INFORME TÉCNICO

ESCUELA DE EDUCACION SECUNDARIA TECNICA N°5

“2 DE ABRIL” – TEMPERLEY – BUENOS AIRES

NOMBRE DEL PROYECTO:

ROBOT SUMO

GRUPO: ***THE SPARTANS***

MATERIA : PROYECTO Y DISEÑO ELECTRÓNICO

AUTORES: Lucas Leandro Martinez, Santiago Matías Pena
Pablo Emanuel Gonzalez Flores, Dante Tomas Figueroa

Alumnos:

Nombre y mail:

Lucas Leandro Martinez

lucasleandromartinez@industrialdetemperley.edu.ar

Santiago Matías Pena

santiagomatiaspena@industrialdetemperley.edu.ar

Pablo Emanuel Gonzalez Flores

pabloemanuelgonzalezflores@industrialdetemperley.edu.ar

Dante Tomas Figueroa

figueroadantethomas@industrialdetemperley.edu.ar

PROFESOR: ING. MARTIN LEGUIZAMON

Descripción del Proyecto

Este Proyecto integra las diferentes áreas de la electrónica ya que se trabaja en el área de desarrollo de hardware, diseñando y construyendo las plaquetas y circuitos electrónicos necesarios para el funcionamiento del robot, desarrollo de software el cual es necesario implementar para cumplimentar la parte de funcionamiento del robot. Además, gracias a esto también se puede trabajar en el área de Robótica. Otra de las materias en las que se opera es en diseñó 3D y montaje del robot, lo que nos sirve como complemento del armado y funcionamiento del robot.

Este proyecto trata acerca de construir un robot de competencia, el cual es de control autónomo y cuenta con, dos sensores, dos driver para motores, dos motores, un batería de lipo y un reductor de voltaje. Este robot sumo es básicamente un auto de peleas sobre una plataforma el cual tiene que cumplir con determinados requisitos de peso, adherencia de sus ruedas y diseño de su pala o cuchilla.

Este complejo instrumento de competencia es muy fácil de utilizar ya que se basa en presionar el interruptor para encenderlo y tocar el pulsador para su activación de esta manera los sensores comenzarán a medir distancia y dependiendo la distancia que detecten enviaran una señal a la placa madre denominada ESP32, este (con la información recibida) les enviará una señal a los motores que actuaran en consecuencia. Todo esto de manera autónoma y sin control humano.

Tiene como objetivo principal servir como proyecto de aprendizaje para cumplimentar las enseñanzas teóricas y poner en práctica los distintos procesos de la electrónica y robótica. Así como también ampliar nuestros conocimientos acerca de los componentes utilizados en electrónica, aprender de su funcionamiento, corregir las fallas que pueda tener, además de ampliar y complementar nuestros conocimientos.

Diagrama de Gantt

Diagrama de tareas

Este diagrama de tareas se distribuye en 4 meses dividiendo cada una de sus etapas en una tarea a realizar por mes.

Áreas: Diseño y Fabricación del PCB - Programación - Diseño 3D - Montaje

Primer mes:

Asignados a la tarea: Lucas Martinez y Santiago Pena
tarea a realizar - Diseño y Fabricación del PCB

En la primera semana se debe averiguar en investigar los componentes específicos necesarios para realizar el proyecto previamente acordado con los 4 integrantes del grupo. También se deberá especificar la cantidad de componentes y de cada uno saber o averiguar sus conexiones y tipos de pines específicos así como sus protocolos de comunicación, todo esto se hará con la finalidad de evitar posibles malas conexiones o conexiones erróneas.

La segunda semana se deberá comenzar con el diseño del PCB, primeramente tendremos que hacer las conexiones y distribución de circuitos pertinentes. Hay que tener en cuenta que no todos los componentes van en la placa y van conectados mediante pineras. Luego de eso podemos empezar con las conexiones de las pistas las cuales tienen como regla a respetar la distancia entre pistas de 0,5mm y no trazar pistas en los bordes de la placa.

Una vez terminado el diseño del circuito impreso la tercera semana deberemos comenzar con su fabricación trabajando, primero en preparar la placa para para el proceso de transferencia térmica limpiandola de las

impurezas que pueda tener.
Luego de verificar la correcta adherencia de las pistas a la placa y que ninguna se haya cortado tocará sumergirla en una ácido especial el cual se encargará de quitar los sobrantes de cobre que no sean parte de las pistas (esto puede tomar algo de tiempo 30 min aproximadamente).
Lo siguiente a realizar es el proceso de agujereado de los pads de la placa, para esto deberemos elegir una mecha adecuada acorde al tamaño del pad (un poco más chica) por lo general 0.75 o 1 mm son tamaños adecuados para realizar este proceso.

En la cuarta semana tras haber agujereado los pads se deberá terminar la placa soldando los componentes necesarios que anteriormente habíamos puesto en nuestro diseño, debemos tener cuidado a la hora de soldar nuestros componentes ya que no debemos calentar demasiado al componente ni a la pista porq esta se puede desprender. También debemos evitar interconexiones entre componentes mediante la soldadura.
Ya realizado este proceso debemos pasar a la etapa de testeo y comprobación para verificar que no haya cortocircuitos a puentes no deseados además de comprobar que el la placa circule el voltaje adecuado. Con esto terminamos el primer mes.

Segundo mes:

Asignado a la tarea: Pablo Gonzalez
Tarea a realizar - Programación

La primera semana se deberá realizar pequeños códigos o partes de códigos de prueba para los sensores y los drivers de esta manera se nos hara más fácil hacer el código general, ya que dividiéndolo por partes podemos verificar el funcionamiento del los componentes en un protoboard o en la misma placa y de esta manera probar si por separado funciona todo como debería.

La segunda semana debemos hacer un código general que abarque los otros dos realizados en la primera semana de esta forma lograremos hacer un código más grande sin perdernos y podremos probar el funcionamiento general de los componentes.

La tercera semana debemos probar todo junto, la placa, los componentes y las conexiones todo junto con los códigos realizado las semanas anteriores, cabe aclarar que antes debes verificar tener instalados los driver para el esp32 en el arduino IDE, elegir y verificar que reconozca el puerto al que se lo conecto y recien ahi subirlo correctamente. Realizamos las pruebas, si funciona todo correctamente habremos terminado, de no ser así habrá que realizar arreglos.

La cuarta semana debemos verificar que todo funcione correctamente de ser así debemos intentar mejorar el código, hacer estrategias para las peleas y optimizarlo, de lo contrario debemos corregir los errores que hayan surgido arreglar los problemas que pueda tener y mejorar su funcionamiento.

Tercer mes:

Asignado a la tarea: Dante Figueroa
tarea a realizar - Diseño 3D

Esta tarea se dividirá en etapas de dos semanas las primeras semanas lo que haremos será decidir que diseño tendrá el robot, haremos varios prototipos, minimo 2 y de esos decidiremos cual es nuestro favorito o el que mejor haya quedado, después de esto tomaremos las medidas de los componentes que deben ir dentro del 3D esto puede ser tamaños y

medidas de la placa, de los motores, del interruptor, bateria y drivers.
Las últimas dos semanas lo que haremos será adaptar el prototipo a las medidas tomadas, terminar de definir los detalles como agujeros para tornillos y etc. Ya luego mandaremos imprimir el 3D una vez terminado.

Cuarto mes:

Asignado a la tarea: Lucas Martinez y Dante figueroa
tarea a realizar - Montaje

Para la etapa de montaje dividiremos esto en áreas:

-**Área de Chasis**:
Esto haremos la primera semana del cuarto mes, preparar el chasis para el 3D. El chasis se basará en una chapa fina y lisa la cual tendremos que cortar a medida (como máximo 20x20cm).
También debemos hacerle los agujeros para los tornillos de la métrica que escojamos y eliminar las impurezas que tenga la chapa para que quede perfectamente preparada para poder armar el resto del robot.

-**Área de Montaje de 3D**
En la segunda semana atornillamos el 3d a la parte de la chapa trabajada la semana anterior. dejaremos todo de forma correcta y correspondiente, cortaremos alguna parte sobrante y ponemos todo a medida.

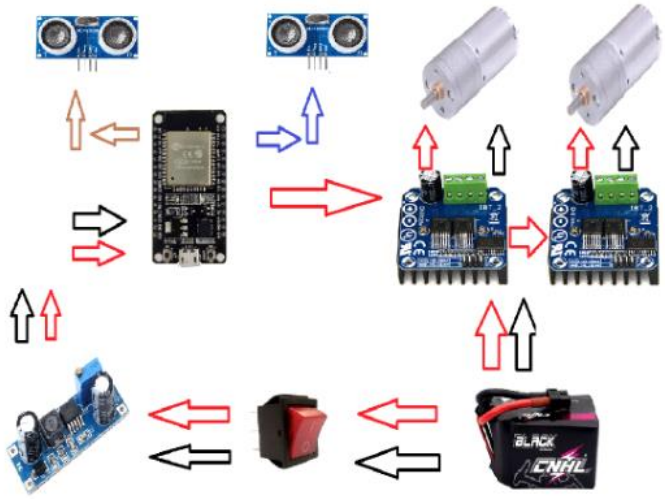
-**Area de Montaje de los Circuitos Eléctricos**
En la tercera semana introduciremos en el 3d todos los circuitos correspondientes entre ellos la placa, los motores, los drivers, los cables de los motores y el cableado de la alimentación.
también acomodaremos los drivers y la placa en las posiciones que pensemos que sean las más óptimas.

- **Área de Pruebas**
En esta etapa, ya la etapa final, se probará de nuevo el funcionamiento del robot pero esta vez con todo el Montaje ya realizado, se verifica que sea óptimo y no afecte a la movilidad del robot y ante cualquier error que muestre se lo intentará corregir.

Lista de materiales

-ESP32	$5.500
-HC-SR04	$2.200
-STEPDOWN	$3.500
-BATERIA LIPO 14 V	$65.000
-PINERAS	$3.500
-INTERRUPTOR	$1.200
-MOTORES	$20.000
-DISEÑO 3D	$5.600
-BORNERAS	$1.000
-DRIVERS	$15.000
-PLACA DE PERTINAX	$4.700
TOTAL:	$142.200

Diagrama en bloques



Como se puede apreciar en la imagen tenemos la disposición mediante flechas de las conexiones del robot, también tenemos la distribución de sus componentes más importantes. Contamos con la división en 4 bloques de la parte de alimentación, la parte de potencia que sería el área donde se encuentran los motores y los drivers que controlan dichos motores.

La tercera etapa es en la cual se encuentra la placa ESP32 que controla y distribuye la información que mediante la programación se envía a todos los circuitos integrados que dispone este robot.
La cuarta etapa cuenta con los dos sensores que, a gran escala, serían como los ojos del robot ya que será mediante lo que estos detecten como actuara o qué acción tomará el robot.

Todo esto estará conectado mediante una placa diseñada y fabricada con la herramientas necesarias que nos permitan hacer un circuito el cual dispondrá de la placa esp32, pineras y bornes para conexiones de cables tanto de alimentación como de pines de datos y etapas que permitan la interconexión entre componentes externos como los sensores, los drivers y el step down.

Diagrama de Conexiones

Aquí se muestra la disposición de las conexiones de los componentes, es decir, en este diagrama podemos ver más en detalle como va conectado exactamente cada componente y en que
pines va conectado.

A diferencia del otro esquema que era más general, en este se muestra cada pin tanto de alimentación como de datos. En el caso de los sensores podemos ver que disponen de 4 pines VCC; GND; ECHO; TRIG; donde VCC y GND son los pines de alimentación positivo y negativo y los pines ECHO y TRIG son los pines de transmisión y recepción de datos.

Por otro lado los driver muestran la disposición y conexión de sus fines más importantes y sus respectivas alimentaciones. Ya que cuenta con varias entradas de alimentación debido a que (en la parte de las borneras) tenemos las borneras de las puntas en donde una es la entrada de alimentación positiva de la batería y la otra es la salida de alimentación positiva que va a los motores.
Por otra parte en las primeras tenemos las entradas de alimentación para la parte lógica y los dos pines que van a determinados pines del ESP32 los cuales se utilizan para controlar los motores. Los demás pines serán explicados más en detalles junto con las explicación de funcionamiento del driver.

Ahora tenemos el reductor de voltaje o STEP DOWN mediante el cual daremos energía a la parte lógica, este funciona mediante 4 pines. dos de entrada, dos de salida, por los dos de entrada ingresaran los 16v de la batería y por los dos de salida saldrá el voltaje al cual lo hayamos

regulado, en este caso serán 5v.
En este caso los 5v van al esp32 y al resto del circuito lógico. Cabe aclarar que cada step down cuenta con un voltaje mínimo y máximo el cual será explicado más adelante.

Ya por último queda explicar los pines de la batería e interruptor en el cual se conectaran negativo y positivo (GND; VCC) y que antes de conectarse al step down tendrán una salida por la que alimentarán con 16v a los motores ya que si les damos energía con 5v sería muy poco y no tendrían la fuerza suficiente para moverse.

ESP32

El ESP32 es un microcontrolador de bajo costo y bajo consumo de energía que integra Wi-Fi y Bluetooth, lo que lo hace ideal para una variedad de aplicaciones en el ámbito de la electrónica y la conectividad inalámbrica. Fue desarrollado por Espressif Systems, una empresa china especializada en soluciones de conectividad IoT (Internet de las cosas).

Algunas características clave del ESP32 incluyen:

Procesador de Doble Núcleo: El ESP32 tiene un procesador dual-core Xtensa LX6, lo que permite una mayor capacidad de procesamiento y multitarea.

Wi-Fi y Bluetooth Integrados: El ESP32 incluye un módulo Wi-Fi IEEE 802.11 b/g/n y un módulo Bluetooth 4.2 BLE (Bluetooth de baja energía), lo que facilita la conectividad inalámbrica en proyectos IoT.

Puertos de Entrada/Salida (GPIO): Ofrece una variedad de pines GPIO que pueden ser utilizados para conectar sensores, actuadores y otros dispositivos electrónicos.

Interfaces de Comunicación: Incluye interfaces para comunicación serie (UART, SPI, I2C, etc.), lo que facilita la interconexión con otros dispositivos.

Periféricos Integrados: Cuenta con una amplia variedad de periféricos integrados, como convertidores analógico a digital (ADC), temporizadores, PWM, y más.

Memoria Flash: Tiene una memoria flash incorporada que permite almacenar programas y datos.

Bajo Consumo de Energía: Diseñado para ser eficiente en términos de consumo de energía, lo que lo hace adecuado para dispositivos alimentados por batería.

El ESP32 es ampliamente utilizado en proyectos de desarrollo de hardware y software, especialmente en el campo del IoT, donde la conectividad inalámbrica es esencial. Puede programarse utilizando el entorno de desarrollo de Arduino, y existen otras herramientas y frameworks disponibles para facilitar el desarrollo de aplicaciones para el ESP32.

El ESP32 tiene una variedad de pines que cumplen diferentes funciones, desde pines de alimentación hasta pines GPIO (Entrada/Salida de Propósito General), pines de comunicación, y más. Explicación general de las categorías principales de pines en el ESP32:

Pines de Alimentación:

VCC (3.3V): Este pin se utiliza para alimentar el ESP32 con una tensión de 3.3 voltios.

GND (Tierra): Conecta este pin a tierra para completar el circuito eléctrico.

Pines de Entrada/Salida de Propósito General (GPIO):

El ESP32 tiene varios pines GPIO que se pueden configurar como entradas o salidas según las necesidades del usuario. Estos pines se utilizan para conectar sensores, actuadores y otros dispositivos.

Los pines GPIO se numeran desde 0 hasta 39. Algunos de estos pines pueden tener funciones adicionales, como pines de entrada analógica (ADC), pines de salida de pulso de ancho modulado (PWM), etc.

Pines de Comunicación:

TX0 y RX0: Pines de transmisión y recepción serial para la comunicación UART.

SCL y SDA: Pines para la comunicación I2C.

MOSI, MISO, SCK, y SS/CS: Pines para la comunicación SPI.

SD2 y SD3: Pines adicionales de datos para la comunicación SPI en modo quad.

Pines de Control de Periféricos:

Pines de control de PWM: Utilizados para generar señales de modulación por ancho de pulso.

Pines de control de interrupciones: Permiten gestionar interrupciones en el microcontrolador.

Pines de Alimentación Especiales:

EN (Enable): Pin de habilitación que se utiliza para encender y apagar el ESP32.

34-39 (Entradas Analógicas): Estos pines pueden usarse como entradas analógicas para medir tensiones analógicas mediante el convertidor analógico a digital (ADC).

Pines de Conexión WIFI - Bluetooth:

inalámbricos: Pines conectados al chip de la comunicación Wi-Fi y Bluetooth.

El ESP32 es un microcontrolador con un diseño interno complejo que incluye varios componentes y subsistemas. Descripción general del funcionamiento interno del ESP32:

Procesador:

El ESP32 utiliza un procesador de doble núcleo Xtensa LX6. Los dos núcleos permiten la multitarea y pueden ejecutar instrucciones de manera independiente.

Memoria:

Memoria Flash: El ESP32 tiene incorporada una memoria flash que se utiliza para almacenar el programa (código) y datos.

Memoria RAM: Se utiliza para almacenar datos temporales y variables durante la ejecución del programa.

Unidad de Procesamiento de Señales Analógicas y Digitales (APB/AD):

Incluye convertidores analógico a digital (ADC) para medir señales analógicas.

Permite la entrada de señales analógicas a través de pines específicos.

Unidad de Procesamiento de Entrada/Salida (GPIO):

Permite la configuración y control de los pines de entrada/salida de propósito general (GPIO).

Unidad de Control de Interrupciones:

Gestiona las interrupciones del sistema y permite que el microcontrolador responda a eventos específicos de manera eficiente.

Unidad de Comunicación Inalámbrica:

Wi-Fi: Incluye un módulo Wi-Fi IEEE 802.11 b/g/n para la conectividad inalámbrica.

Bluetooth: Incorpora un módulo Bluetooth 4.2 BLE (Bluetooth de baja energía).

Unidad de Comunicación Serial:

Permite la comunicación serie a través de UART, SPI, y I2C, lo que facilita la conexión con otros dispositivos y periféricos.

Unidad de Generación de Pulsos de Ancho Modulado (PWM):

Facilita la generación de señales PWM, lo que es útil para el control de motores, luces LED y otros dispositivos.

Unidad de Temporización y Contadores:

Incluye temporizadores y contadores que permiten la medición del tiempo y el control de eventos temporales.

Gestión de Energía:

El ESP32 está diseñado para ser eficiente en términos de consumo de energía, y la gestión de energía incluye mecanismos para entrar en modos de bajo consumo cuando no se requiere un rendimiento completo.

Unidad de Seguridad:

Incluye funciones de seguridad, como aceleradores de criptografía y funciones de almacenamiento seguro.

Estas son solo algunas de las características generales del ESP32. La complejidad del diseño y las capacidades del ESP32 hacen que sea adecuado para una amplia gama de aplicaciones

STEPDOWN

Un regulador Step-Down, también conocido como convertidor de disminución de voltaje o regulador de voltaje buck, es un tipo de fuente de alimentación o convertidor de energía que reduce (o "baja") un voltaje de entrada a un voltaje de salida más bajo. Este tipo de regulador es comúnmente utilizado en aplicaciones donde se requiere una fuente de alimentación con un voltaje menor que el suministrado por la fuente de energía principal, como en la mayoría de los dispositivos electrónicos alimentados por batería.

Aquí hay una explicación básica del funcionamiento de un regulador Step-Down:

Circuito de Control:

El corazón del regulador Step-Down es un circuito de control que gestiona la transferencia de energía desde la entrada hasta la salida. Este circuito controla los interruptores y otros componentes clave.

Interruptor (Switch):

El regulador Step-Down utiliza un interruptor (generalmente un transistor) que se enciende y apaga a una frecuencia relativamente alta. Este

interruptor está conectado en serie con la fuente de alimentación y la carga.

Inductor:

Un inductor se conecta en serie con el interruptor. Cuando el interruptor está cerrado, el inductor acumula energía almacenando la corriente. Cuando el interruptor está abierto, la corriente fluye a través del inductor y se libera energía para la carga.

Diodo (Opcional):

En algunos reguladores Step-Down, se utiliza un diodo para rectificar la corriente a través del inductor cuando el interruptor está abierto. En otros diseños, se utiliza una topología síncrona sin el diodo.

Condensador de Salida:

Un condensador se coloca en la salida del regulador para suavizar la señal de salida y proporcionar energía adicional a la carga durante los periodos en que el interruptor está abierto.

El proceso de regulación se basa en el ciclo de encendido y apagado controlado del interruptor. Al ajustar la duración de los ciclos de encendido y apagado, el regulador puede mantener un voltaje de salida constante incluso si el voltaje de entrada varía o si la carga conectada cambia.

Las ventajas de los reguladores Step-Down incluyen su eficiencia en la conversión de energía (en comparación con reguladores lineales, especialmente para grandes diferencias de voltaje) y su capacidad para operar con baterías durante más tiempo, ya que pueden reducir el voltaje a medida que la batería se descarga.

HC-SR04

Un sensor de ultrasonido es un dispositivo que utiliza ondas ultrasónicas para medir la distancia entre el sensor y un objeto. Estos sensores son comúnmente utilizados en aplicaciones donde se requiere la detección de objetos o la medición de distancias sin contacto. Explicación básica del funcionamiento de un sensor de ultrasonido:

Generación de Ondas Ultrasónicas:

El sensor de ultrasonido incluye un transductor piezoeléctrico que puede generar ondas ultrasónicas. Este transductor convierte señales eléctricas en vibraciones mecánicas, creando ondas sonoras ultrasónicas.

Emisión de Ondas Ultrasónicas:

Cuando se activa, el transductor emite pulsos de ondas ultrasónicas en una dirección específica. Estas ondas se propagan en el aire en forma de un cono estrecho.

Reflexión en un Objeto:

Las ondas ultrasónicas viajan hasta encontrar un objeto en su camino. Cuando estas ondas encuentran un objeto, se reflejan de vuelta hacia el sensor.

Recepción de Ondas Reflejadas:

El mismo transductor que emitió las ondas ahora actúa como receptor. Captura las ondas ultrasónicas reflejadas por el objeto.

Cálculo de la Distancia:

La distancia entre el sensor y el objeto se calcula midiendo el tiempo que tarda una onda ultrasónica en viajar desde el sensor hasta el objeto y regresar. Dado que la velocidad del sonido en el aire es constante, la distancia se puede calcular utilizando la fórmula: Distancia = (Velocidad del Sonido * Tiempo de Vuelo) / 2.

Salida de Datos:

La distancia calculada se convierte en una señal eléctrica que se puede leer o interpretar por un microcontrolador u otro dispositivo de procesamiento. Los sensores de ultrasonido a menudo proporcionan la distancia medida como salida.

Algunos puntos a tener en cuenta sobre los sensores de ultrasonido:

La precisión de la medición depende de factores como la calidad del sensor, la frecuencia ultrasónica utilizada y las condiciones ambientales, como la temperatura y la humedad.

Los sensores ultrasónicos son efectivos para medir distancias en el rango de centímetros a varios metros.

Estos sensores son especialmente útiles en entornos donde otros métodos de detección, como sensores infrarrojos, pueden no ser apropiados debido a condiciones ambientales o características específicas del objeto a detectar.

En resumen, los sensores de ultrasonido aprovechan el principio de eco de las ondas sonoras ultrasónicas para medir distancias sin contacto de manera precisa y eficiente.

DRIVERS BTS7960

El BTS7960 es un controlador de motor de puente H (H-Bridge) diseñado para controlar la velocidad y la dirección de motores de corriente continua (DC). Este tipo de controladores son comunes en aplicaciones de robótica, vehículos eléctricos y otros sistemas que requieren un control preciso de la velocidad y dirección del motor. A continuación, se explica cómo funciona el BTS7960:

Puente H (H-Bridge):

El BTS7960 utiliza una configuración de puente H, que es una disposición de cuatro interruptores (transistores) que permite controlar la dirección del flujo de corriente a través del motor.

Control de Velocidad y Dirección:

El puente H permite controlar tanto la velocidad como la dirección del motor. La inversión de la dirección del motor se logra invirtiendo la polaridad de la corriente que fluye a través del motor.

Control PWM (Modulación de Ancho de Pulso):

Para controlar la velocidad del motor, el BTS7960 utiliza la técnica de modulación de ancho de pulso (PWM). El controlador recibe una señal PWM como entrada, y la anchura del pulso determina la velocidad del motor. Variando el ciclo de trabajo del PWM, se puede variar la velocidad del motor.

Protección de Sobrecorriente:

El BTS7960 generalmente incluye características de protección, como la detección de sobrecorriente. Si la corriente que fluye a través del motor supera un umbral predeterminado, el controlador puede apagar automáticamente la corriente para evitar daños.

Interfaz de Control:

El BTS7960 se controla típicamente mediante una interfaz de control que acepta señales PWM para establecer la velocidad y la dirección del motor. Estas señales pueden provenir de un microcontrolador, un sistema embebido u otros dispositivos de control.

Conexiones de Potencia:

El BTS7960 tiene conexiones para la fuente de alimentación del motor (Vcc) y la conexión a tierra (GND). Además, puede tener conexiones para la alimentación del propio controlador.

Indicadores y Pines de Estado:

Algunos modelos de BTS7960 pueden incluir pines de estado o indicadores LED que muestran la dirección del motor o el estado de la protección.

Disipador de Calor:

Debido a que el BTS7960 puede generar calor durante su funcionamiento, muchos modelos incluyen un disipador de calor para mantener una temperatura de funcionamiento segura.

El BTS7960 es un controlador de motor de puente H que proporciona control bidireccional de corriente continua (DC) y es comúnmente utilizado en aplicaciones de control de motores, como vehículos eléctricos y robótica. Los pines del BTS7960 pueden variar ligeramente según el fabricante y el modelo específicos, pero a continuación, se presenta una descripción general de los pines típicos de este controlador:

PWMA y PWM:

PWMA: Señal de entrada para controlar la velocidad del motor conectado al canal A. Se utiliza una señal de modulación de ancho de pulso (PWM) para ajustar la velocidad del motor.

PWMB: Similar a PWMA, pero para el canal B.

AIN1, AIN2, BIN1, BIN2:

Estos pines controlan la dirección del giro del motor. Al cambiar el estado de estos pines, se invierte la dirección del flujo de corriente a través del motor.

AIN1 y AIN2: Controlan el canal A.

BIN1 y BIN2: Controlan el canal B.

GND (Tierra):

Conexión a tierra del controlador.

Vcc (Fuente de Alimentación):

Este pin se utiliza para alimentar el controlador. Proporciona la fuente de alimentación para el circuito de control interno del BTS7960.

EN (Habilitación):

Al aplicar un voltaje lógico alto (generalmente 5V), se habilita el controlador. En algunos casos, este pin se puede conectar a tierra para deshabilitar el controlador y evitar que los motores se muevan.

FLT (Salida de Falla):

Proporciona una señal que indica la presencia de una falla, como la detección de sobrecorriente. Cuando se activa, puede indicar problemas

en el sistema, y es importante monitorear esta señal para garantizar un funcionamiento seguro.

Vcc Motor:

Este pin suministra la fuente de alimentación para el motor. Se conecta a la fuente de alimentación del motor.

GND Motor:

Conexión a tierra para el motor.

Diagrama de Conexiones del Circuito General

Diagrama de conexiones:

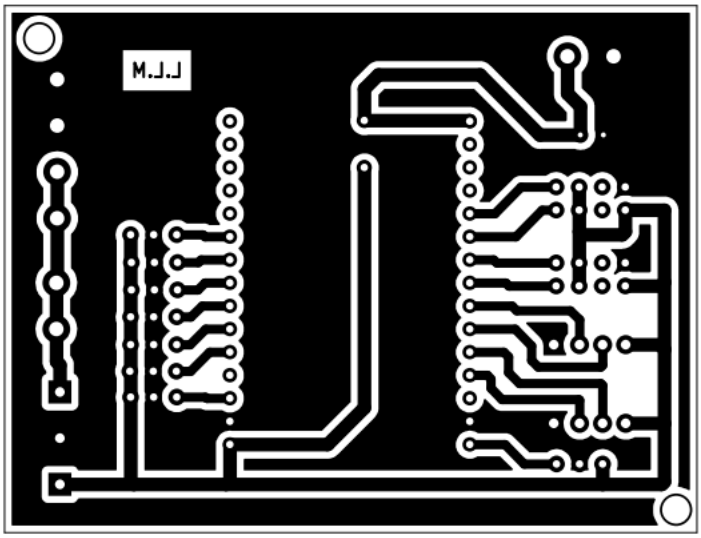
A continuación se muestra una imagen acerca del Esquemático del robot en el cual podemos encontrar todas las conexiones realizadas en la placa y en donde podemos ver más claramente a donde está conectada cada cosa.

Además también podemos apreciar los componentes con los que cuenta el robot los cuales son: ESP32 - borneras de alimentación y bornes del interruptor - pineras para los drivers - sensores de ultrasonido - entrada del pulsador.

PCB:

Aquí tenemos una imagen del PCB en la cual como podemos apreciar están las pistas de cobre que conectan la alimentación de todo el circuito y también se pueden observar las conexiones de los drivers y sensores y sus correspondientes alimentaciones.
También tiene unos agregados los cuales son conexiones para poner sensores de piso por si en algún futuro se decidiera agregarlos.
Cuenta con una salida para un pulsador y con salidas para los motores tanto de VCC como de GND.

Se puede ver cómo está conectada la alimentación de los drivers o la parte lógica de estos la cual dispone de los cuatro pines EN conectados entre sí a una pista de +5v que, a su vez conecta directamente a la alimentación general de +5v.



Diseño 3D del Robot

Para realizar este diseño 3D se utiliza el programa Fusion 360 el cual nos permite a base de dibujos y bocetos desde diferentes vistas y caras, poder

hacer con medidas acotadas la estructura del robot.
También nos permite hacer más de un objeto permitiéndonos hacer la tapa y poder ponerle un logo como imagen. Otras cosas que nos permite hacer esta aplicación son las curvaturas, diagonales y verticales además de que nos deja hacer la rosca para los tornillos de la métrica que queramos.

Otros aspectos y ventajas de esta aplicación son su facilidad para aprender a utilizarla y su sencillez, así como el hecho de que al ser una app relativamente novedosa es bastante usada en el ámbito de la mecanización de piezas. A diferencia de Solidworks esta app es bastante fácil de aprender a usar ya que cuenta con comandos más sencillos y es muy fácil de descargar. También cuenta con la ventaja de poder transformar los archivos que generemos en archivos stl para de esta forma poder generar más fácilmente los archivos de impresión para las impresoras 3D.

CÓDIGO

Aquí se definen los pines a utilizar y para cada componente por ejemplo:

#define MI2 19 significa que el segundo pin del Motor Izquierdo va conectado al pin 19 del esp 32

```
#define TRIG  17

#define ECHO  16

#define TRIG2 4

#define ECHO2 2

#define MD1 5

#define MD2 18

#define MI2 19

#define MI1 21

#define BUTTON_PIN 23
```

```
bool codeExecuted = false;  //estas variables se utilizan para poder hacer que es sumo empiece a sentir despues de que el pulsador es presionado//

unsigned long buttonPressTime = 0;


void setup() { //aquí en el void setup se inicializan los pines  anteriormente definidos//
//como entradas o salidas//

   Serial.begin(115200);

  pinMode(LED_1, OUTPUT);

  pinMode(TRIG, OUTPUT);

  pinMode(ECHO, INPUT);

  digitalWrite(TRIG, LOW);  //low significa apagado así como high encendido, asi como 1 y 0//

  pinMode(LED_2, OUTPUT);

  pinMode(LED_3, OUTPUT);

  pinMode(LED_4, OUTPUT);

  pinMode(TRIG2, OUTPUT);

  pinMode(ECHO2, INPUT);

  digitalWrite(TRIG2, LOW);

  pinMode(BUTTON_PIN, INPUT_PULLUP); //define la resistencia interna del pulsador como pull up//

}


void loop() {  //el void loop es la parte del código que se ejecuta permanentemente//

  if (!codeExecuted) {  //esta parte lo que hace es poner al pin en 0 es decir no a sí presionado para luego con la variable millis ir contando el tiempo //

    if (digitalRead(BUTTON_PIN) == LOW) {

      buttonPressTime = millis();

      codeExecuted = true;

    }

  }
```

```
  if (codeExecuted && (millis() - buttonPressTime >= 5000)) { //aqui lo que hacemos es que si el botón ha sido presionado y pasaron los cinco segundo que se ejecute el código "executecode"//
    executeCode();
  }

void executeCode() {
//definimos y calculamos la distancia y la pasamos a cm para el sensor de ultrasonido en base a una cuenta matemática//
  long t;
  long d;
  digitalWrite(TRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);

  t = pulseIn(ECHO, HIGH);
  d = t / 59;

  Serial.print("distancia: ");
  Serial.print(d);
  Serial.println();
  delay(1000);

  long t2;
  long d2;
  digitalWrite(TRIG2, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG2, LOW);
```

```
  t2 = pulseIn(ECHO2, HIGH);
  d2 = t2 / 59;

  Serial.print("distancia2: ");
  Serial.print(d2);
  Serial.println();
  delay(1000);

  if (d >= 10 && d <= 200 && d2 >= 30 && d2 <= 150) {
//si las distancias respetan la siguiente condición se ejecutará el siguiente bloque//
    digitalWrite(MD1, HIGH);
    digitalWrite(MD2, LOW);
    digitalWrite(MI1, HIGH);
    digitalWrite(MI2, LOW);
    Serial.println(" ataque ");
  } else if (d > 200 && d < 300 && d2 >= 250 && d2 <= 350) {
//si esa condición no se cumple pero si se cumple la siguiente entonces ocurrirá esto//
    digitalWrite(MD1, LOW);
    digitalWrite(MD2, HIGH);
    digitalWrite(MI1, HIGH);
    digitalWrite(MI2, LOW);
    Serial.println(" busqueda ");
  } else {
//y finalmente si nada de eso se cumple ocurrirá lo siguiente//
    digitalWrite(MD1, HIGH);
    digitalWrite(MD2, LOW);
    digitalWrite(MI1, LOW);
```

```
   digitalWrite(MI2, HIGH);
   Serial.println(" busqueda ");
   delay(3000);
  }
}
```

Estos son todos los detalles del desarrollo del proyecto que nosotros aplicamos para lograr realizarlo.

Espero que haya sido de su agrado.

Un saludo cordial del grupo: ***THE SPARTANS*** :)