Differential Impedance Spectrometer

for electrochemical and electrophysiological analysis of fluids and organic tissues

CYBRES®
beyond technology

HANDBOOK and USER MANUAL

edited by Dr. Serge Kernbach

1 Foreword

ATTENTION. Read carefully these instructions. Follow closely the recommendations for measuring weak changes of electrochemical and physiological parameters.

ATTENTION. Make sure that the 5-/16-/26- pin connector on the EIS devices is used properly: the connector should be first fully inserted and then screwed.

ATTENTION. The spectrometer contains the CR2032 lithium battery.

ATTENTION. The device has one USB port for data exchange and power supply. It is necessary to use the USB 3.0 active hub for powering the device. Connect the USB 3.0 hub to the low-noise power source. This USB power supply should not be used by any other device.

ATTENTION. The USB ground loops can distort functionality of the device. To avoid ground loops it is strongly recommended to use commercially available USB-to-USB isolators.

ATTENTION. No calibration is required for differential measurements. For absolute measurements (e.g. conductivity, temperature, etc.) the device should be calibrated regularly. It is recommended to contact the manufacturer for a calibration.

The device is made in accordance with the following European Directives: 2006/95 / EG (Low Voltage), 2004/108/EG (EMC), 2011/65/EU (Directive on the use of hazardous substances in electrical and electronic equipment, 2009/125/EG (eco-design/energy-using products). The device is manufactured in accordance with the latest technological developments. However, there are residual risks. To avoid danger observe the safety instructions. The manufacturer is not liable for damages caused by non-compliance with safety instructions. Children should not play with the device. When leaving the device for a long time, disconnect it from the power supply.

The device does NOT fail under the Directive 2014/32/EU on measuring instruments as MI-001 'WATER METERS' (An instrument designed to measure, memorise and display the volume at metering conditions of water passing through the measurement transducer); MI-005 'MEASURING SYSTEMS FOR THE CONTINUOUS AND DYNAMIC MEASUREMENT OF QUANTITIES OF LIQUIDS OTHER THAN WATER' (An instrument designed to measure continuously, memorise and display the quantity at metering conditions of liquid flowing through the measurement transducer in a closed, fully charged conduit); MI-003 'ACTIVE ELECTRICAL ENERGY METERS'

VERSIONS. This manual v.2.5.2 is based on the firmware v.1190.x and the client program v.1.4.x.

2 Terminology and used notations

EIS – Electrochemical Impedance Spectroscopy.

MU – Measurement Unit. Series of precision measurement systems developed by CYBRES GmbH (Cybertronica Research). The notations MU EIS, CYBRES EIS or CYBRES MU EIS mean the same notion.

FRA – Frequency Response Analysis, the method of signal analysis.

DFT – Discrete Fourier Transformation, the method of signal anaysis.

ADC – Analog to Digital Convertor, the hardware component used to convert analog signals into digital form.

DAC – Digital to Analog Convertor, the hardware component used to convert digital signals into analog form.

DDS – Direct Digital Synthesis, the hardware approach used to generate excitation signals.

RMS – Root Mean Square, also known as the quadratic mean, is defined as the square root of the arithmetic mean of the squares of a set of numbers.

PCB – Printed Circuit Board, it contains electronic components on a non-conductive substrate.

USB – Universal Serial Bus, the interface between computers and electronic devices.

PID – Proportional-Integral-Derivative controller, a hardware - software control system used to keep a predetermined temperature in thermostats.

DA – Detectors-Actuators module, see Section 10.

DI – Deficit Irrigation.

V_V – the excitation signal that drives electrochemical test system.

V_I – the response signal based on the flowing current I thought the test system.

f – the frequency on which the analysis is performed.

$Z(f)$ – impedance of the test system for a harmonic signal of frequency f.

$Re^{FRA}(V_I)$ – real part of the response signal obtained by the FRA analysis.

$Im^{FRA}(V_I)$ – imaginary part of the response signal obtained by the FRA analysis.

t – temperature.

Contents

3 General description

3.1 The device

The CYBRES MU (Measurement Unit) is a bio-hybrid interface device for real-time interactions with various biological, microbiological and fluid systems. It includes:

- differential Electrochemical Impedance Spectrometer (EIS);
- differential analyzer of bio-potentials;
- interface for analog and digital phyto-, bio- and environmental electrodes/sensors.

The MU performs high-resolution differential measurements of ionic properties in liquid or organic samples, enabling electrochemical and physiological analysis of plants, microorganisms, tissues and solutions. This device is designed for precision agriculture (including hydroponics and indoor farms), phytosensing, biosensing, and non-chemical water treatment. Its high sensitivity allows the detection of weak electrochemical and physiological changes caused by environmental and technological factors. The MU conducts real-time data processing and controls actuators like lights, pumps, valves, or relays. It supports instrumentation tasks with autonomous bio-/phyto-/fluid-sensing for complex feedback-driven adaptive scenarios.

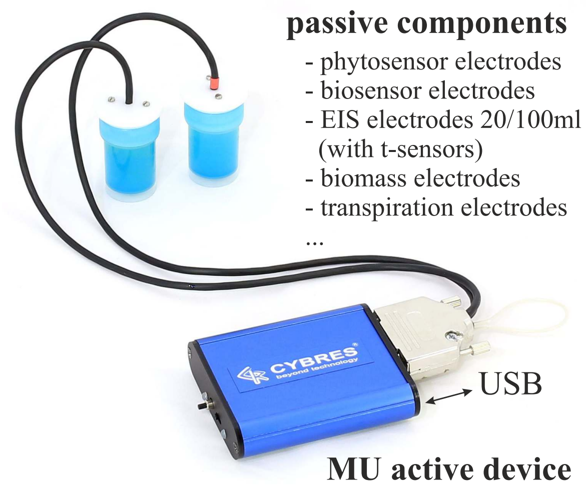


Figure 1: MU system with active and passive components, example with EIS electrodes for electrochemical analysis of fluids.

The structure of MU system is shown in Figure 1. It consists of an active electronic module and replaceable passive components for phytosensing, biosensing and EIS applications. The system features:

- real-time operating system;

- internal embedded sensors;
- interfaces to external analog and digital sensors;
- the thermostatic system with two channel digital PID controller for fluid samples and electronics;
- analytic tools for real-time data analysis.

USB 2.0 is used for data transfer to the host computer and for powering the device. All data are recorded in real time with time stamps and are stored on PC or in the on-board flash memory.

The system measures up to 45 physical parameters (45 physical data channels), and calculates in real time up to 35 numerical/statistical parameters (35 synthetic data channels), which are programmable by users.

ATTENTION. The MU operates in three main modes: 1) **phytosensor** with specific electrodes; 2) **electrochemical impedance spectrometer** for analysis of aqueous solutions; 3) **biosensor** with specific electrodes. Additional sensors in each mode can be turned on/off.

3.2 Applications

3.2.1 Phytosensing and indoor farming

Three main phytosensing applications[1] are shown in Figure 2.

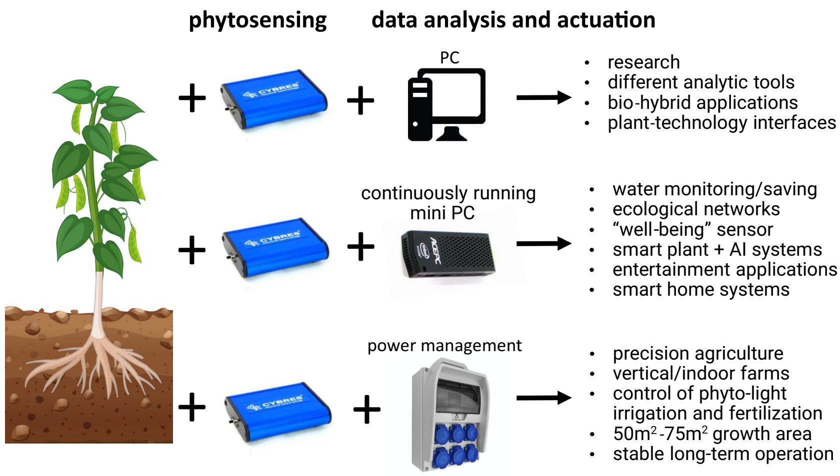
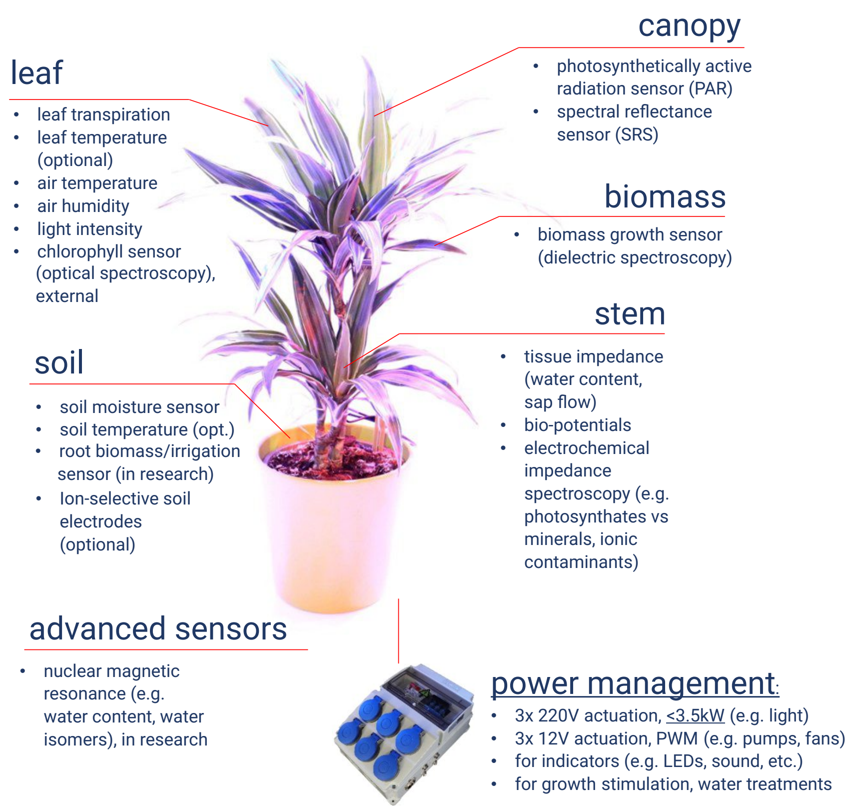


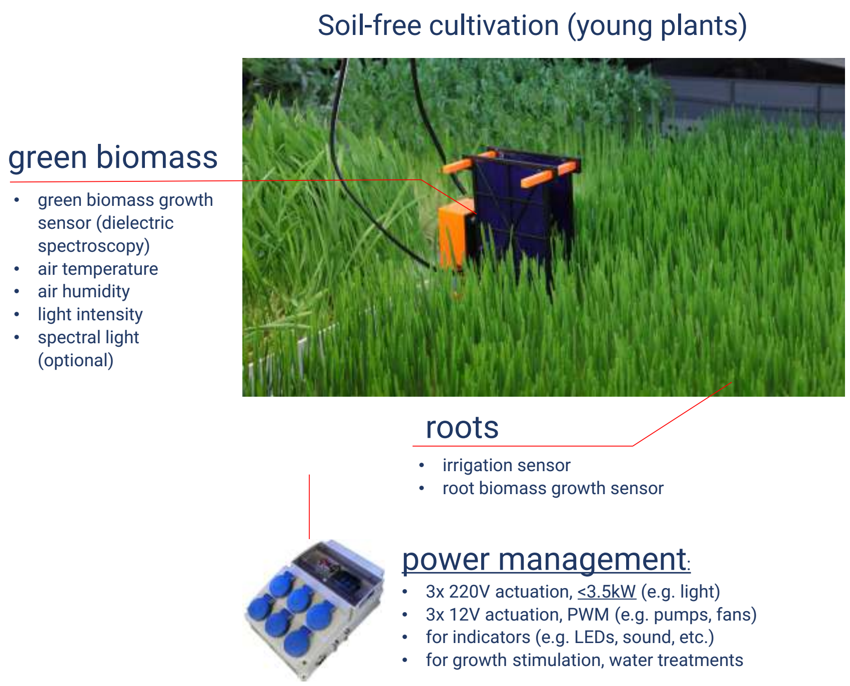
Figure 2: Three main phytosensing applications.

They depends on connected computational and power management systems and allow *in-vivo*, *in-situ* or *in-vitro* electrophysiological measurements of plants or organic tissue, e.g. monitoring

[1] see Kernbach, Biofeedback-Based Closed-Loop Phytoactuation in Vertical Farming and Controlled-Environment Agriculture. Biomimetics 2024, 9, 640, doi: https://doi.org/10.3390/biomimetics9100640



(a)



(b)

Figure 3: Overview of different phytosensors and phytoactuators supported by the MU system for soil-based and soil-free cultivation.

plant physiology and electrophysiology, analysis of bio-potentials and tissue impedances, sap flow, transpiration, soil moisture and other parameters. Several supported phytosensors and phytoactuators for soil-based and soil-free cultivation are shown in Figure 3. The MU can be used as a standalone device for control of hydroponic systems, vertical and indoor farms.

3.2.2 Biofeedback systems and AI phenotyping

The MU represents a tool for design and implementation of bio-hybrid or AI (Artificial Intelligence)-based systems with plants. Such systems can be used to control different agricultural actu-

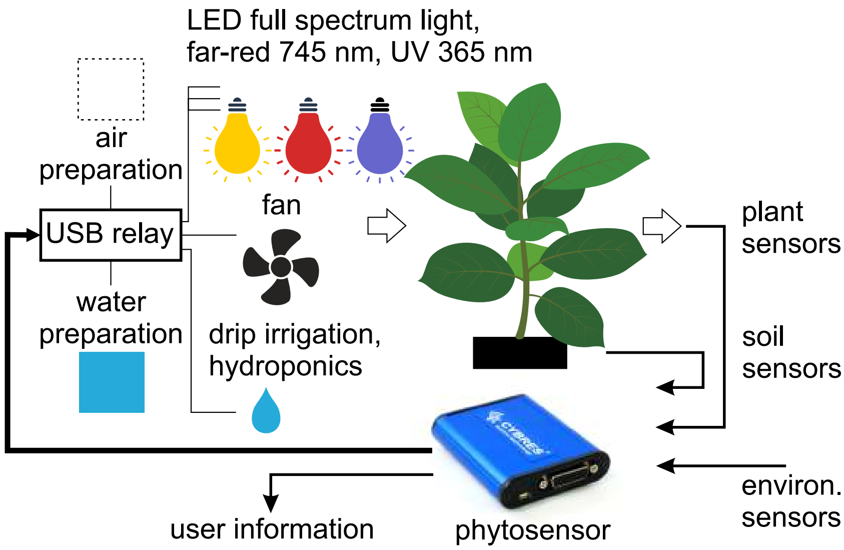


Figure 4: Using the phytosensor in a biofeedback-based system.

ators with biofeedback-based (based on sensor data) protocols, see Figure 4. Water-/air-preparation, fertilization, ozonation/AOP (advanced oxidation process), aeration/OMB (O_2 microbubbles), magnetic treatment and other approaches, can also use this scheme with a biofeedback from plants. Access to physiological and environmental parameters enables using this system in machine learning techniques; phytoactuation provides a possibility to integrate a bio-exploration in AI schemes, e.g. for fast AI phenotyping[2].

3.2.3 Using plants as biosensors

Environmental pollutants, various stressors and stimuli trigger physiological reactions in plant organisms. Biological detection of such factors can be conducted by measuring electrochemical, bio-electric, hydrodynamic and other responses of plants in-situ. Biosensing ap-

[2] see Buss et al, Stimulus classification with electrical potential and impedance of living plants, Bioinspir. Biomim. 18 025003, 2023, doi: https://doi.org/10.1088/1748-3190/acbad2

plications, e.g. for environmental monitoring[3] have great practical relevance due to high sensitivity, simplicity and reliability of sensors, enabling outdoor and field applications. Tomato, tobacco, dracaena or any other plant can be used for biosensing purposes, motivated by a sensitivity of these plants for a particular stressor, see Figure 5.

Figure 5: Examples of biosensors (phytosensor + plant sensitive to a particular stressor or pollutant), here – tomatoes are measuring environmental ozone.

3.2.4 Water monitoring/saving technologies

According to the United Nations World Water Report, agriculture accounts for almost 70% of global water consumption, and 40% of this is wasted due to inadequate irrigation systems, evaporation, and poor water management. In 2011, the global water footprint of agricultural production was 8,362 km^3/year. To meet the growing demand for food and biofuels, agricultural production will need to

[3] see Kernbach, In-situ biological ozone detection by measuring electrochemical impedances of plant tissues, doi: https://doi.org/10.48550/arXiv.2411.16321,2025

increase by almost 50% by 2050 compared to 2012. This is expected to require even more water. Global water demand is expected to increase by 20–30% between 2010 and 2050.

In addition to improving overall water management systems, water saving technologies can also be implemented at the phytosensing level through various deficit irrigation (DI) strategies. DI is a practice where a crop is irrigated with water below the full requirement for optimal plant growth. For example, irrigating crops with -50% water increased water productivity by +24% over full irrigation with yield reduction by -38% (compromise strategy: water vs yield). DI has emerged as a viable approach to increasing agricultural water productivity and to save a significant amounts of water. However this cultivation method deliberately induces drought stress at plant development. By using sensors that measure the physiological state of plants in real time and in vivo, DI strategies can be implemented without damaging plants with better water vs yield ratio. Beside DI, specific sensors, such as biomass

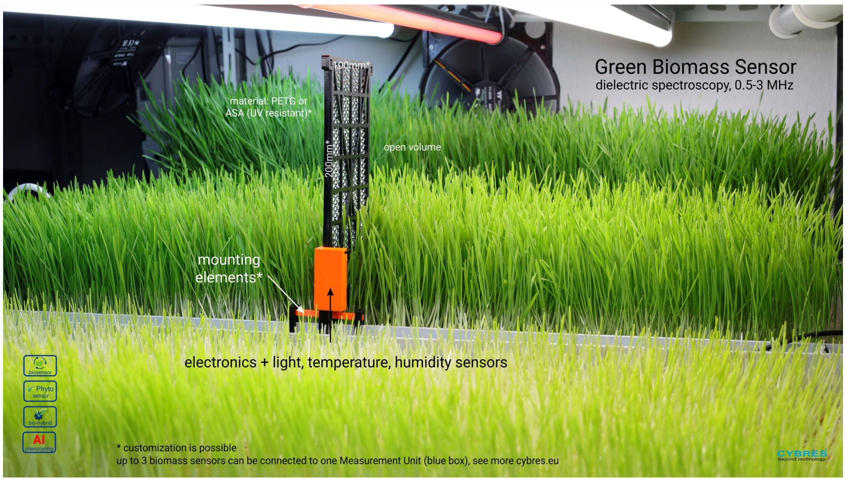


Figure 6: Green biomass sensor (open volume measurement) in indoor farming environment for monitoring of water distribution by root and green biomass, and optimization of water flow.

and irrigation sensors, enable monitoring of water distribution and consumption by root and green biomass, and optimization of energy and water flow in the cultivation facility, see Figure 6.

3.2.5 Microbiology: fermentation and sedimentation

Microbiological applications[4] include bio-sensors that measure fermentation, sedimentation, gas production (or degassing), metabolic production or any other processes that change concentration and mobility of ions in the solution (e.g. control of fermentation

[4] see Kernbach et al, The biosensor based on electrochemical dynamics of fermentation in yeast Saccharomyces cerevisiae, Environmental Research, 213, 2022, 113535, doi: https://doi.org/10.1016/j.envres.2022.113535

activity of yeast), see Figure 7. The MU system is designed for

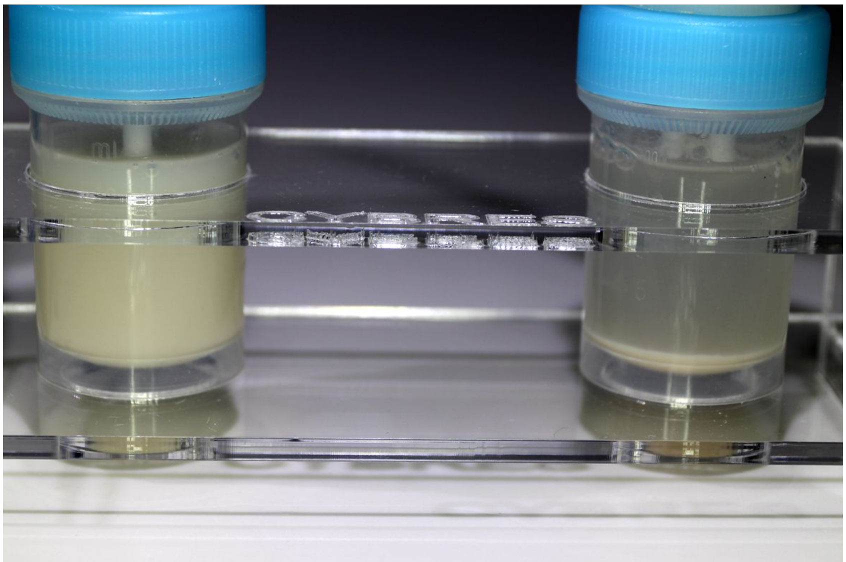


Figure 7: Example of fermentation and sedimentation processed analysed with electrochemical impedance spectroscopy.

long-term monitoring of biological samples, e.g. for quality control purposes or for the analysis of biochemical reactions.

3.2.6 Non-chemical treatments of aqueous solutions

Electrochemical applications[5] are precise industrial fluid measurements and differential fluid meters in research and laboratory usage, detectors of non-chemical treatments by analysing electrochemical changes in fluids (e.g. water filtering, decalcination, removing of organic contaminants), optical or magnetic excitation of

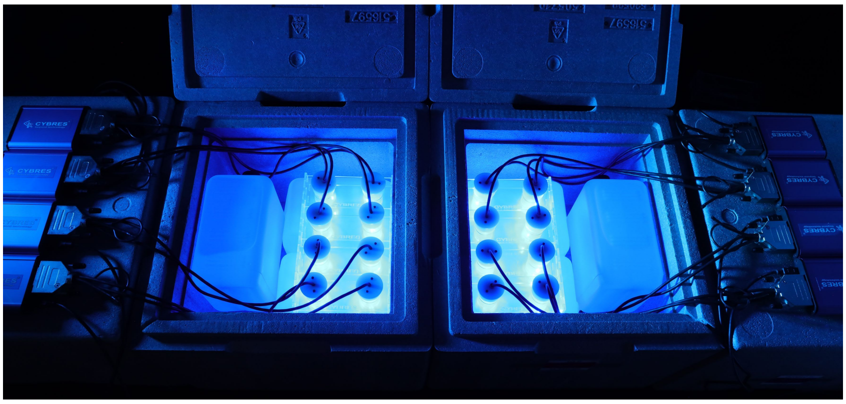


Figure 8: MU systems (MU3 units + open EIS electrodes) used for electrochemical analysis of fluid samples.

[5] see Kernbach, Electrochemical Characterization of Ionic Dynamics Resulting from Spin Conversion of Water Isomers, 2022 J. Electrochem. Soc. 169 067504, doi: https://doi.org/10.1149/1945-7111/ac6f8a

fluids, see Figure 8. These measurements can also be used for analysis of weak interactions in aqueous systems, in particular caused by quantum phenomena in macroscopic systems (e.g. the proton tunneling effect or change of electrochemical reactivity by para- and ortho- isomers of water. The device allows statistically significant measurements of these effects with the standard EIS method.

3.3 Main measurement modes

Depending on the electrodes used, the MU device can perform the measurements shown in Table 1.

Table 1: Main measurements (depending on electrodes/sensors).

N	Measurements	Applications
1	Electrochemical Impedance Spectroscopy (EIS)	ionic analysis, frequency-response analysis, detection of photosynthates vs minerals or ionic contaminates
2	Dielectric Spectroscopy (DS)	green/root biomass growth (fixed frequency DS)
3	Excitation Spectroscopy (ES)	excitation-response analysis, distortion analysis, measuring chlorophyll
4	Dynamics of Impedances	water content, sap flow, metabolic products, analysis of temporal ionic dynamics
5	Dynamics of Biopotentials	plant signalling system (action potential, systemic potential), 4x-electrode EIS, analysis of temporal dynamics
6	Physiological parameters	measuring plant physiology
7	Environmental parameters	environmental measurements

For these purposes, the system has embedded analysis tools, which are shown in Table 2. Considering these tables, the device can operate in 11 different measurement modes, see Table 3, their selection is controlled by settings of the 'DDS mode', 'configuration', checkboxes 'regression analysis', 'excitation' and DA/Python scripts.

ATTENTION. Measurement modes of MU devices depends on samples (plants, biological tissues, fluids, microorganisms), used sensors/electrodes and analytic tools. The measurement methodology is different for pre-processed samples (i.e. experiments before measurements), processing-during-measurements and post-processing of measured data, see corresponding application notes.

3.4 MU-family of devices

The MU device is produced in three main versions:

1. The **modular/reconfigurable version** that includes
   - the main measurement unit MU3 or MU3T, see Figure 9(c) and 9(d);
   - electrodes for measuring plant (electro-)physiology and phytosensing, see Figure 20;

Table 2: Embedded analytic tools.

N	Analytic tools	Applications in real time
1	Regression analysis	analysis of weak signals
2	Statistical analysis	computing statistical parameters
3	Correlation analysis	conducting correlation analysis
4	DA-processors	embedded script for data processing
5	Python scripts	Python script for data processing
6	External tools	analysis by e.g. MS Excel, SPSS and others

Table 3: Main measurement modes of MU devices.

N	Application	DDS mode	Configuration
1	signal distortion analysis, fast statistical analysis	signal scope	EIS
2	analysis of temporal EIS/temperature dynamics, 'experiment during measurement' mode, biosensor applications, 3D time-frequency analysis	continuous modes	EIS, biosensor
3	**regression enabled**, the highest resolution of temporal dynamics	continuous modes	EIS
4	**excitation enabled**, analysis of pre-treated fluidic samples, characterization of non-chemical treatment	continuous modes	EIS
5	**EIS statistics enabled**, statistical analysis of EIS noise, see App.Note 24, pre-treated or on-line treated fluidic samples, characterization of non-chemical treatment	continuous modes	EIS
6	**timed regression/MIND enabled**, specific statistical/regression analysis, see App.Note 26, on-line treatment of fluidic samples, characterization and weak impact factors	continuous modes	EIS
7	frequency analysis, impedance spectroscopy, differential analysis of samples, FRA profiles	frequency modes	EIS
8	environmental measurements	off	EIS
9	electrophysiological measurements of tissues, physiological measurements (with corresponding sensors), differential potential analysis	off	phytosensor
10	electrochemical interface to bio-samples, biological tissues and plants	continuous modes	phytosensor, biosensor
11	correlation analysis	continuous modes	EIS, phyto-bio-sensor

- EIS electrodes with integrated temperature sensors for fluid measurements, see Figure 18(a);
- electrodes for biosensing purposes with additional sensors;
- additional modules, actuators, power management systems;

2. The **embedded system for differential EIS analysis** with thermostabilization of fluidic samples and optical excitation, see Figure 9(a);
3. The **embedded biosensor system** for measuring microbiological samples based on fermentation, sedimentation or ionic processes (with thermostabilization of samples and op-

tical excitation), see Figure 9(b).

Additionally, the MU-family includes the MU actuation board and power management system (that connect different high power real-world actuators, see Figure 9(f)), and EHM-C board (for generating electric and magnetic fields). All devices in the MU family are compatible with each other.

Table 4: Device versions with hardware and software options.

N	Device versions	Hardware	Software
1	(**reconfigurable**) phytosensor basic	1 channel biopotentials & impedance	enabled
2	(**reconfigurable**) phytosensor advanced	+ TransAmb sensor (leaf transpiration) + 2 channels biopotentials & impedance	enabled
3	(**reconfigurable**) phytosensor full	+ sup flow sensor	enabled
4	(**reconfigurable**) EIS	+ 2x EIS open electrodes, + fluidic/environ. t sensors, + excitation spectroscopy	activation code*
5	(**embedded**) EIS	different device with thermostat, + RGB/IR excitation spectroscopy	enabled
6	(**embedded & reconfigurable**) biosensor	+ fermentation module, + RGB/IR excitation spectroscopy	enabled

*transition from phytosensor to EIS/Biosensor requires software activation

The measuring unit (MU) – MU20, MU31, MU32, MU33, MU34, MU34T (with enhanced termostabilization, see Figure 9(c,d)) and MU40 – is based on the 32-bit ARM processor with a real-time operating system (the MU RTOS). It has accurate analog-to-digital and digital-to-analog converters, an internal non-volatile memory, real time clock, low-pass filters and additional sensors. The measuring unit is characterized by a low noise level. Note that the embedded EIS spectrometer uses round 5-pin micro connectors, MU33 uses D-Sub 16-pin connectors and MU34, MU34T – D-Sub 26-pin connectors.

3.5 Features

- main processor: ARM cortex M3 MPU, 80 MHz
- hardware support of analysis: PSoC system
- non-volatile (flash) memory: 512 Mb
- level of noise[1]: $< 1\mu$ V
- sampling frequency: (12-24 bits) up to 1 Msps
- accuracy of temperature stabilization[2]: 0.02C
- temperature resolution[3]: up to 0.001C
- F_{min}, min. frequency: 8 Hz

(a) (b)

(c) (d) (e)

(f)

Figure 9: **(a)** the embedded version of MU EIS differential impedance spectrometer; (1) – measurement cell with two channels (closed by lids on the image), (2) – connector for two electrodes; (3) – hull with RGB LEDs, electronics and additional sensors; (4) – electrode, the channel 1 (with red mark); (5) – electrode, the channel 2; **(b)** the embedded version of biosensor (for comparison – 15ml and 100ml containers); **(c,d)** the measuring unit (MU) MU3 and MU3T (with enhanced termostabilization); **(e)** the MU for differential pH measurements; **(f)** the power management module with 6 switchable outputs.

- F_{max}, max. recommended frequency for EIS-tissue sensing-manual[4]: 0.1MHz-0.3MHz-0.65MHz
- EIS: number of frequency bands: 3 (8-450Hz, 100-10.000Hz, 450Hz-F_{max})
- EIS: ranges of excitation voltage: 0.001-0.01V AC, 0.01-0.1V AC, 0.1-1V AC (max. amplitude ±1V)
- EIS: amplification factors: 50, 500, 5000, 50000
- EIS measurement modes: 1) impedance spectrometer; 2) signal scope; 3) continuous measurements at a constant f; 4) continuous measurements at variable f; 5) Frequency Response Profile (FRP) at a fixed set of frequencies; 6) continuous FRP
- EIS analysis: Amplitudes, FRA Phase, RMS Magnitude, Correlation, Electrochemical stability in time/frequency/time-frequency domains, Statistical analysis, Excitation analysis
- EIS: self-calibration with integrated calibration resistors: 4.99 kOm, 499 Om, 0.1% 25 ppm
- external analytic tool: regression/correlation/statistical analysis
- high-impedance differential potential input: -0.5:+0.5V, 200pA (2x instrumental OA INA333)
- analog input: 0-1V (max. resolution 64 nV); 0-2V (max. resolution 128nV)
- electrode system: two (standard configuration) and four electrode measurements
- conductivity measurement range[5]/conductivity of used water: 0.6μS/cm–200 mS/cm
- support external Gnuplot and Python scripts (Python server is implemented)
- duration of long-term measurements: on the level of weeks
- embedded sensors: 3D accelerometer/magnetometer (optional), two internal temp. sensors, air pressure sensor, RF power meter (450Mhz-2.5Ghz)
- external sensors: analog or digital phyto-, bio- and environmental sensors
- basic accuracy class[6]: 0.5%, 0.1%
- output: 5 MOSFETs (linear and PWM modes) e.g. to control external SSRs or RGB LEDs
- supported buses: UART (front connector, MU34), I2C (back connector)

- typical current consumption[6] at 5V: ≈0.3A
- powering[7]: external active USB3.0 hub
- data interface: USB 2.0

[1] Test conditions: battery power supply, no galvanic isolation on power, all interfaces off, MCU clock 6Mhz, low level of environmental EM noise.

[2] This varies inside a volume of thermo-insulating containers.

[3] This resolution is primarily defined by electronic noise, data are shown for the LM35 precision sensor (10mV/C with 64 nV ADC resolution).

[4] The maximal and minimal frequency depends on the selected analytical tool, the frequency resolution and the requirement on a minimal number of samples in the sweep. The firmware provides different options based on other selected values.

[5] Conductivity measurements (and conductivity calibration) should be performed at one fixed frequency.

[6] This depends on requirements, in-situ calibration capabilities and reconfiguration options, ask info@cybertronica.de.com for more information.

[7] It depends on the MCU clock frequency, number of connected sensors and the thermostat's operating mode, see the Section 5.3.

3.6 Sensors

MU has analog (ADC-DAC) and digital (I2C and UART) interfaces for embedded and external sensors. Depending on applications, all external sensors are grouped into phyto-electrodes, bio-electrodes and EIS electrodes. Different versions of electrodes contain different combinations of sensors. For instance, in the phytosensor mode, the system measures physiological and environmental parameters, shown in Table 5. The following sections provide brief descriptions for separate sensors as well as their combinations in electrodes.

3.6.1 Embedded sensors

The device is equipped with several internal sensors, see Table 5:

- temperature of the thermostat, CPU and electronic module;
- the control of 4.2V power supply (used to monitor interferences on the power line);
- control of the thermostat (used to control the energy level supplied to the measuring part of the device)
- 3D magnetometer (used to control the static magnetic field during experiments) and 3D accelerometer (used to control mechanical impacts, optional)
- 450Mhz-2.5Ghz RF power meter;
- air pressure sensor.

Data from these sensors are available any time in the section 'plot', several sensors (e.g. 4.2V power supply) are available only with specific electrodes.

Table 5: Overview of measured phyto-parameters and phyto-actuation (depending on used electrodes/sensors).

parameters	description
	phytosensing
tissue impedance	differential, 4x Ag99 electrodes, 0.01-1V excitation
electrochemical spectroscopy	time-frequency EIS, fast EIS for *in-situ* sap analysis
biopotentials	differential, 4x Ag99 electrodes, input impedance 10^{-15}Ohm, input bias current ±70pA
leaf traspiration	differential air-humidity-based method, CYBRES
leaf temperature	precision LM35 sensor
thermal sap flow	heat-balance and heat-pulse methods, 3x *t*-sensing, PID stabilized, CYBRES
electrochemical sap flow	4x electrode method, CYBRES
wet green biomass	dielectric spectroscopy, 0.5-1 MHz, CYBRES
root biomass and irrigation	dielectric spectroscopy, 0.5-1 MHz, CYBRES
chlorophyll content	excitation spectroscopy, fluorometry 430nm
	environmental sensing
light, humidity, temperature	APDS-9008-020, HIH-5031-001, LM35CA
EM emission	450Mhz-2.5Ghz RF power meter, MAX2204
magnetometer	3-axis, LIS3MDL
air pressure	internal sensor, BMP280
soil moisture, temperature	capacitive-based sensor, CYBRES
CO_2, PM1-2.5-10, O_3	SCD4x, accuracy ±(40ppm+5%); SPS30, accuracy 10%, CENSIRION
I2C sensors	different digital external sensors, e.g. PAR sensor
water sensors	e.g. conductivity, pH, dissolved oxygen, etc.
	(phyto-)actuation
220V/110V	ON-OFF, 4 channels, up to 3kW (limited by power network), light/spectral light/irrigation
12V	ON-OFF, PWM, 2 channels, up to 10A, irrigation/aeration/fertilization/disinfection O_3, H_2O_2

3.6.2 External temperature sensor

Measuring module supports an external high resolution temperature sensor (Texas Instruments LM35CA), connected to the 26/16/5-pin connector. It has a typical absolute accuracy of ±0.2°C, typical nonlinearity of ±0.15°C and the conversion factor V/°C of +10mV/°C (see more the Datasheet 'LM35 Precision Centigrade Temperature Sensors'). With the ADC resolution of 22 bit, this sensor provides a resolution of relative temperature measurements < 0.001°C. The sensor is useful for monitoring environmental temperature and is included into different sensor sets. If using this sensor in outdoor conditions, protect it from direct sun light!

3.6.3 Phyto: transpiration, environment

Leaf transpiration sensor with several environmental sensors are combined in one sensor (the transAmb stick), shown in Figure 10. Environmental sensors measure the air humidity (the sensor Honeywell HIH-5031-001), light intensity (the sensor Broadcom/Avago APDS-9008-020) and air temperature (with different temperature

Figure 10: The transAbm stick – Leaf transpiration sensor with environmental sensors (air humidity, light intensity, air temperature).

sensors: Texas Instruments LMT70AYFQR or LM35CA). Overview of main parameters is shown in Table 6.

With 22 bit ADC conversion these sensors provide a theoretical resolution of measured parameters as < 0.001°C, < 0.001%, < 0.001 Lux. Data from these sensors are available in the section 'plot', 'plot 1x: external sensors'. The transAmb stick can be used as a high-resolution environmental data logger without transpiration sensor and the leaf clip.

Table 6: Parameters of the transAmb sensor stick.

sampling resolution	20-24 bit
min. resolution of analog input	±64nV
conversion factor V/t of LM35CA sensor	10mV/°C
conversion factor V/t of LMT70AYFQR sensor	5.19mV/°C
conversion factor V/% of HIH-5031-001	≈23.5mV/%
measurement range of light sensor	0-1000 Lux
size of data logger sensor panel	100x10x5 mm

3.6.4 Phyto: EIS and biopotentials

For EIS and tissue impedance measurements, the system provides needle electrodes, see Figure 11(a), and clip electrodes, see Figure 11(b). Biopotentials can be measured only with the needle electrodes. Ag-99 needle electrodes are primarily used for penetration of soft tissues, while clip electrodes are used for hard and woody

(a)

(b)

Figure 11: Differential EIS with **(a)** needle electrodes and **(b)** clip electrodes.

tissues. The distance between EIS electrodes is selected experimentally so that the RMS impedance is below 100 kOhm, typically at the distance of 10 mm. The penetration depth is about 2-3mm to have a stable mechanical contact and reach phloem and xylem tissue. Electrodes can be inserted directly into the leaf or completely penetrate the stem. Parameters of these electrodes are shown in Table 7. After insertion, electrodes can trigger a tissue reaction (lignification of the puncture hole) and an increase in impedance. The electrodes remain functional under such conditions and do not require replacement (tested for up to several months of continuous measurement). Note that EIS electrodes are used for measuring different parameters such as water content, sap flow or photosynthate vs minerals ratio.

Table 7: Parameters of EIS/biopotentials electrodes.

material of needle electrodes	Ag99 (silver)
material of clip electrodes	Cu3Zn2 (brass) or X6CrNiMoTi17-12-2 (stainless steel V4A)
channel 1 wires	white, yellow (labelled)
channel 2 wires	brown, green (labelled)
distance between EIS electrodes	typically 10 mm
distance between biopotential electrodes	maximal distance allowed by wires

ATTENTION. EIS electrodes should be placed approximately at 10 mm from each other, and biopotential electrodes should be placed at a maximum distance from each other. Impedance of both channels should demonstrate similar values (typically ±5kOhm), large differences between the channels lead to a suboptimal measurement range and greater noise.

3.6.5 Phyto: sap flow, water content

The heat-based sap flow sensor is shown in Figure 12. It has two upflow and downflow temperature sensors and a thermo-stabilized heater between them. The sensor supports both the heat-balance and heat-impulse measurements as state-of-the-art approaches for sap flow measurements. The heater temperature is measured by an independent sensor, the PID controller monitors the heating dynamics with high accuracy. Embedded electronics forms all necessary signals. The heat-based sensor represents an invasive mea-

Figure 12: Heat-based sap flow sensor with two upflow and downflow temperature sensors and a heater between them.

surement technique that causes tissue damage if used over a long period of time. This damage is not related to overheating but represents a systemic response upon long-term energy influx to the vascular tissue. Therefore, the heat-based sensor is not recommended to use for phytomonitoring purposes (however it can be produced on request) and is replaced by two-point electrochemical sensor for measuring water content and sap flow, see Figure 13.

Electrodes in two-point EIS are inserted in the upper and lower parts of stem tissues and use the following hydrodynamic model. Effective measurement range with two needles (in one sensor position) is limited by applied excitation potential, electric field and dielectric properties of tissues, and can be represented as a volume V containing fluids, see Figure 13. EIS needles penetrate all tissues including the phloem and xylem layers and can be considered as sensors that measure the enrichment of tissues with ionic fluids in the volume V. The more fluid the tissue contains, the lower its impedance.

Upper and lower EIS sensors in stem have two such volumes V_U and V_L. Water with nutrients is pumped from roots through transverse osmotic pressure and evaporated through leaves via stomata by following vapor pressure deficit (VPD). In the simplest term, the amount of fluids evaporated by leaf transpiration goes through the volume V_U and the amount of fluids obtaining from roots goes through the volume V_L. Due to low transport velocity and structure of phloem and xylem tissues, the stem can be considered as a sponge-like vertical media with a slow refilling between V_U and V_L. The slow refilling means that changes in the upper part (variations of VPD, environmental influences such as ozone-induced stomatal sluggishness, changes of photosynthetic activities) will be first reflected in V_U, changes in the lower part (irrigation, soil moisture, nutrients, water contaminates) will be first reflected in V_L.

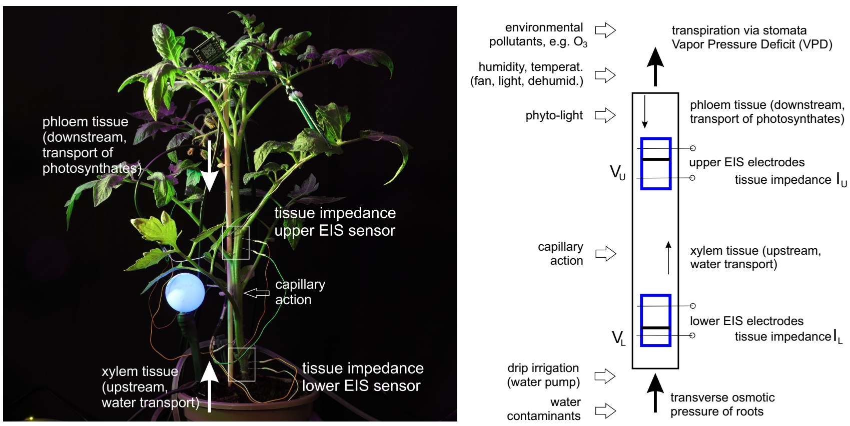


Figure 13: Hydrodynamic model of two-point electrochemical measurements in the upper V_U and lower V_L stem areas for measuring the water content.

Dynamics of upper and lower EIS sensors can be understood in the following way. Turning on the light starts transpiration and photosynthesis, this decreases amount of fluids in the upper part of the stem (in V_U area due to evaporation via stomata and the photosynthetic reaction). This, in turn, is reflected as increasing the impedance of I_U. Production of photosynthates introduces the first transient period until the downflow is stabilized along the whole stem. Any intrinsic or extrinsic variations of VDP, photosynthesis or air pollutants are reflected in the dynamics of the upper EIS sensor – it follows closely the transpiration data. Turning off the light stops transpiration and photosynthesis, the V_U volume is slowly refilling and its impedance is decreasing. This introduces the second transient period until the downflow is stabilized again. Any intrinsic or extrinsic variations of irrigation, osmotic pressure or water contaminants are reflected in the dynamics of V_L volume and data from I_L.

Differential dynamics of V_U and V_L is of especial interest. First of all, plant-common events, such as irrigation, first provide more fluids to V_L then with some delay to V_U. Similarly, turning off the light is first reflected in V_U then with some delay in V_L. In all such cases we observe a delayed reaction between I_U and I_L. Parameters that influence the chemical reactivity and capillary force are measurable in the differential dynamics of I_U and I_L.

Following a physical nature $\Delta T_{upper} - \Delta T_{lower}$ of thermal sap flow measurements, electrochemical $I_U - I_L$ data represent a similar meaning for the region $U - L$ of the stem. However, the difference is that thermal sap flow measurements are conducted in a small region of stem, whereas electrochemical data are applied to much larger U, L region. Based on the discussion about sap flow rate and sap flux density, the temporal dynamics $I_U^{\Delta t}$ and $I_L^{\Delta t}$ correspond to the sap flow at U and L, their differential value $I_U - I_L$ corresponds to the sap flux in the region $U - L$. However, results of thermal and electrochemical measurements do not completely coincide due to their different physical principles.

3.6.6 Phyto: detection of ionic vs organic fluids

EIS sensor in spectral mode can measure ionic content and differentiate between low-/high-ionic fluids such as nutrients and photosynthates. This opens additional possibilities for physiological analysis. The difference between the upflow and downflow, see Figure 13, lies in their content of nutrients and photosynthates, which generate different ionic dynamics. Considering the classic chemical reaction of photosynthesis

$$6CO_2 + 12H_2O \xrightarrow{hv} C_6H_{12}O_6 + 6O_2 + 6H_2O, \qquad (1)$$

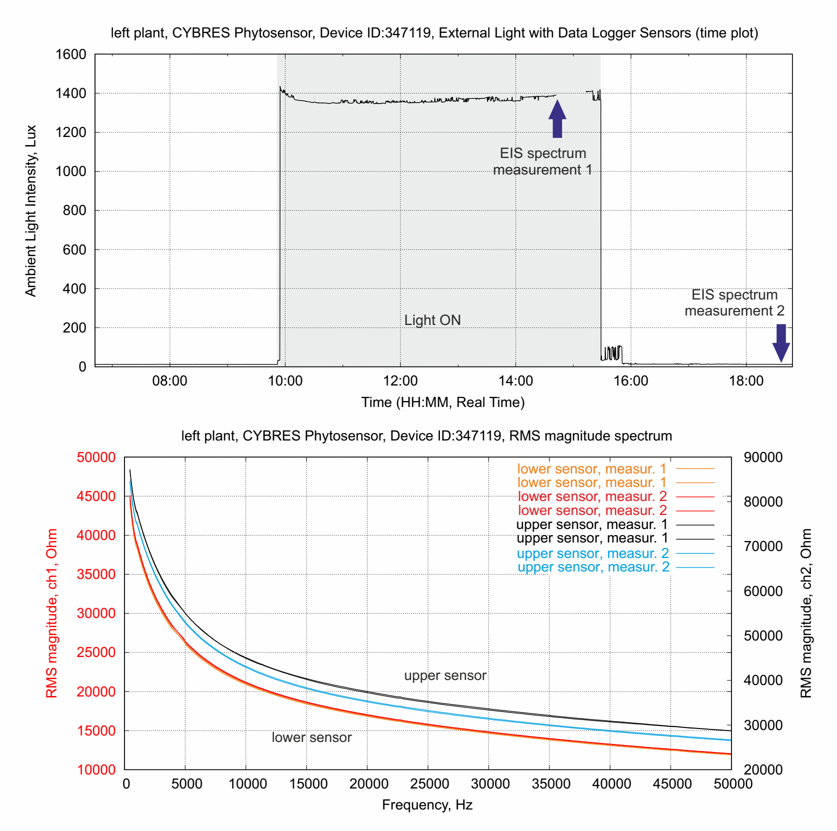


Figure 14: EIS in frequency domain applied to lower and upper sensors. Electrochemical spectrogram of the upper sensor demonstrates clear differences before and after a light excitation (presence of photosynthates in the upper area of stem), whereas the lower sensor has the same ionic content before and after the light excitation (water and nutrients from soil).

we see that the upflow in xylem tissue is represented by the LHS of (1), with H_2O containing different dissolved ions. Among them, potassium (K) and magnesium (Mg) play an important role in ionic dynamics since plants take up both minerals only in their ionic form (as $Mg_2{}^+$ and K^+). Upflow also transports $NO_3{}^-$, $PO_4{}^{3-}$ and $SO_4{}^{2-}$ ions and different ionic solutions of copper Cu and phosphorus P, such as Cu^{2+}, $H_2PO_4{}^-$ and others. The RHS of (1) represents a downflow in phloem tissue, which contains fewer dissolved ionic molecules due to the decrease in H_2O and presence of non-ionic sucrose $C_6H_{12}O_6$.

Since EIS electrodes penetrate all xylem and phloem tissues, electrochemical measurements always contain a combination of upflow and downflow. Different ionic dynamics are detectable by EIS and shown in Figure 14. Spectral EIS is performed twice in the light-on (with photosynthesis) and in the light-off conditions (without photosynthesis) for the lower and upper electrodes. The electrochemical spectrogram of the upper sensor demonstrates clear differences before and after a light excitation (via the presence of photosynthates in the upper area of stem). It is important that the second measurement in the light-off phase demonstrates a lower impedance, as this points to a higher ionic content according to (1). The lower sensor also measures differences between the light-on and light-off phases; however, these differences are much smaller than those in the upper sensor. This reflects a different combination of ionic components and photosynthates in the plant's sap in the leaf and root areas of the stem. Thus, in vivo impedance spectroscopy allows for the discrimination between aqueous solutions containing high levels of ionic nutrients and low levels of ionic photosynthetic products in a hydrodynamic system.

3.6.7 Phyto: soil moisture sensor

Soil sensors use the capacitive principle and measure soil moisture in relative units as % of containing moisture. The phytosensor has two different soil sticks – digital I2C sensor with RGB light ball (it measures the temperature and soil moisture with low resolution) and analog sensor with plastic mounting part (it measures only soil moisture with high sensitivity and resolution), see Figure 15. Since analog and digital sensors use the same data structures, only one of them can be connected to MU at the same time. The soil sensor uses internal values to determine moisture, it must be calibrated to the soil type and its moisture-holding capacity. Parameters of these sensors are shown in Table 8.

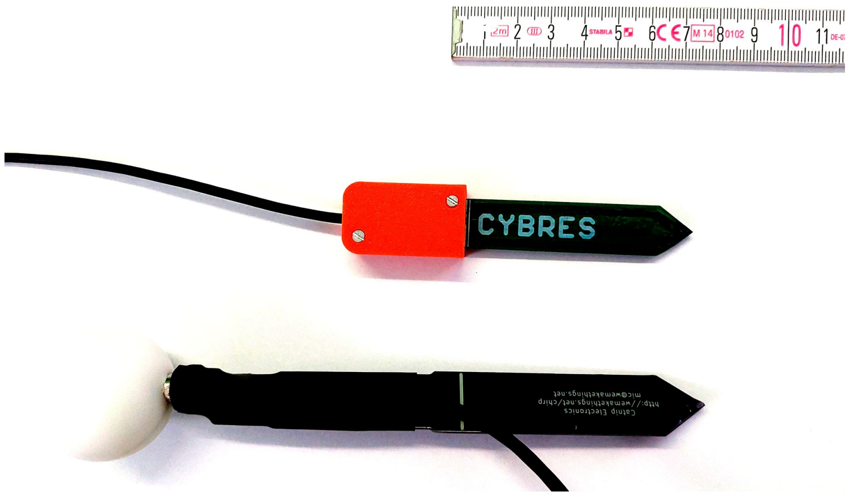


Figure 15: Analog (upper) and digital (lower) soil moisture sensors with 67 mm lower part (in soil).

Calibration. Connect the sensor to the MU module and start the measurement. First, insert the lower part of the sensor into dry soil. The recorded values correspond to the first point, '0% moisture' on the scale. Add water to dry soil until it is completely saturated. The recorded values correspond to the second point,

Table 8: Parameters of the soil moisture sensor.

principle of operation	capacitive sensing, digital and analog interface
frequency	1MHz
duration of single measurement (AC field on, periodical mode)	180ms (all sensors off) – 500ms (all sensors on)
recommended sampling rate	once per 30-60 sec.
power supply	4.2±0.1V, 5mA
output voltage range	0.1-1.1V proportional to soil moisture (pre-calibrated in analog sensor)
dimension (lower part, analog sensor)	15x67x1.5 mm or 15x100x1.5 mm
dimension (upper part, analog sensor)	21.3x36x14 mm
material	PETG or ASA (upper part), coated PCB FR4 (lower part)

'100% moisture' on the scale. All other measurements will represents the moisture in % between 0 and 100 on this scale. The analog soil sensor is pre-calibrate to air (values around 10) – water (values around 100) conditions and possesses better sensitivity and resolution than the digital sensor.

ATTENTION. To avoid root damage, place the soil sensor sideways between the pot and the soil (the sensor side with the CYBRES label should face the soil).

3.6.8 Phyto: green biomass sensor

Green biomass sensor measures the wet biomass of growing plants in a fixed volume, see Figure 16 or in open volume, see Figure 6. The sensing radius of open volume version is about 20 cm from

Figure 16: Green biomass sensor, the version for a fixed volume measurement.

each side of the sensor; it is less complex, however possesses a non-linear sensitivity: objects close to the plate generate a higher response and objects with high water content limit the sensing range (due to high damping factor of water). The green biomass sensors use dielectric spectroscopy at 0.5-3MHz based on different relative dielectric permittivity (dielectric constant) of material (e.g. water – 78.4 at 25C, 60% of sucrose solution – 60.19 at 25C, glycerol – 42.5 at 25C, air – 1.0006). Considering a high air-water dielectric contrast, the sensor primarily measures a water content of organic tissues, its dynamics follows the wet biomass dynamics. Main applications of this sensor are non-disruptive biomass monitoring in young plants, AI phenotyping, control of mass production in indoor farms and other similar cases. If the sensor is placed close to the ground (less than the distance between plates), it can also measure irrigation dynamics in soil-based and soil-free cultivation. Note, the sensor is off between short measurements (typical duration of measurement – 180-500ms) and does not generate the electric field at this time, see more details in Application Note 28.

Table 9: Parameters of the green biomass sensor.

material of electrodes	PETG or ASA, anodized aluminium
operation frequency	0.5-3MHz (dielectric spectroscopy)
duration of single measurement (AC field on, periodical mode)	180ms (all sensors off) – 500ms (all sensors on)
AC field generation (non-periodical mode)	continuously
recommended sampling rate	once per 30-60 sec.
output voltage range	0-2V proportional to wet biomass
sensing range	about 20cm left and right (open volume version, limited by objects with high water content), between the plates left and right (the fixed volume version)
zero point	variable to adapt to different geometries of electrodes
environmental sensors	air humidity – Honeywell HIH-5031-001, light intensity – EVERLIGHT ALS-PDIC15-21C, air temperature – Texas Instruments LM35
power supply	4.2±0.1V, 5mA
dimension of a single plate	100x200 mm, it can be customized by production on demand

The biomass sensor includes environmental sensors (air temperature, air humidity and light intensity – similar to the transAmb stick), one MU can sample data from three biomass sensors (only the biomass sensor 2 can have environmental sensors and thus cannot operate in parallel with transAmb stick). Parameters of this sensor are shown in Table 9. The sensor can be manufactured with customizable geometry on request; the zero point of the measuring scale can be changed (trim potentiometer on the sensor) to adapt the measuring range to different geometries of the sensor.

ATTENTION. All biomass sensors can work in autonomous mode; if powered but not connected to MU, they are operational and continuously generate AC electric field. Only if the sensor is connected to MU and configured (enabled), the sensor enters into the periodical measurement mode and does not emit AC electric field between measurements. The AC electric field is generated only for 180-500ms during measurements. If the sensors is disabled (removed from MU configuration), it enters again into non-periodical measurement mode.

3.6.9 Phyto: root biomass/irrigation (RBI) sensor

The root biomass and irrigation (RBI) sensor is similar to the green biomass sensor and uses the same approach with dielectric spectroscopy. The main difference is the placement and geometry of electrodes – they are placed below the growth containers (electrodes for green biomass sensor are placed above the growth containers), see Figure 17. The RBI sensor is optimized for soil-free

Figure 17: Root biomass and irrigation sensor placed below the growth containers.

cultivation but also works with soil-based systems. Its sensing range extends approximately 20 cm above the tray, but due to the high damping factor of water and moist root biomass, it is limited to the root layer on the tray. Thus, in addition to root biomass, it also measures the ability of the root layer to store water, including in capillary form. Note that the root biomass sensor cannot distinguish between water used for irrigation and water stored in organic tissues. The variable part of dynamics in root biomass sensors can be used to monitor irrigation. Typically, the RBI sensor replaces the soil moisture sensor, allowing three biomass sensors to be connected to one MU (see Application Note 28). Parameters of this sensor are shown in Table 9.

3.6.10 Phyto: chlorophyll sensor

The chlorophyll sensor uses the excitation spectroscopy and measures the fluorescence emitted by chlorophyll molecules after excitation at 430nm. The measured emission is at 680nm (peak) wavelength. Fluorometry is generally more sensitive than spectrophotometry and enables the *in vivo*, *in situ* and real-time measurements. The chlorophyll sensor can also be used for measuring fluorescent phytoplankton in water, however need recalibration due to light backscattering from particles in the water. LED emitter consumes one of actuating channels of MU and thus limits applications for real-time actuation (3 or 4 free actuating channels instead of 6). Note, that chlorophyll stick on the leaf requires periodical displacement in long-term measurements due to local leaf degradation, see more details in Application Note 28.

3.6.11 EIS-fluids: low ionic solutions

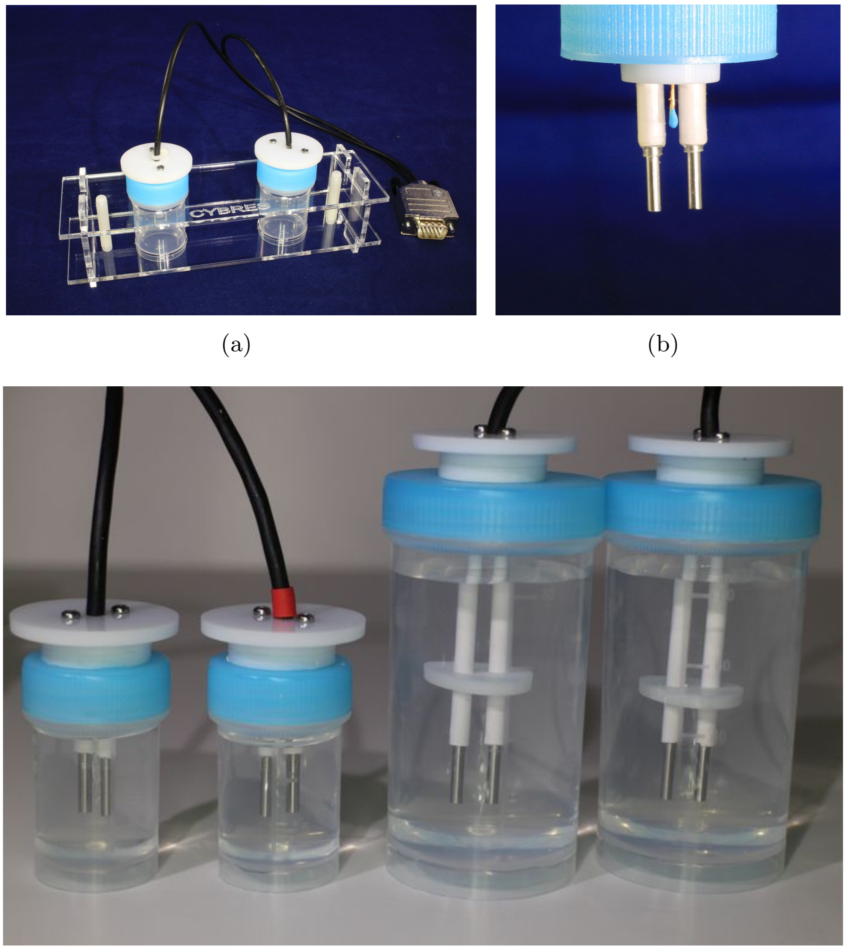
Several applications require open containers without thermostabilization, where fluids are exposed to experimental conditions during measurements. To monitor the fluid temperature in such containers, differential EIS electrodes with integrated temperature sensors have been developed, see Figures 18(a), 18(b). These electrodes are equipped with D-Sub 16/26 connectors compatible with MU34/T measurement units and feature built-in 470/940 nm LEDs for optical excitation, see Table 10. Additionally, they include an antenna for RF sensors, external temperature sensors, and monitor the supply voltage.

Table 10: Parameters of the EIS electrodes EIS-D15EIVL-16/26.

electrodes	stainless steel V4A (1.4571/ASTM AISI 316Ti), d=2mm, X6CrNiMoTi17-12-2
distance between electrodes	12mm
differential measurements	2x channels EIS
optical excitation	470/940nm LEDs inside containers
containers	polypropylene, 2x, 15 ml
additional sensors	2x internal fluid temperature sensors, external RF antenna, supply voltage sensor (4.2V)
minimal fluid measurement temperature	16C
external temperature sensor	LM35
cable length	500 mm

3.6.12 EIS-bio: fermentation and metabolic reactions

Biosensing applications typically require larger volume than fluid applications to provide better homogeneity of dispersions and suspensions. The electrodes EIS-D100EIVL-16 are equal to EIS-D15EIVL-26 but are mounted on 100 ml containers and used primarily for bio-applications with fermentation and sedimentation processes, see Figure 18(c).



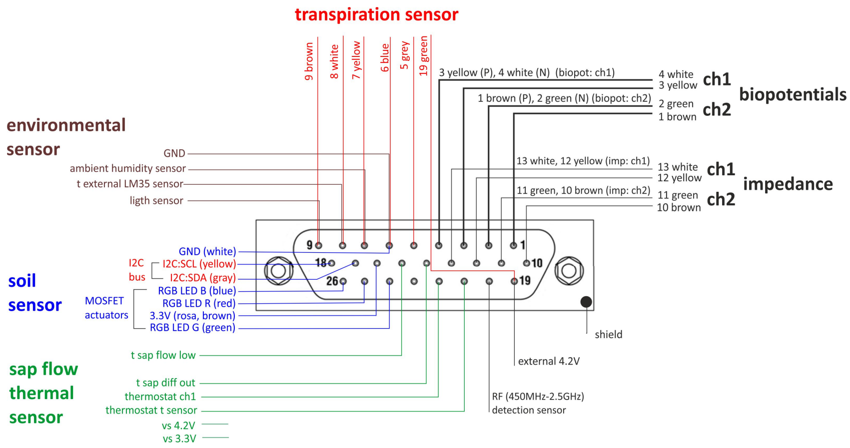
(a) (b)

(c)

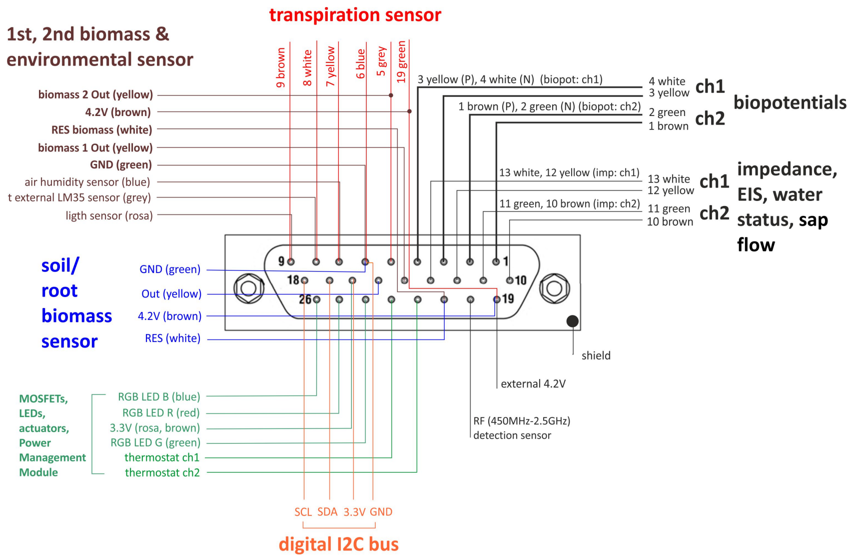
Figure 18: **(a, b)** Differential EIS electrodes EIS-D15EIVL-16/26 with 15 ml containers, integrated temperature sensors, IR/blue LEDs and D-Sub 16/26 connectors; **(c)** Differential EIS electrodes EIS-D100EIVL-16 with 100 ml containers (compared to 15 ml containers) for bio-applications with fermentation and sedimentation processes.

3.6.13 Phytosensing and phytoactuating sets

Since the MU has only one 16- or 26-pin-connector, different bio- and phyto-electrodes/sensors as well as actuators (LEDs, MOSFETS, relay or the high-power management module) are combined to sets and soldered to this connector. Such sets are customizable and depends of requirements of a particular application, however the pinout of the 26-pin-connector defines possible combinations of sensors supported by one measurement unit, see Figure 19.
If more sensors are required, it needs to take two or more MUs. All sets are labelled as Phy- (for phytosensing) or EIS- (for fluid EIS applications) sets, their notation is shown Table 11. Figure 20 shows an example of Phy-IBTSF-26 and Phy-IsBsL-26 sets of electrodes. Antenna for RF power sensors is soldered in all sets.



(a)



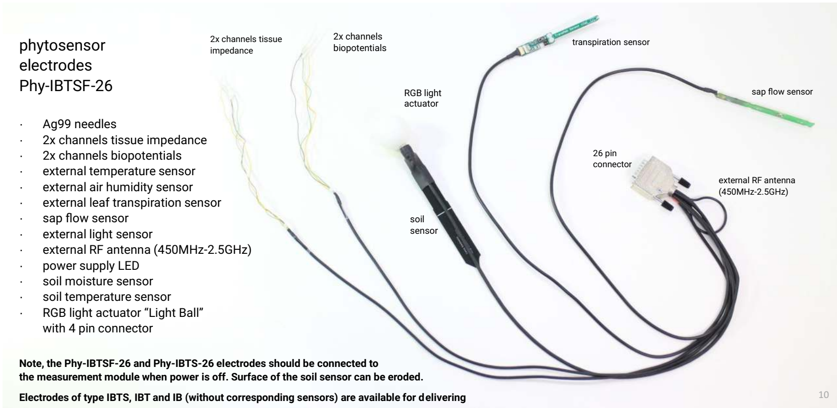
(b)

Figure 19: Pinout of the MU for connecting different phytoelectrodes/sensors; **(a)** v.1; **(b)** v.2.

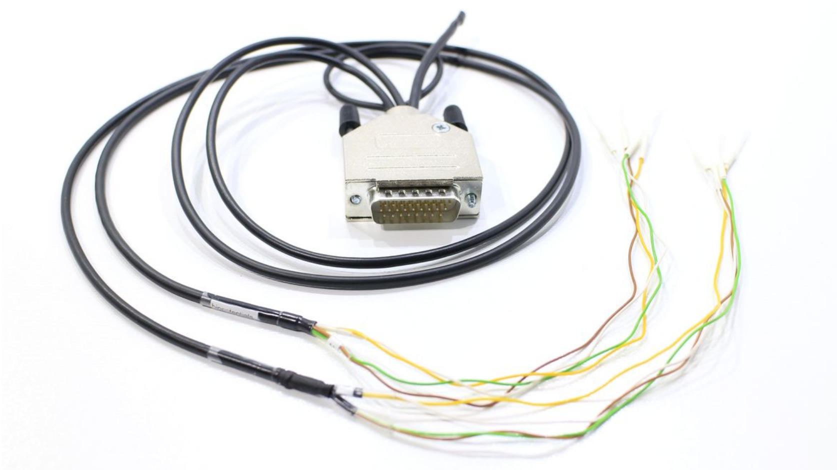
For customization of electrodes, contact info@cybres.eu.

There are several pre-configured ready-to-order sets with short delivery time (all Phyto- versions, beside chlorophyll stick, can be additionally equipped with P-part – 6-pin connector for the high-power management module, see Figure 9(f)):

- **Phy-IBTS-26**, phytoelectrodes (differential impedance/EIS and biopotentials with needles, tranAmb stick, soil sensor)
- **Phy-IsBsL-26** phytoelectrodes (single-channel impedance/EIS and biopotentials with needles, external temperature sensor)
- **Phy-IcTS-26** phytoelectrodes (differential impedance/EIS with clips, tranAmb stick, soil sensor)



(a)



(b)

Figure 20: Example of phytoelectrodes **(a)** Phy-IBTSF-26; **(b)** Phy-IsBsL-26.

- **EIS-D15EIVL-26**, EIS electrodes with 15 ml containers, 470/940 nm excitation, t sensor in fluid (available also in S version with ion-cleaning procedure)
- **Phy-Me-26**, phytoelectrodes (green biomass sensor with 100x200 mm places, with environmental measurements, available in open- or fixed-volume versions)
- **Phy-MeMr-26**, phytoelectrodes (green biomass sensor with 100x200 mm places, with environmental measurements, available in open- or fixed-volume versions, root biomass/irrigation sensor for soil-free cultivation with 50x100 mm plate)

3.6.14 Outdoor applications for phytosensor

The phytosensor is suitable for outdoor use, all plastic parts made of ASA are UV-resistant, and parts made of PETG are partially

Table 11: Notations of electrodes for Phy- (Phytosensing) or EIS- (electrochemical fluid applications) sets.

I	differential impedance electrodes, needle Ag99
B	differential biopotentials electrodes, needle Ag99
Is	single impedance electrodes, needle Ag99
Bs	single biopotentials electrodes, needle Ag99
Ic	differential impedance electrodes, clip Cu_3Zn_2
Ip	single impedance electrodes, clip Cu_3Zn_2
T	transAmb stick (transpiration + environmental sensors)
Te	transAmb environmental stick (without transpiration)
S	soil sensor, digital
Sa	soil sensor, analog
M	green biomass sensor
Me	green biomass sensor + environmental sensors
Mr	root biomass/irrigation sensor
F	thermal sap flow sensor
O	optical chlorophyll sensor
P	6-pin connector from internal MOSFETS for the high-power management module, see Figure 9(f) (it can be used also for low-power LEDs or Solid State Relays)
L	external temperature sensor LM35
V	4.2V supply voltage sensor
D15	EIS fluid cells 15 ml with optical excitation 470nm, 940nm and fluid t sensors
D100	EIS fluid cells 100 ml and fluid t sensors
E	optical excitation 470nm, 940nm
I	immersed in liquid t sensors (thermistors)
26	26-pin-connector
16	16-pin-connector
5	5-pin-connector

UV-resistant. However, the MU electronic module and electrodes should be protected from rain and direct sunlight. Examples of outdoor protection packages IP66/IP67 are shown in Figures 5 and 21(a). There are multiple options for communication and powering, e.g. PoE (Power over Ethernet) solutions, see Figure 21(b), USB-WiFi bridge or embedded systems developed for MU, see Application Note 28. Example of outdoor applications for the phytosensor from the watchPlant project[6] with the 'Orange Box' system for powering and communication[7] is shown in Figure 21(c).

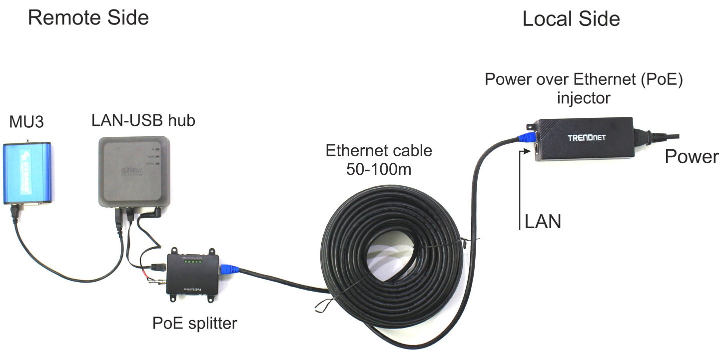
3.6.15 Configuring sensors and processing data

Embedded and external sensors need to be configured in section 'System', the field 'additional sensors', see Figure 22(a). All sensors are divided into three groups:

- environmental sensors: blue group, only one of these options can be selected;
- analog interface sensors: black group, all sensors can be switched on or off, multiple sensors are possible;
- digital interface sensors: green group, all sensors can be switched

[6] https://watchplantproject.eu/
[7] https://github.com/WatchPlant/OrangeBox/wiki

(a)



(b)

(c)

Figure 21: **(a)** Examples of outdoor protection packages for the phytosensor with **(b)** PoE powering and communication; **(c)** Examples of outdoor application from the watchPlant project with the 'Orange Box', developed by the University of Zagreb, Faculty of Electrical Engineering and Computing for powering and communication, image by CYBRES.

on or off, multiple sensors are possible;

The sampling time of an analog or digital sensor is approximately 100 ms. Turning off all sensors shortens the total sampling time in one measurement cycle. Turning on all sensors slows down the overall sampling time in one measurement cycle. If some parameter need to be sampled faster (e.g. EIS data), turn off additional sensors.

Data from corresponding sensors are available in the section 'plot', 'plot 1x: external sensors', see Figure 22(b). Note, if the sensor is configured but not connected, the corresponding sensor data will have arbitrary numbers. If the sensors is switched off, the corresponding sensor data will have '0' values.

Data processing can be conducted in several different ways as real-time or post-processing:

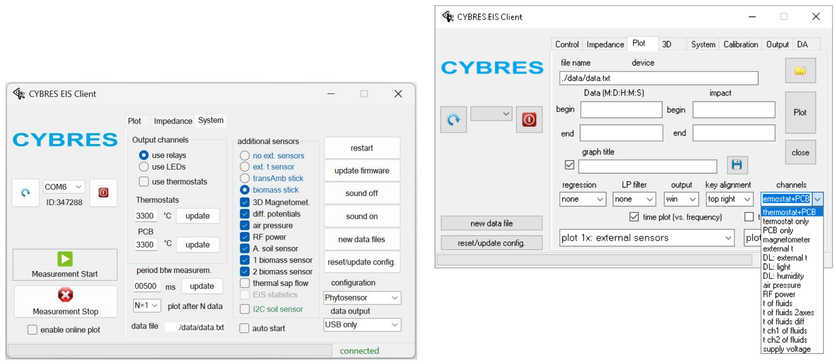
- **with embedded gnupolt scripts**, they are located in the folder '/scripts' and can be can easily adapted for particular requirements;
- **with embedded data processing engine** (e.g. statistical processors) in DA module, they can be configured and executed in real time as synthetic data channels;
- **with real-time python scripts**, the python server is implemented and provide real-time data to any external python program running in parallel to the client program;
- **post-processing of recorded data** by any analytic software.

ATTENTION. The parameter 'period between measurements' introduces a delay between measurement cycles, specified in this field. It does not specify the duration of a single measurement cycle (due to the user-defined variable sampling time of the on/off sensors).

3.7 Documentation

Documentation includes the Extended User Manual, the short Device Description, the short User Manual (translated in different languages), Technical Presentation, publications, application notes and videos:

- EIS spectrometer, all materials
- Phytosensor, all materials
- Extended User Manual
- short Device Description



(a) (b)

Figure 22: **(a)** Configuration of additional onboard and external sensors; **(b)** Selection of additional sensors data for plotting.

- short User Manual
- Technical Presentation
- Application Note 18. Online system for automatic detection of remote interactions based on the CYBRES MU EIS impedance spectrometer;
- Application Note 20. Increasing accuracy of repeated EIS measurements for detecting weak emissions;
- Application Note 24. Analysis of electrochemical noise for detection of non-chemical treatment of fluids;
- Application Note 26. Methodology and protocols of feedback-based EIS experiments in real time;
- Application Note 27. Using regression scan for electrochemical 'treatment-during-measurement' experiments;
- Application Note 28. Using phytosensor in precision agriculture, vertical farms, hydroponics and agricultural AI applications

4 Principles of operation and hardware

4.1 Electrochemical measurements

The EIS meter uses an auto-balancing bridge, where a test system is excited by the voltage V_V, see Figure 23. The signal waveform

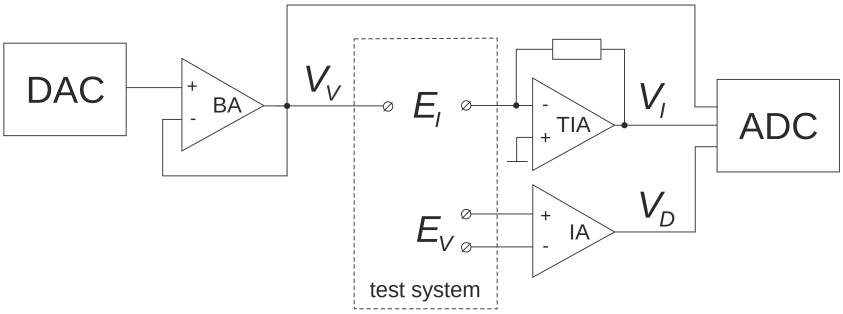


Figure 23: Structure of one measurement channel, the EIS meter has two such channels, see description in text.

for V_V is generated by DAC and is buffered by the buffer amplifier (BA). The flowing current I through the test system is converted into a voltage V_I by the transimpedance amplifier (TIA). Using of TIA shifts the phase between V_V and V_I signals by 90°. Synthesis of the signal V_V occurs by Direct Digital Synthesis (DDS) with 32-bit frequency resolution, the signals are digitalized by two synchronous 1.2 msps ADCs for simultaneous sampling of V_V and V_I signals. The EIS meter uses an external analog circuitry for impedance matching. The electrode pair E_I is used for the current sensing $V_V \rightarrow V_I$ (so-called two electrode system). Another electrode pair E_V is used to sense a differential potential with the instrumental amplifier (IA), this represents so-called four electrode system. MU EIS uses a dynamic adaptation of signal period within 3 frequency bands, the system allows any number of scanning frequencies, see also the section 5.9. The upper frequency limit is 0.65MHz (recommended for EIS 0.1MHz), MU EIS allows using harmonic and non-harmonic signals V_V for driving an electrochemical system (e.g. for fast EIS). The EIS spectrometer does not use the window functions, this allows avoiding specific errors of this approach. To avoid polarization of electrodes at low frequencies and also for measuring extremely purified water, the potential input E_V can be used for so-called four electrode measurements (the standard version of EIS uses the two-electrode approach).

ATTENTION. EIS operates in three main modes: 1) frequency mode as impedance spectrometer (also with fast Frequency Response Profile (FRP) mode based on a set of fixed frequencies), 2) signal scope and 3) continuous measurement (also with continuous FRP).

4.2 (Electro-)physiological measurements

The potential input with E_V electrodes has a very high input impedance (input bias current is about $\pm$70pA), this enables sensing of bio-potentials in electrophysiological applications, e.g. plants, organic tissues, microbial fuel cell (MFC), microorganisms and similar applications. The current flowing through the test system, see Figure 23, can be used not only for tissue impedance measurements, but also for electro-stimulation purposes. Essential advantage of this scheme is the flexible/variable frequency of stimulation, and automatic timing (e.g. a stimulation pulse every 10 sec.). Physiological measurements with plants include data from specific sensors (e.g. leaf transpiration or sup flow sensors), see Figure 24. Physio-

Figure 24: Phytosensor application with specific physiological sensors.

logical measurements with microoganisms include specific sensors and hardware modules (e.g. the biosensor hardware module) that enable performing electrophysiological and physiological measurements with selected biological organisms.

4.3 Thermodynamic measurements

Thermodynamic measurements are primarily related to EIS and are based on reactions (or physical interactions) that produce or consume energy (e.g. $H \uparrow + \downarrow OH \longrightarrow H_2O + \sim 5eV$). Both fluid containers are equipped with NTC temperature sensors immersed into fluids (resistive divider are also inside of the containers), the external temperature is monitored by LM35 temperature sensors, the PCB is thermostabilized and monitored by temperature sensor, the dynamics of stabilized supply voltage (used for temperature sensors) is also monitored. Thus, the system is well suitable for thermodynamic analysis and temperature monitoring of fluids, also with regression and statistical analysis. Thermodynamic analysis can be considered as an independent tool used for assesing results of experiments.

4.4 Optical, magnetic and temperature excitation during measurements

This experimental sensing technology is based on the excitation-response dynamics of samples (organic objects and materials, tissues or fluids) embedded into alternating electric field. The system of samples-in-electric-field is excited in optical, magnetic or thermal way. Varying the frequency of the e-field, an analysis of excitation patterns over the frequency and time delivers information about structure, behavior and dielectric/electrochemical properties of objects and materials. This approach demonstrated a high sensitivity and resolution, for instance, the sensor is able to detect smallest physicochemical differences between samples (even treated in non-chemical way). The applications are detections of low-concentrated chemical contaminations and non-chemical treatments in water quality monitoring, and an express identification of complex biochemical substances in field conditions, material analysis in biology/chemistry, biotechnology, material science, and robotics. This approach is currently implemented in form of optical excitation (UV, RGB, IR LEDs and laser excitation via DA module).

4.5 Environmental measurements

Important feature of the MU system is its capability to measure different environmental parameters during 'main measurements'. There are fixed sensors installed on the PCB, additional sensors installed in electrodes, and replaceable sensors with digital (I2C) and analog interface in 5 pin and 26 pin connectors. All these sensors can be turned on/off, see Sec. 3.6.15. Data channels from these sensors, see Sec. 7.8, can be used by the DA module for advanced numerical/statistic calculations in real time.

4.6 Using biofeedback to control external processes (e.g. light and irrigation)

MU system provides output signals to control external peripheral devices (such as solid state relays, see Sec. 4.8), this enables applications in precise agriculture to control hydroponic and indoor-farming systems by biofeedback (in addition to fixed growth protocols). The hardware and embedded software provide several timers (with real-time and periodical-timing functionality) and PWM controllers to control high-power phytolight, water pumps, fertilization and fans (external high-power management system and micro-pumps are necessary, see additional equipment).

4.7 Used methods of analysis

4.7.1 Frequency response analysis (FRA)

Electrochemical impedance spectroscopy is a laboratory technique in analytical chemistry, in biological research, for example, in the analysis of DNA or structure of tissues, the analysis of surface properties and control of materials, and other applications. This method consists in applying a small AC voltage into a test system and registering a flowing current. Based on the voltage and current ratios, the electrical impedance $Z(f)$ for a harmonic signal of frequency f is calculated. Measured data are fitted to the model of considered system and allow identifying a number of physical and chemical parameters.

A common approach consists in analyzing the frequency response (frequency response analysis – FRA) of the V_I signal, which is based on the discrete Fourier transform (DFT) and synthesis of ideal frequencies. This method is sometimes called as the single point DFT. The digitized time signal $V_I(k)$ with N samples is converted to a frequency signal, containing real $Re^{FRA}(V_I)$ and imaginary $Im^{FRA}(V_I)$ parts

$$Re^{FRA}(V_I(f)) + iIm^{FRA}(V_I(f)) = \\ = \frac{1}{N}\sum_{k=0}^{N-1} V_I(k)\left[\cos(\frac{2\pi fk}{N}) - i\sin(\frac{2\pi fk}{N})\right]. \quad (2)$$

The required by FRA period-stable detection of $V_I(k)$ signal is implemented in hardware in the system-on-chip. The FRA magnitude $M(f)$ and phase $P(f)$ of the signal are calculated as

$$M(f) = \sqrt{Re^{FRA}(V_I(f))^2 + Im^{FRA}(V_I(f))^2}, \quad (3)$$

$$P(f) = \tan^{-1}(Im^{FRA}(V_I(f))/Re^{FRA}(V_I(f))). \quad (4)$$

Additionally, the differential EIS meter uses a phase-amplitude detection of excitation and response signals. The RMS values V_V^{RMS} and V_I^{RMS} (it needs to remember that these signals are frequency f dependent, i.e. $V_V^{RMS}(f)$ and $V_I^{RMS}(f)$) are calculated as

$$V_I^{RMS}(f) = \sqrt{\sum_{k=0}^{N-1}\frac{1}{N}\left(V_I^f(k)\right)^2}, \quad (5)$$

$$V_V^{RMS}(f) = \sqrt{\sum_{k=0}^{N-1}\frac{1}{N}\left(V_V^f(k)\right)^2}. \quad (6)$$

$$(7)$$

They are used for calculating so-called 'RMS resistivity'

$$M^{RMS}(f) = \frac{V_V^{RMS}(f)}{V_I^{RMS}(f)}. \quad (8)$$

$M^{RMS}(f)$, calculated from RMS values, corresponds to the magnitude of impedance $M(f)$, calculated by FRA. The $V_I^f(k)$, $V_V^f(k)$

samples allow calculating two other values – the correlation $C(f)$ and phase $P^C(f)$ (based on the lock-in phase detector for harmonic signals)

$$C(f) = \frac{1}{N}\sum_{k=0}^{N-1} V_I^f(k)V_V^f(k), \quad (9)$$

$$P^C(f) = \frac{180}{\pi}\cos^{-1}(\gamma(f)C(f)), \quad (10)$$

where $\gamma(f)$ is a f-dependent amplitude-based coefficient, detected in V_I, V_V signals. The $P^C(f)$ is equivalent to $P(f)$, calculated by FRA.

Considering $M^{RMS}(f)$ and $P^C(f)$, the value of $C(f)$ contains both the phase and amplitude characteristic of V_I, V_V signals and thus it is the most appropriate as a single output value.

The performed analysis allows identifying five main parameters:

1. the differential amplitude of excitation and response signals (these values are included in all amplitude characteristics obtained by FRA, RMS and correlation approaches);
2. the differential magnitude/conductivity;
3. the differential phase;
4. the differential correlation of excitation and response signals;
5. variations of electrochemical stability of samples in time, frequency and time-frequency domains.

The signal scope mode is suitable for a distortion analysis, e.g. the frequency shift in organic tissues.

Update from firmware v.1187.x. The FRA, RMS and correlation analysis leaded to existence of two parallel output values with similar meaning

1. Values of $Re^{FRA}(V_I(f))$, $Im^{FRA}(V_I(f))$, $M(f)$, $P(f)$ calculated by the FRA approach.
2. Values of $M^{RMS}(f)$, $P^C(f)$, $C(f)$ calculated by RMS and correlation approach.

Since the main task of the differential spectrometer is to measure the difference between two samples, mostly the magnitude and phase of signals are of interest. Performed tests demonstrated that $M^{RMS}(f)$, $P^C(f)$ have better performance than $M(f)$, $P(f)$ in relation to signal/noise ratio and computational needs. In order to simplify the analysis, the FRA is still implemented in the device, however only $P^C(f)$, $C(f)$ calculated by RMS and correlation approach are used for plots (see Tables 16 and 17). The $M^{RMS}(f)$ is denoted as 'RMS magnitude', $1/M^{RMS}(f)$ is denoted as 'RMS conductivity', $P^C(f)$ is denoted as 'FRA phase', the $C(f)$ is denoted as 'correlation'.

4.7.2 Regression analysis

Typical dynamics of EIS data with regression analysis is divided into two phases: the phase B – background recording (time prior to experiment); the E phase is an experiment. Impact identification is based on the difference of EIS dynamics in the B and E phases, see Figure 25 – external influence disturbs the EIS dynamics in the E phase. Disturbances are expressed as statistical values – as standard deviations σ; σ_B characterizes the background, σ_E characterizes the experiment. The ratio

$$\Psi = k\frac{\sigma_E}{\sigma_B} \tag{11}$$

represents the final result: the more intense is the perturbations of the region E in relation to B, the higher is the value of Ψ. The coefficient k reflects the downward or upward trend, $k = -1$ if EIS dynamics goes down in the E phase (less than zero) and $k = 1$ – otherwise. Each EIS sensor has 2 independent channels, both can be used for experiments.

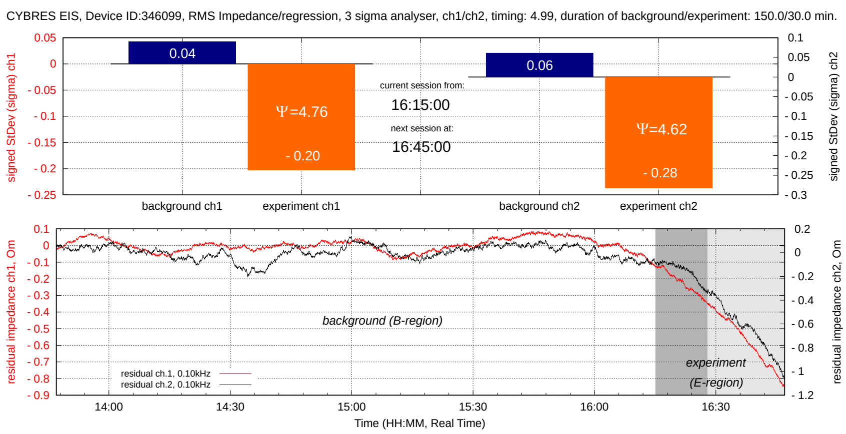


Figure 25: Example of graphical output: (upper graph) bar graphs represent standard deviations in the background and experimental regions of channels 1/2 and '3 sigma rules', the corresponding experimental graph turns orange with a significant change in $\Psi > 3$ and' experimental Ψ '>' averaged Ψ '; (lower graph) residual dynamics of both channels after regression analysis. The grey bar indicates progress of the session.

The main goal of regression analysis is to assess the difference between *expected dynamics* based on approximated data from B region and the *observed dynamics* in the E region, perturbed by 'impact factors'. If there are no differences, it means there are no 'impact factors', otherwise deviations from the expected dynamics allow identifying these factors.

ATTENTION. Regression is performed only in the background region, thus the experimental area will always show a small deviation from 0, which will fast grow when 'impact factors' influence EIS dynamics.

The original data $data(x)$ in the background region B from EIS devices is approximated linearly

$$fit_L(x) = a_l x + b_l, \quad (12)$$

or by the nonlinear function

$$fit_N(x) = a_n x^5 + b_n x^4 + c_n x^3 + d_n x^2 + e_n x + f_n \quad (13)$$

using the Levenberg-Marquardt algorithm, where we consider the residual curve

$$res(x) = fit_{L,N}(x) - data(x). \quad (14)$$

The $res(x)$ function is shown on all regression analysis charts. The function $fit_N(x)$ shows better results than $fit_L(x)$, and behaves more sensitive to small perturbations. If the approximated signal from B region differs from the observed in the E region, it generates a Δf signal, see Figure 26. The value of Δf depends on the intensity of disturbances and can be calibrated for different sensors, time periods or environmental conditions.

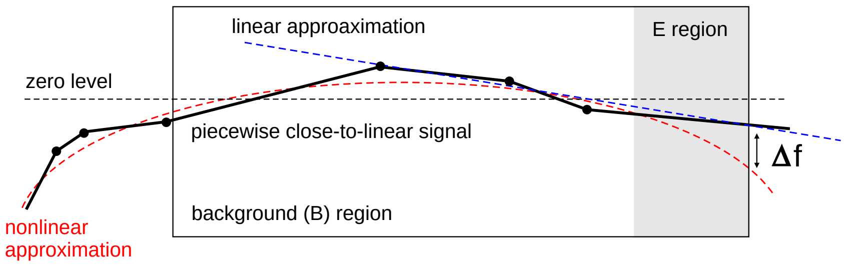


Figure 26: The value of Δf produced by $fit_N(x)$ due to fluctuations. Each point in this piecewise linear curve represents an influence on the sensor.

In fact, Δf always exists in the E region and represents environmental fluctuations and the sensor noise. Since the main goal is to detect new influences (and suppress old ones), these influences can be detected more quickly if Δf is periodically reset, e.g. by shifting the 'old E region' to the background, when the influence on E is completed. This can be done by setting a discrete time interval for sessions and calculating the regression only within this interval. The optimal time for regression is approximately 3x (background recording is 3 times longer than the experiment), a shorter time does not provide a sufficient number of data samples. Thus, 30 minutes of an experiment, about 90-120 minutes of background recording are required.

4.7.3 Correlation analysis

Correlation analysis is an important analytic tool that is implemented in two different ways: 1) calculation of sample-wise $C(f)$ (9) in lock-in phase detector in hardware/firmware; 2) calculation of Pearson's linear correlation coefficient in sliding window on the client level (in Gnuplot and Python scripts), see Fig.27.

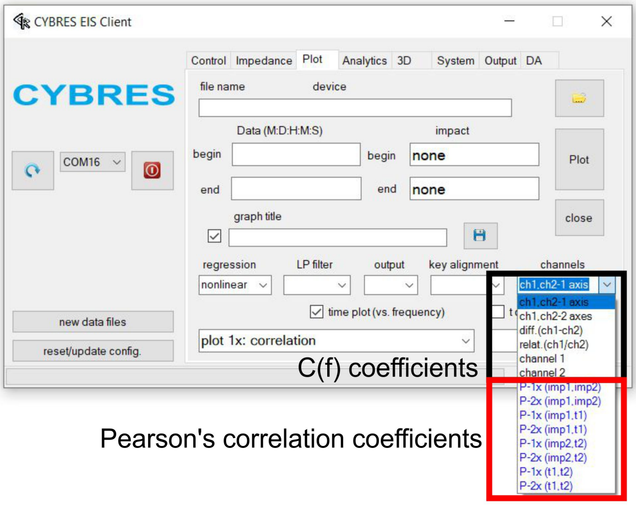


Figure 27: Calculation of correlation with $C(f)$ (9) in lock-in phase detector and Pearson's linear correlation coefficient in sliding window.

The Pearson's correlation coefficients are calculated for impedance and temperature data obtained from EIS devices in real time. First, to remove trends, the original data $data(x)$ are approximated by a nonlinear function of n-order with coefficients k

$$fit(x) = k_n x^n + n_{n-1} x^{n-1} + ... + k_3 x^3 + k_2 x^2 + k_1 x + k_0 \quad (15)$$

using the Levenberg-Marquardt algorithm [1], where we calculate the residual curve

$$m = data(x) - fit(x). \quad (16)$$

ATTENTION. Regression is performed in both background and experimental areas. Gray zone in experimental region is not shown.

Considering $m_i^{(1)}$ and $m_i^{(2)}$ as i-time step samples from two impedance-impedance, impedance-temperature or temperature-temperature channels, the transformation (16) ensures a linear dependency between $m_i^{(1)}$ and $m_i^{(2)}$ in the same scale (due to removal of different trends) that is useful for determining the Pearson's linear correlation coefficient. For calculation we follow the algorithm from [2]

$$r^{m^{(1)},m^{(2)}} = \frac{\sum_i (m_i^{(1)} - \bar{m}^{(1)})(m_i^{(2)} - \bar{m}^{(2)})}{\sqrt{\sum_i (m_i^{(1)} - \bar{m}^{(1)})^2} \sqrt{\sum_i (m_i^{(2)} - \bar{m}^{(2)})^2}}, \tag{17}$$

where $\bar{m}^{(1)}$ is the mean of the $m_i^{(1)}$'s, $\bar{m}^{(2)}$ is the mean of the $m_i^{(2)}$'s. All $r^{m^{(1)},m^{(2)}}$ are calculated as rolling correlations within the window of size t_{synch}.

For calculation of correlations between multiple independent variables we use ideas of 'a multi-way correlation coefficient' [3] based on eigenvalues of symmetrical correlation matrices

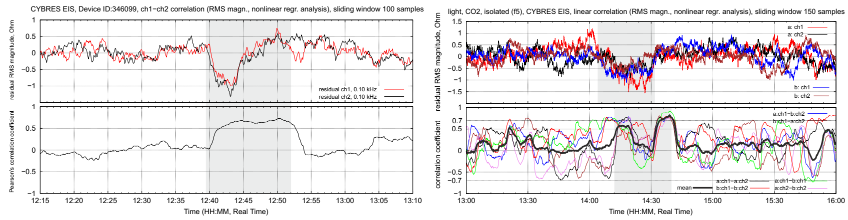
$$\begin{array}{ccccc} & a & b & c & d \\ a & 1 & r_{a,b} & r_{a,c} & r_{a,d} \\ b & . & 1 & r_{b,c} & r_{b,d} \\ c & . & . & 1 & r_{c,d} \\ d & . & . & . & 1 \end{array} \tag{18}$$

where a, b, c, d are independent variables (the coefficient R^2 of multiple correlation used for estimating predictability of the dependent variable from independent variables cannot be applied here). If p is the number of independent measurement data (e.g. water cells or organic samples), it needs to analyze $C(p, 2)$ correlation curves – combination of 2 from p (pairwise correlations from p independent electrochemical oscillators), e.g. $C(6, 2) = 15$, $C(8, 2) = 28$, $C(12, 2) = 66$, etc. Due to large number of rolling correlations, it makes sense to calculate their rolling mean:

$$r_i^{mean} = \frac{1}{C(p,2)} \sum^{n} r_i^{v,g} \mid (v,g) \in C(p,2) \tag{19}$$

Synchronization process manifests when all or several pairs $r^{m^{(1)},m^{(2)}}$ become correlated, this is observable as a peak of rolling mean, see Fig. 28. Exact values of r_i^{mean} depends on the selected size t_{synch} of rolling window. Typically t_{synch} lies between 70 and 200 samples. To find a maximal value of rolling mean, the program additionally scans t_{synch}.

Due to nature of MU EIS device with two channels, calculations of Pearson's correlation coefficients for two channels are performed in gnuplot scripts, multiple EIS/temperature channels are handled by Python scripts (also in real time by using the piping mechanism, see Sec. 9).



(a) (b)

Figure 28: **(a)** Example of a correlated dynamics of electrochemical impedances from two fluidic cells; **(b)** Example of EIS dynamics with 4 cells and six correlation coefficients. Polynomial (15) of 5^{th} order is used for the analysis.

4.7.4 Statistical analysis of electrochemical noise

This analysis is performed in time domain, i.e. with a fixed frequency (DDS mode: continuous measurements with constant f). It is based on measurements of 1) electrochemical stability and 2) fluctuations of current and potential. Both values can be considered as a long-term and a short-term electrochemical noise. Statistical parameters of electrochemical noise changes if a fluid is chemically or non-chemically processed. This analysis is sensitive to the signal range of excitation voltage (see more the Application Note 24 'Analysis of electrochemical noise for detection of non-chemical treatment of fluids').

This approach is well-known[8] in the corrosion monitoring, surface properties analysis, tests of dynamical behaviour (e.g. gas bubble formation). The statistical analysis is used in all applications of MU systems and is related to the values of 'RMS magnitude' $M^{RMS}(t)$, 'FRA phase' $P^C(t)$, the 'correlation' $C(t)$ as well as potentials, calculated at time steps t. It uses resources of the DA module.

This approach involves the notion of statistical moments in the following way. First, the values x_1, x_2, x_3, ..., x_n from $M^{RMS}(t)$, $P^C(t)$, $C(t)$ are stored in corresponding buffers as a moving window over the input data flow. It means that at each step of data sampling, the last input value is always represented as x_n, all remaining data are shifted, $x_n = x_{n-1}$, $x_{n-1} = x_{n-2}$,..., $x_2 = x_1$. The size N of the input buffers is defined by the parameter 'bufferSizeDataPipes' in the init.ini.

The mean of the values x_1, x_2, x_3, ..., x_n in each buffers is calculated as

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i. \tag{20}$$

[8] see e.g. C.A.Loto, Electrochemical Noise Measurement Technique in Corrosion Research, Int. J. Electrochem. Sci.,7,9248-9270,2012.

The second statistical moment is the variance

$$Var(x_1, ...x_n) = \frac{1}{N-1} \sum_{j=1}^{N} (x_j - \mu)^2, \tag{21}$$

its square root represents the standard deviation

$$\sigma(x_1, ...x_n) = \sqrt{Var(x_1, ..., x_n)}. \tag{22}$$

The third moment is the skewness

$$Skew(x_1, ...x_n) = \frac{1}{N} \sum_{j=1}^{N} \left[\frac{(x_j - \mu)}{\sigma} \right]^3 + k_s, \tag{23}$$

and the fourth moment is the kurtosis

$$Kurt(x_1, ...x_n) = \frac{1}{N} \sum_{j=1}^{N} \left[\frac{(x_j - \mu)}{\sigma} \right]^4 + k_k. \tag{24}$$

Since the statistical analysis is performed in real time over continuously sampled data, the expressions (21)-(24) are calculated at each step when new data sample is stored in the input buffers. This creates a time sequence of $Var()$, $Skew()$, $Kurt()$ applied to magnitude, phase and correlation that can be collected in the vector $\overline{\Phi^{1,2}(t)}$:

$$\overline{\Phi^{1,2}(t)} = \begin{bmatrix} Var(M^{RMS}(t)) \\ Var(P^C(t)) \\ Var(C(t)) \\ Skew(M^{RMS}(t)) \\ Skew(P^C(t)) \\ Skew(C(t)) \\ Kurt(M^{RMS}(t)) \\ Kurt(P^C(t)) \\ Kurt(C(t)) \end{bmatrix}. \tag{25}$$

Upper indexes denote the channels: $\overline{\Phi^1(t)}$ represents the channel 1, $\overline{\Phi^2(t)}$ – the channel 2.

Since $\overline{\Phi(t)}$ possesses a temporal behaviour, it needs to represent its dynamics in a compact form. It is implemented by accumulating these values in the following way:

$$V(t) = \sum Var(t), S(t) = \sum Skew(t), K(t) = \sum Kurt(t) \tag{26}$$

The expression (26) can be considered as definite integral between t_1 and t_2 – begin and end of measurements (or some specific time interval between t_1 and t_2):

$$\Delta V = V(t_2) - V(t_1), \Delta S = S(t_2) - S(t_1), \Delta K = K(t_2) - K(t_1) \tag{27}$$

ΔV has the same dimension as $M^{RMS}(t)$, $P^C(t)$ and $C(t)$, however ΔS and ΔK are dimensionless and vary around zero, i.e. their accumulation has a non-increasing character and thus differs from ΔV. To have a similar dynamical behaviour for all ΔV, the coefficients k_s and k_k in (23) and (24) are selected so that to keep an increasing dynamics of all ΔV ($k_s = 3$ and $k_k = 3$).

All Δ are sensitive for dynamics of $\overline{\Phi(t)}$ and are calculated for the first and the second channel as $\Delta^{1,2}$. Following the concept of differential measurements, we define a relation between channels as

$$dV = \frac{\Delta^1 V}{\Delta^2 V}, dS = \frac{\Delta^1 S}{\Delta^2 S}, dK = \frac{\Delta^1 K}{\Delta^2 K}, \tag{28}$$

$$\overline{d\Phi(t)} = \begin{bmatrix} dV(M^{RMS}) \\ dV(P^C) \\ dV(C) \\ dS(M^{RMS}) \\ dS(P^C) \\ dS(C) \\ dK(M^{RMS}) \\ dK(P^C) \\ dK(C) \end{bmatrix} \tag{29}$$

The values of $\overline{d\Phi(t)}$ are plotted as bar charts for 2,3 and 4 moments. This approach is implemented as a two-pass algorithm[9]. This approach uses current and potential noise (with two and four electrode scheme). It needs to remove all filters and to select the smallest excitation voltage (see more the Application Note 24 'Analysis of electrochemical noise for detection of non-chemical treatment of fluids').

4.7.5 Quantum phenomena involved in measurements

The measured values of impedance are affected by several environmental parameters (such as temperature, mechanical distortions or light) and electrochemical parameters of fluids (e.g. the ionization constant, number and mobility of ions). However, not only macroscopic but also microscopic parameters impact the measurements. This is related to the processes of self-ionization (dissolving the water molecules on H_3O^+ and OH^- ions), the proton tunneling effect and change of reconfigurations in molecular clusters of H_2O[10]. The self-ionization of water molecules happens due to fluctuation of electric fields, having quantum origin[11] (among other effects).

[9] W.H.Press, B.P.Flannery, S.A.Teukolsky, W.T.Vetterling, Numerical recipes in C. The art of scientific computing, Cambridge University Press, 1992.

[10] see Richardson et al, Concerted hydrogen-bond breaking by quantum tunneling in the water hexamer prism, *Science*, 351 (6279), 1310–1313, 2016.

[11] see Geissler et al, Autoionization in liquid water, *Science*, 291 (5511): 2121–2124, 2001.

Proton tunneling effect is well-known, it was first discovered in 30s of XX century and explains cases of anomalous conductivity of water[12]. These quantum effects happening on the micro-level between water molecules, ions and protons, causes changes of fluidic parameters on the macro-level, which can be in turn measured as changes of impedance e.g. in the continuous measurement mode, see additional literature in Section 11.2 for more detail.

4.7.6 Real-time signal processing and actuation

The functionality for real-time signal processing and actuation is implemented in the detectors-actuators (DA) module and is described in Sec. 10. The main goals of DA module is to provide a flexible way to create a sensor-actuator system, environmental feedback loops, and to perform fully automated humanless experiments. The DA module implements not only the reactive 'stimuli-response' behavior, it supports the probabilistic interface as event-driven Bayesian (belief) networks and several feedback mechanisms for developing adaptive homeostatic behavior. Real-time dynamical sensor-actuator mapping allows implementing advanced computer learning approaches in bio-hybrid systems by any external software. The DA functionality is implemented in firmware and in the client program.

4.7.7 Excitation spectroscopy

Excitation spectroscopy is currently an experimental technology that includes type (LED, laser, thermal, E/H fields), optical spectra (UV, visible, IR, NIR) and modulation frequencies of excitation, see Figure 29. Separate application notes will describe this approach in more detail.

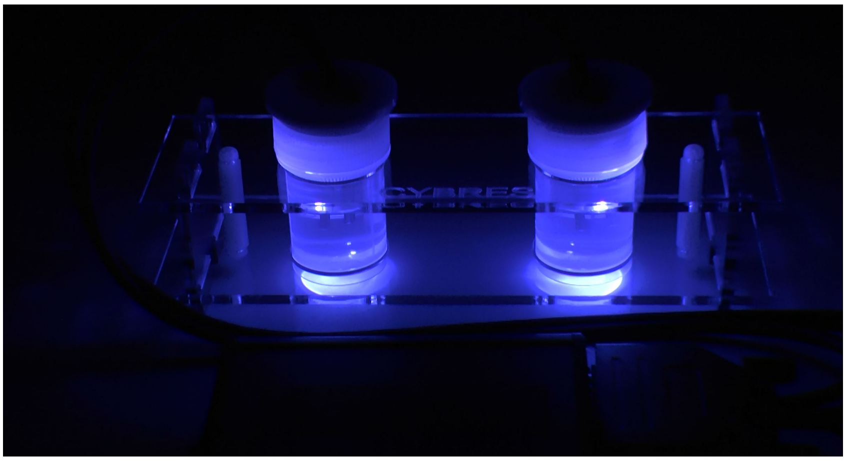


Figure 29: Example: excitation of water samples with 470nm LED light.

[12] see Bockris, Modern Electrochemistry: An Introduction to an Interdisciplinary Area, *Springer Science*, 2012.

4.7.8 Auto-gain-function and minimal impedance

The EIS system implements both manual and automatic selection of gain, required for impedance matching, see Figure 30. If the setting is automatic, the system starts from the amplification factor 50000 and step by step tests all other ranges to find the optimal match. Generally, the auto-gain-function depends on the excitation signal range and amplitude settings. Roughly, the switching (at 1V excitation and 120 amplitude) between 50000 and 5000 occurs at about 49 kOhm of input impedance, between 5000 and 500 – at about 4,9 kOhm, between 500 and 50 – at about 490 Ohm. Operation on the range 50 (impedance <500 Ohm) requires adjusting the excitation voltage to avoid distortions of signal waveform due to saturation of input amplifiers. For instance, reducing the excitation signal range to 0.1V with the amplitude 50 enables measuring the input impedance of about 5 Ohm.

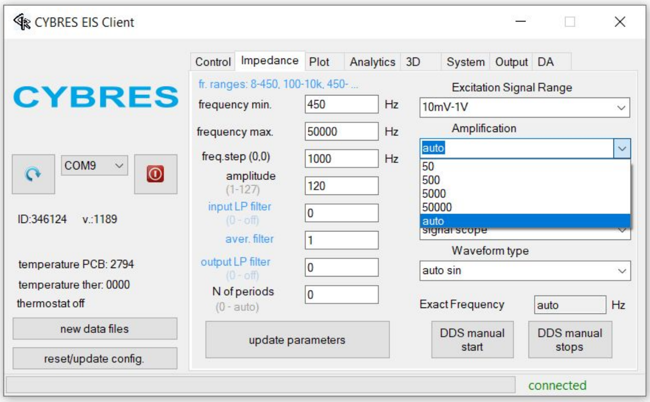


Figure 30: Manual and automatic selection of gain.

For dynamic switching-during-measurement (if the input impedance changes during measurement), the auto-gain-function has a hysteresis of about 1% of dynamic range (e.g. down-switch at 4,9 kOhm, up-switch at 5,5 kOhm). The auto-gain-function is implemented only on **the first channel**, thus users should take care about **comparable input impedance on both channels used for differential measurements**.

ATTENTION. It is recommended always to test the signal distortion in the signal scope mode before starting measurement with unknown input impedance (e.g. with new fluids). Users should never use a high impedance fluid on one channels and a low impedance fluid on the second channel.

4.7.9 Measurement of potentials

EIS system has two high-impedance inputs for measuring potentials (voltage) that can be used for: 1) 4x electrode scheme for fluid conductivity measurements; 2) bio-potential measurements in the phytosensor configuration; 3) general purpose two-channel high-resolution voltage measurements and logging. Usage in 1) or 2) requires different order and timing of sampling. Default configuration is 2), where first the voltage channels are sampled, then impedance channels are sampled, and finally all other sensors are sampled. This order of sampling provides a low level of distortions for sensing bio-potentials. If any other sampling scheme is required for a particular application, users are asked to contact the device manufacturer.

4.7.10 M.I.N.D. methodology and analysis

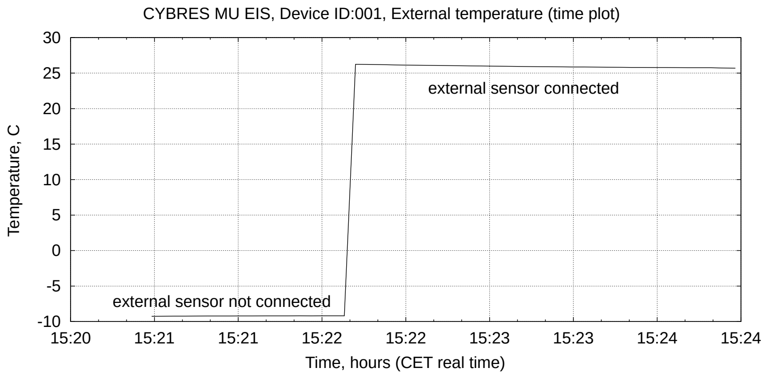
This application of EIS technology is developed for automated and adaptive measurements in feedback-based experiments. It uses a specific scheme of passive thermal stabilisation and continuously adapting statistical evaluation based on three-sigma rule, probability of random occurrence and the Mann-Whitney U-test, see the Application Note 26 'Methodology and protocols of feedback-based EIS experiments in real time'.

Data from this sensor available in the section 'plot', 'plot 1x: external sensors', see Figure 22(b). When the external sensor is not connected, the temperature data will show arbitrary numbers (even e.g. negative values), when the sensor is connected, the plot immediately shows the temperature from the sensor, see Figure 31(b).

ATTENTION. Make sure, that the external sensor is properly connected: the connector should be first fully inserted and then screwed.

ATTENTION. All M.I.N.D. scripts for web- and windows- terminals should be started anew before measurements (i.e. box 'online plot enable' should be first unchecked and then checked again), otherwise all variables will be initialised not correctly.

(a)



(b)

Figure 31: **(a)** External high resolution temperature sensor, connected to the side connector **(Note: connector should be first fully inserted and then screwed)**; **(b)** Plot of unconnected and connected external temperature sensor.

4.8 Control of external solid state relays (light and irrigation)

The system can directly control up to 6 Solid State Relays (SSR), Electromechanical Relays (EMR), high power MOSFETs and other external switching devices, see Fig. 33 and Table 12. Such relays can be used for e.g. periodical irrigation, control of phyto-light or any other actuators without connecting to PC. Periodical timers execute tasks related to on/off switching (with up to 18.2 hours intervals, resolution 1 sec.), they also have 24-hours-mode, enabling on/off of relays at the specified time point. Controlling external actuators can also be used in the biofeedback-based schemes. The system can also serve for actuating high power periphery devices controlled via the USB interface.

There are two ways to connect SSR/EMR: with internal 3.3V or 4.2V, and with external power supply, see Figure 33. Typically, most of modern SSR have 3-32V input for control, thus 3.3V can be

Table 12: Overview of available outputs and their commands for ON/OFF and PWM operations, X - 1 or 0, Y - any symbol, see Table 13.

outputs description

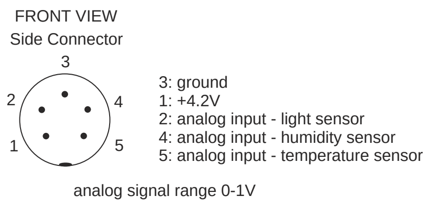
output	available modes	imple-mentation	Current	Notes
R	ON/OFF, PWM	n-channel MOSFET	1A	1;2
G	ON/OFF, PWM	n-channel MOSFET	1A	1;2
B	ON/OFF, PWM	n-channel MOSFET	1A	1;2
ts1	ON/OFF, current, PWM[7]	n-channel MOSFET	1A	1;2
ts2	ON/OFF, current, PWM[7]	n-channel MOSFET	1A	1;2
3.3V	ON/OFF, PWM[7]	power switch	3.3V, 0.3A	3

commands

output	ON/OFF (no EEPROM)	ON/OFF (w EEPROM)	PWM	Notes
R	**wlY**	**wlXY**	**wpX** – set mode **wt** – set freq.	4
G	**wmY**	**wmXY**	**wsX** – set mode **wu** – set freq.	4
B	**wnY**	**wnXY**	**whX** – set mode **wf** – set low freq. **wg** – set high freq.	4;5
ts1	**wqX**	—	PID A	6
ts2	**wrX**	**wiXY**	PID B	6
3.3V	**wvY**	**wvXY**	—	
	wkXXX			set R,G,B
	wk			get short status
	wl,wm,wn,wv			get full status
	—	**w**	—	clear EEPROM

1 – 1A max. current if using external power supply, see Fig. 33
2 – 5A per D-sub pin, GND pin is common for all MOSFETs
3 – current is limited by USB
4 – if PWM is enabled, ON/OFF=1 will start PWM
5 – it uses meander modulation with two frequencies
6 – PWM is controlled by thermostats
7 – possible, but in firmware v1189.49 is not implemented

(a)



(b)

Figure 32: **(a)** External high-resolution environmental data logger (external sensor module) with air humidity, illumination, temperature sensors **(Note: connector should be first fully inserted and then screwed)**; **(b)** Pinout of the side connector (front view on the spectrometer).

used; 4.2V can provide only 200mA current and is used primarily for powering sensors. Note that MOSFETs implement so-called low-side switching, i.e. $+V$ should be connected directly to $'+'$ of SSR, and R,G,B,*ts1*,*ts2* should be connected to $'-'$ of SSR. When using external power supply, $'-'$ should be connected to GND of the system.

Note that 3.3V output is switchable and can be used for ON/OFF purposes. In this way, use external power supply or 4.2V to have 6 switchable outputs. Output *ts1*, *ts2* produce ON/OFF signals or controllable current (depends on configuration), R,G,B output can be used for most of ON/OFF and PWM switching purposes, 3.3V can be used only for ON/OFF operations, see Sec. 4.8.2.

4.8.1 Current and frequency limitations

The system uses MOSFET SI7904BDN (max. 6A) for R,G,B outputs and BUK9K29 (max. 30A) for *ts* outputs, however note that common current is limited by:

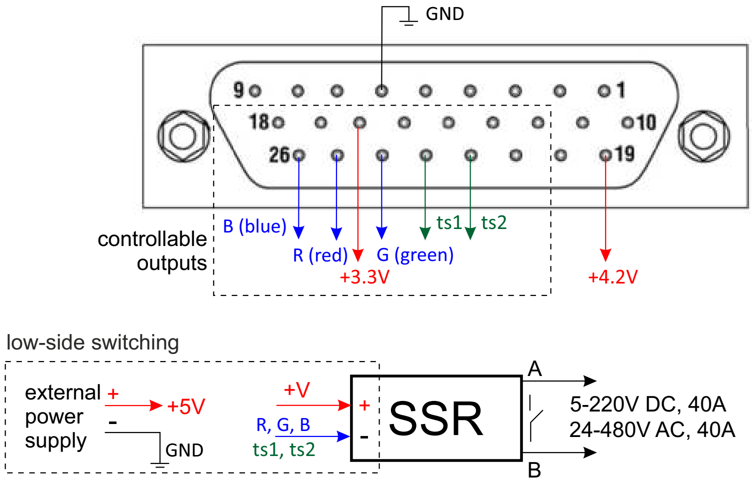


Figure 33: Connecting SSR with 3.3V, 4.2V or external power supply, *ts1*, *ts2* – output of the thermostats. 3.3V output is switchable and can be used for ON/OFF purposes.

1. max. current per D-sub connector (5A per pin, GND pin is common for all outputs);
2. if using 3.3V power source – the max. USB current (0.5A for USB 2.0);
3. if using 4.2V power source – the 0.2A for all outputs;
4. increased noise from switching elements and overall thermal dissipation.

If the external relay consumes a high current in 'ON' state (e.g. EMRs consume 0.1-0.5A in 'ON'-state), use two-step control with external power supply (e.g. the fist-stage SSRs control the second-stage high-current relay). The two-step control will also allow avoiding the switching noise that will distort measurements. Use ALWAYS external MOSFETs for PWM signals, otherwise the high-current switching noise will distort measurements.

PWM frequency range is [367Hz..12MHz] for outputs R, G, B and [0.015259 .. 500 Hz] for low-frequency modulation in output B (it has two frequencies modulations, set 100% duty for one of these parameters to use only high frequency or low frequency modulation). Note that 1) internal MOSFETs SI7904BDN have the rise time about 100 ns that distorts signal at high frequencies; 2) frequency values are converted to period values for internal PWM controllers, thus the actual frequency in high frequency range corresponds to the closet period value, see Figure 34.

Figure 34: PWM signals from R channel (1 kOhm load) at 100kHz, 1Mhz, and 12MHz settings, duty 50%.

ATTENTION. ALWAYS disconnect the power management module when changing system parameters or updating firmware, otherwise the unpredictable behaviour of outputs in the transient state can damage high-power elements.

4.8.2 Combining ON/OFF and PWM signals

All R,G,B, *ts1*, *ts2* outputs can operate in ON/OFF or PWM modes, however they implement different logic.

Generally, ON/OFF or PWM mode can be defined for each output separately, see Table 12. Note that setting PWM parameters will not start the PWM operation. By setting ON/OFF=1 the output in PWM mode is controlled by PWM, by setting ON/OFF=0 the output is driven to 0 (MOSFET is closed). This implementation can be used to control 5V-24V external devices by timers (e.g. to set a speed of water pumps). Turning OFF the PWM mode drives outputs low (MOSFET is closed). PWM parameters are stored in EEPROM, ON/OFF commands have two versions with and without storing in EEPROM (use no-EEPROM commands for frequent ON/OFF operations to minimize degradation of EEPROM).

The PWM or ON/OFF mode will automatically start operation after rebooting, if previous commands are stored in EEPROM.

ATTENTION. To avoid damaging of connected equipment, each of R,G,B outputs can be configured ONLY in OF/OFF or in PWM modes.

R,G outputs.

PWM frequency range is [367Hz .. 12MHz], PWM duty cycle [0 .. 100 %].

B output.

The B output implements modulation with two different low- and high- frequencies (so-called meander modulation), period of low frequency modulation [2 .. 65534 ms] (the frequency range [0.015259 .. 500 Hz]), the high frequency modulation [367Hz .. 12MHz], PWM duty cycle [0 .. 100 %]. Set duty cycle 100% for low frequency to use this output in a single frequency mode.

ts1, ts2 outputs.

These outputs are mainly used in different thermostats in linear current mode, i.e. their PWM signal is used to control a linear mode of MOSFETs. Thus, these outputs can be used only either as current output (via PWM with PID controllers) or in ON/OFF mode.

3.3V output.

3.3V can be used for all ON/OFF purposes (if using external power supply or 4.2V for R,G,B, *ts1*, *ts2* outputs). PWM mode is not available for this output. Its activation can be stored in EEPROM, note that any activation of '**use LEDs**' or '**use thermostats**' will activate 3.3V output and store it in the EEPROM. ON/OFF of 3.3V output will also turn ON/OFF front LED.

> **ATTENTION**. For ON/OFF mode use versions of commands without writing in EEPROM, otherwise the EEPROM will degrade faster. Note that information stored in EEPROM will always return outputs in the predefined state after rebooting.

4.8.3 Selecting LEDs or external devices

The MU system uses R,G,B, *ts1*, *ts2*, *3v3* outputs for different purposes such as thermostats or indicating LEDs. If external SSR-like devices (e.g. power management module) are connected to these output, the system should not use these output as indicators any more. The boxes 'LED/thermostats' defines their behaviour:

LEDs:

- '**use LEDs**' – (value 1), normal use for indication purposes;
- '**do not use LEDs**' – (value 0), the system will use front LEDs for indication, behaviour of outputs is defined only by commands in Table 12 and timers).

Thermostats:

- '**use thermostats**' – (value 1), the system will use outputs ts1 and ts2 as thermostats, thermostats are enabled;
- '**do not use thermostats**' – (value 0), outputs ts1 and ts2 can be used for ON/OFF by commands in Table 12 and timers).

3.3V output: Both LEDs and thermostats require 3.3V external output, thus if '**use LEDs**' or '**use thermostats**' are activated, the 3.3V output will be automatically switched in ON state.

> **ATTENTION**. For controlling external devices via R,G,B, *ts1*, *ts2*, *3v3* outputs, turn off both boxes 'use LEDs' and 'use thermostats'.

4.9 Structure of software

Software includes four levels, see Fig. 35, the lowest device-level runs on the MU device. It includes the real-time operating system – MU OS – developed by CYBRES for programmable-systems-on-chip (PSOC), impedance spectrometer, and parts of the firmware for low-level data handling. The MU OS includes different device drivers, the measurement module, data processing unit, and tasks scheduler. The client-level software runs as the MU client program and performs all main tasks related to device and file management, handling time and excitation. The client program operates with gnuplot and DA (detector-actuator) scripts. Gnuplot scripts represents the third software level and is responsible for graphical utilities, 3D/4D plot and regression functionality. Finally, DA scripts handle statistical data processing, sensor-fusion functionality, perform actuator control and multi-device management. Gnuplot and DA scripts are open for users and can be customized for particular purposes.

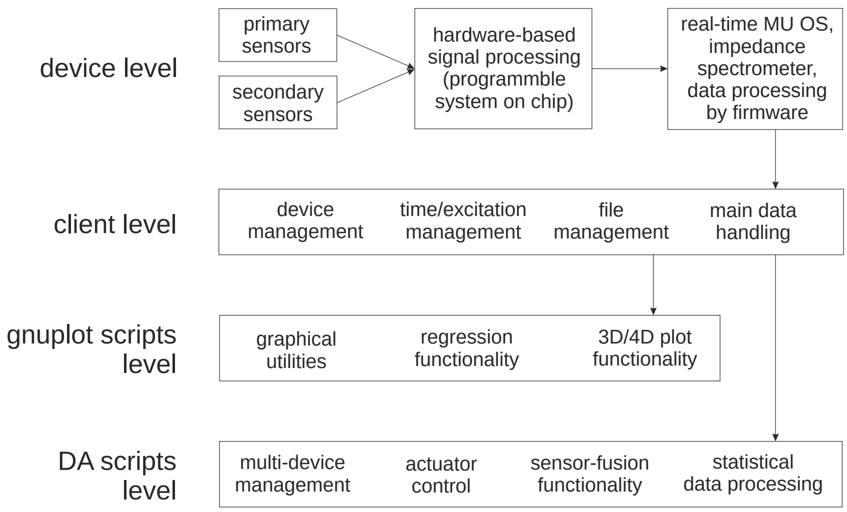
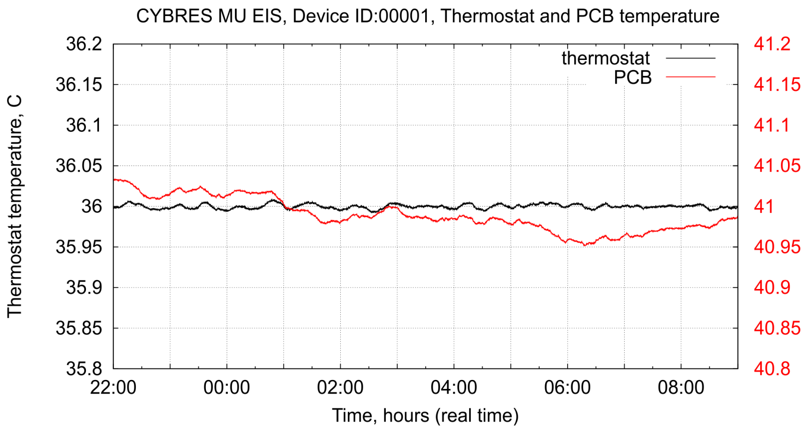
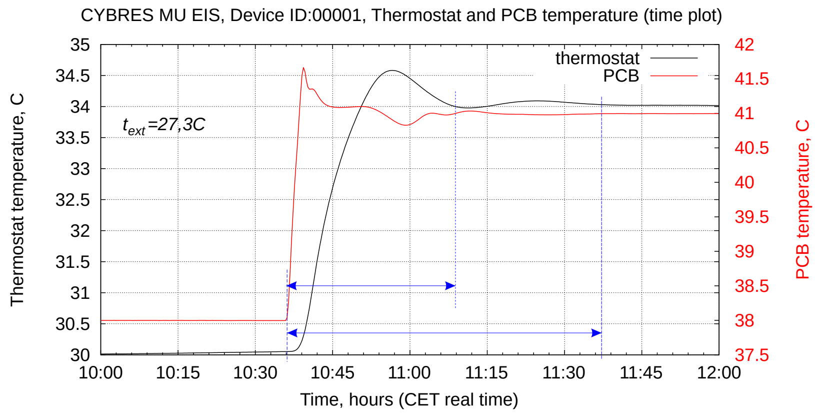


Figure 35: Software structure of the MU system.

4.10 Typical diagrams

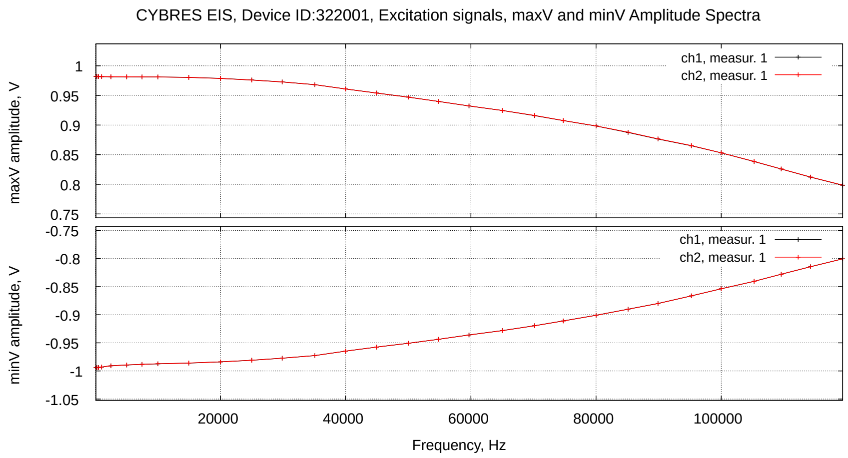


(a)

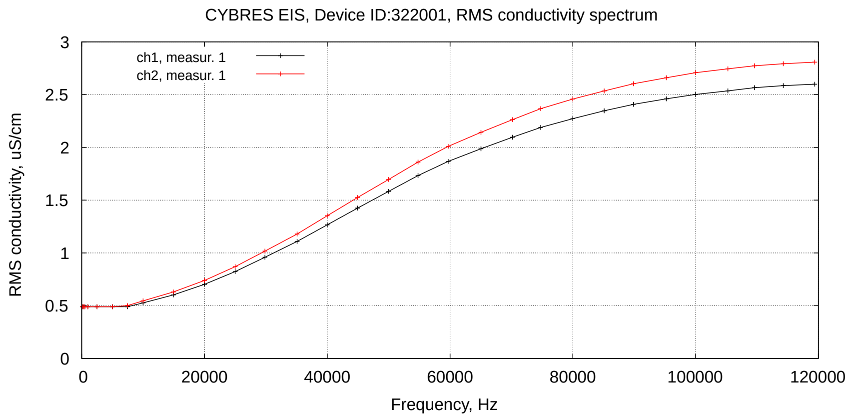


(b)

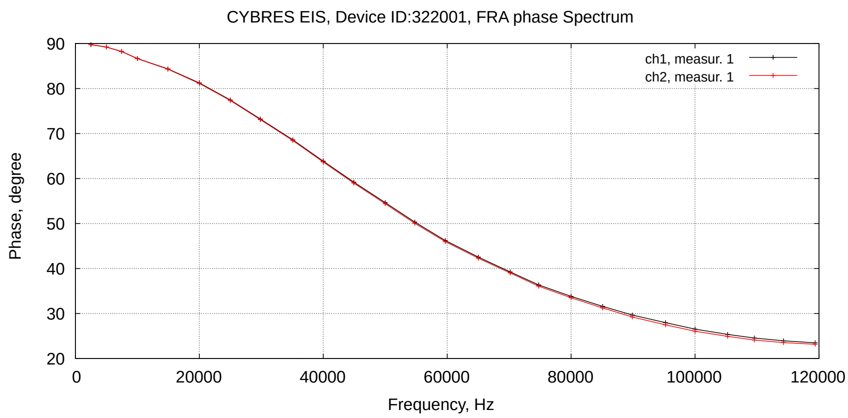
Figure 36: **(a)** Performance of the sample- and PCB- thermostats in continuous measurement mode over 11 hours in empty room. **(b)** Typical time for achieving the set temperature of sample- and PCB- thermostats.



(a)

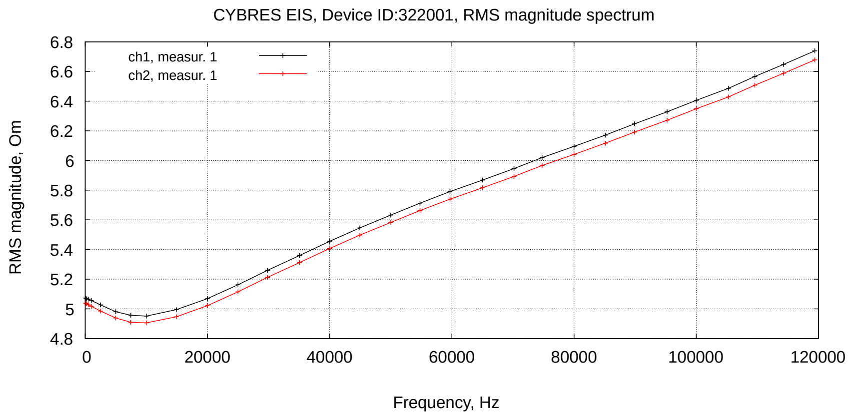


(b)

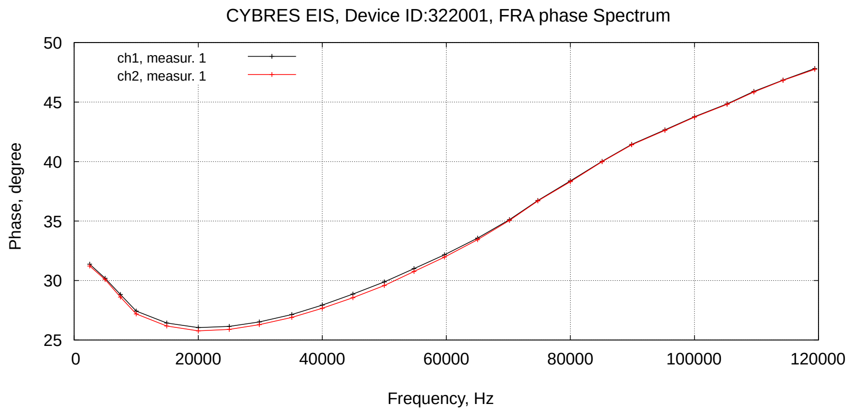


(c)

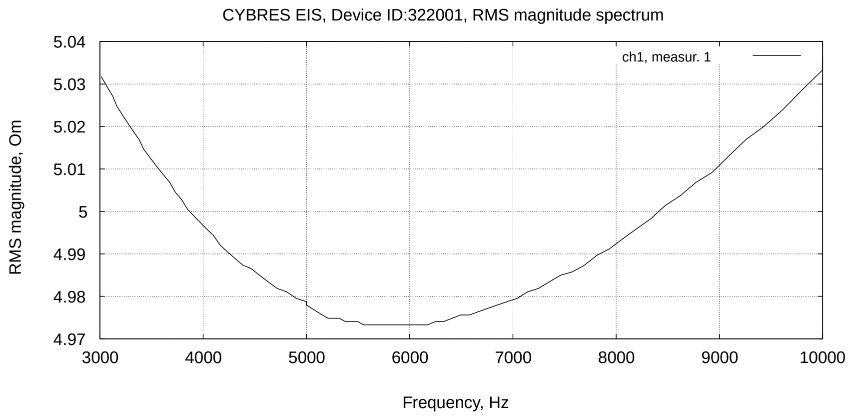
Figure 37: Dynamic range of **(a)** the excitation signals V_V; **(b)** own parasitic RMS conductivity (electrodes are not connected); **(c)** own parasitic FRA phase (electrodes are not connected). Settings: amplitude 120, amplifications 50000, FRP mode (frequencies 100Hz-120kHz), EIS MU3.2a.



(a)

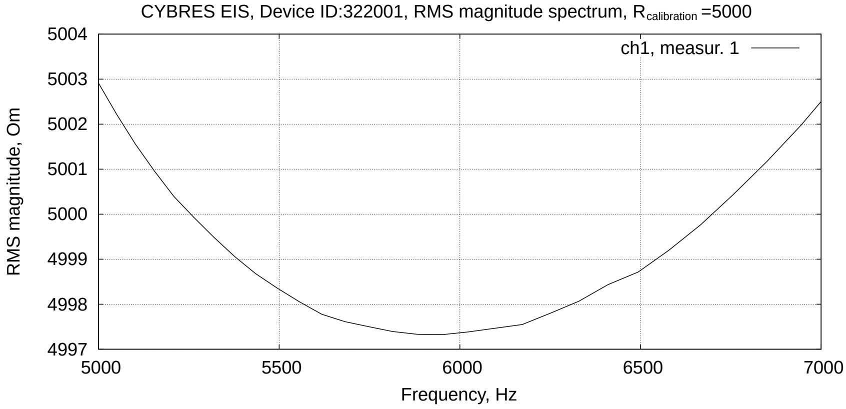


(b)

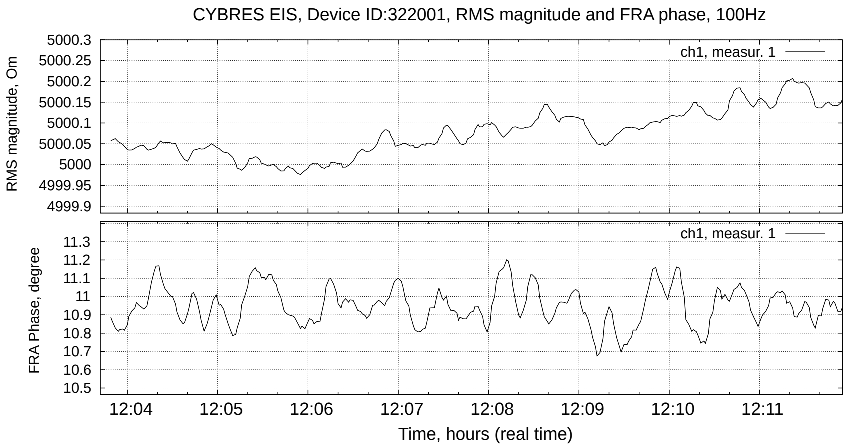


(c)

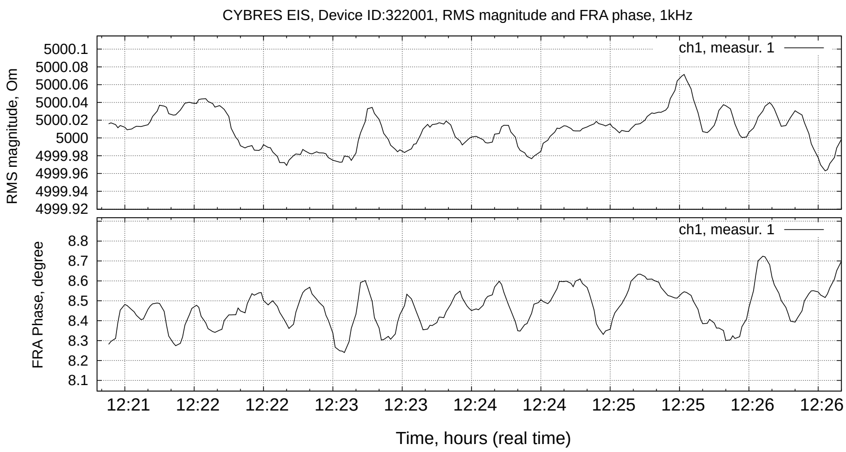
Figure 38: 5 Om resistors (±1%) connected to input EIS channels: **(a)** RMS magnitude; **(b)** FRA phase; **(c)** nonlinearity of dynamics 3kHz-10kHZ (±0.6%). Settings: amplitude 10, amplifications 50, FRP mode (frequencies 100Hz-120kHz), EIS MU3.2a, the correction coefficient 0.77.



(a)

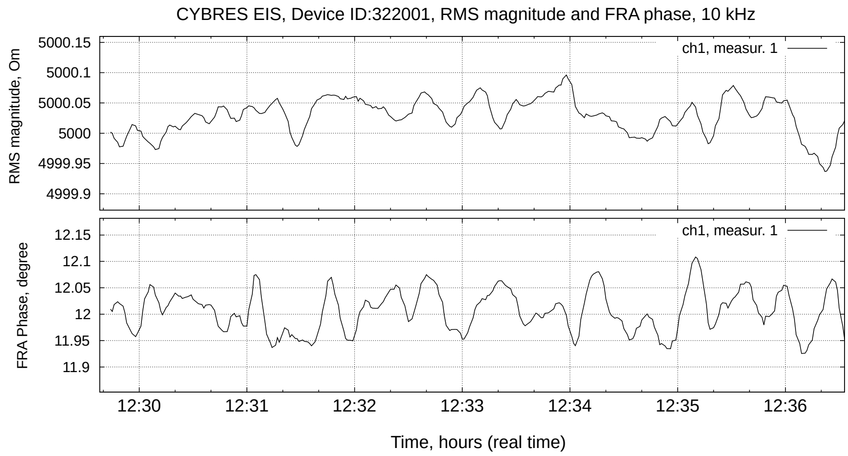


(b)

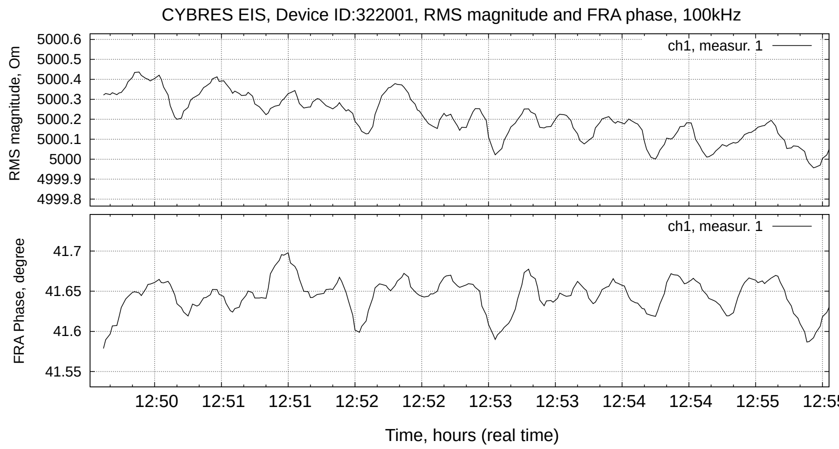


(c)

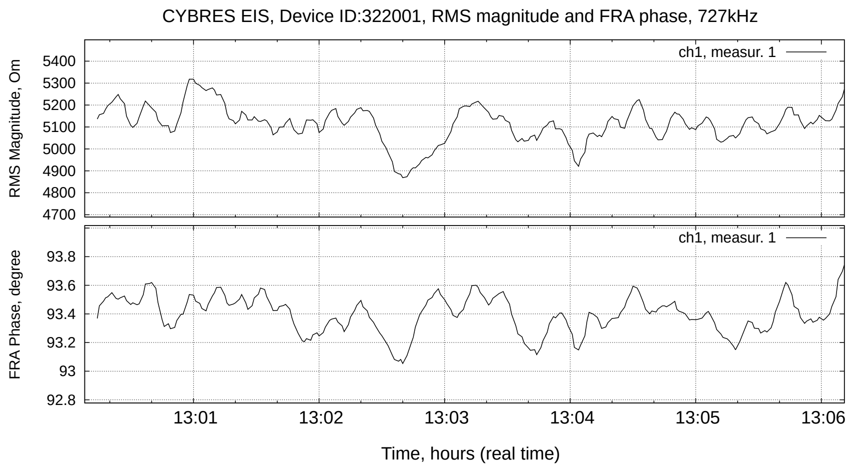
Figure 39: **(a)** Impedance spectra within 5kHz-7kHz of the internal calibration resistor 5 kOm, EIS MU3.2a, the correction coefficient 0.978; **(b)** Noise performance at 100Hz, the correction coefficient 0.93855,; **(c)** Noise performance at 1kHz, the correction coefficient 0.94609. The internal calibration resistor 5 kOm, EIS MU3.2a, thermostats off, the input LP filer – 900, the output LP filer – 3000, the averaging filter – 8, the averaging filter of client program – 5 values.



(a)

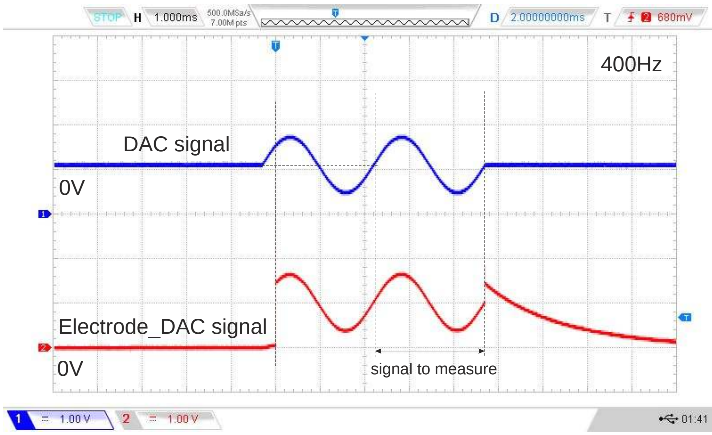


(b)

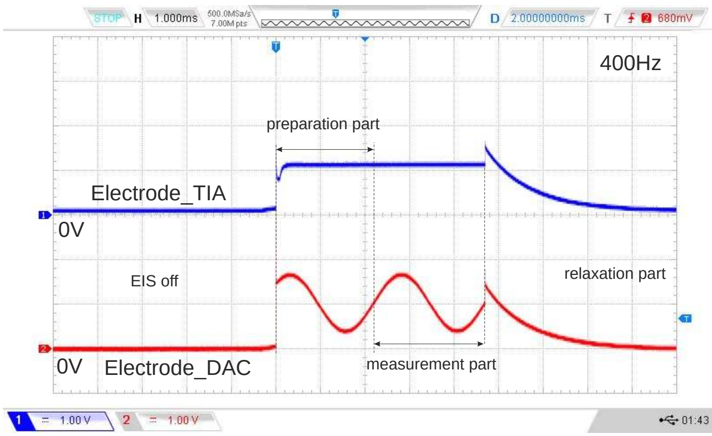


(c)

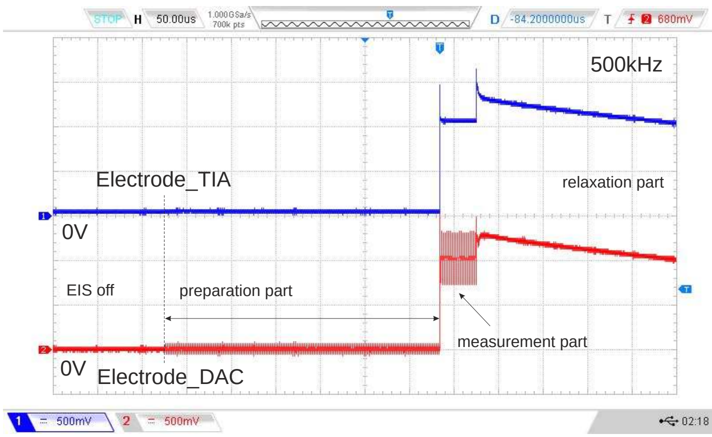
Figure 40: **(a)** Noise performance at 10kHz, the correction coefficient 0.96528; **(c)** Noise performance at 100kHz, the correction coefficient 0.69865; **(c)** Noise performance at 727kHz, the correction coefficient 0.31. The internal calibration resistor 5 kOm, EIS MU3.2a, thermostats off, the input LP filer – 900, the output LP filer – 3000, the averaging filter – 8, the averaging filter of client program – 5 values.



(a)

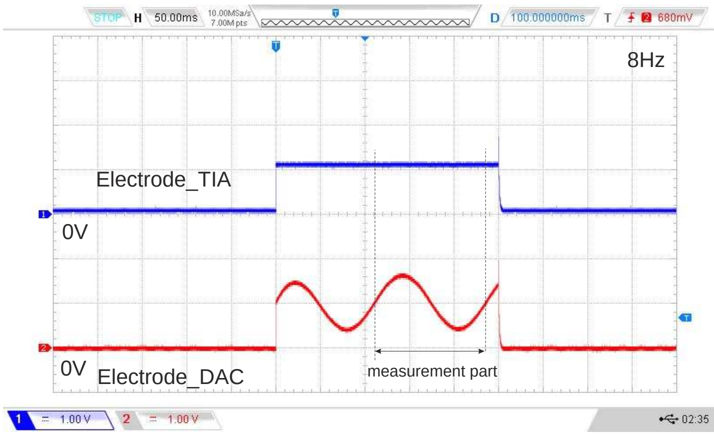


(b)

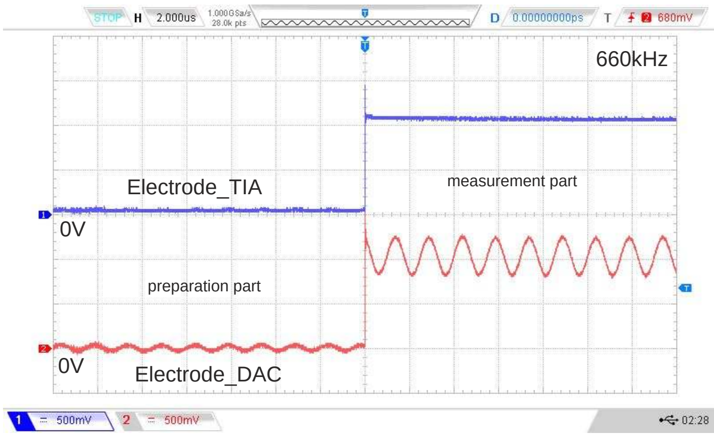


(c)

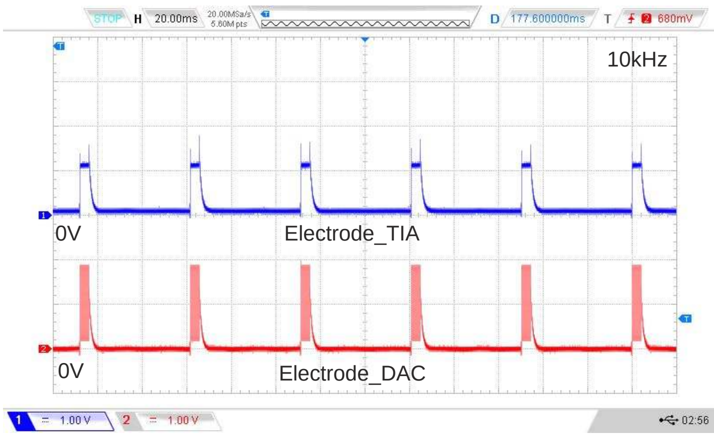
Figure 41: **(a)** Timing of excitation signal at 400 Hz on the DAC output and the external electrode E_{DAC}; **(b)** the same signal as in (a) on the external electrodes E_{DAC} and E_{TIA}. Different phases of excitation signal are shown; **(c)** Preparation phase of excitation signal outside of the measurement part at higher frequencies (500 kHz). Measurement conditions: external resistor 59 kOm connected to electrodes, < 0.01mV difference between electrodes E_{DAC} and E_{TIA} during 'EIS off' and 'relaxations' parts.



(a)

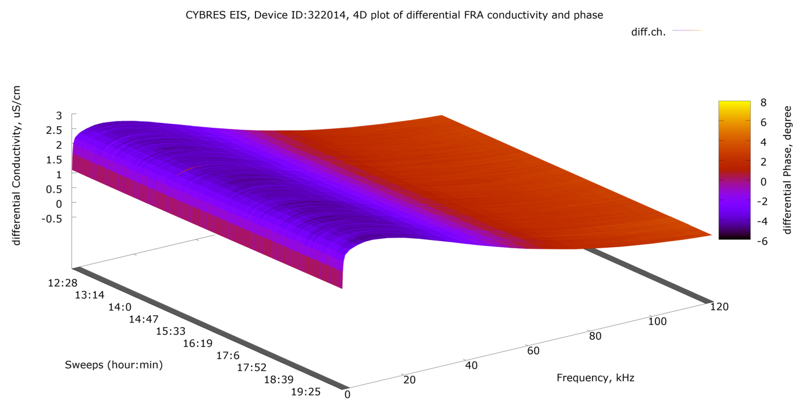


(b)

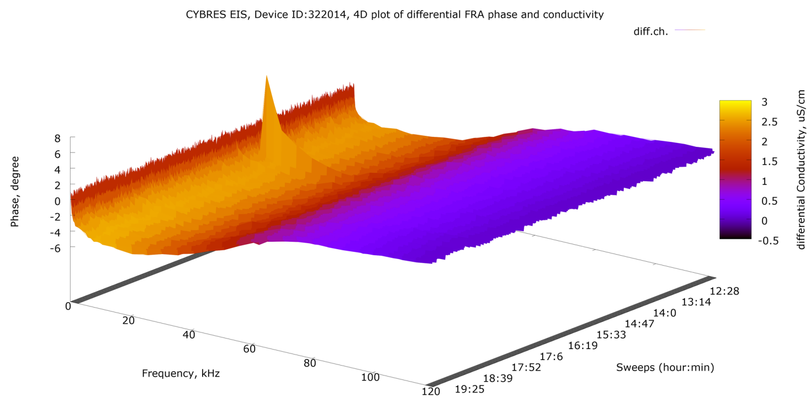


(c)

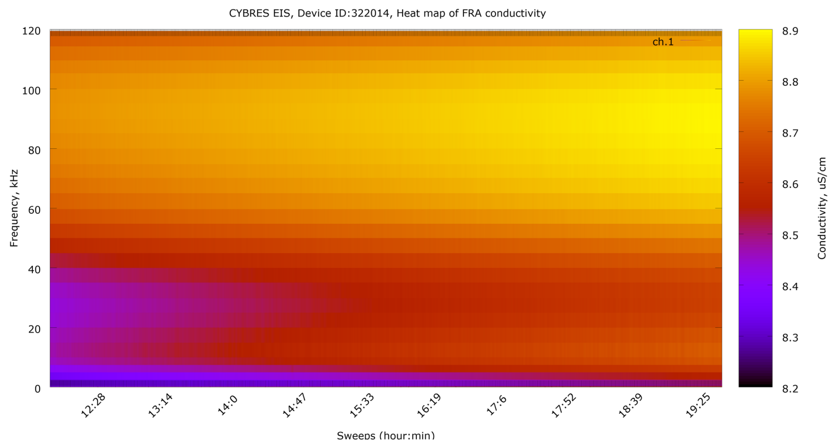
Figure 42: Timing of excitation signal on the external electrodes E_{DAC} and E_{TIA} at **(a)** 8 Hz signal; **(b)** 660 kHz signal; **(c)** 10kHz signal when the averaging filter is set to 6 (averaging within 6 repeated measurements). Measurement conditions: external resistor 59 kOm connected to electrodes.



(a)

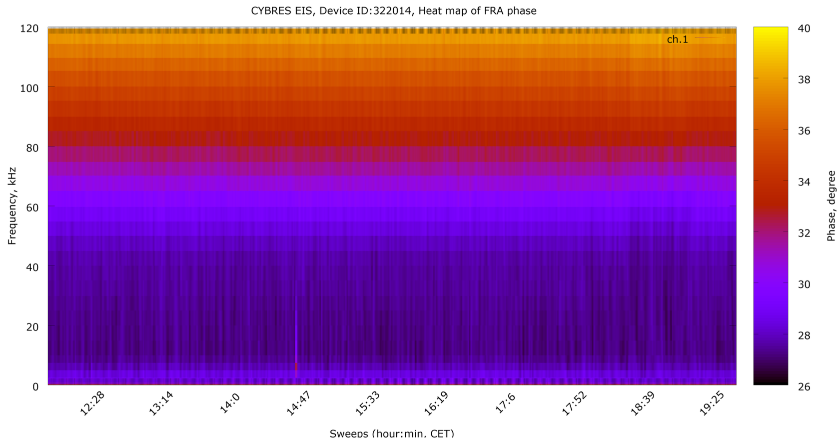


(b)

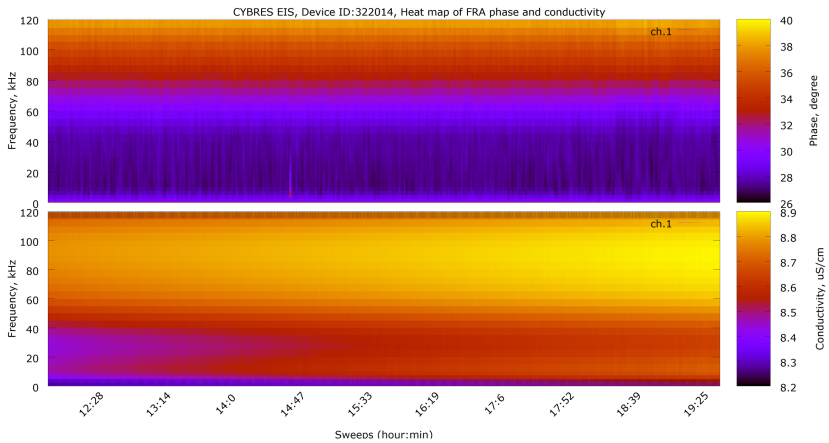


(c)

Figure 43: **(a,b)** Examples of 4D plots in continuous FRP mode; **(c)** Example of heat map plot of conductivity in continuous FRP mode.



(a)



(b)

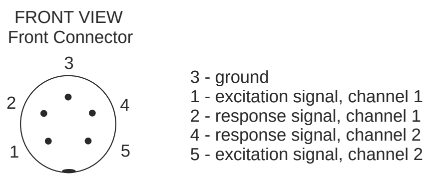
Figure 44: **(a,b)** Examples of heat map plots in continuous FRP mode.

5 Before measurements

5.1 Connectors

The MU system has several implementations such as 'round' and 'blue box' systems. The 'round' measuring module has one USB connector, the start-stop button, and the RGB LEDs, see Figure 45. The start-stop button allows user to start or to stop the mea-

(a)



(b)

Figure 45: **(a)** Connectors on the back side of the round MU EIS: (1) – the USB connector, (2) – the power switch, (3) – the start-stop button, (4) – the side connector for external sensors, (5) – the connector for electrodes, (6) – the left electrode (channel N1, red mark), (7) – the right electrode (channel N2). RGB LEDs are integrated into the hull; **(b)** pinout for the front connector with electrodes (two electrodes version).

surement. The RGB LEDs indicates different modes of operations, USB connectors are used for data exchange with PC.

The 'blue box' system, see Figure 47, has a start-stop button, LED and UART port on front side, and 26 pin connector on the back side, see Figure 46.

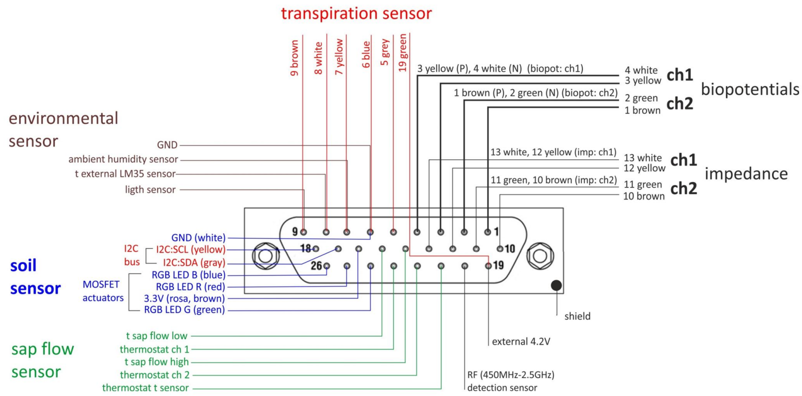


Figure 46: Pinout for the 26-pins connectors on the back side of the MU3 EIS ('blue box').

5.2 Typical connections

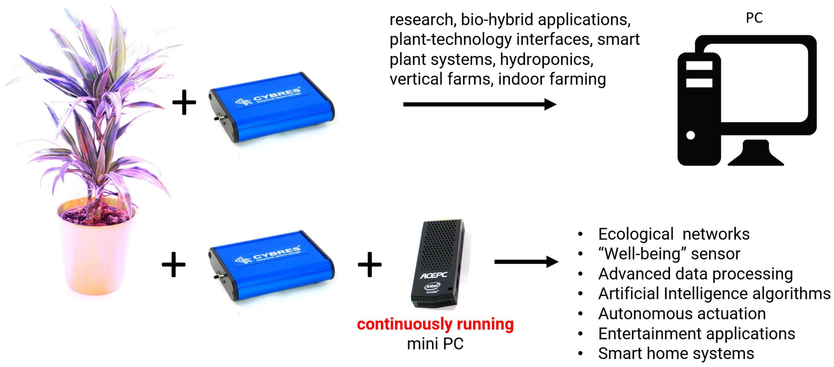
Different application scenarios and connected devices for the MU are shown in Figure 47. The system can operate as stand-alone or connected-to-PC device (if complex computational and communicational capabilities are required, use miniPC). USRT and I2C buses provide different possibilities for connecting additional sensors and actuators.

5.3 Selection of the power supply

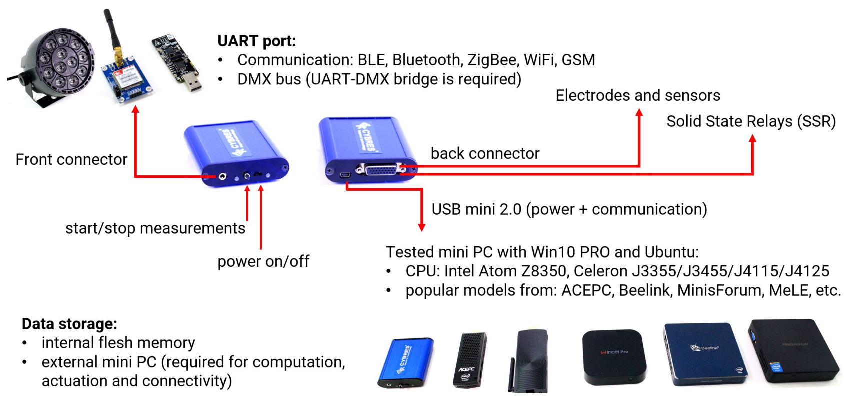
The device uses USB for powering. The USB power supply should have a low noise. This can be achieved by using a battery pack, USB POWERBANK, or USB 3.0 hub that is connected to an uninterruptible power supply with the line filter. It is recommended to connect the spectrometer to PC/Laptop after the USB 3.0 hub.

ATTENTION. It is not recommended to connect any other devices in this network, i.e., the MU EIS device must be a single device on this power supply, see Figure 48. Do not connect/disconnect any Ethernet/USB/Powering devices to the measurement system during measurements.

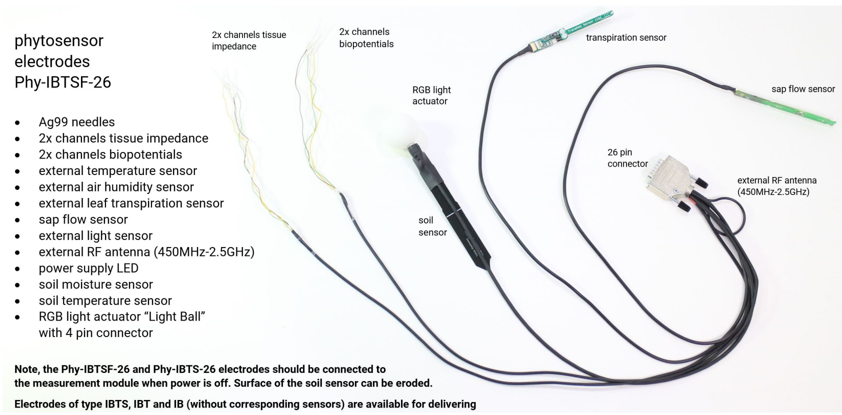
ATTENTION. When thermostats and LEDs are on, during archiving the set temperature at some settings of PID controller, the device can for a short time overstep the 0.5A limit of USB 2.0 (typical consumption is ≈0.3A). To meet the requirement on max. USB current, <u>it is recommended to use an external active USB 3.0 hub</u>, to use the USB 3.0 port (with max. current 0.9A) or to switch off the LEDs. Avoid using long ($> 1.5m$) USB cables.



(a)



(b)



(c)

Figure 47: Different application scenarios and connected devices for the MU.

Figure 48: Connecting an uninterruptible power supply with line filter – the measuring system should be a single device in the network.

5.4 Before starting measurements

1. Start the client program (see Section 6.3).

2. Connect the USB connector to the MU EIS module. The unit will start the operating system and perform a self-test of all systems. The device signals a readiness to work with sound and LED signals (see Section 5.7).

3. Select the com port in the client program and connect the device (see Section 6.3).

4. The device has a nonvolatile real time clock. Time is set at factory calibration. Check the system time and synchronize it with the computer if necessary (use the button 'set time').

5. The device can write data to the internal nonvolatile memory (offline mode) and to PC (online mode). In the offline mode graphics are built after the measurement. This mode allows working without a computer. In the online mode the graphics are build in real time. To change the mode of operation, change the settings in the client program (see online and offline modes of operation).

5.5 Start and stop of measurement

The EIS starts to achieve a predetermined temperature in thermostats immediately after powering the device. If the temperature has not reached the predetermined level, the LEDs of thermostats are purple. When the temperature is at a predetermined level, the LEDs switch to cyan. The achievement of a stable temperature requires about 10-25 minutes (this depends on external conditions).

To start the measurement press the 'start-stop' button. At the beginning of the measurement the LED will switch to green. To stop the measurements it needs to press this button again. Alternatively, press the button 'Start Measurement' and 'Stop Measurement' in the client program, in this case the selection of the online or offline measurement modes are possible (see section 7.4).

5.6 Reading the last measurement result from memory in offline mode

The data in the device memory remain even after the removing of the power supply. To read the data from memory, click 'Read Last Measurement' (offline mode) in the client program. The data of the last measurement of the internal memory will be copied to a hard drive of the connected computer. Large data volumes require some time for the transmission via USB interface.

5.7 The color and sound indication

MU EIS indicates different modes of operation by sound and RGB LEDs.

Sound indicator:

- *Two short beeps* - initialization is performed, the device is ready;

- *One short and one long beeps* - start or stop of measurements;

- *Two long beeps* - errors are detected;

- *One short beep* - the start-stop button is pressed.

RGB LEDs:

- *Cyan* - the device is ready to start the measurement;

- *changing Cyan* - thermostat temperature has not yet reached a predetermined level (if thermostats are enabled);

- *Green* - the unit is performing measurement (start button is pushed, any mode);

- *Short blue flashes* - the device is in the bootload mode (for updating the firmware).

5.8 Setting the thermostat temperature

The sample thermostat must be set to 4-5 °C (the PCB thermostat – 12-14 °C) above the ambient temperature. For example, at a maximum ambient temperature of 26 °C, the sample thermostat can be set to 31-32°C (38-40°C in the electronic module). Too high temperature requires high current from the USB network; the thermostat is unable to regulate the temperature effectively at a too low set temperature.

ATTENTION. The change of the set temperature in thermostats and switching on/off of LEDs must take place prior to measurements. During the measurement, these settings should not be changed, otherwise the measured signal will be distorted.

5.9 Understanding the dependency between signal amplitude, waveform resolution and frequency resolution

The signal amplitude is controlled in digital way by setting the maximal amplitude in the DAC module. The higher is the signal amplitude, the more digital levels can be used by DAC to generate the sinus signal. For instance, the amplitude 10 means that DAC has only 20 values to generate the waveform that may result in a coarse signal form. The maximal amplitude is 127 that provides 254 levels to generate waveforms. It is recommended to keep the amplitude as maximal as possible by selecting the proper amplification range.

It needs to note that too high amplitude of the V_V signal can lead to saturation of the V_I signal. It is recommended to check the signals amplitude in the scope mode before performing measurements.

5.10 Accurate long-term differential measurements up to $10^{-9} - 10^{-11}$ S/cm

Differential approach with thermostabilization of samples allow a great resolution of impedance/condictivity/resistivity measurements. Such EIS measurements are usually related to detection and characterization of weak emissions possessing non-electromagnetic, non-acoustic, non-thermal and non-mechanical nature. Sensing of very small changes of conductivity on the level of $10^{-9}-10^{-11}$ S/cm require an accurate handling of systematic and random inaccuracies during measurements, and strict following the experimental methodology, see the Application Note 20: 'Increasing accuracy of repeated EIS measurements for detecting weak emissions' for further detail.

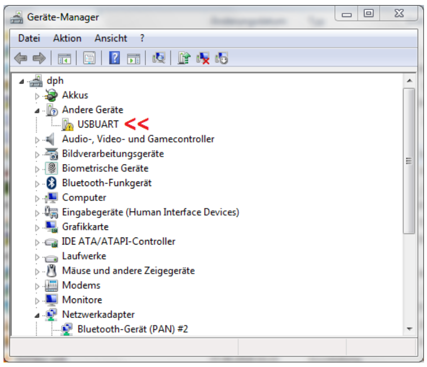
6 The client program

6.1 Software installation

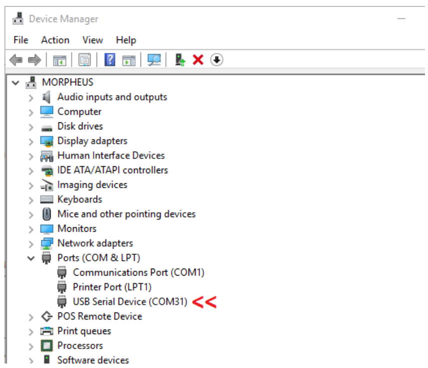
Install the necessary software (Windows 7, 8, 10, 11 are tested):

1. Install the redistributable package for visual C++ 2012 (32/64 bit versions are in the directory 'drivers').
2. Install the drivers for interface the MU EIS device to Windows.

    - For devices, based at **MU3.2** device versions or earlier. Install the driver for 'The CP210x USB to UART Bridge Virtual COM Port' (file is included in the directory 'drivers'). Run CP210xVCPInstaller_x86.exe file for 32 bit version or CP210xVCPInstaller_x64.exe for the 64 bit version of Windows. The driver does not require any additional parameters.



(a) USBUART unrecognized device;



(b) MU EIS recognized by Windows, interface established;

Figure 49: The MU EIS Device in Windows Device Manager.

    - Starting from device version **MU3.3** the other type of USB-UART interface used. After connecting the device first time to PC, Windows will detect the MU EIS device as unrecognized 'USBUART' device, see Figure 49(a).

In the Device Manager with the right button click open the device settings and in manual mode show the pass to the 'USBUART_cdc.inf' file (file is included in the directory 'drivers'). After completing that the MU EIS device should appear as 'USB Serial Device (COM X)' in the Device Manager list under the 'Ports (COM&LPT)' tab, see Figure 49(b). Interface with the device established.

3. The client program does not require installation. All the necessary files are contained in MU-EIS-Client directory. The main program is 'MU-EIS-Client.exe' (_32/64 for 32/64 bit versions), the following subdirectories are used in the installation:

   the subdirectory 'data' has the data files from the device;

   the subdirectory 'images' – graphic files;

   the subdirectory 'scrips' – scripts to control the graphical output;

   the subdirectory 'log' – log-files;

   the subdirectory 'init' – files required for initialization;

   the subdirectory 'web' – files for web-based graphical output;

   the subdirectory 'drivers' – driver, libraries and installation files;

   the subdirectory 'firmware_update' – firmware update (32 and 64 bits) software.

   The client program is tested on 32 and 64 bit versions of Windows 7, 8 and 10.

4. Plotting is carried out by any program that can read numerical data from files (e.g. Microsoft Excel). The developers propose to use the free program gnuplot, however the decision to use this software lies entirely on users.

   ATTENTION. The gnuplot program is a free software, see. http://www.gnuplot.info/. This software does not belong to the MU EIS system. The user is free to choose to use or not to use this program. The latest tested versions of gnuplot are 5.2. and 6.0

   To install the gnuplot program, run the installation file 'gpxxx33-win64-mingw.exe' and install the program in the default directory, see Figure 50.

   ```
   C:\Program Files\gnuplot
   ```

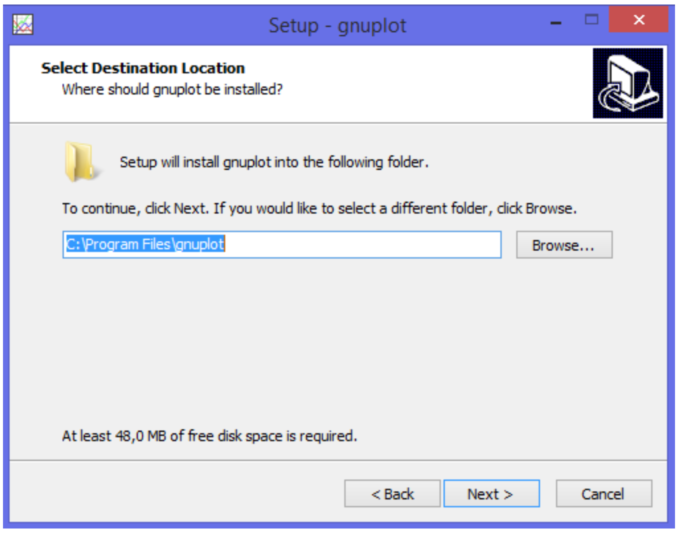


Figure 50: Directory for installing the program gnuplot.

5. Make sure that PATH environmental variable is set, see Figure 51.

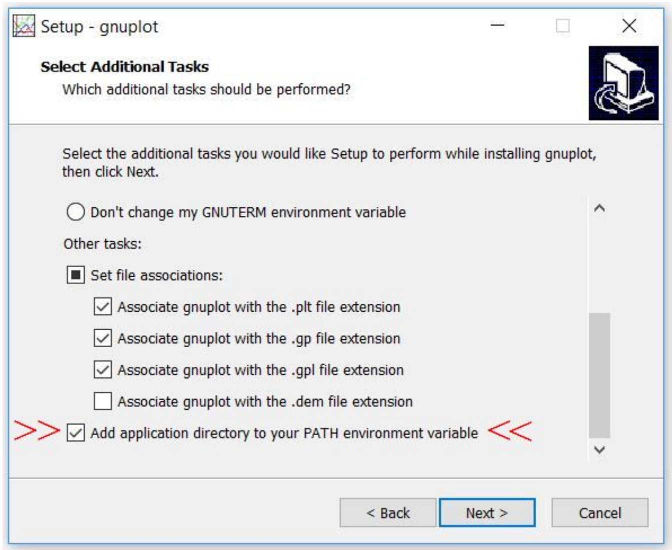


Figure 51: Setting PATH environmental variable.

If the gnuplot was installed into another directory, or the correct installation failed, you need to set the correct path and also in the PATH command, see Figure 52.

These steps need to perform only once when software is first time installed into your PC. Updates of the client program are performed only by replacing the program folder (to delete the old folder and to unpack the new version in any location).

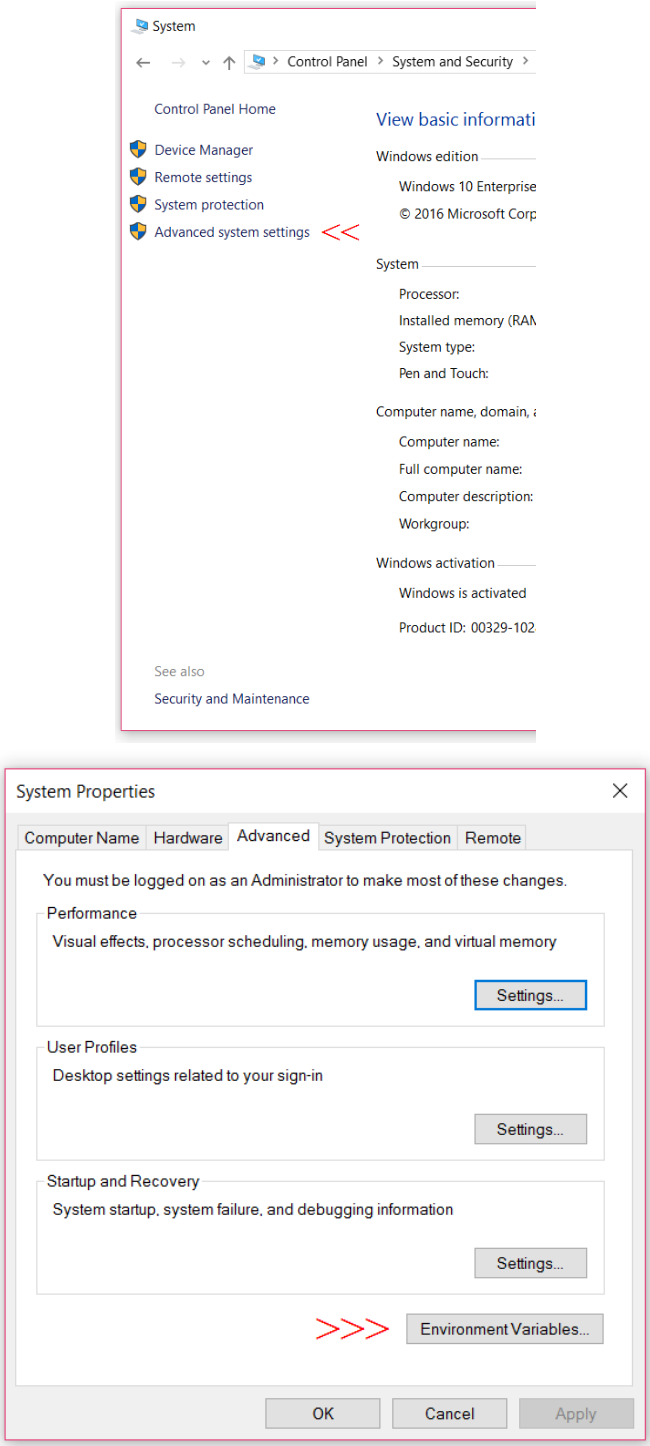


Figure 52: Setting PATH commands.

6.2 The initialization file 'init.ini'

This file is located in the directory 'init' and defines several important parameters of data management.

`1) saveAfterNSamples=N;`

This parameter defines N of data blocks received from the EIS device after that these data will be stored into a file and graphically plotted (default value N=1). Large values of N significantly reduce computational load on a host computer, however also delay displaying of measurement data.

`2) OperatingBaudRate=625000;`

This parameter defines operating baud rate for communication between EIS client program and the EIS device (default value 625000).

`3) MAXSIZEFILE=5242880;`

This defines a maximal size of data file after that a new data file will be created, in bytes (e.g. 1024 = 1 kb, 1048576 = 1 Mb, 10485760 = 10 Mb, 20971520 = 20 Mb). This parameter allows reducing computational load on a host computer related to handling large data files. New data files have the index $\_x$ (x=1,2,3,...).

`4) WEBregressionWindows=30;`

This parameter defines a time of sliding window for regression analysis in WEB plot part, defined in min., e.g. 360 min means 180 min for background measurements and 180 min for online experiment

`5) //outFileName=./data/data.txt;`

This parameter defines initial file name for online and WEB plots, initial name is "./data/data.txt". Uncomment this parameters only if it is necessary.

`6) logFileWrite=0;`

This parameter enables/diables writing into the log file from the output section of the EIS client program (1 – enabled; 0 – disabled). The log files are located in the directory 'log'.

`7) gnuplotLogFileWrite=0;`

This parameter enables/diables writing into the log file commands for the gnuplot (1 – enabled; 0 – disabled).

`8) useGnuplotTerminal=0;`

This parameter enables using the gnuplot terminal for script debugging purposes (useGnuplotTerminal=0 – disabled; useGnuplot-Terminal=1 – enabled)

`9) usingActuators=0;`

The DA module: this enables using of the detector-actuator (DA) system (1 – DA module enabled; 0 – DA module disabled)

`10) asynchronousInteractionDA=0;`

The DA module: this enables asynchronous using of external computer learning software (read Sec. 10.7)

`11) textToSpeechLanguage=en-US;`

The DA module: this defines the language used by text-to-speech (TTS) interface.

`12) DAloggingBehaviour=1;`

The DA module: this defines the data logging behaviour of DA module, see Sec. 10.1.

`13) bufferSizeDataPipes=10;`

This parameter defines the buffer size of short-term, middle-term and long-terms data pipes (all buffers will have the size defined by 'bufferSizeDataPipes'). All numerical processors and detectors (e.g. calculation of moving average, means etc.) operate over these buffers. Note, increasing the value of 'bufferSizeDataPipes' generates computational load, thus the maximal size of buffers is limited by 100, the minimal value is 10. **This value can be changed only at a new start of the client program**.

`14) usePythonPipe=1;`

This parameter enables data exchange via named pipe mechanism for e.g. external python programs (1 - use; 0 - do not use)

ATTENTION. It is recommended to disable all log files since these files can achieve a large volume and will slow down a host PC.

6.3 The client program: the section 'control'

The client program is shown in Figure 53.

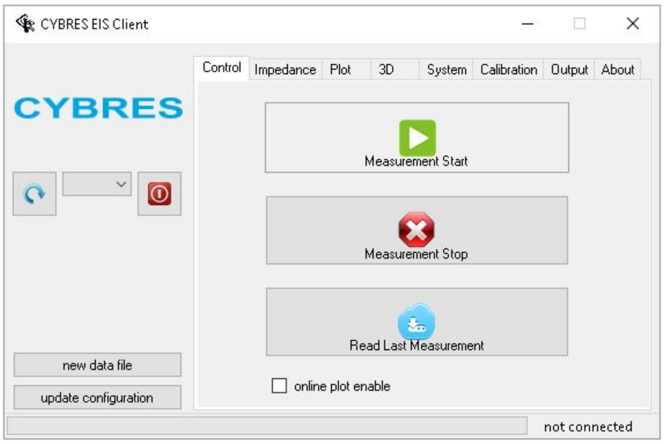


Figure 53: The client program: the section 'control'.

This program has six sections: 'control' (system control), 'impedance' (setting for the EIS measurements), 'plot' (setting for the graphical output), 'system' (system setup), 'calibration' (calibration settings), 'output' (the output window of the operating system and some interactive commands).

The 'control' section has the 'Measurement Start', 'Measurement Stop' and 'Read Last Measurement' commands. To connect the MU EIS, the com port on the right side must be selected, as shown in Figure 54(a). The device transmits the device ID number, the firmware version, temperature and thermostat status on the right panel. The message at the bottom of the window goes from 'not connected' to 'connected' and is highlighted in green.

This section has two buttons and one checkbox:

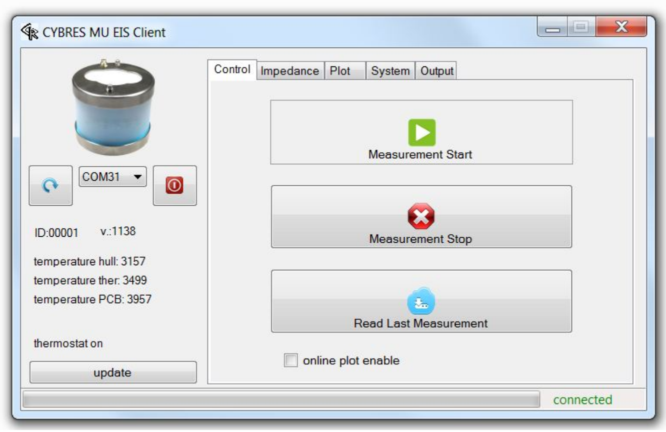
1) **the button 'update configuration'** – by pressing this button the EIS Client reads the configuration files '/init/configureDA.ini' and 'init/init.ini' (note that several changes in these files require the restart of client program) and reads the status information from the EIS device.

2) **the button 'new data file'** – by pressing this button the EIS Client create new data file that stores real time measurements from the device and is used by the plot engine. Creating new data file is required when changing the DDS operation mode due to different format of data representation as well as when the user would like to start a new plot without stopping the device (this button is available in v.1.21 and later versions).

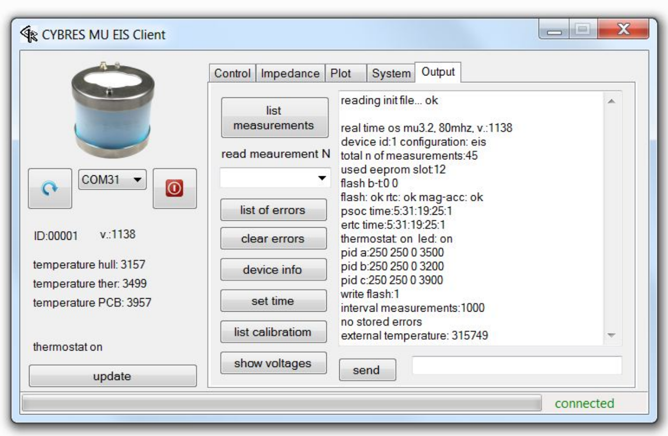
3) **the checkbox 'online plot enable'**: enable this checkbox to plot the real time data. In several cases, when the EIS client

only controls the actuators, performs data processing by numerical processors or any other activity that does not require plotting, disable this checkbox.

In the section 'output' the messages of operating system and the device status are shown, see Figure 54(b).



(a)



(b)

Figure 54: The client program: **(a)** connecting the MU EIS; **(b)** list of messages from the operating system.

The list of stored data files can be viewed and downloaded in the section 'output'. The system allows recording up to 16 measurements in the internal memory.

ATTENTION. Links to all last 16 measurements are stored in the file table, but the data from earlier measurements can be erased by later measurements. It is recommended to download the data from memory immediately after measurements.

6.4 The client program: the section 'system'

Commands in the section 'system' (see Figure 55) and 'output' configure the device. Commands are grouped into three groups: control of the LEDs and thermostats, setting the output data and some commands for operating system.

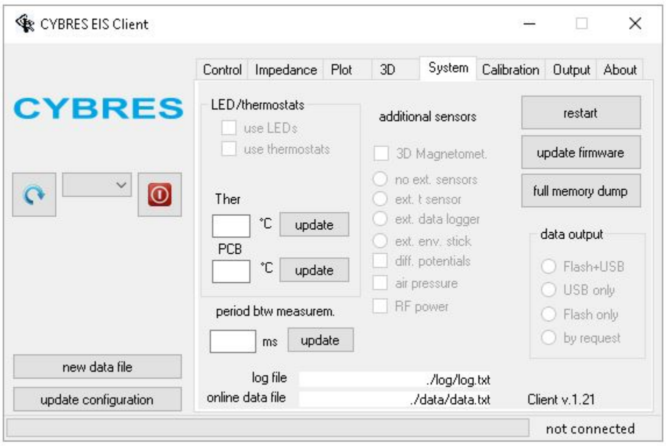


Figure 55: The client program: the section 'system'.

'Period between measurements' determines the rate of measurements in continuous modes, this value sets the period but does not affect the duration of each measurement. 'Data output' controls the output stream of data:

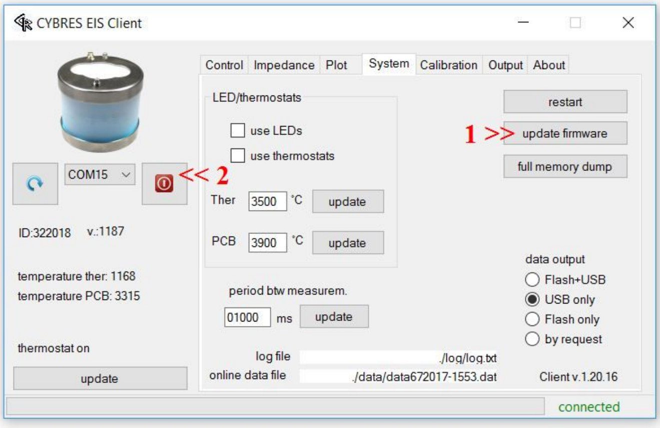
1. **Flash + USB:** data are sent to USB and stored in internal flash memory;
2. **USB only:** data are sent to USB only (it is recommended mode of operation with computer)
3. **Flash only:** data are stored in flash memory (for working in autonomous mode);
4. **by request:** data of one current measurement are sent to USB only by request from host side (when EIS operates with external control system collecting data from different devices).

6.5 Installing new firmware

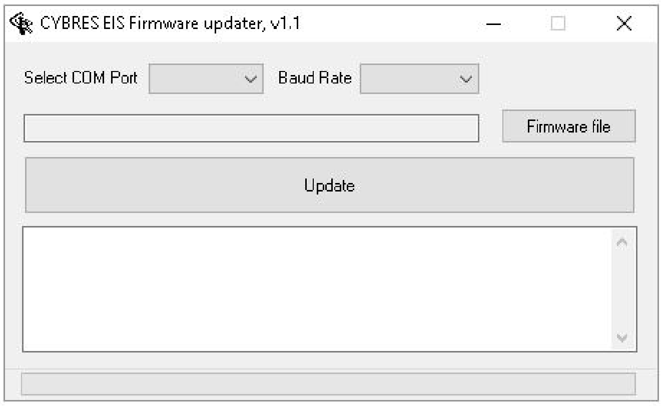
The firmware installation is similar for MU32 and MU33 systems, however it uses different programs. To install a new firmware:

1) Press the 'update firmware' in the 'system' section, see Figure 56. After this, disconnect from the device (press the button with red square near com-port field).

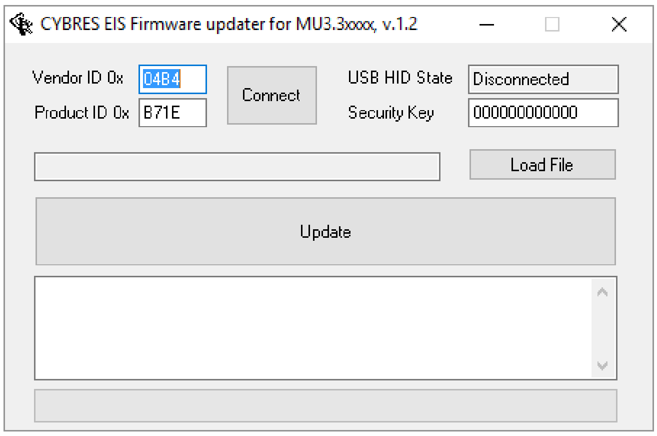
The firmware v.1186.17 (and later versions) allows entering into the update mode by turning on the power and holding the start-



(a)



(b)



(c)

Figure 56: **(a)** For installing new firmware, press the 'update firmware' and disconnect from the device; **(b) For MU32**: Start the firmware update program 'Firmware_update_M32_v12.exe' in the directory './firmware_update/MU32/x64(or x86)/'; **(c) For MU33**: Start the firmware update program 'Firmware_update_M33_v12.exe' in the directory './firmware_update/MU33/'.

stop button. When the MU EIS system is in the update mode (so-called bootloader mode), the LEDs flash with short impulses.

2) **For MU32:** start the firmware update program 'Firmware_update_M32_v12.exe' in the directory './firmware_update/MU32/x64(or x86)/', select the com port, which is connected to the device, set the baudrate at 115200 and select the firmware file. By clicking on the button 'update' a new firmware will be uploaded into the device.

For MU33: start the firmware update program 'Firmware_update_M33_v12.exe' in the directory './firmware_update/MU33/' (vendor ID 04B4 is already set), and select the firmware file. By clicking on the button 'update' a new firmware will be uploaded into the device.

3) After a new firmware is installed, it is recommended to update the system settings by pressing 'initialize variable' in the section 'calibration' (it need to enter the word 'enable' in the window), see Figure 57 and Sec. 6.6.

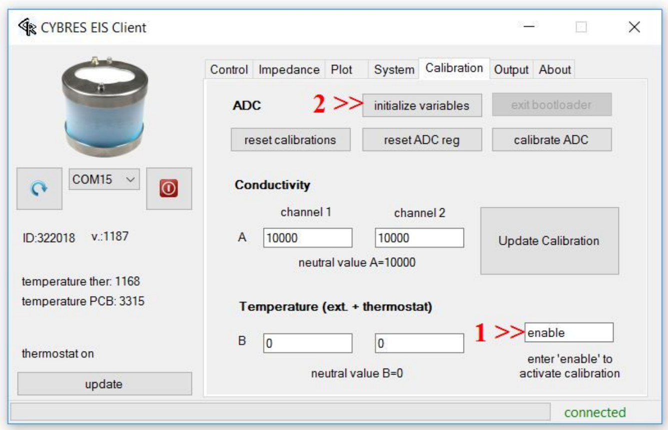


Figure 57: Initialization of variables in the section 'calibration'.

ATTENTION. It is necessary to ensure an uninterrupted power supply during firmware update. Otherwise, the device can be damaged.

6.6 Returning to initial parameters

All setting can be returned to initial parameters.

1. Write 'enable' in the section 'system', see Figure 58.
2. Press reset 'ADC reg' and wait about 1 sec.
3. Press 'initialize variables'.

4. For test purposes press 'show voltages' (1.024 should correspond to 1024xxxx value).

6.7 Self-diagnostics, error codes and functionality tests with internal resistors

The EIS spectrometer possesses several mechanisms of hardware and software self-diagnostics. Each detected failure is marked by the error code, the list of errors is accessible via client program, see Figure 59. All failures are divided into three groups: 1) strong failures related to the core functionality of the EIS device; 2) hardware failure of secondary EIS subsystems; 3) minor deviations from the expected software and hardware behavior. The third group errors can be caused by different reasons, e.g. the device was switched off during memory-writing operations and this created the memory segmentation error. Such failures are usually recovered by the EIS operating system alone. The failures of the first and second groups limit the functionality of the EIS device, it is recommended to contact the manufacturer in these cases.

Tests of EIS functionality can be performed with internal resistors on 500 Om and 5000 Om, see Figure 74. Note that the 500 Om resistor requires the amplification 500, and the resistor 5000 Om requires the amplification 5000. The measurements with internal resistors can be used for performing so-called 'full calibration', see more in Section 8.8.

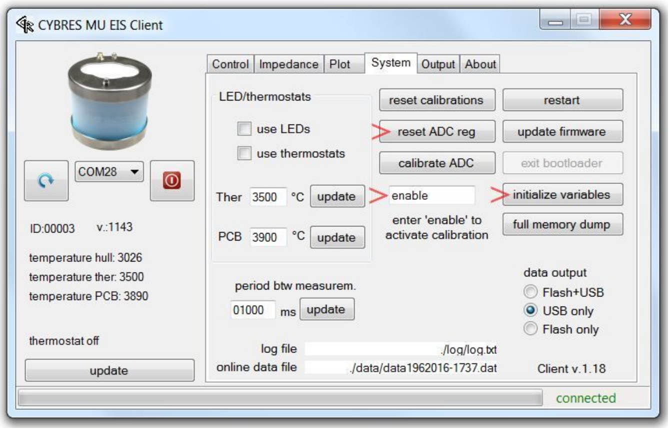
6.8 Communication with the EIS operating system

The operating system of EIS spectrometer maintains an ASCII-symbol-based communication with external devices (via comm-port, the baud rate 625 kBd). Commands and parameters can be transmitted from external devices (e.g. Raspberry Pi, Arduino, BeagleBone or any mobile/stationary devices that have comm-ports) to the spectrometer, and data from the spectrometer can be sent to these devices.

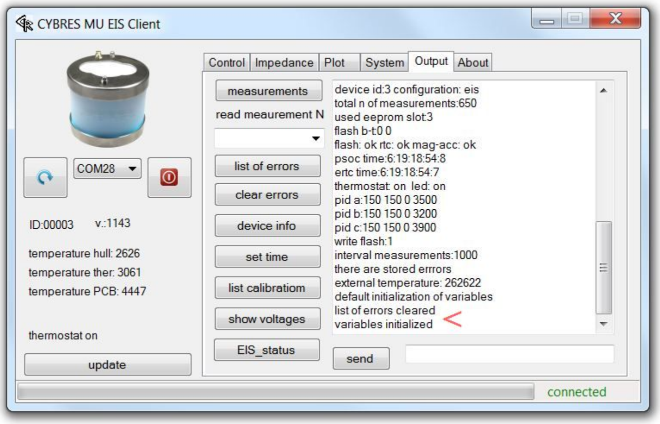
These commands and parameters have the following format

```
k1k2xxxxx
```

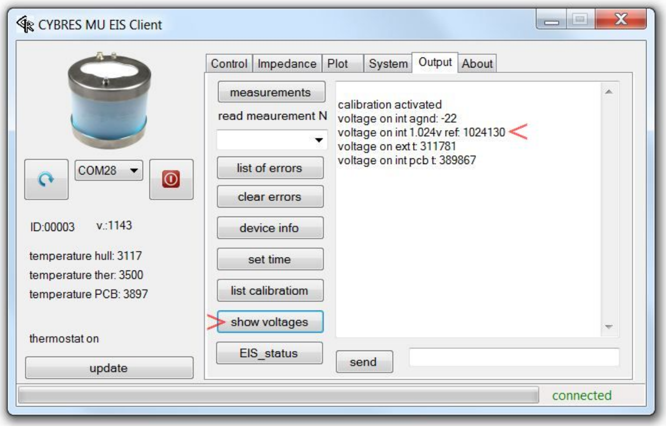
see Table 13, and can be also sent in the client program (or via any terminal program). Open the selector 'Output', the lowest field is used for sending the commands, see Figure 60. Note, after all commands, enter the '*' symbol (the asterisk symbol, without the quotes). Example: if you want to receive the system information by the command 'ss', send: ss*.



(a)



(b)



(c)

Figure 58: Returning to initial parameters.

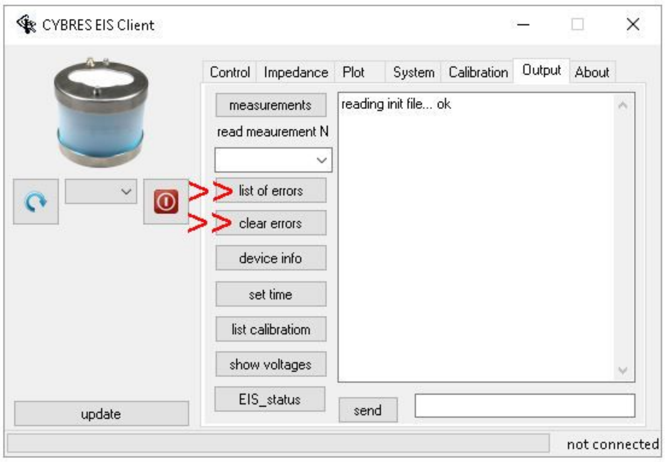


Figure 59: List of hardware and software errors.

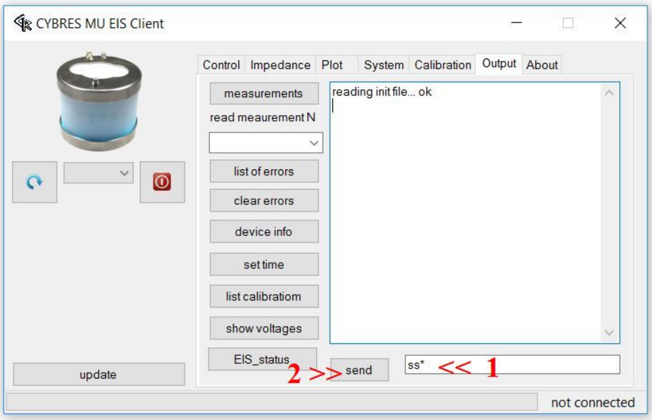


Figure 60: Sending command to operating system, example with the 'ss' command. After the command is introduced, press the button 'send'.

Table 13: List of available device commands.

k1	k2	Parameter	Response	Description
			section 'general'	
.				restart the system
,			c	reset input/output buffers of serial input
:				start bootloader mode (in order to update device firmware)
			section 'system'	
s	s			show all parameters, initial messages
s	r			restart the system
s	b			start bootloader mode (in order to update device firmware)
s	e			find the latest slot in EEPROM
s	g			show parameters stored in EEPROM
s	f	X		data printing mode. Parameter x: **0** - write data into FLASH and USB **1** - write data into USB **2** - write data into FLASH **3** - write data by request only
s	w			send list of errors
s	q			erase list of errors
s	l			list all 16 stored measurements
s	v			initialize variable with default values
s	y			send system info messages
s	z			clear list of calibrations
s	3			set test error, number 12
s	4			RGB LEDs operation enabled
s	5			RGB LEDs operation disabled
s	j			get the device ID
s	a			get measurement (work only in device by request mode)
s	c	X		turn on/off buzzer operation. Parameter **X**: **0** - turn off **1** - turn on
s	c			get buzzer operation status
s	h			check if interpreter works
s	m	X		set the device measurement configuration.Parameter **X**: **2** - EIS configuration **3** - PHY configuration **4** - ENV configuration
s	m			get the current device measurement configuration
s	k			reset the device init flag. After reset the default values of variables will be set
s	t	X		run the entire MU device test function. Parameter **X**: **1** - get result and additional info parameters **0** - get only result of test
s	p	X		enable/disable start of measurement after power on. Parameter **X**: **1** - enable auto start **0** - disable auto start
s	i	XXXXX		set the resistor in fluid t sensor (default 39k: 18C-40C; others e.g. 50k: 15C-40C; 62k: 10C-40C). Parameter **X**: **1-65534** - set the value

			0 - show the current value
s	n	X	set ADC sampling mode for fluid temperature measurements (default 0: 22 bits sampling; 1: 24 bits sampling). Parameter **X**: **1** - 24 bits sampling (slow) **0** - 22 bits sampling (default)
section 'measurement'			
m	s		start measurements
m	p		stop measurements
m	r		read last measurement data from FLASH
m	j		print all calibration data list
m	l	XX	read **XX**-th measurement from FLASH
m	i	XXXXX	set interval between measurements **XXXXX** in ms
m	q		measure and show reference values: GND, 1.024 mV and PCB temp value
m	9		measure and show temperatures values
m	b	CCXXXXXXX	set the offset calibration vector for all channels in ADC. channel - **CC**, value - **XXXXXXX**
m	v	CCXXXXXXX	set the gain calibration vector for all channels in ADC. channel - **CC**, value - **XXXXXXX**
m	n	XXXX	set the channel 1 temperature offset - only the first channel calibration
m	d		set differential temperature calibration
m	0		reset temperature calibration
m	e	X	turn on/off Potentials measurements. Parameter **X**: **0** - turn off **1** - turn on
m	e		get Potentials measurements status
m	f	X	turn on/off RF Power measurement. Parameter **X**: **0** - turn off **1** - turn on
m	f		get RF Power measurement status
m	t	X	set the mode of environmental measurement. Parameter **X**: **0** - turn off **2** - ext. temperature sensor **4** - environmental transAmb measurement stick **8** - environmental biomass stick
m	t		get environmental measurement mode
m	1 (one)	X	turn on/off Transpiration measurement. Parameter **X**: **0** - turn off **1** - turn on
m	1		get the status of Transpiration measurement
m	2	X	turn on/off Sap Flow measurement. Parameter **X**: **0** - turn off **1** - turn on
m	2		get the status of Sap Flow measurement
section 'EIS parameters'			

y	l	X		set the TIA amplification gain. Parameter **X**: **0** - gain 50 **1** - gain 50 **2** - gain 5000 **3** - gain 50000
y	w	X		set the EIS measurement channel. Parameter **X**: **0** - internal calibration 5000 Ohm Resistor **1** - internal calibration 500 Ohm (MU3.2) / 50 kOhm (MU3.3) Resistor **2** - Differential (Both) Channels **3** - Single Channel
y	j	XXX		set SAR ADC frequency clock divider
y	q	XXX		set number of analyzed periods
y	n	X		set the EIS Measurement Mode. Parameter **X**: **0** - EIS off **1** - impedance spectroscope **2** - signal scope **3** - continuous measurement (fixed f) **4** - continuous measurement (variable f) **5** - Frequency Response Profile (FRP) **6** - continuous FRP
y	x	XXX		set coefficient of the Input Low-Pass Filter (LPF)
y	g	XXXX		set coefficient of the Output Low-Pass Filter (LPF)
y	l	XXXXXX		set low frequency boundary for spectrum analysis, **Hz**
y	r	XXXXXXX		set frequency step for spectrum analysis, **Hz/10**
y	h	XXXXXX		set high frequency boundary for spectrum analysis, **Hz**
y	y	X		set the Waveform Signal Range. Parameter **X**: **1** - **±8 mV - ±1020 mV** Range **2** - **±1 mV - ±127 mV** Range **3** - **±0.12 mV - ±16 mV** Range
y	a	XXX	wave-dac amplitude: XXX	set the output voltage Waveform amplitude. Parameter **XXX**: Range **1 - 127**. One amplitude count correspond for each Signal Range to: **1 - 8 mV** **2 - 1 mV** **3 - 0.12 mV**
y	e	XXX		set the exposition time before measurement in **ms**
y	f	XXX		set the averaging level in filter
y	s	XX		set the Waveform Type. Parameter **XX**: **00** - auto (default, recommended) **01** - sinus, 1280 samples **02** - sinus, 640 samples **03** - sinus, 320 samples **04** - sinus, 160 samples **05** - sinus, 80 samples **06** - sinus, 40 samples

			07 - sinus, 20 samples **08** - sinus, 8 samples **09** - square, 4 samples **10** - constant at offset voltage
y	p		stop Waveform generator in manual mode, turn on reference level
y	m		start Waveform generator in manual mode. The all parameters set at the *'Impedance'* tab will be used
y	i		setup the initial (default) Waveform generator values
y	t		get the Waveform generator status
y	u	X	run test function for impedance measurement module. Parameter **X**: **1** - get result and additional info parameters **0** - get only result of test
y	k	X	turn on/off excitation IR LED (this function for compatibility purposes only, use instead **wiX** commands, note it does not operate with 3.3V pin). Parameter **X**: **0** - turn off **1** - turn on – **(no symbols)**: get short status
		section 'accelerometer/magnetometer'	
a	c	X	turn on/off accelerometer/magnetometer sensor operation. Parameter **X**: **0** - turn off **1** - turn on
a	c		get status of acc./mag. sensor operation
a	1		read acc./mag. sensor device ID
a	2		read temperature from acc./mag. sensor
a	3		enable read temperature from acc./mag. sensor
a	4		read magnetometer data from acc./mag. sensor
a	5		read accelerometer data from acc./mag. sensor
a	6		write control register 1 at acc./mag. sensor
a	t		run test function of acc./mag. sensor
		section 'date and time', periodical timers	
t		YYMMDDHHMMSS	set current time value. Format **YYMMDDHHMMSS**: **YY** - two digit year value (ex. 2017: **17**) **MM** - month value (January - first month: **01**) **DD** - day of the month value **HH** - hours value **MM** - minutes value **SS** - seconds value
t		MMDDHHMMSS	set current time value. Format **MMDDHHMMSS**. Without year value.
t	1		get current device RTC time
t	r		get current external RTC time
t	s		set the current time to external RTC from device RTC

t	p		set the current time to device RTC from external RTC
t	y	XXXX	set the current year value. **XXXX** - 4-digit year value
t	t		run the test function for external RTC
t	q	xHHMMSS	set 'on'-time for 24h-periodical timer, x – is the number of timer, **note**: 1) timers run independently of the 'start'-'stop' of measurements; 2) 24h-mode of the timer 1 starts if 'on'-/'off'-times are set and 'on'-/'off'-periods are zero; 3) MOSFET of LEDs are used to control external solid state relay; 4) timers can interfere with other LED functions (make sure to off all other functions); 5) minimal time can be >1 sec.; 6) be careful with 'this day'/'next day' conditions for 24h timer
t	w	xHHMMSS	set 'off'-time for 24h-periodical timer, x – is the number of timer
t	z	xSSSSSS	set 'on'-period in sec. for periodical timer, x – is the number of timer, **note**: 1) periodical mode has higher priority than 24h mode; 2) if both 'on'-/'off-' times in 24h mode are set, the timer operates only within the specified time period, otherwise it starts immediately and counts endless; 3) the value SSSSSS is limited by 65535 sec. (=18.2 hours)
t	u	xSSSSSS	set 'off'-period in sec. for periodical timer, x – is the number of timer
t	a	xY	set the output to be processed by the timer, x – is the number of timer, Y – the output channels: **1** - r output **2** - g output **3** - b output **4** - ts1 output **5** - ts2 output **6** - 3.3V output
t	o	xY	start/stop the timers (x is the number of timer, e.g. **to11*** – turn on the timer 1), parameter Y: **0** - turn off **1** - turn on – **(no symbols for Y)**: get status, e.g. **to1*** – get status of the timer 1 – **(no symbols for x and Y)**: get status for all timers, e.g. **to***
t	l	xY	set ON/OFF log for timers (x is the number of timer, e.g. **to11*** – turn on the timer 1), parameter Y: **0** - turn off **1** - turn on – **(no symbols for Y)**: set all times, e.g. **tl1*** – set ON for all timers

				– **(no symbols for x and Y)**: get status, e.g. **tl*** – get status of all timers
t	m			reset parameters of all timers (this is used to disable the whole timer functionality)
section 'PID controllers'				
PID A controller - External EIS Thermostat **PID C controller** - PCB Thermostat				
p	o		thermostat enabled	thermostats enabled
p	e		thermostat disabled	thermostats disabled
p	p	XXXX	pid-a kp coeff: XXXX	set proportional coefficient for PID A controller
p	i	XXXX	pid-a ki coeff: XXXX	set integral coefficient for PID A controller
p	d	XXXX	pid-a kd coeff: XXXX	set differential coefficient for PID A controller
p	t	XXXX	pid-a goal t: XXXX	set goal temperature for PID A controller
p	g		get all PID parameters	
p	r		set default PID parameters	
c	p	XXXX	pid-c kp coeff: XXXX	set proportional coefficient for PID C controller
c	i	XXXX	pid-c ki coeff: XXXX	set integral coefficient for PID C controller
c	d	XXXX	pid-c kd coeff: XXXX	set differential coefficient for PID C controller
c	t	XXXX	pid-c goal t: XXXX	set goal temperature for PID C controller
section 'flash'				
f	i			get full ID of FLASH
f	c			test FLASH device
f	s		s	read status register
f	f		f	read flag register
f	q		q	write enable
f	d		d	write disable
f	e		e	erase die 0 (all data will be lost)
f	t		t	erase die 1 (all data will be lost)
f	b		b	FLASH restart
f	r			FLASH read from current address page
f	k			FLASH read from zero page
f	g			read FLASH non-volatile register
f	h			write FLASH non-volatile register 4 byte mode
f	1		dump...	read all data in FLASH
f	2		find free block	FLASH defragmented: find next free block to write data
f	0			show SPI devices chip selects control bits
section 'sensors' and 'controllers' (pressure, sap flow, soil, etc.)				
h	a			**pressure sensor** initialize function
h	b			perform test pressure sensor function
h	c			pressure sensor de-initialize function
h	d			get pressure value
h	e			get temperature value
h	f			get pressure and temperature value

h	g	X	turn on/off pressure sensor operation. Parameter **X**: **0** - turn off **1** - turn on
h	g		get the status of pressure sensor operation
h	s	X	**thermal sap flow sensor** **0** - initialization **1** - print parameters **2Y** - turn on/off thermostat pulse mode **3** - set default parameters **4Y** - prepare or stop measurements **5** - set default parameters
h	u	XXX	set the sap flow sensor heater duty PWM
h	t	XXXXX-YYYYY	set the parameters of sap flow sensor for pulse heater mode
h	y	YX	**green/root biomass and analog soil moisture sensors**, Y – N of sensors (1,2,3): **1** - soil or root biomass sensors **2** - biomass sensor 1 (it includes environmental sensors) **3** - biomass sensor 2 (without environmental sensors), X: **0** - on **1** - off
h	y		**(only hy*)**: get status of biomass and analog soil moisture sensors
h	y	Z	electric field excitation for all green/root biomass and analog soil moisture sensors, Z: **0** - periodical excitation on (regular mode, duration of excitation before measurement about 200ms) **1** - periodical excitation off (test mode, excitation does not stop between measurements)
h	o	X	**soil moisture sensor on I2C bus** **0** - initialization **1** - rebooting **2** - reset **3** - get version **4** - get moisture **5** - get temperature **6** - get light **9** - set slip mode
h	k	YX	**embedded controllers** (EmCons). Note that all EmCon operate only in measurement mode over real-time data, Y – N of controller (1,2,3,...), X: **0** - on **1** - off – **(YX: no symbols)**: get status
h	l	YXXXXXX	set the compare value for EmCons, Y – N of controller (1,2,3,...), XXXXXX – 6 digits value. Note that EmCons compare original raw data without conversion to output formats, use the Client program for automatic conversions

h	m	YXX	set the input channel for EmCons, Y – N of controller (1,2,3,...), XX – 2 digits value (e.g. 25, 26, 27, 34), note that not all channels are available for EmCons
h	n	YX	set the output channel for EmCons, Y – N of controller (1,2,3,...), X: **0** - timer 1 **1** - timer 2 **2** - timer 3 **3** - R channel **4** - G channel **5** - B channel **6** - ts1 channel **7** - ts2 channel **8** - 3.3v channel
h	q	YX	set the operation for EmCons, Y – N of controller (1,2,3,...), X: **0** - ’>’ **1** - ’<’ **2** - ’=’
section ’output channels’ and ’LED, sound indicators’			
w	k	ABC	turn on/off RGB LEDs, this function does not store values in EEPROM (use for frequent calls). **A** - Red; **B** - Green; **C** - Blue. Parameter **ABC**: **0** - turn off **1** - turn on
w	l	X	turn on/off **R channel** (Red LED). Parameter **X**: **0** - turn off (not stored in EEPROM, use for frequent calls) **1** - turn on (not stored in EEPROM, use for frequent calls) **0x** - turn off (stored in EEPROM, x – any symbol) **1x** - turn on (stored in EEPROM, x – any symbol) – **(no symbols)**: get short status
w	m	X	turn on/off **G channel** (Green LED). Parameter **X** is the same as in R channel.
w	n	X	turn on/off **B channel** (Blue LED). Parameter **X** is the same as in R channel.
w	q	X	turn on/off **ts1** channel. Parameter **X** is the same as in R channel.
w	i	X	turn on/off **ts2** channel (e.g. excitation by IR LED). Parameter **X** is the same as in R channel.
w	v	X	turn on/off **3.3V** output, note this output is used for powering front green LED, external LEDs and external thermostats, it is automatically ON if ’use LEDs’ and ’use thermostats’ are set. Parameter **X** is the same as in R channel.

w	p	X	set PWM mode for **PWM R channel** (Red LED). This function is stored in EEPROM. It does not turn ON/OFF the output (use turn on/off function for this). Parameter **X**: **0** - PWM mode off **1** - PWM mode on – **(no symbols)**: get PWM status
w	s	X	set PWM mode for **PWM G channel** (Green LED). This function is stored in EEPROM. It does not turn ON/OFF the output (use turn on/off function for this). Parameter **X** is the same as in PWM R channel.
w	h	X	set PWM mode for **PWM B channel** (Blue LED). This function is stored in EEPROM. It does not turn ON/OFF the output (use turn on/off function for this). Parameter **X** is the same as in PWM R channel.
w	t	XXXXXXXX-YYY	set the PWM parameters for **PWM R channel** (Red LED), see notes in Sec. 4.8.1. **XXXXXXXX** - PWM Frequency in Hz, [367 .. 12000000]. **YYY** - PWM Duty Cycle in %, [0 .. 100]
w	u	XXXXXXXX-YYY	set the PWM parameters for **PWM G channel** (Green LED), see notes in Sec. 4.8.1. **XXXXXXXX** - PWM Frequency in Hz, [367 .. 12000000]. **YYY** - PWM Duty Cycle in %, [0 .. 100]
w	g	XXXXXXXX-YYY	set the PWM parameters of HIGH Frequency for **PWM B channel**, see notes in Sec. 4.8.1. **XXXXXXXX** - PWM Frequency in Hz, [367 .. 12000000]. **YYY** - PWM Duty Cycle in %, [0 .. 100]
w	f	XXXXX-YYY	set the PWM parameters of LOW Frequency for **PWM B channel**, see notes in Sec. 4.8.1. **XXXXX** - PWM low frequency multiplied by 100 in range [0.015259.. 500 Hz], e.g. the value 50000 correspond to 500Hz . **YYY** - PWM Duty Cycle in %, [0 .. 100]
w	j	X	turn on/off LED RGB B (Blue) Channel with preparation of external powering (it is used compatibility only). Parameter **X**: **0** - turn off **1** - turn on
w	b	X	turn on/off front RED LEDs. Use the command **wvX** to turn on/off green front LED. Note that this command turns on/off the second internal 3.3V power source. Parameter **X**: **0** - turn off

			1 - turn on
w	c	XY	This function uses the PWM controller connected to 3.3V outputs and front LEDs. In the firmware version 1189.49 this functionality is reserved for future applications, it implements only one-step 'blinking'. **X** - Green LED; **Y** - Red LED. Parameter **XY**: **0** - LED don't blink **1** - LED blink
w	d	X	test Buzzer. Parameter **X**: **1** - Normal Beep **2** - Short Beep **3** - Long Beep **4** - Long Short Beep – **(no symbols)**: get status

Table 14: Device return parameters to response for **ss** and **sy** commands.

k1	Return parameter	Description
I		begin marker, each system message should start with it
D	XXXXX	Device ID
V	XXXX	firmware version
F	X	flash write parameter
P	XXXXX	time period between measurements, **ms**
O	XXXX	goal temperature of PID A, $^{\circ}C \cdot 100$
C	XXXX	goal temperature of PID B, $^{\circ}C \cdot 100$
S	XXXX	goal temperature of PID C, $^{\circ}C \cdot 100$
H	X	thermostat status, on/off
Q	XXXX	temperature thermostat A, $^{\circ}C \cdot 100$
W	XXXX	temperature thermostat B, $^{\circ}C \cdot 100$
U	XXXX	temperature thermostat C, $^{\circ}C \cdot 100$
G	X	RGB LEDs status, on/off
E	X	waveform generator signal range
N	XXX	waveform generator signal amplitude
J	XXXX	coefficient of the Output Low-Pass Filter (LPF)
K	XXXXXX	low frequency boundary for spectrum analysis, Hz
L	XXXXXX	high frequency boundary for spectrum analysis, Hz
)	XXXXXXX	frequency step for spectrum analysis, Hz·10
B	XXX	the averaging level in filter
(	XX	waveform type
!	X	EIS measurement mode
”	XXX	number of analyzed periods
$	X	TIA amplification gain
%	X	EIS measurement channel
&	XXX	coefficient of the Input Low-Pass Filter (LPF)
[	XXXXX	additive calibration vector Ch. 1 EIS
]	XXXXX	additive calibration vector Ch. 2 EIS

{	XXXXX	multiplicative calibration vector Ch. 1 EIS
}	XXXXX	multiplicative calibration vector Ch. 2 EIS
$\sim$	XXXX	additional sensor parameters. Format (0 - turn off, 1 - turn on): **Bit 0** - accelerometer measurement **Bit 1-3** - environmen. measurements: **000** - off **001** - external temperature sensor only **010** - environmental measurement via the transAmb stick **100** - environmental data from the biomass sensor **Bit 4** - potential measurement **Bit 5** - air pressure measurement **Bit 6** - RF power measurement **Bit 7-9** - sensors on the I2C bus **000** - off **001** - I2C soil sensor (stick with RGB light ball) **010** - reserved **100** - reserved **Bit 10-12** - analog biomass sensors **000** - off **001** - soil/root biomass sensor **010** - biomass sensor 1 **100** - biomass sensor 2 **Bit 13-15** - reserved
Y		end marker, each system message should end up with it
A		Start of measurement data
Z		End of measurement data

Table 15: List of Device errors.

Error Number	Description
1	flash write not enabled
2	flash write fail
3	flash address is not %ff
4	flash block is not empty
5	flash initialization fail
6	acc/mag sensor initialization fail
7	RTC (Real Time Clock) initialization fail
10	caution: FLASH die 0 reset
11	caution: FLASH die 1 reset
12	caution: test data
13	caution: Thermostat is too low
14	caution: PCB thermostat is too low
15	caution: ADC doesn't calibrated
16	caution: soil i2c sensor initialization fail
17	caution: depth i2c sensor initialization fail
18	pressure sensor initialization failed
19	device ID corrupted, contact manufacturer!
20	uart-usb sending message failed, timeout
21	DelSig ADC measure data failed, timeout

22	time date problem, internal time not equal to external RTC time

7 Graphical output

7.1 Graphical output with the Microsoft Excel™

The program Microsoft[13] Excel™can be used for building graphics from the data, produced by the MU EIS. Start the Microsoft Excel, import the data file (see meaning of numerical fields in Section 7.8), select different graphical possibilities provide by this program.

7.2 Graphical output with the Gnuplot

The MU EIS provides support for gnuplot as a graphical engine via the pipe mechanism (StdIn is switched to the EIS client). StdOut output of gnuplot is set to external terminal and by using 'useGnuplotTerminal=1;' is available for debugging purposes. Gnuplot scripts are prepared for building various graphics, see the following sections for more details. It is also possible to start several gnuplot graphs in parallel for indication of different parameters in real time.

7.3 Graphical output with other software packages

Any software package can be used for plotting graphics from the ASCII data, provided by the MU EIS. Follow instruction to these software packages.

7.4 Online and Offline graphs

The device has the ability to plot data during the measurement (online mode) and after the measurement (offline mode). Offline mode allows operating the MU EIS without a computer. The device records all data into the internal non-volatile memory and graphics are built after the measurement. In the online mode, the device must be connected to a computer, the data are transmitted to the computer during the time of measurement. Selection of the operating mode takes place in the client program, in the section 'system', the field 'data output'.

ATTENTION. Operating the 'start-stop' button on the MU EIS does not change the settings of the 'data output' field. Make sure that the device has a correct setting for the output data.

7.5 The client program: the section 'plot'

Plotting data should be located on the HDD of a computer:

1. Start the client program and select the section 'plot', see Figure 61.

[13] Registered trade mark belongs to Microsoft Corporation.

2. Open the file, select the necessary options for graphical output. The start and end of measurement are displayed in the appropriate fields. Fields ’impact begin’ and ’impact end’ are necessary for impact analysis – these fields should be filled up manually. Graphics can be entitled by a name in the ’title’. These settings can be saved by clicking on the button with the floppy disk icon.

3. Select the desired type of graphical output from the list of ’graph to plot’, see Table 16, choose the level of filtering, and output device (terminal, png or eps files). Click on the ’plot’ button.

4. The check-box ’differential values’ turn on or off the calculation of differential values from both channels. All changes in settings are immediately visible on graphs.

5. To close the external graphical console, press the button ’close plot’.

ATTENTION. Opening data file create a graphical console. This console remains visible when a new data file is opened, however the client program has assess only to the last opened graphical console.

7.6 Downloading data from the device

To download data from internal flash to HDD:

1. Connect the MU EIS as described above.

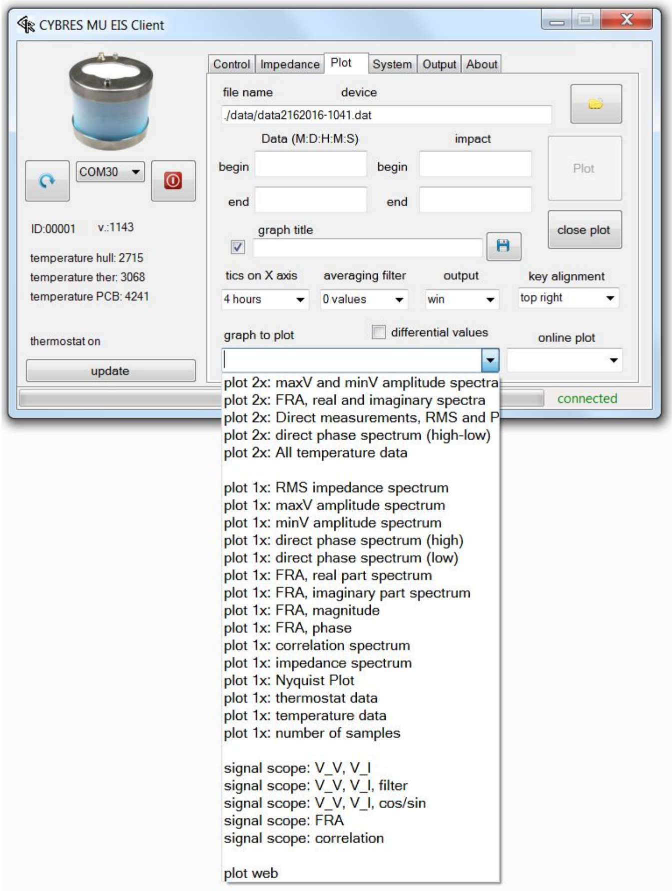
2. In the client program select the section ’control’ or ’output’. To operate in the offline mode, the ’data output’ field should be set in a mode ’Flash + USB’ or ’Flash’!

3. Download the data of the last measurement (the section ’control’, the button ’Read Last Measurement’). The unit also stores the last 16 measurements, the date of recording and file size. They can be seen by clicking ’list measurements’ in the section ’output’, as shown in Figure 62. These data can be downloaded from the device by selecting the number of measurements of the field ’read measurement N’. Downloaded data are stored in the subdirectory ’data’.

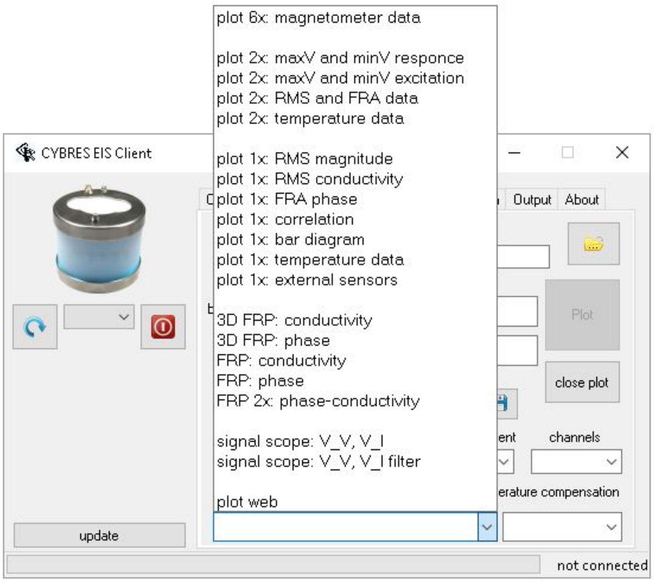
7.7 Graphical output in the online mode

For the construction of online charts ’data output’ field should be located in a mode Flash + USB or USB!

1. Connect the USB connector to the MU EIS device (before measurement).



(a)



(b)

Figure 61: The client program: the section 'plot': **(a)** the firmware v. <1187.x; **(b)** the firmware v. >1187.x.

Table 16: Table of available types of graphs (with gnuplot program), the firmware v. <1187.x.

Plot	
plot 6x: magnetometer data	six graphs, 3D magnetometer and accelerometer data
plot 2x: maxV and minV amplitude spectra	two graphs, see description below
plot 2x: FRA, real and imaginary spectra	two graphs, see description below
plot 2x: Direct measurements, RMS and Phase spectra	two graphs, see description below
plot 2x: All temperature data	two graphs, see description below
plot 1x: RMS magnitude	based on the RMS values as $Z^{RMS} = \frac{V_V^{RMS}}{V_I^{RMS}}$ for all frequencies f.
plot 1x: maxV amplitude	maximal detected amplitude of the V_I, V_V signals, measured in V. Its value should be in the range between 0V and +1V.
plot 1x: minV amplitude	minimal detected amplitude of the V_I, V_V signals, measured in V. Its value should be in the range between -1V and 0V.
plot 1x: direct phase spectrum	phase shift between V_V and V_I signals, measured in degrees related to signal period. V_V and V_I are shifted by $\pm 180^\circ$ to each other.
plot 1x: FRA, real part	values of $Re^{FRA}(V_I)$ for all frequencies f.
plot 1x: FRA, imaginary part	values of $Im^{FRA}(V_I)$ for all frequencies f.
plot 1x: FRA magnitude	magnitude of $Re^{FRA}(V_I)$ and $Im^{FRA}(V_I)$, calculated as $\sqrt{Re^{FRA}(V_I)^2 + Im^{FRA}(V_I)^2}$
plot 1x: FRA phase	phase of $Re^{FRA}(V_I)$ and $Im^{FRA}(V_I)$, calculated as $\tan^{-1}(\frac{Im^{FRA}(V_I)}{Re^{FRA}(V_I)})$.
plot 1x: correlation	correlation between V_V and V_I signals calculated as $\frac{1}{N}\sum_{k=0}^{N-1} \tilde{V}_I^f(k)\tilde{V}_V^f(k)$, where N is the number of samples inside a period, $\tilde{V} = V - 2047$. This value works for harmonic as well as nonharmonic driving signals.
plot 1x: FRA conductivity	calculated as 1/magnitude.
plot 1x: Nyquist Plot	relation between real and imaginary parts of impedance calculated by FRA.
plot 1x: temperature data	the temperature of sample and PCB thermostats during measurements.
plot 1x: number of samples	the number of data samples in the stored signal period.
signal scope: V_V, V_I	signal scope mode for excitation V_V and response V_I signals.
signal scope: V_V, V_I, filter	signal scope mode for excitation, response and low-pass signals.
signal scope: V_V, V_I, cos/sin	excitation V_V and response V_I and basic sin() and cos() vectors for FRA analysis.
signal scope: FRA	FRA data $V(k)\left[\cos(\frac{2\pi fk}{N}) - i\sin(\frac{2\pi fk}{N})\right]$ separate for $Re^{FRA}(V_I)$ and $Im^{FRA}(V_I)$
signal scope: Correlation	(see description for correlation spectrum).
plot web	reserved for the web-based plot

Table 17: Table of available types of graphs (with gnuplot program), the firmware v. >1187.x.

Plot	
plot 6x: magnetometer data	six graphs, 3D magnetometer and accelerometer data
plot 2x: maxV and minV amplitudes of response signals	two graphs, maximal detected amplitude of the V_I signals, measured in V, the range ± 1V
plot 2x: maxV and minV amplitudes of excitation signals	two graphs, maximal detected amplitude of the V_V signals, measured in V, the range ± 1V
plot 2x: RMS magnitude and FRA phase	two graphs, see description below
plot 2x: Temperature and Impedance data	two graphs, see description below
plot 1x: RMS magnitude	calculated based on the RMS values as $Z^{RMS} = \frac{V_V^{RMS}}{V_I^{RMS}}$, harmonic and nonharmonic signals
plot 1x: RMS conductivity	calculated based on the $1/Z^{RMS}$, harmonic and nonharmonic signals
plot 1x: FRA direct phase	phase shift between V_V and V_I (harmonic signals only), measured in degrees related to signal period, calculated as lock-in detector based on $\cos^{-1}(V_I^f V_I^f)$, measured in degrees related to signal period, V_V and V_I are originally shifted by 90° to each other
plot 1x: correlation spectrum	correlation between V_V and V_I signals calculated as $\frac{1}{N}\sum_{k=0}^{N-1}(V_I^f(k)V_V^f(k))$, where N is the number of samples inside a period, harmonic and nonharmonic signals
plot 1x: bar diagram	shows the difference between initial and 'last values' of impedance as a bar, the position of a last value can be selected in the field 'plot values', this is useful in continuous mode to show the rate of changes between channels
plot 1x: temperature data	the temperature of sample and PCB thermostats during measurements
plot 1x: external sensors data	date from high-resolution external sensors connected to the device
4D plot: magnitude	continuous FRP mode only, it shows the frequency response in time, Z axis is the value of impedance, the heat axis – the value of phase
4D plot: phase	continuous FRP mode only, it shows the frequency response in time, Z axis is the value of phase, the heat axis – the value of impedance
heat map: magnitude	continuous FRP mode only, variation of magnitude in time and frequency
heat map: phase	continuous FRP mode only, variation of phase in time and frequency
2x heat map: magnitude & phase	continuous FRP mode only, variation of magnutude/phase in time and frequency
signal scope: V_V, V_I	signal scope mode for excitation V_V and response V_I signals.
signal scope: V_V, V_I, filter	signal scope mode for excitation, response and low-pass signals.
plot web	reserved for the web-based plot

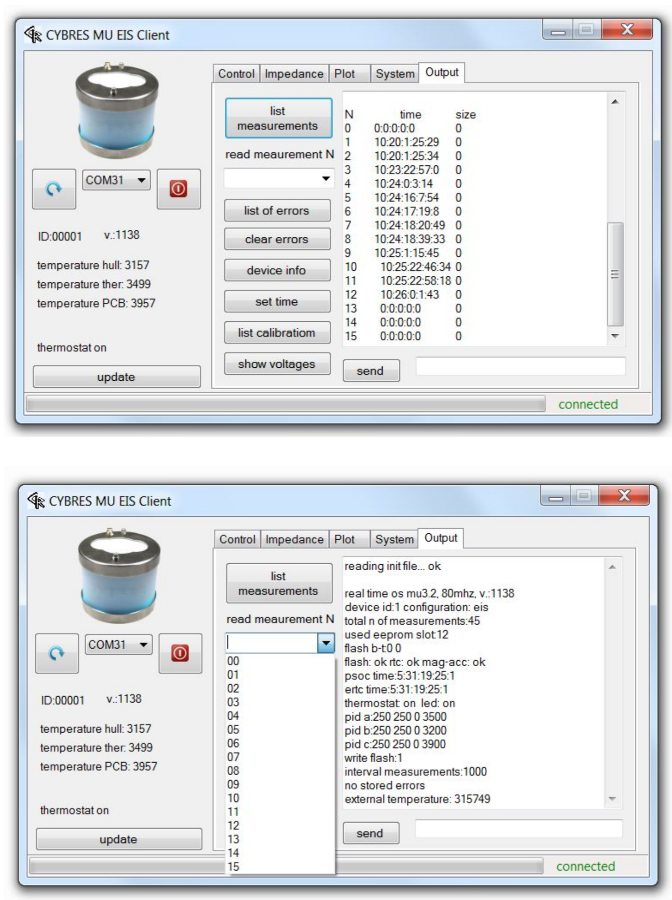


Figure 62: List of 16 measurements stored in the MU EIS.

2. Select the com port in the client program and to connect the MU EIS device. Make sure that 'online plot enable' in the section 'control' is enabled, see Figure 53).

3. Select the desired type of plot, see the table 16, select the number of measurement points to be displayed (all of the last N measurements) and click 'Start Measurement'. Plot of the last N measurement works correctly only if the new plot was launched since the beginning of measurements.

ATTENTION. The file name, where the data are recorded in the online mode, indicated in the section 'system'. This data file may also be analyzed using different tools for numeric analysis.

7.8 Fields of data files in continuous and spectral measurement modes

The main data file (obtained in the DDS mode 'impedance spectroscope' and in all continuous modes) is located in the 'data' directory. The file has the name '**dataDDMMYYYY-Hm.dat**', where DD - day, MM - month, YYYY - year, H - hour, m - minutes. Thus, it is possible to uniquely identify these files. This file has 'append' mode, i.e. each new measurement is added to previous ones. The maximal size of data files is limited by the variable MAXSIZEFILE in the configuration file 'init/init.ini'. If the current file is larger than the limit, a new file is created '**dataDDMMYYYY-Hm_ZZ.dat**', where ZZ – is an iterative number of a new file. Note, that 25% last content of the old file is transference to a new data file to keep a continuous data flow (e.g. for the long-term regression analysis).

Each data file has the record

```
#
#ID Xxxxx firmware version FV yyyy
current data CD DD.MM.YYYY HH: mm: ss
#
```

where xxxxx - device ID of the measurement, yyyy - the version number of the firmware, and the recording date in the format DD.MM.YYYY HH:mm:ss (ss – seconds). The measurement start and ends are indicated by fields $\#ms$ and $\#me$ in the format MM:DD:HH:mm:ss.

The data in the file are arranged in 45 data columns (before the firmware v1189 – into 33 data columns) with spaces between the columns. Each line represents the result of a single measurement with time stamp.

1: Time stamp of each measurement in the form YY.MM.DD.HH.mm.ss;

2: Frequency of the sweep. This data field is multiplied by 10, i.e. 11011 means 1101.1Hz;

Channel 1 data

3: VImax – values of maximal amplitude (upper peak) of the V_I signal;

4: VImin – values of maximal amplitude (lower peak) of the V_I signal;

5: RMS – the magnitude calculated based on the RMS values;

6: Phas – values of the phase shift between V_V and V_I signals;

7: VVmax – values of maximal amplitude (upper peak) of the V_V signal;

8: VVmin – values of maximal amplitude (lower peak) of the V_V signal;

9: Corr – correlation between V_V and V_I signals for the sweep frequency;

Channel 2 data

10:VImax 11:VImin 12:RMS 13:Phas 14:VVmax 15:VVmin 16:Corr

System data

17: t-PCB – temperature of the PCB

18: t-thermost – temperature of the thermostat, e.g. 26.234C is defined as 262340

Magnetometer and accelerometer data

19, 20, 21 – magnetometer data on axes X, Y, Z

22, 23, 24 – accelerometer data on axes X, Y, Z

External sensors

25: external temperature, e.g. 26.234C is defined as 262340 (note that different t-sensors represent their data in different format, see description of sensors) (with the sensor data logger)

26: external light (with the sensor data logger)

27: external humidity (with the sensor data logger)

28: differential potential, channel 1 (with the phytosensor)

29: differential potential, channel 2 (with the phytosensor)

30: RF power emission

31: transpiration sensor data (with the phytosensor electrodes advanced)

32: sap flow sensor data (with the phytosensor electrodes advanced) or coded temperature of fluids (t-ch1 t-ch2)

33: air pressure

34: soil moisture with analog or digital I2C sensors (both sensors are differently calibrated)

35: soil temperature (with digital I2C soil sensor) or biomass 1 sensor

36: biomass 2 sensor

37-43: empty (reserved for different I2C sensors)

44,45: reserved for statistical package/DA module to encode the temperature of fluids (t-ch1, t-ch2)

The statistical package 'EIS statistics' represents a kind of synthetic (virtual) sensor for measuring electrochemical noise that is available only in the EIS mode (it can be turned on/off by users) and that writes 24 data channels into the main stream (channels 46-69):

46, 47: accumulated variation of impedance (ch1, ch2) 48, 49: accumulated variation of correlation (ch1, ch2) 50, 51: accumulated variation of phase (ch1, ch2) 52, 53: accumulated variation of temperature (ch1, ch2)

54, 55: accumulated skewness of impedance (ch1, ch2) 56, 57: accumulated skewness of correlation (ch1, ch2) 58, 59: accumulated skewness of phase (ch1, ch2) 60, 61: accumulated skewness of temperature (ch1, ch2)

62, 63: accumulated kurtosis of impedance (ch1, ch2) 64, 65: accumulated kurtosis of correlation (ch1, ch2) 66, 67: accumulated kurtosis of phase (ch1, ch2) 68, 69: accumulated kurtosis of temperature (ch1, ch2)

Note, depending on the enabled or disabled 'EIS statistics' sensor, the next available data channel is 46 or 70 (the maximal number of data channels is 80).

The device writes the header with a short notation for the used data fields:

#mb 19:10:05:16:01:35 (start time)

#id 332033 (device id)

#ta 3214 (tia amplification, tia input channel, signal range, dsp mode)

#1t 2f [3vi_ma 4vi_mi 5mag 6pha 7vv_ma 8vv_mi 9cor] [10vi_ma 11vi_mi 12mag 13pha 14vv_ma 15vv_mi 16cor] [17t_pcb 18t_ther] [19,20,21mag] [22,23,24acc] [25,26,27ext(t,light,hum)] [28dv1 29dv2

30rf 31tr 32sp 33pr][34sm 35st 36lght 37empt 38empt 39empt 40empt 41empt 42empt 43empt 44empt 45empt]

The device ID and the used mode of measurements is written into the sensor data each 100 samples. Data columns >45 are used for statistical calculations and data management by the DA module, and are flexible (i.e. programmable by users). Totally, 80 data channels are possible in the main file.

7.9 Fields in the signal scope data file

The signal scope data file (obtained in the DDS mode 'signal scope') is located in the 'data' directory. The file has the name '**dataDDMMYYYY-Hm_sig.dat**', where DD - day, MM - month, YYYY - year, H - hour, m - minutes. Thus it is possible to uniquely identify files. This file has 'replace' mode, i.e. each new measurement replaces the previous ones.

The data in the file are arranged in several columns with a single space between the columns. Each line represents the result of a single measurement. The data, frequency and other parameters are indicated in the file before measurement data.

```
1 2 3 4 5 (other fields are used for statistical calculations)
```

1: Data Sample (it goes from 1 to a maximal value at a given frequency);

2: Digitalized values of the V_V signal. It represents an integer number of signal periods;

3: Digitalized values of the V_I signal. It represents an integer number of signal periods;

4: Filtered values of the V_V signal with a digital low-pass filter;

5: Filtered values of the V_I signal with a digital low-pass filter;

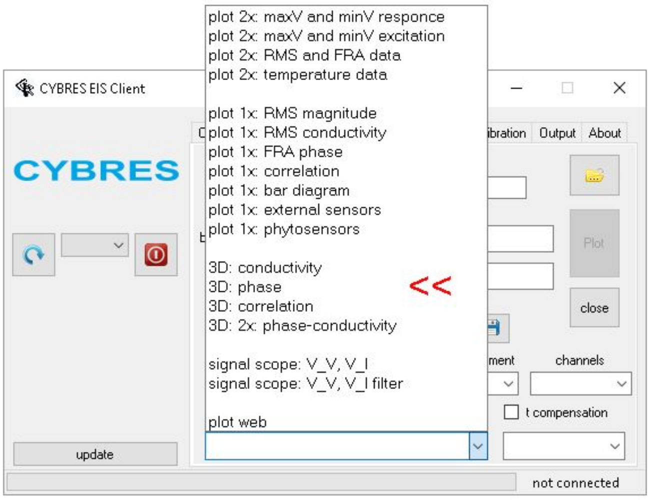
Data fields 6-10 are used by the statistical module for calculating statistical moments (see App. Note 26). Note that the mode 30 iteration replaces this file and indicates only statistical moments.

7.10 3D/4D plots

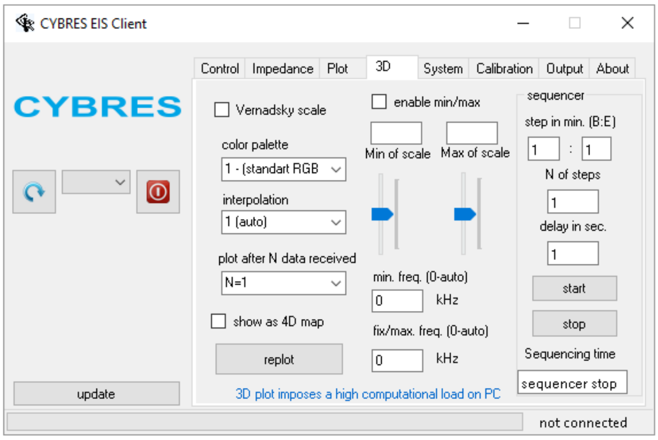
3D/4D plots are available only for the 'continuous measurements at variable f' and 'continuous FRP' modes. Note that this mode imposes a high computational load on PC. Selection of data and options for the plots are shown in Figure 63.

Each of these plots can be represented in original units, in Vernadsky scale and as 4D plots, see examples in Figure 64. More detail on the calculation and philosophy of the Vernadsky scale can be found in the publications[14]. In general, the Vernadsky scale

[14] S.Kernbach, I.Kuksin, O.Kernbach, A.Kernbach, *The Vernadsky scale – on*



(a)



(b)

Figure 63: 3D/4D plot **(a)** available plots; **(b)** options for the plot.

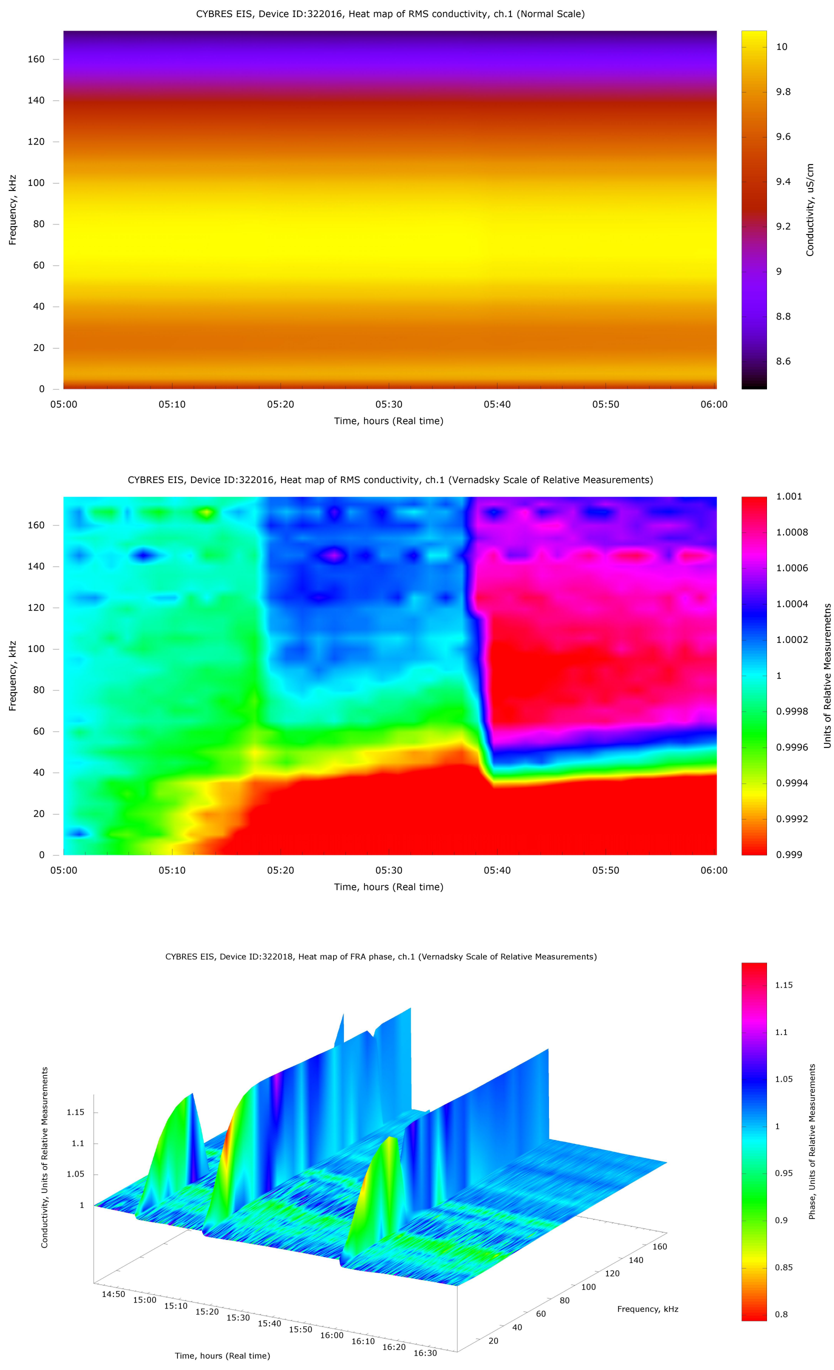


Figure 64: **(a)** 3D Graphical representation as the heat map, the color palette RGB; **(b)** The same data as in (a), but represented in the Vernadsky scale, the color palette HSV; **(c)** The representation of RMS conductivity and FRA phase of impedance in the Vernadsky scale in the form of 4D graph.

provides better dynamic range than the representation in original units and, thus, a higher resolution. The options 'color palette' and 'interpolation' allows customizing the representation in this mode. The option 'plot after N data received' allows reducing the computational load for real-time plots – the program will generate only one 3D/4D graphs after N received data sets.

The option 'enable min/max' provide a flexible way for defining the heat map boundary and, thus, a flexible color representation of the data. To use this option, set the maximal and minimal values (e.g. from automatically plotted 3D diagrams) in the corresponding fields and then adjust them. For replotting the data, press the button 'replot'.

Options 'min. freq (0-auto)' and 'fix/max. freq (0-auto)' allow to set up frequency range in the 3D plot (this changes only the representation, the scanned frequencies are not changed). The field 'fix/max. freq (0-auto)' allow to set one frequency for transition from 3D plots to 2D plots, i.e. all 2D graphs will be plotted at this frequency in 3D mode.

Sequencer allows animating the plot from data files (e.g. for demonstration or video producing purposes). It takes 'begin' and 'end' times from the corresponding fields in the section 'plot' and stepwise adds to them time steps defines in the fields 'B:E' (and then replots the diagrams). By setting '0' in the field e.g. 'B', it is possible to animate only the end time (e.g. the right boundary of the plot will 'move'). By non-zero settings in both fields 'B:E', both boundaries will 'move'. The time window to 'move' should be defined in the fields 'begin' and 'end' times in the section 'plot'. The number of steps is defined in the field 'N of steps'. The field 'delay in sec.' introduces a delay between steps, since 3D plot can take some time for generating graphs. When the output in the section 'plot' is set to files (e.g. png files), the sequencer will generate step-by-step graphical files with time displacement.

7.11 The web plot

The web plot mode uses the gnuplot script to write data in html files. This feature is useful when access to graphical information should be shared in a large auditoria, e.g. in internet.

metrology of EIS in time-frequency domain, IJUS, 15–16(5), 143–150, 2017

8 Measurements

8.1 The client program: the section 'impedance'

Setting of main EIS parameters is performed in the section 'impedance', see Figure 65. Settings are split into two columns: numeri-

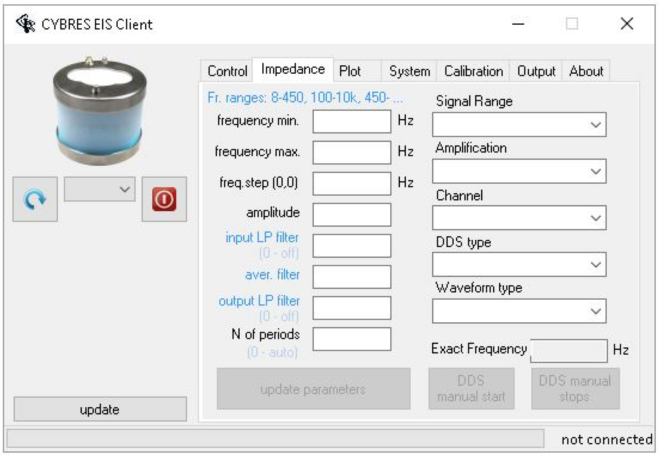


Figure 65: Example of data in the section 'impedance', see Table 18 for explanation.

cal values and pre-selected values (combo-boxes). The first column describes parameters of frequency sweeps, sample exposition time before measurement, filter settings, etc, see Table 18. Numerical values should be stored in the device by 'update parameters'. The DDS generator can be started manually with fixed frequency by 'generator start' and stopped by 'generator stop' (do not use it in the 'impedance spectroscope' and 'signal scope' modes).

8.2 Measurements in 'spectroscopy' modes

The spectroscopy modes ('impedance spectroscope' and 'Frequency Response Profile' – 'FRP') are intended for analyzing electrochemical behavior of samples in frequency domain. The thermostat can be turned on or off during these measurements. Using of the option 'temperature compensation' is described in Section 8.6. The 'impedance spectroscope' performs measurements at frequencies selected by user.

The 'FRP' scans 43 fixed frequencies (in Hz: 100, 250, 500, 1000, 2500, 5000, 7500, 10000, 15000, 20000, 25000, 30000, 35000, 40000, 45000, 50000, 55000, 60000, 65000, 70000, 75000, 80000, 85000, 90000, 95000, 100000, 105000, 110000, 115000, 120000, 125000, 130000, 135000, 140000, 145000, 150000, 155000, 160000, 165000, 170000, 175000, 190000, 200000) at each sweep.

Table 18: Parameters in the section 'impedance'.

Parameter	Description
Frequency min.	Start frequency for a sweep. In the 'signal scope' and 'continuous measurement' modes it defines a signal frequency used for measurements.
Frequency max.	Stop frequency for a sweep (no meaning for the signal scope and continuous measurements/FRP modes).
Frequency step	Steps for increasing frequency during a sweep (no meaning for the signal scope and continuous measurements/FRP modes).
Amplitude	Amplitude of excitation signal (range 1-127), counted in internal register values. Selecting the amplitude and amplification, the response signal should not be distorted, see Figure 68(c).
Input low-pass filter	Coefficient of the low-pass filter for smoothing the input V_V and V_I data. This value is divided by 1000, i.e. the value of 100 corresponds to 0.1. The filter is switched off at the value 0.
Aver. filter	The number of iterations used for data averaging. All data are smoothed inside of the N iterations (i.e. each sweep at each frequency is repeated N times).
Output low-pass filter	Coefficient of the low-pass filter for smoothing all output values. This value is divided by 10000, i.e. the value of 1000 corresponds to 0.1. The filter is switched off at the value 0.
N of periods	Experimental setting, specifying the number of signal periods inside V_V and V_I. At the value 0 the filter is switched off.
Amplification	The amplification factor (50, 500, 5000, 50000). This value (together with the amplidute of excitation signal) defines the amplitude of response signal. Selecting the amplitude and amplification, the response signal should not be distorted, see Figure 68(c).
Channel	input channels (calibration resistor 5000 Om, calibration resistor 500 Om, differential channels, single channel). Use the option 'differential channels' for measurements of two samples, 'single channel' – for measurement of channel N1.
DDS type	Type of signal analysis (impedance spectroscope, signal scope, continuous measurement, FRP, continuous FRP, off)
Waveform type	Defines the excitation signal V_V ('auto' mode can be used in most cases).
update parameters	Press this button to send the updated value to the device. The combo-boxes (Amplification, Channel, DDS type, Waveform type) send updated values automatically (without pressing the button 'update parameters').
DDS manual start/stop	Manual start/stop of the V_V signal generator at the 'Frequency min.'.

To activate these modes, select in 'DDS type' the option 'impedance spectroscope'/'Frequency Response Profile'. Note, the min./max. frequencies are defined as integer numbers, the step can be set with one position after decimal point, see Figure 66. The FRP mode is not affected by these settings, it runs always the predetermined set of frequencies.

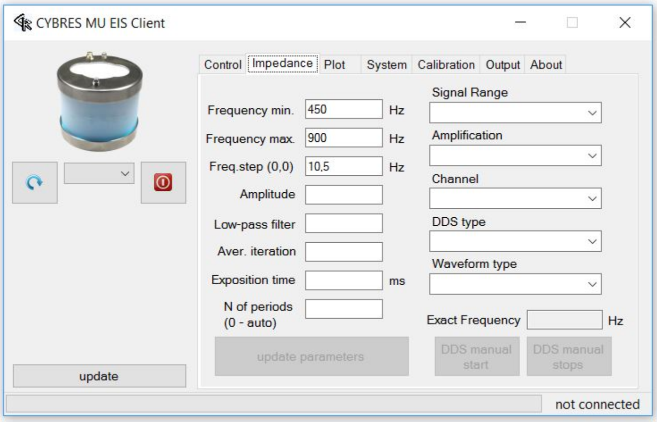


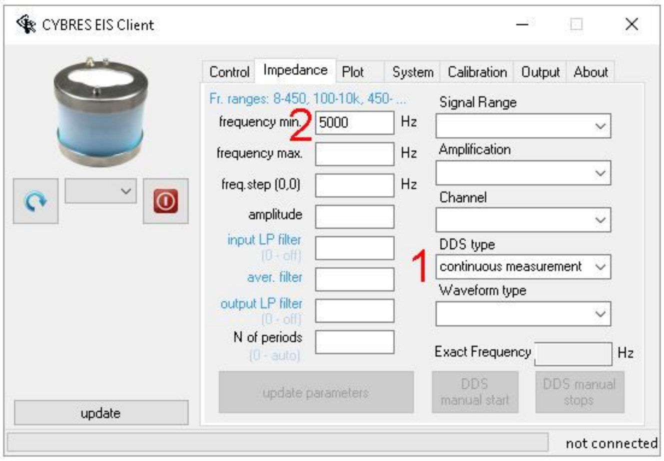
Figure 66: Example of frequency settings, min. frequency – 450 Hz, max. frequency – 900 Hz, step 10,5 Hz.

8.3 Measurements in 'continuous' modes

The continuous modes ('continuous measurement' and 'continuous FRP') allow conducting long-term measurements (in terms of days, weeks or months). The 'continuous measurement' performs measurements at one fixed frequency (selected by user), the 'continuous FRP' scans 30 frequencies at each sweep. The continuous modes are intended for analyzing electrochemical stability/changes of samples in time or in time-frequency domain.

All parameters from the spectroscope mode, see Table 16, are also available, the thermostat can be turned on or off. Additionally, this mode supports 3D and 4D plot (as heat maps for 'continuous FRP' only), the file 'xxx_3D.dat' is generated for these plots. The activation of the 'continuous measurement' mode is shown in Figure 67. The meaning and usage of the check boxes 'regression analysis' and 'temperature compensation' is described in Sections 8.5 and 8.6.

Activation of the 'continuous FRP' is similar to 'continuous measurement' mode. Since FRP mode has frequency dependent components, the time behavior in 'continuous FRP' is reconstructed from the written data. Thus, the time tics on X-axis can be selected in different way than the time tics on X-axis of other plots.



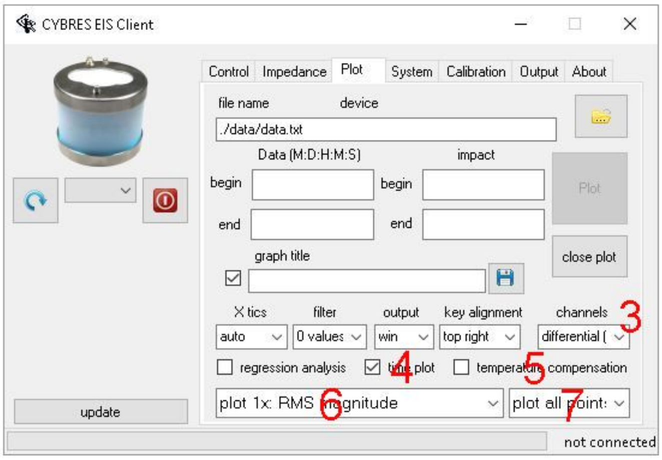
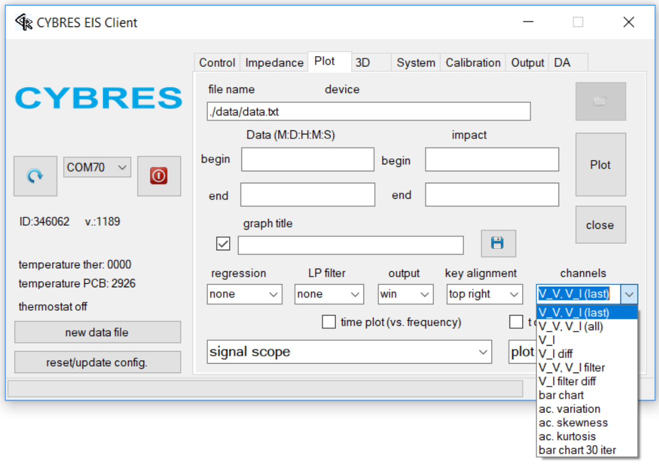


Figure 67: The 'continuous measurements' mode. To activate this mode: 1) select the DDS mode 'continuous measurement'; 2) set up the frequency that will be used for continuous measurements; 3) select the channel; 4) activate the box 'time plot'; 5) check/uncheck the temperature compensation; 6) select the used graph to plot; 7) select the required temporal resolution (all data, last minute, last 3 minutes, last 10 minutes, last 60 minutes and so on).

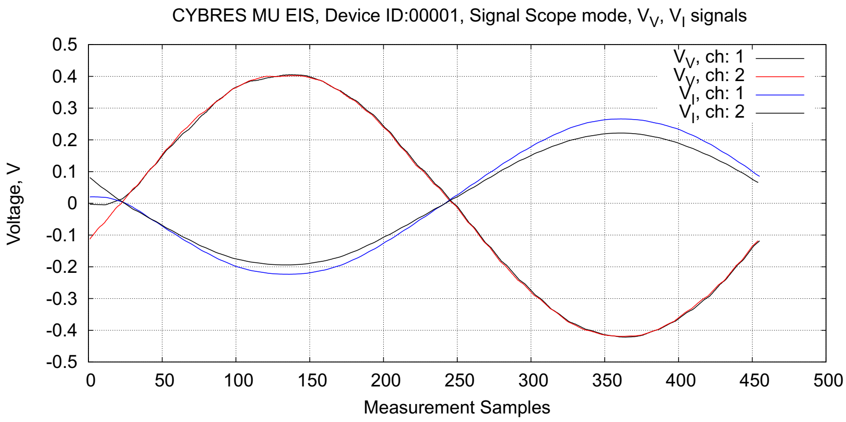
ATTENTION. Parameters of the continuous modes are similar to the spectroscope modes. It should be noted all graphics in these modes have time on the X axis, whereas the spectroscope modes have the frequency on the X axis.

8.4 Measurements in the 'signal scope' mode

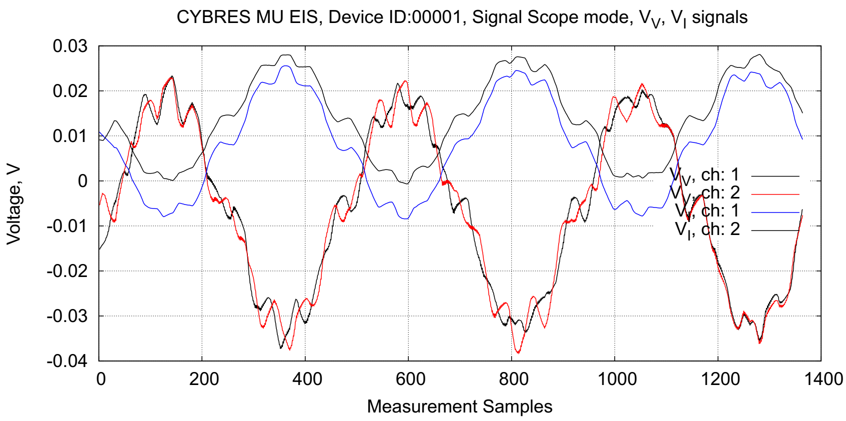
Signal scope mode allows visual observation of excitation V_V and response V_I signals. To activate this mode, select in the section 'Impedance' the DDS mode 'signal scope', in the section 'plot' any graph with the signal scope, the obtained plots are shown in Figure



(a)



(b)



(c)

Figure 68: **(a)** Example of graphical output in the 'signal scope' mode; **(b)** example of undistorted signals; **(c)** Examples of distorted signals – parameter 'amplitude' is too low;

68. One of main purposes of this mode is to detect distortions of excitation and response signals, to perform their statistical analysis (based on accumulated statistical moments, see App.Note 24) and to test the amplification range (i.e. to test the selection of water). Figure 68(c) demonstrates some examples of distorted and undistorted signals. To ensure valid measurements, it is recommended to perform the distortions analysis in the 'signal scope' mode after changing any EIS parameters or using a new fluid.

New in the firmware 1189. The firmware v1189 and the client v1.3 (and all later versions) support the 'auto' amplification – at each 'Measurement Start' the system sets first the range 50000 and tests whether the response signal is saturated. If the saturation is detected, the amplification goes to 5000 and the saturation test is performed again. This procedure is executed until the proper amplification range is found. The saturation tests is also performed at each sampling during continuous measurements. Thus, if the fluid conductivity will increase due to electrochemical degradation, the EIS system will automatically find a proper amplification range during long-term measurements.

The amplification function 'auto' produces some behavioral responses that need to be considered. First of all, the scope mode will demonstrate results of all saturations tests, see Figure 69 (with the option 'all'). To avoid this behaviour, select the required fixed amplification range (50000, 5000, 500, 50).

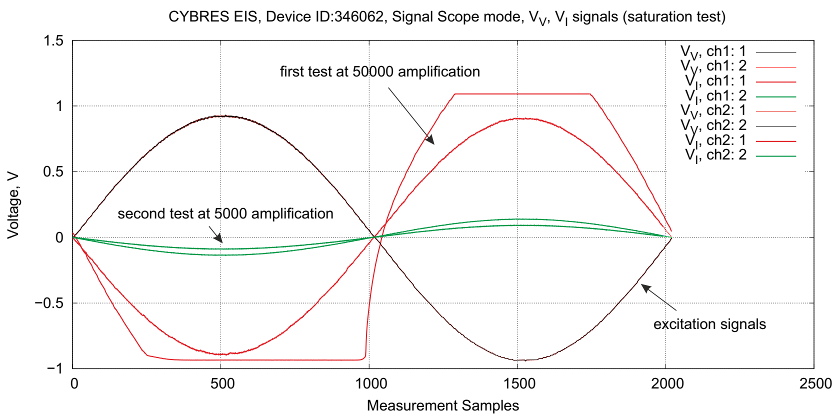


Figure 69: Saturation test in the signal scope mode. The EIS system performs several measurements and demonstrates results of all saturations tests (the option 'all').

Statistical tests 'bar chart' and all accumulated (ac.) variance, kurtosis, skewness and total will be performed for all tests, however indicated will be only the last one, i.e. with the correct amplification factor, see Figure 70.

The statistical test 'bar chart 30 iterations' does not work in the 'auto' amplification mode, because each iteration in this measure-

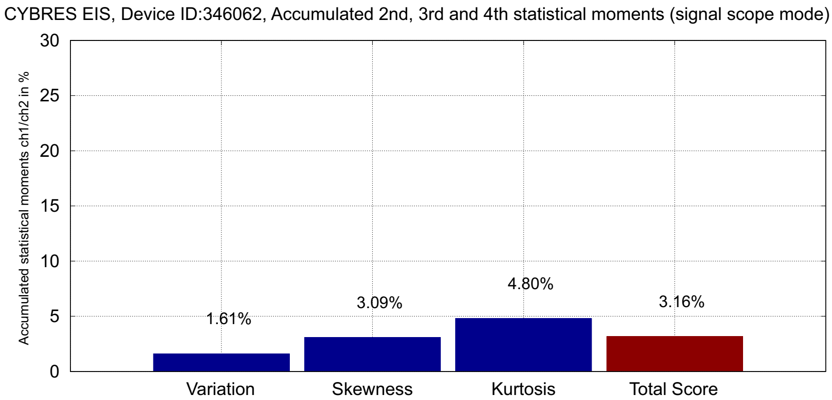


Figure 70: Statistical tests 'bar chart' for the saturation test from Figure 69.

ment mode starts with a new saturation test. To perform iterative statistical test, select the requited fixed amplification range, select 'bar chart 30 iter' and press 'Measurement Start'. During this measurement it is recommended to turn off the sound indication.

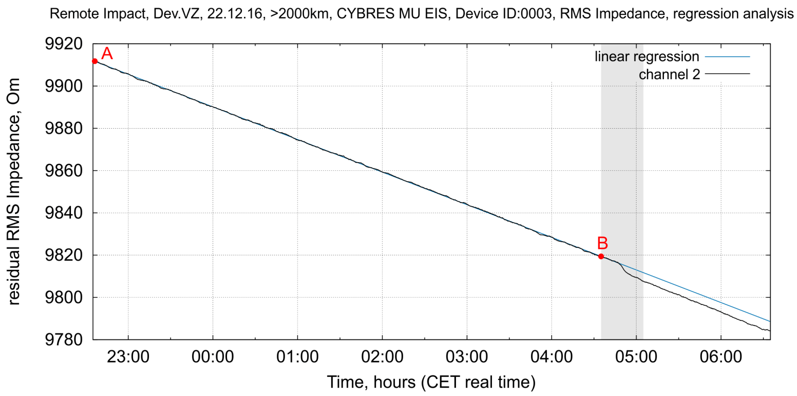
> **ATTENTION**. It is strongly recommended to perform one measurement in the 'signal scope' mode after changing any EIS parameters or using a new fluid.
> All statistical tests in the signal scope mode work also with older versions of firmware.

8.5 Regression analysis

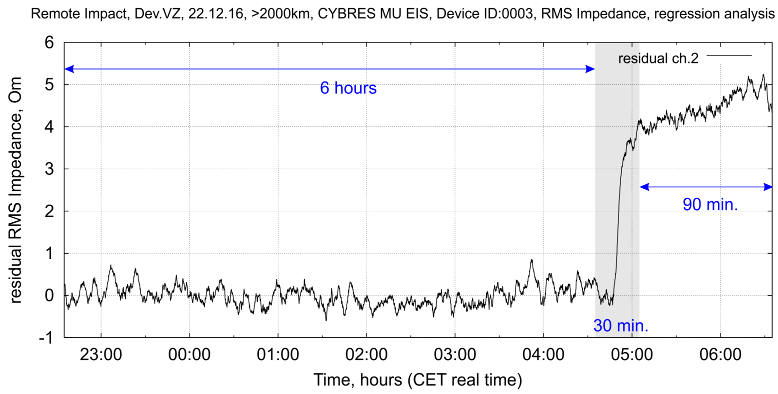
The regression analysis is used in the 'continuous measurement' mode and allows an essential increase of dynamical range and thus the resolution of measurements. This analysis requires a specific methodology of measurement, which should be divided into three phases with fixed duration: the phase 1 – the time prior to exposure; the phase 2 – the impact, the phase 3 – the time after impact, see Figure 71(a).

The Figure 71(b) shows an application of a linear regression between the start point of time window (the point A) and the beginning of impact (the point B). The residual curve is obtained by subtracting the linear regression from the raw sensor data. Selection of different time intervals for online regression analysis (i.e. during measurement) is shown in Figure 71(c). Timing of offline regression analysis (i.e. analysing stored data from file after measurements) can be specified by user in the fields 'Data (M:D:H:M:S)' and 'Impact'.

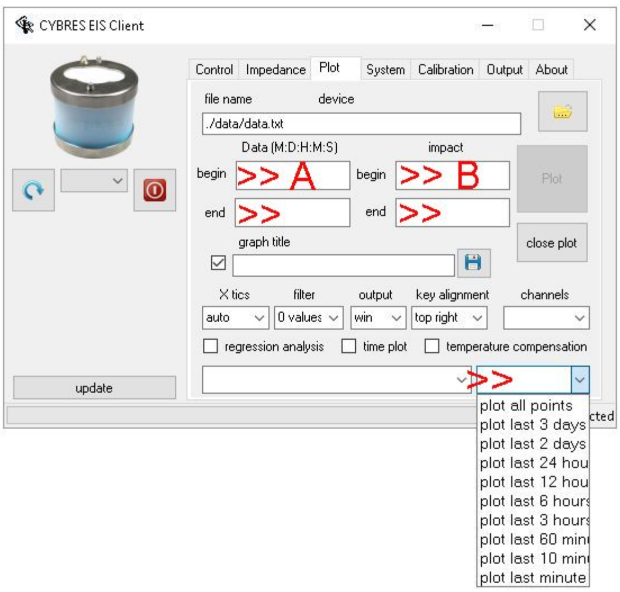
In general, it recommended to keep the time before and after impact as large as possible and to make the impact as short as possible. This allows reducing a probability of random fluctuations



(a)



(b)



(c)

Figure 71: The example of regression analysis. **(a)** Raw sensor data and the linear regression; **(b)** The residual curve obtained by subtracting the linear regression from the raw sensor data; **(c)** Different timing of regression analysis.

of measured signals during the impact and avoiding wrong conclusions from such measurements. The regression is performed based on the gnuplot function 'fit', refer to gnuplot manual for further detail.

8.6 Temperature compensation

The EIS spectrometer implements two different temperature modes. When the thermostat is turned on, the fluids have a fixed temperature set by user. This mode is recommended for long-term measurements, for cases when properties of two fluids are compared with each other or the temperature-dependent dynamics should be analysed.

However, all measurements in time or frequency domains can be also performed when the thermostat is turned off. This mode is useful e.g. for fast express analysis or in cases when samples are measured outside of the thermostat. To improve the accuracy of measurement in such cases the temperature compensation is implemented by using the coefficient γ:

$$\gamma = (1 + 0.019 * (25.0 - t)), \tag{30}$$

where t is taken from the thermostat sensor (as default mode) or from external temperature sensor. This coefficient re-calculates all conductivity/magnitude values to 25°C. It works in the following modes: impedance spectroscope, continuous measurement, FRP and continuous FRP. The temperature compensation can be enabled or disabled by the check box 'temperature compensation', see Figure 67.

It is recommended to insert water samples into the thermostat (even if the thermostat is off) for performing measurements with temperature compensation. When samples are not in the thermostat (and t from (30) is taken from the external temperature sensor), the equalization of temperature between both water containers is not performed and temperature compensation will be not efficient.

> **ATTENTION**. The temperature compensation has an approximative character and introduces inaccuracy into measurements. Use the thermostat for accurate measurements, especially for the long-term measurements.

8.7 Noise reduction, averaging and low-pass filters

Noise appears due to two factors: measurement noise appeared in samples and the low-signal noise appeared in the differential channel (due to similar signals and low amplitude of their difference).

Large random measurement noise indicates wrong preparation of samples and electrodes or some environmental EM noise. Check the sections 8.10 and 8.11 for preparation of sample, electrodes and environmental conditions.

> **ATTENTION**. Noised V_V and V_I signals can also mean that the amplification and signal amplitude are set in a wrong way in relation to the measured fluid (check the signal levels in the 'signal scope' mode).

The EIS system has several hardware and software filters intended for noise reduction, their structure is shown in Figure 72(a). Users have access to averaging and input/output LP filters, their coefficients are set in the section 'Impedance', see Figure 72(b). These filters are implemented in the EIS spectrometer. Additionally, the client program provides the averaging filter for plotting values, see Figure 72(c), it can be used even for filtering already measured data from files.

Generally, the noise can be reduced by increasing the value of averaging, setting the coefficients of input/output low-pass filters, and selecting larger averaging in the section 'plot'. Coefficients of LP filters are defined as

$$0 < \alpha_{input-LP} < 1, \tag{31}$$

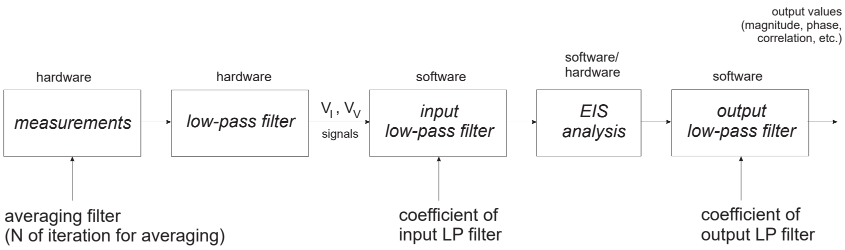
$$0 < \alpha_{output-LP} < 1, \tag{32}$$

and implemented as $N/1000$ (for input LP) and $N/10000$ (for output LP), where N is set in the client program, shown in Figure 72(c). The value '0' switches off the LP filters.

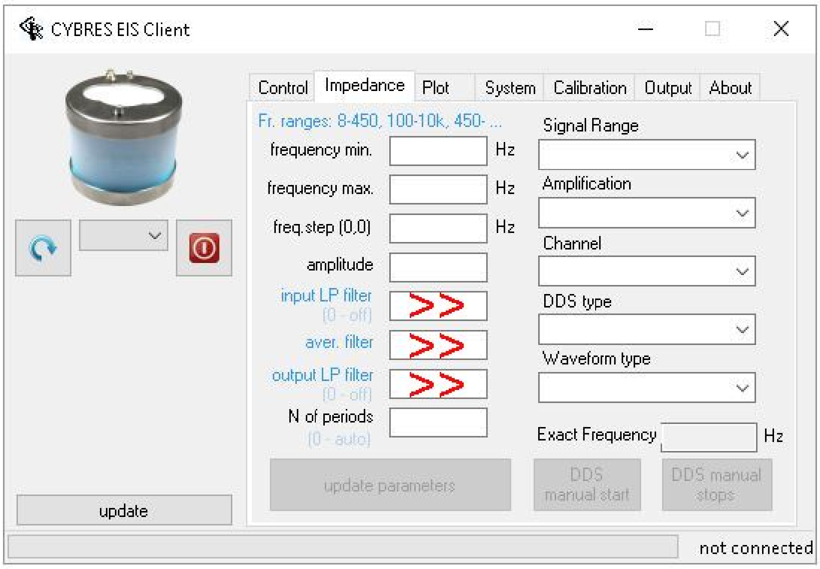
> **ATTENTION**. Settings of input/output LP filters influence analytic tools, for instance, change the phase of impedance. Use the same settings for measurements that needs to be compared with each other.

8.8 Calibration

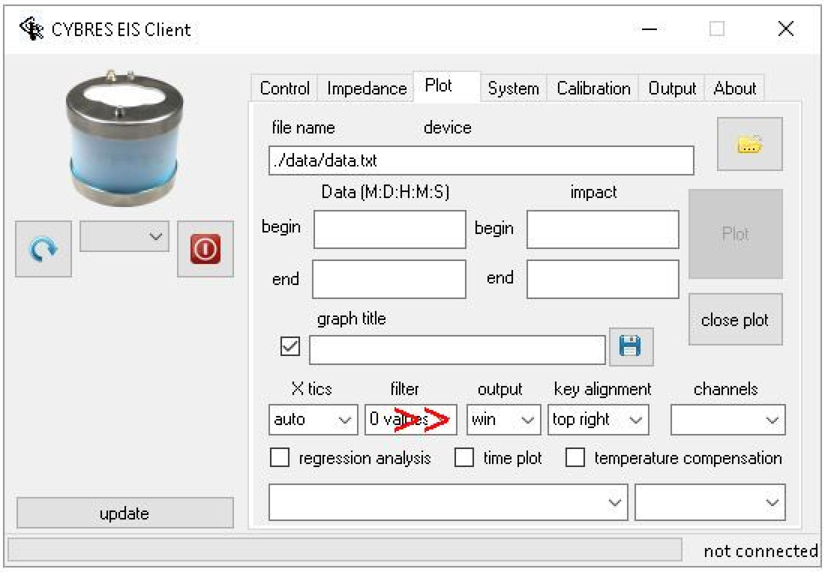
All EIS values in differential mode (time-differential or channel-differential) are measured in relation to each other. The time-differential dynamics considers the relation between values at different times from the same channel, the channel-differential dynamics considers the relation between different channels. In these cases the EIS values are measured as 'relative' (not 'absolute') and the device does not need a calibration. The calibration is required only in two cases – for measuring an absolute value of conductivity



(a)



(b)



(c)

Figure 72: **(a)** Structure of hardware and software filters in the EIS spectrometer; **(b)** Setting the coefficients of filters implemented in EIS spectrometer; **(c)** Averaging filter in the client program (it can be used even for additional filtering of data plotted from files).

at a fixed frequency or for linearizing the frequency dependent EIS dynamics.

Measuring conductivity at fixed frequency (firmware). Firmware provides functionality for calibration at fixed frequency. Calibration of the cell constant should be performed by using a calibration fluid with specified conductivity after the thermostats archived the pre-defined temperature. The calibration is valid only for one fixed frequency (usually in the range of 3-5 kHz), if the frequency is changed, the re-calibration of conductivity for a new frequency is required. For performing the calibration, open the section 'calibration', see Figure 73. The multiplicative coefficient 1.0000 corresponds to the value 10000 (e.g. 0.6754 corresponds to 6754). Enter the coefficient for each channel so that the measured conductivity is equal to the specified conductivity of the calibration fluid (at given temperature). It needs to take into account that exact conductivity values change depending on light conditions, CO_2 absorbtion and fluid's degassing (see more the CYBRES Application Notes 18 and 20).

Calibration coefficient in the Client program. The Client program provides calibration coefficients for all EIS channels. It can be found in the script file '/scripts/commonFunctions.dat':

```
# calibration coefficient for FRA measurements
ImpedanceCoeff=1
```

Note this coefficient can be used only for temporary purposes (it should be equal 1 during measurements).

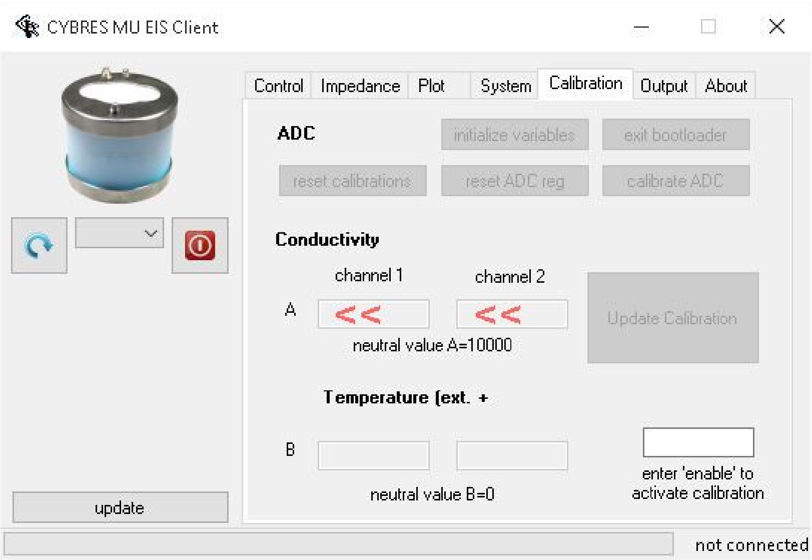


Figure 73: Calibration of conductivity at a fixed frequency.

Linearization of the frequency dependent EIS dynamics. The analog circuitry of the EIS spectrometer varies its own parameters (e.g. amplification factors) depending on the used frequency. This variation represents a systematic error if the differential approach is not used. To remove this error, the device can be calibrated over the selected frequency range. The EIS has two 'onboard' calibration resistors of 4.99 kOm and 499 Om of 0.1%, 25 ppm accuracy, which can be connected to electronics instead of external electrodes, see the section 6.7. Selection of electrodes or calibration resistors is performed in the client program in the box 'Channel' – 'calibration resistor 5000 Om', 'calibration resistor 500 Om', 'differential channels', 'single channel' (two last options connects electrodes), see Figure 74. These resistors allow calibrating analog circuitry over the used frequency range. Users should record first the frequency dynamics on the calibration resistor and then should perform the measurement with connected electrodes (with the same setting). Finally, the calibrated values should be subtracted from the measured values (with an external program).

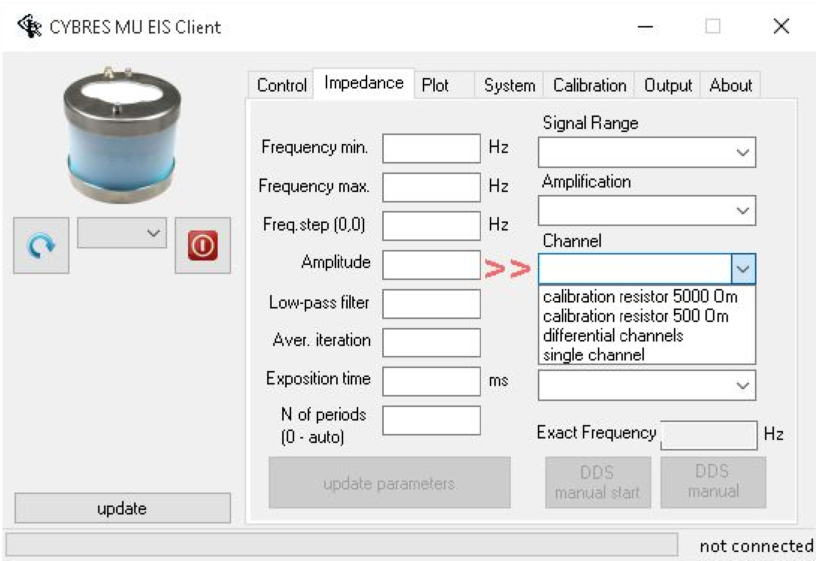


Figure 74: Selection of calibration resistors or external electrodes.

ATTENTION. Since both channels has the same frequency-dependent dynamics, calibration in differential mode of operation is not required. However, the absolute-values-measurements require a periodical calibration at selected parameters.

8.9 Double differential measurements

The differential measurement approach uses two identical channel A (experimental) and B (control), which apply the same exciting signal V_V. Spectrograms A and B are subtracted from each other, the resulting differential spectrum have a constant bias at all frequencies if the samples A and B are equal to each other. A changing bias of differential spectrum indicates differences in samples. When both samples are prepared at the same temperature, in the same EM, light and other conditions, the difference is caused only by exposure to experimental factors.

For comparing the samples before and after the exposure, use the double differential method. It requires at least two differential measurements. In the first one the samples A and B are measured before impact, these data are used for calibration. The second measurement of A and B is performed after exposing the sample A. The measurement result consists of two differential curves: the control and experimental ones. The difference between them allows making conclusions about impact on the sample A. It is recommended to perform the double differential measurements with four samples A, B, C an D, where A and B are measured as a control pair, C and D are measured as an experimental pair after the sample D is exposed to experimental factors, see the Application Note 20: 'Increasing accuracy of repeated EIS measurements for detecting weak emissions' for further detail.

8.10 Where to perform measurements

The device can be installed in any place, however, a care should be taken on maintaining a constant temperature in the room during measurements. Thermostats have some reaction time, thus, the less is the variation of ambient temperature during measurements, the less is the distortion of measured parameters. A closed room (without opening outside doors or windows) satisfy these requirements.

Measurements of weak and ultra-weak impact factors require additional arrangements. First, the measuring system should be carefully isolated from the environment. The device and the place should be not illuminated by direct sunlight (e.g. basements are well suitable for such measurements). Samples and the device should be not placed near electrical wiring of any kind, computer communication systems, and any devices that produce EM emission (mobile phones or WiFi devices). It is recommended not to install the device close to walls, especially bearing or outer walls and metal constructions. It is also recommended that the operator (experimenter) avoids contacts with samples and work only a short time close to the device during measurements.

ATTENTION. The level of isolation of the measuring system and samples from the environmental impacts is crucial for the quality of accurate measurements. It is recommended to avoid sunlight, WiFi devices and the experimenter near the device and samples during measurement of weak and ultra-weak impact factors.

8.11 Preparation of samples and electrodes

Most of organic or anorganic liquids can be used for measurements. Aggressive acids and alkali should be avoided otherwise they will corrode stainless steel electrodes. Alcohol should not be used for cleaning the measuring cell (electrodes can be cleaned with alcohol). It is recommended to use different electrodes and containers for different types of fluids.

Samples after placing into the thermostat require some time to equalize temperature. Recommended time is about 15-20 minutes for 10-15 ml of liquid sample. The achievement of set temperature is indicated by the system (LED will switch to 'normal operation').

Make sure that there are **no gas micro-bubbles** on electrodes when the fluid is filled in the measurement container. To remove micro-bubbles, wait until the pre-set thermostat temperature is archived, make one measurement, carefully remove electrodes from containers and then insert them again. After the pre-set thermostat temperature is archived, the device is ready for measurements. To test this, electrodes can be removed from containers again, no differences should be measured compared to the previous results.

ATTENTION. Alcohol should not be used for cleaning the measuring cell (electrodes can be cleaned with alcohol).
Perform measurements after the sample temperature is equalized (15-20 minutes after the placement of samples in thermostat). Make sure that there are no gas bubbles on electrodes. To remove micro-bubbles, wait until the pre-set thermostat temperature is archived, make one measurement, carefully remove electrodes from containers and then insert them again.

8.12 How to ensure a valid measurement

1. Right selection of the V_V amplitude and amplification factor. The V_V and V_I signals should not be distorted and should be inside of -1V and 1V ranges (it can be seen on the maxV/minV graphs and in the signal scope mode). The amplitude of V_V should correspond to the nature of tested electrochemical system, e.g. biological systems require a small amplitude of excitation signals. Since the measurement approach interacts with samples, it is rec-

ommended to operate with small amplitudes of V_V signal (over the noise range).

2. Right selection of the V_V waveform. It is recommended to use harmonic signal with a maximal number of samples (use the 'auto' option for signal settings). Some analysis, e.g. correlation, can be performed with non-harmonic signals, however results of other analytic tools will be distorted.

3. Right selection of the frequency range. Plot the number of data samples in the stored signal period, these values should be <2047 and decrease with increasing the frequency f. Too low value leads to a large noise for all analytic tools.

4. Make sure that electrodes have no micro-bubbles. Perform several control measurements, where electrodes are removed and inserted into containers. Note the maximal difference obtained in this case.

5. Make sure that the samples remain electrochemically stable, i.e. repeated measurements without removing electrodes provide the same results. In case the values changes each time, decrease the amplitude of V_V signal. When the sample are still electrochemically unstable (before exposing to experimental factors), replace the sample – they are not suitable for EIS measurements.

6. Exposure by experimental impact factors can change electrochemical stability of samples. When measurement results are changing at repeated measurements after exposure, this appeared instability is a result by itself.

7. Perform several measurements inside one experiment and perform several independent experiments to accumulate large statistics. Never consider a single measurement or a single experiment as a proof for any weak and ultra-weak impact factors.

8. Only when the differential measurement is essentially larger/smaller then the repeated control measurements (with inserting/removing electrodes), the experimental impact factor could be considered as measured, taking into account other factors, see additional literature.

9. Make sure that the impacted samples are stored in a proper place between repeated measurements. Conventional and unconventional environmental factors (and operation by humans) impact samples, thus repeated measurements, e.g. 24, 48 hours after the original measurement, will be different due to impact of these additional factors. Always consider the impact of CO_2 absorption by water (which also depends on temperature) at repetitive measurements. Warmed up and cooled down samples at temperature t_0 will have different parameters than samples continuously stored at t_0.

9 Real-time AI applications with Python

For advanced numerical analysis, AI (artificial intelligence) applications and real-time actuation, EIS client can provide data to a python server via named pipe mechanism, see Fig. 75. The pipe name

```
\\.\\pipe\\EISClientPipe
```

and it can be used for any asynchronous external data processing algorithms (not only python) that run on the same machine as the EIS client. If these algorithms are executed on another computer use the EIS socket mechanism.

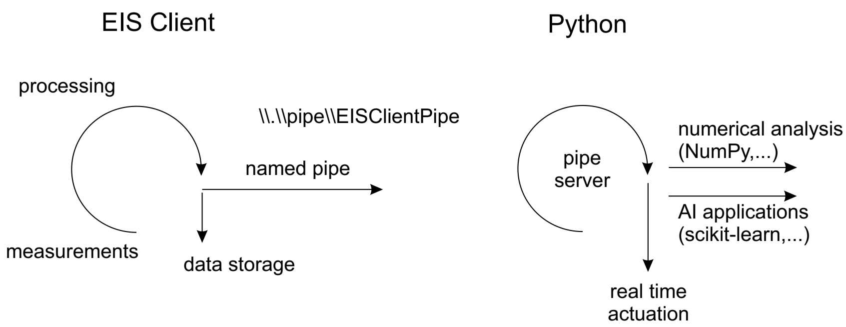


Figure 75: Connection between EIS Client and python application via named pipe.

Adding python programs represents a flexible approach for used-defined data processing that can be modified any time by end-user. To activate the data exchange via named pipe enable

```
usePythonPipe=1;
```

in *init.ini* file. Data are provided to the pipe at the same time and in the same structure as they are stored in files – line by line with all data columns (see Section 7.8). If the parameter

```
saveAfterNSamples=x;\psi
```

is set to x, the pipe will provide x lines of data after x measurement cycles – the python program needs to decode these data.

If *usePythonPipe* is activated, the EIS client starts a new instance of cmd shell and executes

```
python .\python\python_server.py
```

each time when the device is connected to the client. Users need to instal python and all necessary libraries, e.g.

```
python -m pip install pywin32
python -m pip install pandas
python -m pip install numpy
...
```

Example of the 'python_server.py' is provided, for any other applications it needs to take into account a specific multithreading execution of the pipe client: each time when data are ready, it opens a new pipe handler, stores data to pipe and then closes the handler. The pipe server (python or any other program) has to create a named pipe, wait until the client is connected to the pipe, read date, process them and again create a new pipe and wait for connection on the next cycle of data measurements.

ATTENTION. This data exchange is asynchronous, the EIS client does not wait until the python server finished the data processing – for complex programs requiring a long execution time, it is recommended to use the option 'saveAfterNSamples=x;'.

The pipe data exchange can be used with real-time signal processing algorithms embedded in the EIS Client, see Section 10, after setting

```
usingActuators=1;
```

corresponding data fields will be filled with the processed data and can be used in the python programs. Since DA signal processing is fast, this can represent an efficient implementation of complex statistical or numerical analysis in real time.

10 DA module: real-time signal processing and actuation

The detectors-actuators (DA) module and methods described in this section have four main goals:

1. to provide a flexible way to create a sensor-actuator system, e.g. to detect specific signals (signal patterns) in all sensor data and to react on these changes in sound-, music-, speech-, light- way; turning on/off some physical devices; by electrical stimulation or sending messages (e.g. in files, IP addresses or twitter accounts);

2. to create environmental feedback loops, experimental feedback loops or to connect electrochemical/biological sensors with real-world actuators, in particular to explore in-/out-system dependencies and homeostatic behavior, to develop complex demonstration scenarios and setups;

3. to enable performing fully automatic experiments, where a human operator (an experimentator) is removed from operation of all sensing, transmitting or emitting devices, especially in such experiments that involve quantum phenomena in macroscopic systems;

4. to enable a real-time data analysis by numerical processors and creation of synthetic (virtual) sensors by performing a sensor fusion from different physical sensors.

This functionality is useful for the EIS spectrometer, all MU/EHM-C control modules (e.g. the Poynting vector generator), phytosensor and environmental (bio)sensor applications. In total there are about 500 possible real and virtual sensors and about 300 possible actuators provided by CYBRES MU hardware and software – this enables a large number of possible fully automated scenarios and experiments. Real-time dynamical sensor-actuator mapping allows implementing advanced computer learning approaches in bio-hybrid systems by any external software. The DA module implements not only the reactive 'stimuli-response' behavior, it supports the probabilistic interface as event-driven Bayesian (belief) networks, Petri nets and several feedback mechanisms for developing adaptive homeostatic behavior.

To enable the detector-actuator module, in the file './init/init.ini' set the parameter

```
usingActuators=1;
```

Setting to '0' will disable this functionality. The real-time analysis and detector-actuator mapping operate in parallel to existing graphical/plotting engines on the MU system and EIS Client, and do not interfere with them. Numerical processors operate with existing data channels and can write new channels into the main data file. The DA module operate with real-time data obtained from MU devices as well as with data read from a file.

Working with the DA module, follow several rules:

1. The DA module provides functionality for exploration and experimentation purposes for users. Like any programming language, it assumes an 'open-ended' character and can produce an unpredictable behavior for connected actuators. Use this module and connected actuators on your own risk. The producer of the MU system is not liable for any direct or potential damage arising out of use of the DA module.

2. Numerical computations consume resources on your PC. For some 'slow' PC, samples from the MU devices can follow faster than their numerical processing. If the client program obtains a new measurement sample, but the processing of the previous one is still not finished, it will create an overflowing of input buffer and unpredictable behavior of the client program. In such cases increase the 'period btw. measurements' (in section system)and 'plot after N data received' (in section 3D) in order to provide more time for numerical computations.

3. Logging the behavior of DA module generates a large amount of information. Use the logging functionality only for debugging purposes and switch it off for normal runs. Use the actuators A1-A20 for indication purposes, they can write specific messages into the file '/log/messagesDA.txt' and in the main file.

4. The data channels 1-45 are produced by MU devices and represent results of physical measurements. The enabled statistical package 'EIS statistics' writes 24 data channels into the main data stream into the field 46-69. Thus, depending on the enabled/disabled 'EIS statistics', numerical processors and A1-A20 of the DA module can also write own results into the main data stream after the channel 45 or 69. Use this functionality carefully. Use the gnuplot script '/script/printDeveloper.dat' (examples are provided in the script) for plotting these new data channels.

10.1 The real-time detectors

The MU system and CYBRES EIS Client are able to perform in-hardware and in-software real time signal processing by means of embedded numerical processors and real-time detectors. Examples of numerical processors are the mean, standard deviation or z score calculations. Examples of detectors are the time interval detector, the peak detector, the cyclical change detector, the noise levels detector, the gradient change detector, time detector and others. Each processor and detector Dx is implemented as an independent module. Detectors take signals from short-term, middle-term or long-term data pipes and can be configured by user for processing any of the existing data channels [1-33] (data from real sensors such as potential measurements, electrophysiology, electrochemical data or external sensors, see Sec. 7.8 for mapping of sensors and data channels) and data channels prepared by numerical processors. All pipes prepare their data automatically:

- **short-term data:** each data sample is prepared with timing defined in 'system' → 'period between measurements' (usually in sec. time step);
- **middle-term data:** data samples are selected with time steps in minutes. It is implemented as 'selecting one sample of short-term data' when the short-term buffer is full (cycle of short-term data data collection);
- **long-term data:** data samples are selected with time steps in hours. It is implemented as 'selecting one sample of middle-term data' when the middle-term buffer is full (cycle of middle-term data data collection);

The size of buffer for all data pipes is controlled by the parameter 'bufferSizeDataPipes' from the file '/init/init.ini'. Thus, different data pipes provide a possibility to analyze different time scales in the sensor data. Each enabled detector writes the result of detection into the output vector, see Figure 76. When the corresponding

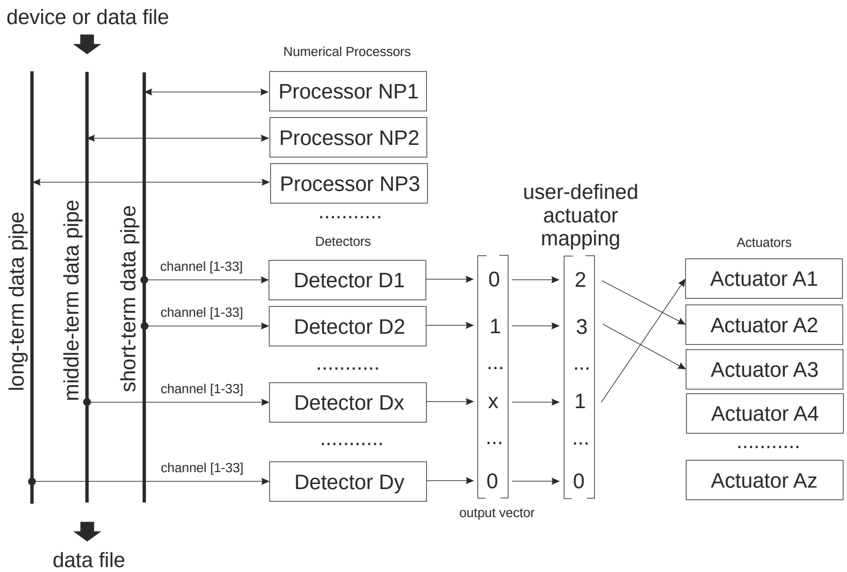


Figure 76: Schematic representation of the detector-actuator coupling.

detector Dx detect the necessary changes in the signal (the 'true' condition), it writes '1' in the output vector, otherwise (the 'false' condition) it write '-1'. When the detector Dx was not executed at some step, it writes '0' in the output vector. Detectors can also write some numbers in the output vector, e.g. the noise level detector writes the noise level in the output vector. All detectors operate in parallel, they start when the corresponding data pipe issues the signal 'new data are ready'. The detector is switched off when its input data channel is configured as '0'. All detectors can be individually configured with one or several parameter (see Table 19).

Logging the detector behavior. Each detector and actuator provide information about the output of signal processing algorithms and executed activities at each event 'new data received'. This information is necessary for testing purposes and for adjustment of parameters. The DA module provides several ways to log this information. First of all, the parameter 'DAloggingBehaviour' in the file './init/init.ini' specifies the logging behaviour

```
DAloggingBehaviour=0;
```

where 0 – DA module does not provide data logging; 1 – DA module writes only the main DA related logging information in the output window of EIS Client; 2 – DA module writes detailed DA related logging information in the output window of EIS Client; 3 – only the main DA-related logging information is written into the file './log/DAlog.txt'; 4 – detailed DA-related logging information is written into the file './log/DAlog.txt'; 5 – main information into the output screen and file; 6 – detailed information the output window and file. Secondly, by setting

```
logFileWrite=1;
```

the whole logging information (not only from the DA module) from the output window of EIS Client will be written into the file './log/log.txt'. Thus setting DAloggingBehaviour=1; or DAloggingBehaviour=2; can be used with this option. The third way to log specific information is to use the actuators A1-A20 with comments and specified marks (%T, %D, %S, %B and so on). This information is written into the file './log/messagesDA.txt'.

ATTENTION. It is recommended to enable the data logging of DA module for testing and developmental purposes. Since the logging generates a large amount of information, disable it by setting 'DAloggingBehaviour=0' for long-term runs. Note that all alerts, e.g. encountered errors, will be written into the Client output window independently of setting 'DAloggingBehaviour'.

10.2 The detector-actuator mapping

The mapping between real-time detectors and actuators is defined by user with the mapping vector. The position in the mapping vector points to the detector, the value in this position points to the connected actuator. For instance, in the following mapping vector $\begin{bmatrix} 2 \\ 3 \\ 1 \\ \ldots \end{bmatrix}$, the value 2 written in the position 1 means that the detector D1 will turn on the actuator A2 in the case of positive detection (the 'true' condition), see Figure 77. Several detectors

Figure 77: Assigning the detector D1 to the actuator A2.

can be also configured for detecting the 'false condition' by defining the negative value of mapping 'D-x=Ay', see Figure 78. For

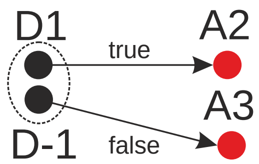


Figure 78: Assigning the detector D1 to the actuator A2 (the 'true' condition) and to A3 (the 'false' condition).

example, the following configuration

```
I1=4;
P1=1;
D1=1;
D-1=2;
```

is equivalent to the following expression

```
if (data[4][x]>data[4][x-1]) call A1; else call A2;
```

Note that not all detectors have the 'false' condition, see Table 19.

ATTENTION. Settings of detectors, actuators and detector-actuator mapping can be dynamically changed during execution by an external program. This approach can underlay some computer learning strategies or establishing environmental feedbacks in bio-hybrid systems.

10.3 Actuators

The list of available actuators includes the following groups:

1. **wavOut/mpegOut device:** play sound .wav/.mp3 file from position 'x' to position 'y';
2. **sound device:** change volume, change right/left stereo balance;

3. **MIDI device:** generate musical MIDI tones;
4. **text-to-speech (TTS) device:** generate the voice message;
5. **logical/probabilistic device:** to compute different logical/probabilistic operations and expressions from detectors;
6. **adaptation mechanisms:** several instruments and methods to implement adaption in probabilistic networks;
7. **RGB LED device:** turn on/off R,G,B components of LED connected to the MU system;
8. **external physical devices connected to the MU system:** e.g. turn on/off lamps/pumps connected to the MU system (by sensing the ASCI commands in COM-port);
9. **electrical stimulation device:** generate the electrical stimulation by the MU impedance measurement system;
10. **external physical devices connected e.g. to USB port:** generic control of external devices;
11. **'intelligent house' devices and systems:** on/off and parametric control of these devices;
12. **send message to file:** write text message to file;
13. **send message to IP port:** send text message to specific IP port in internet/intranet;
14. **send message to twitter account:** send text message to specific twitter account.

Each actuators is configured separately, see Table 20. Below are shown several tested commands for some actuators:

- **USB relay with FTDI driver** (e.g. type SainSmart 4/8 channel USB Relay Board): the board is controlled by 8-bit binary number, where each bit represents the relay state (0 = off, 1 = on), e.g. '10001001' would turn on relays 8, 4, and 1, all others would turn off. The value 137 ('10001001') should be send via actuator A21-A40, e.g. 'A21=COM6 9600 %H15' will turn on the relay 1,2,3,4, 'A21=COM6 9600 %H0' will turn off all relay (COM6 is an example, it can be different in another system). Note that the control software for the USB Relay Board should be executed before starting the client.

- **Energenie EG-PMS2**, Programmable 6-Socket Power Outlet Strip (USB version): this device is controlled by own software 'pm.exe', use actuators A191-A200 for control, e.g. 'A191=C:\Program Files\Power Manager\pm -on -Device2 -Socket1' will turn on the the socket 1 on the device 2; 'A192=C:\Program Files\Power Manager\pm -off -Device2 -Socket1;' will turn it off. Note that 'pm.exe' should be executed before starting the client.

- **Pololu Mini Maestro 6/12/18/24-Channel USB Servo Controller**, it enables USB based control over PWM motor drives, use it with A21-A40 actuators and '%Hx' marks for controlling the motors (see user manual of the Pololu devices).

- **using MU actuators with boards MU3.0, MU3.1,...**

'A21=COM6 625000 wk111*' will turn on the RGB LED on the connected MU device on COM6 (see Table 13 for the list of MU OS commands). Note that A21-A40 operate with additional MU devices. Use A41-A60 for the main MU device, where the client program is used to handle the measured data, e.g. 'A41=wk111*' in this case.

10.4 Dynamic configuration of detectors, actuators and the detector-actuator coupling

Configuration of detectors, actuators and the detector-actuator coupling is stored in the file './init/configurationDA.ini'. This file is read when starting the Client program or by pressing the button 'update'. This file has five following sections. The section 'I' defines the input channel of the corresponding detector:

```
\\section I: configuring input channel (int) for detectors
I1=4
I2=4
I3=0
I4=0
...
```

For instance, 'I1=4' means that the detector D1 has an input channel 4 and 'I3=0' means that the detector D3 is off.

The section 'P' provides specific parameters for detectors (see Table 19)

```
\\section P: configuring detectors (int value)
P1=445345
P2=354
P3=0
P4=0
...
```

Each detector has either no parameters (defined as '0') or some numbers (e.g. the threshold on the noise detector).

The section 'D' defines the detector-actuator mapping:

```
\\section D: configuring the detector-actuator mapping
D1=2
D2=3
D-2=5
D3=1
```

```
D4=0
...
```

For instance, 'D1=2' means that the detector 1 is mapped with the actuator 2, 'D4=0' means that the detector D4 is not connected to any actuator. The expressions 'D2=3' and 'D-2=5 mean that the 'true' condition of D2 is connected to A3, the 'false' condition – to A5.

The section 'A' defines the parameters for actuators:

```
\\section A: configuring actuators
A1=./sound/relax1.wav from 0 to 1000
A2=./sound/relax2.wav from 0 to 1000
A3=0
A4=0
...
```

These parameters are specific for each actuator, see Table 20. For instance the 'A1=./sound/relax1.wav from 0 to 1000' will play the sound file './sound/relax1.wav' from position 0 ms. to position 1000 ms.

The section 'B' is an optional section and defines probabilistic parameters Dx→(Ay|Bx) for the Bayesian network of detectors-actuators (see the Sec. 10.8):

```
\\section B: configuring probabilistic transitions
B1=50;
B2=60;
...
```

Initially all B parameters have 100 (i.e. all probabilities are 1). When the probabilistic transition mechanism is not used, this section can be omitted. In this example the transitions for D1 and D2 are defined as

D1→(Ay|B1)⇒D1→(Ay|0.5);

D2→(Ay|B2)⇒D2→(Ay|0.6);

Order of processing. All detectors are processed sequentially in order of their appearance. The output vector of detectors is processed from 0 to N, i.e. output of the detector 10 will be processed before the detector 100. Changing the priority of processing is possible by setting 'Dx=-y'. Here all mappings with 'Ay' defined as 'Dx=-y' will be processed before 'Ay' defined as 'Dx=y'. For instance, 'D10=5' (the 'true' condition of the detector 10 is assigned to the actuator 5) will be processed before 'D30=5' in case if both detectors fire at the same time; setting 'D30=-5' will be precessed before 'D10=5'. Note that all random and time detectors do not provide the functionality for changing the processing order.

ATTENTION. Each parameters has an unique identificator (e.g. 'A1', 'I2'). All identificators can be written in arbitrary order, when the same identificators are encountered several times, the last one will be used to configure the corresponding parameter. All identificators with '0' parameters can be omitted, i.e. all default values are '0' (all 'B' parameters are initialised with 100).

10.5 Numerical processors

Numerical processors are small software modules that only process the input data and write the result back into the main data stream, see Figure 79. Numerical processors are executed before detectors and actuators. Results of numerical processors represent new data channels and can be handled by detectors in the same way as data channels from the measurement device. Since these new data channels will be written into the file for plotting, the numerical processing of sensor data can be integrated into the main plotting engine on the level of gnuplot scripts. This feature provides to users a possibility to extend the real-time data processing and to integrate it with automatic plotting system.

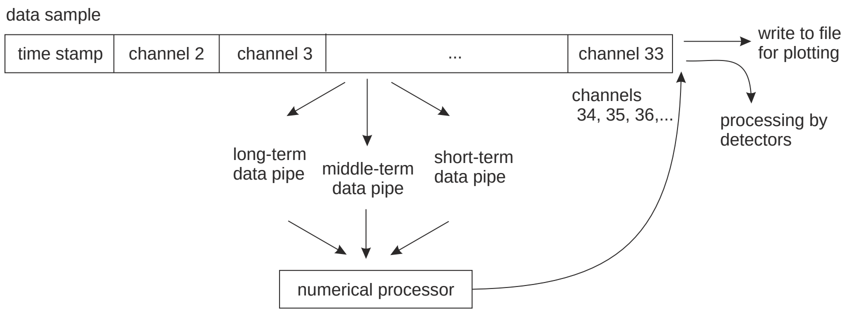


Figure 79: Concept of numerical processors.

Numerical processors represent a specific case of detectors without actuator-mapping capability, i.e. the section 'D' is not defined for them. Configuration of numerical processors follows a normal procedure defined for detectors: it needs to specify the input channel by 'I' and the parameters by 'P' (if necessary). The number of output data channel for writing results is defined by the order of processors: the first used processor writes data into the channel 34, the second – into the channel 35 and so on. The list of available numerical processors is shown in Table 19. To keep the computational load on a reasonable level, it is allowed to operate only 20 numerical processors in parallel.

In the following example, the mean μ (the processor 151) and the standard deviation σ (the processor 161) are calculated for the input channel 17 within the short-term buffer:

```
I151=17;
```

```
I161=17;
```

Results are written into the main data stream as the channel 34 (μ) and 35 (σ) and can be further processes by gnuplot scripts and all detectors-actuators.

For plotting and experimentation with numerical processors, the plot option 'developmental plot' is available, it includes 10 subplot and is controlled by the script 'scripts/printDeveloper.dat' (e.g. for plotting the first subplot use the sections 'if (printGraphSelector==1) plot [][-1:1] NaN t"' in this script).

Numerical processors can also operate with data read from file for post-experimental data processing. In this case, if the parameter 'usingActuators=1;' is set and an existing data file will be opened for plotting, a new data file 'dataxxxx-xxxx_NP.dat' is created and the plotting engine will work with this new file. For multiple processing steps this new file can be opened again, it will result in appearing the data file 'dataxxxx-xxxx_NP_NP.dat'. In this way the post-processing steps can be iterated multiple number of times.

Note, that all numerical processors are executed in order of their appearance. For example,

```
I161=17;
I191=34;
```

the processor '161' calculates the standard deviation and write it into the channel 34, the integrator '191' takes the value from the channel 34 and write the result into the channel 35. Thus, both processors can be executed during one run-time cycle of the DA module.

> **ATTENTION**. Numerical processors take values from data channels as 'they are', without any corrections (e.g. of decimal points, see data format in Sec. 7.8). Internal calculations are performed in the format of 'long double'.

10.6 Connecting different actuators to one or several detectors with 'and' 'or' conditions

Several detectors can be assigned to the same actuator by specifying, e.g.

```
D1=101;
D2=101;
```

Here two detectors D1 and D2 are assigned to A101, see Figure 80. They are connected by the logical 'or' operator, i.e. the A101 will

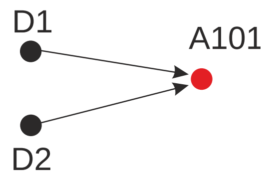


Figure 80: Several detectors assigned to the same actuator by the logical 'or' operation.

be executed only one time if any of D1 or D2 is activated. Connecting several detectors to the same actuator by the logical 'and' operation is more difficult since it requires to know the behaviour of all detectors addressing the same actuator (the number of such detectors can be changed during the run time). For these purposes the so-called 'and' actuators A151-A159 are developed. They can be addressed in this way

```
D1=151;
A151=101 1 2;
```

Here D1 first addresses A151, which calculates positive and negative responses of all specified detectors (D1 and D2). If the outputs of both D1 and D2 are equal to 1, it immediately calls the related actuator A101 on the same step (no time delay), see Figure 81.

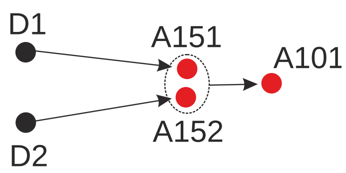


Figure 81: Several detectors assigned to the same actuator by the logical 'and' operation.

Several actuators can be assigned to the same detector by using equal detectors working in parallel, e.g.

```
I1=4;
I2=4;

P1=1;
P2=1;

D1=101;
D2=102;
```

Here two detectors D1 and D2 are parametrized to execute the same detection; one of them is assigned to A101, another to A102, see Figure 82. Alternative way to assign multiple actuators to the same detector is to use the sequential actuators (replicators) A160-A169 that can call up to 50 other actuators in the predetermined order, e.g.

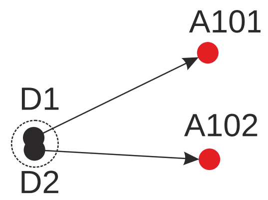


Figure 82: Using two equal detectors to assign several actuators.

```
I1=4;
P1=1;
D1=160;
A160=101 102;
A101=...
A102=...
```

Note that the parameters of A101 and A102 should be described, otherwise they will be not executed. The sequential actuators A141-A150 call one random actuator from the list.

Blocking behaviour. In cases of 'several actuators assigned to the same detector' some actuators within the same group can be sensitive to multiple calls at the same time. The text-to-speech interface or the motor/position control are example of actuators that cannot be triggered multiple number of times at the same event. Thus, these actuators have internal blocking behaviour that pass only the first activation and block all others (see Table 20 for details).

10.7 Using external software to implement computer learning strategies

The MU hardware and EIS Client can interact with external programs (e.g. for implementing the computer learning strategies) by means of files './log/outputVector.dat' and '.init/configureDA.ini'. To enable this functionality, in the file './init/init.ini' set the parameter

```
asynchronousInteractionDA=1
```

to '1'. Setting to '0' will disable this functionality. If the parameter 'asynchronousInteraction' is set to '1', on each run of processing sensor data the EIS client will first read the parameters from the file '.init/configureDA.ini', perform data processing and then write results of analysis –all components of the output vector as symbolic string in overwrite mode – into the file './init/outputVector.dat', see Figure 83.

An external program can read the file './log/outputVector.dat' (for input data) and modify the file '.init/configureDA.ini'. Note that enabling this functionality will slow down the execution of the real-time analyzing module. Increasing the timing parameters defined in 'system' → 'period between measurements' al-

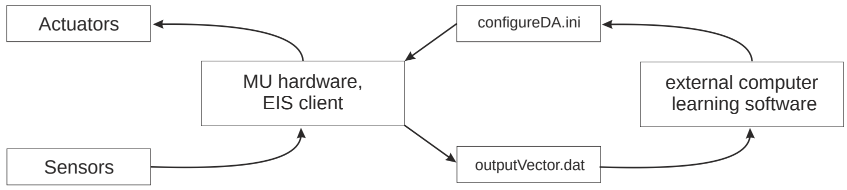


Figure 83: Asynchronous interaction between MU/EIS Client and external program.

lows reducing the computational load. Note that actuators A1-A20 can write different output values and comments into the file './log/messagesDA.txt', use this mechanism for interactions with external software.

ATTENTION. Enabling 'asynchronousInteractionDA=1' will in fact disable all internal adaptation mechanisms, e.g. for probability changing, since the system will be set on each run to the parameters specified in './init/outputVector.dat'. Use this option carefully.

10.8 Implementing complex scenarios as event-driven Bayesian networks

In several cases, the values obtained from sensors and detected by detectors D_i represent probabilistic events that can be considered as a part of complex belief network. The probabilistic way of representing complex relationships between sensor events and automated reactions is of advantage for many different scenarios.

Let us first consider the following example. The detector D1 positively detected the condition $data[i] > data[i-x]$. It can react in several ways:

1. with the probability 0.3 it can send a text message in a log file with the actuator A1, i.e. D1→(A1|0.3);
2. with the probability 0.4 it can activate some device on MU bus with the actuator A21, D1→(A21|0.4);
3. with the probability 0.2 it can activate text-to-voice interface with some text e.g. 'I like it' with the actuator A101, D1→(A101|0.2).

Corresponding to Bayesian networks, D1→(A1|0.3), D1→(A21|0.4) and D1→(A101|0.2) are independent from each other and can happen at the same time. Now we consider the same case, however from the view point of actuators. The actuator A61 'play .wav file' will be activated

1. with the probability 0.6 if D2 is active, i.e. A61→(D2|0.6);

2. with the probability 0.4 if D3 is active, A61→(D3|0.4).

This can be written as a common condition A61→(D2|0.6,D3|0.4). Both cases are represented in Figure 84. Thus, the detector-actuator

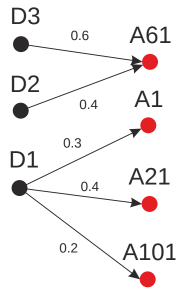


Figure 84: Example of representation of Detector-Actuator system as a Bayesian network with probabilistic A→(D) and A→(D) relations.

Bayesian network can include the nodes and transition of two types: D→(A) and A→(D). To follow the notation from Sec. 10.6 we set that

A61→(D2|0.6,D3|0.4) ⇒ D2→(A61|0.6) *or* D3→(A61|0.4),

i.e. all incoming transitions into one node are treated by the logical *or* operation (see Sec. 10.6 for the logical *and* operation).

Important difference to Bayesian networks is the event driven character of the MU/EIS Client system. All transitions should happen between acquiring new data samples. In fact, it is impossible to asynchronously activate such nodes that are not directly activated by detectors. Thus, we need to consider the step-wise probabilistic dynamics between two iterative events: 'new data available' and 'actions finished', as shown in Figure 85. One of consequences

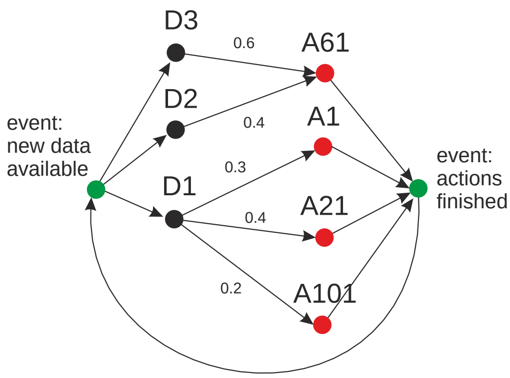


Figure 85: Step-wise dynamics of the event-driven MU/EIS Client system.

is the difficulties with calculations based on conditional probability tables defined for all possible combinations of 'true' and 'false' events. For instance, **'true' and 'false' conditions** from detector D1 trigger independent probabilistic transitions for A2 and A3

```
if (D1==true) call (A2|0.3); else call (A3|0.4);
```

shown in Figure 86. Since the probabilistic interface is determined

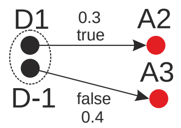


Figure 86: Assigning the detector D1 to the actuator A2 (the 'true' condition) and to A3 (the 'false' condition).

for the transition, calculation of both probabilities is independent from other. This scheme has no handlers for cases 'A2 is not activated', 'A3 is not activated', 'A2 and A3 are not activated' and similar.

Representation as the event-driven iterations, shown in Figure 85, is useful for stationary experimental systems that do not change over time. In this case we assume that the probabilistic network has a fixed structure and behavior.

If the experimental system is not stationary (it changes over time), the belief network should have some mechanisms to adapt the structure and behavior. In general these topics touch external learning approaches, discussed in Sec. 10.7. The system internally provides several mechanisms to implement adaptation in probabilistic networks, e.g. actuators A121-A140 that change the probability of some transition on the next step. Using these mechanisms

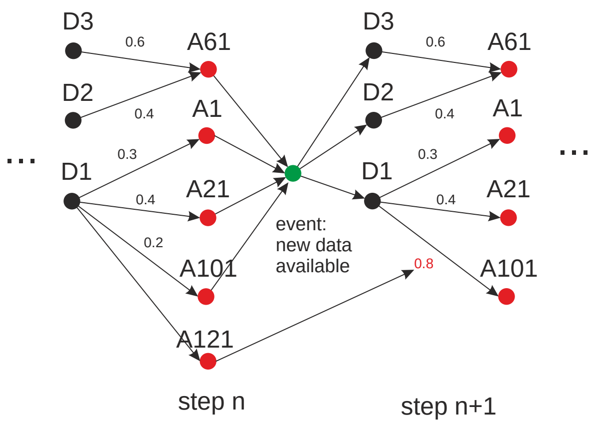


Figure 87: Bayesian detector-actuator network for non-stationary experimental system, the actuator A121 modifies the probabilistic transition on the next step.

allows modifying the reaction D→(A|p) and adapting the network to the experimental system.

Implementation. The mechanism of probabilistic transition is implemented on the stage 'selection of activities' (mapping between detectors and actuators). To enable a probabilistic transition, the mapping is considered as Dx→(Ay|z). The integer value of 'Bx=z' defines the probability of 'true' transition between 0 and 100, value of 'B-x=z' defines the probability of 'false' transition. During initialization, all values of 'Bx' and 'B-x' are assigned to 100 that defines Dx→(Ay|1) and enables non-probabilistic transition per default. To introduce a probabilistic transition, set 'Bx<100'. The random number generator based on the Random-Class of the .NET Framework returns the random value within [1-100] at each transition for each actuators. The condition 'if(random[1-100]<=Bx)' is used for probability checking.

Example. The following setting

```
I1=4;
I2=4;

P1=1;
P2=1;

D1=101;
D2=102;

B1=50;
B2=50;

A101=I like it;
A102=I hate it;
```

defines two equal detectors D1/d2 for the input channel 4 ('I1=4') with parameter 1 ('P1=1'), i.e. it defines the detection condition

$data[4][i] > data[4][i-1]$.

In case of a positive reaction, the D1/D2 will call the actuators A101 and A102 ('D1=101', 'D2=102') with the probability of 0.5 (B1=50, B2=50), i.e. D1→(A101|0.5), D2→(A102|0.5). Each actuator represents the text-to-speech interface that will say 'I like it' (A101) or 'I hate it' (A102) (with the blocking behavior).

Similar behavior can be obtained by using 'true' and 'false' conditions from D1:

```
I1=4;

P1=1;

D1=101;
D-1=102;

B1=50;
B-1=50;
```

```
A101=I like it;
A102=I hate it;
```

Not that that 'false' conditions are defined not for all detectors.

Blocking behaviour in Bayesian networks. In the above considered example, the probabilistic transition can trigger both A101 and A102 at the same time. For some actuators it does not represent a problem, however some actuators can be sensitive to multiple calls at the same time (see also the Sec. 10.6). As mentioned, such actuators have internal blocking behaviour that pass only the first activation and block all others. This blocking mechanism will interfere with probabilistic transitions, you can use this effect, for instance, for implementing the selection behavior at such nodes.

ATTENTION. Probabilistic interface is defined only to single transitions like $Dx \to Ax$, all replicator-like transitions (e.g. A160-A169) cannot be used with 'B' keys.

10.9 Implementing complex scenarios as event-driven Petri nets

In cases when reactive and probabilistic behaviour does not fit the goals of particular scenario, the DA module provides the multi-token Petri-nets-like mechanism for implementing event-driven reactions. In fact, the implemented detector-actuator coupling, shown in Fig. 88(a), can be represented in the Petri net form, shown in Fig. 88(b), taking into account specific event-driven character of the DA module. The Petri places are represented by Dx and Ax

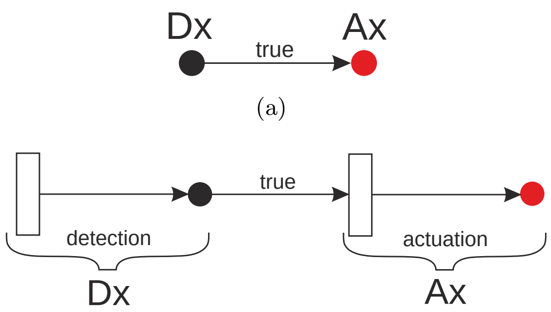


Figure 88: **(a)** the implemented detector-actuator coupling; **(b)** the corresponding Petri net form.

states, the Petri transitions are implicitly defined for all actuators and detectors, and Petri arcs are implemented in the condition mechanism. The important component of Petri nets is the token system that describes a concurrent behaviour of the network and the activated places. Since the concurrent behaviour of $Dx \to Ax$ is defined by detectors Dx, tokens are primarily used here for activation of states.

In the case of DA module, tokens are z-variables of A171-A180 (and corresponding A181-A190) – 10 different tokens – which can cake take any integer values. The actuators A211-A220 analyze the value of corresponding z (each of A211-A220 corresponds to A171-A180) and can call an actuator. Following the concept of executing $Dx \to Ax$ at one step, all A211-A220 belong the 'replicator'-type of actuators, i.e. all related activities are executed at the same step.

For example, consider the shown in Fig. 89(a) fragment of Petri net. It defines four places and transitions, which however can be understood in two different ways. The transition can happen due to activation by different detectors (without considering internal tokens), or the transition is triggered by the same detector but the selection of activities is controlled by the token, 89(b). The first case represents in fact a reactive behaviour and can be covered by a normal mapping $Dx \to Ax$. More interesting case appears when the same detector should activate different actuators based on tokens.

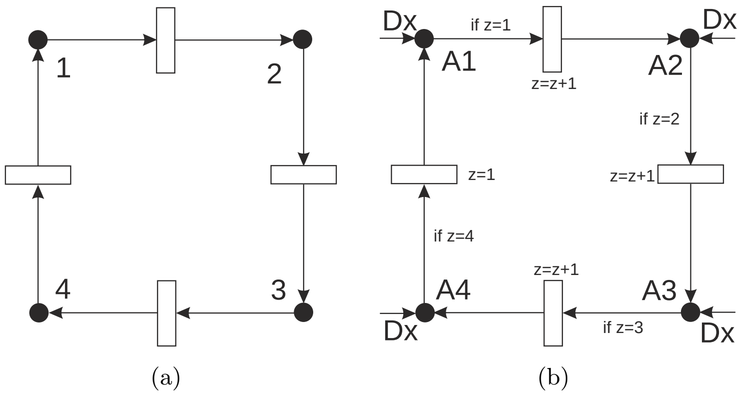


Figure 89: **(a)** Fragment of Petri net with four places and transitions; **(b)** Modified fragment taking into account activation by detectors and tokens.

The fragment shown in Fig. 89(b) can be transformed in the form, suitable for the DA module, as shown in Fig. 90. The executable code is shown below:

```
-- define detector and connect to replicator
I1=4;
P1=1;
D1=160;     call replicator A160

-- define replicators
A160=171 211;  call A171 and A211
```

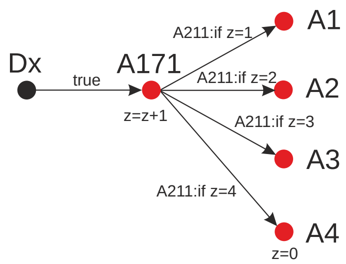


Figure 90: The fragment from Fig. 89(b) transformed in the form, suitable for the DA module.

```
A161=181 4;    call A181 and A4

-- define token system
A171=1 0;   increase z0 by 1
A181=0 0;   set z0 to 0
A211=1 1 2 2 3 3 4 161;
```

Note that by using A171=0 x; or A181=0 x; (for all A171-A190) the values of z0...z9 can be set on the value x. Taking into account multiple tokens and detectors, the Petri-nets-like behaviour provides rich possibilities for actuation.

Probabilistic transition in Petri nets. There are two types of transitions in the fragment shown above: 1) single transition like the detector – replicator $D1 \rightarrow A160$; 2) multiple transitions like the replicator $A160 \rightarrow A171$ and $A160 \rightarrow A211$, the token system $A211$. Following the rule for probabilistic system, only the single transitions can be assigned with probabilistic value by 'B' keys. Thus, $D1 \rightarrow A160$ can be used as probabilistic transition.

10.10 Exploring homeostatic feedbacks with DA module

Corresponding to Wikipeadia, 'homeostasis can be defined as the stable state of an organism and of its internal environment; as the maintenance or regulation of the stable condition, or its equilibrium; or simply as the balance of bodily functions'. Homeostatic mechanisms have many different implementations – from ecological up to cellular homeostasis with different behavior and functionality. Generally, homeostatic systems posses a number of interesting features and effects. The DA module provides mechanisms to create homeostatic feedback loops in experimental systems of different nature and to explore their properties and behavior.

Homeostatic feedback loops have several specific properties needed to take into account:

1. Homeostasis refers to the most important (vital) functionality of the systems. Destroying or essentially perturbing this functionality will reflect in destroying the system (in term of a 'living', 'common' or 'whole' system);

2. The experimental system should possesses some temporal dynamics with stable and unstable states;

3. Homeostatic feedback has usually the negative feedback character, its purpose is to return the system into a stable state;

Typical homeostatic feedback loops are created in bio-hybrid systems with living organisms, for example plans and light, microbiological organisms (e.g. yeasts) and heat, see Figure 91. Creating

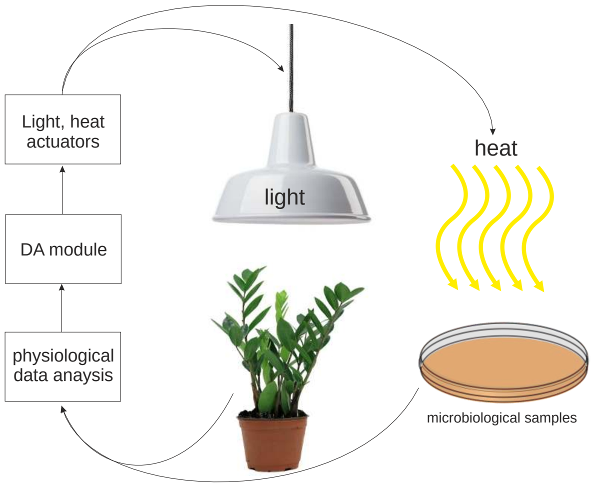


Figure 91: Example of simple homeostatic feedback loops with a plan and lights, and microorganisms (e.g. yeasts) and heat.

homeostatic feedback in non-living systems is more difficult since it is difficult to define what is a 'vital' functionality of non-living system. It is proposed to be guided in such cases by the rule 1 'destroying functionality will destroy the system' in the form of e.g. creating oscillating system with two different channels. For example, the in the water research such channels can be the 'temperature' t and 'electrochemistry' (conductivity c). The temperature and conductivity in a small range of 0-30C can be approximated by the linear equation

$$EC_t = EC_{25}[1 + a(t - 25)] \tag{33}$$

where EC_t is the conductivity at the temperature t, EC_{25} – the conductivity at $t = 25$C, a is a compensation coefficient in the range of 0.0191–0.025. Oscillating temperature will create oscillating conductivity, mapping the electrochemical detector and the heat actuator will create an oscillating behavior, see Figure 92, with a number of interesting properties.

We exemplify these ideas by several examples. The simple threshold-based feedback loop can be implemented by the following condition

```
if (data[25][i]<26C) call A41; else call A42
```

that is defined by

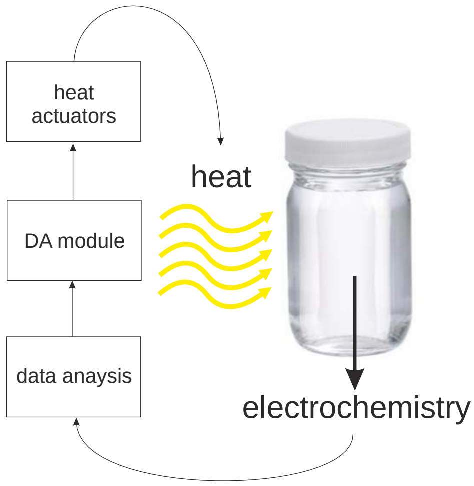


Figure 92: Example of a simple homeostatic feedback loop with oscillating electrochemical system.

```
I11=25;
P11=260000 x;

D11=41;
D-11=42;

A41=wk111*;
A42=wk000*;
```

We use the threshold-based detector D11 configured for the input channel 25 (external temperature), the D11=41 and D-11=42 define 'true' and 'false' transitions. The actuator A41 uses the connected MU system to turn on the RGB LEDs (see MU OS commands in Table 13), the A42 – to turn off these LEDs. By combining several threshold-based detectors it is possible to implement (to some extent) the proportional component of the PID controller.

More interesting behaviour will appear by using the trend detector

```
if (data[25][i]>data[25][i-1]) z++; else z--;
if (-100>z) call A41;
if (z>100) call A42;
```

which represents to some extent the integral component from the PID controller. This feedback loop is implemented in the following way

```
I1=25;
P1=1;

D1=172;
D-1=181;
```

```
A172=100 41;
A181=-100 42;

A41=wk111*;
A42=wk000*;
```

The threshold-based and the trend-based feedback loops can be combined.

The following example, see Figure 93, was used in the demonstration video to create an oscillating behaviour of LED, controlled by light and temperature sensors to keep the temperature stable at defined value. This behaviour is created by the following script:

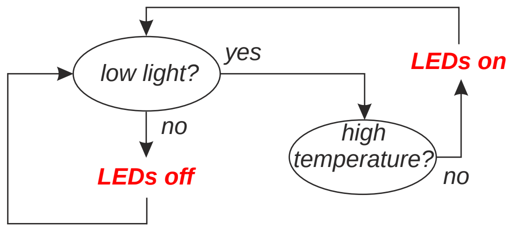
```
I11=25; // light
P11=x 5000;
D11=151;
D-11=42;

I12=24; // temperature
P12=x 243000;
_D12=151;

A151=41 11 -12; // the 'and' actuator
A41=wk111*;   // LED on
A42=wk000*;   // LED off
```

The probabilistic interface can be also used for creating the dynamical feedback loop, where external events change the probability of transition. The previous example can be rewritten as

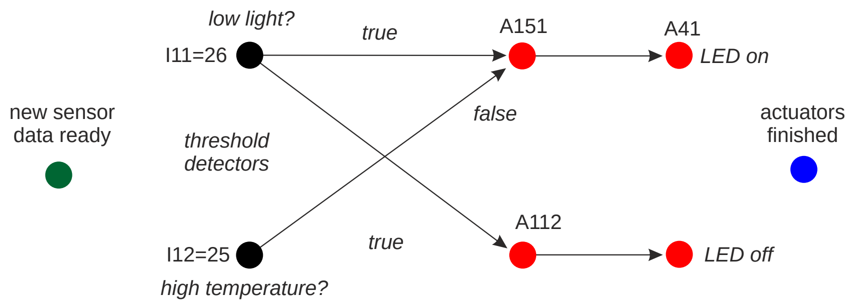
```
if (data[26][i]>data[26][i-1]) A121(A41,-1); else A122(A42,-1);
if (data[25][i]<26C) call (A41|0.1); else call (A42|0.2)
```

Here, the increasing external light (trend detector on the data channel 26) decrease the probability to switch on the LED by A41 (calling A121 with parameters A121=A41 -1), the decreasing external light will increase the probability to turn off the LED by A42 (calling A122 with parameters A122=A421 -1). The A41 and A42 will be activated by the temperature threshold 26C and activated with the probability 0.1 for the 'true' condition (A41—0.1) and with the probability 0.2 for the false condition (A42—0.2). This feedback loop is implemented in the following way

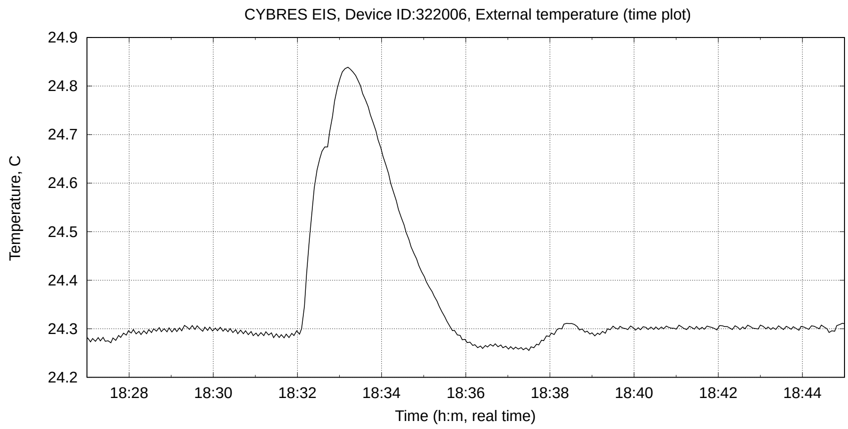
```
I1=25;
P1=1;

D1=172;
D-1=181;

A172=100 41;
A181=-100 42;
```



(a)



(b)



(c)

Figure 93: Example of homeostatic feedback loops, shown in the demonstration video, to create an oscillating behaviour of LED, controlled by light and temperature sensors to keep the temperature stable at defined value. **(a)** Block-diagram, **(b)** the network representation and **(c)** the temperature dynamics (perturbation is shown).

```
A41=wk111*;
A42=wk000*;
```

An example of creating an oscillating behavior with probabilistic interface is shown in Sec. 10.12 (the example 8).

10.11 Text-to-speech interface

The DA module uses the text-to-speech (TTS) engine to generate voice messages by A101-A120. The TTS system can generate messages on any of installed languages (voices). There are three ways to set the TTS patterns.

1) The language 'en-US' is set per default, when no other settings are encountered, the 'en-US' will be used.

2) The parameter 'textToSpeechLanguage' in the './ini/ini.ini' file determines the default language used by TTS engine. For instance

```
textToSpeechLanguage=de-DE;
```

defines the default 'de-DE' TTS patterns.

3) Finally, each message in A101-120 can define its language

```
A101=de-DE%Ich mag dich!;
A102=I like you!;
```

The A101 will use 'de-DE' TTS patterns/voice, whereas A102 will use default TTS patterns/voice.

Make sure, that at leats one of the selected languages is installed on the PC. The TTS interface is initialized if any of A101-A120 parameters is specified. The list of all available TTS patterns/voice will be shown during initialization.

For installing new TTS languages follow the Windows-Support 'Install a new Text-to-Speech language in Windows' (e.g. instructions for the Windows 10: 'Select the Start button, and then select Settings > Time & Language > Region & Language', 'Select Add a language and then choose the language you want from the list', 'After the new language has been installed select it in the Region & Language list, and then select Options', 'Under Language options > Speech, select Download').

ATTENTION. TTS language patterns/voices use the default windows file encoding. This feature was tested on Windows 10 with a few selected TTS patterns/voices only.

10.12 Examples of detector-actuator couplings

1) The detector D1 uses data from the channel 4 to detect the condition $x[i] > x[i-1]$, it is connected with the actuator A1 that in case of a positive detection will play the file './sound/forest.wav' from 0 ms to 3000 ms.

```
I1=4;
P1=1;
D1=1;
A1=./sound/forest.wav from 0 to 3000;
```

2) The detector D1 uses data from the channel 20 to detect the condition $x[i] > x[i-2]$, it is connected with the actuator A2 that in case of a positive detection will play the file './sound/forest.wav' with probability of 50%.

```
I1=20;
P1=2;
D1=2;
A2=./sound/forest.wav from 0 to 3000;
B1=50;
```

3) The detector D81 at the sample with the time stamp 18:03:27:14:35:00 will activate one time the actuator A31 that writes the message 'actuator 31 fired at %T' in the file '.log/messagesDA.txt';

```
I81=1;
P81=18:03:27:14:35:00 0 1;
D81=31;
A31=actuator 31 fired at %T;
```

4) The detector D82 starting from the time stamp 18:03:27:14:35:00 will activate 5 times the actuator A31 randomly within 60 sec interval.

```
I82=1;
P82=18:03:27:14:35:00 60 5;
D82=31;
A31=actuator 31 fired at %T;
```

5) The detector D83 starting from the time stamp 18:03:27:14:35:00 will activate the actuator A31 infinite number of times. It will start each new time randomly within the next 360 sec interval.

```
I83=1;
P83=18:03:27:14:35:00 360 -1;
D83=31;
A31=actuator 31 fired at %T;
```

6) The detector D83 and D84 starting from the time stamps '18:03:27:14:00:00' and '18:03:27:18:00:00' will activate the actuators A31 and A32 randomly within 3600 sec interval.

```
I83=1;
I84=1;
P83=18:03:27:14:00:00 3600 1;
P84=18:03:27:18:00:00 3600 1;
D83=31;
D84=32;
A31=actuator 31 (on) fired at %T;
A32=actuator 32 (off) fired at %T;
```

7) The detector D91 starting from the **computer clock time** 18:03:27:14:35:00 (all devices are disconnected) will activate the actuator A31 10 times with intervals of 5 sec.

```
P91=18:03:27:14:35:00 0 10;
D91=31;
A31=actuator 31 fired at %T;
```

8) The main pat of this example is shown in Figure 94. The actu-

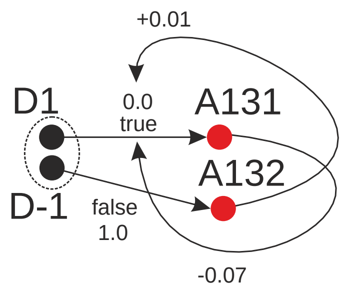


Figure 94: Example of nonsymmetric feedback loops applied to the probability of 'true' transition.

ator A170 generate random samples each 1 sec. D1 is configured on the input data channel 15. The 'true' condition call A131, the 'false' condition call A132. The actuator A131 decreases the probability of 'true' transition by 7, A132 increases it by 1. Initial probability is 0.

```
A170=1000;

I1=15;
P1=1;
D1=131;
D-1=132;
A131=1 -7;
A132=1 1;
B1=0;
B-1=100;

I2=1;
P2=1;
D2=1;
A1=\%B1;
```

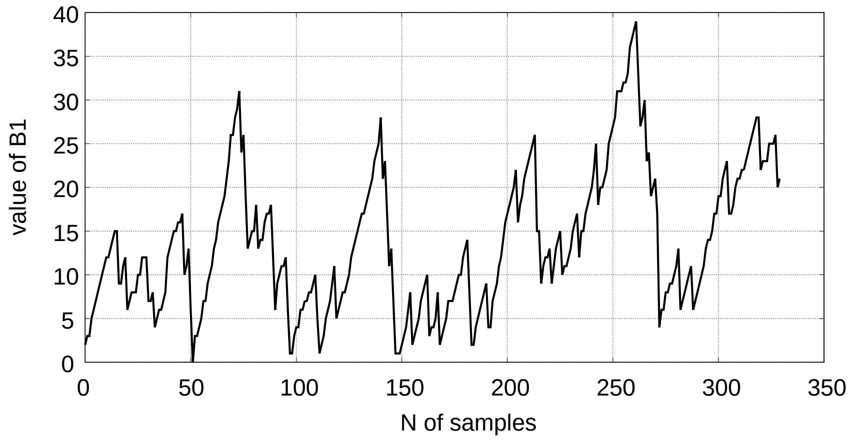


Figure 95: Dynamics of the probability of 'true' transition for the fragment shown in Figure 94.

The D2 and A1 are configured to write the values of B1 into file './log/messagesDA.txt' each time step. This script creates an oscillating behavior of the transition with B1. Dynamics of B1 is shown in Figure 95.

9) The time stamp, the mean, stDev and running average for the short-range data pipe are printed into the file './log/messagesDA.txt' each time step

```
I41=3;
P41=0 0;
D41=1;
D-41=1;
A1=\%T \%M41 \%E41 \%A41;
```

10.13 Detailed description of implemented detectors and actuators

The following Tables 19 and 20 describe the implemented detectors, numerical processors and actuators.

Table 19: Available real-time detectors and numerical processors (L – symbolic label of detectors, IP – input parameters).

L	IP	data pipe	description
			Signal level and peak detectors
D1-D5	x	short-term	simple relation detector $data[k][i] > data[k][i-x]$, where the data channel k is defined by the I parameter, and i is the index of the current sample. If x is larger than size of the data array used for analysis, $x = 1$. The 'true' condition is available at 'Dz=k' expression, the 'false' condition is available by defining 'D-z=k' (and Bx=k/B-x=k for probabilistic transitions). It is useful for detecting monotonic trends (in combination with A171-A190), counting increasing or decreasing events (in combination with A171-A180 or A181-A190) or random-signal-related detection (e.g. as background sound reactions in polyphony mode).
D6-D10	x	middle-term	the same as D1-D5 but defined for middle-term data pipe.
D11-D20	x y	short-term	the threshold detector $x > data[i] > y$. It is useful for detecting boundary values of external sensors (e.g. temperature or humidity). Setting x or y to non-numeric value will switch off the corresponding condition, e.g. P11=m 20; implements the condition $data[i] > 20$, P11=10 m; implements the condition $10 > data[i]$, P11= 10 20; implements the condition $10 > data[i] > 20$. The 'true' condition is available at 'Dz=k' expression, the 'false' condition is available by defining 'D-z=k', e.g. D11=41; D-11=42; defines actuator 41 for 'true' and 42 for 'false' conditions (and Bx=k/B-x=k for probabilistic transitions). This detector can be used for creating alarm signals, generating feedback loops, simple thermostat applications or automatic (e.g. watering) modes.
D21-D30			reserved.

D41-D60	x y	short-term	step detector implemented as ((data[k][i]-mean)> x*stDev) & ((data[k][i]-mean)/mean> y), the input parameter x represents the threshold of z score and y – the threshold of moving average; x determines steps in terms of variation between means and stDev (in z score, usually between 2 and 4), y – as variation of means (in %, usually between 30% and 80%). The filter detects positive and negative steps. The 'true' detection of positive steps are assigned to the 'Dxx' actuators, the 'true' detection of negative steps are assigned to the 'D-xx' actuators. The 'false' conditions are not detected. Note that steps placed close than 10 data samples to each other are recognized not as steps but as an increased noise (thus the noise detector can be useful in such cases). Data time horizon is limited by the short-term data pipe, i.e. the detector cannot recognize slowly-increasing steps. This detector can be used for any peak-detection, step-detection and indication purposes, in particular with the actuator A1-A20 it can write in files the values of mean, stDev and moving average $100 * abs(data[k][i] - mean)/mean$ for the selected data channel k.
D61-D80			reserved.
			Time and Random detectors
D81-D90	x y z	—	generating event at specified or random time. The time format of x 'yy:mt:dd:hh:mm:ss' (year:month:day:hour:min:sec – each with two digits representation) defines the start time, y defines the maximal time interval for generating a new random start (in sec), z is the number of iterations ('0' – infinite number of iterations, '1' – only one iteration and so on). The values of x y z should be separated by whitespace character. The time stamps are obtained from data samples. The values of y and z determine the behaviour of this function. If $z > 0$, the positive output is periodically generated z times randomly between 'yy:mm:dd:hh:ss' and 'yy:mm:dd:hh:mm:ss'+y. The next positive output will be randomly generated starting from the last firing point (not from $x + y$). After each firing, the value of z is decreased by 1. If $z = -1$ the detector operates infinite number of times. If $y = 0$ the positive output is generated z times, when the time stamp, obtained from data samples, will be the first time greater than x. The random number generator is based on the Random-Class of the .NET Framework. These functions are useful for performing activities at specific or random time points with running measurement equipment.

D91-D100	$x\ y\ z$ —	these functions are equal to D81-D90 with the difference that time stamps are obtained from computer clock once per 5 seconds, i.e. the measurement device can be in offline mode. Note, that the internal timers start and stop by initializing the system from the file './init/init.ini', in particular they start if any of P91-P100 has non-empty components (it is not necessary to define 'I' components), and they stop when all P91-P100 have empty components. Make sure to switch off these detectors if they are not required. These functions are useful for managing different activities at specified or random time points without running measurement equipment. For example, the following statement P91=18:03:27:14:35:00 0 10; D91=31; will start at 18:03:27:14:35:00 (computer clock time) and will activate A31 each minute 10 times in total.

D101-D130 $x\ y\ z$ —	periodical 'day-hour' timer, the time format of x 'yy:mt:dd:hh:mm:ss' defines the start time (use 2 digits representation, e.g. '5' should be written as '05'). Behaviour of these detectors is defined by y: $y = -1$ – the timer fires if 'ss' from x is equal to 'ss' from computer clock; $y = -2$ – the timer fires if 'mm' from x is equal to 'mm' from clock; $y = -3$ – the timer fires if 'hh' from x is equal to 'hh' from clock; $y = -4$ – the timer fires if 'hh'='hh' and 'mm'='mm' from x and from clock; $y = -5$ – the timer fires if 'dd'='dd' and 'hh'='hh' and 'mm'='mm' from x and from clock; $y = -6$ – the timer fires if 'mt'='mt' and 'dd'='dd' and 'hh'='hh' and 'mm'='mm' from x and from clock. The parameter z defines the number of iteration, the value '-1' (or any negative number) defines infinite number of iteration. Time stamps are obtained from computer clock once per 1 sec (for $y = -1$) or per 1 min (for $y < -1$). Note, that the internal timer starts and stops by initializing the system from the file './init/init.ini', in particular they start if any of P101-P130 has non-empty components (it is not necessary to define 'I' components), and they stop when all P101-P130 have empty components. Make sure to switch off these detectors if they are not required. These functions are useful for managing different periodic activities at specified time points. Use several such timers with shifted x for periodical on/off switching. For example, P101=01:01:01:00:00:00 -1 -1; P102=01:01:01:00:30:00 -1 -1; D101=1; D102=2; will execute A1 and A2 with the time interval of 30 minutes infinite number of times. Another behaviour the timer is achieved if $y >= 0$, the timer is periodically firing after y seconds (initial firing is immediately executed at a starting moment). In the following example two timers are used to execute A1 and A2 (or switch on/off some actuators) infinite number of times P101=01:01:01:00:00:00 10 -1; on timer D101=1; P102=01:01:01:00:00:03 10 -1; off timer D102=2; (here 3 sec on and 7 sec off, it starts from off time). Note that the value of x is added to the to the next firing time, i.e. the first time is calculated as $current\ time + x + y$, whereas the next firing time is $x+y$. It allows introducing a phase shift between several timers (only hour, min, sec in x are used for the phase shift). In this example P101=01:01:01:00:00:00 0 1; D101=41; P102=01:01:01:00:00:00 60 1; D102=42; A41 and A42 will be executed one time with interval of 60 sec between each other (initial firing is executed at a starting moment). The same behaviour is obtained also in this case P101=01:01:01:00:00:00 0 1; D101=41; P102=01:01:01:00:01:00 0 1; D102=42;

D131-D150		reserved.
		Numerical processors
D151-D160	short-term	it calculates the mean value $\mu = \frac{1}{N}\sum_{i=1}^{N}(data[k][i])$ for the input buffer of short-term data pipe applied to the channel k. If the short-time buffer is not fully loaded (during N first samples), the values of μ will float. For example I151=5; enables calculation of the mean value for the input channel 5, the result will be written in the channel 34.
D161-D170	short-term	it calculates the stDev value $\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(data[k][i]-\mu)^2}$ for the input buffer of short-term data pipe applied to the channel k. If the short-time buffer is not fully loaded (during N first samples), the values of σ will float. For example I161=15; enables calculation of the stDev value for the input channel 15, the result will be written in the channel 34.
D171-D180	short-term	it calculates the moving average value (in %) of current data sample $a = \frac{100*(data[k][i]-\mu)}{\mu}$ applied to the channel k. If the short-time buffer is not fully loaded (during N first samples), the values of a will float. For example I171=28; enables calculation of the moving average value for the input channel 28, the result will be written in the channel 34.
D181-D190	short-term	it calculates the z score of current data sample $z = \frac{data[k][i]-\mu}{\sigma}$ applied to the channel k (note that the denominator has only σ, not $\sigma/\sqrt(N)$, where N is the length of the moving window as defined by 'bufferSizeDataPipes' in 'init/init.ini'). If the short-time buffer is not fully loaded (during N first samples), the values of z will float. For example I181=5; enables calculation of z-score for the input channel 5, the result will be written in the channel 34.
D191-D210	short-term	it calculates the cumulative sum $\sum_{i=1}^{\infty}(data[k][i])$ of all input data applied to the channel k. This processor is useful as an integrator for repressing the results 'as one number', e.g. on the bar diagram. This processor uses the long double (8 bytes representation) variables for storing these values. For example I191=34; enables calculation of the cumulative sum for the input channel 34, the result will be written in the channel 35.

D211-D220	M	short-term	it calculates the cumulative sum $\sum_{i=1}^{M}(data[k][i])$ of input data within the moving windows of the size M, applied to the channel k. The value of M should be defined in the parameter 'P211' only one time, all D211-D220 will have the same size M of the moving window. Note, changing M is possible only by restarting the program. This processor is useful as a moving integrator for calculating cumulative values. If using it for accumulating statistical values (e.g. Z score), divide the final result by $\sqrt(M)$ in the plotting script. This processor uses the long double (8 bytes representation) variables for storing these values. If the short-time buffer is not fully loaded (during N first samples), the values of z will float. Example: I211=14; $\longrightarrow$ read ch.14 and accumulate in ch.34 P211=100; $\longrightarrow$ the buffer (moving window) size is 100 Note that M can be changed by 'P201' only at a new start of the client program.
D221-D230	M	short-term	it transforms temperature data from thermistor-sensors in containers with fluids into the degree of Celsius. Example: I221=26; $\longrightarrow$ read temperature from ch.26 (ch1), transform it and write into in ch.34
D231-D240		short-term	it calculates the skewness $skew = \frac{1/N\sum_{i=1}^{N}(data[k][i]-\mu)^3}{\left(1/(N-1)\sum_{i=1}^{N}(data[k][i]-\mu)\right)^{3/2}}$ for the input buffer of short-term data pipe applied to the channel k. If the short-time buffer is not fully loaded (during N first samples), the values of $skew$ will float. For example I231=15; enables calculation of the $skew$ value for the input channel 15, the result will be written in the channel 34.
D241			Specific numerical processor that calculates statistical data for the EIS continuous mode (the 2nd, 3rd, 4th order statistics for Impedance, Correlation and Phase), see more in Application Note 24. Analysis of electrochemical noise for detection of non-chemical treatment of fluids. It takes 18 data channels (positions 34-51 in the data file). The next available position is 52 (the maximal number of data channels is 70).
D242			Specific numerical processor that calculates statistical data for the EIS signal scope mode (the 2nd, 3rd, 4th order statistics for raw signals), see more in Application Note 24. Analysis of electrochemical noise for detection of non-chemical treatment of fluids.
D243			Specific numerical processor that calculates statistical data for the biosensor (the 2nd, 3rd, 4th order statistics for Impedance, Correlation, Phase and Temperature), see more in Application Note 24. Analysis of electrochemical noise for detection of non-chemical treatment of fluids. This processor can be also used in the EIS mode, it takes 26 data channels (positions 34-59 in the data file). The next available position is 60 (the maximal number of data channels is 70).
D244			Specific numerical processor that calculates statistical data for the biosensor (the 2nd, 3rd, 4th order statistics for Impedance, Correlation, Phase and Temperature) for data accumulated in **MU32 systems**, see more in Application Note 24. Analysis of electrochemical noise for detection of non-chemical treatment of fluids. It takes 26 data channels (positions 34-59 in the data file). The next available position is 60 (the maximal number of data channels is 70).

D245	Specific numerical processor that **re-calculates** statistical data **read from file** for the biosensor (the 2nd, 3rd, 4th order statistics for Impedance, Correlation, Phase and Temperature) and write them into _ND.dat file, see more in Application Note 24. Analysis of electrochemical noise for detection of non-chemical treatment of fluids. It takes 26 data channels (positions 34-59 in the data file). The next available position is 60 (the maximal number of data channels is 70).
D246	Specific numerical processor that **re-calculates** statistical EIS data **read from file** (the 2nd, 3rd, 4th order statistics for Impedance, Correlation and Phase) and write them into _ND.dat file, see more in Application Note 24. Analysis of electrochemical noise for detection of non-chemical treatment of fluids. It takes 26 data channels (positions 34-51 in the data file). The next available position is 52 (the maximal number of data channels is 70).
D247	Specific numerical processor that converts EEG data from MUSE-2 headband (streamed via Muse Monitor, see CSV Specification at musemonitor.com/FAQ.php) into the format used by EIS system.
D248	Specific numerical processor that converts old MIND log files (format 'N, time') into a common format used by EIS system (format 'time, N').
D249-D250	reserved.

Table 20: Available actuators (L – symbolic label of detectors, IP – input parameters (only one line of text).

L	IP	description
		Files and COM port actuators
A0	–	empty actuator.
A1-A20	*text*	write the *text* either into the file ./log/messagesDA.txt in append mode with time stamp or into the main data stream. If the first symbol of *text* starts from '&' the output will be written into the main data stream in positions after data channels 1-33 and data chennels produced by numerical processors (use this actuator carefully since it can make the output file unreadable by gnuplot scripts). If the *text* does not start by '&', the output goes to ./log/messagesDA.txt. The marks: '%T' – insert the time stamp instead of '%T'; '%D' – insert the number of calling detector; '%S' – insert the current data sample with all fields, note that '#' is the comment mark for gnuplot, thus '%S # text' can be used for generating data for gnuplot with comments. '%Bx' or '%B-x' – insert the current value of 'Bx' (probability of transition x). '%Vx' – insert the x-component of the output vector; '%W' – insert the whole output vector. '%Mx' – current mean value of the short-term buffer detected by Dx (D41-D60 should be configured); '%Ex' – current stDev value of the short-term buffer detected by Dx (D41-D60 should be configured); '%Ax' – current moving average value of the short-term buffer detected by Dx (D41-D60 should be configured); '%Zx' – insert the value of internal variable z for A171-A180;

A21-A40	*portN* *baudRate* *text*	open serial port *portN* at *baudRate*, send *text*, close the COM port (Important: this functions should not be used for the already connected device by EIS Client, use for these cases A41-A60). This function is useful for controlling any USB/COM devices that accept ASCII/Hex messages on COM ports. In particular, it can be used for controlling any of the MU and EHM-C devices, connected real-world actuators, USB relay, USB PWM robot controllers and similar devices. Make sure that the device allocates the specified serial port *portN* (e.g. COM1, COM2 COM3, ...), otherwise the action will be not executed. This actuator can be used for sending *text* as a string into the COM port. For sending numerical values use the mark '%Hx' – it writes the decimal value 'x' (not as a symbol) into the port, the mark '%Dx' executes delay in ms (implemented as Thread::Sleep() – newer use this with a large delay >500 ms). Marks '%Hx' and '%Dx' can be repeated in any order. String and '%Hx'/'%Dx' modes should not be mixed – use for this different actuators. The mark '%Cx' introduces user-comment that is ignored by the system. Examples: A21=COM5 9600 %H1 %D2 %C test example %H3; send to COM5 with 9600 baud rate the decimal number '1', wait 2 ms and send the decimal number '3', the notation %H1 %D2 %H3 is equivalent to %H1%D2%H3; A21=COM3 625000 sdf; send to COM5 with 625000 baud rate the string 'sdf'.
A41-A60	*command*	send the *command* via already opened COM port to the connected MU device (Important: this functions should be used only for the connected device by EIS Client). This function is useful for automated control of the connected MU device (see the list of available commands in Table 13). Make sure that only one *command* is issued via one actuator. These function generates error when the COM port is not opened (MU device is not connected).

Sound, music and voice actuators

A61-A80	./sound/file from x to y	it is used to play wav/mpeg/mp3 files from position x (ms) to y (ms) on both sound channels. All files can be played in parallel (polyphony). At each new call, the file will be played again independently whether it was finished in the previous call or not.
A81-A90	./sound/file from x to y	the same as A41-60, but it plays wav/mpeg/mp3 file on the right sound channel.
A91-A100	./sound/file from x to y	the same as A41-60, but it plays wav/mpeg/mp3 file on the left sound channel.
A101-A130	*text*	open the text-to-speech interface and read the text message. This group of actuators has a blocking mechanism, i.e. only the first activation during one cycle of data processing will be accepted. It is recommended to prepare short messages. Parameters *text* can be defined with the language definition as e.g. A101=en-US%I like it; (make sure this TTS voice is installed) or as A101=I like it; by using default TTS settings.

A241-A260	%Fx %Dx	it generates beep-like sound signals at the frequency F (specified in Hertz in the range 37 through 32,767) with duration D (specified in ms). This actuator is based on the windows beep function, it is synchronous (the client freezes until the sound finishes) – use only for producing short acoustic signals. For long signals combine different actuators connected with detectors for creating complex acoustic signaling system. %Fx %Dx pairs in one actuator can be iterated many times, the signal is produced at the %Dx command. Example: A241=%F200 %D200 %F3000 %D100; it produces two signals at 200Hz with duration 200ms and at 3kHz with duration 100ms. This notation is equivalent to %F200%D200%F3000%D100.

Logical, sequential and probabilistic actuators

A131-A140	x y	it changes the probability of transitions in the Bayesian network, x defines the number of 'true' transition to change with 'Bx=z' parameter ($-x$ defines the 'false' condition with 'B-x=z'), y determines the new value as $z = z + y$ (z is the old value of the transition). Boundary values $0 \leq z \leq 100$ of z limit this function. If $y = 0$, it set 'Bx=50' or 'B-x=50'.
A141-A150	$x1$ $x2$ $x3$...	it calls one random actuator from the list of Ax1, Ax2, Ax3... with equal probability for each actuator (the maximal number of parameters is 50, the minimal number is 2). Parameters of all Ax1, Ax2, Ax3... should be specified otherwise the corresponding actuator will be not executed. The call is performed on the same step. For example A141=101 102; this actuator will call randomly A101 or A102.
A151-A159	x $y1$ $y2$ $y3$...	logical 'and' actuator; this actuator consider 'true' conditions of input detectors Dy1, Dy2, Dy3 or 'false' conditions of input detectors D-y1, D-y2, D-y3 and so on (the maximal number of parameters is 50). The actuator Ax will be activated only if output states of all detectors are equal to +1 or -1. If any of y will contain a non-numeric value, this actuator will not process all remaining detectors 'Dy'. This actuator should be called only one time by any of Dy1, Dy2, Dy3,....This actuator, together with a normal logical 'or' input function, allows creating state automata and different logical behaviour of the detector-actuator mapping, see Sec. 10.6.
A160-A169	$x1$ $x1$ $x2$ $x3$...	replicator of calls; this actuator will call all enlisted actuators Ax1, Ax2,..., the order of calls is determined by the order of x (the maximal number of parameters is 50, the minimal number is 2). If any of x contains a non-numeric value, this actuator will not process all remaining Ax. Each of A160-169 has an 'enabling' register that can be controlled via A201-A210, e.g. if enabling[160]=1, the actuator A160 will be executed. All 'enabling' registers are set to 1 during initialization. For example A160=1 21 35 171; it defines the list of A1, A21, A35, A171 that will be sequentially executed by calling A160. Note, this is the replicator-like actuator, it does not use the probabilistic interface.

A170	x	it creates a random sample data stream, all fields are initialized between 0 and 100. A171 is used to emulate the physical MU device for test purposes. The value of x (in ms, $x = 1000$ is 1 sec interval) defines time interval for generating new data, setting $x = 0$ switches A170 off.
A171-A180	x y	this actuator increases the internal variable z by 1 at each call (all A171-A180 have independent z; A171-A180 and A181-A190 have pairwise shared variables, e.g. A171 and A181 share the same z). The value x is used as a threshold for the condition $z > x$, if 'true' is appeared, the actuator 'Ay' will be activated on the same step. For example 'A171=10 1;' means if the $z > 10$ is true, the actuator A1 will be called. The value of $-x$ limits z to x, this counter can be used for a fast response counter, e.g. 'A171=-10 0;' limit z to 10. This actuator is useful for counting the number of detections (from the calling detector 'D') and then providing a reaction if $z > x$. Note that parameters for the referred 'Ay' should be defined otherwise it will be not activated. All z are set to 0 when reading the file '.init/configureDA.ini'. Corresponding z can be set to 0 (or any number n) by calling A171-A180 or A181-A190 with parameters $x = 0$ and $y = 0$ (or $x = 0$ and $y = n$). For example A171=1 0; $\rightarrow if\ (z1 > 1)\ call\ A0$ A171=0 1; $\rightarrow z1 = 1$
A181-A190	x y	this actuator is similar to A171-A180, but it decreases the internal variable z by 1 at each call and check the condition $z < x$. For example 'A181=-10 1;' means if the $z < -10$ is true, the actuator A1 will be called. Combination of both actuators A171-A180 and A181-A190 allows implementing the trend detectors, see Sec.10.10. This actuator also supports the limiting behavior of z, e.g. the value of $+x$ limits z to x, this counter can be used for a fast response counter, e.g. 'A181=10 0;' limit z to -10. Corresponding z can be set to 0 (or any number n) by calling A181-A190 with parameters $x = 0$ and $y = 0$ (or $x = 0$ and $y = n$). For example A181=-1 0; $\rightarrow if\ (z1 < -1)\ call\ A0$ A181=0 1; $\rightarrow z1 = 1$
A191-A200	x	this actuator executes external command in CMD (Command Prompt) mode, in particular it can start any external program with specific parameters, change windows environment and similar activities that are usually performed in the CMD window. Output of CMD execution is indicated as the output and can be logged.

A201-A210	$x\ y$	it changes the 'enabling' registers for A160-A169, y defines which from A160-A169 is targeted, $y = 1$ sets the register 'enabled' and $y = 0$ sets the register 'disabled'. These actuators are useful for switching ON/OFF the group of actuators, e.g. by periodical timer or by any of detectors. For example A201=160 1; A202=160 0; the actuator A201 enables the A160 (i.e. all actuators enlisted in A160 will be executed) and A202 disables the A160 (i.e. all actuators enlisted in A160 will be not executed). The actuators A201-210 are executed only if $x = [160..169]$ and $y = [0, 1]$.
A211-A220	$x1\ y1\ x2\ y2...$	it implements the token transitions based on z variables from A171-A180 (in total 10 different tokens-variables). Each pair $x1\ y1$ represents the conditions $if\ (z == x1)\ call\ y1$. The number of $x\ y$ pairs is limited by 50. For example, the expression A211=1 1 2 160; means $if\ (z1 == 1)\ call\ 1$ $if\ (z1 == 2)\ call\ 160$ Note, this is the replicator-like actuator, it does not use the probabilistic interface.
A221-A230	$p1 - p8$	it controls the plotting behaviour of the client program. Parameters p define the following GUI control elements (index 0 means the first elements, negative values of parameters or values outside of ranges do not change corresponding control elements): p1: index of the 'plot' combobox p2: index of the 'output' combobox p3: index of the 'channels' combobox p4: index of the 'filter' combobox p5: index of the 'plot all points' combobox p6: 0 – 'online plot' checkbox 'off', 1 – 'on' (checked) p7: 1 – single plot (equivalent to a single press of the 'plot' button) p8: graph title (whole text after p7), empty title will be also accepted Example: A221=9 4 0 0 2 0 1 Control Attempt,;
A231		it initializes all internal variables used in statistical processing, reset the first measurement time, update time in the device and set up new data file. This function is useful for a new run of statistical calculations with 'clean' settings.

11 Others

11.1 Delivery, warranty and additional equipment

MU EIS system has several versions and additional subsystems/electrodes and typically includes:

1. the measuring module
2. electrodes with connectors
3. 10x 15 ml measuring containers
4. one holder for two measuring containers
5. one USB B – USB A cable
6. user manual (downloadable online from the manufacturer home page)

The warranty covers the measuring module and is twenty four months (two years) from the date of sale. This warranty does not cover electrodes and any additional supplies like cable, holder or containers.

Address:


```
Cybertronica Research (CYBRES GmbH)
Melunerstr. 40
70569 Stuttgart
Germany
+49-711-41001901
info@cybertronica.de.com
www.cybertronica.de.com
```


Additional equipment is available, check for more information

```
www.cybertronica.de.com/products
www.cybertronica.de.com/products/MU-EIS-spectrometer
```

11.2 Additional literature

Additional descriptions, application notes as well as literature can be found at

`www.cybertronica.de.com/products`
`www.cybertronica.de.com/products/MU-EIS-spectrometer`

11.3 Known Issues

– *Thermostat is unstable or does not reach the set temperature.* **Reason:** The set temperature is too low or too high, the USB powering does not provide enough current. **Resolution:** set the sample thermostat to 4-5°C (the PCB thermostat – 10-12°C) above the ambient temperature, use the active USB 3.0 hub, avoid using long USB cable ($> 1,5$ meters).

– *Strong difference in amplitude of signals between channels.* **Reason:** There is always a small difference between channels explained by a variation of electronic and mechanic components. There is also a transient process in the begin of each measurement, when temperature of containers is not equalized, and dynamics of channels undergoes different perturbations. Large and persistent difference can be explained by a) different fluids (chemical contamination of fluids); b) problems in mechanical connector; c) problem of electrodes or electronic elements. **Resolution:** a) wait until the transient process is finished; b) set the calibration coefficients to compensate the difference; b) change the fluids; c) re-connect electrodes. When the problem persists, contact the manufacturer.

– *Almost all spectral characteristics are decreasing with frequency.* **Reason:** This is a normal effect related to limited bandwidth of analog components. **Resolution:** perform full calibration over the whole frequency range.

– *Firmware updating program cannot find the 'Bootload_Utils.dll'.* **Reason:** Currently unknown, it appears seldom on some computers. **Resolution:** The file 'Bootload_Utils.dll' should be in the same directory as the booloader program, the user should have the access rights for this directory. When the problem persists, try to use the firmware updating program on another computer.

– *One or several plots do not work in online mode, only empty graphs are shown. Data can be plotted only in offline mode.* **Reason:** Wrong system data/time is set (e.g. typical problem with summer/winter time). **Resolution:** Connect to PC/laptop with a properly set data/time and press the button 'set time' in the section 'output'.

– *When the client program (only with the hardware MU3.3/3.4) is crashed, the client program cannot be canceled.* **Reason:** Handling of USB-COM protocols by Windows driver. This error happens very seldom and it was noted primarily on Windows 10. **Resolution:** Turn off the MU3.3/3.4 device and then turn on it again (or to remove and then to insert the USB cable).

11.4 Published works about this system

References

[1] J. Dennis, R. Schnabel, Numerical methods for unconstrained optimization and nonlinear equations, Prentice-Hall Series in Computational Mathematics, Englewood Cliffs, 1983. doi:10.1137/1.9781611971200.

[2] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipes in C (2Nd Ed.): The Art of Scientific Computing, Cambridge University Press, New York, NY, USA, 1992.

[3] B. M. Taylor, A multi-way correlation coefficient, arXiv preprint arXiv:2003.02561 (2020). doi:10.48550/arXiv.2003.02561.

[4] S. Kernbach, Supernatural. Scientifically proven facts (rus), ISBN: 978-5-906789-00-6, Algorithm. Moscow, 2015.

[5] S. Kernbach, O. Kernbach, I. Kuksin, A. Kernbach, Y. Nepomnyashchiy, T. Dochow, A. Bobrov, The biosensor based on electrochemical dynamics of fermentation in yeast *Saccharomyces Cerevisiae*, Environmental Research 213 (2022) 113535. doi:10.1016/j.envres.2022.113535.

[6] S. Kernbach, Electrochemical characterisation of ionic dynamics resulting from spin conversion of water isomers, Journal of The Electrochemical Society 169 (6) (2022) 067504. doi:10.1149/1945-7111/ac6f8a.

[7] S. Kernbach, Distant monitoring of entangled macro-objects, NeuroQuantology 17 (3) (2019) 19–42. doi:10.14704/nq.2019.17.3.1977.

[8] S. Kernbach, V. Zamsha, Y. Kravchenko, Experimental approach towards long-range interactions from 1.6 to 13798 km distances in bio-hybrid systems, NeuroQuantology 14 (3) (2016) 456–476. doi:10.14704/nq.2016.14.3.917.

[9] S. Kernbach, Replication attempt: Measuring water conductivity with polarized electrodes, Journal of Scientific Exploration 27 (1) (2013) 69–105.

[10] S. Kernbach, I. Kuksin, O. Kernbach, On accurate differential measurements with electrochemical impedance spectroscopy, WATER 8 (2017) 136–155. doi:10.14294/WATER.2016.8.

[11] S. Kernbach, Exploration of high-penetrating capability of LED and laser emission. Parts 1 and 2 (rus), Nano- and microsystem's technics 6,7 (2013) 38–46,28–38.

[12] S. Kernbach, The minimal experiment (rus), IJUS 4 (2) (2014) 50–61.

[13] S. Kernbach, I. Kuksin, O. Kernbach, A. Kernbach, On metrology of electrochemical impedance spectroscopy in time-frequency domain, IJUS 143–150 (5) (2017) 62–87. doi:10.17613/zh5b-jd94.

[14] H. Hamann, S. Bogdan, A. Diaz-Espejo, L. Garcia Carmona, V. Hernandez-Santana, S. Kernbach, A. Kernbach, A. Quijano-Lopez, B. Salamat, M. Wahby, Watchplant: Networked bio-hybrid systems for pollution monitoring of urban areas, ALIFE 2021: The 2021 Conference on Artificial Life, 2021. doi:10.1162/isal_a_00377.

[15] S. Kernbach, Biophysical effects of the circular poynting vector emitter, IJUS 19-20 (6) (2018) 78–97. doi:10.17613/sp8t-gz69.

[16] S. Kernbach, Device for measuring the plant physiology and electrophysiology, IJUS 4 (138) (2016) 12–13. doi:10.17613/4xyq-8z28.

[17] S. Kernbach, O. Kernbach, Programmable Phantom Effect (rus), IJUS 10 (3) (2015) 19–31.

[18] S. Kernbach, V. Zamsha, Y. Kravchenko, Long and super-long range device-device and operator-device interactions, IJUS 1 (1) (2013) 24–42.

[19] S. Kernbach, O. Kernbach, On precise *pH* and *dpH* measurements (rus), IJUS 5 (2) (2014) 83–103.

[20] S. Kernbach, O. Kernbach, Detection of ultraweak interactions by precision dph approach (rus), IJUS 9 (3) (2015) 17–41.

[21] S. Kernbach, O. Kernbach, Impact of structural elements on high frequency non-contact conductometry, IJUS 12-13 (4) (2016) 47–68.

[22] S. Kernbach, O. Kernbach, Reliable detection of weak emissions by the EIS approach, IJUS 14 (4) (2017) 65–79.

[23] H. Hamann, M. Soorati, M. Heinrich, D. Hofstadler, I. Kuksin, F. Veenstra, M. Wahby, S. Nielsen, S. Risi, T. Skrzypczak, P. Zahadat, P. Wojtaszek, K. Støy, T. Schmickl, S. Kernbach, P. Ayres, Flora robotica – an architectural system combining living natural plants and distributed robots, Proceedings of ECAL 2017: the 14th European Conference on Artificial Life (2017) 16 doi:10.48550/arXiv.1709.04291.

[24] R. Thenius, D. Moser, J. Varughese, S. Kernbach, I. Kuksin, O. Kernbach, E. Kuksina, N. Mišković, S. Bogdan, T. Petrović, A. Babić, F. Boyer, V. Lebastard, S. Bazeille, G. Ferrari, E. Donati, R. Pelliccia, D. Romano, G. Van Vuuren, C. Stefanini, M. Morgantin, A. Campo, T. Schmickl, subCULTron - cultural development as a tool in underwater robotics, in: P. R. Lewis, C. J. Headleand, S. Battle, P. D. Ritsos (Eds.), Artificial Life and Intelligent Agents, Springer International Publishing, Cham, 2018, pp. 27–41. doi:10.1007/978-3-319-90418-4_3.

[25] L. García-Carmona, S. Bogdan, A. Diaz-Espejo, M. Dobielewski, H. Hamann, V. Hernandez-Santana, A. Kernbach, S. Kernbach, A. Quijano-López, N. Roxhed, B. Salamat, M. Wahby, Biohybrid systems for environmental intelligence on living plants: Watchplant project, in: Proceedings of the Conference on Information Technology for Social Good, GoodIT '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 210–215. doi:10.1145/3462203.3475885.

[26] S. Kernbach, Electric-field-coupled oscillators for collective electrochemical perception in biohybrid robotics, Bioinspiration & Biomimetics (2022). doi:10.48550/ARXIV.2203.15786.

[27] S. Kernbach, Phytosensor (plant-device system), biohybrid interface device, technical presentation. doi:10.13140/RG.2.2.11591.78248.
URL https://cybertronica.co/download/Phytosensor_technical_presentation_CYBRES.pdf

[28] I. Jerman, P. Dovc, V. Pericek Krapez, Impedance spectrometry research of uhd solutions - a pilot study, International Journal of High Dilution Research 17 (2018) 9–10. doi:10.51910/ijhdr.v17i2.931.

[29] M. Wahby, M. K. Heinrich, D. Hofstadler, J. Petzold, I. Kuksin, P. Zahadat, T. Schmickl, P. Ayres, H. Hamann, Robotic sensing and stimuli provision for guided plant growth, Journal of Visualized Experiments (07 2019). doi:10.3791/59835.

[30] H. Hamann, M. Wahby, T. Schmickl, P. Zahadat, D. Hofstadler, K. Stoy, S. Risi, A. Faina, F. Veenstra, S. Kernbach, I. Kuksin, O. Kernbach, P. Ayres, P. Wojtaszek, Flora robotica - mixed societies of symbiotic robot-plant bio-hybrids, in: 2015 IEEE Symposium Series on Computational Intelligence, 2015, pp. 1102–1109. doi:10.1109/SSCI.2015.158.

[31] S. Kernbach, I. Volkov, The bioscope: two replications, IJUS 7 (3) (2015) 34–50. doi:10.17613/t18x-s748.