

Design and Implementation of an Omnidirectional Robot with a 5-Degree-of-Freedom Robotic with Bluetooth Control

Marcelo Jaime Quispe Ccachuco, Dr.¹, César Pio Castillo Cáceres, Dr.², Edson Jean-lu Rodriguez Garcia, Br.³, y Cristian Alvaro Villca Balcon Br.⁴.

^{1,4}Universidad Católica de Santa María, Perú, mquispec@ucsm.edu.pe, edson.rodriguez@ucsm.edu.pe

^{2,3}Universidad Católica de Santa María, Perú, ccastillo555@yahoo.es, 74126657@ucsm.edu.pe

Abstract—This project involves the successful implementation of a cutting-edge robotic system, optimally integrating the capabilities of an omnidirectional robot with a horizontal mobile base, with mecanum wheels for movement in all directions, with a 5 DOF robotic arm attached to it to increase the workspace. The incorporation of Bluetooth communication implies the creation of a manipulation interface (APK) that amplifies its functionalities, allowing remote control and interaction in real time. Furthermore, the inclusion of a sophisticated movement memory system gives the robot the ability to store and remember intricate movement sequences, through a loop created in the idle code (Arduino), contributing to its adaptability and efficiency in various applications, such as robots. pick and place that do it cyclically, which will be programmed and executed on the arduino Mega, along with its design that includes the 4 nema stepper motors for the mecanum wheels and 5 servomotors for the manipulation part of the upper robot with its drives and corresponding electronic components on its PCB board. The solid parts were manufactured in inventor to corroborate the maximum scope and final visualization of the assembly, in order to carry out a simulation of the most critical component of the robot with its respective plans, considering 3D printing and laser cutting for the manufacture of these components. and its corresponding assembly. Mecanum wheels offer a speed of 12 cm per second according to the programming in the Arduino Mega, allowing smooth and precise movements in multiple directions. The implementation and construction considered the structural resistance and a maximum load capacity of 200 grams. Performance optimization was achieved through trajectory planning algorithms, and thorough testing and adjustments ensured safe and reliable operation. Mecanum wheels are widely used in the robotics industry, especially in applications that require maneuverability and precise movements in tight spaces. The Mecanum wheel platform allows the

robot to move in any direction without changing its orientation, thanks to the special arrangement of rollers on the wheels that enables lateral and diagonal displacement.

Keywords—Movil Robot, Omnidirectional Movement, DOF, Bluetooth Communication, Mecanum Wheels

Digital Object Identifier: (only for full papers, inserted by LACCEI).
ISSN, ISBN: (to be inserted by LACCEI).
DO NOT REMOVE

Diseño e Implementación de un Robot Omnidireccional con Brazo Robótico de 5 Grados de Libertad Controlado por Bluetooth

Marcelo Jaime Quispe Ccachuco, Dr.¹, César Pío Castillo Cáceres, Dr.², Edson Jean-lu Rodriguez Garcia, Br.³, y Cristian Alvaro Villca Balcon Br.⁴.

^{1,4}Universidad Católica de Santa María, Perú, mquispec@ucsm.edu.pe, edson.rodriguez@ucsm.edu.pe

^{2,3}Universidad Católica de Santa María, Perú, ccastillo555@yahoo.es, 74126657@ucsm.edu.pe

Resumen- Este proyecto involucra la implementación exitosa de un sistema robótico de vanguardia, integrando de manera óptima las capacidades de un robot omnidireccional con base móvil horizontal, con ruedas mecanum para su desplazamiento en toda dirección, con un brazo robótico de 5 GDL acoplado sobre esta para incrementar el espacio de trabajo. La incorporación de la comunicación Bluetooth implica, la creación de una interfaz de manipulación (APK) que amplifica sus funcionalidades, permitiendo el control remoto e interacción en tiempo real. Además, la inclusión de un sofisticado sistema de memoria de movimientos dota al robot la capacidad de almacenar y recordar secuencias de movimientos intrincadas, mediante un bucle creado en el código de la placa Arduino, contribuyendo a su adaptabilidad y eficiencia en diversas aplicaciones, como los robots pick and place que lo hacen de manera cíclica, las cuales serán programados y ejecutados en el arduino Mega, junto con su diseño que incluye los 4 motores paso a paso nema para las ruedas mecanum y 5 servomotores para la parte de la manipulación del robot superior con sus drives y componentes electrónicos correspondientes en su placa pcb. Las piezas sólidas fueron manufacturados en inventor para corroborar el alcance máximo y la visualización final del ensamble, con el fin de realizar una simulación del componente más crítico del robot con sus planos respectivos, considerando la impresión 3d y cortadora laser para la manufactura de estos componentes y su ensamble correspondiente. Las ruedas Mecanum ofrecen una velocidad de 12 cm por segundo según la programación en el Arduino Mega, permitiendo movimientos suaves y precisos en múltiples direcciones. La implementación y construcción consideró la resistencia estructural y una capacidad de carga máxima de 200 gramos. La optimización del rendimiento se logró mediante algoritmos de planificación de trayectoria, y pruebas y ajustes exhaustivos garantizaron una operación segura y confiable. Las ruedas Mecanum se utilizan ampliamente en la industria de la robótica, especialmente en aplicaciones que requieren maniobrabilidad y movimientos precisos en espacios reducidos. La plataforma de ruedas Mecanum permite que el robot se mueva en cualquier dirección sin cambiar su orientación, gracias a la disposición especial de rodillos en las ruedas que permite el desplazamiento lateral y diagonal.

Palabras clave – Robot Móvil, Movimiento Omnidireccional, GDL, Comunicación Bluetooth, Ruedas Mecanum.

I. INTRODUCCIÓN

La fabricación inteligente y la logística han impulsado avances significativos en la tecnología de robots móviles en los últimos años. La demanda de robots móviles autónomos en fábricas inteligentes ha crecido sustancialmente, lo que requiere robots

que sean flexibles, adaptables y seguros para navegar en entornos de trabajo complejos. [1-5]

Para abordar los requisitos de tales entornos, la elección de ruedas para robots móviles es de suma importancia. Las ruedas omnidireccionales, equipadas con rodillos en sus bordes, resuelven las limitaciones de las ruedas tradicionales [6].

Estos rodillos permiten el movimiento en una dirección mientras permiten un deslizamiento libre en la dirección perpendicular. Al utilizar tres o más de estas ruedas, los robots obtienen un control total sobre su movimiento, permitiendo un desplazamiento y rotación libres en un plano sin necesidad de girar las ruedas. Este diseño simplifica la estructura general del robot y mejora su maniobrabilidad.[7]

A su vez las ruedas mecanum, un tipo de rueda omnidireccional, mejoran aún más la movilidad al incorporar pequeños rodillos inclinados a un ángulo de 45 grados respecto al eje principal de rotación. Este diseño único capacita al robot para realizar movimientos complejos en diversas direcciones dentro de espacios reducidos. La utilización de ruedas Mecanum en montacargas, por ejemplo, mejora significativamente la eficiencia operativa, aumentando así la productividad para las empresas.[7]

Combinar robots omnidireccionales con ruedas mecanum y brazos robóticos con 5 grados de libertad (gdl) aborda las limitaciones de movilidad de los brazos robóticos tradicionales. Integrar el movimiento de las ruedas omnidireccionales con el brazo del robot proporciona una solución a esta limitación. Sin embargo, lograr una alta precisión en dicha integración puede plantear desafíos [8].

Cabe señalar que el robot utilizara el microcontrolador ATMEGA de ARDUINO.

Asimismo, al momento de utilizar la comunicación Bluetooth para conectar Arduino UNO y Android. Arduino puede ser conectado al módulo Bluetooth mediante el protocolo UART. Según los comandos recibidos desde Android, se controla el movimiento del robot. La salida constante de un sistema mecánico, junto con la repetibilidad, son inmejorables. Estos robots, son reprogramables y pueden adaptarse para diversas aplicaciones, al ser transferibles.[9]

Sin embargo para establecer dicha comunicación se utiliza la app inventor.

Una de las posibilidades de enriquecer este tipo de dispositivo con sus propios programas, es MIT App Inventor, una aplicación web desarrollada por Google y actualmente gestionada por el Instituto Tecnológico de Massachusetts (MIT). App Inventor se basa en la programación de arrastrar y soltar en un entorno gráfico, tanto en el diseño de diálogos como en la codificación de aplicaciones. Cada vez más investigadores promueven su trabajo utilizando los conceptos de internet móvil.[11]

En este proyecto se realiza la suma de todos los incisos anteriores, mostraremos paso a paso como se realizó el diseño, implementación, incluyendo impresión, armado, conexión, programación y uso de distintas plataformas para obtener la manipulación de nuestro robot omnidireccional con un brazo robótico de 5 GDL.

II. MARCO TEÓRICO

Como se observa en la figura 1 una rueda omnidireccional se considera como un disco rodante de radio R centrado en el punto C en un plano horizontal, el plano del disco es siempre vertical. Sea τ el vector unitario a lo largo de los ejes de los rodillos y el ángulo entre el plano de la rueda y el vector perpendicular al eje del rodillo[12]

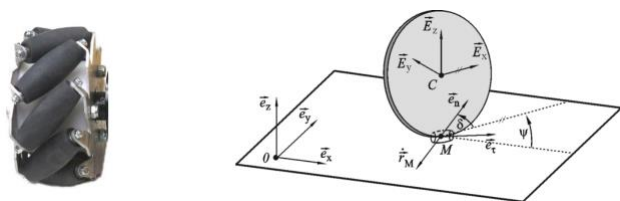


Fig. 1 Cinematica y cinética de una rueda mecanum

El principio de funcionamiento y los movimientos se muestran en las Figuras 2-4

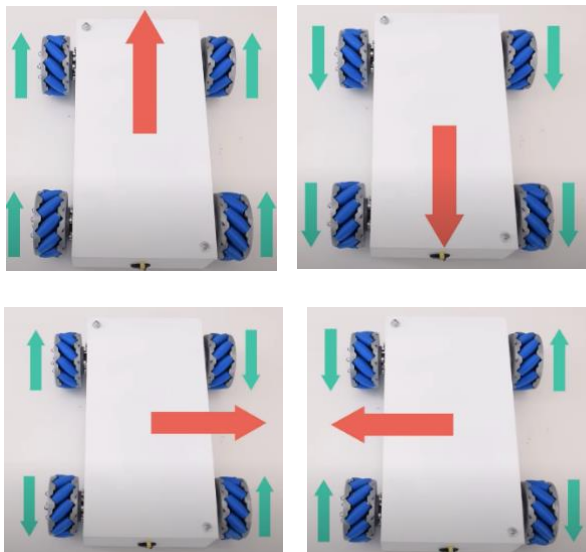


Fig. 2 Movimiento adelante y hacia atrás, movimiento izquierda y derecha

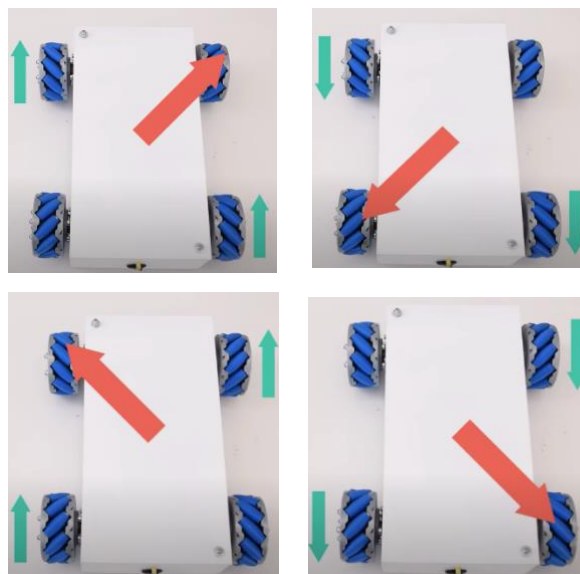


Fig. 3 Movimientos diagonales superiores e inferiores

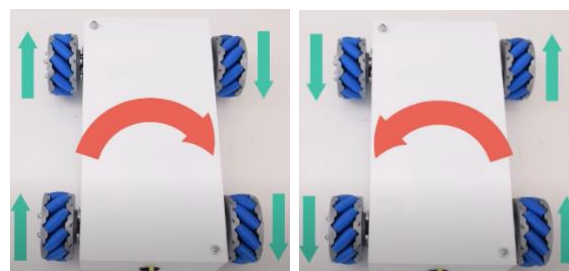


Fig. 4 Movimientos rotacionales

Por otro lado, la interfaz de usuario de MIT App Inventor incluye dos editores principales: el editor de diseño y el editor de bloques. El editor de diseño(Figura 5), es una interfaz de arrastrar y soltar para organizar los elementos de la interfaz de usuario (UI) de la aplicación. El editor de bloques (Figura 6) es un entorno en el cual los creadores de aplicaciones pueden diseñar visualmente la lógica de sus aplicaciones utilizando bloques codificados por colores que se ensamblan como piezas de rompecabezas para describir el programa. Para facilitar el desarrollo y las pruebas, App Inventor proporciona una aplicación móvil llamada App Inventor Companion (o simplemente "el Companion") que los desarrolladores pueden utilizar para probar y ajustar el comportamiento de sus aplicaciones en tiempo real. De esta manera, cualquiera puede construir rápidamente una aplicación móvil y comenzar de inmediato a iterar y probar.[13]

Como observamos en la figura 6, El call bluetooth (cuadro morado) realiza toda la sucesión que le antecede, en este caso el cuadro verde(condicionales) y rosado(set o acciones), a través de la orden numero 2 que es un contacto físico o pulsador que ya se grabó en la interfase MIT APP graficada en el la imagen 5 , para ser mas exacto giro a la derecha.

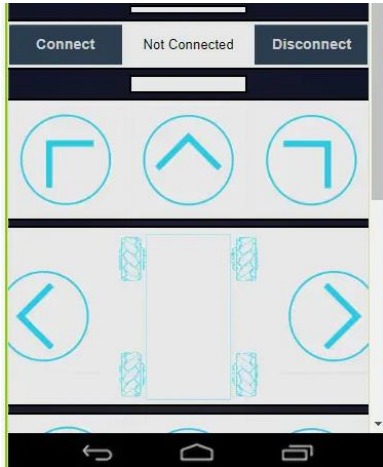


Fig. 5 Modelo editor de diseño



Fig. 6 Modelo editor de bloques

III. MATERIALES Y MÉTODOS

A. Brazo robótico

Como se muestra a detalle en la figura 7, para el diseño del sistema se buscó tener el mayor rendimiento con el menor costo posible. Para la estructura de las articulaciones del robot se realizó impresión 3D con filamento PLA, estas fueron ensambladas con pernos del servomotor.

El modelo mecánico fue diseñado en el software autodesk inventor, el modelo consta de 5 articulaciones, es una herramienta llamada pinza, estas trabajan en equipo para garantizar un movimiento y recogida de piezas, el brazo robótico consta de 6 grupos, los tres primeros grupos, la cintura, el hombro y el codo, están constituidos por servos MG996R, y para los siguientes 2 grupos, el movimiento de la muñeca y el paso de la muñeca, estarán compuestas por servomotores SG90, para finalizar el último grupo llamado pinza esta compuesta con el servomotor SG90.

Los primeros 3 grupos transmisión constituidos por servomotores MG996R de 11 kg/cm, cuando el brazo robótico esta extendido en su longitud máxima alcanza una distancia de 26.4 cm desde el primer grupo ubicado en la base, hasta el extremo de la pinza, hallando una relación de torque necesario

para el brazo robótico de 26.4 cm de longitud y un actuador de 11kg/cm tendríamos la libertad de mover 416.7 gramos en el extremo de la pinza, pero debemos recordar el peso del diseño 3D el peso de los actuadores, pernos y tuercas, estos componentes mencionados llevan un peso en conjunto de 368 gramos, entonces en el extremo del robot se podrá levantar como máximo 48.6 gramos.

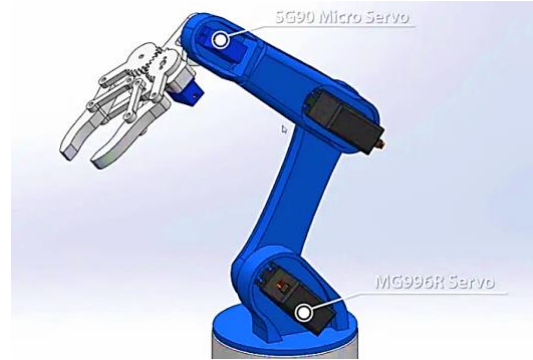


Fig. 7 brazo robótico de 5 grados de libertad y una herramienta

B. Base del robot con ruedas mecanum

La base del brazo robótico está conformado por una estructura de mdf de 4 milímetros de espesor, al diseñar la estructura de mdf se debe tener en cuenta el espesor del material y la posición de cada motor paso a paso tomando en cuenta sus medidas, por tal motivo quede una base sólida, los 4 motores nema 17 deben estar unidos a esta plataforma, de la misma manera cada motor debe contar con ruedas mecanum previamente diseñadas y armadas.

Para un movimiento libre de forma horizontal debemos considerar un tipo de ruedas que tienen elementos en forma de barril, distribuidos a su alrededor, con 45° de inclinación con respecto al eje de rotación de la rueda como muestra la figura 8, este diseño desarrolla que la fuerza ejercida sobre la rueda sea en dirección diagonal cuando esta es movida en sentido horario o antihorario, al revolucionar en un patrón predefinido utilizamos estas fuerzas diagonales, por tal motivo, el robot puede moverse libremente de manera horizontal.

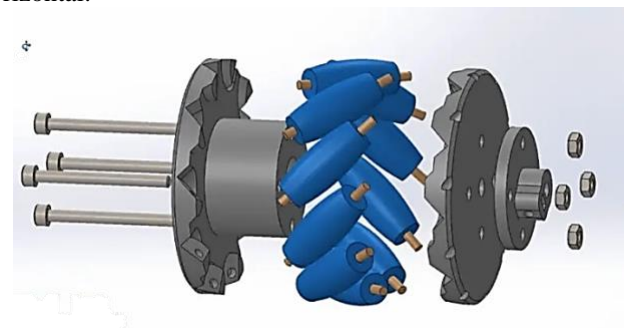


Fig. 8 plano de explosión rueda mecanum ensamblada

Se debe tener en cuenta que se necesitan dos tipos de ruedas mecanum, las denominadas ruedas mecanum par lado zurdo y diestro, la diferencia entre los mencionados es la orientación de los rodillos, estos deben ser instalados en ubicaciones específicas de la plataforma de robot, el eje de rotación del rodillo superior de cada rueda debe apuntar al centro de la plataforma del robot como muestra la figura 9 con esta configuración de posiciones de las ruedas mecanum, si las cuatro ruedas giran en sentido horario, el movimiento resultante del robot en conjunto será hacia adelante, si el sentido de giro de las ruedas es antihorario, el robot en conjunto se moverá hacia atrás

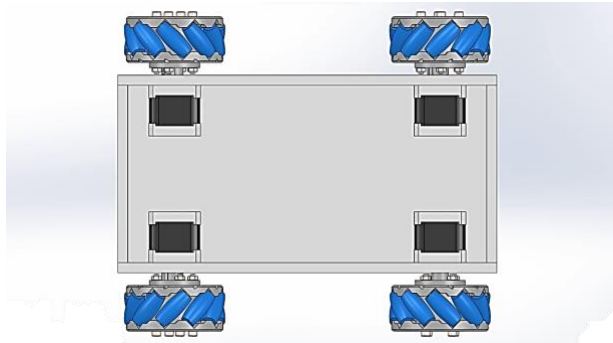


Fig. 9 Base o plataforma del robot omnidireccional

Para moverse hacia la derecha, las ruedas derechas deben girar dentro del robot, mientras que las ruedas izquierdas deben girar fuera del robot. La fuerza resultante debido a los rodillos colocados en diagonal hará que el robot se mueva hacia la derecha. Lo mismo, pero al revés sucede cuando se mueve hacia la izquierda. Con estas ruedas también podemos conseguir un movimiento en dirección diagonal girando sólo dos ruedas.

Al obtener las piezas de las ruedas mecanum impresas en 3D, optamos por realizar los ejes de los rodillos, para este

proceso se utilizó alambre de acero con 3.31mm de diámetro y con 4 cm de largo cada una con relieve a un extremo para su sujeción, al otro extremo será sellado con pegamento industrial, después del ensamble de los ejes con las piezas 3D, estas ruedas deben ser aseguradas con 4 pernos y tuercas 4M, de 2 pulgadas de longitud, por ambos lados

Los motores paso a paso deben ser ensamblados a la plataforma base, para esto serán asegurados con pernos 3M con una longitud de 12 milímetros, es importante que cada motor paso a paso debe estar ajustado de los 4 puntos de apoyo para una mejor estabilidad, seguidamente se debe fijar las ruedas al eje del motor, de esta manera tenemos la base de nuestro robot lista para ser unida con el brazo robótico, figura 10.

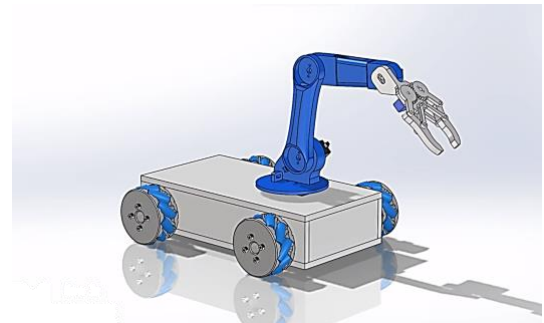


Fig. 10 Base o plataforma del robot omnidireccional unido con el brazo robótico

C. Diagrama del circuito

En la figura 11. podemos encontrar el diagrama del circuito del proyecto y como debe ser su conexión de cada uno de los componentes.

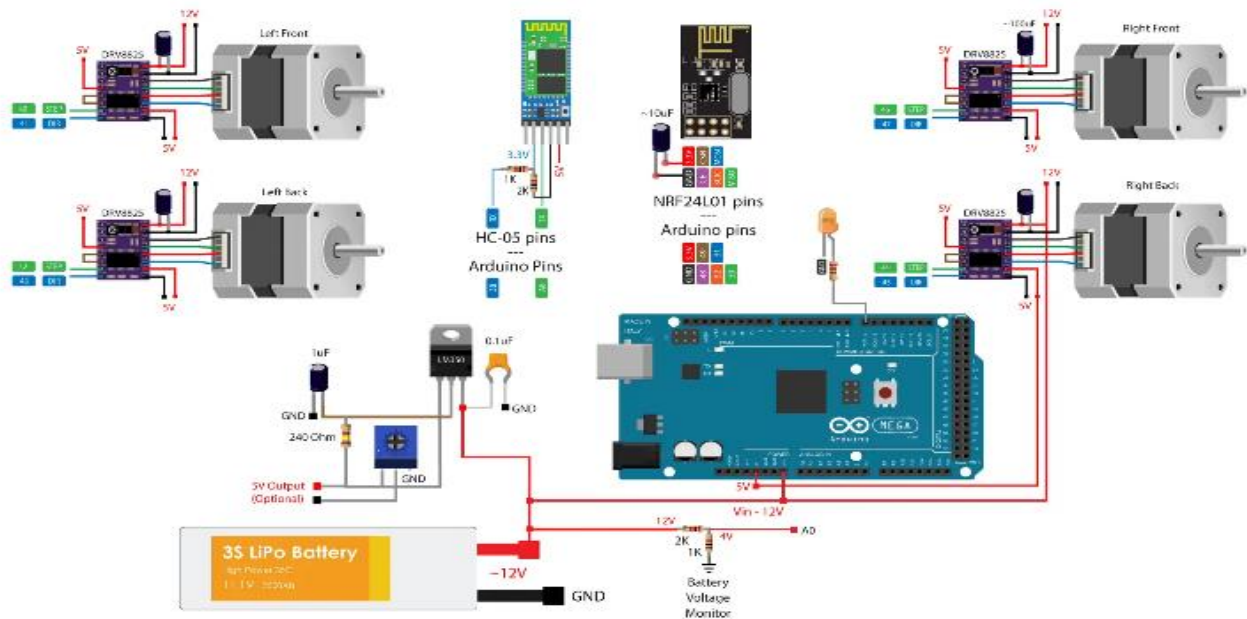


Fig. 11 Diagrama de conexión del circuito

De este modo controlamos los 4 motores paso a paso nema 17 usando 4 controladores de paso a paso, DR 8825, para alimentar los motores paso a paso, los servomotores y el resto del robot usaremos una fuente de 12 voltios de 3.2 amperios, para la comunicación bluetooth se utilizara el módulo bluetooth HC-05, este circuito también cuenta con un divisor de voltaje el cual tiene como finalidad monitorear el voltaje de la batería y las conexiones led para cuando el voltaje baje de los 11 voltios .

Por otro lado, el circuito también incluye un regulador de voltaje a 5 voltios para la alimentación de los servomotores y el ventilador que será agregado para disipar el calor que emite el regulador de voltaje LM 305

D. Diseño de la pcb

Para mantener los componentes electrónicos organizados y evitar el uso de cables innecesarios se diseñó una PCB personalizada utilizando el software de circuitos EasyEDA, El diseño de esta PCB nos permitirá la libertad y facilidad de realizar varias pruebas con los drivers, motores paso a paso y servomotores, también se incluyeron conexiones de 12 v, 5 v, y GND, en la figura 12 se puede observar a detalle el diseño de la placa con conexiones directas al Arduino mega y le lugar de cada componente como capacitores, drives, resistencias, pulsadores y entradas de señal DC

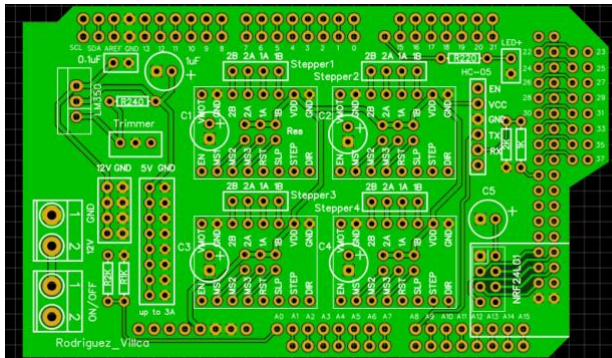


Fig. 12 Placa pcb

Por razones costo -tiempo se dio la solución de realizar una pcb con baquelita perforada (galleta), siguiendo el diagrama de la PCB, ensamblamos los componentes más pequeños como capacitores, resistencias y condensadores, también se utilizó espadines de 7 bloques para los drivers y para la conexión de los motores paso a paso después de estar soldada la placa PCB, se realiza la conexión de los servomotores, drives, motores y de la fuente de alimentación.

E. Código Arduino

Este código para Arduino mega ha sido desarrollado con el objetivo de controlar el robot de ruedas Mecanum con la

aplicación personalizada teniendo la capacidad de almacenar los movimientos y luego repetirlos todo esto dentro de un bucle dentro del código para ello explicaremos en los siguientes párrafos

Primero debemos definir los 6 servomotores, los 4 motores paspo a paso y la comunicación bluetooth al mismo tiempo que algunas librerías para el programa, para la velocidad máxima de los motores paso a paso debe ser modificado en la sección de configuraciones del código, definimos los pines a los que están conectados los servos la configuración del brazo robótico debe iniciar en su posición inicial

```
#include <SoftwareSerial.h>
#include <AccelStepper.h>
#include <Servo.h>
```

```
Servo servo01;
Servo servo02;
Servo servo03;
Servo servo04;
Servo servo05;
Servo servo06;
```

```
SoftwareSerial Bluetooth(A8, 38); // Arduino(RX, TX) - HC-05 Bluetooth (TX, RX)
```

```
// Definir los motores paso a paso y los a pines a utilizar
AccelStepper LeftBackWheel(1, 42, 43); // (Type:driver, STEP, DIR) - Stepper1
AccelStepper LeftFrontWheel(1, 40, 41); // Stepper2
AccelStepper RightBackWheel(1, 44, 45); // Stepper3
AccelStepper RightFrontWheel(1, 46, 47); // Stepper4
```

```
#define led 14
```

```
int wheelSpeed = 1500;
```

```
int lbw[50], lfw[50], rbw[50], rfw[50]; // matrices para almacenar posiciones/pasos
```

```
int servo1Pos, servo2Pos, servo3Pos, servo4Pos, servo5Pos, servo6Pos; // posición actual
```

```
int servo1PPos, servo2PPos, servo3PPos, servo4PPos, servo5PPos, servo6PPos; // posición anterior
```

```
int servo01SP[50], servo02SP[50], servo03SP[50], servo04SP[50], servo05SP[50], servo06SP[50]; // para almacenar posiciones /pasos
```

```
int speedDelay = 20;
```

```
int index = 0;
```

```
int dataIn;
```

```
int m = 0;
```

```
void setup() {
```

```
    // establecer valores iniciales para los motores paso a paso
    LeftFrontWheel.setMaxSpeed(3000);
```



```

LeftBackWheel.setMaxSpeed(3000);
RightFrontWheel.setMaxSpeed(3000);
RightBackWheel.setMaxSpeed(3000);
pinMode(led, OUTPUT);
servo01.attach(5);
servo02.attach(6);
servo03.attach(7);
servo04.attach(8);
servo05.attach(9);
servo06.attach(10);
Bluetooth.begin(38400); // Default baud rate of the Bluetooth
module
Bluetooth.setTimeout(5);
delay(20);
Serial.begin(38400);
// movimiento del brazo robotico a la posición inicial
servo1PPos = 90;
servo01.write(servo1PPos);
servo2PPos = 100;
servo02.write(servo2PPos);
servo3PPos = 120;
servo03.write(servo3PPos);
servo4PPos = 95;
servo04.write(servo4PPos);
servo5PPos = 60;
servo05.write(servo5PPos);
servo6PPos = 110;
servo06.write(servo6PPos);
}

```

También tenemos la sección del bucle, en el cual comprobamos si existen datos entrantes

```

// Comprobar datos entrantes
if (Bluetooth.available() > 0) {
  dataIn = Bluetooth.read(); // leer los datos
}

```

Estos datos son enviados desde la aplicación de Android, esta aplicación de Android será creada utilizando la aplicación en línea MIT App Inventor, figura 13.



Fig. 13 Desarrollo diseño brazo robotico mit app

Esta aplicación contará con botones simples para el usuario, la app envía números de byte cuando se hace clic en los botones, por tal, según del botón en el cual se realice el clic, le daremos una indicación al Arduino, como ejemplo, si recibimos como entrada el número 2, la plataforma de las ruedas mecanun avanzarán en sentido horario, utilizando la función personalizada moveForward

```

if (dataIn == 2) {
  m = 2;
}
//
if (m == 2) {
  moveForward();
}

```

Esta configuración realiza el movimiento de los cuatro motores paso a paso para que giren hacia adelante

```

void moveForward() {
  LeftFrontWheel.setSpeed(wheelSpeed);
  LeftBackWheel.setSpeed(wheelSpeed);
  RightFrontWheel.setSpeed(wheelSpeed);
  RightBackWheel.setSpeed(wheelSpeed);
}

```

Para el movimiento en cualquier otra dirección, solo debe cambiar el sentido de giro de las ruedas mecanun

Para controlar el brazo robótico utilizamos el mismo método. Nuevamente, tenemos botones en la aplicación y al mantener presionados los botones, las articulaciones del brazo del robot se mueven en la dirección particular, el código Arduino entra en el bucle while de ese número, y permanece ahí hasta que tocamos el botón, porque en ese momento enviamos el número 0 que significa que el robot no debe hacer nada.

```

// Move servo 1 in positive direction
while (m == 16) {
  if (Bluetooth.available() > 0) {
    m = Bluetooth.read();
  }
  servo01.write(servo1PPos);
  servo1PPos++;
  delay(speedDelay);
}
// Move servo 1 in negative direction
while (m == 17) {
  if (Bluetooth.available() > 0) {
    m = Bluetooth.read();
  }
  servo01.write(servo1PPos);
  servo1PPos--;
  delay(speedDelay);
}

```

Por tanto, dependiendo de los botones tocados, los servos se mueven en dirección positiva o negativa. El mismo principio de funcionamiento se aplica a todos los servomotores. Para cambiar la velocidad de movimiento, utilizamos los valores provenientes del control deslizante que oscilan entre 100 y 250.

```
// If arm speed slider is changed
if (dataIn > 101 & dataIn < 250) {
    speedDelay = dataIn / 10; // Change servo speed (delay
time)
}
```

Dividiéndolos por 10 obtenemos valores de 10 a 25, que se utilizan como retraso en microsegundos en los bucles while para accionar los servos.

Para almacenar los movimientos del robot, simplemente guardamos las posiciones actuales de los servos y los pasos en matrices, cada vez que se hace clic en el botón Guardar.

```
// If button "SAVE" is pressed
if (m == 12) {
    //if it's initial save, set the steppers position to 0
    if (index == 0) {
        LeftBackWheel.setCurrentPosition(0);
        LeftFrontWheel.setCurrentPosition(0);
        RightBackWheel.setCurrentPosition(0);
        RightFrontWheel.setCurrentPosition(0);
    }
    lbw[index] = LeftBackWheel.currentPosition(); // save
position into the array
    lfw[index] = LeftFrontWheel.currentPosition();
    rbw[index] = RightBackWheel.currentPosition();
    rfw[index] = RightFrontWheel.currentPosition();

    servo01SP[index] = servo1PPos; // save position into the
array
    servo02SP[index] = servo2PPos;
    servo03SP[index] = servo3PPos;
    servo04SP[index] = servo4PPos;
    servo05SP[index] = servo5PPos;
    servo06SP[index] = servo6PPos;
    index++; // Increase the array index
    m = 0;
}
```

Luego, cuando presionamos el botón Ejecutar, llamamos a la función personalizada runSteps(). Esta función personalizada recorre todos los pasos almacenados utilizando algunos bucles for y while

```
if (m == 14) {
    runSteps();

    // If button "RESET" is pressed
```

```
if (dataIn != 14) {
    stopMoving();
    memset(lbw, 0, sizeof(lbw)); // Clear the array data to 0
    memset(lfw, 0, sizeof(lfw));
    memset(rbw, 0, sizeof(rbw));
    memset(rfw, 0, sizeof(rfw));
    memset(servo01SP, 0, sizeof(servo01SP)); // Clear the
array data to 0
    memset(servo02SP, 0, sizeof(servo02SP));
    memset(servo03SP, 0, sizeof(servo03SP));
    memset(servo04SP, 0, sizeof(servo04SP));
    memset(servo05SP, 0, sizeof(servo05SP));
    memset(servo06SP, 0, sizeof(servo06SP));
    index = 0; // Index to 0
}
}
```

Debemos tener en cuenta que comienza desde la primera posición y llega a la última posición, y lo repite una y otra vez. Por lo tanto, al guardar los pasos, necesitamos colocar el robot de manera que, el primer paso tenga la misma posición que el último. Mientras avanzamos los pasos también podremos cambiar la velocidad tanto de la plataforma como del brazo robótico, así como pausar y resetear todos los pasos.

IV. RESULTADOS Y DISCUSIÓN

La plataforma Mit App fue necesaria para la implementación de la interfaz para el móvil.

Se diseñó y construyó un robot omnidireccional de 5 GDL, que se puede desplazar sobre una base móvil con una gran precisión, capaz de posicionarse en cualquier punto intermedio, el cual lleva un valor agregado en cuanto a el espacio de trabajo, su límite de espacio de trabajo es vertical, alcanzando un máximo de 33.2 cm en su punto más alto .

El diseño se validó con el software Inventor y con las pruebas de funcionamiento, los servomotores utilizados fueron suficientes debido a la estructura de material PLA..

Se logro detallar un procedimiento adecuado de armado y puesta a funcionamiento del robot (Figura 14) empleando seguridad, criterio y normas estandarizadas.

En general, este proyecto combina la movilidad omnidireccional de las ruedas Mecanum con la capacidad de manipulación del brazo robótico de 5 grados de libertad, lo que diferencia de otros robots que tienen un espacio de trabajo limitado, Esto abre un amplio abanico de posibilidades para aplicaciones robóticas, como exploración, manipulación de objetos, inspección y más.

Uno de los desafíos mas grandes que se presento fue el factor tiempo, al tener que realizar comunicación entre la plataforma y la interfaz de usuario , por tal se recomienda tener conocimientos previos para poder realizar dicha comunicación



Fig. 14 implementación finalizada ro

V. CONCLUSIONES

Las ruedas mecanum ofrecen una velocidad de 12 cm por segundo según la programación en el arduino mega , permitiendo movimientos suaves y precisos en múltiples direcciones.

La implementación y la construcción consideraron la resistencia estructural y la capacidad de carga de un máximo de 200 gramos.

La optimización del rendimiento se logró mediante algoritmos de planificación de trayectorias, y las pruebas exhaustivas y ajustes garantizaron un funcionamiento seguro y confiable.

Las ruedas mecanum son ampliamente utilizadas en la industria de la robótica, especialmente en aplicaciones donde se requiere maniobrabilidad y movimientos precisos en espacios reducidos.

La plataforma de ruedas Mecanum le brinda al robot la capacidad de moverse en cualquier dirección sin necesidad de cambiar su orientación. Esto es posible gracias a la disposición especial de los rodillos en las ruedas, que permiten el desplazamiento lateral y diagonal.

REFERENCIAS

- [1] A. Odry, R. Fuller, I. J. Rudas, P. Odry, Kalman filter for mobile-robot attitude estimation: novel optimized and adaptive solutions, *Mechanical Systems and Signal Processing*, vol. 110, pp. 569–589, 2018.
- [2] K. J. Kalinski, M. Mazur, Optimal control of 2-wheeled mobile robot at energy performance index, *Mechanical Systems and Signal Processing*, vol. 70, 2016, 373–386.
- [3] S. Zhao, B. Huang, F. Liu, Localization of indoor mobile robot using minimum variance unbiased FIR filter, *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 2, pp 410–419, 2018.
- [4] P. Kassaeiyan, B. Tarvirdizadeh, K. Alipour, Control of tractor-trailer wheeled robots considering self-collision effect and actuator saturation limitations, *Mechanical Systems and Signal Processing*, vol. 127, pp. 388–411, 2019.
- [5] Y. Kantaros, M. M. Zavlanos, Distributed intermittent connectivity control of mobile robot networks, *IEEE Transactions on Automation Science and Engineering*, vol. 62, no. 7, pp 3109–3121, 2017.
- [6] A. M. Univ, E. C. Marseille, ST. M. Rousset, Trajectory reconfiguration for time delay reduction in the case of unexpected obstacles: application to 4-mecanum wheeled mobile robots (4-MWMR) for industrial purposes, vol. 53, no. 2, pp 15653-15658, 2020.

- [7] *Advances in Applied Sciences* 2023; 8(4): 122-130 <http://www.sciencepublishinggroup.com/j/aas> doi: 10.11648/j.aas.20230804.11 ISSN: 2575-2065 (Print); ISSN: 2575-1514 (Online) Design of a Multi-Directional Autonomous Mobile Robot with Robotic Arm for Object Retrieval in Manufacturing Systems Tri Dung Nguyen, Trung Hao Tran, Duy Phuong Dinh
- [8] *Advances in Applied Sciences* 2023; 8(4): 122-130 <http://www.sciencepublishinggroup.com/j/aas> doi: 10.11648/j.aas.20230804.11 ISSN: 2575-2065 (Print); ISSN: 2575-1514 (Online) Design of a Multi-Directional Autonomous Mobile Robot with Robotic Arm for Object Retrieval in Manufacturing Systems
- [9] Piyare, R. and Tazil, M. (2011). "Bluetooth based home automation system using Android phones". IEEE 15TH International symposium on consumer electronics (ISCE).
- [11] T. Mikolajczyk, H. Fuwen, L. Moldovan, A. Bustillo, M. Matuszewski, K. Mowicki, "Selection of machining parameters with android application made using MT APP inventor bookmarks," presented at 11th International Conference Interdisciplinarity in Engineering, INTER-ENG 2017, 5-6 October 2017, Tirgu-Mures, Romania. UTP University of Science and Technology, 7 Kaliskiego, 85-796 Bydgoszcz, Poland; North China University of Technology, 5 Jinyuanzhuang Road, Shijingshan District, 100144 Beijing, China; University of Burgos, Avda Cantabria s/n, 09006 Burgos, Spain.
- [12] F. Becker et al., "An approach to the kinematics and dynamics of a fourwheeled mecanum vehicles," *Special Issue of Scientific Journal of IFToMM "Problems of Mechanics,"* vol. 2, no. 55, pp. [27-28], 2014, Dedicated to the International Conference "Mechanics 2014."
- [13] E. W. Patton et al., "MIT App Inventor: Objectives, Design, and Development," Massachusetts Institute of Technology, Cambridge, MA, USA, e-mails: ewpatton@csail.mit.edu; ewpatton@mit.edu, mikitessenbaum@gmail.com, farzeen@mit.edu, © The Author(s) 2019, S.-C. Kong and H. Abelson (eds.), *Computational Thinking Education* ,https://doi.org/10.1007/978-981-13-6528-7_3.