

Untuk mengontrol dan berkomunikasi dengan **Dynamixel AX-18A** servo motor menggunakan sistem **publish-subscribe** pada **Raspberry Pi**, kita bisa menggunakan framework **ROS (Robot Operating System)**. Langkah-langkah di bawah ini mencakup pengaturan komponen, komunikasi menggunakan modul U2D2, serta implementasi sistem kontrol berbasis ROS dengan pendekatan publish-subscribe.

Komponen yang Dibutuhkan:

1. **Dynamixel AX-18A servo motor:** Servo motor yang akan dikendalikan.
2. **Raspberry Pi:** Komputer kecil yang menjalankan ROS.
3. **U2D2:** Modul komunikasi untuk menghubungkan servo motor Dynamixel ke Raspberry Pi melalui port USB.
4. **Dynamixel SDK:** Software Development Kit (SDK) dari Robotis untuk berkomunikasi dengan servo.
5. **Power supply:** Sumber daya untuk Dynamixel.
6. **ROS (Robot Operating System)**
7. **USB to Micro USB Cable:** Untuk menghubungkan U2D2 dengan Raspberry Pi.

Tahapan Pengontrolan Dynamixel AX-18A dengan Sistem Publish-Subscribe:

- **Set Up Raspberry Pi dan Install ROS**
- **Install Dynamixel SDK dan Dynamixel Workbench**
- **Menghubungkan Dynamixel AX-18A dengan Raspberry Pi**
 - o Sambungkan **Dynamixel AX-18A** ke modul **U2D2** melalui kabel komunikasi **TTL**.
 - o Hubungkan modul **U2D2** ke Raspberry Pi menggunakan kabel **USB**.
 - o Pastikan **power supply** yang sesuai (biasanya 12V) terhubung ke Dynamixel AX-18A.
- **Set Up Communication dan Verifikasi Servo**
 - o Uji koneksi antara Raspberry Pi dan Dynamixel AX-18A dengan menjalankan perintah berikut untuk memeriksa apakah Raspberry Pi terhubung ke servo:

- `roslaunch dynamixel_workbench_controllers dynamixel_controllers.launch`
 - Periksa koneksi pada terminal, pastikan motor AX-18A dikenali. Kamu juga bisa menggunakan GUI Dynamixel untuk memverifikasi komunikasi.
- **Sistem Publish-Subscribe dalam ROS:** Pada sistem publish-subscribe, kita akan membuat node yang mem-publish perintah gerakan ke Dynamixel AX-18A. Servo AX-18A akan bertindak sebagai subscriber yang menerima perintah untuk mengubah posisinya.
 - **Publish node:** Buat node yang bertanggung jawab untuk mengirim perintah (misalnya, posisi servo atau kecepatan) ke AX-18A. Kamu bisa menggunakan **Python** atau **C++**.
 - **Subscribe node:** Node ini bertugas menerima perintah dari publisher dan menggerakkan Dynamixel AX-18A sesuai dengan data yang diterima.
- **Menjalankan Node**
 - Jalankan **Publisher** node untuk mengirim perintah:
 - `roslaunch your_package servo_publisher.py`
 - Jalankan **Subscriber** node untuk menerima dan menjalankan perintah pada servo:
 - `roslaunch your_package servo_subscriber.py`
- **Testing dan Debugging:** Pastikan bahwa publisher mengirimkan perintah posisi ke subscriber, dan servo AX-18A bergerak sesuai dengan perintah yang diterima. Jika terjadi masalah, periksa koneksi fisik dan komunikasi melalui port USB (U2D2) dan baud rate.