

Nao User Manual (6/27): Contents

Intro

Two Nao robots, “Jonson” and “Mörner”, at Halmstad U. are the property of Walid.

He has kindly lent these robots to others to use on some occasions. (There are apparently also two other robots being used at RICE U.; the names seem to have been based on those of a well-known group of four artists who were active in Halland in the 1920s-30s) The robots use different versions of the driver/software—Jonson 1.14.5, and Mörner 1.12.5—so you should know which one you are using.

Note: On the robot I borrowed, Jonson, its name was written on its shoulder and on its case.

The software for Jonson is installed on a computer: IDE8192, a Dell Latitude E6400 running Windows 7. It can also be downloaded from the internet if you register for an account.

There appears to also be a CD with the older software for Mörner somewhere, possibly in a box or at the e-lab with Thomas Lithen. The rest of this guide assumes you have been able to borrow Jonson and IDE8192. The process if you download/install is similar.

Getting Jonson to move using IDE8192/Choregraphe

*Log in to IDE8192. You may need to ask someone who knows the account name and password for logging in: e.g., Martin or Nicolina.

*Take the robot out of its case and attach its battery to its back.

*Connect the robot to IDE8192.

You can run an Ethernet cable from the slot on the back of the robot’s head either directly to IDE8192 or into a wireless router.

If using wireless, turn on the router and connect IDE8192 to the wireless network “robots”; you may have to ask someone for the password, e.g., Martin or Roland.

*Turn on robot by pressing the button on its chest. It takes a few seconds, then the robot should say something: e.g., its name and battery state.

(You can press its chest button again to hear its IP address, but it’s probably not necessary as Choregraphe automatically finds and displays this information).

*Put the robot somewhere where it can move safely without falling or damaging anything.

*Start the software.

The software to use for creating motion files is called “Choregraphe”.

On the student account, there is a shortcut on the desktop. (Else you can start it from Program Files(x86)/Aldebaran/ (I think the bin folder for the .\student account))

*Select free trial and continue (if the free trial has expired you may need to register for another)

*Click in the upper left menu item to open a recent motion file to test the robot.

On the student account several motion files can be selected, which are on the Desktop.

*Press “Connect” (the green wireless button)



Nao chest button



Ethernet (top) and battery (bottom)

In the dialogue box, Jonson's information should be displayed. Select it, and click on Connect.

Note if you right-click you can perform some operations.

Test LEDs makes the robot's eyes changes from cyan to green for a second.

You can also connect to the robot's homepage (user: Nao, pwd: Nao) where you can change its name, its volume setting, and check its IP address etc.

*If there is a problem, an error message will be displayed stating that "The connection with NAOqi is lost. Trying to reconnect automatically in ..."

You will need to do some trouble-shooting. For me, I was able to solve this problem by asking the IT Department to turn off Windows Firewall, which was blocking the software.

*Press the "Run" button (the arrow) to get the robot to move.

A small dialogue box will appear saying that the program is being sent to the robot.

While the robot moves, you may wish to be careful that it does not fall down or damage anything.

*You can press the stop button (the square) to get the robot to stop, and also toggle the volume on the upper right.

*When finished, press the disconnect button.

*Turn off the software, the robot, the computer, and the wireless router if you are using it.

Congratulations! You have (I hope) successfully commanded the Nao Robot!

NAO SDK C++

*The following guides are based on:

https://community.aldebaran.com/doc/2-1/dev/cpp/install_guide.html

*Original information is in Courier New.

Added steps I needed are in Calibri.

*For all of this I assume you are logged into an administrator's account.

*The SDK can be installed on Ubuntu or Windows. Windows was harder for me, therefore the guide below is for Windows. Ubuntu was similar but easier.

NAO SDK C++ WINDOWS GUIDE

*For Windows you need to add a few entries to your system path. You can do this as follows.

Go to "Computer" – "System Properties" – "Advanced" – "Environmental Variables"

In the second textbox called "System Variables", find "Path", edit, and append.

Notes:

*Supported Operating Systems

Microsoft Windows XP Service Pack 3, Microsoft Windows 7 and 8

*Windows limitations

No C++ cross-compiler.

The C++ cross-compiler (used to compile code to be run on the robot) is not available on Windows.

I don't think this is an important demerit. The Windows SDK on my computer was able to compile a hello world program and get the robot to execute it, which is usually what you want to do.

*Windows architecture: it is mandatory to compile for 32-bit. The resulting code will run as well on 32-bit or 64-bit environment.

Installation Steps:

A - Compiler and IDE Installation

B - Cross-platform build system Installation

C - SDK Installation

D - qiBuild Installation

E - Compile and run an example

A - Compiler and IDE

Windows Visual Studio 2008 or Visual Studio 2010 is required.

I think VS 2008 Express may have been discontinued because I could not find anywhere on the Microsoft site to download. VS 2010 is available.

On my computer installing VS 2010 failed with one install file but worked with another.

B - Cross-platform build system

Required: CMake, Python 2.7, pip, qiBuild.

CMake generates makefiles and workspaces for any operating system, allowing a project to be compiled on Windows, Mac, Linux and NAOqi OS.

qiBuild is a tool designed to generate cross-platform projects using CMake.

Install CMake version 2.8.3 or higher from:

<http://www.cmake.org/cmake/resources/software.html>

When prompted at the end of the installation, choose to add CMake to your PATH.

If you already have CMake installed but have not added it to your Path, you need to do it manually as described above.

On my computer the location of CMake was "C:\Program Files (x86)\CMake 2.8\bin;"

Install Python 2.7

You have to also add this to your Path as described above: "C:\Python27;C:\Python27\Scripts;"

C - SDK Installation

The link on the documentation page showed none of the necessary downloads for me.

I emailed the support staff to get the link below which does have them

<https://community.aldebaran.com/resources/archives/>

The one you want for windows is:

naoqi-sdk-1.14.5-win32-vs2010.zip

Extract it on your machine (e.g. to naoqi-sdk).

D - qiBuild Installation

Download pip

Right-click the link to save as at

<https://pip.pypa.io/en/latest/installing.html#install-pip>

or get it from

<https://bootstrap.pypa.io/get-pip.py>

Run it :

```
python get-pip.py
```

Upgrade it:

```
python -m pip install -U pip
```

```
pip install qibuild
```

Configure qiBuild

Run:

```
$ qibuild config --wizard
```

Result: a file is generated in ~/.config/qi/qibuild.xml. It is shared by all the worktrees you will create.

You will be prompted to specify:

CMake path (if not automatically found),

CMake Generator and

the IDE you will use.

Use 'Visual Studio' on Windows.

Note that you can still re-run the config wizard anytime you want.

On my computer the following was output:

```
Found CMake: C:\Program Files (x86)\CMake 2.8\bin\cmake.exe
```

```
:: Please choose a generator
```

```
 1 Visual Studio 6 (default)
```

```
 2 Visual Studio 7
```

```
 3 Visual Studio 10
```

```
 4 Visual Studio 11
```

```
 5 Visual Studio 12
```

6 Visual Studio 7 .NET 2003
7 Visual Studio 8 2005
8 Visual Studio 9 2008
9 Borland Makefiles
...

I selected 3 in hope that this would match VS 2010 express, and then 1 for IDE, which was “none (default)”.

E - Compile and run an example

1.

Create an EMPTY folder where you will want to store your C++ SDK
I chose “C:\nao_sdk\
Open a Command Prompt and go to that folder.
Type this command in order to create a worktree.
\$ qibuild init

2.

Then copy the folder naoqi-sdk into your SDK folder
Enter that folder and go to examples folder.
\$ cd /path/to/SDKfolder/naoqi-sdk/doc/dev/cpp/examples

3.

Type this command to create a default toolchain using the feed from
the C++ SDK:
\$ qitoolchain create mytoolchain /path/to/SDKfolder/naoqi-
sdk/toolchain.xml --default

4.

Type these commands in order to configure and build the
sayhelloworld project:
\$ qibuild configure sayhelloworld
\$ qibuild make sayhelloworld

5.

You can now run the resulting binary from: sayhelloworld/build-
mytoolchain/sdk/bin

On my computer this was not easy to find. The binary was in “C:\nao_sdk\naoqi-
sdk\doc\examples\core\sayhelloworld\build-mytoolchain\sdk\bin”

Also, on my computer, an error message was shown saying a dll could not be found.

If this happens to you, add this to your path: \path\to\SDKfolder\naoqi-sdk\bin

On my computer I needed to add: “C:\nao_sdk\naoqi-sdk\bin”

Open the cpp file and change the line that the robot says to anything you would like it to say.
Congrats, you have used the Nao C++ SDK!

NAO SDK C++ UBUNTU GUIDE

Follow the guide at

https://community.aldebaran.com/doc/2-1/dev/cpp/install_guide.html

Download the right sdk at

<https://community.aldebaran.com/resources/archives/>

For me it was "naoqi-sdk-1.14.5-linux32"

Download pip

Right-click the link to save as at

<https://pip.pypa.io/en/latest/installing.html#install-pip>

or get it from

<https://bootstrap.pypa.io/get-pip.py>

Run it :

```
python get-pip.py
```

Upgrade it:

```
pip install -U pip
```

Install qibuild:

```
pip install qibuild
```