

How to run C-code with fftw?

Option 1:

- With mingw and command line
- Explained in this document for Windows and Mac OS, see point 1 and 2
- Use fastconvolution_GCC.zip from Toledo

Option 2:

- With LINUX (like seen in other courses in the 1st semester of the 3th year)
- Explained in point 3 in this document

Option 3:

- With Visual Studio Community Edition
- Like done in the DSP exercise manual
- After opening project file, click Project -> Retarget Solution
- Remark: exe file ends up in the release directory instead of the main directory
- Use fastconvolution_VS2017W10.zip from Toledo

Tips and tricks if it doesn't work (check this before emailing us!)

- First install the fftw library + set the path (like explained in this document)
- Run the Matlab-code FIRfiltering: here on line 50 the command "eval" runs your C-code
- If this went well, there must be a **file created "result_conv.bin"** in the directory "C:\temp\fastconvolution" (check it gets the date of "now" so you are sure this is just created)
- If this didn't go well
 - Test the "C:\temp\fastconvolution\fastconvolution.exe" yourself by double clicking on it. If this gives on error about "**dll problems**", there is a problem with the fftw library path. So carefully redo the first step then
 - On Windows computers:
 - Add the path to the fftw library through Window's control panel (<https://www.computerhope.com/issues/ch000549.htm>) and **restart your computer**
 - Make sure to run Matlab and the terminal window as administrator
 - Try **running the fastconvolution.exe yourself in the command prompt** (first run the FIRfiltering.m in Matlab to make sure filter.bin and signal.bin are created in your directory, then run the fastconvolution.exe yourself, and then run the FIRfiltering.m in Matlab again)
- Make sure you run the cmd as an administrator
- Make sure you run Matlab as an administrator
- Plan B: put fftw3.lib in the exe folder

1. Option 1 (mingw and command line): Windows

1. Install ***mingw 64bit*** [already installed in PC-classes]

Download from here:

<https://sourceforge.net/projects/codeblocks/files/Binaries/20.03/Windows/codeblocks-20.03mingw-setup.exe/download>

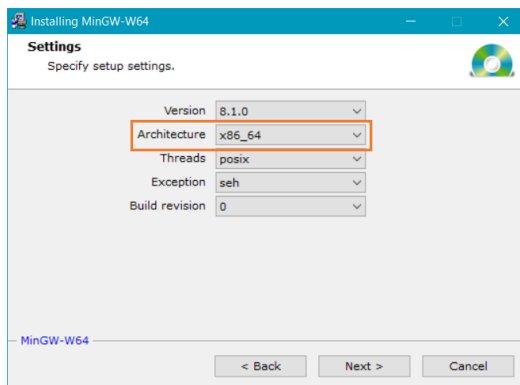
Or from here:

<https://jmeubank.github.io/tdm-gcc/download/>

Or from here:

<https://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win32/Personal%20Builds/mingw-builds/installer/mingw-w64-install.exe/download>

Install and choose as Architecture x86_64 (see screenshot below).



2. Go to the installation directory `C:\Program Files\mingw-w64\x86_64-8.1.0-posix-seh-rt_v6-rev0`

Double click on **mingw-w64.bat** (now the path to use gcc is correctly set). Leave the command prompt open (we will use it in step 5)

(If you want to test if gcc works correctly, you can type the command: `gcc --version`)

3. Download the code **Fastconvolution_GCC.zip** from Toledo and unzip in `C:\temp\fastconvolution`

4. Download the **fftw library** from <ftp://ftp.fftw.org/pub/fftw/fftw-3.3.5-dll64.zip> (use an **Internet Explorer browser!**) and unzip to `C:\temp\fftw` (choose the same main directory as your code!) (or install via

<http://www.fftw.org/install/windows.html>)

5. Add the **fftw library directory** to your **path**:

- In the Windows Search bar, search for and then select: **System (Control Panel)/'Systeem' (Configuratiescherm)**
- Click on the **Advanced system settings/Geavanceerde systeeminstellingen** link.

- Click on **Environment Variables/Omgevingsvariabelen**. In the section **System Variables/Systeemvariabelen** find the PATH environment variable and select it. Click **Edit/Bewerken**.
- In the new window that has been opened, add a new line with the directory of the fftw library (here in this tutorial, it is C:\temp\fftw). Click OK and close all remaining windows by clicking OK.
- Restart your computer. On a command prompt, verify that the directory had been successfully added to the path by typing the command: `echo %path%`
- Alternative for the previous steps: add the fftw library directly to the path in the command prompt with the command: `set path=C:\temp\fftw\;%path%`

6. **Go to your code directory** in the command prompt with the command:
`cd C:\temp\fastconvolution`

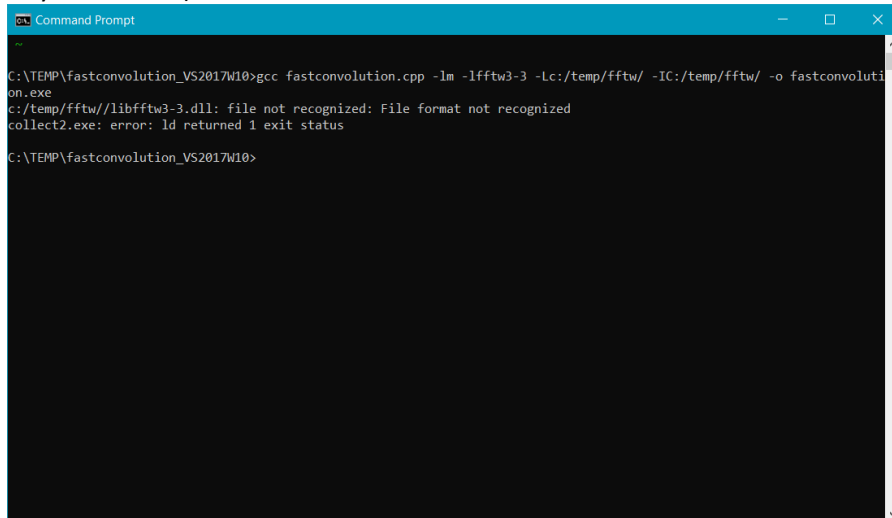
7. **Build** code in the command prompt with
`gcc fastconvolution.cpp -lm -lfftw3-3 -LC:/temp/fftw/ -IC:/temp/fftw/ -o fastconvolution.exe`

8. **Check** if the exe is created in your code directory `C:\temp\fastconvolution\fastconvolution.exe`

9. You are done!

Possible error messages:

*This error you get when you try to mix 32 and 64 bit (if the version of mingw doesn't match the version of your fftw dll)



```

C:\TEMP\fastconvolution_VS2017W10>gcc fastconvolution.cpp -lm -lfftw3-3 -LC:/temp/fftw/ -IC:/temp/fftw/ -o fastconvoluti
on.exe
c:/temp/fftw/libfftw3-3.dll: file not recognized: File format not recognized
collect2.exe: error: ld returned 1 exit status
C:\TEMP\fastconvolution_VS2017W10>

```

2. Option 1 (mingw and command line): Mac OS

1. Open a terminal window

2. Check if GGC available by: `gcc --version`

If not, install gcc, see, e.g., <https://www.mkyong.com/mac/how-to-install-gcc-compiler-on-mac-os-x/>

3. Get fftw source code from <ftp://ftp.fftw.org/pub/fftw/fftw-3.3.8.tar.gz> and unzip to `/Users/Username/Documents/tmp/fftw`

4. Go to fftw directory by

```
cd /Users/Username/Documents/tmp/fftw
```

and compile fftw by

```
./configure && make
```

```
sudo make install
```

5. Download the code **Fastconvolution_GCC.zip** from Toledo and unzip to `/Users/Username/Documents/tmp/fastconvolution`

6. Go to fastconvolution directory by

```
cd /Users/Username/Documents/tmp/fastconvolution
```

and build fastconvolution by

```
gcc fastconvolution.cpp -lm -lfftw3 -L/Users/Username/Documents/tmp/fftw/.libs -l  
/Users/Username/Documents/tmp/fftw/api -o fastconvolution
```

7. You are done!

3. Option 2: Linux

Install FFTW on Linux

Ref: http://micro.stanford.edu/wiki/Install_FFTW3

1. Grab the latest version of FFTW on <http://fftw.org/download.html> (3.3.8 as of writing)
2. Unzip the archive
3. Open the archive folder in a terminal (e.g. ``cd ~/Downloads/fftw-3.3.8``)
4. Run ``./configure --prefix=$HOME/usr --enable-shared=yes``
5. Run ``make --jobs=8``
6. Run ``make install``
 - this installs a bunch of files under ``~/usr/include`` and ``~/usr/lib``

Code fixes

Linker problems when running from Matlab

To fix linker problems when Matlab wants to run the `fastconvolution.out` file, the following command has to be run in the terminal:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:~/usr/lib
```

C-code `fastconvolution.cpp`: int vs long int warning

Replace ``%d`` with ``%ld``

Matlab-code `FIRfiltering.m`: eval

Causes ``Error using fread`` when MatLab tries to load ``result_conv.bin``

Make sure line 50 contains the right file name.

Note: On Linux you have to explicitly state that you want to execute a program from the current working directory. (e.g. ``./fastconvolution.out``)

Segmentation Fault When Running ``fastconvolution.out``

``fastconvolution.out`` throws a segmentation fault when some files it tries to read, are not present. Run ``FIRfiltering.m`` first and make sure it generates all the necessary data files. (Note: A segmentation fault can have many other causes.)

Makefile Commands

- ``make fastconvolution``: compile ``fastconvolution.cpp``
(make sure that the paths in ``Makefile`` are correctly set)
- ``make clean-data``: delete all files that a run of ``FIRfiltering.m`` generates.
- ``make clean``: delete all generated files (i.e. data files and compiler output)