TWM – TracePQM Wattmeter

V1.0.5.0, 2017-12-08

TWM is open source project that is being developed for transparent traceable measurement of electric power and power quality parameters. It is being developed in scope of EMPIR project TracePQM [1].

It consists of two parts: (i) LabVIEW control module and GUI and (ii) Octave/Matlab calculation module. Control module (i) handles the HW, i.e. records the waveforms, selects correction files and initiates calculation of the parameters using control module. Calculation module (ii) is based in the QWTB toolbox [2] and it runs in either Matlab or GNU Octave. Both parts are integrated together.

File formats

For documentation of the file formats generated by the TWM, see attached files in project 'doc' folder.

Supported HW

TWM currently integrates drivers for control of: (i) PXI 5922 digitizer using National Instrument's niScope and niTClk drivers (must be installed on the computer); (ii) Sampling multimeters 3458A (NI's VISA drivers must be installed); (iii) DirectSound Drivers for handling soundcard as an ADC [3]; (iv) Simulated ADC which can generate composite harmonics signal. Note the DirectSound drivers (iii) are implemented only for demonstration purposes so the TWM can be run even without any external HW. It may be usable for low accuracy measurements, however it is not official part of the TracePQM project.

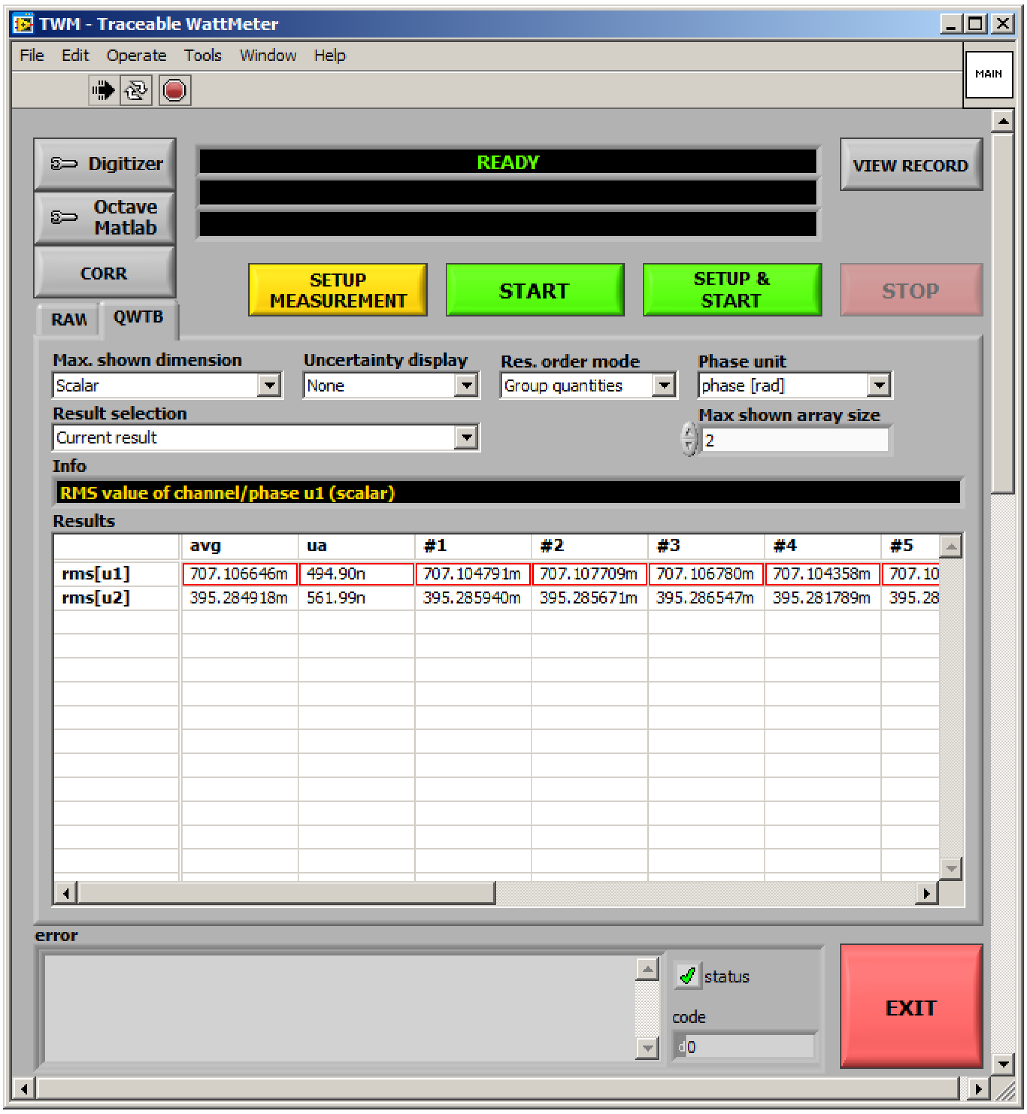
GNU Octave/Matlab interface

Goal of the project TracePQM is to implement interface between TWM and Matlab/GNU Octave so the calculations can run transparently in the M-functions. In the current version both GNU Octave and Matlab are supported via interface GOLPI library [4]. However, not all m-functions are compatible with Matlab yet! Current version of the interface was tested with GNU Octave 4.0.0 and later however it should be possible to run it at least from version 3.6.4. It now works also with Matlab at least from version 2008b.

Brief TWM documentation

The TWM is still in the development and all its functionality is subject to change. At the end of the project, the TWM will be documented with all the details. Following text describes only very basic usage of the current version.

Main panel of the TWM is shown in the Fig. 1. It contains several configuration buttons for configuration of the TWM components which will be described later. The button ‘VIEW RECORD’ can be used to view the last recorded waveform(s). The button ‘STOP’ is ungrayed whenever the TWM is busy and can be used to terminated the pending operation(s). Note it may take some time to respond on the STOP signal. Button ‘CORR’ will invoke panel for selecting the correction files. Note the corrections are not fully implemented in the current version, so the panel won’t be described.



Max. shown dimension: Scalar

Uncertainty display: None

Res. order mode: Group quantities

Phase unit: phase [rad]

Result selection: Current result

Max shown array size: 2

Info: RMS value of channel/phase u1 (scalar)

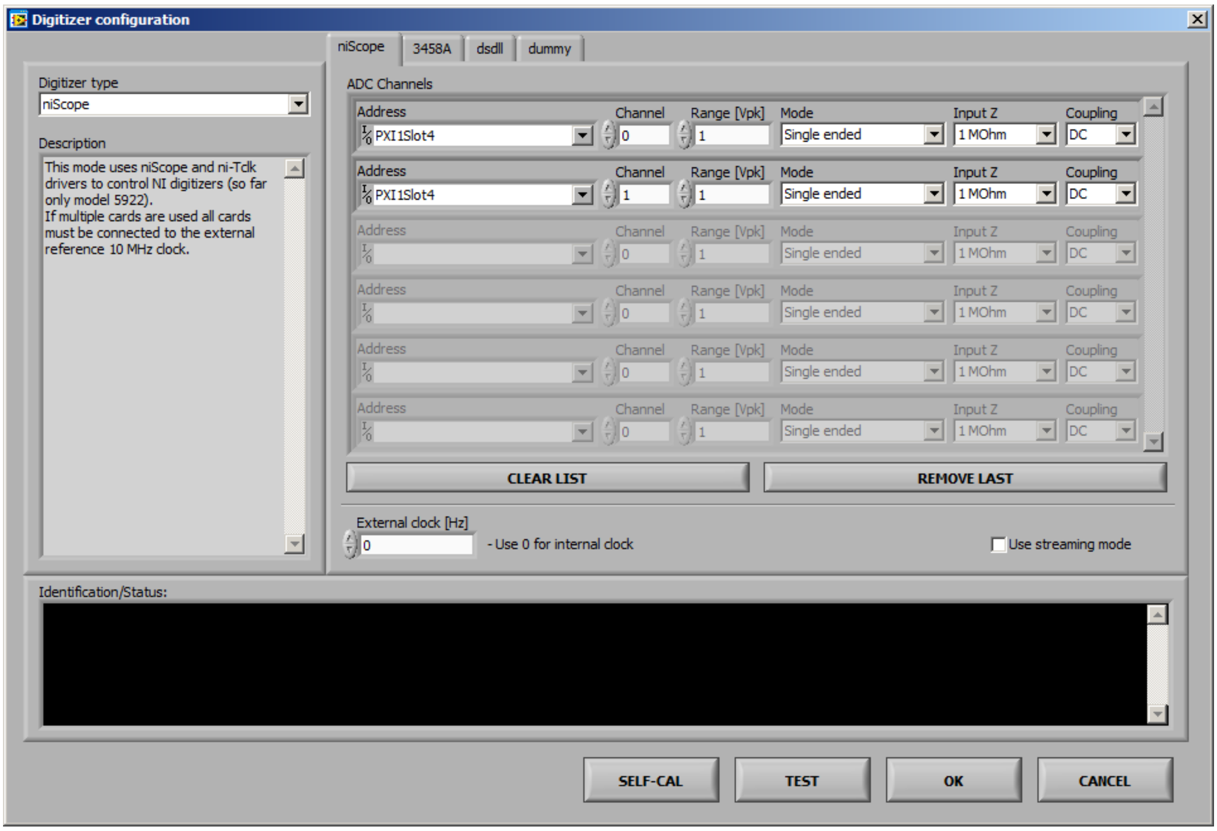
Results

	avg	ua	#1	#2	#3	#4	#5
rms[u1]	707.106646m	494.90n	707.104791m	707.107709m	707.106780m	707.104358m	707.10
rms[u2]	395.284918m	561.99n	395.285940m	395.285671m	395.286547m	395.281789m	395.28

code: 0

Fig. 1 - TWM's main panel

First, the digitizer HW must be configured. The button ‘Digitizer’ must be pressed in order to invoke Digitizer configuration panel. Example of the panels for niScope and 3458A digitizers are shown in the Fig. 2 and Fig. 3. The panel contains digitizer type selector which is used to select the HW to be used for the digitizing. Next, the virtual channels list must be filled in. The drives of 5922 (niScope) and 3458A are designed so the user can enter any count of channels. For 3458A, user must also select synchronization mode and sampling mode of the multimeters. Information panel on the left shows description of the selected mode. If the selected mode requires additional HW such as arbitrary waveform generator (AWG) or a counter the corresponding entries will be ungrayed and must be filled. Note both 5922 and 3458A have option ‘streaming’ which is the mode in which the sample data are streamed directly to the file and the internal memory of the digitizer is not used. This mode will extend the maximum number of samples to record. Each digitizer has 'TEST' function available. The 'TEST' button will initialize the HW and detect eventual errors which will be shown in the black box. If the test is successful, the black box will contain identifiers of the HW. If the digitizer support self-calibration procedure, the button 'SELF-CAL' will be ungrayed.



Digitizer configuration

Digitizer type: niScope

Description:
This mode uses niScope and ni-Tclk drivers to control NI digitizers (so far only model 5922).
If multiple cards are used all cards must be connected to the external reference 10 MHz clock.

ADC Channels

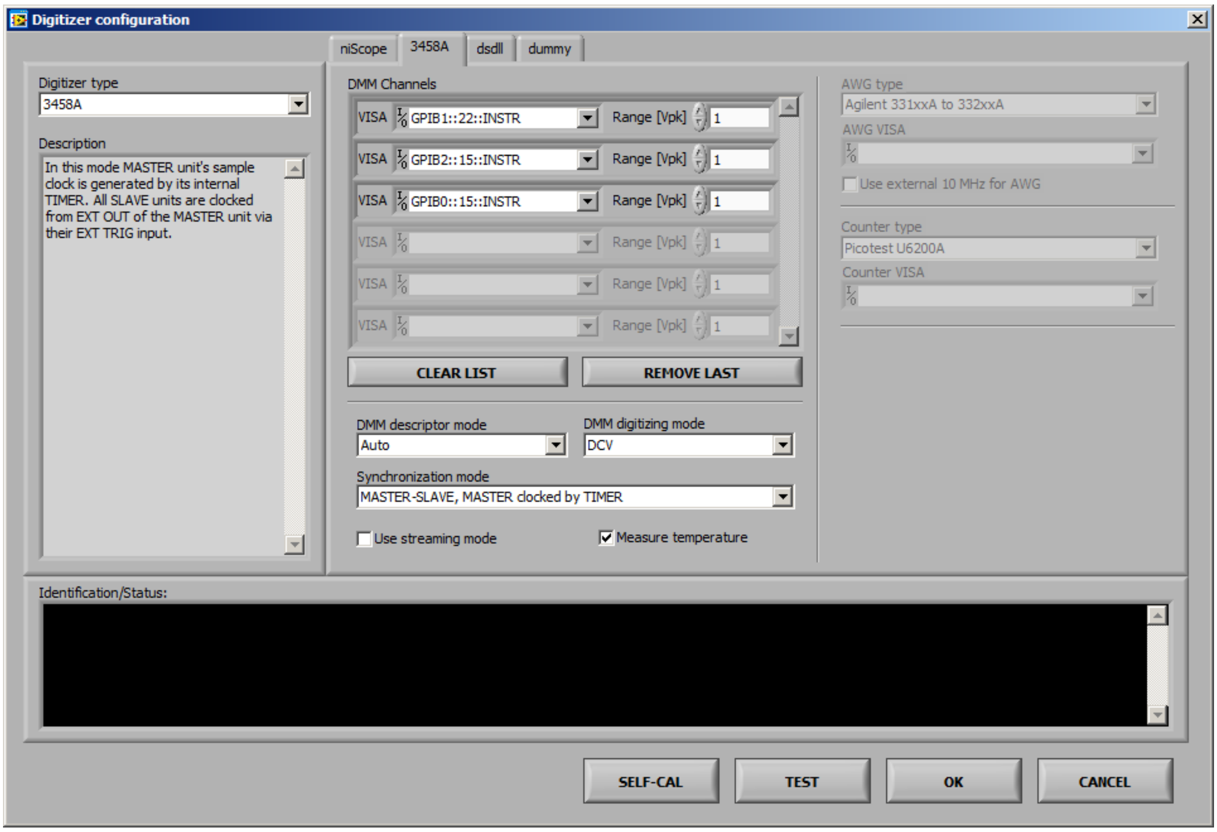
Address	Channel	Range [Vpk]	Mode	Input Z	Coupling
PXI1Slot4	0	1	Single ended	1 MOhm	DC
PXI1Slot4	1	1	Single ended	1 MOhm	DC
	0	1	Single ended	1 MOhm	DC
	0	1	Single ended	1 MOhm	DC
	0	1	Single ended	1 MOhm	DC
	0	1	Single ended	1 MOhm	DC

External clock [Hz]: 0 - Use 0 for internal clock

☐ Use streaming mode

Identification/Status:

Fig. 2 - niScope configuration panel.



Digitizer type: 3458A

Description: In this mode MASTER unit's sample clock is generated by its internal TIMER. All SLAVE units are clocked from EXT OUT of the MASTER unit via their EXT TRIG input.

DMM Channels

VISA GPIB1::22::INSTR Range [Vpk] 1

VISA GPIB2::15::INSTR Range [Vpk] 1

VISA GPIB0::15::INSTR Range [Vpk] 1

VISA Range [Vpk] 1

VISA Range [Vpk] 1

VISA Range [Vpk] 1

DMM descriptor mode: Auto

DMM digitizing mode: DCV

Synchronization mode: MASTER-SLAVE, MASTER clocked by TIMER

□ Use streaming mode ☑ Measure temperature

AWG type: Agilent 331xxA to 332xxA

AWG VISA:

□ Use external 10 MHz for AWG

Counter type: Picotest U6200A

Counter VISA:

Identification/Status:

Fig. 3 - 3458A multimeters configuration panel.

If user wants to process the digitized data automatically, the GNU Octave/Matlab interface must be configured using button ‘Octave/Matlab’ on the main panel of TWM. The panel shown in the Fig. 4 will be displayed. In this panel, user must enter path to the binary folder of the GNU Octave installation or select Matlab mode. By pressing ‘Restart’ the Octave/Matlab will be started. If successful the detected version will be displayed.

The window contains several options:

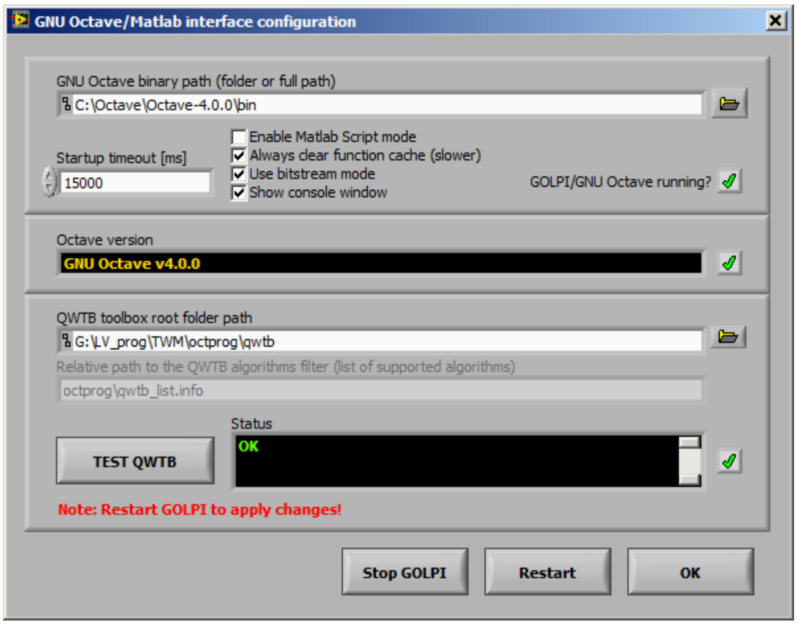
‘**Enable Matlab Script mode**’ – Will execute commands via Matlab instead of GNU Octave

‘**Always clear function cache’** – This is for debugging purposes. It will reload all m-functions before each command. Without this the TWM won’t recognize changes made in the m-files.

‘**Use bitstream mode**’ – This is supported only for GNU Octave and it means the data between Octave and LV will be transferred in a faster way.

‘**Show console window**’ – This is valid only for GNU Octave. It will show debug console where one can inspect traffic between Octave and LV. Note it is read-only console and it may slow down operation of TWM.

When calculation via QWTB toolbox is required, the user must set the path to the QWTB toolbox. The setup of the QWTB is verified by pressing ‘TEST QWTB’. Note the current version of TWM has a local copy of the QWTB with few algorithms placed in the 'octprog'. This is just for development purposes and will be removed in final versions.



GNU Octave/Matlab interface configuration

GNU Octave binary path (folder or full path)
C:\Octave\Octave-4.0.0\bin

Startup timeout [ms]
15000

□ Enable Matlab Script mode
☑ Always clear function cache (slower)
☑ Use bitstream mode
☑ Show console window

GOLPI/GNU Octave running? ✓

Octave version
GNU Octave v4.0.0 ✓

QWTB toolbox root folder path
G:\LV_prog\TWM\octprog\qwtb

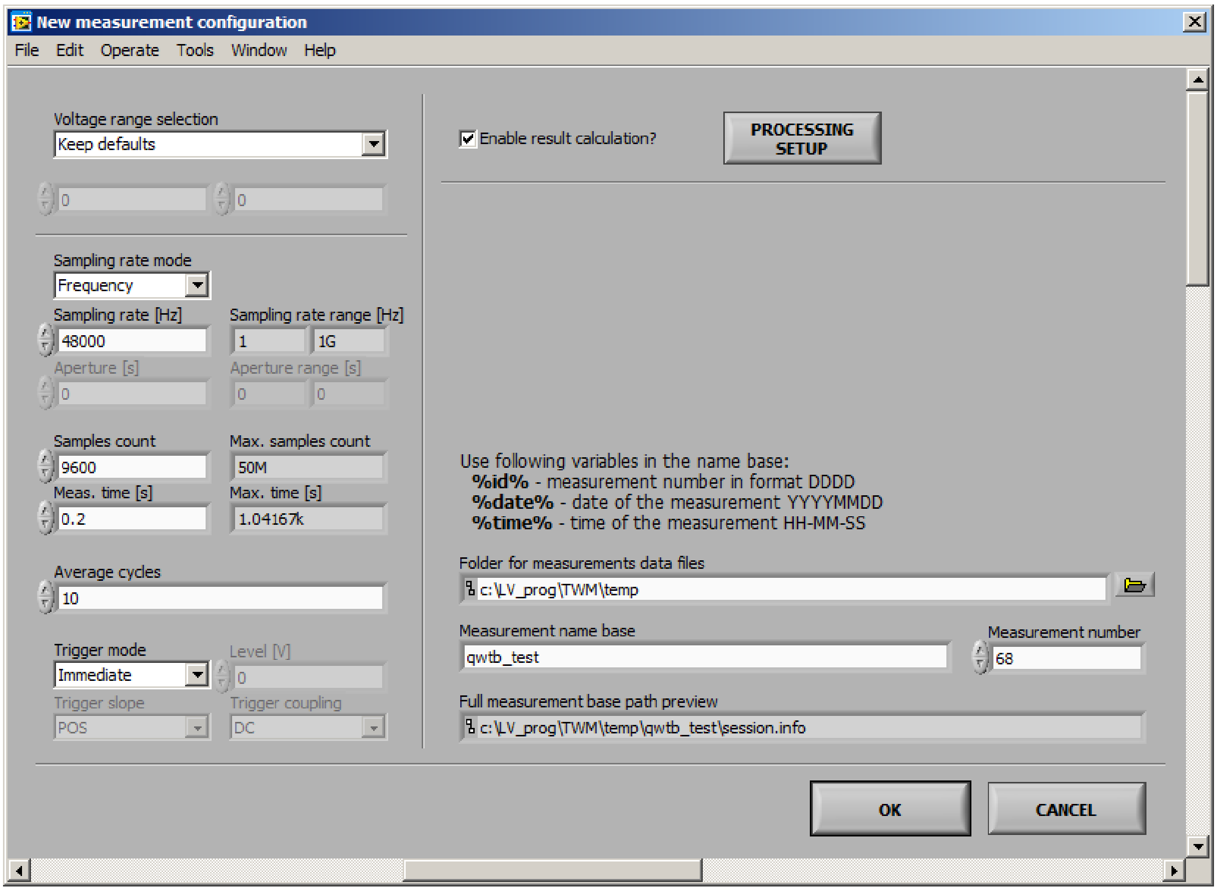
Relative path to the QWTB algorithms filter (list of supported algorithms)
octprog\qwtb_list.info

Status
OK ✓

Note: Restart GOLPI to apply changes!

Fig. 4 - Octave/Matlab configuration panel.

When the basic configurations are done, the new measurement may be configured by pressing 'SETUP MEASUREMENT' or 'SETUP & START'. This will initiate panel show in the Fig. 5. Here, user must setup sampling parameters such as samples count, sampling rate, triggers, etc. User must also setup destination folder for the measured data. User may also use variables in the file name which will be replaced by the automatic text. The actual destination folder with the replacements is always shown.



Voltage range selection: Keep defaults

0 0

Sampling rate mode: Frequency

Sampling rate [Hz]: 48000

Sampling rate range [Hz]: 1 1G

Aperture [s]: 0

Aperture range [s]: 0 0

Samples count: 9600

Max. samples count: 50M

Meas. time [s]: 0.2

Max. time [s]: 1.04167k

Average cycles: 10

Trigger mode: Immediate

Level [V]: 0

Trigger slope: POS

Trigger coupling: DC

☑ Enable result calculation?

Use following variables in the name base:
%id% - measurement number in format DDDD
%date% - date of the measurement YYYYMMDD
%time% - time of the measurement HH-MM-SS

Folder for measurements data files: c:\LV_prog\TWM\temp

Measurement name base: qwtb_test

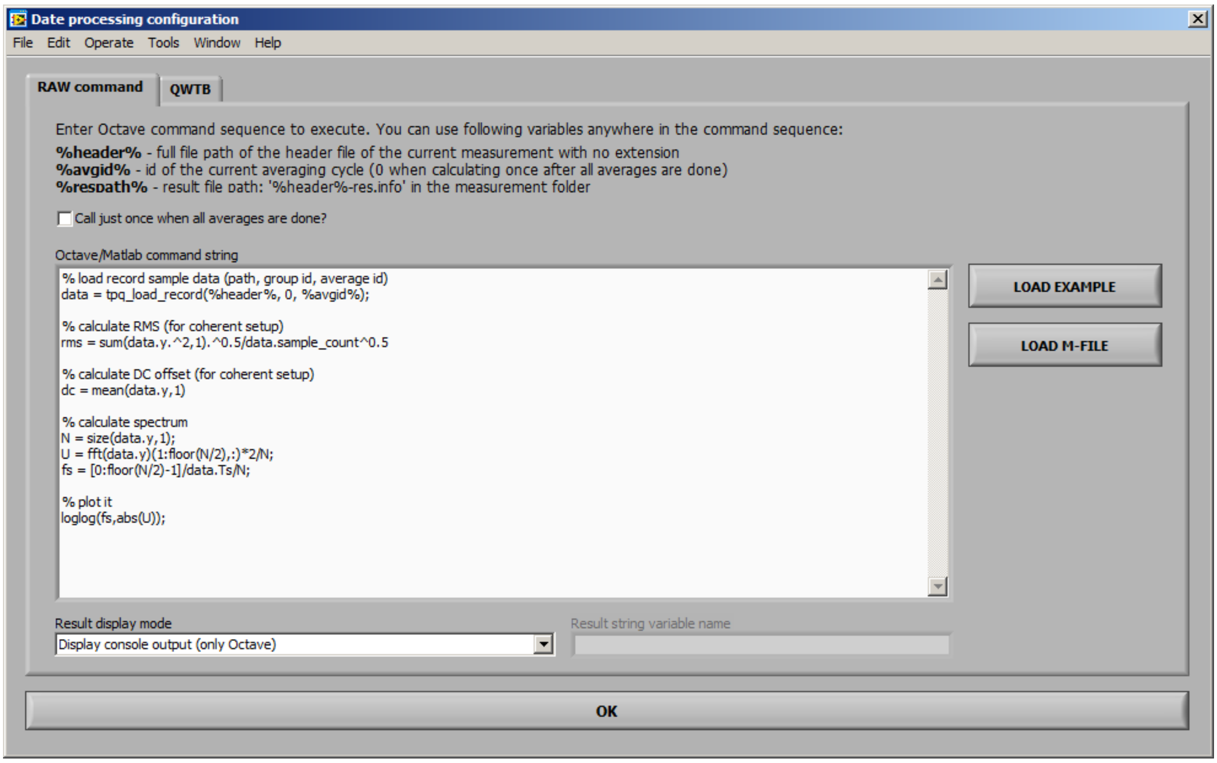
Measurement number: 68

Full measurement base path preview: c:\LV_prog\TWM\temp\qwtb_test\session.info

Fig. 5 - New measurement configuration.

If the calculation should be performed automatically on the recorded data, the user must also configure the processing using by pressing ‘PROCESSING SETUP’ and the checkbox ‘Enable result calculation?’ must be checked. The button ‘PROCESSING SETUP’ will invoke panel show in the Fig. 6 or Fig. 7. By selecting the page ‘RAW command’ a raw command mode is selected. In this mode, the TWM will execute the Octave/Matlab code placed in the text box as it is. Several ways of displaying the results are available.

If ‘QWTB’ page is selected, the TWM will execute the QWTB algorithm on the recorded data. In the QWTB mode, the ‘List of algorithms’ will be filled by the available algorithms. When the algorithm is selected, its information from QWTB toolbox will be displayed on the right. If the algorithm has user definable parameters, such as the one shown in the example in the Fig. 7, the list of the parameters will be shown in the table. User may fill the values. Each algorithm may have uncertainty calculation option, which must be selected if required.


Date processing configuration
File Edit Operate Tools Window Help
RAW command
QWTB
LOAD EXAMPLE
LOAD M-FILE
OK

Enter Octave command sequence to execute. You can use following variables anywhere in the command sequence:

%header% - full file path of the header file of the current measurement with no extension
%avgid% - id of the current averaging cycle (0 when calculating once after all averages are done)
%respath% - result file path: '%header%-res.info' in the measurement folder

□ Call just once when all averages are done?

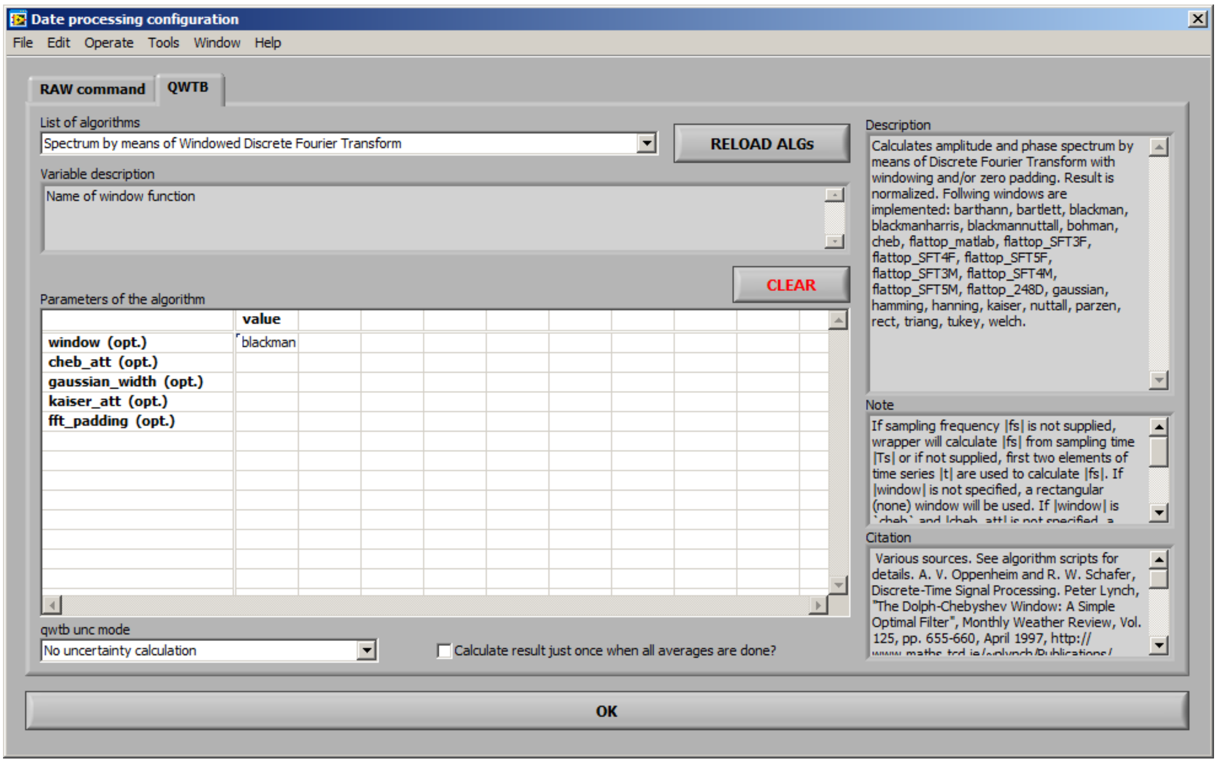
Octave/Matlab command string

```
% load record sample data (path, group id, average id)
data = tpq_load_record(%header%, 0, %avgid%);

% calculate RMS (for coherent setup)
rms = sum(data.y.^2,1).^0.5/data.sample_count^0.5

% calculate DC offset (for coherent setup)
dc = mean(data.y,1)

% calculate spectrum
N = size(data.y,1);
U = fft(data.y)(1:floor(N/2),:)*2/N;
fs = [0:floor(N/2)-1]/data.Ts/N;

% plot it
loglog(fs,abs(U));
```

Result display mode: Display console output (only Octave)

Result string variable name:

Fig. 6 - Configuration of the calculation (RAW commands mode).


Date processing configuration
File Edit Operate Tools Window Help
RAW command
QWTB
RELOAD ALGs
CLEAR
OK

List of algorithms: Spectrum by means of Windowed Discrete Fourier Transform

Variable description: Name of window function

Parameters of the algorithm

	value
window (opt.)	blackman
cheb_att (opt.)	
gaussian_width (opt.)	
kaiser_att (opt.)	
fft_padding (opt.)	

qwtb unc mode: No uncertainty calculation

□ Calculate result just once when all averages are done?

Description

Calculates amplitude and phase spectrum by means of Discrete Fourier Transform with windowing and/or zero padding. Result is normalized. Follwing windows are implemented: barthann, bartlett, blackman, blackmanharris, blackmannuttall, bohman, cheb, flattop_matlab, flattop_SFT3F, flattop_SFT4F, flattop_SFT5F, flattop_SFT3M, flattop_SFT4M, flattop_SFT5M, flattop_248D, gaussian, hamming, hanning, kaiser, nuttall, parzen, rect, triang, tukey, welch.

Note

If sampling frequency |fs| is not supplied, wrapper will calculate |fs| from sampling time |Ts| or if not supplied, first two elements of time series |t| are used to calculate |fs|. If |window| is not specified, a rectangular (none) window will be used. If |window| is `cheb` and |cheb_att| is not specified, a

Citation

Various sources. See algorithm scripts for details. A. V. Oppenheim and R. W. Schafer, Discrete-Time Signal Processing. Peter Lynch, "The Dolph-Chebyshev Window: A Simple Optimal Filter", Monthly Weather Review, Vol. 125, pp. 655-660, April 1997, http://www.maths.tcd.ie/~plynch/Publications/

Fig. 7 - Configuration of the calculation (QWTB toolbox mode).

When the configuration of the measurement is done, user may initiate it by pressing ‘START’ on the main panel. When the record is done and eventual calculation is performed, the results are available on the main panel. In the ‘RAW command’ mode, the result is plain text. In the ‘QWTB’ node, the result is formatted table. User has several options, such as ordering the quantities, selecting the uncertainty display mode, etc. Each row of the table can be displayed as a graph by double clicking the row.

References:

[1] EMPIR project TracePQM,
www: http://tracepqm.cmi.cz/

[2] QWTB Toolbox,
www: https://qwtb.github.io/qwtb/

[3] DirectSound DSDLL library,
www: http://www.elektronika.kvalitne.cz/SW/dsdll/dsdll_eng.html

[4] GOLPI library – Gnu Octave to LabVIEW Pipe Interface,
www: https://github.com/KaeroDot/GOLPI