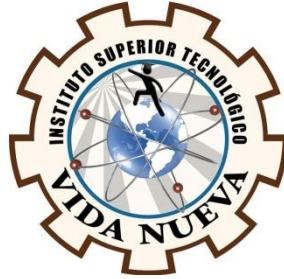


INSTITUTO SUPERIOR TECNOLÓGICO

VIDA NUEVA



CARRERA:

TECNOLOGIA EN ELECTROMECAÁNICA

TEMA DEL PROYECTO DE APLICACIÓN PRÁCTICA

**DISEÑO Y CONSTRUCCIÓN DE UN ROBOT SEGUIDOR DE LÍNEA
EVASOR DE OBSTÁCULOS EMPLEANDO ARDUINO NANO**

AUTOR:

ABRAJAN ARIAS CRISTHIAN SANTIAGO

TUTOR:

ING. MACHAY TISALEMA BYRON ORLANDO

FECHA:

ENERO 2020

QUITO – ECUADOR

CESIÓN DE DERECHOS DE AUTOR

Yo, CRISTHIAN SANTIAGO ABRAJAN ARIAS portador/a de la cédula de ciudadanía 0503401069, facultado/a de la carrera TECNOLOGÍA EN ELECTROMECAÁNICA, autor/a de esta obra certifico y proveo al Instituto Tecnológico Superior Vida Nueva, usar plenamente el contenido del informe con el tema “DISEÑO Y CONSTRUCCIÓN DE UN ROBOT SEGUIDOR DE LÍNEA EVASOR DE OBSTÁCULOS EMPLEANDO ARDUINO NANO”, con el objeto de aportar y promover la lectura e investigación, autorizando la publicación de mi proyecto de titulación en la colección digital del repositorio institucional bajo la licencia de Creative Commons: Atribución-NoComercial-SinDerivadas.

En la ciudad de Quito, del mes de Enero del 2020.

ABRAJAN ARIAS CRISTHIAN SANTIAGO

C.I.: 0503401069

CERTIFICACIÓN DEL TUTOR

En mi calidad de Tutor del Proyecto: “DISEÑO Y CONSTRUCCIÓN DE UN ROBOT SEGUIDOR DE LÍNEA EVASOR DE OBSTÁCULOS EMPLEANDO ARDUINO NANO” en la ciudad de Quito, presentado por el ciudadano CRISTHIAN SANTIAGO ABRAJAN ARIAS, para optar por el título de Tecnólogo en Electromecánica, certifico, que dicho proyecto ha sido revisado en todas sus partes y considero que reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del tribunal examinador que se designe

En la ciudad de Quito, del mes de Enero del 2020.

TUTOR:

ING. MACHAY TISALEMA BYRON ORLANDO

APROBACIÓN DEL TRIBUNAL

Los miembros del jurado examinador aprueban el informe de investigación, sobre el tema: “DISEÑO Y CONSTRUCCIÓN DE UN ROBOT SEGUIDOR DE LÍNEA EVASOR DE OBSTÁCULOS EMPLEANDO ARDUINO NANO” en la ciudad de Quito Del estudiante: CRISTHIAN SANTIAGO ABRAJAN ARIAS de la Carrera en Tecnología en Electromecánica.

Para constancia firman:

.....

.....

DECLARACIÓN DE AUTENTICIDAD

Yo, CRISTHIAN SANTIAGO ABRAJAN ARIAS con cedula de ciudadanía 0503401069 estudiante del Instituto Superior Tecnológico Vida Nueva, declaro que he realizado este trabajo de titulación tomando en consideración citas bibliográficas que se nombran en este texto.

El Instituto Superior Tecnológico Vida Nueva puede utilizar este trabajo de titulación como una ayuda bibliográfica.

En la ciudad de Quito, del mes de Enero del 2020.

ABRAJAN ARIAS CRISTHIAN SANTIAGO

C.I.: 0503401069

AGRADECIMIENTO

Agradezco a Dios por la oportunidad que me ha dado de alcanzar esta meta, a mi madre María Arias, a mi abuelita Carmen Ugsha a mis tíos: Lucrecia-Nestor, que han sido un pilar muy importante durante este proceso, a mi tutor de Tesis ING. Byron Machay que con su conocimiento y gran trayectoria ha logrado guiarme para culminar mi Tesis, a cada uno de los docentes de esta noble Institución como es el Instituto Superior Tecnológico Vida Nueva.

ÍNDICE

CERTIFICACIÓN	i
APROBACIÓN DEL JURADO EXAMINADOR O TRIBUNAL	iii
DECLARACIÓN DE AUTENTICIDAD	iv
1. INTRODUCCIÓN	1
2. ANTECEDENTES	2
3. OBJETIVOS	3
3.1 Objetivo General	3
3.2 Objetivos específicos	3
4. DESARROLLO	4
4.1 Marco teórico – conceptual	4
4.1.1 Robots seguidores de línea	4
4.1.2 Tipos de Robots seguidores de línea.....	4
4.1.3 Locomoción	7
4.1.5 Motores	10
4.1.7 Sensores.....	13
4.1.9 Puente H.....	15
4.1.11 Arduino	18
4.1.13 PISTA	21
4.2 Procedimiento- Metodología	23
4.2.1 Diseño.....	23
4.2.2 Construcción.....	33
4.2.3 Implementación.....	35
5. CONCLUSIONES.....	39
6. RECOMENDACIONES	40
7. REFERENCIAS	41
8. ANEXOS	x

ÍNDICE DE FIGURAS

<i>Figura 1.</i> Seguidor de línea.....	4
<i>Figura 2.</i> Seguidor de línea Turing.....	5
<i>Figura 3.</i> Seguidor de línea velocista.....	6
<i>Figura 4.</i> Robot Weasel.	6
<i>Figura 5.</i> Modelo de un robot de tracción tipo triciclo.....	7
<i>Figura 6.</i> Locomoción Ackerman.	8
<i>Figura 7.</i> Tracción diferencial.....	9
<i>Figura 8.</i> Motor DC.	11
<i>Figura 9.</i> Servomotor.	12
<i>Figura 10.</i> Mircromotor Pololu 50:1.....	12
<i>Figura 11.</i> Sensor QTR8A.....	14
<i>Figura 12.</i> Sensor analógico Sharp.....	15
<i>Figura 13.</i> Puente H.....	15
<i>Figura 14.</i> Puente H L392B.....	16
<i>Figura 15.</i> Drive TB6612FNG.	17
<i>Figura 16:</i> Arduino uno.....	19
<i>Figura 17.</i> Arduino Leonardo.	19
<i>Figura 18.</i> Arduino Mega2560.....	20
<i>Figura 19.</i> Arduino Nano.....	21
<i>Figura 20.</i> Pista para seguidor de línea.	22
<i>Figura 21.</i> Pantalla de programa isis.	23
<i>Figura 22.</i> Selección de componentes.....	24
<i>Figura 23.</i> Resistencia.	24
<i>Figura 24.</i> Terminal de bloque de dos salidas.	25
<i>Figura 25.</i> Selección de terminales de bloque con tres salidas.	25
<i>Figura 26.</i> Terminal de bloque de diez salidas.....	26
<i>Figura 27.</i> Librería de Arduino.	26
<i>Figura 28.</i> Driver TB6612FNG.	27
<i>Figura 29.</i> Elementos en devices.....	27
<i>Figura 30.</i> Conexión inalámbrica del Arduino con el drive TB6612FNG.	28
<i>Figura 31.</i> Sensores Sharp y Arduino.....	28
<i>Figura 32.</i> Diseño del chasis 2D en Autocad 2016.	29

<i>Figura 33.</i> Diseño del chasis 3D en Autocad 2016.	29
<i>Figura 34.</i> Programa Ares proteus.....	30
<i>Figura 35.</i> Chasis en Ares proteus.....	31
<i>Figura 36.</i> Elementos electrónicos en la placa de diseño.	31
<i>Figura 37.</i> Diseño de robot en Isis Proteus.....	32
<i>Figura 38.</i> Diseño impreso en Baquelita.	33
<i>Figura 39.</i> Baquelita en cloruro férrico.	34
<i>Figura 40.</i> Remoción del cobre de la baquelita.	34
<i>Figura 41.</i> Sujeción con pernos del motor pololu.....	35
<i>Figura 42.</i> Robot seguidor de línea.....	35
<i>Figura 43.</i> Librería QTRsensors.....	36
<i>Figura 44.</i> Configuración de pines E/S.	36
<i>Figura 45.</i> Configuración E/S.....	37
<i>Figura 46.</i> Puerto void loop().....	37
<i>Figura 47.</i> Opciones de librería QTRsensors.....	38
Anexo 1. Programa del robot	x
Anexo 2. Plano de AutoCAD.....	xi
Anexo 3. Plano de diseño en Isis.....	xii
Anexo 4. Plano de diseño en Ares.....	xiii

ÍNDICE DE TABLAS

Tabla 1.....	13
Tabla 2.....	14
Tabla 3.....	17
Tabla 4.....	21

1. INTRODUCCIÓN

Este proyecto de aplicación práctico abarca todos los aspectos que implican la construcción del robot como son; el diseño de la PCB, el montaje de elementos electrónicos y programación de la tarjeta Arduino nano, dos sensores Sharp, un QTR8A, un driver Tb6612FNG, que permiten enviar órdenes e instrucciones pertinentes a los motores para su desplazamiento y que sea capaz de seguir una línea y evadir obstáculos. El principio de funcionamiento es que el robot se desplace hacia adelante o hacia atrás conservando su orientación, las ruedas deben girar a la misma velocidad y en la misma dirección. El tipo de locomoción es mediante ruedas de tracción diferencial con un par de motores para que el seguidor cambie de dirección.

La importancia de realizar el robot seguidor de línea evasor de obstáculos es desarrollar el interés de la robótica en las nuevas generaciones ya que este proyecto de aplicación práctico pretende abarcar todos los aspectos que implican el diseño de la parte electrónica mediante el programa Isis Proteus 7.9, la construcción mecánica utilizando el software Ares Proteus y de control empleando software y hardware Arduino.

Es importante la construcción de un seguidor de línea evasor de obstáculos ya que será utilizado en competencias de robots con otras Instituciones Superiores a nivel regional y nacional, motivará a futuros estudiantes a desarrollar interés por la robótica; el robot permanecerá en el laboratorio de la institución para uso didáctico y de esta manera empezar a cultivar el interés por el desarrollo de nuevos avances en el área de la robótica.

2. ANTECEDENTES

Durante siglos el ser humano ha construido máquinas tecnológicas que imiten las partes del cuerpo humano o que ayuden a automatizar ciertas tareas que son complejas y repetitivas. “Desde hace tiempo se han ido incorporando dispositivos para facilitar determinadas tareas dentro de los entornos industriales. La industria pretende incorporar muchos más dispositivos para dotar de la máxima inteligencia a los procesos que se desarrollan dentro de estos entornos, entre otros, robots y drones”. (INCIBE-CERT, 2017,p1)

En el Ecuador se han desarrollado competencias de robots tanto a nivel práctico como teórico, existen 22 categorías de competición entre ellas soccer, micro zumo, mini zumo, mega zumo, concurso de velocidad, robots trepadores, voladores, seguidores de línea y la batalla de robots que es la más popular porque los prototipos de hasta 120 libras se enfrentan en un ring y utilizan armas y dispositivos de golpes para destruir a sus contrincantes.

El equipo de la Escuela Politécnica de Chimborazo (Spoch) ganó en la batalla de 30 libras con su robot Mini Taz que sale victorioso por tercer año consecutivo. Fabián Padilla, miembro del equipo explicó que su robot ha sufrido muchas modificaciones durante estos años para hacerlo más competitivo.

El secreto que esconde Mini Taz está en su arma rotatoria que gira a 600 revoluciones por minuto y tiene un peso de 10 libras lo que genera más potencia para el ataque y una mayor destrucción para sus contrincantes. (La nacion, 2016,p.1)

Los robots seguidores de línea evasores de obstáculos son diseñados para seguir una línea negra en un fondo blanco. Una de las aplicaciones de este robot es realizar el transporte de maquinaria o paquetería, ya que puede llevar un paquete desde un punto a otro evitando chocar con posibles obstáculos dentro de su vía, estos robots se diferencian unos a otros debido a su arquitectura y principalmente el peso que deben transportar.

3. OBJETIVOS

3.1 *Objetivo General*

Diseñar y construir un robot móvil empleando la tarjeta Arduino nano para que sea capaz de seguir una línea y evadir obstáculos.

3.2 *Objetivos específicos*

- Seleccionar y ejecutar un sistema de tracción que permita el movimiento rotacional y lineal del robot seguidor.
- Diseñar el circuito electrónico en Proteus, para la simulación de sensores Sharp, QTR8A y actuadores eléctricos.
- Construir el robot seguidor de línea evasor de obstáculos mediante la implementación de actuadores, Arduino nano y elementos eléctricos en la placa PCB.

4. DESARROLLO

4.1 Marco teórico – conceptual

4.1.1 Robots seguidores de línea

Los Robots seguidores de línea tienen la capacidad de seguir un camino mismo que es trazado por una línea para con el objeto de trasladarse de un lugar a otro, la línea trazada en el suelo tiene que tener contraste con el color del resto del suelo por ejemplo si la línea es negra el suelo tiene que ser blanco y viceversa como se muestra en la figura 1.

Los Seguidores de línea son robots pequeños que intentan desplazarse por una trayectoria dictado por una línea, estos proyectos tomaron de inspiración de robots utilizados en la construcción de automóviles, dichos robots son encargados de llevar objetos desde un punto A a un punto B, generalmente siguen una línea en el suelo para que su programación pueda ser ejecutada. (Ortiz, 2016, p. 9)

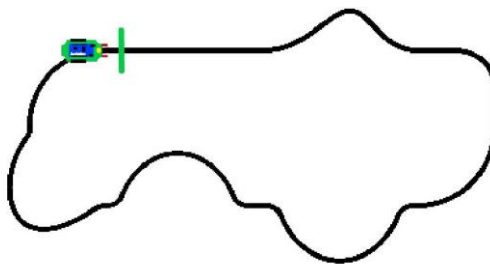


Figura 1. Seguidor de línea
Fuente: Roboticeso (2019)

4.1.2 Tipos de Robots seguidores de línea

Existen diferentes tipos y modelos de robots cada uno diseñado con características que lo diferencia del resto ya que son creados para distintos fines u objetivos, ya sea para competencia de Robots como para el ámbito estudiantil y aplicaciones industriales, entre estos tipos tenemos: Seguidor de línea velocista, seguidor de línea Turing y seguidor de línea robot Weasel.

- **Seguidores de línea Turing**

Este robot Turing es un kit que contiene todos los elementos para armar un robot seguidor de línea y es ideal para estudiantes que deseen adquirir conocimiento y destrezas en la rama de la robótica que están conformados por los siguientes elementos electrónicos.

El Robot Turing está conformado por los siguientes elementos electrónicos: Motores N20 a 2000 RPM con caja reductora de 10:1, barra de sensores QTR-8A, ruedas de estilo pololu, chasis PCB diseñado de forma exclusiva, batería de 7,4v, puente H Dual TB6612FNG, Arduino nano y más. (ElectronicaStore, 2018,p1)



Figura 2. Seguidor de línea Turing
Fuente: ElectronicaStore (2018)

- **Seguidor de línea velocista**

Este tipo de robot velocista utiliza motores de alta velocidad, y ruedas que tengan la característica de mayor adherencia al piso permitiéndole mayor velocidad según, tdrobotica, (2014) “Son indispensables este tipo de ruedas ya que, aunque el robot es un seguidor de línea, al alcanzar velocidades de más de 2 metros por segundo necesita un agarre óptimo en cada curva”. El chasis tiene que ser diseñado en PCB para lograr ocupar menos espacio tiene que ser ligero para que pueda soportar su propio peso y así alcanzar la velocidad deseada, como se observa en la figura 3.

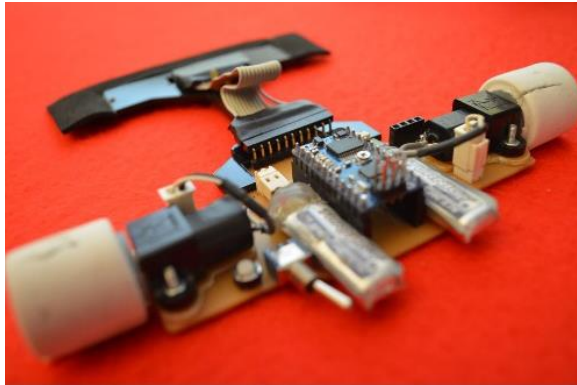


Figura 3. Seguidor de línea velocista
Fuente: tdrobotica (2014)

- **Seguidor de línea Robot Weasel**

Este tipo de robot está diseñado y creado para trasladar objetos pesados de un lugar a otro se utilizan ya en aplicaciones industriales por lo general en centros de almacenaje de mercadería, así como se muestra en la figura 4. Schaefer, (2016) Afirma que:

Los AGV se han convertido ya en un componente en sistemas intralingüísticos plenamente automatizados. La gama de vehículos de SSI SCHAEFER incluye multitud de soluciones innovadoras para unidades de carga pesada y de grandes dimensiones así también de contenedores de carga ligera que permiten crear instalaciones logísticas conectando distintas áreas de trabajo y del almacén.



Figura 4. Robot Weasel
Fuente: Schaefer (2016)

4.1.3 Locomoción

La locomoción ayuda al robot a desplazarse de un lugar a otro, se elige tomando en cuenta aspectos como: forma, estructura y velocidad.

La palabra locomoción en términos específicos hace referencia al movimiento, puede ser este de una persona, un animal, un microorganismo o maquina con el fin de desplazarse de un lugar a otro. La locomoción se clasifica según: la forma, velocidad, estructura y otros elementos de acuerdo al tipo de sujeto al que hace referencia. (Bembibre, 2010, p.5)

4.1.4 Tipos de Locomoción

El tipo de locomoción más utilizado para robots seguidores de línea es mediante ruedas, existen diferentes tipos de tracción, entre las más usadas tenemos: diferencial, triciclo y ackerman.

- **Locomoción tipo triciclo**

En este tipo de locomoción se utilizan tres ruedas una delantera y dos ruedas posteriores, según Moctezuma, (2016) “La rueda delantera sirve para tracción y direccionamiento, el eje trasero con dos ruedas laterales, es pasivo y sus ruedas se mueven libremente”. (p.3). Este tipo de tracción tiene dos funciones, la primera función: la rueda delantera sirve como tracción-direccionamiento y las ruedas posteriores de soporte que se mueven libremente y la segunda función: las dos ruedas posteriores sirven como tracción y la primera rueda de guía, como se puede observar en la figura 5.

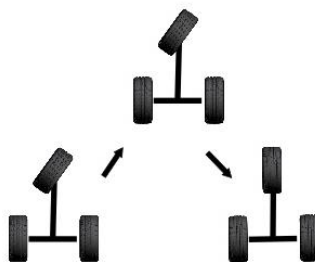


Figura 5. Modelo de un robot de tracción tipo triciclo
Fuente: Moctezuma (2016)

Ventajas

- No hay deslizamiento
- Desplazamiento fácil en línea recta

Inconvenientes

- No puede girar sobre su centro de gravedad

- **Locomoción Ackerman**

El sistema de locomoción Ackerman posee cuatro ruedas, sistema de tracción de los vehículos convencionales las ruedas traseras son accionadas por un solo motor mientras que las delanteras son de direccionamiento. “Este sistema se basa en dos ruedas traseras trayectorias paralelas en el chasis principal de vehículo, mientras que las ruedas delanteras son de tipo direccionamiento y sirven para seguir la trayectoria que debe seguir el robot”. (Sirpa, 2019, p1). Como se muestra en la figura 6.

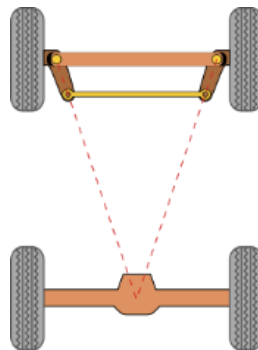


Figura 6. Locomoción Ackerman

Fuente: Proyecto (2018)

Ventajas

- El robot puede moverse instantáneamente adelante o atrás.
- Un sistema simple de cuatro barras controla la dirección.

Desventajas

- No puede moverse lateralmente.

- **Locomoción diferencial**

Este tipo de locomoción se compone de dos ruedas paralelas entre sí, cada rueda se controla en forma independiente por un motor para trasladarse en línea recta cada rueda debe girar a la misma velocidad.

Se componen de 2 ruedas en un eje común, cada rueda se controla independientemente, puede realizar movimientos en línea recta, en arco y sobre su propio eje de contacto de rodamiento, requiere de una o dos ruedas adicionales para balance o estabilidad. (uvpr-robótica, 2013, p1)

En este sistema no hay ruedas directrices, el giro de dirección se realiza modificando la velocidad relativa de las ruedas izquierdas y derecha. En la figura 7 se puede observar el tipo de locomoción diferencial.

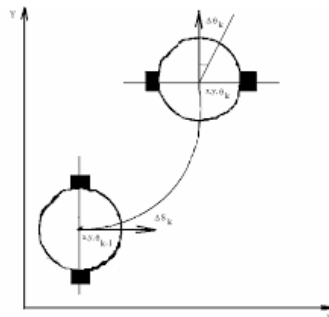


Figura 7. Tracción diferencial
Fuente: Valencia (2009)

Ventajas

- Sistema barato.
- Fácil de implementar.
- Diseño simple.

Inconvenientes

- Difícil de controlar
- Requiere control de precisión para trayectorias rectas.

4.1.5 Motores

Los motores eléctricos son parte fundamental de los robots ya que se encargan de producir movimiento y transforman la energía eléctrica en mecánica y los mismos ayudan al desplazamiento del robot.

Todos los robots incluyen algún sistema capaz de producir movimiento siendo los más comunes los motores de corriente continua (DC) y los servos motores o servos. Los primeros se utilizan casi siempre junto con un sistema de engranajes que reducen la velocidad y proporcionan mayor fuerza. (Superrobotica, 2019, p. 4)

4.1.6 Tipos de motores

Existen varios tipos de motores que se utilizan especialmente en los robots estos son: Motores de corriente continua DC, servomotores, micromotores, para este proyecto se ha utilizado los micromotores, poseen una caja de engranajes que disminuyen su velocidad y aumentan el torque, la utilización de estos motores beneficia al robot ya que por su diseño y características no ocupan mucho espacio en el robot y son livianos.

- **Motores DC**

Los motores eléctricos DC se pueden encontrar en muchos dispositivos, rotan de forma continua cuando se les aplica un voltaje de corriente directa, como se observa en la figura 8 “El motor de corriente continua (motor DC) es una máquina que convierte la energía eléctrica en mecánica, provocando un movimiento rotatorio. En algunas modificaciones, ejercen tracción sobre un riel. Estos motores se conocen como motores lineales”. (Geekbotelectronics, 2019,p1)



Figura 8. Motor DC
Fuente: Geekbotelectronics (2019)

- **Servomotores**

Los servomotores son motores que dan una mayor precisión de rotación permitiendo así controlar la posición del eje del motor.

Los servos se controlan utilizando anchos de pulsos ajustables en la línea de señal. Para los servos estándar enviar un pulso de 5v de 2ms lo gira a 180grados, mientras que las longitudes de pulso que hay en medio escalan de forma lineal. Por ejemplo, un pulso de 1.5 ms gira el motor a 90 grados. Una vez que se ha enviado un pulso, el servo gira a esa posición y se queda ahí hasta que recibe otra instrucción de pulsos. Sin embargo, si se quiere un servo se mantenga en esa posición se resista a que se le empuje y trate de mantener la posición exacta, se tiene que volver a enviar el comando una vez cada 2ms. (Blum, 2014, p. 96)

A diferencia de los motores CC, los servomotores tienen tres pines: Alimentación (por lo general rojo), tierra (normalmente marrón o negro) y señal (por lo general blanco o naranja, estos cables están configurados por colores generalmente en un mismo orden como se muestra en la figura 9.



Figura 9. Servomotor.
Fuente: Pololu (2019).

- **Micromotores**

Los Micromotores poseen una caja de engranajes de metal estos pequeños motores de CC con escobillas están disponibles en una amplia gama.

Con una sección transversal que mide solo 10 × 12 mm, estos pequeños motores de engranajes de CC cepillados están disponibles en una amplia gama de relaciones de transmisión, desde 5: 1 hasta 1000: 1, y con cuatro motores diferentes: alta potencia con carbono de larga duración cepillos (HPCB), de alta potencia (HP), de potencia media (MP) y de baja potencia. Las versiones también están disponibles con un eje de motor extendido opcional. (Pololu, 2019, p. 1)

Para este proyecto se ha utilizado el micromotor pololu 50:1 es un motor de corriente continua. “Tiene caja de engranajes de metal 51.45:1, de 6v de alta potencia cuenta con una sección transversal de 10x12mm y el eje de salida es de 9mmx3mm de diámetro. (Pololu, 2019, p. 1)



Figura 10. Micromotor Pololu 50:1
Fuente: Pololu (2019)

Tabla 1

Características de motor pololu

Relación de reducción	50:1
Relación de reducción	50:1
Velocidad libre del motor a 6 V	625 rpm
Consumo con carga a 6V	1600 mA
Torque máximo a 6V	1.1 kg-cm (15 oz · In)
Tamaño	24 x 10 x 12 mm

Fuente: Pololu (2019)

4.1.7 Sensores

Los sensores son elementos electrónicos capaces de variar magnitudes físicas que se encuentran en el medio, manifiesta la presencia de dicha magnitud y también su medida, para este proyecto se ha seleccionado dos tipos de sensores: sensor QTR 8A para la función de seguidor de línea y Sharp para la evasión de obstáculos.

4.1.8 Tipos de sensores

- **Sensor QTR 8A**

El módulo QTR8A es un dispositivo electrónico que tiene 8 sensores IR led fototransistor montados a 9.5 mm uno del otro, por lo que es un gran detector para un seguidor de línea.

El arreglo sensor de reflexión-QTR 8A está pensado como un sensor de línea, pero puede ser utilizado para identificar la proximidad o como un sensor de reflectancia. El módulo dispone de ocho emisores de infrarrojos y de 8 receptores (fototransistor). El fototransistor se conecta a una resistencia de pull-up para formar un divisor de voltaje que produce una salida de voltaje analógica entre 0V y VIN que suele ser de 5V en función del IR reflejado. Bajo voltaje de salida es una indicación de una mayor reflexión. (tdrobotica, 2019, p1)

Las salidas son todas independientes el encendido/apagado del led resulta ventajosa para el consumo de corriente, como se observa en la figura 11.

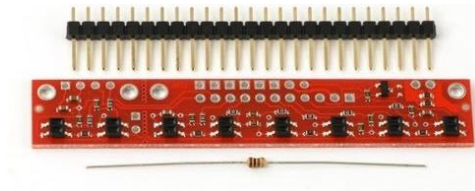


Figura 11. Sensor QTR8A
Fuente: tdrobotica (2019)

Tabla 2
Características del sensor QTR8A

Voltaje de funcionamiento	3.3V-5,0 V
Formato de salida	Voltaje análogo
Distancia óptima de detección	3mm
Máxima distancia de detección recomendada	6mm

Fuente: tdrobotica (2019)

- **Sensor infrarrojo Sharp**

Los sensores infrarrojos Sharp son dispositivos electrónicos diseñados para detectar objetos o señales que se encuentran cerca del sensor. “Los sensores de distancia infrarrojos de Sharp son famosos por medir la distancia entre un proyecto y otros objetos. Al alejarse del objeto que está apuntando hace que disminuya la salida de voltaje”. (Blum, 2014, p. 67)

Este sensor de distancia se puede usar para determinar la distancia desde el sensor hasta el objeto, posee un cable conector JST de tres pines por separado, el cable rojo se conecta a 5v, el cable negro es la tierra y el cable blanco devuelve un voltaje analógico entre 0 y 5v proporcional a la distancia medida, como se muestra en la figura 12.



Figura 12. Sensor analógico Sharp
Fuente: Tecnopura (2019)

4.1.9 Puente H

El puente H es un dispositivo electrónico que permite controlar la velocidad y el sentido de giro de motores electrónicos. “El puente H tiene cuatro estados de funcionamiento principales: abierto frenado, avance y retroceso” (Blum, 2014, p. 87). Siguiendo el esquema, se puede ver que S1 y S3 no se cierran al mismo tiempo ya que esto causaría un corto circuito al igual que S2 y S4 como se muestra en la figura 13.

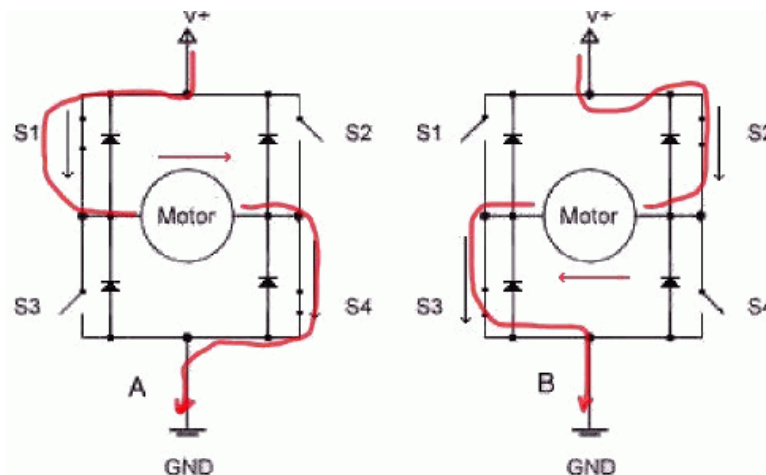


Figura13. Puente H
Fuente: Garcia (2014)

4.1.10 Tipos de puente H

Existen diferentes tipos de puentes H para el control de motores eléctricos, los más usados en el campo de la robótica por sus características técnicas y mejoras en tecnología tenemos el puente H L293B y TB6612FNG.

- **Puente H L293B**

El circuito integrado L293B permite controlar dos motores simultáneamente.

Es importante tener en cuenta que este integrado se alimenta con dos niveles de tensión diferente, uno corresponde a la alimentación propia de integrado, que no debe ser superior a 7V (VSS) y otra es la tensión con la que alimentaremos los motores, pudiendo en este último hacerlo con hasta 36V (VC). (Frank, 2019, p.5)

En la figura14 se puede observar la conexión del motor1 a 3 y 6, motor 2 a los pines 11 y 14, el pin 16 es la alimentación del integrado 5v, mientras que el pin 8 deberá estar conectado a la tensión que alimenta a los motores estando en un rango de (5-36) V.

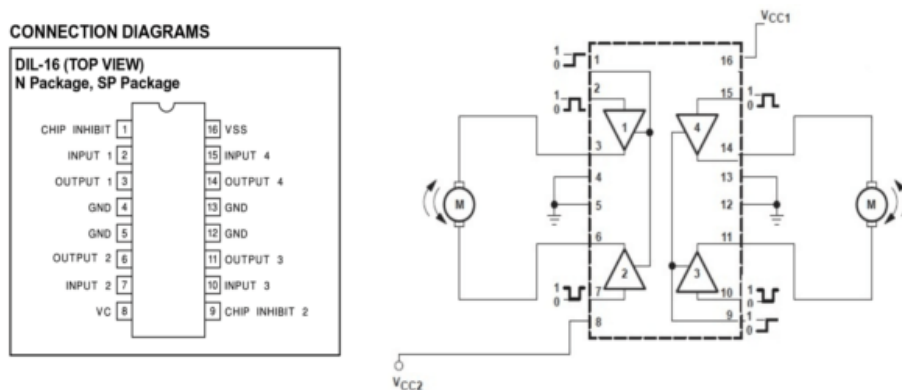


Figura 14. Puente H L392B
Fuente: Frank (2019)

- **Puente H TB6612FNG**

El driver TB6612FNG está diseñado para controlar el sentido de giro y velocidad de motores mediante señales TTL que se pueden obtener de micro controladores y tarjetas de desarrollo como Arduino, Raspberry Pi y Launchpads de Texas instruments.

El driver para motores TB6612FNG posee dos puentes H, puede controlar hasta dos motores de DC con una corriente constante de 1.2A (3.2A pico). Dos señales de entrada (IN1 y IN2) pueden ser usadas para controlar el motor en uno de cuatro modos posibles: CW (giro en sentido de las manecillas del reloj), CCW (en contra de las manecillas), short-brake y stop. Las dos salidas de motores (A y B) pueden ser controladas de manera separada, la velocidad de cada motor es controlada mediante una señal PWM con una frecuencia de hasta 100kHz. El pin STBY cuando es puesto en HIGH coloca al motor en modo de standby. El driver posee diodos internos de protección. (Nylampmechatronics, 2018, p.1)

Tabla 3
Características del driver TB6612FNG

Chip	TB6612FNG (Toshiba)
Canales	2 (soporta 2 motores DC o 1 motor PAP)
Voltaje de Potencia (VMOT)	5V - 15V
Voltaje Operación (VCC)	2.7V - 5.5V
Capacidad de corriente	1.2A (picos de hasta 3A)
Potencia máxima disipada	1W
Posee diodos internos de protección	

Fuente: Nylampmechatronics (2018)



Figura 15. Drive TB6612FNG
Fuente: Nylampmechatronics (2018)

4.1.11 Arduino

El Arduino es un circuito integrado de software y hardware libre, fácil de utilizar y es capaz de realizar operaciones matemáticas a gran velocidad, utiliza un microcontrolador Atmel.

En el corazón de todo Arduino hay una unidad de microcontrolador Atmel (MCU). La mayoría de las placas de Arduino utiliza un microcontrolador AVR ATmega. El lenguaje de programación de Arduino que proporciona acceso a periféricos del microcontrolador, incluidos convertidores de analógico a digital (ADC), buses de comunicación (I2C y SPI), pines de entrada y salida para fines generales, e interfaces de serie. (Blum, 2014, p. 24)

La placa de Arduino puede alimentarse directamente mediante el propio cable USB o a través de una fuente de alimentación externa como puede ser una batería de 9v o un pequeño transformador.

Los límites de alimentación de la placa de Arduino están entre los 6 y 12 v, hay que saber si la placa se alimenta con menos de 7v, la salida de tensión de regulación a 5v puede dar menos que este voltaje y si sobrepasa de 12v, probablemente se dañe la placa. (Tojeiro, 2014, p.4)

4.1.12 Tipos de Arduino

Hay muchas placas de Arduino y los fabricantes están lanzando placas nuevas con diferentes características en este capítulo se verán los diferentes tipos de Arduino y cada una de sus características.

- **Arduino Uno**

El Arduino uno es una placa que utiliza un microcontrolador de código abierto. "Arduino Uno Utiliza un microprocesador ATmega 328p con MCU principal utiliza un chip convertidor de USB a serie" (Blum, 2014, p. 27). La placa contiene pines de entradas y salidas digitales y

analógicas que se pueden conectar a otros circuitos y a varias placas de expansión, como se muestra en la figura 16.



Figura 16: Arduino uno
Fuente: ARDUINO (2013)

- **Arduino Leonardo**

La placa Arduino Leonardo es una placa de microcontrolador que a diferencia del Arduino Uno esta posee otras características.

Utiliza como microcontrolador principal el 32U4, que tiene una interfaz USB integrada. Por tanto, no necesita una MCU secundaria para llevar a cabo una conversación de serie a USB. Esto reduce el coste y nos permite hacer cosas únicas como emular un joystick o un teclado. (Blum, 2014, p. 27)

Posee al igual que el Uno un botón de reinicio, como se muestra en la figura 17.

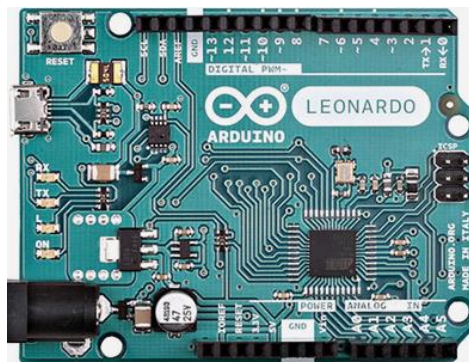


Figura 17. Arduino Leonardo
Fuente: ARDUINO (2013)

- **Arduino Mega 2560**

El Arduino Mega 2560 es una tarjeta de desarrollo de hardware libre construida con el microcontrolador ATmega 2560 del cual lleva su nombre.

Emplea como MCU principal un ATmega 2560 que posee 54 entradas y salidas generales las cuales permiten comunicar con más dispositivos, también tiene más canales ADC y cuatro interfaces de serie de hardware. (Blum, 2014, p.28)

A diferencia de la única interfaz en serie que posee Arduino Uno, esta posee cuatro, como se muestra en la figura 18.

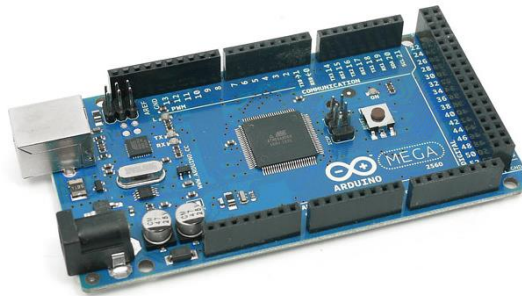


Figura 18. Arduino Mega2560
Fuente: Electrontools (2019)

- **Arduino Nano**

El Arduino nano al igual que las diferentes placas de Arduino es una tarjeta de desarrollo, ideal para proyectos de robótica en los cuales se desee optimizar recursos como espacio y peso.

La tarjeta Arduino nano está diseñada para montarse en el conector de una placa de pruebas, su tamaño reducido hace que sea perfecta para el uso de proyectos más acabados utiliza un microcontrolador ATmega 328, cuenta con 14 pines de E/S 6 de ellos son PWM y posee 8 entradas analógicas. (Blum, 2014, p. 28)

Tabla 4

Características técnicas del Arduino nano

Microcontrolador	ATMega 328
Voltaje de operación	5V
Voltaje de alimentación recomendado	7-12V
I/O digitales	14 (6 son PWM)
Entradas analógicas	8
Memoria flash	32KB
Frecuencia de trabajo	16MHZ
Dimensiones	0,73'' x 1,70''.

Fuente: Elarduino (2019)



Figura 19. Arduino Nano
Fuente: Elarduino (2019)

4.1.13 PISTA

Para el diseño de la pista de la modalidad seguidor de línea y detector de obstáculos se obtuvo en base al reglamento y requerimientos técnicos del CER2019 para el seguidor de línea y SOMIM 2019 para el tamaño de obstáculo a utilizar.

Las pistas tendrán fondo blanco y la línea del camino será de color negro con un ancho de 2cm +- 5%, la pista estará hecha con líneas rectas y curvas, no existirá discontinuidades ni intersecciones, el radio mínimo de las curvas serán de 7.5cm, la aproximación más cercana de la línea de curso a los bordes de la pista será no menos a 15cm, medidos desde el centro de la línea. (CER, 2019, p.1)

La competencia se lo realizara en una pista de 2.40m x 1.2m la superficie de la pista debe contar con un acabado que permita la tracción necesaria de las ruedas del robot, para la detección de obstáculos se utilizaran cilindros de 10cm de diámetro por 15cm de alto. (SOMIM, 2019, p1)

En la figura 20 se observa el diseño de una pista para el robot seguidor de línea.

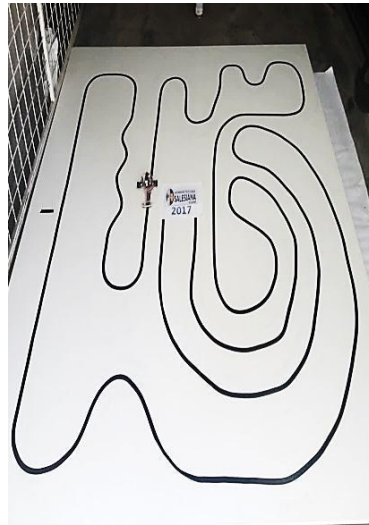


Figura 20. Pista para seguidor de línea
Fuente: POLITECNICA SALESIANA (2017)

4.2 Procedimiento-Methodología

4.2.1 Diseño

- **Diseño en Isis**

1. El software utilizado para el diseño de circuitos impresos PCB es Isis Proteus 7 profesional para la planificación y simulación del proyecto el cual brinda todas las herramientas necesarias, comandos y demás librerías que incluyen elementos electrónicos para la simulación del proyecto para empezar, abrir el programa Isis Proteus, como se observa en la figura 21.

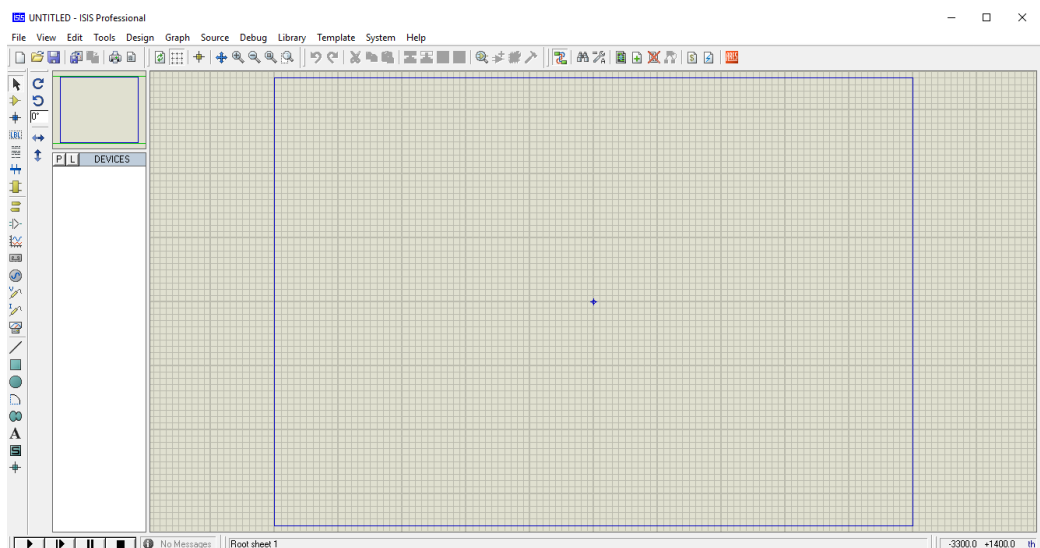


Figura 21. Pantalla de programa Isis
Fuente: Elaboración propia (2019)

2. Seleccionar los componentes en la opción selection mode, presionar la tecla pick from libraries, en la opción keywords, buscar y seleccionar los elementos a utilizar para la simulación del proyecto, como muestra en la figura 22.

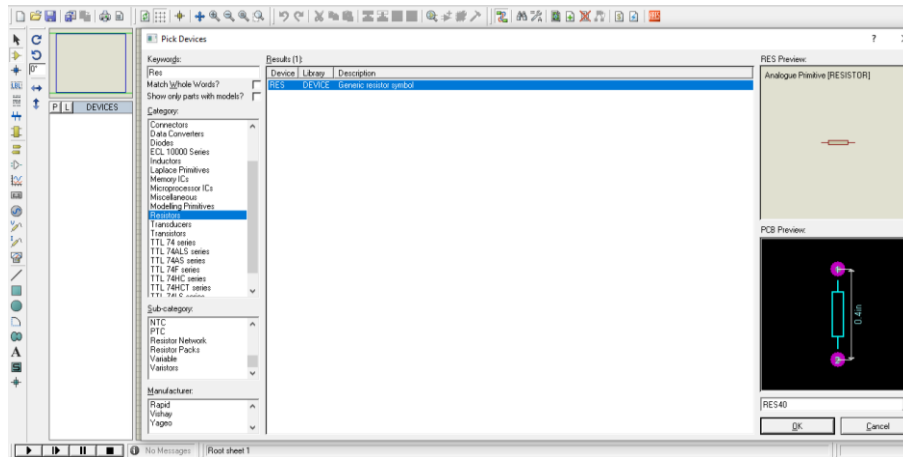


Figura 22. Selección de componentes
Fuente: Elaboración propia (2019)

- Utilizar elementos electrónicos que se muestran en la pantalla PCB preview los elementos que no aparezcan en esta pantalla serán inválidos para el diseño ya que carecen de un paquete PCB, como muestra en la figura23.

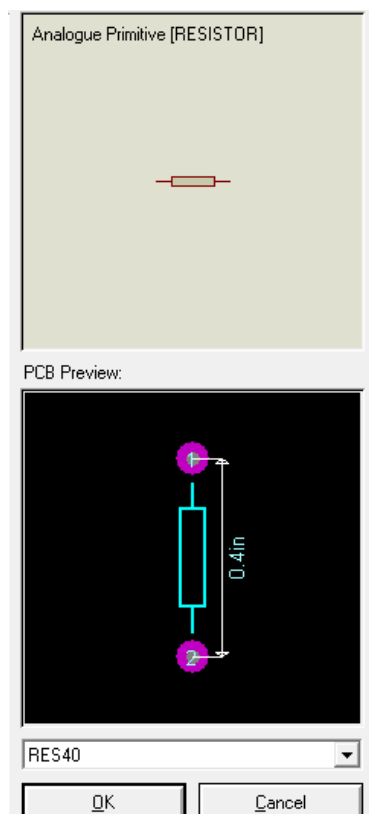


Figura 23. Resistencia
Fuente: Elaboración propia (2019)

4. Seleccionar los componentes terminales de bloque con dos salidas para conectar los motores en la opción selection mode, presionar la tecla pick from libraries, en la opción keywords, como muestra en la figura 24.

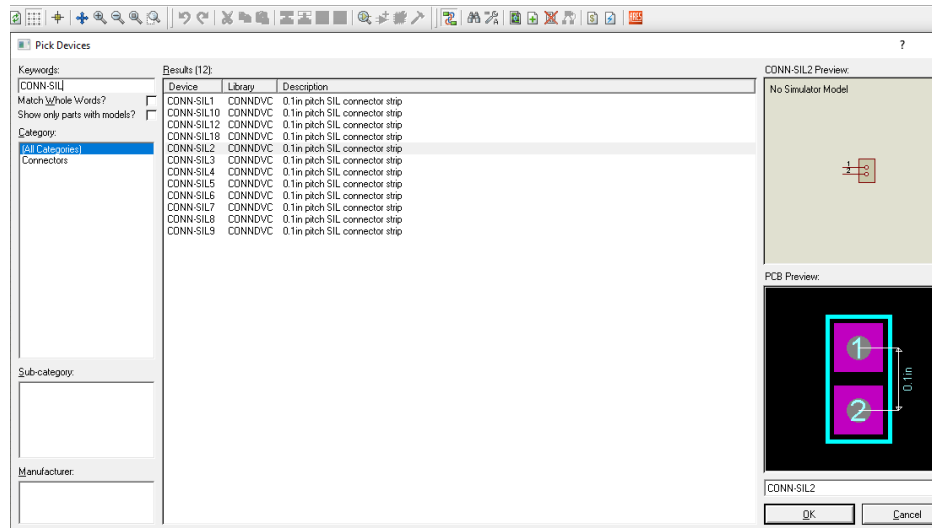


Figura 24. Terminal de bloque de dos salidas
Fuente: Elaboración propia (2019)

5. Seleccionar los componentes terminales de bloque con tres salidas para conectar los sensores infrarrojos Sharp, batería e interruptores en la opción selection mode, presionar la tecla pick from libraries, en la opción keywords, como muestra en la figura 25.

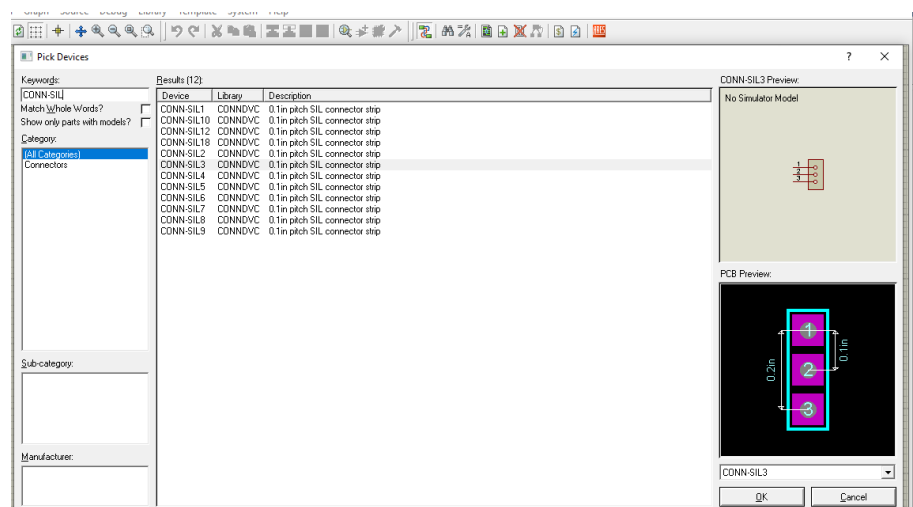


Figura 25. Selección de terminales de bloque con tres salidas
Fuente: Elaboración propia (2019)

- En la opción selection mode, presionar la tecla pick from libraries para la selección del componente terminal de bloque con diez salidas para conectar el sensor QTR8A, como muestra en la figura 26.

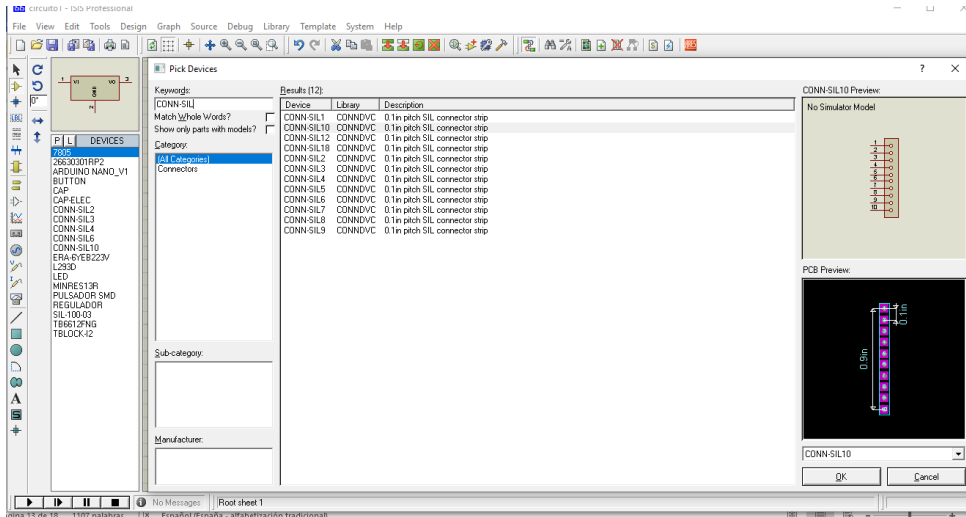


Figura 26. Terminal de bloque de diez salidas
Fuente: Elaboración propia (2019)

- Para la selección del Arduino nano y el drive TB6612FNG en el programa Isis Proteus descargar la librería de Arduino llamada simulinoV.3 misma que contiene dos archivos de librería de Arduino, descomprimir y pegar en la librería de Isis Proteus ubicada en disco local C donde se encuentra instalado el programa, como se muestra en la figura 27.

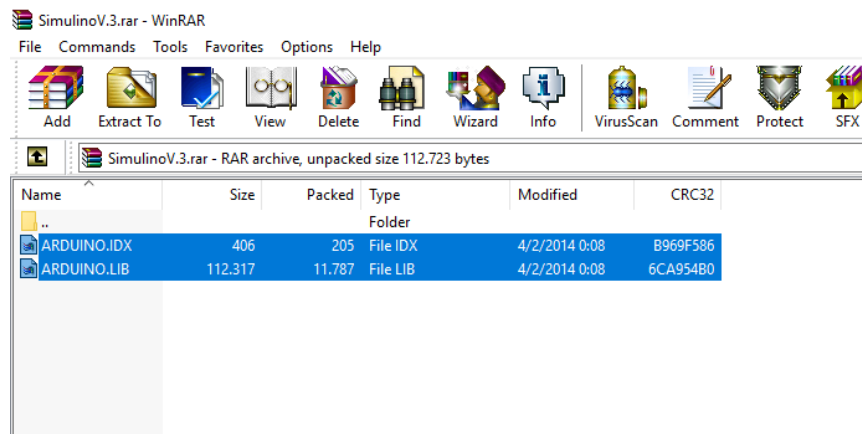


Figura 27. Librería de Arduino
Fuente: Elaboración propia (2019)

8. En la opción selection mode seleccionar el componente drive TB6612FNG como muestra en la figura 28.

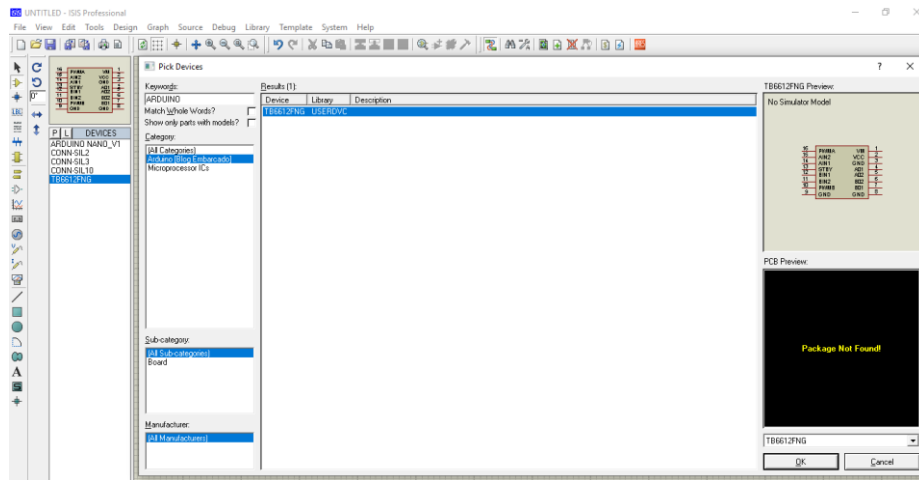


Figura 28. Driver TB6612FNG
Fuente: Elaboración propia (2019)

9. Todos los componentes seleccionados en la opción selection mode, aparecerán en Device selector, como muestra en la figura 29.

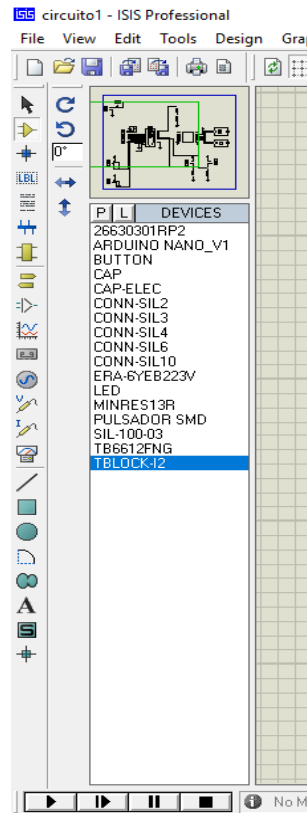


Figura 29. Elementos en dispositivos
Fuente: Elaboración propia (2019)

10. Insertar terminales de tipo default y asignar nomenclaturas que conecta al drive para evitar el cableado, conectado de forma inalámbrica, terminales de tipo POWER para la fuente y terminales tipo BROUND para tierra, como muestra en la figura 30.

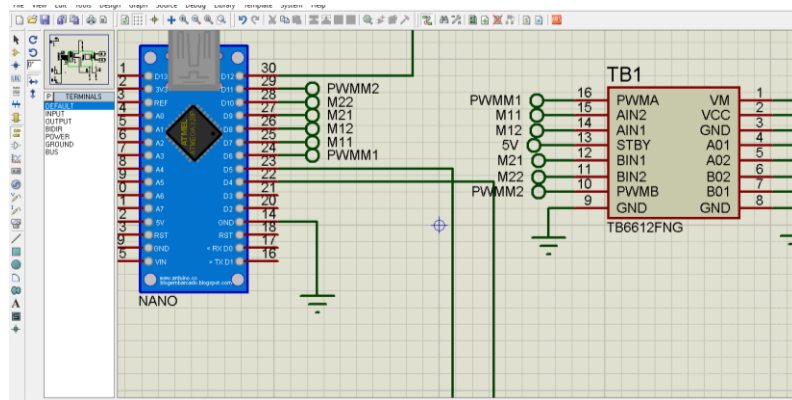


Figura 30. Conexión inalámbrica del Arduino con el drive TB6612FNG
Fuente: Elaboración propia (2019)

11. Conectar los terminales de bloques de 3 salidas J1 y J2 que representan los sensores Sharp a las entradas A6 y A7 del Arduino nano, como muestra en la figura 31.

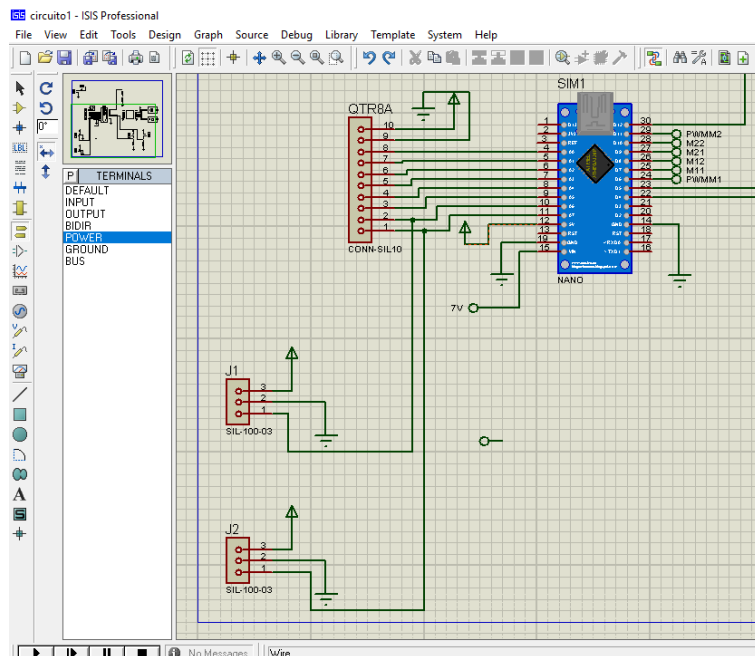


Figura 31. Sensores Sharp y Arduino
Fuente: Elaboración propia (2019)

- **Diseño de chasis en AutoCAD**

1. Para el diseño del chasis se utilizó el programa de diseño asistido por computadora AutoCAD 2016 utilizando comandos básicos como line y extrucción, se dimensionó 96mm de alto, 83mm del ancho de la parte superior donde va ubicado el sensor QTR8A, 103mm de ancho en la parte inferior y 2mm de espesor que equivale al espesor de la baquelita como muestra en la figura 32 y 33 los planos se los puede observar en el anexo 2.

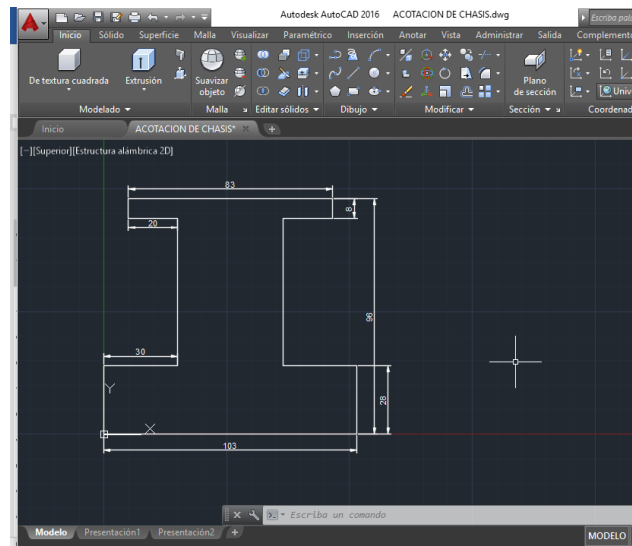


Figura 32. Diseño del chasis 2D en AutoCAD 2016

Fuente: Elaboración propia (2019)

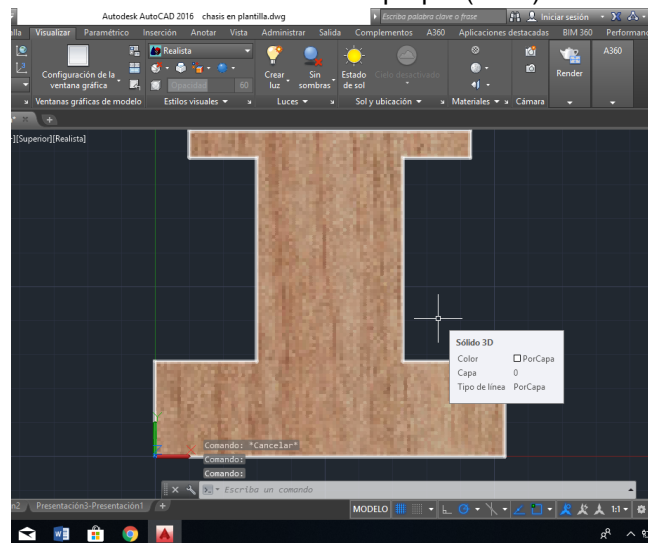


Figura 33. Diseño del chasis 3D en AutoCAD 2016

Fuente: Elaboración propia (2019)

- **Diseño en ares**

1. Para el diseño de la PCB en Ares, transferir el diseño realizado en Isis Proteus desde la opción Transfer To Ares, se podrá observar que todos los componentes que se utilizó en Isis aparecerán en la barra de componentes de Ares, como muestra en la figura 34.

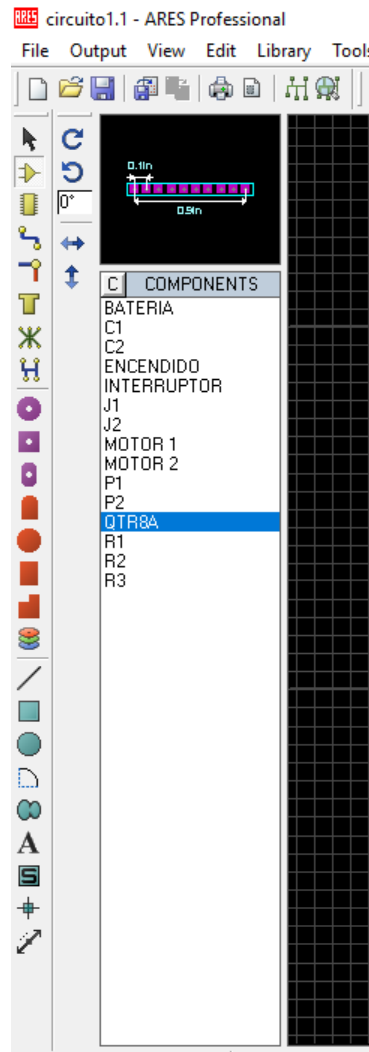


Figura 34. Programa Ares Proteus
Fuente: Elaboración propia (2019)

2. Definir el contorno de la placa seleccionando la opción 2D graphics line mode, opción board edge con las dimensiones realizadas en AutoCAD 2016, como muestra en la figura 35.

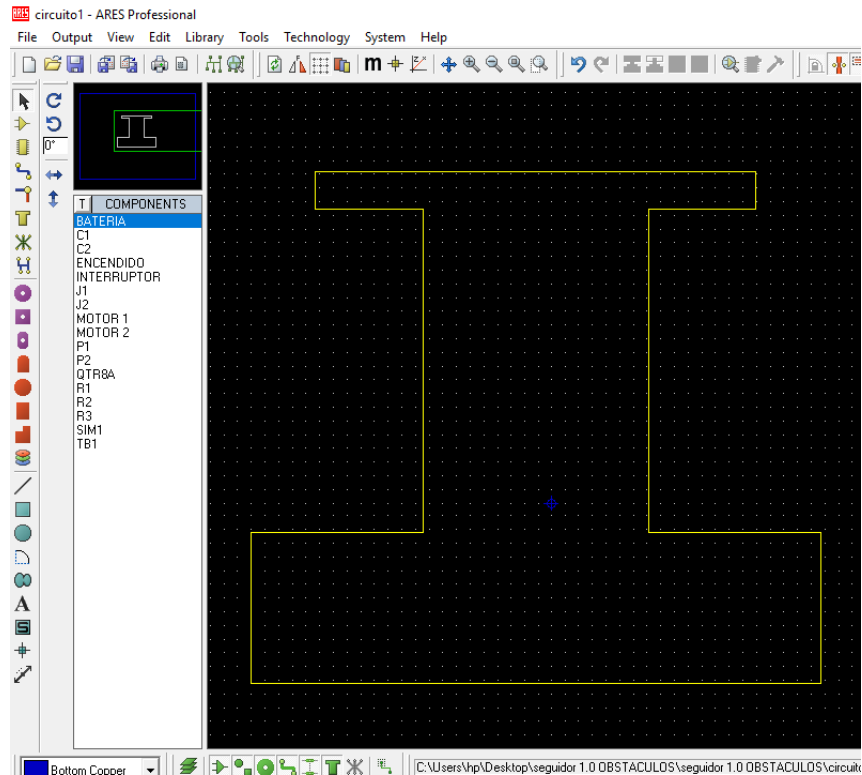


Figura 35. Chasis en Ares Proteus
Fuente: Elaboración propia (2019)

3. Arrastrar los elementos electrónicos hacia dentro del contorno del diseño, desde la opción selection mode y package mode donde se encuentran los microchips, como se muestra en la figura 36.

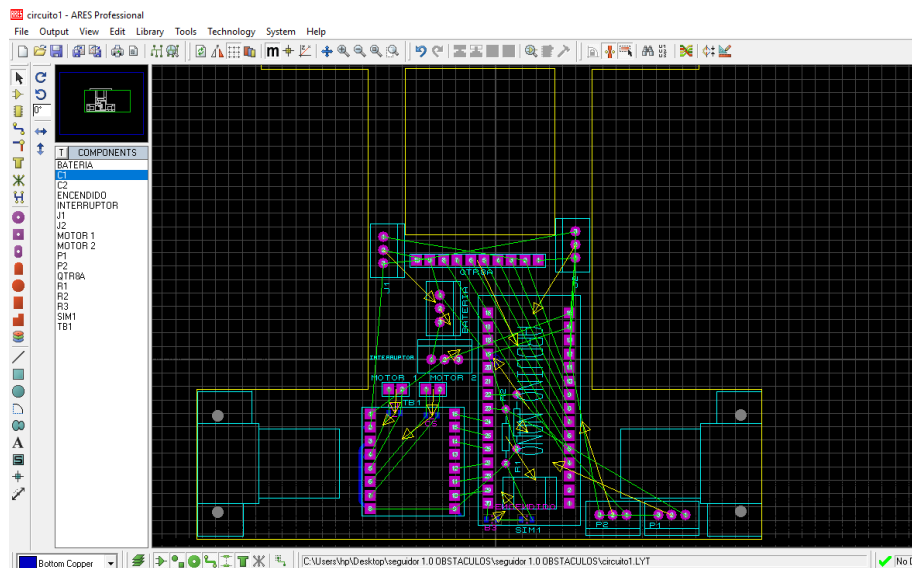


Figura 36. Elementos electrónicos en la placa de diseño.
Fuente: Elaboración propia (2019)

- Utilizar el trazador automático del menú herramientas para la unión automática de los elementos mediante líneas, obteniendo un diseño de placa de forma completa con el mínimo esfuerzo por parte del diseñador y finalmente simular el circuito en busca de posibles errores como muestra en la figura 37, los planos se los puede observar en el anexo 3.

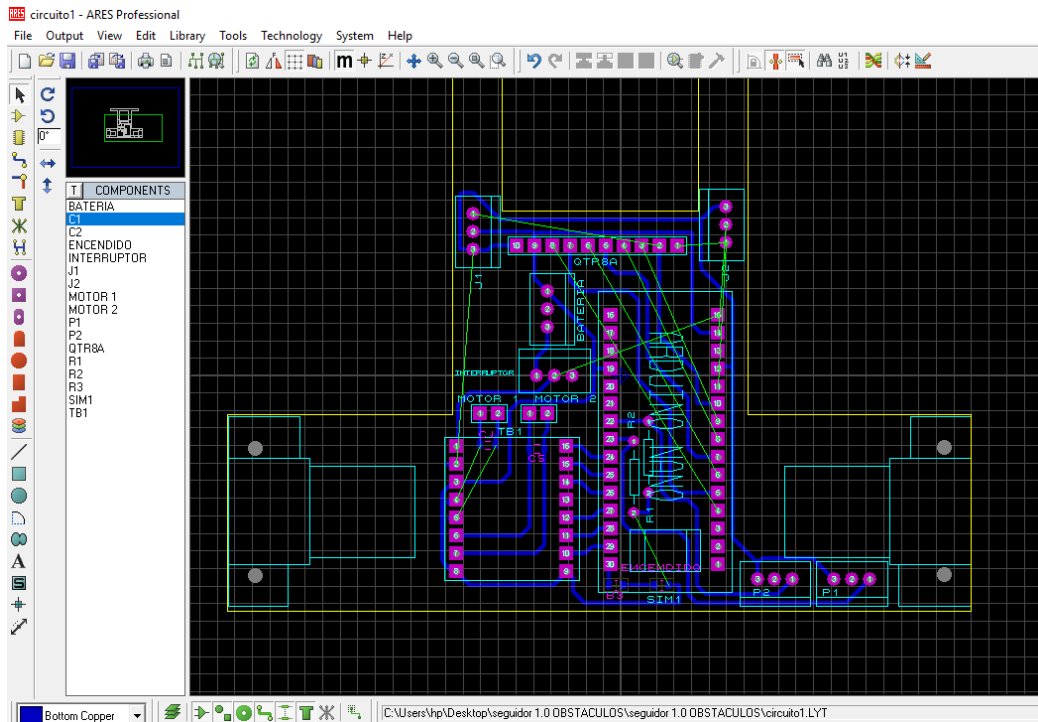


Figura 37. Diseño de robot en Isis Proteus
Fuente: Elaboración propia (2019)

4.2.2 Construcción

1. Una vez listo el diseño realizar la quema de la baquelita para ello utilizar los siguientes elementos como: papel termotransferible, baquelita, cloruro férrico, esponja de brillo, agua y sierra.
2. Realizar la impresión del diseño en papel termotransferible para posterior pegar a la baquelita en el lado de cobre, planchar a alta temperatura para que se adhiera el diseño al cobre de la baquelita, dejar enfriar 5 minutos aproximadamente sobre agua para que tenga mayor adherencia.
3. Retirar cuidadosamente el papel de la baquelita, con un marcador permanente marcar las líneas que se salieron durante la extracción del papel, como se muestra en la figura 38.

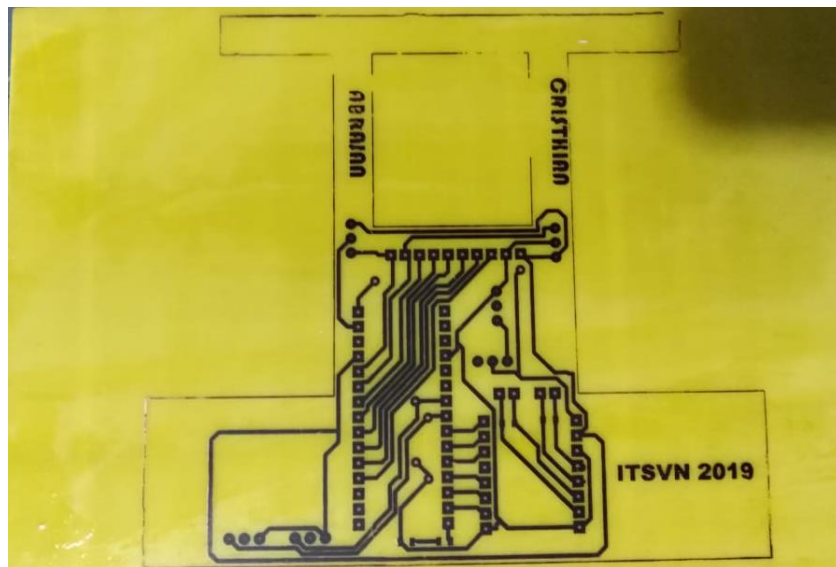


Figura 38. Diseño impreso en Baquelita
Fuente: Elaboración propia (2019)

4. Utilizar cloruro férrico para la remoción del cobre utilizando el método de agitar cuidadosamente el ácido con la baquelita dentro para que el cobre sea removido, como se muestra en la figura 39.



Figura 39. Baquelita en cloruro férrico
Fuente: Elaboración propia (2019)

5. Retirar la placa cuidadosamente y asegurar que se haya removido todo el cobre, limpiar la placa con abundante agua y con esponja de brillo para obtener una placa con buenos acabados, como se muestra en la figura 40.

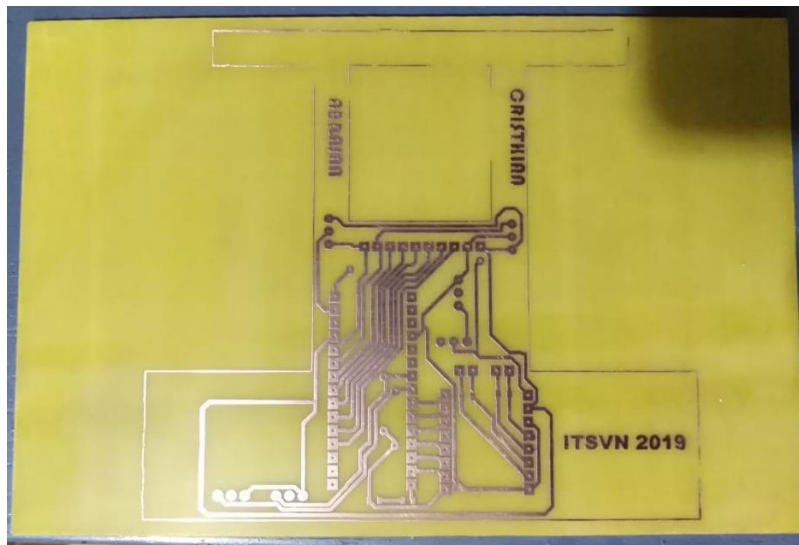


Figura 40. Remoción del cobre de la baquelita
Fuente: Elaboración propia (2019)

4.2.3 Implementación

1. Perforar la placa en los puntos donde va a ir cada elemento utilizando un taladro con broca de 1/32" después limpiar los agujeros con esponja de brillo e insertar los componentes para posterior soldar con caudín y estaño; en el caso de los motores sujetar con pernos como se muestra en la figura 41.



Figura 41. Sujeción con pernos del motor pololu
Fuente: Elaboración propia (2019)

2. Conectar los elementos electronicos con jumpers para posterior programar el robot seguidor, como se muestra en la figura 42.

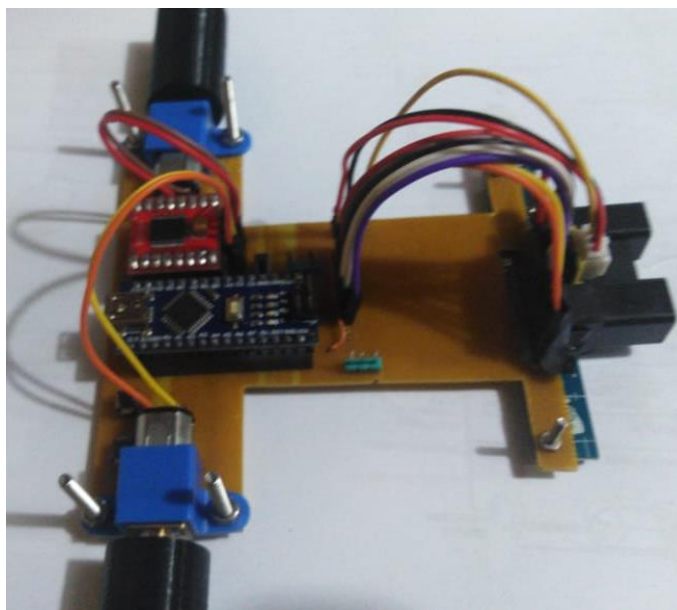


Figura 42. Robot seguidor de línea
Fuente: Elaboración propia (2019)

- **Programación**

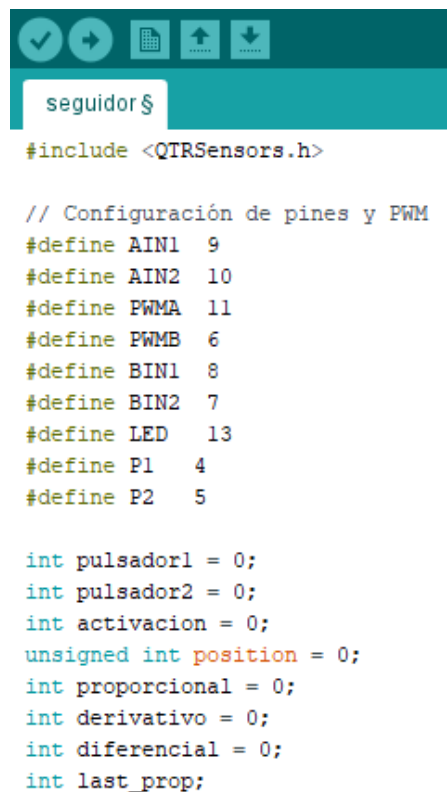
1. Programar el robot en el software Arduino el programa completo se encuentra en el anexo 1, para la programación del robot como primer punto se deberá descargar la librería QTRsensors library, como se muestra en la figura 43.



```
seguidor$  
#include <QTRsensors.h>  
  
// Configuración de pines y PWM
```

Figura 43. Librería QTRsensors
Fuente: Elaboración propia (2019)

2. Configurar los pines y PWM como se muestra en la figura 44.



```
seguidor$  
#include <QTRsensors.h>  
  
// Configuración de pines y PWM  
#define AIN1 9  
#define AIN2 10  
#define PWMA 11  
#define PWMB 6  
#define BIN1 8  
#define BIN2 7  
#define LED 13  
#define P1 4  
#define P2 5  
  
int pulsador1 = 0;  
int pulsador2 = 0;  
int activacion = 0;  
unsigned int position = 0;  
int proporcional = 0;  
int derivativo = 0;  
int diferencial = 0;  
int last_prop;
```

Figura 44. Configuración de pines E/S
Fuente: Elaboración propia (2019)

3. Configurar las entradas y salidas en la opción void setup() que trabajará a una velocidad de datos de bits por segundo de 9600 (baudios) para la transmisión de datos en serie, como se muestra en la figura 45.

```
void setup()
{
  pinMode(LED ,OUTPUT);
  pinMode(BIN2 ,OUTPUT);
  pinMode(BIN1 ,OUTPUT);
  pinMode(PWMB ,OUTPUT);
  pinMode(AIN1 ,OUTPUT);
  pinMode(AIN2 ,OUTPUT);
  pinMode(PWMA ,OUTPUT);
  pinMode(P1 ,INPUT);
  pinMode(P2 ,INPUT);

  Serial.begin(9600);

  for ( int i=0; i<50; i++)
  {
    digitalWrite(LED, HIGH); delay(20);
    qtra.calibrate();
    digitalWrite(LED, LOW); delay(20);
  }
  delay(200);
}
```

Figura 45. Configuración E/S
Fuente: Elaboración propia (2019)

4. En la opción void loop() se encuentra el programa el cual se repetirá cíclicamente una vez que se haya presionado el pulsador1, antes de enviar el programa a la placa verificar el puerto de salida y copilar para verificar que no haya fallos en la programación, como se muestra en la figura 46.

```
void loop()
{
  pulsador1 = digitalRead(P2);

  if(pulsador1 == HIGH)
  {
    delay(300);

    while(digitalRead(P2) == HIGH)
    {
      digitalWrite(LED, HIGH);
      delay(10);
    }
    digitalWrite(LED, LOW);
    activacion = !activacion;
  }

  if(activacion == 1)
  {
    int ADC_SHARP = analogRead(A6);
    if ( ADC_SHARP >= 50 )
    {
      Motoriz(100);
      Motorde(50);
      delay(50)
    }
  }
}
```

Figura 46. Puerto void loop()
Fuente: Elaboración propia (2019)

5. Calibrar los sensores mediante la acción de pasar repetidas veces por la línea negra de fondo blanco sin levantar los sensores, en el software Arduino utilizar la función; `qtra.calibrated ()` de la librería (QTR sensors) para la calibración, la función; `qtr.read(sensor values)` hace lectura de cada sensor se guarda en un rango de 0 a 1000 y la función `int position:` devolverá la posición de sensores, como se muestra en la figura 47.

```
    unsigned int sensorValues[8];
    position = qtra.readLine(sensorValues);

    //Serial.println(position);

    proporcional = position - 3500;
    derivativo = proporcional - last_prop;
    last_prop = proporcional;
    int diferencial = ( proporcional * KP ) + ( derivativo * KD ) ;

    int vmotor_derecho = Velmax + diferencial;
    int vmotor_izquierdo = Velmax - diferencial;

    Motoriz(vmotor_izquierdo);
    Motorde(vmotor_derecho);
}

if(activacion == 0)
{
    Motoriz(0);
    Motorde(0);
    proporcional = 0;
}
}
```

Figura 47. Opciones de librería QTRsensors
Fuente: Elaboración propia (2019)

5. CONCLUSIONES

Se empleó la tracción de tipo diferencial para el desplazamiento en una línea recta y el movimiento rotacional permitiendo que cada rueda actúe de forma independiente mediante la señal enviada desde el sensor al driver.

El diseño y simulación del robot se empleó el software Isis Proteus que permitió simular el circuito electrónico, identificando posibles errores a fin de poder corregirlos y enviarlos a Ares Proteus.

El software Ares Proteus y AutoCAD se emplearon para el diseño del chasis del robot, y el diseño en PCB permite un buen dimensionamiento de cada elemento optimizando espacio en el chasis.

6. RECOMENDACIONES

Es recomendable utilizar llantas de goma estándar color negro ya que tienen un mejor agarre y son adherentes para una buena tracción, obteniendo mayor rendimiento al momento de desplazarse.

Descargar la librería de Arduino llamada simulinoV.3 y agregar a la librería de Isis Proteus que contienen la placa de Arduino nano y drive TB6612FNG, para poder emplearlos en el diseño.

Evitar peso innecesario en el chasis del robot, utilizar una batería lipo de bajo voltaje 7.4v para evitar usar pilas grandes que generen mayor espacio y peso en el robot dificultando así su ágil desplazamiento.

7. REFERENCIAS

- ARDUINO. (2013). *Arduino leonardo*. Recuperado el 09 de noviembre de 2019, de: https://www.arduino.cc/en/Main/Arduino_BoardLeonardo
- ARDUINO. (2013). *Arduino uno*. Recuperado el 09 de noviembre de: <https://arduino.cl/arduino-uno/>
- Bembibre, C. (2010). *Definicion ABC*. Recuperado el 08 de noviembre de 2019, de: <https://www.definicionabc.com/tecnologia/locomocion.php>
- Burbano, P. (2013). *Didactica.com* . Recuperado el 11 de Marzo de 2014, de: <http://www.didactica.com/recursos/reciclaje>
- Burbano, P. (2014). *Proyectos*. Quito: Vida Nueva .
- Blum, J. (2014). *Arduino a fondo*. España: Amaya.
- CER. (2019). *Reglamento seguidor de línea*. Recuperado el 09 de noviembre de 2019, de: <https://cer.cedia.edu.ec/dmdocuments/CER/2019/Reglamentos/ReglamentoSeguidorDeLineaBasicoJr.pdf>
- Elarduino. (2019). *Arduino nano*. Recuperado el 09 de NOVIEMBRE de 2019, de: <https://www.elarduino.com/nano/>
- ElectronicaStore. (2018). *Seguidor de linea Turing*. Recuperado el 08 de noviembre de 2019, de: <https://electronicastore.net/producto/seguidor-de-linea-turing-perfecto-para-iniciar-en-el-mundo-de-la-robotica/>
- Electrontools. (2019). *Arduino mega2560*. Recuperado el 09 de noviembre de 2019, de: <https://www.electrontools.com/Home/WP/2018/06/19/arduino-mega-2560-caracteristicas/>
- Sirpa, E. (2019). *Medium*. Recuperado el 10 de noviembre de 2019, de: <https://medium.com/@ericksirpa/robots-m%C3%A9todos-de-locomoci%C3%B3n-terrestre-2964c68a9523>

- Frank, M. (2019). *Ingmecafenix*. Recuperado el 09 de noviembre de 2019, de: <https://www.ingmecafenix.com/electronica/puente-h-control-motores/>
- Garcia, V. (2014). *Diarioelectronicohoy*. Recuperado el 09 de noviembre de 2019, de: <https://www.diarioelectronicohoy.com/blog/el-puente-h-h-bridge>
- Geekbotelectronics. (2019). *Motores DC*. Recuperado el 09 de noviembre de 2019, de Motores DC: <http://www.geekbotelectronics.com/motores-de-dc/>
- INCIBE-CERT. (2017). *La electrónica e sus inicios*. Obtenido de INCIBE-CERT-ES: <https://www.incibe-cert.es/blog/robots-y-drones-industria-40>
- La nacion. (2016). *Competencia de robot en Ecuador*. Recuperado el 05 de septiembre de 2019, de: <https://lanacion.com.ec/robots-ecuador-se-enfrentan-una-inusual-competencia-la-se-comparten-conocimientos/>
- Moctezuma, L. (2016). *Locomocion tipo triciclo*. Recuperado el 09 de noviembre de 2019, de: <https://es.slideshare.net/luisalfredomoctezuma/pascual/locomocin-triciclo>
- Nylampmechatronics. (2018). *Drive puenteH TB6612FNG*. Recuperado el 09 de noviembre de 2019, de: <https://nylampmechatronics.com/drivers/200-driver-puente-h-tb6612fng.html>
- Pololu. (2019). *Micro gear motor*. Recuperado el 09 de noviembre de 2019, de: <https://www.pololu.com/category/60/micro-metal-gearmotors>
- Pololu. (2019). *Servos*. Recuperado el 09 de noviembre de 2019, de: <https://www.pololu.com/category/205/servos>
- POLITECNICA, S. U. (2017). *Desarrollo de un prototipo de robot movil de competencia multi-categoría*. Quito.

- Proyecto, M. (2018). *Proyecto mecánico*. Recuperado el 09 de noviembre de 2019, de: <https://sites.google.com/site/proyectomecanicovth/>
- Ortiz, D. (2016). *Roboticaparaseguimiento de líneas*. España: Roboticeso. (2019). *Seguidor de línea*. Recuperado el 09 de noviembre de 2019, de: <https://roboticeso.wordpress.com/>
- Schaefer. (2016). *Vehículo de guiado automático*. Recuperado el 08 de noviembre de 2019, de: <https://www.ssi-schaefer.com/es-es/productos/transportadores/agv---veh%C3%ADculo-de-guiado-autom%C3%A1tico/agv---veh%C3%ADculos-de-guiado-autom%C3%A1tico--182648>
- SOMIM. (2019). *Reglamento seguidor de línea*. Recuperado el 09 de noviembre de 2019, de: http://somim.org.mx/congreso2019/Reglamento_Competicion_Siguelinea.pdf
- Superrobotica. (2019). *Motores y servos*. Recuperado el 09 de noviembre de 2019, de: <http://www.superrobotica.com/motores.htm>
- tdrobotica. (2019). *Sensores de línea*. Recuperado el 09 de noviembre de 2019, de: <http://tdrobotica.co/sensor-de-linea-qtr-8a-analogo-arreglo-de-8/173.html>
- tdrobotica.co. (2014). *Seguidor de línea velocista*. Recuperado el 08 de noviembre de 2019, de: <http://aprender.tdrobotica.co/seguidor-de-lineaprofesional/>
- Tecnopura. (2019). *Sensor de distancia analógico Sharp*. Recuperado el 09 de noviembre de 2019, de: <http://www.tecnopura.com/producto/sensor-de-distancia-analogico-sharp-gp2y0a21yk0f-para-medicion-de-10-a-80-cm/>
- Tojeiro, G. (2014). *Taller de arduino*. España: Marcombo.
- uvpr-robótica. (2013). *Sistemas de locomoción de robots*. Recuperado el 09 de noviembre de 2019, de *Sistemas de locomoción de robots*:

<http://uvpr-robotica.blogspot.com/2013/10/metodos-de-locomocion-terrestres.html>

Valencia, V. J. (2009). *Traccion diferencial*. Recuperado el 09 de noviembre de 2019, de: <https://www.redalyc.org/pdf/849/84916680034.pdf>

8. ANEXOS

Anexo1. Programa del robot en Software Arduino

```
#include <QTRSensors.h>

// Configuración de pines y PWM
#define AIN1 9
#define AIN2 10
#define PWMA 11
#define PWMB 6
#define BIN1 8
#define BIN2 7
#define LED 13
#define P1 4
#define P2 5

int pulsador1 = 0;
int pulsador2 = 0;
int activacion = 0;
unsigned int position = 0;
int proporcional = 0;
int derivativo = 0;
int diferencial = 0;
int last_prop;

// VARIABLES A CALIBRAR
//*****
float KP = 0.029;
float KD = 0.049;
int Velmax = 70;
//*****

QTRSensorsAnalog qtra((unsigned char[]) { 1, 2, 3, 4, 5, 6 }, 6, 4);

void setup()
{
    pinMode(LED ,OUTPUT);
    pinMode(BIN2 ,OUTPUT);
    pinMode(BIN1 ,OUTPUT);
    pinMode(PWMB ,OUTPUT);
    pinMode(AIN1 ,OUTPUT);
    pinMode(AIN2 ,OUTPUT);
    pinMode(PWMA ,OUTPUT);
    pinMode(P1 ,INPUT);
    pinMode(P2 ,INPUT);

    Serial.begin(9600);

    for ( int i=0; i<50; i++)
    {
        digitalWrite(LED, HIGH); delay(20);
        qtra.calibrate();
        digitalWrite(LED, LOW); delay(20);
    }
    delay(200);
}
```

```

void loop()
{
  pulsador1 = digitalRead(P2);

  if(pulsador1 == HIGH)
  {
    delay(300);

    while(digitalRead(P2) == HIGH)
    {
      digitalWrite(LED, HIGH);
      delay(10);
    }
    digitalWrite(LED, LOW);
    activacion = !activacion;
  }

  if(activacion == 1)
  {
    int ADC_SHARP = analogRead(A6);
    if ( ADC_SHARP >= 50 )
    {
      Motoriz(100);
      Motorde(50);
      delay(50)

      Motoriz(50);
      Motorde(100);
      delay(50)

      Motoriz(100);
      Motorde(100);
      delay(50)

      Motoriz(50);
      Motorde(100);
      delay(50)

      Motoriz(100);
      Motorde(50);
      delay(50)
    }
  }
}

```



```

    unsigned int sensorValues[8];
    position = qtra.readLine(sensorValues);

    //Serial.println(position);

    proporcional = position - 3500;
    derivativo = proporcional - last_prop;
    last_prop = proporcional;
    int diferencial = ( proporcional * KP ) + ( derivativo * KD ) ;

    int vmotor_derecho = Velmax + diferencial;
    int vmotor_izquierdo = Velmax - diferencial;

    Motoriz(vmotor_izquierdo);
    Motorde(vmotor_derecho);
}

if(activacion == 0)
{
    Motoriz(0);
    Motorde(0);
    proporcional = 0;
}
}

void Motoriz(int value)
{
    if ( value >= 0 )
    {
        digitalWrite(BIN1,LOW);
        digitalWrite(BIN2,HIGH);
    }
    else
    {
        digitalWrite(BIN1,HIGH);
        digitalWrite(BIN2,LOW);
        value *= -1;
        //value = 0;
        if(value >= 25)
        {value = 25;}
    }

    if(value >= 255){value = 255;}
    if(value <= 0){value = 0;}

    analogWrite(PWMB,value);
}

```

```
void Motorde(int value)
{
  if ( value >= 0 )
  {
    digitalWrite(AIN1,LOW);
    digitalWrite(AIN2,HIGH);
  }
  else
  {
    digitalWrite(AIN1,HIGH);
    digitalWrite(AIN2,LOW);
    value *= -1;
    //value = 0;
    if(value >= 25 )
    {value = 25;}
  }

  if(value >= 255){value = 255;}
  if(value <= 0){value = 0;}

  analogWrite(PWMA,value);
}
```

Fuente: Elaboración propia (2019)

1

2

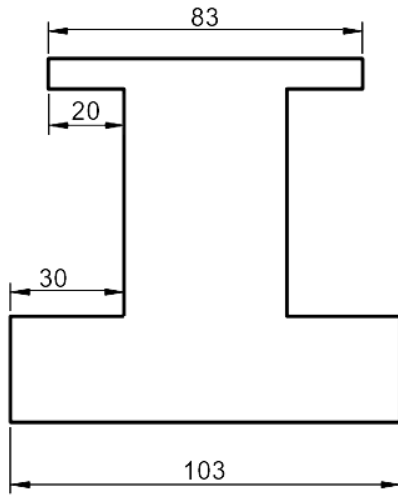
3

4

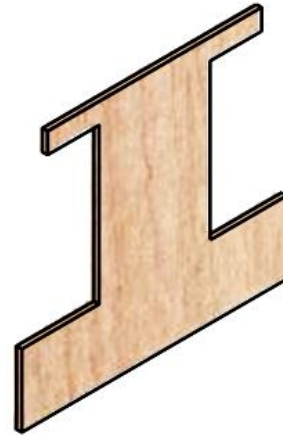
A

Anexo 2. Diseño de chasis en AutoCAD 2016

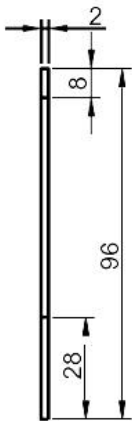
B



C



D

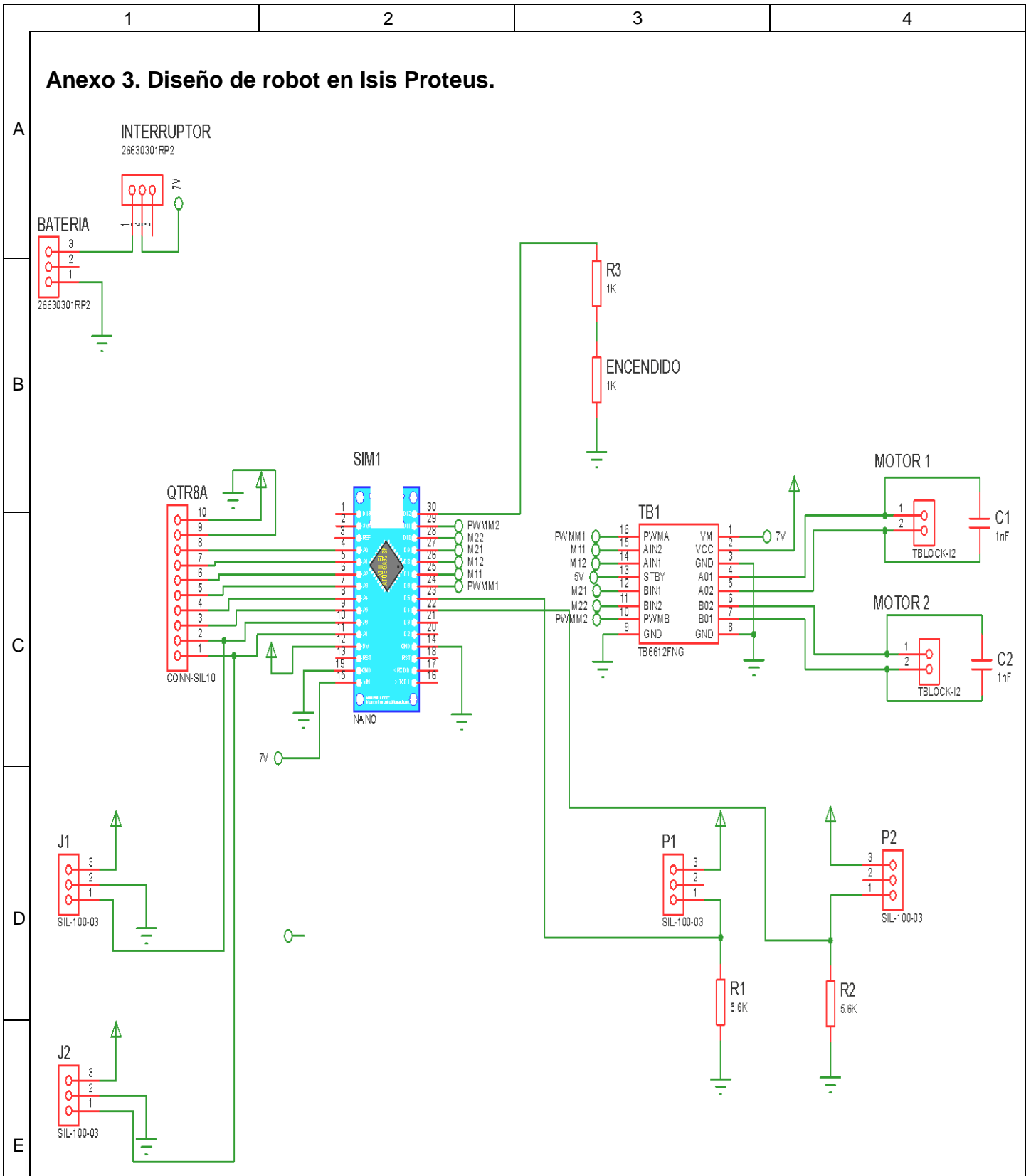


E

				Tolerancia		Material	Resina fenólica
					Fecha	Nombre	Denominación
				Dib.	03/11/2019	Abrajan C.	Chasis en baquelita
				Rev.	02/12/2019	Ing. Machay B.	
				Aprov.	02/12/2019	Ing. Machay B.	
				Instituto Superior Tecnológico	Número de Dibujo		Escala
				Vida Nueva	01 de 03		1:2
Edición	Modificación	Fecha	Nombre		Sustitución		



Anexo 3. Diseño de robot en Isis Proteus.



				Tolerancia		Material		
					Fecha	Nombre	Denominación	
				Dib.	03/11/2019	Abrajan C.	Diseño en Isis	
				Rev.	02/12/2019	Ing. Machay B.		
				Aprov.	02/12/2019	Ing. Machay B.		
				Instituto Superior Tecnológico Vida Nueva			Número de Dibujo	Escala
							02 de 03	1:1
Edición	Modificación	Fecha	Nombre				Sustitución	



