LEGO Mindstorms - leJOS and SCCharts

Program LEGO Mindstorms with leJOS and SCCharts

Overview

Mindstorms is a product family from Lego, with sensors, motors and a programmable brick. The newest iteration of the product family is the EV3 programmable brick. Its predecessors are NXT and RCX. In the following we will see how to develop applications for the NXT brick.

Several open-source, third-party replacements for the offical Lego firmware have been developed. These support many well known programming languages, such as Java, C/C++, Python, Lua, etc. In the following we will use KIELER SCCharts to program Mindstorms running the Lego Java Operating System (leJOS). Therefore we will first install leJOS NXJ and flash its firmware. Afterwards we will create a simple SCCharts project in KIELER that we will compile and deploy to the NXT brick.

If you want to learn the SCCharts langugage first, you can follow these links:

- Introduction to SCCharts
- SCCharts Syntax
- SCCharts Examples

Download and install leJOS

Download and extract the newest archives for your Operating System from Sourceforce (Linux/Mac) or use the Setup.exe (Windows).

The further installation is explained in detail at https://lejos.sourceforge.io/nxt/nxj/tutorial/Preliminaries/GettingStarted.htm.

Do not forget to flash the download leJOS firmware to the Mindstorms brick as explained in the tutorial!

After installing leJOS make sure that all commands are available in the PATH.

Make sure that this is also the case for gcc or javac/java if you plan to simulate your model.

You might have to start KIELER from the console to access all comands in the PATH.

Known issues or remarks

Linux

You can switch your currently used java version via **sudo update-alternatives --config java**

You can find a guide to install leJOS and all dependencies here: https://wiki.ubuntuusers.de/LeJOS_f%C3%BCr_Lego_NXT/. You need to add the LEJOS_NXT_JAVA_HOME environment variable (which should point to your java 8 installation).

You can flash your NXT using KIELER if you configure the LEJOS_HOME in KIELER via preferences>lejos>...

On Linux there is an issue when uploading the firmware because of a kernel module (http://ubuntuforums.org/showthread.php?t=1123633). If you can't upload the firmware with your Linux OS, add **`blacklist cdc_acm`** at the very end of the file **`/etc/modprobe.d/blacklist.conf`**. Afterwards execute **`sudo rmmod cdc_acm`**. This will remove the cdc_acm module from the kernel and prevent its restart. Now try to flash the firmware again.

Another issue is that the development package of **libusb** has to be installed. On Ubuntu you can do this by using **`sudo apt-get install libusb-dev`**

(Only necessary if your current java jdk has a higher version than java 8): If the ant tasks fails because jni.h is missing edit the build.xml in line 12 from <condition property="jni.include.dir" value="${java.home}/../include"> to <condition property="jni.include.dir" value="${java.home}/include"> if you use java 11 or later, since this was build for java 8 where the java executable was in /jre/bin and not in /bin.

Furthermore, to use USB connection, a java library has to be compiled via **ant**. To do this perform **cd /path/to/leJOS/build** and start **ant**. If the ant build tool is not installed on your system, you can do so via **sudo apt-get install ant**.

On Linux there is an issue that the NXT is found, but uploading any file to it does not progress after **Found nxt <name>**. Try to connect via bluetooth (-b option) if this is the case.

Windows

leJOS requires a 32 java installation as mentioned in the leJOS setup guide, Java 8 is confirmed to work.

The setup.exe of the current LEGO Fantom driver for Windows (1.2.0) has an awkward issue. If you get an error message (Developer Error) because an .msi file could not be found, don't panic. The file is part of the downloaded archive (in the Products folder) but you have to start it manually. You may also try to install the necessary drivers by following the Guide to install on Windows 10 on Reddit.

If you get a runtime error when starting the setup.exe also don't panic. The file is part of the downloaded archive (in the Products folder) but you have to start it manually.

If installing the fantom driver fails or does not work try to connect via bluetooth.

Mac OS X

The **leJOS NXJ** tools require a **32 Bit** version of Java. However, newer 32 Bit versions of Java are not longer available for Mac. Thus to use leJOS the installation of **Java 1.6 is required**, which is the last one that supports a 32 Bit mode. You can download the installer for Java 1.6 from https://support.apple.com/kb/dl1572?locale=en_US. It will install Java 1.6 to */System/Library/Frameworks/JavaVM.framework/Versions/1.6.0/Home* so that the environment variable *LEJOS_NXT_JAVA_HOME*, which is set in the installation instructions, points to the correct path.

The environment variable LEJOS_NXT_JAVA_HOME is set in the installation tutorial by editing ~/.profile. However, on a Mac the environment variables defined in this file are not visible for GUI Applications, only for apps started from terminal. Thus to use leJOS together with KIELER, one either has to start KIELER from terminal or set the environment variable so that all GUI applications can access it. However, this does not seem to be trivial on Mac (see also http://stackoverflow.com/questions/135688/setting-environment-variables-in-os-x).

Test the Mindstorm

A simple Hello World application for the Mindstorms is developed as part of the leJOS tutorial.

If get a compilation error, set the java compliance level (target and source are necessary if you are using java 11 or something like that and have not set *LEJOS_NXT_JAVA_HOME* to a java 8 installation).

```
nxjc -target 1.8 -source 1.8 HelloWorld.java
```

If this works with your device, you are able to start using KIELER to develop applications for the NXT brick. However you might still need a java 1.8 or lower, since KIELER does not add target and source per default.

Renaming your NXT brick

If you plan to use bluetooth to connect to your NXT it might help to change the name of your NXt brick.

You can do that via the **nxjbrowse** command (if USB does not connect add the -b option).

Select your NXT and click connect.

Via *Set Name* the name of your brick can be changed.

Download and Configure KIELER

Download and unpack the nightly build of KIELER for your OS. It is available at the Downloads page. No configuration is needed.

You do not need the preinstalled NXJ plugin in KIELER, but leJOS has to be installed and all commands have to be in the PATH.

> You may want to start KIELER from the terminal otherwise the nxj commands may not be visible.

Creating an Example Project

The following shows how to create a project, which will turn on a light if a button is pressed.

Create a new project:

1. Choose *File* > *New* > *Project* > *General* > *Project* (do **never** convert your project in an Xtext-project)
2. The project is created.
3. Create a SCChart inside the project.
4. Open the SCCharts modeling perspective (*Window* > *Perspective* > *Open Perspective* > *Other)*
5. Edit the model
6. Compile the model using the KIELER compiler (using the *Netlist-Based Deployment (NXT)* compilation system)

Edit the Model:

Change the contents of the model file to the following code and save it.

Floodlight.sctx

```
scchart Floodlight {

    input bool @macro "Button", "ENTER" button

    output bool @macro "Floodlight", "S1" light

    initial state lightOff
    if button do light = true go to lightOn

    state lightOn
    if !button do light = false go to lightOff
}
```

This model will start in the state lightOff. If the button is pressed, it will turn on the light and change to the corresponding state, where the light is turned off, as soon as the button is not pressed anymore.

The annotations on the input and output variable are used to define which wrapper code is used to set / read them. **@macro "Button", "ENTER**" will set the input variable to true iff the orange enter button is pressed. **@macro "Floodlight", "S1**" on the output variable will turn on the red led of the light sensor that is attached to port S1 iff the variable is true.

The available wrapper code snippets are defined below.

Note: The Floodlight of the EV3 has a pretty high latency when switching between on and off.

Compile the model:

Select the Netlist-Based Deployment (NXJ) or Netlist-Based Deployment (NXJ) Bluetooth to deploy to the NXT.

This compilation chain is accessible via the KIELER Compiler View.

Available Wrapper Code Snippets

There are several wrapper code snippets that can be used as annotations on input and output variables in the model file. These snippets are inserted in the main file template as part of the project build. The available snippets are listed here.

Using the Remote Console (RConsole)

The display of the **NXT brick** is rather small compared to a Monitor. To ease debugging, one can print to a Remote Console (RConsole). This enables easier collection for example of sensor data.

Use the RConsole compilation system to deploy to the NXT (Currently called *Netlist-Based Deployment (NXT) via RConsole*). Start the **nxjconsoleviewer** tool. Now, when **starting the application**, the brick tries to connect with the nxjconsoleviewer. **Press the *Connect*** button. If connected succesfully, RConsole.println(...) commands will be written to this window.

To stop this nxt is this mode the ENTER and ESC button (orange and dark grey) have to be pressed at the same time. Yu have to disconnect and reconnect via the **nxjconsoleviewer** tool if you restart the application.

The **EV3 brick** has a similar feature. However it does not require any code changes. Just run the ev3console program in the bin directory of your leJOS installation from command line. The output of the brick will be printed to this command line.

Problem Solving

The following presents typical issues and how to solve them.

Issue	Typical Error Messages	Description	Solution
leJOS EV3 does not support Java 8	"java.lang. UnsupportedClassVersionError" "unsupported major. minor version"	You compile the sources in your project with Java 8 and upload them to the brick. However the lejos EV3 does not support Java 8	Go to the project properties and switch to Java 7 (Right Click on project > Properties > Java Compiler > Compiler compliance level)
Uploading to the brick does not respond		You compile a file successfully and when uploading the result, the connected brick is found. Anyway the upload does not terminate and does not react.	Flash the brick with the current leJOS firmware. If the brick is recognized correctly and the attempt to upload a compiled file fails then the firmware on the brick might be outdated. If connecting via bluetooth works add -b to the uxjupload/nxj command.
Compilation and uploading works from command line but not when using KIELER	This Java instance does not support a 32-bit JVM. Please install the desired version.	You can compile and upload code to the brick using the command line tools but when using KIELER an error message apprears because Java does not support 32-bit JVM.	Set the LEJOS_NXT_JAVA_HOME environment variable, such that it points to an 32-bit JDK and is visible for GUI applications (or at least KIELER). The process to do so differs on every OS. As alternative, execute KIELER from terminal.
Brick does nothing after program finished and prints "Program exit"		A program was uploaded and finished without errors. Afterwards the brick prints "Program exit" but does not open the main menu.	This is normal behaviour if uploading a program in debug mode instead run mode (*Debug As* instead *Run As* in Eclipse). To get back to the main menu, press the ENTER and ESCAPE button of the brick at the same time.
Cannot simulate but compile	Javac/gcc not found	Javac and gcc have to be available in the path	Add javac/gcc to your path
Cannot simulate but compile	Cannot find file	KIELER has no rights to write to the file system	Use a workspace in which KIELER has write access
LEFT and RIGHT button cannot be pressed at the same time		If you press the ENTER and LEFT button both are true, if you press the ENTER and RIGHT button both are true. If you press the LEFT and RIGHT button only the LEFT button is true.	Do not write a program that requires the LEFT and RIGHT button to be presssed at the same time.
Too many variables	No more than 255 fields expected	If your generated Java Code uses more than 255 variables it cannot run on the NXT.	Try using less super states or concurrent regions (during actions create concurrent regions).
KIELER and compilation use two different java versions	Main has been compiled by a more recent version of the Java Runtime (class file version 55.0), this version of the Java Runtime only recognizes class file versions up to 52.0		Make sure KIELER uses the same java version as the one in the path (for example Java 11). Lejos can still use java 8, if you set the *LEJOS_NXT_JAVA_HOME* variable (in windows this is a 32-bit java installation)
NXT program cannot be stopped in RConsole mode		You deploy a program to the NXT with RConsole enabled	Press Esc (dark grey button) and Enter (organge Button) at the same time