



Dokumentation Robot Controller Windows

Henrik Wahlström

January 17, 2023

1 Intro

Guiden är skriven och testad 2022-01-24 av dåvarande IT-amanuens Henrik Linder, med Windows Version 10.0.19042 Build 19042, Python 3.10.2, Firefox version 96.0.2.

Finns inga som helst garantier att den kommer vara kompatibel med andra versioner eller andra webbläsare.

Genom guiden kommer jag använda `monospace text skriven så här` för att indikera kommandon som skrivs in i terminalen.

Tveka inte att höra av dig till mig (Henrik) på it@tekniskfysik.se vid frågor.

Enormt tack till Fredrik Peteri Harr som har utvecklat koden och gjort den tillgänglig för oss!

2 Installation

Ladda ner GitHubrepot från <https://github.com/frepet/Robot-Arm-Controller> genom att klicka på den gröna knappen "Code" och sen "download zip". (Om du använt github tidigare går det självklart bra att göra på annat sätt)

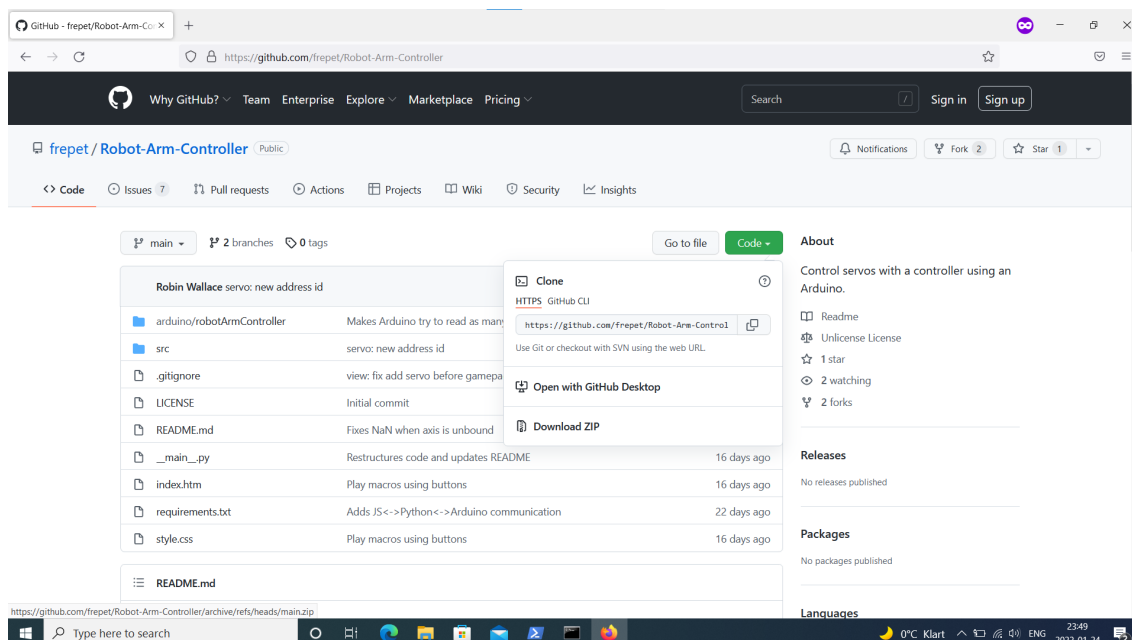


Figure 1: Hur det ser ut att ladda ner repot från github.

Extrahera mappen till valfri plats (i resten av guiden kommer jag utgå ifrån att den ligger i mappen "Documents").



Öppna en terminal (CMD eller powershell)

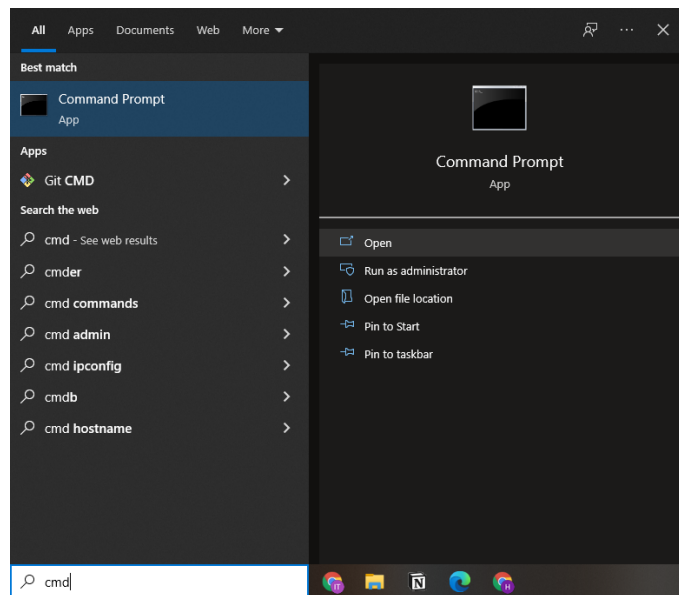


Figure 2: Öppna CMD (powershell bör funka lika bra)

Skriv `python`. Om python är installerat så ser det ut så här:

```
C:\Users\AMTF\Documents\Robot-Arm-Controller-main\Robot-Arm-Controller-main>python
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Figure 3: Hur det ser ut efter kommandot `python` om python är installerat.

isåfall skriv `exit()` för att gå tillbaka ut till terminalen.

Om python inte är installerat så kommer du tas till installationssidan för python i Microsoft Store: installera då python därifrån och skriv sedan `python` igen i terminalen för att se att det är installerat (`exit()` för att avsluta).

Gå till den extraherade mappen i filhanteraren, klicka dig in tills du ser filen "index.htm", och kopiera path dit. I mitt fall blir den

`C:\Users\AMTF\Documents\Robot-Arm-Controller-main\Robot-Arm-Controller-main.`

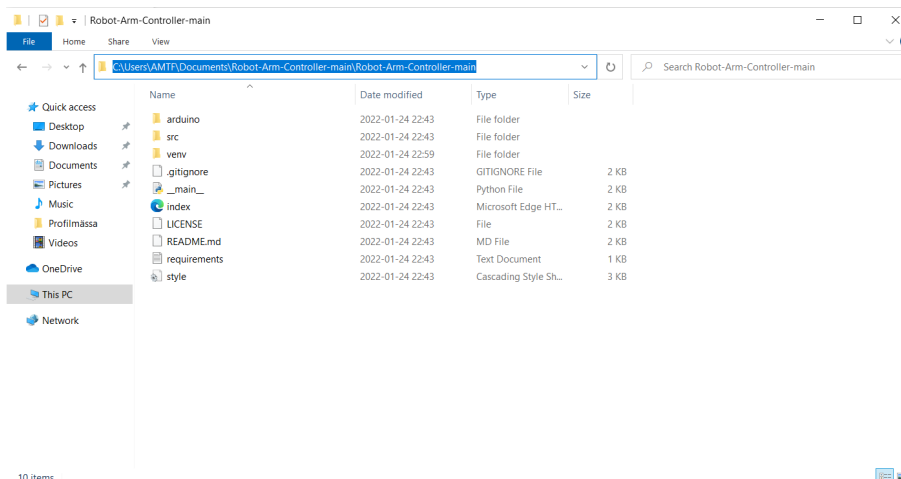


Figure 4: Mappen som du ska kopiera path till (path markerad blå).

Gå till terminalen, skriv `cd {PATH}`, alltså i mitt fall

`cd C:\Users\AMTF\Documents\Robot-Arm-Controller-main\Robot-Arm-Controller-main`

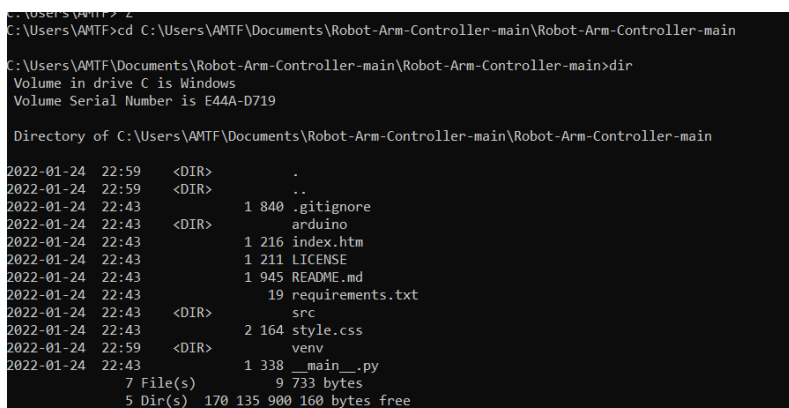


Figure 5: Hur det ser ut att byta till robotarmsmappen med `cd` och sen visa innehållet med `dir`.

Om du skriver `dir` nu så borde du se samma filer som i mappen du laddade ner.

Nu när python är installerat så kan du följa instruktionerna i example setup på github (kan även läsas längst ner i det här dokumentet eller i filen "README.md" som finns i githubrepot du laddade ner).

Steg 1 och 2 bör fungera utan problem, annars fråga IT-amanuens eller felsök själv.

Steg 3 kan ge error om att du inte får köra scripts eller liknande, isåfall kan du behöva skriva `set-executionpolicy RemoteSigned -Scope CurrentUser` för att tillåta PowerShell att köra scripts.

När du följt alla 5 steg, **se till att öppna index.htm med Firefox** då de andra webbläsarna kanske inte fungerar med handkontrollen. Verkar ha något att göra med hur webbläsarna hanterar JS-modulen "gamepad" som används för att läsa data från kontrollen. Det kan ha löst sig sedan förra IT-amanuens testade, men om ni använder Firefox så undviker ni problemet.

Om allt funkade som det ska så är det bara att trycka på "connect" i webbläsarfönstret för att ansluta till Romeokortet, och sen "Add servo" för att lägga till en servo. Du kan se på den lilla displayen uppe till vänster vilka knappar som lyser upp när du trycker på kontrollen. Det är lättaste sättet att lista ut vilken axel/knapp du ska välja i servons meny. Du kan även använda de olika skjutreglagen för att begränsa hur långt servomotorerna kan vrida sig.

När du har en konfiguration du är nöjd med så kan du spara ner den med "save" så att du kan ladda upp den med "load" om du inte vill lägga till alla servos igen nästa gång du ska använda



dem.

3 Tips

OBS Om du inte gjort några ändringar i Arduinokoden som kommer med repot så kommer Romeo utgå ifrån att du har 6 motorer anslutna, första på pin 8, andra på pin 9 osv. När du lägger till servos med "Add servo" så kommer de läggas till i den ordningen, så förslagsvis sätter du en första servo på pin 8 för att testa att allt funkar.

När du klickar på "connect" kommer servomotorerna röra sig till sitt mittenläge alternativt den vinkeln du ställt in. Det är därför smart att testa motorerna innan de är monterade. Annars finns det risk att robotarmen försöker röra sig längre än vad den fysiskt kan och knäcks alternativt förstör servot. Efter det kan ni montera motorerna med vetskapen att armen inte gör för farliga rörelser vid uppstart.

4 Felsökning

Om du koppar ur usb kabeln eller Romeo kortet tappar kontakt med datorn så kommer programmet sluta fungera och en omstart krävs. Då måste du gå in i terminalfönstret och trycka ned knapparna CTRL+C på tangentbordet för att stoppa programmet. Sen kan ni starta programmet igen som tidigare. Ett tips är att trycka pil-uppåt för att gå tillbaka till det senaste kommandot så behöver du inte skriva om allt igen.

Om det var länge sen du kopplade in Romeokortet, eller om du bytte till en annan usb port, kan porten ha förändrats. Då måste du gå öppna Arduino IDEn och kolla vilken ny port som kortet är inkopplat på och använda den när du startar programmet på nytt.

5 Example setup (kopierad från github)

1. Upload Arduino software to Romeo board:
 - (a) Connect Romeo to computer using the USB cable.
 - (b) Open Arduino sketch 'arduino/robotArmController.ino' using Arduino IDE.
 - (c) Make sure the correct COM port is selected under Tools- Port.
 - (d) Make sure the correct Board is selected under Tools- Board- Arduino Uno.
 - (e) Upload sketch to Romeo using Sketch- Upload.
2. Create a virtual environment for Python3:
 - Windows: 'python.exe -m venv venv'.
 - Linux: 'python -m venv venv'.
3. Activate the virtual environment:
 - Windows: 'venv/Scripts/activate.bat'.
 - Linux: '. venv/bin/activate'.
4. Install required Python modules using 'pip install -r requirements.txt'.
5. Start Python program with the COM port where the Arduino is connected:
 - Windows: 'python.exe . COM1 8765'. (Substitute COM1 with correct COM port)
 - Linux: 'python . /dev/ttyACM0 8765'. (Substitute ttyACM0 with correct COM port)