UNIVERSIDAD DE TARAPACÁ
Universidad del Estado

Ingeniería
Computación e Informática

BRAZO ROBÓTICO

Integrantes: Eduardo Suaña
Dylan Calderón
Matías Agriano
Benjamín Sucso

Profesor: Baris Klobertanz
Asignatura: Proyecto I

ÍNDICE

INTRODUCCIÓN

En este proyecto, se construyó un brazo robótico utilizando el sistema LEGO Spike Prime para realizar tareas de movimiento y agarre. El objetivo fue crear un programa en Python que permitiera manejar el robot de forma inalámbrica desde un computador mediante Bluetooth. Gracias a esto, el usuario puede controlar con precisión el giro, la altura y la garra del brazo a través de una pantalla con botones fáciles de usar.

REQUERIMIENTOS FUNCIONALES

- **RF1 MOVIMIENTO ROTATIVO**

El brazo robótico debe ser capaz de rotar su base en 360 grados.

- **RF2 MOVIMIENTO FLEXIBLE**

El brazo robótico debe poder articular el codo para alcanzar un objeto.

- **RF3 MANIPULACIÓN DE CARGA**

La garra debe abrirse y cerrarse para sujetar firmemente un bloque de LEGO estándar sin dejarlo caer durante el traslado.

- **RF4 TELEOPERACIÓN**

El sistema debe permitir el control manual de todos los motores a través de la interfaz gráfica.

- **RF5 PARADA DE EMERGENCIA**

La interfaz debe contar con una función para detener todos los motores inmediatamente en caso de una situación de riesgo.

REQUERIMIENTOS NO FUNCIONALES

• USABILIDAD

La interfaz gráfica debe permitir que un usuario sin experiencia previa logre conectar el robot y realizar un movimiento básico en menos de 30 segundos. Los controles deben estar etiquetados claramente en español.

• RENDIMIENTO

El tiempo de respuesta entre la pulsación de un botón en la interfaz (Cliente) y la reacción del motor del robot (Servidor) debe ser inferior a 200 milisegundos para garantizar una teleoperación fluida.

• SEGURIDAD

En caso de pérdida de conexión Bluetooth, el robot debe detener todos sus motores automáticamente en un lapso no mayor a 1 segundo para evitar daños.

• DISPONIBILIDAD

El sistema debe ser capaz de mantener la conexión activa y operativa durante al menos 10 minutos continuos, tiempo estimado para una demostración estándar.

ARQUITECTURA DE SOFTWARE

01

Modelo de trabajo: El sistema funciona como un equipo de dos partes: un Cliente (el computador) y un Servidor (el robot LEGO).

02

El cerebro: Es donde está la pantalla de control (GUI). Aquí el usuario da las órdenes usando botones creados con la herramienta Tkinter.

03

El ejecutor: Recibe las órdenes y mueve los motores. Funciona con un programa llamado MicroPython que lee los comandos enviados desde el PC.

04

La conexión: Se comunican sin cables a través de Bluetooth (BLE), lo que permite manejar el brazo a distancia de forma rápida.

IMPLEMENTACIÓN CLIENTE

01 **Lenguaje utilizado:** Todo el programa de control fue programado en MicroPython.

02 **Interfaz Visual (GUI):** Se utilizó la herramienta Tkinter para crear una ventana con botones fáciles de usar.

03 **Procesamiento de órdenes:** El PC traduce las acciones del usuario en la pantalla en comandos digitales que el robot puede ejecutar.

04 **Conexión al Robot:** Se utilizó una herramienta llamada pybricksdev que permite enviar las órdenes por bluetooth de forma automática.

05 **Control Total:** Desde esta interfaz se puede conectar el robot, mover los motores, abrir o cerrar la garra.

IMPLEMENTACIÓN SERVIDOR

01 Lenguaje utilizado: El código interno que mueve los motores también utiliza MicroPython.

02 Cerebro del sistema: Se utiliza el Hub de LEGO Spike Prime como el receptor que procesa todas las instrucciones.

03 Ejecución de órdenes: El robot recibe los comandos digitales del PC y los convierte en movimientos físicos reales.

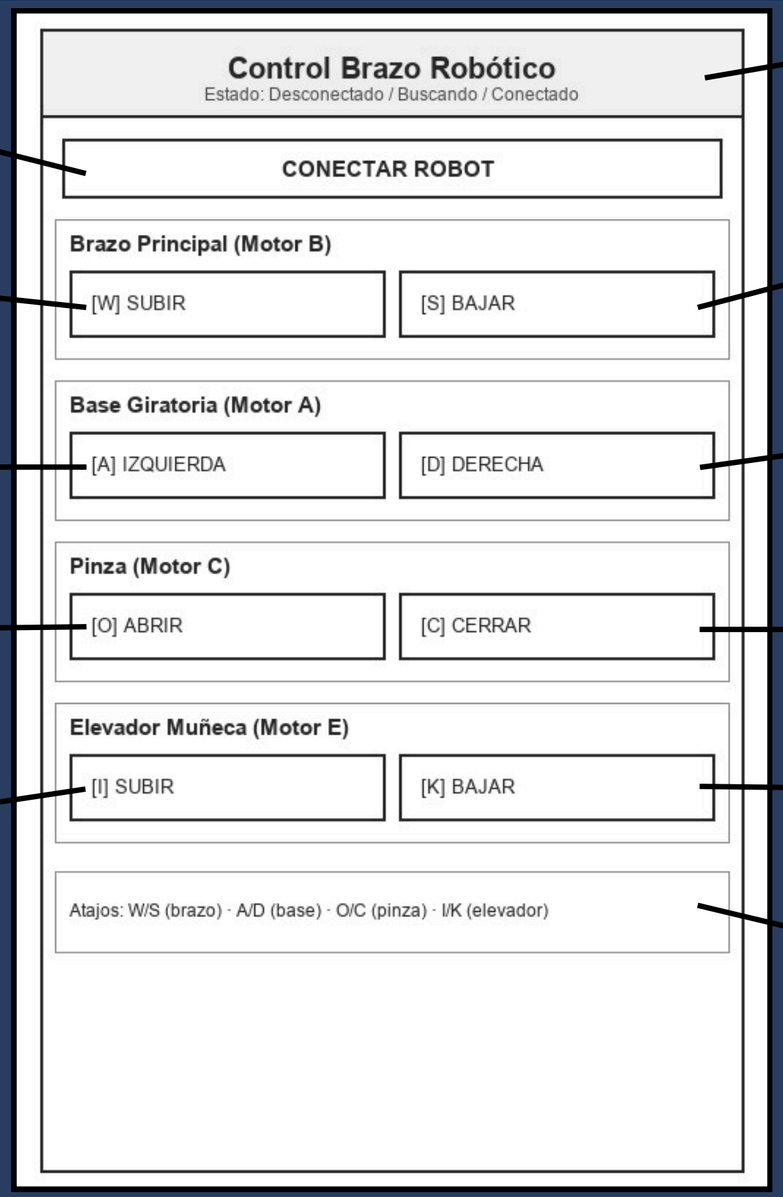
04 Sistema de Control: Usa la librería Pybricks para gestionar con precisión la velocidad y el ángulo de cada motor.

05 Acción física: Desde aquí se ejecutan las órdenes de girar la base, mover el codo y accionar la garra de forma inmediata.

IMPLEMENTACIÓN GUI

Se diseñó una ventana de control con Python y Tkinter que sirve como centro de mando. Esta interfaz permite conectar el robot por Bluetooth y manejar con precisión los movimientos de la base, el brazo y la pinza. Además, incluye indicadores que muestran en tiempo real si el sistema está conectado correctamente o si hay algún error, facilitando un manejo claro y directo del robot.

WIREFRAME GUI



Botón que inicializa la conexión por BLE entre el cliente y el hub del robot.

Botón que permite subir el brazo principal de robot.

Botón que permite rotar la base del robot en sentido antihorario.

Botón que permite abrir la garra o pinza del robot.

Botón que permite elevar la muñeca del robot.

Apartado que permite visualizar el estado de conexión con el robot.

Botón que permite bajar el brazo principal del robot.

Botón que permite rotar la base de robot en sentido horario.

Botón que permite cerrar la garra del robot.

Botón que permite bajar la muñeca del robot.

Apartado de atajos para controlar el robot con teclas.

FUNDAMENTO DE LOS MOVIMIENTOS

Objetivo del Análisis: Confirmar que la potencia de los motores es suficiente para elevar la carga de la mina (20 cm) de forma segura y sin interrupciones.

Datos:

Masa de carga(m): 0,15 kg
Gravedad (g): 9,8 m/s^2
Distancia(h): 0,2 m

Cálculo del peso:

$$P = m \cdot g \longrightarrow 0{,}15\ kg \cdot 9{,}8\ m/s^2 = 1{,}47\ N$$

Cálculo de Trabajo Mecánico:

$W = P \cdot h \longrightarrow 1{,}47\ N \cdot 0{,}20\ m = 0{,}294\ J$

Resultado de validación:

El requerimiento de 0,294 Joules asegura que el brazo puede realizar la tarea de elevación sin forzar los motores, garantizando un funcionamiento estable y duradero.

ESTADO ACTUAL

Requisito Inicial	Estado	Detalle del cumplimiento
Control remoto	Cumplido	El sistema utiliza comunicación BLE gestionada por la clase RobotController para enviar comandos desde el PC al Hub Lego Spike Prime.
Control de los motores	Cumplido	Los cuatro motores responden correctamente a los comandos enviados por la GUI y los atajos de teclado.
Estabilidad de posición	Cumplido	Se implementó la función motor.hold() en los comandos de liberación de los ejes verticales asegurando que el brazo mantenga la carga contra la gravedad.
Interfaz de usuario	Cumplido	Se desarrolló una interfaz gráfica en tkinter que visualiza el estado de la conexión y proporciona un control táctil y por teclado, optimizando la usabilidad.
Capacidad de manipulación	Cumplido	El brazo puede realizar un ciclo completo de trabajo (acercamiento, agarre, elevación, movimiento lateral y liberación) con la carga nominal.

PROBLEMAS Y SOLUCIONES

01

Problema: Bloqueo de la Interfaz Gráfica (GUI)

Solución: Implementación de asyncio (multitarea).

02

Problema: Caída del brazo por gravedad.

Solución: Uso de freno activo (motor.hold()).

03

Problema: Latencia en la respuesta

Solución: Optimización de protocolo a caracteres simples.

CONCLUSIONES

Se logró construir un brazo robótico funcional que se controla de forma inalámbrica uniendo el hardware con el software. El aprendizaje clave fue el uso de programación asíncrona, técnica que se aplicó para evitar que la pantalla de control se bloqueara al conectar el Bluetooth.

Además, tuvimos que corregir ciertas partes del informe como lo es la arquitectura del software, puesto que, no se específico adecuadamente sus componentes.

MUCHAS

GRACIAS