
Poseidon PHP Logger manual

PHP application for data recording and displaying graphs from several Poseidon units.

Application for collecting and visualization of the data measured by Poseidon units from HW group production.

Poseidon units are designed for measurement of temperature, humidity and other values and data are accessible through XML file. Poseidon PHP Logger application is designed for periodical reading (script called by cron) of these data from one or more units, their storing to MySQL database, followed by displaying the measured data:

- actual data
- day's graph
- week's graph
- month's graph

Different Poseidon units can connect various number of sensors (1-42), when number of units can vary, while the unit is defined by an IP address and the port to which listens its web server. Practically there is no use for monitoring more than 20 sensors, considering WWW color resolution (256 colors) used in graphs.

If a unit is not read, the error e-mail is sent to the defined address.

When the measured value is out of the allowed range (both values are saved in unit's configuration), the e-mail is sent to the defined address.

SW requirements (it is available)

- Apache 2.0 and higher
- PHP 5.0 and higher
- MySQL 5.0 and higher

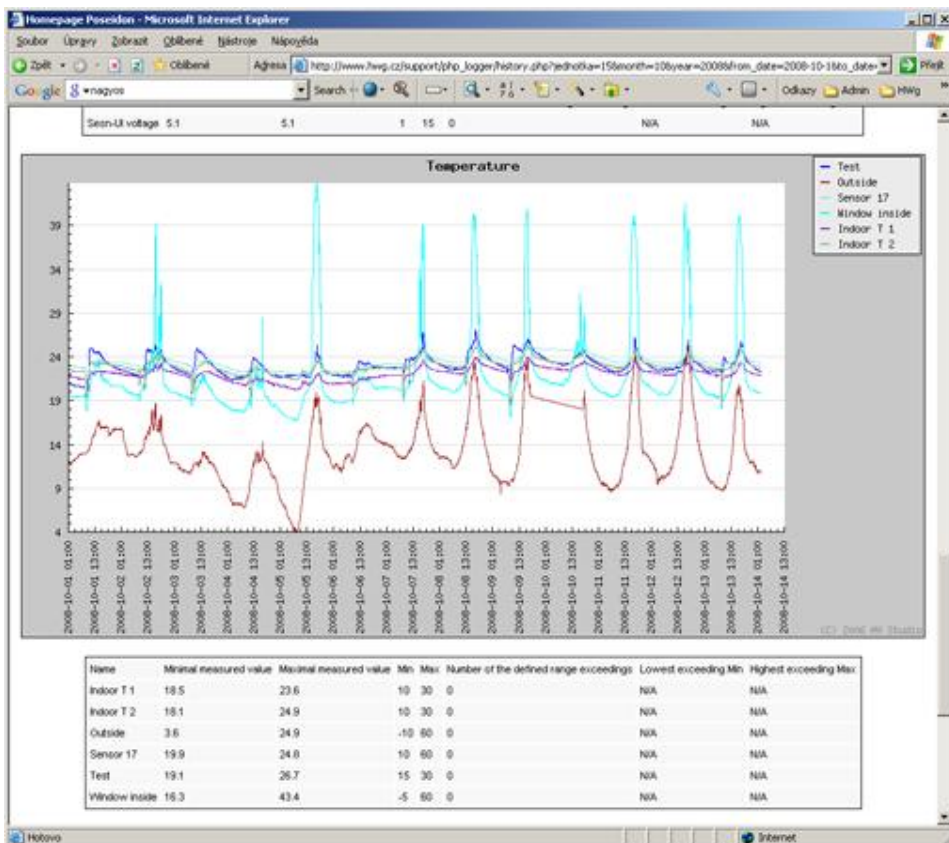
Note: Backward compatibility might be available according to the specific configuration.

Basic application configuration

Application has the configuration file (config.php) with the definition of the connection to DB, table names definition, in order not to get in conflict with current names of tables. In the similar way is defined the list of IP addresses and ports of monitored devices and the email address for sending error messages. The maximum number of monitored devices depends on the network speed, practically 10 devices are maximum.

Loaded and stored information – Poseidon unit

- IP address
- DeviceName (from XML)
- VendorID (from XML)
- Title (from XML)



Loaded and stored information – Sensors (all from XML)

- Name
- Units
- Value
- Min
- Max

Loaded and stored information – Binary inputs (all from XML)

- Name
- Value

If needed, other information, e.g. *time*, are saved as well, however these entirely depend on the server and scripts.

If the script cannot make a connection with the device, the information e-mail is sent but nothing is saved to the database.

Wrong read data cause an information e-mail, but they are not stored to the database.

Graphical interface of „Visitors“

- Generated pages :
- Homepage
- History of xxx unit

Homepage

Contains the list of monitored devices (Poseidon units) sorted by the *DeviceName*. Each *DeviceName* is linked with named anchor at the same page. This anchor contains a table containing the list of measured sensors sorted by the *Name*, and possibly grouped by *Units*. For each sensor there is:

actual value – *Values* + *Units*,

- Min
- Max

If the *Value* is out of the range defined by *Min* and *Max*, the value is written highlighted (under-colored, or by red color).

The table contains the list of inputs + actual status.

History of xxx unit

The page contains graphs similar to MRTG.

Graph in different colors (each sensor has a different color) displays measured values.

Bellow each graph there is a table containing information about each sensor for the defined interval:

- *Name*
- Minimal measured value
- Maximal measured temperature
- *Min*
- *Max*
- Number of times the values exceeded the defined range
- Highest exceeding under *Min*
- Highest exceeding over *Max*

+ number of binary inputs changes in the specific period

The page ends with a table containing the last 10 binary input changes:

- Name
- Value
- Time

Interface modification

- **File config.php**
Use this file to set necessary data for connection to a database, sending e-mails and a list of monitored IP addresses.
- **File graf.php**
On the line 103 you can find the definition of the used colors (up to 43) that are applied to the graph. The list of available colors is included in documentation.
- File **head.php** and **foot.php** (used for user-defined header and footer).
- **File cron.php**
Used for periodic data reading. Must be filled in cron tab with the required interval.
- **File db_optimize.php**
Used to clear the Database after disconnection of unused units and sensors.