

Instituto Tecnológico de Estudios Superiores de Monterrey



Implementación de robótica inteligente & Manchester Robotics

Manchester Robotics: Challenge 1

Profesores:

Rigoberto Cerino Jiménez

Dr. Mario Martinez

Integrantes

Daniel Castillo López A01737357

Emmanuel Lechuga Arreola A01736241

Paola Rojas Domínguez A01737136

3 de Abril de 2025

Índice

Índice.....	1
Resumen.....	2
Objetivos.....	2
Introducción.....	2
Puzzlebot (elementos y su interconexión).....	3
Comunicación SSH con el Puzzlebot.....	3
Teleop Twist Keyboard.....	4
Solución del problema.....	4
Resultados.....	4
Conclusiones.....	5
Referencias.....	6

Resumen

El resumen presenta una síntesis de los aspectos más relevantes del proceso de investigación y desarrollo llevado a cabo, destacando los objetivos, la metodología empleada y los principales resultados obtenidos.

El mini reto (challenge 1) tiene como objetivo principal familiarizarse con el puzzlebot, comprendiendo su estructura, interconexiones y métodos de comunicación, Para ello se configura el Puzzlebot Jetson y se establece una comunicación remota a través de wifi utilizando SSH.

Además, se instala y utiliza el paquete teleop twist keyboard en un dispositivo externo para teleoperador el puzzlebot, verificando sus velocidades y aceleraciones máximas y mínimas. Se emplea también ROS 2 para la publicación y recepción de mensajes en los tópicos de control de velocidad y encoders, asegurando una correcta interacción con el robot.

Objetivos

En esta sección se presentan el objetivo general y los objetivos particulares del reto, los cuales están enfocados

Objetivo General: Familiarizarse con el Puzzlebot, comprendiendo sus componentes principales, interconexiones y métodos de comunicación, así desarrollar la capacidad para teleoperarlo mediante comandos del teclado en un dispositivo externo.

Objetivos particulares:

- Configurar un Puzzlebot Jetson.
- Comunicarse de manera remota desde el Puzzlebot mediante un punto de acceso WiFi (Hotspot) y conectarse al Puzzlebot a través de ssh.
- Utilizar teleop twist keyboard en un dispositivo externo para operar de forma remota el Puzzlebot
- Verificar las velocidades y aceleraciones máximas/mínimas que puede manejar el Puzzlebot, para las velocidades lineales y angulares.

Introducción

La introducción presenta el tema de investigación, proporcionando el contexto necesario para comprender la relevancia del reto a realizar. Se ofrece una visión general que facilita la construcción de un esquema mental sobre las metodologías utilizadas, además de los conceptos y tecnologías que se abordarán en el desarrollo del reporte.

Este reto se desarrolla para el aprendizaje y desarrollo de sistemas de robótica móvil utilizando ROS 2. Se busca proporcionar a los estudiantes una comprensión práctica del Puzzlebot, un robot de código abierto desarrollado con ese mismo fin y que por medio de la jetson nano es posible.

Se introducen conceptos clave como la configuración del sistema, la comunicación remota mediante SSH y Wifi así como el control de robot a través de comandos de teclado usando Ros 2. Además, se exploran las velocidades y aceleraciones del puzzlebot con el fin de evaluar su desempeño en tareas con uso teleoperativo.

Este ejercicio es fundamental para adquirir habilidades en la integración de hardware junto con el software en robótica, sentando las bases para desarrollar aplicaciones avanzadas para la automatización y el control de robots móviles.

Componentes principales del Puzzlebot

El Puzzlebot es una plataforma de robótica móvil diseñada para el aprendizaje, la investigación y el desarrollo de algoritmos avanzados en navegación autónoma, visión por computadora e inteligencia artificial. Su arquitectura modular permite a los usuarios trabajar con diferentes configuraciones de hardware y software, adaptándose a distintos niveles de complejidad según la aplicación deseada.

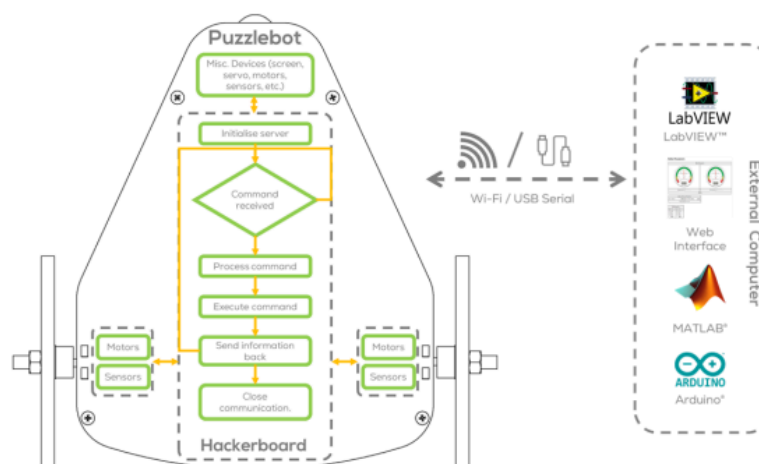


Figura 1: Diagrama de sistema de control

Como presenta la Figura 1 da una descripción detallada del robot móvil Puzzlebot, que está basado en una Hackerboard, que actúa como el procesador principal del sistema que está directamente conectado con los dispositivos periféricos como motores, sensores y otros dispositivos que se pueden agregar.

- Plataforma de procesamiento
 - El puzzlebot cuenta con diferentes ediciones que utilizan diversas plataformas de procesamiento como por ejemplo podemos encontrar con características específicas.
 - **Hacker Edition:** Utiliza la Hackerboard, una plataforma diseñada para el procesamiento en tiempo real, ideal para tareas como control de bajo nivel, navegación, evitación de obstáculos y SLAM basado en LiDAR
 - **NVIDIA Jetson Edition:** Equipada con una Jetson Nano u Orin Nano, que combinada con la Hackerboard así permite el desarrollo de algoritmos avanzados en inteligencia artificial y visión por computadora
 - **RPi Edition:** Basada en la Raspberry Pi 5, una plataforma asequible y versátil, ideal para aprendizaje y desarrollo de algoritmos de localización, planificación de rutas y navegación
- Sensores
 - **LiDAR:** Utilizado para mapeo avanzado y navegación autónoma.
 - **Cámara Raspberry Pi:** Incluida en la edición Jetson y RPi, útil para tareas de visión por computadora.
 - **Sensor TOF (Time of Flight):** Presente en la edición Jetson, permite mediciones de distancia con alta precisión
- Motores

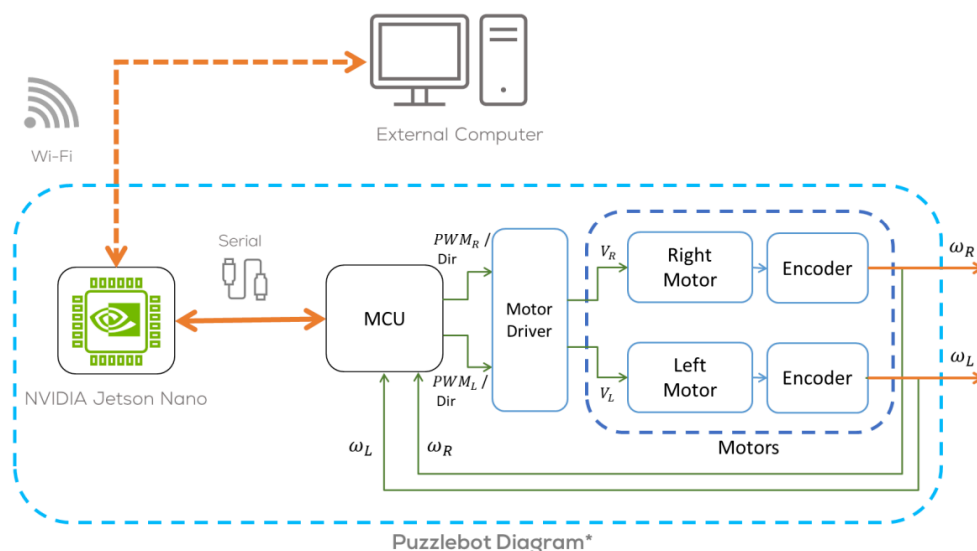


Figura 2. Diagrama de sistema del puzzlebot.

- Como se muestra en la Figura 2, los motores están conectados a la plataforma de procesamiento y esto permite aplicar algunos algoritmos de control para garantizar un movimiento eficiente.

- **Sistemas de comunicación**
 - El sistema de Puzzbot cuenta con diversas opciones para la comunicación y el control.
 - **Modo autónomo:** En este debido caso se forma por la programación y el control de este mismo se basa en su plataforma de procesamiento
 - **WI-FI y comunicación en serie:** Esto permite tener conexión con el control del robot mediante una computadora externa esta misma conexión es compatible con múltiples entornos de desarrollo como: ROS, MATLAB, LabVIEW.

Análisis del funcionamiento de la "Hackerboard" y su comunicación con el Puzzle Bot

La Hackerboard es el núcleo de control de bajo nivel del Puzzlebot, diseñada para procesamiento en tiempo real, en la Figura 3, podemos apreciar sus componentes.

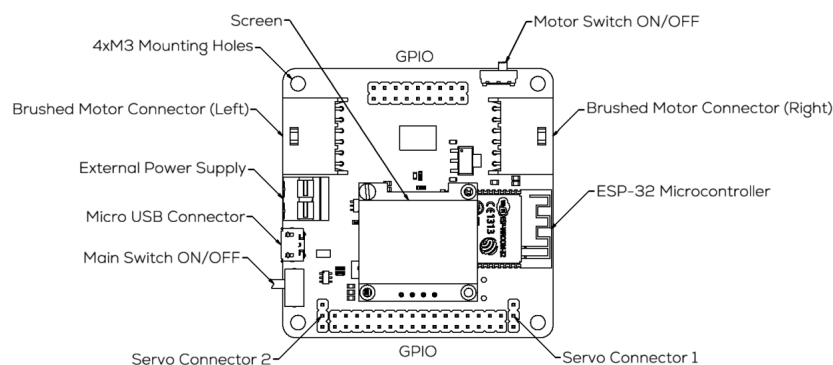


Figura 3. Hackerboard.

Funciones Clave

Procesamiento en tiempo real:

- Ejecuta algoritmos críticos como control de motores, evasión de obstáculos y SLAM básico (2d LIDAR) y navegación.
- Compatible con sensores como encoders, LIDAR y cámaras.

Comunicación con el Puzzlebot:

- **Modo autónomo (Configuración Standalone):**
 - La Hackerboard ejecuta directamente el código cargado por el usuario (arduino IDE).
 - Usa bibliotecas de MCR2 para interactuar con sensores y actuadores.
- **Modo externo:**
 - Se comunica con una computadora externa o una placa de alto rendimiento (Jetson/RPi) mediante USB Serial o Wi-Fi.
 - Transmite datos de sensores (ej: lecturas de encoders) y recibe comandos de movimiento (ej: velocidades de rueda).

Arquitectura Técnica

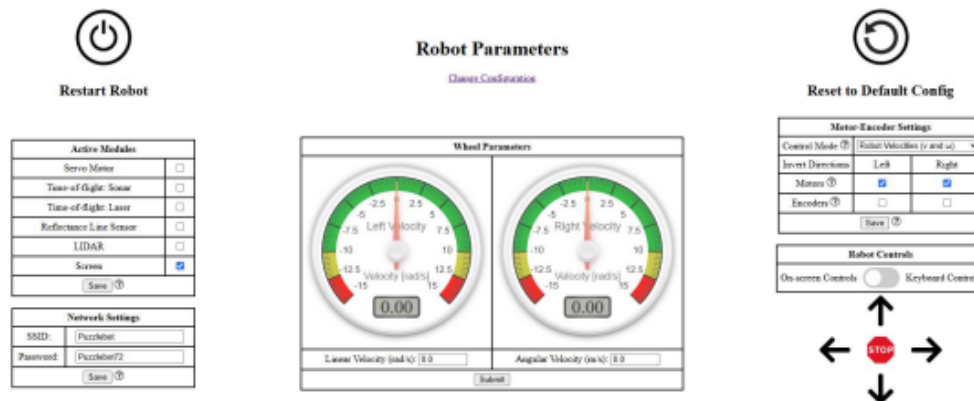


Figura 4. Parámetros del robot

Entradas/Salidas:

- Sensores: Encoders, LIDAR, cámaras, TOF (Time-of-Flight).
- Actuadores: Motores DC, servomotores, LEDs (p. 11MCR2_Puzzlebot_Introduction_V2.pdf).

Protocolos de Comunicación:

- UART y SPI para comunicación con periféricos.
 - Para esto se deben de configurar en la página web proporcionada, que tiene un aspecto como se presenta en la Figura 1.
- ROS Serial para integración con ROS (p. 21).

Ejemplo de funcionamiento de Flujo de Datos

1. La Hackerboard lee datos de un encoder. Figura 4.
2. Procesa la información para calcular la odometría.
3. Envía los resultados al Jetson Nano (vía serial) para decisiones de navegación.
4. Recibe comandos de velocidad y los aplica a los motores.

Comunicación SSH con el Puzzlebot

El protocolo Secure Shell (SSH) es un estándar criptográfico que permite la administración remota segura de dispositivos conectados a una red. SSH proporciona autenticación robusta y encriptación de datos, evitando accesos no autorizados y ataques de intermediarios. (Stallings, 2017).

Para establecer una conexión SSH con un Puzzlebot, es esencial que tanto el ordenador desde el cual se accede como el robot estén conectados a la misma red.

Pasos para configurar una comunicación SSH:

1. Configurar un punto de acceso WiFi (Hotspot) en el Puzzlebot. Para facilitar la conexión remota, el Puzzlebot puede configurarse como un punto de acceso (hotspot), permitiendo que otros dispositivos se conecten directamente a él sin necesidad de un enrutador externo.
2. Obtener la Dirección IP del Robot
3. Acceder al Robot mediante SSH con el comando: `ssh usuario@direccion_ip_robot`
4. Se solicitará la contraseña correspondiente al usuario especificado, una vez ingresada correctamente se establecerá la conexión segura.

Con la sesión SSH activa, es posible ejecutar comandos, transferir archivos y monitorear el estado del Puzzlebot de manera remota. Gracias a SSH, el Puzzlebot puede integrarse en redes más amplias y ser operado a distancia, facilitando su uso en escenarios experimentales.

Teleop Twist Keyboard.

Solución del problema

A continuación, se describe la metodología empleada para alcanzar los objetivos del reto, detallando los elementos utilizados, las funciones implementadas y el desarrollo del código de programación.

Empleamos una metodología en búsqueda de encontrar la velocidad máxima y mínima que pueda soportar los motores de nuestro Puzzlebot, usando los comandos proporcionados para poder controlar de manera remota los valores para nuestro dispositivo y encontrar los valores que mejor se acomodan.

Después de desarrollar la conexión Wi-Fi entre el Puzzlebot con nuestra computadora externa este proceso se hará por medio de Micro-ROS. Ejecutamos el agente de Micro-ROS para verificar la disponibilidad de los tópicos de ROS 2 y poder ejecutar un topic list para seleccionar el canal de las velocidades de los motores DC.

A continuación instalamos en el dispositivo externo el ROS teleop twist keyboard package, esto nos permitirá poder controlar el dispositivo de forma remota con nuestro teclado, dicho comando establecerá las instrucciones y los cambios que se realicen al dispositivo. Con esta herramienta nos permite incrementar o decrecer la velocidad de nuestros motores, planteando la posibilidad de saber cuales son los valores máximos y mínimos que puede soportar el Puzzlebot, para la velocidad lineal y angular.

Nuestro método establece en poder ver los valores de velocidad del motor derecho e izquierdo con el comando “ *ros2 topic echo /VelocituEncR*” y “ *ros2 topic echo /VelocituEncL*” en la terminal como los valores de velocidad en nuestros motores, así podremos incrementar la lineal y angular de nuestros motores analizando hasta cuál es su valor máximo que es capaz de alcanzar y por el otro lado de igual manera decrecer los valores hasta que nuestro motor llegue a cero en otros término que detenga su movimiento.

Resultados

En este apartado se insertan las evidencias del funcionamiento del reto, agregando descripciones de lo que representa cada una de estas, así como un análisis de tales resultados, para saber si se consideran resultados satisfactorios o completos.

[Video de Evidencia](#)

Conclusiones

Por último, se presentan los logros alcanzados a lo largo del desarrollo del reto, analizando el grado de cumplimiento de los objetivos planteados. Se evalúa si estos fueron alcanzados en su totalidad, identificando las razones detrás de su éxito o, en caso contrario, los factores que pudieron haber limitado su cumplimiento. Asimismo, se proponen posibles mejoras a la metodología utilizada, con el propósito de optimizar los resultados obtenidos.

Referencias

En este apartado se anexan los elementos consultados para el desarrollo del tema de investigación.

Stallings, W. (2017). *Cryptography and network security: Principles and practice* (Seventh edition). Pearson.

ManchesterRoboticsLtd. (s. f.-c).

TE3002B_Intelligent_Robotics_Implementation_2025/Week1/Presentations/PDF/MC

R2_Puzzlebot_Jetson_Ed_ROS2.pdf at main ·

ManchesterRoboticsLtd/TE3002B_Intelligent_Robotics_Implementation_2025.

GitHub.

https://github.com/ManchesterRoboticsLtd/TE3002B_Intelligent_Robotics_Implementation_2025/blob/main/Week1/Presentations/PDF/MCR2_Puzzlebot_Jetson_Ed_ROS2.pdf

ManchesterRoboticsLtd. (s. f.-b).

TE3002B_Intelligent_Robotics_Implementation_2025/Week1/Presentations/PDF/MC

R2_Puzzlebot_Introduction_V2.pdf at main ·

ManchesterRoboticsLtd/TE3002B_Intelligent_Robotics_Implementation_2025.

GitHub.

https://github.com/ManchesterRoboticsLtd/TE3002B_Intelligent_Robotics_Implementation_2025/blob/main/Week1/Presentations/PDF/MCR2_Puzzlebot_Introduction_V2.pdf

ManchesterRoboticsLtd. (s. f.-e).

TE3002B_Intelligent_Robotics_Implementation_2025/Week1/Presentations/PDF/MC

R2_Robotics_Introduction_V3.pdf at main ·

ManchesterRoboticsLtd/TE3002B_Intelligent_Robotics_Implementation_2025.

GitHub.

https://github.com/ManchesterRoboticsLtd/TE3002B_Intelligent_Robotics_Implementation_2025/blob/main/Week1/Presentations/PDF/MCR2_Robotics_Introduction_V3.pdf