

1. Pruebas de abril 30

Prueba 1: Uso de cliente de audio robot G1 individual y con movimiento de brazos con amr_sdk

Objetivo:

Actualizar la sdk de C++ para usar la API de audio que se ofrece en el idioma inglés y realizar pruebas.

Procedimiento:

1. Actualización del SDK 2 de C++

- Se recomienda borrar la SDK2 que se tiene y volver a realizar el proceso de instalación. Ya que no se tiene acceso al comando git pull
 - Se borra la carpeta de unitree_sdk2
 - Se vuelve a clonar el repositorio con

git clone https://github.com/unitreerobotics/unitree_sdk2.git

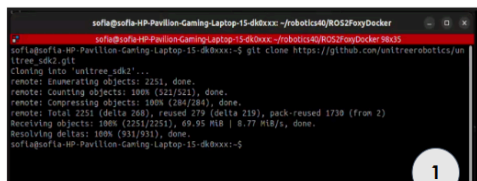
- Se instala la librería con los pasos que ya hemos realizado

Requerimientos:

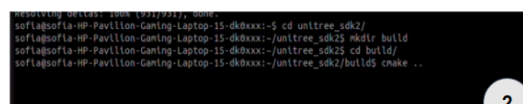
- Ubuntu 20.04

Abrir terminal y ejecutar desde la línea de comandos.

```
>>git clone https://github.com/unitreerobotics/unitree_sdk2.git
>>cd unitree_sdk2
>>mkdir build
>>cd build
>>cmake ..
```

A terminal window showing the execution of 'git clone' and 'cd' commands. The output shows the repository being cloned and the directory being changed. A small circle with the number '1' is at the bottom right of the terminal window.

```
sofia@sofia-HP-Pavilion-Gaming-Laptop-15-dhxxxx: ~/robotics40/ROS2FoxyDocker
sofia@sofia-HP-Pavilion-Gaming-Laptop-15-dhxxxx:~$ git clone https://github.com/unitreerobotics/unitree_sdk2.git
Cloning into 'unitree_sdk2'...
remote: Enumerating objects: 2251, done.
remote: Counting objects: 100% (521/521), done.
remote: Compressing objects: 100% (284/284), done.
remote: Total: 2251 (delta 1043), reused 219 (delta 219), pack-reused 1730 (from 2)
Receiving objects: 100% (2251/2251), 69.05 MiB | 8.77 MiB/s, done.
Resolving deltas: 100% (931/931), done.
sofia@sofia-HP-Pavilion-Gaming-Laptop-15-dhxxxx:~$
```

A terminal window showing the execution of 'mkdir build' and 'cd build' commands. The output shows the directory being created and the directory being changed. A small circle with the number '2' is at the bottom right of the terminal window.

```
sofia@sofia-HP-Pavilion-Gaming-Laptop-15-dhxxxx:~$ cd unitree_sdk2/
sofia@sofia-HP-Pavilion-Gaming-Laptop-15-dhxxxx:~/unitree_sdk2$ mkdir build
sofia@sofia-HP-Pavilion-Gaming-Laptop-15-dhxxxx:~/unitree_sdk2$ cd build/
sofia@sofia-HP-Pavilion-Gaming-Laptop-15-dhxxxx:~/unitree_sdk2/build$
```

- g1_autogermana_demo.cpp -> copiar y pegar en
unitree_sdk2/example/g1/high_level/

- En `unitree_sdk2/example/g1/CMakeLists.txt` ingresar al archivo y poner en el archivo

```
add_executable(g1_audio_client_demo audio/g1_audio_client_demo.cpp)
target_link_libraries(g1_audio_client_demo unitree_sdk2)
```

```
add_executable(g1_autogermana_demo
high_level/g1_autogermana_demo.cpp)
target_link_libraries(g1_autogermana_demo unitree_sdk2)
```

```
add_executable(g1_audio_client_demo audio/g1_audio_client_demo.cpp)
target_link_libraries(g1_audio_client_demo unitree_sdk2)

add_executable(g1_autogermana_demo high_level/g1_autogermana_demo.cpp)
target_link_libraries(g1_autogermana_demo unitree_sdk2)

find_package(yaml-cpp QUIET)
if(yaml-cpp_FOUND)
  if (${yaml-cpp_VERSION} VERSION_GREATER_EQUAL "0.6")
    message(STATUS "Found yaml-cpp version ${yaml-cpp_VERSION}")
    add_executable(g1_dual_arm_example low_level/g1_dual_arm_example.cpp)
    target_link_libraries(g1_dual_arm_example PRIVATE unitree_sdk2 yaml-cpp)
    target_compile_definitions(g1_dual_arm_example PUBLIC BLIB_DIR="${CMAKE_CURRENT_SOURCE_DIR}/low_level/behavior_lib/")
  else()
    message(STATUS "yaml-cpp version ${yaml-cpp_VERSION} is too old, skipping build of g1_dual_arm_example.")
  endif()
endif()
```

- Se debe ir a **unitree_sdk2/build** y realizar la compilación con **cmake ..** y seguidamente **make**
- Debe terminar:

```
[ 75%] Built target g1_ankle_swing_example
[ 78%] Built target g1_dex3_example
[ 81%] Built target g1_audio_client_example
Scanning dependencies of target g1_audio_client_demo
[ 83%] Building CXX object example/g1/CMakeFiles/g1_audio_client_demo.dir/audio/g1_audio_client_demo.cpp.o
[ 84%] Linking CXX executable ../../bin/g1_audio_client_demo
[ 84%] Built target g1_audio_client_demo
Scanning dependencies of target g1_autogermana_demo
[ 86%] Building CXX object example/g1/CMakeFiles/g1_autogermana_demo.dir/high_level/g1_autogermana_demo.cpp.o
[ 87%] Linking CXX executable ../../bin/g1_autogermana_demo
[ 87%] Built target g1_autogermana_demo
[ 90%] Built target go2w_stand_example
[ 93%] Built target go2w_sport_client
[ 96%] Built target b2w_stand_example
[100%] Built target b2w_sport_client
sofia@sofia-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/unitree_sdk2/build$
```

4. Ejecución de códigos

- Para la ejecución de solo validación de solo audio ir a **unitree_sdk2/build/bin** y ejecutar `./g1_audio_client_demo networkInterface`

- Para la ejecución de solo validación de solo audio ir a **unitree_sdk2/build/bin** y ejecutar **./g1_autogermana_demo networkInterface**

5. Corrección de errores de instalación

Solucionando error: No se encuentra asociada la librería la cual está ubicada en usr/local/lib.

```
sofia@sofia-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/unitree_sdk2/build/bin$ ./test_subscriber
./test_subscriber: symbol lookup error: ./test_subscriber: undefined symbol: ddsi_sertype_v0
```

Para visualizar la librería

>>ldd ./test_subscriber

Como se observa, si se encuentra una librería con ubicación diferente tal como la siguiente imagen:

```
sofia@sofia-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/unitree_sdk2/build/bin$ ldd ./test_subscriber
linux-vdso.so.1 (0x00007ffffd5ee000)
libddsc.so.0 => /opt/ros/foxy/lib/x86_64-linux-gnu/libddsc.so.0 (0x000070cfe7c6a000)
libddscxx.so.0 => /home/sofia/unitree_sdk2/thirdparty/lib/x86_64/libddscxx.so.0 (0x000070cfe7a00000)
libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x000070cfe7c19000)
libstdc++.so.6 => /lib/x86_64-linux-gnu/libstdc++.so.6 (0x000070cfe781e000)
libgcc_s.so.1 => /lib/x86_64-linux-gnu/libgcc_s.so.1 (0x000070cfe7803000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x000070cfe7611000)
libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x000070cfe7c11000)
/lib64/ld-linux-x86-64.so.2 (0x000070cfe7ee8000)
libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x000070cfe74c2000)
```

Se deben ejecutar los siguientes comandos:

>>export LD_LIBRARY_PATH=/usr/local/lib:\$LD_LIBRARY_PATH

>>echo 'export LD_LIBRARY_PATH=/usr/local/lib:\$LD_LIBRARY_PATH' >> ~/.bashrc

>>source ~/.bashrc

```
sofia@sofia-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/unitree_sdk2/build/bin$ export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH
sofia@sofia-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/unitree_sdk2/build/bin$ echo 'export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH' >> ~/.bashrc
sofia@sofia-HP-Pavilion-Gaming-Laptop-15-dk0xxx:~/unitree_sdk2/build/bin$ source ~/.bashrc
```

Una vez solventado el error, se puede ejecutar nuevamente y comprobar el funcionamiento adecuado

>>./test_subscriber

Pueden presentarse errores de compilación (make o sudo make install) de librerías como:

fatal error: yaml-cpp/yaml.h: No such file or directory

Para solucionarlo se debe instalar la librería de desarrollo y volver a compilar

```
>> sudo apt update  
>>sudo apt install libyaml-cpp-dev
```

Si se presenta un error del tipo:

Fatal error: eigen3/Eigen/Dense: No such file or directory

Instalar la librería correspondiente con:

```
>>sudo apt update  
>>sudo apt install libeigen3-dev
```

Prueba 2: Movimiento circular o triangular con movimientos articulaciones superiores para bandera

Procedimiento:

1. Conexión y Verificación Inicial:

- Asegúrate de tener conexión Ethernet estable entre el PC y el robot G1.
- Enciende el robot en modo normal (R1 + X).

2. Ejecución

- Se debe contar con el archivo g1_demo_movimiento_bandera.py y ubicarlo en la carpeta deseada
- Ejecutar ubicado en la carpeta deseada

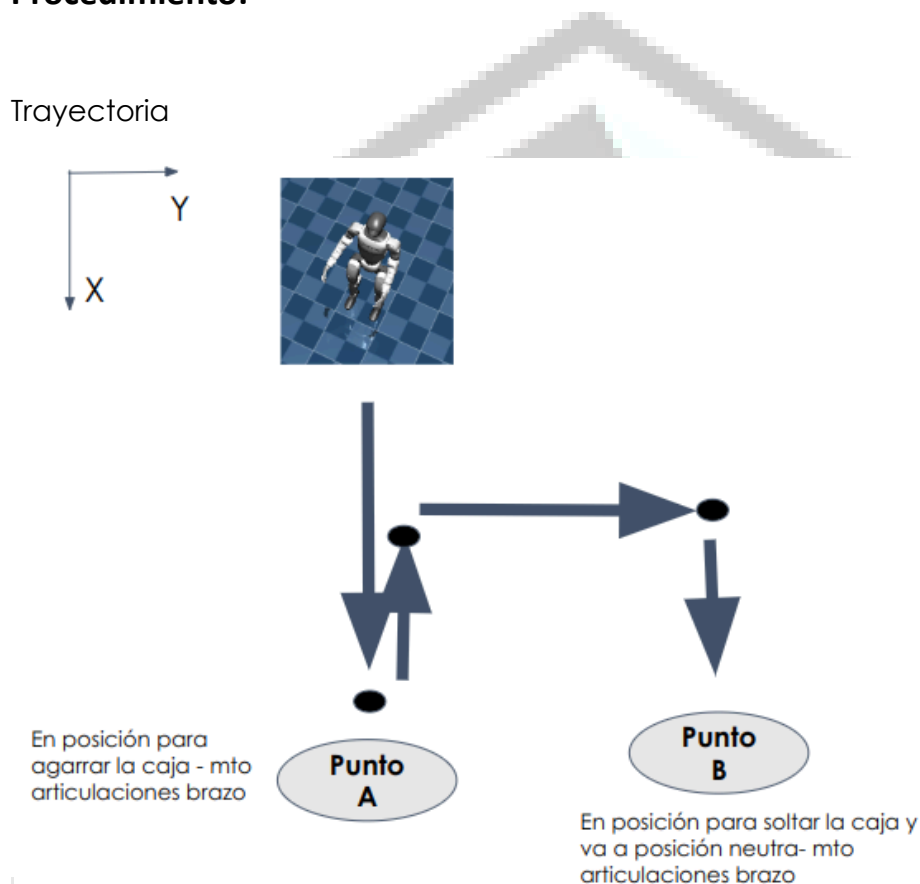
python3 g1_demo_movimiento_bandera.py

- Se escribe en el terminal el nombre de la interfaz deseada conectada al robot.
- Se debe poner en el terminal si se desea ejecutar un círculo con 1 o un triángulo con 2.

Prueba 3: Movimiento para secuencia de agarrar caja

Procedimiento:

Trayectoria



1. Conexión y Verificación Inicial:

- Asegúrate de tener conexión Ethernet estable entre el PC y el robot G1.
- Enciende el robot en modo normal (R1 + X).

2. Ejecución

- Se debe contar con el archivo autogermanaCajaDemo.py y ubicarlo en la carpeta deseada.
- Ejecutar ubicado en la carpeta deseada

python3 autogermanaCajaDemo.py

- Se escribe en el terminal el nombre de la interfaz deseada conectada al robot
- Para ajuste de los valores se debe mover para la trayectoria

```
# Lista de movimientos: (x_vel, y_vel, yaw_vel, duracion)
movimientos = [
    (0.4, 0.0, 0.0, 3.0),      # Paso 1 Avanza hasta antes de la caja
    (0.4, 0.0, 0.0, 1.0),      # Paso 2 Termina de llegar a la caja
    (-0.4, 0.0, 0.0, 2.0),     # Paso 3 Retrocede un poco
    (0.0, 0.0, -0.5, 3.14),    # Paso 4 Gira 90 grados
    (0.4, 0.0, 0.0, 3.0),      # Paso 5 Avanza
    (0.0, 0.0, 0.5, 3.14),     # Paso 6 Gira 90 grados
    (0.4, 0.0, 0.0, 2.0)      # Paso 7 Llega a punto B de la caja
]
```

Donde primer parámetro es velocidad lineal, segundo parámetro es velocidad lateral, tercer parámetro es velocidad de rotación y el último es la duración del movimiento.

- Para ajuste de posiciones de los brazos modificar en, se puede apoyar con mujoco

```
# Paso 1: (nombre, lista de posiciones articulares)
self.pasos = [
    ("Paso 1", [
        GLJointIndex.LeftShoulderPitch: 0.0,
        GLJointIndex.LeftShoulderRoll: 0.35,
        GLJointIndex.LeftShoulderYaw: 0.0,
        GLJointIndex.LeftElbow: 1.5,
        GLJointIndex.LeftWristRoll: 0.0,
        GLJointIndex.LeftWristPitch: 0.0,
        GLJointIndex.LeftWristYaw: 0.0,
        GLJointIndex.RightShoulderPitch: 0.0,
        GLJointIndex.RightShoulderRoll: -0.35,
        GLJointIndex.RightShoulderYaw: 0.0,
        GLJointIndex.RightElbow: 1.5,
        GLJointIndex.RightWristRoll: 0.0,
        GLJointIndex.RightWristPitch: 0.0,
        GLJointIndex.RightWristYaw: 0.0,
        GLJointIndex.WaistYaw: 0.0,
        GLJointIndex.WaistRoll: 0.0,
        GLJointIndex.WaistPitch: 0.0
    ]),
    ("Paso 2", [
        GLJointIndex.LeftShoulderPitch: 0.0,
        GLJointIndex.LeftShoulderRoll: 0.35,
        GLJointIndex.LeftShoulderYaw: 0.4,
        GLJointIndex.LeftElbow: -0.1,
        GLJointIndex.LeftWristRoll: 0.0,
        GLJointIndex.LeftWristPitch: 0.0,
        GLJointIndex.LeftWristYaw: 0.0,
        GLJointIndex.RightShoulderPitch: 0.0,
        GLJointIndex.RightShoulderRoll: -0.35,
        GLJointIndex.RightShoulderYaw: -0.4,
        GLJointIndex.RightElbow: -0.1,
        GLJointIndex.RightWristRoll: 0.0,
        GLJointIndex.RightWristPitch: 0.0,
        GLJointIndex.RightWristYaw: 0.0,
        GLJointIndex.WaistYaw: 0.0,
        GLJointIndex.WaistRoll: 0.0,
        GLJointIndex.WaistPitch: 0.0
    ]),
    ("Paso 3", [
        GLJointIndex.LeftShoulderPitch: 0.0,
        GLJointIndex.LeftShoulderRoll: 0.35,
        GLJointIndex.LeftShoulderYaw: 0.0,
        GLJointIndex.LeftElbow: -0.1,
        GLJointIndex.LeftWristRoll: 0.0,
        GLJointIndex.LeftWristPitch: 0.0,
        GLJointIndex.LeftWristYaw: 0.0,
        GLJointIndex.RightShoulderPitch: 0.0,
        GLJointIndex.RightShoulderRoll: -0.35,
        GLJointIndex.RightShoulderYaw: 0.0,
        GLJointIndex.RightElbow: -0.1,
        GLJointIndex.RightWristRoll: 0.0,
        GLJointIndex.RightWristPitch: 0.0,
        GLJointIndex.RightWristYaw: 0.0,
        GLJointIndex.WaistYaw: 0.0,
        GLJointIndex.WaistRoll: 0.0,
        GLJointIndex.WaistPitch: 0.0
    ])
]
```