1 Introduction

Omega software is a tool for

- Controlling instruments to acquire data
- Customizing and automating measurements
- Plotting data on screen
- Saving data to disk

Omega software consist of

- Manuals. For latest versions read it online at http://www.norecs.com/Omega/Omega.html or downloaded from http://www.norecs.com/index.php?page=Software -> Omega
- Executable file, which can be downloaded from http://www.norecs.com/index.php?page=Software -> Omega
- Physical items such as license dongle and communication cables, which are delivered to the end user.

Key points

- Upon purchase, the software license and cables are delivered, the executable and manual is to be downloaded by the user at the mentioned address
- It is important to read the 'Introduction' and 'Getting started' parts of the manual first

1.1 Getting Started

Getting started chapter is intended to be read in the order it is here presented. It will introduce the key aspects and concept of the preparational phase, and cover the acts required for software installation. Later chapters cover the operational features of the software.

1.2 Unpacking

Standard delivery of Omega may contain:

Name	Occurrence	Physical description	Functional description
Documents	Always	Folded paper	Intro and important links
USB license dongle	Always	Blue USB plug with key tag	License for Omega
USB-RS232	Optional	White/Blue USB cable	USB port to furnace
Null modem	Obsolete	Gray adapter	Extension for above item
M/F adapter	Obsolete	DB-9 adapter	Extension for above item
USB-RS485	Optional	Black USB adapter	USB port to furnace
USB A-B cable	Optional	Black 3m USB cable	Extension to above item
Prologix	Optional	Yellow box	USB to GPIB adapter
USB A-B cable	Optional	Black 3m USB cable	Extension to above item
LAN cable	Optional	CAT-6 LAN cable	Computer to instrument
BEAD thermistor	Optional	Small bead in plastic bag	Can be used for thermocouple cold junction compensation

Each customer may use Omega for varying thigs, so not everyone gets all the items.

1.3 System requirements

Computer requirements

- Space required by the program is only ~10Mb. Additional space is required for the data saved on disk
- User interface requires a mouse
- Minimum of 1024 x 768 screen resolution, higher resolution is recommended
- Free USB ports. Required number depends on the purpose the software will be used for. In case amount of free USB ports is not sufficient, acquire USB hub to split one port into many.

Omega is written in native Windows code, so it does not require any prerequisites or updates of the system.

Distribution of the software is a single executable file. There is no installation process.

1.4 Program location

The software will create folders and write files to disk. On the first execution of the software it will ask for a 'data location' where to save it's data. The program writes relatively often to disk, so it is recommended to avoid network driver or external drives. The data location will not be asked again (unless the executable is ran from another location). If required, the 'data location' can be reset from the tools/about tab.

The 'data save location' is the only thing Omega saves in the computer registry. It is saved under:
HKEY_CURRENT_USER\Software\NorECsAS\Omega\Omega2At\

Be advised that later versions of Windows restrict software writing to disk in specific folders (for example "Program files", and for user accounts that are not 'Administrators'. It is therefore not possible to use Omega from "Program files".

1.5 Omega software license

Omega software license

The Omega software is free to be copied, distributed and used as it is without the license. The software will not interact with measurement instruments without the USB license dongle installed. It is advisable to accuire a copy of the software prior to purchase or delivery to test it's features.

One license dongle is provided with the software, and it entitles the for any future updates. Plug in the USB license dongle to a USB port on the computer that is meant to be used with the instruments. The dongle will be recognized automatically on all Windows versions. If not, see troubleshooting section for further advice.

A purchased copy of the software also comes with assortment of cables and adapters explained elsewhere.

This arrangement allows one computer to run Omega in the purpose of data aquisition and measurements control, but unlimited copies of the software can be ran on other computers to for inspecting the data collected.

The latest version of the software can be downloaded at NORECS.COM website:
http://www.norecs.com/index.php?page=Software

The main purpose of this software is data aquisition by operating external instruments. In this role the software interfaces to a number of external instruments, and sends instructions to these instruments. Any damage caused to and/or by these instruments is sole responsibility of the end user regardless of the cause. A good measurement design setup copes in a safe way with technical failures such as instrument failures, electricity blackouts, software errors and so forth. A good setup has independent ventilation, alarms for fire, flammable and toxic gases, cut off limiters and so forth. The software has no intelligence of it's own. If it is instructed to enable flow of gas for mass flow controllers assigned to hydrogen and oxygen, and then to ignite spark with power supply, it will do it.

Furthermore to be acknowledged, the software is complicated, difficult, and it has bugs. It has steep learning curve. However, it has unique set of features which make it extremely useful for the niche purpose it has; high temperature electrochemical studies such that are done with Probostat sample holder or similar.

1.6 Mode of operation

Omega can be used in two ways; measuring with instruments and/or reviewing the results.

A copy of the software that is used to measure, located for example in a computer in the lab. All the hardware related topics in this manual refer to this case.

Another copy, or copies, can be used elsewhere to plan or review measurements. This does not require any hardware other than a computer. In this mode the software does not even need a license. All hardware related topics in this manual can be ignored for such use.

The program is free to be copied and ran in other computers, such as home or office computers. It is thus possible to further review the saved measurements, and tune or create the plots and graphs in the comfort of a proper working desk, or finalize publishable graphs or study relations between the measured properties using the math parser.

When inter-using same measurement files on separate computers, it is adisable to have same regional settings on both computers. Measurement (.mea) files are saved and opened as text, and the part of the file that saves the graphs, will save their labels etc. as well, in plain text. Different codepage or regional settings results in some cases to corruption of the plots (clors, names...) but not the raw data.

It is recommended to familiarize with the software and the saved examples on an office computer before trying the software in the lab.

Running the program to review and process measurements will work on all Windows operating systems.

Running the program to measure will work in Windows XP or later.

1.7 GPIB Network

Plug in the Prologix USB-GPIB converter to a USB port with the USB A-B cable shipped with the software. The operating system will automatically recognize the device as Virtual COM port (It is possible to see the COM port appear and disappear from device manager in the control panel when instrument is plug in or out). If not, refer to chapter manual installation of GPIB converter.

Connect the USB-GPIB converter to a GPIB instrument, or GPIB instrument network. If using more than one instrument, all GPIB instruments must be connected to each other with GPIB cables and the converter must connect to that network of devices. For more information on GPIB networks refer to chapter GPIB network considerations.

One or more instruments interconnected by GPIB cables. Each instrument must have unique address between 1 and 32 that specifies that instrument in the network. Some instruments require the address to be even, or odd, or the trailing address to be unused. Refer to instrument manuals. For older instruments it is safest not to use consequent numbers when operating with aged GPIB instruments, but addresses in increments of 2.

1.8 GPIB Instruments

Due to various reasons the instruments may or may not be be preset to correct settings.

For example to use Keithley 2000, the following settings should be applied. This can be done from the front panel of the instrument.

- Make sure the instrument is in 'GPIB ON' (and not in RS232 mode). Press shift and digits, arrows to navigate, enter to apply, exit to cancel.
- To make the search faster, change the GPIB address of the instrument to 1 (as consequence the instrument search only needs up to address 1, this makes it faster to try things).
- Select the language SCPI

Each instrument has different settings, some have none or the defaults are already suitable.

1.9 Furnaces

To connect to a furnace with Eurotherm controller, plug in the FTDI USB-RS232 converter to a free USB port, it will get automatically identified. Connect the other end of USB-RS232 converter to the furnace. For more information about connections to a furnace or a network of furnaces, refer to chapter Furnace connections.

1.10 Mass flow devices

Vögtlin smart series, and Bronkhorst el-flow series mass flow meters and controllers are supported in the software. However, the auto search does not support these instruments and they need to be manually inserted to the device configuration file.

1.11 Electrodes and cabling

Names vary greatly depending your source. When setting up a system one might refer to a scientific paper, application note, instrument manual, sample holder manual, software manual and so forth.

The various electrode contacts, wires, cables, plugs, sockets and instrument channels are called with different names in different places. Even within a single manual/paper the naming of these may alter, especially when considering separate measurement methods, and between measurement guides, instrument guides, sample holder guides and software guides the naming of such things is guaranteed to be conflicting.

Knowing and understanding what to connect where is essential, and this quick-reference table can be helpful in remembering or figuring out all the acronyms. It is also essential that the acronyms and names do not define what goes on in the cables and electrodes. The names are just concepts.

Context	Name	Explanation
BNC cable	Lead	BNC is a name for a cable with coaxial nature; inner lead (usually connected to electrode) and protective shield around the lead protecting the signal in the lead from interference.
	Shield	Shields may be grounded or not, connected together or kept separate. Sometimes an electrode can be connected to a shield. All depends of the instrument requirements and measurement settings.
Thermocouple compensation		Cable that acts just as the thermocouple of the same type, but within reduced temperature range, used to connect the instrument to the measurement cell.

Context	Name	Explanation
wire		
Sample	Electrode	A single lead electrode.
ProboStat	HC	High current electrode.
	HV	High voltage electrode.
	LV	Low voltage electrode.
	LC	Low current electrode.
	ILV	Inner low voltage. Alternative or additional to IV.
	ILC	Inner low current. Alternative or additional to IC.
Multimeter	Channel	Pair of electrodes. Often called High and Low, or + and -, or red and black.
	Channel pair	Two channels used together in modes measuring with 4 electrodes. Often called Input and Sense, corresponding to Current and Voltage accordingly.
	Lead	Connected between the electrode at the sample and the instrument terminal.
Impedance spectrometer	Terminal	A socket for electrode and shield to connect to. (Often BNC cable type)
	High current	H Cur, Generator output, I high
	High voltage	Voltage high, high potential, V high
	Low voltage	Voltage low, low potential, V low
	Low current	Current, Low current, I low, L current
	Guard	Contact for guard electrode

Possibly useful link covering the same subject:
http://www.norecs.com/index.php?page=148&faq=76&open=11

1.12 Basic concepts

User can define a measurement from basic blocks called nodes, which (mostly) in turn inform the correct instrument to perform the correct task. The nodes blocks can measure things such as impedance, voltage, current, resistance, temperature and flow. In addition the nodes can contain instructions for the instruments to perform, such as measurement precision settings, new target temperature for the furnace, new gas flow setpoint for flowmeter, new amount of voltage to apply and so forth. These instructions can be static or dynamic. The dynamic instructions can be as simple as functions of time, or complex and conditional structures as multipart n^{th} degree splines, but the main division is that nodes either measure or do a task that does not receive any result. The measured data is stored in the memory as well as on a file, and user can choose what parts of the data is plotted on the graphs. Omega is developed for measurements with ProboStat™ sample holder, but can just as well be used for any type of electrical measurements that can be achieved with the supported instruments.

Typical uses include, but not restricted to:

- AC and DC resistance/conductance measurements with 2, 3 or 4 points
- Impedance spectroscopy
- Breakdown of impedance values to components
- Temperature measurement and control of setpoint and ramp rate from/to furnaces
- Flow measurement and control from/to mass flow controllers
- Temperature measurements with thermocouples and thermistors using multimeters
- DC voltage measurement (for EMF transport number, fuel cell test, Seebeck coefficient, oxygen sensors, thermocouples, etc.)
- DC current measurements (pressure sensors, flowmeters and in general all sorts of sensors with analog outputs)
- DC current control (electrochemical cells, heaters, coulometric titration, motors, pumps, etc.)
- Commanding signal switching instruments (multiple input sources, rotating electrodes for van der Pauw measurement, reverse and average voltage measurements to cancel thermal voltages etc.)

- Plot measured raw data or use mathematical expressions and functions such as solve van der Pauw or convert (thermocouple) voltages to temperatures
- Saving graphs as images with publishable quality
- Exporting data in csv format, and impedance results as Z-View or Equilibrium circuit format

Omega can control GPIB devices such Solartron, Agilent, Hewlett-Packard, Novocontrol etc. and serial port devices such as Eurotherm 2216 or Vögtlin Smart mass flow controllers.

The software has auto-search feature to find attached devices automatically (when devices support this feature) eliminating the need for tedious manual configuring, and possibility to save and load the configuration files.

Measurements in Omega can go on for as long as you need them to, even for months. There is no limit for amount of simultaneous measurements, so a conductivity measurement can run indefinitely indicating the equilibriums while the instrument can measure other things such as sweeps meanwhile.

1.13 About learning curve and errors

In general it is easy to make a program when it's purpose is well defined. This software is not one of those. Omega is vague, and very flexible. This requires high level of abstraction in it's design. This means it may feel unnecessary difficult for a specific purpose, and compared to a program that only performs that one purpose, it is. The good side is that enlightened user may perferm almsot anything with Omega.

The program has been in use for years and all critical problems have been eliminated, however, minor bugs remain. They mainly relate to user actions, so it is possible to learn what actions cause these warnings. The warning will bring out a pop-up with a buttons "Continue application" and "Close application". Most of the times this is nothing serious and it is safe to keep using the software.

Normally the information in this "bug report" can be ignored, as it gets emailed to NorECs if network is available, or saved to disk when not.

In general it seems that using one file, removing and and adding nodes and graph-related entities to the same file over and over again can cause the file to have artifacts that will cause these errors. Starting new, fresh files for new measurements is a good policy in any case.

1.14 Resources

Resources:

http://www.norecs.com/index.php?page=Application+notes

http://www.norecs.com/work/_files/__PUBLIC_FILES__/software/Omega/

2 Omega features

The features and concepts of Omega software.

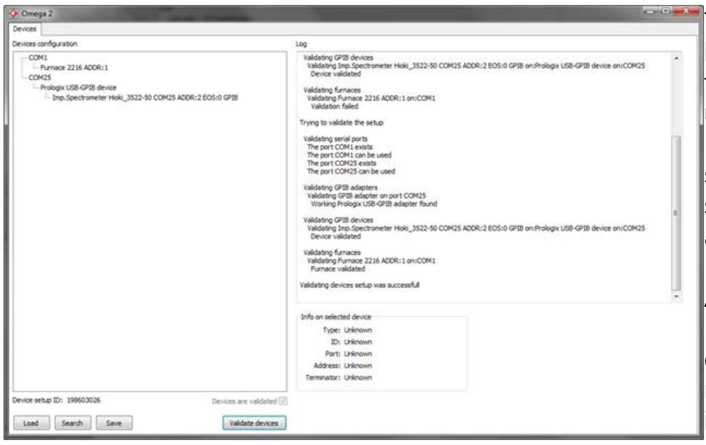
2.1 Introduction

Omega can control a range of instruments, use them to perform operations such as measurements, in a sequence.

A focus is given to one instrument at a time to perform desired measurement or task and this moment of focus is called node. Each measurement can contain any number of nodes and the nodes are performed from beginning to end and then repeated again.

In a simple measurement a node would query temperature from a furnace and another node would measure sample conductivity. The retrieved data is collected and indexed so that each loop of the measurement has data from all the nodes.
All the data can also be plotted on graphs real time, and all the collected data is also available to be referred to in the internal logic of the software; the software is aware of the data it is collecting. This allows some powerful possibilities for the advanced user. It is possible to automate actions based on the values measured.

2.2 Devices tab



```
COM1
  Furnace 2216 ADDR:1
COM25
  Prologix USB-GPIB device
    Imp.Spectrometer Hioki_3522-50 COM25 ADDR:2 EOS:0 GPIB
```

```
 Validating GPIB devices
  Validating Imp.Spectrometer Hioki_3522-50 COM25 ADDR:2 EOS:0 GPIB on:Prologix USB-GPIB device on:COM25
   Device validated

 Validating furnaces
  Validating Furnace 2216 ADDR:1 on:COM1
   Validation failed

Trying to validate the setup

 Validating serial ports
  The port COM1 exists
  The port COM1 can be used
  The port COM25 exists
  The port COM25 can be used

 Validating GPIB adapters
  Validating GPIB adapter on port COM25
   Working Prologix USB-GPIB adapter found

 Validating GPIB devices
  Validating Imp.Spectrometer Hioki_3522-50 COM25 ADDR:2 EOS:0 GPIB on:Prologix USB-GPIB device on:COM25
   Device validated

 Validating furnaces
  Validating Furnace 2216 ADDR:1 on:COM1
   Furnace validated

Validating devices setup was successfull
```

```
Info on selected device
      Type: Unknown
        ID: Unknown
      Port: Unknown
   Address: Unknown
Terminator: Unknown

Device setup ID: 198603026          Devices are validated ✓
```

To be able to measure, the software needs to know exactly what instruments are connected to the computer and how. Devices tab is a location used to find out and manage those setups. Features include load setup, autosearch setup, save setup and validate setup. Information on the search and validation process gets shown in the area on the right.

Any setup can be saved to disk and loaded back later. When loaded, it needs to be validated in order for it to be able measure. It is possible to manually edit the device setup files to add instruments that fail the auto search feature or ones that have not yet been added to the software. The device setup files (.odc) are in fact just text files to be opened in notepad or similar. Examples section contains number of files for reference.

For Omega to be able to measure, a validated devices setup is required. For existing measurement to be able to continue, exactly the same instrument setup is needed to be valid. This is handled by comparing 'assigned device setup' and 'required device setup'.

The connected devices can be either scanned automatically, or loaded from a previously saved file. In case the setup is loaded from a file it also needs to be verified so the software knows that the loaded setup really exists and match the assumption (loaded data) perfectly. Verifying a setup takes significantly less time than a full scan, so it is wise to keep a saved file of each favorite setup.

Some(older) instruments may not be possible to add through autosearch. In such cases they possibly can be added manually by edting the devices setup file.

On the program startup a default setup is automatically loaded if a file called ‘default.odc’ exists.

After loading a setup from a disk the setup needs to be validated. To do so click the validate button. If the loaded configuration can be verified by the validation process, the setup is marked as valid ('Devices are validated' checkbox) and the setup can be used for measurements. In case the validation (or the search) fails, it is recommended to retry few times before considering other measures.

Each measurement keeps track of what setup was used to measure. Continuing an old measurement requires the exact same devices setup to be present and valid that was used to measure the data in the first place. Each device setup has unique identification (ID) number which must match with the number stored in the measurement. If the current setup ID number does not match the setup ID of the measurement, the measurement will not measure.

It is possible to force-update the measurements setup ID to the current setup ID, to force the measurement to continue. To do this use the 'Update to current' button on measurements tab. If any of the previously used instruments is missing or on different port or address, things will fail. Force-updating is useful if nothing was removed from the device setup, but just added instead.

Each found instrument is automatically assigned a role; the role can be impedance spectrometer, multimeter, furnace, mass flow controller or unknown. The role determines the type of tasks the instrument can be used to measure or perform. During the search or validation of devices setup, the GPIB instruments get initialized with settings that are tested to work with the software. In case it happens the program one time stops working flawlessly with some instrument, it is best to close the program, unplug the GPIB converter and the USB-RS232, turn off all the instruments on the GPIB network for a brief moment and back on again, connect everything back on, start the program, load the devices setup and validate it, possibly adding a computer restart to the process as well.

The USB-GPIB and the USB-RS232 adapters will show as COM ports on the computer. If these are not found automatically, it is possible to go to device manager on windows (Windows key + R, then type devmgmt.msc and hit enter). Then expand the Ports-section and see what ports are listed.

2.3 Devices configuration files

Omega needs to know exactly what instruments are connected. These datails are called the device configuration and they can be saved as a file. The filename extension *.odc but it is a normal text file that can be edited with for example notepad. These files are kept in "Device Setups" folder under the Omega installation folder. File named default.odc will be automatically loaded to Omega on startup.

File definitions

Example of *.odc file

```
COM1
	Eurotherm 2216 COM1 ADDR:1
COM33
	Prologix USB-GPIB device
		Multimeter K2000 COM33 ADDR:4 EOS:0 GPIB here_is_something
		Multimeter PowersupplyE3642A COM33 ADDR:5 EOS:0 GPIB here_is_something
```

Spaces in the beginnings of the lines are 0, 1 or 2 tabulators. All the remaining gaps are spaces.

```
Serial_port_name
	Eurotherm Model Serial_port_name Modbuss_address
Serial_port_name
	Prologix USB-GPIB device
		Instrument_type Instrument_id Serial_port_name GPIB_address String_terminator Validation_type ID_string_from_Instrument
```

Instrument type

- U-role
- Multimeter
- Imp.Spectrometer
- PotGal

The software must validate any devices setup to know that every instrument is present as defined. No measurement can run if the setup was not successfully validated.

GPIB instrument validation can be bypassed entering **** instead og GPIB as Validation_type in the devices file.

Methods

Modbus over serial port to Vogtlin mass flow controllers
DDE server for Bronkhorst mass flow controllers
GPIB over serial port to GPIB instruments

Support for un-cooperative instruments

Old instruments do not indentify themselves and cannot be found using autosearch. They need to be entered manually into the file.

Example items for un-foundable instruments

PotGal Solartron1287 COM44 ADDR:6 EOS:0 **** None

Manual testing of instruments

Once instrument has been added to the devices file, it can and should be tested with the tools section GPIB tool. Select the instrument in question and send some GPIB command to it that you know will reply. Such command can be for example *IDN? in general or instrument that does not support *IDN? see the device manual.

1287 "?VN" → Returns some numbers (valid command for 1287)

2.4 Autosearch

Known instruments reserving consequent addresses

Some instruments reserve consequent address(es) after the primary address for outputting data to plotter devices. That technique is obsolete, but theose addresses still need to be reserved to avoid network failure.

Example: Device address is 6, and it automatically outputs all results to address 7. If another normal instrument has address 7, it will receive a load of communication that it can not make any sense of, and will be unusable. Most likely whole network will not work.

The information of using additional addresses is not necessarily in the manual. It cannot even easily be tested in some cases. For this reason here is a list of known instruments using 'talker' addresses.

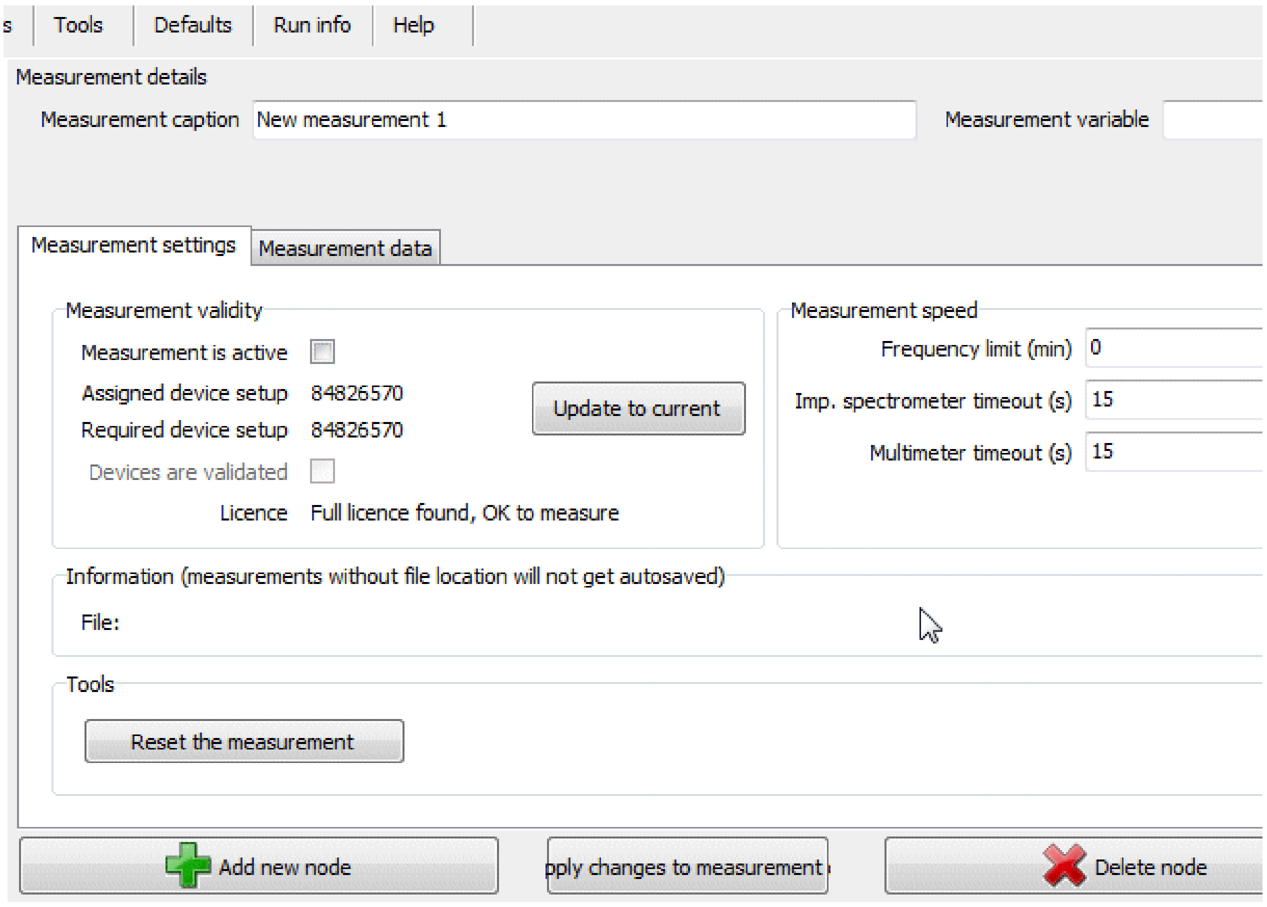
- Solartron 1260 (FRA)
- Solartron 1287 (PotGal)
- PAR 273 (PotGal)
- HP 4192A (FRA)

2.5 Measurements tab

All measurement related aspects are managed on the measurements tab. Measurements can be created, loaded, saved, deleted and modified here.

Measurements consist of nodes, and similarly the nodes can be edited at this location. It is also possible to view, manipulate and export the measured raw data on this tab under node data.

Button 'Save all' will save all open measurements and take a note of which measurements are open. Button 'Open all' will check this note and try to open the corresponding measurements. Measurements not saved at least once, will not be autosaved either.



Measurement details

Measurement caption: New measurement 1
Measurement variable:

Measurement validity

Measurement is active	☐
Assigned device setup	84826570
Required device setup	84826570
Devices are validated	☐
Licence	Full licence found, OK to measure

Measurement speed

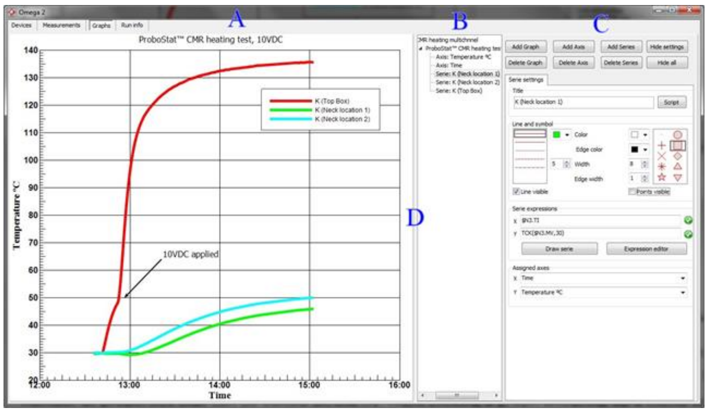
Frequency limit (min)	0
Imp. spectrometer timeout (s)	15
Multimeter timeout (s)	15

Information (measurements without file location will not get autosaved)

File:

Tools

2.6 Graphs tab



The graphs tab allows the user to plot anything based on the measured data, mathematical expressions, list of predefined functions or any combination of them.
Measurement may have any number of graphs and a graph may have any number of axes, series and series markers. Graphs, axes, series and markers have a number of properties that can be edited on the fly.

Right clicking a graph (in area A) will bring up a menu with additional tools and image & data export options. Clicking items in area B brings out their properties for editing in area C. In case multiple graphs, graphs share the area A, and are displayed in graph title order. To distribute all graphs evenly on screen, click a measurement in the area B, to highlight a specific graph, click that graph on area B.

To claim more space for area A, use Hide settings and Hide all buttons, as they will expand the area reserved for displaying the graphs. Dragging from the vertical splitter (D) on the right side of the graph will bring back the hidden settings console.

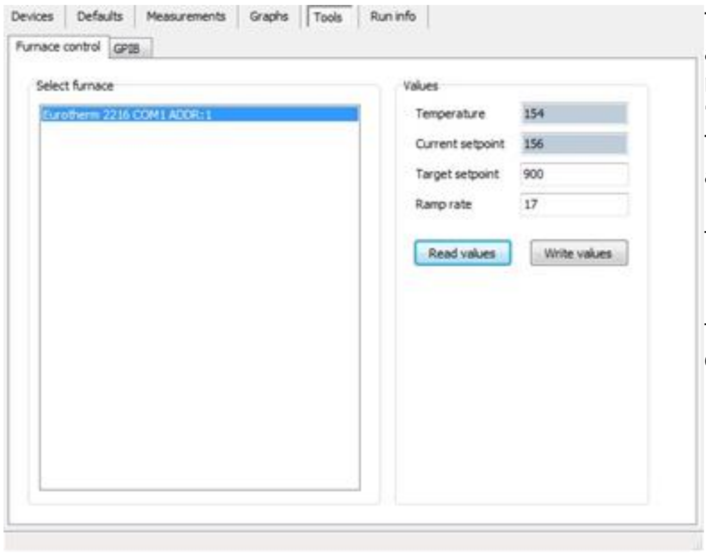
In case of multiple graphs, press shift and left click on a graph will focus on that graph and show the other graphs as small thumbnails. To zoom in on the graph area, left click and drag the mouse pointer. To un-zoom, left click outside the axes. Right clicking a measurement will bring up a pop-up menu.

2.7 Tools tab

Set of useful tools, that mostly should only be used only when no measurements are running.

Using these tools is not normally required for typical use of the software.

2.8 Furnace



Select furnace
Eurotherm 2216 COM1 ADDR:1

Values

Temperature	154
Current setpoint	156
Target setpoint	900
Ramp rate	17

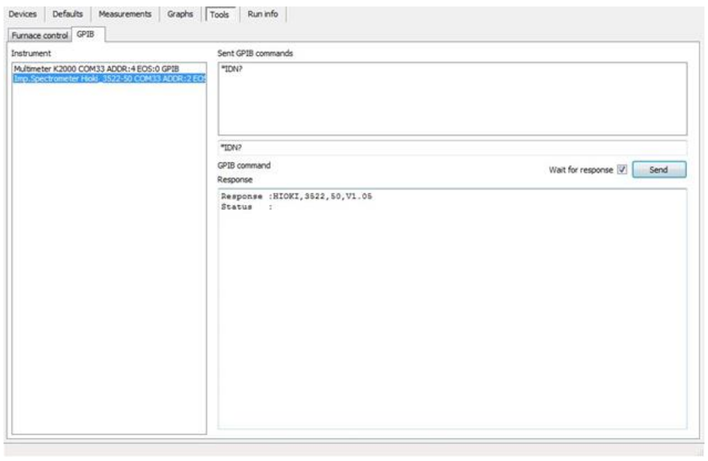
The furnace control tab allows the user to read and write furnace values. Grayed out fields are read only and will not be sent to the furnace on 'write values' event.
The units of temperature, setpoint and ramp rate are specific to other settings of the furnace. Please refer to the chapter about furnace control for more specific details.

Note, that this tool should not be used when there are active measurements running that may communicate with the furnace.

2.9 GPIB tool

The GPIB tool allows user to send GPIB commands (and possibly Prologix control commands) to the instruments. This tool is meant to be used to design and test unusual instrument settings, and is never to be used during the measurements.

Note, that this tool should not be used when there are active measurements running that may communicate with the furnace.



Multimeter K2000 COM33 ADDR:4 EOS:0 GPIB
Imp.Spectrometer Hioki_3522-50 COM33 ADDR:2 EO

```
*IDN?
```

```
*IDN?
```

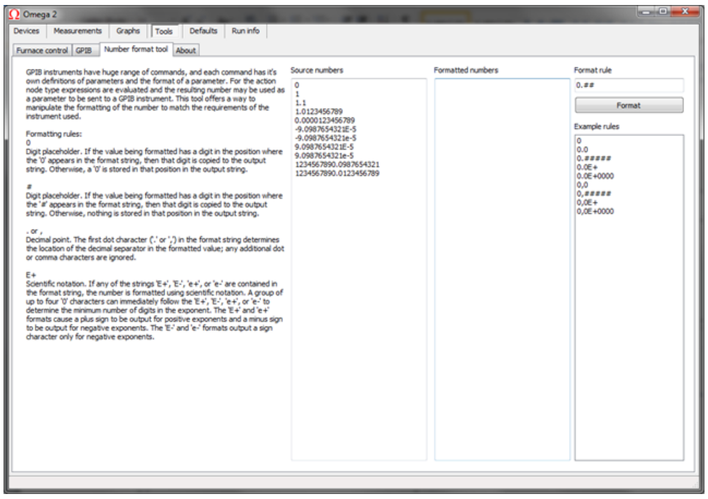
```
Response :HIOKI,3522,50,V1.05
Status   :
```

2.10 Number format tool

To control the GPIB instruments the software sends them instructions, typically text and numbers. Mostly each instrument or at least instruments across one manufacturer have individual rules in what sort of format the numbers needs to be sent in...

It is not obvious, but the amount of ways to represent value 1 is infinite.

1
1.0
1,0
1.00
1,00
01
001
1E0
1.0E0
1E01
1E+1



GPIB instruments have huge range of commands, and each command has it's own definitions of parameters and the format of a parameter. For the action node type expressions are evaluated and the resulting number may be used as a parameter to be sent to a GPIB instrument. This tool offers a way to manipulate the formatting of the number to match the requirements of the instrument used.

Formatting rules:
0
Digit placeholder. If the value being formatted has a digit in the position where the '0' appears in the format string, then that digit is copied to the output string. Otherwise, a '0' is stored in that position in the output string.

#
Digit placeholder. If the value being formatted has a digit in the position where the '#' appears in the format string, then that digit is copied to the output string. Otherwise, nothing is stored in that position in the output string.

. or ,
Decimal point. The first dot character ('.' or ',') in the format string determines the location of the decimal separator in the formatted value; any additional dot or comma characters are ignored.

E+
Scientific notation. If any of the strings 'E+', 'E-', 'e+', or 'e-' are contained in the format string, the number is formatted using scientific notation. A group of up to four '0' characters can immediately follow the 'E+', 'E-', 'e+', or 'e-' to determine the minimum number of digits in the exponent. The 'E+' and 'e+' formats cause a plus sign to be output for positive exponents and a minus sign to be output for negative exponents. The 'E-' and 'e-' formats output a sign character only for negative exponents.

Source numbers
```
0
1
1.1
1.0123456789
0.0000123456789
-9.0987654321E-5
-9.0987654321e-5
9.0987654321E-5
9.0987654321e-5
1234567890.0987654321
1234567890.0123456789
```

Formatted numbers

Format rule
```
0.##
```

Example rules
```
0
0.0
0.#####
0.0E+
0.0E+0000
0,0
0,#####
0,0E+
0,0E+0000
```

Although these mean same, they are all different to the instrument and to the software, and when those two communicate, they need to agree on the rules how things are represented.

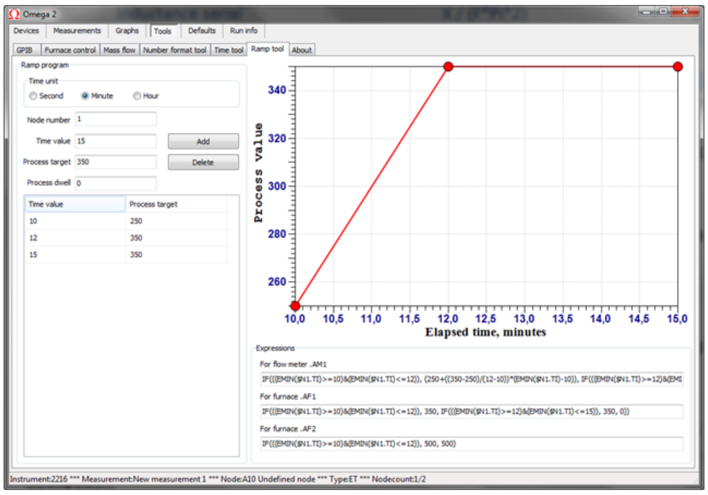
This tool helps the user to understand and design the number formatting rules for cases when they need to send their own, possibly dynamic commands to the instruments.

‘Source numbers’ is a list of numbers that will get formatted using the ‘format rule’ on the click of the ‘format’ button. Resulting formations are listed in the ‘formatted numbers’.

2.11 Ramp tool

The ramp program tool allows user to define program of process value as function of, for exaple time. The desired ramp-program is converted to expressions that are ready to be used with the mass flow controller AU node or with the furnace control AU node.

‘Time unit’ defines the unit of the X axis. ‘Node number’ identifies the node that is used in the expression to define the elapsed time of the measurement. ‘Add’ and ‘Delete’ buttons add to end of the program a new point, the values of fields ‘time value’ and ‘process target’. ‘Process dwell’ defines what the process target value is after the ramp program has finished.



Time unit: Second [✓ Minute] Hour

Node number: 1

Time value: 15

Process target: 350

Process dwell: 0

Time value	Process target
10	250
12	350
15	350

For flow meter .AM1

```
IF(((EMIN($N1.TI)>=10)&(EMIN($N1.TI)<=12)), (250+((350-250)/(12-10))*(EMIN($N1.TI)-10)), IF(((EMIN($N1.TI)>=12)&(EMI
```

For furnace .AF1

```
IF(((EMIN($N1.TI)>=10)&(EMIN($N1.TI)<=12)), 350, IF(((EMIN($N1.TI)>=12)&(EMIN($N1.TI)<=15)), 350, 0))
```

For furnace .AF2

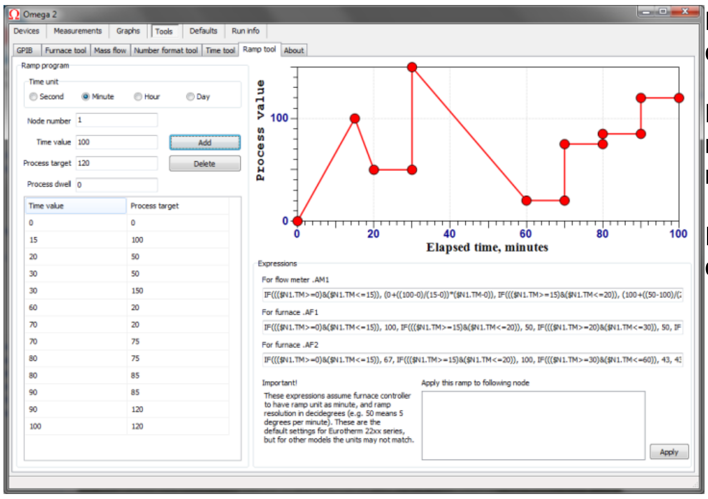
```
IF(((EMIN($N1.TI)>=10)&(EMIN($N1.TI)<=12)), 500, 500)
```

```
Instrument:2216 *** Measurement:New measurement 1 *** Node:A10 Undefined node *** Type:ET *** Nodecount:1/2
```

The ramp program is visible as list of points and as graphical representation.

- Expression .AM1 is to be copied to AU node type mass flow control.
- Expression .AF1 signifies the ‘furnace setpoint’ and is to be copied to AU node type furnace control expression .AF1.
- Expression .AF2 signifies the ‘furnace ramp rate’ and is to be copied to AU node type furnace control expression .AF1.

Expressions can be applied to existing AU type nodes by selecting a node from the list on bottom right corner and clicking apply. The safety values for the expressions are automatically updated also.



Ramp program

Time unit: Second [✓ Minute] Hour Day

Node number	1
Time value	100
Process target	120
Process dwell	0

Time value	Process target
0	0
15	100
20	50
30	50
30	150
60	20
70	20
70	75
80	75
80	85
90	85
90	120
100	120

Expressions

For flow meter .AM1

```
IF((($N1.TM>=0)&($N1.TM<=15)), (0+((100-0)/(15-0))*($N1.TM-0)), IF((($N1.TM>=15)&($N1.TM<=20)), (100+((50-100)/(
```

For furnace .AF1

```
IF((($N1.TM>=0)&($N1.TM<=15)), 100, IF((($N1.TM>=15)&($N1.TM<=20)), 50, IF((($N1.TM>=20)&($N1.TM<=30)), 50, IF
```

For furnace .AF2

```
IF((($N1.TM>=0)&($N1.TM<=15)), 67, IF((($N1.TM>=15)&($N1.TM<=20)), 100, IF((($N1.TM>=30)&($N1.TM<=60)), 43, 4
```

Important!

These expressions assume furnace controller to have ramp unit as minute, and ramp resolution in decidegrees (e.g. 50 means 5 degrees per minute). These are the default settings for Eurotherm 22xx series, but for other models the units may not match.

Apply this ramp to following node

Ramp programs for furnaces and mass flow controllers can be designed with the 'ramp tool'.

Points are added and deleted quickly and the resulting expressions can be easily applied to the relevant instruments.

It is possible to control multiple instrument each of with their own ramp programs.

2.12 Mass flow tool

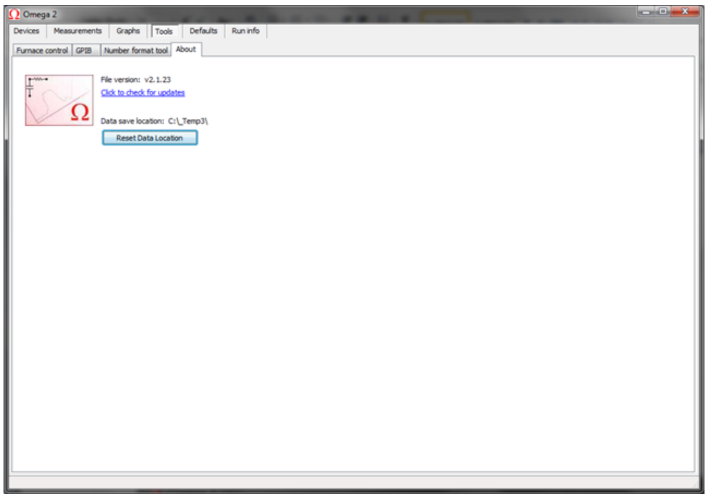
With this tool it is possible to manually control connected Red-y mass flow meters and controllers from Vögtlin. Also very rudimentary control for Bronkhorst mass flow meters and controllers when their FlowDDE server is running.

2.13 Time tool

Helps user to convert real world times to a Delphi-format number that can be used in expressions to perform real time (as opposed to elapsed measurement time) dependent actions. These numbers can be operated against the $N1.TI which is the current time of the node in this Delphi-format.

2.14 About

About tab displays information on the program version and clickable link to check for updates. The default data location is also shown and a button to reset it.



File version: v2.1.23

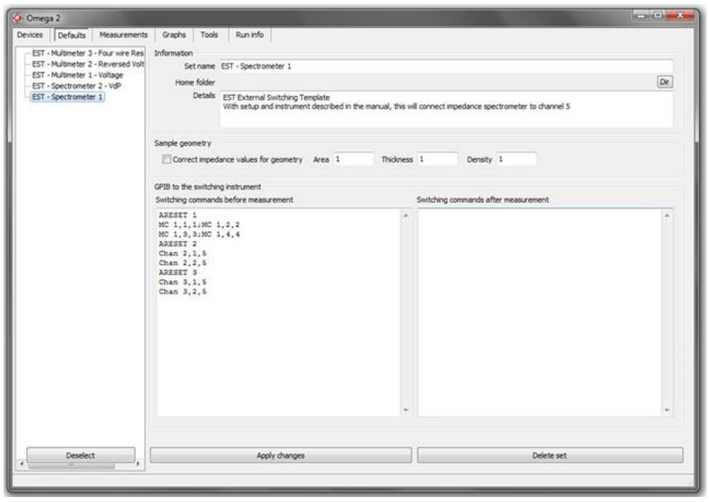
Data save location: C:_Temp3\

2.15 Defaults tab

Defaults tab allows users to create default sets for reoccurring situations. These sets and the values in them can later be applied to measurements with a click of a button; this saves the user from typing the same (usually sample related) values repeatedly.

These are very much dependant of the current exsiting setup and mainly aimed for users who are using single set of instruments through a multiplexer to connect to a number of measurment cells in sequences.

2.16 Measurement entity



EST - Multimeter 3 - Four wire Res
EST - Multimeter 2 - Reversed Volt
EST - Multimeter 1 - Voltage
EST - Spectrometer 2 - VdP
EST - Spectrometer 1

Information
Set name: EST - Spectrometer 1
Home folder:
Details: EST External Switching Template
With setup and instrument described in the manual, this will connect impedance spectrometer to channel 5

Sample geometry
☐ Correct impedance values for geometry Area 1 Thickness 1 Density 1

GPIB to the switching instrument
Switching commands before measurement

```
ARESET 1
MC 1,1,1;MC 1,2,2
MC 1,3,3;MC 1,4,4
ARESET 2
Chan 2,1,5
Chan 2,2,5
ARESET 3
Chan 3,1,5
Chan 3,2,5
```

Switching commands after measurement

A measurement is a set of instructions what to do and how to measure. These instructions are called nodes. One single node instructs one instrument and measures one specific property such as voltage or impedance and stores this data. A measurement can have any number of nodes measuring range of properties. This measured raw data can be viewed as it is, or further processed and displayed on the graphs section.

A measurement may also have any number of graphs, with custom series displayed on them. A series can access and display the raw data from any node, or further on combine the data from several nodes and calculate with them before plotting the result on the screen. In addition to nodes, graphs and series, a measurement has a set of general properties such as measurement name and "speed limit".

With all of these settings it is possible to construct elaborate measurement methods such as van der Pauw measurements, Seebeck coefficient measurements as well as simple measurements such as sample conductivity versus sample temperature.

For example, a Seebeck coefficient measurement would consist of a node to measure the furnace temperature for general knowledge, a second node to measure temperature at the ‘hot end’ of the sample, a third node to measure temperature at the ‘cold end’ of the sample, fourth node to measure sample voltage, and a fifth node to measure sample voltage reversed (for elimination of thermal emf of the electrodes themselves). Additionally one might have extra node measuring a resistance of a thermistor (resistance changes as function of temperature) at cold juction, to know the exact room temperature.

Typically there is only one measurement active at a time, but in case there are more, all the measurements open and marked active and otherwise valid for measuring, will try to measure. One by one, in a sequence and in alphabetical order of measurement name they measure or perform their tasks, and start over from the first.

Similarly each active node inside a measurement will perform it's task, retrieve the requested data, and remember it with information of the exact moment of the measurement performed. Nodes also perform in their alphabethic order, and when all nodes have performed, the focus is passed to the next measurement. The exception to this, is the impedance sweep node type that will perform the whole defined sweep in one go, and then become inactive. It is also posible to create a measurement where each frequency of impedance sweep is it's own phase and other measurements or nodes can perform between these steps.

For a measurement to run a number of conditions must be valid.

- A valid license must be present, namely a USB license stick on a USB port
- Devices are validated
- Assigned device ID must match with the current device ID
- Measurement must be marked as active
- Measurement must have node(s) active

Note that changes to the measurement (and node) properties take effect only after a full measurement loop has finished.

Measurement duration

Measurement can be activated or deactivated only by hand, so it will stay idle or it will keep measuring forever until user action is taken. However, the nodes inside a measurement can individually be programmed to start or stop to measure or perform actions using simple, or complicated rules.

Multiple measurements

It can be useful to have multiple measurements open, either to measure in sequence or just for viewing.
If the measurement lab happens to have multitude of instruments, or a signal switching device, the software can measure multiple samples in separate sample holders, switching the focus between measurements and instruments back and forth.

Measurement and node execution order

Most times there are multiple nodes per measurement and sometimes even multiple measurements measuring at the same time. The order of their execution may be vital, so measurements and their nodes are executed in the order of their caption, as shown in the Measurements tab. By editing the caption of a measurement or a node it is possible to alter the execution order. New nodes are named with a prefix to help ordering them.

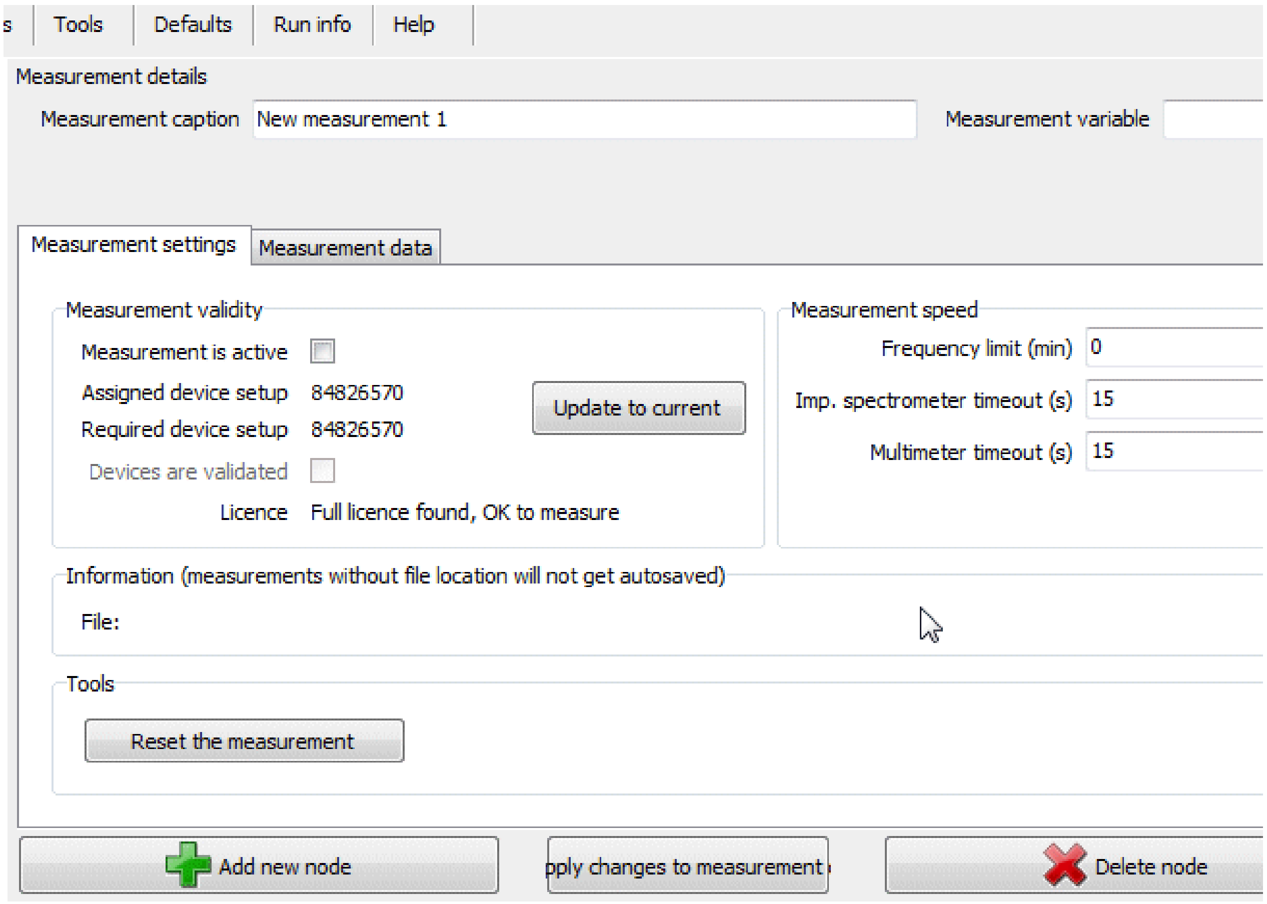
2.17 File

Everything concerning a measurement can be saved to a single text based file. There is no limit on how many different measurement files can be open in the program. Any measurement open in the program also gets automatically save on the file where it was opened and also into the /Measurements/Autosave folder.

In case when opening several instances of the same file (which is useful in some cases), it is good to remember they all share the same file as their origin and when saving (manual, autosave or save all) these different instances will write themselves to the same file, overwriting the previous save and resulting in loss of data. It is thus recommended, after opening multiple instances of one file, to save each instance with separate file name right after opening.

Every 60 minutes all open measurements are saved to Measurements\Autosave\ folder with a suffix – save_xxx where xxx is an increasing number.

2.18 Settings



Measurement details

Measurement caption: New measurement 1

Measurement variable:

Measurement validity

Measurement is active ☐

Assigned device setup 84826570

Required device setup 84826570

Devices are validated ☐

Licence Full licence found, OK to measure

Measurement speed

Frequency limit (min) 0

Imp. spectrometer timeout (s) 15

Multimeter timeout (s) 15

Information (measurements without file location will not get autosaved)

File:

Tools

The ‘Measurement is active’ checkbox will define if the software will try to measure this measurement. If unchecked, the measurement will be open but idle, and it will not measure or perform.

All measurements and nodes that are active get measured over and over again, and the ‘Measurement speed - frequency limit’ defines in minutes the lowest frequency the measurement can be repeated. A value of 1.5 would not allow the measurement in question to measure the more often than every 90 seconds. A value of 0 would measure as fast as possible, but the actual speed depends of what other measurements are being measured and how long they take. In other words the meaning is ‘Do not measure more often than every x.x minutes’.

Allowing long term measurements to measure as fast as they can will eventually result in huge amounts of datapoints and possibly slowing down the performance such as screen redrawing when resizing, loading and saving measurements or exporting data. For example, measurement with 40 000 datapoints on a modern computer took 2 minutes to open; while another measurement with 15 000 datapoints took only 15 seconds. The delays are dictated by computer processor speed and available memory. When frequent data points are required, it is possible to later on to cull the old data from areas of low interest within the node data tab.

Impedance spectrometer timeout is a limit of time that is allowed for the impedance spectrometer to try to measure a single point before the operation is cancelled. Sometimes it is just impossible for the instrument to get a valid reading, and there may be other measurements that still need to keep running. If long integration times, or other features that require lot of time, are enabled on the spectrometer, consider increasing this time. Extra time required for below zero frequencies is automatically added to the timeout time using 1/used frequency. Note, that when using averaging by instrument (average of multiple readings), this may not be sufficient. Multimeter timeout is defines the maximum available time for multimeters to perform their task. In the information box is the location and name of the measurement file.

The 'Required devices setup' id is a unique number identifying which instrument setup was used to measure this measurement. The 'Assigned devices setup' id is unique number identifying the devices setup currently assigned to the program. The software cannot measure unless these two id's are the same.

In case the devices setup has changed, or the user is trying to open and adopt a measurement made by someone else these two id's do not match. The measurement can be forced to the current 'Assigned devices setup' by clicking 'Update to current'. In such case the user needs to make sure all and each of the nodes are updated to match the present devices setup by selecting and applying the correct instruments and settings.

Clicking the 'Update to current' will prompt following text and Yes/No option. "*Assigning a new device ID to a measurement may stop your measurement from working, but it may also be required in order to make your measurement work. In case the devices setup have changed only due to new instruments being added, the measurement can be assigned the new Device ID and will work just as before. In case devices were removed or reconfigured, you need to go through all nodes and reconfigure those nodes in the measurement that refer to the absent or changed instruments. This also applies in case you are opening a template measurement and want to make it work with your instrument setup. The new Device ID for the measurement will be the one currently validated in the program, and the measurement will be made inactive. There is no permanent harm in answering yes, just as long as you do not save and overwrite any old saved measurements.*"

2.19 Expressions

Many aspects in the functionality of the software rely on expressions, and expressions are referred often in the manual. Expressions are user defined or pre-provided instructions or formulas, that get evaluated at run time. In few words, expressions are a way to refer to the data collected, instructions what to plot or conditions when to start or stop a measurement.

The expression is evaluated and the resulting number used to control the behavior of the software. In the simplest case, an expression is just a number, and this, when evaluated, will always result to that number. The expression can however include references to measured data, use mathematical operators, functions, conditionals and so forth. It is quite like using the formulas in Microsoft Excel, and will be explained in detail later in the manual.

When an expression is evaluated, the parser will look through the expression for variables. A variable is an acronym of characters and numbers, such as $N1.MV (Node 1, measured voltage). All these variables will be replaced by the actual values of data. If the measured voltage was 5.1, the $N1.MV would be replaced by 5.1. Once all variables have been replaced, the parser will see if there are any function calls in the expressions. Functions will take the parameters in the brackets after the function call acronym, calculate with the values, and then return some result. This is done for all variables, function calls and mathematical operators until a numeric result is reached, or in case of an error, a error indicator NaN is given as the value.

What is then done with the result depends of the context. If expression was a serie y-expression the result determines the y coordinate of a point added to a graph and series. If the expression was node start expression the value is used to determine if the node should perform or not.

2.20 Operators and functions

Basic math operators and functions

Acronym	Full name	Notes
+	Addition	Returns the sum of left side added to right side.
-	Subtraction	Returns the subtraction of right side from the left side.
*	Multiplication	Returns the multiplication of left side and right side.
/	Division	Returns the left side divided by the right side.
^	Exponentiation	Returns the left side to the power of the right side.

Acronym	Full name	Notes
%	Integer division	Returns the integer part of left side divided by right side.
() [] {}	Brackets	Any level of nesting brackets is supported. Expression inside brackets gets evaluated first.
< > = <= =>	Comparison	Returns 0 if comparison is false and 1 if comparison is true. -10 > 20 = 0 or -10 < 20 = 1.
&	Comparison and	Returns 1 if left side <>0 and right side <> 0
|	Comparison or	Returns 1 if left side <>0 or right side <> 0
ABS(A)	Absolute value	Return the absolute value of A.
ATAN(A)	Arctangent	Returns Arctangent of A.
CEIL(A)	Ceiling	Returns the next highest integer value.
COS(A)	Cosine	Returns cosine of angle A.
COSH(A)	Cosine hyperbolic	Returns hyperbolic cosine of angle A.
COTAN(A)	Cotangent	Returns Cotangent of A.
EXP(A)	Exponent	Returns e raised to the power of A.
FLOOR(A)	Floor	Returns the next lowest integer value.
IF(A,B,C)	If then else	Returns B if A is <> 0 (A=true), returns C if A = 0 (A=false).
INTPOW(A,B)	Integer power	Raises A to an integral power of B. INTPOW(2, 3) = 8 or INTPOW(2, 3.4) = 8
LN(A)	Logarithm natural	Returns natural logarithm of A.
LOG(A)	Logarithm 10	Returns 10 based logarithm of A.
LOGN(A,B)	Log base	Returns the log base A of B. Example: LOGN(10, 100) = 2 or LOGN(2,8) = 3
MAX(A,B)	Maximum	Returns the higher of A and B
MIN(A,B)	Minimum	Returns the lower of A and B
MOD(A,B)	Modulus	Returns modulus of trunc(A) / trunc(B). MOD(5,3) = 2 or MOD(5.5,3) = 2
POW(A, B)	Power	Raises A to the power of B. Effectively same as A^B.
RANDOM(A)	Random float	Generates a random floating point number such that 0 <= Result < A.
RND(A)	Random integer	Generates a random integer number such that 0 <= Result < A.
SIGN(A)	Sign	Returns -1 if A<0; +1 if A>0, 0 if A=0.
SIN(A)	Sine	Returns sine of angle A.
SINH(A)	Sine hyperbolic	Returns hyperbolic sine of angle A.
SQR(A)	Square	Returns A^2
SQRT(A)	Square root	Returns square root of A.
TAN(A)	Tangent	Returns Tangent of A.
TRUNC(A)	Truncate	Discards the fractional part of a number.

Advanced math functions

Acronym	Full name, links	Notes
EHOUR(T)	Elapsed hours	Returns amount of hours elapsed between the first point of any node of the measurement and the parameter T, a given time value of a node such as $N5.TI Set axis scaling to normal.
EMIN(T)	Elapsed minutes	Returns amount of minutes elapsed between the first point of any node of the measurement and the parameter T, a given time value of a node such as $N5.TI Set axis scaling to normal.
ESEC(T)	Elapsed seconds	Returns amount of seconds elapsed between the first point of any node of the measurement and the parameter T, a given time value of a node such as $N5.TI Set axis scaling to normal.
TCK(V, T)	Thermocouple K type	Returns a temperature in °C. Parameter V is voltage (from K-type thermocouple) with valid range of -6 mV to 54 mV. Parameter T is the ambient temperature in °C with valid range of -100°C to 100°C. Function returns -

		1000°C if parameters are out of valid range. For the parameter T a known room temperature can be entered or for more accurate results a true temperature can be calculated using TTT function or by other means.
TCS(V, T)	Thermocouple S type	Returns a temperature in °C. Parameter V is voltage (from S-type thermocouple) with valid range of -0.235 mV to 18.693 mV. Parameter T is the ambient temperature in °C with valid range of -100°C to 100°C. Function returns -1000°C if parameters are out of valid range. For the parameter T a known room temperature can be entered or for more accurate results a true temperature can be calculated using TTT function or by other means.
TPP(T)	Thermopower Platinum	Returns the absolute thermoelectric power of Platinum in Volts/degree Celcius. Parameter T is temperature, given in degrees Celcius. Function fitted from data presented in Thermoelectricity in metals and alloys - R.D.Barnard, Thermoelectric power of metals – Blatt, with maximum error (from the data in the book) of 0.08 microvolts above 250°C region.
TT2(R)	Thermistor to Temperature	TT2(R) Thermistor To Temperature returns a temperature in °C. Parameter R is resistance measured from thermistor type EPCOS serie:B57861S type:B57861S0103F45 The function is valid between 2986Ω-32650Ω, a temperature range of 0 to 55°C accordingly. One thermistor of this type delivered with the software.
VDP(A, B)	Van Der Pauw	Returns R when $\exp(-\pi A/R)+\exp(-\pi B/R)=1$ Parameter A and B are resistances measured according the van der Pauw method described in the manual. Note that A and B cannot be of opposing signs or zero. Function returns -1000 if parameters are out of valid range. To get the specific resistivity multiply with thickness of the lamellae.
MOSE (T,E,A,B)	Oxygen partial pressure with MOSE sensor	Returns pO2 in atmospheric pressure. T=Temperature in Celsius, E=measured sensor voltage, A and B are calibration constants provided with the sensor. Use Ax and Bx from the included excel file, cell E54 and E55.
ISNAN(A)	Is not a number	Returns 1 if A is not a number, otherwise 0
NERPO2 (E,T,R)	Nernst pO2	Returns partial pressure of oxygen in atmospheric pressure (ATM) when measured with zirconia electrolyte with reference gas. E is the sensor voltage, T is the temperature in Celsius, R is the reference gas pO2 in ATM.
NEREMF (REF,EXP,T)	Nernst emf	Returns electromotive force in Volts over solid zirconia electrolyte when pO2 of both sides (REF for reference and EXP for experiment) and temperature (T) is known.

Node data types and advanced data properties. Prefix $Nx

Acronym	Node types	Description	Formula used	Notes
$I	Any expression	Count of data		No prefix needed
TI	All	Time elapsed, delphi type number		From 1st datapoint. Deleting node data will affect this
TS	All	Time elapsed, seconds		From 1st datapoint. Deleting node data will affect this
TM	All	Time elapsed, minutes		From 1st datapoint. Deleting node data will affect this
TH	All	Time elapsed, hours		From 1st datapoint. Deleting node data will affect this
TD	All	Time elapsed, days		From 1st datapoint. Deleting node data will affect this
FAM	All	Time in minutes since the node was first active. Independent of first data in node.		
LAM	All	Time in minutes since the node was last active. Independent of first data in node.		
$TIME	Any expression	Time now, delphi type number		No node prefix needed
RS	IS, IC	Resistance serial		Saved from instrument, raw field 1
DF1 V1	All	Whatever is in the field		Datafield1
DF2 V2	All	Whatever is in the field		Datafield2

DF3 V3	All	Whatever is in the field		Datafield3
RS	IS, IC	Resistance serial Saved from instrument, raw field 1		RS. Many instruments will return negative resistance in some cases. Absolute value is still correct.
X	IS, IC	Reactance		Saved from instrument, raw field 2
F	IS, IC	Frequency		Saved from instrument, raw field 3
P	IS, IC	Phase angle	arctan(X/RS)	arctan = $\tan^{-1}$ Do not use, kept only for backwards compatibility
PA2	IS, IC	Phase angle	arctan2(X, RS)	The normal inverse tangent function has a range between -π/2 and π/2, or -90 to 90 degrees. Many instruments, when measuring difficult samples or in noisy situations, return the real part of impedance measurement with negative polarity (with correct amplitude). The regular arctan function is not able to handle the calculation correctly. Arctan2 is improved function that can handle whole range of input values from -180 to 180 degrees.
Z	IS, IC	Impedance	sqrt(RS*RS+X*X)	sqrt = Square root
Y	IS, IC	Admittance	1 / Z	
G	IS, IC	Conductance	Y*cos(-P)	-P = negative phase angle
B	IS, IC	Susceptance	Y*sin(-P)	-P = negative phase angle
RP	IS, IC	Resistance parallel	1 / G	
LS	IS, IC	Inductance serial	X / (F*Pi*2)	F*Pi*2 = Angular velocity
LP	IS, IC	Inductance parallel	1 / (B*F*Pi*2)	F*Pi*2 = Angular velocity
CS	IS, IC	Capacitance serial	1 / (X*F*Pi*2)	F*Pi*2 = Angular velocity
CP	IS, IC	Capacitance parallel	B / (F*Pi*2)	F*Pi*2 = Angular velocity
SF	IS, IC	Sweep is finished		0 or NaN if sweep has not finished, 1 if sweep has finished.
M2	M2	2-wire DC resistance		
M4	M4	4-wire DC resistance		
MV	MV	DC voltage		
MC	MC	DC Current		
PV	P1	Potentiostat voltage		
PC	P1	Potentiostat current		
OX	OX	Oxygen partial pressure		
WSP	ET	Working setpoint		For node type ET second data field, acronym WSP is the working setpoint of the furnace.
ET	ET	Measured temperature		The node variables .MAX1, .MAX2, .MAX3, .MIN1, .MIN2 and .MIN3 return the highest or lowest value available in the nodes corresponding raw data field, for example MAX2 for IC type node will return reactance.
FL	FL	Measured flow		
FLS	FL	Flow setpoint		
AF1	AU	New furnace setpoint		

AF2	AU	New furnace ramp rate
AG1	AU	New value 1
AG2	AU	New value 2
AG3	AU	New value 3
AM1	AU	New flow setpoint
MIN1	All	Lowest value of field 1
MIN2	All	Lowest value of field 2
MIN3	All	Lowest value of field 3
MAX1	All	Highest value of field 1
MAX2	All	Highest value of field 2
MAX3	All	Highest value of field 3

Expressions using information from series. Prefix $Sx

Variable	**Explanation**
.C	The amount of data points in the series.
.XS	The sum of all X-components.
.XAV	Average of X-components.
.XMA	Highest X-value in series.
.XMI	Lowest X-value in series.
.YS	The sum of all Y-components in series.
.YAV	Average of all Y-components in series.
.YMA	Highest Y-value in series.
.YMI	Lowest Y-value in series.
.Y	Series Y value.
.LRR	Range of Linear regression / linear least squares method.
.LRA	Offset of calculated line: Y = .LRA + .LRB * X
.LRB	Slope of calculated line: Y = .LRA + .LRB * X
.LRC	Statistical uncertainty of the .LRA
.LRD	Statistical uncertainty of the .LRB
.LRE	Average of the X values in coefficient range.
.LRF	Average of the Y values in coefficient range.
.LRG	Variance of the X values in coefficient range.
.LRH	Variance of the Y values in coefficient range.
.LRI	The probability (r^2) the linear model fits to the empirical data points.
.LRJ	The correlation (R) of the linear regression.
.LRK	The sum of the residuals (square deviations) from the calculated line to the measured data points.
.LRMA	Maximum Y-value within the range of the coefficient tool.
.LRMI	Minimum Y-value within the range of the coefficient tool.

2.21 Expression editor

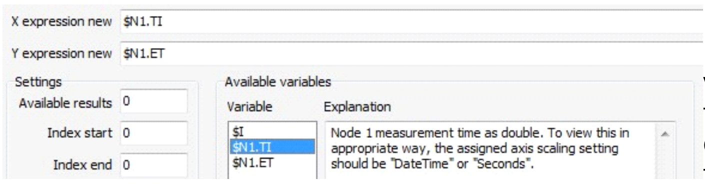
Expression editor window is a part of the Series properties page. In the expression editor the user can design and test new expressions to plot. Expression can hold numbers, mathematical operators, brackets, function calls, and variables. Expression can range from a single number or variable to an elaborate mathematical expression. The expression is solved and the result determines the coordinates of the point to plot: the X expression returns the X-coordinate, and Y expression returns the Y-coordinate of the point to plot. A counter runs through all the data in the measurement and solves the points to plot. The expression parser has access to all of the data in all the nodes of the measurement but it also allows advanced calculations with the data.

When plotting series, the parser will loop through all the data a measurement has, and make the coordinate point for each instance of data. In the expression editor window it is possible to limit the range of the index by editing the values of Index start and Index end. Normally the actual value of the index is not needed for anything, but variable $I will result as the numeric value of the index. If a measurement had 15 points of data, plotting the $I for both expressions would just draw straight line with points such as 1,1 2,2 … 15,15

Sometimes it is useful to plot functions not related to measured data. In these cases the $I variable is very useful as it can be set to loop from any given value to any other given value.
To overcome the full integer step nature of the loop, one can divide the number in the formula, or one can use the $I variable shortcuts.

Expressions X: $I and Y: SIN($I.1) with range from 0 to 300 would plot a nice sin function, while with Y: SIN($I) the plot would be too stepped and with Y:SIN($.2) the plot would not cover the whole range of SIN function.

Expressions example 1



X expression new: $N1.TI
Y expression new: $N1.ET

Settings
Available results: 0
Index start: 0
Index end: 0

Available variables

Variable	Explanation
$I $N1.TI $N1.ET	Node 1 measurement time as double. To view this in appropriate way, the assigned axis scaling setting should be "DateTime" or "Seconds".

In the simplest form, an expression is just a variable identifying a specific measurement result from one of the nodes of the measurement. For example, if node 1 was defined to measure furnace temperature, the expression for X axis could be time of the measurement point, and expression for Y axis could be the temperature itself and thus the plot would be temperature of the furnace against time. The corresponding variables for these are X: $N1.TI and Y: $N1.ET

2.22 Nodes

A node is an instruction to measure a specific property such as voltage or impedance. Each node is holds the details about an instrument to measure with, the settings used to measure, possible instructions on how and where to connect to. Some node types behave differently, an impedance sweep type node performs more than one measurement, all in one go, and action type node performs automation tasks but does not measure anything.
Node has an acronym to designate its type, field to hold the time of measurement, 3 fields to hold measured data, and range of settings and instructions described later in this chapter.

Acronym	Node type	Data type 1	Data type 2	Data type 3	Notes
IC	Four wire impedance	Resistance (Ω)	Reactance (Ω)	Used frequency (Hz)	
IS	Four wire impedance sweep	Resistance (Ω)	Reactance (Ω)	Used frequency (Hz)	Measures whole impedance sweep in one go, according given settings.
M2	Two wire resistance	Resistance (Ω)			
M4	Four wire resistance	Resistance (Ω)			
MV	Voltage	Voltage (V)			
MC	Current	Current (A)			
ET	Temperature	Temperature (Instrument specific, °C, °K or °F)			Data is read from Eurotherm reader or controller in whatever form the controller is configured to display, normally °C.
AU	Action node				Does not measure but performs tasks instead.
FL	Flow	Measured flow (Instrument specific unit)			Unit of the flow can be seen on the instrument label.

Nodes of type IC and IS use impedance spectrometers, nodes of type M2, M4, MV and MC use multimeters, node ET use Eurotherm reader/controller (furnace), node FL uses Vögtlin mass flow controller/meter and node AU can perform tasks with any type of instrument. Additionally the imp spectrometer and multimeter nodes can command additional GPIB instrument such as switch to multiplex the signal cables between sample(s) and instrument(s).

Each node performs these tasks in this specific order

Phase	Explanation
Node active	Determines if the node should perform itself or be skipped. The following aspects are evaluated and the results need to evaluate to yes for the node to get performed. • Is node is marked active • Is the start condition not zero • Is the stop condition zero • For IS type node the ‘sweep status’ must be not ‘finished’
Switching	• When no switching (or internal switching) is selected for node, nothing is done. • When manual switching is selected, the software prompts the user to switch the cables and pauses the program until user confirms the switching has been done. • When external switching is selected, the software sends GPIB commands to the selected switching instrument.
GPIB	The user may specify GPIB commands to be sent to the instrument for additional

Phase	Explanation
commands	control. Some of these settings may be overruled by the commands the software automatically sends in next step.
Instrument initialization	The software will set the instrument to proper mode of operation for the node type.
Call measurement	The software will ask the instrument to perform the measurement. In case the node is action node, no measurement is performed.

2.23 Properties

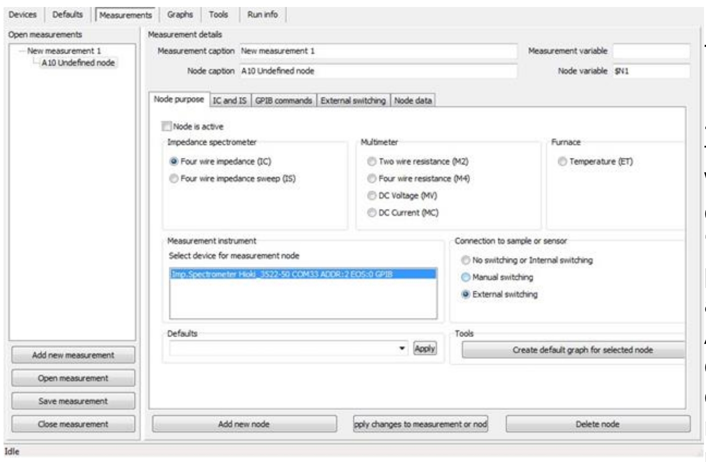
The node to be edited is selected on the list on the left. Selected node details are shown on the right. All changes must be confirmed by clicking ‘Apply changes…’ button on the bottom.

Property	Types	Comments
Node type	All	Type of the node, possible values IC, IS, M2, M4, MV, MC, ET, FL, AU.
Node caption	All	Captions determine the execution order of nodes and inform the user of the purpose of the node.
Node variable	All	This non-editable property defines the variable to be used in the expressions to access the data in the node.
Node is active	All	When executing a measurement, inactive nodes are ignored, active ones will measure and perform.
Instrument	All	List of possible and relevant instruments are shown here for the selected node. The selected instrument is the one used to perform the node task.
Connection	All but ET, FL, AU	Type of the connection scheme of the node. See chapter Switching for more details.
Voltage	IC, IS	Defines the voltage used for AC signal. Refer to instrument manual to see if RMS or peak is used.
Frequency	IC	Defines the frequency used for AC signal.
Frequency start	IS	Defines the start frequency for sweep.
Frequency end	IS	Defines the end frequency for sweep.
Sweep points	IS	How many points to measure in the sweep. Frequency is changed logarithmically not linearly
Frequency step	IS	Read only field. Shows the step of frequency for the given parameters.
Correct impedance values	IC, IS	Enables or disables correction for sample volume. Default graph axis and plot names in this manual and in the software will regardless refer to the total values and not volume specific values.
Area	IC, IS	The RS and X raw data retrieved from instrument will be corrected before storing into computer memory with *Area/Thickness/Density[2] Names for the corrected properties will not change; resistivity is called resistance and so forth.
Thickness	IC, IS	
Density	IC, IS	
GPIB before measuring a point	All but ET, FL, AU	Additional instructions to the instrument that control the behavior of the instrument. See chapter GPIB for more details
GPIB after measuring a point	All but ET, FL, AU	Text that controls the behavior of the instrument. See chapter GPIB for more details
Select switching instrument	All but ET when external switching is used	All the GPIB instruments not belonging to spectrometers or multimeters are shown here. Clicking one will select the instrument to perform external switching.
Switching commands before measurement	All but ET when external switching is used	The GPIB commands used to perform the desired switching.
Switching commands after measurement	All but ET when external switching is used	The GPIB commands used to perform after the desired switching.
Start condition	All	An expression determining when measurement becomes active. Any non-zero value for this means the node will try to perform. Additional to other conditions.
Stop condition	All	An expression determining when measurement stop being active. Any non-zero value for this means the node will not perform. Additional to other conditions.[1]

The **node data tab** and the **action tab** are discussed in their own chapters.

In addition to the properties, two tools exist on the main page of node properties; Apply defaults and Create graph. To apply defaults set, select the set name from the drop down list and click apply. More information about the defaults can be found in the 'defaults' chapter.
Most of the time the measurements are simple, and the graph to accompany the measurement can be predicted. The 'create default graph' button will create a graph, the axes and plots that make most sense for the measurement node in question. The graph can easily be customized and plots for other nodes can be added to the graph on the graphs tab, explained later in this manual.

'Reset IS node' button will delete all data in the node, and activate the sweep to be made again with the existing settings.



New measurement 1
A10 Undefined node

Measurement caption: New measurement 1
Measurement variable:
Node caption: A10 Undefined node
Node variable: $N1

Node is active

[✓ Four wire impedance (IC)]

Select device for measurement node: Imp.Spectrometer Hioki_3522-50 COM33 ADDR:2 EOS:0 GPIB

[✓ External switching]

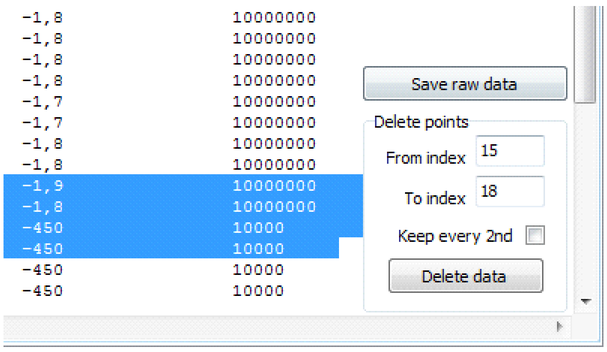
Idle

The purpose of node captions is to
1) describe the node
2) help define and alter the order of the nodes
When measurement executes, the nodes execute in alphabetical order of their captions. 'Sorting' nodes is easy by adding or editing small prefixes in their names, such as A10, A20, A30 and B05.
Also when in expression editor window, node caption is showed in the explanations box for each node variable. For measurements measuring more than one instance of a specific node type it may be worth the effort to name the nodes such as 'A1 sample voltage straight', 'A2 sample voltage reversed', 'A3 Top thermocouple voltage'.

Each time a node or measurement tries to execute, start and end conditions are checked. Conditions can be such as; now, certain time and date, never, but also an expression to be validated. (Currently just checkbox to measure or not) For example test if the temperature is stable at 900°C before allowing a sweep type node to start.

2.24 Data tab

The node data tab contains no properties for the node but instead displays the measured data, holds the means for deleting data points and to save the raw data in various formats.



```
-1,8      10000000
-1,8      10000000
-1,8      10000000
-1,8      10000000
-1,7      10000000
-1,7      10000000
-1,8      10000000
-1,8      10000000
-1,9      10000000
-1,8      10000000
-450      10000
-450      10000
-450      10000
-450      10000
```

The node data tab creates a text list of all data in the selected node. The first column is the index of the data, second column is the time the measurement was made, and the next one to three columns contains the actual data. The data columns are named appropriately depending of the node type, showing for example 'voltage' for MV node or 'resistance, reactance and frequency' for IC or IS type node.
Clicking the 'save raw data' opens a save file dialog to save the data from the selected node. The saved format is text based .ome file for all node types, and additionally optional formats .z and .ecq for sweep node types. The formats are variations of a typical .csv format which is a text based file easily opened, converted and imported into many other programs.
Deleting data points deletes the same data from all the other nodes in the measurement from same index numbers. Exception to this is the IS node type; No other node deletes data from the IS node nor does deleting data from IS node does not delete it from any other nodes.
Data gets deleted between the specified indexes when Delete data is clicked. Indexes can be specified by keyboard or by painting the lines by mouse. If 'Keep every 2nd' checkbox is checked, only every second data point will be deleted between the specified indexes.

Typically a measurement goes on for longer than just one measurement point. When a single measurement loop is done, a typical action would wait for some time and then redo all the nodes again, and repeat this until certain amount of results has been acquired, or indefinitely. All data acquired during one of these loops share the same index. Normally the user doesn't need to be aware of the indexing system as it is taken care of automatically. However, there are few cases where the user should be aware of the index related behavior:

1. A node is added to a measurement after other nodes have already been measuring. In such case the newly added node will be filled with undefined NaN (Not a Number) values all the way up to the current index of the other nodes. It is not possible to evaluate expressions that have references to NaN values and such attempts will be skipped.

2. Node of IS type. Impedance sweep node type does not share index with the other node types. While measurements and nodes take turns while executing, a sweep node performs full and complete sweep (many measurement operations) in its turn and will never execute after that. The index of IS has thus no relevance with the index of other data types and it is not sensible to combine data from IS type node with other nodes in the expression.

   It may, however be useful to do multiple sweeps with same amount of data points. In such case the frequencies and the indexes of separate sweeps match and data from separate sweeps can be combined or compared, for example sweeps with different samples or different voltages or sweeps without the sample to eliminate the electrode characteristics.

Easiest way to grasp the concept of node data index is to go to the 'measurements tab', select a measurement on the left, and go to the 'measurement data' tab. The raw data of all the nodes in the measurements is listed with common index and time for each line (IS type nodes are ignored from the listing).

Index	Time	Node 1 Voltage	Node 2 Voltage	Node 3 Voltage
0	9:00:00	3.0	0.1	-5
1	9:00:10	3.0	1.1	-5

Index	Time	Node 1 Voltage	Node 2 Voltage	Node 3 Voltage
2	9:00:20	3.0	2.1	-5
3	9:00:30	3.0	3.1	-5

In the fictional data here the measurement took 3 separate voltages with 3 separate MV nodes with approximately 10 second interval.

2.25 Data time format

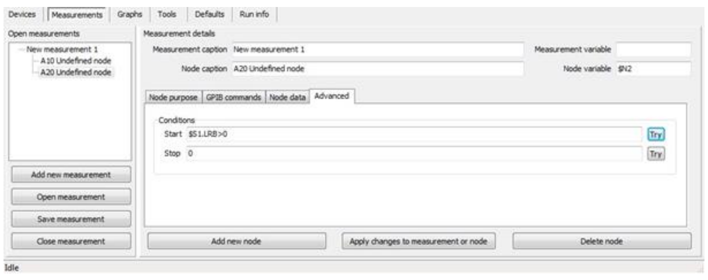
The time data type in node has a peculiar format called Delphi datetime. As the name suggests, it holds information about both the date and the time. In this notation the integer part of the number represents the days passed since 30th of December 1899, and the fractional part of a value is fraction of a 24 hour day that has elapsed.
Not that user needs to bother, but as an example the 27th of September 2012 at 3PM would be 41179,625 in this format. The 41179 days divided by 365 days is some 112.8 representing elapsed years, while the 24*0,625 is 15 or in other words 3PM.
The format is mostly handled and converted automatically, but the user needs to be aware of one thing; when plotting data on the graphs tab, the axis does not know what kind of data is being plotted. Axis starts by default with a scaling setting of 'normal' it means it will display numbers in a linear scale. Changing the axis scaling setting to DateTime makes the axis automatically convert any number to a standard date and time format assuming the numbers in the mentioned way.
For the computer this is the best way to store both date and time information, but for the user it may not be as convenient especially if custom expressions is required to be plotted. For that reason range of other time parameters are also offered.

2.26 Operators and functions

For additional flexibility, a tool in the tools tab called 'time tool' allows user to convert and list date and time as the Delphi datetime numbers.
It is also possible to use functions to plot elapsed time. Use functions such as elapsed seconds ESEC(), EMIN() or EHOUR(), and provide the time (.TI) as parameter.

2.27 Advanced node properties

Typically a node will perform if it is marked active on the node purpose tab. With start and stop conditions it is possible to further control when a node will perform. The start and stop conditions are expressions defining whether the node will measure. For node to measure it must naturally be marked as active, but by editing the start and stop conditions it is possible to have the node measure only when specific conditions are valid.



New measurement 1
A10 Undefined node
A20 Undefined node

Measurement caption: New measurement 1
Measurement variable:
Node caption: A20 Undefined node
Node variable: $N2
Start: $S1.LRB>0
Stop: 0

The expressions for both are freely selectable by the user, while typical uses would be expressions relating to absolute time (such as time and date), elapsed time (as in x hours after the measurement started) and equilibriums of interest such as new temperature or conductivity. Refer to chapter 'series coefficient and variables' for examples of suitable expressions, and 'General expression operators and functions' for information about the behavior of useful math operators such as <, >, & and |, and 'Node time data type' for information about the time format.

For the start condition, any value but 0 will allow the node to measure, while for the end condition any value other than 0 will stop the node from measuring. The default value for start is 1 and stop is 0 and with these the node will immediately start to measure and continue indefinitely.

Clicking the 'Try' button will evaluate the expression and give feedback. Use Time tool on tools tab for easy time conversions.

Condition / Expression	Explanation	Suggestions for usage
$N1.TM>45	Results 1 if measurement has been running for more than 45 minutes otherwise 0.	After elapsed time
$TIME>41290.592	If system time greater than 16th of Jan 2013, 14:12 this results 1, otherwise 0.	At specific time
$TIME	$TIME alone with the 'Try' button returns the current time. 0.04167 = 1 hour 0.00006944 = 1 minute	
$S1.LRB<0	If the coefficient tool for Series 1 of this measurement has downward slope this results 1, otherwise 0.	After peak, release, etc.

2.28 Data variables

When plotting data on graphs or accessing measured data with expressions, a special variable tells the computer what specific part of measured data the user wants to access. The variable consists of two parts separated by a full stop. The first part identifies the node, and the latter part identifies the property to return, for example $N1.MV

Each node has a unique variable that is assigned on the moment of node creation. First node created would be referred in the formulas with $N1 and eight node created as $N8. This node variable is visible on measurements tab when node is selected.
Depending of the node type, the list of available properties for the node will vary, and are listed later. In addition each node has a common property for the exact time of when the measurement took place, and it is accessed with acronym TI so for example node 1 measurement time would be $N1.TI and if the node 1 was of type MV measuring voltage, the data could be accessed with a variable $N1.MV Further on, plotting $N1.MV versus $N1.TI would draw a plot showing us how the voltage developed during the measurement. The node types M2, M4, MV, MC, ET and FL all access their data with the acronym same as the node type designator. For node types IC and IS there are many additional properties and acronyms to access either the raw data itself or some impedance related calculated properties.

In impedance related nodes (IC and IS), instead of one, three fields of raw data exist. They, and other calculated properties can be accessed with the acronyms in the table below.

2.29 Operators and functions

Remember that if the correction for sample geometry is enabled, this raw data and properties are volume specific regardless of what they are called here or on the graphs.
Note that some instruments return the absolute values of resistance and reactance instead of possibly negative values.

2.30 Action node

The action node type allows automation and tuning of connected instruments. The node will send to the selected instrument instructions or commands with user defined numerical values, or with values resulting from user defined expressions. Effectively, the node allows automation of the instruments, such as new temperature setpoints with furnaces, new gas flow with flowmeters, new level of applied power from power source and so forth. All the actions can be bound to be a function of time, or triggered from certain conditions. Using action nodes to automate measurement systems is powerful tool, but requires the user to understand the behavior of the targeted instrument, the commands or functions used to control the instruments and also the behavior of the Omega software and the evaluation of expressions.
For action type nodes the action tab will be visible. For each node appropriate instruments are shown and a

one must be selected to be the target instrument for the actions. Action node has three distinct types; Furnace (.AF), Mass flow controller (.AM) or GPIB instrument (.AG), and depending of the type a range of other settings becomes visible.

2.31 For furnace

The furnace type action node (AF) has two expressions; AF1 and AF2 and the fields correspond to furnace terms 'target setpoint' and 'ramp rate' accordingly. On the execution of the node, each expression gets evaluated and the value possibly sent to the target furnace. The AF type node requires the user to be familiar with furnace controller concepts. Refer to your furnace manual and use your furnace without Omega software to familiarize with the mentioned concepts.
The evaluated expression values are also stored as 'measured data' into the corresponding data fields. The third data field (AF3) holds values either 0 or 1, signifying if the setpoint and ramp rate was sent to the furnace; 0 for not, and 1 if value were sent. New setpoint is sent to the furnace only in case if AF1 or AF2 (or both) change to a new value.

It is important to understand that the software controls the setpoint and ramp rate of the controller of the furnace, not the temperature of the furnace. The furnace temperature is result of the actions of the furnace controller and is affected by things such as furnace controller settings and configuration, thermal mass, heat convection, conduction, radiation, thermal mass and so forth. The furnace controller does, to its best ability try to achieve the temperature requested by the user.
The furnace controller needs to be configured in such way that it is possible to control the furnace with sending values to setpoint and the ramp rate, Modbus addresses 2 and 35 accordingly. This includes details such as setting the 'selected setpoint' to setpoint 1, selecting appropriate unit for setpoint and ramp rate, and setting safe minimum and maximum restrictions for setpoint 1.

For AF type action node the expressions will be evaluated and rounded to closest integer value. When (and only when) either of the values change (compared to previous stored values) the new combination is sent to the selected furnace.
The max fields act as override/safety feature that limit the resulting value of the expression. Typing 900 to the Max field next to .AF1 will limit the setpoint sent to the furnace to 900 regardless of the expression value. Ramp program tool in tools section will help to generate expressions for ramp programs.

Note that value 0 for AF2 would disable the ramp function of the furnace controller. If the ramp rate function is disabled, the furnace tries to heat as fast as possible, potentially destroying heated items with thermal shock. For this reason it is not possible to turn off the ramp rate, and the minimum value for AF2 is limited to 1. For fast heating action specify a sufficiently high value for ramp rate.

Also note that the unit of ramp rate and the unit of setpoint are not defined through this program, but instead on the Eurotherm controller.

2.32 Mass flow devices

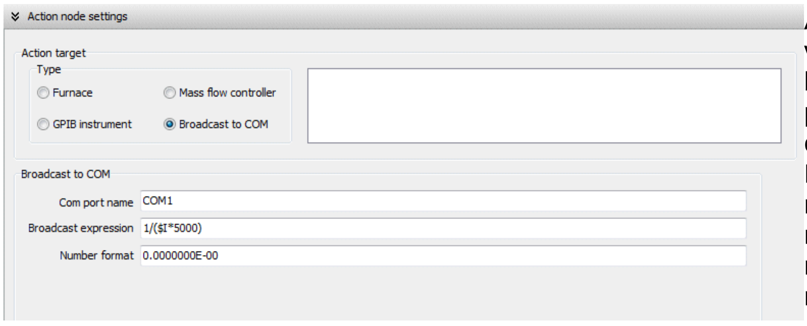
Action node for mass flow controller (.AM) type can send new setpoint to a selected flow controller. Max field limits the maximum value evaluated from the .AM1 expression. New setpoint gets sent on each execution of the node.
Ramp program tool in tools section will help to generate expressions for ramp programs.

2.33 GPIB devices

GPIB instrument type action node (AG) has three expressions; AG1, AG2 and AG3 and additionally a GPIB code field. On execution the GPIB code gets sent to the selected instrument. The code can include text and numbers, but also three variables $AG1, $AG2 and $AG3. These variables will be replaced with the resulting values of the corresponding expressions.

The results of the expressions can and need to be formatted to a number format that the instrument recognizes. For each variable a field with formatting instructions exists. For more on formatting numbers refer to the 'number format tool' on the tools tab.
With the AG node type it is possible to control any aspect of any GPIB instrument. It is possible for example to turn power source off when certain condition has been reached, or apply current as function of time. It is possible to generate complicated waveforms or functions and apply them on the desired property of the instrument.

2.34 Broadcast to COM


Action node settings

Action target

Type

○ Furnace ○ Mass flow controller

○ GPIB instrument [✓ Broadcast to COM]

Broadcast to COM

Com port name	COM1
Broadcast expression	1/($I*5000)
Number format	0.0000000E-00

Action node can also be made to write values to any free serial port. This may be usefull for many purposes. The Com port name must be valid, since it is not checked by the software in any way. Expression is the value to send, and number format is the format of the number. See numbers format tool for more information about formatting numbers.

2.35 GPIB commands before and after measurement

All nodes except the AU node type do measure the specified property automatically without the user having to add any additional commands. Advanced users may however want and need to have additional control over the behavior of the instruments. The fields for ‘GPIB commands before and after’ the measurements allow the user to enter additional commands to be sent to the instrument before and after the measurement. These commands may set the type and the properties of the measurement such as averaging, integration time, bias level and so forth. Everything in the instrument behavior can be customized with the GPIB commands. For supported devices some GPIB commands are provided by the program, but can be edited by the user. To know the GPIB commands your instrument is capable, please refer to the instrument manual.

It is also possible to use unknown devices for the role of multimeter, in case the device really is a multimeter but not recognized by the software. Connect the instrument to the GPIB network and do autoscan. It will be shown as unknown device. Assign a node of the type for multimeters, and provide the GPIB commands by hand to the GPIB before and after fields. Compare the existing commands and refer to the instrument manual to find similar commands. The instrument result must be formatted by the following rules to be able to be read back to the software.

Result text must consist of an optional sign (+ or -), a string of digits with an optional decimal point, and an optional mantissa. The mantissa consists of 'E' or 'e' followed by an optional sign (+ or -) and a whole number. Leading and trailing blanks are ignored

Decimal point may be either a dot or a comma.

Use this only as temporary solution and preferably request the instrument to be added to the software instead.

For user convenience, some commands are handled automatically by the software. These include the initialization of the instruments, the call for measurement and retrieving the result and resetting the instrument in case of timeout.

2.36 Types

2.36.1 ET

Node type ET is meant for reading the furnace temperature, and needs no extra configuration or settings. A furnace in the network is selected and the node reads the temperature.
The temperature is read from the furnace controller and is thus subject to all the settings of the controller such as thermocouple type, unit, filter time etc. On most furnaces the controller has its own thermocouple somewhere close to the heated zone, but this temperature is not necessarily same as the sample temperature.

Some furnaces are tailor made for ProboStat, and allow the control thermocouple to be overridden by a thermocouple from the sample, contact NorECs AS for further information on these models.

2.36.2 MV

Node type MV measures DC voltage between two terminals and is main building block for many types of measurements.
MV type node can also be used for custom resistance measurements, when a separate power source is connected to a circuit feeding well defined currents. In case switching is involved, a proper settling time for the current must be assured before taking a voltage measurement.

2.36.3 M2

2-point 2-wire DC resistance measurements

M2 is two wire resistance measurement measures the sample and wire resistance. M2 is suitable for measuring the thermistor resistance or other high resistances. For thermistor measurements it may be a good idea to set low NPLC- and averaging settings of the multimeter, as to save the thermistor from heating due repeated and long exposure to measurement current.

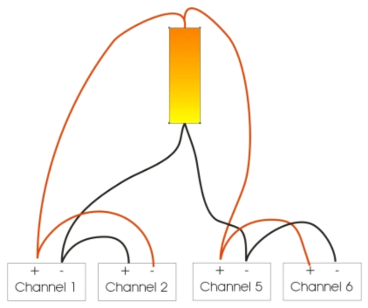
2.36.4 M4

2-point 4-wire DC resistance measurements

M4 is four wire resistance measurement using DC current. The setup may be 4-wire 2-point where the electrodes connect close to the sample, or just as well 4-wire 4-point where each electrode has its own contact to the sample.
The resistivity, conductance and conductivity can easily plotted just as well as resistance.

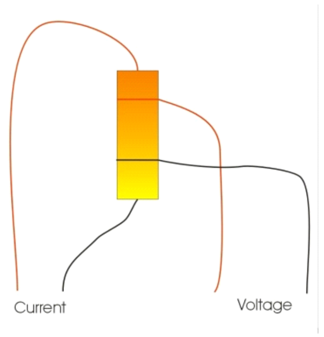
Resistance of the sample:	$N1.M4
Resistivity of the sample:	$N1.M4*(sample thickness or length/sample area)
Conductance of the sample:	1/$N1.M4
Conductivity of the sample:	1/resistivity

For M4 measurements the same considerations of thermal offsets and fluctuations apply than for the DC voltage measurements, but the remedies are more complicated as there are more wires.
Some multimeters have offset compensation correction, which can be used for improved results.
If the material is a mixed ionic electronic conductor, electronic conduction tends to dominate the measurement more than bulk properties suggest.



It is possible to use reverse and average for 2-point 4-wire in the same way as with the MV method.
M4 node uses 2 channels of the multimeter (such as 1 and 5) and with reversing the measurement additional 2 channels (such as 2 and 6). Reverse voltage leads but not the current leads, and sum the parts together instead of subtracting, as resistance is always positive.

To get the resistivity or conductivity with 4-wire 4-point on a bar sample the main factors for accuracy is the placement of the sense/voltage electrode contacts. A tiny groove on the sample to hold the electrode in place is a good idea and it helps to define the distance between the electrodes.



The reverse and average can be used in the same way as for 4-wire 2-point measurement, and some cases it may be possible to eliminate the temperature gradients by placing the bar horizontally.
If the temperature gradient gives too much error, it is also possible to replace each electrode with S type thermocouple, measure each sample related aspect with the Pt (-) lead of the TC, measure temperature at each contact point using the + and the – leads of each thermocouple, and from these temperatures, using the thermopower of platinum function, solve how much of a voltage between two points is from the contribution from the Pt wire and how much is from the sample. With this information and an external current source one can eliminate the effect of the temperature gradient. However, much easier is just to revert to AC measurements (the IC node type).

2.36.5 MC

MC type node can be used to measure DC current, for example from sensors, transmitters or gauges such as pressure or flow. The scale of such sensors is often from 4-20mA and the corresponding property is linearly scaled, for example from 0 to 15bar overpressure, thus following expression would result in sensor overpressure in bars.
EXP: ($N1.MV-0.004)/0.016*15

2.36.6 IC

Utilizing impedance spectrometer and node types IC and IS various types of complex impedance related measurements are possible. With AC measurements the raw data is more than just Time (TI) and the measured property, the data point includes Resistance, Reactance and Frequency for user convenience, and the acronyms ICR, ICX and ICF respectively. (Add G and B to node raw data types as well.)

2.36.7 2- and 3-point AC conductivity measurement

The software always runs impedance instruments in 4-terminal mode. It is up to the user to connect the terminals to the sample in a way to achieve 2- or 3-point measurement.

2.36.8 4-point AC conductivity measurement

For a bar sample measurement is done largely in same way as for DC 4-point.

2.36.9 IS

2.36.10 AU

Action node is a versatile node that can be used for controlling instruments. The use can be for example switching relays, controlling GPIB instruments that are not integrated to the software by default, sending dynamic values (as ASCII) to com ports, sending binary data to com ports, communicate with Vögtlin mass flow controllers or Eurotherm PID controllers and other such tasks.

…

Broadcast to com port feature opens the com port named, such as COM11. Be aware there is no validation of any kind here. The port must exist and be usable or the program will fail. Once the port is open, the software evaluates the broadcast expression given, formats the resulting value with the format number string, and sends the resulting text to the port terminated with chr(13).

If the text field "Static binary" has any text in it, the normal broadcast to com feature will not execute, but instead the static binary data is sent. How this feature operates is, that it expects tabulated integer data in the text field. A single line with tabulated (space separated) integers gets converted to an array of byte and sent to the port as binary data. The text field allows for multiple lines. This feature is meant to be used with https://www.controlanything.com/ and similar relay boards, that use binary arrays as their command language.

Turn on immediate relay state update (commanded actions will take place immediately):
254 25

The following line addresses bank 1 in the relay board:
254 49 1

Turn off all existing relays (connections might exist from previous nodes):
254 29

The following line turns on relay 1 above addressed bank:
254 8

The following line addresses bank 2 in the relay board:
254 49 2

Turn off all existing relays (connections might exist from previous nodes):
254 29

The following line turns on relay 1 above addressed bank:
254 8

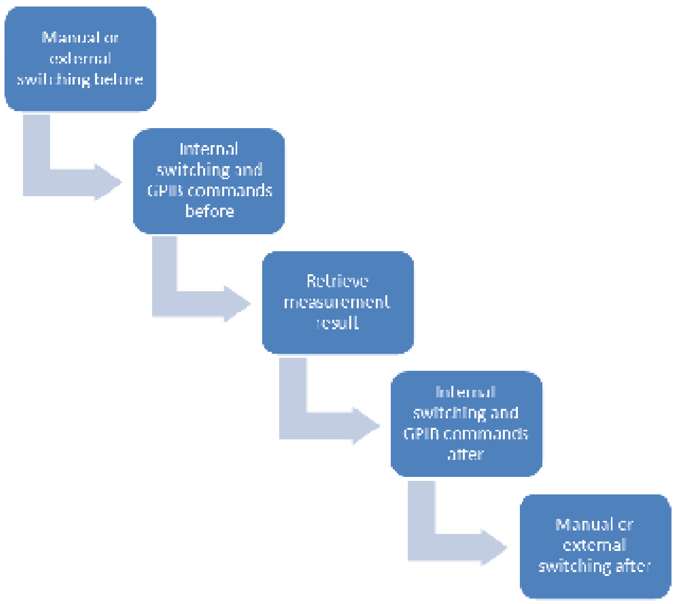
With the above lines, relay 1 on both bank 1 and bank 2 is turned on. The next node could perform some operation on the established connection scheme.

2.37 Signal switching

For simple measurements required amount of nodes is typically low and each measured node is connected directly to the instrument measuring that node. For example in conductance versus temperature one node controls the impedance spectrometer that connects directly to the sample, and another node connects directly to the furnace controller to read temperature. Such setup requires no switching at all, and in the program this is called **no switching**.

In many cases more information is needed, for example in Seebeck coefficient measurements. To avoid need to have one instrument for each measured property, many multimeters have a number of internal channels to connect to, and GPIB commands to let the instrument know which channel to measure. This is the second type of switching possible in Omega, called **internal switching**. Each measured property is connected to a channel in the multimeter and the instructions to the instrument what channel to measure are sent with node's GPIB before and GPIB after commands.

In some cases impedance spectrometer or another instrument that does not have internal channels needs to connect to more than one sample or needs to rotate the contact points to the sample for measuring van der Pauw setup. For these cases the program supports **manual switching** or **external switching.** Manual switching is, surprisingly, manual; before and after measuring a point, Omega pauses and prompts the user to switch cables. External switching allows user to send commands to an additional GPIB controlled device that can takes care of connecting the right instrument to the right sample with correct order of cables.


Manual or external switching before
Internal switching and GPIB commands before
Retrieve measurement result
Internal switching and GPIB commands after
Manual or external switching after

Each type of switching action takes place both before and after measuring the point. The before part connects in a correct way while the after part undoes the connection after measurement. Undoing the connection is important in case there are other measurements running.
See appendix and examples for description of cabling schemes and example commands to send.

2.37.1 Internal

Internal switching for the multimeter role is done using the GPIB before and after commands. For each supported instrument commands to close (connect) and to open (disconnect) channels can be found in the default GPIB commands provided by the software. They are comment lines that can be activated by removing the comment character(s) on the beginning of the line. The channel to connect to is a number in the command that can also be altered.

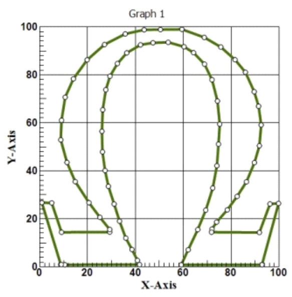
2.37.2 Manual

It is a good idea to mark all cables with visible distinct labels, and make switching instruction scheme prior to measuring so that the switching itself takes place relatively quickly without mistakes. Some examples can be found in the Measurement examples chapter.

2.37.3 External

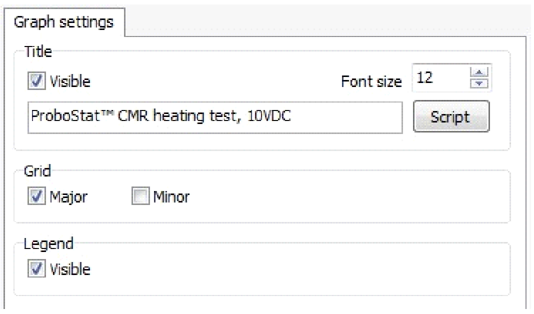
To use external switching the user must know the switching instrument well. Typically the instrument has MUX and a Multiplexer, within different modules. External switching has its own tab in the node settings, where a device is selected, and the GPIB commands are defined. Some examples can be found in the Additional devices information chapter.

3 Graphs



Measurement can have any number of graphs. A graph holds axes, markers and series and the series contain the instructions of how to display the data and possibly if any calculations are done to it.
The looks of the graph and the look of all the items on it can be customized according the user preferences. The default look of a graph is close to a publishable, and it is possible to save the graph as an image, or export all the plotted data at any time.
Right click on a graph will bring up a popup menu, with the following options.

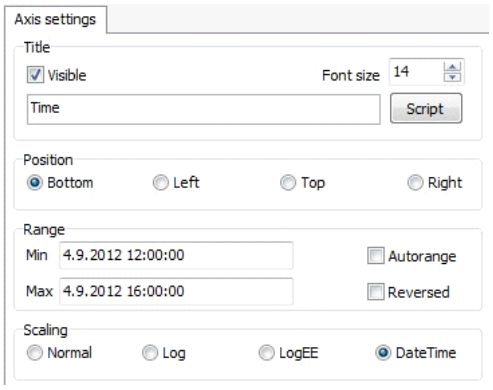
Menu title	Explanation
1:1 Proportions	Hides all but the clicked graph and resizes it to 1:1 width-height ratio. Use this before saving the graph as picture in case the picture is meant to be published and is required to be equal width and height. It is good to resize the window to have the image as close to the intended final size; as resizing it later may reduce the quality. To resume other graphs just click the measurements list on the right on a measurement or on a graph.
1:1 and zero start main axes	Makes both main axes to start from zero and to end in same value, and turns off auto scaling. The two main axes are the axes created by default when a graph is created.
Save image as…	Opens a save file dialog to specify folder and filename. Saves the clicked graph as bmp, jpeg and png.
Save data as…	Opens a save file dialog to specify folder and filename. Saves all the data of all the series on the graph to a csv format file. Includes also measurement name, graph name, series names, expressions and assigned axis names.
Show series variables	Opens a windows with all sorts of information about all the series on the graph.
Add marker for…	Inserts arrow and text assigned to series specified. See more on markers later in this chapter.


Graph settings
Title
☑ Visible Font size 12
ProboStat™ CMR heating test, 10VDC

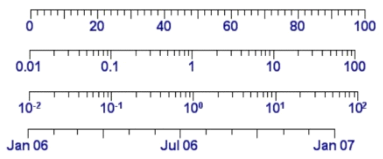
Grid
☑ Major ☐ Minor
Legend
☑ Visible

Title of the graph is shown above the graph centered or it can be hidden by unchecking the visible property. The font size and the text can be set by the properties shown on the picture, or for more advanced caption control a caption editor is brought up by clicking the Script button. The caption editor allows features such as sub and superscript, bold, italic, under lined, special characters, color etc.
Minor (dense) and major grid lines can be shown or hidden by grip properties.
Legend box showing all the names and line styles of the series can be visible or hidden by the property.

3.1 Axes


Axis settings
Title
[✓] Visible
Font size: 14
Time
Script
Position
[● Bottom] Left Top Right
Range
Min: 4.9.2012 12:00:00
Max: 4.9.2012 16:00:00
[ ] Autorange
[ ] Reversed
Scaling
Normal Log LogEE [● DateTime]

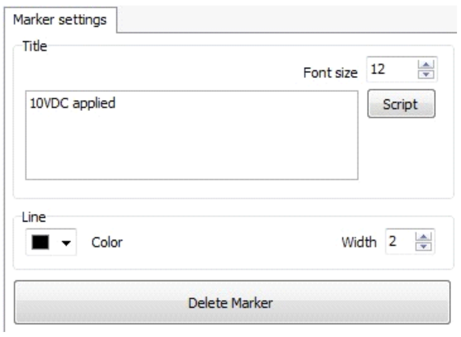
An axis has a range of properties both for the aesthetics and for the behavior.
For title same applies as for the graph title.
Position property defines on which side of the graph an axis resides.
Range property is by default set to auto range, which will automatically scale to show all data from the series assigned to this axis. In auto range is disabled, the user definable values for minimum and maximum range are effective.



Reversed property changes the direction from left to right or bottom to top, to the inverse.

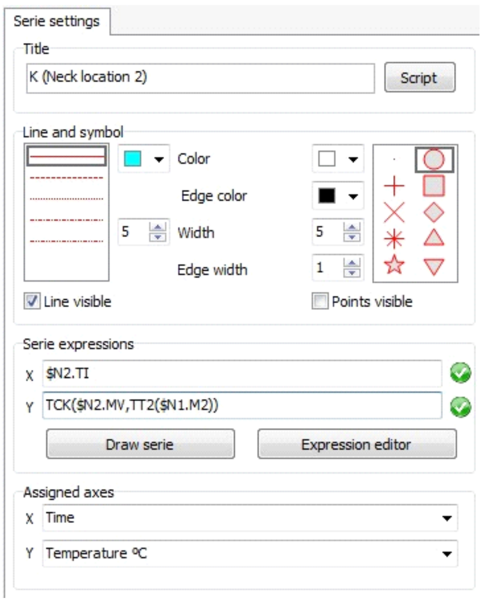
Normal scaling spreads values evenly on the axis, while Log and LogEE use logarithmic scaling. DateTime converts a value representing time into a readable format with date and/or time.

3.2 Markers


Marker settings
Title
Font size: 12
10VDC applied
Script
Line
Color
Width: 2
Delete Marker

Markers can be added by right clicking a graph and selecting ‘Add marker for…’ followed by a series caption. Marker needs to be assigned to a specific series so they keep pointing at the correct location when the graph rescales. The arrow and the text are moveable with click and drag, and just the text can be moved in which case the arrow adjusts (Drag from the round symbol). Clicking a marker will bring up the editable properties such as caption, sizes and colors to the right panel on the right. This panel also hosts a button to delete the selected marker.

3.3 Series


Serie settings

Title: K (Neck location 2)

Line and symbol

Color: 🟦

Edge color: ■

Width: 5

Edge width: 1

Color: ☐

Width: 5

[✓] Line visible

[ ] Points visible

Serie expressions

X: $N2.TI ✓

Y: TCK($N2.MV,TT2($N1.M2)) ✓

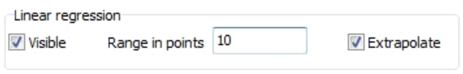
Assigned axes

X: Time

Y: Temperature ºC

Series is a representation of data. Series has expressions for X and Y coordinates which define what values are shown on the graph. At simplest, the formula is just the acronym for the raw measured data, but it can be complicated mathematical expression with functions. From the picture:
$N2.TI is the time of a measurement point from node 2.

TCK($N2.MV,TT2($N1.M2)) returns Temperature of a K-type thermocouple calculated from a voltage $N2.MV, and from a cold junction compensation temperature. The latter being calculated from resistance measured with node 1, from a thermistor.
Things such as line color and thickness can be edited as well as the point symbol, size and color. Series has a caption that will show on the legend of the graph if the legend is enabled, as usual, the caption have sub and superscript so it is easy to mark up chemical formulas on the caption.
A series needs to be assigned and axis to both X and Y-axes property, and this can be done with a dropdown box showing all possible axes of the graph.
If and when the nodes receive new data, all relevant series are automatically updated. When not receiving new data, click ‘draw series’ button to clear and redraw the whole series.
A small green check icon indicates the validity of an expression, and red fail mark indicates error in the expression. Series with erroneous expressions cannot be plotted.
‘Expression editor’ button opens a tool for designing and testing expressions, and will be covered elsewhere in the manual.


Linear regression

[✓] Visible   Range in points: 10   [✓] Extrapolate

Each series hosts a coefficient tool with settings to toggle visibility, the range and option to extrapolate the coefficient line. The coefficient is a linear least squares calculated line. The calculation always includes the last point of the series and backwards the amount defined by the ‘range in points’. The extrapolate setting will continue drawing the coefficient line left until it meets the edge of the graph.
Right clicking a graph and selecting ‘show series’ variables’ will pop-up a window with listings of all series and all the variables for each series. Series’ variables start with $S and a digit indicating the series followed by a full stop and the variable identifier.
Series values and some properties like maximum and minimum are also accessible in each series. $S1.Y will allow the user to capture Y value of a series to another expression.

3.4 Operators and functions

This information can be used in many ways but one of the most useful purposes is to automatically determine when equilibrium (or some such) conditions have been reached and to execute further orders, such as starting a waiting measurement. In particular, .LRB in general is the rate of change in Y-axis units / X-axis units, and the LRMI and LRMA indicate the minimum and maximum values in the range of the coefficient tool.
Note that the .L??? variables are not calculated unless the coefficient tool is marked visible.
Also note that the .L??? variables always refer to the current state of the coefficient tool and thus redrawing a series with .L??? variable(s) in the expression will not study the old values of the target series.

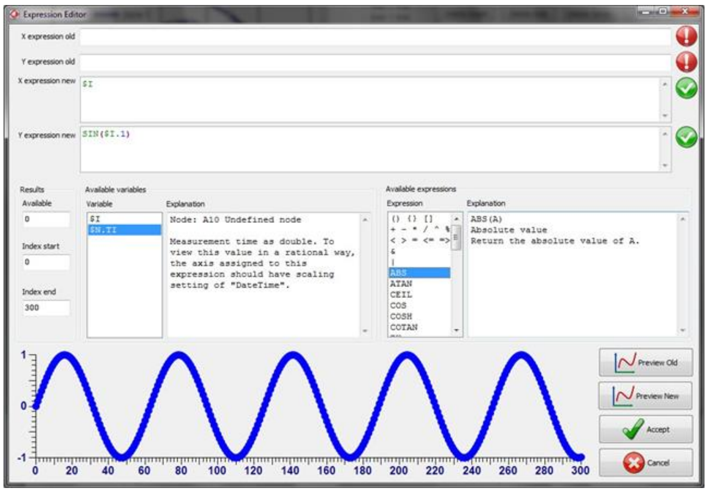
Coefficient example – known target value

Node 1 ($N1) measures furnace temperature, and is plotted in a graph as series 1 ($S1). The measurement frequency limit is 1 min, and coefficient range is 15. In addition the X-expression of the series is $N1.TM so that X-axis represents elapsed minutes, instead of the time and date ($N1.TI) which would give hard to understand slope and offset (slope would be °C/day). Also, this assumes the furnace is set to output Celsius and not Kelvin or Fahrenheit.
With such settings the following expression:
($S1.LRMI>899)&($S1.LRMA<901)
Result as 1 if all the values (in last 15+ minutes) of furnace temperature are between 899°C and 901°C, and in 0 otherwise.
The > results in 0 if false and 1 if true, and the & results 1 if both sides are true, 0 otherwise, see functions.

Coefficient example – rate of change/equilibrium

Node 1 ($N1) measures sample resistance, and is plotted in a graph as series 1 ($S1). The measurement frequency limit is 5 min, and coefficient range is 24. In addition the X-expression of the series is $N1.TM so that X-axis represents elapsed minutes, instead of the time and date ($N1.TI) which would give hard to understand slope and offset (slope would be ohms/day).
With such settings the following expression:
($S1.LRB>-0.01)&($S1.LRB<0.01)
Result as 1 if the rate of change of the measured resistance in the last two hours (5x24=120 minutes) is between -0.01 and 0.01, and 0 otherwise.

3.5 Expression editor window



X expression old:
Y expression old:
X expression new: $I
Y expression new: SIN($I.1)

Results
Available: 0
Index start: 0
Index end: 300

Available variables

Variable	Explanation
$I $N.TI	Node: A10 Undefined node Measurement time as double. To view this value in a rational way, the axis assigned to this expression should have scaling setting of "DateTime".

Available expressions

Expression	Explanation
() {} [] + - * / ^ % < > = <= => & | ABS ATAN CEIL COS COSH COTAN	ABS(A) Absolute value Return the absolute value of A.

The expression editor windows shows the current X and Y expressions of the series as '… expression old' with a visual indicator on the right if the expression is valid.
The '… expression new' is mainly what will be edited and tested in this window. 'Preview old' and 'Preview new' buttons will draw the old and new expressions at the bottom of the window. 'Accept' will assign the new expressions to the series and cancel will go back without any changes to the series.
Results available will tell you how many data points the measurement has, and Index start and Index stop will allow manual setting of the range to plot. If the Index end is left to 0, the plotting process will loop to the amount of available data.
All possible variables for the measurement in question are listed under 'Available variables' and clicking them will provide with short explanation what the expressions are. All variables start with $ sign, followed by letter N for node, and then a digit identifying which node is in question. The rest of the variable is formed of full stop and then a two character data type identifier and possibly another full stop and a character. The available data type depends of the settings of the node, but a .TI (for Time) is common to all nodes. If a Node 1 was set to measure temperature from Eurotherm controller, the available variables would be $N1.TI and $N1.ET and of Node 2 was set to measure voltage with multimeter, available nodes would also include $N2.TI and $N2.MV. For the impedance related measurement there are more variables; If Node 3 was set to measure continuous impedance, available nodes would also include $N3.TI, $N3.IC.R for real part of the impedance, $N3.IC.X for imaginary part of the impedance and $N3.IC.F for frequency used to measure.
Similarly all possible math operators and functions are shown in another box and clicking them will show a brief description of what they do. The operators and functions are explained in detail later in this chapter.

Expression example

Measurements may require more than just plotting one node and the raw measured data; often a range of other nodes are required as well as some calculations, combining the data as well as data conversions. The parser has a range of built in functions for advanced tasks such as solving van der Pauw equation, for thermocouple voltage to temperature conversions and such.
To get accurate temperature result, one would typically read a thermocouple voltage with one multimeter channel, and a thermistor resistance with another. The thermistor would be located adjacent to the junction where the dissimilar thermocouple or thermocouple compensation leads end, and similar leads such as copper wires start. The resistance of thermistor is a known function of temperature. One thermistor of specific type is delivered with the software by default and more on request.
From these two nodes the actual temperature can be calculate d using the built in functions. Plotting such temperature could look like this:
X: $N2.TI
Y: TCK($N2.MV,TT2($N1.M2))
The TT2 function first converts the measured thermistor resistance to a temperature, and the TCK function takes the measured thermocouple voltage (K-type thermocouple), and the cold junction temperature as parameters and returns the temperature at the hot junction of the thermocouple.

4 Other features

A collection of other features that require explaining, but do not deserve main level page.

4.1 Multiple instances of Omega

Only run one instance of Omega on one computer at one time.

Multiple instances of Omega can run on same computer simultaneously, but it is highly unadvisable: Validating the license dongle will occasionally fail, also if the separate instances try to access same devices and instruments at the same time; it will time out and fail. It is however left to be possible in case it is required as advanced user may benefit from such functionality.

Running Omega at different locations on a computer can be beneficial, but can also get confusing with the save locations and autosaves and so forth.

4.2 Bug report

In case of unexpected events in the program, Omega will display a dialog about the situation. In most cases it is safe to click 'continue program'. These bug reports get emailed to the author automatically in case internet access is available. If no internet access is avalaible the report will get saved on the disk as a text file.

4.3 Leak report

When the software is closed it gathers and presents a report about how well the software performed. The report is called 'Leak report' after memory leaks. It is programming jargon and means aproximately: How many resources the software reserved, but did not release after use. All unreleased resources will use up a part of the system memory so it is important for programmer to be aware of these leaks. Normal user can close the window without saving the report.

5 Tutorials

5.1 Thermocouple measurement

5.1.1 Principle

The two legs of a thermocouple are each made of different material and each produce a voltage when exposed to a temperature gradient. Since the tip of the thermocouple is connected, we can measure the voltage of the whole circuit; e.g. the sum of the two voltages from each leg, and the total voltage is called thermocouple voltage, or from here on 'voltage'. There are many different types of thermocouples (made from different combinations of materials each with their perks and boons, but here we concentrate on K- and S types. So the thermocouple is formed of a tip, and two separated legs. The tip is usually located at the point of interest and the two legs in ambient temperature.

A compensation cable is something that produces exactly same voltage as thermocouple of the same type, but has lesser range towards high temperatures. Some thermocouple materials are expensive so sometimes thermocouples themselves are extended at the cold end with use of compensation cables. Polarity of connections is of utmost importance.

Two wires that are of identical material exposed to a temperature gradient do generate identical and opposite voltage that always result to zero (when connected to a same circuit).

A cold junction is considered to be the point where the dissimilarity ends and similarity begins in a circuit i.e. the thermocouple or thermocouple compensation wire ends, and two leads of identical material begin. Sometimes the aforementioned go all the way to the measurement instrument, but the temperature inside such instrument may be unstable or difficult to measure. In this example the measurement instrument is connected to the thermocouple circuit with identical leads.

The thermovoltage generated is a function of temperature gradient between the legs and the tip, and the total temperature. A gradient of 50°C at room temperature produce different voltage than same gradient at 1000°C. This makes interpreting the voltage bit more complicated, as all lookup tables and conversion functions are based on the assumption that the cold junction is in an ice bath. This is inconvenient, so many devices have a way to know or estimate the ambient temperature and take it into account when calculating the tip temperature.

The thermocouple conversion functions in Omega can take in the cold junction temperature as static number; user estimation, or as measured and calculated parameter (denoted as another expression)

When the cold junction temperature is known, a reverse lookup is done with the thermocouple function to see how much voltage would such temperature produce. This voltage is then added to the measured thermocouple voltage, and the new total is fed to the conversion function and a corresponding temperature is received.

The functions in Omega user interface to calculate temperatures from thermocouples are TCS(V; T) and TCK(V; T) for S- type and K type accordingly. V is the thermocouple voltage and T is the cold junction compensation temperature in °Celsius. If the measurement does not need to be accurate, the user can just enter for example 25 as the T parameter assuming the cold junction is in temperature controlled room with temperature always around 25°C.

If the measurement needs to be accurate, the T can be an expression that calculates the cold junction temperature from a resistance measured from a thermistor (chip that has resistance as function of temperature) located right next to the cold junction, or the user can enter 0 and place the junction in ice bath.

The TT2(R) function converts measured resistance to temperature in °C.

Let us assume the user has set up Node 1 to measure the thermocouple voltage, and optionally Node 2 to measure the resistance of thermistor located at the cold junction.

To calculate the generic temperature of the S-type thermocouple tip when the cold junction is not controlled, the user would use expression TCS($N1.MV,25) in the graph section to plot it or elsewhere where such expressions are accepted in the program. The $N1.MV signifies the measured voltage from node 1, and the 25 signifies the users best estimate of the temperature of the cold junction. Needlessly to say, when someone opens the windows in the laboratory, the result of this expression changes even if the temperature at the thermocouple tip remains the same.

So to guard against this and to get the actual accurate temperature one needs to place the thermistor (one provided with each instance of Omega) next to the cold junction, and set up a node to measure the resistance.

In such case the T term would be replaced with TT2(R), and the R with $N2.M2 signifying the measured (2-wire) resistance of the measurement node 2. In total the expression would look like TCS($N1.MV, TT2($N2.M2))

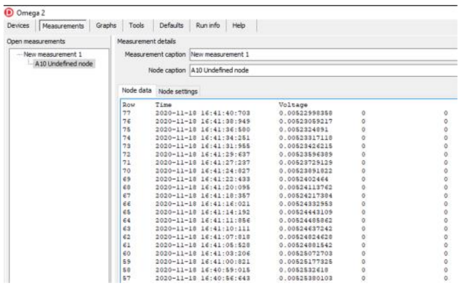
5.1.2 Setup

The user needs a multimeter with multiple channels or two separate devices for the two measurements made. We assume the user has Keithley 2000 with 10 channel card fitted. The thermocouple is connected to channel 1 and the thermistor is connected to channel 8. The instrument has a physical button to select front or back terminals, since the card is in the back, back terminals option must be selected with the mechanical selector.

Node is se set up to be voltage measurement with Keithley 2000. In the GPIB commands field before the user needs to enable the 'channel close 1' command, and in the GPIB commands after measurement field 'open all channels' command. Likewise for the Node 2 the user must select 2-wire resistance measurement on Keithley 2000, and close channel 8 and open it accordingly.

The actual commands to open and close the internal channels are:
:ROUT:CLOSE (@1)
:ROUT:OPEN:ALL
Where the 1 is the channel to be closed (i.e. connected).

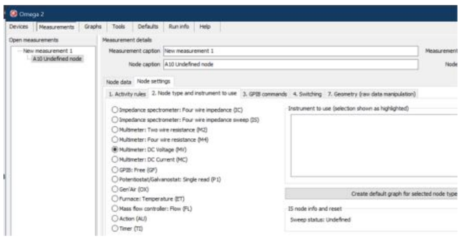
5.1.3 Converting into temperature



Measurement caption: New measurement 1
Node caption: A10 Undefined node

```
Row    Time                        Voltage
77     2020-11-18 16:41:40:703     0.00522998358     0     0
76     2020-11-18 16:41:38:949     0.00523059217     0     0
75     2020-11-18 16:41:36:580     0.0052324891      0     0
74     2020-11-18 16:41:34:251     0.00523317118     0     0
73     2020-11-18 16:41:31:955     0.00523426215     0     0
72     2020-11-18 16:41:29:637     0.00523596389     0     0
71     2020-11-18 16:41:27:237     0.00523729129     0     0
70     2020-11-18 16:41:24:827     0.00523891822     0     0
69     2020-11-18 16:41:22:433     0.0052402464      0     0
68     2020-11-18 16:41:20:095     0.00524113762     0     0
67     2020-11-18 16:41:18:357     0.00524217384     0     0
66     2020-11-18 16:41:16:021     0.00524332953     0     0
65     2020-11-18 16:41:14:192     0.00524443109     0     0
64     2020-11-18 16:41:11:856     0.00524485862     0     0
63     2020-11-18 16:41:10:111     0.00524637242     0     0
62     2020-11-18 16:41:07:818     0.00524824628     0     0
61     2020-11-18 16:41:05:528     0.00524881542     0     0
60     2020-11-18 16:41:03:206     0.00525072703     0     0
59     2020-11-18 16:41:00:821     0.00525177325     0     0
58     2020-11-18 16:40:59:015     0.0052532618      0     0
57     2020-11-18 16:40:56:643     0.00525380103     0     0
```

Once the user is able to acquire the thermocouple voltage data as seen in this figure it is time to add a graph to the measurement. The Graph allows user to plot either the acquired raw data or further process and display the resulting data.

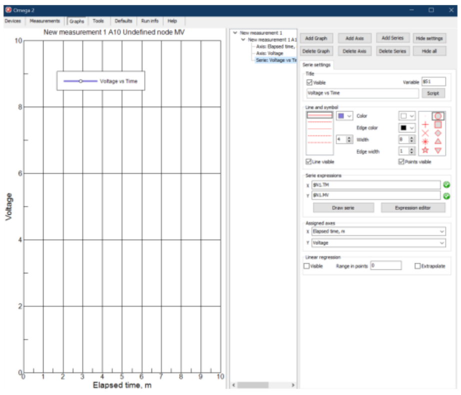


Measurement caption: New measurement 1
Node caption: A10 Undefined node

- ○ Impedance spectrometer: Four wire impedance (IC)
- ○ Impedance spectrometer: Four wire impedance sweep (IS)
- ○ Multimeter: Two wire resistance (M2)
- ○ Multimeter: Four wire resistance (M4)
- ● Multimeter: DC Voltage (MV)
- ○ Multimeter: DC Current (MC)
- ○ GPIB: Free (GF)
- ○ Potentiostat/Galvanostat: Single read (P1)
- ○ Gen'Air (OX)
- ○ Furnace: Temperature (ET)
- ○ Mass flow controller: Flow (FL)
- ○ Action (AU)
- ○ Timer (TI)

Sweep status: Undefined

New graph can be created either by clicking the "Create default graph for selected node type". In this case a graph will be created that plots measured voltage on Y axis against elapsed time on X axis, or with some more effort on the Graphs tab by clicking add graph, and series, and setting their properties manually.

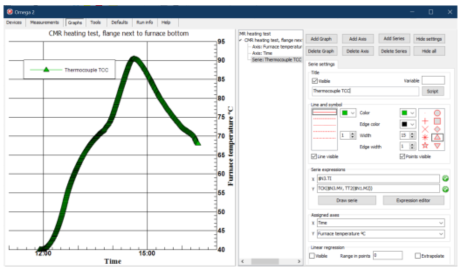


☑ Visible Variable: $S1
Voltage vs Time
Width: 4; Width: 8; Edge width: 1
☑ Line visible ☑ Points visible
X: $N1.TM
Y: $N1.MV
X: Elapsed time, m
Y: Voltage
☐ Visible Range in points: 0 ☐ Extrapolate

What is plotted is determined by the serie expressions. By default the $N1.TM for X is the time elapsed since the start of the node 1 in minutes and for Y the $N1.MV is the measured voltage.

To plot the temperature the measured voltage needs to be converted using suitable functions explained in detail earlier.

To convert the voltage measured from S-type thermocouple into temperature (without proper cold junction compensation) the X expression must be changed to TCS($N1.MV,25), and the serie and axis captions and labels changed accordingly.



☑ Visible Variable:
Thermocouple TCC
Width: 1; Width: 15; Edge width: 1
☑ Line visible ☑ Points visible
X: $N3.TI
Y: TCK($N3.MV, TT2($N1.M2))
X: Time
Y: Furnace temperature °C
☐ Visible Range in points: 0 ☐ Extrapolate

In this image the Thermocouple type is K, and the voltage is measured on Node 3, and the cold junction compensation resistance is measured on node 2.

Function for S-type thermocouple is TCS(A,B) and for K-type TCK(A,B) where A is the thermocouple voltage and B is the cold junction temperature in °C. TT2 is a resistance to temperature conversion function for the thermistor type provided with Omega. This function can be replaced at first with estimated cold junction temperature, a static value of for example 25.

6 Advanced measurements

By combining nodes and expressions it is possible to measure methods described in the ProboStat manual, such as Conductivity vs. Temperature or partial pressure, Seebeck coefficient, 4-point conductivity on disc or on a bar, impedance spectroscopy, combined Seebeck coefficient and resistance measurements, transport number measurements, fuel cell tests and so forth. After reading the measurement examples later in this manual, it will be easy to adjust the measurements to the users own preferences and design completely new methods as well. With AU type node it is also possible to automate certain aspects of the measurement such as furnace temperature control, cable switching, gas mixing, powering up resistance heaters etc.

Library of files

6.1 Multichannel multimeter preparation

These pictures are for Keithley 2000 with 10 channel scanner card.

For convenience connect all 10 channels to terminals of your choice. On this picture channels are connected to miniature thermocouple socket.

Use same type and length of wire for all terminals to avoid thermovoltages. Twist together the wire pairs for each channel and mark the pair with a channel number.

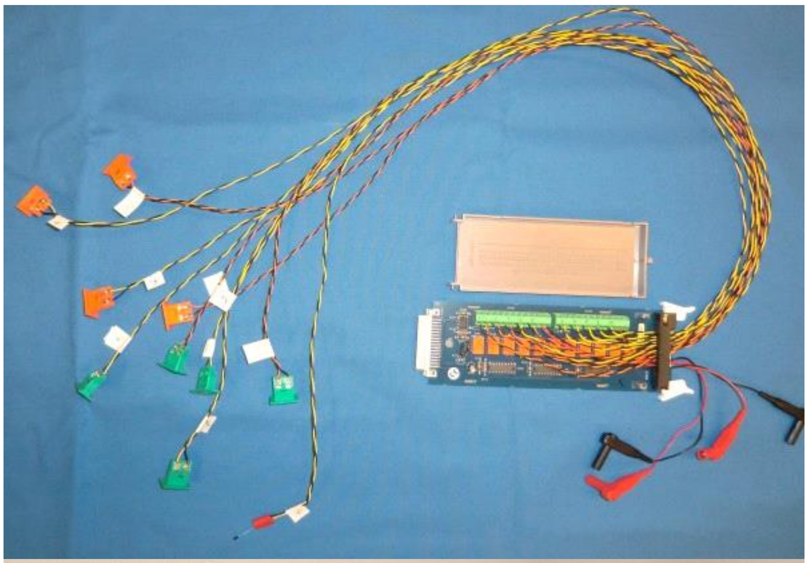
End each cable pair with a convenient way to quick connect them to another leads such as Thermocouple wires or compensation cables or BNC cables.

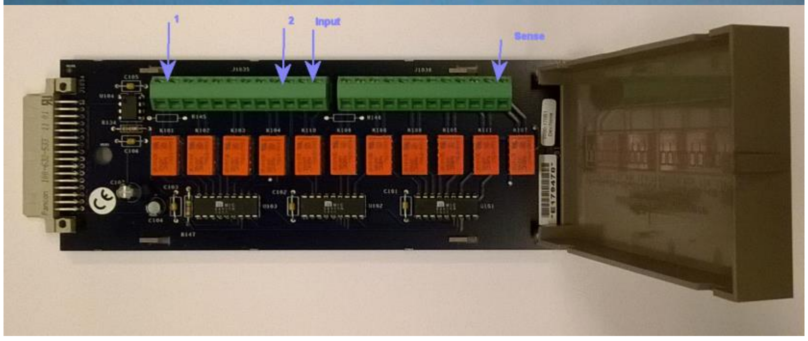
For temperature measurements the 'cold junction compensation' is necessary. It can be either estimated or measured. With Omega it is possbile to measure accurately with a bead thermistor. Request a bead thermistor from NorECs and connect it to one of the channels, e.g. channel 10. Measure channel 10 with M2 type node for 2-wire resistance. The software includes function TT2 to convert this resistance to temperature.

The purpose for the thermistor is to observe the temperature of the point where thermocouple- or thermocouple compensation wires change to the (between themselves identical) wires coming from the multimeter. Knowing the temperature of this point is essential for accurate thermocouple measurements. Keep all of the wire transitions and the bead thermistor close to each other, perhaps even wrap them inside bubble foam or such to get more even temperature gradient between all of them.

With this card it is possible to measure voltage, current, 2-wire resistance and 4-wire resistance or any combination of these.

Channels 1-5 are on bank 1, and channels 6-10 on bank 2. Bank 1 is connected to the input on back of the instrument, while bank 2 is connected to the sense on the back of the instrument. More details about the card in the manual accompanying the switch card.





6.2 Multichannel multimeter preparation v2

These pictures are for Keithley 2000 with 10 channel scanner card.

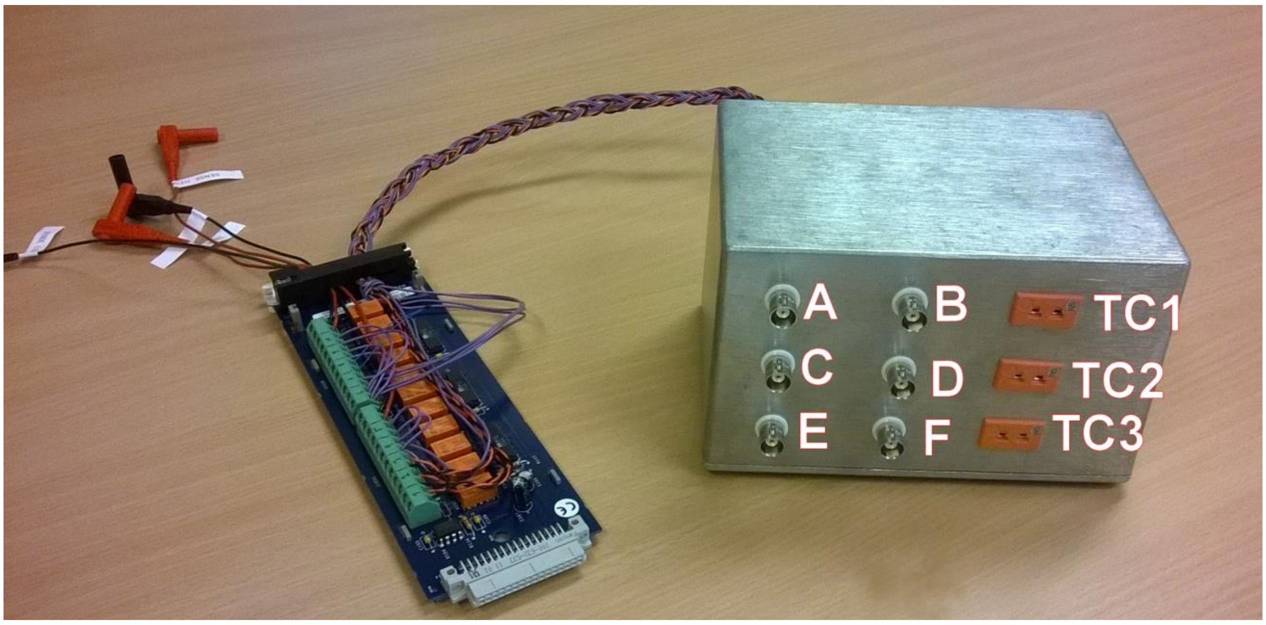
For convenience all 10 channels from the Multimeter are connected to terminals on the box.

Multimeter channel	Box terminals	Node in template	Notes
1	TC1	N$1.MV	Use with Function TCS
2	TC2	N$2.MV	Use with Function TCS
3	TC3	N$3.MV	Use with Function TCS
4	A-B	N$4.MV	Voltage between A-B
5	B-A	N$5.MV	
6	C-D	N$6.MV	Voltage between C-D
7	E-F	N$7.MV	Voltage between E-F
8	Thermistor	N$8.M2	Use with function TT2
9	C-D	N$9.M4	With 4Ω mode used together with Channel 9 for normal 4-v
10	D-C	N$10.M4	With 4Ω mode used together with Channel 10 for reversed

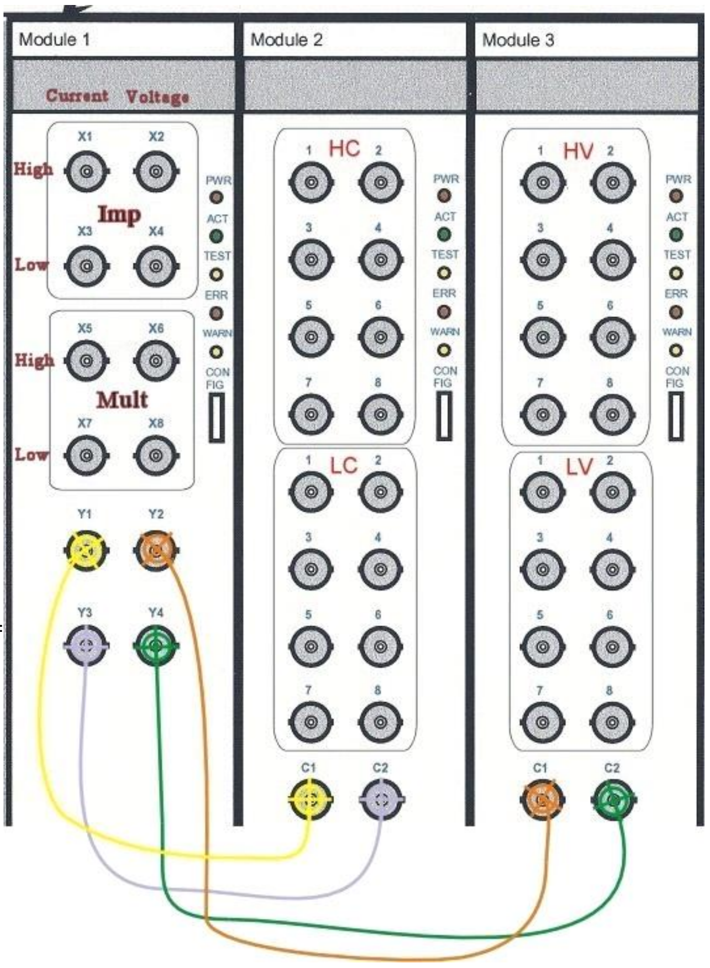
Some examples using the Omega template "K2000 channel preparation 2.mea":

- TCS($N2.MV, TT2($N8.M2)) to get temperature of TC2

- N$7.MV for voltage measured between E-F
- ($N9.M4+$N10.M4)/2 to get four wire resistance of AC-BD with reverse and average (reverse current direction, reverse electrode order)



6.3 External switching instruments



External switching instruments do not have much in common amongst themselves. They are often modular and can host a number of different modules in various numbers of slots. What features are needed and in which slots they are installed depends of the application, but in general the switch should be able to select and connect 4 out of 8 BNC cables and connect them to a range of different targets. In the example device that range is 8. For this, 3 modules and some cabling connecting the modules are needed. In addition to the switching instrument markings (X1…X8, Y1...Y4, C1…C2, Module 1...3) some labels were added to the picture to help identify and conceptualize the regions. The colored lines between modules represent user installed permanent BNC cables.

On slot 1 the module installed is something called a matrix. It can connect any combination of the X contacts to any combination of Y contacts. The slots 2 and 3 host modules called multiplexers. They connect one lead to one of eight. The modules in the picture are actually double multiplexers; so they hold 2x 1 to any of 8, for example module 2 the C1 connects to one in the HC region, and C2 connects to one in the LC region.

The double regions in module 2 and 3 represent the 4 electrodes coming from the sample, and are connected to the Y1-4 on the Matrix module with BNC cables illustrated as colored lines. The impedance spectrometer terminals are connected to the X1-4 and the multimeter (without capability for internal switching) terminals to the X5-8. With this setup we have ability to connect either multimeter or impedance spectrometer (or other devices), their four terminals in any desired order to any of the 4 electrodes coming from any of 8 different samples allowing straight, reversed, rotated or short circuited electrodes, and enabling thus normal measurements as well as reverse and average and van der Pauw method.

With the setup it is easy and fast to share high accuracy multimeter and impedance spectrometer between eight ProboStat sample holders; the software takes care of giving focus to each of the running measurements in turns connecting, disconnecting and rotating electrodes according to given instructions and measurement needs.

All unidentified instruments on the GPIB network are labeled ‘Unknown’ and these instruments can be selected for External Switch role and controlled entirely by the user defined GPIB commands.
The external switching examples and external switching GPIB code in this manual refer to an exact case; instrument on the picture and the examples is Pickering 10921 (with described modules setup). GPIB commands for this instrument on same line must be separate with ; or broken into several lines.

External switching example for conductivity measurement and for van der Pauw measurement.

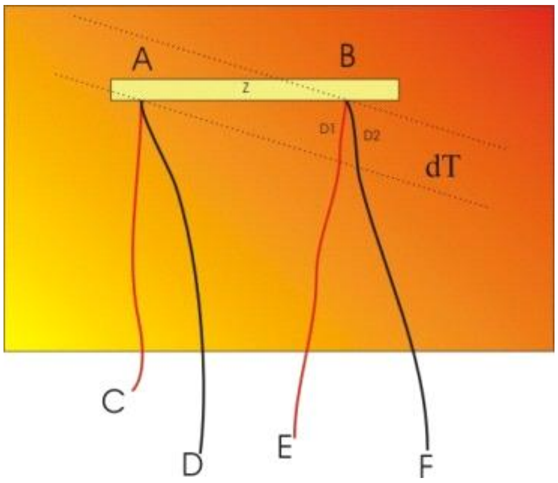
Need	Phase	Command	Explanation
Conductivity measurement with impedance spectrometer for sample connected to channel 7. **Node 1**	Before measuremen t	ARESET 1 ARESET 2 ARESET 3 MC 1,1,1;MC 1,2,2 MC 1,3,3;MC 1,4,4 Chan 2,1,7 Chan 2,2,7 Chan 3,1,7 Chan 3,2,7	Open all connections on module 1 Open all connections on module 2 Open all connections on module 3 On module 1 connect X1 and Y1; On module 1, connect X2 and Y2 On module 1 connect X3 and Y3; On module 1 connect X4 and Y4. Connect C1 to HC 7 on module 2 Connect C2 to LC 7 on module 2 Connect C1 to HV 7 on module 2 Connect C2 to LV 7 on module 2
	After measuremen t		Nothing needed (As long as possible other measurement do not assume anything)
In case van der Pauw measurement is required, adding node 2 would achieve rotating the electrodes from HC HV LV LC one step ahead. Modules 2 and 3 are already connected to the sample so the commands for them can be omitted.	Before measuremen t	ARESET 1 MC 1,1,2;MC 1,2,3 MC 1,3,4;MC 1,4,1	Open all connections on module 1 On module 1 connect X1 and Y2; On module 1, connect X2 and Y3; On module 1 connect X3 to Y4; On module 1 connect X4 to Y1.
	After measuremen t		Nothing needed (As long as possible other measurement do not assume anything)

External switching example for voltage measurement and for voltage measurement with reverse and average.

Need	Phase	Command	Explanation
Voltage measurement forward **and reverse for sample connected to channel 5.** **Node 1**	Before measuremen t	ARESET 1 ARESET 2 ARESET 3 MC 1,6,2;MC 1,8,4 Chan 3,1,5 Chan 3,2,5	Open all connections on module 1 Open all connections on module 2 Open all connections on module 3 On module 1 connect X6 to Y2; On module 1 connect X8 to Y4. Connect C1 to HV 5 on module 3 Connect C2 to LV 5 on module 3
	After measuremen t		Nothing needed (As long as possible other measurement do not assume anything)
In case reverse and average is required, adding node 2 for reversed **voltage measurement.**	Before measuremen t	ARESET 1 MC 1,6,4;MC 1,8,2	Open all connections on module 1 On module 1 connect X6 to Y4; On module 1 connect X8 to Y2.
	After measuremen t		Nothing needed (As long as possible other measurement do not assume anything)

6.4 Reverse and average

Reverse and average is achieved with two MV nodes and some switching.

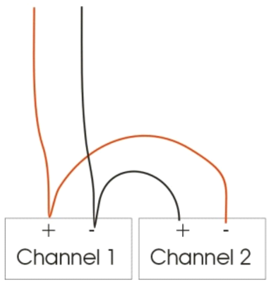


When only wanting the sample voltage and the accuracy of voltage measurement is critical, it is best to assume that unknown temperature gradients exist over the sample and the electrodes, and prepare for them.
A single voltage measurement (with electrodes C and E of similar material) will in addition to sample voltage Z return also the voltage D1 on the electrode.
To eliminate the thermal emf of the electrodes on a voltage measurement, one would measure sample voltage with node 1, and again the same sample voltage with node 2, but with inverted electrodes.
Subtracting the second result from the first and dividing by 2 will eliminate the $EMF_{electrode}$ contribution and result as pure sample voltage.

X: ($N1.TI + $N2.TI)/2	Mean time of the two measurement nodes
Y: ($N1.MV - $N2.MV)/2	Pure sample voltage

Furthermore, when interested in sample voltage without any contribution from the possible temperature gradient, such as fuel cell test or gas permeation measurement, the same method can be used to reveal any significant gradient over the sample by plotting:

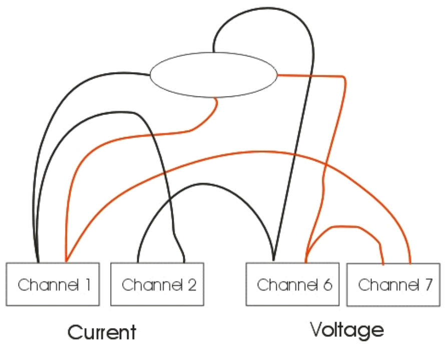
X: ($N1.TI + $N2.TI)/2	Mean time of the two measurement nodes
Y: $N1.MV – ABS($N2.MV)	Distance from zero indicates temperature gradient over sample. Zero may indicate no temperature gradient exists, but it may that the sample emf just cancels the electrode emf if the coefficients happen to be close to each other.



For best results this should be, at first, tested at target temperature but with exactly same atmosphere on both sides of the sample, and only then with the target atmospheres.
The cabling scheme can be set to be manual, internal, external or even direct if measuring each node with a separate voltmeter.
The image: A suggestion for connections for internal switching.

6.5 Van der Pauw

4-points van der Pauw DC resistance measurements



Van der Pauw measurement with multimeter utilizes two 4-point resistance measurements and the VDP function.
For example on a Keithley 2000 with scanner card, when a resistance of a disc sample is measured with channel 1 (node 1 as M4), it pairs up with channel 6. Channel 1 act as Input/Current and channel 6 acts as Voltage/Sense. Another node (Node 2 as M4) measures similarly but with channel 2 and 7. With connection schematics as in the picture, the Node 2 has rotated the electrodes one step ahead. Solving for the bulk resistivity involves VDP function and disc thickness.
EXP: VDP($N1.M4, $N2.M4)*disc thickness*correction factor

6.6 4-points van der Pauw AC resistance measurements

As Impedance spectrometers typically do not have channels, the electrode rotation must be arranged by manual (hand) or external switching (switching instrument). See switching instrument introduction chapter for more information.
When using AC instrumentation for van der Pauw measurements, it is necessary to ensure that the frequencies used are such that the imaginary part of the impedance is negligible.
It may be useful to plot the measured resistances and reactances individually as well as the solved resistivity.

Series	X-Axis	X-Expression	Y-Axis	Y-Expression
Node 1 resistance	Time	$N1.TI	Resistance	$N1.ICR
Node 1 reactance	Time	$N1.TI	Reactance	$N1.ICX
Node 2 resistance	Time	$N2.TI	Resistance	$N2.ICR
Node 2 reactance	Time	$N2.TI	Reactance	$N2.ICX
Resistivity	Time	($N1.TI+$N2.TI) /2	Resistivity	VDP($N1.ICR, $N2.ICR)*disc thickness

6.7 Thermocouples

To measure thermocouple temperatures three things must be known, the thermocouple type, the thermocouple voltage and the temperature at the cold end of the thermocouple or thermocouple compensation wire. Thermocouple voltage can be measured with MV node (node 1), and the cold end temperature can be measured with M2 node (node 2) on a thermistor. Locate the thermistor adjacent to the junction where the dissimilar thermocouple or thermocouple compensation leads end, and similar leads such as copper wires start. The resistance of thermistor is a known function of temperature. One thermistor of specific type is delivered with the software by default and more on request.
From these two nodes the actual temperature can be calculate d using the built in functions. Temperature plotting expression could look like this:

X:	$N1.TI	With X-axis type set to datetime.
Y:	TCK($N1.MV,TT2($N2.M2))	Temperature at the tip of the thermocouple

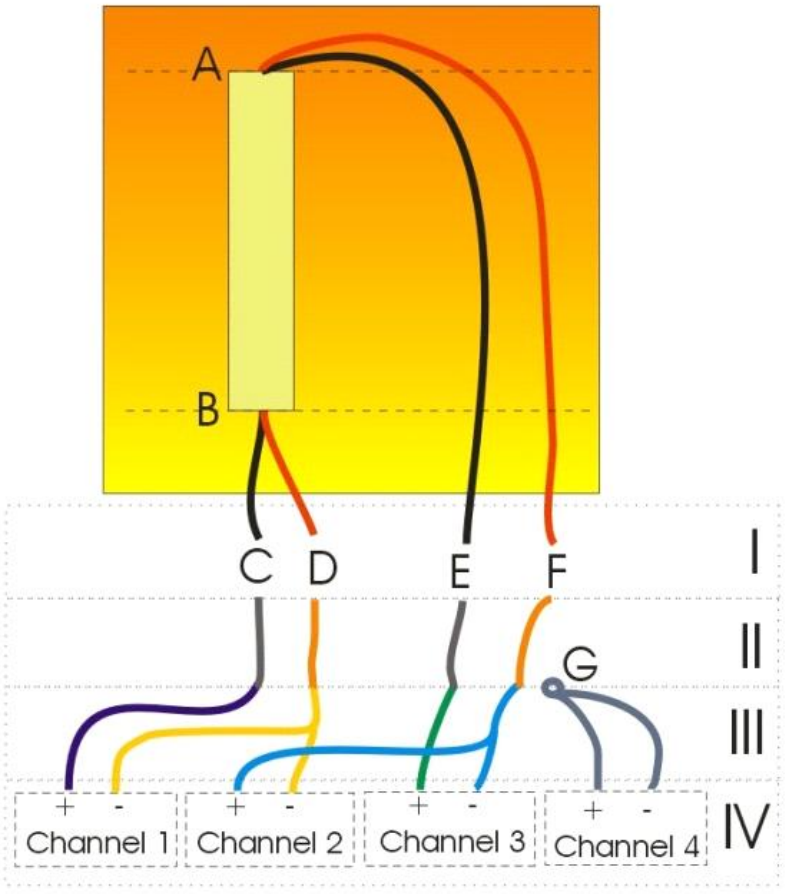
The TT2 function first converts the measured thermistor resistance to a temperature, and the TCK function takes the measured thermocouple voltage (K-type thermocouple), and the cold junction temperature as parameters and returns the temperature at the hot junction of the thermocouple.
The software supports TCK and TCS for K and S type thermocouples accordingly. Other types can be added on request. These functions first convert the temperature given as parameter to voltage of that thermocouple type, between 0°C degrees and the given temperature, add this voltage to the measured voltage, and convert that sum to a temperature using high degree polynomial fitted from well proven thermocouple voltage tables. These functions are not valid for below zero temperatures.
To validate these functions, TCK for example, plot X expression as $I looping from 0 to 50000 representing the microvolts for the positive temperature range of the thermocouple, and for Y plot the TCK function with cold junction compensation of 0 degrees, like Y: TCK($I/1000000, 0)
In case using internal switching, and for example Keithley 2000 with scanner card, the thermocouple connected to channel 1 and the thermistor to channel 10, the switching commands (excluding all other GPIB command being sent) would be:

- Node 1, GPIB before: :ROUT:CLOSE (@1) Connect channel 1
- Node 1, GPIB after: :ROUT:OPEN:ALL Open all channels
- Node 2, GPIB before: :ROUT:CLOSE (@10) Connect channel 1
- Node 2, GPIB after: :ROUT:OPEN:ALL Open all channels

Add thermistor CJC

6.8 Seebeck coefficient

Seebeck coefficient measurement explained here is done with just S-type thermocouples C-D and E-F connected to points B and A respectively. The sample voltage$_{ab}$ is measured using the Platinum leads of the thermocouple and the temperatures at A and B using the respective thermocouples.



Considering the schematics and roman numbers on the right:

Unmarked ares with orange gradient is the sample inside the sample holder which in turn is inside a furnace.

Zone I is the cold area of a sample holder such ProboStat, where typically the thermocouple leads are switched to thermocouple compensation cables which lead to the outside wall of the sample holder and the panel mounted thermocouple sockets.

Zone II is where thermocouple compensation cables connect to the wall mounts of the sample holder, and lead to a region where the compensation cables are switched to leads of identical material. Where the thermocouples or compensation cables change to normal electrical wires, is where the thermistor G is located.

Zone III is where wires of identical material connect to a multimeter channels.

Zone IV is the multimeter and the scanner card inside it.

The measurement in Omega will need 4 nodes, and for sake of clarity we instruct node 1 to measure channel 1, node 2 to measure channel 2 etc. Nodes 1 to 3 are of MV types and measure voltage, and node 3 is type M2 and measures resistance using 2 wires. We can then instruct the software to calculate all relevant information from these measured values.

Channel 1 measures temperature at the bottom of the sample. This is done by converting the resistance of the thermistor to a temperature, converting the measured voltage to a temperature difference, and adding those two together. In Omega this is done in the graph section while plotting values. Exact values for expressions can be found in the table below.

Channel 2 measures sample voltage. Channel 3 measures temperature at the top of the sample and Channel 4 measures thermistor resistance at the junction of the thermocouple compensation wire to a normal wire.

In addition to measuring these values they need to be converted to more understandable format, and this is done in the graphs section. Create new graph for the measurement, name the X-axis to time in minutes, and the Y axis to main temperature. Add three additional Y-axes and name them delta temperature, sample voltage and Seebeck coefficient. Make sure these new axes are all loced in the Y-direction.

Add 6 new series for the graph, and name them accordingly the table below. For each serie assign the correct x- and y-axes, adjust the looks and colors of the serie so they are distinguishable from eachother.

Series	X-Axis	X-Expression	Y-Axis	Y-Expression	Notes
Top temperature	Time	$N3.TM	Main temperature	TCS($N3.MV, TT2($N4.M2))	Absolute temperature calculated from thermocouple voltage and cold junction temperature for the thermocouple at the top of the sample.
Bottom temperature	Time	$N1.TM	Main temperature	TCS($N1.MV, TT2($N4.M2))	Absolute temperature calculated from thermocouple voltage and cold junction temperature for the thermocouple at the top of the sample.
Delta T	Time	$N3.TM - $N1.TM	Delta temperature	TCS($N3.MV, TT2($N4.M2)-TCS($N1.MV, TT2($N4.M2))	Top thermocouple temperature minus bottom thermocouple temperature.
Sample voltage	Time	$N2.TM	Sample voltage	$N2.MV	Sample voltage including the lead thermovoltage. Lead F is exposed to larger temperature gradient than lead D and this voltage is part of the plotted result.
Sample voltage - Voltage F_{ab}	Time	$N2.TM	Sample voltage	$N2.MV-(TPP((TCS($N3.MV, TT2($N4.M2))+TCS($N1.MV, TT2($N4.M2)))/2)*TCS($N3.MV, TT2($N4.M2)-TCS($N1.MV, TT2($N4.M2)))	Function TPP is used to calculate the thermovoltage in the lead F which is subtracted from the sample voltage. (TCS($N3.MV, TT2($N4.M2))+TCS($N1.MV, TT2($N4.M2)))/2 is the average temperature of the sample. TPP((TCS($N3.MV, TT2($N4.M2))+TCS($N1.MV, TT2($N4.M2)))/2) is the thermopower of platinum per degree at the average temperature of the sample. (TPP((TCS($N3.MV, TT2($N4.M2))+TCS($N1.MV, TT2($N4.M2)))/2)*TCS($N3.MV, TT2($N4.M2)-TCS($N1.MV, TT2($N4.M2))) is the thermopower of platinum per degree at the average temperature of the sample, multiplied by the sample temperature gradient.
Seebeck coefficient	Time	$N2.TM	Seebeck coefficient	$N2.MV-(TPP((TCS($N3.MV, TT2($N4.M2))+TCS($N1.MV, TT2($N4.M2)))/2)*TCS($N3.MV, TT2($N4.M2)-TCS($N1.MV, TT2($N4.M2))) / (TCS($N3.MV, TT2($N4.M2))-TCS($N1.MV, TT2($N4.M2)))	Here the corrected sample voltage is divided by the sample temperature gradient and plotted against time. More logical would be to plot against sample average temperature, with X-expression (TCS($N3.MV, TT2($N4.M2))-TCS($N1.MV, TT2($N4.M2)))

Remarks:

- It is useful to plot channels 1 to 3 separately, as well as separate series indicating the delta T, and another indicating sample voltage per delta T. From all of these plotted on the screen, it is easy to see the effects of heating and cooling, and to conclude that most stable Seebeck coefficient is measured in isothermal conditions. The duration of stepping of the temperature can be used to study the heat convection, conduction and radiation before thermal equilibrium is reached.

- It is good to note that the thermopower for Platinum is negative, so it is most likely of opposite sign than the sample thermopower, and is thus adding to the total voltage read on channel 2.

- The often neglected aspect of electrode voltage F_{ab} due dT can be negated using reverse and average, but as reverse and average is already covered elsewhere in the manual we use the TPP (ThermoPower Platinum) function to calculate and remove the F_{ab}. If other material than Platinum is used for the leads F_{ab} cannot be eliminated with TPP function.

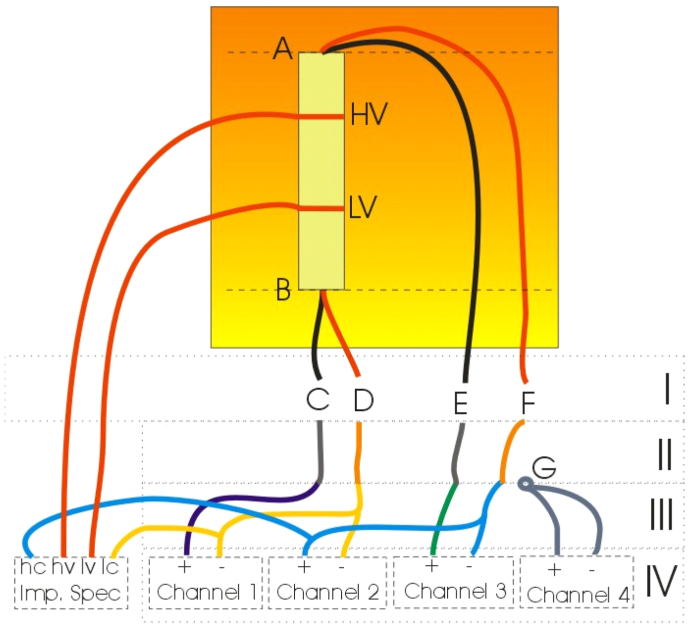
- If the user instead has direct contacts on sample for sample voltage those leads will be connected to channel 2 instead of leads D and F.

Related files

6.9 Open and Short circuit compensations

Measuring a several nodes as IS type it is possible to obtain the electrode open and short circuit characteristics per signal type and per frequency. A node can measure the sweep with the sample removed from between the electrodes or the electrodes short circuited where the sample is missing. Another node can then measure with the sample in place. As long as the amount of sweep points is identical for all sweeps the user can plot series combining the information for different sweeps.

6.10 Combined Seebeck coefficient and resistance measurement

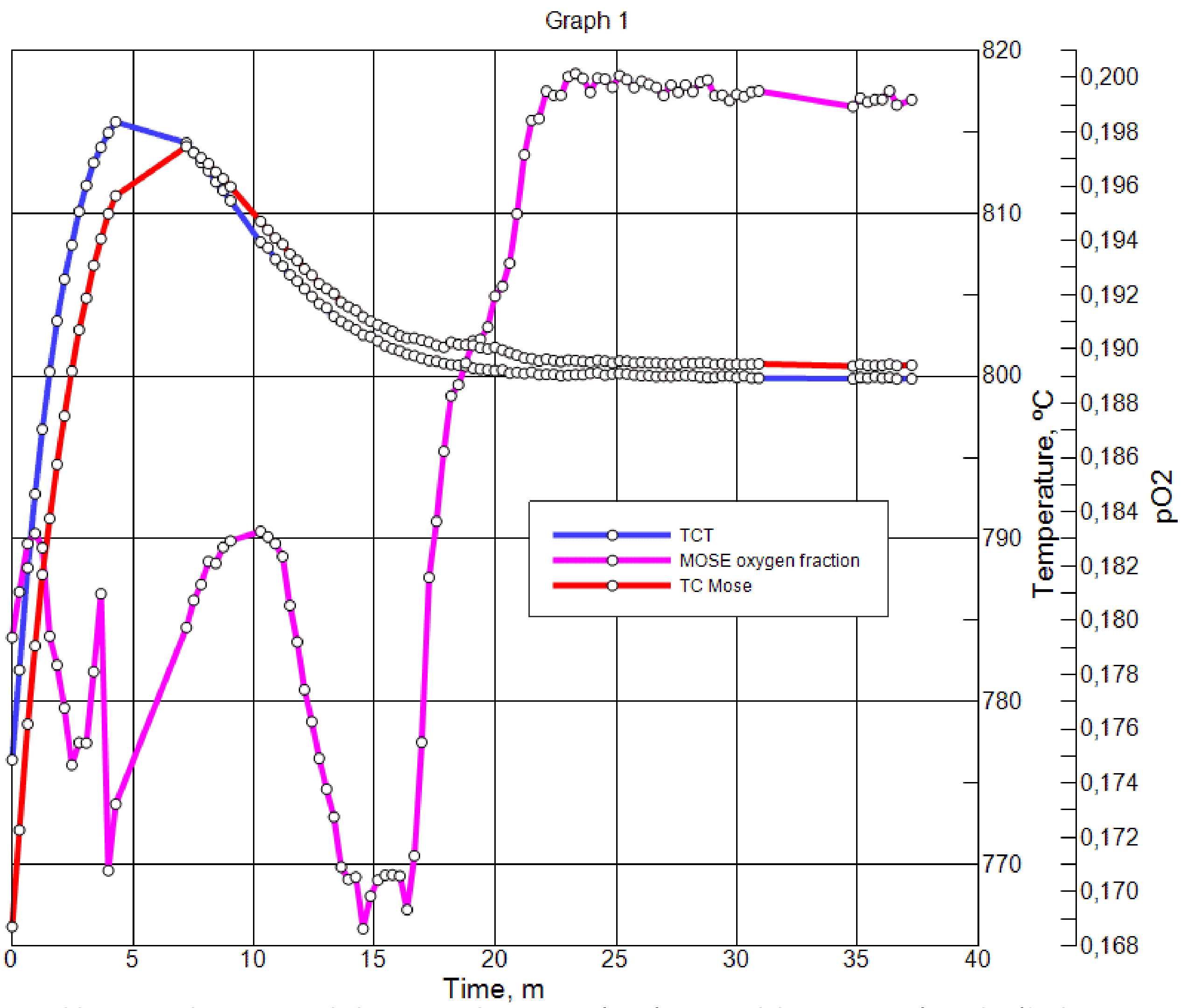


The setup is similar to the Seebeck coefficient measurement example and in addition an impedance spectrometer is added and connected according the schematics.
Besides the nodes of the Seebeck measurement, node 4 (type IC) measures the resistance using the impedance spectrometer. The high input impedance of the passive spectrometer assures the voltage measurements are not affected, while all the channels on the multimeter are open when the spectrometer measures.

Solving for resistivity or conductivity requires the distance between HV and LV, and the measurement of that distance is the single greatest source of error in the measurement.

6.11 MOSE

MOSE (Miniature oxygen sensor) can be measured with Omega. As seen in the graph it is important to let the temperature stabilize to get correct result from the MOSE: Same atmosphere was present during the temperature stabilization.



Two things must be measured, the S-type thermocouple voltage, and the sensor voltage itself. The MOSE unit utilizes only 3 wires, so these must be measured in turns. The Sensor voltage itself must be measured with voltmeter with at least 1G ohm internal resistance, otherwise the MOSE will get uncabrated.

MOSE function for expression parser is used to plot the resulting pO_2.

6.12 Measuring MOSE with Omega software

Omega software with suitable multimeter (such as Keithley 2000 with 10 channel switch) can be used to measure all required properties and to calculate the final pO_2. Instrument will measure two voltages and a resistance in rapid sequence. From the resistance the software will calculate cold junction compensation; and use the thermocouple voltage and sensor voltage to solve the Nernst equation for the pO_2 with built-in function for MOSE.

The function to calculate and plot the pO2 (fraction of pure O2 at atmospheric pressure) in Omega is

MOSE(T,E,A,B) where

T is sensor temperature in Celsius
E is sensor voltage (with odd results change the polarity)
A calibration factor Ax from supplied excel sheet (in cell E54)
B calibration factor Bx from supplied excel sheet (in cell E55)

To solve for temperature one can use TCS and TT2 functions explained further in the Omega manual. TT2 converts thermistor resistance to temperature; this is our cold junction compensation. TCS solves for S type thermocouple temperature.

TT2(10000)

Thermistor has roughly 10 000 ohms at room temperature, so the above gets evaluated to 25, or to your ambient temperature when you use the resistance measured from the thermistor. The thermistor should be located where the thermocouple ends and identical measurement wires start.

TSC(0.005, 25)

First parameter is the thermocouple voltage, second is the cold junction compensation in Celsius. In this case the above would evaluate to 590°C (565 from the thermocouple voltage and 25 from the cold junction compensation).

Final expression

End result would look something like MOSE(TSC($N1.MV, TT2($N2.M2)), $N3.MV, A, B) where E is replaced with sensor voltage e.g. $N3.MV, A and B are the calibration factors. Node 1 is sensor thermocouple voltage, Node 2 is resistance of the thermistor located at the end of the thermocouple compensation cables, used as cold junction compensation, and Node 3 is the sensor voltage.

7 Instruments

7.1 Solartron 1287

In general 1287 can be used for potentiostatic or -dynamic, galvanostatic or -dynamic, open circuit voltage measurement measurements or in conjunction with FRA to enhance the bias added to the signal from FRA.

All the above functionality is possible with Omega, but operating the instrument requires good knowledge of the methods used, the Omega software and the instrument itself. This article acts as a guide with these high threshold subjects.

7.1.1 Prerequisites

Use 1287 alone or with 12xx series FRA.

First task is to add the 1287 to the GPIB network. Since the 1287 is old instrument it has some limitations; it has no way of identifying itself on the network. This means it can not be autosearched by Omega. The user must know on what COM port the GPIB network is located, and what address the instrument has. The user must manually add the instrument to the devices configuration file. This old instrument additionally needs a validation trick.

Once the devices setup is done and the instrument resets after the validation (the initialization procedure) proceed to the next phase.

7.1.2 Creating the measurement

Add a new measurement and name it appropriately. Potentio/galvano static/dynamic measurements all follow this setup. For open circuit voltage measurement leave the first node off. In this example we will create potentiodynamic measurement.

This measurement needs two nodes, AU type node for controlling the voltage or current and for installing the instrument in correct mode, and P1 type node for retrieving the results. Add two nodes and make the first AU and the second P1 type.

7.1.3 First node, AU-type

This assumes all the 'normal' properties such as 'node is active' are set by the end user.

The first node is to set and keep the instrument in correct mode, and to send command to adjust the potential.

On the AU node in the “GPIB before measurement” field enter the following:

```
// Lines starting with // are comments.
// All commands explained in manual page 4.17 onwards
BY1  // Standby (0|1) Full|Half.
IL6  // Current limit (0|1|2|3|4|5|6) 2uA|20uA|200uA|2mA|20mA|200mA|2A
OL1  // Current limit action (0|1|2) Cutout|Limit|No limit
PX3  // First read parameter: Vre1-Vre2 (P1 field 1, field name Voltage)
PY5  // Second read parameter: I (P1 field 2, field name Current)
PO0  // Mode (0|1) Potentiostat|Galvanostat
PW1  // Polarization on
```

Click apply. This code puts and keeps the instrument in potentiostatic mode and define few other key parameters.

In the “Action node setting” select “GPIB instrument” and “PotGal 1287…” and click apply.

The AG1 expression will determine the amplitude of the potential or current that we will apply. The expression can be just a static number for static measurements, or it can be an elaborate expression generating any type of waveform for the amplitude. The “Max” field is a safety feature and will define the minimum(negative maximum) and maximum values that the expression can yield. Enter for AG1 expression

```
IF(MOD($I/40,2)=0, MOD($I,40)/10, ABS(40-MOD($I,40))/10)+2
```

Each instrument has it's own format for numbers and commands. Just because expression evaluates to a number does not mean that number is represented in correct format so that the instrument would understand it. Instrument manuals in chapters for remote control often define what the format for number must be. For 1287 and 1260 the number format is +n.nnnnE±xx

So the result from the expression must be formatted before sending it to the instrument and for this reason the Omega utilizes some rules for number formatting. The formatting can be tested in the tools section / number format tool. If you decide to try that out while reading this, remember to click apply before leaving the node properties manager or you will lose all the changes you made. For 1287, the expression to use, and to enter into “format rule for .AG1 is “0.0000E+00”

Finally we need to send the new amplitude to the instrument. Enter into GPIB code field the following:

```
// 0.0000E+00 is the format F for 1287, page 4.17
// AG1 = Linear zigzag between 2 and 6
PV$AG1
//#SLEEP 2000  //This would add 2 second additional delay after applying the new voltage and before
measuring the current.
```

All the // lines are comments and will not be sent to the instrument. PV is a code for the instrument to apply voltage. $AG1 is the result of AG1 expression formatted with the format rule.

7.1.4 Second node, P1-type

Mark active, Select type P1 and click apply. Select 1287 as instrument and click apply.

7.1.5 Conclusion

Measurement is ready to start once it is marked active.

For potentiostatic measurement just use the desired voltage value as numbers in the expression field for AG1 instead of expression.

For galvanostatic or -dynamic measurements the following lines needs to be changed:

Parameter	Location	Old value
Operation	Node 1 → GPIB commands before	PO0 //Potentiostat
Amplitude	Node 1 → Action node settings → GPIB code	PV$AG1
Wave expression	Node 1 → Action node settings → AG1	IF(MOD($I/40,2)=0, MOD($I,40)/10, ABS(40
AG1 safety limit	Node 1 → AG1 → max	15

Parameter	Location	Old value
Node name	Node 1 → Caption	A10 Set voltage

Related files

7.2 Gen'Air

Setnag Gen'Air can be read with Omega for Oxygen partial pressure. It is also possible to connect a programmable power supply to both Omega and Gen'Air, and control the applied pump voltage with Omega and the power supple.

Gen'Air does not support autosearch, so the instrument must be added manually to the devices configuration file.

```
COM1
         ASCII
                  GENAIR GENAIR COM1 ADDR:0 EOS:0 ****
```

Maximum measurement frequency is 5 seconds, or 0.085 Hz

7.3 HP Hewlett Packard 3458A

This old reliable instrument has it's autotrigger on by default and this prevents autosearch from working. Select "trigger: hold" using the front panel before trying to autosearch with Omega.

Instruments entry in devices configuration file is such as:

```
COM44
         Prologix USB-GPIB device
                  Multimeter HP3458A COM44 ADDR:1 EOS:0 GPIB IDN?:HP3458A
```

7.4 Instruments without auto search

Some older instruments do not support all standard GPIB commands and cannot thus be found or validated by auto search. Setting up and working such instruments require little extra effort. Other instruments just have too large address space to be scanned automatically, Imagine searching all COM ports for All types instruments and instrument networks, using all possible protocols, for all possible instrument responses, from all possible instrument addresses. It would take too long time. So the subcategories provide information on other device types that need to be added manually.

7.5 HP 4192 A

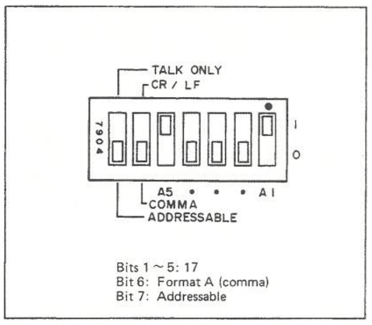


Figure 3-35. HP-IB Control Switch

7.6 HP4192A

Make sure the A6 and A7 switches are position 0, at ‘Comma’ and at ‘Addressable’.

The Instrument address is determined with switches from A1 to A5. A1 represents 0 or 1, A2 is 0 or 2, A3 is 0 or 4, A4 is 0 or 8 and A5 is 0 or 16. With the 0 value being when the switch is down. In the example, the device address is 17 (1+16).
The ID for the device file is HP4192A.

7.7 Vögtlin Smart series mass flow controllers

Vögtlin Smart series mass flow controllers have the mass sensors inside the gas flow (mems technology) allowing very accurate and quick feedback and control of the gas flow. They are the optimal choice for most research purposes. Vögtlin also offers more traditional capillary type mass flow controllers. NORECS can sell both type to any caustomer as long as the sale is part of a larger measurement system sale. Otherwise turn to your local Vögtlin distributor.

Find out the modbuss addresses that your MFCs use, either from the instrument printed label or ask for the information from NORECS.

The .odc file example entry for Vögtlin Smart series controller is

```
COM38
        MassFlowController GSC COM38 ADDR:2
```

Same keywords and format can also be used for Mass flow meter, but they will not control the flow, only measure, and sending flow control commands to them is untested and may cause problems.

There is also a toll on the tools tab to help with the mass flow controllers. To change advanced properties in the MFCs, use the Vögtlin Get Red-Y software. Please note that once done using it, the Get Red-Y must be killed using task manager, at least on windows 10 and for versions 5.6 and earlier. Otherwise it will leave a process running and will keep the opened COM ports open so no other process can use the ports.

7.8 Bronkhorst mass flow controllers and Omega

Operating Bronkhorst mass flow meters or controllers (MFC from now on in this page) differs somewhat from operating other instruments with Omega. The network and network communications protocol for the MFC's is set up according user preferences and manufacturer specifications. The network is connected to a computer, again, according to the manufacturer specifications. a DDE server is ran on the computer that takes care of low-level communications between all the instruments on the network and it also provides interface for Omega to control and read values from the instrument.

Once the DDE server is running and can communicate with the instruments, Omega can be set to access the data. Autosearch in Omega is not supported at this time. Instruments are set up by manually editing the device configurations file.

```
DDE
        FLOWDDE
                MassFlowController ELFLOW FLOWDDE ADDR:1
                MassFlowController ELFLOW FLOWDDE ADDR:2
                MassFlowController ELFLOW FLOWDDE ADDR:3
```

In where the only thing to edit is the number of instruments (by adding more lines) and their address/channel (must be unique number and match with DDE setup).

7.9 Operation in Omega

The instrument flow is read as units that the instrument is configured to measure, but the new setpoint is sent as a fraction of 32000 where 0 is zero capacity and 32000 is 100% is capacity. For example instrument rated for 100 mln/min of Air, would aim for 50 mln/min of air if the setpoint sent to the instrument was 16000.

7.10 Devices setup file

```
COM33
	Prologix USB-GPIB device
		Imp.Spectrometer HP4192A COM33 ADDR:1 EOS:0 ****
```

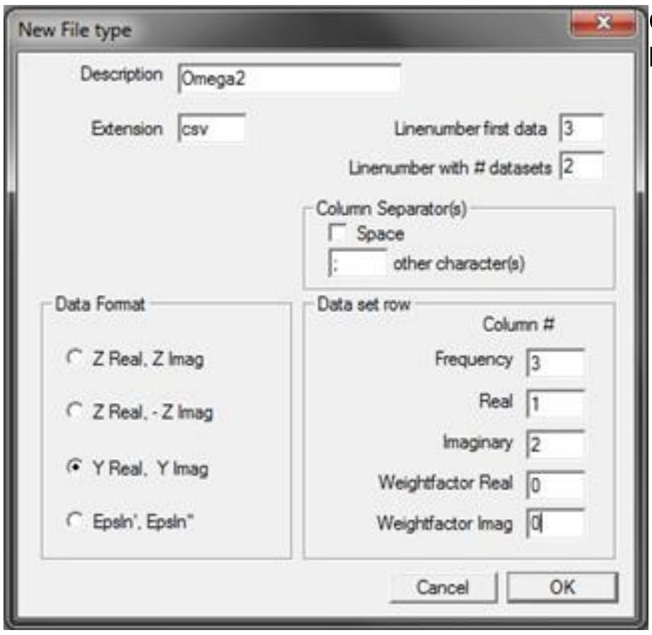
The device setup file has all the information about the instruments the software is using. The file can be named anything and ends with .odc type. It is just a text file that can be created / edited with notepad. To use for example HP4192A, lines according the picture must be added and edited.
First line, ‘COM33’ is the Serial port where the Prologix USB-GPIB device is connected. The number must match the port on your system. Easiest way to determine this is to go to windows control panel and device manager, and open serial ports from the list. Unplug and re-plug the Prologix USB-GPIB and see which port disappears and reappears. This is the port name to use in the device setup file.
Second line starts with tab (tabulator space) followed by the mentioned text.
Third line defines the instrument type and details. Starting with two tab’s followed by instrument type, instrument ID, again the com port name, instrument address, end of string identifier and text ‘GPIB’ all separated by single space. Edit this line to match your instrument details. Replacing the text GPIB with **** tells the validation process to automatically assume that this instrument is correct.

It is a good idea to scan the setup first, then save the setup to disk and then edit the unsupported instrument into the file. This way the possible other instruments already exist in the file.
For the reference, here is another file:

```
COM1
	Eurotherm 2216 COM1 ADDR:1
COM33
	Prologix USB-GPIB device
		Imp.Spectrometer Hioki_3522-50 COM33 ADDR:2 EOS:0 GPIB
		Multimeter K2000 COM33 ADDR:4 EOS:0 GPIB
```

8 Appendix

8.1 ECQWin file settings



New File type

Description: Omega2
Extension: csv
Linenumber first data: 3
Linenumber with # datasets: 2

Column Separator(s)
- [ ] Space
- ; other character(s)

Data Format
- Z Real, Z Imag
- Z Real, - Z Imag
- [✓ Y Real, Y Imag]
- Epsln', Epsln"

Data set row

	Column #
Frequency	3
Real	1
Imaginary	2
Weightfactor Real	0
Weightfactor Imag	0

Configure a new file type according the details in the picture to open saved sweep data.

8.2 GPIB network considerations

GPIB network is the GPIB devices connected to each other with GPIB cables.

A GPIB network has 31 addresses, from 0 to 31. Each device must have individual address. It is preferable to use only every second address, as some older devices have hidden features on a sub-address +1 to the main address, and these may cause conflicts if main addresses are consecutive numbers.

In the obscure case that 15 instruments are not enough, one can use every address in the network to get up to 31 instruments. It is actually possible to add as many USB-GPIB converters as necessary while each of them is able to control up to 31 separate GPIB instruments.

8.3 Furnace connections

Any furnace with Eurotherm series 2200, 2400 or 3200 controller that has MODBUS communications protocol can be connected to the Omega software.
The MODBUS communications protocol is either a selectable option in the menu of the controller, or a physical card inside the controller. It is possible to replace this card in case it is of some other type such as BiSync protocol.
In addition to the controller model and protocol, the wiring has significance. Most furnaces have RS232 straight or cross over cabling. These can be connected to the software with the USB-RS232 adapter.
One USB-RS232 adapter, one null modem and one mini gender changer are included with the software. With these it is possible to connect to most furnaces. For additional pieces, USB to RS-422 or RS-485 adapters and Eurotherm communication chips contact NorECs AS.
In case having troubles with furnace connections, check the Eurotherm controller data transmit settings such as baud rate, data bits, parity and stop bit settings from Eurotherm controller menus, to match the ones of the virtual USB serial port representing the USB-RS232 converter. These will appear on the control panel/device manager of the Windows operating system.

8.3.1 Connecting to a single furnace

- Unpack the 'USB to serial port cable' (later referred as USB-RS232 cable) and connect the USB end to your computer.
- Locate the serial port on your furnace.
- On Elite furnaces, plug a 'Null modem adapter' to the furnace. Plug the USB-RS232 cable to the furnace.
- On all other furnaces, try plugging the USB-RS232 cable directly to the furnace. If it doesn't fit, use the 'Mini gender changer' to succeed. Sometimes both the furnace and the USB-RS232 cable have small nuts to screw the counterpart onto with small screws. When both cable and the device have these nuts, they do not fit together! For your convenience we have included the mini-gender changer part as it allows to overcome this physical obstruction.
- The Modbus address of the furnace needs to be 1 or 2 for the device autoscan to find the furnace.

USB-RS232 adapter

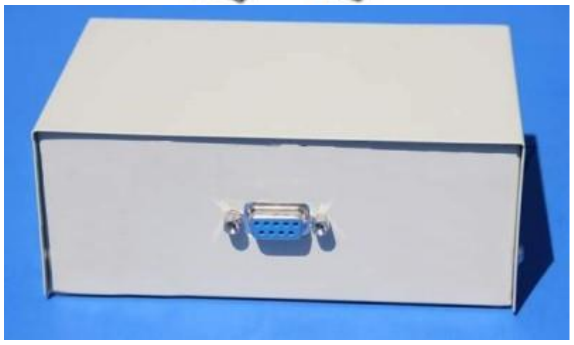
Null modem adapter

Mini gender changer

Furnace back panel serial port

USB-RS232 adapter

Null modem adapter



8.3.2 Connecting to more than one furnace

Easiest way to connect multiple furnaces to Omega software is to have multiple USB to RS-232 converters and connect each of them to the computer and the other end to a separate furnace according the instructions for connecting ‘to a single furnace’.
However, it is possible to have network of furnaces connected to each other, according the RS-422 or RS-485 protocol. To connect to this network one would need USB-RS422 or USB-RS485 converter.
When using a network of furnaces, please note that the automatic scan starts from address 1, and only scans one extra address after a furnace was found (otherwise it needs to scan the whole address space of 256 addresses which would take a lot of time). In other words, the Furnace addresses needs to be such as 1, 2, 3, 4 and 5. If another furnace is in the network with address 87, it will not be found as last address to be scanned in this case would be 6.
However, instead of networking the furnaces, much easier option is to add separate USB-RS232 converter for each furnace.

8.4 Supported instruments

Role	Device
Impedance spectrometer	Solartron 1260 Novocontrol Alpha Analyzer with ZG4 (Treats all as AT of B series) Hioki 3522-50 HP 4192A (With no auto search*)
Multimeter	Keithley 2000 series (With or without cards) Agilent 34970A (With or without cards) Agilent 34420A (Micro-Ohm meter)
Furnace	Eurotherm 2200, 2400 and 3200 series with MODBUS
External switch	Examples for Pickering
Power supply	Keithley 2200 series, Keithley 2400 series.
Mass flow meter, Mass flow controller	Vogtlin Smart, Bronkhorst El-flow
Potentiostat	Solartron 1287

9 Troubleshooting

9.1 Manual installation of GPIB converter

In case the Prologix USB-GPIB converter does not get recognized as Virtual COM port, installing the device can be done manually. The Omega program contains the required installation scrip. If you have never started the Omega program, run it once and close it. Look for the folder 'DLL' under the location of the Omega, and run the 'FTDI.EXE' in that folder. It will install and configure the Prologix USB-GPIB. In case you are prompted to select the type, use 'VCP drivers' or 'Virtual COM port' drivers.

9.2 USB dongle did not install automatically

9.3 Frequently asked questions

9.4 Examples

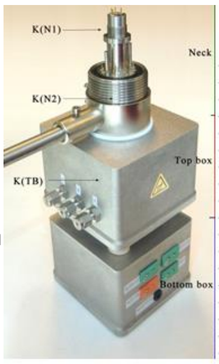
Most examples consist of disc or bar samples at high temperatures. Typical uses for the software are high temperature measurements with ProboStat. However there is no reason not to use the software for other purposes. For more detailed information about setting up the sample and the electrodes, refer to ProboStat manual.

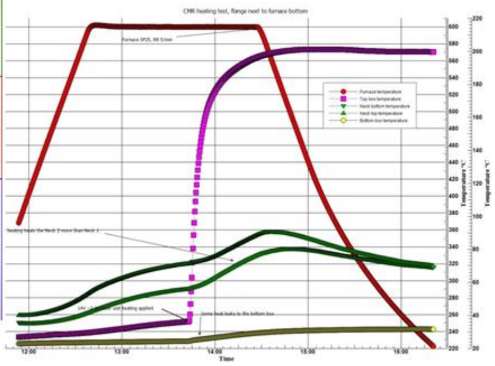
9.5 Sample files

Some sample files with measured data are created automatically to a folder Measurements/Examples/

9.6 CMR heating test

This example measurement is a real life study of ProboStat CMR (modified version of ProboStat for Ceramic Membrane Reactors) base unit heat behavior. In addition to the hot zone ceramics, the base unit itself is heated to avoid coking inside the gas lines. The measurement studied heat behavior of different regions of the base unit. The base unit was placed 1cm below bottom of a furnace that was heated to 600°C (with the reactor ceramics in the furnace), and later on the base unit heating was turned on. The measurement utilizes Keithley 2000 multimeter with 10 channel scanner card.





Node 1 reads thermistor resistance inside Bottom box.
Node 2 reads the temperature of the furnace.
Node 3 reads K-type thermocouple voltage at K(N1).
Node 4 reads K-type thermocouple voltage at K(N2).
Node 5 reads K-type thermocouple voltage at K(TB).

Corresponding series plot the temperature read or calculated from the data against time, with some markers to identify the events and phenomena's.

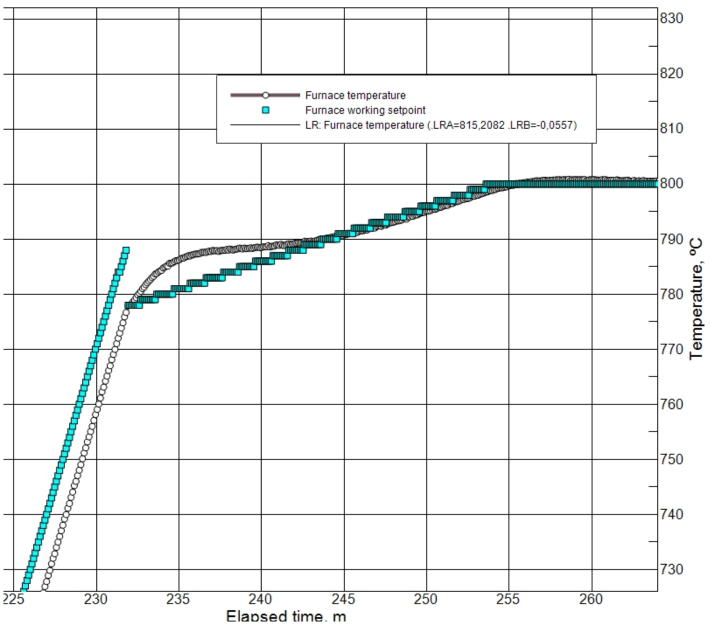
9.7 Temperature comparisons

This example is about reading the furnace temperature in 3 different ways. S- and K-type thermocouples and A Thermistor were bundled together and placed into a furnace. The S-type Thermocouple was used to

control the furnace. Another thermistor was placed in the junction where thermocouple wires change to wires of identical material.
Node 1 reads the resistance of a thermistor in the furnace.
Node 2 reads the resistance of a thermistor at cold junction.
Node 3 reads the voltage of K-type Thermocouple.
Node 5 reads the temperature announced by the furnace, measured by S-thermocouple.
As the thermistor resistance to temperature function is only meant to convert room temperatures, the function is valid only for temperatures below 55°C, this resistance could not be converted to temperature, but it aligns very well with the K and S type thermocouple readings.

9.8 Dynamic ramp rate for furnace

Avoid furnace overshoot (specially if controlling the furnace with thermocouple inside the ProboStat) by applying dynamic ramp rates. Here ramp rate was cut down from 10 degrees per minute to 1 degree per minute when the measured temperature was within 25ºC of the target. (Resulting in maximum overshoot of 0.8ºC)



For the ET (Furnace read) node type

.ET is the temperature received from furnace.

.WSP is the working setpoint received from the furnace.

9.9 Automatic sweeps

9.9.1 Introduction

In high temperature electrochemistry a typical goal is to obtain a number of impedance sweeps performed on the sample at multitude of temperatures and or atmospheres. This process may take considerable amount of time and effort as it is typically controlled manually. In general the instruments (furnace and impedance spectrometer) do not communicate with each other, and the user must initiate the measurements when conditions are deemed suitable.

With Omega software the impedance sweeps can be automated and in this application note the process is briefly described.

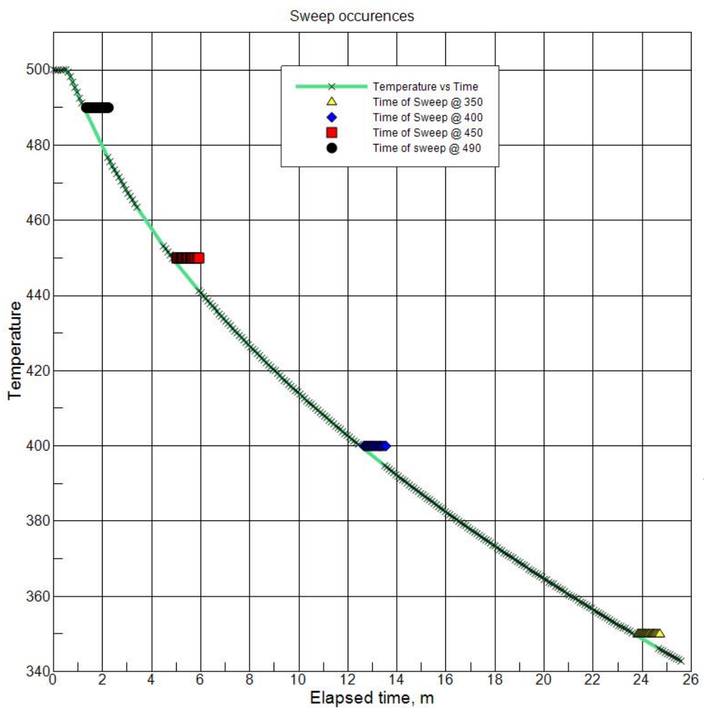
9.9.2 Physical setup

This setup included Omega software running on a computer, ProboStat sample holder, impedance spectrometer and a furnace. The impedance spectrometer was connected to the sample holder, and the sample holder was placed inside the furnace. The computer was connected to both the furnace and the impedance spectrometer and the software controlled both devices according user specified instructions. The control thermocouple for the furnace is located next to the sample, so temperature read from the furnace is the actual sample temperature. This type of furnace with possibility for default thermocouple override simplifies the process.

The details of the setup and the connections are specified in the Omega and ProboStat manuals.

9.9.3 Software functionality

In general, the Omega software reads the furnace temperature periodically. A number of pre-defined sweeps wait for right conditions to occur. The conditions can simply be such as "The furnace temperature is higher/lower than x" (case 1), or as well more advanced, such as "599 ºC < Furnace temperature < 601 ºC AND Furnace temperature rate of change < +/- 0.1 ºC / min (for the last 30 minutes)" (case 2). On the latter case the sweep would only start after the temperature has properly stabilized, while the earlier would start the sweep even if the furnace temperature kept changing as in this picture. Both cases are covered below.



Case 1: Sweeps while temperature changes

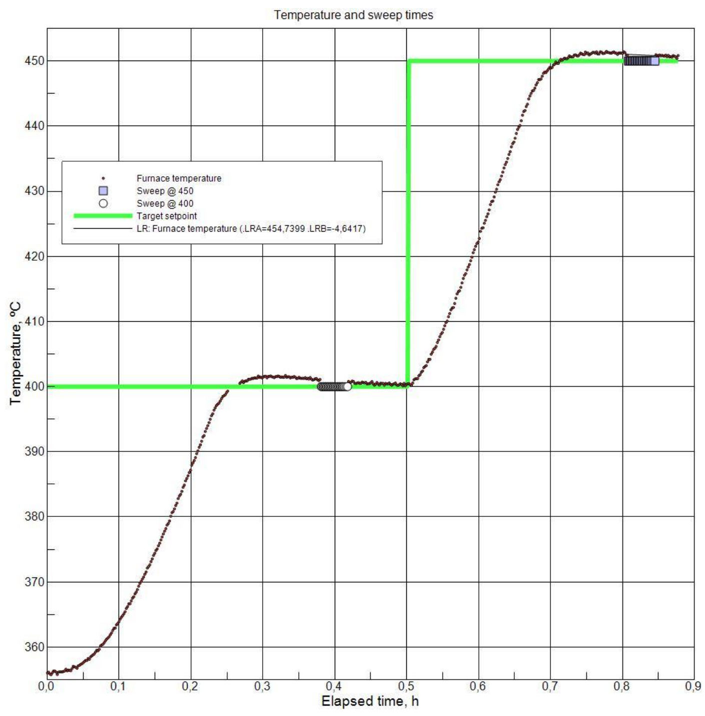
Fastest way to gather a number of impedance sweeps of the sample is to do so at constantly rising or declining temperature. In case of a sweep with 50 points, execution of such sweep takes about 1.5 minutes (if frequencies are above 1 Hz) and the change of temperature over the sample during this time may in some cases be considered negligible. If that is not the case, proceed to case 2.

Picture 1

Sweeps taken on a fast declining temperature where the cooling rate of a furnace is not restricted in any way. Note that this is not recommended behaviour and will put lot of thermal stress on ceramics across the temperature gradient, and is presented here just as a reference to demonstrate the (worst possible) change in the sample temperature during taking of a sweep.

The short horizontal lines represent the time it took for the sweeps to perform and the sloped line represents the temperature measured at the sample. It is easy to see the temperature keeps changing during the sweeps (each of which consisted of 50 points of >1 Hz frequencies and lasted roughly 90 seconds each).

The heating or cooling rate of a furnace can be set to any desired value, for example such as 5º C/min. In such conditions during a sweep the sample temperature would change approximately 7.5 ºC/min. As mentioned earlier this may or may not be acceptable.



Case 2: Sweeps when temperature is stable

In case temperature needs to be stable when taking sweeps, the conditions determining when a sweep should start are slightly more complicated. Studying the stability of a measured property requires calculation of rate of change, and this can be achieved by utilizing the coefficient tool of the temperature series. In the Graph section add a serie for the temperature and add a coefficient tool (tick the linear regression checkbox) for that serie. The tool automatically calculates (with linear least squares) coefficient of the x last points of the series. This real time information can be used to determine when sweep should start.

Picture 2

Furnace temperature was controlled with the software in steps of 50 ºC per every 30 minutes. The light green line represents the setpoint (target temperature) information sent to the furnace with ramp rate 5º C/min (desired rate of change in temperature). Series with small dots represent the measured furnace temperature. The big circles and squares represent the times when a sweeps were being performed.

In addition to specific furnace temperature, the angle of last x point is used to determine when the furnace temperature rate of change is small enough. The unit of the rate of change is Y axis unit / X axis unit, and in this case ºC were plotted against hours.

The starting condition for 'sweep at 400' was (399 ºC < Furnace temperature < 401 ºC) AND (rate of change < +/- 15 ºC/hour). In detail this was written:

($N1.ET>399)&($N1.ET<401)&($S1.LRB<15)&($S1.LRB>-15)

where the & symbol equals the AND mathematical operator, the $N1.ET represents the measured temperature of the furnace, and $S1.LRB is the slope of the linear least squares coefficient of the series 'Furnace temperature' plot. The unit of the plot is degrees per hour, so LRB can be used as it is with no conversion to minutes etc.

For the Linear regression tool, the LRA is the offset in units of Y axis and LRB is slope in units of Y axis per X axis. (Units are relevant since the X axis can be minutes, hours, days, dates, or maybe even not time related at all.)

After the Sweep is done, we aim to go to a new temperature, and for simplicity, this is here done time dependent. It can also be done just studying if the sweep has finished, but time dependency is easier for this example.

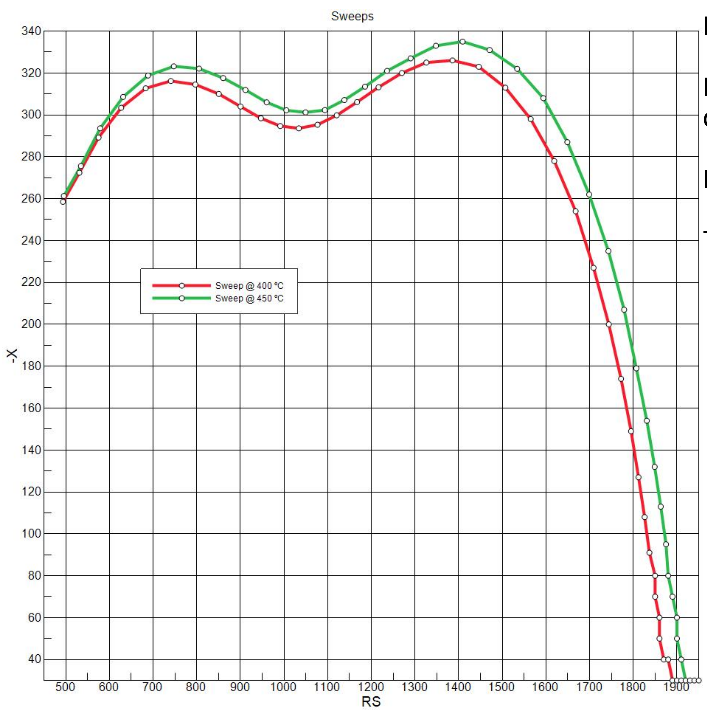
New setpoint gets sent to furnace each time the expression TRUNC(EHOUR($N1.TI)*2)*50+400 evaluates to a new value.

EHOUR($N1.TI) returns elapsed time since the beginning of the measurement, and TRUNC drops the fractional part of a number. So every 60 minutes this returns the next integer (since we multiply by 2 we twice as often, every 30 minutes). This is multiplied by 50, and 400 added, thus getting stepped setpoints from 400 onwards with step of 50, applied every 30th minute. For this the user must know roughly how long the ramp will take and how long it will take for the temperature to stabilize.

Now, an important notice here is that the furnace temperature is not the same as the sample temperature. Moreover, stable sample temperature is not the same thing as stable samnple conditions. Depending on the sample and used atmosphere, the sample may need long time to reach chemical equilibrium. Better than testing for stable temperature would be to test for stable conductivity. And when the conductivity has proven stable, the sweeps are made and then sent to new temperature. This, obviously cannot be made with time dependent furnace control, instead before going to new temperature, the automation must evaluate if the sweep has been finished or not.

Setup guide for Automatic sweeps as function of conductivity

Please note that the actual sweep data is presented on the next image.



Representing the gathered data

Data from each performed sweep can be plotted on separate or shared graphs.

Picture 3

The sweep data itself can be plotted separately.

9.10 Automatic sweeps 2

9.10.1 Introduction

This page is in-detail explanation of how to create a set up for automated sweeps, and also a continuation of page http://www.norecs.com/index.php?page=Automated+sweeps

This page may seem overly complicated, but once this setup is created it is easy to reuse and alter it for other purposes. Links to images of the procedures can be found throughout this page.

9.10.2 Measurement plan

The aim is to have automatic impedance sweeps of the sample at 3 temperatures and to have the sweeps triggered only after the resistivity has stabilized.

This page is divided into steps and the example files are downloadable from http://www.norecs.com/index.php?page=Software -> Omega -> Measurement files -> Automated sweeps

These measurement files can be made to fit any device configuration as long as a furnace and an impedance spectrometer are available. Please read the whole guide through first.

The sample is standard ProboStat system test sample; YSZ with painted electrodes on it. The sample is 86% dense, round electrodes have diameter of 10.7mm and the sample is 1.2mm thick. The sample becomes conductive only with Oxygen ions so the measurements start from elevated temperature.

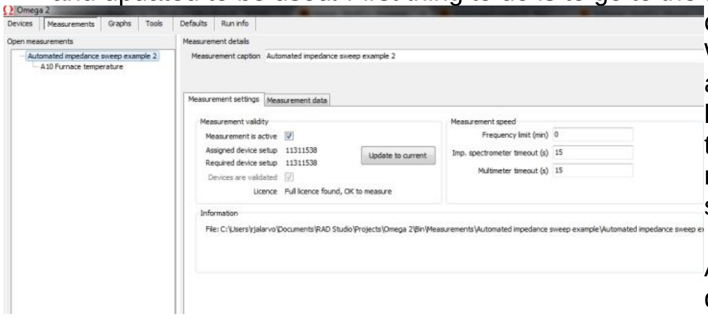
9.10.3 Device configuration

9.10.4 Hioki and Furnace

These measurements were made with the above instrument setup. None of these files will automatically work on a setup that is different from the one used. Naturally each user will have different ports and addresses assigned to their instruments.

This example is provided for the users to make their own measurement by following these steps, but the user may also want to use or open these save files in their own setup.

To make these files work on other device configurations, the user need to have corresponding setup: a furnace and impedance spectrometer. On a validated setup the measurement file can then be opened and updated to be used. First thing to do is to go to the measurement properties and click the 'update to current' device setup as seen in this picture.

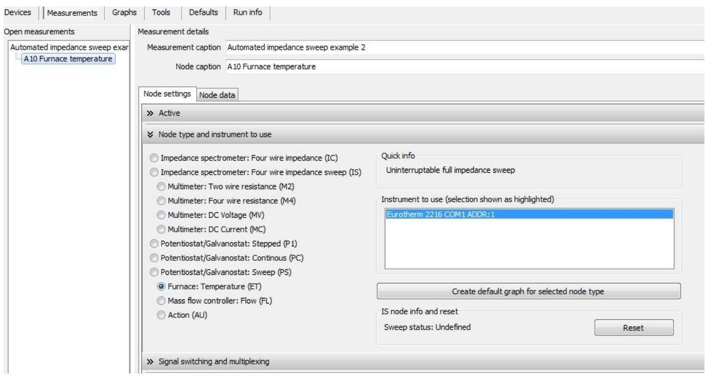


Measurement caption: Automated impedance sweep example 2
Measurement is active: ✓
Assigned device setup: 11311538
Required device setup: 11311538
Devices are validated: ✓
Licence: Full licence found, OK to measure
Frequency limit (min): 0
Imp. spectrometer timeout (s): 15
Multimeter timeout (s): 15
File: C:\Users\rjalarvo\Documents\RAD Studio\Projects\Omega 2\Bin\Measurements\Automated impedance sweep example\Automated impedance sweep ex

What this button does is forces the program to accept the fact that the measurement was made by different setup of instruments, but regardles of that still to use the measurement. This can naturally lead to lot of problems if the setup is not similar enough and if the user is not careful.

After forcing the measurement to use the new device setup, each of the nodes in that measurement must be checked and updated. The nodes were set to use an instrument that does not exist anymore or can not be found at the same port and address. The user must select a new corresponding instrument to use and click apply. Instruments are selected on the 'Node type and instrument to use', for action node (AU) on the 'Action node settings' tab and if external switching was used (not in this example) then also on the 'External switching tab'.

Select the desired instrument, make sure it's highlighted and click apply.



Open measurements
- Automated impedance sweep exar
  - A10 Furnace temperature

Measurement details
Measurement caption: Automated impedance sweep example 2
Node caption: A10 Furnace temperature

Node type and instrument to use
- Impedance spectrometer: Four wire impedance (IC)
- Impedance spectrometer: Four wire impedance sweep (IS)
- Multimeter: Two wire resistance (M2)
- Multimeter: Four wire resistance (M4)
- Multimeter: DC Voltage (MV)
- Multimeter: DC Current (MC)
- Potentiostat/Galvanostat: Stepped (P1)
- Potentiostat/Galvanostat: Continous (PC)
- Potentiostat/Galvanostat: Sweep (PS)
- [✓ Furnace: Temperature (ET)]
- Mass flow controller: Flow (FL)
- Action (AU)

Quick info: Uninterruptable full impedance sweep
Instrument to use (selection shown as highlighted): Eurotherm 2216 COM1 ADDR:1
IS node info and reset
Sweep status: Undefined

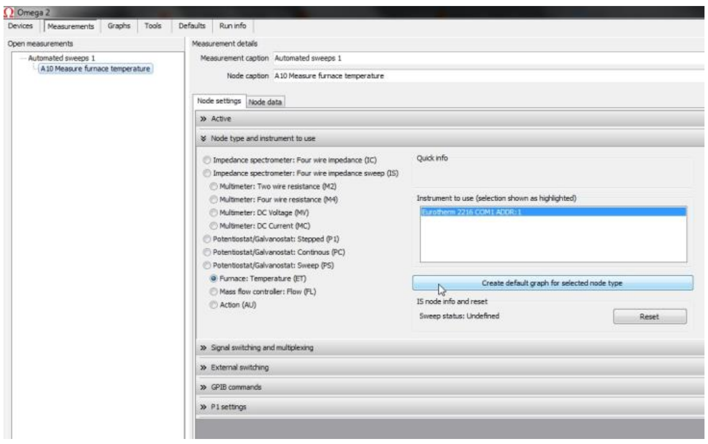
9.10.5 Step 1

Create new measurement, rename it 'Automated sweeps' and click Apply. Mark the measurement active ans click apply.

Add a node, select node type ET and click apply. Select the furnace to use, rename the node to 'A10 Measure furnace temperature' , mark the active and click Apply. Note that the nodes are sorted by name and executed in that order. Having a prefix such as A10 helps to sort things. (Names starting with number are not possible).

9.10.6 Step 2

Add a default graph for the ET node type.



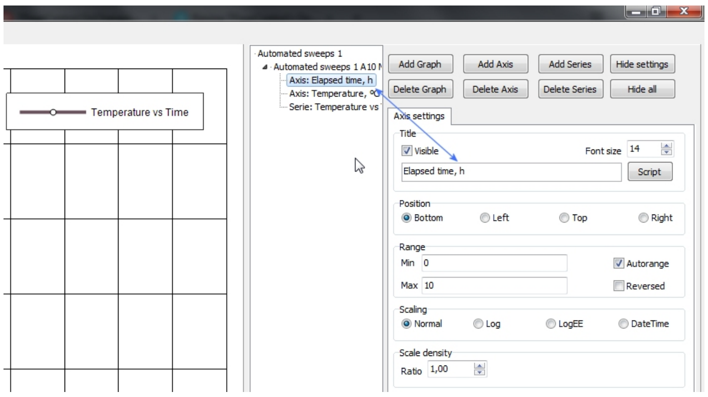
Open measurements
- Automated sweeps 1
  - A10 Measure furnace temperature

Measurement details
Measurement caption: Automated sweeps 1
Node caption: A10 Measure furnace temperature

Node type and instrument to use
- Impedance spectrometer: Four wire impedance (IC)
- Impedance spectrometer: Four wire impedance sweep (IS)
- Multimeter: Two wire resistance (M2)
- Multimeter: Four wire resistance (M4)
- Multimeter: DC Voltage (MV)
- Multimeter: DC Current (MC)
- Potentiostat/Galvanostat: Stepped (P1)
- Potentiostat/Galvanostat: Continous (PC)
- Potentiostat/Galvanostat: Sweep (PS)
- [✓ Furnace: Temperature (ET)]
- Mass flow controller: Flow (FL)
- Action (AU)

Quick info
Instrument to use (selection shown as highlighted): Eurotherm 2216 COM1 ADDR:1
IS node info and reset
Sweep status: Undefined

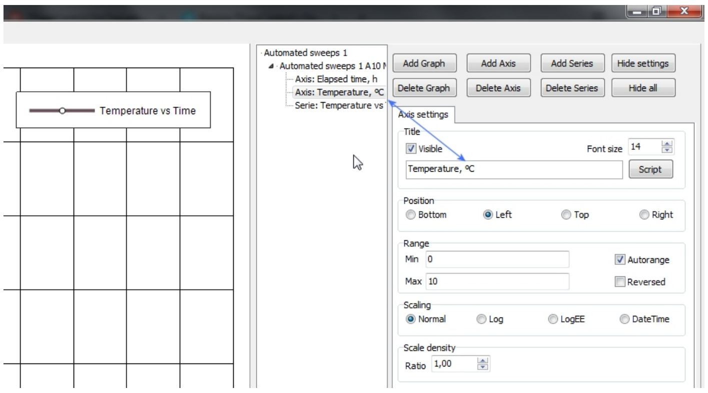
Rename the time axis to Time, h instead of Time, m



Automated sweeps 1
Automated sweeps 1 A10 N
Axis: Elapsed time, h
Axis: Temperature, °C
Serie: Temperature vs

Title: [✓ Visible] Font size 14
Elapsed time, h
Position: [✓ Bottom] Left Top Right
Range: Min 0, Max 10 [✓ Autorange] Reversed
Scaling: [✓ Normal] Log LogEE DateTime
Scale density: Ratio 1,00

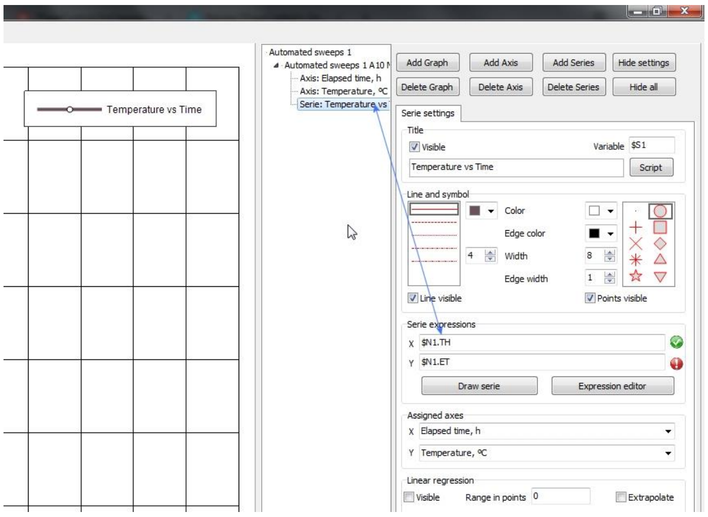
Rename the temperature axis to Temperature, °C



Automated sweeps 1
Automated sweeps 1 A10 N
Axis: Elapsed time, h
Axis: Temperature, °C
Serie: Temperature vs

Title: [✓ Visible] Font size 14
Temperature, °C
Position: Bottom [✓ Left] Top Right
Range: Min 0, Max 10 [✓ Autorange] Reversed
Scaling: [✓ Normal] Log LogEE DateTime
Scale density: Ratio 1,00

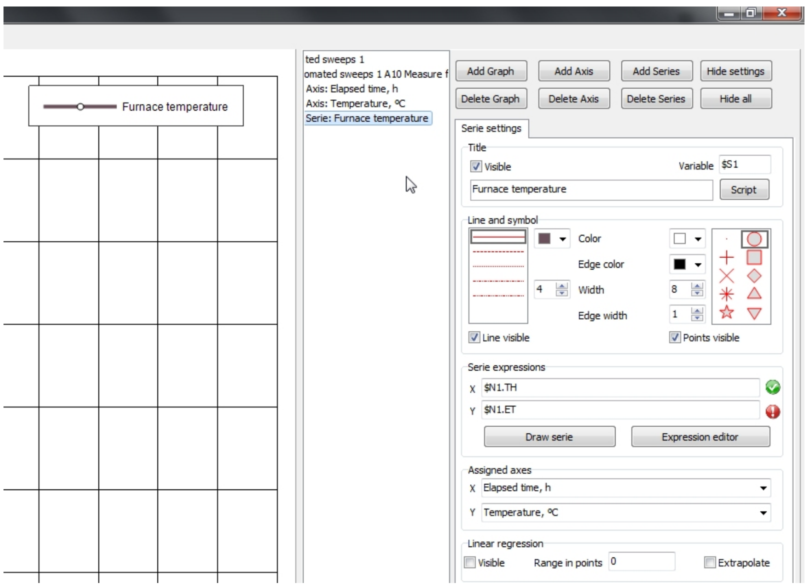
For the Temperature serie change the X expression to $N1.TH instead of the default $N1.TM (H for hours, M for minutes)



Automated sweeps 1
Automated sweeps 1 A10 N
Axis: Elapsed time, h
Axis: Temperature, °C
Serie: Temperature vs

Title: [✓ Visible] Variable $S1
Temperature vs Time
Line and symbol: Color, Edge color, Width 4, Width 8, Edge width 1
[✓ Line visible] [✓ Points visible]
Serie expressions:
X $N1.TH
Y $N1.ET
Assigned axes:
X Elapsed time, h
Y Temperature, °C
Linear regression: Visible, Range in points 0, Extrapolate

and the name to 'Furnace temperature'.



ted sweeps 1
omated sweeps 1 A10 Measure f
Axis: Elapsed time, h
Axis: Temperature, ºC
Serie: Furnace temperature

Furnace temperature

Title
[✓] Visible — Variable: $S1
Furnace temperature — Script

Line and symbol
4 — Color
Edge color
Width: 8
Edge width: 1
[✓] Line visible
[✓] Points visible

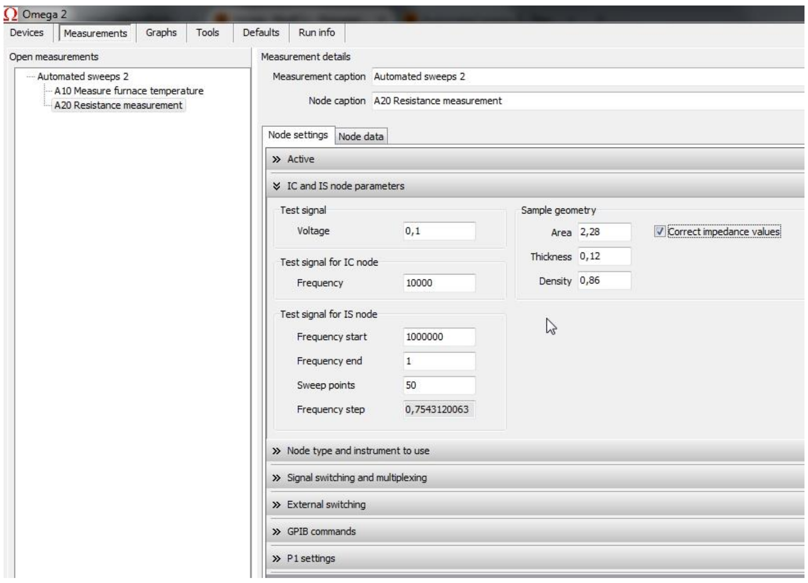
Serie expressions
X: $N1.TH
Y: $N1.ET

Assigned axes
X: Elapsed time, h
Y: Temperature, ºC

Linear regression
[ ] Visible — Range in points: 0 — [ ] Extrapolate

9.10.7 Step 3

To know if the sample is at chemical equilibrium, we need to measure the resistivity, plot it, and have linear least squares tool on the plot so that we know it's rate of change. First, add node, make it of IC type and select the impedance spectrometer to use. Mark node active, rename the node to 'Resistance measurement'. Since we speak of resistivity, we need to correct the measured values for geometry. This is done on the IS and IC tab.



Open measurements
Automated sweeps 2
A10 Measure furnace temperature
A20 Resistance measurement

Measurement details
Measurement caption: Automated sweeps 2
Node caption: A20 Resistance measurement

Test signal
Voltage: 0,1

Test signal for IC node
Frequency: 10000

Test signal for IS node
Frequency start: 1000000
Frequency end: 1
Sweep points: 50
Frequency step: 0,7543120063

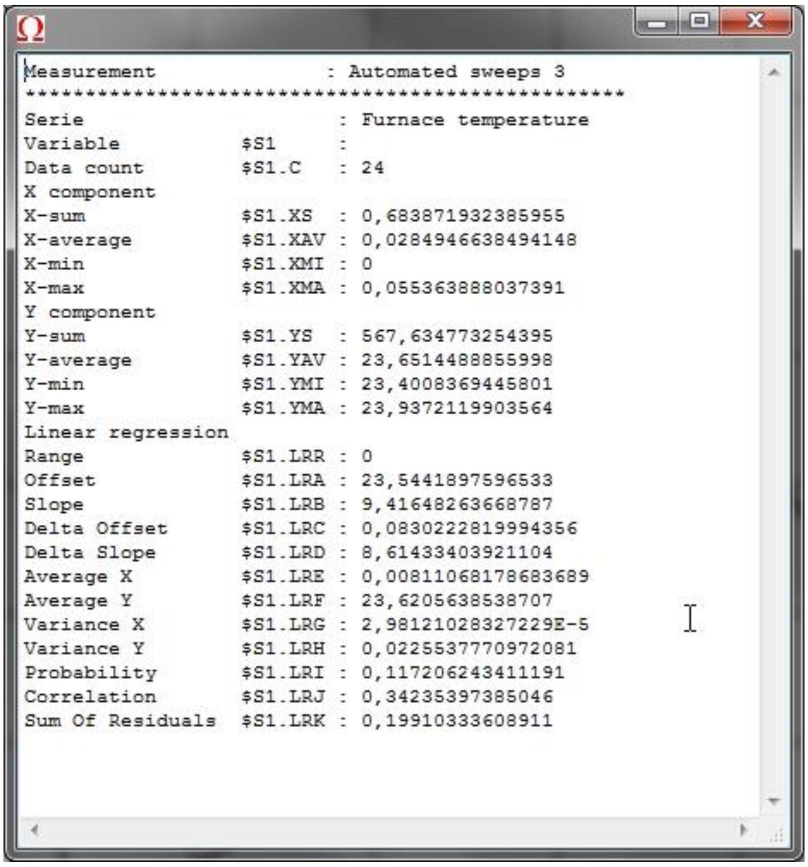
Sample geometry
Area: 2,28 — [✓] Correct impedance values
Thickness: 0,12
Density: 0,86

Node type and instrument to use
Signal switching and multiplexing
External switching
GPIB commands
P1 settings

Enter sample dimensions and tick the 'Correct for geometry' checkbox. The unit for electrode area and thickness must be same and will be the unit of the resistance related plots also such as Ohm*cm or Ohm*mm.

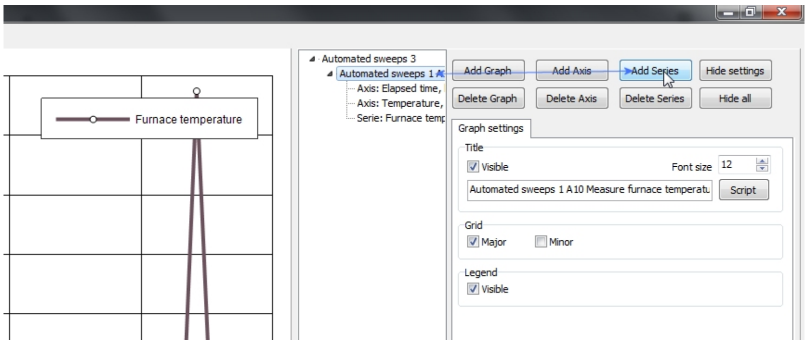
9.10.8 Step 4

To have access to the rate of change and a number of other properties



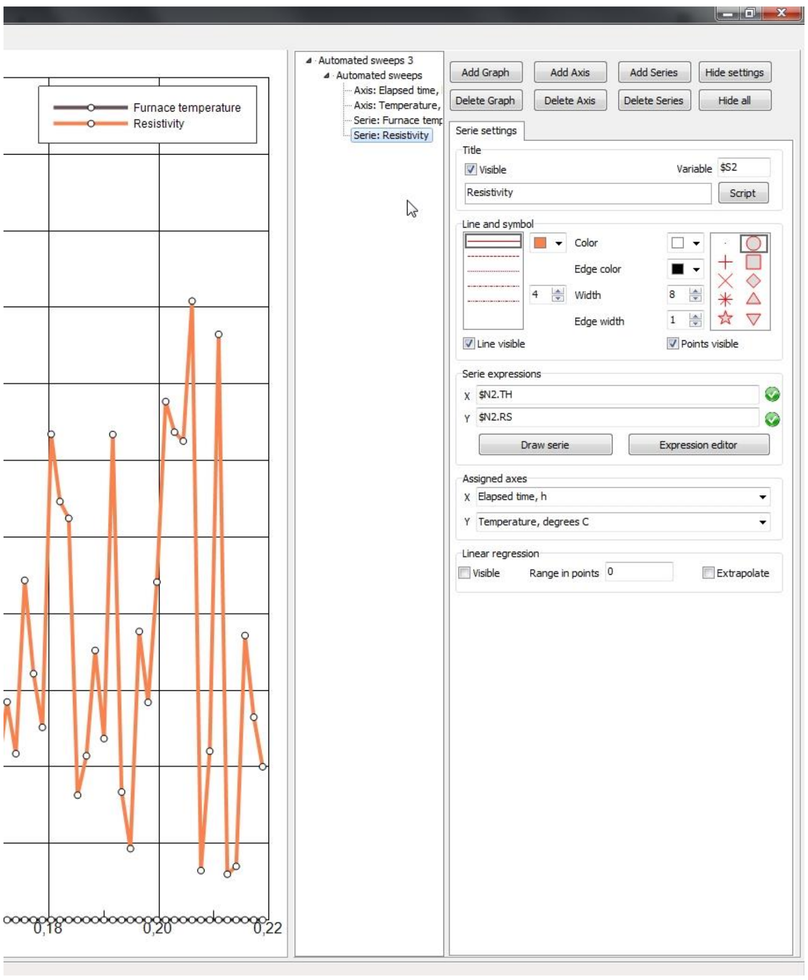
```
Measurement                  : Automated sweeps 3
***************************************************
Serie                         : Furnace temperature
Variable           $S1      :
Data count         $S1.C    : 24
X component
X-sum              $S1.XS   : 0,683871932385955
X-average          $S1.XAV  : 0,0284946638494148
X-min              $S1.XMI  : 0
X-max              $S1.XMA  : 0,055363888037391
Y component
Y-sum              $S1.YS   : 567,634773254395
Y-average          $S1.YAV  : 23,6514488855998
Y-min              $S1.YMI  : 23,4008369445801
Y-max              $S1.YMA  : 23,9372119903564
Linear regression
Range              $S1.LRR  : 0
Offset             $S1.LRA  : 23,5441897596533
Slope              $S1.LRB  : 9,41648263668787
Delta Offset       $S1.LRC  : 0,0830222819994356
Delta Slope        $S1.LRD  : 8,61433403921104
Average X          $S1.LRE  : 0,00811068178683689
Average Y          $S1.LRF  : 23,6205638538707
Variance X         $S1.LRG  : 2,98121028327229E-5
Variance Y         $S1.LRH  : 0,0225537770972081
Probability        $S1.LRI  : 0,117206243411191
Correlation        $S1.LRJ  : 0,34235397385046
Sum Of Residuals   $S1.LRK  : 0,19910333608911
```

the measured value needs to be plotted on the graph. To plot the resistivity we select the graph tab and select the graph and click 'add new serie' button.



[✓] Visible   Font size 12
Automated sweeps 1 A10 Measure furnace temperatu
[✓] Major   [ ] Minor
[✓] Visible

Edit the new serie



- Automated sweeps 3
  - Automated sweeps
    - Axis: Elapsed time,
    - Axis: Temperature,
    - Serie: Furnace temp
    - Serie: Resistivity

Serie settings

Title
- ☑ Visible — Variable: $S2
- Resistivity

Line and symbol
- Color: 🟧
- Edge color: ■
- 4 — Width: 8
- Edge width: 1
- ☑ Line visible — ☑ Points visible

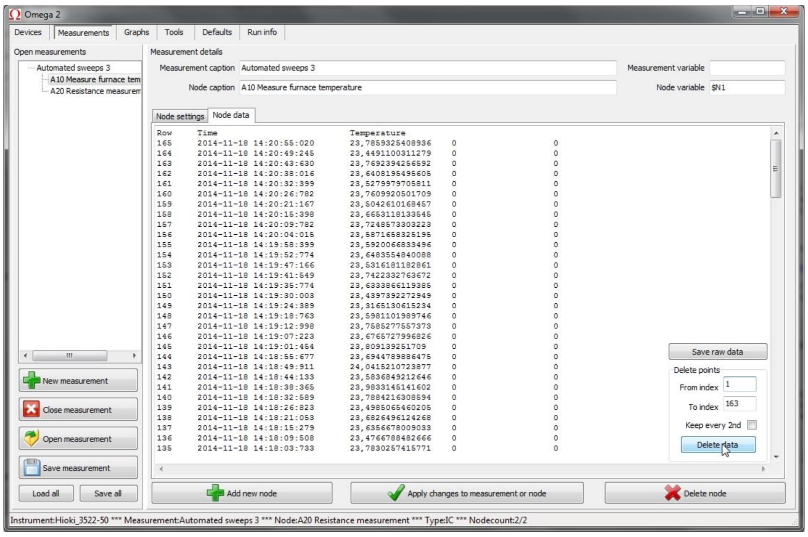
Serie expressions
- X: $N2.TH
- Y: $N2.RS

Assigned axes
- X: Elapsed time, h
- Y: Temperature, degrees C

Linear regression
- ☐ Visible — Range in points: 0 — ☐ Extrapolate

to have Resistivity as name, $N2.TH as X expression and $N2.RS as Y expression.

If the new graph does not align with the old one on X-axis it is due to the fact that the X component we are using (.TH) is elapsed time in hours since beginning of the node, and the second node was just created, so different amount of time have elapsed for the nodes. To remedy this we need to delete all the old points before the second node existed. Go to the node 1 and to the node data tab. Enter to the delete data tool



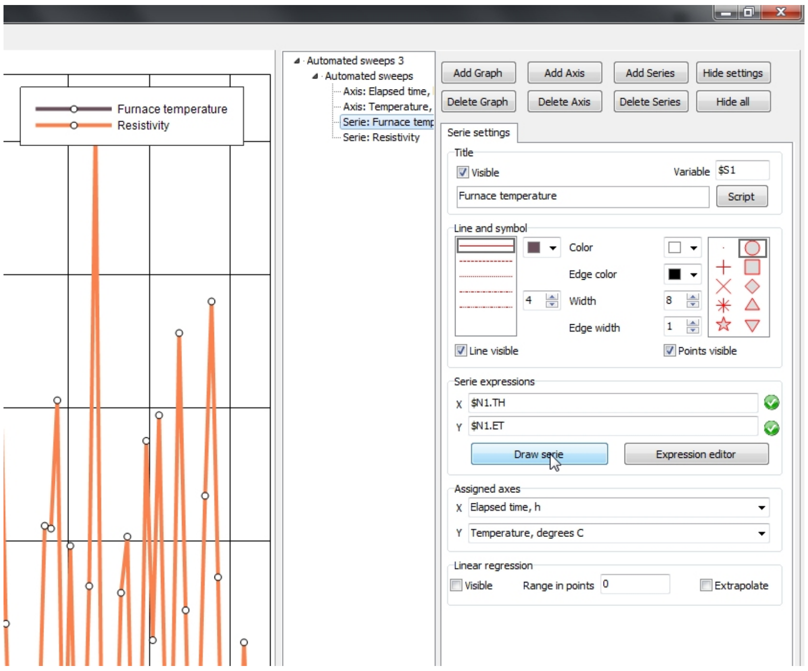
Measurement details
Measurement caption: Automated sweeps 3
Node caption: A10 Measure furnace temperature
Measurement variable:
Node variable: $N1

```
Row     Time                        Temperature
165     2014-11-18 14:20:55:020     23,7859325408936    0       0
164     2014-11-18 14:20:49:245     23,4491100311279    0       0
163     2014-11-18 14:20:43:630     23,7692394256592    0       0
162     2014-11-18 14:20:38:016     23,6408195495605    0       0
161     2014-11-18 14:20:32:399     23,5279979705811    0       0
160     2014-11-18 14:20:26:782     23,7609920501709    0       0
159     2014-11-18 14:20:21:167     23,5042610168457    0       0
158     2014-11-18 14:20:15:398     23,6653118133545    0       0
157     2014-11-18 14:20:09:782     23,7248573303223    0       0
156     2014-11-18 14:20:04:015     23,5871658325195    0       0
155     2014-11-18 14:19:58:399     23,5920066833496    0       0
154     2014-11-18 14:19:52:774     23,6483554840088    0       0
153     2014-11-18 14:19:47:166     23,5316181182861    0       0
152     2014-11-18 14:19:41:549     23,7422332763672    0       0
151     2014-11-18 14:19:35:774     23,6333866119385    0       0
150     2014-11-18 14:19:30:003     23,4397392272949    0       0
149     2014-11-18 14:19:24:389     23,3165130615234    0       0
148     2014-11-18 14:19:18:763     23,5981101989746    0       0
147     2014-11-18 14:19:12:998     23,7585277557373    0       0
146     2014-11-18 14:19:07:223     23,6765727996826    0       0
145     2014-11-18 14:19:01:454     23,809139251709     0       0
144     2014-11-18 14:18:55:677     23,6944789886475    0       0
143     2014-11-18 14:18:49:911     24,0415210723877    0       0
142     2014-11-18 14:18:44:133     23,5836849212646    0       0
141     2014-11-18 14:18:38:365     23,9833145141602    0       0
140     2014-11-18 14:18:32:589     23,7884216308594    0       0
139     2014-11-18 14:18:26:823     23,4985065460205    0       0
138     2014-11-18 14:18:21:053     23,6826496124268    0       0
137     2014-11-18 14:18:15:279     23,6356678009033    0       0
136     2014-11-18 14:18:09:508     23,4766788482666    0       0
135     2014-11-18 14:18:03:733     23,7830257415771    0       0
```

Delete points
From index: 1
To index: 163
Keep every 2nd: ☐

Instrument:Hioki_3522-50 *** Measurement:Automated sweeps 3 *** Node:A20 Resistance measurement *** Type:IC *** Nodecount:2/2

from index 1 to index (the highest number you see there) and click delete. After done, go to the graphs and select each serie at a time and click redraw serie button for each.



Title
☑ Visible Variable: $S1
Furnace temperature

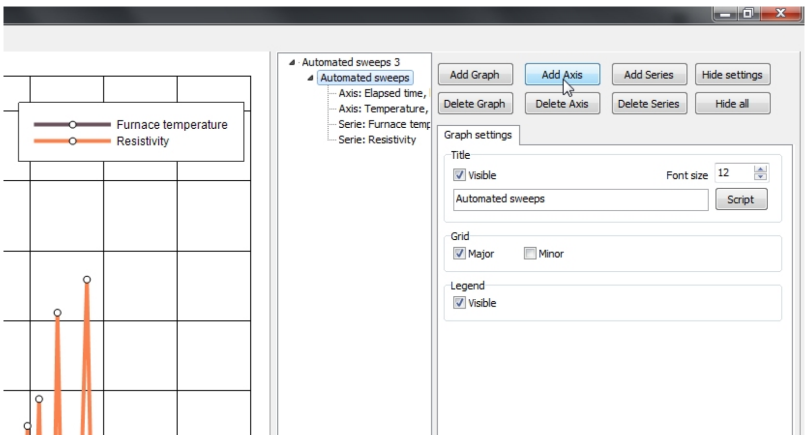
Line and symbol
Color
Edge color
Width: 4 / 8
Edge width: 1
☑ Line visible ☑ Points visible

Serie expressions
X: $N1.TH
Y: $N1.ET

Assigned axes
X: Elapsed time, h
Y: Temperature, degrees C

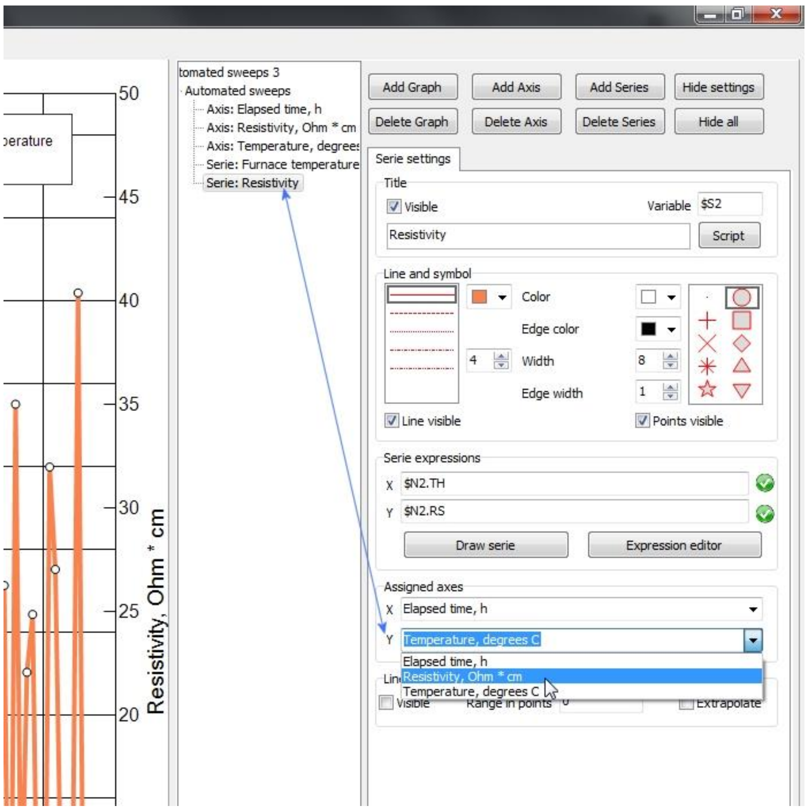
Linear regression
☐ Visible Range in points: 0 ☐ Extrapolate

Next we need to add new Y-axis for the resistivity since it does not necessarily share the scale of furnace temperature very well. Select the graph and click add new axis.



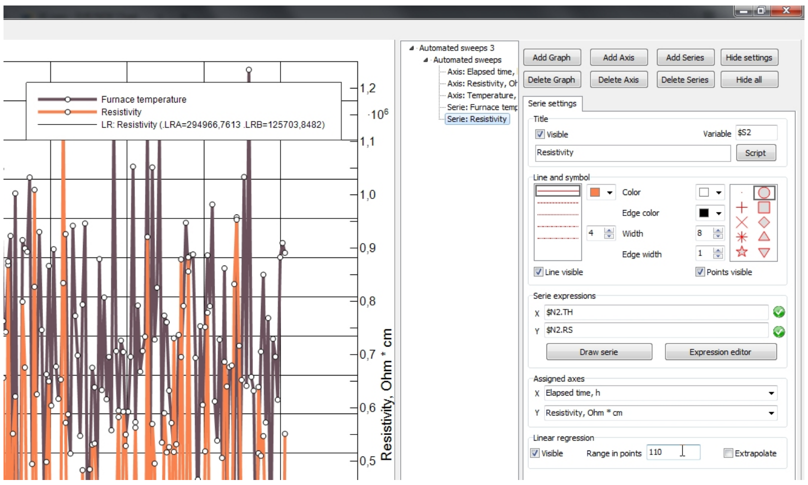
Title
☑ Visible Font size 12
Automated sweeps Script
Grid
☑ Major ☐ Minor
Legend
☑ Visible

Then select the new axis and edit its name axis to Resistivity, Ohm*cm or the unit you used. On the Resistivity serie edit the Y-axis and select the 'Resistivity' axis to assign the data to be plotted against the newly created Y-axis.



Title
☑ Visible Variable $S2
Resistivity Script
Line and symbol
Color
Edge color
4 Width 8
Edge width 1
☑ Line visible ☑ Points visible
Serie expressions
X $N2.TH
Y $N2.RS
Draw serie Expression editor
Assigned axes
X Elapsed time, h
Y Temperature, degrees C
Elapsed time, h
Resistivity, Ohm * cm
Temperature, degrees C
☐ Visible Range in points 0 ☐ Extrapolate

Now enable the coefficient tool for the Resistivity serie by ticking the checkbox, and selecting number of points to include in the calculation.



Automated sweeps 3
Automated sweeps
Axis: Elapsed time,
Axis: Resistivity, Oh
Axis: Temperature,
Serie: Furnace temp
Serie: Resistivity

Serie settings

Title
[✓] Visible
Variable: $S2
Resistivity

Line and symbol
Color
Edge color
Width: 4
Width: 8
Edge width: 1
[✓] Line visible
[✓] Points visible

Serie expressions
X: $N2.TH
Y: $N2.RS

Assigned axes
X: Elapsed time, h
Y: Resistivity, Ohm * cm

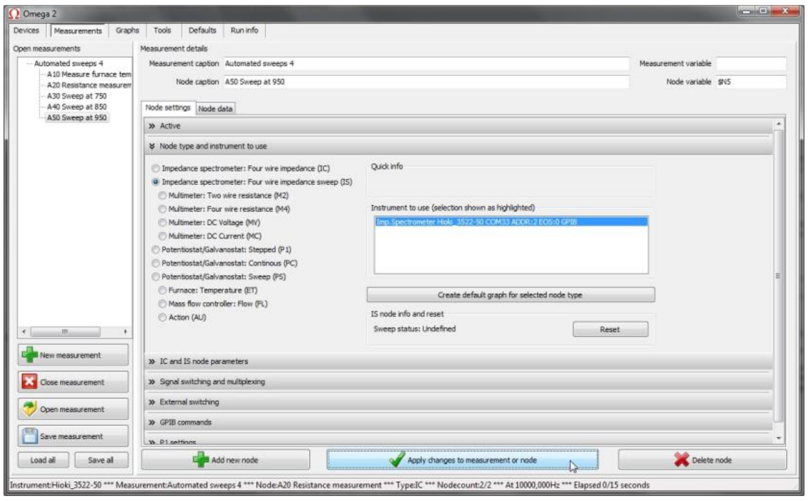
Linear regression
[✓] Visible
Range in points: 110
[ ] Extrapolate

From the Node data we can see that we are getting one point of data per 5-6 seconds, making our rate 11 points per minute or 660 points per hour. There is no correct value to enter here and it depends on what the user is hoping to accomplish. Obviously observing a shorter time gives faster automation but may leave user hoping for better accuracy (for example if value is oscillating at a frequency that is the observed time). In some measurements the sample equilibrium can take weeks to settle. In this example we will use 20 minutes, aka 220 points.

9.10.9 Step 5

We now have all the prerequisites for judging if the sample is at equilibrium, next we need to add the automation nodes for the furnace control and the sweeps. Our plan is to have impedance sweeps at 750, 850 and 950ºC after the resistivity has stabilized.

First we need 3 new nodes for the 3 sweeps. Add 3 new nodes and rename them accordingly.



Open measurements
Automated sweeps 4
A10 Measure furnace tem
A20 Resistance measurem
A30 Sweep at 750
A40 Sweep at 850
A50 Sweep at 950

Measurement details
Measurement caption: Automated sweeps 4
Measurement variable:
Node caption: A50 Sweep at 950
Node variable: $N5

( ) Impedance spectrometer: Four wire impedance (IC)
(•) Impedance spectrometer: Four wire impedance sweep (IS)
( ) Multimeter: Two wire resistance (M2)
( ) Multimeter: Four wire resistance (M4)
( ) Multimeter: DC Voltage (MV)
( ) Multimeter: DC Current (MC)
( ) Potentiostat/Galvanostat: Stepped (P1)
( ) Potentiostat/Galvanostat: Continous (PC)
( ) Potentiostat/Galvanostat: Sweep (PS)
( ) Furnace: Temperature (ET)
( ) Mass flow controller: Flow (FL)
( ) Action (AU)

Quick info

Instrument to use (selection shown as highlighted)
Imp.Spectrometer Hioki_3522-50 COM33 ADDR:2 EOS:0 GPIB

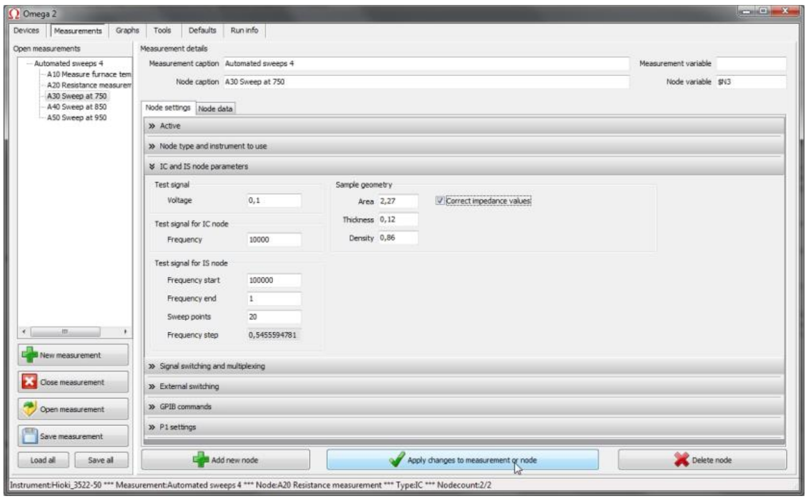
IS node info and reset
Sweep status: Undefined

Instrument:Hioki_3522-50 *** Measurement:Automated sweeps 4 *** Node:A20 Resistance measurement *** Type:IC *** Nodecount:2/2 *** At 10000,000Hz *** Elapsed 0/15 seconds

Leave all new nodes in active for now, but make them of type IS and select the impedance spectrometer. Remember to click apply.

Node 3, 4 and 5

Enter the desired Sweep parameters



Measurement details
Measurement caption: Automated sweeps 4
Node caption: A30 Sweep at 750
Measurement variable:
Node variable: $N3

Test signal
Voltage: 0,1

Test signal for IC node
Frequency: 10000

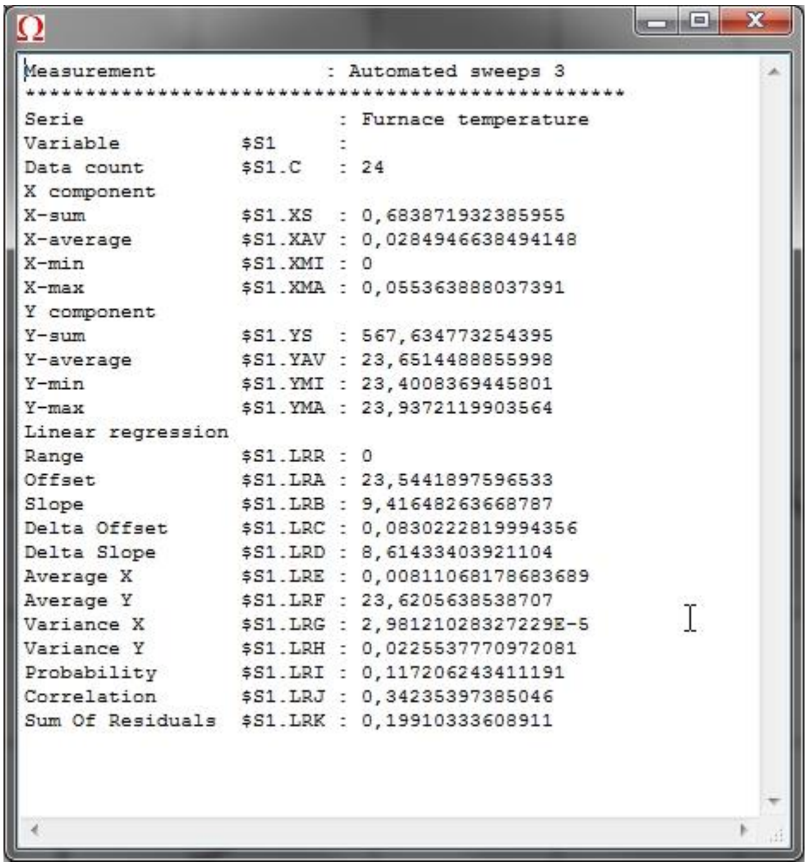
Test signal for IS node
Frequency start: 100000
Frequency end: 1
Sweep points: 20
Frequency step: 0,5455594781

Sample geometry
Area: 2,27
Thickness: 0,12
Density: 0,86
[✓] Correct impedance values

Instrument:Hioki_3522-50 *** Measurement:Automated sweeps 4 *** Node:A20 Resistance measurement *** Type:IC *** Nodecount:2/2

such as start and end frequency and click apply. Note that the geometry needs to be entered again. This software is designed to support multiple samples and does not know to which sample it is connected to. (From version 2.2.1 on it is possible to clone nodes to save typing in all the details.)

The 'active' tab parameters determine when a node is active. At it's simplest it is just an On/Off selector, but can be used for timing and automation. The expression parser can perform logical operations such as < or > based on the measured properties such as $N1.ET (node 1 measured furnace temperature), $N2.RS (node 2 measured resistance) or the graphical series such as series minimum, maximum, average, or as in this case, the angle of the coefficient of the linear least squares tool. The node variable can be found on the node properties page (the property endings are listed in the manual), and the Serie variable can be found from the Graph tab: right click on a graph -> show serie variables to see summary of serie related variables.



```
Measurement                  : Automated sweeps 3
**************************************************
Serie                         : Furnace temperature
Variable          $S1         :
Data count        $S1.C       : 24
X component
X-sum             $S1.XS      : 0,683871932385955
X-average         $S1.XAV : 0,0284946638494148
X-min             $S1.XMI : 0
X-max             $S1.XMA : 0,055363888037391
Y component
Y-sum             $S1.YS      : 567,634773254395
Y-average         $S1.YAV : 23,6514488855998
Y-min             $S1.YMI : 23,4008369445801
Y-max             $S1.YMA : 23,9372119903564
Linear regression
Range             $S1.LRR : 0
Offset            $S1.LRA : 23,5441897596533
Slope             $S1.LRB : 9,41648263668787
Delta Offset      $S1.LRC : 0,0830222819994356
Delta Slope       $S1.LRD : 8,61433403921104
Average X         $S1.LRE : 0,00811068178683689
Average Y         $S1.LRF : 23,6205638538707
Variance X        $S1.LRG : 2,98121028327229E-5
Variance Y        $S1.LRH : 0,0225537770972081
Probability       $S1.LRI : 0,117206243411191
Correlation       $S1.LRJ : 0,34235397385046
Sum Of Residuals  $S1.LRK : 0,19910333608911
```

For node 3 we use the following start condition:

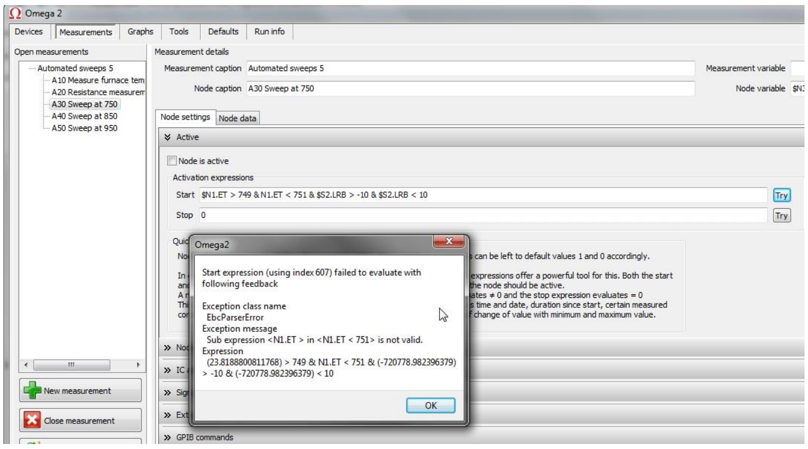
$N1.ET > 749 & $N1.ET < 751 & $S2.LRB > -10 & $S2.LRB < 10

The node will not start until the whole sentence this evaluates to true. In words, we added four conditions that all must be true for the node to start:

Measured furnace temperature must be higher than 749
Measured furnace temperature must be lower than 751
Slope of the linear least squares coefficient must be > -10
Slope of the linear least squares coefficient must be < 10
Effectively point 3 and 4 mean that the rate of change for measured property must be between -10 and 10. The unit of this comparison depends what is being plotted on the serie, in this example it is resistivity per hour.

When these conditions are fulfilled, the node becomes active and does what it is set to do.

For filling in and testing the expressions the user also have a 'Try' button next to the expression that does not do anything except try to evaluate the expression and give feedback in case of errors in the formula. In this image a missing $ renders the expression un-parseable and is fixed in this picture.



Automated sweeps 5
A10 Measure furnace tem
A20 Resistance measurem
A30 Sweep at 750
A40 Sweep at 850
A50 Sweep at 950

Measurement details
Measurement caption: Automated sweeps 5
Node caption: A30 Sweep at 750
Measurement variable:
Node variable: $N3

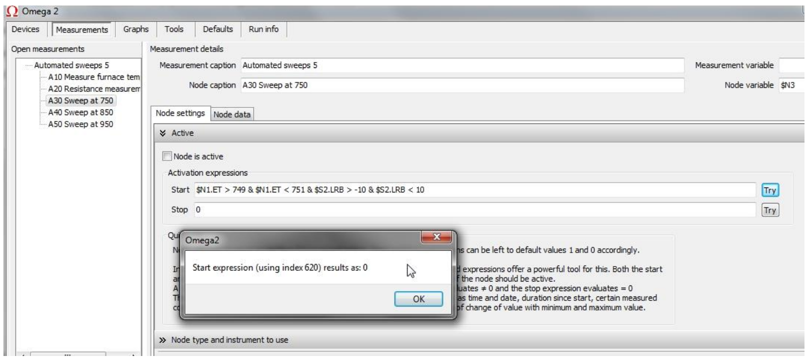
☐ Node is active
Activation expressions
Start: $N1.ET > 749 & N1.ET < 751 & $S2.LRB > -10 & $S2.LRB < 10
Stop: 0

Omega2

```
Start expression (using index 607) failed to evaluate with
following feedback

Exception class name
  EbcParserError
Exception message
  Sub expression <N1.ET > in <N1.ET < 751> is not valid.
Expression
  (23.8188800811768) > 749 & N1.ET < 751 & (-720778.982396379)
> -10 & (-720778.982396379) < 10
```

s can be left to default values 1 and 0 accordingly.
expressions offer a powerful tool for this. Both the start
the node should be active.
ates ≠ 0 and the stop expression evaluates = 0
s time and date, duration since start, certain measured
f change of value with minimum and maximum value.



Automated sweeps 5
A10 Measure furnace tem
A20 Resistance measurem
A30 Sweep at 750
A40 Sweep at 850
A50 Sweep at 950

Measurement details
Measurement caption: Automated sweeps 5
Node caption: A30 Sweep at 750
Measurement variable:
Node variable: $N3

☐ Node is active
Activation expressions
Start: $N1.ET > 749 & $N1.ET < 751 & $S2.LRB > -10 & $S2.LRB < 10
Stop: 0

Omega2

```
Start expression (using index 620) results as: 0
```

s can be left to default values 1 and 0 accordingly.
d expressions offer a powerful tool for this. Both the start
f the node should be active.
uates ≠ 0 and the stop expression evaluates = 0
as time and date, duration since start, certain measured
of change of value with minimum and maximum value.

Since in this example this node is of type IS, we do not need to set the stop expression. By nature IS nodes will perform from start to end and then become inactive automatically.

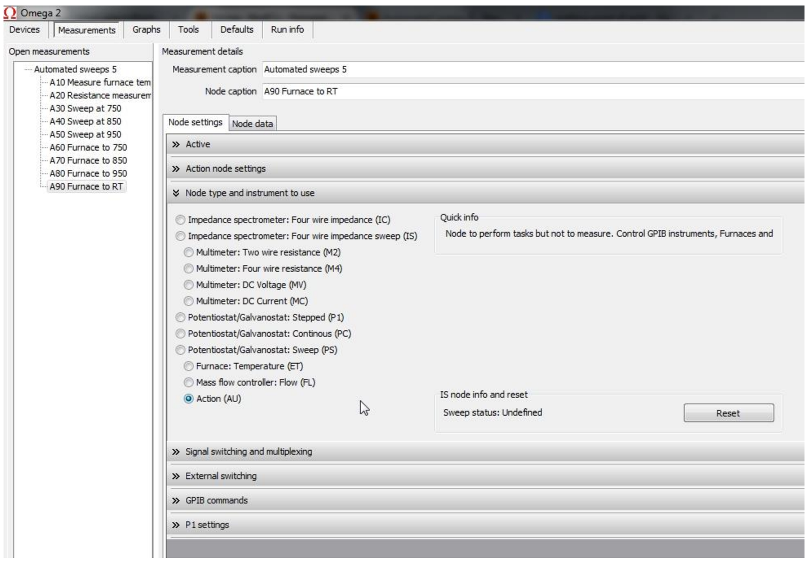
Perform the same changes to the nodes 4 and 5 with start expression adjusted for the temperatures required.

9.10.10 Step 6

To send new setpoint to the furnace we need action node (AU). To make things simple, we will add 4 of them. Three to bring the temperature to 750, 850 and 950 accordingly. One to bring the temperature down to room temperature once everything is done. Alternatively a single node can be used to do all this, and will be included as alternative: Save file 6A is with four nodes, save file 6B is with one node. Both approaces are explained here.

9.10.11 Step 6A

Add 4 nodes and rename the according this picture, and make all four the type AU.



Open measurements
- Automated sweeps 5
  - A10 Measure furnace tem
  - A20 Resistance measurem
  - A30 Sweep at 750
  - A40 Sweep at 850
  - A50 Sweep at 950
  - A60 Furnace to 750
  - A70 Furnace to 850
  - A80 Furnace to 950
  - A90 Furnace to RT

Measurement details

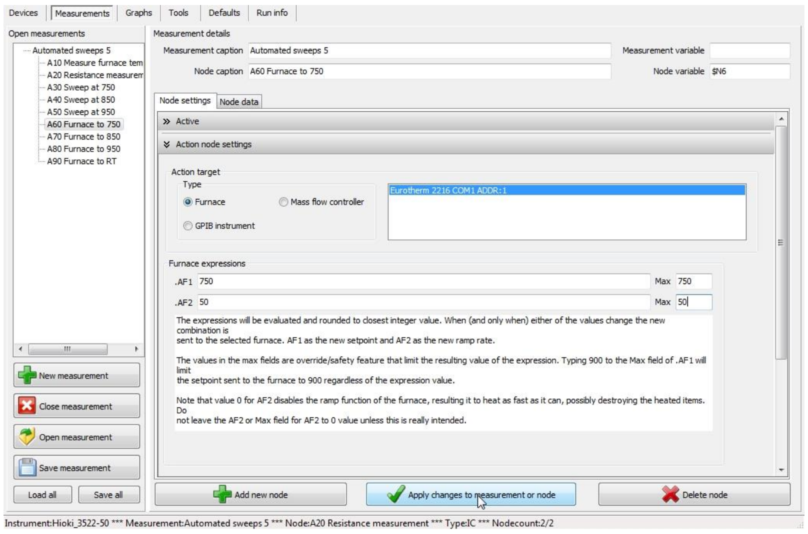
Measurement caption: Automated sweeps 5
Node caption: A90 Furnace to RT

Node type and instrument to use
- Impedance spectrometer: Four wire impedance (IC)
- Impedance spectrometer: Four wire impedance sweep (IS)
- Multimeter: Two wire resistance (M2)
- Multimeter: Four wire resistance (M4)
- Multimeter: DC Voltage (MV)
- Multimeter: DC Current (MC)
- Potentiostat/Galvanostat: Stepped (P1)
- Potentiostat/Galvanostat: Continous (PC)
- Potentiostat/Galvanostat: Sweep (PS)
- Furnace: Temperature (ET)
- Mass flow controller: Flow (FL)
- [✓ Action (AU)]

Quick info
Node to perform tasks but not to measure. Control GPIB instruments, Furnaces and

IS node info and reset
Sweep status: Undefined

Edit each of them



Open measurements
- Automated sweeps 5
  - A10 Measure furnace tem
  - A20 Resistance measurem
  - A30 Sweep at 750
  - A40 Sweep at 850
  - A50 Sweep at 950
  - A60 Furnace to 750
  - A70 Furnace to 850
  - A80 Furnace to 950
  - A90 Furnace to RT

Measurement details

Measurement caption: Automated sweeps 5
Node caption: A60 Furnace to 750
Measurement variable:
Node variable: $N6

Action target
Type: [✓ Furnace] Mass flow controller GPIB instrument
Eurotherm 2216 COM1 ADDR:1

Furnace expressions
.AF1 750 Max 750
.AF2 50 Max 50

The expressions will be evaluated and rounded to closest integer value. When (and only when) either of the values change the new combination is
sent to the selected furnace. AF1 as the new setpoint and AF2 as the new ramp rate.

The values in the max fields are override/safety feature that limit the resulting value of the expression. Typing 900 to the Max field of .AF1 will limit
the setpoint sent to the furnace to 900 regardless of the expression value.

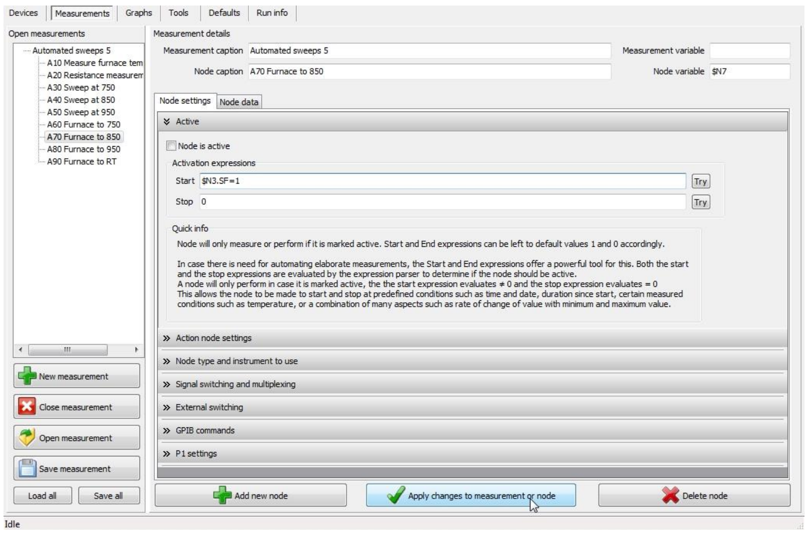
Note that value 0 for AF2 disables the ramp function of the furnace, resulting it to heat as fast as it can, possibly destroying the heated items. Do
not leave the AF2 or Max field for AF2 to 0 value unless this is really intended.

Instrument:Hioki_3522-50 *** Measurement:Automated sweeps 5 *** Node:A20 Resistance measurement *** Type:IC *** Nodecount:2/2

to have action target as furnace and select the furnace from the list. Click apply for each edit you make. AF1 is the new setpoint and AF2 is the ramp rate (with same unit as the furnace controller has, typically degrees per minute or deci-degrees per minute. In this example the furnace uses deci degrees so 50 would result to 5 per minute) Use 750, 850, 950 and 0 as AF1 and 50 as AF2 for the four nodes.

Last and most important we set the parameters to activate the nodes. The node 6 can start the heating as soon as we activate everything so the start and stop conditions can remain as they are.

The node 7 (Furnace to 850) can only start after the sweep at 750 has finished, so for the start condition here we enter $N3.SF=1



Measurement details

Measurement caption: Automated sweeps 5
Measurement variable:
Node caption: A70 Furnace to 850
Node variable: $N7

☐ Node is active

Activation expressions
Start: $N3.SF=1
Stop: 0

Quick info

Node will only measure or perform if it is marked active. Start and End expressions can be left to default values 1 and 0 accordingly.

In case there is need for automating elaborate measurements, the Start and End expressions offer a powerful tool for this. Both the start and the stop expressions are evaluated by the expression parser to determine if the node should be active.
A node will only perform in case it is marked active, the the start expression evaluates ≠ 0 and the stop expression evaluates = 0
This allows the node to be made to start and stop at predefined conditions such as time and date, duration since start, certain measured conditions such as temperature, or a combination of many aspects such as rate of change of value with minimum and maximum value.

The $N3.SF is a property to test if a sweep is finished: the value is either 0 (undefined or unfinished) or 1 (sweep has finished). Testing for $N3.SF=1 is true (or 1) when the sweep has finished, and false (or 0) otherwise.

Last step is to mark all nodes active.

9.10.12 Alternative step 6B

Instead of having 4 nodes for the temperature control, it is possible to achieve the same functionality with just one node.

We only add one node after the Sweep nodes, and make the node 6 as AU type, click apply and go to the action node settings. Select Furnace control and the correct furnace from the list. AF1 is the new setpoint and AF2 is the ramp rate (with same unit as the furnace controller has, typically degrees per minute or deci-degrees per minute as in this example so 50 would result to 5 per minute)

This node will use AF1 expression to control the furnace setpoint. As feedback it will use the sweep nodes, if they are finished or not.

Consider expression: IF A=X then B, otherwise C. We will utilise the expression parser here to control the furnace depending which sweeps have finished.

IF($N3.SF=0, 750, 0) This expression would would evaluate to 750 is the node 3 sweep has finished, otherwise it would evaluate to 0. But instead of 0, we can add more evaluations like below:

IF($N3.SF=0, 750, IF($N4.SF=0, 850, IF($N5.SF=0, 950, 0)))

This can be read in words as follows:

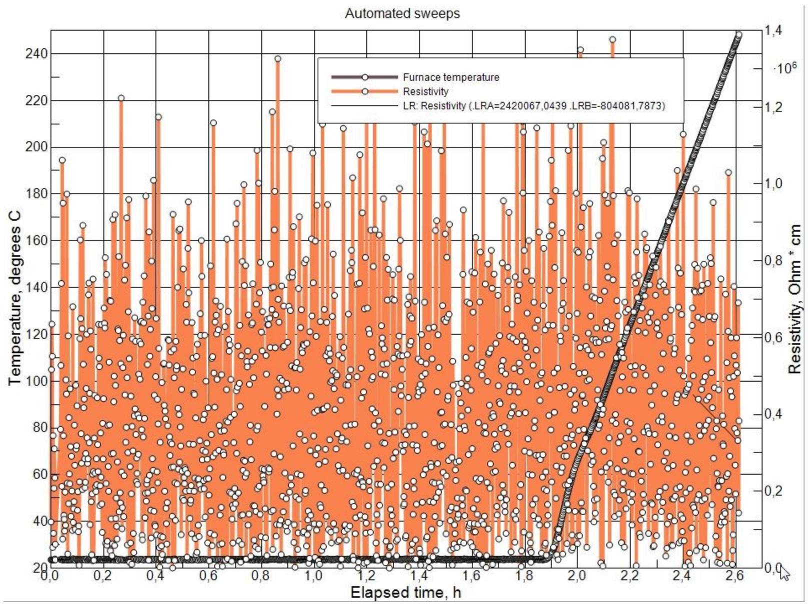
> If node 3 sweep has finished, result of this all is 750, but if not, then
>
> > If node 4 sweep has finished, result of this all is 850, but if not, then
> >
> > > If node 5 sweep has finished, result of this all is 950, but if not, then the result is 0

We can use the above expression as expression for the furnace setpoint, AF1 and 50 for AF2, the ramp rate.

To measure, mark all nodes active.

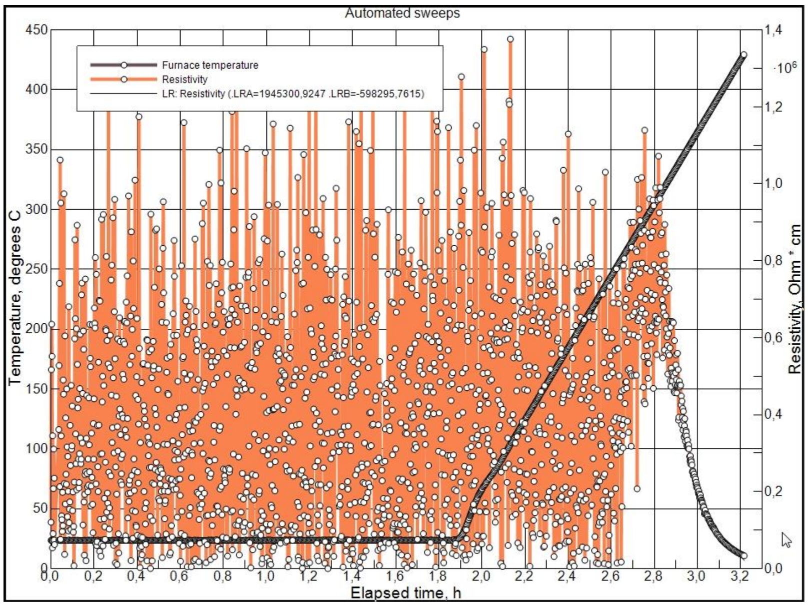
9.10.13 Followup, steps 7-13

All nodes active and the measurement started we can see how the process develops.



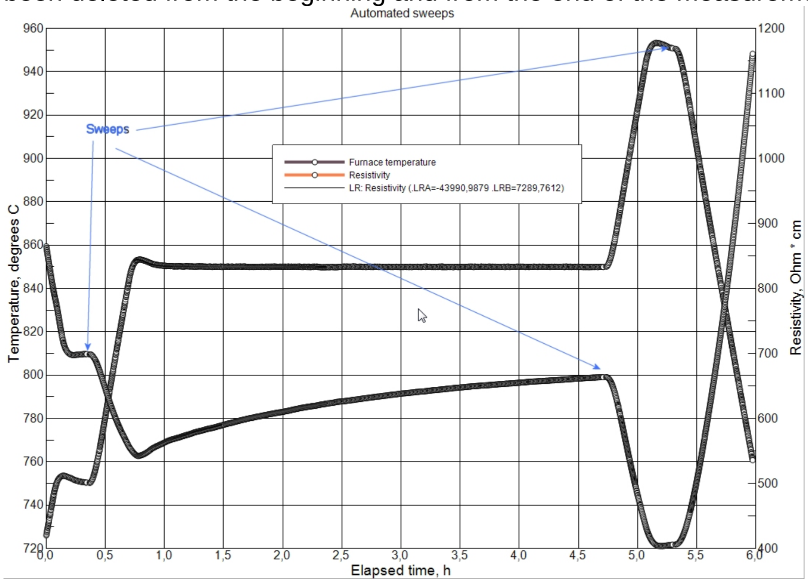
We see the resistivity graph filling the whole screen and with very high values. This is because the sample is not yet conductive so we measure high resistances. Also, the Resistivity axis zooms in as close as it can to the values. Once the values start to change, like they do for temperature, the graph starts to make more sense. Also notable is the coefficient tool for the resistivity serie on the bottom right corner.

A little later the sample has started to conduct resulting decreasing resistivity as seen here.



I will delete the old data points so that the interesting features of the graph can be better seen.

The measurement after the whole automation is performed (file 13B). Note that some data points has been deleted from the beginning and from the end of the measurement. You can easily see



that Sweeps 3 and 5 took place relatively soon after the temperature had stabilized while Sweep 4 took a long time before it started. Something that could have easily been missed had the sweeps initiated just by hand after reaching the target temperature.

9.11 Waveforms

Measurements may require user specified waveforms, and the expression parser in Omega software is powerful enough to create almost any type of wave forms. This is quick reference index to some typical wave forms, from index 0 to 100.

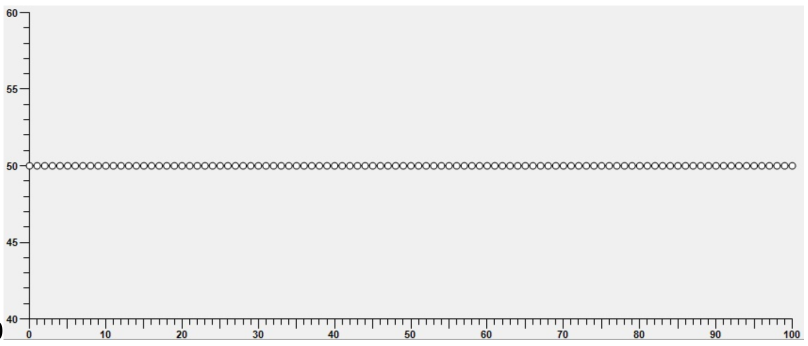
Creating your own wave forms is easy. Create new measurement, create new graph, add a series, and edit the properties of the series. On the properties area click the 'Expression editor' and design your own waveforms. Remember to set the range, from 'Index start' to 'Index stop'. It is advisable to use expression $I for X value when designing waveforms.

In these examples the function is connected to the index number of the data point, the $I. However, there is no limitation to what information can be used as the function input, it can be any measured values, elapsed time, time of the day, rate of change on some plotted value or number of other things. Typical value to use instead if $I would be elapsed minutes e.g. $N1.TM

Explanation and expressions used

Straight line

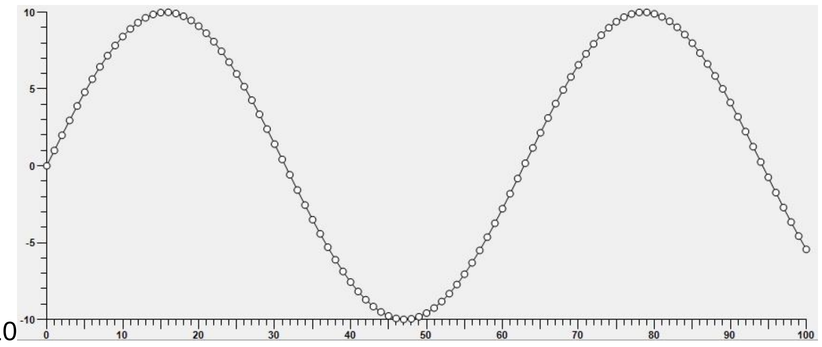
X: $I



Y: 50

Sine wave

X: $I

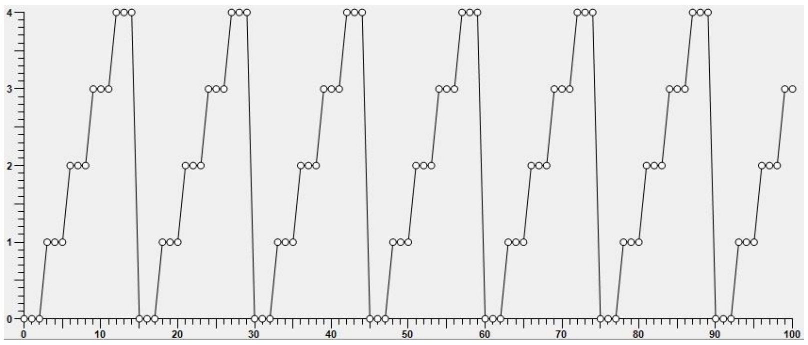


Y: SIN($I/10)*10

Flat multilevel recurring

X: $I
Y: Mod($I/3,5)

Explanation and expressions used

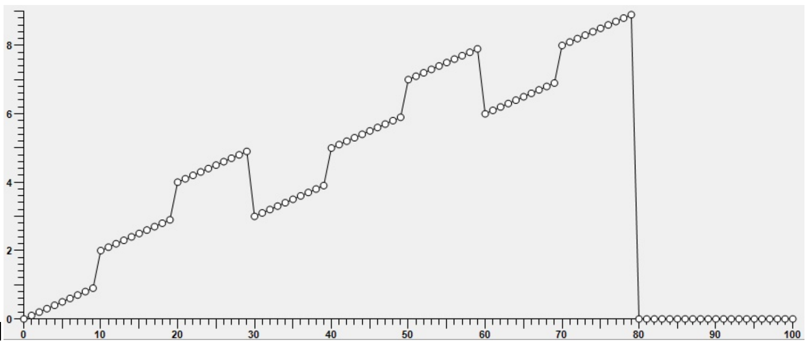


Function Mod(A,B) returns the modulus ('integer leftover') of A divided by B in Euclidean fashion

This is highly useful function in generating repeating waveforms.

Simplified, the MOD($I/X, Y) generates pattern of X repeats per level and Y levels.

Using IF for safety

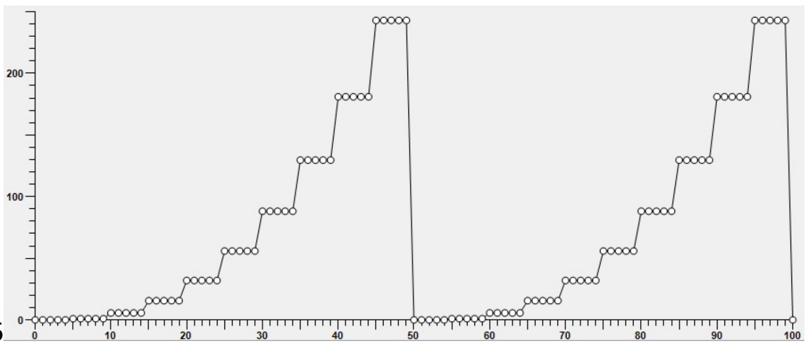


X: $I
Y: IF($I<80, Mod($I/10,3)+($I/10), 0)

Function IF(A,B,C). If A is 0, return C, else return B.

Progressive power step

X: $I
Y: POW(MOD($I/5,10), 2.5)



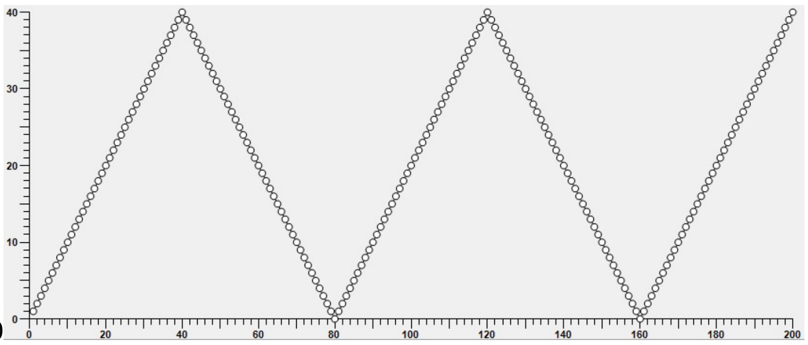
Mod expression to power of 2.5

Saw

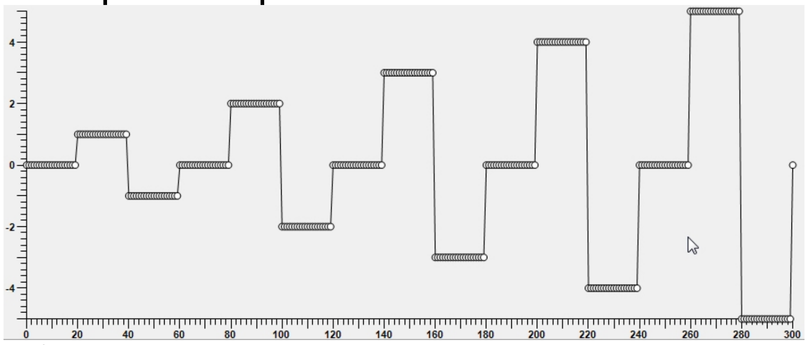
X: $I
Y: IF(MOD($I/40,2)=0, MOD($I,40), ABS(40-MOD($I,40)))

Explanation and expressions used

Index 0-200



Bias or potential step



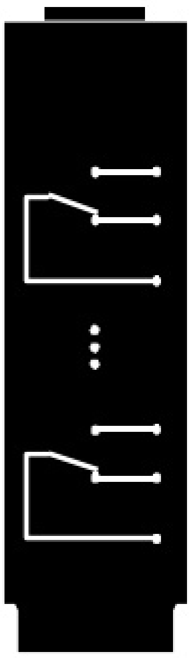
X: \$I
Y: (MOD(\$I/60,5)+1)*(MOD((\$I+20)/20,3)-1)

Instead of \$I, the elapsed minutes can be used: \$N1.TM

9.12 Van der Pauw electrode rotation

In this application note is a short description of utilizing 'Agilent 34970A Data Acquisition/Switch Unit' with '34903A 20-Channel Actuator / General-Purpose Switch' with Omega software to achieve automated rotating of four electrodes for Van der Pauw measurements.

Details of the actuator module



34903A 20-Channel Actuator / General-Purpose Switch
• 300 V, 1 A actuation and switching
• SPDT (Form C) latching relays
• Breadboard area for custom circuits
• For detailed information and a module diagram, see page 168.

Use this module for those applications that require high-integrity contacts or quality connections of non-multiplexed signals. This module can switch 300 V, 1 A (50 W maximum switch power) to your device under test or to actuate external devices. Screw terminals on the module provide access to the Normally-Open, Normally-Closed, and Common contacts for each of the 20 switches. A breadboard area is provided near the screw terminals to implement custom circuitry, such as simple filters, snubbers, or voltage dividers.

This module contains 20 independent, SPDT (Form C) latching relays. Screw terminals on the module provide access to the Normally-Open, Normally-Closed, and Common contacts for each switch. This module does not connect to the internal DMM. A breadboard area is provided near the screw terminals to implement custom circuitry, such as simple filters, snubbers, and voltage dividers. The breadboard area provides the space necessary to insert your own components but there are no circuit board traces here. You must add your own circuitry and signal routing.

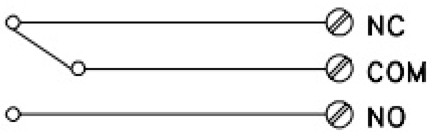
Intended behaviour

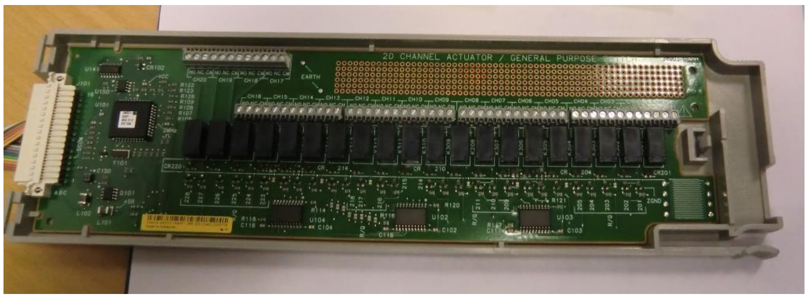
The state of the switches 1-4 (later referred as channels) are all changed at the same time to either open or closed, effectively achieving rotating electrodes; The instrument contacts HC, HV, LV and LC (High current, High voltage, Low voltage and Low current) connecting to the electrodes A, B, C and D.
Wiring details

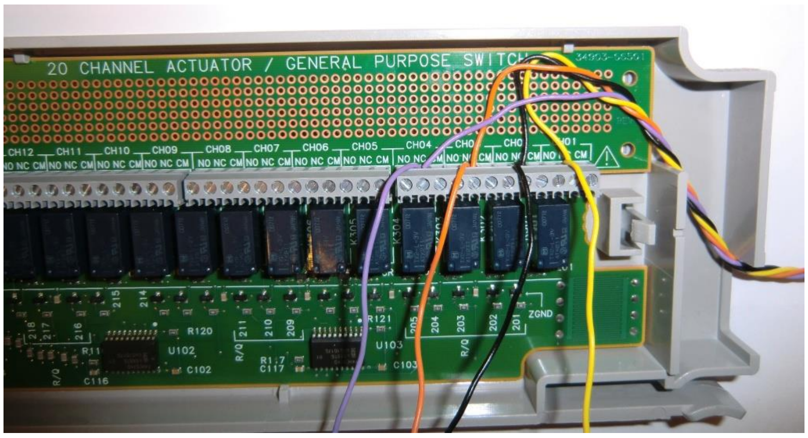
First the four electrode wires are connected to each channel on the NC (Normally closed) terminal, with additional short wire piece also attached to the terminal.
The additional piece from each terminal is connected to the NO (Normally opne) terminal of the previous channel. Channel 1 NO connects to channel 4 NO.
The four instrument terminal wires are connected to each channels CM (Common) terminal.
Rotating the electrodes with Omega software

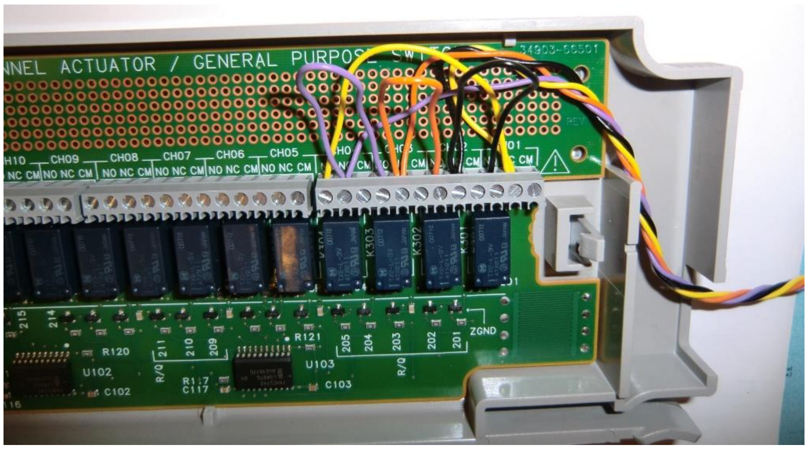
The channels were opened on every second measurement loop and closed the other every second loop.

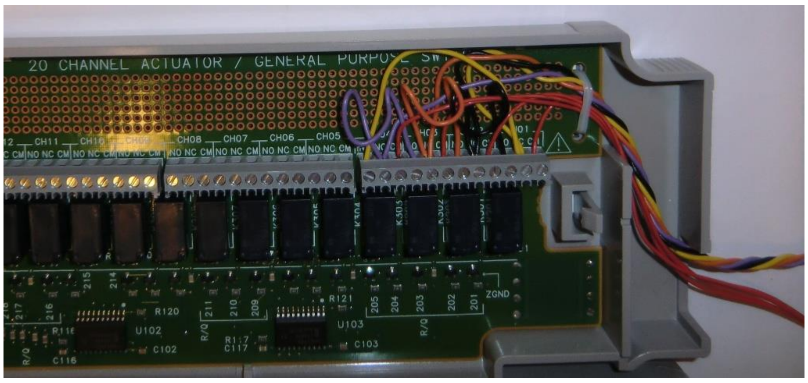
Measurement file for the control can be found in the files section, named 'VdP template for Agilent 34970A w

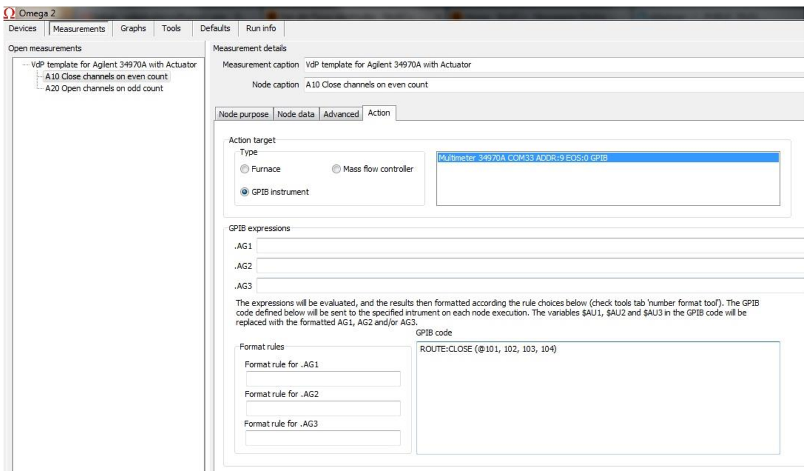



20 CHANNEL ACTUATOR / GENERAL PURPOSE


20 CHANNEL ACTUATOR / GENERAL PURPOSE SWITCH


NNEL ACTUATOR / GENERAL PURPOSE


20 CHANNEL ACTUATOR / GENERAL PURPOSE


Omega 2

Devices | Measurements | Graphs | Tools | Defaults | Run info

Open measurements
- VdP template for Agilent 34970A with Actuator
  - A10 Close channels on even count
  - A20 Open channels on odd count

Measurement details

Measurement caption: VdP template for Agilent 34970A with Actuator

Node caption: A10 Close channels on even count

Node purpose | Node data | Advanced | Action

Action target

Type: ( ) Furnace ( ) Mass flow controller [✓ GPIB instrument]

Multimeter 34970A COM33 ADDR:9 EOS:0 GPIB

GPIB expressions

.AG1

.AG2

.AG3

The expressions will be evaluated, and the results then formatted according the rule choices below (check tools tab 'number format tool'). The GPIB code defined below will be sent to the specified intrument on each node execution. The variables $AU1, $AU2 and $AU3 in the GPIB code will be replaced with the formatted AG1, AG2 and/or AG3.

Format rules

Format rule for .AG1

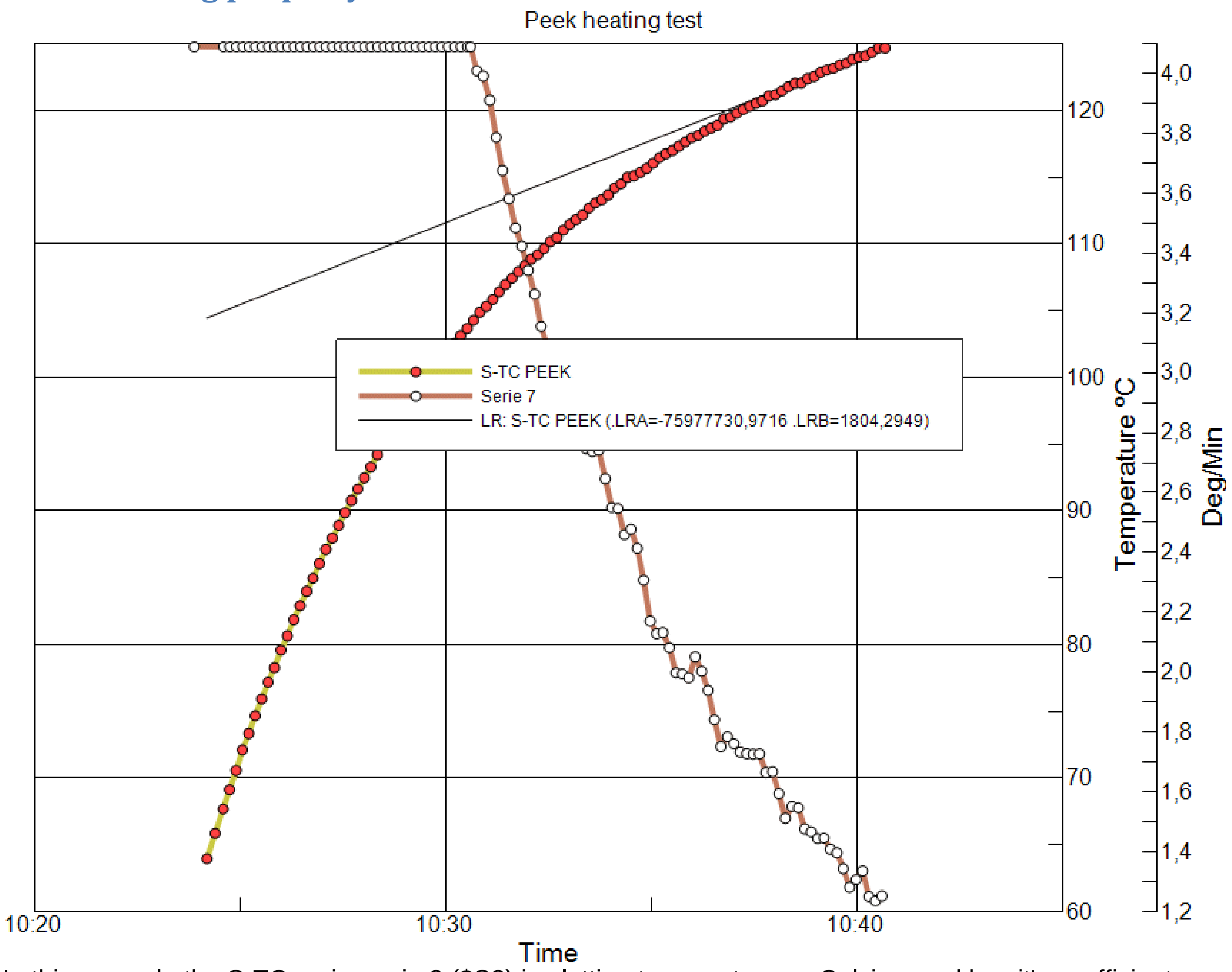
Format rule for .AG2

Format rule for .AG3

GPIB code

```
ROUTE:CLOSE (@101, 102, 103, 104)
```

9.13 Plotting property of a series



In this example the S-TC serie, serie 2 ($S2) is plotting temperature as Celcius, and has it's coefficient enabled to be calculated from 10 last points. The Serie 7, has Y-expression: $S2.LRB/24/60 which will plot the angle, the .LRB of the serie 2. Since this serie has full days as X-component, we divide by 24 hours and 60 minutes to get the coefficient angle plotted as degrees perminute.

Note, that the Serie related properties are live data, calculated only for the latest point of a serie. The flat part of the Serie 7 in the beginning is a display of this: all points that existed before the Serie 7 was enabled, were plotted with the value of the latest point. This can be avoided by configuring the serie related properties to be plotted before the measurement is started. Hitting a redraw button will draw a flat line of the latest value.

9.14 FAQ + Frequenty asked question

9.15 How do I add DC bias to impedance spectrometer?

Current version of Omega includes property boxes to define bias for measurements with impedance spectrometer on the measurement node tab. Information below is how to do it manually and for older versions of the software.

To add bias you must utilize the "GPIB commands before" and "GPIB commands after" boxes in the node that makes the measurement.

For Solartron 1260
To enable 0.5 V bias, add the line below goes to the box called "GPIB commands before"
VB 0.5
#SLEEP 1000

To disable to bias, add the line below goes to the box called "GPIB commands after"
VB 0

For Novocontrol
To enable 0.5 V bias, add the line below goes to the box called "GPIB commands before"
DCE=1
DCV=0.5
#SLEEP 1000

To disable to bias, add the line below goes to the box called "GPIB commands after"
DCV=0
DCE=0

The #SLEEP 1000 will wait 1000 milliseconds before continuing with the measurement, allowing the voltage to settle.

9.16 Nothing happens, I cannot measure

- A valid USB license must be present
- Devices are validated?
- "Assigned device setup" ID matches the "Required device setup" ID?
- Measurement file is active?
- Node(s) is/are active?

Read about devices setup and measurement properties

10 Expressions

Users can freely define expressions anywhere in the program where the field is recognized as expression. The expressions can be both static or dynamic.

10.1 Node properties

Acronym	Short for	Explanation
.FAM	First active minutes	Each node remembers when it first became active and was executed. $N1.FAM returns the elapsed time in minutes from the first activation of node $N1.
.LAM	Last active minutes	$N1.LAM returns the elapsed time in minutes from the the moment when the node $N1 was last active and executed.

11 Using Omega with Zurich Instruments Impedance spectrometer MFIA

Many modern impedance analyzers do not offer the ability to connect and command them through the GPIB port and standard, but use networking or USB connection with their own interface. This approach limits the universality of such instruments, as in order to control such instrument from 3rd party software the programmer must work and add abstraction layer (unique for each manufacturer and instrument) instead of using a common standard. In many cases this extra work is not worth the trouble but in the case of the aforementioned instrument we decided to add it to Omega software. Not all Zurich instruments MFIAs, but only those purchased from NORECS.

Due to the nature of the interface some extra steps are necessary to be able to make the instrument usable in Omega. Or in other words, there is no automatic device finding feature in the application programming interface provided by Zurich instruments, so the device cannot be found automatically and needs to be entered by hand into the device configuration file.

1. Before starting with Omega, make sure to install the LabOne software package from Zurich instruments, and make sure your instrument is discoverable and you can use it from LabOne software. Using the 'MF Device Finder' open your instrument. Write down the instrument serial such as mf-dev4625 and the port such as 8004.

2. Go to where the LabOne was installed, such as C:\Program Files\Zurich Instruments\LabOne\API\C\lib, and copy ziAPI-win32.dll and ziAPI-win64.dll files into the folder where you have the Omega executable.

3. In Omega Device Setups folder open a device configuration file (.odc) of your choice, or create a new one using 'save device setup' feature in Omega. Add the following text to the configuration file.

```
API
	ZHINST
		Imp.Spectrometer mf-dev4625 8004
```

4. The API is a new line, ZHINST is a new line but with one tab stroke before it, and the last line starts with two tab strokes, then text "Imp.Spectrometer", space, your instrument serial in the format mentioned i.e. mf-devxxxx where xxxx is the instrument serial, space, and the port the instrument is on. These details can be found using the LabOne software and/or the ZHInst device finder software.In Omega open this device configuration file and click validate, and Omega will check if the software can connect to the instrument.