

Unitree Go2を使ったハンズオンセミナー



目次

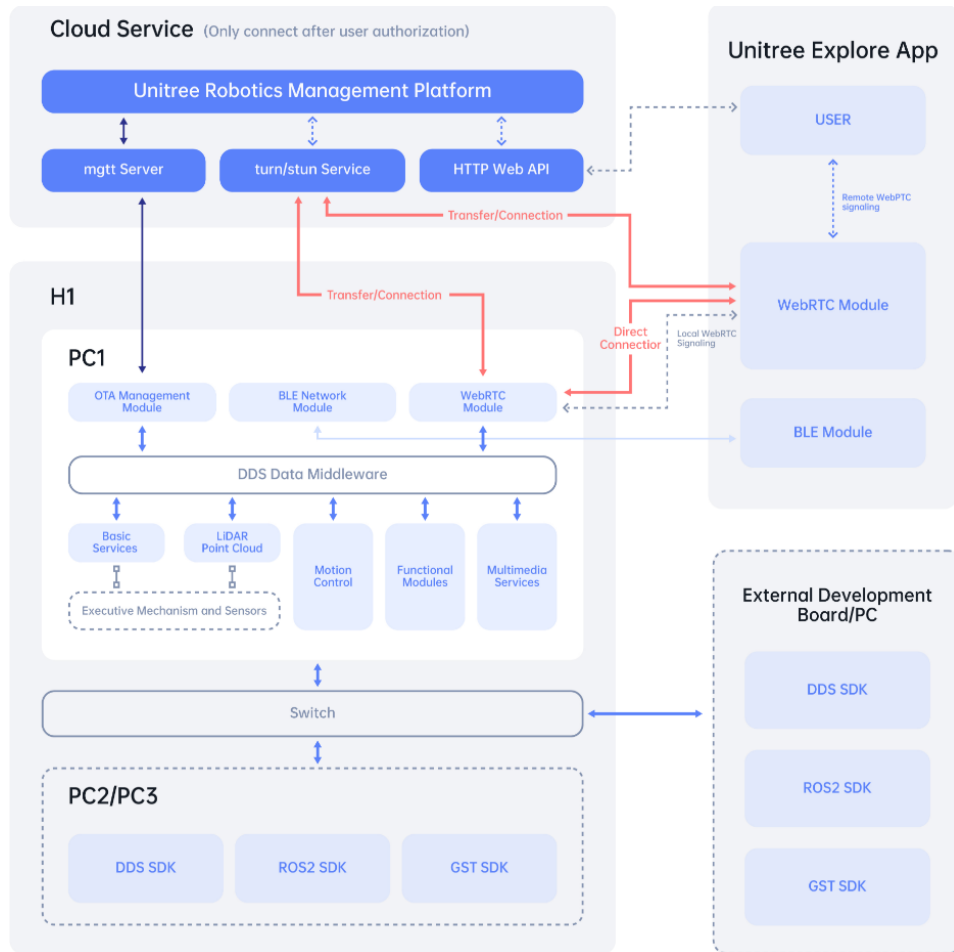
- **開発概要**

- Go2とH1の開発環境の比較
- Go2のDDS Data Middlewareを見てみよう

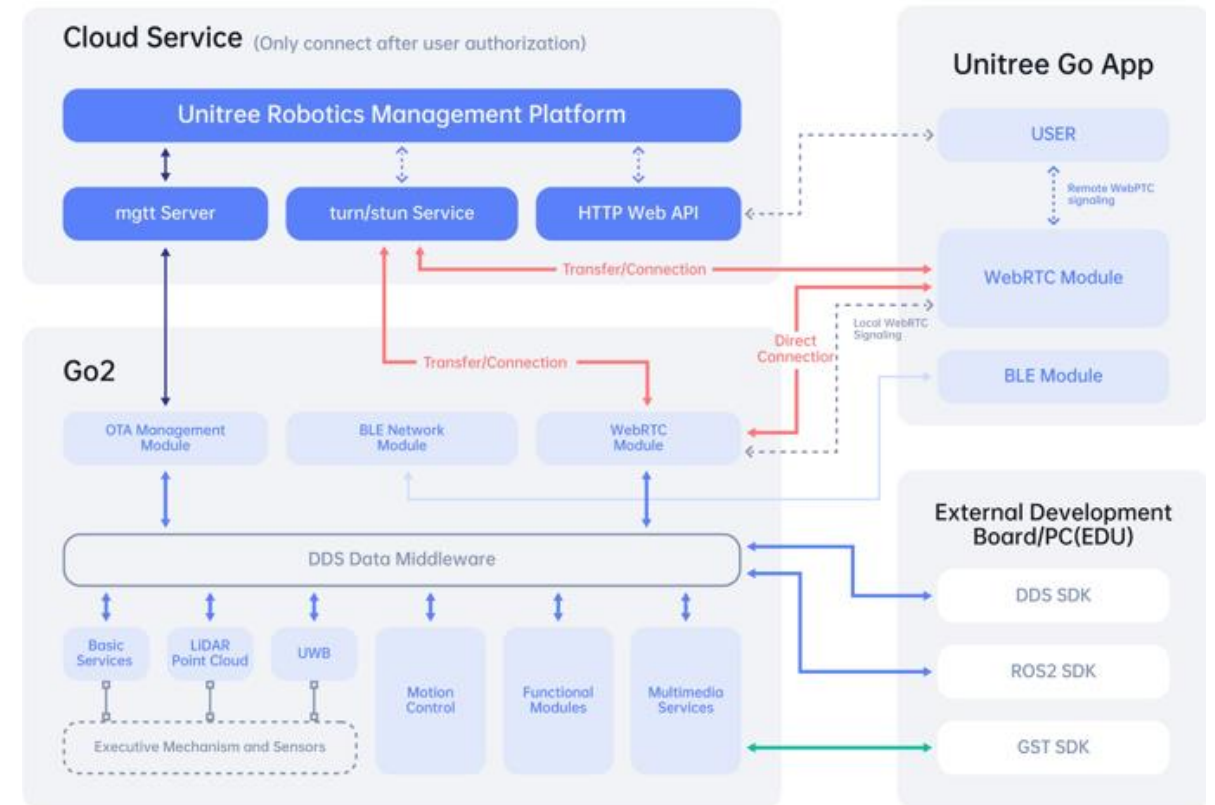
- **ハンズオン**

- 画像の取得
- マーカー検出
- 認識をトリガーに動作

Go2とH1の開発環境の比較



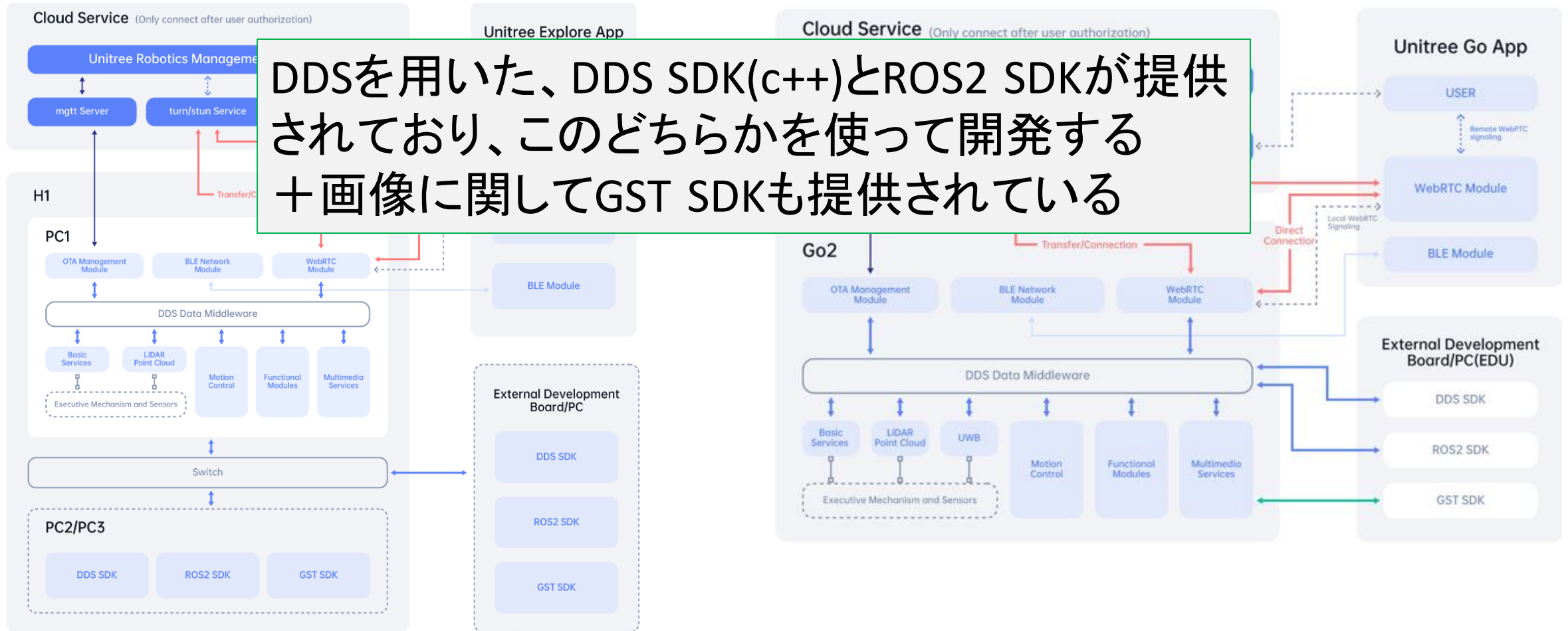
H1



Go2

Go2とH1の開発環境の比較

DDSを用いた、DDS SDK(c++)とROS2 SDKが提供されており、このどちらかを使って開発する
+画像に関してGST SDKも提供されている



H1

Go2

Go2のDDS Data Middlewareを見てみよう (ROS2)

- `$ cd Go2_demo`
- `$ source unitree_ros2/setup.sh`
- `$ ros2 topic list`
- `$ ros2 topic echo /sportmodestate`

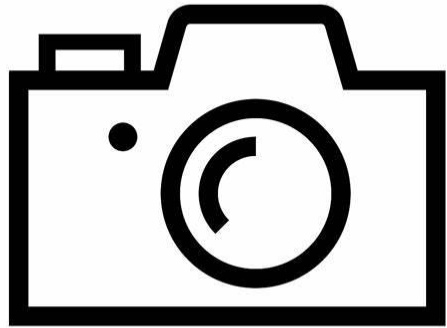
ROSに頼らないDDSトピックも配信されており、unitree_sdk2を用いて使用可能。

ただしunitree_ros2とできることはほとんど同等であり、一般的なユーザーはROS環境を使用するため、割愛する

今回ハンズオンのゴール



構成



①画像認識



②マーカ一検出

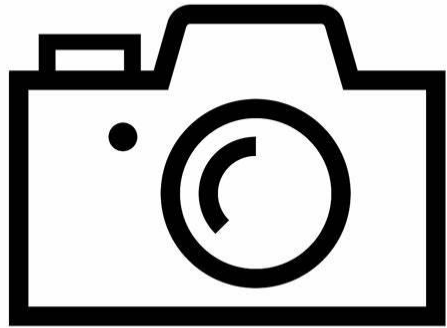


③認識をトリガーに動作

画像の取得

- **内部Docking Stationを使用する場合、Nomachineを使用して接続**
 - Host:192.168.123.18
 - Username:unitree
 - Password:123
- **今回は外部コンピューターを使用する**

①画像認識



①画像認識



②マーカ―検出

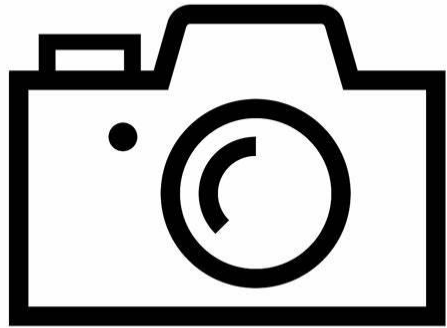


③認識をトリガーに動作

①画像認識

- Gstreamer
- `$ gst-launch-1.0 udpsrc address=230.1.1.1 port=1720 multicast-iface=eth0 ! queue ! application/x-rtp, media=video, encoding-name=H264 ! rtpH264depay ! h264parse ! avdec_h264 ! videoconvert ! Autovideosink`
- `$ cd ~/Go2_demo/1.1`
- `$ bash go2_front_camera_gst.sh`

② マーカー検出



① 画像認識



② マーカー検出



③ 認識をトリガーに動作

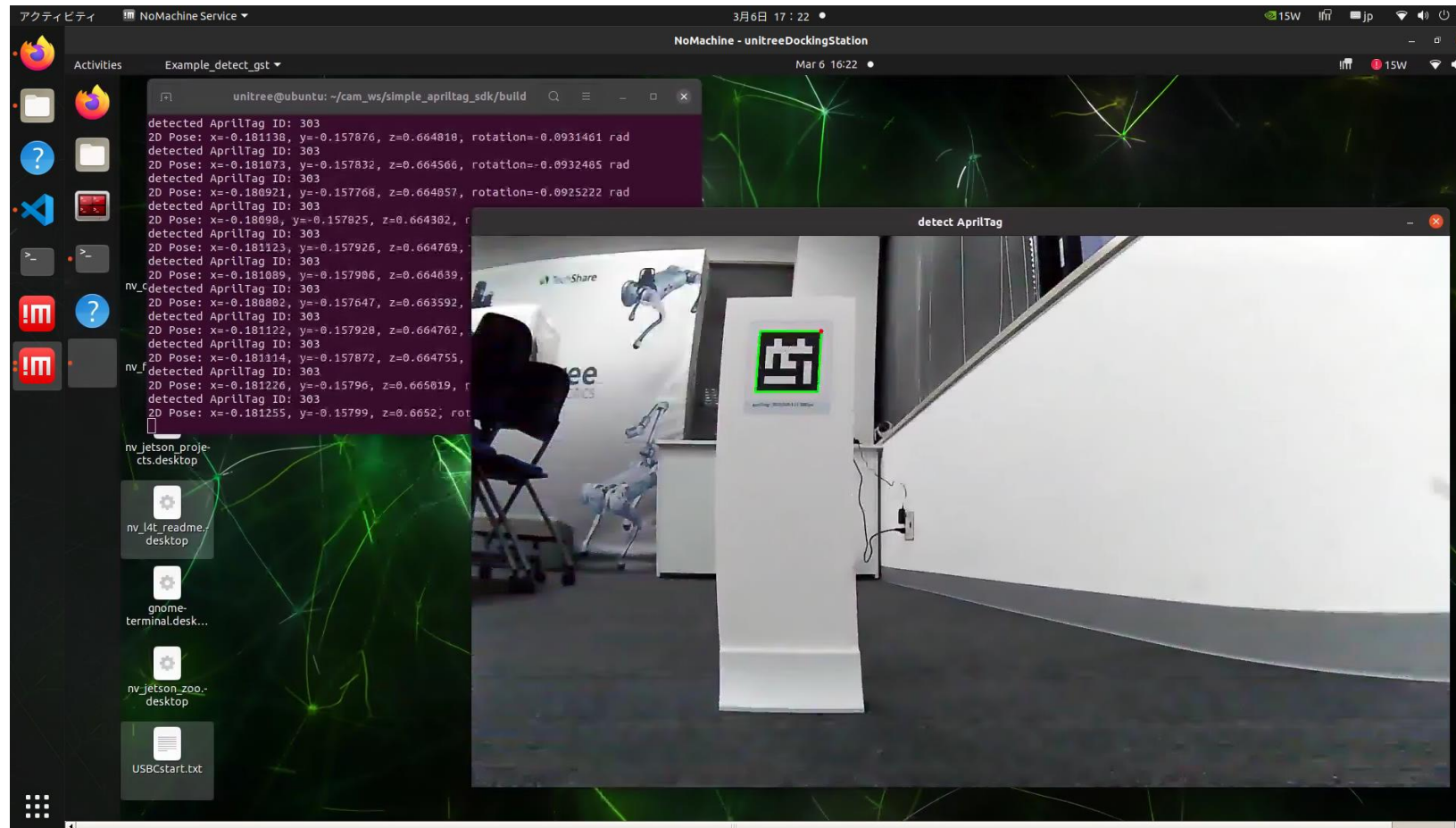
② マーカー検出

- **apriltagを認識して強調**
 - `$ cd ~/Go2_demo/1.2/apriltag_cpp`
 - `$ mkdir build`
 - `$ cd build`
 - `$ cmake ..`
 - `$ make`

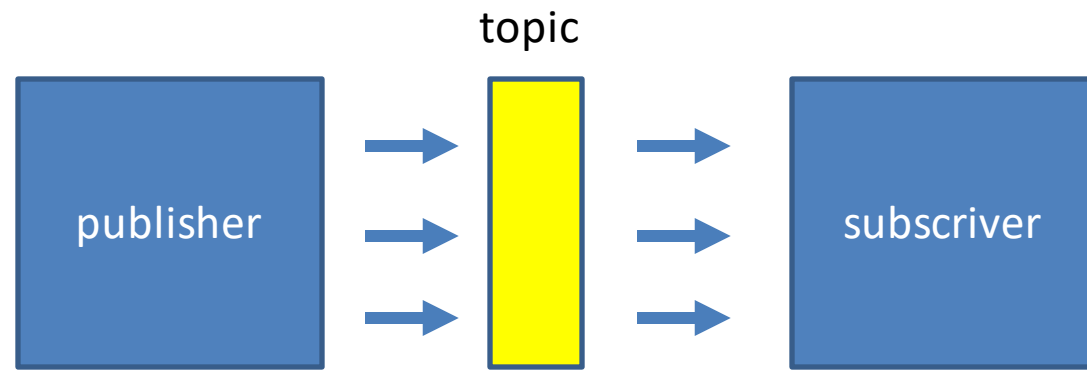
② マーカー検出

- 実行

- `$./example_detect_gst eth0`

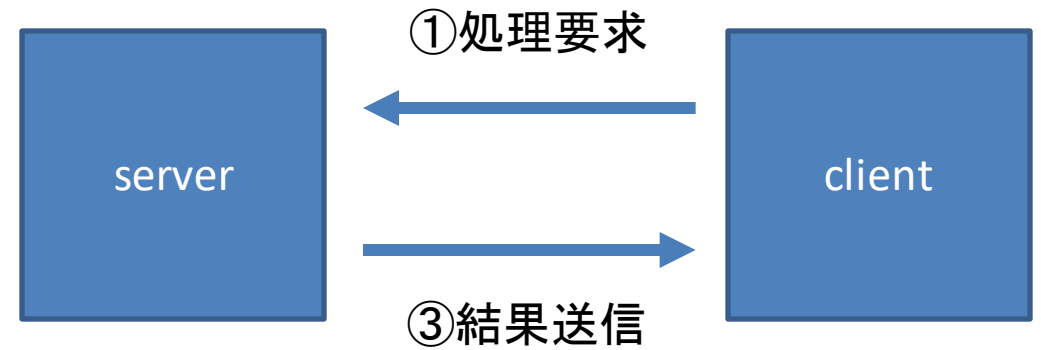


- topic



Topicに常に情報が送られる

- service



②要求受信・処理実行

④結果受信

② マーカー検出

- ターミナルを新しく開く

- Terminal1(server)
 - \$ cd ~/Go2_demo/detect_apriltag_ws
 - \$ source
../unitree_ros2/unitree_ros2_setup.sh
 - \$ source install/setup.bash
 - \$ ros2 run apriltag_service
service_server_gst

- Terminal2(client)
 - \$ cd ~/Go2_demo/detect_apriltag_ws
 - \$ source
../unitree_ros2/unitree_ros2_setup.sh
 - \$ source install/setup.bash
 - \$ ros2 run apriltag_service
service_client

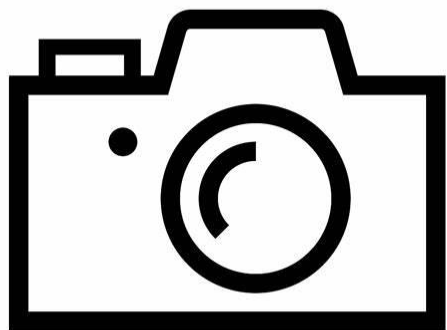

```

Activities Terminal May 16 13:45
unitree@ubuntu: ~/detect_apriltag_ws
terminate called after throwing an instance of 'rclcpp::exceptions::RCLError'
  what(): could not create service: rcl node's context is invalid, at /tmp/binarydeb/ros-foxy-rcl-1.1.14/src/rcl/node.c:441
unitree@ubuntu:~/detect_apriltag_ws$ ros2 run apriltag_service service_server_gst
[ WARN:0] global /home/unitree/Downloads/opencv-4.1.1/modules/videoio/src/cap_gstreamer.cpp (933) open OpenCV | GStreamer warning: Cannot query video position: status=1, value=0, duration=-1
[INFO] [1715835889.843344486] [apriltag_service]: AprilTag Service Ready.
^C[INFO] [1715838097.207403301] [rclcpp]: signal_handler(signal_value=2)
unitree@ubuntu:~/detect_apriltag_ws$ ros2 run apriltag_service service_server_gst
[ WARN:0] global /home/unitree/Downloads/opencv-4.1.1/modules/videoio/src/cap_gstreamer.cpp (933) open OpenCV | GStreamer warning: Cannot query video position: status=1, value=6, duration=-1
[INFO] [1715838212.877800589] [apriltag_service]: AprilTag Service Ready.
[INFO] [1715838212.879539832] [apriltag_service]: Service request received
no marker!
[INFO] [1715838224.497365796] [apriltag_service]: Service request received
ects.desktop
nv_l4t_readme.-desktop

を待っています...
[INFO] [1715838204.449199459] [rclcpp]: サービスが利用可能になるのを待っています...
[INFO] [1715838205.449548107] [rclcpp]: サービスが利用可能になるのを待っています...
[INFO] [1715838206.449927982] [rclcpp]: サービスが利用可能になるのを待っています...
[INFO] [1715838207.450292583] [rclcpp]: サービスが利用可能になるのを待っています...
[INFO] [1715838208.450634742] [rclcpp]: サービスが利用可能になるのを待っています...
[INFO] [1715838209.450987262] [rclcpp]: サービスが利用可能になるのを待っています...
[INFO] [1715838210.451320957] [rclcpp]: サービスが利用可能になるのを待っています...
[INFO] [1715838211.451597968] [rclcpp]: サービスが利用可能になるのを待っています...
[INFO] [1715838212.451918592] [rclcpp]: サービスが利用可能になるのを待っています...
[INFO] [1715838212.904949364] [rclcpp]: 検出失敗
unitree@ubuntu:~/detect_apriltag_ws$ ros2 run apriltag_service service_client
[INFO] [1715838224.519190237] [apriltag_client_simple]: 成功: AprilTag: ID 304 at [x: 0.077417, y: 0.121651, z: 0.468190, rotation: 0.137212]
unitree@ubuntu:~/detect_apriltag_ws$

```


③ 認識をトリガーに動作



① 画像認識



② マーカー検出



③ 認識をトリガーに動作

③ 認識をトリガーに動作

- Terminal1(server)

- \$ cd ~/Go2_demo/detect_apriltag_ws
- \$ source ../unitree_ros2/unitree_ros2_setup.sh
- \$ source install/setup.bash
- \$ ros2 run apriltag_service service_server_gst

- Terminal2(client)

- \$ cd ~/Go2_demo/detect_apriltag_ws
- \$ source ../unitree_ros2/unitree_ros2_setup.sh
- \$ source install/setup.bash
- \$ ros2 run go2_demo tag_move_demo





TechShare