

**GUIDE FOR SAMPLING POWER AND POWER QUALITY MEASUREMENTS**

|  |
| --- |
|  |

CHAPTER 3

USER’S GUIDE FOR THE OPEN SOFTWARE TOOL

BLANK PAGE

**GLOSSARY**

**PQ** : Power Quality

To be completed…

BLANK PAGE

**OVERVIEW**

This document produces a user’s guide for the open software tool which will help end-users to acquaint themselves with the tool. The software is described in terms of the user interface and the configuration of the digitizers. The guide describes the integration of new digitizers into the system and the format of the calibration datasets for components used in the modular setup. It also provides information about power and PQ tests included in the initial database of calculation algorithms and about the implementation of new algorithms.

BLANK PAGE

**CONTENTS** To be updated !

[A INTRODUCTION 9](#_Toc4431300)

[A.1 TWM manual 9](#_Toc4431301)

[A.1.1 Installation 9](#_Toc4431302)

[A.1.1.a Installing TWM tool 9](#_Toc4431303)

[A.1.1.b Installing TWM prerequisites 10](#_Toc4431304)

[A.1.1.c Installing GNU Octave 11](#_Toc4431305)

[A.1.1.d Installing Matlab 11](#_Toc4431306)

[A.1.2 Startup 12](#_Toc4431307)

[A.1.3 User guide 12](#_Toc4431308)

[A.1.3.a Configuring the processing environment 13](#_Toc4431309)

[A.1.3.b Configuring the digitizer 15](#_Toc4431310)

[A.1.3.c Configuring setup and corrections 18](#_Toc4431311)

[A.1.3.d Configuring new measurement 20](#_Toc4431312)

[A.1.3.d.1 Sampling setup assistant 21](#_Toc4431313)

[A.1.3.d.2 Configuring data processing 23](#_Toc4431314)

[A.1.3.e Initiating measurement 26](#_Toc4431315)

[A.1.3.f Viewing the results 26](#_Toc4431316)

[A.1.3.g Batch processing 29](#_Toc4431317)

[A.1.4 Resources 29](#_Toc4431318)

[B SOFTWARE CONFIGURATION OF THE BUILT-IN DIGITIZERS 31](#_Toc4431319)

[C INTEGRATION OF NEW TYPES OF DIGITIZERS 32](#_Toc4431320)

[C.1 LabVIEW environment 32](#_Toc4431321)

[C.2 LabWindowsTM/CVI environment 32](#_Toc4431322)

[D POWER AND PQ TESTS 33](#_Toc4431323)

[D.1 Algorithms 33](#_Toc4431324)

[D.2 Integration of new algorithms 33](#_Toc4431325)

[D.2.1 LabVIEW environment 33](#_Toc4431326)

[D.2.2 LabWindowsTM/CVI environment 33](#_Toc4431327)

[E TITLE 1 34](#_Toc4431328)

[E.1 Title 2 34](#_Toc4431329)

[E.1.1 Title 3 34](#_Toc4431330)

[E.1.1.a Title 4 34](#_Toc4431331)

[E.1.1.a.1 Title 5 34](#_Toc4431332)

[E.1.1.a.1.1 Title 6 34](#_Toc4431333)

**FIGURES** To be updated !

[Figure A.1: Location of TWM download on the TWM GitHub webpage. 10](#_Toc4431334)

[Figure A.2: Downloading development version of TWM from GitHub webpage. 10](#_Toc4431335)

[Figure A.3: Front panel of TWM. 12](#_Toc4431336)

[Figure A.4: TWM processing environment configuration. 14](#_Toc4431337)

[Figure A.5: TWM GNU Octave package assistant. 15](#_Toc4431338)

[Figure A.6: TWM configuration panel for the niScope digitizers (for NI 5922). 16](#_Toc4431339)

[Figure A.7: TWM configuration panel for Agilent 3458A sampling multimeters. 17](#_Toc4431340)

[Figure A.8: TWM transducer configuration panel. 19](#_Toc4431341)

[Figure A.9: TWM digitizer corrections configuration panel. 20](#_Toc4431342)

[Figure A.10: TWM new measurement configuration panel. 21](#_Toc4431343)

[Figure A.11: TWM sampling setup assistant. 22](#_Toc4431344)

[Figure A.12: TWM sampling setup assistant - fundamental frequency measurement panel. 22](#_Toc4431345)

[Figure A.13: TWM processing configuration panel (raw Matlab commands mode). 24](#_Toc4431346)

[Figure A.14: TWM processing configuration panel. 24](#_Toc4431347)

[Figure A.15: TWM multicore processing setup panel. 25](#_Toc4431348)

[Figure A.16: TWM multicore processing GNU Octave servers. 26](#_Toc4431349)

[Figure A.17: TWM record viewer panel. 27](#_Toc4431350)

[Figure A.18: TWM spectrum viewer panel. 27](#_Toc4431351)

[Figure A.19: TWM processing results viewer. 28](#_Toc4431352)

[Figure A.20: TWM results table popup menu. 28](#_Toc4431353)

[Figure A.21: TWM QWTB batch processing panel. 29](#_Toc4431354)

[Figure E.1 : Blue Exemple 34](#_Toc4431355)

[Figure E.2 : Red Exemple 34](#_Toc4431356)

**TABLES** To be updated !

[Table A‑1: Typical GNU Octave packages required for TWM. 11](#_Toc4431357)

[Table E‑1 : Six\*Three exemple 34](#_Toc4431358)

[Table E‑2 : Three\*Two example 34](#_Toc4431359)

# INTRODUCTION

***WP3 - A3.3.1***

***CMI****, INRIM will produce guidance documentation on the start-up, including installation of the software, and the user interface for both LabVIEW and LabWindowsTM/CVI environments (the 2 GUIs) based on information from the report on the software tool from A2.4.5.*

*CMI will produce the guidance for the LabVIEW environment and INRIM for the LabWindowsTM/CVI environment, respectively.*

To be completed…

## TWM manual

Following chapters will describe installation and basic usage of the TWM tool [1].

### Installation

Installation of TWM consists of several components:

1. Download and unpacking of the TWM tool itself from GitHub [1].
2. Installation of the prerequisites for the TWM run, which are:
   1. LabVIEW 2013 Runtime Engine. This is needed to run any LabVIEW application.
   2. Drivers of all integrated instruments, which are currently:
      1. NI VISA drivers for handling GPIB bus.
      2. niScope drivers for handling NI digitizers.
3. Installation of the GNU Octave or Matlab for data processing (optional).

Steps 1) and 2) are mandatory. Current version of TWM is built with several digitizers integrated. Unfortunately it means the TWM application will not run unless there are drivers for all integrated. So it is necessary to install VISA drivers even for use with NI 5922 cards and “niScope” drivers even for use only with Agilent 3458A multimeters. Note this unpleasant “feature” will be fixed in the future by building several version of TWM having support only for selected instrument.

Step 3) is needed when processing of the recorder waveforms is to be performed in place. Without GNU Octave or Matlab, the TWM can only record the waveforms.

#### Installing TWM tool

TWM tool requires no installation. Its files just must be unpacked from the ZIP to any user folder, e.g.:

c:\Program Files (x86)\TWM

The build releases of TWM are relatively rare and usually contains version that was at least partially tested. It can be downloaded from GitHub webpage (see Figure A.1).

However, user may also download development version which will run only with development version of LabVIEW installed on the computer. Note the TWM was intentionally developed in LabVIEW 2013 Base version. The development version can be obtained either by cloning the TWM Git or manually by downloading the Git as a ZIP file (see Figure A.2).

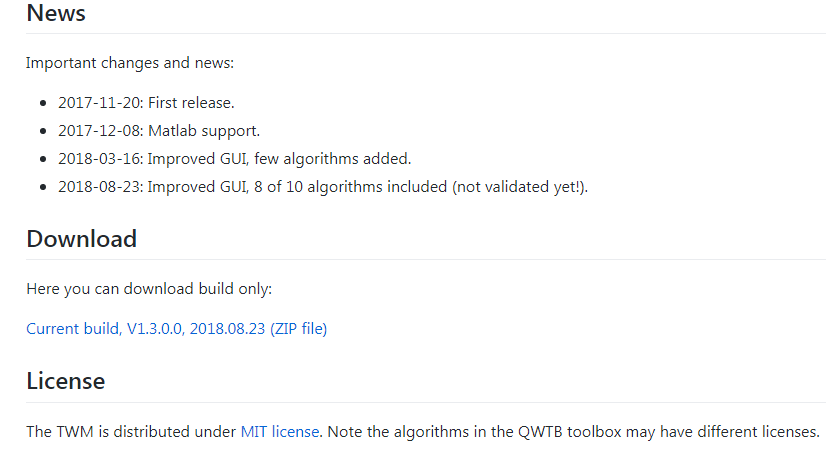


Figure A.1: Location of TWM download on the TWM GitHub webpage.

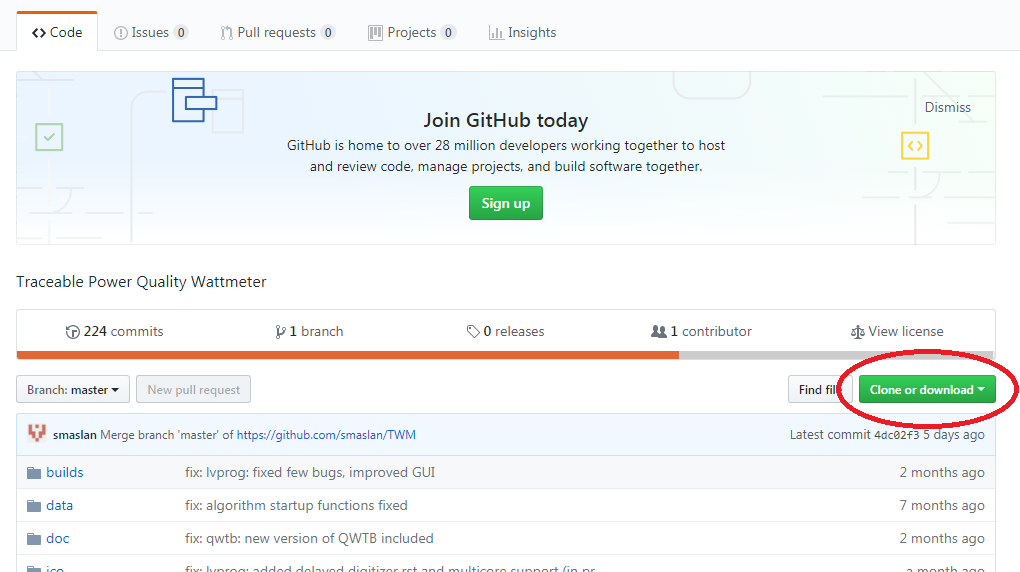


Figure A.2: Downloading development version of TWM from GitHub webpage.

#### Installing TWM prerequisites

LabVIEW applications in general requires large amount of external libraries and drivers to run. First required is LabVIEW 2013 Runtime Engine 32bit [4]. The version must be 32bit as the TWM is built as a 32bit application.

Next component are the VISA drivers, which are needed for communication via GPIB. At least version 5.4 should be installed. The newer versions are partially compatible, see [5] for selection.

Last needed component are niScope drivers, which are used to communicate with NI 5922 digitizers. These supported versions are 4.0.5 to 16.1 (see [6] for details).

#### Installing GNU Octave

In order to enable calculations of the PQ parameters from the waveforms, user must install GNU Octave (or Matlab). The TWM was tested with Octave version 4.0.0 and later. Note the preferred version of GNU Octave is always 64bit, as it has much higher memory limit. That may be useful for some memory demanding algorithms.

The download and guidance to the GNU Octave can be found at [2]. Some of the algorithms do require additional packages to be installed and loaded to the GNU Octave after its startup. This is ensured by creating a startup file in the user folder. GNU Octave will always execute the sequence of the command on the startup. E.g. in Windows, for the user named “user”, the file location will be:

c:\Documents and Settings\user\.octaverc

Note the file name actually starts with the dot! Example of the content of the file may be following:

more off;

pkg load io;

pkg load optim;

pkg load signal;

pkg load statistics;

pkg load outliers;

graphics\_toolkit('gnuplot');

One of the main uses of this file is usually to load the packages to the memory on startup. Note the packages must be installed first. Some distributions of the GNU Octave ask user to select the packages to install during the installation. Other distributions are plain without packages and those must be downloaded and installed separately following the documentation [2]. Typical packages needed for the currently implemented algorithms are listed in Table A‑1.

Table A‑1: Typical GNU Octave packages required for TWM.

|  |  |
| --- | --- |
| **Package** | **Description** |
| io | Input/output to external formats (CSV files, etc.) |
| optim | Non-linear optimization toolkit. |
| outliers | Outlier tests and removal. |
| signal | Signal processing routines (e.g. filters). |
| statistics | Additional statistical functions. |
| multicore | Multicore calculations. |
| golpi | More effective data exchange between LabVIEW and GNU Octave. |

Package “multicore” is not necessary, however it can be used to significantly speedup Monte Carlo uncertainty evaluation of some of the algorithms. Package “golpi” is designed to speed up the data interchange between GNU Octave and LabVIEW GOLPI library [7]. Note the GOLPI library should offer automatic installation of the “golpi” package to the user when the mode of communication is selected (see below).

#### Installing Matlab

Matlab [3] is alternative processing environment for the TWM, however it should be 100% compatible. TWM was tested with versions 2008b and later. Typical Matlab installation should contain all required packages and they are loaded automatically, so no additional actions after installation should be required.

### Startup

When all required components are installed, the TWM can be started by its executable “TWM.exe”. When no component is missing, the front panel should appear with no error messages. If some driver is missing, LabVIEW will throw an error with explanation which component cannot be located. Typical missing components are “niScope.dll”, “niTclk.dll” (part of “niScope” drivers) or VISA drivers. If TWM requests the libraries, follow the installation guidance in section A.1.1. TWM requires no other configurations prior starting the application itself.

### User guide

Main panel of TWM is shown in Figure A.3. All subpanels with particular configurations can be invoked from the main panel. Error indicator at the bottom will show eventual error message of the TWM. User must configure the system before any measurement can be taken. This is done by buttons Digitizer for digitizer selection and configuration, button Octave/Matlab for processing environment selection and button HW corrections for selection of the transducer and digitizer connection and correction files.

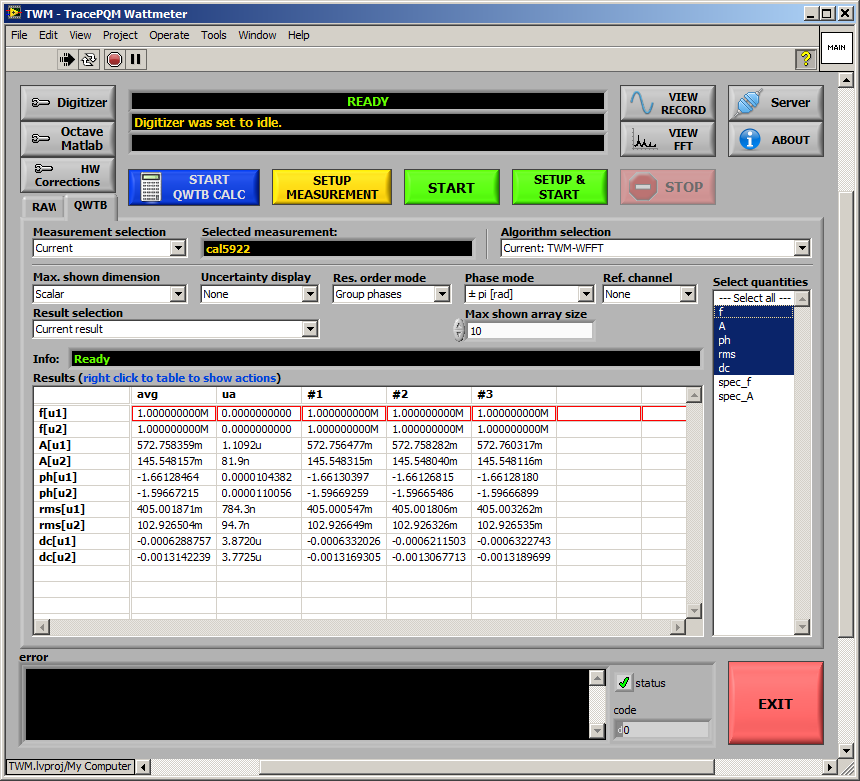


Figure A.3: Front panel of TWM.

#### Configuring the processing environment

First step before any processing can be done is to setup the processing environment, i.e. Matlab or GNU Octave by pressing button Octave/Matlab on the front panel. This will invoke panel shown in Figure A.4. The main option is checkbox Enable Matlab script mode. This will select Matlab for processing. If unchecked, the GNU Octave is selected. Matlab has no other options in the top frame. For GNU Octave user must manually list the GNU Octave binary path of the selected GNU Octave installation. Example for version up to 4.x.x is following:

C:\Octave\Octave-4.2.2\bin

Note the new version starting with Octave 5.x.x changed the internal location of the binary to either of following subfolders depending on 32/64-bit version selection:

C:\Octave\Octave-5.1.0.0\mingw32\bin

C:\Octave\Octave-5.1.0.0\mingw64\bin

Next, user may enable option Always clear function cache, which has effect only for debugging. This will force GNU Octave to reload the scripts before every command to ensure the changes in them are recognized (do not use for normal operation – it will only slow down operation). Another option is Use bitstream mode, which will enable faster communication between LabVIEW and GNU Octave. This requires the package “golpi” to be installed. TWM should offer user automatic installation. However, it is not necessary as there is mostly only small amount of data transferred between the environments. Last option is Show console window which will show GNU Octave console window for debugging. Note the Matlab will always show its console window. Do not close neither the Octave’s or Matlab’s console during TWM usage as it will terminate the application! Validity of the setup can be tested by button Restart.

Another option to be done is selection of the QWTB toolbox root folder path. Although TWM should work with main QWTB distribution, the TWM algorithms are distributed along with the TWM, so always set path to the local copy of QWTB in the TWM installation subfolder:

.\octprog\qwtb

Validity of the setup can be tested by TEST QWTB.

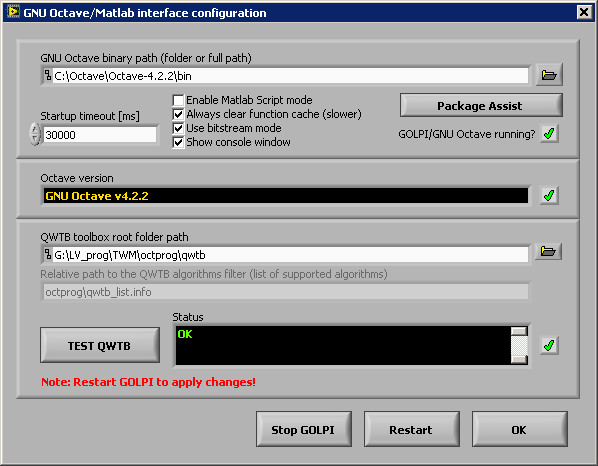


Figure A.4: TWM processing environment configuration.

A special package assistant shown in Figure A.5 was made to reduce complexity of initial GNU Octave setup. The panel can be invoked by pressing Package Assist button in GNU Octave/Matlab panel. The left list should show the installed packages in the selected installation of GNU Octave. The right panel shows which packages are already loaded after startup or are selected to load by TWM. Select and shift packages you want to load on startup by right or left buttons. Note the right list will always contain at least the packages that are loaded by startup file “.octaverc”. Also note to apply the changes you must restart Octave in panel Figure A.4. The selected packages to load should be automatically loaded after the TWM is restarted.

The assistant panel also contains button Install packages. This button will display open dialog which can be used to select downloaded package (extension “\*.tar.gz”) and TWM will try to install the package to the Octave. The success can be tested by pressing Reload Packages. The newly installed packed must be moved to right panel to be loaded after Octave restart.

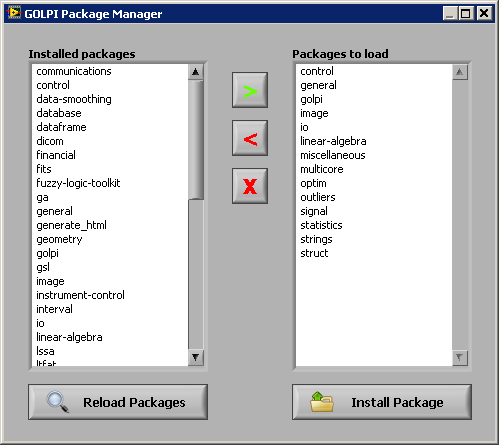


Figure A.5: TWM GNU Octave package assistant.

#### Configuring the digitizer

The button Digitizer on the main TWM panel must be pressed in order to invoke Digitizer configuration panel. Example of the panels for the niScope and 3458A digitizers are shown in Figure A.6 and Figure A.7. The panel contains Digitizer type selector which is used to select the HW to be used for digitizing. Below is a text frame with brief description of the selected digitizer.

Next option On device close at the bottom-left of the panel is used to determine behavior of the TWM after the digitizers were used. Selection of Immediate reset will reset all HW related to the digitizer to the safe state. Selection of Delayed reset will do the same when the HW is not used for preset timeout. This second option is preferred way especially for the sampling multimeters 3458A as it saves the relays for repeated measurements. Option Force reset on open will issue reset command to the instruments related to the digitizer every time the digitizer is initialized. Normally this should be disabled.

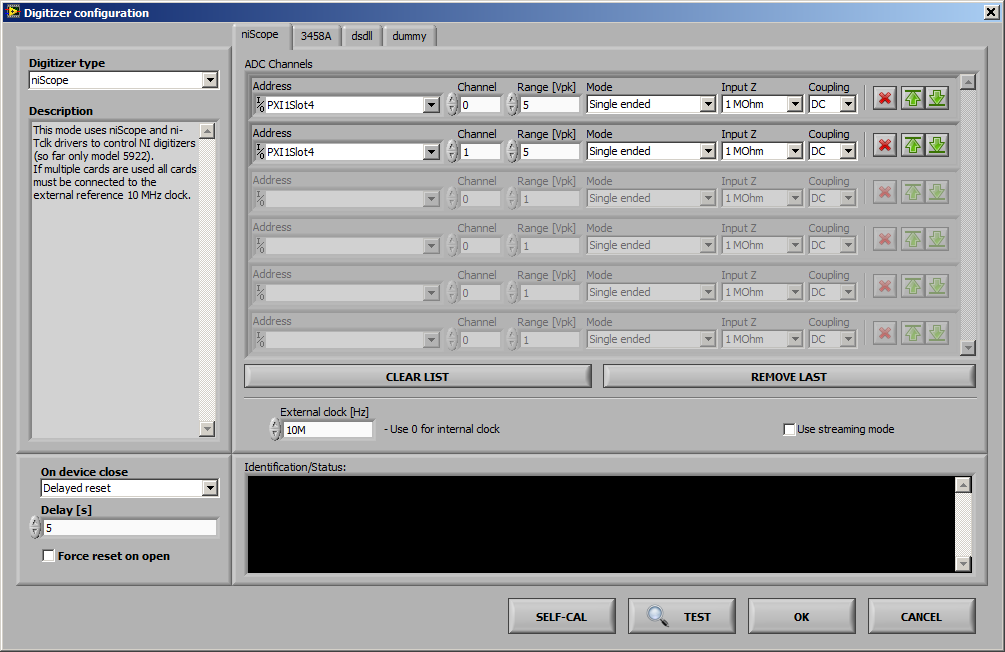


Figure A.6: TWM configuration panel for the niScope digitizers (for NI 5922).

Each digitizer has one private page with specific options. The page for niScope digitizers is shown in Figure A.6. User can select any number of physical channels of any number of digitizing cards by adding or removing rows to the table ADC Channels. Each virtual channel (row) contains Channel ID (note niScope counts channels from zero), input Range [Vpk], which is maximum applied voltage (e.g., 5 for range ±5 V), Input Z is input impedance of the channel and Coupling is mode of input coupling. The icons on the right of each row enables reordering or removal of row(s). Note the physical channels of the card(s) which are not selected are automatically set to ground to prevent buildup of noise at near channels. The driver supports PLL synchronization to the external 10 MHz source, which is needed for coherent sampling or when synchronizing multiple cards. Type in the nominal sync. frequency to the External clock [Hz] entry to enable the PLL. Use value of 0 to switch to the internal clock. Last option is Use streaming mode, which will enable runtime streaming of the sample data from the cards to the disk storage. This option still uses the internal card memory, but it reads the sample data along with the sampling, so unless the disk storage is too slow, user may digitize indefinitely. Validity of the setup can be tested by the button TEST. Eventual problems will be shown in the black area below. The SELF-CAL button will initiate self-calibration routine of all selected digitizer cards.

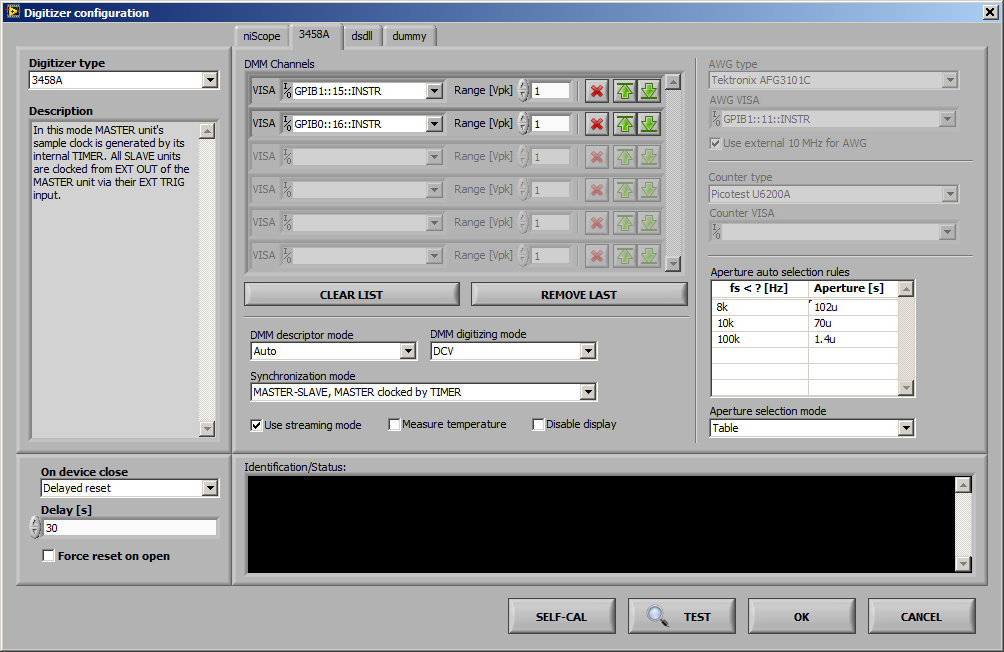


Figure A.7: TWM configuration panel for Agilent 3458A sampling multimeters.

Configuration panel for the digitizers 3458A is shown in Figure A.7. User may create any number of virtual channels by adding or removing rows to the table DMM Channels. VISA is visa address of the 3458A instruments. Range [Vpk] is nominal range of the DMM. Typical value for digitizing is 1 V. Color icons on the right enable reordering or removal of the row(s). Driver has no internal limitation of the channels count. Control DMM descriptor mode allows to change the way the multimeters are identified. Selection of Sequential will generate fake identifiers “HP3458A, channel 1”, “HP3458A, channel 2”, etc. These are identifiers that must match channel names in the digitizer correction files. Another option From calibration string means the TWM will scan the calibration string of each DMM (command “CALSTR?”) and if it discovers serial number in there, it will use it as an identifier. Using CALSTR for serial numbers is common practice for distinguishing the DMMs via GPIB. Selection of Auto will choose automatically with priority of calibration strings. Next option is DMM digitizing mode, which selects the digitizing method. Available modes are DCV, DSDC and DSAC (refer to 3458A manual for meaning). Option Synchronization mode determines the way how multiple DMMs are synchronized. Four selections are possible:

1. MASTER-SLAVE, MASTER clocked by TIMER: Master (first DMM in the list) clocks itself from internal TIMER. All other DMMs takes sample clock (EXT TRIG) from master’s EXT OUT. This option means sampling rate of the whole virtual digitizer is determined by the MASTER’s internal TIMER and thus by MASTER’s reference clock.
2. MASTER-SLAVE, MASTER clocked by AWG: Master EXT TRIG is clocked from the arbitrary waveform generator (AWG). All slaves are clocked via EXT TRIG from MASTER’s EXT OUT. The sampling clock is determined by the AWG.
3. All clocked by AWG: EXT TRIG of all DMMs are clocked from the AWG.
4. All clocked by TIMER: All DMM’s are clocked from their internal TIMERs. This special mode is dedicated for DMMs with modified internal reference clocks, which can be locked to the external 10 MHz source. The absolute time shift between the channels is determined by additional counter connected between EXT OUT outputs of the DMMs. Note this is intended for two DMMs only.

Typical selection when non-coherent sampling is acceptable is (i). For coherent sampling use modes (ii) or (iii). Note in modes (ii) and (iii) user must define AWG type and AWG VISA address and in mode (iv) user must define Counter type and Counter VISA address.

Note the modes (ii) and (iii) can be used with an external non-controlled source of sampling pulses, which may be e.g. some PLL synchronizing unit or other instrument’s output. In that case the AWG type has to be set to dummy, which means TWM will ignore it. The sample clock generator will be connected to the EXT TRIG of master (ii) or EXT TRIG of all DMMs (iii). Typical example of such operation is calibration of Fluke 6100 calibrator series, which are equipped by a “Sample Ref Output” connector. This output can be enabled and it outputs fundamental frequency multiplied by certain factor. It is convenient e.g. for achieving coherent sampling.

Next option is Use streaming mode, which will enable direct readout of the samples via GPIB without the limitation of the internal DMM memory. This will enable up to 16 MSamples per channel to be captured in one record. However, this mode requires one GPIB controller for each DMM and it may fail for the highest sampling rates. The fail will be indicated by beep and message “trigger too fast” while sampling. The streaming method is faster, as the data are read continuously. Another option is Measure temperature, which will read the internal DMM temperature once per record. Note it takes additional time as the reading is made via the main ADC. Last option is Disable display which will turn the CFL displays off while digitizing. It is recommended to reduce the noise.

Last option is selection mode of the DMM integration time (aperture) made by control Aperture selection mode. Three options are available:

1. Manual: user selects the aperture in the new measurement configuration.
2. Table: TWM selects the aperture based on the rules in table Aperture auto selection rules depending on the sampling frequency. This option is helpful to limit possible apertures to few calibrated values. The example shown in Figure A.7 defines to select aperture of 102 µs up to 8 kSa/s sampling rate, 70 µs up to 10 kSa/s and 1.4 µs up to 100 kSa/s.
3. Maximum: TWM selects the maximum possible aperture based on empirical formula. Note each aperture needs separate calibration of DMMs, because residual errors of TWM changes e.g. when crossing 100 µs aperture.

Configuration of the virtual digitizer can be confirmed and tested by pressing TEST. Any error detected should appear in red color in the black area. Autocalibration of the DMMs can be initiated by the SELF-CAL button.

#### Configuring setup and corrections

User must define connection of the transducers and their correction files before any meaningful measurement can be done. This is done by pressing button HW Corrections on the main panel, which will invoke panel shown in Figure A.8. The panel contains two pages. One for configuration of Transducers and another one for Digitizers corrections. The formats of the correction files is described in [13].

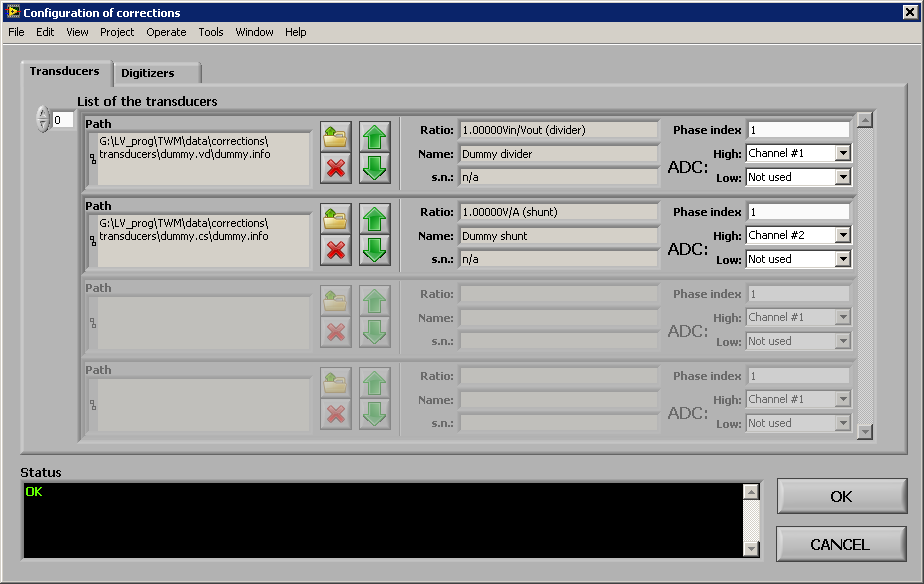


Figure A.8: TWM transducer configuration panel.

The page Transducers allows user to create list of transducers. User may create multiple transducers by adding or removing the rows to the List of the transducers. Each row designates one transducer, which may be either for voltage or current channel. Open icon is used to select the transducer correction file (voltage divider or shunt). When the load is successful, the type and nominal ratio will be shown in the middle of the row. User must then assign Phase index to each transducer. This is integer number that is used by TWM to pair the voltage and current transducers for algorithms such as power. Selectors High and Low are used to map the transducer output(s) to particular digitizer virtual channels. High channel must be always assigned. For differentially connected transducers, user must also assign the Low side digitizer channel. For single ended mode assign Not used to the Low. Every time the change is made, TWM will check the validity and show either OK or error message in the black area. Note the TWM does not require the transducers to be defined for single input algorithms, such as PSFE. It will load default unity transfer transducer. But it is required for multi-input algorithms such as power, because TWM must know which is voltage and current and to which phase it belongs. Also note the TWM installation contains examples including “dummy” divider and shunt correction with unity transfers in the TWM folder:

./data/corrections/transducers

These should be used if no input correction is needed.

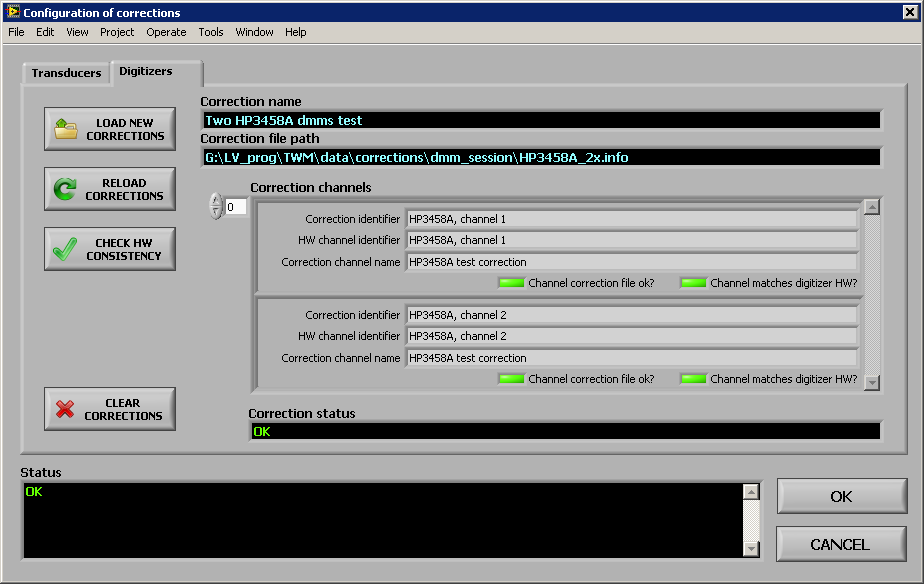


Figure A.9: TWM digitizer corrections configuration panel.

Digitizers page shown in Figure A.9 is used for loading the digitizer correction file. The correction file is selected by pressing LOAD NEW CORRECTIONS. This is verified by pressing CHECK HW CONSISTENCY. The content of the correction file must match the configured digitizer, i.e. the channel names and order must be the same. RELOAD CORRECTIONS will just reload the same correction file in case it was modified. CLEAR CORRECTIONS is used to clear digitizer corrections. TWM will work without digitizer corrections selected normally, except no correction will be applied.

#### Configuring new measurement

New measurement configuration can be invoked by button SETUP NEW MEASUREMENT or button SETUP&START. These will open panel shown in Figure A.10. The panel is used to setup Sampling rate, Aperture time of the digitizer ADC, Samples count to record or Measurement time, measurement Repetition cycles, additional Repetition settling for extra delay between repetitions. Next group of options is Trigger mode and its parameters. Mostly it is not needed, so Immediate can be used. Note the mode Level is always derived from the first channel of the digitizer. This decision was made because of the DMM synchronization modes MASTER-SLAVE which do not allow principally different operation. The mode External is some form of external trigger input dependent on the selected digitizer. For DMMs it is EXT TRIG input of the MASTER DMM (if not used for AWG input). For NI 5922 it is the TRIG input on the first card in the digitizer channels list. The panel also contains Voltage range selection. Normally this is not needed, because the ranges can be set in the digitizer configuration, but the default ranges can be overridden from here, when other than Keep defaults option is selected. The TWM will allow user to set the same range to all virtual channels, or separately to odd and even (intended for voltage and current channels).

The mandatory part of this panel is setup of the destination folder for the measurement data. User must select some Folder for measurement data files to which the TWM will store the data and results. User must also choose a Measurement name base, which is subfolder to be created in the Folder for measurement data files. This name may contain variables as shown in the panel which will be replaced. Preview of the final path to the measurement is shown in the indicator Full measurement base path. Normally, if user does not require to archive the measured data for future processing, the Measurement name base may be still the same and TWM will simply overwrite the previous data by new ones (it will always ask to delete old data). By adding the %id% variable to the name, TWM will auto increment the Measurement number every time the measurement is taken, so the data will be archived in the Folder for measurement data files.

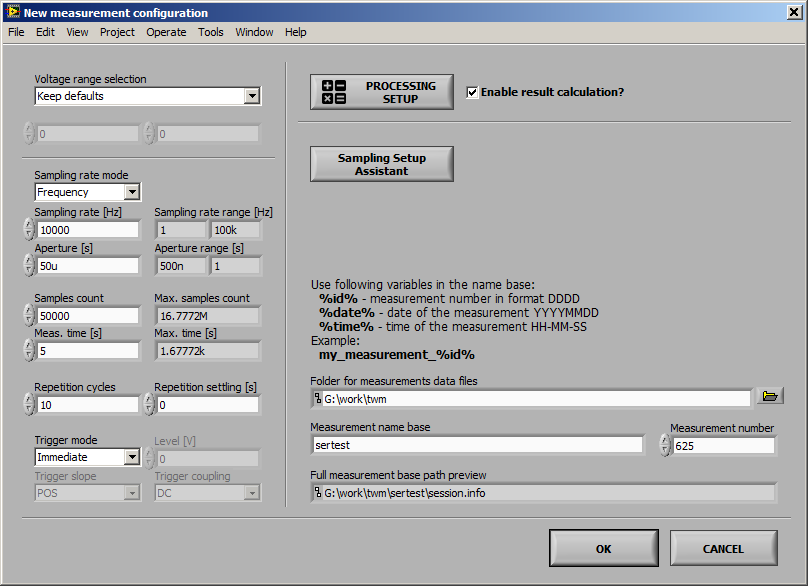


Figure A.10: TWM new measurement configuration panel.

Processing setup and Sampling Setup Assistant are described in the following chapters. Option Enable result calculation must be checked to perform the data processing immediately after digitizing. Otherwise TWM will only record the data to the measurement folder(s) and the processing can be initiated afterwards.

When the setups are done, user may confirm by pressing OK or CANCEL to return back. If the panel was invoked by SETUP&START, the measurement will start immediately after OK.

##### Sampling setup assistant

In some cases user may need to setup coherent sampling. The New measurement configuration panel has integrated Sampling Setup Assistant for easing this task. Panel of the assistant is shown in Figure A.11. Purpose of this panel is to measure or manually enter frequency of the fundamental component of the signal and calculate combination of the sampling rate and samples count which will result in coherent sampling. The frequency can be measured using subpanel shown in Figure A.12 that can be invoked by pressing MEASURE. Alternatively, if it is known, it may be entered directly to the Reference frequency fr. The frequency can be multiplied by a factor P/Q and shifted by given offset to resulting Fundamental frequency f0. At this point user must enter search range of the sampling rate Minimum fs and fs max tolerance and total measurement time range Min integration t., Max integration t. or Min periods, Max periods counts. By pressing FIND ROUGH, the TWM will select nearest possible sampling rate and time, but not coherent. This is just to show user rough calculated parameters of the sampling such as DFT bin spacing, periods count, etc. By pressing FIND COHERENT, TWM will call experimental m-function that will try to find coherent setup in the set ranges. The function is quite complicated and may take a while, so Max timeout [s] control is used to set maximum time to spend on the calculation. Control Max setup error defines how accurately TWM searches the coherent setup. Note the digitizers may have very limited step of sampling period or frequency, so there may be no coherent setup for irrational fundamental frequencies f0. This may be prevented by selecting higher tolerance.

The subpanel for automatic frequency measurement of the reference frequency shown in Figure A.12 is basically a reading loop for a selected counter. It will perform up to Max averages readings and calculates average frequency. When OK is pressed, the measured value will be transferred to the Sampling setup assistant panel.

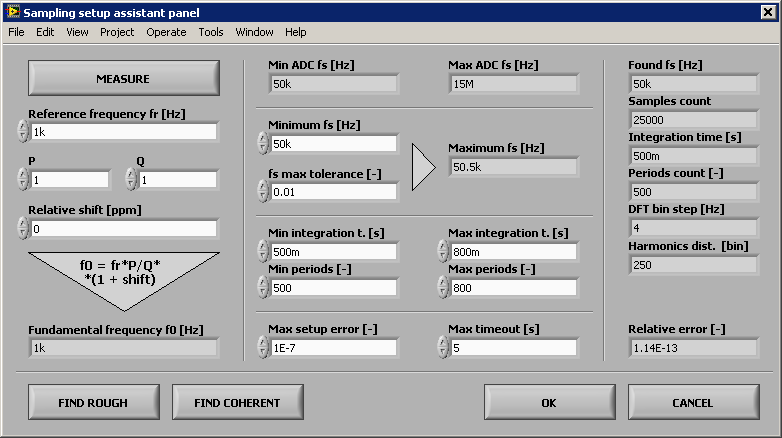


Figure A.11: TWM sampling setup assistant.

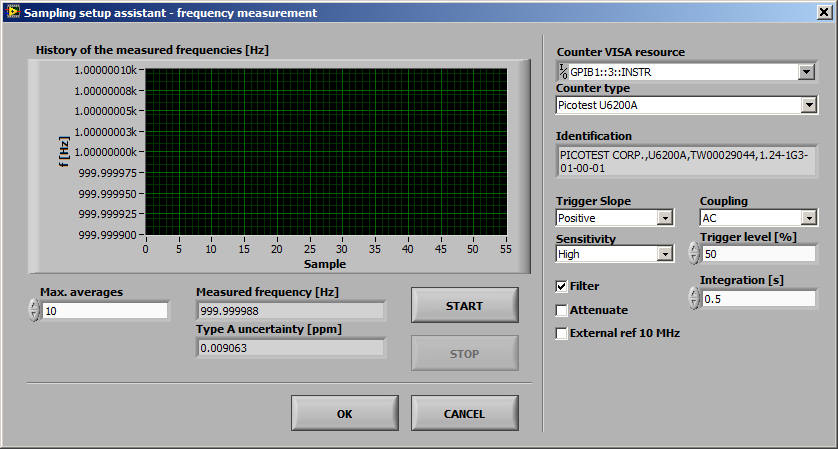


Figure A.12: TWM sampling setup assistant - fundamental frequency measurement panel.

##### Configuring data processing

Panel for configuration of the processing is accessible from the New measurement configuration panel shown in Figure A.10 or from batch processing panel Figure A.21 (see section A.1.3.g). There are two main modes of processing which are selected by the page selector. The RAW command is a simple debugging mode which enables user to simply type in or load m-code to execute on the raw measured sample data (see panel shown in Figure A.13).

The other and preferred option is the QWTB mode, which is shown in Figure A.14. When the panel is opened, TWM will fetch the available List of algorithms from QWTB distribution linked in the panel GNU Octave/Matlab configuration panel in Figure A.4. Note this operation may take some time during which the controls will be grayed. The algorithms may be reloaded manually by pressing the RELOAD ALGs button. Note the actual displayed list is not full list of QWTB algorithms. It is intentionally limited by the filtering file (see [12] for details):

.\ocprog\qwtb\_list.info

This decision was made to remove algorithms which are not compatible with TWM. When no error occurred and correct version of QWTB was selected, the list should contain some 10+ algorithms. By selecting one, TWM will fetch the algorithm info, which may also take some time. The description should appear in the indicators on the right side. The LED indicators will indicate features of the algorithm. The table Parameters of the algorithm will be filled by the user parameters of the algorithm. User may enter the numeric or text into the table following the algorithm manual [8]. The table can be cleared by the button CLEAR PARAMETERS. Note each row name may contain a suffix in the parentheses. Opt. means the parameter is optional. Alt. means it has alternative parameters (details can be found in QWTB guide [9]). Description of the selected parameter (row) is shown in the Variable description indicator.

Bottom part of the panel contains several specialized options. First, user can segment the recorded waveform by selecting Start offset of the first sample and by limiting the amount of input data to process by Input data length. Another option is Calculate result just once…, which means to send multiple records to the algorithm at once. This is currently supported only by the TWM-THDWFFT algorithm. Last option is Calculate uncertainty just once, which means the uncertainty is calculated only for the first repetition cycle (record) and it id disabled in following cycles to save processing time.

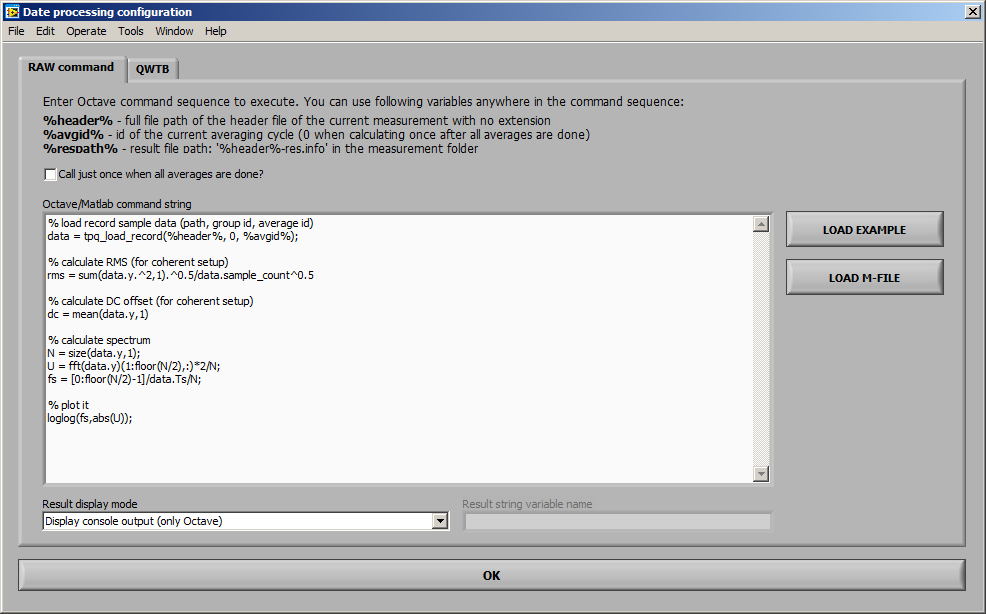


Figure A.13: TWM processing configuration panel (raw Matlab commands mode).

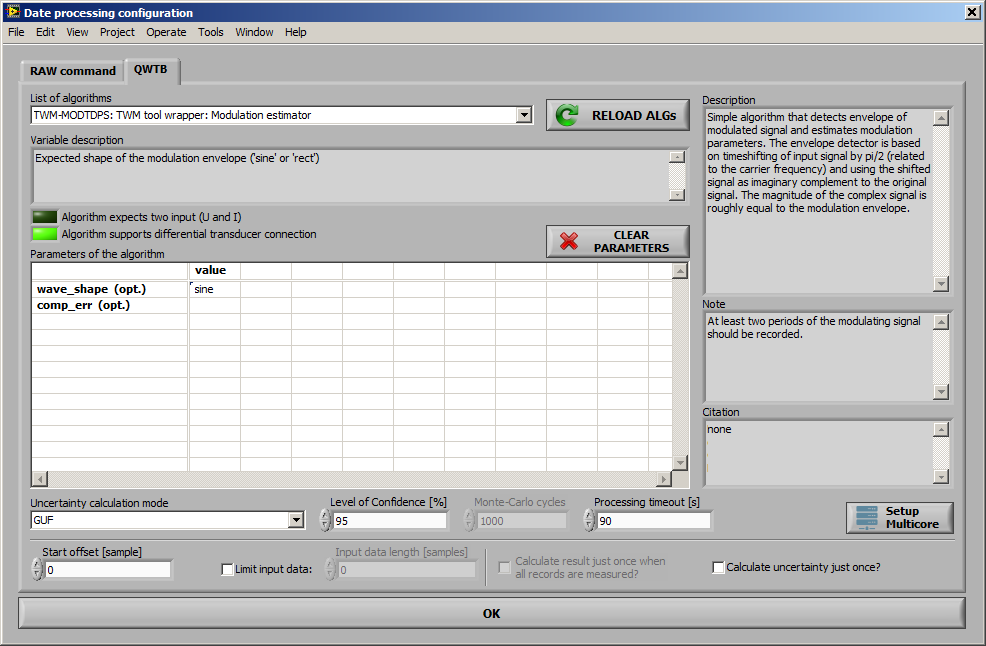


Figure A.14: TWM processing configuration panel.

Next, user should select Uncertainty calculation mode. Depending on the algorithm, there may be None (no uncertainty calculation), GUF (GF method or fast estimator) or Monte Carlo (numeric Monte Carlo calculation). User should also enter desired Level of Confidence for the calculation. For Monte Carlo there is also entry Monte Carlo cycles which should be always set to at least 1000 iterations (refer to algorithms manual [8] for optimal setup). Control Processing timeout [s] is maximum allowed calculation timeout. Note this will work only for GNU Octave as Matlab interface has no abort capability from LabVIEW. Also note if the timeout is reached, the TWM will throw an error, but the processing still runs in the GNU Octave, so Octave must be eventually restarted using the Octave/Matlab configuration panel.

The Monte Carlo calculation has one more advanced option – the Setup Multicore. By pressing the button, the panel in Figure A.15 will be shown. The default Execution mode is Singlecore, which means all iterations of the Monte Carlo will be performed on a single core. Option Multicore uses “parfor” command for Matlab or “pracellfun” command from “optim” package [11] for GNU Octave. Note the “parcellfun” will not work in Windows in current version of GNU Octave. Last option is Multistation, which will work only for GNU Octave and it uses package “multicore” [10] which must be installed and loaded. This package is using shared Multistation jobs folder to distribute so called job files, which are then processed by any number of GNU Octave servers that can be started by pressing Start Servers. If the operation was successful, the panel shown in Figure A.16 will appear. Note all the servers print to the same console so the content will be most likely unreadable. The only purpose is to see the processing is happening and to enable easy termination. The servers can be terminated by simply closing their console window. This option has significant effect on the performance of Monte Carlo calculation, however it should be used only after the Singlecore was tested and working properly.

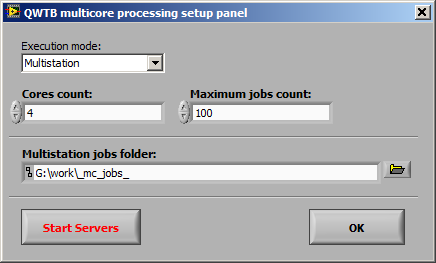


Figure A.15: TWM multicore processing setup panel.

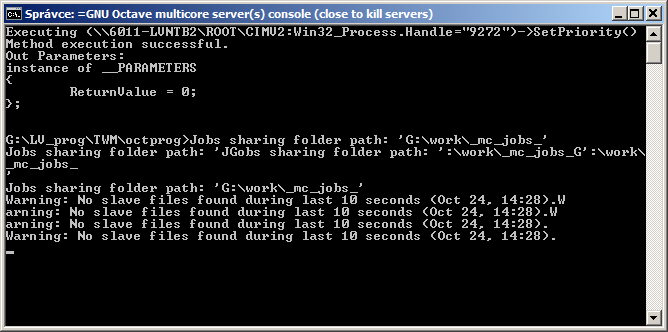


Figure A.16: TWM multicore processing GNU Octave servers.

#### Initiating measurement

When everything is configured, the measurement can be initiated by pressing the START or SETUP&START buttons on the main panel. Note pressing the START will still ask for new setup if the HW configuration was changed. The ongoing measurement can be terminated by pressing the STOP button at any time.

#### Viewing the results

User may inspect the digitized waveforms using the button VIEW RECORD at any time. The panel shown in Figure A.17 will be displayed. This panel rereads the data from saved record on runtime, so it may be a little slow for the long records. That is why it is equipped by entry Max samples to display to limit the viewed data amount. The panel can be displayed permanently along with the TWM.

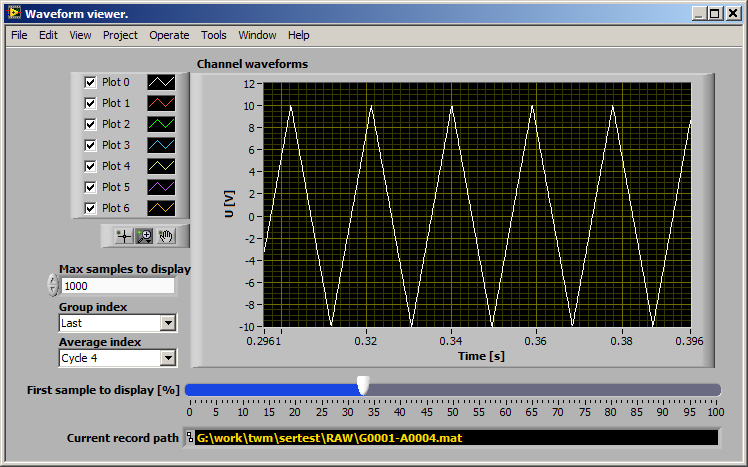


Figure A.17: TWM record viewer panel.

If the algorithm contains the spectra of defined in the “qwtb\_list.info”, they can be displayed in dedicated panel shown in Figure A.18. The panel can be shown by pressing button VIEW FFT on main panel Figure A.3. The panel shows the spectra calculated by selected algorithm. So it will not display spectrum if processing is disabled or selected algorithms does not return it as one of the output quantities.

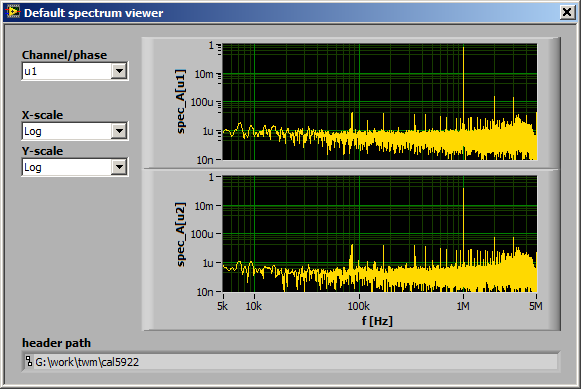


Figure A.18: TWM spectrum viewer panel.

The results of the processing are displayed in the main panel of TWM. Depending on the processing mode, TWM will display either the RAW panel or QWTB results panel. The RAW panel is just a plain text. The QWTB is shown in Figure A.3. First, it is important to mention the result viewer operates independently to the measurement. It can be used to view the results while TWM is digitizing and calculating new results. However, all the results formatting and querying is performed in GNU Octave/Matlab, so when the data are being processed, the display of results will be halted.

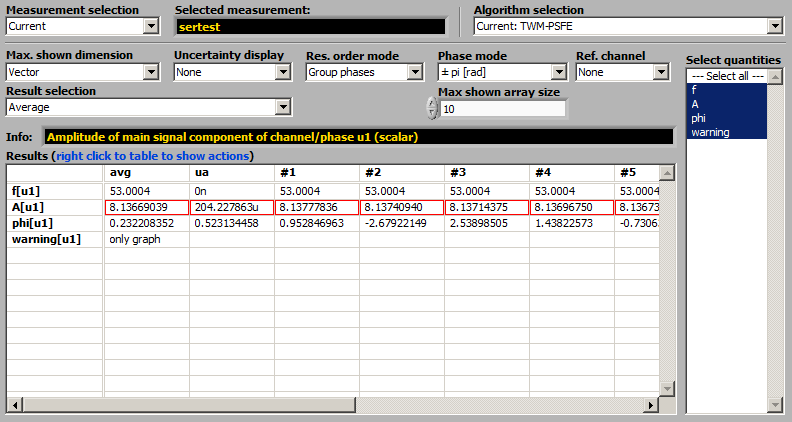


Figure A.19: TWM processing results viewer.

First control to use is Measurement selection. To speed up the TWM operation, user may choose None to stop result viewer. Next option is Current, which will show currently measured results. Last option is Load from file to select previously processed results. Next control is Algorithm selection, which will contain names of all algorithms that were applied to the particular measurement. Next, user must select Max. shown dimension entry to define how the results should be displayed. Option Scalar means TWM will show only scalar quantities and it will show the results from all repetition cycles at once. Option Vector or Matrix will show scalars and vectors (and matrices), so it can show only one result at once. User may select desired result by entry Result selection. TWM can display Current result, selected one or Average of all available. Note the averaging for large results count may take considerable time. Next, user may Select quantities to display individually. Next option is Uncertainty display, Res. order mode to regroup the quantities and Phase mode to select phase display format. Max. shown array size limits the size of vector and matrix quantities to be displayed, because TWM was not designed for showing large amounts of data as a table. If the size is exceeded, TWM will display “only graph” instead. Ref. channel is used to display phase differences between channels. This is useful e.g. for measuring phase shift by PSFE or similar algorithm.

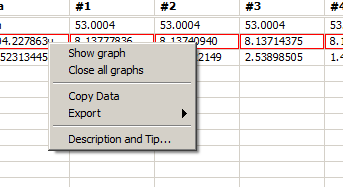


Figure A.20: TWM results table popup menu.

The result quantities and their uncertainties are shown in the table Results. By selecting the row, TWM will display description from the QWTB toolbox. Also note there are several options available by right click to the table content (see Figure A.20). First, the data can be exported to MS Excel (it must be installed first). Next, any row can be shown as a graph. This is useful for spectra or viewing the history of given quantity. Note the graphs are generated by GNU Octave/Matlab so they remain opened until closed manually or via Close all graphs option.

#### Batch processing

TWM can be used as a sampling tool only and the processing can be done later. Also user may need to calculate additional parameters from already digitized signals. TWM is equipped by the batch processing tool for this case. The tool shown in Figure A.21 can be invoked by pressing START QWTB CALC. User must select the Measurement session. Next, the tool will show available measurement groups and records, which must be selected first. Next, the algorithm must be selected by pressing the PROCESSING SETUP button. After the selection, the processing can be started by pressing START and eventually terminated by pressing STOP. The new measurement results will be shown in the results viewer in the main panel. Note calculating repeatedly the same algorithm will always override past results for the same algorithm. However, the results from the other algorithms will be untouched and available for displaying.

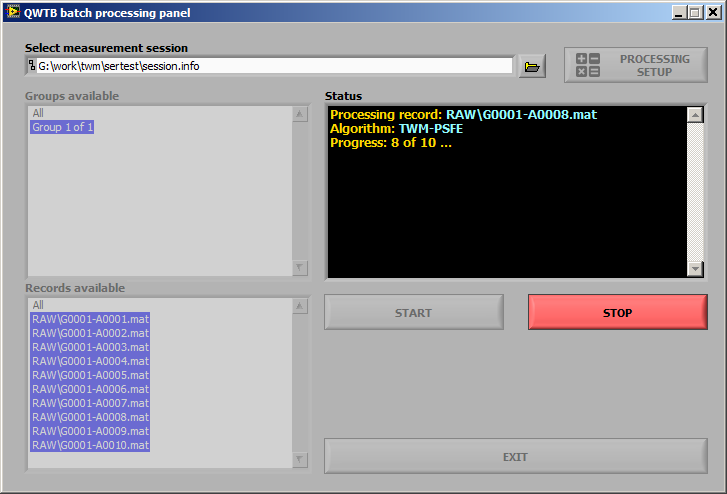


Figure A.21: TWM QWTB batch processing panel.

### Resources

1. Stanislav Mašláň, et al. *TWM tool*, url:   
   <https://github.com/smaslan/TWM>
2. John W. Eaton, *GNU Octave - Scientific Programming Language*, url:   
   <https://www.gnu.org/software/octave/>
3. MathWorks, *Matlab*, url: <https://www.mathworks.com/products/matlab.html>
4. National Instruments, *LabVIEW Run-Time Engine 2013 (32-bit)*, url:   
   <http://www.ni.com/download/labview-run-time-engine-2013/4059/en/>
5. National Instruments, *LabVIEW and VISA drivers compatibility*, url:   
   <http://www.ni.com/product-documentation/53413/en/>
6. National Instruments. *LabVIEW and niScope drivers compatibility*, url:   
   <http://www.ni.com/product-documentation/53540/en/>
7. *GOLPI library – GNU Octave to LabVIEW Pipe Interface*, url:   
   <https://github.com/KaeroDot/GOLPI>
8. *A244 TWM algorithms description*, Available at “./TWM/doc” or url:   
   <https://github.com/smaslan/TWM/blob/master/doc/A244%20Algorithms%20description.pdf>
9. *QWTB – Q-Wave toolbox*, url:   
   <https://qwtb.github.io/qwtb/>
10. Stanislav Mašláň, Martin Šíra, Markus Buehren, *Octave-Multicore package*, url:  
    <https://gitlab.com/KaeroDot/octave-multicore>
11. Hayato Fujiwara, Jaroslav Hajek, Olaf Till, *GNU Octave package parallel*, url:   
    <https://octave.sourceforge.io/parallel/index.html>
12. Stanislav Mašláň, *A232 TWM file formats and concept*, available at “./TWM/doc”, url:  
    <https://github.com/smaslan/TWM/blob/master/doc/A231%20Data%20exchange%20format%20and%20file%20formats.docx>
13. Stanislav Mašláň, *A231 Corrections Files Reference Manual*, available at “./TWM/doc”, url:   
    <https://github.com/smaslan/TWM/blob/master/doc/A231%20Correction%20Files%20Reference%20Manual.docx>.

# SOFTWARE CONFIGURATION OF THE BUILT-IN DIGITIZERS

***WP3 - A3.3.2***

***INRIM*** *with support from Metrosert will produce guidance documentation on the software configuration of the built-in digitizers (bandwidth, aperture time, sampling rate etc.), taking into account information from A3.1.2, A3.1.3 and A2.4.3.*

CMI: This is so called rubbish activity for EURAMET – does no sense. All that I can imagine here is to write aperture vs sampling rate limit for 3458, selection of sampling mode for 3458, selection of range – not all are good for sampling, see Rado’s book. Something similar for 5922.

To be completed…

# INTEGRATION OF NEW TYPES OF DIGITIZERS

***WP3 - A3.3.3***

***INRIM*** *and CMI will produce guidance documentation on the integration of new types of digitizers (in addition to the NI 5922 digitizer and the sampling DMM 3458A) including integration of calibration datasets of all the components used into the open software tool, based on information from A2.4.3 and A2.4.4.*

*CMI will produce the guidance for the LabVIEW environment and INRIM for the LabWindowsTM/CVI environment.*

To be completed…

## LabVIEW environment

CMI: This will be basically copy of A2.4.x but question is how detailed it should be? Should this be just informative with reference to details in A2.4.3?

This may be simply about the correction datasets but there is already reference manual to the corrections in the GitHub of TWM, folder “doc”. It contains all correction together because there is no sense to split it between digitizer and transducers.

## LabWindowsTM/CVI environment

# POWER AND PQ TESTS

***WP3 - A3.3.4***

***TUBITAK*** *and CMI will jointly produce a document briefly describing the power and PQ tests included in the initial database of ~10 algorithms from A2.3.2-A2.3.6 and will produce guidance on the integration of new algorithms into the open software tool using input from A2.4.4.*

CMI: I assume this should be just a summary of detailed algorithms description from A2.3.x. Cannot be done before that.

The integration will be identic for both systems as long as INRIM will follow the CMI guidance on the way of communication between CVI and Matlab. In that case the integration is mostly just following Q-Wave project toolbox QWTB. That is already documented. The draft of the all corrections to be passed to the algorithm is already made in the TWM GitHub, folder “doc”!

To be completed…

## Algorithms

## Integration of new algorithms

### LabVIEW environment

### LabWindowsTM/CVI environment

# TITLE 1

Title 1 title 1 title 1 title 1 title 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table E‑1 : Six\*Three exemple

## Title 2

Title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2 title 2.



Figure E.1 : Blue Exemple

### Title 3

Title 3 title 3 title 3 title 3 title 3 title 3 title 3 title 3 title 3 title 3 title 3 title 3 title 3 title 3 title 3 title 3 title 3.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

Table E‑2 : Three\*Two example

#### Title 4

Title 4 title 4 title 4 title 4 title 4 title 4 title 4 title 4 title 4 title 4 title 4 title 4 title 4 title 4 title 4 title 4 title 4 title 4.

##### Title 5

Title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5 title 5.



Figure E.2 : Red Exemple

###### Title 6

Title 6 title 6 title 6 title 6 title 6 title6 title 6 title 6 title6 title 6 title 6 title 6 title6 title 6 title 6 title6 title 6 title 6 title 6 title 6 title 6 title 6 title 6 title 6 title 6 title 6.