Dynamixel AX-18A merupakan servo yang memiliki torsi yang cukup kuat dan sangat penting untuk pergerakan yang membawa bobot berat. Jelaskan secara bertahap cara kontrol dan komunikasi servo tersebut menggunakan sistem publish subscribe pada Raspberry Pi! (perkirakan juga penggunaan komponen modul lainnya, U2D2 misalnya)

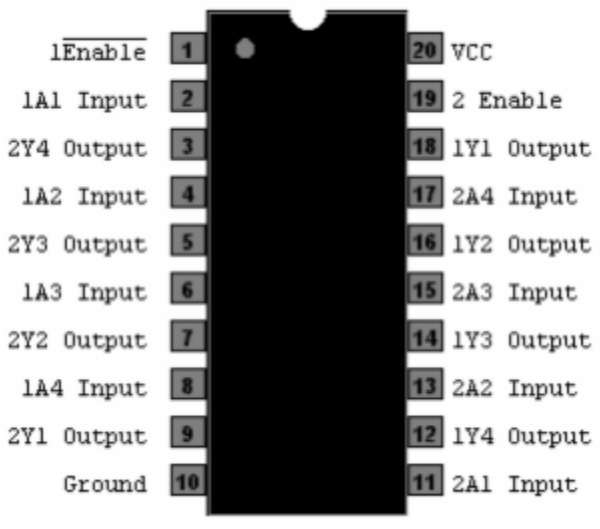
Nama: Dietrich Pepalem Tarigan

NIM: 10223037

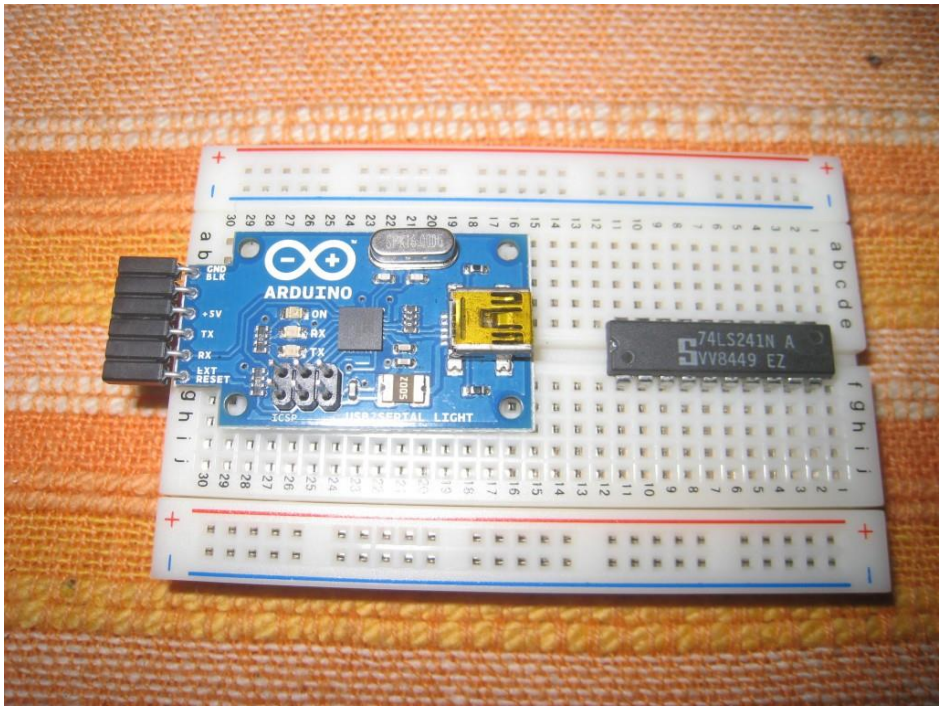
Dynamixel servomotors are like Ferrari in the world of the cars : the best servomotors on the market. Some characteristics of these servos:

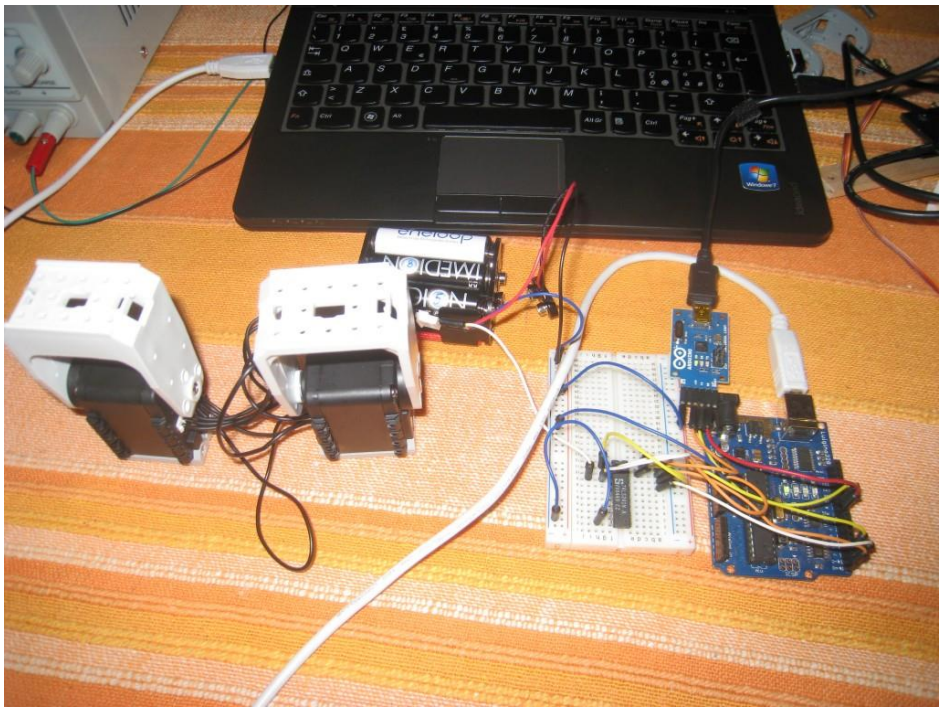
- Operational Torque: 144 oz-in (10.4 kg.cm)
- 1 / 2 duplex multi-drop serial bus
- 1M bps serial communication
- Reports position, speed, load, voltage, and temperature
- Full rotation mode
- 300 ° angular position in 1024 increments
- Speed and torque control in 1024 increments
- Built in LED status indicator / alarm
- Shutdown on max / min voltage, load or temperature
- Single cable network connection

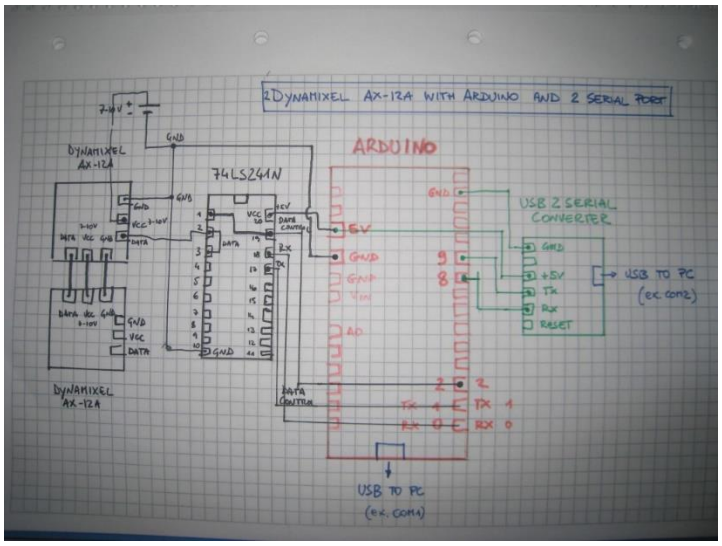
Servomotors ini mengalami kesulitan untuk terhubung ke Arduino. Faktanya, komunikasi setengah dupleks ke 1Mbps membutuhkan sirkuit tambahan untuk membuat koneksi ke Arduino jika ada beberapa servo yang akan dihubungkan. Satu servo dapat dihubungkan langsung ke Arduino, dalam kasus beberapa aktuator, perlu menggunakan penyangga tri-state, yang ditempatkan di antara Arduino dan AX-12A. Penyangga tri-state sederhana adalah 74LS241N.



Protokol Dynamixel adalah protokol serial, jadi, di sisi Arduino, buffer 74LS241 harus terhubung ke port serial dan kemudian pada pin 0 dan 1. Ini adalah masalah: jika port serial digunakan oleh Dynamixel, bagaimana cara menampilkan informasi melalui monitor Serial Arduino IDE? Kita membutuhkan port serial lain. Library Arduino NewSoftSerial dapat membantu kita, tetapi kita juga membutuhkan konverter USB-serial. Author memilih konverter USB2Serial yang dibuat oleh tim Arduino. Ini adalah gambar semua komponen yang diperlukan untuk menggunakan Servo Dynamixel.


ARDUINO
74LS241N A
VV8449 EZ





Untuk mengontrol dan berkomunikasi dengan servo **Dynamixel AX-18A** menggunakan sistem **publish-subscribe** pada **Raspberry Pi**, berikut adalah langkah-langkah yang bisa diikuti:

1. Persiapan Hardware

Komponen yang diperlukan:

- Raspberry Pi (dapat menggunakan model apa pun)
- Servo Dynamixel AX-18A
- Modul U2D2 (untuk konversi komunikasi USB ke TTL)
- Beberapa kabel jumper
- Power supply yang sesuai untuk servo (9-12 Volt)

Koneksi:

- Hubungkan **U2D2** ke Raspberry Pi menggunakan USB.
- Sambungkan servo AX-18A ke U2D2 menggunakan kabel komunikasi.
- Pastikan power supply terhubung dengan benar ke servo (power positif ke V+ dan ground ke GND).

2. Konfigurasi Raspberry Pi

1. **Aktifkan UART:**
    - Edit file /boot/config.txt untuk mengatur init_uart_clock=16000000.
    - Gunakan perintah sudo stty -F /dev/ttyAMA0 1000000 untuk mengatur baud rate.
2. **Nonaktifkan getty pada UART:**

- Edit file /boot/cmdline.txt dan hapus semua opsi yang merujuk pada ttyAMA0.
- Edit /etc/inittab dan komentar pada semua baris yang berkaitan dengan ttyAMA0.

3. **Reboot Raspberry Pi** agar perubahan diterapkan.

3. Implementasi Sistem Publish-Subscribe

Sistem publish-subscribe dapat diimplementasikan menggunakan **MQTT** atau protokol sejenis. Berikut langkah-langkah dasar untuk melakukan ini:

1. **Instalasi Broker MQTT:**
   - Gunakan broker MQTT seperti Mosquitto. Instalasi dapat dilakukan dengan perintah:

     bash

     Copy code

     sudo apt-get install mosquitto mosquitto-clients

2. **Instalasi Pustaka MQTT di Python:**
   - Instal pustaka Paho MQTT untuk Python:

     bash

     Copy code

     pip install paho-mqtt

3. **Pengkodean Program:**
   - Buat program Python yang berfungsi sebagai publisher untuk mengirim perintah ke servo, dan subscriber untuk menerima umpan balik dari servo. Contoh:

     python

     Copy code

     import paho.mqtt.client as mqtt

     def on_connect(client, userdata, flags, rc):

     * print("Connected with result code " + str(rc))*

     def on_message(client, userdata, msg):

     * print(msg.topic + " " + str(msg.payload))*

```
client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message

client.connect("broker.hivemq.com", 1883, 60)

client.subscribe("dynamixel/control")
client.loop_start()

# Untuk mengirim perintah
client.publish("dynamixel/control", "command_to_move")
```

4. Kontrol Servo Dynamixel

Setelah sistem publish-subscribe diatur, Kita bisa menggunakan pustaka seperti **Dynamixel SDK** untuk mengirim perintah ke servo. Berikut adalah langkah-langkah untuk mengontrol servo:

1. **Instalasi Dynamixel SDK:**
    - Ikuti petunjuk dari repo GitHub Dynamixel.
2. **Pengkodean Kontrol:**
    - Buat script yang menggunakan SDK untuk mengontrol servo berdasarkan perintah yang diterima dari broker MQTT. Kita dapat menggunakan perintah seperti setGoalPosition untuk menggerakkan servo ke posisi yang diinginkan.
3. **Uji Sistem:**
    - Jalankan program dan uji pergerakan servo dengan mengirim perintah melalui topik MQTT.

REFRENSI:

1. Dynamixel AX-12A and Arduino: how to use the Serial Port « Robottini
2. Oppedijk/DynamixelPi: Control Dynamixel from Raspberry Pi (github.com)