Application Note 28. Using phytosensor in precision agriculture, vertical farms, hydroponics and agricultural AI applications

Serge Kernbach

***Abstract*—This application note describes the use of phytosensors in various agricultural systems such as vertical farms, precision agriculture (PA), controlled environment agriculture (CEA) or hydroponics that require control of light, irrigation, fertilization and other parameters. The phytosensor implements fixed-protocol and feedback-based growth schemes. The same approach can be applied to water/air treatment systems. Access to sensor data obtained from plant physiology, soil and environmental sensors as well as the possibility of phytoactivation allow the use of this system for agricultural AI applications and machine learning techniques. The complexity of agricultural use is low and suitable for untrained personnel. The system features several mechanisms to increase safety in real-world applications. AI use offers a full spectrum of different sensor/analytic tools and Python support.**

firmware version: $>$ 1189.49; client version: $>$ 1.4

I. Introduction

The application note 28 (AN28) refers to the feedback-based scheme, shown in Fig. 1. Such system can be used to control

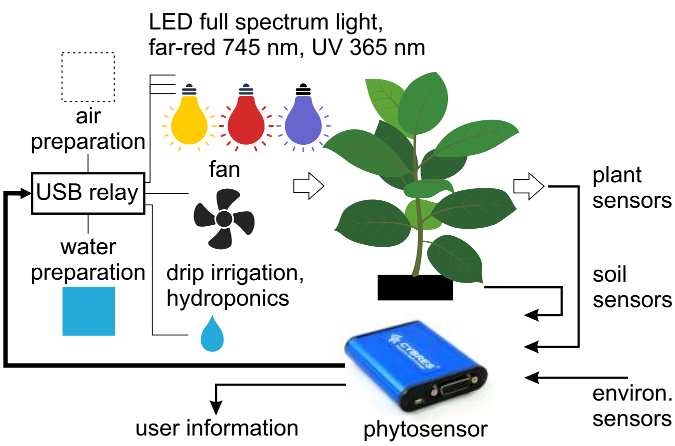


Fig. 1. Using the phytosensor in a closed-loop system.

different agricultural actuators such as light, irrigation or fertilization with fixed-growth (based on periodical timers) and feedback-based (based on sensor data) protocols. Water-/air-preparation, e.g. pH control, adding fertilizers, ozonation/AOP (advanced oxidation process), aeration/OMB (O_2 microbubbles), magnetic treatment and other approaches, can also use the same scheme and combine it with a bio-feedback 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 (artificial intelligence) schemes. The AN28 shortly describes agricultural usage in Sec. II, water-/air- preparation in Sec. III, AI applications in Sec. V, and provides several details of the power management module in Sec. VI.

CYBRES GmbH, Research Center of Advanced Robotics and Environmental Science, Melunerstr. 40, 70569 Stuttgart, Germany, Contact author: serge.kernbach@cybertronica.de.com

A. *Overview of sensing*

The phytosensing system measures and records multiple physiological and environmental parameters, see Table I and the user manual [1]. Capabilities of sensing depends on used elec-

TABLE I
Overview of measured parameters and analytic tools.

parameters	description
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 $\pm$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
light, humidity, temperature	APDS-9008-020, HIH-5031-001, LM35CA
EM emission	450Mhz-2.5Ghz RF power meter, MAX2204
soil humidity, temperature	capacitive-based sensor, CYBRES
CO_2, PM1-2.5-10	SCD4x, accuracy $\pm$(40ppm+5%); SPS30, accuracy 10%, CENSIRION
I2C sensors	different digital external sensors
water sensors	e.g. conductivity, pH, dissolved oxygen, etc.
	real-time analytic tools
regression/spectral/correlation/statistical analysis	

trodes, connected to the measurement unit (MU) with real-time operating system. The phytosensor also implements different analytic approaches, required to process obtained data in real time – totally 37 physical data channels and up 80 synthetic ones with the sampling rate 1-99 sps. The measured data can be accessed via ASCII communication protocol and the client program. For python applications the MU implements a named pipe mechanism, see more details in the user manual.

B. Overview of actuation

The MU system can directly control up to 6 Solid State Relays (SSR), Electromechanical Relays (EMR), high power MOSFETs and other external switching devices. 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, 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 via USB, see Table III. For controlling different actuators, the phytosensor requires the power management module, shown in Fig. 7, sensing and output channels of both devices are shown in Fig. 2.

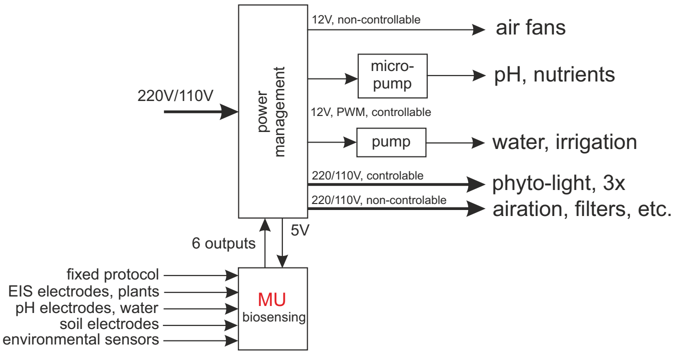


Fig. 2. Using phytosensor with the power management module.

C. Measuring physiology of plants in real time

Real-time measurement of plant physiology in vertical farms can utilize different strategies and generally depends on the used species, size of facility, age of plants and several other factors. Tested are two main approaches for productive plants.

1) Small plants: using dedicated plants for electrophysiological measurements (e.g. tissue impedance). Such an approach involves a homogeneity considerations for the used plants and can be applied for leafy green, microgreens and similar types of production, see Fig. 3(a). The selected plants can be used for measurements for a longer period than it is used for productive purposes (e.g. microgreens can be harvested every week, plants for measurements can be used for measurements several weeks in the same facility). The same approach can be applied to plant roots (see [2], [3]), especially wheat and wheatgrass production, by using schemes proposed in these works. For more general overview of impedance spectroscopy for plant science we can recommend [4].

2) Production of large plants such as tomato or other similar species typically takes a considerably longer time than microgreens and thus can support multiple types of sensors installed directly on plants, see Fig. 3(b). The measurement plant has the same developmental stage as all other plants; this increases the reliability of measurement data. Two-tree measurement points (phytosensors) are enough for 75-100m^2 of vertical production facility. Tested are also one-point measurements, however here the plant for measurement should be carefully selected.

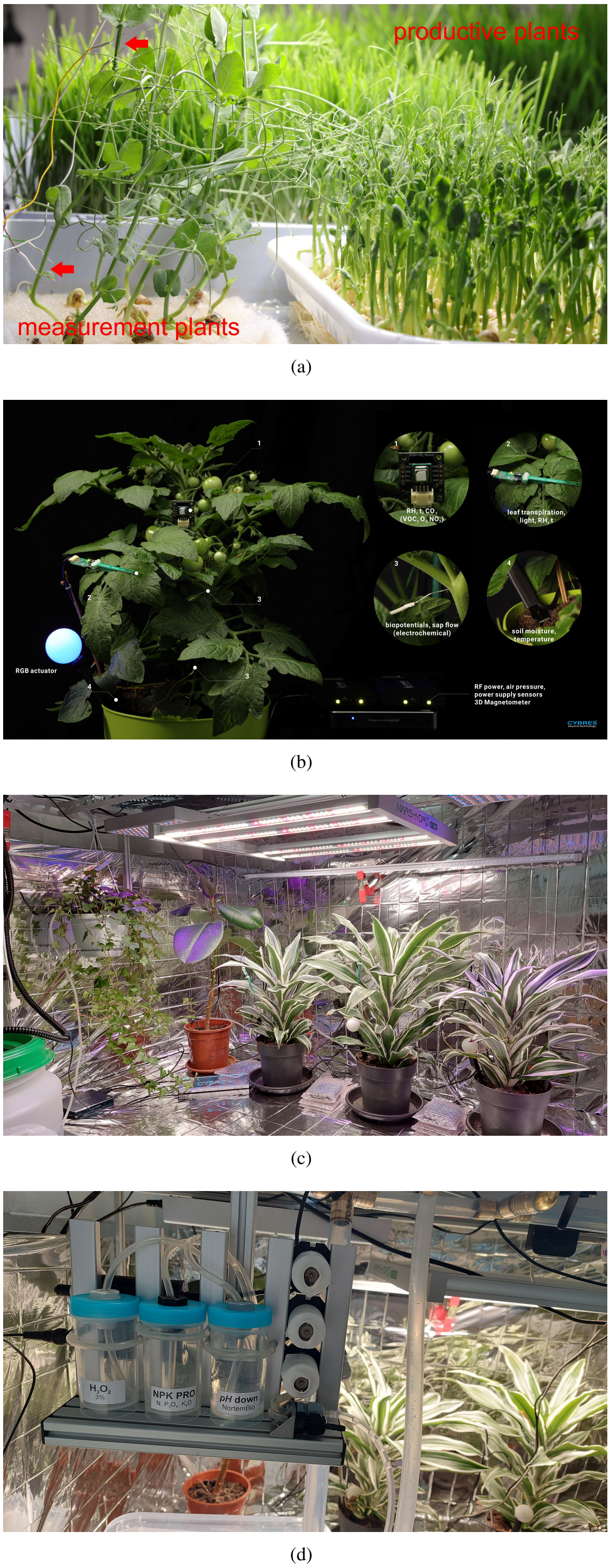


Fig. 3. **(a)** Strategy 1 for productive plants: measuring electrophysiology of small plants, e.g. for microgreen production; **(b)** strategy 2 for productive plants: measuring physiology of large plants, such as tomato, with multiple sensors; **(c,d)** Example of measuring non-productive plants and controlling fluidic system.

The same approach can be used for different non-productive plants, see Fig. 3(c), where the phytosensor controls the irrigation, primary light, additional light and pH/nutrients of the setup with plants, see Fig. 3(d). Outdoor applications are a bit different than indoor applications, since they involve such aspects as a long-range communication (see Sec. V-A), influence of environment on sensors and plants (rain, UV radiation from sun exposure, wind), but generally all sensors can be also installed in outdoor plants considering protection from a direct exposure by rain and strong UV emission, see Fig. 4.

Fig. 4. Example of outdoor application for measuring physiology of tomato (with protection from a direct exposure by rain and strong UV emission).

II. Agricultural Usage

A. Connecting and configuring the phytosensor

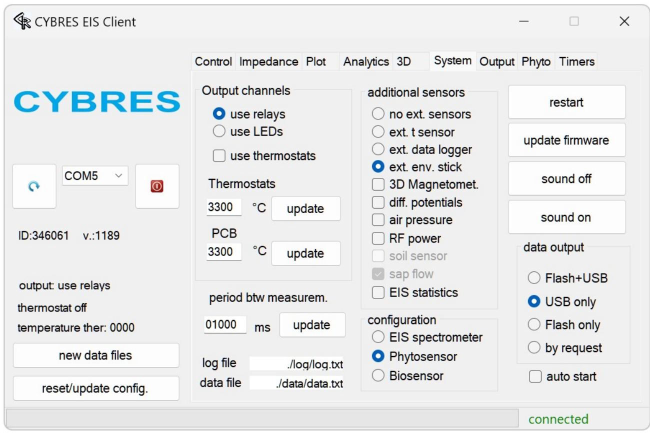
Agricultural usage has a minimal complexity and consists of three following steps.

Step 1. Connect the phytosensor to USB from PC, start the client program. Set up the configuration: 'phytosensors', output channels: 'use relays', additional sensors: the used electrodes, see Fig. 5(a). To prevent damaging external equipment, always set 'using relays' when the power management module is connected to the phytosensor.

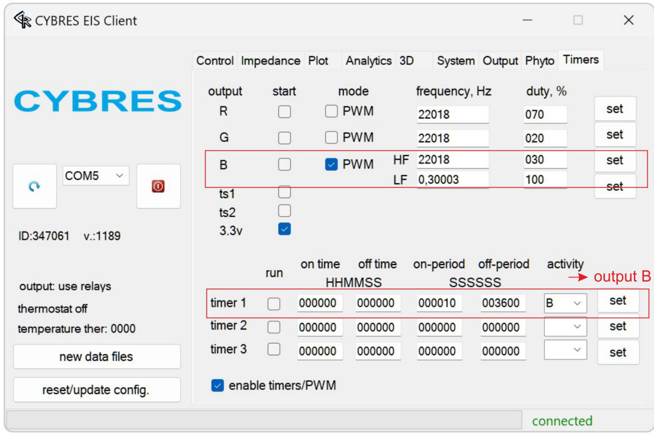
Step 2. Configure the output channels (depending on the available and used equipment). It is recommended to use 12V water/fertilizer/pH pumps (use PWM outputs with 16-22kHz modulation on channels R,G,B to set up the speed of pumps), 220V/110V devices (e.g. phyto-light) can use channels ts1, ts2, 3V, see Fig. 5(b). IMPORTANT: Disconnect the power management module during all preparatory steps (e.g. configuration or firmware update). If using fixed growth protocols: setup the periodical timers 1-3 for ON/OFF at the specified time.

Step 3. This step depends on the selected controlling strategy: fixed-growth protocols – use timers from the step 2; feedback-based protocols – configure controllers, see Fig. 5(c) or DA/Python scripts, as described in Sec. V.

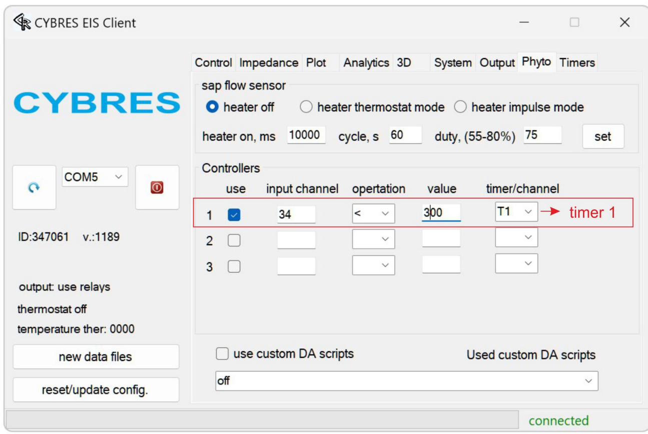
Final step. If operating in **autonomous mode**: disconnect the phytosensor from PC and connect to USB power supply from the power management module, the phytosensor will start autonomous operation (all settings are stored in internal



(a)



(b)



(c)

Fig. 5. Three steps for configuring the phytosensor: **(a)** system and additional sensors; **(b)** output channels and timers; **(c)** embedded controllers.

Example in (b,c) shows setting for automatic irrigation (controller 1 → timer 1 → output channel B), where low soil moisture (<300) triggers the *Timer1* that turns on the water pump for 10 sec (and prevents using water pump during next hour) on the *output channel B* with PWM duty 30% (12V water pump with adjustable pressure), see description in text.

memory). If operating in **client mode**: this mode is used e.g. to provide data to external systems such as AI applications, data collection systems, environmental monitoring networks, etc. – leave the system connected to PC.

B. Fixed periodical protocols

This mode uses periodical timers, see Fig. 5(b). All timers run independently of the 'start'-'stop' of measurements, each timer can operate on any output channel (even the same output channel can be operated by multiple timers).

- In the 24h-mode the timer starts if 'on'-/'off'-times are set and 'on'-/'off'-periods are zero – it means the timer fires ON and OFF at the specified 24h-time.
- The periodical mode starts if 'on'-/'off'-periods are set and 'on'-/'off'-times are zero – it means the timer fires a periodical ON and OFF endless.
- If both 'on'-/'off-'times and periods are set, the timer operates periodically only within the specified 24h-time period.

Note: 1) the value HHMMSS is limited by 245959 (=24 hour), the value SSSSSS is limited by 65535 sec. (=18.2 hours); 2) periodical mode has higher priority than 24h mode; 3) minimal time can be >1 sec.; 4) be careful with 'this day'/'next day' conditions for 24h timer.

C. Feedback-based protocols

This mode is implemented in three ways:

- **with embedded controllers** in autonomous mode (without PC and client program), see Sec. II-D.
- **with DA/Python scripts** (with PC and client program) More complex protocols require DA/Python scripts for data processing and the client program, see Sec. V.
- **direct control by external systems**, (e.g. with Raspberry/Rock PI), by writing command in the COM port, see Sec. V.

Please contact manufacturer for more details and custom protocols for different plant species.

D. Embedded controllers

They are used for autonomous executing of simple feedback-based protocols and implement the scheme **controllers** $\rightarrow$ **timers** $\rightarrow$ **output channels**. Controllers operate over real-time data channels from the Table II with a simple condition:

```
if (data-channel) condition (value)
then (turn ON output)
else (turn OFF output)
```

For *output* any of timers or output channels can be used, the *condition* has different implementations (e.g.'=','>','<','$|a| < b$', etc.). For example, Fig. 5 shows an automatic irrigation

```
if (data34[i]<300) then run(Timer1)
```

where *Timer1* turns on the water pump for 10 sec (and prevents using water pump during next hour) on the *output channel B* with PWM duty 30% (12V water pump with adjustable pressure). Note that embedded controllers work with real-time

TABLE II
FIELDS IN OUTPUT DATA STREAM (ASCII STREAM FROM COM PORTS, FILES AND NAMED PIPES).

field[1]	description
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	
19-21	magnetometer data on axes X, Y, Z (optional, depends on hardware)
22-24	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	I2C sensor: soil moisture (with the phytosensor electrodes advanced)
35	I2C sensor: soil temperature (with the phytosensor electrodes advanced)
36	I2C sensor: ambient light (with the phytosensor electrodes advanced)
37-43	empty (reserved for different I2C sensors)
Real-time synthetic data channels prepared by client program	
44,45	reserved for statistical package/DA module to encode the temperature of fluids (t-ch1, t-ch2)
46-69	synthetic (virtual) sensors for measuring electrochemical noise, available only in the EIS mode with the statistical package 'EIS statistics' (it can be turned on/off by users)
70-80	reserved

1 – firmware v.1189.49 supports channels 25, 26, 27, 28, 34 for using in embedded controllers

TABLE III
OVERVIEW OF AVAILABLE OUTPUTS AND THEIR COMMANDS FOR ON/OFF AND PWM OPERATIONS, X - 1 OR 0, Y - ANY SYMBOL, SEE DESCRIPTION IN [1].

outputs description of MU system

output	available modes	implementation	Current	Notes	Shared usage
R	ON/OFF, PWM	n-channel MOSFET	1A	1;2	RGB LEDs
G	ON/OFF, PWM	n-channel MOSFET	1A	1;2	RGB LEDs
B	ON/OFF, PWM	n-channel MOSFET	1A	1;2	RGB LEDs
ts1	ON/OFF, current, PWM[7]	n-channel MOSFET	1A	1;2	thermostat 1
ts2	ON/OFF, current, PWM[7]	n-channel MOSFET	1A	1;2	thermostat 2, thermal sap flow sensor, IR LED
3.3V	ON/OFF, PWM[7]	power switch	3.3V, 0.3A	3	RGB LEDs, green front LED, thermostats, I2C sensors, thermal sap flow sensor

commands to control output channels

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
	wl,wm,wn,wv			get short status
	wp,wu,wf,wg			get full status

commands to control periodical timers

N	24h mode	periodical mode	activity, start/stop	Notes
1,2,3	**tq, tw**	**tz, tu**	**ta, to**	

commands to control embedded controllers (embCon)

N	set i/o channels	set compare value	start/stop	Notes
1,2,3	**hm, hn**	**hl**	**hk**	

1 – 1A max. current if using external power supply, see Fig. 8
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

data and only run when measurements are started. When measurements are stopped, embedded controllers are also stopped.

Embedded controllers can implements most of simple operations related to irrigation, pH up and down, control of temperature and supplementary light.

III. WATER-/AIR- PREPARATION

Water-/Air- preparation follows the same approach with fixed or feedback-based protocols, where the feedback can be taken from any plant- or water- parameters (see Table II) in real time. For instance, the simplest water thermostat (without embedded PID controller) is implemented by

```
if (data32[i]<22000) then run(outputR)
```

where the heater is connected to the *output channel R*, and the temperature is set to 22.000 degree C. For more details please follow other application notes and online presentations, see Sec. VII.

IV. SAFETY MECHANISMS

The phytosensor implements several safety mechanisms:

1. If connections (USB power or data) between the phytosensor and power management module are lost, all high power/PWM outputs of the power management module are off;
2. After reboot (e.g. surge on 220V/110V power line), the phytosensor returns to the previous settings of timers/controllers/output channels and measurements (if 'auto start' is set), but all timers will start counting anew;
3. PWM duty is internally verified each time before using it (to prevent using wrong pump speed);
4. The phytosensor has 5V powering with protection of overcurrent;
5. Galvanic isolation between output signals and EMR/SSR.
6. The phytosensor and power management module have the protection class IP44, so make sure that both devices do not come into contact with water in case of damage to the irrigation system;

7. Always disconnect the phytosensor from the power management module if setting new parameters, updating the firmware, testing DA/Python scripts or performing similar operations.

It needs to underline:

1. **CYBRES is making efforts to improve and further develop the safety mechanisms. However, use with any real-world actuators such as lighting or irrigation systems is the responsibility of the customer.**

2. **The installation and further operation of the phytosensing system may only be carried out by certified and trained personnel in compliance with local legal regulations.**

3. **CYBRES does not assume any liability arising out of the application or use of any product, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages.**

V. AI APPLICATIONS

A. *Getting real-time data from the phytosensor*

There are three ways to obtain data from the phytosensor:

- reading ASCII data from COM ports (possible on any platforms and operating systems), see description of ASCII commands and data formats in [1].
- using named pipe mechanisms (possible on Windows OS), suitable e.g. for python scripts, see description in [1].
- reading .dat files, written by client (possible on Windows OS).

Example of reading data from the phytosensor via ASCII stream from COM port on Linux OS can be found at **https://github.com/WatchPlant/OrangeBox/wiki**.

It is built around the Rock Pi S single-board computer running Ubuntu (many thanks to Marko Krizmancic).

B. *Actuating high-power devices from external applications*

All connected high-power devices can be actuated by issuing the corresponding ASCII commands, see Table III.

C. *Real-time AI applications with python*

For advanced numerical analysis, AI applications and real-time actuation, the client program can provide data to a python server via named pipe mechanism on Windows OS, see Fig. 6. The pipe name

```
\\.\\pipe\\EISClientPipeX
```

where X is the number of pipe; it can be used for any asynchronous external data processing algorithms (not only python) that run on the same machine as the client program. If these algorithms are executed on another computer use the socket mechanism.

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=X;
```

in *init.ini* file (X is the number of pipe). Data are provided to the pipe at the same time and in the same structure as they are

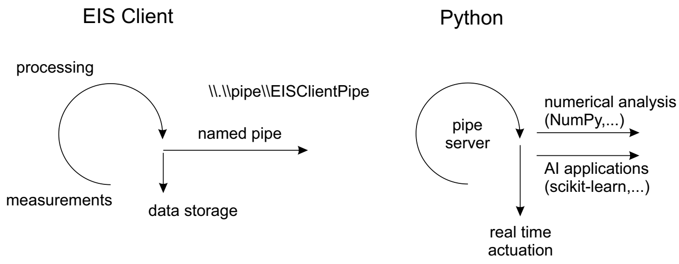


Fig. 6. Connection between Client program and python application via named pipe.

stored in files – line by line with all data columns (see [1]). If the parameter

```
saveAfterNSamples=x;
```

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 client program 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.

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

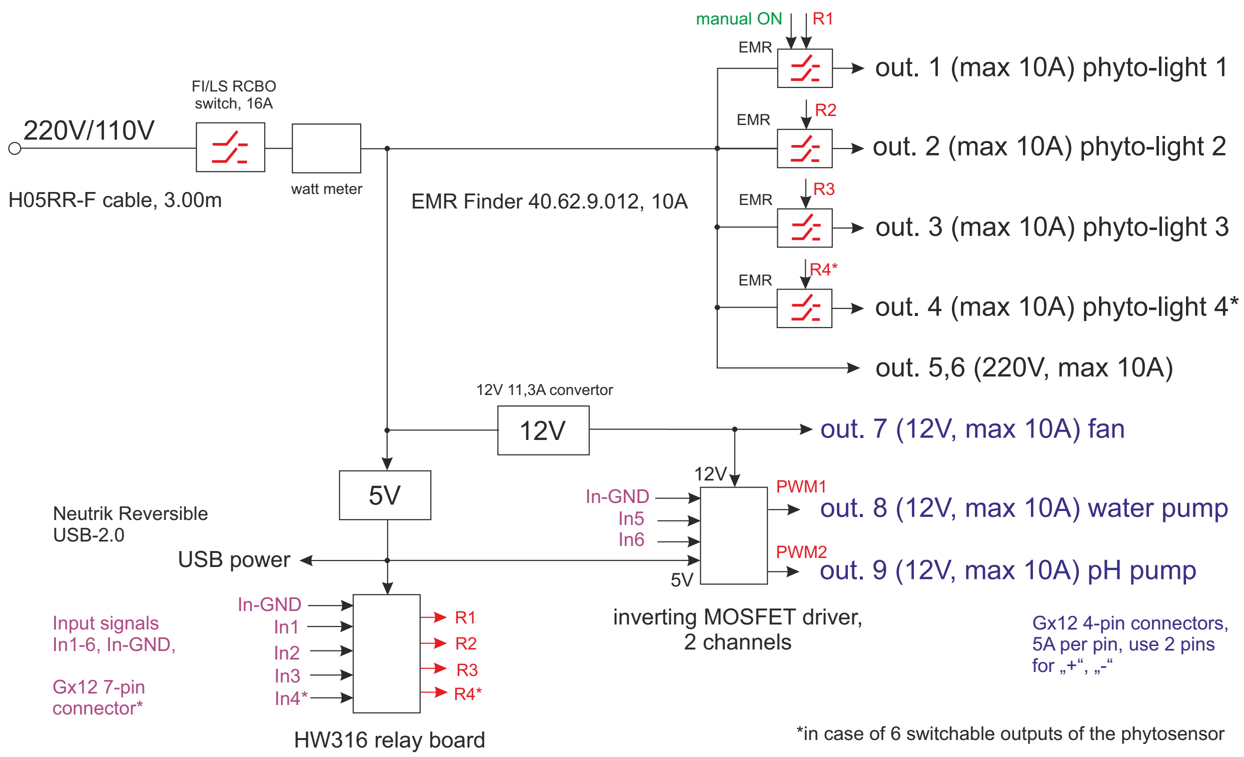
```
usingActuators=1;
```

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

VI. DETAILS OF THE POWER MANAGEMENT MODULE

A. *Electrical connections*

The phytosensor uses internal n-channels MOSFET to control output channels R,G,B, *ts1*, *ts2* and the power switch to control 3.3V. There are two ways to connect external SSR/EMR: with internal 3.3V or 4.2V, and with external power supply, see Figure 8. Typically, most of modern SSR have 3-32V input for control, thus 3.3V can be used; 4.2V can provide only 200mA current and is used primarily for powering sensors. Note that n-channel MOSFETs implement so-called low-side



(a)

(b)

Fig. 7. The power management module with 5 or 6 switchable outputs, different configurations are possible: **(a)** structure; **(b)** pinout.

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.

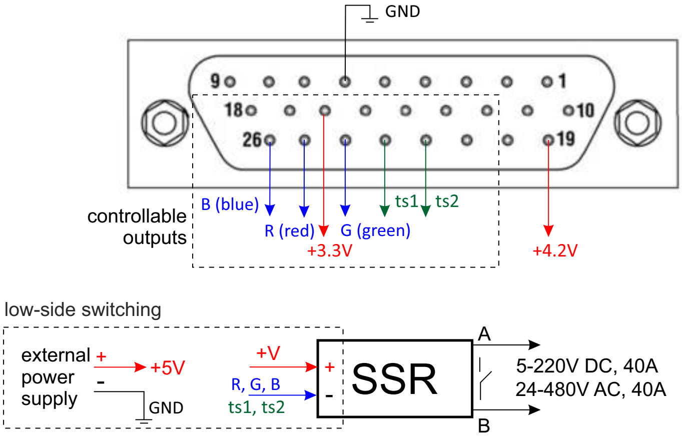


Fig. 8. Connecting SSR with 3.3V, 4.2V or external power supply, *ts1*, *ts2* – output of the thermostats. 3.3V output is switchable and can be used for ON/OFF purposes.

Note that 3.3V output is switchable and can be used for ON/OFF purposes as a normal port (not a low side switcher). 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.

Important: the low side PWM on the phytosensor requires the inverting MOSFET driver as shown in Fig. 9.

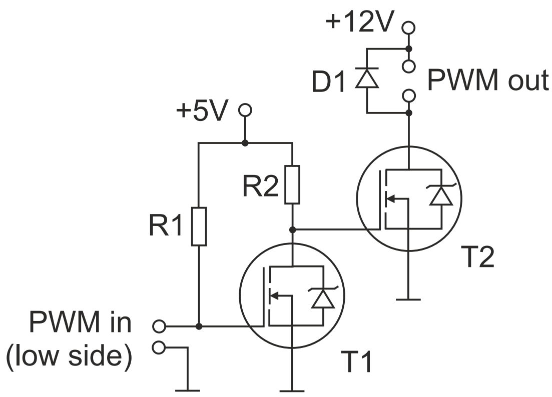


Fig. 9. Scheme of the inverting MOSFET driver, example values for 16-22kHz PWM with up to 10A current: R1, R2 – 5kOhm, T2 IRF520.

Structure of the power management module is shown in Fig. 7. It includes several EMR for switching high-current outputs, MOSFET drivers, 5V and 12V power supplies and protecting RCBO modules. The power management module consists of off-the-shelf components mounted on the DIN rail and can be ordered in CYBRES shop.

B. LEDs and Relays

Since internal MOSFETs are used to control LEDs and external power equipments, their behaviour obeys the following rules:

- **'output channels'** (in tab 'System') – this is the main flag that specifies the role of output channels: either for LEDs or for external equipments. If 'use relays' is selected, the MU system does not use these channels for indication purposes (only timer and manual on/off operations). If 'using LEDs' is on, indicators, timers and manual operations are possible. Setting or unsetting 'output channels' does not change any channels;
- If 'LEDs' or 'thermostats' are selected in 'output devices', this automatically enables 3.3V output, but this is not stored in EEPROM;
- The green front LED is can be changed only in two cases: 1) manual on/off by users (on/off of the 3.3V output), if the flag 'use LEDs' and 'use thermostats' are off; 2) if the flag 'use LEDs' and 'use thermostats' are on, this automatically turns on the green front LED and the 3.3V output is powered.
- The red front LED is reserved for indication purposes (note it also turns on/off the secondary 3.3V power source, which is not available on the 26 pin connector) and can be also turn on/off manually.

The status register describes usage of all output channels (accessible via **wp*** command), its bitwise format: (PWM-B PWM-G PWM-R ts1 ts2 B G R).

C. Shared usage

The setting 'EIS Spectrometer' or 'Phytosensor' in 'configuration' (the tab 'system') as well as additional sensors defines usage of output channels and their powering. For instance, if the thermal sap flow sensor is used, output channels 3v, ts2 are not available as switchable output channels. Please configure your input/output sensors/channels according to the used devices.

VII. Further Reading

1) It is recommended to read carefully the user manual [1]. Short overview can be found in a technical presentation [5].

2) Application notes for fluid analysis and water treatment:

- Application Note 20. Increasing accuracy of repeated EIS measurements [6]
- Application Note 24. Analysis of electrochemical noise for detection of non-chemical treatment of fluids [7]
- Application Note 27. Using regression scan for 'treatment-during-measurement' EIS experiments [8]

3) Publications on electrochemical impedance spectroscopy [9], [10], [11] and plant sensors [12], [13], [14], [15].

4) YouTube demonstration and tutorial videos: https://www.youtube.com/@BiohybridSystems

VIII. Disclaimer

This application note describes agricultural applications of the phytosensor, however the installation and further operation of the phytosensing system may only be carried out by certified and trained personnel in compliance with local legal regulations. When using this approach, recommendations and corresponding implementations in CYBRES MU devices –

considered as a service from CYBRES GmbH – its experimental character in relation to methodology, equipment, accuracy or reliability should be always considered. CYBRES GmbH does not assume any liability arising out of the application or use of any part of this service, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages of any kind. CYBRES GmbH can any time change this service, add/remove diverse functionality. CYBRES GmbH follows the data protection rules, any requests/data exchange remain confidential as long as other agreements with corresponding partners are not achieved. Graphical images are allowed to copy when the word 'CYBRES' remains on caption to these images. Any citations or references on graphical/technical material should include links to CYBRES GmbH.

References

[1] CYBRES, Differential Impedance Spectrometer for electrochemical and electrophysiological analysis of fluids and organic tissues. Handbook and User Manual, CYBRES GmbH, 2024.
URL https://cybertronica.co/download/MU-EIS_Manual_en.pdf

[2] Y. Liu, D. Li, J. Qian, B. Di, G. Zhang, Z. Ren, Electrical impedance spectroscopy (eis) in plant roots research: a review, Plant Methods 17 (11 2021). doi:10.1186/s13007-021-00817-3.

[3] K. Jin, J. Shen, R. Ashton, R. White, I. Dodd, A. Phillips, M. Parry, R. Whalley, The effect of impedance to root growth on plant architecture in wheat, Plant and Soil 392 (07 2015). doi:10.1007/s11104-015-2462-0.

[4] M. Van Haeverbeke, B. De Baets, M. Stock, Plant impedance spectroscopy: a review of modeling approaches and applications, Frontiers in Plant Science 14 (2023). doi:10.3389/fpls.2023.1187573.

[5] 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

[6] S. Kernbach, Application Note 20. Increasing accuracy of repeated EIS measurements, CYBRES GmbH, 2017.

[7] S. Kernbach, Application Note 24. Analysis of electrochemical fluctuations for fast impedance spectroscopy, CYBRES GmbH, 2018. doi:10.13140/RG.2.2.10504.23047.
URL http://cybertronica.de.com/download/CYBRES_Application_Note_24.pdf

[8] S. Kernbach, Application Note 27. Using regression scan for electrochemical 'treatment-during-measurement' experiments, CYBRES GmbH, 2020. doi:10.13140/RG.2.2.24161.93280.
URL http://cybertronica.de.com/download/CYBRES_Application_Note_27.pdf

[9] 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.

[10] S. Kernbach, O. Kernbach, Environment-dependent fluctuations of potentiometric ph dynamics in geomagnetic field, Electromagnetic Biology and Medicine 41 (4) (2022) 409–418, pMID: 36200513. doi:10.1080/15368378.2022.2125527.

[11] 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.

[12] S. Kernbach, Device for measuring the plant physiology and electrophysiology, IJUS 4 (138) (2016) 12–13. doi:10.17613/4xyq-8z28.

[13] E. Buss, T. Aust, M. Wahby, T.-L. Rabbel, S. Kernbach, H. Hamann, Stimuli classification with electrical potential and impedance of living plants: Comparing discriminant analysis and deep learning methods, Bioinspiration & biomimetics 18 (02 2023). doi:10.1088/1748-3190/acbad2.

[14] 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.

[15] 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.