

DISEÑO, CONSTRUCCIÓN Y CONTROL DE UN ROBOT SCARA CON ENFOQUE EN TECNOLOGÍA IoT

DESIGN, CONSTRUCTION AND CONTROL OF A SCARA ROBOT WITH AN IOT TECHNOLOGY APPROACH

Jorge Miguel Jaimes Ponce

Universidad Autónoma Metropolitana, Unidad Azcapotzalco, México
jjp@azc.uam.mx

Aldair Angel Rivera Sanabria

Universidad Autónoma Metropolitana, Unidad Azcapotzalco, México
al2183000368@azc.uam.mx

Marco Antonio Islas Herrera

Universidad Autónoma Metropolitana, Unidad Azcapotzalco, México
maih@azc.uam.mx

Oscar Herrera Alcántara

Universidad Autónoma Metropolitana, Unidad Azcapotzalco, México
oha@azc.uam.mx

Roberto Alfonso Alcántara Ramírez

Universidad Autónoma Metropolitana, Unidad Azcapotzalco, México
raar@azc.uam.mx

Mario Alberto Lagos Acosta

Universidad Autónoma Metropolitana, Unidad Azcapotzalco, México
mlagos@azc.uam.mx

Recepción: 14/noviembre/2024

Aceptación: 24/diciembre/2025

Resumen

Este artículo presenta el desarrollo de un sistema de control para un robot SCARA de cuatro grados de libertad. El diseño y la construcción del hardware incluyeron el desarrollo de una placa PCB para el circuito electrónico que controla el robot utilizando sistemas embebidos. Este hardware, combinado con tecnología IoT, permite el control del brazo robótico. Para el desarrollo del software, se implementa un control PID para el control del motor. Sensores de posición magnéticos (AS5600) proporcionaron retroalimentación al sistema, y un bróker local que utiliza el protocolo MQTT facilitó el intercambio continuo de datos para un

control preciso de la rotación de las articulaciones. Los movimientos suaves y precisos resultantes demuestran el logro exitoso de los objetivos del proyecto.

Palabras Clave: IoT, MQTT, PID, robot SCARA, sensores magnéticos.

Abstract

This article presents the development of a control system for a four-degree-of-freedom SCARA robot using IoT technology. The hardware design and construction included developing a PCB board for the electronic circuit that controls the robot using embedded systems. This hardware, combined with IoT technology, enables control of the robotic arm. For software development, PID control was implemented for motor control. Magnetic position sensors (AS5600) provided feedback to the system, and a local broker using MQTT protocol facilitated continuous data sharing for precise joint rotation control. The resulting smooth and precise movements demonstrate the successful achievement of the project objectives.

Keywords: IoT, magnetic sensors, MQTT, PID, SCARA robot.

1. Introducción

Los robots SCARA (Selective Compliance Articulated Robot Arm) son ampliamente utilizados en diversas aplicaciones industriales debido a su destreza, precisión y velocidad.

Los Robots de tipo SCARA son una buena opción que cumple con los requerimientos de la industria 4.0, ya que son de tamaño pequeño, lo cual los vuelve más accesibles. Además, estos pueden ser montados tanto en bases o pedestales como en el suelo [Chikurtev, 2023].

Tradicionalmente, estos robots se han controlado mediante sistemas de control centralizados. Sin embargo, el advenimiento de la tecnología IoT ha abierto nuevas posibilidades para el control de robots distribuido y flexible.

Actualmente la Industria 4.0 (que alude a la 4ta revolución industrial) se enfoca en la creación de fábricas inteligentes en donde cada uno de los elementos que las conforman y que estén involucrados en los procesos de producción puedan estar siempre comunicados unos con otros [Dalenogare, 2018], esto con el fin de mejorar

los procesos de producción, conseguir una reducción en los costos y minimizar la carga de trabajo hacía los empleados [Chikurtev, 2023].

Como sistema digital de control para cualquier robot existen diferentes opciones, una de las cuales es la Raspberry Pi 3B utilizada en este proyecto, que son computadoras de bajo costo y cuenta con procesador multinúcleo a una velocidad de procesamiento de 1.2 GHz además de 1 GB de RAM, siendo una plataforma de hardware abierto con una gran comunidad de usuarios y una amplia gama de software disponible. Esto facilita la programación y adaptación del sistema de control a las necesidades específicas del robot SCARA, [Marcin, 2019]. Una gran ventaja es que la Raspberry Pi cuenta con puertos USB, Ethernet, WiFi y Bluetooth integrados, lo que permite la conexión con otros dispositivos. Para el operador de un robot, la transmisión inalámbrica de comandos de movimiento desde una computadora host al controlador del robot ofrece mayor comodidad, utilizando las herramientas adecuadas, desde RTOS [Manh-Tuan, 2016], hasta Python sobre Linux.

El sistema SCARA es más barato que los robots de seis ejes, como el robot PUMA, que lo hace más accesible para los consumidores. Además, el uso de la impresión 3D para los componentes del brazo y la elección de motores paso a paso y controladores contribuyen a la reducción de costos. [Chung, 2022]. Uno de los objetivos es proporcionar una plataforma robótica económica para investigadores y académicos, permitiendo el aprendizaje de los fundamentos de la robótica, [Fatima, 2019]. Un aspecto muy importante, es la simulación con herramientas como Matlab con base a la obtención del modelo matemático, por ejemplo, con un análisis cinemático mediante el método Denavit-Hartenberg, con el cual se valida el funcionamiento del robot SCARA, [José Cornejo, 2022], [Subhashini, 2024], incluyendo leyes de control clásico como el PID, [Dürdane, 2023].

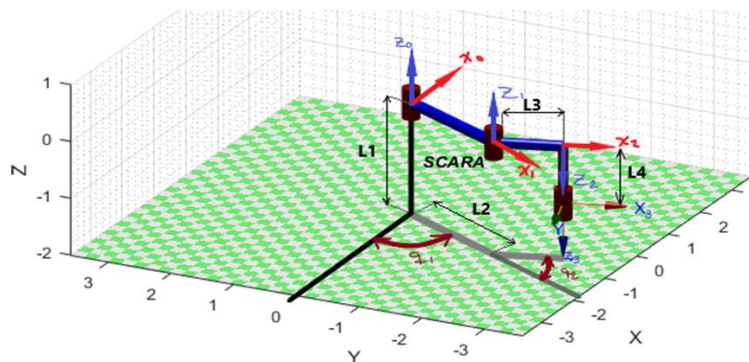
Por lo anterior se decide diseñar y construir un robot SCARA para incursionar en el control de robots manipuladoras de forma remota, con base en las tendencias de la industria 4.0 que incluye el internet de las cosas IIoT (Industrial Internet of Things), aunque en la investigación de desarrollos similares prácticamente no hay información.

2. Métodos

En este trabajo, se desarrolla un sistema de control para un robot SCARA de cuatro grados de libertad utilizando tecnología IoT [Flores, 2023]. El sistema consta de dos bloques principales: hardware y software. Por lo tanto, inicialmente para lograr el desarrollo de la programación se realiza el análisis cinemático directo e inverso con base en el método de Denavit-Hartenberg (DH) [Tay, 2022].

Análisis de cinemática directa

El método Denavit-Hartenberg (D-H) es una convención para representar las configuraciones espaciales de eslabones y articulaciones en un robot. Es especialmente útil para robots manipuladores, como el SCARA (Selective Compliance Assembly Robot Arm). Para resolver el problema de análisis cinemático directo desarrolla el modelo que se observa en la Figura 1. Un robot SCARA típicamente tiene cuatro grados de libertad (DOF): tres rotacionales y uno prismático, determinando el análisis en términos de $L1$, $L2$, $L3$, $L4$, $q1$, $q2$, $q3$ y $q4$.



Fuente: elaboración propia

Figura 1 Modelo en 3D desarrollado en MATLAB.

Para cada par de eslabones consecutivos, se definen cuatro parámetros D-H:

- θ_i : Ángulo de rotación alrededor del eje z_{i-1} (base del robot).
- d_i : Desplazamiento a lo largo del eje z_{i-1} .
- a_i : Longitud del eslabón, o distancia entre los ejes z_{i-1} y z_i medida a lo largo del eje x_i .
- α_i : Ángulo de giro, o ángulo entre los ejes z_{i-1} y z_i .

De acuerdo con el algoritmo, se llena la tabla de D-H (Tabla 1) considerando que las variables del sistema se diferencian por el asterisco, en dónde las variables q_1 , q_2 y q_4 son angulares y q_3 es lineal. Una vez definidos los sistemas de coordenadas, se pueden llenar los parámetros D-H en la tabla. Se toma en cuenta la matriz de transformación de D-H, para obtener las matrices de transformación para cada eslabón, Ecuación 1. Se procede a obtener las matrices de transformación de los eslabones, y con la finalidad de hacer más fáciles los cálculos se propone abreviar $\sin()$ y $\cos()$ por S y C respectivamente, Ecuaciones 2 a 5.

Tabla 1 Parámetros considerados para aplicar el método D-H.

A_n^{i-1}	z_{i-1}		x_i	
	θ_{i_x}	d_{i_x}	a_{i_z}	α_{i_z}
A_1^0	q_1	L_1	L_2	0
A_2^1	q_2	d	L_3	0
A_3^2	0	q_3	0	0
A_4^3	q_4	$-L_4$	0	0

Fuente: elaboración propia

$$A = \begin{pmatrix} \cos(\theta_i) & -\cos(\alpha_i) \sin(\theta_i) & \sin(\alpha_i) \sin(\theta_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i) \cos(\theta_i) & -\sin(\alpha_i) \cos(\theta_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

$$A_1^0 = \begin{pmatrix} Cq_1 & -Sq_1 & 0 & L_2 Cq_1 \\ Sq_1 & Cq_1 & 0 & L_2 Sq_1 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

$$A_2^1 = \begin{pmatrix} Cq_2 & -Sq_2 & 0 & L_3 Cq_2 \\ Sq_2 & Cq_2 & 0 & L_3 Sq_2 \\ 0 & 0 & -1 & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3)$$

$$A_3^2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4)$$

$$A_4^3 = \begin{pmatrix} Cq_4 & -Sq_4 & 0 & 0 \\ Sq_4 & Cq_4 & 0 & 0 \\ 0 & 0 & 1 & -L_4 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

La matriz de transformación queda como en Ecuación 6, con $T_4^0 = A_1^0 \cdot A_2^1 \cdot A_3^2 \cdot A_4^3$.

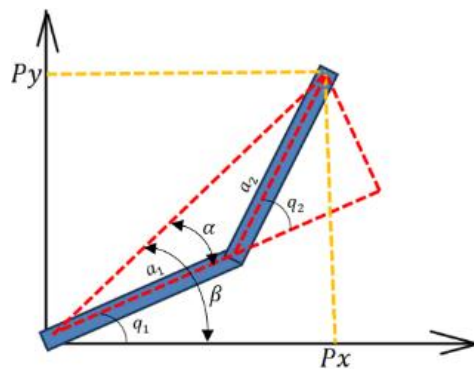
$$T_4^0 = \begin{pmatrix} Cq_1 & -Sq_1 & 0 & L_2 Cq_1 \\ Sq_1 & Cq_1 & 0 & L_2 Sq_1 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} Cq_2 & -Sq_2 & 0 & L_3 Cq_2 \\ Sq_2 & Cq_2 & 0 & L_3 Sq_2 \\ 0 & 0 & -1 & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} Cq_4 & -Sq_4 & 0 & 0 \\ Sq_4 & Cq_4 & 0 & 0 \\ 0 & 0 & 1 & -L_4 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6)$$

Utilizando la herramienta de MATLAB se realiza la multiplicación de las matrices obtenidas, y se obtiene la matriz de transformación que relaciona la primera articulación con la cuarta articulación, Ecuación 7.

$$T_4^0 = \begin{pmatrix} \cos(q_1 + q_2 - q_4) & -\sin(q_1 + q_2 - q_4) & 0 & L_3 \cos(q_1 + q_2) + L_2 \cos(q_1) \\ \sin(q_1 + q_2 - q_4) & \cos(q_1 + q_2 - q_4) & 0 & L_3 \sin(q_1 + q_2) + L_2 \sin(q_1) \\ 0 & 0 & 1 & L_1 + d - L_4 - q_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (7)$$

Análisis de cinemática inversa

Debido a que se trata de un robot tipo SCARA y posee 4 grados de libertad, se empleara el método geométrico para desarrollar el análisis de la cinemática inversa. Este método consta de obtener las coordenadas articulares q_1 y q_2 , a través de relaciones trigonométricas ubicadas en el plano XY, Figura 2.



Fuente: elaboración propia

Figura 2 Vista del plano XY para aplicar método.

Para calcular el valor de q_2 se procede aplicando la ley de cosenos, Ecuación 8.

$$c^2 = a^2 + b^2 - 2ab \cos(\alpha) \quad (8)$$

Aplicado al problema mostrado en la Figura 2, se obtiene la Ecuación 9.

$$\left(\sqrt{P_x^2 + P_y^2}\right)^2 = a_1^2 + a_2^2 - 2a_1a_2 \cos(\pi - q_2) \quad (9)$$

Cómo $\cos(\pi - \theta) = -\cos(\theta)$, sustituyendo se obtiene la Ecuación 10.

$$\cos(q_2) = \frac{P_x^2 + P_y^2 - (a_1^2 + a_2^2)}{2a_1a_2} \quad (10)$$

Desarrollando la ecuación 10 para despejar el ángulo q_2 se obtiene la Ecuación 11.

$$q_2 = \arccos\left(\frac{P_x^2 + P_y^2 - (a_1^2 + a_2^2)}{2a_1a_2}\right); \text{ si } \{P_x < 0 \text{ \& } P_y < 0\} \rightarrow q_2 < 0 \quad (11)$$

Para determinar el ángulo de q_1 , a partir de lo observado en la Figura 2, se obtiene la Ecuación 12. Donde α y β corresponden a las Ecuaciones 13 y 14.

$$q_1 = \beta - \alpha \quad (12)$$

$$\beta = \arctan\left(\frac{P_y}{P_x}\right) \quad (13)$$

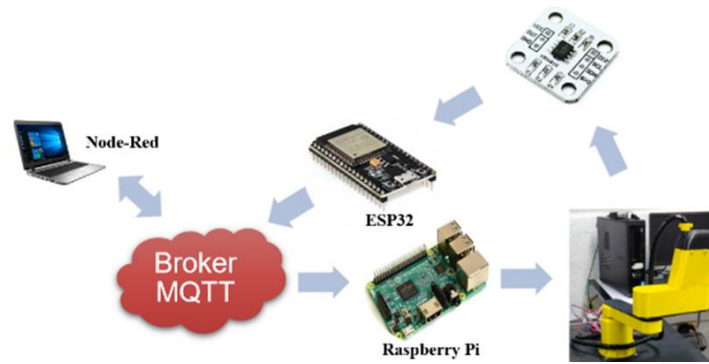
$$\alpha = \arctan\left(\frac{a_2 \sin(q_2)}{a_1 + a_2 \cos(q_2)}\right) \quad (14)$$

Por lo que al sustituir en la ecuación 12, las ecuaciones 13 y 14, se obtiene la ecuación 15, para determinar el valor del ángulo q_1 . Con base en este análisis, no se consideraron los parámetros q_3 y q_4 .

$$q_1 = \arctan\left(\frac{P_y}{P_x}\right) - \arctan\left(\frac{a_2 \sin(q_2)}{a_1 + a_2 \cos(q_2)}\right) \quad (15)$$

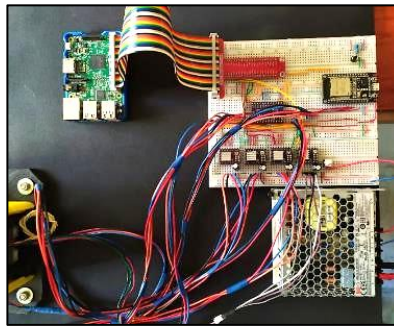
Hardware

El hardware del sistema se compone de una placa PCB que contiene el circuito electrónico que controla el robot (Figura 3). La placa PCB incluye dos sistemas embebidos Raspberry Pi [Tkaczick,2019] y ESP32, controladores de motor (DRV8825), sensores de posición magnéticos (AS5600) y otros componentes electrónicos necesarios. Se pruebo el circuito realizado en tres tablillas de experimentación antes de desarrollar la PCB para comprobar su funcionamiento, la Figura 4, muestra el ensamble de los diferentes dispositivos electrónicos.



Fuente: elaboración propia

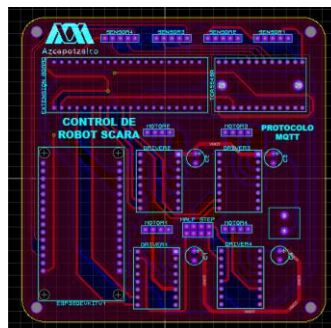
Figura 3 Circuito de control en placa de pruebas.



Fuente: elaboración propia

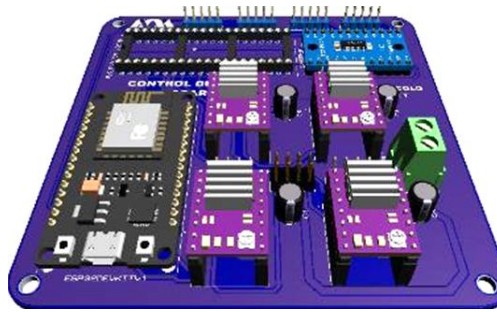
Figura 4 Circuito de control en placa de pruebas.

Una vez que se comprueba el funcionamiento del circuito de control, se procede a realizar la PCB, la Figura 5, muestra el resultado obtenido por medio del software Proteus. La tarjeta electrónica se diseña con las dimensiones de 10 *cm* de largo por 10 *cm* de ancho. Existe una sección en Proteus que permite visualizar en tres dimensiones el resultado obtenido de la PCB diseñada, la Figura 6, muestra el posible resultado. De esta manera se aprueba la producción de la tarjeta.



Fuente: elaboración propia

Figura 5 PCB del circuito de control para el robot SCARA.

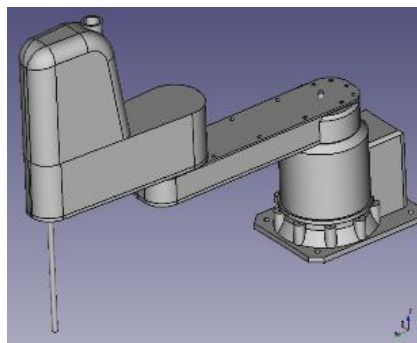


Fuente: elaboración propia

Figura 6 Visualización en 3D de la PCB.

Software

El software del sistema para control del robot SCARA, Figura 7, se compone de dos partes principales: el firmware del microcontrolador y la aplicación de control del usuario. El firmware del microcontrolador es responsable de controlar los motores del robot de acuerdo con las instrucciones recibidas de la aplicación de control del usuario. La aplicación de control del usuario permite controlar el robot de forma remota a través de una interfaz gráfica.



Fuente: elaboración propia

Figura 7 Visualización en 3D del robot SCARA.

Se desarrolla el firmware de la Raspberry con la finalidad de controlar los cuatro motores a pasos destinados a mover cada articulación del robot. Con base en la información recibida de los sensores manipulados por la ESP32. Se aplica control PID a cada articulación para obtener movimientos suaves y precisos, por lo que este control clásico se programa, la Figura 8, muestra el inicio del programa y los factores k , k_i y k_d configurados.

Se utiliza una tarjeta con base en el ESP32, la cual es para manipular la información recibida de los sensores magnéticos adaptados a los cuatro motores a pasos y encargados de realizar la retroalimentación de la posición de cada articulación. Para lograr la comunicación con los sensores, se utiliza el protocolo I2C y al tener la misma dirección todos, se hace uso del multiplexor TCA9548A. La Figura 9, muestra la programación aplicada en la ESP32. Además, se desarrolla una interfaz de usuario utilizando *Node-RED*. Para evitar el uso de un bróker externo y que posiblemente presente alta latencia, también se instala MOSQUITTO en una computadora personal, la Figura 10, muestra el comando ejecutado después de la instalación del programa y que indica la inicialización exitosa del bróker.

```
import paho.mqtt.client as mqtt
import RPi.GPIO as GPIO
import time
import json

# Pines GPIO
DIR1 = 4
STEP1 = 17
DIR2 = 27
STEP2 = 22
DIR3 = 10
STEP3 = 9
DIR4 = 11
STEP4 = 0

# Configuración de GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup([DIR1, DIR2, DIR3, DIR4], GPIO.OUT)
GPIO.setup([STEP1, STEP2, STEP3, STEP4], GPIO.OUT)

# Configuración MQTT
broker_address = "broker.emqx.io"
port = 1883
topics = ["Positions", "Degrees"]

# Constantes de dirección
CW = True
CCW = False

# Parámetros PID (por motor)
pid_params = [
    {"K": 50, "ki": 0.05, "kd": 0.0000}, # Motor 1
    {"K": 50, "ki": 0.05, "kd": 0.0000}, # Motor 2
    {"K": 55, "ki": 0.01, "kd": 0.0001}, # Motor 3
    {"K": 15, "ki": 0.001, "kd": 0.000}  # Motor 4
]
```

Fuente: elaboración propia

Figura 8 Programación realizada en la Raspberry Pi.

```
#include <AS5600.h>
#include <Wire.h>
#include <TI_TCA9548A.h>
#include <ArduinoJson.h>
#include <WiFi.h>
#include <PubSubClient.h>

//Definición de pines a utilizar de la ESP32
#define BUILTIN_LED 2
#define GIR2 27
#define GIR3 26
#define GIR4 13
#define GIR5 14
#define MSG_BUFFER_SIZE (100)

// Configuración con los valores de la RED WIFI y servidor MQTT
const char* ssid = "SSID";
const char* password = "Password";
const char* mqtt_server = "broker.emqx.io";

//Creación de tipo de dato para cada sensor
AS5600 as56001; // GIR2
AS5600 as56002; // GIR3
AS5600 as56003; // GIR4
AS5600 as56004; // GIR5

//Datos necesarios para la conexión vía WIFI
WiFiClient espClient;
```

Fuente: elaboración propia

Figura 9 Programación aplicada en la ESP32.

```
Simbolo del sistema - mosquito -v -c test.conf
Microsoft Windows [Versión 10.0.19045.4651]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Acer Aspire E5>cd c:/program files/mosquitto

c:\Program Files\mosquitto>mosquitto -v -c test.conf
1721171212: mosquitto version 2.0.18 starting
1721171212: Config loaded from test.conf.
1721171212: Opening ipv6 listen socket on port 1883.
1721171212: Opening ipv4 listen socket on port 1883.
1721171212: mosquitto version 2.0.18 running
```

Fuente: elaboración propia

Figura 10 Inicialización del bróker MOSQUITTO con el símbolo del sistema.

También se instala *Node-RED* en la misma computadora, la Figura 11, muestra el comando utilizado para activar *Node-RED* y los mensajes que indican el correcto funcionamiento, se logra conectar la *ESP32* y la *Raspberry Pi* con el bróker, lo que facilita la comunicación con base en el protocolo MQTT.

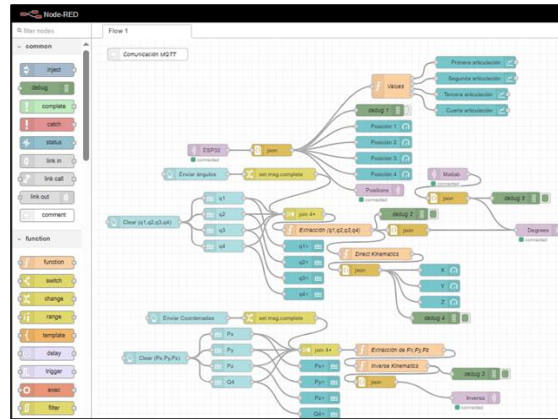
```
node-red
Microsoft Windows [Versión 10.0.19045.4651]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Acer Aspire E5>node-red
16 Jul 17:07:04 - [info]
Welcome to Node-RED
=====
16 Jul 17:07:04 - [info] Node-RED version: v3.1.3
16 Jul 17:07:04 - [info] Node.js version: v20.11.0
16 Jul 17:07:04 - [info] Windows_NT 10.0.19045 x64 IE
16 Jul 17:07:05 - [info] Loading palette nodes
16 Jul 17:07:07 - [info] Dashboard version 3.6.2 started at /ui
16 Jul 17:07:07 - [info] Settings file : C:\Users\Acer Aspire E5\node-red\settings.js
16 Jul 17:07:07 - [info] Context store : 'default' [module=memory]
16 Jul 17:07:07 - [info] User directory : C:\Users\Acer Aspire E5\node-red
16 Jul 17:07:07 - [warn] Projects disabled : editorTheme.projects.enabled=false
16 Jul 17:07:07 - [info] Flows file : C:\Users\Acer Aspire E5\node-red\flows.json
16 Jul 17:07:07 - [warn]
-----
Your flow credentials file is encrypted using a system-generated key.
If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
16 Jul 17:07:08 - [info] Server now running at http://127.0.0.1:1880/
16 Jul 17:07:08 - [info] Starting flows
16 Jul 17:07:08 - [info] Started flows
16 Jul 17:07:08 - [info] [mqtt-broker:eca403307e8369d1] Connected to broker: mqtt://loc
```

Fuente: elaboración propia

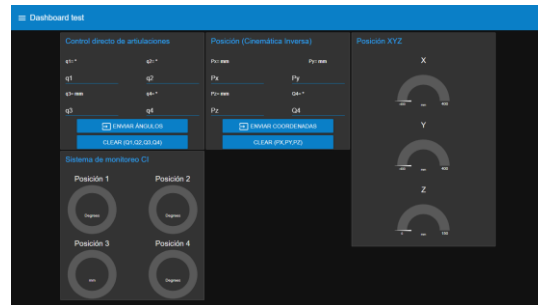
Figura 11 Inicialización de *Node-RED* a través del símbolo del sistema.

Node-RED puede trabajar con diferentes protocolos de comunicación, sin embargo, para el proyecto se decide utilizar *MQTT*. Entonces en la misma computadora personal, se inicia el ambiente de *Node-RED* para desarrollar la conexión de nodos necesaria, Figura 12. *Node-RED* tiene una sección para visualizar la interfaz de usuario, donde se pueden enviar posiciones, coordenadas cartesianas y al mismo tiempo monitorear en tiempo real las posiciones del robot. La Figura 13 muestra el resultado obtenido donde se observa la sección del control directo de las articulaciones y una segunda sección donde se configura la posición en coordenadas cartesianas.



Fuente: elaboración propia

Figura 12 Conexión de nodos realizada en Node-RED.



Fuente: elaboración propia

Figura 13 Interfaz de usuario controla y monitorea las posiciones del robot.

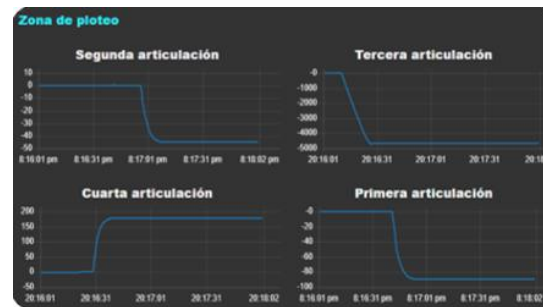
3. Resultados

El sistema de control desarrollado se prueba exhaustivamente y funciona de manera efectiva. El robot es capaz de realizar una variedad de tareas, incluyendo: Diseño y desarrollo de sistema electrónico de control, Bróker funcional para protocolo MQTT y Programación de PID.

Los resultados experimentales demuestran que el sistema de control es preciso y confiable, debido a que, siempre llega a la posición indicada a pesar de la pérdida de pasos que tienen los motores a paso cuando se obstaculiza su movimiento. NodeRED es tan versátil que el dashboard no sólo se puede modificar desde una computadora, por lo tanto, se realiza las pruebas de movimiento con un smartphone, lo importante es que este se conecte a la misma red wifi donde se encuentra el bróker. Se configura para $P1[0,0,0,0]$ a $P2[-90, -45, 52, 180]$ y a $P3[60, -30, 37, 0]$, tal como lo muestran las Figuras 14 y 15.



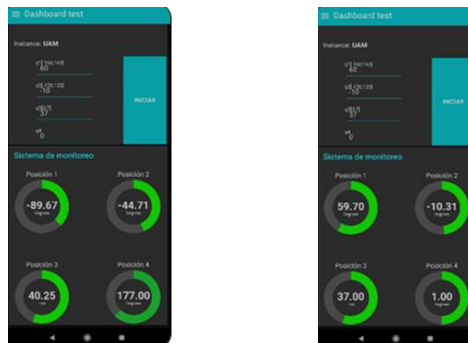
a) Estado inicial de la interfaz. b) Configuración de $P[-90, -45, 52, 180]$.



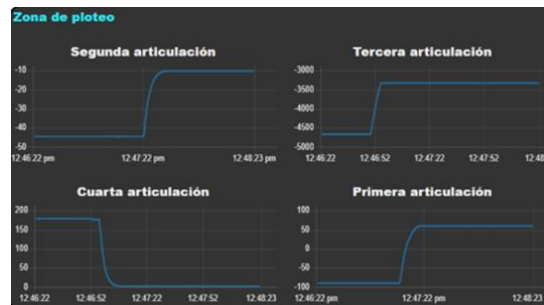
c) Respuesta de cada una de las articulaciones.

Fuente: elaboración propia

Figura 14 Primera prueba de movimiento.



a) Configuración de la posición. b) Posición alcanzada.



c) Respuesta de cada articulación.

Fuente: elaboración propia

Figura 15 Segunda prueba de movimiento.

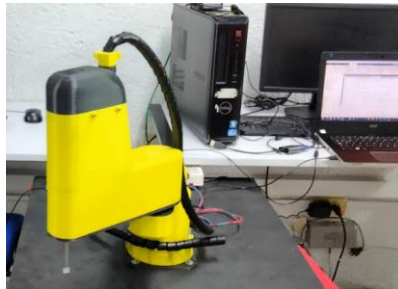
4. Discusión

El sistema de control desarrollado en este trabajo tiene varias ventajas. En primer lugar, el sistema es distribuido, lo que lo hace más flexible y escalable. En segundo lugar, el sistema utiliza tecnología IoT, lo que lo hace más fácil de integrar con otros sistemas. En tercer lugar, el sistema es preciso y confiable debido a la aplicación de control PID, [Dürdane, 2023].

Surgió la cuestión de si es posible abrir dos símbolos del sistema en simultaneo y después de investigar se logró trabajar con las dos aplicaciones al mismo tiempo. Cabe mencionar que esta es la primera etapa del proyecto, debido a que la aplicación de control PID y la implementación de ecuaciones de cinemática resultaron un éxito, se planea como segunda etapa la aplicación de las ecuaciones resultantes de la dinámica del robot.

5. Conclusiones

En este trabajo, se ha desarrollado un sistema de control para un robot SCARA de cuatro grados de libertad utilizando tecnología IoT, Figuras 16 y 17.



Fuente: elaboración propia

Figura 16 Robot SCARA.



Fuente: elaboración propia

Figura 17 Hardware del Robot SCARA.

El sistema se ha probado exhaustivamente y funciona de manera efectiva. El sistema tiene varias ventajas sobre los sistemas de control tradicionales y tiene un gran potencial para su uso en aplicaciones industriales. Para controlar las articulaciones, se utiliza información de los sensores AS5600, comunicados mediante el protocolo I2C. Se identificó la necesidad de un multiplexor, específicamente el TCA9548A, para aplicaciones con múltiples sensores del mismo tipo.

Aunque la implementación de una red de microcontroladores no estaba contemplada inicialmente, se programó el control utilizando la ESP32 para la adquisición de datos de los sensores y la Raspberry Pi 3B para el control de los actuadores. Las pruebas demostraron resultados satisfactorios, lo que sugiere la viabilidad de continuar el desarrollo del proyecto.

6. Referencias y Bibliografía

- [1] Apostol, M., Kaczmarczyk, G., & Tkaczyk, K., (2019). SCARA robot control based on Raspberry Pi. 2019 20th International Carpathian Control Conference (ICCC). IEEE. <https://doi.org/10.1109/ICCC.2019>.
- [2] Chikurtev, D., Al-Moghrabi, N., & Chikurteva, A., (2023). Control of SCARA robot and rotary table in ROS and Gazebo as part of a cyber-physical system. In 2023 9th International Conference on Control, Decision and Information Technologies (CoDIT) (pp. 1355–1360). IEEE. <https://doi.org/10.1109/CoDIT.2023>.
- [3] Chung-Wen, H., Lin, Y. T., & Jiang, G. Y., (2022). A SCARA-based cake painting system. In 2022 IEEE International Conference on Consumer Electronics – Taiwan (ICCE-TW). IEEE. <https://doi.org/10.1109/ICCE-TW.2022>.
- [4] Cornejo, J., Cruz, V., Carrillo, F., & Cerda, R., (2022). Mechatronics design and kinematic simulation of SCARA robot to improve safety and time processing of Covid-19 rapid test. In 2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT). IEEE. <https://doi.org/10.1109/ICEEICT.2022>.
- [5] Dalenogare, L. S., Benitez, G. B., Ayala, N. F., & Frank, A. G., (2018). The expected contribution of Industry 4.0 technologies for industrial performance.

- International Journal of Production Economics, 204, pp. 383–394.
<https://doi.org/10.1016/j.ijpe.2018.08.019>.
- [6] Deliktaş, D., Aktaş, R., İlgen, S., & Gülbahçe, E., (2023). Performance evaluation of a SCARA robot manipulator for trajectory tracking based on error criteria minimization. In 2023 7th International Symposium on Innovative Approaches in Smart Technologies (ISAS). IEEE. <https://doi.org/10.1109/ISAS.2023>.
- [7] Fatima, S., Farooq, A., Irfan, M., Ansar, T., & Chumuang, N., (2019). Design and fabrication of an affordable SCARA 4-DOF robotic manipulator for pick and place objects. In 2019 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing. IEEE. <https://doi.org/10.1109/IJAINLP.2019>.
- [8] Flores, J. P., (2023). Diseño e implementación de un brazo robótico, basado en IoT, para el laboratorio de la carrera de ingeniería electrónica, Tesis de grado, Universidad Mayor de San Andrés.
- [9] Ha, M. T., & Kang, C. G., (2015). Wireless-communicated computed-torque control of a SCARA robot and two-dimensional input shaping for a spherical pendulum. In 2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI). IEEE. <https://doi.org/10.1109/URAI.2015>.
- [10] Subhashini, T., Tanisha, A., & Rao, T. R., (2024). Modelling, simulation and kinematic analysis of a modified SCARA manipulator (RRRRP). In Proceedings of the 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS 2024). IEEE. <https://doi.org/10.1109/ICCROBINS.2024>.
- [11] Tay, S. H., Choong, W. H., & Yoong, H. P., (2022). A review of SCARA robot control system. In 2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAET), pp. 1–6. IEEE. <https://doi.org/10.1109/IICAET.2022>.
- [12] Tkaczyk, M., Apóstol, M., & Kaczmarczyk, G., (2019). SCARA robot control based on Raspberry Pi. In 2019 20th International Carpathian Control Conference (ICCC), pp. 1–6. IEEE. <https://doi.org/10.1109/ICCC.2019>.