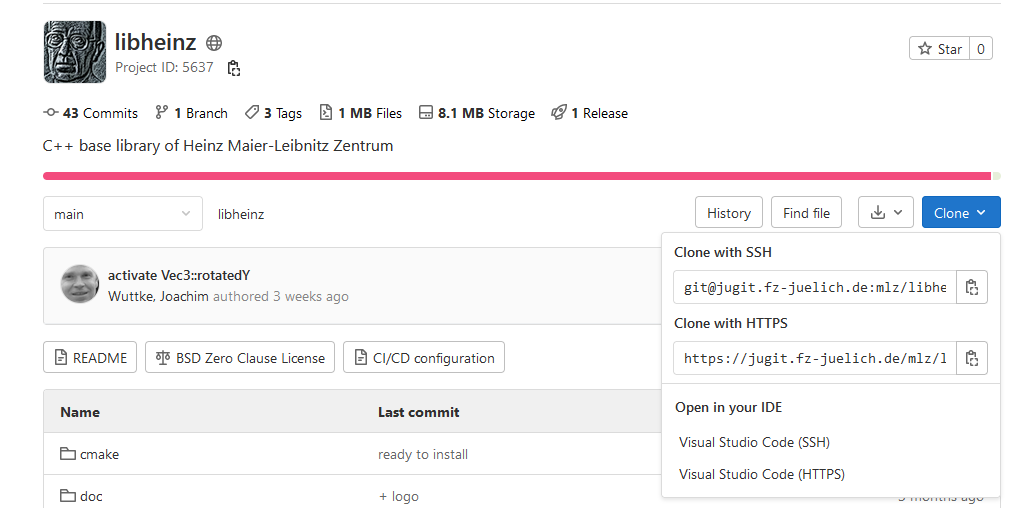
**INSTALLATION OF LIBHEINZ**

Clone libheinz library from gitlab

<https://jugit.fz-juelich.de/mlz/libheinz>

Clone option Open in your IDE VisualStudio Code (HTTPS)



Clone the folder from VSCode here -> D:\libheinz

KEEP the -DCMAKE\_INSTALL\_PREFIX=C:/opt/x64 option : it is where everything is created on the disk. Commande du type : cmake .. -DCMAKE\_INSTALL\_PREFIX=MONCHEMIN

* follow the script for windows here : D:\libheinz\.gitlab-ci.yml
* Just change the version of the generator (here Visual Studio 17 2022)

windows:

  tags:

  - Windows

  stage: build

  script:

    - cmake -G "Visual Studio 17 2022" -A x64 -T host=x64 -DCMAKE\_INSTALL\_PREFIX=C:/opt/x64 -S . -B build

    - cd build

    - cmake --build . --config Release

    - Get-Location

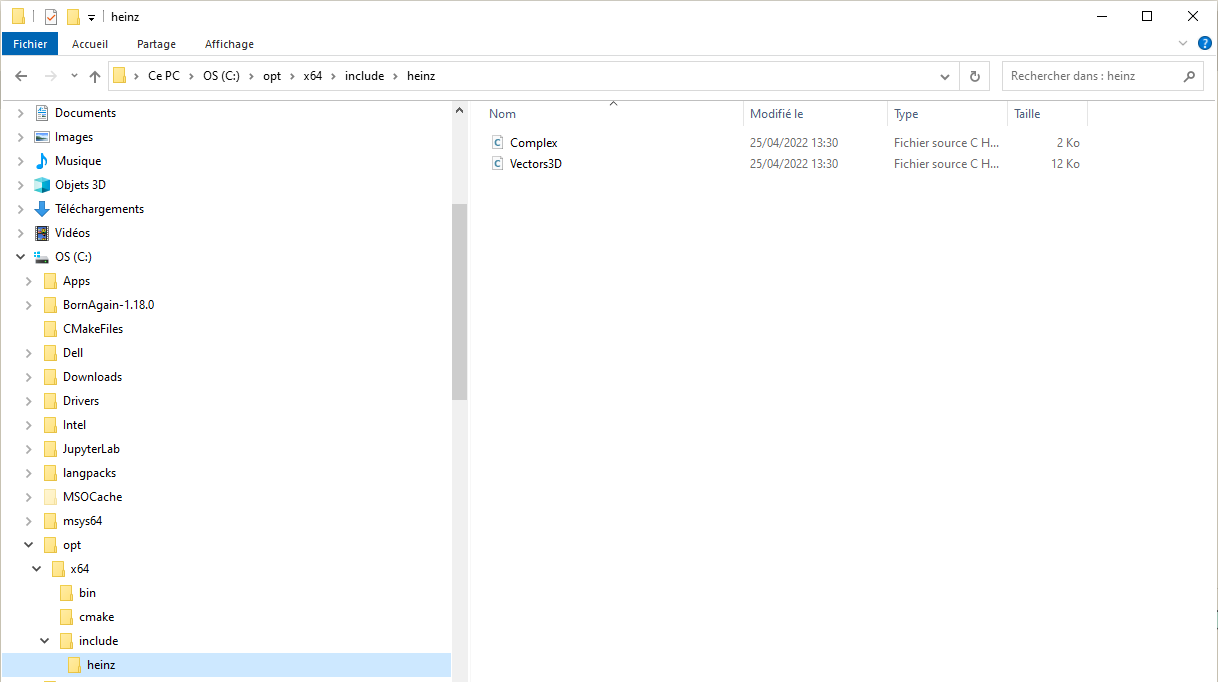
    - dir

*# debug-only: - dumpbin.exe /HEADERS lib\liblmfit.lib*

    - ctest -C Release --output-on-failure

    - cmake --install .

Library is created here : C:\opt\x64



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Still need to do this second build because a path later is pointing there … don’t know why ?

set(LibHeinz\_INCLUDE\_DIR "C:/Program Files (x86)/LibHeinz/include")

and not the right one :

set(LibHeinz\_INCLUDE\_DIR "C:/opt/x64/include")

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Create an empty build folder and open the ‘libheinz’ folder in VSCode

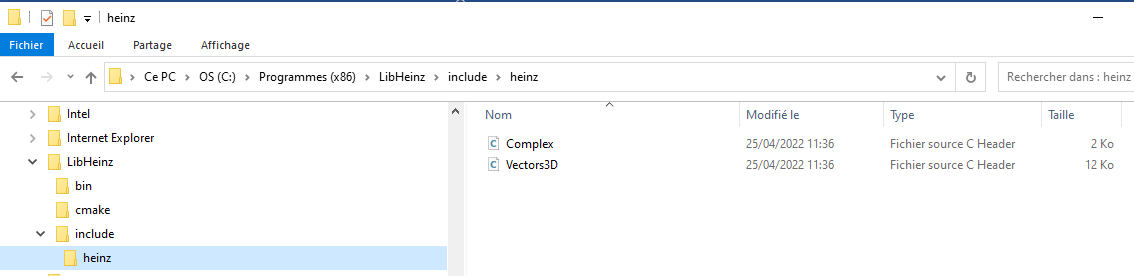
mkdir build

cd build

cmake ..

cmake - - install .

Verify that the library is installed here :

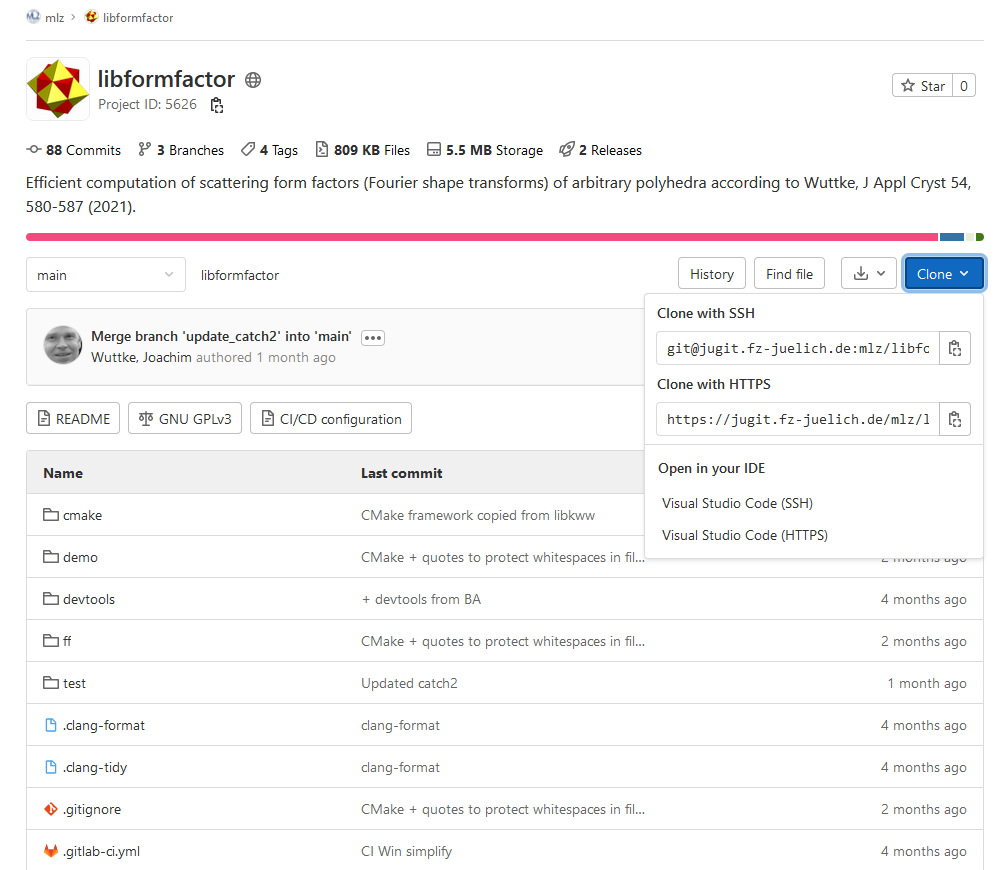


\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**INSTALLATION OF LIBFORMFACTOR**

<https://jugit.fz-juelich.de/mlz/libformfactor>

Clone the folder from VSCode here -> D:\libformfactor



Follow the script contained in the file :

* follow the script for windows here : D:\libheinz\.gitlab-ci.yml
* Just change the version of the generator (here Visual Studio 17 2022)

windows:

  tags:

  - Windows

  stage: build

  script:

    - cmake -G "Visual Studio 17 2022" -A x64 -T host=x64 -DCMAKE\_INSTALL\_PREFIX=C:/opt/x64 -S . -B build

    - cd build

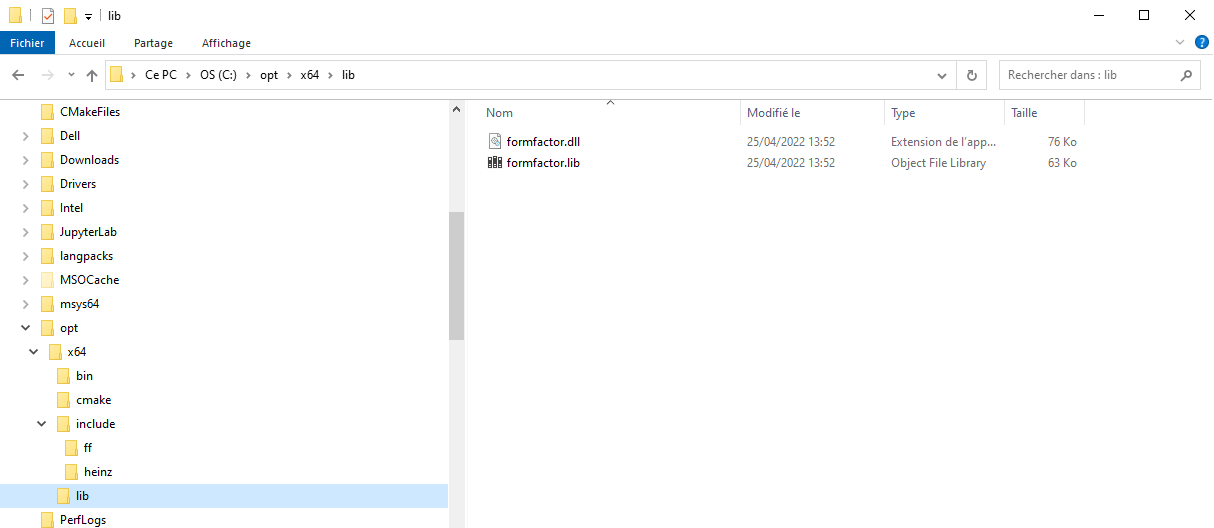
    - cmake --build . --config Release

    - ctest -C Release --output-on-failure

    - cmake --install .

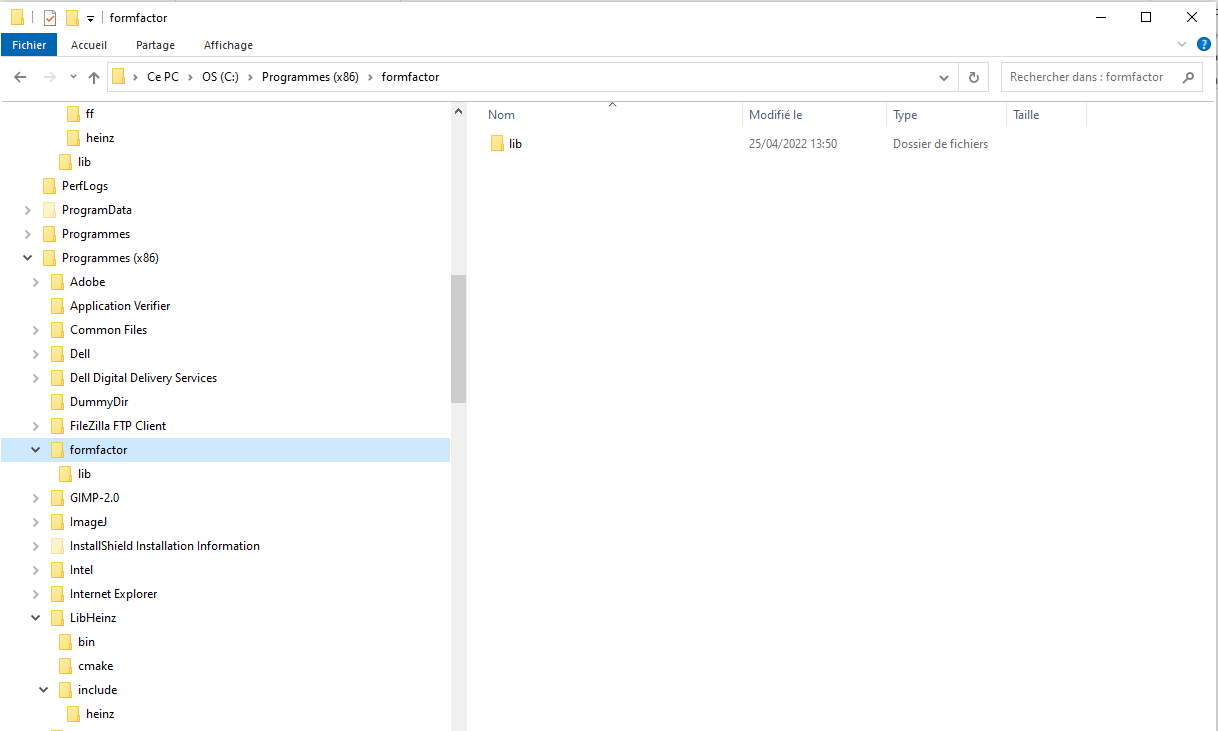
Installation finished, looks like this :

Correct new location is C:\opt\x64



But there is also in either C:\Programmes (x86) or C:\Programmes :

An empty formfactor folder (lib folder is empty as well)



**DEMO1**

Simple test.cpp file writing a text in output

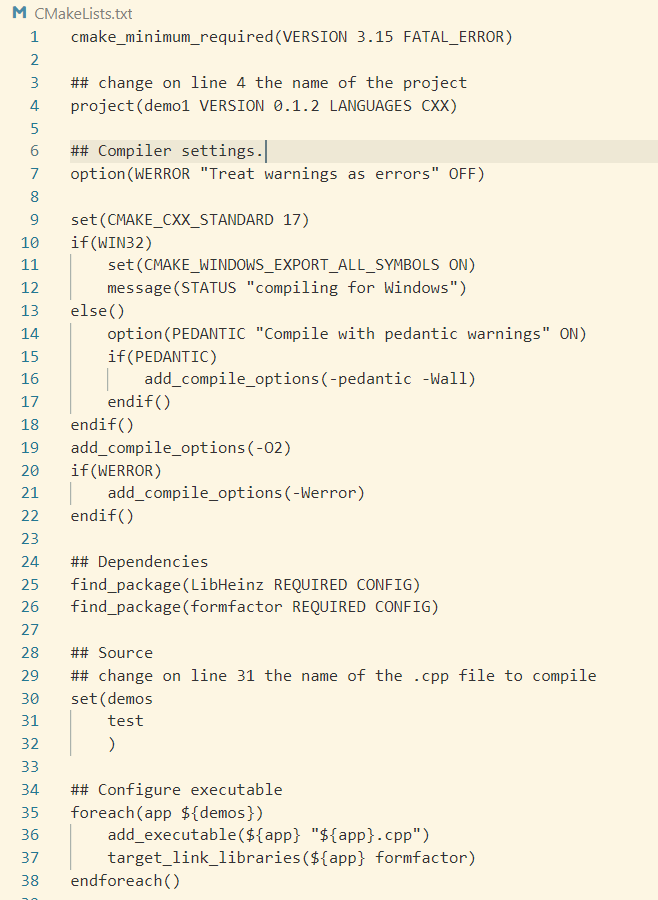


* Create folder demo1 here D:\libformfactor\demo1
* Put two files inside :

test.cpp

CMakelists.txt

* In CMakelists.txt, change the project name and the source file



* Create a new terminal in VS CODE and type :

    - cmake -G "Visual Studio 17 2022" -A x64 -T host=x64 -DCMAKE\_INSTALL\_PREFIX=C:/opt/x64 -S . -B build

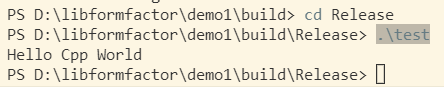
    - cd build

    - cmake --build . --config Release

- cd Release

- .\octahedron

test.exe is created in D:\libformfactor\demo1\build\Release  !!!



**To continue !!!!!!!!!**

**DEMO**

PS D:\libformfactor\demo> cmake -G "Visual Studio 17 2022" -A x64 -T host=x64 -S . -B build

-- Selecting Windows SDK version 10.0.19041.0 to target Windows 10.0.19044.

-- The CXX compiler identification is MSVC 19.30.30709.0

-- Detecting CXX compiler ABI info

-- Detecting CXX compiler ABI info - done

-- Check for working CXX compiler: C:/Program Files (x86)/Microsoft Visual Studio/2022/BuildTools/VC/Tools/MSVC/14.30.30705/bin/Hostx64/x64/cl.exe - skipped

-- Detecting CXX compile features

-- Detecting CXX compile features - done

-- compiling for Windows

-- Configuring done

-- Generating done

-- Build files have been written to: D:/libformfactor/demo/build

PS D:\libformfactor\demo\build> cd build

PS D:\libformfactor\demo\build> cmake --build . --config Release

Microsoft (R) Build Engine version 17.0.0+c9eb9dd64 pour .NET Framework

Copyright (C) Microsoft Corporation. Tous droits réservés.

Checking Build System

Building Custom Rule D:/libformfactor/demo/CMakeLists.txt

octahedron.cpp

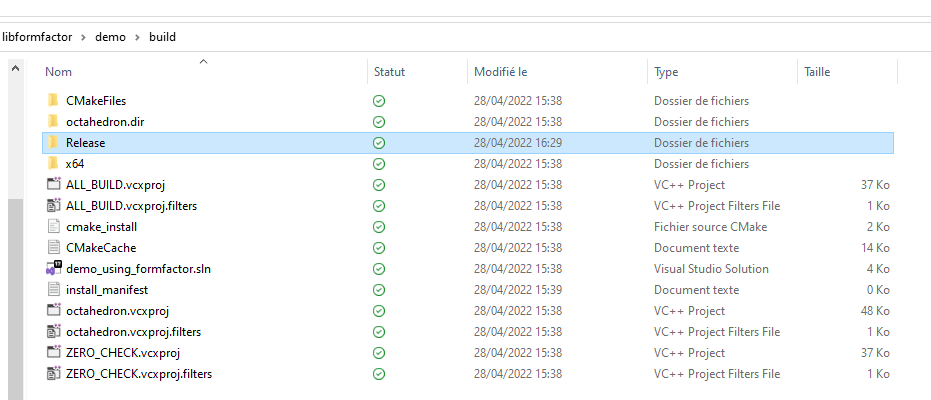
octahedron.vcxproj -> D:\libformfactor\demo\build\Release\octahedron.exe

Building Custom Rule D:/libformfactor/demo/CMakeLists.txt

PS D:\libformfactor\demo\build> cmake --install .

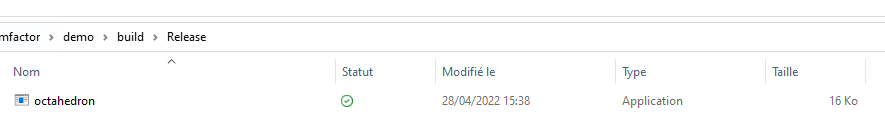
-- Install configuration: "Release"

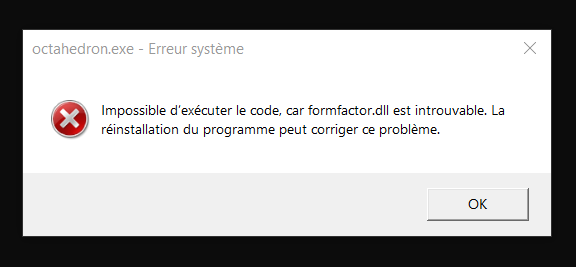
The demo/build file now contains plenty of files, notably the Release folder.



This Release folder contains an octahedra.exe file.

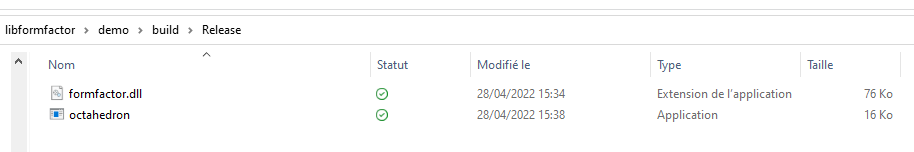
However, launching it as is only gives an error.



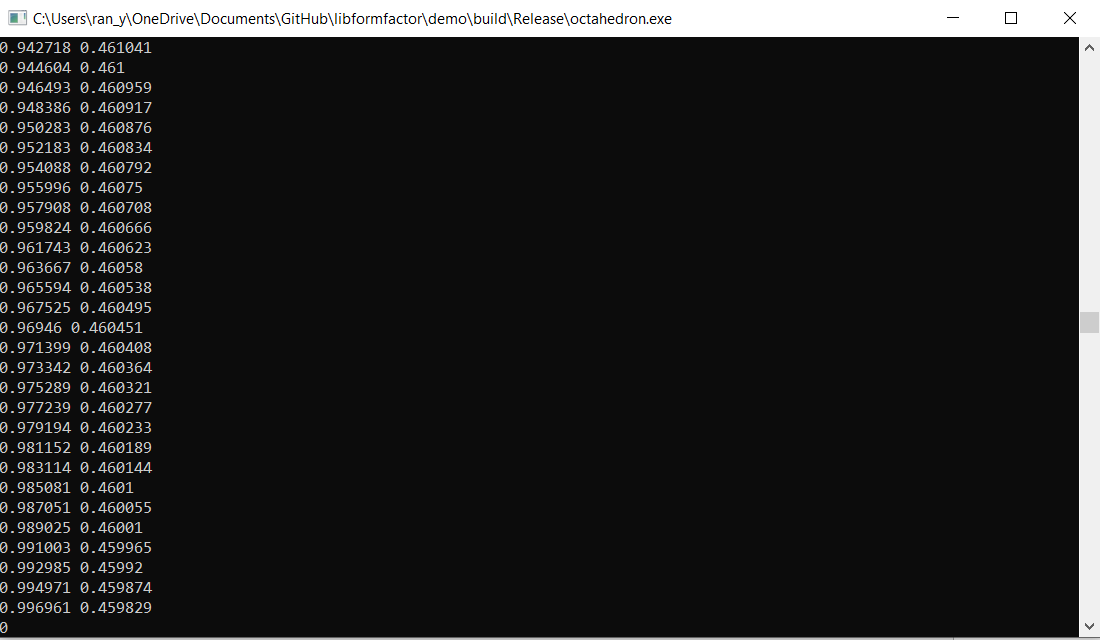


In order to make this program work, you need to find a formfactor.dll file and add it to the folder.

If you have previously properly followed all the instructions in this document, you may find one in D:/libformfactor/build/bin/Release



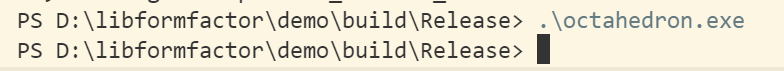
Now, when launching octahedron.exe, the program should be running !



Instructions for demo, in D:\libformfactor\demo

* cmake -G "Visual Studio 17 2022" -A x64 -T host=x64 -S . -B build
* cd build
* cmake --build . --config Release
* cmake --install .

\*\*\*\* OLD STUFF \*\*\*\*\*\*\*\*\*\*\*\*\*\*



Create an empty build folder and open the ‘libheinz’ folder in VSCode

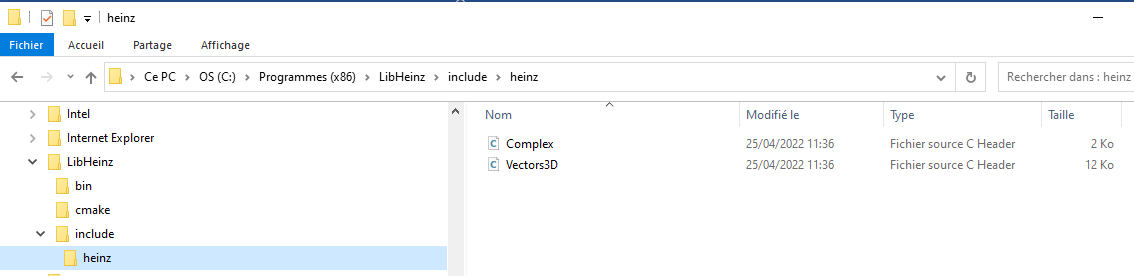
mkdir build

cd build

cmake ..

cmake - - install .

Verify that the library is installed here :



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**IMPORTANT TO DO** : Go to the following cmake file :

C:\Program Files (x86)\LibHeinz\cmake\ LibheinzConfig.cmake

Add "…" in line 12 :

#### set(LibHeinz\_INCLUDE\_DIR C:/Program Files (x86)/LibHeinz/include)

set(LibHeinz\_INCLUDE\_DIR "C:/Program Files (x86)/LibHeinz/include")

**This allows the path to be properly read ☺**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TO TEST \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

set(LibHeinz\_INCLUDE\_DIR "C:/opt/x64/include")