

Jawaban 3d

Oleh Muhammad Akmal
NIM 19624235 | STEI-K

KONTROL DYNAMIXEL AX-18A

Untuk mengontrol **Dynamixel AX-18A** menggunakan sistem **publish-subscribe** di Raspberry Pi, perlu adanya komunikasi serial dengan protokol khusus yang mendukung Dynamixel, serta library seperti **ROS (Robot Operating System)** untuk implementasi model publish-subscribe. Berikut ini adalah langkah-langkah rinci cara mengontrol dan berkomunikasi dengan servo **Dynamixel AX-18A** menggunakan **Raspberry Pi** dan **U2D2** sebagai konverter USB-to-Dynamixel.

KOMPONEN YANG DIBUTUHKAN:

- **Raspberry Pi** (untuk menjalankan ROS dan mengontrol servo)
- **Dynamixel AX-18A** (servo motor)
- **U2D2** (USB ke TTL converter untuk komunikasi Dynamixel)
- **Power supply** (biasanya 12V untuk Dynamixel)
- **Kabel konektor** (untuk menghubungkan U2D2 dengan servo Dynamixel)
- **ROS (Robot Operating System)** di Raspberry Pi
- **Dynamixel SDK** atau **dynamixel_workbench** untuk mengontrol servo melalui ROS
- **Breadboard dan kabel jumper** (jika diperlukan untuk komponen tambahan)

1 INSTALASI SISTEM OPERASI DAN ROS DI RASPBERRY PI

- a. Pastikan Raspberry Pi telah terinstall **Raspberry Pi OS** dan sudah memiliki koneksi ke internet.
- b. Instal ROS (misalnya **ROS Noetic**) di Raspberry Pi. Berikut adalah perintah dasar untuk instalasi ROS:

```
sudo apt update
sudo apt install ros-noetic-desktop-full
```

- c. Setelah instalasi selesai, setup lingkungan ROS:

```
echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

2 INSTALASI DYNAMIXEL SDK DAN DYNAMIXEL_WORKBENCH

Dynamixel SDK adalah software development kit yang disediakan oleh Robotis untuk berkomunikasi dengan servo Dynamixel.

Instal SDK untuk Dynamixel dan paket **dynamixel_workbench** mempermudah penggunaan Dynamixel dengan ROS.

```
sudo apt install ros-noetic-dynamixel-sdk
sudo apt install ros-noetic-dynamixel-workbench
```

3 SETUP HARDWARE: MENGHUBUNGKAN RASPBERRY PI DENGAN DYNAMIXEL

- Hubungkan **U2D2** ke Raspberry Pi melalui port **USB**.
- Sambungkan **Dynamixel AX-18A** ke **U2D2** dengan konektor TTL.
- Sambungkan power supply 12V ke Dynamixel untuk memberikan daya yang cukup.
- Pastikan kabel data dan daya sudah terpasang dengan benar.

4 MENGONFIGURASI PARAMETER DYNAMIXEL

Setiap servo Dynamixel memiliki **ID** unik. Pastikan bahwa ID untuk Dynamixel AX-18A telah diatur dengan benar. Anda bisa menggunakan Dynamixel Wizard (GUI Tool dari Robotis) untuk mengubah parameter jika diperlukan.

- Di terminal, jalankan perintah berikut untuk memeriksa konektivitas servo:

```
roslaunch dynamixel_workbench_controllers
dynamixel_controllers.launch
```

- Periksa apakah servo terdeteksi dan pastikan ID dan Baudrate sudah benar.

5 IMPLEMENTASI MODEL PUBLISH-SUBSCRIBE DI ROS

Dalam ROS, kontrol servo dapat dilakukan menggunakan model **publish-subscribe**, di mana Raspberry Pi akan bertindak sebagai **publisher** yang mengirimkan perintah ke topik tertentu, dan servo akan bertindak sebagai **subscriber** yang menerima perintah tersebut.

- Subscriber (Servo Dynamixel):** Dynamixel akan berlangganan (subscribe) ke topik yang berisi perintah pergerakan (misalnya posisi, kecepatan, atau torsi).
- Publisher (Raspberry Pi):** Raspberry Pi akan menerbitkan (publish) data ke topik tertentu untuk mengontrol servo.

Langkah-langkah:

- Membuat Node Publisher di Raspberry Pi**

- Buat node **publisher** yang akan mengirimkan perintah ke Dynamixel. Sebagai contoh, kita bisa mengontrol posisi servo menggunakan pesan ROS.
- Buat file **Python** atau **C++** yang menerbitkan perintah posisi servo, misalnya:

```
# Kontrol servo dengan menerbitkan data ke topik /dynamixel/command
import rospy
from std_msgs.msg import Int32

def publisher():
    rospy.init_node('servo_controller', anonymous=True)
    pub = rospy.Publisher('/dynamixel/command', Int32,
        queue_size=10)
    rate = rospy.Rate(10) # 10 Hz

    while not rospy.is_shutdown():
        position = 512 # Nilai posisi servo tengah (0 - 1023)
        rospy.loginfo(position)
        pub.publish(position)
        rate.sleep()

if __name__ == '__main__':
    try:
        publisher()
```

```
except rospy.ROSInterruptException:
    pass
```

- b. **Membuat Node Subscriber di Dynamixel:** Node ini sudah disediakan dalam paket **dynamixel_workbench** yang akan secara otomatis mengontrol servo berdasarkan data yang diterima di topik **/dynamixel/command**.

- c. **Menjalankan Kontrol Servo:**

- i. Jalankan perintah berikut untuk meluncurkan kontrol servo:

```
roslaunch dynamixel_workbench_controllers
dynamixel_controllers.launch
```

- ii. Di terminal lain, jalankan node publisher yang telah dibuat:

```
roslaunch <package_name> <script_name>.py
```

6 MENGGUNAKAN U2D2 UNTUK KOMUNIKASI TTL

U2D2 bertindak sebagai konverter antara Raspberry Pi (USB) dan servo Dynamixel (TTL). Pastikan Raspberry Pi mengenali perangkat **U2D2** sebagai port serial, biasanya **/dev/ttyUSB0**.

Pastikan baudrate diatur sesuai dengan baudrate default dari servo AX-18A, yaitu 57600 atau yang sudah dikonfigurasi sebelumnya.

- a. Untuk memastikan bahwa **U2D2** berfungsi dengan baik, cek port serial menggunakan perintah:

```
ls /dev/ttyUSB*
```

- b. Jika **/dev/ttyUSB0** muncul, maka **U2D2** sudah terhubung dengan benar dan siap digunakan untuk mengirimkan perintah ke servo.

7 MONITORING STATUS SERVO DAN FEEDBACK

Untuk memonitor status servo (seperti torsi, suhu, dan posisi) gunakan **dynamixel_workbench**. Hal ini sangat penting untuk memastikan servo bekerja sesuai yang diharapkan dan mencegah overheating atau kegagalan mekanis.

Gunakan perintah berikut untuk memonitor status:

```
rostopic echo /dynamixel/status
```

Menggunakan sistem **publish-subscribe** di **ROS** memungkinkan Raspberry Pi untuk mengontrol servo Dynamixel AX-18A secara fleksibel dan modular. Dengan bantuan konverter **U2D2**, Raspberry Pi dapat berkomunikasi dengan servo menggunakan protokol TTL, memungkinkan pengontrolan posisi, kecepatan, dan torsi secara real-time.