR04



Nuestro robot usa tres sensores de ultrasonido, los HC-SR04 al frente y a los costados para medir distancias a las paredes y ajustar su dirección. Este sensor es muy preciso, detectando de 2 a 450 cm.

**450 centímetros**.



Al recibir una señal en Trig, emite pulsos ultrasónicos; el pin Echo mide el tiempo que tarda el eco en regresar, y el microcontrolador usa ese tiempo para calcular la distancia.

**Este sensor posee cuatro terminales que permiten su funcionamiento:**

* **VCC:** (Voltaje de Corriente Continua) Es la entrada de alimentación del sensor y requiere 5v para operar.
* **Trig:** (Disparador) Su función es enviar una señal de ultrasonido para iniciar la medición.
* **Echo:** (Eco) Este pin mide el tiempo que tarda el pulso de ultrasonido en regresar al sensor, lo que permite medir la distancia.
* **GND:** (Tierra) Este es el conector a tierra del circuito



Pixy 2

Nuestro robot utiliza la cámara Pixy 2 para una tarea crucial: identificar el color de los semáforos (ya sea rojo o verde) y así poder navegar la pista correctamente, esquivando obstáculos según la señal



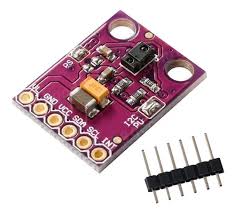
La Pixy 2 se conecta a nuestro sistema a través de varios pines clave:

* **Pines de Alimentación (VCC):** La Pixy 2 puede ser alimentada con 5V.
* **Pines de Tierra (GND):** Cuenta con múltiples pines GND para asegurar una conexión a tierra robusta.

**Pines de Comunicación:**

* **SPI (MISO, MOSI, CLK, CS):** Para comunicación de alta velocidad con el Arduino, enviando datos sobre los objetos detectados.
* **I2C (SDA, SCL):** Otra opción de comunicación, útil para conectar con una variedad de microcontroladores.
* **UART/Serial (TX, RX):** Una forma sencilla de enviar datos seriales a nuestro sistema.
* **Entrada analógica:** Para posibles configuraciones o lecturas externas.
* **Salidas digitales:** Para controlar directamente otros componentes basados en lo que detecte la Pixy 2.

Sensor de color (GY-9960)



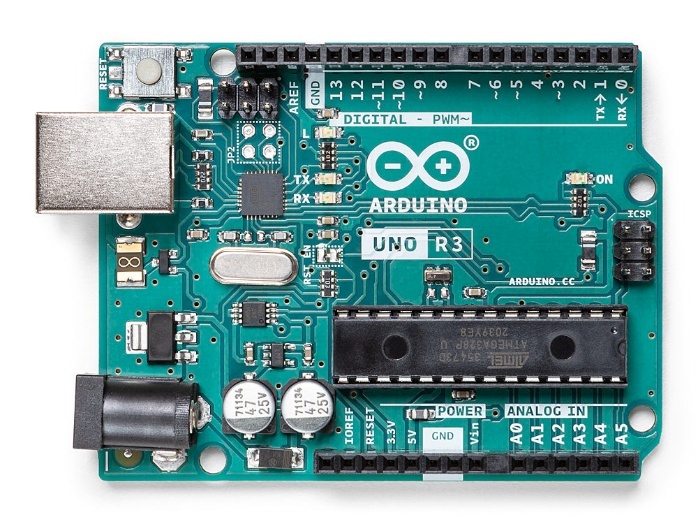
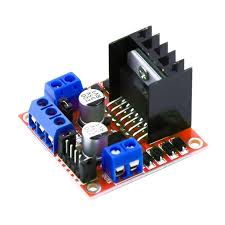
El GY-9960 es una solución avanzada que no solo identifica colores primarios, sino que también mide la intensidad de la luz ambiental, lo que le permite a nuestro robot adaptarse a las variaciones de iluminación del entorno. Su comunicación con el microcontrolador es eficiente, utilizando un protocolo I2C que facilita la transmisión de datos.

Para dotar a nuestro robot de la capacidad de interpretar el color de los semáforos, hemos integrado el sensor de color GY-9960. Este componente es fundamental, actuando como los "ojos" del robot que identifica el color, permitiéndole tomar decisiones informadas sobre cuándo avanzar o detenerse en la pista.

**Este sensor cuenta con pines esenciales para su funcionamiento:**

* **VCC:** Proporciona la alimentación eléctrica, que generalmente es de 3.3V o 5V.
* **GND:** La conexión a tierra para asegurar la estabilidad del circuito.
* **SDA y SCL:** Los canales dedicados para la transmisión de datos y la sincronización con el microcontrolador.
* **INT:** Un pin opcional que puede generar una interrupción, alertando al microcontrolador de un cambio significativo de color, lo que optimiza la capacidad de respuesta del robot.

2.6 Microcontroladores



Puente H L298N

Arduino Uno

Para nuestro robot, el **Puente H L298N** actúa como el "comunicador" que transforma las órdenes de nuestro microcontrolador en el movimiento del **único motor de corriente continua**. Es vital porque amplifica la señal del Arduino para mover el motor, permitiéndonos controlar con precisión tanto la dirección como la velocidad del robot.

Los microcontroladores en nuestro robot es el cerebro electrónico que, en un solo chip, integra una CPU, memoria y puertos de entrada/salida. Su función primordial es coordinar todas nuestras operaciones: recibe datos de los sensores para entender el entorno, los procesa según gracias a nuestra programación para tomar decisiones, y luego envía comandos a los actuadores (como el motor y el servo) para que el robot se mueva y realice sus tareas de forma autónoma

Para la gestión integral de nuestro robot, hemos seleccionado el Arduino Uno como la unidad central de procesamiento. Esta placa de desarrollo trabaja como el cerebro del sistema, encargado de recibir e interpretar los datos provenientes de los sensores, y de ejecutar las acciones correspondientes a través de los actuadores, asegurando una operación coordinada y precisa.

3. Estrategias para la ronda 1

Nuestra primera gran decisión fue enfocarnos en el reto número 1. Aquí, el mayor desafío era la definición de la dirección. Para superarlo, ideamos la siguiente estrategia: usar las líneas naranja y azul de la cancha como puntos de referencia clave. Estas líneas nos indican si el robot debe girar en sentido horario o antihorario.

Una vez que el robot ejecuta el primer giro, sabemos que los siguientes giros en esa sección de la pista seguirán el mismo sentido. Por ello, nuestra estrategia es: después de que el sensor de color confirme que hemos detectado una línea de un color diferente al blanco (es decir, naranja o azul), el robot automáticamente iniciará el giro. Esto nos permite una navegación fluida.

Para que el robot vaya recto usamos el giroscopio para que siempre este yendo con el ángulo correcto y el ultrasonido mediante un sistema PID (proporcional, integral y derivada) determinamos si tiene que girar para alinearse a la distancia correcta de la pared de afuera por si alargan la pared de adentro y que no tenga un efecto en la rutina

Para estacionarnos al iniciar nuestra secuencia de prueba, la primera tarea fue registrar meticulosamente las distancias iniciales a la pared exterior y a la pared interior utilizando los sensores ultrasónicos. Estos valores sirvieron como nuestros puntos de referencia. Tras completar las tres vueltas programadas, el robot se posicionará con la mayor precisión posible en el punto de partida original. En ese momento, volverá a leer las distancias de ambas paredes. Para validar el retorno, comparara cuidadosamente estas últimas mediciones con las iniciales, permitiendo un margen de error mínimo (de aproximadamente 1-2 centímetros).

Para acceder a nuestro diagrama del desafio 1: [..\..\Downloads\diagrama de flujo 2.pdf](file:///C:\Users\eduar\Downloads\diagrama%20de%20flujo%202.pdf)

3.3 Estrategias para el desafío 2

en este segundo reto, tomamos en cuenta que hay que pulsar cantidades de tiempo distintas para que el robot pueda moverse al sentido horario y antihorario, la estrategia principal es que el robot identifique los obstáculos con la cámara pixy, y en base a lo que vea, determine a cuanta distancia debe estar de la pared exterior, haciendo que cuando vea rojo o verde tomé la decisión, y en base a la distancia en la que se encuentre de los pilares, que el ángulo de giro se pronuncie más o menos.

tomando en cuenta que si es verde el vehículo debe ir a la izquierda de éste, en caso de que vaya en sentido antihorario, se sumará más distancia a la que debe estar el vehículo de la pared exterior, y si es rojo tendría que ir a la derecha de este, así que en el mismo caso tendría que tener menos distancia a tomar en cuenta en base a la pared exterior.

En el caso de que sea horario, sería el caso contrario, así que si detecta que viene un pilar verde, irá a la izquierda de este y se le restará menos a la distancia entre la pared y el robot, y con los rojos se le sumará.

cuando vaya a estacionarse, para saber en qué sección deberá quedarse, contara las veces que ha girado, y si giro 12 veces ,sabrá que debe estacionarse, así que este va a seguir avanzando hasta que la cámara identifique el estacionamiento, después avanzara un poco mas para quedar de retro frente a el estacionamiento y se echara para atrás girando para quedar en paralelo.

Nuestro diagrama de flujo del desafío dos

[Untitled (6)[1].pdf](file:///C:\Users\eduar\AppData\Local\Microsoft\Windows\INetCache\IE\RSY4HU3U\Untitled%20(6)%5b1%5d.pdf)

**Conectores XT30 para un Puente H Seguro**

Usar conectores XT30 es clave para tus conexiones al puente H. Son pequeños pero fuertes, perfectos para alimentar motores porque manejan bien la corriente sin calentarse. A diferencia de las soldaduras o tornillos que pueden fallar, los XT30 dan una conexión sólida y eficiente. Esto evita cortocircuitos, asegura que tus motores reciban energía constante para un rendimiento fiable, y su diseño a prueba de errores simplifica el montaje y el mantenimiento.

Interruptor de Corte Rápido para Emergencias

Un interruptor de corte rápido en la línea principal de energía es indispensable. Ya sea un botón de emergencia grande o una palanca, te permite cortar la energía al instante si algo sale mal: un cortocircuito, un movimiento inesperado del robot, o cualquier situación de riesgo. En un entorno como la WRO, donde la rapidez es vital, esta parada de emergencia puede evitar daños a los componentes (motores, puente H, sensores), prevenir accidentes o proteger el campo de juego. Además, hace que las pruebas sean más seguras y fáciles.

Haber usado una sp32

Este mismo tiene muchas mas opciones disponibles para poder hacer cosas variadas, puedes conectarlo a otro sp32, conectarlo al wifi y asi puedes programarlo por bluetooth, así seria mas cómodo, tiene mas puertos en menor espacio, la única desventaja es que no hay tantas librerías como en Arduino

4. Opciones de mejoras