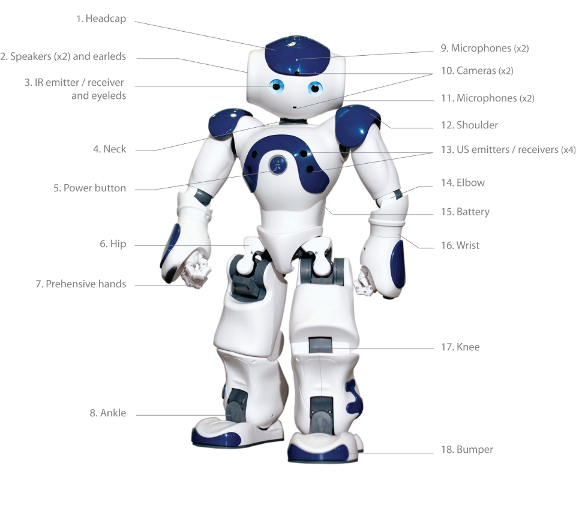
Nao



Contents

1. Ways to work with the Nao 3

1.1. With Choregraphe only 3

1.2. Through SSH connection 3

1.3. Using the SDK 3

# Ways to work with the Nao

First read <http://doc.aldebaran.com/2-1/dev/naoqi/index.html> for a good introduction on the NAOqi framework.

Summary: NAOqi is a framework with many modules (shared objects) with functionality like motion, vision, audio, memory etc. They can be accessed by creating a proxy to this module first. The proxy can run on the robot or remote on a laptop. The proxy will connect through TCP/IP with a broker running on the robot. The broker will have the list of available libraries loaded (in ‘autoload.ini’). The libraries contain modules and methods and the broker will pass the call to the prober library. There are two brokers: mainBroker() for local calls and myBroker() for remote calls.

## With Choregraphe only

In Choregraphe a Nao behavior is created by using boxes and connecting them. One can use existing boxes or create new ones. The types are: Diagram, Timeline, Python or Dialog box. A box can contain classes derived from ‘GeneratedClass’ with standard methods such as ‘onInput\_onStart()’ or ‘’ onInput\_onStop(). When the behavior executes the classes are instantiated and the box methods are called when the input is stimulated. The underlying event mechanism is configured in Choregraphe when connecting two boxes.

One can write a Python script which makes calls to the Nao API through proxies. Example call to obtain a text to speech proxy: ‘tts = ALProxy("ALTextToSpeech")’. Note that in this case no IP address and port is required as the the Python module and the ALTextToSpeech module are in the same broker. This proxy can then be used to make subsequent calls to the text to speech API, for example: ‘tts.say("hello World!")’.

Debugging the Python script can be done printing to the Choregraphe Log viewer like: ‘self.logger.info("\*\*\*\*\*\* This is a info message \*\*\*\*\*\*")’.

## Through SSH connection

With e.g. ‘ssh nao@192.168.1.137’ one can connect with a laptop to the Nao. Using FileZilla it is possible to transfer files between the laptop and the Nao. The home directory is ‘/home/nao’. The behaviors are located in ‘/home/nao/.local/share/PackageManager/apps’. Behaviors are in ‘.xar’ files which are ‘.xml’ files containing Python code within <content> tags.

One can write a Python script which makes calls to the Nao API through proxies. Example call to obtain a text to speech proxy: ‘tts = ALProxy("ALTextToSpeech", “localhost", 9559)’. This proxy can then be used to make subsequent calls to the text to speech API, for example: ‘tts.say("hello World!")’.

## Using the SDK

Installing the SDK on a laptop gives the most freedom in working with the Nao. It is possible to develop behaviors in languages like C++, Java or Python. The behavior can be run from the laptop. This means the Nao is controlled from the laptop which makes calls to the Nao API through proxies. Example call to obtain a text to speech proxy: ‘tts = ALProxy("ALTextToSpeech", “192.168.1.137", 9559)’. The IP address and the port number are of the remote broker ‘myBroker’.

At the moment OS X El Capitan is not supported.