



Tecnológico de Monterrey

Reto semanal 1. Manchester Robotics

Carlos Adrián Delgado Vázquez A01735818

Alfredo Díaz López A01737250

Juan Paulo Salgado Arvizu A01737223

03 de abril del 2025

Resumen

En este primer challenge se realizó el setup al PuzzleBot Jetson escribiendo la imagen correcta proporcionada en la página oficial de Nvidia, usando Balena Etcher. De la misma manera se activó un hotspot para poder comunicarse con la jetson vía comunicación ssh. Una vez realizado esto se corrió el nodo en micro ros para publicar en los tópicos de velocidad del PuzzleBot. Finalmente se instaló el paquete teleop twist keyboard para poder manejar el PuzzleBot remotamente.

Objetivos

El objetivo es familiarizarse con el PuzzleBot Jetson, configurarlo correctamente y establecer comunicación mediante ROS2, con la intención de poner las bases para futuros desafíos de navegación autónoma

Para lograrlo, se necesita lo siguiente:

- Configuración del PuzzleBot Jetson y activación del hotspot
- Ejecución del agente de micro-ROS
- Teleoperación remota del PuzzleBot

Introducción

El avance de la robótica móvil ha permitido el desarrollo de plataformas educativas y experimentales que facilitan la investigación en navegación autónoma, control de movimiento y comunicación en sistemas embebidos (Robotics, s.f.). El PuzzleBot es un robot educativo basado en hardware y software libre, diseñado para trabajar con el sistema operativo ROS 2 (Robot Operating System 2), proporcionando un entorno flexible para pruebas e implementaciones en robótica, en la Ilustración 1, vemos un ejemplo de Puzzlebot.

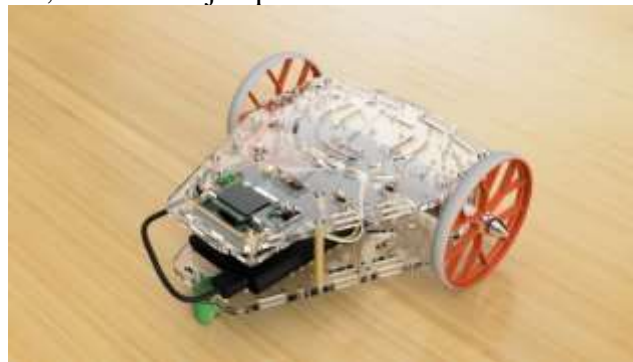


Ilustración 1 Puzzlebot

El PuzzleBot está compuesto por varios elementos clave, incluyendo una unidad de procesamiento, motores con encoders para la medición de velocidad, sensores y una interfaz de comunicación. Para este proyecto, se utilizó un Jetson Nano, como se ve en la Ilustración 2, una plataforma de cómputo embebida diseñada para aplicaciones de inteligencia artificial y robótica, la cual actúa como nodo central del sistema, permitiendo el procesamiento de datos y la ejecución de algoritmos de control (Zelmer, 2024).



Ilustración 2 Jetson Nano

Una de las herramientas fundamentales utilizadas en este proyecto es la comunicación SSH (Secure Shell), un protocolo de red que permite el acceso remoto seguro a dispositivos embebidos. A través de SSH, es posible establecer una conexión entre la computadora de desarrollo y el Jetson Nano, lo que facilita la configuración del entorno, la ejecución de comandos y la interacción con el PuzzleBot sin necesidad de conexión física directa. Este método es ampliamente utilizado en sistemas distribuidos y en el desarrollo de aplicaciones robóticas, ya que permite el control remoto sin afectar la ejecución del sistema en tiempo real (Deyimar, 2025). En la Ilustración 3, podemos ver el funcionamiento de este mismo.

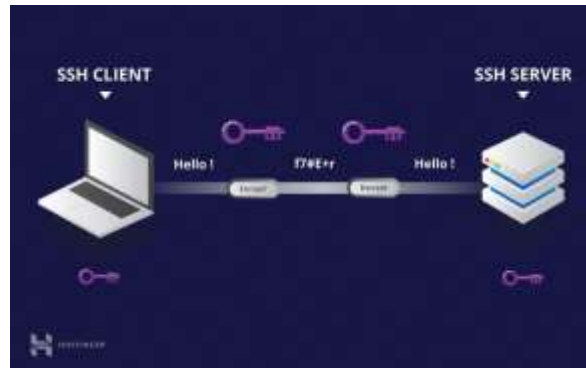


Ilustración 3 Ejemplo SSH

Para la operación remota del PuzzleBot, se implementó el paquete Teleop Twist Keyboard, una herramienta de ROS 2 que permite controlar el movimiento del robot utilizando el teclado de una computadora externa. Este paquete publica mensajes de tipo `geometry_msgs/Twist` en el tópico `/cmd_vel`, que son interpretados por el sistema del robot para ajustar su velocidad lineal y angular. Con esta implementación, se pueden realizar pruebas de velocidad, verificando la respuesta del sistema en distintos escenarios.

A continuación, se mostrará los controles del mismo paquete (ROS, s.f.).

Reading from the keyboard and Publishing to Twist!

Moving around:

u i o
j k l
m , .

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%
anything else : stop

CTRL-C to quit

Finalmente, podemos ver en la , un ejemplo de como funciona el Teleop Twist Keyboard.



Ilustración 4 Funcionamiento del Teleop Twist Keyboard

Solución del problema

METODOLOGÍA:

1. Configuración del entorno

- Se configuró la Jetson Nano para generar una red Wi-Fi y permitir la conexión SSH.
- Se estableció la conexión SSH con el PuzzleBot mediante:
 - ***Ssh puzzlebot@10.42.0.101***

2. Verificación de comunicación con micro-ROS

- Se ejecutó el agente de micro-ROS en la Jetson Nano:
 - ***ros2 run micro_ros_agent micro_ros_agent serial --dev /dev/ttyUSB0***
(ADJUNTAR CAPTURA DE LA CONEXIÓN CON EL AGENTE)
- Se verificó la correcta comunicación listando los tópicos disponibles:
 - ***ros2 topic list***
- Se confirmó la presencia de los tópicos /cmd_vel, /VelEncL, /VelEncR, entre otros.
(ADJUNTAR LA CAPTURA DE LOS TOPICOS)

3.Publicación y suscripción a tópicos

- Se enviaron comandos de velocidad al robot publicando en /cmd_vel:
 - *ros2 topic pub /cmd_vel geometry_msgs/Twist "linear: {x: 0.2, y: 0.0, z: 0.0} angular: {x: 0.0, y: 0.0, z: 0.5}"*
- Se monitorearon las respuestas de los encoders en /VelEncL y /VelEncR:
 - *ros2 topic echo /VelEncL*
ros2 topic echo /VelEncR

4.Implementación de teleoperación en un PC externo:

- Se instaló el paquete de teleoperación en un PC externo:
 - *sudo apt install ros-humble-teleop-twist-keyboard -y*
- Se ejecutó el nodo de teleoperación para controlar el PuzzleBot con el teclado:
 - *ros2 run teleop_twist_keyboard teleop_twist_keyboard*
- Se probó el control de velocidad y aceleración del robot.

DESCRIPCIÓN DE ELEMENTOS Y FUNCIONES IMPLEMENTADAS

- **Jetson Nano:** Procesador principal del PuzzleBot, ejecutando ROS2.
- **Micro-ROS Agent:** Permite la comunicación entre el microcontrolador y ROS2.
- **Tópicos ROS2:** Manejan la comunicación del robot:
 - /cmd_vel: Control de movimiento.
 - /VelEncL y /VelEncR: Datos de velocidad de los encoders.
- **Teleop_twist_keyboard:** Control remoto del robot mediante teclado.

Resultados

1. Conexión ssh:

```
(base) brunene@brunene-rog:~$ ssh puzzlebot@10.42.0.100
puzzlebot@10.42.0.100's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 4.9.337-tegra aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

57 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Wed Jan 17 13:04:57 2024 from 10.42.0.197
```

2. Agente serial:

```
1: puzzlebot@jetson: ~
puzzlebot@jetson:~$ ros2 launch puzzlebot_ros micro_ros_agent.launch.py
[INFO] [launch]: All log files can be found below /home/puzzlebot/.ros/log/2024-01-17-13-04-51-528687-jetson-5378
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [micro_ros_agent-1]: process started with pid [5380]
[micro_ros_agent-1] [1705496692.434475] info | TermiosAgentLinux.cpp | init
| running... | fd: 3
[micro_ros_agent-1] [1705496692.435247] info | Root.cpp | set_verb
ose_level | logger setup | verbose_level: 4
[micro_ros_agent-1] [1705496693.246022] info | Root.cpp | create_c
lient | create | client_key: 0x289EAB81, session_id:
0x81
[micro_ros_agent-1] [1705496693.246150] info | SessionManager.hpp | establis
h_session | session established | client_key: 0x289EAB81, address: 0
[micro_ros_agent-1] [1705496693.287008] info | ProxyClient.cpp | create_p
articipant | participant created | client_key: 0x289EAB81, participant_
id: 0x000(1)
[micro_ros_agent-1] [1705496693.302910] info | ProxyClient.cpp | create_t
opic | topic created | client_key: 0x289EAB81, topic_id: 0x
000(2), participant_id: 0x000(1)
[micro_ros_agent-1] [1705496693.315728] info | ProxyClient.cpp | create_s
ubscriber | subscriber created | client_key: 0x289EAB81, subscriber_i
d: 0x000(4), participant_id: 0x000(1)
[micro_ros_agent-1] [1705496693.330325] info | ProxyClient.cpp | create_d
```

3. Teleop_twist_keyboard: jj


```
(base) brunene@brunene-rog:~$ 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
```

4. Despliegue de tópicos:

```
(base) brunene@brunene-rog:~$ ros2 topic list
/LaserDistance
/ServoAngle
/VelocityEncL
/VelocityEncR
/cmd_vel
/parameter_events
/robot_vel
/rosout
```

5. Valores máximo y mínimo de tópico cmd_vel en la variable “speed”:

Máximo:

Linear: Se estimo la velocidad angular máxima en un
aproximado de entre 0.935336137867581

```
currently: speed 0.6388471674527564 turn 1.0
currently: speed 0.7027318841980321 turn 1.0
currently: speed 0.7730050726178354 turn 1.0
currently: speed 0.850305579879619 turn 1.0
currently: speed 0.935336137867581 turn 1.0
currently: speed 1.0288697516543392 turn 1.0
currently: speed 0.9259827764889054 turn 1.0
```

Angular: Para determinar la velocidad angular maxima, se puso para que el robot girara a la izquierda con la tecla 'j', para que de esta manera no hubiese velocidad lineal, establecimos que la máxima es 8.424634370721177 hasta que ya no hay aumento de velocidad y se mantiene fija.

```
linear:
  x: 0.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 8.424634370721177
```

```
currently: speed 0.9646149645000008 turn 7.736119715997406
currently: speed 0.9646149645000008 turn 6.962507744397666
currently: speed 0.9646149645000008 turn 7.658758518837433
currently: speed 0.9646149645000008 turn 8.424634370721177
```

Mínimo:

Linear:

```
currently: speed 0.05202818496709231 turn 1.0
currently: speed 0.04682536647038308 turn 1.0
currently: speed 0.042142829823344774 turn 1.0
currently: speed 0.0379285468410103 turn 1.0
currently: speed 0.034135692156909266 turn 1.0
currently: speed 0.03072212294121834 turn 1.0
currently: speed 0.027649910647096508 turn 1.0
currently: speed 0.02488491958238686 turn 1.0
currently: speed 0.027373411540625548 turn 1.0
currently: speed 0.030110752694688107 turn 1.0
```

Angular: Para determinar la velocidad angular mínima, se puso para que el robot girara a la izquierda con la tecla 'j', para que de esta manera no hubiese velocidad lineal, establecimos que la mínima es 0.466119397241335253 antes de que el movimiento no se vea constante y/o fluido.


```
currently:      speed 0.9646149645000008      turn 0.7895035858581051
currently:      speed 0.9646149645000008      turn 0.7105532272722946
currently:      speed 0.9646149645000008      turn 0.6394979045450652
currently:      speed 0.9646149645000008      turn 0.5755481140905586
currently:      speed 0.9646149645000008      turn 0.5179933026815028
currently:      speed 0.9646149645000008      turn 0.46619397241335253
```

Video de funcionamiento en el link:

<https://drive.google.com/drive/folders/165HZ3d5rXFkVSevcOc3GTfi9YNB8zOE?usp=sharing>

Conclusiones

Se logró establecer la comunicación entre el Jetson Nano y el PuzzleBot mediante micro-ROS, permitiendo el control remoto a través de ROS 2 y la verificación de velocidades con los encoders. Además, se implementó el paquete teleop twist keyboard para operar el robot externamente.

Si bien los objetivos principales se cumplieron, se presentaron dificultades, como la interrupción ocasional de la conexión SSH, lo que afectó la estabilidad del control. Para mejorar el sistema, se recomienda optimizar la conexión de red y evaluar posibles ajustes en la configuración del Jetson Nano.

Este proyecto permitió comprender la integración de ROS 2 en robótica móvil y sienta las bases para futuras mejoras en control y navegación.

Bibliografía

Deyimar, A. (12 de Febrero de 2025). *¿Cómo funciona el SSH?* Obtenido de Hostinger Tutoriales: https://www.hostinger.com/mx/tutoriales/que-es-ssh#%C2%BFQue_es_SSH

Robotics, M. (s.f.). *Puzzlebot*. Obtenido de Manchester Robotics: <https://manchester-robotics.com/puzzlebot>

ROS. (s.f.). *teleop_twist_keyboard*. Obtenido de ROS.org: https://wiki.ros.org/teleop_twist_keyboard

Zelmer, S. (16 de Octubre de 2024). *Globalising Robotics Education: Introducing Puzzlebot, by Manchester Robotics*. Obtenido de University Of Manchester - Innovation Factory: <https://www.uominnovationfactory.com/projects/robotics-for-everyone-puzzlebot/>