Mbed 3.0 Development Environment Setup Howto

12-mbed\_devenv\_setup\_how\_to\_notes.docx

20151222

(previous version 11-yotta\_how\_to\_notes.docx (201506016))

Table of Contents

[Limitations 1](#_Toc413339336)

[Installation Requirements 2](#_Toc413339337)

[Installing On Windows 4](#_Toc413339338)

[Pre-requisites 5](#_Toc413339339)

[Install Git for Windows (git gui and git bash) 5](#_Toc413339340)

[Installing Python 2.7.9 for Windows 10](#_Toc413339341)

[Installing pyCryptio 20](#_Toc413339342)

[Install CMake 23](#_Toc413339343)

[Install Ninja 29](#_Toc413339344)

[Install arm-none-eabi-gcc 31](#_Toc413339345)

[Install Yotta 34](#_Toc413339346)

[Yotta commands 35](#_Toc413339347)

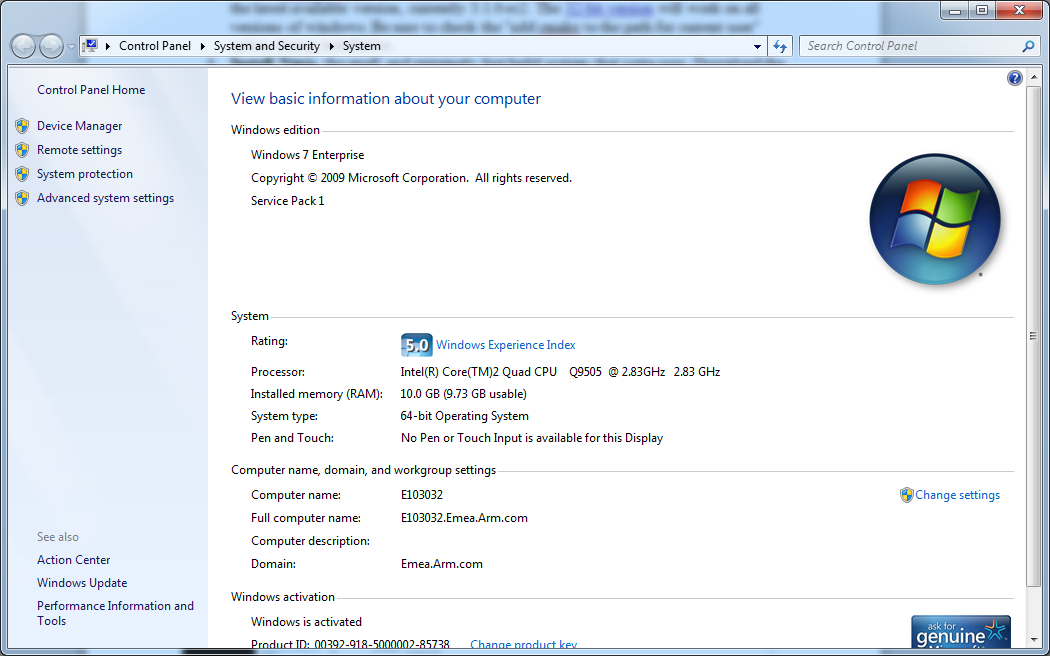
[Installing mbed-ls 38](#_Toc413339348)

# Limitations

There is a problem on windows that:

1. Python installs dlls into c:\Windows\System32, which mean the installation of all python files cannot be isolated in a specific install directory.
2. 2 versions of python cannot easily and cleanly be installed on the same machine.
3. The 32 bit and 64 bit versions of Python cannot be installed on the same machine because they install conflicting binaries in the c:\Windows\System32 dir, leading to unreliable behaviour that is difficult to track down.

This howto focuses on installing the 32 bit version of 1 version of Python only. For other installations of multiple versions see a different guide. The 32 vs 64 bit status of the OS is reported here:



Its fine installing Python 32bit version on a 64bit OS.

Due to 1 above, its not possible to install all the applications to a tools dir e.g. mbed\_tools, and expect to be able to move that directory from one machine to another. You have to run the installers to get the dlls in the system32 dir.

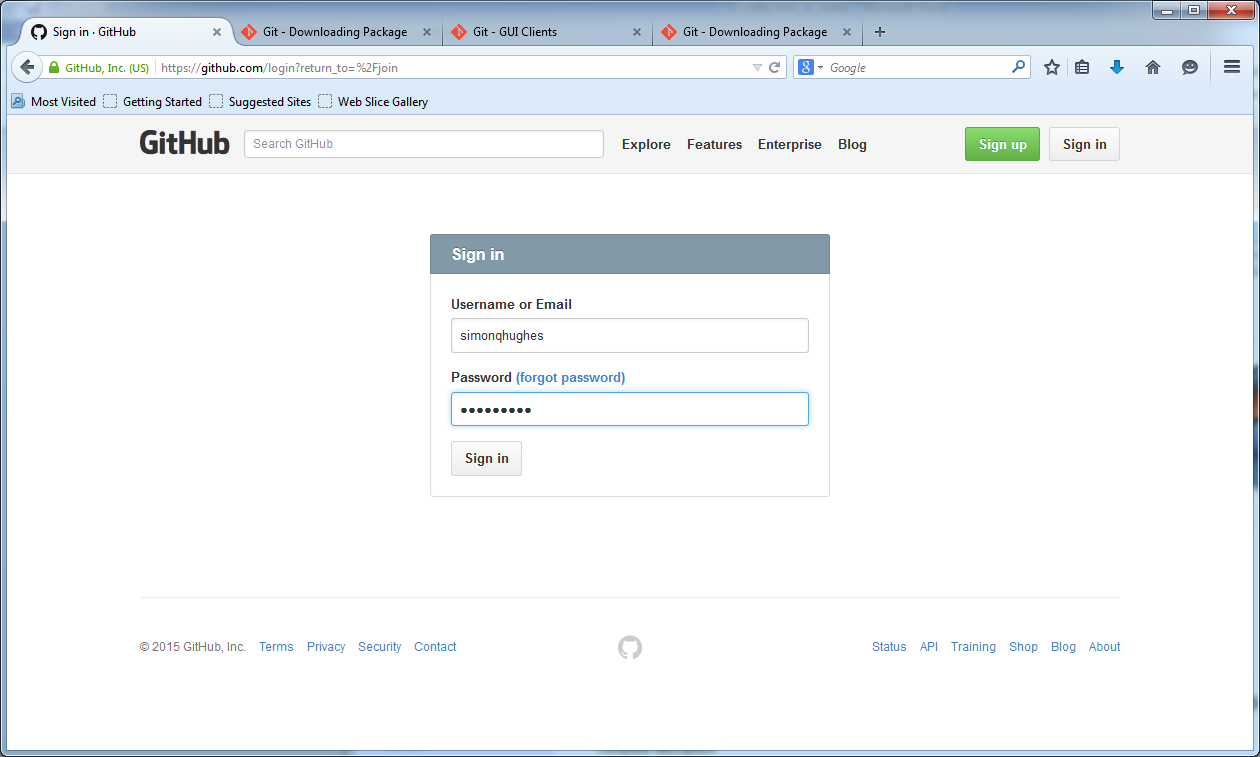
# Installation Requirements

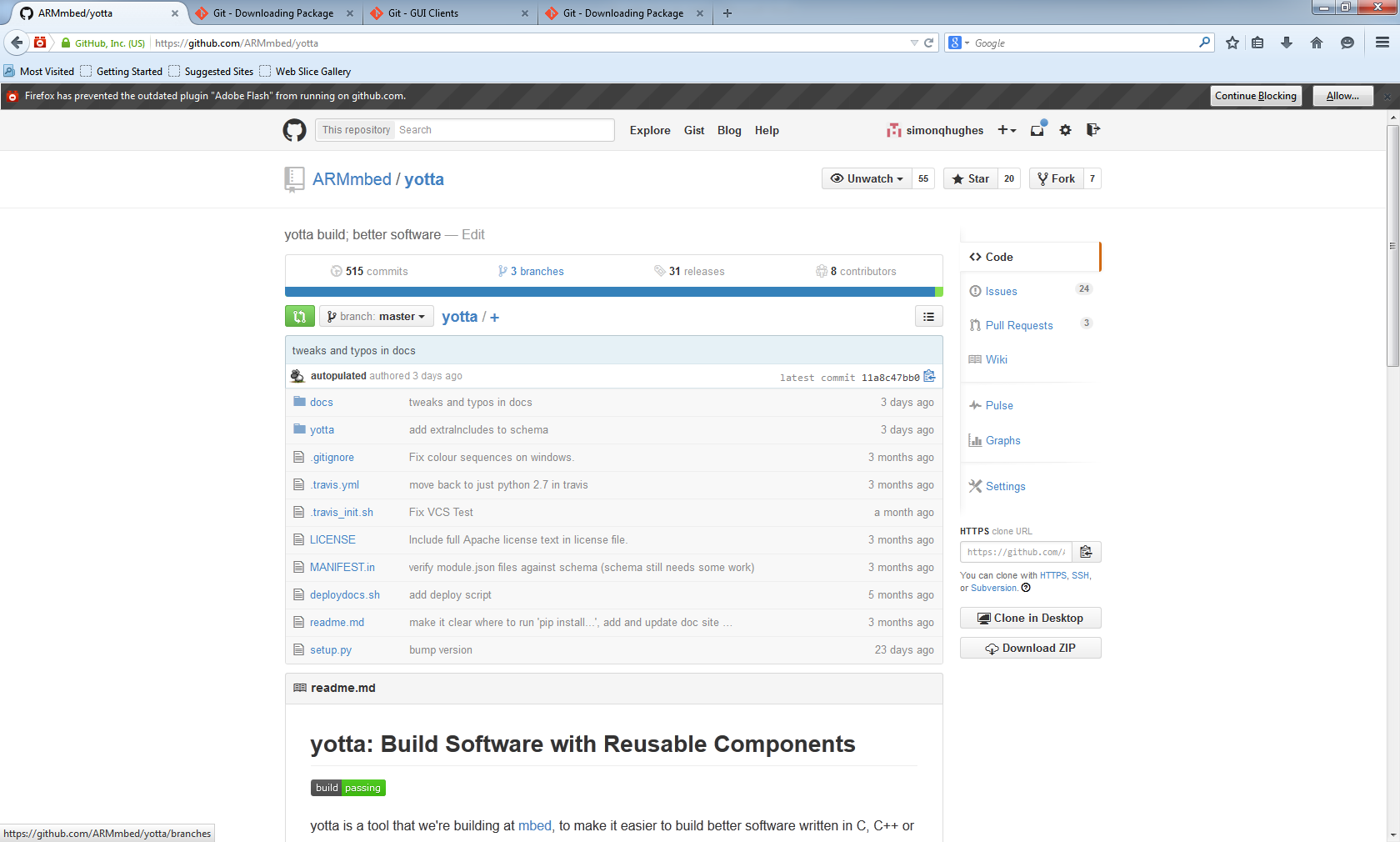
These are the tools you need to install:

1. Git Gui
2. Python2.7
3. Pycyrpto
4. Cmake
5. Ninja
6. Yotta

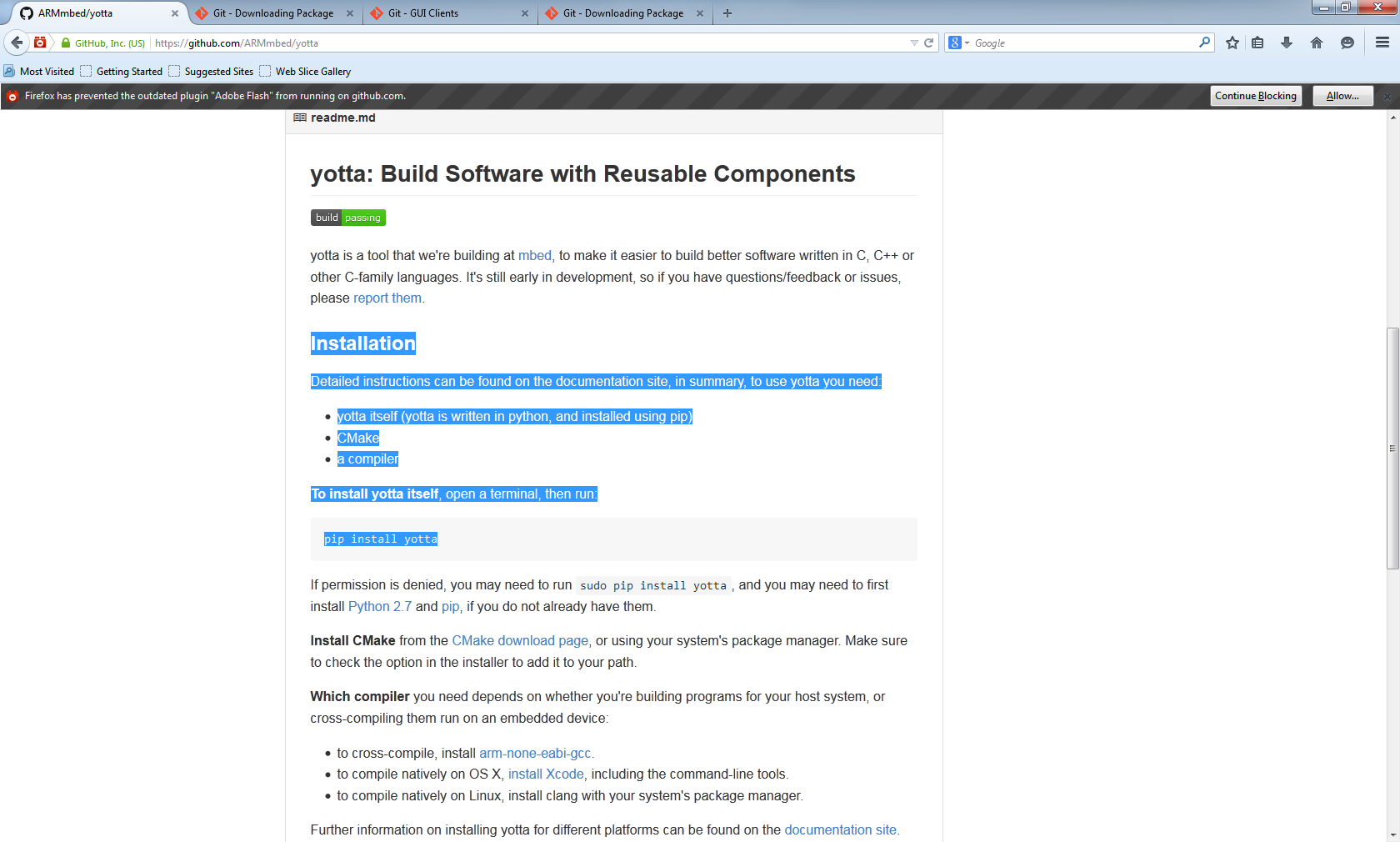
We will follow the notes in the following section taken from the ARMmbed yotta project README.md.

Sign into your github account and you’ll see the private repos when your permissions have been added:

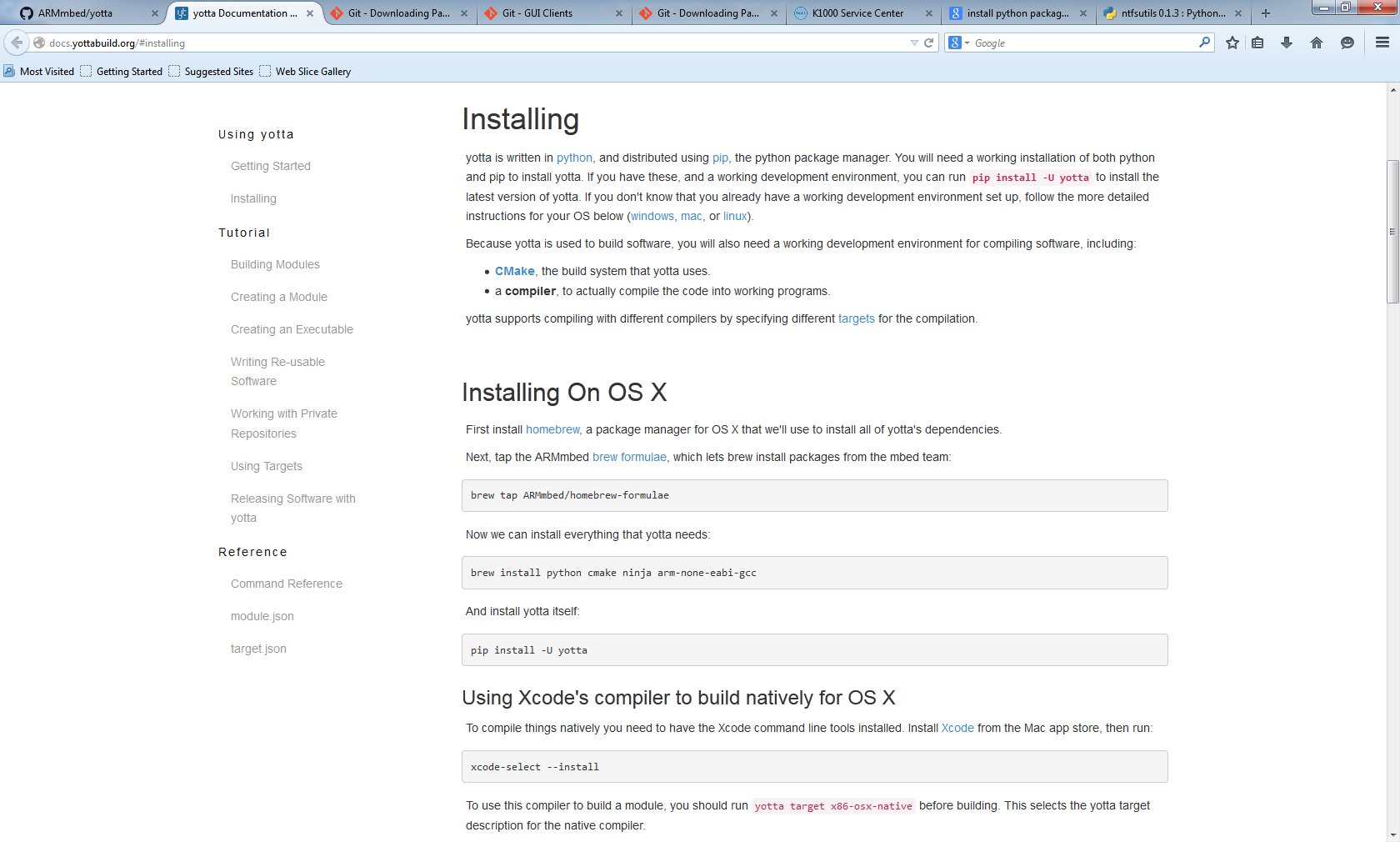




And this is the important bit:



The instructions on the documentation site link goes here:



The instructions from this page for installing for windows are reproduced in the following section. Review that sites notes before proceeding as things may have changed since the time of writing.

## Installing On Windows

1. **Install** [**python**](https://www.python.org/downloads/release/python-279/). You **must** install [python 2.7.9](https://www.python.org/downloads/release/python-279/) for yotta to work on windows. Select either the [x86-64 installer](https://www.python.org/ftp/python/2.7.9/python-2.7.9.amd64.msi) if you use 64-bit windows, or the [x86 installer](https://www.python.org/ftp/python/2.7.9/python-2.7.9.msi) if you use 32-bit windows.

**During installation, be sure to select the "add to path" option.** This will let you run python easily from a command prompt.

1. Install PyCrypto 2.6 for Python 2.7 from [Voidspace](http://www.voidspace.org.uk/python/modules.shtml#pycrypto). Select either the [32-bit installer](http://www.voidspace.org.uk/downloads/pycrypto26/pycrypto-2.6.win32-py2.7.exe) or the [64-bit installer](http://www.voidspace.org.uk/downloads/pycrypto26/pycrypto-2.6.win-amd64-py2.7.exe) matching the python version that you installed.

If pip cannot be found, you need to add the python scripts directory to your path. This is C:\Python27\Scripts unless you selected a different directory during python installation. [These instructions](http://docs.yottabuild.org/#windows-path) will guide you through the process.

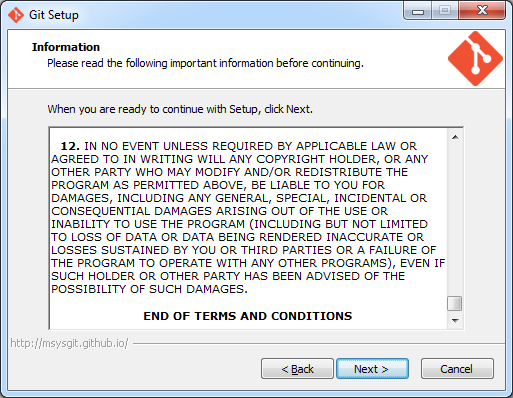
1. **Install** [**CMake**](http://www.cmake.org/download/). yotta uses CMake to generate makefiles that control the build. Select the latest available version, currently 3.1.0-rc2. The [32-bit version](http://www.cmake.org/files/v3.1/cmake-3.1.0-rc3-win32-x86.exe) will work on all versions of windows. Be sure to check the "add cmake to the path for current user" option during installation.
2. **Install Ninja**, the small and extremely fast build system that yotta uses. Download the release archive from the [releases page](https://github.com/martine/ninja/releases/download/v1.5.3/ninja-win.zip), and extract it to a directory (for example C:\ninja. Add this directory to [your path](http://docs.yottabuild.org/#windows-path).
3. Install the [**arm-none-eabi-gcc**](http://docs.yottabuild.org/#windows-cross-compile) **cross-compiler** in order to build software to run on embedded devices.
4. Finally, **open cmd.exe and run pip install -U yotta** to install yotta itself.

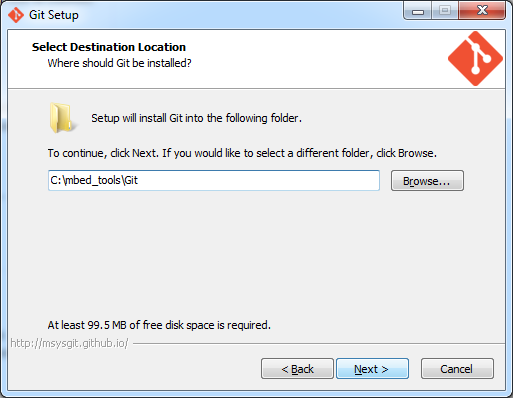
# Pre-requisites

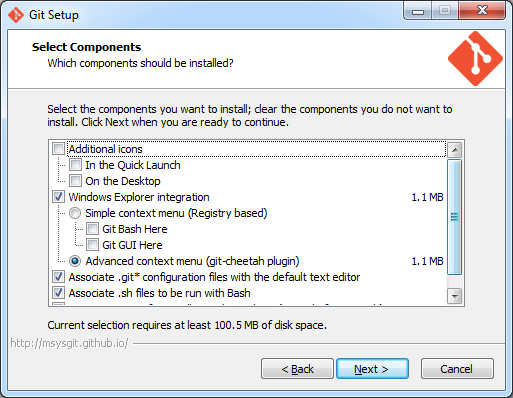
1. First, de-install any pre-existing versions of the applications and reboot, so you know you start from clean.

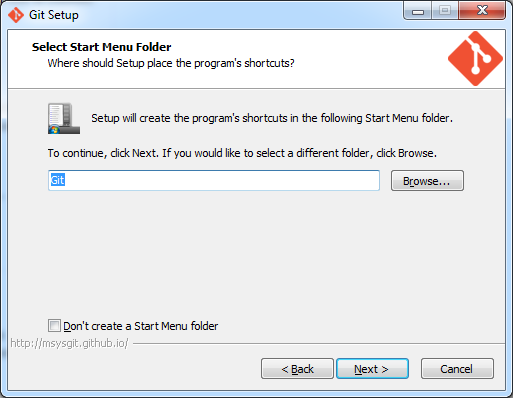
# Install Git for Windows (git gui and git bash)

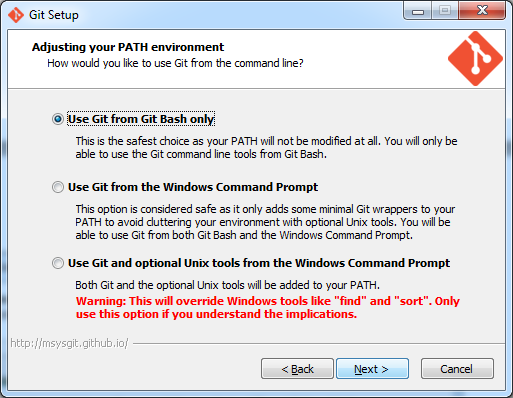


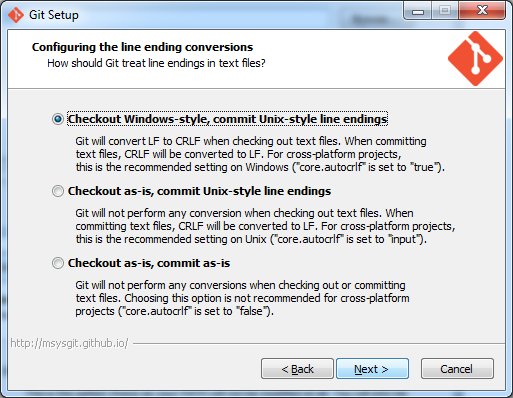


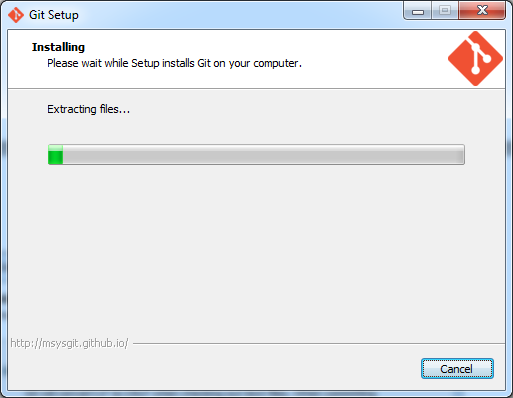




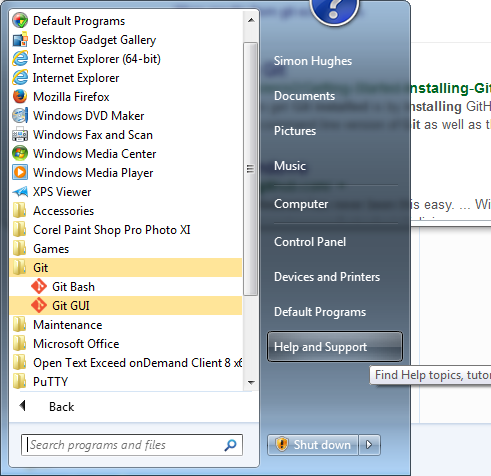








Finish dialog here.

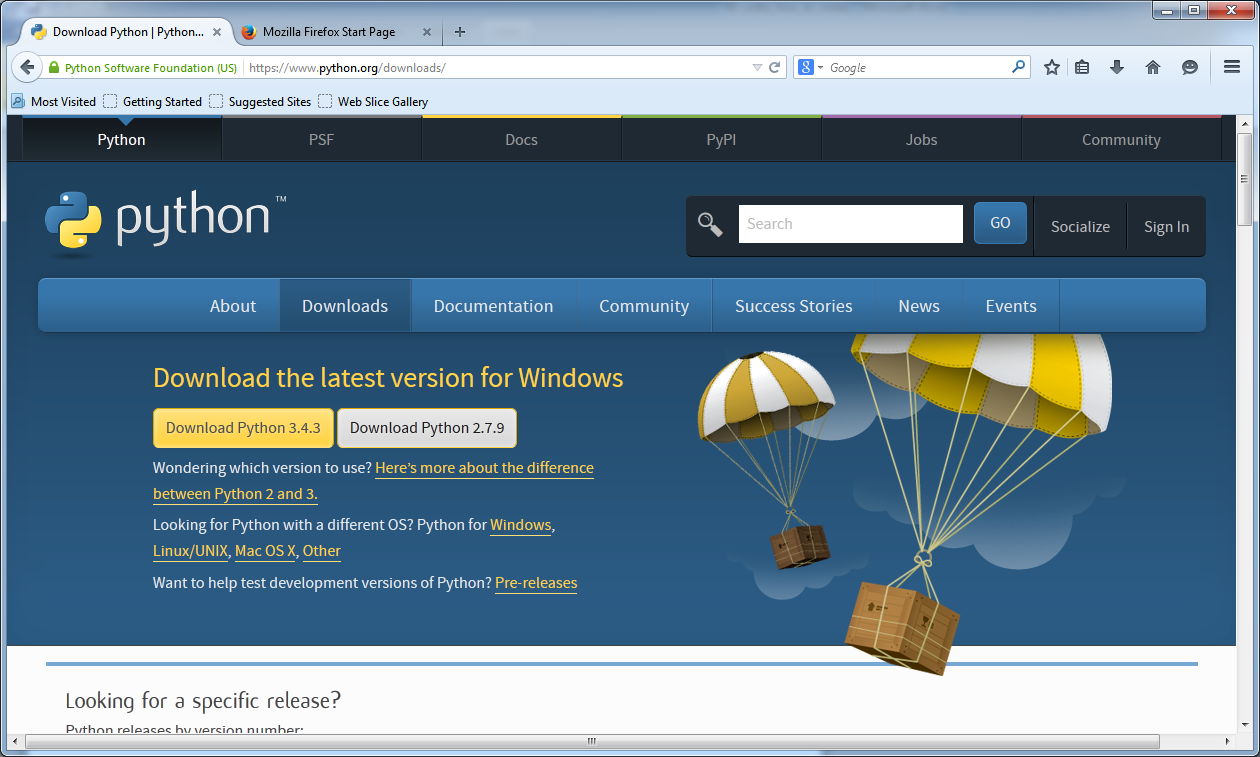


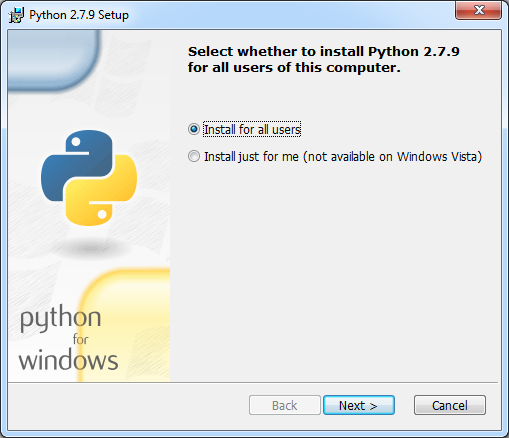
# Installing Python 2.7.9 for Windows

Install python 2.7.9 or later. Select 32 bit version. Python 2.7.9 includes pip (python install package) whereas earlier versions don’t so there will be additional messing around to install pip later.

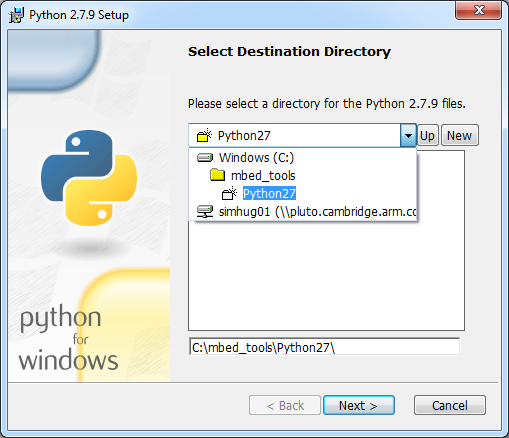
<https://www.python.org/downloads/>

Get python-2.7.9.msi.

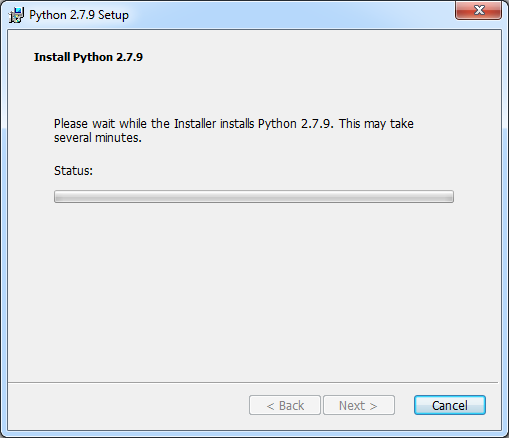




I’m installing all applications into c:\mbed\_tools

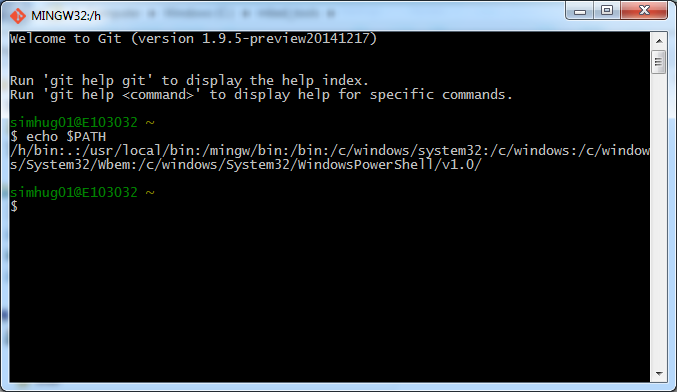




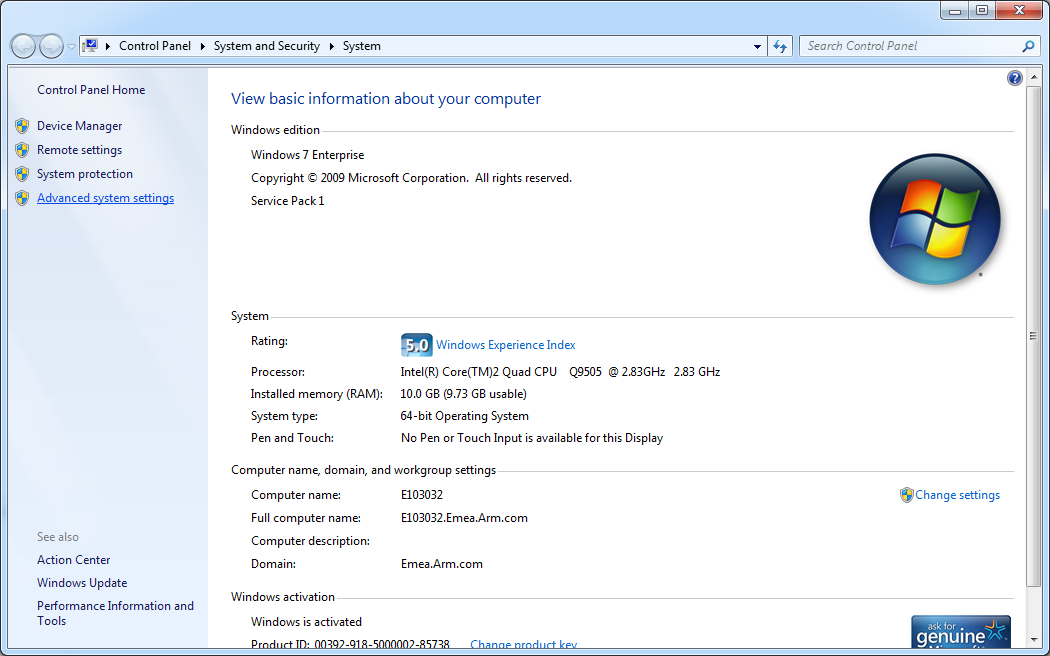


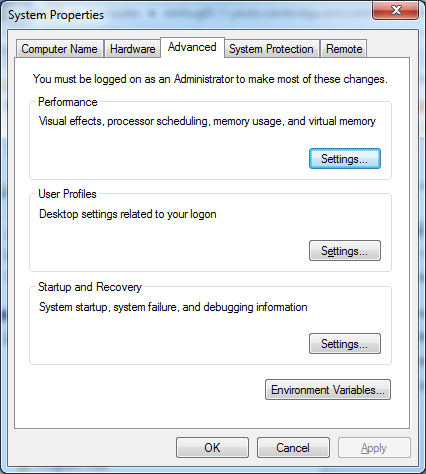


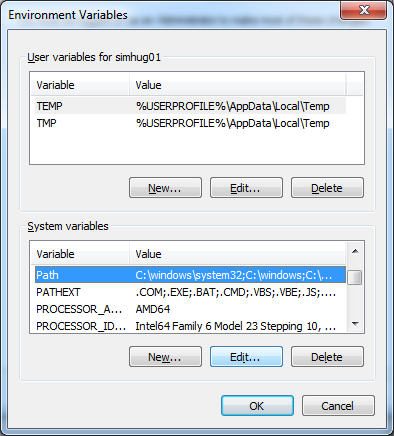
Start git bash:

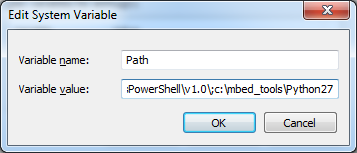


Note the path to python is not on the path yet so we need to do that:



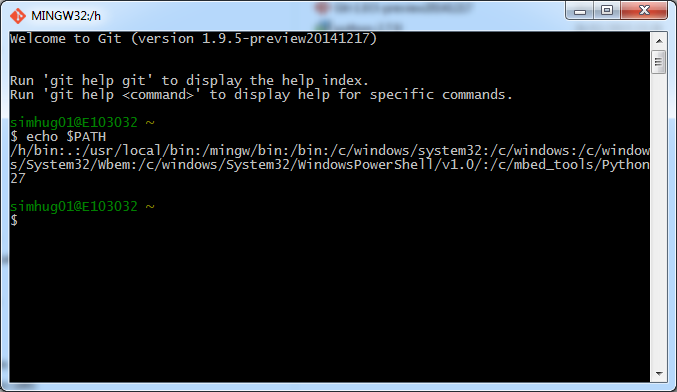




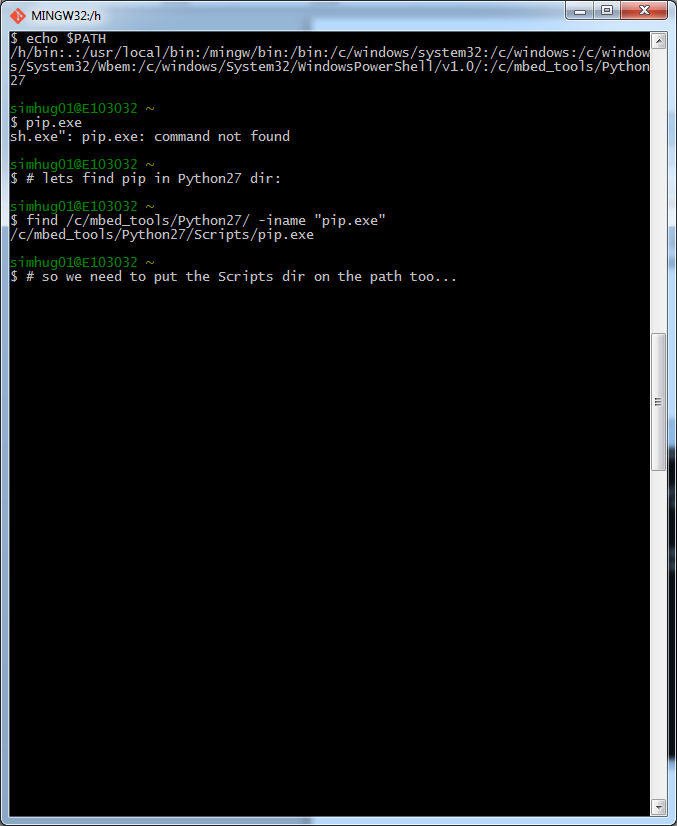


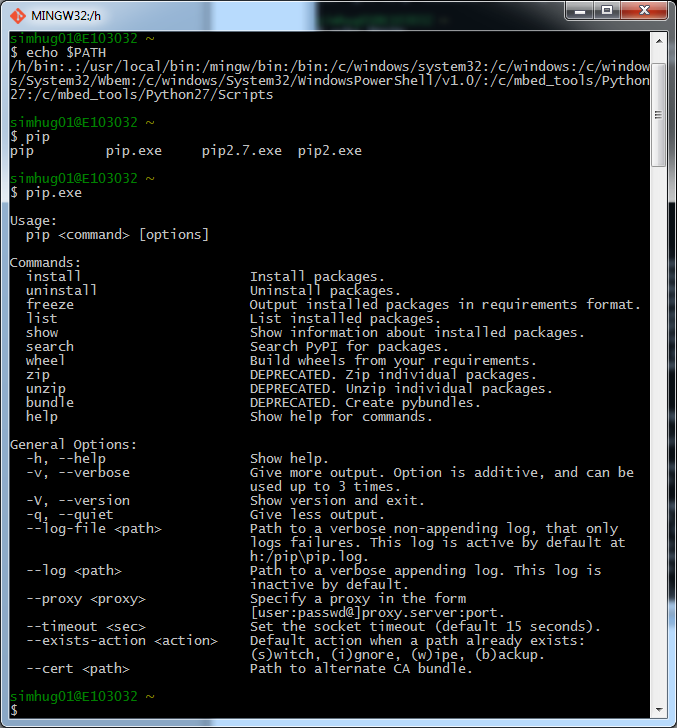
Then restart git bash and echo $PATH:

Now python is on the path:



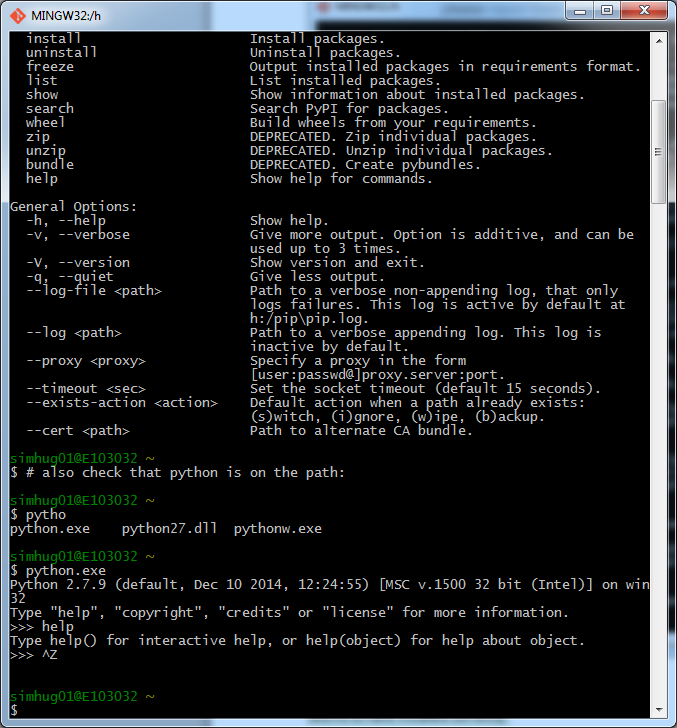
Adding the Python27\Scripts to the PATH to get access to pip.exe



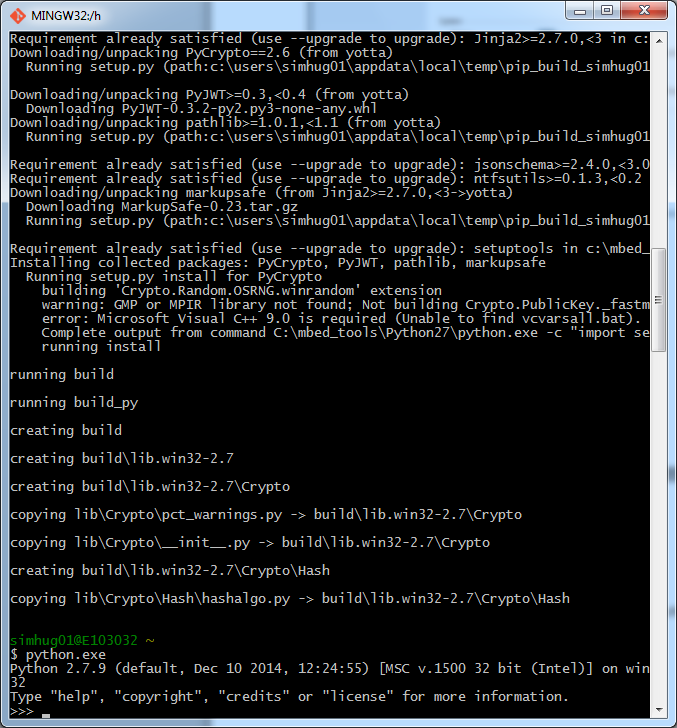


So now pip is working.

Check python interactive environment works and inspect version:



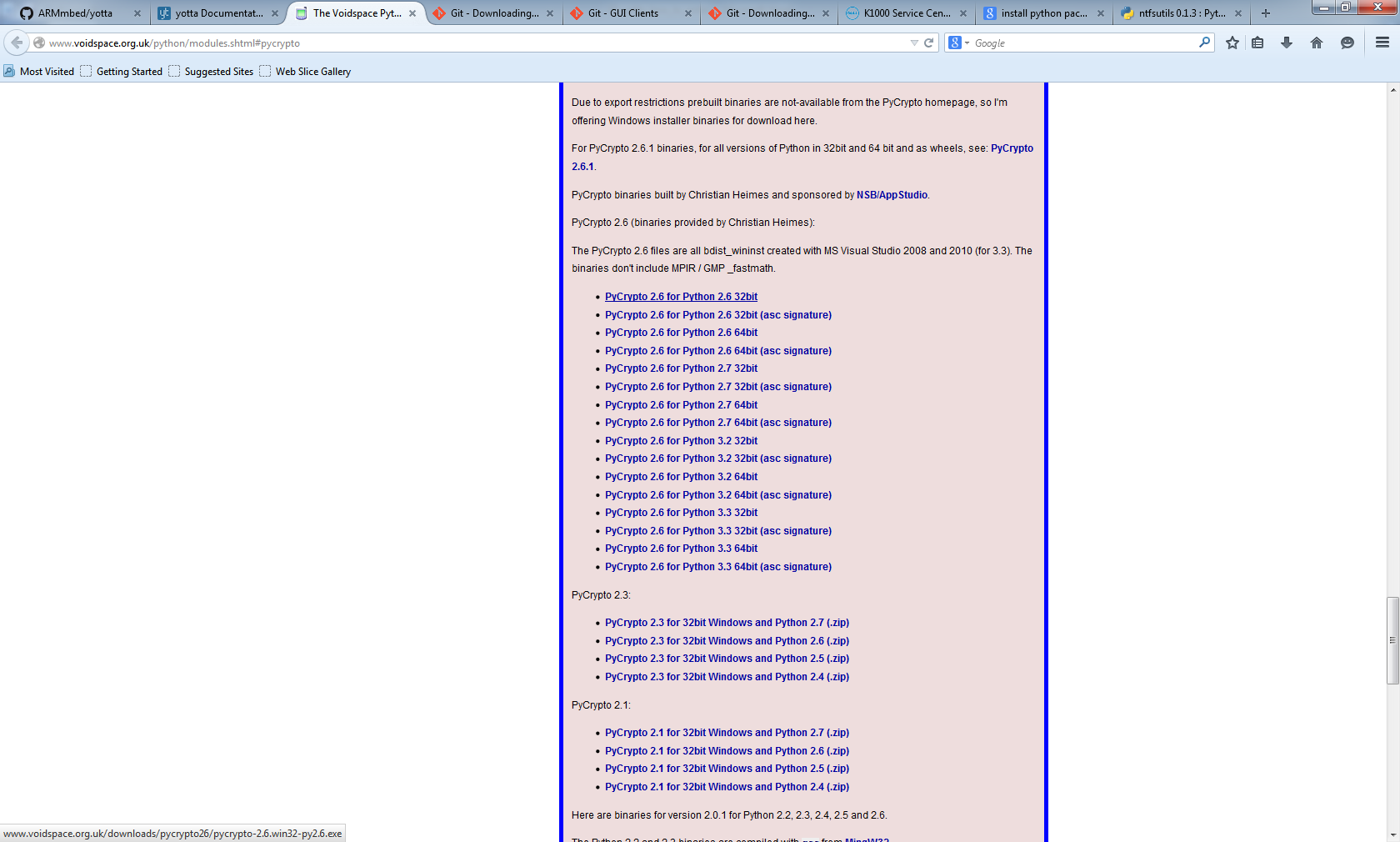
# Installing pyCryptio

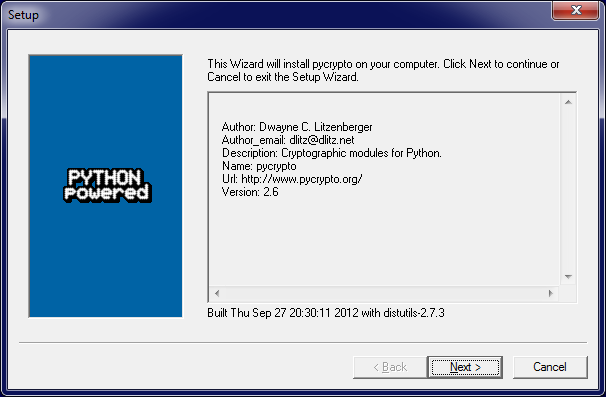


So I’m running python32 bit (see above) so I’m going to install the 32 bit version of pyCrypto:

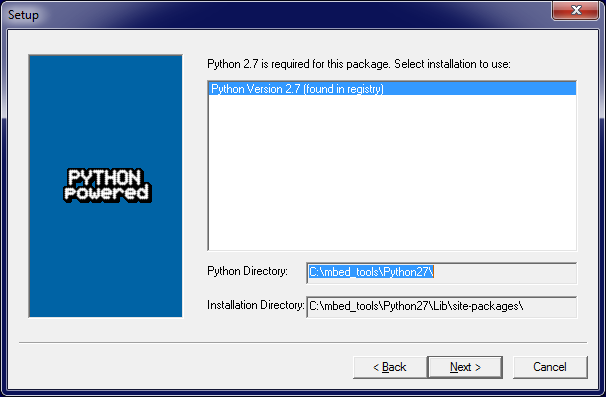
pycrypto-2.6.win32-py2.7.exe

which can be seen in the list below on the voidspace site.

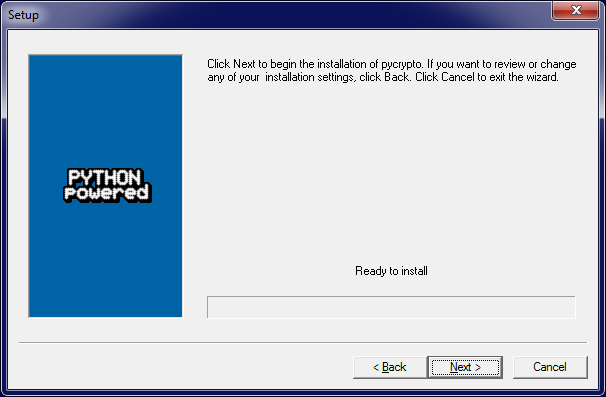




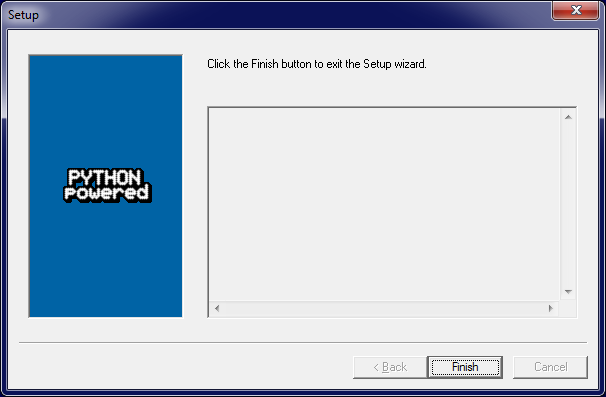
Press next. It has automatically detected the corrected dir for 32 python as shown in the next pic:



Press next:



Press next again:



press finish ok everything done.

Now check if pycrypto is installed in python27 32 bit

simhug01@E103032 /cygdrive/c/mbed\_tools

$ find Python27 -iname "pycrypto\*"

Python27/Lib/site-packages/pycrypto-2.6-py2.7.egg-info

Python27/pycrypto-wininst.log

simhug01@E103032 /cygdrive/c/mbed\_tools

Yes, its installed!

Todo here: insert installation process for install yotta in python27\_64

# Install CMake

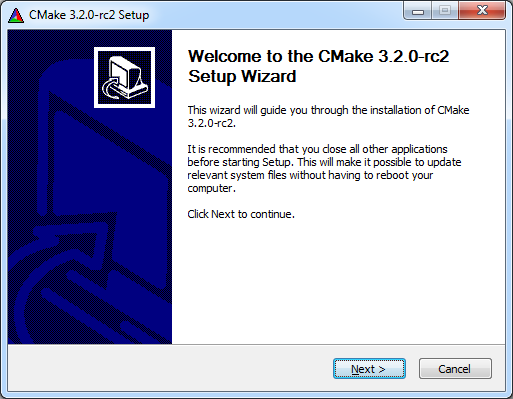
Here are the instruction from docs.yottabuild.org about cmake installation:

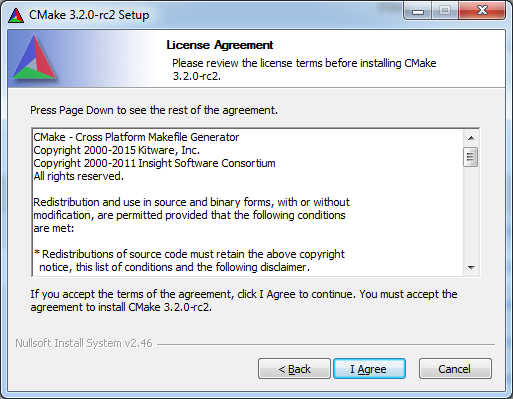
**Install** [**CMake**](http://www.cmake.org/download/). yotta uses CMake to generate makefiles that control the build. Select the latest available version, currently 3.1.0-rc2. The [32-bit version](http://www.cmake.org/files/v3.1/cmake-3.1.0-rc3-win32-x86.exe) will work on all versions of windows. Be sure to check the "add cmake to the path for current user" option during installation.

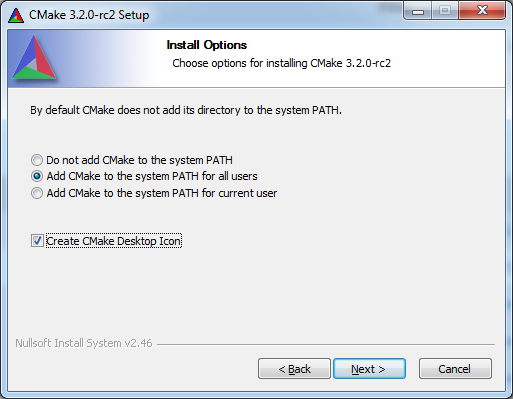
<http://www.cmake.org/download/>

using the latest version which is this:

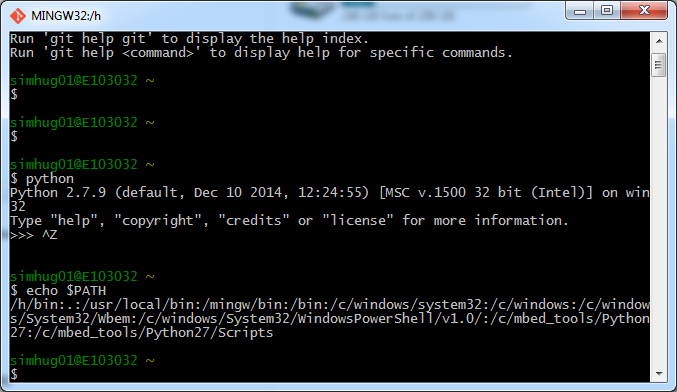
cmake-3.2.0-rc2-win32-x86.exe



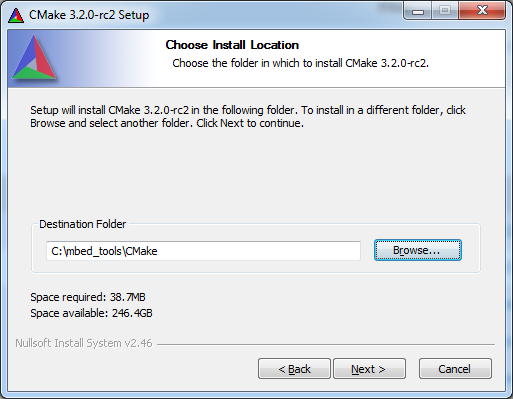


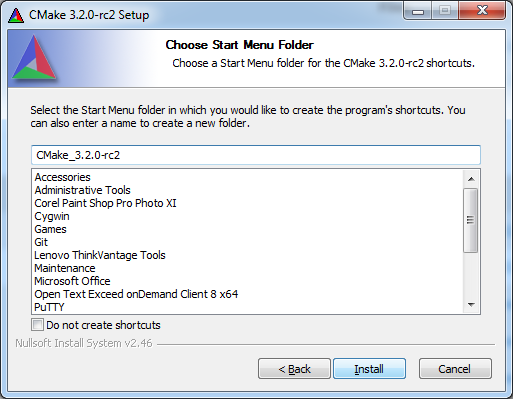


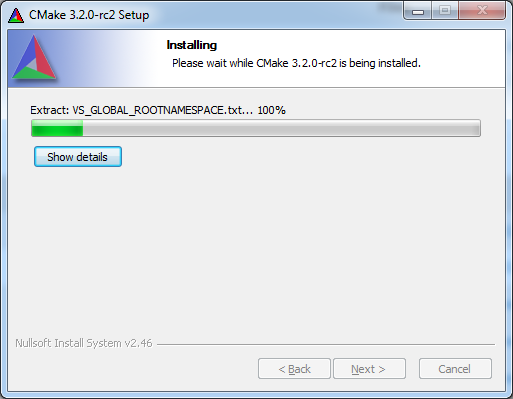
This is the path before CMake changes it:

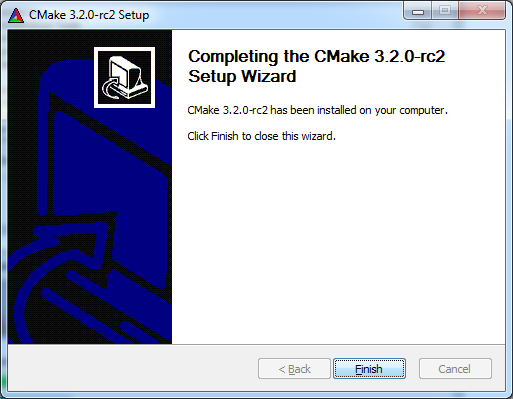


Notice I’ve changed the path to the install dir in the next dialog to be installed in my tools folder:

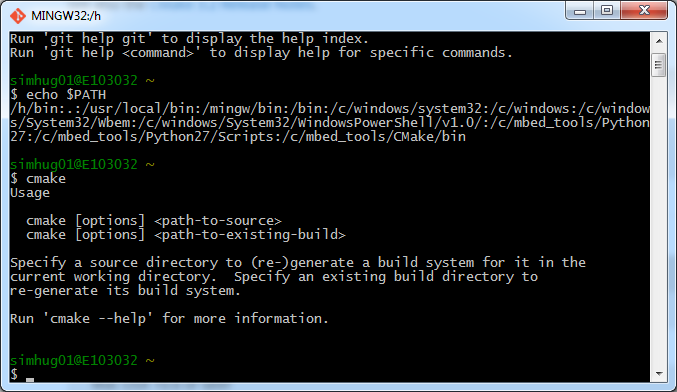








Now see how the path has been modified to include cmake/bin dir:



If this didn’t happy correctly then modify as appropriate

# Install Ninja

**Install Ninja**, the small and extremely fast build system that yotta uses. Download the release archive from the [releases page](https://github.com/martine/ninja/releases/download/v1.5.3/ninja-win.zip), and extract it to a directory (for example C:\ninja. Add this directory to [your path](http://docs.yottabuild.org/#windows-path).

<https://github.com/martine/ninja/>

Ninja is a small build system with a focus on speed.

<http://martine.github.com/ninja/>

See the manual -- <http://martine.github.com/ninja/manual.html> or

doc/manual.asciidoc included in the distribution -- for background

and more details.

To build, run ./configure.py --bootstrap. It first compiles all non-test

source files together, then re-builds Ninja using itself. You should

end up with a 'ninja' binary in the source root.

Run './configure.py --help' for more configuration options.

Run './ninja -h' for Ninja help.

Installation is not necessary because the only required file is is the

resulting ninja binary. However, to enable features like Bash

completion and Emacs and Vim editing modes, some files in misc/ must be

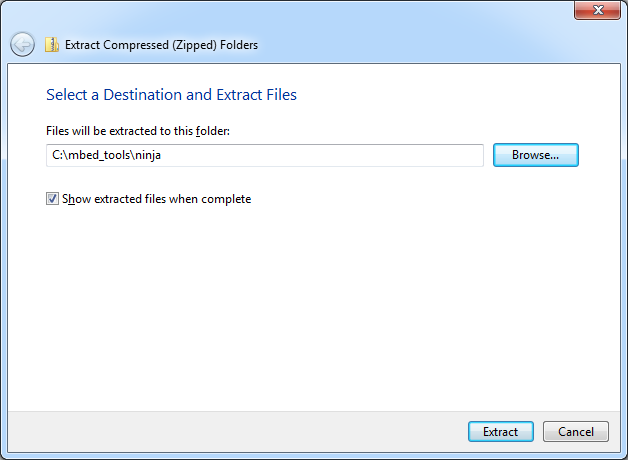
copied to appropriate locations.

If you're interested in making changes to Ninja, read HACKING.md first.

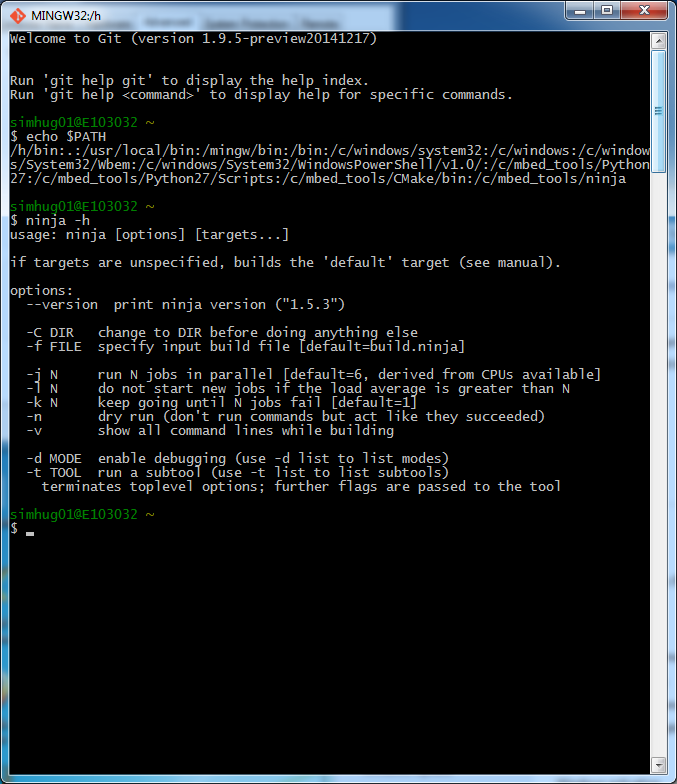
Releases are available here:

https://github.com/martine/ninja/releases/

ninja-win.zip

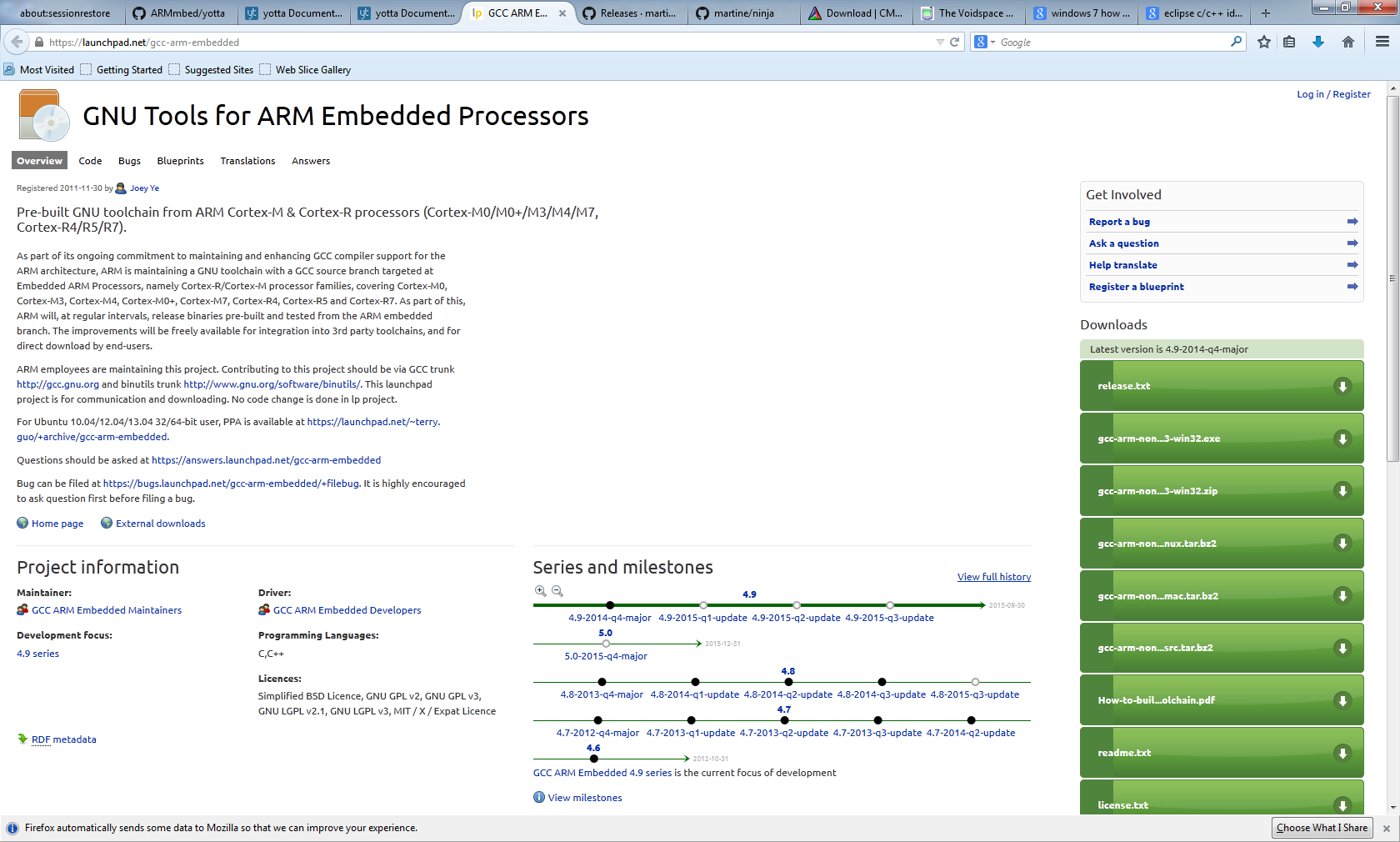


Modify path to include this

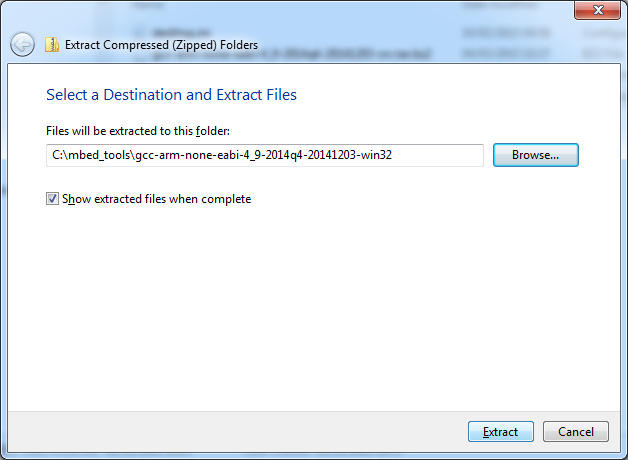


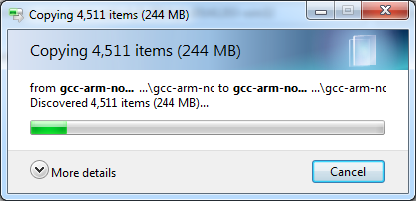
# Install arm-none-eabi-gcc

Install the [**arm-none-eabi-gcc**](http://docs.yottabuild.org/#windows-cross-compile) **cross-compiler** in order to build software to run on embedded devices.

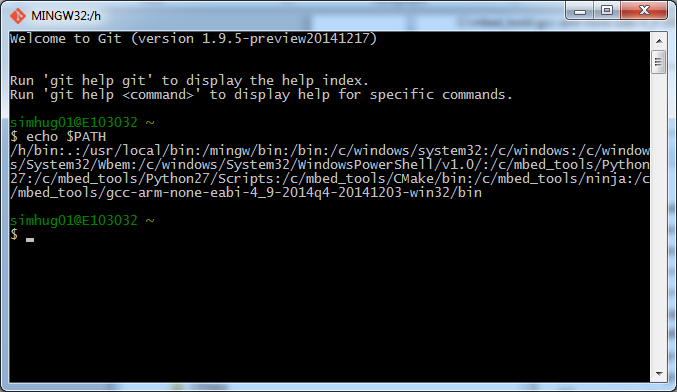


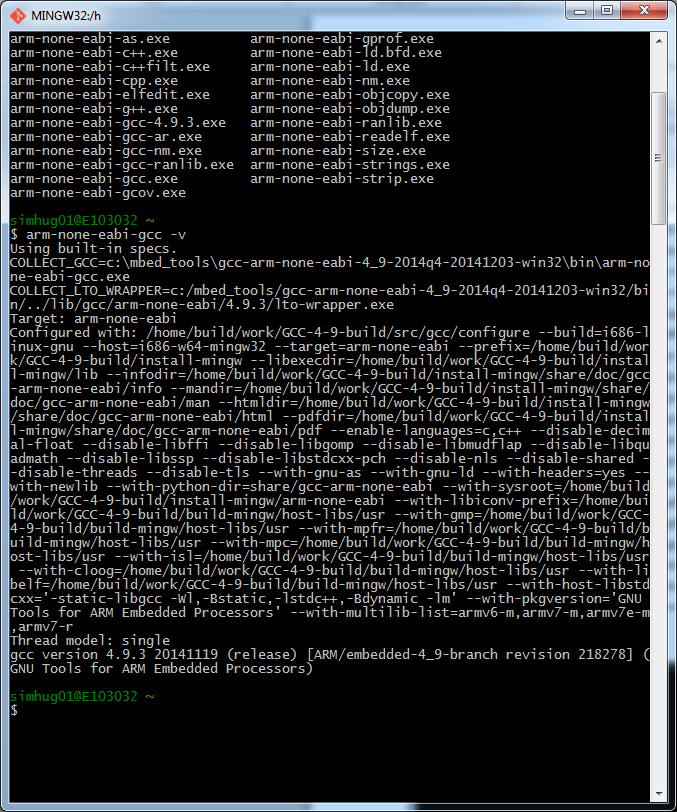
gcc-arm-none-eabi-4\_9-2014q4-20141203-win32.zip





Now, need to put gcc\_arm bin dir on the path





# Install Yotta

Run the following command to install yotta:

pip install yotta

Note also that yotta has been installed for 32 bit version only:

simhug01@E103032 /cygdrive/c/mbed\_tools

$ find Python27 -iname "yotta\*"

Python27/Lib/site-packages/yotta

Python27/Lib/site-packages/yotta-0.0.43-py2.7.egg-info

Python27/Scripts/yotta-script.py

Python27/Scripts/yotta.exe

Python27/Scripts/yotta.exe.manifest

simhug01@E103032 /cygdrive/c/mbed\_tool

# Yotta commands

Create a build of mbed-something

Set the target you’re building for by using the target command:

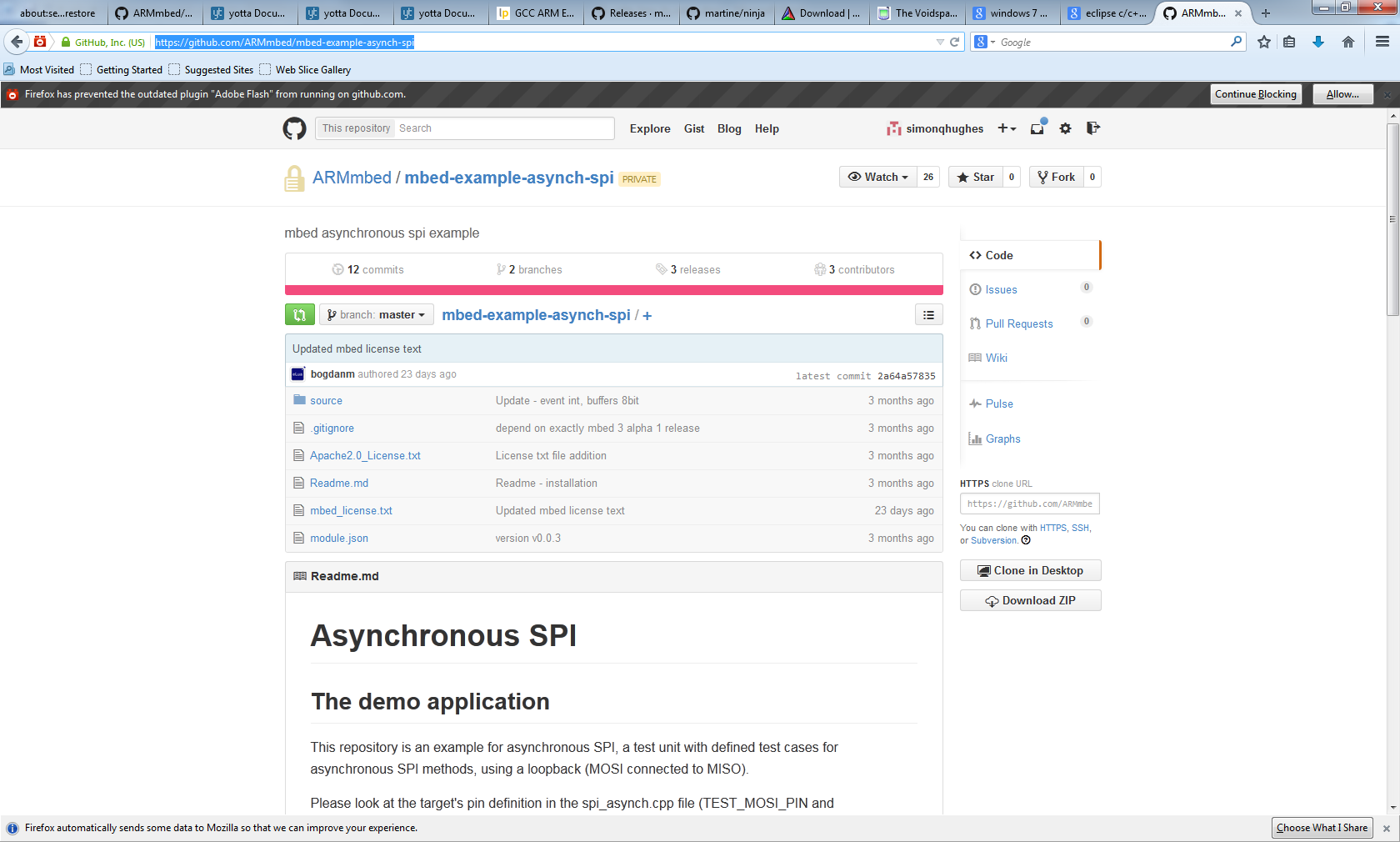
yotta target frdm-k64f-gcc

note the command is specifying a {target, compiler} tuple.

yotta target

i.e. without args list the current

Take a clone of the repository of interest e.g. the one below:



simhug01@E103032 /c/develop/public/jobs/yr2015/2209/work/20150304/proj

git clone <https://github.com/ARMmbed/mbed-example-asynch-spi>

Having setup ssh keys with github, this is how to get a clone of the repository:

simhug01@E103032 /cygdrive/c/develop/public/jobs/yr2015/2209/work/20150304/proj

$ git clone git@github.com:ARMmbed/mbed-example-asynch-spi.git

Cloning into 'mbed-example-asynch-spi'...

simhug01@E103032 /cygdrive/c/develop/public/jobs/yr2015/2209/work/20150304/proj

$ ls -al

total 8

d---rwx---+ 1 simhug01 Domain Users 0 Mar 4 17:47 .

d---rwx---+ 1 simhug01 Domain Users 0 Mar 4 13:43 ..

d---rwx---+ 1 simhug01 Domain Users 0 Mar 4 17:47 mbed-example-asynch-spi

d---rwx---+ 1 simhug01 Domain Users 0 Mar 4 17:25 mbed-ls

simhug01@E103032 /cygdrive/c/develop/public/jobs/yr2015/2209/work/20150304/proj

$ ls -la mbed-example-asynch-spi/

total 54

d---rwx---+ 1 simhug01 Domain Users 0 Mar 4 17:47 .

d---rwx---+ 1 simhug01 Domain Users 0 Mar 4 17:47 ..

d---rwx---+ 1 simhug01 Domain Users 0 Mar 4 17:47 .git

----rwx---+ 1 simhug01 Domain Users 44 Mar 4 17:47 .gitignore

----rwx---+ 1 simhug01 Domain Users 11156 Mar 4 17:47 Apache2.0\_License.txt

----rwx---+ 1 simhug01 Domain Users 25897 Mar 4 17:47 mbed\_license.txt

----rwx---+ 1 simhug01 Domain Users 526 Mar 4 17:47 module.json

----rwx---+ 1 simhug01 Domain Users 3396 Mar 4 17:47 Readme.md

d---rwx---+ 1 simhug01 Domain Users 0 Mar 4 17:47 source

simhug01@E103032 /cygdrive/c/develop/public/jobs/yr2015/2209/work/20150304/proj

from yotta\_example\_script\_20150304\_2.txt

==================================================================================

Script started on Wed, Mar 04, 2015 6:25:30 PM

\_]0;/cygdrive/c/develop/public/jobs/yr2015/2209/work/20150304/proj/mbed-example-asynch-spi\_

\_[32msimhug01@E103032 \_[33m/cygdrive/c/develop/public/jobs/yr2015/2209/work/20150304/proj/mbed-example-asynch-spi\_[0m

$ yotta \_\_[Kbuild

\_[1m\_[33mwarning\_[39m: \_[22m\_[33muvisor-lib has invalid module.json:\_[0m

\_[1m\_[33mwarning\_[39m: \_[22m\_[33m author value [u'Milosch Meriac <milosch.meriac@arm.com>', u'Alessandro Angelino <alessandro.angelino@arm.com>'] is not valid under any of the given schemas\_[0m

\_[22m\_[32minfo\_[39m: \_[22m\_[39mgenerate for target: frdm-k64f-gcc 0.0.10 at C:\develop\public\jobs\yr2015\2209\work\20150304\proj\mbed-example-asynch-spi\yotta\_targets\frdm-k64f-gcc\_[0m

mbedOS.cmake included

GCC-C.cmake included

mbedOS-GNU-C.cmake included

GCC-GXX.cmake included

mbedOS-GNU-CXX.cmake included

GCC version is: 4.9.3

-- The ASM compiler identification is GNU

-- Found assembler: C:/mbed\_tools/gcc-arm-none-eabi-4\_9-2014q4-20141203-win32/bin/arm-none-eabi-gcc.exe

GNU-ASM.cmake included

GNU-ASM.cmake included

-- Configuring done

-- Generating done

-- Build files have been written to: C:/develop/public/jobs/yr2015/2209/work/20150304/proj/mbed-example-asynch-spi/build/frdm-k64f-gcc

[1/173] Building C object ym/cmsis-core/yotta\_dummy\_lib\_cmsis\_core/CMakeFiles/cmsis-core.dir/dummy.c.obj

[2/173] Building ASM object ym/uvisor-lib/source/CMakeFiles/uvisor-lib.dir/MK64FN1M0XXX12/uvisor-GCC\_ARM.s.obj

[3/173] Linking ASM static library ym/uvisor-lib/source/uvisor-lib.a

[4/173] Linking C static library ym/cmsis-core/yotta\_dummy\_lib\_cmsis\_core/cmsis-core.a

[5/173] Building C object ym/cmsis-core-freescale/yotta\_dummy\_lib\_cmsis\_core\_freescale/CMakeFiles/cmsis-core-freescale.dir/dummy.c.obj

[6/173] Linking C static library ym/cmsis-core-freescale/yotta\_dummy\_lib\_cmsis\_core\_freescale/cmsis-core-freescale.a

[7/173] Building C object ym/cmsis-core-k64f/source/CMakeFiles/cmsis-core-k64f.dir/0aa689a138613051884deb73dfbf830e/yotta\_modules/cmsis-core-k64f/source/cmsis\_nvic.c.obj

[8/173] Building C object ym/cmsis-core-k64f/source/CMakeFiles/cmsis-core-k64f.dir/0aa689a138613051884deb73dfbf830e/yotta\_modules/cmsis-core-k64f/source/system\_MK64F12.c.obj

[9/173] Linking C static library ym/cmsis-core-k64f/source/cmsis-core-k64f.a

[10/173] Building C object ym/mbed/source/CMakeFiles/mbed.dir/88f16d0e8fad361b7a6f2d32f99fd9a0/20150304/proj/mbed-example-asynch-spi/yotta\_modules/mbed/source/us\_ticker\_api.c.obj

[11/173] Building CXX object ym/mbed/source/CMakeFiles/mbed.dir/28a794d84957a21a8553d0da91138238/proj/mbed-example-asynch-spi/yotta\_modules/mbed/source/FunctionPointer.cpp.obj

[12/173] Building C object ym/mbed/source/CMakeFiles/mbed.dir/88f16d0e8fad361b7a6f2d32f99fd9a0/20150304/proj/mbed-example-asynch-spi/yotta\_modules/mbed/source/semihost\_api.c.obj

Etc.

==================================================================================

At the end, yotta should have built everything and there will be some binaries in the tree.

# Installing mbed-ls

$ git clone git@github.com:ARMmbed/mbed-ls.git

Cloning into 'mbed-ls'...

remote: Counting objects: 144, done.

remote: Compressing objects: 100% (12/12), done.

Receiving objects: 45% (emote: Total 144 (delta 6), reused 0 (delta 0), pack-r

Receiving objects: 47% (68/144)

Receiving objects: 100% (144/144), 36.36 KiB | 0 bytes/s, done.

Resolving deltas: 100% (89/89), done.

Checking connectivity... done.

simhug01@E103032 /c/develop/public/jobs/yr2015/2209/work/20150304/proj

$

simhug01@E103032 /c/develop/public/jobs/yr2015/2209/work/20150304/proj/mbed-ls (

master)

$ python setup.py install

running install

running bdist\_egg

running egg\_info

creating mbed\_ls.egg-info

writing requirements to mbed\_ls.egg-info\requires.txt

writing mbed\_ls.egg-info\PKG-INFO

writing top-level names to mbed\_ls.egg-info\top\_level.txt

writing dependency\_links to mbed\_ls.egg-info\dependency\_links.txt

writing entry points to mbed\_ls.egg-info\entry\_points.txt

writing manifest file 'mbed\_ls.egg-info\SOURCES.txt'

reading manifest file 'mbed\_ls.egg-info\SOURCES.txt'

writing manifest file 'mbed\_ls.egg-info\SOURCES.txt'

installing library code to build\bdist.win32\egg

running install\_lib

running build\_py

creating build

creating build\lib

creating build\lib\mbed\_lstools

copying mbed\_lstools\lstools\_base.py -> build\lib\mbed\_lstools

copying mbed\_lstools\lstools\_darwin.py -> build\lib\mbed\_lstools

copying mbed\_lstools\lstools\_ubuntu.py -> build\lib\mbed\_lstools

copying mbed\_lstools\lstools\_win7.py -> build\lib\mbed\_lstools

copying mbed\_lstools\main.py -> build\lib\mbed\_lstools

copying mbed\_lstools\\_\_init\_\_.py -> build\lib\mbed\_lstools

creating build\bdist.win32

creating build\bdist.win32\egg

creating build\bdist.win32\egg\mbed\_lstools

copying build\lib\mbed\_lstools\lstools\_base.py -> build\bdist.win32\egg\mbed\_lst

ools

copying build\lib\mbed\_lstools\lstools\_darwin.py -> build\bdist.win32\egg\mbed\_l

stools

copying build\lib\mbed\_lstools\lstools\_ubuntu.py -> build\bdist.win32\egg\mbed\_l

stools

copying build\lib\mbed\_lstools\lstools\_win7.py -> build\bdist.win32\egg\mbed\_lst

ools

copying build\lib\mbed\_lstools\main.py -> build\bdist.win32\egg\mbed\_lstools

copying build\lib\mbed\_lstools\\_\_init\_\_.py -> build\bdist.win32\egg\mbed\_lstools

byte-compiling build\bdist.win32\egg\mbed\_lstools\lstools\_base.py to lstools\_bas

e.pyc

byte-compiling build\bdist.win32\egg\mbed\_lstools\lstools\_darwin.py to lstools\_d

arwin.pyc

byte-compiling build\bdist.win32\egg\mbed\_lstools\lstools\_ubuntu.py to lstools\_u

buntu.pyc

byte-compiling build\bdist.win32\egg\mbed\_lstools\lstools\_win7.py to lstools\_win

7.pyc

byte-compiling build\bdist.win32\egg\mbed\_lstools\main.py to main.pyc

byte-compiling build\bdist.win32\egg\mbed\_lstools\\_\_init\_\_.py to \_\_init\_\_.pyc

creating build\bdist.win32\egg\EGG-INFO

copying mbed\_ls.egg-info\PKG-INFO -> build\bdist.win32\egg\EGG-INFO

copying mbed\_ls.egg-info\SOURCES.txt -> build\bdist.win32\egg\EGG-INFO

copying mbed\_ls.egg-info\dependency\_links.txt -> build\bdist.win32\egg\EGG-INFO

copying mbed\_ls.egg-info\entry\_points.txt -> build\bdist.win32\egg\EGG-INFO

copying mbed\_ls.egg-info\requires.txt -> build\bdist.win32\egg\EGG-INFO

copying mbed\_ls.egg-info\top\_level.txt -> build\bdist.win32\egg\EGG-INFO

zip\_safe flag not set; analyzing archive contents...

creating dist

creating 'dist\mbed\_ls-0.1.5-py2.7.egg' and adding 'build\bdist.win32\egg' to it

removing 'build\bdist.win32\egg' (and everything under it)

Processing mbed\_ls-0.1.5-py2.7.egg

Copying mbed\_ls-0.1.5-py2.7.egg to c:\mbed\_tools\python27\lib\site-packages

Adding mbed-ls 0.1.5 to easy-install.pth file

Installing mbedls-script.py script to c:\mbed\_tools\Python27\Scripts

Installing mbedls.exe script to c:\mbed\_tools\Python27\Scripts

Installing mbedls.exe.manifest script to c:\mbed\_tools\Python27\Scripts

Installed c:\mbed\_tools\python27\lib\site-packages\mbed\_ls-0.1.5-py2.7.egg

Processing dependencies for mbed-ls==0.1.5

Searching for PrettyTable>=0.7.2

Reading https://pypi.python.org/simple/PrettyTable/

Best match: prettytable 0.7.2

Downloading https://pypi.python.org/packages/source/P/PrettyTable/prettytable-0.

7.2.zip#md5=0c1361104caff8b09f220748f9d69899

Processing prettytable-0.7.2.zip

Writing c:\users\simhug01\appdata\local\temp\easy\_install-hyy2gd\prettytable-0.7

.2\setup.cfg

Running prettytable-0.7.2\setup.py -q bdist\_egg --dist-dir c:\users\simhug01\app

data\local\temp\easy\_install-hyy2gd\prettytable-0.7.2\egg-dist-tmp-dhgnm1

zip\_safe flag not set; analyzing archive contents...

Adding prettytable 0.7.2 to easy-install.pth file

Installed c:\mbed\_tools\python27\lib\site-packages\prettytable-0.7.2-py2.7.egg

Finished processing dependencies for mbed-ls==0.1.5

simhug01@E103032 /c/develop/public/jobs/yr2015/2209/work/20150304/proj/mbed-ls (

master)

$

This has now been installed in the Python 32bit subdir:

simhug01@E103032 /cygdrive/c/mbed\_tools

$ find "Python27" -iname "mbedls\*"

Python27/Scripts/mbedls-script.py

Python27/Scripts/mbedls.exe

Python27/Scripts/mbedls.exe.manifest

simhug01@E103032 /cygdrive/c/mbed\_tools

10-yotta\_how\_to\_notes.docx

20150506

11-yotta\_how\_to\_notes.docx

20150616

Sometimes I run in to problems with yotta e.g. on windows on the limitation of the 260 chars in a file path causes problems. Here’s one I encountered on yotta v0.4.3:

e.g. the following:

\_[1m\_[31merror\_[39m: \_[22m\_[31mInstalled module mbed-hal doesn't match specification >=0.1.0,<0.2.0 (when trying to find mbed-hal at ~0.1.0 for mbed-hal-frdm-k64f)\_[0m

\_[1m\_[31merror\_[39m: \_[22m\_[31mInstalled module cmsis-core doesn't match specification >=0.1.0,<0.2.0 (when trying to find cmsis-core at ~0.1.0 for cmsis-core-k64f)\_[0m

\_[1m\_[31merror\_[39m: \_[22m\_[31mInstalled module mbed-hal doesn't match specification >=0.1.0,<0.2.0 (when trying to find dependencies for mbed-hal-frdm-k64f)\_[0m

\_[1m\_[31merror\_[39m: \_[22m\_[31mInstalled module cmsis-core doesn't match specification >=0.1.0,<0.2.0 (when trying to find dependencies for cmsis-core-k64f)\_[0m

\_[22m\_[32minfo\_[39m: \_[22m\_[39mgenerate for target: frdm-k64f-gcc 0.0.17 at d:\datastore\public\jobs\yr2015\2212\work\20150616\lwm2m-client\yotta\_targets\frdm-k64f-gcc\_[0m

\_[1m\_[31merror\_[39m: \_[22m\_[31mInstalled module mbed-hal doesn't match specification >=0.1.0,<0.2.0 (when trying to find dependencies for mbed-hal-frdm-k64f)\_[0m

\_[1m\_[31merror\_[39m: \_[22m\_[31mInstalled module cmsis-core doesn't match specification >=0.1.0,<0.2.0 (when trying to find dependencies for cmsis-core-k64f)\_[0m

\_[1m\_[31merror\_[39m: \_[22m\_[31mInstalled module mbed-hal doesn't match specification >=0.1.0,<0.2.0 (when trying to find dependencies for mbed-hal-frdm-k64f)\_[0m

\_[1m\_[31merror\_[39m: \_[22m\_[31mInstalled module cmsis-core doesn't match specification >=0.1.0,<0.2.0 (when trying to find dependencies for cmsis-core-k64f)\_[0m

\_[1m\_[31merror\_[39m: \_[22m\_[31mRequired dependency "mbed-hal" of "mbed 3.0.3 at d:\datastore\public\jobs\yr2015\2212\work\20150616\lwm2m-client\yotta\_modules\mbed" is not installed.\_[0m

\_[1m\_[31merror\_[39m: \_[22m\_[31mRequired dependency "cmsis-core" of "mbed 3.0.3 at d:\datastore\public\jobs\yr2015\2212\work\20150616\lwm2m-client\yotta\_modules\mbed" is not installed.\_[0m

GCC version is: 4.9.3

CMake Error at test/CMakeLists.txt:56 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:79 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:102 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:125 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:148 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:171 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:194 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:217 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:240 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:263 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:286 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:309 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:332 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:355 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:378 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:401 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:424 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:447 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:470 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

CMake Error at test/CMakeLists.txt:493 (add\_executable):

add\_executable cannot create target "lwm2m-client-test-lwm2m" because

another target with the same name already exists. The existing target is

an executable created in source directory

"d:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/test".

See documentation for policy CMP0002 for more details.

-- Configuring incomplete, errors occurred!

See also "D:/datastore/public/jobs/yr2015/2212/work/20150616/lwm2m-client/build/frdm-k64f-gcc/CMakeFiles/CMakeOutput.log".

\_[1m\_[31merror\_[39m: \_[22m\_[31mcommand ['cmake', '-D', 'CMAKE\_BUILD\_TYPE=RelWithDebInfo', '-G', 'Ninja', '.'] failed\_[0m

mbed-ls: detecting connected mbed-enabled devices...

mbed-ls: detected K64F, console at: COM21, mounted at: E:

yotta: search for mbed-target:k64f

found target 'frdm-k64f-gcc'

found target 'frdm-k64f-armcc'

mbedgt: available targets:

got yotta target 'frdm-k64f-gcc'

mbedgt: calling yotta to build your sources and tests: yotta -v --target=frdm-k64f-gcc,\* build

got yotta target 'frdm-k64f-armcc'

This is how you normally upgrade yotta

$ pip install yotta –U

This time it worked. See below:

Downloading/unpacking yotta from https://pypi.python.org/packages/source/y/yotta/yotta-0.4.3.tar.gz#md5=b579612f9e0b2145c486987e13d96bd1

Running setup.py (path:c:\users\simhug01\appdata\local\temp\pip\_build\_simhug01\yotta\setup.py) egg\_info for package yotta

Requirement already up-to-date: semantic-version>=2.3.1,<3 in c:\mbed\_tools\python27\lib\site-packages (from yotta)

Requirement already up-to-date: requests>=2.5,<3 in c:\mbed\_tools\python27\lib\site-packages (from yotta)

Requirement already up-to-date: PyGithub>=1.25,<2 in c:\mbed\_tools\python27\lib\site-packages (from yotta)

Requirement already up-to-date: colorama>=0.3,<0.4 in c:\mbed\_tools\python27\lib\site-packages (from yotta)

Requirement already up-to-date: hgapi>=1.7,<2 in c:\mbed\_tools\python27\lib\site-packages (from yotta)

Requirement already up-to-date: Jinja2>=2.7.0,<3 in c:\mbed\_tools\python27\lib\site-packages (from yotta)

Downloading/unpacking cryptography>=0.8 from https://pypi.python.org/packages/cp27/c/cryptography/cryptography-0.9.1-cp27-none-win32.whl#md5=1a2c38aab79edb4cc144e87130cdd1fa (from yotta)

Downloading/unpacking PyJWT>=1.0,<2.0 from https://pypi.python.org/packages/2.7/P/PyJWT/PyJWT-1.3.0-py2.py3-none-any.whl#md5=99b518852928e8c66a703400c2fb7197 (from yotta)

Downloading PyJWT-1.3.0-py2.py3-none-any.whl

Requirement already up-to-date: pathlib>=1.0.1,<1.1 in c:\mbed\_tools\python27\lib\site-packages (from yotta)

Downloading/unpacking jsonschema>=2.4.0,<3.0 from https://pypi.python.org/packages/py2.py3/j/jsonschema/jsonschema-2.5.1-py2.py3-none-any.whl#md5=20a47c9d9bc9357d8c731cfc19e3f968 (from yotta)

Downloading jsonschema-2.5.1-py2.py3-none-any.whl

Downloading/unpacking argcomplete>=0.8.0,<1.0 from https://pypi.python.org/packages/2.7/a/argcomplete/argcomplete-0.8.9-py2.py3-none-any.whl#md5=92afca9b9a5642efdf5d187a86cabc0e (from yotta)

Downloading argcomplete-0.8.9-py2.py3-none-any.whl

Requirement already up-to-date: mbed-test-wrapper>=0.0.2,<0.1.0 in c:\mbed\_tools\python27\lib\site-packages (from yotta)

Requirement already up-to-date: valinor>=0.0.0,<1.0 in c:\mbed\_tools\python27\lib\site-packages (from yotta)

Requirement already up-to-date: ntfsutils>=0.1.3,<0.2 in c:\mbed\_tools\python27\lib\site-packages (from yotta)

Requirement already up-to-date: markupsafe in c:\mbed\_tools\python27\lib\site-packages (from Jinja2>=2.7.0,<3->yotta)

Downloading/unpacking idna from https://pypi.python.org/packages/2.7/i/idna/idna-2.0-py2.py3-none-any.whl#md5=40fa688bb01833b2a807fcc40ddaa8c0 (from cryptography>=0.8->yotta)

Requirement already up-to-date: pyasn1 in c:\mbed\_tools\python27\lib\site-packages (from cryptography>=0.8->yotta)

Requirement already up-to-date: enum34 in c:\mbed\_tools\python27\lib\site-packages (from cryptography>=0.8->yotta)

Requirement already up-to-date: six>=1.4.1 in c:\mbed\_tools\python27\lib\site-packages (from cryptography>=0.8->yotta)

Requirement already up-to-date: ipaddress in c:\mbed\_tools\python27\lib\site-packages (from cryptography>=0.8->yotta)

Downloading/unpacking setuptools from https://pypi.python.org/packages/3.4/s/setuptools/setuptools-17.1.1-py2.py3-none-any.whl#md5=8a540687a8341088e2b280ad2ceb4e5b (from cryptography>=0.8->yotta)

Downloading/unpacking cffi>=0.8 from https://pypi.python.org/packages/cp27/c/cffi/cffi-1.1.2-cp27-none-win32.whl#md5=f3ecc74b102254f9be7341e99955e224 (from cryptography>=0.8->yotta)

Downloading/unpacking functools32 (from jsonschema>=2.4.0,<3.0->yotta)

Downloading functools32-3.2.3-1.tar.gz

Running setup.py (path:c:\users\simhug01\appdata\local\temp\pip\_build\_simhug01\functools32\setup.py) egg\_info for package functools32

C:\mbed\_tools\Python27\lib\site-packages\setuptools\dist.py:285: UserWarning: Normalizing '3.2.3-1' to '3.2.3.post1'

normalized\_version,

warning: no files found matching '\*.txt'

no previously-included directories found matching 'build'

no previously-included directories found matching 'dist'

no previously-included directories found matching '.git\*'

Requirement already up-to-date: pyyaml>=3,<4 in c:\mbed\_tools\python27\lib\site-packages (from valinor>=0.0.0,<1.0->yotta)

Downloading/unpacking pyOCD>=0.3,<1.0 from https://pypi.python.org/packages/source/p/pyOCD/pyOCD-0.4.2.zip#md5=55325a23acf9988430b3de6c1be36b5b (from valinor>=0.0.0,<1.0->yotta)

Running setup.py (path:c:\users\simhug01\appdata\local\temp\pip\_build\_simhug01\pyOCD\setup.py) egg\_info for package pyOCD

Downloading/unpacking project-generator>=0.5.7,<1.0 from https://pypi.python.org/packages/source/p/project\_generator/project\_generator-0.5.10.zip#md5=cb9c9d507d76d5097535fa84cd292323 (from valinor>=0.0.0,<1.0->yotta)

Running setup.py (path:c:\users\simhug01\appdata\local\temp\pip\_build\_simhug01\project-generator\setup.py) egg\_info for package project-generator

Downloading/unpacking pycparser from https://pypi.python.org/packages/source/p/pycparser/pycparser-2.14.tar.gz#md5=a2bc8d28c923b4fe2b2c3b4b51a4f935 (from cffi>=0.8->cryptography>=0.8->yotta)

Running setup.py (path:c:\users\simhug01\appdata\local\temp\pip\_build\_simhug01\pycparser\setup.py) egg\_info for package pycparser

warning: no previously-included files matching 'yacctab.\*' found under directory 'tests'

warning: no previously-included files matching 'lextab.\*' found under directory 'tests'

warning: no previously-included files matching 'yacctab.\*' found under directory 'examples'

warning: no previously-included files matching 'lextab.\*' found under directory 'examples'

Downloading/unpacking pywinusb (from pyOCD>=0.3,<1.0->valinor>=0.0.0,<1.0->yotta)

Running setup.py (path:c:\users\simhug01\appdata\local\temp\pip\_build\_simhug01\pywinusb\setup.py) egg\_info for package pywinusb

no previously-included directories found matching 'dist'

Installing collected packages: yotta, cryptography, PyJWT, jsonschema, argcomplete, idna, setuptools, cffi, functools32, pyOCD, project-generator, pycparser, pywinusb

Found existing installation: yotta 0.4.0

Uninstalling yotta:

Successfully uninstalled yotta

Running setup.py install for yotta

Installing yotta-script.py script to C:\mbed\_tools\Python27\Scripts

Installing yotta.exe script to C:\mbed\_tools\Python27\Scripts

Installing yotta.exe.manifest script to C:\mbed\_tools\Python27\Scripts

Installing yt-script.py script to C:\mbed\_tools\Python27\Scripts

Installing yt.exe script to C:\mbed\_tools\Python27\Scripts

Installing yt.exe.manifest script to C:\mbed\_tools\Python27\Scripts

Found existing installation: cryptography 0.9

Uninstalling cryptography:

Successfully uninstalled cryptography

Found existing installation: PyJWT 1.1.0

Uninstalling PyJWT:

Successfully uninstalled PyJWT

Found existing installation: jsonschema 2.4.0

Uninstalling jsonschema:

Successfully uninstalled jsonschema

Found existing installation: argcomplete 0.8.8

Uninstalling argcomplete:

Successfully uninstalled argcomplete

Found existing installation: idna 1.1

Uninstalling idna:

Successfully uninstalled idna

Found existing installation: setuptools 16.0

Uninstalling setuptools:

Successfully uninstalled setuptools

Found existing installation: cffi 0.9.2

Uninstalling cffi:

Successfully uninstalled cffi

Running setup.py install for functools32

C:\mbed\_tools\Python27\lib\site-packages\setuptools\dist.py:285: UserWarning: Normalizing '3.2.3-1' to '3.2.3.post1'

normalized\_version,

warning: no files found matching '\*.txt'

no previously-included directories found matching 'build'

no previously-included directories found matching 'dist'

no previously-included directories found matching '.git\*'

Found existing installation: pyOCD 0.3.3

Uninstalling pyOCD:

Successfully uninstalled pyOCD

Running setup.py install for pyOCD

Installing pyocd-flashtool-script.py script to C:\mbed\_tools\Python27\Scripts

Installing pyocd-flashtool.exe script to C:\mbed\_tools\Python27\Scripts

Installing pyocd-flashtool.exe.manifest script to C:\mbed\_tools\Python27\Scripts

Installing pyocd-gdbserver-script.py script to C:\mbed\_tools\Python27\Scripts

Installing pyocd-gdbserver.exe script to C:\mbed\_tools\Python27\Scripts

Installing pyocd-gdbserver.exe.manifest script to C:\mbed\_tools\Python27\Scripts

Installing pyocd-tool-script.py script to C:\mbed\_tools\Python27\Scripts

Installing pyocd-tool.exe script to C:\mbed\_tools\Python27\Scripts

Installing pyocd-tool.exe.manifest script to C:\mbed\_tools\Python27\Scripts

Found existing installation: project-generator 0.5.9

Uninstalling project-generator:

Successfully uninstalled project-generator

Running setup.py install for project-generator

Installing project\_generator-script.py script to C:\mbed\_tools\Python27\Scripts

Installing project\_generator.exe script to C:\mbed\_tools\Python27\Scripts

Installing project\_generator.exe.manifest script to C:\mbed\_tools\Python27\Scripts

Installing pgen-script.py script to C:\mbed\_tools\Python27\Scripts

Installing pgen.exe script to C:\mbed\_tools\Python27\Scripts

Installing pgen.exe.manifest script to C:\mbed\_tools\Python27\Scripts

Found existing installation: pycparser 2.10

Uninstalling pycparser:

Successfully uninstalled pycparser

Running setup.py install for pycparser

warning: no previously-included files matching 'yacctab.\*' found under directory 'tests'

warning: no previously-included files matching 'lextab.\*' found under directory 'tests'

warning: no previously-included files matching 'yacctab.\*' found under directory 'examples'

warning: no previously-included files matching 'lextab.\*' found under directory 'examples'

Build the lexing/parsing tables

Running setup.py install for pywinusb

no previously-included directories found matching 'dist'

Skipping installation of C:\mbed\_tools\Python27\Lib\site-packages\pywinusb\\_\_init\_\_.py (namespace package)

Installing C:\mbed\_tools\Python27\Lib\site-packages\pywinusb-0.3.6-py2.7-nspkg.pth

Successfully installed yotta cryptography PyJWT jsonschema argcomplete idna setuptools cffi functools32 pyOCD project-generator pycparser pywinusb

Cleaning up...

If it didn’t work, then can take the source tarball from the yotta releases page, unroll it and do

$ python setup.py install

This will upgrade the existing installation.