Task 1. ROS 2 Workspace Setup

```
edwin@edwin:~/dev_ws/src/xarm_ros2$ ls
 attach_detach     xarm_api
 demo              xarm_controller
'I HUb .pdf'       xarm_description
 ihub.webm         xarm_gazebo
 LICENSE           xarm_moveit_config
 msg_gazebo        xarm_moveit_servo
 parol6_pipeline   xarm_msgs
 ReadMe_cn.md      xarm_planner
 ReadMe.md         xarm_sdk
 thirdparty        xarm_vision
 uf_ros_lib
```

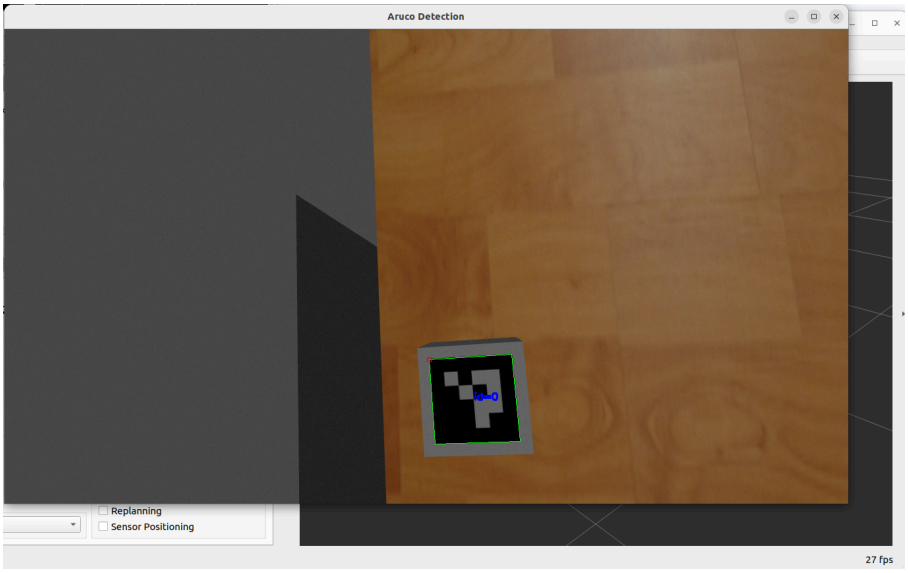
Task 2. Perception Pipeline

Aruco Pose

The `ArucoPoseDetector` node detects a specified ArUco marker in an RGB-D camera stream and publishes its pose in the robot base frame.
It subscribes to `/color/image_raw` for RGB images and `/aligned_depth_to_color/image_raw` for depth data.
The node calculates the marker's center in image coordinates, retrieves the corresponding depth, and converts it to 3D position in the camera frame.
Rotation offsets and a configurable Z-offset are applied to adjust the marker pose relative to the robot.
The pose is transformed from the camera frame to the base frame using a TF2 buffer and lookup.
A `PoseStamped` message is published on `/detected_box_pose` for use in downstream tasks like pick-and-place.
The node also broadcasts a TF frame `detected_aruco_marker` for visualization in RViz.
Debugging, warnings, and informative logs help ensure correct marker detection, depth validity, and TF transformations.

Pick Controller

The `PickController` node subscribes to `/detected_box_pose` and waits for a trigger on the `/start_picking` service to initiate a pick operation.
It stores the latest detected box pose and throttles logs for pose updates.
The node initializes a MoveIt `MoveGroupInterface` for the `lite6` planning group, setting planning time, number of attempts, and reference frame.
When triggered, it sets the end-effector target pose to the raw ArUco-detected coordinates without modifying orientation or Z-offset.
The node plans a path to the target pose and executes it using MoveIt, checking for planning success.



Aruco Detection

```
edwin@edwin:~/dev_ws/src/xarm_ros2$ ros2 topic echo /detected_box_pose
header:
  stamp:
    sec: 618
    nanosec: 27000000
  frame_id: link_base
pose:
  position:
    x: 0.24267831056380057
    y: 0.14282618110404924
    z: 0.08236940770852247
  orientation:
    x: 0.9999999926749198
    y: -4.5889779510004e-07
    z: -0.00011988474310765182
    w: -1.6661272303257158e-05
---
header:
  stamp:
    sec: 618
    nanosec: 294000000
  frame_id: link_base
pose:
  position:
    x: 0.24267831056380118
    y: 0.142826181104049
    z: 0.08236940770852252
  orientation:
    x: 0.9999999926749198
    y: -4.5889779510004e-07
    z: -0.00011988474310768668
    w: -1.6661272303021615e-05
---
```

Pose Detection data

Task 4 Pick-and-Place Application Node

For Pick and place operation use attach detach pkg and use command

```
Attach an Object

Attach a box to the robot’s end-effector:

  ros2 service call /AttachDetach msg_gazebo/srv/AttachDetach "{model1:
'UF_ROBOT', link1: 'link6', model2: 'aruco_box', link2: 'link_0', attach:
                                  true}"


Detach an Object

Detach the box from the end-effector:

  ros2 service call /AttachDetach msg_gazebo/srv/AttachDetach "{model1:
'UF_ROBOT', link1: 'link6', model2: 'aruco_box', link2: 'link_0', attach:
                                 false}"
```