

# COMMUNICATOR P2S

## Technical Description





**CONTENTS**

**1. INTRODUCTION ..... 7**

**2. COMMUNICATOR DESIGN ..... 8**

2.1 CENTRAL PROCESSOR UNIT (CPU) ..... 9

2.2 BASIC BOARD ..... 10

    2.2.1 *Input/output interface* ..... 10

        2.2.1.1 CS inputs ..... 10

        2.2.1.2 CS outputs ..... 10

        2.2.1.3 CS communication ..... 10

    2.2.2 *Relay outputs* ..... 11

    2.2.3 *Power supply* ..... 11

    2.2.4 *RS485 communication interface* ..... 11

2.3 MODEM 1 ..... 12

2.4 MODEM 2 ..... 12

    2.4.1 *PSTN Modem* ..... 12

    2.4.2 *RS232C interface* ..... 13

    2.4.3 *ISDN terminalni adapter* ..... 13

**3. COMMUNICATOR FUNCTIONS ..... 14**

3.1 INPUTS ..... 15

    3.1.1 *Checking input data* ..... 15

    3.1.2 *Function of impulse retaining* ..... 16

    3.1.3 *Serial meter inputs* ..... 17

3.2 OUTPUTS ..... 18

    3.2.1 *Impulse outputs* ..... 18

    3.2.2 *Control outputs* ..... 18

3.3 TIME FUNCTIONS ..... 19

    3.3.1 *Real time control* ..... 19

    3.3.2 *Real time program clock* ..... 19

    3.3.3 *Internal Real Time Clock (RTC circuit)* ..... 19

    3.3.4 *DCF77 (option)* ..... 19

    3.3.5 *Time functions of program clock* ..... 19

        3.3.5.1 *Measuring periods* ..... 19

        3.3.5.2 *Time window of pulse retaining* ..... 19

        3.3.5.3 *Repeatable time alarms* ..... 20

        3.3.5.4 *Billing periods* ..... 21

        3.3.5.5 *Data saving periods* ..... 21

        3.3.5.6 *Control of internal tariff program* ..... 21

3.4 MEASURING RESULTS ..... 22

    3.4.1 *Adaptation of values of input meter impulses - measuring level 0* ..... 25

    3.4.2 *Calculation of positive sums - measuring level 1* ..... 27

    3.4.3 *Calculation of functions - measuring level 2* ..... 28

    3.4.4 *Output results - measuring level 9* ..... 30

3.5 BILLING RESULTS ..... 31

3.6 TARIFF PROGRAM ..... 33

3.7 REGISTERS ..... 40

3.8 FORMAT LISTINGS ..... 42

    3.8.1 *General* ..... 42

    3.8.2 *Format Strings* ..... 42

    3.8.3 *Format Outputs* ..... 50

    3.8.4 *Format Lists* ..... 51

    3.8.5 *Format Commands* ..... 51

    3.8.6 *Units* ..... 52

3.9 SAVING ..... 53

3.10 COMMUNICATION ..... 56

    3.10.1 *General* ..... 56

    3.10.2 *Types of channels* ..... 56

3.10.3 Communication functions .....	56
3.10.3.1 Terminal .....	56
3.10.3.2 Printer .....	63
3.10.3.3 DIN19244 Master .....	63
3.10.3.4 DIN 19244 Slave .....	64
3.10.3.5 IEEE802.3 Slave is not realized yet .....	75
3.10.3.6 IEC 1107 Master .....	76
3.10.3.7 Cascade .....	79
3.10.4 Modem communication .....	81
3.10.5 Communication through the DLC modem .....	83
3.11 CONSOLE .....	84
3.11.1 General .....	84
3.11.2 Keyboard .....	84
3.11.3 Display .....	85
3.11.4 Local manipulation .....	85
3.11.5 LED's .....	95
3.12 SYSTEM MESSAGES .....	96
<b>4. GRUOPS OF REGISTERS .....</b>	<b>97</b>
4.1 SYSTEMATTR .....	100
4.2 DEVICEID .....	103
4.3 USERS .....	104
4.4 DATETIME .....	105
4.5 TIMEATTR .....	106
4.6 PERFORM .....	108
4.7 MEASPER .....	109
4.8 DAYLIGHT CHANGE PARAMETERS .....	110
4.9 EVENTS .....	111
4.10 TIMESYNCHRO .....	114
4.11 LEV0RESULT .....	115
4.12 LEV1RESULT .....	116
4.13 LEV2RESULT .....	117
4.14 LEV9RESULT .....	118
4.15 BANKRESULT .....	119
4.16 BANKRESULTL .....	120
4.17 BANKSTAMP .....	121
4.18 BANKSTAMPL .....	122
4.19 ATATTR .....	123
4.20 SERIALRESULT .....	124
4.21 IEC1107MREADATTR .....	126
4.22 IEC1107MRESETATTR .....	127
4.23 CALLATTR .....	128
4.24 DLCATTR .....	129
4.25 TARSTAT .....	131
4.26 TARDAYRES .....	132
4.27 TARWEEKRES .....	133
4.28 TARSEASRES .....	134
4.29 TARHOLRES .....	135
4.30 TARRES .....	136
4.31 ALARMATTR .....	137
4.32 IPORTATTR .....	138
4.33 OPORTATTR .....	139
4.34 IMPULSEBARRIERE .....	140
4.35 CONFIGPORT .....	141
4.36 IPORTPARAM .....	142
4.37 OPORTPARAM .....	143
4.38 LEV0PARAM .....	144
4.39 LEV1PARAM .....	146
4.40 LEV2PARAM .....	148
4.41 LEV9PARAM .....	150

4.42 BANKPARAM.....	151
4.43 CHANNELPARAM.....	153
4.44 ATPARAM.....	155
4.45 SERIALPARAM.....	156
4.46 IEC1107MREADPARAM.....	158
4.47 IEC1107MRESETPARAM.....	160
4.48 CALLSTATION.....	161
4.49 TELNUMPARAM.....	162
4.50 CALLPARAM.....	163
4.51 DLCPARAM.....	165
4.52 CASCADE.....	166
4.53 TARDAYPROG.....	167
4.54 TARWEEKPROG.....	168
4.55 TARSEASPROG.....	169
4.56 TARHOLPROG.....	171
4.57 TARPROG.....	172
4.58 ALARMPARAM.....	173
4.59 UNIT.....	175
4.60 FORMSTR.....	176
4.61 FORMOUT.....	177
4.62 FORMLST.....	178
4.63 FORMCMD.....	179
4.64 PROFILEPAR.....	180
4.65 PROFILEATT.....	181
4.66 PROFILESTAT.....	182
4.67 POOLATTR.....	183
<b>5. TECHNICAL DATA.....</b>	<b>184</b>
5.1 CS LINES.....	184
5.1.1 CS input lines.....	184
5.1.1.1 Synchronization input.....	185
5.1.2 CS output lines.....	186
5.1.3 CS communication lines.....	186
5.2 RELAY OUTPUTS.....	186
5.3 POWER SUPPLY.....	187
5.3.1 Main supply.....	187
5.3.2 Internal stand-by power supply.....	188
5.3.3 Supply of CS lines.....	188
5.4 COMMUNICATION.....	189
5.4.1 RS485.....	189
5.4.2 RS232C.....	189
5.4.3 PSTN modem.....	190
5.4.4 DLC Modem.....	190
5.4.5 Optical interface.....	190
5.4.6 ISDN terminal adapter.....	191
5.5 INTERNAL CLOCK.....	192
5.5.1 Internal time standard.....	192
5.6 MEMORY.....	192
5.7 LOCAL DISPLAY.....	192
5.8 OPERATION CONDITIONS.....	192
5.8.1 Ambient temperature.....	192
5.8.2 Air humidity.....	192
5.8.3 Dust protection.....	192
5.9 DIELECTRIC STRENGTH AND RESISTANCE TO INTERFERENCES.....	193
5.9.1 Insulation strength.....	193
5.9.2 Resistance to voltage surges 1,2/50.....	193
5.9.3 High frequency disturbance test.....	194
5.9.4 Resistance to fast voltage surges - burst.....	194
5.9.5 High frequency electromagnetic field.....	194

5.9.6 <i>Electrostatic discharges</i> .....	194
5.10 LEVEL OF HARMONIC DISTURBANCES .....	194
5.11 DEGREE OF RF INTERFERENCES .....	194
5.12 DIMENSIONS.....	195
5.12.1 <i>Fixing dimensions</i> .....	195
5.12.2 <i>Mass</i> .....	195
5.13 CONNECTION DIAGRAM .....	196
<b>6. TABLES OF REGISTERS .....</b>	<b>197</b>
<b>7. INDEX .....</b>	<b>207</b>

## 1. INTRODUCTION

The P2S communicator represents, on one hand, a data recorder from POREG2 generation, and on the other hand, a communicator for remote data collection on electric energy consumption via various communication media and from different measuring devices. Primary communication path is a public telephone network, and primary measuring devices are electricity meters.

The P2S communicator 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 P2 communicator is intended for building into measuring sets of contractual consumers in medium-voltage network. Measuring equipment of these consumers can consist of precision electricity meters with impulse outputs or STOM outputs, induction meters without or with a tariff device as well as static meters without or with a tariff device.

The application field also involves the field of the so-called modems for transmission of meter values. These modems enable remote data collection from the electricity meters in compliance with IEC 1107 standard.

## 2. COMMUNICATOR DESIGN

The Communicator P2S is based on a modular construction and in this way enables various combinations:

- CPU with optical communication (IR) and LCD
- basic board

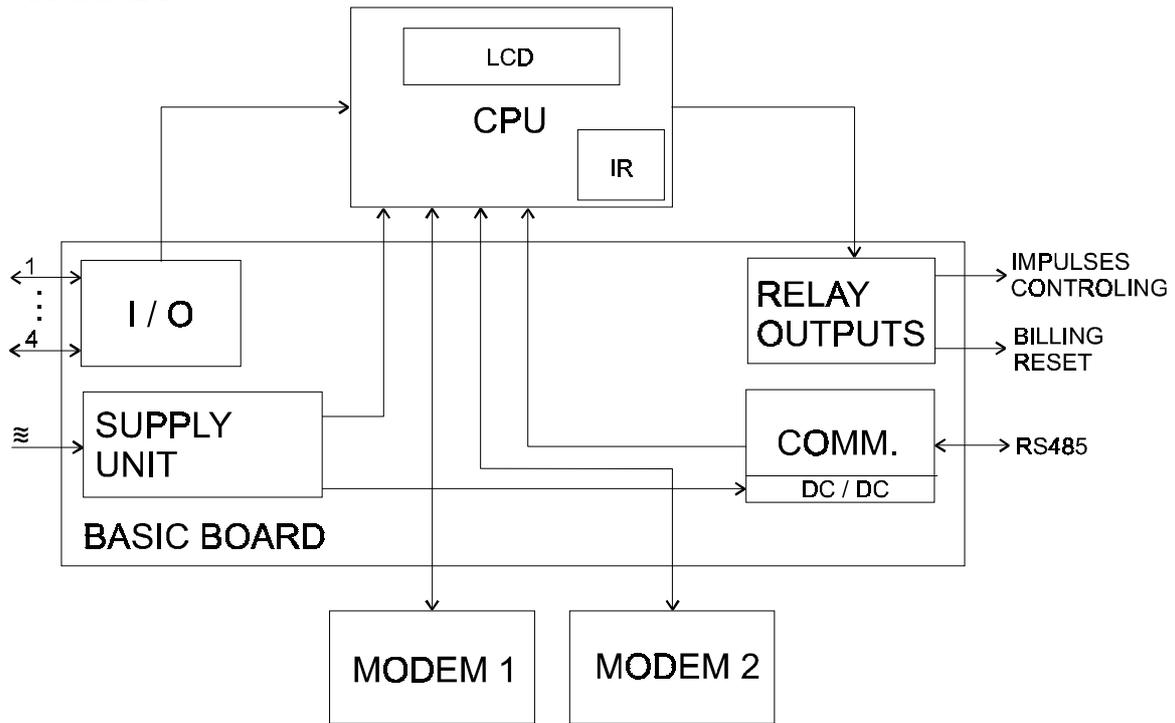


Figure 2-1 Communicator design

Figure 2-1 shows completely equipped communicator. Two modules are added to the basic equipment, i.e. MODEM 1 which is intended for DLC Modem and MODEM 2 which is intended for PSTN Modem, ISDN terminal adapter or RS232C interface. A communicator can be equipped with both modules at the same time.

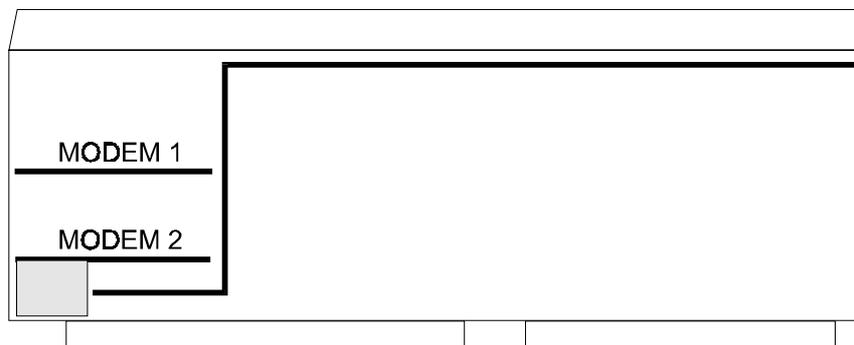


Figure 2-2 Position of modems

## 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 two 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).

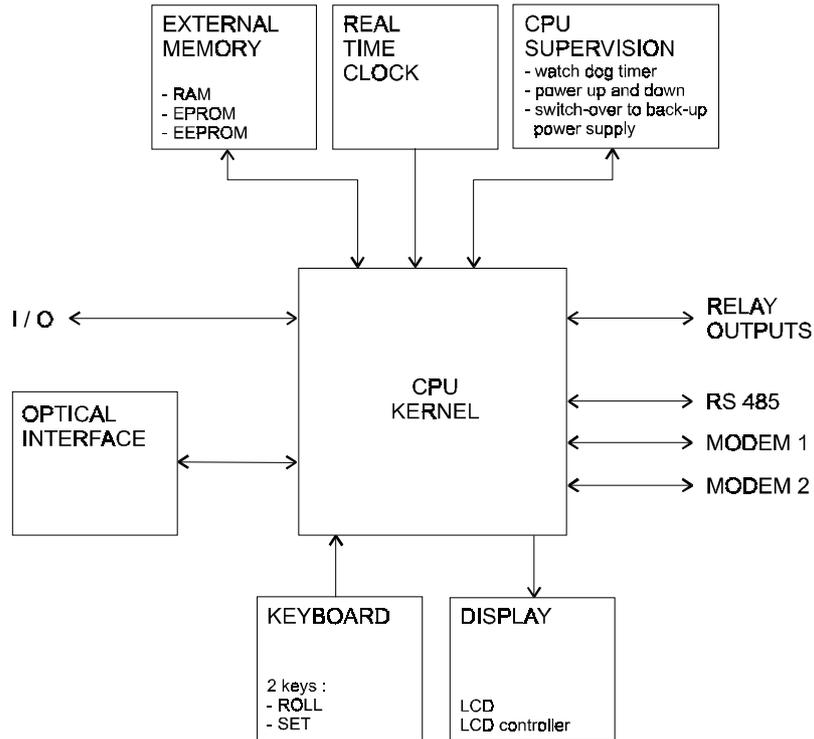


Figure 2-3 CPU

Settings on a CPU:

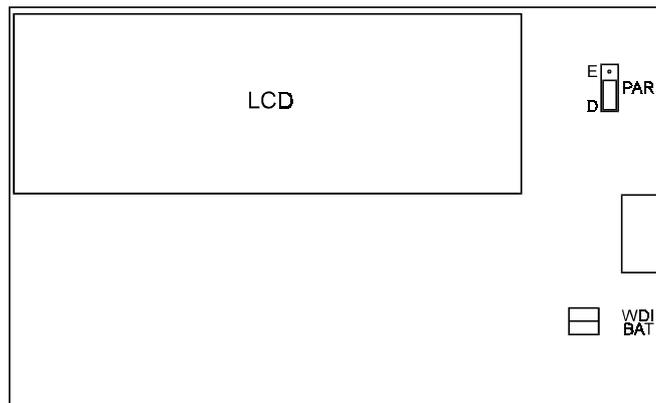


Figure 2-4 CPU - Settings

A jumper BAT is open when a device is in a warehouse; it interrupts a supply from back-up capacitor.  
 A jumper WDI is open only for the needs of servicing.  
 A jumper PAR in "E" position permits parametering of a device with parameters which influence in the results.

## 2.2 BASIC BOARD

The basic board connects individual communicator parts into a functional whole. It consists of input/output interface, relay outputs, supply unit and RS485 communication interface.

### 2.2.1 Input/output interface

The interface consists of four input/output lines of CS type which are electrically isolated from internal circuits. Lines can act as inputs, outputs or communication lines, depending of program setting. They are active which means that no external supply source is required.

#### 2.2.1.1 CS inputs

The input module involves inputs which are electrically isolated from internal circuits.

Regarding their functions the inputs are divided to control and impulse inputs. Control inputs are intended for the reception of control 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 four types:

- two-wire input in compliance with DIN 43864 (S0)
- two-wire input for active current (S1a)
- two-wire input for inactive current (S1b)
- three-wire input (S3)

is not realized yet

All inputs are programmable, which means that the function can be defined for each input. An input represents one two-conductor input of S0, S1a or S1r type. Two successive inputs can be united in a pair where one pair of inputs represents one three-conductor input (S3).

##### **Two-wire input in compliance with DIN 43864 (S0)**

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. Characteristics of input comply with DIN 43846 standard.

##### **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.

##### **Two-wire input for inactive current (S1r)**

During the active signal (impulse) the circuit on signal lines is open. 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.

##### **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.

#### 2.2.1.2 CS outputs

Outputs comply with DIN 43864 standard and are mainly intended for impulse outputs. However, they can be also used as control outputs by means of special adapters.

#### 2.2.1.3 CS communication

CS lines can be used for communication by the method of a current loop.

## 2.2.2 Relay outputs

The communicator is provided with two relay outputs: an impulse relay and a relay for a billing reset. The impulse relay is a semi-conductor type with an active contact, while the relay for a billing reset is a classical bistable relay with make and break contacts.

The impulse relay can perform the function of a control or impulse output. The control output is intended for control of external devices (alarms, tariff devices, etc.), and the impulse output for transmitting the results of data local processing (results of billing levels) to further processing.

The relay for a billing reset is intended for control of external devices. Its function is to reset billing results to initial values. It can also be used as an impulse output.

## 2.2.3 Power supply

It is intended to supply communicator 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. A supply unit is not provided with stand-by power supply in case of voltage failure.

The power supply module ensures voltages for:

- CPU
- basic board
- modem 1
- modem 2

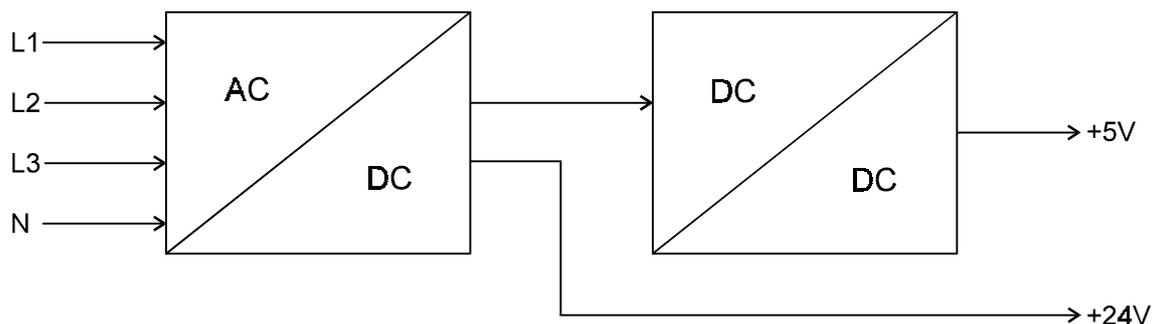


Figure 2-5 Power supply

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

## 2.2.4 RS485 communication interface

RS485 communication interface is built in a communicator and functions as a link between a communicator and communication equipment or devices. It takes over prepared data from the communicator (meter states with time marks, messages, etc.) in regular time intervals or on external request, and stores as well as transmits them along communication lines. It can also operate as input unit for serial inputs. External lines are electrically isolated from internal. DC/DC converter provides for corresponding voltages on external lines.

Transmission rate is set in stages from 150 to 57600 bit/s.

Characteristics of communication interfaces are stated in TECHNICAL DATA Chapter

### 2.3 MODEM 1

Modem 1 is intended for "DLC Modem". This is a communication interface which is intended for communication via low-voltage network.

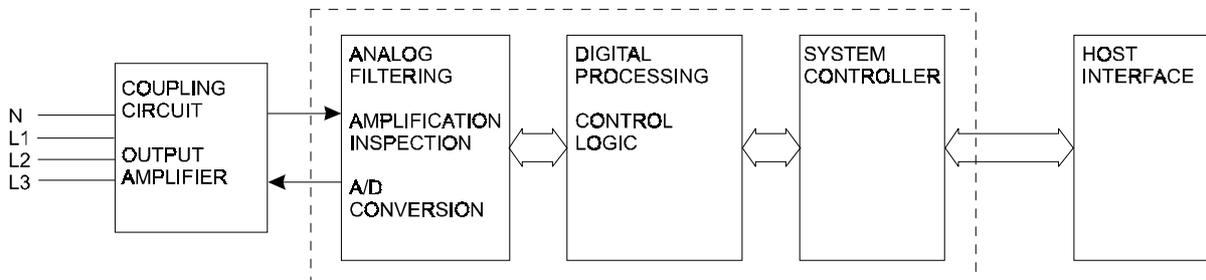


Figure 2-6 DLC modem

### 2.4 MODEM 2

Modem 2 is intended for communication. It can be either "PSTN Modem", "ISDN terminal adapter" or RS232C interface.

#### 2.4.1 PSTN Modem

It is intended for connection to telephone network. A service telephone can also be connected. A special network provides for inter-disconnection of a telephone and a modem.

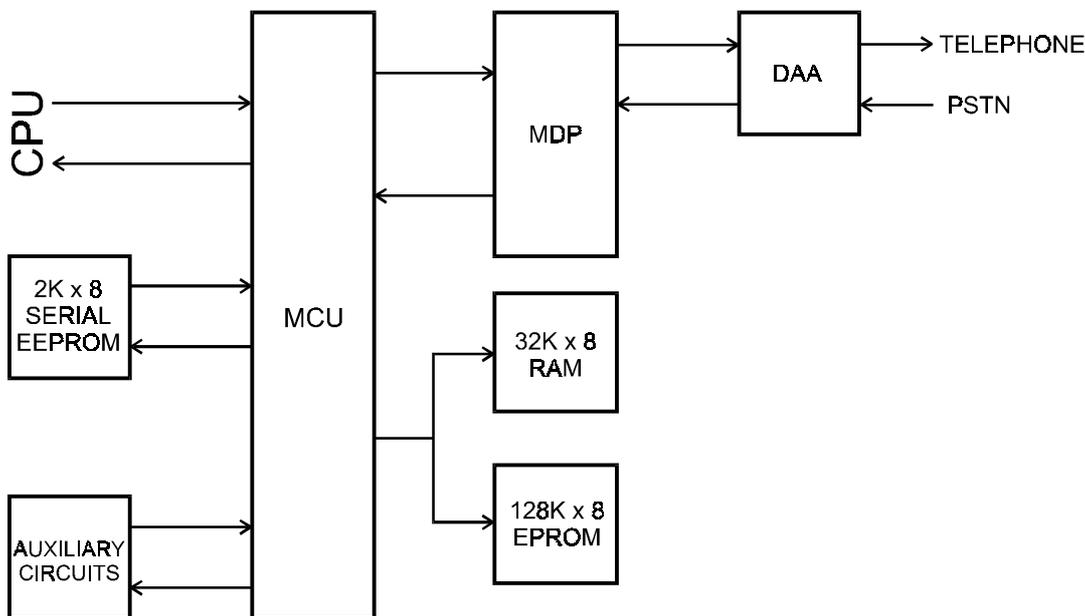


Figure 2-7 PSTN modem

The modem is program adjustable. A user adjustment and commands are stored in EEPROM, while general program settings of the modem operation are stored in EPROM.

### 2.4.2 RS232C interface

RS232C communication module is additional communicator equipment and functions as a link between the communicator and communication equipment or devices. It takes over prepared data from the communicator (meter states with time marks, messages, etc.) in regular time intervals or on external request, and stores as well as transmits them along communication lines. It can also operate as input unit for serial inputs. External lines are electrically isolated from internal. DC/DC converter provides for corresponding voltages on external lines.

The RS232C interface may be installed instead of the AT modem (item 2.4.1.) or instead of the ISDN terminal adapter (item 2.4.3.).

### 2.4.3 ISDN terminalni adapter

An ISDN terminal adapter (either TA+POI or TA+POCM) may be installed instead of the AT modem or the RS232C interface. It connects the communicator and the SO Basic Rate Interface.

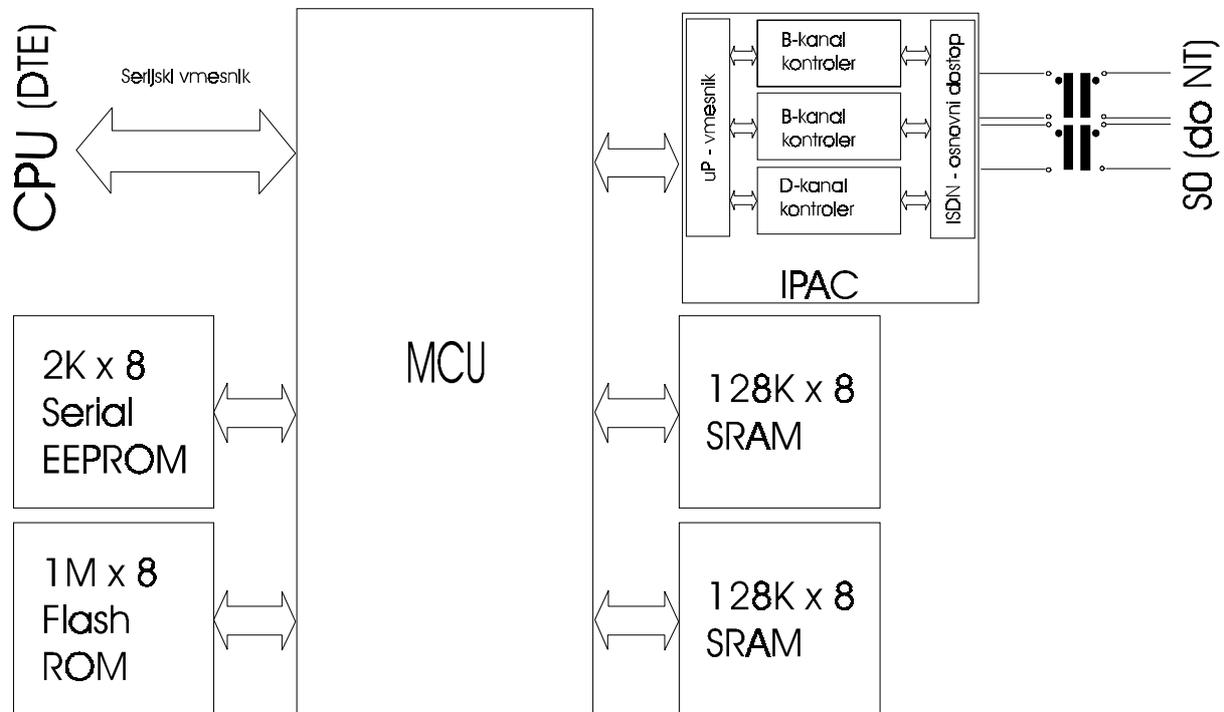


Figure 2-8 ISDN terminal adapter

The manufacturer has adjusted the terminal adapter to an automatic mode of operation.

### 3. COMMUNICATOR FUNCTIONS

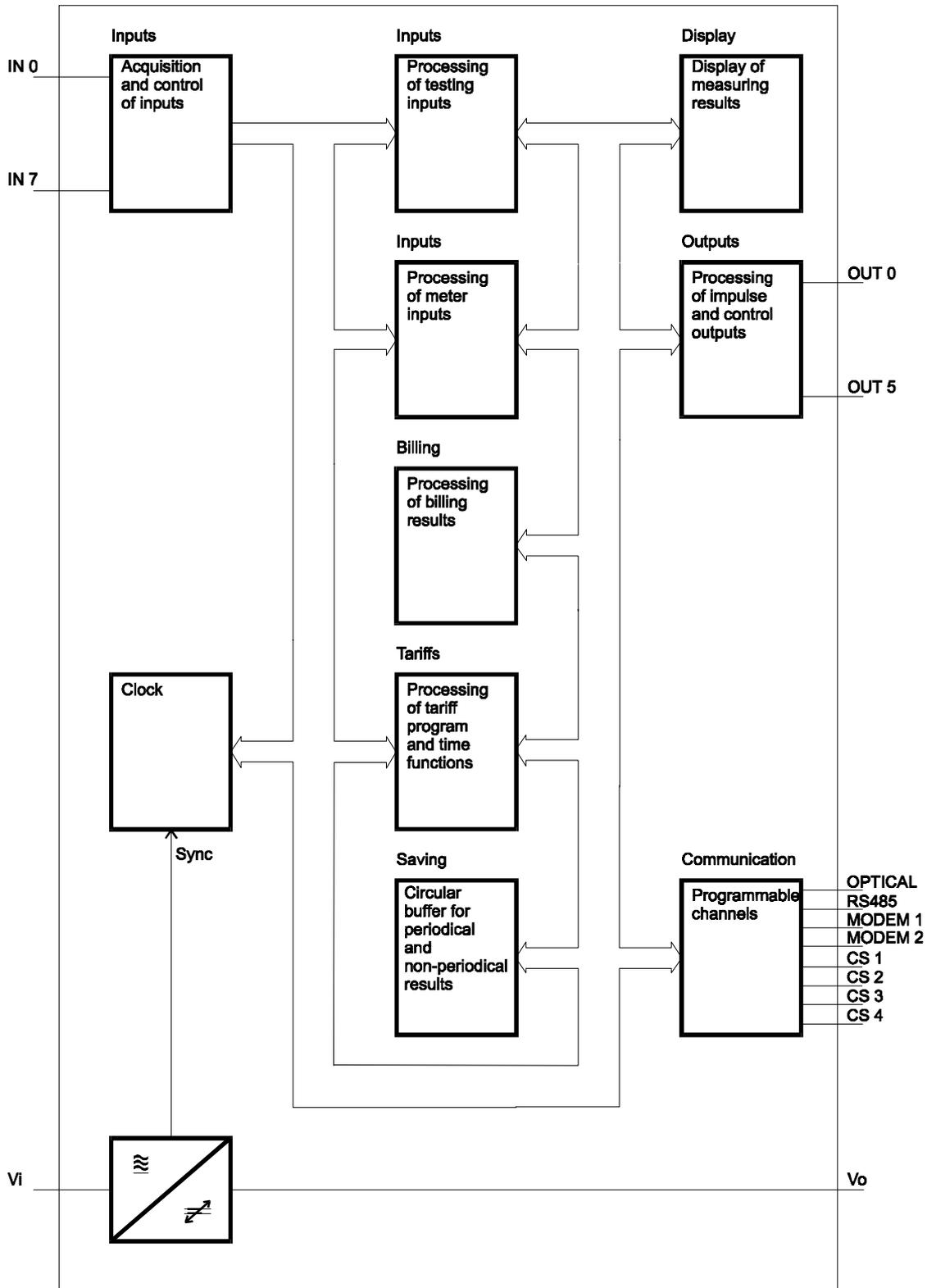


Figure 3-1 Block diagram of program functions

### 3.1 INPUTS

For correct processing of input impulses it is necessary to be familiar with their significance. Therefore the communicator 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
- control inputs

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

- S0 two-wire input in compliance with DIN 43864
- S1a two-wire input for active current
- S1r two-wire input for inactive current
- 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 ( $t \leq 10$  ms) and tested as described in Item CHECKING INPUT DATA. During the test the detected interferences activate alarm messages.

The type and control parameters of impulse inputs are set in the IPORTPARAM group of registers.

#### Serial meter inputs

The communicator is provided with serial communication inputs of meter values for meter read-out. Data on meter impulses are received via an RS485 serial interface in compliance with the communication protocol (DIN 19244) or via CS lines in compliance with IEC 1107 standard.

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.

#### Control inputs

They are intended for the reception of special signals. One of the following functions can 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 interferences and false impulses the communicator tests input meter impulses. The following is tested regarding the input type:

- for two-wire inputs

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 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 communicators regarding meter impulses. It is especially used at control measurements when impulses are transmitted from one source to two or more communicators. As the system time of communicators 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 communicator 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  $\pm 1$  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.

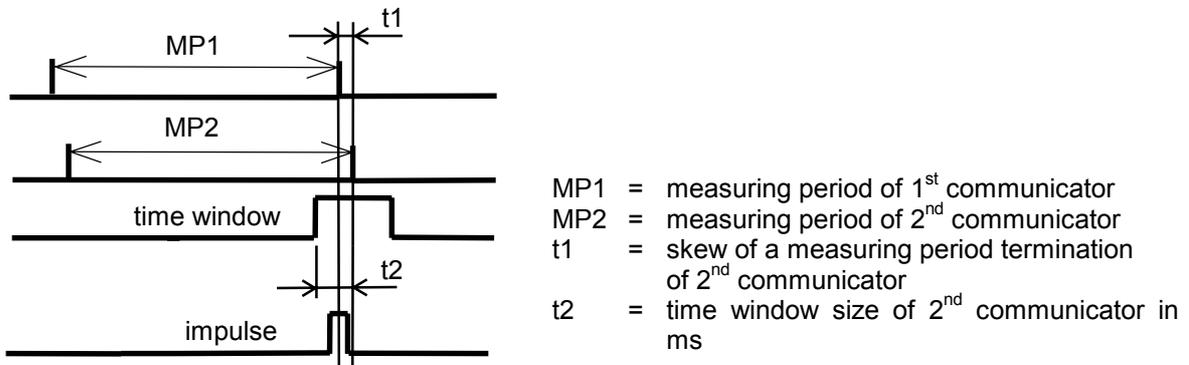
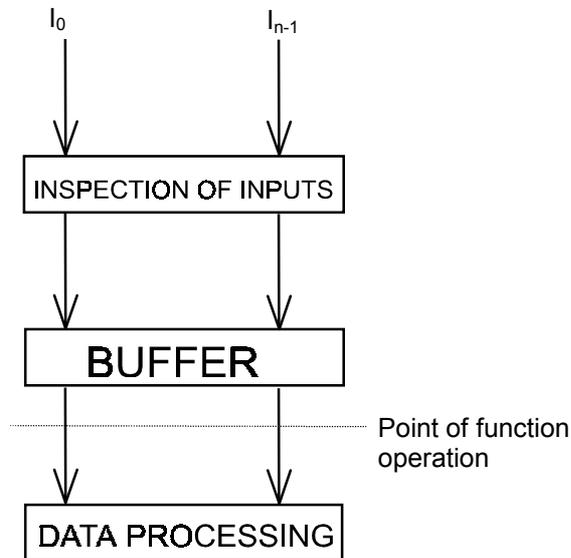


Figure 3-2 Function of pulse retaining

Figure 3-2 shows an example of impulse which occurred at MP1 termination. 1<sup>st</sup> communicator recognises the impulse as valid only in the next period and processes it. The second communicator 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 communicator, 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 communicators are equalised.



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

**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!**

Figure 3-3 Point of function operations

### 3.1.3 Serial meter inputs

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

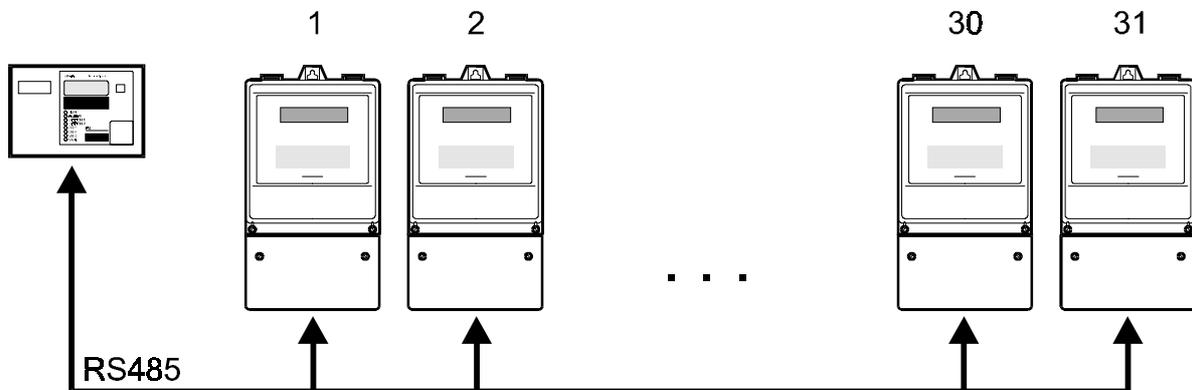


Figure 3-4 SDT net

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 communicator 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 communicator 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 the communicator. In the next step the communicator 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 depend on that setting. The setting data are in the meter documentation.

The communicator 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 meters reading are in the SERIALPARAM group of registers.

## 3.2 OUTPUTS

For a communication with connected devices the communicator is provided with outputs. Outputs transmit both impulse information and different messages. Transmission function is set for each input separately. Outputs are divided to passive (semi-conductor relay and classical bistable relay) and active outputs (CS lines).

### 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 communicator 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 communicator 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 communicator 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 generation

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 tariff devices

Outputs can generate a signal for resetting the meters with tariff devices. An output with relays with change-over contacts is intended for this; if not, two outputs with a relay with a single contact can be used instead.

A signal for resetting the meters can be repeated periodically or it is generated on the user's demand. Periodical repetition is connected to a time alarm. Parameters are set in a group of registers IEC1107MRESETPARAM. A user can reset the meters with a command for meter resetting which is then sent to the communicator. The command can be sent via a terminal protocol (BR command) or in compliance with DIN19244 protocol (ASDU 150 request).

When a time alarm occurs or when a demand for meter resetting is accepted, the values in a group of registers IEC1107MRESETATTR are changed. The value of the second field in the communicator is changed from 0 to 1 or vice-versa. The output relay follows this modification. The relay is determined in a group of registers OPORTPARAM.

### 3.3 TIME FUNCTIONS

#### 3.3.1 Real time control

Communicator 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 communicator. 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.

#### 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 capacitor supported in order to eliminate the possibility of losing the information on time and date in case of power failure on a communicator main power supply.

#### 3.3.4 DCF77 (option)

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 communicator 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 pulse 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.  $\pm 1$  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 communicator 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 communicator 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 communicator 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	1940..2099
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	yearlyeast	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 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 connected 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). Data, which are available as registers in a communicator, 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.

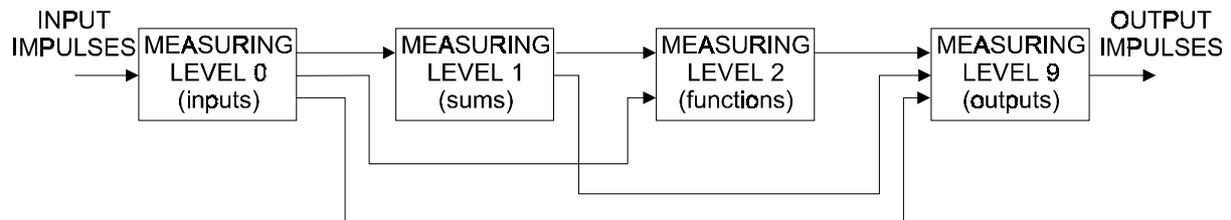


Figure 3-5 Transmission of measuring results

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!

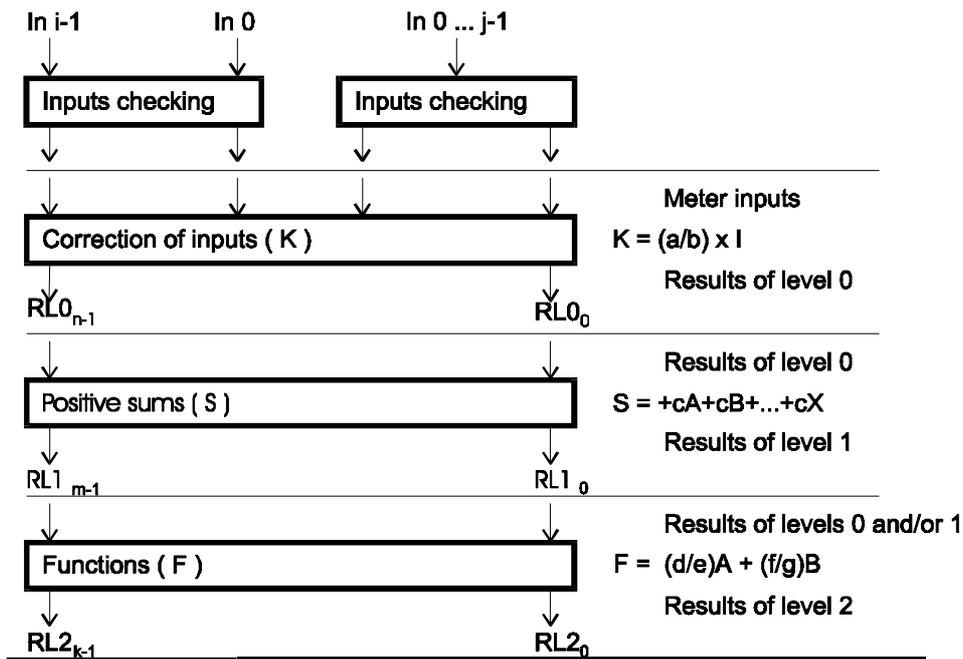
- UD\_COUNTER\_D DECADE UP/DOWN COUNTER; Positive and negative changes are permitted. The state can be changed from -99999999 to +99999999. Overflow of maximal value changes the result state to a minimal value and vice-versa!!
- UD\_PCOUNTER\_D DECADE UP/DOWN POSITIVE COUNTER; Positive and negative changes 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 communicator 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 communicator constant (CORRECTION OF INPUTS - measuring level 0). The communicator 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	<i>i</i> = number of meter impulse inputs <i>j</i> = number of meter serial inputs <i>n</i> = common number of meter inputs <i>m</i> = number of positive sums <i>k</i> = number of functions  <i>a,b,...,g</i> = integer coefficients <i>A,B,....,X</i> = results of lower levels
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	

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 communicator constant to an optional external constant.

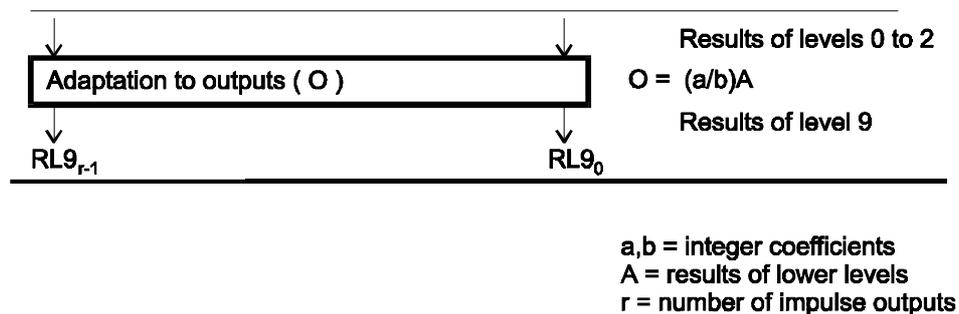


Figure 3-7 Measuring results - level 9

### 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 communicator value, i.e. the pulse transmitter (meter) constants are converted to the communicator constant. It has to be warned that the communicator 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 communicator 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 communicator ( $E_{REC}$ ) 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 communicator 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 communicator 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 communicator constant ( $k_{REC}$ ), which is 2 kWh/imp. The impulse value in the communicator is thus 2kWh, and the value of input impulse is 5/3 kWh.

$k_{REC} = 2 \text{ kWh}$  (impulse value in the communicator)  
 $k_{IN} = 5/3 \text{ kWh}$  (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 communicator  $n_{REC}$  of each input (representing the results of measuring level 0 -  $R0_i$ ) are used for further processing in the communicator:

- 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 communicator number in a group of registers IPORPARAM which represents a required input
- serial meter input      a communicator number in group of registers SERIALRESULT which represents a required input

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:

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

**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<sub>7</sub> to c<sub>0</sub> or c<sub>15</sub> to c<sub>8</sub> etc. This number is converted into decimal number N<sub>DEC</sub> 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 out of the 4 inputs is added to the sum. The record according to the formula for calculating the sums is as follows:

$$R1_i = 0R0_0 + 1R0_1 + 0R0_2 + 1R0_3 + 0R0_4 + 0R0_5 + 0R0_6 + 0R0_7$$

We have got a block with 8 results. Since there are no results from R<sub>04</sub> to R<sub>07</sub>, their parameter "c" equals 0. The block parameters are the following:

block 0-7								
parameter	c <sub>7</sub>	c <sub>6</sub>	c <sub>5</sub>	c <sub>4</sub>	c <sub>3</sub>	c <sub>2</sub>	c <sub>1</sub>	c <sub>0</sub>
value	0	0	0	0	1	0	1	0

Calculation of both decimals is:

$$N0_{DEC} = 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 170 = SIGN0_7$$

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.

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

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

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 communicator 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).

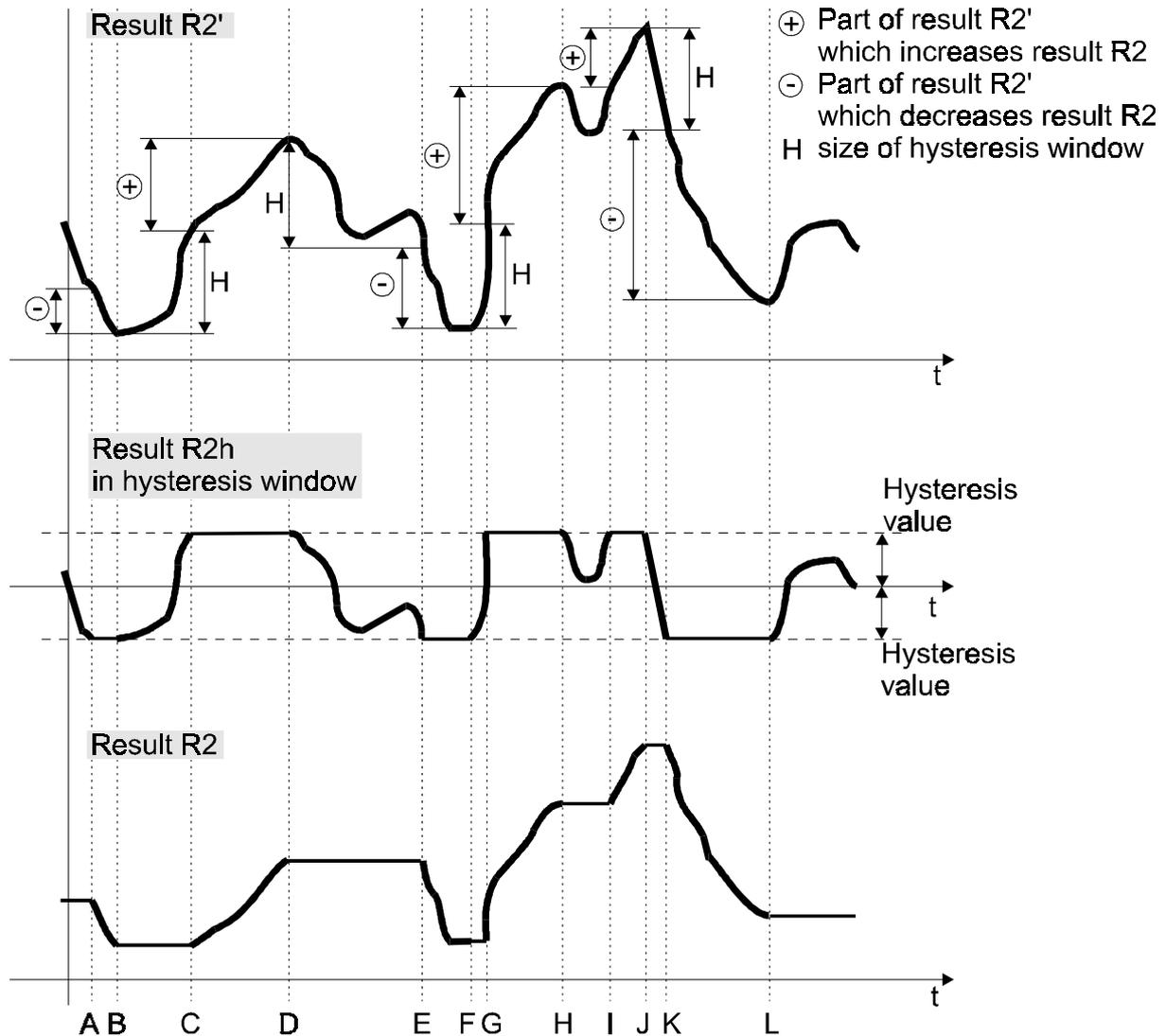


Figure 3-8 Hysteresis

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 communicator constant, into output result of different constants.

Conversion is performed similarly as on measuring level 0 by formula:

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

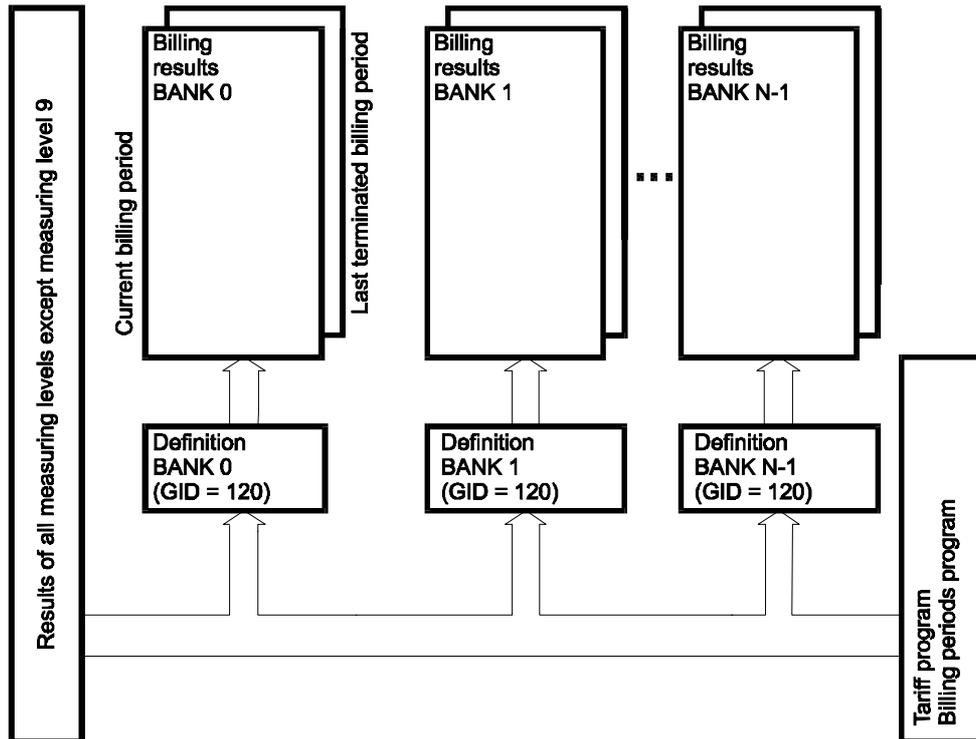
A number of results **R9** is equal to a number of impulse outputs of the communicator *i*. Factors **a** and **b** represent conversion of the **R** result from **A** level (0, 1 or 2).

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.



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	Last terminated billing period	Value 0	Time stamp 0
	Value 1	Time stamp 1		Value 1	Time stamp 1
	Value 2	Time stamp 2		Value 2	Time stamp 2
	Value 3	Time stamp 3		Value 3	Time stamp 3
	Value 4	Time stamp 4		Value 4	Time stamp 4
	Value 5	Time stamp 5		Value 5	Time stamp 5
	Value 6	Time stamp 6		Value 6	Time stamp 6
	Value 7	Time stamp 7		Value 7	Time stamp 7
	(GID = 20)	(GID = 22)		(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;** 1<sup>st</sup> 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 2<sup>nd</sup> 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
  - **2<sup>nd</sup> maximum;** 2<sup>nd</sup> biggest maximum (M2); into a 2<sup>nd</sup> maximum a value which is greater of a momentary value of 2<sup>nd</sup> maximum and at the same time smaller of the maximum value is recorded; a momentary value is recorded into 3<sup>rd</sup> greatest maximum
  - **average of two maximums;**  $PV2 = (M + M2) / 2$
  - **3<sup>rd</sup> maximum;** 3<sup>rd</sup> biggest maximum (M3); into a 3<sup>rd</sup> maximum a value which is greater of a momentary value of 3<sup>rd</sup> maximum and at the same time smaller of the 2<sup>nd</sup> 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 termination of the billing period depends on the termination of the renewal period.

- 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.

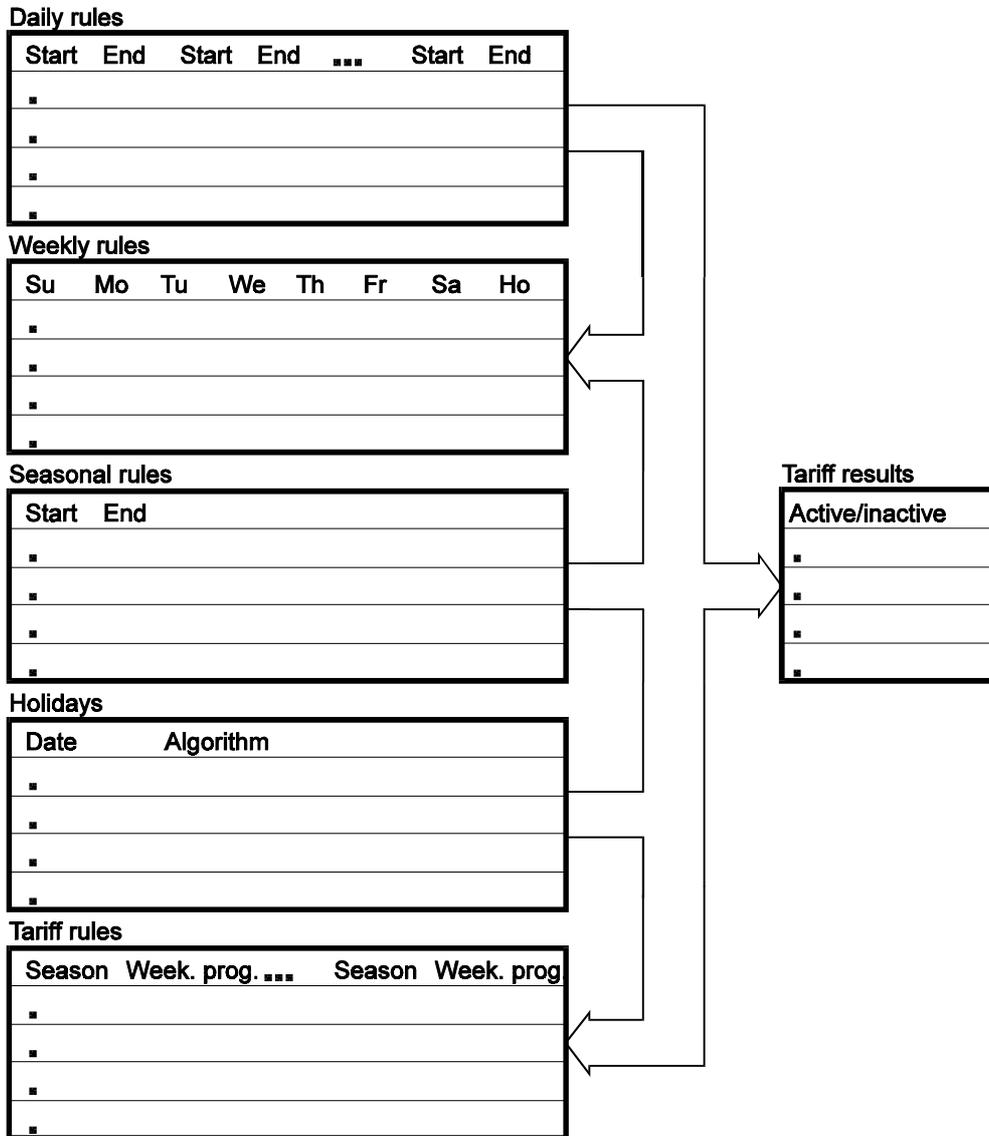


Figure 3-11 Tariff program

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:



A record of daily rules would be:

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

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

	start		end		validity
	hour	min	hour	min	
4 <sup>th</sup> rule (3)	00	00	24	00	1 <sup>st</sup> 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		validity
	hour	min	hour	min	
5 <sup>th</sup> 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 1<sup>st</sup> tariff. The second season does not have the 3<sup>rd</sup> tariff; therefore the 2<sup>nd</sup> tariff is valid in that time. The 3<sup>rd</sup> season is equal to the 2<sup>nd</sup> 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		validity
	hour	min	hour	min	hour	min	hour	min	
1 <sup>st</sup> rule (0)	00	00	06	00	22	00	24	00	1 <sup>st</sup> tariff for working day for 1 <sup>st</sup> season
2 <sup>nd</sup> rule (1)	06	00	16	00					2 <sup>nd</sup> tariff for working day for 1 <sup>st</sup> season
3 <sup>rd</sup> rule (2)	16	00	22	00					3 <sup>rd</sup> tariff for working day for 1 <sup>st</sup> season
4 <sup>th</sup> rule (3)	00	00	24	00					1 <sup>st</sup> tariff for Sunday and holiday for 1 <sup>st</sup> season
5 <sup>th</sup> rule (4)	00	00	06	00	22	00	24	00	1 <sup>st</sup> tariff for working day for 2 <sup>nd</sup> season
6 <sup>th</sup> rule (5)	06	00	22	00					2 <sup>nd</sup> tariff for working day for 2 <sup>nd</sup> season
7 <sup>th</sup> rule (6)	00	00	24	00					1 <sup>st</sup> tariff for Sunday and holiday for 2 <sup>nd</sup> season
8 <sup>th</sup> rule (7)	00	00	06	00	22	00	24	00	1 <sup>st</sup> tariff for working day for 3 <sup>rd</sup> season
9 <sup>th</sup> rule (8)	06	00	22	00					2 <sup>nd</sup> tariff for working day for 3 <sup>rd</sup> season
10 <sup>th</sup> rule (9)	00	00	24	00					1 <sup>st</sup> tariff for Sunday, holiday and Saturday for 3 <sup>rd</sup> season
11 <sup>th</sup> rule (10)	00	00	00	00					invalid rule

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

	start		end		start		end		validity
	hour	min	hour	min	hour	min	hour	min	
1 <sup>st</sup> rule (0)	00	00	06	00	22	00	24	00	1 <sup>st</sup> tariff for working day for three seasons
2 <sup>nd</sup> rule (1)	06	00	16	00					2 <sup>nd</sup> tariff for 1 <sup>st</sup> season
3 <sup>rd</sup> rule (2)	16	00	22	00					3 <sup>rd</sup> tariff for 1 <sup>st</sup> season
4 <sup>th</sup> rule (3)	00	00	24	00					1 <sup>st</sup> tariff for holiday and Sunday for three seasons
5 <sup>th</sup> rule (4)	00	00	00	00					invalid rule
6 <sup>th</sup> rule (5)	06	00	22	00					2 <sup>nd</sup> tariff for 2 <sup>nd</sup> and 3 <sup>rd</sup> 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
X <sup>st</sup> 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 1<sup>st</sup> season
- 2 rules for two tariffs in 2<sup>nd</sup> season
- 2 rules for two tariffs in 3<sup>rd</sup> season
- 1 invalid rule.

As for the 1<sup>st</sup> and 2<sup>nd</sup> seasons the rule of the 1<sup>st</sup> 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
1 <sup>st</sup> rule (0)	3	0	0	0	0	0	0	3	1 <sup>st</sup> tariff for 1 <sup>st</sup> and 2 <sup>nd</sup> seasons
2 <sup>nd</sup> rule (1)	4	1	1	1	1	1	1	4	2 <sup>nd</sup> tariff for 1 season
3 <sup>rd</sup> rule (2)	4	2	2	2	2	2	2	4	3 <sup>rd</sup> tariff for 1 <sup>st</sup> season
4 <sup>th</sup> rule (3)	4	5	5	5	5	5	5	4	2 <sup>nd</sup> tariff for 2 <sup>nd</sup> season
5 <sup>th</sup> rule (4)	3	0	0	0	0	0	3	3	1 <sup>st</sup> tariff for 3 <sup>rd</sup> season
6 <sup>th</sup> rule (5)	4	5	5	5	5	5	4	4	2 <sup>nd</sup> tariff for 3 <sup>rd</sup> season
7 <sup>th</sup> 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 2<sup>nd</sup> season and rule 5 for the 3<sup>rd</sup> 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.

1 <sup>st</sup> season		2 <sup>nd</sup> season		3 <sup>rd</sup> season		1 <sup>st</sup> season	
1.1.1997	1.4.1997			15.9.1997	1.11.1997	31.12.1997	
00:00:00	3:00:00			2:00:00	00:00:00	24:00:00	

A record of seasons would be:

1 <sup>st</sup> rule (0)	start of season									validity
	year	mon	day	Wd	alg	hour	min	sec	alg	<b>1<sup>st</sup> season</b>
	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	
2 <sup>nd</sup> rule (1)	start of season									
	year	mon	day	Wd	alg	hour	min	sec	alg	<b>2<sup>nd</sup> season</b>
	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	
3 <sup>rd</sup> rule (2)	start of season									
	year	mon	day	Wd	alg	hour	min	sec	alg	<b>3<sup>rd</sup> season</b>
	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 1<sup>st</sup> season occurs at the start and the end of a year, the year consists of 4 seasons. Therefore the 4<sup>th</sup> record is also required:

4 <sup>th</sup> rule (3)	start of season									
	year	mon	day	Wd	alg	hou	min	sec	alg	<b>1<sup>st</sup> season</b>
	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.

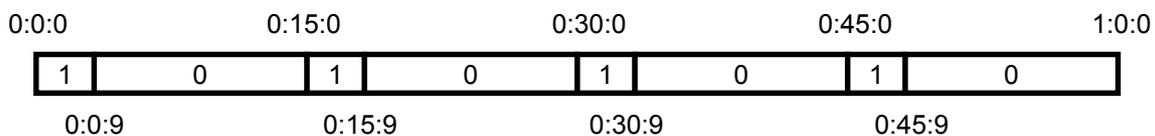
Example:

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

5 <sup>th</sup> 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 9<sup>th</sup> second. Repetition every 15 minutes gives the measuring period length, and the length of season (9 seconds) gives the impulse length.

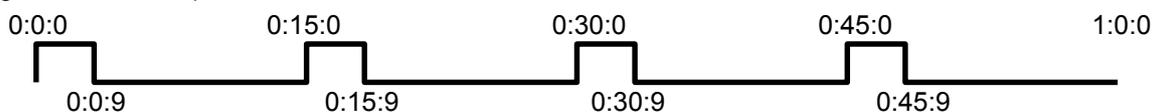
The result is active for 9 seconds every 15 minutes.



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

	type	status	gid	rid	fid	Validity
1 <sup>st</sup> rule (0)	2	0	92	4	0	<b>Dependency on seasonal result</b>

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 15<sup>th</sup> August would be:

	year	mon	day	Wd	alg	validity
X. rule (x)	X	8	15	X	1	<b>15<sup>th</sup> Aug for each year</b>

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	<b>Easter Monday every year</b>

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
1 <sup>st</sup> rule (0)	0	0	1	0	2	4	3	0	1 <sup>st</sup> tariff
2 <sup>nd</sup> rule (1)	0	1	1	3	2	5	3	1	2 <sup>nd</sup> tariff
3 <sup>rd</sup> rule (2)	0	2	1	6	2	6	3	2	3 <sup>rd</sup> 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 3<sup>rd</sup> 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 communicator are organized in groups of registers representing communicator 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 communicator. 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



### 3.8 FORMAT LISTINGS

#### 3.8.1 General

Format listings are intended for writing out of data from communicator 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 communicator 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.
    - $V = V_{MP} \times 3600 / MP$
    - $V_{MP}$  - 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 %09.1aF	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 %09.1sF	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 %09.1mF	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 %09.1dF	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 %09.1eF	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  $V = V_{MP} \times 3,600 / MP$ .

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 %09.1pF	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:

$$a1 = \text{number of repetitions} / 256 \quad (a1 = \text{whole number, rounded down})$$

$$a2 = \text{number of repetitions} - (a1 * 256)$$

**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 %09.1rF	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:

$$a1 = 1000 / 256 = 3,9 \Rightarrow a1 = 3$$

$$a2 = 1000 - (3 \times 256) \Rightarrow a2 = 232$$

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 %09.3uF	R1 00123.456

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

Communicator 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
%tF	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
%DT	03.03.1973T

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	00000000123

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 %09.1rF	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 communicator 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 3) 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:

$$a1 = 1000 / 256 = 3,906 \Rightarrow a1 = 3$$

$$a2 = 1000 - (256 * 3) = 232$$

Format string: »Kum. In. 3\n%xP %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 communicator 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 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 communicator 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 communicator 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.

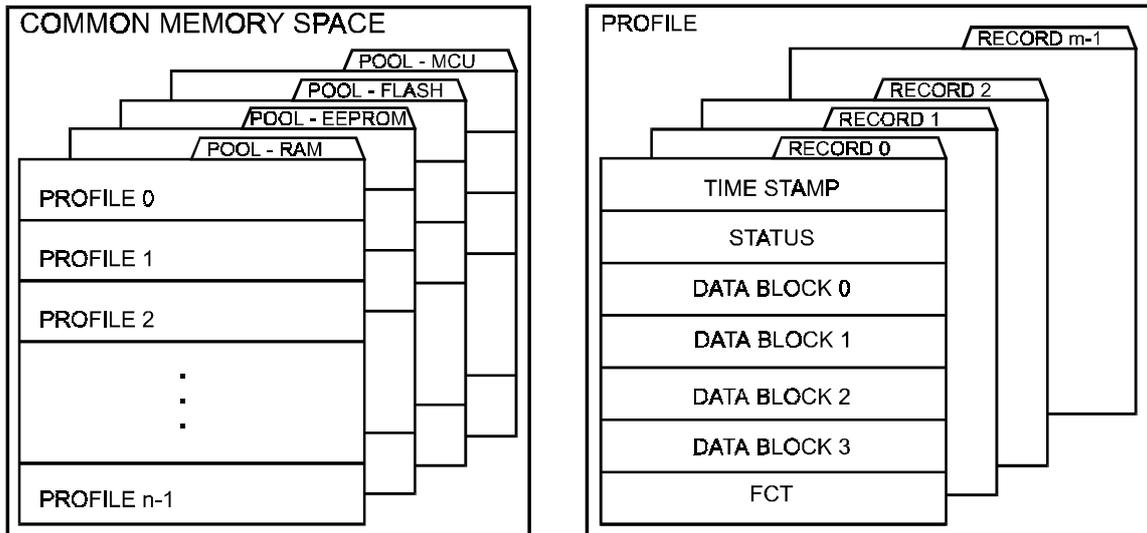


Figure 3-13 Saving structure

The common memory space comprehends all memory media in communicator, 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 communicator and the user is denied the access to it.

The stored data are written in register fields of the communicator. 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:

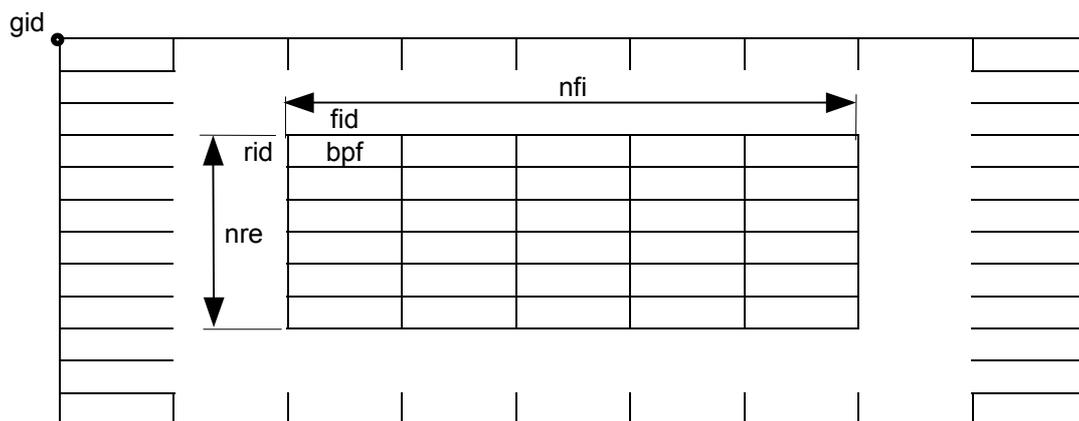


Figure 3-14 Block of registers

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 communicator 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:

Findings:

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

1. from location (L), you get the POOL size
2. from event (E) and event index (EI), the communicator 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)
4. Calculation of the record size (RS)

Data is accessible also at profile arguments.

$$(ND \times MU) + 8 = RS \text{ (byte)} \quad ; 8 \Rightarrow \text{the sum of time code(4), status(2) and control code(2)}$$

5. Calculation of profile size (PS)

Data is accessible also at profile arguments.

$$POOL \times PR = PS \text{ (byte)}$$

6. Calculation of the number of records in a profile (NR):

$$PS / RS = NR$$

7. Calculation of saving capacity for particular data (C)

$$SP \times NR = C \text{ (s)}$$

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 4 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 = 4 and that the size of data entry MU = 2 bytes (defined by profile parameters in the PROFILEPAR group of registers).
4. The record size  $\Rightarrow RS = (4 \times 2) + 8 = 16$  bytes (information can be read in the profile attributes).
5. The profile size  $\Rightarrow PS = 204800 \times 100\% = 204800$  bytes (information can be read in the profile attributes).
6. The number of records  $\Rightarrow NR = 204800 / 16 = 12800$
7. Time period  $\Rightarrow C = 12800 \times 900 = 11520000 \Rightarrow 133.3$  days

Calculation shows that the memory of 200 kbytes can store the impulses for 133 days from 4 inputs at 15-minute measuring period and the memory entry of 2 bytes per input.

## 3.10 COMMUNICATION

### 3.10.1 General

Communicator enables data transmission via interfaces for serial communication (RS232, RS485, optical interface, modems, CS lines). These communication channels with their functions permit data transmission between a communicator and a central place, data transmission between a communicator and measuring equipment, communicator 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

Due to different devices which are connected to communication channels and have different demands, different types of communication channels are required:

- **direct type** communication in compliance with IEC1107 and RS standards
- **AT modem** PSTN modems (generation of AT commands for manipulation with PSTN modem)
- **DLC modem** network modems (generation of commands for manipulation with DLC modem)
- **CATV modem** cable modem (is not realized yet)

### 3.10.3 Communication functions

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

- Terminal
- Tiskalnik
- 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 communicator and terminal equipment. Data are transmitted in command mode. The communicator receives commands and answers them. All commands are in a text form and are terminated with a Carriage Return mark (0Dh). The communicator 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 communicator.

The »device code« is defined by communicator with which the communication will be carried out, the »user code« and the »user password« determine the rights of access to communicator. The

communicator 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 communicator is permitted only for the channel through which this command has been sent and is valid until the locking of communicator by the **LO** command or until the switching off of communicator.

#### **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 communicator. 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 communicator 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 P2S 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 &lt;register address&gt;

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 &lt;register address&gt; &lt;register fields&gt;

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 all 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-conformation 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 communicator 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.

- 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.

**MR - Meter Read**

Syntax: MR <channel> <meter address> <option>

The command for meter read-out with a tariff device via a communicator. The communicator functions as the element between the service customer and the meter. Two communication channels are to be defined in the communicator. The former should be adjusted to the Terminal, the latter should be defined as an IEC1107Master for meter reading according to the IEC 1107 standard (one of the CS lines). When the request for reading is received, the communicator finds a corresponding channel (a parameter <channel>) and connects with a certain meter (a parameter <meter address>). After connection the meter identification is sent back and data reading is started. At the end of reading, data are checked and sent back to the reading service customer.

A parameter <channel> can have a value from 0 to 3 and defines which **available** IEC 1107 channel will be used. This means that the communicator searches for the **first available** IEC 1107 channel in case of the channel = 0 parameter. If less IEC 1107 channels are defined as required by the parameter, the communicator communicates about the error.

The parameter <meter address> is optional and defines the meter identification number. It is required only for those meters which support the meter addressing. This option can be used when there are more meters in a loop, and defines which meter is being read-out.

Immediately after the data transfer has been completed, the <option> parameter enables keeping the data transferred from the meter. The -K parameter is entered.

Meter readout is performed in three phases:

- READING acceptance of the request and connection with the meter
- IDENT identification of the meter and transmission mode: IDENT ISK,1200,C,LZ15M22-V1.03
- DATA data transmission from the meter into the communicator; data check and transmission from the communicator to the service customer.
- COMPLETE termination of operation

If an error occurs during one of the indicated phases of reading, the communicator returns the message:

- FAILED unsuccessful meter reading on the CS channel or the channel is not properly set
- COMPLETE BCC ERROR data transferred, but with an error
- ERROR CHANNEL IS IN D MODE the channel is adjusted to the D mode and does not allow meter reading

**MD - Meter Data**

Syntax: MD <channel> <option>

A command for reading of already read-out data which are stored in the communicator buffers. Data can be read from the meters with a tariff device by the communicator on one or several channels automatically or on request. The last read-out data are always available for each channel separately. This command enables the access to those data. The communicator channel, through which the required data were read out, is used as a parameter. Immediately after the data transfer has been completed, the <option> parameter enables keeping the data transferred from the meter. The -K parameter is entered.

If the data transfer has failed, the communicator returns the message  
DATA IS NOT AVAILABLE.

**MI - Meter Ident**

Syntax: MI <channel>

A command for reading identification string of the last read-out meter with a tariff device. The communicator channel operates as a <channel> parameter through which the data have been read out.

If the data transfer has failed, the communicator returns the message  
DATA IS NOT AVAILABLE.

**BR - Billing Reset**

Syntax: BR

The command for resetting of all meters with a tariff device, which have their resetting input connected to the corresponding input of communicators. The command changes the state of the ST field in the IEC1107MRESETATTR group of registers. The connection of the output and the ST field is defined in the OPORTPARAM group of registers.

**MC - Meter Command**

Syntax: MC <channel> <meter address> <meter password> <command> <attributes> <option>

The command for transmitting a command should be carried out (according to the IIEC1107 standard) for the meter with a tariff device. Two communication channels should be defined in the communicator. The former should be adjusted to the Terminal, the latter should be defined as the IEC1107Master for meter reading according to the IEC 1107 standard (one of the CS lines). After the meter has received a command, the Meter Command communicator looks for an adequate channel (the <channel> parameter) and contacts a certain meter (the <meter address> parameter). If the meter password is vacant (""), then an access to only unprotected data is possible. The data are made available through the IEC1107 command (the <command> and <attributes> parameters). Possible commands having their attributes according the IEC1107 standard are the following:

- P Password
- W Write
- R Read
- E Execute
- B Break

The IEC1107 command is executed in the following phases:

READING	receiving the request and connection with the meter
IDENT	meter identification and transfer mode, e.g. IDENT ISK, 1200, C, LZ15M22-V1.03
DATA	data transfer from a meter to the communicator, data checking and data transfer from the communicator to the user
COMPLETE	termination of operation

If an error occurs during one of the indicated phases of reading, the communicator returns the message:

FAILED	unsuccessful meter reading on the CS channel or the channel is not properly set;
COMPLETE BCC ERROR	data transferred, but with an error
ERROR CHANNEL IS IN D MODE	the channel is adjusted to the D mode and does not allow meter reading

After the IEC1107 command has been carried out, there are the following possible terminations of operation defined by the <option> parameter:

- -K Keep the data in the buffer (communicator) after the transfer has succeeded.
- -P Retain the meter in the "Programming mode".

When there is a major quantity of data, the latest data start to remove the older ones from the screen display. After the data transfer has been completed, the data are again available through the MD command; and the meter identification may be found out through the MI command.

**MS - Meter Synchronize**

Syntax: MS

The command for synchronization of the meter with its tariff device.

### **3.10.3.2 Printer**

The communication function is intended for communication between a communicator 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 communicator and the measuring equipment which supports the serial transmission of meter states in compliance with the DIN 19244 standard. The communicator 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 communicator 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 communicator 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 communicator 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). Communicator is a secondary station, and a central station is a primary station.

The communicator 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 communicator 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 communicator 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 communicator is not accessible or is not foreseen.

Manufacturer and product (ASDU type 1)

Messages with time stamp (ASDU type 3)

System time (type 4)

Billing meter states - 4 octets (ASDU type 8)

Billing meter states in a period - 4 octets (ASDU type 12)

Working meter states - 4 octets (ASDU type 24)

Working meter states in a period - 4 octets (ASDU type 28)

Meter states transmission involves transmission of internal communicator 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 communicator registers are mapped into individual meter values in compliance with DIN 19244.

Registers representing meter states:

Communicator register	Register address	DIN meter state	DIN address
Results of level 0	10.0 to 10.7	Billing meter state	0 to 7
Results of level 1	11.0 to 11.3	Billing meter state	8 to 11
Results of level 2	12.0 to 12.3	Billing meter state	12 to 15

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

A communicator 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 communicator 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 1 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. Communicator registers save values in pre-defined types of data. Details on types of data which are stored in communicator 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.

Request for IEC1107 Meter Reading (ASDU type 150)

A request for IEC1107 Meter Reading is an ASDU of primary station with which primary station requests reading of data from the meter according to standard IEC1107 or billing reset of meters. ASDU IEC1107 Meter Reading Request contains the following information objects:

ASDU field	Contents of ASDU field	Field type
Channel	Channel index	UI8
Flags	Operation	CP8
Meter Address	Meter Address	OS256

Field "Flags" is 8 bit compound field (CP8 {MR, MD, MI, BR, RESERVED, OK, OP, RESERVED}) and can have the following values:

- 1 MR meter reading BS1[1]
- 2 MD meter data of last reading BS1[2]
- 4 MI meter identification of last reading BS1[3]
- 8 BR billing reset of all meters (hardware reset) BS1[4]
- 16 RESERVED reserved BS1[5]
- 32 OK data keeping after transfer has been completed BS1[6]
- 64 OP retaining the meter in the "Programming mode" BS1[7]
- 128 RESERVED reserved BS1[6]

Meter reading starts, when the primary station transmits ASDU 150 by the MR operation. The secondary station returns the information about the conditions on the IEC1107Master channel (ASDU 135, INFO OBJECT = STATUS).

Now the primary station can transmit the request for reading the read out data through the ASDU 150 request and by the MD operation. The secondary station again returns ASDU 135. If the data in the secondary station were available, then ASDU 135 contains the read out data (INFO OBJECT = DATA).

After the transfer has been finished, the read out data are deleted from buffer.

**Note:**

The DIN19244 standard does not support the Transparent mode!

Request IEC1107 Meter Command (ASDU type 151)

The request IEC1107 Meter Command is the ASDU of the primary station by which the primary station sends the request (according to the IEC1107 standard) to the secondary station. The secondary station answers by ASDU 135.

The ASDU request IEC1107 Meter Command includes the following information objects:

ASDU field	Contents of ASDU field	Field Type
Channel	Number of the channel through which the communication will be carried out	UI8
Address	Meter address	OS264
Password	Password for access to the meter	OS72
Command	Command according to IEC1107	OS24
Command message	Command arguments according to IEC1107	OS632

The result of the command performed is accessible through the ASDU 150 request and by the MD operation.

The IEC1107 Meter CommandEx (ASDU type 152) Request

The IEC1107 Meter CommandEx request is ASDU of the primary station, by which it transmits a command (according to the IEC1107 standard) to the secondary station. The secondary station answers by ASDU 135.

The ASDU IEC1107 Meter CommandEx Request includes the following information objects:

ASDU field	Contents of ASDU field	Field type
Options	Request options	CP16
Channel	Number of the channel through which the communication will be carried out	UI8
Address	Meter address	OS264
Password	Password for access to the meter	OS72
Command	Command according to IEC1107	OS24
Command message	Command arguments according to IEC1107	OS632

The "Options" field is a complex one - 16 bits in length (CP16 {OD, OK, RESERVED}), and it may include the following values:

- 1 OK data keeping after transfer has been completed BS1[1]
- 2 OP retaining the meter in the "Programming mode" BS1[2]

The bits from 3 to 16 are reserved (RESERVED - UI14[3..16])!

The result of the command performed is accessible through the ASDU 150 request and by the MD operation. After the transfer has been finished, the read out data are deleted from buffer.

### 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. Communicator registers save the values in pre-defined data types. Details on data types which are saved in communicator 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 mark for values	DINTIME B
PROFPER	Period of storing 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. Communicator registers save values in predefined types of data. Details on types of data, which are saved in communicator 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 to 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. Communicator registers store the values in default data types. Details on data types stored in communicator 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.

Meter Response (type 135)

Meter response is secondary station ASDU on the requirement of a primary station. The meter response involves various information on data obtained by meter read-out in compliance with IEC1107. ASDU meter response can, according to the requirement, consist of the following information objects:

a)

ASDU field	ASDU field contents	Field type
CHANNEL	Channel number	UI8
INFO OBJECT	Object marking (=1)	UI16
IDENT	Meter identification string	CP160

Identification string consists of three fields:

- MFK designation of manufacturer OS24[1...24]
- ZBAUD transmission rate OS8[25...32]
- IDENT identification OS128[32...160]

b)

ASDU field	ASDU field contents	Field type
CHANNEL	Channel number	UI8
INFO OBJECT	Object marking (=2)	UI16
STATUS	Channel state	UI8

A channel state can have the following values:

- 0 read-out on a channel started
- 1 a channel busy
- 2 error at data transmission

c)

ASDU field	ASDU field contents	Field type
CHANNEL	Channel number	UI8
INFO OBJECT	Object marking (=3)	UI16
DATA	Data	modified

Data string consists of two fields:

- SN successive number of the first bit in data block UI16[1...16]
- DATA data OS8n[1...8n]

**3.10.3.5 IEEE802.3 Slave**

**is not realized yet**

The communication function is intended for communication between the communicator 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**

This function is designed for a direct communication between the communicator and the meter according to the IEC 1107 standard. It enables meters reading and transfer of meter messages to the central station. By using different commands from the central station it is possible to read out different data from the meter. Meter reading may be done by pressing the key for reading on the meter or it may be automated by the P2S communicator.

The communicator supports the IEC 1107 A, B, C and D modes of data transfer. It is possible to connect up to 4 meters without their identification numbers (they may be equal) or up to 16 meters with their own identification numbers (up to 4 meters on one CS line) to four CS lines.

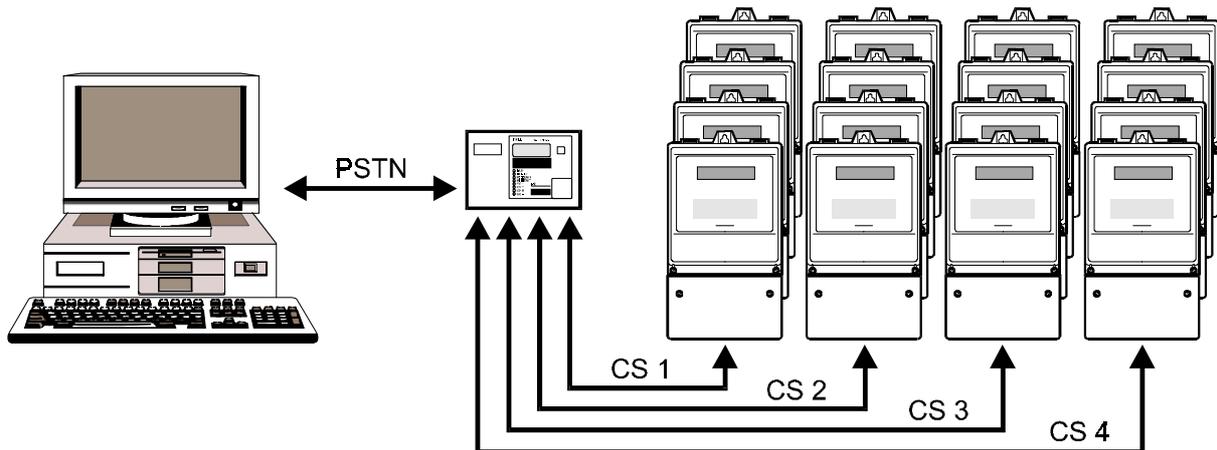


Figure 3-15 IEC 1107 Master

The channel function should be adjusted to the **IEC 1107 Master**. The channel function, through which the communicator is connected with the computer, may be adjusted either to the **Terminal** or the **DIN 19244 Slave**. The program (POREG2View, HyperTerminal, ...), operating in the computer, is familiar with the corresponding commands for data transfer through the communicator - from the meter to the computer. The commands for the Terminal and DIN19244 have the same structure and are of equal importance to the full support for the IEC1107 standard.

**Meter reading operations according to IEC 1107**

Meter reading operations according to the IEC 1107 standard are installed in the communicator as basic functions of that standard. These functions use the IEC 1107 messages that are transmitted between the communicator and the meter. The IEC 1107 messages are:

REQUESTMSG	Request
IDENTMSG	Identification
ACKOPTMSG	Confirming / selecting the option
DATAMSG	Data
ACKMSG	Confirmation
NACK	Repeating the request
PROGCMDMSG	Message argument
PROGCMDMSGOPT	Message argument that may be divided into individual blocks
PROGDATAMSG	Data in the programming mode of work
PROGDATAMSGOPT	Data in the programming mode of work that may be divided into blocks
ERRORMSG	Error

Meter Readout Operation

Meter Readout operation supports the IEC 1107 mode A, B and C meter reading.

Operation arguments are:

- MeterAddress            If the meter supports the setting with identification numbers, then it is used for access to more than one meter on the same CS channel.
- Options                 Operation option controlling the performing of operation:
  - K            Keep the message after the transfer has been completed

Figure 3-16 shows the Meter Readout Operation.

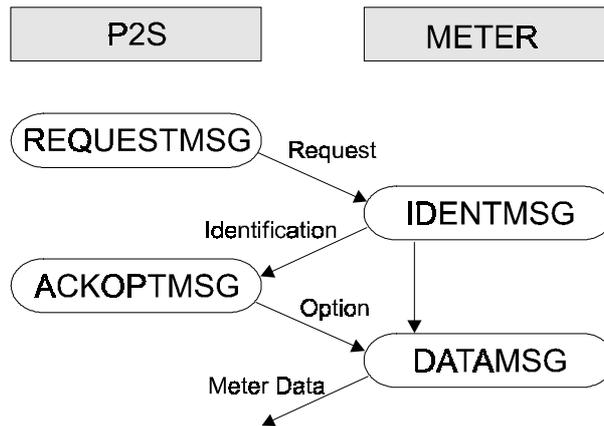


Figure 3-16 IEC 1107 Meter Readout

When using the IEC 1107 mode A and B meter reading, the communicator transmits the request (REQUESTMSG), receives the corresponding identification (IDENTMSG) and data (DATAMSG). When using the IEC 1107 mode C meter reading, the communicator transmits some options (ACKOPTMSG) and receives data (DATAMSG).

Meter Command Operation

Meter Command Operation supports the IEC 1107 mode C of performing commands when the meter is set in the Programming mode. Various data in the meter are available (registers, profiles, records of events etc.) by performing this operation. As soon as the communicator receives the meter data, they are transferred to the central station immediately.

After the operation has been completed, the following options are possible:

- The communicator may retain the data block in its memory (the K option).
- At the end of the operation, the communicator automatically transmits the BREAK command if there is no request for the meter to be retained in the Programming mode (the P option).

The operation arguments are:

- MeterAddress            If the meter supports the setting with identification numbers, then it is used for access to more than one meter on the same CS channel.
- Password                Geslo za dostop do zaščitениh podatkov v števcu.
- Command                A command for the meters in compliance with IEC 1107 standard.
  - P       Password
  - W       Write
  - R       Read
  - E       Execute
  - B       Break
- Command message    The argument in compliance with IEC 1107 standard.
- Options                 Operation option controlling the performing of operation:
  - K       Keep the message after the transfer has been completed.
  - P       retaining the meter in the "Programming mode".

Operation:

The communicator transmits the request (REQUESTMSG) and receives the identification (IDENTMSG). Then the option is sent (ACKOPTMSG) and changed over from initial communication rate to the agreed one. The meter transmits the PO password (PROGCMDMSG). If P1 password (the operation argument) entails characters, it is sent by the communicator to the meter (PROGCMDMSG). If the password is correct, the meter confirms this (ACKMSG). If P1 password is empty, this phase is skipped and the command which follows has the access only to unprotected data in the meter. Then the communicator transmits the command and its arguments. At the end, the break is transmitted (BREAK) if the option for keeping the connection is not selected.

Figure 3-17 shows how the command in the Meter Command operation is performed:

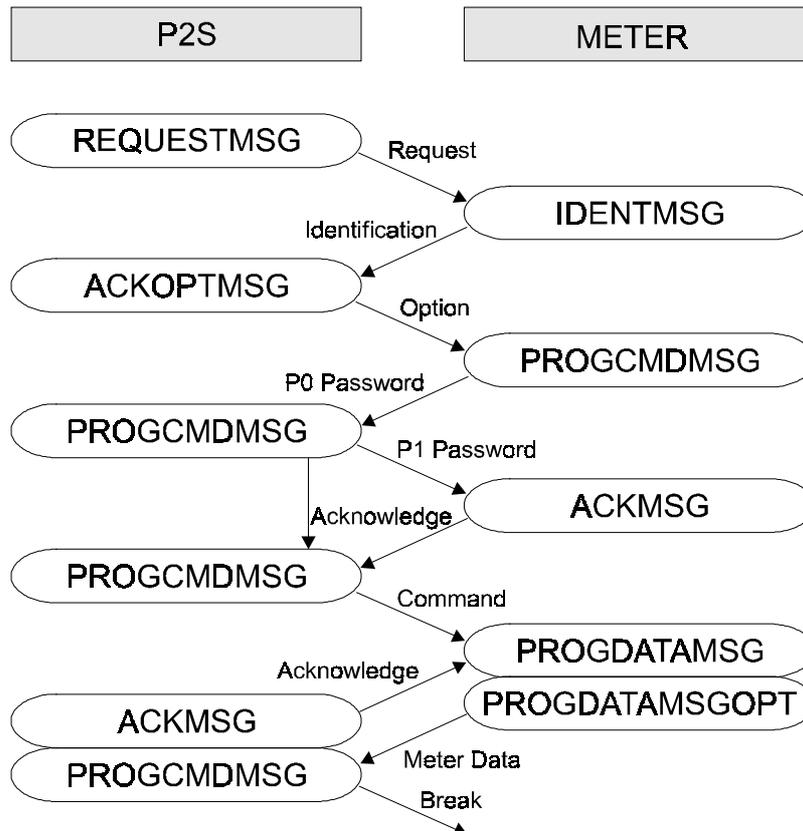


Figure 3-17 IEC 1107 Command execution

### 3.10.3.7 Cascade

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

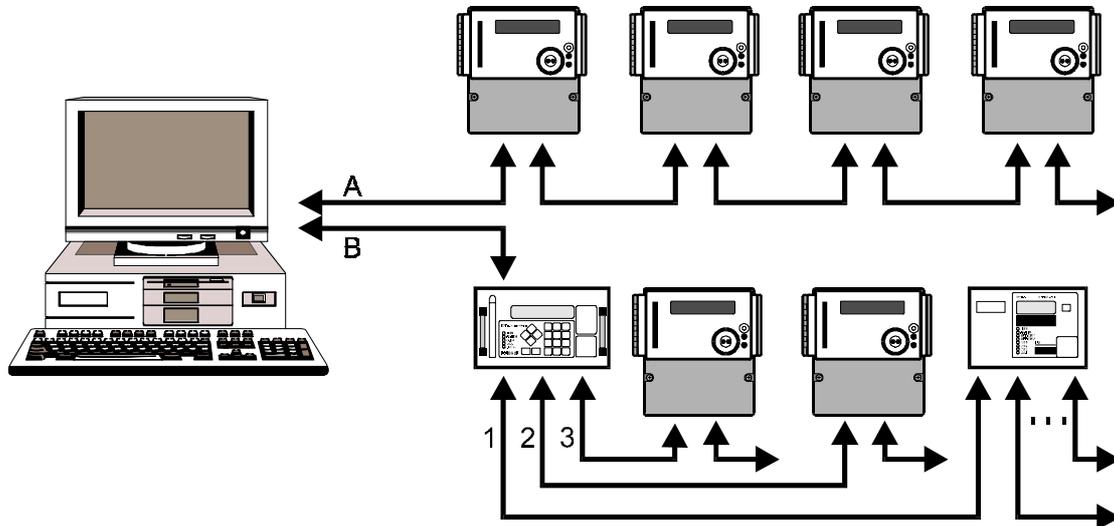


Figure 3-18 Cascade connection - network

Figure 3-18 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.

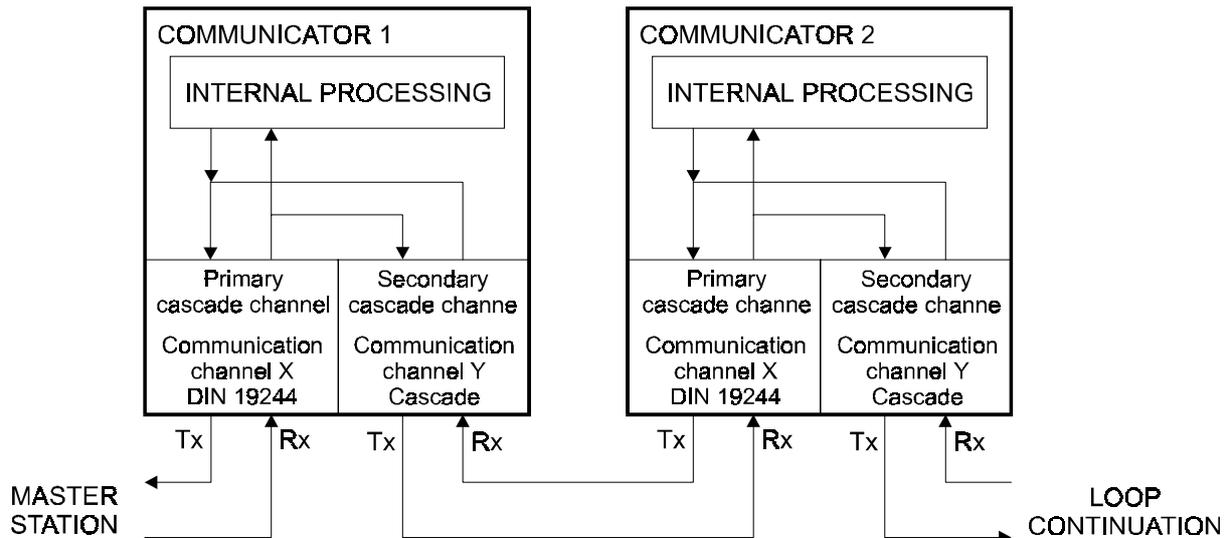


Figure 3-19 Cascade connection - connecting method

Figure 3-19 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 communicator 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 communicator, a communicator, etc.).

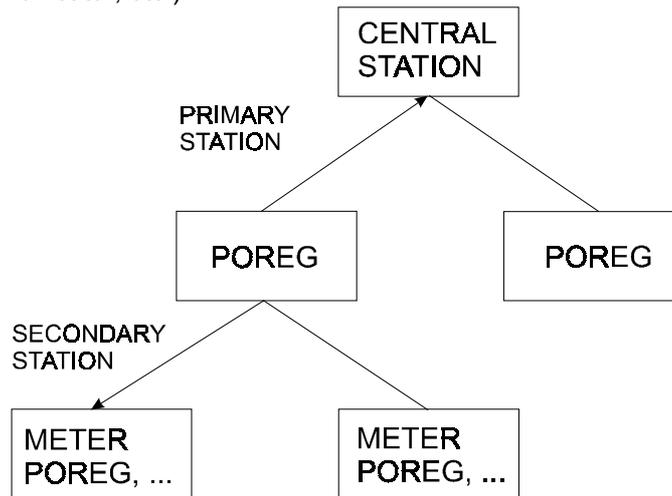


Figure 3-20 Modem communication

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 communicator 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 3-21):

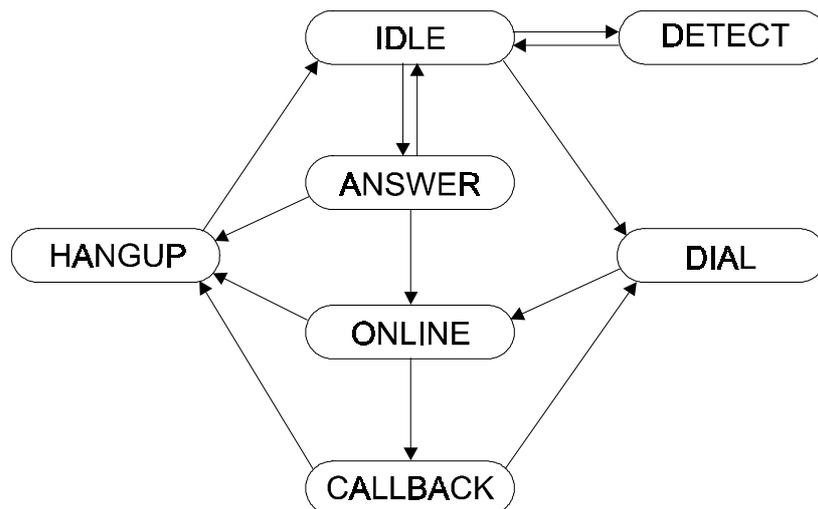


Figure 3-21 Status of modem channel

The channel is usually in IDLE state. The communicator 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 communicator 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 call; CALLBACK function is switched on in the modem

In IDLE state it is waiting for a certain number of ringings. Then the communicator 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 communicator 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 communicator), the communicator calls the caller (DIAL). After successful establishment of a link (ONLINE) and communication between the communicator and the caller, the line is interrupted (HANGUP) and IDLE state is restored.

- automatic call of the communicator when time alarm is activated

When time alarm is activated, the communicator calls the station which is defined with parameters for that call. After successful establishment of a link (ONLINE) and communication between the communicator 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.

### 3.10.5 Communication through the DLC modem

The DLC modem may be connected to the communication channel MODEM 1 being adequately defined (the DLC modem). The DLC modem is designed for the communication in the low-voltage network. Therefore it needs no special wiring. The DLC modem is a part of the communication network (hereinafter referred to as network). The network is confined to one transformer station.

Figure 3-22 shows a general scheme of the communication network for the DLC modems.

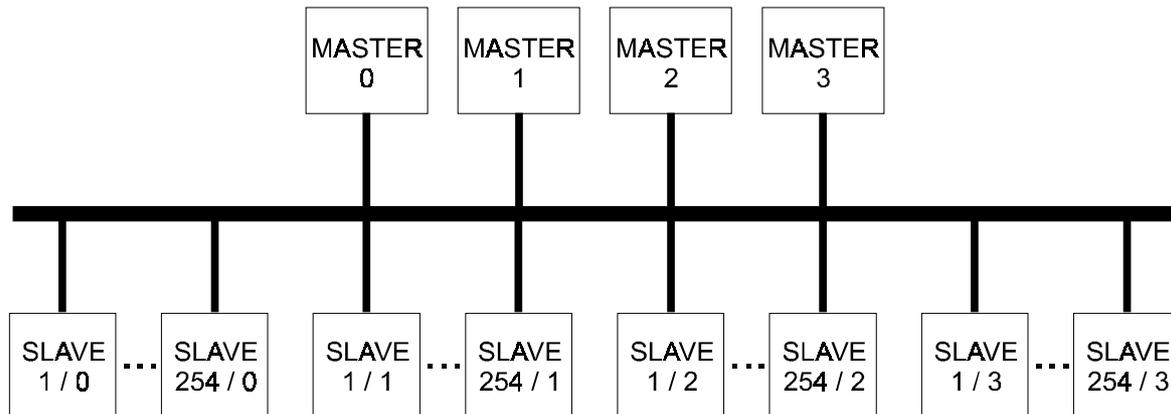


Figure 3-22 The DLC network

MASTER and SLAVE represent the MASTER and SLAVE station. The station consists of both a device and the DLC modem. The device is called computer with the MASTER station. The device is called communicator with the SLAVE station. Four MASTER stations may be connected to the network. Each of them may have up to 254 SLAVE stations. The arrangement of individual stations in the network is not essential.

Each station has its own identification number. There are numbers from 0 to 3 for the MASTER stations, from 1 to 254 for the SLAVE stations. However, it is necessary to define the SLAVE stations' belonging to the MASTER station. Individual SLAVE station may belong to only one MASTER station. The combination of their identity number and their belonging to a certain station undoubtedly specifies the SLAVE station in the network. The MASTER station constantly checks the list of the SLAVE stations belonging to it.

**Note:**

There is the 1.X in 2.X software version for the DLC modem. The above-mentioned data about the number of the MASTER and SLAVE stations refer to the 1.X versions. The 2.X versions support 8 MASTER stations and 4096 SLAVE stations for each MASTER station. In accordance with this, the quantity of identification numbers is increased.

### 3.11 CONSOLE

#### 3.11.1 General

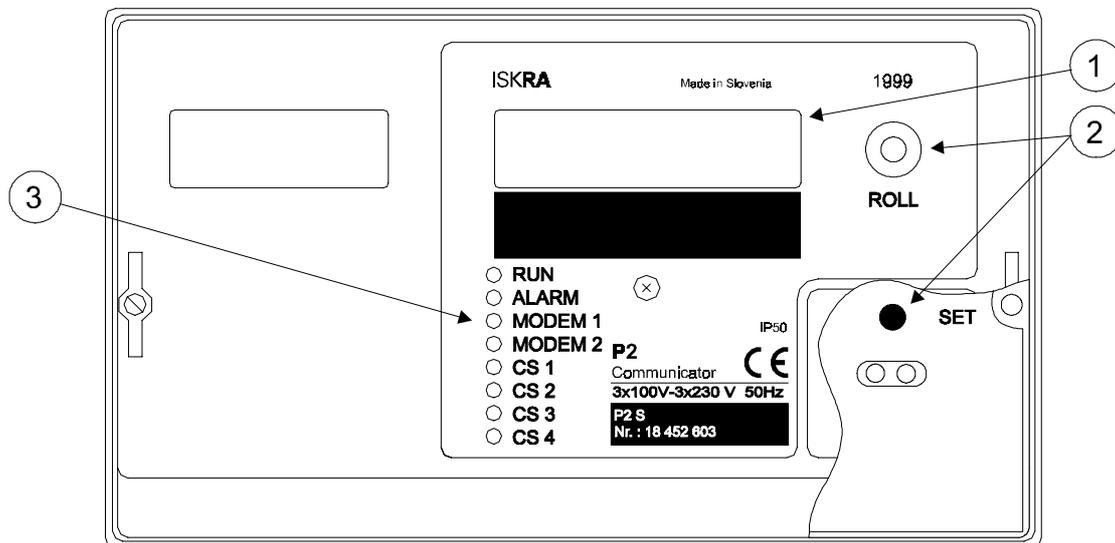


Figure 3-23 Console

The communicator is provided with a local LCD (1), a keyboard (2) and 8 LED's (3) 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

The communicator is provided with a simple keyboard with only two function keys:

- ROLL - key for shifting forward (roll) for shifting within the menus and information
- SET - key for confirmation of selection

SET key is accessible only when the communicator cover is opened!

If any key is pressed for more than 2 s, its function is repeated every 100 ms. The keyboard supports also double pressing the key, i.e. the same key is immediately pressed twice.

Double pressing of ROLL key is used for displaying of format listings (Display/Autoscroll), at a survey of register contents (Display/Register), and at a survey of registers which are stored in profile-in (Display/Profile Register).

Double pressing the SET key activate the main menu.

### 3.11.3 Display

The communicator 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 communicator information.

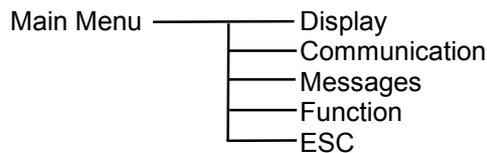


### 3.11.4 Local manipulation

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 SET key. The main menu is displayed:



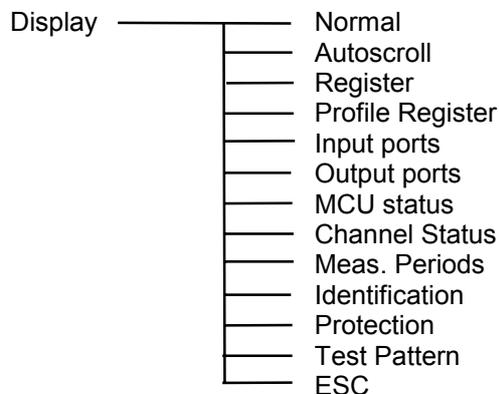
By pressing the ROLL key it is possible to move within the main menu and select one of the options of the main menu.



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.

#### Display menu

**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.



**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:

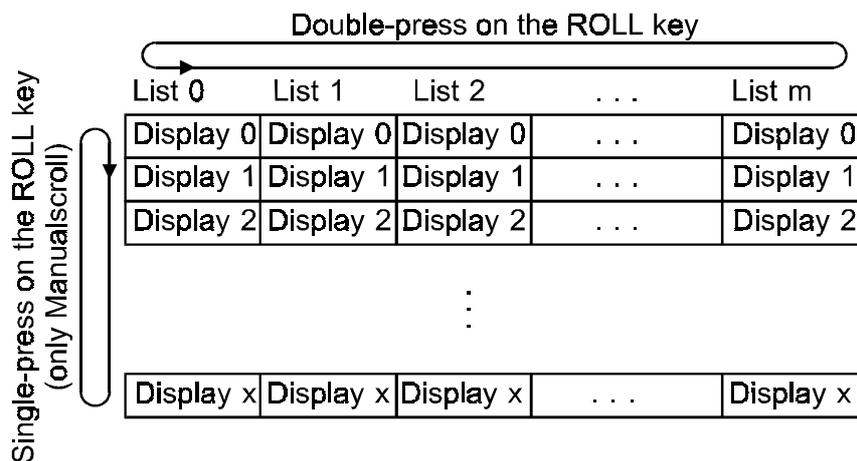
Act.1 Energy  
12345.678 kWh

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 double-pressing the SET 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 communicator switches to Autoscroll mode after certain time (see Format Commands). The communicator 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 3-24):



"x", "y", "z" and "n" can vary !

"m" for Autoscroll and Manualscroll can be different !

Figure 3-24: ROLL key - »Autoscroll« and »Manualscroll« display

**Register** operation mode of display records value of selected register on a display. The mode is selected by choosing the Register option from the Display menu. Then the register address should be entered and selection confirmed. Individual numbers of the register address can be entered by means of the ROLL key, so that numbers are increased automatically. The next number is obtained with the SET key.

Entry of 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 increases the number of register. Each double-pressing on the ROLL key increases the number of field. 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.

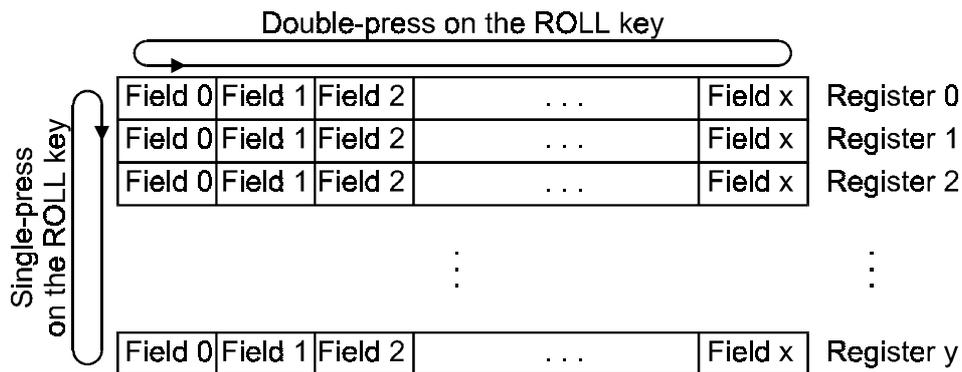


Figure 3-25: ROLL key - »Register« display

Example:

Single presses on the ROLL key:

10.0.0  
10.1.0  
10.2.0  
...  
10.x.0

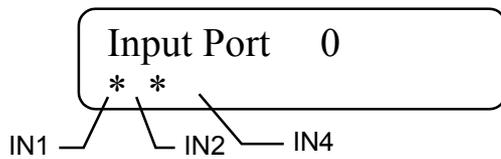
Double presses on the ROLL key:

10.0.0  
10.0.1  
10.0.2  
...  
10.0.x

You can return to the main menu by double-pressing the SET key.



**Input ports** operation mode displays the status of inputs. 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:



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

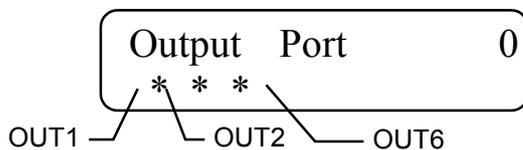
Table 3-4 Display: Input ports

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.

You can return to the main menu by double-pressing the SET key.

**Output ports** operation mode displays the status of outputs. 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:



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.

You can return to the main menu by double-pressing the SET key.

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



Communication lines are marked **Opt.** (optical communication), **Ch1** (communication channel 1), ... and **Ch8** (communication channel 8). Selection can be made with ROLL key. The following messages are available:

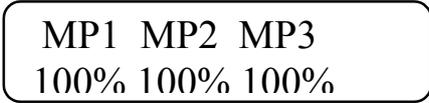
Line state	Phase	Explanation
<b>Idle</b>	<b>Init</b>	Line is in rest position
<b>Modem Detect</b>	<b>Init</b> <b>Wait</b> <b>User Init</b>	Check of modem presence <ul style="list-style-type: none"> <li>• modem initialisation</li> <li>• waiting for modem response</li> <li>• initialisation of modem with a command, defined in a group of registers ATPARAM</li> </ul>
<b>Dial</b>	<b>Init</b> <b>Command</b> <b>Wait</b> <b>Connect</b> <b>Connect Delay</b>	Link establishment via a modem <ul style="list-style-type: none"> <li>• setting to a line</li> <li>• call</li> <li>• waiting for modem response</li> <li>• establishment of modem link</li> <li>• stabilisation of conditions on a line</li> </ul>
<b>Answer</b>	<b>Init</b> <b>Wait</b>	Response to external call <ul style="list-style-type: none"> <li>• setting to a line</li> <li>• waiting for establishment of modem link</li> </ul>
<b>Callback</b>	<b>Init</b> <b>Delay</b> <b>Escape</b>  <b>Wait</b> <b>On Hook</b>	Establishment of link via a modem after the external call if CALLBACK function is active <ul style="list-style-type: none"> <li>• setting to a line</li> <li>• stabilisation of conditions on a line</li> <li>• ESCAPE sequence for link interruption is sent by a modem</li> <li>• waiting for a response</li> <li>• interruption of a line</li> </ul>
<b>Hangup</b>	<b>Init</b> <b>Delay</b> <b>Escape</b> <b>Wait</b> <b>On Hook</b>	Interruption of a line <ul style="list-style-type: none"> <li>• initialisation of line interruption</li> <li>• obvious delay before ESCAPE sequence</li> <li>• ESCAPE sequence sent by a modem</li> <li>• waiting for a response</li> <li>• line interruption</li> </ul>
<b>Online</b>	<b>Init</b> <b>Dial Init</b> <b>Answer Init</b> <b>Wait Connect</b> <b>Connected</b>	A link is established (for modem and RS232 or RS485) <ul style="list-style-type: none"> <li>• establishment of link in a direct mode</li> <li>• establishment of connection after modem call</li> <li>• establishment of link after modem response</li> <li>• waiting for CO command</li> <li>• CO command carried out</li> </ul>

Table 3-6 Channel status - Messages

You can return to the main menu by double-pressing the SET key.

**Meas. Periods**

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



You can return to the main menu by double-pressing the SET key.

**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.



Checksum is arithmetic sum of all memory locations except the last two where pre-defined checksum is recorded.

You can return to the main menu by double-pressing the SET key.

**Protection**

State of protection with a switch is displayed.



You can return to the main menu by double-pressing the SET key.

**Test Pattern**

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

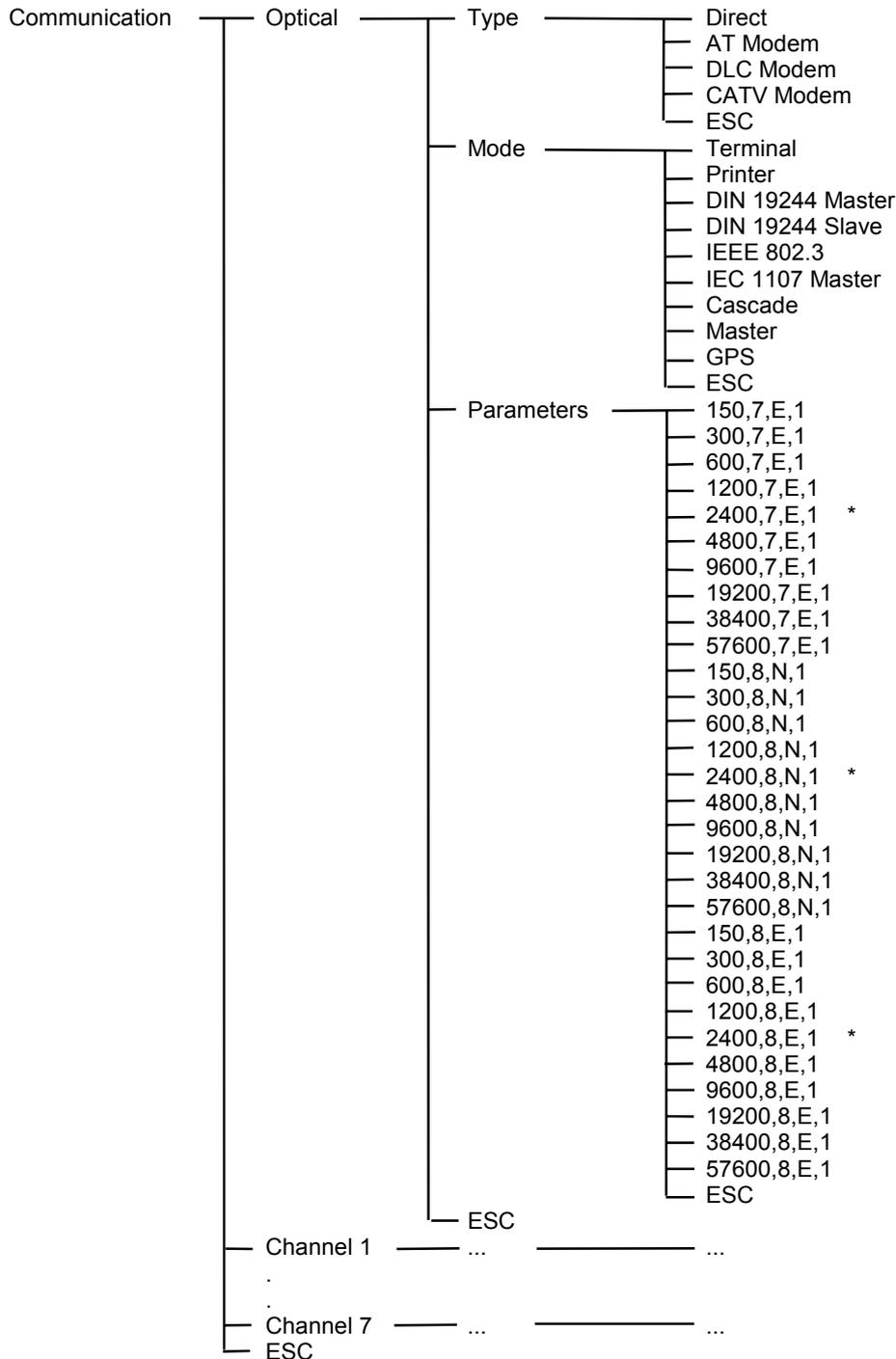


You can return to the main menu by double-pressing the SET key.

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).

**Optical** channel is reserved only for optical connection, other channels are intended for a modem, RS232, RS485 or CS channels. **AT Modem** communication type is selected for a PSTN modem, **DLC Modem** for a DLC modem, and a **direct** type is selected for the others. Other parameters are set regarding the channel purpose. CATV modem has not been realised yet! The functions Master in GPS have likewise not been realised yet!



\* - recommended setting for optical communication

Messages menu

Messages menu is designed for processing the system messages recorded on the lists for confirming and removing something. The "Acknowledge" option on the confirmation list means "confirm"; the "Clear" option on the deletion list means "remove".



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 see a survey of messages. At the end of the survey all surveyed messages are checked (Acknowledge list) or deleted (Clear list) with the SET key. Simultaneous surveying and confirming (deleting) the messages with the SET key are also possible.

An interruption of either confirming or deleting the messages is possible by double pressing the SET 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" menu item is intended for switching on and off the messages displaying

Messages  
Disnlav

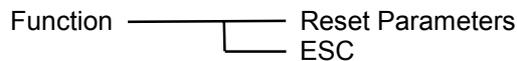
At the first device switch-on or reset, the message display is switched off.

Display  
Off

By pressing the ROLL key, the message display is switched on or off. If the device is protected against parameter setting with the switch, the setting mode can not be changed!

Function menu

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

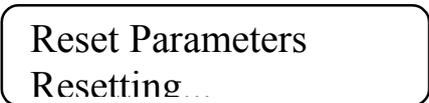


**Reset Parameters** function

This function resets the functioning parameters of communicator to the initial values.



The function starts by pressing on the SET key.



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



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 over from summer to winter time is carried out every year on first Sunday after 24<sup>th</sup> October (the DAYLIGHT group of registers)
- external time synchronization is switched off (TIMESYNCHRO group of registers)
- function of impulse retaining is switched off (IMPULSBARRIERE group of registers)
- all CS lines are set to UART (CONFIGPORT group of registers )
- all impulse-inputs are of the S1a type; the shortest duration of an import impulse is 40 ms, and the longest impulse duration is not checked (IORTPARAM 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; connections of results with group of registers LEV9PARAM are arranged in order (OORTPARAM group of registers)
- a counter type for inputs from 0 to 3 is set to 1 (U\_COUNTER\_B), and for inputs from 4 to 7 to 0 (NULL); correction factors for all results of the measuring level 0 are set on 1; the input type for inputs from 0 to 3 is set to IMPULSE, others are set to NULL; 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 (LEV2PARAM 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, CALLSTATION, TELNUMPARAM)
- all parameters for communication via a DLC modem are set to 0 (zero) (a group of registers PLCPARAM)
- 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)

### 3.11.5 LED's

The communicator is provided with eight LED's which are intended for fast survey of a communicator operation. The LED's from up downwards indicate:

- RUN           communicator operation
- ALARM       indication of the error (the ERROR system message)
- MODEM 1   communication via MODEM 1 module (DLC modem)
- MODEM 2   communication via MODEM 2 module (PSTN modem, ISDN TA, RS232C)
- CS 1       activity on CS 1 line (communication or impulses)
- CS 2       activity on CS 2 line (communication or impulses)
- CS 3       activity on CS 3 line (communication or impulses)
- CS 4       activity on CS 4 line (communication or impulses)

Activity on CS lines depends on the setting of the lines:

- CS line as input                           pulse transmitter has closed the line
- CS line as output                         communicator generates an impulse
- CS line as a communication line       communication in both directions

If CS line is defined as input-output line, the operation of a corresponding LED can be checked on a display (Menu Display/Input Ports or Menu Display/Output Ports).

### 3.12 SYSTEM MESSAGES

The communicator 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 and recorded in registers. A system message is also displayed by an alarm indicator on the front panel.

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. These messages cannot activate the alarm output.

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 serious 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 or 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;
- a suitable alarm output is activated if it is defined.

The communicator 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 commands 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. GRUOPS 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 communicator. 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 characteristic data with its structure, permission for the access to data, description of individual fields, etc. Characteristic 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 characteristic data of a group the data on a number of fields in a register (FPR) is also added.

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 ... FD5 = <basic field type > + <number of bytes in the field>, <number of fields>

FPR =

Tabular display:

	FPR-1	
RPG-1		

Table 4.X Group of registers - NAME

Contents of fields:

Field	Contents	Type	Length
-------	----------	------	--------

A group of registers in a communicator:

<b>Group</b>	<b>GID</b>	<b>Contents of data</b>	<b>Page</b>
SYSTEMATTR	0	System attributes	100
DEVICEID	1	Communicator's identification code	103
USERS	2	List of users	104
DATETIME	3	System date and time	105
TIMEATTR	4	Time functions attributes	106
PERFORM	5	Performance measurements	108
MEASPER	6	Measuring periods	109
DAYLIGHT	7	Parameters for shifting from summer to winter time	110
EVENTS	8	Communicator internal events	111
TIMESYNCHRO	9	Parameters of external synchronization	114
LEV0RESULT	10	Measuring results of level 0	115
LEV1RESULT	11	Measuring results of level 1	116
LEV2RESULT	12	Measuring results of level 2	117
LEV9RESULT	19	Measuring results of level 9	118
BANKRESULT	20	Billing results - current billing period	119
BANKRESULTL	21	Billing results - terminated billing period	120
BANKSTAMP	22	Time of billing results - current billing period	121
BANKSTAMPL	23	Time of billing results - terminated billing period	122
ATATTR	81	Attributes of modem communication	123
SERIALRESULT	82	Results of serial meter inputs	124
IEC1107MREADATTR	83	Attributes of meter read-out	126
IEC1107MRESETATTR	84	Attributes of meter reset	127
CALLATTR	87	Attributes of calls	128
PLCATTR	88	Attributes of DLC modem	129
TARSTAT	89	Tariff status	131
TARDAYRES	90	Results of daily tariff rules	132
TARWEEKRES	91	Results of weekly tariff rules	133
TARSEASRES	92	Results of seasonal tariff rules	134
TARHOLRES	93	Results of rules for holidays	135
TARRES	94	Results of tariff programs	136
ALARMATTR	95	Attributes of time alarms	137
IPOINTATTR	101	State of input impulse lines	138
OPOINTATTR	102	State of output impulse lines	139
IMPULSEBARRIERE	106	Parameters of function of impulse retaining	140
CONFIGPORT	107	Parameters of CS lines	141
IPOINTPARAM	108	Parameters of inputs	142
OPOINTPARAM	109	Parameters of outputs	143
LEV0PARAM	110	Parameters of measuring results of level 0	144
LEV1PARAM	111	Parameters of measuring results of level 1	146
LEV2PARAM	112	Parameters of measuring results of level 2	148
LEV9PARAM	119	Parameters of measuring results of level 9	150
BANKPARAM	120	Parameters of billing results	151
CHANNELPARAM	180	Parameters of serial channels	153
ATPARAM	181	PSTN modem parameters	155
SERIALPARAM	182	Parameters of serial meter inputs	156
IEC1107MREADPARAM	183	Parameters of meter read-out	158
IEC1107MRESETPARAM	184	Parameters for meter reset	160
CALLSTATION	185	Parameters of stations	161
TELNUMPARAM	186	Station telephone numbers	162
CALLPARAM	187	Parameters for call via a modem	163
PLCPARAM	188	DLC modem parameters	165
CASCADE	189	Parameters of Cascade communication function	166
TARDAYPROG	190	Parameters of daily tariff rules	167

<b>Group</b>	<b>GID</b>	<b>Contents of data</b>	<b>Page</b>
TARWEEKPROG	191	Parameters of weekly tariff rules	168
TARSESPROG	192	Parameters of seasonal tariff rules	169
TARHOLPROG	193	Parameters of rules for holidays	171
TARPROG	194	Parameters of tariff programs	172
ALARMPARAM	195	Parameters of time alarms	173
UNITS	200	Measuring units	175
FORMSTR	201	Format strings	176
FORMOUT	202	Format listings	177
FORMLST	203	Format lists	178
FORMCMD	204	Format commands	179
PROFILEPAR	210	Parameters of saving into profiles	180
PROFILEATTR	211	Attributes of saving into profiles	181
PROFILESTAT	212	Status information of saving into profiles	182
POLLATTR	213	Information on POOL	183

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: <table border="1"> <thead> <tr> <th>Bit</th> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>• 0</td> <td>1</td> <td>failure of phase R</td> </tr> <tr> <td>• 1</td> <td>2</td> <td>failure of phase S</td> </tr> <tr> <td>• 2</td> <td>4</td> <td>failure of phase T</td> </tr> <tr> <td>• 3</td> <td>8</td> <td>device unlocked (parametration allowed)</td> </tr> <tr> <td>• 4</td> <td>16</td> <td>device was opened</td> </tr> <tr> <td>• 5</td> <td>32</td> <td>power supply failure at processor</td> </tr> </tbody> </table> Value in this field changes dynamically. Several simultaneous different events means summing up of individual bit 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 was opened	• 5	32	power supply failure at processor	UBINARY	4
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 was opened																						
• 5	32	power supply failure at processor																						
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: <ul style="list-style-type: none"> <li>• 1 result of the last calculation of security code (Checksum) does not equal the entered value</li> </ul> Value in this field preserves itself.	UBINARY	4																					
Alarm	The system alarm (a bit map of the PC field in the group of registers EVENTS)	UBINARY	4																					
Warning	The system warning (a bit map of the PC field in the 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																					

Field	Contents	Type	Length
SOnDur	Duration (s) of power supply in phase S	UBINARY	4
TFailCount	Serial number of power supply failure in phase T	UBINARY	4
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, RL, WL, PAR, EE  
 RPG = 1  
 BPR = 9  
 FD0 = ASCII + 9, 1  
 FPR = 1

Tabular display:

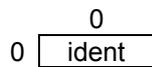


Table 4-3 Group of registers **Napaka! Slog ni definiran.**

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 communicator. 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:

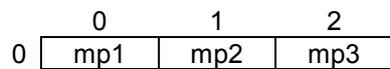


Table 4-8 Group of registers **Napaka! Slog ni definiran.**

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, WL, 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 yearlyeast 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 communicator. During the operation the communicator 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	<ul style="list-style-type: none"> <li>0 - program restart</li> <li>1 - CPU warm start</li> <li>2 - CPU cold start</li> <li>17 - memory card module warm start</li> <li>18 - memory card module cold start</li> <li>33 - printer module warm start</li> <li>34 - printer module cold start</li> <li>49 - communication module warm start</li> <li>50 - communication module cold start</li> </ul>
System self test error	<ul style="list-style-type: none"> <li>1 - CPU program memory</li> <li>2 - CPU parameter memory</li> <li>3 - CPU data memory</li> <li>17 - memory card module program memory</li> <li>18 - memory card module parameter memory</li> <li>19 - memory card module data memory</li> <li>33 - printer module program memory</li> <li>34 - printer module parameter memory</li> <li>35 - printer module data memory</li> <li>49 - communication module program memory</li> <li>50 - communication module parameter memory</li> </ul>
Power supply error	<ul style="list-style-type: none"> <li>1 - CPU power supply shutdown</li> <li>2 - CPU power supply disturbance</li> <li>3 - CPU power supply 1 defect</li> <li>4 - CPU power supply 2 defect</li> <li>8 - Power supply phase R OFF - IE Extension</li> <li>9 - Power supply phase S OFF - IE Extension</li> <li>10 - Power supply phase T OFF - IE Extension</li> <li>11 - Power supply phase R ON - IE Extension</li> <li>12 - Power supply phase S ON - IE Extension</li> <li>13 - Power supply phase T ON - IE Extension</li> </ul>
Battery error	<ul style="list-style-type: none"> <li>1 - CPU module battery error</li> <li>17 - memory card module battery error</li> <li>33 - printer CPU module battery error</li> <li>49 - communication module battery error</li> <li>65 - memory card module battery error</li> </ul>
Data overrun	<ul style="list-style-type: none"> <li>1 - CPU data overrun MP1</li> <li>2 - CPU data overrun MP2</li> <li>3 - CPU data overrun MP3</li> <li>10 - CPU data overrun messages</li> <li>17 - memory card module data overrun MP1</li> <li>18 - memory card module data overrun MP2</li> <li>19 - memory card module data overrun MP3</li> <li>26 - memory card module data overrun messages</li> <li>32 - printer module data overrun</li> <li>49 - communication module data overrun MP1</li> <li>50 - communication module data overrun MP2</li> <li>51 - communication module data overrun MP3</li> <li>58 - communication module data overrun messages</li> </ul>

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 communicator 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 = 8

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
7	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 = 4

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
1	cmp1	cmp2	cmp3	pmp1	pmp2	pmp3	mmp1	mmp2	mmp3	crun
2	cmp1	cmp2	cmp3	pmp1	pmp2	pmp3	mmp1	mmp2	mmp3	crun
3	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 = 4

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
1	cmp1	cmp2	cmp3	pmp1	pmp2	pmp3	mmp1	mmp2	mmp3	crun
2	cmp1	cmp2	cmp3	pmp1	pmp2	pmp3	mmp1	mmp2	mmp3	crun
3	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 = 4  
BPR = 10  
FD0 = SBINARY + 4, 1  
FPR = 1

Tabular display:

	0	1
0	cro	cum
1	cro	cum
2	cro	cum
3	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 = 16  
 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
15	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 = 16  
 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
15	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 = 16  
 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
15	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 = 16  
 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
15	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 = 8  
 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
7	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 PSTN 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 MODEM 1 - CHANNEL 1
02	events - communication channel MODEM 2 - CHANNEL 2
03	events - communication channel RS485 - CHANNEL 3
04	events - communication channel CS 1 - CHANNEL 4
05	events - communication channel CS 2 - CHANNEL 5
06	events - communication channel CS 3 - CHANNEL 6
07	events - communication channel CS 4 - CHANNEL 7

### 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 last reading.	UBINARY	1
SEQ	Number of the meter reading till now. The number is read from the meter. Possible values are from 0 to 31.	UBINARY	1

The STATUS field can have the following values:

- 0 - data is 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 - meter counter manual adjustment
- CY - Carry - meter counter has been inverted

Individual registers represent:

<b>register</b>	<b>meaning</b>
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 IEC1107MREADATTR

A group of registers involves information on reading via communication channels, determined with parameters in a group of registers IEC1107MREADPARAM.

GID = 83  
 ATTR = R  
 RPG = 8  
 BPR = 34  
 FD0 = UBINARY + 4, 2  
 FD1 = ASCII + 4, 1  
 FD2 = UBINARY + 2, 1  
 FD3 = UBINARY + 1, 1  
 FD4 = ASCII + 17, 1  
 FPR = 6

Tabular display:

	0	1	2	3	4	5
0	RT	RC	MA	B	MO	ID
7	RT	RC	MA	B	MO	ID

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

Contents of fields:

Field	Contents	Type	Length
RT	Time of the last meter read-out expressed as a number of seconds from 1.1.1970 from 00:00	UBINARY	4
RC	Successive number of the last read-out	ASCII	4
MA	Manufacturer	UBINARY	4
B	Transmission rate (baud)	UBINARY	2
MO	Transmission type in compliance with IEC1107 (A, B, C or D)	UBINARY	1
ID	Meter identification string	ASCII	17

Information on the last meter read-out with a tariff device in compliance with IEC1107 involves time of the last read-out (RT), a successive number of read-out (RC) as well as data on identification of the read-out meter like the manufacturer (MA), transmission rate (B), transmission type (MO) and meter identification string (ID).

#### 4.22 IEC1107MRESETATTR

The group contains the resetting attributes of a meter with tariff device. Parameters are set in the IEC1107MRESETPARAM group of registers.

GID = 84  
 ATTR = R  
 RPG = 1  
 BPR = 6  
 FD0 = UBINARY + 4, 1  
 FD0 = UBINARY + 1, 1  
 FPR = 2

Tabular display:

	0	1
0	RT	ST

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

Contents of fields:

Field	Contents	Type	Length
RT	Time of the last reset expressed as a number of seconds from 1.1.1970 from 00:00	UBINARY	4
ST	Condition of relay which is changed whenever time alarm is activated (field value can be 0 or 1)	UBINARY	1

For resetting of meters with a tariff device the following is defined in a group of registers OPORTPARAM for a corresponding relay output:

- for a relay with change-over contacts

	0	1	2	3	4
109.X	2	0	84	0	1

- for two relays with single contact

	0	1	2	3	4
109.X	2	0	84	0	1
109.(X+1)	2	1	84	0	1

### 4.23 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 4-24 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" <ul style="list-style-type: none"> <li>system message from the ALARM messages group if "ALARM" was "a cause"</li> <li>A channel through which data from the meter were read-out if the cause was "IEC1107DATA"</li> </ul>	UBINARY	1

#### 4.24 DLCATTR

The group contains attributes of DLC modem.  
Parameters are set in the DLCPARAM group of registers.

GID = 88  
ATTR = R  
RPG = 11  
BPR = 2  
FD0 = UBINARY + 2, 1  
FPR = 1

Tabular display:

	0
0	REVISION
1	TYPE
2	ADDRESS
3	MODE
4	RATE
5	TUNE
6	GAIN
7	CONNECT COUNT
8	CONNECT FAIL COUNT
9	RESET COUNT
10	ERROR

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

Contents of fields:

Field	Contents	Type	Length
REVISION	Version of the modem software installed in the device	UBINARY	2
TYPE	Type of station: 0 - SLAVE 1 - MASTER	UBINARY	2
ADDRESS	Modem address If the station type is that of SLAVE, then there are values from 1 to 254 (Rev. 1.X) in the field and / or from 1 to 4095 (Rev. 2.X). If the station type is that of MASTER, then there are values from 0 to 3 (Rev. 1.X) in the field and / or from 0 to 7(Rev. 2.X).	UBINARY	2
MODE	Operation mode of DLC modem if it is set as a MASTER station: 0 - TEST 1 - CONTINUOUS Permanent connection between MASTER station and one of SLAVE stations. 2 - AUTOPOLL A device calls SLAVE stations one by one according to the previously determined list. 3 - BROADCAST A device simultaneously sends messages to all SLAVE stations which belong to it. 4 - CONNECTED If the DLC modem is set like the SLAVE station, then the single valid value is 4, which appears at successful modem connection with the modem in the MASTER station.	UBINARY	2
RATE	Data transfer speed between the two modems (bits/s).	UBINARY	2
TUNE	The number of currently used frequency couple for data transfer.	UBINARY	2
GAIN	Modem input gain.	UBINARY	2
CONNECT COUNT	The number of attempts of reestablishing a connection to all substations. If the DLC modem is set like the SLAVE station, the values in this field are not essential!	UBINARY	2
CONNECT FAIL COUNT	The number of failed attempts of reestablishing a connection to all substations. If the DLC modem is set like the SLAVE station, the values in this field are not essential!	UBINARY	2
RESET COUNT	The number of resets. A reset is activated by the processor.	UBINARY	2
ERROR	Type of error 0 - NONE No error. 1 - NAK The last communication between DLC modem and a device was not successful. 2 - TIMEOUT Error on a communication line between a DLC modem and a device.	UBINARY	2

### 4.25 TARSTAT

The group contains status information on tariffs.

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

Tabular display:

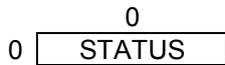


Table 4-26 Group of registers **Napaka! Slog ni definiran.**

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







### 4.29 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:

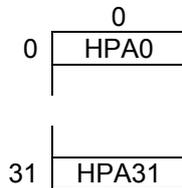


Table 4-30 Group of registers **Napaka! Slog ni definiran.**

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

### 4.30 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:

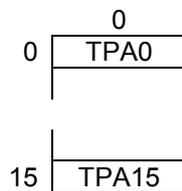


Table 4-31 Group of registers **Napaka! Slog ni definiran.**

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

### 4.31 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 4-32 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

### 4.32 IPORTATTR

The group contains the attributes of input impulse lines.

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

Tabular display:

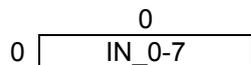


Table 4-33 Group of registers **Napaka! Slog ni definiran.**

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

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 3	IN 2	IN 1	IN 0

- IN 0 - CS input 1
- IN 1 - CS input 2
- IN 2 - CS input 3
- IN 3 - CS input 4
- - not used

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).

### 4.33 OPORTATTR

The group contains the attributes of output lines.

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

Tabular display:

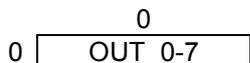


Table 4-34 Group of registers **Napaka! Slog ni definiran.**

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 5	OUT 4	OUT 3	OUT 2	OUT 1	OUT 0

- OUT 0 - relay output 1
- OUT 1 - relay output 2
- OUT 2 - CS output 1
- OUT 3 - CS output 2
- OUT 4 - CS output 3
- OUT 5 - CS output 4
- - not used

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).

### 4.34 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 4-35 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

### 4.35 CONFIGPORT

A group involves parameters of CS lines for determining line type. Setting does not influence in the last four registers of register groups IPORTPARAM, OPORTPARAM and CHANNELPARAM. Each parameter excludes other two parameters.

GID = 107  
 ATTR = R, W, WP, WL, PAR, EE  
 RPG = 4  
 BPR = 1  
 FD0 = UBINARY + 1, 1  
 FPR = 1

Tabular display:

	0
0	line type
1	line type
2	line type
3	line type

Table 4-36 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
line type	The following values are possible: 0 - NULL                   line not used 1 - INPUT/OUTPUT       input/output 2 - UART                   communication line	UBINARY	1

If line type is set to 1 (INPUT/OUTPUT), the line can be used only as INPUT (group of registers IPORTPARAM) or as OUTPUT (group of registers OPORTPARAM)!

register	meaning
00	parameter for line 0 (CS 1)
01	parameter for line 1 (CS 2)
02	parameter for line 2 (CS 3)
03	parameter for line 3 (CS 4)

### 4.36 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.

CS lines should be defined in a group of registers CONFIGPORT as INPUT/OUTPUT!

GID = 108  
 ATTR = R, W, WP, WL, PAR, EE  
 RPG = 4  
 BPR = 3  
 FD0 = UBINARY + 1, 3  
 FPR = 3

Tabular display :

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

Table 4-37 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 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 (CS 1)
01	parameter for input 1 (CS 2)
02	parameter for input 2 (CS 3)
03	parameter for input 3 (CS 4)

### 4.37 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).

CS lines should be defined in a group of registers CONFIGPORT as INPUT/OUTPUT!

GID = 109  
 ATTR = R, W, WP, WL, PAR, EE  
 RPG = 6  
 BPR = 5  
 FD0 = UBINARY + 1, 5  
 FPR = 5

Tabular display :

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

Table 4-38 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. 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 relay output 1
01	parameter for relay output 2
02	parameter for output 0 (CS 1)
03	parameter for output 1 (CS 2)
04	parameter for output 2 (CS 3)
05	parameter for output 3 (CS 4)

4.38 LEVOPARAM

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:

$$\frac{k_{VH}}{k_{REG}} = \frac{a}{b}$$

$k_{IN}$  [energy/impulse] is the constant of the meter and  $k_{REC}$  [energy/impulse] is the constant of communicator.

- GID = 110
- ATTR = R, W, WP, 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	type_input	index
7	tip	correction a	correction b	type_input	index

Table 4-39 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
Type	Parameter defining the type of result correction:	UBINARY	2
	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		

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

### 4.39 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:

$$R1_i = a_{i,0}R0_0 + a_{i,1}R0_1 + \dots + a_{i,n-1}R0_{n-1}$$

Parameters are  $a_{i,0}, \dots, a_{i,n-1}$ , 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 = 4  
 BPR = 4  
 FD0 = UBINARY + 2, 1  
 FD1 = UBINARY + 1, 1  
 FPR = 2

Tabular display :

	0	1
0	type	SIGN0-7
1	type	SIGN0-7
2	type	SIGN0-7
3	type	SIGN0-7

Table 4-40 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
Type	Parameter defining the type of result correction:	UBINARY	2
	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!		
	7 - UD COUNTER D Positive and negative changes are		

Field	Contents	Type	Length
	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

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 <sub>7,0</sub>	a <sub>6,0</sub>	a <sub>5,0</sub>	a <sub>4,0</sub>	a <sub>3,0</sub>	a <sub>2,0</sub>	a <sub>1,0</sub>	a <sub>0,0</sub>	
SIGN0-7	=	0	0	0	0	1	1	1	1	= 0Fh = 15

#### 4.40 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 = 4  
 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
1	type	a	b	level A	result A	c	d	level B	result B	H
2	type	a	b	level A	result A	c	d	level B	result B	H
3	type	a	b	level A	result A	c	d	level B	result B	H

Table 4-41 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	UBINARY	2

Field	Contents	Type	Length
	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 - measuring level 0 1 - measuring level 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 - measuring level 0 1 - measuring level 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

#### 4.41 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  $a_i$ ,  $b_i$  and  $A_i$ , 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 = 4  
 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
1	type	correction a	correction b	level A	result A
2	type	correction a	correction b	level A	result A
3	type	correction a	correction b	level A	result A

Table 4-42 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_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

### 4.42 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 = 16  
 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
15	TR	PRR	RR	TBP	IBP	OTBP	TT	TI	CL	RC0	...	RC7

Table 4-43 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
TR	Type of result: 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 ( $V = MPV \times 3600 / MP$ )	UBINARY	1
PRR	Period of result renewal: 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	UBINARY	1
RR	Renewal of a result : 0 - NULL is not renewed 1 - WRITE $BV = V$ 2 - CUMULATE $BV = BV + V$ 3 - MAX WRITE if $V > BV \Rightarrow BV = V$ 4 - MAX CUMULATE if $V > BV \Rightarrow BV = BV + V$ 5 - MAX2 WRITE 6 - MAX2 AVERAGE 7 - MAX3 WRITE 8 - MAX3 AVERAGE	UBINARY	1
TBP	Type of billing period: 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	UBINARY	1
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.

### 4.43 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.

CS lines should be defined in a group of registers CONFIGPORT as UART!

GID = 180  
 ATTR = R, W, WP, PAR, EE  
 RPG = 8  
 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
7	T	F	BR	DB	P	SB

Table 4-44 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
T	Type of connection 0 - Direct 1 - AT modem (also ISDN terminal adapter) 2 - DLC modem 3 - reserved for the CATV 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 - 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 MODEM 1 - CHANNEL 1
02	communication channel MODEM 2 - CHANNEL 2
03	communication channel RS485 - CHANNEL 3
04	communication channel CS 1 - CHANNEL 4
05	communication channel CS 2 - CHANNEL 5
06	communication channel CS 3 - CHANNEL 6
07	communication channel CS 4 - CHANNEL 7

#### DLC modem

A communicator is supported by only one DLC modem which can be connected only to a CHANNEL 1 communication channel.

Parameters BR, DB, P and SB are not important at a DLC modem application! The DLC modem requires the following settings which are set automatically:

BR = 1200  
DB = 8  
P = 78 (NONE)  
SB = 1

Due to specific communication via a network the application of communication functions which do not depend on time frames are recommended (Terminal, DIN19244).

#### ISDN terminal adapter

ISDN terminal adapter has not its own communication type. AT modem is applied (parameter T; value 1).

#### Meter reading according to IEC1107

Recommended settings for meter reading according to IEC1107 are:

F = 5 (IEC1107Master)  
BR = 300 (initial rate)  
DB = 7  
P = 69 (Even)  
SB = 1

The transmission rate via the line (a BR parameter) represents the transmission initial rate at meter reading in compliance with IEC 1107 protocol. It depends on the communication mode and the meter. For "Meter Command" and "Mode D" the initial rate is 300 bauds. If the meter does not support the rate change-over, the initial rate is equal to the meter rate.

#### 4.44 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 4-45 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 communicator sends its initialisation string. If a modem sends OK message, the communicator sends initialisation string which is defined in this register. Initialisation string, defined in the communicator, 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!

#### 4.45 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 4-46 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 - MODEM 1 2 - CHANNEL 2 - MODEM 2 3 - CHANNEL 3 - RS485 4 - CHANNEL 4 - CS 1 5 - CHANNEL 5 - CS 2 6 - CHANNEL 6 - CS 3 7 - CHANNEL 7 - CS 4 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:

<b>register</b>	<b>meaning</b>
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

#### 4.46 IEC1107MREADPARAM

A group involves parameters for meter reading in compliance with IEC1107 standard.

GID = 183  
 ATTR = R, W, WP, PAR, EE  
 RPG = 8  
 BPR = 65  
 FD0 = UBINARY + 1, 2  
 FD1 = ASCII + 17, 1  
 FD2 = ASCII + 9, 1  
 FD3 = ASCII + 3, 1  
 FD4 = ASCII + 33, 1  
 FD5 = UBINARY + 1, 1  
 FPR = 7

Tabular display:

	0	1	2	3	4	5	6
0	TYPE	CHANN	MA	PASS	CO	CO_M	AL
7	TYPE	CHANN	MA	PASS	CO	CO_M	AL

Table 4-47 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
TYPE	The function is defined by the channel type 0 Inactive 1 Meter reading meter reading on demand from the device 2 »D mode« only meter reading; the device waits for data 3 Meter command command mode of work; beside meter reading, meter programming is also possible	UBINARY	1
CHANN	Defines the number of communication channel used for meter reading: 0 - OPTICAL 1 - CHANNEL 1 - MODEM 1 2 - CHANNEL 2 - MODEM 2 3 - CHANNEL 3 - RS485 4 - CHANNEL 4 - CS 1 5 - CHANNEL 5 - CS 2 6 - CHANNEL 6 - CS 3 7 - CHANNEL 7 - CS 4	UBINARY	1
MA	Identification number of the meter; its maximum length is 16 characters	ASCII	17
PASS	Password for access to the meter; its maximum length is 8 characters	ASCII	9
CO	Basic command according to the IEC1107 standard of maximum 2 characters	ASCII	3
CO	Command arguments according to the IEC1107 standard	ASCII	33
AL	Index on time alarm in the ALARMPARAM group of registers	UBINARY	1

A register represents the parameters for reading of meters with tariff devices according to the IEC1107 standard. Several registers can be connected to the same communication channel. As the communication channel has only one buffer store for the IEC1107 telegram, old data are copied at every fresh data reading. For this reason, reading of several meters through the same communication channel should be synchronized with reading of data from temporary memory.

It can be determined when (AL) and through which communication channel (CHANN) the reading from the meter with a particular identification (MA) will be carried out.

Basic communication setups for meters reading through the CS channels are:

- The CONFIGPORT group of registers, registers from 0 (CS 1) to 3 (CS 4):
  - Line type = 2 (UART)
  
- The CHANNELPARAM group of registers, registers from 0 to 3:
  - F (Function) = 0 (Terminal) or 3 (DIN19244Slave)
  - BR (Baud rate) = 300 - 57600
  - DB (Data Bits) = 8
  - P (Parity) = 78 (None) for Terminal or 69 (Even) for DIN19244
  - SB (Stop Bits) = 1
  
- The CHANNELPARAM group of registers, registers from 4 (CS 1) to 7 (CS 4):
  - T (Type) = 0 (direct)
  - F (Function) = 5 (IEC1107Master)
  - BR (Baud rate) = 300
  - DB (Data Bits) = 7
  - P (Parity) = 69 (Even)
  - SB (Stop Bits) = 1

The setups in details may be also different. It is essential only that the setups on both devices are equal.

#### 4.47 IEC1107MRESETPARAM

A group involves parameters for resetting of meters with a tariff device with which it is set when a state of reset signal is to be changed in the ST field in a group of registers IEC1107MRESETATTR.

GID = 184  
 ATTR = R, W, WP, PAR, EE  
 RPG = 1  
 BPR = 2  
 FD0 = UBINARY + 1, 2  
 FPR = 4

Tabular display:



Table 4-48 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents		Type	Length
TYPE	0	INACTIVE	UBINARY	1
	1	ACTIVE		
AL	Index for time alarm in a group of registers ALARMPARAM		UBINARY	1

When time alarm occurs, all meters with a tariff device are reset. They are connected to the output which follows ST field in a group of registers IEC1107MRESETATTR.

### 4.48 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 4-49 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.

#### 4.49 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:

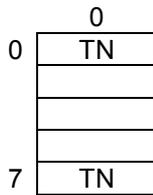


Table 4-50 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
TN	Telephone number of called station which is max. 20 characters long	ASCII	20

### 4.50 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 4-51 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 - MODEM 1 2 - CHANNEL 2 - MODEM 2 3 - CHANNEL 3 - RS485 4 - CHANNEL 4 - CS 1 5 - CHANNEL 5 - CS 2 6 - CHANNEL 6 - CS 3 7 - CHANNEL 7 - CS 4	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: <ul style="list-style-type: none"> <li>systemic message from the ALARM messages group, if "ALARM" is "a cause"</li> <li>a channel through which data from a meter were read out if the cause is "IEC1107DATA"</li> </ul>	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” <b>and</b> “a cause” are fulfilled

### 4.51 DLCPARAM

A group involves parameters which are required by the device for communication via a DLC modem. Parameters consist of a station type, a MASTER station and SLAVE station addresses.

GID = 188  
 ATTR = R, W, WP, PAR, EE  
 RPG = 1  
 BPR = 4  
 FD0 = UBINARY + 1, 4  
 FPR = 4

Tabular display:



Table 4-52 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
TYPE	Type of station: 0 - SLAVE 1 - MASTER	UBINARY	1
MASTER ADDR.	MASTER station address: Rev. 1.X: 0 - 3 Rev. 2.X: 0 - 7	UBINARY	1
RELAY HOPS	Switching on the relay mode of modem operation. The parameter is determined by the number of relay stations between the MASTER station and the station called. Possible values: 1 - 8 Value 0 (nought) signifies switching off the relay mode of operation. The parameter refers to the MASTER station only!	UBINARY	1
RETRY COUNT	Maximum number of attempts to reestablish connections with the subordinate station. Possible values: 1 - 8 The acquired value is 0 (nought) and determines the maximum number of attempts (8). The parameter refers to the MASTER station only!	UBINARY	1

Between parameters the following relations are valid:

TYPE	MASTER ADDR.
0 (SLAVE)	It defines to which MASTER station a device belongs. The device responds only to calls from this MASTER station.
1 (MASTER)	it defines the device address in a network.

### 4.52 CASCADE

A group involves parameters for communication function Cascade.

GID = 189  
 ATTR = R, W, WP, PAR, EE  
 RPG = 1  
 BPR = 2  
 FD0 = UBINARY + 1, 2  
 FPR = 2

Tabular display:

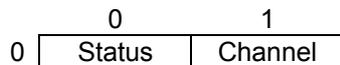


Table 4-53 Group of registers **Napaka! Slog ni definiran.**

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 - MODEM 1 2 - CHANNEL 2 - MODEM 2 3 - CHANNEL 3 - RS485 4 - CHANNEL 4 - CS 1 5 - CHANNEL 5 - CS 2 6 - CHANNEL 6 - CS 3 7 - CHANNEL 7 - CS 4	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)!

### 4.53 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
31	hour	minute	hour	minute	...	hour	minute	daylight

Table 4-54 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.

#### 4.54 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 4-55 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.

**4.55 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
0	SY	SM	SD	SWD	SADR	SH	SN	SS	SATR
15	SY	SM	SD	SWD	SADR	SH	SN	SS	SATR
	9	10	11	12	13	14	15	16	17
0	TY	TM	TD	TWD	TADR	TH	TN	TS	TATR
15	TY	TM	TD	TWD	TADR	TH	TN	TS	TATR

Table 4-56 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 yearlins repeated each year on a day and month; if Sunday, it is moved to Monday
- 4 yearlyeast 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.

### 4.56 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 4-57 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 yearlyeast 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

### 4.57 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 4-58 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.

### 4.58 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 4-59 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 communicator. 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 communicator.

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	yearlyeast	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.**

### 4.59 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 = 8  
 BPR = 9  
 FD0 = SBINARY + 1, 1  
 FD1 = ASCII + 8, 1  
 FPR = 2

Tabular display:

	0	1
0	exponent	text
7	exponent	text

Table 4-60 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:

Communicator internal unit : 0.1 Wh (one impulse in communicator 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

### 4.60 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 = 16  
 BPR = 40  
 FD0 = ASCII + 40, 1  
 FPR = 1

Tabular display:

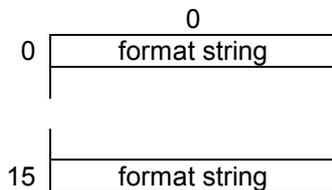


Table 4-61 Group of registers **Napaka! Slog ni definiran.**

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 differentiated value in a measuring period
  - r multiple application of format string
  - 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 text
  - S profile record stamp
  - P a field of register in a profile of the last terminated saving period

### 4.61 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 = 16  
 BPR = 11  
 FD0 = UBINARY + 1, 11  
 FPR = 11

Tabular display:

	0	1	2	...	9	10
0	FS	ARG0	ARG1		ARG8	ARG9
15	FS	ARG0	ARG1		ARG8	ARG9

Table 4-62 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

### 4.62 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 = 8  
 BPR = 16  
 FD0 = UBINARY + 1, 16  
 FPR = 16

Tabular display :

	0	1	2	...	14	15
0	Fo0	Fo1	Fo2	...	Fo14	Fo15
7	Fo0	Fo1	Fo2	...	Fo14	Fo15

Table 4-63 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
...			
Fo15	Index of format output 15 in a list	UBINARY	1

### 4.63 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 = 8  
 BPR = 5  
 FD0 = UBINARY + 1, 5  
 FPR = 5

Tabular display:

	0	1	2	3	4
0	FL	device	channel	alarm	delay
7	FL	device	channel	alarm	delay

Table 4-64 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	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 - MODEM 1
- 2 - CHANNEL 2 - MODEM 2
- 3 - CHANNEL 3 - RS485
- 4 - CHANNEL 4 - CS 1
- 5 - CHANNEL 5 - CS 2
- 6 - CHANNEL 6 - CS 3
- 7 - CHANNEL 7 - CS 4

**4.64 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 4-65 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
L	Location - media to which data are being saved 0 - NULL are not saved 1 - RAM internal RAM memory 2 - EEPROM internal EEPROM memory 3 - FLASH internal FLASH memory	UBINARY	1
P	Percent of pool which will be occupied	UBINARY	1
SD	Saving event : 0 - NULL is not saved 1 - Measuring period termination of measuring period triggers saving 2 - Alarm internal alarm triggers saving	UBINARY	1
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 rid - identification of the first register in a block of registers fid - identification of the first register field in a block of registers nre - number of registers in a block of registers nfi - number of fields in a block of registers 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.

### 4.65 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 storing 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 4-66 Group of registers **Napaka! Slog ni definiran.**

Contents of fields:

Field	Contents	Type	Length
L	Location - media to which data are being saved: 0 - NULL are not saved 1 - RAM internal RAM memory 2 - EEPROM internal EEPROM memory 3 - FLASH internal FLASH memory	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.

### 4.66 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 4-67 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

### 4.67 POOLATTR

This group contains information on location and the size of pool of the memory medium.

GID = 213  
 ATTR = R  
 RPG = 4  
 BPR = 8  
 FD0 = UBINARY + 4, 2  
 FPR = 2

Tabular display:

	0	1
0	Position	Size
1	Position	Size
2	Position	Size
3	Position	Size

Table 4-68 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

## 5. TECHNICAL DATA

### 5.1 CS LINES

CS lines comply with DIN 66348 standard. They are performed as a current loop with one supply source (CL1). All lines have minus terminals (21,23, 25 and 27) connected. CS lines can act as inputs, outputs or communication lines.

#### 5.1.1 CS input lines

Functions of input lines are universal for all types of inputs (meter impulse inputs, serial inputs, control inputs).

There are the following types:

a) Two-wire input in compliance with DIN 43864

Meters with pulse transmitters are connected.

b) Two-wire input for active current

The same characteristics are valid as for input which complies with DIN 43864. Longer impulse lines are possible. During the active signal a circuit on signal lines is closed.

Maximal values:

- maximal frequency  $f_{\max} = 10 \text{ Hz}$
- minimal impulse length  $t_{\min} = \text{program adjustable}$

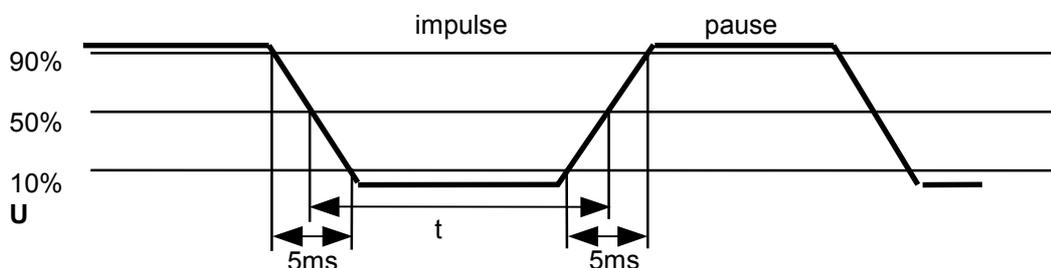


Figure 5-1 Time diagram - Two-wire input for active current

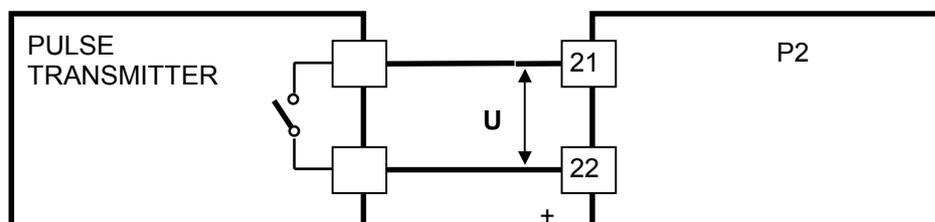


Figure 5-2 Connection scheme for a typical two-wire input for active current

**c) 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.

**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 frequency  $f_{max} = 40 \text{ Hz}$
- transition time  $t_{min} = \text{program adjustable}$

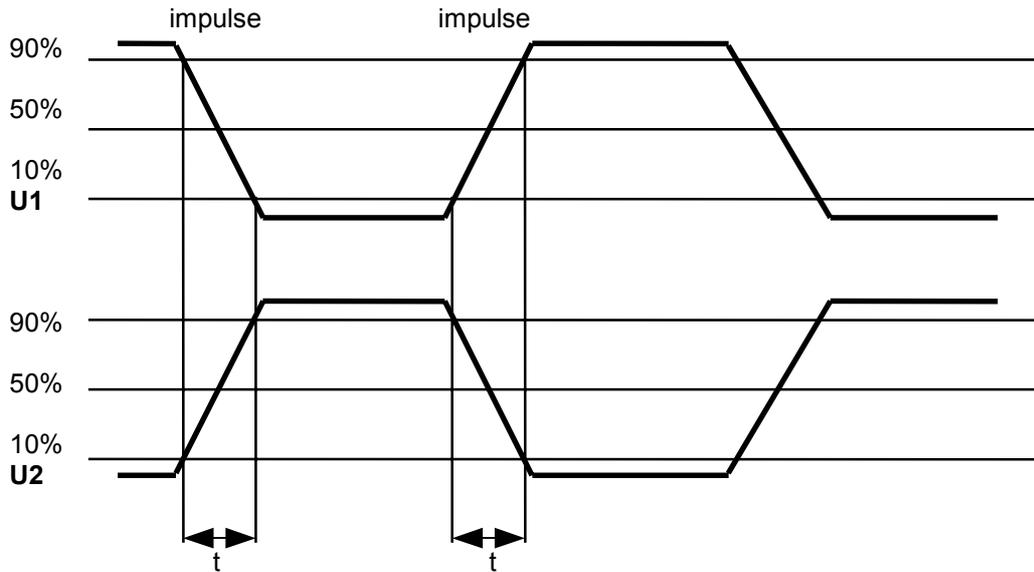


Figure 5-3 Time diagram - Three-wire input

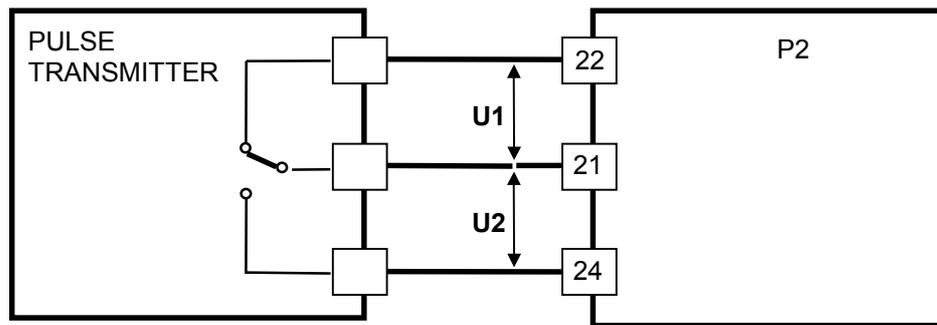


Figure 5-4 Connection scheme for a typical three-wire input

**5.1.1.1 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  $\pm 0,5$  minute

### 5.1.2 CS output lines

Impulse or serial meter data can be transmitted. They can be used as control outputs together with corresponding adapters.

### 5.1.3 CS communication lines

They can be used for serial communication on the basis of a current loop. Max. current is 20 mA. Max. communication rate depends on the distance and conditions and is being decreased by distance.

## 5.2 RELAY OUTPUTS

Relay outputs are performed as passive outputs in a form of a passive contact by means of a semi-conductor relay and a bistable mercury relay. Relay outputs are programmable. They can take a function of impulse or signal outputs. They can take a function of impulse outputs or signal outputs.

Type and characteristics of switching:

parameter	semiconductor relay	bistabil relay	unit
maximal permanent current	0,15	8	A
range of switching voltage	0,1 - 250	250	V
life time limited with mechanical characteristics	-	$10^7$	switchings
life time at maximal electric loads	-	$10^5$ (8A, 250V AC)	switchings
internal resistance of closed contact (protection resistance):	16	1	$\Omega$
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

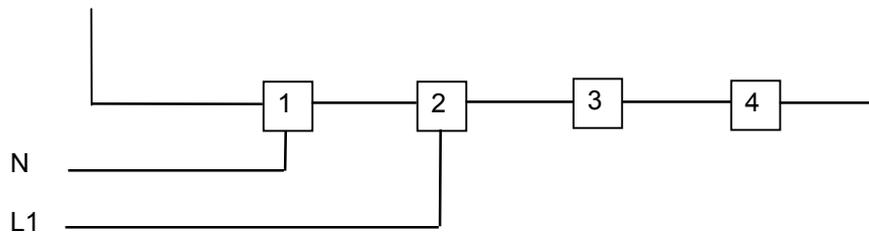
**5.3 POWER SUPPLY**

**5.3.1 Main supply**

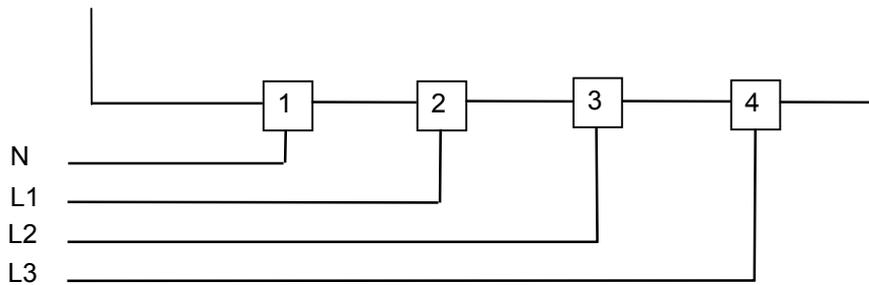
	single-phase	three-phase
1 <sup>st</sup> range of power supply voltage	100 - 230 V	3 x 100 V - 3 x 230 V
2 <sup>nd</sup> range of power supply voltage	58 - 100 V	3 x 58 V - 3 x 100 V
power supply tolerance	-10, +15%	
range of power supply voltage frequency	45 - 65 Hz	
consumption in one phase	max. 10 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 (range 1 and 2)



b) Three-phase power supply (range 1 and 2)



By the power supply for the 2<sup>nd</sup> range the N line is not compulsory.

c) phase-to-phase supply (range 1 only)

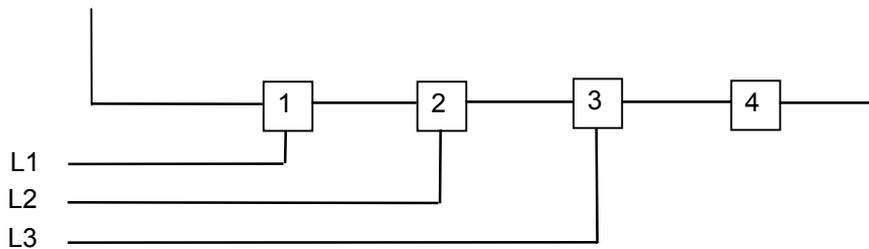


Figure 5-5 Connection scheme - Power supply

### 5.3.2 Internal stand-by power supply

Internal clock is protected with capacitor and keeps real time in case of main power supply failure.

Characteristics of back-up capacitor:

- capacity of Li-battery: 1 F
- time of real time keeping when other power supply sources are switched off: 14 days
- self-discharging by 25°C : 1 % per year

### 5.3.3 Supply of CS lines

Supply of CS lines (e.g. meter inputs) complies with IEC 1107 standard.

## 5.4 COMMUNICATION

### 5.4.1 RS485

The interface RS485 is integrated in a device and is intended for connection with a PC and/or meters into a local network in compliance with EIA RS485 standard.

General data:

- transmission rate: 150 - 57600 bit/s
- transmission mode : two-wire
- maximal line length : 1200 m
- maximal number of communication places : 32 (computer + 31 devices)

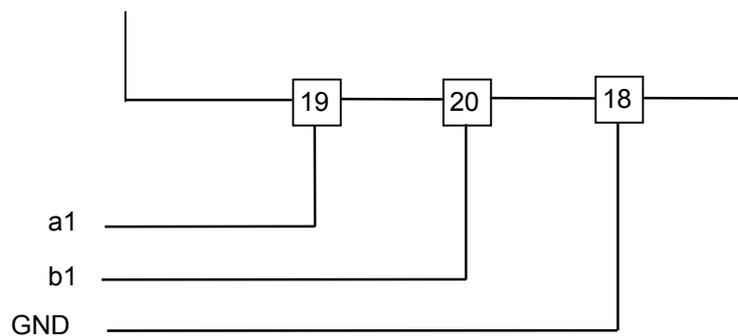


Figure 5-6 Connection scheme - RS485

### 5.4.2 RS232C

An RS232C interface is performed in a form of a replaceable module and is intended for a direct communication with external DCE devices in compliance with the EIA RS232C standard.

General data:

- transmission rate: 150 - 57600 bit/s
- transmission mode: asynchronously, full duplex
- maximal length of lines: 15 m

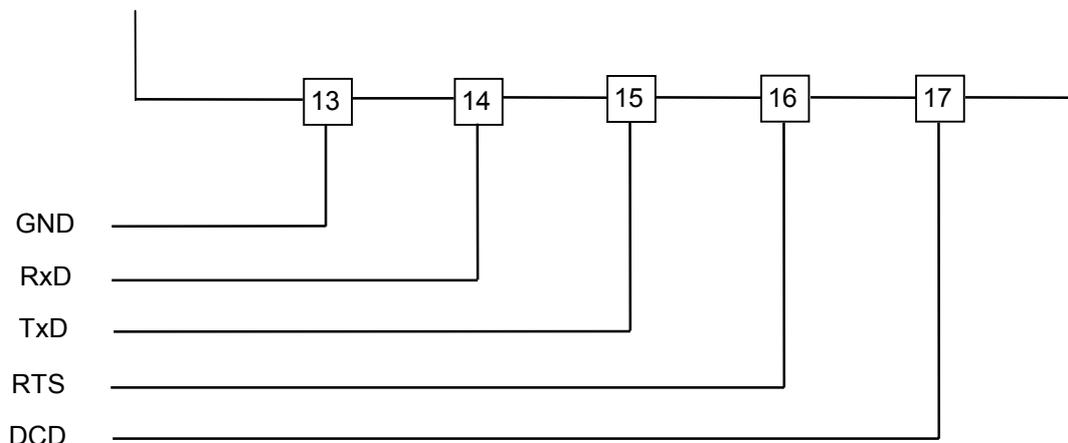


Figure 5-7 Connection scheme - RS232

### 5.4.3 PSTN modem

The interface for a PSTN modem is performed as an exchangeable module and is intended for communication over PSTN.

Standards:

CCITT V.21, V.22, V.22bis and V.23  
Bell 212A and 103  
V.42 LAPM and MNP 2-4  
V.42bis and 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 <sub>RMS</sub> , max. 120 V <sub>RMS</sub> |
| • ringing frequency                        | 15 - 68 Hz   |
| • input impedance for a two-wire line      | 900 Ω (by 1 kHz)                                     |

### 5.4.4 DLC Modem

It is intended for communication over low-voltage network. Connection terminals are the same as for communicator supply.

General data:

- |   |   |
|---|---|
| • communication type                      | Master - Slave  |
| • protocol                                | modified IEC 1107   |
| • meter-modem transmission rate           | 1200 bit/s  |
| • transmission rate over electric network | 400, 800, 1212 and 3200 bit/s   |
| • selection of transmission rate          | manual or automatic   |
| • modulation type                         | FSK, frequency jump, 4-times by 2 frequency channels, transmission channel adjustable manually or automatically |
| • used frequency band                     | 9 - 95 kHz  |
| • used CRC code                           | standard industrial CRC-24 code   |
| • connection of modem or meter to network | SLAVE single-phase<br>MASTER three-phase  |
| • setting range                           | MASTER has 8 addresses, 4094<br>SLAVE units belong to each address;<br>32752 units in total                     |
| • max. line length                        | secondary side of supply transformer<br>10kV/230V   |
| • meters calling mode                     | auto-polling, continuous connection and broadcast call  |
| • consumption                             | 2 VA  |
| • max. number of communication positions  | 32 (communicator + 31 meters)   |

### 5.4.5 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 communicator where it is fixed with a magnet.

- Transmission rate 2400 bit/s.
- Maximal permitted probe shift from a stated position 30°

### 5.4.6 ISDN terminal adapter

The interface for the ISDN terminal adapter is designed as a changeable module. The module enables the connection between an asynchronous interface with the TTL levels and the SO ISDN interface. It is intended for the communication in the ISDN network through the basic BRI access (Basic Rate Interface - 2B + D).

The terminal adapter is designed in accordance with the following ISDN recommendations and standards:

#### S0 interface

- I.430 and I.431 OSI layer 1
- Q.920(I.440), Q.921(I.441) and ETS 300 125 OSI layer 2
- Q.930(I.450), Q.931(I.451) and ETS 300 102 and X.31 OSI layer 3

#### Technical data:

- ISDN interface: S0
- D- channel - protocols:  
X.31 (X.25 in D-channel)  
DSS1 (with the implemented UUS and SUBADDRESSING service)  
Instead of DSS1 also others possible at one's option, national protocols(1TR6,...)
- B- channel - protocols:  
V.110 1200 - 19200 bps  
HDLC asynchronous / synchronous conversion - PPP  
HDLC transparent transfer (octets folded in the HDLC frame)  
Byte transparent transfer  
X.75 - data transfer 64000 bit/s  
and the TA remote control  
V.120  
X.31 (X.25 in B-channel)  
ML PPP (connecting the B-channels) 2 × 64000 bit/s
- Serial interface:  
Functionalism: V.24,  
Electrical characteristic: TTL nivoji  
Transfer speed: 1200 - 230400 bit/s asinhrono  
Transfer speed setup: automatic (autobauding) or  
through AT commands
- Procedures of selection and configuration:  
AT commands (Hayes standard)  
PAD X.3  
Hotline 108 (DTR)  
Connect Always (permanent connection)  
Configurator
- Setting the parameters:  
By means of an asynchronous terminal connected to a serial interface (TTL levels)  
By remote control through the ISDN network  
Configuration parameters are stored in the non-volatile memory
- Entering the code:  
Locally through serial interface  
By remote control through the ISDN network
- Power supply:  
Voltage + 5V  
Max. power consumption < 250 mA
- Approvals:  
Europe (the approval does not include Switzerland)

## 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 capacitor.

- 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

Internal memory is EEPROM type (8 kbytes) and static RAM (128 kbytes).

## 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 5.6 mm high.

## 5.8 OPERATION CONDITIONS

### 5.8.1 Ambient temperature

- working temperature range: 0°C - +50°C
- temperature of storing - 20°C - +65°C

### 5.8.2 Air humidity

- relative air humidity  
at application: max. 75 %
- 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/OUTPUTS no. of terminals: 21, ... , 28
- RELAY OUTPUTS no. of terminals: 7, 8, 10, 11, 12
- POWER SUPPLY no. of terminals: 1, 2, 3, 4
- RS 485 no. of terminals: 18, 19, 20
- RS232C (when a module is inserted) no. of terminals: 13, ... , 17
- DLC MODEM (when a module is inserted) no. of terminals: 1, 2, 3, 4
- POST MODEM (when a module is inserted) no. of terminals: 2, 3, 4, 5 (RJ13)
- ISDN TA (when a module is inserted) no. of terminals: 3, 4, 5, 6 (RJ45)
- 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	$500\sqrt{2}V$	2000V	$2000\sqrt{2}V$

**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 6 kV in a longitudinal direction in all circuits for the following groups of electric terminals:
  - RELAY OUTPUTS
  - SUPPLY
- b) A test is performed with amplitude 6 kV in transverse direction between the following groups of electric terminals and earth.
  - RELAY OUTPUTS
  - SUPPLY
- c) A test is performed in a transverse direction
  - with amplitude 6 kV
    - between RELAY OUTPUTS and other groups
    - between SUPPLY and other groups
  - with amplitude 1 kV
    - between POST MODEM and RS485, INPUTS/OUTPUTS
    - between ISDN TA and RS485, INPUTS/OUTPUTS
    - between RS485 and RS232C, INPUTS/OUTPUTS
    - between RS232C and INPUTS/OUTPUTS

Groups of electric terminals are defined in Item INSULATION STRENGTH.

### 5.9.3 High frequency disturbance test

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:
  - RELAY 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:
  - RELAY OUTPUTS
  - POWER SUPPLY
- c) A test is performed with amplitude 2.5 kV in transverse direction between different groups of electric terminals:
  - RELAY OUTPUTS
  - POWER SUPPLY

### 5.9.4 Resistance to fast voltage surges - burst

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 between all SUPPLY and RELAY OUTPUTS terminals and earth.
- b) A test is performed with amplitude 1 kV in all other electric terminals. 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 interferences, generated by the device, complies with C.I.S.P.R.22 standard - class B.

**5.12 DIMENSIONS**

**5.12.1 Fixing dimensions**

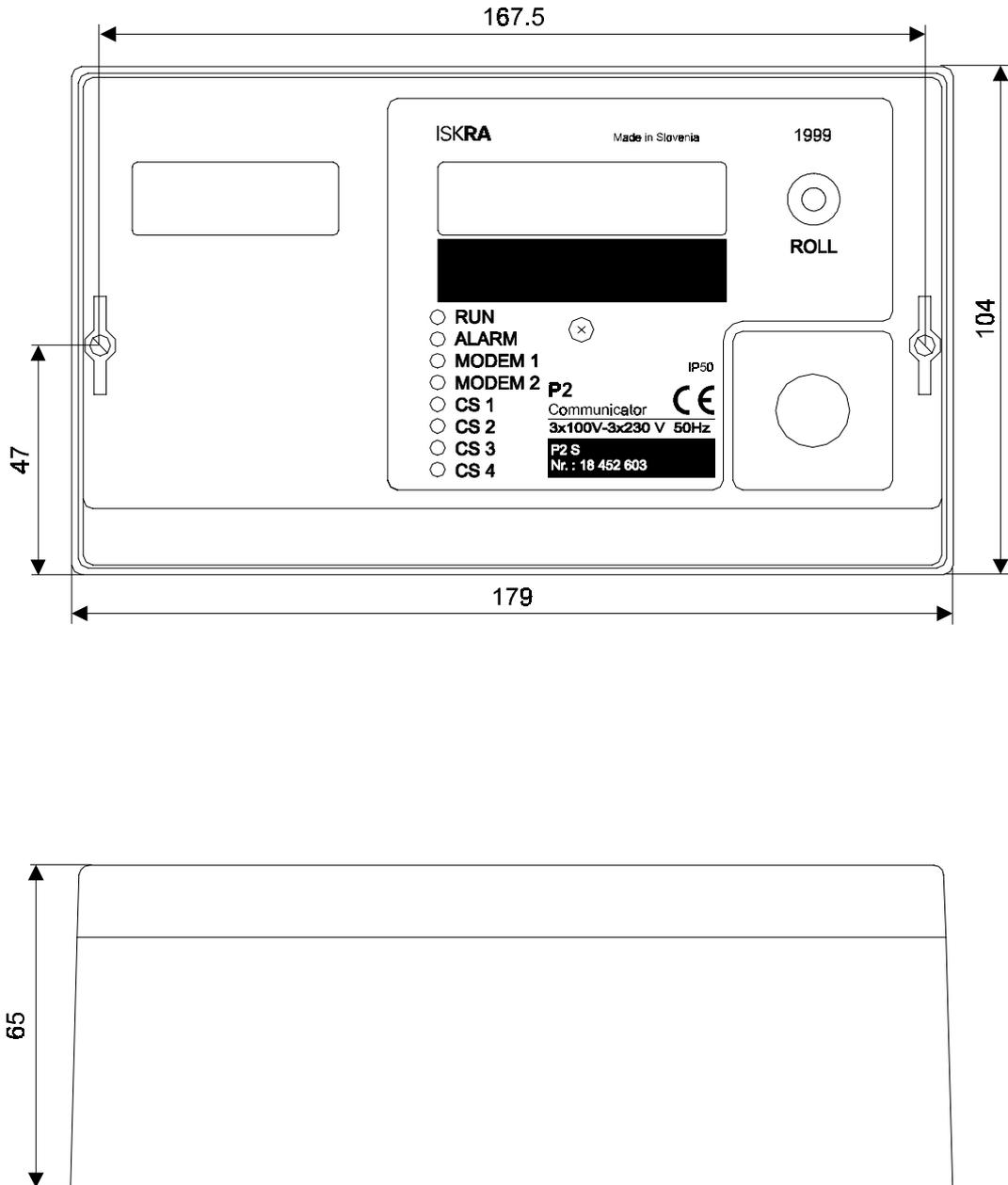


Figure 5-8 Fixing dimensions

**5.12.2 Mass**

Communicator is 1.3 kg heavy.

5.13 CONNECTION DIAGRAM

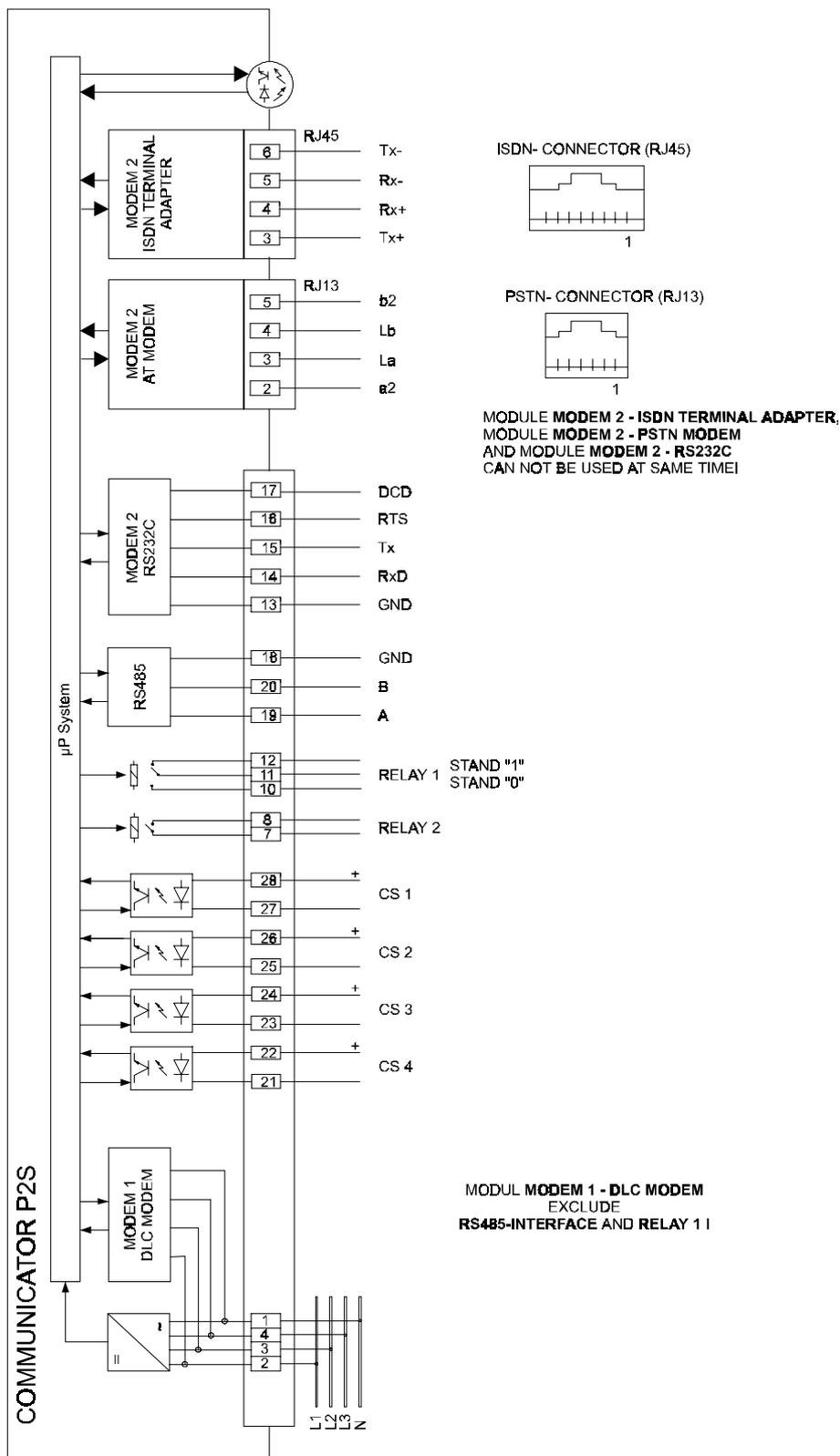


Figure 5-9 Connection diagram

6. TABLES OF REGISTERS

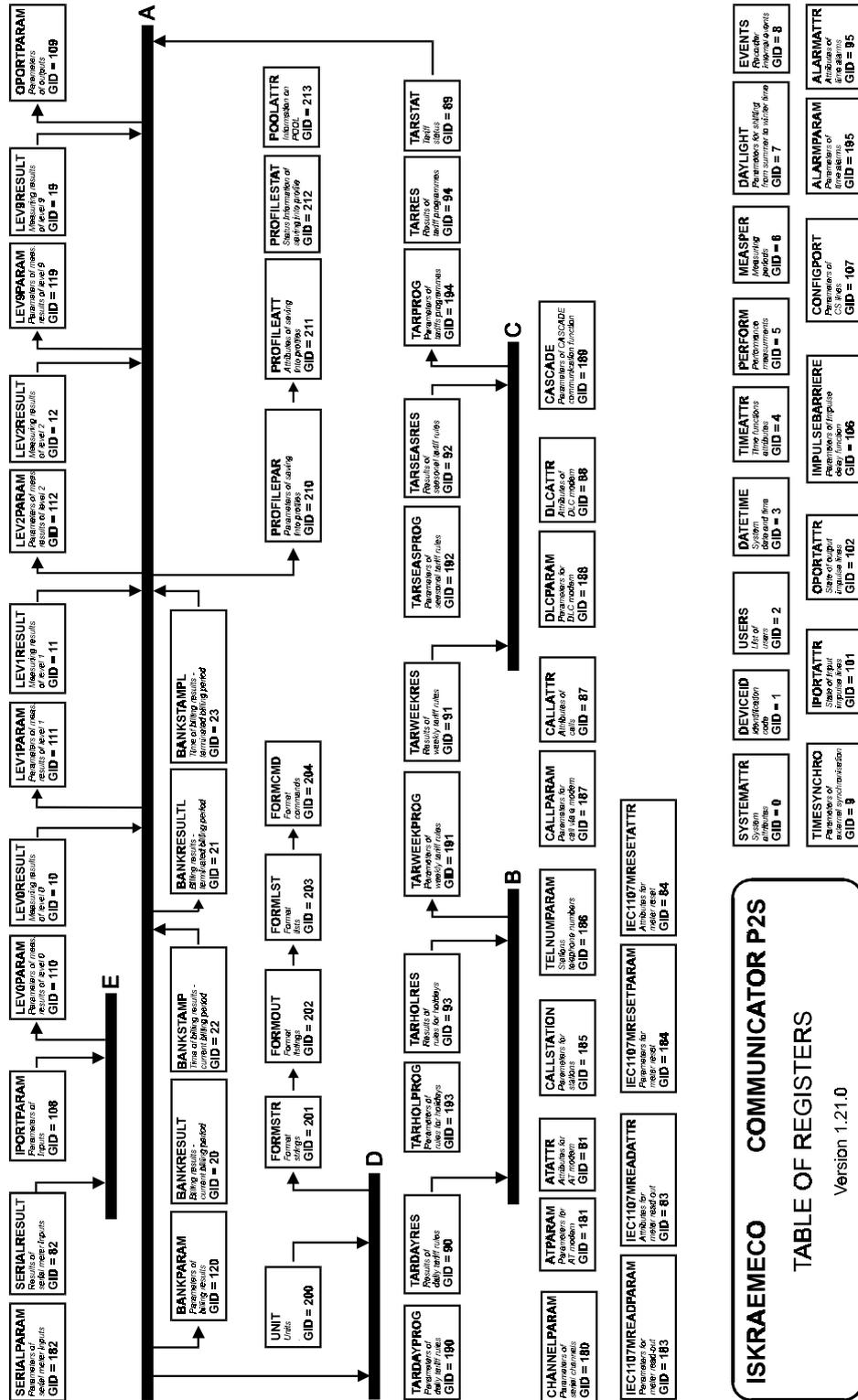


Figure 6-1 Table of registers - All registers

**ISKRAEMECO COMMUNICATOR P2S**  
**TABLE OF REGISTERS**  
 Version 1.21.0

SYSTEMATTR	
<i>System attributes</i>	
<b>GID = 0</b>	
Descript.	Contents
00	00
01	Status
02	Error
03	Alarm
04	Warning
05	PowerFailCount
06	PowerFailStamp
07	PowerFailDuration
08	PowerOnCount
09	PowerOnStamp
10	PowerOnDuration
11	PhaseRFailCount
12	PhaseRFailStamp
13	PhaseRFailDuration
14	PhaseROnCount
15	PhaseROnStamp
16	PhaseROnDuration
17	PhaseSFailCount
18	PhaseSFailStamp
19	PhaseSFailDuration
20	PhaseSOnCount
21	PhaseSOnStamp
22	PhaseSOnDuration
23	PhaseTFailCount
24	PhaseTFailStamp
25	PhaseTFailDuration
26	PhaseTOnCount
27	PhaseTOnStamp
28	PhaseTOnDuration
29	WatchDogCount
30	WatchDogStamp
31	RestartCount
	RestartStamp

DEVICEID	
<i>Identification code</i>	
<b>GID = 1</b>	
Descript.	IDENT
00	00

USERS			
<i>List of users</i>			
<b>GID = 2</b>			
Descript.	Code	Password	Rights
00	00	01	02
01			
02			
03			
04			
05			
06			
07			

DATETIME								
<i>System date and time</i>								
<b>GID = 3</b>								
Descript.	YEAR	MONTH	DAY	WDAY	HOUR	MIN	SEC	D.LIGHT
00	00	01	02	03	04	05	06	07

TIMEATTR			
<i>Time functions attributes</i>			
<b>GID = 4</b>			
Descript.	CONT.	CONT.	CONT.
00	00	01	02
01	TMP1	DMP1	SNMP1
02	TMP2	DMP2	SNMP2
03	TMP3	DMP3	SNMP3
04	TD	DD	SND
05	TM	DM	SNM
06	TY	DY	SNY
	S	DS	SNS

PERFORM		
<i>Performance measurement</i>		
<b>GID = 5</b>		
Descript.	LM	MM
00	00	01
01		
02		
03		
04		
05		
06		
07		
08		
09		
10		

MEASPER			
<i>Measuring periods</i>			
<b>GID = 6</b>			
Descript.	MP1	MP2	MP3
00	00	01	02

DAYLIGHT					
<i>Parameters for shifting from summer to winter time</i>					
<b>GID = 7</b>					
Descript.	YEAR	MONTH	DAY	WDAY	ALG
00	00	01	02	03	04
01					

Figure 6-2 Table of registers - System (Part 1)

EVENTS				
<i>Recorder internal events</i>				
<b>GID = 8</b>				
Descript.	PC	SC	TE	CE
00	01	02	03	
00				
01				
02				
03				
...				
253				
254				

TIMESYNCHRO				
<i>Parameters of external synchronisation</i>				
<b>GID = 9</b>				
Descript.	TYPE	PORT	DP	OS
00	01	02	03	
00				

IPOINTTR	
<i>State of input impulse lines</i>	
<b>GID = 101</b>	
Descript.	IN_0-7
00	00
00	

OPOINTTR	
<i>State of output impulse lines</i>	
<b>GID = 102</b>	
Descript.	OUT_0-7
00	00
00	

IMPULSEBARRIERE		
<i>Parameter of impulse delay function</i>		
<b>GID = 106</b>		
Descript.	BT	AT
00	00	01
00		
01		
02		

CONFIGPORT	
<i>Parameters of CS lines</i>	
<b>GID = 107</b>	
Descript.	Contents
00	00
00	
01	
02	
03	

CS 1  
CS 2  
CS 3  
CS 4

ALARMPARAM									
<i>Parameters of time alarms</i>									
<b>GID = 195</b>									
Descript.	Y	M	D	DW	AD	H	MIN	S	AT
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		
<i>Attributes of time alarms</i>		
<b>GID = 95</b>		
Descript.	TA	NA
00	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
00	00	01	02	03	04	05
01						
02						
03						
04						
05						
06						
07						

OPTICAL  
MODEM 1  
MODEM 2  
RS 485  
CS 1  
CS 2  
CS 3  
CS 4

**CASCADE**  
Parameters of CASCADE communication function  
**GID = 189**

Descript.	STATUS	CHANNEL
00	00	01
01		

**ATPARAM**  
AT modem parameters  
**GID = 181**

Descript.	IS	IP	AT	RTA	ASTRH	ASTRM	ASTPH	ASTPM	PD	RC
00	00	01	02	03	04	05	06	07	08	09
01										
02										
03										
04										
05										
06										
07										

**ATATTR**  
Attributes of modem communication  
**GID = 81**

Descript.	LI	LD	RT	FC	DC
00	00	01	02	03	04
01					
02					
03					
04					
05					
06					
07					

OPTICAL  
MODEM 1  
MODEM 2  
RS 485  
CS 1  
CS 2  
CS 3  
CS 4

**CALLSTATION**  
Parameters of stations  
**GID = 185**

Descript.	ST	SD	SP	CT	CI
00	00	01	02	03	04
01					
02					
03					
04					
05					
06					
07					

**TELNUMPARAM**  
Station telephone numbers  
**GID = 186**

Descript.	TN
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
00	00	01	02	03	04	05	06
01							
02							
03							
04							
05							
06							
07							

**CALLATTR**  
Attributes of calls  
**GID = 87**

Descript.	ST	FL	LS	LC	LCI
00	00	01	02	03	04
01					
02					
03					
04					
05					
06					
07					

**DLCPARAM**  
Parameters for DLC modem  
**GID = 188**

Descript.	TYPE	MASTER ADDR.	RELAY HOPS	RETRY COUNT
00	00	01	02	03
01				
02				
03				
04				
05				
06				
07				

**DLCATTR**  
Attributes of DLC modem  
**GID = 88**

Descript.	MODE
00	00
01	REVISION
02	TYPE
03	ADDRESS
04	MODE
05	RATE
06	TUNE
07	GAIN
08	CONNECT COUNT
09	CONNECT FAIL COUNT
10	RESET COUNT
11	ERROR

Figure 6-4 Table of registers - Communication

<b>IEC1107MREADPARAM</b> Parameters for meter read-out <b>GID = 183</b>							
Descript.	TYPE	CHANN	MA	PASS	CO	CO_A	AL
<del>00</del>	00	01	02	03	04	05	06
00							
01							
02							
03							
04							
05							
06							
07							

<b>IEC1107MREADATTR</b> Attributes of meter read-out <b>GID = 83</b>						
Descript.	RT	RC	MA	B	MO	ID
<del>00</del>	00	01	02	03	04	05
00						
01						
02						
03						
04						
05						
06						
07						

<b>IEC1107MRESETPARAM</b> Parameters for meter reset <b>GID = 184</b>		
Descript.	TYPE	AL
<del>00</del>	00	01
00		

<b>IEC1107MRESETATTR</b> Attributes of meter reset <b>GID = 84</b>		
Descript.	RT	ST
<del>00</del>	00	01
00		

Figure 6-5 Table of registers - IEC 1107

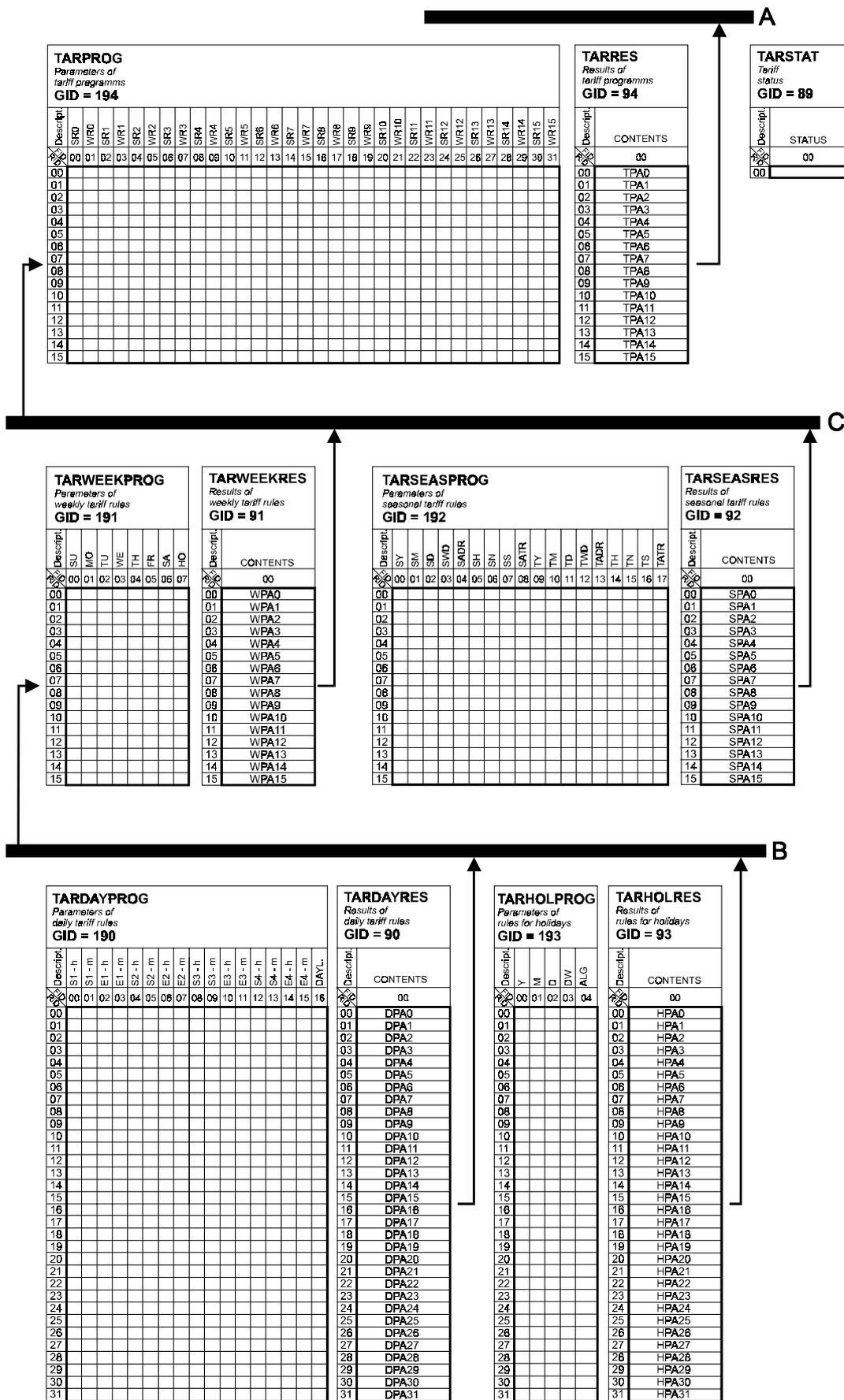


Figure 6-6 Table of registers - Tariff program



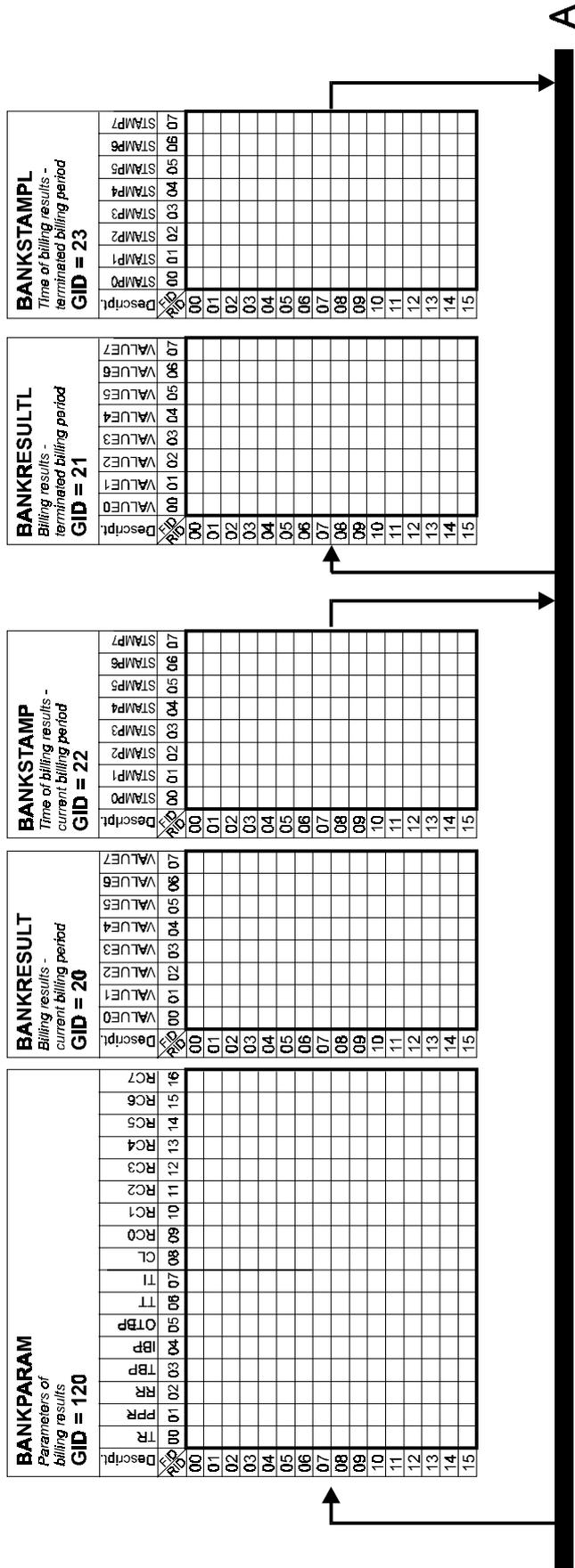
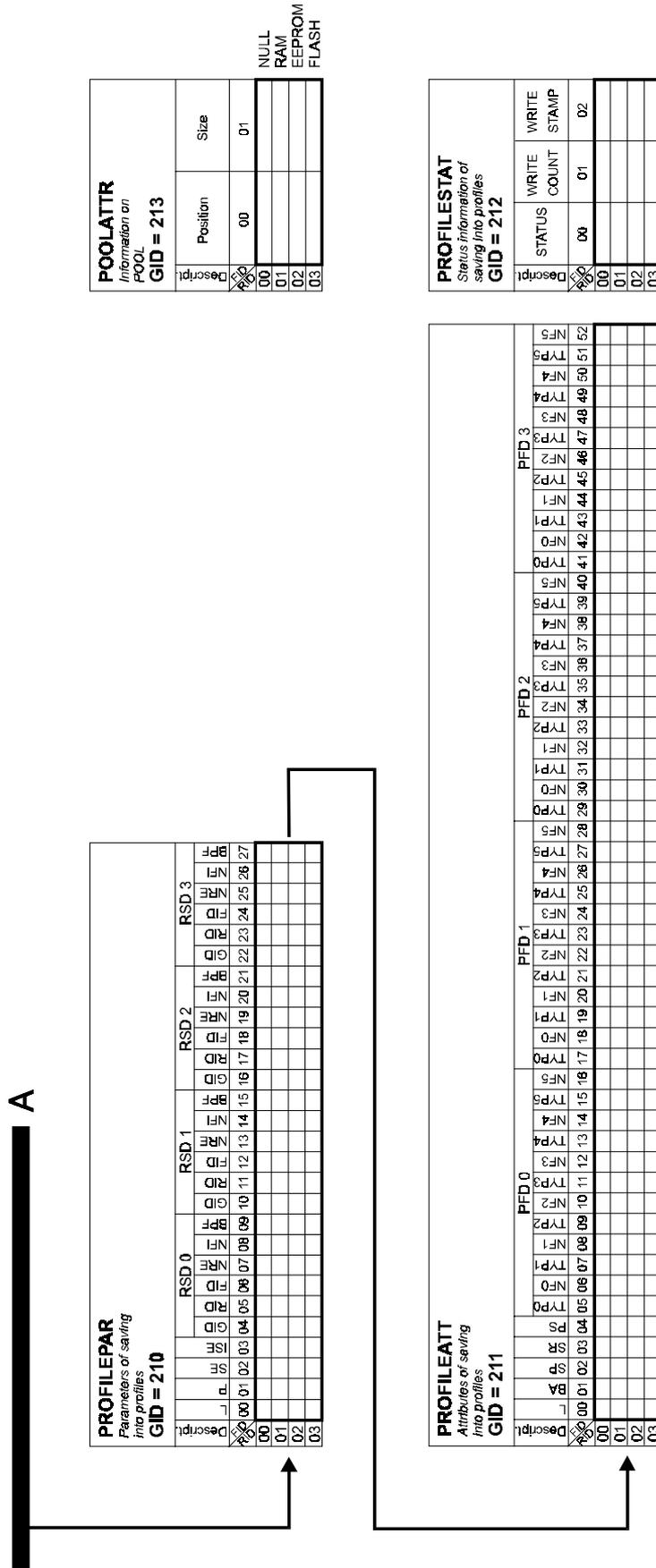


Figure 6-8 Table of registers - Billing results



**PROFILEPAR**  
Parameters of saving  
into profiles  
**GID = 210**

Descriptor	RSD 0			RSD 1			RSD 2			RSD 3																		
	TY	NF	PS																									
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	
01																												
02																												
03																												

**POOLATTR**  
Information on  
POOL  
**GID = 213**

Descriptor	Position	Size
00	00	01
01		
02		
03		

NULL  
RAM  
EEPROM  
FLASH

**PROFILEATT**  
Attributes of saving  
into profiles  
**GID = 211**

Descriptor	PFD 0			PFD 1			PFD 2			PFD 3																																											
	TY	NF	PS																																																		
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	
01																																																					
02																																																					
03																																																					

**PROFILESTAT**  
Status information of  
saving into profiles  
**GID = 212**

Descriptor	STATUS	WRITE COUNT	WRITE STAMP
00	00	01	02
01			
02			
03			

Figure 6-9 Table of registers - Saving

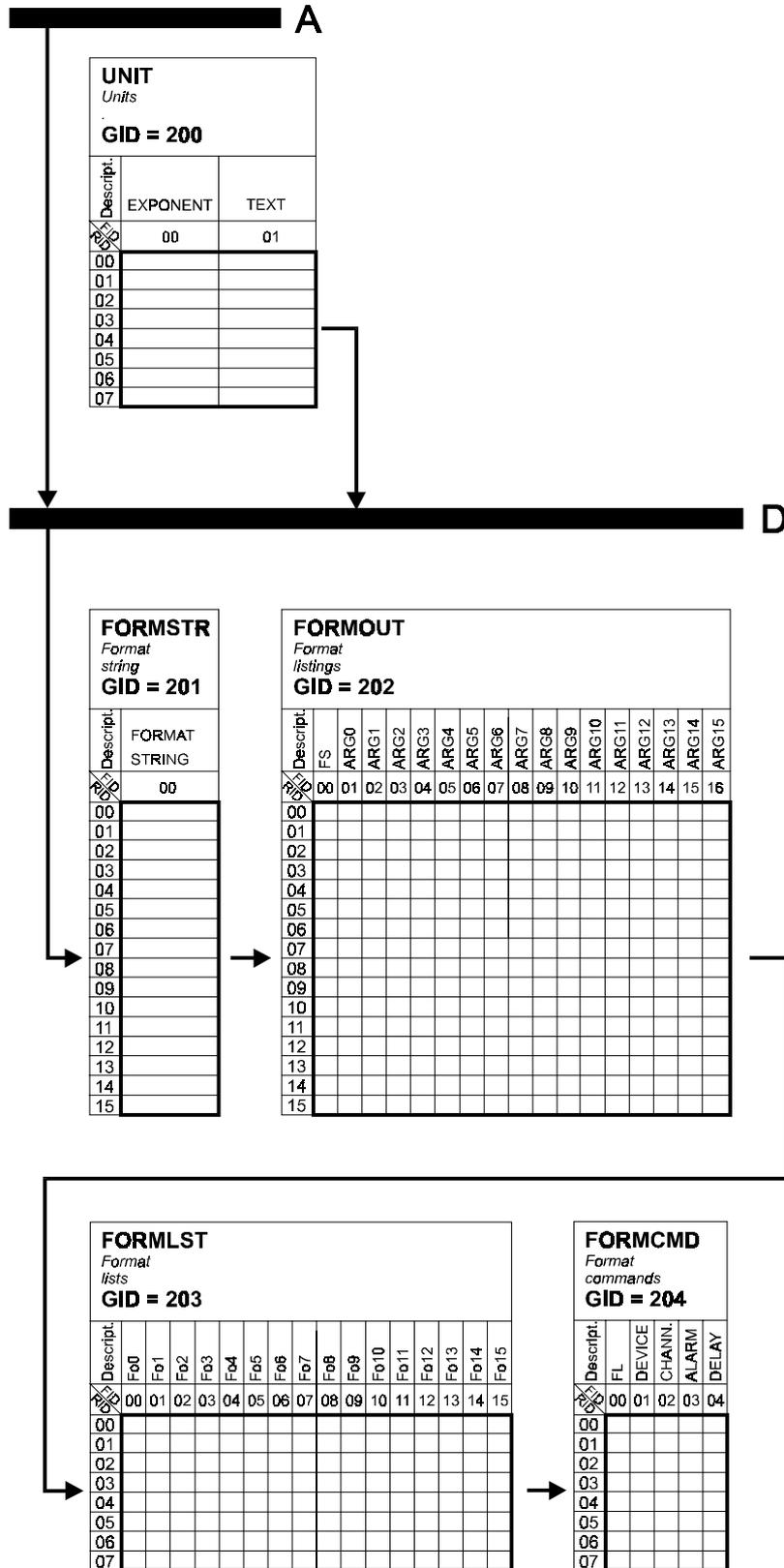


Figure 6-10 Table of registers - Format listings

## 7. INDEX

### Figure:

Figure 2-1 Communicator design.....	8
Figure 2-2 Position of modems.....	8
Figure 2-3 CPU.....	9
Figure 2-4 CPU - Settings.....	9
Figure 2-5 Power supply.....	11
Figure 2-6 DLC modem.....	12
Figure 2-7 PSTN modem.....	12
Figure 2-8 ISDN terminal adapter.....	13
Figure 3-1 Block diagram of program functions.....	14
Figure 3-2 Function of pulse retaining.....	16
Figure 3-3 Point of function operations.....	16
Figure 3-4 SDT net.....	17
Figure 3-5 Transmission of measuring results.....	22
Figure 3-6 Measuring results - from levels 0 to 2.....	24
Figure 3-7 Measuring results - level 9.....	24
Figure 3-8 Hysteresis.....	29
Figure 3-9 Billing results.....	31
Figure 3-10 Values in banks.....	31
Figure 3-11 Tariff program.....	33
Figure 3-12 Group of registers.....	41
Figure 3-13 Saving structure.....	53
Figure 3-14 Block of registers.....	54
Figure 3-15 IEC 1107 Master.....	76
Figure 3-16 IEC 1107 Meter Readout.....	77
Figure 3-17 IEC 1107 Command execution.....	78
Figure 3-18 Cascade connection - network.....	79
Figure 3-19 Cascade connection - connecting method.....	80
Figure 3-20 Modem communication.....	81
Figure 3-21 Status of modem channel.....	81
Figure 3-22 The DLC network.....	83
Figure 3-23 Console.....	84
Figure 3-24: ROLL key - »Autoscroll« and »Manualscroll« display.....	86
Figure 3-25: ROLL key - »Register« display.....	87
Figure 3-26 ROLL Key - Display "Profile Register".....	88
Figure 5-1 Time diagram - Two-wire input for active current.....	184
Figure 5-2 Connection scheme for a typical two-wire input for active current.....	184
Figure 5-3 Time diagram - Three-wire input.....	185
Figure 5-4 Connection scheme for a typical three-wire input.....	185
Figure 5-5 Connection scheme - Power supply.....	187
Figure 5-6 Connection scheme - RS485.....	189
Figure 5-7 Connection scheme - RS232.....	189
Figure 5-8 Fixing dimensions.....	195
Figure 5-9 Connection diagram.....	196
Figure 6-1 Table of registers - All registers.....	197
Figure 6-2 Table of registers - System (Part 1).....	198
Figure 6-3 Table of registers - System (Part 2).....	199
Figure 6-4 Table of registers - Communication.....	200
Figure 6-5 Table of registers - IEC 1107.....	201
Figure 6-6 Table of registers - Tariff program.....	202
Figure 6-7 Table of registers - Measuring results.....	203
Figure 6-8 Table of registers - Billing results.....	204
Figure 6-9 Table of registers - Saving.....	205
Figure 6-10 Table of registers - Format listings.....	206

**Table:**

Table 3-1 Conversion of impulses - calculating flow.....	26
Table 3-2 Weekly rules - example .....	36
Table 3-3 Information on saving - STATUS.....	54
Table 3-4 Display: Input ports .....	89
Table 3-5 Display: Output ports .....	89
Table 3-6 Channel status - Messages .....	90
Table 4-1 Groups of registers .....	99
Table 4-2 Group of registers SYSTEMATTR.....	100
Table 4-3 Group of registers DEVICEID .....	103
Table 4-4 Group of registers USERS.....	104
Table 4-5 Group of registers DATETIME.....	105
Table 4-6 Group of registers TIMEATTR.....	106
Table 4-7 Group of registers PERFORM.....	108
Table 4-8 Group of registers MEASPER .....	109
Table 4-9 Group of registers DAYLIGHT CHANGE PARAMETERS.....	110
Table 4-10 Group of registers EVENTS.....	111
Table 4-11 Group of registers TIMESYNCHRO .....	114
Table 4-12 Group of registers LEV0RESULT .....	115
Table 4-13 Group of registers LEV1RESULT .....	116
Table 4-14 Group of registers LEV2RESULT .....	117
Table 4-15 Group of registers LEV9RESULT .....	118
Table 4-16 Group of registers BANKRESULT .....	119
Table 4-17 Group of registers BANKRESULTL.....	120
Table 4-18 Group of registers BANKSTAMP.....	121
Table 4-19 Group of registers BANKSTAMPL.....	122
Table 4-20 Group of registers ATATTR.....	123
Table 4-21 Group of registers SERIALRESULT .....	124
Table 4-22 Group of registers IEC1107MREADATTR .....	126
Table 4-23 Group of registers IEC1107MRESETATTR .....	127
Table 4-24 Group of registers CALLATTR .....	128
Table 4-25 Group of registers DLCATTR .....	129
Table 4-26 Group of registers TARSTAT .....	131
Table 4-27 Group of registers TARDAYRES .....	132
Table 4-28 Group of registers TARWEEKRES .....	133
Table 4-29 Group of registers TARSEASRES.....	134
Table 4-30 Group of registers TARHOLRES.....	135
Table 4-31 Group of registers TARRES .....	136
Table 4-32 Group of registers ALARMATTR .....	137
Table 4-33 Group of registers IPORTATTR .....	138
Table 4-34 Group of registers OPORTATTR.....	139
Table 4-35 Group of registers IMPULSEBARRIERE.....	140
Table 4-36 Group of registers CONFIGPORT .....	141
Table 4-37 Group of registers IPORTPARAM .....	142
Table 4-38 Group of registers OPORTPARAM .....	143
Table 4-39 Group of registers LEV0PARAM .....	144
Table 4-40 Group of registers LEV1PARAM .....	146
Table 4-41 Group of registers LEV2PARAM .....	148
Table 4-42 Group of registers LEV9PARAM .....	150
Table 4-43 Group of registers BANKPARAM .....	151
Table 4-44 Group of registers CHANNELPARAM.....	153
Table 4-45 Group of registers ATPARAM.....	155
Table 4-46 Group of registers SERIALPARAM .....	156
Table 4-47 Group of registers IEC1107MREADPARAM .....	158
Table 4-48 Group of registers IEC1107MRESETPARAM .....	160
Table 4-49 Group of registers CALLSTATION .....	161
Table 4-50 Group of registers TELNUMPARAM .....	162
Table 4-51 Group of registers CALLPARAM .....	163
Table 4-52 Group of registers DLCPARAM.....	165
Table 4-53 Group of registers CASCADE.....	166

Table 4-54 Group of registers TARDAYPROG.....	167
Table 4-55 Group of registers TARWEEKPROG.....	168
Table 4-56 Group of registers TARSEASPROG.....	169
Table 4-57 Group of registers TARHOLPROG.....	171
Table 4-58 Group of registers TARPROG.....	172
Table 4-59 Group of registers ALARMPARAM.....	173
Table 4-60 Group of registers UNIT.....	175
Table 4-61 Group of registers FORMSTR.....	176
Table 4-62 Group of registers FORMOUT.....	177
Table 4-63 Group of registers FORMLST.....	178
Table 4-64 Group of registers FORMCMD.....	179
Table 4-65 Group of registers PROFILEPAR.....	180
Table 4-66 Group of registers PROFILEATT.....	181
Table 4-67 Group of registers PROFILESTAT.....	182
Table 4-68 Group of registers POOLATTR.....	183
Table 5-1 Outputs - Type and characteristics of switching.....	186

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 4) 206 40 00, Telefax: (+386 4) 206 43 76  
Published by Iskraemeco, Marketing  
Data subject to alternation without notice.