calmet

LabVIEW Driver for C300 Power Calibrator

Three Phase Source and Automatic Tester **C300**

LabVIEW DRIVER FOR
THREE PHASE POWER CALIBRATOR
AND POWER ENGINEERING APPARATUS TESTER
type C300

USER'S MANUAL

calmet Ltd.
POLAND, 65-472 ZIELONA GORA, Kukulcza 18
tel.+48 68 324 04 56 fax+48 68 324 04 57
e-mail: mail@calmet.com.pl internet: www.calmet.com.pl

C300 LabView Driver user manual
2011-02

CONTENTS

1. FIELD OF APPLICATION

This document is an user manual of LabVIEW driver for Calibrator C300 designed by Calmet Ltd. Document described driver installation process, functions included into the driver and examples of use the driver in application.

2. DRIVER INSTALLATION

2.1. System requirements

Driver was written for *National Instruments LabView 2009* program.

For communication with the calibrator driver use dedicated DLL library *CalmetC300.dll*, which was created in *Microsoft .Net 2.0*. platform and for using the driver user have to install *Microsoft .Net 2.0 Framework.* It can be downloaded from *Microsoft Download Center* site:

http://www.microsoft.com/downloads/en/default.aspx

2.2. Installation process

Driver is distributed in a *ZIP* archive. All files from archive should be copied in to *instr.lib* catalog in *LabVIEW* program localization. Default localization is:

C:\Program Files\National Instruments\LabVIEW 2009\instr.lib

Driver is automatically added by the LabVIEW, when it is starting. In the functions palette it is localized in *Express/Output/Instr Drivers/Calmet C300/* directory.

3. FUNCTIONS

Each function has an *error in* input and *error out* output, that can be used to handle errors, if they occur during program execution.

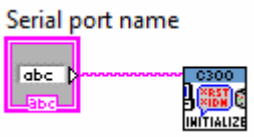
3.1. Tree.vi

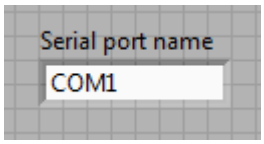
The VI Tree displays all the user-callable VIs of the instrument driver in an organized table.

3.2. Initialize.vi

Before calling any function provided by the driver, it must to establish connection with the C300 calibrator by the *Initialize.vi* calling subroutine.

Required parameter is the name of the serial port to be used for communication with the calibrator such as *COM1* or other serial ports.





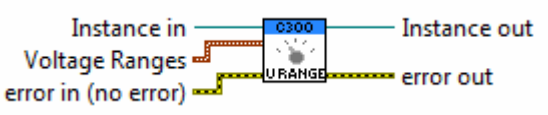
After initializing the calibrator in the *Instance out* of the terminal, the handle of calibrator connection is available. It must be used by all subsequent sub-controllers used.

3.3. Close.vi

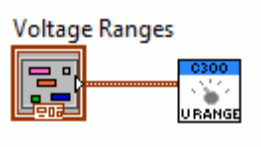
Function *Close.vi* must be used for terminate the connection in any program, where C300 driver was used. *Instance in* input must be connected with handle from *Initialize.vi* program.

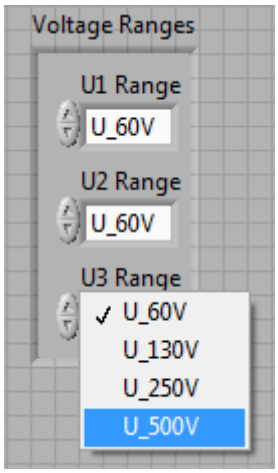
3.4. Basic functions

3.4.1. SetVoltageRanges.vi

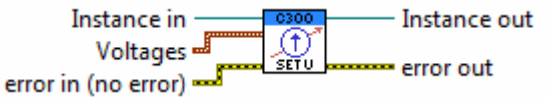


Sets the voltage range for each voltage output channel. The *Voltage Ranges* can be one of the following values: 1 – 60V; 2 – 130V; 3 – 250V; 4 – 500V.

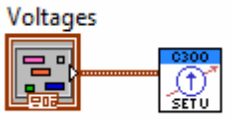




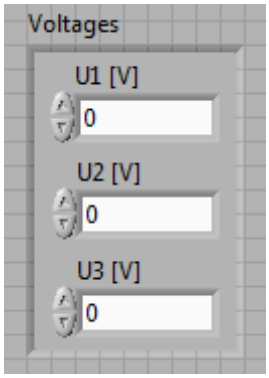
3.4.2. SetVoltages.vi



Sets the voltage for each voltage output channels.

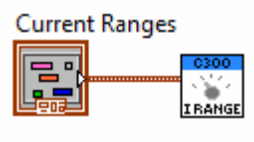


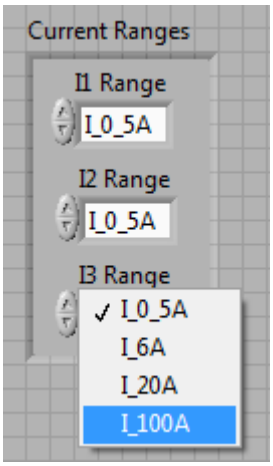
Voltages (cluster of 3 elements)
U1 [V] (single [32-bit real (~6 digit precision)])
U2 [V] (single [32-bit real (~6 digit precision)])
U3 [V] (single [32-bit real (~6 digit precision)])



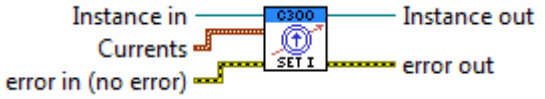
3.4.3. SetCurrentRanges.vi

Sets the current range for each current output channel. The *Current Ranges* can be one of the following values: 1 – 0.5A; 2 – 6A; 3 – 20A; 4 – 120A.

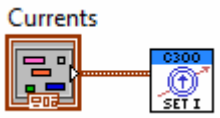




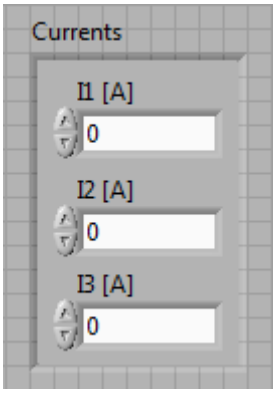
3.4.4. SetCurrents.vi



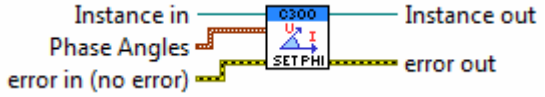
Sets the current for each current output channels.



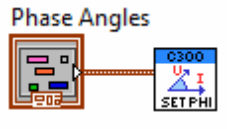
Currents (cluster of 3 elements)
I1 [A] (single [32-bit real (~6 digit precision)])
I2 [A] (single [32-bit real (~6 digit precision)])
I3 [A] (single [32-bit real (~6 digit precision)])



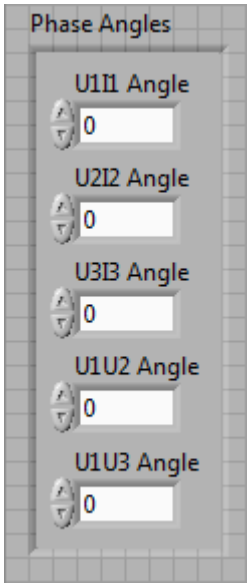
3.4.5. SetPhaseAngles.vi



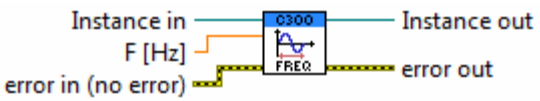
Setting the phase angles and angles between voltages.



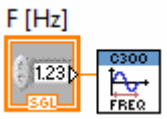
Phase Angles (cluster of 5 elements)
U1I1 Angle (single [32-bit real (~6 digit precision)])
U2I2 Angle (single [32-bit real (~6 digit precision)])
U3I3 Angle (single [32-bit real (~6 digit precision)])
U1U2 Angle (single [32-bit real (~6 digit precision)])
U1U3 Angle (single [32-bit real (~6 digit precision)])

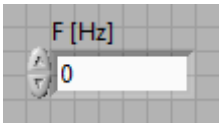


3.4.6. SetFrequency.vi



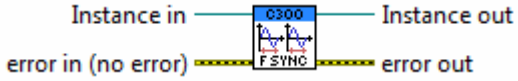
Setting the frequency of the output signal.





F [Hz] (single [32-bit real (~6 digit precision)])

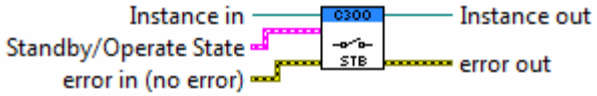
3.4.7. SynchronizeFrequency.vi



Enable frequency synchronization with the power network.

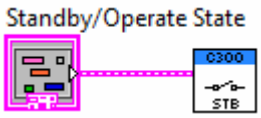
For disable the synchronization just send the new frequency by the *SetFrequency.vi* function.

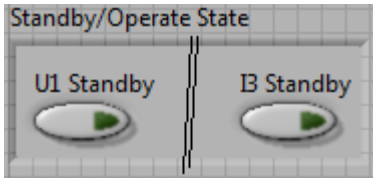
3.4.8. SetStanbyOperateState.vi



Sets the calibrator’s standby/operate state for each output channel.

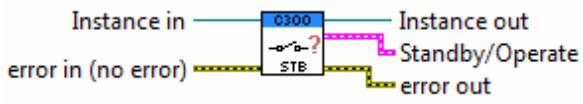
Boolean TRUE means, that output channel is turned off (Standby), and Boolean FALSE, that output is turned on (Operate).





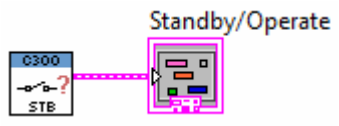
Standby/Operate State (cluster of 6 elements)
- U1 Standby (boolean (TRUE or FALSE))
- U2 Standby (boolean (TRUE or FALSE))
- U3 Standby (boolean (TRUE or FALSE))
- I1 Standby (boolean (TRUE or FALSE))
- I2 Standby (boolean (TRUE or FALSE))
- I3 Standby (boolean (TRUE or FALSE))

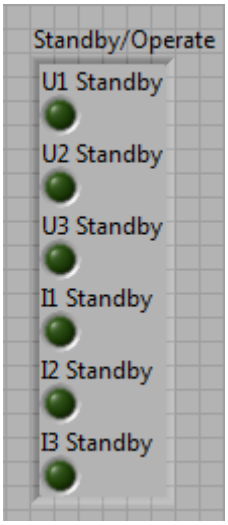
3.4.9. GetStandbyOperate.vi



Gets the calibrator's standby/operate state for each output channel.

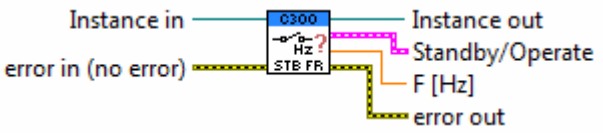
Boolean TRUE means that output channel is turned off (Standby), and Boolean FALSE that output is turned on (Operate).



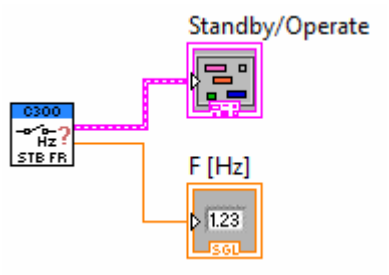


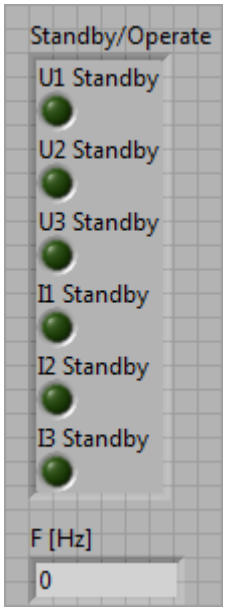
Standby/Operate (cluster of 6 elements)
- U1 Standby (boolean (TRUE or FALSE))
- U2 Standby (boolean (TRUE or FALSE))
- U3 Standby (boolean (TRUE or FALSE))
- I1 Standby (boolean (TRUE or FALSE))
- I2 Standby (boolean (TRUE or FALSE))
- I3 Standby (boolean (TRUE or FALSE))

3.4.10. GetStandbyOperateFrequency.vi



Gets the calibrator's standby/operate state for each output channel and power network frequency measurement from calibrator's internal frequency meter.

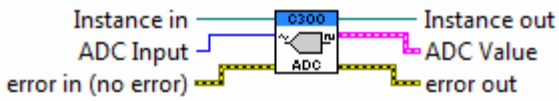




Standby/Operate (cluster of 6 elements)
U1 Standby (boolean (TRUE or FALSE))
U2 Standby (boolean (TRUE or FALSE))
U3 Standby (boolean (TRUE or FALSE))
I1 Standby (boolean (TRUE or FALSE))
I2 Standby (boolean (TRUE or FALSE))
I3 Standby (boolean (TRUE or FALSE))

F [Hz] (single [32-bit real (~6 digit precision)])

3.4.11. GetADCValue.vi



Gets the measurement from internal ADC for a given channel. It also provides information about measurements units.

The *ADC Input* can be one of the following values:

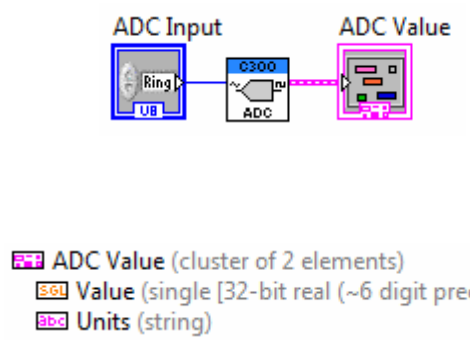
0 – RANGE_10V_DC

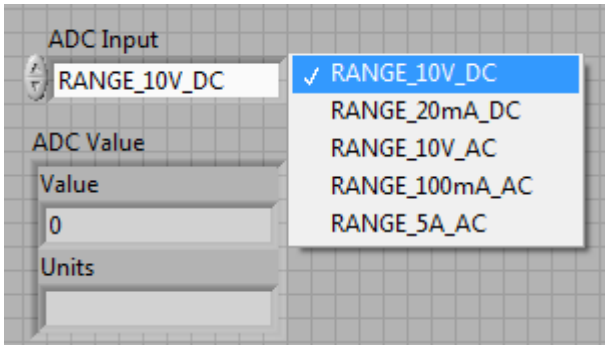
1 – RANGE_20mA_DC

2 – RANGE_10V_AC

3 – RANGE_100mA_AC

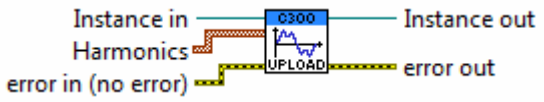
4 – RANGE_5A_AC





3.5. Harmonics

3.5.1 UploadHarmonics.vi



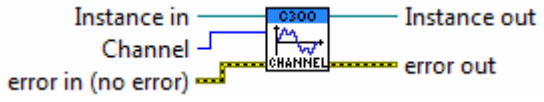
Uploads a new harmonics shape to the calibrators buffer.

This shape can be copied from buffer to the shape memory of individual calibrator outputs by the *SetChannelHarmonics.vi* function.

The *Harmonics* input expect an array of clusters with number, relative amplitude and angle of harmonics.

See *Calmet C300 Harmonics.vi* example how to use this function.

3.5.2. SetChannelHarmonics.vi



Sets channel harmonics signal shape, based on a shape uploaded to calibrator's buffer.

Before calling this VI you must upload harmonics to the calibrator's buffer (by calling *UploadHarmonics.vi*).

The *Channel* input can be one of the following values:

0 – DEFAULT_SIN

1 – U1

2 – U2

3 – U3

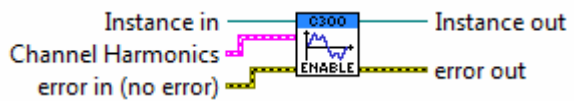
4 – I1

5 – I2

6 – I3

See *Calmet C300 Harmonics.vi* example how to use this function.

3.5.3. SetHarmonics.vi



Enables/disables channel harmonics. Before calling this VI you need to upload harmonics (calling *UploadHarmonics.vi*) to calibrator's buffer and call *SetChannelHarmonics.vi* to copy uploaded signal shape to the channel. Then you can enable or disable channel harmonics.

The *Channel Harmonics* input is an cluster of Boolean variables represents: U1, U2, U3, I1, I2, I3 channel, where TRUE means, that channel use its harmonics shape and FALSE, that use default sinusoidal shape.

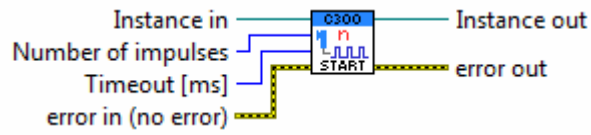
See *Calmet C300 Harmonics.vi* example how to use this function.

3.6. Photohead

3.6.1. StartPhotoheadImpulseCounter.vi

Starts the photohead impulse counter. You can read impulse counter values (number, time, status) by calling *GetImpulseCounterValues.vi*.

3.6.2. StartPhotoheadImpulseByNumber.vi

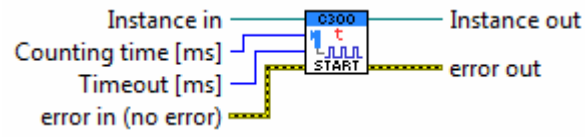


Start the photohead impulse counter to count a given number of impulses (*Number of impulses* input) and start counting a time in milliseconds from first to last impulse counted.

You can read impulse counter, time, and status by calling *GetImpulseCounterValues.vi*.

Timeout input set the maximum time in milliseconds between counted impulses.

3.6.3. StartPhotoheadImpulseByTime.vi

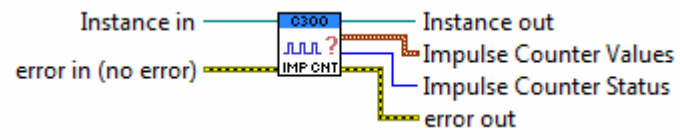


Start the photohead impulse counter to count the impulses in a given time in milliseconds (*Counting time* input) .

You can read impulse counter, time, and status by calling *GetImpulseCounterValues.vi*.

Timeout input set the maximum time in milliseconds between counted impulses.

3.6.4. GetImpulseCounterValues.vi



Gets the photohead *Impulse counter values* such as: number of counted impulses, measured time in milliseconds from 1st to last impulse counted, internal time in milliseconds of the test procedure, and *Impulse Counter Status*.

The *Impulse Counter Status* output can be one of the following values:

0 – NOT_READY (procedure is not finished yet, *Impulse Counter Values* are temporary)

1 – READY (procedure is finished)

2 – ERROR (timeout error occurs)

Remember, that if you want to correctly calculate the electricity meter error of full *n* turns of the rotor disc you must count *n+1* impulses.

3.6.5. StopPhotoheadImpulseCounter.vi

Stops the photohead impulse counter.

Function stops photoheads impulses counting and procedure time measurement.

You can read impulse counter, time, and status by calling *GetImpulseCounterValues.vi*.

3.7. Utilities

3.7.1. GetCalibratorInfo.vi

Gets the calibrator's description like firmware version and serial number.

3.7.2. VoltageRangeChangeNeeded.vi

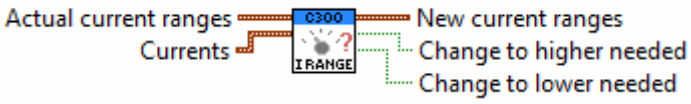
This VI provides information if a voltage range change is needed. It also computes the new voltage ranges.

Voltage ranges terminal input must be connected to the actual voltage ranges of the calibrator, and the *Voltages* input must be connected to voltages value demand.

If there is a need of voltage range change to higher on any channel, the *Change to higher needed* Boolean TRUE signal is asserted. If it is possible to switch voltage range to lower on any channel, the *Change to lower needed* Boolean TRUE signal is asserted.

Calculated voltage ranges from *New voltage ranges* output are always optimal and can be send to the calibrator by using *SetVoltageRanges.vi* function.

3.7.3. CurrentRangeChangeNeeded.vi



This VI provides information if a current range change is needed. It also computes the new current ranges.

Actual current ranges terminal input must be connected to the actual current ranges of the calibrator, and the *Currents* input must be connected to currents value demand.

If there is a need of current range change to higher on any channel, the *Change to higher needed* Boolean TRUE signal is asserted. If it is possible to switch current range to lower on any channel, the *Change to lower needed* Boolean TRUE signal is asserted.

Calculated current ranges from *New current ranges* output are always optimal and can be send to the calibrator by using *SetCurrentRanges.vi* function.

3.7.4. SampleDUT_VoltageMeter.vi

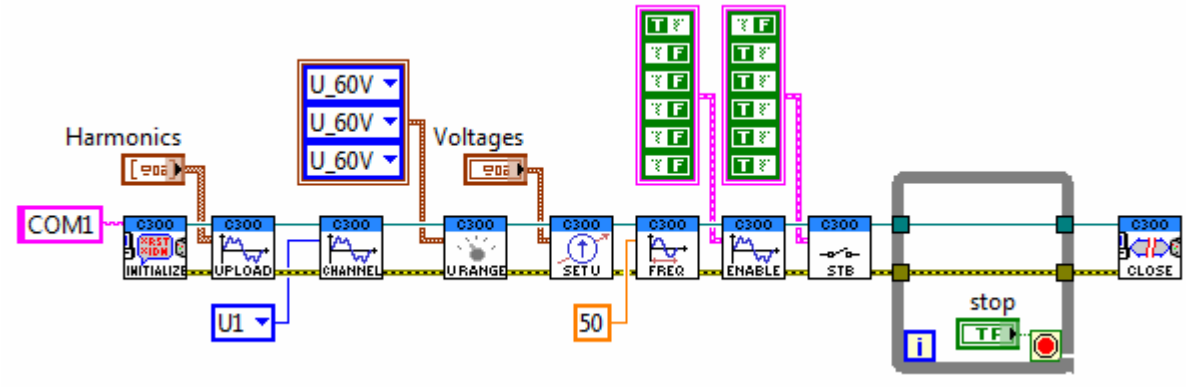
This is fake Device Under Test (DUT) provided only for driver testing purpose. It emulates a simple voltage meter with an measurement error.

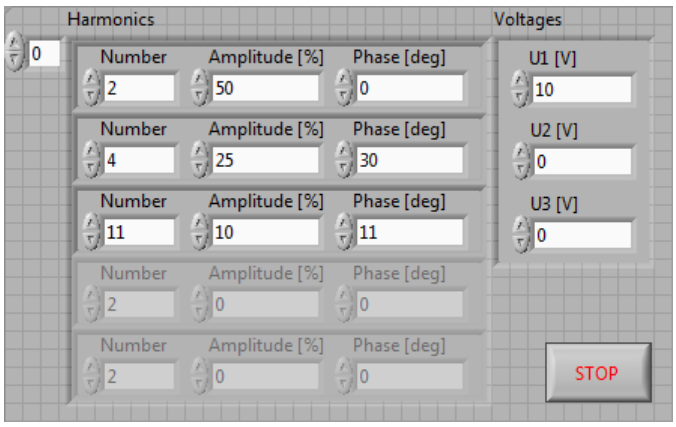
4. EXAMPLES

Examples for applications with C300 driver are placed in *Examples* folder of driver.

4.1. Generating a harmonics signals

Calmet C300 Harmonics.vi program demonstrate how to use LabVIEW C300 Driver to generate simple voltage with harmonics shape.

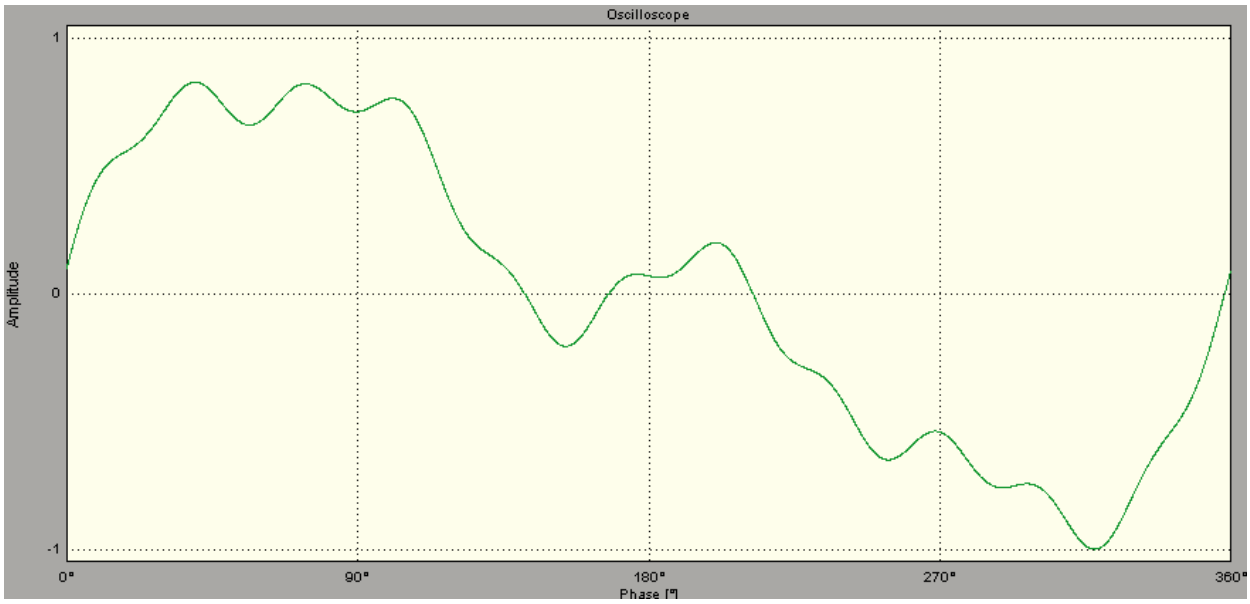




Program is going to connect with the calibrator by the *COM1* serial port. The harmonics shape defined in control table *Harmonics* is sending to the shape buffer and then copping to the U1 channel shape memory. After that, the voltage ranges (60V), values of voltages from *Voltages* table (10V for U1) and the base frequency (50Hz) of signal are sending. Next, the U1 channel is enabling for harmonics and turning in to the operate mode.

The U1 voltage is generating the harmonics signal as long as the *STOP* button is not pressed. If the *STOP* button is pressed, connection with the calibrator is closing and calibrator is going to automatically turning of all outputs in to the standby state.

For harmonics defined in the *Harmonics* table, output U1 voltage should have shape like on the picture below:

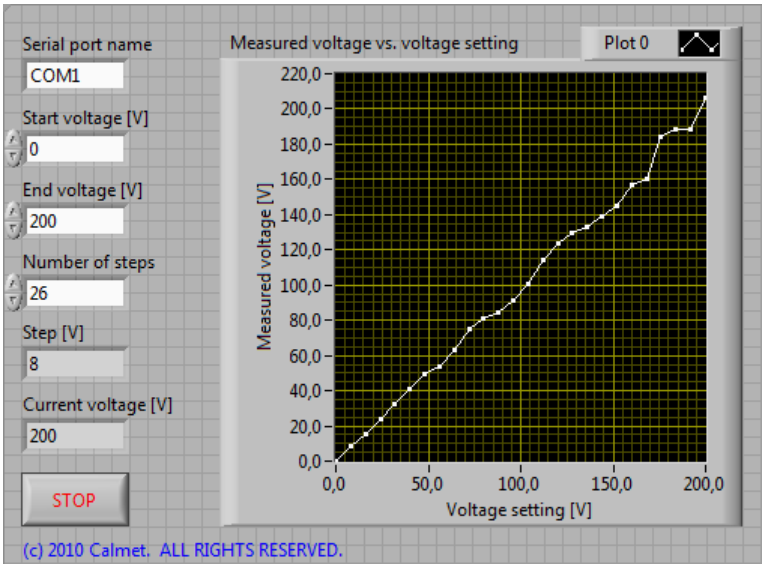


4.2. Automatic characteristics for voltage meter

For presenting how to perform measurement of electronic equipments characteristics, three example projects can be helpful. All examples realized the same task, but the ways how the measured points are determining are different.

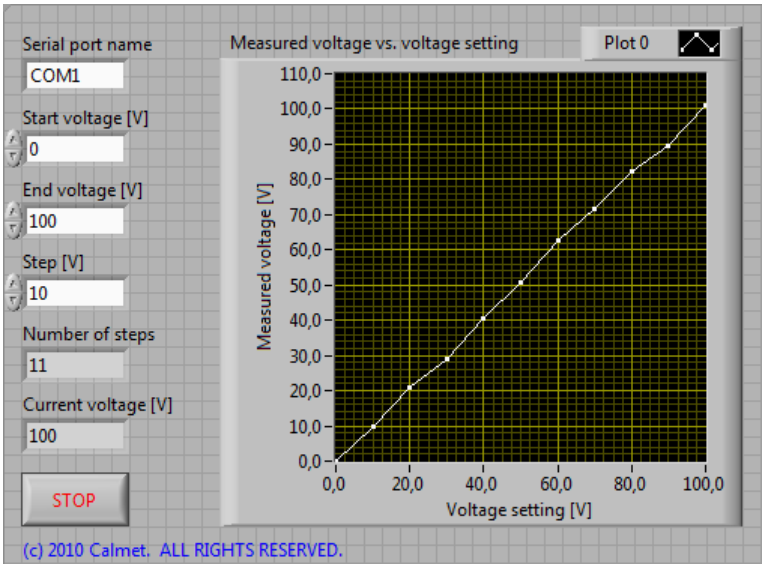
Calmet C300 Simple1ChVoltageTestProcedureByNumber.vi

First way is taking to characteristic constant number of measurement points from specific range of measured voltage.



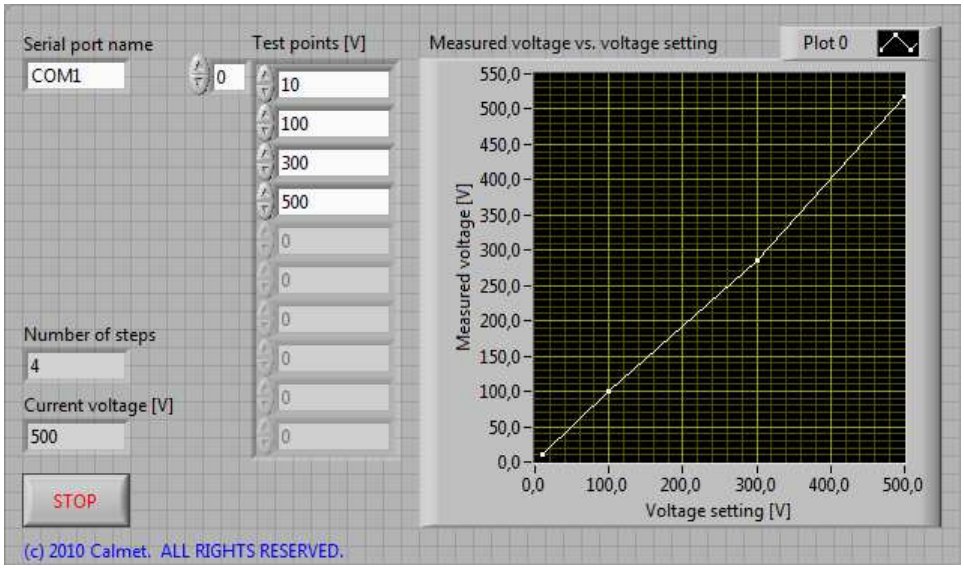
Calmet C300 Simple1ChVoltageTestProcedureByStep.vi

Second way is taking to characteristic measurement points calculated with constant step of voltage defined by user.



Calmet C300 Simple1ChVoltageTestProcedureByTable.vi

Last presented way is taking to characteristic measurement points defined bay user in table.



For use example programs in real measurement system, user have to use a virtual instrument, which takes one measurement from Device Under Test. Virtual instrument created by user should be placed instead of DUT in example program. It is necessary to initialize DUT instrument before *for loop* in program and closing communication with DUT after *for loop*.