Advanced OEM Solutions

Software Examples

OEM-PA/OEM-MC/FMC

Version 1.0

REVISION HISTORY

Date	Version	Description
2014/12/10	1.0	General Edit

CONTENTS

1. Introduction

This document describes all AOS-provided examples for working the OEM Hardware units. These basic examples are designed to illustrate various functions contained in the APIs. The source code is in included so that the user may see the code used to control the OEM Hardware. Software examples and tools explained here include:

- Using high level APIs (Wizard Example)
- Using medium level APIs (Customized Example)
- Using low level APIs (Application Example)

2. Example using High Level API (OEMPAWizardExample)

The Wizard is built with:

- Input parameters (specimen, probes, scan).
    - A special input parameter is used to update output parameters from inputs.
- Output parameters (delays etc…).

The Wizard is useful for building configuration files ("OEMPA files") with the proper focal laws required for your application.

2.1 Acquisition channel

The Kernel Wizard can be configured for single or multiple acquisition channels. This is saved in the software configuration file. The hierarchical tree of the Kernel Wizard is different depending on this choice. If you save the Kernel file with one mode and change to another mode, it will not be possible to reload this file. For example: if you save a Kernel file with the single-channel mode, it is not possible to load this file if you change to the multiple-channel mode.

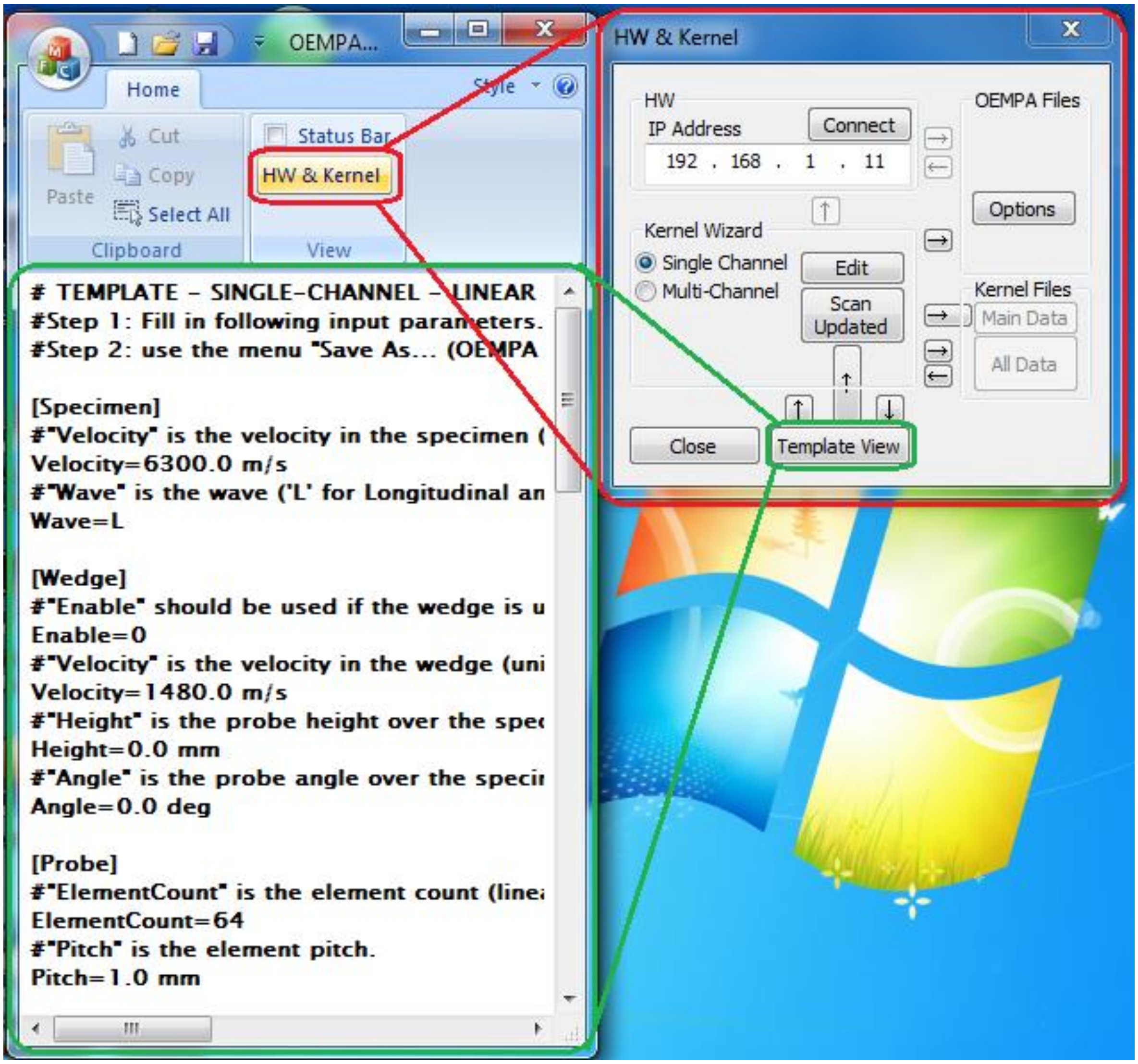
For single-channel acquisition, the template is sufficient to edit all main input parameters for a simple scan, and the "Toolbox" (See *Software_Utilities.pdf*) is not required. But for multiple acquisition channels, the template is only able to set a limited subset of input parameters. You need to use the "Toolbox" to edit the complete set of input parameters.

2.2 Example Description

The application has two items:

- Single view to edit the template
- Single dialog to interact with the environment (hardware, disk, Kernel Wizard)

The button "HW & Kernel" in the ribbon can be used to display the dialog. The button "Template" in the dialog is used to symbolize the template content. The window will resemble the following:



```
# TEMPLATE - SINGLE-CHANNEL - LINEAR
#Step 1: Fill in following input parameters.
#Step 2: use the menu "Save As... (OEMPA

[Specimen]
#"Velocity" is the velocity in the specimen (
Velocity=6300.0 m/s
#"Wave" is the wave ('L' for Longitudinal an
Wave=L

[Wedge]
#"Enable" should be used if the wedge is u
Enable=0
#"Velocity" is the velocity in the wedge (uni
Velocity=1480.0 m/s
#"Height" is the probe height over the spe
Height=0.0 mm
#"Angle" is the probe angle over the speci
Angle=0.0 deg

[Probe]
#"ElementCount" is the element count (line
ElementCount=64
#"Pitch" is the element pitch.
Pitch=1.0 mm
```

In the "HW & Kernel" dialog, you can change the acquisition channel mode to either "Single Channel" or "Multiple Channel". Be careful: **Kernel file formats for single channel and multiple channel are not compatible** with each other.

2.3 Template view

This view can be used to edit the template. The template is a text file saved on the disk. It is also possible to edit it with notepad. Two different files are used, resembling the following (in the case of Windows 7):

- For single channel: "C:\ProgramData\AOS\OEMPA [version]\Files\Kernel\OEMPAWizardExampleSingleChannel.txt"

- For multiple channels: “C:\ProgramData\AOS\OEMPA [version]\Files\Kernel\OEMPAWizardExampleMultipleChannel.txt”

In those files, lines that begin with “#” are comments.

2.3.1 Single channel

The template has different sections, with the following input parameters:

- Specimen: this is a plate specimen
  - Velocity
  - Wave (longitudinal or transversal)
- Wedge
  - Enable/disable
  - Velocity
  - Height (distance between the middle of the probe and the specimen)
  - Angle (only one angle is available in the template)
- Probe: this is a linear 1D plane phased array
  - Element count
  - Element pitch
  - Frequency
- Scan: this is a linear scan
  - Aperture element count
  - Emission depth
  - Reception depth (dynamic depth focusing is possible)
  - Attack angle (beam angle in the specimen)
  - Element start, stop and step of the aperture first element
- Ascan: start, range and timeslot duration
- Cscan: definition of each gate

If these input parameters are insufficient, then access of all parameters can be gained through use of the “Toolbox” (same as editing the multiple-acquisition channel wizard).

2.3.2 Multiple channel

There are many input parameters, so the template is useful only to show how to interface the high level API (Kernel Wizard). Here are parameters in the template:

- The system (section “WizardSystem”)
  - The probe count used to inspect the specimen.
  - Ability to enable wedge for each probe.
- The channels (section “Channels”, one for each device)

- Probe count on the connector of your device.
  - This value is smaller or equal to the probe count in the section “WizardSystem”.
  - If you have only one device, this value is equal to the probe count in the section “WizardSystem”.
- Scan count for each probe on the device.
  - Single or multiple scan for each probe.

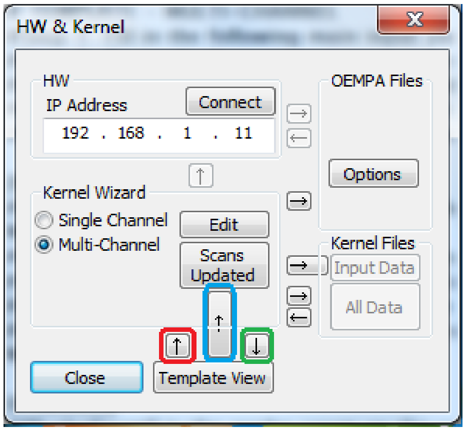
Here are two examples of multiple channels with two scans for each (scan count of section “Channels”):

- One linear scanning and one sectorial scanning for the same probe.
- One linear scanning for the first probe and one sectorial scanning for the second probe (on the same connector of the same device).

To fill many input parameters (specimen, probe), you should use the “Toolbox”. The “Toolbox” can be used to edit all input parameters, so the template is not required. The template is only an example for demonstrating the use of the high level API (Kernel Wizard).

2.4 Dialog

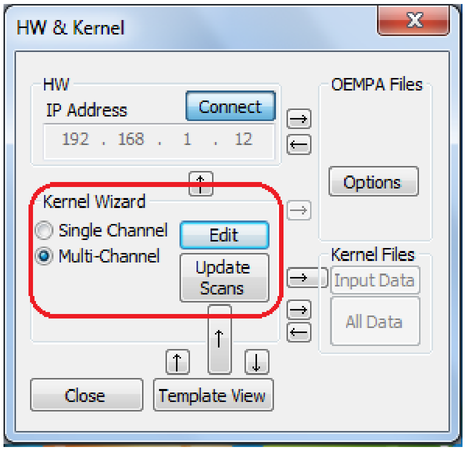
The dialog is the following:



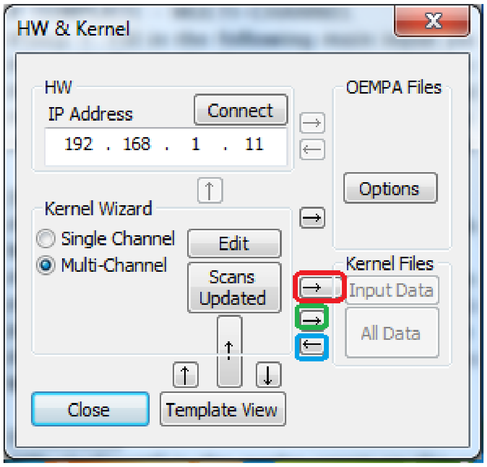
IP Address 192 . 168 . 1 . 11

○ Single Channel
● Multi-Channel

- The red button: to save the template and update the wizard with its parameters.
- The blue button: same as the red after scans are updated (output parameters of the wizard are also updated).
- The green button: to read the wizard and update the template (the view is also updated).

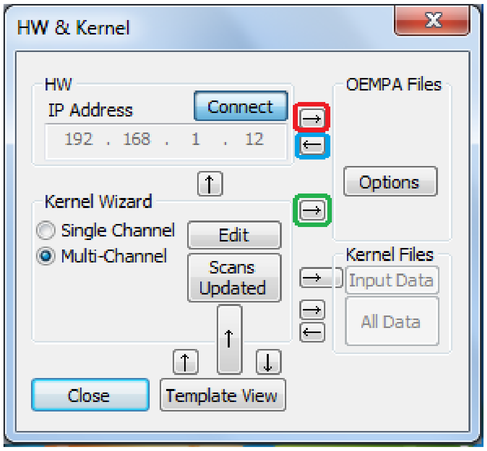


- "**Single Channel**": to configure the wizard as a single acquisition channel.
- "**Multi-Channel**": to configure the wizard as multiple acquisition channels.
- "**Edit**" button: to run the "Toolbox" and edit all available parameters.
- "**Update Scans**" button: to update output parameters of the wizard. Once it has been updated, the button is named "**Scans Updated**".

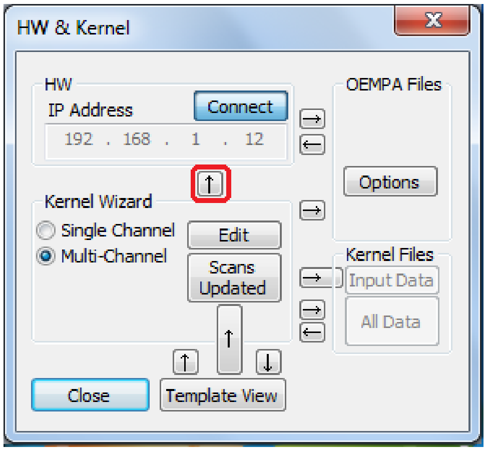


- The red button: to save only input parameters on the disk.
- The blue button: to load kernel file from the disk, this file has been saved previously with the red button or the green button.
- The green button: to save input and outputs parameters on the disk.

And once you are connected with the device:



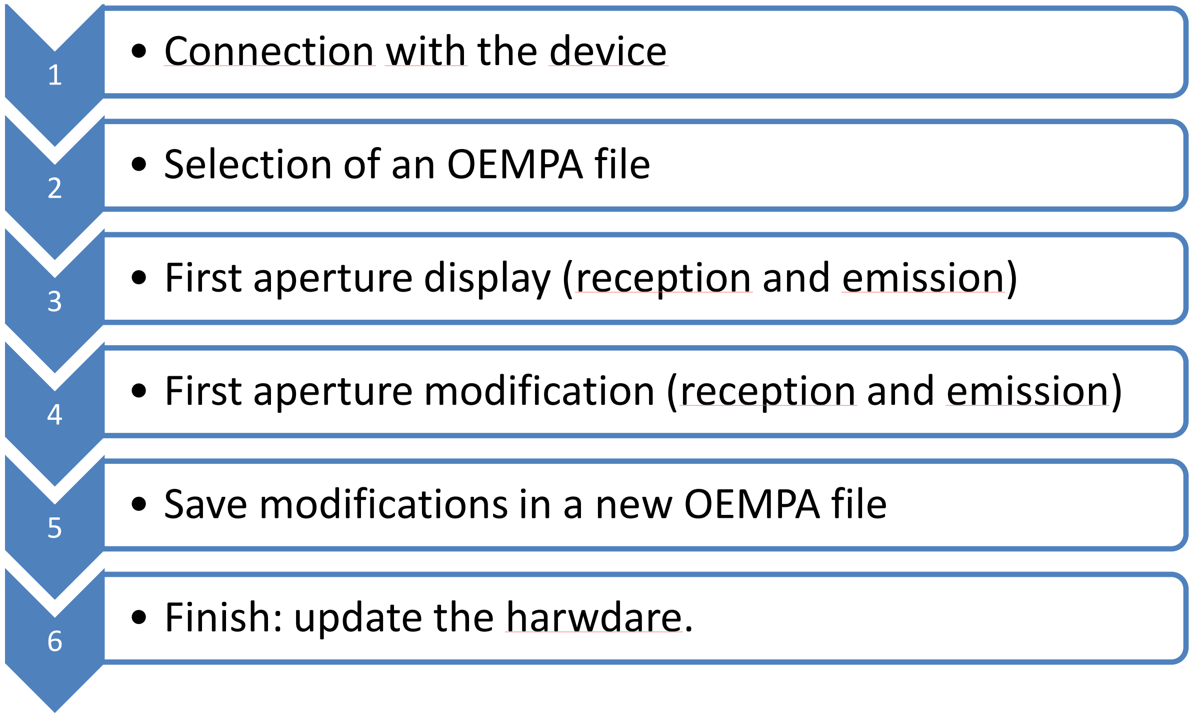
- The red button: to read-back the device and save FW parameters on the disk (OEMPA file format).
- The blue button: to update the device from OEMPA configuration file.
- The green button: to save Wizard output parameters on the disk (OEMPA file format).



- The red button: to update the HW with the Wizard output parameters
  - No file is used. It is quicker than using an OEMPA file.

3. Example using Medium Level API (OEMPACustomizedExample)

“OEMPACustomizedExample” shows the step-by-step process for loading an OEMPA file. The OEMPA file is not loaded immediately, however, so it is possible for the user to change some of the loaded configuration data in computer memory before updating the hardware. Here are the steps:



1 • Connection with the device

2 • Selection of an OEMPA file

3 • First aperture display (reception and emission)

4 • First aperture modification (reception and emission)

5 • Save modifications in a new OEMPA file

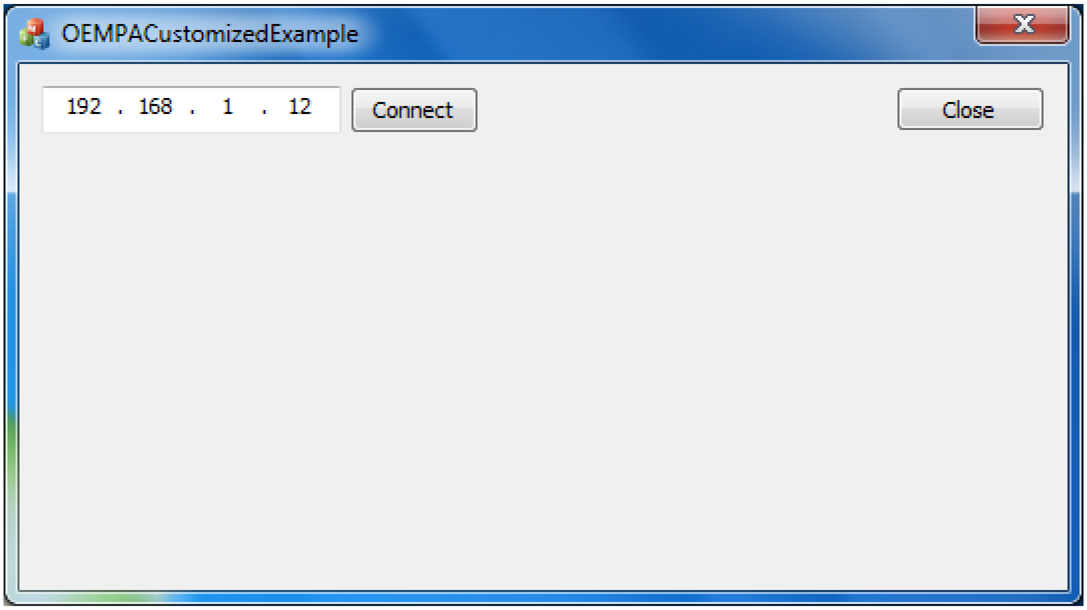
6 • Finish: update the harwdare.

The application focuses only on the first aperture, however, the process is the same for any other aperture. The current software example can only edit the following data before updating the HW:

- Element list (emission/reception) but element count cannot be modified
- Reception gain apodization (useful for normalization)
- Reception beam correction (useful for calibration)
- Emission pulse width

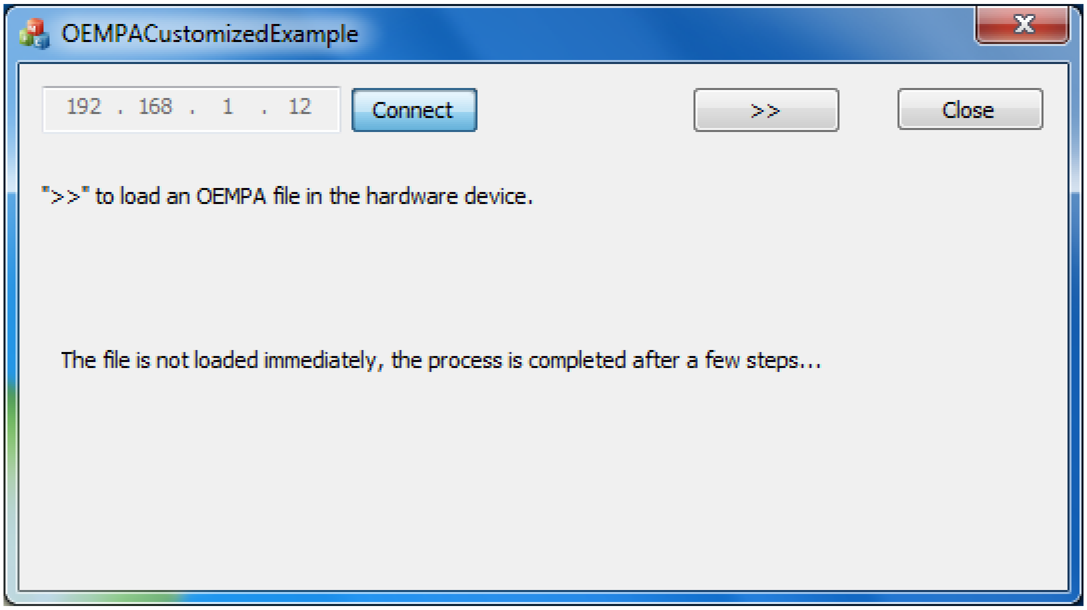
3.1 Connection

The connection process uses the low level API, also used in “OEMPAApplicationExample” (Section 4.1, C++ Example). Enter the IP address of and press connect.



3.2 File selection

The file loading process is the same as “OEMPAApplicationExample”.

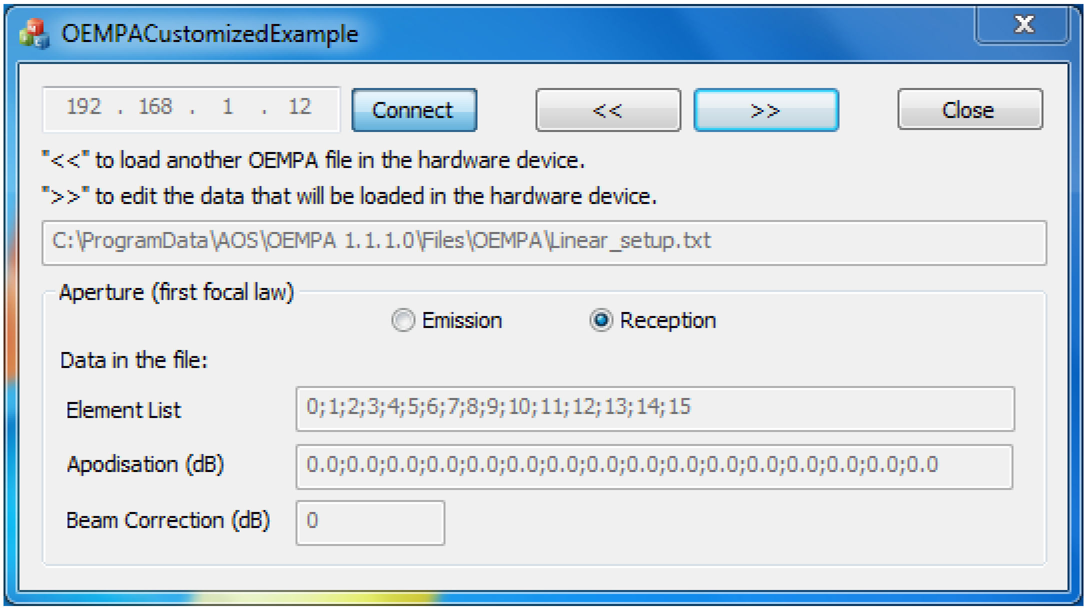


">>" to load an OEMPA file in the hardware device.

The file is not loaded immediately, the process is completed after a few steps...

3.3 Aperture display

Only the first aperture is displayed for emission and reception.



192 . 168 . 1 . 12

"<<" to load another OEMPA file in the hardware device.

">>" to edit the data that will be loaded in the hardware device.

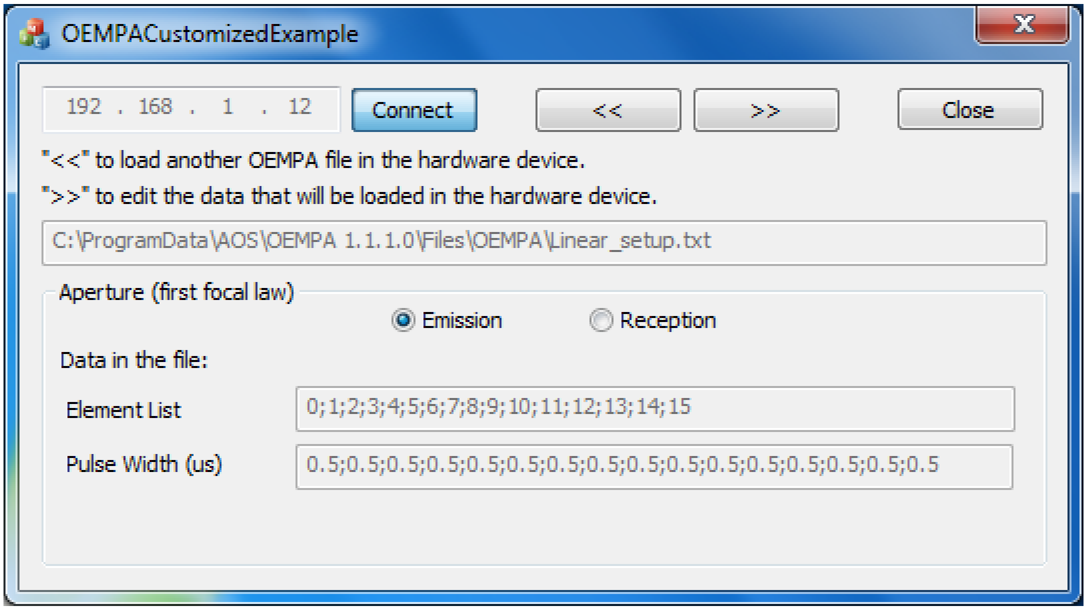
C:\ProgramData\AOS\OEMPA 1.1.1.0\Files\OEMPA\Linear_setup.txt

Aperture (first focal law)

○ Emission ● Reception

Data in the file:

Element List	0;1;2;3;4;5;6;7;8;9;10;11;12;13;14;15
Apodisation (dB)	0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
Beam Correction (dB)	0



192 . 168 . 1 . 12

"<<" to load another OEMPA file in the hardware device.

">>" to edit the data that will be loaded in the hardware device.

C:\ProgramData\AOS\OEMPA 1.1.1.0\Files\OEMPA\Linear_setup.txt

Aperture (first focal law)

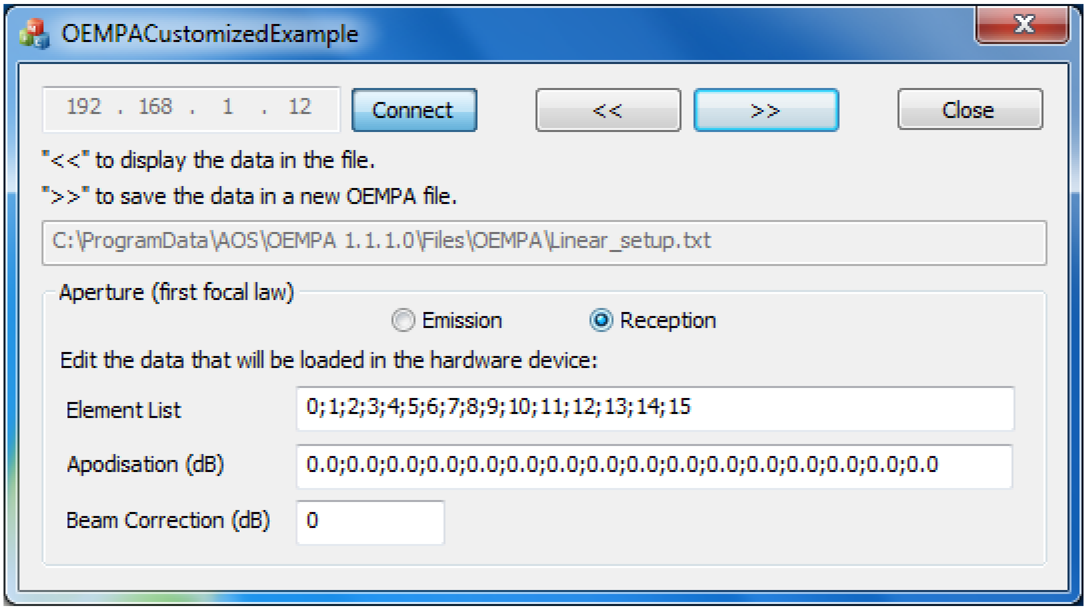
● Emission ○ Reception

Data in the file:

Element List	0;1;2;3;4;5;6;7;8;9;10;11;12;13;14;15
Pulse Width (us)	0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5

3.4 Aperture addition

You can edit any parameter, but you cannot change the element count.



192 . 168 . 1 . 12

"<<" to display the data in the file.

">>" to save the data in a new OEMPA file.

C:\ProgramData\AOS\OEMPA 1.1.1.0\Files\OEMPA\Linear_setup.txt

Aperture (first focal law)

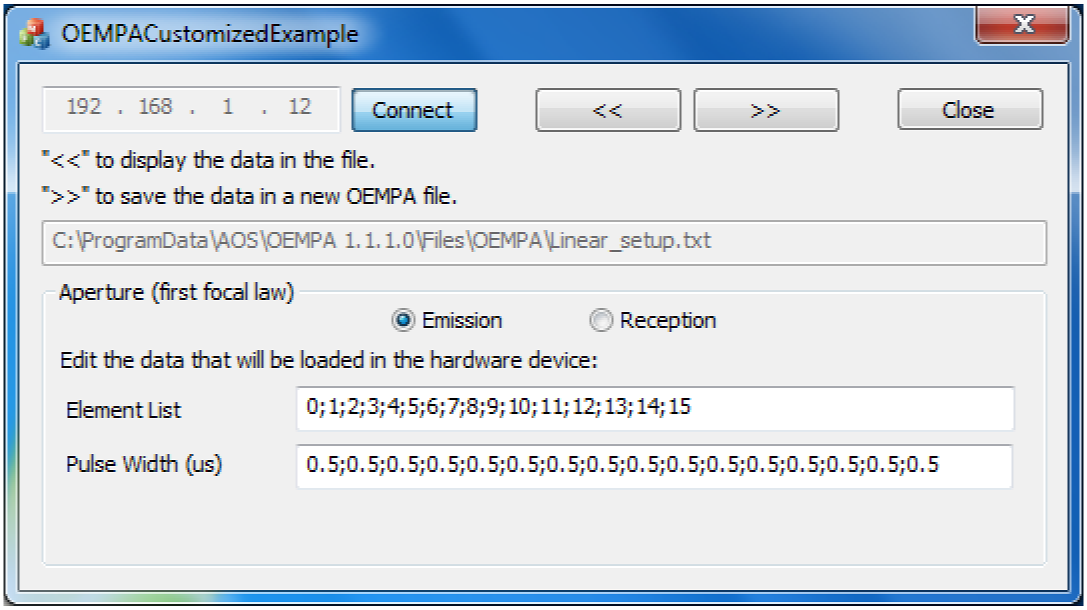
○ Emission ● Reception

Edit the data that will be loaded in the hardware device:

Element List: 0;1;2;3;4;5;6;7;8;9;10;11;12;13;14;15

Apodisation (dB): 0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0

Beam Correction (dB): 0



192 . 168 . 1 . 12

"<<" to display the data in the file.

">>" to save the data in a new OEMPA file.

C:\ProgramData\AOS\OEMPA 1.1.1.0\Files\OEMPA\Linear_setup.txt

Aperture (first focal law)

● Emission ○ Reception

Edit the data that will be loaded in the hardware device:

Element List: 0;1;2;3;4;5;6;7;8;9;10;11;12;13;14;15

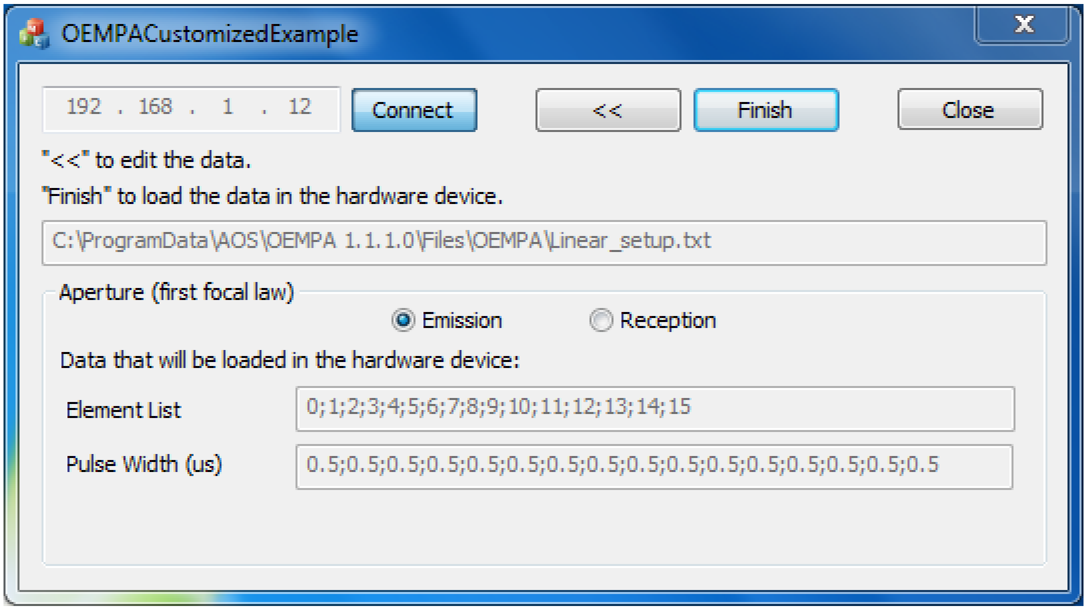
Pulse Width (us): 0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5

3.5 Save modifications

You can save modifications in a new OEMPA file, and you can edit and compare this file with the original one.

3.6 Update HW

Press “Finish” to update the HW.



"<<" to edit the data.

"Finish" to load the data in the hardware device.

C:\ProgramData\AOS\OEMPA 1.1.1.0\Files\OEMPA\Linear_setup.txt

Aperture (first focal law)

● Emission ○ Reception

Data that will be loaded in the hardware device:

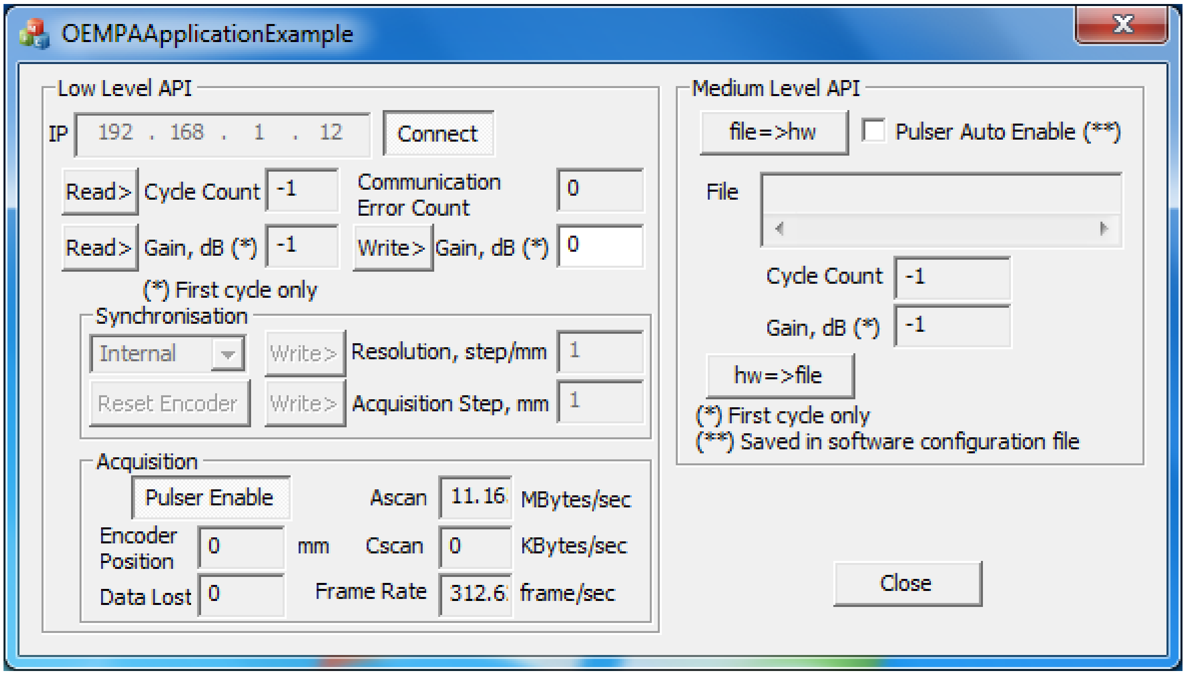
Element List: 0;1;2;3;4;5;6;7;8;9;10;11;12;13;14;15

Pulse Width (us): 0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5;0.5

4. Example using Low Level API (OEMPAApplicationExample)

4.1 C++ Example

This application, named “OEMPAApplicationExample“, is the basic example to demonstrate proper use of the driver. This application is also an example for using the basic features of the customized API.



IP 192 . 168 . 1 . 12

Cycle Count -1

Communication Error Count 0

Gain, dB (*) -1

Gain, dB (*) 0

(*) First cycle only

Internal

Resolution, step/mm 1

Acquisition Step, mm 1

Ascan 11.16 MBytes/sec

Encoder Position 0 mm

Cscan 0 KBytes/sec

Data Lost 0

Frame Rate 312.6 frame/sec

☐ Pulser Auto Enable (**)

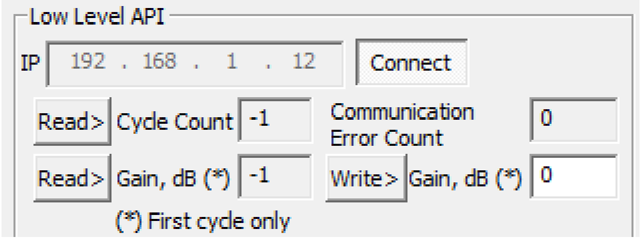
File

Cycle Count -1

Gain, dB (*) -1

(*) First cycle only
(**) Saved in software configuration file

Further explanation about each control of this dialog:



IP 192 . 168 . 1 . 12

Cycle Count -1

Communication Error Count 0

Gain, dB (*) -1

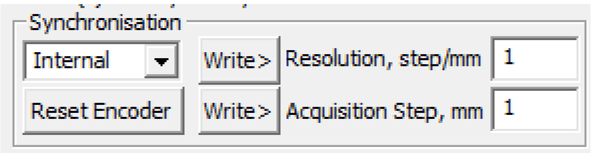
Gain, dB (*) 0

(*) First cycle only

- “IP”: IP address of your device.
- “Connect”: to connect/disconnect with the device.
- “Write>”: to write the gain (dB) of the first cycle.
- “Read>”: to read back gain and cycle count from the device.
- “Communication Error Count”: to display any communication error.

If an IO board is integrated into your device, you can access the synchronization group:

- "Internal": pulser trigger is generated by an internal timer. The device returns no encoder information to the computer.
- "Encoder": each time the encoder has been increased by an "Acquisition Step", the pulser trigger is generated. Encoder information is returned from the device to the computer.
- "Mixed": pulser trigger is generated by an internal timer and encoder information is also returned from the device to the computer.

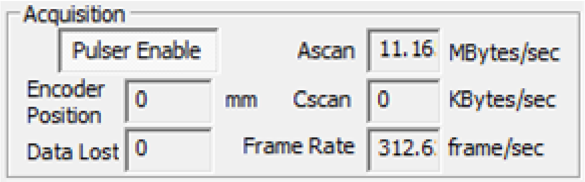


Internal

Resolution, step/mm 1

Acquisition Step, mm 1

- Combo box: you can select "Internal" "Encoder" or "Mixed" synchronization. See paragraph "Trigger" in in *Software_API.pdf* and "Trigger Modes" in *Hardware_Introduction.pdf*.
- "Write>": you can change the encoder resolution and the acquisition step (unit is millimeter).
- "Reset Encoder": press this button to reset the encoder.



Ascan 11.16 MBytes/sec

Encoder Position 0 mm

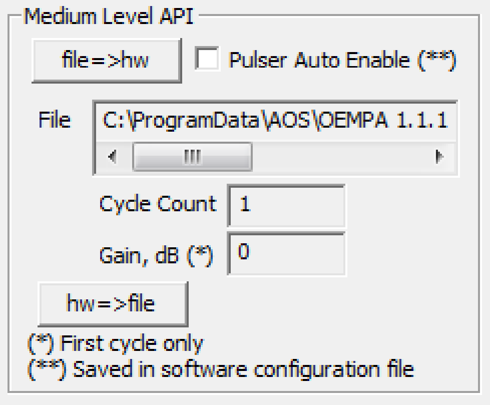
Cscan 0 KBytes/sec

Data Lost 0

Frame Rate 312.6. frame/sec

- "Pulser Enable": to enable pulser (you need to load an OEMPA setup file before enabling)
- "Data Lost": when acquisition throughput is too high some acquisition data could be lost.
- "Encoder Position": current position of the encoder.

The following section describes basic features of the medium level API. Please notice that "File", "Gain", and "Cycle Count" are collected from the loading callback function, they are not read directly from the file:



☐ Pulser Auto Enable (**)

File: C:\ProgramData\AOS\OEMPA 1.1.1

Cycle Count: 1

Gain, dB (*): 0

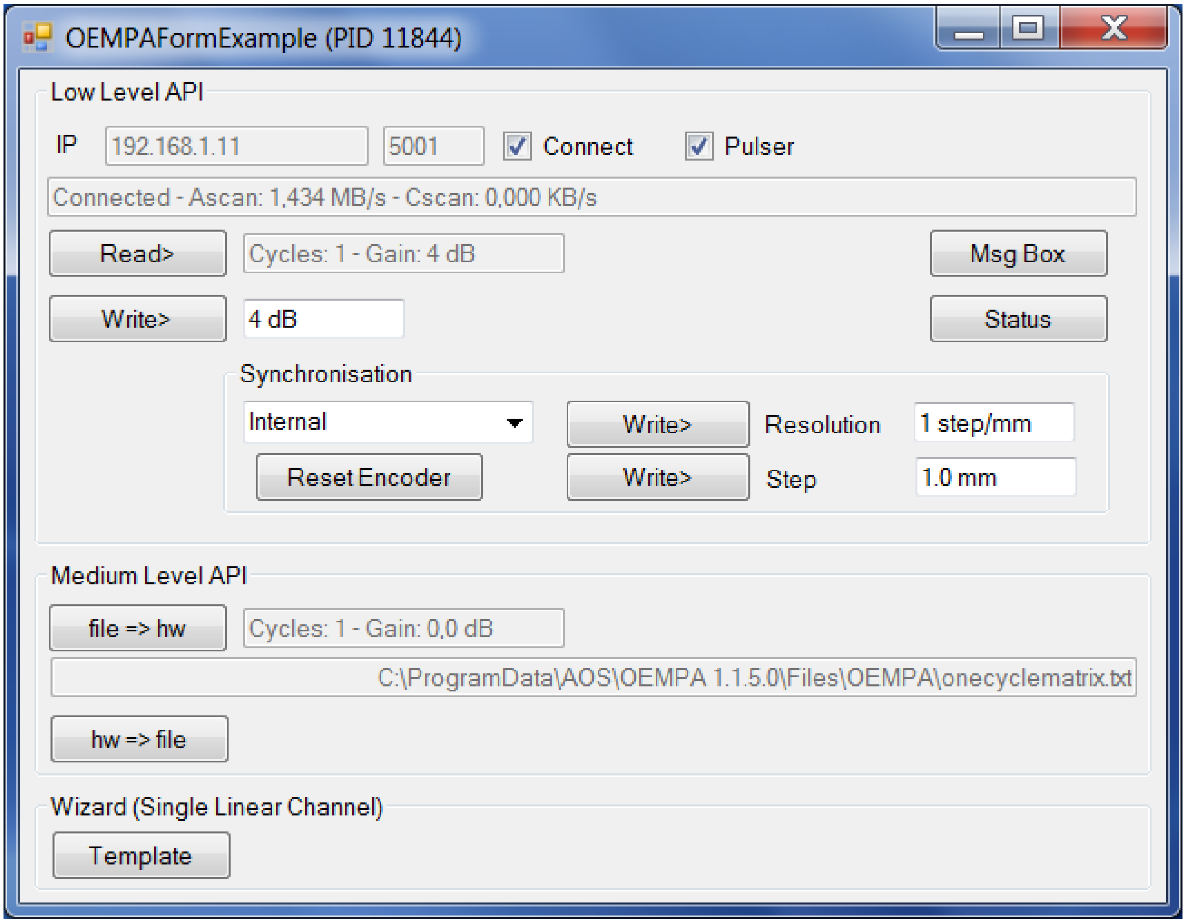
(*) First cycle only
(**) Saved in software configuration file

- “file=>hw”: to load an OEMPA setup file.
- “Pulser Auto Enable”: see in the driver documentation function “CSWDevice::SetUnlockDefaultEnablePulser”.
- “File”: display the file name path of the last file that was loaded.
- “Gain”, “Cycle Count”: gain and cycle count in the last loaded file.
- “hw=>file”: to read back the device and save configuration data in an new OEMPA setup file.

Notice: after the first connection, the driver automatically loads the default setup. In this case “File”, “Gain”, and “Cycle Count” are updated from this default file. Some options can be used to manage the behavior of the connection process. Please contact AOS for more information.

4.2 C# Example

This application, named “OEMPAFormExample“, is the basic example to demonstrate proper use of C# driver. This application is also an example for using the basic features of the customized API. All “Gain” is only for the first cycle.



IP 192.168.1.11 5001 ☑ Connect ☑ Pulser

Connected - Ascan: 1,434 MB/s - Cscan: 0,000 KB/s

Cycles: 1 - Gain: 4 dB

4 dB

Internal

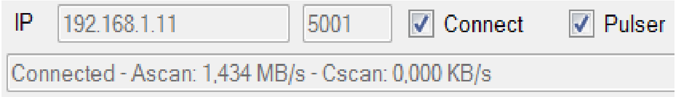
Resolution 1 step/mm

Step 1.0 mm

Cycles: 1 - Gain: 0,0 dB

C:\ProgramData\AOS\OEMPA 1.1.5.0\Files\OEMPA\onecyclematrix.txt

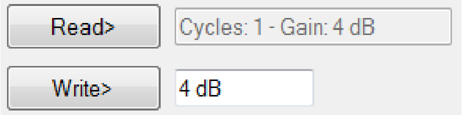
Further explanation about each control of this dialog:



IP 192.168.1.11 5001 ☑ Connect ☑ Pulser

Connected - Ascan: 1,434 MB/s - Cscan: 0,000 KB/s

- “IP”: IP address of your device.
- “Connect”: to connect/disconnect with the device.
- “Pulser”: to enable pulser (you need to load an OEMPA setup file before enabling).
- Display:
    - Connection state
    - Encoder value (if IO board available and if synchronization is different than internal)
    - A-scan throughput.
    - Communication error count.
    - Acquisition data lost.



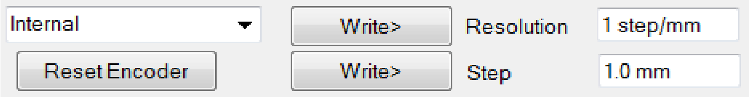
Cycles: 1 - Gain: 4 dB

4 dB

- “Read>”: to read back gain and cycle count from the device.
- “Write>”: to write the gain (dB) of the first cycle.

If an IO board is integrated in your device, you can access the synchronization group:

- “Internal”: pulser trigger is generated by an internal timer. The device returns no encoder information to the computer.
- “Encoder”: each time the encoder has been increased of the “Acquisition Step” the pulser trigger is generated. Encoder information is returned from the device to the computer.
- “Mixed”: pulser trigger is generated by an internal timer and encoder information is also returned from the device to the computer.



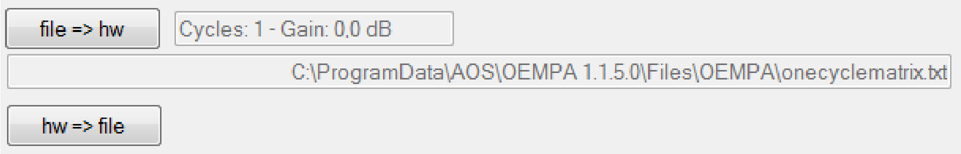
Internal

Resolution 1 step/mm

Step 1.0 mm

- Combo box: you can select “Internal” “Encoder”, “Mixed”, or “ExternalSequence” synchronization. See the paragraph 4.4.4 “Trigger” in *Software_API.pdf* and paragraph 5.1.4 “Trigger Modes” in *Hardware_Introduction.pdf*.
- “Write>”: you can change the encoder resolution and the acquisition step (unit is millimeter).
- “Reset Encoder”: press this button to reset the encoder.

The following section describes the medium level API basic features. Please notice that “File”, “Gain”, and “Cycle Count” are collected from the loading callback function, not read directly from the file:

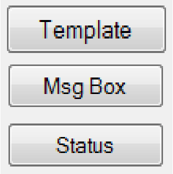


Cycles: 1 - Gain: 0,0 dB

C:\ProgramData\AOS\OEMPA 1.1.5.0\Files\OEMPA\onecyclematrix.txt

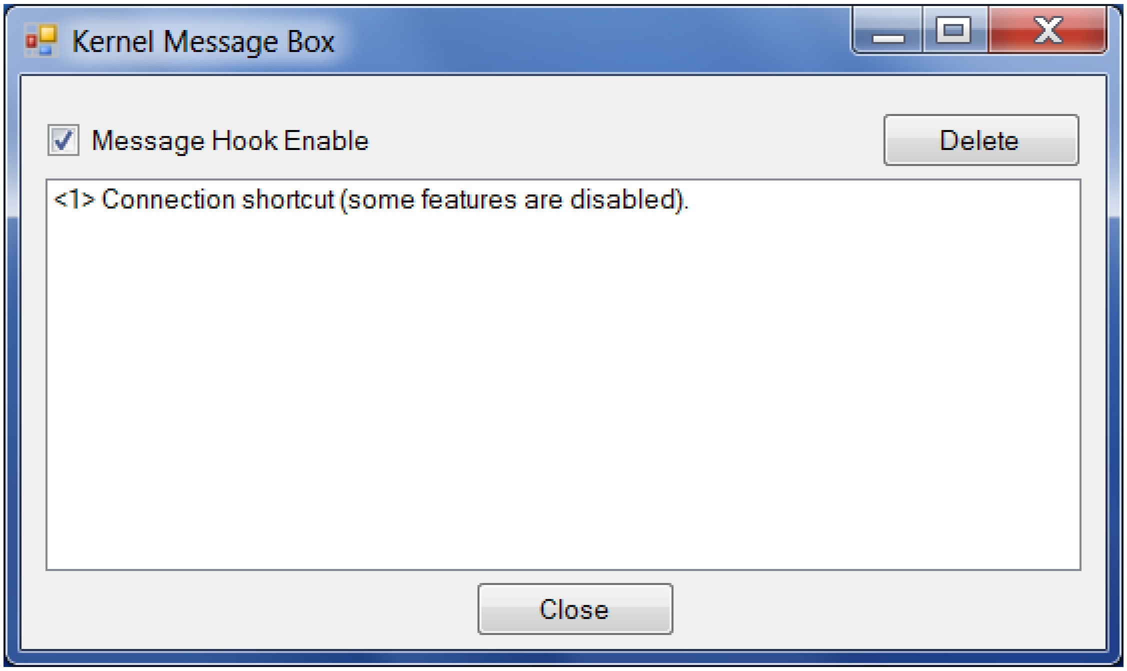
- “file=>hw”: to load an OEMPA setup file.
- Display: cycle count and gain (first cycle) in the last loaded file, and name of that file.
- “hw=>file”: to read back the device and save configuration data in an new OEMPA setup file.

Notice: after the first connection, the driver automatically loads the default setup. In this case “File”, “Gain”, and “Cycle Count” are updated from this default file. Some options can be used to manage the behavior of the connection process. Please contact AOS for more information.



- The button "Template" can be used as an example to show how to call the third level API.
- The button "Msg Box" is an example to catch any popup displayed by the kernel. A new dialog is displayed (see below).
- The button "Status" saves important information in a text file (HW identity, digital inputs and temperatures).

4.2.1 Dialog "Kernel Message Box"



<1> Connection shortcut (some features are disabled).

<u>Note :</u> To catch popups that are normally displayed by the kernel, you have to check the checkbox "Message Hook Enable". The "Delete" button can be used to reset content of the listbox.