nxcBoost

What is nxcBoost?

NxcBoost is a personal library with a variety of useful features which should make writing programs in NXC easier and more efficient. If I had used some functions a lot in my project, I saved them in the library. When the number of functions starting to get more and more I wrote a documentation by using the tool “doxygen”. With this publication, the library is made available to everyone.

What is the goal?

Many things are rather complicated to program in nxc. For example one of those things is the text output on the LCD. You have to think carefully how to divide the space so you do not overwrite any data. This can be very complex in big projects. The easiest solution is for example the console.
Other functions are not good implemented in the NXC-API. This includes the motor approximation. If you want e.g. turn the motor exactly 90 degrees with 100% power then the motor often overshoots the gool with about 50 degrees. The library offers here an alternative with two motor controllers.

How are the funtions?

Exept for the GUI module, all functions are written such that, unlike NXC API functions, the first letter is lowercase. Further more a user can use a functions easily but also use a extended version. That means whenever you see a function definition with a "=" then you do not have to pass this variable to the function. Take a look at this example output function with a console:

Definition: *cout(string output, unsigned long time = NA, bool endl = NA)*

Usage: *cout(“Hello World”);*

or: *cout(“Hello World ”, SEC_1, false);*

In the first usage, the so-called standard parameters are used. (here NA)
You now probably wondering what NA is...
This simply means that in advance another function can determine what you would like to use as standard parameter in this case.

Anything else?

The library is open source. This means you are allowed to edit, upgrade and even publish the whole project again. (Taking into account the GNU GPLv2)
If you want to do so, please inform me previously before you publish your project.

Some codes were taken from other projects and customized for nxc and the library.
I would like to thank David Gilday and RWTH Archen for the source codes of their motor controllers!

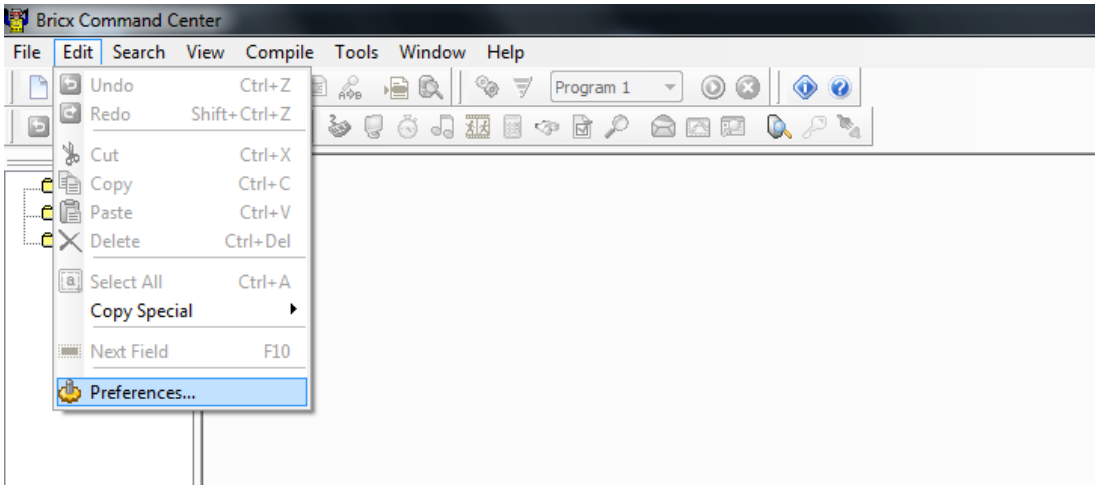
I recommend that you take a look at the tutorial.
There are also some info further down.

Tutorial - Getting started

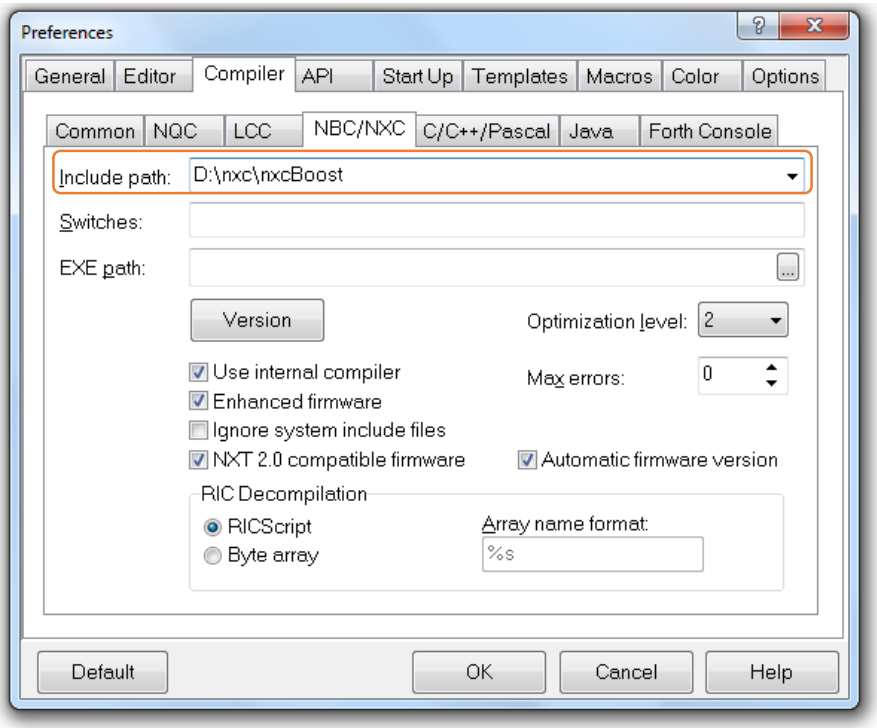
First download the whole nxcBoost folder and save it to your desired directory.

You have to show BricxCC the location of nxcBoost folder so that you can use all functions uncomplicated in your projects.

Go to Edit -> Preferences.



Write or copy the path to your nxcBoost folder into the line as shown in the picture.
(Of course it must not be the same as mine!)

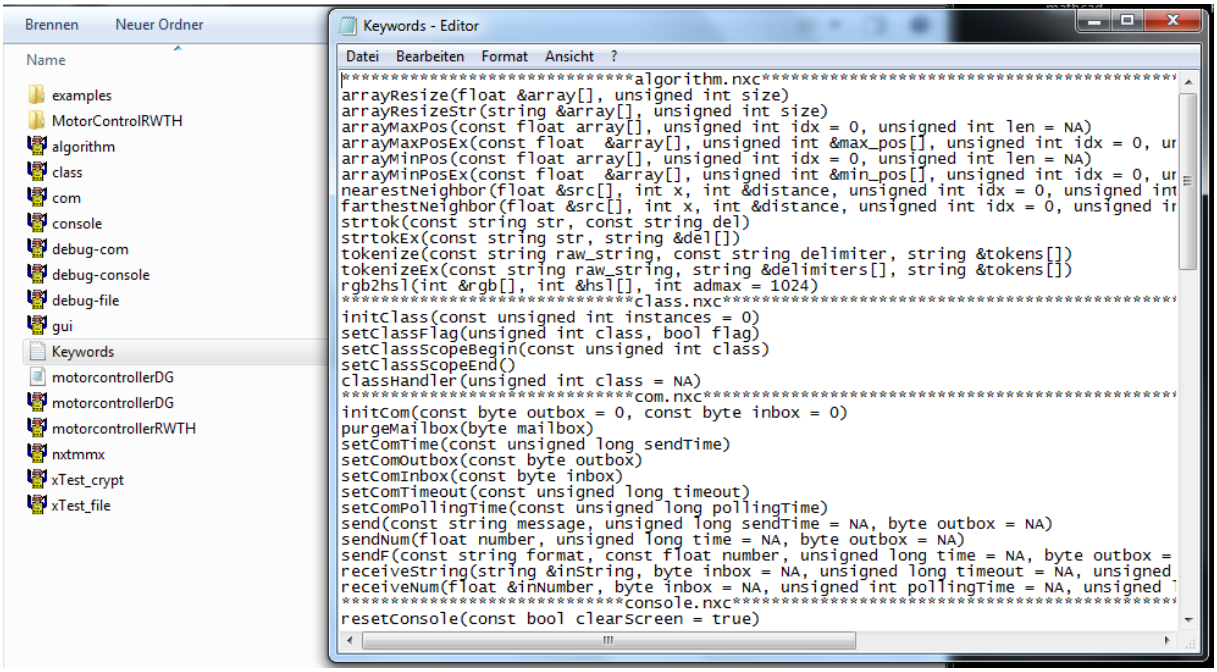


This here is optional:

You surely know that NXC-API functions which are getting highlighted if you write them right and you can see the function parameters if your write "(".

This is called "auto-complete" and it can be very useful when you do not know exactly which parameters you have to pass. It often saves time because you do not have to look at the documentation.

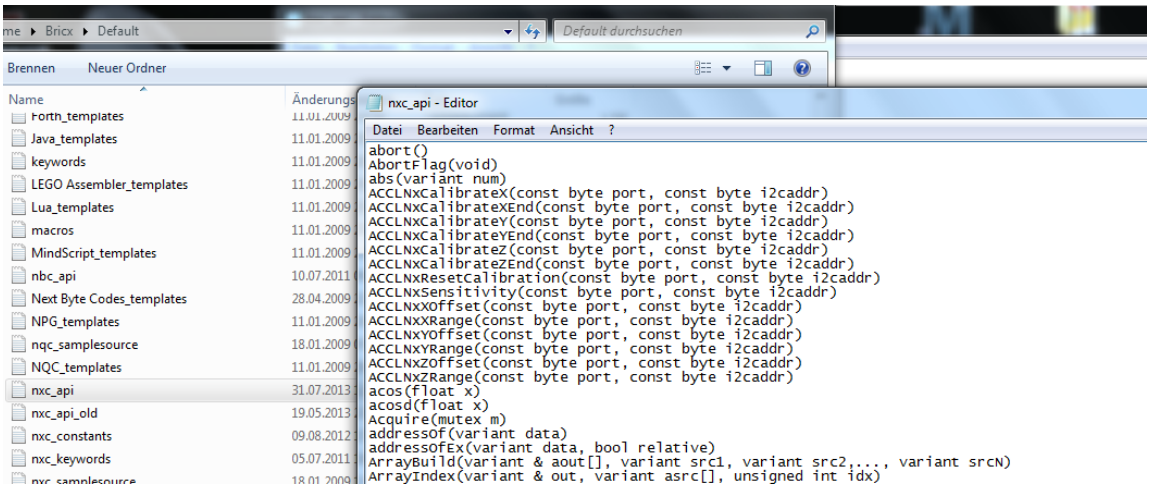
There is a text document in the nxcBoost folder with the name "keywords". Open this document.



Navigate to the directory where you have installed BricxCC.

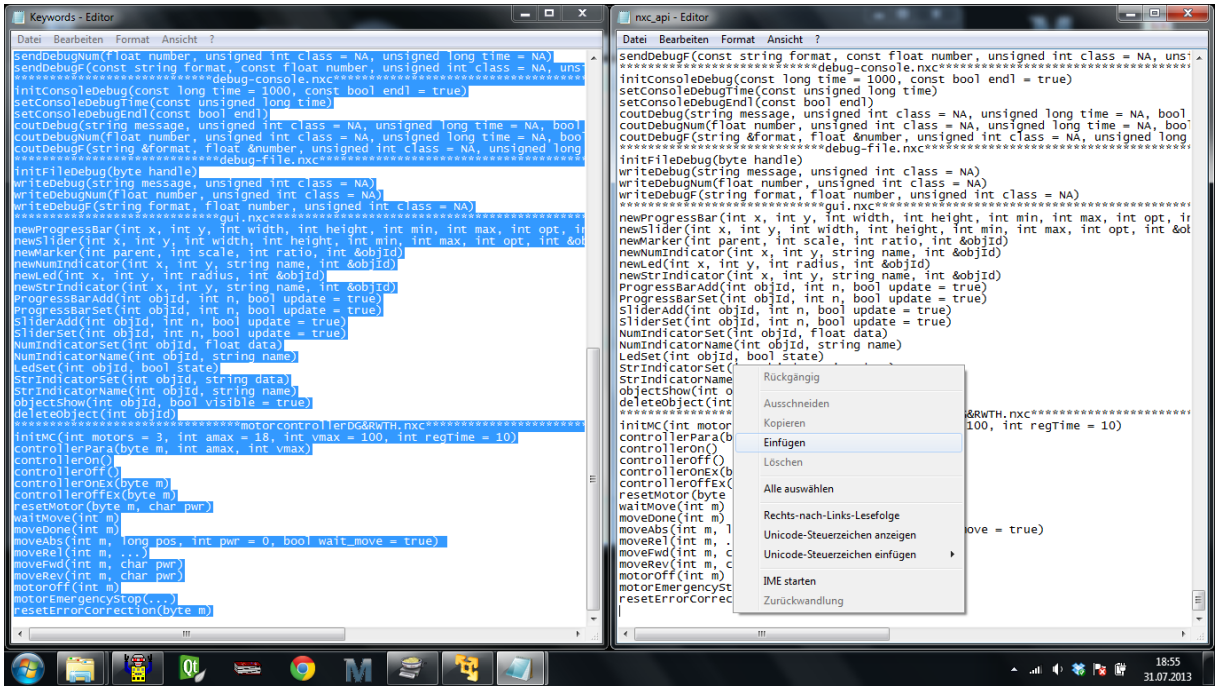
Of course, this does not have to be same as my path. (eg also C :/ Program Files / BricxCC)

Head to the directory "default" (at the top). Open the file "nxc_api".



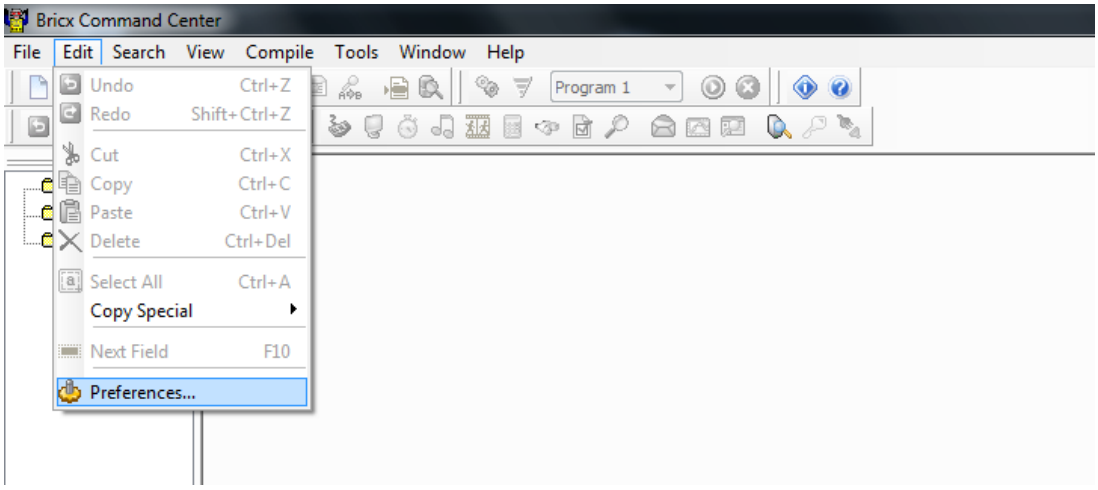
Tip: It's good to make a copy of the file "nxc_api" and save it under the name "nxc_api_save" in the same directory.

You can quickly undo everything if you rename the file back to "nxc_api".
Now you copy all the text from "keywords" and append it to the file "nxc_api".

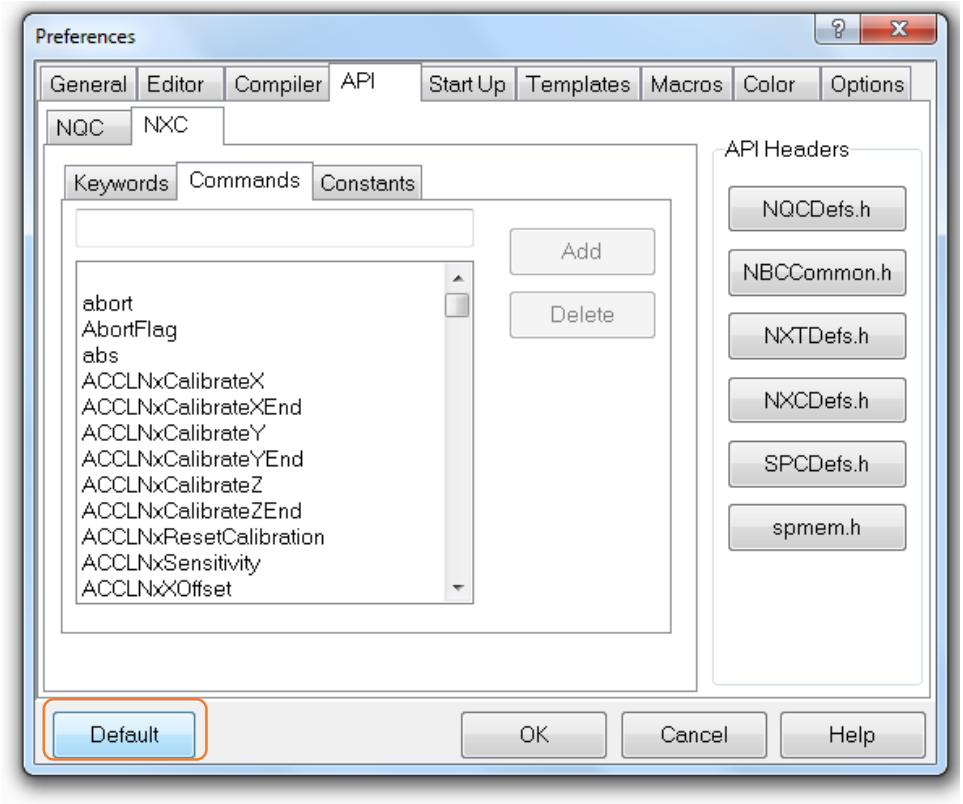


Save the file "nxc_api" and open BricxCC.

Navigate to the settings:



Go to tab API / NXT and press the “Default” button then “OK”.



Try if it works: Simply open a new sheet and tap "cout".

Of course it is also possible to copy only the keywords of individual modules.
The modules are divided by stars "*".

So what now?

I would say, go to the "examples" folder and begin with "ex_console1". Some knowledge about the console would be not bad because it is used in many examples. Of course you need also the latest firmware!

Please report any bugs directly to me!