KMB systems, s.r.o.
Dr. M. Horákové 559, 460 06 Liberec 7
tel. +420 485 130 314, fax +420 482 739 957
email: kmb@kmb.cz, internet: www.kmb.cz

Description of SMY 33 / SMZ 33 Communication Protocols.

Programmer's Manual

(valid from firmware version 7.3 up)

11/2008

1. Data Transmission Between SMY 33 / SMZ 33 and Host System

SMY 33 and SMZ 33 instruments have the optional RS-485 or RS-232 interface feature. They can be monitored and controlled from a host system (usually a PC) using such a communication link. This manual provides description of the methods of communication with these instruments. The reader's elementary knowledge of instrument parameters and the C programming language is assumed.

Data that can be transmitted between the instrument and host system are arranged using the following structure:

1. "Status" contains information about instrument status (serial number, error flags, etc.) to be read only
2. "ID" contains information about instrument model (serial number, model, etc) to be read only
3. „RTC" real time counter to be read or written
4. "Config" contains instrument parameter settings (VT or CT conversion, etc.) to be read or written
5. "OutputConfig" contains output relays and pulse outputs parameter settings to be read or written
6. "ElMeter" contains electricity meter state be read and tariff settings to be read or written
7. "Profiles" contains daily profiles state and settings to be read or written
8. "ActAllData" allows to read currently measured data from the instrument

The host system receives instrument information and measurement data by reading the relevant structure component from the instrument or it can for example change a parameter, start a requested operation , etc., by writing into a relevant structure component.

The contents of each structure component are described in separate chapters further below.

2. Communication Protocols

Data transmission between the instrument and host system takes place via an asynchronous serial communication line (COM port) using the RS-485 or the RS-232 interface. In case of the RS-232, a number of instruments can be connected to one communication line and an RS-232/RS-485 converter with an automatic data flow direction switch or host system software controlled data flow direction switching can be used at the host system side.

The instruments' standard feature is an incorporated KMB in-house protocol. The baud rate can be set from 300 to 19,200 Bd (default setting at 9,600 Bd).

2.1 KMB Communication Protocol

The communication channel uses the setting of 8 data bits, no parity, and one stop bit. The address and data flow rate can be set. The communication protocol employs the master-slave philosophy. In response to receiving a proper message or command the instrument sends back a relevant reply.

The messages all have a uniform format:

1. instrument address (1 byte)
2. message length (in bytes) less the checksum at the end; it is the message body length + 3 (1 byte)
3. type of message (1 byte)
4. message body – varies in accordance with type of message
5. checksum – sum of all other bytes in the message, modulo 256 (1 byte)

When the instrument receives a command, it sends back a relevant reply where the type-of-message byte contains zero. If the type-of-message byte has a non-zero value, the command could not be carried out for a reason.

Some messages do not have a body.

The instrument sends back a reply within 600 ms from receiving a command. A time gap between bytes corresponding to the time of transmitting maximum 2 characters (2 bytes) is allowed while receiving or transmitting a command.

2.1.1 Message Description

Messages corresponding with the following chart can be used to read or write data.

message # (hex)	type of message
0x01	instrument identification
0x10	set RTC state
0x11	read RTC state
0x14	read instrument status - Status
0x26	read instrument setting – Config
0x27	write instrument setting – Config
0x30	read instrument outputs setting – OutputConfig
0x31	write instrument outputs setting – OutputConfig
0x32	read elmeter tariff setting
0x33	write elmeter tariff setting
0x34	read elmeter state
0x35	clear elmeter
0x36	read daily profiles state and actual profile data - ProfilStatus
0x37	write S-profile date
0x38	clear M-profile
0x39	read profiles date&time
0x3A	read all currently measured data – ActAlldata

The following chapters look primarily at these messages in more detail.

2.1.2 Instrument Identification (0x01)

The instrument identifies itself on request.

Command:

| 0x01 | 0x03 | 0x01 | 0x05 |

Reply:

| 0x01 | 0x11 | 0x00 | identification | checksum |

identification (broken down byte by byte, altogether 14 bytes):
DeviceNo % 256
DeviceNo / 256
DeviceType % 256
DeviceType / 256
PropsType % 256
PropsType / 256
Firmware version
Reserved
RemoteAddress
Reserved
Reserved
Reserved
Reserved
Reserved

2.1.3 Setting RTC 0x10

Master writes actual date&time into the instrument.

Command:

For example, actual date&time is 2003.8.15 10:29:00. Master sends following string (address 1 supposed) :

| 0x01| 0x09 | 0x10 | 0x03|0x08|0x15|0x10|0x29|0x00|checksum|

Reply :

| 0x01| 0x03| 0x00 | checksum |

2.1.4 Reading RTC 0x11

Master asks for instrument real time counter state.

Command:

| 0x01| 0x03| 0x11 | checksum |

Reply :

| 0x01| 0x09 | 0x00 | 0x03|0x08|0x15|0x10|0x29|0x00|checksum|

2.1.5 Reading instrument state (Status) – 0x14

The instrument provides information on its state – Status (52 bytes).

Command:

| 0x01| 0x03 | 0x14 | checksum = 0x18 |

Reply:

| 0x01| 0x37| 0x00 | ..... message body = Status (52 bytes) ..... | checksum |

2.1.6 Reading Instrument Setting – Config (0x26)

The instrument provides information on its setting – Config (28 bytes).

Command: | 0x01 | 0x03 | 0x26 | 0x2A |

Reply: | 0x01 | 0x1F | 0x00 | Config(28 bytes) | checksum |

2.1.7 Writing Instrument Setting – Config (0x27)

You can change the instrument setting by writing the Config information into it.

Command: | 0x01 | 0x1F | 0x27 | Config(28 bytes) | checksum |

Reply: | 0x01 | 0x03 | 0x00 | 0x04 |

Note: The DeviceAddr and RemoteBdRate variables may not be altered via the communication line, so you can write any values in them.

2.1.8 Reading Instrument Outputs Setting (OutputConfig) – 0x30

The instrument provides information on its setting – OutputConfig (20 bytes).

Command:

| 0x01| 0x03 | 0x30 | checksum = 0x34 |

Reply:

| 0x01| 0x17| 0x00 | ..... message body = OutputConfig (20 bytes) ..... | checksum |

2.1.9 Writing Instrument Outputs Setting (OutputConfig) – 0x31

You can change the instrument outputs behavior setting by writing the OutputConfig (20 bytes) information into it.

Command:

| 0x01| 0x17 | 0x31 | ..... message body = OuputConfig (20 bytes) ..... | checksum |

Reply:

| 0x01| 0x03| 0x00 | checksum = 0x04 |

2.1.10 Reading Instrument ElMeter Tariff Setting (ElmerTarif) – 0x32

Instrument replies ElmerTarif (6 bytes).

Command:

| 0x01| 0x03 | 0x32 | checksum = 0x36 |

Reply:

| 0x01| 0x09| 0x00 | ..... message body = ElmerTarif (6 bytes) ..... | checksum |

2.1.11 Writing Instrument ElMeter Tariff Setting (ElmerTarif) – 0x33

Master writes ElmerTarif (6 bytes) into the instrument.

Command:

| 0x01| 0x09| 0x33 | ..... message body = ElmerTarif (6 bytes) ..... | checksum |

Reply:

| 0x01| 0x03| 0x00 | checksum = 0x04 |

2.1.12 Reading Instrument ElMeter State (ElmerActData) – 0x34

Instrument replies ElmerActData (94 bytes).

Command:

| 0x01| 0x03 | 0x34 | checksum = 0x38 |

Reply:

| 0x01| 0x61| 0x00 | ..... message body = ElmerActData (94 bytes) ..... | checksum |

2.1.13 Clearing Instrument ElMeter – 0x35

Command:

| 0x01| 0x04 | 0x35 | 0x01| checksum = 0x3B |

Reply:

| 0x01| 0x03| 0x00 | checksum = 0x04 |

2.1.14 Reading All Currently Measured Data from Instrument – ActAlldata (0x3A)

The instrument sends the measurement data – ActAlldata (218 bytes).

Command: | 0x01 | 0x03 | 0x3A | 0x3E |

Reply: | 0x01| 0x50| 0x00 | ActAlldata (218 bytes) | checksum|

2.2 Modbus-RTU Communication Protocol

SMY 33 / SMZ 33 instruments can be set to use the Modbus-RTU protocol where the address, baud rate and parity bit (even / odd / no parity) are specified.

The instrument sends back a reply within 600 ms from receiving a command. A gap between bytes corresponding to maximum 1.5 characters (bytes) is allowed while receiving a command or transmitting a reply.

The "broadcast" mode is not supported.

The following functions have been implemented:

function code	function description	application
03	Read Holding Registers	reading instrument setting – Status, Config, OutputConfig, ElmerTarif
04	Read Input Registers	reading currently measured data from instrument – ActAllData, ElmerActData
06	Preset Single Register	writing Config, OutputConfig, ElmerTarif, Elmer clearing
08	Diagnostics – 00 – Return Query Data 01 – Restart Comm Option 10 – Clear Ctrs & Diag. Register 11 – Return Bus Message Count	basic diagnostic functions
16	Preset Multiple Registers	similar to 06 - Preset Single Register

Access to data structure components is provided using read/write from/to relevant registers as shown in the chart in the following chapters. Single-byte data are stored in a register in the format of 0x00nn where nn is a single-byte parameter. The data structure components that hold the instrument status information are stored in an array of 'holding' registers. The currently measured data can be read as the contents of the 'input' registers. Each structure component is stored within the array of registers using the base addresses shown in the following table (for the 'holding' registers). The currently measured data are stored in the array of 'input' registers at the base address of 0.

structure component	base address
IDENTIFICATION	0x0200
RTC	0x0300
STATUS	0x0400
CONFIG	0x0700
OUTPUTCONFIG	0x0800
ELMERTARIF	0x0B00
CLRELMER	0x0B80

2.2.1 Status

The base address is 0x0400. Detailed description in *Data Structures* chapter.

Offset	Parameter	Read(R)/Write(W)	Length [bytes]
0x0000	RamErr	R/W	1
0x0001	WDHafs	R	1
0x0002	PWOffs	R	1
0x0003	RecordMode	R/W	1
0x0004	PwrConf	R/W	2
0x0005	Iconf[0]	R/W	1
0x0006	Iconf[1]	R/W	1
0x0007	Iconf[2]	R/W	1

0x0008	Iconf[3]	R/W	1
0x0009	Uconf	R/W	1
0x000A	FrConf	R/W	1
0x000B	Tconf	R/W	1
0x000C	AuxConf	R/W	1
0x000D	HDOTrigLevel	R/W	1
0x000E	Reserve	R/W	1
0x000F – 0x0010	MaxRecs	R	4
0x0011 – 0x0012	NoRecs	R	4
0x0013 – 0x0014	CurrRecNo	R	4
0x0015 – 0x001A	LastRecTime	R	6
0x001B – 0x0020	StartRecTime	R	6
0x0021	RecInt	R/W	1
0x0022	DeviceNo	R	1
0x0023	NoteBookSize	R	1
0x0024	DeviceType	R	1
0x0025	SoftVersion	R	1
0x0026	Reserve6	R/W	1

2.2.2 Config

The base address is 0x0700. All registers are the Read/Write kind.

Offset	Parameter	Length in Bytes
0x0000 + 0x0001	VT	4
0x0002 + 0x0003	CT	4
0x0004	NomPwr	2
0x0005	InputType	1
0x0006	DeviceAddr	1
0x0007	RemoteBdRate	1
0x0008	Reserved	4
0x000A	CANDeviceAddr	2
0x000B	NomU	2
0x000C + 0x000D	Reserved	4
0x000E	Temp4mA	2
0x000F	Temp20mA	2

For a more detailed specification and coding of each parameter, see “Config” data structure description.

2.2.3 OutputConfig

The base address is 0x0800. All registers are the Read/Write kind.

Offset	Parameter	Length in Bytes
0x0000	PulseOut	1
0x0001 + 0x0002	rkWh	4
0x0003 + 0x0004	Limit	4
0x0005 + 0x0006	Hysteresis	4
0x0007	DelayTime	2
0x0008	ReleCgf	1
0x0009 + 0x000A	Reserved	4

2.2.4 RTC

The base address is 0x0300. All registers are the Read/Write kind.

Offset	Parametrr	Length in Bytes
0x0000	Year & Month	2
0x0001	Day & Hour	2
0x0002	Minute & Second	2

2.2.5 Identification

The base address is 0x0200. All the registers are Read Only.

Address Offset	Parameter	Length in Bytes
0x0000	DeviceNo (serial number)	2
0x0001	DeviceType (model)	2
0x0002	Props Type (0x0030 value)	2
0x0003	Firmware Version	1
0x0004	Remote Address	1

The DeviceType parameter follows.

SMY Instrument Model	Value
SMY33 – basic model	0900
SMY33T – with temperature sensor input	0901
SMY33R – with output relays	0902
SMY33RT – with temp. sensor input & relays	0903

In case of remote communication interface, the SMY DeviceType parameter part 09XX is modified as follows :

SMY Instrument with remote comm. link	Value
SMY33xx CAN	0Bxx
SMY33xx 485	0Dxx
SMY33xx COM	0Fxx

Example: SMY33RT/ 485 ..... DeviceType = 0D03.

SMZ Instrument Model	Value
SMZ33 - basic model	1100
SMZ33T – – with temperature sensor input	1101
SMZ33R – with output relays	1102
SMZ33E – with electricity meter	1104
SMZ33ERT – with elmeter & relays & temp.	1107

In case of remote communication interface, the SMZ DeviceType parameter part 11XX is modified as follows :

SMZ Instrument with remote comm. link	Value
SMZ33xx CAN	13xx
SMZ33xx 485	15xx
SMZ33xx COM	17xx

2.2.6 ElmerTarif

The base address is 0x0B00. All registers are the Read/Write kind.

Offset	Parametrr	Length in Bytes
0x0000 + 0x0005	ElmerTarif[0] + ElmerTarif[5]	6

2.2.7 ClrElmer

The base address is 0x0B80. All registers are the Read-Only kind.

Offset	Parametrr	Length in Bytes
0x0000	by writing 0x0100-constant the elmeter is cleared	1

2.2.8 Currently Measured Data

The base address is 0x0000. The currently measured data can be read as the value of the 'input' registers ('Read Input Registers' function).

Register Address	Parameter	Length in Bytes
0x0000	U1	2
0x0001	U2	2
0x0002	U3	2
0x0003	LU	1
0x0004	I1	2
0x0005	I2	2
0x0006	I3	2
0x0007	I4	2
0x0008	Cos fi 1	1
0x0009	Cos fi 2	1
0x000A	Cos fi 3	1
0x000B	Fr + Contacts	1
0x000C	Temperature	1
0x000D	PF1	1
0x000E	PF2	1
0x000F	PF3	1
0x0010	U12	2
0x0011	U23	2
0x0012	U31	2
0x0100+ 0x0101	P1	4
0x0102 + 0x0103	P2	4
0x01040+ 0x0105	P3	4
0x0106 + 0x0107	Q1	4
0x0108 + 0x0109	Q2	4
0x010A + 0x010B	Q3	4
0x010C + 0x010D	S1	4
0x010E + 0x010F	S2	4
0x0110 + 0x0111	S3	4

0x0200	THDU1	1
0x0201	THDU2	1
0x0202	THDU3	1
0x0210	HarU phase 1, order 2	1
0x0211	HarU phase 1, order 3	1
...	...	...
0x0227	HarU phase 1, order 25	1
0x0228 + 0x022F	reserved	16
0x0230	HarU phase 2, order 2	1
0x0231	HarU phase 2, order 3	1
...	...	...
0x0247	HarU phase 2, order 25	1
0x0248 + 0x024F	reserved	16
0x0250	HarU phase 3, order 2	1
0x0251	HarU phase 3, order 3	1
...	...	...
0x0267	HarU phase 3, order 25	1
0x0268 + 0x026F	reserved	16
0x0300	THDI1	1
0x0301	THDI2	1
0x0302	THDI3	1
0x0310	HarI phase 1, order 2	1
0x0311	HarI phase 1, order 3	1
...	...	...
0x0327	HarI phase 1, order 25	1
0x0328 + 0x032F	reserved	16
0x0330	HarI phase 2, order 2	1
0x0331	HarI phase 2, order 3	1
...	...	...
0x0347	HarI phase 2, order 25	1
0x0348 + 0x034F	reserved	16
0x0350	HarI phase 3, order 2	1
0x0351	HarI phase 3, order 3	1
...	...	...
0x0367	HarI phase 3, order 25	1
0x0368 + 0x036F	reserved	16

For a more detailed specification and coding of each parameter, see “ActAllData” data structure description.

2.2.9 Elmeter Status

The electricity meter status can be read as the value of the ‘input’ registers (‘Read Input Registers’ function).

The base address is 0x0400.

Register Address	Parameter	Length in Bytes
0x0400+ 0x0417	ElmerEnergy	48
0x0418+ 0x041A	ElmerClrTime	6
0x041B+ 0x0422	QHMaxPwr	16
0x0423+ 0x042E	QHMaxPwrTime	24

For a more detailed specification and coding of each parameter, see “ElmerActData” data structure description.

3. Data Structure

The description convention is in the C language style. Multibyte variables are stored in the high to low order (highest byte first, lowest byte last). Because the structure components contain all the parameters required for the instrument's operation not only in the online mode, there are comments only with the parameters that are directly related to the online mode.

Status :

```
typedef  struct {
        uchar RamErr;
                        //*   RamErr structure
                        //*   ----------------------
                        //*   D7...RAM backup error
                        //*   D6...RTC backup error
                        //*   D5...EPROM checksum error
                        //*   D4...
                        //*   D3...
                        //*   D2...spare Config error
                        //*   D1...calibration error
                        //*   D0...EEPROM checksum error

        uchar WDHafs;
        uchar PWOffs;
        uchar RecordMode;
        uint  PwrConf;

        uchar IConf[4];
        uchar UConf;
        uchar FrConf;
        uchar TConf;
        uchar AuxConf;
        uchar HDOTrigLevel;

        uchar Reserve5;

        ulong MaxRecs;
        ulong NoRecs;
        ulong CurrRecNo;

        uchar LastRecTime[6];
        uchar StartRecTime[6];

        uchar RecInt[2];
        uint  DeviceNo;
        uchar NoteBookSize;
        uchar DeviceType;
        uchar SoftVersion;
        uchar Reserve6;
        ulong     THDConf;
      } SType;      /*Statustype*/
//=============================================================================
```

Config :

```
typedef  struct {
        ulong Mtn;                  /* VT nominal primary voltage                  */
```

```
        ulong Mtp;              /* current transformer ratio, coding:                  */
                                /*      bits 30-0: CT nominal primary current          */
                                /*      bit 31:   0...CT nominal secondary current 1A*/
                                /*                1... CT nominal secondary current 5A*/
        uint  NomPwr;           /* nominal power [kVA]                                 */
        uchar InputType;        /* connection&communication configuration              */
                        /* b0.....RemoteComType : 0 = KMB protocol                     */
                        /*                        1 = MODBUS RTU protocol              */
                        /* b1.....Modem :         0 = not connected                    */
                        /*                        1 = connected                        */
                        /* b3,2  MODBUS parity:       00      - none                   */
                        /*                            01      - even                   */
                        /*                            10      - odd                    */
                        /* b4..... 0 – Star or Delta                                   */
                        /*         1 - Aron                                            */
                        /* b5..... 0 - Star                                            */
                        /*         1 – Delta                                           */
                        /* b6..... 0 – U via VT                                        */
                        /*         1 – U directly                                      */
                        /* b7..... 1                                                   */

        uchar   DeviceAddr;     /* address                                             */
        uchar   RemoteBdRate;   /* low nibble: communication rate:                     */
                                /* for COM:                                            */
                                /*      ; 0...50    Bd -                               */
                                /*      ; 1...150   Bd -                               */
                                /*      ; 2...300   Bd -                               */
                                /*      ; 3...600   Bd -                               */
                                /*      ; 4...1200  Bd -                               */
                                /*      ; 5...2400  Bd -                               */
                                /*      ; 6...4800  Bd -                               */
                                /*      ; 7...9600  Bd -                               */
                                /*      ; 8...19200 Bd -                               */
                                /*      ; 9...38400 Bd - atd.                          */
                                /* for CAN:                                            */
                                /*      ; 0...5 kbit/s                                 */
                                /*      ; 1...10 kbit/s                                */
                                /*      ; 2...20 kbit/s                                */
                                /*      ; 3...50 kbit/s                                */
                                /*      ; 4...100 kbit/s                               */
                                /*      ; 5...125 kbit/s                               */
                                /*      ; 6...200 kbit/s                               */
                                /*      ; 7...250 kbit/s                               */
                                /*      ; 8...500 kbit/s                               */
                                /*      ; 9...800 kbit/s                               */
                                /*      ; 10...1000 kbit/s                             */

        ulong   Res0;           /* reserved                                            */
        uint    CANDeviceAddr;      /* address (2-1023) for CAN                        */
        uint    NomU;               /* nominal voltage                                 */
                                    /* at meas. via VT secondary nominal voltage */
        uchar   Res3;           /* reserved                                            */
        uchar   Res4;           /* reserved                                            */
        uchar   Res5;           /* reserved                                            */
        int Temp4mA;            /* temperature at 4 mA [grad C]                        */
        int Temp20mA;           /* temperature at 20 mA [grad C]                       */
} CType;        /*Configuration, 28 bytes*/
// ====================================================================
```

OutputConfig :

```
typedef  struct {
        uchar PulseOut;          /* b1,b0... Pulse Output 1 type                    */
                                 /* b5,b4... Pulse Output 1 type                    */
                                 /* 00... Active, Import                            */
                                 /* 01... Active, Export                            */
                                 /* 10... Reactive, Import                          */
                                 /* 11... Reactive, Export                          */
                                 /* b3... 1 Pulse Output 1 on, 0 Pulse Output 1 off */
                                 /* b7... 1 Pulse Output 2 on, 0 Pulse Output 2 off */
                                 /* b2... 1 – RTC synchronisation enabled           */
                                 /* b2... 0 - RTC synchronisation disabled          */
        uint rkWh[2];            /* No. of impulses per 100Wh                       */
        uint Limit[2];           /* relay treshold value [0.01%]                    */
                                 /* at 0xFFFF relay function disabled               */
        uint Hysteresis[2];      /* relay treshold hysteresis [0.01%]               */
                                 /* hysteresis range is Limit +- Hysteresis         */
        uchar DelayTime[2];      /* relay action delay                              */
                                 /* 0-96 [ x 5 seconds]                             */
                                 /* 97-200 [ x 15 seconds]                          */
                                 /* 201-252 [ x 30 seconds ]                        */
        uchar ReleCfg[2];        /* b1,b0 – relay action control phase              */
                                 /* 00 - phase 1                                    */
                                 /* 01 - phase 2                                    */
                                 /* 10 - phase 3                                    */
                                 /* 11 – 3-phase value ( for powers, 3PF)           */
                                 /* b5,b4,b3,b2 -  relay action control quantity    */
                                 /* 0000 - U                                        */
                                 /* 0001 - I                                        */
                                 /* 0010 - PF                                       */
                                 /* 0011 - F                                        */
                                 /* 0100 - T                                        */
                                 /* 0101 - P                                        */
                                 /* 0110 - Q                                        */
                                 /* 0111 - S                                        */
                                 /* 1000 - cos                                      */
                                 /* 1001 – THDU                                     */
                                 /* 1010 – THDI                                     */
                                 /* b7,b6 – relay active value                      */
                                 /* 00 – over switch on                             */
                                 /* 01 - over switch off                            */
                                 /* 10 - below switch on                            */
                                 /* 11 – below switch off                           */
        uchar Res[3];            /* reserved                                        */
      } OCType;                  /*Outputs Configuration - 20 bytes                 */
//=========================================================================
```

ElmerTarif :

```
uchar ElmerTarif[6];
                /*each hyte contains four bit pairs,each pair bears tariff number of approp. hour */
                /* hour 00: byte 0, bits 1,0....00=tariff 0, 01=tariff 1, 10= tariff 2 */
                /* hour 23: byte 5, bits 7,6....00=tariff 0, 01=tariff 1, 10= tariff 2 */
                /* 11....tariff unspecified */
//=========================================================================
```

ElmerActData :

```
xdata union
        {
          uchar Arr[48];
          ulong St[3][4]; /* indexes -[tariff][work value]                                  */
        } ElmerEnergy;   /* 48 bytes                                                       */
                         /* electricity meter state since last clearing                    */
                         /* in watthours, in order AE,AI,RE,RI for 3 tariffs               */
                         /* VT and CT ratio not included                                   */

xdata union
        {
          uchar Arr[6];
        } ElmerClrTime; /* Elmeter & q-hPmax clear date&time                               */
                         /* length 6 bytes                                                 */

xdata long  QHMaxPwr[4];  /* quarter-hour active power maximum values P1, P2, P3, 3P       */
                         /* VT and CT ratio not included                                   */
xdata union
    {
        uchar Arr[6];
    } QHMaxPwrTime[4] ;  /* quarter-hour active power maximum time stamps                  */

//=========================================================================
```

ActAllData :

```
typedef struct
        {
        uchar RamErr;    // RamErr copy from Status
        uint U[3];       // phase voltages :
                         //coding: U [0.1V]: 0x0000 - 0V, 0x0001 - 0,1V, .... 0xFFFF – PwrOff
                         // VT ratio not included – if VT primary voltage
                         // defined (Mtn != 0xFFFFFFFF), the value to be multiplied
                         // by „Mtn“ (in Config structure) and divided by „NomU“
        uint LU;         // reserved
        int I[4];        // phase currents:
                         // coding:        if(I[n] == 0x0000) -> 0A
                         //                if(I[n] == 0x3E80) -> Inom (5A)
                         //                if(I[n] == 0x7FFF) -> PwrOff
                         // CT ratio not included – the value to be multiplied by
                         //  CT-ratio (see„Mtp“ in Config structure)
                         // the fourth value irrelevant
        char PF[3];      // phase power factors:
                         // coding : if(PF == 0) -> power factor = 0L
                         //                ...
                         //        if(PF == 90) -> power factor = 0.90L
                         //                ...
                         //        if(PF == 99) -> power factor = 0.99L
                         //        if(PF == 100) -> power factor = 1.00
                         //        if(PF == -99) -> power factor = 0.99C
                         //                ...
                         //        if(PF == -90) -> power factor = 0.90C
                         //                ...
                         //        if(PF == -1) -> power factor = 0.01C
                         //        if(PF == -100) -> power factor = 0C
```

```
        uchar Fr;
                        // frequency of U1
                        // coding :     if(Fr == 0) -> 37.2Hz, resolution    0.1 Hz
                        //              if(Fr == 1) -> 37.3Hz
                        //              ...
                        //              if(Fr == 177) -> 54.9Hz
                        //              if(Fr == 178) -> 55.0Hz, resolution  0.5 Hz
                        //              if(Fr == 179) -> 55.5Hz
                        //              if(Fr == 180) -> 56.0Hz
                        //              ...
                        //              if(Fr == 254) -> 93.0Hz
                        //              if(Fr == 255) -> value not defined
        uchar T;        // temperature – value in 0.1mA, need to be recalculated according
                        // Temp4mA and Temp20mA values (Config structure)
        uchar Contacts; // irrelevant
        char Kos[3];    // phase cos-phi values, coding like PF
        uint Upp[3];    // line voltages, coding like U
        long P[3];      // phase active powers
                        // coding :     1W ~ 0x4E200
                        //              0x7FFFFFFF... value not defined
                        // VT and CT ratio not included – if VT primary voltage
                        // defined (Mtn != 0xFFFFFFFF), the value to be
                        // multiplied by „Mtn“ (in Config structure) and
                        // divided by „NomU“
                        // then value to be multiplied by CT-ratio (see„Mtp“ in Config structure)
        long Q[3];      // reactive phase powers, coding like P
        long S[3];      // apparent phase powers, coding like P
        THDAHar thdHar[2]; // THD&Harmonics, [0]...voltage, [1]...current,  coding below
        } ActAllData;   // total 218 bytes


typedef struct{
        uchar THD[3];           /* coding :                                  */
                                /* 0 ~ 100 ...step 0.5%,    i.e. 0 ~ 50 %    */
                                /* 101 ~ 200 ... step 2.5%, i.e. 50,5 ~ 300 % */
                                /* 201 ~ 254 ... step 10%,  i.e. 310 ~ 840 %  */

        uchar Har[3][24];       /* coding :                                  */
                                /* 0 ~ 50 ... step 0.1%,    i.e. 0 ~ 5 %     */
                                /*51 ~ 70 ... step 0.5%     i.e. 5.5% ~ 15 % */
                                /* 71 ~ 90... step 2.5%,    i.e. 17.5 ~ 65 %  */
                                /* 91 ~126... step 5%,      i.e. 70 ~ 240 %   */
        }THDAHar;

//=========================================================================
```