EMS data acquisition principles

Jiří Kučera
EMS Brno, CZ
Brno, July 2018

General features

The main idea standing by the beginning of development of data acquisition systems was

- unity of hardware and software
- versatility, flexibility and complexity of systems
- software support of data handling including pre-processing

Dataloggers – most important topics

All our dataloggers are designed with respect to

- lowest possible power consumption
- maximum accuracy, mainly by the measurement of microvolt signals, maximum resistivity against noise.
- quality prior to manufacturing costs
- user friendly handling
- memory issues

Details follows.

Power consumption

All dataloggers we manufacture are based on 16-bit microprocessor ATX Mega (Atmel/Microchip) since more powerful processors or modules have incomparably higher power demands without any significant advantage.

It limits the maximal rate of measurement and needs more time of programming but even quite large multichannel dataloggers can run from primary cells or with only small solar panels.

Typical idle current of data logging sensors or kits is around 10μA, large loggers around 100μA.

Typical operating current during the measurement is mostly up to 30 mA.

Accuracy

For reasonable accuracy dataloggers have usually eight voltage ranges, mostly 20 mV to 2500 mV. The resolution on the lowest voltage is better that 1 μV, real accuracy ± 2 μV incl. reading uncertainty.

Before the measurement of each input the A/D convertor is reset in terms of offset and full scale range. All inputs are arranged as differential.

For the measurement of RTD sensors dataloggers we use only four-wire connection. The excitation voltage is also measured before the measurement of each sensors.

The A/D convertors (Analog Devices) use the sigma/delta principle, the measurement of each input takes approx. 200 msec.

Counting inputs have range 65,000 pulses with special setting for low frequency pulses (typically rain gauge).

Quality prior to costs

- stainless steel
- highest quality plastic materials
- more expensive but uncompromising design in terms of stability and reliability

This is our general approach to device development.

User friendly handling

Fast data download. It is not minor issue. One will realize this when downloading data in rain or cold weather. It is allowed by extremely compressed data format. Measured values are stored in memory on two bytes only. This corresponds with 16-bit resolution. The time stamp is entered only at midnight (or by any interruption of regular operation like change of datalogger configuration, reading of actual values, data download etc.) The conversion to physical values is performed according to configuration which is also stored in datalogger memory.

Consequently, the number of values in the datalogger memory is nearly half of memory size in bytes.

User friendly handling (continuation)

Note 1: Prevailing way of storing data on memory card or memory stick has some disadvantages, mainly by small logging kits or logging sensors placed in harsh environment, when it is necessary to open the enclosure with the risk of getting water or water vapor inside the system.

In case of not sufficient internal memory capacity when it is impossible or complicated to download data on plot there is a possibility to store data on SD card (RailBox, GreyBox lines). Here, the data are stored as one-day data files. Mini32 software in “Bulk conversion” option quickly and easy converts and chains these raw files to one DCV file for next processing.

User friendly handling (continuation)

Note 2: Nowadays, on-line data transfer to Internet is more and more often demanded. Therefore, the size of memory is not a big issue and the internal memory serves just as a buffer for case of transmission problems. Our GPRS or LAN modems are available also with SD card for data storing. Such a system is therefore well secured for case of failure of data transmission to Internet.

Note 3: All our small system with internal battery are watertight and even ready for immersion to water or long term placement in mud thanks to infrared data access. Bigger dataloggers (RailBox, EdgeBox) are intended for installation to an enclosure. They are not protected again environment and they are equipped with RS232 connector (Stereo Jack 2.5 mm). On the contrary, Greybox SDI-12 dataloggers are designed as sturdy watertight aluminum boxes. Therefore, they don’t need any additional protective enclosures.

Memory issues

EMS dataloggers contain internal non-volatile flash memory. Small loggers (Minikin and MicroLog lines) have mostly 128k or 256k memory, enough for at least 50,000 or 100,00 measured values, respectively. Larger loggers have 512k memory.

Measured data are stored as two bytes number and each midnight completed with a header containing data structure and time stamp. Hence, the maximum number of records in memory depends on the frequency of measurement and number of measured inputs (channels). Small numbers of measured values a day cause that relative large part of memory is occupied with midnight headers. Besides of regular data, memory also contains the whole configuration of the measurement including way of calculation of physical values, channel description etc.

The memory size in terms of days of recording until the memory is full or when the oldest data would be overwritten is calculated automatically in “Configuration” option of Mini32.

Off line is this information available also in “More>Capacity info” option.

Memory issues (continuation)

The data memory can be set to two modes

- overwriting; oldest data are continuously overwritten with newest data

- conservative mode; datalogger stops operation when the memory is full

Both modes have dis- and advantages - it depends on the user which part of measuring period is more important for him in case of late download.

Note 1: Complicated organization of data in memory with raw two-byte numbers and headers are relatively sensitive to potential errors. It was problem of old systems when it happened that the data structure was damaged and data were unreadable. It happened mostly by firmware complexity, static discharge or by some undetected reasons. Now it occurs only very seldom. Anyway, for such a case there is the possibility to save the whole memory content to file. This file can be sent to manufacturer for fixing.

Memory issues (continuation)

Note 2: The file name containing whole memory content by old data logging systems is created in form mmdd.hcm (month, day).

Newer dataloggers have this name the same as normal downloaded file but with "_HCM" on the end of file name (for instance XY_2017_05_11_HCM.hex). Those files can be mostly directly converted to DCV file.

Anyway, in case of problem the file can be sent to manufacturer for free repair.

Data structure and handling

The other issues related to data structure in the datalogger memory are

- datalogger configuration
- data handling
- on line data sending to third party system

Datalogger configuration

The configuration is fully supported by Mini32 software. The user can

- chose sensors (relevant to connected datalogger) from the Mini32 library or use its own calculation from electrical values to physical values (explained later).

- chose time interval on the measurement and interval of averaging and storing to memory. There is also the possibility to measure in averaging intervals only. It saves energy and it suits for variables with low dynamics like soil temperature, soil moisture etc.

Note: All time intervals used for setting dataloggers or calculation in Mini32 are always integer fragments of minute, hour or day. This arrangement assures synchronization with those units. For instance: Minute intervals allowed are only 1, 2, 3, 4, 5, 6, 10, 15, 20, 30. Therefore, the measurement must be performed always in the entire hour and similarly in the entire minute and day. This arrangement significantly simplified the data processing.

Datalogger configuration (continuation)

Another important issue related to data structure and data handling is the strict rule that any channel of datalogger must not contain incompatible values (output from sensors measuring different variables).

Therefore, each change of datalogger configuration in Mini32 is checked for this compatibility. If the software finds an incompatibility it warns user against continuing until the device code is changed and the datalogger is initialized (erasing memory, update of system time). That means that the logger starts to create new data set of different structure with another device code.

The reason for this is that all data saved in the memory are processed according the current configuration. That means, for instance, that when a temperature sensor has been replaced with a humidity sensor, previously recorded values from the temperature sensor would be misinterpreted as humidity data.

Datalogger configuration (continuation)

This is the main catch of operation of EMS dataloggers. The solution would be to store each configurations to memory but it would seriously influence the whole data logging concept. Also in small dataloggers would more configuration sections occupy a significant part of memory.

On the other hand, this feature prevents data mismatch in one file.

Nevertheless, some users could find this issue uncomfortable and we are going to solve it in close future.

Data handling

With respect to specific way of data storing into the datalogger memory there are two main type of files:

- HEX file. This file is created during the data download. It has small size since the measured values are still stored on two bytes. This file nevertheless contains all information concerning the measurement; not only numbers but also the way of conversion to physical values, time intervals of measurement etc. Simply the whole configuration. Therefore, this file can be used as the base for creating of similar configuration of the same datalogger. Since the HEX file can be easy at any time converted to DCV format, it makes is suitable for archiving.
- DCV file. It is created immediately after the downloading of the HEX file. Its size is ca four times larger than HEX file since it already contains decompressed physical values. This file is still “hardware dependent” which means that it contains all hardware relevant information as HEX file.

Data handling (continuation)

Note 1: HEX file can be reopen for different reasons:

- when the DCV file was damaged and it has to be refreshed
- when there is a wrong channel setting. The channel configuration (wrong sensor for instance) can be fixed in order to create correct DCV file.
- STX (or STG created by older Mini32 versins) file containing the datalogger configuration can be exported from HEX file (Save setup) and possibly edited for any next use (“More>Off-line edit”). STX file can be exported from DCV file, either.

Note 2: “Hardware dependent” DCV file mainly keeps measured variables in fixed order exactly according to input channels. The main advantage is that it is impossible to join together files coming from different dataloggers. This feature get lost after “User defined calculation”, “File mixing”, “Export” from graphic screen etc.

Data handling (continuation)

Note 3: DCV file stores not only values but also the whole history its life (calculation scripts, parent files, notes of data erasing, editing etc.). The file also carry "User defined views" of graphics and last view of graphic screen before the file was closed. All these information are shown after pushing of "History" button or in separate window of file manager when opening new file.

Note 4: A specific way of working of Mini32 is creating next files after each significant operation like file mixing and chaining, programming calculation etc. See "Mini32_outline.pdf" file for fast overview. Some of those file have different structure with respect to original file/files and therefore they are already not "hardware dependent".

Note 5: The Mini32 software is result of continuous development since 1998. During that time many useful functions have been included for fast and smart processing of even very long time series.

Data handling (continuation)

Note 6: Electric values.

These are principally measured values before calculation to physical values. Mostly voltage, number of pulses but also a special values measured directly inside the datalogger (datalogger temperature, internal reference, ratio, etc.). By some single-purpose dataloggers those electric value are the same as physical values, as well as by SDI-12 sensors.

The file containing only electric values can be obtained after checking “El. val.” box before pushing “Next”. The filename will end with “#” for easy distinguishing from normal files.

This file helps to find the problem in configuration and/or for better understanding of sensor behavior. Note that the channel description in this file remains untouched.

On line data use by third party

This is another issue related to the specific data structure of EMS systems. On-line data sent by dataloggers on request are unreadable. This is a problem when the data are sent to third party system via LAN or GPRS data transfer. For this case we provide a server software. This software is composed of one file and it has to be installed on user's computer (with Windows® OS). The software converts received data to text format and writes them to daily files. Also, it manages gaps caused by temporary transmission problems (up to one months) by talk-back.

Authorized users who use our cloud system of data visualization and archiving can download data in text, XLS or DCV format at any time.

Common datalogger setting

EMS dataloggers measure input values in regular intervals and store them to memory also in regular intervals as averages. Interval of averaging/storing is usually longer than measuring interval because variables with high dynamics has to be measured more often then it is demanded step between following records. Both intervals are always integer fragments of second, minute, hour or day and averaging/measuring interval ratio must be integer in order to come through the whole higher time unit.

There are principally two kinds of sensor – self powered and sensors which needs a power supply. Only few such a sensors must be powered continuously (typically sap flow sensors); most of sensors can be powered during the measurement only. Since some of those sensors need a time until they are ready, the datalogger must power them in advance. Therefore there is a warm-up time interval that can be set to 1, 2, 3, 4 or 5 seconds.

Common datalogger setting

Note: Be advised that sensors which are measured in intervals of storing only can be also powered only in these intervals. For this purpose, each datalogger of EdgeBox and RailBox line has separate terminals marked as “Pm” (powering each measurement) or “Ps” (in storing intervals). The warm-up time is set independently for both powering modes (RailBox only).

Common datalogger setting (continuation)

Device code – it's a name of datalogger which is shown by each communication and which makes first two characters of data filename when this name is created automatically. Anyway, even if the user change the name of file, the device code is always displayed in opened file. From historical reasons the device code has only two characters (special characters are not allowed!). Unfortunately, there is impossible to extend it because of back compatibility with older files.

Note 1: By such operations with files when new file with the structure independent on hardware is created, the device code of original files are add to channel description in order to keep reference to original file.

Note 2: By default, the device code is set the same as last two characters of datalogger serial number (factory setting).

Note 3: Users of more loggers on the same locality are advised to think twice of meaningful assignment of device codes.

Common datalogger setting (continuation)

Channel types.

There are mainly two types of analogue measuring inputs – voltage and pulse.

Most of sensors have voltage or current output, both can be easy measured by voltage channels.

Pulses are mostly generated by tipping-bucket rain gauges and by mechanical anemometers. Here is counted the number of pulses coming within measuring period. Because of averaging, pulses coming very seldom could be lost due to rounding. Therefore, the pulse inputs can be set into two modes – fast and slow. Fast pulses are counted as mentioned above, low frequency pulses are counted during the whole interval of averaging. In both modes the maximum capacity of counters is 65,536 pulses which corresponds with two bytes number.

Common datalogger setting (continuation)

Channel setting.

The easiest way is to choose the sensor from Mini32 library (gauges). If the library does not contain the sensor which should be connected, the library offers some general equations with editable parameters:

- linear
- 2-nd order polynomial
- inversely proportional

and some other special function (resistance threshold etc.)

Note: Besides of gauge selection it is necessary to set also voltage range (pulse mode “slow” or “fast” by counters) and recording mode (in measuring or averaging interval). Important information are shown in grey area of channel setting window.

Common datalogger setting (continuation)

Note 1: It is necessary to distinguish between the measured voltage and the maximum value allowed at each input terminal.

The input voltage is measured as voltage difference between both input terminals (Low, Hi). The limit is ± 2.5 V.

Nevertheless, the maximum voltage on any input terminal **must not exceed 5 V** against negative pole of power supply. Overrun of this threshold on any input terminal will cause interference between inputs and consequently makes false measurement. Take this into account when the measuring sensors are powered from the same power source as the datalogger.

Note 2: Sometimes happens that due to damaged cable between the powered sensor and datalogger input the power supply voltage appears on an input terminal. New dataloggers EdgeBox and RailBox line are protected against a serious datalogger damage. The datalogger in such a case emits warning continuous beep.

Common datalogger setting (continuation)

The editing od the datalogger configuration is made in Mini32 environment. When the configuration is finished, it must be sent to datalogger (“Send” button). It can be also saved to file (“Save setup”) for a next use.

Do not forget to initialize the datalogger before starting new job or in case of serious change of configuration. The software will recommend this.

Note 1: Do not underestimate the channel description. It helps to later data processing.

Note 2: When the data should be send on-line to EMS Internet application, **each channel description has to be unique -** no two descriptions must be the same! Mini32 is watching for this.

Actual version of Mini32 does already check this.

Common datalogger setting (continuation)

Note 3: There is a rule for names of variable displayed in Mini32 graphs:

- when the channel description is missing, the gauge name is displayed

  Air Temperature [deg.C]

- when the channel description exists, it is displayed only description

  at 5 m

- when the channel description begins with asterisk “*”, both gauge and description names are displayed

  Air Temperature [deg.C] at 5 m (asterisk is omitted by software)

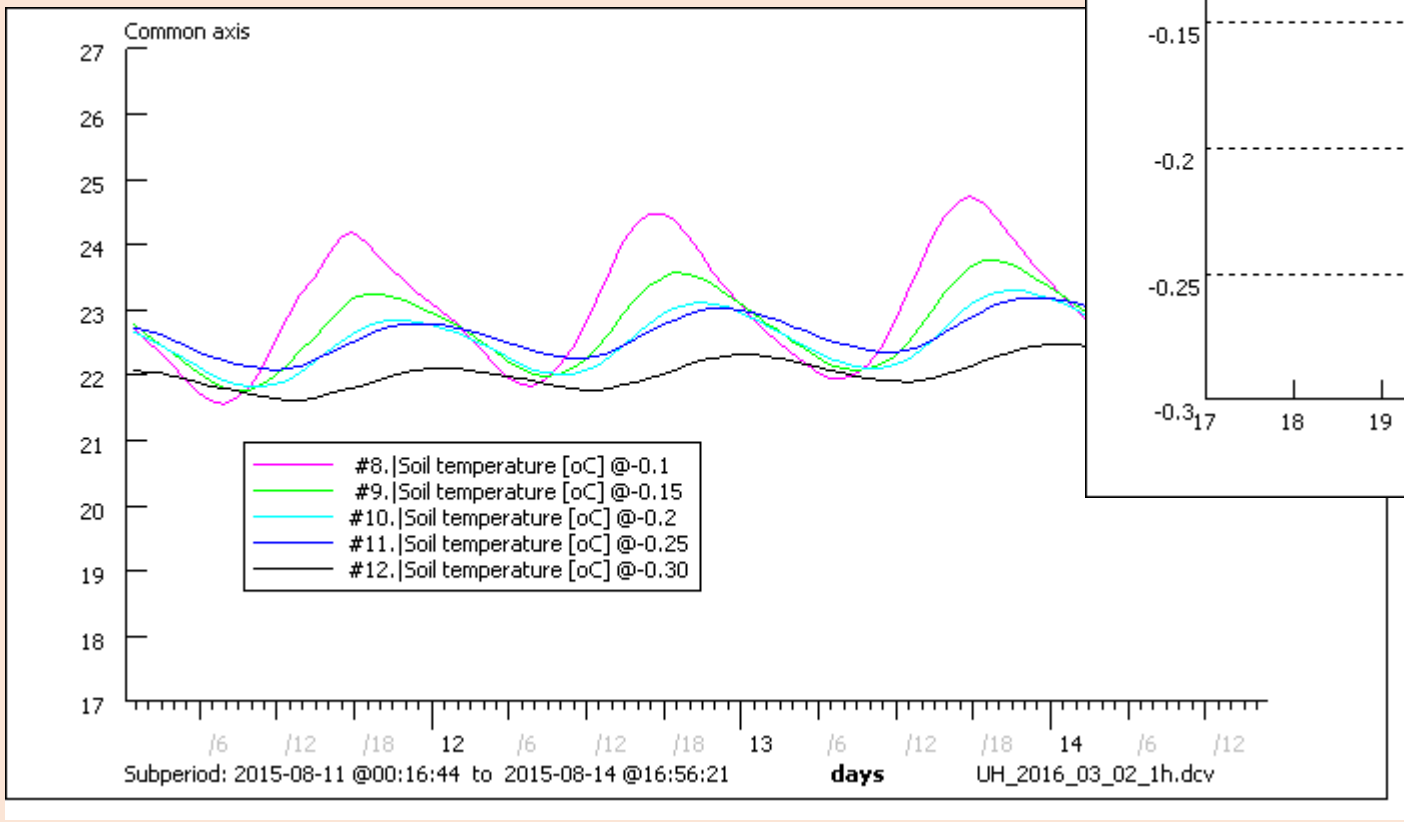
Common datalogger setting (continuation)

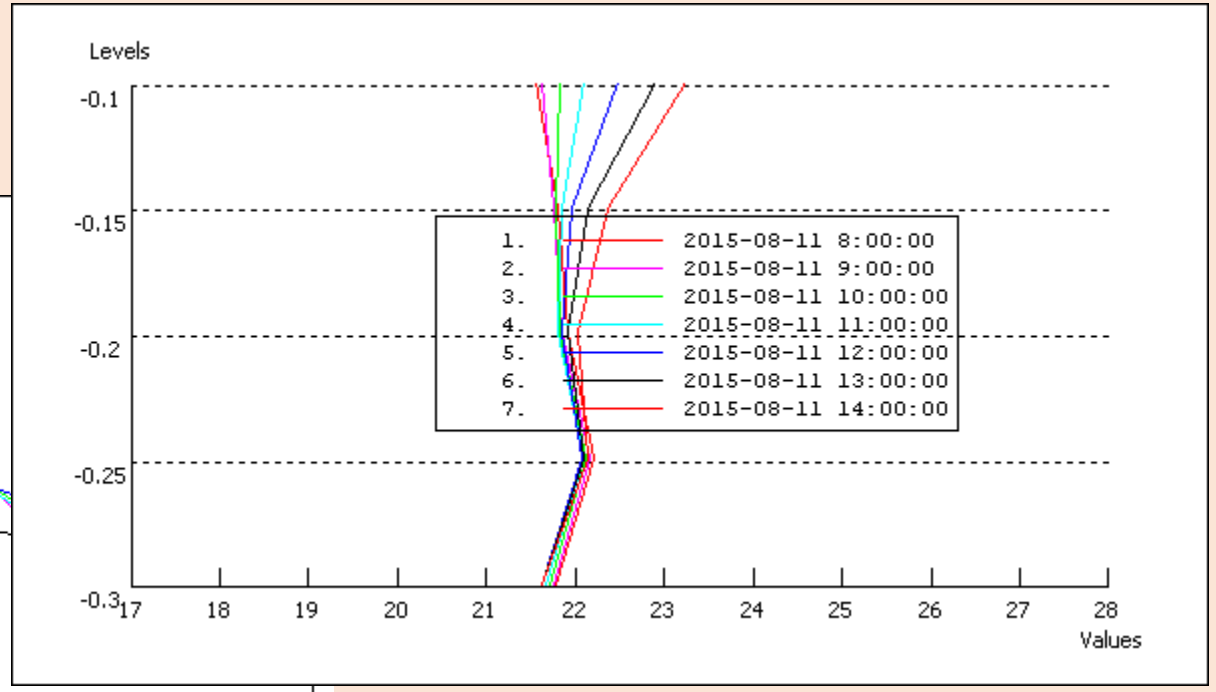
Note 4: When the channel description ends with a number behind “@”, Mini32 understand is as vertical level and use this value for drawing of vertical profiles (pop-up menu in graphic screen):

Soil Temperature [deg.C] @-0.1
Soil Temperature [deg.C] @-0.2
Soil Temperature [deg.C] @-0.3
etc.

See next page.

Vertical profiles - example





SDI-12 network

There are three EMS multi-purpose dataloggers supporting SDI-12:

- MicroLog SDI MP or MP/E
- RailBox RB line
- GreyBox N2N (network-to-network)

The first type is small datalogger with internal battery intended for common low power sensors like, EMS dendrometers, radiation sensors, air temperature and humidity sensors, Decagon MPS-6, Campbell CS 650 etc.

The second type is modular concept with up to 64 voltage inputs and 8 counters and one or three port SDI-12 network.

All system based on this line is ready for completing with GPRS modem, LAN modem, external memory, power module ready for connection of solar panels, interface for sensors with frequency output or sensors which measure resistance with AC signal.

The third type is three- or six-port SDI-12 datalogger containing complete solar power management, GPRS modem and SD memory card in one sturdy aluminum box.

SDI-12 dataloggers – general features

Main goal of operation with SDI-12 sensors is to keep data structure compatible with analog sensor environment. Configuration of SDI-12 sensors is available in separate part of the Mini32 software. Those variables measured by SDI-12 sensors are added behind analog channels and there is no fundamental difference in handling between analog and digital channels.

At some variables coming from SDI-12 sensors is even possible to select gauges as at analog channels.

However, on the other hand, all SDI-12 sensors which should be used this way, must be included in Mini32 sensor library. If an unknown sensor is found during scanning of the SDI-12 network, user is offered to save sensor information to a file. This file is placed to EMSdata folder on the desktop and the user is asked to send it to EMS. Upgraded sensor library will be placed on EMS internet server and all Mini32 users will be encouraged to download the new library file from EMS server.

SDI-12 network features

The configuration of network with SDI-12 sensors is fully supported by Mini32 (Configuration>More>SDI-12 config). Please note the base rule of each SDI-12 network: Each sensor connected to the same port must have unique address in this network! If there are two sensors with the same address, the whole network crashes and its behavior is unpredictable! Note that Mini32 have the option for reading sensor address, but only if the sensor is connected to port #1 (all other sensors must be disconnected). Larger systems have a special terminals for connecting of examined sensor.

The addresses are usually written in numbers and letter (0 - 1, a - z and A - Z). Mini32 shows them in this "official" version but also in the order starting with "0 to 64".

Note: Most of SDI-12 sensors measure more variables. For instance, SF 8X sap flow sensor can produce up to 9 variables. The maximum number of variables in DCV file is 250.

SDI-12 network features (continuation)

EMS dataloggers supports the measurement in the interval of averaging similarly as at analog channels. This can be set in “analog” configuration table the same way as at analog channels.

Because SDI-12 sensors need various way of powering, there is the possibility to power them in measuring/averaging intervals or permanently. The sensor library preset automatically permanent powering at those sensors which need this.

The configuration of SDI-12 sensors can be saved to file similarly as at analog dataloggers. There is also possibility for advanced users to prepare configuration off-line.

Note: Current version of Mini32 allows user to create a file containing common texts of channel description. Use this feature for faster configuration.

SDI-12 network configuration

The configuration of network with SDI-12 sensors is fully supported by Mini32 (Configuration>More>SDI-12 config).

The general procedure is like this:

- set or check addresses of all sensors (one after the other). Connect all sensors to network
- scan the network. Be advised that the order of variables in final DCV file shouldn't follow the sensor addresses; use drop&drag method to change the sensor order if demanded.
- unfold sensors to "modes"
- open "modes"
- mark variables which should be measured
- fill-in channel descriptions. Manually, or use the pop-up menu to fill the preset description from the sensor library write description manually. The sensor address and port number can be added similar way.
- send configuration to "analog configuration screen"
- continue as by common dataloggers. The variable description can be edited here, too.

More on SDI-12 configuration you find in separate manual.

End of the first volume.