

Trabajo / Examen de Unitree Go2

Alejandro Cruz Aguilar

1. Resumen de la tarea que marcaste como objetivo	2
2. Qué objetivos conseguiste	2
3. Problemas y Soluciones	2
4. Fuentes que consultaste	3
5. Ejemplos de código que diseñaste	4

1. Resumen de la tarea que marcaste como objetivo

El propósito principal de este proyecto consistió en crear una conexión eficaz entre un sistema operativo Ubuntu y el robot Go2 de Unitree, posibilitando la implementación a distancia de programas en Python para regular el movimiento del robot. Con esta conexión, se intentó enviar varios scripts de Python que incluyan instrucciones concretas de movimiento, garantizando así una comunicación constante y la adecuada interpretación de las instrucciones por el robot.

2. Qué objetivos conseguiste

Durante el desarrollo del proyecto, se lograron los siguientes objetivos:

- **Configuración del sistema:** Se estableció la conexión entre Ubuntu y el robot Go2 mediante SSH y red local.
- **Descarga de programas:** Se obtuvieron los paquetes y SDK necesarios desde la página oficial de Unitree y su repositorio en GitHub.
- **Ejecución de ejemplos:** Se probaron scripts de ejemplo proporcionados por Unitree para verificar el correcto funcionamiento del robot.
- **Desarrollo y ejecución de código personalizado:** Se implementaron scripts de Python que permiten controlar el movimiento del robot.
- **Monitoreo en tiempo real:** Se capturaron logs de ejecución para analizar el desempeño del sistema y la respuesta del robot.

3. Problemas y Soluciones

Durante la implementación del proyecto, se identificaron algunos problemas clave y sus respectivas soluciones:

- **Problema con la tarjeta de red:** En el tutorial oficial, la tarjeta de red aparecía individualmente porque el ordenador utilizado tenía dos conexiones Ethernet. Sin embargo, en nuestro caso, solo contábamos con una. Para solucionar este problema, fue necesario conectarnos con la tarjeta de red "enpos3" e instalar una tarjeta WiFi adicional para poder establecer la conexión con el robot.
- **Reinstalación de dependencias tras reiniciar el servidor:** Cada vez que el servidor se apagaba, al volver a encenderlo era necesario reinstalar las dependencias requeridas para ejecutar los programas en el robot. Este problema se resolvió mediante la automatización del proceso de instalación de dependencias en el inicio del sistema.

4. Fuentes que consultaste

Para la implementación del proyecto, se utilizaron las siguientes fuentes de consulta:

- [Guía de inicio rápido de Unitree](#): Proporciona instrucciones para la configuración inicial del robot Go2, incluyendo desempaquetado, actualización de firmware y operaciones básicas.
- [Documentación de Python para Unitree Go2](#): Ofrece ejemplos y guías para el desarrollo de aplicaciones en Python que interactúan con el robot Go2, detallando funciones y métodos disponibles en el SDK.
- [Servicios básicos de Unitree Go2](#): Describe las funcionalidades esenciales del robot, como modos de operación, control de movimiento y manejo de sensores integrados.
- [Servicios deportivos de Unitree Go2](#): Explora aplicaciones orientadas al deporte y entretenimiento, mostrando cómo el Go2 puede participar en actividades físicas y juegos interactivos.
- [Repositorio oficial del SDK de Unitree en GitHub](#): Contiene el código fuente del SDK de Python para el Go2, facilitando la integración y desarrollo de aplicaciones personalizadas.
- [Sección de descargas de Unitree](#): Ofrece firmware, software y manuales necesarios para la correcta operación y actualización del robot Go2.
- [Documentación del Go2 Driver](#): Proporciona detalles técnicos sobre la interfaz de red del Go2, incluyendo instrucciones para la configuración de la conexión, verificación de la red, visualización del estado del robot, teleoperación y activación de modos específicos.

5. Ejemplos de código que diseñaste

En lugar de incluir el código completo en este documento, se ha almacenado en un archivo independiente debido a su extensión. El archivo contiene un script en Python que permite controlar el movimiento del robot Go2 mediante el uso del SDK de Unitree.

El archivo python, está justo en el zip entregado

Este script define una clase Custom que gestiona el envío de comandos de bajo nivel al robot para realizar distintos movimientos predefinidos. Incluye la inicialización de la comunicación, la configuración de los motores y la ejecución de las secuencias de movimiento.

Este trabajo representa un avance en el control remoto de robots, permitiendo futuras mejoras en autonomía e integración con tecnologías avanzadas como visión artificial e inteligencia artificial.