

How to RTDE

Practical tips to set up “Real Time Data Exchange”

*by Henning Forbech, 4TECH Robotics ApS
Version: 1.0 Last update: 20/10 2023*

Scope:

The purpose of this document is to help users to get through the RTDE setup process with less practical problems.

When I first time have to set up a RTDE connection to a UR robot I ran into more practical problems than expected. I started this document to collect and save all information on how to setup a UR robot with data input from an external computer.

During the process is also grow into a history of all the complications and hints on how to solve some of the more practical problems in getting a RTDE connection up and running. So, it is not just a straight forward blueprint for setting up RTDE. In between there are some of my mishaps in this journey.

Universal Robots have some information on how to setup RTDE and more help can be found on different internet fora. But a lot of the more practical problem are not documented or, at least, I have not been able to find this information.

Tips on programming and using RTDE is not a part of this document.

Help with updates:

If you find this document useful, please help improving it by adding more information. You can also help by clearing out misunderstandings and errors that I have made.

When you make changes to the document, please update the version number and date in the header. You are also welcome to add our name to the document.

Background:

As part of a project, I wanted to send information from a Raspberry Pi to a UR robot. It should be easy to send this one-way information by the Real Time Date Transfer (RTDE).

Universal Robots have published a small Python module that should make it possible to make a direct connection from a Raspberry Pi to the UR controller. I couldn't get this program to compile and went for a solution where a Windows PC should make the connection between the Raspberry Pi and the UR controller. Data was then sent from the Raspberry Pi to this Connecting PC by WiFi and UDP. The challenge was now to get the Connection PC to send data to the UR controller by RTDE.

This description is based on a connecting between a PC running Windows 10 and a UR5 CB31 running PolyScope 3.15.

Windows PC for connecting to UR-controller with RTDE:

The RTDE connection to the UR controller is made from a PC running Windows 10 Pro. This computer must be set up with Python to run the program that make the RTDE connection to the UR controller.

IP-address and ports must also be set up to match the UR controller and the LAN cable must be connected.

In this document I will only describe how to setup the PC, the robot and the connection between them.

SimpleStudent have made a nice series of videos on how to set up and run RTDE for the first time. This is a must see when you are ready run the RTDE connection.

The first three videos give a good understanding of how to get RTDE going.

<https://youtube.com/playlist?list=PLnJ9fSRnDN3B1wEuxQY4thTWyGoT2N0yd>

<https://www.youtube.com/@simplestudent279>

I will only describe how to install the programs and set up the computer.

Python program:

The RTDE program for the Windows computer is a small program that must run from Python.

If you are not familiar with programming in Python the smartest is to install and use the same programs as SimpleStudent is using for his nice video tutorials on how to set up the communication.

First you must install an editor, PyCharm, for Python.

Go to the address below, download and install the program.

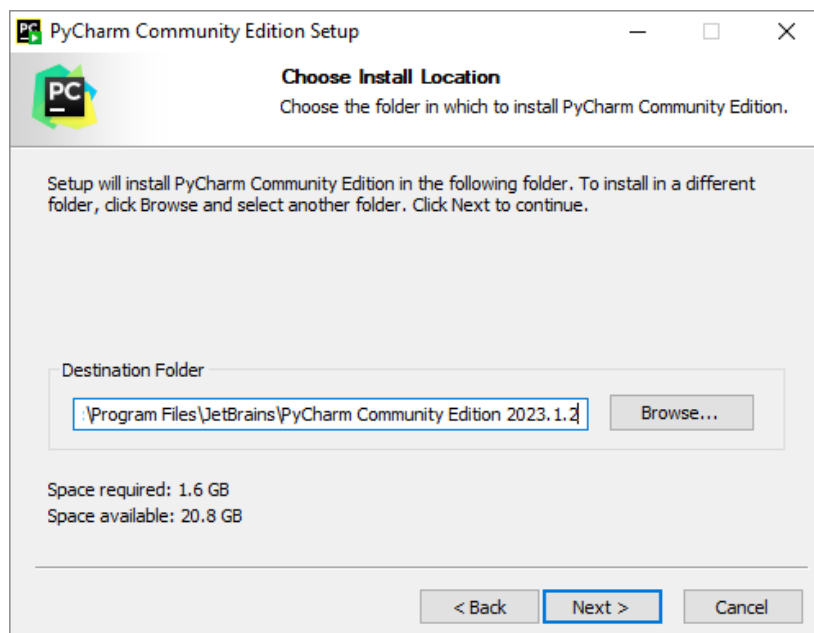
"PyCharm Community" is the free version of the program:

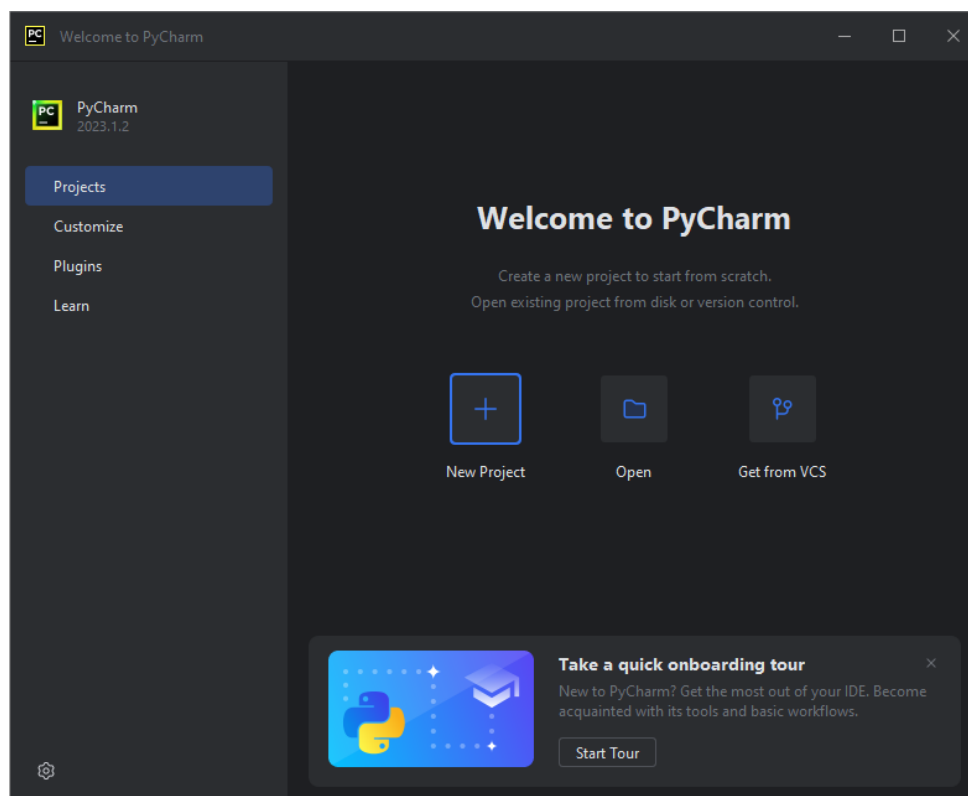
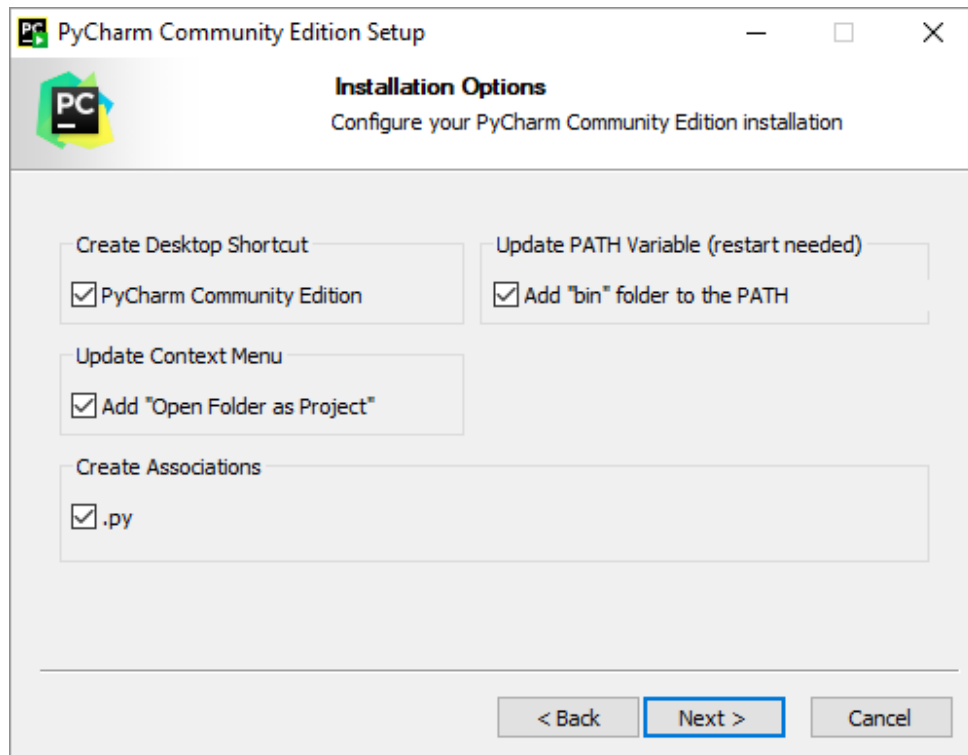
<https://www.jetbrains.com/pycharm/download/other.html>

[https://download.jetbrains.com/python/pycharm-community-](https://download.jetbrains.com/python/pycharm-community-2023.1.2.exe?_ga=2.210391829.167791767.1686485657-512205707.1686485657)

[2023.1.2.exe?_ga=2.210391829.167791767.1686485657-](https://download.jetbrains.com/python/pycharm-community-2023.1.2.exe?_ga=2.210391829.167791767.1686485657-512205707.1686485657)

[512205707.1686485657](https://download.jetbrains.com/python/pycharm-community-2023.1.2.exe?_ga=2.210391829.167791767.1686485657-512205707.1686485657)





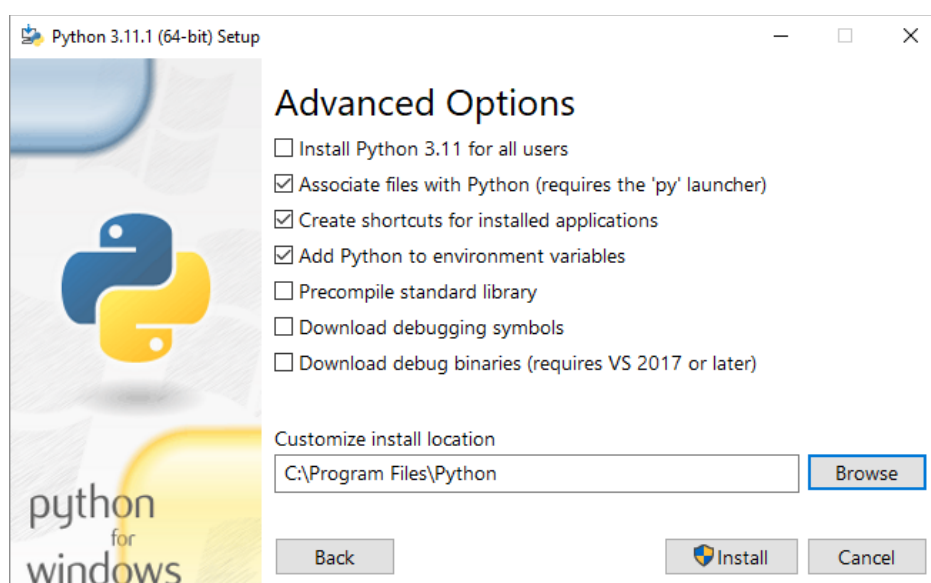
Installing Python:

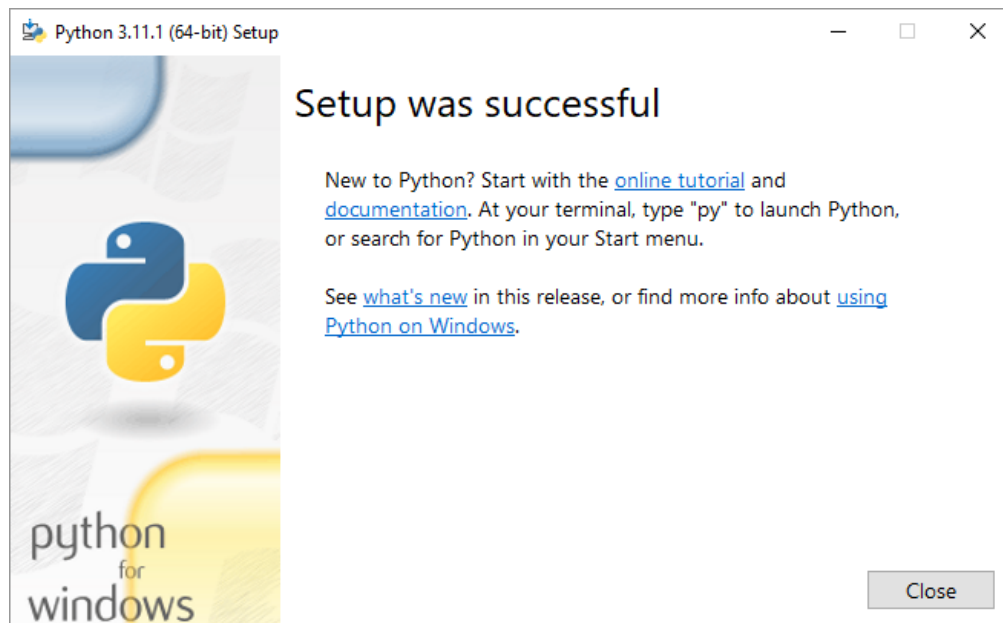
When you install the PyCharm editor, it will also install Python on your computer. But that version of Python will be installed under UserApps and it may be difficult/impossible to update this program. Windows will block it from running an automatically update!

You can solve this by installing Python in a folder outside UserApps. Go to Python.org and download a free version of Python.
Link: <https://www.python.org/downloads/windows/>

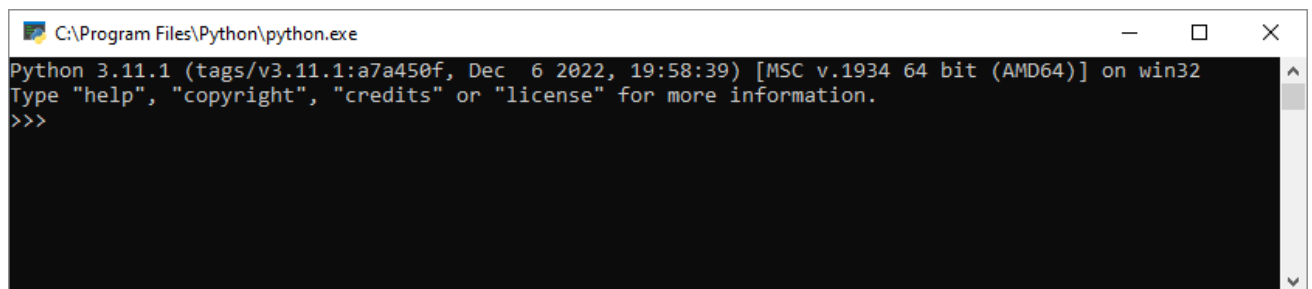
Make sure to get a version that fit your computer and it should be version 3.9 or later.

Make a new folder for this program (e.g. C:/Python or C:\Program Files/Python) and run the installation as Administrator.

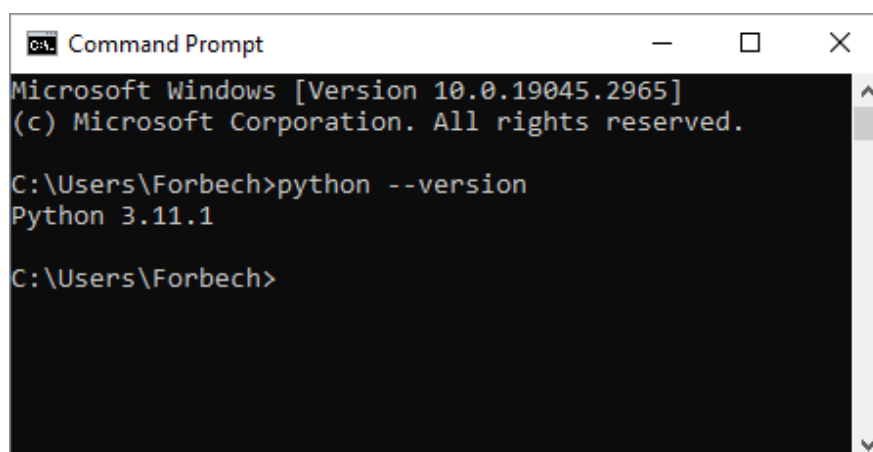




Once installed open the command prompt with "Window"+R and CMD, go to the folder and run the python.exe
At the top of the window, you will see the version number for your Python.



From the command prompt you can also just type: python --version



Python for PyCharm:

When you are about to run a python program from the PyCharm editor it will ask you which Python to use.
Make sure to use the one you installed outside UserApps.

Installing the RTDE program for Python:

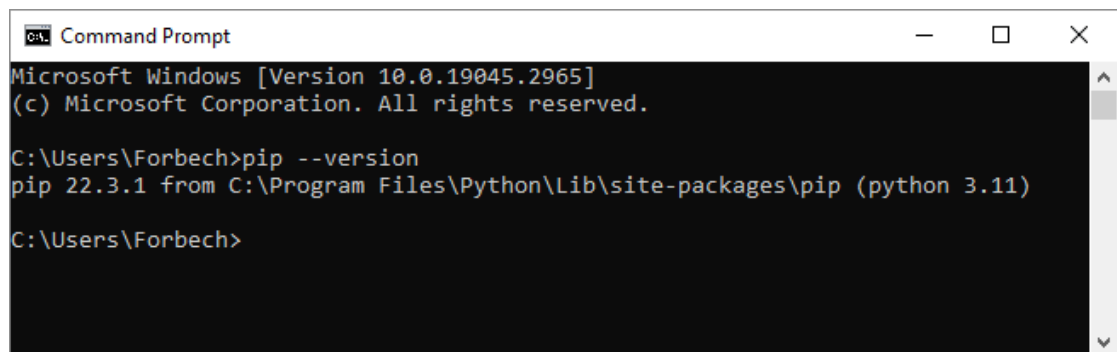
Python uses a lot of libraries for running programs. These libraries must be installed for the Python to use them.

Most important for this project is the RTDE library from Universal Robots. This program can be downloaded for free from GitHub but the simplest is just to let Python find and install it by running the PIP program. But first you must be sure to have the right version of the PIP program.

PIP:

You will need at least version 19.3 of PIP (RTDE will not install with older versions of PIP).

Check your PIP version from the command prompt: `pip --version`



```
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Forbech>pip --version
pip 22.3.1 from C:\Program Files\Python\Lib\site-packages\pip (python 3.11)

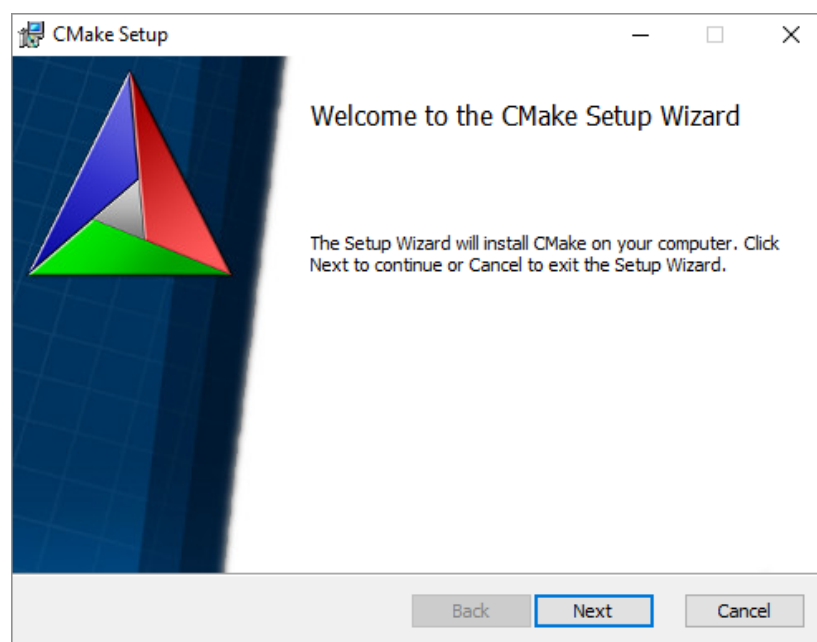
C:\Users\Forbech>
```

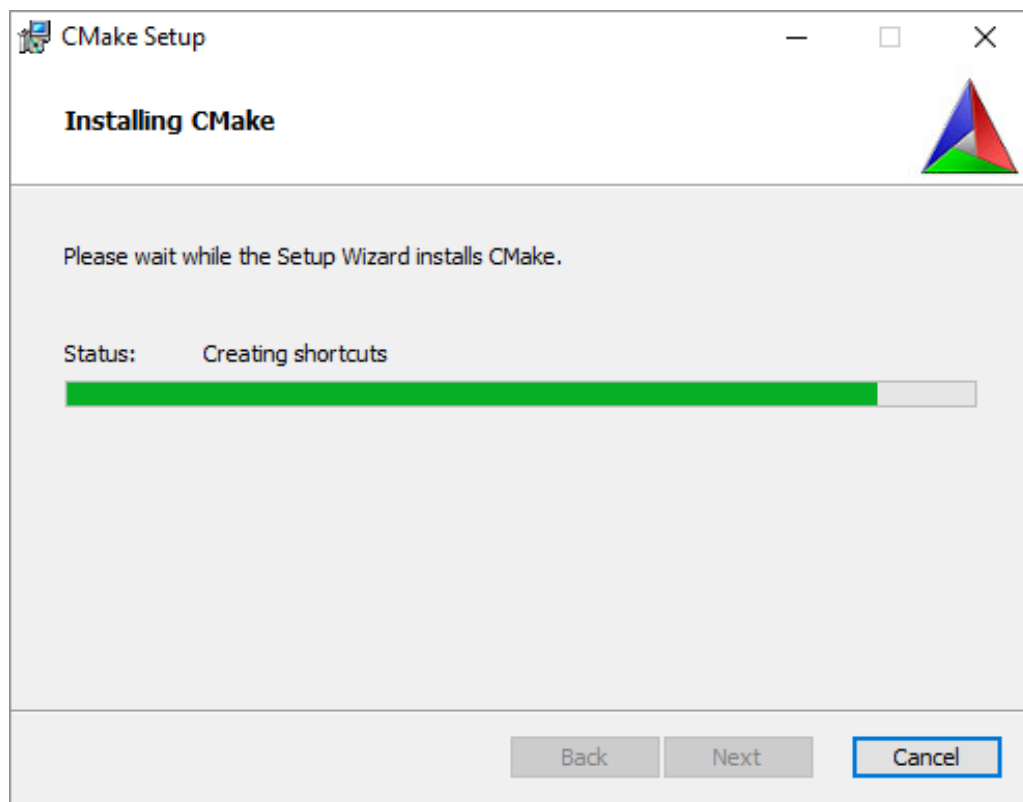
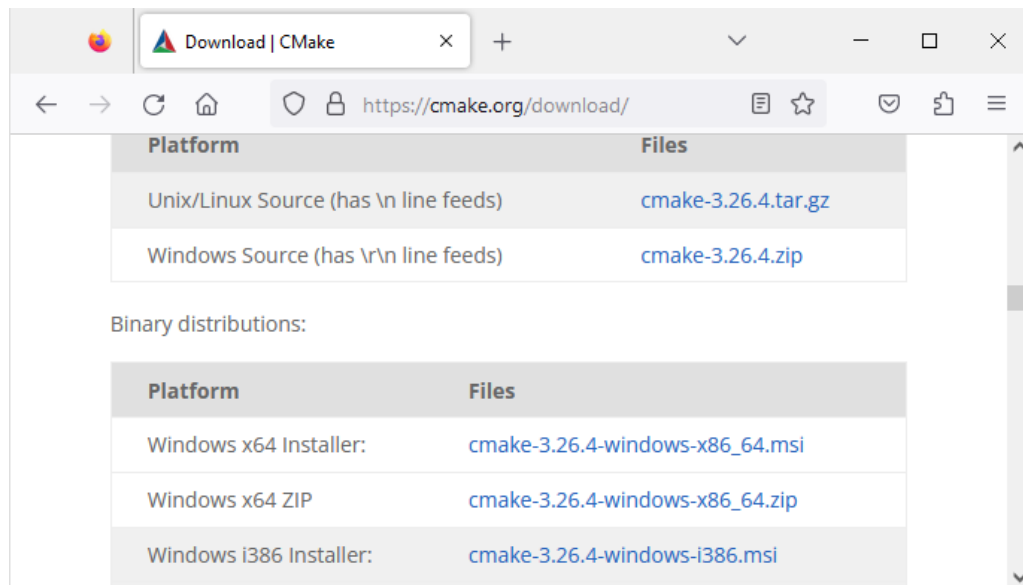
If you need to install a newer version of the PIP program, use the command:
`python -m pip install --upgrade pip`

For Python to install RTDE it will need two programs called Cmake and Boost. It can be a nightmare to get these programs installed but here we go:

Cmake:

Download and install Cmake from: <https://cmake.org/download/>



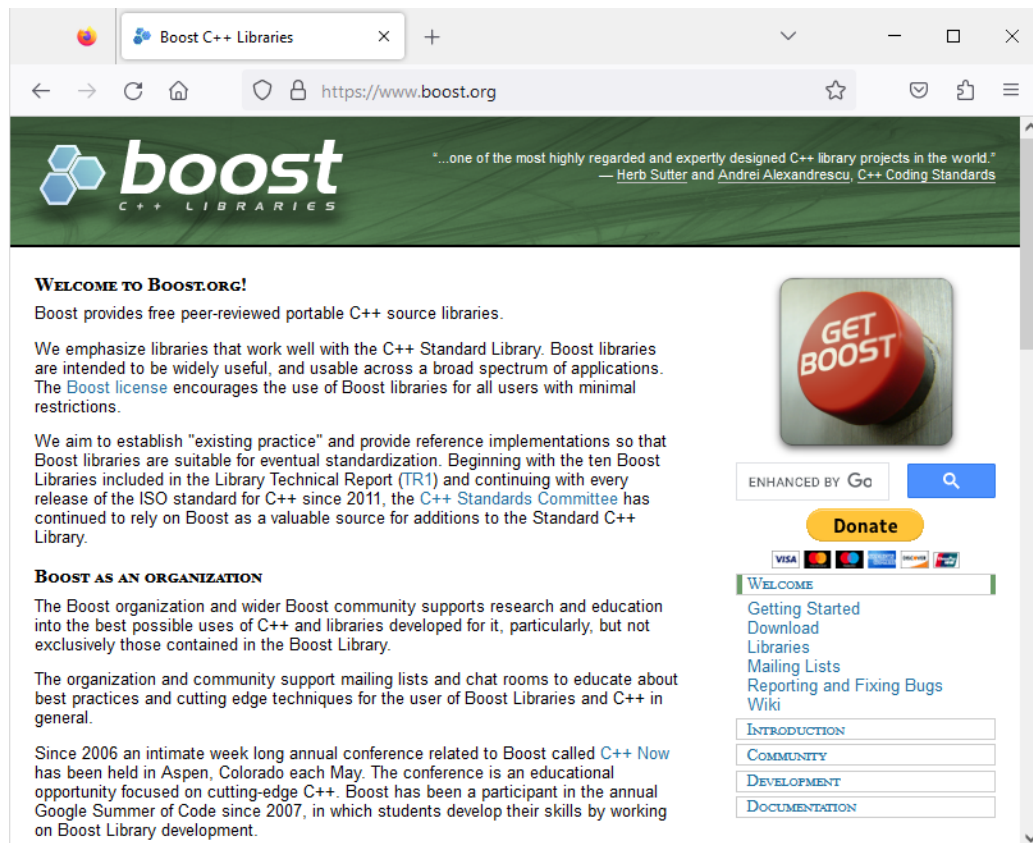


Boost:

You will also need a program called Boost.
It is a free C++ source library.
Go to: <https://www.boost.org>

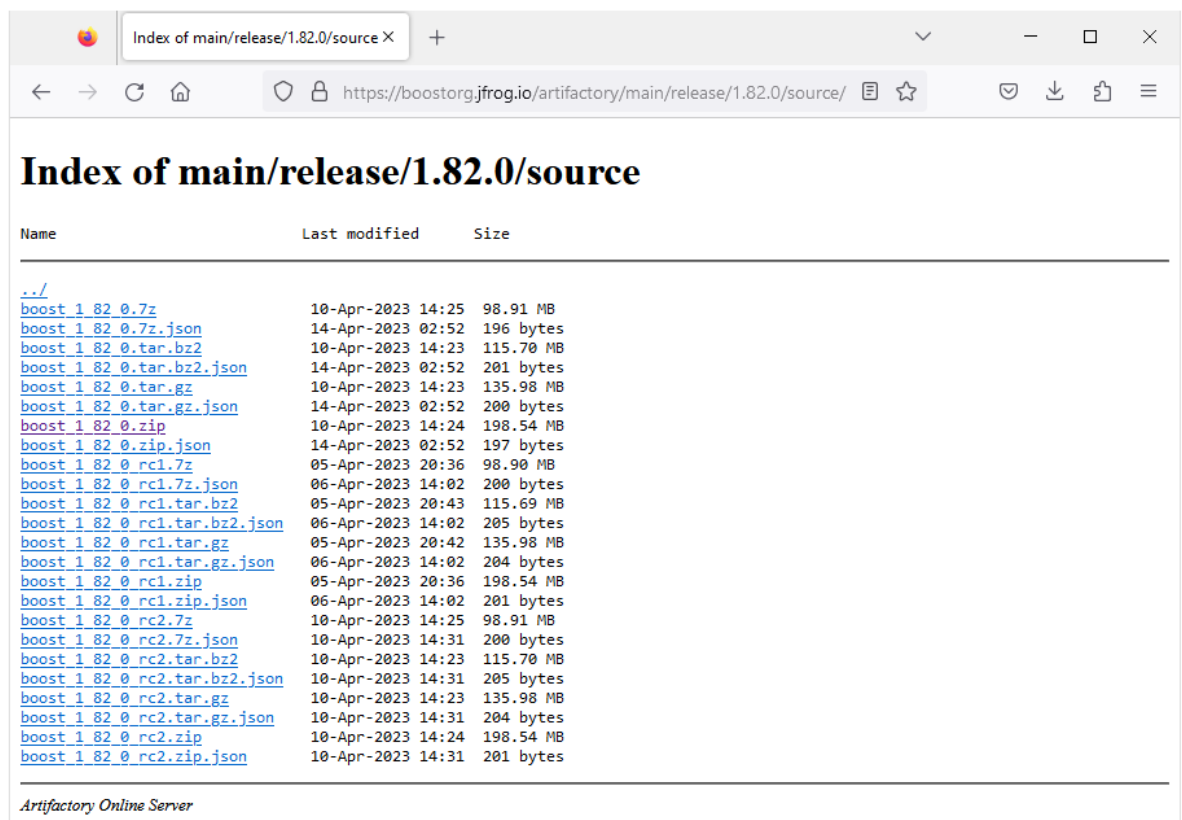
Warning:

This first attempt to install Boost will fail!
Look below for a fix



The screenshot shows the Boost C++ Libraries website. The header features the Boost logo and a quote: "...one of the most highly regarded and expertly designed C++ library projects in the world." — Herb Sutter and Andrei Alexandrescu, C++ Coding Standards. The main content area includes a "WELCOME TO BOOST.ORG!" section, a "Boost provides free peer-reviewed portable C++ source libraries." statement, and a "Boost AS AN ORGANIZATION" section. On the right, there is a "GET BOOST" button, a "Donate" button, and a "WELCOME" section with links to "Getting Started", "Download", "Libraries", "Mailing Lists", "Reporting and Fixing Bugs", and "Wiki".

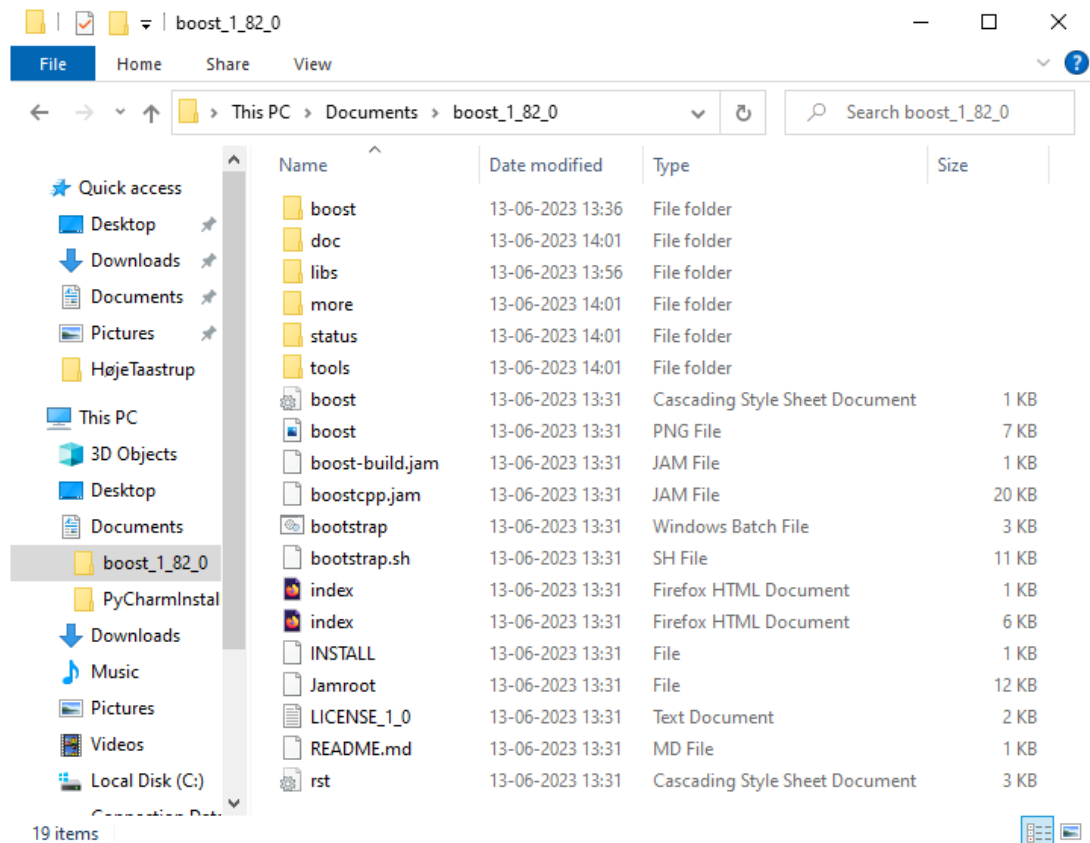
Go to "Download" and download the 200 Mb zip file and unzip on your hard disk (704 Mb unzipped).



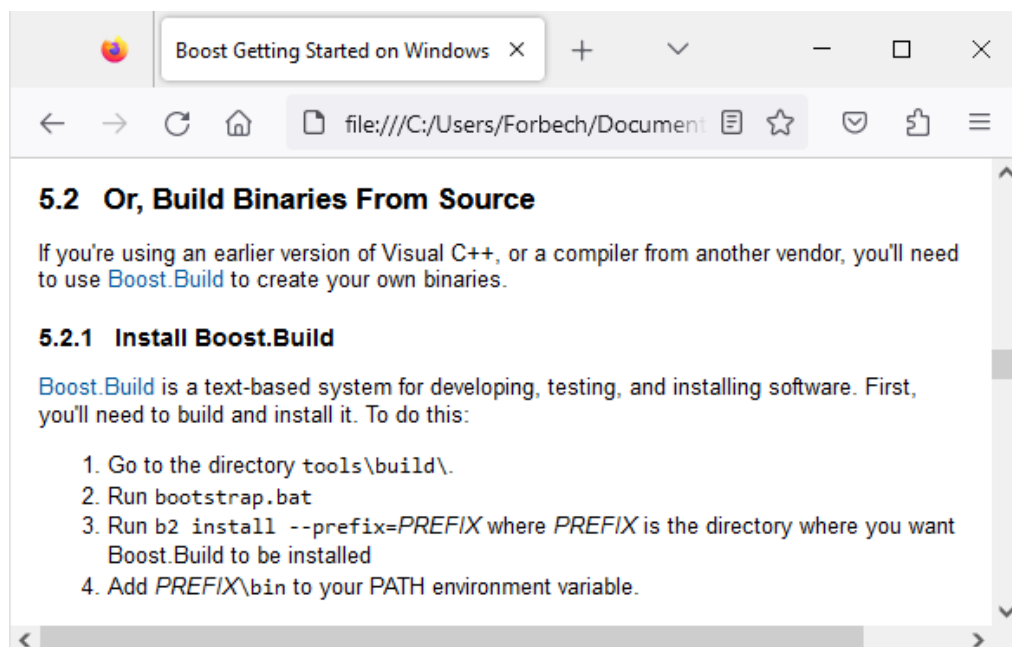
The screenshot shows the "Index of main/release/1.82.0/source" page. It lists various download options for Boost 1.82.0, including source code, binaries, and documentation. The table below summarizes the listed files and their sizes.

Name	Last modified	Size
./		
boost_1_82_0.7z	10-Apr-2023 14:25	98.91 MB
boost_1_82_0.7z.json	14-Apr-2023 02:52	196 bytes
boost_1_82_0.tar.bz2	10-Apr-2023 14:23	115.70 MB
boost_1_82_0.tar.bz2.json	14-Apr-2023 02:52	201 bytes
boost_1_82_0.tar.gz	10-Apr-2023 14:23	135.98 MB
boost_1_82_0.tar.gz.json	14-Apr-2023 02:52	200 bytes
boost_1_82_0.zip	10-Apr-2023 14:24	198.54 MB
boost_1_82_0.zip.json	14-Apr-2023 02:52	197 bytes
boost_1_82_0_rc1.7z	05-Apr-2023 20:36	98.90 MB
boost_1_82_0_rc1.7z.json	06-Apr-2023 14:02	200 bytes
boost_1_82_0_rc1.tar.bz2	05-Apr-2023 20:43	115.69 MB
boost_1_82_0_rc1.tar.bz2.json	06-Apr-2023 14:02	205 bytes
boost_1_82_0_rc1.tar.gz	05-Apr-2023 20:42	135.98 MB
boost_1_82_0_rc1.tar.gz.json	06-Apr-2023 14:02	204 bytes
boost_1_82_0_rc1.zip	05-Apr-2023 20:36	198.54 MB
boost_1_82_0_rc1.zip.json	06-Apr-2023 14:02	201 bytes
boost_1_82_0_rc2.7z	10-Apr-2023 14:25	98.91 MB
boost_1_82_0_rc2.7z.json	10-Apr-2023 14:31	200 bytes
boost_1_82_0_rc2.tar.bz2	10-Apr-2023 14:23	115.70 MB
boost_1_82_0_rc2.tar.bz2.json	10-Apr-2023 14:31	205 bytes
boost_1_82_0_rc2.tar.gz	10-Apr-2023 14:23	135.98 MB
boost_1_82_0_rc2.tar.gz.json	10-Apr-2023 14:31	204 bytes
boost_1_82_0_rc2.zip	10-Apr-2023 14:24	198.54 MB
boost_1_82_0_rc2.zip.json	10-Apr-2023 14:31	201 bytes

Artifactory Online Server



You only need to install the BoostBuild.
Follow this description from the “Boost Getting Started on Windows” documentation.



Open the folder and run the bootstrap.bat

```

C:\Windows\system32\cmd.exe

Volume in drive C has no label.
Volume Serial Number is BA99-4E63

Directory of C:\Users\Forbech\Documents\boost_1_82_0

13-06-2023  14:44    <DIR>          .
13-06-2023  14:44    <DIR>          ..
13-06-2023  13:36    <DIR>          boost
13-06-2023  13:31             850 boost-build.jam
13-06-2023  13:31             989 boost.css
13-06-2023  13:31            6.308 boost.png
13-06-2023  13:31           20.013 boostcpp.jam
13-06-2023  13:31            2.486 bootstrap.bat
13-06-2023  13:31           10.811 bootstrap.sh
13-06-2023  14:01    <DIR>          doc
13-06-2023  13:31             769 index.htm
13-06-2023  13:31            5.418 index.html
13-06-2023  13:31             291 INSTALL
13-06-2023  13:31           11.947 Jamroot
13-06-2023  13:56    <DIR>          libs
13-06-2023  13:31           1.338 LICENSE_1_0.txt
13-06-2023  14:01    <DIR>          more
13-06-2023  13:31             542 README.md
13-06-2023  13:31            2.608 rst.css
13-06-2023  14:01    <DIR>          status
13-06-2023  14:01    <DIR>          tools
                13 File(s)         64.370 bytes
                8 Dir(s)  23.446.142.976 bytes free

C:\Users\Forbech\Documents\boost_1_82_0>bootstrap.bat

```

This is what I got:

```

C:\Windows\system32\cmd.exe

C:\Users\Forbech\Documents\boost_1_82_0>bootstrap.bat
Building Boost.Build engine
LOCALAPPDATA=C:\Users\Forbech\AppData\Local
could not find "vswhere"
Call_If_Exists "..\bin\VCVARS32.BAT"
###
### Using 'msvc' toolset.
###

C:\Users\Forbech\Documents\boost_1_82_0\tools\build\src\engine>"cl" /nologo /MP
/MT /TP /Feb2 /wd4996 /O2 /GL /EHsc -DNDEBUG builtins.cpp class.cpp command
.cpp compile.cpp constants.cpp cwd.cpp debug.cpp debugger.cpp execcmd.cpp execn
t.cpp execunix.cpp filent.cpp filesys.cpp fileunix.cpp frames.cpp function.cpp
glob.cpp hash.cpp hcache.cpp hdrmacro.cpp headers.cpp jam.cpp jamgram.cpp lists
.cpp make.cpp make1.cpp md5.cpp mem.cpp modules.cpp native.cpp object.cpp optio
n.cpp output.cpp parse.cpp pathnt.cpp pathsys.cpp pathunix.cpp regex.cpp rules
.cpp scan.cpp search.cpp jam_strings.cpp startup.cpp subst.cpp sysinfo.cpp time
stamp.cpp variable.cpp w32_getreg.cpp modules/order.cpp modules/path.cpp module
s/property-set.cpp modules/regex.cpp modules/sequence.cpp modules/set.cpp /link
kernel32.lib advapi32.lib user32.lib
'"cl"' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Forbech\Documents\boost_1_82_0\tools\build\src\engine>dir *.exe
Volume in drive C has no label.
Volume Serial Number is BA99-4E63

Directory of C:\Users\Forbech\Documents\boost_1_82_0\tools\build\src\engine

File Not Found

Failed to build Boost.Build engine.

C:\Users\Forbech\Documents\boost_1_82_0>

```

There is a bug in the program and it will not install!!!!

How to fix the Boost problem:

I'm not sure what the problem is with the Boost 1.82.0 but I have earlier used the Boost 1.81.0. This program also has a bug (!) but I have found a way to bypass this problem.

You will find the install files for Boost 1.18.0 here:

https://www.boost.org/users/history/version_1_81_0.html

Unzip the files to your hard disk, e.g., C:\boost_1_81_0

Bootstrap.bat will go to C:\boost_1_81_0\tools\build\src\engine and look for b2.exe and then delete this file. But there is no b2.exe here and the program stop with an error.

This bug can be fixed by creating a b2.exe file and put it in the C:\boost_1_82_0\tools\build\src\engine directory.

This fake b2.exe file will be found and deleted by the bat program and then it will build the real b2.exe program.

Stupid error but this fix solves the problem.

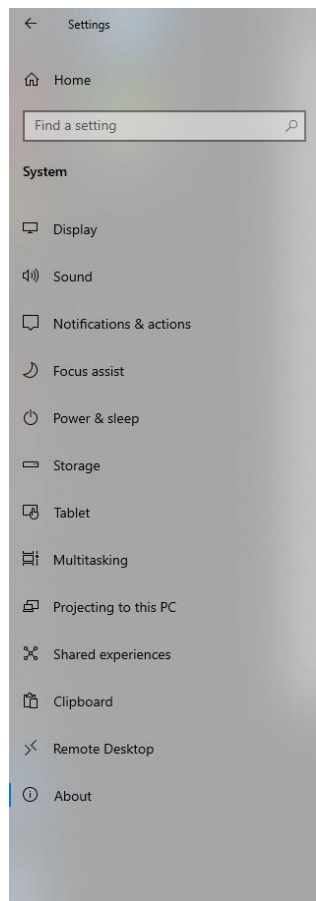
Now we can run the b2 program to install Boost.

Boost path update in Windows:

Now we must add the path for the new boost directory to the windows environmental variables. Follow this description:

To add a path to the PATH environment variable

1. On the **Start** menu, right-click **Computer**.
2. On the context menu, click **Properties**.
3. In the **System** dialog box, click **Advanced system settings**.
4. On the **Advanced** tab of the **System Properties** dialog box, click **Environment Variables**.
5. In the **System Variables** box of the **Environment Variables** dialog box, scroll to **Path** and select it.
6. Click the lower of the two **Edit** buttons in the dialog box.
7. In the **Edit System Variable** dialog box, scroll to the end of the string in the **Variable value** box and add a semicolon (;).
8. Add the new path after the semicolon.
9. Click **OK** in three successive dialog boxes, and then close the **System** dialog box.



This page has a few new settings

Some settings from Control Panel have moved here, and you can copy your PC info so it's easier to share.

Related settings

[BitLocker settings](#)

[Device Manager](#)

[Remote desktop](#)

[System protection](#)

[Advanced system settings](#)

[Rename this PC \(advanced\)](#)

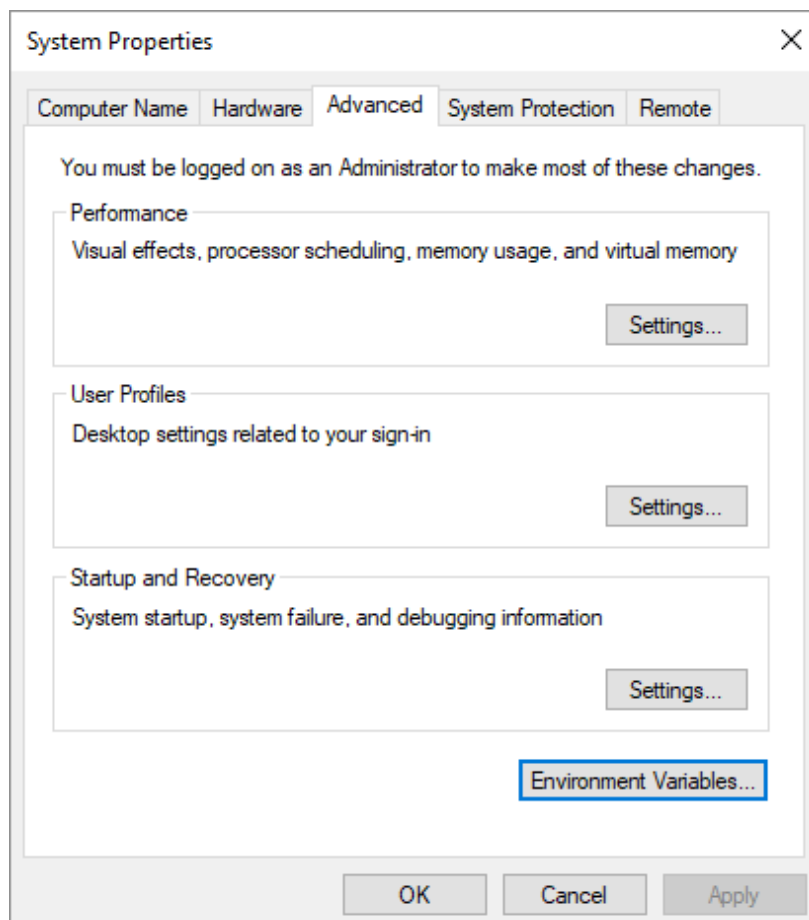
Help from the web

[Finding out how many cores my processor has](#)

[Checking multiple Languages support](#)

[Get help](#)

[Give feedback](#)



%SystemRoot%\system32
 %SystemRoot%
 %SystemRoot%\System32\Wbem
 %SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
 %SYSTEMROOT%\System32\OpenSSH\
 C:\Program Files\CMake\bin
 C:\boost_1_82_0\boost\bin

New
 Edit
 Browse...
 Delete
 Move Up
 Move Down
 Edit text...

OK
 Cancel

Installing RTDE:

Now you are finally ready to install RTDE.

The latest RTDE can be found here: <https://pypi.org/project/ur-rtde/>
From the command prompt use the command: `pip install ur-rtde`

If everything goes as planned you will get a confirmation the RTDE program is installed and can be used from your Python program.

When I first installed RTDE on an old laptop I got this conformation:

```

--proxy <proxy>          Specify a proxy in the form scheme://[user:passwd@]proxy.server:port.
--retries <retries>       Maximum number of retries each connection should attempt (default 5 times).
--timeout <sec>           Set the socket timeout (default 15 seconds).
--exists-action <action> Default action when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup,
                          (a)bort.
--trusted-host <hostname> Mark this host or host:port pair as trusted, even though it does not have valid or any
                          HTTPS.
--cert <path>             Path to PEM-encoded CA certificate bundle. If provided, overrides the default. See 'SSL
                          Certificate Verification' in pip documentation for more information.
--client-cert <path>      Path to SSL client certificate, a single file containing the private key and the
                          certificate in PEM format.
--cache-dir <dir>         Store the cache data in <dir>.
--no-cache-dir            Disable the cache.
--disable-pip-version-check Don't periodically check PyPI to determine whether a new version of pip is available for
                          download. Implied with --no-index.
--no-color                Suppress colored output.
--no-python-version-warning Silence deprecation warnings for upcoming unsupported Python versions.
--use-feature <feature>   Enable new functionality, that may be backward incompatible.
--use-deprecated <feature> Enable deprecated functionality, that will be removed in the future.

C:\users\forbe>pip install --user ur_rtde
Collecting ur_rtde
  Downloading ur_rtde-1.5.5-cp310-cp310-win_amd64.whl (2.4 MB)
-----
2.4/2.4 MB 4.3 MB/s eta 0:00:00
Installing collected packages: ur_rtde
Successfully installed ur_rtde-1.5.5
C:\users\forbe>

```

This old laptop was unstable and I decided to setup a new Connection PC for my project. I went through all these steps but when I finally tried to install RTDE, it failed!

After spending a lot of time trying to solve the problem I gave up and made a desperate attempt to get the new PC running with setup from the old laptop. I made a copy of the laptop hard disk to the PC hard disk. This is normally not possible but to my big surprise the PC booted into Windows. After a lot of updates and restarts I had the Windows installation from the laptop running on the PC.

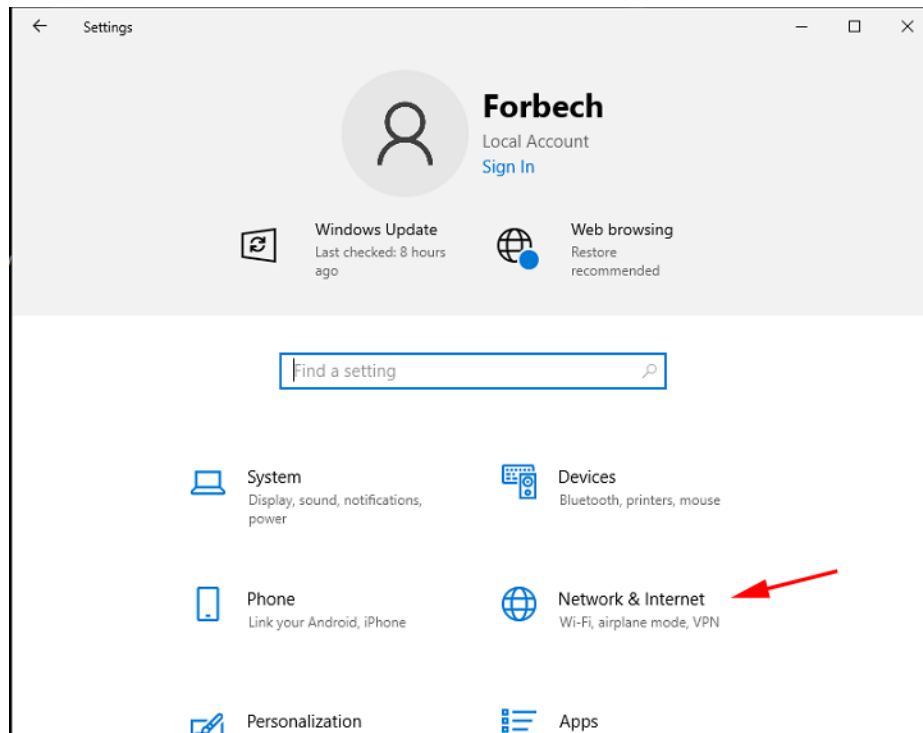
And RTDE was even running on the new hardware!!!

Later I realized that RTDE will only run on Windows Pro, not the Home edition!

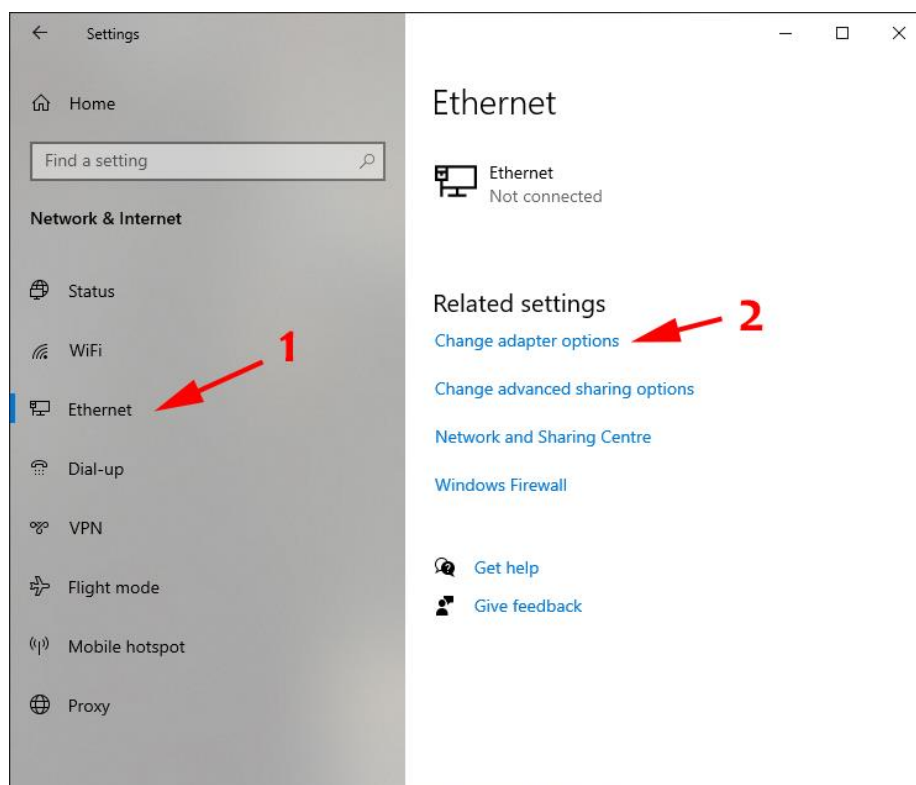
Windows network settings:

We are now ready to check the settings for the network adapter on the Connection PC. We have to find the IP address for the adapter where the net cable is attached.

Open Windows Settings and click on the Network & Internet.



Go to "Ethernet" and open "Change adapter options"

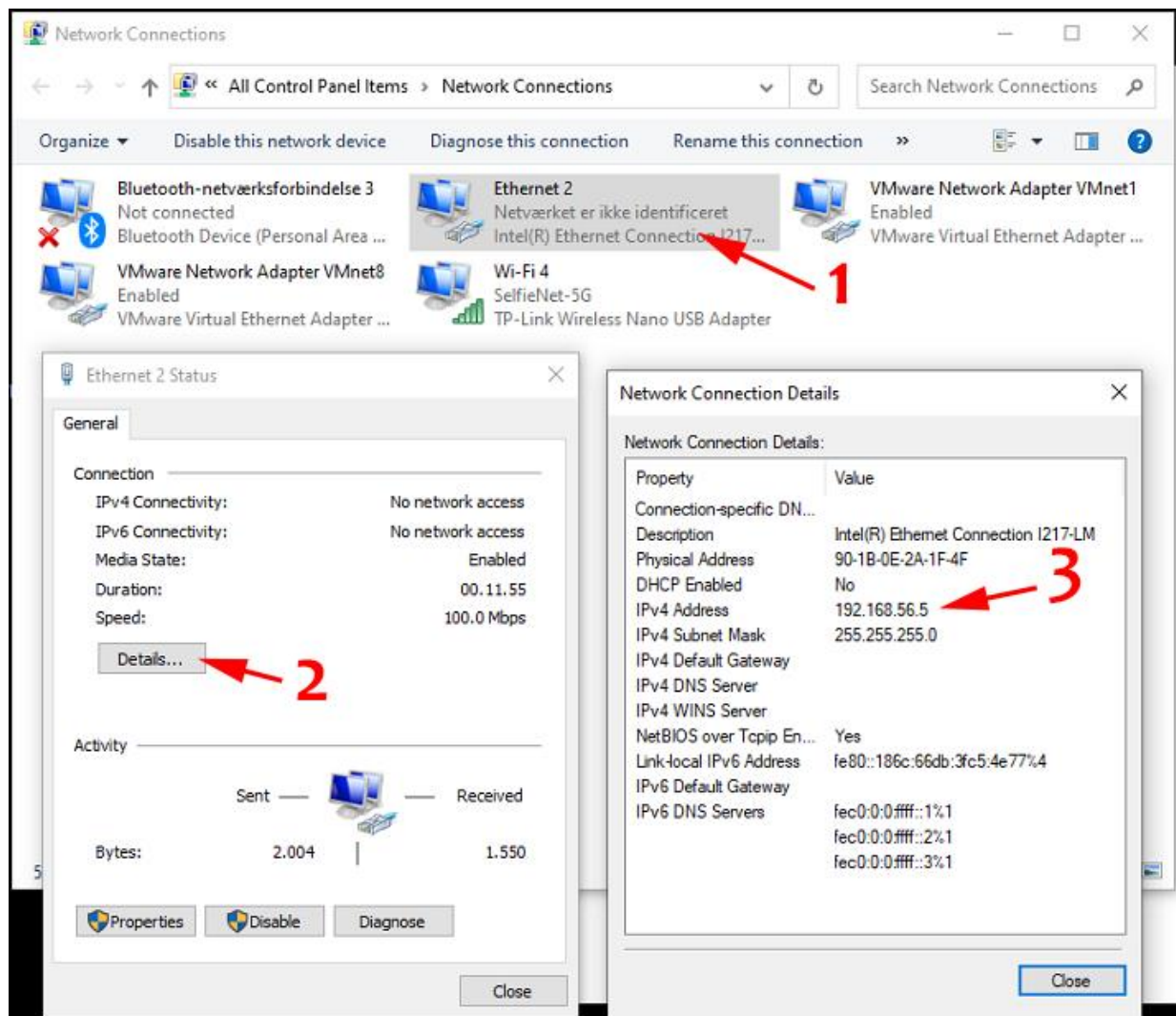


Here you will see all the network connections on the computer. WiFi, Bluetooth and different network adapters. The adapter for your LAN cable will be named "Ethernet" and maybe a number. You can check if it the right adapter by plugging or unplugging the LAN cable.

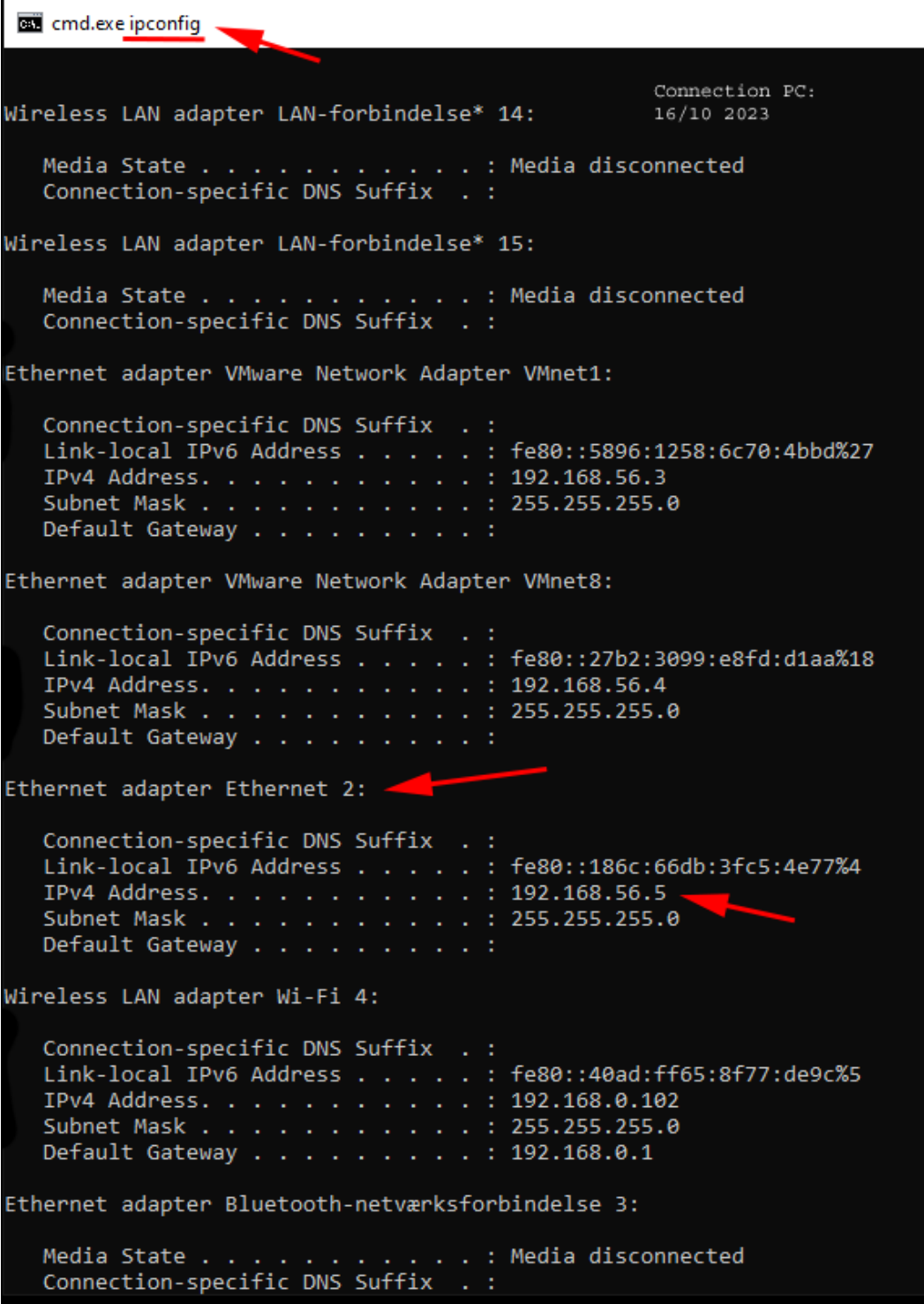
Click on the Ethernet to see status of the adapter and continuer by clicking on "Details".

You will now get a lot of information on this adapter. Look for the "IPv4 Address" for the adapter.

It is this IP address you will need to set up the connection on the UR controller.



If you are more hardcore you can also just open the Windows commando line and use "ipconfig" to get the information.



```
cmd.exe ipconfig

Wireless LAN adapter LAN-forbindelse* 14:
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter LAN-forbindelse* 15:
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter VMware Network Adapter VMnet1:
    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::5896:1258:6c70:4bbd%27
    IPv4 Address. . . . . : 192.168.56.3
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

Ethernet adapter VMware Network Adapter VMnet8:
    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::27b2:3099:e8fd:d1aa%18
    IPv4 Address. . . . . : 192.168.56.4
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

Ethernet adapter Ethernet 2:
    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::186c:66db:3fc5:4e77%4
    IPv4 Address. . . . . : 192.168.56.5
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

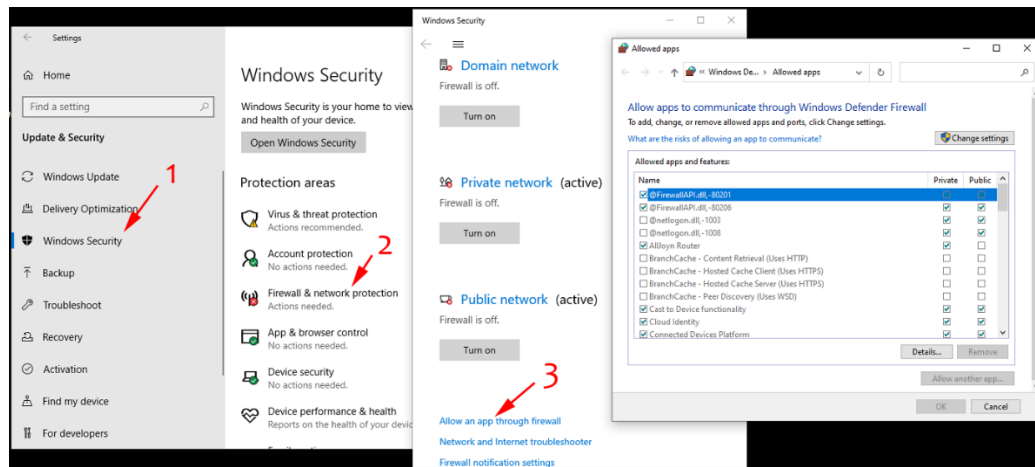
Wireless LAN adapter Wi-Fi 4:
    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::40ad:ff65:8f77:de9c%5
    IPv4 Address. . . . . : 192.168.0.102
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1

Ethernet adapter Bluetooth-netværksforbindelse 3:
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :
```

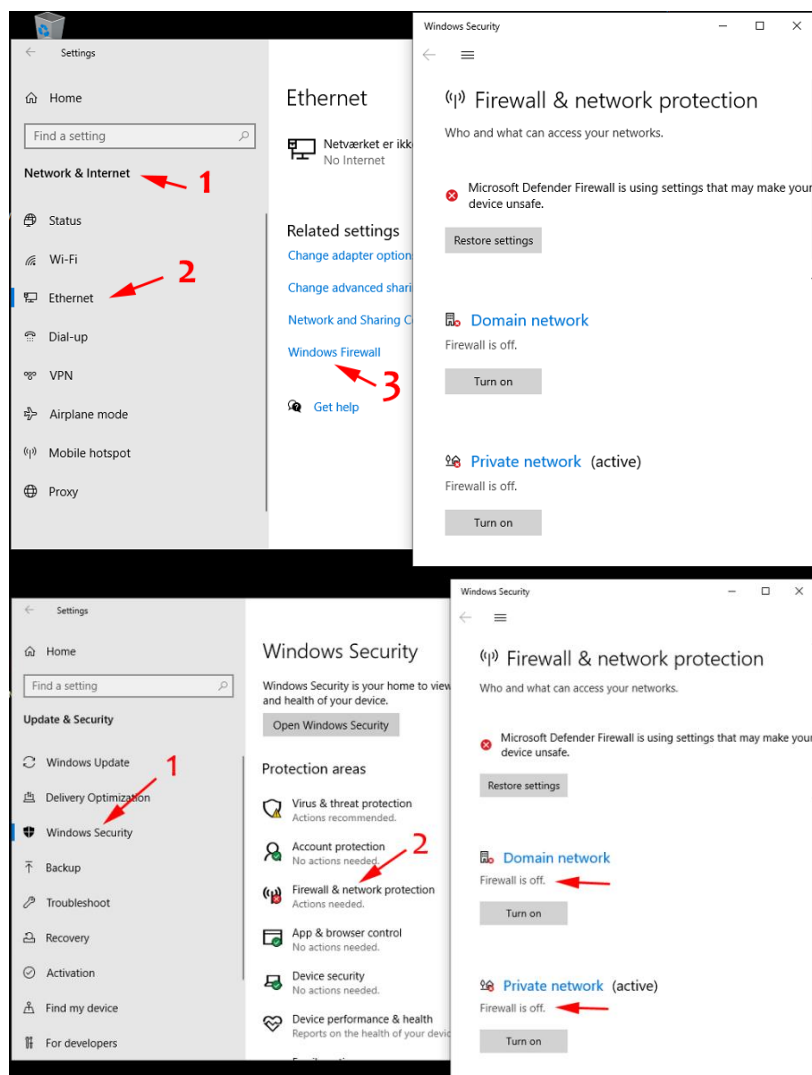
Allowing a program through Windows firewall:

Windows firewall may give some problems when the RTDE program try to send data to the UR controller.

You can allow python through the firewall with these steps:



For a start it is probably easier just to turn off the Firewall. When you got RTDE running you can try to turn the Firewall back on again. There are two ways to turn off Firewall:

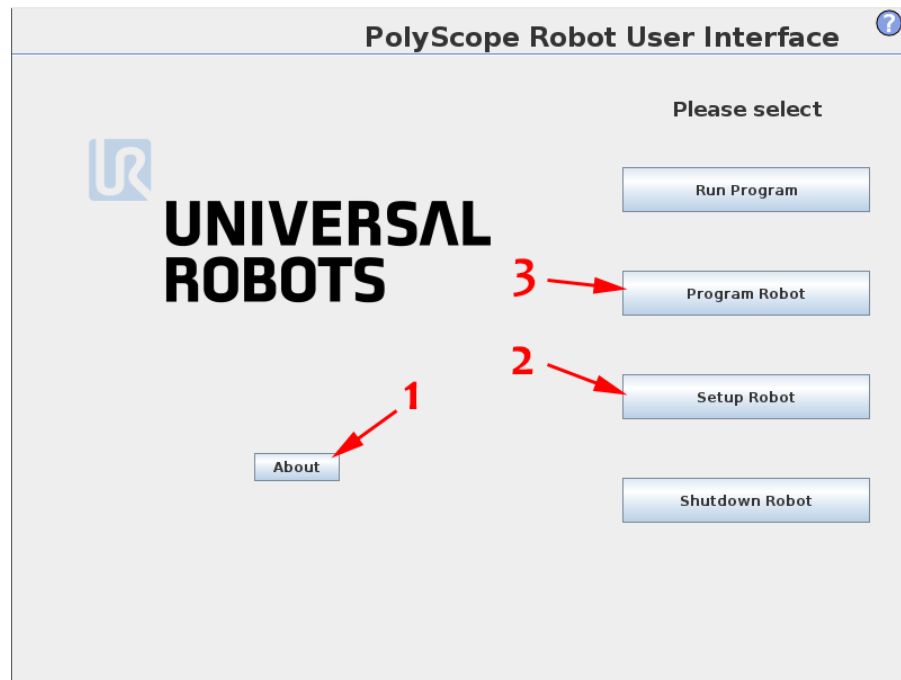


Setting up the UR controller:

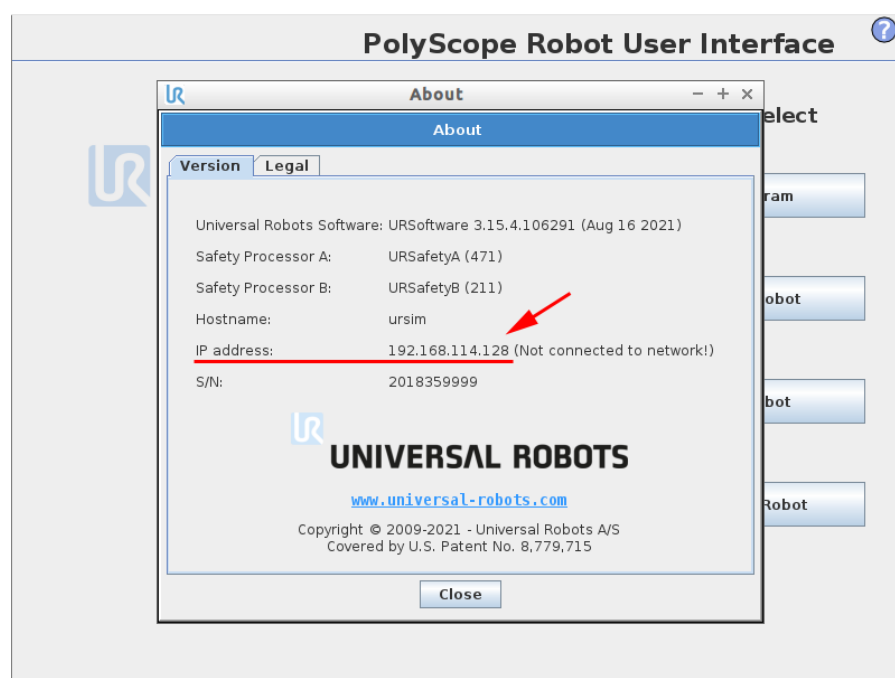
To prepare for a RTDE connection both the UR controller and the computer to connect to it need to be setup.

Here follow first a description on how to set up the UR controller and then how to setup the computer to make the connection.

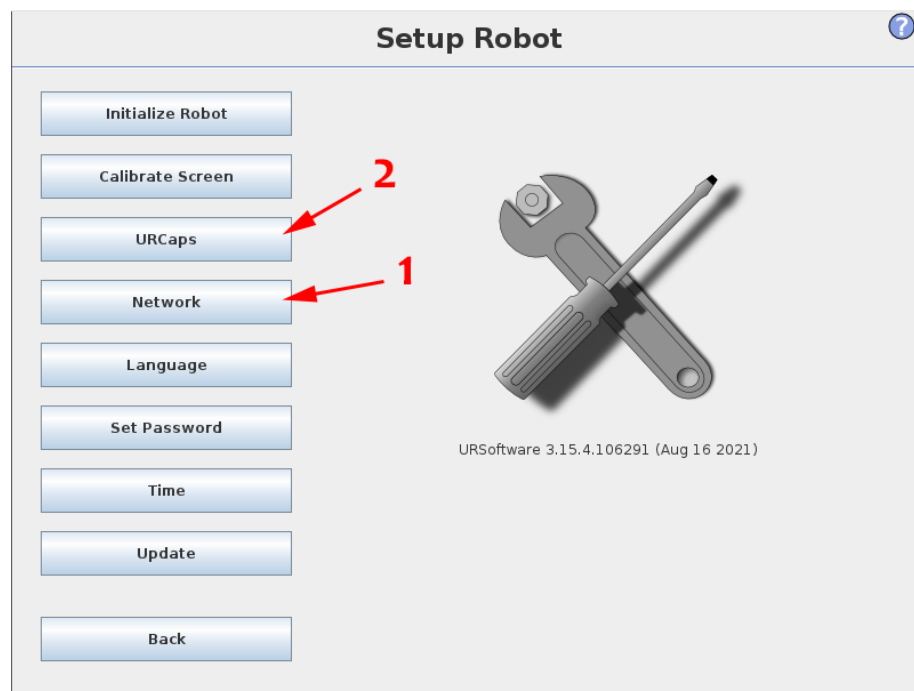
Go to the opening screen for the UR controller start by checking the status of your robot. Next you must do some settings from this start screen and later you must start programming the robot to some more settings to the installation.



The "About" button will give you information on the IP address for the robot.

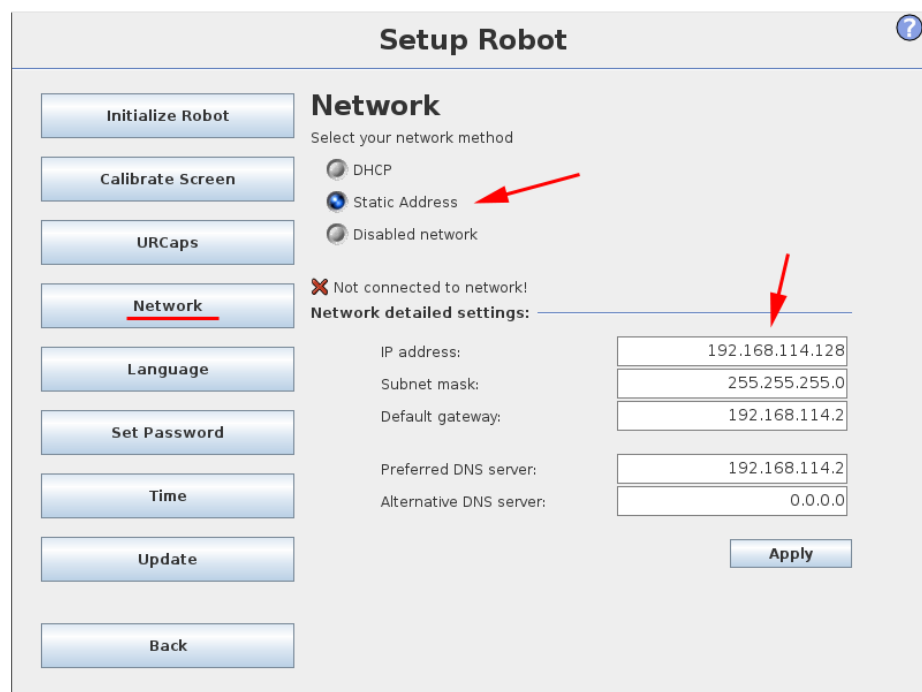


From the "Setup Robot" button you must set the network and check the URcaps.



Network:

To use RTDE the UR controller must be setup with a fixed IP address.



The IP address on the UR controller and the Windows PC must belong to the same IP subnet. That is the numbers in the first three groups must be the same but the last group must be close, but not the same.

Example: If your PC is 192.168.1.1 the UR controller could be 192.168.1.2

If you cannot change the IP address on the Windows computer you must change the IP address on the UR controller.

Subnet mask must be set to 255.255.255.0

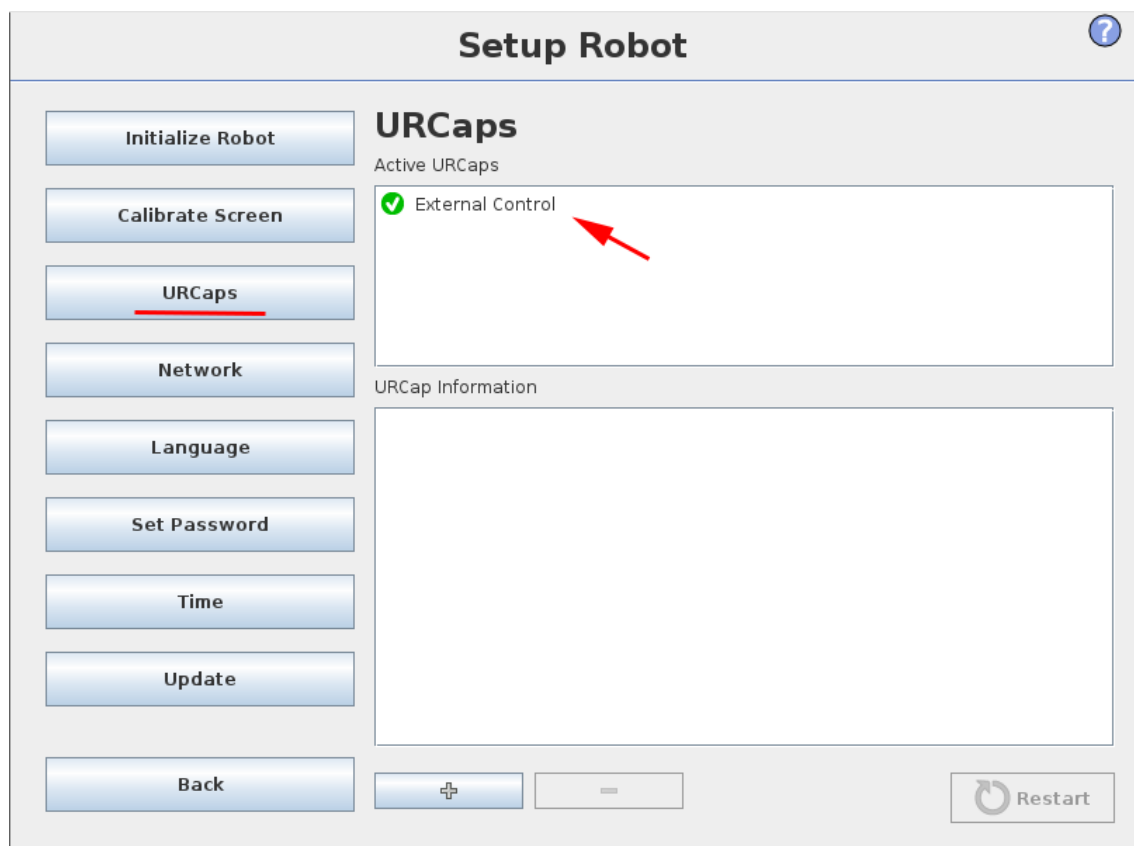
Default gateway and DNS information will probably be applied automatically by the controller.

Go back and click the "About" button again to check the new IP address.

UR Cap: External Control:

The UR controller will need a UR cap for external communication.

From the startup screen you can choose "Setup Robot" and "URCaps" to check if this UR Cap is installed.



If "External Control" is not on the list of active Caps it must be installed.
If the CAP is missing PolyScope will also later show an error message.

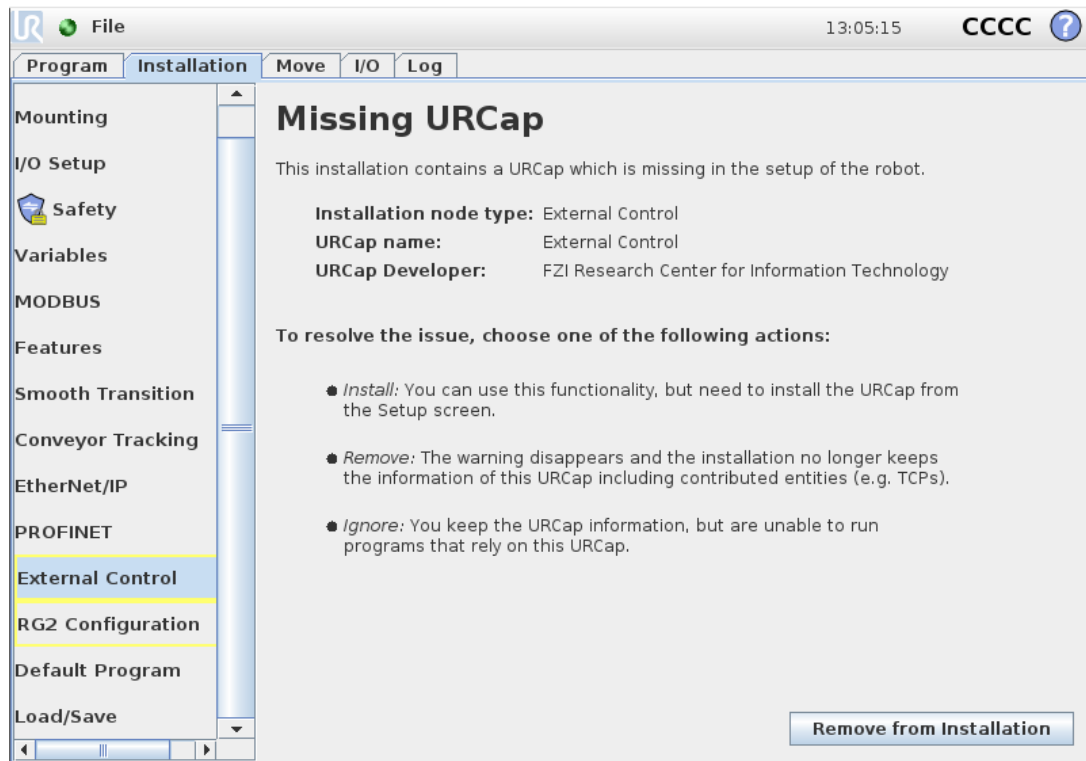
This CAP is free and can be downloaded here:

https://github.com/UniversalRobots/Universal_Robots_ExternalControl_URCap/releases

I'm not sure if this UR cap is necessary for robots in the e-Series.

More information here:

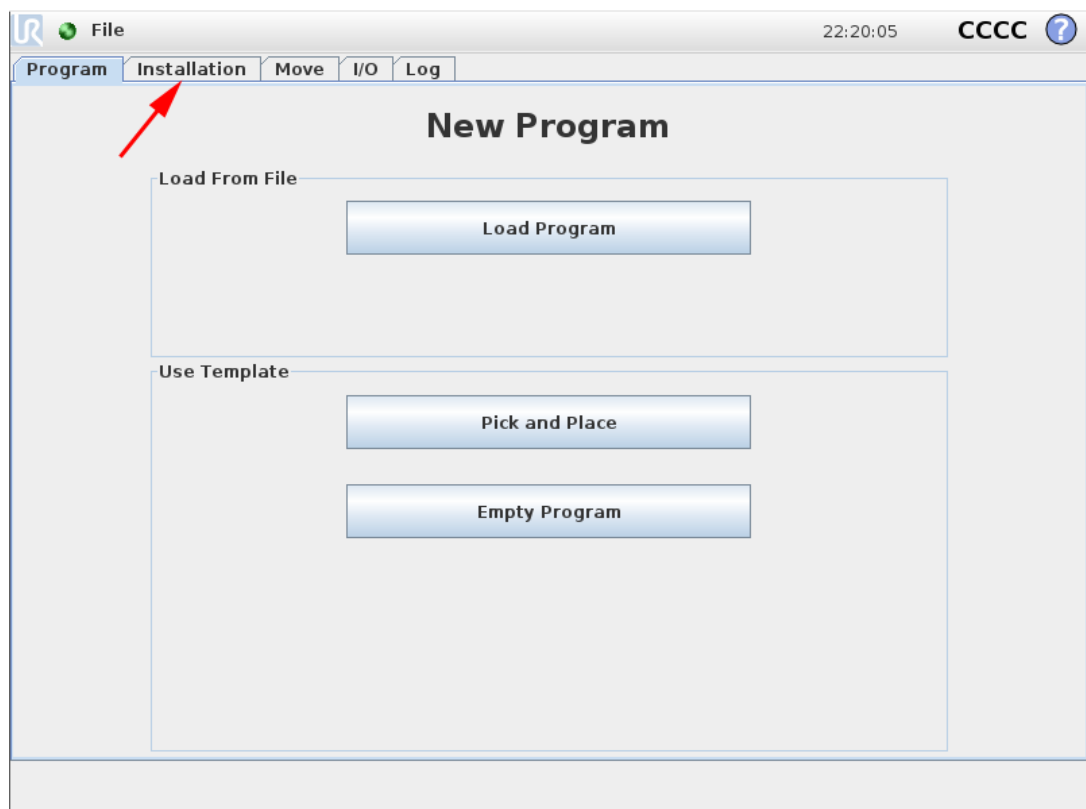
https://github.com/UniversalRobots/Universal_Robots_ROS_Driver/blob/master/ur_robot_driver/doc/install_urcap_e_series.md



Installation:

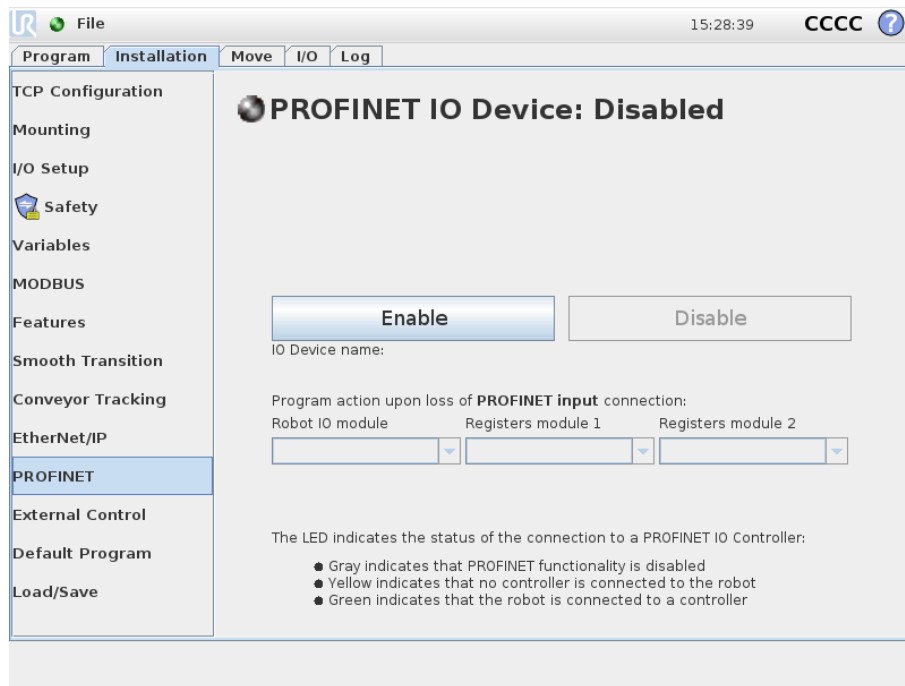
From the start screen press "Program Robot"

You don't need to start a new program or open a program. You only have to go to "Installation". Here you have to do some settings.



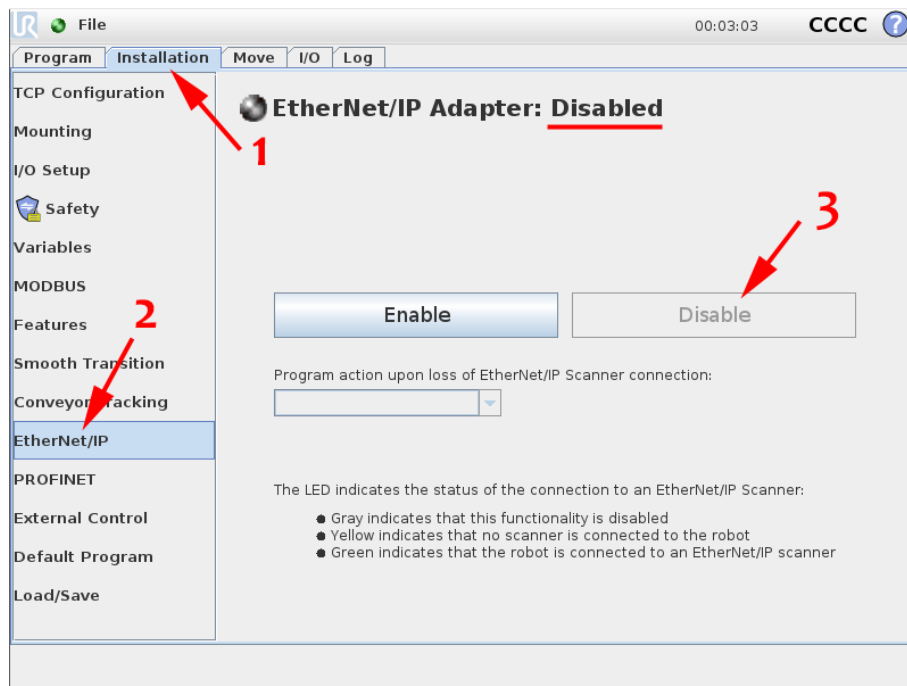
Disable Profinet:

Under the "Installation" in Polyscope you must disable "PROFINET"
The Profi network will block the RTDE communication.



Disable Ethernet:

Under "Installation" and "Ethernet/IP" you must also disable the ethernet/IP adapter. This may seem odd when you try to setup an external connection with a LAN cable to this adapter.



It's a little different for eSeries but this link shows how:

https://github.com/UniversalRobots/Universal_Robots_ROS_Driver/issues/204

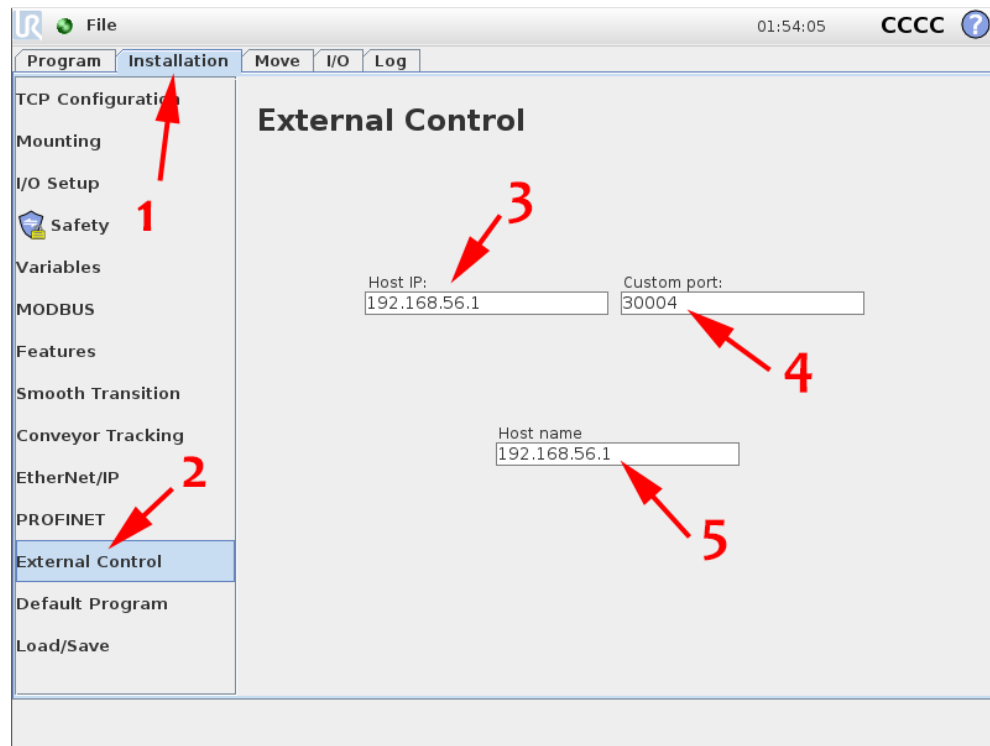
External Control:

To use RTDE the External Control must be setup under "Installation".

The "Host IP" is the IP address for the remote host.

That is the IP address for the net card on the Windows PC.

The "Custom port" must be 30004 for RTDE



Ping the connection:

To check the cable connection before you start the RTDE program you can ping the UR controller from the Windows PC.

Open the commando prompt and ping the IP address for the UR controller.

If the network connection is running you will get something like this:

```

C:\Windows\System32>ping 192.168.56.1

Pinging 192.168.56.1 with 32 bytes of data:
Reply from 192.168.56.1: bytes=32 time<1ms TTL=64
Reply from 192.168.56.1: bytes=32 time<1ms TTL=64
Reply from 192.168.56.1: bytes=32 time<1ms TTL=64
Reply from 192.168.56.1: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.56.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Windows\System32>
  
```


IP reset:

After a restart of the UR controller there is a good chance that the IP addressers have been reset to old values. This is an issue with Linux. But now you know how to set them again.

LAN cable:

RTDE is based on direct LAN cable connection between a computer and the UR controller.

Normally you connect two computers in a network via a router, but it is also possible to connect two computers directly. To do this direct connection you have to use a cross over cable. In this cable the data lines are "crossed over" to feed the output from the first computer into the input on the second computer.

In a network with a router the LAN cables are straight through patch cables. The router will be handling the connection of the send and receive lines.

For the RTDE connection to the UR controller, you must use a patch cable. This is the same cable as you would use to connect a computer to a router. The UR controller must have a build in router or must have been set up to handle the switch over of the incoming and outgoing data.

Data registers:

When you send data to the UR controller these data must be sent to registers where the PolyScope program can get access to them.

The different data registers will accept one type of data. The register you plan to use must fit with the data you want to send.

Check out here:

<https://www.universal-robots.com/articles/ur/interface-communication/real-time-data-exchange-rtde-guide/>

The naming of some of the data registers are a little odd.

In the data transfer program on the Windows computer the ini file set the registers to receive the data.

The input registers for data with double precision are named "double", e.g. "Input_double_1".

When you want to use the data in PolyScope you must assign the data from the register to a variable. But these registers are now named "single"

The data you just send via RTDE to "Input_double_1" must now be loaded from "Input_single_1"

This is a bug in the PolyScope program. The incoming data must be 32 bit (double precision) but in the UR controller treat these data as 16 bit data (single precision). On bit level the 32 bit data is truncated to 16 bit and saved as single precision data.

More info at the UR forum:

<https://forum.universal-robots.com/t/rtde-precision-lost-from-double-to-float-precision/7623/13>

Ready to go:

You should now be ready to go.

I will strongly recommend you to take a look at the videos from SimpleStudent. They will guide you through the setup of RTDE.

The first three videos give a good understanding of how to get RTDE going.

<https://youtube.com/playlist?list=PLnJ9fSRnDN3B1wEuxQY4thTWyGoT2N0yd>

<https://www.youtube.com/@simplestudent279>

**More links:**

<https://pypi.org/project/ur-rtde/>

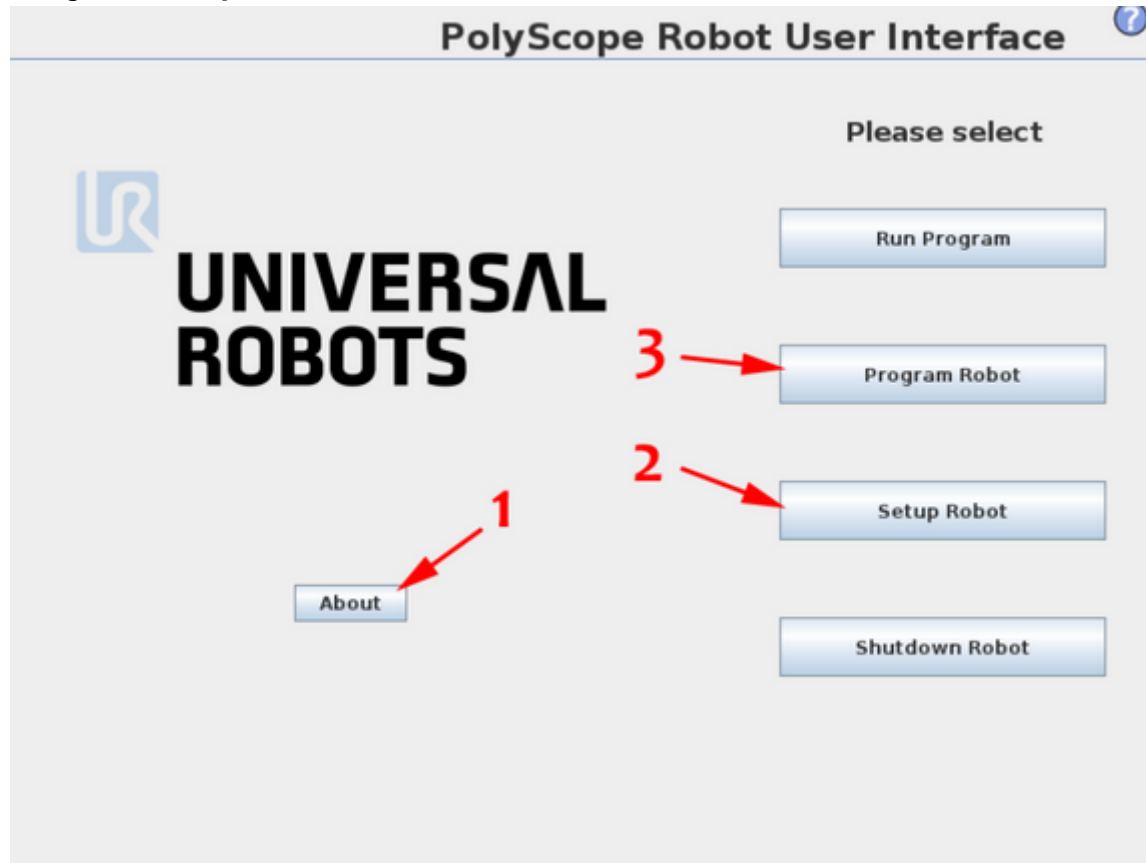
https://github.com/UniversalRobots/RTDE_Python_Client_Library

<https://www.universal-robots.com/articles/ur/interface-communication/real-time-data-exchange-rtde-guide/>

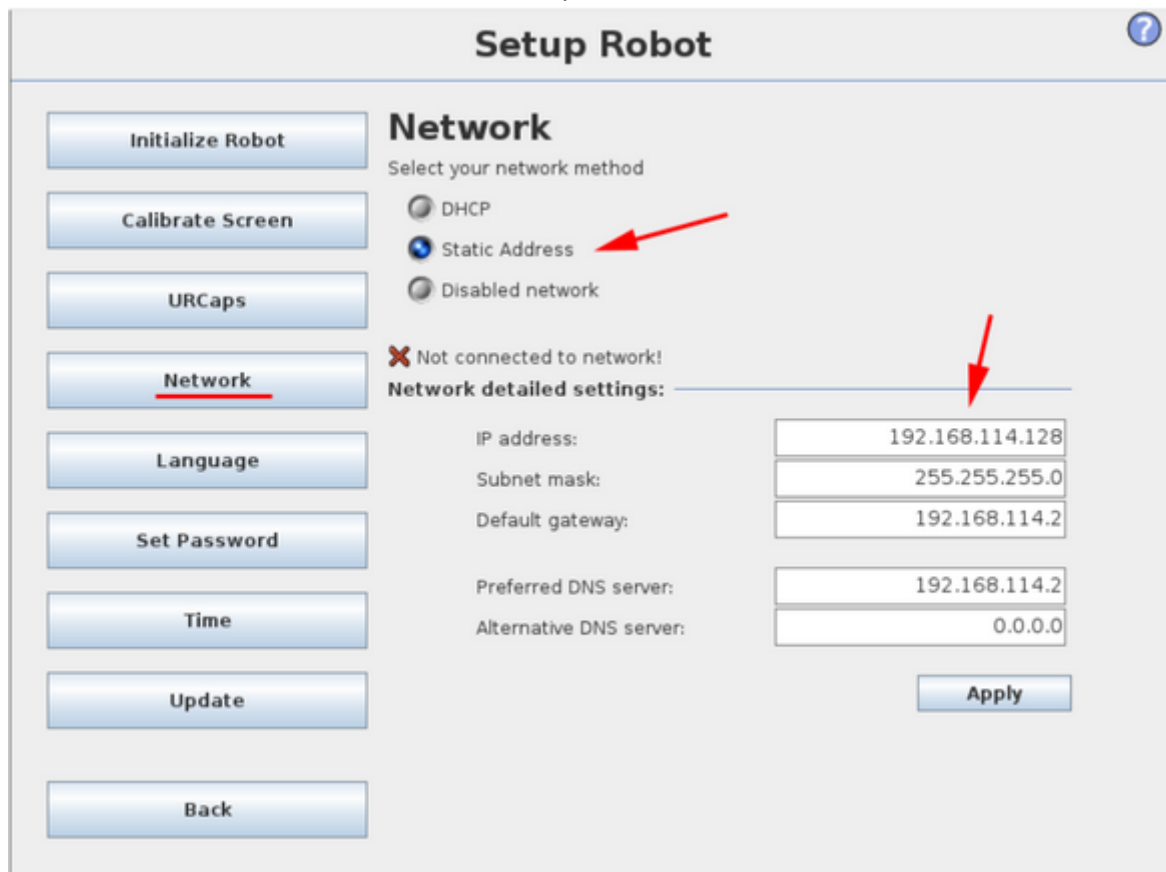
Setting Up RTDE with a UR Robot on Ubuntu

1. Find the IP Address of the UR Robot

- Navigate to **Setup Robot** on the UR controller.



- To use RTDE, the UR controller **must** be set up with a fixed IP address.



- Your **PC's Ethernet connection** must be on the **same subnet** as the robot.
 - Example: If the robot's IP is **192.168.114.X**, your PC should have an IP like **192.168.114.Y**.

2. Check the Connection of Your PC (Ubuntu)

Run the following command to check your network interfaces:

```
ip a
```

Look for your **Ethernet connection**. It may look something like this:

```

ap@pc: ~
TX packets 200  bytes 32435 (32.4 KB)
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ap@pc:~$ ^C
ap@pc:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp7s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 08:97:98:b8:86:62 brd ff:ff:ff:ff:ff:ff
3: wlp0s20f3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 9c:29:76:f5:6d:24 brd ff:ff:ff:ff:ff:ff
    inet 10.126.73.92/19 brd 10.126.95.255 scope global dynamic noprefixroute wlp0s20f3
        valid_lft 7173sec preferred_lft 7173sec
    inet6 fe80::476b:4592:976c:cbde/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:e9:08:52:66 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
ap@pc:~$
  
```

2: enp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000

- The correct interface typically has the "**fq_codel**" keyword.
- **Ensure that no IP address is assigned** to this connection (i.e., no `inet 192.168.x.x` entry).
- **Note the Ethernet ID** (e.g., `enp7s0`).

3. Set Up the IP Address of Your PC

Assign a static IP to your Ethernet interface and bring it up:

```

sudo ip addr add 192.168.1.X/24 dev <ethernet_id>
sudo ip link set <ethernet_id> up
  
```

Example:

```

sudo ip addr add 192.168.1.104/24 dev enp7s0
sudo ip link set enp7s0 up
  
```

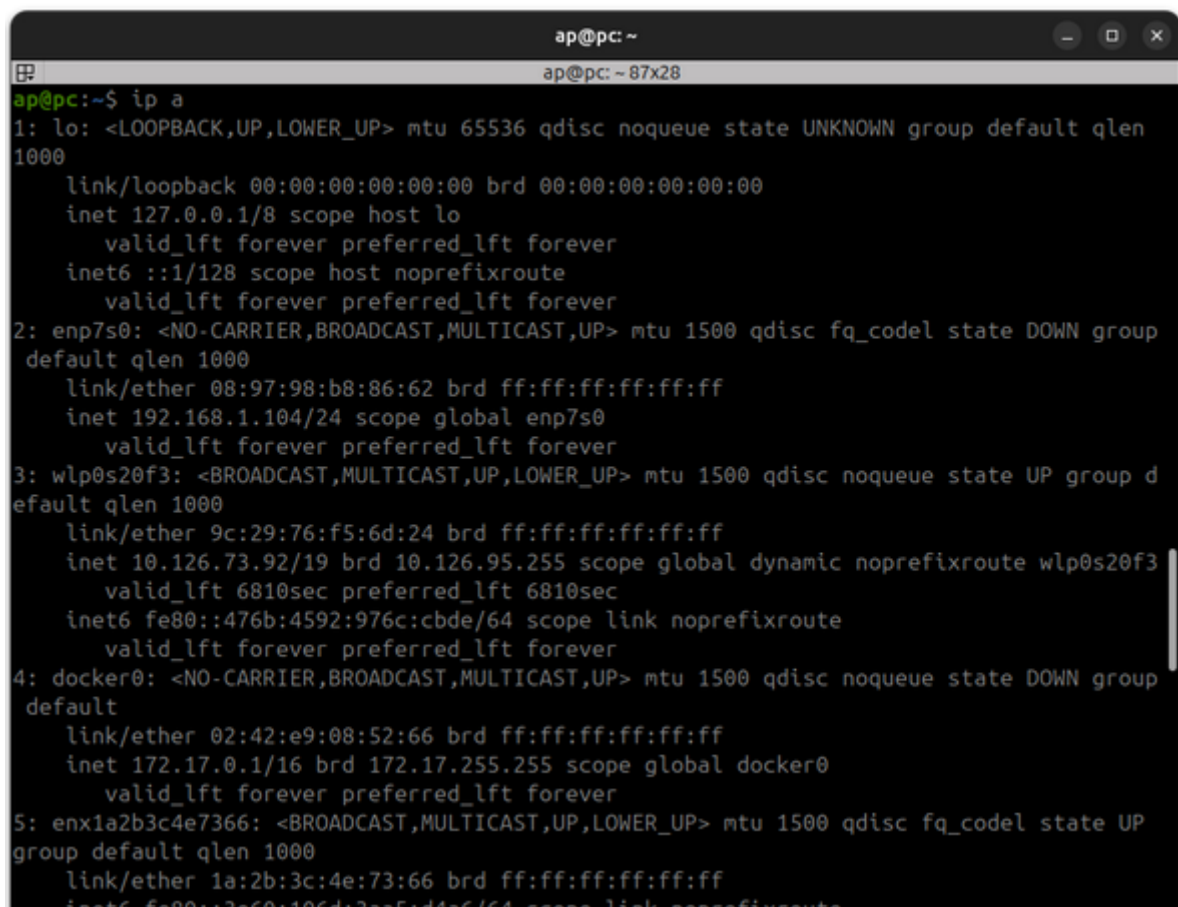
- If the commands run successfully, **there will be no output.**

4. Verify That the IP Address Has Changed

Run:

```
ip a
```

- You should now see an assigned IP, e.g., `inet 192.168.1.104/24`.



```
ap@pc: ~  
ap@pc: ~ 87x28  
ap@pc:~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: enp7s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000  
    link/ether 08:97:98:b8:86:62 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.1.104/24 scope global enp7s0  
        valid_lft forever preferred_lft forever  
3: wlp0s20f3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000  
    link/ether 9c:29:76:f5:6d:24 brd ff:ff:ff:ff:ff:ff  
    inet 10.126.73.92/19 brd 10.126.95.255 scope global dynamic noprefixroute wlp0s20f3  
        valid_lft 6810sec preferred_lft 6810sec  
    inet6 fe80::476b:4592:976c:cbde/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default  
    link/ether 02:42:e9:08:52:66 brd ff:ff:ff:ff:ff:ff  
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0  
        valid_lft forever preferred_lft forever  
5: enx1a2b3c4e7366: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 1a:2b:3c:4e:73:66 brd ff:ff:ff:ff:ff:ff  
    inet6 fe80::3a60:106d:3a35:d436/64 scope link noprefixroute
```

Example:

5. Check the Connection to the Robot

Ping the robot to confirm the connection:

```
ping 192.168.1.X
```

- If successful, you should see a response similar to:

```
64 bytes from 192.168.1.X: icmp_seq=1 ttl=64 time=0.123 ms
```

```
ap@pc:~$ ping 192.168.1.30
PING 192.168.1.30 (192.168.1.30) 56(84) bytes of data.
64 bytes from 192.168.1.30: icmp_seq=1 ttl=64 time=0.287 ms
64 bytes from 192.168.1.30: icmp_seq=2 ttl=64 time=0.200 ms
64 bytes from 192.168.1.30: icmp_seq=3 ttl=64 time=0.204 ms
64 bytes from 192.168.1.30: icmp_seq=4 ttl=64 time=0.252 ms
64 bytes from 192.168.1.30: icmp_seq=5 ttl=64 time=0.203 ms
64 bytes from 192.168.1.30: icmp_seq=6 ttl=64 time=0.230 ms
```

6. Check if RTDE is Working

Check if Ports Are Open

Run:

```
nc 192.168.1.X 30002
```

Expected output: A connection confirmation.

```
ap@pc:~$ nc -vz 192.168.1.30 30002
Connection to 192.168.1.30 30002 port [tcp/*] succeeded!
```

Check the Robot State

Run:

```
nc 192.168.1.X 29999
robotmode
```

Expected output: The **robot mode** based on its current state.

```
ap@pc:~$ nc 192.168.1.30 29999
Connected: Universal Robots Dashboard Server
robotmode
Robotmode: POWER_OFF
```

This should get your UR robot and RTDE connection properly configured! 🚀