



# Documentation Robot Controller Windows

Henrik Wahlström

January 17, 2023

## 1 Intro

This guide is written and tested 2022-01-24 by then IT-amanuens Henrik Linder, with Windows Version 10.0.19042 Build 19042, Python 3.10.2, Firefox version 96.0.2.

There are no guaranties that it will be compatible with other versions or other web browsers.

Throughout this guide I will use `monospace text written like this` to indicate commands to write in the terminal.

No not hesitate to reach out to me (Henrik) at [it@tekniskfysik.se](mailto:it@tekniskfysik.se) if you have any questions.

Huge thanks to Fredrik Peteri Harr that has developed the code and made it available to us!

## 2 Installation

Download the GitHub repo from <https://github.com/frepet/Robot-Arm-Controller> by pressing the green button "Code" och then "download zip" (If you've used GitHub before you can of course do this in another way).

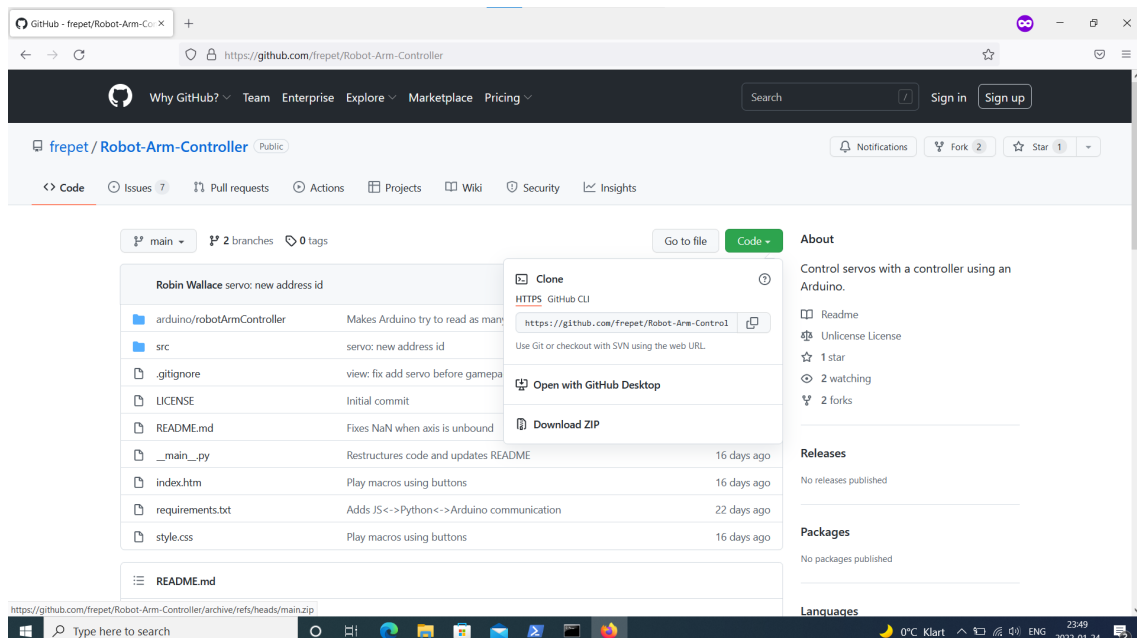


Figure 1: How it looks like to download the repo from GitHub.

Extract the folder to a location of your choosing (In the rest of the guide I will assume that you chose the folder "Documents").



Open a terminal (CMD or Powershell)

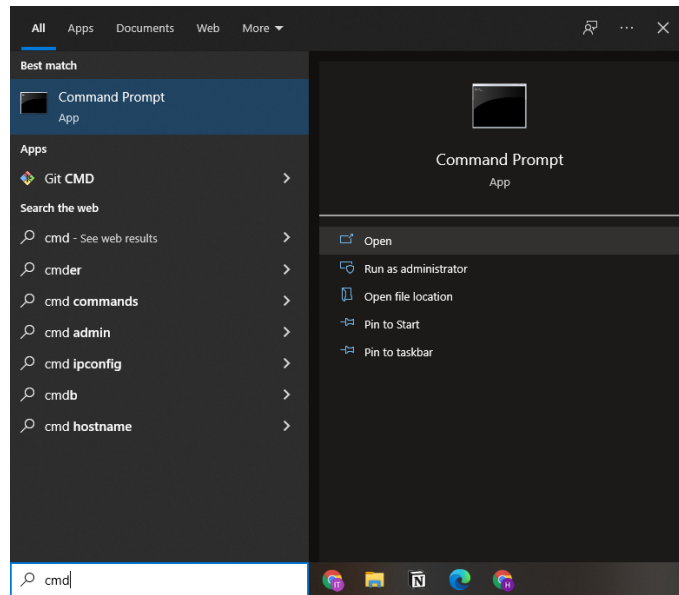


Figure 2: Open CMD (Powershell should work as well).

Write `python`. If Python is installed it will look like this:

```
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: How it looks like after the command `python` if python is installed.

In that case write `exit()` to go back out to the terminal.

If Python is not installed then you will be taken to the install page for Python in Microsoft Store: install Python from there and then write `python` again in the terminal to see that it is installed (`exit()` to go back).

Then go to the extracted folder in the file explorer, navigate until you see the file "index.htm", and copy the path to there. In my case it will be

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

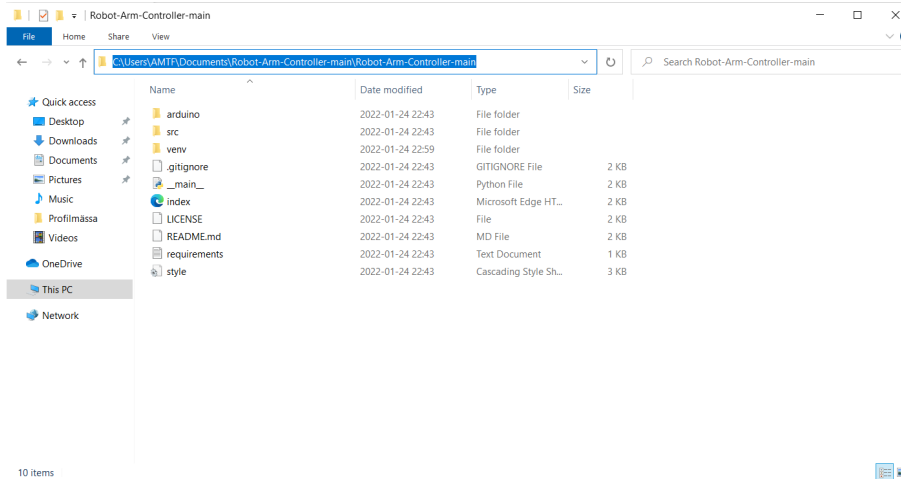


Figure 4: The folder that you should copy the path to (path highlighted in blue).

Go to the terminal, write `cd {PATH}`, which in my case is

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

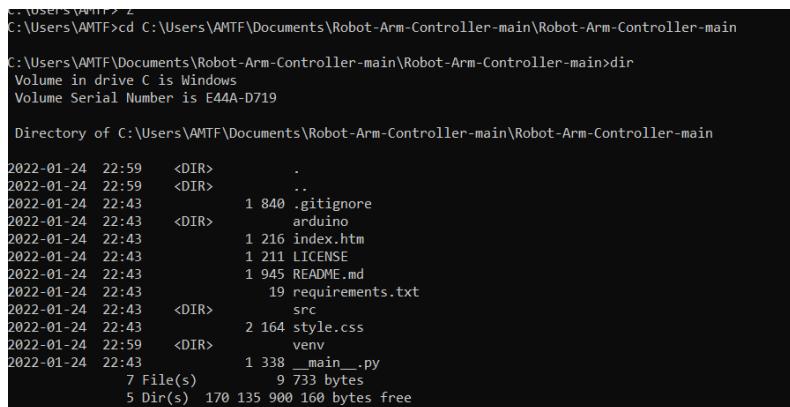


Figure 5: How it looks like to navigate to the robot-arm folder with `cd` and then show its content with `dir`.

If you write `dir` now you should see the same files that were in the folder that you downloaded.

Now that Python is installed you can start following the instructions in the example setup on GitHub (can also be read at the bottom of this document or in the file "README.md" that is located in the GitHub repo that you downloaded).

Step 1 and 2 should work without issues, otherwise ask the IT-amanuens or try again.

Step 3 can give errors if you are not allowed to write scripts for example, in that case you may need to write `set-executionpolicy RemoteSigned -Scope CurrentUser` to allow PowerShell to write scripts.

When you have followed all 5 steps, **make sure to open the file "index.htm" with Firefox** since the other web browsers might not work with the hand controller. it seems to have something to do with how the other browsers handle the JS-module "gamepad" that is used to read data from the controller. This issue might have been resolved in the time since the last IT-amauens tested it, but if you use Firefox you avoid the issue.

If everything works as it is supposed to then you just have to press "connect" in the browser window to connect to the Romeo card, and then "Add servo" to add a servo. You can see on the little display up to the left which buttons light up when you press them on the controller. This is the easiest way to figure out which axis/button you should choose in the servos' menu. You can also use the sliders to limit how far the servos are allowed to rotate.



When you have a configuration that you are happy with then you can save it with "save" so that you can load it with "load" if you don't want to add all servos next time you use them.

### 3 Tips

**NOTE** If you haven't made any changes to the Arduino code that comes with the repo, the Romeo will assume that you have 6 servos attached, with the first on pin 8, the second on pin 9 and so on. When you add servos with "Add servo", they will be added in that order, so a suggestion would be to connect the first servo to pin 8 to test that everything works.

When you press "connect" the servo motors will move to its middle position or the angle that you have set it to. It is therefore smart to test the motors before they are mounted. Otherwise there is a risk that the robot arm will try to move further than it physically can so that it breaks or possibly destroys the servo. After that you can mount the motors with the knowledge that the arm will not do any dangerous movements at startup.

### 4 Felsökning

If you disconnect the usb cable or the Romeo card loses contact with the computer the program will stop working and a restart is needed. Then you must go back to the terminal window and press the keyboard buttons CTRL+C to stop the program. Then you can start the program again as before. A tip is to press the up-arrow key to go back to the previously written command so that you don't have to type it all again. 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 för att stoppa programmet. Sen kan ni starta programmet igen som vanligt. Ett tips är att trycka pil-uppåt för att gå tillbaka till det senaste kommandot så behöver du inte skriva om allt igen.

IF it was a while since you connected the Romeo card, or if you used another usb port on your computer, then the port may have changed. You then must open the Arduino IDE and check what new port the card is connected on and use that when you start the program again.

### 5 Example setup (copied from 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)