

Control de movimientos en el robot Pepper

Andres Felipe Rojas Y Luis Angel Rivera Guzman

25/04/2025

1 Introducción

En este informe se describen los pasos realizados para programar movimientos sencillos en el robot Pepper, utilizando las librerías correspondientes, la herramienta Choregraphe, y la conexión mediante SSH para la creación manual de un archivo Python en el sistema del robot.

2 Investigación de Librerías

A continuación, se presenta una breve descripción de las librerías utilizadas para el funcionamiento de Pepper:

- **qi**: Permite la comunicación entre el robot y los servicios de control de movimiento, voz, sensores, etc.
- **argparse**: Facilita la lectura de argumentos de línea de comandos para configurar la IP y el puerto de conexión.
- **sys**: Permite interactuar con el sistema operativo y gestionar errores de ejecución.
- **os**: Interactúa con archivos, directorios y variables del sistema operativo.
- **almath**: Proporciona herramientas matemáticas avanzadas para cálculos de movimiento y postura en robots.
- **math**: Proporciona funciones matemáticas estándar para cálculos como ángulos, distancias, etc.
- **motion**: Controla directamente los movimientos físicos de Pepper (cabeza, brazos, piernas, etc.).
- **httplib**: Permite realizar conexiones HTTP para comunicarse con servicios web.
- **json**: Facilita la lectura y escritura de datos estructurados en formato JSON.

3 Instalación y Uso de Choregraphe

- Descargar Choregraphe desde el sitio oficial de SoftBank Robotics.
- Instalar la suite en el equipo siguiendo el asistente de instalación.
- Abrir Choregraphe y conectar a Pepper ingresando la IP en la barra de conexión.
- Crear un proyecto nuevo, añadiendo bloques de movimiento y voz para diseñar una coreografía sencilla.

4 Conexión SSH y Creación del Archivo Python

1. Abrir la terminal en el equipo.
2. Conectarse a Pepper usando:
`ssh nao@<192.168.0.106>`
3. Crear un archivo Python con el siguiente comando:
`nano coreografia_simple.py`
4. Insertar el siguiente código en el archivo:

5 Código de la Coreografía Sencilla

```
import qi
import argparse
import sys
import time

def main(session):
    motion_service = session.service("ALMotion")
    posture_service = session.service("ALRobotPosture")
    tts = session.service("ALTextToSpeech")

    posture_service.goToPosture("StandInit", 0.5)

    motion_service.setAngles("HeadYaw", 0.5, 0.2)
    time.sleep(1)
    motion_service.setAngles("HeadYaw", -0.5, 0.2)
    time.sleep(1)
    motion_service.setAngles("HeadYaw", 0.0, 0.2)

    tts.say("Hola, soy Pepper. Me estoy moviendo!")
```

```

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument("--ip", type=str, default="127.0.0.1",
                        help="192.168.0.106")
    parser.add_argument("--port", type=int, default=9559,
                        help="Puerto NAOqi")
    args = parser.parse_args()

    try:
        connection_url = "tcp://" + args.ip + ":" + str(args.port)
        app = qi.Application(["CoreografiaSimple", "--qi-url=" + connection_url])
    except RuntimeError:
        print("No se pudo conectar a Pepper.")
        sys.exit(1)

    app.start()
    main(app.session)

```

6 Pantallazo de Conexión SSH

A continuación se muestra un pantallazo del ingreso al robot Pepper mediante SSH:

```

C:\Users\ASUS>ssh nao 192.168.0.106
"ssh@nao" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\ASUS>ssh nao@ 192.168.0.106
"ssh@nao" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\ASUS>ssh nao@192.168.0.106
ssh: connect to host 192.168.0.106 port 22: Connection timed out

C:\Users\ASUS>ssh nao@192.168.0.106
The authenticity of host '192.168.0.106 (192.168.0.106)' can't be established.
ED25519 key fingerprint is SHA256:AQvG8R1zozJyyH8epTebVK+YusdHSLanclASnV6U0L4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.106' (ED25519) to the list of known hosts.
(nao@192.168.0.106) Password:
(nao@192.168.0.106) Password:
Pepper [0] ~$ ls
Backend.py      Mateus.py       SDIII           dikemigue.py   presentaciones
Backend.py.save Mateus.py.save  TesisEmociones dikemigue.py.save pro.py
Backend.py.save.1 Mateus.py.save.1 chatbotDeepseek.py escritorio     recordando
Dialogo.py      Movimiento_pepper.py chatbotDeepseek.py.save fotos_pson     server.py
Foto.py         Movimiento_pepper.py.save chatbotDeepseekVmej.py fotos_pson     test
Logos.py        Movimiento_pepper.py.save.1 chatgpt.py       movimientos_LN.py
Mateus..py      Negal.py        diagnosis        my_web         naoqi
Pepper [0] ~$

```

Figure 1: Conexión SSH al robot Pepper

7 Video de la Coreografía en Ejecución

Se puede visualizar el video de la ejecución de la coreografía del robot Pepper en el siguiente enlace:

Ver video de coreografía

8 Explicación de Librerías en el Código

- **qi**: Conexión a los servicios internos de movimiento y voz.
- **argparse**: Configura la IP y el puerto para conectarse al robot.
- **sys**: Permite finalizar la ejecución del programa si hay errores.
- **time**: Controla pausas entre movimientos para que sean naturales.

9 Conclusión

Se logró establecer la conexión con el robot Pepper, diseñar una coreografía básica mediante Choregraphe y también crear un programa sencillo en Python para controlar movimientos de manera manual, comprendiendo el uso de las librerías principales asociadas a su funcionamiento.