

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.....	3
Introducción.....	4
Componentes principales del Puzzlebot.....	4
Plataformas de Procesamiento.....	5
Sensores Integrados.....	5
Sistema de Actuación.....	5
Sistemas de Comunicación.....	6
La Hackerboard y su Interacción con el Puzzlebot.....	6
Arquitectura Técnica del Puzzlebot.....	7
Ejemplo de Flujo de Datos en el Puzzlebot.....	7
Comunicación SSH con el Puzzlebot.....	8
Teleop Twist Keyboard.....	9
Solución del problema.....	9
Resultados.....	11
Conclusiones.....	12
Referencias.....	13

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 en las primeras interacciones con el Puzzlebot

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.

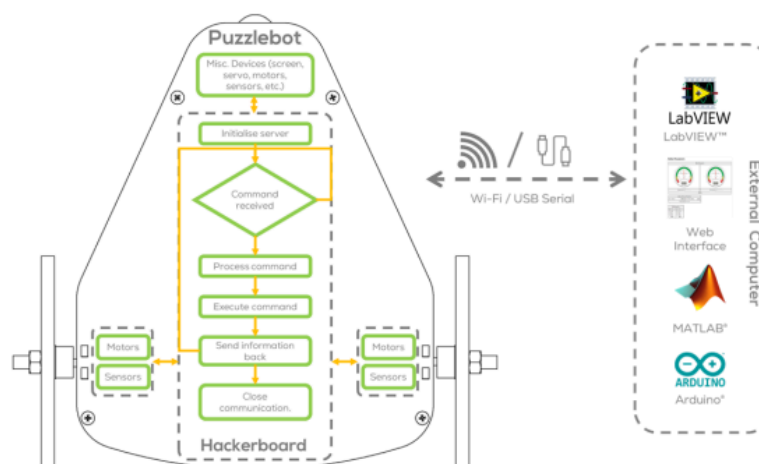


Figura 1: Diagrama de sistema de control

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.

Plataformas de Procesamiento

El Puzzlebot cuenta con diferentes ediciones que emplean diversas plataformas de procesamiento, cada una con características específicas:

- **Hacker Edition:** Utiliza la Hackerboard, una plataforma especializada en procesamiento en tiempo real, ideal para el control de bajo nivel, la navegación, la evitación de obstáculos y la implementación de algoritmos de SLAM basados en LiDAR.
- **NVIDIA Jetson Edition:** Equipada con una Jetson Nano u Orin Nano, que en combinación con la Hackerboard, permite el desarrollo de algoritmos avanzados en inteligencia artificial y visión por computadora.
- **RPi Edition:** Basada en la Raspberry Pi 5, una plataforma versátil y asequible, idónea para el aprendizaje y el desarrollo de algoritmos de localización, planificación de rutas y navegación.

Sensores Integrados

El Puzzlebot está equipado con diversos sensores que le permiten recopilar información del entorno y mejorar su capacidad de navegación:

- **LiDAR:** Utilizado para la generación de mapas y la navegación autónoma.
- **Cámara Raspberry Pi:** Disponible en las ediciones Jetson y RPi, es utilizada para tareas de visión por computadora.
- **Sensor TOF (Time of Flight):** Presente en la edición Jetson, permite realizar mediciones de distancia con alta precisión.

Sistema de Actuación

El sistema de actuación del Puzzlebot está compuesto por motores conectados a la plataforma de procesamiento, lo que permite la aplicación de algoritmos de control para optimizar el movimiento del robot. En la Figura 2 se ilustra esta relación, destacando cómo los motores reciben instrucciones de la plataforma para ejecutar desplazamientos eficientes.

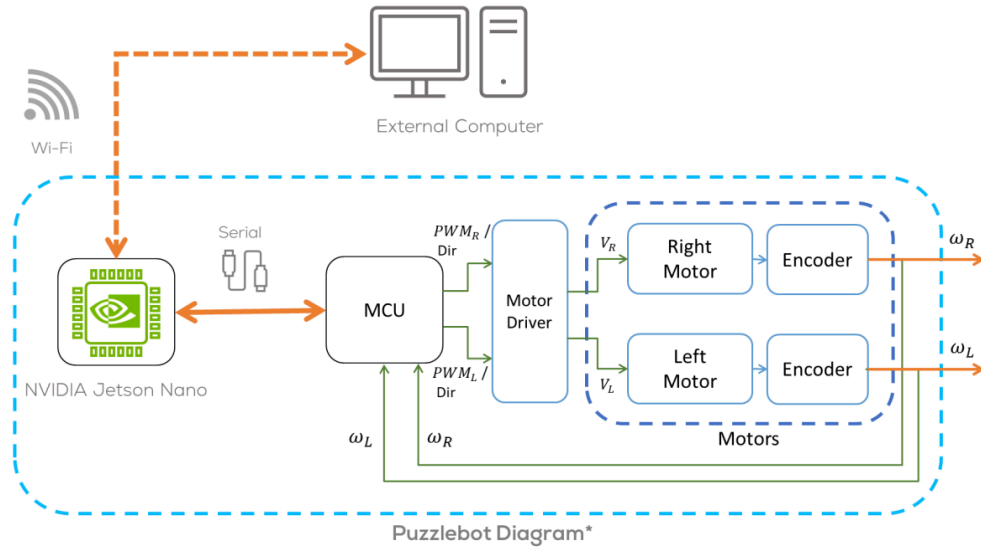


Figura 2. Diagrama de sistema del puzzlebot.

Sistemas de Comunicación

El Puzzlebot incorpora diferentes opciones para la comunicación y el control, facilitando su integración en diversos entornos de desarrollo:

- **Modo autónomo:** En esta configuración, el robot opera de manera independiente, ejecutando la programación preestablecida en su plataforma de procesamiento.
- **Wi-Fi y comunicación en serie:** Estas opciones permiten la interacción con computadoras externas y la integración con entornos de desarrollo como ROS, MATLAB y LabVIEW.

La Hackerboard y su Interacción con el Puzzlebot

La Hackerboard constituye el núcleo de control de bajo nivel del Puzzlebot, diseñada para procesamiento en tiempo real. Como se observa en la Figura 3, esta plataforma ejecuta funciones clave como el control de motores, la evitación de obstáculos y la implementación de SLAM en dos dimensiones utilizando LiDAR.

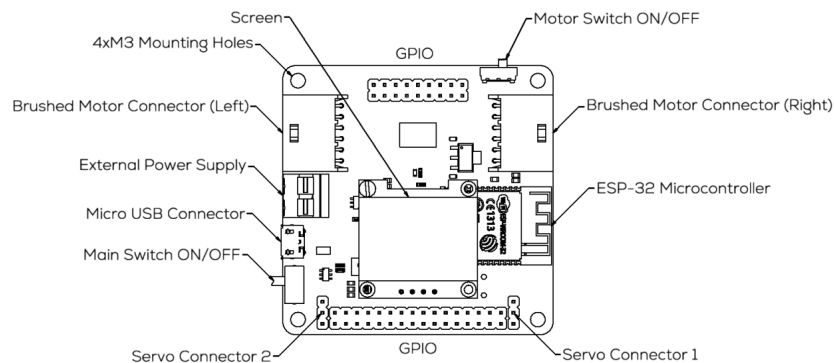


Figura 3. Hackerboard.

La Hackerboard permite dos modos de operación principales:

1. **Modo autónomo (Standalone):** En esta configuración, la Hackerboard ejecuta el código cargado por el usuario mediante el entorno de desarrollo Arduino IDE, utilizando bibliotecas de MCR2 para la interacción con sensores y actuadores.
2. **Modo externo:** En este modo, la Hackerboard se comunica con una computadora externa o con una plataforma de alto rendimiento (Jetson o RPi) a través de USB Serial o Wi-Fi. Transmite datos de los sensores, como las lecturas de los encoders, y recibe comandos de movimiento que se aplican a los motores.

Arquitectura Técnica del Puzzlebot

La arquitectura del Puzzlebot está compuesta por diversas entradas y salidas que permiten la interacción con su entorno y su integración en sistemas de control más complejos en la Figura 4 se pueden apreciar algunos de estos.

- **Entradas/Sensores:** Incluye encoders, LiDAR, cámaras y sensores TOF.
- **Salidas/Actuadores:** Comprende motores DC, servomotores y LEDs.
- **Protocolos de comunicación:** Emplea UART y SPI para la comunicación con periféricos, así como ROS Serial para la integración con el sistema ROS.

Ejemplo de Flujo de Datos en el Puzzlebot

El funcionamiento del Puzzlebot se basa en la transmisión eficiente de datos entre sus componentes. Por ejemplo, la Hackerboard puede leer datos de un encoder (Figura 4), procesar la información para calcular la odometría y enviar los resultados a una Jetson Nano a través de comunicación serial para la toma de decisiones de navegación. Posteriormente, la Jetson Nano envía comandos de velocidad a la Hackerboard, que los aplica a los motores para ejecutar los desplazamientos correspondientes.

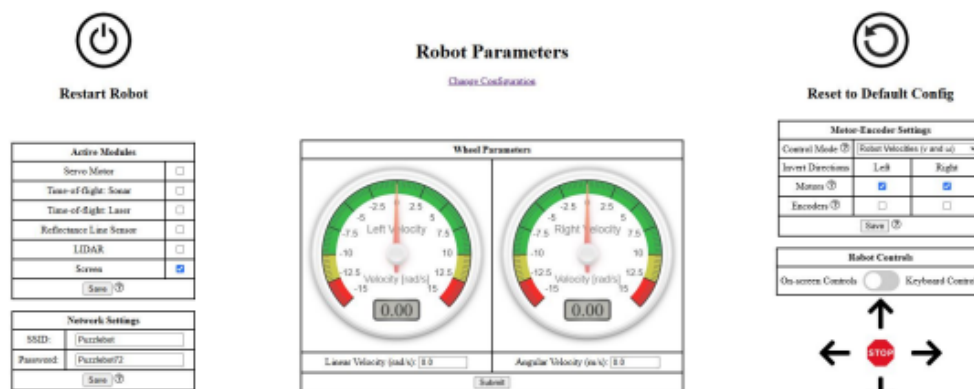


Figura 4. Parámetros del robot

Con esta arquitectura, el Puzzlebot ofrece un entorno versátil para la investigación y el desarrollo en robótica, permitiendo la implementación de algoritmos complejos en un sistema modular y flexible.

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.

El teleop Twist Keyboard es un paquete de ROS que permite controlar el robot móvil de manera remota utilizando el teclado de la computadora, dicho paquete permite enviar comandos de velocidad lineal y angular al robot que son interpretados por el nodo que controla el movimiento del Puzzlebot.

```
danielc@dan-L0Q-16IRH8:~$ ros2 run teleop_twist_keyboard teleop_twist_keyboard

This node takes keypresses from the keyboard and publishes them
as Twist/TwistStamped messages. It works best with a US keyboard layout.
-----
Moving around:
  u      i      o
  j      k      l
  m      ,      .

For Holonomic mode (strafing), hold down the shift key:
-----
  U      I      O
  J      K      L
  M      <      >

t : up (+z)
b : down (-z)

anything else : stop

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%

CTRL-C to quit

currently:      speed 0.5      turn 1.0
```

Figura 5: Terminal instrucciones Teleop Twist Keyboard

De esta forma su instalación es descargar el paquete dentro de la computadora externa y mientras nosotros estamos conectados a la red Wi-Fi del robot, podremos establecer los comandos para ajustar la velocidad de sus motores para moverse en la dirección indicada, estos están establecidos en la terminal como lo vemos en la Figura.

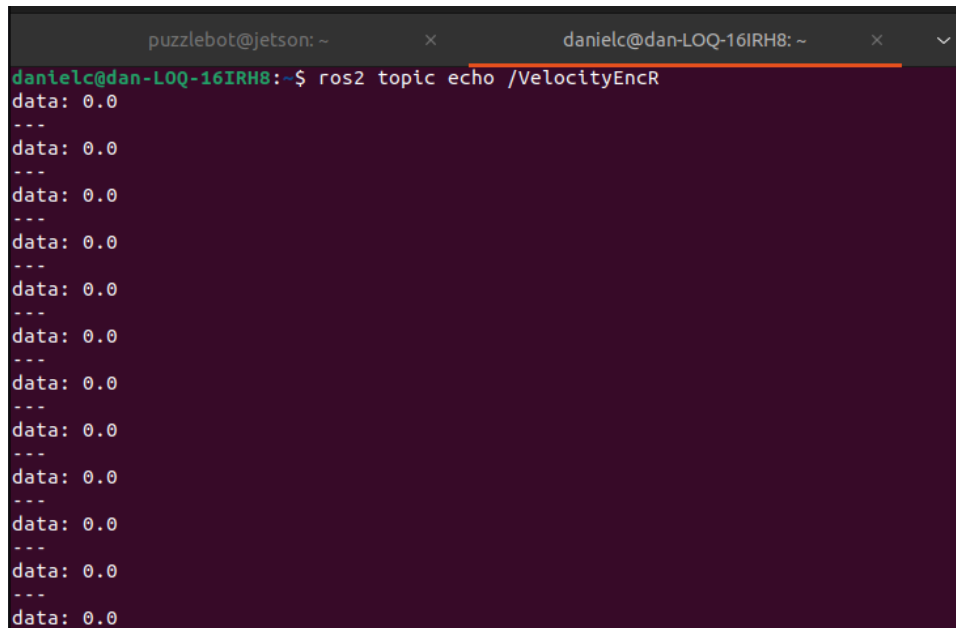
Solución del problema

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 esté 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.

A terminal window with a dark background and light text. The window title bar shows two tabs: 'puzzlebot@jetson: ~' and 'danielc@dan-LOQ-16IRH8: ~'. The active tab is 'danielc@dan-LOQ-16IRH8: ~'. The terminal content shows the command 'ros2 topic echo /VelocityEncR' being executed, followed by a series of 'data: 0.0' outputs, each preceded by three dashes '---'.

```
puzzlebot@jetson: ~ x danielc@dan-LOQ-16IRH8: ~ x v
danielc@dan-LOQ-16IRH8:~$ ros2 topic echo /VelocityEncR
data: 0.0
---
data: 0.0
---
data: 0.0
---
data: 0.0
---
data: 0.0
---
data: 0.0
---
data: 0.0
---
data: 0.0
---
data: 0.0
---
data: 0.0
---
data: 0.0
---
data: 0.0
```

Figura 6:Terminal echo motor derecho

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 podemos encontrar en la Figura , muestra 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.

Con las pruebas que desarrollamos, notamos que al iniciar el movimiento de nuestros motores necesitan estar desde cierto punto para que el cambio de velocidad repentina de cero al movimiento, no reinicie nuestro robot. Así concluimos que nuestra velocidad media para el inicio del robot sea

currently: speed 0.19371024450000007 turn 1.0

Los resultados de nuestra velocidad maxima y minima de la angular es respectivamente de

currently: speed 0.19371024450000007 turn 7.400249944258172

currently: speed 0.19371024450000007 turn 0.39093953617604027

De igual manera en la velocidad lineal siendo el maximo y minimo es de

currently: speed 0.00046800017613798494 turn 1.0

En el siguiente enlace, se dirigirá a un video en la plataforma de youtube, donde se puede observar en paralelo el funcionamiento del puzzlebot junto con la pantalla de la laptop por la cual se le van mandando los comandos para que se vayan ejecutando.

[Video de Evidencia](#)

Se cumple de manera satisfactoria la parte de los objetivos tanto del objetivo general como de los objetivos particulares, donde se cumple con la configuración del puzzlebot, la parte de familiarizarse con el sistema para la obtención de velocidades máximas y mínimas tanto lineal como angular, comunicación remota por medio de wifi con ayuda de un hotspot.

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.

La investigación y desarrollo del reto Challenge 1 han permitido una comprensión práctica del Puzzlebot, desde su configuración hasta su teleoperación mediante ROS 2. A través de la implementación de una comunicación remota vía SSH y el uso del paquete teleop twist keyboard, se logró verificar los límites de velocidad y aceleración del robot. Además, se estableció un análisis detallado de su arquitectura técnica y sistemas de comunicación, permitiendo una integración efectiva entre hardware y software. Los resultados obtenidos demuestran el cumplimiento de los objetivos planteados, sentando una base sólida para el desarrollo de aplicaciones avanzadas en robótica móvil con ayuda del puzzlebot y sus componentes.

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.

ROS Package: teleop_twist_keyboard. (s. f.). https://index.ros.org/p/teleop_twist_keyboard/

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