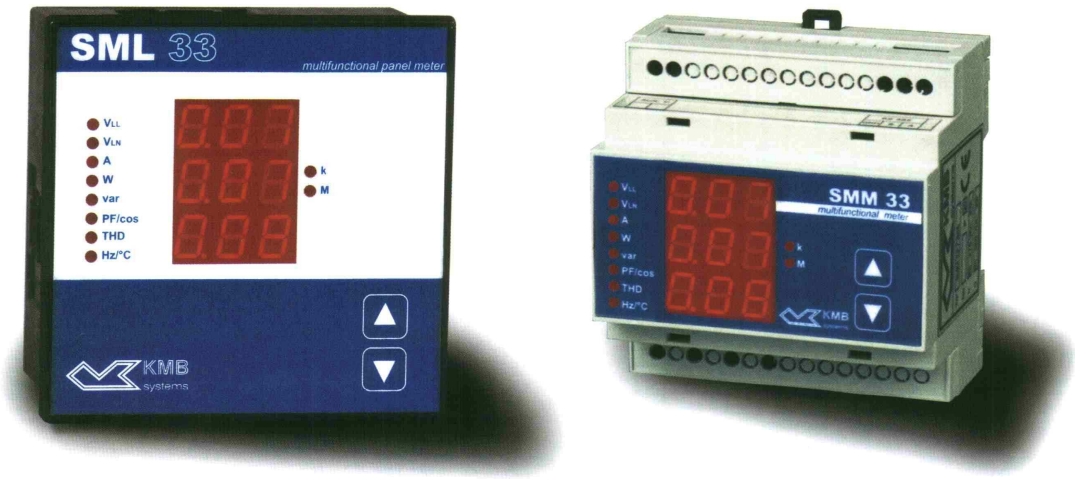
KMB systems, s.r.o.
Dr. M. Horákové 559, 460 06 Liberec 7, Czech Republic
tel. +420 485 130 314, fax +420 482 736 896
email : kmb@kmb.cz, internet : www.kmb.cz

SML 33 / SMM 33 / SMN 33

Description of SML/SMM/SMN 33 Communication Protocols.

Programmer's Manual



09/2008

1. Data Transmission Between SML/SMM/SMN 33 and Host System

SML 33, SMM 33 and SMN 33 instruments have the optional RS-485 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. "Config" contains instrument parameter settings (VT or CT conversion, etc.) to be read or written
4. "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 interface. 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 2,400 to 38,400 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
0x26	read instrument setting – Config
0x27	write instrument setting – Config
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 Reading Instrument Setting – Config (0x26)

The instrument provides information on its setting – Config.

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

Reply:
| 0x01 | 0x13 | 0x00 | Config | checksum |

2.1.4 Writing Instrument Setting – Config (0x27)

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

Command:
| 0x01 | 0x13 | 0x27 | Config | checksum |

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

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

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

The instrument sends the measurement data – ActAlldata.

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

Reply:
| 0x01| 0x5D| 0x00 | ActAlldata | checksum|

2.2 Modbus-RTU Communication Protocol

SML 33 / SMM 33 / SMN 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
04	Read Input Registers	reading currently data – ActAllData
06	Preset Single Register	writing Config
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
CONFIG	0x0700

2.2.1 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	2
0x0004	Remote Address	2

The DeviceType parameter follows.

Instrument Model	Value
SML 33	0x1000
SMM 33	0x1001
SMN 33	0x1002

2.2.2 Config

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

Offset	Parameter	Length in Bytes
0x0000 + 0x0001	Voltage Transformer (VT)	4
0x0002 + 0x0003	Current Transformer (CT)	4
0x0004	Default Frequency	2
0x0005	Input Type	1
0x0006	Device Addr	1
0x0007	Remote BdRate	1
0x0008	Displayable Values	2
0x0009	Display Manner	2

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

2.2.3 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).

a) SML/SMM Instruments

Register Address	Parameter	Length in Bytes
0x0000-0x0001	ULN1	4
0x0002-0x0003	ULN2	4
0x0004-0x0005	ULN3	4
0x0006-0x0005	I1	4
0x0008-0x0009	I2	4
0x000A-0x000B	I3	4
0x000C-0x000D	ULL1	4
0x000E-0x000F	ULL2	4
0x0010-0x0011	ULL3	4
0x0012-0x0013	P1	4
0x0014-0x0015	P2	4
0x0016-0x0017	P3	4
0x0018	FI1*10000	2
0x0019	FI2*10000	2
0x001A	FI3*10000	2
0x001B	UTHD*100	2
0x001C	UTHD*100	2
0x001D	UTHD*100	2

0x001E	ITHD*100	2
0x001F	ITHD*100	2
0x0020	ITHD*100	2
0x0021	UTHDA*100	2
0x0022	UTHDA*100	2
0x0023	UTHDA*100	2
0x0024-0x0025	VAR1	4
0x0026-0x0027	VAR2	4
0x0028-0x0029	VAR3	4
0x002A	Temperature*100	2
0x002B	Frequency*100	2
0x002C	Meter Status	2
0x002D-0x002E	3f - P	4
0x002F-0x0030	3f - VAR	4

b) SMN Instruments (have additional current In)

Register Address	Parameter	Length in Bytes
0x0000-0x0001	ULN1	4
0x0002-0x0003	ULN2	4
0x0004-0x0005	ULN3	4
0x0006-0x0005	I1	4
0x0008-0x0009	I2	4
0x000A-0x000B	I3	4
0x000C-0x000D	In	4
0x000E-0x000F	ULL1	4
0x0010-0x0011	ULL2	4
0x0012-0x0013	ULL3	4
0x0014-0x0015	P1	4
0x0016-0x0017	P2	4
0x0018-0x0019	P3	4
0x001A	FI1*10000	2
0x001B	FI2*10000	2
0x001C	FI3*10000	2
0x001D	UTHD*100	2
0x001E	UTHD*100	2
0x001F	UTHD*100	2
0x0020	ITHD*100	2
0x0021	ITHD*100	2
0x0022	ITHD*100	2
0x0023	UTHDA*100	2
0x0024	UTHDA*100	2
0x0025	UTHDA*100	2
0x0026-0x0027	VAR1	4

0x0028-0x0029	VAR2	4
0x002A-0x002B	VAR3	4
0x002C	Temperature*100	2
0x002D	Frequency*100	2
0x002E	Meter Status	2
0x002F-0x0030	3p - P	4
0x0031-0x0032	3p - VAR	4

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

Config :

```
typedef  struct {
        ulong Mtn;                  // VT conversion, 0xffffffff = not used
        ulong Mtp;                  // CT conversion, 0xffffffff = not used
        uint  DefaultFreq;          // default frequency if no measured value is available
        uchar InputType;            // measurement inputs configuration
                                    // b0-b3 – not used
                                    // b4-b6 connection configuration
                                    //       000 – single phase
                                    //       001 – two-phase
                                    //       010 – three-phase Y
                                    //       011 – three phase delta
                                    //       100 – Aaron wiring
                                    // b7    0 – measurement using VT
                                    //       1 – direct measurement
        uchar   DeviceAddr;         // address (1÷253)
        uchar   RemoteBdRate;       // low nibble specifies baud rate
                                    // for standard asynchronous communication
                                    //       0 – 2400  Bd
                                    //       1 – 4800  Bd
                                    //       2 – 9600  Bd
                                    //       3 – 19200 Bd
                                    //       4 – 38400 Bd
        uint    DisplayableValues;  // bitmap value determines whether data whose number is specified as bit
                                       position increased by 1 is selectable with menu navigation buttons
        uchar   DisplayManner /SelectedDisplayValue/
                                    // low nibble –data selected for continual view
                                    //       1 – line voltage UL-L
                                    //       2 – phase voltage UL-N
                                    //       3 – current IL
                                    //       4 – active phase power
                                    //       5 – active three-phase power
                                    //       6 – reactive phase power
                                    //       7 – reactive three-phase power
                                    //       8 – phase power factor
                                    //       9 – three-phase power factor
                                    //       10 – average cosine (1st harmonic DPF only)
                                    //       11–harmonic distortion in UL-L
                                    //       12 – harmonic distortion in UL-N
                                    //       13 – harmonic distortion in IL
                                    //       14 – frequency
                                    //       15 – temperature in instrument location
                                    // high nibble  0– display cycles through selected data in three-second intervals
                                    //       1 – display keeps showing last data selected with button
                                    //       2 – display shows data selected with button for 10 seconds
                                    //       and then data selected for continual view

} tConfig;                          // Configuration 16 byte
```

ActAllData :

SML/SMM Instrument Format :

```
typedef struct{
        float uLN[3];       // phase voltages          [float IEEE-574]
        float i[3];         // phase currents          [float IEEE-574]
        float uLL[3];       // line voltages           [float IEEE-574]
        float p[3];         // power                   [float IEEE-574]
        int fi[3];          // angle multiplied by ten thousand to calculate cos(phi) in radians
        int uTHD[3];        // phase voltage harmonic distortion as percentage multiplied by 100
        int iTHD [3];       // current harmonic distortion as percentage multiplied by 100
        int uTHDA[3];       // line voltage harmonic distortion as percentage multiplied by 100
        float var[3];       // reactive power          [float IEEE-574]
        int temperature;    // temperature in degrees centigrade multiplied by 100
        int frequency;      // frequency in Hertz multiplied by 100
        byte CFGChng;       // configuration alteration counter, increased by 1 with each alteration, from 255 to 0
        byte ErrStat;       // meter status – bitmap, status indicated by relevant bit being set (value 1)
                            // bit 0 – meter not configured
                            // bit 1 – EEPROM checksum error
                            // bit 2 – EEPROM has been restored from backup
                            // bit 3 through 6 – reserved
                            // bit 7 – frequency not detected, default value applied
} tAllActData;
```

SMN Instrument Format (additional current In = I4 in compare with the SML/SMM):

```
typedef struct{
        float uLN[3];       // phase voltages          [float IEEE-574]
        float i[4];         // phase currents          [float IEEE-574]
        float uLL[3];       // line voltages           [float IEEE-574]
        float p[3];         // power                   [float IEEE-574]
        int fi[3];          // angle multiplied by ten thousand to calculate cos(phi) in radians
        int uTHD[3];        // phase voltage harmonic distortion as percentage multiplied by 100
        int iTHD [3];       // current harmonic distortion as percentage multiplied by 100
        int uTHDA[3];       // line voltage harmonic distortion as percentage multiplied by 100
        float var[3];       // reactive power          [float IEEE-574]
        int temperature;    // temperature in degrees centigrade multiplied by 100
        int frequency;      // frequency in Hertz multiplied by 100
        byte CFGChng;       // configuration alteration counter, increased by 1 with each alteration, from 255 to 0
        byte ErrStat;       // meter status – bitmap, status indicated by relevant bit being set (value 1)
                            // bit 0 – meter not configured
                            // bit 1 – EEPROM checksum error
                            // bit 2 – EEPROM has been restored from backup
                            // bit 3 through 6 – reserved
                            // bit 7 – frequency not detected, default value applied
} tAllActData;
```