K.Anter | M.Greiner | J.Vatter | J.Weghake
Direction: H.Schnaubelt

SoftBank Robotics

NAO 6

LEARN IT

TECHNIK LPE

sprachliches + naturwissenschaftlich-technologisches
bertha-von-suttner
gymnasium neu-ulm

THE BASICS

Learn it NAO[6] - The Basics
1st edition 2019

Authors: Kai Anter, Marcel Greiner, Jonas Vatter, Jannes Weghake
Direction: Heike Schnaubelt (OStR, senior councilor of studies)
Editor: Tony Schuster (StR, councilor of studies)
Program contribution: Caspar Sachsenmaier

All program examples and screenshots for NAO[6] were tested and taken in the environment Choregraphe (Version 2.8.X).
It is possible to transfer the content to earlier NAO generations, however their correct functioning cannot be guaranteed. The same applies the other way around.
In this book, "NAO" always refers to the sixth generation of NAO.

TECHNIK LPE

Technik-LPE GmbH
info@technik-lpe.com
www.technik-lpe.de

Friedrichsdorfer Landstraße 64
69412 Eberbach / Germany
Phone: +49 6271 944650-1
Fax: +49 6271 944650-2

We,

the team of authors comprised of **Kai Anter**, **Marcel Greiner**, **Jonas Vatter**, and **Jannes Weghake**, under the direction of **Heike Schnaubelt (OStRin)**, are pleased to present our book
"Learn it NAO⁶ - The Basics".

It was created as part of our study skills seminar "The real and virtual STEM classroom of the future" in cooperation with the company **Technik-LPE GmbH**.

Our target group:

Anyone who is interested – anyone at all!

We have set ourselves the following goals for the book:

- NAO⁶ is intended as an ideal introduction to humanoid robotics.
- This book, NAO⁶ with a laptop/notebook, and the required materials are all you need to learn about NAO⁶.
- We want to spark interest in the STEM field through playful exploration.

We will achieve these goals through:

- Clear illustrations
- Step-by-step instructions
- Practical exercises with suggested solutions

We hope we can inspire you to explore this new universe and that this book will be of help to you in doing so.

Have fun discovering this new NAO-verse!

Your author team

Kai Anter | Marcel Greiner | Jonas Vatter | Jannes Weghake

A. NAO

A.1. Tips and tricks

Here are several tips we have gathered while working with NAO. They should help you to use your NAO for as long as possible without any incidents.

1. Do not work with your NAO on tables or other raised surfaces.
2. Disconnect the Ethernet and charging cables from NAO after you have transferred the program (risk of tripping).
3. Charge NAO while it is not in use.
4. Do not set the speed of the NAO movements above 80% as it could overbalance (recommended values: 60% - 70%).
5. Always let your NAO move on an even and stable surface, otherwise it is more likely to fall over.
6. Do not lift up NAO while it is sitting as this triggers the Fall Detection. (The Fall Detection is a procedure that is always active on NAO. If NAO falls, it prevents serious damage. NAO goes into a protective position very quickly. Your fingers could become trapped when this happens.)
7. Only move NAO by holding it under the arms in a standing position.
8. Do not leave NAO unattended for a longer period of time.
9. Use the Pose Library with caution. If you want NAO to stand up, use the Stand Up box. (The Pose Library is a collection of several positions that NAO can assume. You will learn about the Stand Up box later in C.2.1.5.)
10. Only store NAO in the styrofoam box or the transport case that is available separately, or lean it against a wall with a piece of fabric in between (risk of tipping, damage).
11. When NAO is switched off, the battery needs around 2 hours to charge from 0% to 100%, when switched on around 2.5 hours.
12. It needs around 2-3 minutes to start up.
13. Keep the styrofoam protection provided as storage for NAO.
14. If you work with Choregraphe 2.8.X, only use NAO[6]. If you work with NAO[6], only use the latest version, Choregraphe 2.8.X.

A. 2. Sensors

Interacting with, recognizing, and understanding surroundings is an important fundamental skill for humans. And depending on the area in which they are used, robots need to be able to do this, too. As NAO is a humanoid robot, or one that looks like a person, it should also "function" like a human. To enable it to do this, several sensors and actuators are built into NAO. This chapter briefly explains where some of them are installed.

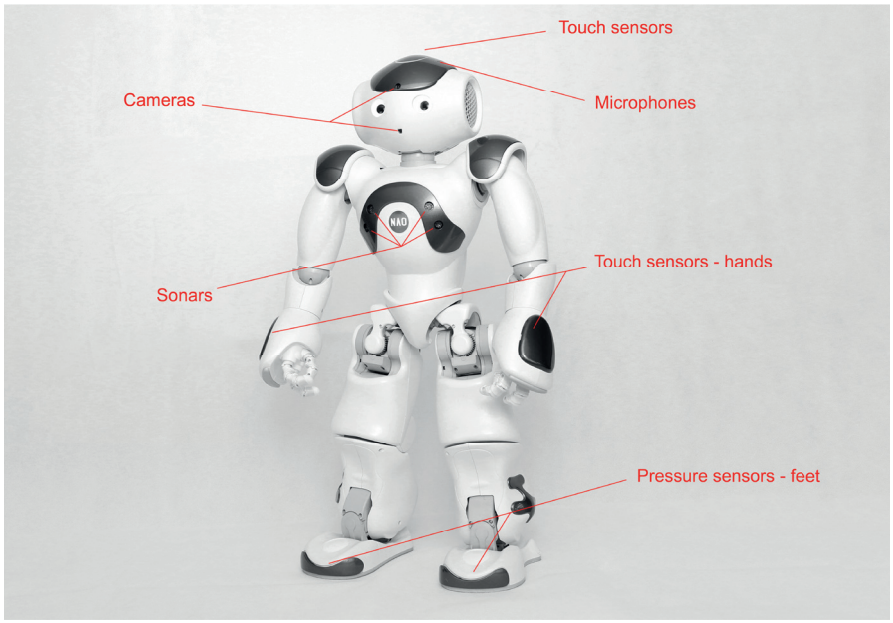


Fig. 1 – Brief overview of NAO's sensors

NAO's head

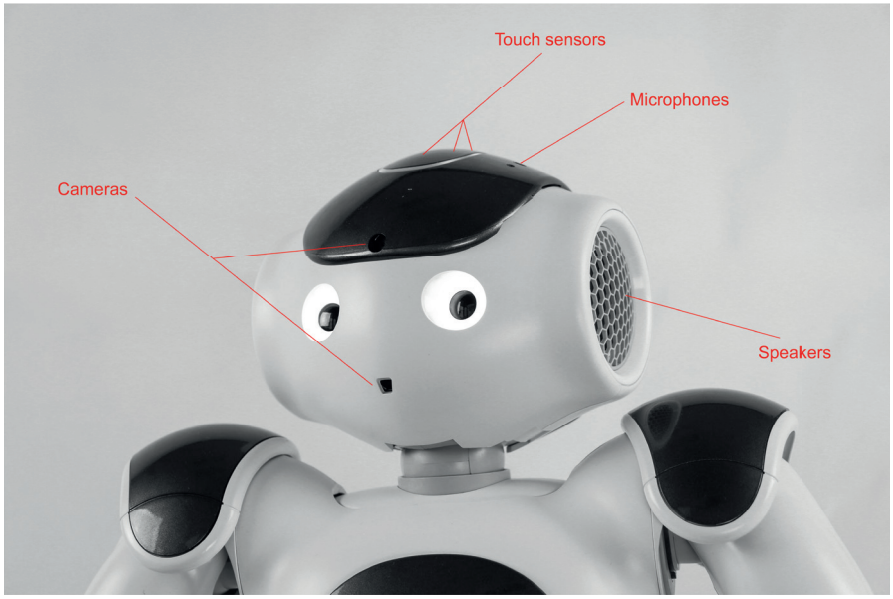


Fig. 2 – NAO's head

On its head, NAO has two very important sensors, two cameras, and four microphones. Plus, there are two speakers fitted. Finally, there are the tactile, or touch, sensors on the top of its head. They can be controlled using a box (you will learn what a box is in C.1.) in Choregraphe (see Tactile Head box, C.2.1.7).

NAO's chest

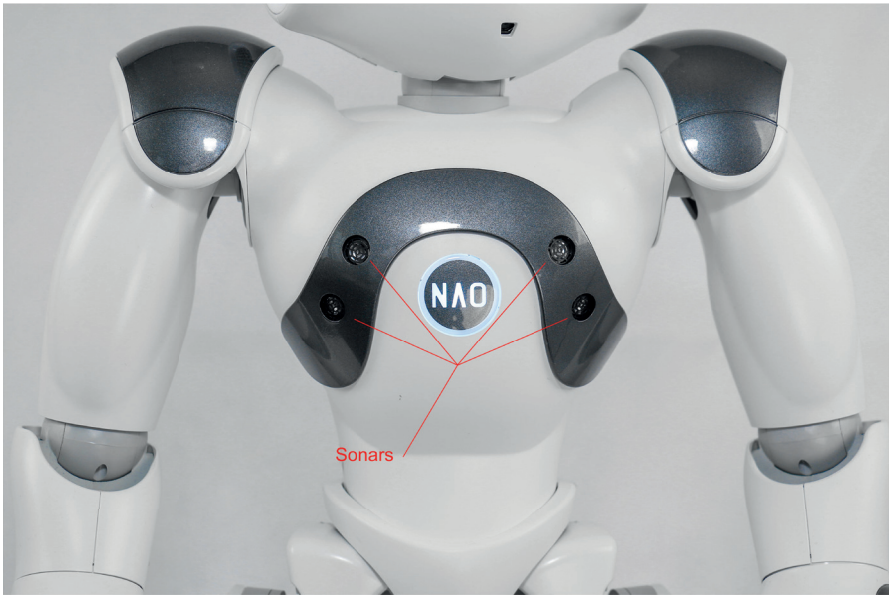


Fig .3 – NAO's chest

On its chest are four ultrasonic or sonar sensors. They can measure distances from walls, people, or objects.

NAO's hands

Fig. 4 – NAO's hands

There are three tactile sensors installed in each hand, which can also be controlled using a box (see Tactile Hands box, C.2.1.8).

NAO's feet

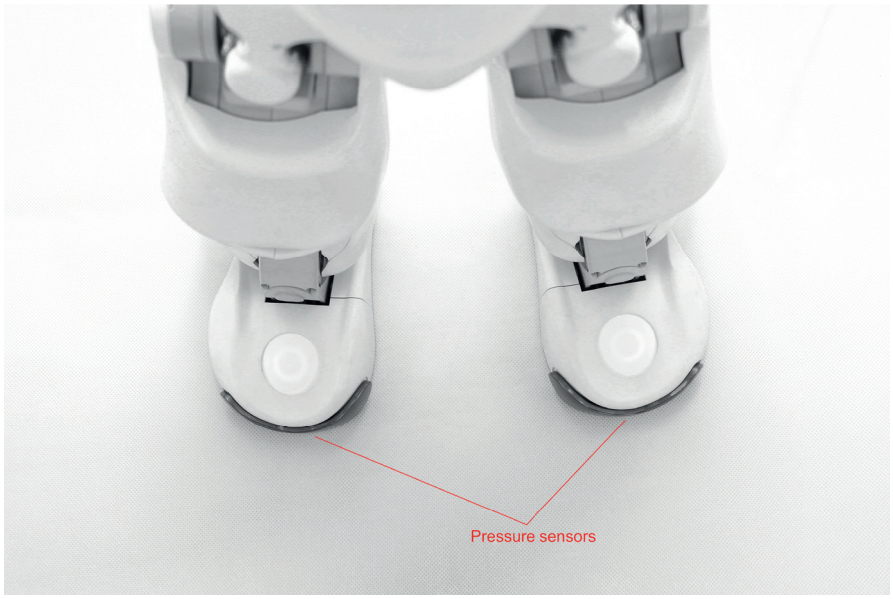


Fig. 5 – NAO's feet

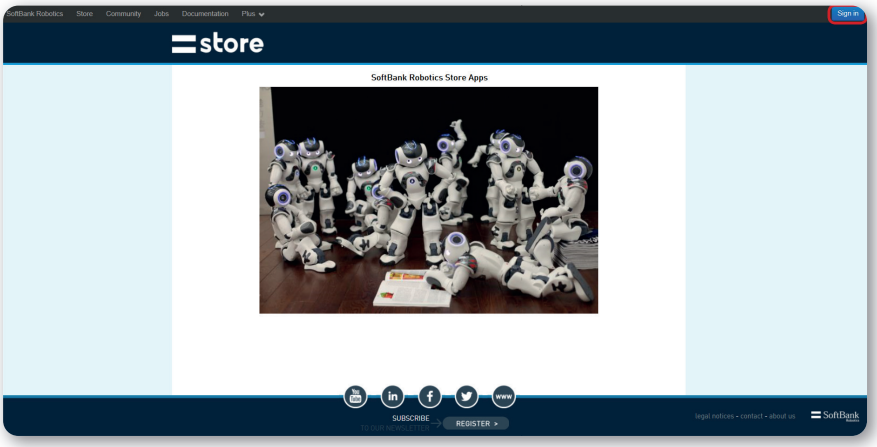
NAO has one pressure sensor on each foot. The term for these sensors is "Bumper". So if you come across this word, you'll know what it refers to.

A. 3. Ready-made programs

Particularly difficult programs are complicated and time-consuming to create. To ensure that you still have fun with your NAO, several ready-made programs are available on the Internet. You can download them and install them on your NAO. Programs like this are called applications, or apps. There are also apps that have several functions. These "big apps" are called channels.

One application is the "Walk Together Demonstration". You can use it to direct NAO to any position you like. Here's how to install this application as an example. Other applications or channels are installed in the same way.

First go to the website "***https://cloud.aldebaran-robotics.com/***".

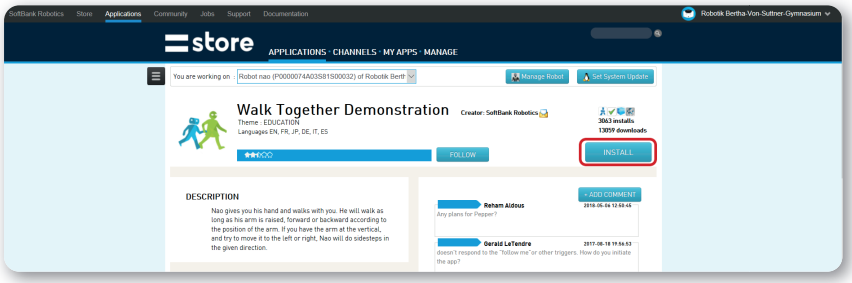


Screen 1.1 – SoftBank store

Then click the Sign in button to sign in. You are taken to this page:

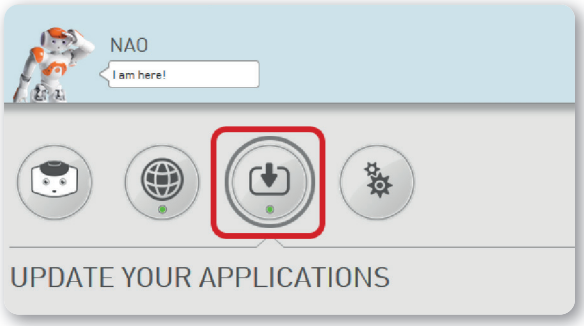
Screen 1.2 – SoftBank applications

Choose an application you like the look of and click on it. (The apps are sorted into three categories: Best rated, New apps, and Selected for you.)



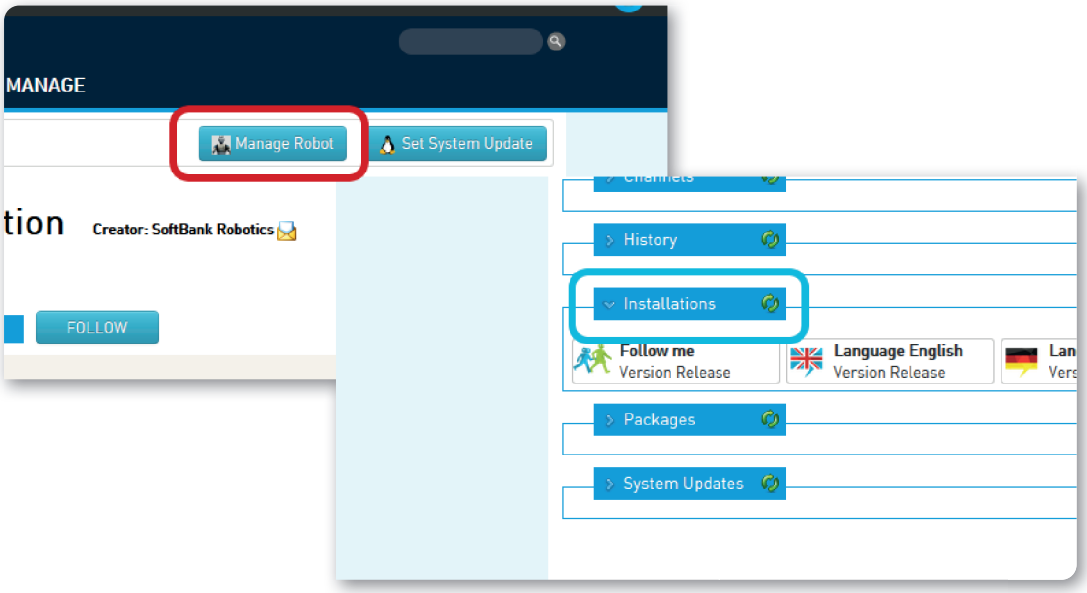
Screen 1.3 – Walk Together app

To install the application, simply click on Install and the NAO displayed in the top middle (your NAO) will be equipped with the selected application.



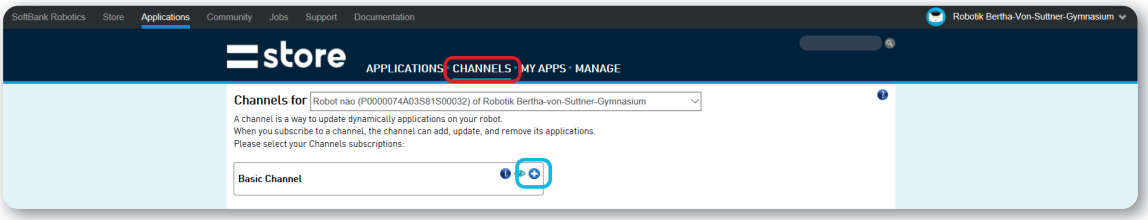
Screen 1.4 – Update application

To properly install it on the robot, you have to download the files to NAO on the robot page under Update application.



Screen 1.5 – Manage robot

To check whether NAO has saved the application, you can click on Manage Robot and view and uninstall all installed content under Installations.



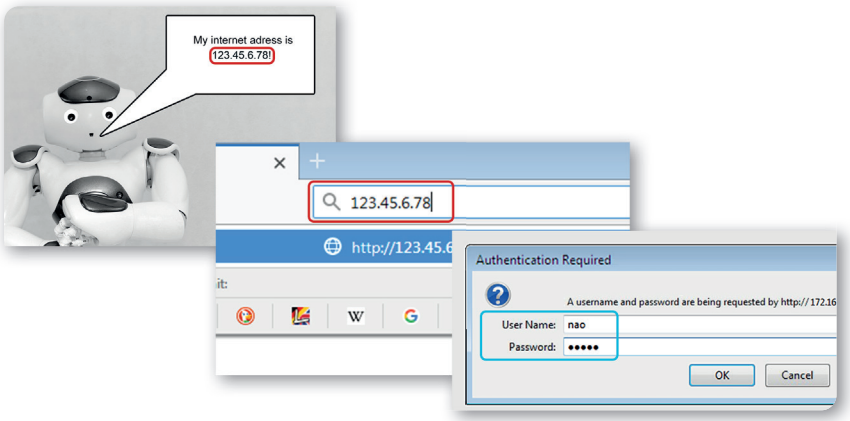
Screen 1.6 – SoftBank channels

To subscribe to a channel, go to the Channels tab.
Click on the blue plus symbol and you are subscribed to the channel.

The difference between a channel and an application is that a channel consists of several applications and updates itself automatically as soon as it connects to a WIFI network. The only channel available so far is the “Basic channel”: This adds dialog skills (you will learn what a dialog is later in C.3.1.1) and reactions to touch. It is necessary for making use of the full potential of Autonomous Life. (Autonomous Life is a behavior of NAO. It involves several programs running independently. How it works is explained in B.3.)

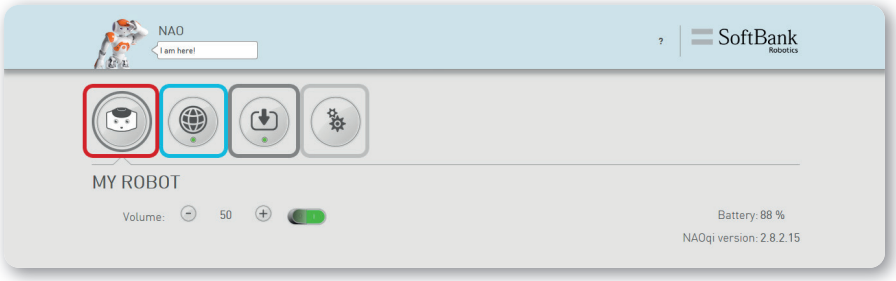
B. First steps

B.1. Robot page



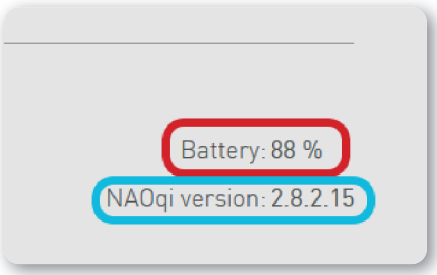
Screen 2.1 – How to get to the robot page

You can find the robot page by pressing the chest button on your NAO once, entering the IP address in your browser, and signing in with your username and password.



Screen 2.2 – Robot page

You are taken to this page. Here you can manage the volume, WiFi connection settings, applications, account settings, language, time zone, and Autonomous Life.

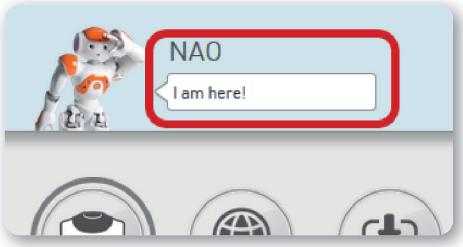


Screen 2.3 – Battery level

Here you can call up information about the battery level and current NAOqi version.

Screen 2.4 – Settings

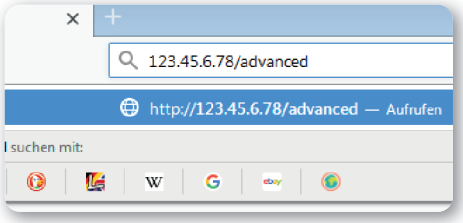
It is also possible to turn off, reboot, and perform a factory reset for NAO here.



Screen 2.5 – Speech bubble

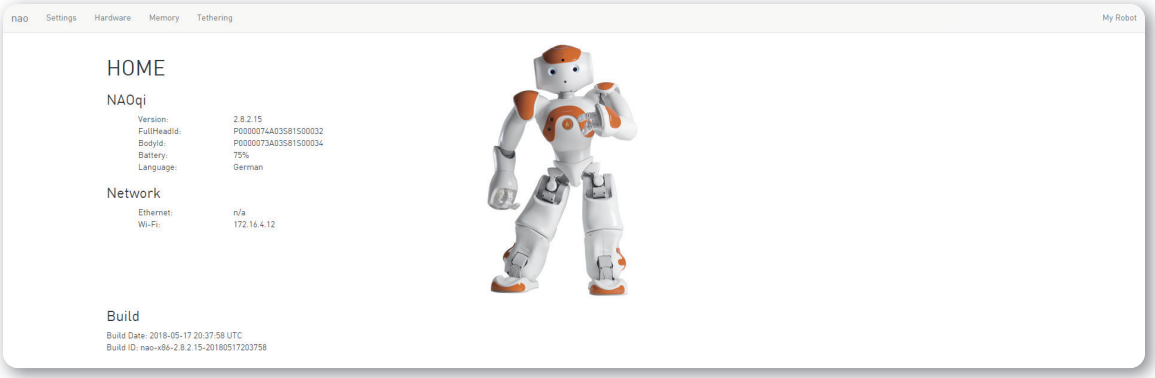
In addition, you can use the speech bubble at the top left to get NAO to say something directly via text input.
There is another website for more detailed information about NAO.

However, this website is intended more for the purpose of analysis, as in the worst case you can make NAO unusable by changing the settings.



Screen 2.6 – Advanced IP address

To go to this page, type "**/advanced**" after the IP address after signing in.



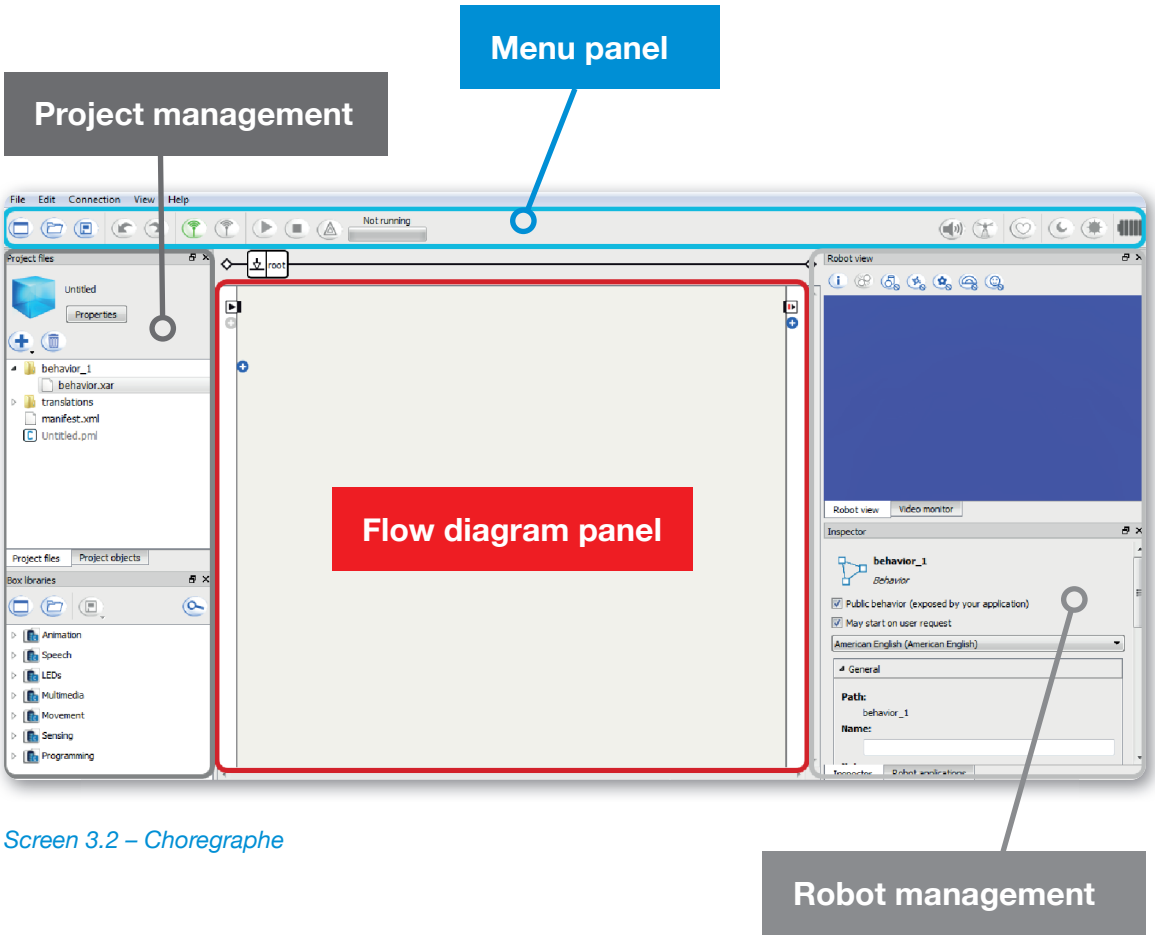
Screen 2.7 – Robot page, advanced

Here you can switch off the fall manager reflexes, see the serial numbers of the parts fitted, call up current temperatures, view the memory, and manage Internet settings.

B. 2. Choregraphe

Screen 3.1 – Choregraphe symbol

Choregraphe is the most important program you use when working with NAO. With Choregraphe you can create programs, write dialogs, or set NAO's behavior. However, Choregraphe provides you with many more options, too. For example, you can have it display what the camera sees, change settings like volume and language, create a preview for your program, or change the source code for boxes. As the interface shows all these different areas, here is an overview for you. Individual components are explained in more detail at the end.



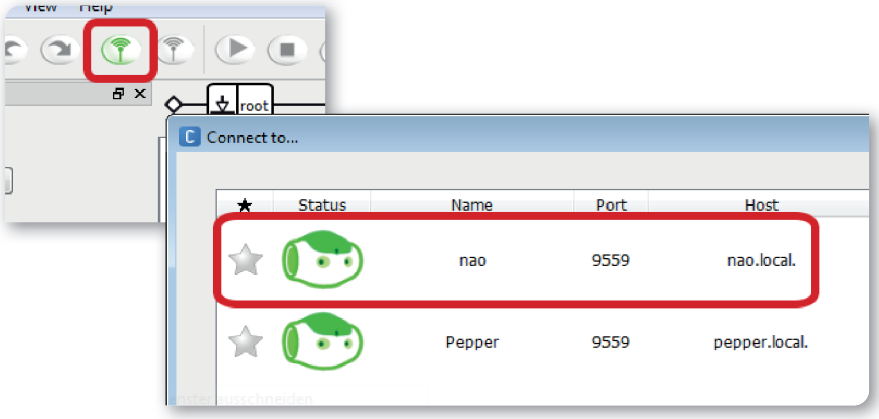
Screen 3.2 – Choregraphe

Choregraphe can be split into these sections:

The toolbar, left side:

Screen 3.3 – Choregraphe: Top toolbar, left side

Here you can open or save your project or create a new project, just like in other programs. You can also undo actions. Most important, however, are the interaction options with the robot. First of all, you can connect it to the computer.



Screen 3.4 – Connect to the robot

Click on the green antenna and select the robot. (If you have problems connecting, restart the robot, your computer, and/or Choregraphe and try again.)

Screen 3.5 – Choregraphe: Top toolbar, left side, detail

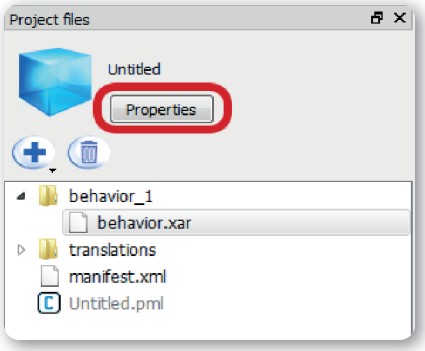
When a connection is established (it doesn't matter if it's a simulated robot or a real one) you can start your program with the green play button and stop it with the red stop button. If there is an error in your program, the red warning triangle is displayed. The progress bar shows the progress when loading a program (when the bar is completely green, the program is running).

The toolbar, right side:

Screen 3.6 – Choregraphe: Top toolbar, right side

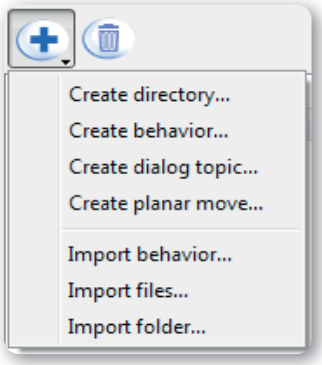
You can use these buttons to control your NAO. Here you can activate the Animation Mode, which allows you to move NAO into any position you like manually. You can also put NAO into the Autonomous Life mode here. Plus, you can change the volume or switch between the rest and active modes. Finally, if a connection is established a battery indicator shows you how full NAO's battery is.

The project files



Screen 3.7 – Choregraphe: Project files

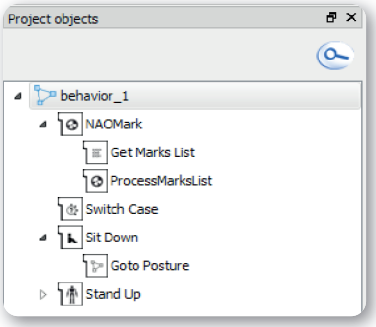
The project files are where you manage your project. Under Properties you can change which languages are supported by your project. You can also delete files from or add them to your project.



Screen 3.8 – Choregraphe: Select project files

There are several options available to you: You can create new directories, import other folders or files, and create dialogs.

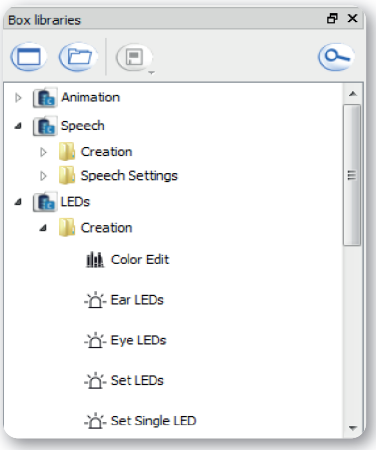
The project objects



Screen 3.9 – Choregraphe: Project objects

Here you can look at a list of all the boxes used in the project. You will learn how this box principle works C.1.
To help you understand: Your program will later be built from several building blocks; these building blocks are all listed here.
This is especially useful for large or complex projects, as it allows you to find a specific box quickly.

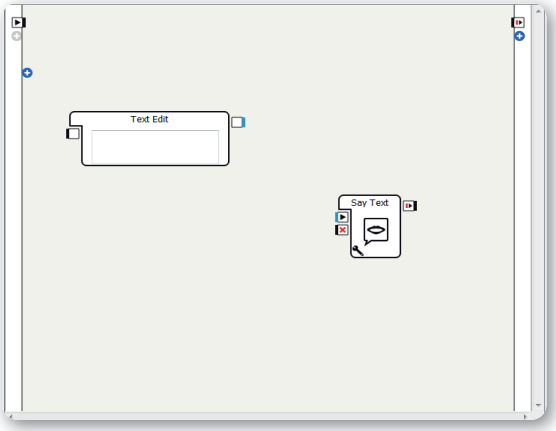
The box libraries



Screen 3.10 – Choregraphe: Box libraries

This section is one of the two most important ones when programming with Choregraphe. Here you can find all the boxes (program parts with a specific function that are required for programming) that you can use as building blocks. If you click on the magnifying glass button, you can search for boxes. (A more detailed explanation of the boxes is given in chapters C.2., C.3., and C.4.)

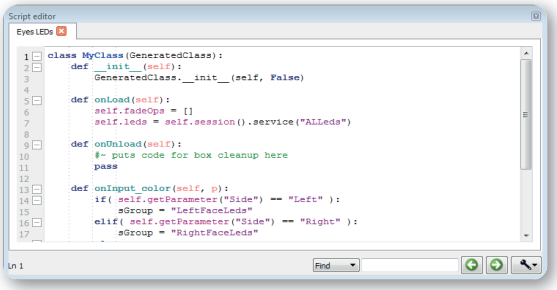
The flow diagram panel



Screen 3.11 – The flow diagram panel

This is the most important section in Choregraphe. It is where you create, modify, and connect all programs. You can drag and drop the boxes from the box library onto this flow diagram panel by holding down the left mouse button. Then you can interact with the box here and connect it to other boxes.

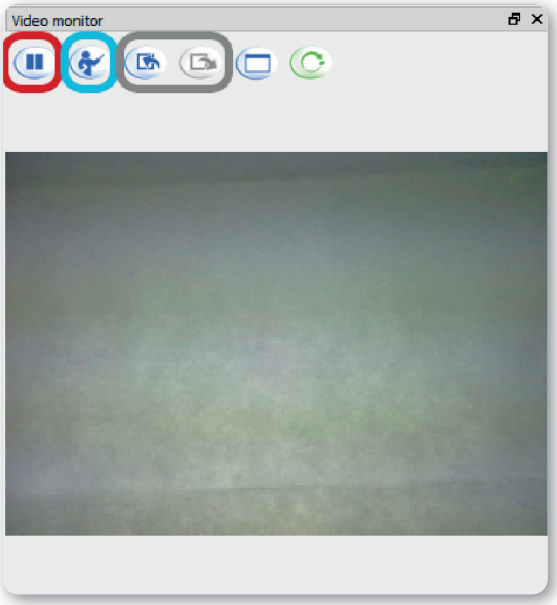
The source code



Screen 3.12 – Choregraphe: Script editor

In simple terms, the source code is the script that tells the program what should happen and when. In order to modify the source code, you need certain other programs (e.g. BlueJ, VisualStudio, etc.). In Choregraphe there is a script editor already installed. Here, this is a display window for the content of a box. That means you can change the source code of a box or write your own box. You can also create a dialog with it (see C.3.1.1.).

The video monitor

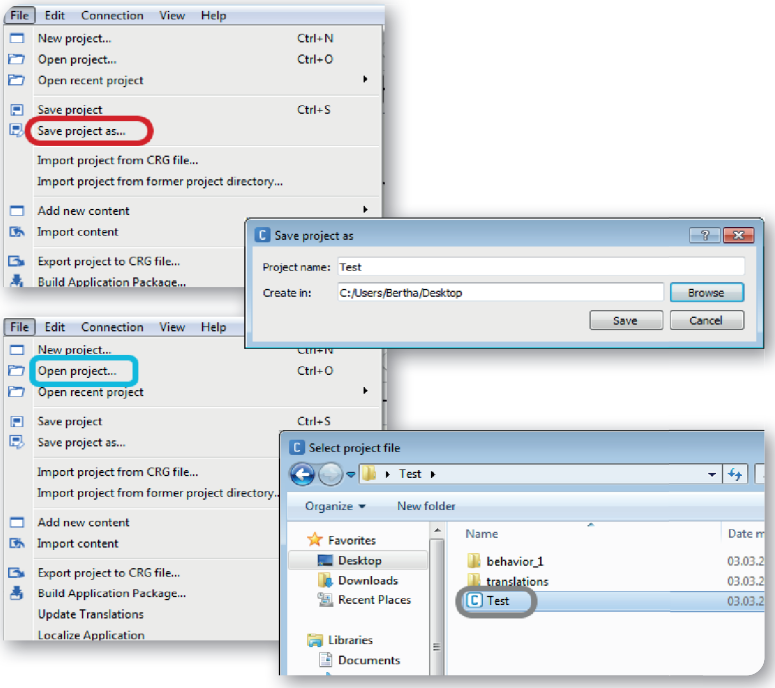


Screen 3.13 – Choregraphe: Video monitor

This window shows a live feed of NAO's camera. If you want, you can stop it. You also need the camera to control the “Vision Recognition” box (see C.3.1.15.). You can teach, import, or export an object.

Now you are familiar with the Choregraphe interface and can create programs. You can choose between two different methods for saving these programs.

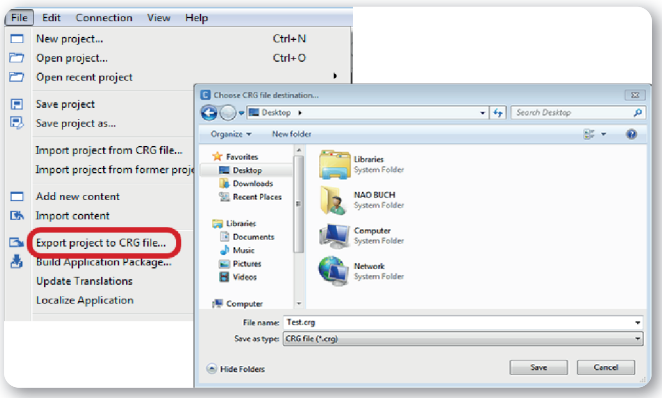
Save as project folder



Screen 3.14 – Choregraphe: Save as/open project folder

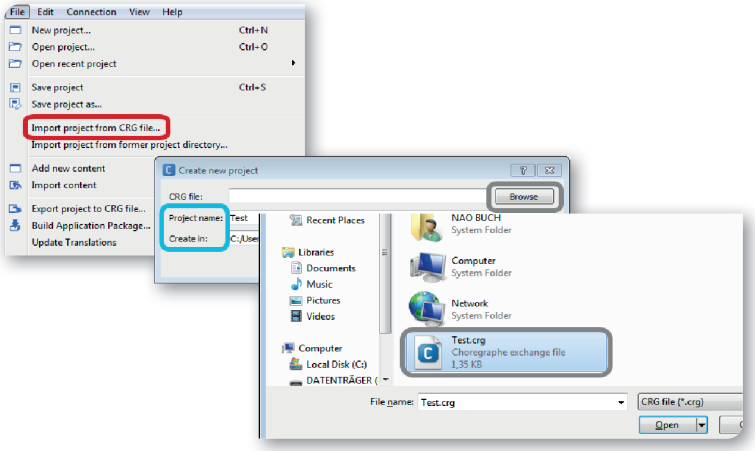
If you click on File > Save project as..., you can give the project a name and select the save location. To open the file again later, click in Choregraphe on File > Open project... and select the .pml file in the project folder. You can also double-click on it, but make sure that the version of Choregraphe is correct. (If you have several versions installed, the oldest version always opens.)

Save as Choregraphe file (.crg):



Screen 3.15 – Choregraphe: Save as .crg file

To save the project in a single file, click on File > Export project to CRG file.... Now you can name the file and select a save location. However, once you saved your project in a single file, you can't make any more changes to it.



Screen 3.16 – Choregraphe: Open the .crg file

You can open files like this in Choregraphe, too. To do so, click on File > Import project from CRG file.... Now you need to give the new project folder a name and select the save location. Then select the file you want to open. You can also double click on the file and enter a name and save location this way, too.

B. 3. Autonomous Life

Autonomous Life is activated by default on every robot, which means that your NAO stands up and looks around expectantly and inquisitively each time you start it.

Screen 4.1 – The "Autonomous Life" button

You can deactivate the Autonomous Life by switching it on or off at the top right of the screen in Choregraphe. However, if you do this, NAO will no longer be able to communicate or move independently.
In order for it to begin to develop its "own life", you need to download the Basic Channel and other applications from the Aldebaran store.
(See A.3.)
You should be aware that only the languages English, French and Japanese have so far been implemented in the Basic Channel.
In order to communicate with it, you must switch the robot language to English on the robot page.

You can find a current list of the speech commands for the Basic Channel at:

http://doc.aldebaran.com/2-8/family/nao_user_guide/basic_channel_conversation_nao.html?highlight=basic

or scan QR code

Some examples of speech commands:

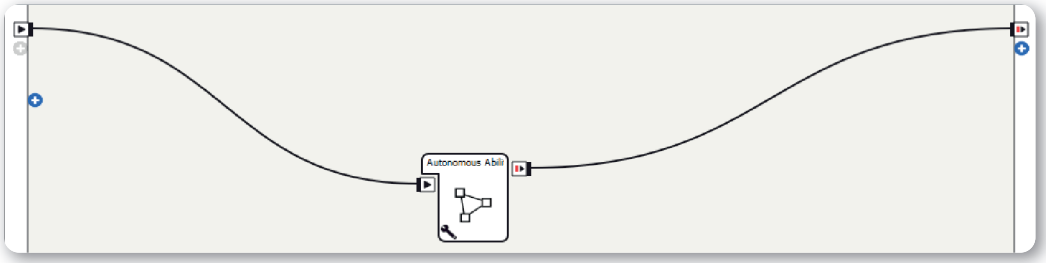
"Can you stand up?"	Makes NAO stand up.
"Raise your arms."	Makes NAO lift its arms.
"Introduce yourself."	NAO introduces itself.
"Can we be friends?"	You are asking NAO if you can be friends.

Note:

The phrases given are just standard formulations. There is a certain amount of flexibility, for example NAO accepts both "How can I call you?" and "What's your name?".

Once you have worked with NAO for a while, you can play with the parameters. How? – This is explained in the following section.

To make more detailed settings to your Autonomous Life, you can use the “Autonomous Abilities” box.

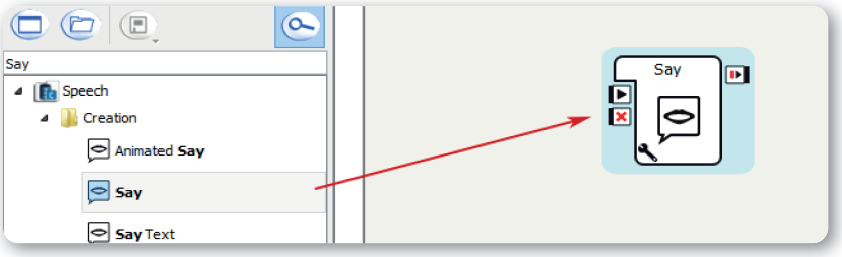


Screen 4.2 – “Autonomous Abilities” box

This box manages whether “Basic Awareness”, “Background Movements”, “Listening Movements”, “Speaking Movements”, and “Autonomous Blink” are each switched on or off.

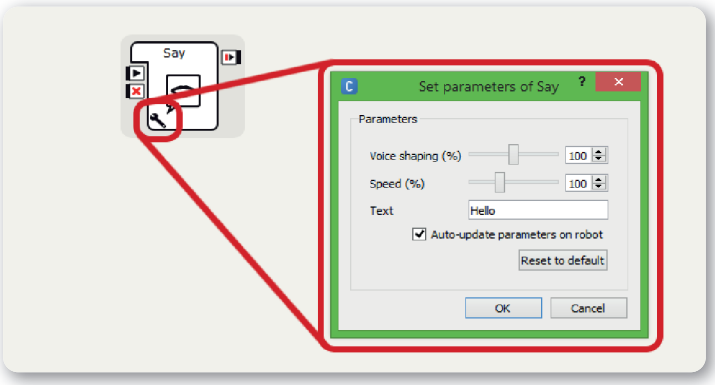
C. First programs

C. 1. What is a box?



Screen 5.1 – Box in Choregraphe

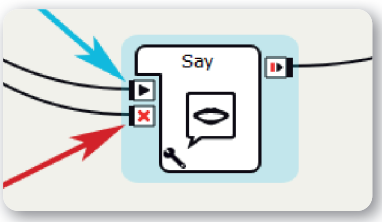
Basically, you can program in Choregraphe by dragging and dropping boxes onto a flow diagram panel. Each box contains a complete program part. Several of these program parts and their functions are explained in the following chapters. However, program parts don't always have to run in the same way but can be changed or modified.



Screen 5.2 – The parameters of a box

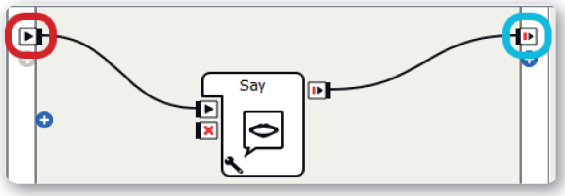
To do this, you can click on the wrench in the corner of a box. As most boxes have their own parameter options, they are explained individually in the following chapters. In order to be able to use the function of a box, the boxes need a signal that tells them to start and how.

Screen 5.3 – In/outputs of a box



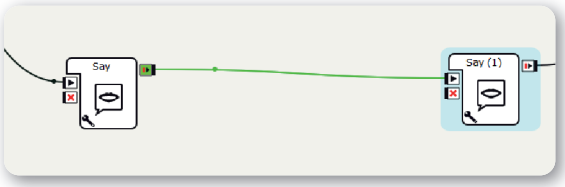
This is done using inputs for starting and outputs for sending the signal to another box. Each box in Choregraphe has one or more inputs (on the left of the box). Almost every box also has one or more outputs (on the

right of the box). Inputs with a black triangle or play symbol start the box. Inputs with a red cross stop the box. (In this book, these inputs are referred to collectively as "standard inputs".)



Screen 5.4 – Starting the program

To start the program you have made later, there is a start signal which is sent when you start the program by pressing the play button (see B.2.). To make the program stop automatically at the end you can connect the last box(es) to an end.



Screen 5.5 – Connecting boxes

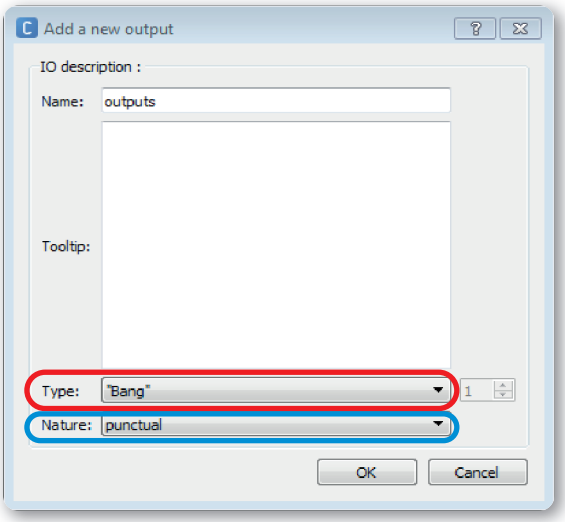
To better visualize the signal transfer, outputs emit green objects at their connection which move along the connection to the input.

Screen 5.6 – Colored outputs

A signal is not simply just a signal. There are various types (types of signal are also called "data types").
You can identify the data type of the inputs/outputs by the color of the stripe on the edge.
The standard data type "bang" is visualized by a black stripe. It simply transfers a signal to start the box.
The data type "dynamic" is alterable (different data types can be accepted, e.g. a number, a letter, a text, etc.) and is mostly used for positions or coordinates. The color of this type is gray. The data type “number” is indicated by a yellow stripe. A blue stripe indicates the "string" data type.

You will need inputs and outputs later. Here is a brief explanation of how you create them.

By right-clicking on the existing input/output you can add a new input/output via "Edit output/input".



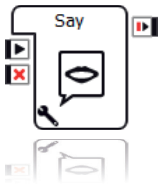
Screen 5.7 – Adding a new output

Under "Type" you set the data type. "Nature" describes when the output is triggered, "onStopped" is triggered as soon as the box is stopped or has run. "Punctual" means that you have to specify when the output is triggered within the box or the source code.

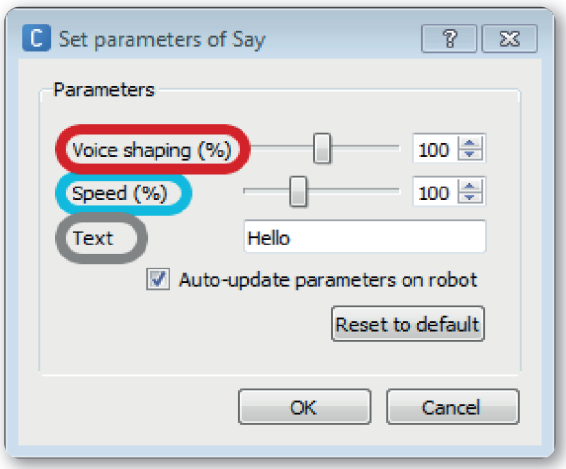
C. 2. Level 1: Beginner

C. 2. 1. Simple boxes

1. The “Say” box

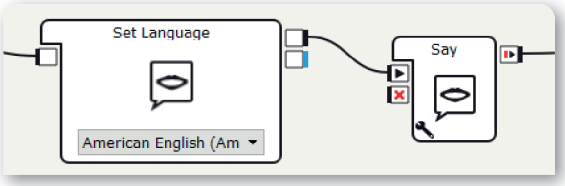


Explanation:
The “Say” box is one of the most essential boxes for your NAO. You can use this box to make your NAO say a text.



Screen 6.1 – Parameters of the “Say” box

By clicking on the wrench in the box you can set several parameters. (Have you forgotten what the wrench is?
No problem, see C.1.) You can set the voice shaping and the speed. You can also enter the text here that you want NAO to say.



Screen 6.2 – “Set Language” box with “Say” box

Note:
You should place a “Set Language” box in front of this box.

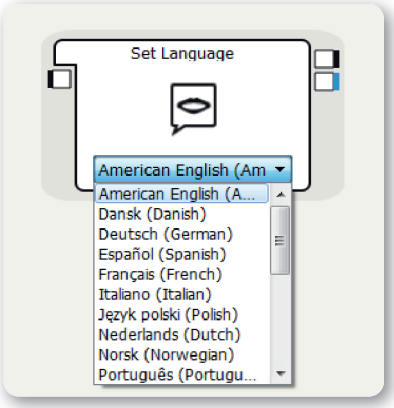
Screen 6.3 – Flow diagram with “Say Text” box

If you want to say longer texts, use the Text Edit box in combination with the “Say Text” box.

What you need:
Relevant language package

2. The “Set Language” box

Explanation:
You can use this box to tell your NAO which language it should speak.

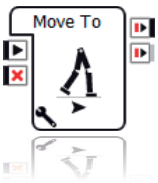


Screen 6.4 – Parameters of the “Set Language” box

All you need to do is select the language from the menu.
The input is on the left side of the box. On the right side of the box, at the top, is the output once the language has been set. In the event of an error, the box issues a signal at the bottom output.

Note:
Setting the language is important, otherwise NAO will speak English.

What you need:
NAO language packages



3. The “Move To” box

Example:

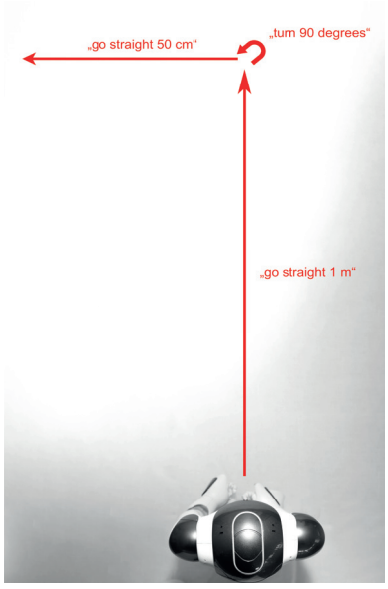
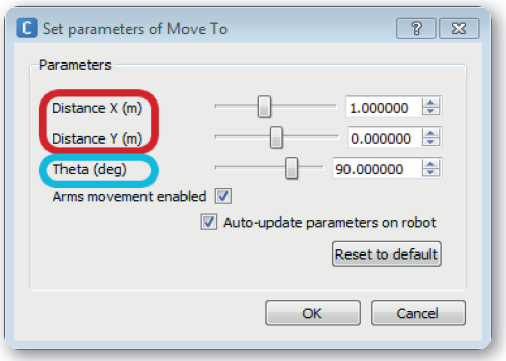


Fig. 6 – NAO follows a specific path

This is what NAO's path will look like if it first walks one meter forward then turns 90° and walks forward again 50 cm. This requires 3 boxes.

Explanation:
You can use this box to make your NAO move to a specific position.



Screen 6.5 – Parameters of the "Move To" box

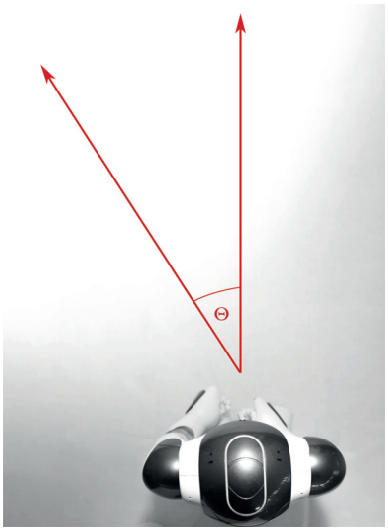
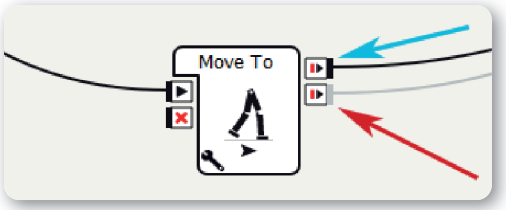


Fig. 7 – The theta angle

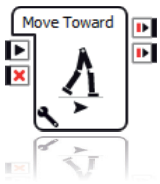
First, NAO turns by the theta angle, that's the angle by which it turns away from its current axis. Then it processes the X- and Y-values at the same time.



Screen 6.6 - Outputs of the "Move To" box

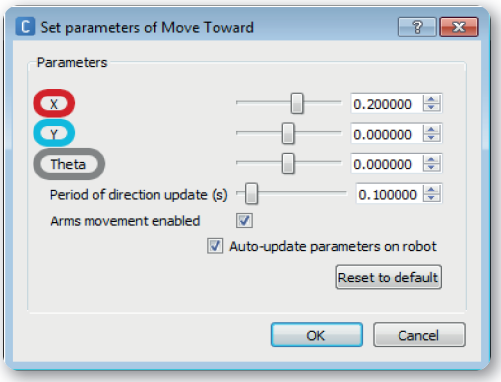
As soon as the values have been processed, it stands still and issues a signal at the Success output.
If it is stopped by anything, for example if it falls or something similar, it issues a signal at the bottom Error output.

Note:
Negative values mean that NAO walks backward or to the left. The Error signal contains the missing distances (X, Y, Theta).



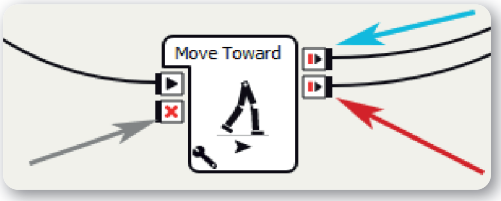
4. The “Move Toward” box

Explanation:
You can use this box to make your NAO walk in a particular direction.



Screen 6.7 – Parameters of the “Move Toward” box

The X-parameter controls how far NAO walks forward, the Y-parameter controls how far it walks to the side. The theta-parameter describes the angle of rotation.



Screen 6.8 - Outputs of the “Move Toward” box

As soon as it has been stopped by a Stop input, NAO issues a signal at the top output. The bottom output is triggered if NAO was stopped by an obstacle.

Note:
Negative values mean that NAO walks backward or to the left. NAO will carry on walking until the box is stopped. So always watch it and use the box with care.

5. The “Stand Up” box

Example:

This is what NAO looks like when it has just stood up.

Explanation:
This box is also a fundamental box in Choregraphe. You can use this box to make your NAO stand upright from any position, so it's a very important box.

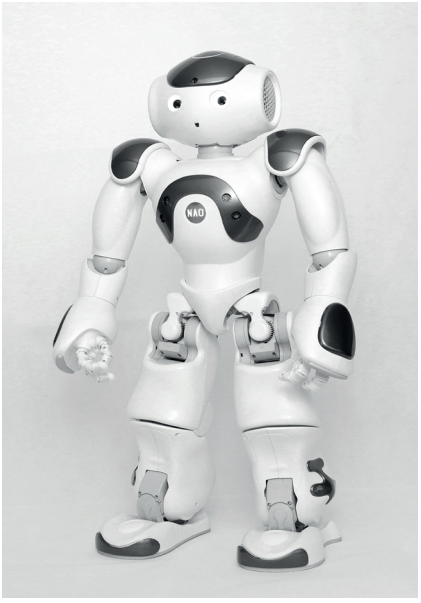
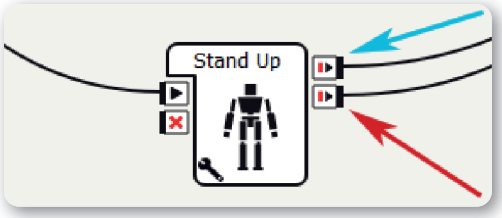
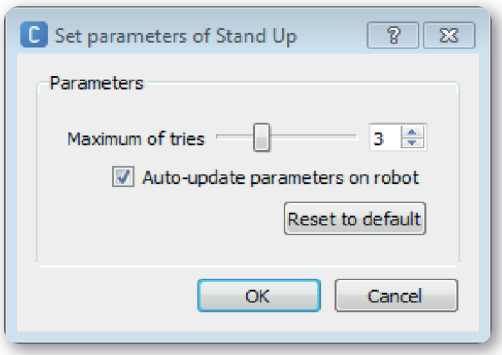


Fig. 8 – NAO is standing



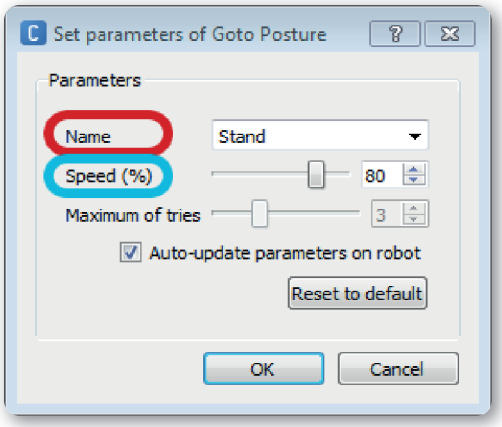
Screen 6.9 - Outputs of the "Stand Up" box

The "Stand Up" box has two outputs on its right side. The top one is for when NAO has stood up successfully, the bottom one is for when NAO has not managed to stand up.



Screen 6.10 – Parameters of the "Stand Up" box

You can now see a black wrench in the bottom left corner of this box. Here you can set how often NAO should try to stand up.



Screen 6.11 – Parameters of the "Goto Posture" box

Double-click to open the "Goto Posture" box, where you can click on the wrench in the box and set the speed of the sitting procedure. You can also set the type of position that NAO should move to here.

Note:

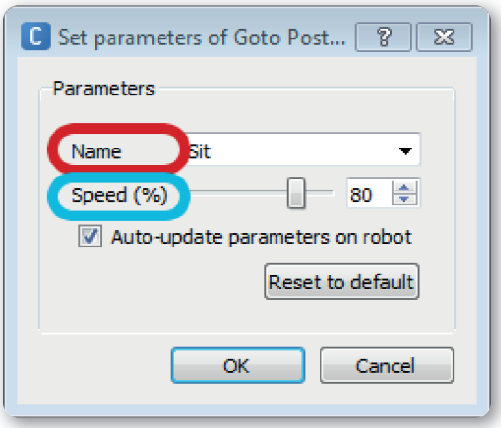
Never set the speed higher than the recommended value of 80%, as anything above this endangers your NAO; it could overheat or NAO could overbalance and fall.

6. The “Sit Down” box

This is what NAO looks like when it has just sat down.

Explanation:
Now your NAO can stand up, however it should also be able to sit down. The Sit Down box is very similar to the Stand Up box. It also has two outputs on the right, one for an Error and one for a Success.

Fig. 9 – NAO is sitting



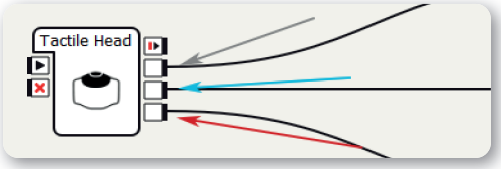
Screen 6.12 – Parameters of the “Sit Down” box

Double-click to go to the box, where you can click on the wrench and set the speed of the sitting procedure. You can also set the type of position that NAO should move to here.

Note:
For the speed, don't use anything higher than the range between 60% and 70% in order to avoid damaging your NAO.

7. The “Tactile Head” box

Explanation:
This box manages the selection buttons on NAO's head. The box issues a separate signal for each button.



Screen 6.13 – Outputs of the “Tactile Head” box

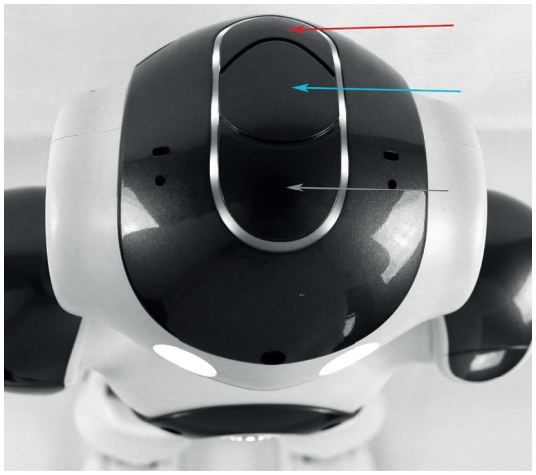


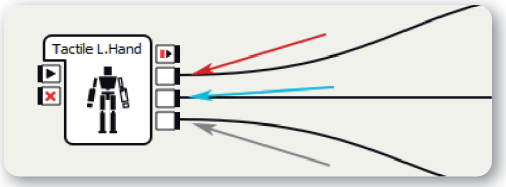
Fig. 10 – The tactile sensors

The top output is for the front sensor, the middle one for the middle sensor, and the bottom one for the rear sensor.
In addition to the standard inputs Start and Cancel and the Stop output, the three other outputs are important here. They each represent one of the three head sensors.

Note:
You can also make all three outputs run on one input. This way you can start a box when the head is touched – regardless of which of the sensors was touched.

8. The “Tactile Right/Left Hand” box

Explanation:
NAO doesn't just have sensors on its head but also on its hands. To manage the signals generated by these touches, there is the “Tactile Right Hand” or “Tactile Left Hand” box for the right and left hands respectively. The box issues a separate signal for each of the three sensors.



Screen 6.14 – Outputs of the “Tactile Left Hand” box

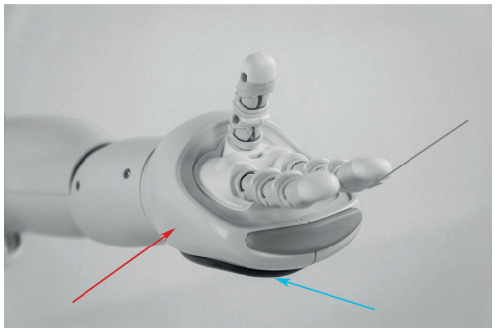


Fig. 11 – The “Tactile Left Hand” sensors

The top box is for the left sensor, the middle one for the rear sensor, and the bottom one for the right sensor.
For each of the two boxes there are the standard inputs Start and Cancel, the Stop output, and the three outputs: the top one for a touch on the left side of NAO's hand, the middle one for a touch on the underside, and the bottom one for a touch on the right side. The sides are correct when NAO's palm is facing up.

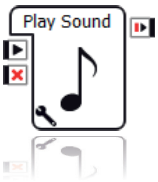
Note:
When you use these sensors, you should be careful of NAO's fingers.

9. The “Hands” box

Explanation:
You can use this box to control NAO's hands. You can choose whether NAO opens or closes both hands or only one.

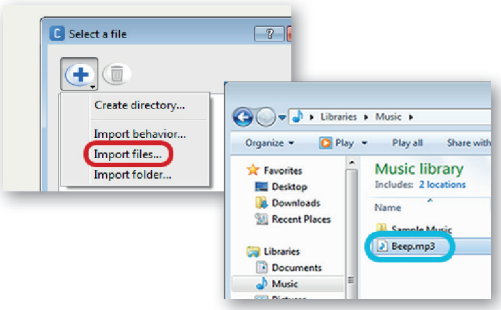
Screen 6.15 – Parameters of the “Hands” box

To set the parameters, click on the wrench. You can now select the side(s) and the action here.
On this box there is only the standard input and the output.



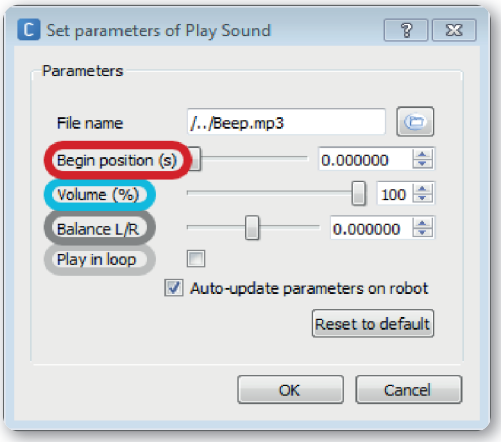
10. The “Play Sound” box

Explanation:
This box allows you to make your NAO play any sound you choose. However, it does require several steps. First, click on the wrench in the bottom left corner.



Screen 6.16 – Import file

Now you can select the file you want to play. To do this, click on the folder, click the plus symbol, and select "Import files...". Now you can import a file from your computer.

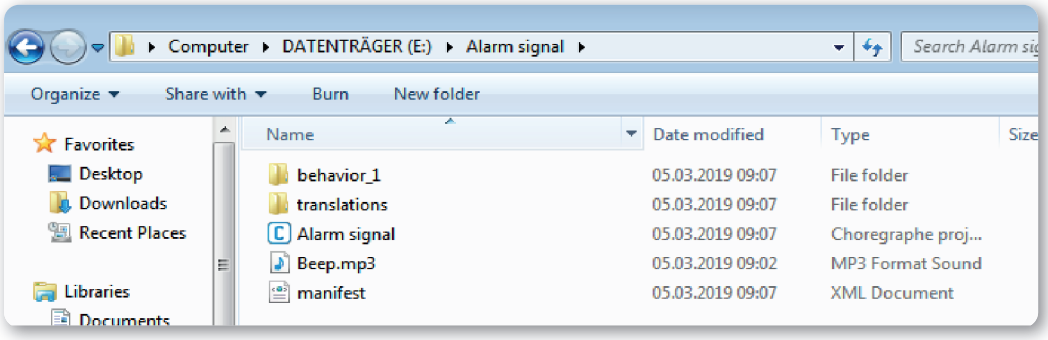


Screen 6.17 – Parameters of the “Play Sound” box

Next you can set several parameters. You can specify the begin position, change the volume, and set the balance for both speakers on NAO. Plus, you can set the check mark next to play in loop. This makes NAO play the sound over and over.
For this box there are only the two standard inputs and the output.

Note:

At the end, the sound file must be located in the folder for the project. You can also create another subfolder for this.



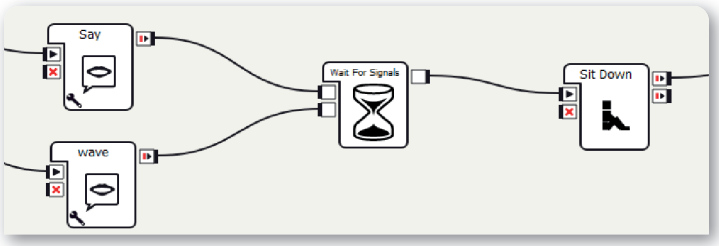
Screen 6.18 – File in the project folder

What you need:

A sound file (e.g. .mp3, .wma, .m4a).

11. The “Wait for Signals” box

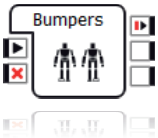
Example:



Screen 6.19 – Example of the “Wait for Signals” box

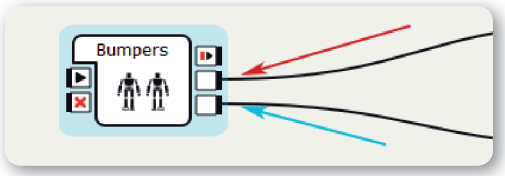
Here is the box with two connected inputs. This means that in this case, NAO must have said "Hello" and finished waving before it sits down.

Explanation:
This box requires two signals in different inputs for it to trigger an output signal. So two conditions need to be met.



12. The “Bumpers” box

Explanation:
This box detects touches on NAO's feet. The box then issues a signal for the right and the left side.



Screen 6.20 – Outputs of the “Bumpers” box

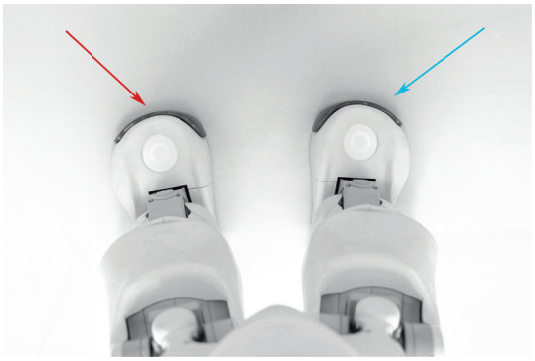
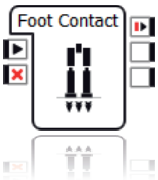


Fig. 12 – The “Bumpers” box from the robot's perspective

The top box represents the left sensor, the bottom one the right sensor.

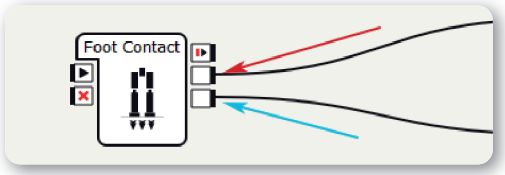
In addition to the Start and Cancel inputs there are the Stop output and one output each for the signal issued when there is a touch on the left or right.

Note:
If both bumpers are pressed, the box issues two signals.



13. The “Foot Contact” box

Explanation:
You can use this box to check whether NAO is still in contact with the floor. The box issues a signal if only one foot is on the floor or if neither foot is on the floor.

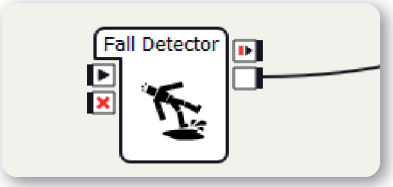


Screen 6.21 – Outputs of the “Foot Contact” box

The top box represents touching the floor with at least one foot, the bottom one represents a loss of floor contact.
Here there are the inputs Start and Cancel, the Stop output, and the outputs described above.

14. The “Fall Detector” box

Explanation:
If NAO has fallen over, the “Fall Detector” box recognizes this fall and issues a signal 0.5 seconds after the fall.



Screen 6.22 – Output of the “Fall Detector” box

In addition to the two Start and Cancel inputs and the Stop output, the box has an output for when NAO has fallen over.

Note:
The robot should not stand up too quickly, otherwise parts might break or the robot might fall over again.

C. 2. 2. Simple examples

This chapter and chapter C.3.2. give several examples based on the boxes you have already learned about. The intention of these examples is to complement the dry, theoretical content of the reference book for boxes directly with your own experiences and impressions. In this way, we hope to achieve a lasting learning effect. The end product should be a large program that uses all boxes that are important for programming and understanding other programs. Later, you should also be able to take parts of this program and insert them into your own ones.

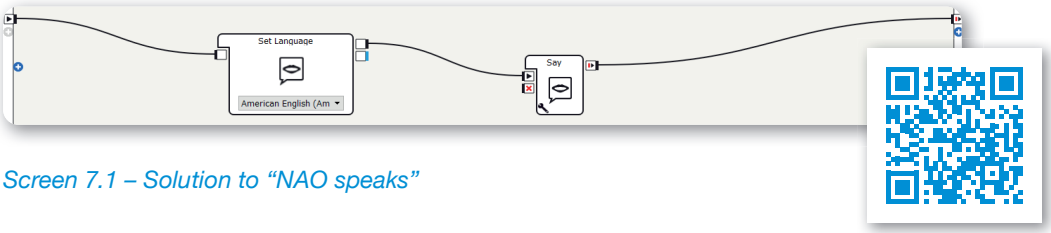
Using the QR codes after each of the examples, you will find suggested solutions in video form and as a program.

1. NAO speaks

Aim: From now on, make NAO speak the language you want. Make it say "Hello world".

You need these boxes: “Set Language” ***(see C.2.1.2.); “***Say” ***(see C.2.1.1.)***

What to do:



Screen 7.1 – Solution to “NAO speaks”

Drag the boxes “Set Language” and “Say” onto your flow diagram panel. Connect them as shown in the screenshot. Then in the “Say” box replace "Hello" with "Hello world" or any other text you like. Now all you need to do is set the language if you want a language other than English.

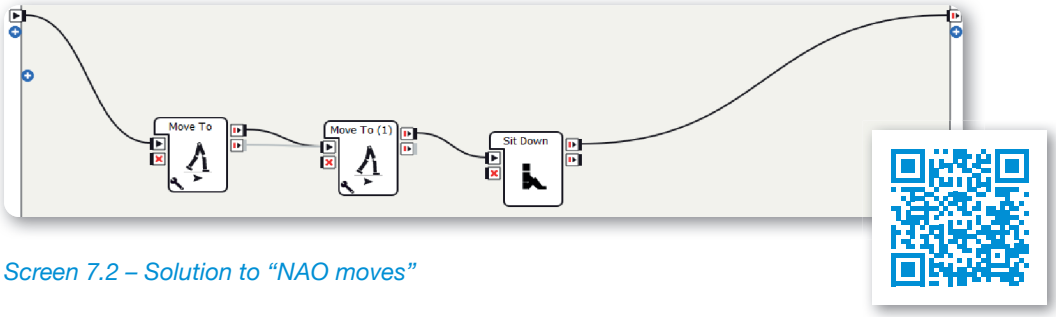
Vary the pitch of the voice and the talking speed. You can also play around with the language. English should be available free of charge.

2. NAO moves

Aim: Next, make your NAO turn 90° to the left, then walk diagonally one meter forward and half a meter left and then sit down.

You need these boxes: “Move To” ***(see C.2.1.3.);*** “Sit Down” ***(see C.2.1.6.)***

What to do:



Screen 7.2 – Solution to “NAO moves”

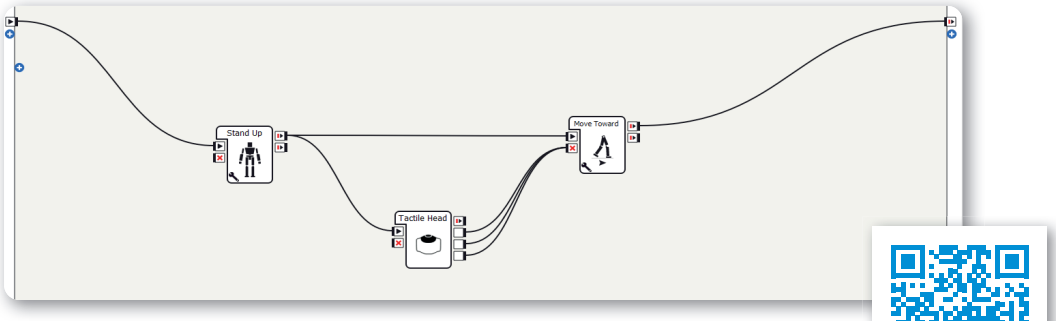
Drag one “Sit Down” and two “Move To” boxes onto your flow diagram panel. Connect them as shown in the screenshot. In the first “Move To” box, set all parameters to 0, set theta to 90°. In the second “Move To” box you need to set the X-parameter to one, the Y-parameter to 0.5, and theta to zero.
Now vary the angle, the value for the distance traveled, or add more “Move To” boxes.

3. NAO learns to walk

Aim: Make NAO stand up and walk straight ahead until you touch it on the head.

You need these boxes: “Move Toward” ***(see C.2.1.4.);*** “Stand Up” ***(see C.2.1.5.);*** “Tactile Head” ***(see C.2.1.7.)***

What to do:



Screen 7.3 – Solution to “NAO learns to walk”

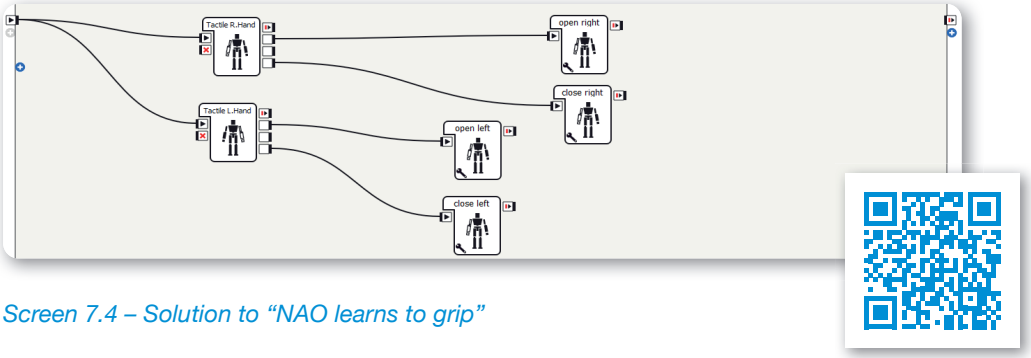
Drag the boxes “Stand Up”, “Tactile Head”, and “Move Toward” onto your flow diagram panel. Connect them as shown in the screenshot. Set the X-parameter of the “Move Toward” box to its maximum (1) and set the Y-parameter and the theta-parameter to 0. Leave the other boxes at their basic settings.
Vary which sensors need to be touched or change the parameters of the “Move Toward” box.

4. NAO learns to grip

Aim: Make NAO move its hands. When you touch its hand on the left side, it should open its hand; when you touch it on the right side, it should close it. Make this function available for both hands.

You need these boxes: “Tactile Right Hand” ***(see C.2.1.8.);*** “Tactile Left Hand” ***(see C.2.1.8.); “***Hands” ***(see C.2.1.9.)***

What to do:



Screen 7.4 – Solution to “NAO learns to grip”

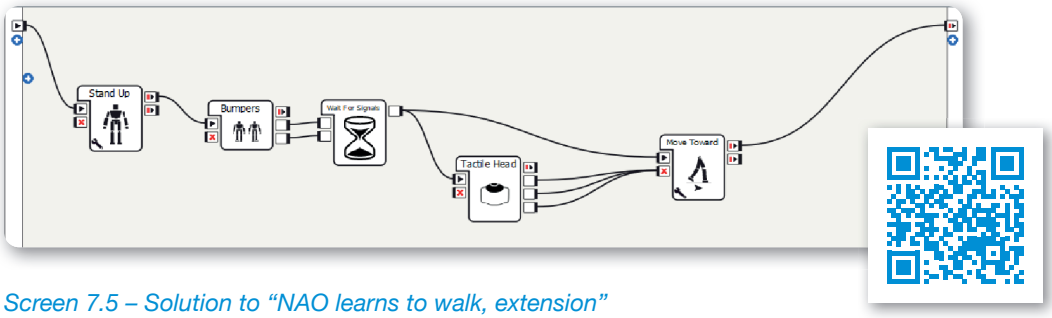
Drag the boxes “Tactile R. Hand”, “Tactile L. Hand”, and four “Hands” boxes onto your flow diagram panel. Connect the boxes as shown in the screenshot. Name the “Hands” boxes as shown in the screenshot (Right open, Right close, etc.). To do this, right-click on the box and select "Edit box". You can enter the new name here. Now all you have to do is set the actions. To do this, always select the relevant hand and make it open once and close once.
Now vary where NAO needs to be touched to open/close the hand or whether both hands should be activated at the same time.

5. NAO learns to walk – extension

Aim: Extend your program from task 3 so that NAO stands up but doesn't start to walk until both bumpers (pressure sensors on the feet) are touched.

You need these boxes: [Boxes from task 3]; “Bumpers” ***(see C.2.1.12.);***
“Wait For Signals” ***(see C.2.1.11.)***

What to do:



Screen 7.5 – Solution to “NAO learns to walk, extension”

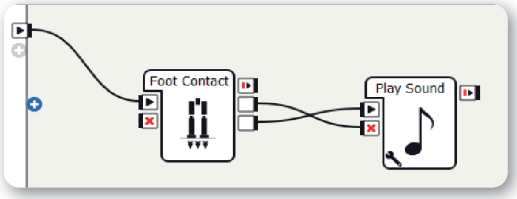
Now drag the boxes Bumpers and Wait For Signal onto your flow diagram panel.
Connect the new boxes as shown in the screenshot.
Now test for yourself and vary some parameters.

6. Alarm signal

Aim: If NAO were to fall over, it would be good if it could let you know. So have NAO play a beep as soon as it loses contact with the floor.

You need these boxes: “Foot Contact” ***(see C.2.1.13.);*** “Play Sound” ***(see C.2.1.10.)***

What to do:



Screen 7.6 – Solution to “Alarm signal”

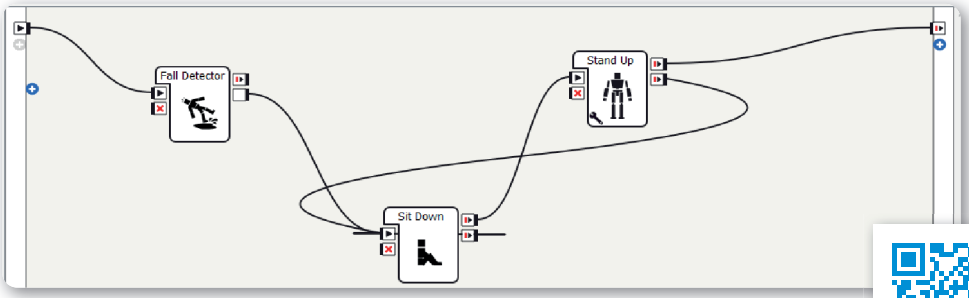
Drag the boxes Foot Contact and Play Sound onto your flow diagram panel and connect them as shown in the screenshot. Import a beep (to find out how this works, see the explanation for the box in chapter C.2.1.10.). Then set the check mark next to Play in loop.

7. NAO saves itself

Aim: If NAO has fallen over, make it stand up again. If it is lying on the ground, make NAO sit up and then stand.

You need these boxes: Fall Detector ***(see C.2.1.14.);*** Stand Up ***(see C.2.1.5.)***; Sit Down ***(see C.2.1.6.)***

What to do:



Screen 7.7 – Solution to "NAO saves itself"

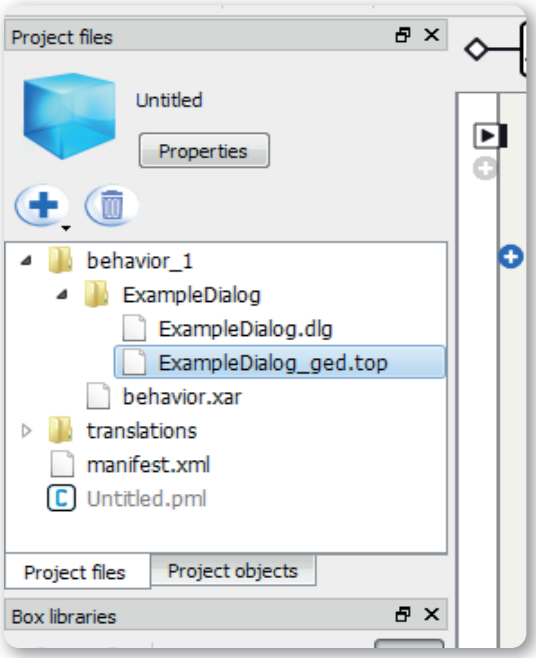
Drag the boxes Fall Detector, Sit Down, and Stand Up onto your flow diagram panel. Connect them as shown in the screenshot. To prevent accidents, you should always set the speed within the Stand Up box to 40%.

C. 3. Level 2: Advanced

C. 3. 1. Moderate boxes

1. The “Dialog” box

Explanation:
This box lets you program a dialog between NAO and you. Drag this box onto your flow diagram panel and set "English" or the relevant language (if you can't find the language you want, see the Note below). In the "Project files" (see the Note for how to find them) under "behavior_1", a new folder and the file "ExampleDialog_ged.top" should now have appeared. Double click to open the file.



Screen 8.1 – The “Dialog” script

A script editor should now open. In order to use the “Dialog” box, you have to set rules; here are the rules for the dialog:
The row **`"u:(e: onStart) Hello, I am NAO"`** is triggered when the box is activated.

The content of the parentheses is always the condition for the sentence spoken by NAO that follows. If you write **`"$onStopped=1"`** after a sentence, the output is triggered and the box is ended.

You can also define several conditions for a response:
`u:(["bye" "see you"]) Bye`
The response is triggered after both "bye" and "see you".

Also, you can determine whether a word needs to be said:
`u:(how are you {doing}?) Good, and you?`
The word "doing" is not strictly necessary for NAO to say its response phrase.

You can also write responses from NAO to responses from the person it is talking to, making it a conversation:

```
u:(["how are you {doing}" "how are you doing"]) I'm well, and
you?
      u1:(good) Great, I'm pleased to hear it!
      u2: …
      u1:(bad) That's a shame.
```

u1 is only considered if a response was just given to u. In theory, this hierarchy can be expanded as far as you like.

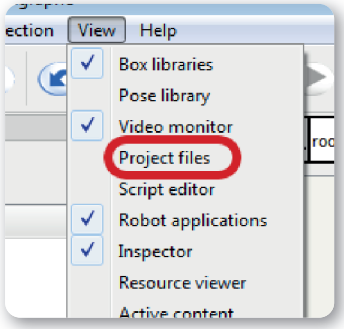
You can also group several responses in one response pool:

```
concept:(hello) [hello hi "good morning" "good afternoon" "good
evening" hey "hi there" "what's up"]
```

You initiate this with the keyword "concept:". Individual words do not need to be demarcated by "", responses with several words need to be demarcated.
Another option for using the “Dialog” box is to use it as speech recognition. To do this, simply add an output to it and set a condition.

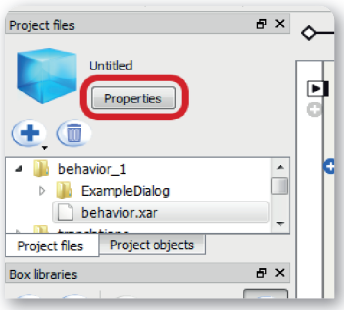
```
u:(right) $Outputright
```

Note:



Screen 8.2 – Project files in View

If you have purchased the language package but it is not displayed, click on View and then on Project files.



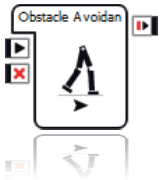
Screen 8.3 – Properties in Project files

In the Properties you can now tick the box next to the relevant language. Now the relevant language can also be selected as the language for the box.

What you need:

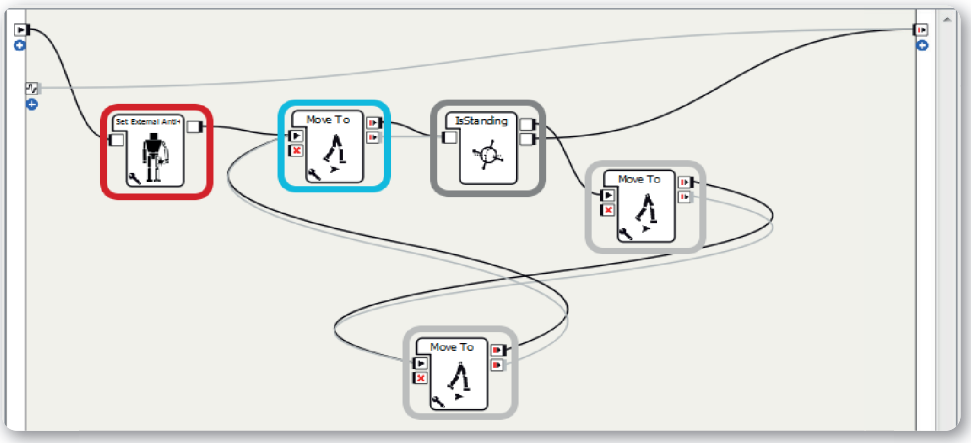
Relevant language package

2. The “Obstacle Avoidance” box



Explanation:
You can use this box to make NAO walk straight ahead until it reaches an obstacle. Then it turns to the right and continues walking.

Sequence:



Screen 8.4 – Function of the “Obstacle Avoidance” box

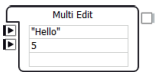
1. Collision warnings are switched on (checked via sensor)

2. Walk a certain distance (can be stopped by collisions)

3. Check whether NAO is standing

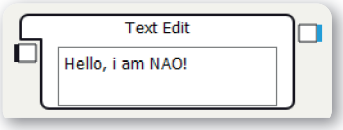
NAO is standing:	Walk backward a certain distance, turn to the right, skip to 2.
NAO is not standing:	Send signal to the output

Note:
The box is also stopped if the event "robothasfallen", meaning the robot has fallen over/is no longer standing, is triggered. This diagram is contained in Choregraphe and is a subcomponent of an algorithm that steers NAO out of a maze.



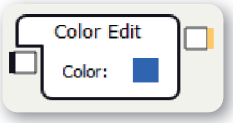
3. The “Text/Number/Multi/Color Edit” boxes

Explanation:
As the name suggests, these boxes are for processing texts, numbers, colors. In response to their start signal, they issue a fixed variable to their output.



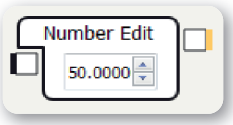
Screen 8.5 – The “Text Edit” box

In the Text Edit box you can enter a specific text.



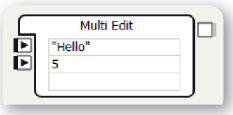
Screen 8.6 – The “Color Edit” box

You can use the Color Edit box to specify the colors of the LEDs in NAO’s eyes



Screen 8.7 – The “Number Edit” box

The Number Edit box forwards predetermined numbers. These are required, for example, when you want to assign numbers to signals so that you can tell them apart.

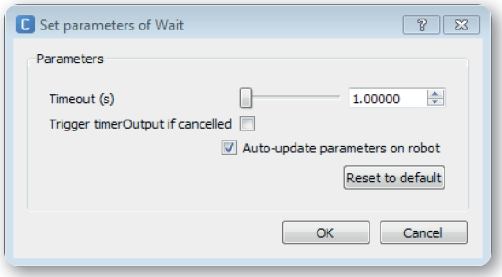


Screen 8.8 – The “Multi Edit” box

If you want to differentiate between several different values, you can use the Multi Edit box. (For example, several boxes issue a signal. You want to send different content with these signals.)

4. The “Wait/Delay” box

Explanation:
These two boxes have very similar functions. The Delay box delays the input signal by a certain time. This box can also delay several signals at once.

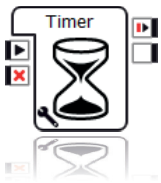


Screen 8.9 – Parameters of the “Wait” box

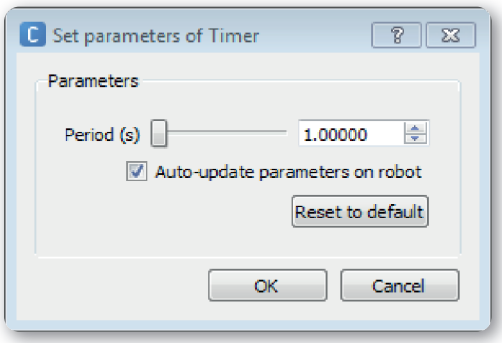
The Wait box also passes on a signal after a certain time. You reset the waiting time by triggering the Start input again.

Note:
In the “Delay” box you can set the parameters so that an Output signal is issued as soon as a Stop input signal arrives.

5. The “Timer” box



Explanation:
This box sends signals to its bottom output at a specific time interval.

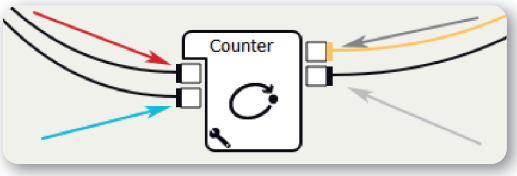


Screen 8.10 – Outputs of the “Timer” box

It is started by a Start input and issues a signal from the top output as soon as it is stopped by a Stop input.

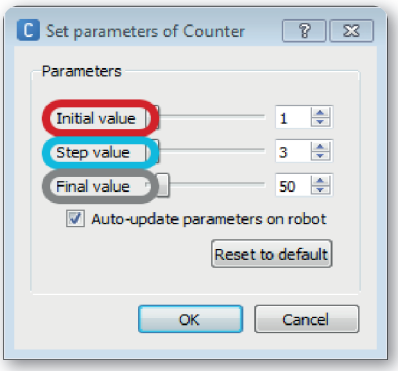
6. The “Counter” box

Explanation:
You can use this box to count signals that are sent to the box. As soon as a maximum is reached, which you can specify, an end signal is issued.



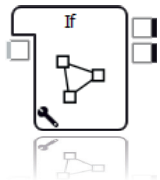
Screen 8.11 – In/outputs of the “Counter” box

There are two inputs on the left side of the box: The top one is the one that counts signals. The bottom one resets the counter. The yellow output issues the current value whenever the counter is changed. The other output issues a one-time signal as soon as the final value you selected is reached.



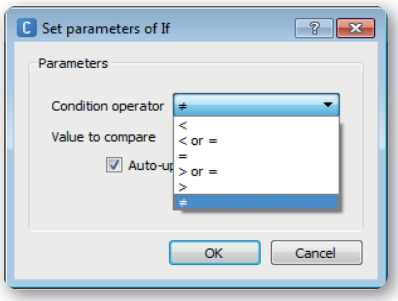
Screen 8.12 – Parameters of the “Counter” box

You can use the wrench symbol to specify the final value, step value, and initial value.



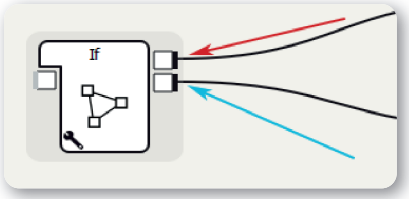
7. The “If” box

Explanation:
You can use this box to compare the value of a signal with a value you have selected.



Screen 8.13 – Parameters of the “If” box

The input on the left receives the value and compares it with the fixed value of the box. You can select from greater than, greater than *OR* equal to, equal to, less than, less than *OR* equal to, not equal to.

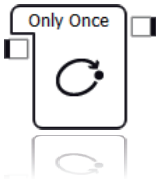


Screen 8.14 – Outputs of the "If" box

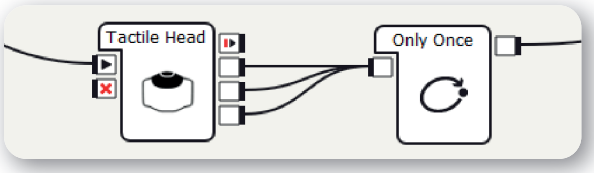
Depending on whether the statement applies, one of the outputs is triggered: The top one sends a signal if the statement is true, so the bottom output sends a signal if the statement is false.

Note:
You can also compare texts (or strings), but only work with statements that are either equal or not equal here.

8. The "Only Once" box



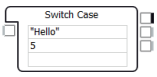
Example:



Screen 8.15 – Example with the Tactile Head

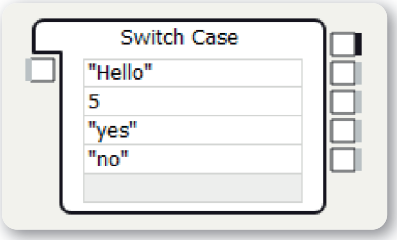
It doesn't matter here how often which of NAO's head sensors is touched. The box only forwards a signal. This avoids several box starts in succession.
Explanation:
You can use this box to specify that a signal is only forwarded once, even if several signals are sent by the previous box. This is helpful with an activation via the Tactile Head sensor, for instance (see example).

Note:
The "Only Once" box resets itself when the main box is discharged.



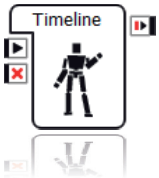
9. The “Switch Case” box

Explanation:
This box describes a multiple selection between the signals Variable or Text.
The input receives the variable or text and compares it with the cases you have specified. As soon as the input matches a case, the output for the same row is triggered. If the input does not match any of the options, the top black output is triggered.



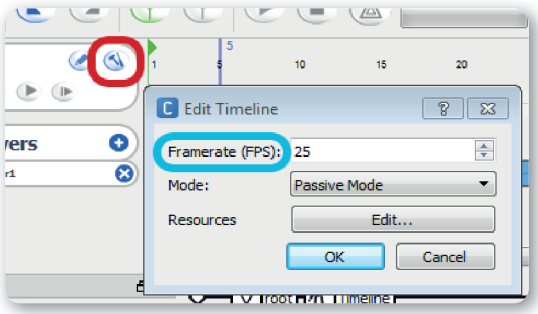
Screen 8.16 – The “Switch Case” box

Note:
Words (or strings) or whole texts must be in quotation marks ("") in the internal text field, otherwise they are not recognized. There is no limit to the different cases.



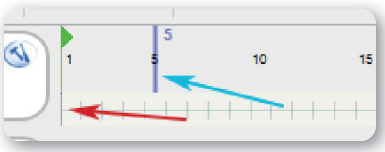
10. The “Timeline” box

Explanation:
You can use this box to make NAO assume positions, or frames, of your choosing, even several in succession. To create simple sequences, all you need to do is move NAO into the desired position using the Animation Mode and then press F8 within the “Timeline” box. This saves the position at the top in the timeline. The numbers on the timeline describe the frame rate.



Screen 8.17 – The parameters for the “Timeline”

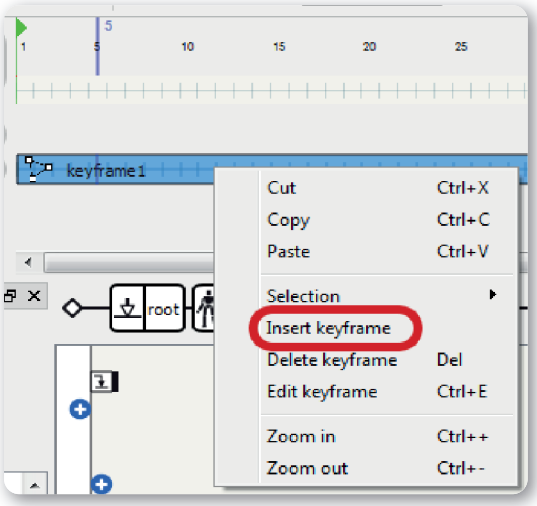
You can set this number by clicking on the little tool symbol. Normally it is set to 25 frames per Second (FPS), but you can change it to whatever you like.



Screen 8.18 – The “Timeline”

The green flag represents the start of the timeline, the red one shows the end.
The blue line shows the current point in time.

Below the timeline you will see a long blue bar labeled Keyframe1. This is the keyframe that is called up for frame 1. Think of keyframes as a diagram that is started as soon as the start frame is called up and ends when the next keyframe begins.

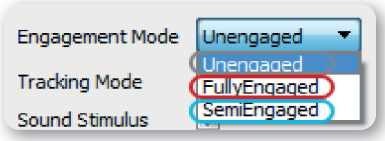


Screen 8.19 – The keyframes of the timeline

To insert a new keyframe, right-click on the keyframe bar and then on Insert keyframe.

11. The “Basic Awareness” box

Explanation:
If you want NAO to be aware of its surroundings, you need the “Basic Awareness” box. You can use it to set whether and how NAO should react, and to what. As there are many different possibilities for setting NAO, they are described individually.
You can use the Engagement Mode to specify how NAO reacts to different stimuli.



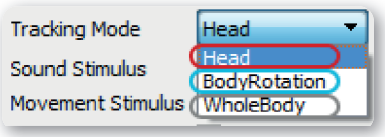
Screen 8.20 – The Engagement Mode parameter

1. Fully Engaged: As soon as NAO directs its attention to a stimulus, it does not react to any other stimuli.

2. Semi Engaged: When NAO has directed its attention to a stimulus and another stimulus is received, it looks in this direction but turns back to the original stimulus after a short time.

3. Unengaged: NAO reacts to every stimulus and "forgets" the previous one.

The Tracking Mode determines how NAO observes its target.



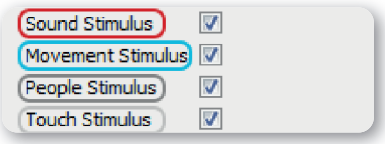
Screen 8.21 – The Tracking Mode parameter

1. Head: NAO only moves its head.

2. Body Rotation: NAO moves its head and its body; it stays standing where it is though.

3. Whole Body: NAO "bends and stretches" to stay in visual contact with the stimulus.

Finally, you can select which stimuli NAO should react to. Simply place a check mark next to the stimulus you want.



Screen 8.22 – The Stimulus parameter

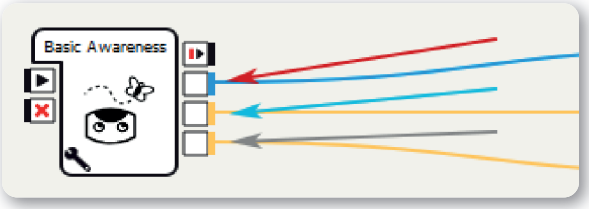
Sound Stimulus – NAO reacts to sounds

Movement Stimulus – NAO reacts to movements

People Stimulus – NAO reacts when he has detected a person

Touch Stimulus – NAO reacts to contact with its pressure or touch sensors

In addition to the two standard inputs and the stop output, the box has one for detecting a stimulus, one for detecting a person, and one for losing the detected person.

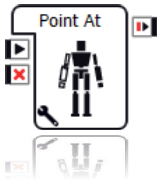


Screen 8.23 – Outputs of the "Basic Awareness" box

The blue box issues a signal as soon as NAO has received a stimulus. The top yellow box represents detecting a person, the bottom one for when NAO loses sight of this person.

Note:
If you don't want NAO to react to any stimuli, you can remove all the check marks.

12. The "Point At" box



Example:
This is what it looks like when NAO points to something.

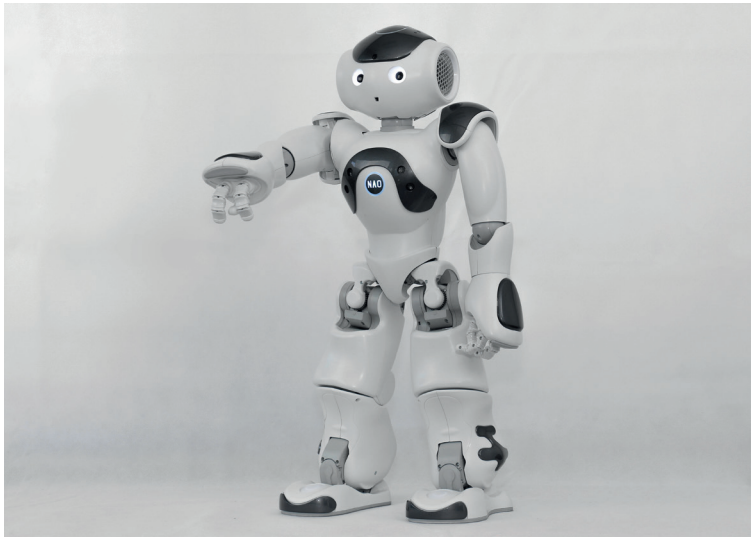
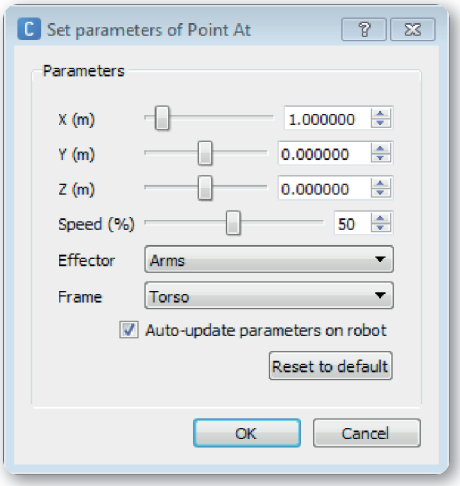


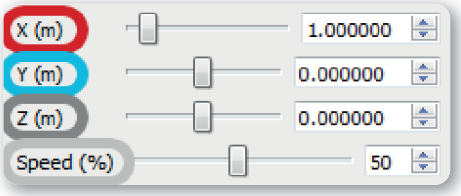
Fig. 13 – NAO points straight ahead

Explanation:
You can use this box to make your NAO point to a specific place:



Screen 8.24 – The parameters of the "Point At" box

There are several parameters available for precisely specifying the position of NAO's hand:



Screen 8.25 – The X/Y/Z/Speed parameters

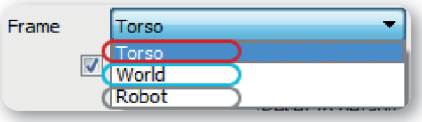
Use the Z-slider to control how high/low NAO points.

Use the Y-slider to control how far to the side NAO points.

Use the X-slider to control how close to or far away from NAO's body its hands should be.

There is also a Speed slider that you can use to set the speed.

In addition, you can set the frame of reference from which the information should apply. You can select torso, robot, and world.



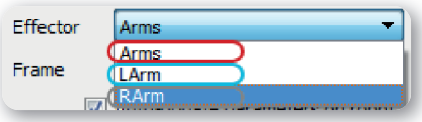
Screen 8.26 – The Frame parameter

Selecting Torso means that the movement takes place from the middle of the chest.

Selecting World means that NAO moves relative to its position.

Selecting Robot means that the movement takes place from hip height.

You can also set which of NAO's extremities should be moved.



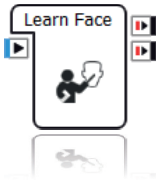
Screen 8.27 – The Effector parameter

You can select from both arms, only the left arm, or only the right arm. NAO starts as soon as a signal reaches the Start input.

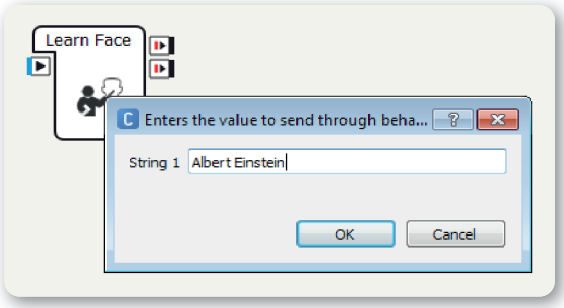
Note:

Make sure that NAO is always standing on a flat, non-slip surface. Otherwise there is a risk of NAO being damaged.

13. The “Learn Face” box



Explanation:
This box is required before using the Face Recognition box. You can use it to save faces. To do this, drag it onto the flow diagram panel and start the program. Then double-click on the input of the box and enter a name for the face.



Screen 8.28 – Entry for the “Learn Face” box

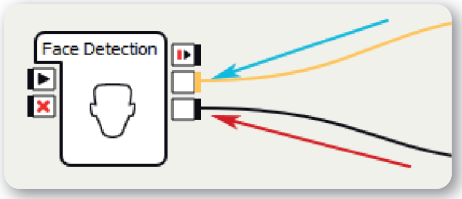
Now go to NAO and hold your face in front of its camera. As soon as it has detected the face, NAO's eyes light up green. If it doesn't detect the face, it's eyes light up red and you need to try again.

Note:
You can use the Unlearn All Faces box to erase the faces from NAO's memory.

14. The “Face Recognition”/”Face Detection” box

Explanation:
With the “Face Detection” box, NAO detects faces and issues a signal with the number of faces detected.

Screen 8.29 – Outputs of the “Face Detection” box



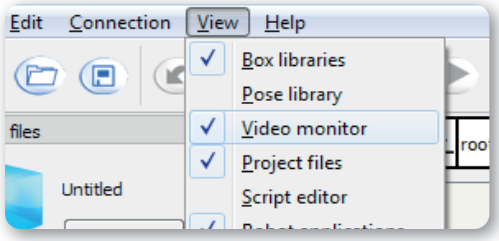
If NAO does not detect a face, it issues a signal at the black output. At the yellow output, NAO issues the number of faces.

In oder to recognize the faces with the “Face Recognition” box, you first need to save the faces with the “Learn Face” box. This box issues a signal when NAO recognizes a face.

Note:
The signal of the “Face Recognition” box is the name of the face recognized.

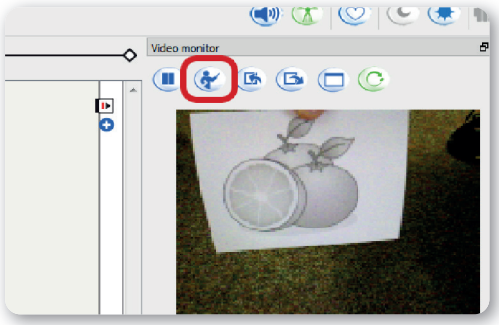
15. The “Vision Recognition” box

Explanation:
This box allows your NAO to recognize certain objects or images. However, you need to perform several steps before this is possible. First you need to save the object to be recognized. To do this, you need to be connected to your NAO (see B.2. Choregraphe).



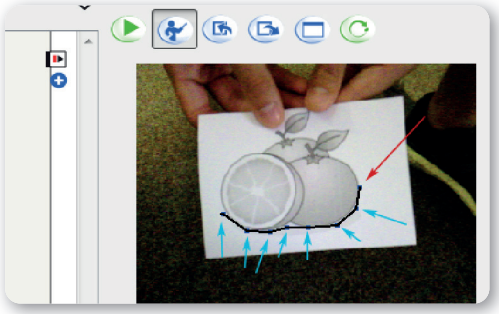
Screen 8.30 – Video monitor under View

Now open the Video monitor in Choregraphe (Toolbar > View > Video monitor).



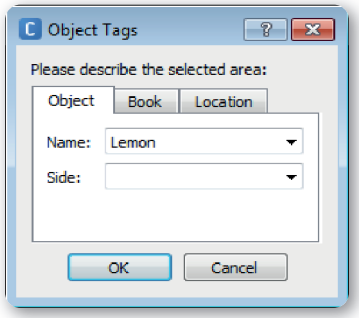
Screen 8.31 – “Learn Face” under Video monitor

Then hold the object or image in front of the camera and click on the learn button in the Video monitor.

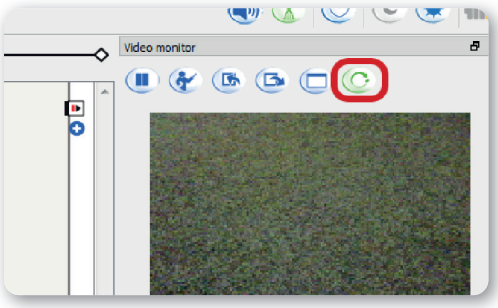


Screen 8.32 – Select area

When the countdown has finished, the image is frozen and you need to select the area. To do this, you first select a point on the edge of the object. Now you can place several points around the object.

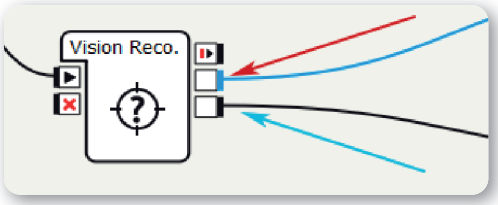


Screen 8.33 – Object attributes



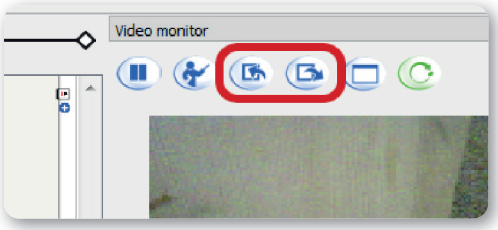
Screen 8.34 – Upload Database

Once you have marked around the edge of the object, click on the starting point again. Now give your object a name and, if you want, a side. Once you have clicked on Save, you need to upload your database – or what you just loaded into your computer's cache – to the robot.



Screen 8.35 – Outputs of the "Vision Recognition" box

If NAO recognizes the object it issues a signal at the top output. If it does not recognize it, NAO issues a signal at the bottom output.



Screen 8.36 – Import/Export Database

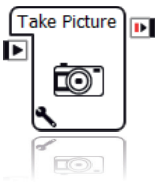
If you want to save the database, click on Export "Vision Recognition" Database. To upload a saved database, click on Import "Vision Recognition" Database and select a file.

Note:

Only one database can be saved on NAO at once. You should never forget to upload the database to NAO. The signal issued is the name of the object.

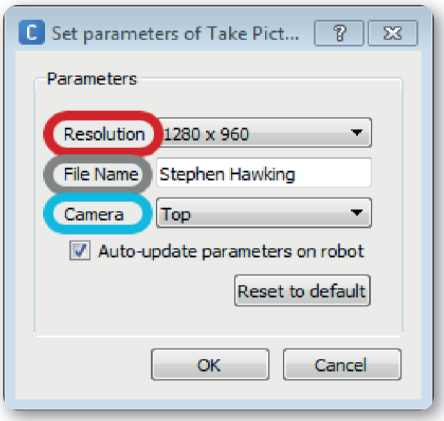
What you need:

An object or an image (also works very well with black and white images).



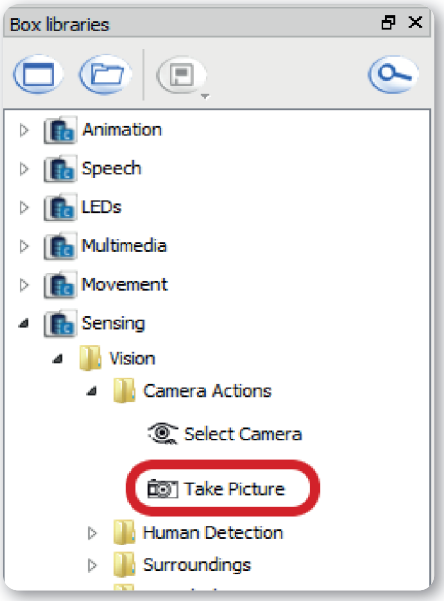
16. The “Take Picture” box

Explanation:
You can use this box to make NAO take a picture with one of its built-in cameras.



Screen 8.37 – Parameters of the “Take Picture” box

You can set which resolution and which camera should be used to take the picture. You can also enter a file name.

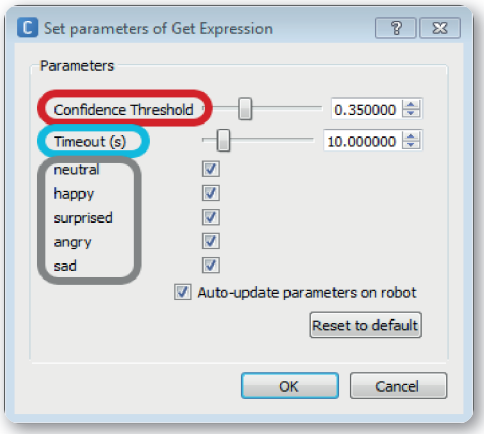


Screen 8.38 – Where is the “Take Picture” box?

Note:
Use the “Take Picture” box from the Sensing section. Do not use the one from the Animation section.

17. The “Get Expression” box

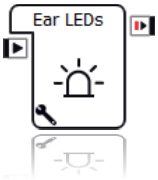
Explanation:
This box allows NAO to recognize your expression.



Screen 8.39 – Parameters of the “Get Expression” box

You can use the Confidence Threshold to set how confident NAO must be before it issues a signal.
The threshold works like this: NAO compares certain templates with the facial expressions to be recognized. Several points in the face are considered. For example, where the corners of the mouth are when laughing. If there are 10 points in the face for an expression, at a setting of 0.5 or 50%, NAO must determine five points correctly.

Once the timeout expires, NAO stops looking for an emotion and issues a signal at the error output. You can also set which emotions NAO should recognize.

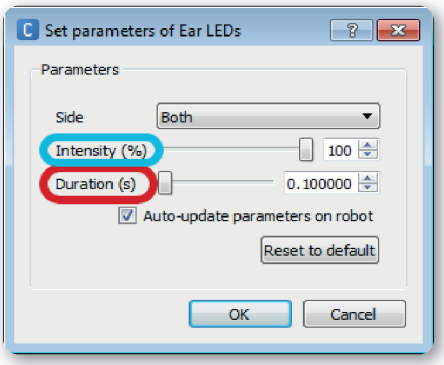


18. The “LED” boxes

Example:
NAO's ear LEDs are lit up.

Fig. 14 – Ear LEDs are lit up

Explanation:
You can use the “LED” boxes to control NAO's lights. There is a separate box for each light. So you can use the Eye “LED” box to make NAO's eyes light up, the Ear “LED” box for its ears, or you can use the Single Eye “LED” box to make a specific eye and the Single Ear “LED” box to make a specific ear light up. Plus, you can control specific LEDs. To do this, use the Set “LED”s box or the Set Single “LED” box.

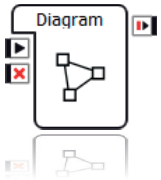


Screen 8.40 – Parameters of the Ear “LED”s box

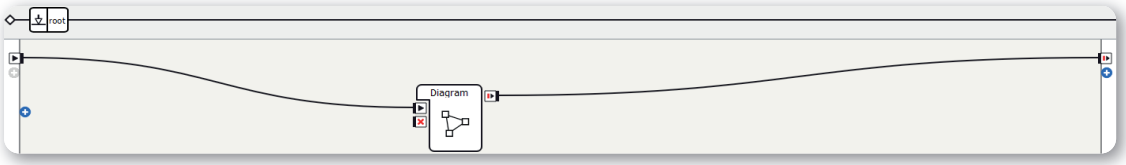
In the boxes, you can set the duration for which the LEDs light up and sometimes also the intensity.

Note:
You can find the numbers of the individual LEDs in the documentation. You need these numbers for the Set Single “LED” box, for instance.

19. The “Diagram” box

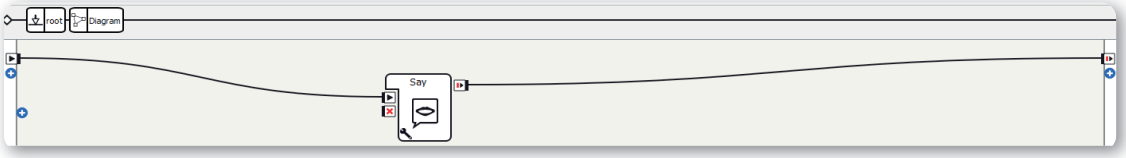


Explanation:
You can use this box to summarize your project in clear sections.



Screen 8.41 – The “Diagram” box

The inputs of the “Diagram” box correspond to the inputs inside the “Diagram” box. An output inside the “Diagram” box leading out of it represents the output of the “Diagram” box.



Screen 8.42 – Inside the “Diagram” box

The onStop input ends all processes inside the “Diagram” box. For example, this resets the “Counter” box or the “Only Once” box. So all cached values are reset to 0.

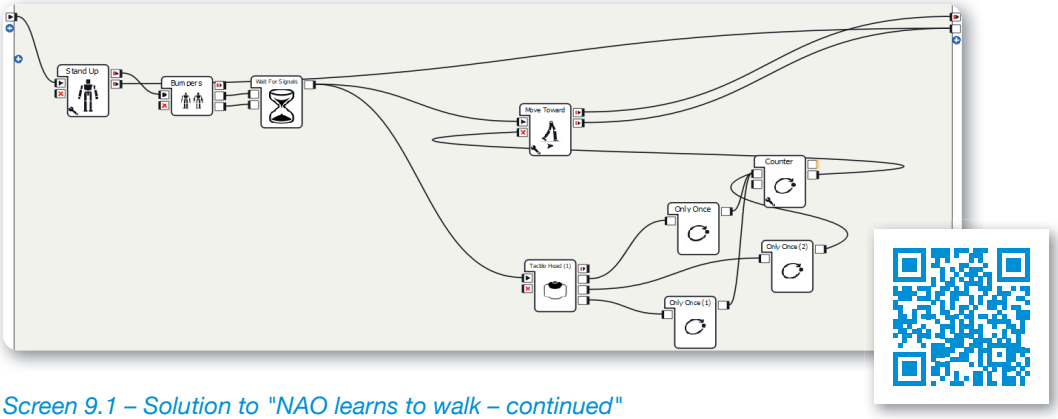
C. 3. 2. Moderate examples

1. NAO learns to walk – continued

Aim: First let's expand the program from C.2.2.5. so that NAO walks until all three head sensors have been touched.

You need these boxes: ***[Boxes from task C.2.2.3.]; [Boxes from task C.2.2.5.];*** "Only Once" ***(see C.3.1.8.);*** "Counter" ***(see C.3.1.6.)***

What to do:



Screen 9.1 – Solution to "NAO learns to walk – continued"

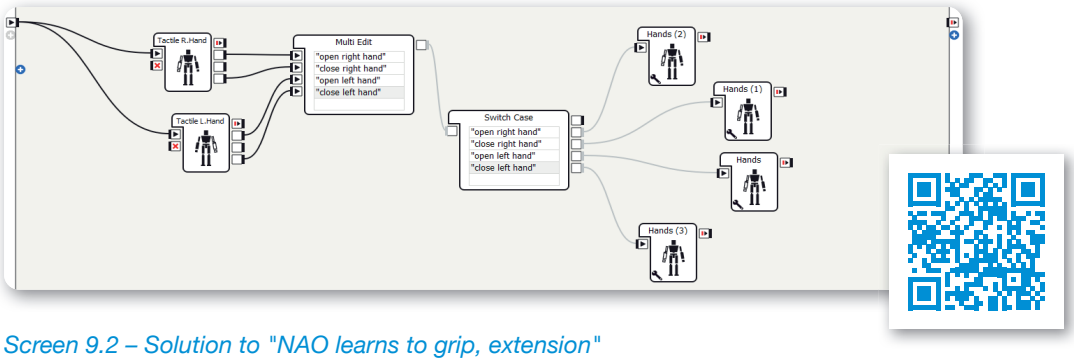
Drag three "Only Once" boxes and one "Counter" box into the project. Now connect the boxes as shown in the screenshot. Set the "Counter" box so that three Input signals are needed until an onReintitialized output is triggered. (To do this, you have to set the Initial Value to 0, the Step Value to 1, and the Final Value to 3.)

2. NAO learns to grip – extension

Aim: Now let's expand the program from C.2.2.4. so that the signals are converted into words (string signals) using a Multi Edit box and then a case distinction is made using a "Switch Case" box.

You need these boxes: ***[Boxes from task C.2.2.4.];*** "Switch Case" ***(see C.3.1.9.);*** "Multi Edit" ***(see C.3.1.3.)***

What to do:



Screen 9.2 – Solution to "NAO learns to grip, extension"

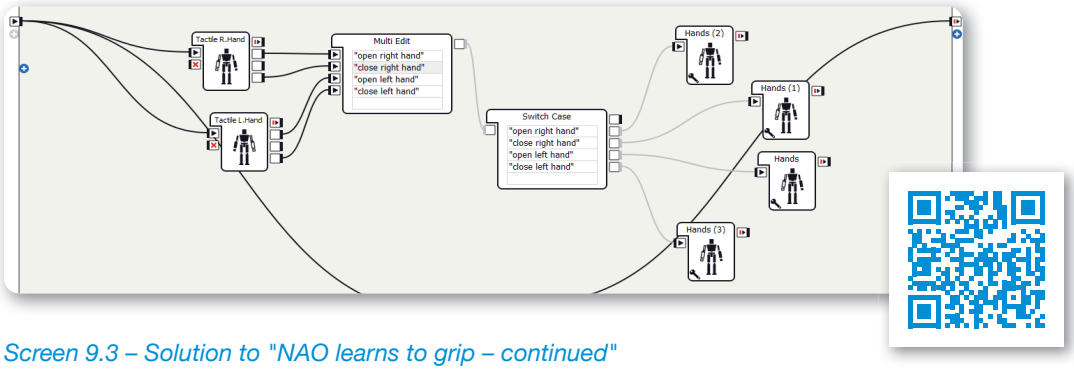
Drag a “Multi Edit” box and a “Switch Case” box into the project. Now connect the boxes as shown in the screenshot and enter the parameter values if the boxes (see above).

3. NAO learns to grip – continued

Aim: Change the project from 2. so that NAO stops the program itself after 20 seconds.

You need these boxes: ***[Boxes from task C.2.2.4.]; [Boxes from task 2.];*** “Timer” ***(see C.3.1.5.)***

What to do:



Screen 9.3 – Solution to "NAO learns to grip – continued"

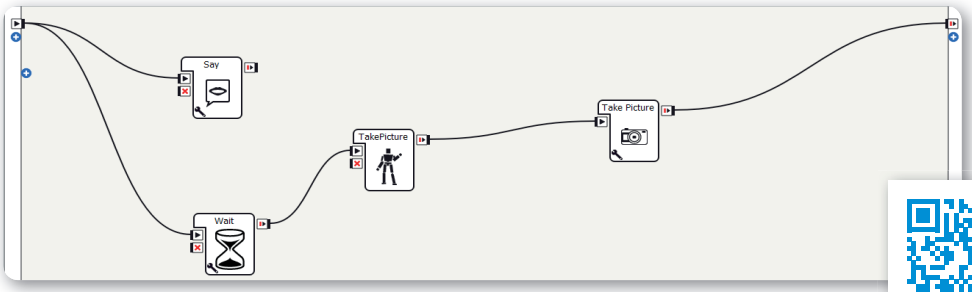
Drag a “Timer” box into the project. Connect the boxes as shown in the screenshot. Now set 20 seconds inside the “Timer” box.

4. NAO as a cameraman

Aim: Now let's make NAO take a picture with a delay.

You need these boxes: “Wait” ***(see C.3.1.4.);*** “Take Picture” ***(see C.3.1.16.)***

What to do:



Screen 9.4 – Solution to "NAO as a cameraman"

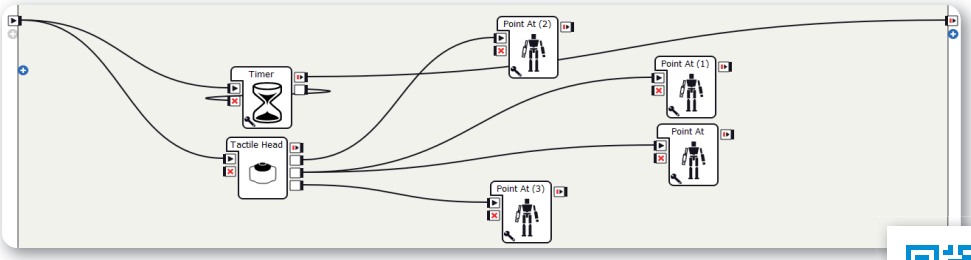
Drag the “Wait” and “Take Picture” boxes into the new project. The “Take Picture” timeline and the “Say” box have only been entered to illustrate the delay. Now connect the boxes as shown in the screenshot. Next, set a delay of your choosing.

5. NAO responds

Aim: The next project is intended to make NAO lift its hands when it is touched on the head. It should lift its right arm up to the side when it is touched on the rear head sensor. When the front sensor is touched NAO should raise its left arm up to the side. When the middle head sensor is touched, it should lift both hands to the front. Here again, the program should be ended after a certain time.

You need these boxes: Tactile Head ***(see C.2.1.7.);*** “Timer” ***(see C.3.1.5.);*** “Point At” ***(see C.3.1.12.)***

What to do:



Screen 9.5 – Solution to "NAO responds"

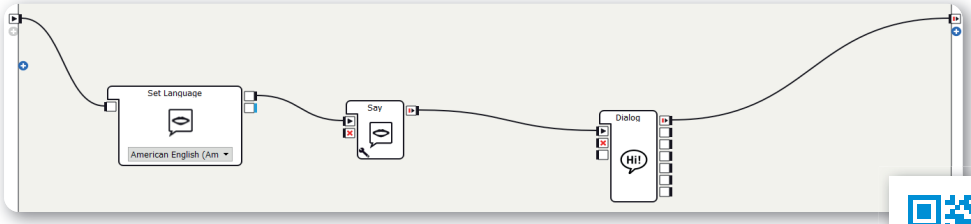
Drag the Tactile Head, “Timer”, and “Point At” boxes into the new project. Connect the boxes as shown in the screenshot.
Now vary the Y- and Z-coordinates of the “Point At” boxes to set the positions as you want them.

6. NAO can do it all – part 1

Aim: Now that we have created several individual programs, we want to join them together. To do this, we'll use the program from C.2.2.1. This end program should make it possible for NAO to call up selected individual programs when asked.

You need these boxes: ***[Boxes from task C.2.2.1.];*** “Dialog” ***(see C.3.1.1.)***

What to do:



Screen 9.6 – Solution to "NAO can do it all"

Drag a “Dialog” box into the project and open it.

```
topic: ~ExampleDialog()
language: enu
u:(e:onStart) say the codeword!
u:(walk)$walk=1
u:(take a picture) $takeapicture=1
u:(respond) $respond=1
u:(Timeline) $timeline=1
u:(arms) $arms=1

u:(e:input2) hey that was cool!
```

Screen 9.7 – Dialog for "NAO can do it all"

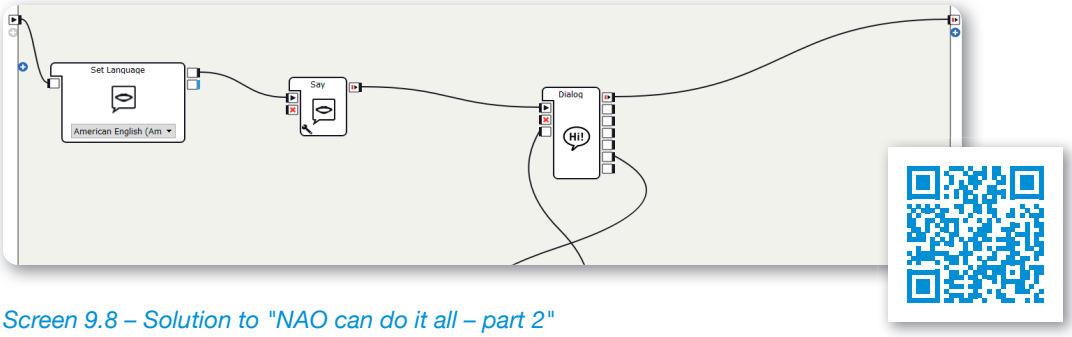
Write this or a corresponding source code in the “Dialog” box and connect the boxes as shown. Now create the outputs and the input mentioned.

7. NAO can do it all – part 2

Aim: We will now add a timeline or animation, either that you have made or from the Pose Library, to the program from task 6.

You need these boxes: ***[Boxes from task 6.];*** Elephant ***(from the Pose Library)***

What to do:



Screen 9.8 – Solution to "NAO can do it all – part 2"

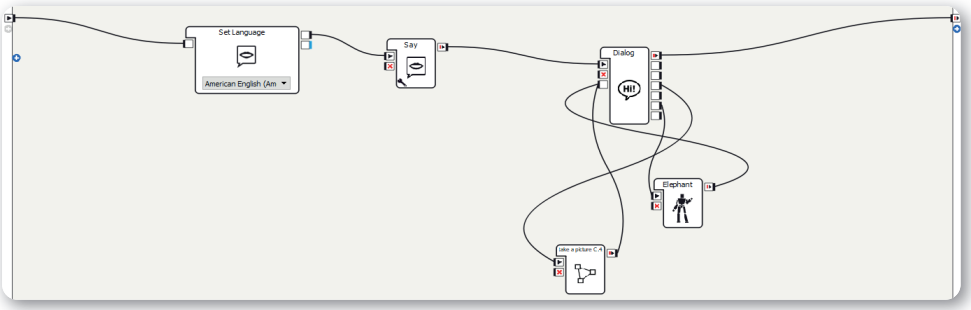
For demonstration purposes, we have used an entertaining animation called "Elephant" from the Box Libraries.
Connect the relevant output of the “Dialog” box to the animation and the output of the animation to input2 of the “Dialog” box.

8. NAO can do it all – part 3

Aim: To avoid confusion and keep your project from task 6 clear, all selected individual programs are packed in “Diagram” boxes. We recommend giving the “Diagram” boxes exactly the same names as the corresponding output. Now add all subprograms that you created before into the program.

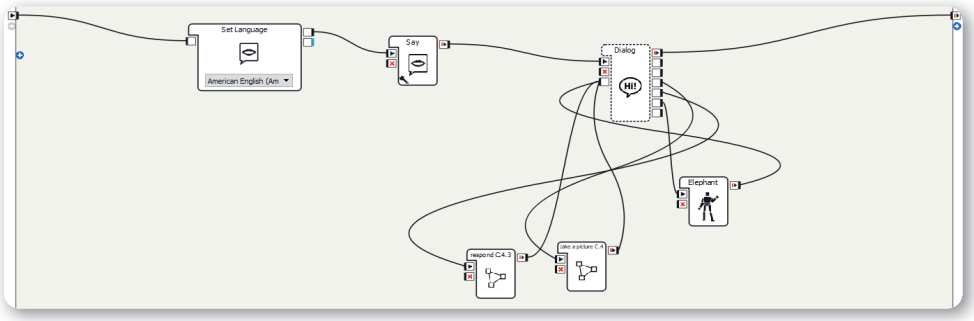
You need these boxes: ***[Boxes from task 6.]; [Boxes from the relevant tasks];*** “Diagram” ***(see C.3.1.19.)***

What to do:



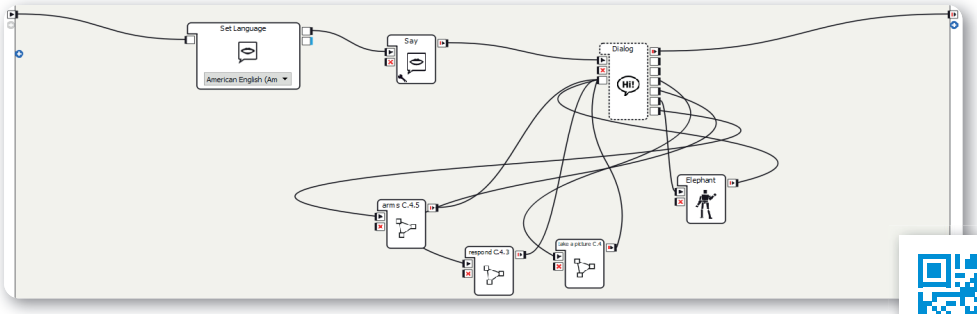
Screen 9.9 – Solution to "NAO can do it all – part 3" with C.3.2.4.

First you insert the program "NAO as a cameraman" from ***C.3.2.4.*** into a diagram and connect the boxes accordingly.



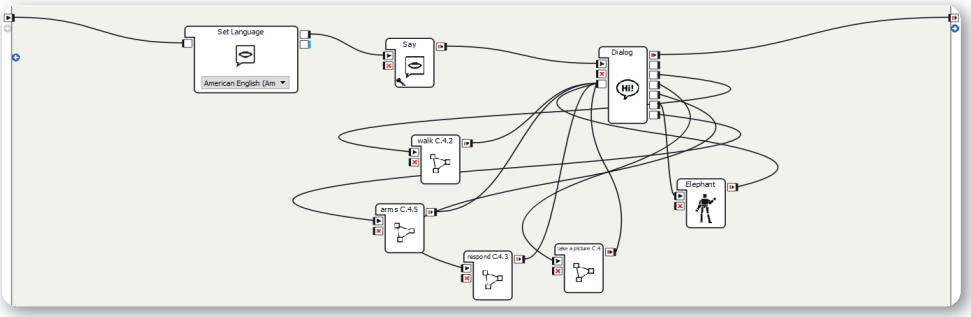
Screen 9.10 – Solution to "NAO can do it all – part 3" with C.3.2.3.

Then insert the program "NAO learns to grip – continued" from ***C.3.2.3.*** into a diagram and connect the boxes accordingly.



Screen 9.11 – Solution to "NAO can do it all – part 3" with C.3.2.5.

Next insert the program "NAO responds" from ***C.3.2.5.*** into a diagram and connect the boxes accordingly.



Screen 9.12 – Solution to "NAO can do it all – part 3" with C.3.2.1.

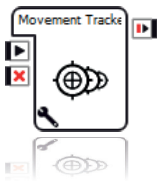
Finally, insert the program "NAO learns to walk – continued" from ***C.3.2.1.*** into a diagram and connect the boxes accordingly.

Now there are no limits to your creativity. Add more programs or expand the existing ones.

C. 4. Level 3: Expert

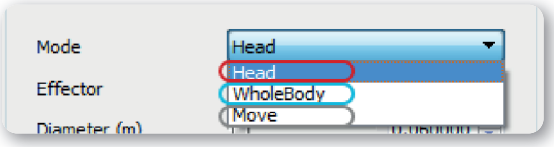
C. 4. 1 Difficult boxes

1. The Tracker boxes



The Tracker boxes make NAO follow different objects, movements, and targets. You can set several parameters here. Not all settings are available for every box. All the possible parameters are listed below.

The Mode parameter



Screen 10.1 – The Mode parameter

Here you can set how NAO should track the target. NAO can watch it (Head), NAO can turn on the spot so it doesn't lose sight of the object (WholeBody), or NAO can walk after the object (Move).

The Effector parameter

Screen 10.2 – The Effector parameter

Here you can set whether and how NAO should show that it has noticed the object or movement. Select None if you want NAO to give no indication. Otherwise you can select whether NAO should point to the object or movement with both arms (Arms) or only with the right arm (RArm) or the left arm (LArm).

The Diameter parameter

Screen 10.3 – The Diameter parameter

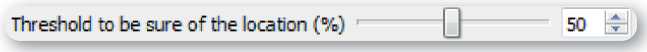
This parameter is only available with the Red Ball Tracker. You can use this to enter the diameter of the ball. This is essential, otherwise NAO could identify other red objects as the ball.

The Size parameter

Screen 10.4 – The Size parameter

You only need this parameter with the LandMark Tracker. You have to set the size of the LandMark here.

The "Threshold to be sure of the location" parameter



Screen 10.5 – The "Threshold to be sure of the location" parameter

This parameter is only available with the Sound Tracker. It allows you to set how sure NAO needs to be that it has actually detected the sound before it issues a signal.

The Width parameter

Screen 10.6 – The Width parameter

You can use this parameter with the Face Tracker box to set the width of the face to be recognized.

The Sensitivity parameter

Screen 10.7 – The Sensitivity parameter

This parameter is only available with the Movement Tracker and the Sound Tracker. You can use it to set how strong the movement or sound must be for NAO to recognize it. The higher the sensitivity, the smaller or quieter the movements or sounds that are recognized.

The Distance/Threshold X parameter

Screen 10.8 – The Distance/Threshold X parameter

These parameters are available with several boxes. Here you enter the distance in meters that NAO should maintain from the target on the X-axis (horizontally). With the Threshold you can also specify the area in which the object can be located at Distance X without NAO moving. (This means: NAO goes to the object and maintains the Distance ± the Threshold.)

The Distance/Threshold Y parameter

Screen 10.9 – The Distance/Threshold Y parameter

This parameter is very similar to the X-parameter. The only difference is that this variable represents the deviation from the object to the right or left. That's why there are also negative (left from NAO's perspective) and positive (right from NAO's perspective) setting options.

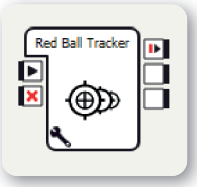
The Theta/Threshold Theta parameter

Screen 10.10 – The Theta/Threshold Theta parameter

Theta describes the angle within which the object may be located. Threshold again describes the maximum permitted deviation from the specified values.

The following section gives a brief explanation of what the different Tracker boxes do.

a) Red Ball Tracker



Screen 10.11 – The Red Ball Tracker box

With this Tracker, NAO can follow a red ball. Once it has reached the ball, it stands still as shown in the picture.

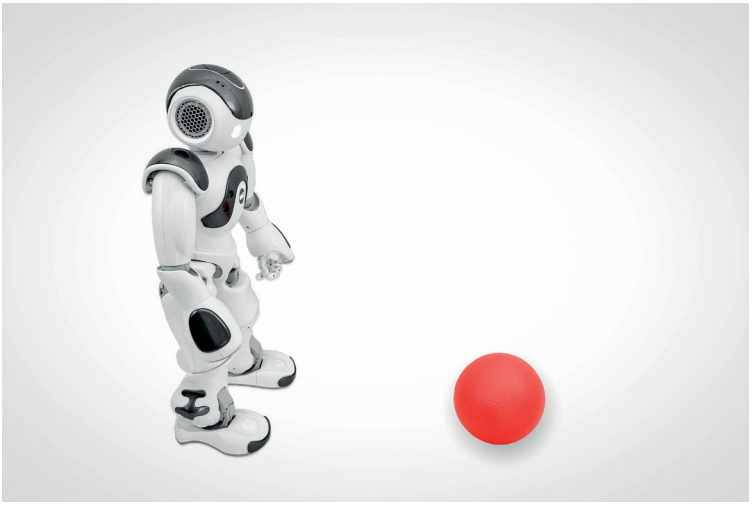


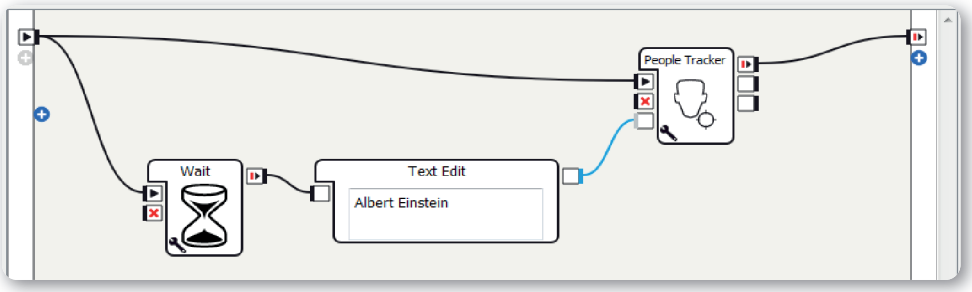
Fig. 15 – NAO in front of the red ball

b) Sound Tracker

Screen 10.12 – The Sound Tracker box

Make NAO follow a sound with this Tracker.

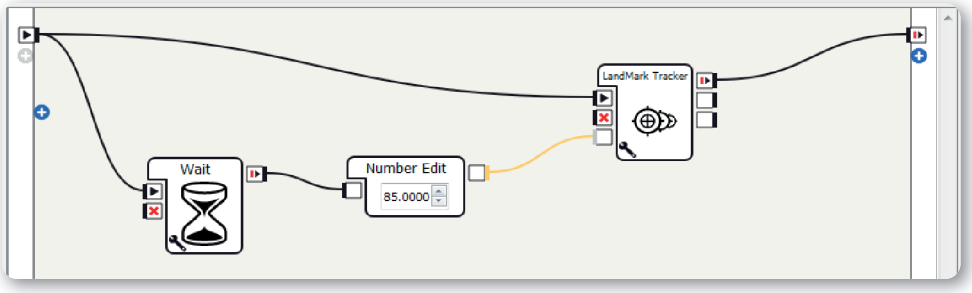
c) People Tracker



Screen 10.13 – Flow diagram with the People Tracker box

This Tracker makes NAO follow certain faces. To do this, you need to specify the ID or name of the face that was saved previously with the “Learn Face” box. Use the Text Edit box here. However, the text can only be sent once the Tracker has been started. To do this, put a Wait box in front of the Text Edit box (see screenshot).

d) LandMark Tracker



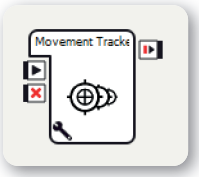
Screen 10.14 – Flow diagram with the LandMark Tracker box

This Tracker makes NAO walk to a precisely defined Naomark. You need to specify the ID or number of the LandMark for this. To do so, use a Number Edit box. Here again, the number can only be sent once the Tracker has been started. To do this, put a Wait box in front of the Number Edit box again (see screenshot).

The LandMark Tracker works in exactly the same way. However, you can enter several LandMarks here.
When NAO reaches the LandMark, it stands still in front of it.

Fig. 16 – NAO in front of a LandMark

e) Movement Tracker



Screen 10.15 – The Movement Tracker box

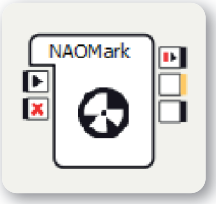
This Tracker detects movements in NAO's surroundings. Other than the Start input, no other inputs are necessary.

f) Face Tracker

Screen 10.16 – The Face Tracker box

If you want NAO to recognize and follow an unspecified face, you need this Tracker. Here too, no other inputs are necessary other than the Start input.

2. The “NAOMark” box



Screen 10.17 – The “NAOMark” box

You can use this box to make NAO recognize Naomarks. Naomarks are round symbols with holes inside a circle. Each Naomark has its own number. When it is detected, this number is also issued as the output signal at the yellow output.

You can really experiment and play around with this box. You can use it for all kinds of things in your projects and programs.

What you need:
Naomark (you can download these from the SoftBank website or the virtual archive that accompanies the book).

3. The “Python Script” box

You can use the Python Script box to insert your own individual boxes, mechanisms, and complex programs in Choregraphe.
However, you should only use the box if you already have some knowledge of Python and have worked with it.
Here is some information on the specific Choregraphe methods, events, and invocations:

`"def onLoad":`

Is invoked when the “Python Script” box is loaded, which means when the diagram containing the “Python Script” box is invoked.

`"def onUnload":`

Is invoked when the box is unloaded (see above).

`"def onInput_onStart":`

Is invoked when a signal arrives at the "onStart" input.

`"def onInput_onStop":`

Is invoked when a signal arrives at the "onStop" input.

`"def onInput_[Inputname]":`

Is invoked when a signal arrives at the [Inputname] input.

Variables within a method (also known as "local") are defined via variablename = "test" and variables within the entire Python script (also known as "global") are defined via self.variablename = "test".

You can create your own inputs and use them in the Python source code by defining a method:

```
def onInput_*Inputname*(self):pass
```

You can activate outputs by invoking this method:

```
self.*Outputname*
```

If the input signal has a data type other than "Bang", the transfer parameter p can be used in this way:

```
def onInput_*Inputname*(self, p): Variable = p
```

If you add a parameter to the “Python Script” box and want to use it, you can use this method:
self.getParameter("*Parametername*")
If you want to use an API, or a subprogram provided by a software system of other programs for connection to the system, you need to do this via a value assignment:

(http://doc.aldebaran.com/2-8/naoqi/index.html)

```
tts = ALProxy("ALTextToSpeech")
tts.say("Hello world")
```

You need these commands to show messages in the debug log:

`self.log("my message")`	Message with the significance level "info"
`self.logger.fatal("my message")`	Message with the significance level "fatal"
`self.logger.error("my message")`	Message with the significance level "error"
`self.logger.warning("my message")`	Message with the significance level "warning"
`self.logger.info("my message")`	Message with the significance level "info"
`self.logger.debug("my message")`	Message with the significance level "debug"

You can find more detailed information in the SoftBank documentation:

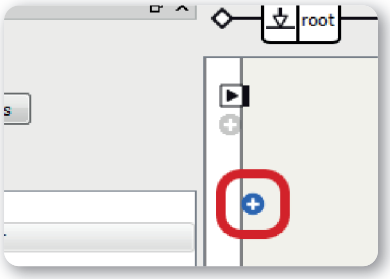
http://doc.aldebaran.com/2-8/software/Choregraphe/objects/python_script.html

You can also use other programming languages, such as C++, Javascript, and ROS via Cmake. More information on this is available in the NAOqi documentation:

http://doc.aldebaran.com/2-8/dev/programming_index.html#other-sdks

4. The “Subscribe to Event” box

An event is triggered once its conditions are fulfilled. For example, the event "robotHasFallen" is triggered as soon as the robot falls.



Screen 10.18 – Add event

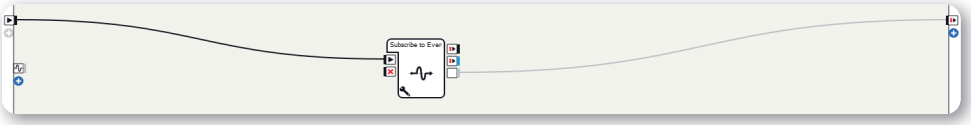
You can access an event by clicking on the bottom plus on the left of the flow diagram and entering an event name.

If you want to create a new event, you can click on "Add a new key…" and enter an event name.

The “Raise Event” box triggers an event as soon as it receives an input.

Screen 10.19 – The “Raise Event” box

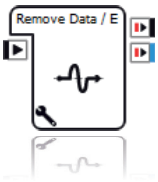
You need to enter the event name using the wrench.
The onStopped output issues the value of the previous input.



Screen 10.20 – The “Subscribe to Event” box

The “Subscribe to Event” box is triggered as soon as the event set using the wrench is invoked. The onEvent output issues the value of the event.

The blue onError output issues the error message.

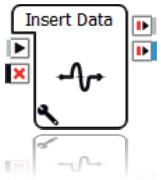


5. The “Remove Data/Event” box

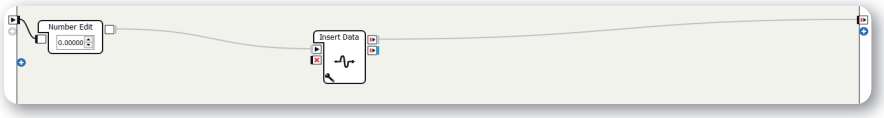
Screen 10.21 – The “Remove Data/Event” box

This box deletes the value in the specified file path within your project, such as an event or a number value.

6. The “Insert Data/Get Data” box



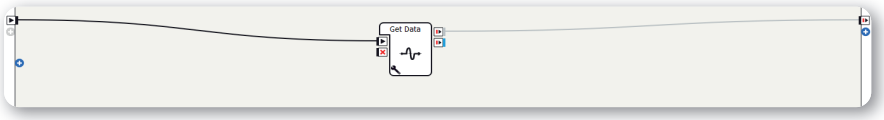
You can use the “Insert Data” box to save data directly to NAO.



Screen 10.22 – The “Insert Data” box

You need to set the save location in the "key" field via the wrench.
If you want to save a value now, route the value to the onStart input using a Number Edit box, for example.
The box then saves the value in the save location set.

You can use the “Get Data” box to read out a value saved on NAO.



Screen 10.23 – The “Get Data” box

As soon as the box is triggered, it reads the value of the save location and issues the value found via the onStopped output.

The blue onError output issues the error type as a string.

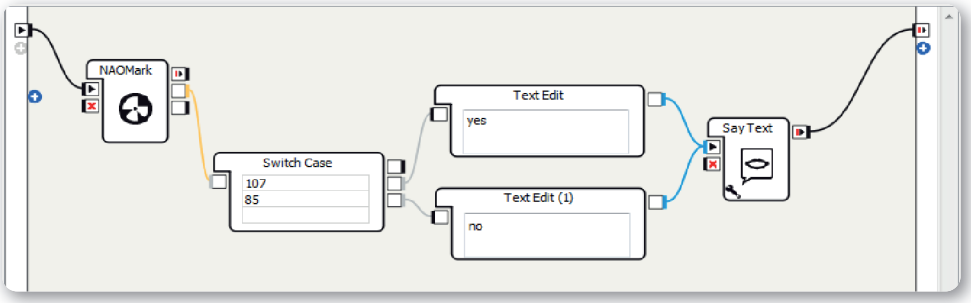
C. 4. 2 Difficult examples

1. NAO's choice

Aim: Make NAO say "yes" or "no" depending on the Naomark it detects.

You need these boxes: NAOMark, “Switch Case” ***(see C.3.9.)***, “Text Edit” ***(see C.3.3.),***
“Say Text” ***(see C.1.1.)***

What to do:



Screen 11.1 – Solution to "NAO's choice"

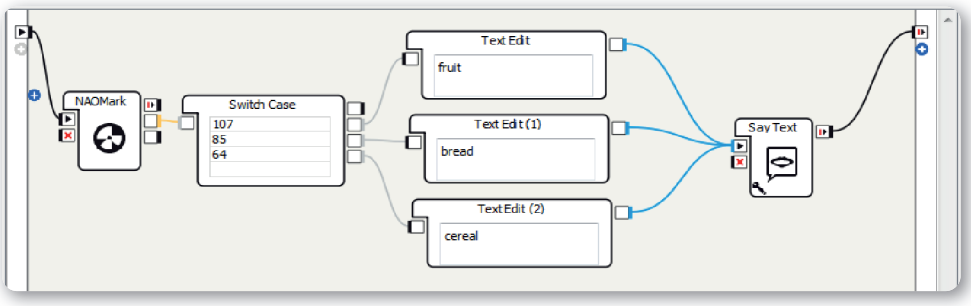
Drag the boxes NAOMark, “Switch Case”, “Say Text”, and two “Text Edit” boxes onto your flow diagram panel. Connect them as shown in the screenshot. Write the numbers of the two different Naomarks in the “Switch Case” box. Now all you need to do is enter "yes" and "no" in the Text Edit boxes.

2. NAO as a warehouse operative

Aim: Make NAO recognize the content of boxes (e.g. when used as a warehouse robot).

You need these boxes: NAOMark, “Switch Case” ***(see C.3.9.),*** “Text Edit” ***(see C.3.3.)***,
“Say Text” ***(see C.1.1.)***

What to do:



Screen 11.2 – Solution to "NAO as a warehouse operative"

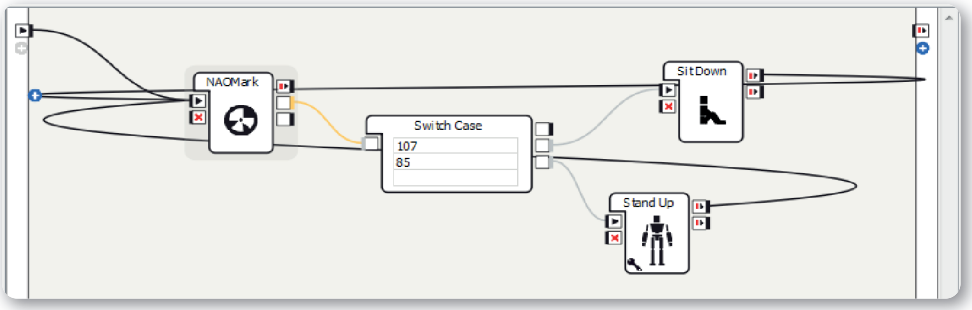
Drag the boxes NAOMark, “Switch Case”, “Say Text”, and three Text Edit boxes onto your flow diagram panel. Connect them as shown in the screenshot. Write the numbers of the three different Naomarks in the “Switch Case” box. Now all you have to do is enter the content of the boxes, for example "Fruit", "Muesli", and "Bread" into the Text Edit boxes.

3. NAO is controlled remotely

Aim: Make it possible to control NAO using Naomarks. First make it able to sit down and stand up.

You need these boxes: NAOMark, “Switch Case” ***(see C.3.9.)***, “Sit Down” ***(see C.1.6.),*** “Stand Up” ***(see C.1.5.)***

What to do:



Screen 11.3 – Solution to "NAO is controlled remotely"

Drag the boxes NAOMark, “Switch Case”, “Sit Down”, and “Stand Up” onto your flow

diagram panel. Connect them as shown in the screenshot. Write the numbers of the two different Naomarks in the “Switch Case” box.

Now you have reached the end of our book. If you read and worked through the book carefully, you should now have a comprehensive basis for realizing your own projects. This book is a milestone, not just for you but also for us. Realizing our dream of writing our own book has been an enriching experience for us.
Teamwork was another important aspect in this cooperative project. As in every major collaboration, mutual trust is essential for working successfully and it meant that through this really close cooperation, we evolved from classmates to friends.

We would like to express our warmest thanks for all the support we received from:

Heike Schnaubelt (OStR, senior councilor of studies) ***(Direction)***	for her encouragement, ideas, perseverance, management, and above all for arranging this project
Tony Schuster (StR, councilor of studies) ***(Editor)***	for his critical and helpful analysis
Mark Lörz (OStD, senior director of studies) ***(Headmaster)***	for his openness and open-mindedness in realizing the project
Markus Nerlich ***(Managing Director, Technik-LPE GmbH)***	for enabling this project
Christiane Schulz ***(Head of Marketing & Sales, Technik-LPE GmbH)***	as our constant contact through the joint cooperation
Markus Wagner ***(Graphic designer, Technik-LPE GmbH)***	for his advice and compiling the content of our book

Finally, thank you for reading our book and working through it – hopefully you're now motivated to take a look at our second book "Do it NAO[6] - Creative Project Ideas".

Neu-Ulm, November 2018


Kai Anter


Marcel Greiner

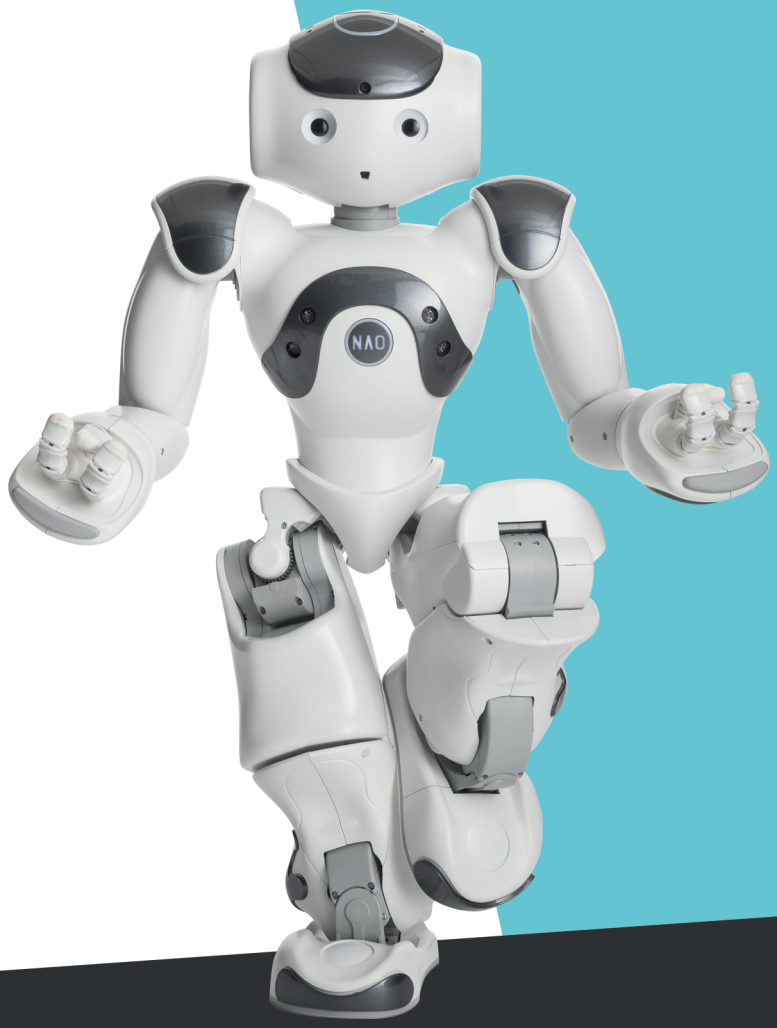

Jonas Vatter


Jannes Weghake

SoftBank Robotics

EXPAND YOUR COMPETENCE

SCIENCE | TECHNOLOGY | ENGINEERING | MATHEMATICS |



More information:
www.softbankrobotics.com/emea

Contact:
emea-sales@softbankrobotics.com

SoftBank Robotics Europe
NAO Robot

@SBREurope
@NaoRobot
@PepperTheRobot

SoftBank Robotics Europe

@sbreurope

SoftBank Robotics Europe