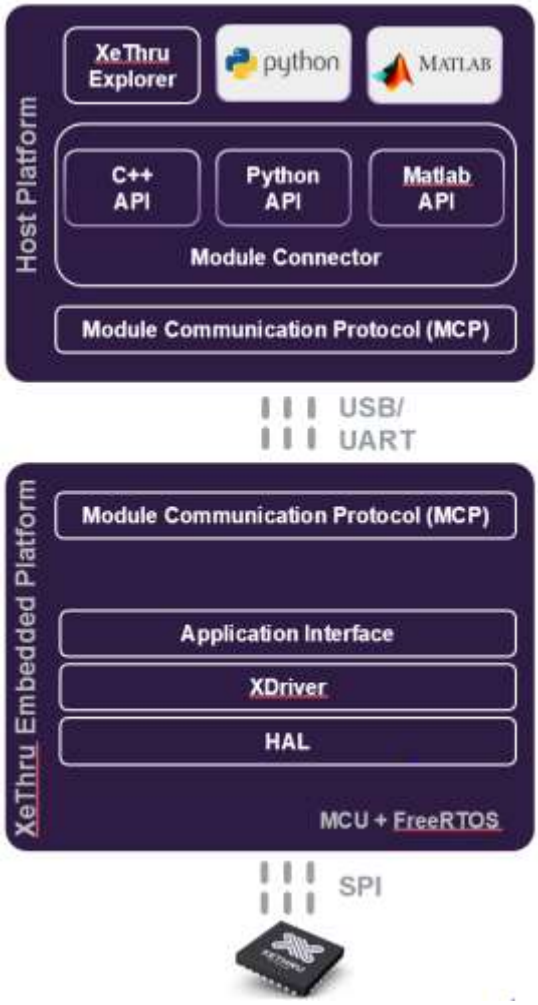
1 MODULECONNECTOR

Module Connector is a host library that communicates with your XeThru-module. Learning moduleconnector is a critical part of learning how to use the XeThru modules. Both the module and ModuleConnector use the module communication protocol (MCP). MCP is the lowest level communication protocol implementation for the XeThru modules, used on the module firmware and XeThru host components. ModuleConnector supports both synchronous and asynchronous messages. It also includes API wrappers for MATLAB, Python and C++ programming languages.



To download the moduleconnector for windows visit the Xethru resources website. Extract the content of the moduleconnector .zip to a folder and rename it to "Moduleconnector". This folder is self contained and no installation is needed. To learn more about moduleconnector's functions, variables, constructs etc. go to "Moduleconnector\doc\index.html".

1.1 C++:

1.1.1 Setup C++ API:

For Windows:

Start with downloading and installing MinGW -64 and MSYS -64

You will also need to add their /bin directories to your PATH

Now run the MakeFile in the "\ModuleConnector\examples" folder

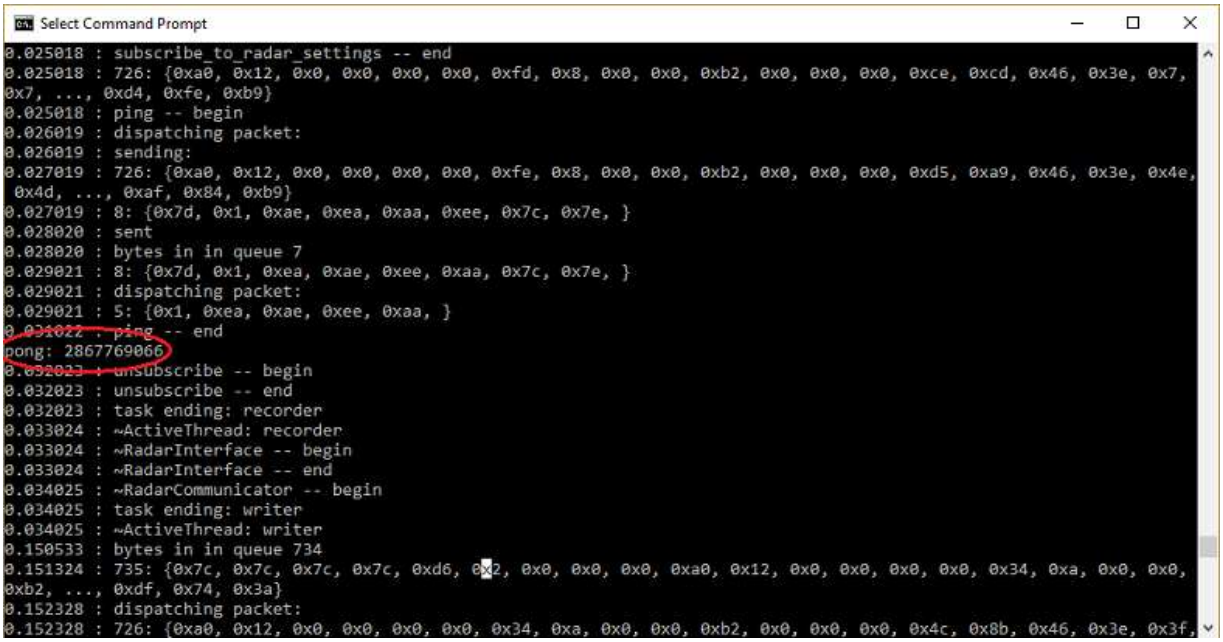
This will compile and link all the files with the ModuleConnector library

To see how this is done manually, visit "file:///ModuleConnector/doc/html/link_cpp.xhtml"

1.1.2 C++ Example

Now lets try to run ping on your module:

- After running the MakeFile in the example folder, you should see that every .cpp fil should have a corresponding .exe file
- Run the ping.exe file from the terminal by writing "ping COM'X'" where X is the comport your module is connected to
- The program will then return the pong



1.2 Python:

1.2.1 Setup python API

To run examples and python code via the moduleconnector it is strongly recommended to download and install Anaconda. Anaconda is an open data science platform which is very useful for doing data analysis.

If you choose not to use Anaconda just download packages with the

- python -m pip install -U pip <package>

The following packages are not required for all examples, but most examples includes at least one of them

Run cmd with following commands (if conda isn't system path variable, just run commands in anaconda prompt):

$ conda install numpy

$ conda install pyserial

$ conda install configobj

Next you will need to run the setup.py located in the "Moduleconnector\python" directory

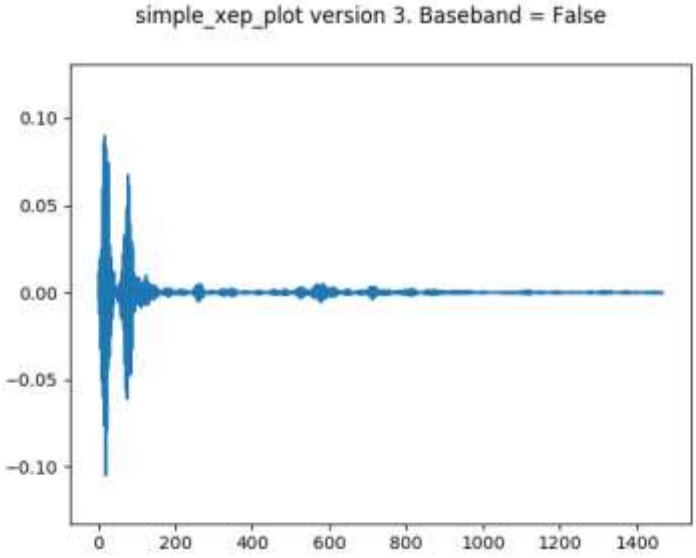
The python examples are located in the "Moduleconnector\python\pymoduleconnector\examples" directory.

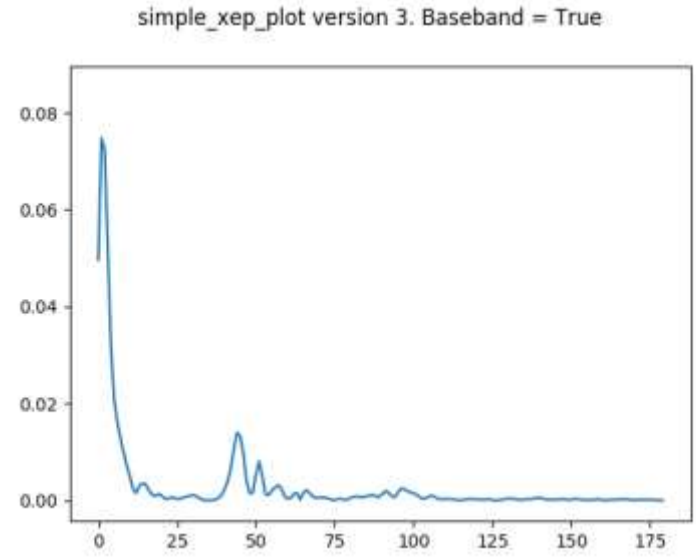
- Now cd to "ModuleConnector\python\pymoduleconnector\examples"
- When running your examples it is important to include the module as argument like so: python x4m300_presence_simpleoutput.py -d COM3
- You can locate your modules COM port by opening the Windows device manager and look for "Bossa program port" under "Ports Program Port"
- Now you are ready to run some example code on your module.
- Not all examples works on all modules, please read the .py example code to find out which module it will work on

1.2.2 Python Example:

Here is an example on how to use many of the functions for xethru modules and plot them in python: simple_xep_plot.py (only supports X4 modules)

- This example sets up the X4 as a motion detector and shows a plot of amplitude by distance
- Please read through and run the simple_xep_plot.py example. Then make sure you understand the code and what each function does.
- Try changing the variable XXXXX from YY to ZZ to switch between plotting RF data (?) and baseband data


simple_xep_plot version 3. Baseband = False
0.10
0.05
0.00
−0.05
−0.10
0
200
400
600
800
1000
1200
1400


simple_xep_plot version 3. Baseband = True
0.08
0.06
0.04
0.02
0.00
0
25
50
75
100
125
150
175

1.3 Matlab:

1.3.1 Setup Matlab API:

In matlab, open the command terminal and cd into "Moduleconnector\matlab\examples".

Now you need to add some directories to your path.

Write:

- addpath('\PATH_TO_DIR\Moduleconnector\matlab\')
- addpath('\PATH_TO_DIR\Moduleconnector\lib')
- addpath('\PATH_TO_DIR\Moduleconnector\include')

(where PATH_TO_DIR is the path to the directory Moduleconnector is located in.)

If you try to run a example and you get the error message

please visit mathoworks and download and install a compiler for your matlab.

We recommend using the free support packages from MinGW.

1.3.2 Matlab Example:

Now we will try to run the example "configure_radar_with_xep.m" (only supports X4 modules)

- Make sure you read through the script and understand how it works
- After adding all necessary paths, change the variable 'COMPORT' to the com your device is connected to.
- You can locate your modules COM port by opening the Windows device manager and look for “Bossa program port” under "Ports Program Port
- Run the example

- This example demonstrates how you can configure the radar chip using XEP(Xethru embedded platform).
- It also uses the data float message function to read the raw baseband data. The data float messages in queue are printed out in the terminal.
- The recordings from the example should end up in the example folder

Abbildung 1 Fig 7