nBlocksStudio1.0 Installation

Table of Contents

nBlocksStudio1.0 Installation

nBlocksStudio1.0 is using mbed-cli infrastructure including GCC compiler, for the microprocessor code compilation.
The Graphical interface is written in Python and is accessing the n-Blocks.net server when it starts, to get the definition of the Nodes

Install Python

Both mbed-cli and nBlocksStudio are based on Python.

- Download 32 bit windows installer for Python 2.7 (dont use 64bit Python installer, issues with pip)
- **Configure installer for adding Python path to system path** and complete Python 2.7 installation
- Restart Windows (otherwise pip fails)
- Test Python and pip from command line:

```
Python --version
pip --version
```

Both should work from any directory
If pip version is later than 9.0.3 mbed-cli might not work [1], so change to 9.0.3:

```
pip install -U pip==9.0.3
```

Install mbed-cli manually

References: [2] [3] [4]

- Install Git:
  - Download Git [5]
  - Run the .exe (for windows) to install it
- Install Mercurial:
  - Download Mercurial [6]

    - Run the .exe (for windows) to install it
- Install GCC:
    - Download GCC [7)]
    - Run installer
    - **Add Path to environment variable** and complete installation
- Verify Installations:

```
arm-none-eabi-gcc --version
git --version
hg --version
python --version
pip --version
```

- Install mbed cli from command prompt:

```
pip install mbed-cli
```

- Verify mbed installation:

```
mbed --version
```

- 07-March-2020, install Microsoft Visual C++ Compiler for Python 2.7: [8)]

mbed tips

mbed online compiler how to use OS2 or OS5

References: [9)]

```
 To make sure you are using the desired library delete the mbed
library from your project in the online compiler and then
  import either the mbed-dev (if you would like to use mbed OS 2)
or the mbed-os (if you prefer mbed OS 5).
```

How to import an existing program from mbed online compiler to mbed-cli and compile it

- From online compiler right click on the project and publish (or FORK) in to something like this: frequency_counter

    - If the project is copy from somebody else's project, probably have to fork to your own mbed space
- With administrative rights(???):

```
 mbed import
https://developer.mbed.org/users/chalikias/code/frequency_coun
ter/
```

- for a list of supported targets:

```
mbed compile -S
```

- Finnaly:

```
mbed compile -m LPC1768 -t GCC_ARM
```

How to create new mbed-cli project for OS2 or OS5

References: [10)]

- new OS5 project:

```
mbed new project1
```

- new OS2 project:

```
mbed new project2 --mbedlib
```

- new OS5 project at current directory:

```
mbed new .
```

Alternative method to import OS2 program from mbed online compiler

I had issues with the import method described above. This Alternative method works.Tested for Target LPC11U35_401 [20 May 2020]

- From mbed online compiler right-click on a **Program** in the **Program Workspace**: Export Program using

```
Export
```

  Toolchain

```
ZIP archive (with repositories)
```

- Extract the ZIP file to a Temporary folder.
- Create a new folder as new mbed OS2 project. This folder will be the project working folder:

```
mbed new project_name --mbedlib
```

- **From Inside** the Temporary folder copy all files and folders.
- Paste the copied files and folders, to **project_name** folder created from **mbed new project_name –mbedlib**
  - Just overwrite existing files when Windows asks
- From Command line Compile:

```
mbed compile -t GCC_ARM -m LPC11U35_401
```

Update mbed OS to a specific version

- go to mbed-os directory inside the project forlder

```
cd mbed-os
```

- Check the active release:

```
mbed releases
```

- update to the desired release, for example: mbed-os-5.7.7

```
update mbed-os-5.7.7
```

- Check again for the active release:

```
mbed releases
```

Install n-Blocks Studio

- Install all needed Python modules:

```
pip install panda3d
pip install pywin32
pip install pyuserinput
```

- Download the studio from bitbucket, using command line:

```
 hg clone
https://nimbus_it@bitbucket.org/nblocksstudio/prj_nblocksstudi
o
```

- Or update an existing nBlocksStudio directory using command line, but from inside the directory

```
 hg update
```

- Copy **mbed_code** outside the version controlled directory, so it will not be affected by future updates of the studio
- Go to **the new mbed_code** folder with command line
- Create new mbed **OS2** project inside the new folder: (preferable for now, OS5 is not tested well yet with nStudio)

```
 ...\mbed_code> mbed new . --mbedlib
```

- Or a new mbed **OS5** project inside the new folder:

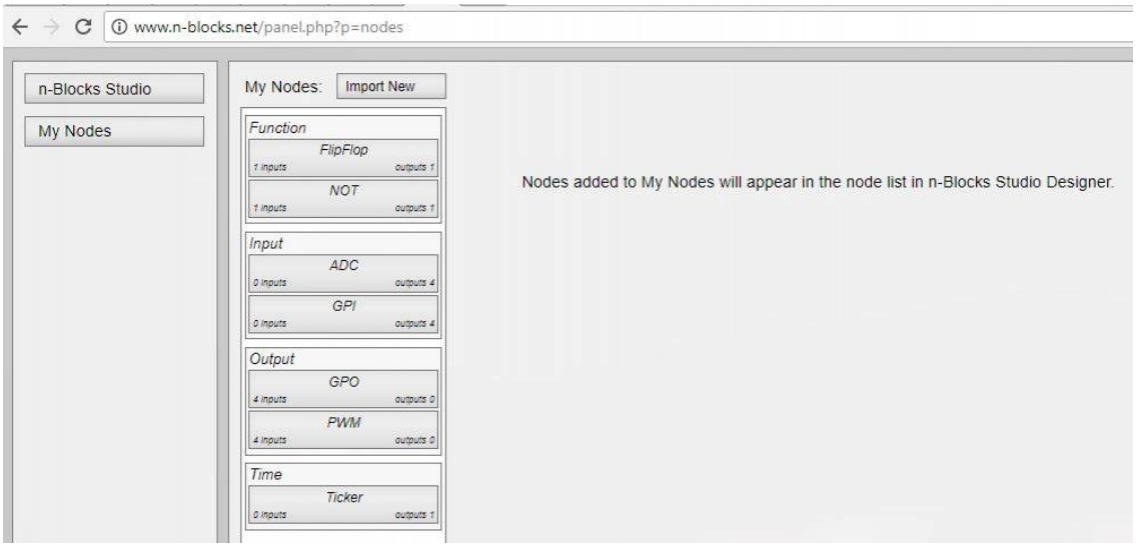
```
 ...\mbed_code> mbed new .
```

- Add any missing libraries:

```
 ...\mbed_code> mbed deploy
```

Use n-Blocks Studio

- Login to your nBlocksStudio [11)] account and add **nodes** to your account



- Start nBlocksStudio, from **Studio** folder with command line:

```
....\Studio\Launch.cmd
```

- On 07.02.2019 I had to edit nworkbench.py, to change[t]kinter to [T]kinter
- On 05.03.2020 this does not seem to be needed any more...

```
 from tkinter import TclError
```

to

```
from Tkinter import TclError
```

- ...and to install missing python modules:

```
 pip install enum
```

```
 pip install pywinusb
```

- ...one more March 2020:

```
 pip install requests
```

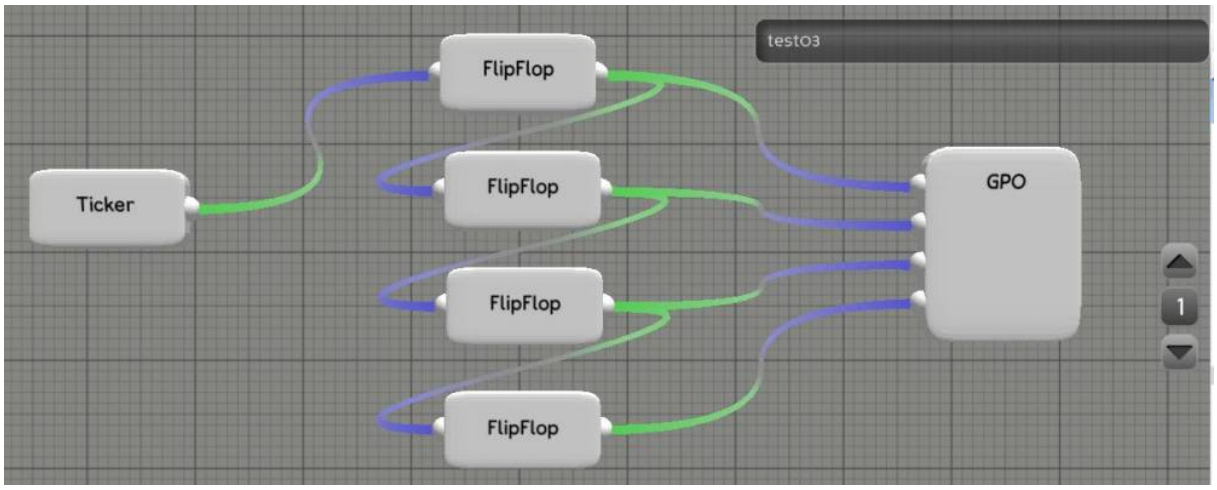
When exporting, save into the file **main.cpp** inside folder **mbed_code**, and build with:

```
C:....\mbed_code> mbed compile -t GCC_ARM -m LPC1768
```

Above creates a [.bin] file. (at ...\mbed_code\BUILD\LPC1768\GCC_ARM)
To automate the build process and use a different target n-Block/processor, call one of the **[tested]** batch files

- LPC1768.cmd (n-Bed)
- LPC11u35.cmd (n-DAP)
- L432KC.cmd (n-LP)
- L152RE.cmd (n-L15, n-CPU48)

4 bit binary counter example with n-Blocks Studio



This example diagram (a 4 bit binary counter) creates the main.cpp below:

```
/* ==================================================================
*
 *       Automatically generated by n-Blocks Studio Designer
*
 *
*
 *                          www.n-blocks.net
*
 * ==================================================================
*/
#include "nblocks.h"


// -*-*- List of node objects -*-*-
nBlock_Ticker                 nb_nBlockNode0_Ticker;
nBlock_FlipFlop               nb_nBlockNode13_FlipFlop;
nBlock_FlipFlop               nb_nBlockNode3_FlipFlop;
nBlock_GPO                    nb_nBlockNode12_GPO;
nBlock_FlipFlop               nb_nBlockNode14_FlipFlop;
```

```
nBlock_FlipFlop                nb_nBlockNode15_FlipFlop;

// -*-*- List of connection objects -*-*-
nBlockConnection          n_conn0(&nb_nBlockNode0_Ticker, 0,
&nb_nBlockNode3_FlipFlop, 0);
nBlockConnection          n_conn1(&nb_nBlockNode3_FlipFlop, 0,
&nb_nBlockNode12_GPO, 0);
nBlockConnection          n_conn2(&nb_nBlockNode3_FlipFlop, 0,
&nb_nBlockNode13_FlipFlop, 0);
nBlockConnection          n_conn3(&nb_nBlockNode13_FlipFlop, 0,
&nb_nBlockNode12_GPO, 1);
nBlockConnection          n_conn4(&nb_nBlockNode13_FlipFlop, 0,
&nb_nBlockNode14_FlipFlop, 0);
nBlockConnection          n_conn5(&nb_nBlockNode14_FlipFlop, 0,
&nb_nBlockNode12_GPO, 2);
nBlockConnection          n_conn6(&nb_nBlockNode14_FlipFlop, 0,
&nb_nBlockNode15_FlipFlop, 0);
nBlockConnection          n_conn7(&nb_nBlockNode15_FlipFlop, 0,
&nb_nBlockNode12_GPO, 3);


// -*-*- Main function -*-*-
int main(void) {
    SetupWorkbench();
    while(1) {
        // Your custom code here!
    }
}
```

References

1)
https://github.com/ARMmbed/mbed-os/issues/6763
2)
mbed 3 minutes quickstart video
3)
mbed os handbook
4)
mbed-cli: How to install manually
5)
download GIT
6)

download Mercurial
7)

download GCC from developer.arm.com
8)

http://aka.ms/vcpython27
9)

configure mbed-online for MBED-OS-2 or MBED-OS-5
10)

mbed-cli: creating and importing programs
11)

nBlocksStudio page

IMPORTANT NOTICE - PLEASE READ CAREFULLY

Nimbus Centre reserve the right to make changes, corrections, enhancements, modifications, and improvements to Nimbus Centre products and/or to this document at any time without notice.

All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

Address: Cork Institute of Technology Campus, Bishopstown, Cork

Phone: (021) 433 5560