POREG 2

Technical Description

CONTENTS

1. INTRODUCTION

The POREG 2 Data Recorder is intended for registration of impulses, for their local processing and transmission to further processing. The operation mode is simply defined by entering corresponding parameters via communication paths, and partly by a direct entry on the data recorder. Input and billing values are transmitted to the data recorder display and/or to the memory card or to higher billing levels. Its design enables you to use the data recorder in different measuring combinations.
The data recorder enables the following functions:

- collection of measurements on consumption of different types of energy (electricity, gas, warm water, etc.)
- collection of measurements in a form of impulse signals
- collection of measurements via communication interfaces
- collection of control signals
- generation of control signals
- local processing of collected data
- producing of measuring and billing results
- real time management
- time functions
- internal tariff program
- operation monitoring in a form of events recording
- long-term data storing
- local data display
- data transmission via communication interfaces

The data recorder is intended for building into measuring sets of contractual customers at medium-voltage network. The measuring equipment of the customers can be precision electricity meters with impulse outputs, induction meters without or with a tariff device as well as static meters without or with a tariff device.

2. DATA RECORDER DESIGN

The POREG 2 Data Recorder is based on a modular construction and in this way enables various combinations:

- CPU with optical communication (IR) and display (LCD)
- power supply module (POWER SUPPLY)
- communication module with double combined RS232C/RS485 interface (COMM.)
- input module (INPUTS)
- output module (OUTPUTS)
- memory card unit (MCU)

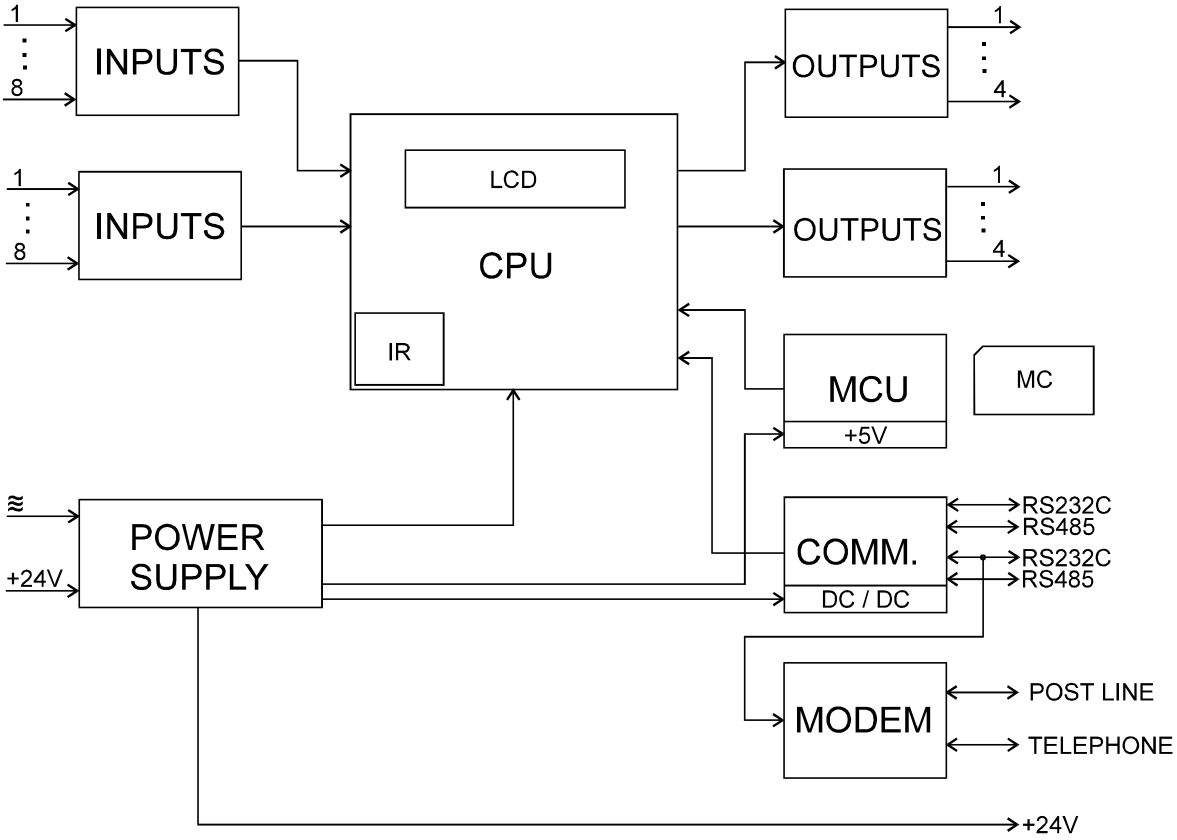


Figure 2-1 Data recorder design

Figure 2-1 shows fully equipped recorder where several possibilities are available:

- basic equipment + input module
- basic equipment + output module
- basic equipment + modem
- basic equipment + input module + output module
- basic equipment + input module + output module + modem
- ......

2.1 CENTRAL PROCESSOR UNIT (CPU)

CPU monitors and communicates with peripheral modules (inputs, outputs, communications, etc.), and performs local data processing.
A processor 68332 is a kernel of the unit. Information on device functioning, settings and different results obtained by means of three keys are displayed on a LCD. The program, used by the processor, is stored in EPROM (data are kept also in case of power failure). Parameters and data are stored in EEPROM and RAM (supported by back-up power supply on the panel). Two asynchronous serial interfaces and an optical interface are used for communication, while an interface for a memory card is used for additional communication. Input and output signals are transmitted via extended input-output signals up to input and output modules.

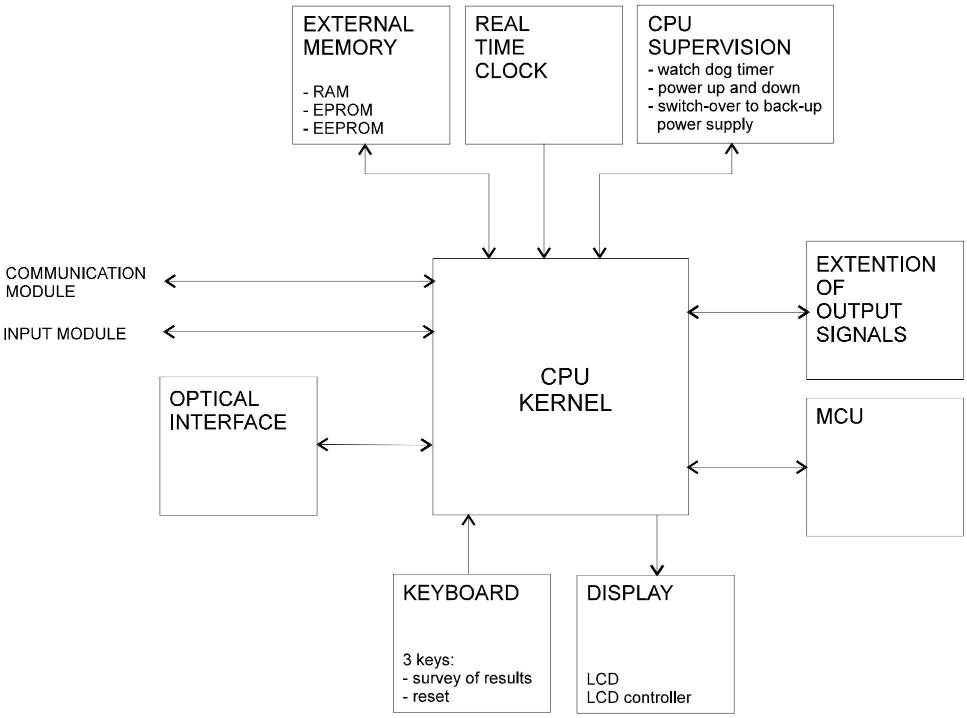


Figure 2-2 CPU

Settings on a CPU:

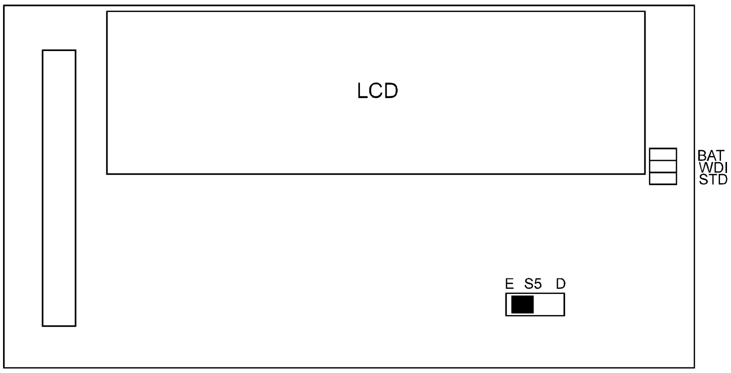


Figure 2-3 CPU - settings

A jumper BAT is open when a device is in a warehouse; it interrupts a supply from a Li-battery.
A jumper WDI is open only for the needs of servicing.
A jumper STD is always closed.
A jumper S5 in "E" position permits parametering of a device with parameters which influence in the results.

2.2 POWER SUPPLY MODULE

It is intended to supply data recorder with corresponding voltages. Three-phase network voltage is used as a power source. A power supply unit assures corresponding output voltages also when two phases fail, and in case of greater deviations of voltage from rated network voltage. Output voltages are electrically isolated from input voltages. In case of network voltage failure the power supply unit operates for a certain time also via external power supply source.

The power supply module ensures voltages for:

- CPU
- input modules
- output modules
- communication module
- memory card unit
- external application (supply of input lines)

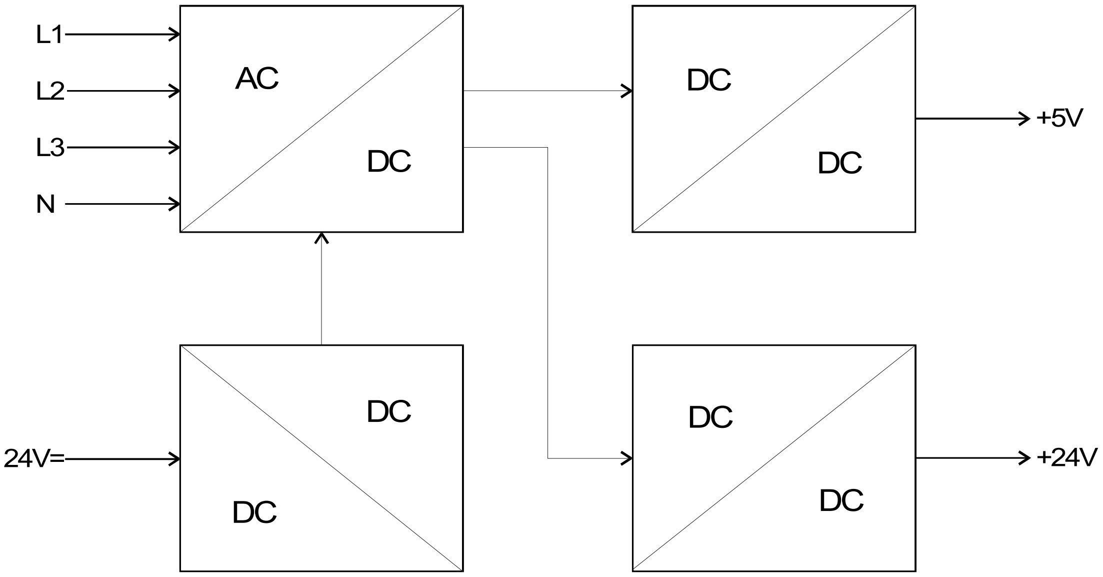


Figure 2-4 Power Supply Module

Characteristics of the power supply module are stated in Chapter TECHNICAL DATA.

2.3 COMMUNICATION MODULE

A communication module is used as an additional equipment and functions as a link between a data recorder and communication equipment or devices. It collects prepared data (meter states with time indications, messages etc.) from the data recorder in regular time intervals or on external request, stores them and transmits them via communication lines.
It consists of two combined RS232C/485 interfaces with electrical isolation between external and internal lines, and a DC/DC converter for supply of external signal lines.

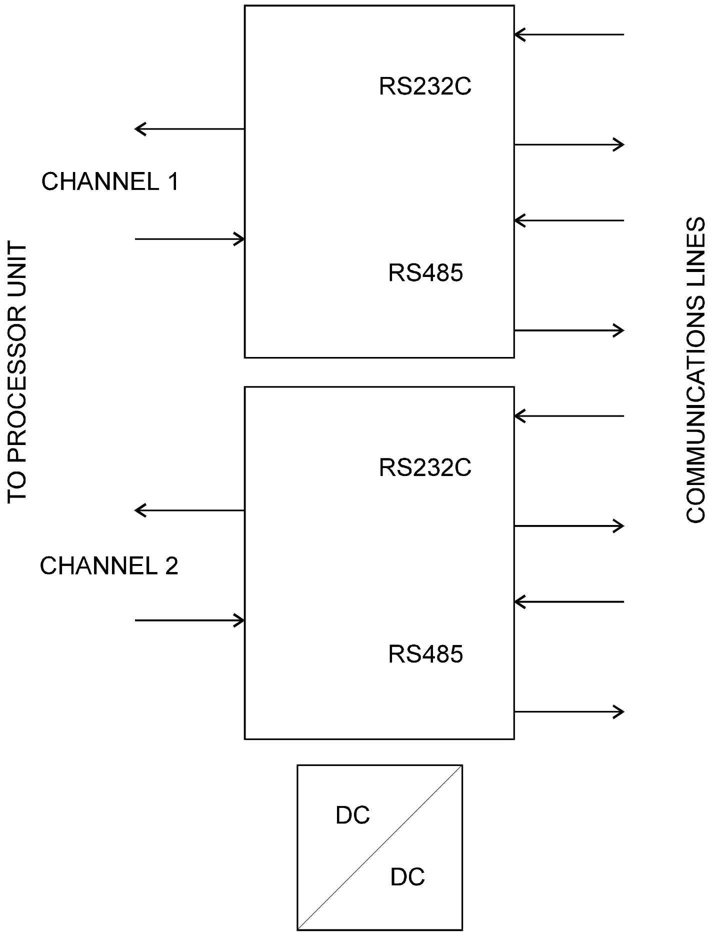


Figure 2-5 Communication Module

Communication via channels 1 and 2 is performed simultaneously. For an individual channel (1 or 2) a simultaneous data transmission to communication lines RS232C and RS485 is possible but a simultaneous data reception from the two communication lines is not allowed.

Physical communication
V.24/V.28 communication for a direct connection with a device.
Transmission rate is defined in usual stages between 150 and 57600 Bit/s
Characteristics of communication interfaces are stated in Chapter TECHNICAL DATA..

2.4 MODEM

A modem is a data recorder additional equipment and supplements communication capabilities of the recorder. It occupies one communication channel RS232C.

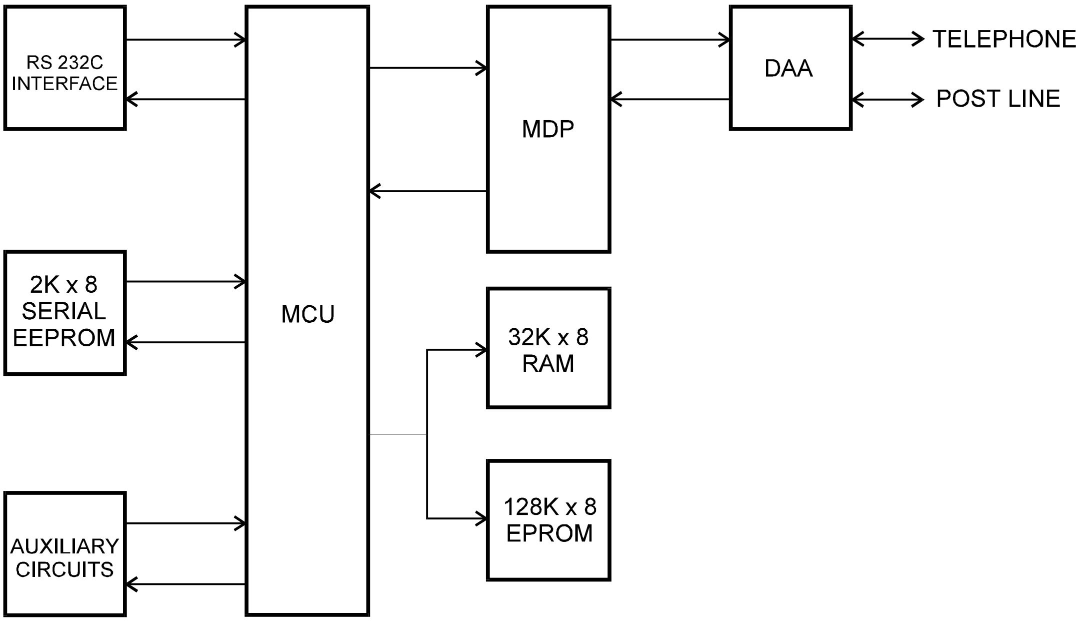


Figure 2-6 Modem

A servicing telephone can be connected. A special circuit prevents from interference between a telephone and a modem. A modem can be set by a program. In EEPROM the user settings and commands are stored, and in EEPROM general program settings on the modem operation.

2.5 INPUT MODULE

The input module involves inputs which are electrically isolated from internal circuits.
Regarding their functions the inputs are divided to testing and impulse inputs. Testing inputs are intended for the reception of testing signals like time synchronization, external tariff control etc. Impulse inputs are intended for the reception of impulse from electricity meters.
Both testing and impulse inputs are divided to following types:

- two-wire input for active current (S1a)
- two-wire input for inactive current (S1b)
- two-wire input for double current (S2) is not realized yet
- three-wire input (S3) is not realized yet

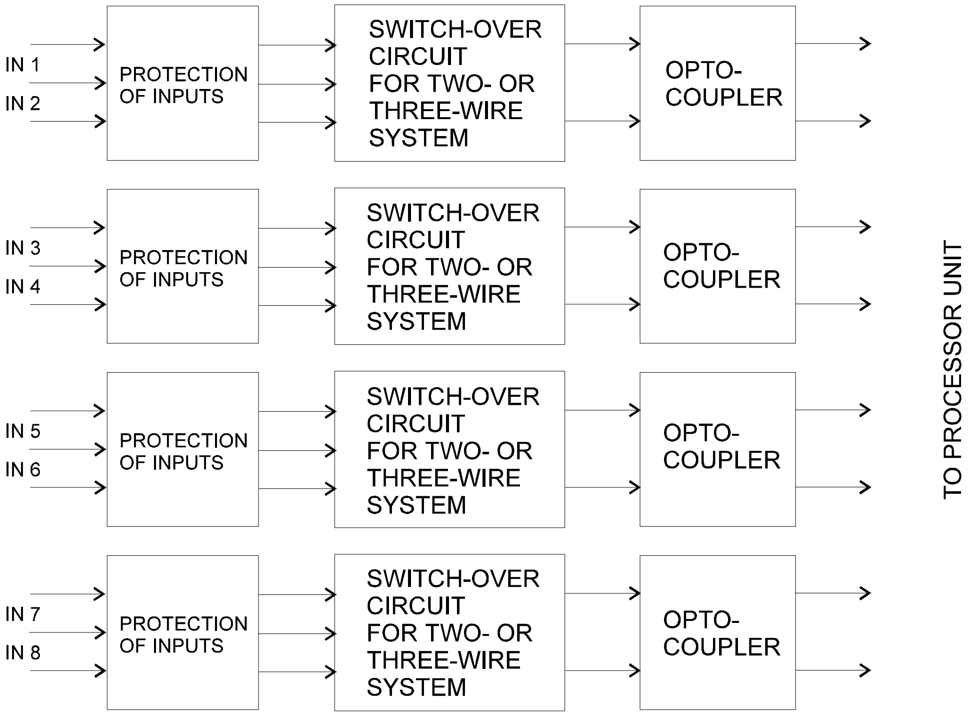


Figure 2-7 Input Module

All inputs are programmable, which means that it is possible to determine a function to each input separately. Inputs are united in pairs where one pair of inputs can represent two two-wire inputs (S1a, S1r or a combination), one two-wire input for double current or one three-wire input. However, new combinations of inputs are available on the level of pairs.
Connection mode of inputs is described in TECHNICAL DATA.

2.5.1 Two-wire input for active current (S1a)

During the active signal (impulse) the circuit on signal lines is closed. Each impulse with defined length represents defined energy quantity. Input is inspected by checking a minimal and maximal impulse length. Both limit values can be set.

2.5.2 Two-wire input for inactive current (S1r)

When a signal (impulse) is active, a circuit on signal lines is open. Each impulse of defined length of duration represents a defined quantity of energy. Input is inspected by checking minimal and maximal impulse length. Both limit valid can be set.

2.5.3 Two-wire input for double current (S2) is not realized yet

In case of a double-current input a signal transition on signal lines from +24V to -24V and vice-versa represents a defined energy quantity. The input is inspected by checking the transition switching time. Changed polarity (transition) should occur in time which is shorter than defined maximal time for transition.

2.5.4 Three-wire input (S3) is not realized yet

In case of a three-conductor input a circuit closing on one signal line and simultaneous opening on another signal line represents a defined energy quantity. The input is inspected by checking simultaneity of switching on both lines. Time between switching on lines should be shorter from a defined maximal time for switching.

2.5.5 Selection between two- and three-wire system

Selection between two- or three-wire system is made in a combination with program settings. Selection is made on a module by jumpers (S1-S8).

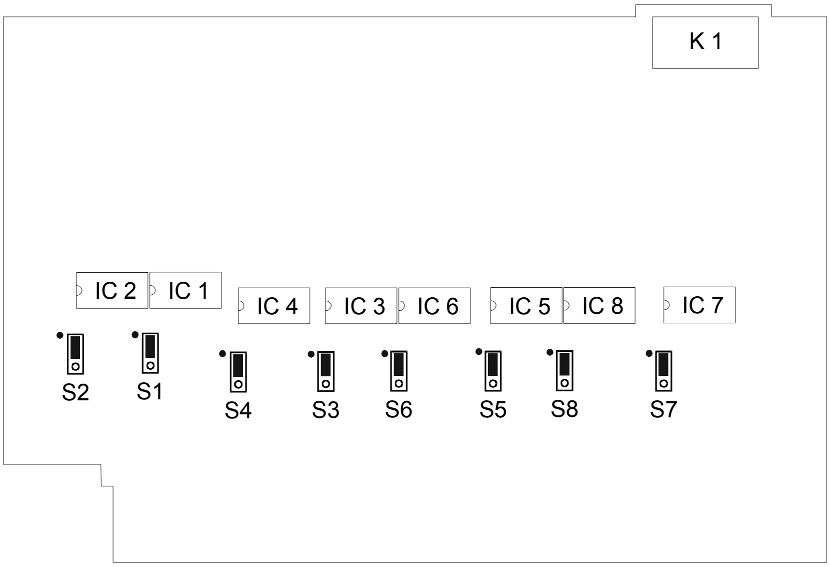


Figure 2-8 Input module - settings

Setting of jumpers on a Figure 2-8 corresponds to the setting of inputs for a two-wire system.
Settings:

two-wire system	S1 - jumper above (beside a dot)	three-wire system	S1 - jumper below
	S2 - jumper above (beside a dot)		S2 - jumper bellow

Inputs are always set in pairs (IN1 - IN2, IN3 - IN4, etc.); one pair represents a corresponding pair of jumpers (S1 - S2, S3 - S4, etc.).

More detailed data are stated in Chapter TECHNICAL DATA.

2.6 OUTPUT MODULE

An output module consists of universal outputs which are electrically isolated from internal circuits. Outputs are performed with solid-state or mercury relays.
A module consists of four outputs. They are functionally divided to control and impulse outputs. Control outputs are intended for a control of external devices (alarms, tariff devices etc.), while impulse outputs are intended for sending the results of data local processing (results of calculated levels) to further processing.
Transmission function can be set for each output separately.

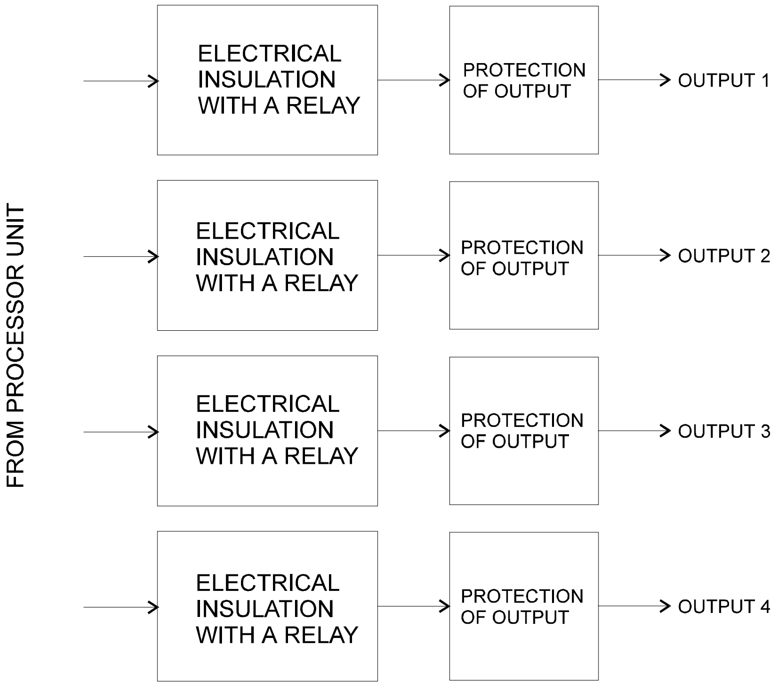

FROM PROCESSOR UNIT
ELECTRICAL INSULATION WITH A RELAY
PROTECTION OF OUTPUT
OUTPUT 1
ELECTRICAL INSULATION WITH A RELAY
PROTECTION OF OUTPUT
OUTPUT 2
ELECTRICAL INSULATION WITH A RELAY
PROTECTION OF OUTPUT
OUTPUT 3
ELECTRICAL INSULATION WITH A RELAY
PROTECTION OF OUTPUT
OUTPUT 4

Figure 2-9 Output module

Characteristics of the outputs and connection mode are stated in Chapter TECHNICAL DATA.

2.7 MEMORY CARD MODULE

A module is intended for connection of a memory card (MC) which is used as an additional way of communication or data transmission. The module supports only those memory cards for which 5V supply is required.

3. DATA RECORDER FUNCTIONS

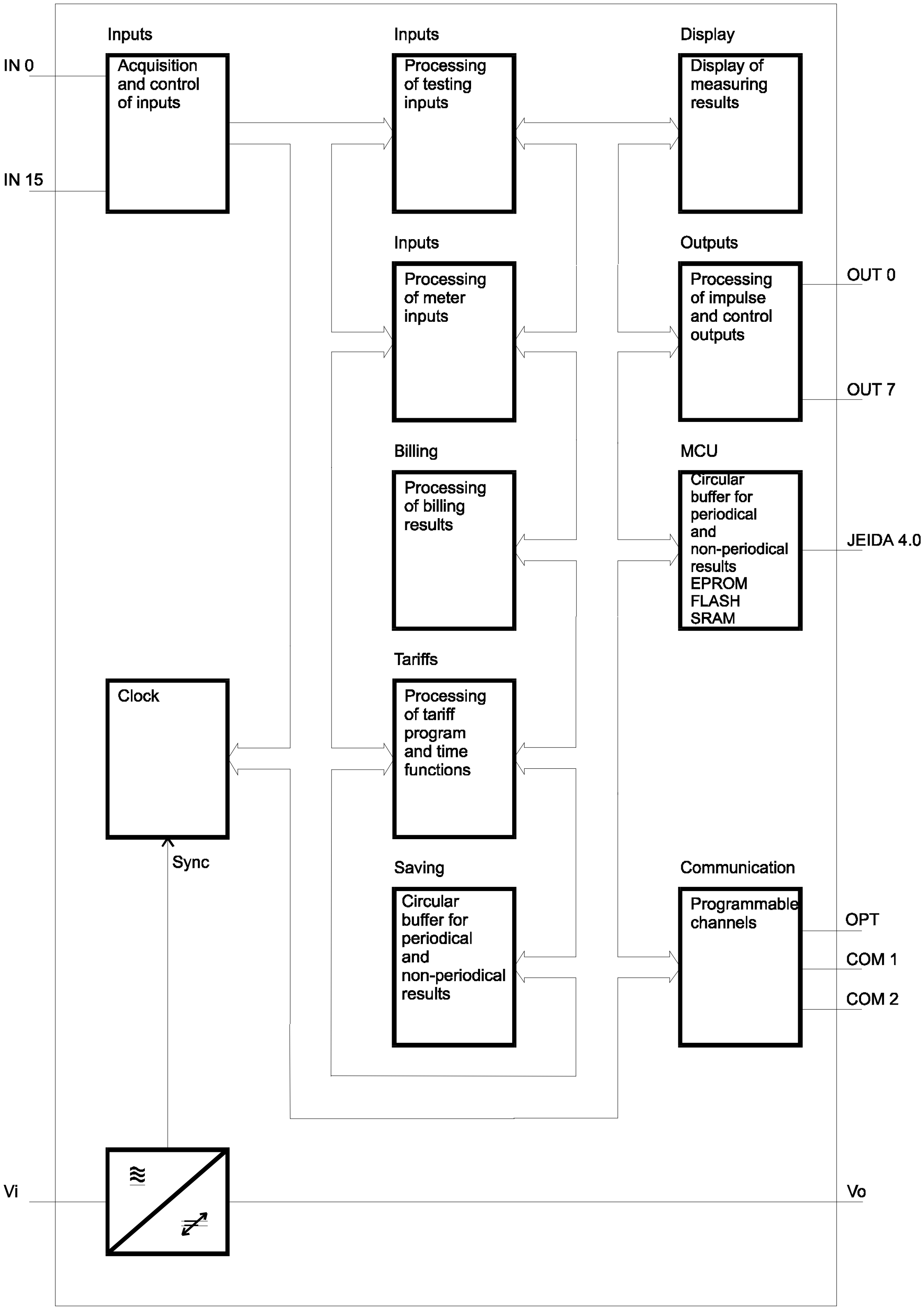

Inputs
Acquisition and control of inputs
IN 0
IN 15
Inputs
Processing of testing inputs
Display
Display of measuring results
Inputs
Processing of meter inputs
Outputs
Processing of impulse and control outputs
OUT 0
OUT 7
Billing
Processing of billing results
MCU
Circular buffer for periodical and non-periodical results EPROM FLASH SRAM
JEIDA 4.0
Clock
Tariffs
Processing of tariff program and time functions
Sync
Saving
Circular buffer for periodical and non-periodical results
Communication
Programmable channels
OPT
COM 1
COM 2
Vi
Vo

Figure 3-1 Block diagram of program functions

3.1 INPUTS

For correct processing of input impulses it is necessary to be familiar with their significance. Therefore the data recorder can define the group and type of input in a program. Regarding the input type other parameters of inputs can also be set..
Groups of inputs are

- impulse meter inputs
- serial meter inputs
- testing inputs

Impulse meter inputs and testing inputs are divided to the following types:

- S1a two-wire input for active current
- S1r two-wire input for inactive current
- S2 two-wire input for double current is not realized yet
- S3 three-wire input is not realized yet

Impulse meter inputs
They are intended for a reception of meter impulses. Received impulses are sampled with the highest frequency (t ≤ 10 ms) and tested as described in Item CHECKING INPUT DATA. During the test the detected interference's activate alarm messages.
The type and control parameters of impulse inputs are set in the IPORTPARAM group of registers.

Serial meter inputs
Serial communication inputs of meter values for meter read-out are provided in the register. Data reception on meter impulses is performed via serial interface by the standard communication protocol. Each communication channel has a buffer where data from the read-out meter are stored. These data are kept until the next data read-out from the meter. Data reading on one channel does not influence in data stored in another channel.
The parameters of serial inputs are set in the SERIALPARAM group of registers.

Testing inputs
They are intended for the reception of special signals. One of the following functions cane be defined for each input separately:

- external clock synchronization
- tariff control

3.1.1 Checking input data

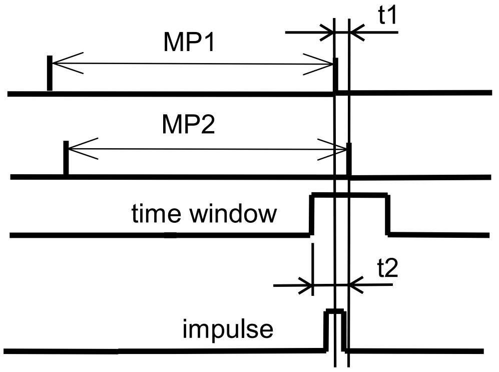
Meter impulses
For the protection of inputs against interference's and false impulses the data recorder tests input meter impulses. The following is tested regarding the input type:

- for two-wire inputs for active or reactive current
  Maximum and minimum impulse length is tested. The impulse length depends on conditions on a line therefore basic impulse length can be set in the program for each input separately. If the received impulse is shorter than defined minimal impulse length, it is not considered. However, if it is longer than maximal defined impulse length, a corresponding alarm is activated.
- for two-wire inputs for double current is not realized yet
  Switching time between polarities of double current is checked. If the impulse switching time is longer than the defined one, an alarm is activated.
- for three-wire inputs is not realized yet
  Simultaneity of switching between lines or a transition time between circuit opening on one line and simultaneous closing of a circuit on another line is tested. If a transition time is longer than the defined one, an alarm is activated and the impulse is sent for further processing.

Digital data on meter status
On serial meter inputs the sent data are tested according to the regulations on communication protocols.

3.1.2 Function of impulse retaining

This function is intended for synchronisation of recorders regarding meter impulses. It is especially used at control measurements when impulses are transmitted from one source to two or more recorders. As the system time of recorders is not completely synchronised in spite of synchronisation, measuring periods termination is not synchronised as well. If an impulse occurs at the measuring period termination, one recorder could count the impulse into a current measuring period, while another one could count it into the next measuring period. Difference amounts to 1 impulse which is still permissible by the majority of users. This function allows that in a certain time window about the measuring period termination the sent impulses are not counted into a current measuring period. Impulses which arrive at that time are temporarily stored into a buffer store. After the function window termination the impulses are transmitted into the next measuring period.
A parameter of this function is a size of a time window which is formed around the measuring period termination. Maximal length of a function window is ±1 s. It is set separately for the period before and after termination of the measuring period in steps by 1 ms. The function is set for each measuring period separately (a group of registers IMPULSEBARRIERE). A size of a temporary memory is defined with a maximal frequency of input impulses and maximal size of a time window.



MP1 = measuring period of 1st recorder
MP2 = measuring period of 2nd recorder
t1 = skew of a measuring period termination of 2nd recorder
t2 = time window size of 2nd recorder in ms

Figure 3-2 Function of pulse retaining

Figure 3-2 shows an example of impulse which occurred at MP1 termination. 1st recorder recognises the impulse as valid only in the next period and processes it. The second recorder recognises the impulse in a current measuring period as a valid one. In this way a number of arrived impulses in a measuring period termination differs on one impulse. If a function of impulse retaining is used on the second recorder, the last impulse is not counted into a current measuring period but is transmitted into the following measuring period. In this way numbers of arrived impulses in the terminated measuring period of both recorders are equalised.

The function influences in measuring results of current measuring periods on all measuring levels except on measuring level 9.

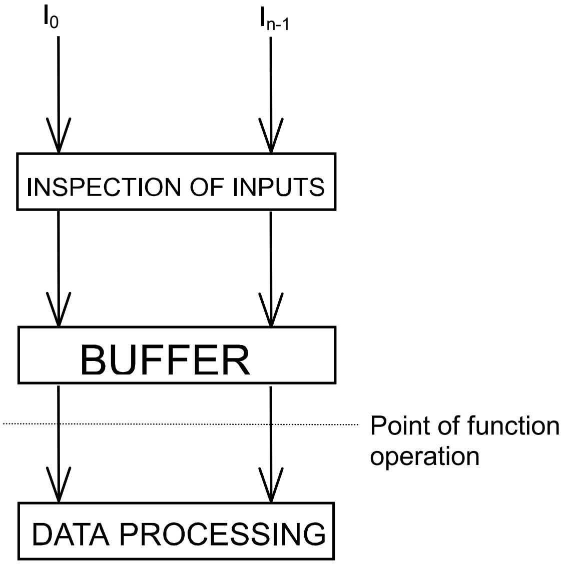


Figure 3-3 Point of function operations

Warning:
The function influences in time sequence of impulses and thus correctly continued processing of results. Therefore it is to be used only in accordance with all participants!
Time window is set in the factory to 0!

3.1.3 Serial meter inputs

Some meters are provided with the RS485 communication buffer for serial gathering of data. The recorder is able to gather such data according to the SDT protocol which is based on DIN 19244 standard. Figure 3-4 shows a network for serial gathering of meters data.

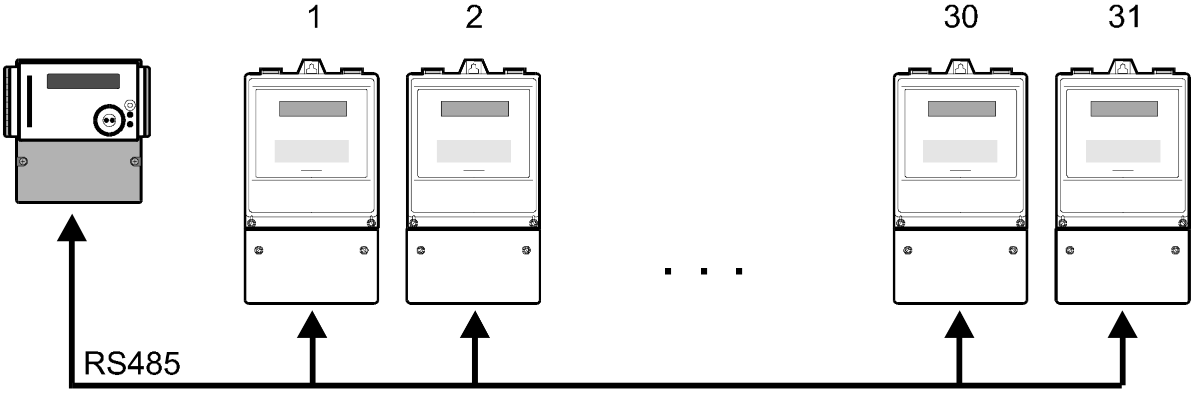


Figure 3-4 SDT network

It is permitted for a single RS485 net to include not more than 31 meters. Each meter should have its own identification number. 0 is a special number. The recorder uses it for simultaneous freezing the results in all the meters.

Meter reading should be carried out periodically by means of time alarms (the ALARMPARAM group of registers). When a time alarm is activated, the data recorder transmits the command “FREEZE” to all meters. That command causes the meter to copy the readings from the counters in order to enter them in the registers that are accessible to data recorder. In the next step the data recorder reads out the registers' values and stores them in the SERIALRESULT group of registers. At the first reading, these values are not entered in the measuring results. The differences between the new and old values are calculated as late as the next meter reading . These differences are then entered in the measuring results (the LEVORESULT group of registers). If for some reason the new value has not been read out or it is false, the old values both in serial and measuring results are not copied. The error type is shown in the STATUS field of the SERIALRESULT group of registers.

When reading a meter it is essential to be familiar both with its identification number in the net and the setting of its registers, because the contents being in individual registers depends on that setting. The setting data are in the meter documentation.

Any data recorder is able to process up to a maximum of 32 serial results. One register from a meter represents a serial result. That means reading one register from 32 meters and/or two registers from 16 meters etc.

The settings for meter reading are in the SERIALPARAM group of registers.

3.2 OUTPUTS

For a communication with connected devices the data recorder is provided with relay outputs. Relay outputs transmit both impulse information and different messages. Transmission function is set for each output separately.

3.2.1 Impulse outputs

Meter impulse outputs
Input meter values and internally obtained results can be transmitted through these outputs in quasi real time. Connection of inputs and/or results to outputs can be adjusted.
The parameters of output impulse lines are set in the OPORTPARAM group of registers.

Time delay
The data recorder is - regarding its processing rate - designed in such a way that the result impulse transmission according to input impulses is performed without noticed delay (quasi real time). It means that the data recorder receives coming impulses with the highest required frequency and stores, processes and transmits them with the highest possible transmitted frequency.

3.2.2 Control outputs

Several data recorder outputs can have the function of transmission and communication. Their function and connection to output contacts is independently defined also for these outputs.

Generation of signals of measuring period termination
All measuring periods can be transmitted. A part of a tariff program is used for generating the signals of the beginning of measuring periods. The measuring period and length of a measuring period signal are defined as a season. The measuring period length is defined as algorithm of repetition of the beginning and the end of one season, while the signal length of the measuring period beginning is defined as the length of one season.
Minimal length of the measuring period signal is 1 s.

Alarm handover
Output relays can be defined as alarm relays which follow the registers in a group of registers SYSTEMATTR. They can have different priorities (e.g.: warning, disturbance).

Tariff generation
Outputs can follow the internal tariff program or control input for tariff and in this way control external devices with tariff signals.

Resetting of meters with integrated tariff device
Data recorder whit firmware 1.21 does not support this function!

3.3 TIME FUNCTIONS

3.3.1 Real time control

POREG2 data recorder controls a real time program clock to which several time functions like measuring period control, control of time alarm repeatability, tariff evaluation of measuring results etc. are connected. Different types of synchronization of a program clock with real time sources are available, e.g. real time internal clock, DCF77 receiver (impulse or telegram), synchronization impulse source, real time source accessible via serial channels.
Synchronization parameters are set in the TIMESYNCHRO group of registers.

3.3.2 Real time program clock

Real time program clock is a real time source for all time functions of data recorder. It has data on a system time (hour, minute, second) and a date (year, month, day, day in a week, summer/winter time). It is possible to set a standard time (winter time) or a time by means of automatic switch-over between a winter and a summer time. Switching over from winter to summer time may be optional. It is set in the DAYLIGHT group of registers.

3.3.3 Internal Real Time Clock (RTC circuit)

A internal real time clock (RTC circuit) is a source which generates an exact information on time and date (year, month, day, day in a week, hour, minute and second). A circuit is battery supported in order to eliminate the possibility of loosing the information on time and date in case of power failure on a recorder main power supply.

3.3.4 DCF77 (option) is not realized yet

Synchronization of real time program clock can be done with a DCF77 real time receiver. Two types of synchronization are available, i.e. with a synchronization impulse or with a DCF77 telegram. In case of a synchronization impulse time is synchronized to the full minute when impulse is present. Such synchronization is possible only in a time window +/-30 seconds from a full minute. In case of synchronization with DCF77 telegram, time telegrams are completely read and plausibility is checked. In case of perfect reception of DCF77 telegram, synchronization of a program clock takes place every minute.
DCF telegram is transmitted from a transmitter in Frankfurt, Germany. The transmitter range is approx. 2000 km. Conditions for reception depend on the distance and local conditions. For a good and reliable reception a good quality antenna is obligatory.

3.3.5 Time functions of program clock

Real time clock is used by a data recorder for several time functions.

3.3.5.1 Measuring periods

Measuring periods represent time periods for collection of measuring results. At termination of a measuring period measuring and billing results are restored and stored. Three measuring periods are available. They can be set in steps of seconds.

3.3.5.2 Time window of impulse retaining

At termination of any measuring period a time window of impulse limitation is generated. At time window duration impulses are captured only and are not processed. A time window is set in steps by 1 msec up to max. ±1 sec.

3.3.5.3 Repeatable time alarms

Repeatable time alarms represent repeatable events which are repeated by certain rules in certain time sequence. When a certain time alarm generates a signal, it is transmitted to all users and different internal functions of the data recorder are activated. Time alarm signals can be used for activation of data saving into profiles, for restoring of billing measuring results, for activation of automatic printing or listing to a terminal etc. In the data recorder there is a certain number of time alarms which can be set with time alarm parameters (group of registers ALARMPARAM). When time alarm for a certain function of the data recorder is used, a reference (time alarm index) is referred to.

Time alarm parameters are:

- for repeatable date
  year, month, day, day in a week and, repetition algorithm
- for repeatable time
  hour, minute, second and repetition algorithm

Possible values for individual parameters are:

year	1990..2089
month	1..12
day	1..31
day in a week	0= Sunday, 1 = Monday... , 6 = Saturday
time	0..23
minute	0..59
second	0..59

Repetition algorithms for date are:

0	once	repeated once exactly on day, month and year
1	yearly	repeated each year on day and month
2	yearlywd	repeated each year on "a day in a week "after "a day" and "month"
3	yearlyns	repeated each year on a day and month; if Sunday, it is moved to Monday
4	yearlyaeast	repeated each year for X days after Easter (X = parameter "day" = 0- 255)
5	yearlybeast	repeated each year for X days before Easter (X = parameter "day" = 0 - 255)
6	monthly	repeated each month on a certain day
7	monthlywd	repeated each month on a day in a week after a certain day
8	weekly	repeated each week on a day in a week
9	daily	repeated each day

Repetition algorithm for time are:

0	once	unique event at time (hour, minute, second)
1	hours	every X hours at a minute or second (X = parameter "hour" = 0 - 23)
2	hour	every hour at a minute or second
3	minutes	every Y minutes on a "second" (Y = parameter "minute" = 0 - 59)
4	minute	every minute on a "second"
5	seconds	every Z seconds (Z = parameter "second" = 0 - 59)
6	second	every second

Time alarm signal is generated when a system time and date comply with time alarm parameters.

WARNING

1. Repetition algorithms for weekdays from 1 to 5 are connected to a calendar year, the 6 and 7 algorithms are connected to a month and the algorithm 8 is connected to a week. The time alarms being set for their activation outside the indicated frameworks will not be properly activated!
2. The first weekday is Monday with the algorithm 8!

3.3.5.4 Billing periods

Billing periods are time periods where billing results are calculated. The billing period termination is activated either by a time alarm or an external impulse. Since a billing period is bound to one of the measuring periods, it is terminated at the end of a measuring period.
For each billing result there are two results, i.e. a result of current billing period and a result of the last terminated billing period. At the end of a billing period a current result is transmitted into a result of the last terminated billing period. A result transmission can be a simple operation of copying or other operation like addition can be used. For a termination of a billing period a signal of time alarm can be used. For individual groups of billing results different time alarms are applied.
More about billing results is stated in Chapter BILLING RESULTS.

3.3.5.5 Data saving periods

Data saving periods are time periods in which data are saved in profiles. Profiles are deposits of data, where the oldest data come out when new ones are added (circular arrangement). Individual profiles can be found on different media for data saving (internal RAM, internal EEPROM, Memory Card etc.). Data, which are available as registers in a data recorder, can be periodically saved. For each profile separately it is possible to say which data are saved in it, and when and where they are to be saved. Time alarm signal can be used for termination of saving period. For individual profiles different time alarms can be used.
More about the application of profiles is given in Chapter SAVING.

3.3.5.6 Control of internal tariff program

Internal tariff program involves elements for tariff evaluation of billing results. These elements are individual rules, like daily rules, weekly rules, seasonal rules, holidays and tariff rules. Tariff results are calculated according to the above rules. Tariff results are calculated for individual rules. Billing results are then connected to individual tariff results and in this way enable multi-tariff measurement.

3.4 MEASURING RESULTS

Measuring results are results of measurements performed in measuring periods. They are organised by levels. Inputs are processed on the lowest level and outputs on the highest. Intermediate levels are intended for different results processing from lower measuring levels.

The results of all measuring levels occur in a form of registers, separately for each measuring level. On each measuring level, except on measuring level 9, measurements of results for 3 measuring periods take place in parallel. Registers on these measuring levels consist of 10 result fields:

- 3 fields for the results of current measuring periods
- 3 fields for the results of previous measuring periods
- 3 fields for cumulative results of measuring periods
- 1 field for cumulative result of measuring periods

On an individual measuring level there are the results of current measuring periods and cumulative current results. At a measuring period termination the result of a current measuring period is transmitted into the result of a previous measuring period. It is then added up to a cumulative result of a measuring period. Then the result of a current measuring period is deleted.

The results of measuring level 9 do not depend on measuring periods.

Figure 3-5 shows transmission of results from lower measuring levels to higher levels.

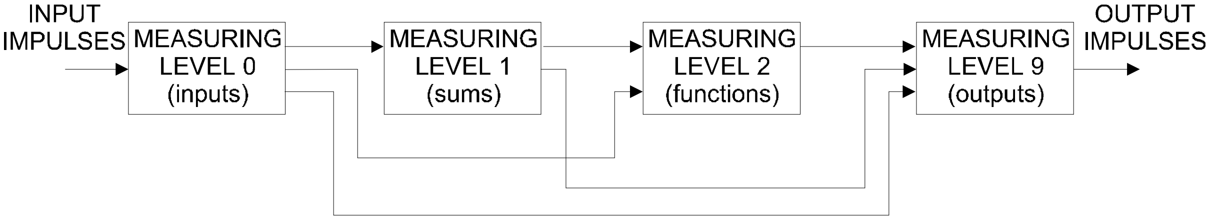

INPUT IMPULSES
MEASURING LEVEL 0 (inputs)
MEASURING LEVEL 1 (sums)
MEASURING LEVEL 2 (functions)
MEASURING LEVEL 9 (outputs)
OUTPUT IMPULSES

Figure 3-5 Transmission of measuring results

The results of lower measuring levels can simultaneously occur on different higher measuring levels. However, the results of lower measuring levels occur in different operations of the same higher measuring level. In this way input impulses can be transmitted to impulse outputs via the results of measuring levels 0 and 9. At the same time, the same results of measuring level 0 form more different sums on measuring level 1. The results of these sums can be used again on higher measuring levels.

On all measuring levels changes of results can be negative or positive. Therefore a filter is included into the results processing. It defines which changes of results are permitted. The filter is stated in a form of a parameter “type” for each measuring result on all measuring levels separately. The following settings are available:

- NULL A measuring result is not changed
- U_COUNTER_B BINARY UP COUNTER; Only positive changes are permitted. The state can be changed from -2^{31} to +2^{31}. Overflow of maximal value changes the result state to a minimal value!
- U_PCOUNTER_B BINARY UP POSITIVE COUNTER; Only positive changes are permitted. The state can be changed only from 0 to +2^{31}. Overflow of maximal value changes the result state to a minimal value!
- UD_COUNTER_B BINARY UP/DOWN COUNTER; Positive and negative changes are permitted. The state can be changed from -2^{31} to +2^{31}. Overflow of maximal value changes the result state to a minimal value and vice-versa!!
- UD_PCOUNTER_B BINARY UP/DOWN POSITIVE COUNTER; Positive and negative changes are permitted. The state can be changed only from 0 to +2^{31}. Overflow of maximal value changes the result state to a minimal value and vice-versa!
- U_COUNTER_D DECADE UP COUNTER; Only positive changes are permitted. The state can be changed from -99999999 to +99999999. Overflow of maximal value changes the result state to a minimal value!
- U_PCOUNTER_D DECADE UP POSITIVE COUNTER; Only positive changes are permitted. The state can be changed only from 0 to +99999999. Overflow of maximal value changes the result state to a minimal value!

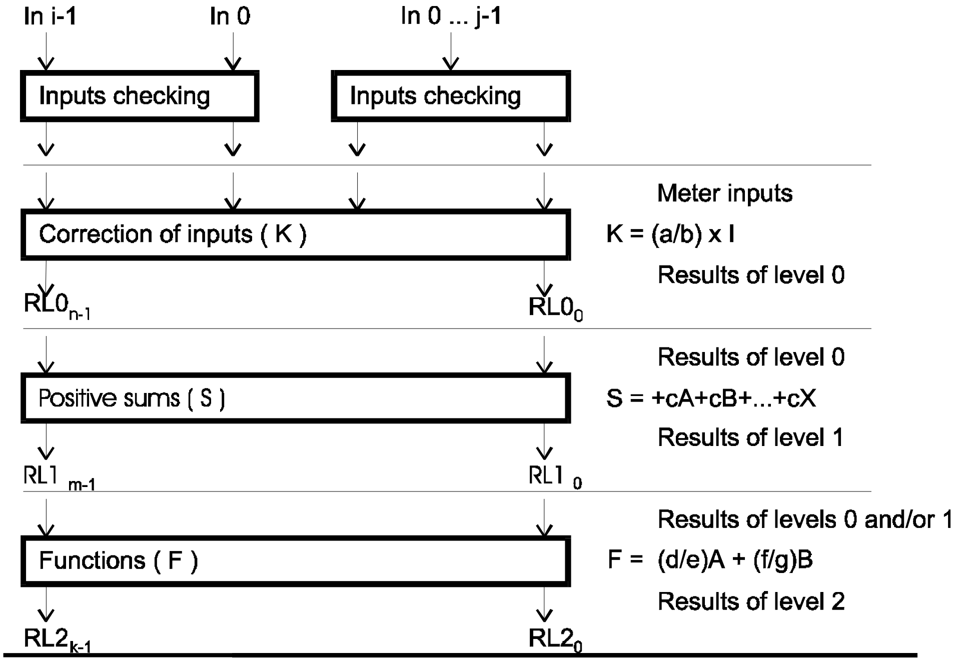
- UD_COUNTER_D DECADE UP/DOWN COUNTER; Positive and negative changes are permitted. The state can be changed from -99999999 to +99999999. Overflow of maximal value changes the result state to a minimal value and vice-versa!!
- UD_PCOUNTER_D DECADE UP/DOWN POSITIVE COUNTER; Positive and negative changes are permitted. The state can be changed only from 0 to +99999999. Overflow of maximal value changes the result state to a minimal value and vice-versa!

Impulse processing

Impulses enter the recorder via impulse inputs (In(0) - In(i-1)) and/or via serial channels (In(0) - In(j-1)). After the inspection of inputs the impulse values on individual inputs are adapted to the recorder constant (CORRECTION OF INPUTS - measuring level 0).The recorder constant is optional. Decimal values are recommended which facilitates further processing (especially at generation of impulses and listing of results).

On measuring level 1 positive sums are calculated. Input data are results of measuring level 0. As input data are separated by inputs, optional input data can be included into an individual sum. The same input data can simultaneously occur in several sums but not in the same one.

Measuring level 2 is intended for calculation of different functions. The results of measuring levels 0 and 1 can occur as input data. The same input data can simultaneously occur in several functions.



Results have a form of registers:

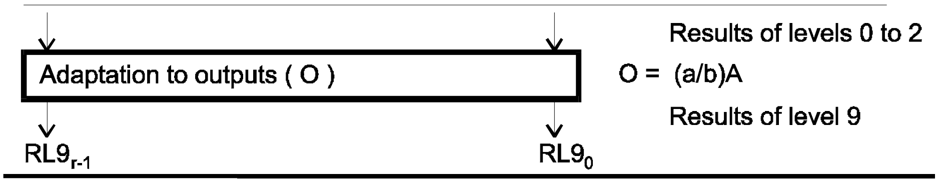
Current MP1	Field 0
Current MP2	Field 1
Current MP3	Field 2
Previous MP1	Field 3
Previous MP2	Field 4
Previous MP3	Field 5
Cumulative MP1	Field 6
Cumulative MP2	Field 7
Cumulative MP3	Field 8
Current cumulative	Field 9

i = number of meter impulse inputs
j = number of meter serial inputs
n = common number of meter inputs
m = number of positive sums
k = number of functions

a,b,..,g = integer coefficients
A,B,...,X = results of lower levels

Figure 3-6 Measuring results - from levels 0 to 2

The results from previous measuring levels can be sent to impulse outputs. On measuring level 9 these results are adapted from a recorder constant to an optional external constant.



a,b = integer coefficients
A = results of lower levels
r = number of impulse outputs

Figure 3-7 Measuring results - level 9

3.4.1 Adaptation of values of input meter impulses - measuring level 0

Owing to easier and faster data processing the values of input meter impulses are converted into impulses with the recorder value, i.e. the pulse transmitter (meter) constants are converted to the recorder constant. It has to be warned that the recorder constant is apparent and is not recorded in any register. The user should be careful to consider the right constant at further processing. It is possible to define different recorder constants at completely separated processing. However, a mix of the results of the processing will not produce correct results.
Conversion is done by means of correction factor written in a form of an integer fraction where input energy (E_{IN}) and energy in a recorder (E_{REC}) are equal:

$$E_{IN} = E_{REC}$$

Energy equals to the product of impulse value (k = impulse constant) and impulse number (n):

$$E_{IN} = k_{IN} \times n_{IN}$$
$$E_{REC} = k_{REC} \times n_{REC}$$

Energy equality gives:

$$k_{IN} \times n_{IN} = k_{REC} \times n_{REC}$$

Ratio between the constants or a number of impulses gives a correction factor (a/b):

$$\frac{k_{IN}}{k_{REC}} = \frac{n_{REC}}{n_{IN}} = \frac{a}{b}$$

The above formula shows that the ratio between a number of impulses is inversely proportioned to the ratio between constants.

Correction factor is usually calculated from the meter and the recorder constants expressed in energy per impulse.
At an entry of a correction factor into parameters of a measuring result (a group of registers LEV0PARAM) a sign of a correction factor can be additionally defined. In this way a sign of result change can be defined but a sign should comply with a type of a result change.

A principle of conversion is valid both for impulse meter inputs and serial meter inputs. Conversion is determined for each result separately.
At impulse conversion integer mathematics with a transfer of a remainder is used at conversion of impulses. This means that at calculation only the whole result is considered and the remainder is transferred to the next calculation. The basic formula, which is used in the recorder for calculation of impulses, is as follows:

$$R0_i = \frac{(O + a_i)}{b_i}$$

R0 is the result of measuring level 0, **O** is a remainder of previous calculation, **i** is the result index. Parameters **a** and **b** represent a correction factor, calculated by the above procedure.

Example:
A constant of a pulse transmitter (k_{IN}) is 600 imp/MWh, which means 0.6 imp/kWh or 3/5 imp/kWh. This constant should be converted to the recorder constant (k_{REC}), which is 2 kWh/imp. The impulse value in the recorder is thus 2kWh, and the value of input impulse is 5/3 kWh.
k_{REC} = 2 kWh (impulse value in the recorder)
k_{IN} = 5/3 kWh (input impulse value)

$$\frac{k_{IN}}{k_{REC}} = \frac{5}{3 \times 2} = \frac{5}{6} = \frac{a}{b}$$

The following calculating flow is obtained by the above stated data:

$$R0_i = \frac{(O+5)}{6}$$

Impulse no. n_{IN}	Fraction	Fraction result	Fraction remainder (O)	No. of process. impulse n_{REC}
1	(0+5)/6	0	5	0
2	(5+5)/6	1	4	1
3	(4+5)/6	1	3	2
4	(3+5)/6	1	2	3
5	(2+5)/6	1	1	4
6	(1+5)/6	1	0	5
7	(0+5)/6	0	5	5
8	(5+5)/6	1	4	6

Table 3-1 Conversion of impulses - calculating flow

According to the above procedure the impulses of data recorder n_{REC} of each input (representing the results of measuring level 0 - $R0_i$) are used for further processing in the data recorder:

a) for “on-line” impulse processing (generation of output impulses in quasi real time complying with the results)
b) for digital processing of meter values (sums, balance, value of measuring periods, daily and monthly sums)

As processing of meter values is performed together for meter impulse inputs and serial digital inputs of meter values, in every input module for meter inputs all arrivals of meter impulses are added to previous meter states (8 digits), and are periodically transmitted into digital processing of meter values.

Connection of the result with input lines

Two parameters are used for connection of the result with an input. They define a type of input line (parameter “type_input”) and its number (parameter “index”). An impulse meter input or a serial meter input can be defined as a type of input line. An optional result can be connected with an optional input line. The sequence is not important. In this way input lines can be combined regarding the type of measurement. It is also possible that one input line occurs in several results of measuring level 0. Regarding the type of input line a number of input line indicates:

• impulse meter input	a register number in a group of registers IPORTPARAM which represents a required input
• serial meter input	a register number in group of registers SERIALRESULT which represents a required input

Example:

Two four-quadrant meters with pulse transmitters for each quadrant are connected to the recorder on input lines from 0 to 7 in the following order: A1+, A1-, R1+, R1-, A2+, A2-, R2+ R2-. At parameter setting of measuring level 0 the following group of results is obtained by a sequence of numbers of input lines 0, 4, 1, 5, 2, 6, 3 and 7:

A1+, A2+, A1-, A2-, R1+, R2+ R1- and R2-.

The first two results represent positive active energy A+, the next two represent negative active energy A-, the next two positive reactive energy R+ and the last two negative reactive energy R-.

In this way the results of measuring level 0 are arranged in logical groups which facilitate further processing. This means, for example, that the result 1 of measuring level 0 is not necessarily a reflection of a measurement on input line 1. If this way is used, the user should pay attention to the significance of individual results.

Parameters of measuring level 0 are described in a group of registers LEV0PARAM.

The results are located in a group of registers LEV0RESULT. At connection of results with impulse outputs only the result modification is considered.

3.4.2 Calculation of positive sums - measuring level 1

On this measuring level positive sums are calculated. The results of measuring level 0 are used as input data. As the results of measuring level 0 can have negative values, calculation of differences between individual measuring results is also possible.
Formula for calculation is:

$$R1_i = c_{i,0}R0_0 + c_{i,1}R0_1 + ... + c_{i,n-1}R0_{n-1}$$

R1 is the result of measuring level 1, **i** is the result index, and **n** is a number of registers of level 0. Only different results of measuring level 0 - **R0** can be placed into the sum. They are defined with parameter "**c**". Value 0 indicates that the result is not added to the sum, while value 1 indicates participation of the result in the sum.

At parameter setting it is necessary to take into consideration that the results of measuring level 0 are handled by blocks with 8 results each. A number of blocks is equal to the results of division of all results of measuring level 0 with number 8. This number is a number of fields of registers in a group of registers LEV1PARAM.
Each block represents 8-bit number where each bit represents one parameter “c”. The sequence of parameters is from c_7 to c_0 or c_{15} to c_8 etc. This number is converted into decimal number N_{DEC} for each block separately (a group of registers LEV1PARAM, parameters SIGNX-Y). Conversion is done by formula:

$$N_{DEC} = c_7 2^7 + c_6 2^6 + c_5 2^5 + c_4 2^4 + c_3 2^3 + c_2 2^2 + c_1 2^1 + c_0 2^0$$

Example:
Every second input of 14 inputs is added into the sum. A record by formula for calculation of sums is as follows:

$$R1_i = 0R0_0 + 1R0_1 + 0R0_2 + 1R0_3 + 0R0_4 + 1R0_5 + 0R0_6 + 1R0_7 + \\ + 0R0_8 + 1R0_9 + 0R0_{10} + 1R0_{11} + 0R0_{12} + 1R0_{13}$$

A record is divided to blocks by 8 results each: the first block from $R0_0$ to $R0_7$, the second one from $R0_8$ to $R0_{15}$. As there are no results $R0_{14}$ and $R0_{15}$, their parameter “c” is equal to 0. Parameters for each block separately are:

block 0-7

parameter	c_7	c_6	c_5	c_4	c_3	c_2	c_1	c_0
value	1	0	1	0	1	0	1	0

block 8-15

parameter	c_{15}	c_{14}	c_{13}	c_{12}	c_{11}	c_{10}	c_9	c_8
value	0	0	1	0	1	0	1	0

Calculation of both decimals is:

$$N0_{DEC} = 1\times 2^7 + 0\times 2^6 + 1\times 2^5 + 0\times 2^4 + 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0 = 170 = SIGN0\_7$$
$$N1_{DEC} = 0\times 2^7 + 0\times 2^6 + 1\times 2^5 + 0\times 2^4 + 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0 = 42 = SIGN8\_15$$

The results of measuring level 1 are located in a group of registers LEV1RESULT. At results connection with impulse outputs only modification of the result is considered.

3.4.3 Calculation of functions - measuring level 2

Measuring level 2 enables calculation of different functions from the results of levels 0 and/or 1. Basic formula is:

$$R2 = \frac{d_i}{e_i} A_i + \frac{f_i}{g_i} B_i$$

A and **B** are the results of lower billing levels; **d**, **e**, **f** and **g** are integer coefficients with positive or negative sign. Index "**i**" represents a number of the result of the billing level 2.
Parameters of measuring level 2 are described in a group of registers LEV2PARAM.

R2' is temporary result of measuring level 2. The result of measuring level 2 (**R2')** depends on **R2'** and hysteresis function. Definition of dependency is as follows:
*Relative modification of the result **R2'**, which is greater of double value of hysteresis, is expressed as decrease or increase of the result **R2.***
This means that **R2** will follow **R2'** if the hysteresis value is 0. If it is more than 0, a part of energy will be apparently lost in the hysteresis window. Function of hysteresis is similarly described in continuation.

The results of measuring level 2 can occupy negative values. However, such results can not be used for generation of impulses, but the formula can be changed in such a way that the results are positive. At connection of results to impulse outputs only modification of the result is considered.

Example:
Result R2 is connected to an impulse outputs via the results of measuring level 9. Values of entry quantities are A = 10 and B = 15, hysteresis is 0.

$R2 = R2' = \frac{1}{1}10 + \frac{-1}{1}15 = -5$ as the result is negative, no impulses are generated on output

$R2 = R2' = \frac{1}{1}15 + \frac{-1}{1}10 = +5$ as the result is positive, a corresponding number of impulses is generated on output

The example shows the possibility of separation of output impulses regarding the result sign. Two functions can be generated, as shown above, and the results are connected with two impulse outputs.

Hysteresis
Hysteresis is intended especially for application in the systems with distribution regulation. The recorder can generate impulses which are used for this purpose. If modifications of energy direction are proportionally small and frequent and each modification of energy direction would entail generation of control impulses, hysteresis function is used. It is set in such a way that only major modifications of energy flow cause generation of control impulses.

Warning:
Factory setting is 0! Changes are permitted only if in accordance with all partners involved!

Operation of hysteresis is shown with the following diagram (Figure 3-8).

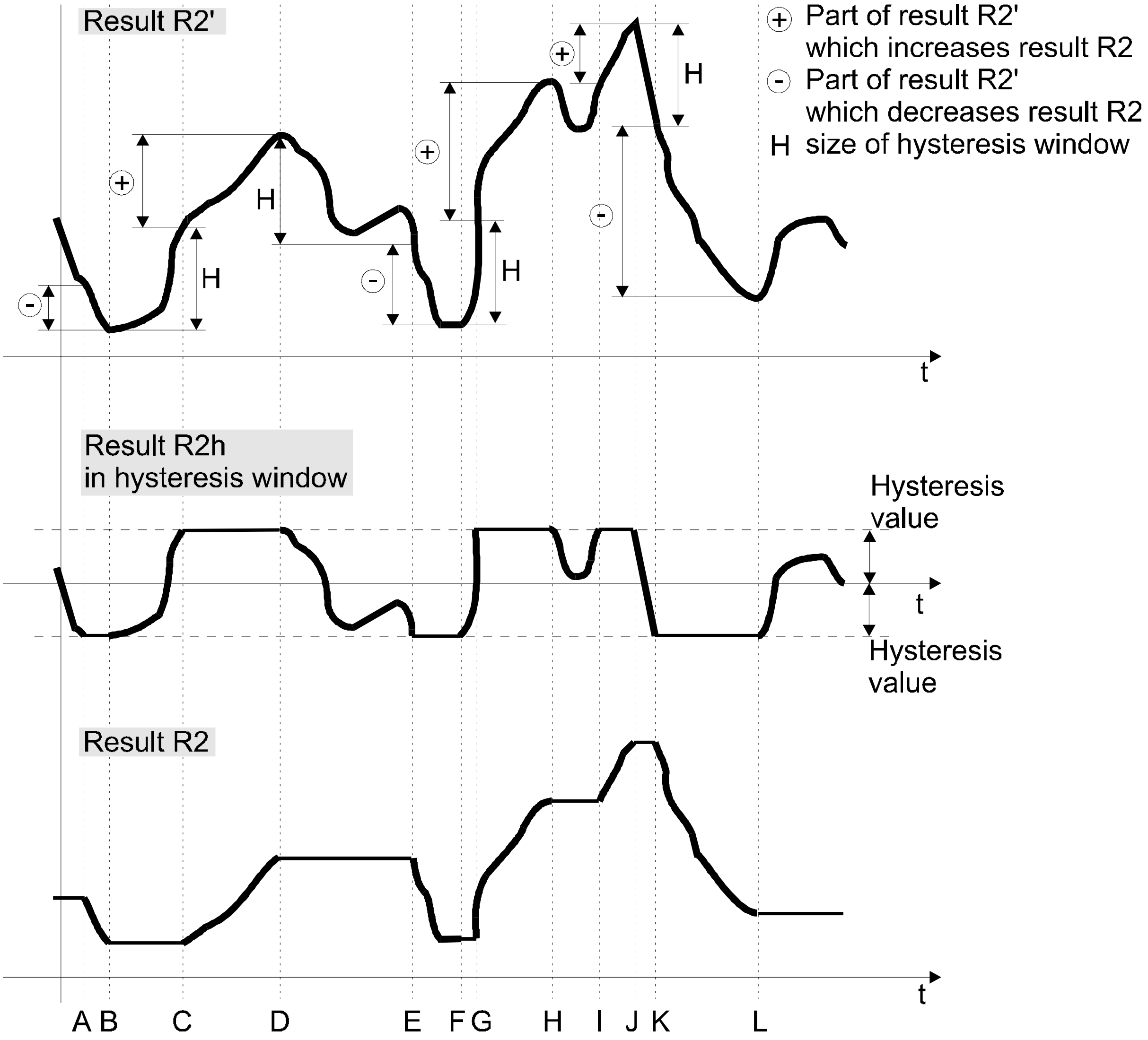

Result R2'
+ Part of result R2' which increases result R2
- Part of result R2' which decreases result R2
H size of hysteresis window
+
-
H
t
Result R2h in hysteresis window
Hysteresis value
t
Hysteresis value
Result R2
t
A B C D E F G H I J K L

Figure 3-8 Hysteresis

An example of a diagram of hysteresis operation is started when the results **R2’** and **R2** have optional values and when the **R2h** value in the hysteresis window is 0. It has to be warned that the values of the results **R2’** and **R2** can be different. The **R2h** result is internal and can not be noticed by the user!

The **R2h** result in the hysteresis window follows **R2’** until a lower hysteresis limit is obtained (point A). As the **R2’** is still falling, this is expressed in the **R2** result while the **R2h** result in the hysteresis window remains on the lower limit. In the same time the hysteresis window is dropping. The **R2** result follows the **R2’** result until the **R2’** result start to rise (point B). Then the **R2** result remains on the obtained value while the **R2h** result is climbing in the hysteresis window (section B-C). The **R2h** result value in point B now represents a lower limit of hysteresis. The **R2h** result in the hysteresis window now follows the **R2’** result until **R2’** modification reaches double value of hysteresis (point C). As the **R2’** result is still rising, this is reflected on the **R2** result while **R2h** remains on the upper limit and the hysteresis window is rising (section C-D). When the **R2’** result begins to fall, the **R2’** value remains on the obtained value (section D-E). In this section the **R2** result is falling first, then it rises and fall again. The oscillation of the **R2’** result is not evident on the **R2** result as the first decrease of the **R2’** result is smaller of double value of hysteresis. When the **R2’** result starts to fall after a short rise, a total decrease of the R2’ result exceeds double value of hysteresis (point E). Further decrease of the **R2’** result is then followed by the **R2** result up to point F where the **R2’** result starts to rise again. As already explained, rising of the **R2’** result is evident on **R2** result only when modification is greater than double value of hysteresis (section G-H). On H-I section smaller oscillation occurs. It does not entail any modification of the **R2** result which is also the intention of hysteresis. The upper limit of hysteresis window is not modified as well. On **I**-J section the **R2** result rises again from the value in point H onwards. Section J-L shows the difference between the modification of the **R2+** and **R2** results.

The results of measuring level 2 are located in a group of registers LEV2RESULT. At connection of results with impulse outputs only modification of the result is considered.

3.4.4 Output results - measuring level 9

Measuring level 9 is intended for conversion of the results of lower measuring levels, expressed in the recorder constant, into output result of different constants.
Conversion is performed similarly as on measuring level 0 by formula:

$$R9_i = \frac{a_i}{b_i} A_R$$

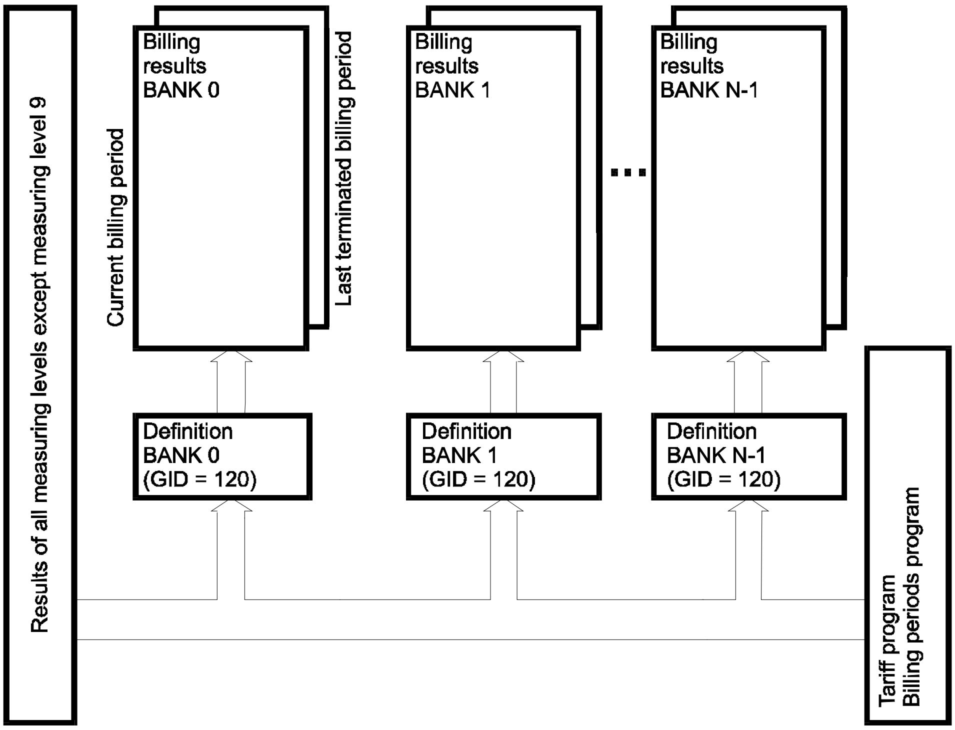
A number of results **R9** is equal to a number of impulse outputs of the recorder **i**. Factors **a** and **b** represent conversion of the **R** result from **A** level (0, 1 or 2).

Parameters of calculation are entered in a group of registers LEV9PARAM, and the results in a group of registers LEV9RESULT.

The results of this level are specific as they are increased regarding the results of lower measuring levels and decreased when output results are generated as output impulses. Generation of impulses on the basis of negative results is not possible.

3.5 BILLING RESULTS

The results of all measuring levels (except the results of measuring level 9), a tariff program and a billing period enter into the processing as input data. Billing results are organised into banks of results and are recorded in a form of registers. Each bank consists of the results of a current billing period and the results of the last terminated billing period with the belonging time marks about the result record. Results of a current billing period are recorded into RAM and the results of the last terminated billing period into EEPROM owing to data security. Regeneration of current billing results is performed separately by tariffs.


Results of all measuring levels except measuring level 9
Current billing period
Last terminated billing period
Billing results BANK 0
Billing results BANK 1
Billing results BANK N-1
...
Definition BANK 0 (GID = 120)
Definition BANK 1 (GID = 120)
Definition BANK N-1 (GID = 120)
Tariff program
Billing periods program

N = Number of banks of billing results = number of registers in a group of registers GID = 120

Figure 3-9 Billing results

A record in a bank (a register) consists of a value (a result) and of a time stamp.

Current billing period

Value 0	Time stamp 0
Value 1	Time stamp 1
Value 2	Time stamp 2
Value 3	Time stamp 3
Value 4	Time stamp 4
Value 5	Time stamp 5
Value 6	Time stamp 6
Value 7	Time stamp 7
(GID = 20)	(GID = 22)

Last terminated billing period

Value 0	Time stamp 0
Value 1	Time stamp 1
Value 2	Time stamp 2
Value 3	Time stamp 3
Value 4	Time stamp 4
Value 5	Time stamp 5
Value 6	Time stamp 6
Value 7	Time stamp 7
(GID = 21)	(GID = 23)

Figure 3-10 Values in banks

Results which will be recorded into billing registers are described with bank parameters. A time stamp indicates a time of a record.

The following parameters are defined to each bank:

- result type; a result can be recorded as energy or power in a period. Power in a period is calculated by formula $V = V_{MP} \times 3600 / MP$. V_{MP} is input value in a measuring period, MP is size of a measuring period in seconds. Each bank can have its own result type.
- updating period; updating of results is connected to one measuring period. Result updating is performed at termination of a measuring period. Each bank can have its own updating period.
- updating; data updating depends on the operation which is performed at termination of updating period. Each bank can have its own updating operation. Possible operations are:
  - **writing**; new data is recorded into the result of a current billing period and previous result is deleted
  - **communication;** new data is added up to the result of a current billing period
  - **maximum**; 1st maximum (M); new data is recorded into the result of a current billing period if it is greater of a momentary entered value; the latter is recorded into 2nd biggest maximum
  - **maximum cumulation;** new data is added up to the result of a current billing period if it is greater of a momentary recorded value
  - **2nd maximum**; 2nd biggest maximum (M2); into a 2nd maximum a value which is greater of a momentary value of 2nd maximum and at the same time smaller of the maximum value is recorded; a momentary value is recorded into 3rd greatest maximum
  - **average of two maximums**; PV2 = (M + M2) / 2
  - **3rd maximum**; 3rd biggest maximum (M3); into a 3rd maximum a value which is greater of a momentary value of 3rd maximum and at the same time smaller of the 2nd maximum value is recorded;
  - **average of three maximums**; PV3 = (M + M2 + M3) / 3
- billing period type; it is defined which event terminates the billing period. The event can be repeatable time alarm or external impulse. When the event occurs, the billing period is terminated and then a time stamp of current billing results is recorded and a billing operation is performed. Each bank can have its type of a billing period.
  Important!
  The billing period termination is connected to the repeatable time termination.
- billing operation; the operation which is performed at transmission of current results into the results of the last terminated billing period. Each bank can have its billing operation:
  - **writing**; new data, which is recorded into the result of the last terminated billing period while the previous one is deleted
  - **communication** new data is added up to the result of the last terminated billing period
  - **writing and deletion** new data is recorded into the result of the last terminated billing period. The result of current billing value is then deleted
  - **cumulation and deletion;** new data is added up to the result of the last terminated billing period. The result of a current billing value is then deleted
- tariff; an individual bank can be connected to the tariff which can be controlled either with a tariff program or external impulse. Non-tariff results can also be recorded.
- measuring results; an individual bank can be connected to registers on an individual measuring level except with registers on measuring level 9. Values from the last terminated measuring periods are used as measuring results.

3.6 TARIFF PROGRAM

A tariff program is a basis for billing results. Structure of a tariff program is shown in the following block diagram (Figure 3-11).

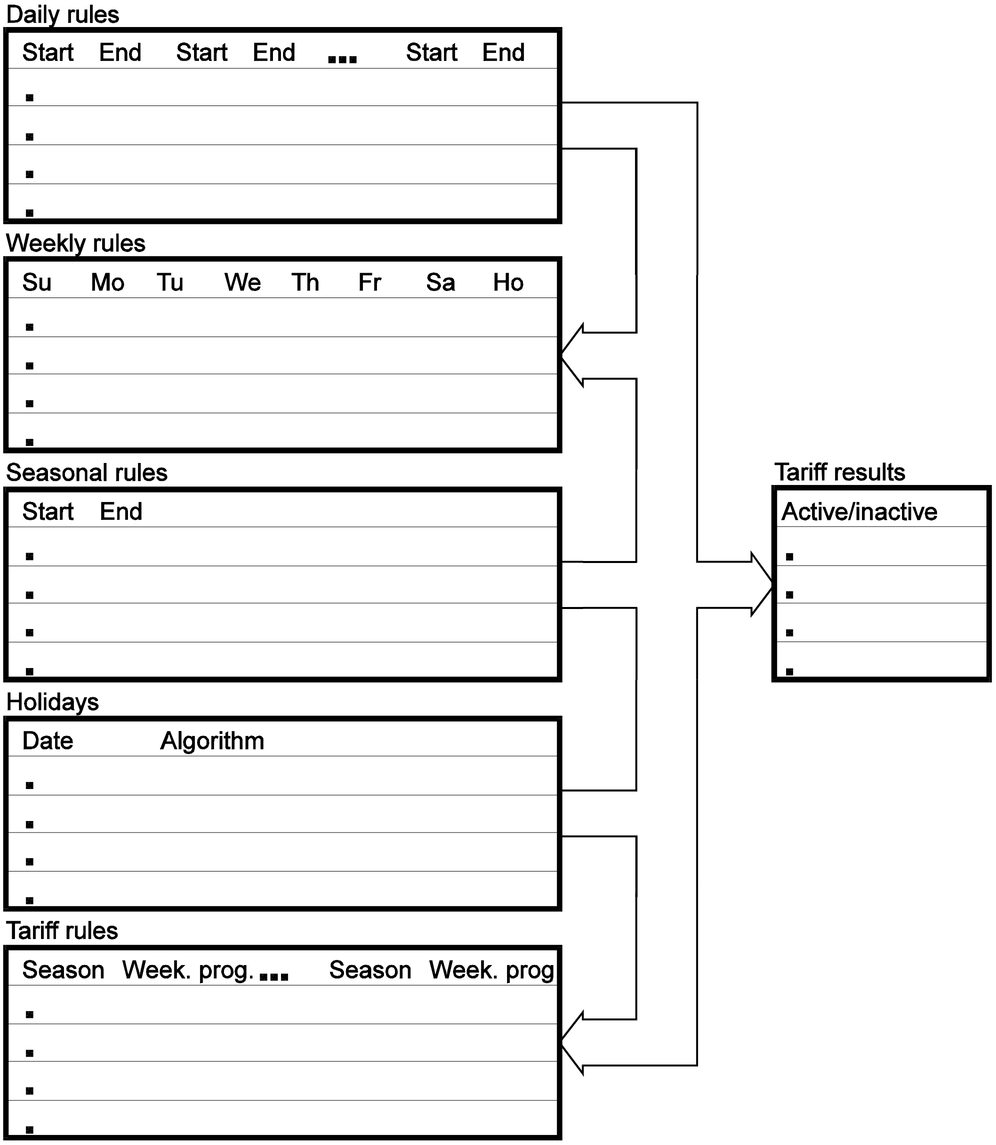


Figure 3-11 Tariff program

A program consists of parameters and results. Parameters are organised as daily rules (a group of registers TARDAYPROG), weekly rules (a group of registers TARWEEKPROG), seasonal rules (a group of registers TARSEASPROG), holidays (a group of registers TARHOLPROG) and tariff rules (a group of registers TARPROG). From parameters the program forms results for each group separately. The results of these groups occur as parameters in other groups and are internally important. Only the results of tariff rules occur as parameters for further processing.

Daily rules

In a group of registers TARDAYPROG the pairs of switching times, which divide a day into several parts, are defined. A pair of switching times consists of time of tariff switch-on (hour, minute) and time of tariff switch-off (hour, minute). One daily rule consists of only the part of the day in which the same tariff is valid. A daily rule is always defined from 00:00 to 24:00. For one day as many daily rules are defined as there are different tariffs in a day.

An example of defining a working day with three tariffs:

1st tariff	2nd tariff	3rd tariff	1st tariff

00:00 6:00 16:00 22:00 24:00

A record of daily rules would be:

	start		end		start		end		
	hour	min	hour	min	hour	min	hour	min	validity
1st rule (0)	**00**	**00**	**06**	**00**	**22**	**00**	**24**	**00**	**1st tariff**
2nd rule (1)	**06**	**00**	**16**	**00**					**2nd tariff**
3rd rule (2)	**16**	**00**	**22**	**00**					**3rd tariff**

A record of a daily rule for a holiday or Sunday when e.g. only 1st tariff is valid would be:

	start		end		
	hour	min	hour	min	validity
4th rule (3)	**00**	**00**	**24**	**00**	**1st tariff**

For further processing it is recommended to define at least one non-defined rule (00:00, 00:00), whose result is always 0. It is used where no daily rule is to be valid.
A invalid rule is when:

- initial and final times are equal
- final time is shorter from the initial
- initial time is shorter than 0 or longer than 23
- final time is shorter than 0 or longer than 24

A record of invalid daily rule (as initial and final times are equal) would be:

	start		end		
	hour	min	hour	min	validity
5th rule (4)	**00**	**00**	**00**	**00**	**invalid rule**

A number of daily rules depends on a number of tariffs in an individual day and a number of different allocations of tariff times in a day regarding the whole year. One more invalid rule is added up to that number. A calendar year can be divided into several seasons. Each season usually has differently allocated tariff times in a day. If allocations of tariff times of one season (possibly only at one season) match with another season, a daily rule of the first season can be used for another season .

Example:
A year consists of three seasons which have the same arrangement of tariff times for the 1st tariff. The second season does not have the 3rd tariff; therefore the 2nd tariff is valid in that time. The 3rd season is equal to the 2nd one, only that a tariff rule for Sunday and a holiday is valid on Saturday. A common number of daily rules would be as follows:

	start		end		start		end		
	hour	min	hour	min	hour	min	hour	min	validity
1st rule (0)	**00**	**00**	**06**	**00**	**22**	**00**	**24**	**00**	**1st tariff for working day for 1st season**
2nd rule (1)	**06**	**00**	**16**	**00**					**2nd tariff for working day for 1st season**
3rd rule (2)	**16**	**00**	**22**	**00**					**3rd tariff for working day for 1st season**
4th rule (3)	**00**	**00**	**24**	**00**					**1st tariff for Sunday and holiday for 1st season**
5th rule (4)	**00**	**00**	**06**	**00**	**22**	**00**	**24**	**00**	**1st tariff for working day for 2nd season**
6th rule (5)	**06**	**00**	**22**	**00**					**2nd tariff for working day for 2nd season**
7th rule (6)	**00**	**00**	**24**	**00**					**1st tariff for Sunday and holiday for 2nd season**
8th rule (7)	**00**	**00**	**06**	**00**	**22**	**00**	**24**	**00**	**1st tariff for working day for 3rd season**
9th rule (8)	**06**	**00**	**22**	**00**					**2nd tariff for working day for 3rd season**
10th rule (9)	**00**	**00**	**24**	**00**					**1st tariff for Sunday, holiday and Saturday for 3rd season**
11th rule (10)	**00**	**00**	**00**	**00**					**invalid rule**

As daily rules for the 1st are equal in all three seasons, daily rules for 1st tariff for 2nd and 3rd seasons can be omitted. The rules for 3rd season can be omitted as well as they are equal to 2nd season. The only difference is that in the 2nd season they are valid from Monday to Saturday, and in the 3rd season from Monday to Friday. In this way only 6 daily rules are defined:

	start		end		start		end		
	hour	min	hour	min	hour	min	hour	min	validity
1st rule (0)	**00**	**00**	**06**	**00**	**22**	**00**	**24**	**00**	**1st tariff for working day for three seasons**
2nd rule (1)	**06**	**00**	**16**	**00**					**2nd tariff for 1st season**
3rd rule (2)	**16**	**00**	**22**	**00**					**3rd tariff for 1st season**
4th rule (3)	**00**	**00**	**24**	**00**					**1st tariff for holiday and Sunday for three seasons**
5th rule (4)	**00**	**00**	**00**	**00**					**invalid rule**
6th rule (5)	**06**	**00**	**22**	**00**					**2nd tariff for 2nd and 3rd seasons**

in a group of registers TARDAYRES there are results of daily rules which show whether current time is in active part of a day (1) or not (0).

Weekly rules

In a group of registers TARWEEKPROG daily rules are defined for individual days in a week and for a holiday. The program checks a current date regarding a day in a week and a holiday. If a current day is a holiday, a daily rule for a holiday is used! If a current day is not a holiday, a program checks a state of the results of daily rules. One weekly rule combines daily rules for one tariff. The result of a weekly rule (a group of registers TARWEEKRES) is active (1) if at least one daily rule is defined for a certain day. Indexes of daily rules records occur as parameters.

For further processing it is recommended to define at least one invalid rule whose result is always 0. It is used where no weekly rule is to be valid. An invalid rule is a rule when an index of a daily rule is recorded as a parameter for days in a week and for a holiday. It is:

- invalid (see an example of an invalid daily rule)
- greater of a common number of all possible daily rules (a number of all registers in a group of registers TARDAYPROG).

Example for invalid record would be:

	Sun	Mon	Tue	Wed	Thu	Fri	Sat	hol	validity
Xst rule (x)	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**invalid rule**

Weekly rules are defined for each season separately. If daily or weekly rules are different in individual seasons, other weekly rules are defined for each season. A number of weekly rules depends on a number of tariffs and seasons. One invalid rule is added to that number.

From the previous example the following would be required:

- 3 rules for three tariffs in 1st season
- 2 rules for two tariffs in 2nd season
- 2 rules for two tariffs in 3rd season
- 1 invalid rule.

As for the 1st and 2nd seasons the rule of the 1st tariff is equal, seven rules are used. Records for weekly rules for both seasons would be:

	Sun	Mon	Tue	Wed	Thu	Fri	Sat	hol	validity
1st rule (0)	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**3**	**1st tariff for 1st and 2nd seasons**
2nd rule (1)	**4**	**1**	**1**	**1**	**1**	**1**	**1**	**4**	**2nd tariff for 1 season**
3rd rule (2)	**4**	**2**	**2**	**2**	**2**	**2**	**2**	**4**	**3rd tariff for 1st season**
4th rule (3)	**4**	**5**	**5**	**5**	**5**	**5**	**5**	**4**	**2nd tariff for 2nd season**
5th rule (4)	**3**	**0**	**0**	**0**	**0**	**0**	**3**	**3**	**1st tariff for 3rd season**
6th rule (5)	**4**	**5**	**5**	**5**	**5**	**5**	**4**	**4**	**2nd tariff for 3rd season**
7th rule (6)	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**invalid rule**

In a group of registers TARWEEKRES there are the results of weekly rules which show whether a weekly rule is active (1) or not (0). The following results are obtained from the above table of record:

Rule	Sunday and holidays	Working days from Monday to Friday				Working days Saturday			
	0 - 24	0 - 6	6 - 16	16 - 22	22 - 24	0 - 6	6 - 16	16 - 22	22 - 24
0	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**
1	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**
2	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**0**
3	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
4	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**
5	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**
6	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Table 3-2 Weekly rules - example

Table 3-2 shows that rules 0 and 4 are active during Sunday and holidays. During working days from Monday to Friday different rules are active, depending on momentary time or a daily rule which is valid at the moment. From 00:00 to 6:00 and from 22:00 to 24:00 rule 0 is active. From 6:00 to 22:00 all three rules are valid at the same time. This is allowed as rule 3 is intended for the 2nd season and rule 5 for the 3rd season. The same is valid also for Saturday when two or more rules are active at the same time, and they are intended for different seasons.

Seasonal rules

They are defined in a group of registers TARSEASPROG. A calendar year is divided to individual seasons. Each season is defined with the time when the season starts (accurate to a second) and the time when it ends (accurate to a second). Both times are defined with repeatable time intervals for date and hour (see item 3.3.5.3.). The program checks whether current date is in active or inactive part of the year for a certain season.

An example of definition of the year 1997 with three seasons.

1st season	2nd season	3rd season	1st season

1.1.1997 00:00:00 — 1.4.1997 3:00:00 — 15.9.1997 2:00:00 — 1.11.1997 00:00:00 — 31.12.1997 24:00:00

A record of seasons would be:

1st rule (0)	start of season										validity
	year	mon	day	Wd	alg	hour	min	sec	alg		**1st season**
	1997	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**		
	end of season										
	year	mon	day	Wd	alg	hour	min	sec	alg		
	1997	**4**	**1**	**0**	**0**	**3**	**0**	**0**	**0**		
2nd rule (1)	start of season										
	year	mon	day	Wd	alg	hour	min	sec	alg		**2nd season**
	1997	**4**	**1**	**0**	**0**	**3**	**0**	**0**	**0**		
	end of season										
	year	mon	day	Wd	alg	hour	min	sec	alg		
	1997	**9**	**15**	**0**	**0**	**2**	**0**	**0**	**0**		
3rd rule (2)	start of season										
	year	mon	day	Wd	alg	hour	min	sec	alg		**3rd season**
	1997	**9**	**15**	**0**	**0**	**2**	**0**	**0**	**0**		
	end of season										
	year	mon	day	Wd	alg	hour	min	sec	alg		
	1997	**11**	**1**	**0**	**0**	**0**	**0**	**0**	**0**		

As the 1st season occurs at the start and the end of a year, the year consists of 4 seasons. Therefore the 4th record is also required:

4th rule (3)	start of season										
	year	mon	day	Wd	alg	hou	min	sec	alg		**1st season**
	1997	**11**	**1**	**0**	**0**	**0**	**0**	**0**	**0**		
	end of season										
	year	mon	day	Wd	alg	hou	min	sec	alg		
	1997	**12**	**31**	**0**	**0**	**24**	**0**	**0**	**0**		

Legend: Wd = day in a week

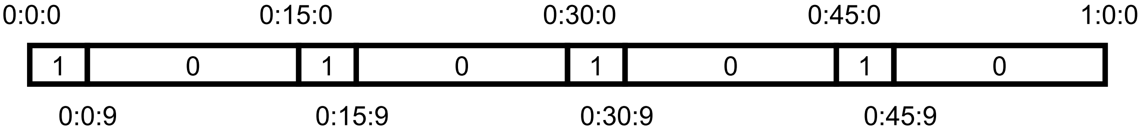
The results of seasonal rules are stated in a group of registers TARSEASRES and indicate whether a seasonal rule is active (1) or not (0).

Their special feature is the possibility of generating impulses of termination of measuring periods. The length of termination of a measuring period is defined as a season. With a corresponding algorithm of repetition of date and time the start of impulse, i.e. the measuring period length is defined. It should be emphasised that there is no connection between definitions of measuring periods (a group of registers MEASPER) and those rules. The user is responsible for their correctness.

Example:
At termination of 15-minute measuring period a 9-second impulse is generated. The record of the rule would be:

5th rule (4)	start of season or impulse									validity
	year	mon	day	Wd	alg	hour	min	sec	alg	
	X	**X**	**X**	**X**	**9**	**X**	**15**	**0**	**3**	
	end of season or impulse									
	year	mon	day	Wd	alg	hour	min	sec	alg	
	X	**X**	**X**	**X**	**9**	**X**	**15**	**9**	**3**	

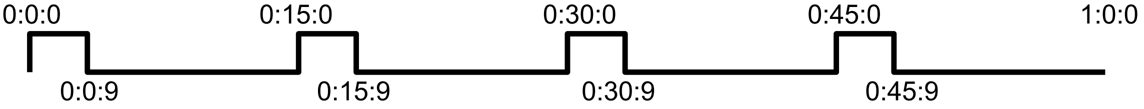
Where parameters **X** are recorded it is not important what is entered as the algorithm of that parameter is not considered. The record of the rule says that the start is repeated every day (algorithm = 9) every 15 minutes at 0 second (algorithm = 3). . Similar is valid also for the end, only that it is repeated every day every 15 minutes at 9th second. Repetition every 15 minutes gives the measuring period length, and the length of season (9 seconds) gives the impulse length.
The result is active for 9 seconds every 15 minutes.



The result is connected with an optional free impulse output via a group of registers OPORTPARAM:

	type	status	gid	rid	fid	Validity
1st rule (0)	**2**	**0**	**92**	**4**	**0**	**Dependency on seasonal result**

Signal form on output would be:



As generation of measuring periods is also connected to the system time as tariff rules are, measuring periods and impulses of termination of measuring periods are synchronised.

Holidays
Individual dates are defined as holidays for the whole year. They are defined in a group of registers TARHOLPROG with a date and the rule for a date which tells the repetition mode (see item 3.3.5.3.). In this way it is possible to define optional holidays. The program checks if the current date is a holiday or not.

Example:
The record of a holiday for 15th August would be:

	year	mon	day	Wd	alg	validity
X. rule (x)	**X**	**8**	**15**	**X**	**1**	**15th Aug for each year**

Algorithm **1** (repetition every year at certain month and day) is used. Therefore parameters “year” and “day in week” are not important.

A record of a rule for Easter Monday would be:

	year	mon	day	Wd	alg	validity
X rule (x)	**X**	**X**	**1**	**X**	**4**	**Easter Monday every year**

Algorithm **4** is used (repetition every year for a number of days (parameter “day”) after Easter). Therefore parameters “year”, “month” and “day in a week” are not important. A parameter “day” indicates a number of days after Easter!

The results of rules for holidays are stated in a group TARHOLRES and indicate whether a current date is a holiday (1) or not (0).

Tariff rules

In a group of registers TARPROG seasonal and corresponding weekly rules are defined as parameters. A number of tariff rules depends on a number of tariff. One tariff rule can consist of several pairs of seasonal-weekly rules.

Example:

A tariff rule with three records of tariff rules (for each tariff separately) can be made from previous examples for weekly and seasonal rules:

	sea1	wee1	sea2	wee2	sea3	wee3	sea4	wee4	validity
1st rule (0)	**0**	**0**	**1**	**0**	**2**	**4**	**3**	**0**	**1st tariff**
2nd rule (1)	**0**	**1**	**1**	**3**	**2**	**5**	**3**	**1**	**2nd tariff**
3rd rule (2)	**0**	**2**	**1**	**6**	**2**	**6**	**3**	**2**	**3rd tariff**

Legend: sea = season
wee = week

Other pairs of seasonal and weekly rules should be defined in such a way that at least one rule (weekly or seasonal) in pair is always invalid. Such pair is represented by two pairs **sea2/wee2** and **sea3/wee3** in record of the 3rd rule where weekly rule 6 is always invalid.

The result (in a group of registers TARRES) is active if both rules are in a pair are active (seasonal and weekly).

3.7 REGISTERS

Data in a data recorder are organized in groups of registers representing data recorder data base. Groups of registers are arranged by different areas like operation parameters, measuring results, test results etc.
Each group of registers consists of registers; the registers consist of fields which are basic units of data recording in a data recorder. Fields can be different types (numerical and text types) in which the values of different data and references to other data are recorded.
A group of registers has its characteristic data, i.e. its construction, rights for access to data, description of individual fields etc.
Characteristic data for a group of registers are:

• identification number of a group	GID
• group attributes	ATTR
• number of registers in a group	RPG
• number of bytes in one register	BPR
• description of fields in a register	FD0, FD1, FD2, FD3, FD4, FD5

Identification number of a group unequivocally identifies a group of registers and enables access to registers and fields in a group.

Attributes of a group define the rights of access to data in a group of registers and some characteristics of a group. The following attributes are available:

• R	reading permitted
• W	writing permitted
• RP	reading protected with a password
• WP	writing protected with a password
• RL	reading protected with a switch
• WL	writing protected with a switch
• PAR	data in a group are parameters
• EE	data in a group are recorded in EEPROM

A **number of registers in a group** indicates a number of registers in a group.

A **number of bytes in a register** indicates total number of bytes in one register regardless a number and type of fields in a register.

Description of fields in a register tells you how individual fields in a register are recorded in a binary form, how many and which type of individual fields there are:

- type of field and his length
  A type of a field indicates a binary record for individual field. Four basic types are defined:
  - signed binary (SBINARY)
  - unsigned binary (UBINARY)
  - BCD
  - ASCII
- number of fields of that type

Individual types are based on 4 types and have the following numerical values:

BYTE	UBINARY+1	unsigned binary; 1 byte long	0 to 255
WORD	UBINARY+2	unsigned binary; 2 bytes long	0 to 65535
WORD24	UBINARY+3	unsigned binary; 3 bytes long	0 to 2^{24}
ULONG	UBINARY+4	unsigned binary; 4 bytes long	0 to 2^{32}
SBYTE	SBINARY+1	signed binary; 1 byte long	-128 to 127
INT	SBINARY+2	signed binary; 2 bytes long	-32768 to 32767
INT24	SBINARY+3	signed binary; 3 bytes long	-2^{23} to 2^{23}-1
LONG	SBINARY+4	signed binary; 4 bytes long	-2^{31} to 2^{31}-1
BCD	BCD+(1-63)	BCD from 1 to 63 bytes long	
ASCII	ASCII+(1-63)	ASCII from 1 to 63 bytes long	

Access to data in a group of registers is possible via an address of individual data consisting of three addresses:

- address of group (GID)
- address of register (RID)
- address of field (FID)

Via the above three addresses a fast and easy access to data in a recorder is possible.

Individual groups of data can be represented as tables of fields (Figure 3-12):

Group X

	field 0	field 1	...	field m
register 0				
register 1				
...				
register n				data a

Figure 3-12 Group of registers

Address of data **a** in a group **X**, with **n** registers and **m** fields in a register is: **X.n.m**

3.8 FORMAT LISTINGS

3.8.1 General

Format listings are intended for writing out of data from data recorders or any formatted text on different output units. Data can be sent to the local display device, terminal output unit or printer. The data to be displayed as well as their format selected. In case of automatic transmission, the time of display can also be chosen. Format listings include particular parameter sub-structures which define the above mentioned functions. These sub-structures are:

- format strings
- format outputs
- format units
- format lists
- format commands

The principle of work with format listings demands first the decision on which data will be displayed and where. You must be aware that the display of data recorder is limited to 2 lines and 16 characters and that the printing output is restricted by the width of paper. Format strings determine the text of listing and the variables, which are written in special symbols. The exact look of listing is further defined by the help of format outputs, where the variables are defined, and by the help of format units. Once all listings are defined and formatted, they can be arranged into format lists where the time sequence of format listings is defined. Several such lists can be determined. In order to get these lists written out you should define where and when a particular list will be displayed by the format commands.
Detailed information is given in continuation.

3.8.2 Format Strings

Format strings present format specifications used to control the display of format outputs. The control of format outputs display means to specify in which way the arguments of these format outputs will be transformed, formatted and written out. Several format outputs can use the same format string. Internally, the format strings are written in the FORMSTR group of registers.

Format string
Format string
...
...
Format string

A format string consists of two types of objects: common ASCII characters, which can be simply copied to the output device, and transformation specifications which seize the arguments from the format output and transform, format and write them out at the output device. The arguments are byte values defined by format outputs.
An example of format string for text »Hello world!« is:

Hello world!

An example of simple specification like time is:

%T

This example shows that a special display form is used for such specification:

% [flags] [width] [.prec] [type modifier] type

Every transformation specification starts with the percent symbol (%) and proceeds with the following strings:

- **flags** : alignment, display of signs; no argument needed
  - - left-aligned specification result
  - + the sign is always present
  - blank the sign is always present (plus as blank)
- **width** : minimum number of characters at output; no argument is needed as it is written directly into the specification. If the specification result has less characters, spaces or zeros (0) are added to the left. A decimal point counts as character.
  - n minimum number of characters at output; spaces added to the left
  - 0n minimum number of characters at output; zeros added to the left
- **.prec** : number of decimal places at output: no argument is needed as it is written directly into the specification.
  - .n number of decimal spaces at output (decimal point does not count)
- **type modifier** : specification modification; argument is needed; there can be only one in one specification!
  - a an argument added to the specification result
  - s an argument subtracted from the specification result
  - m specification result multiplied by an argument
  - d specification result divided by an argument
  - e exponent of specification result
  - p calculation of power during measuring period; argument defines index of measuring period.
    $V = V_{MP}$ x 3600 / MP
    V_{MP} - number of impulses in measuring period
    MP - length of measuring period in seconds
  - r multiple use of a format string; argument defines the number of repetitions (maximum 255)
  - u exponent of result unit; argument defines the register in the UNIT group of registers where the information is read from the first field (200.u.0)
  - t display of values in a date form DD.MM.YY HH:MM W(S)
    W = winter time, S = summer time; no argument needed!
  - x multiple use of a format string (maximum 65,535);
    demands **two** arguments, which are calculated as follows:
    a1 = number of repetitions / 256 (a1 = whole number, rounded down)
    a2 = number of repetitions - (a1 * 256)
- **type** : specification type; there can be only one in a specification!
  - D date; no argument needed, the display format is DD.MM.YYYY
  - T time; no argument needed, the display format is HH:MM:SS
  - F register field from which the information is read; demands three (3) arguments (GID, RID and FID), in case of multiple use of a format string, this is a title of the first register - the register number grows by each repeated display
  - R register from which the information is read; demands two (2) arguments (GID and RID), in case of multiple use of a format string, this is a title of the first register - the register number grows by each repeated display
  - M serial number of multiple use of format string; no argument needed, it always starts counting at 0, tied to the »**r**« or »**x**« type modifier
  - N serial number of display; no argument needed, it always starts counting at 0, tied to the »**r**« or »**x**« type modifier
  - U text of the result unit; argument defines the register in the UNIT group of registers where the information is read from the second field (200.u.1)
  - S profile record stamp; no argument needed, the display format is DD.MM.YY HH:MM W(S), tied to the use of parameter (P) and to the currently defined record
  - P register field in a profile of the last saving period; demands four (4) arguments (GID, RID, FID and PROFILE), in case of multiple use of a format string, the record stamp reduces, keeping pace with the saving period of these profiles

All characters, up to the character which is not indicated above, are taken into account for a specification string. The order of characters is also very important. The character belonging to the upper list but written after the character which should follow it, is read as text! This action interrupts a specification string. The significance of characters depends on the size of letters!
Beside the specifications, which follow the »%« symbol, another independent specification is used for transformation, which is utilized for writing in a new line. The specification format is **n** and demands no argument. The specification is 2 characters long.
The first three places after the percent (%) symbol are meant for the »flags«, »width« and ».prec« characters. They form the display of a variable. Variables are written into a field which has a defined number of characters. This number is defined by the »Width« character. The variable can occupy the whole width of a field or only a part of it. If you wish that the result is always aligned to the left, use »-« for the »flags« character. Otherwise the variable is always aligned to the right side of the field. The variable can also have a negative value. In this case, one place in the field is reserved for the »-« sign.

The following examples demonstrate the use of parameters, but do not explain how the arguments are entered when they are needed. Attributes will be illustrated in detail under the point 3.8.3!

An example of entering the text »R1« and the variable (it will occupy 3 characters) into the field (its width is 9 characters) without a character for left alignment. The »**F**« character means reading of the variable from the register field. It demands three arguments. The first argument defines the group of registers (GID), the second defines the register (RID) and the third defines the field (FID) from which the information will be read. The needed arguments are 10, 0 and 9. In this example, the chosen value in the field is 123.

format string	display
R1 %9F	R1 123

An example of variable entry (3 characters) into a field (9 characters) with a character for left alignment:

format string	display
R1 %-9F	R1 123

When determining the width of field, you can define by entering »**0n**« that the blank space would fill up with the so-called leading zeros when displaying the variable. The »n« parameter defines the width of field.
An example of variable entry (3 characters) into a field (9 characters) without a character for left alignment with leading zeros:

format string	display
R1 %09F	R1 000000123

An example of variable entry (3 characters) into a field (9 characters) with a character for left alignment with leading zeros:

format string	display
R1 %-09F	R1 123

The display shows only the value 123, which means that there are no leading zeros when left alignment is applied.
The »**.prec**« characters define the number of decimal places at variable entry. The characters »**.1**« mean a display to one decimal place, however one place is reserved for a decimal point. Therefore, for display of whole numbers there remains:

total number of characters - sign for decimal point - number of characters for decimal numbers.

An example of variable entry (3 characters) to one decimal point into a field (9 characters) without a character for left alignment with leading zeros:

format string	display
R1 %09.1F	R1 0000123.0

All these characters do not demand specification arguments and are not always necessary for display. Specification arguments are the data which are read from the format outputs by the specification.

The following characters modify specifications (»**type modifier**«) and specification types (»**type**«). With few exceptions, all these characters demand arguments! »Type modifier« characters are not always necessary for display! All »type modifier« characters are small letters.

Specification results can be processed by simple mathematical operations like addition, subtraction, multiplication, division and multiplication by 10^n.

An example of variable entry (its value is 123) where the value **3** is added to the variable (»**a**« parameter). The required arguments are **3**, 10, 0 and 9.

format string	display
R1 %09.1**a**F	R1 0000126.0

A similar example of variable entry (its value is 123) where the value **3** is subtracted from the variable (»**s**« parameter). The required arguments are **3**, 10, 0 and 9.

format string	display
R1 %09.1**s**F	R1 0000120.0

The display differs only in parameter »**s**«. The argument, entered in format outputs, remains the same.

An example of variable entry (its value is 123) where the variable is multiplied by 3 (»**m**« parameter). The required arguments are **3**, 10, 0 and 9.

format string	display
R1 %09.1**m**F	R1 0000369.0

An example of variable entry (its value is 123) where the variable is divided by 3 (»**d**« parameter). The required arguments are **3**, 10, 0 and 9.

format string	display
R1 %09.1**d**F	R1 0000041.0

An example of variable entry (its value is 123) where the variable is multiplied by 10^3 (»**e**« parameter). The required arguments are **3**, 10, 0 and 9.

format string	display
R1 %09.1**e**F	R1 0123000.0

As the arguments are positive values only, the result cannot be multiplied by 10^3! This can be done by dividing the result by 1,000 (»**d**« parameter).

Calculation of power from measuring result for a particular measuring period is also a simple operation. The »**p**« parameter is used here. In this case, the formula is $V = V_{MP}$ x 3,600 / MP.

An example of power display in one minute (60 seconds) measuring period. The index of this measuring period is **2**. Specification, read from the results of measuring level 0, presents energy and its value is 123. The required arguments are **2**, 10, 0 and 9.

format string	display
R1 %09.1**p**F	R1 0007380.0

Many consecutive displays can be defined by one format string. This is made possible by »**r**« and »**x**« parameters. They differ only in that »r« demands one attribute and enables only 255 displays and the »x« parameter demands two attributes and enables 65,535 displays. The arguments can be calculated in the following manner:

a1 = number of repetitions / 256 (a1 = whole number, rounded down)
a2 = number of repetitions - (a1 * 256)

A format string allows the application of only one parameter, i.e. »r« or »x«!

An example of display of five results of measuring level 0 with the values 1111, 2222, 3333, 4444, and 5555. The required arguments are **2**, 10, 0 and 9.

format string	display
R1 %09.1**r**F	R1 0001111.0
	R1 0002222.0
	R1 0003333.0
	R1 0004444.0
	R1 0005555.0

To display 1,000 results, the »x« parameter is used and the two arguments are:

a1 = 1000 / 256 = 3,9	a1 = 3
a2 = 1000 - (3 × 256)	a2 = 232

The required arguments would be **3**, **232**, 10, 0 and 9.

The »**u**« parameter performs the same operation as the »**e**« parameter. However, this argument presents a register index in the UNIT group of registers (GID = 200). From the first field (0) of this register (200.u), a parameter, which can have a negative value, is read. The advantage this parameter has over the »d« parameter (division) is that the largest divisor the latter provides is 255. The »u« parameter is most frequently used in connection with the »U« parameter when displaying the result units.

An example of »u« parameter utilization:
The value of result from the measuring level 0 is 123456. This result is divided by 1000 or multiplied by 10^{3}. Therefore, the text »-3« is written in the **0** register of the UNIT group of registers in the 0 field. The result will be displayed to three decimal places accuracy.
The required arguments are **0**, 10, 0 and 9.

format string	display
R1 %09.3**u**F	R1 00123.456

Parameters »r« and »f« have no influence on the »**u**« parameter.

Data recorders contain also values which represent time. They are entered in the form of a number of elapsed seconds from 1.1.1970 at 00:00:00. The »**t**« parameter is designed for display of these values. It transforms these values into a date. The display of date has the following form: DD.MM.YY HH:MM T. The parameter does not require an argument!
An example of transformation of the value 100000000 into a date is as follows:

format string	display
%**t**F	03.03.73 09:46 W

The characters needed to carry out this display are the »**type**« characters. Actually, they define what will be displayed. All »type« characters are capital letters.

Warning!
One specification can have only one »type« character!

A display of date could be an example of format string with a minimum length. The »D« parameter does not require arguments!

format string	display
%**D**	03.03.1973

The same holds also for display of time, only that here the »T« parameter should be used.

format string	display
%**T**	12:03:55

Considering the warning above, the last example can serve as an illustration of wrong utilization of the »type« characters.

format string	display
%D**T**	03.03.1973**T**

The »T« parameter does not read as »type« character but as text. A correct display would be:

format string	display
%D**%T**	03.03.197312:03:

This display is also not »correct«, because the number of used characters exceeds the number of characters in one line of display. The example is unreadable and if the space is added between the two results, one space for result display is lost again. In this case, the »**n**« specification should be used for writing into the next display line.

format string	display
%D**n**%T	03.03.1973 12:03:55

The »**F**« and »**R**« parameters are used to display the register values. »F« means reading from the register field in a group of registers and requires three arguments: GID, RID and FID. »R« means reading from the entire register in a group of registers and requires three arguments: GID and RID.
A display from a field was presented among the first examples above. The following is an example of reading from an entire register with 10 fields. The value 123 is written into the last field. The required arguments are 10 and 0.

format string	display
%**R**	000000000123

It is evident that the values of individual fields are written out one after another without a break. In this case, the display is still correct. However, if there were two-digit values in each field, the specification result would be too long for complete writing out in a display.

In case of repetition of displays with »r« and »x« parameters, the »R1« text was added in front. This text can change to show the serial number of the used format string (»**M**« parameter) or the serial number of display (»**N**« parameter). Instead of »R1«, »R%M« is written. Both parameters (M and N) start counting from 0 and are tied to the use of »r« of »x« parameters.
The required arguments are **5**, 10, 0 and 9.

format string	display
R%**M** %09.1**r**F	R0 0001111.0
	R1 0002222.0
	R2 0003333.0
	R3 0004444.0
	R4 0005555.0

The »N« parameter is most applicable at format lists with many format strings which are used for multiple displays. Here the »N« parameter gives the total number of displays and the »M« parameter gives the number of displays in a single format string.

An example of two identical format strings in a format list which will show two consecutive strings and display numbers. There are five repetitions. Defining of several format strings will be explained under the point 3.8.4.

format string	display
M%M N%N %09.1rF	M0 N0 0001111.0
	M1 N1 0002222.0
	M2 N2 0003333.0
	M3 N3 0004444.0
	M4 N4 0005555.0
	M0 N5 0001111.0
	M1 N6 0002222.0
	M2 N7 0003333.0
	M3 N8 0004444.0
	M4 N9 0005555.0
	M0 N0 0005555.0

The »N« parameter counts from 0 to the number which is smaller by one than the number of all displays in a format list. In this case it is 9. Then it starts counting from 0 again.

Similarly as the »u« parameter, the »U« parameter is also used for display of result units. All units, which appear in display, are defined in the UNIT group of registers and then they incorporate into individual displays by the »U« parameter. The »U« parameter requires an argument which is tied to the second field in the register.

An example of result unit display where the »**Wh**« unit is written in the second register field 0 in the UNIT group of registers. The result value of measuring level 0 is 123.
The required arguments are 10. 0, 9 and **0**.

format string	display
%09.1F %**U**	0000123.0 **Wh**

Parameters »r« and »f« have no influence on the »**U**« parameter.

A data recorder stores data into profiles. The »**P**« and »**S**« parameters are used to display the stored values. The »P« parameter is used to display the values from the register field in the profile of the last terminated saving period. It requires four arguments: GID, RID, FID and PROFILE. The GID defines the group of registers, RID defines the register, FID defines the field and the PROFILE defines profiles where the saving of data is carried out. The »S« parameter is used for display of time stamp of the stored data in the form of DD.MM.YY HH:MM W(S). It is tied to the »P« parameter. No argument is needed!
An example of display from the **0** profile if the value »**123456**« of the register field 10.0.9 was stored on **11.3.1997** at **14:17**. The required arguments are 10, 0, 9 and 0.

format string	display
%**P**\n%**S**	**123456** **11.03.97 14:17 W**

When a multiple display is used, this is reflected as movement along consecutive time displays in a profile.

Example 1
Format string for display of field energy values (kWh) in a register for four consecutive registers (10.0.0 to 10.3.0) to three decimal places:

Format string: »Act.%aM Energy \n%09.3rduF %U«
Unit 0: -3, » kWh«
Registers: 10.0.0 - 1234567
10.1.0 - 2345678
10.2.0 - 3456789
10.3.0 - 4567890
Arguments: 1, 4, 10, 0, 10, 0, 0, 0
%aM - takes one argument (a=1) and adds it to the serial number of multiple use of the string
%09.3rduF - takes six arguments (r=4, d=10, u=0, F=10.0.0) for the display of value
%U - takes one argument (U=0) for the display of unit
Display: »Act.1 Energy«
»00123.456 kWh«
»Act.2 Energy«
»00234.567 kWh«
»Act.3 Energy«
»00345.678 kWh«
»Act.4 Energy«
»00456.789 kWh«

Example 2
Impulses (current, past and cumulative values) from individual inputs (in order from input 0 to input 7) are stored into profile every 15 minutes. The display should show the stored cumulative values for input 3 for the last hour.
Format string for display of the field 9 of the register 3 of the group of registers 10 in the profile 1 for four consecutive record stamps:

Format string: »Kum. In. 3\n%rP %U«
Unit 0: 0, » imp«
Registers: 10.3.9 - 4567890 entry at the last terminated saving period
10.3.9 - 3456789 entry 15 minutes before
10.3.9 - 2345678 entry 15 minutes before
10.3.9 - 1234567 entry 15 minutes before

Arguments: 4, 10, 3, 9, 1, 1
%rP - takes five arguments (r=4, P=10.3.9.1.) for the display of values
%U - takes one argument (U=1) for the display of unit
Display: »Kum. In. 3«
»4567890 imp«
»Kum. In. 3«
»3456789 imp«
»Kum. In. 3«
»2345678 imp«
»Kum. In. 3«
»1234567 imp«

Example 3
The data are the same as in the Example 1, only that now you observe 1,000 consecutive entries. Instead of the »**r**« parameter, the »**x**« parameter should be used which requires two arguments:
a1 = 1000 / 256 = 3,906 a1 = 3
a2 = 1000 - (256 * 3) = 232

Format string: »Kum. In. 3\n%**x**P %U«
Arguments: **3**, **232**, 10, 3, 9, 1, 1
%rP - takes six arguments (x=3, 232, P=10.3.9.1) for the display
%U - takes one argument (U=1) for the display of unit

3.8.3 Format Outputs

Format outputs are a definition for format listings, which use a format string and arguments of format output. Several format outputs can utilize the same format strings for display of various data from the recorder in the same format configuration. You should define format strings and format arguments for each format output. All format arguments have byte values and are used successively with regard to the individual transformation specifications in a format string. There should be defined format arguments for every format specification, otherwise the result displays are unpredictable. Format arguments which are not used by any format specification are ignored. Internally, the format outputs are written in the FORMOUT group of registers.

Format output 0	Format string	Arg.	Arg.	...	...	Arg.
Format output 1	Format string					
...	...					
...	...					
Format output n	Format string					

A »format string« refers to format string and »Arg« are individual argument values.
An example of format output entry is done on the basis of the Example 1 under the point 3.8.2. The format string is marked by 0 index and has the following form:
»Act.%aM Energy \n%09.3rduF %U«

	Format string	Arg.1	Arg.2	Arg.3	Arg.4	Arg.5	Arg.6	Arg.7	Arg.8
		a	M	d	u	F	F	F	U
Format output 0	0	1	4	10	0	10	0	0	0

The use of the same format string in two format commands with different arguments would then be:

	Format string	Arg.1	Arg.2	Arg.3	Arg.4	Arg.5	Arg.6	Arg.7	Arg.8
		a	M	d	u	F	F	F	U
Format output 0	0	1	4	10	0	10	0	0	0
Format output 1	0	1	7	10	1	10	0	0	1

The example presents the use of the same format string, only that now the second one repeats itself seven times (Arg.2). Thus, the results from the register fields from 10.0.0 to 10.6.0 will be displayed. An entry in register 1 of the UNIT group of registers is used as unit and unit exponent.
The use of this two lists first gives the display of four results followed by other seven, that is 11 altogether.

3.8.4 Format Lists

Format lists are lists of format outputs following each other in a display. In this way, they create a sequence of displays, which follow in time intervals or are periodically displayed as a whole. This depends on how the format list is used. The same format output can be repeatedly included into a format list. Internally, the format lists are written as the FORMLST group of registers where each register refers to format outputs.

Format list 0	Format output	Format output	...	...	Format output
Format list 1	Format output				
...	...				
...	...				
Format list n	Format output				

The reference 255 is used to close the format list.

An example of format list is done on the basis of the example under the point 3.8.3. where both format outputs are repeated twice. They repeat themselves in the following sequence:
format output 0, format output 1, format output 0 and format output 1.
Afterwards, this sequence repeats itself.

	Format output	Format output	Format output	Format output	Format output
Format list 0	0	1	0	1	255

3.8.5 Format Commands

Format commands refer the use of format lists for automatic writing out of format listings. Internally, the format lists are written as the FORMCMD group of registers, which contain information for execution of format commands.

Format command 0	Format list	Output device	Output channel	Time alarm	Delay
Format command 1	Format list				
...	...				
...	...				
Format command n	Format list				

The »**Format list**« parameter refers to the format list and is used by a particular format command. Several format commands can use the same format list.
The »**Output device**« parameter defines a device to which the format list will be sent. Potential devices are:

- local view presentation on device display
- terminal output transfer to terminal device via communication channel
- printer output writing out on printer connected to a communication channel

Several format commands can utilize the same output device.
The »**Output channel**« parameter has various functions with regard to the »Output device« parameter.

- local view »Output channel« = 0 manual changing of displays (Manualscroll)
  »Output channel« = 1 automatic changing of displays (Autoscroll)
- terminal output parameter defines the communication channel
- printer output parameter defines the communication channel

Several format commands can utilize the same output channel.
The »**Time alarm**« parameter refers to the repeating time alarm which starts the display in time intervals. Time alarm is used only for terminal and printer outputs. Format commands, which start simultaneously, are carried out successively.

The »**Delay**« parameter is a time interval between individual displays of a format list. When displaying on local display (»Output device« parameter = 1), this parameter has two meanings with regard to the »Output channel«:

- »Output channel« = 0 "Delay« parameter defines the time from the last pressing on the »ROLL« key to the time of return to automatic change of display
- »Output channel« = 1 »Delay« parameter defines time intervals between individual displays from the format list.

The number of format commands depends on the number of format lists which will be displayed and on the number of devices. At least one format list and one format command for one device are required for writing out by printer or for transfer to the terminal.
The following is needed to write out data on local view:

- at least one format list
- at least two format commands of which one defines the »Channel 0« and the other the »Channel 1«.
- display set on »Autoscroll« functioning (see the chapter on *Local Manipulation*)

In case of terminal or printer outputs, a serial communication channel set to appropriate functioning mode should be chosen as the channel of output device. Thereupon, a display of format list on this channel starts at every time alarm for corresponding format command.
An example of format commands which display a format list on the display of data recorder is done on the basis of the example under the point 3.8.4. The index of format list is 0.

	Format list	Output device	Output channel	Time alarm	Delay
Format command 0	0	1	1	X	5
Format command 1	0	1	0	X	20

This example shows that the format list 0 is displayed on the local display (»Output device« = 1). The output channel 0 (Manualscroll) and the output channel 1 (Autoscroll) should be defined for this purpose. Defining of both channels means the definition of two format commands (0 and 1). As the format list is written out on the local display, the »Time alarm« parameter is not required. Any value can be entered into this field. The »Delay« parameter for the first entry (Autoscroll) defines automatic change of display every five seconds. For the second entry (Manualscroll) it defines automatic change of displays 20 seconds after the last pressing on the ROLL key by which the change of display is carried out manually.

3.8.6 Units

Units are physical values which can be measured (energy, power, flow, etc.) and their results displayed. An exponent and text should be defined for every unit. The exponent tells us how much is one quantum of the result and the text shows how the unit should be written in format listings. There are two indexes at defining the format type, which are tied to the unit:

- u: defines the multiplication factor 10^n; the index reads the first field in »u« register.
- U: the index reads the second field in »U« register.

When defining the specification arguments, attention should be paid that these two arguments point to the same register!

Example:

Internal data recorder unit:	0.1 Wh
Entry in the UNIT group of registers:	-4, »kWh »
Format listing:	09.4 (the form of result display is XXXX.XXXX)

The display of one internal unit is: 0000.0001 kWh

3.9 SAVING

Data can be stored separately into the internal memory (RAM, EEPROM, FLASH), into the external memory unit or into both locations at the same time. The form of data entry is the same for all memory media. They represent the basis for data saving. The principle of saving rests on the distribution of memory media space. Figure 3-13 illustrates the structure of stored data.

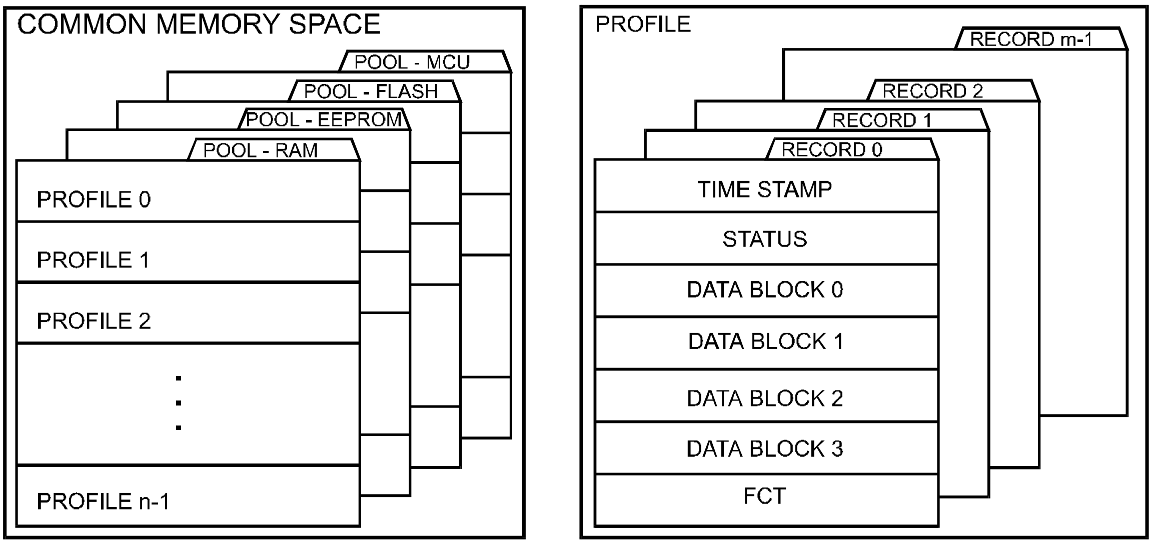


Figure 3-13 Saving structure

The common memory space comprehends all memory media in data recorder, which are designed for saving of data. A memory medium represents the POOL level, but this does not mean that the pool occupies the entire medium space. The pool can be divided into separate PROFILES. The total number of profiles (n) is defined by the number of registers in the PROFILEPARAM group of registers. Profiles can all be located in one pool or distributed among several pools. The size of profile in a pool is defined by the user.

The profile is a circular memory with data arranged in time intervals. Data in a profile are written in RECORDS. The number of records in a profile depends on the size of profile and the size of records. All records in a profile have the same size. The size of record is defined by four data blocks (DATA BLOCK 0 - DATA BLOCK 3), time stamp (TIME STAMP) and control code (FCT).

The time stamp (TIME STAMP) is the entry of time at the beginning of saving period. The time entry is in the form of seconds elapsed from 1.1.1970 at 00:00 hours. The time stamp cannot be accessed directly but can be read together with other data. Detailed information on reading of data from profiles are given in the chapter on *Communication* - PR command for terminal communication and ASDU type 132, ASDU type 133, ASDU type 146 and the ASDU type 147 for communication according to the DIN 19244 protocol.

Information on events which influence the values in a saving period are stored into the STATUS block. The information entry is in bit format and is 16 bits long. The value of information depends on the number of different events which occur during one saving period.
The current values for each profile separately are written into the PROFILESTAT group of registers. The initial value in each saving period is 0. At the occurrence of various events the values of these events are added up. At the end of saving period, these values are stored along with the data. Thus, information on events in the current period is available for each saving period.

Bit	Value	Meaning	
0	1	POWER_FAIL	- Failure of power supply at processor
1	2	POWER_ON	- Activation of power supply at processor
2	4	TYPESYNC_LONGPERIOD	- Saving period is longer due to time synchronization
3	8	TIMESYNC_SHORTPERIOD	- Saving period is shorter due to time synchronization
4	16	DATA_ADJUSTED	- Data were manually adjusted
5	32	DATA_OVERRUN	- Data exceeded the maximum value defined for the data type
6	64	DATA_INVALID	- Data intended for saving were invalid
7-14		UNUSED	- Unused bits
15	32768	SYSTEM_ERROR	- System error which can be generated by examining the Checksum

Table 3-3 Information on saving - STATUS

A saving period starts at the end of measuring period or at the occurrence of internal alarm. The period lasts until the next termination of measuring period or until the next internal alarm.

A control code is a special entry used for checking the correctness of the entered data. The code is calculated by data recorder and the user is denied the access to it.

The stored data are written in register fields of the data recorder. Actually, when the data blocks are defined, the field blocks in individual register groups are defined. A single block of data block is tied to the data from one group of registers only! Record as well as the profile are limited to the data from maximum four register groups!
A description of register block consists of a title and the size of block:

title:	gid	number of a group of registers
	rid	number of the first register in a saving block
	fid	number of the first field in a register
size:	nre	number of registers in a saving block
	nfi	number of fields in a register
	bpf	number of bytes in a field

Figure 3-14 shows a block of data in a group of registers:

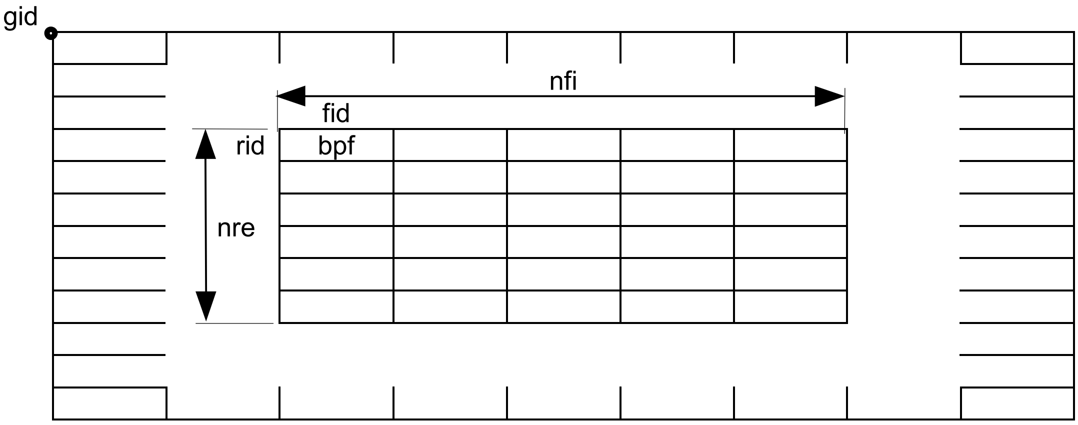


Figure 3-14 Block of registers

The »nre« and »nfi« parameters define the size of data block. They are limited by the size of a group of registers (nre_{max} = number of all registers in a group) and the size of registers (nfi_{max} = number of all fields in a register).
The »bpf« parameter defines the size of data to be stored. This size is not necessarily the same as the size of data at the original location. The field from which the data is read has a certain permanent size. The data in the field assumes this size but does not always reach it. Therefore, the size of data is

reduced to the requisite size by the »bpf« parameter. Attention should be paid that the defined size is not too small, because in this case, the data will be wrong. On the other hand, the size bigger than necessary would mean wasting of memory space.
An example of reduced size is saving of data from a field of four bytes. As the data will not reach the size bigger than two bytes, 2 should be written for »bpf«.
The size of record depends on the amount of data to be stored. The size of data depends on the size of each of the four data blocks. The size of time stamp (4 bytes), the size of status (2 bytes) and the control code (2 bytes) should be added to this. The blocks of data in one record can be of different sizes.

WARNING
Data blocks must not exceed 512 bytes!

The number of records in a profile is defined by the time period of data saving. It depends on the size of a pool and the percentage of space the profile occupies in a pool.
The data recorder calculates saving attributes from these parameters. The user can view them in the PROFILEATTR group of registers. He does not have any influence on these attributes, however, he can check the correctness of saving settings.

Calculation of Data Saving Capacity
Consumption of memory depends on the settings of the aforementioned profiles. The calculation formula is as follows:

Setup checking: location (L), percent (PR), event (E), event index (EI), registers (REG).

Findings:
1. from location (L), you get the POOL size
2. from event (E) and event index (EI), the data recorder calculates the saving period (SP), which can be examined from the profile attributes.
3. from description of registers for appropriate data, you can get the size of memory unit (MU) and the number of data to be stored (ND)

Calculation:
4. Calculation of the record size (RS)
Data is accessible also at profile arguments.
(ND x MU) + 8 = RS (byte) ; 8 the sum of time code(4), status(2) and control code(2)
5. Calculation of profile size (PS)
Data is accessible also at profile arguments.
POOL x PR = PS (byte)
6. Calculation of the number of records in a profile (NR):
PS / RS = NR
7. Calculation of saving capacity for particular data (C)
SP x NR = C (s)

The data in this example of calculation are invented and are as follows:
Saving location is in RAM; percent of the pool memory space intended for data is 100, saving is done for every measuring period; saving period is 15 minutes, data from 16 inputs are stored.
1. The size of pool defined from the location is 200 kbytes (204800 byte).
2. The saving period given by profile arguments is 900 seconds or 15 minutes (information can be read in the profile attributes).
3. Input registers inform that the number of data ND = 16 and that the size of data entry MU = 2 bytes (defined by profile parameters in the PROFILEPAR group of registers).
4. The record size RS = (16 x 2) + 8 = 40 bytes (information can be read in the profile attributes).
5. The profile size PS = 204800 x 100% = 204800 bytes (information can be read in the profile attributes).
6. The number of records NR = 204800 / 40 = 5120
7. Time period C = 5120 x 900 = 4608000s 53.3 days

Calculation shows that the memory of 200 kbytes can store the impulses for 53 days from 16 inputs at 15-minute measuring period and the memory entry of 2 bytes per input.

3.10 COMMUNICATION

3.10.1 General

POREG 2 enables data transmission via interfaces for serial communication (RS232 or RS485, internal modem). These communication channels with their functions permit data transmission between a data recorder and a central place, data transmission between a data recorder and measuring equipment, data recorder parametering, data printing etc. Data can be transmitted via different communication paths, like public telephone network, rented lines, direct connections etc.
Data protection on communication paths is performed with identification codes and passwords. They are also protected indirectly by checking the application of communication paths and its switching-off if there is no communication for more than 10 minutes.

3.10.2 Types of channels

Owing to various devices which are connected to communication channels and which set different requirements various types of communication channels are necessary. For usual communication, in compliance with IEC1107, DIN 19244 and RS standards, a **direct type** of a channel and for PSTN modems an **AT modem** type of channel are used. The difference between them is that the **AT modem** type of channel generates AT commands for control of modem.

3.10.3 Communication functions

Communication channels can perform different functions which can be defined for each channel separately:

- Terminal
- Printer
- DIN 19244 Master
- DIN 19244 Slave
- IEEE 802.3 (is not realized yet)
- IEC 1107 Master
- Cascade
- Master (is not realized yet)
- GPS (is not realized yet)

3.10.3.1 Terminal

Terminal function is intended for data transmission between a data recorder and terminal equipment. Data are transmitted in command mode. The data recorder receives commands and answers them. All commands are in a text form and are terminated with a Carriage Return mark (0Dh). The data recorder answers are in a text form as well. Basic command syntax is the following:

<COMMAND> <PARAMETERS>

The following commands are available:

CO - Connect
Syntax: CO <device code> <user code> <user password>

A command for entry of security code which grants the access to data recorder.
The »device code« is defined by data recorder with which the communication will be carried out, the »user code« and the »user password« determine the rights of access to data recorder. The data recorder has an internal list of users with their passwords and rights. If the »user code« does not

match any of the defined users, the access is denied. If the »user code« is correct and the »user password« is wrong, the user has only the minimum rights.
The »user password« can be entered at a later stage by the use of **PA** command.
Access to data recorder is permitted only for the channel through which this command has been sent and is valid until the locking of data recorder by the **LO** command or until the switching off of data recorder.

PA - Password

Syntax: PA <user password>

A command for entry of the »user password« which determines the rights of access to the data stored in data recorder. In the event of correct entry of the »user password«, you gain the right of access to protected data and is valid until the locking of data recorder by the LO command or until the switching off of the device.

LO - Lock

Syntax: LO

A command for reactivation of prevention of access to protected data. In order to regain permission for the access to protected data the **CO** command should be used.

DA - Date

Syntax: DA <date>

The command for reading and setting up of the system date. The command parameters represent a »date« with the form of DD.MM.YY. If the date is not stated, reading of system date is carried out. Otherwise, the setting up of system time is done. Because the settings are protected by security code, you should have the proper rights. If you do not have these rights, the answer you get is »ERROR«. When reading the date, the answer to command is the system data and when setting up the data the answer is the confirmation of settings. If the date in parameters is correct, the system date changes to the new value and returns the answer »OK«. If not, the system time does not change and the answer is »ERROR«.

TI - Time

Syntax: TI <time>

A command for reading and setting of the system time. Parameters of the command represent time which has the form hh:mm:ss. If time is not stated, reading of the system time is performed, otherwise the system time is set. Because the settings are protected by security code, you should have the proper rights. If permission is not obtained, the answer is "ERROR". At time reading the answer to the command is system time, and at time setting it is confirmation of setting. If time in parameters is correct, the system time is set to the new value and answer is "OK", otherwise the system time is not changed and answer is "ERROR".

FL - Format List

Syntax: FL <format list>

A command for activation of format list. "Format list" parameter is the index of a format list which has to be activated. The answer to the command is a string of listings of format outputs which are followed as defined in the parameters of the format list. Individual listings are terminated with CR.

FO - Format Output

FO <format listing>

A command for activation of listing of the format output. The "format listing" parameter is an index of format output which is to be activated. The answer to the command is listing of the format output which is terminated with CR.

ID - Identification

Syntax: ID

The command for identification of a device. The answer to this command is device identification which consists of fields describing the manufacturer, make, firmware version and checksum firmware with indication of validity.

```
ISKRA  POREG2 VX.XX
Checksum:YYYY  Ok
```

X.XX - version number
YYYY - hexadecimal entry of checksum
The checksum is an arithmetic sum of all memory locations except the last two where the default checksum is written.

RR - Register read

Syntax: RR <register address>

A command for register reading. An individual register field, the whole register, or a complete register group can be read. The register address which is to be read is given as a parameter. The register address has the form of GID.RID.FID, where individual fields identify groups, registers and fields. If the field address is omitted, the whole register is read; if the register address and fields are omitted, the whole group of registers is read. The answer to the command for register reading are register fields which are separated with an interval and terminated with CR. When a group of registers is read, individual registers are terminated with CR. The answer has the following form:

<register address > <register field>

Some registers are protected against reading (attribute RP). For this reason, you should have the proper rights of access to the protected registers. If you do not have the rights of access to the group of registers or if the register address is wrong, the answer is »ERROR«.

RW - Register write

Syntax: RW <register address> <register fields >

Command for recording into the register. It is possible to write into an individual field, the whole register or into a group of registers. The register address and values of field which are to be entered into the register fields are given as parameters. "Register address" has the form of GID.RID.FID. where an individual field identifies a group, a register and a field. If the field address is omitted, writing into all register fields is performed. A number and value of fields which are entered as parameters should match with a number and values of fields which are defined for each group of registers. If it is not known how registers are organized for an individual group, the **RI** command, providing al these data, can be used. The same values are written in all registers in case of entry into all registers. The answer to the command for recording into the register is conformation or non-confirmation of the record:

confirmation: “OK”
non-confirmation : “ERROR”

Some registers are protected against writing (attribute WP). For this reason, you should have the proper rights of access to the protected registers. If you do not have the rights of access to the group of registers or if the register address is wrong, the answer is »ERROR«.

RI - Register info

Syntax: RI <GID>

A command for reading the information on the organization of a group of registers. Information on the organization of a group of registers involves data on a number of registers in a group, a number and type of fields in registers, data on permission for access to the group, etc. The answer to the command for reading the information on the group of registers has the following form:

<information on a group of registers>

Individual fields in information on a group of registers is as follows:

- address of the group of registers
- attributes of the group of registers (expressed as a sum of numerical records of individual attributes)
- number of registers in a group
- number of bytes in a register
- type of the field 0 (expressed as a number)
- number of fields of type 0
- type of the field 1 (expressed as a number)
- number of fields of type 1
- type of the field 2 (expressed as a number)
- number of fields of type 2
- type of the field 3 (expressed as a number)
- number of fields of type 3
- type of the field 4 (expressed as a number)
- number of fields of type 4
- type of the field 5 (expressed as a number)
- number of fields of type 5

8 types of attributes with the following numerical values are defined:

R	permitted reading	1
W	permitted writing	2
RP	reading protected with password	4
WP	writing protected with password	8
RL	reading protected whit a switch	16
WL	writing protected whit a switch	32
PAR	data in a group are parameters	64
EE	data in a group are recorded in EEPROM	128

4 basic types of fields with the following numerical values are defined:

unsigned binary (UBINARY)	0
signed binary (SBINARY)	64
BCD	128
ASCII	192

Individual types are based on 4 types and have the following numerical values:

BYTE	UBINARY+1	unsigned binary; 1 byte long	1
WORD	UBINARY+2	unsigned binary; 2 bytes long	2
WORD24	UBINARY+3	unsigned binary; 3 bytes long	3
ULONG	UBINARY+4	unsigned binary; 4 bytes long	4
SBYTE	SBINARY+1	signed binary; 1 byte long	65
INT	SBINARY+2	signed binary; 2 bytes long	66
INT24	SBINARY+3	signed binary; 3 bytes long	67
LONG	SBINARY+4	signed binary; 4 bytes long	68
BCD	BCD+(1-63)	BCD from 1 to 63 bytes long	128 - 191
ASCII	ASCII+(1-63)	ASCII from 1 to 63 bytes long	192 - 255

More about internal organization of registers is said in Chapter REGISTERS. If a group of registers, for which the information on organization has been required, does not exist, the answer is "ERROR".

PR - Profile Register

PR <profile> <register address> <initial time> <final time >

A command for reading the registers from a profile. It is similar to reading the registers, only that in this case it is necessary to say from which profile and for which time period the registers are to be read. "Profile" is profile index which is to be used for reading. A data recorder supports several profiles which can be situated in different memory media, where different data are saved. "Register address" has the form GID.RID.FID where individual fields identify groups, registers and fields. If the field address is omitted, the whole register is read; if both the register and field address are omitted, the whole group of registers is read. "Initial time" and "final time" have the form DD:MM:YY hh:mm and represent a standard time for the time period for which registers are to be read.
Selected registers are periodically saved into profiles. It means that only those registers are accessible which are saved in the profile. All other registers which are not accessible are provided with the question mark (?) in the answer beside the value. It indicates that the value is not valid. The answer to the command for register reading from a profile has the following form:

<time stamp> <status> <period> <register address> <register fields>

Individual fields in the answer are as follows:

- time stamp
  It indicates when a register was written into a profile. Time stamp is recorded as a number of seconds from 1 January 1970 at 00:00. In answer has a form **DD.MM.YYYY hh:mm:ss dl** where individual fields are:

DD	day in a month (1 - 31)
MM	month (1 - 12)
YYYY	year (1900 …)
hh	hour (0 - 23)
mm	minute (0 - 59)
ss	second (0 - 59)
dl	winter/summer time (W = winter, S = summer)

- status
  Information on the events which influence in storing. Information is recorded in a form of a number. Conversion of this number in a binary record gives bit image. Bits from 0 to 15 represent individual events which have their own values:

Bit	Value	Meaning	
0	1	PWRFAIL	power supply failure at processor
1	2	PWRON	power supply activation at processor
2	4	TIMESYNC_LP	saving period longer due to time synchronization
3	8	TIMESYNC_SP	saving period shorter due to time synchronization
4	16	DATAADJUST	data changed manually
5	32	DATAOVERRUN	data exceeded the maximum value; permission depends on the type of data
6	64	DATAINVALID	data to be stored are invalid
15	32768	SYS_ERROR	system error

  The bits from 7 to 14 are unused.
  The number written in this field is repeated at the end of the saving period. If the number is equal to zero (0), it means that there have been no events which might impact on storing.
- period
  period of writing data into profiles in seconds.
- register address
  Address of the read out register.
- register fields
  Register fields provide values of individual fields in the register.

Some registers are protected against reading (attribute RP). For this reason, you should have the proper rights of access to the protected registers. If you do not have the rights of access to the group of registers or if the register address is wrong, the answer is »ERROR«. You get the same answer if the entry in a profile for a particular time does not exist.

3.10.3.2 Printer

The communication function is intended for communication between a data recorder and the printer. Format listings, defining the printer for the output device, are used for printing. Printings are performed automatically at certain time alarms. Both the contents and the sequence of printings can be defined with parameters of format listings.
More detailed information on format listings is given in the Chapter FORMAT LISTING.

Warning: Communication with a printer is possible only if the printer is provided with a serial port or a converter from a series to parallel communication is used!

3.10.3.3 DIN19244 Master

The communication function is intended for communication between the data recorder and the measuring equipment which supports the serial transmission of meter states in compliance with the DIN 19244 standard. The data recorder functions as a master and periodically collects meter states of the measuring equipment.
Additional data can be found in the DIN 19244 standard; Teil 10, Teil 52, Teil 501 and in standards IEC 870-5-1 to IEC 870-5-5.

3.10.3.4 DIN 19244 Slave

The communication function is intended for communication between the data recorder and the central computer in compliance with the DIN 19244 standard. The function enables transmission of meter states (measuring and billing results), internal events in the device, system time and, by extension, also reading and writing of data recorder registers. Therefore this function is primarily intended for remote readout of measuring results, and secondarily also for the device parametering.
Additional data can be found in the DIN 19244 standard, Teil 10, Teil 52, Teil 501 and in standards IEC 870-5-1 to IEC 870-5-5. Types of fields occurring at transmission are marked as:

type no.	field type	symbol
1	UNSIGNED INTEGER	UI
2	INTEGER	I
3	UNSIGNED FIXED POINT	UF
4	FIXED POINT	F
5	REAL	R
6	BITSTRING	BS
7	OCTETSTRING	OS

DIN 19244 standard, Part 501, specifies transmission of meter data to a primary and secondary station. Data transmission is based on IEC 870-5-1 to IEC 870-5-5 standards which are based on OSI reference model. DIN 19244 Slave function implements the following layers and functions of this model:

Physical layer:
Asynchronous serial transmission, RS232 or RS 485 or a modem, baud rate 150 to 57600, 8 data bits, even parity, 1 stop bit.
Warning: If DIN 19244 Slave function is used, a corresponding serial channel on a data recorder should be set to 8 data bits and even parity, except for communication via modem where there is no parity.

LLC layer:
A telegram format is FT 1.2 of fixed or changeable length. Transmission procedure is asymmetrical (IEC 870-5-2-, paragraph 5). POREG 2 Data Recorder is a secondary station, and a central station is a primary station.
The data recorder implements the following functions of a secondary station on the LLC layer (IEC 870-5-2):

• Function 0	CONFIRM	ACK
• Function 1	CONFIRM	NACK, message is not accepted
• Function 8	RESPOND	User Data
• Function 9	RESPOND	NACK, data not available
• Function 11	RESPOND	Status of link

Application layer:
An application layer implements ASDU structures which represents application data transmitted between a primary and secondary station. A data recorder uses standard and non-standard ASDU structure for transmission of measuring results, parameters and stored measuring results. Non-standard ASDU structures are realized within specified manufacturer's specific ASDU types.

A data recorder implements the following standard ASDU structure:

Primary station

Request of Producer and Product (ASDU type 65)

Request of Messages with time stamp (ASDU type 67)

Request of System time (type 68)

Request of Next ASDU (type 69)

Request of Cumulative Billing Meter - youngest (ASDU type 72)

Request of Cumulative Billing Meter - oldest (ASDU type 73)

Request of Cumulative Billing Meter - time (ASDU type 74)

Request of Cumulative Billing Meter - range (ASDU type 75)

Request of Period Billing Meter - youngest (ASDU type 76)

Request of Period Billing Meter - oldest (ASDU type 77)

Request of Period Billing Meter - time (ASDU type 78)

Request of Period Billing Meter - range (ASDU type 79)

Request of Cumulative Working Meter - youngest (ASDU type 80)

Request of Cumulative Working Meter - oldest (ASDU type 81)

Request of Cumulative Working Meter - time (ASDU type 82)

Request of Cumulative Working Meter - range (ASDU type 83)

Request of Period Working Meter - youngest (ASDU type 84)

Request of Period Working Meter - oldest (ASDU type 85)

Request of Period Working Meter - time (ASDU type 86)

Request of Period Working Meter - range (ASDU type 87)

Secondary station

ASDU without an object (ASDU type 0)
ASDU without an object is used when a certain information in a data recorder is not accessible or is not foreseen.

Manufacturer and product (ASDU type 1)

Messages with time stamp (ASDU type 3)

System time (type 4)

Cumulative Billing Meter - 4 octets (ASDU type 8)

Period Billing Meter - 4 octets (ASDU type 12)

Cumulative Working Meter - 4 octets (ASDU type 24)

Period Working Meter - 4 octets (ASDU type 28)

Meter states transmission involves transmission of internal data recorder registers which represent meter registers. Transmission is performed for both working and billing registers. Transmitted meter values are valid for different time periods (time period, day, month, year). They can be updated with different measuring periods, they can be cumulative or period values, values for a near or distant past. Internal data recorder registers are mapped into individual meter values in compliance with DIN 19244.

Registers representing meter states:

Data recorder register	Register address	DIN meter state	DIN address
Results of level 0	10.0 to 10.31	Billing meter state	0 to 31
Results of level 1	11.0 to 11.15	Billing meter state	32 to 47
Results of level 2	12.0 to 12.7	Billing meter state	48 to 55

Fields in registers are used at individual requirements. if the requirement is for the last measuring period, the values are then collected from the registers, otherwise they are searched in profiles.

Register field	Register field address	DIN request
Current MP1	0	
Current MP2	1	
Current MP3	2	
Previous MP1	3	Period, MP1
Previous MP2	4	Period, MP2
Previous MP3	5	Period, MP3
Cumulative MP1	6	Cumulative, MP1
Cumulative MP2	7	Cumulative, MP2
Cumulative MP3	8	Cumulative, MP3
Cumulative	9	

A data recorder saves only values in measuring periods, therefore daily, monthly and yearly meter states are not realized.

A data recorder does not implements the following standard ASDU structures:

Request of Configuration parameters (type 66)

Configuration parameters (type 2)

Cumulative Billing Meter - 3 octets (ASDU type 9)

Cumulative Billing Meter - 2 octets (ASDU type 10)

Cumulative Billing Meter - 1 octet (ASDU type 11)

Period Billing Meter - 3 octets (ASDU type 13)

Period Billing Meter - 2 octets (ASDU type 14)

Period Billing Meter - 1 octet (ASDU type 15)

Cumulative Working Meter - 3 octets (ASDU type 25)

Cumulative Working Meter - 2 octets (ASDU type 26)

Cumulative Working Meter - 1 octet (ASDU type 27)

Period Working Meter - 3 octets (ASDU type 29)

Period Working Meter - 2 octets (ASDU type 30)

Period Working Meter - 1 octet (ASDU type 31)

A data recorder implements the following non-standard ASDU structures:

ASDU structures have the following form:

DUI
Information object
Common time mark (option)

Primary station

Request for reading registers (ASDU type 140)

A requirement for reading registers is ASDU of a primary station with which a primary station requires reading of registers of secondary station. The requirement is performed for a block of registers in a group of registers. A secondary station answers to the requirement with data of registers (ASDU type 130). ASDU of a requirement for reading registers contains the following information object:

ASDU field	**Contents of ASDU field**	**Field type**
GID	Group address	UI8
RID	Register address	UI8
FID	Field address	UI8
NRE	Number of registers	UI8
NFI	Number of fields	UI8

A request for writing into registers (ASDU type 141)

A request for writing into registers is ASDU of a primary station with which a primary station requests writing into registers of a secondary station. The request is performed for a block of registers in a group of registers. A secondary station answers with a confirmation or refuse of the registers entry. ASDU of a request for writing into registers contains the following information objects:

ASDU field	**Contents of ASDU field**	**Field type**
GID	Group address	UI8
RID	Register address	UI8
FID	Field address	UI8
NRE	Number of registers	UI8
NFI	Number of fields	UI8
REGTYPE 1	Type of data	UI8
REGVALUE 1	Value of data	variable
...		
...		
REGTYPE n	Type of data	UI8
REGVALUE n	Value of data	variable

Address and size of a block of registers being written is stated in the fields GID, RID, FID, NRE, NFI. A contents of individual fields is stated in the fields from REGTYPE 1-REGVALUE to REGTYPE n-REGVALUE n. The number of all pairs of REGTYPE-REGVALUE is NRE x NFI.

REGTYPE represents a type of data which follows REGVALUE. Data recorder registers save values in pre-defined types of data. Details on types of data which are stored in data recorder registers are stated in Chapter REGISTERS.

Request for reading the information on registers (ASDU type 142)

A request for reading the information on registers is ASDU of a primary station with which a primary station requests reading of the information on registers of a secondary station. A secondary station answers to the request with data on information of registers (ASDU type 131). ASDU of a request for reading the information on registers contains the following information objects:

ASDU field	Contents of ASDU field	Field type
GID	Group address	UI8

Request for entry of a protection code (ASDU type 143)

A requirement for entry of a protection code is ASDU of a primary station with which a primary station requires entry of a protection code which prevents the access to data which are protected with it. A protection code has 8 characters. When a protection code is entered correctly, permission for the access to protection data is obtained. ASDU contains the following information objects:

ASDU field	Contents of ASDU field	Field type
PASSWORD	Password	UI8 * 8

Request for activation of a protection code (ASDU type 144)

Request for activation of a protection code is ASDU of a primary station with which the station requests activation of a protection code which prevents the access to data which are protected with it. ASDU is without information objects.

ASDU field	Contents of ASDU field	Field type

Request for system time setting (ASDU type 145)

A request for system time setting is ASDU of a primary station with which a primary station requests setting of a system time in a secondary station. ASDU contains the following information objects:

ASDU field	Contents of ASDU field	Field type
SYSTIME	System time	DINTIME B

SYSTIME represents a system time (DIN 19244, Part 501, item 7.4.3) which is to be set in a secondary station.

Request for reading profile registers (ASDU type 146)

A request for reading the registers is ASDU of a primary station with which a primary station requests reading of registers of a secondary station in profiles. A request is performed as a block of registers in a group of registers for a time period. A secondary station answers to the request with data of profile registers (ASDU type 132) for a time period from the start time to the end time. ASDU of a request for reading the registers contains the following information objects:

ASDU field	Contents of ASDU field	Field type
GID	Group address	UI8
RID	Register address	UI8
FID	Field address	UI8
NRE	Number of registers	UI8
NFI	Number of fields	UI8
PROF	Profile which is read	UI8
FROMTIME	Start time of time period	DINTIME B
TOTIME	End time of time period	DINTIME B

Warning: Start time is included into interrogation while end time is not. All missing values are returned as invalid values.

Request for expanded reading of profile registers (ASDU type 147)

A request for expanded reading the registers is ASDU of a primary station with which a primary station requests reading of registers of a secondary station in profiles. A request is performed as a block of registers in a group of registers for a time period. A secondary station answers to the request with data of profile registers (ASDU type 133) for a time period from the start time to the end time. ASDU of a request for reading the registers contains the following information objects:

ASDU field	Contents of ASDU field	Field Type
GID	Group address	UI8
RID	Register address	UI8
FID	Field address	UI8
NRE	Number of registers	UI8
NFI	Number of fields	UI8
PROF	Profile from which data is read	UI8
FROMTIME	Initial time of time period	DINTIME B
TOTIME	Termination time of time period	DINTIME B

Warning: The initial time is included in query, but the termination time is not. All missing values return as invalid values.

Secondary Stations

Secondary stations return data on request received from the primary station. In case, when the register data are returned, a dynamic adjustment of data types is used for quicker transfer. This means that a data is written out in the shortest time possible. Detailed information of data types are given in the chapter on REGISTERS. Dynamic adjustment of data is used for the ASDU type 130, ASDU type 132 and for the ASDU type 133.

Example:
The data type in register fields in the LEV0RESULT group of registers is ULONG (UBINARY+4). The data is 4 bytes long. In case where the first three bytes equal 0, the data type changes into BYTE (UBINARY+1). Thus, instead of 4 bytes only 1 byte is transferred.

Registers data (ASDU type 130)

ASDU of a secondary station, from which a primary station has requested reading, is registers data. ASDU of registers data contains the following information objects:

ASDU field	Contents of ASDU field	Field type
GID	Group address	UI8
RID	Register address	UI8
FID	Field address	UI8
NRE	Number of registers	UI8
NFI	Number of fields	UI8
REGTYPE 1	Type of data	UI8
REGVALUE 1	Value of data	variable
...		
...		
REGTYPE n	Type of data	UI8
REGVALUE n	Value of data	variable

The address and size of a block of registers are in fields GID, RID, NRE, NFI, while the contents of fields of registers are in fields REGTYPE 1, REGVALUE 1 to REGTYPE n, REGVALUE n. A number of REGTYPE, REGVALUE pairs is NRE * NFI. REGTYPE represents a type of data which follows into REGVALUE. Data recorder registers save the values in pre-defined data types. Details on data types which are saved in data recorder registers are stated in Chapter REGISTERS.

Information on registers (ASDU type 131)

Information on registers is ASDU of a secondary station from which a primary station has requested reading. ASDU information on registers contains the following information objects:

ASDU field	Contents of ASDU field	Field type
GID	Group address	UI8
ATTR	Group attributes	UI8
RPG	A number of registers in a group	UI8
BPR	A number of bytes in a register	UI8
FD0T	Description of a field - field type	UI8
FD0C	Description of a field - a number of fields	UI8
FD1T	Description of a field - field type	UI8
FD1C	Description of a field - a number of fields	UI8
FD2T	Description of a field - field type	UI8
FD2C	Description of a field - a number of fields	UI8
FD3T	Description of a field - field type	UI8
FD3C	Description of a field - a number of fields	UI8
FD4T	Description of a field - field type	UI8
FD4C	Description of a field - a number of fields	UI8
FD5T	Description of a field - field type	UI8
FD5C	Description of a field - a number of fields	UI8

Information on registers contain data on internal organization of a group of registers. More detailed information on organization and concept of registers are stated in Chapter REGISTERS.

Profile registers (type 132)

Profile registers is ASDU of a secondary station from which a primary station requests reading. Data of profile registers contain values of registers for a certain time in the past. Values are collected from a profile which has been set in a request. If for a certain time no values of registers are saved in a profile values are marked as invalid. ASDU of profile registers data contains the following information objects:

ASDU field	Contents of ASDU field	Field type
GID	Group address	UI8
RID	Register address	UI8
FID	Field address	UI8
NRE	Number of registers	UI8
NFI	Number of fields	UI8
PROFTIME	Time stamp for values	DINTIME B
PROFPER	Period of saving into profiles	UI32
REGTYPE 1	Type of data	UI8
REGVALUE 1	Value of data	variable
...		
...		
REGTYPE n	Type of data	UI8
REGVALUE n	Value of data	variable

Address and size of a block of registers are stated in fields GID, RID, FID, NRE, NFI, and contents of fields in registers is stated in fields REGTYPE 1, REGVALUE 1 to REGTYPE n, REGVALUE n. A number of pairs REGTYPE, REGVALUE is NRE * NFI. REGTYPE 0 without REGVALUE indicates that value is invalid or that it does not exist.

REGTYPE represents a type of data which follows into REGVALUE. Data recorder registers save values in predefined types of data. Details on types of data, which are saved in data recorder registers, are stated in Chapter REGISTERS.

Expanded data of profile registers (ASDU type 133)

Expanded data of profile registers are the ASDU of secondary station from which the primary station demands reading. The data of profile registers include register values for a particular time in the past. The values are taken from a profile, set in the demand. In case that register values for a particular time in a profile are not stored, these values are marked as invalid. ASDU of register data includes the following information objects.

ASDU field	Contents of ASDU field	Field Type
GID	Group address	UI8
RID	Register address	UI8
FID	Field address	UI8
NRE	Number of registers	UI8
NFI	Number of fields	UI8
NPER	Number of saving periods into profiles. Gives the number of data blocks in the continuation of transfer.	UI8
PROFPER	Period off saving into profiles	UI32
BLOCK (0)	First block of data t be transferred	
BLOCK (1)	Second block of data to be transferred	
...		
BLOCK (NPER-1)	Last block of data to be transferred	

The address and size of the register block is to be found in the GID, RID, FID, NRE and NFI fields. Data blocks to be transferred have the following structure.

STAMP	Time stamp for data values	UI32
STATUS	Value status	UI16
REGTYPE 1	Data type	UI8
REGVALUE 1	Data value	Changing
...		
...		
REGTYPE n	Data type	UI8
REGVALUE n	Data value	Changing

The contents of individual register fields is written in the fields from REGTYPE 1, REGVALUE 1 to REGTYPE n and REGVALUE n. There are NRE * NFI pairs of REGTYPE, REGVALUE. REGTYPE 0 without REGVALUE indicates that the value is invalid or does not exist.

REGTYPE represents a data type which follow the REGVALUE. Data recorder registers store the values in default data types. Details on data types stored in data recorder registers are given in the chapter on REGISTERS.

Each block of data has a time stamp of data values at the beginning (STAMP) and the status of these values (STATUS). Detailed information on statuses is given in the chapter on STORING.

3.10.3.5 IEEE802.3 Slave **is not realized yet**

The communication function is intended for communication between the data recorder and the central computer in compliance with the IEEE802.3 standard. The function enables transmission of device registers. Therefore this function is primarily intended for remote readout of measuring results, and secondarily also for device parametering.

3.10.3.6 IEC 1107 Master

The data recorder with the 1.21 software version does not support this function!

3.10.3.7 Cascade

This function is intended for serial connection of data recorders through communication channels. For this purpose, the data recorder should have at least two communication channels.

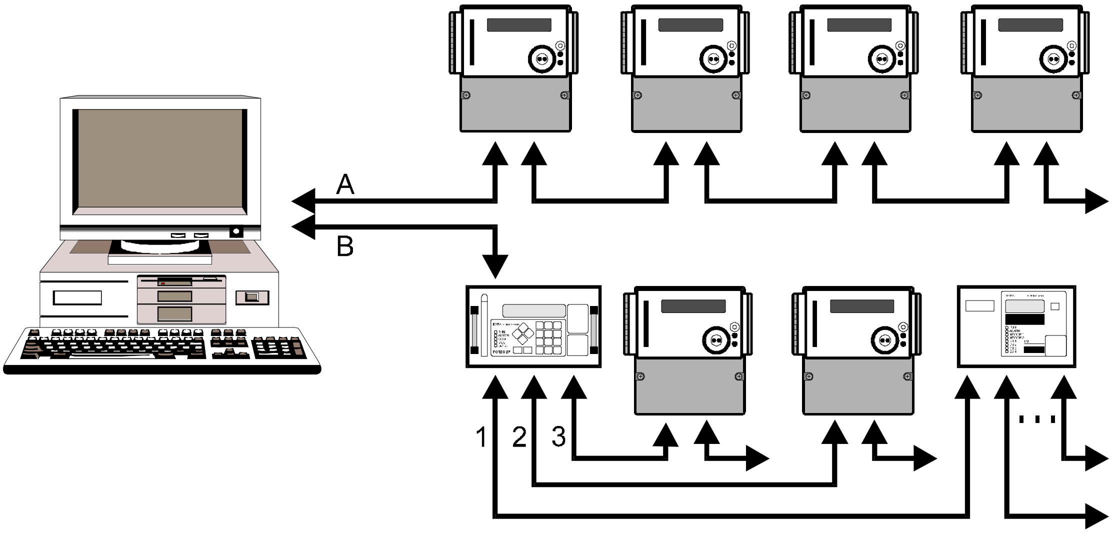


Figure 3-15 Cascade connection - network

Figure 3-15 shows an example of serial connection of different devices. Two independent communication lines A and B are connected to the central station (Master Station). Each line represents a loop. There are four devices connected to the loop A, but their number can also be larger. Devices in one loop can be different and connected in any order. All devices in the loop A use one communication channel as the primary cascade channel and the other one as the secondary cascade channel.

The loop B is an example of star connection of devices. It splits into three loops in the first device (1, 2 and 3). This means that the first device uses one primary cascade channel and three secondary cascade channels. These three secondary cascade channels represent new loops. The loops 2 and 3 continue similarly as the loop A. However, this does not mean that the two loops are unable to split further. An example of additional splitting is the loop 1, which splits into several new loops.

There are various possibilities of connection. However, it holds true for all connections that, from the perspective of the master station, **it is not allowed** to communicate with several devices in a single loop at the same time. For this reason, all devices in a network **should** have different identification numbers.

A communication channel of the selected device opens by an appropriate command (e.g., CO in terminal mode) and an identification number. Every information or demand sent to the loop appears in all devices simultaneously. Yet, only the device with an open communication channel will answer. At termination of communication with the device, its channel should close. Only then the communication with another device is permitted.

This is also valid in case of devices star connection.

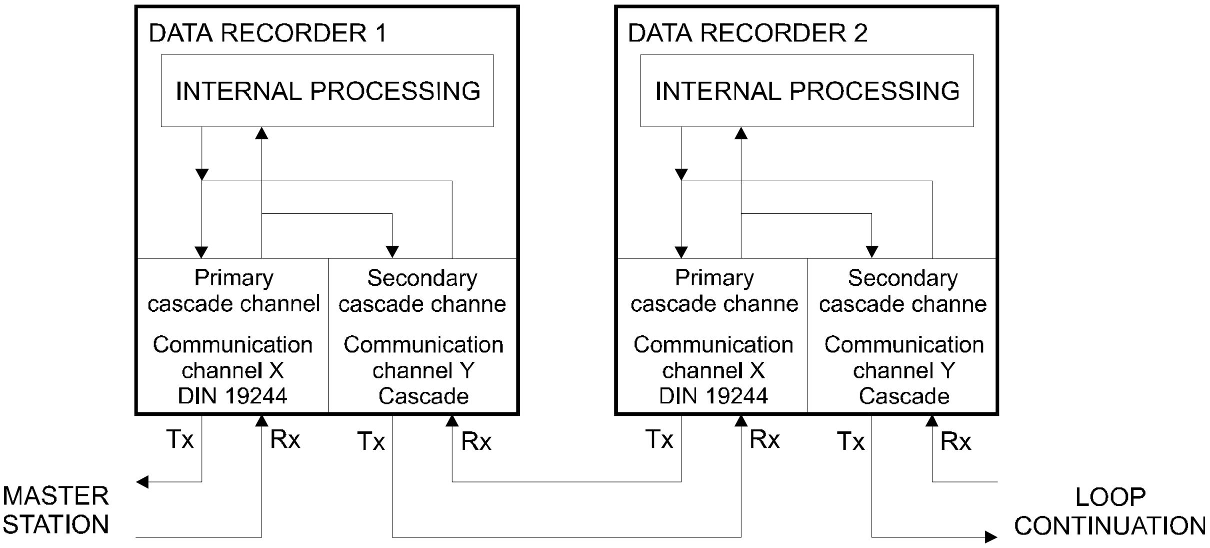


Figure 3-16 Cascade connection - connecting method

Figure 3-16 shows the manner of serial connection of two devices in a loop. Each device can have only one primary cascade channel and several secondary cascade channels. One of the communication channels is chosen for the **primary cascade channel** from the CASCADE group of registers. The function of this channel (set up in the CHANNELPARAM group of registers, parameter F) should not be set on »Cascade«, but on a communication function (in the example above, the two primary cascade channels are set up on DIN192244 Slave).

Communication channel intended for connection with the next device is defined as »Cascade« in the CHANNELPRAM group of registers. In this way, the channel is defined as **secondary cascade channel**, and at the same time, the connection with the primary cascade channel is made.

Primary cascade channels of all devices in a loop are supposed to have the same function. They can have a different function (e.g. Terminal), but at that time, this device is disconnected from communication. In order to communicate with this device, the communication protocol at the master station should be changed. Nevertheless, every device should have its own identification number!

Devices in a loop can work at different speed. It is important at such loop configuration that the speed of the secondary cascade channel of a device matches the speed of the primary cascade channel of the next device.

If the device has more than two communication channels, more communication channels can be defined as secondary cascade channels. All secondary cascade channels can be connected with the primary cascade channel at the same time.

A demand, sent from the master station through the Rx line, occurs simultaneously at the device and its secondary cascade channel on Tx line. Thus, the demand is instantly transferred to all devices in a loop. The data is then returned through the Tx line of the primary cascade channel. If there is another device between the device and master station, these data occur on the Rx line of the secondary cascade channel of the intermediate device and then they are transferred through internal connection to the Tx line of the primary cascade channel. During this process, the returning data do not participate in internal processing and do not influence the functioning of this device.

3.10.4 Modem communication

A modem can be connected to each communication channel which is correspondingly defined (AT modem). Besides other advantages of modem communication the recorder also offers the possibility of defining a daily time window when a modem responds, independent calls of stations as well as a function of call-back (CALLBACK).
Independent calls of stations are performed when time alarm is activated. Stations which are being called are defined in an internal list and are of a primary or a secondary type. Primary stations are a level higher (a central station) while secondary stations are those which are a level lower (a meter, a data recorder, a communicator, etc.).

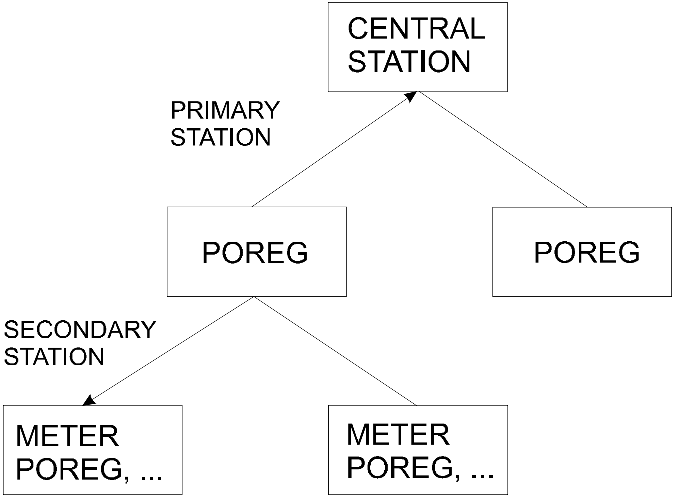


Figure 3-17 Modem communication

CALLBACK function enters additional security element into communication via a modem. The basic purpose of this function is that at external call a caller introduces himself and then the recorder calls him back. It calls him back only when the caller is defined in its internal list of stations and all necessary data for a call are known.
States of a modem channel are shown in the following diagram (Figure 3-18):

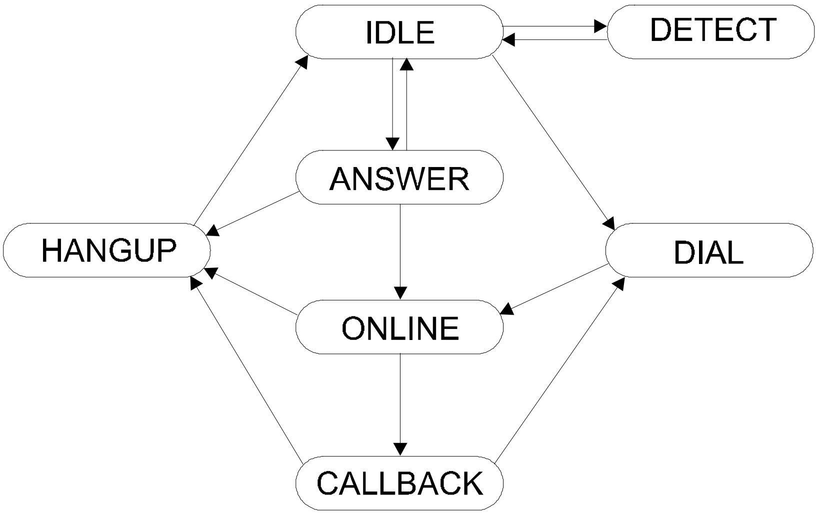


Figure 3-18 Status of modem channel

The channel is usually in IDLE state. The recorder checks the modem presence in intervals on a channel and performs initialisation (DETECT). Then the following functions are available:

- external call; AUTOANSWER function is activated in the modem
  In IDLE state it is waiting for a certain number of ringings. Then the recorder tries to set to a line (ANSWER) via the modem. If setting to the line is unsuccessful, the channel returns to IDLE state. In case of successful setting to a line (ONLINE), communication between devices is performed. At termination the line is interrupted (HANGUP), and IDLE state is restored.
- external cal; CALLBACK function is switched on in the modem
  In IDLE state it is waiting for a certain number of ringings. Then the recorder tries to set to a line (ANSWER) via the modem. If setting to the line is unsuccessful, the channel returns to IDLE state. In case of successful setting to a line (ONLINE) the recorder returns CALLBACK to the caller and interrupts the line. If all conditions are fulfilled for return call (identification string of a caller corresponds to identification string of a station which is recorded in the internal list of the recorder), the recorder calls the caller (DIAL). After successful establishment of a link (ONLINE) and communication between the recorder and the caller, the line is interrupted (HANGUP) and IDLE state is restored.
- automatic call of the recorder when time alarm is activated
  When time alarm is activated, the recorder calls the station which is defined with parameters for that call. After successful establishment of a link (ONLINE) and communication between the recorder and the station, the line is interrupted (HANGUP) and IDLE state is restored.

In ANSWER and ONLINE states the situation on the channel is checked. If there is no data rate in a certain time, the line is interrupted.

3.11 CONSOLE

3.11.1 General

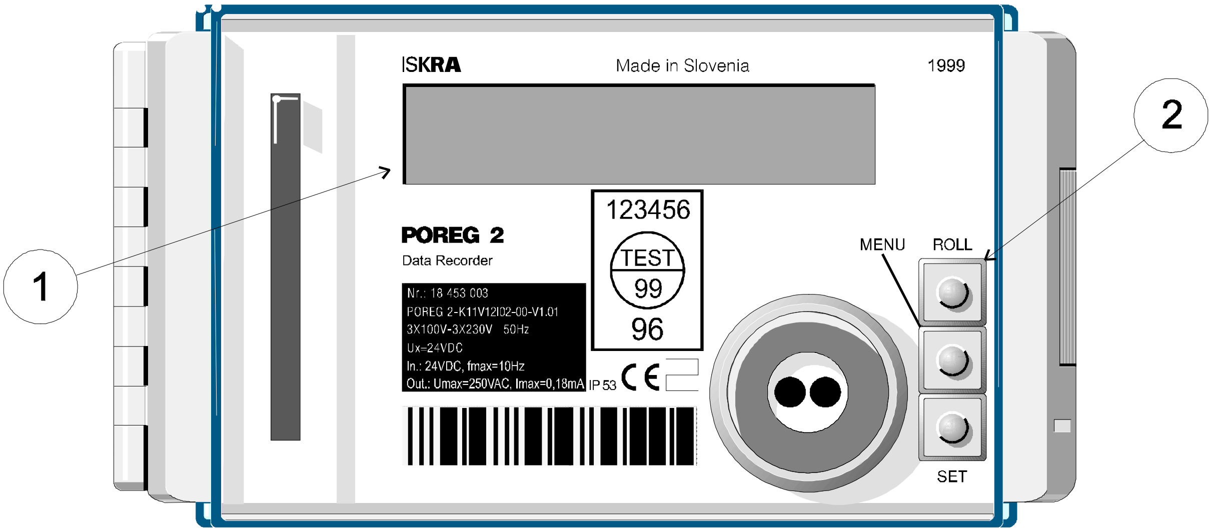


Figure 3-19 Console

The data recorder is provided with a local LCD (1) and keyboard (2) enabling local manipulation with the device. Local manipulation permits a display of measuring results or operation parameters as well as setting of operation parameters. Results can be displayed on request or automatically with the Autoscroll function. For automatic display the format lists are used to specify which information is to be displayed and how fast the format list is to be scrolled. It is also possible to set the operation parameters. However it is recommended - owing to a large number of parameters - to set parameters via communication channels. Local manipulation is performed via the system of menus which permit fast access to individual functions.

3.11.2 Keyboard

POREG 2 data recorder is equipped with a simple keyboard with only three function keys:

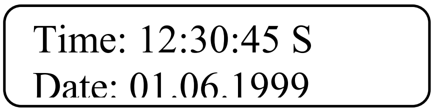
ROLL	- key for moving along the menus (roll) and individual information
MENU	- key for activating the menu system
SET	- key for confirming the selection

Pressing on any key, which is longer than two seconds causes repetition of the function, every 100 milliseconds. The keyboard also enables double-pressing of the key. Double-pressing means to rapidly press and release the same key.

Double-pressing on the ROLL key is used for format displays (Display/Autoscroll), for viewing the contents of registers (Display/Register) and for examination of registers stored in profiles (Display/Profile Registers).

3.11.3 Display

The data recorder is provided with a local LCD enabling alphanumerical display of 2x16 characters. It is intended for a display of measuring results, operation parameters, menus for a selection of functions of local manipulation as well as for a display of other data recorder information.

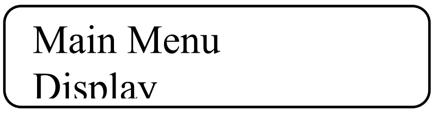


Time: 12:30:45 S
Date: 01.06.1999

When pressing a key, a display illumination is activated. After the last pressing, a display remains illuminated for 3 minutes.

3.11.4 Local manipulation

A local manipulation is performed via a system of menus. It allows a clear and fast access to the functions of local manipulation. The system of menus is activated with the MENU key. The main menu is displayed:



Main Menu
Display

By pressing the ROLL key it is possible to move within the main menu and select one of the options of the main menu.

Main Menu
- Display
- Communication
- Messages
- Function
- ESC

Options of each menu can be submenus or final selections when certain data are entered or a function of local manipulation is activated. Option ESC is intended for restoring a previous level without modifications of settings. When SET key is pressed, a selection of option is confirmed or a required parameter is entered.

<u>*Display menu*</u>

Display menu enables setting the operation mode of a local display. In individual operation modes different information is displayed. The operation mode is selected from the menu Display.

Display
- Normal
- Autoscroll
- Register
- Profile Register
- Input ports
- Output ports
- MCU status
- Channel Status
- Meas. Periods
- Identification
- Protection
- Test Pattern
- ESC

Normal operation mode displays a current time, indication of a winter (W) or summer (S) time as shown below:

```
Time: 12:30:45 S
Date: 01.06.1999
```

Autoscroll function works only if the format listings are defined. Channel 0 as well as channel 1 should be defined in the parameters of format commands. These two channels then use an appropriate list for individual displays. Channel 0 (Manualscroll) is used for writing out of format lists and by successive pressing the ROLL key you can change the listings in a particular list. Channel 1 (Autoscroll) is used for automatic writing out of format lists where individual displays from the list follow in certain time intervals. The format of displays and their sequence is defined by the parameters of format listings (see the chapter on *Format Listings*). The following example of display shows the value of register for active energy 1:

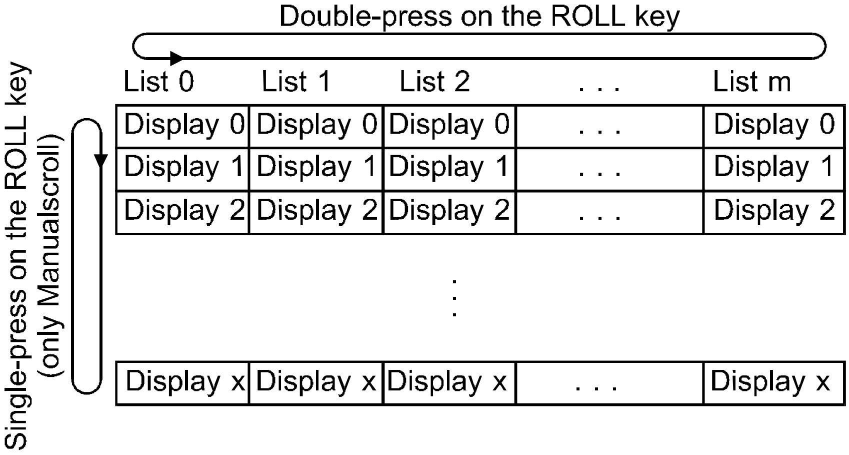
```
Act.1 Energy
12345.678 kWh
```

After the confirmation of selection, the »Autoscroll« mode of display starts and when the ROLL key is pressed, the »Manualscroll« mode of display begins. You can exit both functions by pressing the MENU key.

You can view several format lists in both modes. You select among the lists by double-pressing on the ROLL key. If the display is in Autoscroll mode, you can choose among all lists, which are defined for automatic change of displays (channel 1). If the display works in the Manualscroll mode, you can choose among all lists, which are defined for manual change of displays (channel 0). When you stop the manual view of displays, the data recorder switches to Autoscroll mode after certain time (see Format Commands). The register will now show the first among the format lists defined for automatic scrolling.

The lists intended for automatic change of displays and the lists for manual change of displays can differ. It is only important that there is at least one list for automatic and at least one for manual change of displays.

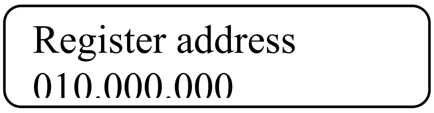
The same principle of use of the ROLL key is valid for both modes of display (Figure 3-20):



"x", "y", "z" in "n" can vary !
"m" for Autoscroll and Manualscroll can be different !

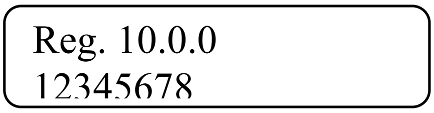
Figure 3-20 ROLL key - »Autoscroll« and »Manualscroll« display

The **Register** mode of display writes out the value of selected register. You activate the mode by selecting the Register option from the Display menu. Then you should enter the address of register, confirm the selection and enter the numbers of register address by the ROLL key. Each pressing on the ROLL key increases the number by one. All nine numbers should be entered! You can move to the next place by the SET key.
Entry of the register address:



Register address
010.000.000

Skipping one or more places while entering the numbers can do correction of the entry. From the place, where you have made a mistake, you press only the SET key until you reach the end of entry. Because the address is not complete, the cursor will move to the beginning of entry. After the entry of the last number, you get the display of register contents by pressing the SET key:



Reg. 10.0.0
12345678

Now, you can view the contents of individual registers in this group by pressing the ROLL key. Each pressing on the ROLL key, which is longer than two seconds, automatically increases the number of register. Automatic increase of the field number is not possible.
View of the next group of registers is possible only by entering new register address.

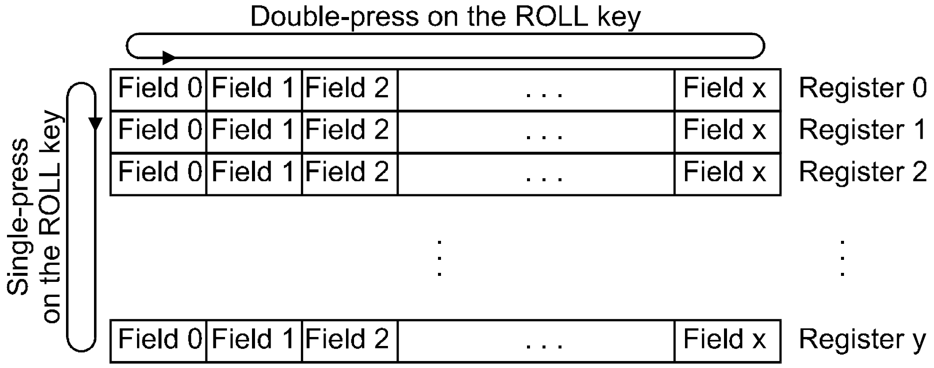


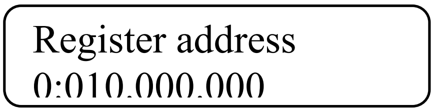
Figure 3-21 ROLL key - »Register« display

Example:

Single-press on the ROLL key:	Double-press on the ROLL key:
10.0.0	10.0.0
10.1.0	10.0.1
10.2.0	10.0.2
...	...
10.x.0	10.0.x

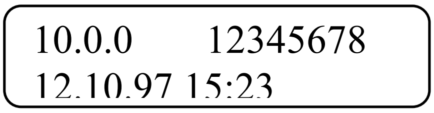
You can return to the main menu by pressing the MENU key.

The **Profile Register** mode of display writes out the value of selected register, which is stored in a profile. You activate the mode by selecting the Profile Register option from the Display menu. First you enter the number of profile where the data are stored and is intended for reading. Then you should enter the address of register, confirm the selection and enter the numbers of register address by the ROLL key. Each pressing on the ROLL key increases the number by one. All ten numbers should be entered! You can move to the next place by the SET key.
Entry of the register address:



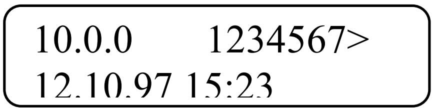
```
Register address
0:010.000.000
```

Skipping one or more places while entering the numbers can do correction of the profile address. From the place, where you have made a mistake, you press only the SET key until you reach the end of entry. Now press the SET key again. Because the address is not complete, the cursor will move to the beginning of entry. After the entry of the last number, you get the register address, contents of the register and the time stamp.



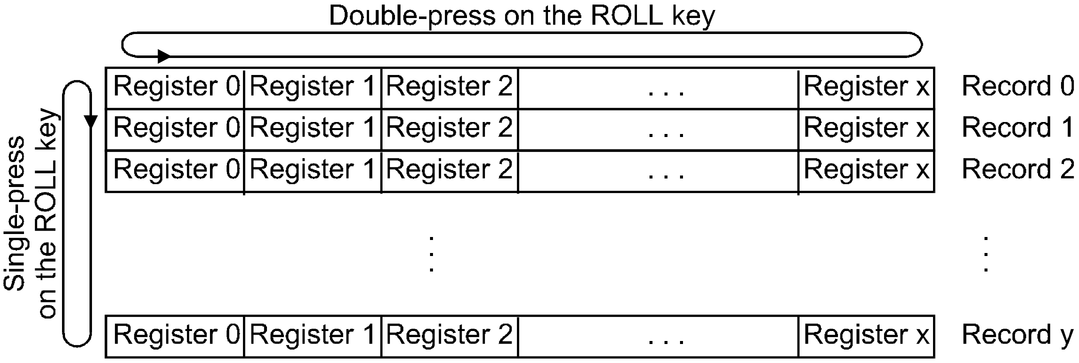
```
10.0.0      12345678
12.10.97 15:23
```

If the display of register contents has more than 8 characters, the »>« sign appears instead of the last character. In this case, display of the entire content is not possible.



```
10.0.0      1234567>
12.10.97 15:23
```

Now, you can view the contents of a profile by pressing the ROLL key. Each pressing on the ROLL key means moving through the profile backward by one record, which is reflected in changes of the time stamp. The number of these movements depends on the number of records in a profile. Each double-press on the ROLL key increases the number of register. Pressing on the ROLL key, longer than two seconds, means automatic moving through records. Automatic increase of the register number and the change of field number is not possible.
View of the next profile is possible only by entering new profile address.



Register 0	Register 1	Register 2	. . .	Register x	Record 0
Register 0	Register 1	Register 2	. . .	Register x	Record 1
Register 0	Register 1	Register 2	. . .	Register x	Record 2
		⋮			⋮
Register 0	Register 1	Register 2	. . .	Register x	Record y

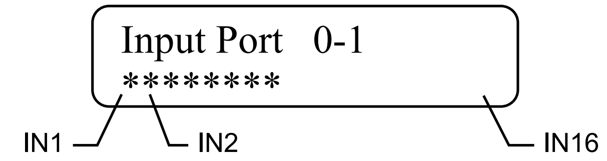
Figure 3-22 ROLL Key - Display "Profile Register"

Example:

Single-press on the ROLL key:	Double-press on the ROLL key:
10.0.0 11.5.97 11:00:00	10.0.0 11.5.97 10:00:00
10.0.0 11.5.97 10:45:00	10.1.0 11.5.97 10:00:00
10.0.0 11.5.97 10:30:00	10.2.0 11.5.97 10:00:00
...	...
10.0.0 x.x.xx x:x:x	10.x.0 11.5.97 10:00:00

You can return to the main menu by pressing the MENU key.

Input ports operation mode displays the status of input ports by 2 ports (2×8 inputs) simultaneously. It is intended for testing successful connection to input lines. This operation mode is selected with Input ports option in the Display menu.
Listing of input ports status:



```
Input Port   0-1
********
```

Significance of the characters on a display:

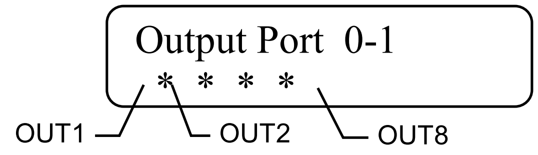
Character	Reception of impulse	
	input for active current S1a	input for inactive current S1r
*	YES	NO
_	NO	YES
no character	input inactive	input inactive

Table 3-4 Display: Input ports

If an input is inactive it means that for this input the parameter “input type” is set to NULL (0) in a group of registers IPORTPARAM.

When MENU key is pressed, a main menu is restored.

Output ports operation mode displays the status of output lines (8 outputs) at once. It is intended for testing successful connection to output lines. This operation mode is selected with Output ports option in the Display menu.
Listing of output ports status:



```
Output Port  0-1
 *  *  *  *
```

Significance of the characters on a display:

Character	Impulse transmission
*	YES
_	NO
no character	output inactive

Table 3-5 Display: Output ports

If an output is inactive it means that for this output the parameter “output type” is set to NULL (0) in a group of registers OPORTPARAM.

When MENU key is pressed, a main menu is restored.

The **MCU Status** mode of display shows the status of memory card. In case the card is not installed, the following is written out:

```
MCU:     No Card
```

An example of the write-out when the memory card is new and has never been formatted:

```
MCU      No CIS
```

If the memory card is installed, the status shows the type of memory card, its capacity, whether it is formatted or not (only a formatted card ca be used for entering of data) and the status data like Write Protect (WP) and Low Battery (BAT).
An example of display when the card is installed:

```
MCU: FLASH 1024K
Format OK
```

An example of display where the card is protected against writing (formatting is not possible):

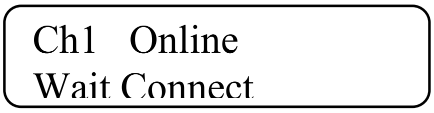
```
MCU: FLASH 1024K
Format OK     WP
```

An example of display when the battery of memory card has lost its voltage (formatting depends on the state of battery).

```
MCU: FLASH 1024K
Format OK BAT
```

You can return to the main menu by pressing the MENU key.

Channel Status display operation mode displays a state of communication lines and their phases.



```
Ch1   Online
Wait Connect
```

Communication lines are marked **Opt**. (optical communication), **Ch1** (communication channel 1) and **Ch2** (communication channel 2). Selection can be made with ROLL key.
The following messages are available:

Line state	Phase	Explanation
Idle	**Init**	Line is in rest position
Modem Detect	 **Init** **Wait** **User Init**	Check of modem presence • modem initialisation • waiting for modem response • initialisation of modem with a command, defined in a group of registers ATPARAM
Dial	 **Init** **Command** **Wait** **Connect** **Connect Delay**	Link establishment via a modem • setting to a line • call • waiting for modem response • establishment of modem link • stabilisation of conditions on a line
Answer	 **Init** **Wait**	Response to external call • setting to a line • waiting for establishment of modem link
Callback	 **Init** **Delay** **Escape** **Wait** **On Hook**	Establishment of link via a modem after the external call if CALLBACK function is active • setting to a line • stabilisation of conditions on a line • ESCAPE sequence for link interruption is sent by a modem • waiting for a response • interruption of a line
Hangup	 **Init** **Delay** **Escape** **Wait** **On Hook**	Interruption of a line • initialisation of line interruption • obvious delay before ESCAPE sequence • ESCAPE sequence sent by a modem • waiting for a response • line interruption
Online	 **Init** **Dial Init** **Answer Init** **Wait Connect** **Connected**	A link is established (for modem and RS232 or RS485) • establishment of link in a direct mode • establishment of connection after modem call • establishment of link after modem response • waiting for CO command • CO command carried out

Table 3-6 Channel status - Messages

When MENU key is pressed, a main menu is restored.

Meas. Periods
State of all three measuring periods expressed as percentage is displayed.

```
 MP1  MP2  MP3
100% 100% 100%
```

When MENU key is pressed, a main menu is restored.

Identification
When this option is selected, information on the device is displayed: type, firmware version and firmware checksum with the indication of validity. If checksum is correct, OK is recorded for indication of validity, otherwise it is not recorded.

```
POREG2   VX.XX
Checksum:FFFF Ok
```

Checksum is arithmetic sum of all memory locations except the last two where pre-defined checksum is recorded.
When MENU key is pressed, a main menu is restored.

Protection
State of protection with a switch is displayed.

```
Hardlock: Locked
```

When MENU key is pressed, a main menu is restored.

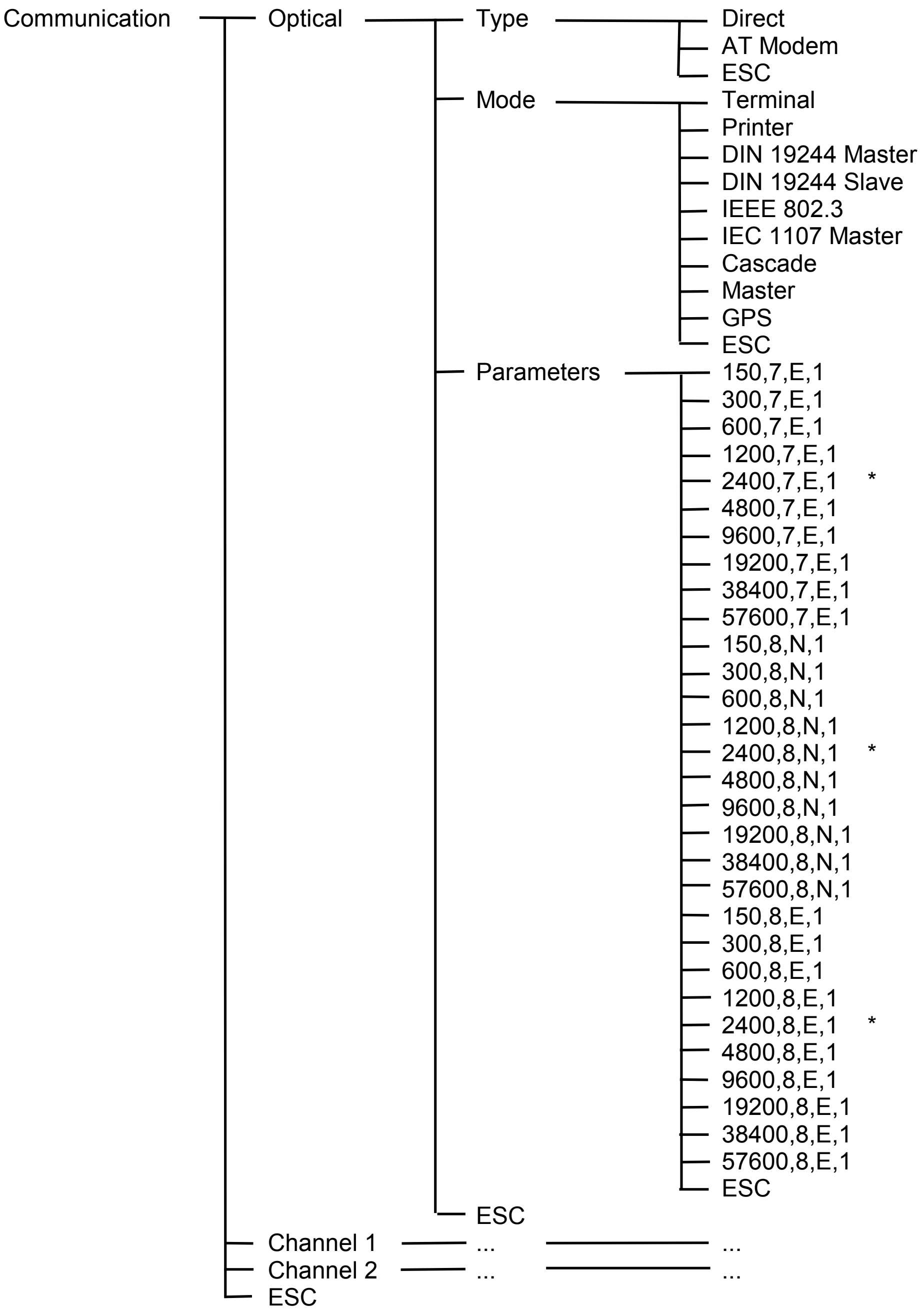
Test Pattern
When this option is chosen, all matrix pixels for all display characters are illuminated on a display.

```
€€€€€€€€€€€€€€€€
€€€€€€€€€€€€€€€€
```

When MENU key is pressed, a main menu is restored.

Communication menu

Communications menu is intended for setting parameters of communication channels. For each communication channel the type of connection, the operation mode and parameters of serial communication can be set. Different communication functions can be selected for the operation mode while standard parameters can be selected for parameters of serial communication (baud rate, a number of data bits, parity, and a number of stop bits).
The **Optical** channel is reserved for optical communication while other channels are intended for RS232 and RS485 connections.
For a PSTN modem **AT modem** communication type is selected, and for others a **direct** type. Other parameters are set regarding the channel. Functions Master and GPS are not realized yet!



* - recommended setting for optical communication

Messages menu

Message menu is designed for processing the system messages recorded in the lists for confirming and removing something. The “Acknowledge” option on the confirmation list means “confirm”; the “Clear” option on the removal list means “remove”.

Messages
- Acknowledge
- Clear
- Display
- ESC

The messages on both lists are displayed in the following way:

[Message Class]: [Primary Code]/[Secondary Code]

Warning, **Alarm** or **Error** may refer to “Message Class”. “Primary Code” and “Secondary Code ” correspond to the code record in the EVENT group of registers.

An example of conveying the message of the processor cold boot:

```
Warning: 01/02
```

By pressing the ROLL key it is possible for you to scroll the screen display down. By pressing the SET key you may acknowledge or clear all the messages that you have read by the end of surveying. Simultaneous reading and acknowledging (clearing) the messages are also possible by pressing the SET key only.
The return to the Messages Menu is automatic after the last message has been either acknowledged or cleared from the list.
An iterruption of either confirming or removing the messages is also possible by pressing the MENU key.

If there are no messages on the list, you can read the warning:

```
No Messages
```

By pressing any key you may return to the Messages Menu.

The "Display" option is designed for switching on / off the displaying of messages:

```
Messages
Display
```

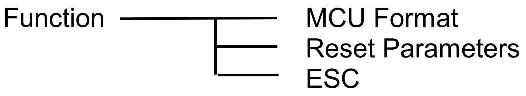
Messages display is switched off by first switching on or resetting the device.

```
Display
Off
```

Messages display is switched on / off by pressing the ROLL key. If the device is protected from parameter setting by using a switch, it is not possible to alter the setting.

Function menu

Function menu is intended for activation of special processing functions, like parameter reset and formatting of memory card.



MCU Format function

```
Function
MCU Format
```

This function is used for formatting the memory card. The data can be written on the memory card only when it is formatted. Formatting of the memory card is done according to the PCMIA standard. The file structure of the card is the DOS FAT file system. For entering of data, the POREG2.MCF file is created on the card. Measuring data in the form of profiles are written in this file. Beside all this, the card contains also information on data recorder identification, which enables the identification of card relations. Saving of data on the memory card belonging to the data recorder where it has been formatted is not possible.
Pressing on the SET key starts formatting:

```
MCU Format
Formatting...
```

If the card is not inserted or is inserted badly, this warning is displayed:

```
MCU Format
Not inserted
```

After successful formatting, the following message appears on the screen:

```
MCU Format
OK - ESC
```

If an error occurs during formatting or if there is something wrong with the card, you get the message:

```
MCU Format
Failed - ESC
```

Repeated formatting is possible by pressing on the SET key.

After each card formatting, the **Profile Reload** function starts. The task of this function is to preserve the data in profiles of the memory card. However, it works only if there is another profile defined in the data recorder, which stores the same information as the profile of the memory card. In this case, the function transfers the data covering a one-day time period.
The **Profile Reload** function starts whenever an already formatted card is installed.

Reset Parameters function
This function resets the functioning parameters of data recorder to the initial values.

```
Function
Reset Parameters
```

The function starts by pressing on the SET key.

```
Reset Parameters
Resetting...
```

If the device is protected from parameter setting, it is also protected from resetting parameters. Thus, the following message is displayed:

```
Reset Parameters
Failed - ESC
```

The consequences of resetting are:

- the device once again gets its manufacturer’s code 00000000 (DEVICEID group of registers)
- all users lose their rights of access; the access is possible only by manufacturer’s settings (USERS group of registers)
- measuring periods are set to: MP1 = 900, MP2 = 300 and MP3 = 60 (MEASPER group of registers)
- switching from winter time to summer time is carried out every year on the first Sunday following March 24 (DAYLIGHT group of registers)
- switching from summer time to winter time is carried out every year on the first Sunday following October 24 (DAYLIGHT group of registers)
- external time synchronization is switched off (TIMESYNCHRO group of registers)
- impulses delay function is switched off (IMPULSBARRIERE group of registers)
- all impuls inputs are of the S1a type; the shortest duration of an import impulse is 40 ms, and the longest impulse duration is not checked (IPORTPARAM group of registers)
- all output lines are of the impulse type; the length of output impulse signal is 80 ms and the duration of interrupt is 120 ms; the connections to the LEV9PARAM group of registers are arranged one after another (OPORTPARAM group of registers)
- a counter type for inputs from 0 to 15 is set to 1(U_COUNTER_B), and for inputs from 16 to 31 to 0 (NULL); correction factors for all results of the measuring level 0 re set on 1; the inputs from 0 do 15 are of the IMPULSE type; others are of the NULL type; connection of results with input lines are arranged in order (LEV0PARAM group of registers)
- a counter type is set to 0 (NULL); all sum definitions are erased (LEV1PARAM group of registers)
- a counter type is set to 0 (NULL); all function definitions are erased; the hysteresis functions are switched off (LEVPARAM group of registers)
- a counter type is set to 0 (NULL); all definitions for connection of results with outputs are erased (LEV9PARAM group of registers)
- billing processing exists no more: definitions of banks are erased (BANKPARAM group of registers)
- all communication channels are direct, of the terminal type, transfer speed is 2400 baud’s, the number of data bits is 8, parity is »none« and the number of »stop« bits is 1 (CHANNELPARAM group of registers)
- parameters for modem functioning (ATMODEM group of registers) are:

- initialization of modem every 60 seconds without additional settings
- automatic response on the second ringing during the whole day
- tonal selection
- the largest number of unsuccessful calls is 3 (callback)

- all serial meter inputs are switched off (SERIALPARAM group of registers)
- reading of meters according to the IEC1107 standard is switched off (IEC1107MREADPARAM group of registers)

- possibility of meter resetting is switched off (IEC1107MRESETPARAM group of registers)
- all parameters for a call via a modem are set to 0 (zero) (a group of registers CALLPARAM)
- cascading of devices is switched off (a group of registers CASCADE)
- tariff program is erased (TARDAYPROG, TARWEEKPROG, TARSEASPROG, TARHLPROG and TARPROG groups of registers)
- all alarms are erased (ALARMPAR group of registers)
- all definitions of format displays are erased (FORMSTR, FORMOUT, FORMLST, FORMCMD groups of registers)
- saving into profiles exists no more (PROFILEPAR group of registers)
- displaying the system messages is switched off (Messages menu)

3.12 SYSTEM MESSAGES

Data recorder constantly controls proper operation of the system, memory contents of the program, the data and the device management. When an error occurs, a system message is displayed (if displaying the messages is switched on) and recorded in registers.

System messages are divided into 4 classes:

- MESSAGE
- WARNING
- ALARM
- ERROR

There are messages without indicating any consequences in the MESSAGE class of messages. These messages cannot activate the alarm output and are recorded in the EVENT group of registers.
In the WARNING class of messages there are messages indicating that the device is not operating properly. The corresponding bits are held in the WARNING register of the SYSTEMATTR group of registers.
In the ALARM class of messages there are warnings that there is something wrong with the device. The device needs a repair, but not servicing. The measuring results are invalid. The corresponding bits are held in the ALARM register of the SYSTEMATTR group of registers.
In the ERROR class of messages there are warnings of critical system errors. The device needs servicing! The corresponding bits are held in the ERROR register of the SYSTEMATTR group of registers.

When a message out of the WARNING, ALARM and ERROR classes is displayed, then

- a suitable bit in the suitable register of the SYSTEMATTR group of registers is held;
- a system message is recorded in events of the EVENTS group of registers;
- a system message is recorded on the confirmation list;
- a suitable message is displayed (if displaying the messages is switched on) ;
- a suitable alarm output is activated if it is defined.

Data recorder has two internal lists of system messages (a list for acknowledging them and a list for clearing them). All new system messages are recorded on the confirmation list. They remain on the list up to the time of acknowledging them, when they are transferred to the list for clearing them. If a system message from both lists is equal to a new one, the new system message is not recorded on the confirmation list.
A system message may be cleared by pressing any key; the mode of displaying is restored. Clearing the message has no impact on either acknowledging or clearing the system messages!
An alarm output may be any impulse output connected to the class of messages and / or to the “Warning”, “Alarm” or “Error” registers of the SYSTEMATTR group of registers. Each value being different from nought in these registers activates a corresponding output. It remains active up to the time of acknowledging all the system messages from an adequate class of messages.

The messages are acknowledged and then cleared by carrying out the commands from the Messages Menu. If the reason for a system message still exists, the procedure is repeated when a system message appears.

4. GROUPS OF REGISTERS

Each group of registers consists of registers, and the registers consist of the fields which are basic units of the data saving in the data recorder. The fields can be of different types (numerical and text types) where values of individual data or references to other data are recorded.
A group of registers has its own characteristical data with its structure, permission for the access to data, description of individual fields, etc. Characteristical data of the group of registers are:

• identification number of a group	identification of register groups (GID)
• group attributes	the rights for the access to data and some characteristics (ATTR)
• number of registers in a group	(RPG)
• number of bytes in one register	(BPR)
• description of fields in a register	6 fields descriptions of fields (FD0 … FD5) (field type + field length, number of fields for that type)

Besides characteristical data of a group the data on a number of fields in a register is also added (FPR).

Individual groups of registers are described by their successive numbers.

A group of registers is represented as follows:

4.X NAME OF GROUP

Short description of a group.

GID =
ATTR =
RPG =
BPR =
FD0 ... FD3 = <basic field type > + <number of bytes in the field>, <number of fields>
FPR =

Tabular display:

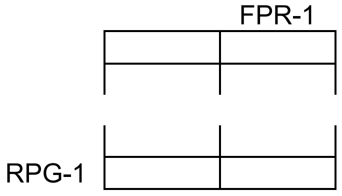


Table 4.X Group of registers - Name

Contents of fields:

Field	Contents	Type	Length

A group of registers in a data recorder:

Group	GID	Contents of data	Page
SYSTEMATTR	0	System attributes	97
DEVICEID	1	Recorder identification code	100
USERS	2	List of users	101
DATETIME	3	System date and time	102
TIMEATTR	4	Time functions attributes	103
PERFORM	5	Performance measurements	105
MEASPER	6	Measuring periods	106
DAYLIGHT	7	Parameters for shifting from summer to winter time	107
EVENTS	8	Recorder internal events	108
TIMESYNCHRO	9	Parameters of external synchronization	111
LEV0RESULT	10	Measuring results of level 0	112
LEV1RESULT	11	Measuring results of level 1	113
LEV2RESULT	12	Measuring results of level 2	114
LEV9RESULT	19	Measuring results of level 9	115
BANKRESULT	20	Billing results - current billing period	116
BANKRESULTL	21	Billing results - terminated billing period	117
BANKSTAMP	22	Time of billing results - current billing period	118
BANKSTAMPL	23	Time of billing results - terminated billing period	119
ATATTR	81	Attributes of modem communication	120
SERIALRESULT	82	Results of serial meter inputs	121
CALLATTR	87	Attributes of calls	123
TARSTAT	89	Tariff status	124
TARDAYRES	90	Results of daily tariff rules	125
TARWEEKRES	91	Results of weekly tariff rules	126
TARSEASRES	92	Results of seasonal tariff rules	127
TARHOLRES	93	Results of rules for holidays	128
TARRES	94	Results of tariff programs	129
ALARMATTR	95	Attributes of time alarms	130
IPORTATTR	101	State of input impulse lines	131
OPORTATTR	102	State of output impulse lines	132
IMPULSEBARRIERE	106	Parameters of impulse delay function	133
IPORTPARAM	108	Parameters of inputs	134
OPORTPARAM	109	Parameters of outputs	135
LEV0PARAM	110	Parameters of measuring results of level 0	136
LEV1PARAM	111	Parameters of measuring results of level 1	138
LEV2PARAM	112	Parameters of measuring results of level 2	140
LEV9PARAM	119	Parameters of measuring results of level 9	142
BANKPARAM	120	Parameters of billing results	144
CHANNELPARAM	180	Parameters of serial channels	146
ATPARAM	181	AT Modem parameters	147
SERIALPARAM	182	Parameters of serial meter inputs	148
CALLSTATION	185	Parameters of stations	149
TELNUMPARAM	186	Station telephone numbers	150
CALLPARAM	187	Parameters for call via a modem	151
CASCADE	189	Parameters of Cascade communication function	153
TARDAYPROG	190	Parameters of daily tariff rules	154
TARWEEKPROG	191	Parameters of weekly tariff rules	155
TARSESPROG	192	Parameters of seasonal tariff rules	156
TARHOLPROG	193	Parameters of rules for holidays	158
TARPROG	194	Parameters of tariff programs	159
ALARMPARAM	195	Parameters of time alarms	160
UNITS	200	Measuring units	162
FORMSTR	201	Format strings	163

Group	GID	Contents of data	Page
FORMOUT	202	Format listings	164
FORMLST	203	Format lists	165
FORMCMD	204	Format commands	166
PROFILEPAR	210	Parameters of saving into profiles	167
PROFILEATTR	211	Attributes of saving into profiles	168
PROFILESTAT	212	Status information of saving into profiles	169
POLLATTR	213	Information on POOL	170

Table 4-1 Groups of registers

4.1 SYSTEMATTR

The group contains the system attributes of device.

GID = 0
ATTR = R, W, RP, WP, WL
RPG = 32
BPR = 4
FD0 = UBINARY + 4, 1
FPR = 1

Tabular display:

	0
0	Status
1	Error
2	Alarm
3	Warning
4	PwrFailCount
5	PwrFailStamp
6	PwrFailDur
7	PwrOnCount
8	PwrOnStamp
9	PwrOnDur
10	RFailCount
11	RFailStamp
12	RFailDur
13	ROnCount
14	ROnStamp
15	ROnDur
16	SFailCount
17	SFailStamp
18	SFailDur
19	SOnCount
20	SOnStamp
21	SOnDur
22	TFailCount
23	TFailStamp
24	TFailDur
25	TOnCount
26	TOnStamp
27	TOnDur
28	WatchDogCount
29	WatchDogStamp
30	RestartCount
31	RestartStamp

Table 4-2 Group of registers **Napaka! Slog ni definiran.**

Content of fields:

Field	Contents	Type	Length
Status	System status of device; value is entered in form of a bit map. Basic value is 0, when all bits are set on 0. Bits from 0 to 5 have the following values: **Bit Value Meaning** • 0 1 failure of phase R • 1 2 failure of phase S • 2 4 failure of phase T • 3 8 device unlocked (parametration allowed) • 4 16 device opened • 5 32 power supply failure at processor Value in this field changes dynamically. Several simultaneous different events means summing up of individual bit values.	UBINARY	4
Error	Critical system error; value is entered in form of a bit map. Basic value is 0, which means that the device functions correctly. Bit 0 has the following value: • 1 result of the last calculation of security code (Checksum) does not equal the entered value Value in this field preserves itself.	UBINARY	4
Alarm	System alarm (a bit map of field PC in group of registers EVENTS)	UBINARY	4
Warning	System warning (a bit map of field PC in group of registers EVENTS)	UBINARY	4
PwrFailCount	Serial number of power supply failure at processor	UBINARY	4
PwrFailStamp	Time stamp of the last power failure at processor written as the number of seconds elapsed from 1.1.1970 at 00:00	UBINARY	4
PwrFailDur	Total time(s) of all power supply failures at processor	UBINARY	4
PwrOnCount	Serial number of power supply activation at processor	UBINARY	4
PwrOnStamp	Time stamp of the last power supply activation at processor written as the number of seconds elapsed from 1.1.1970 at 00:00	UBINARY	4
PwrOnDur	Total time of (s) of power supply at processor	UBINARY	4
RFailCount	Serial number of power supply failure in phase R	UBINARY	4
RFailStamp	Time stamp of the last power supply failure in phase R written as the number of seconds elapsed from 1.1.1970 at 00:00	UBINARY	4
RFailDur	Duration (s) of the last power supply failure in phase R	UBINARY	4
ROnCount	Serial number of power supply activation in phase R	UBINARY	4
ROnStamp	Time stamp of the last power supply activation in phase R written as the number of seconds elapsed from 1.1.1970 at 00:00	UBINARY	4
ROnDur	Duration (s) of power supply in phase R	UBINARY	4
SFailCount	Serial number of power supply failure in phase S	UBINARY	4
SFailStamp	Time stamp of the last power supply failure in phase S written as the number of seconds elapsed from 1.1.1970 at 00:00	UBINARY	4
SFailDur	Duration (s) of the last power supply failure in phase S	UBINARY	4
SOnCount	Serial number of power supply activation in phase S	UBINARY	4
SOnStamp	Time stamp of the last power supply activation in phase S written as the number of seconds elapsed from 1.1.1970 at 00:00	UBINARY	4
SOnDur	Duration (s) of power supply in phase S	UBINARY	4
TFailCount	Serial number of power supply failure in phase T	UBINARY	4

Field	Contents	Type	Length
TFailStamp	Time stamp of the last power supply failure in phase T written as the number of seconds elapsed from 1.1.1970 at 00:00	UBINARY	4
TFailDur	Duration (s) of the last power supply failure in phase T	UBINARY	4
TOnCount	Serial number of power supply activation in phase T	UBINARY	4
TOnStamp	Time stamp of the last power supply activation in phase T written as the number of seconds elapsed from 1.1.1970 at 00:00	UBINARY	4
TOnDur	Duration (s) of power supply in phase R	UBINARY	4
WatchDogCount	Serial number of processor reset activation	UBINARY	4
WatchDogStamp	Time stamp of the last processor reset activation written as the number of seconds elapsed from 1.1.1970 at 00:00	UBINARY	4
RestartCount	Serial number of starting the processor	UBINARY	4
RestartStamp	Time stamp of the last starting of processor written as the number of seconds elapsed from 1.1.1970 at 00:00	UBINARY	4

4.2 DEVICEID

A group has identification string of the device.

GID = 1
ATTR = R, W, RP, WP, WL, PAR, EE
RPG = 1
BPR = 9
FD0 = ASCII + 9, 1
FPR = 1

Tabular display:

	0
0	ident

Table 4-3 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
ident	8 numbers long identification string of the device	ASCII	9

4.3 USERS

The group contains the list of users with their passwords and rights.

GID = 2
ATTR = R, W, RP, WP, RL, WL, PAR, EE
RPG = 8
BPR = 20
FD0 = ASCII + 9, 1
FD1 = ASCII + 9, 1
FD2 = UBINARY + 2, 1
FPR = 3

Tabular display:

	0	1	2
0	code	password	Rights
7	code	password	Rights

Table 4-4 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
Code	User's code consisting of 8 characters	ASCII	9
Password	User's password consisting of 8 characters	ASCII	9
Rights	Rights of access to registers, written in a form of bit map	UBINARY	2

The rights determine to which data the access is allowed for each user separately:

bit	value	data protected by attribute	
0	1	R	reading allowed
1	2	W	writing allowed
2	4	RP	data reading protected by password
3	8	WP	data entering protected by password
4	16	RL	data reading protected by switch
5	32	WL	data entering protected by switch

Minimum rights, required by a group of registers, are needed for access, but they can be also bigger. If the »rights« parameter value equals zero (0), only reading of time and date is possible. Combination of rights is possible by summing up the values of individual rights.

Registers represent a list of users:

register	**meaning**
0	user 1
1	user 2
2	user 3
3	user 4
4	user 5
5	user 6
6	user 7
7	user 8

4.4 DATETIME

A group includes data on current date and time.

GID = 3
ATTR = R
RPG = 1
BPR = 9
FD0 = UBINARY + 2, 1
FD1 = UBINARY + 1, 6
FD2 = UBINARY + 2, 1
FPR = 8

Tabular display:

	0	1	2	3	4	5	6	7
0	year	month	day	wday	hour	minute	second	daylight

Table 4-5 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
year	Year (1990 - 2089)	UBINARY	2
month	Month (1 - 12)	UBINARY	1
day	Day (1 - 31)	UBINARY	1
wday	Day in a week (0 - 6 ; 0 = Sunday)	UBINARY	1
hour	Hour (0 - 23)	UBINARY	1
minute	Minute (0 - 59)	UBINARY	1
second	Second (0 - 59)	UBINARY	1
daylight	Summer time (0 = standard, 1 = summer time)	UBINARY	2

4.5 TIMEATTR

A group consists of time attributes like times of the termination of measuring periods, termination of a day, a month, a year and synchronization, duration of measuring periods, a year, a month, a day and synchronization as well as successive numbers of termination.

GID = 4
ATTR = R
RPG = 7
BPR = 12
FD0 = UBINARY + 4, 3
FPR = 3

Tabular display:

	0	1	2
0	TMP1	DMP1	SNMP1
1	TMP2	DMP2	SNMP2
2	TMP3	DMP3	SNMP3
3	TD	DD	SND
4	TM	DM	SNM
5	TY	DY	SNY
6	S	DS	SNS

Table 4-6 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
TMP1	Termination of measuring period 1 is time stamp of the last termination of measuring period 1. Time stamp is written as a number of seconds from 1 January 1970 at 00:00.	UBINARY	4
DMP1	Duration of current measuring period 1 in seconds	UBINARY	4
SNMP1	Successive number of the last termination of measuring period 1	UBINARY	4
TMP2	Termination of measuring period 2 is time stamp of the last termination of measuring period 2. Time stamp is written as a number of seconds from 1 January 1970 at 00:00.	UBINARY	4
DMP2	Duration of current measuring period 2 in seconds	UBINARY	4
SNMP2	Successive number of the last termination of measuring period 2	UBINARY	4
TMP3	Termination of measuring period 3 is time stamp of the last termination of measuring period 3. Time stamp is written as a number of seconds from 1 January 1970 at 00:00.	UBINARY	4
DMP3	Duration of current measuring period 3 in seconds	UBINARY	4
SNMP3	Successive number of the last termination of measuring period 3	UBINARY	4
TD	Termination of day is time stamp of the last termination of day	UBINARY	4
DD	Duration of a day in seconds	UBINARY	4
SND	Successive number of the last termination of day	UBINARY	4
TM	Termination of month is time stamp of the last termination of month	UBINARY	4
DM	Duration of a month in seconds	UBINARY	4
SNM	Successive number of the last termination of month	UBINARY	4
TY	Termination of year is time stamp of the last termination of year	UBINARY	4
DY	Duration of a year in seconds	UBINARY	4
SNY	Successive number of the last termination of year	UBINARY	4
S	Synchronization is a time stamp of the last synchronization	UBINARY	4
DS	Time from the last synchronization in seconds	UBINARY	4
SNS	Successive number of the last synchronization	UBINARY	4

4.6 PERFORM

A group includes measurements of efficiency of main processes in the data recorder. Each measurement consists of the last measurement and maximal measurement.

GID = 5
ATTR = R, W, WP, WL
RPG = 11
BPR = 12
FD0 = UBINARY + 4, 2
FPR = 2

Tabular display:

	0	1
0	LM	MM
10	LM	MM

Table 4-7 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
LM	Number of msec used by the process in the last performance	UBINARY	4
MM	Maximal number of msec used by the process in one performance	UBINARY	4

Registers represent basic processes which are measured. These are:

register	**process**
0	alarm generation
1	results calculation
2	saving into profiles
3	calculation of tariff results
4	automatic format listings
5	processing of console
6	processing of serial inputs
7	processing of memory card
8	processing of input ports
9	system processes
10	sum of all processes

4.7 MEASPER

A group includes data on measuring periods.

GID = 6
ATTR = R, W, RP, WP, WL, PAR, EE
RPG = 1
BPR = 12
FD0 = UBINARY + 4, 3
FPR = 3

Tabular display:

	0	1	2
0	mp1	mp2	mp3

Table 4-8 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
mp1	Measuring period 1 in seconds	UBINARY	4
mp2	Measuring period 2 in seconds	UBINARY	4
mp3	Measuring period 3 in seconds	UBINARY	4

4.8 DAYLIGHT CHANGE PARAMETERS

In the group are two registers; the first one for the beginning of summer time, and the second one for the beginning of winter (standard) time.

GID = 7
ATTR = R, W, RP, WP, PAR, EE
RPG = 2
BPR = 6
FD0 = UBINARY + 2, 1
FD1 = UBINARY + 1, 4
FPR = 5

Tabular display: :

	0	1	2	3	4
0	year	month	day	wday	alg
1	year	month	day	wday	alg

Table 4-9 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
year	Year (1990 - 2089)	UBINARY	2
month	Month (1 - 12)	UBINARY	1
day	Day (1 - 31)	UBINARY	1
wday	Day in a week (0 - 6 ; 0 = Sunday)	UBINARY	1
alg	Algorithm of repetition (0 - 9)	UBINARY	1

Repetition algorithms for date are:

0	once	repeated once exactly on day, month and year
1	yearly	repeated each year on day and month
2	yearlywd	repeated each year on "a day in a week "after "a day" and "month"
3	yearlyns	repeated each year on a day and month; if Sunday, it is moved to Monday
4	yearlyaeast	repeated each year for X days after Easter (X = parameter "day" = 0- 255)
5	yearlybeast	repeated each year for X days before Easter (X = parameter "day" = 0 - 255)
6	monthly	repeated each month on a certain day
7	monthlywd	repeated each month on a day in a week after a certain day
8	weekly	repeated each week on a day in a week
9	daily	repeated each day

4.9 EVENTS

A group includes the records of internal events in the data recorder. During the operation the data recorder registers important events which influence in correct operation. These data are stored in a ring buffer. Each record of event includes data on what happened and when the event occurred.

GID = 8
ATTR = R
RPG = 255
BPR = 10
FD0 = UBINARY + 1, 2
FD1 = UBINARY + 4, 2
FPR = 4

Tabular display:

	0	1	2	3
0	PC	SC	TE	CE
254	PC	SC	TE	CE

Table 4-10 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
PC	Primary code is primary identification of the event or event field. Possible fields are: 1 - initializing 2 - system self test error 3 - power supply error 4 - battery error 5 - data overrun 6 - data loss 7 - time messages 8 - module error 9 - impulse error on input 10 - impulse capture error on serial input 11 - external messages 12 - impulse overrun on output 13 - meter compare error 14 - register overrun 15 - parameter change 17 - manual entering 18 - warning 19 - disturbance	UBINARY	1
SC	Secondary code is secondary identification of the event and defines the event more in detail in certain area of events	UBINARY	1
TE	Time of event is time stamp of the event and is recorded as a number of seconds from 1 January 1970 at 00:00	UBINARY	4
CE	Contents of the event is additional information on the event which depends on individual area of events.	UBINARY	4

A list of secondary codes in primary codes:

Primary codes	Secondary codes
Initializing	0 - program restart
	1 - CPU warm start
	2 - CPU cold start
	17 - memory card module warm start
	18 - memory card module cold start
	33 - printer module warm start
	34 - printer module cold start
	49 - communication module warm start
	50 - communication module cold start
System self test error	1 - CPU program memory
	2 - CPU parameter memory
	3 - CPU data memory
	17 - memory card module program memory
	18 - memory card module parameter memory
	19 - memory card module data memory
	33 - printer module program memory
	34 - printer module parameter memory
	35 - printer module data memory
	49 - communication module program memory
	50 - communication module parameter memory
Power supply error	1 - CPU power supply shutdown
	2 - CPU power supply disturbance
	3 - CPU power supply 1 defect
	4 - CPU power supply 2 defect
	8 - Power supply phase R OFF - IE Extension
	9 - Power supply phase S OFF - IE Extension
	10 - Power supply phase T OFF - IE Extension
	11 - Power supply phase R ON - IE Extension
	12 - Power supply phase S ON - IE Extension
	13 - Power supply phase T ON - IE Extension
Battery error	1 - CPU module battery error
	17 - memory card module battery error
	33 - printer CPU module battery error
	49 - communication module battery error
	65 - memory card module battery error
Data overrun	1 - CPU data overrun MP1
	2 - CPU data overrun MP2
	3 - CPU data overrun MP3
	10 - CPU data overrun messages
	17 - memory card module data overrun MP1
	18 - memory card module data overrun MP2
	19 - memory card module data overrun MP3
	26 - memory card module data overrun messages
	32 - printer module data overrun
	49 - communication module data overrun MP1
	50 - communication module data overrun MP2
	51 - communication module data overrun MP3
	58 - communication module data overrun messages

Primary codes	Secondary codes
Data loss	1 - CPU data loss MP1 2 - CPU data loss MP2 3 - CPU data loss MP3 10 - CPU data loss messages 17 - memory card module data loss MP1 18 - memory card module data loss MP2 19 - memory card module data loss MP3 26 - memory card module data loss messages 32 - printer module data loss 49 - communication module data loss MP1 50 - communication module data loss MP2 51 - communication module data loss MP3 58 - communication module data loss messages
Time messages	1 - DCF reception disturbed 2 - time synchronization disturbed 3 - time daylight change to summer time 4 - time daylight change to standard time 5 - manual change of time 6 - system time/DCF time difference to high 7 - system time/RTC time difference to high 8 - DCF reception disturbed for 24 hours
Module error	17 - memory card module disturbed 18 - memory card not present 19 - memory card defect 20 - memory card invalid type 21 - memory card invalid size 33 - printer module disturbed 34 - printer module no paper 35 - printer module timeout 36 - printer module switched out 48 - communication module disturbed
Impulse error on input	0 ... 127 - number of input
Impulse error on serial input	0 ... 127 - number of serial input
External message	0 ... 127 - number of control input
Impulse overrun on output	0 ... 127 - number of output
Meter compare error	0 ... 127 - number of compare pair
Register overrun	0 .. 127 - number of register
Parameter change	1 - CPU parameter change 16 - memory card module parameter change 32 - printer module parameter change 48 - communication module parameter change

4.10 TIMESYNCHRO

A group contains parameters of external time synchronization. With these parameters a function of control input is added to optional impulse input.

GID = 9
ATTR = R, W, RP, WP, PAR, EE
RPG = 1
BPR = 4
FD0 = UBINARY + 1, 4
FPR = 4

Tabular display :

	0	1	2	3
0	TYP	PORT	DP	WS

Table 4-11 Group of registers **Napaka! Slog ni definiran.**

Contents of fields :

Field	Contents	Type	Length
TYP	A type defines if synchronization is connected to a port or not: 0 NONE 1 PORT	UBINARY	1
PORT	A port defines a number of impulse input which is used for synchronization.	UBINARY	1
PP	Permitted period (in minutes) defines time interval with which time synchronization is permitted (0 - 240)	UBINARY	1
WS	Synchronization window (in seconds) defines maximal deviation of synchronization moment from a complete minute, so that synchronization is still performed (0 - 30).	UBINARY	1

Synchronization is performed periodically as it is defined by a parameter "permitted period".
A value "synchronization window" indicates that recorder current time is fast or slow regarding the synchronization. It is synchronized at a complete minute:

- forward, if synchronization occurs from 0 to 30 seconds before a complete minute
- backward, if synchronization occurs from 0 to 29 seconds after a complete minute

Setting of "synchronization window" parameter to 30 indicates a window -30/+29 seconds!

4.11 LEV0RESULT

Registers of results of level 0 represent meter results which are captured on impulse inputs or via serial meter inputs.
Parameters are set in a group of LEV0PARAM registers.

GID = 10
ATTR = R, W, WP, WL
RPG = 32
BPR = 58
FD0 = SBINARY + 4, 10
FPR = 10

Tabular display:

	0	1	2	3	4	5	6	7	8	9
0	cmp1	cmp2	cmp3	pmp1	pmp2	pmp3	mmp1	mmp2	mmp3	crun
31	cmp1	cmp2	cmp3	pmp1	pmp2	pmp3	mmp1	mmp2	mmp3	crun

Table 4-12 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
cmp1	Result of current measuring period 1	SBINARY	4
cmp2	Result of current measuring period 2	SBINARY	4
cmp3	Result of current measuring period 3	SBINARY	4
pmp1	Result of previous measuring period 1	SBINARY	4
pmp2	Result of previous measuring period 2	SBINARY	4
pmp3	Result of previous measuring period 3	SBINARY	4
mmp1	Cumulative result of measuring period 1	SBINARY	4
mmp2	Cumulative result of measuring period 2	SBINARY	4
mmp3	Cumulative result of measuring period 3	SBINARY	4
crun	Current cumulative result	SBINARY	4

Results of current measuring periods and cumulative current result are currently updated. At termination of a measuring period the result of a current measuring period is transmitted to the result of a previous measuring period. It is then added up to the cumulative result of a measuring period. A current measuring period result is then deleted.

4.12 LEV1RESULT

Result registers of level 1 represent meter results which are calculated from the results of level 0 and are positive sums of the results of level 0.
Parameters are set in a LEV1PARAM group of registers.

GID = 11
ATTR = R, W, WP, WL
RPG = 16
BPR = 56
FD0 = SBINARY + 4, 10
FPR = 10

Tabular display:

	0	1	2	3	4	5	6	7	8	9
0	cmp1	cmp2	cmp3	pmp1	pmp2	pmp3	mmp1	mmp2	mmp3	crun
15	cmp1	cmp2	cmp3	pmp1	pmp2	pmp3	mmp1	mmp2	mmp3	crun

Table 4-13 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
cmp1	Result of current measuring period 1	SBINARY	4
cmp2	Result of current measuring period 2	SBINARY	4
cmp3	Result of current measuring period 3	SBINARY	4
pmp1	Result of previous measuring period 1	SBINARY	4
pmp2	Result of previous measuring period 2	SBINARY	4
pmp3	Result of previous measuring period 3	SBINARY	4
mmp1	Cumulative result of measuring period 1	SBINARY	4
mmp2	Cumulative result of measuring period 2	SBINARY	4
mmp3	Cumulative result od measuring period 3	SBINARY	4
crun	Current cumulative result	SBINARY	4

Results of current measuring periods and cumulative current result are currently updated. At termination of a measuring period the result of a current measuring period is transmitted to the result of a previous measuring period. It is then added up to the cumulative result of the measuring period. A current measuring period result is then deleted.

4.13 LEV2RESULT

Result registers of level 2 represent meter results which are calculated from the results of levels 0 and 1.
Parameters are set in a group of registers LEV2PARAM.

GID = 12
ATTR = R, W, WP, WL
RPG = 8
BPR = 68
FD0 = SBINARY + 4, 10
FPR = 10

Tabular display:

	0	1	2	3	4	5	6	7	8	9
0	cmp1	cmp2	cmp3	pmp1	pmp2	pmp3	mmp1	mmp2	mmp3	crun
7	cmp1	cmp2	cmp3	pmp1	pmp2	pmp3	mmp1	mmp2	mmp3	crun

Table 4-14 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
cmp1	Result of current measuring period 1	SBINARY	4
cmp2	Result of current measuring period 2	SBINARY	4
cmp3	Result of current measuring period 3	SBINARY	4
pmp1	Result of previous measuring period 1	SBINARY	4
pmp2	Result of previous measuring period 2	SBINARY	4
pmp3	Result of previous measuring period 3	SBINARY	4
mmp1	Cumulative result of measuring period 1	SBINARY	4
mmp2	Cumulative result of measuring period 2	SBINARY	4
mmp3	Cumulative result of measuring period 3	SBINARY	4
crun	Current cumulative result	SBINARY	4

Results of current measuring periods and cumulative current result are currently updated. At termination of a measuring period the result of a current measuring period is transmitted to the result of a previous measuring period. It is then added up to the cumulative result of the measuring period. A current measuring period is then deleted.

4.14 LEV9RESULT

A group includes the results of level 9.
Parameters are set in a group of registers LEV9PARAM.

GID = 19
ATTR = R, W, WP, WL
RPG = 8
BPR = 14
FD0 = SBINARY + 4, 2
FPR = 2

Tabular display:

	0	1
0	cro	cum
7	cro	cum

Table 4-15 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
cro	Current result for output	SBINARY	4
cum	Cumulative result	SBINARY	4

Current results of level 9 are specific as they are being increased regarding the results of lower levels, and decreased when output impulses are being sent through output lines.

4.15 BANKRESULT

A group of registers includes billing results for a current measuring period. They are organized in banks; each register represents one bank. In the fields are billing values which are updated at termination of measuring periods.
A corresponding tariff (internal program or external tariff control) should be active which enables mathematical operation be performed.
When a measuring period is terminated, the results are transmitted into the results of the last terminated billing period.
Parameters are set in a group of registers BANKPARAM.

GID = 20
ATTR = R, W, WP, WL
RPG = 32
BPR = 32
FD0 = SBINARY + 4, 8
FPR = 8

Tabular display:

	0	1	2	3	4	5	6	7
0	value0	value1	value2	value3	value4	value5	value6	value7
31	value0	value1	value2	value3	value4	value5	value6	value7

Table 4-16 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
value0	Value of current billing result 0	SBINARY	4
value1	Value of current billing result 1	SBINARY	4
value2	Value of current billing result 2	SBINARY	4
value3	Value of current billing result 3	SBINARY	4
value4	Value of current billing result 4	SBINARY	4
value5	Value of current billing result 5	SBINARY	4
value6	Value of current billing result 6	SBINARY	4
value7	Value of current billing result 7	SBINARY	4

If the value in the field exceeds $\pm 2^{31}$, the value is set to 0 (zero)!

4.16 BANKRESULTL

Billing results of the last terminated billing period are organized by banks where each register represents one bank. In the fields are billing values which are updated at the end of a billing period. Billing values are values which were obtained on the basis of current billing results.

GID = 21
ATR = R, W, WP, WL, EE
RPG = 32
BPR = 32
FD0 = SBINARY + 4, 8
FPR = 8

Tabular display:

	0	1	2	3	4	5	6	7
0	value0	value1	value2	value3	value4	value5	value6	value7
31	value0	value1	value2	value3	value4	value5	value6	value7

Table 4-17 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
value0	Value of billing result 0 for the last terminated billing period	SBINARY	4
value1	Value of billing result 1 for the last terminated billing period	SBINARY	4
value2	Value of billing result 2 for the last terminated billing period	SBINARY	4
value3	Value of billing result 3 for the last terminated billing period	SBINARY	4
value4	Value of billing result 4 for the last terminated billing period	SBINARY	4
value5	Value of billing result 5 for the last terminated billing period	SBINARY	4
value6	Value of billing result 6 for the last terminated billing period	SBINARY	4
value7	Value of billing result 7 for the last terminated billing period	SBINARY	4

If the value in the field exceeds $\pm 2^{31}$, the value is set to 0 (zero)!

4.17 BANKSTAMP

When a billing result is changed, the time of a change is always recorded in a form of time stamp. It has a value of a number of elapsed seconds from 1 January 1970 up to the change.

GID = 22
ATTR = R, W, WP, WL
RPG = 32
BPR = 32
FD0 = UBINARY + 4, 8
FPR = 8

Tabular display:

	0	1	2	3	4	5	6	7
0	stamp0	stamp1	stamp2	stamp3	stamp4	stamp5	stamp6	stamp7
31	stamp0	stamp1	stamp2	stamp3	stamp4	stamp5	stamp6	stamp7

Table 4-18 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
stamp0	Time stamp 0 for current billing result	UBINARY	4
stamp1	Time stamp 1 for current billing result	UBINARY	4
stamp2	Time stamp 2 for current billing result	UBINARY	4
stamp3	Time stamp 3 for current billing result	UBINARY	4
stamp4	Time stamp 4 for current billing result	UBINARY	4
stamp5	Time stamp 5 for current billing result	UBINARY	4
stamp6	Time stamp 6 for current billing result	UBINARY	4
stamp7	Time stamp 7 for current billing result	UBINARY	4

4.18 BANKSTAMPL

When current billing results are transmitted into billing results of the last terminated billing period, time stamp is transmitted at the same time. Time stamp has the value of a number of elapsed seconds from 1 January 1970 up to the change.

GID = 23
ATTR = R, W, WP, WL, EE
RPG = 32
BPR = 32
FD0 = UBINARY + 4, 8
FPR = 8

Tabular display:

	0	1	2	3	4	5	6	7
0	stamp0	stamp1	stamp2	stamp3	stamp4	stamp5	stamp6	stamp7
31	stamp0	stamp1	stamp2	stamp3	stamp4	stamp5	stamp6	stamp7

Table 4-19 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
stamp0	Time stamp 0 for billing result of the last terminated billing period	UBINARY	4
stamp1	Time stamp 1 for billing result of the last terminated billing period	UBINARY	4
stamp2	Time stamp 2 for billing result of the last terminated billing period	UBINARY	4
stamp3	Time stamp 3 for billing result of the last terminated billing period	UBINARY	4
stamp4	Time stamp 4 for billing result of the last terminated billing period	UBINARY	4
stamp5	Time stamp 5 for billing result of the last terminated billing period	UBINARY	4
stamp6	Time stamp 6 for billing result of the last terminated billing period	UBINARY	4
stamp7	Time stamp 7 for billing result of the last terminated billing period	UBINARY	4

4.19 ATATTR

A group involves information on the state of individual communication lines, but only those which are defined as AT Modem (a group of registers CHANNELPARAM).

GID = 81
ATTR = R
RPG = 3
BPR = 16
FD0 = UBINARY + 4, 3
FD1 = UBINARY + 2, 2
FPR = 5

Tabular display:

	0	1	2	3	4
0	LI	LD	RT	RC	DC
1	LI	LD	RT	RC	DC
2	LI	LD	RT	RC	DC

Table 4-20 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
LI	Time of last initialisation of AT modem expressed as a number of elapsed seconds from 1.1.1970 from 00:00	UBINARY	4
LD	Time of the last detection of AT modem expressed as a number of elapsed seconds from 1.1.1970 from 00:00	UBINARY	4
RT	A number of milliseconds from the last ring. (max. 10000)	UBINARY	4
RC	A number of ringings indicates a successive number of ringing which is compared with a number of ringings necessary for a response (RTA field in a group of registers ATPARAM). If within 10 seconds there is no ringing, a number of ringings is set to 0.	UBINARY	2
DC	A number of ringings indicates a successive number of call which is compared with a number of repeated calls in case of unsuccessful call (RC field in a group of registers ATPARAM). A number of calls is set to 0 in case of successful call or when a number of calls reaches a number of repeated calls in case of unsuccessful call (RC).	UBINARY	2

Individual registers represent:

register	**meaning**
00	events - communication channel 0 - OPTICAL
01	events - communication channel (RS232C / RS485) 1 - CHANNEL 1
02	events - communication channel (RS232C / RS485 or MODEM) 2 - CHANNEL 2

4.20 SERIALRESULT

The group contains the results of serial meter inputs.
Parameters are set in the SERIALPARAM group of registers.

GID = 82
ATTR = R
RPG = 32
BPR = 14
FD0 = UBINARY + 4, 1
FD1 = SBINARY + 4, 1
FD2 = UBINARY + 1, 2
FPR = 4

Tabular display:

	0	1	2	3
0	STAMP	LVALUE	STATUS	SEQ
31	STAMP	LVALUE	STATUS	SEQ

Table 4-21 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
STAMP	Time stamp of the last data transmission. It is recorded as a number of elapsed seconds from 1/1/1970 from 00:00.	UBINARY	4
LVALUE	The last transmitted value. At the end of data transmission a current value is calculated and then the last value is copied with a momentary read value.	SBINARY	4
STATUS	Status information on the last reading.	UBINARY	1
SEQ	Number of the meter readings that have been done so far. The number is read out from the meter. Possible values are from 0 to 31.	UBINARY	1

The STATUS field may include the following values:
0 - data are correct
1 - IV
2 - CA
3 - CA and IV
4 - CY
5 - CY and IV
6 - CY and CA
7 - CY and CA and IV

Legend:
IV - Invalid - invalid data
CA - Counter Adjusted - manual adjustment of the meter counter
CY - Carry - meter counter has turned

Individual registers represent:

register	meaning
00	result of serial input 0
01	result of serial input 1
02	result of serial input 2
03	result of serial input 3
.	
.	
.	
30	result of serial input 30
31	result of serial input 31

4.21 CALLATTR

The group contains call attributes.
Parameters are set in the CALLPARAM group of registers.

GID = 87
ATTR = R
RPG = 8
BPR = 8
FD0 = UBINARY + 1, 2
FD1 = UBINARY + 4, 1
FD2 = UBINARY + 1, 2
FPR = 5

Tabular display:

	0	1	2	3	4
0	ST	FL	LS	LC	LCI
7	ST	FL	LS	LC	LCI

Table 4-22 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
ST	State of call 0 - IDLE 1 - PENDING	UBINARY	1
FL	State of condition for a call 0 - NULL no condition 1 - ALARM alarm is a condition	UBINARY	1
LS	Time of the last call during elapsed seconds from 1.1.1970 from 00:00	UBINARY	4
LC	A number of condition “cause” 0 - NULL 1 - ALARM 2 - IEC1107DATA	UBINARY	1
LCI	Index of condition “cause” • A system message from the ALARM class of messages, if the cause was “ALARM” • A channel through which data from the meter were read-out if the cause was “IEC1107DATA”	UBINARY	1

4.22 TARSTAT

The group contains status information on tariffs.

GID = 89
ATTR = R
RPG = 1
BPR = 2
FD0 = UBINARY + 2, 1
FPR = 1

Tabular display:

	0
0	STATUS

Table 4-23 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
STATUS	Status of individual tariff rules Value entered in the field is a bit map of tariff rule results from the TARRES group of registers. Status of individual bits reveals whether a tariff rule is active or not. A tariff rule is active if the bit value of tariff rule is 1	UBINARY	2

Bits have the following meanings:

bit	**meaning**
0	result status of tariff rule 0
1	result status of tariff rule 1
2	result status of tariff rule 2
3	result status of tariff rule 3
4	result status of tariff rule 4
5	result status of tariff rule 5
6	result status of tariff rule 6
7	result status of tariff rule 7
8	result status of tariff rule 8
9	result status of tariff rule 9
10	result status of tariff rule 10
11	result status of tariff rule 11
12	result status of tariff rule 12
13	result status of tariff rule 13
14	result status of tariff rule 14
15	result status of tariff rule 15

4.23 TARDAYRES

A group involves results of a daily tariff program. A daily tariff program consists of individual daily rules. Each rule divides a day to active and inactive part of a day. For each rule there is a result which indicates whether a current time is in active or inactive part of a day.
Parameters of daily rules are set in a group of registers TARDAYPROG.

GID = 90
ATTR = R
RPG = 32
BPR = 1
FD0 = UBINARY + 1, 1
FPR = 1

Tabular display:

	0
0	DPA0
31	DPA31

Table 4-24 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
DPA(0 - 31)	Status of daily rule result: 0 - inactive part of a day 1 - active part of a day	UBINARY	1

Individual registers represent:

register	**meaning**
00	result of a daily rule 0
01	result of a daily rule 1
02	result of a daily rule 2
03	result of a daily rule 3
.	
.	
.	
30	result of a daily rule 30
31	result of a daily rule 31

4.24 TARWEEKRES

A group involves results of a weekly tariff program. A weekly tariff program consists of individual weekly rules. Each weekly rule involves daily rules (reference for a daily rule) for each day in a week or a holiday. For each weekly rule there is a result which indicates for current time whether a daily rule for a current day in a week is active or not. If a current day is a holiday, a daily rule for a holiday is used. Parameters of weekly rules are set in a group of registers TARWEEKPROG.

GID = 91
ATTR = R
RPG = 16
BPR = 1
FD0 = UBINARY + 1, 1
FPR = 1

Tabular display:

	0
0	WPA0
15	WPA15

Table 4-25 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents		Type	Length
WPA(0 - 15)	Status of weekly rule result:	0 - inactive 1 - active	UBINARY	1

Individual registers represent:

register	**meaning**
00	result of a weekly rule 0
01	result of a weekly rule 1
02	result of a weekly rule 2
03	result of a weekly rule 3
04	result of a weekly rule 4
05	result of a weekly rule 5
06	result of a weekly rule 6
07	result of a weekly rule 7
08	result of a weekly rule 8
09	result of a weekly rule 9
10	result of a weekly rule 10
11	result of a weekly rule 11
12	result of a weekly rule 12
13	result of a weekly rule 13
14	result of a weekly rule 14
15	result of a weekly rule 15

4.25 TARSEASRES

A group involves results of a seasonal tariff program. A seasonal tariff program consists of individual seasonal rules. Each defines a part of a year in which a rule is active. For each rule there is a result which indicates whether current time is in active or inactive part of a year.
Parameters of seasonal rules are set in a group of registers TARSEASPROG.

GID = 92
ATTR = R
RPG = 16
BPR = 1
FD0 = UBINARY + 1, 1
FPR = 1

Tabular display:

	0
0	SPA0
15	SPA15

Table 4-26 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents		Type	Length
SPA(0 - 15)	Status of seasonal rule result:	0 - inactive part of a year 1 - active part of a year	UBINARY	1

Individual registers represent:

register	**meaning**
00	result of a seasonal rule 0
01	result of a seasonal rule 1
02	result of a seasonal rule 2
03	result of a seasonal rule 3
04	result of a seasonal rule 4
05	result of a seasonal rule 5
06	result of a seasonal rule 6
07	result of a seasonal rule 7
08	result of a seasonal rule 8
09	result of a seasonal rule 9
10	result of a seasonal rule 10
11	result of a seasonal rule 11
12	result of a seasonal rule 12
13	result of a seasonal rule 13
14	result of a seasonal rule 14
15	result of a seasonal rule 15

4.26 TARHOLRES

A group involves results of a tariff program for holidays. It consists of individual rules for a holiday. Each defines a day in a year as a holiday. For each rule there is a result which indicates whether a current day is a holiday or not.
Parameters of rules for holidays are set in a group of registers TARHOLPROG.

GID = 93
ATTR = R
RPG = 32
BPR = 1
FD0 = UBINARY + 1, 1
FPR = 1

Tabular display:

	0
0	HPA0
31	HPA31

Table 4-27 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
HPA(0 - 31)	Status of holiday rule result : 0 - not a holiday 1 - a holiday	UBINARY	1

Individual registers represent:

register	**meaning**
00	result of a holiday rule 0
01	result of a holiday rule 1
02	result of a holiday rule 2
03	result of a holiday rule 3
.	
.	
.	
30	result of a holiday rule 30
31	result of a holiday rule 31

4.27 TARRES

A group involves results of a tariff program. It consists of individual rules. A tariff rule uses seasonal and weekly tariff rules for definition of active status of a tariff result. In this way a year can be divided into several seasons. In each a different weekly tariff rule is valued. A tariff rule result is active if a weekly rule is active for at least one active seasonal rule.
Parameters are set in a group of registers TARPROG.

GID = 94
ATTR = R
RPG = 16
BPR = 1
FD0 = UBINARY + 1, 1
FPR = 1

Tabular display:

	0
0	TPA0
15	TPA15

Table 4-28 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents		Type	Length
TPA(0 - 15)	Status of tariff rule result :	0 - inactive 1 - active	UBINARY	1

Individual registers represent:

register	**meaning**
00	result of a tariff rule 0
01	result of a tariff rule 1
02	result of a tariff rule 2
03	result of a tariff rule 3
04	result of a tariff rule 4
05	result of a tariff rule 5
06	result of a tariff rule 6
07	result of a tariff rule 7
08	result of a tariff rule 8
09	result of a tariff rule 9
10	result of a tariff rule 10
11	result of a tariff rule 11
12	result of a tariff rule 12
13	result of a tariff rule 13
14	result of a tariff rule 14
15	result of a tariff rule 15

4.28 ALARMATTR

A group involves attributes of time alarms. They consist of time of the last alarm and a number of generated alarms.
Parameters are set in a group of registers ALARMPARAM.

GID = 95
ATTR = R
RPG = 16
BPR = 6
FD0 = UBINARY + 4, 1
FD1 = UBINARY + 2, 1
FPR = 2

Tabular display:

	0	1
0	TA	NA
15	TA	NA

Table 4-29 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
TA	Time of alarm is time stamp of the last alarm. It is recorded as a number a elapsed seconds from 1 January 1970 until 00:00.	UBINARY	4
NA	Successive number of the last alarm	UBINARY	2

4.29 IPORTATTR

The group contains the attributes of input impulse lines.

GID = 101
ATTR = R
RPG = 1
BPR = 2
FD0 = UBINARY + 1, 2
FPR = 2

Tabular display:

	0	1
0	IN_0-7	IN_8-15

Table 4-30 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
IN_0-7	Status of input impulse lines from 0 to 7 written in a decimal form of bit map	UBINARY	1
IN_8-15	Status of input impulse lines from 8 to 15 written in a decimal form of bit map	UBINARY	1

Bits have the following meanings:

Field	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
IN_0-7	IN 7	IN 6	IN 5	IN 4	IN 3	IN 2	IN 1	IN 0
IN_8-15	IN 15	IN 14	IN 13	IN 12	IN 11	IN 10	IN 9	IN 8

The status of bit is 1 if an impulse is present at the corresponding input. The field value is written as sum of values of all those bits with status 1. The bit values are:

bit	7	6	5	4	3	2	1	0
value	128	64	32	16	8	4	2	1

The bit map of a field can be compared with the display of input lines status (Menu Display, selection Input Ports).

4.30 OPORTATTR

The group contains the attributes of output lines.

GID = 102
ATTR = R
RPG = 1
BPR = 1
FD0 = UBINARY + 1, 1
FPR = 1

Tabular display:

	0
0	OUT_0-7

Table 4-31 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
OUT_0-7	Status of output impulse lines from 0 to 7 written in a decimal form of bit map	UBINARY	1

Bits have the following meanings:

Field	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
OUT _0-7	OUT 7	OUT 6	OUT 5	OUT 4	OUT 3	OUT 2	OUT 1	OUT 0

The status of bit is 1 if an impulse is generated at the corresponding output. The field value is written as sum of values of all those bits with status 1. The bit values are:

bit	7	6	5	4	3	2	1	0
value	128	64	32	16	8	4	2	1

The bit map of a field can be compared with the display of output lines status (Menu Display, selection Output Ports).

4.31 IMPULSEBARRIERE

The group contains the parameters of impulse windows. These parameters define the width of impulse limiting window before and after the termination of measuring period. They are defined for each measuring period separately.

GID = 106
ATTR = R, W, WP, WL, PAR, EE
RPG = 3
BPR = 4
FD0 = UBINARY + 2, 2
FPR = 2

Tabular display:

	0	1
0	BT	AT
1	BT	AT
2	BT	AT

Table 4-32 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
BT	Size of impulse restraining window before the end of measuring period in ms. Set from 0 to 1s at a pace of 1 ms.	UBINARY	2
AT	Size of impulse restraining window after the end of measuring period in ms. Set from 0 to 1s at a pace of 1 ms.	UBINARY	2

The registers represent:

register	**meaning**
00	window size for measuring period 1
01	window size for measuring period 2
22	window size for measuring period 3

4.32 IPORTPARAM

A group involves parameters of input lines. They are used to define a type of input lines and parameters for checking the operation. Input lines can generate two types of a result - a current status and impulses.

GID = 108
ATTR = R, W, WP, WL, PAR, EE
RPG = 16
BPR = 3
FD0 = UBINARY + 1, 3
FPR = 3

Tabular display :

	0	1	2
0	type of input	maximal length	minimal length
15	type of input	maximal length	minimal length

Table 4-33 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
type of input	The following values are available: 0 - NULL - input not used 1 - S1A - input is two-wire for active current 2 - S1R - input is two-wire for reactive current 3 - S2/S3 - input is three-wire	UBINARY	1
maximal length	Maximal impulse length expressed in minimal impulse lengths. If parameter is zero (0), maximal length is checked.	UBINARY	1
minimal length	Minimal impulse length expressed in ms	UBINARY	1

If input type is S2/S3, two successive inputs (e.g. even, odd) are used. Only definition of the first input in a pair is valid. The result is displayed only for the first input in a pair, as well.

Individual registers represent:

register	**meaning**
00	parameter for input 0
01	parameter for input 1
02	parameter for input 2
03	parameter for input 3
.	
.	
.	
14	parameter for input 14
15	parameter for input 15

4.33 OPORTPARAM

A group involves parameters of output lines. Output lines are used for impulse or status outputs. In the first case the output line functions as an impulse multiplier. It transmits impulses which are cumulated in a result for output impulses (LEV9RESULT) with maximal output frequency. In case of status output the output line follows the internal register (e.g. TARRES).

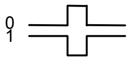
GID = 109
ATTR = R, W, WP, WL, PAR, EE
RPG = 8
BPR = 5
FD0 = UBINARY + 1, 5
FPR = 5

Tabular display :

	0	1	2	3	4
0	output type	status / op	signal / gid	pause / rid	link / fid
7	output type	status / op	signal / gid	pause / rid	link / fid

Table 4-34 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
output type	Output types are: 0 - NULL - output is not used 1 - IMPULSE - impulse output 2 - REGISTER - register tracing	UBINARY	1
status / op	For impulse outputs the initial status is defined. 0 -  1 - For status outputs a normal (0) or negated (1) operation is defined.	UBINARY	1
signal / gid	For impulse outputs duration of impulse in ms (1-255) is defined. For status outputs the address of a group of registers is defined.	UBINARY	1
pause / rid	For impulse outputs duration of impulse pause in ms (1-255) is defined. For status outputs the register address is defined.	UBINARY	1
link / fid	For impulse outputs a connection to the result of output impulses (reference) is defined. For status outputs the address of a register field is defined.	UBINARY	1

Individual registers represent:

register	meaning
00	parameter for output 0
01	parameter for output 1
.	
.	
.	
06	parameter for output 6
07	parameter for output 7

4.34 LEV0PARAM

The group contains the parameters of measuring results of the level 0. Measuring results of the level 0 represent the results of meter inputs collected at impulse inputs or through serial meter inputs. After collecting, correction of inputs is carried out and the results are stored into the registers of the level 0. The correction is carried out according to the following formula:

$$R0_i = \frac{a_i}{b_i} I_j$$

The parameters are **a**$_i$, **b**$_i$ and **I**$_j$, where correction factors are defined for each result and the connection with input. Index **i** is the number of register, and index **j** is the number of input. The factors **a** and **b** are calculated according to the following formula:

$$\frac{k_{IN}}{k_{REC}} = \frac{a}{b}$$

k$_{IN}$ [energy/impulse] is the constant of the meter and **k**$_{REC}$ [energy/impulse] is the constant of data recorder.

GID = 110
ATTR = R, W, WP, PAR, EE
RPG = 32
BPR = 8
FD0 = UBINARY + 2, 1
FD1 = SBINARY + 2, 2
FD2 = UBINARY + 1, 2
FPR = 5

Tabular display:

	0	1	2	3	4
0	type	correction a	correction b	type_input	index
31	tip	correction a	correction b	type_input	index

Table 4-35 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
Type	Parameter defining the type of result correction: 0 - NULL A measuring result is not changed 1 - U_COUNTER_B Only positive changes are permitted. The state can be changed from -2^{31} to $+2^{31}$. Overflow of maximal value changes the result state to a minimal value! 2 - U_PCOUNTER_B Only positive changes are permitted. The state can be changed only from 0 to $+2^{31}$. Overflow of maximal value changes the result state to a minimal value! 3 - UD_COUNTER_B Positive and negative changes are permitted. The state can be changed from -2^{31} to $+2^{31}$. Overflow of maximal value changes the result state to a minimal value and vice-versa!! 4 - UD_PCOUNTER_B Positive and negative changes are permitted. The state can be changed only from 0 to $+2^{31}$. Overflow of maximal value changes the result state to a	UBINARY	2

Field	Contents	Type	Length
	minimal value and vice-versa! 5 - U_COUNTER_D Only positive changes are permitted. The state can be changed from -99999999 to +99999999. Overflow of maximal value changes the result state to a minimal value! 6 - U_PCOUNTER_D Only positive changes are permitted. The state can be changed only from 0 to +99999999. Overflow of maximal value changes the result state to a minimal value! 7 - UD_COUNTER_D Positive and negative changes are permitted. The state can be changed from -99999999 to +99999999. Overflow of maximal value changes the result state to a minimal value and vice-versa!! 8 - UD_PCOUNTER_D Positive and negative changes are permitted. The state can be changed only from 0 to +99999999. Overflow of maximal value changes the result state to a minimal value and vice-versa!		
Correction a	Positive or negative number which represents factor a	SBINARY	2
Correction b	Positive or negative number which represents factor b	SBINARY	2
Type_input	Parameter defining the type of input: 0 - NULL result is not connected with the input 1 - IMPULSE impulse meter input 2 - SERIAL serial meter input	UBINARY	1
Index	Reference on physical input line. In case of impulse meter input it is tied to the IPORTPARAM group of registers and in case of serial meter input to the SERIALRESULT group of registers.	UBINARY	1

4.35 LEV1PARAM

A group involves parameters of measuring results of level 1. Results of level 1 are calculated from the results of level 0 and are positive sums of results of level 0. Calculation is performed by formula:

$$R1_i = a_{i,0}R0_0 + a_{i,1}R0_1 + ... + a_{i,n-1}R0_{n-1}$$

Parameters are $\mathbf{a_{i,0}}$, ..., $\mathbf{a_{i,n-1}}$, where for each result of level 0 it is defined whether it is added up to the result of level 1 or not. Index **i** expresses a register number while index **n** expresses a number of registers of level 0.

GID = 111
ATTR = R, W, WP, WL, PAR, EE
RPG = 16
BPR = 6
FD0 = UBINARY + 2, 1
FD1 = UBINARY + 1, 4
FPR = 5

Tabular display :

	0	1	2	3	4
0	type	SIGN0-7	SIGN8-15	SIGN16-23	SIGN24-31
15	type	SIGN0-7	SIGN8-15	SIGN16-23	SIGN24-31

Table 4-36 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
Type	Parameter defining the type of result correction: 0 - NULL A measuring result is not changed 1 - U_COUNTER_B Only positive changes are permitted. The state can be changed from -2^{31} to $+2^{31}$. Overflow of maximal value changes the result state to a minimal value! 2 - U_PCOUNTER_B Only positive changes are permitted. The state can be changed only from 0 to $+2^{31}$. Overflow of maximal value changes the result state to a minimal value! 3 - UD_COUNTER_B Positive and negative changes are permitted. The state can be changed from -2^{31} to $+2^{31}$. Overflow of maximal value changes the result state to a minimal value and vice-versa!! 4 - UD_PCOUNTER_B Positive and negative changes are permitted. The state can be changed only from 0 to $+2^{31}$. Overflow of maximal value changes the result state to a minimal value and vice-versa! 5 - U_COUNTER_D Only positive changes are permitted. The state can be changed from -99999999 to +99999999. Overflow of maximal value changes the result state to a minimal value! 6 - U_PCOUNTER_D Only positive changes are permitted. The state can be changed only from 0 to +99999999. Overflow of maximal value changes the result state to a minimal value!	UBINARY	2

Field	Contents	Type	Length
	7 - UD_COUNTER_D Positive and negative changes are permitted. The state can be changed from -99999999 to +99999999. Overflow of maximal value changes the result state to a minimal value and vice-versa!! 8 - UD_PCOUNTER_D Positive and negative changes are permitted. The state can be changed only from 0 to +99999999. Overflow of maximal value changes the result state to a minimal value and vice-versa!		
SIGN0-7	Signs for results of level 0 from 0 to 7. A parameter is organized with bits where LSB indicates summing up of the result 0 to the result of level 1, and MSB indicates summing up of the result 7 to the result of level 1.	UBINARY	1
SIGN8-15	Signs for results of level 0 from 8 to 15. A parameter is organized with bits where LSB indicates summing up of the result 8 to the result of level 1, and MSB indicates summing up of the result 15 to the result of level 1.	UBINARY	1
SIGN16-23	Signs for results of level 0 from 16 to 23. A parameter is organized with bits where LSB indicates summing up of the result 8 to the result of level 1, and MSB indicates summing up of the result 15 to the result of level 1.	UBINARY	1
SIGN24-31	Signs for results of level 0 from 24 to 31. A parameter is organized with bits where LSB indicates summing up of the result 8 to the result of level 1, and MSB indicates summing up of the result 15 to the result of level 1.	UBINARY	1

Example:

Summing up of results 0, 1, 2 and 3 to the result of level 1 is written as:

RESULT		7	6	5	4	3	2	1	0	
PARAMETER		$a_{7,0}$	$a_{6,0}$	$a_{5,0}$	$a_{4,0}$	$a_{3,0}$	$a_{2,0}$	$a_{1,0}$	$a_{0,0}$	
SIGN0-7	=	0	0	0	0	1	1	1	1	= 0Fh = **15**

4.36 LEV2PARAM

A group involves parameters of measuring results of level 2. Results of level 2 are calculated from the results of level 0 and/or 1. Functions are calculated as follows:

$$R2_i = \frac{a_i}{b_i} A_i + \frac{c_i}{d_i} B_i$$

Parameters are **a_i**, **b_i**, **c_i**, **d_i**, **A_i** and **B_i**, where correction factors and connection to the results on levels 0 and 1 are defined for each result. Index **i** expresses a register number.

GID = 112
ATTR = R, W, WP, WL, PAR, EE
RPG = 8
BPR = 18
FD0 = UBINARY + 2, 1
FD1 = SBINARY + 2, 2
FD2 = UBINARY + 1, 2
FD3 = SBINARY + 2, 2
FD4 = UBINARY + 1, 2
FD5 = UBINARY + 4, 1
FPR = 10

Tabular display:

	0	1	2	3	4	5	6	7	8	9
0	type	a	b	level A	result A	c	d	level B	result B	H
7	type	a	b	level A	result A	c	d	level B	result B	H

Table 4-37 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
Type	Parameter defining the type of result correction: 0 - NULL A measuring result is not changed 1 - U_COUNTER_B Only positive changes are permitted. The state can be changed from -2^{31} to $+2^{31}$. Overflow of maximal value changes the result state to a minimal value! 2 - U_PCOUNTER_B Only positive changes are permitted. The state can be changed only from 0 to $+2^{31}$. Overflow of maximal value changes the result state to a minimal value! 3 - UD_COUNTER_B Positive and negative changes are permitted. The state can be changed from -2^{31} to $+2^{31}$. Overflow of maximal value changes the result state to a minimal value and vice-versa!! 4 - UD_PCOUNTER_B Positive and negative changes are permitted. The state can be changed only from 0 to $+2^{31}$. Overflow of maximal value changes the result state to a minimal value and vice-versa! 5 - U_COUNTER_D Only positive changes are permitted. The state can be changed from -99999999 to +99999999. Overflow of maximal value changes the result state to a minimal value!	UBINARY	2

Field	Contents	Type	Length
	6 - U_PCOUNTER_D Only positive changes are permitted. The state can be changed only from 0 to +99999999. Overflow of maximal value changes the result state to a minimal value! 7 - UD_COUNTER_D Positive and negative changes are permitted. The state can be changed from -99999999 to +99999999. Overflow of maximal value changes the result state to a minimal value and vice-versa!! 8 - UD_PCOUNTER_D Positive and negative changes are permitted. The state can be changed only from 0 to +99999999. Overflow of maximal value changes the result state to a minimal value and vice-versa!		
a	Positive or negative number which represents factor a	SBINARY	2
b	Positive or negative number which represents factor b	SBINARY	2
level A	Reference to level of measuring results. 0 - level of measuring results 0 1 - level of measuring results 1	UBINARY	1
result A	Reference to measuring result which is considered at calculation	UBINARY	1
c	Positive or negative number which represents factor c	SBINARY	2
d	Positive or negative number which represents factor d	SBINARY	2
level B	Reference to level of measuring results. 0 - level of measuring results 0 1 - level of measuring results 1	UBINARY	1
result B	Reference to measuring result which is considered at calculation	UBINARY	1
H	Hysteresis Factory setting is 0! Changes are permitted only if in accordance with all partners involved!	UBINARY	4

4.37 LEV9PARAM

The group contains the parameters of measuring results of the level 9. Measuring results of the level 9 are the results of output impulses and are calculated according to the following formula:

$$R9_i = \frac{a_i}{b_i} A_i$$

The parameters are $\mathbf{a}_i$, $\mathbf{b}_i$ and $\mathbf{A}_i$, where correction factors are defined for each result and the connection with results of the levels 0. 1 or 2. Index **i** is the number of register.

GID = 119
ATTR = R, W, WP, WL, PAR, EE
RPG = 8
BPR = 8
FD0 = UBINARY + 2, 1
FD1 = SBINARY + 2, 2
FD2 = UBINARY + 1, 2
FPR = 5

Tabular display:

	0	1	2	3	4
0	type	correction a	correction b	level A	result A
7	type	correction a	correction b	level A	result A

Table 4-38 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
Type	Parameter defining the type of result correction: 0 - NULL A measuring result is not changed U_PCOUNTER_B Only positive changes are permitted. The state can be changed only from 0 to $+2^{31}$. Overflow of maximal value changes the result state to a minimal value!	UBINARY	1
Correction a	Positive or negative number which represents factor a	SBINARY	1
Correction b	Positive or negative number which represents factor b	SBINARY	1
Level A	Reference on the level of measuring results: 0 - measuring level 0 1 - measuring level 1 2 - measuring level 2	UBINARY	1
Result A	Reference on measuring result, which is taken into account in calculations.	UBINARY	1

Results of the level 9 are specific as they increase with regard to the results of lower levels and decrease when the output impulses are sent through output lines.

Individual registers represent:

register	meaning
00	parameter for output of measuring result 0
01	parameter for output of measuring result 1
02	parameter for output of measuring result 2
03	parameter for output of measuring result 3
04	parameter for output of measuring result 4
05	parameter for output of measuring result 5
06	parameter for output of measuring result 6
07	parameter for output of measuring result 7

4.38 BANKPARAM

A group involves parameters of billing results which are organized in banks by 8 results. For each bank of results it is defined which measuring results are calculated, what billing result they form, what is a billing period, which tariff program is used by a bank as well as which operations are performed at the end of a measuring period and a billing period.

GID = 120
ATTR = R, W, WP, WL, PAR, EE
RPG = 32
BPR = 17
FD0 = UBINARY + 1, 17
FPR = 17

Tabular display:

	0	1	2	3	4	5	6	7	8	9		16
0	TR	PRR	RR	TBP	IBP	OTBP	TT	TI	CL	RC0	...	RC7
31	TR	PRR	RR	TBP	IBP	OTBP	TT	TI	CL	RC0	...	RC7

Table 4-39 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
TR	Type of result:	UBINARY	1
	0 - NULL is not calculated		
	1 - Period value in a measuring period is calculated (MPV)		
	2 - Dif. period power in a measuring period is calculated (V = MPV × 3600 / MP)		
PRR	Period of result renewal:	UBINARY	1
	0 - is not renewed		
	1 - MP1 is renewed at the end of measuring period 1		
	2 - MP2 is renewed at the end of measuring period 2		
	3 - MP3 is renewed at the end of measuring period 3		
RR	Renewal of a result :	UBINARY	1
	0 - NULL is not renewed		
	1 - WRITE BV = V		
	2 - CUMULATE BV = BV + V		
	3 - MAX WRITE if V > BV BV = V		
	4 - MAX CUMULATE if V > BV BV = BV + V		
	5 - MAX2 WRITE		
	6 - MAX2 AVERAGE		
	7 - MAX3 WRITE		
	8 - MAX3 AVERAGE		
TBP	Type of billing period:	UBINARY	1
	0 - NULL billing period is not terminated		
	1 - ALARM repeatable time alarm terminates a billing period		
	2 - PORT impulse at input port terminates a billing period		
IBP	Index of a billing period represents index of a time alarm or input port which is used for termination of a billing period	UBINARY	1

Field	Contents	Type	Length
OTBP	Operation at termination of a billing period defines what operation is performed at transmission of current billing results (BV) to the results of the last terminated billing period (LBV). Available operations are: 0 - NULL no renewal 1 - WRITE LBV = BV 2 - CUMMULATE LBV = LBV + BV 3 - WRITE & CLEAR LBV = BV ; BV = 0 4 - CUMMULATE & CLEAR LBV = LBV + BV ; BV = 0	UBINARY	1
TT	Tariff type defines which internal tariff program or external input controls tariff for a bank. Tariff control indicates that current results are renewed only when tariff is active for a bank. The following types of tariff control are available: 0 - NULL no tariff control 1 - TARIFF internal tariff program 2 - PORT status at input port	UBINARY	1
TI	Tariff index indicates an index of internal tariff program or index of input port which is used for a tariff control	UBINARY	1
CL	Connection level is a reference for a level of measuring results and is common for all billing results of one bank: 0 - measuring level 0 1 - measuring level 1 2 - measuring level 2	UBINARY	1
RC0 ... RC7	Registers of connection are indexes of individual measuring results which are used for billing results in a bank. Values from fields pmp1, pmp2 or pmp3 are used as measuring results, depending on a period of result restoring.	UBINARY	1

Operations for restoring the result “MAX2 WRITE”, “MAX2 AVERAGE”, “MAX3 WRITE” or “MAX3 AVERAGE” refer to the operation for restoring the result “MAX WRITE” which is the first operation for calculation of maximum. Banks where these operations are defined should be recorded immediately after the bank where “MAX WRITE” operation is defined for restoring the result. All other parameters of banks should also be equal.
When these operations for restoring the result are used, only one bank with the operation of result restoring “MAX WRITE” may be used in a group! The sequence of banks with these operations of result restoring is not important.

4.39 CHANNELPARAM

A group involves parameters of communication channels. Parameters of communication channels consist of type of connection, functions which is being performed by a communication channel and of parameters of serial communication like bit rate, a number of data bits, parity and a number of stop bits.

GID = 180
ATTR = R, W, WP, PAR, EE
RPG = 3
BPR = 8
FD0 = UBINARY + 1, 2
FD1 = UBINARY + 2, 1
FD2 = UBINARY + 1, 3
FPR = 6

Tabular display:

	0	1	2	3	4	5
0	T	F	BR	DB	P	SB
1	T	F	BR	DB	P	SB
2	T	F	BR	DB	P	SB

Table 4-40 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
T	Type of connection 0 - Direct 1 - AT Modem	UBINARY	1
F	Function which is performed by a comm. channel 0 - Terminal 1 - Printer 2 - DIN 19244 Master 3 - DIN 19244 Slave 4 - reserved for IEEE 802.3 5 - reserved for IEC 1107 Master 6 - Cascade 7 - reserved for Master 8 - reserved for GPS	UBINARY	1
BR	Baud rate is a bit rate of data transmission along the line. Maximal rate is 57600 baud's. Other standard rates are: 150, 300, 600, 1200, 2400, 4800, 9600, 19200 and 38400.	UBINARY	2
DB	Data bits indicates a number of data bits. Possible values are 7 and 8.	UBINARY	1
P	Parity for data transmission checking. The following is available: 78 = N - none 69 = E - even 79 = O - odd	UBINARY	1
SB	Indicates a number of stop bits. Values 1 or 2 are available.	UBINARY	1

Individual registers represent:

register	**meaning**
00	communication channel 0 - OPTICAL
01	communication channel (RS232C / RS485) 1 - CHANNEL 1
02	communication channel (RS232C / RS485 or MODEM) 2 - CHANNEL 2

4.40 ATPARAM

A group involves parameters which define the modem operation. Parameters are common for all connected modems on communication channels which are defined as AT Modem (T field in a group of registers CHANNELPARAM).

GID = 181
ATTR = R, W, WP, PAR, EE
RPG = 1
BPR = 52
FD0 = ASCII + 40, 1
FD1 = UBINARY + 4, 1
FD2 = UBINARY + 1, 8
FPR = 10

Tabular display:

	0	1	2	3	4	5	6	7	8	9
0	IS	IP	AT	RTA	ASTRH	ASTRM	ASTPH	ASTPM	PD	RC

Table 4-41 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
IS	Initialisation string, max. 40 characters long	ASCII	40
IP	Initialisation period in seconds defines initialisation time interval	UBINARY	4
AT	Response time to call 0 - NULL 1 - AUTO 2 - CALLBACK	UBINARY	1
RTA	Number of ring signals necessary for response	UBINARY	1
ASTRH	Start of response time window - hour (0 - 23)	UBINARY	1
ASTRM	Time of response time window - minute (0 - 59)	UBINARY	1
ASTPH	End of response time window - hour (0 - 24)!	UBINARY	1
ASTPM	End of response time window - minute (0 - 59)	UBINARY	1
PD	Call type 0 - TONE 1 - PULSE	UBINARY	1
RC	Number of repeated calls indicates how many times link set-up will be attempted by a device	UBINARY	1

At termination of initialisation period the recorder sends its initialisation string. If a modem sends OK message, the recorder sends initialisation string which is defined in this register. Initialisation string, defined in the recorder, is:

```
ATE0 V1 Q0 &C0 &D0 &K0 S0=0
```

Warning:
For correct operation of the modem it is recommended not to change the modem settings in the internal initialisation string with the user initialisation string!

4.41 SERIALPARAM

A group consists of parameters of serial meter inputs. Parameters of serial meter inputs consist of a type of input, used communication channel, address of connected meter, address of read register and alarm at which reading is performed.

GID = 182
ATTR = R, W, WP, PAR, EE
RPG = 32
BPR = 6
FD0 = UBINARY + 1, 2
FD1 = UBINARY + 2, 1
FD2 = UBINARY + 1, 2
FPR = 5

Tabular display:

	0	1	2	3	4
0	CHT	CH	LA	RA	AL
31	CHT	CH	LA	RA	AL

Table 4-42 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
CHT	A type of input: 0 - INACTIVE 1 - SDT	UBINARY	1
CH	A channel defines a communication channel through which communication with a meter is performed. 0 - OPTICAL 1 - CHANNEL 1 2 - CHANNEL 2 Selected channel should be correspondingly set.	UBINARY	1
LA	A link address defines a meter address. If only one meter is connected, its address can be 0; if several meters are connected (max. 31), the address is between 1 and 31.	UBINARY	2
RA	A register address defines which data will be read from a meter. Details are available in the meter documentation.	UBINARY	1
AL	An alarm defines when meter reading is performed. It is connected with alarms defined in a group of registers ALARMPARAM.	UBINARY	1

Individual register represents:

register	**meaning**
00	parameters of serial meter input 0
01	parameters of serial meter input 1
.	
.	
.	
30	parameters of serial meter input 30
31	parameters of serial meter input 31

4.42 CALLSTATION

A group involves parameters of called stations or devices.

GID = 185
ATTR = R, W, WP, PAR, EE
RPG = 8
BPR = 23
FD0 = UBINARY + 1, 1
FD1 = ASCII + 10, 1
FD2 = ASCII + 10, 1
FD3 = UBINARY + 1, 2
FPR = 5

Tabular display:

	0	1	2	3	4
0	ST	SID	SP	CT	CI
7	ST	SID	SP	CT	CI

Table 4-43 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
ST	Type of link: 0 - NONE no called station 1 - PRIMARY called station is for a level higher 2 - SECONDARY called station is for a level lower	UBINARY	1
SID	Identification number of called station which is max. 9 characters long	ASCII	10
SP	A password of called station which is max. 9 characters long	ASCII	10
CT	Call type: 0 - NONE call is not performed 1 - DIAL call is performed	UBINARY	1
CI	Call index is index of telephone number (0 - 7) from a group of registers TELNUMPARAM	UBINARY	1

Registers represent a list of stations which can be called by the station. If in a group of registers ATPARAM the “AT” parameter is set to “CALLBACK”, this list is a list of authorised stations. This is the additional security parameter against unauthorised calls via a modem.

4.43 TELNUMPARAM

A group involves a list of telephone numbers.

GID = 186
ATTR = R, W, WP, PAR, EE
RPG = 8
BPR = 20
FD0 = ASCII + 20, 1
FPR = 1

Tabular display:

	0
0	TN
7	TN

Table 4-44 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
TN	Telephone number of called station which is max. 20 characters long	ASCII	20

4.44 CALLPARAM

A group involves parameters necessary for the device for calling via a modem. Parameters consist of a call type, a number of communication channel which the modem is connected to, an index of called device and an event at which the call occurs as well as on the cause for calling.

GID = 187
ATTR = R, W, WP, PAR, EE
RPG = 8
BPR = 7
FD0 = UBINARY + 1, 7
FPR = 7

Tabular display:

	0	1	2	3	4	5	6
0	CLT	CLC	CLS	CLE	CLEI	CS	CSI
7	CLT	CLC	CLS	CLE	CLEI	CS	CSI

Table 4-45 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
CLT	Call type: 0 - NULL 1 - OUTGOING	UBINARY	1
CLC	A number of communication channel which the modem is connected to. 0 - OPTICAL 1 - CHANNEL 1 2 - CHANNEL 2	UBINARY	1
CLS	Indexes of called station from a list in a group of registers CALLSTATION	UBINARY	1
CLE	An event at which a call occurs: 0 - NULL 1 - ALARM	UBINARY	1
CLEI	Index of event from a group of registers ALARMPARAM	UBINARY	1
CS	Cause for a call 0 - NULL 1 - ALARM 2 - IEC1107DATA	UBINARY	1
CSI	Index of cause is: • a system message from the ALARM class of messages, if a cause is “ALARM” • a channel through which data from a meter were read out if the cause is “IEC1107DATA”	UBINARY	1

A call depends on conditions “an event” (CLE) and “a cause” (CS) which define when a call is performed. The following table is valid:

“event”	“cause”	operation
0	0	No call
0	1	Calls if “a cause” is fulfilled, i.e. when the system message occurs, or when the data (IEC1107) are available.
1	0	A call if “an event” is fulfilled, i.e. when an alarm is activated
1	1	A call if “an event” **and** ”a cause” are fulfilled

4.45 CASCADE

A group involves parameters of Cascade communication function.

GID = 189
ATTR = R, W, WP, PAR, EE
RPG = 1
BPR = 2
FD0 = UBINARY + 1, 2
FPR = 2

Tabular display:

	0	1
0	Status	Channel

Table 4-46 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
Status	Status of connection: 0 - INACTIV 1 - ACTIV	UBINARY	1
Channel	Primary cascade channel: 0 - OPTICAL 1 - CHANNEL 1 2 - CHANNEL 2	UBINARY	1

Warning:
A communication channel of a device which is defined as a primary cascade channel of cascade communication should not have **Cascade** function set (a group of registers CHANNELPARAM, parameter F)!

4.46 TARDAYPROG

A group of parameters of daily tariff regulations. They consist of beginning and end of individual parts of a day which divide a day into active or inactive parts. Individual daily rule also indicates for how many hours individual parts of a day are shifted in a summer time.

GID = 190
ATTR = R, W, WP, PAR, EE
RPG = 32
BPR = 17
FD0 = UBINARY + 1, 17
FPR = 17

Tabular display::

	0	1	2	3		14	15	16
	start1		end1			end4		
0	hour	minute	hour	minute	...	hour	minute	daylight
	start1		end1			end4		
31	hour	minute	hour	minute	...	hour	minute	daylight

Table 4-47 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
start1 ... start4	Indicates when section 1 ... 4 (hours, minutes) starts		
	start - hour (0 - 23)	UBINARY	1
	start - minute (0 - 59)	UBINARY	1
end1 ... end4	Indicates when section 1 ... 4 (hours, minutes) finishes		
	end - hour (0 - 24)!	UBINARY	1
	end - minute (0 - 59)	UBINARY	1
daylight	Indicates for how many daily sections in a summer time are shifted (hours)	UBINARY	1

At calculation of results of daily tariff rules it is stated if current time within any start/end pair (start ≤ current time < end) exists. In this case a daily rule is active.

4.47 TARWEEKPROG

A group involves parameters of weekly and tariff rules. Weekly tariff rules consist of indexes of daily tariff rules which are used for individual days in a week or a holiday.

GID = 191
ATTR = R, W, WP, PAR, EE
RPG = 16
BPR = 8
FD0 = UBINARY + 1, 8
FPR = 8

Tabular display:

	0	1	2	3	4	5	6	7
0	Sun	Mon	Tue	Wed	Thu	Fri	Sat	hol
15	Sun	Mon	Tue	Wed	Thu	Fri	Sat	hol

Table 4-48 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
Sun ... Sat	Daily rule index which is valid for a certain day in a week	UBINARY	1
hol	Daily rule index which is valid for a holiday	UBINARY	1

At calculation of daily rule results it is stated whether a daily rule is active for a current day in a week. If a current day is a holiday, a daily rule for a holiday is checked. A holiday has precedence over checking.

4.48 TARSEASPROG

A group involves parameters of seasonal tariff rules. A seasonal tariff rule consists of a date of the beginning and the end of a season. Each date consists of a starting date and a rule of repetition.

GID = 192
ATTR = R, W, WP, PAR, EE
RPG = 16
BPR = 20
FD0 = UBINARY + 2, 1
FD1 = UBINARY + 1, 8
FD2 = UBINARY + 2, 1
FD3 = UBINARY + 1, 8
FPR = 18

Tabular display:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	SY	SM	SD	SWD	SADR	SH	SN	SS	SATR	TY	TM	TD	TWD	TADR	TH	TN	TS	TATR
15	SY	SM	SD	SWD	SADR	SH	SN	SS	SATR	TY	TM	TD	TWD	TADR	TH	TN	TS	TATR

Table 4-49 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
SY	Season start year (1990 to 2089)	UBINARY	2
SM	Season start month (1 - 12)	UBINARY	1
SD	Season start day (1 - 31)	UBINARY	1
SWD	Day in a week of season start (0 = Sunday, 1 = Monday, ...)	UBINARY	1
SADR	Season start algorithm of date repetition	UBINARY	1
SH	Season start hour (0 - 23)	UBINARY	1
SN	Season start minute (0 - 59)	UBINARY	1
SS	Season start second (0 - 59)	UBINARY	1
SATR	Season start algorithm of time repetition	UBINARY	1
TY	Season termination year (1990 to 2089)	UBINARY	2
TM	Season termination month (1 - 12)	UBINARY	1
TD	Season termination day (1 - 31)	UBINARY	1
TWD	Day in a week of season termination (0 = Sunday, 1 = Monday, ...)	UBINARY	1
TADR	Season termination algorithm of date repetition	UBINARY	1
TH	Season termination hour (0 - 24)!	UBINARY	1
TN	Season termination minute (0 - 59)	UBINARY	1
TS	Season termination second (0 - 59)	UBINARY	1
TATR	Season termination algorithm of time repetition	UBINARY	1

At calculation of results of seasonal rules it is checked if a current date is within a time period which is defined by the beginning and the end of a season (season start ≤ current date < season end). In this case the result of seasonal rule is active.

Repetition algorithms for date are:

0	once	repeated once exactly on day, month and year
1	yearly	repeated each year on day and month
2	yearlywd	repeated each year on "a day in a week "after "a day" and "month"
3	yearlyns	repeated each year on a day and month; if Sunday, it is moved to Monday
4	yearlyaeast	repeated each year for X days after Easter (X = parameter "day" = 0- 255)
5	yearlybeast	repeated each year for X days before Easter (X = parameter "day" = 0 - 255)
6	monthly	repeated each month on a certain day
7	monthlywd	repeated each month on a day in a week after a certain day
8	weekly	repeated each week on a day in a week
9	daily	repeated each day

Napaka! Vira sklicevanja ni bilo mogoče najti.
Seasonal rules can be used for generating of measuring period signals. A measuring period signal can be defined at the beginning of the measuring period. The following example illustrates the defining of measuring period signal (9s) at the beginning of the 15-minute measuring period:

	0	1	2	3	4	5	6	7	8
	LZ	MZ	DZ	DTZ	AZ	UZ	NZ	SZ	ČZ
0	X	X	X	X	9	X	15	0	3

	9	10	11	12	13	14	15	16	17
	LK	MK	DK	DTK	AK	UK	NK	SK	ČK
0	X	X	X	X	9	X	15	9	3

The result which appears in the TARSEASRES group of registers in register 0 (valid in this example), connects itself with any output in the OPORTPARAM group of registers.

4.49 TARHOLPROG

A group involves parameters of tariff rules for holidays. They consists of a date of a holiday and algorithm for holiday repetition

GID = 193
ATTR = R, W, WP, PAR, EE
RPG = 32
BPR = 6
FD0 = UBINARY + 2, 1
FD1 = UBINARY + 1, 4
FPR = 5

Tabular display:

	0	1	2	3	4
0	Y	M	D	DW	ALG
31	Y	M	D	DW	ALG

Table 4-50 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
Y	Year (1990 - 2089)	UBINARY	2
M	Month (1 - 12)	UBINARY	1
D	Day (1 - 31)	UBINARY	1
DW	Day in a week (0 = Sunday, 1 = Monday, ...)	UBINARY	1
ALG	Algorithm of date repetition	UBINARY	1

At calculation of results of tariff rules for holidays it is checked whether a current date corresponds to a date of the rule. If it is the case, a tariff rule for a holiday is active. A current day is a holiday if at least one rule for a holiday is active.

Repetition algorithms for date are:

0	once	repeated once exactly on day, month and year
1	yearly	repeated each year on day and month
2	yearlywd	repeated each year on "a day in a week "after "a day" and "month"
3	yearlyns	repeated each year on a day and month; if Sunday, it is moved to Monday
4	yearlyaeast	repeated each year for X days after Easter (X = parameter "day" = 0- 255)
5	yearlybeast	repeated each year for X days before Easter (X = parameter "day" = 0 - 255)
6	monthly	repeated each month on a certain day
7	monthlywd	repeated each month on a day in a week after a certain day
8	weekly	repeated each week on a day in a week
9	daily	repeated each day

4.50 TARPROG

A group involves parameters of tariff programs. Tariff programs combines individual tariff rules into a uniform tariff program. For each tariff program it is defined in which seasons certain weekly rules are valid.

GID = 194
ATTR = R, W, WP, PAR, EE
RPG = 16
BPR = 32
FD0 = UBINARY + 1, 32
FPR = 32

Tabular display:

	0	1	2	3		30	31
0	SR0	WR0	SR1	WR1	...	SR15	WR15
15	SR0	WR0	SR1	WR1	...	SR15	WR15

Table 4-51 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
SR0 ... SR15	Index of seasonal rule	UBINARY	1
WR0 ... WR15	Index of weekly rule	UBINARY	1

At calculation of results of tariff program it is checked if at least one pair: seasonal rule / weekly rule, which has both seasonal and weekly rule active, exists. In this case a tariff program is active.

4.51 ALARMPARAM

A group involves parameters of repeatable time alarms. Parameters consist of a date and time of alarm and algorithm for repetition regarding date or time.

GID = 195
ATTR = R, W, WP, PAR, EE
RPG = 16
BPR = 10
FD0 = UBINARY + 2, 1
FD1 = UBINARY + 1, 8
FPR = 9

Tabular display:

	0	1	2	3	4	5	6	7	8
0	Y	M	D	DW	AD	H	MN	S	AT
15	Y	M	D	DW	AD	H	MN	S	AT

Table 4-52 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
Y	Year (1990 - 2089)	UBINARY	2
M	Month (1 - 12)	UBINARY	1
D	Day (1 - 31)	UBINARY	1
DW	Day in a week (0 = Sunday, 1 = Monday, ...)	UBINARY	1
AD	Algorithm for date repetition	UBINARY	1
H	Hour (0 - 23)	UBINARY	1
MN	Minute (0 - 59)	UBINARY	1
S	Second (0 - 59)	UBINARY	1
AT	Algorithm for time repetition	UBINARY	1

Time alarms are repeatable time events which trigger internal processing in a data recorder. At calculation of alarms which are being generated it is checked if a current date corresponds to a date of a time alarm, and if a current time corresponds to the time of time alarm. In this case a signal is generated for an alarm which is used by various functions of internal processing in a data recorder.

Repetition algorithms for date are:

0	once	repeated once exactly on day, month and year
1	yearly	repeated each year on day and month
2	yearlywd	repeated each year on "a day in a week "after "a day" and "month"
3	yearlyns	repeated each year on a day and month; if Sunday, it is moved to Monday
4	yearlyaeast	repeated each year for X days after Easter (X = parameter "day" = 0- 255)
5	yearlybeast	repeated each year for X days before Easter (X = parameter "day" = 0 - 255)
6	monthly	repeated each month on a certain day
7	monthlywd	repeated each month on a day in a week after a certain day
8	weekly	repeated each week on a day in a week
9	daily	repeated each day

Napaka! Vira sklicevanja ni bilo mogoče najti.

4.52 UNIT

A group involves units of format listings. Units consist of a unit exponent and unit text.

GID = 200
ATTR = R, W, WP, WL, PAR, EE
RPG = 16
BPR = 9
FD0 = SBINARY + 1, 1
FD1 = ASCII + 8, 1
FPR = 2

Tabular display:

	0	1
0	exponent	text
15	exponent	text

Table 4-53 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
exponent	Exponent for unit listing	SBINARY	1
text	Unit text (max. 8 characters)	ASCII	8

At formatted listings it is possible to define that a value is listed in a certain unit. An exponent indicates the value exponent being listed, while the text can be listed next to the value.

Example:	
Recorder internal unit :	0.1 Wh (one impulse in data recorder represent 0.1 Wh)
Record in UNIT group register:	-4, “kWh “
Listing format:	09.4 (form of result is XXXX.XXXX)
A listing for one internal unit:	0000.0001 kWh

4.53 FORMSTR

A group involves format strings which represents strings for control of format listing. Format strings consist of two types of objects, a text for listing and specifications for conversion.
More detailed information for format strings is given in Chapter FORMAT LISTING.

GID = 201
ATTR = R, W, WP, WL, PAR, EE
RPG = 32
BPR = 40
FD0 = ASCII + 40, 1
FPR = 1

Tabular display:

	0
0	format string
31	format string

Table 4-54 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
format string	Max. 40 characters	ASCII	40

- flags : alignment, listing of signs
  - - left aligned specification result
  - + sign always present
  - blank sign always present (plus as blank)
- width : minimal number of characters on output.
  - n minimal **n** characters on output
  - 0n minimal **n** characters on output
- .prec : number of decimal places on output
  - .n **n** decimal places on output (decimal point is not considered)
- type modifier : modification of argument type
  - a adding up of argument to specification result
  - s subtraction of argument from specification result
  - m multiplication of specification result by argument
  - d division of specification result by argument
  - e specification result exponent
  - p calculation of power in a measuring period
  - r multiple application of format string (max. 255)
  - u result unit exponent
  - t display of value in a form (DD.MM.YY hh:mm:ss W(S)); no argument required!
  - x multiple application of format string (max. 65535); **2** argument required!
- type : argument type
  - D date
  - T time
  - F register field from which it is read
  - R register where information is read from
  - M successive number of multiple application of format string
  - N successive number of listing
  - U result unit text
  - S profile record stamp
  - P a field of register in a profile of the last terminated saving period

4.54 FORMOUT

A group involves format outputs. They consist of format strings and arguments of format outputs. A format string monitors collection, conversion and formatting of format output arguments in a format listing. Format output arguments are collected regarding the specifications of conversion in a format string. A number and values which are taken by arguments depend on individual specification for conversion.
More detailed information on format strings and format outputs is stated in Chapter FORMAT LISITING.

GID = 202
ATTR = R, W, WP, WL, PAR, EE
RPG = 32
BPR = 17
FD0 = UBINARY + 1, 17
FPR = 17

Tabular display:

	0	1	2		15	16
0	FS	ARG0	ARG1	. . .	ARG14	ARG15
31	FS	ARG0	ARG1	. . .	ARG14	ARG15

Table 4-55 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field		Contents	Type	Length
FS		Index of format string for format listing	UBINARY	1
ARG0 ARG15	...	Format output argument	UBINARY	1

4.55 FORMLST

A group involves parameters of format lists. Format lists consist of individual format outputs. A format list is terminated by a value 255 for a format output. They are intended for listing of several format outputs which follow each other. For each listing it is possible to say which format outputs are to be listed one after another.
More detailed information on the application of format lists is stated in Chapter FORMAT LISTING.

GID = 203
ATTR = R, W, WP, PAR, EE
RPG = 16
BPR = 32
FD0 = UBINARY + 1, 32
FPR = 32

Tabular display :

	0	1	2		14	15
0	Fo0	Fo1	Fo2	. . .	Fo30	Fo31
15	Fo0	Fo1	Fo2	. . .	Fo30	Fo31

Table 4-56 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
Fo0	Index of format output 0 in a listing (255 is the end of listing)	UBINARY	1
...			
Fo31	Index of format output 31 in a list	UBINARY	1

4.56 FORMCMD

A group involves parameters of format commands. Format commands consist of data which cite to which output device format lists are to be listed, when they are listed and which are time intervals between listings.
More detailed information on format commands are given in Chapter FORMAT LISTING.

GID = 204
ATTR = R, W, WP, WL, PAR, EE
RPG = 16
BPR = 5
FD0 = UBINARY + 1, 5
FPR = 5

Tabular display:

	0	1	2	3	4
0	FL	device	channel	alarm	delay
15	FL	device	channel	alarm	delay

Table 4-57 Group of registers **Napaka! Slog ni definiran.**

Contents of fields

Field	Contents	Type	Length
FL	Index of format list for listing	UBINARY	1
device	Output device for listing: 0 - NULL is not listed 1 - DISPLAY listed to a display 2 - TERMINAL listed via a port for a terminal 3 - PRINTER listed via a port to a printer	UBINARY	1
channel	Channel for output devices if several channels are available	UBINARY	1
alarm	Index of repeatable time alarm used for triggering a listing (group of registers ALARMPARAM)	UBINARY	1
delay	Time interval between listings in a format list in seconds	UBINARY	1

For "Output device 1" (display) it is valid that a channel is:

- 0 = “Manualscroll” mode of format list displaying when a listing is changed by pressing the ROLL key. A parameter "delay" defines time of the last pressing the ROLL key and restoring the "Autoscroll" display mode.
- 1 = “Autoscroll” mode of format list displaying in time intervals which are defined with parameters "delay"

For "Output device 1” both display modes should be defined.

For Output device 2” (terminal) and "Output device 3" (printer) a serial communication channel is set with a "channel" parameter:
0 - OPTICAL
1 - CHANNEL 1 (RS232/RS485)
2 - CHANNEL 2 (RS232/RS485 - MODEM)

4.57 PROFILEPAR

A group involves parameters for saving data into profiles. Profiles are structures which enable periodical saving of registers to different media for saving (profile pools). For each profile it is possible to define where data will be saved, how much profile pool will be occupied, when they are saved and which data are saved.
More detailed information on data saving is given in Chapter SAVING.

GID = 210
ATTR = R, W, WP, WL, PAR, EE
RPG = 4
BPR = 28
FD0 = UBINARY + 1, 28
FPR = 28

Tabular display :

	0	1	2	3	4 - 9	10 - 15	16 - 21	22 - 27
0	L	P	SE	ISE	RSD0	RSD1	RSD2	RSD3
1	L	P	SE	ISE	RSD0	RSD1	RSD2	RSD3
2	L	P	SE	ISE	RSD0	RSD1	RSD2	RSD3
3	L	P	SE	ISE	RSD0	RSD1	RSD2	RSD3

Table 4-58 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
L	Location - media to which data are being saved	UBINARY	1
	0 - NULL are not saved		
	1 - RAM internal RAM memory		
	2 - EEPROM internal EEPROM memory		
	3 - FLASH internal FLASH memory		
	4 - MCU memory card		
P	Percent of pool which will be occupied	UBINARY	1
SD	Saving event :	UBINARY	1
	0 - NULL is not saved		
	1 - Measuring period termination of measuring period triggers saving		
	2 - Alarm internal alarm triggers saving		
ISD	Index of saving event indicates which measuring period (0 - MP1, 1 - MP2, 2 - MP3) or which internal alarm (group of registers ALARMPARAM) triggers saving.	UBINARY	1
RSD0 ... RSD3	Description of definition of a block of registers being saved. It consists of six fields:		
	gid - identification of a group of registers	UBINARY	1
	rid - identification of the first register in a block of registers	UBINARY	1
	fid - identification of the first register field in a block of registers	UBINARY	1
	nre - number of registers in a block of registers	UBINARY	1
	nfi - number of fields in a block of registers	UBINARY	1
	bpf - max. number of bytes per field	UBINARY	1

The attributes of saving are calculated regarding the parameters of saving registers into profiles. They indicate how the profiles were distributed into profile pools. Each modification of saving attributes causes profile deleting and thus permanent loss of recorded information.

4.58 PROFILEATT

A group involves attributes of saving registers into profiles. Attributes consist of data on the location where the registers will be recorded, where and how large a profile in a profile pool is, the structure which is recorded, a saving period and a description of fields which form a record.
More detailed information on saving data into profiles is given in Chapter SAVING.

GID = 211
ATTR = R
RPG = 4
BPR = 50
FD0 = UBINARY + 2, 1
FD1 = UBINARY + 4, 4
FD2 = UBINARY + 1, 48
FPR = 53

Tabular display:

	0	1	2	3	4	5 - 16	17 - 28	29 - 40	41 - 52
0	L	BA	SP	SR	PS	PFD0	PFD1	PFD2	PFD3
1	L	BA	SP	SR	PS	PFD0	PFD1	PFD2	PFD3
2	L	BA	SP	SR	PS	PFD0	PFD1	PFD2	PFD3
3	L	BA	SP	SR	PS	PFD0	PFD1	PFD2	PFD3

Table 4-59 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
L	Location - media to which data are being stored: 0 - NULL are not saved 1 - RAM internal RAM memory 2 - EEPROM internal EEPROM memory 3 - FLASH internal FLASH memory 4 - MCU memory card	UBINARY	2
BA	Base address is address of the beginning of a profile in a profile pool	UBINARY	4
SP	Size of profile in bytes	UBINARY	4
SR	Size of record in bytes	UBINARY	4
PS	Period of saving in seconds	UBINARY	4
PFD0 ... PFD3	Description of fields being saved. They consist of:		
	type of field 0	UBINARY	1
	number of fields 0	UBINARY	1
	type of field 1	UBINARY	1
	number of fields 1	UBINARY	1
	type of field 2	UBINARY	1
	number of fields 2	UBINARY	1
	type of field 3	UBINARY	1
	number of fields 3	UBINARY	1
	type of field 4	UBINARY	1
	number of fields 4	UBINARY	1
	type of field 5	UBINARY	1
	number of fields 5	UBINARY	1

More detailed information on types of fields in registers is given in Chapter REGISTERS and in Chapter COMMUNICATION, command RI.

4.59 PROFILESTAT

This group contains status information on saving of registers into profiles. Information consists of status, serial number of saving into profiles and time stamps of the last saving. Detailed information on saving of data into profiles is given in the chapter on SAVING.

GID = 212
ATTR = R
RPG = 4
BPR = 10
FD0 = UBINARY + 2, 1
FD1 = UBINARY + 4, 2
FPR = 3

Tabular display:

	0	1	2
0	Status	WriteCount	WriteStamp
1	Status	WriteCount	WriteStamp
2	Status	WriteCount	WriteStamp
3	Status	WriteCount	WriteStamp

Table 4-60 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
Status	Information on events which influence saving. The bits from 0 to 15 represent an event with proper values: 0 - NONE no events 1 - PWRFAIL power supply failure at processor 2 - PWRON power supply activation at processor 4 - TIMESYNC_LP saving period longer due to time synchronization 8 - TIMESYNC_SP saving period shorter due to time synchronization 16 - DATAADJUST data changed manually 32 - DATAOVERRUN data exceeded the maximum value; permission depends on the type of data 64 - DATAINVALID data to be saved are invalid 32768 - SYS_ERROR system error The bits from 7 to 14 are unused. The value entered in this field is restored at the termination of saving period. The new value is a sum of values of different events written in a decimal form.	UBINARY	2
WriteCount	Serial number of all saving into profiles	UBINARY	4
WriteStamp	Time stamp of the last saving into profiles written as the number of seconds elapsed from 1.1.1970 at 00:00	UBINARY	4

4.60 POOLATTR

This group contains information on location and the size of pool of the memory medium.

GID = 213
ATTR = R
RPG = 5
BPR = 8
FD0 = UBINARY + 4, 2
FPR = 2

Tabular display:

	0	1
0	Position	Size
1	Position	Size
2	Position	Size
3	Position	Size
4	Position	Size

Table 4-61 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
Position	Location of the beginning of profile	UBINARY	4
Size	Size of memory space in pool intended for saving into profiles (byte)	UBINARY	4

The registers represent:

register	**meaning**
00	memory medium 0 - NULL
01	memory medium 1 - RAM
02	memory medium 2 - EEPROM
03	memory medium 3 - FLASH
04	memory medium 4 - MCU

5. TECHNICAL DATA

5.1 INPUT LINES

5.1.1 Input

Input lines require external power supply. They differ in size and type of input voltage. Input modules combine input lines of the same size and type of input voltage. Different modules can be built in a register. Data for voltage of input lines involve all possible voltages for a certain type of voltage.
Input lines functions correspond to all types of inputs (meter impulses, testing).
There are four types:

a) Two-wire input for active current
Meters with pulse transmitters are connected. During the active signal a circuit on signal lines is closed.
Maximal values:

• maximal voltage	U_{max} =	24V=
• maximal frequency	f_{max} =	10 Hz
• minimal impulse length	t_{min} =	program adjustable

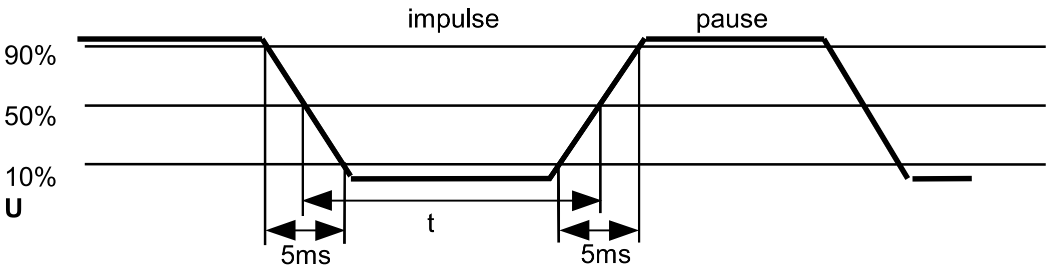


Figure 5-1 Time diagram - Two-wire input for active current

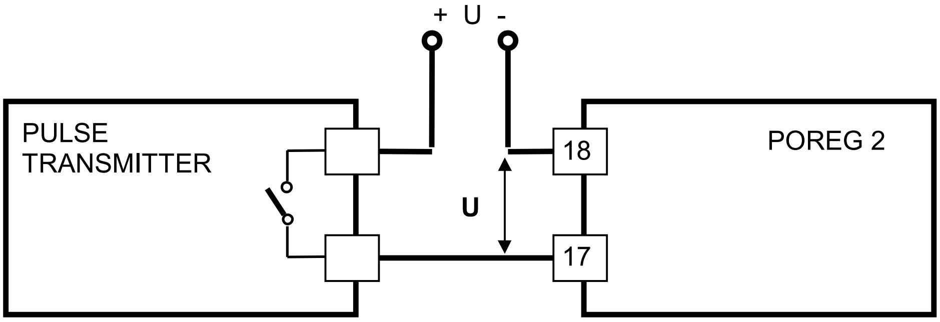


Figure 5-2 Connection scheme for a typical two-wire input for active current

* Terminal 18 on POREG 2 is common with input 2. The same is valid for all other pairs of inputs, i.e. the middle terminal covers a pair of inputs (21, 14, 27, 30, 33, 36 and 39).

b) Two-wire input for inactive current
The same characteristics are valid as for input for active current. During the active signal the circuit on signal lines is open.

c) Two-wire input for double current is not realized yet
Meter with pulse transmitter are connected. Transition of signal on signal lines from +24 V to -24 V or vice versa represents a certain quantity of energy.
Maximal values:

• maximal voltage	U_{max} =	±24 V
• maximal frequency	f_{max} =	40 Hz
• duration of transition	t =	program adjustable

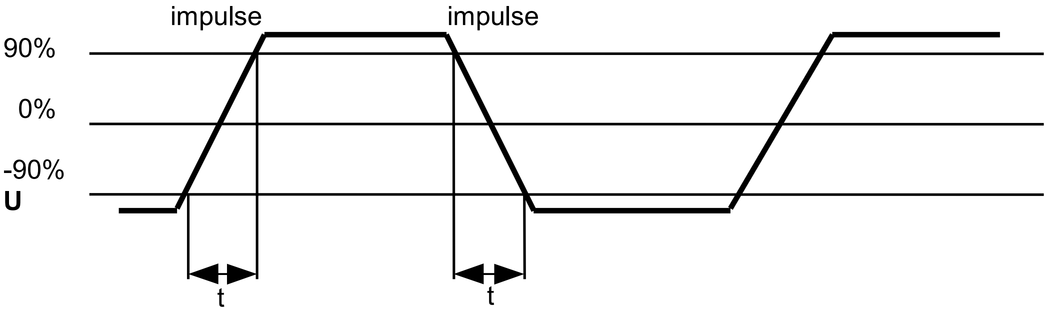


Figure 5-3 Time diagram - Two-wire input for double current

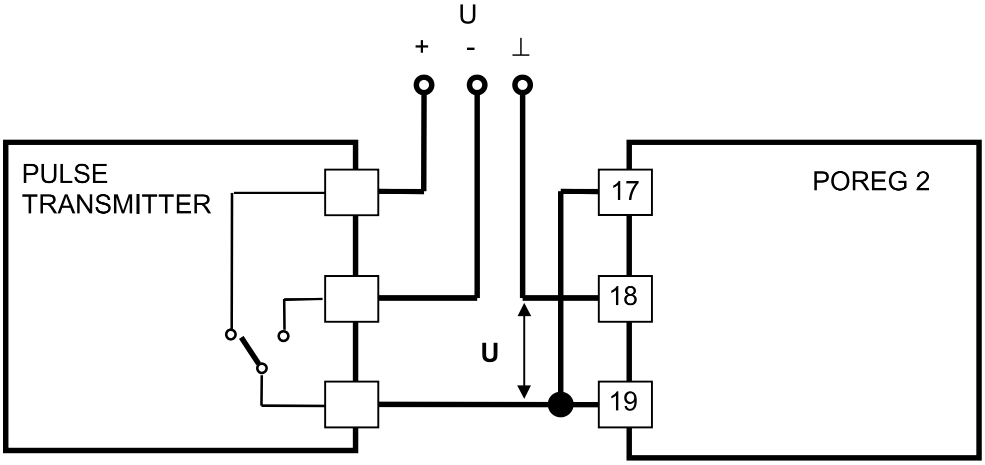


Figure 5-4 Connection scheme for a typical two-wire input for double current

d) Three-wire input is not realized yet
Meters with pulse transmitters are connected. Simultaneous closing of one line and opening of another line or vice versa represents a certain quantity of energy.
Maximal values:

• maximal voltage	U_{max} =	24V=
• maximal frequency	f_{max} =	40 Hz
• transition time	t_{min} =	program adjustable

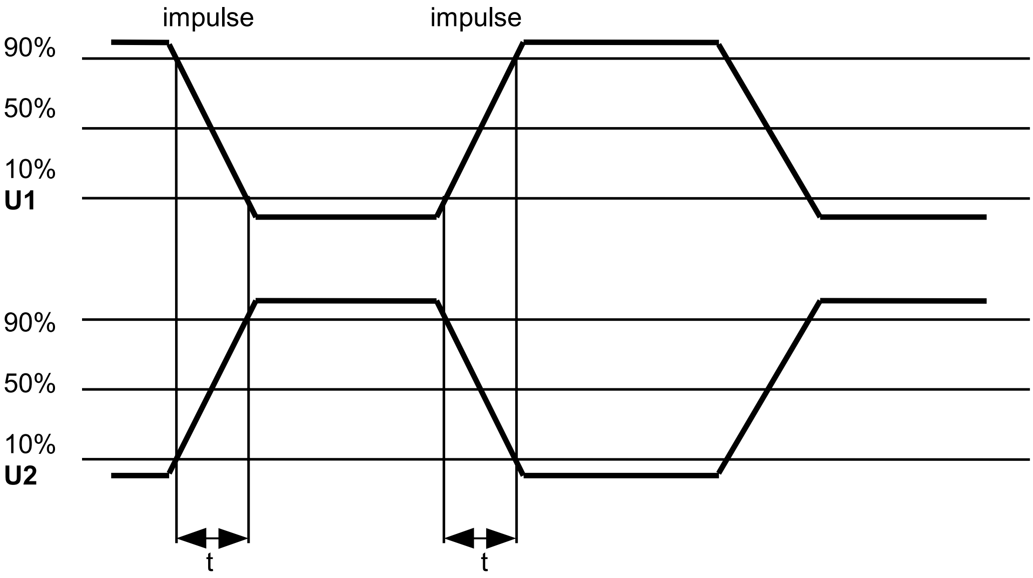


Figure 5-5 Time diagram - Three-wire input

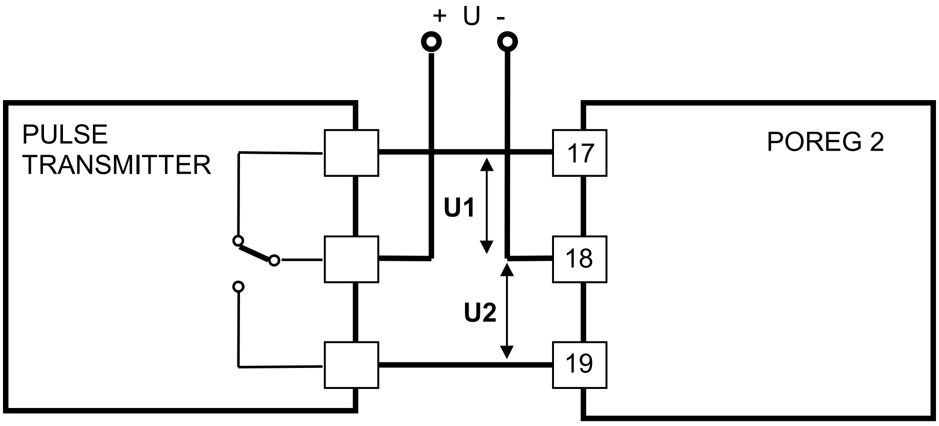


Figure 5-6 Connection scheme for a typical three-wire input

5.1.2 Synchronization input

Synchronization input is used for connecting external synchronization devices. Any above stated type of inputs can be used.

- time of synchronization positive transition of signal in time complete minute ± 0,5 minute

5.2 OUTPUT LINES

Impulse outputs are performed in a form of passive contact by means of semiconductor relays or mercury relays. All output lines can be programmed. They can take a function of impulse outputs or signal outputs..

Type and characteristics of switching:

parameter	semiconductor relay	mercury relay	unit
maximal permanent current	0,15	0,18	A
range of switching voltage	0,1 - 250	0,1 - 250	V
life time limited with mechanical characteristics	-	$5x10^6$	switchings
life time at maximal electric loads	-	$1x10^6$	switchings
internal resistance of closed contact (protection resistance):	46	30	Ω
form of impulses	program adjustable from 1 to 255 ms in steps by 1 ms separately for time of pause and impulse		

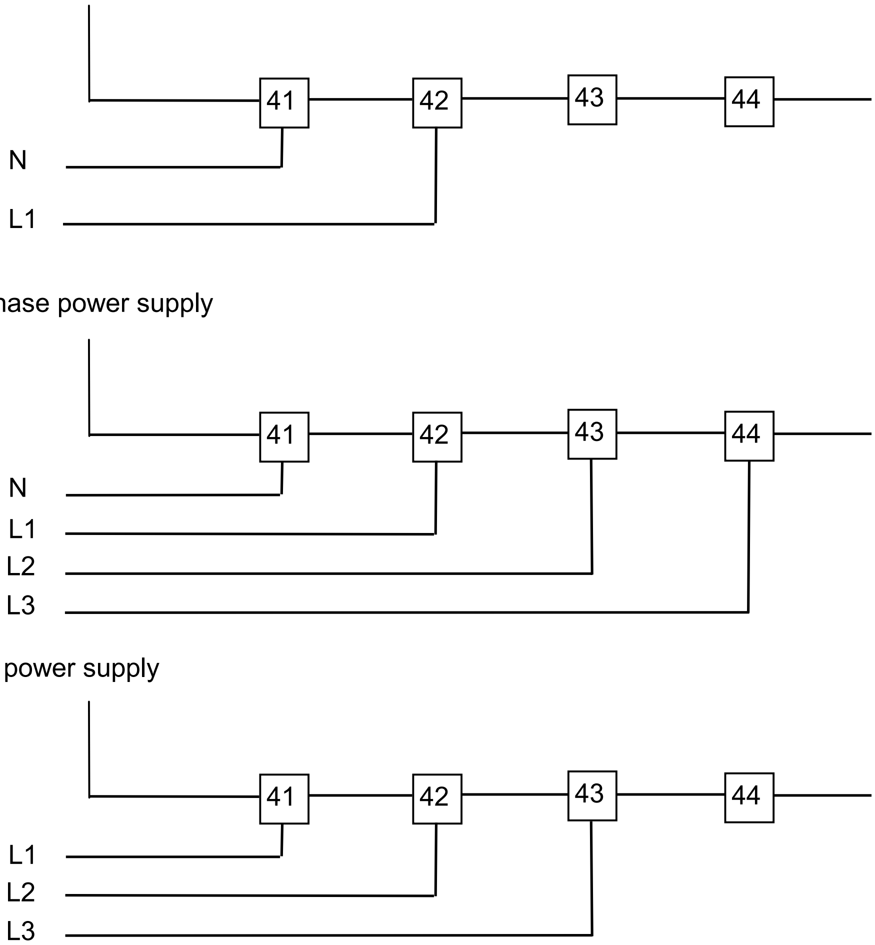
Table 5-1 Outputs - Type and characteristics of switching

5.3 POWER SUPPLY

5.3.1 Main supply

	single-phase	three-phase
range of power supply voltage	100 - 230 V (-10, +15%)	3 x 100 V - 3 x 230 V (-10, +15%)
range of power supply voltage frequency	45 - 65 Hz	
consumption in one phase	max. 20 VA	
protection elements	2 A	
time of reaction of protection element	M, T	

A slow protection element is recommended (mark T)!



a) Single-phase power supply

b) Three-phase power supply

c) 57/100V power supply

The phase presence on connections 41 is compulsory!

Figure 5-7 Connection scheme - Power supply

5.3.2 Internal stand-by power supply

Internal clock is protected with Li-battery and keeps real time in case of main power supply failure.
Characteristics of back-up battery:

• type of Li-battery:	CR 2450 N
• capacity of Li-battery:	500 mAh
• time of real time keeping when other power supply sources are switched off:	4 years
• battery life:	min.10 years from delivery
• self-discharging by 25°C :	1 % per year

Li- battery should be replaced every ten years irrespective of a degree of battery discharging, i.e. to operation conditions.

5.3.3 External stand-by power supply

External power supply is required for the case of main power supply failure.

• voltage	24 V ±15%
• time of switching to stand-by power supply	uninterrupted switching

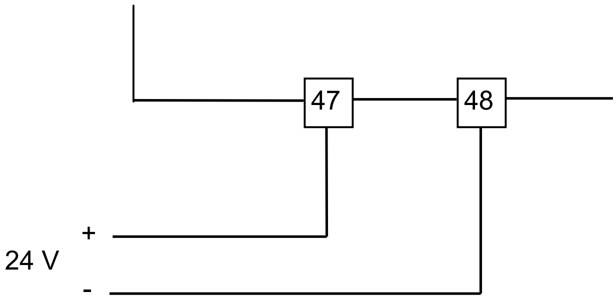


Figure 5-8 Connection scheme - External stand-by power supply

5.3.4 Supply of input lines

For supplying input lines (e.g. meter inputs) voltage which is generated by the data recorder can be used.

• voltage	24 V
• maximal current	0,18 A
• protection element	electronic circuit protection

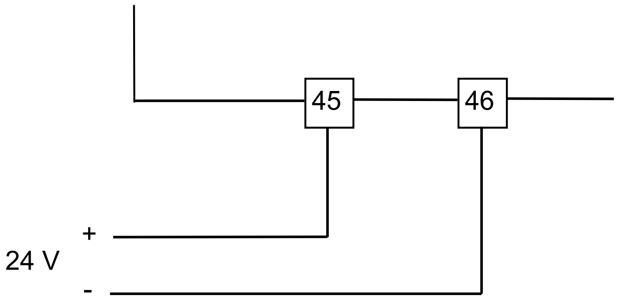


Figure 5-9 Connection scheme - Supply of input lines

5.4 COMMUNICATION

5.4.1 RS485/RS232C

The module 2x RS485/RS232C is integrated in a device and is intended for connection with a PC into a local network in compliance with EIA RS485 standard or external DCE devices in compliance with EIA RS232C standard.

General data:

RS485

• transmission rate:	150 - 57600 bit/s
• transmission mode :	two-wire
• maximal line length :	1200 m
• maximal number of communication places :	32 (computer + 31 devices)

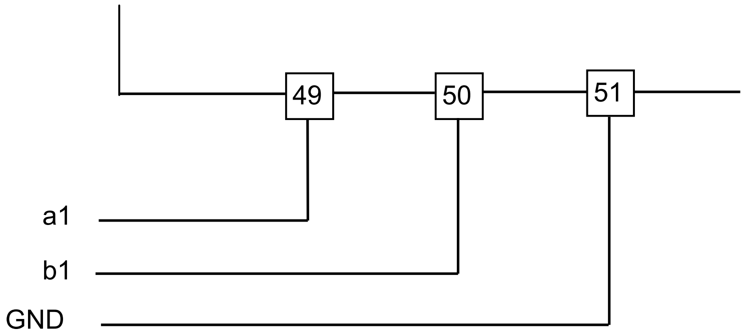


Figure 5-10 Connection scheme - RS485

RS232C

• transmission rate:	150 - 57600 bit/s
• transmission mode:	asynchronously, fullduplex
• maximal length of lines:	15 m

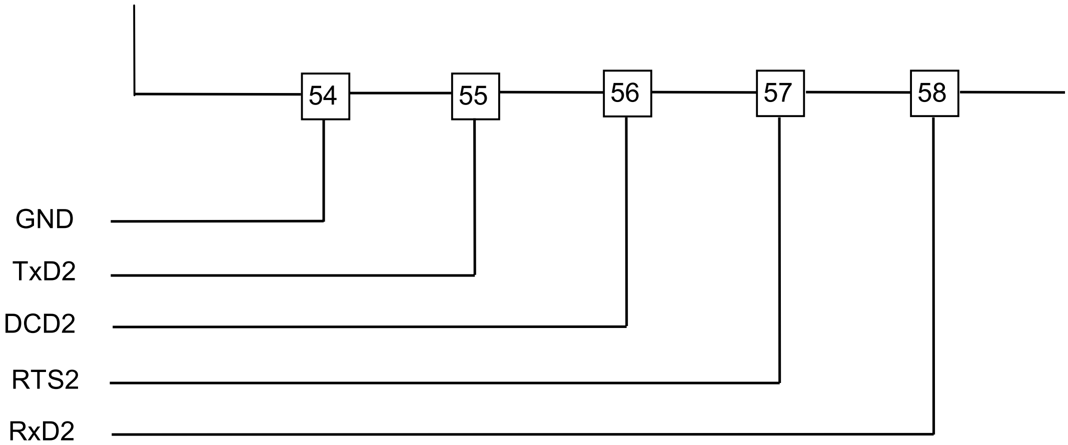


Figure 5-11 Connection scheme - RS232

5.4.2 Modem

Standards:	CCITT V.21, V.22, V.22bis in V.23 Bell 212A in 103 V.42 LAPM in MNP 2-4 V.42bis in MNP 5
Electrical data:	
• DC resistance	300 Ω
• max. line current of loop	80 mA
• detection of loop line current	15 - 100 mA
• transmission rate to DTE	150 - 19200 baud
• insulation strength between line and DTE	2 kV
• transmission level	-6 dBm ... -13 dBm
• receiving level	-6 dBm ... -43 dBm
• ring level	min. 27 $V_{RMS,}$ max. 120 V_{RMS}
• ringing frequency	15 - 68 Hz
• input impedance for a two-wire line	900 Ω (by 1 kHz)

5.4.3 Optical interface

Optical interface is intended for data transmission with a hand-held terminal or with a portable PC. Infrared light is used as a transmission media.
Data transmission is performed with a special probe through a window on a recorder where it is fixed with a magnet.

• Transmission rate	2400 bit/s.
• Maximal permitted probe shift from a stated position	30°

5.5 INTERNAL CLOCK

5.5.1 Internal time standard

Internal time standard is performed with a special computer unit with a crystal oscillator and is protected with a Li-battery (see Item 3.2.).
Silicious crystal:

• Frequency :	32768 Hz ±10ppm (25°C)
• Aging: deviation of frequency owing to aging	± 5 ppm
• Temperature stability : maximal deviation of frequency in a complete temperature operating range	- 120 ppm (-10°C to + 70°C)

5.6 MEMORY

5.6.1 Internal memory

Internal memory is EEPROM type (32 kbytes) and static RAM (256 kbytes).

5.6.2 Memory card

• card type		JEIDA 4.0 (68.pol)
• memory	type	static RAM or FLASH (5V)
	capacity	max. 16 MB
• temperature of storing		- 20°C to + 70°C
• battery	type	lithium CR 2025
	voltage	3 V
	capacity	2 years at memory capacity 64k 1 year at memory capacity 128k

WARNING : *The card is usually supplied without a battery!*

5.7 LOCAL DISPLAY

LCD with 16 characters in two lines is used. Character set is alphanumerical with additional characters. Records are n x 16 characters long. 2 x 16 characters are displayed at the same time. Characters are 8 mm high and are illuminated from the rear side.

5.8 OPERATION CONDITIONS

5.8.1 Ambient temperature

• working temperature range:	0°C - +50°C
• temperature of storing (without battery):	- 20°C - +65°C
• temperature of storing (with Li-battery):	- 20°C - +55°C

5.8.2 Air humidity

• relative air humidity	
at application:	max. 90 %
at storing:	max. 90 %
• direct contact with water is not permitted	

5.8.3 Dust protection

Protection degree, reached with a casing, is IP50 (IEC529).

5.9 DIELECTRIC STRENGTH AND RESISTANCE TO INTERFERENCES

5.9.1 Insulation strength

Dielectric strength test complies with IEC 255-4 standard.
a) A test is performed between all groups of electric terminals and earth.
b) A test is performed among all groups of electric terminals.

Groups of electric terminals are defined as:

• INPUTS	no. of terminals:	17, ... , 40
• OUTPUTS	no. of terminals:	1, ..., 12
• POWER SUPPLY	no. of terminals:	41, 42, 43, 44
• STAND-BY POWER SUPPLY	no. of terminals:	47, 48
• SUPPLY OF EXTERNAL LINES	no. of terminals:	45, 46
• RS 485	no. of terminals:	49, 50, 51, 52, 53
• RS 232C	no. of terminals:	54, ... , 63
• EARTH	definition:	metal plate pressed on a rear side, electrically connected with fixing screws

Test duration:
a) For a type test: 1 minute with values stated in a table.
b) For routine test: 1 minute with values stated in a table or 1 second with 1.1-time values stated in a table.

	Circuits connected to voltages			
	60V or less		60V to 500V	
Test voltage	a.c.	d.c.	a.c.	d.c.
	500V	500√2V	2000V	2000√2V

5.9.2 Resistance to voltage surges 1,2/50

Resistance test to voltage surges complies with IEC 255 - 4 standard (class III)

a) A test is performed with amplitude 5 kV in a longitudinal direction in all circuits for the following groups of electric terminals:
- INPUTS
- OUTPUTS
- SUPPLY

b) A test is performed with amplitude 5 kV in transverse direction between all groups of electric terminals and earth.

c) A test is performed with amplitude 5 kV in transverse direction between different groups of electric terminals.

Groups of electric terminals are defined in Item INSULATION STRENGTH.

5.9.3 Test with damped oscillatory wave 1 MHz

A test with high-frequency oscillating wave complies to IEC 255 - 4 standard (class III)

a) A test is performed with amplitude 1 kV in a longitudinal direction in all circuits for the following groups of electric terminals:
   - INPUTS
   - OUTPUTS
   - POWER SUPPLY

b) A test is performed with amplitude 2.5 kV in transverse direction between earth and the following groups of electric terminals:
   - INPUTS
   - OUTPUTS
   - POWER SUPPLY

c) A test is performed with amplitude 2.5 kV in transverse direction between different groups of electric terminals:
   - INPUTS
   - OUTPUTS
   - POWER SUPPLY

5.9.4 Fast transient burst test

A test of resistance to fast voltage surges complies with IEC 801 - 4 standard (Class III)

a) A test is performed with amplitude 2 kV in supply terminals between all supply terminals and earth..

b) A test is performed with amplitude 1 kV in all other electric terminals (inputs, outputs, modem or RS485/RS232C). Connection between connection cables and induced interference is capacitive.

5.9.5 High frequency electromagnetic field

A test complies with IEC 801-3 standard.

• Severity degree:	3
• Size of electrostatic field:	10 V/m
• Frequency range:	27 MHz to 500 MHz

5.9.6 Electrostatic discharges

A test complies with IEC 801-2 standard.

• Severity degree:	3
• Test voltage:	8kV + 10 %

5.10 LEVEL OF HARMONIC DISTURBANCES

Required level of harmonic disturbances which are generated by the device comply with EN 60555 part 2 - 3/4 1987.

5.11 DEGREE OF RF INTERFERENCES

The required degree of radio-frequency interference’s, generated by the device, complies with C.I.S.P.R.22 standard - class B.

5.12 DIMENSIONS

5.12.1 Fixing dimensions

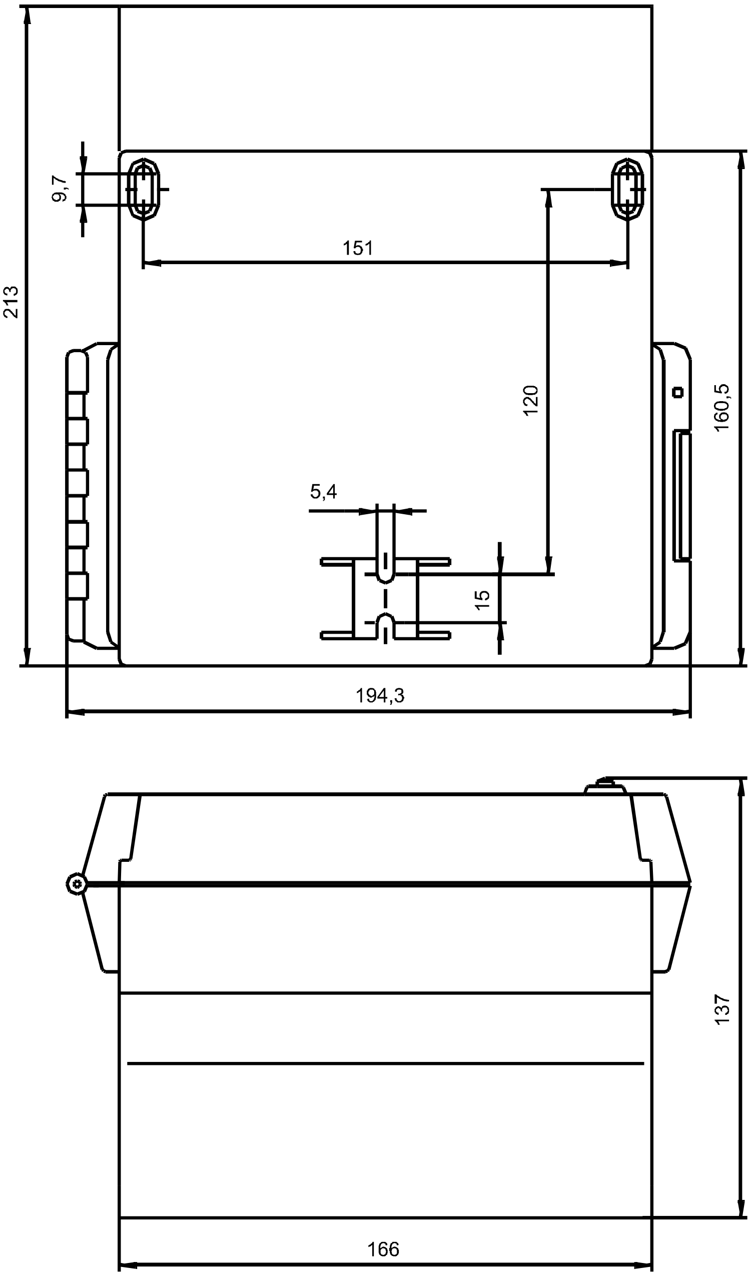


Figure 5-12 Fixing dimensions

5.12.2 Mass

POREG 2 device is 1.3 kg heavy.

5.13 CONNECTION DIAGRAM

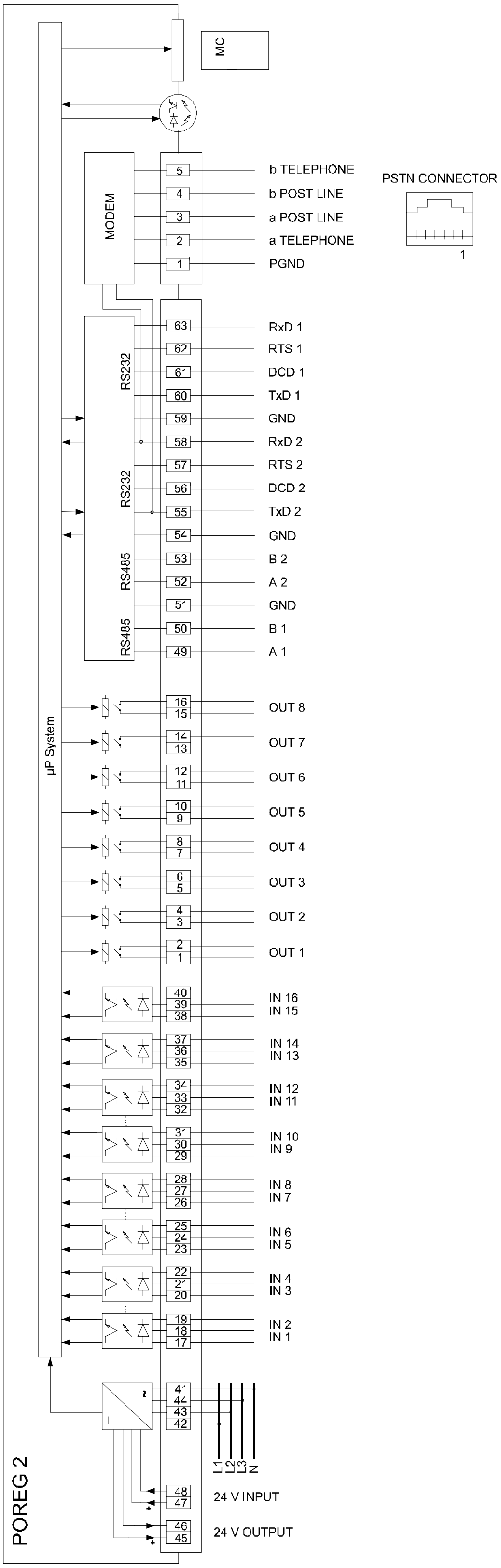


Figure 5-13 Connection diagram

6. TABLES OF REGISTERS

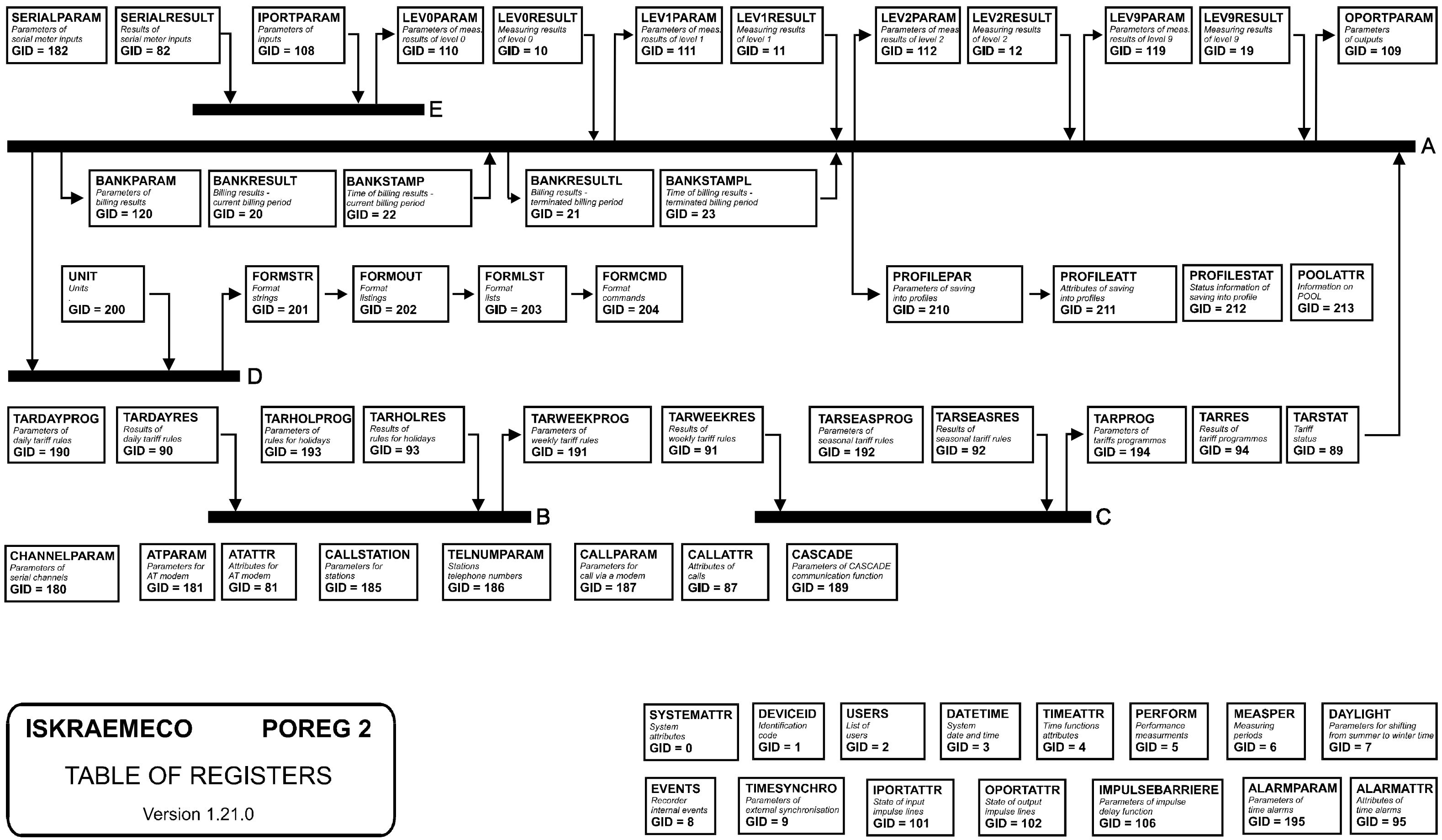


Figure 6-1 Table of registers - All registers

SYSTEMATTR
System attributes
GID = 0

Descript.	Contents
FID / RID	00
00	Status
01	Error
02	Alarm
03	Warning
04	PowerFailCount
05	PowerFailStamp
06	PowerFailDuration
07	PowerOnCount
08	PowerOnStamp
09	PowerOnDuration
10	PhaseRFailCount
11	PhaseRFailStamp
12	PhaseRFailDuration
13	PhaseROnCount
14	PhaseROnStamp
15	PhaseROnDuration
16	PhaseRFailCount
17	PhaseSFailStamp
18	PhaseSFailDuration
19	PhaseSOnCount
20	PhaseSOnStamp
21	PhaseSOnDuration
22	PhaseTFailCount
23	PhaseTFailStamp
24	PhaseTFailDuration
25	PhaseTOnCount
26	PhaseTOnStamp
27	PhaseTOnDuration
28	WatchDogCount
29	WatchDogStamp
30	RestartCount
31	RestartStamp

DEVICEID
Identification code
GID = 1

Descript.	IDENT
FID / RID	00
00	

USERS
List of users
GID = 2

Descript.	Code	Password	Rights
FID / RID	00	01	02
00			
01			
02			
03			
04			
05			
06			
07			

DATETIME
System date and time
GID = 3

Descript.	YEAR	MONTH	DAY	WDAY	HOUR	MIN	SEC	D.LIGHT
FID / RID	00	01	02	03	04	05	06	07
00								

TIMEATTR
Time functions attributes
GID = 4

Descript.	CONT.	CONT.	CONT.
FID / RID	00	01	02
00	TMP1	DMP1	SNMP1
01	TMP2	DMP2	SNMP2
02	TMP3	DMP3	SNMP3
03	TD	DD	SND
04	TM	DM	SNM
05	TY	DY	SNY
06	S	DS	SNS

PERFORM
Performance measurment
GID = 5

Descript.	LM	MM
FID / RID	00	01
00		
01		
02		
03		
04		
05		
06		
07		
08		
09		
10		

MEASPER
Measuring periods
GID = 6

Descript.	MP1	MP2	MP3
FID / RID	00	01	02
00			

DAYLIGHT
Parameters for shifting from summer to winter time
GID = 7

Descript.	YEAR	MONTH	DAY	WDAY	ALG
FID / RID	00	01	02	03	04
00					
01					

Figure 6-2 Table of registers - System (Part 1)

EVENTS
Recorder internal events
GID = 8

Descript.	PC	SC	TE	CE
FID / RID	00	01	02	03
00				
01				
02				
03				
253				
254				

TIMESYNCHRO
Parameters of external synchronisation
GID = 9

Descript.	TYPE	PORT	DP	OS
FID / RID	00	01	02	03
00				

IPORTATTR
State of input impulse lines
GID = 101

Descript.	IN_0-7	IN_8-15
FID / RID	00	01
00		

OPORTATTR
State of output impulse lines
GID = 102

Descript.	OUT_0-7
FID / RID	00
00	

IMPULSEBARRIERE
Parameter of impulse delay function
GID = 106

Descript.	BT	AT
FID / RID	00	01
00		
01		
02		

ALARMPARAM
Parameters of time alarms
GID = 195

Descript.	Y	M	D	DW	AD	H	MN	S	AT
FID / RID	00	01	02	03	04	05	06	07	08
00									
01									
02									
03									
04									
05									
06									
07									
08									
09									
10									
11									
12									
13									
14									
15									

ALARMATTR
Attribues of time alarms
GID = 95

Descript.	TA	NA
FID / RID	00	01
00		
01		
02		
03		
04		
05		
06		
07		
08		
09		
10		
11		
12		
13		
14		
15		

Figure 6-3 Table of registers - System (Part 2)

CHANNELPARAM
Parameters of serial channels
GID = 180

Descript.	T	F	BR	DB	P	SB	
FID / RID	00	01	02	03	04	05	
00							OPTICAL
01							RS 232/485
02							RS 232/485/MODEM

CASCADE
Parameters of CASCADE communication function
GID = 189

Descript.	STATUS	CHANNEL
FID / RID	00	01
00		

ATPARAM
AT modem parameters
GID = 181

Descript.	IS	IP	AT	RTA	ASTRH	ASTRM	ASTPH	ASTPM	PD	RC
FID / RID	00	01	02	03	04	05	06	07	08	09
00										

ATATTR
Attributes of AT modem
GID = 81

Descript.	LI	LD	RT	RC	DC	
FID / RID	00	01	02	03	04	
00						OPTICAL
01						RS 232/485
02						RS 232/485/MODEM

CALLSTATION
Parameters of stations
GID = 185

Descript.	ST	SID	SP	CT	CI
FID / RID	00	01	02	03	04
00					
01					
02					
03					
04					
05					
06					
07					

TELNUMPARAM
Station telephone numbers
GID = 186

Descript.	TN
FID / RID	00
00	
01	
02	
03	
04	
05	
06	
07	

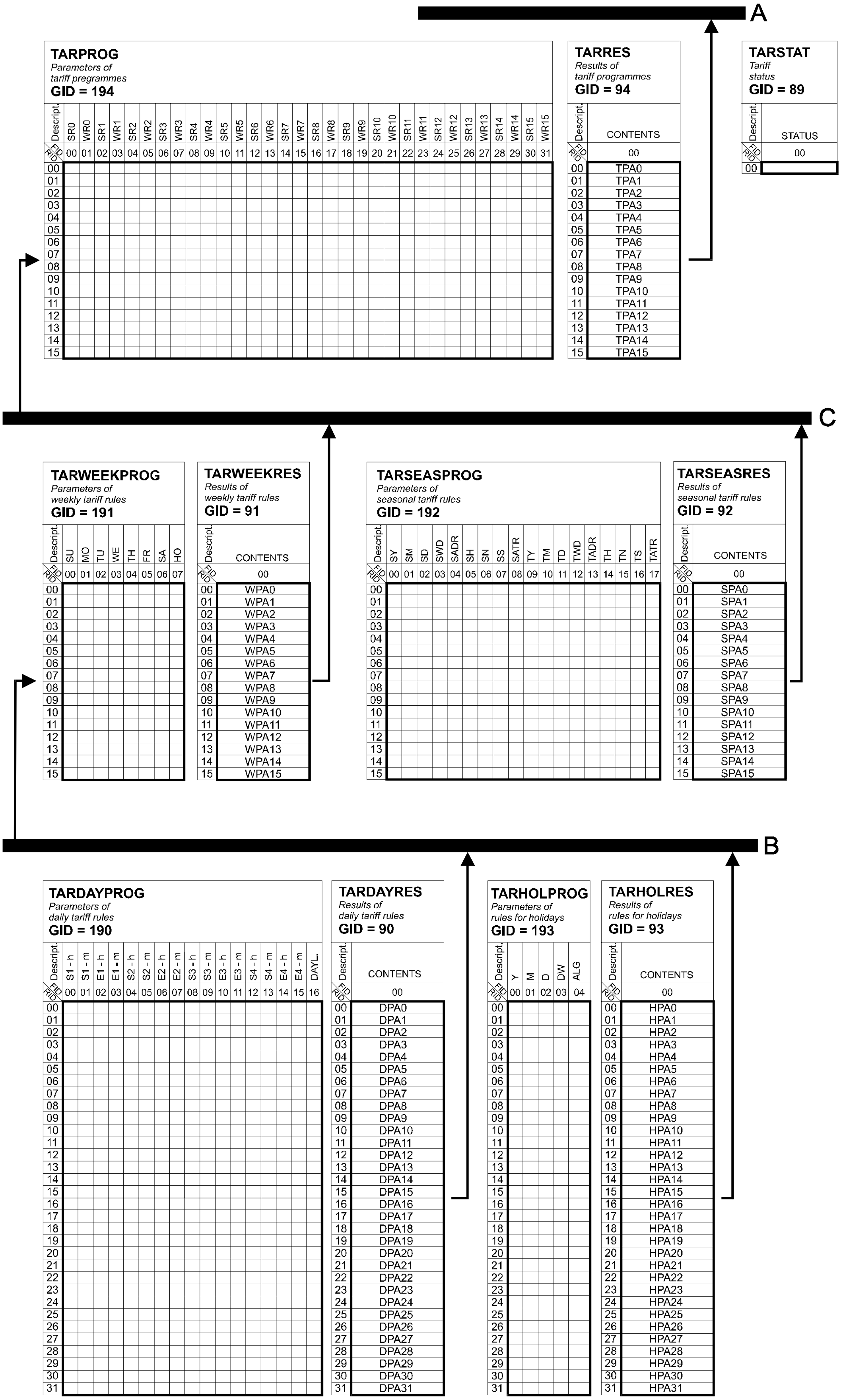
CALLPARAM
Parameters for call via a modem
GID = 187

Descript.	CLT	CLC	CLS	CLE	CLEI	CS	CSI
FID / RID	00	01	02	03	04	05	06
00							
01							
02							
03							
04							
05							
06							
07							

CALLATTR
Attributes of calls
GID = 87

Descript.	ST	FL	LS	LC	LCI
FID / RID	00	01	02	03	04
00					
01					
02					
03					
04					
05					
06					
07					

Figure 6-4 Table of registers - Communication



TARPROG
Parameters of
tariff pregrammes
GID = 194

Descript.	SR0	WR0	SR1	WR1	SR2	WR2	SR3	WR3	SR4	WR4	SR5	WR5	SR6	WR6	SR7	WR7	SR8	WR8	SR9	WR9	SR10	WR10	SR11	WR11	SR12	WR12	SR13	WR13	SR14	WR14	SR15	WR15
FID RID	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
00																																
01																																
02																																
03																																
04																																
05																																
06																																
07																																
08																																
09																																
10																																
11																																
12																																
13																																
14																																
15																																

TARRES
Results of
tariff programmes
GID = 94

Descript.	CONTENTS
FID RID	00
00	TPA0
01	TPA1
02	TPA2
03	TPA3
04	TPA4
05	TPA5
06	TPA6
07	TPA7
08	TPA8
09	TPA9
10	TPA10
11	TPA11
12	TPA12
13	TPA13
14	TPA14
15	TPA15

TARSTAT
Tariff
status
GID = 89

Descript.	STATUS
FID RID	00
00	

TARWEEKPROG
Parameters of
weekly tariff rules
GID = 191

Descript.	SU	MO	TU	WE	TH	FR	SA	HO
FID RID	00	01	02	03	04	05	06	07
00								
01								
02								
03								
04								
05								
06								
07								
08								
09								
10								
11								
12								
13								
14								
15								

TARWEEKRES
Results of
weekly tariff rules
GID = 91

Descript.	CONTENTS
FID RID	00
00	WPA0
01	WPA1
02	WPA2
03	WPA3
04	WPA4
05	WPA5
06	WPA6
07	WPA7
08	WPA8
09	WPA9
10	WPA10
11	WPA11
12	WPA12
13	WPA13
14	WPA14
15	WPA15

TARSEASPROG
Parameters of
seasonal tariff rules
GID = 192

Descript.	SY	SM	SD	SWD	SADR	SH	SN	SS	SATR	TY	TM	TD	TWD	TADR	TH	TN	TS	TATR
FID RID	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17
00																		
01																		
02																		
03																		
04																		
05																		
06																		
07																		
08																		
09																		
10																		
11																		
12																		
13																		
14																		
15																		

TARSEASRES
Results of
seasonal tariff rules
GID = 92

Descript.	CONTENTS
FID RID	00
00	SPA0
01	SPA1
02	SPA2
03	SPA3
04	SPA4
05	SPA5
06	SPA6
07	SPA7
08	SPA8
09	SPA9
10	SPA10
11	SPA11
12	SPA12
13	SPA13
14	SPA14
15	SPA15

TARDAYPROG
Parameters of
daily tariff rules
GID = 190

Descript.	S1 - h	S1 - m	E1 - h	E1 - m	S2 - h	S2 - m	E2 - h	E2 - m	S3 - h	S3 - m	E3 - h	E3 - m	S4 - h	S4 - m	E4 - h	E4 - m	DAYL.
FID RID	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
00																	
01																	
02																	
03																	
04																	
05																	
06																	
07																	
08																	
09																	
10																	
11																	
12																	
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	
29																	
30																	
31																	

TARDAYRES
Results of
daily tariff rules
GID = 90

Descript.	CONTENTS
FID RID	00
00	DPA0
01	DPA1
02	DPA2
03	DPA3
04	DPA4
05	DPA5
06	DPA6
07	DPA7
08	DPA8
09	DPA9
10	DPA10
11	DPA11
12	DPA12
13	DPA13
14	DPA14
15	DPA15
16	DPA16
17	DPA17
18	DPA18
19	DPA19
20	DPA20
21	DPA21
22	DPA22
23	DPA23
24	DPA24
25	DPA25
26	DPA26
27	DPA27
28	DPA28
29	DPA29
30	DPA30
31	DPA31

TARHOLPROG
Parameters of
rules for holidays
GID = 193

Descript.	Y	M	D	DW	ALG
FID RID	00	01	02	03	04
00					
01					
02					
03					
04					
05					
06					
07					
08					
09					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					
31					

TARHOLRES
Results of
rules for holidays
GID = 93

Descript.	CONTENTS
FID RID	00
00	HPA0
01	HPA1
02	HPA2
03	HPA3
04	HPA4
05	HPA5
06	HPA6
07	HPA7
08	HPA8
09	HPA9
10	HPA10
11	HPA11
12	HPA12
13	HPA13
14	HPA14
15	HPA15
16	HPA16
17	HPA17
18	HPA18
19	HPA19
20	HPA20
21	HPA21
22	HPA22
23	HPA23
24	HPA24
25	HPA25
26	HPA26
27	HPA27
28	HPA28
29	HPA29
30	HPA30
31	HPA31

Figure 6-5 Table of registers - Tariff program

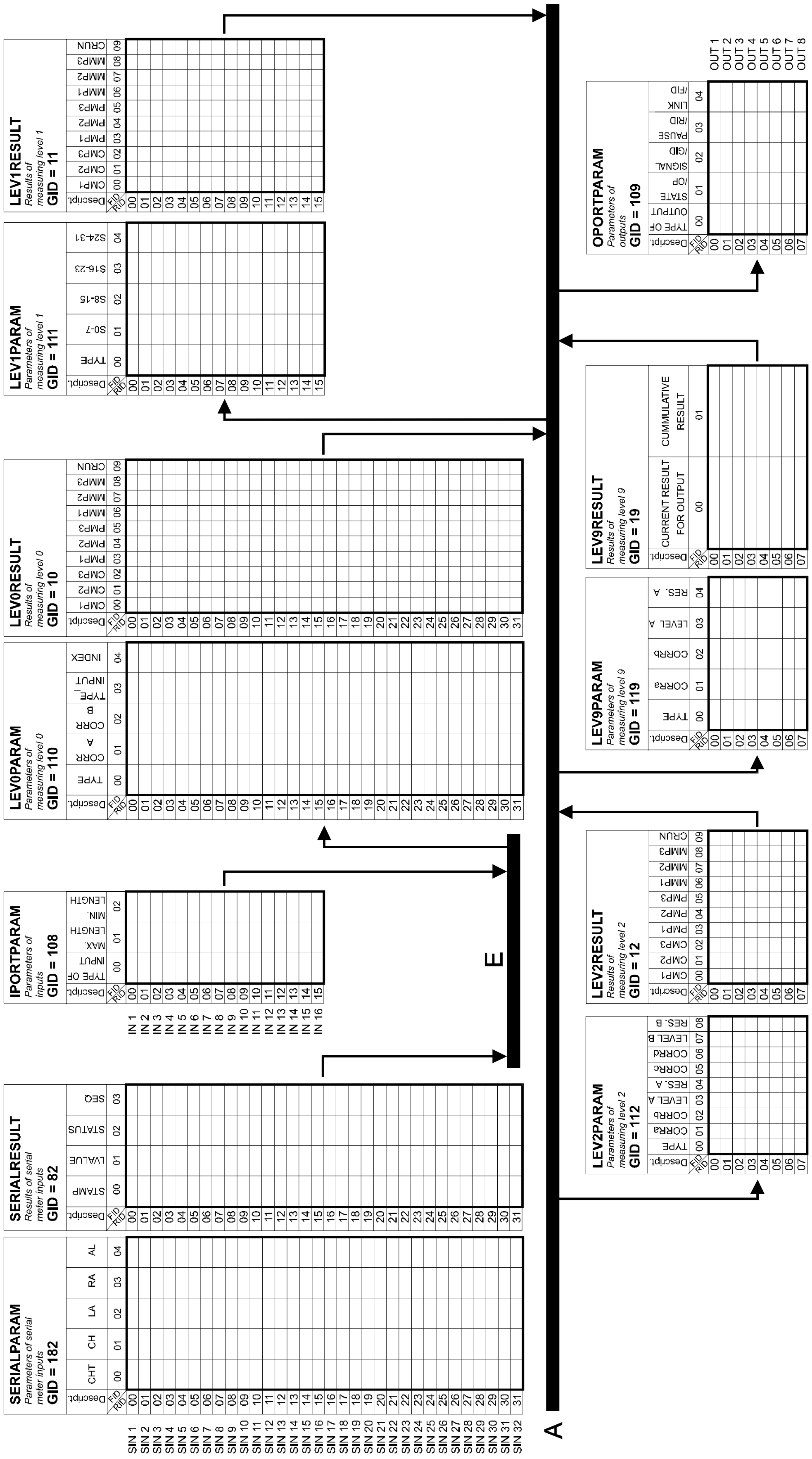


Figure 6-6 Table of registers - Measuring results

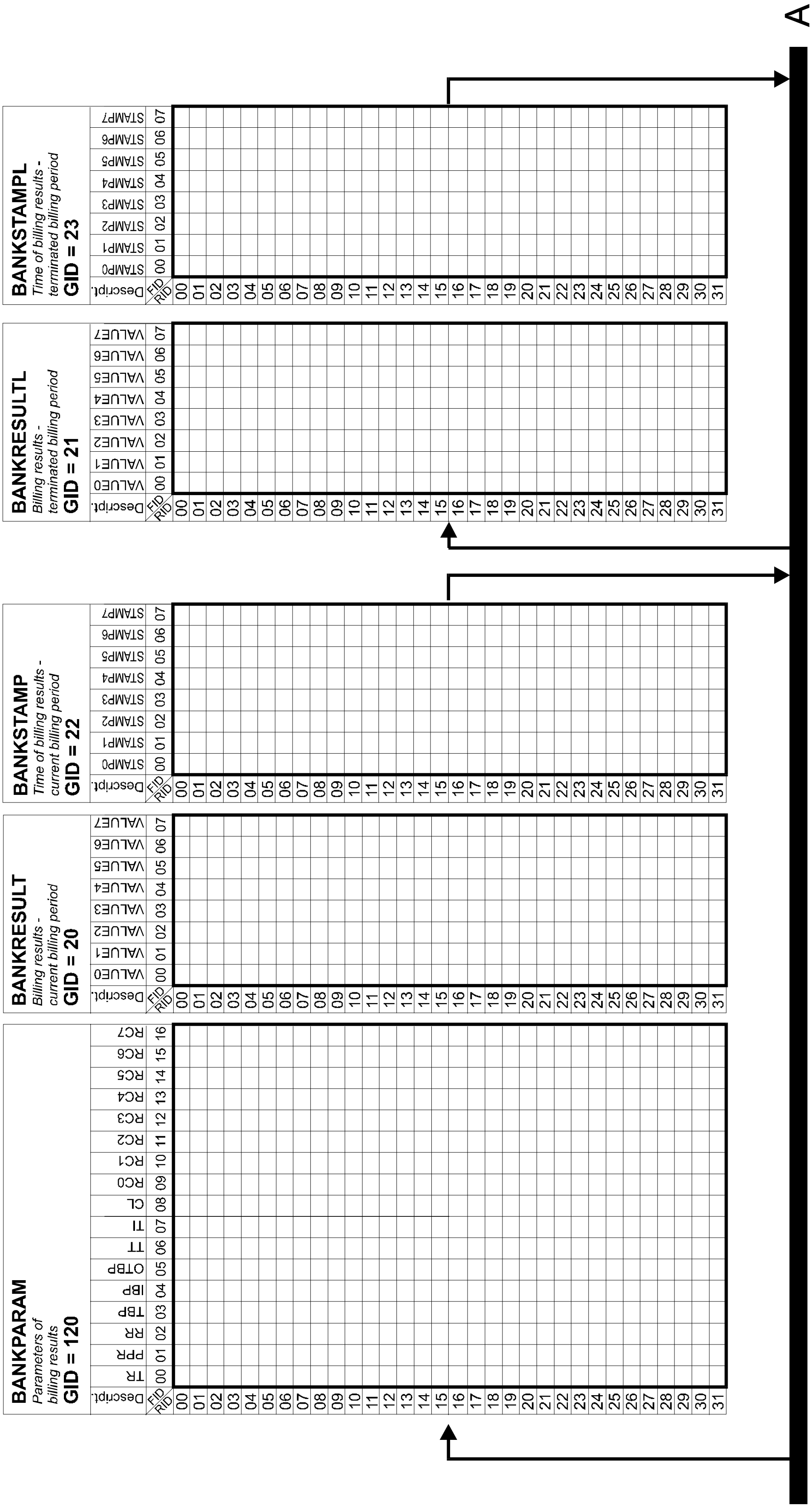
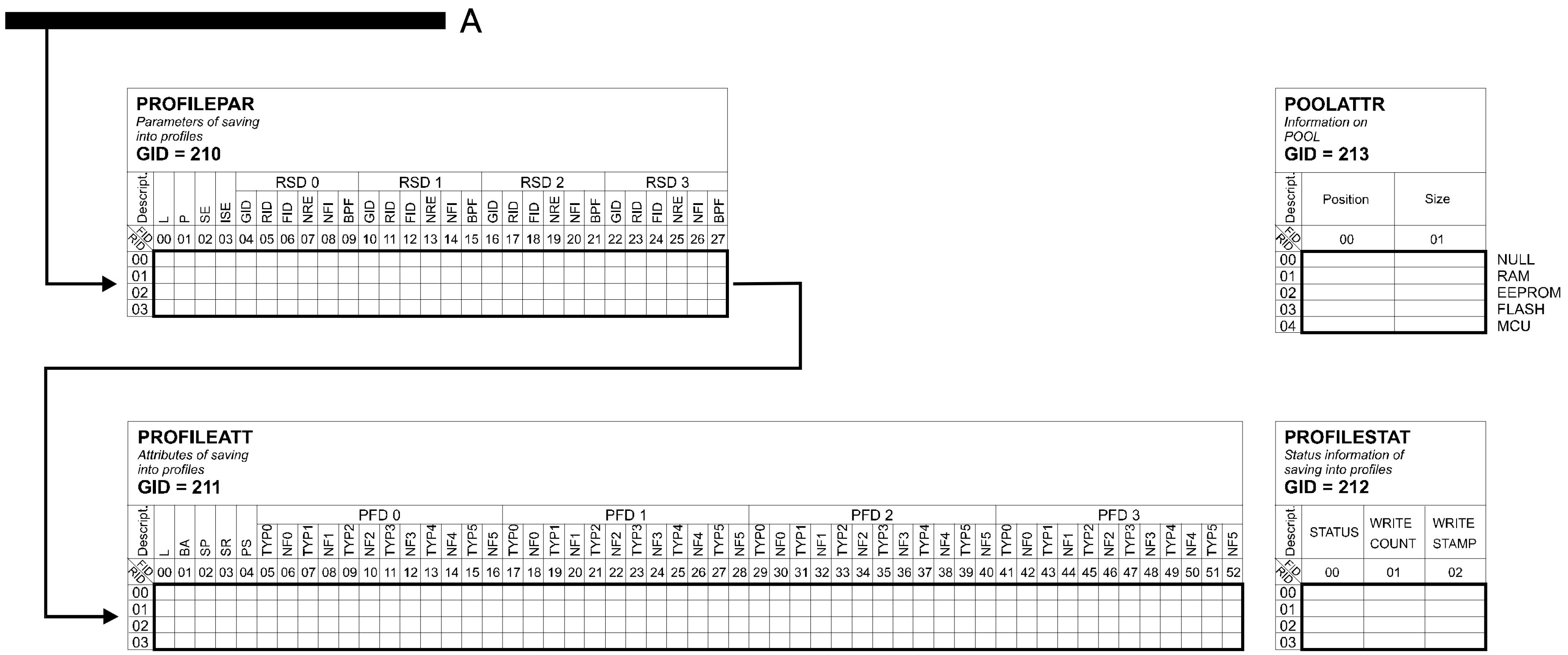


Figure 6-7 Table of registers - Billing results



PROFILEPAR
Parameters of saving into profiles
GID = 210

Descript.	L	P	SE	ISE	RSD 0						RSD 1						RSD 2						RSD 3					
					GID	RID	FID	NRE	NFI	BPF	GID	RID	FID	NRE	NFI	BPF	GID	RID	FID	NRE	NFI	BPF	GID	RID	FID	NRE	NFI	BPF
FID / RID	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
00																												
01																												
02																												
03																												

POOLATTR
Information on POOL
GID = 213

Descript.	Position	Size	
FID / RID	00	01	
00			NULL
01			RAM
02			EEPROM
03			FLASH
04			MCU

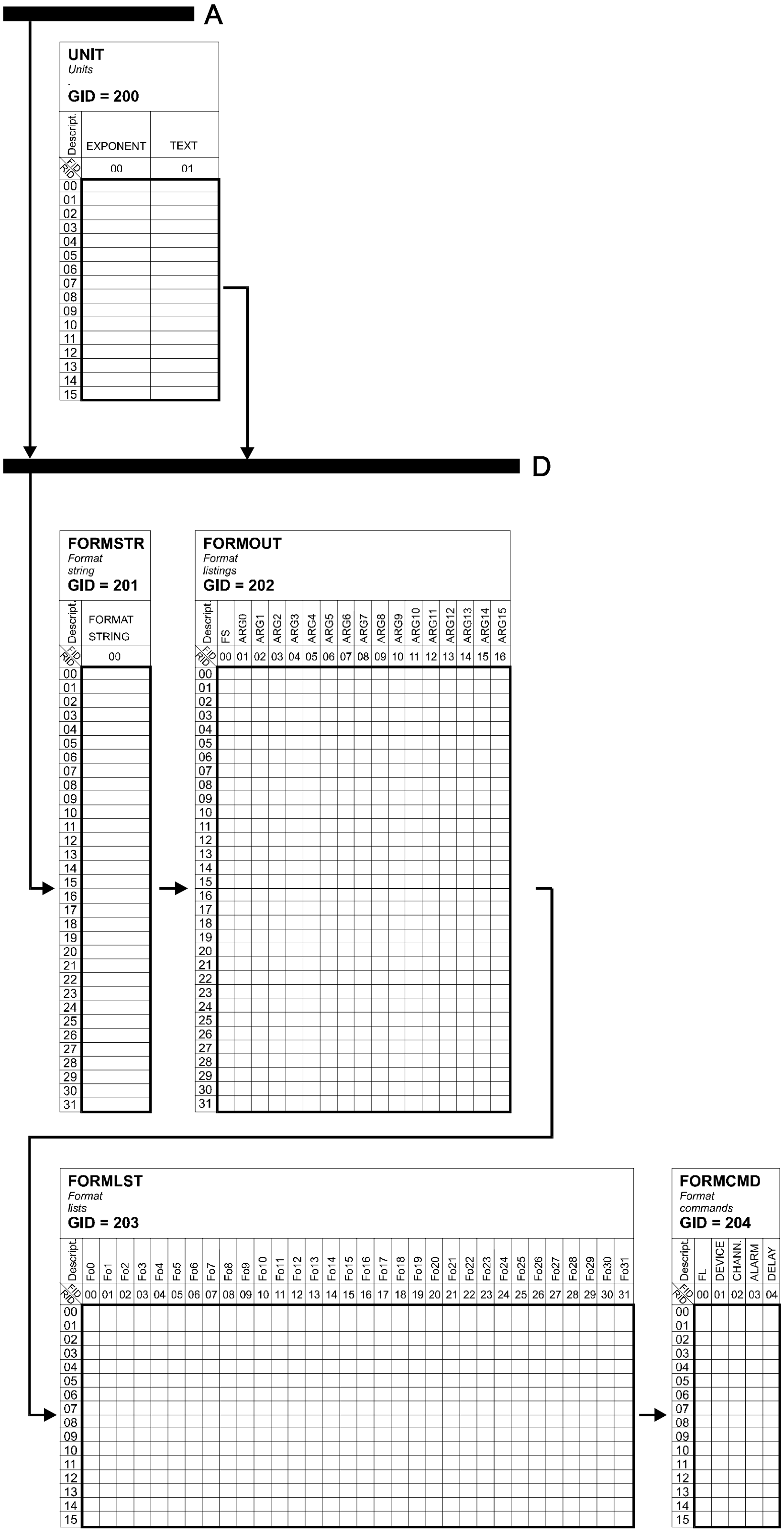
PROFILEATT
Attributes of saving into profiles
GID = 211

Descript.	L	BA	SP	SR	PS	PFD 0												PFD 1												PFD 2												PFD 3											
						TYP0	NF0	TYP1	NF1	TYP2	NF2	TYP3	NF3	TYP4	NF4	TYP5	NF5	TYP0	NF0	TYP1	NF1	TYP2	NF2	TYP3	NF3	TYP4	NF4	TYP5	NF5	TYP0	NF0	TYP1	NF1	TYP2	NF2	TYP3	NF3	TYP4	NF4	TYP5	NF5	TYP0	NF0	TYP1	NF1	TYP2	NF2	TYP3	NF3	TYP4	NF4	TYP5	NF5
FID / RID	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
00																																																					
01																																																					
02																																																					
03																																																					

PROFILESTAT
Status information of saving into profiles
GID = 212

Descript.	STATUS	WRITE COUNT	WRITE STAMP
FID / RID	00	01	02
00			
01			
02			
03			

Figure 6-8 Table of registers - Saving



UNIT
Units
.
GID = 200

Descript.	EXPONENT	TEXT
FID RID	00	01
00		
01		
02		
03		
04		
05		
06		
07		
08		
09		
10		
11		
12		
13		
14		
15		

FORMSTR
Format string
GID = 201

Descript.	FORMAT STRING
FID RID	00
00	
01	
02	
03	
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	

FORMOUT
Format listings
GID = 202

Descript.	FS	ARG0	ARG1	ARG2	ARG3	ARG4	ARG5	ARG6	ARG7	ARG8	ARG9	ARG10	ARG11	ARG12	ARG13	ARG14	ARG15
FID RID	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
00																	
01																	
02																	
03																	
04																	
05																	
06																	
07																	
08																	
09																	
10																	
11																	
12																	
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	
29																	
30																	
31																	

FORMLST
Format lists
GID = 203

Descript.	Fo0	Fo1	Fo2	Fo3	Fo4	Fo5	Fo6	Fo7	Fo8	Fo9	Fo10	Fo11	Fo12	Fo13	Fo14	Fo15	Fo16	Fo17	Fo18	Fo19	Fo20	Fo21	Fo22	Fo23	Fo24	Fo25	Fo26	Fo27	Fo28	Fo29	Fo30	Fo31
FID RID	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
00																																
01																																
02																																
03																																
04																																
05																																
06																																
07																																
08																																
09																																
10																																
11																																
12																																
13																																
14																																
15																																

FORMCMD
Format commands
GID = 204

Descript.	FL	DEVICE	CHANN.	ALARM	DELAY
FID RID	00	01	02	03	04
00					
01					
02					
03					
04					
05					
06					
07					
08					
09					
10					
11					
12					
13					
14					
15					

Figure 6-9 Table of registers - Format listings

7. INDEX

Figure:

Table:

Owing to periodical improvements of our products the supplied products can differ in some details from data stated in the document.

Iskraemeco, Energy Measurement and Management
4000 Kranj, Savska loka 4, Slovenia
Telephone: (+386 64) 26 40, Telefax: (+386 64) 264 376
Published by Iskraemeco, Marketing
Data subject to alternation without notice.